HideZeroOne
INE – Cyber Sec
www.hide01.ir

1 / 152 00:00 / 00:00

PREV NEXT

OUTLINE

Table of Contents

Module 01 | Incident Handling Process

PREV NEXT

OUTLINE

Learning Objectives

By the end of this module, you should have a better understanding of:

- ✓ The Incident Handling* process.
- ✓ The mission, structure, scope, activities, and responsibilities of an Incident Handling Team.

**In this course's context, the terms "Incident Handling" and "Incident Response" are synonymous.*

OUTLINE

1.1

Incident Handling Definition & Scope

PREV NEXT

OUTLINE

1.1.1 Incident Handling Definition

Incident Handling is the well-defined course of action whenever a computer or network security incident occurs.

PREV NEXT

OUTLINE

1.1.1 Incident Handling Definition

According to the Computer Security Incident Handling Guide by NIST, only events with negative consequences are considered security incidents.

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

OUTLINE

1.1.1 Incident Handling Definition

Such events can be:

- System crashes,
- Packet floods,
- Unauthorized use of system privileges,
- Unauthorized access to sensitive data, and
- Execution of destructive malware.

PREV NEXT

OUTLINE

1.1.1 Incident Handling Definition

NOTE

SOCs or CSIRT teams are known to suffer from alert fatigue; this is why during this course we will show you which events and alerts deserve your utmost attention, in addition to making you comfortable with a variety of log formats and “context-providing” techniques.

OUTLINE

1.1.2 Incident Handling Scope

It should also be noted, that incident handling is not only about intrusions.

Malicious insiders, availability issues and loss of intellectual property all fall under the scope of incident handling as well.

PREV NEXT

OUTLINE

1.2

Incident Handling Process

PREV NEXT

OUTLINE

1.2 Incident Handling Process

As an incident handler, your daily activities will include discussing how an attacker attempted or managed to break into a system, in addition to preventing, detecting and responding to such attempts.

PREV NEXT

OUTLINE

1.2 Incident Handling Process

An incident handler should be completely aware of attacker techniques, tactics, and procedures.

Specifically, he/she should possess a good understanding of how attackers operate during all stages of the cyber kill chain.

https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

PREV NEXT

OUTLINE

1.2 Incident Handling Process

Then, and only then, can an incident handler be in the position of not only anticipating attacks but also proposing defensive measures against them.

PREV NEXT

OUTLINE

1.2 Incident Handling Process

All the established incident handling guides and processes were built to help organizations prepare, defend and respond to all stages of the cyber kill chain and effectively counteract intrusions in general.

PREV NEXT

OUTLINE

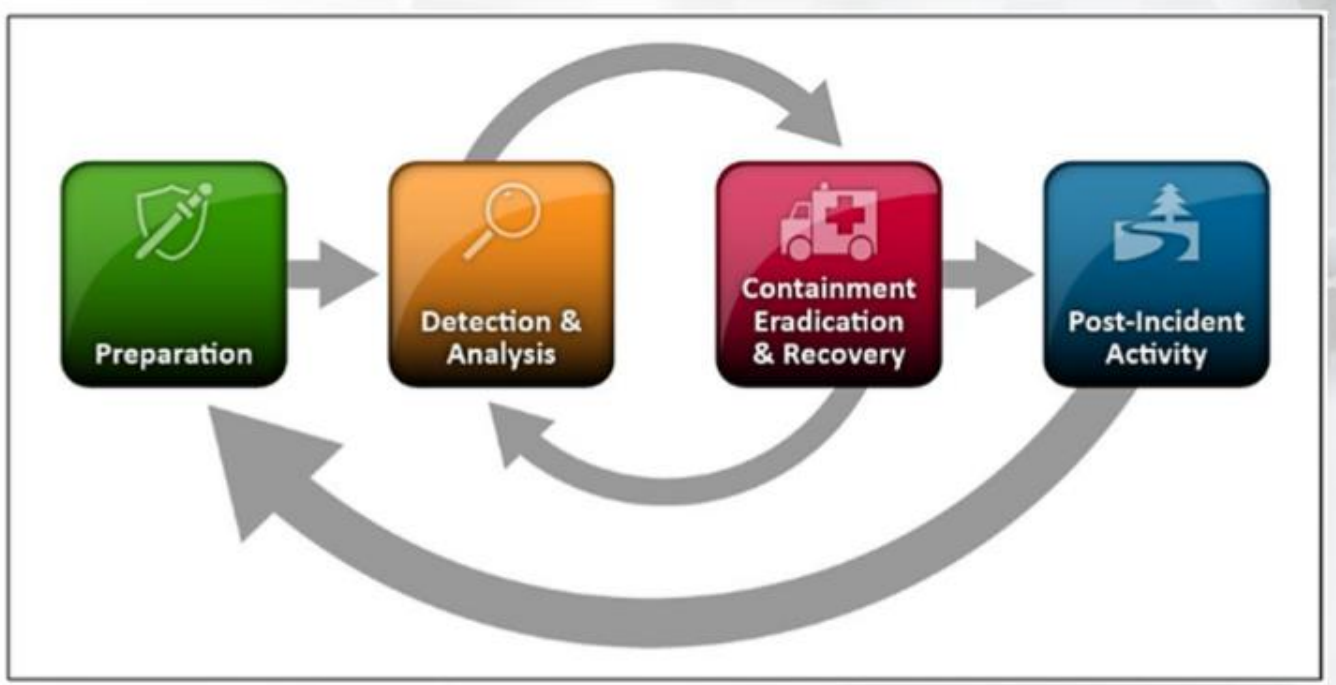
1.2 Incident Handling Process

According to NIST, the incident handling process consists of four (4) phases:

- Preparation
- Detection & Analysis
- Containment, Eradication & Recovery
- Post-Incident Activity

OUTLINE

1.2 Incident Handling Process



PREV NEXT

OUTLINE

1.2 Incident Handling Process

Those four (4) phases of the incident handling process are also known as the ***incident response life cycle***.

They can be seen as a roadmap for incident handlers so that they know what they should do and how to proceed next when handling and responding to an incident.

OUTLINE

1.2.1 Incident Handling Process – Preparation

PREV NEXT

OUTLINE

1.2.1 Incident Handling Process – Preparation

The Preparation phase of the incident handling process includes everything related to an organization's incident handling readiness.

Employees **Documentation** **Defensive Measures**

PREV NEXT

OUTLINE

1.2.1 Incident Handling Process – Preparation

Employees

Documentation

Defensive Measures

- A Skilled Response Team
- IT Security Training
- Security Awareness/Social Engineering Exercises, etc.

PREV NEXT

OUTLINE

1.2.1 Incident Handling Process – Preparation

Employees

Documentation

Defensive Measures

- Well-defined policies

PREV NEXT

1.2.1 Incident Handling Process – Preparation

Employees | **Documentation** | Defensive Measures

- **Well-defined policies**

Ensure you have the right to monitor and collect the required amount of evidence. Advice from the legal department is required.

OUTLINE

1.2.1 Incident Handling Process – Preparation

Employees

Documentation

Defensive Measures

- Well-defined policies
- Well-defined response procedures

1.2.1 Incident Handling Process – Preparation

Employees | **Documentation** | Defensive Measures

- Well-defined policies
- **Well-defined response procedures**

Based on those, you will decide how you will handle “major” incidents.
- Should the respective cybercrime unit be notified?
- Contain immediately or closely monitor the intruder?

Agreement of the upper-management is required.

OUTLINE

1.2.1 Incident Handling Process – Preparation

Employees

Documentation

Defensive Measures

- Well-defined policies
- Well-defined response procedures
- Breach/incident communication plan(s)
- Maintaining a chain of custody of actions

OUTLINE

1.2.1 Incident Handling Process – Preparation

Employees

Documentation

Defensive Measures

- A/V, (H)IDS, DLP, EDR, Security Patches
- SIEM, UTM, Threat Intelligence
- NSM, Central Logging, Honeypots, etc.

1.2.1 Incident Handling Process - Preparation

Preparation Key Points

- Multi-disciplinary team:
  - Incident Handlers | Forensic Analysts | Malware Analysts | Support from NOC, Legal, PR Depts.
- Determine scheduling / minimum time to respond.
- Incident handlers should have unrestricted and on-demand access to systems.

PREV NEXT

1.2.1 Incident Handling Process - Preparation

Preparation Key Points (cont.)

- Establish a SPOC.
- Establish effective reporting capabilities.
- Incident Handling Starter Kit:
  - Data Acquisition software | Read-only diagnostic software | Bootable Linux environment | HDs, Ethernet TAP, Cables, Laptop

PREV NEXT

1.2.2 Incident Handling Process – Detection & Analysis

Preparation Additional Information

In the previous slides, we summarized the most important aspects of the Preparation phase.

For more information, please refer to Computer Security Incident Handling Guide by NIST.

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

1.2.2 Incident Handling Process – Detection & Analysis

PREV NEXT

OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

The Detection & Analysis phase of the incident handling process includes everything related to detecting an incident:

- Means of detection: Sensors (FW, IDS, Agents, Logs, etc.) | Personnel (Need to be trained)
- Information and knowledge sharing
- Context-aware threat intelligence
- Segmentation of the architecture
- Good understanding of / visibility in your network

OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

Detection & Analysis Key Points

- Assign a Primary Incident Handler:
  - Usually serves under the incident handling team's manager
  - In charge when handling incidents of specific class and severity
- Establish trust and effective information sharing.
- Safeguard information sharing:
  - In case of a network under your supervision being compromised, alternative communications should be established. Good options are cloud-based services featuring end-to-end encryption, emails powered by PGP, S/MIME, etc. Avoid solutions that can be intercepted by an attacker in a privileged network position (Man in The Middle)

OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

Detection & Analysis Key Points (cont.)

- Establish levels of detection by logically categorizing your network
  - An effective and actionable way to logically categorize your network is by considering the following levels:

    Network perimeter | *Host perimeter* | *Host-level* | *Application-level*

- Establish baselines, extend visibility, and know your limits:
  - Catching all intrusions or intrusion attempts is unlikely. Advanced adversaries exist.
  - Admins, SOC/CSIRT members are expected to be able to spot deviations from normal network or host state/behavior (this requires baselining).
  - Visibility should be extended (and fine-tuned) as much as possible, so that data exist for later analysis.

1.2.2 Incident Handling Process – Detection & Analysis

Establish levels of detection by logically categorizing your network

Let’s analyze the Detection & Analysis key point above.

OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

As already covered, an effective and actionable way to logically categorize your network is by considering the following levels:

- Network perimeter
- Host perimeter
- Host-level
- Application-level

PREV NEXT

OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

Let’s see some detection examples at each of the levels we just mentioned.

OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

Detection at the **network perimeter** level occurs, as the name suggests, on the network.

Firewalls, internet-facing NIDS, IPS, DMZ systems, etc. can assist such detection activities.

PREV NEXT

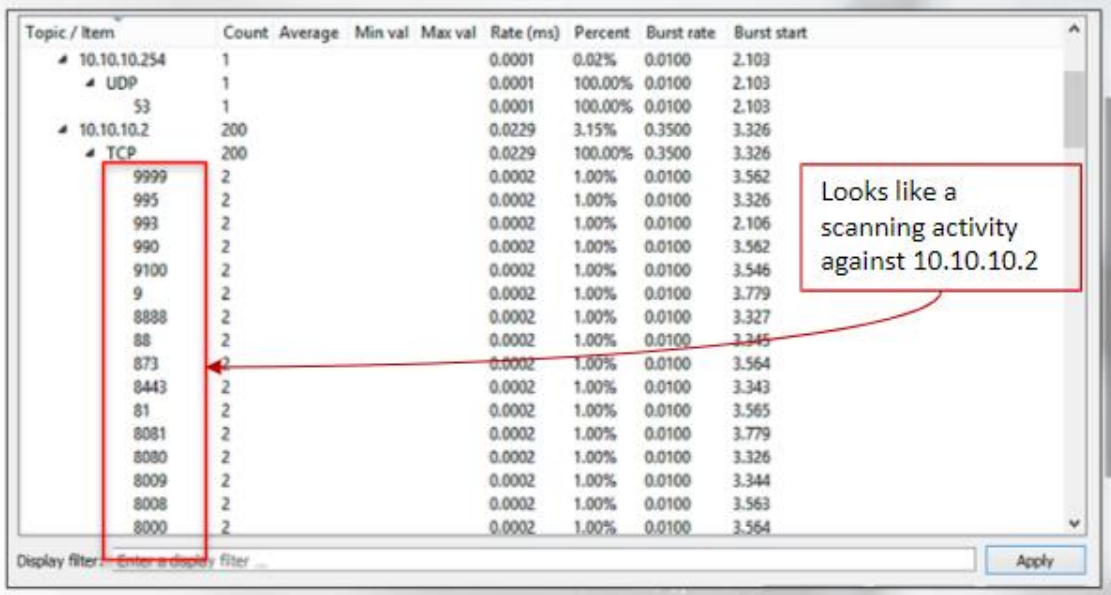
OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

Network perimeter detection example

Let's suppose that we are analyzing a given packet capture in Wireshark and we go to *Statistics -> IPv4 Statistics -> Destinations and Ports*.

What we see is the following.



https://www.wireshark.org/

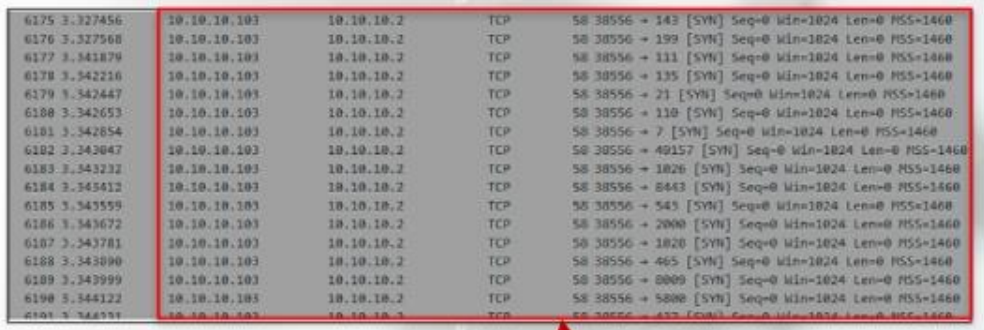
PREV NEXT

OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

Network perimeter detection example

We can also make sure we are dealing with a scanning case by looking at the packet capture sequence.



This is clearly a scanning activity against 10.10.10.2, initiated by 10.10.10.103.

OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

Network perimeter detection example

This is an example of detection at the network perimeter level since we analyzed packets crossing the network.

OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

Detection at the **host perimeter** level occurs whenever we analyze data a host receives from the network or sends out to the network.

Local firewalls or HIPS systems can assist such detection activities.

OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

Network perimeter detection examples

Let's suppose that we are analyzing a host and specifically, we are checking its network and internet connections using *netstat*.

We come across the following.

-o flag: Show the ProcessID
-b: Show the listening EXE and associated DLLs

```
>> netstat -naob
```

```
Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING       692
  RpcSs
 [svchost.exe]
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING       4
 Can not obtain ownership information
  TCP    0.0.0.0:3389           0.0.0.0:0              LISTENING       404
  CryptSvc
 [svchost.exe]
  TCP    0.0.0.0:5985           0.0.0.0:0              LISTENING       4
 Can not obtain ownership information
  TCP    0.0.0.0:9999           0.0.0.0:0              LISTENING       5328
 [winlogin.exe]
  TCP    0.0.0.0:47001          0.0.0.0:0              LISTENING       4
 Can not obtain ownership information
  TCP    0.0.0.0:49152          0.0.0.0:0              LISTENING       472
 [wininit.exe]
```

A suspicious looking executable named winlogin.exe is listening on port 9999.

1.2.2 Incident Handling Process – Detection & Analysis

Network perimeter detection examples

Suppose now, that you were informed about other organizations of your industry being targeted by a Linux malware that utilizes port 22 for its communications.

Let's proactively check the network and internet connections of a critical Linux-based host.

1.2.2 Incident Handling Process – Detection & Analysis

Network perimeter detection examples

We could have done so by executing the below.

```
>> lsof -i :22
```

Do you think we are safe?

```
:~# lsof -i :22
:~#
```


PREV NEXT

OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

Network perimeter detection examples

When it comes to detection, “Redundancy” is a good thing.

So, let’s make sure by performing the below redundant check.

```
>> netstat -anp | grep :22
```

Output excerpt...

```
[redacted]:~# netstat -anp | grep :22
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 172.1.1.136:22          172.1.1.1:39967         ESTABLISHED -
```

1.2.2 Incident Handling Process – Detection & Analysis

Network perimeter detection examples

What happened here is that the Linux-based malware utilized a kernel-level rootkit to conceal the SSH daemon process in the kernel process table.

- "*lsof*" scans the process table, this is why it missed the concealed process and the accompanying open sockets
- "*netstat*" on the other hand, focuses on the open socket list, and only if we instruct it (-p flag), it tries to locate the associated processes in the process table. Regardless of "netstat" finding associated processes or not, it will still output open sockets.

http://www.dmi.unipg.it/bista/didattica/sicurezza-pg/seminari2008-09/seminario_neri/seminario_neri.pdf

1.2.2 Incident Handling Process – Detection & Analysis

Network perimeter detection examples

The bottom line is, “Redundancy” and taking advantage of all the available toolkit is advised (taking into consideration time constraints of course).

OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

Network perimeter detection examples

The previously mentioned cases were examples of detection at the host perimeter level since we analyzed data coming in and out of the hosts.

OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

Whenever performing detection activities at the host (or network) perimeter level, consider the following:

1. Utilize packet destinations (network perimeter) and identified ports (network and host perimeter) to identify the running services at the respective host, using internet resources such as IANA.

2. Are the identified services actually running and part of your organization?

3. If not, check for port abuse through resources, such as https://www.speedguide.net/ports.php, to identify possible malware.

Example: We see a packet trying to reach port 21 of a host, or we see a host listening on port 21. It could be FTP traffic if our organization includes such functionality or malicious traffic if it doesn't.

http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

1.2.2 Incident Handling Process – Detection & Analysis

Detection at the **host level** occurs whenever we analyze data residing in the host.

A/Vs and EDR solutions, as well as the users themselves, can assist in such detection activities.

OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

Host-level detection example

An example of host-level detection is a user being warned about a malicious executable by the host's endpoint protection solution.

	Filename	Risk	Action	Risk Type
	cloudcar.exe	Bloodhound.Sonar.9	Quarantined	Application
	cloudcar.exe	Bloodhound.Sonar.9	Quarantined	Application
	socar.exe	Bloodhound.Sonar.9	Quarantined	Application

OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

Detection at the **application level** occurs whenever we analyze application logs.

Web application logs, service logs, etc. can assist in such detection activities since they offer valuable insight, such as user operations, user input, etc., all accompanied by the respective time they were executed/submitted.

PREV NEXT

OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

Application-level detection example

An example of application level detection is an analyst spotting abnormal behavior when statistically analyzing IIS logs.

cs-uri-stem	cs-uri-query	Total	Max Time	Avg Time
/CMSCore/SiteContent/Mock/shelly.asp	[redacted]	1	18637	18637
/CMSCore/SiteContent/Mock/shelly.asp	[redacted]	1	18537	18537
/		2	1772	1518
/CMSCore/SiteContent/Mock/minion.aspx		5	374	118

Overly long execution times **may** indicate the existence of a web shell.

1.2.2 Incident Handling Process – Detection & Analysis

Log-reviewing resource examples:

- https://docs.microsoft.com/en-us/azure/security/azure-log-audit
- http://httpd.apache.org/docs/current/logs.html

1.2.2 Incident Handling Process – Detection & Analysis

System Administrators & Detection

Administrators can play a crucial role when it comes to detection.

At the end of this module, in the Appendix, are two cheat sheets that can assist administrators in detecting abnormal processes/services, files, network usage, scheduled tasks, user accounts, third-party components, etc.

Go to the Appendix on slide/page 135 to view the cheat sheets.

PREV NEXT

OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

System Administrators & Detection

The provided cheat sheets are meant to be used to detect most common intrusions.

Don't worry, this is just the beginning, in the next modules we will make you capable of detecting the vast majority of attacks, including the most advanced and evasive ones.

Go to the Appendix on slide/page 135 to view the cheat sheets.

1.2.2 Incident Handling Process – Detection & Analysis

System Administrators & Detection

Make sure periodic calls with Administrators are scheduled to go through any abnormalities being spotted and also fine-tune this proactive and cheat sheet-based detection methodology.

Go to the Appendix on slide/page 135 to view the cheat sheets.

OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

Finally, before calling an event an actual incident, consider the following:

- Could this be a user oversight?
- Could this actually be the case or is this an improbability?
- Scrutinize all evidence.
- Base your decision on prior knowledge of the normal behavior.

1.2.2 Incident Handling Process – Detection & Analysis

In case you are handling an actual incident, ask yourself the following damage-estimation questions:

- Identify the vulnerability's exploitation impact.
  (Remote code execution should be handled much more quickly than information disclosure.)
- Are there any crown jewels that can be affected?
- What are the minimum requirements for effective exploitation?
  (A privileged position within the LAN; just an internet connection; Valid credentials; Default config; etc.?)
- Is this being actively exploited in the wild?
  (You can assess the adversary's sophistication level this way.)
- Is there a proposed remediation strategy?
- Is there threat intel/evidence that suggests increased spreading capabilities?

OUTLINE

1.2.2 Incident Handling Process – Detection & Analysis

Detection & Analysis Additional Information

In the previous slides, we summarized the most important aspects of the Detection & Analysis phase.

For more information, please refer to Computer Security Incident Handling Guide by NIST.

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

OUTLINE

1.2.3 Incident Handling Process – Containment, Eradication & Recovery

OUTLINE

1.2.3 Incident Handling Process – Containment, Eradication & Recovery

The Containment, Eradication & Recovery phase of the incident handling process, includes everything related to:

- Preventing an incident from getting worse (i.e., preventing the intruder from getting any deeper) ← Containment
- Eliminating intruder artifacts, understanding the root cause, attack vectors and TTPs in general, utilizing backups and improving ← Eradication
- Restoring and monitoring to make sure nothing evaded detection ← Recovery

PREV NEXT

OUTLINE

1.2.3 Incident Handling Process – Containment, Eradication & Recovery

Let’s start by looking into the activities the Containment phase consists of.

PREV NEXT

OUTLINE

1.2.3 Incident Handling Process – Containment, Eradication & Recovery

Containment is divided into the following subphases:

Short-Term Containment

Render the intrusion ineffective

System Back-Up

Long-Term Containment

Make sure the intruder is locked out of the affected host and network

PREV NEXT

OUTLINE

1.2.3.1 Incident Handling Process – Before Containment

Let's suppose an incident was declared, and you arrive on site.

After going over the information collected during the Detection & Analysis phase, you should:

- Identify if you are dealing with a malicious insider or not,
- Isolate the area under investigation, and
- Utilize incident casualty forms.

PREV NEXT

OUTLINE

1.2.3.1 Incident Handling Process – Before Containment

Since we are now dealing with a declared incident, we need to classify it based on the information we analyzed, its possible impact, and its extend.

Type Impact Extent

1.2.3.1 Incident Handling Process – Before Containment

Incident Classification

Type | Impact | Extent

- Denial of Service
- External Exploitation
- Internal Exploitation
- Information Leakage
- Malware
- Malicious Email

1.2.3.1 Incident Handling Process – Before Containment

Incident Classification

Type **Impact** Extent

- Incident affecting critical system(s)
- Incident affecting non-critical system(s)

OUTLINE

1.2.3.1 Incident Handling Process – Before Containment

Incident Classification

Type — **Impact** — Extent

- Incident affecting critical system(s)
- Incident affecting non-critical system(s)

Impact is tightly connected with the **Response Time.**

OUTLINE

1.2.3.1 Incident Handling Process – Before Containment

Incident Classification

Type — **Impact** — Extent

- Incident affecting critical system(s)
- Incident affecting non-critical system(s)
- Incident affecting asset that requires no immediate investigation

OUTLINE

1.2.3.1 Incident Handling Process – Before Containment

Incident Classification

Type | Impact | **Extent**

- Extensive compromise, including sensitive customer information
- Manageable intrusion and spreading

OUTLINE

1.2.3.1 Incident Handling Process – Before Containment

Incident Classification

Type | Impact | **Extent**

- Extensive compromise, including sensitive customer information
- Manageable intrusion and spreading

Extent is tightly connected with the **escalation level**. For example, should the CISO's office or upper management be informed?

OUTLINE

1.2.3.1 Incident Handling Process – Before Containment

Incident Classification

Type — Impact — **Extent**

- Extensive compromise, including sensitive customer information
- Manageable intrusion and spreading
- Immediately detected or easily contained intrusion

OUTLINE

1.2.3.1 Incident Handling Process – Before Containment

Incident Classification

Below is a great resource on classifying incidents from the *Forum of Incident Response and Security Teams* (FIRST):

https://www.first.org/resources/guides/csirt_case_classification.html

OUTLINE

1.2.3.1 Incident Handling Process – Before Containment

Note that when it comes to escalating an incident or needing help in handling it, there should be a senior management member (CIO/CISO/head of the legal dept., etc.) who is “close” to the incident handling team.

OUTLINE

1.2.3.1 Incident Handling Process – Before Containment

Incident Communication

Such a member should be the first upper management individual who will be informed of an incident and provided with notes from the first responder.

OUTLINE

1.2.3.1 Incident Handling Process – Before Containment

Incident Communication

Also, note that the communication flow should include both security and other management individuals.

This will ensure that the affected business units will be kept in the loop.

OUTLINE

1.2.3.1 Incident Handling Process – Before Containment

There will be times when the incident handling team will be required to handle multiple incidents; this is exactly why there should be an incident tracking mechanism.

Take into consideration that system administrators, help desks, and the established incident reporting mechanism may all create tickets for the same incident.

PREV NEXT

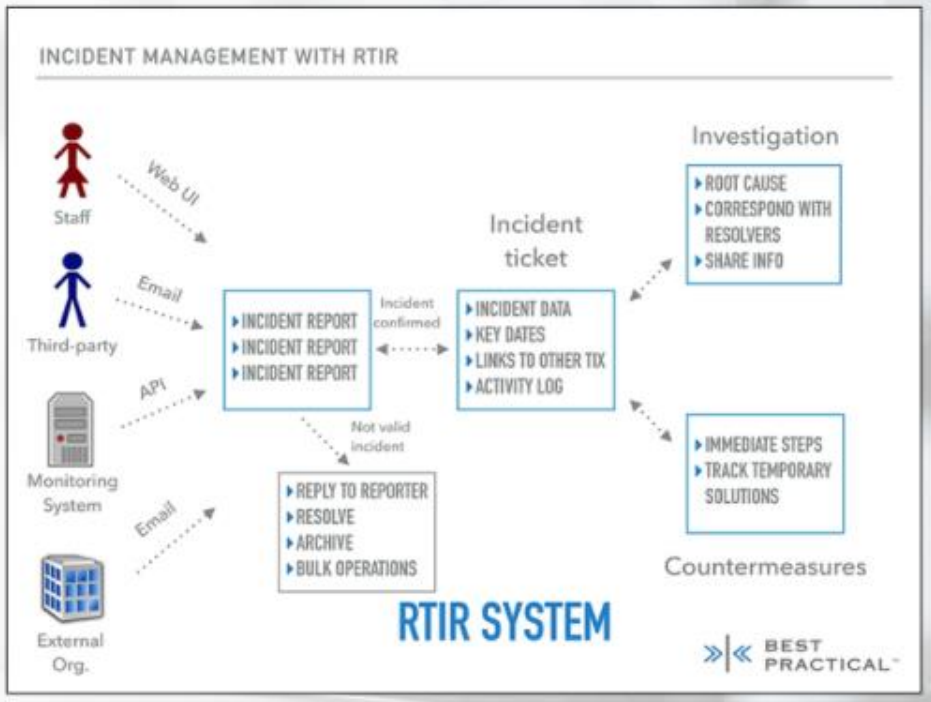
OUTLINE

1.2.3.1 Incident Handling Process – Before Containment

Incident Tracking

Tools such as Request Tracker for Incident Response (RTIR) should be in place to consolidate all tickets and information for more effective response activities.

More incident management tools can be found in the following repository:
https://github.com/meirwah/awesome-incident-response#incident-management



https://bestpractical.com/rtir/

OUTLINE

1.2.3.1 Incident Handling Process – Before Containment

Before we even start the Containment process, we should be extremely careful not to let the attacker(s) know our operations; this means, no submitting of identified binaries to cloud A/V solutions and no interacting with any identified IP addresses just yet.

Act normally…

PREV NEXT

OUTLINE

1.2.3.1 Incident Handling Process – Before Containment

As we mentioned previously, Containment is divided into the following subphases:

- Short-term Containment
- System Back-up
- Long-term Containment

PREV NEXT

OUTLINE

1.2.3.2 Incident Handling Process – Short-term Containment

Let's start with Short-term Containment.

During this subphase, we should try to render the intrusion ineffective, without altering the machine's hard drive (we need to image it for forensic activities).

To do so, we can disable network connectivity or even disconnect the machine from the power line, in extreme occasions.

PREV NEXT

OUTLINE

1.2.3.2 Incident Handling Process – Short-term Containment

Let's start with Short-term Containment.

During this subphase, we should try to render the intrusion ineffective without altering the machine's hard drive (we need to image it for forensic activities).

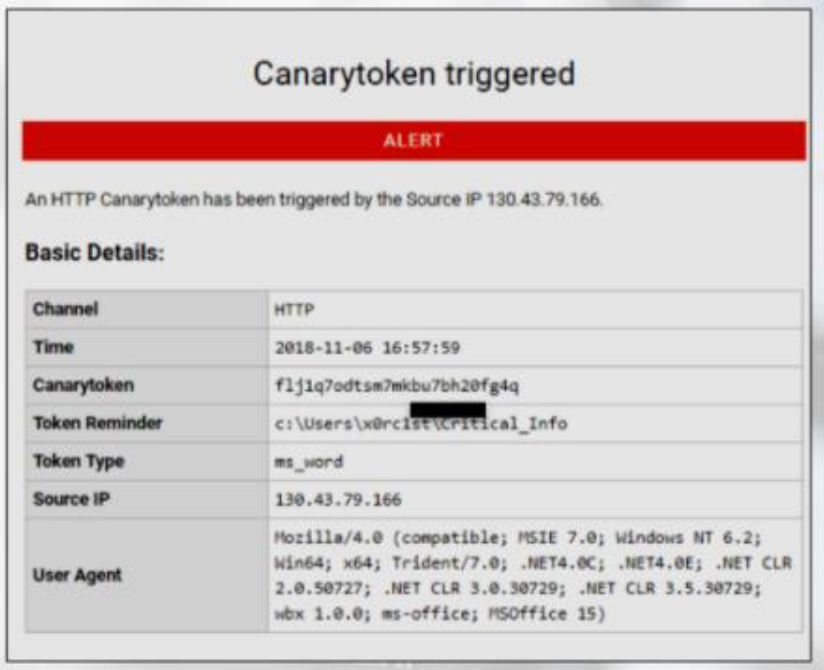
To do so, we can disable network connectivity or even disconnect the machine from the power line, in extreme occasions.

- Place the machine in a separate/isolated VLAN
- Change DNS
- Isolate the machine through router or firewall configurations

Always formally inform the respective business unit manager if you decide to do so, even ask permission.

1.2.3.2 Incident Handling Process – Short-term Containment

TIP: As an additional defense measure to track attackers, you could use http://canarytokens.org/generate to generate documents that call back to a designated email or webhook URL the moment they are accessed, mentioning the source IP address and other information.



Canarytoken triggered

ALERT

An HTTP Canarytoken has been triggered by the Source IP 130.43.79.166.

Basic Details:

Channel	HTTP
Time	2018-11-06 16:57:59
Canarytoken	flj1q7odtsm7mkbu7bh20fg4q
Token Reminder	c:\Users\x0rc1st\Critical_Info
Token Type	ms_word
Source IP	130.43.79.166
User Agent	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2; Win64; x64; Trident/7.0; .NET4.0C; .NET4.0E; .NET CLR 2.0.50727; .NET CLR 3.0.30729; .NET CLR 3.5.30729; wbx 1.0.0; ms-office; MSOffice 15)

1.2.3.3 Incident Handling Process – System Back-Up

The next step is to make images of the affected system(s) for forensics activities.

Before we go deeper into imaging, there is an important note to remember.

PREV NEXT

OUTLINE

1.2.3.3 Incident Handling Process – System Back-Up

Data Acquisition

To preserve the evidence, you're **not** supposed to work on the original machine when investigating and you're also **not** supposed to analyze and work on the first image you take.

PREV NEXT

OUTLINE

1.2.3.3 Incident Handling Process – System Back-Up

Data Acquisition

The original image is usually verified (which we'll cover later) and then saved alongside other parameters to protect it from tampering, while all the work is done on copies of the original image.

OUTLINE

1.2.3.3 Incident Handling Process – System Back-Up

Data Acquisition

When it comes to data acquisition, we have to consider the order of volatility.

For example, the data within the machine's RAM should be acquired first, since they are a lot more volatile than their on-disk counterparts.

PREV NEXT

OUTLINE

1.2.3.3 Incident Handling Process – System Back-Up

Data Acquisition

The storage mediums can be arranged from the most volatile to the least, as follows:

OUTLINE

1.2.3.3 Incident Handling Process – System Back-Up

Data Acquisition

Types of Data Acquisition:

Data acquisition techniques and methods can be divided into:

Static Acquisition

Dynamic / Live Acquisition

Choosing which technique to apply depends on data volatility and the incident

OUTLINE

1.2.3.3 Incident Handling Process – System Back-Up

Data Acquisition

Static Acquisition is the acquiring process of data that are not volatile. By not volatile, we mean data that will not be affected by a system restart.

Such acquisition is usually performed on hard disks and flash disks.

PREV NEXT

OUTLINE

1.2.3.3 Incident Handling Process – System Back-Up

Data Acquisition

Dynamic Acquisition is the acquiring process of data that are volatile. By volatile we mean data that will be heavily altered or even lost by any user action or system restart.

Such acquisition is usually performed while a system is still powered on and without performing any prior actions. As running processes use RAM, it is very likely to find stored passwords, messages, domain names and IP address belonging to those processes.

PREV NEXT

OUTLINE

1.2.3.3 Incident Handling Process – System Back-Up

Data Acquisition

Note that, volatile data can also exist on disk.

One example is a memory page that was moved to a hard disk due to paging, temporary files, and even log files.

https://medium.com/@esmerycornielle/memory-management-paging-43b85abe6d2f

1.2.3.3 Incident Handling Process – System Back-Up

Data Acquisition

As we mentioned previously, there will be times when a system's OS cannot be entirely trusted. An example of such a case is a system containing a rootkit.

Dead acquisition is the acquisition type that should be employed when handling such an incident. Dead acquisition is usually performed with the help of the system's own hardware.

PREV NEXT

OUTLINE

1.2.3.3 Incident Handling Process – System Back-Up

Acquisition approaches:

There are two main approaches in which data acquisition could be performed, each with a different output.

1. The first way is **from disk drive to image file (imaging)**

2. The second one is **from disk drive to disk drive (cloning)**

PREV NEXT

1.2.3.3 Incident Handling Process – System Back-Up

Data Acquisition

We will focus our attention on the first acquisition approach:

1. **From disk drive to image file (imaging)**

PREV NEXT

OUTLINE

1.2.3.3 Incident Handling Process – System Back-Up

Data Acquisition

Disk drive to image (imaging), as the name suggests, mirrors the under investigation hard disk's content into an image file.

Imaging a drive creates what is called a "**forensic image**". The advantage of this method is scalability and efficiency.

OUTLINE

1.2.3.3 Incident Handling Process – System Back-Up

Data Acquisition – Write Blockers

As mentioned earlier, the risk of altering the original evidence while acquiring data is high. For this reason, many tools/software exist that can assist us in acquiring data in a safer manner.

Such tools are Write Blockers. Write Blockers ensure that data acquisition is performed without the risk of losing or altering data.

PREV NEXT

OUTLINE

1.2.3.3 Incident Handling Process – System Back-Up

Data Acquisition – Write Blockers

Write Blockers achieve this, as their name suggests, by blocking the hard disk from writing.

Write blockers could either be hardware-based or software-based.

OUTLINE

1.2.3.3 Incident Handling Process – System Back-Up

Data Acquisition – Write Blockers

1. WiebeTech® Forensic UltraDock from CRU Inc.

2. Tableau Forensic Imager TD3 from Guidance Software

OUTLINE

1.2.3.3 Incident Handling Process – System Back-Up

Data Acquisition

Like with any other thing in IT Security, integrity is of key importance.

There should be a way to validate the integrity of the acquired evidence.

PREV NEXT

OUTLINE

1.2.3.3 Incident Handling Process – System Back-Up

Data Acquisition – Evidence Integrity

Hash Functions are usually employed to validate the acquired evidence.

Hash functions are **One Way Cryptographic Functions** which map data of arbitrary size to a bit string of a fixed size (a hash); this hash can be seen as a fingerprint.

The slightest change to the source file will result in a totally different hash.

PREV NEXT

OUTLINE

1.2.3.3 Incident Handling Process – System Back-Up

Data Acquisition – Evidence Integrity

All calculated hash strings should be stored and communicated safely since they will be used to prove that the acquired data has not been altered.

PREV NEXT

OUTLINE

1.2.3.3 Incident Handling Process – System Back-Up

Data Acquisition – Evidence Integrity

Many hash functions being used nowadays. **For example:**

1. SHA-1,
2. SHA-2,
3. SHA-3, and
4. Finally, the old MD-5.

SHA-1 and MD-5 aren't secure since they suffer from collisions.

PREV NEXT

OUTLINE

1.2.3.4 Incident Handling Process – Long-term Containment

Before moving on to the Long-term Containment phase, communicating with your ISP may be needed, especially in cases of DDoS attacks, worms, or phishing campaigns; this is due to the fact that ISPs not only have greater visibility when it comes to attacks in the wild, but they also keep useful logs.

1.2.3.4 Incident Handling Process – Long-term Containment

Data Acquisition – Evidence Integrity

Now, it is time to decide the containment approach.

For example, if you are not still able to determine the attacker's actions or even motives, you may recommend leaving the machine intact and closely monitor the attacker's next moves.

1.2.3.4 Incident Handling Process – Long-term Containment

Remember, that the containment approach should first go through the respective business unit manager/representative.

Critical systems can't easily go down since they are related to core business processes or operations.

PREV NEXT

OUTLINE

1.2.3.4 Incident Handling Process – Long-term Containment

After the imaging and live acquisition activities are completed, we can start performing long-term containment activities.

If the business unit manager/representative agreed on taking the system down, we can go straight to the Eradication phase, where we eliminate every attacker-related actions and residuals.

If the affected system should stay as is, then it is time for long-term containment activities such as:

- Affected & related system patching
- (H)IDS insertion
- Password(s) and trust(s) changes
- Additional ingress/egress rules (router & firewall)
- Drop packets associated with a source or destination identified in the incident
- Eliminate attacker access etc.

Keep administrators and business unit managers/representatives in the loop.

PREV NEXT

OUTLINE

1.2.3.5 Incident Handling Process – Eradication

Long-term containment is effectively a band-aid, Eradication should take place in order to make sure the attacker is locked out of the affected machine and network.

During the Eradication procedure, we will have to identify the root cause and indicators of the incident. ← **Use information from the Detection & Analysis and Containment procedures.**

Another thing to do during Eradication is to isolate the intrusion and identify the attack vector.

Reformatting and reinstalling the OS or identifying a clean backup and reloading the data ONLY is a bad strategy. Drive-wiping is advised before doing anything else.

OUTLINE

1.2.3.5 Incident Handling Process – Eradication

Important phases of Eradication are eliminating attacker residuals, such as malware and improving defenses.

Eliminating attacker residuals includes:

- Removing malware such as backdoors, rootkits, malicious kernel-mode drivers, etc.
  - In case of a Rootkit, zero the drive out, reformat and rebuild the system for trusted install media.
- Thoroughly analyze logs to identify credential reuse through Remote Desktop, SSH, VNC, etc.

Improving defenses includes:

- Configuring additional router & firewall rules;
- Obscuring the affected system's position;
- Null routing; and,
- Establishing effective system hardening, patching, and vulnerability assessment procedures, etc.

Other systems in the network may suffer from the same vulnerability.

PREV NEXT

OUTLINE

1.2.3.6 Incident Handling Process – Recovery

After the Eradication procedure is completed, it is time for Recovery. During Recovery, we will bring the affected system(s) back to production.

Key points to consider are:

Process System Recovery

Restore of Operations

Monitoring

PREV NEXT

1.2.3.6 Incident Handling Process – Recovery

Process System Recovery

Restore of Operations

Monitoring

Once the affected system is restored, ask the business unit to perform QA activities to ensure the system's running condition.

Also, ask the business unit to ensure the system includes everything needed for their operations.

OUTLINE

1.2.3.6 Incident Handling Process – Recovery

Process System Recovery

Restore of Operations

Monitoring

A decision has to be made regarding when the restored system will enter production again.

Consult/coordinate with the business unit for this matter.

OUTLINE

1.2.3.6 Incident Handling Process – Recovery

Process System Recovery

Restore of Operations

Monitoring

Once the restored system is back to production:

- Keep a close eye for oversights. Stealthy backdoors may still exist.
- Network, as well as host-based intrusion systems, should be utilized, looking for signs/patterns/signatures related to the original attack.
- Thoroughly analyze critical logs and events for signs of re-infection or re-compromise.

OUTLINE

1.2.3.6 Incident Handling Process – Recovery

Process System Recovery

Restore of Operations

Monitoring

What to look for during the weeks (or even months) to come:

- Changes to registry keys and values. `reg \\[MachineName]`
- Abnormal processes via wmic. `wmic /node:[MachineName] /user:[Admin] /password:[password]` or `ps` for Linux
- Abnormal user accounts. `wmic useraccount list brief` or `net user` commands or `cat /etc/passwd` for Linux

https://www.commandlinefu.com/commands/browse contains a lot of commands/one-liners to acquire various system information.

OUTLINE

1.2.3 Incident Handling Process – Containment, Eradication & Recovery

Containment, Eradication & Recovery Additional Information

In the previous slides, we summarized the most important aspects of the Containment, Eradication & Recovery phase.

For more information, please refer to Computer Security Incident Handling Guide by NIST.

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

PREV NEXT

OUTLINE

1.2.4 Incident Handling Process – Post-Incident Activity

PREV NEXT

OUTLINE

1.2.4 Incident Handling Process – Post-Incident Activity

Finally, we have reached the Post-incident Activity phase, which is when we take a deep breath and report the identified weaknesses, oversights, blind spots, etc., regarding both our processes and technological measures.

OUTLINE

1.2.4 Incident Handling Process – Post-Incident Activity

It is a known fact that attackers keep getting more and more sophisticated.

Incident handling was never (and will never be) trivial; this is exactly why the Post-Incident Activity phase is important.

1.2.4 Incident Handling Process – Post-Incident Activity

Right after recovery, the respective incident handling team should start constructing an objective, accurate and thorough report regarding the lessons learned from handling the incident.

PREV NEXT

OUTLINE

1.2.4 Incident Handling Process – Post-Incident Activity

This is not to say that the report should contain only the identified weaknesses, oversights, and blind spots. Working processes and successful detection methods should also be included.

Don't be afraid to mention how effective you were against specific stages of the attack.

PREV NEXT

OUTLINE

1.2.4 Incident Handling Process – Post-Incident Activity

Schedule a meeting when things cool down to discuss this report with all involved parties, such as system administrators, affected business unit representatives, IT security team, etc.

Focus your energy on improving your processes, technological measures and visibility.

PREV NEXT

OUTLINE

1.2.4 Incident Handling Process – Post-Incident Activity

Post-Incident Activity Additional Information

In the previous slides, we summarized the most important aspects of the Post-Incident Activity phase.

For more information, please refer to Computer Security Incident Handling Guide by NIST.

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

PREV NEXT

OUTLINE

1.2 Incident Handling Process

Of course, part of your incident handling activities will be analyzing captured or live traffic as well as network flows.

We will focus on that aspect in the next section.

PREV NEXT

OUTLINE

1.3

The Course's Scope

PREV NEXT

OUTLINE

1.3 The Course's Scope

During the course, we will apply the incident handling process, we just covered, against each phase of the cyber kill chain (among other things).

More specifically, a *Practical Incident Handling* section exists, that includes the below modules:

- **Preparing & Defending Against Reconnaissance**
- **Preparing & Defending Against Scanning and Information Gathering**
- **Preparing & Defending Against Exploitation**
- **Preparing & Defending Against Post-exploitation**

1.4

Incident Handling Forms

PREV NEXT

OUTLINE

1.4 Incident Handling Forms

There are Incident Handling Forms, which will come in handy during incident handling. Let's look at some important forms you should preprint and use.

Incident Contact List

Incident Detection

Incident Casualties

Incident Containment

Incident Eradication

PREV NEXT

1.4 Incident Handling Forms

Incident Contact List | Incident Detection | Incident Casualties | Incident Containment | Incident Eradication

This form should contain the contact details of the organization's:

- CISO / CIO
- SPOC of the incident handling or CSIRT team
- Legal department contact
- Public relations contact
- ISP SPOC
- Local cybercrime unit etc.

1.4 Incident Handling Forms

IHRP

This form should contain information such as:

- The first person who detected the incident
- The incident's summary (type of incident, incident location, incident detection details, etc.)

1.4 Incident Handling Forms

Incident Contact List | Incident Detection | **Incident Casualties** | Incident Containment | Incident Eradication

IHRP

This form should contain information such as:

- Location of affected systems
- Date and time incident handlers arrived
- Affected system details (one form per affected system is advised)
  - Hardware vendor
  - Serial number
  - Network connectivity details
    - Host Name | IP Address | MAC Address

1.4 Incident Handling Forms

IHRP

This form should contain information such as:

- Isolation activities per affected system
  - Was the affected system isolated?
    - Date and time the system was isolated
    - Way of system's isolation
- Back-up activities per affected system
  - Handler who performed the restoration
  - Back-up details etc.

1.4 Incident Handling Forms

IHRP

This form should contain information such as: (one form per affected system is advised)

- Handler(s) performing investigation on the system
- Was the incident's root cause discovered?
  - Incident root cause analysis
- Actions taken to ensure the incident's root cause was remediated and the possibility of a new incident eliminated

1.4 Incident Handling Forms

NOTE

Always archive any incident-related communications, on a per-incident basis.

Any incident's communication flow and content should be readily available.

PREV NEXT

OUTLINE

1.5

Appendix

PREV NEXT

OUTLINE

Appendix

Windows Cheat Sheet

- **User Accounts**
  - Identify curious-looking accounts in the Administrators group [use *lusrmgr.msc* for GUI access]
  - Related Command: `net user`
  - Related Command: `net localgroup administrators`
- **Processes** (focus on those running with high privileges)
  - Identify abnormal processes [use *taskmgr.exe* for GUI access]
  - Related Command: `tasklist`
  - Related Command: `wmic process list full`
  - Related Command: `wmic process get name,parentprocessid,processid`
  - Related Command: `wmic process where processid=[pid] get commandline`
- **Services**
  - Identify abnormal services [use *services.msc* for GUI access]
  - Related Command: `net start`
  - Related Command: `sc query | more`
  - Related Command (associate running services with processes): `tasklist /svc`

PREV NEXT

OUTLINE

Appendix

Windows Cheat Sheet

- **Scheduled Tasks** **(focus on those running with high privileges or look suspicious)**
  - Identify curious-looking scheduled tasks [you can go to Start -> Programs -> Accessories -> System Tools -> Scheduled Tasks for GUI access to scheduled tasks]
  - Related Command: `schtasks`
- **Extra Startup Items**
  - Identify users' autostart folders
  - Related Command: `dir /s /b "C:\Documents and Settings\[username]\Start Menu\"`
  - Related Command: `dir /s /b "C:\Users\[username]\Start Menu\"`
- **Auto-start Reg Key Entries**
  - Check the below registry keys for malicious autorun configurations [use regedit for GUI access and inspect both HKLM and HKCU] ← You can also scrutinize every auto-start location through the Autoruns MS tool
  - *HKLM\Software\Microsoft\Windows\CurrentVersion\Run*
  - *HKLM\Software\Microsoft\Windows\CurrentVersion\Runonce*
  - *HKLM\Software\Microsoft\Windows\CurrentVersion\RunonceEx*
  - Related Command: `reg query [reg key]`

https://docs.microsoft.com/en-us/sysinternals/downloads/autoruns

OUTLINE

Appendix

Windows Cheat Sheet

- **Listening and active TCP and UDP ports**
  - Identify abnormal listening and active TCP and UDP ports
  - Related Command: `netstat -nao 10`
- **File Shares**
  - All available file shares of a machine should be justified
  - Related Command: `net view \\127.0.0.1`
- **Files**
  - Identify major decreases in free space [you can use the file explorer's search box and enter *"size:>5M"*
- **Firewall Settings**
  - Examining current firewall settings to detect abnormalities from a baseline
  - Related Command (XP/2003): `netsh firewall show config`
  - Related Command (Vista-Win8 +): `netsh advfirewall show currentprofile`

PREV NEXT

OUTLINE

Appendix

Windows Cheat Sheet

- **Systems connected to the machine**
  - Identify NetBIOS over TCP/IP activity
  - Related Command: `nbtstat -S`
- **Open Sessions**
  - Knowing who has an open session with a machine is of great importance
  - Related Command: `net session`
- **Sessions with other systems (NetBIOS/SMB)**
  - Identify sessions the machine has opened with other systems
  - Related Command: `net use`
- **Log Entries**
  - Identify curious-looking events [you can use eventvwr.msc for GUI access to logs]
  - Related Command: `wevtutil qe security`

Appendix

Windows Cheat Sheet

Useful investigation software

- Process Explorer
- Process Monitor

https://docs.microsoft.com/en-us/sysinternals/downloads/process-explorer
https://docs.microsoft.com/en-us/sysinternals/downloads/procmon

PREV NEXT

OUTLINE

Appendix

Linux Cheat Sheet

- **User Accounts**
  - Identify curious-looking accounts in */etc/passwd*
  - Related Command: `passwd -S [User_Name]`
  - Related Command: `grep :0: /etc/passwd`
  - Related Command: `find / -nouser -print`
- **Log Entries**
  - Identify curious-looking events such as:
    - Large number of authentication or login failures (telnetd, sshd etc.)
    - Overly long and strange-looking strings being passed as input (buffer overflow attempt)
- **Resources**
  - Identify deviation from normal resource utilization
  - Related Command (system CPU load, "load average"): `uptime` ← Compare this to a baseline
  - Related Command (memory utilization): `free` ← Compare this to a baseline

Display UID 0 and GID 0 accounts

Identify the existence of attacker-created temp users.

Appendix

Linux Cheat Sheet

- **Running Processes** (focus on those running with root privileges)
  - Identify abnormal processes that could indicate malicious activity
  - Related Command: `ps aux`
  - Related Command (more details): `lsof -p [pid]`
- **Services**
  - Identify abnormal services
  - Related Command: `service --status-all` ← RedHat and Mandrake use `chkconfig -list` instead
- **Scheduled Tasks** (focus on cron jobs configured by root or any UID 0 account)
  - Identify curious-looking scheduled tasks
  - Related Command: `crontab -l -u [account]`
  - Related Command: `cat /etc/crontab`
  - Related Command: `cat /etc/cron.*`

PREV NEXT

OUTLINE

Appendix

Linux Cheat Sheet

- **Listening and active TCP and UDP ports**
  - Identify abnormal listening and active TCP and UDP ports
  - Related Command: `lsof -i` ← Compare this to a baseline
  - Related Command: `netstat -nap` ← Compare this to a baseline
- **ARP**
  - Identify abnormal IP – MAC mappings
  - Related Command: `arp -a` ← Compare this to a baseline
- **Files**
  - Identify curious-looking files
  - Related Command (abnormal SUID root files): `find / -uid 0 -perm -4000 -print` Related Command (overly large files): `find /home/ -type f -size +512k -exec ls -lh {} \;`

Appendix

Linux Cheat Sheet

Useful investigation software

- Chkrookit
- Tripwire / AIDE

http://www.chkrootkit.org/
https://github.com/Tripwire/tripwire-open-source
https://sourceforge.net/projects/aide/

PREV NEXT

Labs

Enterprise-wide Incident Response: Part 1 - GRR

In this lab, you will learn how to utilize the GRR Incident Response framework in order to perform quicker and more efficient IR activities.

During the lab, you will have the opportunity to detect (fileless) malware, various stealthy persistence techniques and privilege escalation attempts on a heterogeneous and enterprise-like network.

Enterprise-wide Incident Response: Part 2 - Velociraptor

In this lab, you will learn how to utilize the Velociraptor Incident Response framework in order to perform quicker and more efficient IR activities.

During the lab, you will have the opportunity to detect fileless malware, as well as leverage specific Velociraptor capabilities to proactively monitor endpoints on a heterogeneous and enterprise-like network.

PREV NEXT

1.6

References

PREV NEXT

OUTLINE

References

Computer Security Incident Handling Guide by NIST

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

Cyber kill chain

https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

Wireshark

https://www.wireshark.org/

Kernel-level rootkit

http://www.dmi.unipg.it/bista/didattica/sicurezza-pg/seminari2008-09/seminario_neri/seminario_neri.pdf

References

IANA

http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

Speed Guide – Port Database

https://www.speedguide.net/ports.php

Azure Logging & Auditing

https://docs.microsoft.com/en-us/azure/security/azure-log-audit

Apache HTTP Server Documentation Version 2.4

http://httpd.apache.org/docs/current/logs.html

References

CSIRT Case Classification (Example for Enterprise CSIRT)

https://www.first.org/resources/guides/csirt_case_classification.html

Request Tracker for Incident Response

https://bestpractical.com/rtir/

Incident Response Tools & Resources

https://github.com/meirwah/awesome-incident-response#incident-management

Canarytokens

http://canarytokens.org/generate

PREV NEXT

References

Paging

https://medium.com/@esmerycornielle/memory-management-paging-43b85abe6d2f

Commandlinefu

https://www.commandlinefu.com/commands/browse

Autoruns

https://docs.microsoft.com/en-us/sysinternals/downloads/autoruns

Process Explorer

https://docs.microsoft.com/en-us/sysinternals/downloads/process-explorer

References

Process Monitor

https://docs.microsoft.com/en-us/sysinternals/downloads/procmon

Chkrookit

http://www.chkrootkit.org/

Tripwire

https://github.com/Tripwire/tripwire-open-source

AIDE

https://sourceforge.net/projects/aide/

PREV NEXT

OUTLINE

Labs

Enterprise-wide Incident Response: Part 1 - GRR

In this lab, you will learn how to utilize the GRR Incident Response framework in order to perform quicker and more efficient IR activities.

During the lab, you will have the opportunity to detect (fileless) malware, various stealthy persistence techniques and privilege escalation attempts on a heterogeneous and enterprise-like network.

Enterprise-wide Incident Response: Part 2 - Velociraptor

In this lab, you will learn how to utilize the Velociraptor Incident Response framework in order to perform quicker and more efficient IR activities.

During the lab, you will have the opportunity to detect fileless malware, as well as leverage specific Velociraptor capabilities to proactively monitor endpoints on a heterogeneous and enterprise-like network.