599.5
Action on Objectives, Threat Hunting, & Incident Response

SANS

Action on Objectives, Threat Hunting, & Incident Response

SANS

This page intentionally left blank.

Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- Day 4: Lateral Movement
- **Day 5: Action on Objectives, Threat Hunting & Incident Response**
- Day 6: APT Defender Capstone

SEC599.5

Domain dominance
  - Dominating the AD - Basic strategies
  - Golden Ticket, Skeleton Key, DCSync and DCShadow
  - Detecting domain dominance
  - Exercise: Domain dominance

Data exfiltration
  - Common exfiltration strategies
  - Exercise: Detecting data exfiltration

Leveraging threat intelligence
  - Defining threat intelligence
  - Exercise: Leveraging threat intelligence with MISP & Loki

Threat Hunting & incident response
  - Proactive threat hunting strategies
  - Exercise: Hunting your environment using OSQuery
  - incident response process
  - Exercise: Finding malware using Volatility & YarGen

This page intentionally left blank.

Throughout the course, we've discussed a variety of techniques that are used by adversaries to move laterally through a Windows environment and obtain administrative access. Remember that our adversaries are persistent and would like to persist the access they achieve!

We will now discuss a number of techniques that are used to consolidate / persist administrative access to AD:

- Dump the NDTS.dit file.
- Creation of a Domain Administrator (DA) account.
- Creation of a Golden Ticket.
- Creation of a Skeleton Key.
- Use of DCSync or DCShadow.

What's Next? Domain Dominance (AD Persistence)
Throughout the course, we've discussed a variety of techniques that are used by adversaries to move laterally through a Windows environment and obtain administrative access. Remember that our adversaries are persistent and would like to persist the access they achieve!

So, what is the next step in the adversary's toolbox? Typically, an adversary will attempt one of the following tricks to consolidate / persist administrative access to AD:

- Dumping the NTDS.dit file, from which all credentials can be extracted.
- Simply creating a Domain Administrator (DA) account (preferably with strong credentials).
- Creating a Kerberos Golden Ticket that will allow long-term access to the environment.
- Using the DCSync attack.
- Creating a Skeleton Key, which will allow the adversary to authenticate as any user in the environment.

Let's discuss these techniques in some more detail!

Domain Dominance – Obtaining Access to Back-Up NTDS.dit File (1)

As discussed before, credentials in the Active Directory are stored in the ntds.dit file (encrypted with system key). This file is only accessible with elevated privileges. Furthermore, it is **locked by the OS** while the Domain Controller is running.

Different techniques could be used to obtain access to the file:

- Should the adversary have already obtained administrative access to the domain, he could use tools such as the Volume Shadow Copy Service to create a read-only copy and steal the file.
- Badly secured backups of the Domain Controller drives (e.g. open network shares) could allow an adversary to extract the file without administrative privileges.
- Specialized, open-source tools can be used to extract hashes from the ntds.dit file. These hashes can be used in subsequent Pass-the-Hash attacks or even be cracked using offline password crackers.

Domain Dominance – Obtaining Access to Back-Up NTDS.dit File (1)
Advanced adversaries will often attack the Active Directory infrastructure of their targets because they contain the "keys to the kingdom." Once they are inside, they will focus on Active Directory: If they obtain domain admin rights, they will have unlimited access to the resources of that domain. Once inside, they will also try to achieve persistence: Obtain unconditional access to the administrative functions of Active Directory, even when credentials are changed.

There are many attacks possible with Active Directory. Here, we want to focus on attacks to obtain credentials and/or access. As we saw, the Active Directory database is stored inside a file called ntds.dit. And the hashes are protected by an encryption key stored in the system registry.
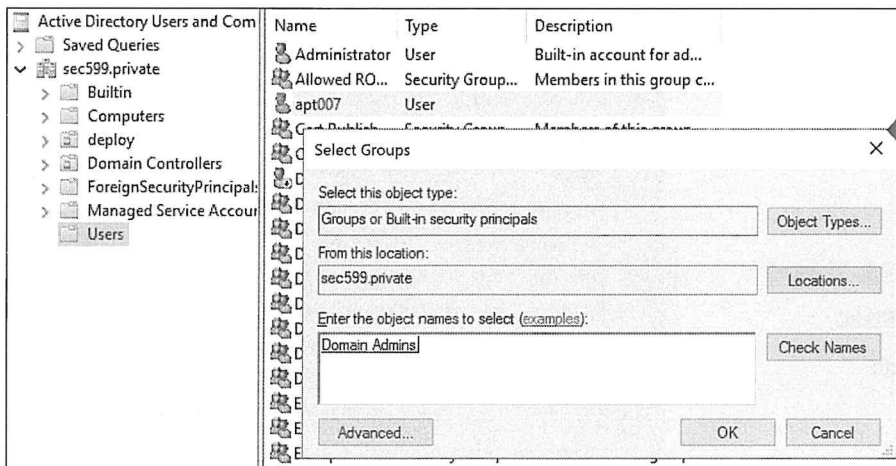
When an adversary obtains a copy of these files, he can extract hashes and recover passwords, among other things.

These files are present on a Domain Controller, but when the Domain Controller is up and running, these files are in use and cannot be copied. There is a workaround, however; shadow copies. Shadow copies (aka VSS, Volume Snapshot Service) is a Microsoft technology to create local backups of files. This technology can be used to recover backup copies of the ntds.dit and SYSTEM files.

Adversaries don't always need access to a Domain Controller to obtain these files. When centralized backups are made of the Domain Controllers, these files will be found on the backup servers, too. It is important to adequately protect these files, even in backups.

Once an adversary has obtained a copy of the ntds.dit and SYSTEM files, he can proceed to the extraction of hashes and recovery of passwords. There are several open-source tools that can read these files and decrypt hashes: Examples are ntdsxtract and secretsdump. For large databases (10,000 and more users), ntdsxtract tends to be slow (can take several days), but secretsdump is much faster (a couple of hours). Both can output the NTLM hashes and LM hashes (when present) in different formats.

Active Directory Users and Com
> Saved Queries
> sec599.private
  > Builtin
  > Computers
  > deploy
  > Domain Controllers
  > ForeignSecurityPrincipal
  > Managed Service Accoun
  > Users

Name	Type	Description
Administrator	User	Built-in account for ad...
Allowed RO...	Security Group...	Members in this group c...
apt007	User	

Select Groups ×

Select this object type:

Groups or Built-in security principals   [Object Types...]

From this location:

sec599.private   [Locations...]

Enter the object names to select (examples):

Domain Admins   [Check Names]

[Advanced...]   [OK]   [Cancel]

New Domain Admin
Simple yet highly effective, adversaries could opt to create a new Domain Administrator account on the environment, often with a password that does not expire!

This is highly visible, any additions to administrative groups in Active Directory should be reported and analyzed by your monitoring teams!

Domain Dominance – Creating a Domain Admin Account
A simple attack to achieve persistence in Active Directory is to create a new domain admin user with a password that never expires.

Once an adversary has administrative rights to the domain, he can create a new Domain Administrator. This does not necessarily imply that the adversary must obtain Domain Administrator rights, but just obtain the right to create new users and assign them to groups. A Domain Administrator has these rights, but in Active Directory, these rights can also be delegated to administrative users that don't have domain admin rights. Because the Domain Administrator is such a powerful account, many organizations will only provide this account to a select group of security staff members.

Other users that need to perform common administrative tasks, like managing users, are only given the necessary rights to do this. But once a user has the right to create a new user and assign it to arbitrary groups, he can create a Domain Administrator.

This created account will give the adversary domain admin access to the domain as long that the account is not discovered and removed. It is therefore important to monitor your Active Directory infrastructure for the creation of new accounts with administrative rights.

Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- Day 4: Lateral Movement
- **Day 5: Action on Objectives, Threat Hunting & Incident Response**
- Day 6: APT Defender Capstone

SEC599.5

Domain dominance
- Dominating the AD - Basic strategies
- Golden Ticket, Skeleton Key, DCSync and DCShadow
- Detecting domain dominance
- Exercise: Domain dominance

Data exfiltration
- Common exfiltration strategies
- Exercise: Detecting data exfiltration

Leveraging threat intelligence
- Defining threat intelligence
- Exercise: Leveraging threat intelligence with MISP & Loki

Threat Hunting & incident response
- Proactive threat hunting strategies
- Exercise: Hunting your environment using OSQuery
- incident response process
- Exercise: Finding malware using Volatility & YarGen

This page intentionally left blank.

We discussed Kerberos attacks in quite some depth yesterday! One of the possible attack vectors we looked at was the creation of a "Golden Ticket"! A Golden Ticket is "nothing more" than a "special" TGT created by an adversary.

Golden Ticket (encrypted using KDC LT key)

Start / End / MaxRenew: 05/12/2018 07:12:18 ; 05/12/2028 17:12:18 ; 12/12/2028 07:12:18 ;
Service Name: krbtgt; synctechlab
Target Name: krbtgt; synctechlabs
Client Name: domain.admin; sync
Flags: 40e10000
Session Key: 0x00000012eb212eb 45eb4124af9010bf13f...<snip>

Privilege Account Certificate (PAC)
Username: DOMAIN.ADMIN
SID: S-1-5-21-409 ... <snip>
Groups: Domain Admins ... <snip>

Signed using Target LT Key
Signed using KDC LT Key

Golden Ticket – Prerequisites

In order to create a valid TGT (with a valid PAC), an adversary requires:

• The Target LT Key
• The KDC LT Key

In case of a TGT, these keys are identical (krbtgt). Adversaries would thus have to obtain the NTLM hash of the krbtgt account (RC4) or the AES key (AES)!

Domain Dominance – Introducing the Golden Ticket

Another method to obtain a ticket for pass-the-ticket attacks is to use Mimikatz to generate a Golden Ticket. A Golden Ticket is a ticket-granting-ticket providing maximum access for a maximum period of time. It's the mother of all tickets; there is no ticket that provides more access.

We discussed Kerberos attacks in quite some depth yesterday! One of the possible attack vectors we looked at was the creation of a "Golden Ticket"! A Golden Ticket is "nothing more" than a "special" TGT created by an adversary.

In order to create a valid TGT (with a valid PAC), an adversary requires:

- The Target LT Key.
- The KDC LT Key.

In case of a TGT, these keys are identical (krbtgt)). Adversaries would thus have to obtain the NTLM hash of the krbtgt account (RC4) or the AES key (AES)!
When Active Directory is compromised, the NTLM hash or AES key of the krbtgt account can be extracted from memory (see our sekurlsa::lsa example). All the other information needed to create a Golden Ticket is not secret information but can be obtained readily. For example, the name of the domain will be needed. Although the domain name of your organization is not something you publicize, it will not be difficult for an adversary to obtain the name of your domain.

Note that the NTLM hash or AES key of the krbtgt account can also be extracted from the Active Directory database and its backups. So, it is essential to protect this data.

The name Golden Ticket refers to the Willi Wonka movie. In this movie, children can win a Golden Ticket that provides them full access to a fabulous chocolate factory.

Domain Dominance – Kerberos Flow with Golden Ticket

When a Golden Ticket is used by an adversary, the first interaction is a TGS-REQ (request for a Service Ticket) using the forged TGT (the Golden Ticket). There is no prior credential submission or AS-REQ / AS-REP!

Golden Ticket (TGT)

Start / End / MaxRenew: 05/12/2018 07:12:18 ; 05/12/2028 17:12:18 ; 12/12/2028 07:12:18 ;
Service Name: krbtgt; synctechlab
Target Name: krbtgt; synctechlabs
Client Name: domain.admin; sync
Flags: 40e10000
Session Key: 0x00000012eb212eb 45eb4124af9010bf13f...<snip>

Privilege Account Certificate (PAC)
Username: DOMAIN.ADMIN
SID: S-1-5-21-409 ... <snip>
Groups: Domain Admins ... <snip>

Signed using Target LT Key
Signed using KDC LT Key

3. TGS-REQ
4. TGS-REP
5. Usage

KDC
File Server

Illustration inspired by "Abusing Microsoft Kerberos - Sorry you guys don't get it." Benjamin Delpy (Blackhat USA 2014)

Domain Dominance – Kerberos Flow with Golden Ticket

Let's have a quick look at the Kerberos flow when a Golden Ticket is in play. When a Golden Ticket is used by an adversary, the first interaction is a TGS-REQ (request for a Service Ticket) using the forged TGT (the Golden Ticket). There is no prior credential submission or AS-REQ / AS-REP!

One might assume that the lack of AS-REQ / AS-REP can be a trigger for detection. Unfortunately, Kerberos is a stateless protocol, hence the AD does not keep track of prior AS-REQ / AS-REP sequences before a TGS-REQ is performed. If one were to implement a rule like this, an avalanche of false positives would appear (e.g. TGTs generated by DC01 and subsequently used to request a Service Ticket on DC02).

We discussed Kerberos attacks in quite some depth yesterday! One of the possible attack vectors we looked at was the creation of a "Golden Ticket"! A Golden Ticket is "nothing more" than a "special" TGT created by an adversary.

So, what properties does a Golden Ticket have?

- It's created by an adversary *WITHOUT* any interaction with the DC (it's "homemade ☺"). This is possible because Kerberos is a "stateless" protocol.
- As discussed in the previous slide, though, it would require an adversary to obtain the KDC. Long Term Key (which should not be easy to get!)
- It's typically a TGT for an administrative account (e.g. RID 500 in the domain or a Domain Administrator).
- It's typically valid for a long time (10 years by default).

Domain Dominance – Golden Ticket Properties
So what properties does a Golden Ticket have? What makes it so special? A Golden Ticket is, as previously discussed, nothing more than a forged TGT. However, it does have some very interesting features:

- It's created by an adversary *WITHOUT* any interaction with the DC (it's "homemade"). This is possible because Kerberos is a "stateless" protocol.
- As discussed in the previous slide, though, it would require an adversary to obtain the KDC Long Term Key (which should not be easy to get!)
- It's typically a TGT for an administrative account (e.g. RID 500 in the domain or a Domain Administrator).
- It's typically valid for a long time (10 years by default).

Golden Tickets were initially described in Benjamin Delpy's BlackHat USA 2014 presentation "Abusing Microsoft Kerberos - Sorry you guys don't get it." This presentation is often referred to as "the Golden Ticket talk." You can find the slides here:

https://www.blackhat.com/docs/us-14/materials/us-14-Duckwall-Abusing-Microsoft-Kerberos-Sorry-You-Guys-Don't-Get-It.pdf

Domain Dominance – Creating a Golden Ticket with Mimikatz – Step I

Golden Ticket – Step I
Using Mimikatz, a Golden Ticket can be generated using the following information:

- KDC LT key (e.g. KRBTGT NTLM hash)
- Domain admin account name
- Domain name
- SID of domain admin account

All of these values can be obtained by any user in the domain, except for the KDC LT key!

Domain Dominance – Creating a Golden Ticket with Mimikatz – Step 1
In this example, we explain what is needed to create a Golden Ticket with Mimikatz. Mimikatz's command Kerberos::golden can be used to generate a Golden Ticket. This is a pure "computational" command: It does not require administrative rights, and it does not require access to the domain. Generating a Golden Key can be done on any Windows computer in the world with normal access rights, provided that the necessary input values are known.

- The first input, the most difficult value to obtain, is the NTLM hash of the Kerberos account (krbtgt). As stated before, this hash can be obtained from compromised Domain Controllers or Active Directory databases.
- The second value is the name of the administrative account. In our example, that is root.
- The third value is the domain name: sec599.private in our example.
- And the fourth and last value is the SID of the admin account.

That is the only input that is needed to create a Golden Ticket. Remark that Mimikatz has many more options for this kerberos::golden command but discussing these is not in scope (this is not an offensive course!). The only extra option we used here is the /ticket: option to write the Golden Ticket to disk.

From the Lifetime line on the screen, you can see that this Golden Ticket is valid for 10 years! This means that as long that you do not change the password of the krbtgt account, the ticket will grant access to your complete Active Directory infrastructure for the next 10 years!

From the output, you can clearly see that the administrative account (user ID 500) is a domain admin (group ID 512) and an enterprise admin (group ID 519).

```
mimikatz 2.1.1 x86 (oe.eo)                                    _  □  X

mimikatz # kerberos::ptt golden.ticket.bin

* File: 'golden.ticket.bin': OK

mimikatz # kerberos::list

[00000000] - 0x00000017 - rc4_hmac_nt
   Start/End/MaxRenew: 10/07/2017 18:08:29 ; 8/07/2027 18:08:29 ; 8/07/2027 18:0
8:29
   Server Name      : krbtgt/sec599.private @ sec599.private
   Client Name      : root @ sec599.private
   Flags 40e00000   : pre_authent ; initial ; renewable ; forwardable ;

mimikatz #
```

Golden Ticket – Step 2
In this second attack step, the adversary can now re-inject the ticket in Windows memory, thereby readying it for use when we try to attempt accessing a service that relies on Kerberos authentication (e.g. accessing a Windows share).

Once a Golden Ticket is generated, the only way to mitigate the attack is to change the password of the krbtgt account twice (It has a hard-coded password history of 2 + the KDC will also attempt to validate a TGT with hashes in the password history!). This will, however, invalidate all tickets and could have production impact!

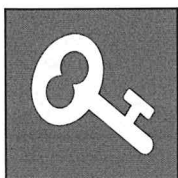**Domain Dominance – Creating a Golden Ticket with Mimikatz – Step 2**
To use the previously generated ticket, we issue the Mimikatz command Kerberos::ptt golden.ticket.bin. This injects the ticket into Windows' memory, ready to be used when we attempt to connect to a service.

The Kerberos::list command can be used to list all the tickets that we have. We can see that the servername is krbtgt/sec599.private: This tells us that this is a ticket-granting ticket for the sec599.private domain. The user is root of the sec599.private domain.

If we look at the start and end dates, we see 2017 – 2027. That's an extremely long period for a ticket: 10 years (remember, the default lifetime of a TGT is 10 hours). That's because here, we used a Golden Ticket. Now when we try to connect to a share, for example, this injected ticket will be used to obtain a ticket to the share we want to access (this requires Kerberos, thus we need to use the fileserver name to access, and not its IP address, as this would result in NTLM authentication which does not work with tickets).

Once a Golden Ticket is generated, the only way to mitigate the attack is to change the password of the krbtgt account twice (It has a hard-coded password history of 2 + the KDC will also attempt to validate a TGT with hashes in the password history!). This will, however, invalidate all TGTs immediately and could thus have production impact!

Another AD persistence attack we would like to highlight is the "Skeleton Key" attack, which is also been added as a built-in module in Mimikatz. A Skeleton Key is a key that opens all the locks in a building. In the same way a Skeleton Key can "unlock" all systems in the domain!

How does the "Skeleton Key" attack work?

- The "Skeleton Key" essentially means that a backdoor is implanted on the Domain Controller.
- This backdoor runs in memory and adapts the running Domain Controller in such a way that a single password (the skeleton password) can be used to log on with any account.
- As it runs in memory, it does not persist by itself (but can, of course, be scripted or persisted using one of the persistence strategies we've seen above).
- This only affects one Domain Controller, so the adversary would have to target all Domain Controllers for maximum effect...

Domain Dominance – Introducing the Skeleton Key
Another AD persistence attack we want to illustrate is an attack to achieve persistence inside a domain. Of course, the Golden Ticket is the ultimate persistence mechanism, but this Skeleton Key attack also merits our attention.

A Skeleton Key is a special key that opens all the locks inside a building. Each door lock inside the building requires its own key to be opened, but all the locks can also be opened with a special, single key: The Skeleton Key. Mimikatz provides a Skeleton Key attack for Active Directory.

When the Mimikatz skeleton attack is executed on a Domain Controller, a bit of code is patched in memory so that all accounts can be logged in with a special password ("mimikatz"). This does not remove the existing passwords. It is now possible to log into an account using 2 different passwords: One can log in with the normal account password, or one can login with the Skeleton Key password.

By patching the code of the LSA process, the functions that validate credentials are modified: They still accept the normal password, but also accept the skeleton password.

This has to be done on a Domain Controller; it does not work on non-Domain Controllers. But this does not mean that an attacker can only log in with the skeleton password on a Domain Controller: The password works on all domain members that use this compromised Domain Controller for authentication.

If there is more than one Domain Controller, the attack needs to be executed on all Domain Controllers to be sure that the skeleton password will work in all cases.

```
  mimikatz 2.1.1 x64 (oe.eo)


  .#####.    mimikatz 2.1.1 (x64) built on Jun 18 2017 18:46:28
 .## ^ ##.   "A La Vie, A L'Amour"
 ## / \ ##   /* * *
 ## \ / ##    Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'    http://blog.gentilkiwi.com/mimikatz           (oe.eo)
  '#####'                                      with 21 modules * * */

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # misc::skeleton
[KDC] data
[KDC] struct
[KDC] keys patch OK
[RC4] functions
[RC4] init patch OK
[RC4] decrypt patch OK

mimikatz # _
```

Skeleton Key in Action
In the screenshot on the left, we can observe Mimikatz installing a "Skeleton Key" backdoor on the Domain Controller.

Note the simplicity of the commands... This will now allow anyone to authenticate as any user in the domain with the Skeleton Key password ("mimikatz").

Might be a good idea to write a script to check successful authentication using the password "mimikatz", right? ☺

Domain Dominance – Skeleton Key in Action
In the example above, we can see the Skeleton Key attack. Since this will patch the code of the LSA process in memory, administrative rights are required, and the debug privilege needs to be enabled. An attacker just needs to run the command "misc::skeleton" and voila, the Skeleton Key is installed on the Domain Controller.

When there are several Domain Controllers inside a domain, domain members connect to a Domain Controller for authentication via a kind of load-balancing scheme.

This means that to be 100% reliable, the Skeleton Key attack needs to be performed on all Domain Controllers. Otherwise, a domain member might authenticate to a Domain Controller that does not have the Skeleton Key patch applied in memory.

Since this is a patch in memory, simply rebooting the Domain Controller removes the Skeleton Key. But, of course, attackers can install an autorun entry for Mimikatz to run automatically when the Domain Controller boots.
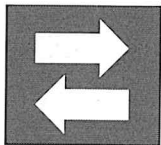
Benjamin Delpy, the author of Mimikatz, has pioneered many attacks on Windows security, and this has led to security improvements in Windows. In collaboration with Vincent Le Toux, Benjamin worked out another attack on Active Directory: Impersonating a Domain Controller.

How does the "DCSync" attack work?

- For availability reasons, many ADs have multiple Domain Controllers. Each Domain Controller has a copy of the AD database, and updates to this database on a Domain Controller need to be propagated to the other Domain Controllers in due time. This is called Active Directory replication.

- Mimikatz has a command (DCSync) that will make any computer that runs Mimikatz impersonate a Domain Controller to a target Domain Controller to obtain the credentials stored in this Domain Controller (provided administrative credentials are available).

DCSync essentially has the same impact as copying the ntds.dit database file! Once an attacker successfully launches an attack like this, all passwords in the domain are compromised and everything is to be changed!

Domain Dominance – Replicating the Domain – DCSync

Benjamin Delpy, the author of Mimikatz, has pioneered many attacks on Windows security, and this has led to security improvements in Windows. In collaboration with Vincent Le Toux, Benjamin worked out another attack on Active Directory: Impersonating a Domain Controller.

For availability reasons, administrators deploy more than one Domain Controller in an Active Directory infrastructure. Each Domain Controller has a copy of the Active Directory database, and updates to this database on a Domain Controller (for example the creation of a new user) need to be propagated to the other Domain Controllers in due time. This is called Active Directory replication: A set of methods and protocols to synchronize the database of Active Directory Domain Controllers.

Vincent and Benjamin have worked out these methods and protocols for use by Mimikatz: Mimikatz has a command (DCSync) that will make any computer that runs Mimikatz impersonate a Domain Controller to a target Domain Controller to obtain the credentials stored in this Domain Controller.

Of course, normal users cannot access this information. One needs domain admin rights to be able to participate in data replication. A Golden Ticket can provide these admin rights.

DCSync can dump the hashes of all users, or of a selected user.

Domain Dominance – Replicating the Domain – DCSync – Example

This example shows the DCSync command. By issuing "kerberos::dcsync /user:administrator" command, we send a request to a Domain Controller for an administrator's credentials. The command "kerberos::dcsync" would list the credentials of all users.

When this command is issued without extra options, Mimikatz selects the domain and the Domain Controller automatically, and extract the credentials from this Domain Controller via replication using the Directory Replication Service Remote (MS-DRSR) protocol.

This is a very powerful attack: Once an attacker has obtained domain admin credentials, he/she can use DCSync to connect to a Domain Controller and extract the credentials of the krbtgt account. This data can then be used to create a Golden Ticket, and then it is game over: The only recourse you have is to change the krbtgt account password. This password never expires and is never changed, unless it is done manually. If you discover that the krbtgt NTLM hash has been compromised, you will have to change the password.

It is possible to detect and prevent a DCSync attack. MS-DRSR network traffic should only occur between Domain Controllers. If you detect MS-DRSR network traffic between a Domain Controller and a workstation, you know a DCSync attack took place.

If you segment your Domain Controllers in a dedicated network segment, with advanced firewalls as chokepoints between the other network segments, you can block MS-DRSR traffic outside the Domain Controller network segment.

Domain Dominance – Becoming a Domain Controller – DCShadow
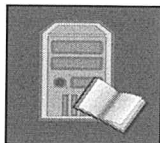
"They told me I could be anything I wanted, so I became a Domain Controller." With this tag line, Benjamin Delpy and Vincent Le Toux introduced their new AD attack strategy early 2018, called "DCShadow". While it appears very similar to DCSync, it's more intrusive and definitely different!

DCShadow rose to prominence after BlackHat USA 2018, where Vincent Le Toux and Benjamin Delpy presented their research. The slides can be found here: https://www.dcshadow.com/us-18-Delpy-LeToux-So-I-Became-A-Domain-Controller.pdf. It has a public website with information as well on https://www.dcshadow.com.

How does the "DCShadow" attack work? There are 4 main steps in a DCShadow attack:

- As a prerequisite, the adversary needs to obtain Domain Administrator rights.
- Using Mimikatz, the adversary temporarily registers the workstation he / she is using as a DC.
- The adversary crafts a change in the schema that benefits him (e.g. change the password hash of a sensitive account).
- Using Mimikatz, the adversary can now trigger replication, which forces a legitimate DC to commit the change!

The changes performed by DCShadow are done using replication, which renders them almost "invisible" for normal Windows event logs. Windows normally relies on event logs generated on the source DC that creates the changes. As in this case, this DC does not exist; no Windows event logs are being generated! We will discuss this further in the "detection" section!

```
mimikatz 2.1.1 x64 (oe.eo)                                              —    □    X
mimikatz # lsadump::dcshadow /object:CN=Administrator,CN=Users,DC=SYNCTECHLABS,DC=COM /attribute:description /value:"DCS
hadow was here!"
** Domain Info **

Domain:        DC=synctechlabs,DC=com
Configuration: CN=Configuration,DC=synctechlabs,DC=com
Schema:        CN=Schema,CN=Configuration,DC=synctechlabs,DC=com
dsServiceName: ,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=synctechlabs,DC=com
domainControllerFunctionality: 7 ( WIN2016 )
highestCommittedUSN: 274541

** Server Info **

Server: DC.synctechlabs.com
  InstanceId  : {95bfcc95-81c9-4225-a71b-a896128a68df}
  InvocationId: {81d3022b-6d8a-4109-ae4a-ceff7f43de9S}
Fake Server (not already registered): WINDOWS02.synctechlabs.com

** Attributes checking **

#0: description

** Objects **

#0: CN=Administrator,CN=Users,DC=SYNCTECHLABS,DC=COM
  description (2.5.4.13-d rev 1):
    DCShadow was here!
    (44004300530068006100640086f0077002000770061007300200068006500072006500021000000)

** Starting server **

> BindString[0]: ncacn_ip_tcp:WINDOWS02[49917]
> RPC bind registered
> RPC Server is waiting!
** Press Control+C to stop **
```

DCShadow – Crafting Change

It's important to note that DCShadow only relies on features that were there by design! It does not abuse any exploit or vulnerability (similar to DCSync).

In the screenshot to the left, we have prepared a small change and are changing the default Administrator account to have a description of "DCShadow was here!".

This step is to be executed in the context of "NT AUTHORITY\SYSTEM", so we need to perform a "token::elevate" first!

Domain Dominance – Becoming a Domain Controller – DCShadow in Action (1)

The screenshot above is step 1 in a DCShadow attack. We are crafting a change that we will subsequently push. It's important to note that DCShadow only relies on features that were there by design (i.e. the Directory Replication Service)! It does not abuse any exploit or vulnerability (similar to DCSync).

In the screenshot to the left, we have prepared a small change and are changing the default Administrator account to have a description of "DCShadow was here!". We should now leave this window open and launch a second Mimikatz window that can be used to initiate the push (we will see this on the next slide). This first step is to be executed in the context of "NT AUTHORITY\SYSTEM", so we need to perform a "token::elevate" first!

```
mimikatz 2.1.1 x64 (oe.eo)                                              –   □   ×

 .#####.   mimikatz 2.1.1 (x64) built on Sep 25 2018 15:08:14
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
 ## / \ ##  /*** 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 # lsadump::dcshadow /push
** Domain Info **

Domain:        DC=synctechlabs,DC=com
Configuration: CN=Configuration,DC=synctechlabs,DC=com
Schema:        CN=Schema,CN=Configuration,DC=synctechlabs,DC=com
dsServiceName: ,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=synctechlabs,DC=com
domainControllerFunctionality: 7 ( WIN2016 )
highestCommittedUSN: 274543

** Server Info **

Server: DC.synctechlabs.com
    InstanceId  : {95bfcc95-81c9-4225-a71b-a896128a68df}
    InvocationId: {81d3022b-6d8a-4109-ae4a-ceff7f43de95}
Fake Server (not already registered): WINDOWS02.synctechlabs.com

** Performing Registration **

** Performing Push **

Syncing DC=synctechlabs,DC=com
Sync Done

** Performing Unregistration **

mimikatz #
```

DCShadow – Triggering Replication

In the next step, we now trigger replication by running the "lsadump::dcshadow /push" command!

The change we crafted and prepared in the previous step will now be persisted.

It's interesting to note how the "promoted" workstation is afterwards unregistered again (it does not become a permanent Domain Controller).

Domain Dominance – Becoming a Domain Controller – DCShadow in Action (2)

Once we have prepared the change (previous step), we should leave the command prompt running on the "RPC Server is waiting" prompt. We will now launch a second Mimikatz window (using the Domain Administrator context this time).

In the second Mimikatz window, we now trigger replication by running the "lsadump::dcshadow /push" command! The change we crafted and prepared in the previous step will now be persisted. It's interesting to note how the "promoted" workstation is afterwards unregistered again (it does not become a permanent Domain Controller).

Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- Day 4: Lateral Movement
- **Day 5: Action on Objectives, Threat Hunting & Incident Response**
- Day 6: APT Defender Capstone

SEC599.5

Domain dominance
  Dominating the AD - Basic strategies
  Golden Ticket, Skeleton Key, DCSync and DCShadow
  Detecting domain dominance
  Exercise: Domain dominance
Data exfiltration
  Common exfiltration strategies
  Exercise: Detecting data exfiltration
Leveraging threat intelligence
  Defining threat intelligence
  Exercise: Leveraging threat intelligence with MISP & Loki
Threat Hunting & incident response
  Proactive threat hunting strategies
  Exercise: Hunting your environment using OSQuery
  incident response process
  Exercise: Finding malware using Volatility & YarGen
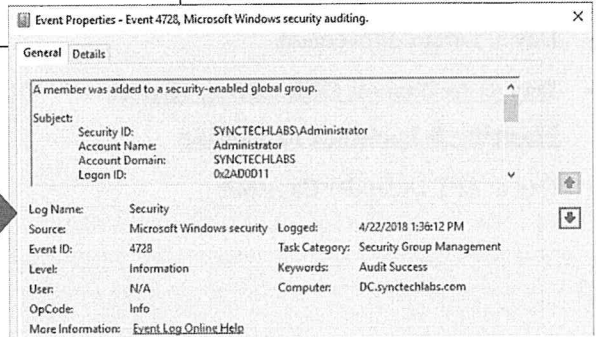
This page intentionally left blank.

```
C:\WINDOWS\system32>net users erik Password123! /domain /add
The request will be processed at a domain controller for domain synctechlabs.com.

The command completed successfully.


C:\WINDOWS\system32>net group "Domain Admins" erik /domain /add
The request will be processed at a domain controller for domain synctechlabs.com.

The command completed successfully.
```

User Erik being added and placed in the "Domain Admins" group. This command can be run on any domain-joined machine, provided domain admin privileges are available!

In the Domain Controller's security log, an event ID 4728 is generated, revealing what user (in this case Administrator) added "erik" to the Domain Admins group!

Event Properties - Event 4728, Microsoft Windows security auditing.   ×

General | Details

A member was added to a security-enabled global group.

Subject:
  Security ID: SYNCTECHLABS\Administrator
  Account Name: Administrator
  Account Domain: SYNCTECHLABS
  Logon ID: 0x2AD0D11

Log Name: Security
Source: Microsoft Windows security   Logged: 4/22/2018 1:36:12 PM
Event ID: 4728   Task Category: Security Group Management
Level: Information   Keywords: Audit Success
User: N/A   Computer: DC.synctechlabs.com
OpCode: Info
More Information: Event Log Online Help

Detecting a Domain Admin Account Being Added
The addition of a domain admin account in a target network is rather noisy and is thus easy to detect for us defenders. Consider the screenshots:

- In the first screenshot, a user "erik" is added on the domain. Note that the commands are executed from a domain-joined workstation where Domain Administrator privileges where available.
- In the second screenshot, we can observe the "4728" event generated in the Domain Controller's security log. This is a rather infrequent event and should thus be cause for investigation!

Encryption Downgrade Activity
The encryption method of the ETYPE_INFO2 field of KRB_ERR message from CLIENT1 has been downgraded based on previously learned behavior. This may be a result of a Skeleton Key on DC4.

Note   Email   Export to Excel   Details   Open

Encryption Downgrade

user1   CLIENT1   Skeleton Key   DC4

Downgraded field
KRB_ERR : ETYPE_INFO2

Recommendations
• Disconnect the relevant computers from the network or move them into an isolated environment and start a forensics procedure by investigating unknown processes, services, registry entries, unsigned files, and more

A script was developed to try detecting compromised DCs (by attempting to use RC4 tickets). You can find it here:
https://gallery.technet.microsoft.com/Aorato-Skeleton-Key-24e46b73

Detecting the Skeleton Key
Technically, the Skeleton Key adds a key for the "mimikatz" password in memory on the DC.

It's important to note that this is only possible for the RC4 encryption type. Unlike RC4 (which uses the NTLM hash as they key) AES keys use individual salts for all users. It's not feasible to calculate AES keys for all users, hence attackers downgrade the Kerberos encryption type to RC4, allowing for detection!

Microsoft ATA has added a use case to detect this behavior!

Detecting the Skeleton Key

Another technique we can try to detect is the "Skeleton Key" technique. Let's remember that, technically, the Skeleton Key attack adds a key for the "mimikatz" password in memory on the DC. This key is added for all users in the domain.

It's important to note that this is only possible for the RC4 encryption type. Unlike RC4 (which uses the NTLM hash as the key), AES keys use individual salts for all users. It's not feasible to calculate AES keys for all users, hence attackers downgrade the Kerberos encryption type to RC4, allowing for detection! This doesn't mean that a highly targeted attack couldn't abuse a similar technique to plant a "Skeleton Key" for one specific user. This is currently, however, not supported by public tools such as Mimikatz.

Microsoft ATA has added a use case to detect this behavior (mainly focused on the fact that the encryption type is downgraded to RC4)! Another interesting PowerShell script was developed that can remotely check for Skeleton Key backdoors. It does the following:

- Verifies whether the Domain Functional Level (DFL) of current domain supports AES (>= 2008).
- Finds an AES supporting client account (msds-supportedencryptiontypes >= 8).
- Sends a Kerberos AS-REQ to all DCs with only the AES E-type supported (Using python code based on https://github.com/SecWiki/windows-kernel-exploits/tree/master/MS14-068/pykek).
- If AS-REQ fails due to AES encryption not supported, then there's a good chance the DC is infected.

The script can be found here: https://gallery.technet.microsoft.com/Aorato-Skeleton-Key-24e46b73

In order to prevent forged TGTs (Golden Tickets) from being valid for an unreasonable period of time, it's a good idea to periodically change the KDC LT key. This can be achieved by resetting the krbtgt account password. If configured properly and with a reasonable period of time, this shouldn't impact the overall AD environment!
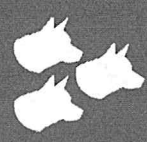
Periodic hunting on end-user systems could be performed to detect TGTs in memory with an unreasonably long validity (e.g. using **klist**). This technique is, however, not suitable for real-time detection and would require vast data collection capabilities!

4769

Using detailed Kerberos logging, we might be able to detect certain anomalies in the forged TGT. Similar to the Silver Ticket (where forged Service Tickets are created), Mimikatz tends to leave behind certain watermarks that could indicate the ticket was forged. Persistent adversaries could, however, further craft a forged ticket in order not to include these watermarks! (**Event ID: 4769**)

Detecting a Golden Ticket
What can we do to defend ourselves from Golden Ticket attacks? This is not straightforward!

In order to prevent forged TGTs (Golden Tickets) from being valid for an unreasonable period of time, it's a good idea to periodically change the KDC LT key. This can be achieved by resetting the krbtgt account password. If configured properly and in a reasonable periodicity, this shouldn't impact the overall AD environment! There is a script available in the Microsoft Technet Gallery:

https://gallery.technet.microsoft.com/Reset-the-krbtgt-account-581a9e51

As previously indicated, Golden Tickets cannot be detected when they are created (as this happens offline without interaction with the AD). Here are a few options to detect the Golden Ticket being used:

- Periodic hunting on end-user systems could be performed to detect TGTs in memory with unreasonably long validity (e.g. using klist). This technique is, however, not suitable for real-time detection and would require vast data collection capabilities!

- Using detailed Kerberos logging, we might be able to detect certain anomalies in the forged TGT. Similar to the Silver Ticket (where forged Service Tickets are created), Mimikatz tends to leave behind certain watermarks that could indicate the ticket was forged. Persistent adversaries could, however, further craft a forged ticket in order not to include these watermarks! The event ID to look for is 4769, where TGT's are submitted to the Domain Controller to request service tickets.

```
Command Prompt                                                      —    □    ×

Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\alan.marshall>klist

Current LogonId is 0:0x48745

Cached Tickets: (4)

#0>     Client: alan.marshall @ SYNCTECHLABS.COM
        Server: krbtgt/SYNCTECHLABS.COM @ SYNCTECHLABS.COM
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x60a10000 -> forwardable forwarded renewable pre_authent name_canonicalize
        Start Time: 12/10/2018 14:22:22 (local)
        End Time:   12/11/2018 0:12:11 (local)
        Renew Time: 12/17/2018 14:12:11 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x2 -> DELEGATION
        Kdc Called: DC.synctechlabs.com
```

In the screenshot above, we can see the output of the "klist" command. This will list all Kerberos tickets (both TGT and Service Tickets) that are in memory. Note the "End Time", which could be reviewed for anomalies / excessive validity times. Some have suggested looking for "RC4" tickets, but this is not a good approach: Mimikatz allows creation of Golden Tickets using AES as well (although it's a bit less well-known)!

Detecting a Golden Ticket Being Used – Klist
The screenshot in the slide shows the output of the "klist" command. This will list all Kerberos tickets (both TGT and Service Tickets) that are in memory. Note the "End Time", which could be reviewed for anomalies / excessive validity times. Some have suggested looking for "RC4" tickets, but this is not a good approach: Mimikatz allows creation of Golden Tickets using AES as well (although it's a bit less well-known)!

As you can see, this is far from an ideal detection mechanism…

Detecting DCSync (Domain Replication Service)

```
Select mimikatz 2.1.1 x64 (oe.eo)
mimikatz # lsadump::dcsync /user:krbtgt
[DC] 'synctechlabs.com' will be the domain
[DC] 'DC.synctechlabs.com' will be the DC server
[DC] 'krbtgt' will be the user account

Object RDN            : krbtgt

** SAM ACCOUNT **

SAM Username          : krbtgt
Account Type          : 30000000 ( USER_OBJECT )
User Account Control  : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration    :
Password last change  : 7/27/2017 8:12:29 PM
Object Security ID    : S-1-5-21-4095063694-3848447163-3403915358-502
Object Relative ID    : 502

Credentials:
  Hash NTLM: a078c51b3fe7a10a7c227af90106a317
    ntlm- 0: a078c51b3fe7a10a7c227af90106a317
    lm  - 0: bd1db11d335bf693643d23c92b438b7a

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 82033c3843aed01d4b0e45cf3fff4bb4

* Primary:Kerberos-Newer-Keys *
    Default Salt : SYNCTECHLABS.COMkrbtgt
    Default Iterations : 4096
    Credentials
```

DRSUAPI	306 DsBind request
DRSUAPI	258 DsBind response
DRSUAPI	830 DsAddEntry request
DRSUAPI	258 DsAddEntry response
DRSUAPI	194 DsUnbind request
DRSUAPI	194 DsUnbind response
DRSUAPI	258 DsBind request
DRSUAPI	258 DsBind response
DRSUAPI	466 DRSUAPI_REPLICA_ADD request
DRSUAPI	434 DsReplicaUpdateRefs request
DRSUAPI	178 DsReplicaUpdateRefs response
DRSUAPI	178 DRSUAPI_REPLICA_ADD response
DRSUAPI	386 DRSUAPI_REPLICA_DEL request
DRSUAPI	178 DRSUAPI_REPLICA_DEL response
DRSUAPI	194 DsUnbind request
DRSUAPI	194 DsUnbind response

When DCSync attacks are mounted against Domain Controllers, IDS engines can detect MS-DRSR traffic being set up!

Detecting DCSync (Domain Replication Service)
In the screenshot above, we are using Mimikatz' built-in function to launch a DCSync attack. We are, however, not performing a full copy, but are limiting ourselves to one of the most precious accounts in the environment: The krbtgt, which is at the core of Kerberos authentication. Though difficult to spot using Windows event logs, IDS engines that look at the network traffic can easily give this away! MS-DRSR traffic between a workstation and a Domain Controller should raise quite a few alarms!

Detecting DCShadow

DRS

As a first recommendation, DCShadow can be easily spotted on the network layer: Similar to DCSync, replication between two systems that are not both Domain Controllers should always be considered suspicious!

EV TX

The changes made by DCShadow are difficult to spot in standard Windows event logging, as the changes originate from replication, not from an actual change. Windows relies on the source Domain Controller to log changes in Windows event logs. As this rogue DC doesn't actually exist, there are no logs.

That being said, DCShadow can be detected by reviewing the Windows event log for the temporary addition of a DC (creation and immediate deletion). Look for **5137** (A directory service object was created) and **5141** (A directory service object was deleted)!

AlsId developed a PowerShell script that can be used to monitor AD environments for DCShadow! You can find it here: https://github.com/AlsidOfficial/UncoverDCShadow/blob/master/README.md

Detecting DCShadow
Can we reliably detect DCShadow? Yes!

First of all, DCShadow can be easily spotted on the network layer: Similar to DCSync, replication between two systems that are not both Domain Controllers should always be considered suspicious! The exact same recommendation that is applicable to DCSync is also applicable to DCShadow!

The changes made by DCShadow are difficult to spot in standard Windows event logging, as the changes originate from replication, not from an actual change. Windows relies on the source Domain Controller to log changes in Windows event logs. As this rogue DC doesn't actually exist, there are no logs. That being said, DCShadow can be detected by reviewing the Windows event log for the temporary addition of a DC (creation and immediate deletion). Look for 5137 (A directory service object was created) and 5141 (A directory service object was deleted)!

Furthermore, AlsId developed a PowerShell script that can be used to monitor AD environments for DCShadow! You can find it here:
https://github.com/AlsidOfficial/UncoverDCShadow/blob/master/README.md

Course Roadmap

SEC599.5

Domain dominance
- Dominating the AD - Basic strategies
- Golden Ticket, Skeleton Key, DCSync and DCShadow
- Detecting domain dominance
- Exercise: Domain dominance

Data exfiltration
- Common exfiltration strategies
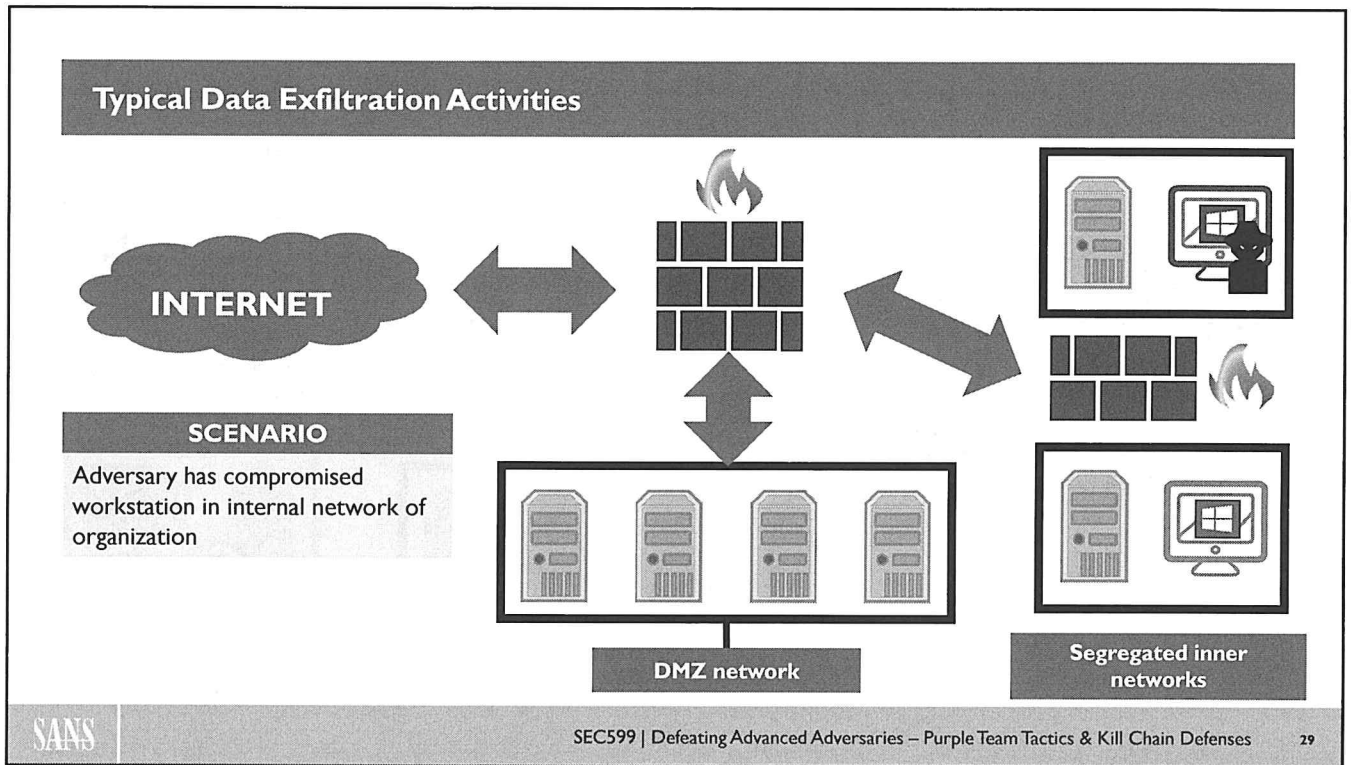- Exercise: Detecting data exfiltration

Leveraging threat intelligence
- Defining threat intelligence
- Exercise: Leveraging threat intelligence with MISP & Loki

Threat Hunting & incident response
- Proactive threat hunting strategies
- Exercise: Hunting your environment using OSQuery
- incident response process
- Exercise: Finding malware using Volatility & YarGen

This page intentionally left blank.

Exercise: Domain Dominance

Please refer to the workbook for further instructions on the exercise!

This page intentionally left blank.

Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- Day 4: Lateral Movement
- **Day 5: Action on Objectives, Threat Hunting & Incident Response**
- Day 6: APT Defender Capstone

SEC599.5

Domain dominance
   Dominating the AD - Basic strategies
   Golden Ticket, Skeleton Key, DCSync and DCShadow
   Detecting domain dominance
   Exercise: Domain dominance
Data exfiltration
   Common exfiltration strategies
   Exercise: Detecting data exfiltration
Leveraging threat intelligence
   Defining threat intelligence
   Exercise: Leveraging threat intelligence with MISP & Loki
Threat Hunting & incident response
   Proactive threat hunting strategies
   Exercise: Hunting your environment using OSQuery
   incident response process
   Exercise: Finding malware using Volatility & YarGen

This page intentionally left blank.

Typical Data Exfiltration Activities

Once (advanced) adversaries have a first foothold in the environment, they will start attempting to reach their objectives. "Actions on objectives" is a broad and generic term that encompasses many activities performed by attackers. One of these activities is data exfiltration. We will now walk through a couple of steps that illustrate how attackers typically steal and exfiltrate interesting data.
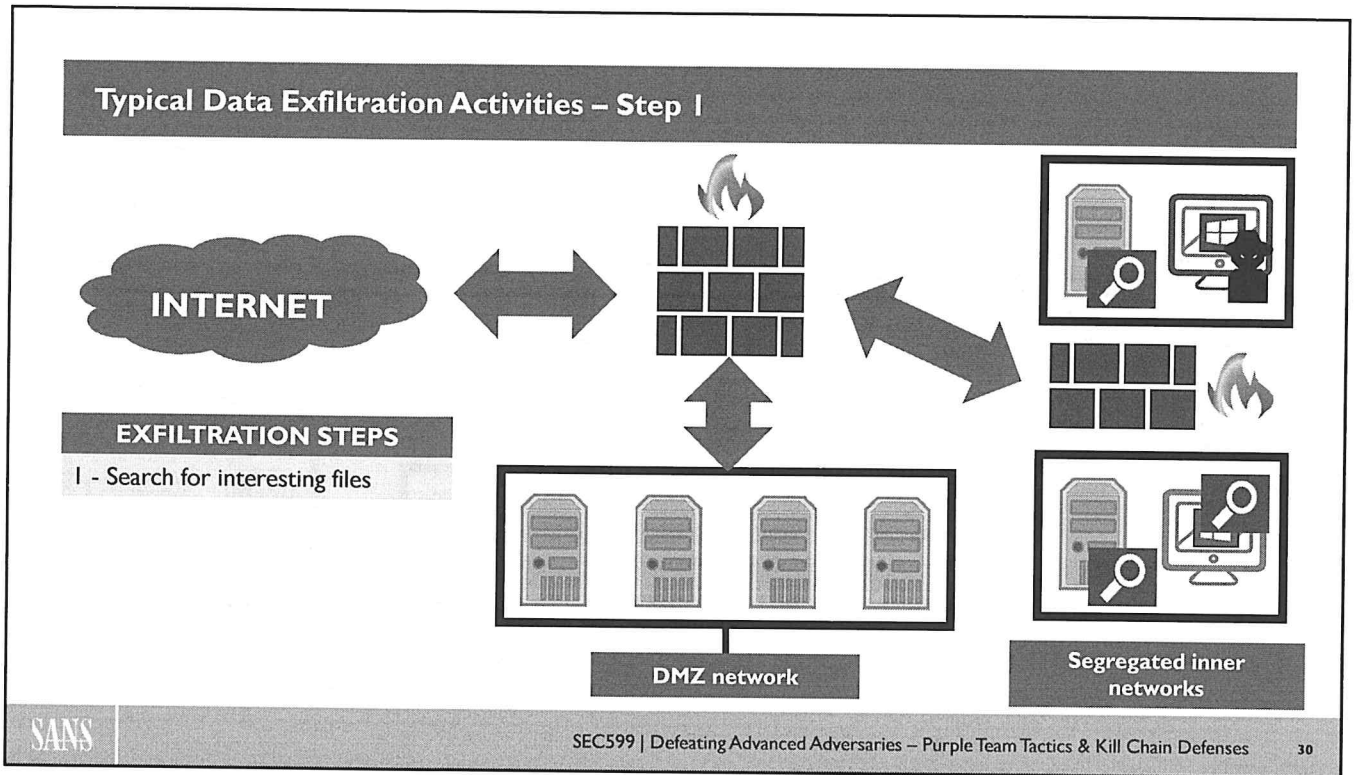
Typical Data Exfiltration Activities – Step 1

As a first step, adversaries will need to search for interesting files. Usually, the adversary is on "foreign soil" when he infiltrates your environment. That typically means he doesn't immediately know where you are storing your crown jewels. In order to achieve this, he will have to search your environment for possibly interesting information.

As he will have to search your environment, he is bound to be rather noisy and could even generate errors that could reveal his activities (e.g. as the current user he compromised doesn't have access to the top-secret information he so desperately wants to obtain). We will discuss this more in-depth in the next series of slides.
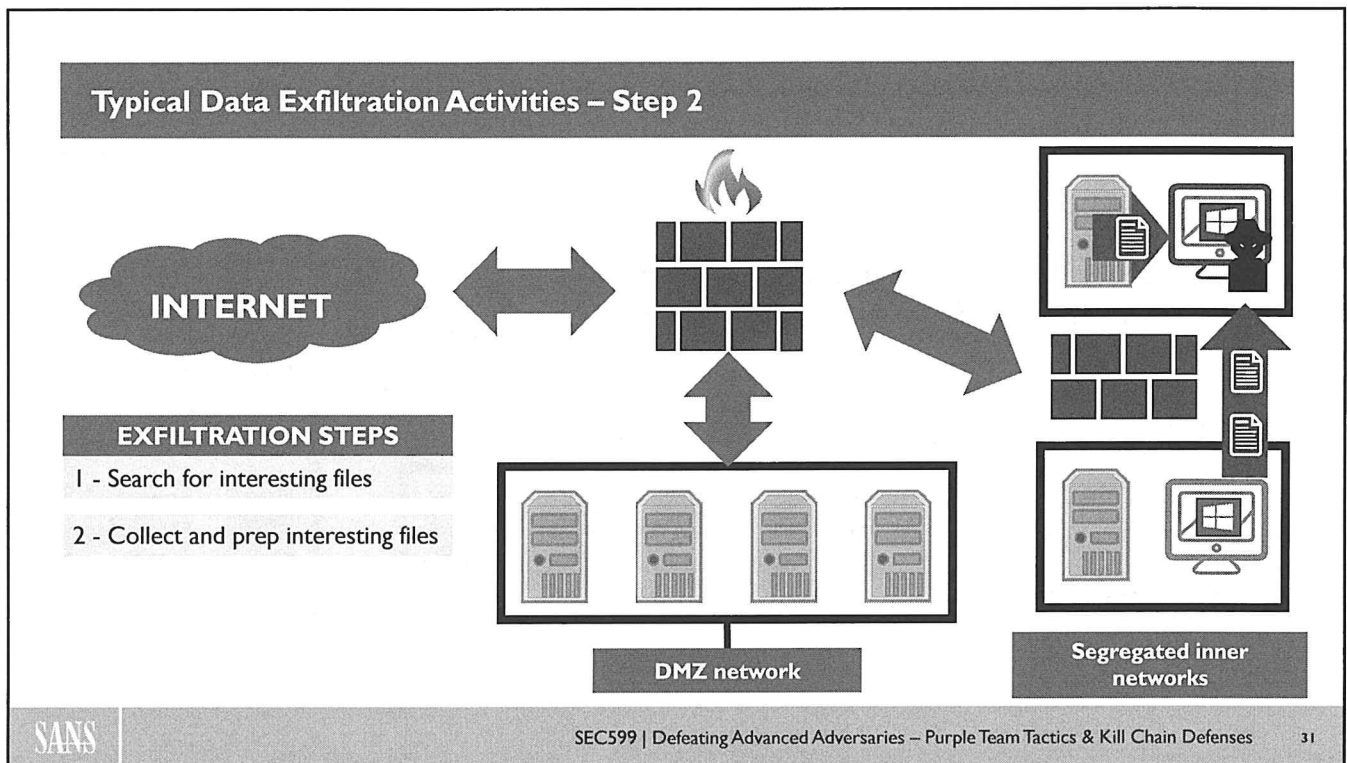
Typical Data Exfiltration Activities – Step 2

A typical data exfiltration tactic used by (advanced) attackers is to gather all collected data to be exfiltrated in one place. Through various search operations we discussed in previous slides, attackers will locate interesting files and data. Often, attackers will steal a lot of data, so that they can sift through it on their own systems, out of reach of corporate surveillance. For example, it happens regularly that attackers copy the complete mailbox of individuals or even all emails on email servers.

This can represent a huge amount of data (gigabytes and more), and it is not practical to go through this data online and exfiltrate it manually with a copy/paste for example. Exfiltrating significant amounts of data requires planning and organization.

What we have observed is that attackers will often gather all the data they deem interesting in one place: For example, inside a folder on a computer system they compromised. All these files will be put inside an archive and may be encrypted.

Typical Data Exfiltration Activities – Step 3

EXFILTRATION STEPS

1 - Search for interesting files

2 - Collect and prep interesting files

3 - Exfiltrate interesting files

DMZ network

Segregated inner networks

INTERNET

SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses   32
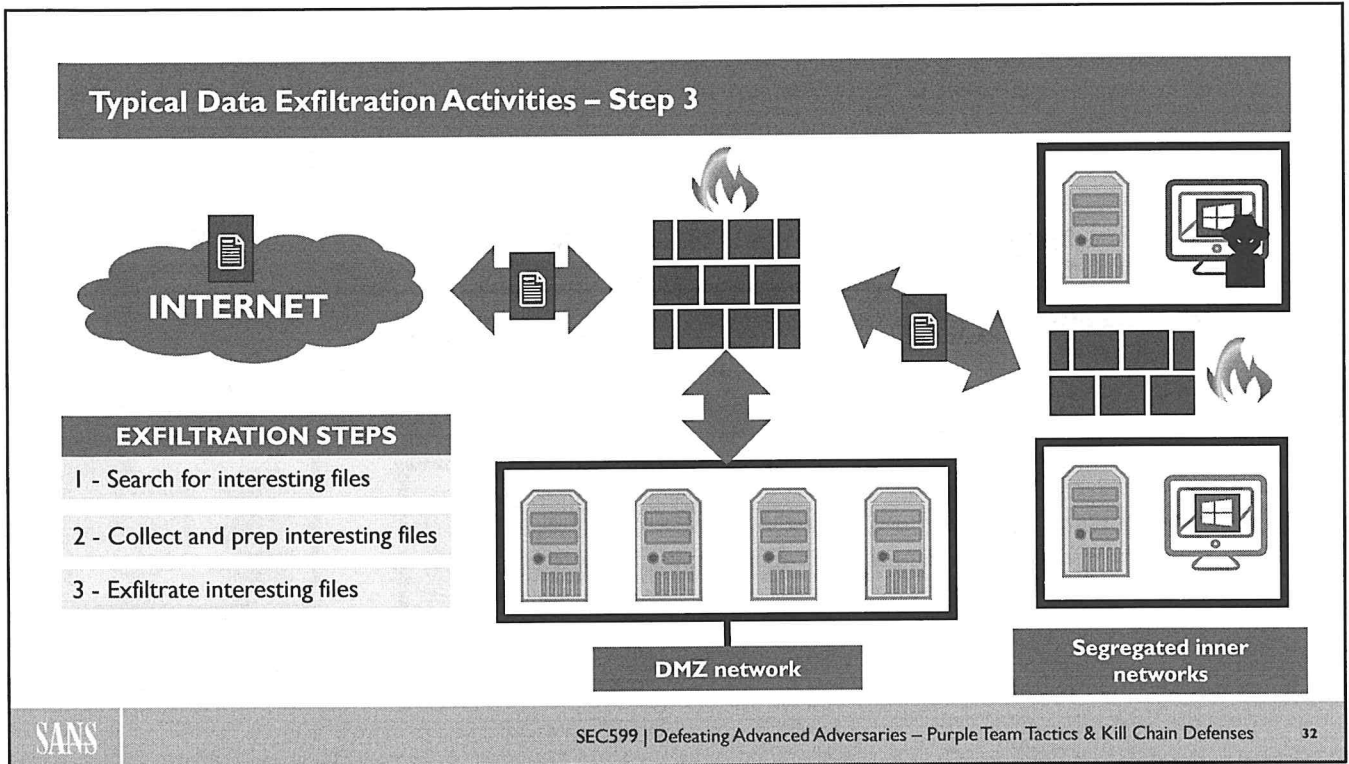
Typical Data Exfiltration Activities – Step 3

Most adversaries will perform data exfiltration over the corporate network infrastructure. That being said, physical exfiltration is possible, but this typically requires the adversaries to have people on-site, which is usually not the case.

Whether attackers will have many or limited options to perform data exfiltration over the network will depend on the design of your corporate network. If it is a flat network connected to the internet, they will not encounter insurmountable obstacles. A properly segmented network will be more difficult for the attackers, both to gather data from different segments and to exfiltrate data to the internet.
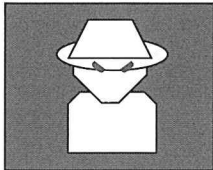
One of the actions on objectives that adversaries will take is collecting data from various sources inside the compromised organization. There is just one problem with that for adversaries: They are blind in your environment and don't know where to look for data!

- Before data can be collected, it must be first identified and located within the environment.
- There are various tools that can search for files and other data.
- Windows built-in search functions can be used to locate files of interest.
- During the infamous "Carbanak" campaign, the adversaries installed monitoring software, so they could spy on users to understand how they could reach their objectives!

Step 1 – Searching Data of Interest

Once (advanced) adversaries are inside our network and have access to our system, they will proceed with the next phase of the attack: "actions on objectives."

"Actions on objectives" is a broad and generic term that encompasses many activities performed by attackers. One of these activities is data exfiltration.

Data exfiltration is stealing your data, put simply. Attackers will search for interesting data stored in your IT infrastructure: Confidential documents, planning, research results, financial data, bookkeeping, company secrets … you name it. Then this data will be collected and exfiltrated: Attackers will transfer it out of your corporate IT infrastructure to a system they control.

Before attackers can exfiltrate data, they have to identify and locate it. This may sound obvious, but in a large organization, it might require some work to sift through files and data to find what attackers are looking for. Windows (and other operating) systems have tools that help users locate files and data, based on metadata (like filename) and file content. These tools can be used by attackers, too, to search for their "grail."

These search functions can vary from basic to sophisticated, like Cortana in Windows 10.

During the infamous "Carbanak" campaign, the adversaries installed monitoring software, so they could spy on users to understand how they could reach their objectives!

Step 1 – Searching Data of Interest – Using Built-in Tools

Step 1 – Searching Data of Interest – Using Built-in Tools

Here, we start with a very basic file search operation using the "old shell", the Windows command-line interpreter cmd.exe. cmd.exe offers various commands, like the dir command. Dir stands for directory: It returns the content of a directory by listing all the files and directories inside a directory, together with some metadata such as the file size.

When the dir command is executed without any arguments or options, it will produce a directory listing of the current directory. Dir can be instructed to list the content of a particular directory, for example, c:\demo. The command is "dir c:\demo", this will produce a directory listing with the content of the c:\demo directory. The dir command can search for specific files, too, with the /s option.

For example, command "dir /s secret.doc" will search for files with name secret.doc inside the current directory and all underlying sub-directories. To search through a complete filesystem, the document to search for should be prefixed with the root directory of the file drive to be searched. For example, for drive C:, the command is "dir /s c:\secret.doc".

PowerShell can, of course, do similar things as cmd.exe. One command that can be used to locate files, is Get-ChildItem. Get-Childitem takes many options. The –Path option allows us to specify where to start searching. In this example, we search in the c:\research-and-development directory.

We can filter for specific names with the –Include option: -Include *confidential* will select all files (and directories) with the string confidential in the filename (* is a wildcard, just like with cmd.exe).

With option –File we search only for files and ignore directories (e.g. directories that match the name *confidential* will not be listed). By default, the Get-Childitem command only searches in the provided directory, and not the underlying directories. To achieve searching through subdirectories, option –Recurse must be provided. Finally, with option –ErrorAction we can make that the Get-Childitem command continues searching even when an error occurs.

Attackers will not only be interested in documents found on the machine they are logged into, but also on remote machines. With the classic command shell, cmd.exe, commands like dir can only be executed on the logged in machine and not on remote machines. A share can be mapped to a drive on the local machine and then be searched through remotely with the dir command, but this will require more network bandwidth and will be less performant because the directory structure has to be transferred from the remote machine to the local machine.

In the example above, we use the command Invoke-Command with option –ComputerName to issue a PowerShell command on remote computer fileserver01. The command is the Get-Childitem command we saw in the previous slide.

The difference between invoking a command on a remote computer, and mapping a remote drive on a local computer, is bandwidth and performance. By issuing the command remotely, it will be faster, because only the result (the output of the command) has to be transferred over the network, and not the complete directory structure.

Some security tools installed on machines can be repurposed by the attacker to search for interesting content.

 Some antivirus scanners (such as ClamAV) allow for searching with customer-defined signatures.
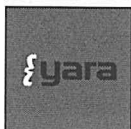
 Other tools include the YARA engine, which can also be tuned with custom rules.

If these tools are not present, the attacker can deploy them without installation.

Step 1 – Searching Data of Interest – Getting Creative
How can advanced attackers "Hijack" existing (security) tools to search for interesting files?

Some security tools installed on machines can be repurposed by the attacker to search for interesting content.

Remember that we discussed antivirus applications and other malware searching tools like YARA: It is possible to use this tool to search for documents, too.
Some anti-virus scanners (like ClamAV) allow searching with customer-defined signatures. Other tools include the YARA engine, which can also be tuned with rules.

If these tools are not present on the compromised machine, the attacker can easily deploy them without installation. ClamAV and YARA do not require installation, just copying the executables and supporting files on the compromised machine is enough to be able to use these tools. They do not require administrative rights to operate.

YARA is a flexible, multipurpose search tool based on rules. The rules that we use are designed to detect malware and other unwanted files, but there is actually nothing to stop a user logged in on a system with YARA from creating his own rules and using YARA to search through the file system.

```
rule ConfidentialDocuments
{
    strings:
        $a = "confidential" ascii wide nocase
        $b = "secret" ascii wide nocase
        $c = "classified" ascii wide nocase

    condition:
        any of them
}
```

We have seen the use of YARA rules to detect activities of our adversaries.

Since YARA is a portable tool, adversaries can use it to search for confidential data.

The following rule will trigger on all files that contain the word "confidential", "secret" or "classified".

Step 1 – Searching Data of Interest – Getting Creative with YARA!
To illustrate how existing security tools can be hijacked to facilitate data exfiltration, we will discuss a YARA rule designed to search for confidential documents.

As we saw, YARA is a flexible, multipurpose search tool based on rules. The rules that we use are designed to detect malware and other unwanted files, but there is actually nothing to stop a user logged in on a system with YARA from creating his own rules and using YARA to search through the filesystem.

In the example above, we illustrate this with a simple rule that will search for documents that contain at least one of these words:

- confidential
- secret
- classified

The options ascii and wide make that YARA will search for these words in ASCII and UNICODE form. The option nocase instructs YARA to disregard the case of a word when matching with these keywords.

This simple rule, when used with YARA to scan through a complete filesystem, will locate all documents that contain one of these keywords.

PREVENT	DETECT
• Ensure the organization knows what data they possess & that it is correctly classified.	• A system-wide search generates a lot of activity on the system being searched.
• Limit user access rights only to data they should be allowed to access ("need-to-know").	• Monitoring for searches through file systems is not trivial, though, there will be several false positives: • Antivirus scanners, search indexers, backup programs, etc.
• Next to limiting user access to data, also consider what type of data you store where... This includes network segmentation, but also even considering storing some data offline!	• Access to network shares can however be monitored (event ID 5140 – "A network share object was accessed"), look for repeated audit failures from one source!

Step 1 – What Can We Do as Defenders?
As always, as defenders we have two types of controls we can put in place:

PREVENT
In order to prevent adversaries from stealing sensitive data, it's important for an organization to know what data they possess and that it is correctly classified. Based upon this classification, user access rights should be highly limited, and users should only have access to what they need to fulfill their daily jobs ("need-to-know" principle).

Next to limiting user access to data, also consider what type of data you store where... This includes network segmentation, but also even considering storing some data offline!

DETECT
It is a fact that searching through a complete file system is "very noisy". When we search for files inside a filesystem with cmd.exe or PowerShell, these programs will open all directories to list the files and directories inside it. Depending on the number of files and directories inside a filesystem, this can require opening 10,000 or more directories, and thus produce a considerable number of activities. Searching with an index is different: The filesystem does not need to be searched through, only the index itself.

One would think that this kind of activity would be monitored, but by default on Windows, accessing files and directories is not logged with Windows events. Windows can be configured to generate events for file activities, but doing this indiscriminately would generate a huge amount of events. And when you would configure this and monitor the results, there will be several false positives. The same behavior is exhibited by legitimate programs, like
- antivirus programs
- Backup programs
- Search indexers

The mentioned types of programs access the complete filesystem, which would create false positives when trying to detect system-wide searches for sensitive information. Exceptions have thus to be created.

A Word on Data Classification

Advanced adversaries will quickly identify important data and try to steal it. Important data must be adequately protected, but do your employees know what data is important and what data is not important?
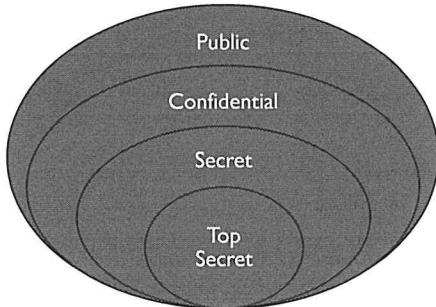
Public
Confidential
Secret
Top Secret

A data classification policy and proper training of your employees will help classify data accordingly.

This data classification policy will dictate the classification of data into different classes (levels).

One way to define a level is to analyze what impact the loss of data will have on the business, and then to specify a data classification level accordingly.

When data is classified, data with a high classification (e.g. confidential) can be separated from public data, for example.

A Word on Data Classification

From the example we gave on searching data, it is clear that advanced adversaries will quickly identify important data and try to steal it. Because important data is crucial to your business, it must be adequately protected. Before it can be adequately protected, it must be identified. But do your employees know what data is important and what data is not important?

A data classification policy and proper training of your employees will help classify data accordingly. This data classification policy will dictate the classification of data into different classes (levels). One way to define a level is to analyze what impact the loss of data will have on the business, and then to specify the data classification level accordingly.

For example, data that would endanger the further existence of the company when it would be leaked would receive the highest classification level and should then be handled accordingly. When data is classified, data with a high classification (e.g. confidential) can be separated from public data, for example.

Here is an example of classification levels:

- Top secret
- Secret
- Confidential
- Public

- Another obvious statement: Data that is offline is hard to steal.
- Advanced adversaries that attack your enterprise via digital intrusion can only steal online data.
- Data that is kept offline is not accessible to a digital intruder.
- Most data has to be online of course, but archived data for example can be stored on storage media that is not directly online, e.g. not served by a file server.
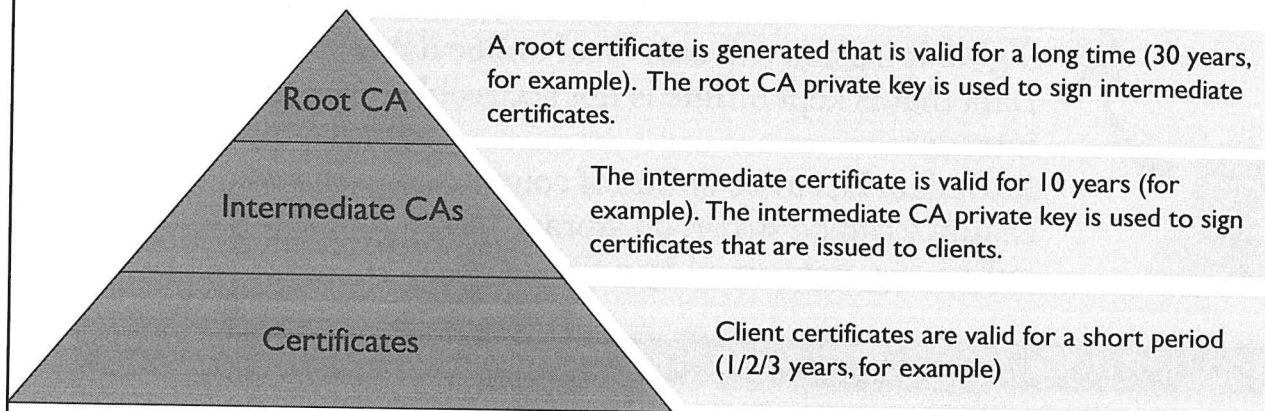
If this is thought through, this will also limit the impact of ransomware attacks!

Should You Store All Your Data Online?

A second obvious statement: Data that is offline is hard to steal. Data that is stored on media that is connected to a server is online and accessible, but data that is stored on media without being connected to a server is offline. It cannot be accessed without connecting the media to a server.

As most advanced adversaries attack your enterprise via digital intrusion (remotely), they have no physical access to the IT assets of your corporation and thus can only access online data.

Data that is kept offline is not accessible to a digital intruder. Most data must be online, of course, to be able to serve its purpose to the business but establishing policies that dictate which data can be stored offline help prevent the scale of data theft. Archived data, for example, can be stored on storage media that is not directly online, e.g. not served by a file server.

Consider:
- Old reports
- Terminated projects
- Old emails
- ...

This type of data might have to be kept for legal reasons, but then it can be archived offline. By not storing "all" of your data online, you can also limit the impact of ransomware attacks!

A good example of highly confidential data that is kept offline can be found in the private key infrastructure (PKI) of certification authorities (CA).

Root CA	A root certificate is generated that is valid for a long time (30 years, for example). The root CA private key is used to sign intermediate certificates.
Intermediate CAs	The intermediate certificate is valid for 10 years (for example). The intermediate CA private key is used to sign certificates that are issued to clients.
Certificates	Client certificates are valid for a short period (1/2/3 years, for example)

Example – Data to Keep Offline (1)
A good example of highly confidential data that is kept offline can be found in the private key infrastructure (PKI) of certification authorities (CA).

Certification authorities are organizations that issue certificates: Signed public keys with metadata. To sign this data, they require private keys. But if those private keys are compromised, CAs can be impersonated and the attacker can issue certificates that cannot be distinguished from legitimate certificates issued by the CA.

To prevent the compromise of private keys, CAs will store critical keys offline.

This is implemented as follows, using a pyramid of at least 2 certificates:

A root certificate is generated that is valid for a long time (30 years for example): This includes the public and private key and metadata.

Then this root certificate is used to sign an intermediate certificate.

In a PKI infrastructure, the private key of the root certificate is no longer needed to operate the infrastructure, as long as the intermediate certificate remains valid and its private key is not compromised.

Storing the private key of the root certificate on removable media kept in a safe, keeps it offline and out of reach of digital intruders.

Intermediate keys can also be protected, stored on so called HSM devices (Hardware Security Modules), these devices store the private key for safekeeping, and they can sign requests, but the key can never be extracted from the HSM.

Organizations should use this mindset and critically assess what information should be kept "online" and what kind of information can be stored "offline"!

Example: Data to Keep Offline (2)
The intermediate certificate has also a public and private key with metadata, but it is typically valid for a shorter period of time. For example, 10 years.

Then this intermediate certificate is used to sign and issue certificates to clients.

This means that the private key of the root certificate is no longer needed to operate the PKI infrastructure: It is the intermediate key that is used to sign certificates, as long as the intermediate certificate remains valid (it expires after 10 years) and is its private key is not compromised.

This architectural design of PKI systems make that the private key of the root certificate can be safely stored offline, it does not have to be online to sign certificates.

The private key of the root certificate can be stored on removable media and kept in a safe, where it is out of reach of digital intruders.

Intermediate keys will also be protected: They are stored on so-called HSM devices (Hardware Security Modules); they store the private key for safekeeping, and they can sign requests, but the key can never be extracted from the HSM.

Case Study – DigiNotar

Having its PKI infrastructure compromised can bankrupt a CA. One example of this was the Dutch CA DigiNotar.

This is a strong example that illustrates the fact that companies can go out of business just because of data theft.

DigiNotar was a Dutch certification authority, owned by VASCO Data Security International. Like many CAs, its root certificates were part of the certificate store of many browsers and operating systems, making those applications trust certificates issued by DigiNotar.

DigiNotar suffered a digital breach: A hacker gained access to DigiNotar's PKI infrastructure.

The hacker issued fraudulent certificates in July 2011. These certificates were used in Iran to conduct man-in-the-middle attacks against users of Google's services.

In August 2011, certificate problems emerged in Iran, and Google detected the fraudulent certificates. It issued a public statement and urged DigiNotar to take action.

DigiNotar did not take timely the appropriate actions, and major browsers started to pull DigiNotar's root certificates from their stores, resulting in HTTPS connections that were no longer trusted. This impacted DigiNotar's business significantly, as clients were forced to obtain new certificates from other CA's.

In September 2011, DigiNotar filed for bankruptcy because major browsers no longer trusted DigiNotar's certificates.

Decoy files?

One method to detect system-wide filesystem searches uses decoy files.

- Decoy files are unimportant files that are designed to attract the attention of attackers.
- For example, a document with an enticing filename like "top-secret-project".
- Access to decoy files is closely monitored.

Another Creative Approach... Detecting Adversaries Accessing Decoy Files
A possible solution to the problem of monitoring and an overflow of alerts is to only monitor a highly specific number of files, which are actually decoy files. When they are interacted with, we can immediately raise an alert.

One method to achieve this is the use of decoy files.

Decoy files are not legitimate corporate files, but they are files planted on filesystems to attract the attention of attackers. They do not contain (important) data but have enticing names to attackers. An example of such a name can be "top-secret-project". Access to these files is closely monitored, while access to other files is not. This tactic will significantly reduce the number of events produced when these files are accessed.

Step 2 – Collecting and Prepping the Data

Attackers will often gather all collected data in one place before exfiltration:

- Encryption
- Compression
- Splitting

Use of archival tools could be detected by filtering the Windows command line logging for typical "archival" syntaxes (archive extensions, tool command line…)
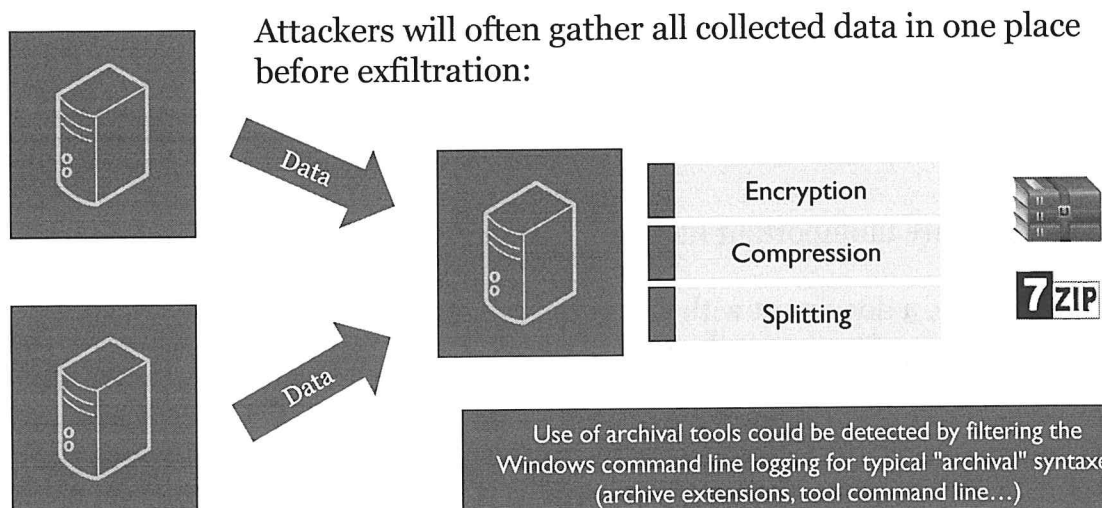
Step 2 – Collecting and Prepping the Data

Once interesting information is located, the data is typically centralized to an internal system that has already been compromised. Here, the data can be prepared to be exfiltrated:

- It can be compressed or split in smaller chunks to hinder detection.
- It can be encrypted to hide the contents of the data that is being exfiltrated.

Given these features, we often see that typical (portable) archiving utilities are used to prepare the data, as they offer both use cases described above. Good candidates include:

- 7z
- RAR
- ...

As a detection strategy, it might be a good idea to keep an eye out for archival tools being used in the environment. One concrete way of doing so would be to attempt detection of archival tools by filtering the Windows command line logging for typical "archival" syntaxes (archive file extensions, tool command line syntaxes…).

In most cases, network exfiltration is the attackers' modus operandi ...

- Attackers will gather all collected data on one machine with network access.

- To speed up network transfer, files will be grouped together in one or more archives and compressed.

- To prevent keyword-based exfiltration detection, the data can be encrypted.

- Classic data transfer methods like HTTP/HTTPS uploads or email attachments will be used.

Step 3 – Exfiltrating the Data – Using the Network

Most advanced adversaries will attack over the network and will not consider physical data exfiltration as an option. They have no physical access to our corporate infrastructure, neither do they have accomplices that do.

Whether attackers will have many or limited options to perform data exfiltration over the network will depend on the design of your corporate network. If it is a flat network connected to the internet, they will not encounter insurmountable obstacles. A properly segmented network will be more difficult for the attackers, both to gather data from different segments and to exfiltrate data to the internet.

Of course, the fact that data leaves your network is normal. Just the mere fact of visiting a website implies that data leaves your network (albeit a very small amount). But for example, the User Agent String of your browser is included as a header, as our cookies, and so on. This is just to illustrate the fact that strictly speaking, there is always data leaving your network. Data leaving your network is normal: Detecting unauthorized data exfiltration can be a challenge.

One very popular avenue for data exfiltration is the use of cloud-based file hosting and file sharing services. There are many cloud-based file hosting and file sharing services available, many with a free tier:

- If you allow the use of these cloud-based file hosting services in your corporate environment, advanced attackers will use them for data exfiltration.
- Proxies and firewalls can be configured to block access to these services.
- Exceptions can be made for particular users.
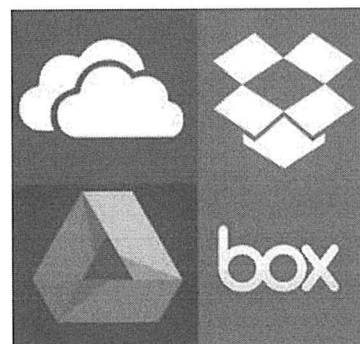
Step 3 – Some Popular Options for Exfiltration
One very popular avenue for data exfiltration is the use of cloud-based file hosting and file sharing services. Many of them have a free tier. Popular examples are:

- OneDrive
- Dropbox
- Google Drive
- Box

If you allow the use of these cloud-based file hosting services in your corporate environment, then it is game over: Advanced attackers will use them for data exfiltration, there is no doubt about that. These services are easy and reliable, they are anonymous (creating an account requires an email to identify, not legal ID) and have no problem operating in environments with proxies.

It is possible to configure proxies and next-generation firewalls to block access to these services, and we highly recommend that you would do this for your corporate environment. These services are the goto-service for data exfiltration.

If there is a business need to allow access to these services, you should identify which users fulfill that business role and make exceptions for these particular users, while blocking access for all other users.

Step 3 – Other Online "Storage"

Next to social media, many websites that are not considered social media allow upload of data that can be retrieved later

Paste sites	Online services

Paste sites

Any data can be posted to a text storage data by converting it first to a non-binary format, like hexadecimal for example. Example sites are:

- pastebin.com
- pastie.org
- codepad.org
- tinypaste.com
- Etc.
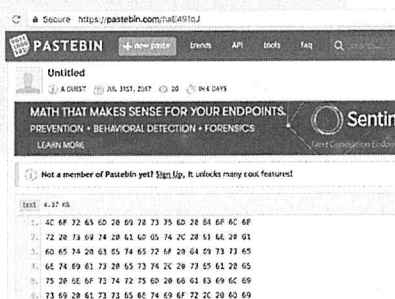
Online services

Other examples of websites that allow posting of data for later retrieval:

- Forums
- Blogs like Wordpress.com
- Source code sharing
- Wikipedia
- VirusTotal
- Etc.

```
4C 6F 72 65 6D 20 69 70 73 75 6D 20 64
72 20 73 69 74 20 61 6D 65 74 2C 20 61
6D 65 74 20 63 65 74 65 72 6F 20 64 69
6E 74 69 61 73 20 65 73 74 2C 20 73 65
75 20 6E 6F 73 74 72 75 6D 20 66 61 63
73 69 20 61 73 73 65 6E 74 69 6F 72 2C
6E 69 6D 75 6D 20 6D 6F 64 65 72 61 74
```

Step 3 – Other Online "Storage"

Paste sites
File sharing and social media is not the sole avenue for attackers to exfiltrate data.

Many websites that are not considered social media still allow upload of data that can be retrieved later. Take for example text storage websites like pastebin.

Pastebin is a website that allows users to (publicly) and anonymously (even without an account) upload text to the website that is visible to all. Such an uploaded text file is called a "pastie" and is accessible to all given the URL that identifies it.

Pastebin is blocked by many corporations because it has been abused in many forms: To share confidential data (knowingly and unknowingly), malware, illegal content, …

As a text storage service, the amount of data that can be uploaded is, of course, limited. Data can be spread over different uploads, but still, exfiltrating one gigabyte of data would be considered impossible.

Although pastebin and similar services allow pasting of text, they do not allow pasting of arbitrary, binary data. This can, however, be easily "solved" by the attackers by converting the data to a non-binary format like hexadecimal or base64, for example. This "text" can then be pasted to pastebin.

We certainly advise blocking text storage services like pastebin, not only because of data exfiltration (even considering the limit size of the upload) but because users tend to use it (accidentally or intentionally) to share confidential data…

Online services

Text storage websites are not the only sites that can be used (or abused) to upload data (data in its binary form or converted to text).

There are many other types of websites that allow posting of data.

Forums and blogging platforms (like wordpress.com) allow for posting of text, but the bandwidth is limited, like with text storage services.

Source code sharing services like GitHub, for example, offer much more capabilities, as well for capacity as for data type (binary data is also accepted).

And then there are less obvious services that can also be used to exfiltrate data.

Take for example VirusTotal, a service that accepts files up to 128MB for antivirus scanning. Any user can upload a file to VirusTotal without an account. What is less known is that all files uploaded to VirusTotal can also be downloaded, provided a subscription fee is paid to VirusTotal.

Wikipedia is another example of a website that can be abused to exfiltrate data. Existing Wikipedia articles can be modified (pasting text, binary data in text form, pictures, …) or new ones can be created. This does not require authentication or authorization. The Wikipedia community will, however, quickly discover such changes and undo them because they are not considered appropriate. A little-known fact is that a history of changes to each page is kept. Even when a change is simply undone.

For example, an attacker can post data to exfiltrate in hexadecimal form to the Wikipedia article on the SANS Institute, and then immediately undo the change he applied. The hexadecimal data will still be easily retrievable in the history of the SANS article.

Another simple way to exfiltrate data is web-based email, like Gmail. Gmail accepts attachments up to 25MB in size.

Depending on what protocols you allow to exit your corporate network, more exotic protocols can be considered, including covert channels such as DNS, ICMP... It should be noted that these would typically not be suitable for large volumes of data!

```
▶ Frame 131: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▶ Ethernet II, Src: Apple_9f:a6:71 (98:01:a7:9f:a6:71), Dst: Technico_4d:02:b8 (c4:ea:1d:4d:02:b8)
▶ Internet Protocol Version 4, Src: 10.10.10.80, Dst: 8.8.8.8
▼ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x5ba9 [correct]
    [Checksum Status: Good]
    Identifier (BE): 37174 (0x9136)
    Identifier (LE): 13969 (0x3691)
    Sequence number (BE): 0 (0x0000)
    Sequence number (LE): 0 (0x0000)
    [Response frame: 132]
    Timestamp from icmp data: Jul 31, 2017 10:56:27.185824000 CEST
    [Timestamp from icmp data (relative): 0.000068000 seconds]
  ▼ Data (48 bytes)
      Data: 08090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f...
      [Length: 48]
```

Example of data exfiltration through ICMP Echo request

Step 4 – Network Exfiltration – Getting Creative

All the data exfiltration examples we saw until now were based on websites. Thus, TCP connections were established to convey HTTP/HTTPS protocols. But there are many other protocols that can be used to exfiltrate data. Whether these can be used in your corporate network, it all depends on the design of your corporate network. Popular protocols that have been used for data exfiltration (and other nefarious purposes) are:

- FTP: File Transfer Protocol
- IRC: Internet Relay Chat
- Email protocol SMTP: Simple Mail Transfer Protocol
- Email protocol POP3: Post Office Protocol
- Email Protocol IMAP: Internet Message Access Protocol

Some of these protocols are also implemented over HTTP, like FTP: It is possible to access and upload files to an FTP server using a browser. And if you allow raw TCP connections (for example over port 80), then large amounts of binary data can be exfiltrated. TCP and protocols based on TCP allow for exfiltrating of large amounts of data quickly.

But if the amount of data to be leaked is small (say 1MB or less), then non-TCP protocols are an option, too. For the sake of presenting as diverse possibilities as possible, we illustrate this with a couple of protocols that have been abused by (advanced) adversaries to exfiltrate data.

ICMP is the protocol that is used to send ping packets. It's a little-known fact that ping packets can contain arbitrary data. In the example above, we see the dissection of a ping request packet in Wireshark. It contains 48 bytes of data. The data that is used depends on the operating system. However, a user can use the ping command on Linux to inject his own data with the pattern option.

On Windows, the ping command does not allow this, but it can be done via the Windows API using scripting for example. The amount of data is very small, but numerous ping packets can be generated to transmit a larger amount of data.

DNS has also been abused, by encoding data in the name of the subdomain, or by using TEXT records.

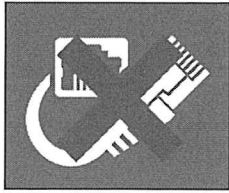## Step 4 – Network Exfiltration – Getting Even More Creative

At Blackhat USA 2017, Itzik Kotler & Amit Klein (SafeBreach Labs) illustrated a technique that abuses "cloud-enabled" AV engines to perform successful data exfiltration in environments with highly restricted outbound filtering. The attack goes as following:

1	The attacker obtains a foothold in the network with a malware sample that evades AV detection.	**4**	The known sample is spotted by the AV engine and sent to a cloud sandbox for further analysis.
2	The attacker walks through the kill chain and successfully collects interesting data.	**5**	The sample is executed in the cloud sandbox of the AV vendor.
3	The data is collected and embedded in a known malware sample, after which it is written to disk.	**6**	Upon execution, the sample sends out the exfiltrated data to the attacker.

In their Proof of Concept, exfiltration was successful for several mainstream AV vendors!

Step 4 – Network Exfiltration – Getting Even More Creative

At Blackhat USA 2017, Itzik Kotler and Amit Klein (SafeBreach Labs) illustrated a technique that abuses "cloud-enabled" AV engines to perform successful data exfiltration in environments with highly restricted outbound filtering. The attack goes as follows:

1. The attacker obtains a foothold in the network with a malware sample that evades AV detection.
2. The attacker walks through the kill chain and successfully collects interesting data.
3. The data is collected and embedded in a known malware sample, after which it is written to disk.
4. The known sample is spotted by the AV engine and sent to a cloud sandbox for further analysis.
5. The sample is executed in the cloud sandbox of the AV vendor.
6. Upon execution, the sample sends out the exfiltrated data to the attacker.

In their Proof of Concept, exfiltration was successful for several mainstream AV vendors (including the likes of ESET, Kaspersky, Avira...). Although we recognize this is a bit of an "exotic" scenario, it shows how difficult it can be to prevent data exfiltration in your environment.

Slides can be found here: https://www.blackhat.com/docs/us-17/thursday/us-17-Kotler-The-Adventures-Of-Av-And-The-Leaky-Sandbox.pdf

Dedicated, persevering attackers will usually find a way to perform data exfiltration. If they have the capability to infiltrate your network successfully and gather data, they are bound to find a way to find a data exfiltration path as well.

This does not mean we should make it easy for them...

Prevention can be done by blocking the most obvious paths for network data exfiltration, like file hosting services. This can, however, be unacceptable in some organizations because of the negative business impact!

Step 4 – Preventing Network Exfiltration

We presented many ways to exfiltrate data and to abuse services to facilitate data exfiltration.

It might be depressing how many ways there are to exfiltrate data, but this is the grim reality we are facing: If your corporate network is connected to the internet, it is impossible to completely prevent data exfiltration. Our corporate network relies too much on diverse protocols and services, that it is impossible to properly prevent abuse on all protocols and services.

Dedicated, persevering attackers will find a way to exfiltrate data. If they have come so far to infiltrate your network successfully and gather data, they are bound to find a way to exfiltrate it.

Prevention can be done by blocking the most obvious paths for network data exfiltration, like file hosting services. But blocking all possible ways to exfiltrate data is an impossible task unless you are in the strictest network environment where internet access is virtually non-existent.

The only thing we can do, for prevention, is to make life harder for our attackers by blocking easy methods of data exfiltration. But even this can be unacceptable in some organizations, because of the negative business impact it can have.

While prevention of data exfiltration is extremely hard, detection of data exfiltration offers us a bit more hope. We recognize two main detection methods:

| Exfiltration detection system: Data Loss Prevention | Signature-based detection: Detect confidential data markers like antivirus or IDS |
| | Behavior-based detection: Detect abnormal patterns in network traffic, like large uploads |
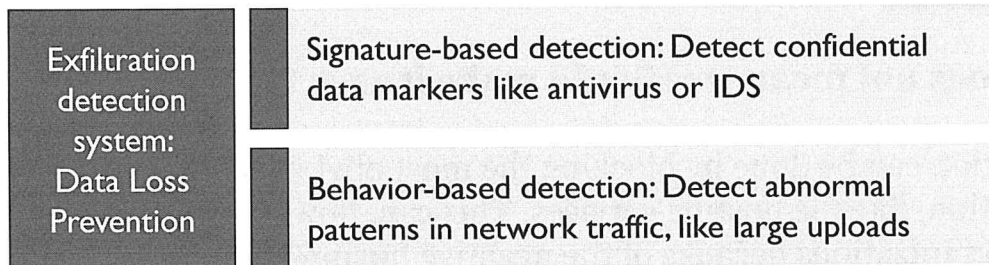
Step 4 – Detecting Network Exfiltration
While prevention of data exfiltration is extremely hard, detection of data exfiltration offers us a bit more hope. There are essentially 2 methods that data exfiltration detection systems use to detect possible data leakage. A data exfiltration detection system is known as a Data Loss Prevention (DLP) solution.

The 2 methods they can employ are:

- Signature-based detection
- Behavior-based detection

DLP solutions are put in place at the perimeter of the corporate network, where they can observe network traffic leaving the corporate network to the internet. With signature-based detection, DLP solutions will detect exfiltration of confidential data via watermarks, similar in the way antivirus and IDS work to detect malicious activity based on signatures.

With behavior-based detection, DLP solutions will detect exfiltration of confidential data by observing abnormal network traffic patterns. This is in large part based on the volume of the data.

Signature-based detection will look for special markers. For example, we can mark all confidential documents with a marker: Secret599. When data is leaked, the marker can be detected by DLP software or even IDS.

```
▼ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4d30 [correct]
    [Checksum Status: Good]
    Identifier (BE): 48950 (0xbf36)
    Identifier (LE): 14015 (0x36bf)
    Sequence number (BE): 8 (0x0008)
    Sequence number (LE): 2048 (0x0800)
    [Response frame: 6]
    Timestamp from icmp data: Jul 31, 2017 11:28:00.387564000 CEST
    [Timestamp from icmp data (relative): 0.000073000 seconds]
  ▼ Data (48 bytes)
        Data: 53656372657453454543353939536563726574534543353939...
        [Length: 48]

0000  c4 ea 1d 4d 02 b8 98 01  a7 9f a6 71 08 00 45 00   ...M.... ...q..E.
0010  00 54 15 a5 00 00 40 01  40 9b 0a 0a 0a 50 08 08   .T....@. @....P..
0020  08 08 08 00 4d 30 bf 36  00 08 59 7e 20 00 00 05   ....M0.6 ..Y~. ..
0030  e9 ec 53 65 63 72 65 74  53 45 43 35 39 39 53 65   ..Secret SEC599Se
0040  63 72 65 74 53 45 43 35  39 39 53 65 63 72 65 74   cretSEC5 99Secret
0050  53 45 43 35 39 39 53 65  63 72 65 74 53 45 43 35   SEC599Se cretSEC5
0060  39 39                                              99
```

Classification marker as seen in network traffic.
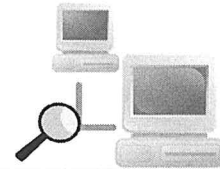
Bypassing this detection can be done with compression or encryption, for example.

Step 4 – Detecting Network Exfiltration – Signature-Based Detection
DLP solutions that are based on signatures will look for special markers (watermark) inside network traffic.

This implies that all data that is confidential must be:

1. Classified according to the appropriate level.
2. Modified to include a watermark that indicates it as classified data.

Take, for example, a confidential document that we mark with watermark Secret599 by adding this watermark as metadata to the document.

When this document is exfiltrated, the DLP software will detect network traffic leaving the corporate network that has a byte pattern corresponding to Secret599. This will raise an alert (or can even block the network transfer, depending on the DLP solution and the corporate network architecture).

It is clear the markers must be selected that are unique, and only to be found in classified data. Otherwise too many false positive alerts will be generated.

A simple detection like this can also be an in-house solution with an IDS and some simple rules.

The DLP solution must be able to inspect traffic in different encodings, and also support various compression methods.

Because bypassing detection would otherwise be trivial by compressing the data (or using another encoding that obfuscated the markers). When attackers encrypt the data to exfiltrate, they could be able to bypass signature-based DLP solutions.

Behavior-based detection will look for abnormal network patterns on data leaving the company. The volume of the data that leaves the network is an important indicator.

Volume user1
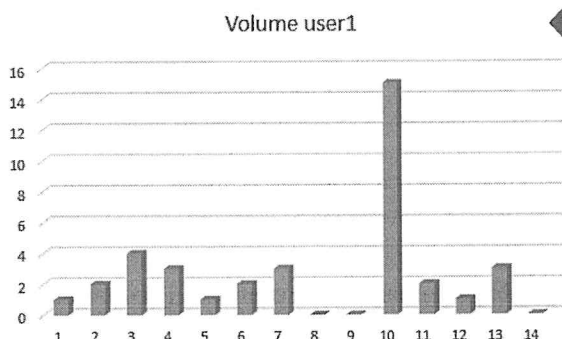
Looking at the data volume per user and per destination can reveal unexpected data transfers. Large, unexpected data transfer can indicate data exfiltration.

Especially at the start, this sort of behavior analysis will produce many false positives; the system must be tuned over time: Automatically (self-learning) or manually (configuring exceptions).

Step 4 – Detecting Network Exfiltration – Behavior-Based Detection (1)

DLP solutions that are based on behavior do not need watermarked documents like signature-based DLP solutions (actually, DLP solutions will offer several detection techniques).

What these DLP solutions look for is abnormal network patterns for data leaving the corporate network. These solutions can be based on learning patterns, where they observe the corporate network traffic for a period of time, assuming that network traffic during that period is normal (aka learning mode), and then are switched over to detection mode where they compare the observed traffic with historical patterns and alert on deviations.

The volume of the data that leaves the corporate network is an important indicator for behavior-based DLP solutions. They will measure the amount of data per user and per destination, and alert on unexpectedly large data transfers. This method, of course, can only be successful if the exfiltrated data is indeed large enough to deviate from the norm. If it is small, this method will not detect it.

There will also be many false positives because large data transfers that deviate from the norm do not necessarily imply malicious intent. Some corporate network activities also have cycles over long periods, like business cycles that occur monthly, tri-monthly or yearly, for example. These business cycles can be linked to large data transfers that are most likely not part of the baseline established during the learning phase.

As we discussed, behavior analysis will produce many false positive alerts, especially at the beginning of the adoption of the solution.

Another method is to tune the system over time:

- automatically
- manually

Automatically involves self-learning (machine learning), where the solution learns itself to convert false positives into true negatives.

Manually involves network administrator intervention: Configuring exceptions to avoid false positives in the future (like a whitelisting operation).

If we cannot determine from the analysis data what happened, we will ultimately have to ask the involved user(s). This can be a difficult task, as non-technical users can experience difficulty to correlate their business activities with network traffic.

Step 4 – Detecting Network Exfiltration – Behavior-Based Detection (2)

A second behavior-based technique is related to the "covert channels" we discussed before. Protocols like DNS provide an interesting option for adversaries to include covert payloads. This would typically, however, rely on "strange" DNS traffic:

- High volume of TXT records
- High entropy in DNS names
- BASE64 encoding in DNS names
- Long DNS names

- High volume of DNS requests from 1 source
- High volume of DNS requests to 1 domain
- High number of hosts resolved for 1 domain
- ...

For different protocols, we can define "anomalies" that we typically wouldn't see much in a corporate environment. Using logging or packet capturing, we can dashboard this traffic and spot anomalies!
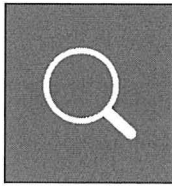
The Bro Exfil detection framework is a compilation of several scripts that work together to detect file uploads in TCP, UDP, and ICMP connections. The Exfil framework has the ability to detect file uploads in most sessions, including encrypted TCP traffic such as HTTPS, SFTP and SCP. The Bro Exfil detection framework detects anomalies by looking for sustained bursts in outbound traffic

The Bro Exfil detection framework contains three main parts:

- A script that defines the business hours.
- A script that defines the network segments to monitor.
- The main script that performs session tracking and performs the calculation of the normal network traffic baseline and burst detection.

Step 4 – Detecting Network Exfiltration – Behavior-Based Detection (3)

The Bro Exfil detection framework is a compilation of several scripts that work together to detect file uploads in TCP connections. The Exfil framework has the ability to detect file uploads in most TCP sessions, including encrypted traffic such as HTTPS, SFTP and SCP. The Bro Exfil detection framework detects anomalies by looking for sustained bursts in outbound traffic.

The Bro Exfil detection framework contains three main parts:

- A script that defines the business hours.
- A script that defines the network segments to monitor.
- The main script that performs session tracking and performs the calculation of the normal network traffic baseline and burst detection.

In order to deploy the Bro Exfil detection framework in your Bro instance, two steps need to be taken:

- Download the scripts to the "site" folder of your Bro instance.
- Update your local.bro file to include the following.
  - Enable Exfil framework by adding .
    - @load bro-scripts/exfil-detection-framework.
  - Define networks to be monitored.
    - redef Exfil::watched_subnets_conn = [x.x.x.x/x, y.y.y.y/y]
  - Define the business hours of your network (in a 24h time format).
    - redef Exfil::hours = [$start_time=x, $end_time=y]

The script can be found here:
https://github.com/reservoirlabs/bro-scripts/tree/master/exfil-detection-framework

Side note: A tale of printer dots:

- Reality Leigh Winner was arrested for leaking top-secret NSA reports detailing Russian hacking before the 2016 elections.
- The report was leaked to The Intercept via a printout, which was reproduced in the online article.
- It is believed that the publication of this printout lead to the arrest of Reality.
- The printout contains special printer dots that identify the printer.
- Reality was one of few people that used this printer.

Step 4 – Physical Exfiltration – A Tale of Printer Dots...

When it comes to physical exfiltration, we want to mention that several interesting methods have been devised to detect or thwart this. These are typically the kind of efforts one would see in military or intelligence operations. The example we want to mention here takes printers as an exfiltration device. Secret information can be printed out, and the printed paper sheets can just be carried out of the building or mailed.

To identify the source of printed documents, a method has been elaborated that involves small, almost invisible dots that are surreptitiously printed on a paper when a document is printed. These dots uniquely identify a printer.

Such a case made the news recently:

- Reality Leigh Winner was arrested for leaking top-secret NSA reports detailing Russian hacking before the 2016 elections.
- The report was leaked to The Intercept via a printout, which was reproduced in the online article.
- It is believed that the publication of this printout led to the arrest of Reality.
- The printout contains special printer dots that identify the printer.
- Reality was one of few people that used this printer.

https://www.eff.org/deeplinks/2017/06/printer-tracking-dots-back-news

To summarize, it will be clear from the examples we gave that depending on your corporate environment, an attacker can have virtually unlimited options to perform data exfiltration.

Although a possible option, physical data exfiltration is typically not the preferred method used by adversaries.

Due to the many options available, prevention can be a (very) daunting task. Detection can be based on patterns and behavior, but both have their limitations.

By mimicking data transfers that are considered "normal" in the victim environment, attackers can easily remain under the radar.

Due to the limited defensive options, organizations should try stopping the attack before it reaches this stage!

Data Exfiltration – Summary
To summarize, it will be clear from the examples we gave that depending on your corporate environment, an attacker can have virtually unlimited options to exfiltrate data.

Although a possible option, physical data exfiltration is typically not the preferred method used by adversaries. Network exfiltration is by far the most commonly used exfiltration method. Typically, the adversary has already set up a C&C channel, which it could (partially) reuse for exfiltration.

The physical environment and corporate network environment need to be locked down to limit the possibilities of data exfiltration, but this cannot be eliminated, as long as corporations need an open work environment and internet access.

Due to the many options available, prevention can be a (very) daunting task. Detection can be based on patterns and behavior, but both have their limitations!

By mimicking data transfers that are considered "normal" in the victim environment, attackers can easily remain under the radar!

Due to the limited defensive options, organizations should try stopping the attack before it reaches this stage...

Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- Day 4: Lateral Movement
- **Day 5: Action on Objectives, Threat Hunting & Incident Response**
- Day 6: APT Defender Capstone

SEC599.5

Domain dominance
- Dominating the AD - Basic strategies
- Golden Ticket, Skeleton Key, DCSync and DCShadow
- Detecting domain dominance
- Exercise: Domain dominance

Data exfiltration
- Common exfiltration strategies
- Exercise: Detecting data exfiltration

Leveraging threat intelligence
- Defining threat intelligence
- Exercise: Leveraging threat intelligence with MISP & Loki

Threat Hunting & incident response
- Proactive threat hunting strategies
- Exercise: Hunting your environment using OSQuery
- incident response process
- Exercise: Finding malware using Volatility & YarGen

This page intentionally left blank.

Exercise: Detecting Data Exfiltration

Please refer to the workbook for further instructions on the exercise!

This page intentionally left blank.

Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- Day 4: Lateral Movement
- **Day 5: Action on Objectives, Threat Hunting & Incident Response**
- Day 6: APT Defender Capstone

SEC599.5

Domain dominance
- Dominating the AD - Basic strategies
- Golden Ticket, Skeleton Key, DCSync and DCShadow
- Detecting domain dominance
- Exercise: Domain dominance

Data exfiltration
- Common exfiltration strategies
- Exercise: Detecting data exfiltration

Leveraging threat intelligence
- Defining threat intelligence
- Exercise: Leveraging threat intelligence with MISP & Loki

Threat Hunting & incident response
- Proactive threat hunting strategies
- Exercise: Hunting your environment using OSQuery
- incident response process
- Exercise: Finding malware using Volatility & YarGen

This page intentionally left blank.

In order to understand what threat intelligence is all about, we first need to do a quick refresh on what a "threat" is. In Risk Management, a risk is typically defined as:

RISK = VULNERABILITY x IMPACT x THREAT

Expanding on this, we can state that a threat is established by evaluating the following components:

- **Capability**: The threat actor is capable of reaching his / her objectives.

- **Intent**: The threat actor is deliberately trying to attain his / her objectives.

- **Opportunity**: Certain conditions exist that could allow a threat actor to reach his / her objectives.

What Is Threat Intelligence? (1)
In order to understand what threat intelligence is all about, we first need to do a quick refresh on what a "threat" is. In Risk Management, a risk is typically defined as **RISK = VULNERABILITY x IMPACT x THREAT.** Theoretically, you could say that if there is no vulnerability, no impact or no threat, there is no risk. However, it is impossible to mitigate all vulnerabilities, all possible impact or all threats without breaking "mission statement" as an organization.

Expanding on this, we can state that a threat is established by evaluating the following components:

- Capability: The threat actor is capable of reaching his / her objectives.
- Intent: The threat actor is deliberately trying to attain his / her objectives..
- Opportunity: Certain conditions exist that could allow a threat actor to reach his / her objectives.

Some examples of using these terms:

- If a threat actor is not capable but has hostile intent and there is an opportunity, we could conclude the threat is insubstantial.
- If a threat actor has hostile intent, is capable, but there is no opportunity, we could conclude the threat is impending.
- If a threat actor is capable, there is an opportunity but there is no a hostile intent, we could conclude there is a potential threat that can materialize.

So, how do we define threat intelligence?

- There's a large number of threat intelligence definitions available
- In line with SANS' CTI methodology (*FOR578 – Cyber Threat Intelligence*), we will define threat intelligence as:

  "Analyzed information about the hostile intent, capability, and opportunity of an adversary."

- The focus on threat intelligence should be placed on the human element.
  ⇒ E.g. Malware **alone** should not be considered a threat!

What Is Threat Intelligence? (2)

In this chapter, we will define threat intelligence and provide some examples of threat intelligence and how it can be used to detect attacks and adversaries. We will first, however, define the concept of threat intelligence.

Please note that there's a large number of threat intelligence definitions available. According to Gartner, for example, 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."

In line with SANS' Cyber Threat Intelligence methodology (as defined in FOR578 – Cyber Threat Intelligence), we will define threat intelligence as:

"Analyzed information about the hostile intent, capability, and opportunity of an adversary."

Different types of definitions are acceptable, as long as the focus is placed on the human element. As an example, malware alone should not be considered a threat to your business. The use of malware by an adversary to compromise your IT environment and steal your crown jewels is, however, a threat.

It's important to make a distinction between different levels of threat intelligence. We chose to create three levels:

Strategic

Strategic threat intelligence includes information on changing risks (e.g. at senior leadership level, a change in business direction would result in an adapted threat landscape).

Operational

Attacker methodologies, tools and tactics. Often referred to as TTPs, these include typical "habits" of adversaries, without providing narrow Indicators of Compromise.

Tactical

At the lowest level, we have tactical threat intelligence, which is often limited to highly technical / narrow Indicators of Compromise of specific attacks / attack campaigns.

What Is Threat Intelligence? (3)

It's important to note that threat intelligence can exist at different levels of an organization. We should make a distinction between different levels of threat intelligence. For the purposes of our course, we will distinct the following levels:

- Strategic intelligence: Strategic threat intelligence includes information on changing risks (e.g. at senior leadership level, a change in business direction would result in an adapted threat landscape).

- Operational intelligence: Attacker methodologies, tools and tactics. Often referred to as TTPs, these include typical "habits" of adversaries, without providing narrow Indicators of Compromise.

- Tactical intelligence: At the lowest level, we have tactical threat intelligence, which is often limited to highly technical / narrow Indicators of Compromise of specific attacks / attack campaigns.

As this is a technical cyber security course, we will mostly discuss tactical and operational threat intelligence!

"Most of the threat intelligence feeds available on the market aren't intelligence at all; they're aggregated reports on malware and spam, rogue IP addresses, and vulnerabilities that can't be tied to a given environment."

Dave DeWalt
ex-CEO FireEye

This is an excellent example of too much IOC-focused intelligence:

- Highly technical IOCs (e.g. domain names, IPs...) are only useful for a short time.
- IOC lists often lack context: Threat actors, TTPs, industry...
- Don't focus on quantity... Instead, focus on quality!
- Many organizations are struggling to correctly USE threat intelligence...

We will touch upon a few of these concepts in this section of the course!

Problems with Threat Intelligence
According to Dave DeWalt, ex-CEO of FireEye:

"Most of the threat intelligence feeds available on the market aren't intelligence at all; they're aggregated reports on malware and spam, rogue IP addresses, and vulnerabilities that can't be tied to a given environment."

This is an excellent example of too much IOC-focused intelligence... There are a few common issues many organizations appear to be facing:

- First of all, highly technical IOCs (e.g. domain names, IPs...) are only useful for a short time. These IOCs are easy to change by adversaries (see next slides), which makes their use highly limited.
- IOC lists often lack context: Threat actors, TTPs, industry... Some so-called "intelligence feeds" only consist of a list of raw Indicators of Compromise, which will result in very little operational value.
- Don't focus on quantity... Instead, focus on quality! Take efforts to collect threat intelligence that is relevant to your organization and the threats you are facing!
- Even if they are collecting the right types of threat intelligence, many organizations are struggling to correctly USE and operationalize it...

We will touch upon a few of these concepts in this section of the course!

Pyramid of Pain (top to bottom):

- TTPs — Tough!
- Tools — Challenging
- Network/Host Artifacts — Annoying
- Domain Names — Simple
- IP Address — Easy
- Hash Values — Trivial

Source: David J. Bianco, personal blog

Tactical & Operational CTI – Introducing David Bianco's Pyramid of Pain

David Bianco produced an illustration of the various types of indicators used in real-time detection and threat hunting, and how effective they are at combating advanced adversaries. David calls this illustration the "Pyramid of Pain."

It represents the amount of "pain" you can inflict on advanced attackers by using a certain type of indicator and denying them the use of attacks that can be detected by these indicator types.

At the base, the pyramid starts with indicators that are trivial to bypass detection by the attackers, and thus inflict a trivial amount of pain to the attackers.

At the top, the pyramid ends with indicators that are difficult to bypass detection by the attackers, and thus inflict a though amount of pain to the attackers.

The pyramid has 6 types of indicators:

- TTPs
- Tools
- Network / Host Artifacts
- Domain Names
- IP Addresses
- Hash Values

Hash values

"This is sample text"
Hash: 636351fcb9197f5e75b845628508bbb1

"This is sample text!"
Hash: 7f5d61b32b03df736c39ec06b2597661

Source: David J. Bianco, personal blog

David Bianco's Pyramid of Pain – Hash Values
We start at the bottom of the pyramid with hash values.

Hash values are cryptographic hash values like the MD5 values of malicious executables used by attackers.

When we rely on hash values to detect adversaries, we inflict a trivial amount of pain to the adversaries: Changing the hash value of a malicious executable is trivial. Changing just a single bit is sufficient to change the hash completely.

Attackers can use specialized tools to generate variants of a malicious executable that are functionally identical but are different at the byte level and thus have different hash values.

IP Addresses

IP addresses used for
- Command & Control infrastructure
- Data exfiltration address

Source: David J. Bianco, personal blog

David Bianco's Pyramid of Pain – IP Addresses
Next is detection based on IP addresses.

IP addresses will be used in network communications for command & control servers, and for data exfiltration servers, for example.

Changing IP addresses for attackers is easy. Detections based on IP addresses can be easily bypassed by changing IP addresses. If attackers rely on DNS to resolve domain names to IP addresses, changing IP addresses is as simple as performing a DNS update.

David Bianco's Pyramid of Pain – Domain Names
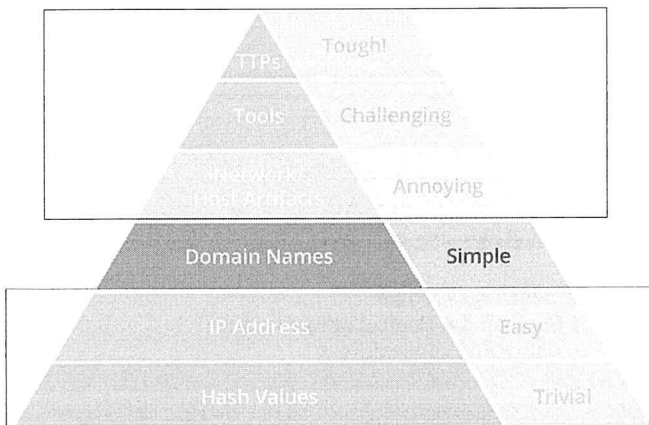
Domain names

Domain names and subdomain names
- Evil.com
- This.is.evil.com

Source: David J. Bianco, personal blog

David Bianco's Pyramid of Pain – Domain Names
Indicators based on domain names are better quality indicators than indicators based on IP addresses.

Domains can be new domain names under a Top-Level Domain (TLD) like .com, for example, evil.com. Or domains can be subdomains under existing domains, like this.is.evil.com for example.

When we use domain names as indicators, our adversaries can simply bypass this detection by changing domain names.

Domain names can be easily obtained and are not expensive.

Depending on the infrastructure used by attackers, changing a domain name can be as simple as changing an entry in a configuration file, or recompiling an executable.

Network / Host Artifacts

- Specific user agents
- URI patterns
- SMTP mailers
- Registry key values
- Files and directory names

Source: David J. Bianco, personal blog

David Bianco's Pyramid of Pain – Network and Host Artifacts

When we detect the activity of our adversaries based on network and host artifacts, we inflict a bit more pain. Example of network and host artifacts are:

- Specific user agents
- URI patterns
- SMTP mailers
- Registry key values
- Files and directory names

Bypassing detection by indicators of network and host artifacts is a bit more annoying to the attackers, as it implies that they have to change patterns used by their tools and malware.

This implies making (small) changes to their code base.
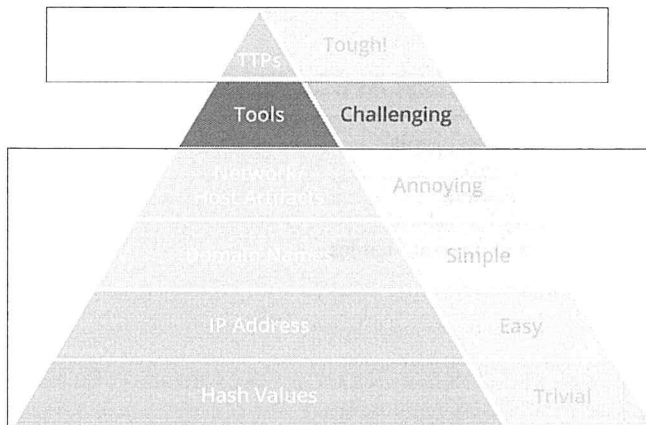
Tools

Things attackers bring with them
- Password crackers
- Post-compromise utilities
- Exploits
- Utilities attackers use to create malicious documents

Source: David J. Bianco, personal blog

David Bianco's Pyramid of Pain – Tools

When we detect advanced adversaries based on the tools they use, we start to inflict major pain. Examples of tools used by (advanced) adversaries are:

- Password crackers
- Post-compromise utilities
- Exploits
- Utilities attackers use to create malicious documents

When we detect the use of these tools, we prevent them from using the tools again in our corporate environment, which makes it challenging for the attackers. Detecting tools here is not based on hash values or User Agent Strings, for example, but on the patterns of events that these tools generate when they are used on a computer system or inside a network.

David Bianco's Pyramid of Pain – TTPs
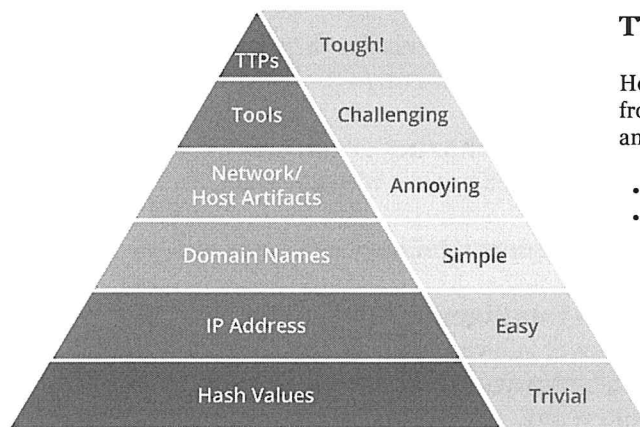
Source: David J. Bianco, personal blog

TTPs: Tactics, Techniques, & Procedures

How the adversary goes about accomplishing their mission, from reconnaissance all the way through data exfiltration and at every step in between:

- Spear-phishing with a trojanized PDF file.
- Dumping cached authentication credentials and reusing them in Pass-the-Hash attacks.

David Bianco's Pyramid of Pain – TTPs

The highest level of pain we can inflict is at the top of the pain pyramid: When we can detect TTPs, we really make it through to our advanced adversaries to change TTPs. TTPs are Tactics, Techniques & Procedures.

How the adversary goes about accomplishing their mission, from reconnaissance all the way through data exfiltration and at every step in between.

Examples:

- Spear-phishing with a trojanized PDF file.
- Dumping cached authentication credentials and reusing them in Pass-the-Hash attacks.
- …

Option 1: Buy a commercial feed.
- PRO: Not a lot of effort involved.
- CON: Often not very tailored intelligence feeds / collection of IOCs.
- CON: High-value feeds are rather expensive ($ / €).

MISP
Threat Sharing

Option 2: Obtain intelligence from sharing communities.
- PRO: More tailored and valuable intelligence (with context).
- CON: Effort to be done
- CON: Only works well if you also share intelligence.

Option 3: Generate your own intelligence
- PRO: Highly valuable intelligence relevant to your organization.
- PRO: You learn a lot about adversaries and your environment.
- CON: Requires expertise and effort.

In an ideal scenario, you combine the three options!

How Do You Obtain Threat Intelligence?
There are several methods to obtain threat intelligence that can be used to help defend your corporate environment. We will list some of the most commonly available options:

Option 1: Buy a commercial threat intelligence feed
- PRO: This does not require a lot of effort from your side; you basically "outsource" the problem
- CON: Many threat intelligence vendors are focused on the collection of hard IOCs and don't focus enough on providing tailored intelligence feeds that provide the required context to correctly operationalize threat intelligence;
- CON: The cost of purchasing a high-value threat intelligence feed can be rather high

Option 2: Obtain intelligence from sharing communities
- PRO: You typically receive more tailored and contextualized threat intelligence, as you share with industry peers that are facing similar threats;
- CON: There is quite a bit of effort to be done: Get out there, go and talk to people…
- CON: In the long run, this type of "obtaining intelligence" only works when you also share intelligence yourself (which again requires effort).

Option 3: Generate your own intelligence
- PRO: Intelligence you generate yourself will have a lot of value for your own organization, as it's been generated from observations that occurred inside your environment;
- PRO: In the process of creating your own intelligence, you will learn a lot about your adversaries AND your own environment;
- CON: Generating your own intelligence requires the most effort from all three options. Furthermore, it requires a certain level of maturity and expertise to do well.

In an ideal scenario, you combine all three options!

Some Free Threat Intelligence Sources – AlienVault OTX
In the screenshot, we can see what AlienVault OTX (Open Threat Exchange) looks like.

Once you have registered a free account, you receive access to different threat intelligence information. Although still very much IOC-focused, you can define "pulses" that help you focus on data relevant to you!

You can find AlienVault OTX at https://otx.alienvault.com

Some Free Threat Intelligence Sources – ThreatCrowd
ThreatCrowd takes a different approach.
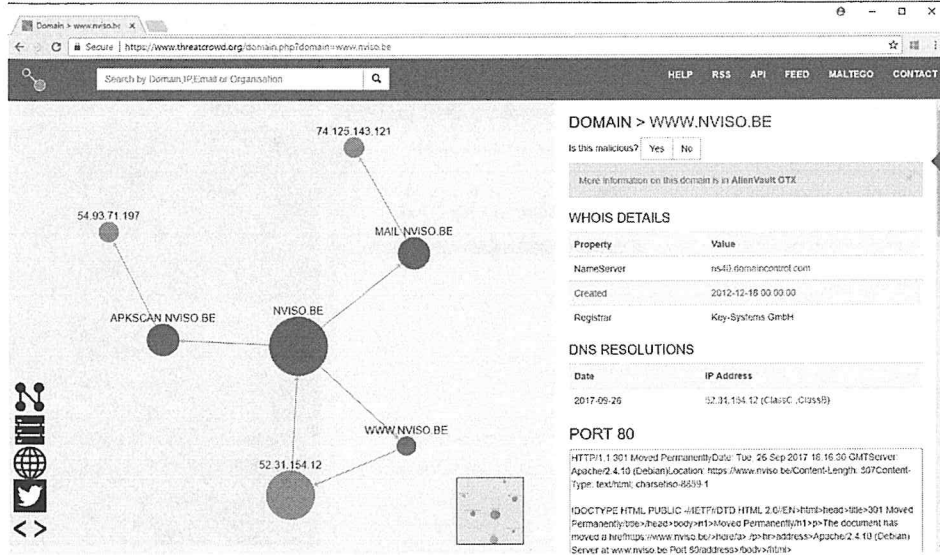
It features a search engine where you can enter your query (e.g. a hostname, website, the name of a malware family, a hash…) and it will provide a schematic overview of information it has linked to your search query!

You can find ThreatCrowd here: https://www.threatcrowd.org/

The Malware Information Sharing Platform (MISP) is a free, open-source project providing a Linux-based application with a web GUI.

MISP instances can be connected to others, so information can be exchanged in a controlled fashion.

MISP users have fine-grained options to decide what information is shared with whom.

Sharing Threat Intelligence – Introducing MISP
We will discuss an open-source threat intelligence sharing platform that is based on IOCs.

The Malware Information Sharing Platform (MISP) is a free, open-source project providing a Linux-based application with a web GUI.

MISP is not only free, open-source software, but it is also designed to share free, open-source threat intelligence. Although it can be used to collect IOCs without sharing.

MISP instances are used to create events and associate IOCs with these events. An event, for example, can be created for the WannaCry ransomware, and an IOC would be the MD5 hash of the WannaCry sample.

When instances are used in a stand-alone fashion, without connecting to other MISP instances (of other organizations), threat intelligence is not shared.

The power of MISP, however, lies in its information-sharing model: MISP instances can be connected to each other via a subscription-based model (this is a technical term, not a commercial term). Events and IOCs created by one organization are then shared through all organizations that connect to that MISP instance.

Of course, for each MISP, it is possible to flag events and IOCs as non-sharable because they contain confidential data that could, for example, compromise the source of this intelligence.
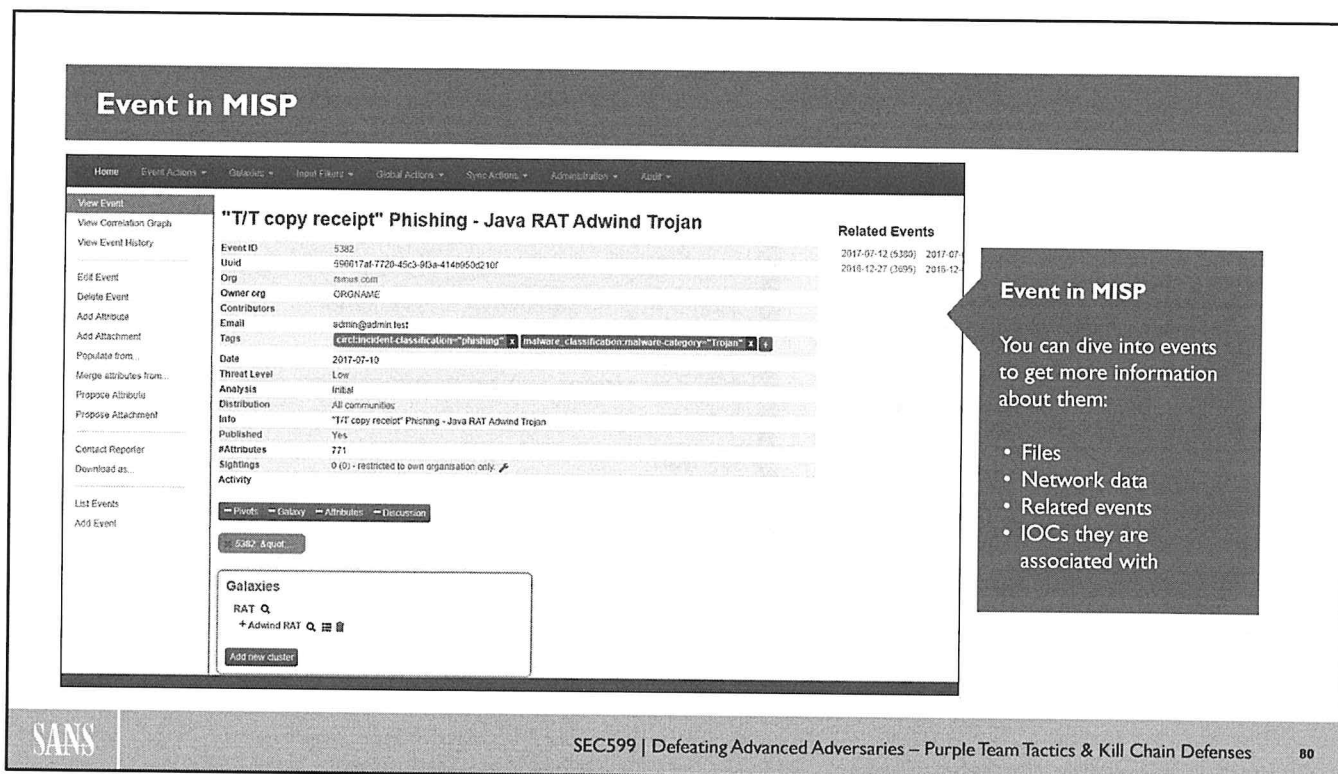
Event in MISP

This is a screenshot of an event in MISP. It is a modern web-based GUI that resembles full applications.

The event pertains to a phishing event for the Adwind Trojan: An attempt at infection with the Adwind Trojan via unwanted email.

The event has the tags "phishing" and "Trojan".

On the right-hand side, we can see links to related events: This can help us look at similar events and discover common IOCs or modus operandi.

This event has 771 attributes that form the IOCs.

One can browse through the attributes to obtain a list of different types of IOCs, like:

- Files
- Network data
- Related events
- IOCs they are associated with

Home	Event Actions ▾	Galaxies ▾	Input Filters ▾	Global Actions ▾				MISP Dstevens ✉ Log out	
✔	🔴	5414	tlp:white	12	2017-07-21	Low	Completed	OSINT - Linux.Bew: un backdoor para el minado de Bitcoin	All
✔	🔴	5413	tlp:white	20	2017-07-21	Low	Completed	OSINT - Rurktar - Spyware under Construction	All

This MISP home view is the overview of events.

We can see two events, with numbers 5414 and 5413.

Next to the number, we have the tag (tlp:white), and on the right the info.

MISP Home Page

The screenshot above is the home view of the MISP GUI interface, which is presented to a user once she has logged on.

This home view presents an overview of the events in the MISP database.

On screen, we can see two events, with numbers 5414 and 5413. They have different columns with information pertaining to these events.

On the left for example (second column), we see the symbol of the organization (MISP instance). The red symbol here is the CIRCL organization, the national CIRT of Luxembourg, which is the main proponent behind MISP.

Then we have the number of the event and the tag (tlp:white).

The numbers 12 and 20 represent the number of attributes (IOCs) for each event.

We have the date the event was created in MISP, the threat level (low) and the status of the analysis. Completed here means that the evidence gathered for the event is complete and fully stored in MISP.

Finally, on the right, we have the info for the event. This is a description for human consumption.

In the next slide, we will see the details for a particular event.

Reviewing an Event in MISP

This is the screen that is presented when we click on a particular event in MISP. It gives an overview of the properties and metadata of an event.

The event displayed here (ID 5413) is for the Rurktar spyware. The fact that the description starts with OSINT indicates that this information was obtained from OSINT sources: Open Source INTelligence. The organization that created this event (CIRCL) has obtained the information from an open source. Usually, the URL of the source, a website article, for example, is provided as an attribute (this is not an IOC).

The event is published (this means that it can be shared), the analysis is complete, and it has 20 attributes.

Tag tlp:white indicates that the information may be sharing under the Traffic Light Protocol (TLP) white status.

TLP has 4 colors as status:

- Red
- Amber
- Green
- White

Red is the highest confidentiality, white the lowest.

Information marked as white may be publicly disseminated outside the organization that provides it, without restriction.

Red marked information may not be shared outside the organization.

			b5		
2017-07-21	Payload delivery	sha256	89110710eddd0da23ea206a6047c252bf1e16a2d1957729973d77a58219e	MSIL.Backdoor.Rurktar.A	
			614b		
2017-07-21	Payload delivery	sha1	a97718c9be9f903c5b797643e458c207ae1c8bee	MSIL.Backdoor.Rurktar.A - Xchecked	
				618908e3d368301a323ee8ae7df9	
2017-07-21	Payload delivery	md5	921b1b1d2f7a228e753fc43da1b3169c	MSIL.Backdoor.Rurktar.A - Xchecke	
				618908e3d368301a323ee8ae7df867db	
2017-07-21	Payload delivery	sha256	54f25a6820b9a0e3fc26bdf4599e7db695ef7aefb7dcefaa2c2581bb58426a4	MSIL.Backdoor.Rurktar.A	
			0		
2017-07-21	Payload delivery	sha1	77ce9ccf740d394e065ece52fc0f282686e672c1	MSIL.Backdoor.Rurktar.A - Xchecked via	
				89110710eddd0da23ea206a6047c252bf	
2017-07-21	Payload delivery	md5	adcb921fa79845abbe3011237f227773	MSIL.Backdoor.Rurktar.A - Xchecked via	
				89110710eddd0da23ea206a6047c252bf	
2017-07-21	Payload delivery	sha256	b4b75bda475ea58f2a5cf3329e311a70fa56745ba6cb2785523fa53139d4e3	MSIL.Backdoor.Rurktar.A	
			7f		
2017-07-21	Payload delivery	sha1	8ee7df7a432b9d1c12ac2dcc85a39686e8ade31e	MSIL.Backdoor.Rurktar.A - Xchecked via	
				54f25a6820b9a0e3fc26bdf4599e7db695	
2017-07-21	Payload delivery	md5	9076a243f64f33841b27643ca71cca1c	MSIL.Backdoor.Rurktar.A - Xchecked via	
				54f25a6820b9a0e3fc26bdf4599e7db695	
2017-07-21	Payload delivery	sha1	d73470779ebef52d5b81cf9bfe3d28465b2879a3	MSIL.Backdoor.Rurktar.A - Xchecked via	

Could not locate the PGP/GPG public key. Powered by MISP 2.4.78

This MISP view presents the attributes of the event we discussed in the previous slide (event ID 5413).

The events listed here are cryptographic hashes (md5, sha1 and sha256) of malware samples.

Reviewing Attributes Linked to an Event (1)

This MISP view presents the attributes of the event we discussed in the previous slide (event ID 5413).

The first column is the date of the event.

Then we have the type of attribute: In this example, the type is "payload delivery." This is used for malware samples that deliver the payload.

Malware samples can be identified with cryptographic hashes. Here, we have the md5, sha1 and sha256 values for various samples.

These are actionable IOCs: We can use these hashes to create ClamAV signatures or YARA rules to detect these hashes or to hunt for them in our environment.

The fact that we have several types of cryptographic hashes for the same sample is good: This allows us to use various detection tools. Sometimes, just one type of hash is published, and it may not be actionable with the tool that you use.

For example, if only a SHA256 hash is published but your scanning tool requires MD5 hashes, the SHA256 hash is of no value to you.

Date	Org	Category	Type	Value		Tags	Comment	Related Events	Feed hits	IDS	Distributi
		Filters: All (26) File Network Financial Proposal Correlation Warnings Show context fields									
2014-03-27		Antivirus detection	text	ZxShell				463		No	All
2014-03-27		Antivirus detection	text	Trojan:Win32/Sakurel.A						No	All
2016-08-01		External analysis	attachment	The_French_Connection_French Aerospace-Focused_CVE-2014-0322.pdf			https://www.crowdstrike.com/blog/french-connection-french-aerospace-focused-cve-2014-0322-attack-shares-similarities-2012/index.html			No	Inherit
2016-08-01		External analysis	link	http://www.crowdstrike.com/blog/french-connection-french-aerospace-focused-cve-2014-0322-attack-shares-similarities-2012/index.html						No	All
2014-03-27		Network activity	hostname	savmpet.com				1168 2104		Yes	All
2014-03-27		Network activity	hostname	oa.ameteksen.com				1168 1228 2098 2104		Yes	All
2014-03-27		Network activity	hostname	secure.safran-group.com						Yes	All

This is also a view of attributes but scrolled to the right.

This shows different columns, and we want to draw your attention to the IDS column.

Reviewing Attributes Linked to an Event (2)

This is also a view of attributes but scrolled to the right.

The category column of an attribute indicates its source. For example, category "external analysis" indicates that the attribute is based on external analysis, not performed by the organization that created the event in MISP.

On the right-hand side of this screenshot, we want to draw your attention to the IDS column.

The value for IDS can be Yes or No, and it indicates if this attribute is suitable for use by an IDS or not (or other detection systems).

For example, attributes that are IP addresses or domain names would often be marked with IDS value, Yes, to indicate that these values can be used to detect malicious activity for this event or can be used in a blacklist for example to prevent malicious activity.

We want to remark that we have observed that it can happen that low-quality IOCs are marked as IDS Yes, and that will result in false positive detections. For example, when malware is automatically analyzed inside a sandbox, connections to DNS servers will often be observed. Like Google's DNS server with IPv4 address 8.8.8.8.

We have seen events where this address (8.8.8.8) was included as an IDS attribute, which would lead to many false positive alerts, as DNS requests to 8.8.8.8 are common in a network and certainly no indication of malicious activity.

Adding New Events in MISP

Finally, we conclude this introduction to MISP with the dialog used to add a new event.

Remark that several initial fields are populated by default.

The date field is populated automatically.

The threat level is set to High by default.

The status of the analysis is set to Initial by default: This means that the analysis has started but is not yet completed. Events with analysis status Initial can be shared, too. Completed analysis is not required for sharing.

An important property is the Distribution field. By default, this is set to "This community only". This means that this event will not be shared with MISP instances of other communities.

To share information with other communities, the value for Distribution must be "All communities" or a limited set of communities.

Let's say we have intelligence; how can we now "operationalize" intelligence? Here are a few excellent use cases for threat intelligence:

Immediately use fresh IOCs in real-time prevention and detection tools (Firewalls, IDS, IPS, SIEM…). This is what many organizations already do in an automated fashion.

Use old IOCs to cross-check archived logs for potential hits and thus signs of previous compromises (which could still be active today!)

Use operational and tactical threat intelligence (e.g. TTPs) to create new hypotheses for threat hunting (which we will discuss in the next chapter).

Use strategic intelligence to improve your cyber security strategy and steer future investment decisions.

Operationalizing Threat Intelligence

Say we have successfully obtained intelligence… This doesn't mean we are not already leveraging the intelligence in order to improve our overall cyber security posture… How can we now "operationalize" intelligence? Here are a few excellent use cases for threat intelligence:

- Immediately use fresh IOCs in real-time prevention and detection tools (Firewalls, IDS, IPS, SIEM…). This is what many organizations already do in an automated fashion. Many firewalls and SIEMs, for example, come with a built-in "intelligence feed", which is typically just a long list of IOCs.
- Use old IOCs to cross-check archived logs for potential hits and thus signs of previous compromises (which could still be active today!). In many organizations, this is one of the main reasons for incident discovery.
- Use operational and tactical threat intelligence (e.g. TTPs) to create new hypotheses for threat hunting (which we will discuss in the next chapter). The idea is that we can understand new TTPs being used by adversaries and use them to define new potential intrusion methods we can check in our environment.
- Finally, use strategic intelligence to improve your cyber security strategy and steer future investment decisions.

As this is a technical course, we will now focus on leveraging technical Indicators of Compromise. An interesting tool for this purpose is the freely available "Loki" IOC scanner (by BSK Consulting / Nextron Systems).

- An IOC scanner is a tool that will use IOCs to scan computer systems' resources (filesystem, registry, memory, ...)
- There are several open-source and commercial IOC scanners on the market, some are even based on scripting languages such as PowerShell!
- Loki was developed by Florian Roth (BSK Consulting / Nextron Systems) and is the free variant of a commercial tool called "Thor".
- Spark is the latest variant of the tool and has support for multiple OSs (it's also free but requires registration!)
- Loki, Spark and Thor use an "intelligent" scoring system (next slide).

Operationalizing Threat Intelligence – Using Loki IOC Scanner

As this is a technical course, we will now focus on leveraging technical Indicators of Compromise. An interesting tool for this purpose is the freely available "Loki" IOC scanner (by BSK Consulting / Nextron Systems).

An IOC scanner is a tool that will use IOCs to scan computer systems' resources (filesystem, registry, memory, ...). It's important to note that there are several open-source and commercial IOC scanners on the market, some are even based on scripting languages such as PowerShell!

Loki was developed by Florian Roth (BSK Consulting / Nextron Systems) and is the free variant of BSK's commercial tool called "Thor". Florian Roth is a known security researcher who has contributed heavily to the community by developing a large number of YARA rules.

Spark is the latest variant of the tool and has support for multiple OSs (it's also free but requires registration!)

Loki, Spark and Thor don't only rely on "hard" IOC matching; they also feature an "intelligent" scoring system (next slide).

Thor and Loki's Scoring Mechanism

Checks		Datei 2: C:\Windows\System32\d.sys		Datei 1: C:\Windows\temp.exe		Datei 3: C:\TEMP\wce.exe	
Service Check		is service = +10			·		·
Process Check			·		·		·
Extension			·	.exe = +3	match = -3	.exe = +3	match = -3
Type	Combination	EXE = +3		EXE = +3		EXE = +3	
File Name Characteristics		[a-z].sys = +15		\temp.exe = +20		wce.(exe|dll) = +40	
Location		\System32 = +8		\Windows = +8		\TEMP = +15	
Size		< 800 kb = +8		< 800 kb = +8		< 800 kb = +8	
Owner			·		·	LOCAL_SYSTEM = +5	
MAC (Timestamp)			·		·		·
YARA Rules	hard IoC		·		·	WCE Editor = +70	
	soft IoC		·	String Match = +14			·
Notice	Score = 44			Warning	Score = 53	Alarm	Score = 141

Thor and Loki are developed by German cyber security firm "Nextron Systems". On their corporate website, they provide some guidance on how scoring is performed!

You will notice how the mechanism combines "known bads" (e.g. YARA rules) with more generic information (file location, size, owner...)

https://www.bsk-consulting.de/apt-scanner-thor/

Thor & Loki's Scoring Mechanism

When running any type of tool in cyber security, "knowing your tools" is a good rule to live by. As you will see in the next few slides, Loki provides an automated analysis of target systems. But how does it determine whether or not a specific executable / process / ... is suspicious?

Thor and Loki are developed by German cyber security firm "Nextron Systems". On their corporate website, they provide some guidance on how scoring is performed. The slide above provides a detailed understanding of how the scoring mechanism works. You will notice how the mechanism combines "known bads" (e.g. YARA rules) with more generic information (file location, size, owner...).

The idea is to ensure both Loki and Thor can do more than just "hard IOC matching."

Loki's Command-Line Interface (1)

Loki is a command-line based tool used to scan for IOCs.

It is written in Python, and in this screenshot, we see the compiled version in action on Windows.

The Python version of Loki itself can also be used to perform IOC scans, but this implies that Python and all the modules Loki depends on (like YARA) must be installed and deployed on the system we want to scan.

Not only does this require an increased system administration effort to deploy and maintain Python and the required modules on all Windows machines in an organization, it also provides a powerful scripting tool that can be abused by attackers.

Therefore, Florian Roth also provides compiled versions of Loki: These are single Windows executables that contain an embedded Python interpreter with modules and the Loki programs. When executed, it will start a Python engine to execute the Loki Python scripts from a temporary folder.

Loki takes several options and arguments; these can be explored with the help (–h) option as shown in the screenshot above.

When Loki is started without arguments or options, it will perform a full scan of the machine it runs on.

Loki's Command-Line Interface (2)

When Loki is started, it will display its banner and copyright notice.

When run for the first time, Loki will fetch its IOC database.

Loki's Command-Line Interface (2)
When Loki is started, it will display its banner and copyright notice.

Loki is a command-line program, but since it can run without arguments or options, it can also be started by just double-clicking it in Windows Explorer.

When Loki is run for the first time, it will notice that it has no IOC database (folder signature-base) and download it from the GitHub repository from Florian Roth for Loki.

It is a ZIP file that contains all the signatures used by Loki (hashes, YARA rules, …).

These signatures will also be downloaded in subsequent uses of Loki when there are updates available.

In this example, we see that Loki was started from a command prompt without administrative rights.

The consequence is that Loki will not be able to scan all Windows resources, like process memory.

Loki's Output (1)

In the screenshot above, we can see that Loki is based on YARA rules.

These YARA rules are part of the signatures downloaded by Loki from the Loki GitHub repository and are initialized at program startup.

From the name of the YARA rules, we can deduce that Loki looks for hacker tools and exploitation tools used by advanced adversaries, and not for common malware.

We see YARA rules for Empire, Regin, tools from the Equation group, …

In this example, we also see that Loki was started from a command prompt without administrative rights.

The consequence is that Loki will not be able to scan all Windows resources, like process memory. For security reasons, a normal user cannot access all resources. Files of other users for example, and the process memory, some registry values, …

To maximize the chances of Loki to perform a successful scan, it must be executed with an account that has administrative rights.

Loki's Output (2)

In the screenshot above, we can see Loki messages appearing while performing a scan of the filesystem of the computer.

To perform a scan of all files in the filesystem, administrative rights are required, because normal users don't have read access to all files.

Here, we can see some suspicious files detected by Loki.

Loki is a scanner that just detects resources that match IOCs, like files: It will not delete or clean a malicious file like an antivirus would do.

The colors (red and amber) which are different from normal messages (green and blue) indicate successful IOC matches.

Here, for example, we see Mimikatz detections.

These are actually the modified Mimikatz versions that the Petya/Notpetya ransomware used to extract credentials from memory to execute lateral movement.

```
                    Administrator: Command Prompt              —  □   ×
[INFO] Scanning Process PID: 1616 NAME: schedul2.exe CMD: "C:\Program Files (x86
)\Common Files\Acronis\Schedule2\schedul2.exe"
[INFO] Scanning Process PID: 1660 NAME: afcdpsrv.exe CMD: "C:\Program Files (x86
)\Common Files\Acronis\CDP\afcdpsrv.exe"
[INFO] Scanning Process PID: 1772 NAME: DbxSvc.exe CMD: C:\windows\system32\DbxS
vc.exe
[INFO] Scanning Process PID: 1892 NAME: svchost.exe CMD: C:\windows\System32\svc
host.exe -k utcsvc
[INFO] Scanning Process PID: 1904 NAME: dasHost.exe CMD: dashost.exe {b2aeb56d-a
b25-479b-98cb0cc9671df6bf}
[INFO] Scanning Process PID: 1924 NAME: dirmngr.exe CMD: "C:\Program Files (x86)
\GNU\GnuPG\dirmngr.exe" --service
[NOTICE] Listening process PID: 1924 NAME: dirmngr.exe COMMAND: "C:\Program File
s (x86)\GNU\GnuPG\dirmngr.exe" --service IP: 127.0.0.1 PORT: 49412
[INFO] Scanning Process PID: 1952 NAME: pg_ctl.exe CMD: "C:\metasploit\postgresq
l\bin\pg_ctl.exe" runservice -N "metasploitPostgreSQL" -D "C:/metasploit/postgre
sql/data"
[INFO] Scanning Process PID: 2000 NAME: ruby.exe CMD: C:\metasploit\ruby\bin\rub
y.exe -C "C:\metasploit\apps\pro\engine" prosvc_service.rb -E production
[NOTICE] Listening process PID: 2000 NAME: ruby.exe COMMAND: C:\metasploit\ruby\
bin\ruby.exe -C "C:\metasploit\apps\pro\engine" prosvc_service.rb -E production
IP: 127.0.0.1 PORT: 50505
[NOTICE] Established connection PID: 2000 NAME: ruby.exe COMMAND: C:\metasploit\
ruby\bin\ruby.exe -C "C:\metasploit\apps\pro\engine" prosvc_service.rb -E produc
tion LIP: ::1 LPORT: 49564 RIP: ::1 RPORT: 7337
```

In this example, we are executing Loki with administrative privileges.

This means that Loki is able to scan process memory.

Running Loki with Administrative Credentials
In the screenshot above, we are executing Loki with administrative rights.

This means that Loki is able to scan process memory.

A normal user can only scan the process memory of processes that run with the same user account. Processes of other accounts cannot be accessed.

An administrator account has the debug privilege, and this privilege can be activated to access processes of other accounts: This means that the memory can be read and scanned.

Loki will only perform process memory scans when it has the debug privilege (e.g. is running under an administrative account). If it is running as a normal user, it will not perform memory scans (even limited to processes of the same account as the one executing Loki).

Loki will use YARA rules to scan the process memory, and it will also report on processes that have open ports, i.e. that are listening for network connections.

Loki will create a log file of its scanning activities.

This text file is created in the same folder that contains the Loki executable.

Loki Generating Log Files
Loki will create a log file of its scanning activities.

This text file is created in the same folder that contains the Loki executable.

This log file does not only contain detection for IOCs, but it also gives an indication what type of scans Loki performs.

From the start of the example above, we can see that Loki looks for a large amount of the following IOC types:

- Filenames (with regular expression patterns).
- Command & control server indicators.
- MD5 hashes
- SHA1 hashes
- SHA256 hashes

It even has a whitelist: False positive hashes.

This log can be processed automatically after a Loki scan for IOC detections.

Loki can also produce a format better suitable to automatic processing with the CSV option.

Threat Intelligence – Summary

Threat intelligence can be an excellent addition to your cyber security toolkit, but there are a few pitfalls you need to evade:

- Ensure you understand the difference between strategic, tactical and operational threat intelligence (and how to use each category).
- Set up a process to obtain valuable threat intelligence that is relevant to your organization.
- Increase your maturity so threat intelligence can be effectively leveraged / operationalized!

SANS has a dedicated course on how to deal with all different concepts of threat intelligence, label FOR578 – Cyber Threat Intelligence.

Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- Day 4: Lateral Movement
- **Day 5: Action on Objectives, Threat Hunting & Incident Response**
- Day 6: APT Defender Capstone

SEC599.5

Domain dominance
- Dominating the AD - Basic strategies
- Golden Ticket, Skeleton Key, DCSync and DCShadow
- Detecting domain dominance
- Exercise: Domain dominance

Data exfiltration
- Common exfiltration strategies
- Exercise: Detecting data exfiltration

Leveraging threat intelligence
- Defining threat intelligence
- Exercise: Leveraging threat intelligence with MISP & Loki

Threat Hunting & incident response
- Proactive threat hunting strategies
- Exercise: Hunting your environment using OSQuery
- incident response process
- Exercise: Finding malware using Volatility & YarGen

This page intentionally left blank.

Exercise: Leveraging Threat Intelligence with MISP & Loki

Please refer to the workbook for further instructions on the exercise!

This page intentionally left blank.

Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- Day 4: Lateral Movement
- **Day 5: Action on Objectives, Threat Hunting & Incident Response**
- Day 6: APT Defender Capstone

SEC599.5

Domain dominance
- Dominating the AD - Basic strategies
- Golden Ticket, Skeleton Key, DCSync and DCShadow
- Detecting domain dominance
- Exercise: Domain dominance

Data exfiltration
- Common exfiltration strategies
- Exercise: Detecting data exfiltration

Leveraging threat intelligence
- Defining threat intelligence
- Exercise: Leveraging threat intelligence with MISP & Loki

Threat Hunting & incident response
- Proactive threat hunting strategies
- Exercise: Hunting your environment using OSQuery
- incident response process
- Exercise: Finding malware using Volatility & YarGen

This page intentionally left blank.

Can We Detect Advanced Adversaries in Real Time?

Traditionally, a lot of effort has been placed on real time detection techniques...

Log centralization

SIEM technology

IDS

24/7 SOC

Alerting

Next-Gen FW

Specific signatures define malicious behavior and alerts when triggered.

Monitoring of these alerts is performed by a multi-tiered SOC team that reviews and categorizes alerts. Upon identification of a confirmed incident, the incident response process is kicked off!

Can We Detect Advanced Adversaries in Real Time?

Is it possible to detect advanced adversaries that infiltrated our corporate networks in real-time? This difference in goals and modus operandi between advanced adversaries and common adversaries is reflected in the methods we apply to detect advances adversaries versus common adversaries.

Traditionally, a lot of effort has been placed on real-time detection technologies. Typical technologies that fit in this area include:

- Log centralization
- SIEM technology
- 24/7 Security Operations Center
- Automated alerting
- Intrusion detection systems
- Next-gen firewall
- …

These real-time detection technologies have been put into place to detect common adversaries, but are they effective to defeat advanced adversaries?

We typically rely on the definition of known bads for real-time detection. Known bads are specific signatures that define malicious behavior and can be used to generate alerts when the signature is triggered.

A known bad, for example, is the IP address of a known command & control server used by a specific ransomware family. We can use an IDS to define a rule that triggers each time we see a TCP connection to that IP address.

Would this rule alert us when malware connects to its command & control server with that IP address? Yes!
Would this rule alert us when malware connects to its command & control server with another IP address? No!
Would this rule alert us when one of our users visits a web site with that IP address? Yes!

This illustrates a couple of problems with real-time detection. A TCP connection to this specific address is indirect evidence of malware in our corporate network. Yes, malware that connects to that IP address will be detected. But non-malicious connections to that IP address will also generate alerts: False positive alerts.

How could these false positive alerts happen? Due to the shortage of IPv4 addresses, IPv4 addresses are shared and reused. Web servers can host many websites with the same Ipv4 address, and after some time, servers are decommissioned and their IPv4 addresses are reused for other purposes.

This sharing and reuse lead to false positive alerts.

"Alarmingly, despite having invested significantly in information security solutions to the point of utilizing dozens of point products, nearly 74% of those surveyed reported that **security events/alerts are simply ignored** because their teams can't keep up with the suffocating volume."

Enterprise Strategy Group study, 2016

Real-Time Detection Issues: Security Alert Fatigue
One problem with real-time detection is security alert fatigue.

Real-time detection methods generate too many false positive alerts, with a negative impact on the motivation and morale of security teams that have to investigate the alerts.

According to an Enterprise Strategy Group study from 2016,

"Alarmingly, despite having invested significantly in information security solutions to the point of utilizing dozens of point products, nearly 74% of those surveyed reported that security events/alerts are simply ignored because their teams can't keep up with the suffocating volume."

This is an alarming trend. With ¾ of the respondents reporting alert fatigue, real-time detection methods cannot be called effective. Inside that large volume of alerts (mostly false positive alerts), some true positive alerts will occur.

But because of alert fatigue, these true positive alerts will be ignored too, and attacks will remain undetected.

Real-Time Detection Issues: Sophisticated Attacks Have Different Steps

Can you really detect sophisticated attacks in real-time? Adversaries take their time when laterally movement throughout your network, it's often not easy to "connect the dots" and realize what's going on...
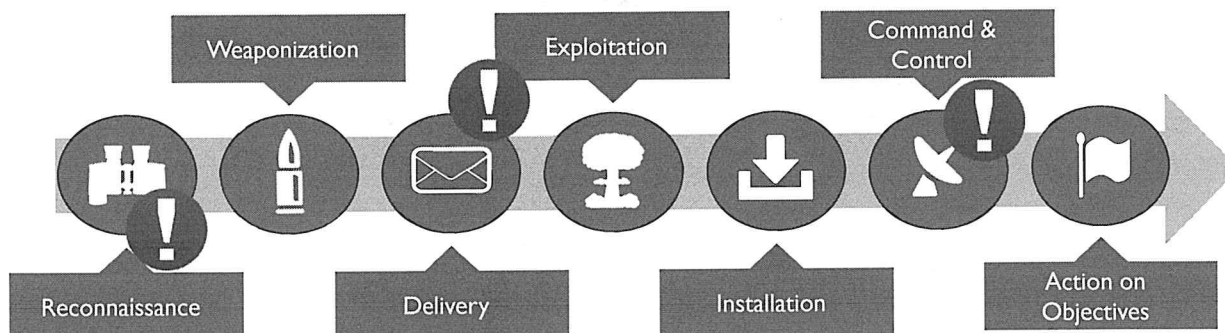
Real-Time Detection Issues: Sophisticated Attacks Have Different Steps
Can you really detect sophisticated attacks in real-time? Adversaries take their time when laterally movement throughout your network, it's often not easy to "connect the dots" and realize what's going on...

To give you a practical example: Consider an adversary that spiders your web site, identifies a "jobs@" email address as a target for phishing, delivers a weaponized CV, infects the system of an HR employee, and sets up a command & control channel.

Your real-time alerting solution might indicate that a spider is running over your web site, but it won't be able to connect the dots to:

- Understand that two days later, a phishing email was sent to your "jobs@" email address with a weaponized CV document (including a malicious macro);
- Notice the C&C channel that is set up from the workstation of the HR employee towards the adversary server.

Can you really detect sophisticated attacks in real-time?

99 days*

The global median time between compromise and detection is 99 days*
(Source: FireEye M-Trends 2017)

Real-Time Detection Issues: Some Statistics

A strong indicator that shows that real-time detection fails to detect sophisticated attacks comes from a FireEye survey published in M-Trends 2017: "The global median time between compromise and detection is 99 days."

This shocking statistic should make us reflect on the strategies we employ! It means that on average, more than 3 months take place between a successful attack inside a corporate network and the detection of said attack by the corporate security teams. Not only is this a long period, but it is the global median time: Half of the attacks take 3 months and longer to be detected (if they get detected at all).

FireEye's report is not unique. Similar statistics have been reported by other security companies and organizations.

Sophisticated attacks can stay under the radar for a long time. Not only does this give the opportunity for advanced adversaries to operate undetected inside our corporate networks for a long period, it also means that our capability to do something about the attack is severely reduced.

Real-Time Detection Issues: Summary

While effective real-time detection would be ideal, there are a few hiccups that often occur:

- Advanced adversaries typically use tailored, currently unknown, techniques (so you have no predefined "alerts" or "signatures" for that).
- Real-time detection has to happen in "real-time" (☺), which often leaves no room for in-depth analytics resulting in false positives and noise.
- Real-time detection is often fully outsourced to external SOCs, which lack context on your environment, again resulting in increased noise!
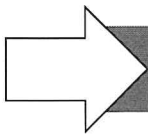
Many organizations are embracing threat hunting and complement their real-time detection efforts with periodical threat hunting efforts. Instead of waiting for a security alert, they go out and search for suspicious behavior themselves!

Real-Time Detection
- Relies on known bads (e.g. signatures, rules...)
- Generates alerts that are to be further investigated by analysts.
- Is typically highly automated .
- Can leverage Indicators of Compromise.
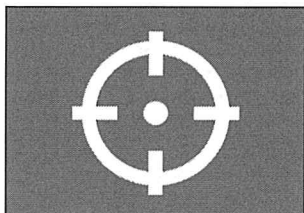
Threat Hunting
- "Hunt" environment for unknown bads.
- Generates confirmed incidents, which trigger incident response.
- Can partially leverage automation, but majority is manual effort
- Can generate Indicators of Compromise.

Threat Hunting vs Real-Time Detection

When we compare threat hunting with real-time detection, many differences will stand out.

While real-time detection is based on known bads, threat hunting is based on anomalies. Real-time detection takes a narrow view on our corporate infrastructure using rules. Threat hunting tries to see the bigger picture and looks for anomalies and strange behavior inside our corporate infrastructure. An alert would be: We detected a TCP connection to IP address X.X.X.X. An anomaly would be: For this day, we see an unusually large volume of data flowing to website XYZ. Another example: Real-time detection would detect a User Agent String used by a known family of malware; threat hunting would uncover User Agent Strings that have never been seen before inside our corporate network.

Real-time detection generates alerts that require further investigation, while hunting generates confirmed incidents which trigger incident response. While an SOC operator has to go through a list of alerts to detect attacks, a threat hunting analyst is presented with a large volume of data that he analyzes. In threat hunting, automation is mainly used to enhance the visibility of that large "data set" (e.g. using the ELK stack for dashboarding can be very powerful).

While real-time detection is highly automated, threat hunting is a manual effort that can be machine assisted.

With regards to threat intelligence, real-time detection typically consumes Indicators of Compromise, while threat hunting can help you generate Indicators of Compromise.

HMM 0 Initial	HMM 1 Minimal	HMM 2 Procedural	HMM 3 Innovative	HMM 4 Leading
• Relies primarily on automated alerting • Little or no routine data collection	• Incorporates threat intelligence indicator searches • Moderate or high level of routine data collection	• Follows data analysis procedures created by others • High or very high level of routine data collection	• Creates new data analysis procedures • High or very high level of routine data collection	• Automates the majority of successful data analysis procedures • High or very high level of routine data collection

Threat Hunting – Maturity Model
Threat hunting can be performed by all organizations (and a lot of them are already doing it, without using the term!). It's important, however, to understand what level of maturity your organization currently has and aim for continuous improvement. In his personal blog, David Bianco defined an interesting "maturity model" for threat hunting. His hunting maturity model (HMM) has 5 levels, ranging from 0 to 4:

Level HMM0 is the initial level.
Organizations with this maturity level rely primarily on automated alerting and have little or no routine data collection.

Level HMM1 is the minimal level.
Organizations with this maturity level incorporate threat intelligence indicator searches and have a moderate or high level of routine data collection.

Level HMM2 is the procedural level.
Organizations with this maturity level follow data analysis procedures created by others and have a high or very high level of routine data collection.

Level HMM3 is the innovative level.
Organizations with this maturity level create new data analysis procedures and have a high or very high level of routine data collection.

Level HMM4 is the leading level.
Organizations with this maturity level automate the majority of successful data analysis procedures and have a high or very high level of routine data collection.

By determining your corporate threat hunting level compared to this model, you know where you are and where you can evolve to.

In the SANS whitepaper "The Who, What, Where, When, Why and How of Effective Threat Hunting," SANS instructors Rob Lee & Robert M Lee consider the "Sliding Scale of Cyber Security":
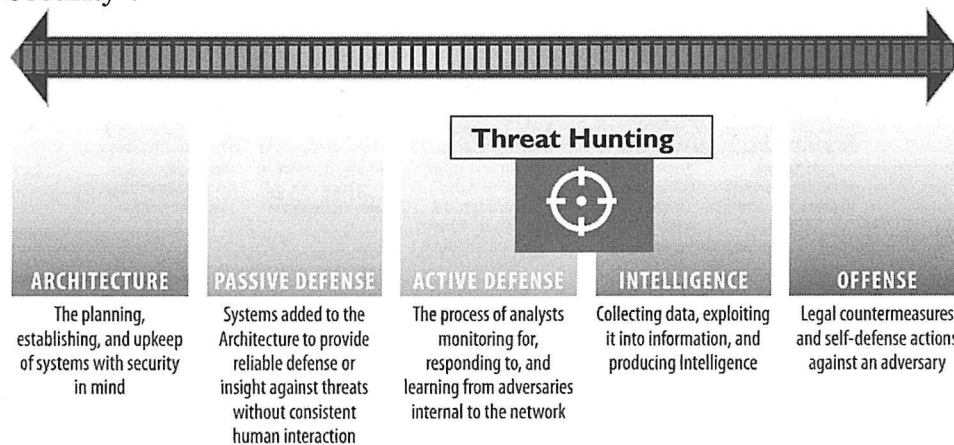
Threat Hunting

ARCHITECTURE	PASSIVE DEFENSE	ACTIVE DEFENSE	INTELLIGENCE	OFFENSE
The planning, establishing, and upkeep of systems with security in mind	Systems added to the Architecture to provide reliable defense or insight against threats without consistent human interaction	The process of analysts monitoring for, responding to, and learning from adversaries internal to the network	Collecting data, exploiting it into information, and producing Intelligence	Legal countermeasures and self-defense actions against an adversary

Threat Hunting – The Sliding Scale of Cyber Security (1)

In the SANS whitepaper "The Who, What, Where, When, Why and How of Effective Threat Hunting" SANS instructors Rob Lee & Robert M Lee define the "Sliding Scale of Cyber Security." This sliding scale is similar to the Threat Hunting Model, but the focus here is on defining 5 different phases of investment organizations can make when tackling cyber security. These phases are:

- **Architecture**: The architecture phase refers to all typical aspects of planning for cyber security. The idea is to ensure design within the organization (including systems, networks, applications...) is done with cyber security in mind. This phase thus attempts to prevent vulnerabilities from arising.
- **Passive defense**: Within passive defense, we consider all tools and systems that are added to the architecture to provide additional defense or insight against threats, WITHOUT consistent human interaction. This thus typically includes firewalls, IDS, IPS...
- **Active defense**: Active defense covers all activities performed by analysts monitoring for, responding to, and learning from adversaries internal to the network.
- **Intelligence**: This phase is the process of collecting data, turning it into information that can be used to generate intelligence / useful knowledge that can help improve an organization's security posture.
- **Offense**: This last category includes LEGAL countermeasures that can be opted for by organizations when defending against adversaries. It's important to note that, for private organizations, offensive options are highly limited, and they typically have a low ROI (investments are typically better placed in other phases of the scale).

On this sliding scale, we can position threat hunting mainly as part of the "Active defense" phase, with an integration of some "Intelligence" fundamentals. Meanwhile, security monitoring can be positioned between "Passive defense" and "Active defense."

ARCHITECTURE	PASSIVE DEFENSE	ACTIVE DEFENSE	INTELLIGENCE	OFFENSE
The planning, establishing, and upkeep of systems with security in mind	Systems added to the Architecture to provide reliable defense or insight against threats without consistent human interaction	The process of analysts monitoring for, responding to, and learning from adversaries internal to the network	Collecting data, exploiting it into information, and producing Intelligence	Legal countermeasures and self-defense actions against an adversary

We can use the sliding scale to understand what the potential ROI of threat hunting in your organization can be!

If there are serious gaps in architecture and passive defense (e.g. the environment is full of vulnerabilities), threat hunting ROI will be limited.

Threat Hunting – The Sliding Scale of Cyber Security (2)
We can use this sliding scale to easily assess to what extent threat hunting can be a valuable investment of time and resources for an organization.

Imagine, for example, an organization that has significant gaps in architecture and passive defense: No proper vulnerability management process is in place and the majority of systems is unpatched. This is a rather extreme example, but it's important to note that in such an environment, threat hunting is not a wise investment: Due to the immature architecture, too much noise will be generated, resulting in ineffective hunting results.

So, how can we start doing threat hunting? ... What do we need?

Experienced analysts who know how attacks work and what to look out for. These people should also understand your environment and know what your crown jewels are.

A large collection of logs that is being generated throughout different parts of your environment. This includes any type of logs (Windows event logs, firewalls...)

A large, centralized, data repository that can be used to collect available logs for your environment.

Visualization tools that can help analysts understand what all of the logs mean and facilitate deeper analysis and investigation.

Threat Hunting – Critical Success Factors

Threat hunting can detect malicious activity by reviewing available logs for anomalies.

Critical success factors for threat hunting are:

- Experienced analysts who know how attacks work and what to look out for. These people should also understand your environment and know what your crown jewels are. A good knowledge of offensive security methods is required to be a good threat hunter: A poacher makes the best gamekeeper. Meanwhile, these people should also understand your environment, which is not an easy "blend".

- A large collection of logs that are being generated throughout different parts of your environment. This includes any type of logs (Windows event logs, firewalls...). Threat hunting relies heavily on logs generated by different systems in your corporate network. These logs are not always enabled by default, so you might need to revise your logging and monitoring strategy.

- A large, centralized, data repository that can be used to collect available logs for your environment. We cannot rely on a myriad of different log types spread across our environment. In order to effectively hunt, we need to collect all logs centrally, so we can easily search, query and analyze them.

- Lastly, as we will be facing vast amounts of data, it's important that data can be correctly visualized, so analysts can create dashboards that facilitate further investigation. Yet again, the ELK stack provides an interesting solution for this!

Threat Hunting – Overall Process
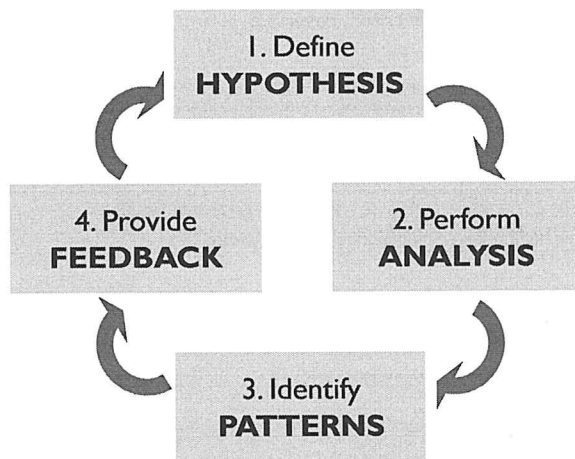
Threat hunting should be an iterative process, where the following actions are performed:

1. **Define hypothesis:** Define a hypothesis that can be tested (e.g. "Adversaries are attempting to infiltrate our organization using phishing mails").
2. **Perform analysis:** Test the hypothesis by analyzing the available logs.
3. **Identify patterns:** Through your analysis, identify potential patterns of malicious behavior.
4. **Provide feedback:** Based on the results of the hunt, provide feedback (e.g. definition of new IOCs or use cases that can be used in real-time detection).

Threat Hunting – Overall process
Threat hunting should be an iterative process, where the following actions are performed:

1. Define hypothesis: Define a hypothesis that can be tested (e.g. "Adversaries are attempting to infiltrate our organization using phishing emails").
2. Perform analysis: Test the hypothesis by analyzing the available logs.
3. Identify patterns: Through your analysis, identify potential patterns of malicious behavior.
4. Provide feedback: Based on the results of the hunt, provide feedback (e.g. definition of new IOCs or use cases that can be used in real-time detection).

Throughout this section of the course, we will zoom in on a few of these phases!

Threat Hunting – Definition of Hypotheses Is Key!

As we've seen in the previous diagram, the first step in threat hunting is the definition of hypotheses. This might sound intimidating, but it's a fairly straightforward process. In order to provide some clarity, we have provided an easy-to-understand example of these hypotheses:

- Intelligence-driven hypothesis: "I know this APT group uses C&C servers hosted in South Africa. I will review my perimeter connectivity for traffic to South African servers."
- Situational awareness hypothesis: "I know the crown jewels for my organization are our new R&D plans, so I will create hypotheses on how these could be stolen."
- Domain expertise hypothesis: "I am an expert in DNS and know DNS could be used as a covert channel to exfiltrate data. I will thus review outgoing DNS traffic for anomalies."

There is no answer to the question: "What type of hypothesis definition is best?" Successful threat hunting combines these different types of hypotheses, as they can all be highly useful in a given situation. Robert M. Lee & David Bianco wrote an excellent whitepaper titled "Generating Hypotheses for Successful Threat Hunting"... a must-read for threat hunters!

As "threat hunting" is becoming increasingly popular, some vendors are trying to sell "fully automated hunting" solutions. However, these don't exist...

Automation can be useful, but we need more!

Threat hunting handles large volumes of data and thus benefits from automation techniques:

- Automatically collecting logs from end-point systems.
- Using data analysis techniques to present data in a meaningful way to hunter.
  - Least-frequency analysis.
  - Visualization and dashboarding techniques.

The crucial part is using the human effort where it is used best: Not to crunch millions of alerts, but to create / define **a data analysis technique** and **reviewing its results!**

Threat Hunting – A Word on Automation
As "threat hunting" is becoming increasingly popular, some vendors are trying to sell "fully automated hunting" solutions. These, however, don't exist...

Automation can be useful, but we need more!

Because threat hunting handles large volumes of data, it can greatly benefit from automation techniques:

- The collection and centralization of logs from end-point systems should be fully automated.
- Data analysis techniques that present the data in a meaningful way to the threat hunting analyst have to be used, like:

  - Least-frequency data analysis techniques
  - Data visualization techniques
  - Dashboarding techniques
  - …

It is crucial to use the human effort where it is used best: Not to crunch millions of alerts, but to create / define a data analysis technique and reviewing its results afterward. Humans are not good at boring, repetitive tasks, but they excel at recognizing patterns.

As we discussed before, log collection is a key part of threat hunting! In an ideal environment, you are already collecting logs from a wide variety of sources. As we don't live in an ideal world, however, it is sometimes up to the hunter to arrange for his own log collection. Here's two interesting approaches:

Agent based	Multiple vendors have agents available via which you are able to extract logs from hosts. Agents allow for easy central management and often have many features such as IOC hunting built in already. However, adding an agent to a host installation is something that is often frowned upon by the workstation/server management team.
Script based	Scripts allow for great flexibility and don't add a load to your hosts when they are not running. This is often a preferred option by incident responders when they need to obtain information fast and don't have time to wait for an agent to be deployed. But scripts require maintenance and might not provide all features an agent has.

Threat Hunting – Collecting Required Logs
As we discussed before, log collection is a key part of threat hunting! In an ideal environment, you are already collecting logs from a wide variety of sources. As we don't live in an ideal world, however, it is sometimes up to the hunter to arrange for his own log collection. Here are two interesting approaches:

Agent-based:
Multiple vendors have agents available via which you are able to extract logs from hosts. Agents allow for easy central management and often have many features such as IOC hunting built in already. However, adding an agent to a host installation is something that is often frowned upon by the workstation/server management team.

Script-based:
Scripts allow for great flexibility and don't add a load to your hosts when they are not running. This is often a preferred option for incident responders when they need to obtain information fast and don't have time to wait for an agent to be deployed. Scripts, however, require maintenance and might not provide all features an agent has.

Later in this course, we will discuss some script-based approaches and agent-based approaches.

When discussing the collection of logs using scripts, one (of many) readily available resources is PSHunt, created and open sourced by Infocyte.

"PSHunt is a Powershell Threat Hunting Module designed to scan remote endpoints for Indicators of Compromise or survey them for more comprehensive information related to state of those systems (active processes, autostarts, configurations, and/or logs)."

You can invoke PSHunt through various channels (WMI, PowerShell Remoting, Scheduled Tasks, and PSExec).

PSHunt has different modules and functions:
- Discovery: Identifies hosts within the network.
- Scanners: Deploys scripts to hosts to collect information such as registry values and OS info.
- Surveys: Deploys scripts to hosts to collect information from those hosts (digs deeper than scanners).
- Analysis: Provides a framework for analyzing and displaying survey and scan results.

Threat Hunting – Collecting Required Logs – Polling for Information
When talking about collecting logs using scripts, one readily available resource is PSHunt, created and open sourced by Infocyte.

"PSHunt is a Powershell Threat Hunting Module designed to scan remote endpoints for indicators of compromise or survey them for more comprehensive information related to state of those systems (active processes, autostarts, configurations, and/or logs)."

You can invoke PSHunt through various channels (WMI, PowerShell Remoting, Scheduled Tasks, and PSExec).

PSHunt has different modules and functions:

- Discovery: Identifies hosts within the network.
- Scanners: Deploys scripts to hosts to collect information such as registry values and OS info.
- Surveys: Deploys scripts to hosts to collect information from those hosts (digs deeper than scanners).
- Analysis: Provides a framework for analyzing and displaying survey and scan results.

PSHunt can be found here:
https://github.com/Infocyte/PSHunt

Threat Hunting – Data Visualization
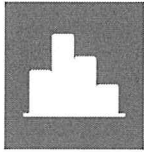
As threat hunting handles massive amounts of data, its parsing and visualization will be of the utmost importance to ensure threat hunters can spend their precious time wisely! Excel should not be your main threat hunting tool!

As we've seen during previous parts of the course, our beloved ELK stack can come in handy here again, as it allows for easy creation of custom visualizations!

For an enterprise environment, consider the following example setup for host-based information:

- Deploy OSQuery enterprise-wide using GPOs.
- Develop queries that will run periodically (e.g. once every day) on all your Windows systems.
- Use Elastic's Filebeat to monitor the OSQuery log file and forward all events to an ELK stack.
- Centrally hunt from your Kibana dashboards.

Threat Hunting Tools – Data Visualization

As threat hunting handles massive amounts of data, its parsing and visualization will be of the utmost importance to ensure threat hunters can spend their precious time wisely! We do not want to have our analysts search through raw log files or create pivot tables in Excel... Excel and Notepad should not be your main threat hunting tools!

As we've seen during previous parts of the course, our beloved ELK stack can come in handy here again, as it allows for easy creation of custom visualizations! It's a wise idea for threat hunters to spend some time getting familiar with Kibana, as the definition of useful visualizations could be key for successful hunting!

For an enterprise environment, consider the following example setup for host-based information:

- Deploy OSQuery enterprise-wide using GPOs.
- Develop queries that will run periodically (e.g. once every day) on all your Windows systems.
- Use Elastic's Filebeat to monitor the OSQuery log file and forward all events to a central ELK stack.
- Centrally monitor and hunt from your Kibana dashboards.

We will implement such an approach in our upcoming exercise!

Threat Hunting – Summary

> Many organizations are already doing threat hunting without actually using the term.

> Depending on the maturity of the organization, threat hunting can provide a high ROI.

> While threat hunting can leverage automation, the process can never be fully automated. The experience of the hunter is of vital importance (e.g. for hypothesis definition).

> Successful threat hunting will require at the very least expert resources, a central repository for logs and data parsing / visualization tooling.

SANS has dedicated courses that tackle threat hunting in much more detail, e.g. "SANS FOR508 – Advanced Digital Forensics, Incident Response, and Threat Hunting"

Threat Hunting – Summary

In summary, we'd like to provide the following key takeaways related to threat hunting:

- Many organizations are already doing threat hunting, without actually using the term. Whenever analysts are actively looking through their environment to identify suspicious behavior, they are doing a (limited) form of threat hunting.
- Although threat hunting can be done by a wide variety of organizations, its effectiveness, and actual ROI will largely depend on the maturity of the organization.
- While threat hunting can leverage automation (e.g. for the collection, parsing and visualization of logs), the process can never be fully automated. The experience and expertise of the hunter is of vital importance (e.g. for hypothesis definition).
- Successful threat hunting will require at the very least expert resources, a central repository for logs and data parsing / visualization tooling.

SANS has a dedicated course that tackles threat hunting in much more detail, e.g. "SANS FOR508 – Advanced Digital Forensics, Incident Response, and Threat Hunting."

Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- Day 4: Lateral Movement
- **Day 5: Action on Objectives, Threat Hunting & Incident Response**
- Day 6: APT Defender Capstone

SEC599.5

Domain dominance
- Dominating the AD - Basic strategies
- Golden Ticket, Skeleton Key, DCSync and DCShadow
- Detecting domain dominance
- Exercise: Domain dominance

Data exfiltration
- Common exfiltration strategies
- Exercise: Detecting data exfiltration

Leveraging threat intelligence
- Defining threat intelligence
- Exercise: Leveraging threat intelligence with MISP & Loki

Threat Hunting & incident response
- Proactive threat hunting strategies
- Exercise: Hunting your environment using OSQuery
- incident response process
- Exercise: Finding malware using Volatility & YarGen

This page intentionally left blank.

Please refer to the workbook for further instructions on the exercise!

This page intentionally left blank.

Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- Day 4: Lateral Movement
- **Day 5: Action on Objectives, Threat Hunting & Incident Response**
- Day 6: APT Defender Capstone

SEC599.5

Domain dominance
- Dominating the AD - Basic strategies
- Golden Ticket, Skeleton Key, DCSync and DCShadow
- Detecting domain dominance
- Exercise: Domain dominance

Data exfiltration
- Common exfiltration strategies
- Exercise: Detecting data exfiltration

Leveraging threat intelligence
- Defining threat intelligence
- Exercise: Leveraging threat intelligence with MISP & Loki

Threat Hunting & incident response
- Proactive threat hunting strategies
- Exercise: Hunting your environment using OSQuery
- incident response process
- Exercise: Finding malware using Volatility & YarGen

This page intentionally left blank.

Incident response is a process that is started when an incident is detected: The organization has detected that (advanced) adversaries have breached security and starts incident response activities.

Incident response activities are typically executed by a dedicated team of experts, the Computer Incident Response Team (CIRT).
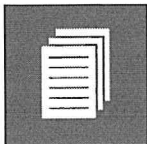
It's almost certain that at one point in time, a security incident will occur. It is thus important to be prepared. Furthermore, several compliance regulations will require a proper incident response plan to be in place (e.g. GDPR).

Incidence Response Process

Incident response is a process that is started when an incident is detected: We have detected that (advanced) adversaries have breached our security. The reaction to this is to put the organization into incident response mode. This does not mean that the complete organization is involved, usually just a small team.

Incident response requires a dedicated team: The Computer Incident Response Team (CIRT). Another well-known name for this type of team is the Computer Emergency Response Team (CERT).

The first CERT created in the world was Carnegie Mellon University's CERT. Carnegie Mellon University has legal rights to the name Computer Emergency Response Team/CERT, and that is why many organizations use another but similar name, like Computer Incident Response Team (CIRT).

It's almost certain that at one point in time, a security incident will occur. It is thus important to be prepared. Furthermore, several compliance regulations will require a proper incident response plan to be in place (e.g. GDPR). If no dedicated team exists in the organization when an incident occurs, a third party specialized in incident response can be contracted. Do take into account, however, that even if you can rely on third-party forensic / malware analysts, incident managers, etc., part of the effort will have to be done by your own teams, as they know the environment best.

Who Should Be Part of Your CIRT?
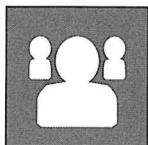
The CIRT is composed of cyber security professionals that know your environment AND know how to handle incidents.

- They prepare and plan ahead to prepare the team to handle incidents.
- Even if it's an external party, they need to have thorough understanding of your environment to perform effective incident response.
- This involves the definition of a clear process to follow in case of an incident, and training and practicing the steps of this process.
- SANS defines a generic 6-step process for incident response (SANS SEC504). Many organizations take this a step further and develop concrete "playbooks" for the most common, expected, security incidents.

Who Should Be Part of Your CIRT?

The Computer Incident Response Team is a dedicated team in an organization that will respond to computer incidents. The team is composed of IT security professionals experienced in incident response. On top of their experience in incident response, team members have experience and skills related to various aspects of IT and computer security. For example, it is not uncommon for a CIRT team member to have the skills to inspect network packets in a packet capture file.

CIRT team members prepare and plan ahead to prepare the CIRT team to handle incidents. Properly handling incidents should be done according to a well-established plan so that no steps are forgotten when an incident is handled.

This involves the definition of a clear process to follow in case of an incident, and CIRT team members should regularly train and practice the steps of this process to be well prepared when an incident occurs. Improvisation will not lead to a good outcome of the incident handling process.

To help prepare CIRTs and CIRT team members to respond to incidents, SANS defines a 6-step process for incident response. This incident response process is covered in SANS training SEC504: "Hacker Tools, Techniques, Exploits, and Incident Handling." Many organizations take this a step further and develop concrete "playbooks" for the most common, expected, security incidents.

As course authors, we've seen several organizations perform varying degrees of incident response with different objectives. Some of the key objectives for incident response include:

One of the most basic incident response objectives is to **contain and eradicate** an incident, after which business operations can return to normal as soon as possible.

In more mature organizations (and when faced with advanced adversaries), an important additional goal is to perform in-depth analysis of the attack in order to **generate threat intelligence**. This can be used to obtain an in-depth understanding of the adversary's attack techniques (and your weaknesses!) in order to further improve defenses to prevent future attacks.

Why Should You Perform Incident Response?
As course authors, we've seen several organizations do varying degrees of Incident Response with different objectives. Some of the key objectives for incident response include:

- One of the most basic incident response objectives is to contain and eradicate an incident, after which business operations can return to normal as soon as possible.
- In more mature organizations (and when faced with advanced adversaries), an important additional goal is to perform in-depth analysis of the attack in order to generate threat intelligence. This can be used to obtain an in-depth understanding of the adversary's attack techniques (and your weaknesses!) in order to further improve defenses to prevent future attacks.

It's important to understand that these objectives are not mutually exclusive. Specific investigations often have both objectives, and there could be scenarios where one objective is more important than the other. Consider a ransomware incident in smaller organizations, for example. The focus will most likely more be on containing and eradicating the incident as soon as possible in order to prevent further damage. In an environment where a more advanced adversary is penetrating several systems and attempting to steal sensitive information, there is probably a lot of benefit to ensure the incident response activities provides insights in the adversary objectives and their TTPs (Tactics, Techniques, Procedures). This would allow the organization to better prepare for future attacks!

SANS' Six-Step Incident Response Process
This is SANS' step-by-step approach to incident response as covered in SANS training SEC504:

1. Preparation
2. Identification
3. Containment
4. Eradication
5. Recovery
6. Lessons Learned

We recommend going through this six-step process (step by step) without skipping any of them. It's important to note that this should always be a continuous process: Once you finish incident response, you derive lessons learned (e.g. TTPs used by your adversaries, missing monitoring capabilities, vulnerabilities in your environment, etc.) that will improve your preparation phase, which will increase the chances of you detecting the incident. We will now discuss these six steps in detail in the upcoming slides.

https://www.sans.org/reading-room/whitepapers/incident/incident-handlers-handbook-33901

OODA LOOP

We could consider incident response as a continuous "battle" between the CIRT and the adversary team. It's important to note that your adversary is also human... We can thus benefit from the **OODA loop!**

Observe: Understand what is happening technically on your network.
Orient: Understand what this attack means: What is the context of this attack? What are the objectives of the adversary? Are your crown jewels at risk?
Decide: Based upon the information collected during the "Observe" and "Orient" steps, decide on the next step. A key pitfall here can be the required authority to decide on next steps.
Act: Effectively implement the action that was decided upon in the previous step.

Observe — Orient — Decide — Act

During a typical incident response engagement, your CIRT team and the adversary run through countless OODA loops. **If your OODA loops are faster than the adversary, you win.** ☺

Incidence Response and the OODA Loop

We could consider incident response as a continuous "battle" between the CIRT and the adversary team. It's important to note that your adversary is also human... We can thus benefit from the OODA loop! OODA stands for Observe, Orient, Act and Decide. OODA loops were initially introduced by US Air Force military strategist John Boyd. The concept has since been applied to a wide variety of subjects, including computer security and incident response. The general ideas behind the steps are the following:

- Observe: Understand what is happening technically on your network. The more your IR team has visibility into what is going on, the more they can understand the attack.
- Orient: Understand what this attack means: What is the context of this attack? What are the objectives of the adversary? Are your crown jewels at risk?
- Decide: Based upon the information collected during the "Observe" and "Orient" steps, decide on the next step. This can often be difficult as the required "authority level" is to be established.
- Act: Effectively implement the action that was decided upon in the previous step.

During a typical incident response engagement, your CIRT team and the adversary run through countless OODA loops. If your OODA loops are faster than the adversary, you win!

Step 1 Preparation

Of course, preparing to respond to an incident takes place before the incident happens.
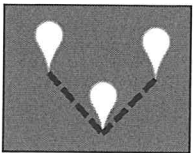
This is arguably the most important step in the incident response process, as it will shape how a Computer Incident Response Team reacts to incidents.

Incidence Response Process – Preparation (1)
The first step is the preparation step.

Of course, the CIRT team and CIRT members prepare to respond to incidents before incidents take place.

If the CIRT team is not (yet) prepared to handle incidents when an incident occurs, then the incident will have to be handled without a plan, or a plan will have to be formulated while the incident response takes place. This is not a good situation: Mistakes will be made, steps of the incident response process will be forgotten or skipped.

This is arguably the most important step in the incident response process, as it will shape how a Computer Incident Response Team reacts to incidents.

Each team member of the CIRT team must be well aware of the plan to follow when they are tasked to handle an incident and apply it accordingly.

A well-prepared plan is essential for a successful outcome of the incident response process.

Step 1   Preparation

Plan Before the Incident

A response plan/strategy to prioritize incidents based upon impact on the organization. A common thing to do is develop "playbooks" for the most common threats against the organization.

A communication plan to define with whom and how to communicate during an incident, including organizations outside the corporate environment such as the general public, law enforcement, customers...

Documentation is key during an incident, as it can be used as evidence in case of legal procedures. Make sure you have a readily available communication infrastructure (consider security as well).

CIRT Team Members

Avoid surprises! Ensure your team includes experienced profiles that have dealt with various aspects of security and forensics before. They should also be trained appropriately.

Have the necessary access to systems (or know how to obtain them) to collect data and evidence.

Have the necessary tools at their disposal to be able to respond to an incident, for example, like forensic disc imaging software.

Incidence Response Process – Preparation (2)

There are several elements that require planning before an incident takes place. This includes:

- A response plan/strategy to prioritize incidents based upon impact on the organization. Depending on the size of your organization, it will not be exceptional that more than one incident occurs at the same time, or that the response to incidents overlaps. As the CIRT team will certainly have limited resources, a plan or strategy must be prepared to deal with multiple incidents: how will we prioritize the handling of multiple incidents in our organization? A reasonable course of action is to prioritize incidents based on the impact they have on our organization: incidents with a high impact should be responded to first. It will not always be clear what the impact of an incident is at the outset and that reprioritization of incidents might be required as the impact becomes clearer.

- A communication plan to define with whom and how to communicate during an incident, including organizations outside the corporate environment like law enforcement. Handling incidents require teamwork and that requires communication. A communication plan does not necessarily have to be complex; it can be a list of people with phone numbers or other communication channels. If an incident handler requires the help of the network team, for example, it should be clear how to contact the network team and how to be assured of their involvement. This might be obvious during working hours in your organization, but when incidents need to be handled outside normal working hours, a good communication plan will prevent wasting time finding key people off hours.

- Documentation plan: Documentation is key during an incident, as it can be used as evidence in case of legal procedures. Documentation is key in incident handling, not only for the last step (Lessons Learned), but also for the process itself, and certainly when the reaction to an incident will include legal action.

A CIRT team must be formed with members that:

- Are experienced in various aspects of security and forensics. All kinds of incidents will happen in a large organization; therefore, it is important to compose a CIRT team with a diverse group of team members. A single team member cannot have all the required skills and expertise to handle all possible incidents properly; diversification is necessary. For example, one team member might be experienced in networking technology while another team member is more experienced in software security.

- Have the necessary access to systems (or know how to obtain them) to collect data and evidence. CIRT team members will have to access systems to collect data and evidence pertaining to the incident that is being responded to. Depending on your organization, CIRT team members might have all necessary accesses in their user profile, or else they will need to obtain the necessary rights when required. To prevent losing time to obtain necessary rights, a plan should be established.

- Have the necessary tools at their disposal to be able to respond to an incident, for example, like forensic disc imaging software. Incident response can be a very technical discipline, requiring very specialized tools and software that is not used by other teams.

- Have been trained appropriately. This is a repetitive process; team members must attend training regularly to keep their skills up-to-date.

An excellent source for incident response playbooks are the Incident Response Methodologies (IRM) developed by CERT Societe Generale.

- Available in cheat-sheet format at https://github.com/certsocietegenerale/IRM.
- These playbooks also follow the SANS incident response process.
- Currently available in English, Spanish and Russian.

Some of the available IRMs include:

- Worm Infections
- Phishing
- Ransomware
- DDoS
- Windows intrusions
- ...

CERT | SOCIETE GENERALE

IRM (Incident Response Methodologies)

The IRMs are an excellent basis that can be further fine-tuned to the specific needs of your organization!

Incidence Response Process – Incident Response Playbooks

In the modern age, many organizations are facing similar threats: It's no surprise that a variety of companies could fall victim to ransomware or DDoS attacks. This means that, as an organization, we shouldn't attempt to reinvent the wheel, as other organizations may have thought of the same problems before (and found a solution).

An excellent source of incident response playbooks is the "Incident Response Methodologies" developed by CERT Societe General, which are available on https://github.com/certsocietegenerale/IRM. The playbooks were designed with the SANS 6-step incident response process in mind, so they fit perfectly with the overall incident response process. They are currently available in a "cheat sheet format" in English, Spanish and Russian.

Over 15 playbooks are currently available, covering some of the most common incident types including: Phishing, ransomware, DDoS...

As an organization, it is recommended to take these playbooks as a basis and further tailor them to your organization!

Incidence Response Process – Example IRM Layout for Ransomware

The screenshot above is the example layout of an IRM for ransomware. You will notice the first two steps of the incident response as defined by SANS!

Step 2 Identification

Q | Prior to the incident response process kicking off, the security monitoring (or threat hunting) capability has identified an incident and alerted the CIRT.

● | As a first step, the CIRT will validate the incident, collect information and attempt to perform an initial scoping of the incident.

⊕ | It is highly likely that the scoping of the incident will evolve as additional analysis is performed (this is to be expected).

Incidence Response Process – Identification

The second step in the incident response process is "Identification."

Prior to the incident response process kicking off, the security monitoring (or threat hunting) capability has identified an incident and alerted the CIRT. For example, take a file server that is experiencing performance issues: It is abnormally slow. It is not unheard of that system administrators attribute performance issues to undetected malware on a system when they do not have a readily available explanation for the performance issues. In such a case, further investigation is required to determine if the performance issue is indeed due to undetected malware running on the server and consuming a significant amount of CPU resources.

Usually, the identification process starts with an anomaly: Operations in the organization deviate from normal routine and a CIRT team member is informed of this deviation. It must be assessed if this deviation is large enough to identify it as an incident. As the example with a slow file server shows, the deviation might not be due to malware, but because of a larger than usual load on the server. If this is, for example, due to a user that is transferring a large amount of files, then it will not be considered as an incident that has to be handled by the CIRT team.

This does not necessarily mean that no action must be undertaken, but it does not fall under the incident response process: The incident response stops with this step. Although if this happens regularly, then Step 6 (Lessons Learned) might be taken to reduce the amount of false alerts.

Determining the scope of an incident is also part of this phase.

Step 3 Containment

TTP	During containment, the CIRT will attempt to **further analyze the incident**, while **preventing bad things from happening**. This does not necessarily mean that affected systems are "disconnected" from the network: Systems could be left online, while the adversary's actions are further observed!
✓	While containing the incident, the CIRT should ensure that: • The adversary does not inflict additional damage. • The adversary does not realize he's been spotted. • The CIRT should attempt to obtain a clear overview of all compromised systems.

The containment phase is a crucial aspect of successful incident response!

Incidence Response Process – Containment

The third step in the incident response process is "Containment." Containment is often misunderstood as being synonymous with immediately disconnecting an infected systems. This doesn't necessarily have to be the case:

During containment, the CIRT will attempt to further analyze the incident, while preventing bad things from happening. This could mean that a system is left online, but will now be subject to increased monitoring, in order to obtain more information on the incident:

- What are the objectives of the adversary?
- How did they initially enter the environment?
- What systems have already been compromised?
- When are the adversaries active?
- ...

While observing the adversaries, it is vital to ensure that:

- The adversaries are not able to inflict additional damage to the affected organization.
- The adversaries don't realize they have been spotted (as they will roll back any activity, destroy traces or even perform destructive activities).

We can compare this to the analogy of a terrorist cell discovered by the FBI. Upon discovery, it is likely that the FBI will increase monitoring to obtain more information on what they are trying to do, as this could be a treasure of threat intelligence. At the same time, they should ensure they can intervene when the threat moves to actually do something bad.

Step 4 — Eradication

- Eradication is the step where the adversary is effectively removed from the environment.

- Eradication is the step where many incident response operations go bad: Some infected systems are forgotten; thus the intrusion starts all over again...

- Proper eradication involves a large, coordinated effort where all infected systems are "cleaned" **AT THE SAME TIME.**

Incidence Response Process – Eradication
The fourth step in the incident response process is "Eradication."

Eradication is the step where the adversary is effectively removed from the environment. This is a highly critical phase of the incident response process, as at this point, the adversary will definitely become aware of the fact that he's been spotted.

Eradication is the step where many incident response operations go bad: Some infected systems are forgotten; thus the intrusion starts all over again... Proper eradication of an advanced adversary will require a large, coordinated effort. The CIRT should coordinate with business and IT when actions are taken. In order not to "show your hand" to the adversary, it's important that all systems are cleaned at the same time!

Step 5 Recovery

In the recovery step, systems are reintroduced in the production environment, in order to continue normal operations.

Care should be taken to move only to this step when the incident has been fully eradicated.

Upon recovery, "cleaned" systems are typically subject to increased monitoring to ensure they are not "reinfected."

Incidence Response Process – Recovery
The fifth step in the incident response process is "Recovery."

In the recovery step, systems are reintroduced in the production environment, in order to continue normal operations.

Care should be taken to move only to this step when the incident has been fully eradicated, and when it is certain that the incident will not happen again if the systems are reintroduced in production.

Upon recovery, most organizations will perform increased monitoring on "cleaned" systems, as they want to ensure the systems are not "reinfected." This monitoring can be highly tailored, as proper analysis of the incident should have resulted in numerous IOCs and TTPs employed by this particular adversary.

Step 6 **Lessons Learned**

Lessons Learned is one of the most important steps of incident response. This is true for the CIRT team, but for the rest of the organization as well!

The goal of this step for the organization is to prevent reoccurrence of similar incidents AND to improve defenses based upon this actual incident.

Throughout the entire IR process, the CIRT team should have generated threat intelligence (IOCs and TTPs) it can now use to perform additional hunting in the environment! This is what we call the "continuous loop"!

Incidence Response Process – Lessons Learned
The sixth and last step in the incident response process is "Lessons Learned."

Lessons Learned is one of the most important steps of incident response. This is true for the CIRT team, but for the rest of the organization as well!

The goal of this step for the organization is to prevent reoccurrence of similar incidents. Incidents must be analyzed to determine the root cause of the incident, and action must be taken accordingly. This analysis is not necessarily to be done solely by the CIRT team; it can be supported by other teams as well.

The goal of this step for the organization is to prevent reoccurrence of similar incidents AND to improve defenses based upon this actual incident.

If the CIRT feels they haven't collected enough information during the previous phases of the incident response activities, it could be worth doing additional analysis. Throughout the entire IR process, the CIRT team should have generated threat intelligence (IOCs & TTPs) it can now use to perform additional hunting in the environment! This is what we call the "continuous loop"!

When to Use the Incident Response Process?

The six steps of incident response as defined by SANS can actually be applied to computer incidents in general: Security related and non-security related. Incidents handled according to the SANS incident response process are security incidents. But security incidents are not the only incidents that occur inside an organization.

As a first remark, it's important to note that incident response should be a cyclical process, including continuous improvement. Once an incident is handled, a big focus should be placed on the "Lessons Learned" phase, where the organization focuses on understanding how similar incidents can be better prevented, detected, AND responded to in the future.

This very idea means that incident response will feed into other processes, such as threat hunting and security monitoring. When the malware that is analyzed during an incident is fully reversed, this will probably lead to IOCs and TTPs that can now be used as additional input in the threat hunting process. Another easier example could be the identification of a "low risk" vulnerability as the root cause of a serious incident, which could provide input to the overall security strategy of the organization, thereby challenging current risk ratings.

Finally, not every incident is the same, which means different incidents could require a tailored approach, where different elements of the incident response process will receive additional attention.

Today, many tools exist in the industry that can support CIRT team members in every stage of the incident response process. This includes both open-source and commercial tools. Some interesting examples of excellent free tools include:

SANS DFIR — DIGITAL FORENSICS & INCIDENT RESPONSE

The free **SANS SIFT** workstation, used by the vast majority of IR teams.

"Volatility" for memory forensics

The **"Autopsy ®"** & **Sleuth Kit®"** toolkits.

GRR RAPID RESPONSE

"GRR" for remote acquisition and analysis

ASSEMBLYLINE

"Assemblyline" is a free malware detection and analysis framework.

TheHive

"TheHive" is an IR collaboration framework.

Cortex

"Cortex" is an extension to TheHive for observable analysis.

The **"Kansa"** PowerShell IR framework

Incident Response – Tools, Tools, Tools...

Today, many tools exist in the industry that can support CIRT team members at every stage of the incident response process. This includes both open-source and commercial tools. Some interesting examples of excellent free tools include:

- The SIFT Workstation is a group of free open-source incident response and forensic tools designed to perform detailed digital forensic examinations in a variety of settings. It can match any current incident response and forensic tool suite. SIFT demonstrates that advanced incident response capabilities and deep dive digital forensic techniques to intrusions can be accomplished using cutting-edge open-source tools that are freely available and frequently updated. SIFT is available at https://digital-forensics.sans.org/community/downloads.
- Volatility is an advanced memory forensics framework. It is considered to be the "go-to" tool for memory forensics. An alternative to Volatility is Rekall, although it offers fewer analysis modules. More information on Volatility can be found here: https://www.volatilityfoundation.org/.
- The Sleuth Kit® is a collection of command line tools and a C library that allows you to analyze disk images and recover files from them. It is used behind the scenes in Autopsy and many other open source and commercial forensics tools. The Sleuth Kit is available at https://www.sleuthkit.org/.
- GRR is a tool for live forensic response, as in that, it is used while the user is still active on the machine. GRR uses a Python agent on the machine, that talks over the internet to a GRR Python server. There is no need for a VPN. Built into the tool is a disk forensics capability Sleuth Kit and a memory forensics capability in the form of Rekall. You can get GRR here: https://github.com/google/grr.
- Assemblyline is a platform for the analysis of malicious files. It is designed to assist IR teams to automate the analysis of files and to better use the time of security analysts. The tool recognizes when a large volume of files is received within the system and can automatically rebalance its workload. Users can add their own analytics, such as antivirus products or custom-built software, into Assemblyline. The tool is designed to be customized by the user and provides a robust interface for security analysts. You can get Assemblyline here: https://cyber.gc.ca/en/assemblyline.

- TheHive is a scalable 3-in-1 open source and free security incident response platform-designed to make life easier for SOCs, CSIRTs, CERTs and any information security practitioner dealing with security incidents that need to be investigated and acted upon swiftly. Collaboration is at the heart of TheHive. Multiple analysts can work on the same case simultaneously. For example, an analyst may deal with malware analysis while another may work on tracking C2 beaconing activity on proxy logs as soon as IOCs have been added by their coworker. Using TheHive's live stream, everyone can keep an eye on what's happening on the platform, in real time. TheHive is available at https://github.com/TheHive-Project/.
- Cortex, an open-source and free software, has been created by TheHive Project to facilitate the analysis of different observables. Observables, such as IP and email addresses, URLs, domain names, files or hashes, can be analyzed one by one or in bulk mode using a Web interface. Analysts can also automate these operations, thanks to the Cortex REST API. As they are developed by the same team, Cortex supports excellent integration with TheHive! You can download Cortex here: https://github.com/TheHive-Project/TheHive.
- Finally, Kansa is a modular incident response framework in PowerShell. It's been tested in PSv2 / .NET 2 and later and works "mostly without issue." You can download Kansa here: https://github.com/davehull/Kansa.

SANS SIFT Workstation

- Free, continuously updated, Ubuntu-based VMware appliance
- Excellent base workstation for IR teams and analysts
- Includes tools for virtually all phases of the IR process
- Used by virtually all IR teams and many of the SANS DFIR courses!

SANS SIFT Workstation

The SIFT Workstation is a group of free open-source incident response and forensic tools designed to perform detailed digital forensic examinations in a variety of settings. It can match any current incident response and forensic tool suite. SIFT demonstrates that advanced incident response capabilities and deep dive digital forensic techniques to intrusions can be accomplished using cutting-edge open-source tools that are freely available and frequently updated.

Rob Lee and his team created and continually update the SIFT Workstation. It's successfully used for incident response and digital forensics and is available to the community as a public service. With over 100,000 downloads to date, SIFT continues to be the most popular open-source incident-response and digital forensic offering next to commercial source solutions.

Offered as an open source and free project, SIFT Workstation is taught in the following incident response courses at SANS:

- Advanced Digital Forensics, Incident Response, and Threat Hunting (FOR508)
- Advanced Network Forensics (FOR572)
- Cyber Threat Intelligence (FOR578)
- Advanced Memory Forensics & Threat Detection (FOR526)

While attackers often leave their footprints on disk, it is sometimes easier (or even required!) to spot malicious behavior in memory. The Volatility framework is an open collection of tools that allows you to investigate and extract digital artefacts from a memory image.

Volatility is the "gold standard" for memory forensics and supports a variety of file formats such as:

- Raw linear sample (dd)
- Hibernation file
- Crash dump file
- VMware saved state and snapshot file
- EWF format (E01)

Volatility – Memory Forensics (1)

While attackers often leave their footprints on disk, it is sometimes easier (or even required) to spot malicious behavior in memory. The Volatility framework is an open collection of tools that allows you to investigate and extract digital artefacts from a memory image. The Volatility framework supports the investigation of memory images of a wide variety of operating systems, including 32- and 64-bit Windows machines as of Windows XP and Server 2003 up until Windows 10 and Server 2016, various 32- and 64-bit Linux kernels and Mac OSX 10.5 up until 10.12.

Next to the variety of supported operating systems, you can also analyze different file formats such as:

- Raw linear sample (dd)
- Crash dump file
- EWF format (E01)
- Hibernation file
- VMware saved state and snapshot file

Volatility usage:
Volatility.exe –f [image] [plugin] --profile=[profile]

Using the "imageinfo" plugin, volatility scans the memory to determine the profile to be used.

```
c:\demo>volatility\volatility.exe imageinfo -f stuxnet.vmem
Volatility Foundation Volatility Framework 2.6
INFO     : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (C:\demo\stuxnet.vmem)
                      PAE type : PAE
                           DTB : 0x319000L
                          KDBG : 0x80545ae0L
          Number of Processors : 1
     Image Type (Service Pack) : 3
                KPCR for CPU 0 : 0xffdff000L
             KUSER_SHARED_DATA : 0xffdf0000L
            Image date and time : 2011-06-03 04:31:36 UTC+0000
      Image local date and time : 2011-06-03 00:31:36 -0400
```

Volatility – Memory Forensics (2)

As nuances (or large portions) of the structure of memory changes between operating systems and operating system versions, Volatility requires you to provide a profile in order to know which data structures, algorithms and symbols to use. By default (and thus without providing the profile flag) the WinXPSP2x86 profile is set. As such for each memory dump of an operating system that doesn't match Windows XP SP2 x86, a profile must be provided using the --profile command line flag.

A typical usage of the volatility command line is the following:
> Volatility.exe –f [image] [plugin] --profile=[profile]

In case you don't know the exact system the memory dump was taken from you can use the "imageinfo" command line parameter. The output of this command will provide a suggestion to what profile should be used. Do note that the "imageinfo" plugin can only be used for Windows operating systems.

If you want to see a list of supported profile names, do the following:
> Volatility.exe --info

```
c:\demo>volatility\volatility.exe -f stuxnet.vmem pslist
Volatility Foundation Volatility Framework 2.6
Offset(V)   Name              PID   PPID  Thds  Hnds  Sess  Wow64 Start                          Exit
----------  ----------------  ----  ----  ----  ----  ----  ----- -----------------------------  -----------------------------
0x823c8830  System            4     0     59    403   ----- 0
0x820df020  smss.exe          376   4     3     19    ----- 0     2010-10-29 17:08:53 UTC+0000
0x821a2da0  csrss.exe         600   376   11    395   0     0     2010-10-29 17:08:54 UTC+0000
0x81da5650  winlogon.exe      624   376   19    570   0     0     2010-10-29 17:08:54 UTC+0000
0x82073020  services.exe      668   624   21    431   0     0     2010-10-29 17:08:54 UTC+0000
0x81e70020  lsass.exe         680   624   19    342   0     0     2010-10-29 17:08:54 UTC+0000
0x823315d8  vmacthlp.exe      844   668   1     25    0     0     2010-10-29 17:08:55 UTC+0000
0x81db8da0  svchost.exe       856   668   17    193   0     0     2010-10-29 17:08:55 UTC+0000
0x81e61da0  svchost.exe       940   668   13    312   0     0     2010-10-29 17:08:55 UTC+0000
0x822843e8  svchost.exe       1032  668   61    1169  0     0     2010-10-29 17:08:55 UTC+0000
0x81e18b28  svchost.exe       1080  668   5     80    0     0     2010-10-29 17:08:55 UTC+0000
0x81ff7020  svchost.exe       1200  668   14    197   0     0     2010-10-29 17:08:55 UTC+0000
0x81fee8b0  spoolsv.exe       1412  668   10    118   0     0     2010-10-29 17:08:56 UTC+0000
0x81e0eda0  jqs.exe           1580  668   5     148   0     0     2010-10-29 17:09:05 UTC+0000
0x81fe52d0  vmtoolsd.exe      1664  668   5     284   0     0     2010-10-29 17:09:05 UTC+0000
0x821a0568  VMUpgradeHelper   1816  668   3     96    0     0     2010-10-29 17:09:08 UTC+0000
0x8205ada0  alg.exe           188   668   6     107   0     0     2010-10-29 17:09:09 UTC+0000
0x820ec7e8  explorer.exe      1196  1728  16    582   0     0     2010-10-29 17:11:49 UTC+0000
0x820ecc10  wscntfy.exe       2040  1032  1     28    0     0     2010-10-29 17:11:49 UTC+0000
0x81e86978  TSVNCache.exe     324   1196  7     54    0     0     2010-10-29 17:11:49 UTC+0000
0x81fc5da0  VMwareTray.exe    1912  1196  1     50    0     0     2010-10-29 17:11:50 UTC+0000
0x81e6b660  VMwareUser.exe    1356  1196  9     251   0     0     2010-10-29 17:11:50 UTC+0000
0x8210d478  jusched.exe       1712  1196  1     26    0     0     2010-10-29 17:11:50 UTC+0000
0x82279998  imapi.exe         756   668   4     116   0     0     2010-10-29 17:11:54 UTC+0000
0x822b9a10  wuauclt.exe       976   1032  3     133   0     0     2010-10-29 17:12:03 UTC+0000
0x81c543a0  Procmon.exe       660   1196  13    189   0     0     2011-06-03 04:25:56 UTC+0000
0x81fa5390  wmiprvse.exe      1872  856   5     134   0     0     2011-06-03 04:25:58 UTC+0000
0x81c498c8  lsass.exe         868   668   2     23    0     0     2011-06-03 04:26:55 UTC+0000
0x81c47c00  lsass.exe         1928  668   4     65    0     0     2011-06-03 04:26:55 UTC+0000
0x81c0cda0  cmd.exe           968   1664  0     ----- 0     0     2011-06-03 04:31:35 UTC+0000   2011-06-03 04:31:36 UTC+0000
0x81f14938  ipconfig.exe      304   968   0     ----- 0     0     2011-06-03 04:31:35 UTC+0000   2011-06-03 04:31:36 UTC+0000
```

Using the plugin "pslist" we can scan the memory for all processes that were running at the time of the memory dump.

Why are there three lsass.exe processes here?

Volatility – Memory Forensics (3)

When analyzing a memory sample, one of the first things usually performed by analysts is to see what processes were running. This information can be obtained by running the "pslist" plugin against the memory image.

Example: volatility.exe –f Stuxnet.vmem pslist

The screenshot on the slide demonstrates the effectiveness of this plugin. The output shows that three lsass.exe processes are running on the system while normally only one lsass.exe process should be running.

Next to the name of the running processes are also the Process ID (PID), Parent Process ID (PPID), the process start time and (when applicable) the process exit time.

```
c:\demo>volatility\volatility.exe -f stuxnet.vmem pstree
Volatility Foundation Volatility Framework 2.6
Name                                          Pid   PPid   Thds   Hnds  Time
-------------------------------------------- ----- ------ ------ ------ ----
 0x823c8830:System                              4      0     59    403 1970-01-01 00:00:00 UTC+0000
. 0x820df020:smss.exe                         376      4      3     19 2010-10-29 17:08:53 UTC+0000
.. 0x821a2da0:csrss.exe                       600    376     11    395 2010-10-29 17:08:54 UTC+0000
.. 0x81da5650:winlogon.exe                    624    376     19    570 2010-10-29 17:08:54 UTC+0000
... 0x82073020:services.exe                   668    624     21    431 2010-10-29 17:08:54 UTC+0000
.... 0x81fe52d0:vmtoolsd.exe                 1664    668      5    284 2010-10-29 17:09:05 UTC+0000
..... 0x81c0cda0:cmd.exe                      968   1664      0 ------ 2011-06-03 04:31:35 UTC+0000
...... 0x81f14938:ipconfig.exe                304    968      0 ------ 2011-06-03 04:31:35 UTC+0000
.... 0x822843e8:svchost.exe                  1032    668     61   1169 2010-10-29 17:08:55 UTC+0000
..... 0x822b9a10:wuauclt.exe                  976   1032      3    133 2010-10-29 17:12:03 UTC+0000
..... 0x820ecc10:wscntfy.exe                 2040   1032      1     28 2010-10-29 17:11:49 UTC+0000
.... 0x81e61da0:svchost.exe                   940    668     13    312 2010-10-29 17:08:55 UTC+0000
.... 0x81db8da0:svchost.exe                   856    668     17    193 2010-10-29 17:08:55 UTC+0000
.... 0x81fa5390:wmiprvse.exe                 1872    856      5    134 2011-06-03 04:25:58 UTC+0000
.... 0x821a0568:VMUpgradeHelper              1816    668      3     96 2010-10-29 17:09:08 UTC+0000
.... 0x81fee8b0:spoolsv.exe                  1412    668     10    118 2010-10-29 17:08:56 UTC+0000
.... 0x81ff7020:svchost.exe                  1200    668     14    197 2010-10-29 17:08:55 UTC+0000
... 0x81c47c00:lsass.exe                     1928    668      4     65 2011-06-03 04:26:55 UTC+0000
.... 0x81e18b28:svchost.exe                  1080    668      5     80 2010-10-29 17:08:55 UTC+0000
.... 0x8205ada0:alg.exe                       188    668      6    107 2010-10-29 17:09:09 UTC+0000
.... 0x823315d8:vmacthlp.exe                  844    668      1     25 2010-10-29 17:08:55 UTC+0000
.... 0x81e0eda0:jqs.exe                      1580    668      5    148 2010-10-29 17:09:05 UTC+0000
.... 0x81c498c8:lsass.exe                     868    668      2     23 2011-06-03 04:26:55 UTC+0000
.... 0x82279998:imapi.exe                     756    668      4    116 2010-10-29 17:11:54 UTC+0000
.. 0x81e70020:lsass.exe                       680    624     19    342 2010-10-29 17:08:54 UTC+0000
 0x820ec7e8:explorer.exe                     1196   1728     16    582 2010-10-29 17:11:49 UTC+0000
. 0x81c543a0:Procmon.exe                      660   1196     13    189 2011-06-03 04:25:56 UTC+0000
. 0x81e86978:TSVNCache.exe                    324   1196      7     54 2010-10-29 17:11:49 UTC+0000
. 0x81e6b660:VMwareUser.exe                  1356   1196      9    251 2010-10-29 17:11:50 UTC+0000
. 0x8210d478:jusched.exe                     1712   1196      1     26 2010-10-29 17:11:50 UTC+0000
. 0x81fc5da0:VMwareTray.exe                  1912   1196      1     50 2010-10-29 17:11:50 UTC+0000
```

Lsass.exe is a process that is started relatively early in the boot of the Windows operating system and should always have winlogon.exe as a parent process.

By using the "pstree" plugin, we can clearly see that two lsass.exe processes have services.exe as a parent process, while only one (the one with the lowest PID) has winlogon.exe as parent process.

Volatility – Memory Forensics (4)

Lsass.exe is a process that is started relatively early in the boot of the Windows operating system and should always have winlogon.exe as a parent process.

By using the "pstree" plugin, we can clearly see that two lsass.exe processes have services.exe as a parent process, while only one (the one with the lowest PID) has winlogon.exe as parent process.

This provides us a clear indication that two of these processes should be further investigated in detail.

Dumping a process from memory:

Volatility.exe –f [image] [plugin] --profile=[profile] procdump –D [location] –p [PID]

```
c:\demo>volatility\volatility.exe -f stuxnet.vmem procdump -D dump\ -p 1928
Volatility Foundation Volatility Framework 2.6
Process(V) ImageBase  Name                  Result
---------- ---------- --------------------- ------
0x81c47c00 0x01000000 lsass.exe             OK: executable.1928.exe
```

Scanning memory using YARA

Volatility.exe –f [image] [plugin] --profile=[profile] yarascan --yara-file=[location]

```
c:\demo>volatility\volatility.exe -f stuxnet.vmem yarascan --yara-file=stuxnet.yar
Volatility Foundation Volatility Framework 2.6
Rule: Stuxnet_Malware_3
Owner: Process lsass.exe Pid 1928
0x009863f8  53 00 48 00 45 00 4c 00 4c 00 33 00 32 00 2e 00   S.H.E.L.L.3.2...
0x00986408  44 00 4c 00 4c 00 2e 00 41 00 53 00 4c 00 52 00   D.L.L...A.S.L.R.
0x00986418  2e 00 00 00 25 00 73 00 25 00 30 00 38 00 78 00   ....%.s.%.0.8.x.
```

Volatility – Memory Forensics (5)

In order to perform additional investigations on a specific process, Volatility provides us with the ability to carve out the specific process from memory. This can be performed using the following command:

Volatility.exe –f [image] [plugin] --profile=[profile] procdump –D [location] –p [PID]

Now, if we already know what we are looking for and we have specific indicators on a malicious process, Volatility allows us to scan the memory dump using YARA rules. This can be performed using the following command:

Volatility.exe –f [image] [plugin] --profile=[profile] yarascan --yara-file=[location]

Autopsy was designed to be an **end-to-end GUI-based forensic analysis platform.** It has different modules that are included out-of-the-box, while others are available from third parties. Some of the modules that are included out of the box include:

- Timeline Analysis - includes an advanced graphical event viewing interface.
- Hash Filtering - used to flag known bad files and ignore known good ones.
- Keyword Search - used to find files related to relevant terms.
- Web Artifacts - allows extraction of history, bookmarks, and cookies from popular browsers.
- Data Carving - used to recover deleted files.
- Multimedia - extracts EXIF data from pictures and videos.
- Indicators of Compromise, scans a computer using STIX.

The **Sleuth Kit is a collection of command-line tools** and a C library that allows you to analyze disk images and recover files from them. It is used behind the scenes in Autopsy and many other open-source and commercial forensics tools.

Autopsy and The Sleuth Kit (by Basis Technology)

Autopsy and The Sleuth Kit were both developed by Basis Technology. Autopsy was designed to be an end-to-end GUI-based forensic analysis platform. It has different modules that are included out-of-the-box, while others are available from third parties. Some of the modules that are included out of the box include:

- Timeline Analysis - includes an advanced graphical event viewing interface.
- Hash Filtering - used to flag known bad files and ignore known good ones.
- Keyword Search - used to find files related to relevant terms.
- Web Artifacts - allows extraction of history, bookmarks, and cookies from popular browsers.
- Data Carving - used to recover deleted files.
- Multimedia - extracts EXIF data from pictures and videos.
- Indicators of Compromise - scans a computer using STIX.

The Sleuth Kit is a collection of command-line tools and a C library that allows you to analyze disk images and recover files from them. It is used behind the scenes in Autopsy and many other open-source and commercial forensics tools.

GRR – GRR Rapid Response

GRR is a tool for live forensic response: It is used while the user is still active on the machine.

GRR works with an agent that is installed on the clients. Central management is performed using a web portal, from which "hunts" and analysis can be performed.

Focus is on forensics on machines that have poor bandwidth and are in remote locations, due to the increase of "homeworkers" and "road warriors."

Cross-platform support (Windows, Linux, OS X), relies on additional tools such as Rekall (for memory forensics) and Sleuth Kit (disk forensics).

GRR – GRR Rapid Response

GRR is a forensic tool. The name itself is a recursive acronym. The first G stands for GRR and not for Google as many people think. RR stands for Rapid Response.

GRR is a tool for live forensic response: It is used while the user is still active on the machine. GRR uses a Python agent on the machine that talks over the internet to a GRR Python server. There is no need for a VPN. Built into the tool is a disk forensics capability Sleuth Kit and a memory forensics capability in the form of Rekall. Both of the tools are open-source.

It is a free, open-source tool for live forensics. It is cross-platform, it can support the following operating systems:

- Windows
- Linux
- OS X

GRR is a powerful forensic tool that has many interesting features for incident handlers. As a live forensic tool over the network, GRR can be used to perform a forensic investigation on a machine that is live and remote. The Python agent must be installed on the machine to be investigated, and then live remote forensics can be done.

GRR is able to do forensics on machines that have poor bandwidth and are on a remote location. More and more "homeworkers" and "road warriors" are becoming a concern for organizations, which is something GRR is hoping to address.
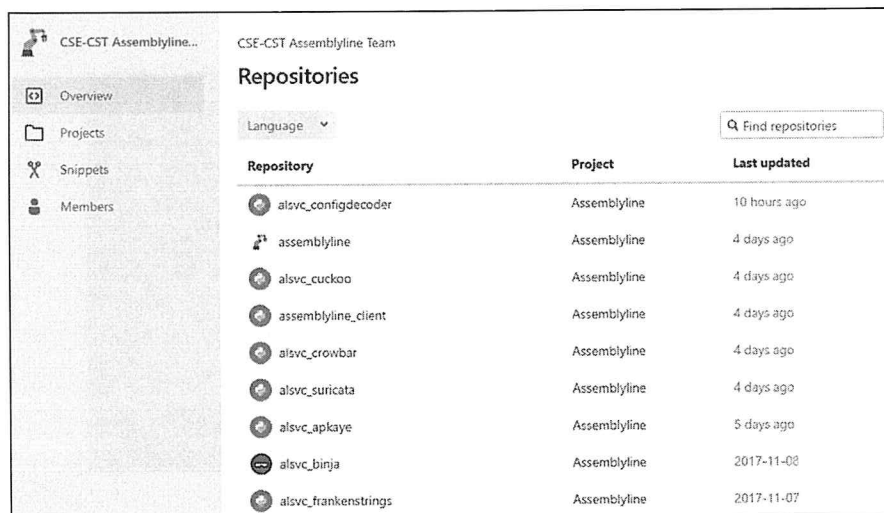
GRR can investigate a large number of machines for known IOCs. Like we showed with Loki, GRR can be provided with a list of IOCs (like cryptographic hashes of malicious files) that are then searched for through all machines under control of GRR. The list of IOCs is downloaded by the agent, who then performs the search for IOCs.

GRR can do remote forensic acquisition of machines, for example, to provide an inventory of all files on the file system of the machine under forensic investigation.

GRR is capable of performing queries on multiple machines to find a program by filename. If, for example, you know that advanced attackers produced a file for data exfiltration (e.g. attacktool.exe), then GRR can search through all your machines looking for files with this name.

GRR – Screenshot of the Interface

In the above screenshot, we can see GRR being used to investigate the filesystem of a remote machine where the GRR agent has been deployed. All the way to the left of this screenshot, we can see a menu that starts with the name of the machine together with its IP address. After we retrieved the flow, we selected "Browse Virtual Filesystem." This gives us access to the forensic data retrieved from the machine, under the form of:

- Memory
- Filesystem
- Registry

We selected drive C: in the treeview (fs / os / C:).
This gives us an overview of the files and directories present in the root of drive C:.

We selected file BOOTSECT.BAK with size 8192 bytes dating from 2013.

From the hex / ASCII dump at the bottom of the screenshot, we can see the content of the boot sector backup file.

Assemblyline

CSE-CST Assemblyline...

- Overview
- Projects
- Snippets
- Members

CSE-CST Assemblyline Team

Repositories

Language ∨		Q Find repositories
Repository	**Project**	**Last updated**
alsvc_configdecoder	Assemblyline	10 hours ago
assemblyline	Assemblyline	4 days ago
alsvc_cuckoo	Assemblyline	4 days ago
assemblyline_client	Assemblyline	4 days ago
alsvc_crowbar	Assemblyline	4 days ago
alsvc_suricata	Assemblyline	4 days ago
alsvc_apkaye	Assemblyline	5 days ago
alsvc_binja	Assemblyline	2017-11-08
alsvc_frankenstrings	Assemblyline	2017-11-07

Assemblyline was released by Canada's CSE (Communications Security Establishment).

It's a malicious file analysis tool, focused strongly on modularity and ability to analyze large volumes of files.

Some interesting modules it includes are Cuckoo, Suricata, YARA...

Assemblyline

Assemblyline is a malicious file analysis tool that was released in October 2017 on BitBucket by Canada's CSE (Communications Security Establishment). Its purpose is to have one tool that can be used to analyze large volumes of potentially malicious files, thereby limiting the workload on the analyst. Assemblyline consists of one central engine that uses different modules to perform analysis, in order to reach an overall "scoring" of files.

An interesting example of how Assemblyline can be used is quoted on its official website:

"A financial officer receives an email from an outside sender that includes a password-protected .zip file that contains a spreadsheet and a Word document with text for an annual report. An hour later the financial officer forwards that email to three colleagues within the department and attaches a .jpeg image of a potential cover for the report.

Assemblyline will start by examining the initial email. It automatically recognizes the various file formats (email, .zip file, spreadsheet, Word document) and triggers the analysis of each file. In this example, the Word document contains embedded malware, although the financial officer is unaware of this. The whole file is given a score when the analysis of each file is complete. Scores over a certain threshold trigger alerts, at which point a security analyst may manually examine the file. The malware within the Word document is neutralized due to further security measures that the organization has already implemented.

When the email is forwarded, Assemblyline automatically recognizes the duplication of files and focuses on new content that may be part of the email, such as the .jpeg image."

TheHive

TheHive (by CERT-BDF) is an open-source incident response framework that focuses on three core pillars:

- Collaborate – multiple SOC and CERT analysts can simultaneously work on an investigation and collaborate through the platform.

- Elaborate – TheHive allows you to create flows and templates to speed up and automate tedious tasks.

- Analyze – TheHive tightly integrates with MISP, which allows for bi-directional communication. The platform allows for quick triage and filtering of IOCs.

TheHive also has a direct integration with Cuckoo Sandbox.

TheHive

It is often so that during investigations multiple teams are collaborating, each of which has their own insights and log sources. In order to share this information and knowledge, it is good to have a platform that allows you to create different cases in which IOCs and case notes can be shared.

TheHive is an open-source and free software released under the AGPL (Affero General Public License) by CERT-BDF. The incident response framework focuses on three core pillars:

- Collaborate – Multiple SOC and CERT analysts can simultaneously work on an investigation and collaborate through the platform.
- Elaborate – TheHive allows you to create flows and templates to speed up and automate tedious tasks.
- Analyze – TheHive tightly integrates with MISP, which allows for bi-directional communication. The platform allows for quick triage and filtering of IOCs.

TheHive also has a direct integration with Cuckoo Sandbox.

More information on TheHive can be found at https://thehive-project.org/.
An introduction to TheHive can be found at https://blog.thehive-project.org/2016/11/07/introducing-thehive/.

Cortex (by CERT-BDF) is an extension to TheHive.

Its focus is on "observable analysis" (IP, URL, email address, files...)

One central platform where an observable can be submitted, after which it is analyzed by "analyzers".

Cortex

Cortex is an extension to TheHive (by CERT-BDF) that tries to solve a common problem frequently encountered by analysts during threat intelligence, digital forensics and incident response: How to analyze observables they have collected, at scale, by querying a single tool instead of several? In some ways, Cortex can be compared to tools like Assemblyline or even IRMA (Incident Response & Malware Analysis).

Something that makes Cortex highly powerful is its excellent integration with TheHive, allowing for easy submission of samples and reporting of analyzer outputs.
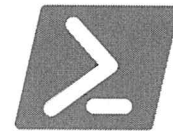
Kansa is a modular incident response framework in PowerShell created by Dave Hull. Next to incident response, it could also be used to obtain endpoint information during threat hunting!

It uses PowerShell Remoting to run modules on hosts within the network to collect data that can be used during incident response engagements, hunts or baseline creation. Kansa is modular. It has a core script, a multitude of data collection modules and different analysis scripts to help you parse the data collected.

Data collection modules include:
- Get-SchedTasks
- Get-Autorunsc
- Get-SvcAll
- Get-LocalAdmins
- Get-MasterFileTable
- Get-IOCsByPath
- Get-RdpConnectionLogs

```
C:\demo\Kansa-master>powershell .\kansa.ps1 -Target $env:COMPUTERNAME -ModulePath .\Modules -Verbose
VERBOSE: Found .\Modules\Modules.conf.
VERBOSE: Running modules:
Get-PrefetchListing
Get-WMIRecentApps
Get-Netstat
Get-DNSCache
Get-ProcsWMI
Get-LogUserAssist
Get-SvcFail
Get-SvcTrigs
Get-WMIEvtFilter
Get-WMIFltConBind
Get-WMIEvtConsumer
Get-PSProfiles
Get-SchedTasks
Get-File
Get-LocalAdmins
```

Kansa – A PowerShell IR Framework
Kansa is a modular incident response framework in PowerShell created by Dave Hull.

It uses PowerShell Remoting to run modules on hosts within the network to collect data that can be used during incident response engagements, hunts or baseline creation.

Kansa is modular. It has a core script, a multitude of data collection modules and different analysis scripts to help you parse the data collected.

Data collection modules include among others:

- Get-SchedTasks
- Get-Autorunsc
- Get-SvcAll
- Get-LocalAdmins
- Get-MasterFileTable
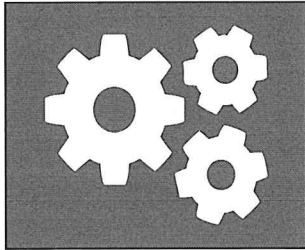- Get-IOCsByPath
- Get-RdpConnectionLogs

Kansa can be found here: https://github.com/davehull/Kansa.

So... We talked about a bunch of tools! An interesting example of how these tools can be used in conjunction is the following:

- Finding a malicious executable in Volatility.
- Dumping the malicious executable using Volatility.
- Creating a YARA rule and scanning the environment for additional samples!

Developing good YARA rules require initial analysis of the samples and fine-tuning cycles to find the right level of (true) positive detection.

To reduce the cost and time of this process, YARA rules can be created automatically, based on samples, and subsequently adapted for better results.

Putting It Together – An Example

So, we talked about a bunch of tools! An interesting example of how these tools can be used in conjunction is the following:

- Finding a malicious executable in Volatility.
- Dumping the malicious executable using Volatility.
- Creating a YARA rule and scanning the environment for additional samples!

Developing YARA rules is a skill; it requires analysis of the sample(s) to come up with an initial rule, and then several cycles of fine tuning to arrive at a right level of detection. This process can be time-consuming and requires specific skills that are not easily found on the market. To reduce the cost of this process, and speed up the development of YARA rules, YARA rule generators can be used.

A YARA rule generator is a program that takes one or more samples as input, performs an analysis of the samples, and generates a YARA rule to detect the samples and similar samples. It is important that the generator create a rule that is not too specific or generic. A rule that is too specific, will only detect the samples that we analyzed, and this can be better done by cryptographic hash matching. A rule that is too generic will generate too many (false) positive detections.

Generated YARA rules are like handcrafted YARA rules: They are contained in text files and are subject to further modifications and tuning. There are several online and offline, free YARA rule generators. The advantage of online generators is that they don't require the installation of programs, but they implicitly share a sample with an online service, which is not always desirable in the case of malware created by advanced adversaries, as it might give away our intentions.

Offline generators don't have this disadvantage, but some of them require many dependencies to be installed, as most of them are Python programs.

Generating YARA Rules Automatically – Tools

Yara Rule Generator

```
C:\Demo>yaraGenerator.py -r wcry_gen -f exe malware\

[+] Generating Yara Rule wcry_gen from files located in: malware\

[+] Yara Rule Generated: wcry_gen.yar

  [+] Files Examined: ['84c82835a5d21bbcf75a61706d8ab549']
  [+] Author Credited: Anonymous
  [+] Rule Description: No Description Provided
[+] YaraGenerator (C) 2013 Chris@xenosec.org https://github.com/Xen0ph0n/YaraGenerator

C:\Demo>
```

Tools:
- **yarGen**
- **YaraGenerator**
- **http://yara-generator.net/**

Online generation

Offline generation

Generating YARA Rules Automatically – Tools

yarGen and YaraGenerator are offline YARA rule generators both written in Python. YaraGenerator has no dependencies, while yarGen has a lot of dependencies.

Both can be found on GitHub:
https://github.com/Neo23x0/yarGen
https://www.joesandbox.com/

Using these tools is rather simple: Run the command-line tool with a few options (for example, a name for the rule to be generated), and provide a sample.

Compared to yarGen, YaraGenerator is rather simple. It will search for strings and use these to generate a rule. While yarGen will also look for binary code to generate the rule. yarGen has an optional online component (Binarly).

Online website yara-generator (http://yara-generator.net) is not affiliated with YaraGenerator, but with the JoeSandbox online malware analysis service. Samples need to be submitted to the online service, which will generate a YARA rule to be downloaded.

Generating YARA Rules Automatically – An Example


```
import "pe"

rule sampe_84c82835a5d21bbcf75a61706d8ab549_vir_0 {
meta:
    description = "Auto-generated simple rule for f
    author = "Joe Sandbox Yara Rule Generator 1.0.0"
    yaraversion = ">= 3.2.0"
...
strings:
    // Relevance: 6.0 Appears at: 76d0
    $s0 = "file error" fullword nocase wide ascii
    // Relevance: 6.0 Appears at: 754f
    $s1 = "cmd.exe /c \"%s\"" fullword nocase wide ascii
    // Relevance: 3.0 Appears at: 75f2
    $s2 = "tasksche.exe" fullword nocase wide ascii
    // Relevance: 3.0 Appears at: 7699
    $s3 = "buffer error" fullword nocase wide ascii
    // Relevance: 3.0 Appears at: 7633
    $s4 = "attrib +h ." fullword nocase wide ascii
    // Relevance: 3.0 Appears at: 78f6
    $s5 = "%s%s%s" fullword nocase wide ascii
    // Relevance: 3.0 Appears at: 76c5
    $s6 = "stream error" fullword nocase wide ascii
    // Relevance: 3.0 Appears at: 76b8
...
    // Opcode sequence from the PE entrypoint
    $entrypointOpCode = { 55 8B ?? 6A ?? 68 ?? ?? ?? ?? 68 ?? ?? ?? ?? }
condition:
    all of them
    // PE Section check
    and pe.sections[0].name contains ".text"
    and pe.sections[1].name contains ".rdata"
    and pe.sections[2].name contains ".data"
    and pe.sections[3].name contains ".rsrc"
    // PE Image characteristic check
    and pe.characteristics & pe.LOCAL_SYMS_STRIPPED
    and pe.characteristics & pe.EXECUTABLE_IMAGE
    and pe.characteristics & pe.LINE_NUMS_STRIPPED
    and pe.characteristics & pe.RELOCS_STRIPPED
    // PE subsystem check
    and pe.subsystem & pe.SUBSYSTEM_WINDOWS_GUI
}
```

Metadata

YARA rule generated online

Searching for many strings

Also searching for binary code.

A complex condition, using strings and PE module data.

Generating YARA Rules Automatically – An Example
This is an example of a YARA rule generated online for a Wannacry sample… As can be seen above, it is an extensive rule (parts have been truncated to fit on a slide).
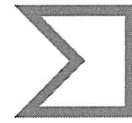
The YARA rule uses the PE module to assist with the analysis of PE files (import "pe"). It contains a metadata section (meta:) with a lot of information about the sample, the generator, and the rule.

Besides searching for strings, the YARA rule also defines binary data ($entrypointOpcode) that corresponds to binary code that will aid in identifying similar samples.

The condition of the rule requires all strings to be found (expression "all of them") and checks several characteristics of the PE file like section names via the PE file module.

VirusTotal provides an interesting YARA-based feature to its commercial customer base: (Retro) hunting!

With the hunting function, users can define a number of YARA rules that are checked for EVERY sample that is uploaded to VirusTotal.

With the retro hunting function, users can search a large data set of old samples (which usually goes back +- 30 days) of uploaded data with a defined set of YARA rules.

For defenders, this could be powerful to detect samples in the wild that are related to malware families previously discovered in the organization!

VirusTotal and Retro Hunting
VirusTotal provides an interesting YARA-based feature to its commercial customer base: (Retro) hunting!

- With the hunting function, users can define a number of YARA rules that are checked for EVERY sample that is uploaded to VirusTotal.
- With the retro hunting function, users can search a large data set of old samples (which usually goes back +- 30 days) of uploaded data with a defined set of YARA rules.

Although a function that is typically used by investigators and malware hunters, this can also be highly powerful for us as enterprise defenders! Once payloads or samples are identified inside the organization, we could develop YARA rules and use these both in our own environment, but also in the wild (on VirusTotal) to detect related malware samples. These related malware samples could provide additional insights in evolving adversary tactics, which allows us to further improve our defenses!

Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- Day 4: Lateral Movement
- **Day 5: Action on Objectives, Threat Hunting & Incident Response**
- Day 6: APT Defender Capstone

This page intentionally left blank.

Please refer to the workbook for further instructions on the exercise!

This page intentionally left blank.

Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- Day 4: Lateral Movement
- **Day 5: Action on Objectives, Threat Hunting & Incident Response**
- Day 6: APT Defender Capstone

SEC599.5

Domain dominance
- Dominating the AD - Basic strategies
- Golden Ticket, Skeleton Key, DCSync and DCShadow
- Detecting domain dominance
- Exercise: Domain dominance

Data exfiltration
- Common exfiltration strategies
- Exercise: Detecting data exfiltration

Leveraging threat intelligence
- Defining threat intelligence
- Exercise: Leveraging threat intelligence with MISP & Loki

Threat Hunting & incident response
- Proactive threat hunting strategies
- Exercise: Hunting your environment using OSQuery
- incident response process
- Exercise: Finding malware using Volatility & YarGen

This page intentionally left blank.

That concludes 599.5! Throughout this section, we've touched upon the following topics:

- We reviewed typical strategies used for "domain dominance".
- We reviewed common data exfiltration strategies.
- We discussed threat intelligence and how it can be generated, shared and consumed using MISP.
- We introduced the concept of threat hunting and how it can be performed using OSQuery.
- Finally, we discussed the incident response process and how a number of these tools can be leveraged during a live investigation!

Tomorrow (SEC599.6), we will put everything together and you will be pitted against an APT that will attempt to infiltrate your environment!

Conclusions for 599.5

That concludes 599.5! Throughout this section, we've touched upon the following topics:

- We reviewed typical strategies used for "domain domination / persistence" strategies.
- We reviewed common data exfiltration strategies.
- We discussed threat intelligence and how it can be generated, shared and consumed using MISP.
- We introduced the concept of threat hunting and how it can be performed using OSQuery.
- Finally, we discussed the incident response process and how a number of these tools can be leveraged during a live investigation!

Tomorrow (SEC599.6), we will put everything together and you will be pitted against an APT that will attempt to infiltrate your environment!

Course Resources and Contact Information

AUTHOR CONTACT
Erik Van Buggenhout
evanbuggenhout@nviso.be
Stephen Sims
ssims@sans.org

SANS INSTITUTE
11200 Rockville Pike
Suite 200
North Bethesda, MD 20852
301.654.SANS (7267)

CYBER DEFENSE CONTACT
Stephen Sims
ssims@sans.org

SANS EMAIL
GENERAL INQUIRIES: info@sans.org
REGISTRATION: registration@sans.org
TUITION: tuition@sans.org
PRESS/PR: press@sans.org

This page intentionally left blank.