599.5
Exfiltration, Cyber Deception, and Incident Response

SANS

THE MOST TRUSTED SOURCE FOR INFORMATION SECURITY TRAINING, CERTIFICATION, AND RESEARCH | sans.org

SEC599.5 | Defeating Advanced Adversaries

SANS

Exfiltration, Cyber Deception, and Incident Response

This page intentionally left blank.

Course Roadmap

- Day 1: Knowing the adversary, knowing yourself
- Day 2: Averting Payload Delivery
- Day 3: Preventing Exploitation
- Day 4: Avoiding Installation, foiling Command & Control & thwarting lateral movement
- **Day 5: Exfiltration, Cyber Deception & Incident Response**
- Day 6: APT Defender Capstone

SEC599.5

Data exfiltration
- Typical data exfiltration strategies
- Exercise: Detecting data exfiltration using Suricata

Cyber deception strategies
- Tricking the adversary
- Exercise: Making your honeypot irresistibly sweet

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

Patrolling your network
- Proactive threat hunting strategies
- Exercise: Hunting your environment using OSQuery / ELK

Incident response
- Incident response process
- Exercise: Responding to an incident using GRR

This page intentionally left blank.

TABLE OF CONTENTS

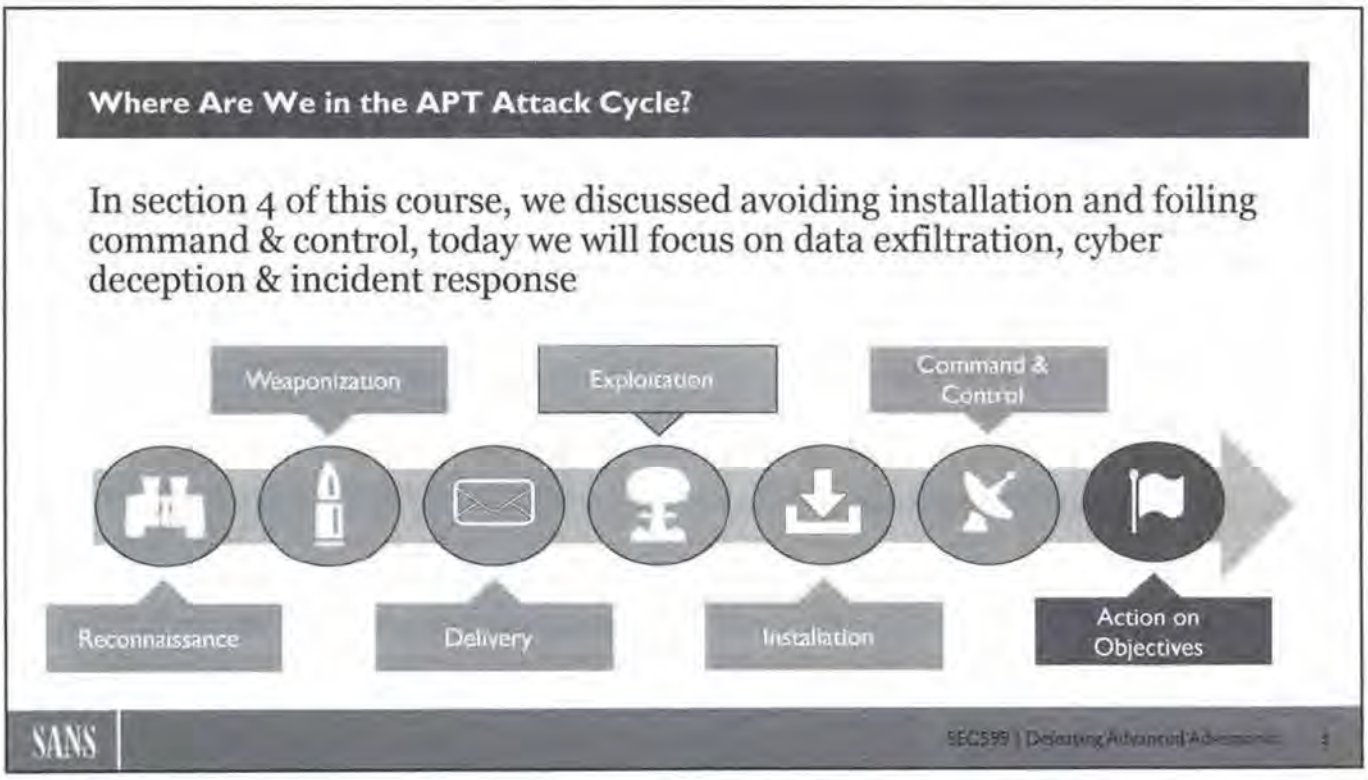
	PAGE
Data exfiltration	06
Typical data exfiltration strategies	06
EXERCISE: Detecting data exfiltration using Suricata	40
Cyber deception strategies	43
Tricking the adversary	46
EXERCISE: Making your honeypot irresistibly sweet	70
Leveraging threat intelligence	73
Defining threat intelligence	74
EXERCISE: Leveraging threat intelligence with MISP & Loki	107
Patrolling your network	109
Proactive threat hunting strategies	110
EXERCISE: Hunting your environment using OSQuery & ELK	132

This page intentionally left blank.

TABLE OF CONTENTS

	PAGE
Incident response	134
Incident response process	135
EXERCISE: Responding to an incident using GRR	165

This page intentionally left blank.



Where Are We in the APT Attack Cycle?
We already started discussing the "Action on Objectives" phase in the APT Attack Cycle yesterday. We will continue our journey today as we focus on data exfiltration, cyber deception & incident response:

Adversary perspective
When adversaries reach their targets through lateral movement, they will "finalize the kill". If the objective is espionage, they will collect and exfiltrate data. If the objective is to interfere with the target, they will start making modifications. This can be corrupting, deleting or overwriting of data and systems, or covertly modify data and configurations to change operations within the target. For example, data modifications can be introduced in payment systems to steal money by wire transfer. We have even observed malware samples that modifies payroll data on cloud systems to introduce new, fake, employees in the staff database and have their wages paid to bank accounts owned by criminals or their money mules.

Defender perspective
When adversaries progress this far in the kill chain, they have defeated the majority of previous defenses. For the adversary, everything is in place for the final push.

Data exfiltration: when the objective is to obtain information, it has to be transferred to the adversaries' systems once it is located and accessed. Exfiltration of data is typically a network activity and as such leaves traces. Large amounts of data exfiltration (gigabytes or terabytes) are detectable by graphing the consumed network bandwidth versus a time axis. Dedicated system can be put in place to monitor for data exfiltration: Data Loss Prevention systems. DLP can be as simple as looking for tags on the network, such as the string "strictly confidential" in uploaded documents. But such simple detections are also simple to bypass. For example, just compressing or encrypting a document before uploading hides all strings inside the document.

Course Roadmap

- Day 1: Knowing the adversary, knowing yourself
- Day 2: Averting Payload Delivery
- Day 3: Preventing Exploitation
- Day 4: Avoiding Installation, foiling Command & Control & thwarting lateral movement
- **Day 5: Exfiltration, Cyber Deception & Incident Response**
- Day 6: APT Defender Capstone

SEC599.5

Data exfiltration
- Typical data exfiltration strategies
- Exercise: Detecting data exfiltration using Suricata

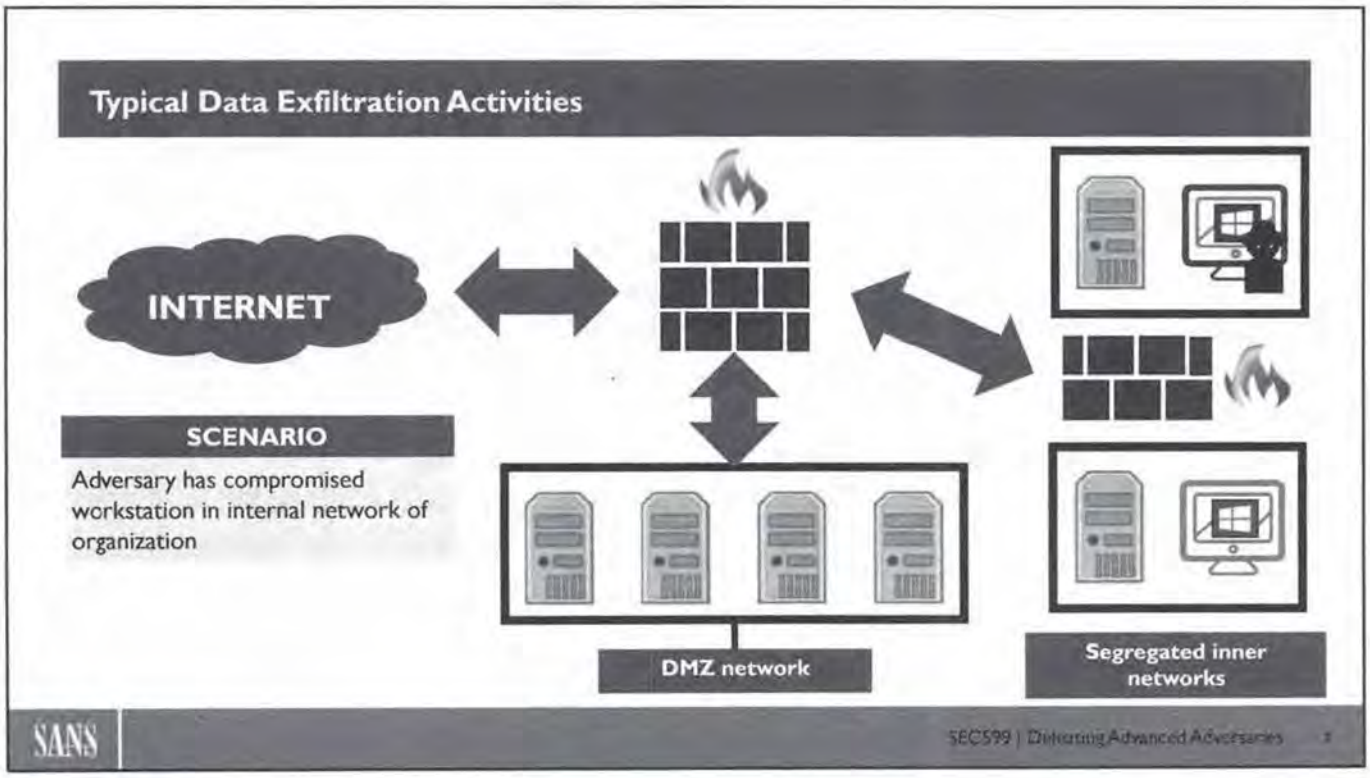
Cyber deception strategies
- Tricking the adversary
- Exercise: Making your honeypot irresistibly sweet

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

Patrolling your network
- Proactive threat hunting strategies
- Exercise: Hunting your environment using OSQuery / ELK

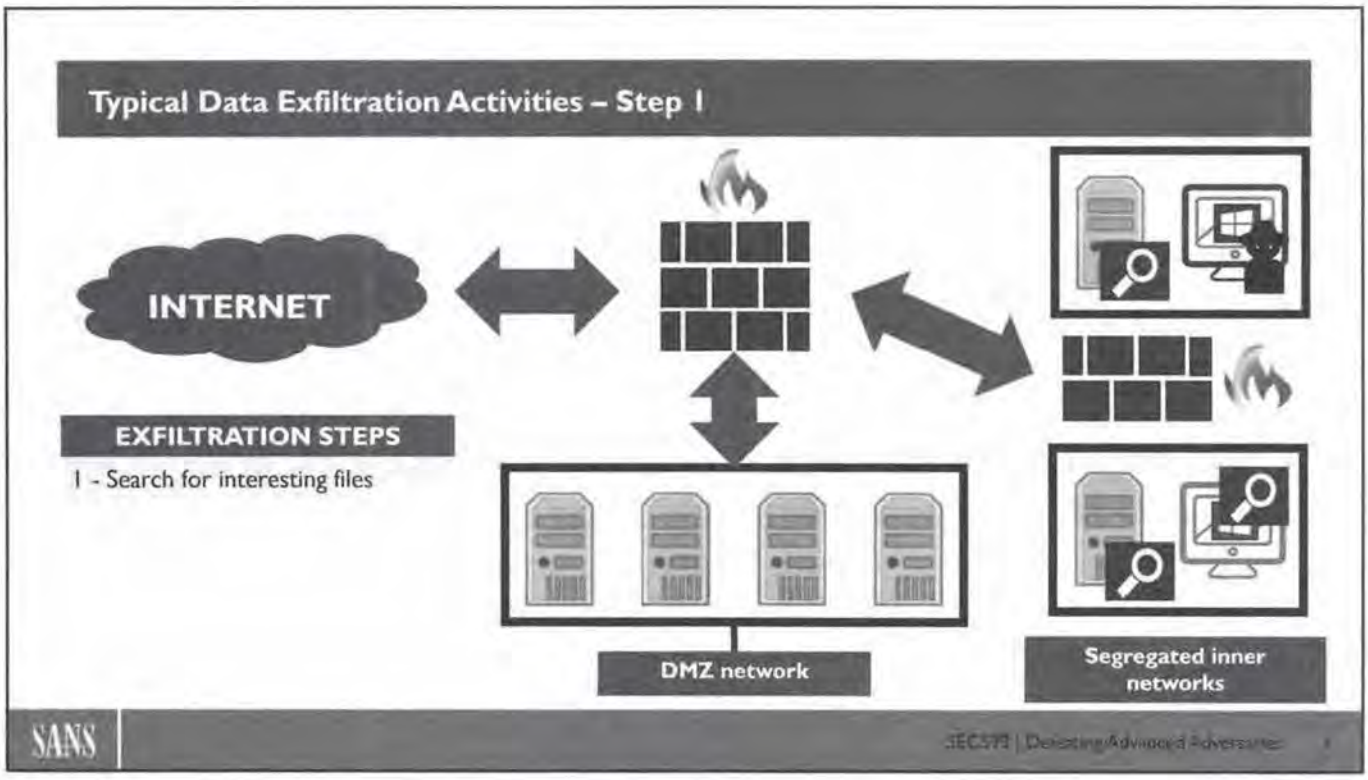
Incident response
- Incident response process
- Exercise: Responding to an incident using GRR

This page intentionally left blank.



Typical Data Exfiltration Activities

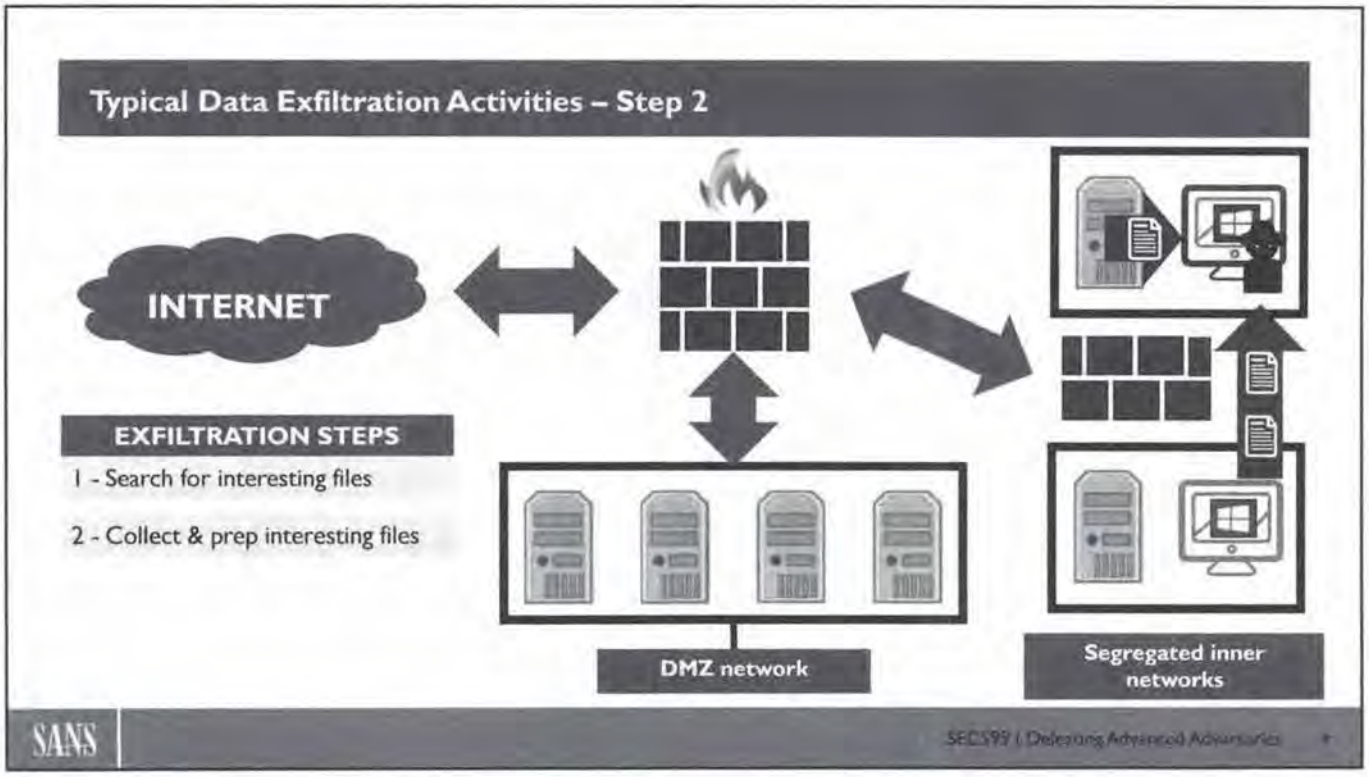
Once (advanced) adversaries have a first foothold in the environment, they will start attempting to reach their objectives. "Actions on objectives" is a broad and generic term, that encompasses many activities performed by attackers. One of these activities is data exfiltration. We will now walk through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

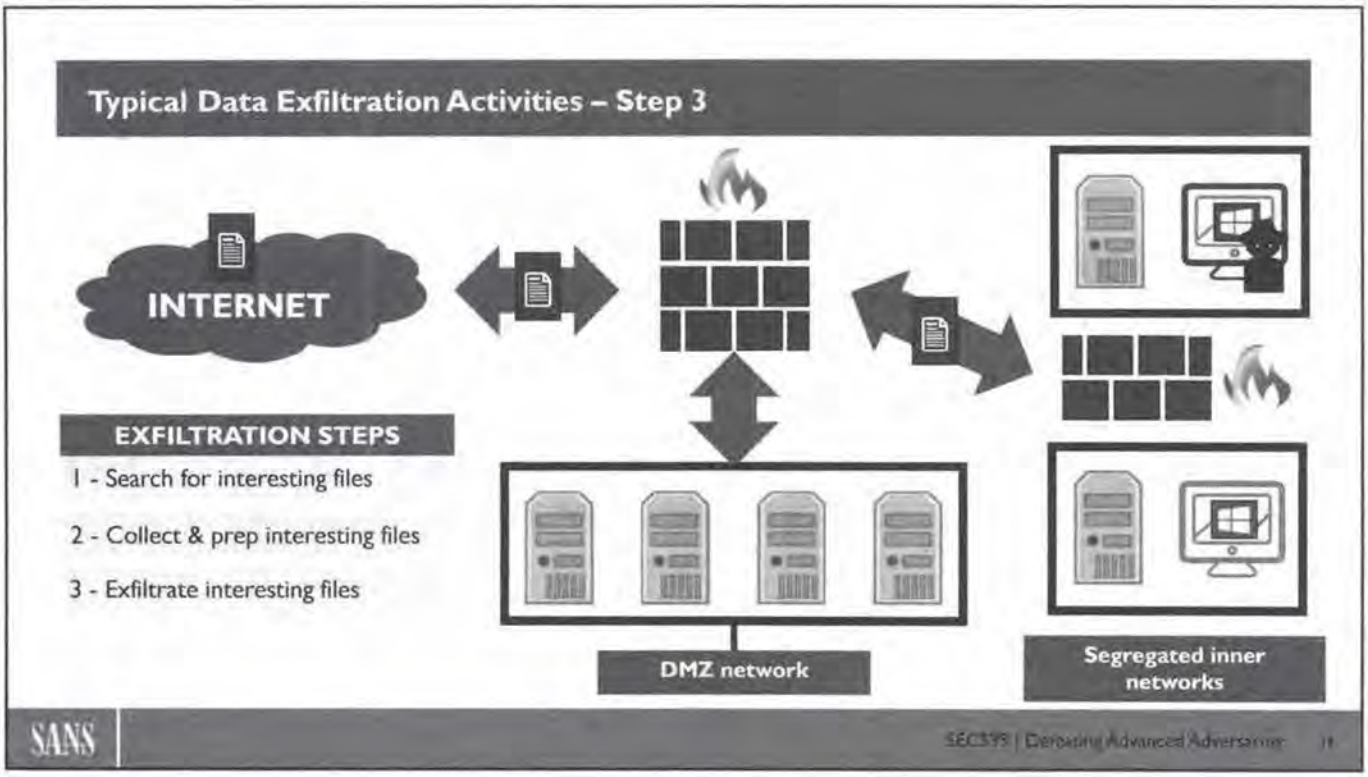


Typical Data Exfiltration Activities – Step 2

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

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.

Step 1 – Searching Data of Interest

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".

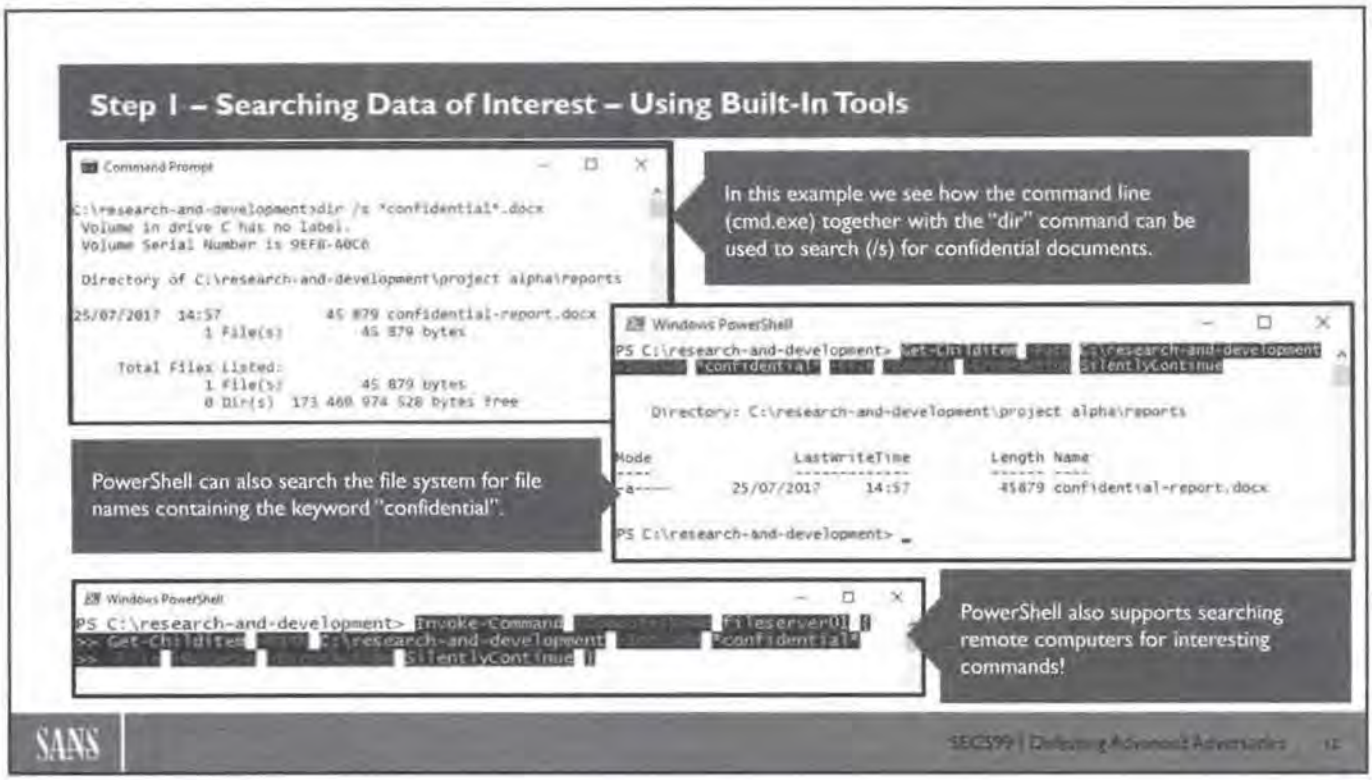
"Actions on objectives" is a broad and generic term, that encompasses many activities performed by attackers. One of these activities is data exfiltration.

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

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.

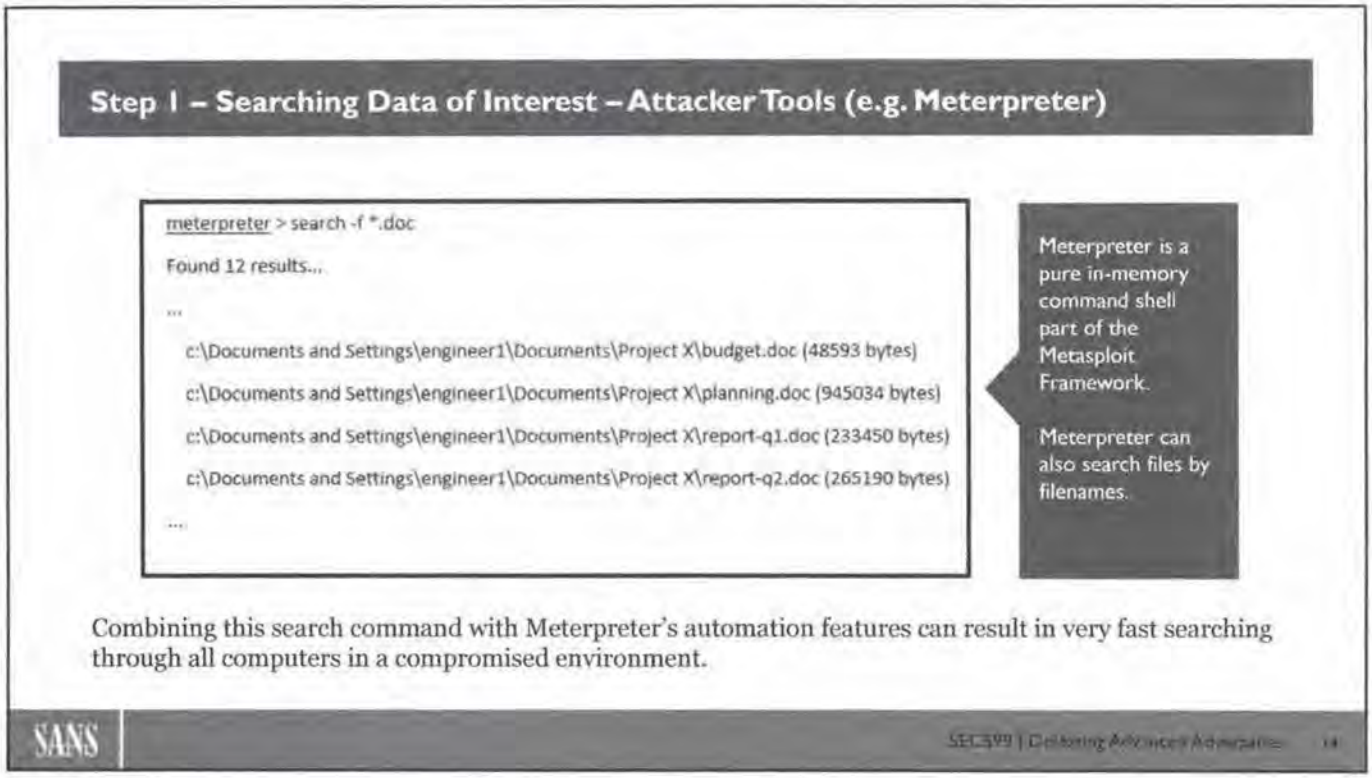
We can filter for specific names with the –Include option: -Include *confidential* will select all files (and directories) with the string confidential in the filename (* is a wildcard, just like with cmd.exe).

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 sub directories, option –Recurse must be provided. Finally, with option –ErrorAction we can make that the Get-Childitem command continues searching even when 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.



Step 1 – Searching Data of Interest – Attacker Tools (e.g. Meterpreter)
Since data exfiltration is such an important goal of attackers, dedicated attacker tools have search capabilities too.

We will illustrate this here with Metasploit's command-line shell Meterpreter.

Meterpreter is a dedicated command-line interpreter with specialized penetration testing commands. The Meterpreter code is not written to disk, but it is injected in an existing process and thus leaves no footprints on the disk.

In a compromised environment, attack tools like Metasploit/Meterpreter can be used to automatically access remote computers and execute commands with its results centralized on the computer compromised by the attacker.

While Meterpreter has many commands to execute attacks, it also has basic commands, for example, to search for files.

In the above example, we see the search command issued to look for Word documents (filename *.doc).

The –f options perform a recursive search, e.g. a search through all sub-directories.

Combining this search command with Meterpreter's automation features can result in very fast searching through all computers in a compromised environment.

Step 1 – Searching Data of Interest – Getting Creative...

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

Some anti-virus scanners (such as ClamAV) allows for searching with customer defined signatures.

Other tools include the YARA engine, which can also be tuned with custom rules.

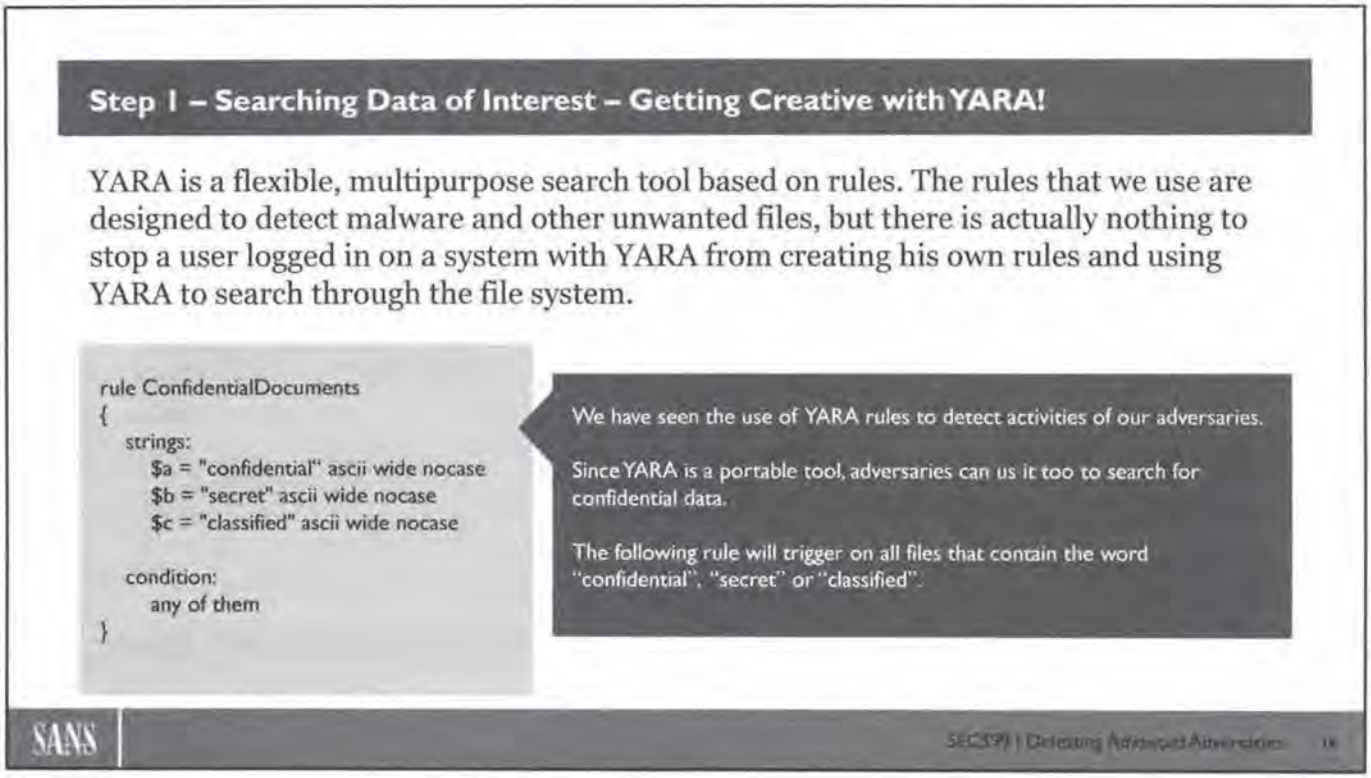
If these tools are not present, the attacker can deploy them without installation.

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 anti-virus 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.



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.

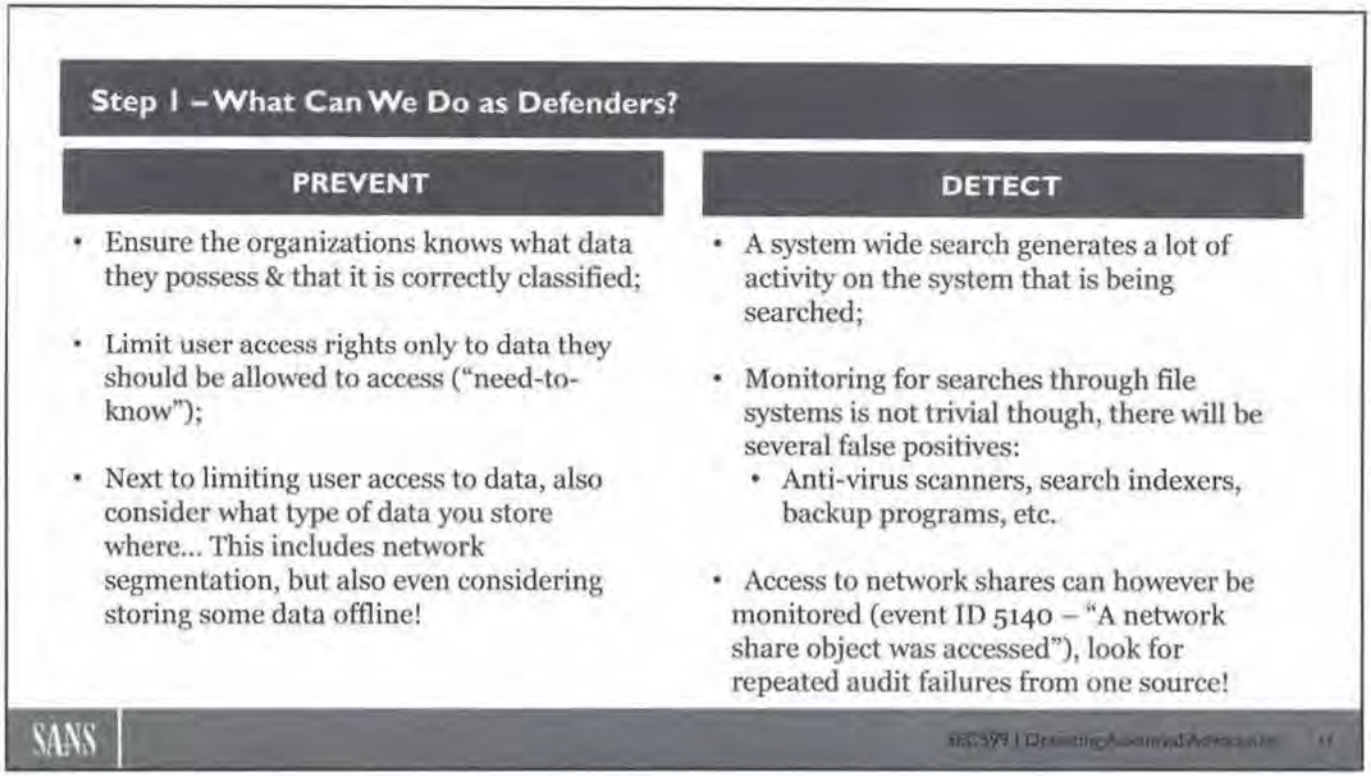
As we saw, YARA is a flexible, multipurpose search tool based on rules. The rules that we use are designed to detect malware and other unwanted files, but there is actually nothing to stop a user logged in on a system with YARA from creating his own rules and using YARA to search through the file system.

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 file system, will locate all documents that contain one of these keywords.



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 & 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 file system 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 file system, this can require opening ten thousand or more directories, and thus produce a considerable number of activities. Searching with an index is different: the file system 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

- anti-virus programs
- Backup programs
- Search indexers

These programs to access the complete file system, and exceptions have to be created when monitoring file system activities, to avoid false positives.

It's important to note though, that adversaries will search remote file systems as well (e.g. go through all Windows shares), which could generate a huge volume of Windows event IDs (event ID 5140)…

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?

- 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.

SANS | SEC595 | Designing Advanced Adversaries 18

A Word on Data Classification

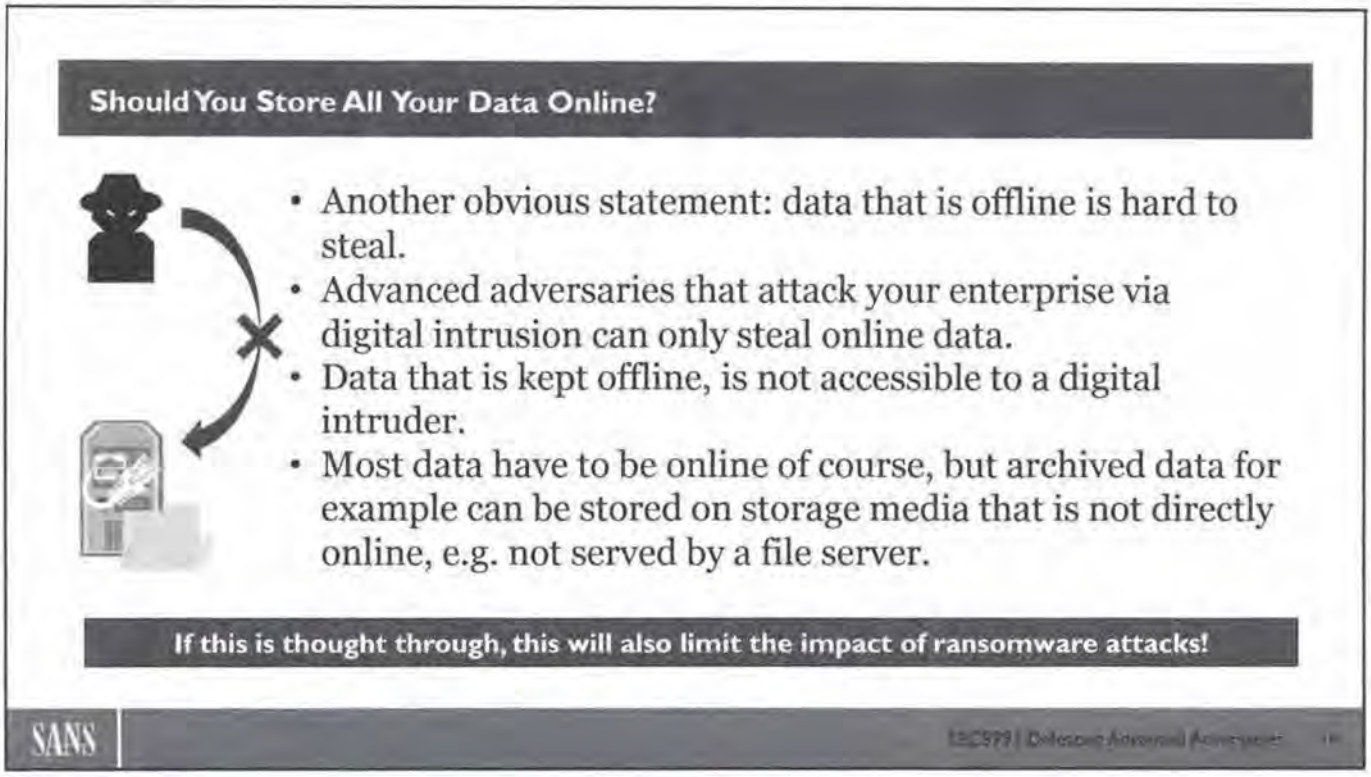
From the example we gave on searching data, it is clear that advanced adversaries will quickly identify important data and try to steal it. Because important data is crucial to your business, it must be adequately protected. Before it can be adequately protected, it must be identified. But do your employees 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 according data classification level.

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



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.

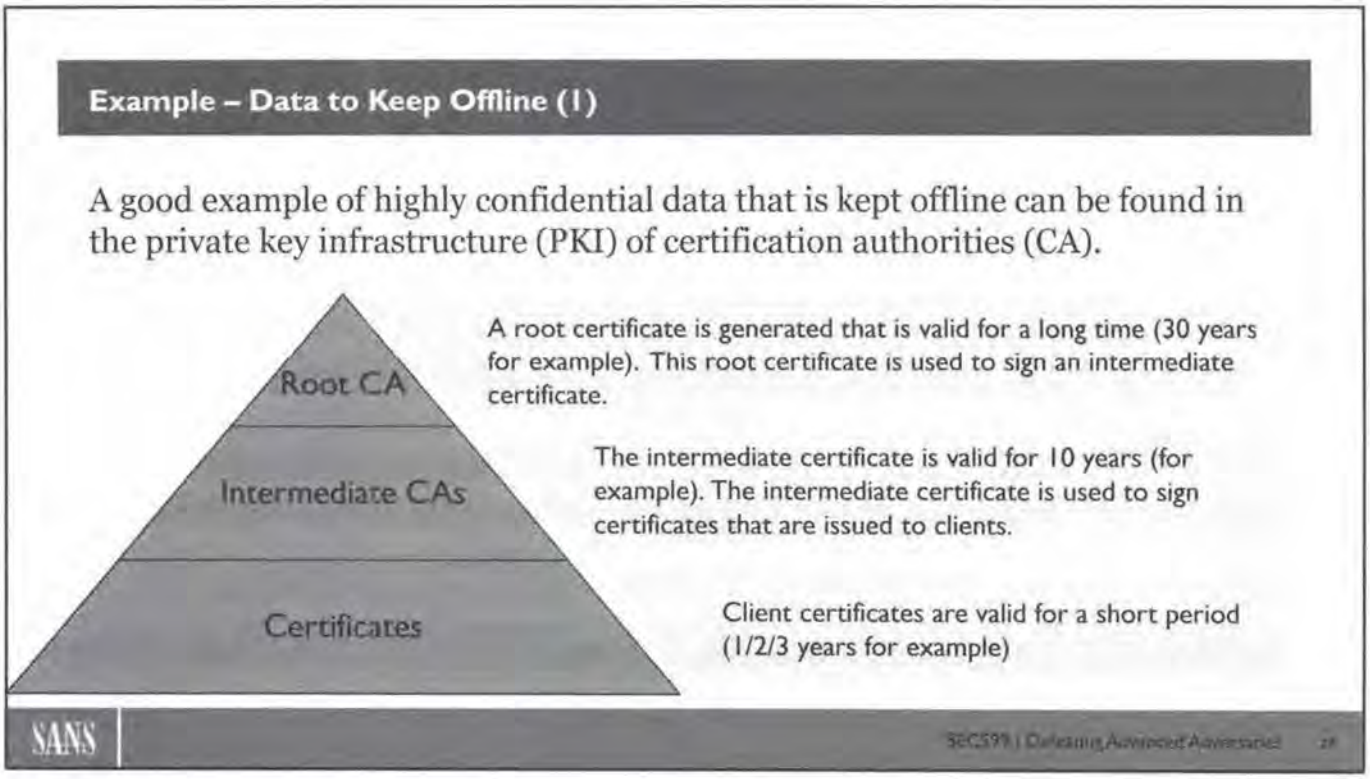
As most advanced adversaries attack your enterprise via digital intrusion (remotely), they have no physical access to the IT assets of your corporation and thus can only access online data.

Data that is kept offline, is not accessible to a digital intruder. Most data have to 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!



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).

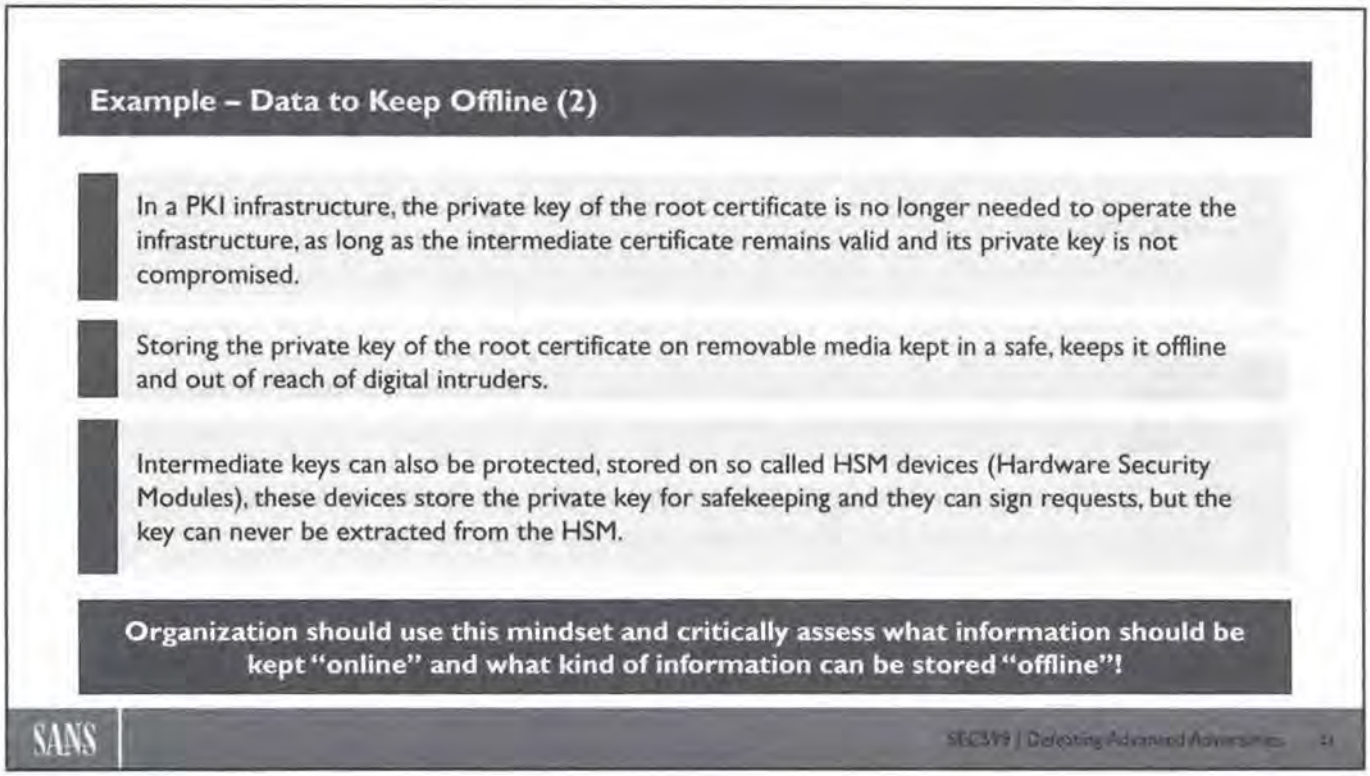
Certification authorities are organizations that issue certificates: signed public keys with metadata. To sign this data, they require private keys. But if those private keys are compromised, CAs can be impersonated and the attacker can issue certificates that cannot be distinguished from legitimate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



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

DigiNotar was a Dutch certification authority, owned by VASCO Data Security International. They suffered from a breach in which an adversary gained access to DigiNotar's PKI infrastructure.

July 2011	These certificates were used in Iran to conduct man-in-the-middle attacks against users of Google's services.
August 2011	Certificate problems emerged in Iran, and Google detected the fraudulent certificates (up to 531 fraudulent certificates were found).
September 2011	DigiNotar filed for bankruptcy because major browsers no longer trusted DigiNotar's certificates.

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.

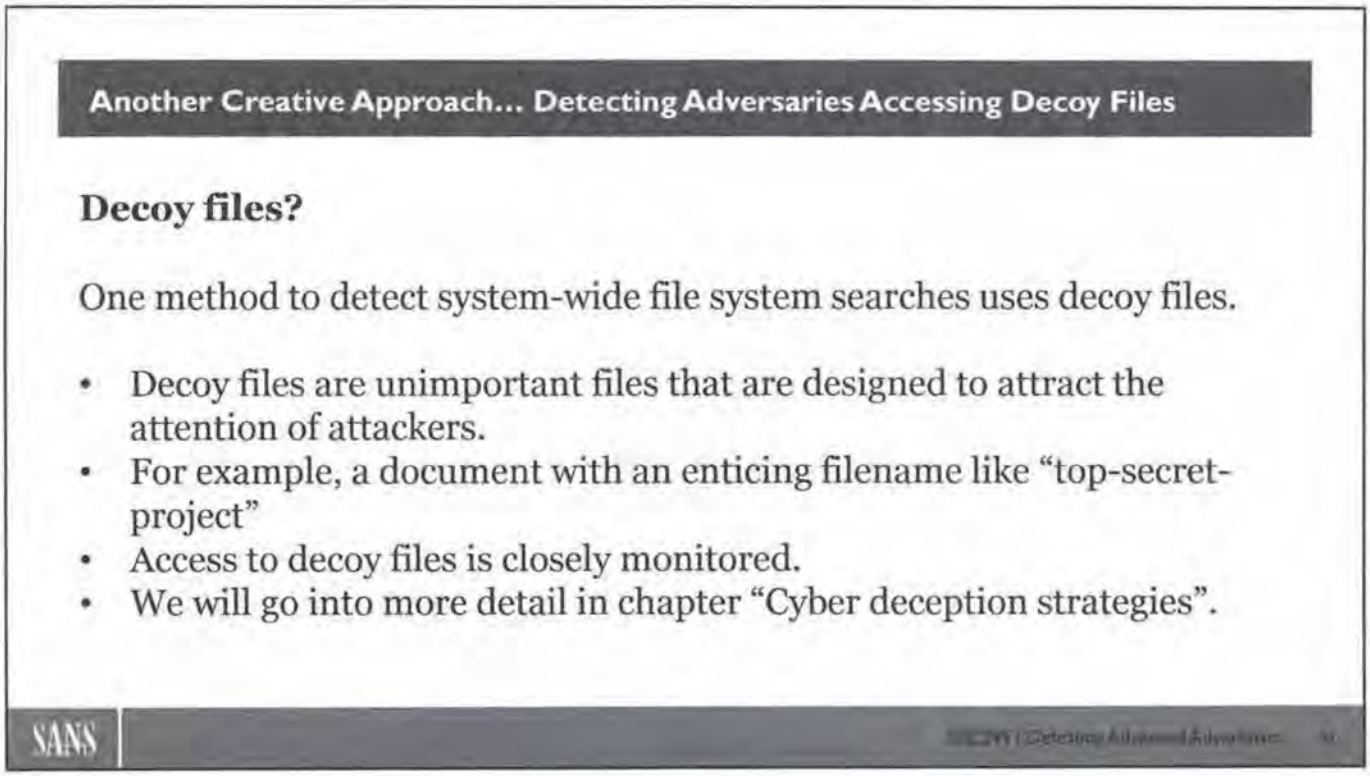
DigiNotar suffered a digital breach: a hacker gained access to DigiNotar's PKI infrastructure.

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 DigiNotars 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.

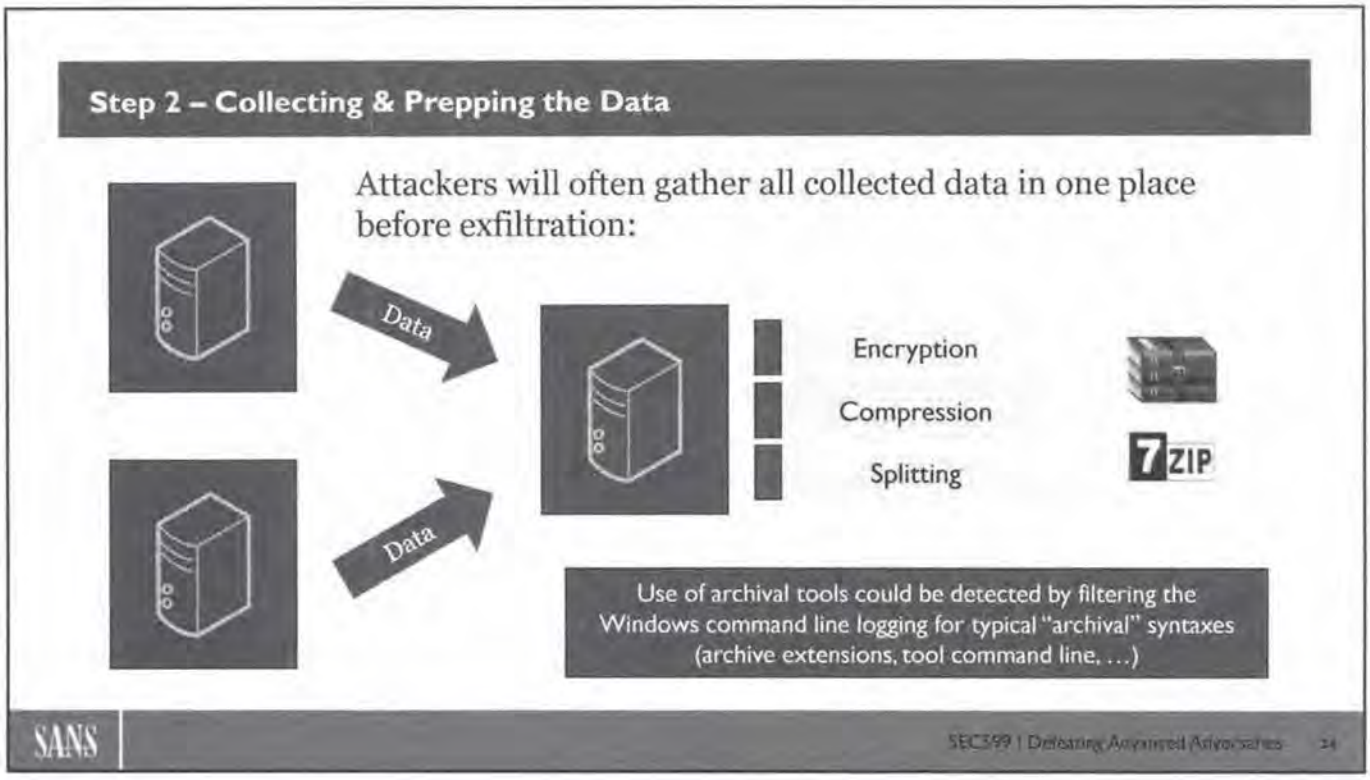


Another Creative Approach... Detecting Adversaries Accessing Decoy Files
A possible solution to the problem of monitoring complete file systems is to limit the number of files we monitor, and devise methods to lead attackers to these files.

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.

We will discuss this in more detail in the next chapter "Cyber deception strategies", but know already that this is a simple method to reduce the overload of activities to monitor.



Step 2 – Collecting & 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, ...).

Step 3 – Exfiltrating the Data – Using the Network

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

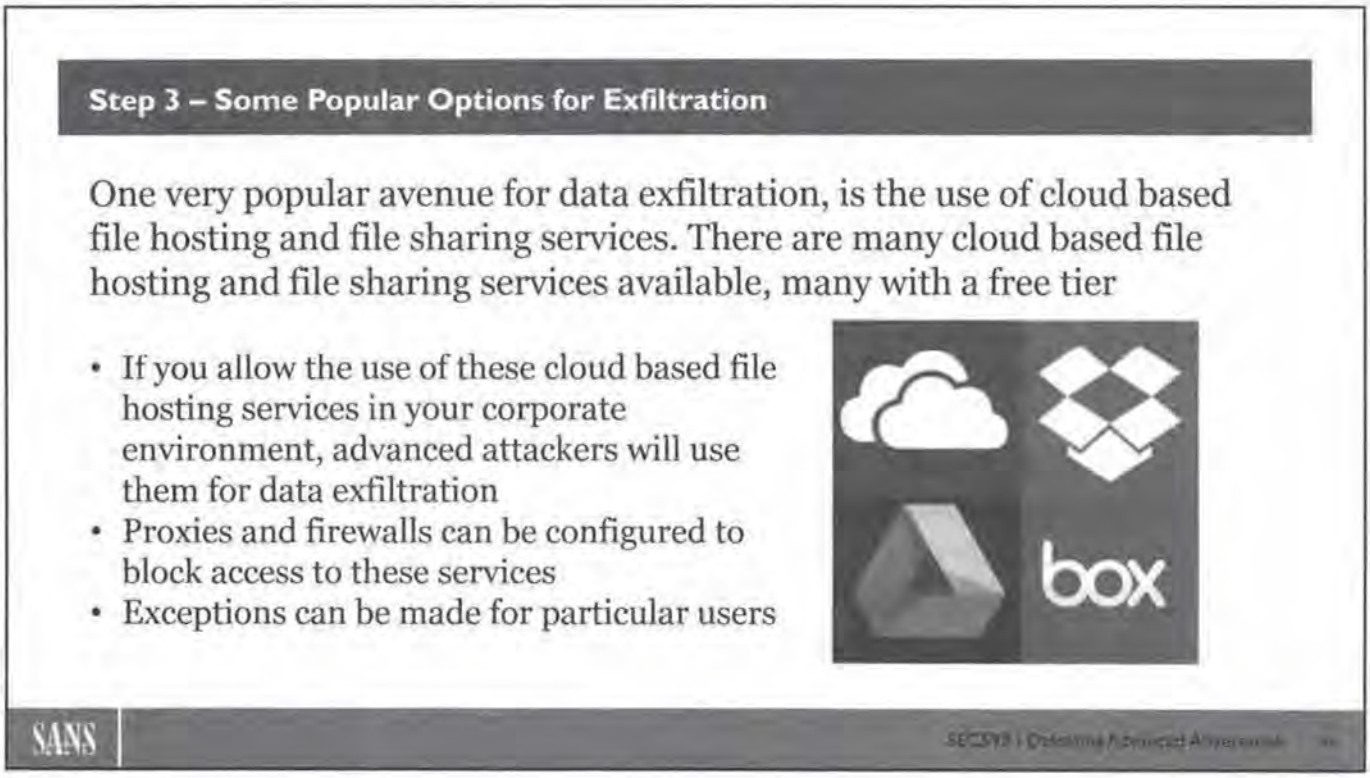
- Attackers will gather all collected data on one machine with network access.
- To speed up network transfer, files will be grouper together in one or more archives and compressed.
- To prevent keyword based exfiltration detection, the data can be encrypted.
- Classic data transfer methods like HTTP/HTTPS uploads or email attachments will be used.

SANS

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.



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

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 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).

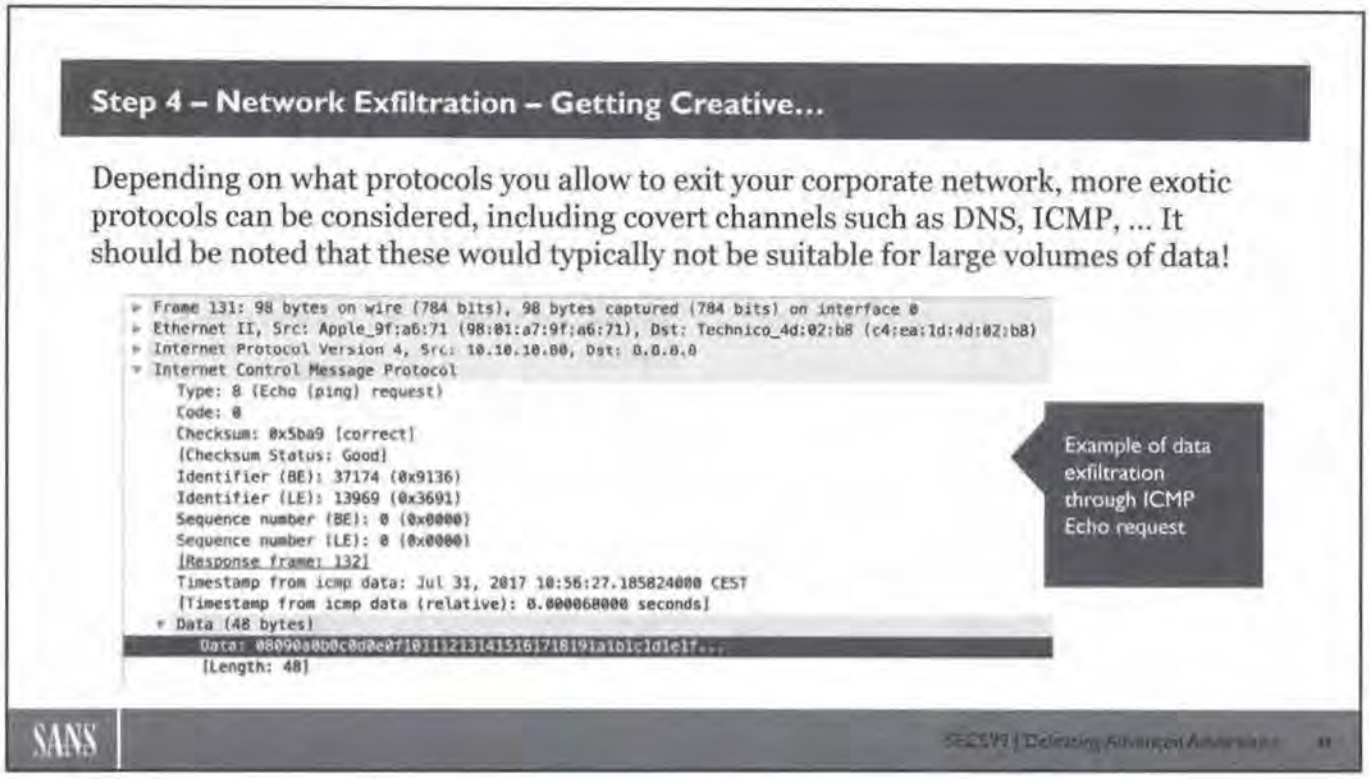
And then there are less obvious services that can also be used to exfiltrate data.

Take for example VirusTotal, a service that accepts files up to 128MB for anti-virus 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.



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

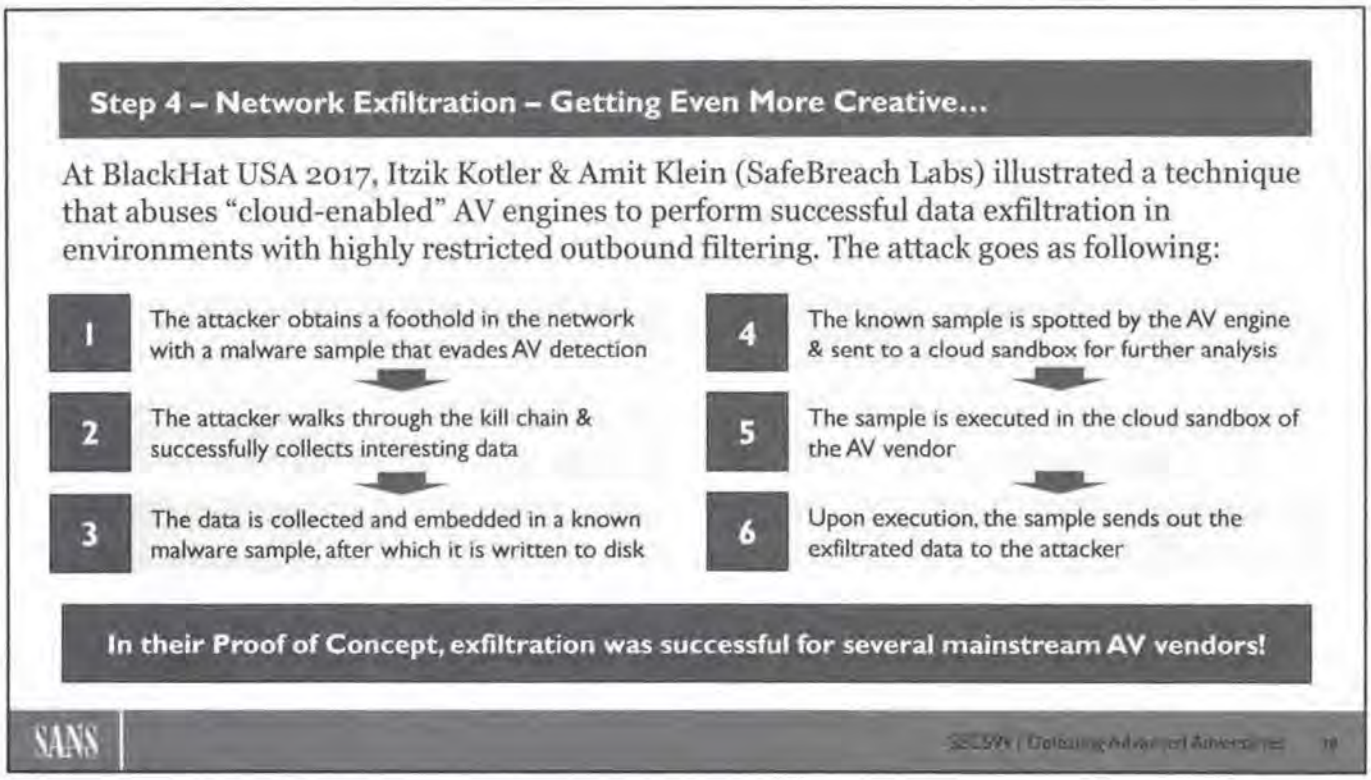
Some of these protocols are also implemented over HTTP, like FTP: it is possible to access and upload files to an FTP server using a browser. And if you allow raw TCP connections (for example over port 80), then large amounts of binary data can be exfiltrated. TCP and protocols based on TCP allow for 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;
2. The attacker walks through the kill chain & 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 & 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

Step 4 – Preventing Network Exfiltration

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!

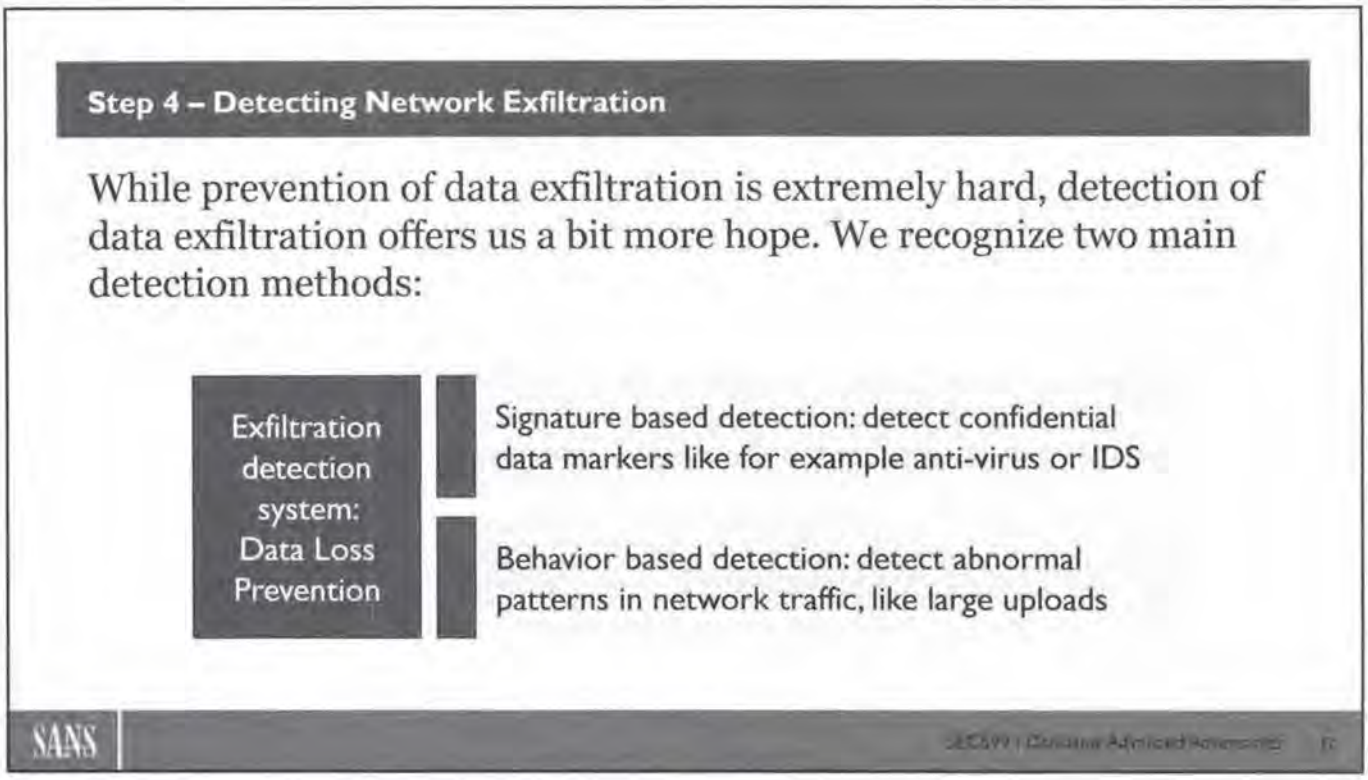
Step 4 – Preventing Network Exfiltration
We presented many ways to exfiltrate data and to abuse services to facilitate data exfiltration.

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 to 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.



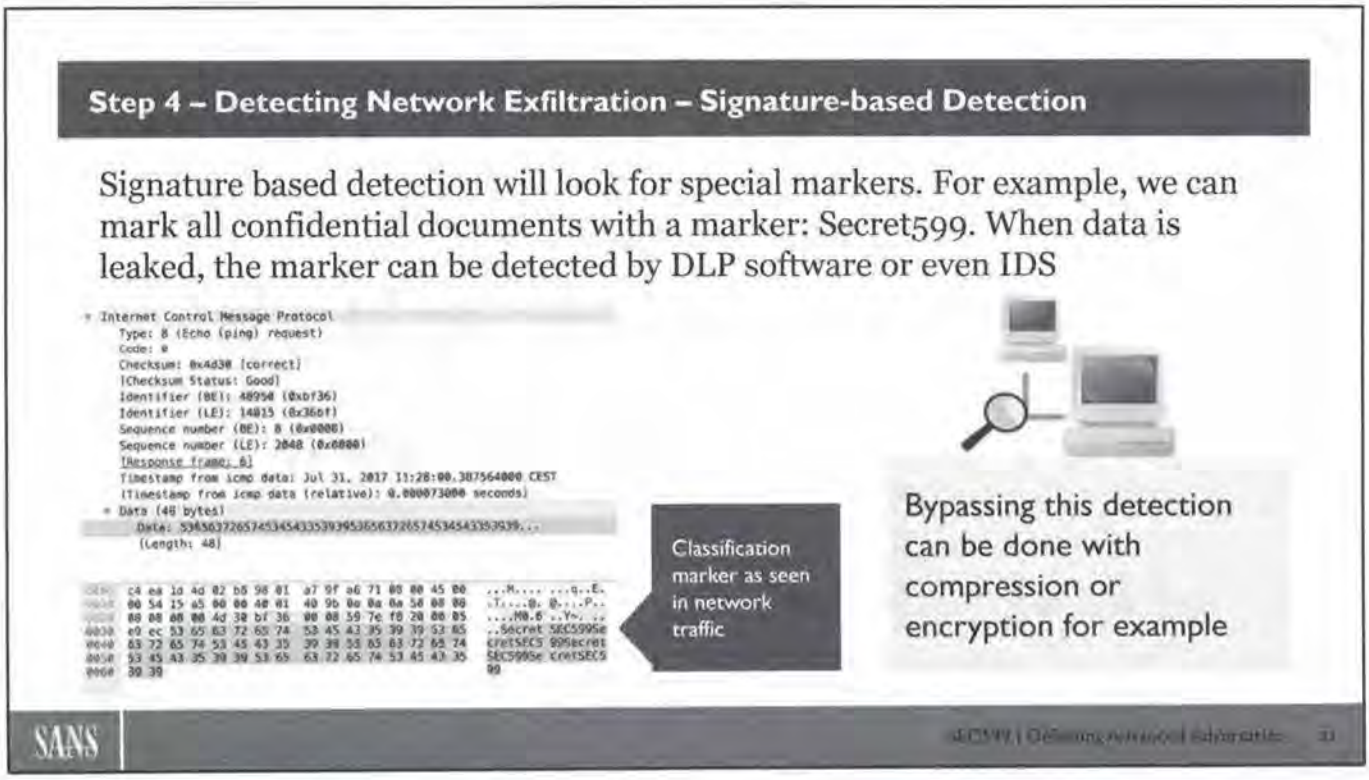
Step 4 – Detecting Network Exfiltration
While prevention of data exfiltration is extremely hard, detection of data exfiltration offers us a bit more hope. There are essentially 2 methods that data exfiltration detection systems used to detect possible data leakage. A data exfiltration detection system 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 observer 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 anti-virus 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.



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, for example by adding this watermark as metadata to the document.

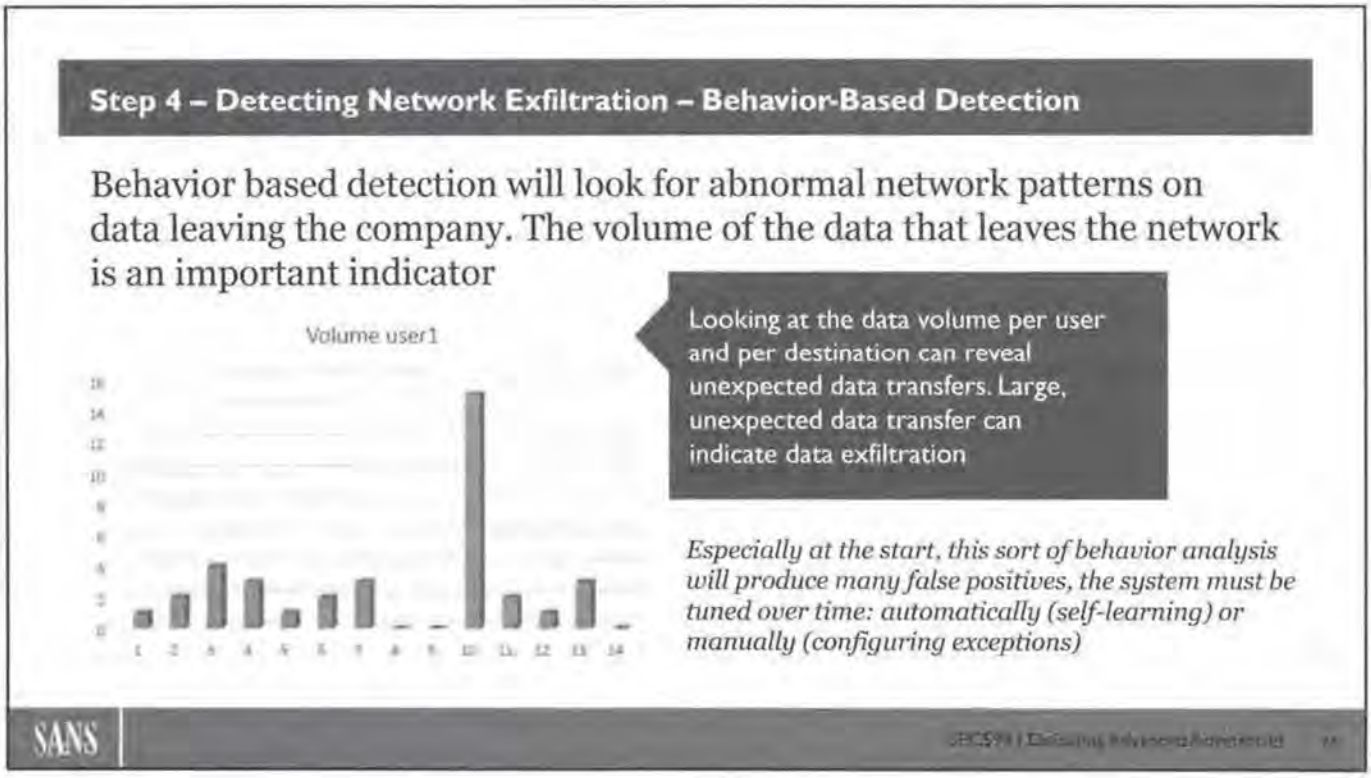
When this document is exfiltrated, the DLP software will detect network traffic leaving the corporate network that has a byte pattern corresponding to Secret599. This will raise an alert (or can even block the network transfer, depending on the DLP solution and the corporate network architecture).

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.



Step 4 – Detecting Network Exfiltration – Behavior-Based Detection
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.

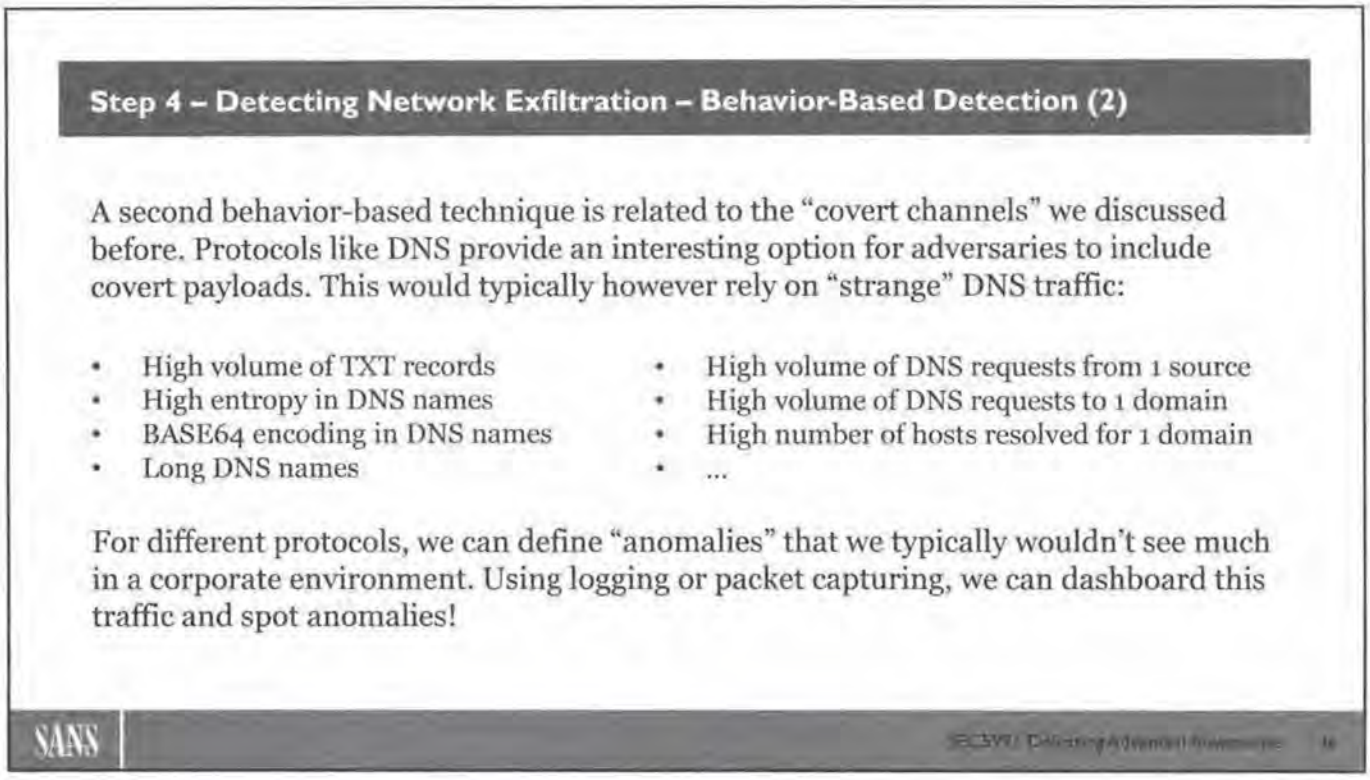
There are several solutions to reduce the rate of false positives, for example like Splunk will analyze actual traffic.

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. Should adversaries want to use this type of strategy however, this would typically rely on "strange" DNS traffic:

- The ratio of A-records vs TXT-records should show that TXT records occur far less in the environment. A high volume of TXT records could indicate a DNS tunnel;
- High entropy in DNS names could reveal randomly generated domain names;
- BASE64 is a preferred encoding mechanism used by many adversaries. Data could be exfiltrated by splitting it over different BASE64 strings;
- Generally speaking, DNS tunneling would most likely lead to long DNS names being resolved;

We could also focus on the volumes of traffic:
- A high volume of DNS requests from 1 source should be a source for investigation;
- A high volume of DNS requests to 1 domain;
- A high number of hosts resolved per 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!

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

A 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.

Physical Exfiltration – Printouts

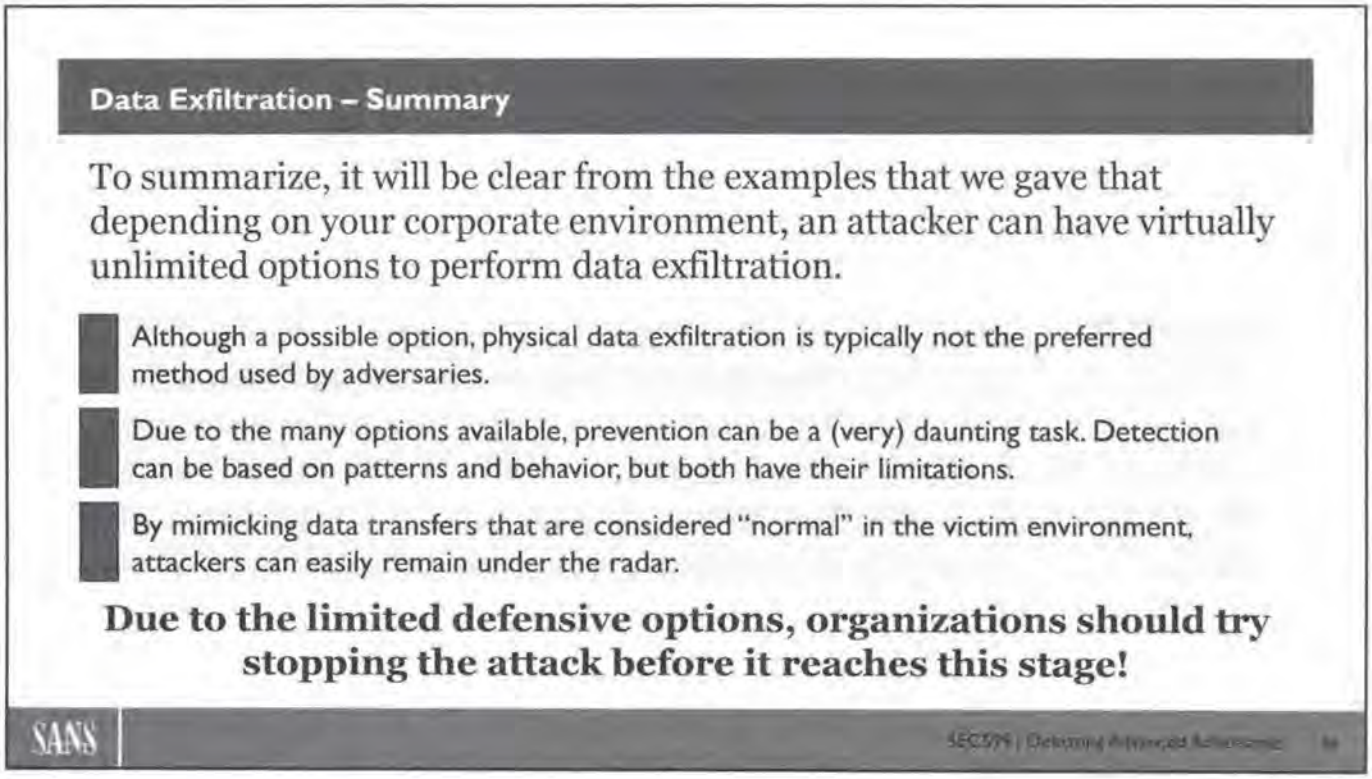
When it comes to physical exfiltration, we want to mention that several interesting methods have been devised to detect or thwart this. These are typically the kind of efforts one would see in military or intelligence operations. The example we want to mention here takes printers as an exfiltration device. 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



Data Exfiltration – Summary

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

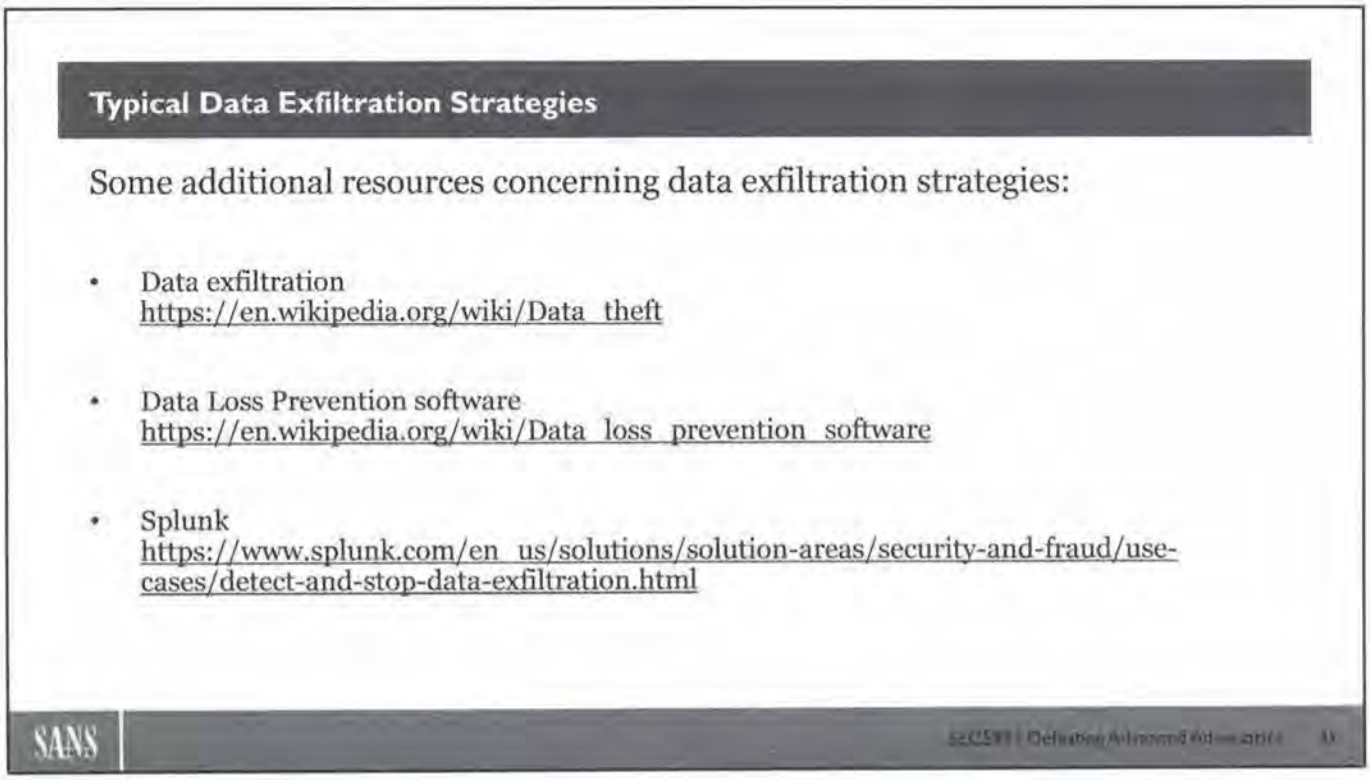
Although a possible option, physical data exfiltration is typically not the preferred method used by adversaries. Network exfiltration is by far the most commonly used exfiltration method. Typically, the adversary has already set up a C&C channel, which it could (partially) reuse for exfiltration.

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…



Typical Data Exfiltration Strategies

Some additional resources concerning data exfiltration strategies:

- Data exfiltration
  https://en.wikipedia.org/wiki/Data_theft
- Data Loss Prevention software
  https://en.wikipedia.org/wiki/Data_loss_prevention_software
- Splunk
  https://www.splunk.com/en_us/solutions/solution-areas/security-and-fraud/use-cases/detect-and-stop-data-exfiltration.html

Course Roadmap

- Day 1: Knowing the adversary, knowing yourself
- Day 2: Averting Payload Delivery
- Day 3: Preventing Exploitation
- Day 4: Avoiding Installation, foiling Command & Control & thwarting lateral movement
- **Day 5: Exfiltration, Cyber Deception & Incident Response**
- Day 6: APT Defender Capstone

SEC599.5

Data exfiltration
- Typical data exfiltration strategies
- Exercise: Detecting data exfiltration using Suricata

Cyber deception strategies
- Tricking the adversary
- Exercise: Making your honeypot irresistibly sweet

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

Patrolling your network
- Proactive threat hunting strategies
- Exercise: Hunting your environment using OSQuery / ELK

Incident response
- Incident response process
- Exercise: Responding to an incident using GRR

This page intentionally left blank.

Exercise – Data Exfiltration Using Suricata

The objective of the lab is to detect data exfiltration taking place in our environment. As data exfiltration is a tricky subject, we will illustrate different methods to try detecting exfiltration.

High-level exercise steps:

1. Detect credit card information that is sent out in clear-text using Suricata;
2. Detect confidential data that is mailed to recipients outside of the organization using Suricata;
3. Using ntop-ng to detect excessive data volumes being uploaded.

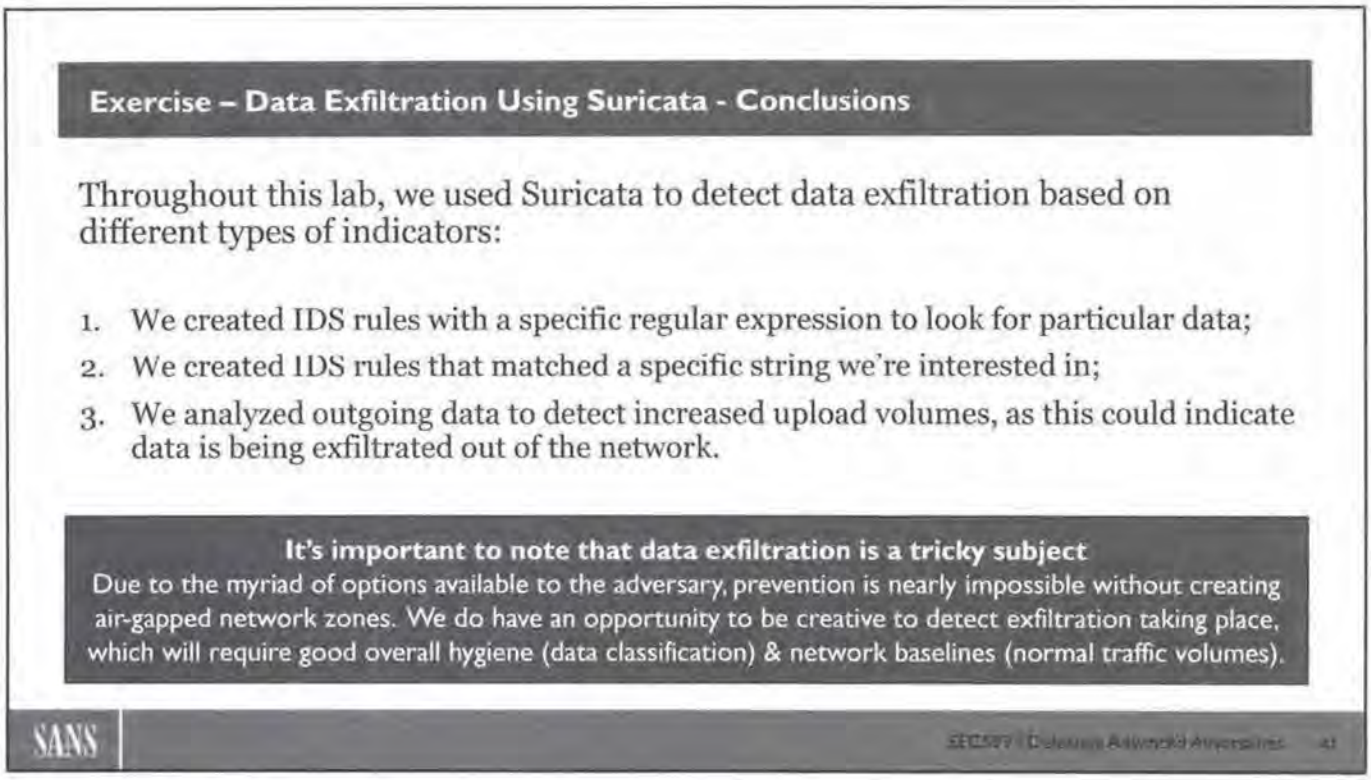
Exercise – Data Exfiltration Using Suricata
The objective of the lab is to detect data exfiltration taking place in our environment. As data exfiltration is a tricky subject, we will illustrate different methods to try detecting exfiltration.

As part of the lab, the following data exfiltration methods will be discussed:

- Credit card information that is sent out in clear-text;
- Confidential data that is mailed to recipients outside of the organization;
- Volume-based analysis for exfiltrated data.

We will rely on Suricata's IDS rule generation for the first two scenario's, while we will use PfSense's "ntop-ng" package to detect excessive data volumes.

For additional guidance & details on the lab, please refer to the LODS workbook.



Exercise – Data Exfiltration Using Suricata – Conclusions

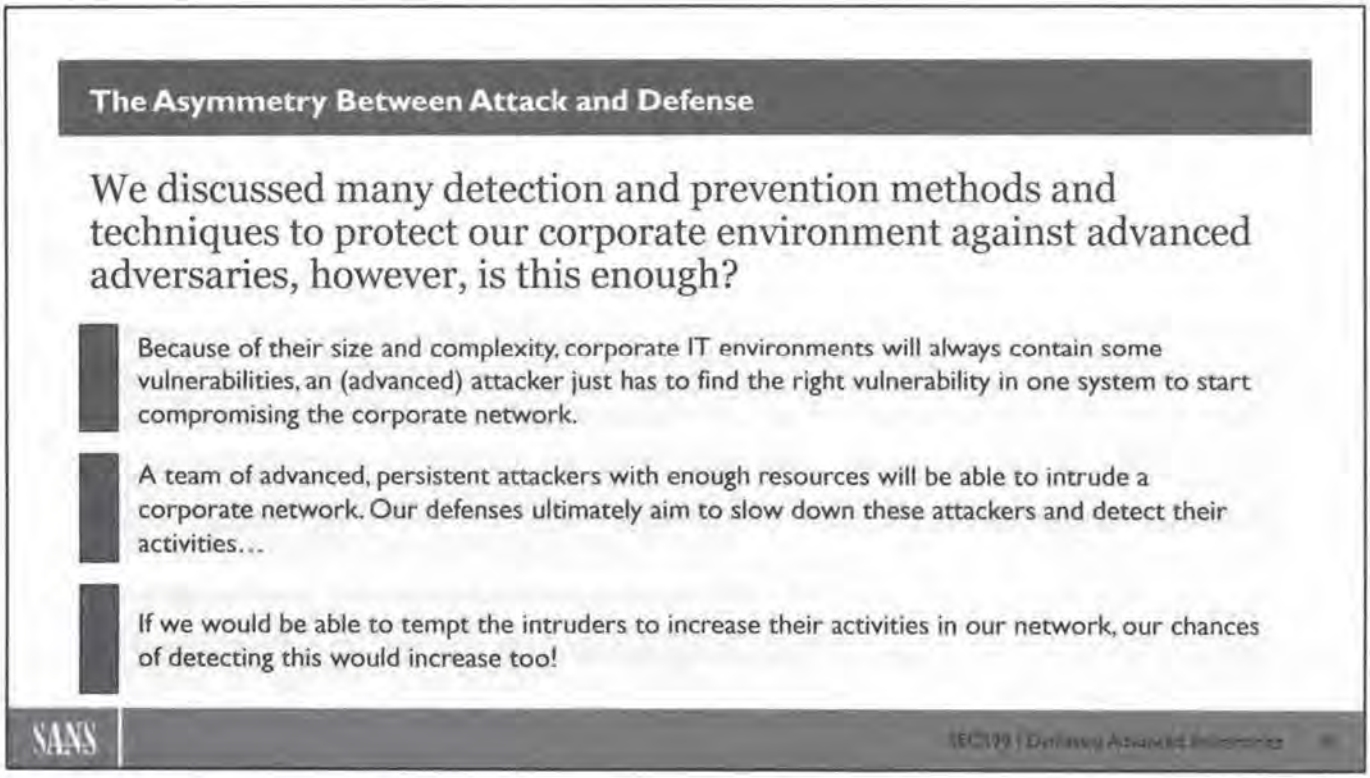
Throughout this lab, we used Suricata to detect data exfiltration based on different types of indicators:

1. We created IDS rules with a specific regular expression to look for particular data;
2. We created IDS rules that matched a specific string we're interested in;
3. We analyzed outgoing data to detect increased upload volumes, as this could indicate data is being exfiltrated out of the network.

It's important to note that data exfiltration is a tricky subject!

Due to the myriad of options available to the adversary, prevention is nearly impossible without creating air-gapped network zones. We do have an opportunity to be creative to detect exfiltration taking place, which will require good overall hygiene (data classification) & network baselines (normal traffic volumes).

This page intentionally left blank.



The Asymmetry Between Attack and Defense

In the preceding days and chapters, we discussed many detection and prevention methods and techniques to protect our corporate environment against advanced adversaries.

Unfortunately for us, because of their size and complexity, corporate IT environments will always contain some vulnerabilities. Complex systems always contain errors, IT environments are certainly not different, quite to the contrary. Errors are not always vulnerabilities that can be exploited, but we certainly have to manage vulnerabilities in our systems.

An (advanced) attacker just has to find the right vulnerability in one system to start compromising the corporate network. Finding one crack in a system or environment is enough to give them a start, that can ultimately lead to full compromise by exploiting a chain of vulnerabilities and/or misconfigurations.

While we, as defenders, have to protect all our systems and mitigate all vulnerabilities. If we miss one type of vulnerability or have one system that is not properly protected, it can be all it takes to get an adversary to compromise our environment.

This is called the asymmetry between attack and defense, we can oversimplify this to:

- Attacker: find 1 weakness
- Defender: mitigate all weaknesses

The asymmetry between attacker and defender is what makes defending the corporate IT infrastructure a resource-intensive job.

Because defenders have to defend more systems than attackers need to attack, defenders need to be in greater numbers or have more resources than the attackers to have a chance at properly defending the corporate network.

A team of advanced, persistent attackers with enough resources will always be able to intrude a corporate network, no matter how much resources we have to defend the corporate network.

As Bruce Schneier puts it:

"Attacks always get better, they never get worse".

This means that over time, attacks improve, and defenses have to keep up, otherwise attackers will succeed. And because of the asymmetry, keeping up for defenders costs more than improving for attackers.

Our defenses ultimately aim to slow down these attackers so that we have a better chance at detecting their activities.

If we would be able to tempt the intruders to increase their activities in our network, our chances of detecting this would increase too: that is the premise behind "tricking the adversary".

Tricking the Adversary (1)

KALI LINUX

- Kali: "The quieter you become, the more you are able to hear"
- SEC599: "The noisier you become, the more they are able to hear you"

- We want the adversary to make more "noise" (e.g. generate more alerts). We will achieve this by tricking the adversary
- We trick the adversary by employing decoy systems and data. Decoys have no real value to our corporation
- We closely monitor decoy systems and data for any unusual activity

SANS | SEC599 | Defeating Advanced Adversaries

Tricking the Adversary (1)
Kali Linux is a well-known Linux distribution for penetration testers.

Its moto is:

"The quieter you become, the more you are able to hear"

As a moto to "tricking the adversary", we will use:

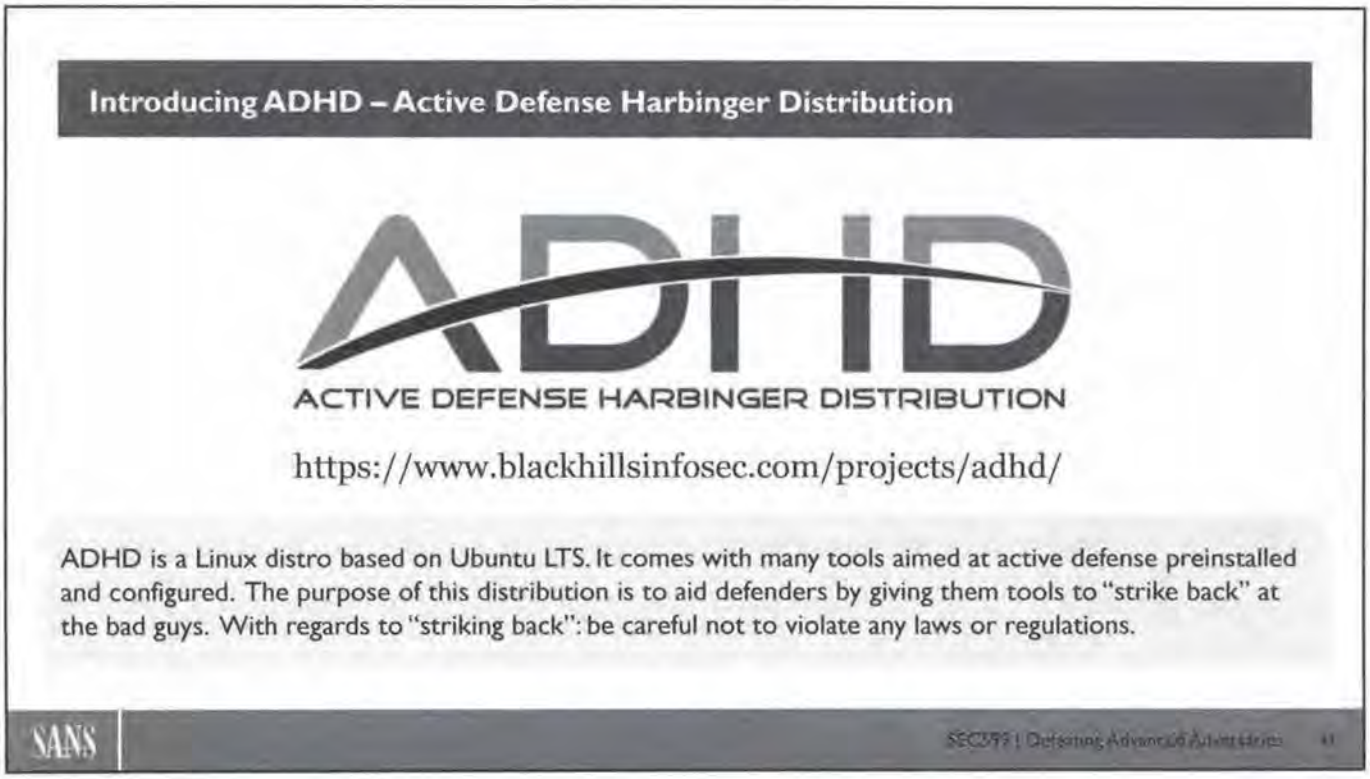
"The noisier you become, the more they are able to hear you"

By tricking the adversary, we want him to make mistakes that will produce noise (events) that we can detect. The idea behind "tricking the adversary" is that we want the adversary to make more "noise".

The adversary wants to be as quiet as possible in our networks and systems (e.g. not generate events and certainly not alerts), so that we will not detect them and that they have more time to perform their malicious deeds. We want the adversary to make more "noise", e.g. generate more alerts, and we will achieve this by tricking the adversary.

The way will achieve this "tricking of the adversary", is by deploying and using decoy systems and data. Decoys are fake systems and fake data. They have no real value to our corporation. We try to make these decoys enticing to the attackers so that they will find and try to compromise these systems and access this data. This will not only slow down the attackers by having them waste their time on systems and data that has no real value, but we will also closely monitor decoy systems and data for any unusual activity.

This enhanced monitoring activity for decoys is our trap: the snare with which we hope to catch attackers!



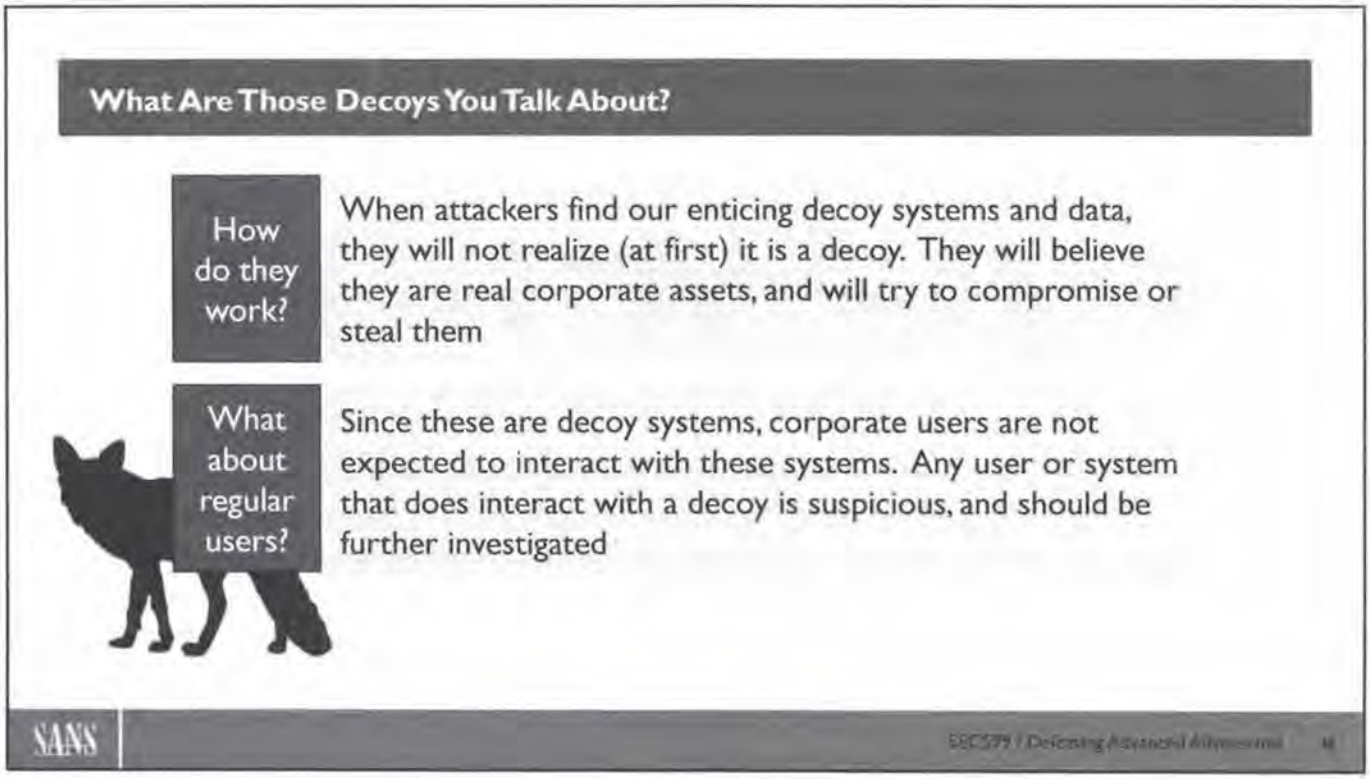
Introducing ADHD – Active Defense Harbinger Distribution

Cyber deception strategies are often used as part of an overall "Active Defense" strategy, where defenders are trying to defeat adversaries by actively engaging them using tricks & dedicated systems set up.

An interesting toolkit for active defense is ADHD (Active Defense Harbinger Distribution). ADHD is a Linux distro based on Ubuntu LTS. It comes with many tools aimed at active defense preinstalled and configured. The purpose of this distribution is to aid defenders by giving them tools to "strike back" at the bad guys. With regards to "striking back": be careful not to violate any laws or regulations.

We will not go through the entire ADHD toolsuite during this course, but we will, for example, use HoneyBadger during an upcoming exercise.

You can find more information on ADHD here on https://www.blackhillsinfosec.com/projects/adhd/.



What Are Those Decoys You Talk About?

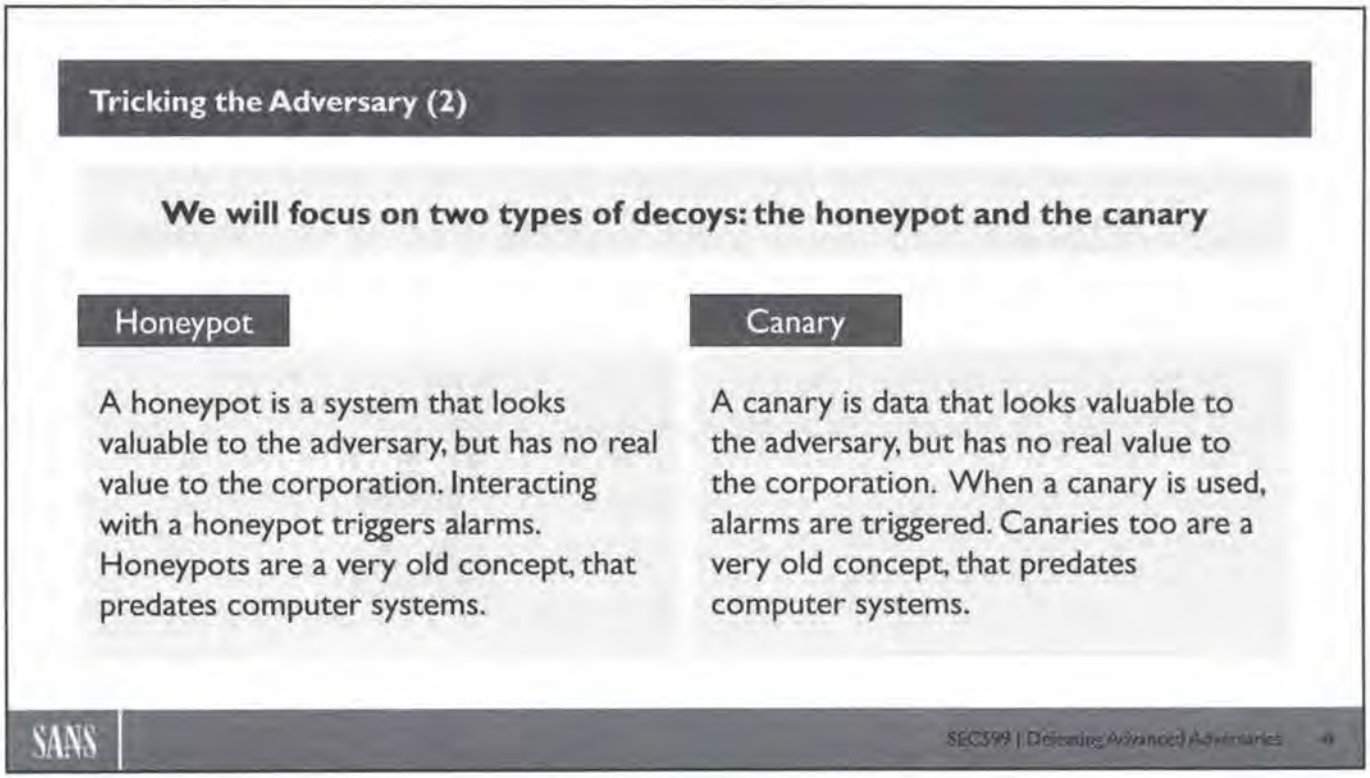
It should not be obvious that a decoy system is a decoy, and not a real system, although not too much time and effort should be made to make a decoy system look like a real system.

When attackers find our enticing decoy systems and data, they will not realize (at first) it is a decoy. That's why it is important that it is not obviously a decoy system, otherwise, the attackers will not be slowed down or tricked into generating alerts.

They will believe that the decoy systems and data are real corporate assets, and if we make them look like valuable corporate assets, attackers will try to compromise or steal them. Disguising a decoy as a valuable corporate asset requires subtlety, we don't want to make it too obvious that it is very valuable, otherwise the attackers might become suspicious. The examples we give here in this book are often too obvious, but that is to make the point clear.

Since these are decoy systems and data, corporate users are not expected to interact with these systems or access this data.

Any user or system that does interact with a decoy is suspicious and should be further investigated. It will, of course, happen that users will discover these assets and try to access these. We will consider these alerts as false positives, but nevertheless, it should be pointed out that these users should be talked to, just to figure out why they wanted to access these systems or data.



Tricking the Adversary (2)
The two types of decoy we will focus on in this chapter are:

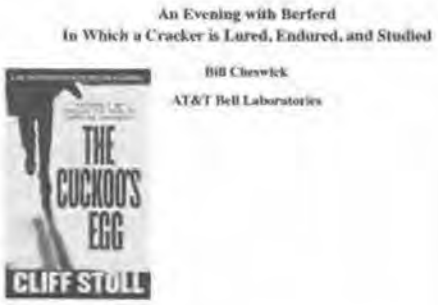
- The honeypot
- The canary

A honeypot is a system that looks valuable to the adversary but has no real value to the corporation. Interacting with a honeypot triggers alarms. Honeypots are a very old concept, that predates computer systems. They have been used by hunters and soldiers to ambush the adversary.

A canary is data that looks valuable to the adversary but has no real value to the corporation. When a canary is used, alarms are triggered. Canaries too, are a very old concept, that predates computer systems. They have been used by hunters and soldiers to find the adversary.

History of the Honeypot

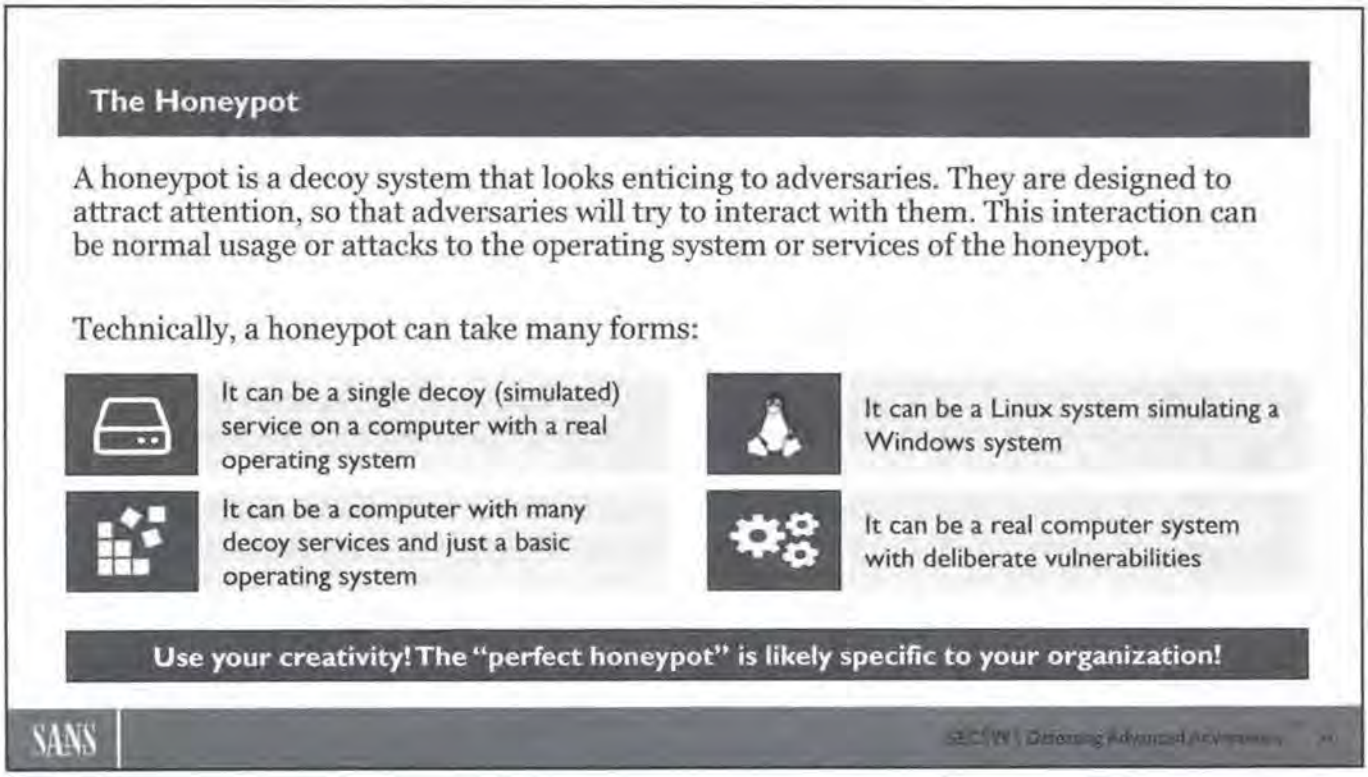
- 1990/1 Books: "The Cuckoo's egg" and "An evening with Berferd"
  - Use deceptive techniques to catch a hacker
- 1998 - the Deception Toolkit
  - Fred Cohen, first publicly available honeypot
- 1998 - CyberCop Sting
  - First commercial honeypot



History of the Honeypot

The idea of honeypots began in 1991 with two publications, "The Cuckoos Egg" and "An Evening with Berferd". "The Cuckoos Egg" by Clifford Stoll was about his experience catching a computer hacker that was in his university searching for secrets. The other publication, "An Evening with Berferd" by Bill Cheswick is about a computer hacker's moves through traps that he and his colleagues used to catch him. Both of these writings were the beginnings of what became honeypots.

The first type of honeypot was released in 1997 called the Deceptive Toolkit. The point of this kit was to use deception to attack back. In 1998 the first commercial honeypot came out. This was called Cybercop Sting.



The Honeypot

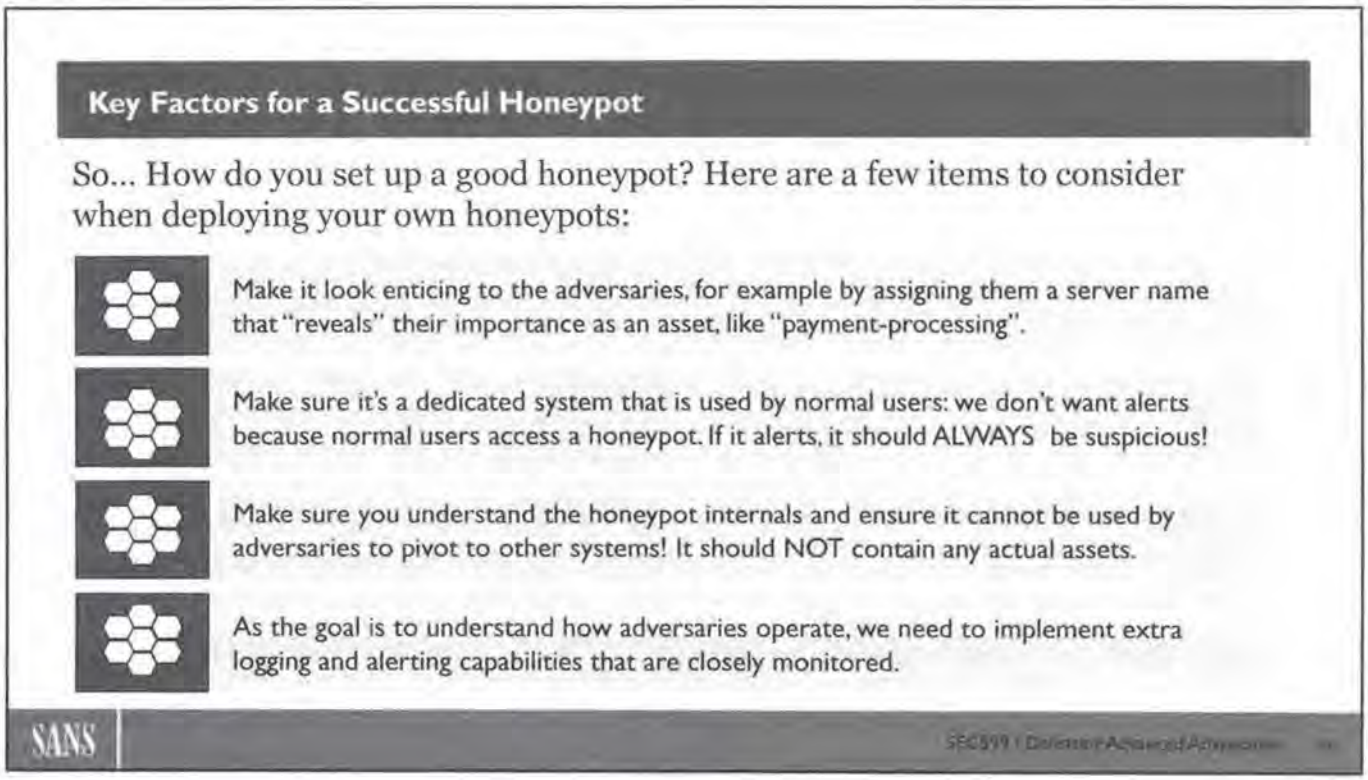
A honeypot is a decoy system that looks enticing to adversaries. They are designed to attract attention so that adversaries will try to interact with them. This interaction can be normal usage or attacks to the operating system or services of the honeypot.

The name honeypot refers to children tales of bears liking honey: bears like honey, and they will go the extraordinary length to obtain honey from bee nests. A honeypot is the ultimate attraction for a bear: it's a container filled with pure honey, ready to be consumed.

Technically, a honeypot can take many forms:

- It can be a single decoy (simulated) service on a computer with a real operating system: we just install a computer with a normal operating system and install a service on it to attract adversaries. This can be a real service like a webserver or a program that simulates the basic features of a webserver.
- It can be a computer with many decoy services and just a basic operating system: we can install an operating system with the minimum amount of services, and then replace those services with decoy services.
- It can be a Linux system simulating a Windows system: for example, we install SAMBA and configure it to resemble a Windows file server as closely as possible.
- It can be a real computer system with deliberate vulnerabilities
- ...

Don't forget to use your creativity! The "perfect honeypot" will most likely be different from organization to organization. If you want to catch targeted attacks, you should do more than just setting up a "generic" honeypot! Think about your crown jewels and make something that actually looks like that!



Key Factors for a Successful Honeypot

We use honeypots to trick the adversary: we want the honeypot to be discovered by an adversary and that it looks enticing so that the adversary will try to interact with it. By interacting with it, we detect the presence of adversaries in our corporate infrastructure.

To achieve these goals, it is important that honeypots:

- Look enticing to the adversaries, for example by assigning them a server name that "reveals" their importance as an asset, like "payment-processing". Honeypots have to be discovered by the adversaries when they search through our networks, and by assigning them enticing names, we try to increase the chances that adversaries will try to access them. But this must not be overdone, otherwise, the attackers will know it is a honeypot. For example, if you call your honeypot server "super-secret-file-server", that's probably overdoing it.
- Are not used (interacted with) by normal users: we don't want alerts because normal users try to access a honeypot. Honeypots should not have anything on them that normal users would need to use. We want to avoid false positive alerts as much as possible. That said, if a normal user does access a honeypot, it should be investigated.
- Have extra monitoring and alerting capabilities that are closely monitored. Honeypots should only generate alerts when adversaries access them, therefore, all alerts should be investigated.
- Should not contain corporate assets: we do not want to attract adversaries with real corporate assets like data and confidential documents. If you want to make a honeypot more enticing with documents, use fake confidential documents.
- Should not be able to be used by adversaries to pivot to other systems: particular care should be taken that a honeypot does not contain tools and services and has access to other systems so that it can be used as a pivot.

Some Example Honeypots – Simple Decoy Administrative Website

Setting up a honeypot doesn't have to be a lot of work... As an example, you could just create a normal server with normal webserver software on it, and then create a fake website that simulates an administrative tool starting with a login page:

This is a honeypot that can be created without requiring specialized honeypot software

Every attempt to login raises an alert and should be investigated

As a bonus, you could include HoneyBadger's code to track all visits to the website!

Some Examples Honeypots – Simple Decoy Administrative Website
In this example, we create a honeypot with a decoy administrative website.

We can create such a honeypot without having to use specialized honeypot software.

To achieve this, we just install a normal server with an operating system and a webserver service.

On this webserver, we create a couple of web pages. The first web page is a page that announces the website as a corporate administrative website and provides a login screen.

The login screen asks for credentials: a potential user has to type in a username and a password. If you use an authentication service in your corporate environment that allows single-sign-on (like Active Directory), this honeypot webserver must be configured so that single-sign-on is disabled: we don't want automatic login attempts when a user visits this page.

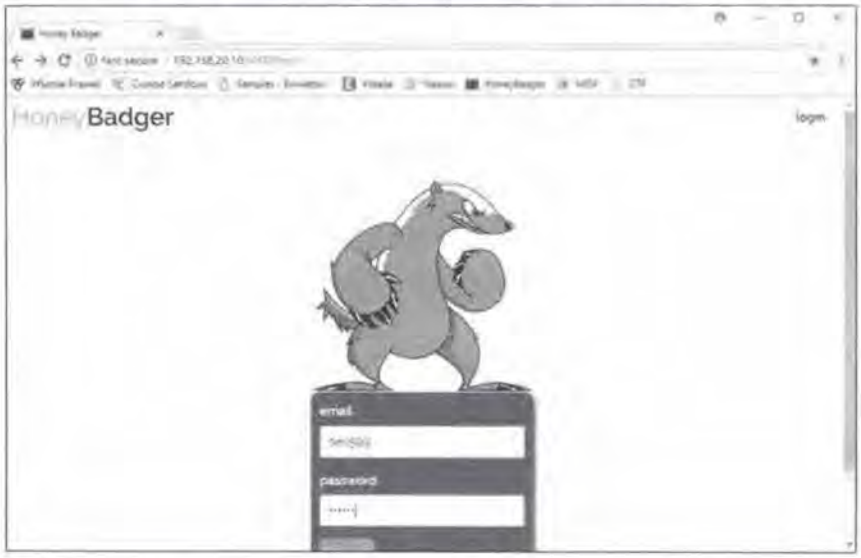
The layout of this first page of the honeypot should be made to attract advanced adversaries and entice them to try to logon to the webserver (with credentials they stole, for example).

The webserver has to be configured to raise an alert for every login attempt, and these alerts should be investigated.

This is a honeypot, it does not contain corporate data or services, normal users should not use it. We only expect adversaries to try to use this webserver.

As a bonus, you could include HoneyBadger's code to track all visits to the website!

Some Example Honeypots – HoneyBadger (ADHD)



HoneyBadger is part of the ADHD framework. It consists of a part of source code that can be added to an existing web site, which will track visitor IP addresses & automatically perform GeoIP localization. It comes with a central admin interface for easy management & follow-up!

We will use it in an upcoming lab!

Some Example Honeypots – HoneyBadger (ADHD)

In the previous slide, we discussed setting up a fake administrative web page to lure adversaries to attempt authentication. But how we can effectively follow up on visits? We could implement our own piece of tracking code, but there are many frameworks already available for this purpose!

An interesting application to do this is "HoneyBadger". HoneyBadger provides an easy-to-use solution, it is part of the Active Defense Harbinger Distribution (ADHD). It consists of two core components:

- A part of source code that can be added to an existing web site, which will track visitor IP addresses & automatically perform GeoIP localization;
- A central management console where you can follow up on "beacons"

You can find more information on HoneyBadger on http://adhdproject.github.io/#!Tools/HoneyBadger.md. We will use it in an upcoming lab!

Some Example Honeypots – Artillery (ADHD)

Another interesting honeypot system that is part of ADHD is **Artillery**. Artillery was developed by TrustedSec / Binary Defense. From its official web site, we have the following features available:

"The purpose of Artillery is to provide a combination of honeypot, file-system monitoring, system hardening, real-time threat intelligence feed, and overall health of a server monitoring-tool; to create a comprehensive way to secure a system."

A few practical use cases for Artillery include:

- Creating listening "honeyports" on the network interface;
- Based upon activity towards these "honeyports", automatically ban violating IPs;
- Monitoring the file system for changes;

Artillery should be deployed as a "honeypot add-on" to existing servers!

Some Example Honeypots – Artillery (ADHD)

Another interesting honeypot system that is part of ADHD is Artillery. Artillery was developed by TrustedSec / Binary Defense. From its official web site, we have the following features available:

"The purpose of Artillery is to provide a combination of honeypot, file-system monitoring, system hardening, real-time threat intelligence feed, and overall health of a server monitoring-tool; to create a comprehensive way to secure a system."

You can find the official documentation here: http://adhdproject.github.io/#!Tools/Artillery.md

Artillery is not your typical honeypot system, as it wasn't designed to be a "standalone-honeypot". Instead, it should be added to a high-value target server that it can protect. A few practical use cases for Artillery include:

- Creating fake listening ports on the target server (so-called "Honeyports");
- Upon activity towards these fake ports, it can automatically alert and ban violating IP addresses;
- Monitor the file system of the target server for changes to important files.

Some Example Honeypots – Kippo Fake SSH

Kippo is a Python program that simulates a SSH server, it is not a real SSH server that gives access to the server. Additionally, Kippo simulates the Linux shell obtained after a successful login.

- Kippo will log all activities, and these can be replayed to see what the attacker did
- Actual commands are not executed: for example, Kippo will download files for the attacker, but not execute them
- If will fake malfunctioning when executing a downloaded file

Kippo is one of the most well-known honeypots out there. A newer evolution (based upon Kippo) is "Cowrie", which is developed by Michel Oosterhof (https://github.com/micheloosterhof/cowrie). Cowrie is also part of ADHD!

Some Example Honeypots – Kippo Fake SSH
In this second honeypot example, we implement a fake SSH service.

This can be done with Kippo: Kippo is a honeypot service that simulates an SSH server. It is written in Python, it is not a real SSH server that gives access to the server.

Kippo can be installed on a normal server, and we recommend using the normal SSH port (port 22) for Kippo. This means that to use Kippo on a Linux machine, the real SSH service should be put on another port or disabled. Kippo should also be able to run on a Windows machine, but this is only recommended if you have SSH servers running on your corporate machines. Otherwise, a single Windows machine with SSH server will stand out to the attackers, and they could realize that this is, in fact, a honeypot.

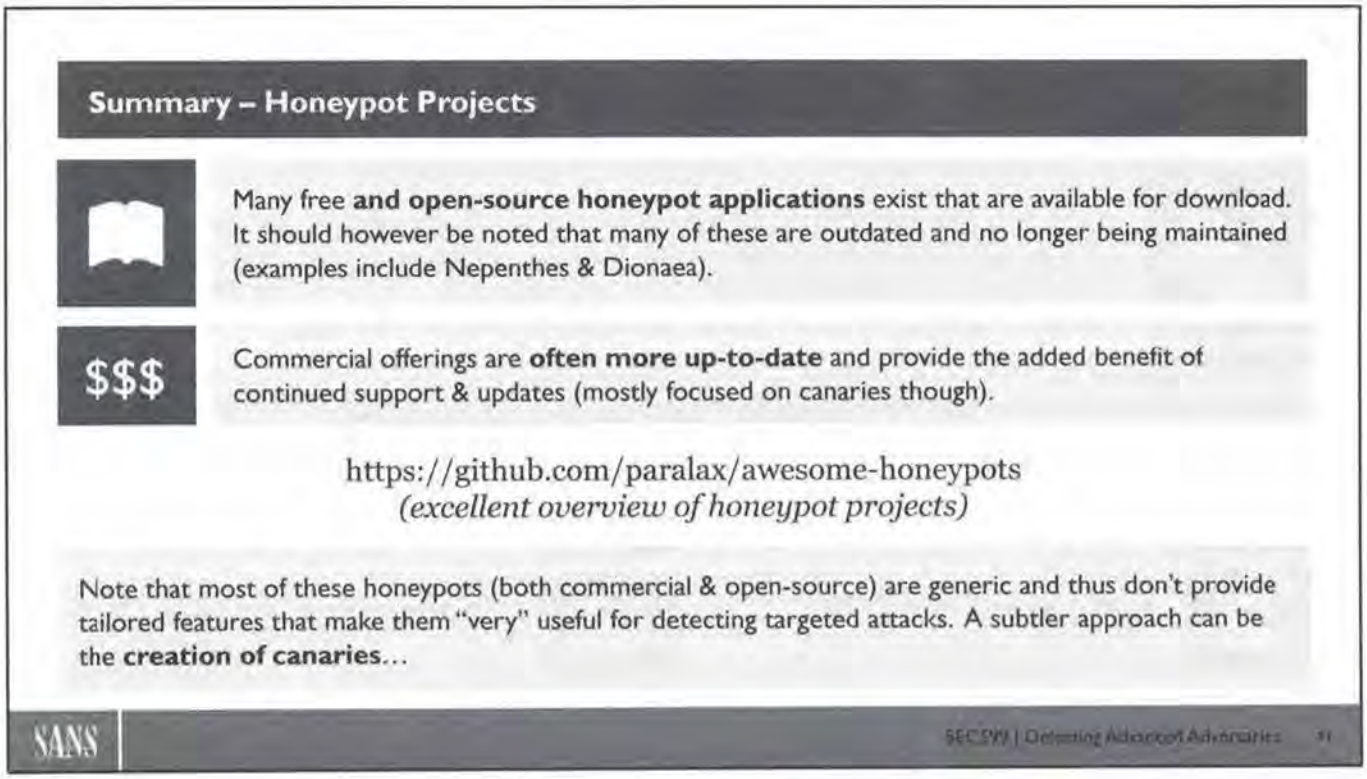
After a user logs on to Kippo via the normal SSH logon procedure, the user will be presented with a classic Linux command-line shell. This is not a real shell (bash for example), but it is simulated by Kippo.

The commands that are typed by the user are also simulated by Kippo, to make it look like a real shell. For example, the attacker can issue a wget or curl command to download his attack tools on the server. Kippo will simulate this by effectively downloading the files and storing them for later review by the Kippo administrators.

Kippo will simulate the presence of the downloaded file in the current directory, but when the attacker tries to run the tool he downloaded, Kippo will not execute the tool but present an error message that tries to persuade the attacker that the file was corrupted during the download.

Kippo will log all activities, and these can be replayed to see what the attacker did. For use as a honeypot in our corporate network, Kippo must be configured to raise alerts when logons are performed.

Kippo is one of the most well-known honeypots out there. A newer evolution (based upon Kippo) is "Cowrie", which is developed by Michel Oosterhof (https://github.com/micheloosterhof/cowrie).



Available Honeypot Projects

There are a lot of free and open-source honeypot applications available for download. Not all of them are properly designed and safe to use in a corporate environment.

Many of these open-source honeypot applications are outdated or no longer maintained. Research into honeypots and development of honeypot applications was popular when Lance Spitzner published his book "Honeypots: Tracking Hackers" in 2003, but over the years, interest has faded a bit. This has resulted in many open-source honeypot projects being abandoned or no longer actively maintained. Examples of these are the well-known Nepenthes and Dionaea open-source honeypots. These days, commercial honeypot offerings are more up-to-date, we will see this when we discuss canaries.

An interesting overview of current honeypot projects can be found at https://github.com/paralax/awesome-honeypots.

Many of these honeypots are not suitable for our purposes: they are designed as "research" honeypots, to research new malware and new zero-day exploits. They are not designed with corporate attackers in mind. Furthermore, most of these honeypots (both commercial & open-source) are generic and thus don't provide tailored features that make them "very" useful for detecting targeted attacks. A subtler approach can be the creation of canaries...

Introducing Canaries

A canary is decoy data that looks enticing to adversaries. The name canary refers to the canaries used in coalmines to alert miners for poisonous gas emanations. We place canaries in places where they can be found and accessed by our adversaries when they search through our corporate assets. Interacting with a canary will trigger an alert.

Canaries can take many different forms, some examples include:

A document, possibly rigged with an alerting system when opened

Unique passwords inside a database

A user account (typically a fake administrator), never to be used

Fake domain hashes can be inserted in the LSA process ("HoneyHashes")

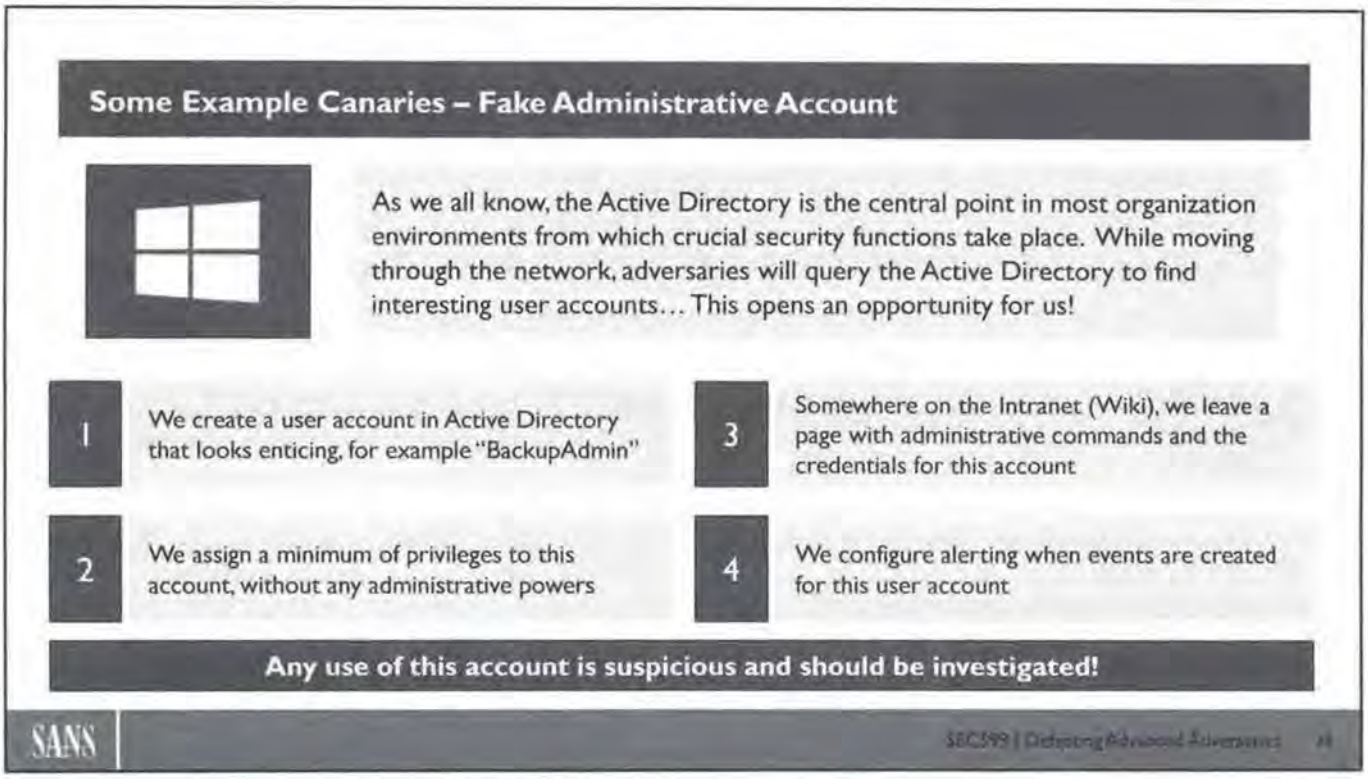
Introducing Canaries

Another popular method to trick adversaries are canaries. A canary is decoy data that looks enticing to adversaries. We place canaries in places where they can be found and accessed by our adversaries when they search through our corporate assets. Interacting with a canary will trigger an alert. The name canary refers to the canaries used in coal mines to alert miners for poisonous gas emanations.

Canaries can take many different forms, some examples include:

- A document, possibly rigged with an alerting system that generates an alert when opened
- A user account (typically a fake administrator), that is never to be used by corporate staff
- Unique passwords inside a database that are never used as real passwords
- Fake domain hashes can be inserted in the LSA process ("HoneyHashes")
- ...

We will discuss a few of these techniques in the next few slides and use them in a few labs.



Some Example Canaries – Fake Administrative Account
As we all know, the Active Directory is the central point in most organization environments from which crucial security functions take place. While moving through the network, adversaries will query the Active Directory to find interesting user accounts… This opens an opportunity for us!

We can create a canary by adding a user account in Active Directory that looks enticing to our potential attackers, for example, BackupAdmin or any other account name that might look important in your corporate environment and would be attractive to attackers to compromise.

Although the metadata of this account (like the account name) should indicate that it is an administrative account, we do only assign a minimum of privileges to this account, without any administrative powers. This account is designed to be compromised: when it is compromised, we don't want to provide an advantage to our attackers when they use this account. Using this account must alert us, it must not be possible to leverage this account.

We configure alerting when events are created for this user account: whenever this account is used (for example logon attempts), we want to be warned. We do this by configuring alerts in our monitoring systems each time events are created by Windows for this particular account.

To increase the chance that attackers will try to use this account, we can plant information on our corporate systems to entice attackers. For example, somewhere on the Intranet (Wiki), we leave a page with administrative commands and the credentials for this account.

Some Example Canaries – HoneyHash

In 2015, SANS Instructor & ISC Handler Mark Baggett wrote an interesting diary post on "Detecting Mimikatz Use On Your Network". The technique he describes involves the creation of fake hashes ("**honeyhashes**") in the **LSA process memory**.

As we all know, Mimikatz is a highly popular tool that is being used both by penetration testers / red teamers and real adversaries. One of its key functions is to dump password hashes belonging to authenticated users from the LSA process memory.

This technique has been further developed and as part of the Empire framework, a "New-HoneyHash.ps1" PowerShell script was created, which allows for the creation of fake hashes in the LSA process memory!

Original post: https://isc.sans.edu/forums/diary/Detecting+Mimikatz+Use+On+Your+Network/19311/

SANS

Some Example Canaries – HoneyHash

In 2015, SANS Instructor & ISC Handler Mark Baggett wrote an interesting diary post on "Detecting Mimikatz Use On Your Network". The technique he describes involves the creation of fake hashes ("honeyhashes") in the LSA process memory. You can read Mark's original ISC diary post here:

Original post: https://isc.sans.edu/forums/diary/Detecting+Mimikatz+Use+On+Your+Network/19311/

As we've seen during day 4 of this course, Mimikatz is a highly popular tool that is being used both by penetration testers / red teamers and real adversaries. One of its key functions is to dump password hashes belonging to authenticated users from the LSA process memory. The idea here is, of course, to trick adversaries using Mimikatz by providing them with fake hashes for accounts that don't actually exist. We can then monitor attempted use of these credentials. Note that this attack technique will not only trick Mimikatz but also other tools that attempt to extract password hashes from memory.

This technique has been further developed and as part of the Empire framework, a "New-HoneyHash.ps1" PowerShell script was created, which allows for the creation of fake hashes in the LSA process memory!

Some Example Canaries – Fake Documents

A second example of a canary is much easier: creating a fake confidential document. The idea this time is to create an attractive document that adversaries would like to steal. This time, however, we have to take into account that the document could leave our environment…

1. We create a document that looks enticing, like confidential-project-planning-v1.doc
2. We add a unique string to this document, for which we configure Google alerts
3. We add VBA macro code to this document that will issue a web request with this unique string to a webserver under our control
4. We add alerting to the logging of the web or DNS server

Any interaction with this document should be investigated!

Example – Fake Confidential Document

A second example of a canary is much easier: creating a fake confidential document. The idea this time is to create an attractive document that adversaries would like to steal. This time, however, we have to take into account that the document could leave our environment…

In our example, we create a Word document, but canaries of this type can also be created with other formats, for example, PDF:

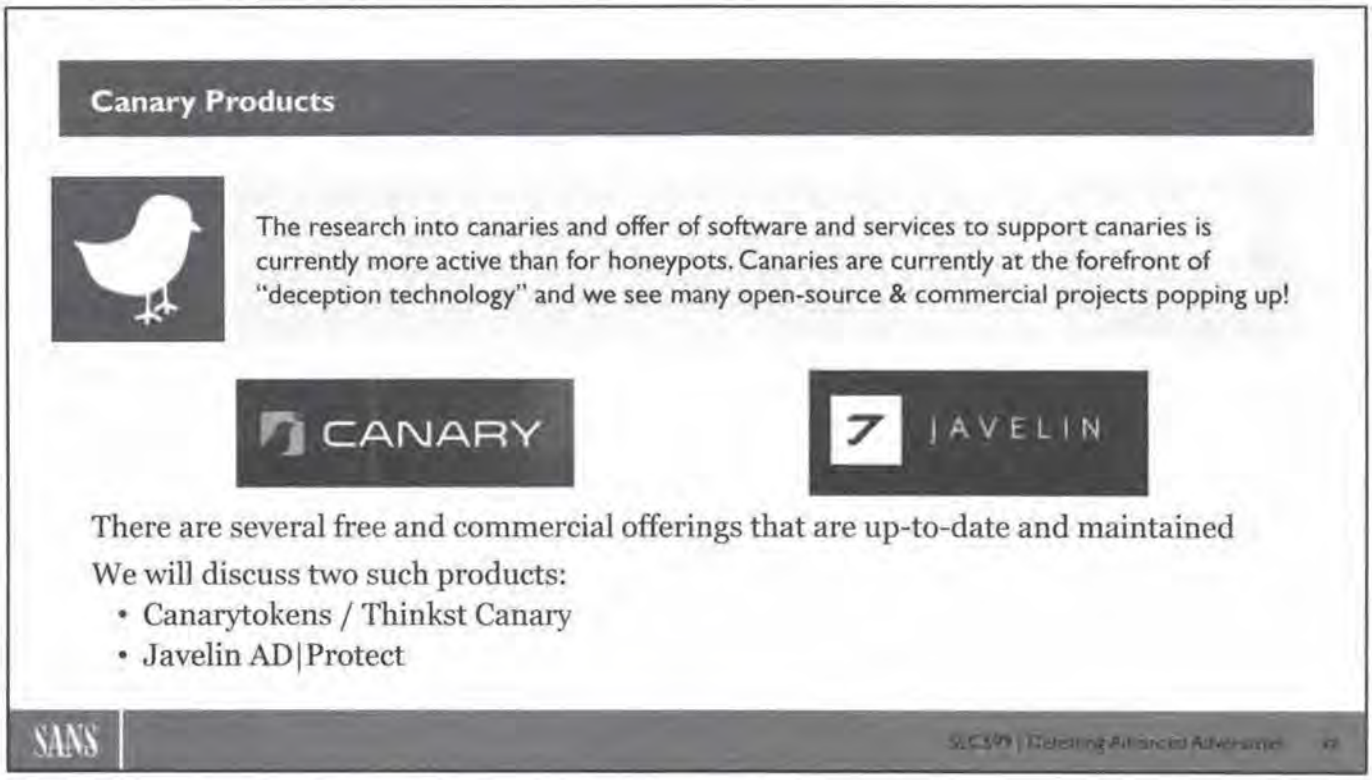
1. We create a Word document with a name that looks enticing, like confidential-project-planning-v1.doc.
2. Next, we want to generate alerts whenever this document is copied or accessed (for example, when it is opened). We can achieve this by adding a unique string to this document (for example, corporate asset JDE0345HND), and then we configure alerts when we see content with this string moving inside our network or even appearing on the Internet.

   Alerts for movements inside our network can be done with IDS for example, we create a simple rule to alert whenever the string JDE0345HND is found. This rule should certainly be implemented in the IDSs on the perimeter of our corporate network so that we are alerted when this canary document is exfiltrated.

   We can also configure Google alerts (or other alerting services) whenever this string is found on a page on the Internet (and of course, indexed by Google). For example, if the content of this document would appear on pastebin we want an alert.

3. Finally, we add VBA macro code to this document that will issue a web request with this unique string to a webserver under out control whenever the document is opened. We add alerting to the logging of the webserver so that we are alerted whenever this request is made to the webserver.

Other methods besides VBA macros exist, for example, templates or embedded, linked objects.



Canary Products

The research into canaries and offer of software and services to support canaries is currently more active than for honeypots. Canaries are currently at the forefront of "deception technology" and we see many open-source & commercial projects popping up!

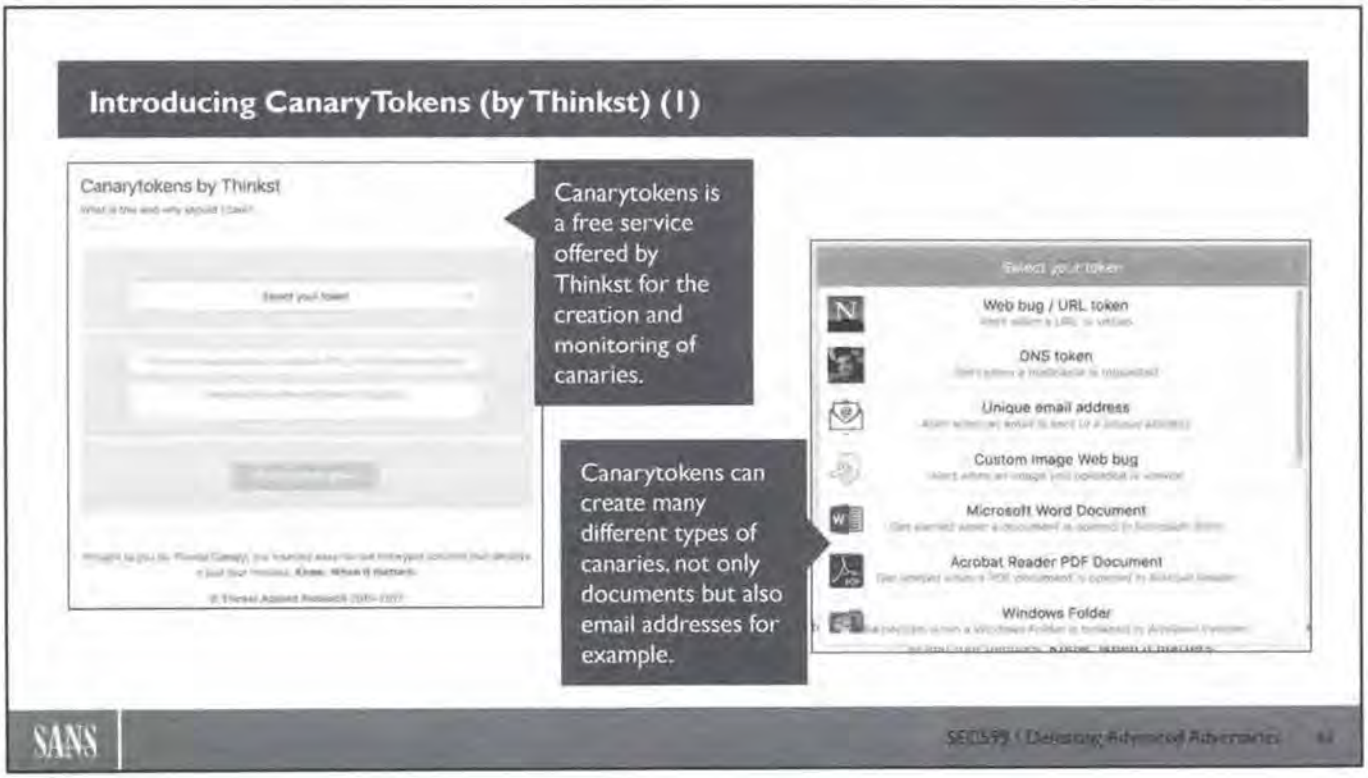
There are several free and commercial offerings for canaries that are up-to-date and actively maintained.

We will discuss 2 products:

- Canarytokens / Thinkst Canary
- Javelin AD|Protect

Canarytokens is a free service offered by Thinkst to create canaries and be alerted whenever these canaries are used. Several types of canaries can be created. Thinkst Canary is Thinkst's commercial offering for canaries. Not only do they provide canaries, but also honeypots.

Javelin AD|Protect is a commercial canary application for Active Directory. When deployed, it will create different types of canaries inside your corporate Active Directory infrastructure and alert you when these canaries are used by attackers.



Introducing CanaryTokens (by Thinkst) (1)
The screenshot above (left) shows the free Canarytokens service as it is provided by Thinkst.

This page can be used to create your own, personalized canary together with the data necessary to alert you whenever this canary is used.

First, you have to select the type of canary you want to create. Many types are possible, varying from documents to account to email addresses, we will discuss this on the next page.

Then you have to provide an email address or URL. This will be used to alert you whenever your canary is accessed. When your canary is used by attackers, an event will be sent to the Canarytokens servers. This is typically a web request or a DNS name resolution. When the Canarytokens servers get this particular event for your canary, they will alert you via the email address or URL you provided.

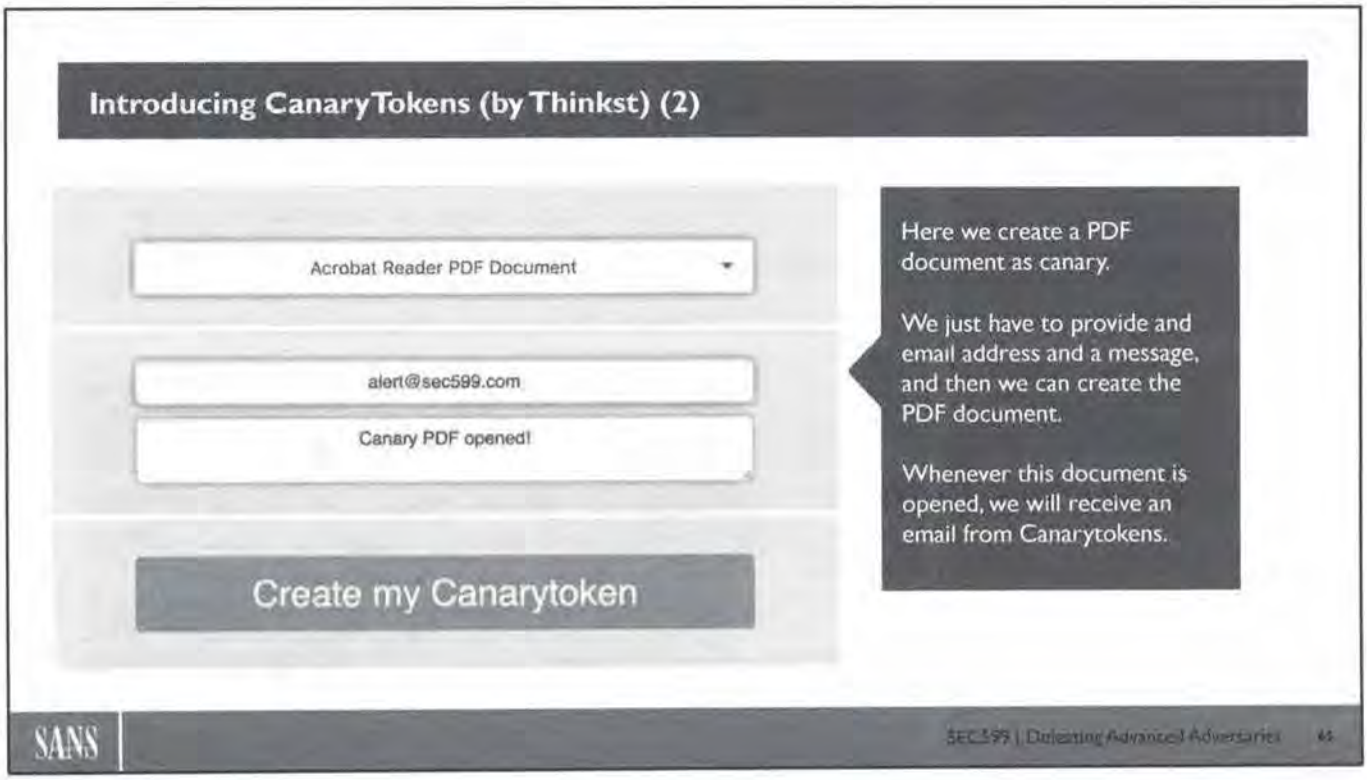
As you can create many canaries with Canarytokens, you are also given the option to provide a reminder note that will help you remind why you created this particular canary. This reminder note will be included in the alert you receive from the Canarytokens servers.

Tricking the adversary
Here we can see a partial list (screenshot right) of the type of canaries you can create using the free Canarytokens service.

Canarytokens can create:
- Web bugs
- DNS tokens
- Email addresses
- Custom images
- Word documents
- PDF documents
- Windows folders

- Custom EXEs
- Canary for cloned website
- SQL server
- QR code
- SVN
- AWS keys

Whenever one of these types is used, you will receive an alert.



Introducing CanaryTokens (by Thinkst) (2)
This is the page you will see when you have completed the fields for the canary you want to create with Canarytokens.

In this example, we create an Acrobat Reader PDF document with a canarytoken that will send an email to address alert@sec599.com whenever the PDF document is opened.

The reminder note "Canary PDF opened!" will be included in the email when an alert is triggered.

By clicking "Create my canarytoken", the PDF is created and available for download, and entries will be created in Canarytokens database to alert you.

The method for PDFs is the following: the PDF that is created by Canarytokens contains a unique URL that is automatically requested whenever the PDF is opened with Adobe Reader. The user will be warned and asked for authorization to execute the web request, however, the DNS request for the domain included in the URL will be issued without any warning. Canarytokens servers will receive this unique DNS request and alert you via email.

Introducing CanaryTokens (by Thinkst) (3)

Thinkst Canary has a commercial honeypot solution, that allows you to create and deploy honeypots in your corporate infrastructure like Windows file servers, Linux web servers ...

Thinkst Canary is the commercial offering of Thinkst. Next to honeypots described above, another advantage it has when compared to the free CanaryTokens service is that it allows you to configure a canary architecture that does not rely on the infrastructure set up by Thinkst, thus, offering increased confidentiality!

Introducing CanaryTokens (by Thinkst) (3)
Canarytokens is a free service and as such has limitations that can impair its use by a large corporation.

For example, corporate clients will want:
- more options for alerting than email and URL
- SLAs to guarantee a certain level of service
- Strict confidentiality: with Canarytokens, the alerts are processed by Thinkst's servers
- ...

These requirements can be accommodated by opting for a commercial solution. Thinkst Canary is the commercial offering of Canarytokens, and this will, for example, allow you to host your own alerting servers so that you don't have to rely on Thinkst and that they are also not aware of potential breaches.

Thinkst Canary not only offers canaries like Canarytokens but also honeypots.

Thinkst Canary has a commercial honeypot solution, that allows you to create and deploy honeypots in your corporate infrastructure like Windows file servers, Linux web servers, ...

Introducing Javelin

When AD|Protect is deployed in your corporate environment, it will create different types of canaries inside your corporate Active Directory infrastructure and alert you when these canaries are used by attackers.

With a few queries to the Active Directory at the point of breach, an attacker can obtain all information to move throughout the topology.

AD | Protect controls the attacker's perspective and provides visibility to Dark Corners that the attacker favors.

Javelin AD | Protect is canary software that is deployed on Active Directory Domain Controllers.

When attackers query Active Directory looking for assets, AD | Protect will respond with decoy assets, making the search for valuable assets more difficult and alert when decoy assets are interacted with.

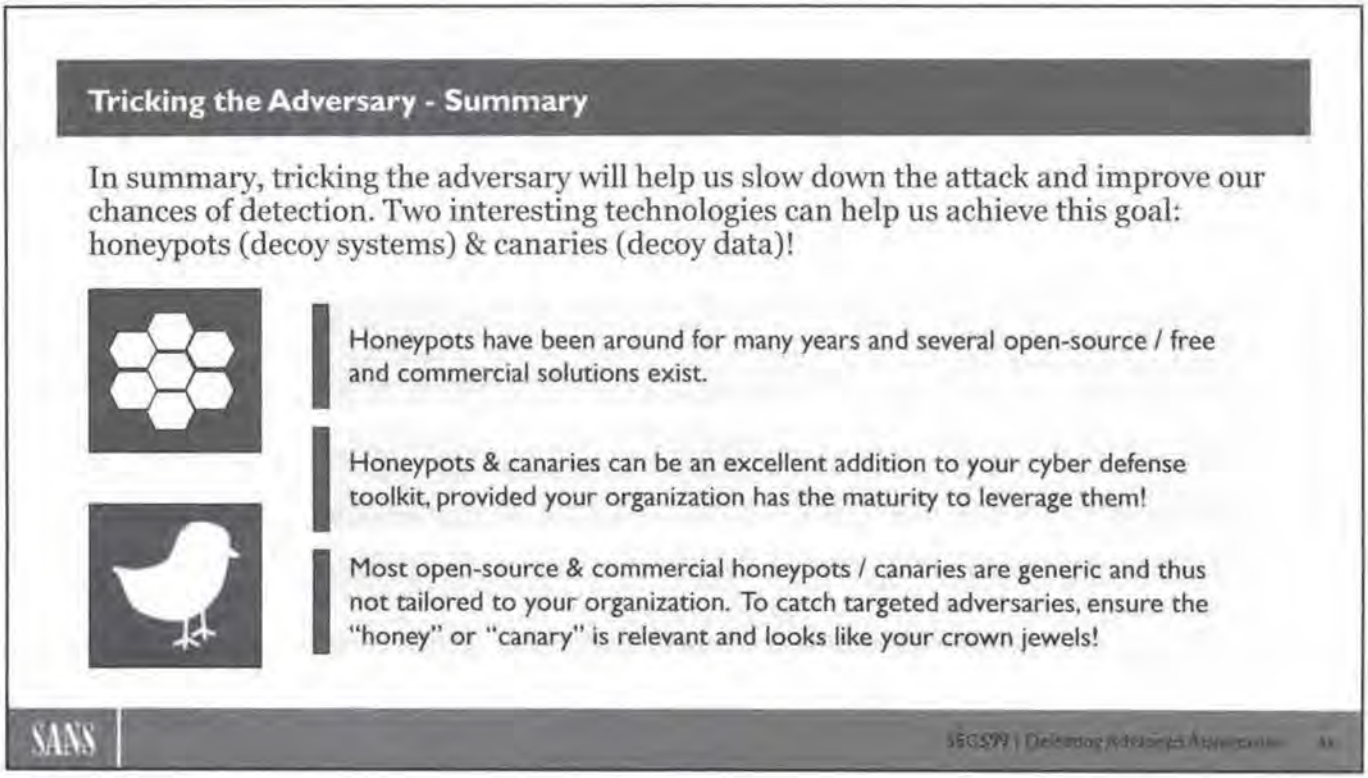
Introducing Javelin
Another commercial canary solution for Active Directory we want to mention is Javelin AD|Protect.

When AD|Protect is deployed in your corporate environment, it will create different types of canaries inside your corporate Active Directory infrastructure and alert you when these canaries are used by attackers.

AD|Protect will not only create canaries like fake administrative accounts and fake Windows servers, but it will offer several other deception technologies.

Active Directory is a repository (directory) for all your corporate (Windows) assets, and can thus be queried to obtain a list of assets. For example, any domain member (a user or a machine like a workstation) can query an Active Directory domain controller to obtain a list of all computers that are members of the Active Directory domain. This function does not require special privileges, all members can query information like this. For example, the Petya/Notpetya ransomware worm would use this functionality to obtain a list of all computers on the network to attack.

For example, AD|Protect can be configured to interfere with this functionality: it can inject a huge number of fake computers in the query results. When attackers query Active Directory domain controllers to obtain a list of all computers, AD|Protect will inject fake computers so that the attackers don't know what results are real and fake, and thus slow down their attack. For example, if you have 1000 workstations in your Active Directory domain, AD|Protect can make it look like there are 10000 workstations in your domain. Not only does this confuse the attackers, but it increases the chances that attackers will try to access a fake workstation significantly, and then AD|Protect will generate alerts.

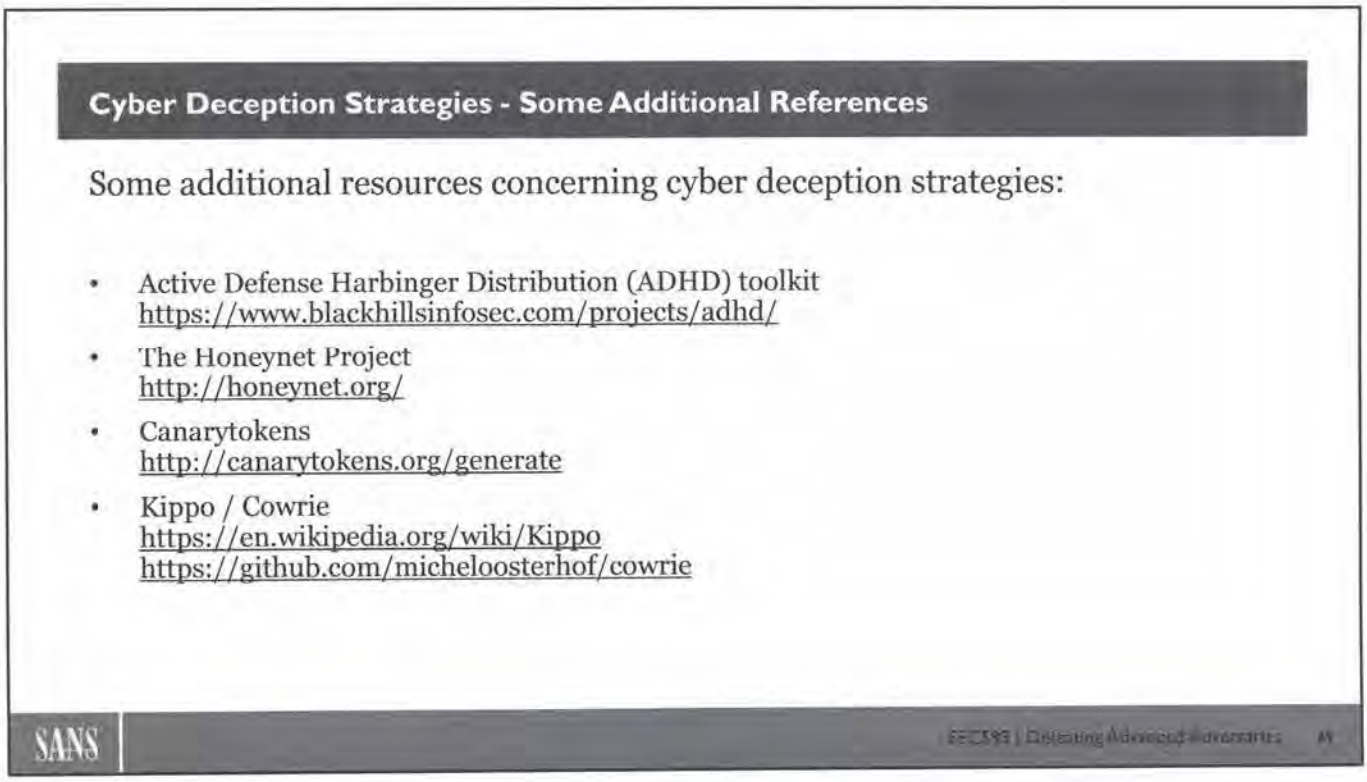


Tricking the Adversary - Summary

In summary, tricking the adversary will help us slow down the attack and improve our chances of detection. Two interesting technologies can help us achieve this goal: honeypots (decoy systems) & canaries (decoy data)! Honeypots are decoy systems: they simulate systems or services and are designed to try to attract the attention of our adversaries. Canaries are decoy data: they simulate corporate data and documents that look enticing to our adversaries and are designed to alert us when this data is consumed.

Here are a few closing thoughts:

- Honeypots have been around for many years and several open-source / free and commercial solutions exist;
- Honeypots & canaries can be an excellent addition to your cyber defense toolkit, provided your organization has the maturity to leverage them!
- Most open-source & commercial honeypots / canaries are generic and thus not tailored to your organization. To catch targeted adversaries, ensure the "honey" or "canary" is relevant and looks like your crown jewels!



Cyber Deception Strategies - Some Additional References
Some additional resources concerning tricking the adversary:

The Honeynet ProjectActive Defense Harbinger Distribution (ADHD) toolkit
https://www.blackhillsinfosec.com/projects/adhd/

http://honeynet.org/

Canarytokens
http://canarytokens.org/generate

Kippo
https://en.wikipedia.org/wiki/Kippo
https://github.com/micheloosterhof/cowrie

Course Roadmap

- Day 1: Knowing the adversary, knowing yourself
- Day 2: Averting Payload Delivery
- Day 3: Preventing Exploitation
- Day 4: Avoiding Installation, foiling Command & Control & thwarting lateral movement
- **Day 5: Exfiltration, Cyber Deception & Incident Response**
- Day 6: APT Defender Capstone

SEC599.5

Data exfiltration
- Typical data exfiltration strategies
- Exercise: Detecting data exfiltration using Suricata

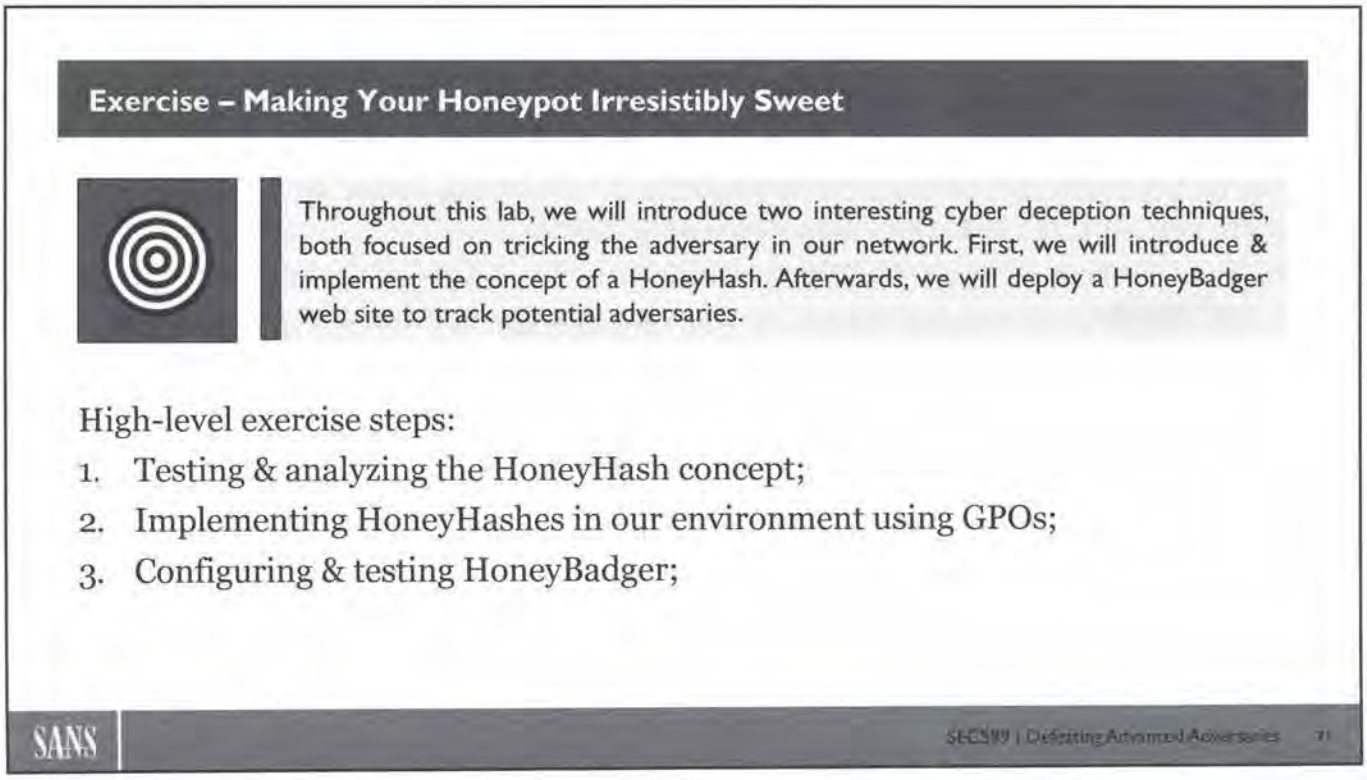
Cyber deception strategies
- Tricking the adversary
- Exercise: Making your honeypot irresistibly sweet

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

Patrolling your network
- Proactive threat hunting strategies
- Exercise: Hunting your environment using OSQuery / ELK

Incident response
- Incident response process
- Exercise: Responding to an incident using GRR

This page intentionally left blank.



Exercise – Making Your Honeypot Irresistibly Sweet
Throughout this lab, we will introduce two interesting cyber deception techniques, both focused on tricking the adversary in our network. First, we will introduce & implement the concept of a HoneyHash. Afterwards, we will deploy a HoneyBadger web site to track potential adversaries.

The following are high-level exercise steps:

1. Testing & analyzing the HoneyHash concept;
2. Implementing HoneyHashes in our environment using GPOs;
3. Configuring & testing HoneyBadger;

For additional guidance & details on the lab, please refer to the LODS workbook.

Exercise – Making Your Honeypot Irresistibly Sweet – Conclusions

Throughout this lab, we used a combination of techniques to set up "traps" in our environment that could help slow down an adversary:

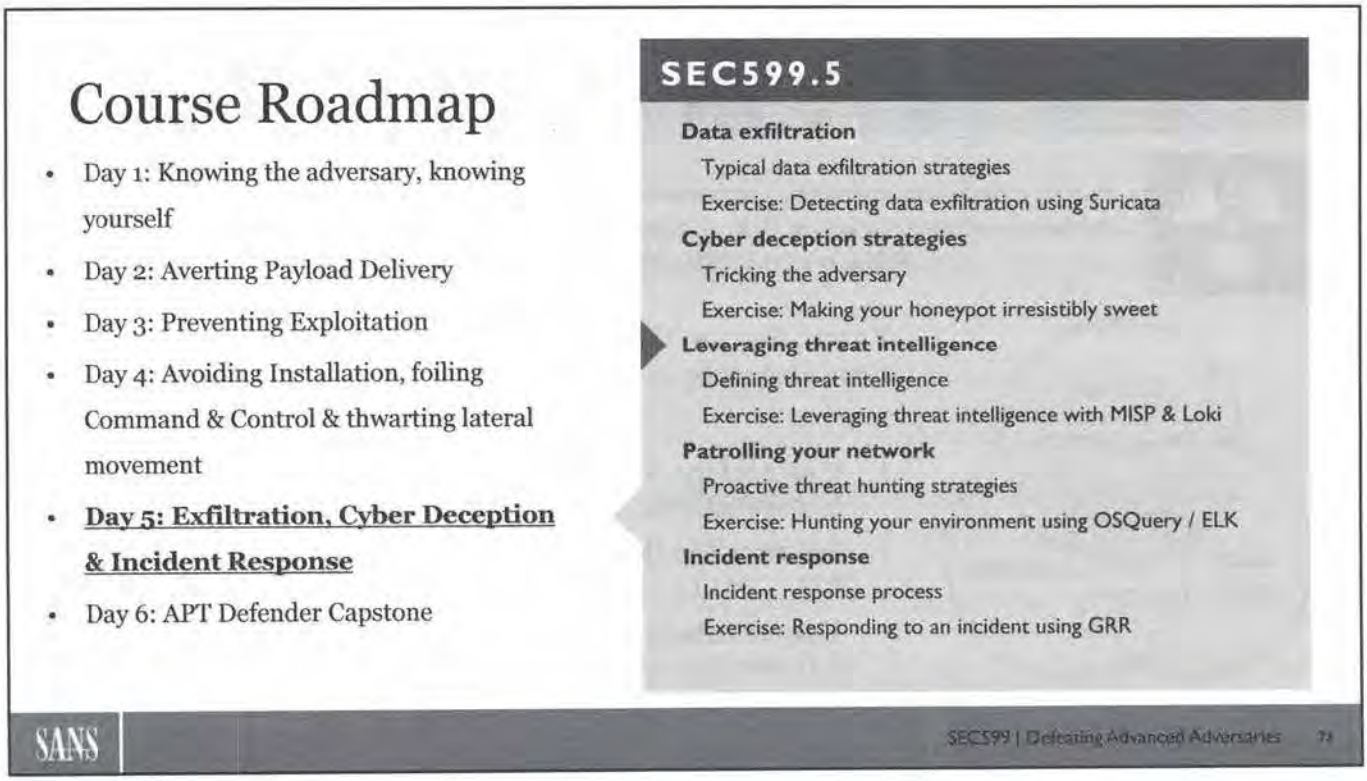
We used a slightly adapted variant of the "New-HoneyHash.ps1" script to implement GPO's to generate fake administrative hashes on all systems in our AD environment.

We used HoneyBadger to demonstrate how easy a "tracking" web site can be set up to track who visits a certain web page.

Exercise – Making Your Honeypot Irresistibly Sweet – Conclusions
Throughout this lab, we used a combination of techniques to set up "traps" in our environment that could help slow down an adversary:

- We used a slightly adapted variant of the "New-HoneyHash.ps1" script to implement GPO's to generate fake administrative hashes on all systems in our AD environment. In our specific case, we created a user called "svcadmin", which would appear to be a technical account with administrative privileges.
- We used HoneyBadger to demonstrate how easy a "tracking" web site can be set up to track who visits a certain web page.

As you can see, we've added some deception technology without spending a time!



This page intentionally left blank.

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

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

What Is Threat Intelligence? (2)

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.

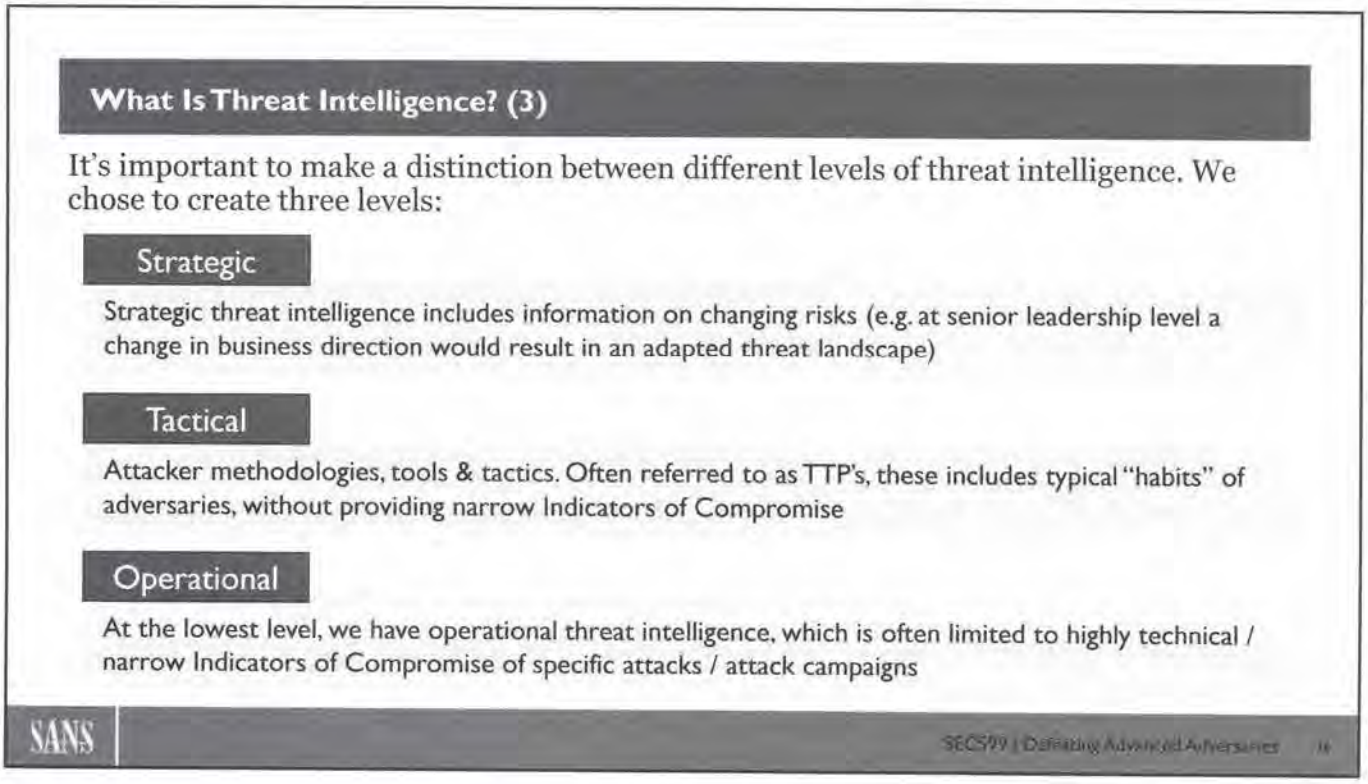
Please note that there's a large number of threat intelligence definitions available. According to Gartner for example, threat intelligence is:

"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.

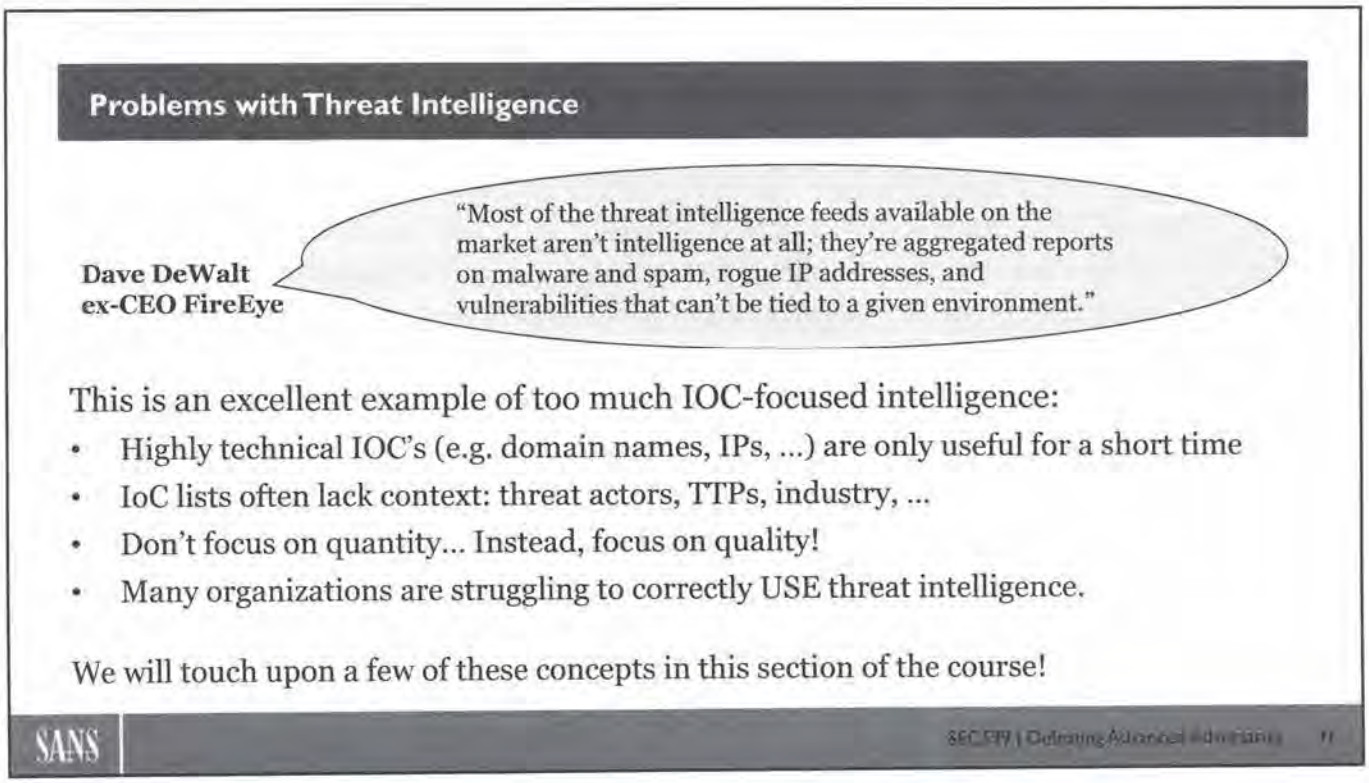


What Is Threat Intelligence? (3)

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)
- Tactical intelligence: Attacker methodologies, tools & tactics. Often referred to as TTP's, these include typical "habits" of adversaries, without providing narrow Indicators of Compromise
- Operational intelligence: At the lowest level, we have operational 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 & operational threat intelligence!



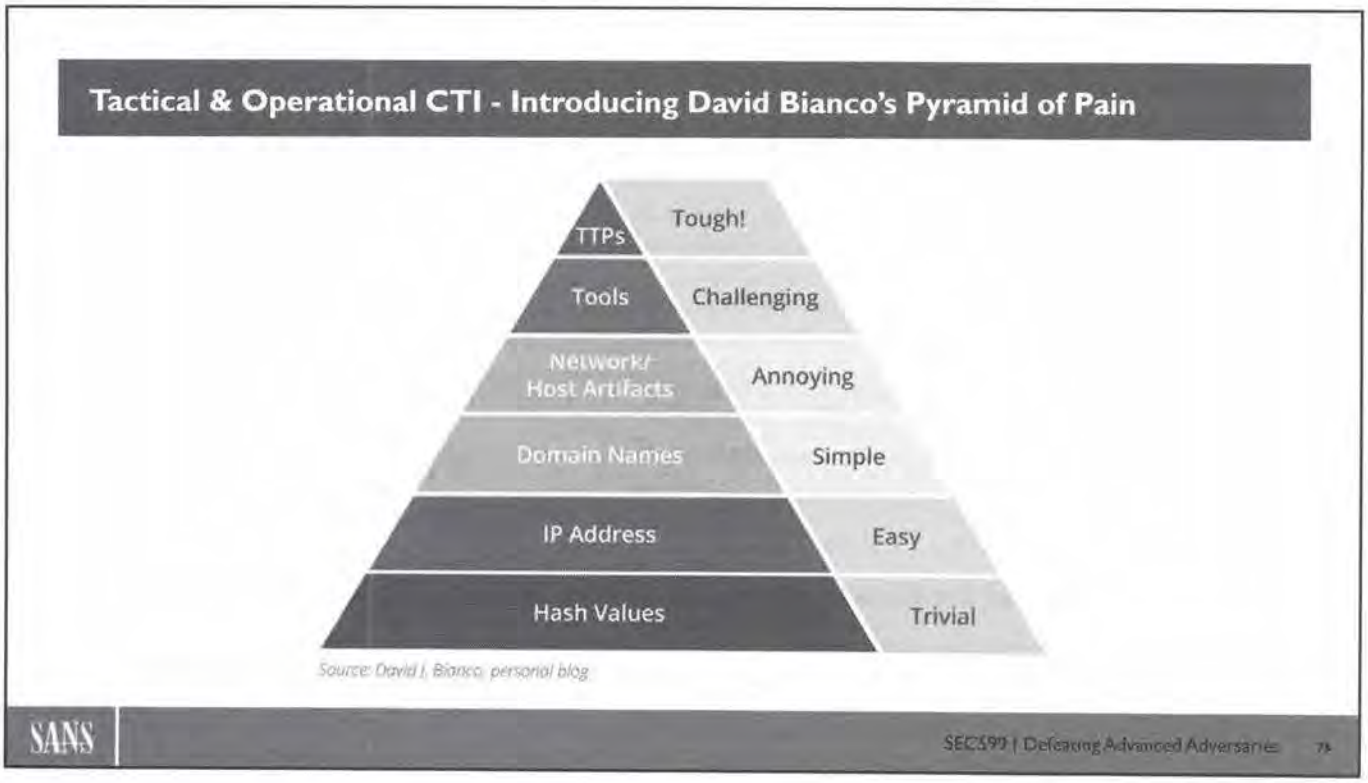
Problems with Threat Intelligence
According to Dave DeWalt, ex-CEO of FireEye:

"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 IOC's (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!



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 affective they are at combatting advanced adversaries. David calls this illustration the "Pyramid of Pain".

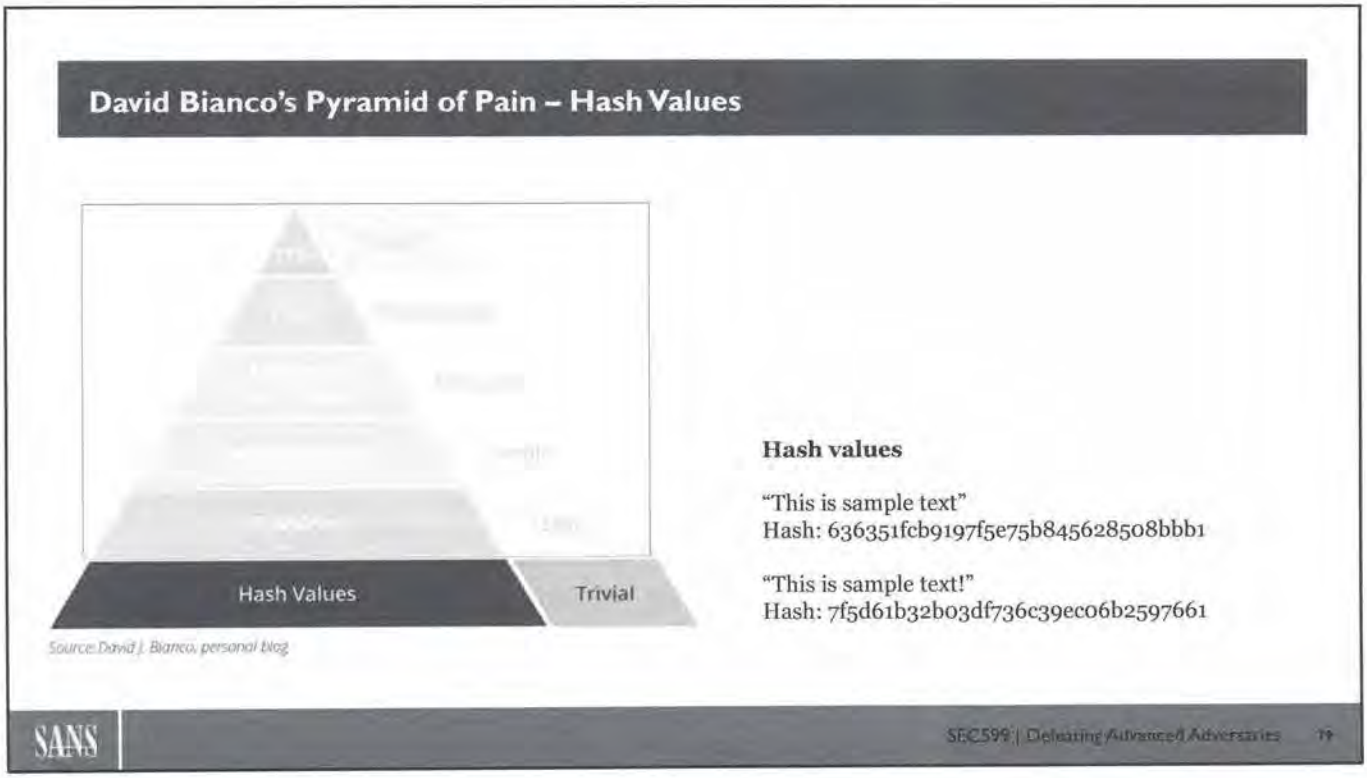
It represents the amount of "pain" you can inflict on advanced attackers by using a certain type of indicator and denying them the use of attacks that can be detected by these indicator types.

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 Address
- Hash Values

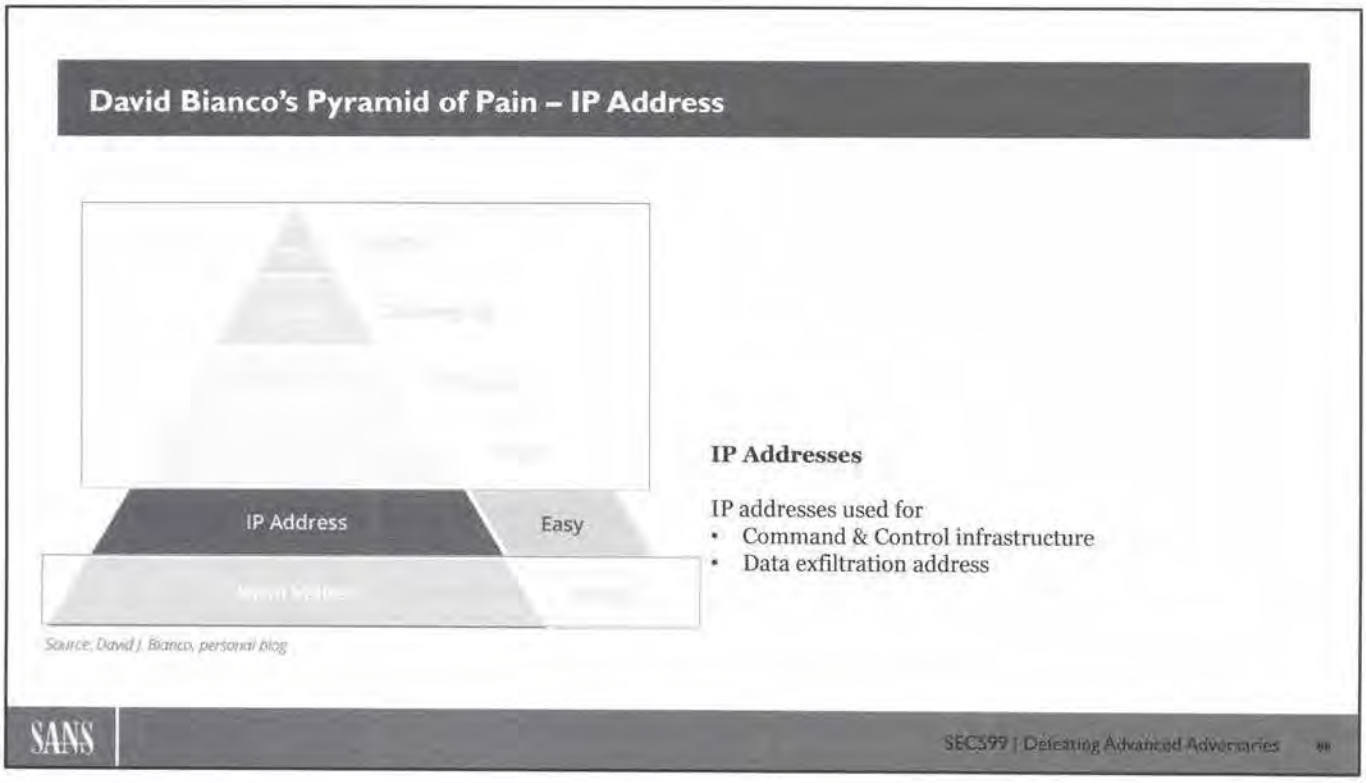


David Bianco's Pyramid of Pain – Hash Values
We start at the bottom of the pyramid with hash values.

Hash values are cryptographic hash values like the MD5 value 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.

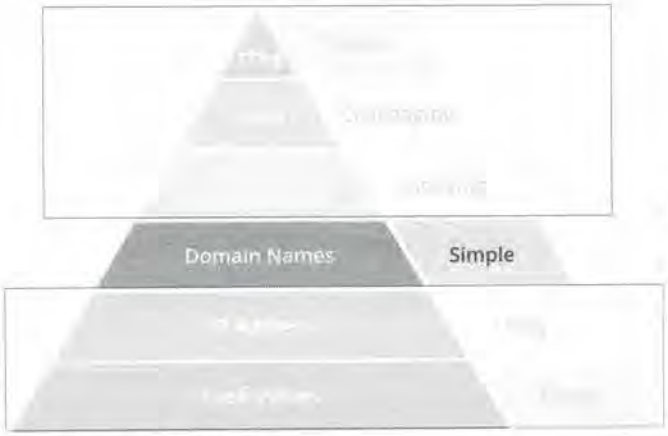


David Bianco's Pyramid of Pain – IP Address
Next is detection based on IP addresses.

IP addresses will be used in network communications for command and 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



Source: David J. Bianco, personal blog

Domain names

Domain names and sub domain names
- Evil.com
- This.is.evil.com

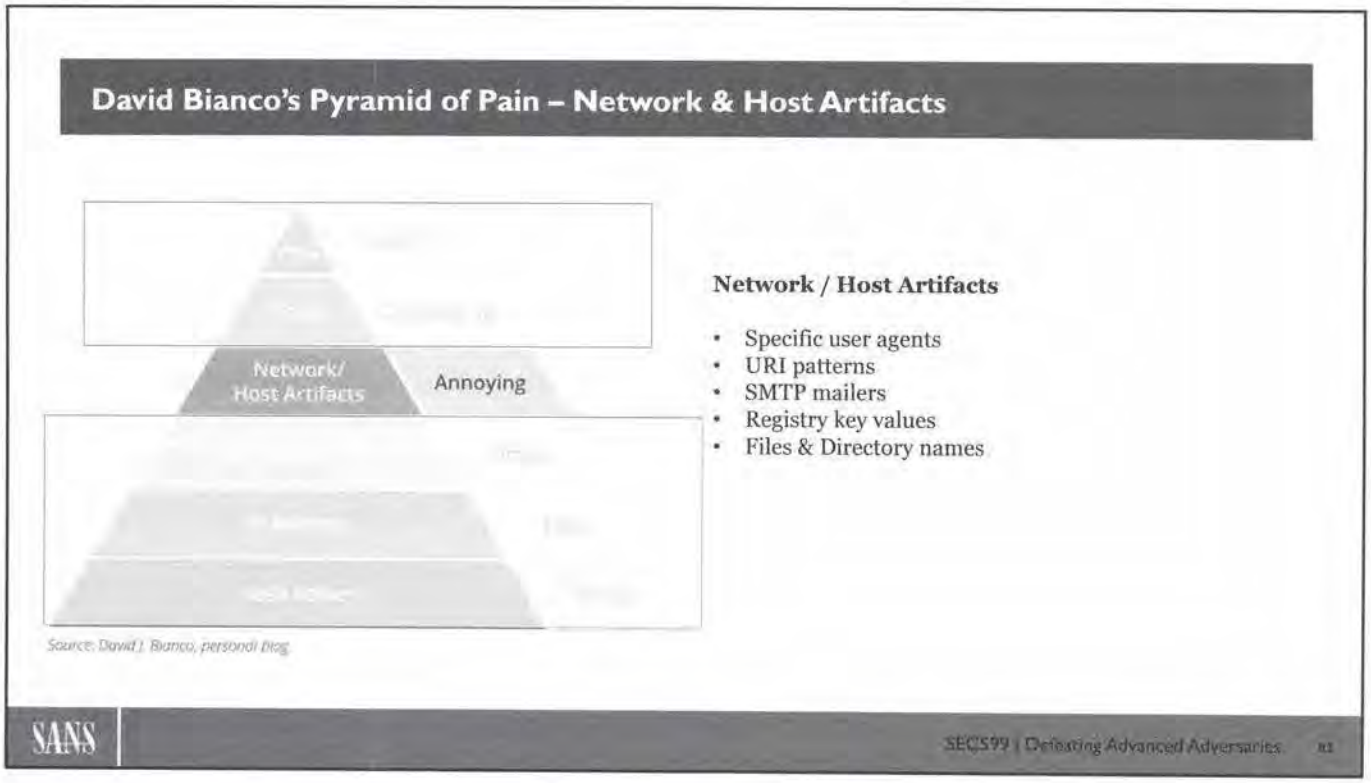
David Bianco's Pyramid of Pain – Domain names
Indicators based on domain names are better quality indicators than indicators based on IP addresses.

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.



David Bianco's Pyramid of Pain – Network & 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 & 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.

David Bianco's Pyramid of Pain – Tools

Source: David J. Bianco, personal blog

Tools

Things attackers bring with them
- Password crackers
- Post-compromise utilities
- Exploits

Utilities attackers use to create malicious documents

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 to 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 – TTP's

Source: David J. Bianco, personal blog

TTP: 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 – TTP's

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
- …

How Do You Obtain Threat Intelligence?

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 & 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 & your environment in the process
- CON: Requires expertise & effort

In an ideal scenario, you combine the three options!

SANS

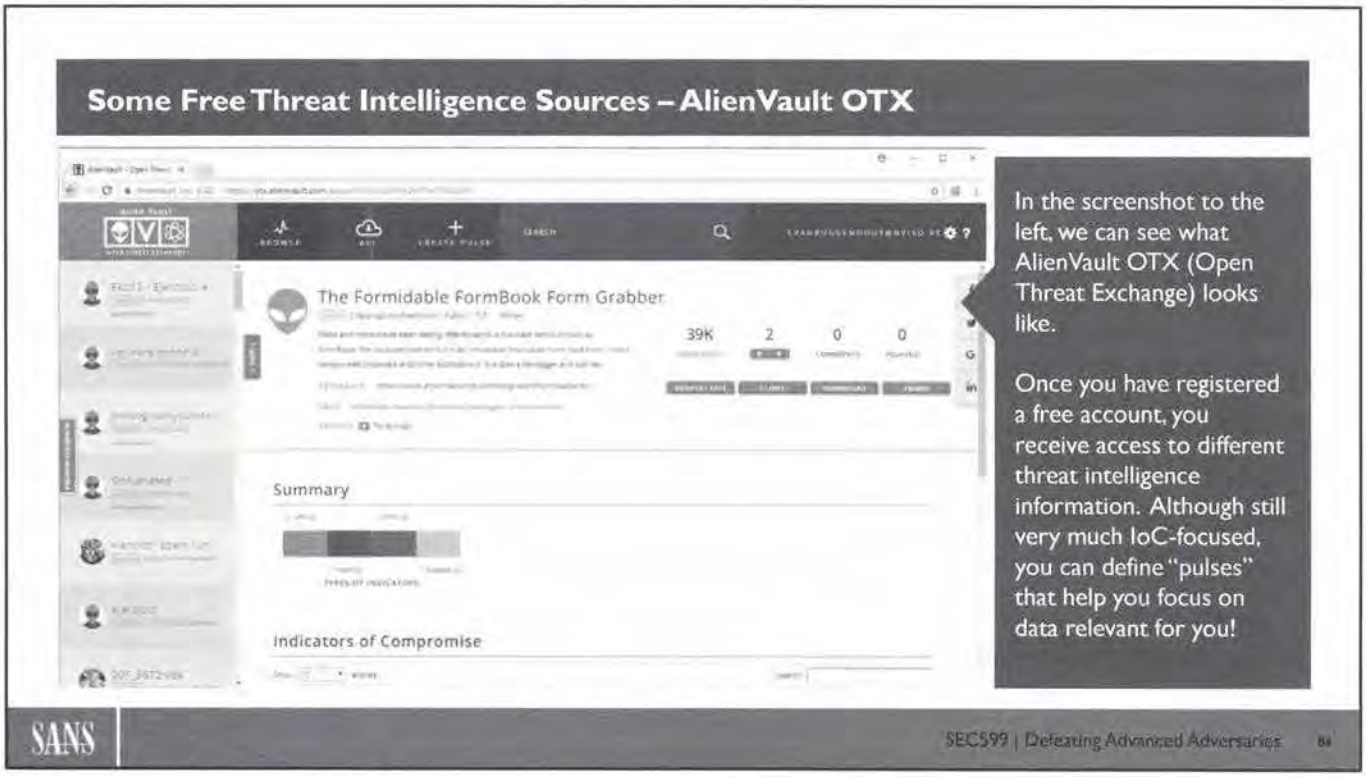
How Do You Obtain Threat Intelligence?
There are several methods to obtain threat intelligence that can be used to help defend your corporate environment. We will list some of the most commonly available options:

Option 1: Buy a commercial threat intelligence feed
- PRO: This does not require a lot of effort from your side, you basically "out-source" 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 & 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 & expertise to do well.

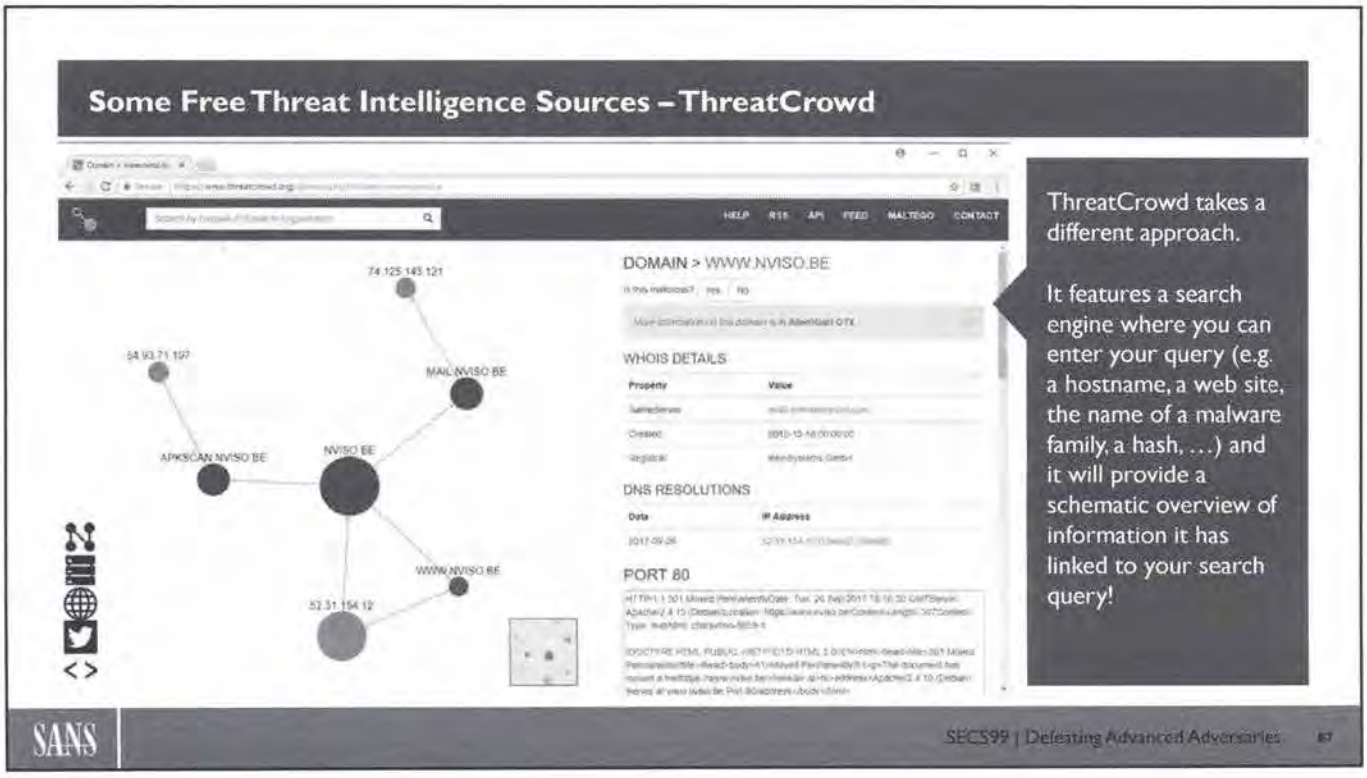
In an ideal scenario, you combine all three options!



Some Free Threat Intelligence Sources – AlienVault OTX
In the screenshot to the left, 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 for you!

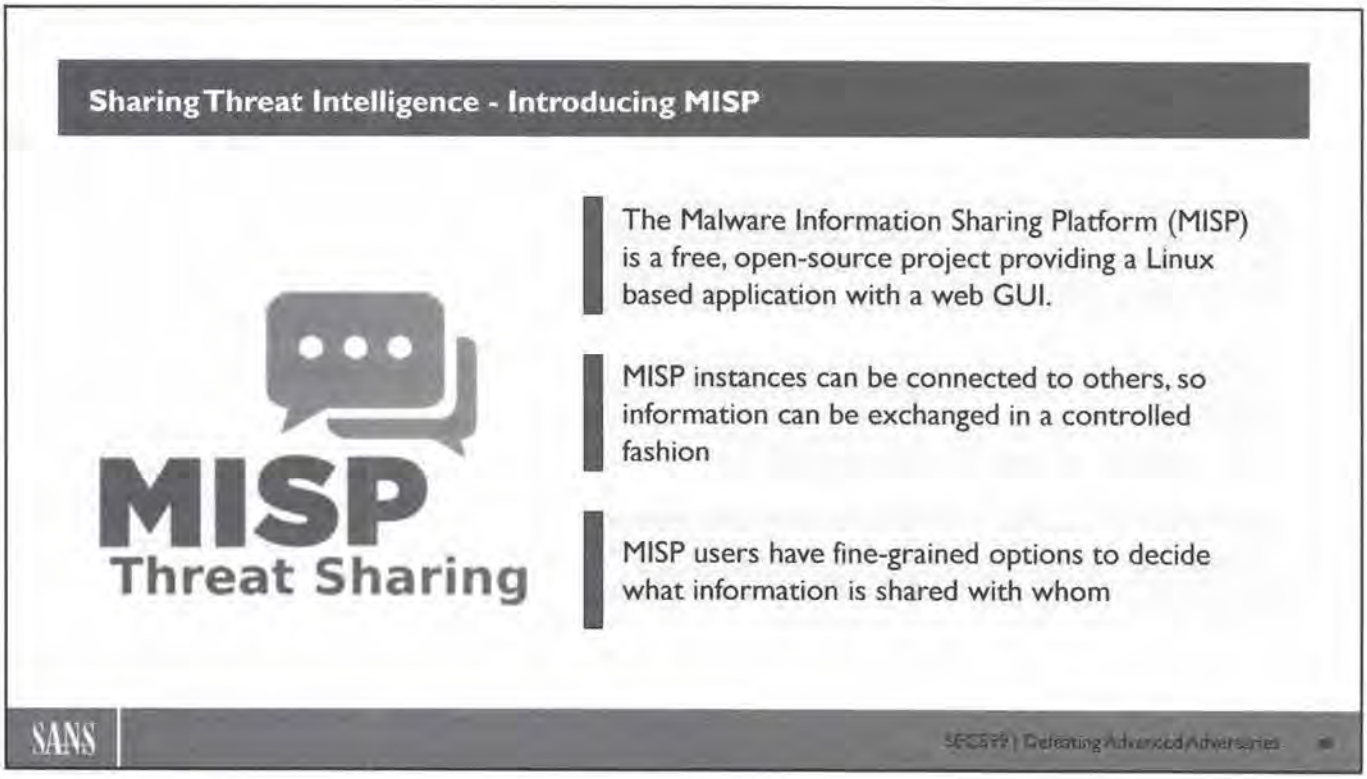
You can find AlienVault OTX at https://otx.alienvault.com



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, a web site, 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 at www.threatcrowd.org!



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.

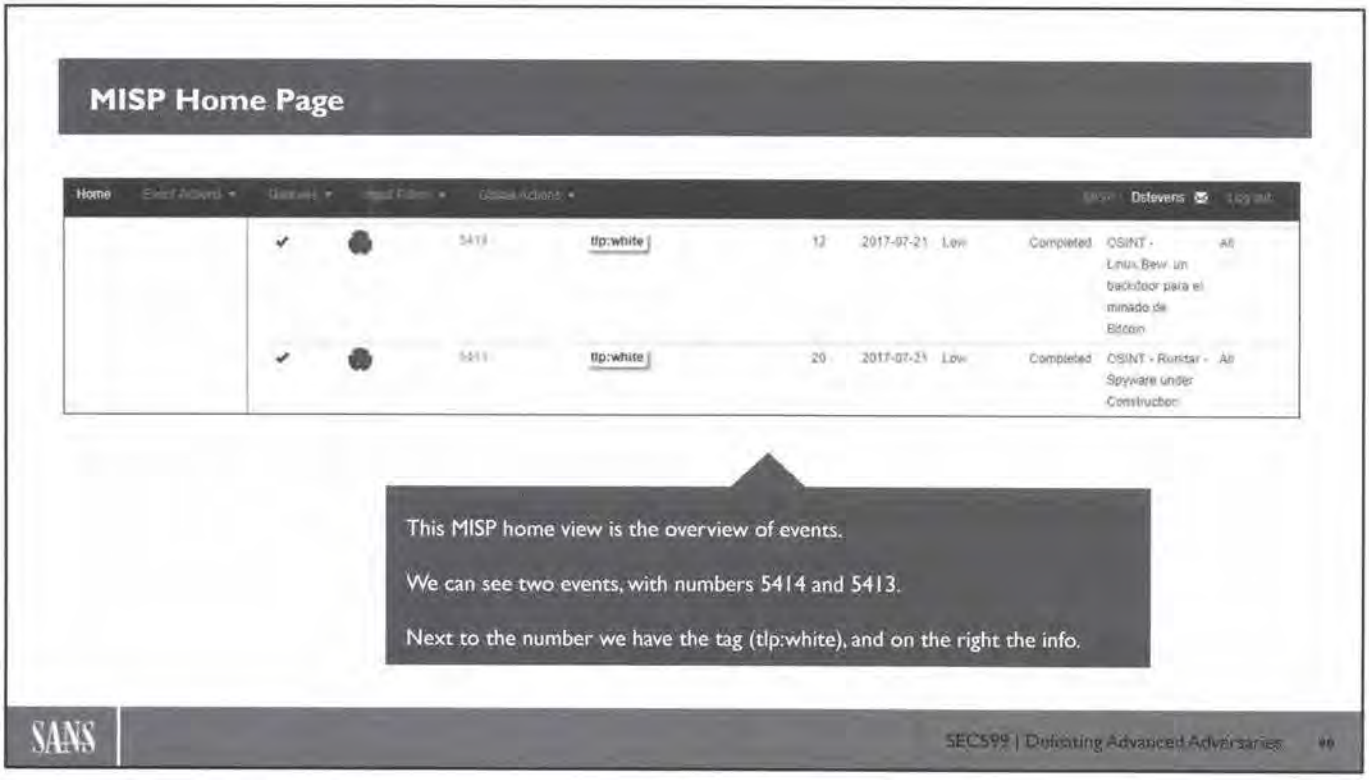
The event has the tags "phishing" and "Trojan".

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



MISP Home Page
The screenshot above is the home view of the MISP GUI interface, that 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.

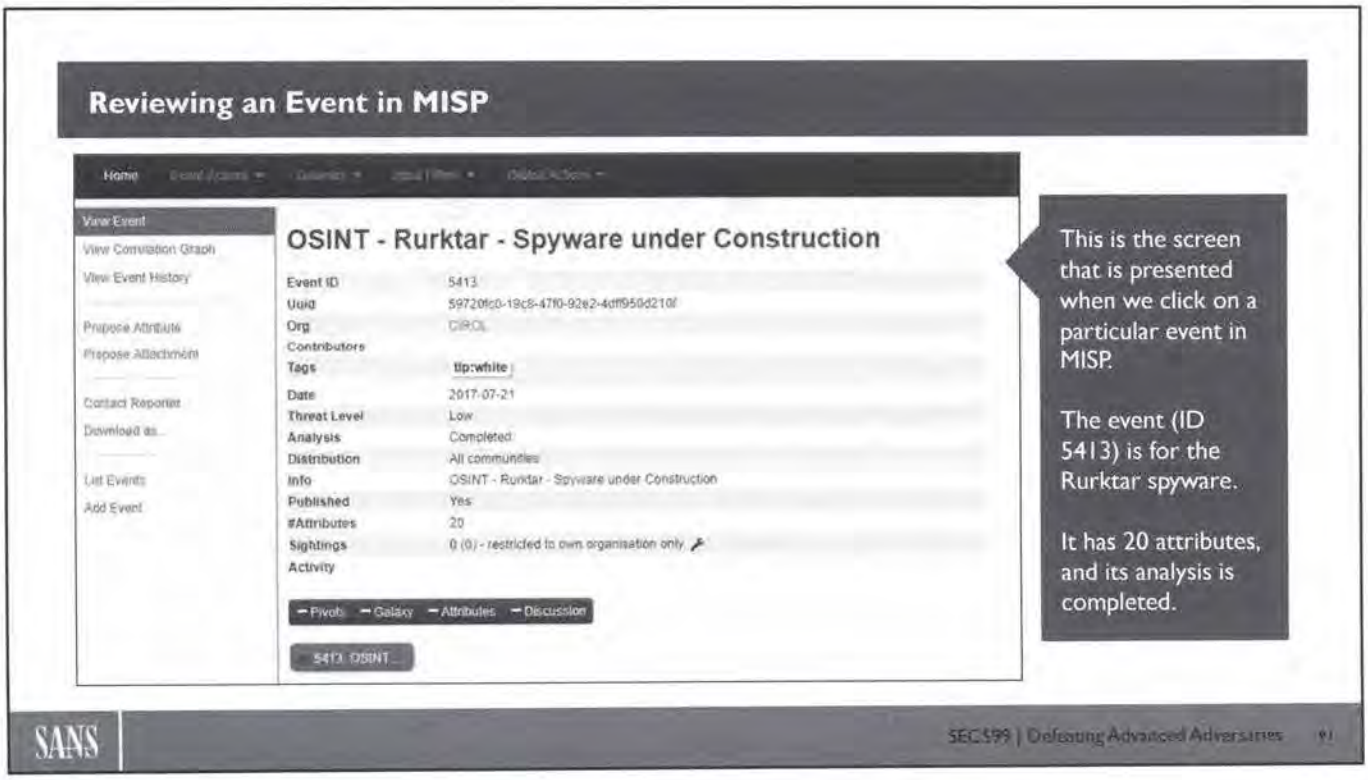
Then we have the number of the event and the tag (tlp:white).

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 web site 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.

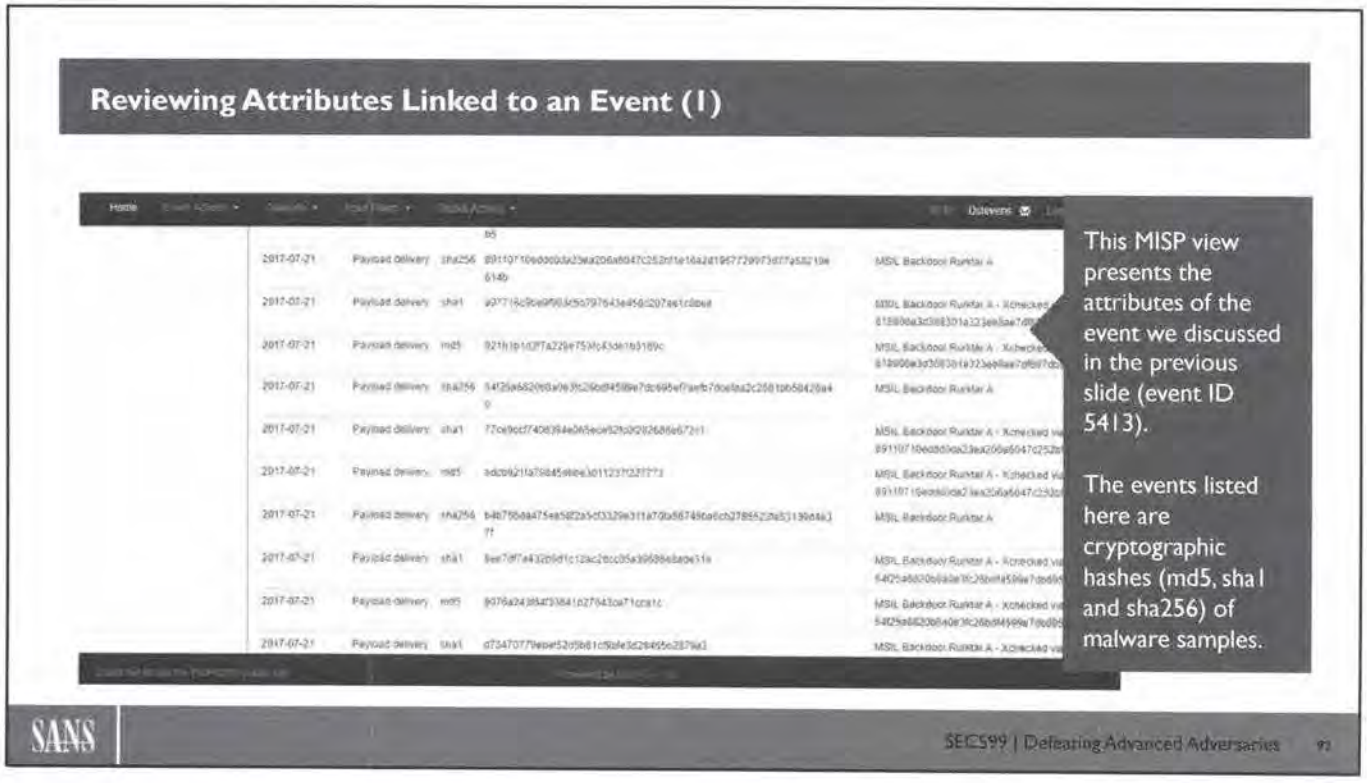
Tag tlp:white indicates that the information may be sharing under the Traffic Light Protocol (TLP) white status.

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.



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.

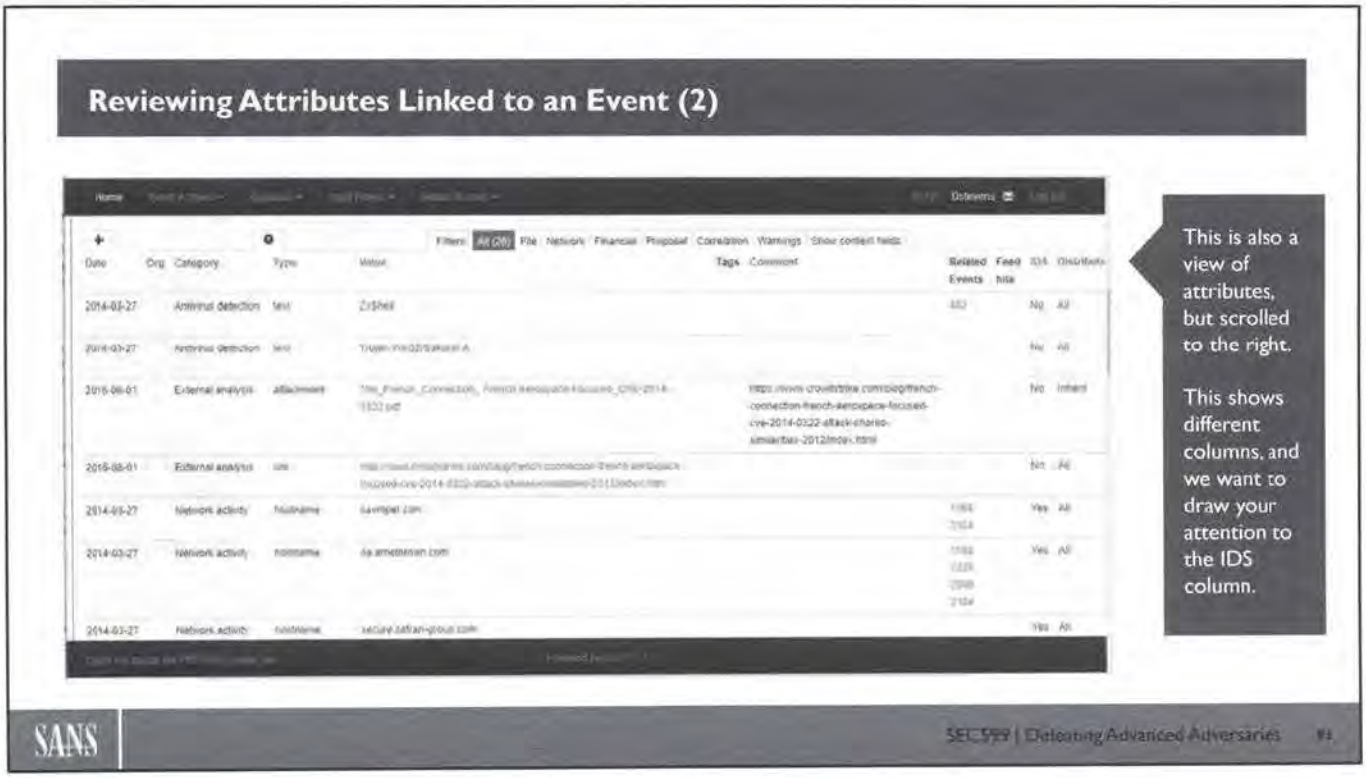
Then we have the type of attribute: in this example, the type is "payload delivery". This is used for malware samples that deliver the payload.

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.



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 request 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.

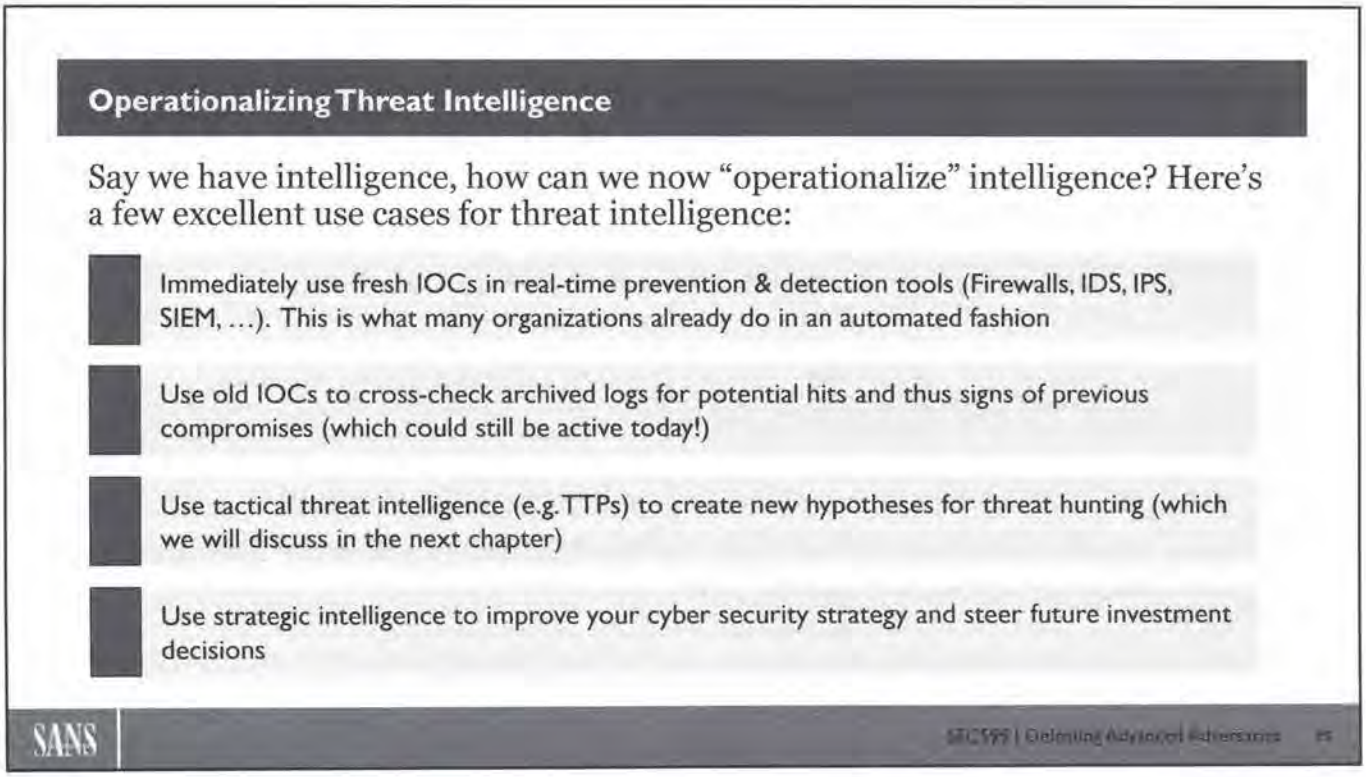
The date field is populated automatically.

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 yet. 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.



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 & detection tools (Firewalls, IDS, IPS, SIEM, …). This is what many organizations already do in an automated fashion. Many firewalls & 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 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 TTP's being used by adversaries and use them to define new potential intrusion methods we can check in our environment;
- Finally, we can use strategic intelligence to improve your cyber security strategy and steer future investment decisions.

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).

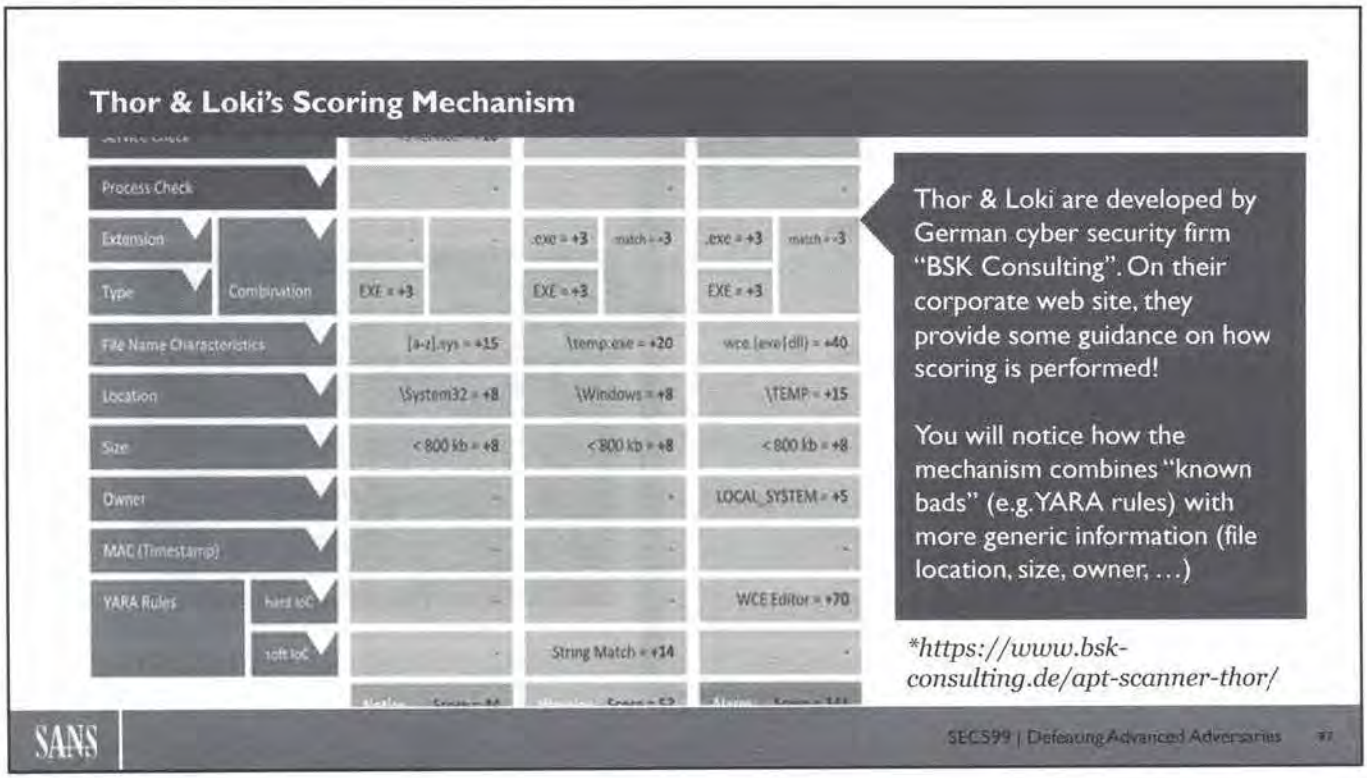
- An IOC scanner is a tool that will use IOCs to scan computer systems' resources (file system, registry, memory, ...)
- There are several open-source and commercial IOC scanners on the market, some are even based on scripting languages such as PowerShell!
- Loki was developed by Florian Roth (BSK Consulting) and is the free variant of BSK's commercial tool called "Thor"
- Both Loki & 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).

An IOC scanner is a tool that will use IOCs to scan computer systems' resources (file system, 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) and is the free variant of BSK's commercial tool called "Thor". Florian Roth is a known security researcher that has contributed heavily to the community by developing a large number of YARA rules.

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

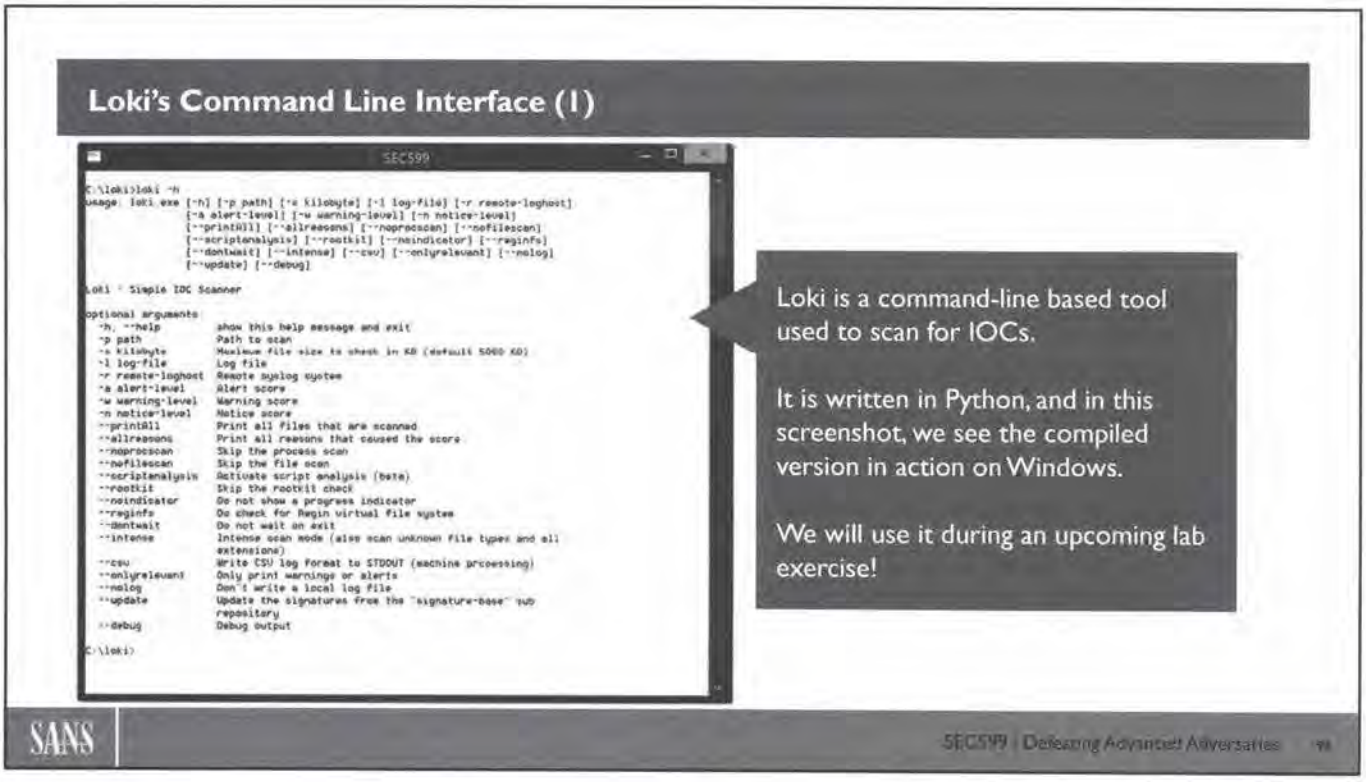


Thor & Loki's Scoring Mechanism

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 & Loki are developed by German cyber security firm "BSK Consulting". On their corporate web site, 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 & Thor can do more than just "hard IOC matching".



Loki's Command Line Interface
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.

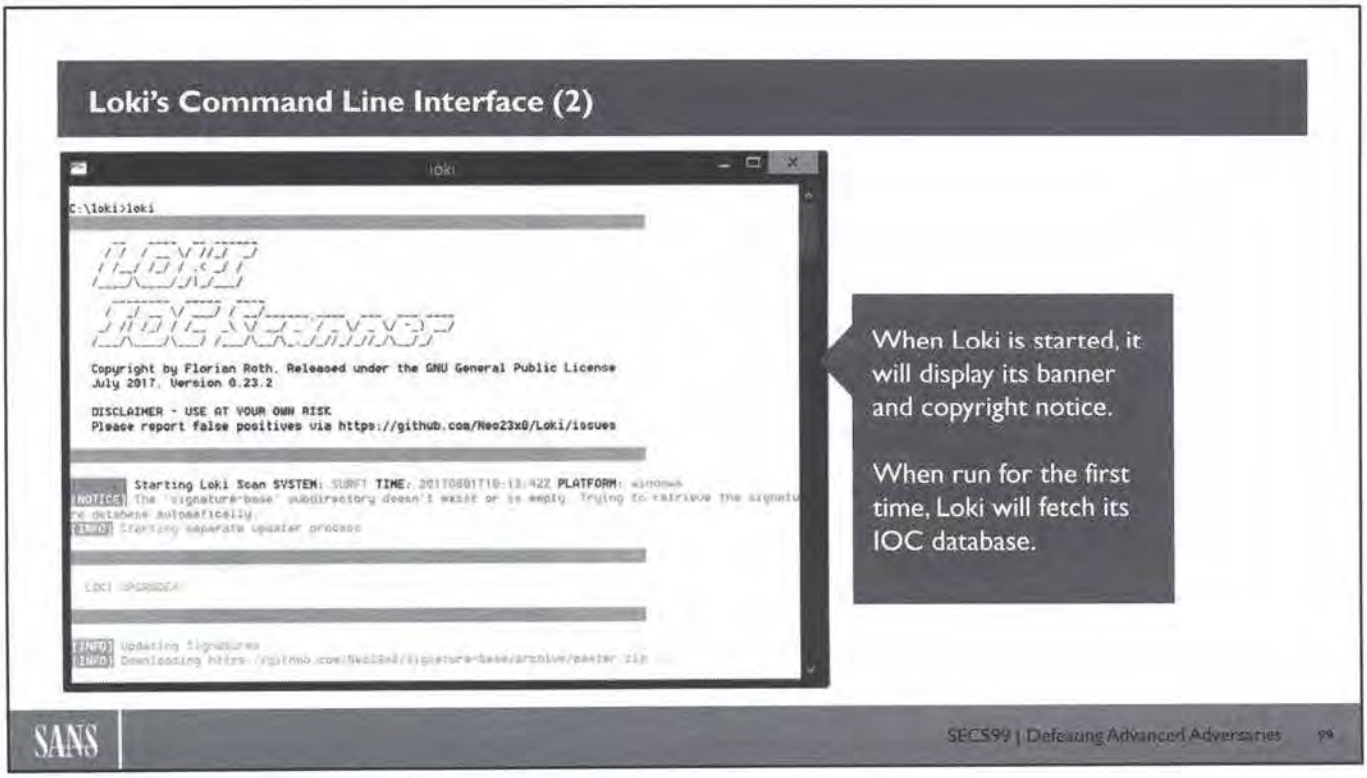
The Python version of Loki itself can also be used to perform IOC scans, but this implies that Python and all the modules Loki depends on (like YARA) must be installed and deployed on the system we want to scan.

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.

Therefor 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.



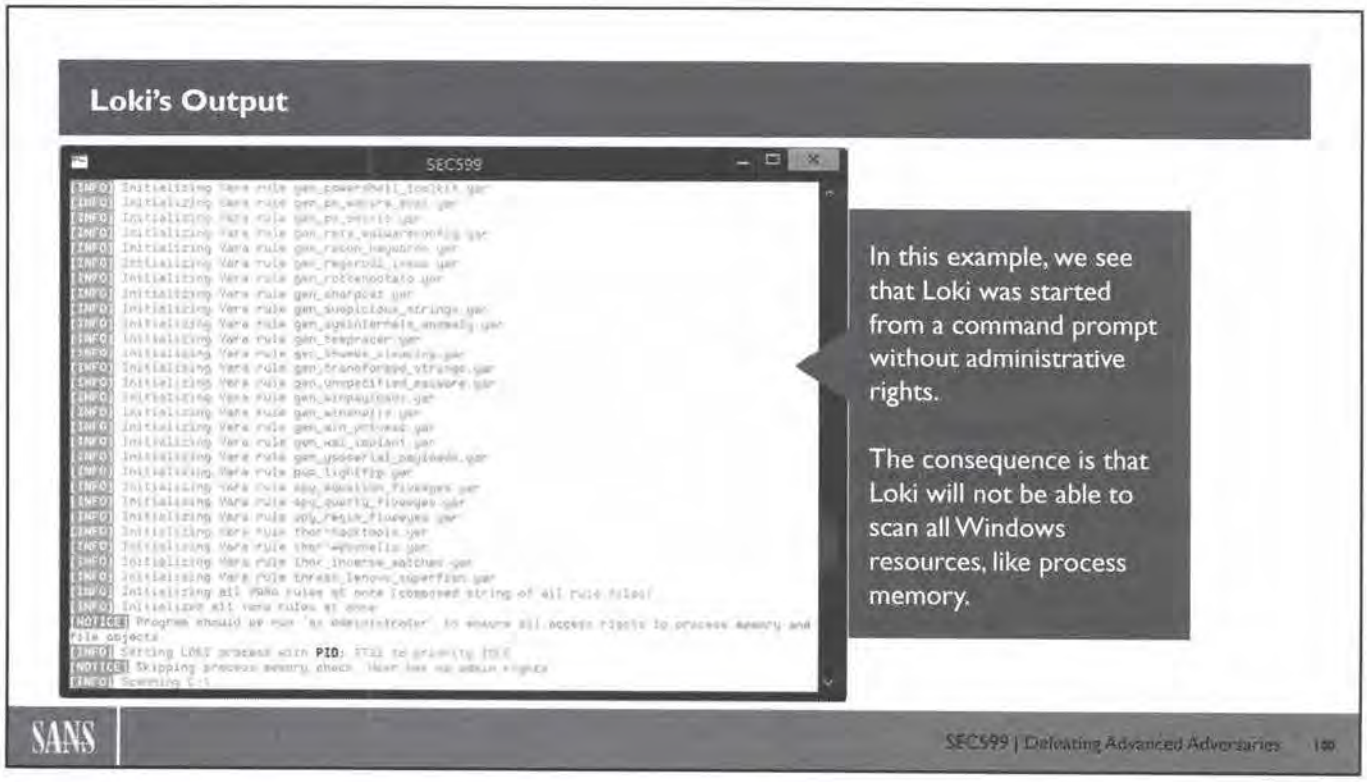
Loki's Command Line Interface (2)
When Loki is started, it will display its banner and copyright notice.

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.



Loki's Output
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.

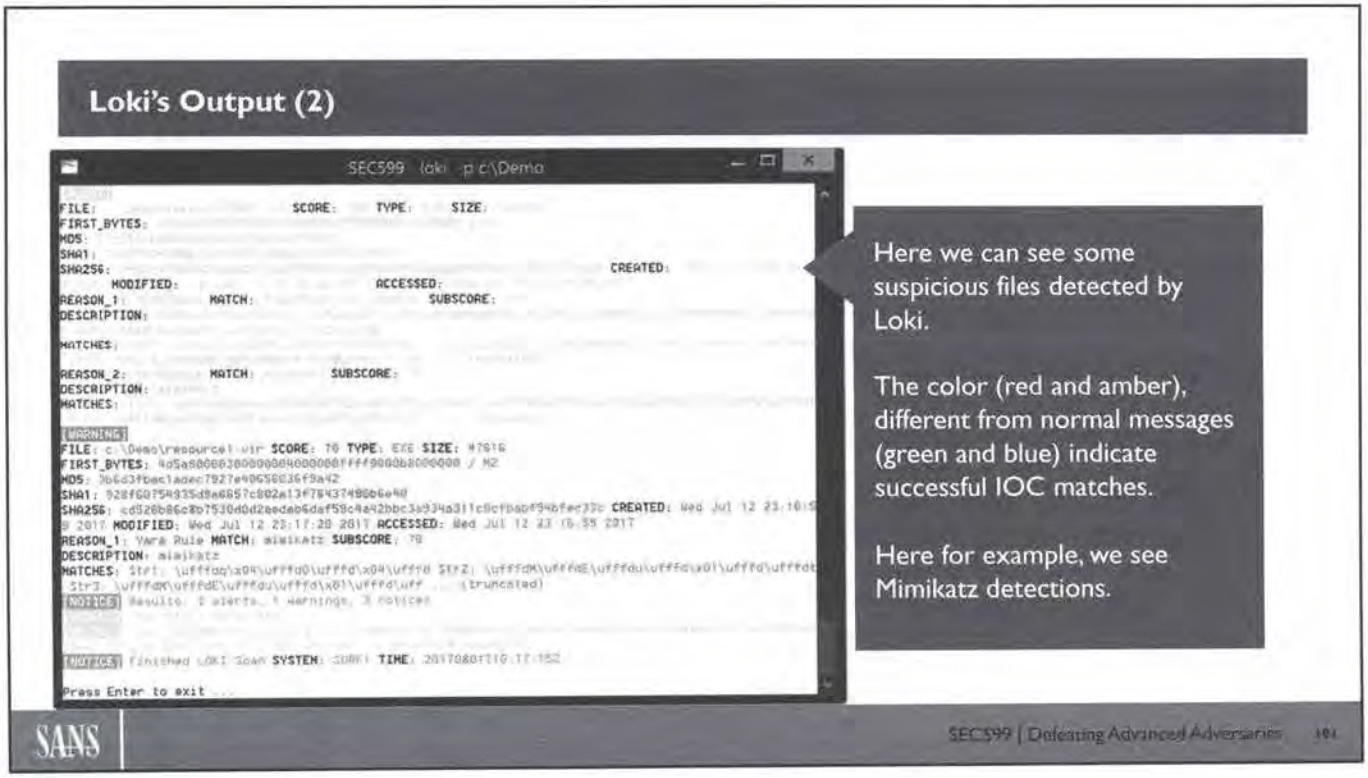
From the name of the YARA rules, we can deduce that Loki looks for hacker tools and exploitation tools used by advanced adversaries, and not for common malware.

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.

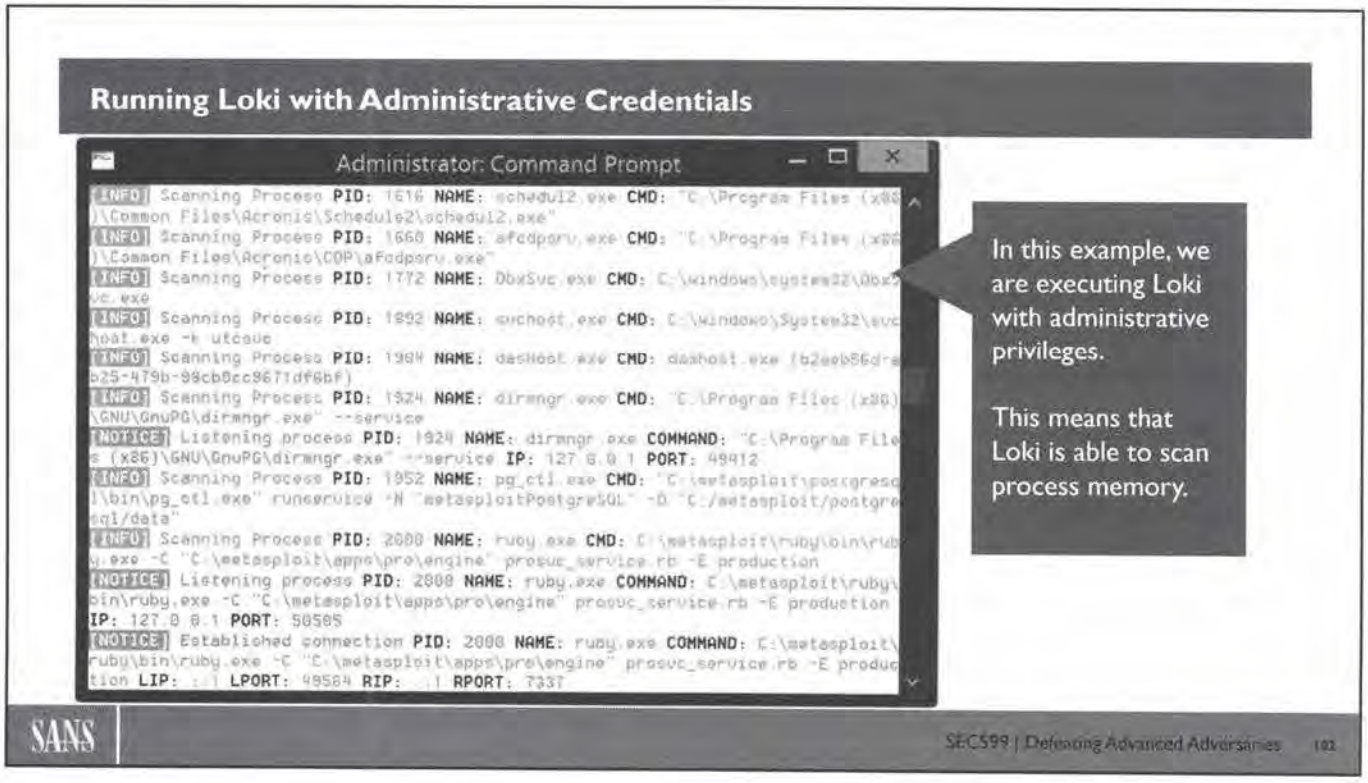
Here we can see some suspicious files detected by Loki.

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

The color (red and amber), 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.



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

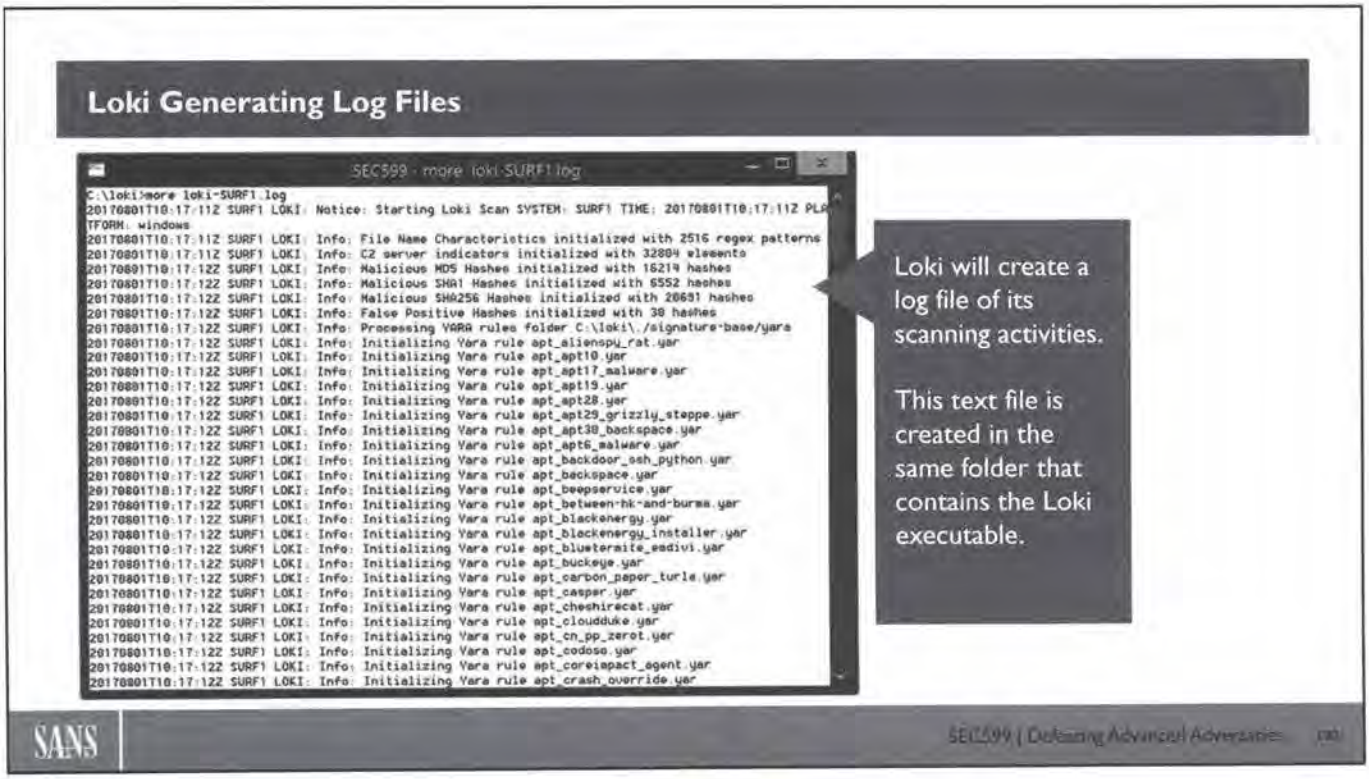
This means that Loki is able to scan process memory.

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 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.

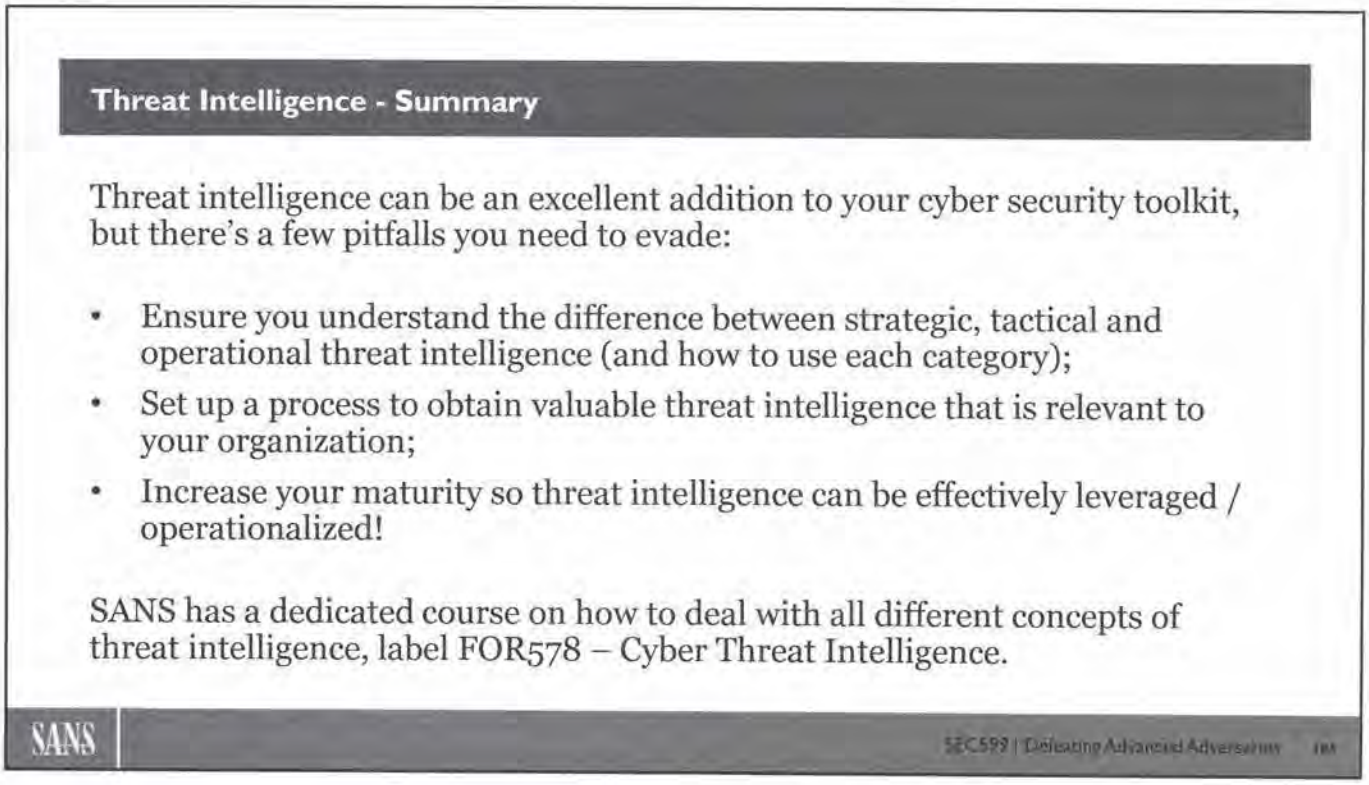
From the start of the example above, we can see that Loki looks for a large amount of the following IOC types:

- 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.

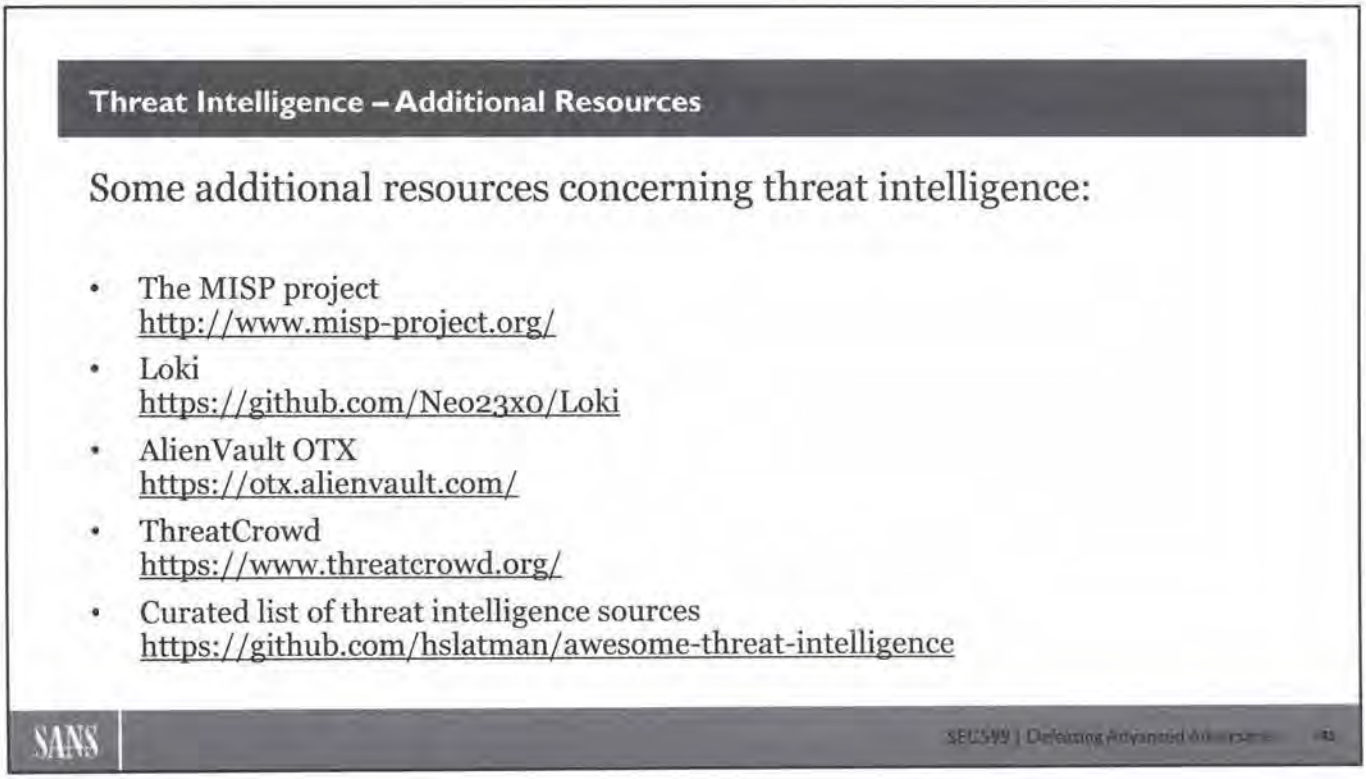
Loki can also produce a format better suitable to automatic processing with the CSV option.



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:

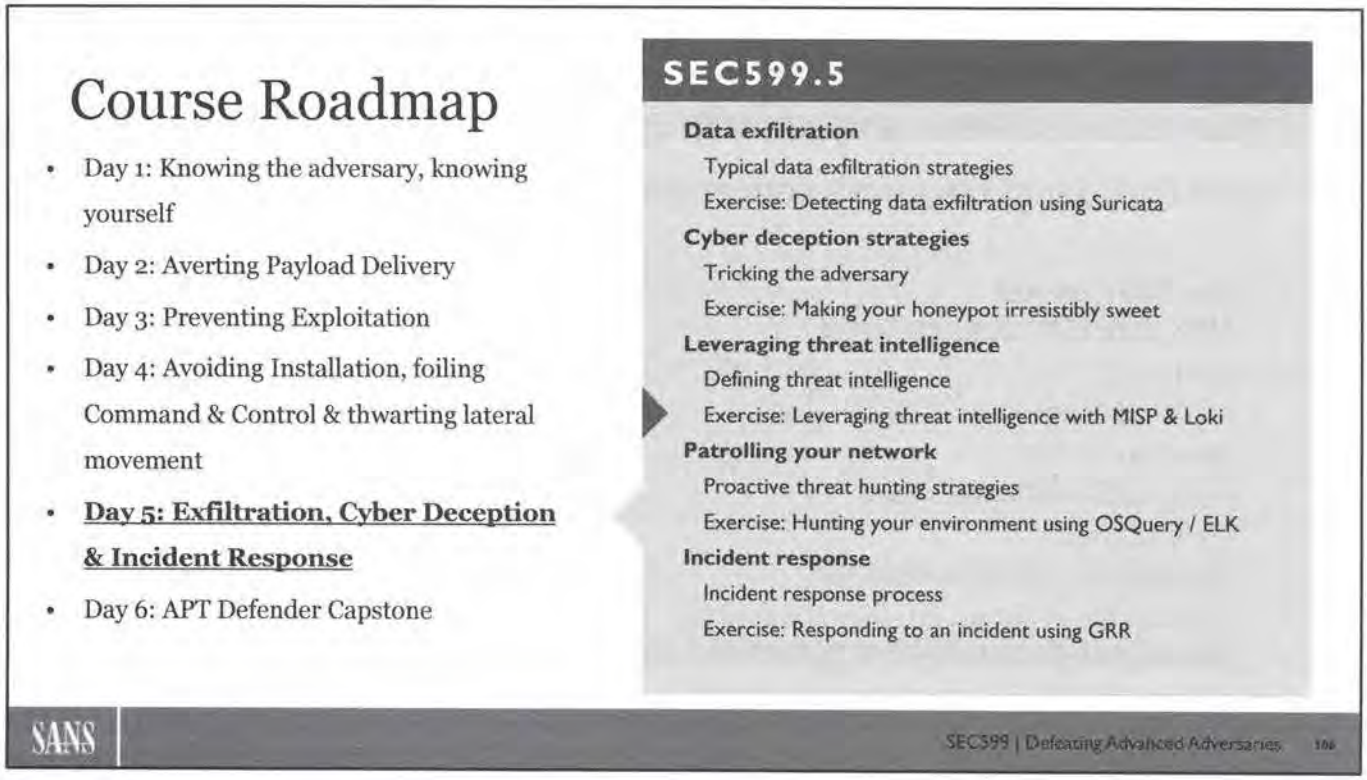
- Ensure you understand the difference between strategic, tactical and operational threat intelligence (and how to use each category);
- Set up a process to obtain valuable threat intelligence that is relevant to your organization;
- Increase your maturity so threat intelligence can be effectively leveraged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

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

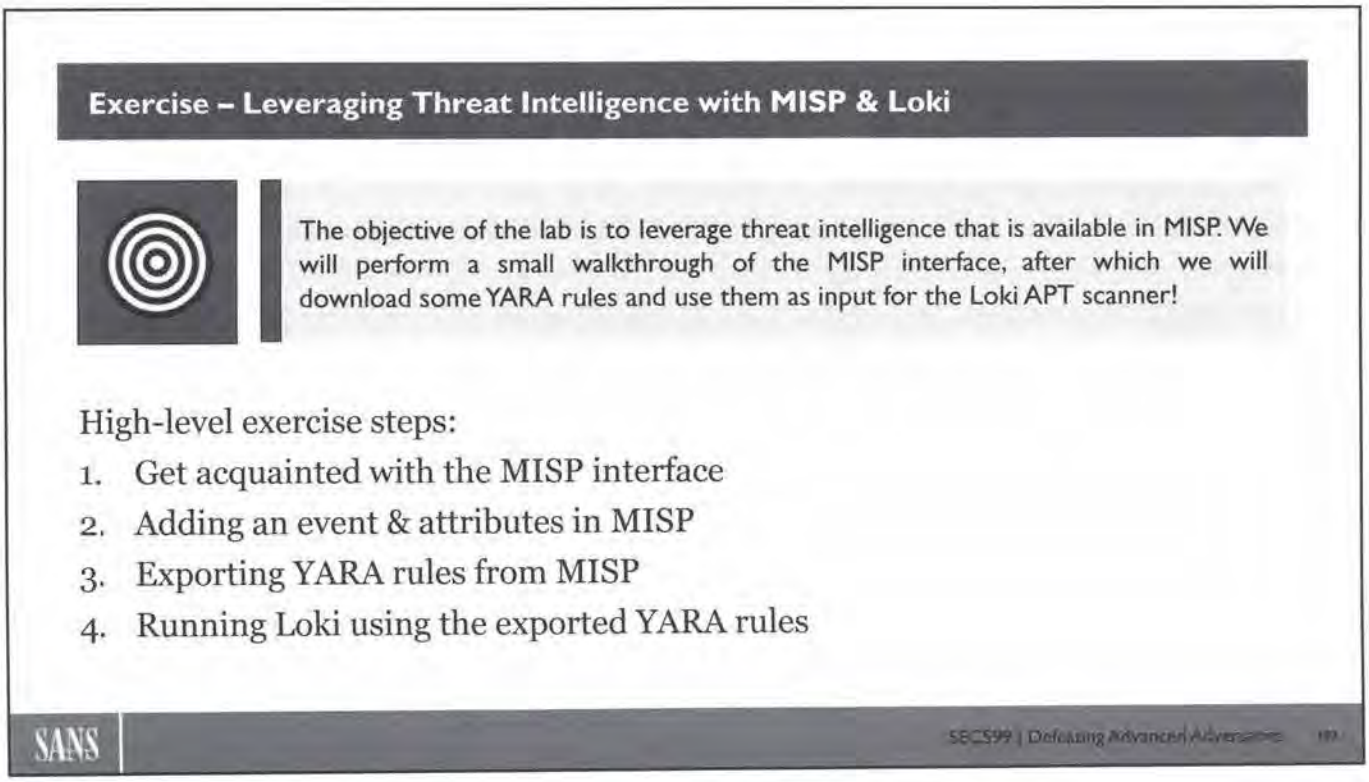


Defining Threat Intelligence
Some additional resources concerning defining threat intelligence:

- The MISP project
  http://www.misp-project.org/
- Loki
  https://github.com/Neo23x0/Loki
- Alienvault OTX
  https://otx.alienvault.com/
- Threatcrowd
  https://www.threatcrowd.org/
- Curated list of threat intelligence sources
  https://github.com/hslatman/awesome-threat-intelligence



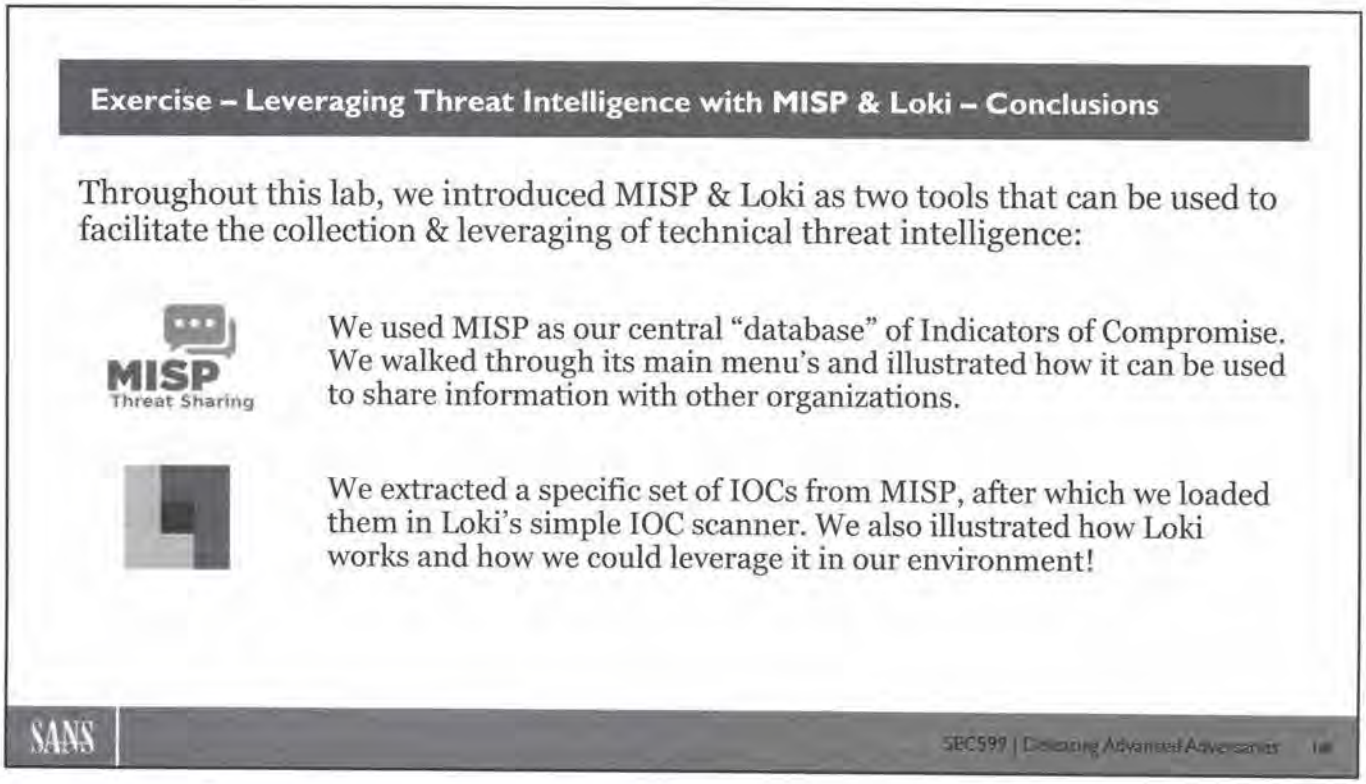
This page intentionally left blank.



Exercise – Leveraging Threat Intelligence with MISP & Loki
The objective of the lab is to leverage threat intelligence that is available in MISP. We will perform a small walkthrough of the MSIP interface, after which we will download some YARA rules and use them as input for the Loki APT scanner!

High-level exercise steps:
1. Get acquainted with the MISP interface
2. Adding an event & attributes in MISP
3. Exporting YARA rules from MISP
4. Running Loki using the exported YARA rules

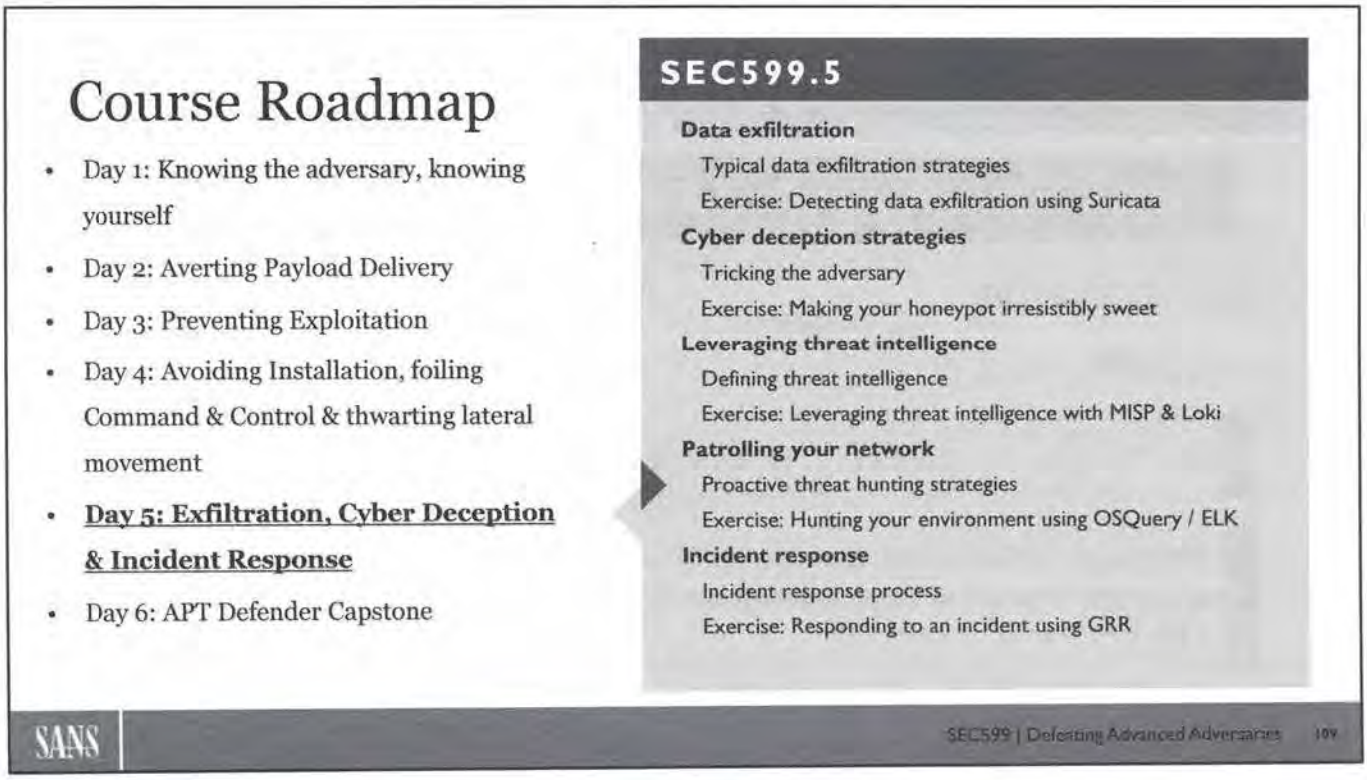
For additional guidance & details on the lab, please refer to the LODS workbook.



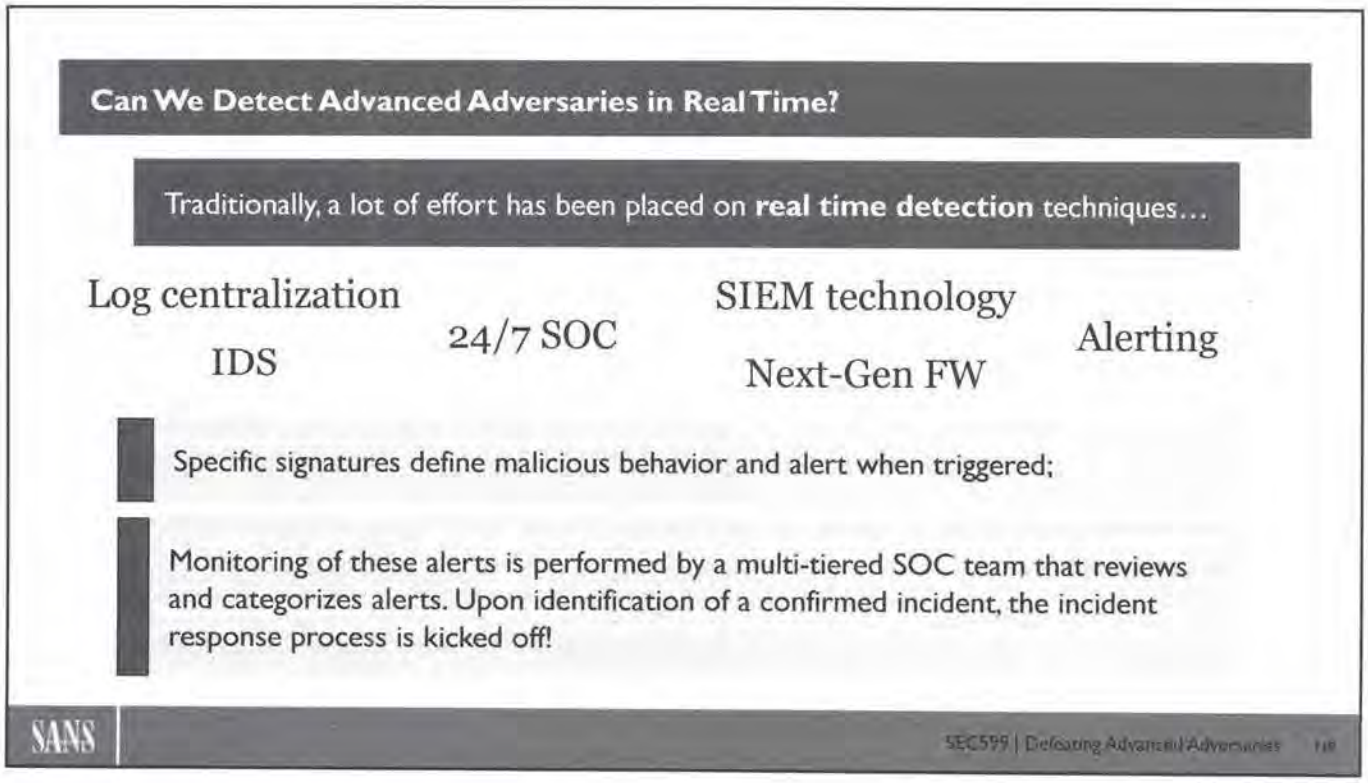
Exercise – Leveraging Threat Intelligence with MISP & Loki – Conclusions
Throughout this lab, we introduced MISP & Loki as two tools that can be used to facilitate the collection & leveraging of technical threat intelligence:

We used MISP as our central "database" of Indicators of Compromise. We walked through its main menu's and illustrated how it can be used to share information with other organizations.

We extracted a specific set of IOCs from MISP, after which we loaded them in Loki's simple IOC scanner. We also illustrated how Loki works and how we could leverage it in our environment!



This page intentionally left blank.



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 on 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 address is reused for other purposes.

This sharing and reuse lead to false positive alerts.

Real-Time Detection Issues: Security Alert Fatigue

"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.

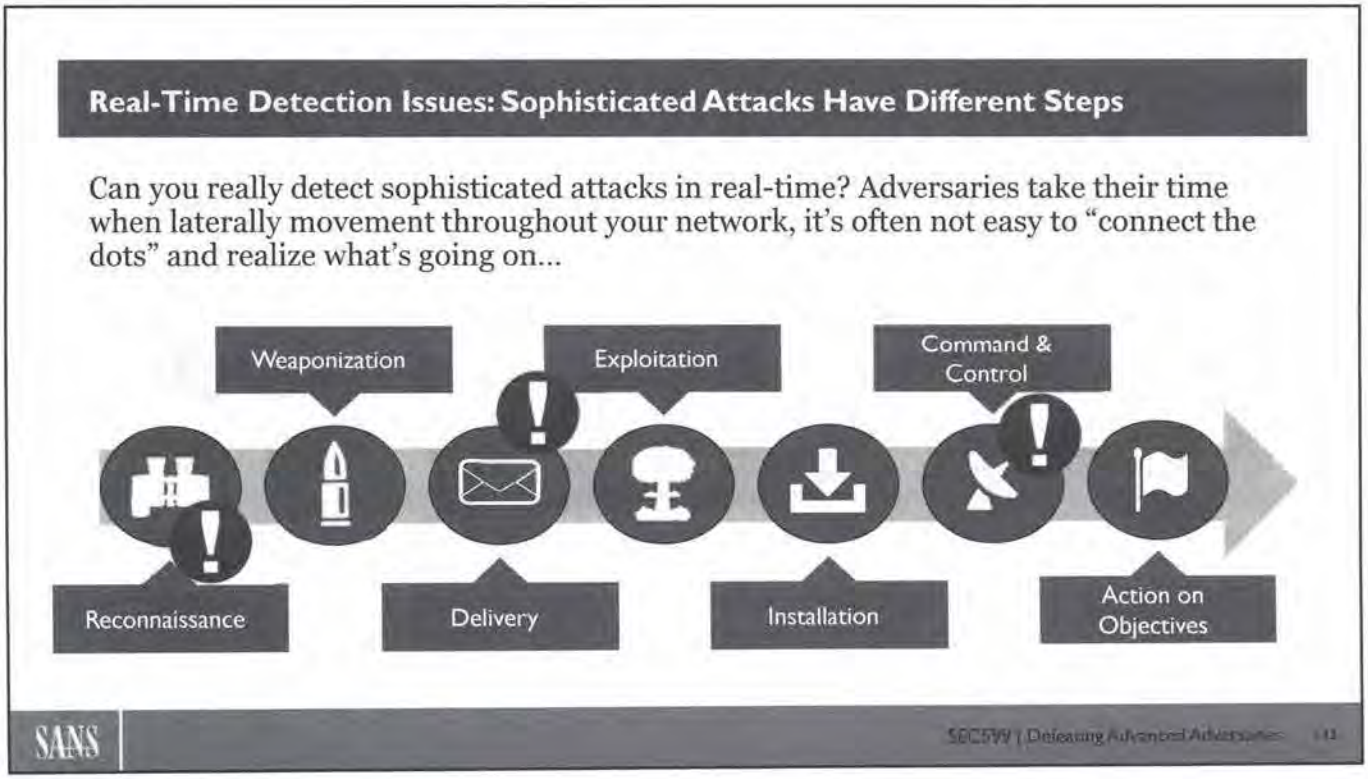
Real-time detection methods generate too many false positive alerts, with a negative impact on the motivation and morale of security teams that have to investigate the alerts.

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 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…

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 setups 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.

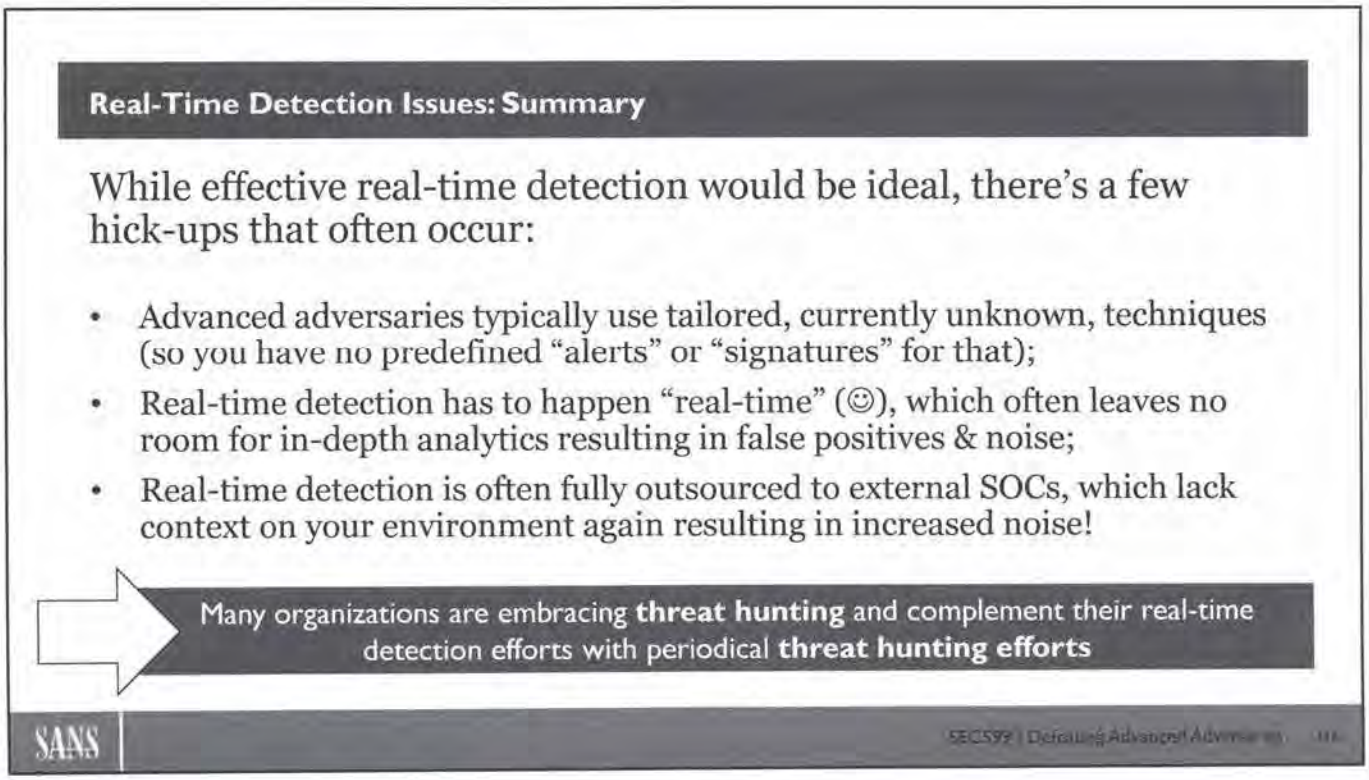
Real-Time Detection Issues: Some Statistics

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

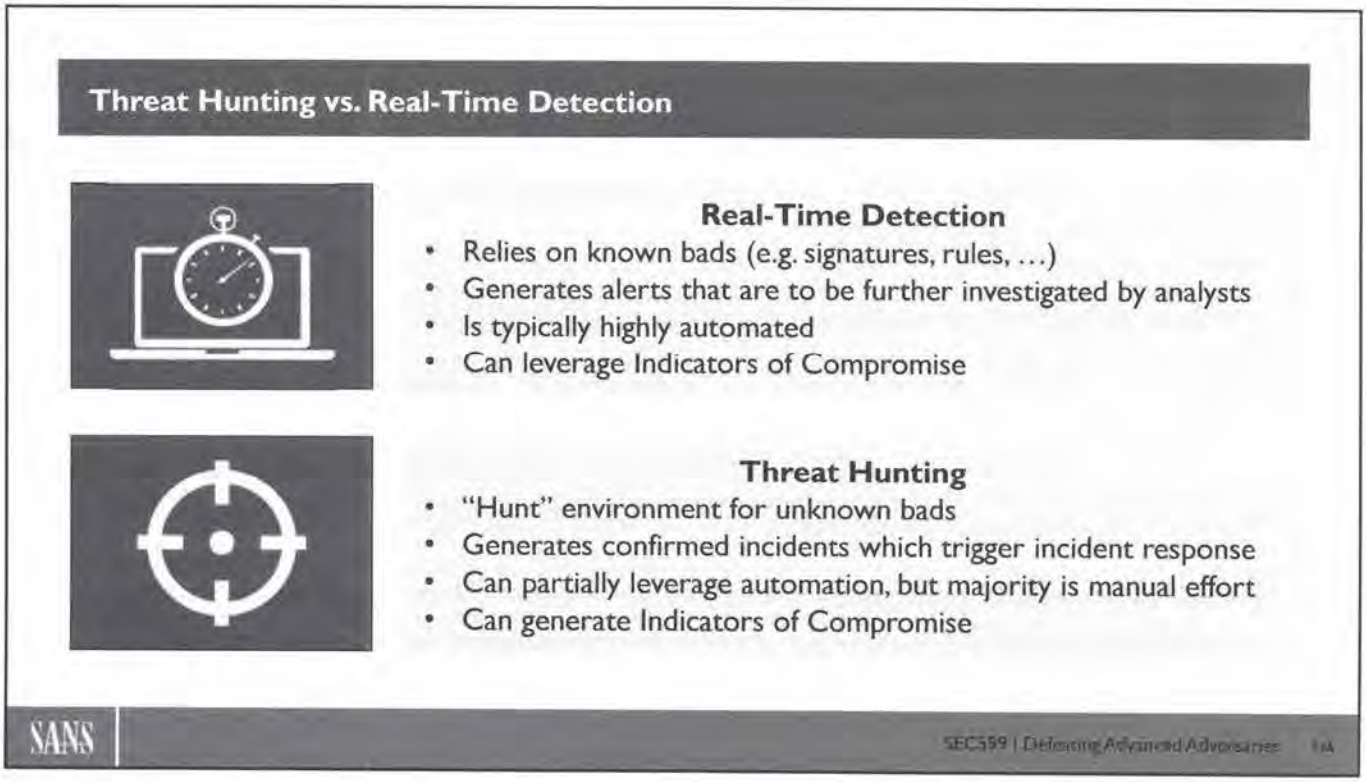


Real-Time Detection Issues: Summary

While effective real-time detection would be ideal, there are a few hick-ups 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 "real-time" (☺), which often leaves no room for in-depth analytics resulting in false positives & 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, we go out and search for suspicious behavior ourselves!



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 of 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 web site 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 a 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).

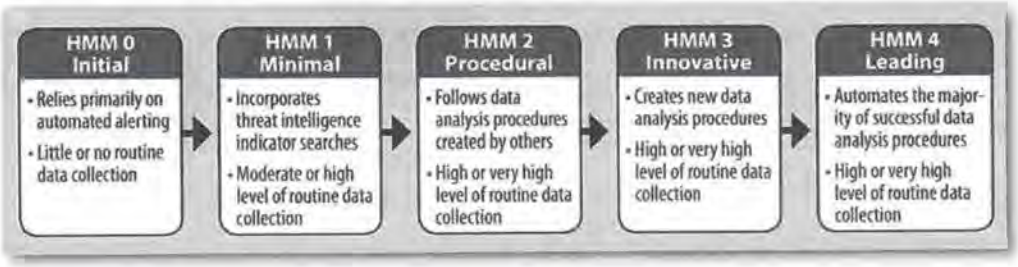
While real-time detection is highly automated, threat hunting is a manual effort that can be machine assisted.

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

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 5-level "maturity model" for threat hunting:



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 5-level "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.

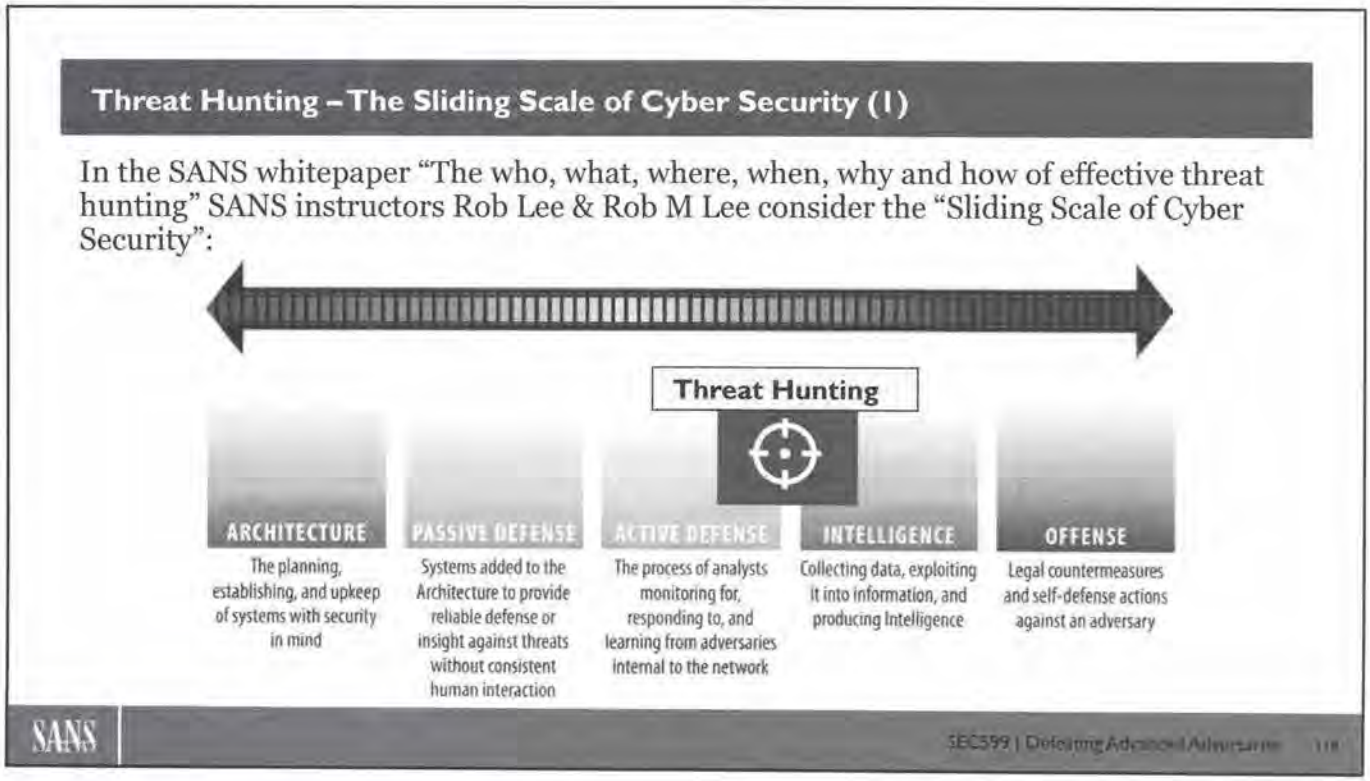
Level HMM1 is the minimal level.
Organizations with this maturity level incorporate threat intelligence indicator searches and have a moderate or high level of routine data collection.

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.



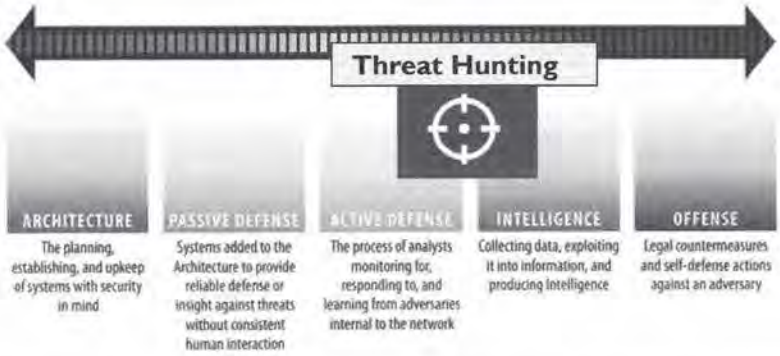
Threat Hunting – The Sliding Scale of Cyber Security

In the SANS whitepaper "The who, what, where, when, why and how of effective threat hunting" SANS instructors Rob Lee & Rob 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 form adversaries internal to the network. **Intelligence**: This phase is the process of collecting data, turning it into information which can be used to generate intelligence / useful knowledge that can help improve an organization's security posture.
- **Offense**: The last category, "offense" 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".

Threat Hunting – The Sliding Scale of Cyber Security (2)

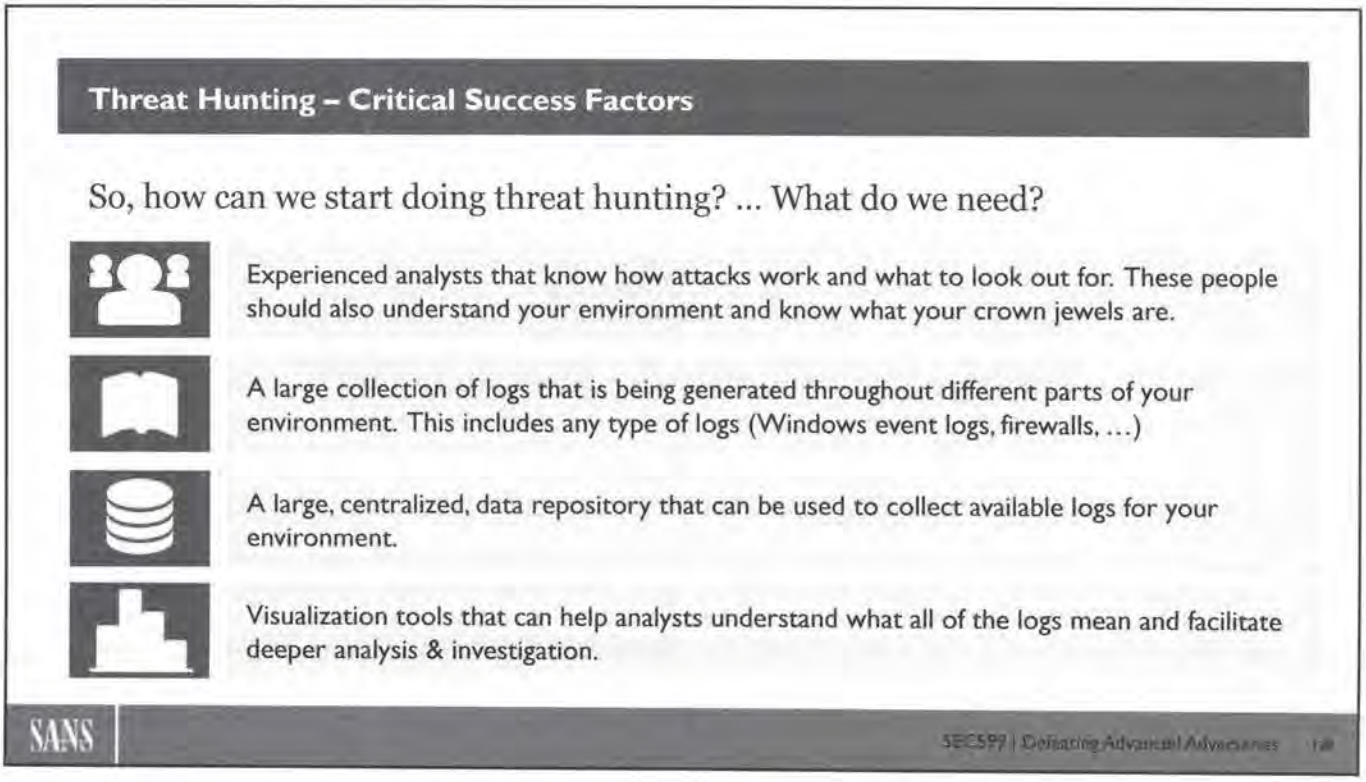


We can use the sliding scale to understand what the potential ROI of threat hunting in your organization can be!

If there are serious gaps in architecture & 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 & resources for an organization.

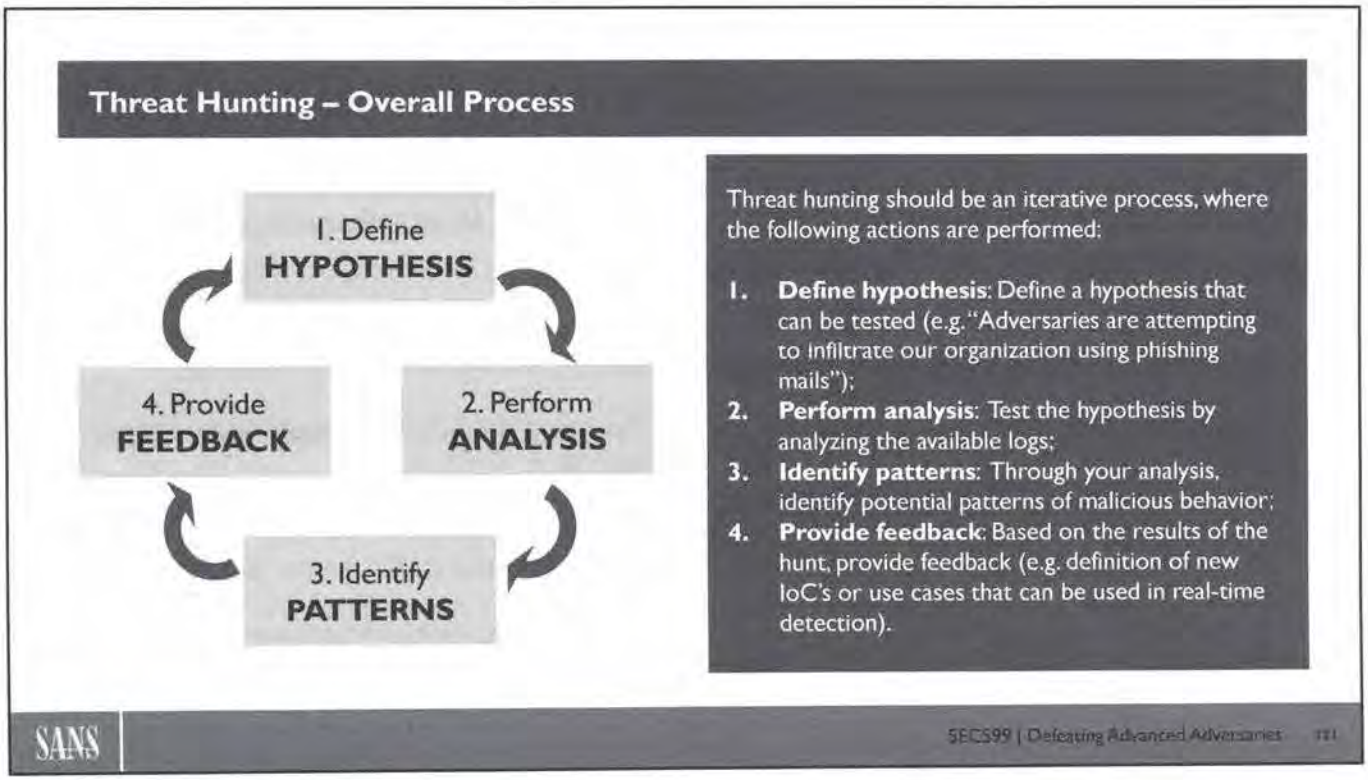
Imagine for example an organization that has significant gaps in architecture & passive defense: No proper vulnerability management process is in place and the majority of systems is unpatched. This is a rather extreme example, but it's important to note that in such an environment, threat hunting is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



Threat Hunting – Critical Success Factors
Threat hunting can detect malicious activity by reviewing available logs for anomalies.

Critical success factors for threat hunting are:

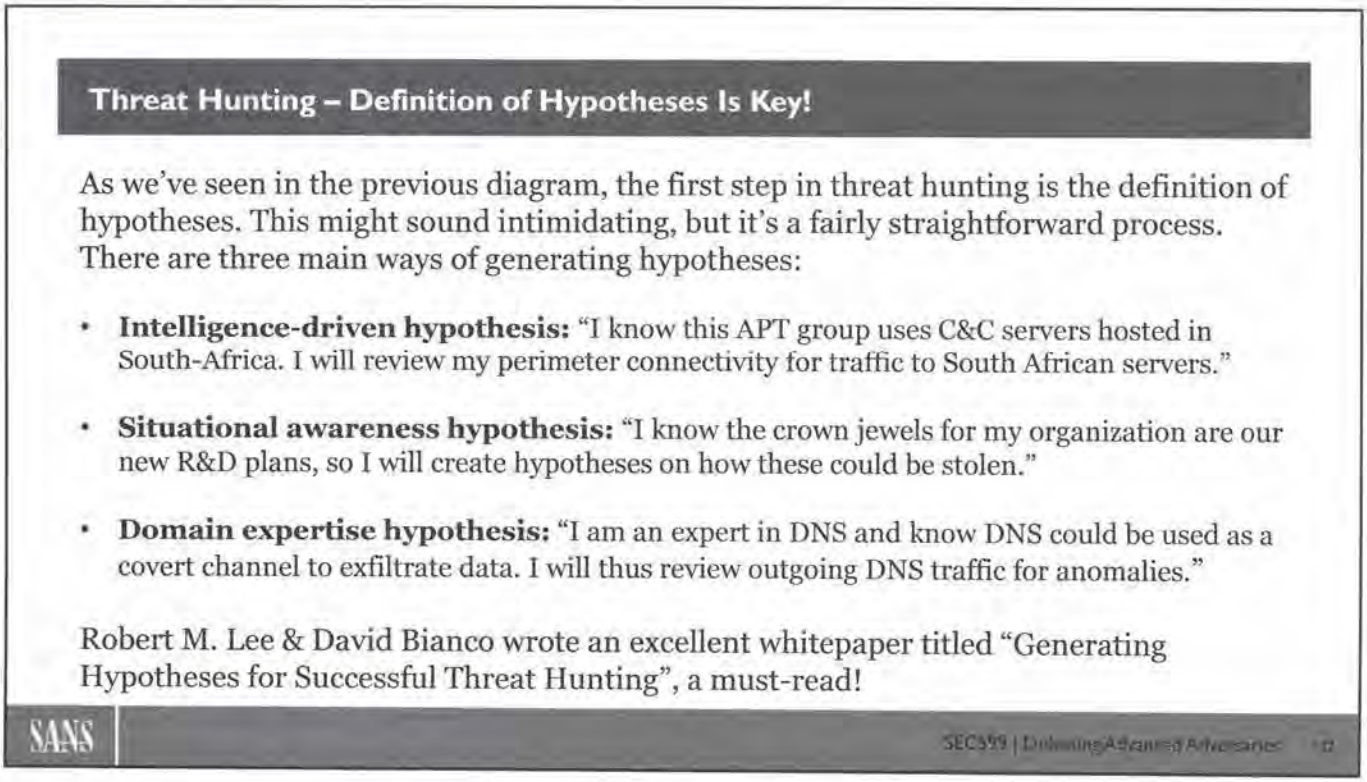
- Experienced analysts that know how attacks work and what to look out for. These people should also understand your environment and know what your crown jewels are. A good knowledge of offensive security methods is required to be a good threat hunter: a poacher makes the best gamekeeper. Meanwhile, 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 & 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 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 IOC's 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!

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!

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

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

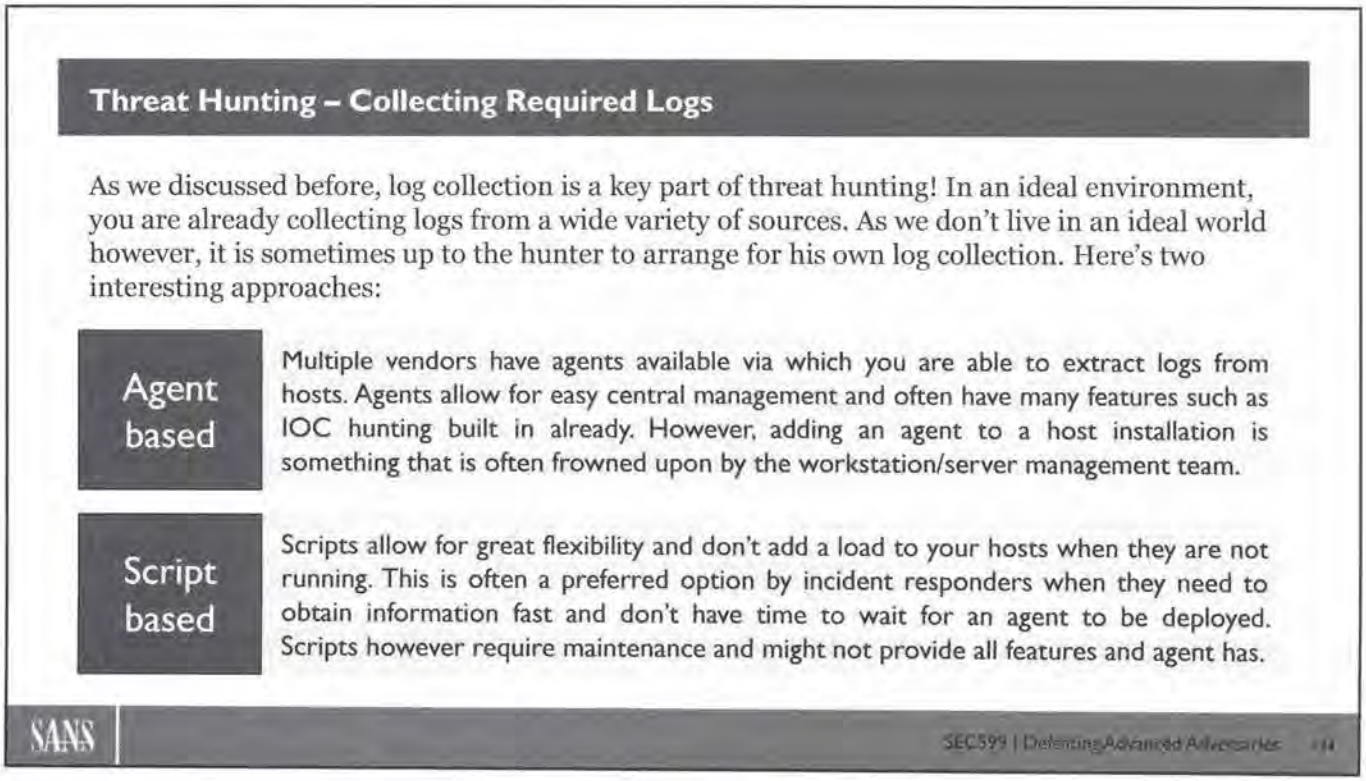
The crucial part is using the human effort where it is used best: not to crunch millions of alerts, but to create / define **a data analysis technique** and **reviewing its results!**

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 system 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.



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 by 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 approached and agent-based approaches.

Threat Hunting – Collecting Required Logs – Polling for Information

When discussing the collection of logs using scripts, one (of many) 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 host to collect information such as registry values and OS info;
- Surveys: deploys scripts to hosts to collect information from that host (digs deeper than scanners);
- Analysis: provides a framework for analyzing and displaying survey and scan results.

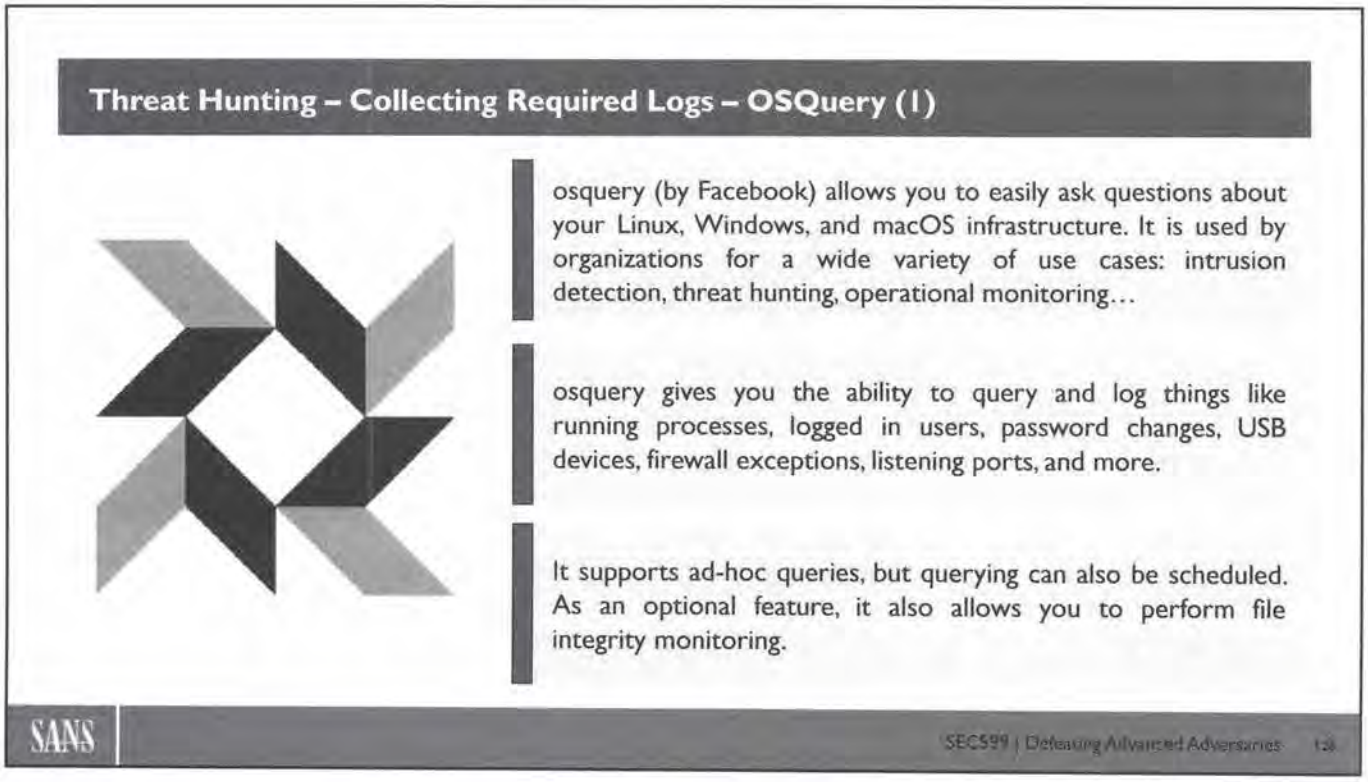
Threat Hunting – Collecting Required Logs – PSHunt
When talking about collecting logs using scripts, one readily available resource is PSHunt created and open sourced by Infocyte.

"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 host to collect information such as registry values and OS info;
Surveys: deploys scripts to hosts to collect information from that host (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 – Collecting Required Logs – OSQuery (1)
osquery (by Facebook) allows you to easily ask questions about your Linux, Windows, and macOS infrastructure. It is used by organizations for a wide variety of use cases: intrusion detection, threat hunting, operational monitoring, …

osquery gives you the ability to query and log things like running processes, logged in users, password changes, USB devices, firewall exceptions, listening ports, and more.

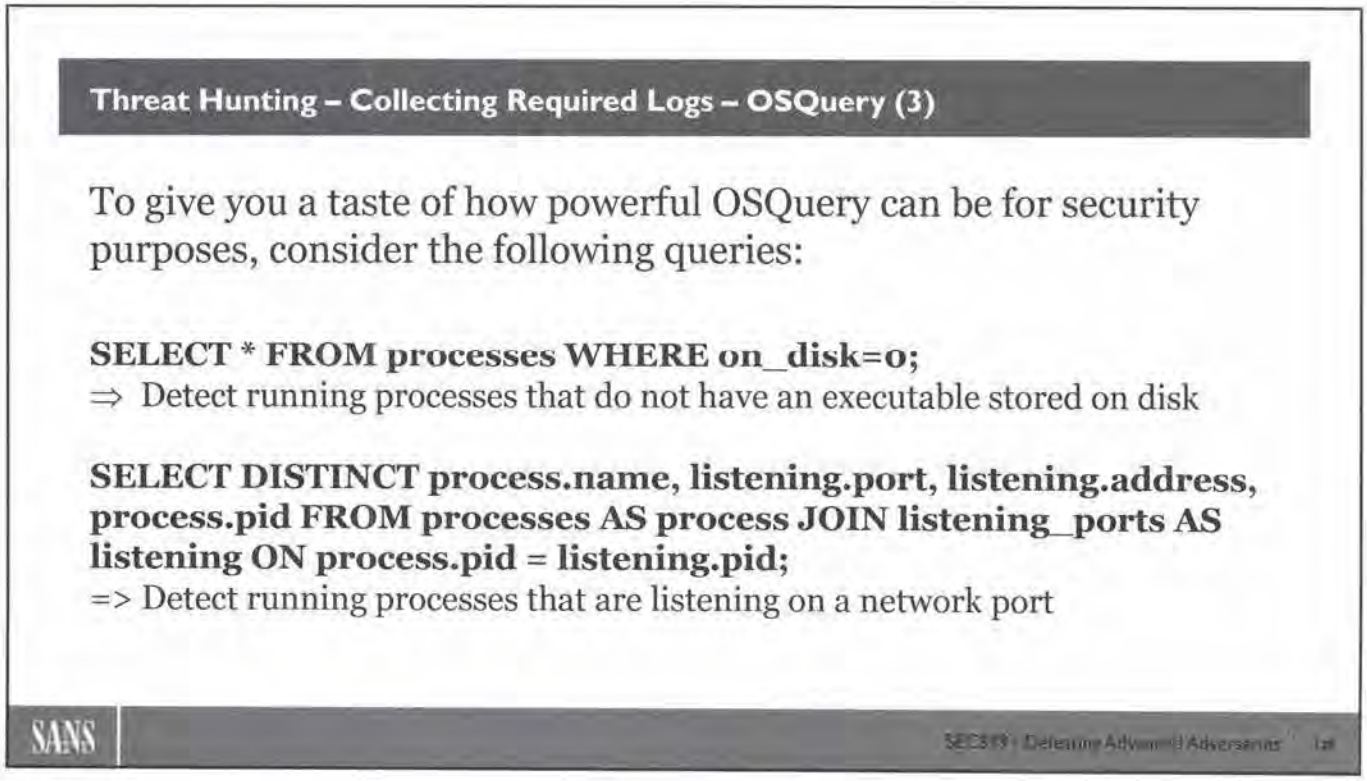
It supports ad-hoc queries, but querying can also be scheduled. As optional features, it also allows you for example to perform file integrity monitoring.

Threat Hunting – Collecting Required Logs – OSQuery (2)

```
osquery> SELECT uid, name FROM listening_ports l, processes p WHERE l.pid=p.pid;
```

OSquery provides a generic SQL query language that can be used by analysts to obtain system information from a wide variety of operating systems. This prevents the need for in-depth OS knowledge during data collection!

Threat Hunting – Collecting Required Logs – OSQuery (2)
In the above slide, we can see an example of how OSQuery facilitates our work: we can use a generic query language to query a wide variety of OS's. OSQuery provides a generic SQL query language that can be used by analysts to obtain system information from a wide variety of operating systems. This prevents the need for in-depth OS knowledge during data collection!

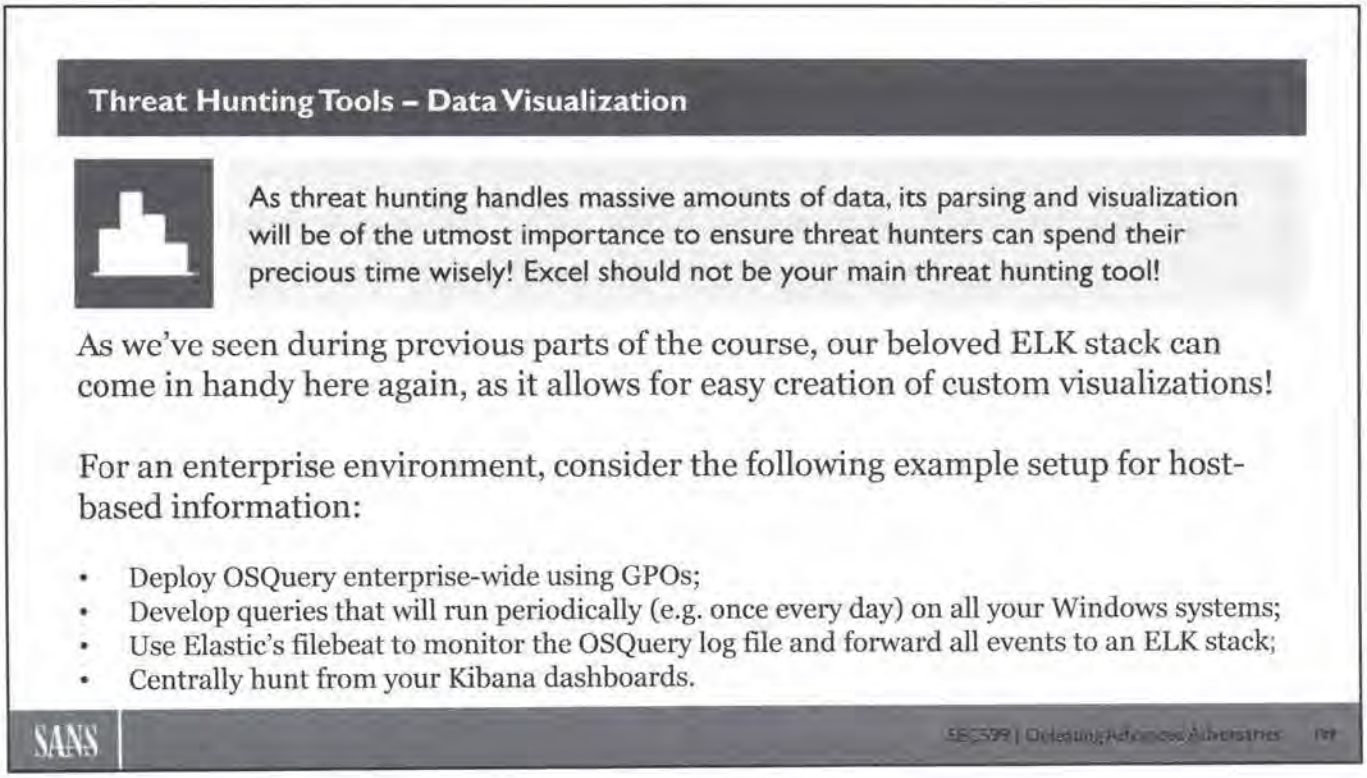


Threat Hunting – Collecting Required Logs – OSQuery (3)
To give you a taste of how powerful OSQuery can be for security purposes, consider the following queries:

SELECT * FROM processes WHERE on_disk=0;
=> Detect running processes that do not have an executable stored on disk;

SELECT DISTINCT process.name, listening.port, listening.address, process.pid FROM processes AS process JOIN listening_ports AS listening ON process.pid = listening.pid;
=> Detect running processes that are listening on a network port

We will touch upon additional examples during our upcoming lab.



Threat Hunting Tools – Data Visualization

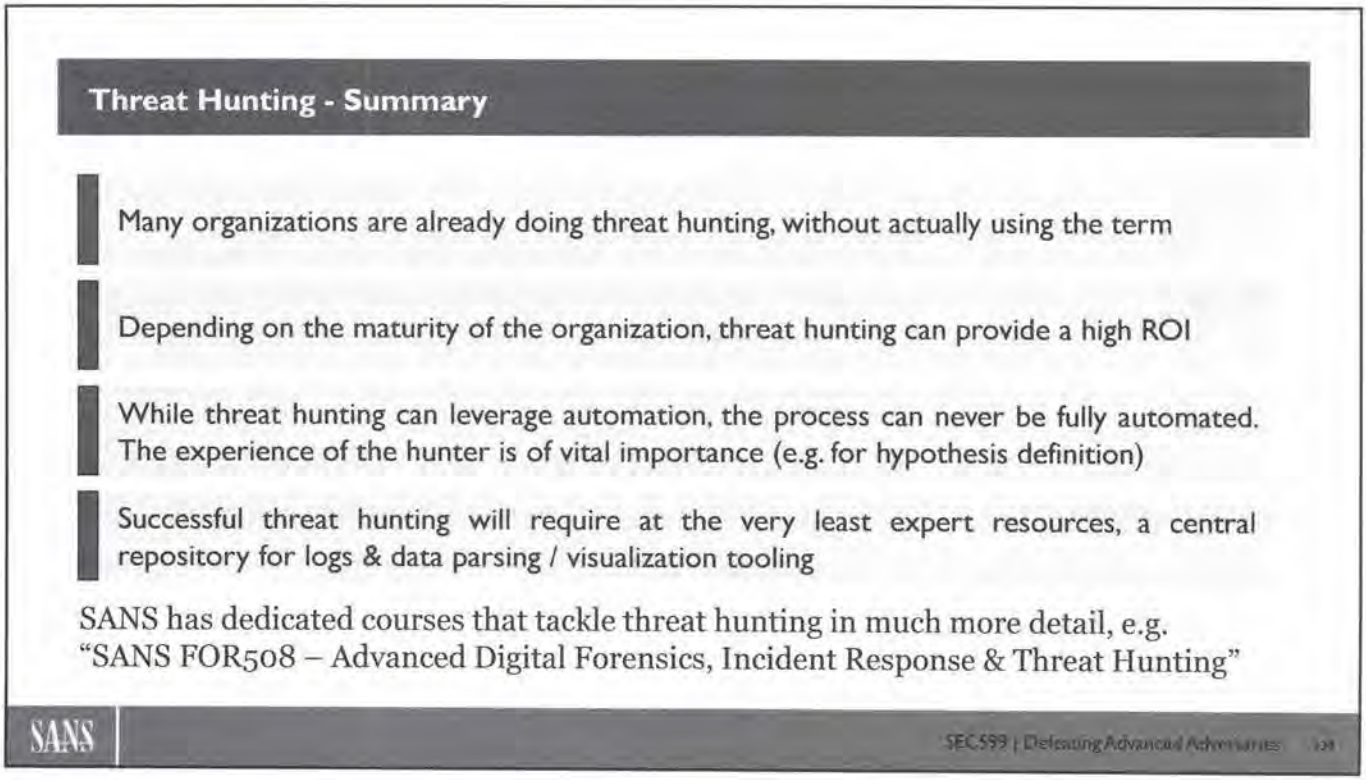
As threat hunting handles massive amounts of data, its parsing and visualization will be of the utmost importance to ensure threat hunters can spend their precious time wisely! We do not want to have our analysts search through raw log files or create pivot tables in Excel… Excel and notepad should not 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 GPO's;
- 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 & hunt from your Kibana dashboards.

We will implement such an approach in our upcoming exercise!



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 & visualization of logs), the process can never be fully automated. The experience & 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 & 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 & Threat Hunting".

Course Roadmap

- Day 1: Knowing the adversary, knowing yourself
- Day 2: Averting Payload Delivery
- Day 3: Preventing Exploitation
- Day 4: Avoiding Installation, foiling Command & Control & thwarting lateral movement
- **Day 5: Exfiltration, Cyber Deception & Incident Response**
- Day 6: APT Defender Capstone

SEC599.5

Data exfiltration
- Typical data exfiltration strategies
- Exercise: Detecting data exfiltration using Suricata

Cyber deception strategies
- Tricking the adversary
- Exercise: Making your honeypot irresistibly sweet

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

Patrolling your network
- Proactive threat hunting strategies
- Exercise: Hunting your environment using OSQuery / ELK

Incident response
- Incident response process
- Exercise: Responding to an incident using GRR

This page intentionally left blank.

Exercise – Hunting Your Environment Using OSQuery & ELK

The objective of the lab is to implement a light-weight collection tool that will collect system information from the different endpoints in our environment. We will use this information to baseline the systems and detect anomalies!

High-level exercise steps:

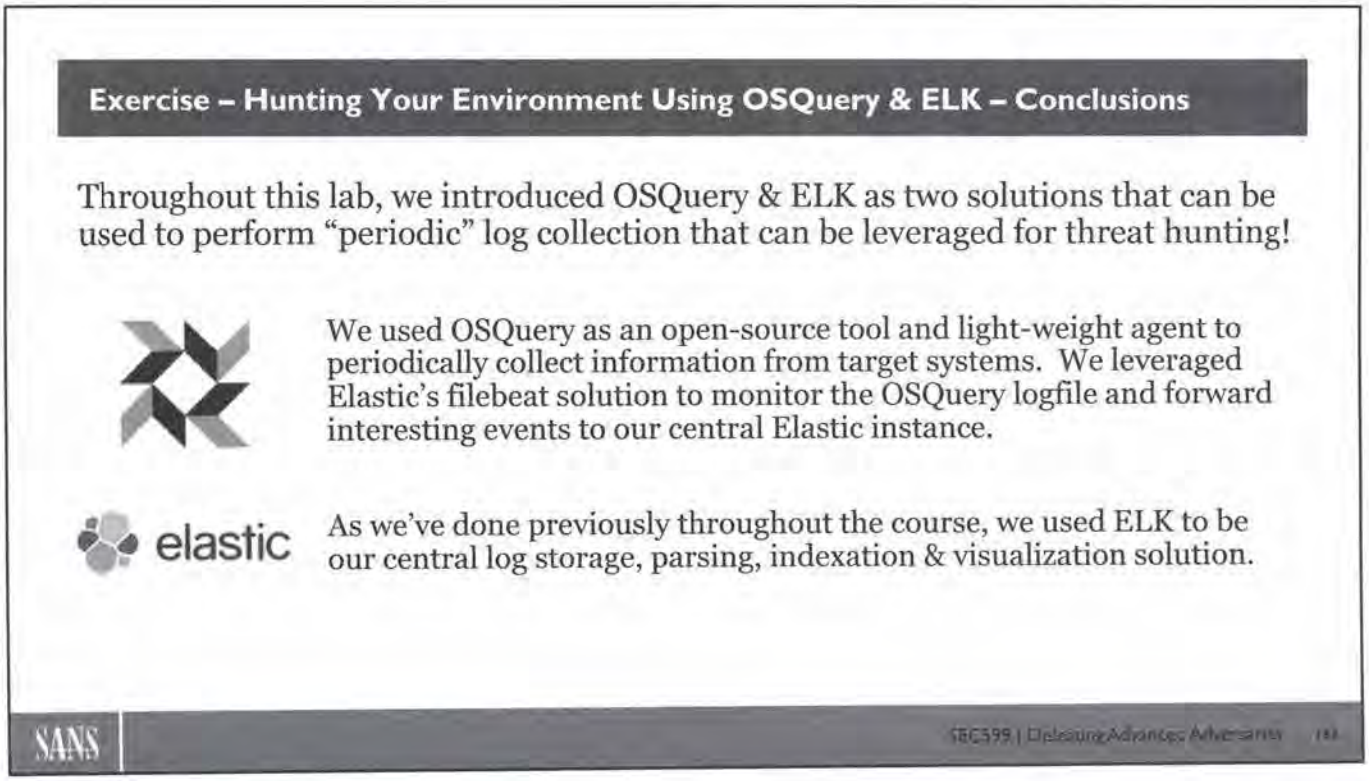
1. Configure OSQuery & test it on our local Windows workstation
2. Create a schedule to run OSQuery periodically
3. Configure filebeat to forward OSQuery output to ELK
4. Optional: Create visualizations in Kibana

Exercise – Hunting Your Environment Using OSQuery & ELK
The objective of the lab is to implement a light-weight collection tool that will collect system information from the different endpoints in our environment. We will use this information to baseline the systems and detect anomalies!

Following are high-level exercise steps

1. Configure OSQuery & test it on our local Windows workstation
2. Create a schedule to run OSQuery periodically
3. Configure filebeat to forward OSQuery output to ELK
4. Optional: Create visualizations in Kibana

For additional guidance & details on the lab, please refer to the LODS workbook.



Exercise – Hunting Your Environment Using OSQuery & ELK – Conclusions
Throughout this lab, we introduced OSQuery & ELK as two solutions that can be used to perform "periodic" log collection that can be leveraged for threat hunting!

We used OSQuery as an open-source tool and light-weight agent to periodically collect information from target systems. We leveraged Elastic's filebeat solution to monitor the OSQuery logfile and forward interesting events to our central Elastic instance.

As we've done previously throughout the course, we used ELK to be our central log storage, parsing, indexation & visualization solution.

Course Roadmap

- Day 1: Knowing the adversary, knowing yourself
- Day 2: Averting Payload Delivery
- Day 3: Preventing Exploitation
- Day 4: Avoiding Installation, foiling Command & Control & thwarting lateral movement
- **Day 5: Exfiltration, Cyber Deception & Incident Response**
- Day 6: APT Defender Capstone

SEC599.5

Data exfiltration
- Typical data exfiltration strategies
- Exercise: Detecting data exfiltration using Suricata

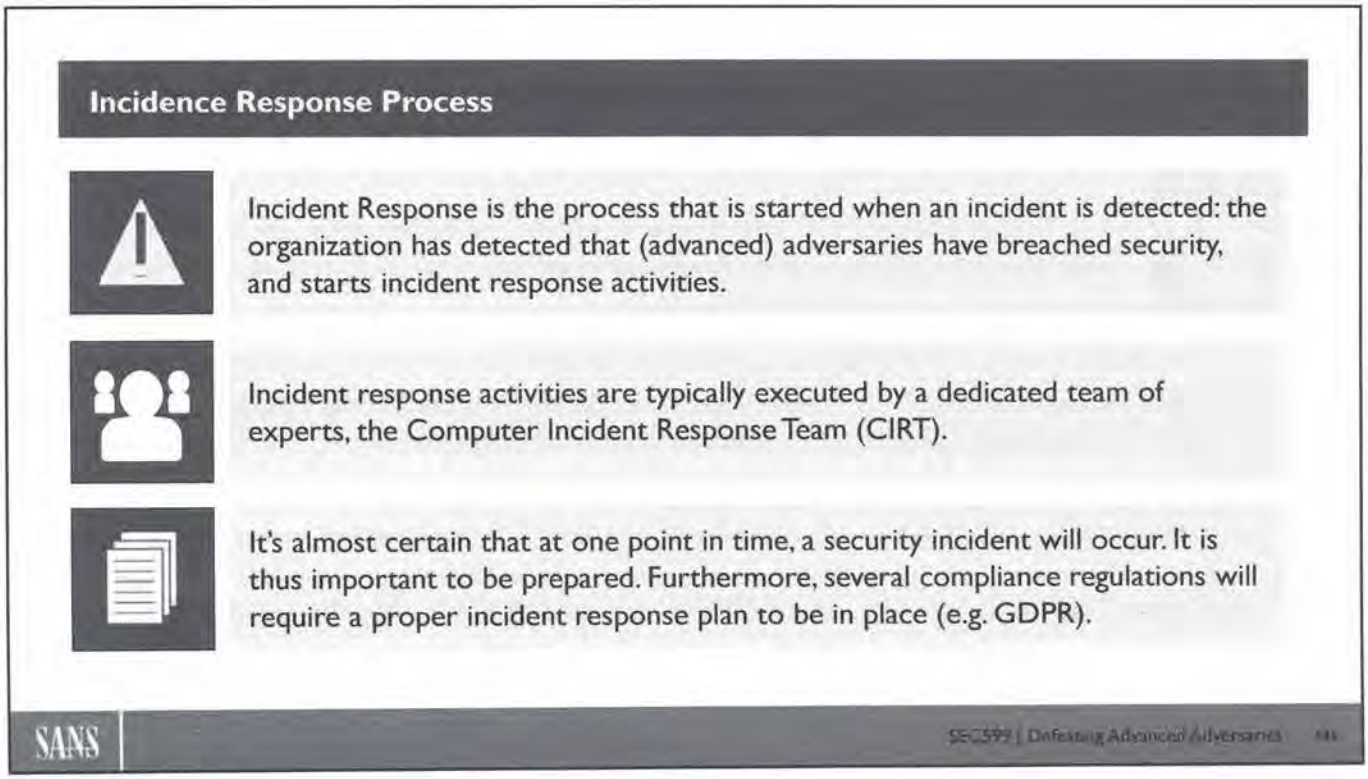
Cyber deception strategies
- Tricking the adversary
- Exercise: Making your honeypot irresistibly sweet

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

Patrolling your network
- Proactive threat hunting strategies
- Exercise: Hunting your environment using OSQuery / ELK

Incident response
- Incident response process
- Exercise: Responding to an incident using GRR

This page intentionally left blank.



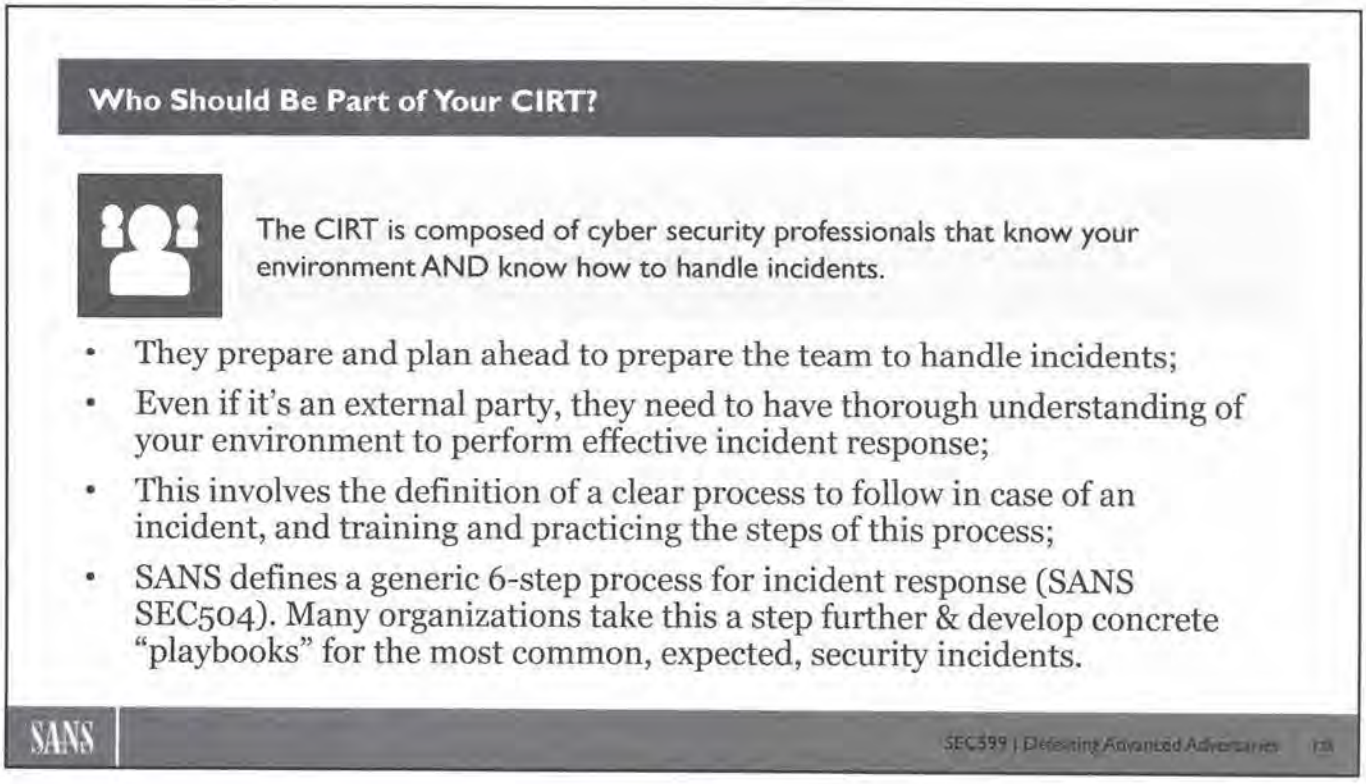
Incidence Response Process

The 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.



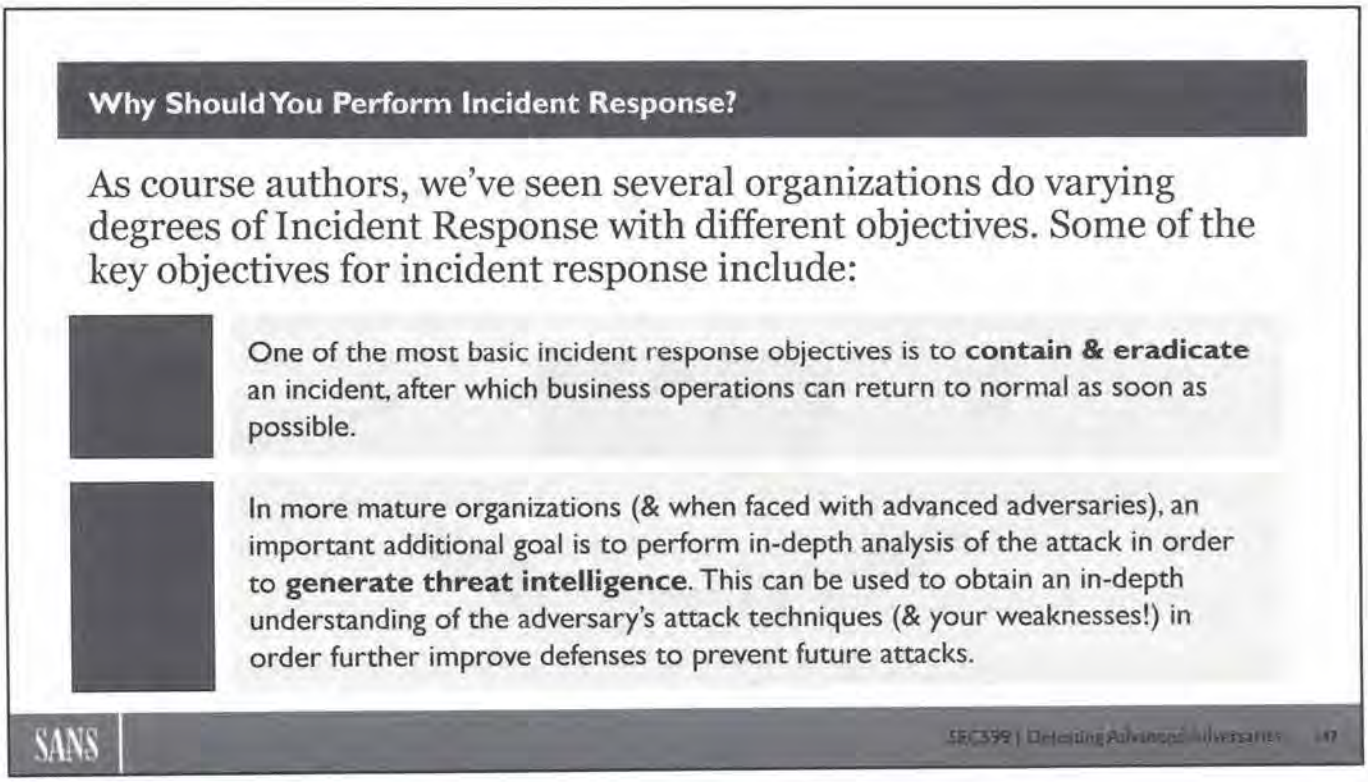
Who Should Be Part of Your CIRT?

The 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.

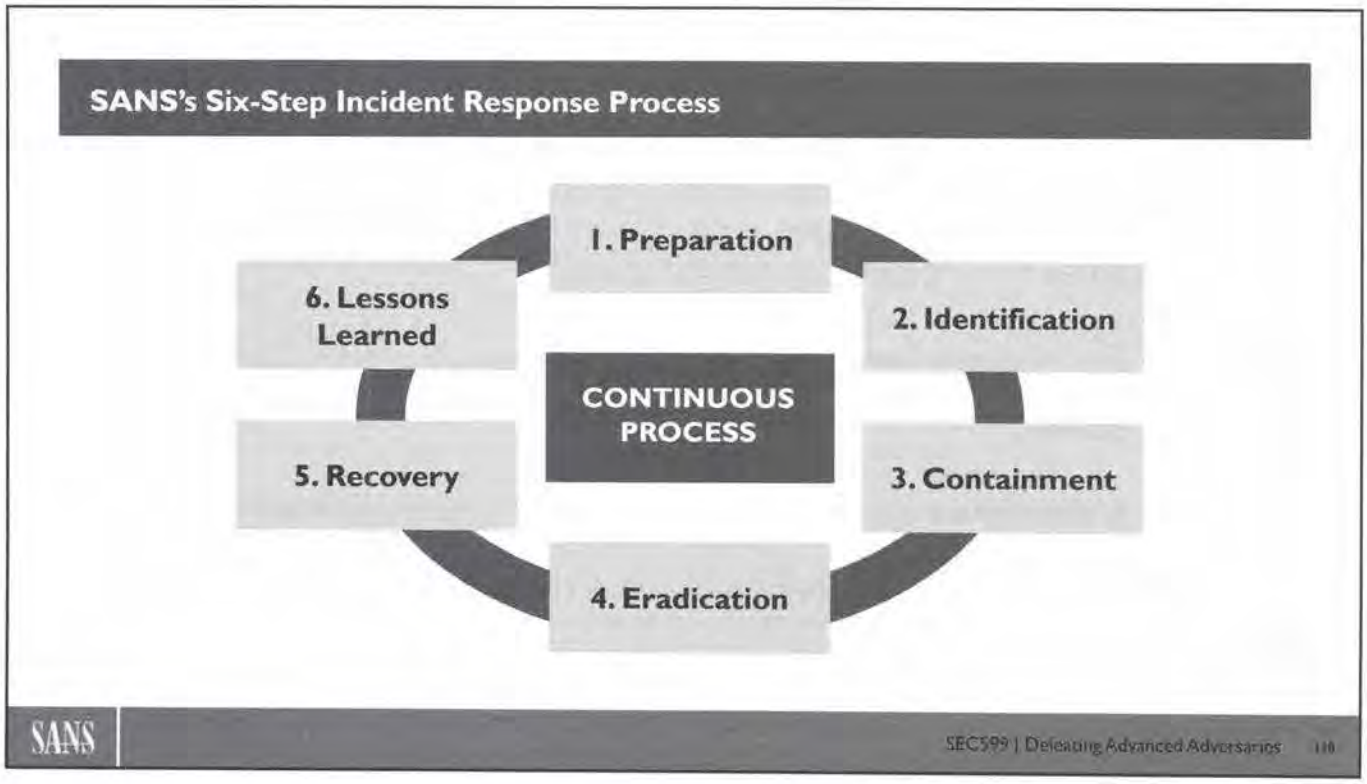
To help prepare CIRTs and CIRT team members responding to incidents, SANS defines a 6-step process for incident response. This incident response process is covered in SANS training SEC504: "Hacker Tools, Techniques, Exploits, and Incident Handling". Many organizations take this a step further & develop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



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 & eradicate an incident, after which business operations can return to normal as soon as possible.
- In more mature organizations (& 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 (& your weaknesses!) in order 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, or there could be scenarios where one objective is more important than the other. Consider a ransomware incident in a smaller organization for example. The focus will most likely more be on containing & 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 for example, there is probably a lot of benefit to ensure the incident response



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

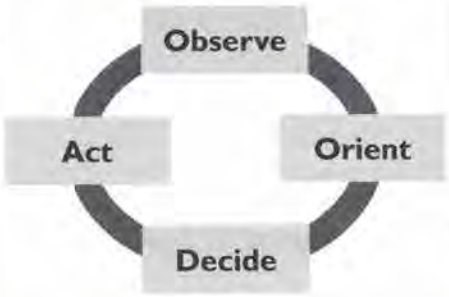
We recommended going through this 6-step process step by step without skipping steps. It's important to note that this should always be a continuous process: Once you finish incident response, you derive lessons learned (e.g. TTPs used by your adversaries, missing monitoring capabilities, vulnerabilities in your environment, …) that will improve your preparation phase, which will increase the chances of you detecting the incident, ….
We will now discuss these 6 steps in detail in the upcoming slides.

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

Incidence Response & the OODA Loop

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" & "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.

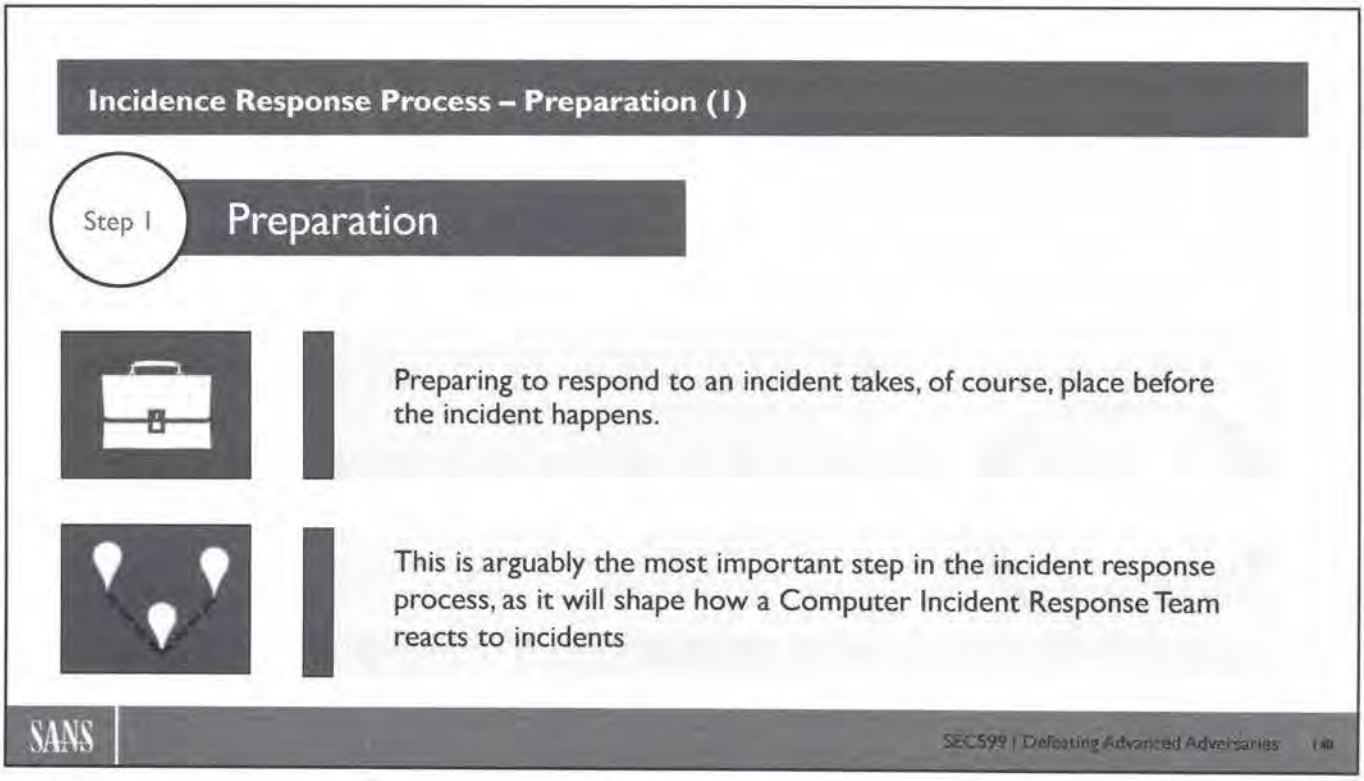
During a typical incident response engagement, your CIRT team & the adversary run through countless OODA loops. **If you OODA loops are faster than the adversary, you win ☺**

Incidence Response & 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 & 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 & 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 on 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" & "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 & the adversary run through countless OODA loops. If you OODA loops are faster than the adversary, you win!



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

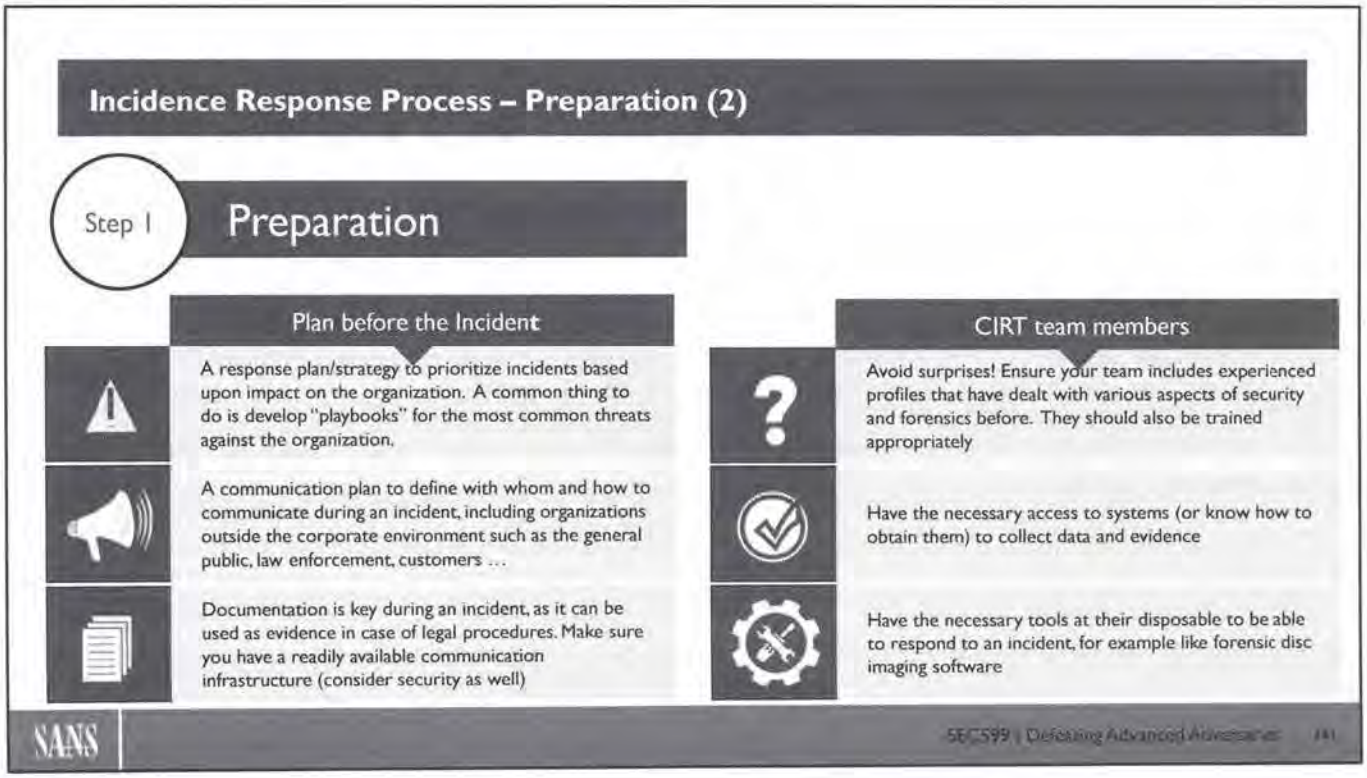
Of course, the CIRT team and CIRT members prepare to respond to incidents before incidents take place.

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, for example, 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.



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. Remark that 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, for example, 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.

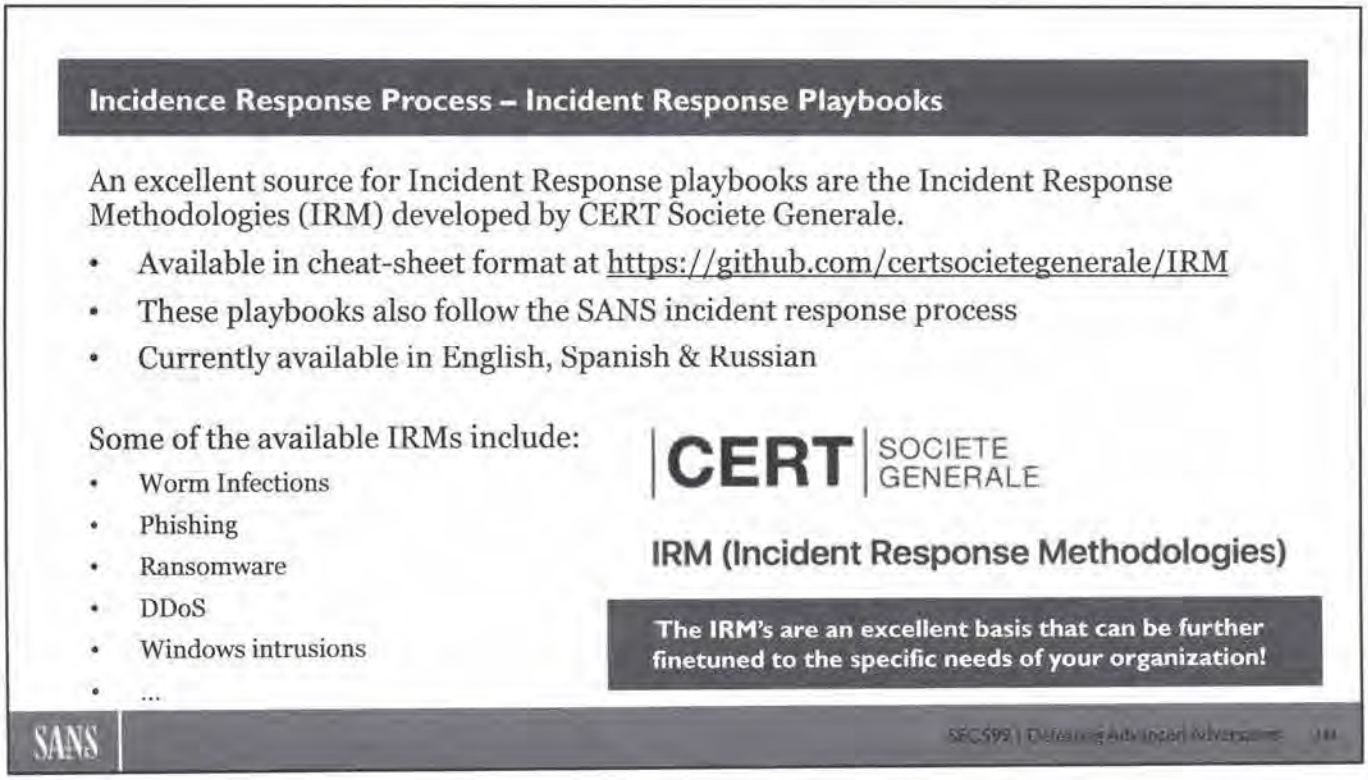
A CIRT team must be formed with members that:

- 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 experience 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 disposable 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.



Incidence Response Process – Incident Response Playbooks

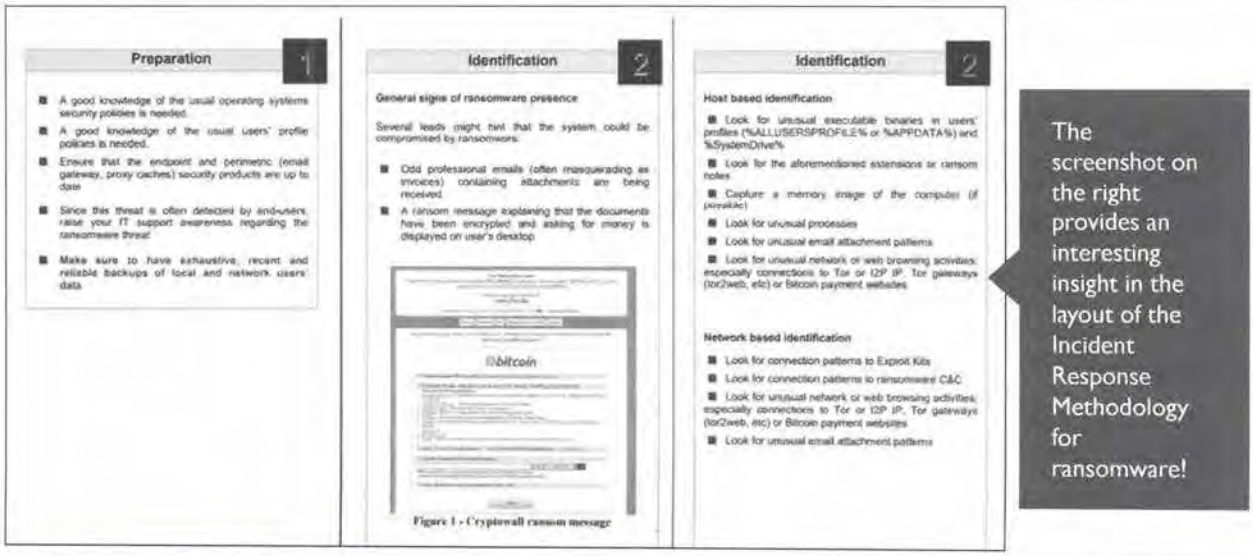
In the modern age, many organizations are facing similar threats: it's no surprise that a variety of companies could fall victim to ransomware or DDoS attacks. This means that, as an organization, we shouldn't attempt to reinvent the wheel, as other organizations may have thought of the same problems before (& found a solution).

An excellent source of Incident Response playbooks are the "Incident Response Methodologies" developed by CERT Societe Generale, 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 & Russian.

Over 15 playbooks are currently available, covering some of the most common incident types including: phishing, ransomware, DDoS, …

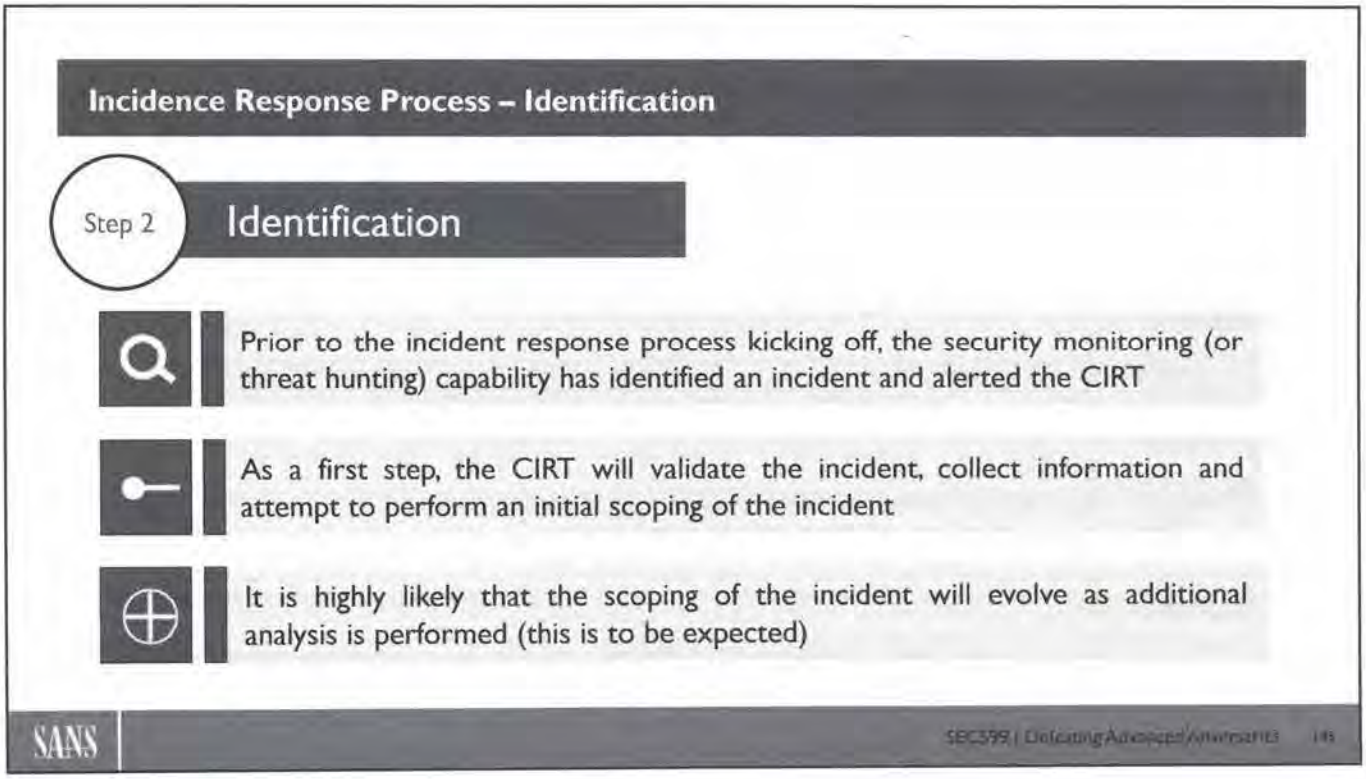
As an organization, it is recommended to take these playbooks as a basis & further tailor them to your organization!

Incidence Response Process – Example IRM Layout for Ransomware



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 2 steps of the Incident Response as defined by SANS!



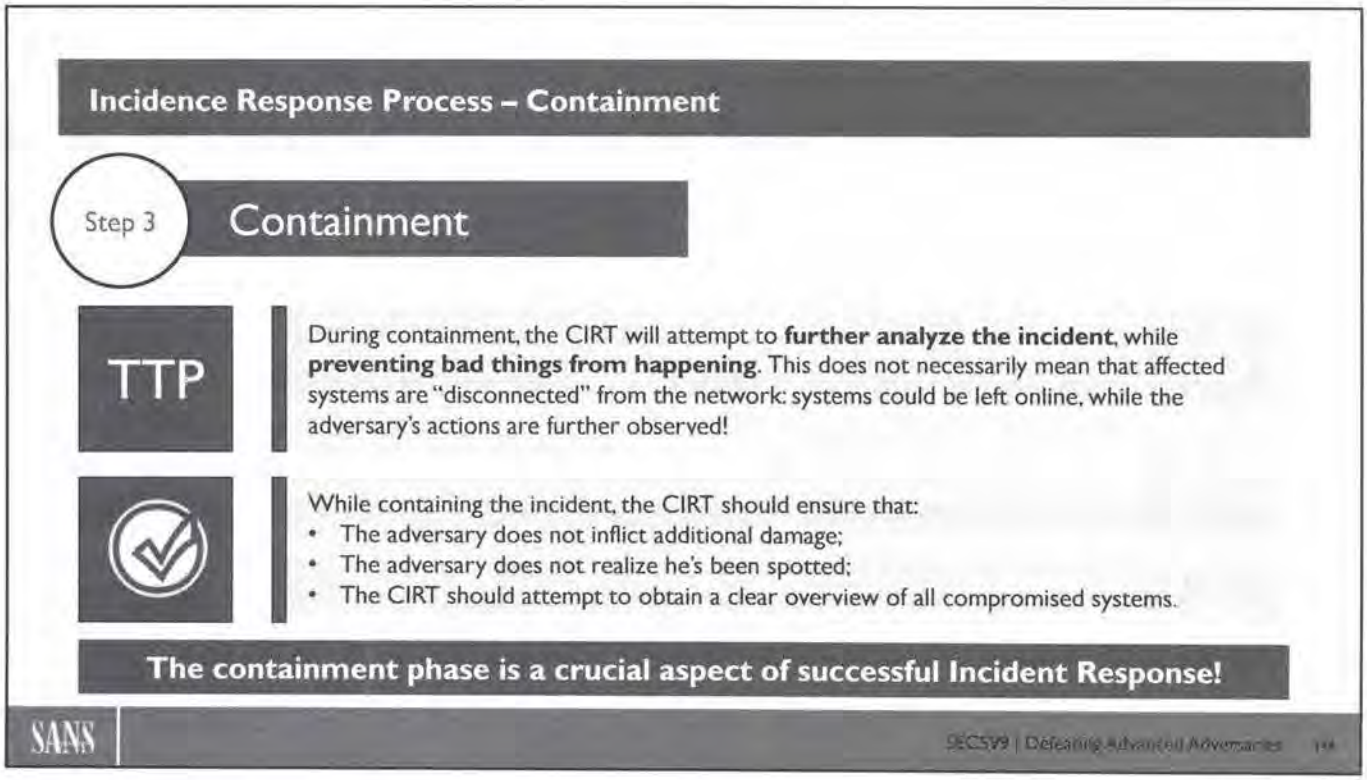
Incidence Response Process – Identification
The second step in the incident response process is "Identification".

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.



Incidence Response Process – Containment

The third step in the incident response process is "Containment". Containment is often misunderstood for being synonymous with immediately disconnecting and infected systems. This doesn't necessarily have to be the case however:

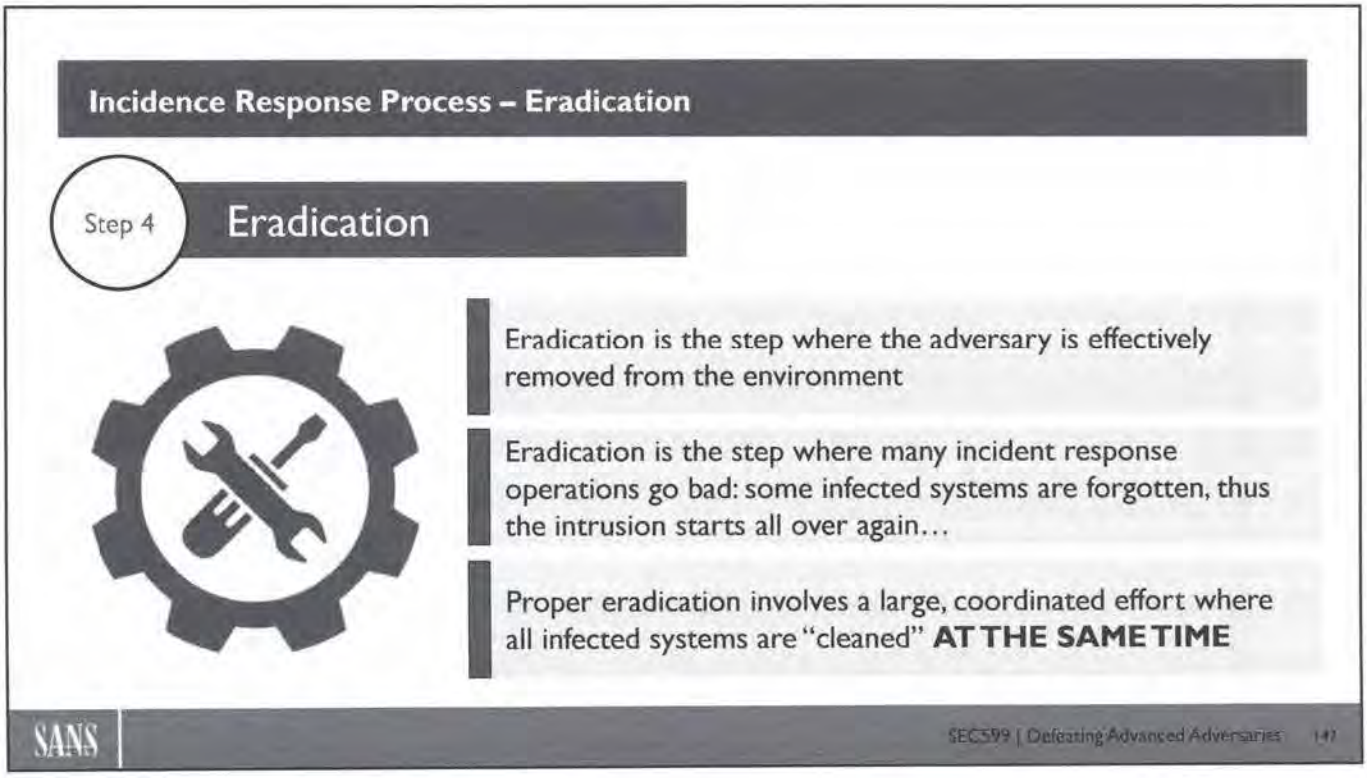
During containment, the CIRT will attempt to further analyze the incident, while preventing bad things from happening. This could mean that a system is left online, but will now be subject to increased monitoring, in order to obtain more information on the incident:

- 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)

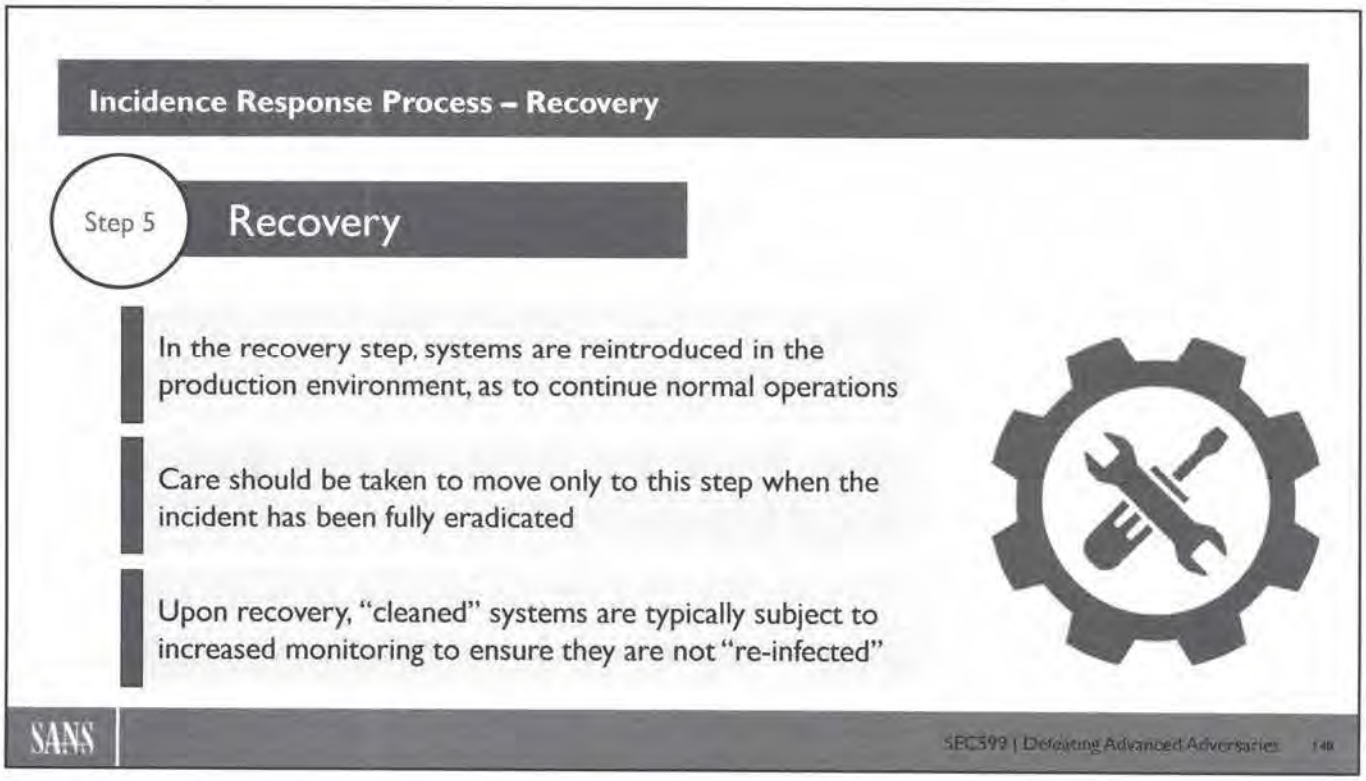
We can compare this to the analogy of a terrorist cell discovered by the FBI. Upon discovery, it is likely that the FBI will increase monitoring to obtain more information on what they are trying to do, as this could be a treasure of threat intelligence. At the same time, they should ensure they can intervene when they would move to do something actually bad.



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!

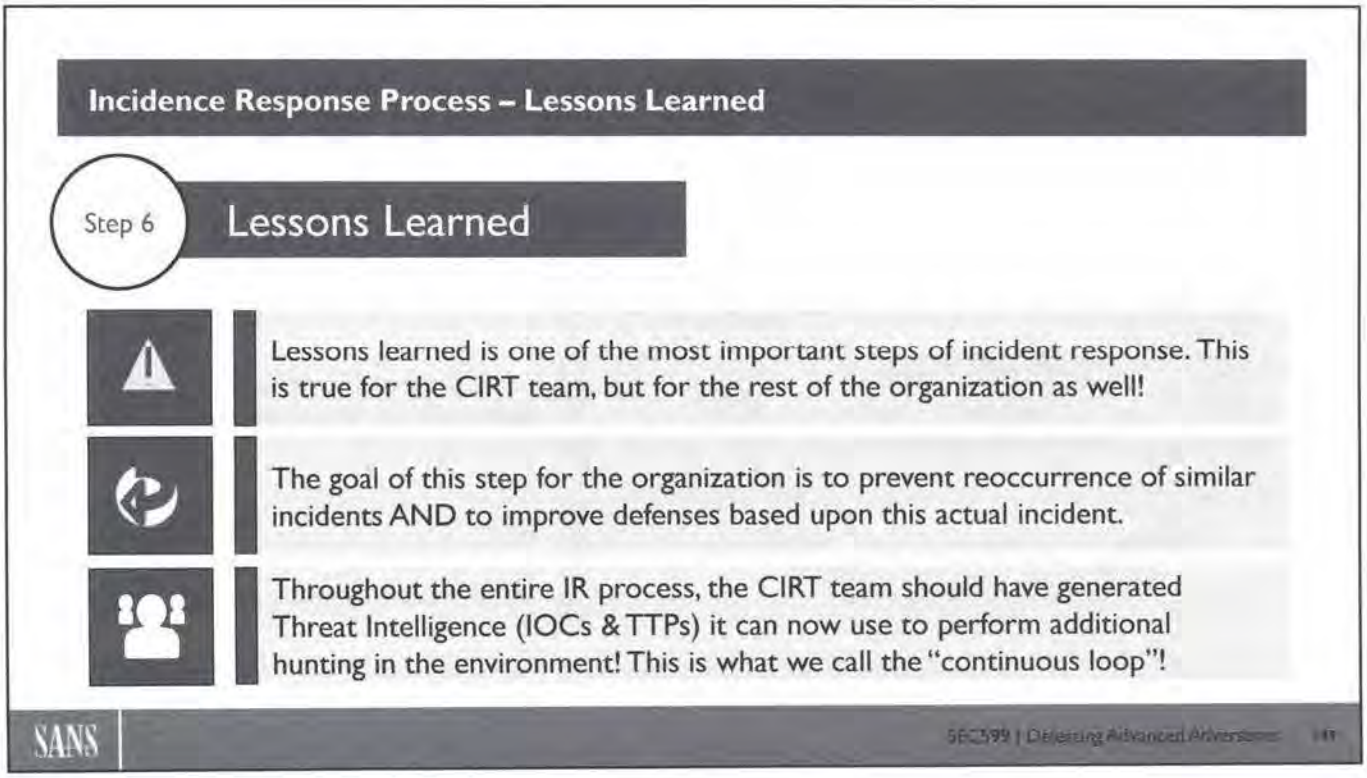


Incidence Response Process – Recovery
The fifth step in the incident response process is "Recovery".

In the recovery step, systems are reintroduced in the production environment, as 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 "re-infected". This monitoring can be highly tailored, as proper analysis of the incident should have resulted in numerous IOCs & TTPs employed by this particular adversary.



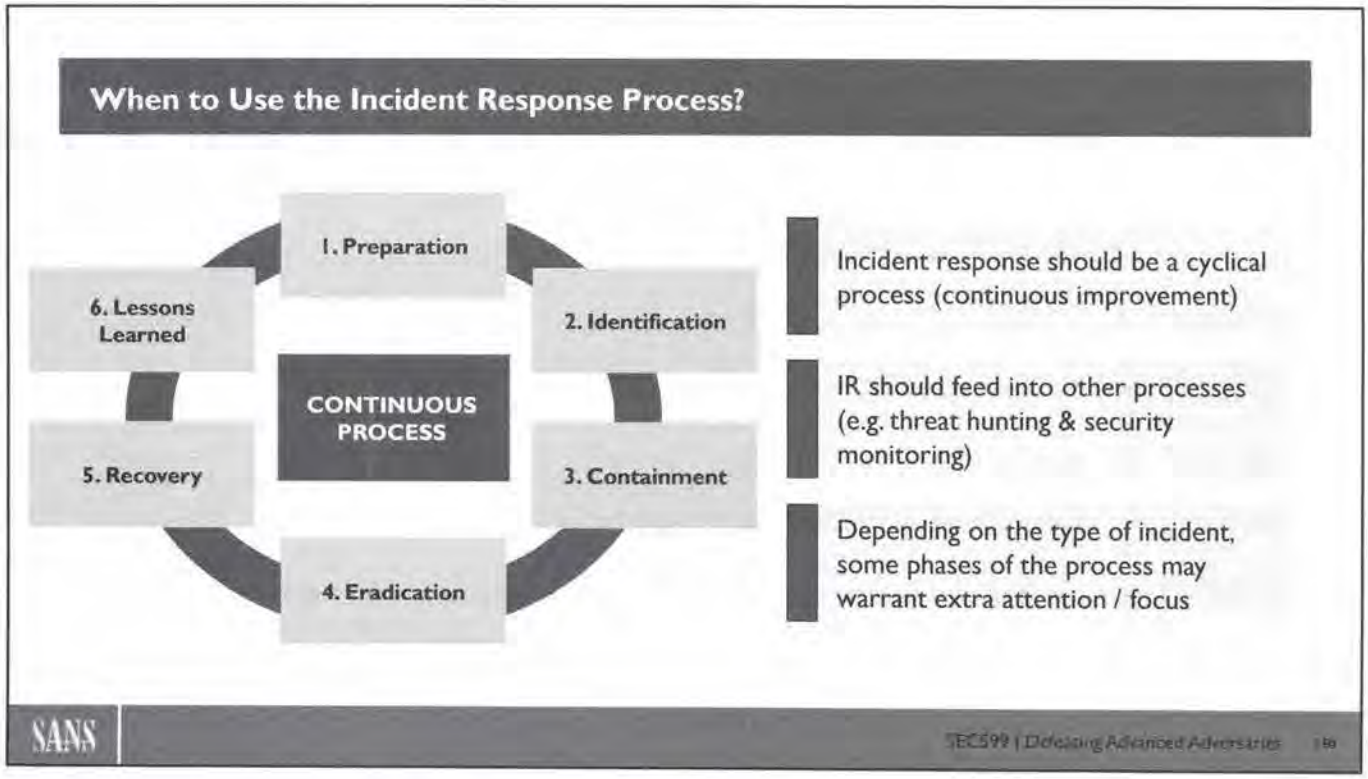
Incidence Response Process – Lessons Learned
The sixth and last step in the incident response process is "Lessons Learned".

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"!



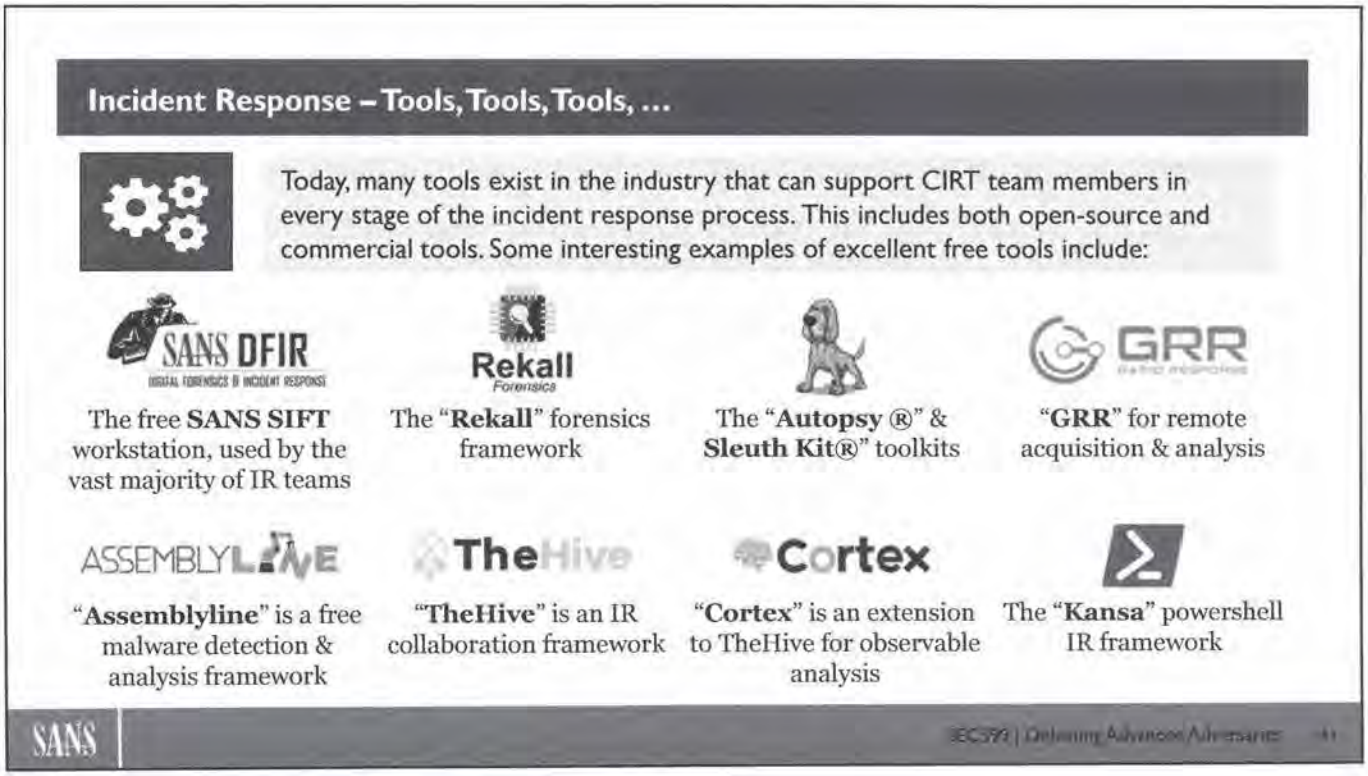
When to Use the Incident Response Process?

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 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 & security monitoring. When the malware that is analyzed during an incident is fully reversed, this will probably lead to IOCs & 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.



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
- Rekall is an advanced forensic and incident response framework. While it began life purely as a memory forensic framework, it has now evolved into a complete platform. Rekall implements the most advanced analysis techniques in the field, while still being developed in the open, with a free and open source license. Many of the innovations implemented within Rekall have been published in peer reviewed papers. Rekall is available at http://www.rekall-forensic.com/
- 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 Sleuthkit and a memory forensics capability in the form of Rekall. You can get GRR at 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, in to 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://www.cse-cst.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/CERT-BDF/TheHive
- 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/CERT-BDF/Cortex
- 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 at https://github.com/davehull/Kansa.

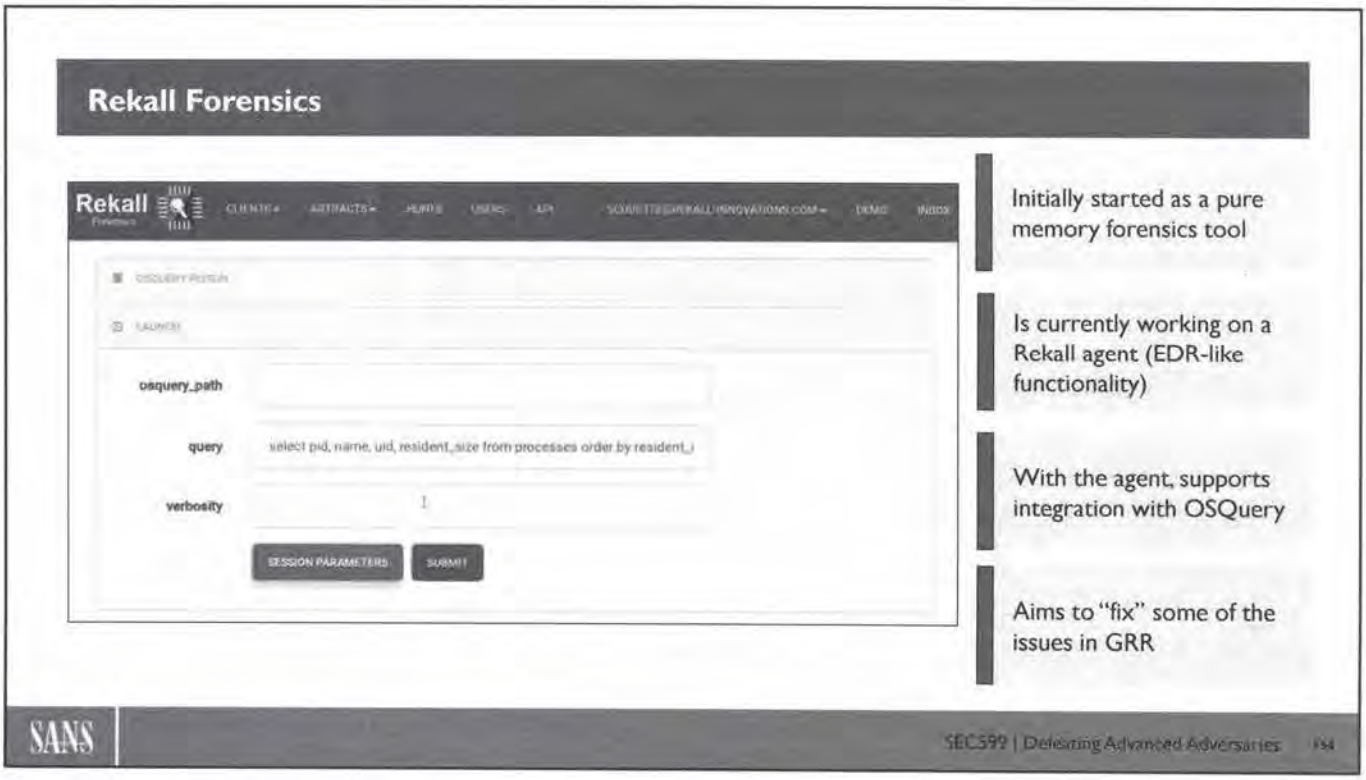
SANS SIFT Workstation

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, the 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, the SIFT Workstation is taught only in the following incident response courses at SANS:

- Advanced Incident Response course (FOR508)
- Advanced Network Forensics course (FOR572)
- Cyber Threat Intelligence (FOR578)
- Memory Analysis In-depth (FOR526)



Rekall Forensics

Rekall is an advanced forensic and incident response framework. While it began life purely as a memory forensic framework, it has now evolved into a complete platform. Rekall implements the most advanced analysis techniques in the field, while still being developed in the open, with a free and open source license. Many of the innovations implemented within Rekall have been published in peer reviewed papers.

Current development efforts are focused amongst others on the Rekall agent, which aims to provide EDR-like functionality. The current version of the agent also supports for example OSQuery queries, making it a highly interesting tool! Rekall Agent is a complete endpoint incident response and forensic tool. The Rekall Agent extends Rekall's advanced capabilities to a scalable, distributed environment. The Rekall Agent is easy to deploy and scale, based on modern cloud technologies. With enterprise grade access control and auditing features built in, the Rekall Agent is suitable to be deployed in small to large scale enterprises to provide unprecedented visibility of endpoint security, and collection and preservation of volatile endpoint evidence. Rekall Agent can be downloaded from GitHub.

Autopsy & the Sleuth Kit (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:

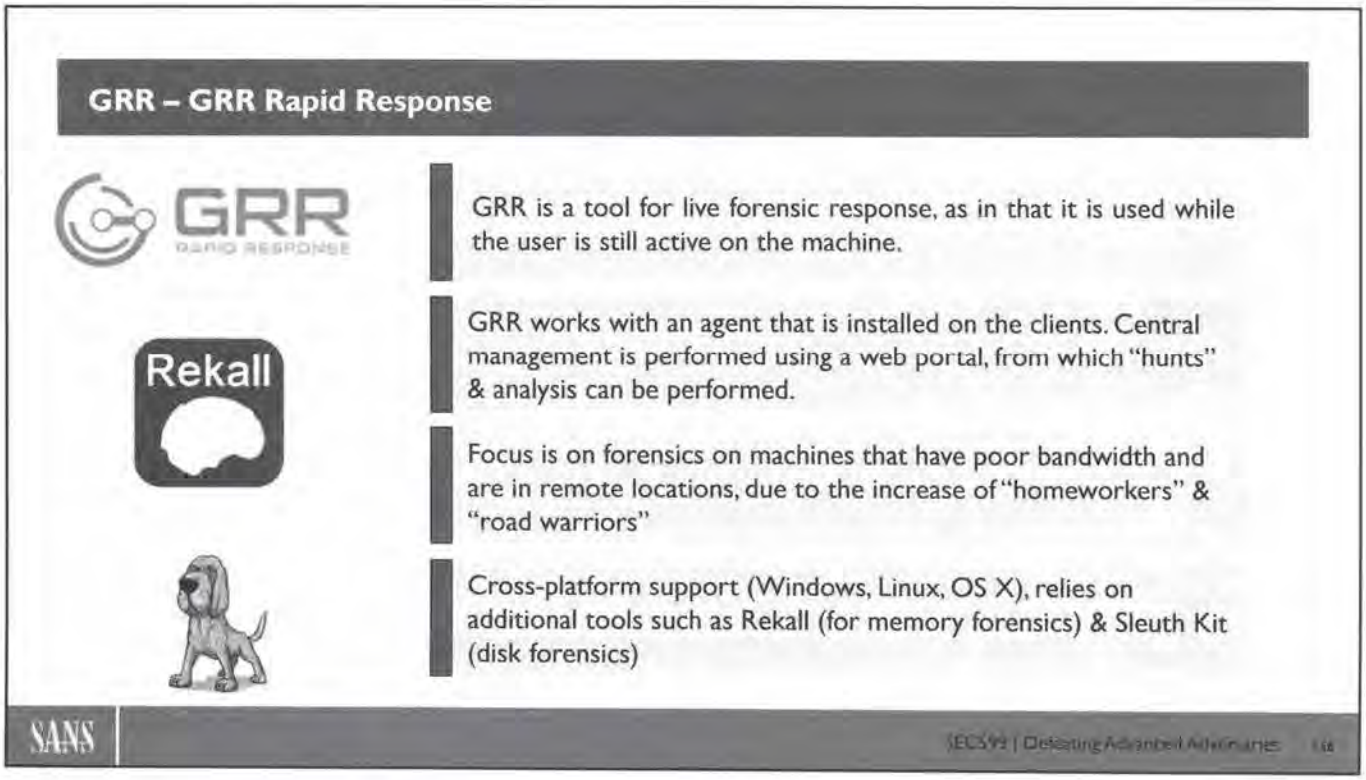
- Timeline Analysis, including an advanced graphical event viewing interface
- Hash Filtering, which can be used to flag known bad files & ignore known goods
- Keyword Search, allowing to find files related to relevant terms
- Web Artifacts, allow extraction of history, bookmarks, and cookies from popular browsers
- Data Carving, allowing recovery of deleted files
- Multimedia, allowing extraction of EXIF data from pictures and videos
- Indicators of Compromise, allowing the scanning of 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 & the Sleuth Kit (By Basis Technology)
Autopsy & 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, including an advanced graphical event viewing interface
- Hash Filtering, which can be used to flag known bad files & ignore known goods
- Keyword Search, allowing to find files related to relevant terms
- Web Artifacts, allow extraction of history, bookmarks, and cookies from popular browsers
- Data Carving, allowing recovery of deleted files
- Multimedia, allowing extraction of EXIF data from pictures and videos
- Indicators of Compromise, allowing the scanning of 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 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, 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 Sleuthkit 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 in a remote location. More and more "homeworkers" & "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, 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
- File system
- 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

Assembyline 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 web site:

"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 which focusses 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 IOC's.

TheHive also has a direct integration with Cuckoo Sandbox

TheHive

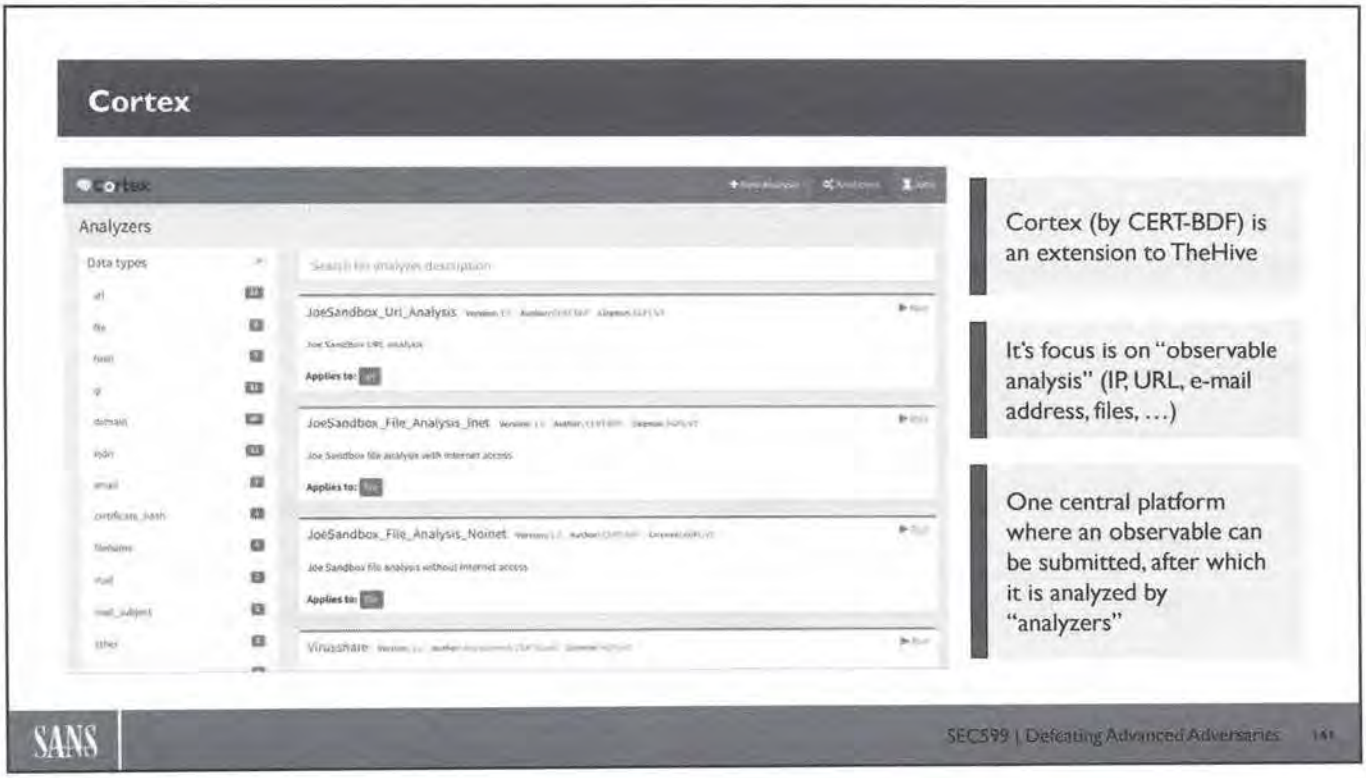
It is often so, that during investigations multiple teams are collaborating, each of which have their own insights and log sources. In order to share this information and knowledge, it is good to have a platform that allows you to create different cases in which IOC's and case notes can be shared.

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 IOC's.

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

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 & reporting of analyzer outputs.

Kansa – A PowerShell IR Framework

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 a.o.:

- Get-SchedTasks
- Get-Autorunsc
- Get-SvcAll
- Get-LocalAdmins
- Get-MasterFileTable
- Get-IOCsByPath
- Get-RdpConnectionLogs

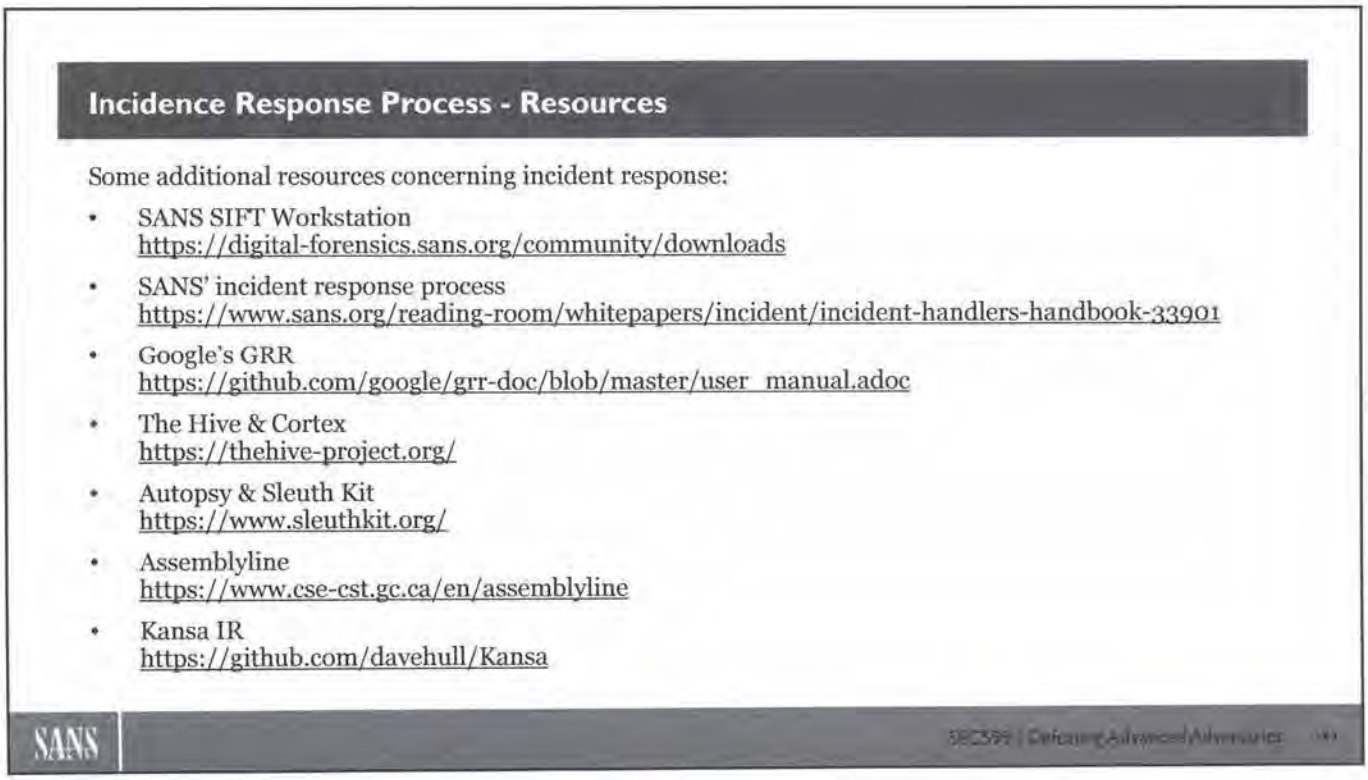
Kansa – A PowerShell IR Framework
Kansa is a modular incident response framework in PowerShell created by Dave Hull.

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 amongst 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



Incidence Response Process – Resources
Some additional resources concerning incident response process:

SANS SIFT Workstation
https://digital-forensics.sans.org/community/downloads

SANS' incident response process
https://www.sans.org/reading-room/whitepapers/incident/incident-handlers-handbook-33901

Google's GRR
https://github.com/google/grr-doc/blob/master/user_manual.adoc

The Hive & Cortex
https://thehive-project.org/

Autopsy & Sleuth Kit
https://www.sleuthkit.org/

Assemblyline
https://www.cse-cst.gc.ca/en/assemblyline

Kansa IR
https://github.com/davehull/Kansa

Course Roadmap

- Day 1: Knowing the adversary, knowing yourself
- Day 2: Averting Payload Delivery
- Day 3: Preventing Exploitation
- Day 4: Avoiding Installation, foiling Command & Control & thwarting lateral movement
- **Day 5: Exfiltration, Cyber Deception & Incident Response**
- Day 6: APT Defender Capstone

SEC599.5

Data exfiltration
- Typical data exfiltration strategies
- Exercise: Detecting data exfiltration using Suricata

Cyber deception strategies
- Tricking the adversary
- Exercise: Making your honeypot irresistibly sweet

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

Patrolling your network
- Proactive threat hunting strategies
- Exercise: Hunting your environment using OSQuery / ELK

Incident response
- Incident response process
- Exercise: Responding to an incident using GRR

This page intentionally left blank.

Exercise – Incident Response Using GRR

During this lab, we will introduce GRR as a remote forensics tools. We will install the GRR agent on one of our Windows workstations, after which we will use GRR to browse the remote filesystem, acquire a remote memory dump & launch a hunt looking for suspicious files!

High-level exercise steps:

1. Deploying GRR on one of our Windows-based endpoints
2. Browsing the remote filesystem from the GRR management console
3. Acquiring a remote memory dump from the GRR management console
4. Launching a hunt looking for suspicious files using GRR

Exercise – Incident Response Using GRR
During this lab, we will introduce GRR as a remote forensics tool. We will install the GRR agent on one of our Windows workstations, after which we will use GRR to browse the remote filesystem, acquire a remote memory dump & launch a hunt looking for suspicious files!

The following are high-level exercise steps we'll need to complete:

1. Deploying GRR on one of our Windows-based endpoints
2. Browsing the remote filesystem from the GRR management console
3. Acquiring a remote memory dump from the GRR management console
4. Launching a hunt looking for suspicious files using GRR

For additional guidance & details on the lab, please refer to the LODS workbook.

Exercise – Incident Response Using GRR – Conclusions

Throughout this lab, we introduced GRR as a solution for remote incident response & forensic analysis / acquisition

We installed a GRR agent on one of our workstations and used it to perform both a live hunt (e.g. for an identified IoC) and for a remote memory acquisition!

It should be noted that GRR is very much a work in progress and it is still actively being improved. While not necessarily offering the same ease-of-use as many commercial tools, it does provide a powerful interface to start doing remote forensics!

Exercise – Incident Response Using GRR – Conclusions
Throughout this lab, we introduced GRR as a solution for remote incident response & forensic analysis / acquisition.

We installed a GRR agent on one of our workstations and used it to perform both a live hunt (e.g. for an identified IOC) and for a remote memory acquisition!

It should be noted that GRR is very much a work in progress and it is still actively being improved. While not necessarily offering the same ease-of-use as many commercial tools, it does provide a powerful interface to start doing remote forensics!

Course Roadmap

- Day 1: Knowing the adversary, knowing yourself
- Day 2: Averting Payload Delivery
- Day 3: Preventing Exploitation
- Day 4: Avoiding Installation, foiling Command & Control & thwarting lateral movement
- **Day 5: Exfiltration, Cyber Deception & Incident Response**
- Day 6: APT Defender Capstone

SEC599.5

Data exfiltration
- Typical data exfiltration strategies
- Exercise: Detecting data exfiltration using Suricata

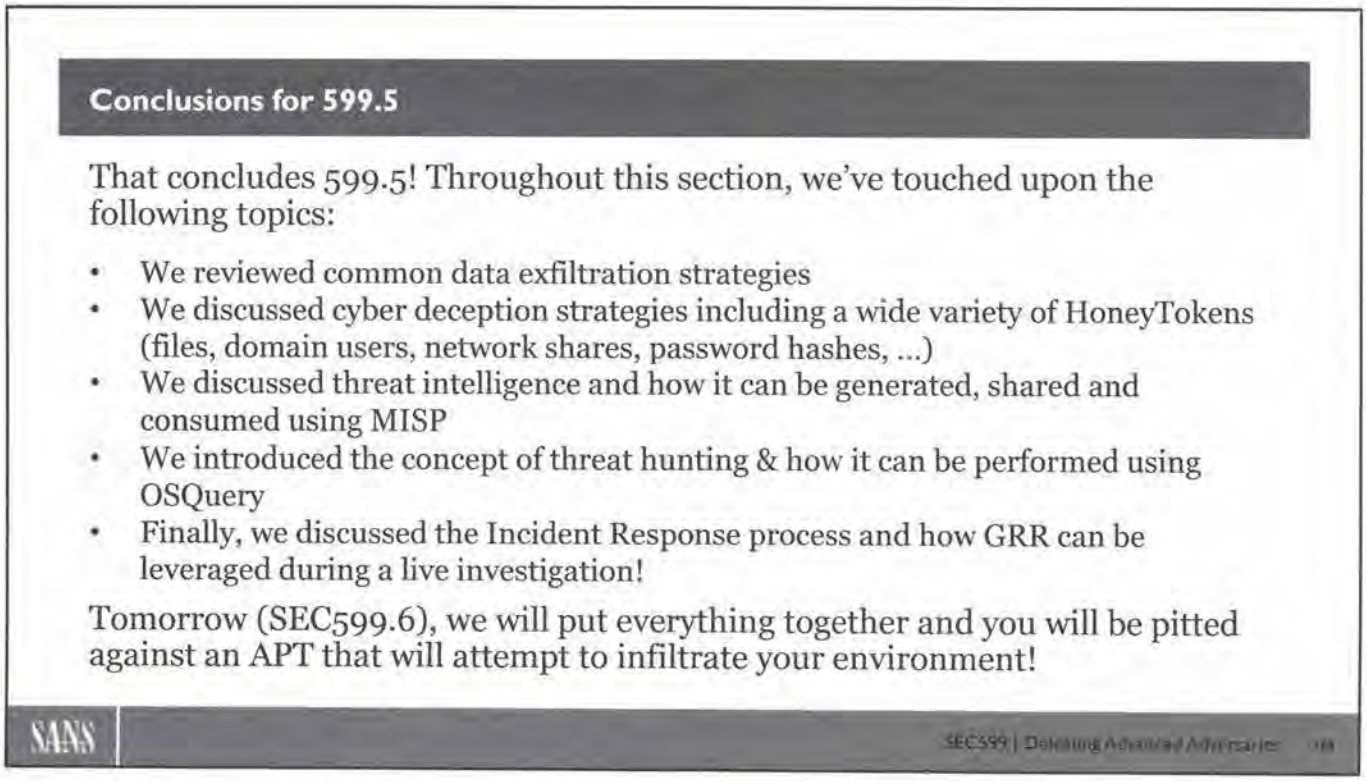
Cyber deception strategies
- Tricking the adversary
- Exercise: Making your honeypot irresistibly sweet

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

Patrolling your network
- Proactive threat hunting strategies
- Exercise: Hunting your environment using OSQuery / ELK

Incident response
- Incident response process
- Exercise: Responding to an incident using GRR

This page intentionally left blank.



Conclusions for 599.5

That concludes 599.5! Throughout this section, we've touched upon the following topics:

- We reviewed common data exfiltration strategies
- We discussed cyber deception strategies including a wide variety of HoneyTokens (files, domain users, network shares, password hashes, ...)
- We introduced the concept of threat hunting & how it can be performed using OSQuery
- We discussed threat intelligence and how it can be generated, shared and consumed using MISP
- Finally, we discussed the Incident Response process and how GRR 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!