SEC450 | BLUE TEAM FUNDAMENTALS: SECURITY OPERATIONS AND ANALYSIS

450.4

Triage and Analysis

SANS

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

PLEASE READ THE TERMS AND CONDITIONS OF THIS COURSEWARE LICENSE AGREEMENT ("CLA") CAREFULLY BEFORE USING ANY OF THE COURSEWARE ASSOCIATED WITH THE SANS COURSE. THIS IS A LEGAL AND ENFORCEABLE CONTRACT BETWEEN YOU (THE “USER”) AND SANS INSTITUTE FOR THE COURSEWARE. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

With the CLA, SANS Institute hereby grants User a personal, non-exclusive license to use the Courseware subject to the terms of this agreement. Courseware includes all printed materials, including course books and lab workbooks, as well as any digital or other media, virtual machines, and/or data sets distributed by SANS Institute to User for use in the SANS class associated with the Courseware. User agrees that the CLA is the complete and exclusive statement of agreement between SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA.

BY ACCEPTING THIS COURSEWARE, YOU AGREE TO BE BOUND BY THE TERMS OF THIS CLA. BY ACCEPTING THIS SOFTWARE, YOU AGREE THAT ANY BREACH OF THE TERMS OF THIS CLA MAY CAUSE IRREPARABLE HARM AND SIGNIFICANT INJURY TO SANS INSTITUTE, AND THAT SANS INSTITUTE MAY ENFORCE THESE PROVISIONS BY INJUNCTION (WITHOUT THE NECESSITY OF POSTING BOND) SPECIFIC PERFORMANCE, OR OTHER EQUITABLE RELIEF.

If you do not agree, you may return the Courseware to SANS Institute for a full refund, if applicable.

User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of the Courseware, in any medium whether printed, electronic or otherwise, for any purpose, without the express prior written consent of SANS Institute. Additionally, User may not sell, rent, lease, trade, or otherwise transfer the Courseware in any way, shape, or form without the express written consent of SANS Institute.

If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this Courseware.

SANS acknowledges that any and all software and/or tools, graphics, images, tables, charts or graphs presented in this Courseware are the sole property of their respective trademark/registered/copyright owners, including:

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There’s an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.

PMP and PMBOK are registered marks of PMI.

SOF-ELK® is a registered trademark of Lewes Technology Consulting, LLC. Used with permission.

SIFT® is a registered trademark of Harbingers, LLC. Used with permission.

Governing Law: This Agreement shall be governed by the laws of the State of Maryland, USA.

SEC450.4

Blue Team Fundamentals: Security Operations and Analysis

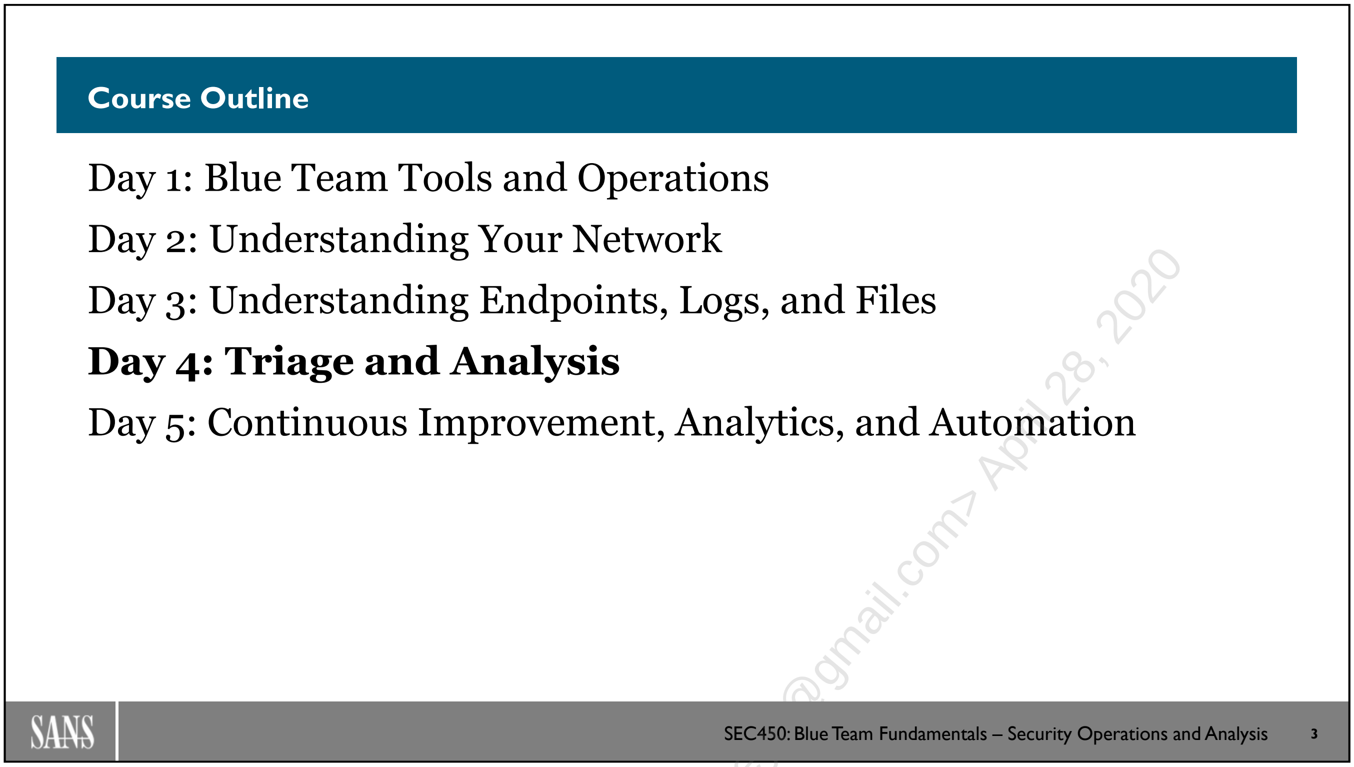
SANS

Triage and Analysis

Welcome to SANS Security 450.4 – Triage and Analysis

TABLE OF CONTENTS

This page intentionally left blank.



Course Outline

Day 1: Blue Team Tools and Operations

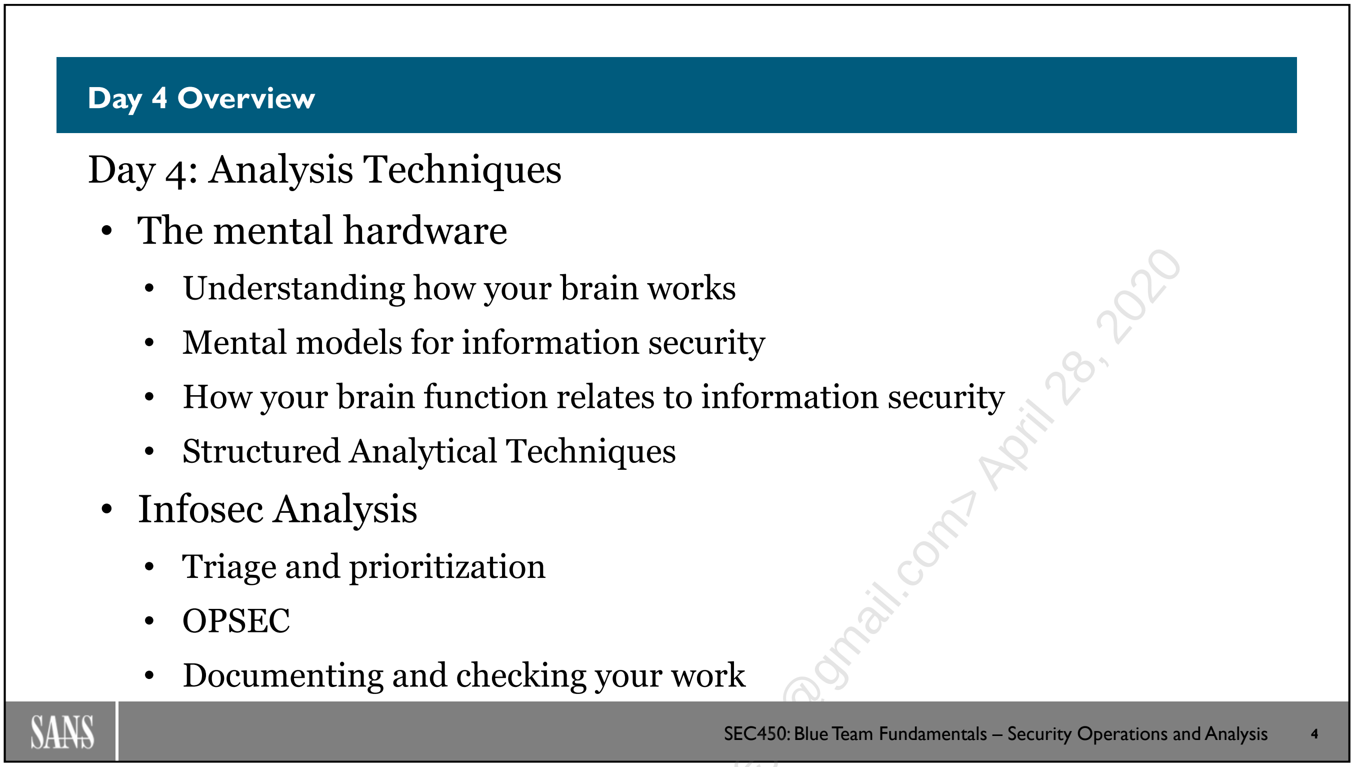
Day 2: Understanding Your Network

Day 3: Understanding Endpoints, Logs, and Files

Day 4: Triage and Analysis

Day 5: Continuous Improvement, Analytics, and Automation

This page intentionally left blank.

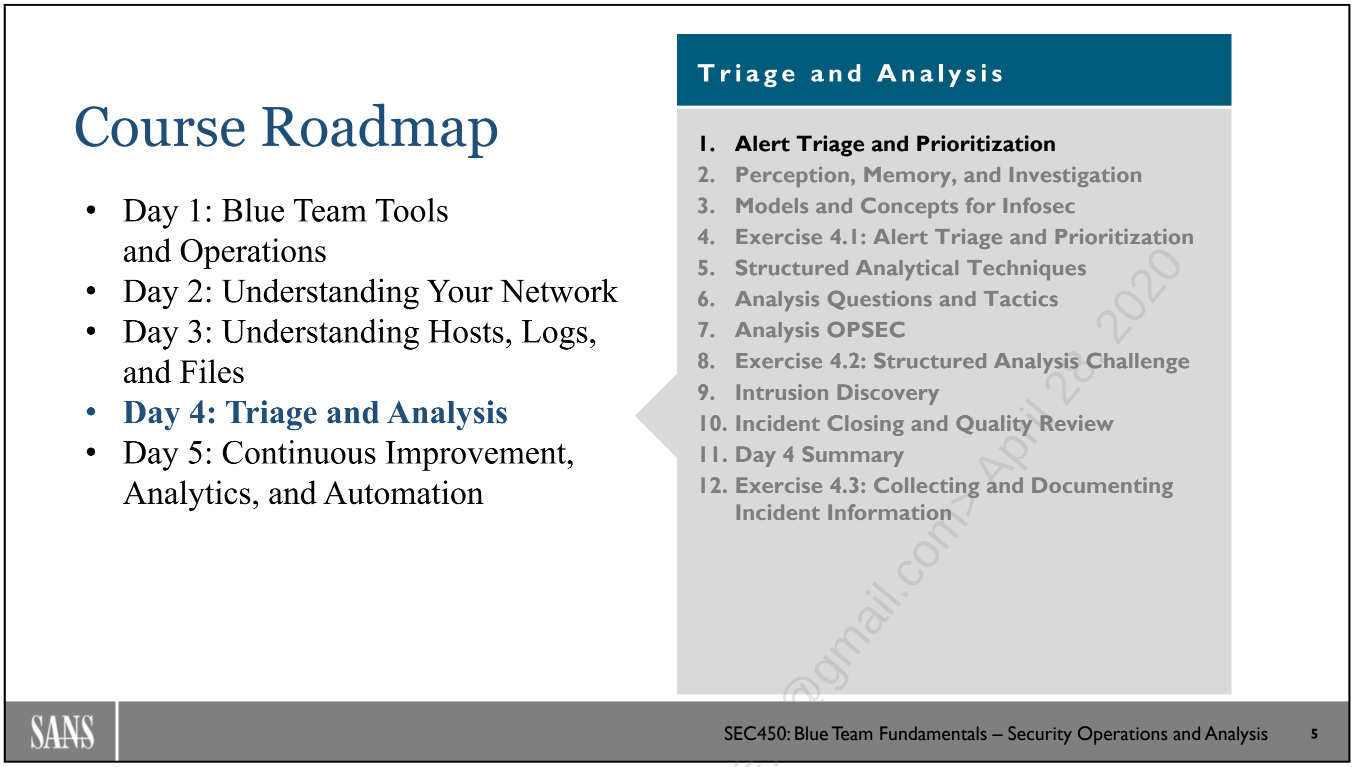


Day 4 Overview

Day 4: Analysis Techniques

- The mental hardware
  - Understanding how your brain works
  - Mental models for information security
  - How your brain function relates to information security
  - Structured Analytical Techniques
- Infosec Analysis
  - Triage and prioritization
  - OPSEC
  - Documenting and checking your work

Welcome to Day 4.



Course Roadmap

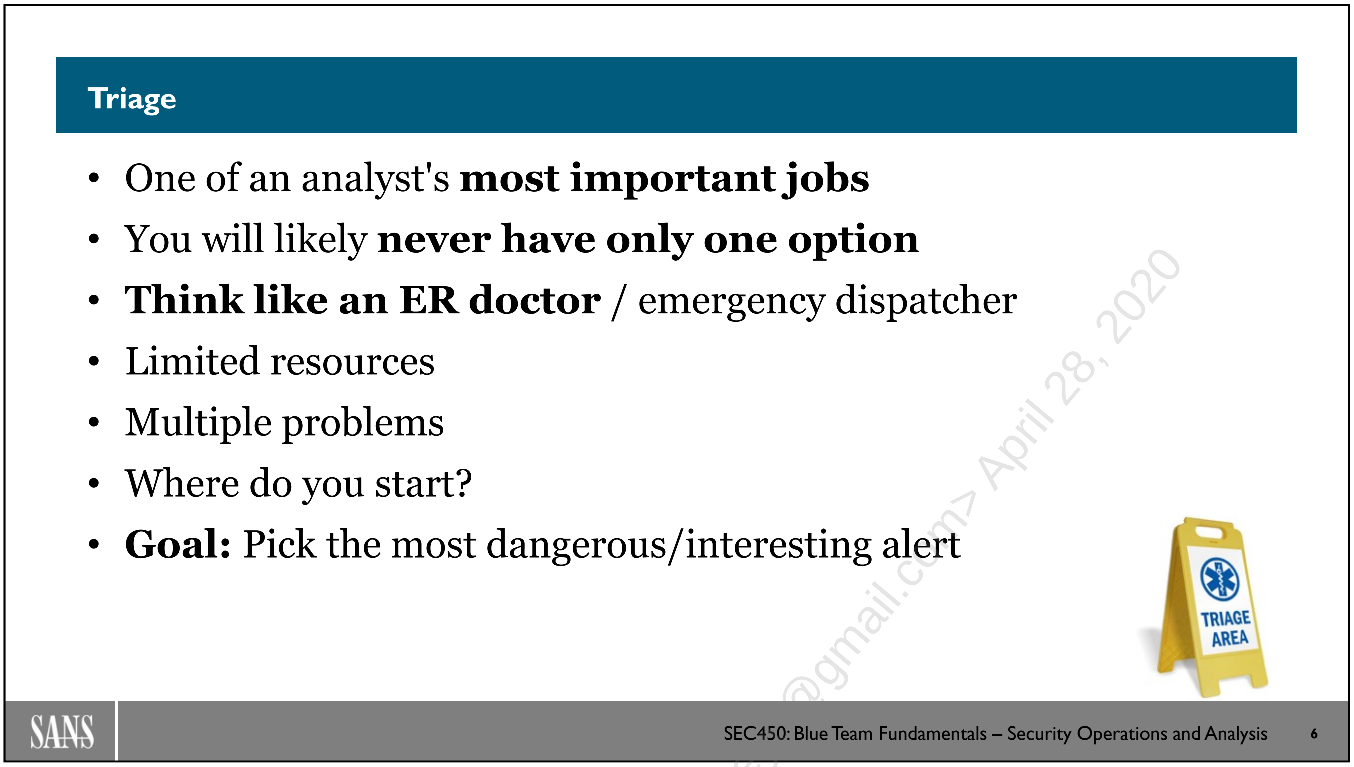
- Day 1: Blue Team Tools and Operations
- Day 2: Understanding Your Network
- Day 3: Understanding Hosts, Logs, and Files
- **Day 4: Triage and Analysis**
- Day 5: Continuous Improvement, Analytics, and Automation

Triage and Analysis

1. **Alert Triage and Prioritization**
2. Perception, Memory, and Investigation
3. Models and Concepts for Infosec
4. Exercise 4.1: Alert Triage and Prioritization
5. Structured Analytical Techniques
6. Analysis Questions and Tactics
7. Analysis OPSEC
8. Exercise 4.2: Structured Analysis Challenge
9. Intrusion Discovery
10. Incident Closing and Quality Review
11. Day 4 Summary
12. Exercise 4.3: Collecting and Documenting Incident Information

SANS

This page intentionally left blank.



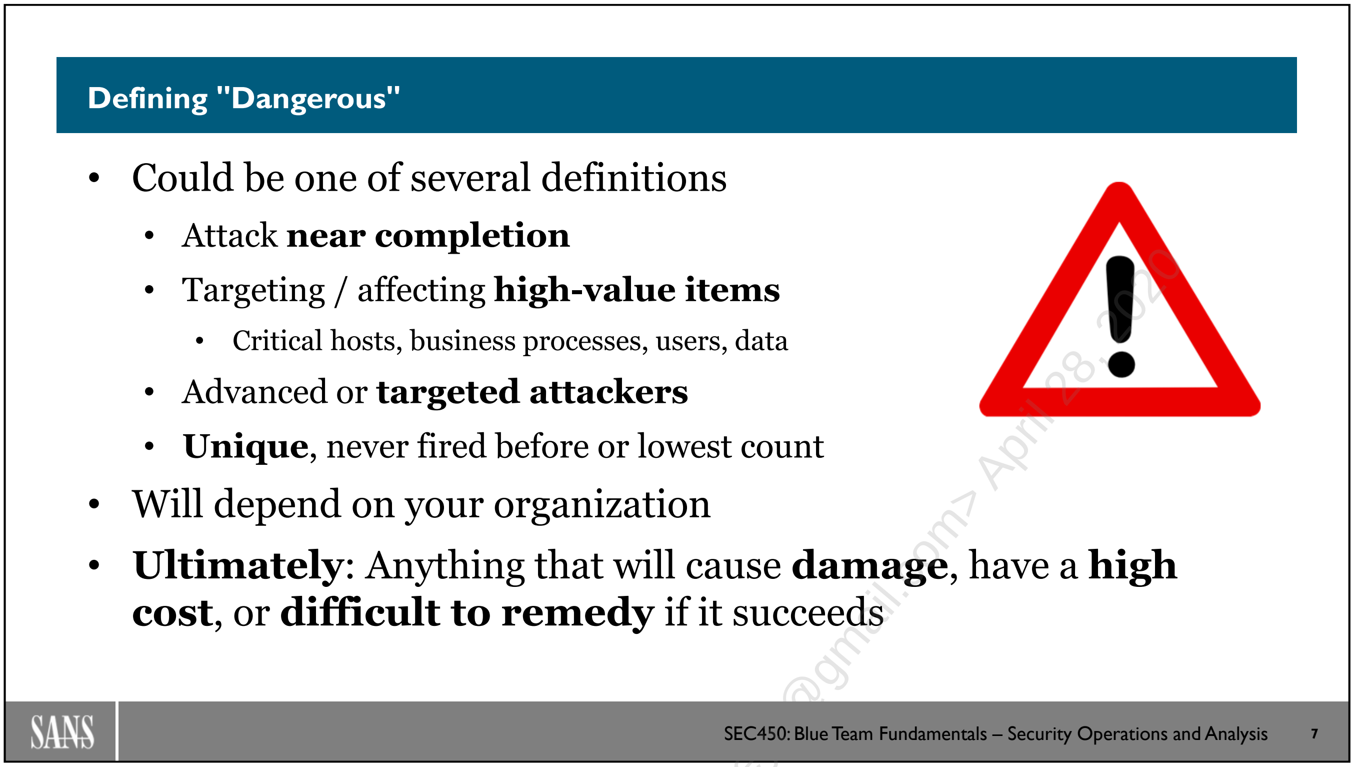
Triage

- One of an analyst's **most important jobs**
- You will likely **never have only one option**
- **Think like an ER doctor** / emergency dispatcher
- Limited resources
- Multiple problems
- Where do you start?
- **Goal:** Pick the most dangerous/interesting alert

TRIAGE
AREA

Triage

Alert triage is one of the analyst's primary jobs. It's not just picking the oldest alert off the top of the pile and jumping in, though; we need to be strategic about it. Think about if you went into the ER for baby delivery and the person who walked in two steps ahead of you with a broken toe was attended to before you, just because they walked in first. Obviously, that wouldn't make any sense. Different issues have different priorities, so the queuing system in the SOC must reflect that, and the analysts must do their best to assess the present situation when taking the next alert.



Defining "Dangerous"

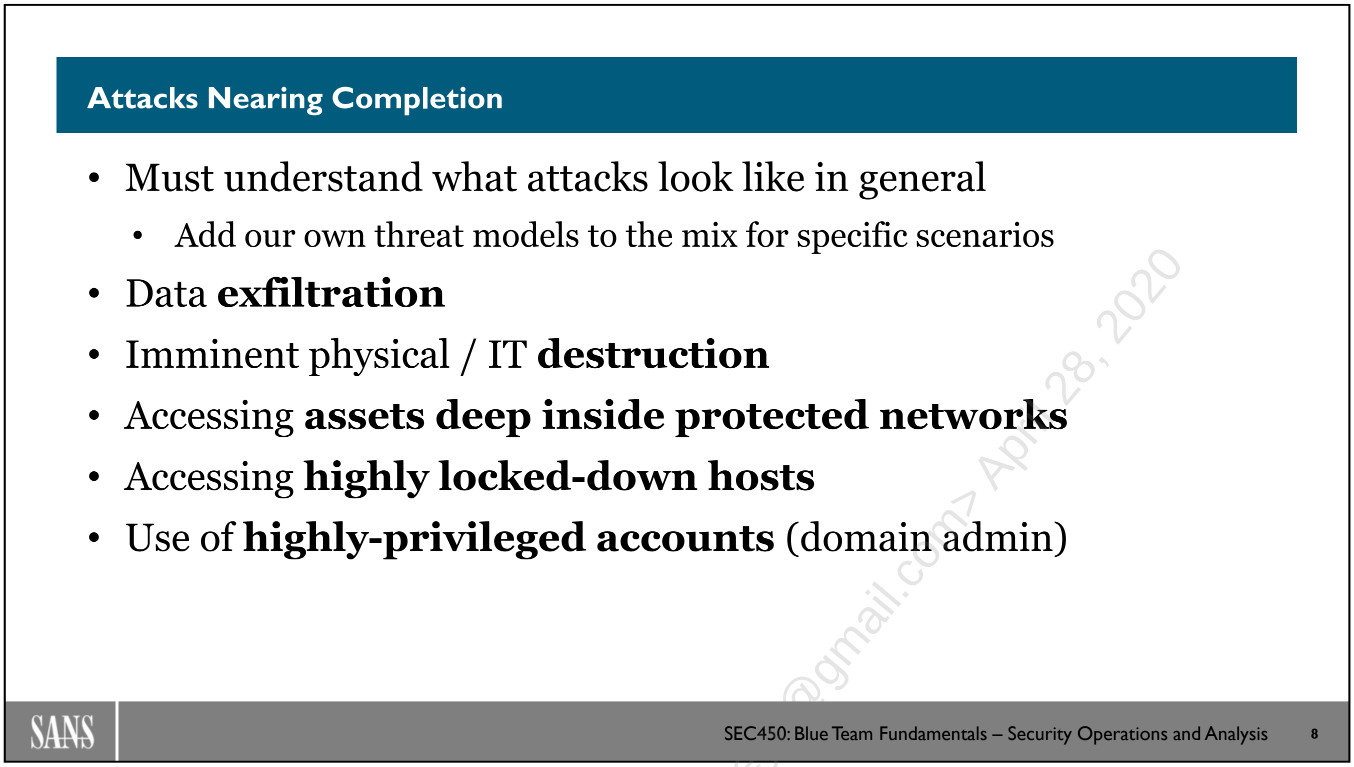
- Could be one of several definitions
  - Attack **near completion**
  - Targeting / affecting **high-value items**
    - Critical hosts, business processes, users, data
  - Advanced or **targeted attackers**
  - **Unique**, never fired before or lowest count
- Will depend on your organization
- **Ultimately**: Anything that will cause **damage**, have a **high cost**, or **difficult to remedy** if it succeeds

Defining "Dangerous"

How do we define dangerous in this sense? There are several ways this could be interpreted. First and foremost, any attack that is on the verge of succeeding and that would cause damage, be difficult to fix, or have an extremely high cost (which everything can be traced back to since time and damage == money) should absolutely be jumped on first.

If there is nothing of this variety, then any attack that appears to be on the track to meet this definition would be considered next. This would be anything that seems to be targeting critical assets, whether it's a host, user, business process, or data. For example, this could be a failed exploit attempt against a critical system, or a virus on a domain admin or users' laptop that has access to sensitive information. Since these attacks may be past the exploit stage but not quite near the completion of their goal, they would generally be next in line.

If there's nothing that meets these criteria, look for advanced or targeted attacks or unique alerts that may be a sign of them. Targeted attackers by definition want something from *you,* which means it's potentially the information that only you have—personal information, corporate intellectual property, engineering plans, or maybe they have plans to wipe everything in the environment like what was done at Sony. In any case, seemingly targeted attacks should always raise up in priority; the problem is identification. Targeted attacks will not be called "OMG APT in your environment, act now!" so we need another way to pick them out. An alternative way of surfacing interesting activity is looking at the bottom of the alerts for which event has fired the least frequently. This technique can lead you to unique alerts that may identify attacks in progress.

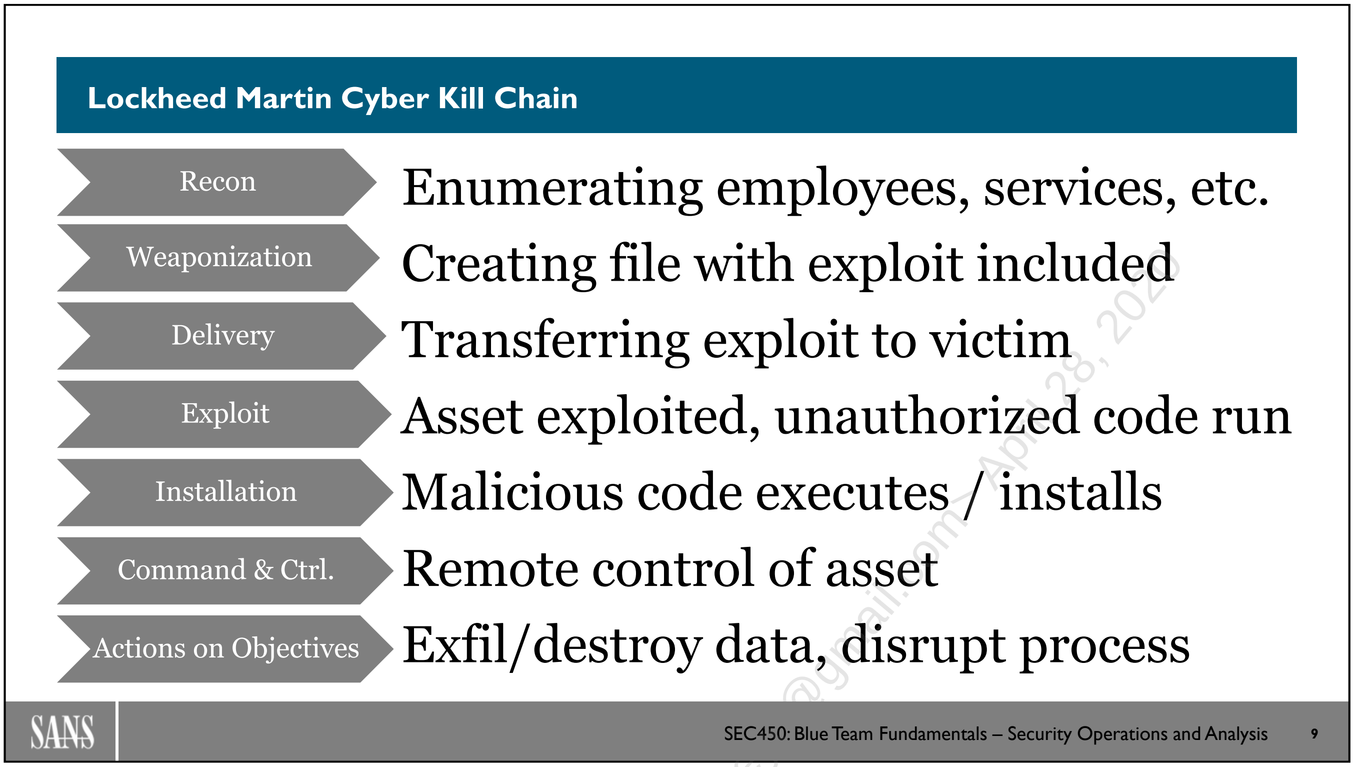


Attacks Nearing Completion

- Must understand what attacks look like in general
  - Add our own threat models to the mix for specific scenarios
- Data **exfiltration**
- Imminent physical / IT **destruction**
- Accessing **assets deep inside protected networks**
- Accessing **highly locked-down hosts**
- Use of **highly-privileged accounts** (domain admin)

Attacks Nearing Completion

To assess for an "attack nearing completion", we need to understand the stages of a typical attack, as well as consider specific scenarios our organization has in our threat model. Some of the typical scenarios that companies fear are data exfiltration, imminent physical danger or mass destruction of IT data. Beyond this, we can infer that if we are seeing malicious activity on hosts that are deep within protected enclaves in the network, have very few privileged accounts allowed to login, or see odd use of any highly privileged accounts where they should not be used, this can be a clue that the end is near!



Lockheed Martin Cyber Kill Chain

Recon	Enumerating employees, services, etc.
Weaponization	Creating file with exploit included
Delivery	Transferring exploit to victim
Exploit	Asset exploited, unauthorized code run
Installation	Malicious code executes / installs
Command & Ctrl.	Remote control of asset
Actions on Objectives	Exfil/destroy data, disrupt process

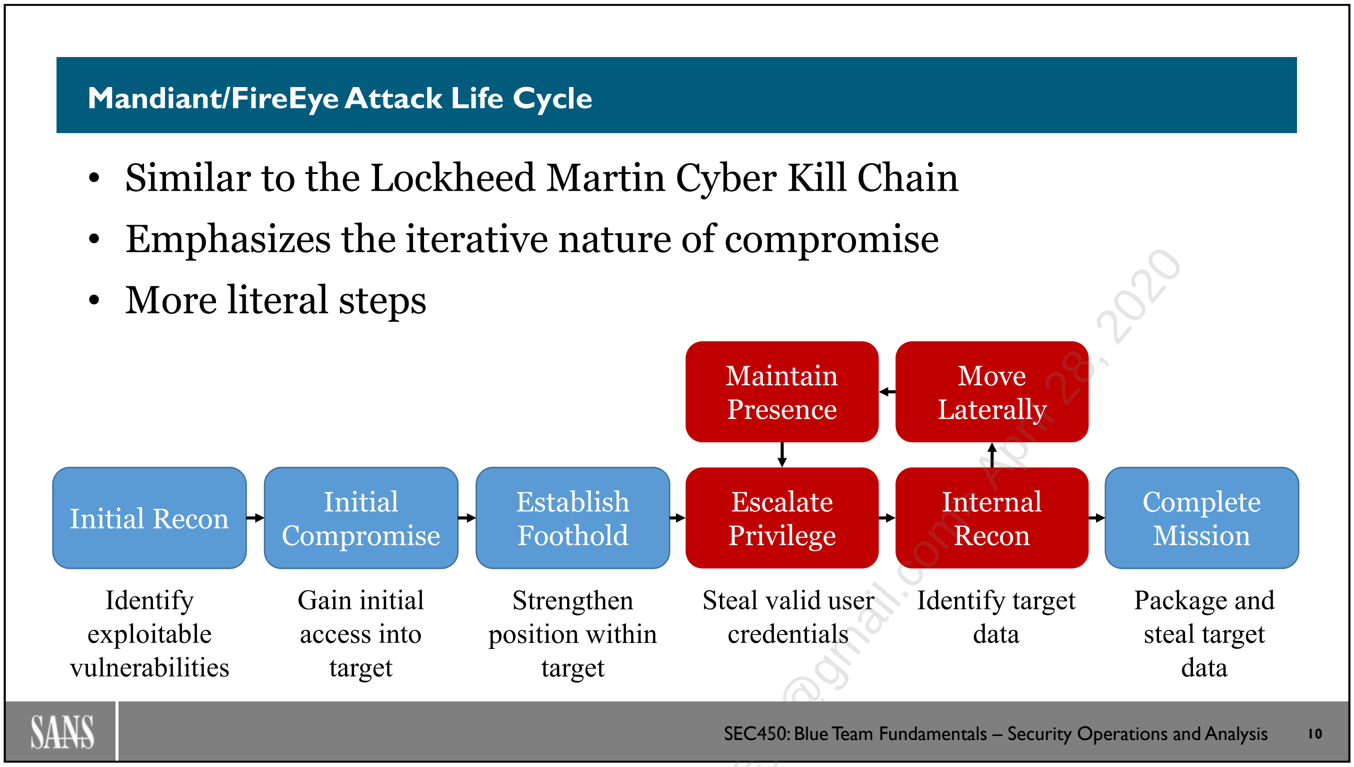
Lockheed Martin Cyber Kill Chain

Created by Eric M. Hutchins, Michael J. Cloppert, and Rohan M. Amin, Ph.D., the Lockheed Martin Cyber Kill Chain[1] is one mental model that almost everyone has heard of. It is a seven-step chart meant to show the phases of a *targeted* attacker or "APT" specifically. Its purpose is to show how to break down an attack into necessary stages, assist defenders with aligning attack indicators to "courses of action", and enable the collection of threat intelligence to link together activities from the same adversary over time. The description on this slide shows an example of what part of an attack might fall under each step of the Kill Chain. As for identifying when an attack is nearing completion, this model should constantly be in your head as a reference during triage of *any* alert. According to this chart, attacks that have had a successful exploit, some sort of code execution or malware install, and are receiving active command and control are potentially on the last step, and therefore command and control-based alerts should receive high priority as well as any direct evidence of the seventh step.

One useful thing to remember about the Cyber Kill Chain (and other attack modeling frameworks) is that some items are performed primarily on the attacker side vs. the victim side, and that other steps generally pertain to either host activity or network transfer of data. For example, the Exploitation and Installation steps are generally host based while the Recon, Delivery, and Command and Control steps are often network based and lean more heavily on network traffic data for identification. Other steps such as Recon and Weaponization are primarily done attacker side and are much harder to detect and act upon. It is also worth noting the weakness in the Kill Chain model—that it does not show the iterative nature of compromise and the fact that often multiple hosts will need to be compromised, leading to more of a "loop" in the later stages of the chain. This means if you see interactive command and control from a desktop, yes that's bad, but they likely will need to get interactive command and control from a higher importance host before they are truly at the "attack almost complete" level.

This loop is better pictured in the Mandiant/FireEye attack life cycle described in the next slide.

[1] https://lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf



Mandiant/FireEye Attack Life Cycle

- Similar to the Lockheed Martin Cyber Kill Chain
- Emphasizes the iterative nature of compromise
- More literal steps

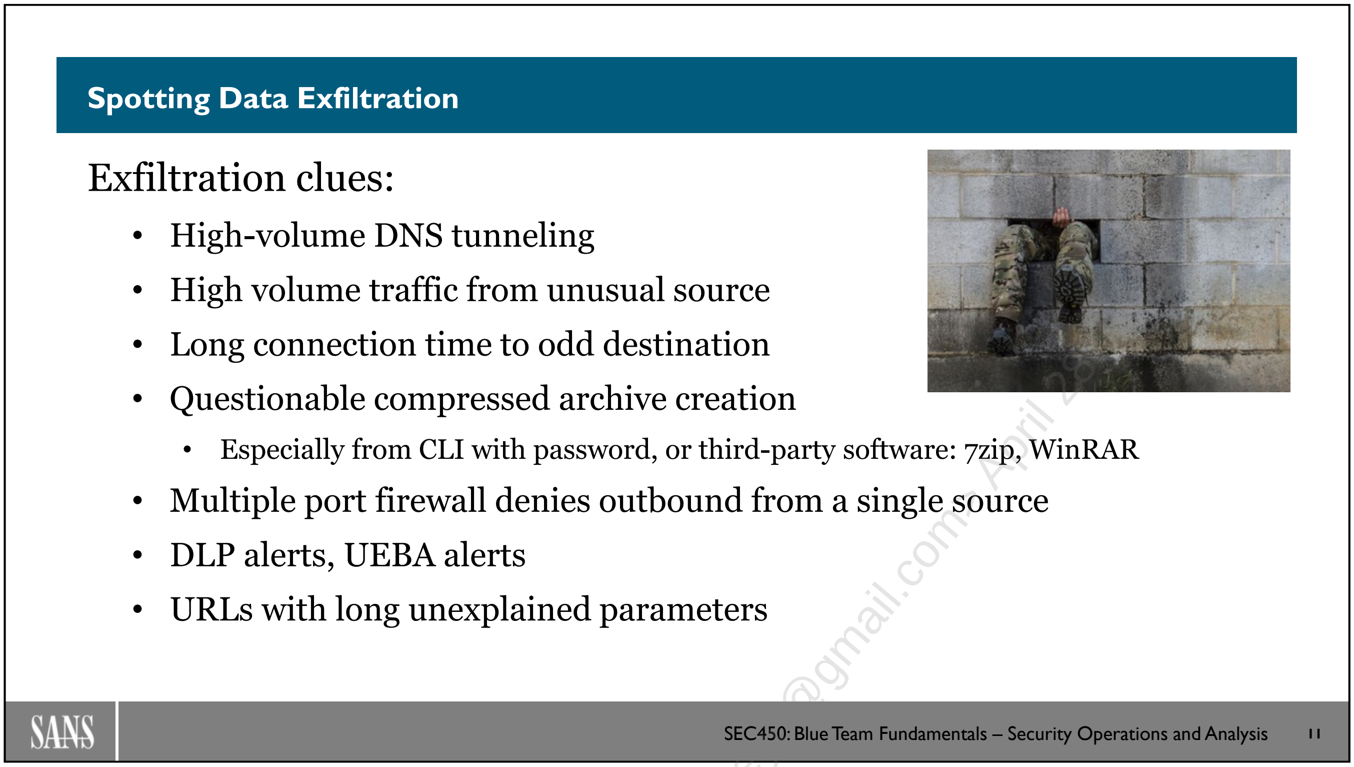
Initial Recon
Identify exploitable vulnerabilities
Initial Compromise
Gain initial access into target
Establish Foothold
Strengthen position within target
Escalate Privilege
Steal valid user credentials
Internal Recon
Identify target data
Move Laterally
Maintain Presence
Complete Mission
Package and steal target data

Mandiant/FireEye Attack Life Cycle

The Mandiant/FireEye Attack Life Cycle is very similar to the Lockheed Martin Cyber Kill Chain; however, it does a slightly better job at portraying the literal reality of many intrusions.[1] Like the Kill Chain, it shows the initial attack stages where recon occurs, but instead of weaponization, delivery, exploitation, and installation, it shows "initial compromise" and "establish foothold". This is a bit more generic compared to "install" from the kill chain and therefore leaves room for the fact that not all intrusions literally involve an install, usually just the ability to remote command a machine to do something and use it as a foothold to get to the rest of the environment. It is then specifically called out that the adversary will likely need to escalate privilege and use that privilege for internal recon to continue to move toward their goal in a lateral fashion. In order to maintain control of the compromised network, maintaining presence (or persistence) is also specifically called out and the cycle shows the circular nature of this activity. Once an attacker has finally escalated privileges and moved laterally enough times, they will have access to their target and be able to complete the mission, which is shown in the final step.

This model is more useful for those who are new to the patterns of attack and have not seen a cyber intrusion before since the steps are much more accurate yet generalized and portray the actual repetitive compromise that is necessary in almost all intrusions. This is also a useful model to keep in your head when triaging alerts, but due to its non-linear nature, isn't quite as easy to say "this has reached kill chain stage 6, therefore they're close to the goal" since there are multiple step 4-7 instances per intrusion, so interpretation may be needed.

[1] Recreated from: https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/ds-threatspace.pdf



Spotting Data Exfiltration

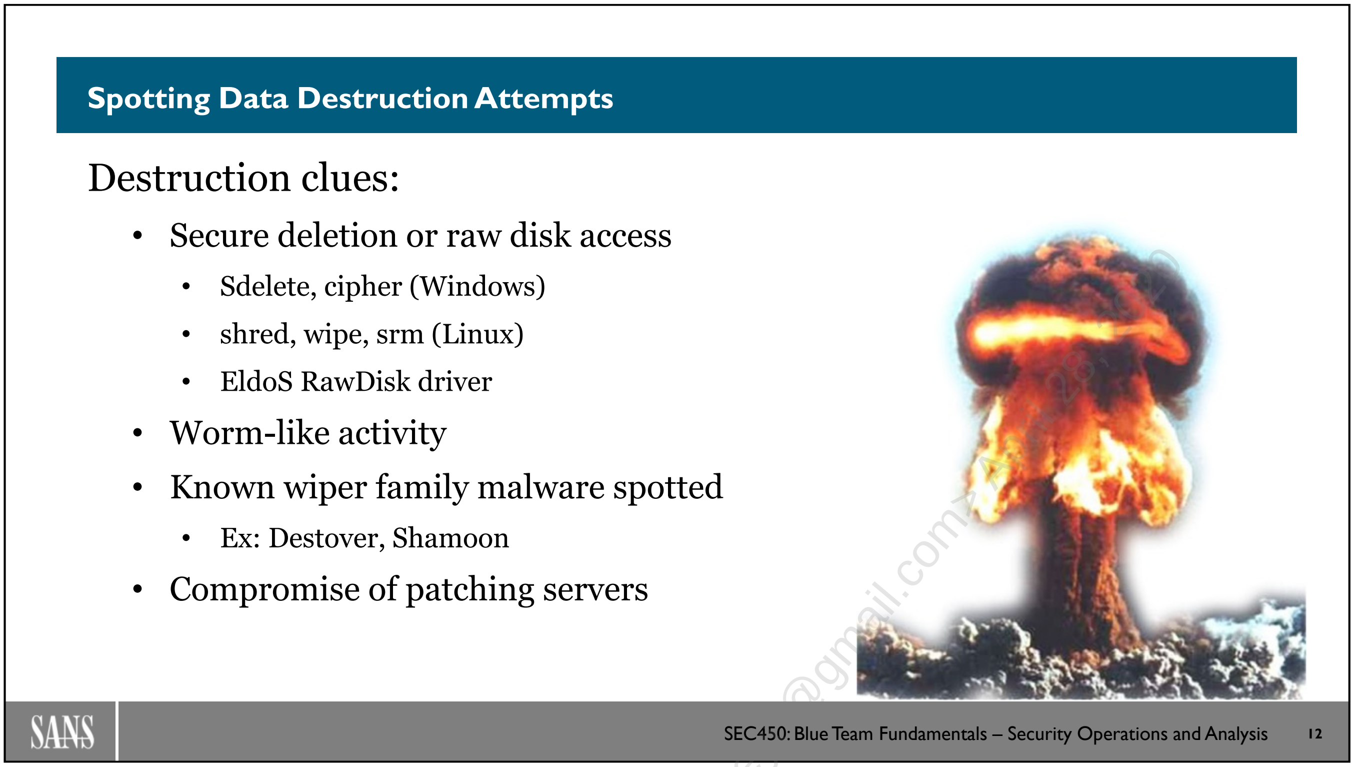
Exfiltration clues:

- High-volume DNS tunneling
- High volume traffic from unusual source
- Long connection time to odd destination
- Questionable compressed archive creation
  - Especially from CLI with password, or third-party software: 7zip, WinRAR
- Multiple port firewall denies outbound from a single source
- DLP alerts, UEBA alerts
- URLs with long unexplained parameters

Spotting Data Exfiltration

How do we spot exfiltration attempts? Put on your black hat for a moment and do some threat modeling. Consider the controls in your environment and the data collection capabilities and system visibility you have. If you were an attacker trying to exfiltrate data, what would it take to achieve? Since this is a late-stage compromise activity, there would be a whole slew of other activities required beforehand. You may either detect evidence of these or the data prep and exfil traffic itself.

To exfiltrate data, you must first access it, if you can access it you generally must stage it somewhere that can send it out and find a way out, prep it, and have the upload succeed. Any of these items can leave a mark in event logs from firewalls, DLP (data loss prevention), UEBA (user and entity behavior analysis), or NetFlow. If you see multiple alerts regarding one source inside your network that seems to be piling up alerts indicating sensitive information has been accessed and see command lines for the host zipping files up, it may be time to act! A large number of firewall failures on different ports from a single machine can be an attacker attempting to find a way out, and a large volume upload or long running connection from a single machine may also be indicative of exfiltration as well. With these situations, it's easier to try to pivot to PCAP or proxy logs to try to identify if the destination receiving the traffic is suspicious. Remember, Dropbox and other file sharing services make great exfil upload points since they may fly under the radar unless they are against policy. Usage of non-standard archive creating tools, especially using passwords, breaking the file into pieces, or just run from the command line can be another good detection (look for 7z or WinRAR executables with password flags). Most people will not use these options or run the tools from the command line, so this should be a low-volume detection. URLs with long unexplained parameters can also be a potential giveaway. If you see lots of connections with URL data that looks like base64 encoded parameters for example, like `/index/__utm.gif?cookie=lFLUISudlF098rgfoldkGOdsolkjgldkdf908fFOEP8p9jmf9w8f3a9=`, it may be a clue that GET parameters are being used to encode data and send it out.



Spotting Data Destruction Attempts

Destruction clues:

- Secure deletion or raw disk access
  - Sdelete, cipher (Windows)
  - shred, wipe, srm (Linux)
  - EldoS RawDisk driver
- Worm-like activity
- Known wiper family malware spotted
  - Ex: Destover, Shamoon
- Compromise of patching servers

SANS

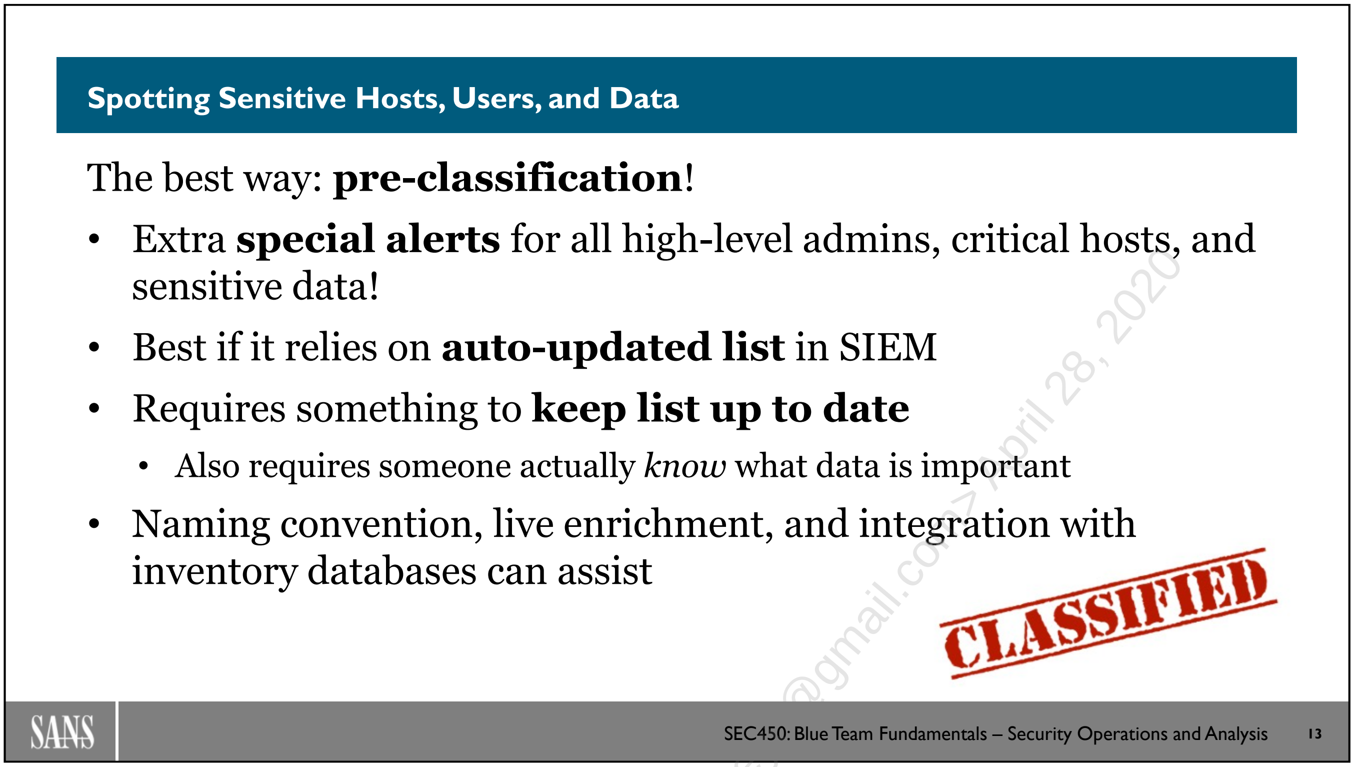
SEC450: Blue Team Fundamentals – Security Operations and Analysis 12

Spotting Data Destruction Attempts

Although spotting the signs of data destruction before it starts happening can be difficult, it is potentially possible in some cases. Your antivirus vendor should pick up on any worm-like malware activity, and whether it's succeeding. This detection combined with any detections for specific malware families known to destroy disks (such as Destover[1] or Shamoon[2]) should immediately ring the alarm. A different way to do it, although possibly after the process has started, is to trigger on the tools for secure data deletion. In the past, attackers have used tools like Sysinternals sdelete or the built-in cipher.exe program in Windows to securely delete files. In Linux, tools like shred, wipe, and srm should be alerted on. Since most people will not be running secure deletion utilities from the command line, false positives for alerts like this should stay to a minimum. Another third potential option is looking for the installation of drivers that may give attackers raw disk access and allow them to dodge NTFS file permissions. In breaches that used Destover and Shamoon, the RawDisk driver from EldoS was used to do just that, and the unexpected driver usage could have been detected by Windows auditing and used as a potential early warning sign.

[1] https://www.symantec.com/connect/blogs/destover-destructive-malware-has-links-attacks-south-korea

[2] https://www.symantec.com/blogs/threat-intelligence/shamoon-destructive-threat-re-emerges-new-sting-its-tail



Spotting Sensitive Hosts, Users, and Data

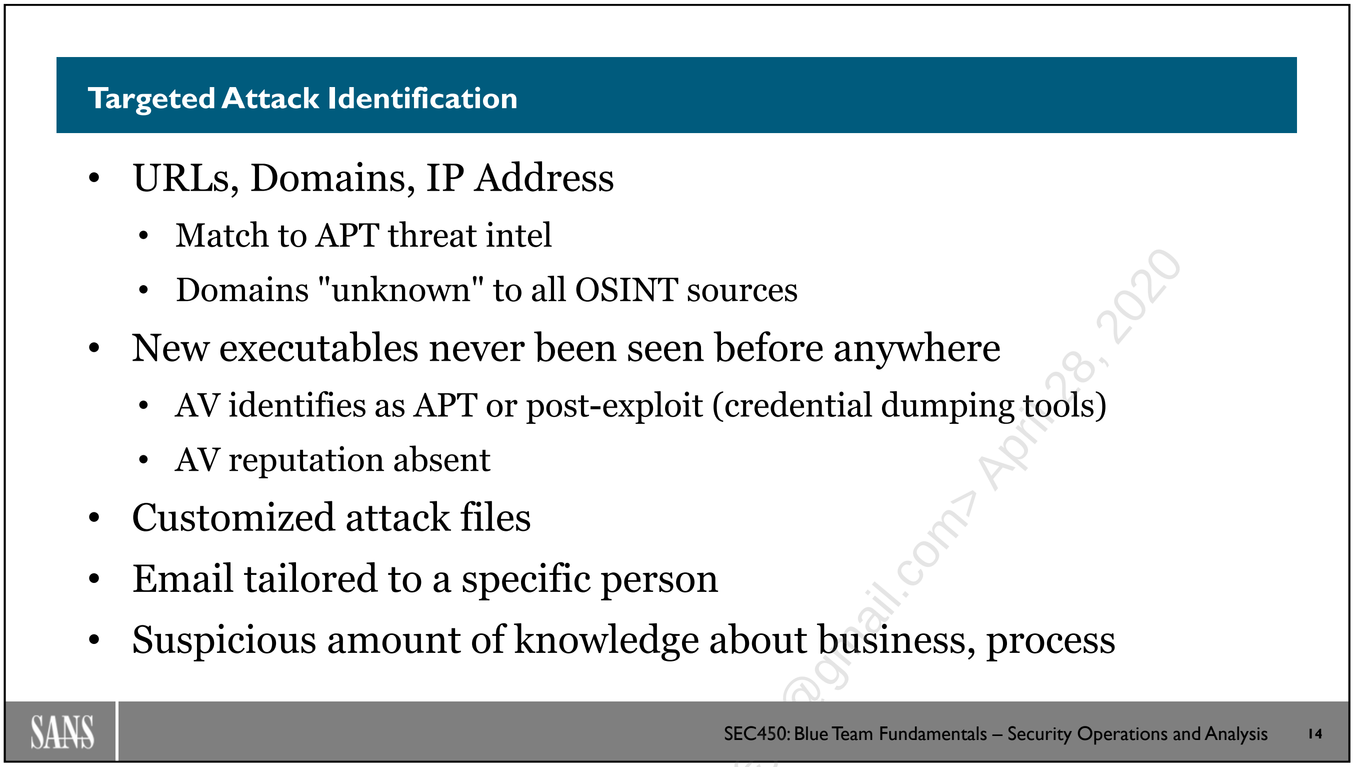
The best way: **pre-classification**!

- Extra **special alerts** for all high-level admins, critical hosts, and sensitive data!
- Best if it relies on **auto-updated list** in SIEM
- Requires something to **keep list up to date**
  - Also requires someone actually *know* what data is important
- Naming convention, live enrichment, and integration with inventory databases can assist

Spotting Sensitive Host, User, and Data Access

How do we know if an alert is telling us that critical users or hosts are being attacked, or if sensitive data is being accessed inappropriately? The best answer is that these entities should have been pre-classified as important and the alert name should give you this context – i.e. "Exploit X attempt on system" vs. Exploit X attempt on *critical* system". The alert itself should include the information that it was against a high-risk system, person, or data, and make that fact abundantly clear. How do we do this? Many SIEMs can tie into asset management and user identity systems. Any time an alert comes in, the criticality of that entity should ideally be looked up in the (hopefully auto-updating) list and modify its priority if it matches one of those systems. Of course, this requires your organization to know where its sensitive data sits and who can access it. That's a different problem that many struggle with.

If you cannot automatically include this data in the alert, automatic enrichments once the alert is selected in your alert aggregation tool should be performed. At a minimum, the manual lookup of the user, asset, and data should be the first step in triage. If nothing else, naming conventions of servers and job titles should be able to provide the context required to determine how risky a situation might be. Seeing "chief engineer" or "help desk" as the job title of a compromised victim should be enough to up the priority of that alert in the triage queue.



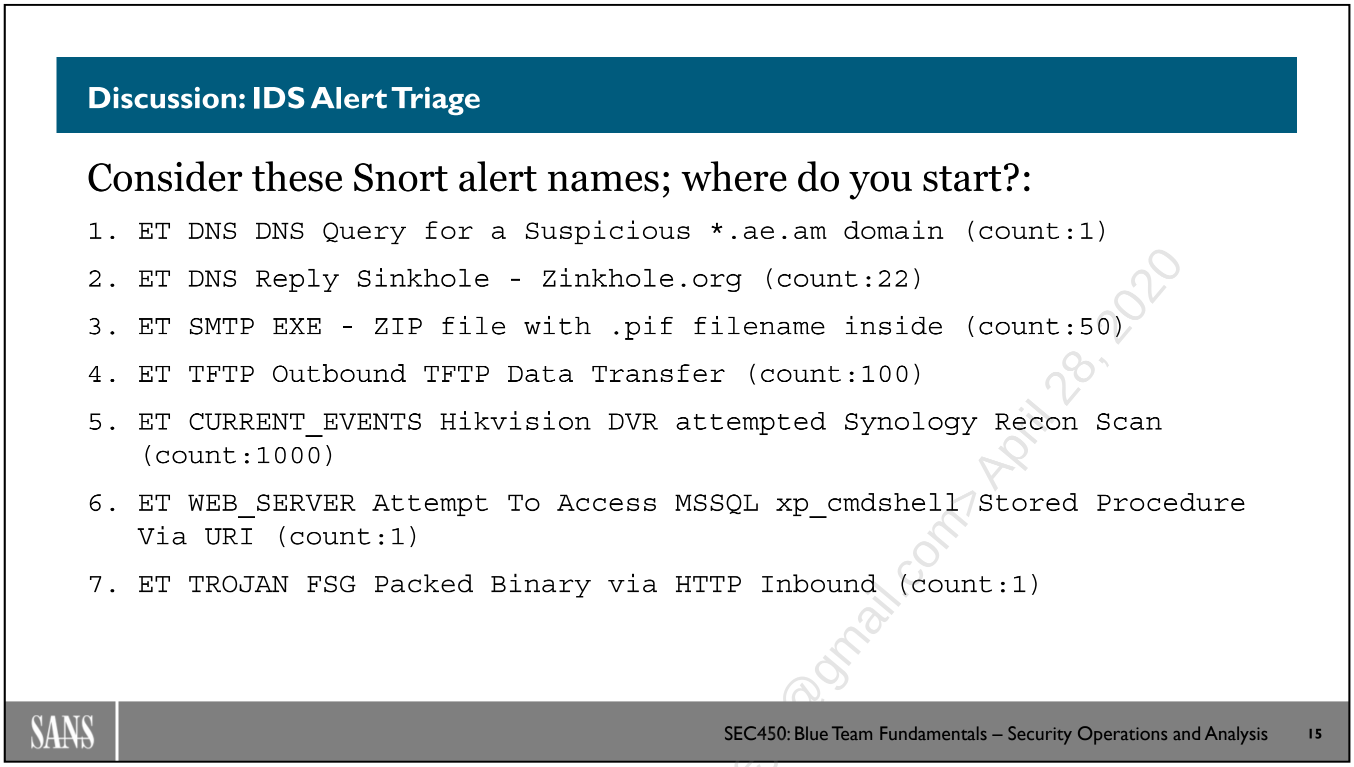
Targeted Attack Identification

- URLs, Domains, IP Address
  - Match to APT threat intel
  - Domains "unknown" to all OSINT sources
- New executables never been seen before anywhere
  - AV identifies as APT or post-exploit (credential dumping tools)
  - AV reputation absent
- Customized attack files
- Email tailored to a specific person
- Suspicious amount of knowledge about business, process

Targeted Attack Identification

What might be some tip-offs in alerts that could signal an attack is being carried out by a particularly dangerous attack group or is targeted toward us? While there are a lot of possible options depending on your situation, here are some of the most obvious red flags:

- URLS: If a URL is a known threat intel indicator match for an attack group, this is a dead giveaway, but what about those that are unknown? In most cases, attack URLs used in opportunistic, high-volume compromise are submitted to community tools like urlscan.io and VirusTotal rather quickly. If you look for references to domains in open source tools and find that they are totally absent, this is potentially concerning.
- Files: Like domains, if a file is identified and named part of a malware family known only to be associated with advanced attack groups, this is a reason to move quickly, but how do we decide on the other files? Again, same as domains, a lack of knowledge about a file is highly suspicious. Many AV suites will submit unknown samples to their cloud databases for analysis and will come back with a reputation score including how common the file is. As Rob Joyce, head of the NSA TAO team, said, "*Let me tell you: If you've got a reputation service and it says that interesting executable that you think you want to run, in the entire history of the Internet has been run one time, and it's on your machine, be afraid, be very afraid.*"
- Any time your organization's logo or other customizations have been made to match your company specifically, this means someone took at least minimal effort to tailor the attack to you. Any alerts with data like this should be looked at with higher suspicion.
- Emails addressed to a single individual, especially with seemingly higher than normal levels of knowledge than an outsider should have should be increased in priority. This implies the attacker has spent time doing research on your company operations or employees. If they're using internal terms and referencing things only employees should know about, consider it targeted unless proven otherwise.



Discussion: IDS Alert Triage

Consider these Snort alert names; where do you start?:

1. ET DNS DNS Query for a Suspicious *.ae.am domain (count:1)
2. ET DNS Reply Sinkhole - Zinkhole.org (count:22)
3. ET SMTP EXE - ZIP file with .pif filename inside (count:50)
4. ET TFTP Outbound TFTP Data Transfer (count:100)
5. ET CURRENT_EVENTS Hikvision DVR attempted Synology Recon Scan (count:1000)
6. ET WEB_SERVER Attempt To Access MSSQL xp_cmdshell Stored Procedure Via URI (count:1)
7. ET TROJAN FSG Packed Binary via HTTP Inbound (count:1)

Discussion: IDS Alert Triage

Let's take a crack at looking at an alert list from Snort and see if we can decide where to start based on the principles we just discussed. Of course, in the real situation you'd have the full Snort alert data, but for this exercise, we'll just work the titles and judge them based on their apparent attack stage.

#1 – This alert means someone tried to go to a potentially malicious website. The question is, was it because they were infected or because they were redirected there in an exploit attempt. Furthermore, we don't know if they reached the site or if it was active. Since there was only one alert registered for this title, we can infer it may have just been a redirect, meaning this could be a deliver/exploit stage, and an unsuccessful one at that. This would likely not jump to the front of the pile.

#2 – 22 alerts for a machine trying to contact a sinkhole domain….hmm. This is similar to the first alert except there are two key differences, the site is now not malicious (since it's a known sinkhole) and the fact there were 22 alerts. A non-existent site makes it better, but the 22 alerts make it worse because we know the device is likely infected since it tried multiple times to reach the domain. For all we know, this is an active infection that may switch to another C2 domain soon, so this is a high priority alert.

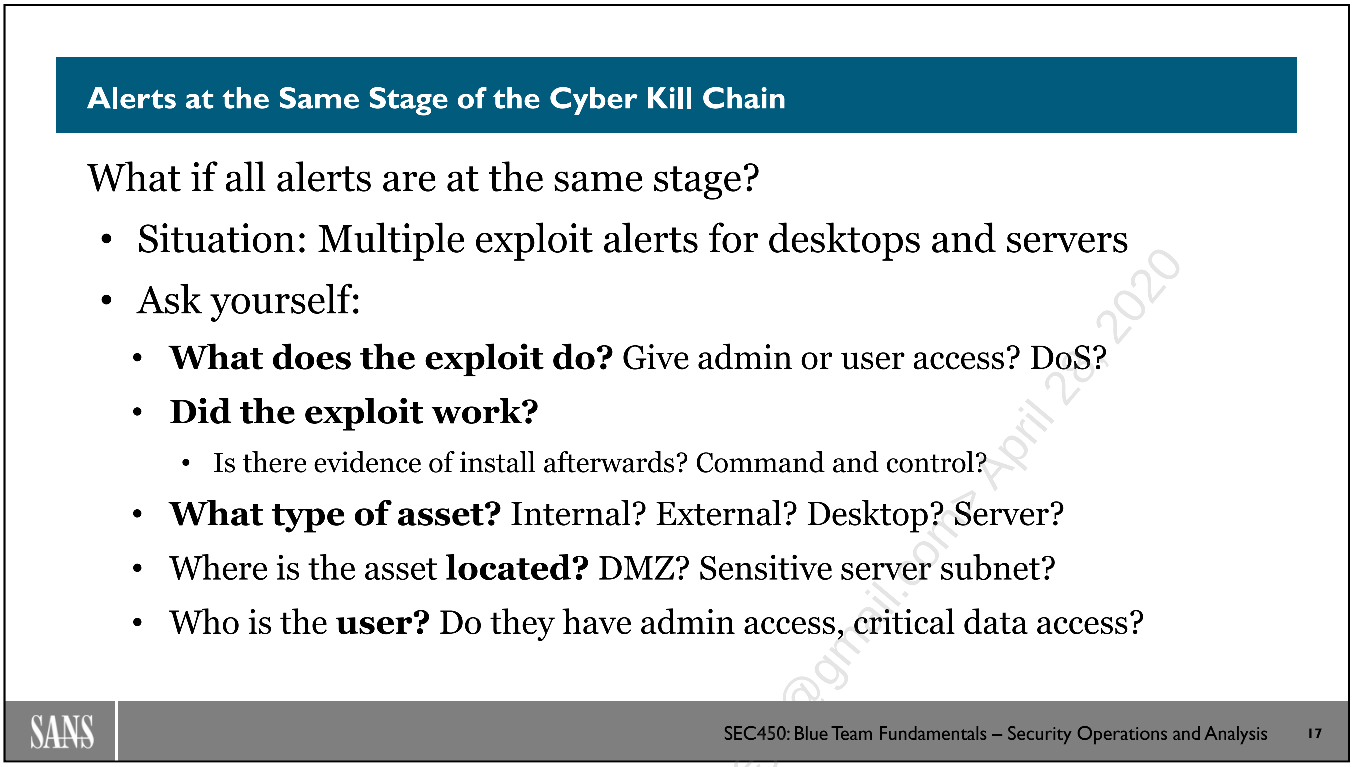
#3 – SMTP with an executable inbound represents a delivery stage attack. It's not the worst-case scenario but it did go to 50 people, so it's just a matter of time before 50 people could become infected if it was not blocked by another tool. This alert may be cleared quickly by checking the delivery status of the email and deleting it from inboxes. If possible, it would be best to analyze the malware and see what it talked to so you could search the SIEM for the site to see if anyone clicked to it before you got to it. This is a medium priority alert. It's early stage but affects multiple people.

#4 – 100 alerts for TFTP outbound traffic. First, do you use TFTP for sending traffic outbound from the organization? In all likelihood you don't, and the count being 100 means this potentially has been happening for a while. What is being sent out? It could be confidential information. It's probably best to check this one first since it could be active exfiltration!

#5 – This alert references the name of a DVR and that it was a recon scan. Since you probably aren't running that DVR in your company and since it's a scan, that means this is a first stage attack that is going to fail, likely putting it squarely at the bottom of the list.

#6 – Let's pretend for the sake of this exercise that this alert was generated for an *internal* web server. If that was the case, this alert would probably be equal in priority to the TFTP exfil. Why? Because someone is trying to laterally move to a server and run commands. The xp_cmdshell command is a built-in feature in MSSQL that allows users from a SQL command line to run operating system commands, if the user they have control of is a privileged one. While it's of course *possible* that this is an admin doing something out of the ordinary, there's also high potential that an attacker in control of your internal SQL server, which likely holds sensitive data. This alert should be attended to immediately.

#7 – This alert is telling us that Snort detected an executable file packaged with "FSG" in an HTTP download. If you aren't familiar with what a packer is, it's a way of making an executable act like a self-extracting zip archive while still having it remain executable. The benefit of doing this for attackers is that the packer (in this case detected as the "FSG" packer), also obfuscates the executable, making it harder to read strings and learn information about it. This is likely an installation stage attack, which may quickly turn into command and control if AV does not catch the program. This alert should probably be triaged in line with the priority of #2 since they both will likely involve an active infection.



Alerts at the Same Stage of the Cyber Kill Chain

What if all alerts are at the same stage?

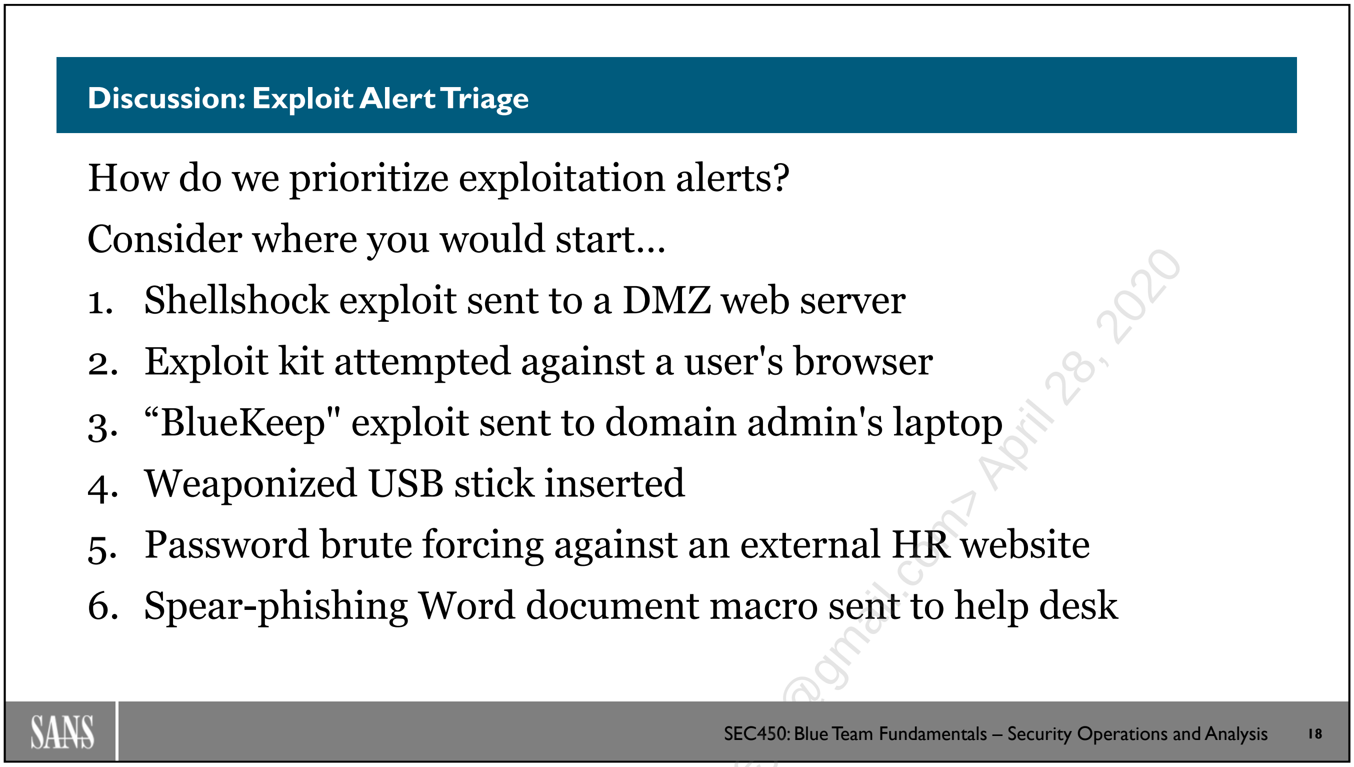
- Situation: Multiple exploit alerts for desktops and servers
- Ask yourself:
  - **What does the exploit do?** Give admin or user access? DoS?
  - **Did the exploit work?**
    - Is there evidence of install afterwards? Command and control?
  - **What type of asset?** Internal? External? Desktop? Server?
  - Where is the asset **located?** DMZ? Sensitive server subnet?
  - Who is the **user?** Do they have admin access, critical data access?

SEC450: Blue Team Fundamentals – Security Operations and Analysis 17

Alerts at the Same Stage of the Cyber Kill Chain

What if you are looking at a bunch of alerts for the same kill-chain stage, such as exploit attempts for various hosts, where do you start then? Again, consider the risk level involved with the successful compromise of each system and move from there. Here are some questions you can ask yourself:

- What does that exploit do if it works? If you are familiar with the exploit and know it's a bad one, like ETERNALBLUE, which if successful gives remote admin access, this can be a reason to up the priority of that alert.
- Did the exploit work? If you see an ETERNALBLUE exploit but know the system has been patched against that exploit, you can then slightly lower the priority knowing that although someone seemingly dangerous is attacking the host, at least they haven't succeeded…yet. It's still a potentially volatile situation though. What if you don't know the patch status of the machine? A look through the traffic for that system may give you an idea if something suspicious happened directly after the exploit attempt.
- What type of asset is it? If the exploit is being launched against an internal server, that's likely the worst-case scenario because it means someone has already penetrated the perimeter and is going after high value data. Alternatively, an exploit on an employee laptop with no sensitive data access can be de-prioritized, as can an old exploit attempt against a server in the DMZ, since the internet is full of exploit attempt background noise.
- Where is the asset located? As previously mentioned, an attack on an internal server should be prioritized over an external one since it implies attackers have already breached the perimeter. But even beyond that, there are likely multiple subnets that can be ranked in importance. Something against your test network may be less of a rush than something against a production server or manufacturing system.
- Who is the user being attacked? Exploits against anyone with admin access or sensitive data knowledge (engineers, scientists, accounting, c-suite, etc.) should be attended to first.



Discussion: Exploit Alert Triage

How do we prioritize exploitation alerts?

Consider where you would start…

1. Shellshock exploit sent to a DMZ web server
2. Exploit kit attempted against a user's browser
3. “BlueKeep" exploit sent to domain admin's laptop
4. Weaponized USB stick inserted
5. Password brute forcing against an external HR website
6. Spear-phishing Word document macro sent to help desk

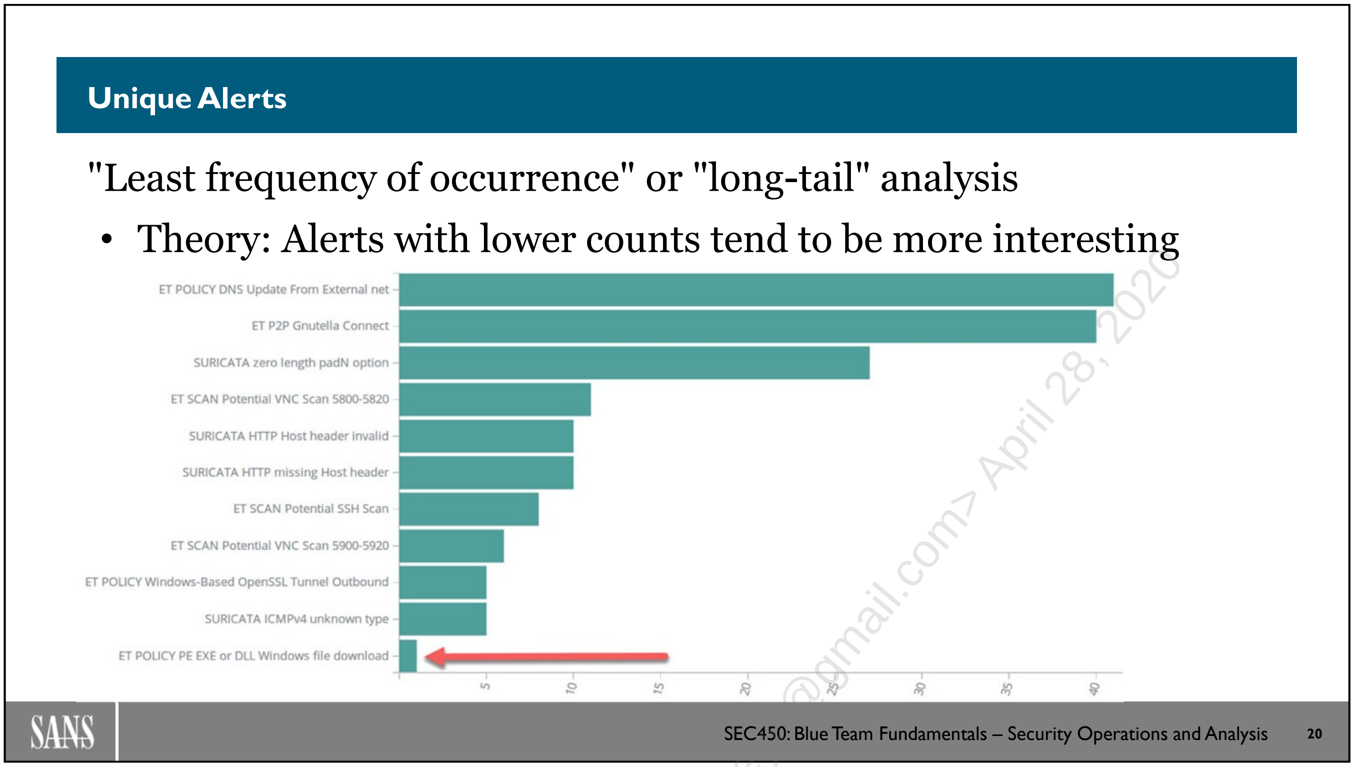
Discussion: Exploit Alert Triage

Consider the six scenarios above. Given only the information listed here, can we decide on which issue would escalate the fastest to an extremely damaging level? Although the situations are underdefined in some cases, we can make a general guess on how each would play out:

1. Shellshock exploit sent to a DMZ web server: The deep technical details of the shellshock exploit are not important to this conversation. What is important, however, is that Shellshock as an exploit allows the attacker to run code as the same user the web server process is running as. If this were a properly configured web server running as a limited user, since it is in the DMZ, this may cause very little damage. In the common occurrence that the server was set up to run as root, however, this attack would allow the external attacker to be root on that server, and potentially facilitate easy privilege escalation beyond the individual server. (See the details on the Equifax breach for an example of this exact scenario).
2. Exploit kit delivered to a user's browser: An exploit kit typically exploits the browser and ends with limited permission due to the browser being run in restricted mode, or perhaps with the malware running as the user who started the browser. In rare occasions, multiple exploits can be delivered together that will make this worse, but generally users who are browsing the internet should not be administrators or start the browser as admin. Therefore, this attack should leave the attacker with the same level of privilege as the user has themselves. In this list, the effects of this attack are relatively low.
3. BlueKeep is a remote, service-side RDP protocol exploit that allows the attacker to become the administrator account on any Windows machine it works against. In this case, since the attack is defined as being launched against a domain administrators' laptop, the attacker would also effectively be able to become domain administrator! This is one of the worst attacks in the list. With this level of access, damage could immediately be done to the enterprise!
4. Weaponized USB sticks are hard to predict. This situation is intentionally vague as the damage that would be caused if it were to succeed depends entirely on the user that inserted the USB drive. It could

be anyone that works for the organization but based purely on the number of employees who are administrators vs. the number who are not. It is likely the attack would not immediately yield damaging privileges.

5. This one is a bit of a trick as it is not an exploit. Password guessing against an external HR portal could end in several ways. Either the attack will get in or they will not; however, this is not an attack that would give the attacker command line access or the ability to run malware in any direct way. Therefore, this attack is least likely to be immediately dangerous in terms of help toward a path of privilege escalation. Once the attacker is in, however, the attacker *could* use the information gained inside the portal to take the next step, such as customized spear-phishing or even bank fraud.
6. Spear-phishing is very similar to the web-based exploit kit. If the user is not running as administrator, then opening a weaponized word document is unlikely to wield the attacker any more permission than the user has themselves. The Microsoft Word process is started by the user; therefore, anything the exploit does will likely end up with the same permissions.



Unique Alerts

"Least frequency of occurrence" or "long-tail" analysis

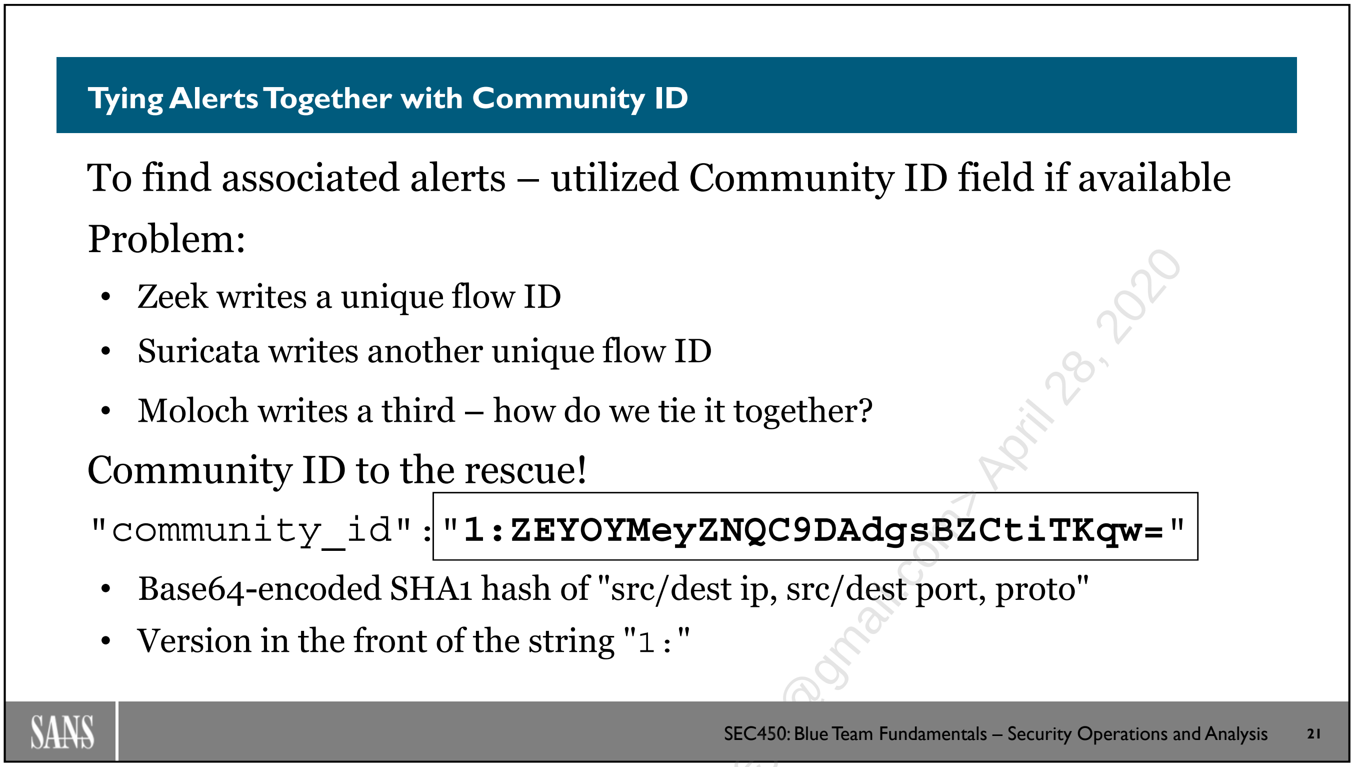
- Theory: Alerts with lower counts tend to be more interesting

Unique Alerts

Another way of deciding where to start triaging is the least frequent alert or the one with the lowest count of hits. Why? Although it doesn't always work perfectly depending on what causes an alert to fire (every packet in a connection vs. the whole connection), the things that tend to happen the most often should in theory be less interesting. If something is happening extremely frequently, the chance of that thing being bad is relatively low compared to the unique alert you have never seen before in your environment. If an APT starts performing activities that have never been done on the network before, this *should* set off alerts that have never been seen. Things that are at the top of the list are more likely to be rules that need to be tuned. This mode of thinking is often called "least frequency of occurrence" or "long-tail" analysis. It is an anomaly focused technique that works on the fact that in most environments there's a Pareto principle like distribution of alerts, where a large majority of what fires represent a small amount of the actual signatures.[1]

In the example above the bottom of the list is "ET POLICY PE EXE or DLL Windows file download". Although this is labeled as a policy violation, if you have a policy that defines people shouldn't be downloading executables and you investigate this and find the executable was being sourced from a poor reputation domain, this could lead you directly to an infection. At the top of the list, we have "ET POLICY DNS Update from External Net", a signature that a Google search indicates is a common false positive for many people and is caused when your DNS server is not properly defined in setup—a candidate for tuning.

[1] https://en.wikipedia.org/wiki/Pareto_principle



Tying Alerts Together with Community ID

To find associated alerts – utilized Community ID field if available

Problem:

- Zeek writes a unique flow ID
- Suricata writes another unique flow ID
- Moloch writes a third – how do we tie it together?

Community ID to the rescue!

`"community_id":` **`"1:ZEYOYMeyZNQC9DAdgsBZCtiTKqw="`**

- Base64-encoded SHA1 hash of "src/dest ip, src/dest port, proto"
- Version in the front of the string `"1:"`

SANS

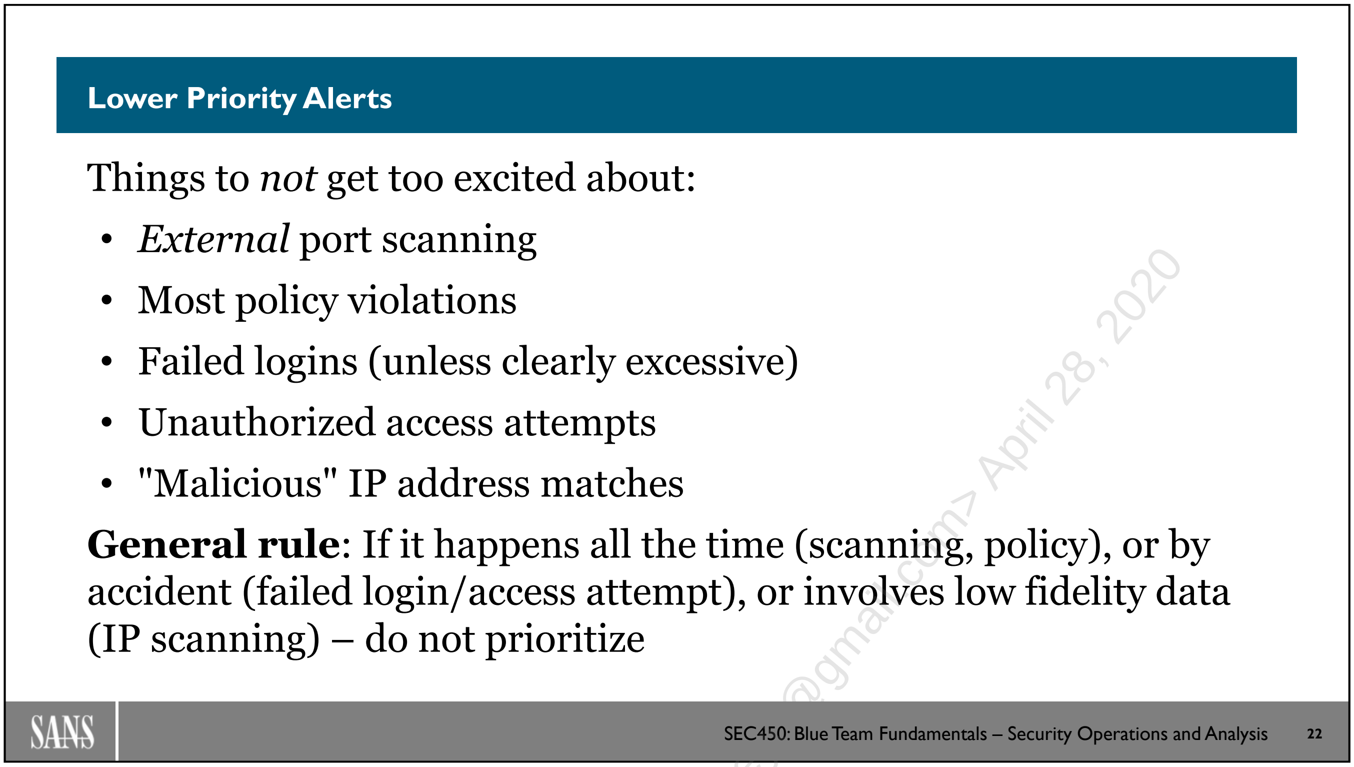
SEC450: Blue Team Fundamentals – Security Operations and Analysis 21

Tying Alerts Together with Community ID

One of the new fields you may see output from your devices that can help with alert investigation and correlation is the "Community ID" string. Described in Christian Kreibich's Suricon 2018 presentation[1], this field is an attempt to standardize how unique NetFlow flow IDs are written across tools, helping analysts tie together information from their PCAP tools with metadata written from NSM software like Suricata or Zeek.

The slide above shows an example of a community ID from a Zeek log. This pseudorandom number is generated by taking the source and destination IP, source and destination port, and protocol, and putting it through a SHA1 hash algorithm followed by base64 encoding the hash (to make the string shorter and easier to read). Although new, this standard has already been implemented in Suricata, Zeek, and Moloch, and is likely to catch on with more tools as time goes on. When faced with three different alerts from different tools and trying to decide if they are the same activity or not, the community string is one way that similar activity can be aggregated.

[1] https://suricon.net/wp-content/uploads/2019/01/SuriCon2018_Kreibich.pdf



Lower Priority Alerts

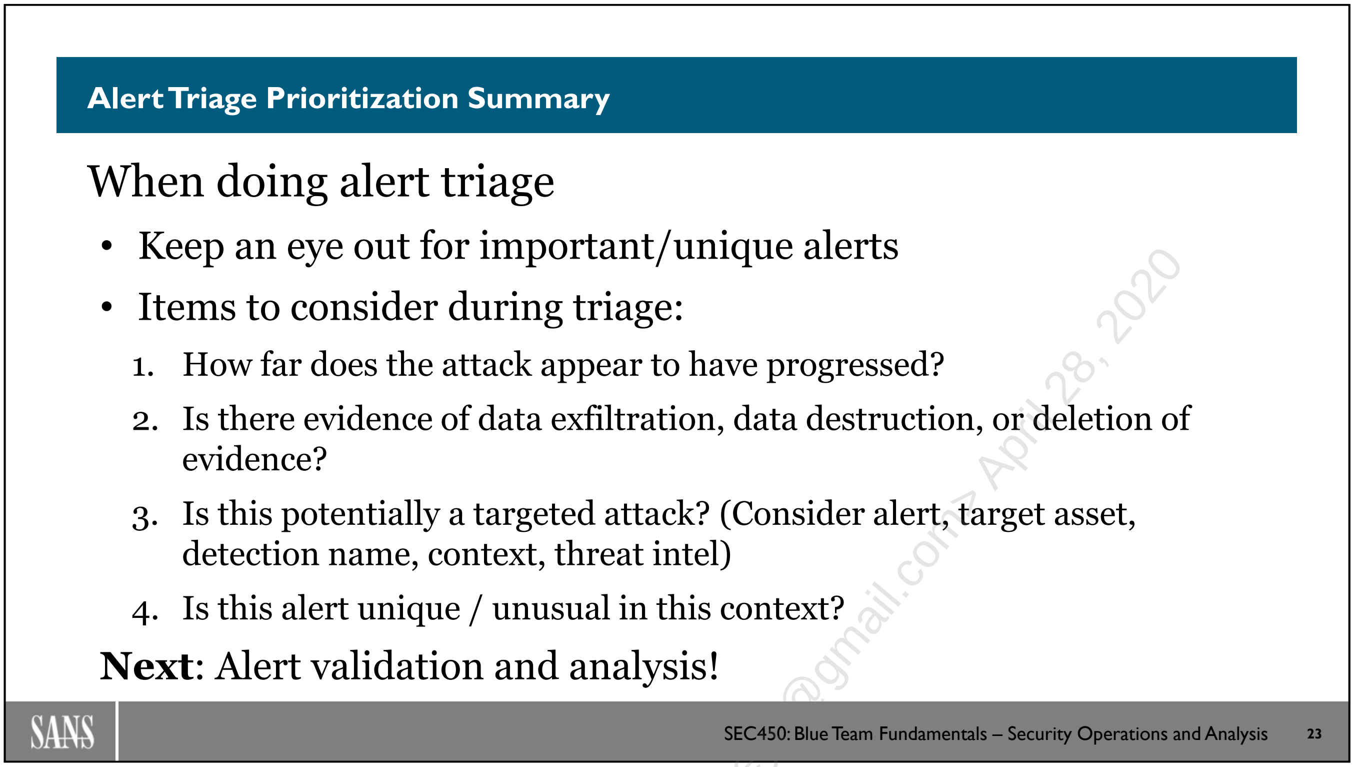
Things to *not* get too excited about:

- *External* port scanning
- Most policy violations
- Failed logins (unless clearly excessive)
- Unauthorized access attempts
- "Malicious" IP address matches

General rule: If it happens all the time (scanning, policy), or by accident (failed login/access attempt), or involves low fidelity data (IP scanning) – do not prioritize

Lower Priority Alerts

In almost every environment, there are events that happen on a continuous basis as background noise or accidents that may set off low priority alerts. Things like low counts of failed logins, unauthorized access attempts, external port scanning, and policy violations. These alerts typically are low fidelity since they are single conditions of something that theoretically *could* be bad, but also could be accidents. Users often type in their password incorrectly multiple times, the internet is always scanning you, and people violate policy by accident often. Since, at these low numbers, it's hard to differentiate advanced attacker activity from a simple mistake, most alerts of this type cannot be meaningfully acted on without significant investigation that will turn up nothing 99.99% of the time. When an alert is clearly excessive and has a destination of a single server or service, a misconfiguration should be ruled out before a "brute-force" attack is assumed—this kind of mistake is a common occurrence.



Alert Triage Prioritization Summary

When doing alert triage

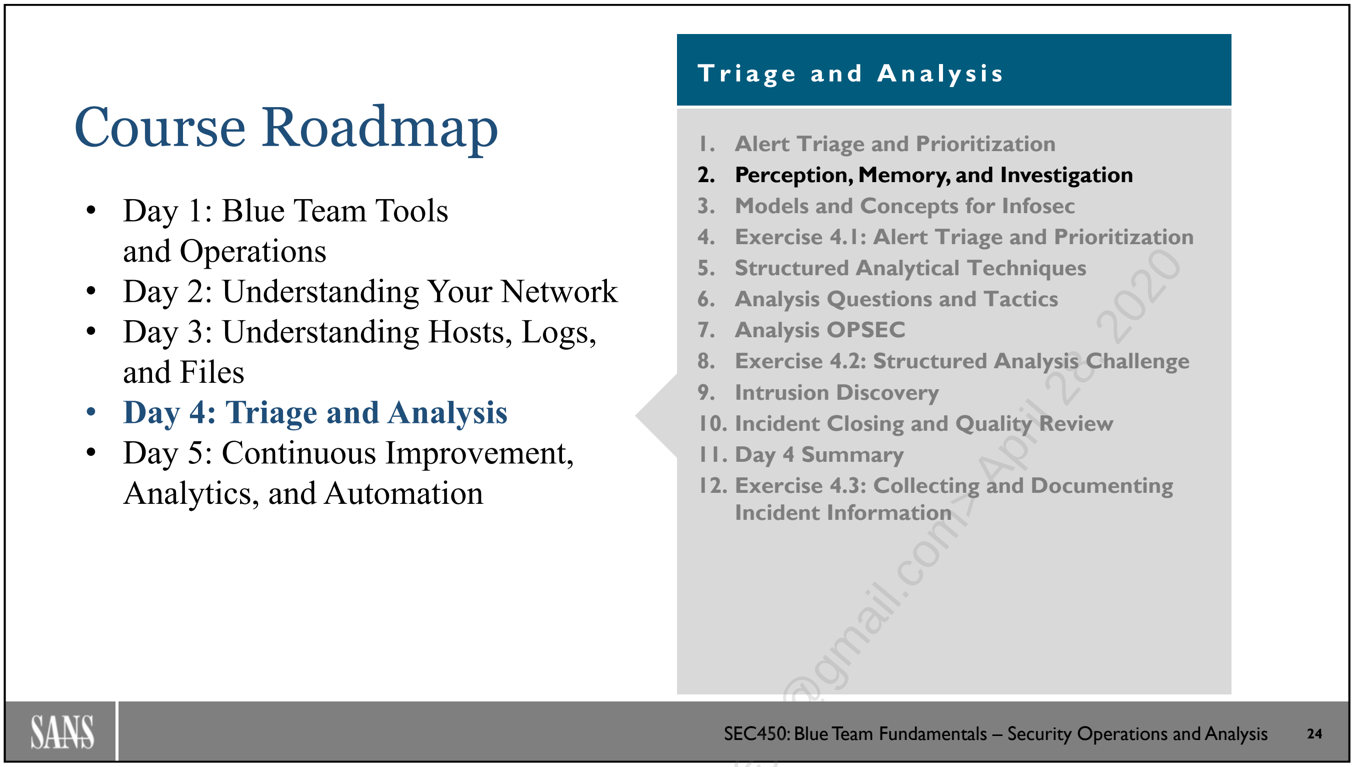
- Keep an eye out for important/unique alerts
- Items to consider during triage:
  1. How far does the attack appear to have progressed?
  2. Is there evidence of data exfiltration, data destruction, or deletion of evidence?
  3. Is this potentially a targeted attack? (Consider alert, target asset, detection name, context, threat intel)
  4. Is this alert unique / unusual in this context?

Next: Alert validation and analysis!

Alert Triage and Prioritization Summary

While there is no rule that can be applied 100% of the time for alert selection and verification, there are guidelines we can use, and continued use and familiarity with your toolset and environment will certainly improve your intuition over time. When choosing an alert, always try to go for the more pressing situation as defined either by how far the attack has progressed or the data/users it is after. Post-exploitation activity is always the most important, especially data exfiltration or destruction, or the preparation to do so. In addition, any alert that indicates either directly or indirectly that it may be associated with a targeted attacker should jump the line to the front of the queue since they present a particularly dangerous adversary. While these situations are not always obvious, threat intelligence can often be leveraged to clarify whether malware is related to targeted attackers or not. Unique alerts can also be an interesting place to start. Displaying all the alerts you have seen over a long period of time and paying attention to the items at the bottom may reveal attacks in your environment you may otherwise have not paid as much attention to.

Once you have selected the most interesting or dangerous looking alert, it's time to move on to the next step. In the next section we'll discuss the theory of alert validation. While this feels like a straightforward step, there are some pitfalls to be aware of, especially given the typical operating environment of a SOC.



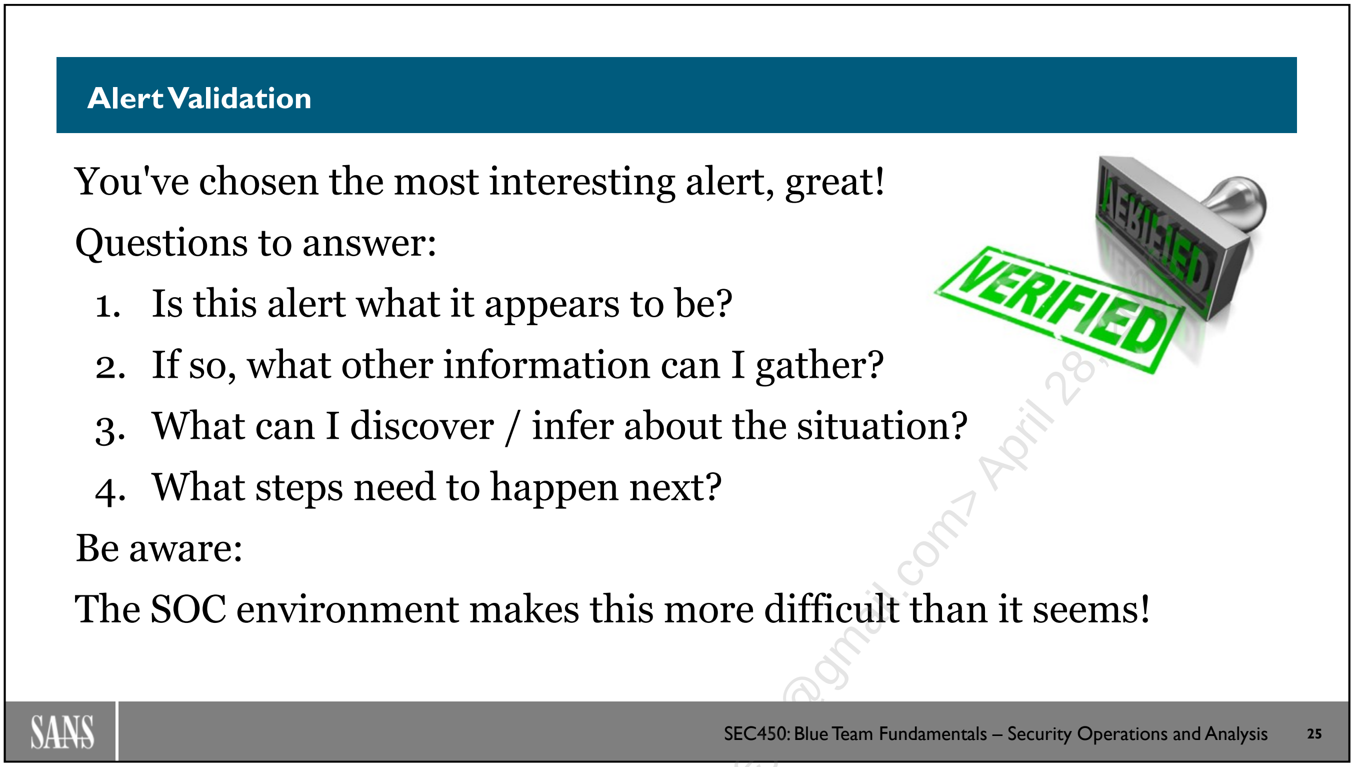
Course Roadmap

- Day 1: Blue Team Tools and Operations
- Day 2: Understanding Your Network
- Day 3: Understanding Hosts, Logs, and Files
- **Day 4: Triage and Analysis**
- Day 5: Continuous Improvement, Analytics, and Automation

Triage and Analysis

1. Alert Triage and Prioritization
2. **Perception, Memory, and Investigation**
3. Models and Concepts for Infosec
4. Exercise 4.1: Alert Triage and Prioritization
5. Structured Analytical Techniques
6. Analysis Questions and Tactics
7. Analysis OPSEC
8. Exercise 4.2: Structured Analysis Challenge
9. Intrusion Discovery
10. Incident Closing and Quality Review
11. Day 4 Summary
12. Exercise 4.3: Collecting and Documenting Incident Information

This page intentionally left blank.



Alert Validation

You've chosen the most interesting alert, great!

Questions to answer:

1. Is this alert what it appears to be?
2. If so, what other information can I gather?
3. What can I discover / infer about the situation?
4. What steps need to happen next?

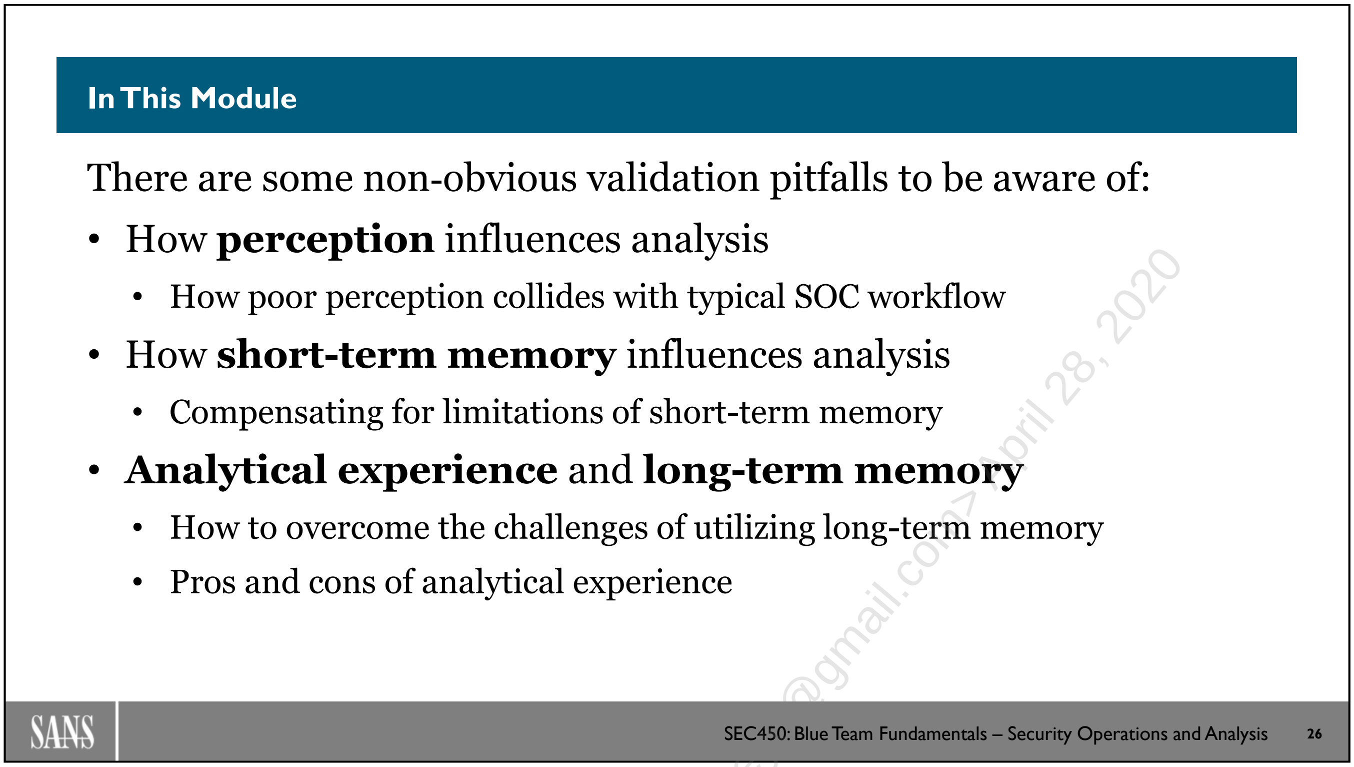
Be aware:

The SOC environment makes this more difficult than it seems!

Alert Validation

Now that you've picked out the most apparently interesting alert, it's time for the next step—validation. In a SOC, false positives are an unfortunate fact of life and, therefore, many of the items you select to triage may not be what they appear. Because of this, we must first validate what the alert seems to indicate has occurred. Questions we want to answer are not only if the alert was correct, but if so, what other information can be gathered, what can we discover or infer has happened, and what should we do next?

Considering analysts are often incentivized to move quickly through alerts, it is easy to make a mistake at this point given the competing priorities. In this section, we'll explore the problem of looking at alerts with minimal evidence under time pressure and see some of the factors that can contribute to our success or failure in this critical phase of the SOC process.



In This Module

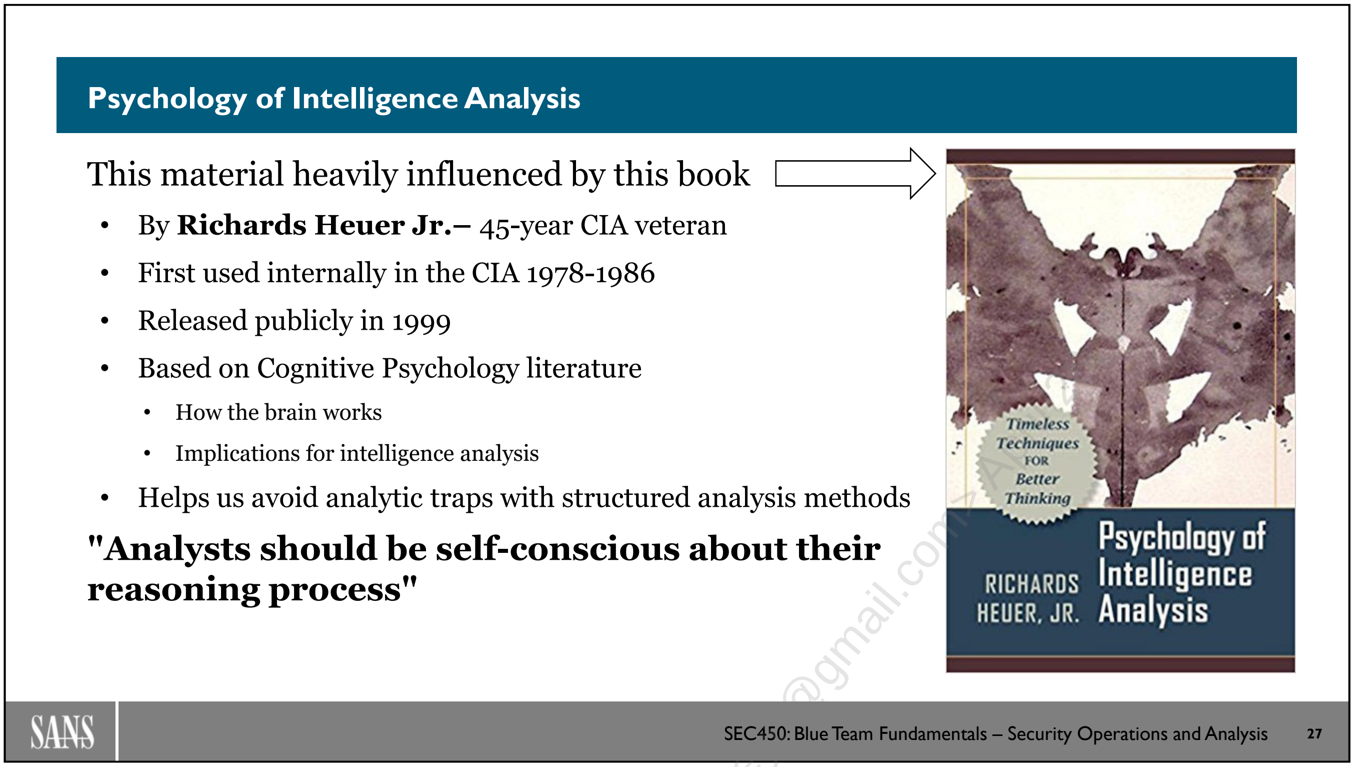
There are some non-obvious validation pitfalls to be aware of:

- How **perception** influences analysis
  - How poor perception collides with typical SOC workflow
- How **short-term memory** influences analysis
  - Compensating for limitations of short-term memory
- **Analytical experience** and **long-term memory**
  - How to overcome the challenges of utilizing long-term memory
  - Pros and cons of analytical experience

In This Module

In the validation phase, there are some non-obvious issues that can arise. These issues stem from the fact that our brains are not well-equipped to make high-quality decisions under less than perfect information without training, and the SOC environment makes it worse by emphasizing speed of analysis.

In this section, we'll explore how our perception, past experiences, short-term, and long-term memory will color our analysis, and how the mind of an experienced analyst differs from that of a newer one. There are pros and cons to be aware of at all stages of development, and step one of ensuring we don't fall victim to common errors is being aware of how our brain works and can work against us. To bring our capabilities to the next level, we must understand how our perception, memory, and judgment work, and how it affects the analysis process.



Psychology of Intelligence Analysis

This material heavily influenced by this book

- By **Richards Heuer Jr.–** 45-year CIA veteran
- First used internally in the CIA 1978-1986
- Released publicly in 1999
- Based on Cognitive Psychology literature
  - How the brain works
  - Implications for intelligence analysis
- Helps us avoid analytic traps with structured analysis methods

"Analysts should be self-conscious about their reasoning process"

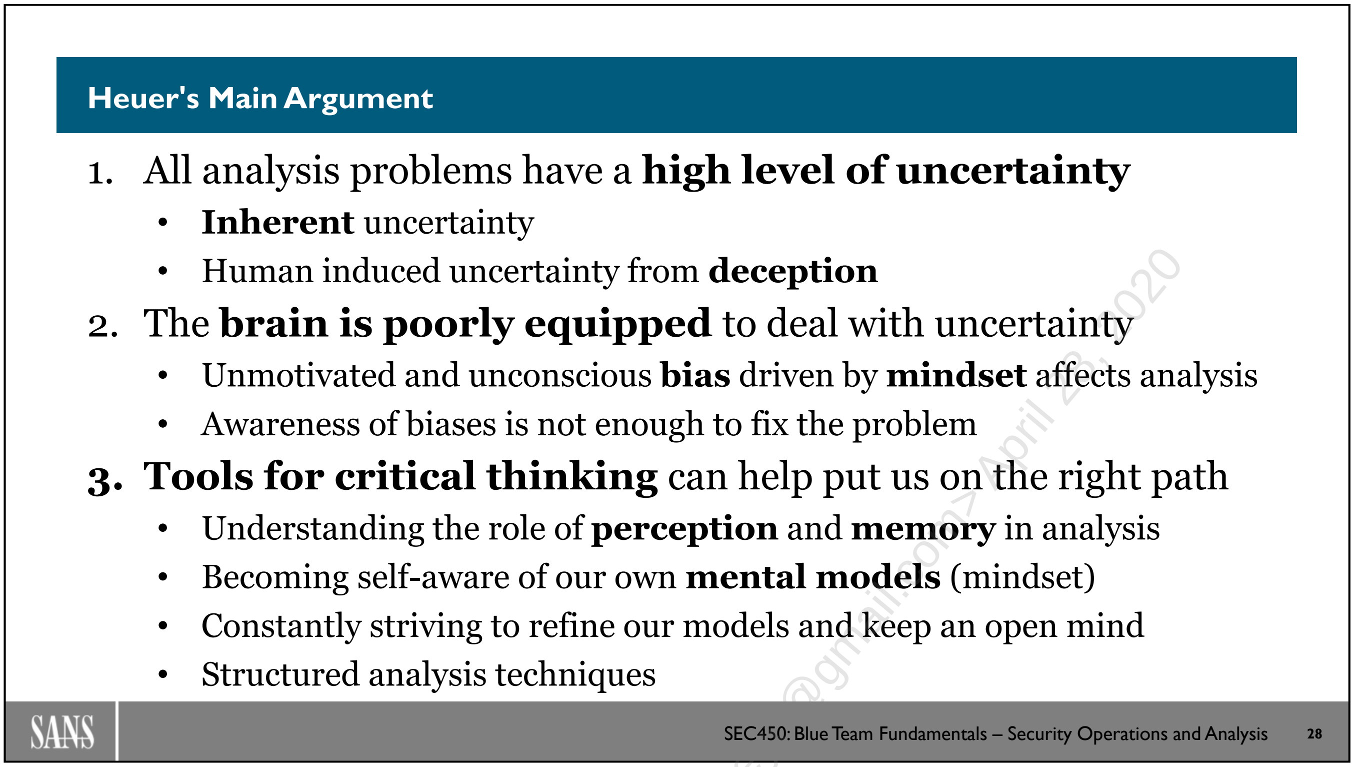
Timeless Techniques FOR Better Thinking
RICHARDS HEUER, JR.
Psychology of Intelligence Analysis

Psychology of Intelligence Analysis

The content of the analysis section of the class is highly influenced and a distillation of some of the key topics in the book *Psychology of Intelligence Analysis* by Richards J. Heuer Jr. Heuer was a 45-year veteran of the CIA and a highly influential person in teaching the art of intelligence analysis, publishing both this book and *Structured Analytic Techniques for Intelligence Analysis* in 2010. The *Psychology of Intelligence Analysis* is fortunately available free as a PDF from the CIA website and is highly recommended reading.[1] [2] The book covers perception, how the brain works, and how your mental models can color your analysis and lead you to misjudge something or fall prey to one of the brain's inherent unmotivated biases. One of the central takeaways from the book is that although our brains are poorly equipped to handle the type of situation that analysis typically presents us, there are strategies we can use to overcome it, but only if we are aware of our own process of thinking.

[1] Heuer, R. (1999). *Psychology of Intelligence Analysis*. Washington, D.C: Center for the Study of Intelligence, Central Intelligence Agency.

[2] https://www.cia.gov/library/center-for-the-study-of-intelligence/csi-publications/books-and-monographs/psychology-of-intelligence-analysis/PsychofIntelNew.pdf



Heuer's Main Argument

1. All analysis problems have a **high level of uncertainty**
   - **Inherent** uncertainty
   - Human induced uncertainty from **deception**
2. The **brain is poorly equipped** to deal with uncertainty
   - Unmotivated and unconscious **bias** driven by **mindset** affects analysis
   - Awareness of biases is not enough to fix the problem
3. **Tools for critical thinking** can help put us on the right path
   - Understanding the role of **perception** and **memory** in analysis
   - Becoming self-aware of our own **mental models** (mindset)
   - Constantly striving to refine our models and keep an open mind
   - Structured analysis techniques

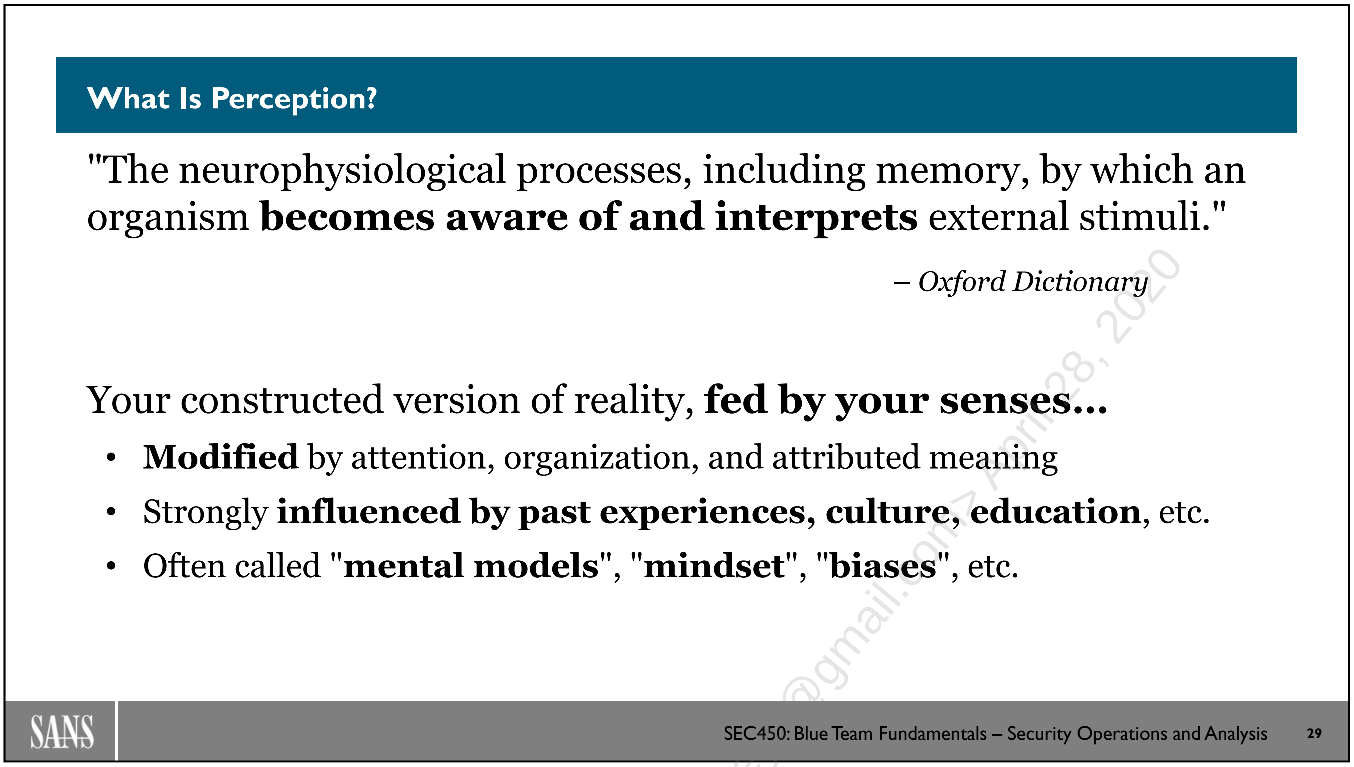
SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 28

Heuer's Main Argument

Heuer's main argument[1] when it comes to this book is that since all analysis problems (whether it be intelligence analysis or alert analysis in information security) contain a high level of uncertainty. Some of the uncertainty is born of limits of our collection capability and is just inherent to the situation; this will never go away and is purely part of the game. The second form of uncertainty is additional uncertainty introduced when our opponents purposefully try to deceive us, making correct analysis even more difficult.

Unfortunately, the human brain is poorly equipped to deal with situations rife with uncertainty of these two types. The uncertainty, combined with the way that analysis typically proceeds by giving incremental information to us over time, makes it extremely difficult to come up with the best prediction given the way the brain works. Studies suggest that even when we are aware of the problems, they will still show up, meaning we need to go further to ensure clear thinking. The method that Heuer suggests will right our path as much as possible is following structured analytic techniques such as the one he introduces in the book. Using these techniques in combination of being aware of our mindset, and constantly striving to keep an open mind and refine our mental models ensures we are doing the best possible job, free of bias, and explicit in all assumptions made.

[1] Heuer, Introduction by Jack Davis, 1999, p. xx



What Is Perception?

"The neurophysiological processes, including memory, by which an organism **becomes aware of and interprets** external stimuli."

– *Oxford Dictionary*

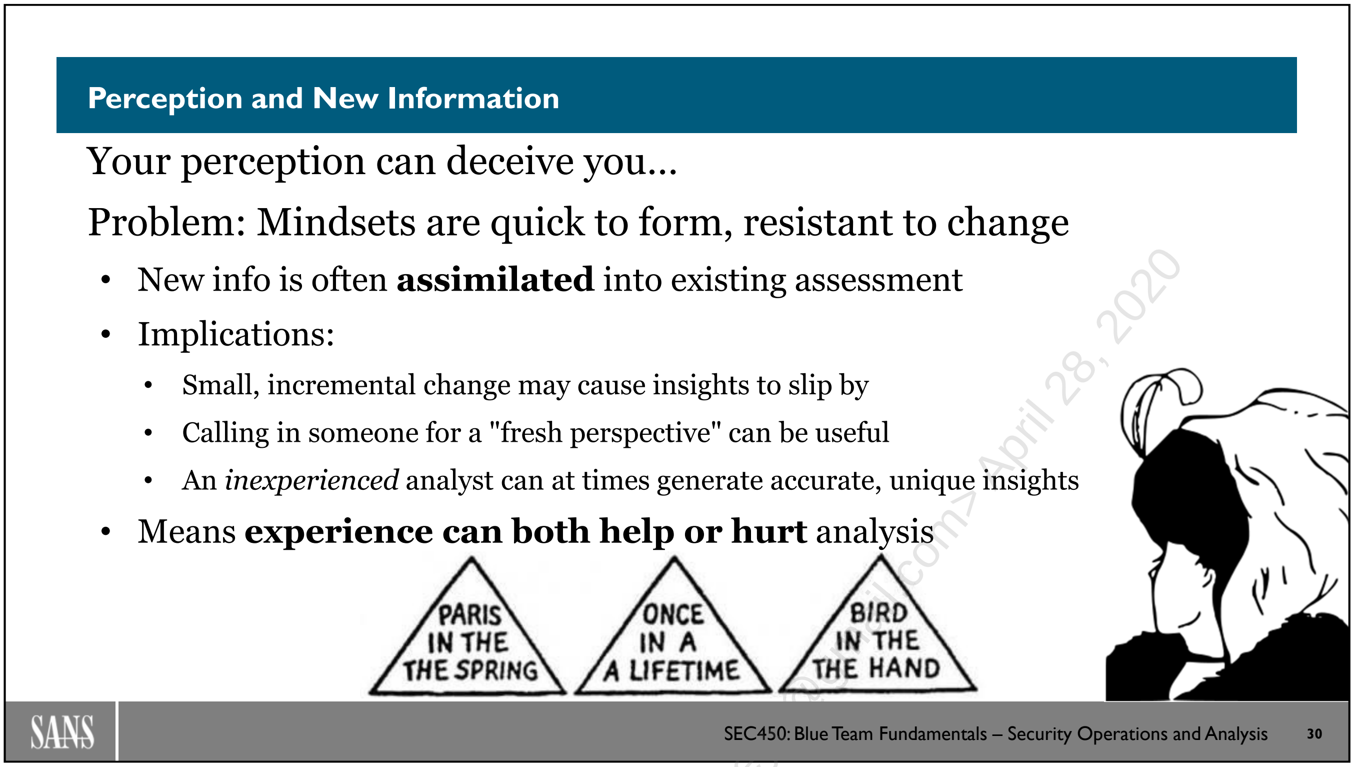
Your constructed version of reality, **fed by your senses...**

- **Modified** by attention, organization, and attributed meaning
- Strongly **influenced by past experiences, culture, education**, etc.
- Often called "**mental models**", "**mindset**", "**biases**", etc.

What Is Perception?

To start off on how these concepts connect to analysis, we first must discuss perception. Perception, according to the Oxford Dictionary is "the neurophysiological processes, including memory, by which an organism becomes aware of and interprets external stimuli." The key part of this definition is the "becomes aware of and interprets." This means perception is not just purely a passive process of information coming into our senses but is an active process informed by our senses but modified by our interpretation. According to Heuer, that interpretation can be and is affected by many things, including your past experiences, cultural values, education, and more. This changing of what is perceived can be viewed as a filter on the incoming data that can alter the way we understand and attribute meaning to it. This filter is often called your "mindset", or "mental model", "biases", "analytical assumptions" and other such names.[1]

[1] Heuer, 1999, p. 7



Perception and New Information

Your perception can deceive you…

Problem: Mindsets are quick to form, resistant to change

- New info is often **assimilated** into existing assessment
- Implications:
  - Small, incremental change may cause insights to slip by
  - Calling in someone for a "fresh perspective" can be useful
  - An *inexperienced* analyst can at times generate accurate, unique insights
- Means **experience can both help or hurt** analysis

PARIS IN THE THE SPRING

ONCE IN A A LIFETIME

BIRD IN THE THE HAND

Perception and New Information

There's another issue with our perception as well, and that is the fact that mindsets form very quickly, and once they do, they are difficult to change.[1] What this means is that once you have a theory, your brain will want you to stick with it, even if you are presented with contradictory evidence, and even when you are conscious of the tendency to do so. The implications of this feature of perception are also interesting. Have you ever been trying to solve a problem for a long time and someone completely new to the situation walks in and immediately produces the answer? This is one of the reasons that may be possible. Coming into a situation with a fresh perspective allows the new person to consider the big picture with all known evidence from the start. As someone who has been slowly given bits of information over time, you're handicapped in the way that your brain is fighting against you producing a new theory, trying to fit new information into the first hypothesis.

There are even some situations that were found to make the effect even worse. According to the studies quoted by Heuer in his book, the tendency to assimilate data is greatest when 1. the information is *more* ambiguous, 2. the assessor is more confident in the validity of the assessment, and 3. the greater the commitment to the established view. We can see how this could become a problem when trying to analyze potentially highly ambiguous situations, especially if we have a lot of experience and are incentivized to stick with our hypothesis.

Consider the drawing in the corner of this page (originally published in Puck magazine in 1915 as a cartoon titled "My Wife and My Mother-In-Law"[2]). What do you see? Some people see a young woman looking away, others see an old woman facing forward. Even if you have seen this popular example before, switching back and forth rapidly between the two perspectives is difficult. This is a simple example of the effect described above. Now imagine trying to do this using data in an investigation when you don't even know the alternative representation exists!

One of the other well-known phenomenons in psychology cited by Heuer is the fact that "we tend to perceive what we expect to perceive." [3] [4] It has been shown and over again with videos like the selective attention test video of the group playing basketball (a great demonstration of this concept if you haven't seen it).[5] It's shocking how blind humans can be to obvious contradictory or out of the ordinary information, simply because we don't expect it (look closely at the words in the triangles on this page). An interesting corollary to this, however, is that we also require more, and more unambiguous information to recognize an unexpected phenomenon than an expected one.

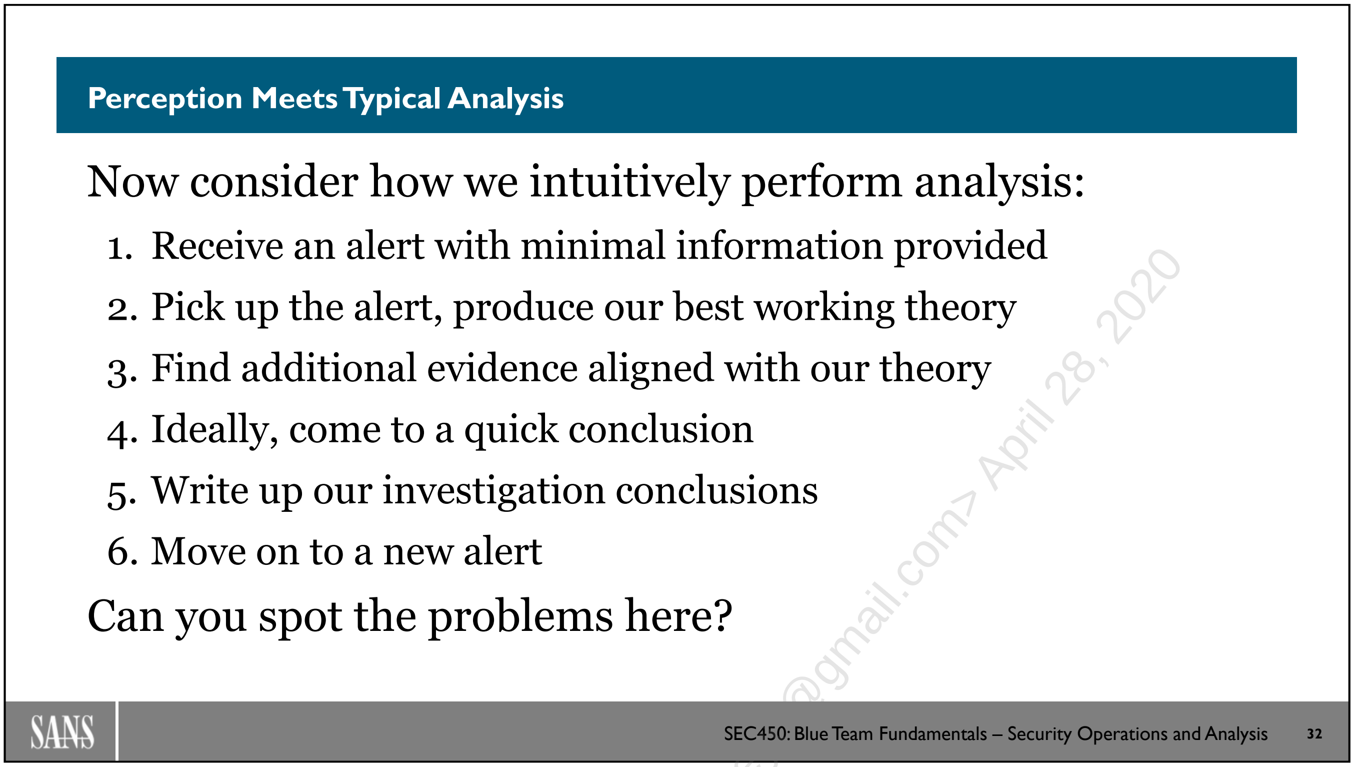
This has implications in alert analysis—when you see the same thing repeatedly, you tend to fall into a pattern of expecting certain things since many alerts are indeed repetitive. The danger, however, is going too far with this, and the corollary to the perception rule means not only will we see what we expect to see, but that we'll need even *more* information that we would expect to correct ourselves. It is these expectations formed over years of experience that will eventually become the mindset that you start to see all analysis tasks through, which can be great in some cases, but potentially misleading in others.

[1] Heuer, 1999, pp. 14-15
[2] Heuer, 1999, p. 12

[3] Heuer, 1999, pp. 8

[4] https://www.psychologytoday.com/us/blog/metacognition-and-the-mind/201806/why-we-stop-noticing-the-world-around-us

[5] https://www.youtube.com/watch?v=vJG698U2Mvo



Perception Meets Typical Analysis

Now consider how we intuitively perform analysis:

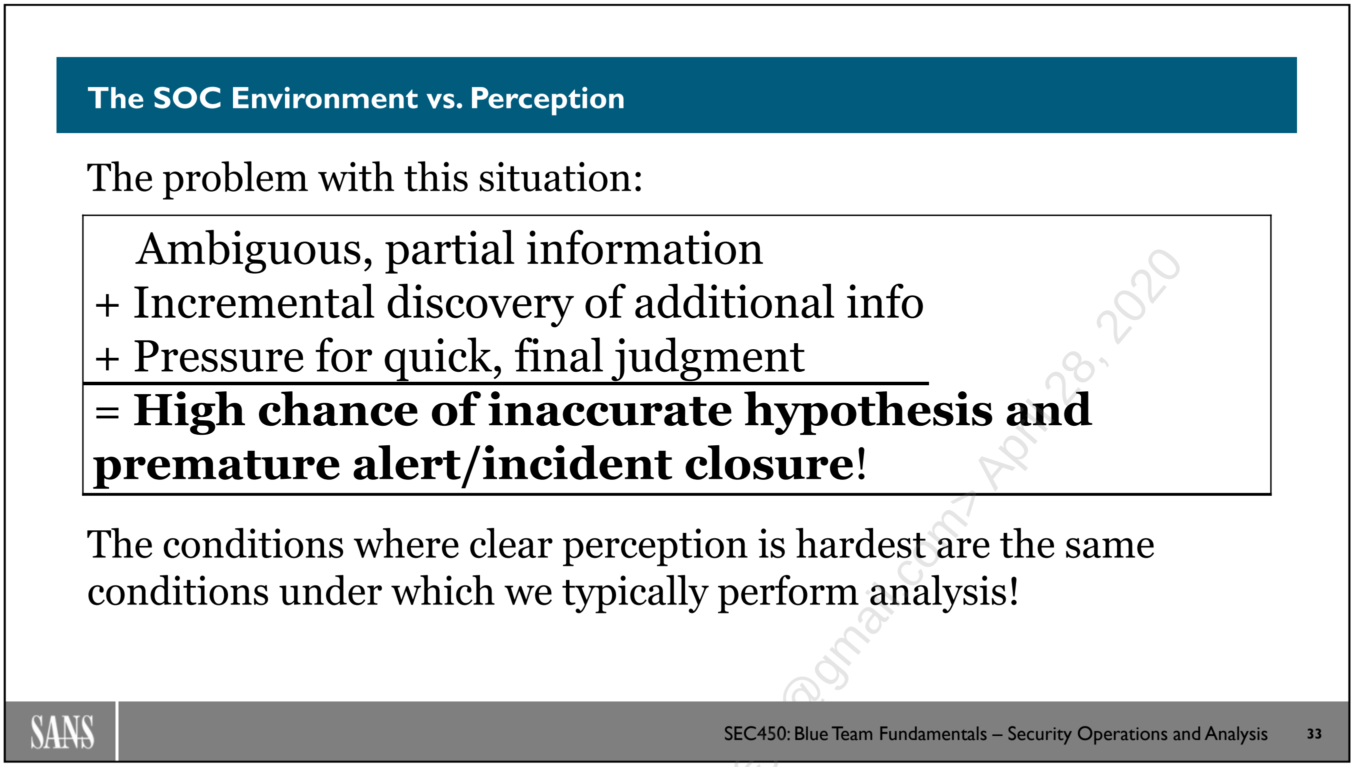
1. Receive an alert with minimal information provided
2. Pick up the alert, produce our best working theory
3. Find additional evidence aligned with our theory
4. Ideally, come to a quick conclusion
5. Write up our investigation conclusions
6. Move on to a new alert

Can you spot the problems here?

Perception Meets Typical Analysis

Given that we are predisposed to making up our mind quickly, are unlikely to change an idea once it is formed, therefore fitting new information into our first theory, consider how many analysts perform analysis: We take an alert off the pile and run through our heads several scenarios that might make sense for what happened. Once the best seeming one is picked, we go to the SIEM and other data sources and start to investigate if our theory makes sense, finding support for that theory. Once we find some amount of data that seems the judgment is confirmed, we write it all up in an incident management system, explain it to management if necessary, and take any remediation actions that are needed, then move on the next of many alerts.

This is a perfect example of how our brains produce a mindset/theory, stick with it, and assimilate all new info to fit it, and this is often the intuitive way analysts will approach analysis. Do you see the potential problem here?



The SOC Environment vs. Perception

The problem with this situation:

Ambiguous, partial information
+ Incremental discovery of additional info
+ Pressure for quick, final judgment
= **High chance of inaccurate hypothesis and premature alert/incident closure**!

The conditions where clear perception is hardest are the same conditions under which we typically perform analysis!

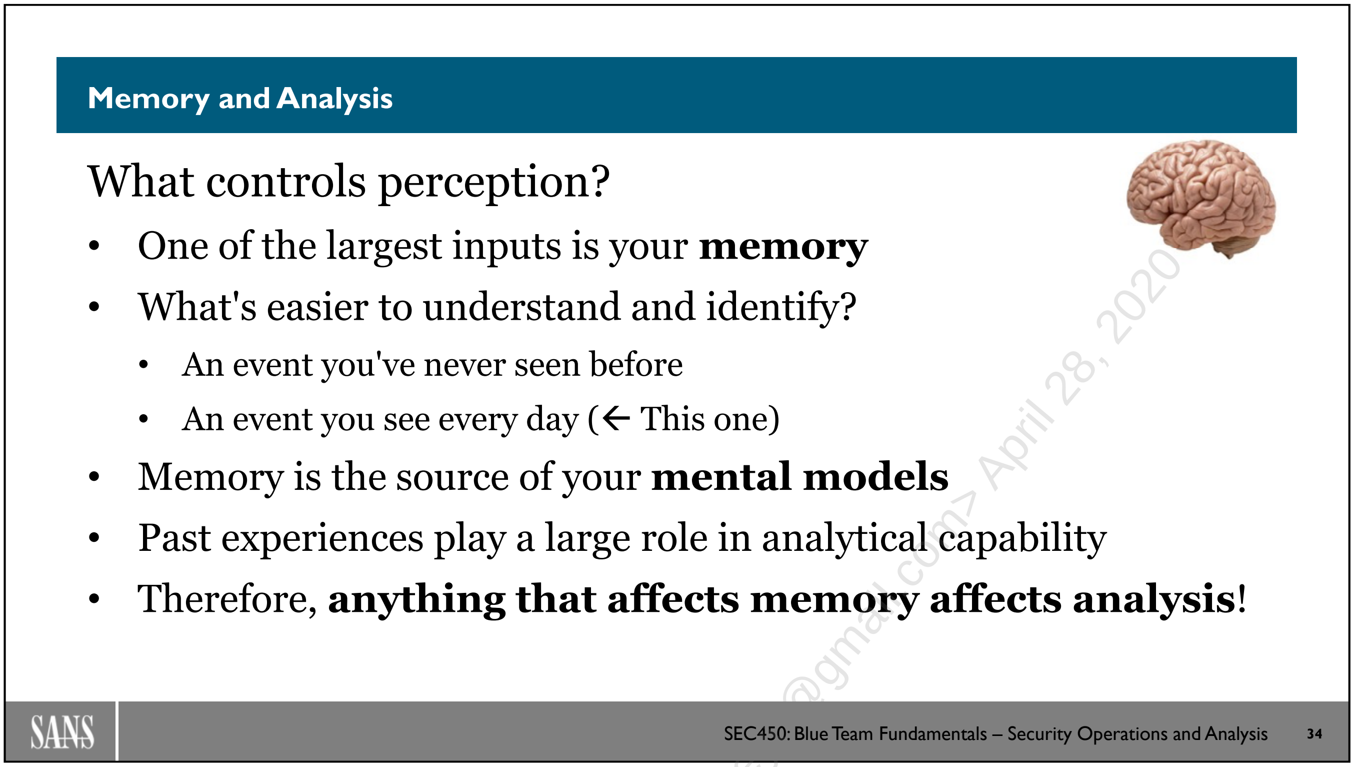
The SOC Environment vs. Perception

The main issue with the given situation is that since information in analysis is typically confusing and comes in piecemeal, we tend to start off making a theory and trying to see if the information you have fits into it. That means in the process described previously, analysts produce a working idea and are inclined to stick with that first hypothesis they thought of until the alert is worked to completion. Even in the face of contradictory information, your brain is unconsciously trying to assimilate newly gained info into the already existing theory you've produced, instead of considering new ones. This happens regardless of whether you know about the process or not. It is a subconscious function of the brain. This means, even if a new theory could be made that is much better than the old one, you are likely to be blind to it!

Instead of looking at the whole of the evidence and seeing if your new evidence matters or would change your mind if all was considered fresh, we tend to force new info to fit into the existing theory. You find yourself fighting to look at the problem with "new eyes" and get stuck in your original path. This explains how someone with no previous knowledge of a situation can sometimes walk in with a "fresh perspective" and immediately spot something that you could not. They didn't have the mental baggage of trying to assimilate an existing theory with new information and instead can look at the situation in totality. They had a clear perception not tainted by previous mindsets.

To make matters worse, we are under pressure in a SOC to move quickly. SLAs and a mountain of alerts mean you might be incentivized to move faster than you should, and faster than you *can* produce thorough analysis. On top of that, once you have made the judgment, written it up, and told management what your assessment is, you are disincentivized to change it. Doing so could create work and make it seem like you are someone who does questionable analysis and can't make up their mind. Given what we now know, you can see how an environment that awards speed combined with incrementally discovered partial information and pressure to make a final decision can be less than conducive to clear perception.

[1] Heuer, 1999, p. 15-16



Memory and Analysis

What controls perception?

- One of the largest inputs is your **memory**
- What's easier to understand and identify?
  - An event you've never seen before
  - An event you see every day (← This one)
- Memory is the source of your **mental models**
- Past experiences play a large role in analytical capability
- Therefore, **anything that affects memory affects analysis**!

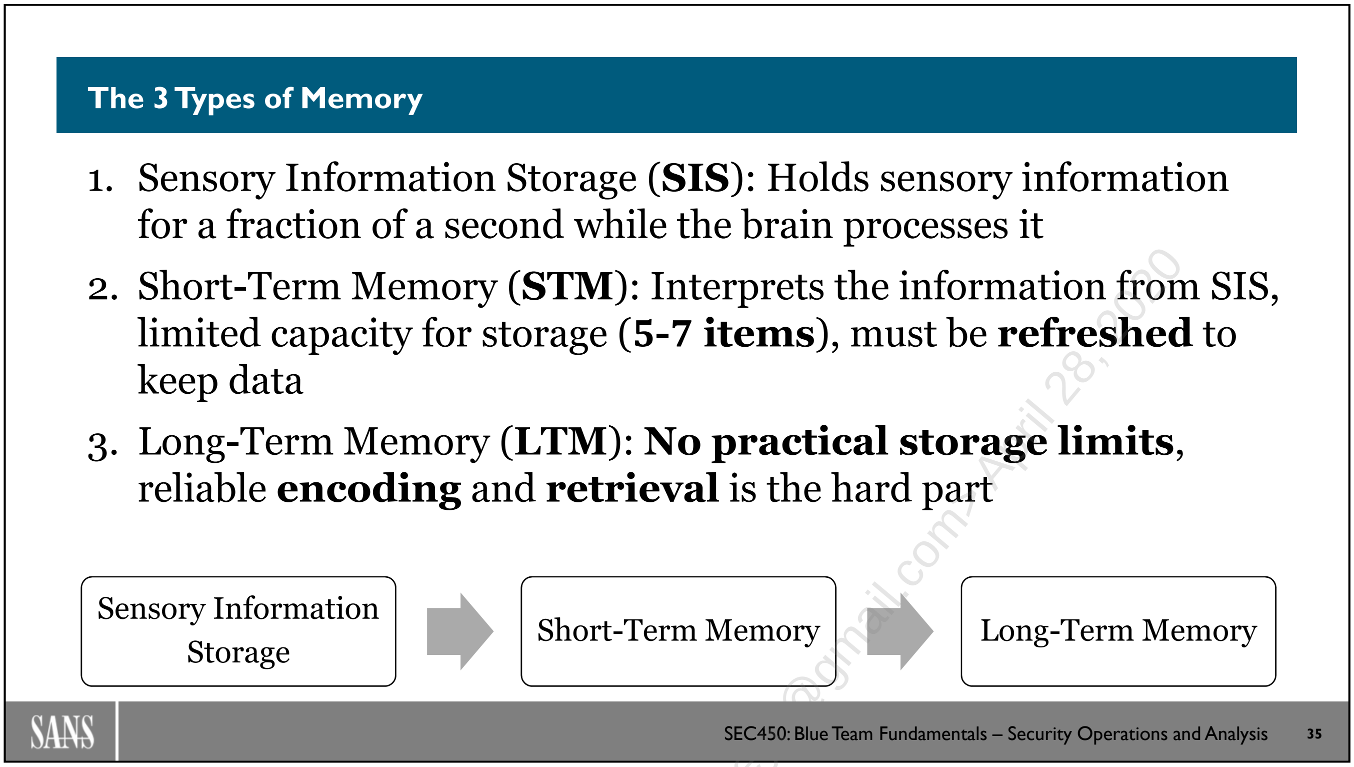
SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 34

Memory and Analysis

While perception plays a very important part in your analytical capability, one of the largest factors in what you perceive is your memory. Your past experiences are one of the sources of your mental models and they can significantly affect how you perceive a given situation (in other words, can accurately and quickly judge what is going on). Therefore, anything that affects memory will directly affect the output of analysis.[1] In this module, we'll look at the nature of short-term and long-term memory to see how they can directly affect the outcome of our investigations.

[1] Heuer, 1999, p. 17



The 3 Types of Memory

1. Sensory Information Storage (**SIS**): Holds sensory information for a fraction of a second while the brain processes it
2. Short-Term Memory (**STM**): Interprets the information from SIS, limited capacity for storage (**5-7 items**), must be **refreshed** to keep data
3. Long-Term Memory (**LTM**): **No practical storage limits**, reliable **encoding** and **retrieval** is the hard part

Sensory Information Storage → Short-Term Memory → Long-Term Memory

The 3 Types of Memory

Heuer describes that there are at least 3 types of memory known to exist[1]:

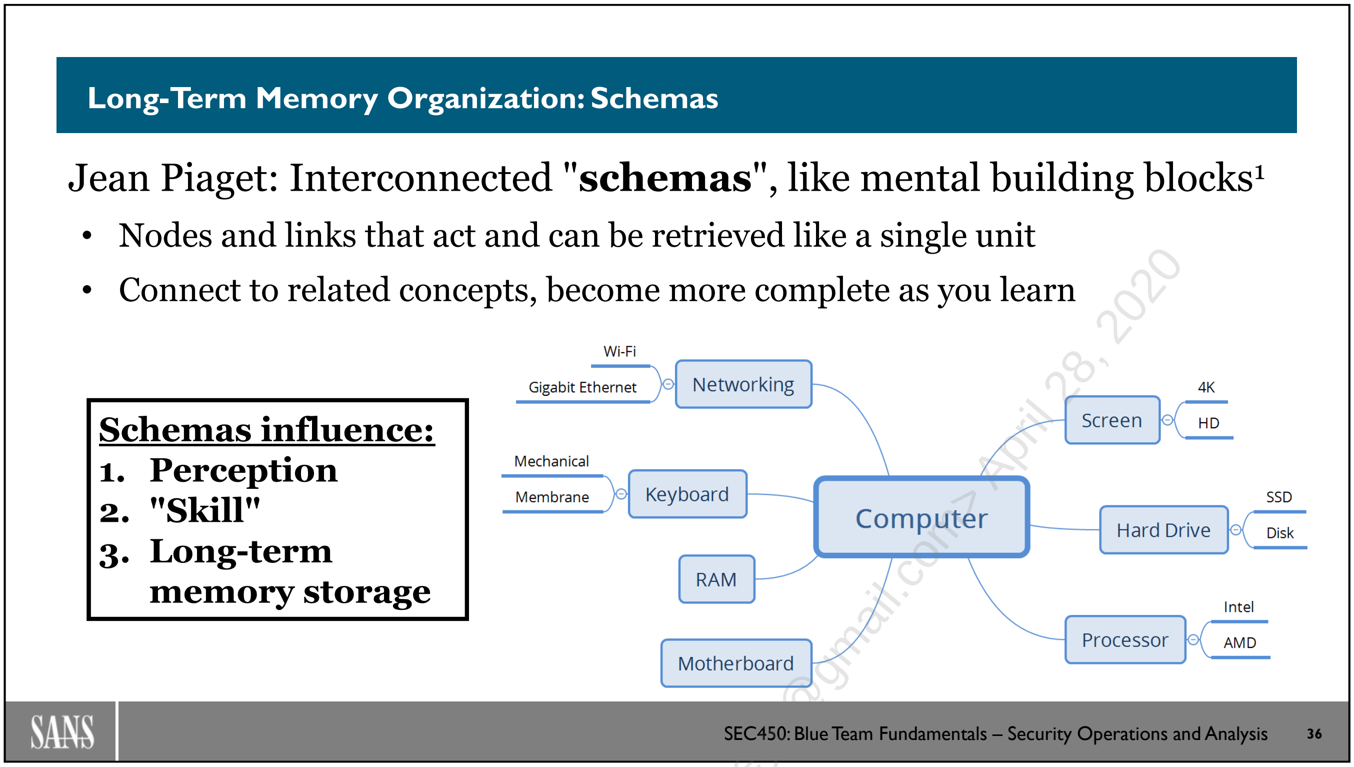
Sensory Information Storage (SIS): SIS holds sensory images for only a fraction of a second. The function of SIS is to hold an event in the brain long enough that it can be processed, even if the event itself has ended.

Short-Term Memory (STM): While SIS holds the complete sensory information for an event, STM steps in to find and store the interpretation of that event, such as the words in a sentence. One of the most important characteristics of STM is that it can hold only a few items at any given time (5-7 is the number commonly quoted). The only way to keep information in STM is repeating it over and over again, possibly passing it into LTM. The limitations of short-term memory, which will be discussed later, drive the importance of writing information down as an investigation is proceeding. We will discuss this further in a later portion of the class.

Long-Term Memory (LTM): Some information that enters STM will eventually be stored away for long-term retrieval in LTM. It is this type of memory that is most useful to understand for analysts. Having a working mental model of how it is organized can also help us understand its function and better deal with limitations.

Our interest lies in understanding the limits of short-term memory and the *organization* of long-term memory. Since memory plays a large role in the analytical process, taking some time to understand the function and operation of memory can help us better master it and become better analysts.

[1] Heuer, 1999, pp. 17-20



Long-Term Memory Organization: Schemas

Jean Piaget: Interconnected "**schemas**", like mental building blocks[1]

- Nodes and links that act and can be retrieved like a single unit
- Connect to related concepts, become more complete as you learn

Schemas influence:

1. **Perception**
2. **"Skill"**
3. **Long-term memory storage**

Long-Term Memory Organization: Schemas

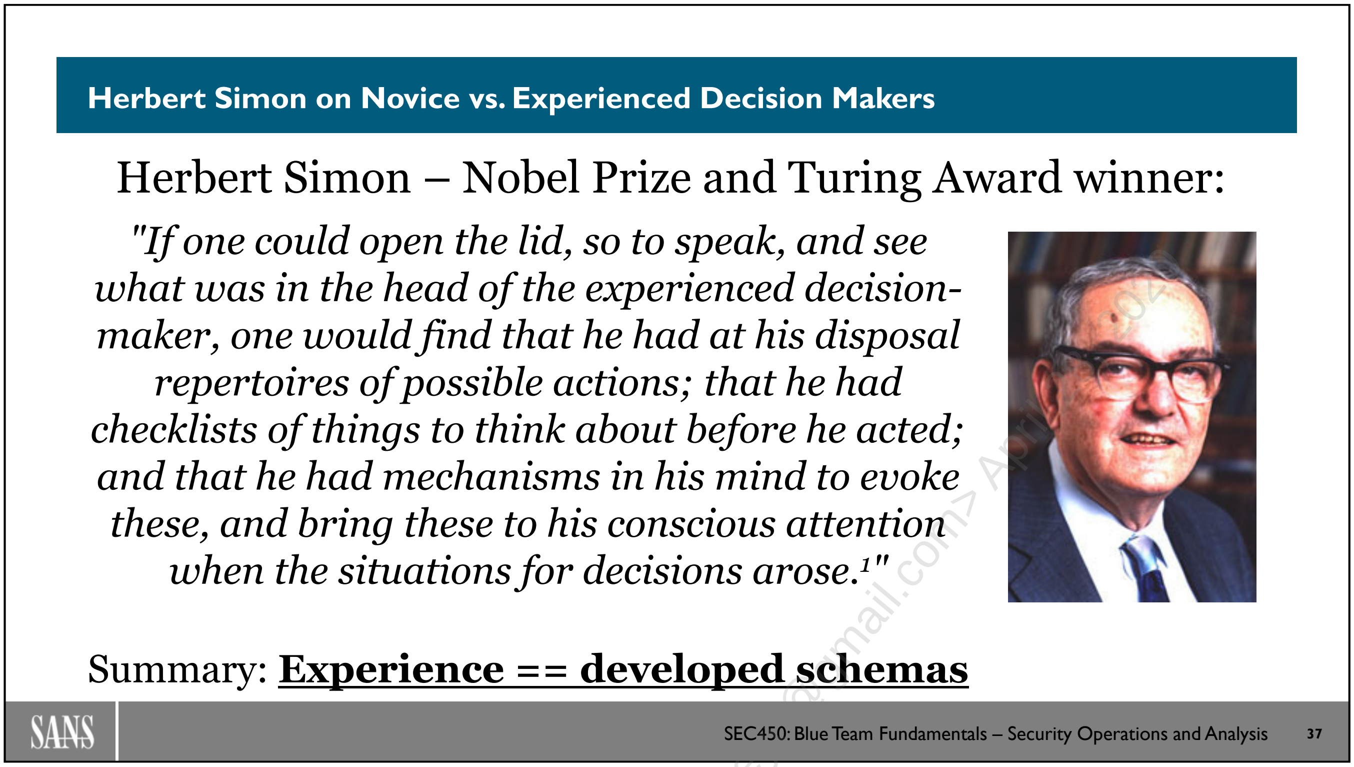
How is memory organized within this web? Although no one knows for sure, one world-renowned researcher and psychologist, Jean Piaget, theorized we could imagine building blocks of concepts and their related entities in what he called "schemas."[1] Schemas store single concepts and situations with which we are familiar and can typically be thought of as a single unit as well as all concepts that easily relate to that schema.[2] One might think of a computer and immediately recall one whenever it is mentioned. If you have only a passing familiarity with computers, this may be tied into concepts such as the keyboard, screen, and power adapter. A highly technical person, however, would likely have a more complex and filled out schema with additional connections. They may also be able to connect the idea to the motherboard, CPU types and features, RAM, and network traffic. This translates to more complex and connected pathways in the web of neurons that store those concepts. The key item of interest to us is how schemata relate to the ability to analyze a situation and learn new concepts—the more completely and deeply you understand a given concept and related items, the easier you will be able to bring in and analyze new but similar information.

As we will see in the next few slides, there are three main reasons we care about schemas:

1. Their presence is what separates a new from an experienced decision maker in a topic or skill.
2. They are *very* highly connected to what we perceive.
3. Tying new concepts to existing schemas facilitates fast, efficient storage in long term memory.

[1] Memory Schemas: https://www.youtube.com/watch?v=Y-AuJiFSpwo

[2] Heuer, 1999, p. 22



Herbert Simon on Novice vs. Experienced Decision Makers

Herbert Simon – Nobel Prize and Turing Award winner:

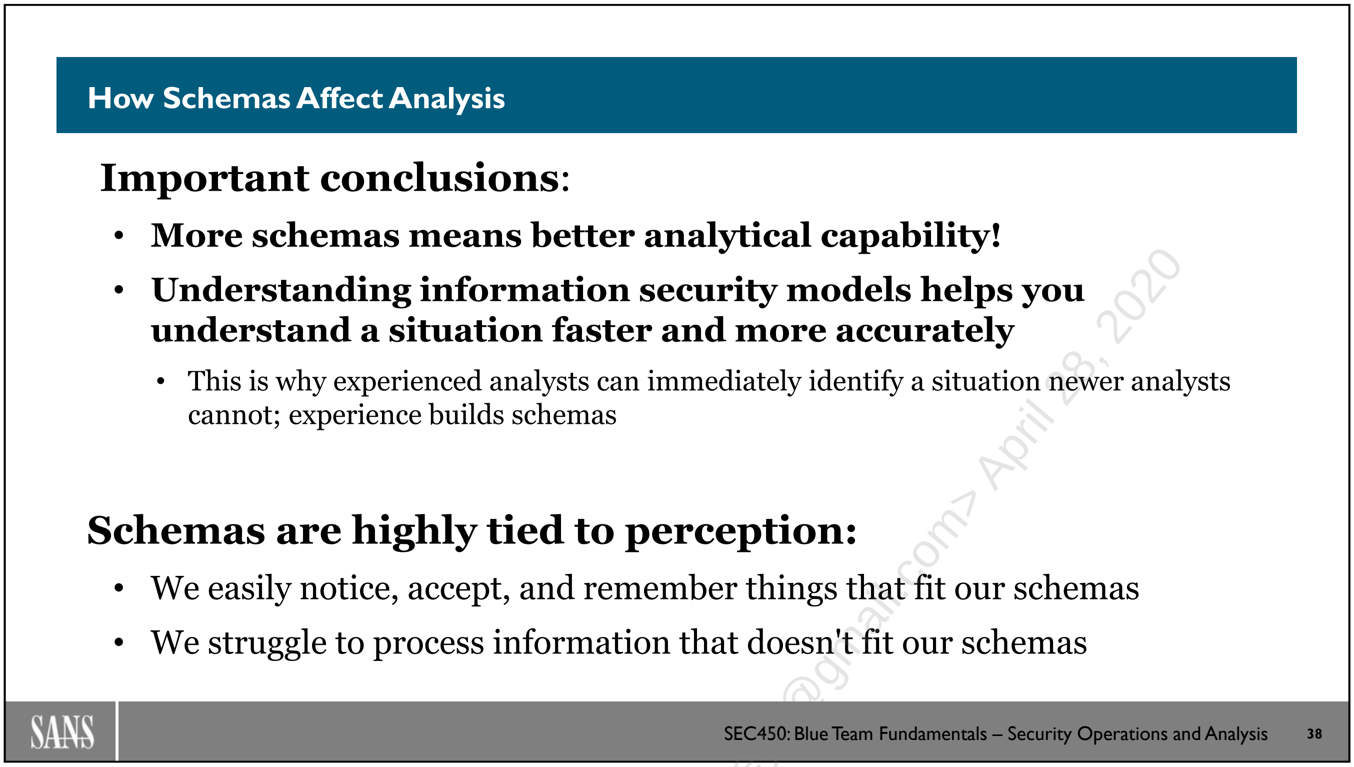
"If one could open the lid, so to speak, and see what was in the head of the experienced decision-maker, one would find that he had at his disposal repertoires of possible actions; that he had checklists of things to think about before he acted; and that he had mechanisms in his mind to evoke these, and bring these to his conscious attention when the situations for decisions arose.[1]"

Summary: **Experience == developed schemas**

Herbert Simon on Novice vs. Experienced Decision Makers

Herbert Simon was a researcher and pioneer in the field of the cognitive processes for decision making. This slide shows one of his famous quotes that relates to the previous slide in respect to what differentiates a seasoned decision maker (or analyst in our case), and a novice. The difference is that the experienced practitioner has a wealth of knowledge to draw on, the schemas to process a situation, and mental models that have been built up over the period of their career to quickly understand the data. It is for this reason that, throughout this class, we will attempt to provide as many frameworks, mental models, and analogies as possible, and use hands-on labs to reinforce the concepts. The goal is to tie in new ideas to schemas you already are familiar with, give you mental models with which to view the data, and give you multiple experiences processing data in this light. The goal is to facilitate quick learning and comprehension of the topics throughout the class.

[1] Simon, H. (1996). *Models of my life*. Cambridge, Mass: MIT Press.



How Schemas Affect Analysis

Important conclusions:

- **More schemas means better analytical capability!**
- **Understanding information security models helps you understand a situation faster and more accurately**
  - This is why experienced analysts can immediately identify a situation newer analysts cannot; experience builds schemas

Schemas are highly tied to perception:

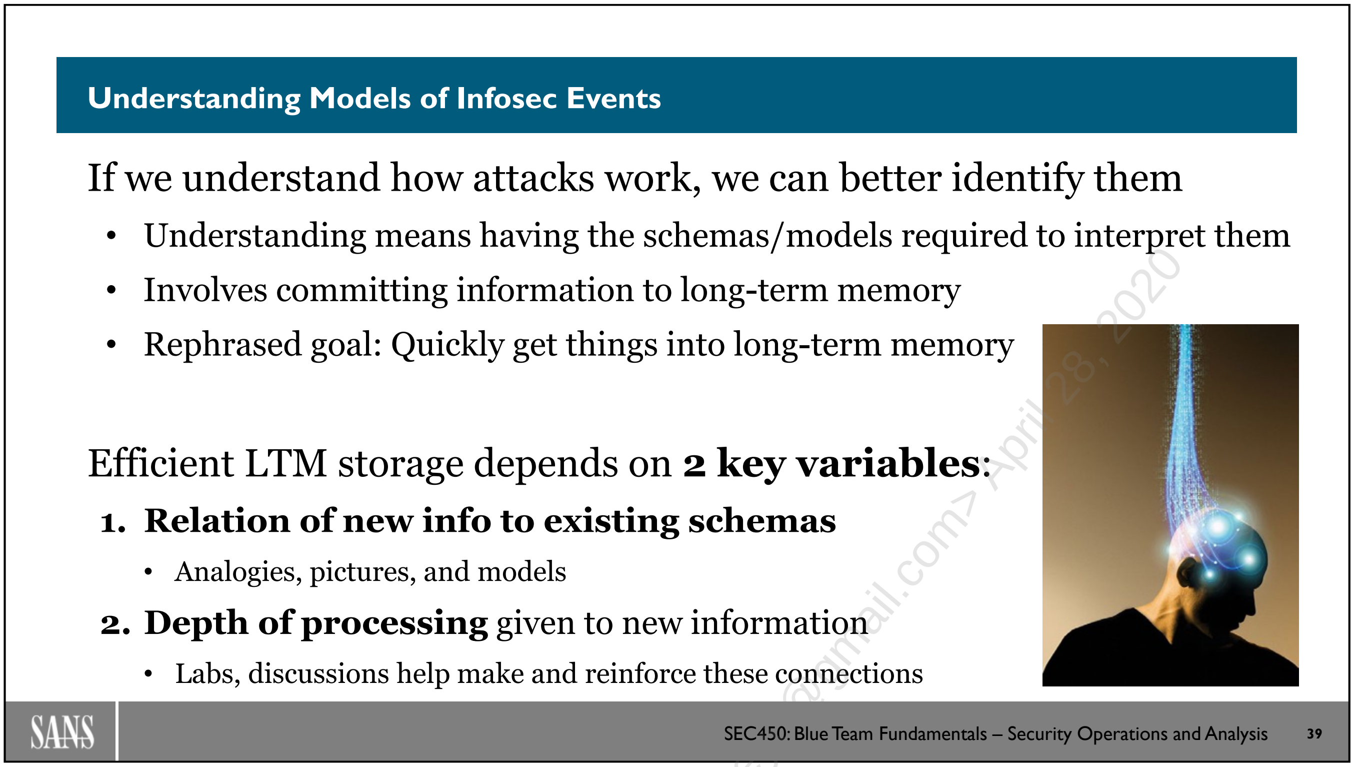
- We easily notice, accept, and remember things that fit our schemas
- We struggle to process information that doesn't fit our schemas

How Schemas Affect Analysis

The thing to know about schemas is that they are also *highly* tied to your perception. Researchers have found this to be true to the extent that the mind struggles and does a poor job at accepting and processing information that does not fit one of our existing schemas. When encountering such information, studies find that people are likely to either quickly forget the contradictory information, dismiss it, or re-interpret it in a way that allows it to still fit within what they know.[1]

This leads us to one of the most important conclusions of this information: *The type and number of related schemas you have available are correlated with your analytical capability*. Those who have more schemas related to information security and attack and defense knowledge will inherently perceive more than those who do not. Whether a concept exists as a schema in the brain or not can make all the difference in a situation and allow a more experienced analyst to spot an attack, interpret data more quickly and accurately, or understanding a new concept quickly. Throughout the class, we will demonstrate as many concepts as possible that will hopefully become new usable patterns in the brain, allowing you to more quickly and intuitively understand data. The next module will be dedicated to defensive and offensive concepts in an attempt to construct some of these important ideas in your head and will serve as a lens you can use to start interpreting data with through the rest of your career.

[1] Heuer, 1999, p. 23



Understanding Models of Infosec Events

If we understand how attacks work, we can better identify them

- Understanding means having the schemas/models required to interpret them
- Involves committing information to long-term memory
- Rephrased goal: Quickly get things into long-term memory

Efficient LTM storage depends on **2 key variables**:

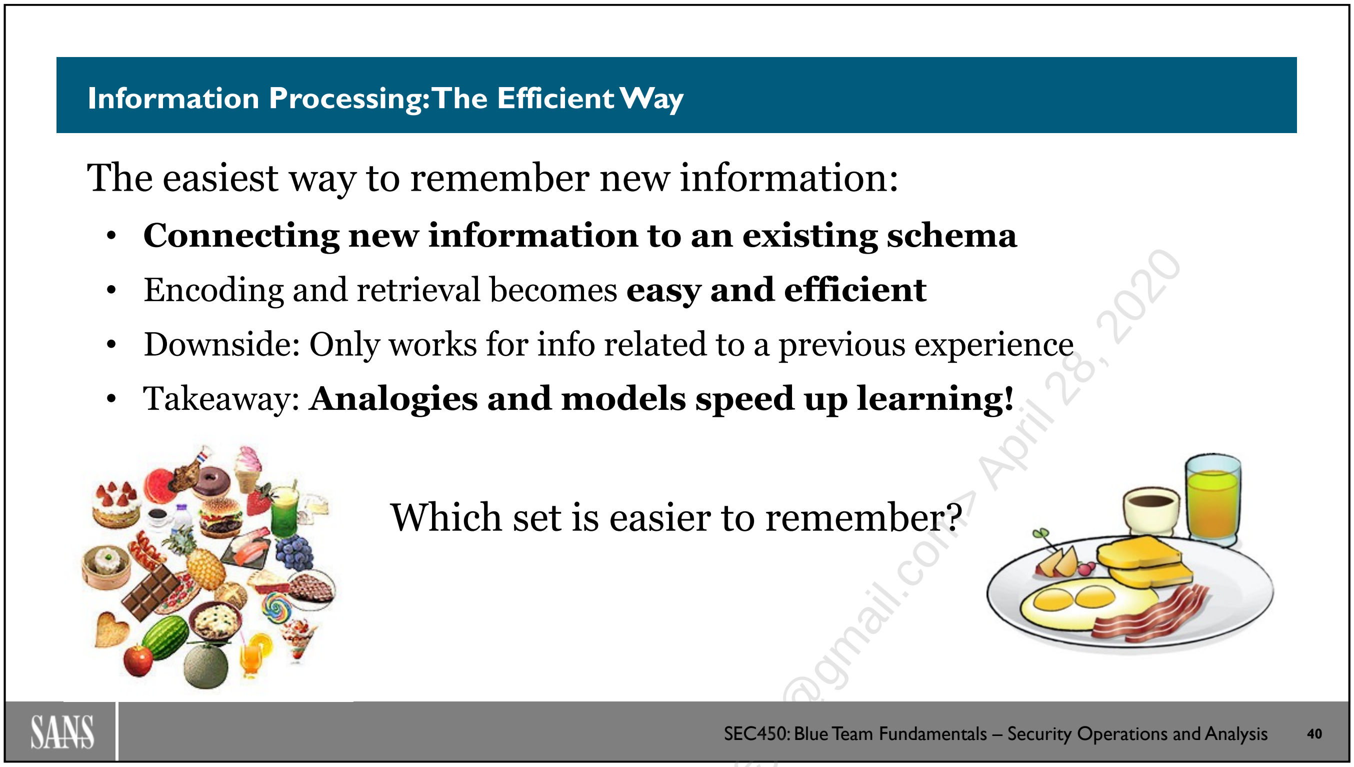
1. **Relation of new info to existing schemas**
   - Analogies, pictures, and models
2. **Depth of processing** given to new information
   - Labs, discussions help make and reinforce these connections

Understanding Models of Infosec Events

Therefore, since this class is all about jump-starting your career as an analyst, we can now say one of our goals generically is to try to present to you as many models (schemas) as possible and help you commit them to long term memory. You've likely heard of the "SANS Firehose" effect—you're learning so much information so fast, it may be hard to recall it all at the end of six days. How do we avoid this to the best possible extent and help you remember information the first time? To remember as much as possible, you must ensure you are learning in an efficient way. One of the most efficient ways is to form an association with new material to information you are already familiar with.

Association depends on two key variables. One is the relation of the new info to something you already know—is the topic totally foreign and dissimilar to anything you are familiar with? If so, it may be much more complicated to learn compared to information where analogies and familiar concepts can be used to explain it. The other way is to exert a higher effort to process the information. Models, analogies, stories, and labs this class strive to help you process the material with the intention that it will boost information retrieval after class is finished. Explaining the concepts in relatable ways, seeing their models (which we will do later), and using it in hands-on labs helps you build associations that play an important part in information retention.

[1] Heuer, 1999, p. 23-24



Information Processing: The Efficient Way

The easiest way to remember new information:

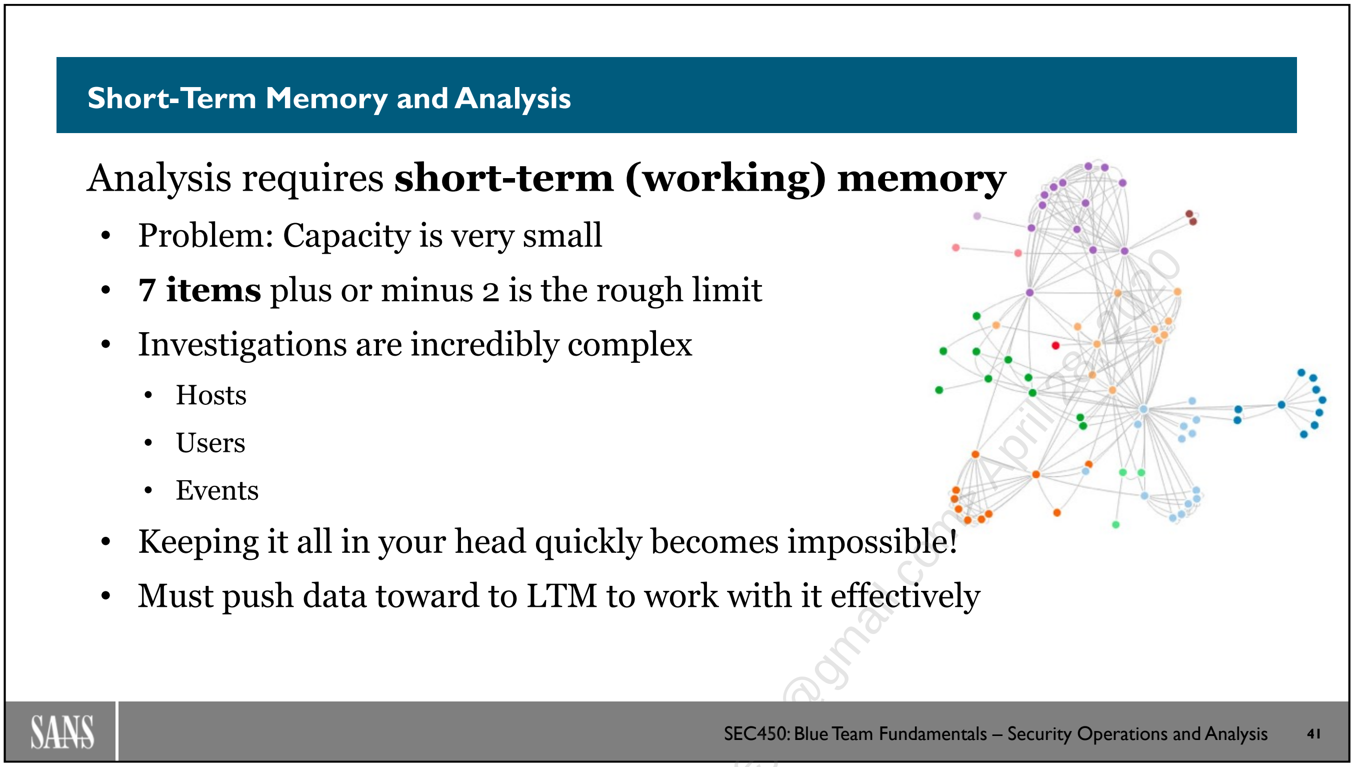
- **Connecting new information to an existing schema**
- Encoding and retrieval becomes **easy and efficient**
- Downside: Only works for info related to a previous experience
- Takeaway: **Analogies and models speed up learning!**

Which set is easier to remember?

Information Processing: The Efficient Way

The most efficient way to move something from short-term memory to long-term memory is the process called "assimilation." As the name sounds, assimilation is bringing in new information, relating it to an already existing schema, and encoding it alongside what is already present.[1] This is the easiest and most ideal way to learn and is why the use of things like analogies are so effective in teaching. When we already understand a similar concept that can be tied to a new concept, the idea can be easily visualized and understood compared to an idea totally foreign to the student. Assuming you can find a similar or related item to compare information to, encoding new information proceeds much more efficiently than creating the artificial connection when making a mnemonic device. If you were to try to remember all the items on the plate in this picture, you would probably be able to do it almost instantaneously compared to the random items on the left of the slide (at least for those of us used to a U.S.-style breakfast). Why is that? Because they all easily can be assimilated with the "typical breakfast foods" schema you already have ingrained in your head. This makes it efficient to store the additional information compared to the random foods that have no resemblance to any category.

[1] Heuer, 1999, p. 25



Short-Term Memory and Analysis

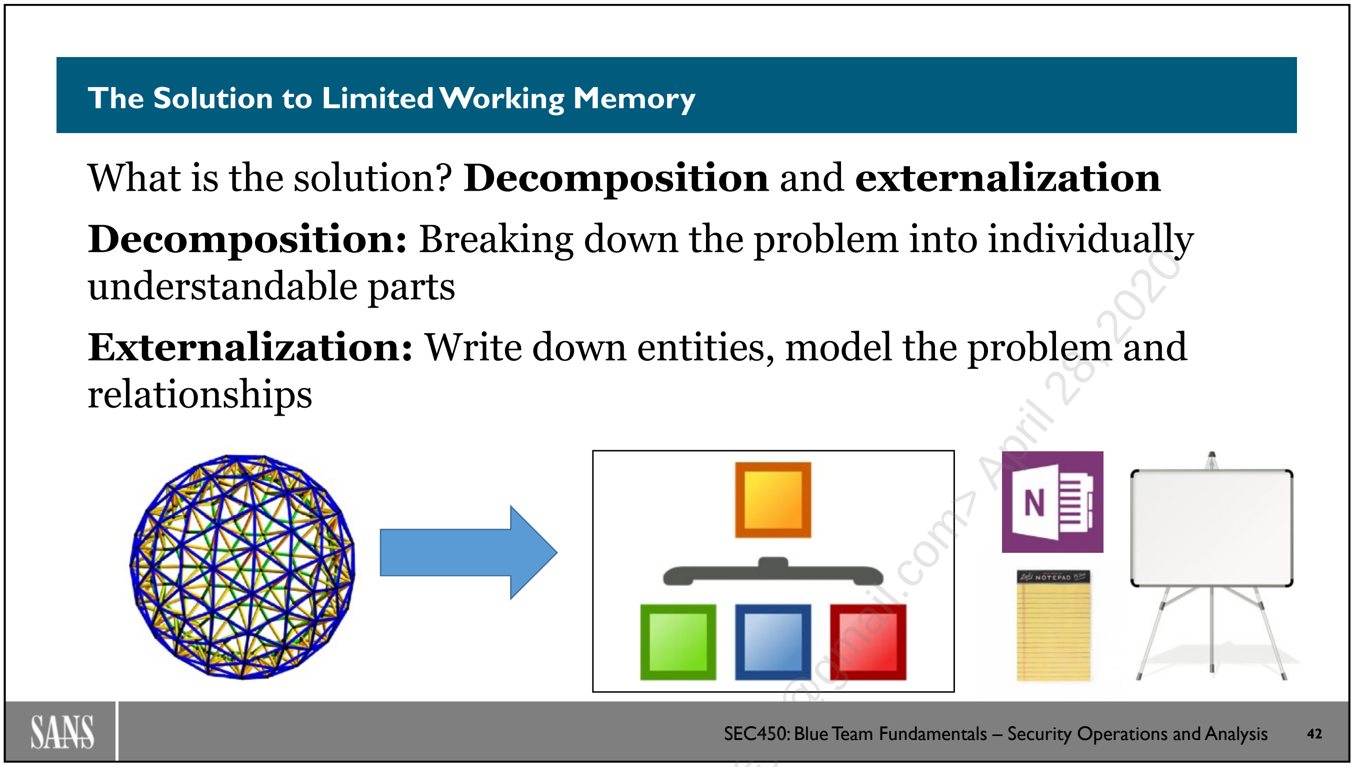
Analysis requires **short-term (working) memory**

- Problem: Capacity is very small
- **7 items** plus or minus 2 is the rough limit
- Investigations are incredibly complex
  - Hosts
  - Users
  - Events
- Keeping it all in your head quickly becomes impossible!
- Must push data toward to LTM to work with it effectively

Short-Term Memory and Analysis

What about short-term memory? The most important thing to know about short-term (sometimes referred to as "working") memory when it comes to analysis is that studies show it can hold roughly 7 items at once, give or take a few.[1] Why does this matter to us? Because when we're trying to understand a relationship between IP addresses, ports, hosts, and users, the number will almost always be higher than 7. This is already too high to deal with, but on top of that, the relationships between them grow even faster as every item is added, making things worse. That means we need a way to supplement our short-term memory capability to even get the information in a workable form. Trying to do it in our head is bound for failure.

[1] Heuer, 1999, pp. 27-28



The Solution to Limited Working Memory

What is the solution? **Decomposition** and **externalization**

Decomposition: Breaking down the problem into individually understandable parts

Externalization: Write down entities, model the problem and relationships

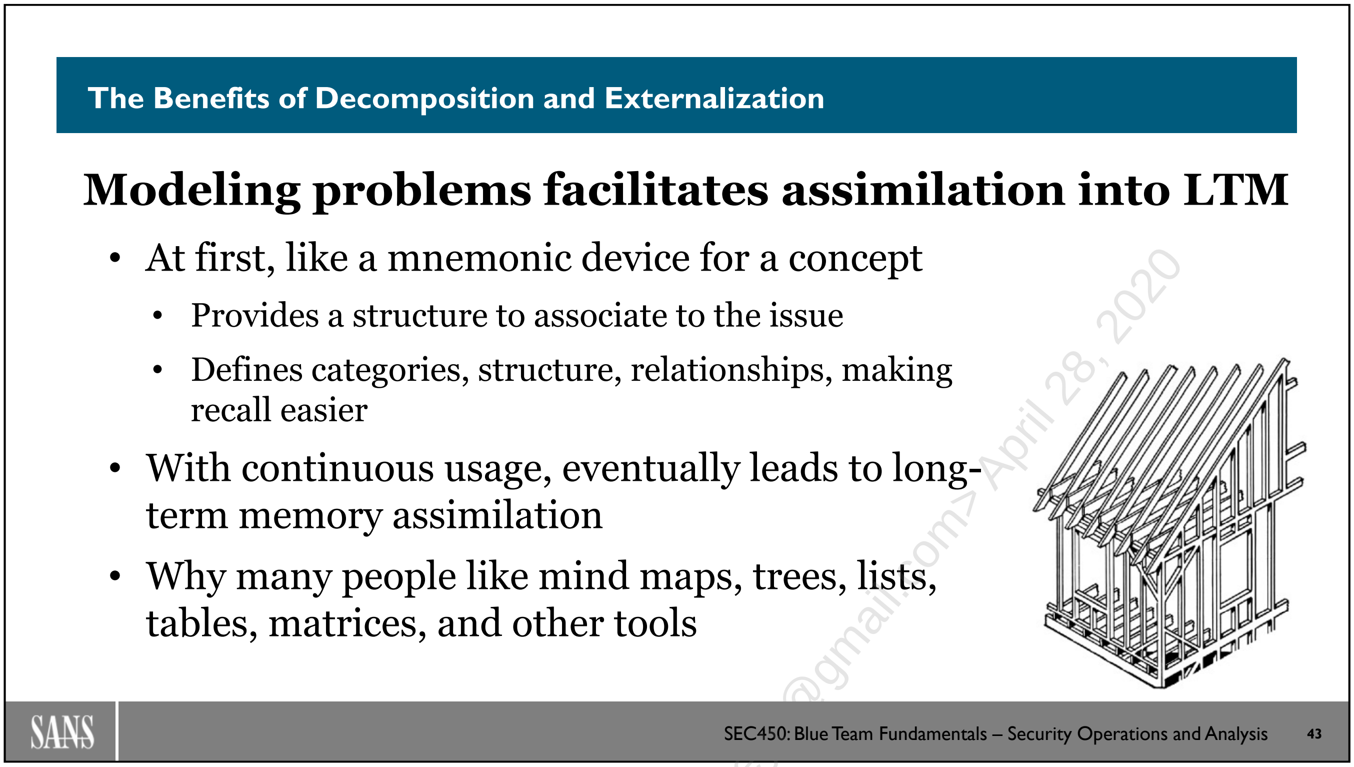
SEC450: Blue Team Fundamentals – Security Operations and Analysis 42

The Solution to Limited Working Memory

So, what is the solution? Heuer recommends using *decomposition* and *externalization* of the problem.

Decomposition of a problem means breaking it down into smaller, easier to understand chunks and taking each of those on separately. Once each piece has been understood, then each piece can be put back together to start to see the bigger picture. This is very similar to systems thinking. We abstract a piece of the problem to be worked on independently, focusing on inputs, outputs, and processes internal to the problem, then join it back up to see how it works within the system at large. Since this will still result in more than we can deal with in our minds, externalization is the way of being able to keep a bunch of disparate pieces at the forefront of our mind as we work on the problem. Externalization means getting the problem out of our head and onto paper, computer, or another medium where we can not only see all the individual pieces without resorting to short term memory, but also are able to map out the relationships between them.[1] Not only will this help us break down a big complex problem into something manageable, it also helps us build a model of the problem in our head and can begin the process of forming schemas that will eventually make their way into long-term memory. This is why so many people take on problem solving using drawings with mind maps, relationship diagrams, trees and lists. These methods give structure to a problem the brain can associate with already existing schemas, which speeds comprehension of the item being analyzed.

[1] Heuer, 1999, pp. 27-28



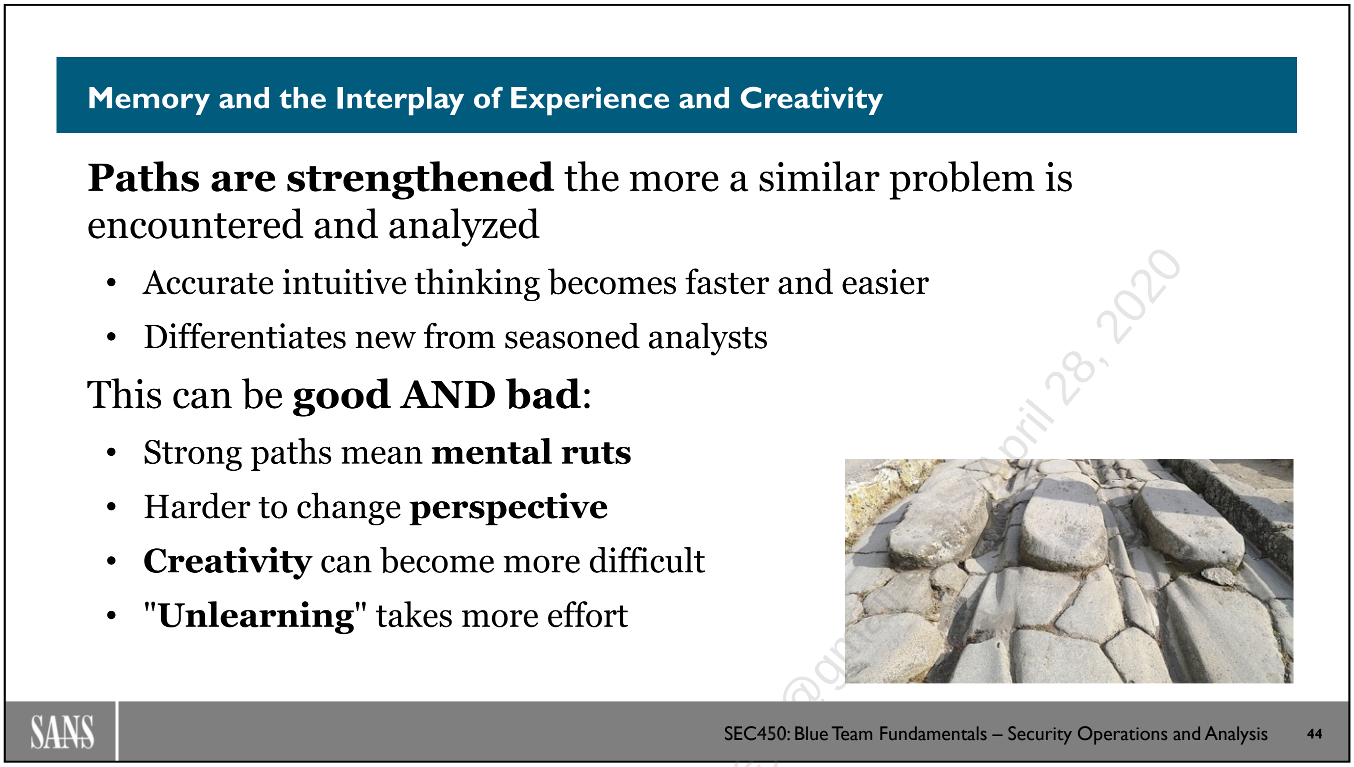
The Benefits of Decomposition and Externalization

Modeling problems facilitates assimilation into LTM

- At first, like a mnemonic device for a concept
  - Provides a structure to associate to the issue
  - Defines categories, structure, relationships, making recall easier
- With continuous usage, eventually leads to long-term memory assimilation
- Why many people like mind maps, trees, lists, tables, matrices, and other tools

The Benefits of Decomposition and Externalization

Like a mnemonic device, the decomposed and modeled version of the problem can now act as an artificial frame in your head for the problem to be related to existing schemas. Though it may take a while for the models to be remembered, with continued use and association, the models will eventually become part of your long-term memory and be assimilated with the concepts you already understand. Once this occurs, the problem is fully and truly understood, and can be worked on much more effectively.



Memory and the Interplay of Experience and Creativity

Paths are strengthened the more a similar problem is encountered and analyzed

- Accurate intuitive thinking becomes faster and easier
- Differentiates new from seasoned analysts

This can be **good AND bad**:

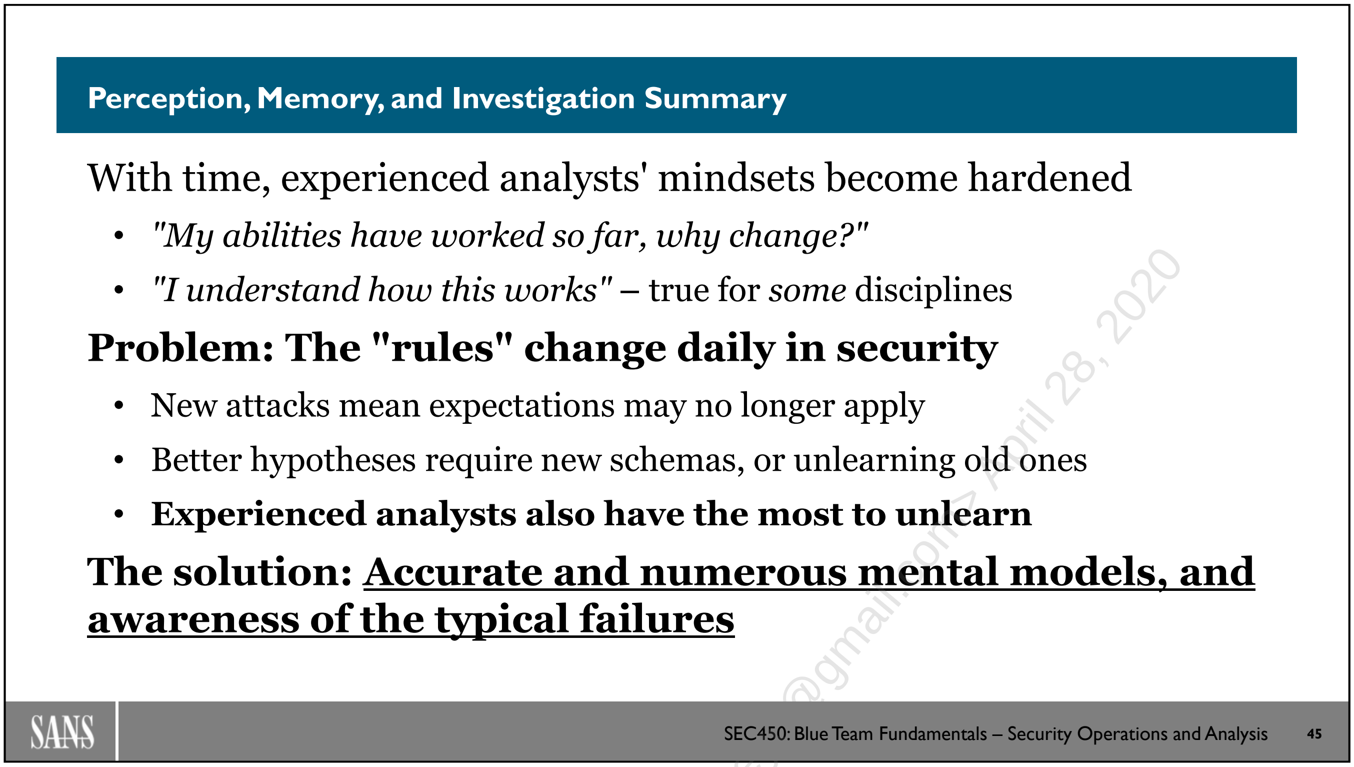
- Strong paths mean **mental ruts**
- Harder to change **perspective**
- **Creativity** can become more difficult
- "**Unlearning**" takes more effort

SEC450: Blue Team Fundamentals – Security Operations and Analysis 44

Memory and the Interplay of Experience and Creativity

So why did we dive into the inner workings of memory in a class on analysis? Because your memories, schemas, and mental models you bring into any analysis situation are all part of the filter through which all data will be perceived. Even the amount of familiarity you have with a concept may modify the hypothesis you generate. The more a similar concept and problem is encountered, the more the pathways used in the brain to process that concept will be strengthened. When pathways are strengthened, it becomes much easier to spot patterns and recognize the maliciousness in a complex situation. This strengthening happens naturally over the course of your career and is what makes the difference between a new and seasoned analyst, but speed and efficiency come with a cost.

Strengthened pathways can become so frequently used that they become the default go-to answer in a situation, even when it may be incorrect, and make it more difficult to see a situation any other way. This is called a mental rut and it is quite similar to a physical road rut in that once it has formed, it becomes painful and difficult to take any other route outside of the well-formed groove. Mental ruts make it more painful to change perspective, creativity more difficult, and "unlearning" a concept takes even more effort.



Perception, Memory, and Investigation Summary

With time, experienced analysts' mindsets become hardened

- *"My abilities have worked so far, why change?"*
- *"I understand how this works"* – true for *some* disciplines

Problem: The "rules" change daily in security

- New attacks mean expectations may no longer apply
- Better hypotheses require new schemas, or unlearning old ones
- **Experienced analysts also have the most to unlearn**

The solution: Accurate and numerous mental models, and awareness of the typical failures

SANS

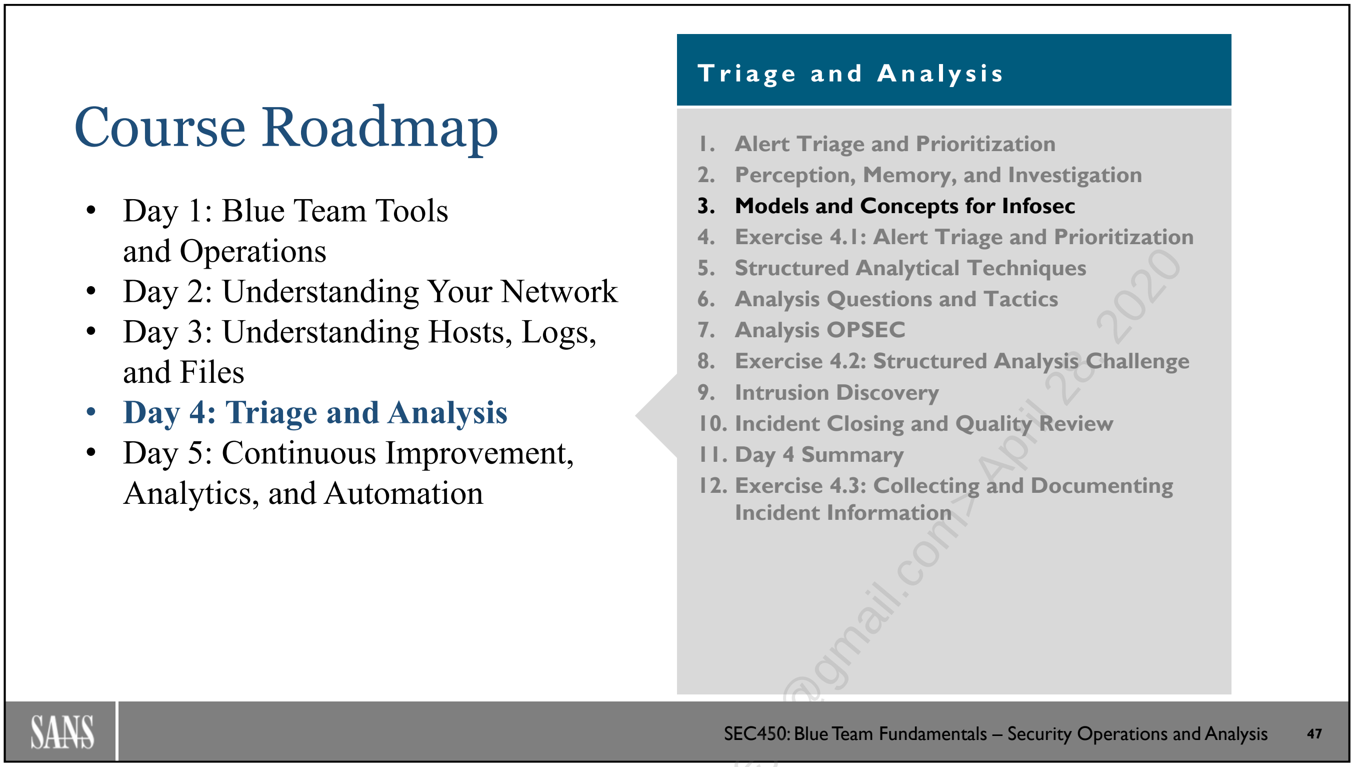
SEC450: Blue Team Fundamentals – Security Operations and Analysis 45

Perception, Memory, and Investigation Summary

Understanding memory gives us a better chance of knowing what drives whether we commit information to long-term memory or not, and the most efficient way to do so, but there are also some pitfalls to be aware of. Although it is an amazing thing when a person is so familiar with a topic it appears to be second nature to them, Heuer points out that there can be pitfalls associated with this level of familiarity for analysts. One issue is that although much training is focused on opening the minds of students, more experienced students tend to be set in their ways, confident in their abilities that have served them well so far. In this respect, they are likely to be mostly correct, but in intelligence analysis and information security in general, the rules are constantly changing. We face new threats and new capabilities from those threats each day, and this prospect means that what we've learned and the schemas we've formed in the past may not continue to apply today or into the future. In this situation, producing a better hypothesis or conclusion may require continuing to form new schemas, or unlearn ones that were previously known, a much harder task. In addition, if new information comes to light in the later stages of an investigation, people tend to not go back and reassess the significance of previous evidence considering the new data. Once a conclusion has been formed that a piece of data doesn't fit with the current hypothesis, it becomes unlikely that it will be re-evaluated and reintroduced given additional information later that may make it relevant again. This naturally can lead to analysts coming to the wrong conclusion.

In summary, why do we cover how memory works before diving into technical material? Because it has a direct effect on your analytical capability, and the development of schemas is what can bring you from a novice to an experienced analyst. We can start to build these schemas quickly if we strive to associate the new material we are learning with already understood concepts. Since memory is formed of many interconnected ideas, having a "connection point" to an already existing form boosts the ability to commit information to long-term memory. When it comes to building long-term memory, processing helps as well. Using artificial structures such as mnemonic devices and problem modeling through decomposition and externalization helps us better understand problems and overcome the hurdles of the limited number of items we can keep in working memory. For this reason, the following sections of this book will focus on established mental models of multiple information security concepts. Modeling items such as infection chains, attack and defense concepts, and threat intelligence cycles can tie them to existing schemas and allow us to exploit the nature of our own mind to rapidly commit them to memory where they can aid in analysis.

[1] Heuer, 1999, pp. 29-30



Course Roadmap

- Day 1: Blue Team Tools and Operations
- Day 2: Understanding Your Network
- Day 3: Understanding Hosts, Logs, and Files
- **Day 4: Triage and Analysis**
- Day 5: Continuous Improvement, Analytics, and Automation

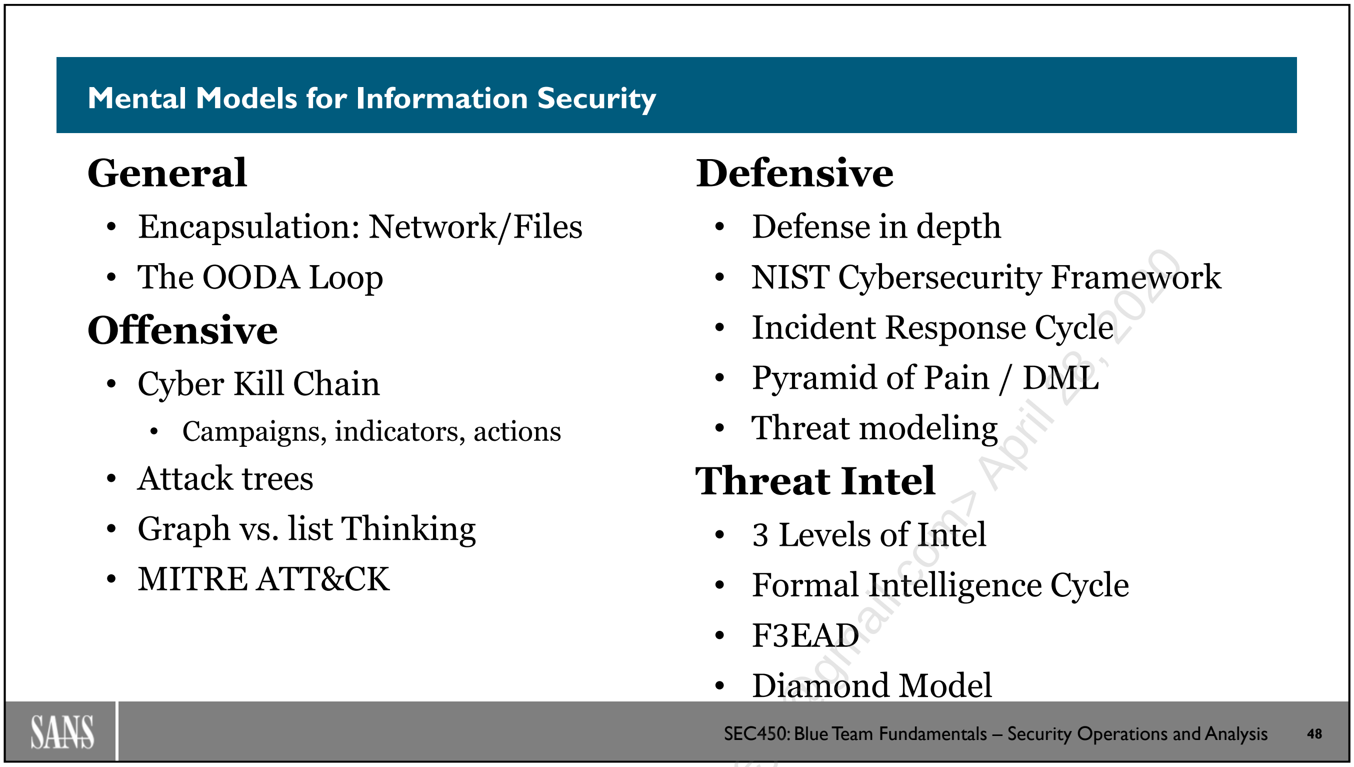
Triage and Analysis

1. Alert Triage and Prioritization
2. Perception, Memory, and Investigation
3. **Models and Concepts for Infosec**
4. Exercise 4.1: Alert Triage and Prioritization
5. Structured Analytical Techniques
6. Analysis Questions and Tactics
7. Analysis OPSEC
8. Exercise 4.2: Structured Analysis Challenge
9. Intrusion Discovery
10. Incident Closing and Quality Review
11. Day 4 Summary
12. Exercise 4.3: Collecting and Documenting Incident Information

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 47

This page intentionally left blank.



Mental Models for Information Security

General

- Encapsulation: Network/Files
- The OODA Loop

Offensive

- Cyber Kill Chain
  - Campaigns, indicators, actions
- Attack trees
- Graph vs. list Thinking
- MITRE ATT&CK

Defensive

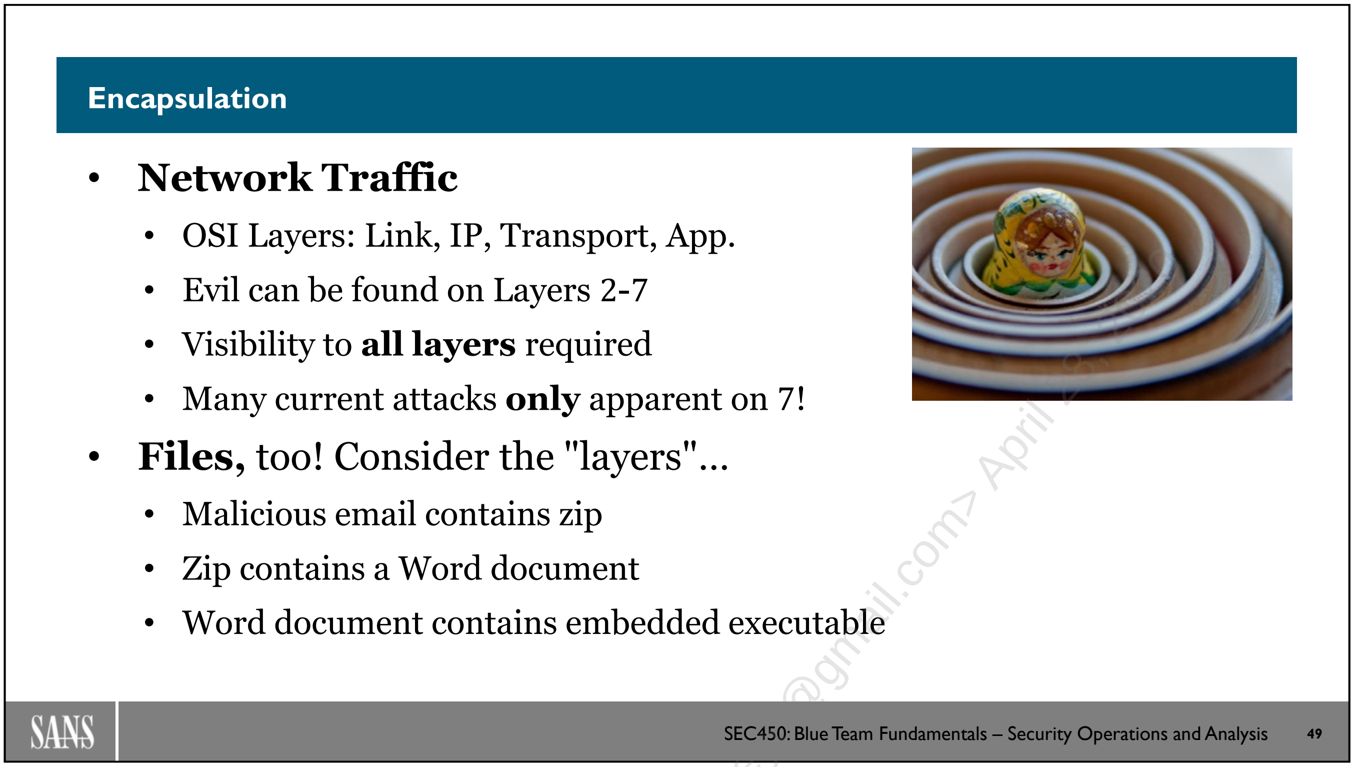
- Defense in depth
- NIST Cybersecurity Framework
- Incident Response Cycle
- Pyramid of Pain / DML
- Threat modeling

Threat Intel

- 3 Levels of Intel
- Formal Intelligence Cycle
- F3EAD
- Diamond Model

Mental Models for Information Security

Although our mindsets for specific attacks must constantly be changing, there are plenty of models that describe attack, defense, and threat intel in high-level ways that can help us understand them. In this module, we'll do a brief review of the most common information security high-level models and show how they can be useful as a tool in our everyday lives. Some of these are general concepts, and some are targeted toward understanding specific processes or options in defense, offense, or intelligence. Each of these mental models brings something unique to the table and understanding them will enrich your ability to dissect attacks and quickly come to a conclusion on the next required action.



Encapsulation

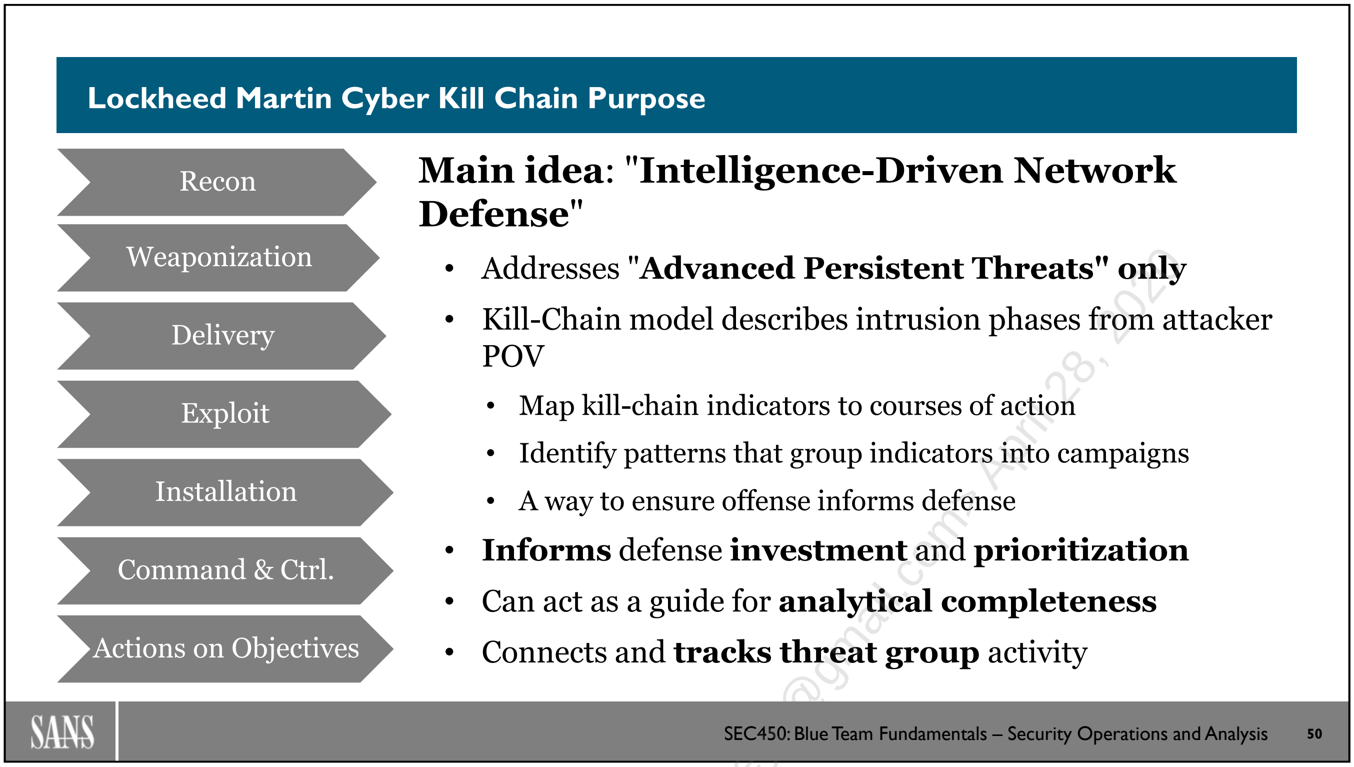
- **Network Traffic**
  - OSI Layers: Link, IP, Transport, App.
  - Evil can be found on Layers 2-7
  - Visibility to **all layers** required
  - Many current attacks **only** apparent on 7!
- **Files,** too! Consider the "layers"...
  - Malicious email contains zip
  - Zip contains a Word document
  - Word document contains embedded executable

Encapsulation

When it comes to attacks, there are generally multiple "layers" of abstraction that must be considered to understand and identify them. For network traffic, this could mean identifying malicious behavior on any of the layers of the network stack. Layer 2-7 attacks are common, with many recent attacks only being identifiable with the ability to identify Layer 7 content.

For example, imagine a malicious email that comes in with a link to a malicious file hosted on a OneDrive (a common occurrence). If the user clicks on it, they will be connecting to a website that is generally seen as not a threat using HTTP(S) protocol, which is also totally normal. This attack will not be identifiable from network traffic looking at the IP address or port of the connection, nor the domain name the user is talking to. It is only if we can see the traffic (potentially requiring SSL decryption) to identify the file download with the Layer 7 HTTP content that we would know that a malicious file was downloaded.

Attacks with files can be thought of in layers as well. Imagine the same email, but with the file directly attached the email this time. If the email comes in with a zip file inside, scanning the zip may or may not reveal anything malicious. If we were, however, able to open the zip, see that it contains a word document, and then open the word document to find it contains an executable file, the attack may become more apparent. The lesson is that just because a file or network transaction looks innocent on the surface, that does *not* mean that digging deeper would not reveal that its true nature is malicious. In many cases, we must recursively take the data we have apart and get "all the way to the bottom" before we can be certain that something malicious isn't occurring.



Lockheed Martin Cyber Kill Chain Purpose

Recon
Weaponization
Delivery
Exploit
Installation
Command & Ctrl.
Actions on Objectives

Main idea: **"Intelligence-Driven Network Defense"**

- Addresses "**Advanced Persistent Threats" only**
- Kill-Chain model describes intrusion phases from attacker POV
  - Map kill-chain indicators to courses of action
  - Identify patterns that group indicators into campaigns
  - A way to ensure offense informs defense
- **Informs** defense **investment** and **prioritization**
- Can act as a guide for **analytical completeness**
- Connects and **tracks threat group** activity

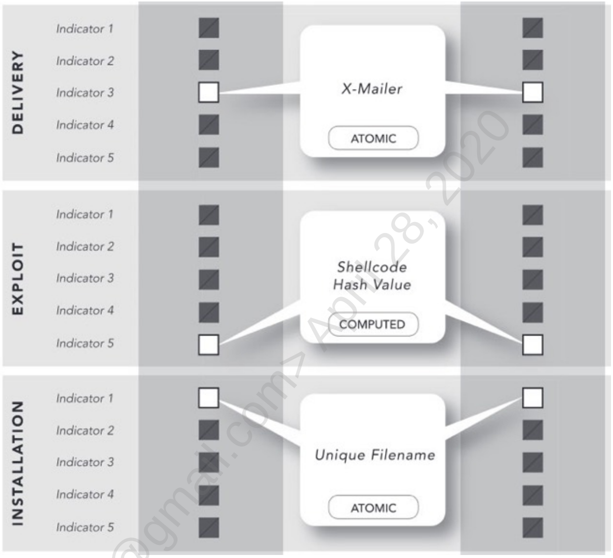
Lockheed Martin Cyber Kill Chain Purpose

The goal of the Cyber Kill Chain is to enumerate all the steps that must be taken during an "APT"-style intrusion. It points out the fact that every step of the chart must occur for adversary success, and if the chain is broken at any point (preferably as early as possible), the attacker will not reach their goal. The Kill Chain gives defenders a model to chart their various defensive controls, preventions, and detections against, and ideally find that they have multiple layers of defense that align to each step in the framework. A network well instrumented for defense in depth would have coverage across all Kill Chain steps and prove a very tough environment for an adversary.

Another important thing to remember about the Cyber Kill Chain is that it gives us a model for a complete intrusion that can be utilized during analysis. When an investigation is picked up starting at any point in the chain, an attempt should be made to block items from all stages, as well as understand and synthesize what happened before and after the point the attack is currently at. If a command and control beacon is found, a complete investigation should include what happened during stage 1-5, and what would've happened for the Actions on Objectives stage were the attack been allowed to continue. We'll get back to this concept later.

Campaign Analysis[1]

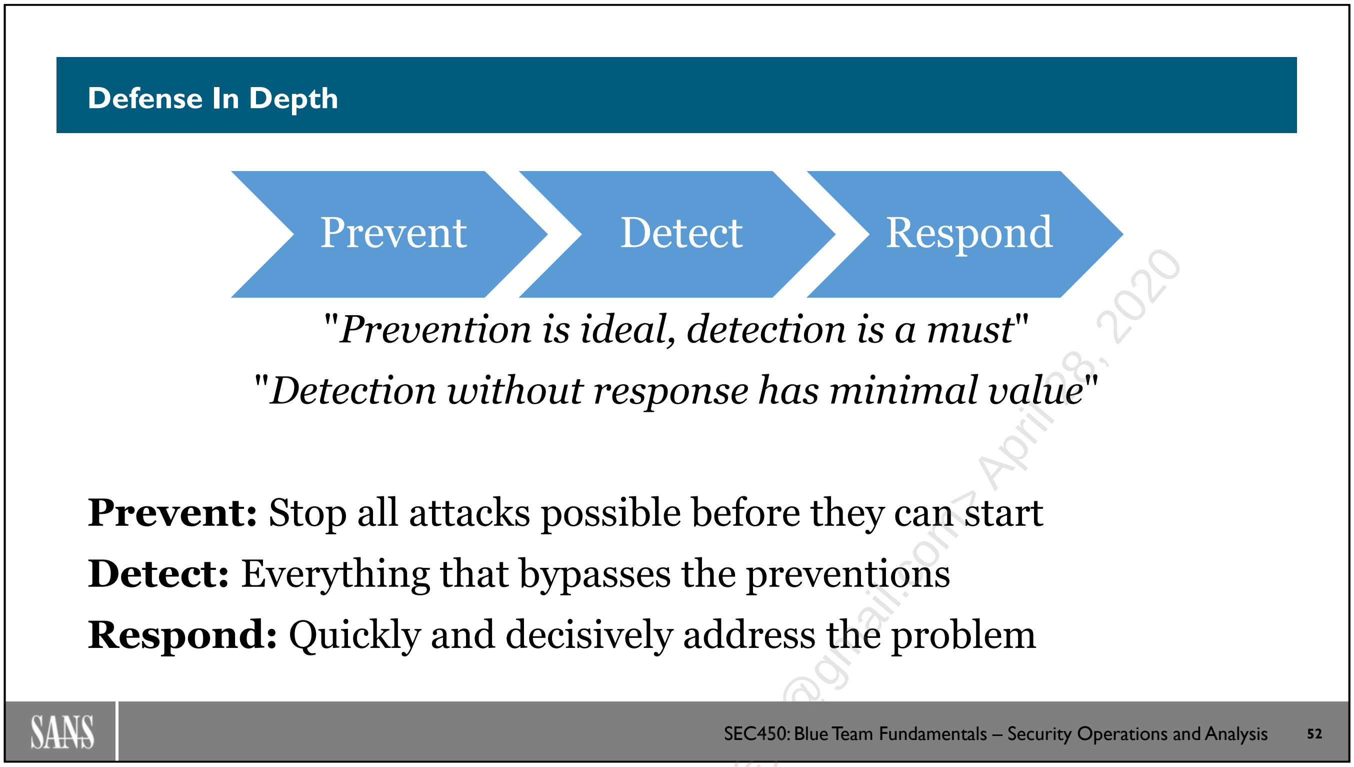
- Track all IOCs across incidents
- Identify commonalities across multiple intrusion chains
- Arrange actions from each actor into attack *campaigns*
- Attribution may be a bonus
- **Goal**: Define attacker TTPs and intent, disrupt with best courses of action



Campaign Analysis

Another contribution in the Cyber Kill Chain paper is the idea of tracking *campaigns*—the totality of the attacker's action as seen in your environment over time. This is done via tracking the indicators and TTPs associated with each individual incident that you experience and laying them out across the kill chain to connect them with other incidents. The idea is that although some indicators that are more volatile such as hashes and IP addresses may change, it is unlikely that ALL indicators and TTPs seen will be changed throughout each incident. When attackers fail to change everything about their tools for each of the attacks, you will be able to associate one action perpetrated by them with another down the road, giving you an idea of how they operate and what they might be interested in. This knowledge can be used as a strategic advantage to feed into the choice of what course of action to take in order to block the activity. You may even be able to use the pattern in combination with other open-source intel to provide attribution to the attacks you see.

[1] https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf



Defense In Depth

Prevent
Detect
Respond

"Prevention is ideal, detection is a must"

"Detection without response has minimal value"

Prevent: Stop all attacks possible before they can start

Detect: Everything that bypasses the preventions

Respond: Quickly and decisively address the problem

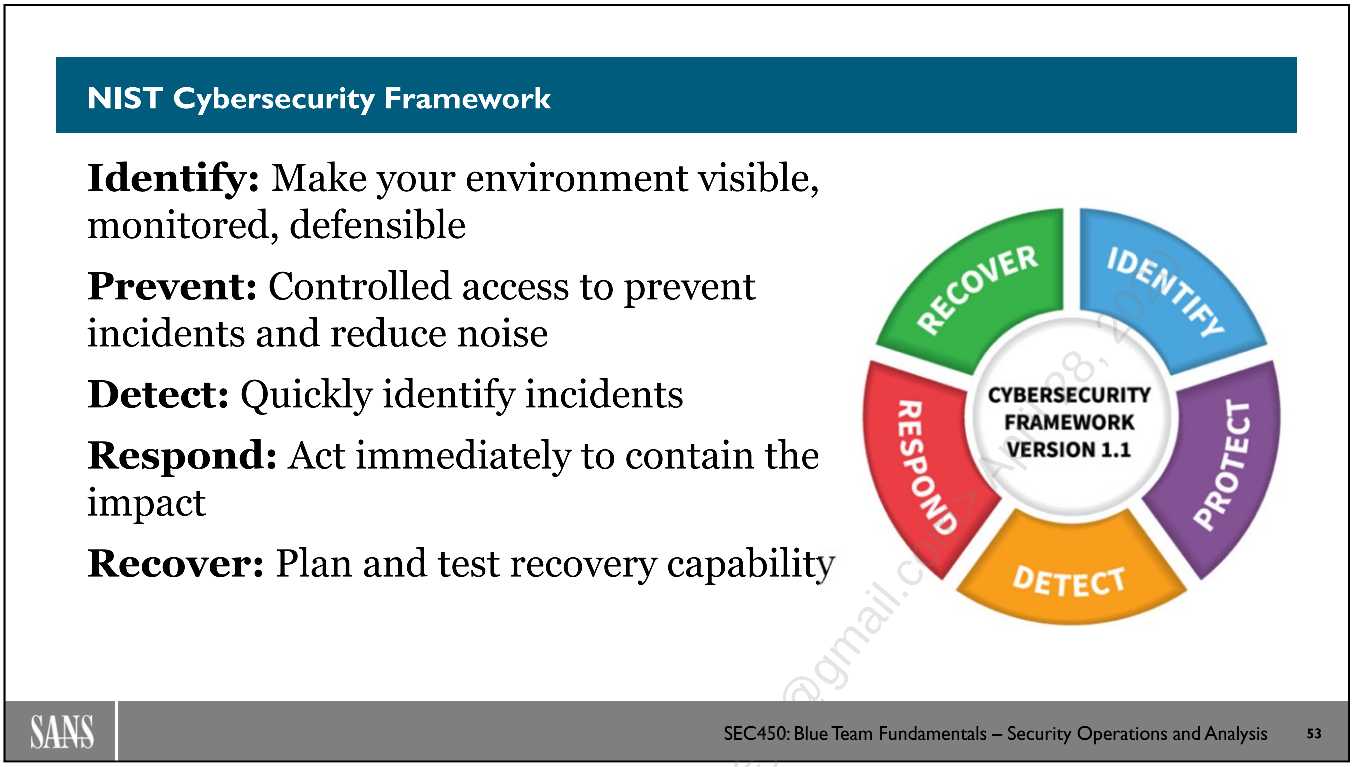
SANS
SEC450: Blue Team Fundamentals – Security Operations and Analysis
52

Defense In Depth

One of very the high-level models of defense in depth process (that SEC401 alumni should know well) is "Prevent, Detect, Respond." This model calls out the necessary parts of a well-rounded defense:

- **Prevent:** All attacks should be stopped as early as possible in the kill chain. Ideally firewalls, antivirus, IPS, or any other active control can take away as much noise as possible in an automatic way so that defenders don't even have to deal with it.
- **Detect:** As we know, no defense is perfect, "compromise is inevitable" remember? Therefore, anything that happens to get by prevention at least needs to be able to be detected. The conversation about IDS vs. IPS makes clear that some attacker techniques are just too low fidelity to apply a flat-out block and therefore are the realm of detection alerts. These alerts must exist as a backup for the things we know will slip by our prevention controls.
- **Respond:** Once an issue has been detected, we must respond. The key part of this is balancing the prevention and detection tools with the ability to execute the response. "Detection without response has minimal value." If you have so many alerts your team is flooded and cannot respond within a reasonable amount of time, you're becoming self-defeating. All that detection capability is for nothing if there is a lack of response.

In addition, you may see this model listed with a step one of "Identify" and a last step of "Recover." Identify speaks to architecting your network in a defensible way, while Recover refers to the ability to return to normalcy in a predictable way after an incident. Both are also obviously important pieces of the whole picture as well.



NIST Cybersecurity Framework

Identify: Make your environment visible, monitored, defensible

Prevent: Controlled access to prevent incidents and reduce noise

Detect: Quickly identify incidents

Respond: Act immediately to contain the impact

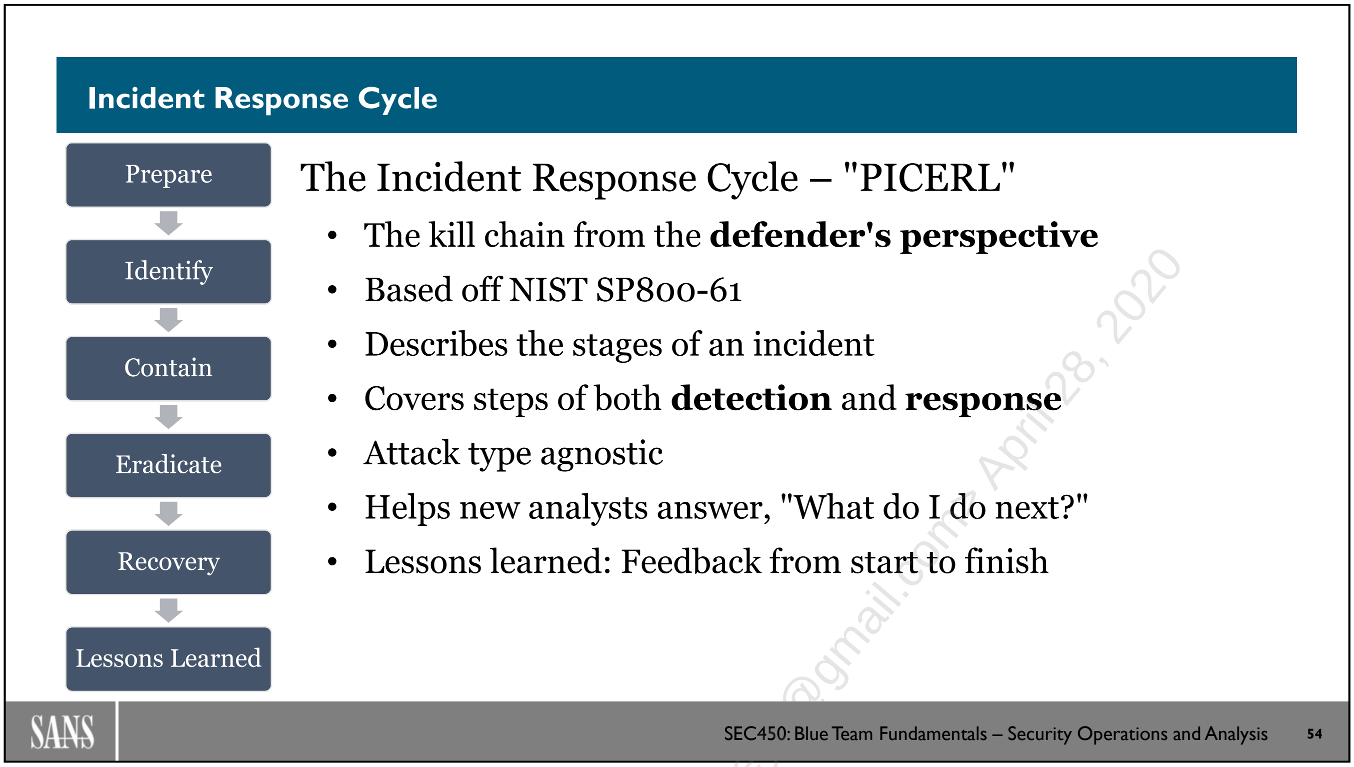
Recover: Plan and test recovery capability

NIST Cybersecurity Framework

The NIST Cybersecurity Framework goes into a bit more detail and adds in the "Identify" and "Recover" step which are assumed in the SANS SEC401 defense-in-depth model on the previous page.[1] In this case, by Identify, NIST means architecting your network for defensibility from the start. If you don't construct a network that can be monitored, preventing and detecting things will be difficult from the start.

Recovery also gets a well-deserved specific shout out in this model. You may technically have backup and recovery solutions, but have you tested to ensure they work? Have you done any structured brainstorming about what scenarios might come to pass that these would fail to account for? The goal of this framework is to show the capabilities you should have in place to make a well-rounded cyber defense operation but is less focused on the actual chronological order of events of an incident, such as the framework we will investigate next.

[1] https://www.nist.gov/cyberframework



Incident Response Cycle

Prepare

Identify

Contain

Eradicate

Recovery

Lessons Learned

The Incident Response Cycle – "PICERL"

- The kill chain from the **defender's perspective**
- Based off NIST SP800-61
- Describes the stages of an incident
- Covers steps of both **detection** and **response**
- Attack type agnostic
- Helps new analysts answer, "What do I do next?"
- Lessons learned: Feedback from start to finish

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 54

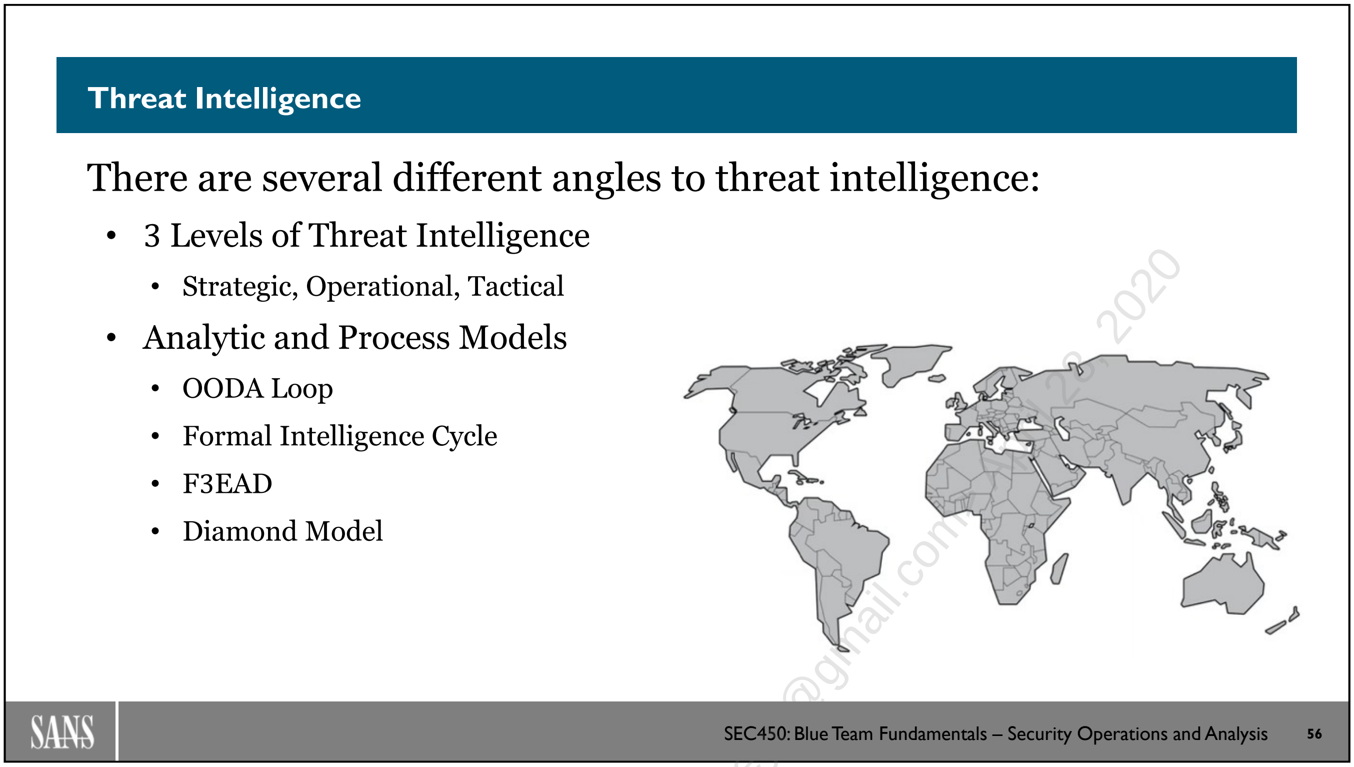
Incident Response Cycle

The incident response cycle is like the defense in depth and NIST Cybersecurity framework model, but is more incident focused. It can most interestingly be described as the same thing as the kill chain but viewed from the defender's side. This cycle is drawn from the NIST SP800-61 Computer Security Incident Handling Guide and is meant to describe the stages of both incident detection and response. Unlike the kill chain, however, the incident response cycle is attack type agnostic, aiming to apply in general whether it is an opportunistic or targeted attack. This model can be a great mental framework for newer analysts to answer the question, "what do I do first", when an incident is occurring. By placing yourself somewhere in the cycle, the incident response cycle steps can be a great guide on "what to do next."

The description of the stages are as follows:

- Prepare: This stage happens before the attack even begins. Preparation is about architecting your detection systems, data collection, and signatures to understand what is happening on the network at any given moment. In a nutshell, it largely encompasses the processes of setting up NSM and CSM.
- Identify: The identification stages occur once defenders sense in any way that an attack is in progress on their network. This phase is all about incident detection and covers up until the point where actual active actions are taken as a response. One of the key goals of threat intelligence operations is to aid this step in being more informed and thorough in response.
- Contain: This stage is about taking the first active steps to disrupt the attacker, likely in a short-term way, as well as starting to plan your longer-term response. You can think of this stage as "stopping the bleeding" so to speak. The goal here is to, at a minimum, gain control of the situation and make sure things can't get any worse. Containment strategy is a topic for another slide but be aware that containment for opportunistic and targeted attacks may vary wildly.
- Eradicate: Longer-term mitigation efforts should start to take effect in the eradication phase. This is where you don't just momentarily disrupt their attack but put the actions in place to lock them out and keep them out. Strategies for eradication vary just as much as containment strategies and the one used in any given situation will highly depend on the nature of the incident and how much risk you are willing to take that you missed something (full host wipe vs. surgical removal).

- Recovery: Recovery takes the incident from the state the incident responders had to create for eradication back to the pre-compromise state of operation. If extra drastic measures have to be taken, such as bringing systems offline or otherwise disrupting operations, this stage is about bringing those things back to normal functioning.
- Lessons Learned: One of the most important stages—the lessons learned phase ensures there is feedback (which is not pictured on the slide) from the end of the cycle back to the preparation stage. Without this feedback mechanism, our systems would not become increasingly hardened over time and the same thing could affect us repeatedly. Questions answered should minimally include "what went well" and "what didn't go well", "what can we do better", and "what will we do differently next time." Although there is much desired to "get back to your normal job" after an incident occurs and skip this step, is a vital piece of the cycle that should be performed as quickly as possible so people don't forget the minor details of how the incident ran and how it could be improved in the future.



Threat Intelligence

There are several different angles to threat intelligence:

- 3 Levels of Threat Intelligence
  - Strategic, Operational, Tactical
- Analytic and Process Models
  - OODA Loop
  - Formal Intelligence Cycle
  - F3EAD
  - Diamond Model

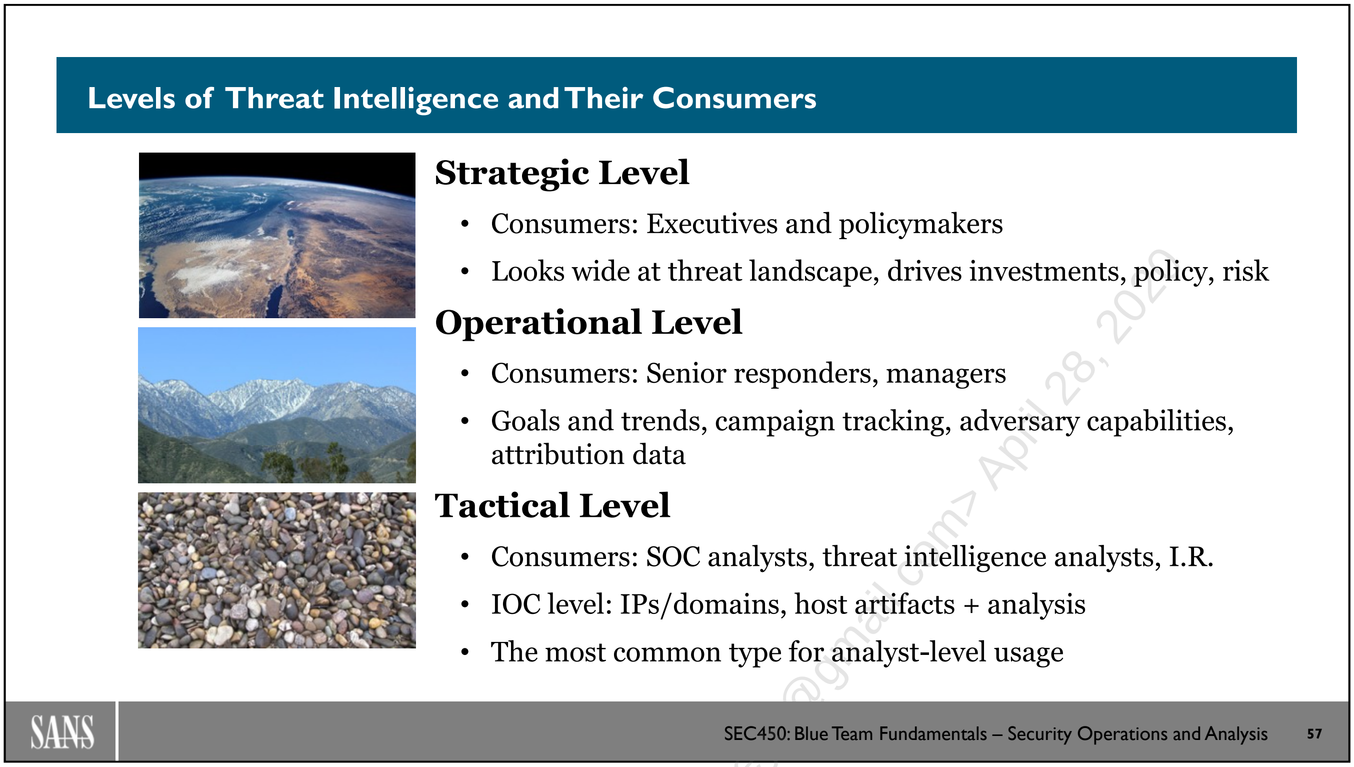
SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 56

Threat Intelligence

Threat intelligence is another huge topic within information security, and there are multiple models that are used to described different processes and goals. Although we won't go super deep on them, there are some threat intelligence frameworks that are useful to be familiar with so that you can converse with experts in the fields. The whole goal of threat intelligence is to give us a strategic and tactical advantage over our attackers through analyzing their movements and knowing TTPs. This doesn't necessarily mean only low-level indicators as many analysts believe, however. There's much more to it than that. What about how threat intelligence is analyzed and how we weave it into our everyday SOC practice? Throughout the next few slides, we'll introduce these topics and some of the frameworks you will hear about in relation to threat intelligence. For those who want to take the topic further, SANS offers the six-day "FOR578: Cyber Threat Intelligence" class.

[1] https://www.sans.org/course/cyber-threat-intelligence



Levels of Threat Intelligence and Their Consumers

Strategic Level

- Consumers: Executives and policymakers
- Looks wide at threat landscape, drives investments, policy, risk

Operational Level

- Consumers: Senior responders, managers
- Goals and trends, campaign tracking, adversary capabilities, attribution data

Tactical Level

- Consumers: SOC analysts, threat intelligence analysts, I.R.
- IOC level: IPs/domains, host artifacts + analysis
- The most common type for analyst-level usage

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 57

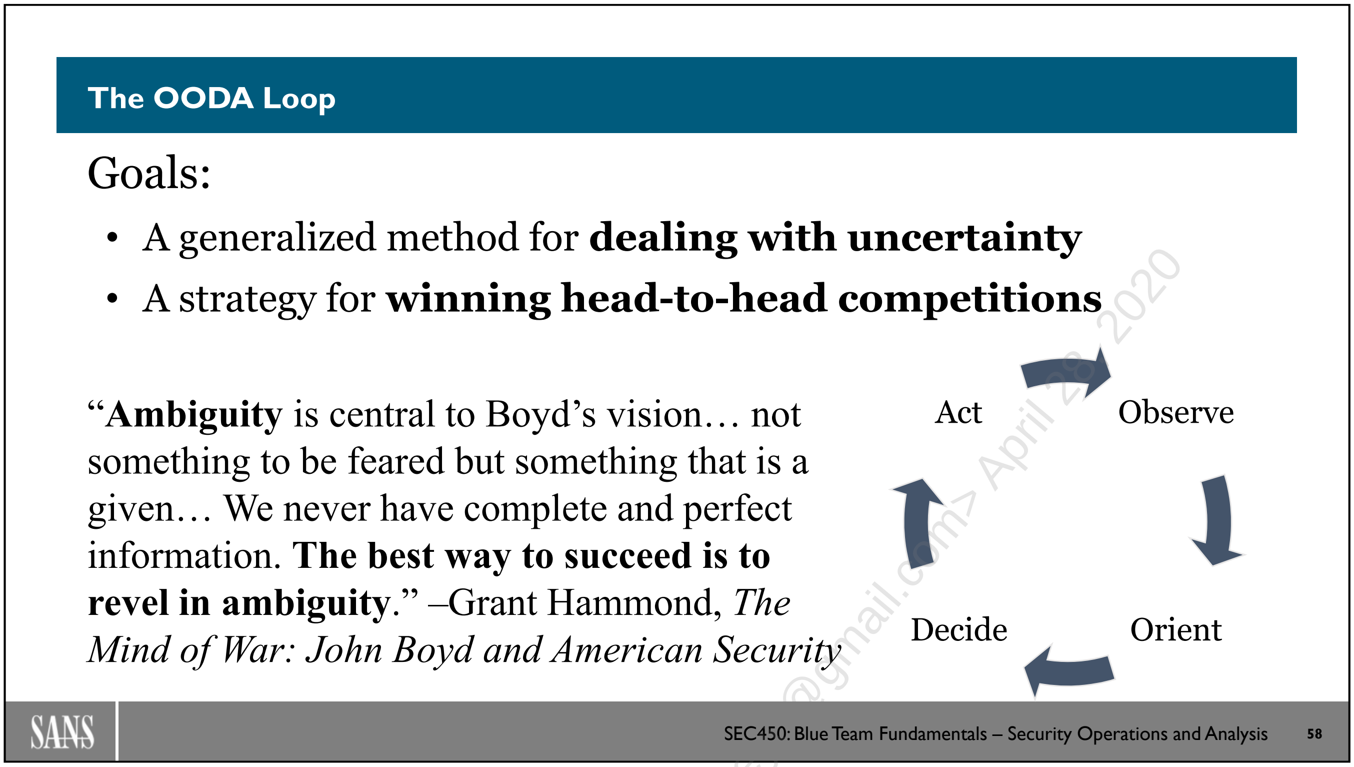
Levels of Threat Intelligence and Their Consumers

Threat intelligence information is often logically split into three types that have different purposes and typical consumers.

Tactical intelligence is the lowest level of intelligence information. It consists of highly perishable information and atomic indicators of compromise that is often used directly for security operations and incident response. This includes information such as IP addresses, domain names, and host-based artifacts. Customers for this type of intelligence include the SOC, IR teams, and threat intelligence specialists.

Operational intelligence is one step up in abstraction from the tactical level. This data supports SOC operations and tracks adversary capabilities fffon a longer scale than a single incident and consists of activities like campaign tracking, attribution, adversary capabilities and intent. It can sometimes be a little difficult to define in that it is often too high level to be tactical, but too low level to be strategic. Consumers of operational intelligence are senior level forensics and incident response personnel, as well as possible SOC managers or directors.

Strategic intelligence is the highest level of threat intelligence consisting of broad items like security strategy, risk assessments, and resource allocation. It is the type of threat intelligence that is typically the concern of, and used by the higher levels of management to set direction and policy.



The OODA Loop

Goals:

- A generalized method for **dealing with uncertainty**
- A strategy for **winning head-to-head competitions**

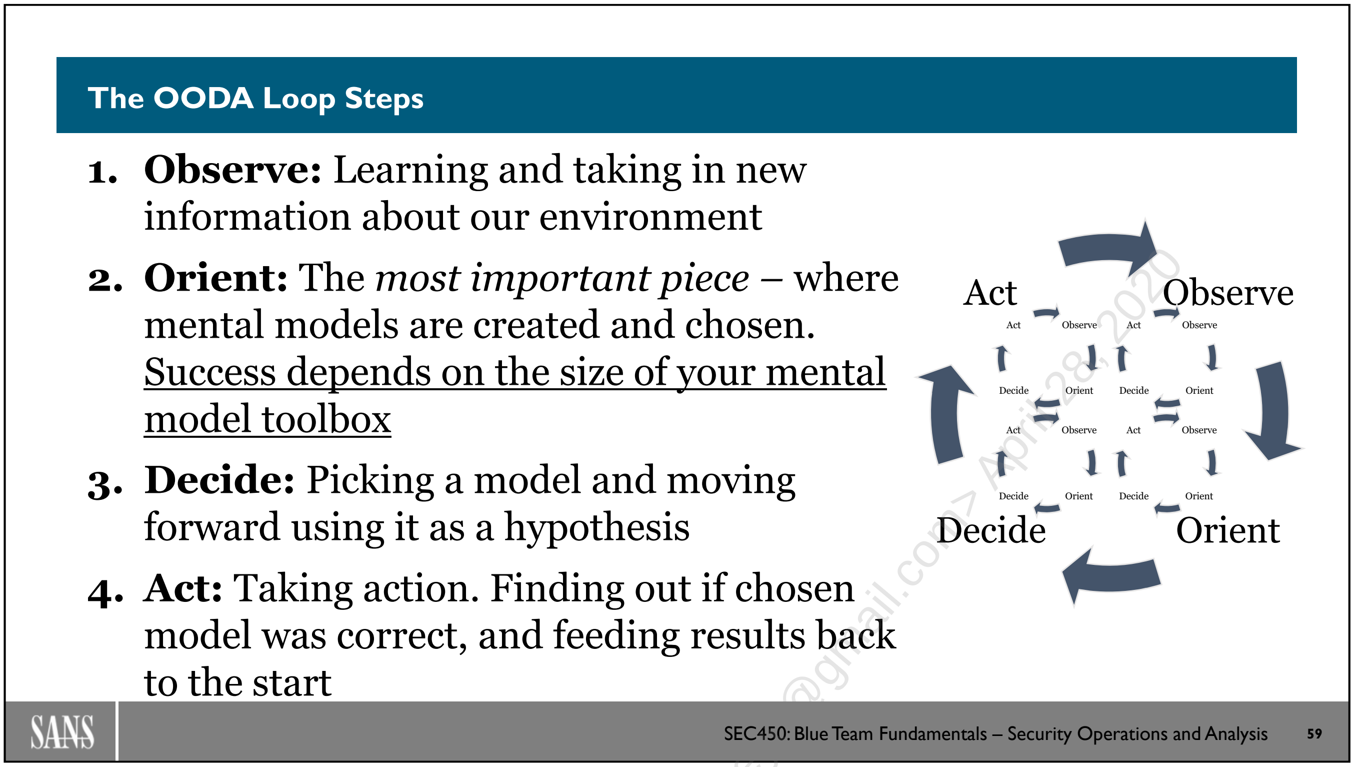
"**Ambiguity** is central to Boyd's vision… not something to be feared but something that is a given… We never have complete and perfect information. **The best way to succeed is to revel in ambiguity**." –Grant Hammond, *The Mind of War: John Boyd and American Security*

The OODA Loop

Designed by fighter pilot and military strategist John Boyd in the 1960s, the OODA or "Observe, Orient, Decide, Act" loop was a model to show how even with the disadvantage of poorer technical capabilities and with imperfect information, a fighter pilot could still win in a dogfight using quick, decisive action. Boyd imagined the loop stages running constantly in each pilot's mind and whichever pilot could iterate through the loop faster, orienting to their environment and situation and making faster decisions, would ultimately be victorious. Its lessons apply far outside of planes though. In general, the OODA loop is a learning system for dealing with uncertainty, and a strategy for winning any head-to-head competition, something that is extremely applicable to the SOC operations and incident response. In short, it calls as discussed previously, the tendency for people to not shift their mental models fast enough as circumstances change throughout the situation. This lack of clear observation leads to making suboptimal choices, and ultimately the loss of whatever competition or battle is occurring.

This history of Boyd and a deeper dive on its meaning and application to everyday life can be found in this highly-recommended in depth blog post titled "The Tao of Boyd: How to Master the OODA Loop" from Bret and Kate McKay.[1]

[1] https://www.artofmanliness.com/articles/ooda-loop/



The OODA Loop Steps

1. **Observe:** Learning and taking in new information about our environment
2. **Orient:** The *most important piece* – where mental models are created and chosen. Success depends on the size of your mental model toolbox
3. **Decide:** Picking a model and moving forward using it as a hypothesis
4. **Act:** Taking action. Finding out if chosen model was correct, and feeding results back to the start

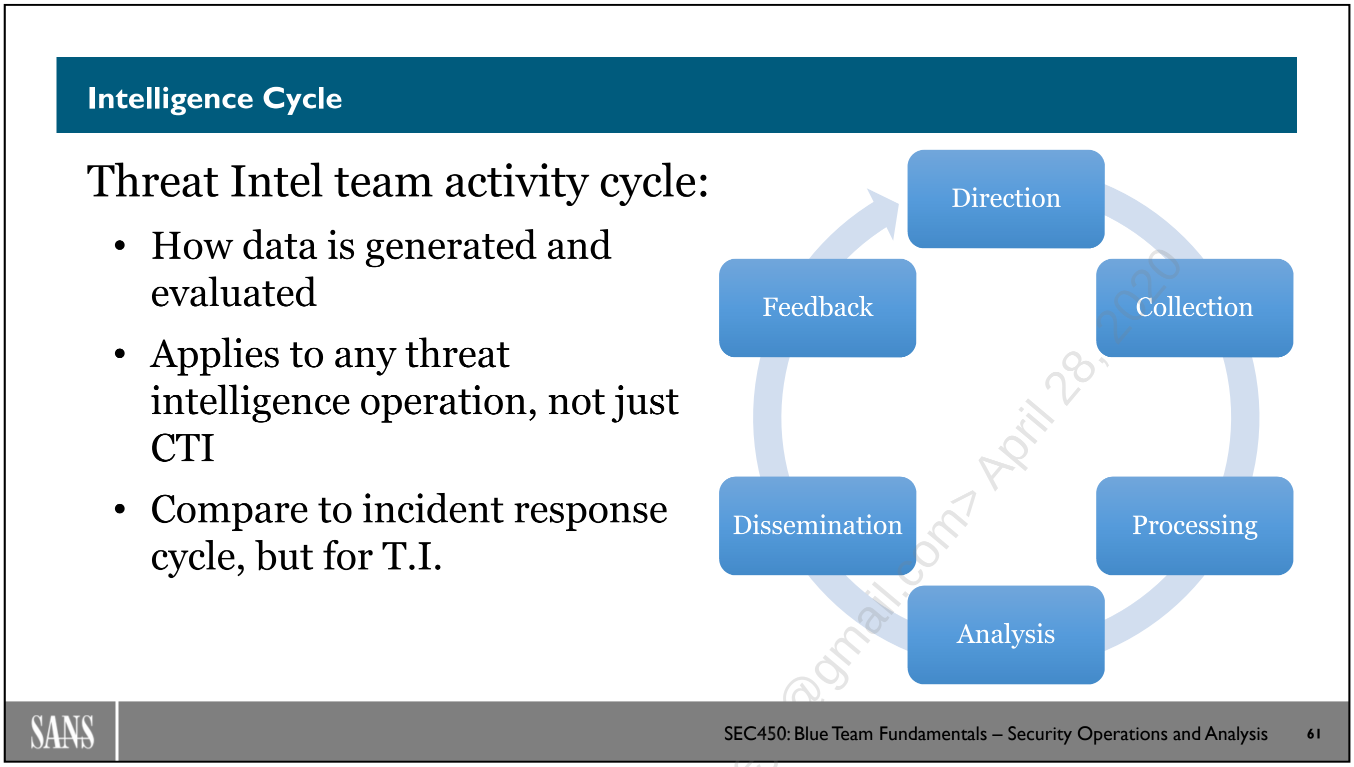
The OODA Loop Steps

The OODA loop stages can be generalized to the meanings shown on the slide above. However, when it comes to security operations, the OODA loop stages can be mapped directly to phases of threat intelligence gathering or incident response as well.[1]

- Observe: This stage is all about the collection of information that could be useful. For information security in specific, this would be NSM and CSM data—logs, files, and packets captured from the endpoints and network.
- Orient: Of all the phases, the orient phase has the biggest impact on the cycle because it is the phase where we take our previously known mental models and context of the situation and combine the newly gathered info to hopefully perceive a coherent picture of reality. Without a correct model fed by clear perception, accurate schemas and models in memory, and the willingness to update those models frequently, this is where things can fall off the rails if we are not careful. For security operations, this is taking the collected information about the attack and attacker and combining it with network and user context, and putting together a picture of the attack in progress and its goals.
- Decide: After the information has been gathered, put into context and a model of the situation created through the Observe and Orient stages, it's time to decide what to do about it. This stage is all about coming up with possible strategies to move forward given the information and context that has been supplied and picking one of them. For security operations, this is deciding on how to best disrupt the attacker such that we will be able to have an advantage over them in the coming loop iterations. If we can move faster than they can, we can stay ahead and command the situation.
- Act: The Act stage is all about following through on the course of action decided upon at the Decide stages. It also brings feedback of the action, whether successful or unsuccessful, back to the start of the loop as an input to the Observe stage. For security operations, this would be taking the action against the attacker to block, disrupt, or observe them based on the decisions from the previous stage.

The key characteristic of the OODA loop to keep in mind is the speed at which each party can run the loop. According to Boyd and as shown on the slide, if you can make your loop four times within the time the adversary can run it a single time (sometimes called being "inside" their loop), you are highly likely to be the winner in the situation. This is obviously tied with the operations tempo and capabilities of the SOC. Teams that can prevent, detect, and respond *faster* and iterate on those items are likely to be better at defense.

[1] Roberts & Brown, 2017, Chapter 2



Intelligence Cycle

Threat Intel team activity cycle:

- How data is generated and evaluated
- Applies to any threat intelligence operation, not just CTI
- Compare to incident response cycle, but for T.I.

Direction

Collection

Processing

Analysis

Dissemination

Feedback

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 61

Intelligence Cycle

Although we will not dive too deep into threat intelligence in this class beyond some of the functions, mental models, and how the group interfaces with the SOC, the formal intelligence cycle is a more specific breakdown of the activity a threat intel team undertakes to generate and evaluate data. These stages apply whether we are talking about cyber threat intelligence or threat intelligence in general. Reference for the steps can be found on the CIA website.[1] The intelligence cycle can be compared to the incident response cycle that we saw earlier, encompassing data gathering, analysis, taking action and feedback, but for threat intelligence products instead of incident response.

The steps are as follows:

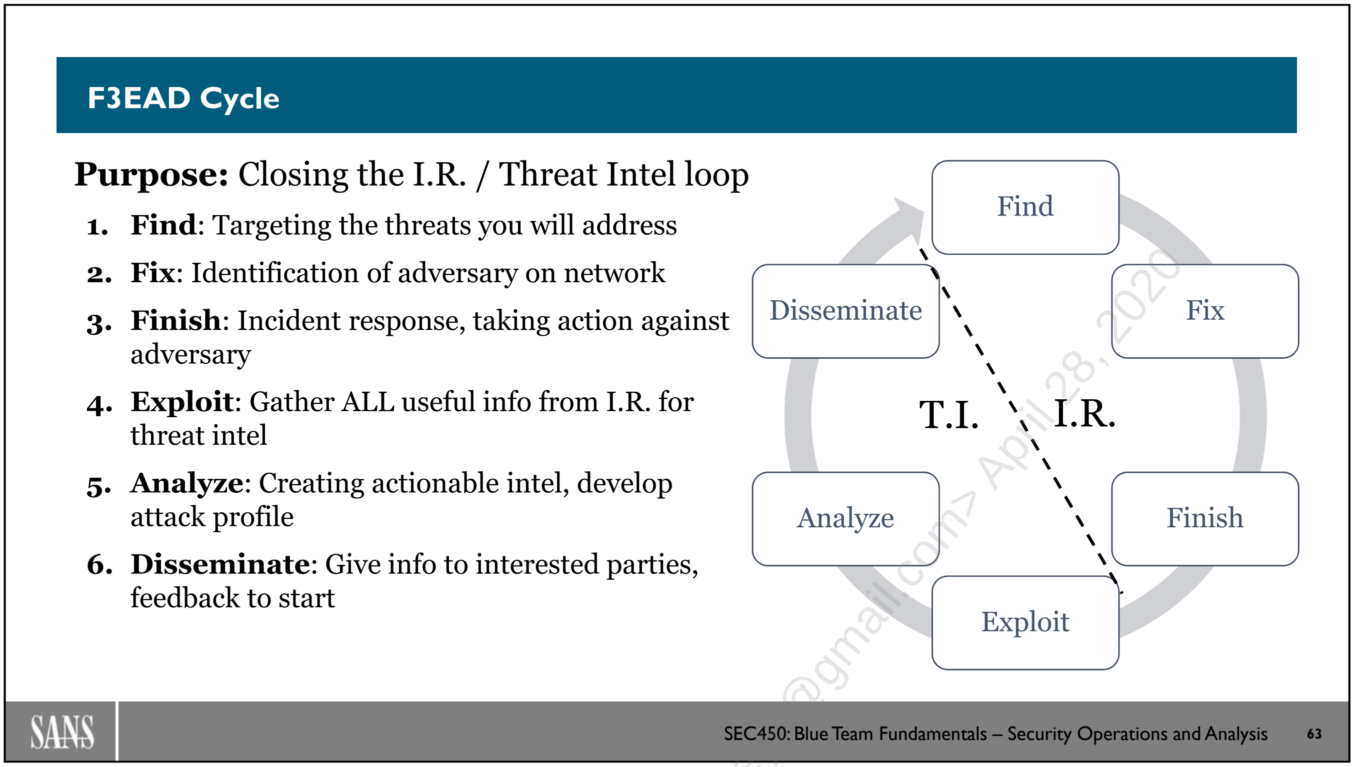
- Direction: Deciding the question that the threat intelligence is meant to answer.
- Collection: Gathering of the information needed to answer the question. Whereas for NSM and CSM, we might want to de-duplicate data, for threat intel, this may include overlapping data for corroboration purposes.
- Processing: Cleaning up the data and making it organized and usable. Many of the tasks we must perform for logs show up here, too—normalization, indexing, filtering, enrichment, etc., can all be a part of making the data workable.
- Analysis: The main goal of this section is to answer the questions the Direction stage defined as the goal. This is where the ideas and principles of structured analysis we learn in the book should be implemented by the threat intel team as well, likely to an even larger and more stringent degree. Things like listing key assumptions and information gaps can be critical to producing an analytic product that can be trusted and understood.
- Dissemination: This is one of the most important pieces. Intelligence that is generated is just as useless as detection without response. The work is done so that the conclusions can be acted upon. The dissemination step is all about getting the results into the hands of the people that need it.

- Feedback: This is feedback from the receiver of the intelligence as to whether the question set in the Direction phase was successfully answered or not. If it has not been, the team may need to go back and perform additional analysis or even more data collection until the consumer is satisfied with the results.[2]

As you can see, threat intel teams do much more work than just collecting indicators of compromise and putting them into our security tools. They must operate on three different levels of data and ensure that all their analysis is logically sound and meets the requirements of the request given to them. It is a complicated and specialized job within information security, which is why this role is often split into different team members instead of trying to have the SOC analyst team or IR team do it all.

[1] https://www.cia.gov/kids-page/6-12th-grade/who-we-are-what-we-do/the-intelligence-cycle.html

[2] Roberts & Brown, 2017, Chapter 2



F3EAD Cycle

Purpose: Closing the I.R. / Threat Intel loop

1. **Find**: Targeting the threats you will address
2. **Fix**: Identification of adversary on network
3. **Finish**: Incident response, taking action against adversary
4. **Exploit**: Gather ALL useful info from I.R. for threat intel
5. **Analyze**: Creating actionable intel, develop attack profile
6. **Disseminate**: Give info to interested parties, feedback to start

F3EAD

The F3EAD acronym stands for Find, Fix, Finish, Exploit, Analyze, Disseminate and was born as a targeted methodology for special operations teams in the armed forces. It is different from the OODA loop and formal intelligence cycle models in that it is meant to address two specific things: That intelligence should lead to meaningful changes and improvements in operations, and that the operations and intelligence cycles should not just feed back into themselves, but into each other as well.[1]

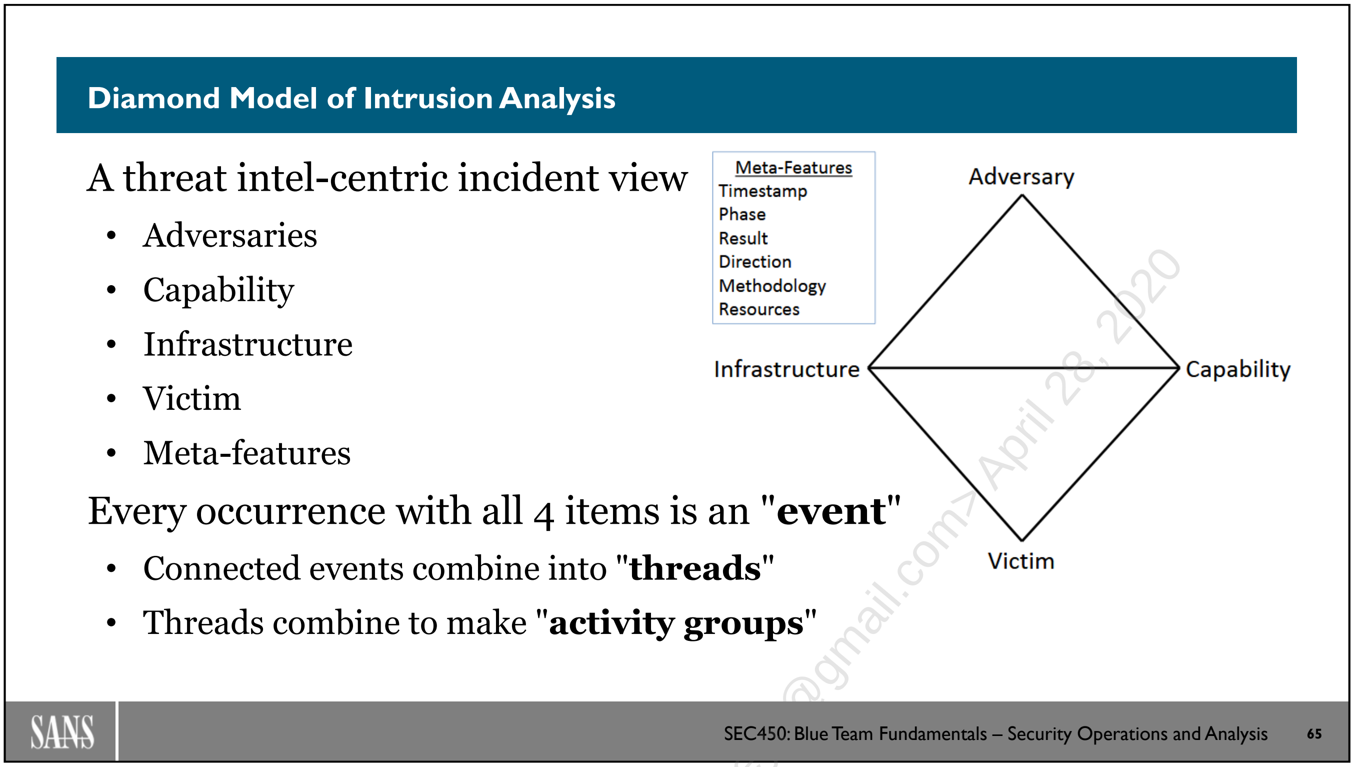
The way that this works is through the steps listed on the slide, which are a mashup of both the incident response cycles and the formal intelligence cycle where each one ends with an input to the other, making this a virtuous cycle of incident response feeding threat intelligence and vice versa. The F3EAD cycle bridges the gap between the two groups and is *one of the best high-level views of how security operations and threat intelligence groups should work hand in hand* to improve each other in a continuous manner.

Since the naming convention is a bit different, compacted into fewer steps, and slightly more confusing due to the origin of the cycle step names, here is how you should interpret each step:

- Find: The targeting step, what will your adversary try to do to you? What are the threats you are facing? This should come from the outputs of previous incidents as well as analysis from the threat intelligence team.
- Fix: This term can be confusing. It is "fix" as in "get a fix on the target", not fix an infected machine. This stage can be compared to the "Identify/Detection" stages of other frameworks.
- Finish: This is like the contain/eradicate/remediate stages in the PICERL model in that it is the active and final portion of the incident response piece of the cycle before.
- Exploit: Again, the term exploit is used very differently here. In this model, we are "exploiting" the information we gained from doing the incident response actions to feed into the threat intelligence portion of the cycle. It is similar to the collection phase of typical threat intelligence models.

- Analyze: This is the analysis step where we take the raw data from the previous collection focus stage and extract the possible conclusions about what the attacker is doing and what they might do in the future.
- Disseminate: Distribution of the tactical, operational and strategic-level intelligence and closing the loop back to the security operations and incident response teams. This is where all the new information produced through analysis of incident data makes it back into new collection and detection methods for the environment.[1]

[1] Roberts & Brown, 2017, Chapter 2



Diamond Model of Intrusion Analysis

A threat intel-centric incident view

- Adversaries
- Capability
- Infrastructure
- Victim
- Meta-features

Every occurrence with all 4 items is an "**event**"

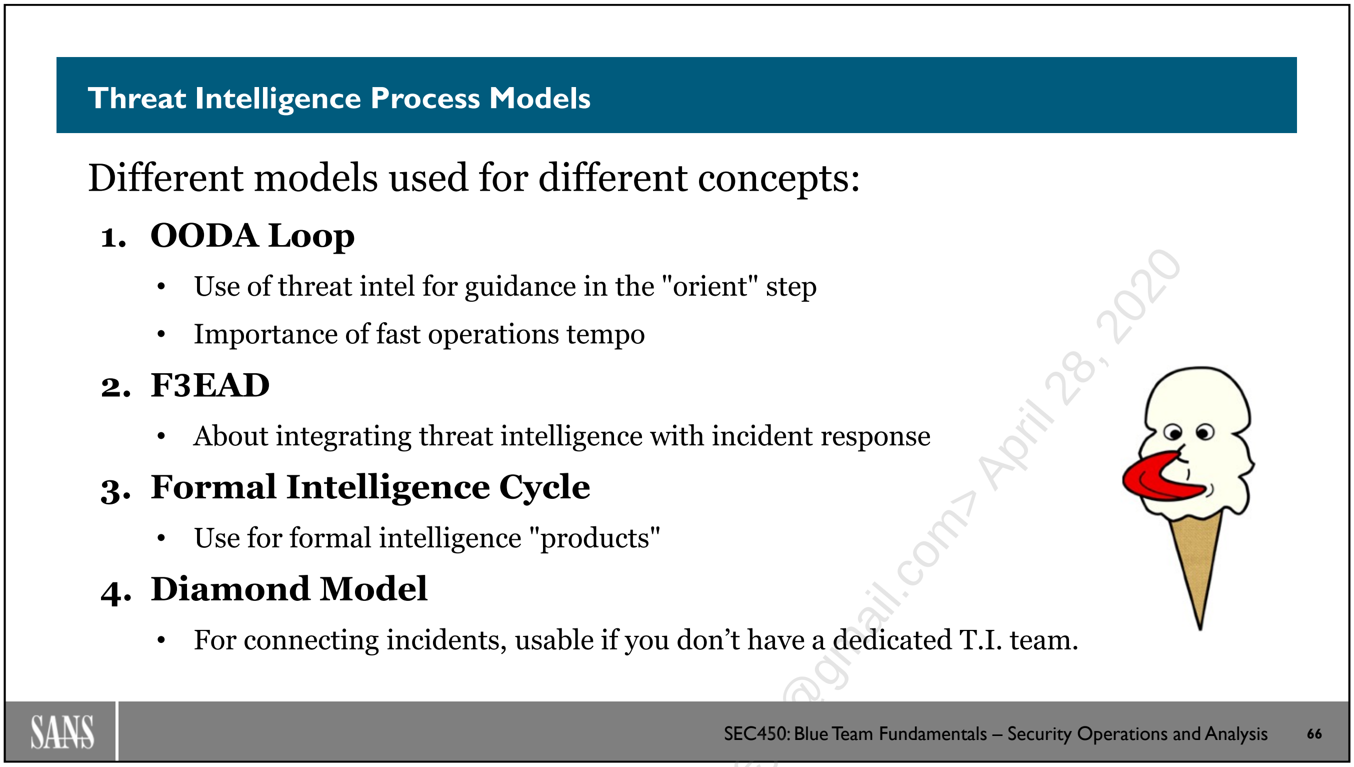
- Connected events combine into "**threads**"
- Threads combine to make "**activity groups**"

Diamond Model of Intrusion Analysis

Do you ever wonder how threat intelligence analysts mentally model an incident and if it is different from the typical analyst? The Diamond Model of Intrusion Analysis is one popular way to look at incidents in a threat intelligence-centric way that complements the kill chain and is also be helpful for clarifying the variables analysts must consider and deal with. The model was originally designed in a paper by Christopher Betz, Andrew Pendergast, and Sergio Caltigirone and describes *events,* or how and *adversary* deploys a *capability* over some *infrastructure* against a *victim* to form *activity threads.*[1] Each instance of an activity with all four items creates an event, events go into threads, and threads can be collected into activity groups that can be grouped based on these features, or any of the meta-features that may link the two items.

Analyzing common victims, infrastructure, or capabilities may assist with attribution to a common attacker and can guide the threat intelligence team on how to best align defense against that attacker in the future. The model aligns well with the kill chain in that each event can be assigned a kill-chain stage to view the threads and activity groups on the attack in totality, and when compared against other incidents, overlap may become apparent through one of the features that would not have been obvious if it weren't for this style of analysis. Focusing on the "north-south" access of the diamond can enumerate socio-political-based motivations, and aspirations of the attacker whereas "east-west" access analysis can highlight commonalities in the technology and capabilities the attacker has and uses against its victims.

[1] http://www.activeresponse.org/wp-content/uploads/2013/07/diamond.pdf



Threat Intelligence Process Models

Different models used for different concepts:

1. **OODA Loop**
   - Use of threat intel for guidance in the "orient" step
   - Importance of fast operations tempo
2. **F3EAD**
   - About integrating threat intelligence with incident response
3. **Formal Intelligence Cycle**
   - Use for formal intelligence "products"
4. **Diamond Model**
   - For connecting incidents, usable if you don't have a dedicated T.I. team.

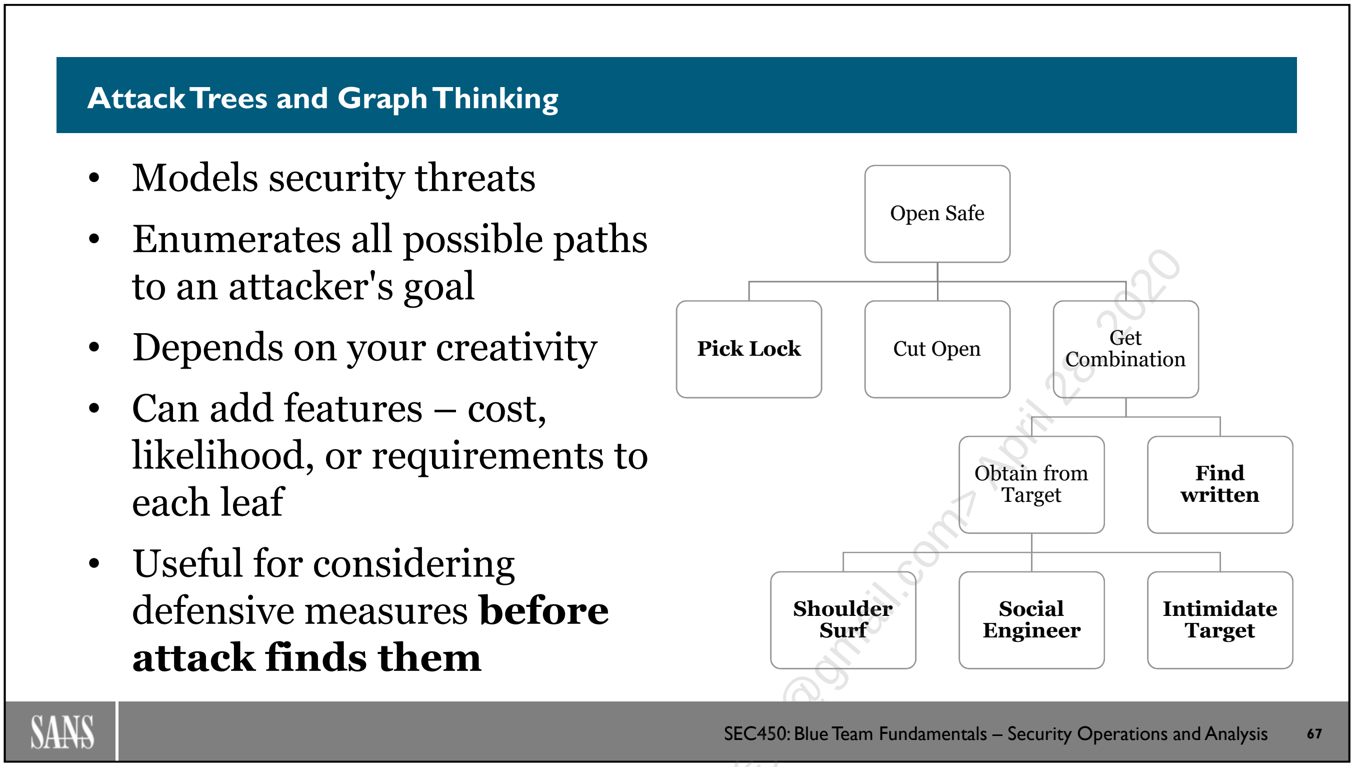
SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 66

Threat Intelligence Process models

So, when thinking about threat intelligence, which model is most relevant to us? The F3EAD concept sticks out as one of the most useful for analysts as it starts and ends with the incident detection and response process. This is one of the most important takeaways—threat intel should not only feed our ability to detect attacks, but detected attacks need to feed back to the threat intel group. Without this cycle, we risk turning our threat intel operation into what the community sometimes calls a "self-licking ice cream cone", or in other words, a system that self-perpetuates itself with no other purpose. Threat intel isn't collecting threat feeds and IOCs for their own health. These must be used as an input to our process and our process must be used as an output to theirs.

OODA loops are useful as well, but it is not as much of a specific model of threat intelligence as it is a reminder that acting quickly and understanding the situation accurately *using* threat intelligence is what will give us the advantage. The Diamond model can be interesting as well as a way of mapping out several incidents over time you suspect might be related. If you are a small team and do not have a formal threat intelligence team, it can be an interesting exercise to try to connect various attacks over time and lead your team toward developing an intelligence-driven incident response capability.

[1] Roberts, S. & Brown, R. (2017). *Intelligence-Driven Incident Response: Outwitting the Adversary*. Sebastopol, CA: O'Reilly Media.



Attack Trees and Graph Thinking

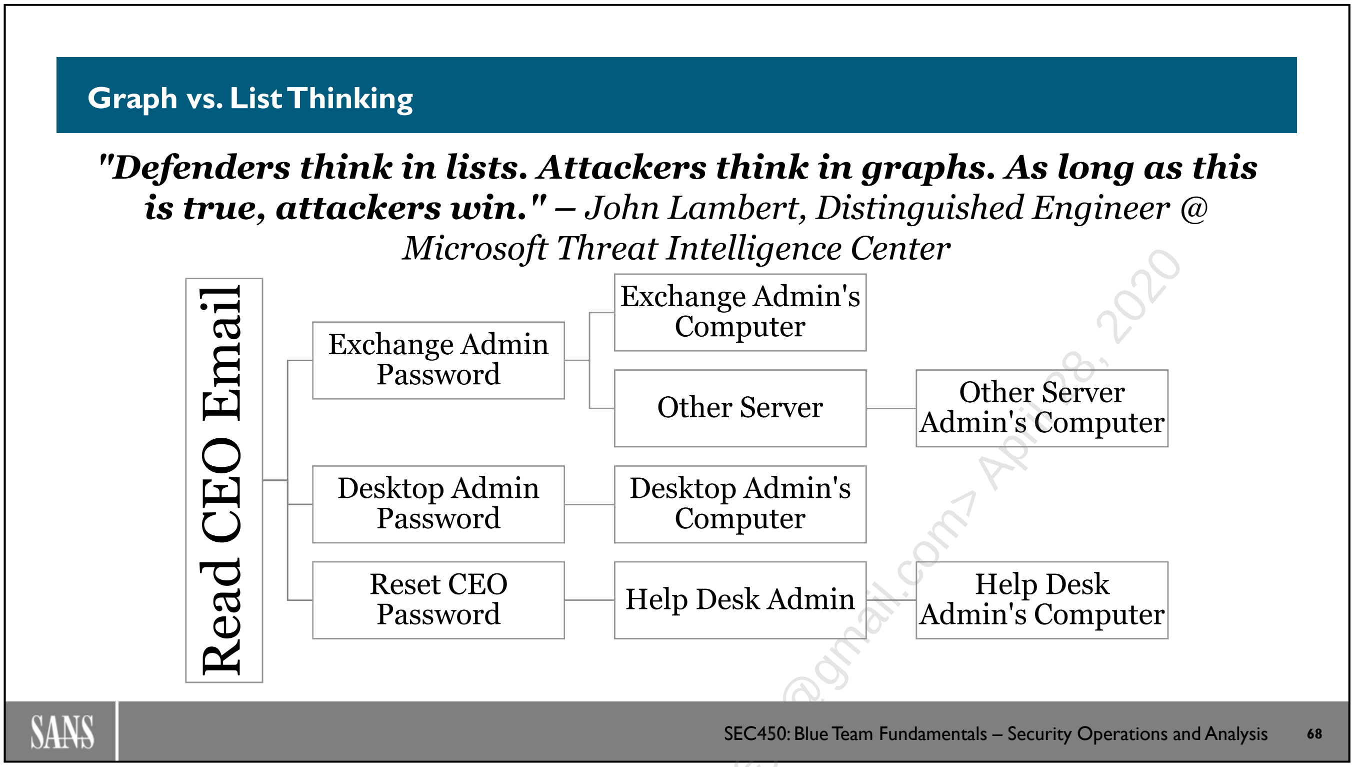
- Models security threats
- Enumerates all possible paths to an attacker's goal
- Depends on your creativity
- Can add features – cost, likelihood, or requirements to each leaf
- Useful for considering defensive measures **before attack finds them**

SEC450: Blue Team Fundamentals – Security Operations and Analysis 67

Attack Trees and Graph Thinking

One concept you'll likely hear about frequently in information security (and several times throughout this course) is the idea of graph thinking. One-way graph thinking can be applied to the information security realm in making what are called "attack trees." Attack trees are a structured problem-solving technique (remember how useful we said those were?) for coming up with a way to reach a specific goal (Bruce Schneier explained them well in his blog post all the way back in 1999 here[1]). They start with a single node at the top, the goal of the attack, and take one step back attempting to enumerate all possible methods and steps that could be used to achieve that goal. Although this alone is useful, you can make them even fancier by adding logical conditions, weightings, likelihoods and other metadata to pack even more analytic power into them. In the case of the tree above, it is enumerating ways we could open a safe. On the top line are the three main methods, picking the lock, cutting it open, or getting the combination. From there, we then enumerate possible ways of achieving those items. For getting the combination, this could be finding it or obtaining it from someone that already knows it. To do *that* we could either watch them enter it, socially engineer it out of them, or threaten them. The idea from the attacker side is to come up with the best possible plan of action, and alternatives if one goes wrong. From *our* side however, these become a great way of anticipating attacker moves and enumerating protections we could use to make sure they will not work.

[1] https://www.schneier.com/academic/archives/1999/12/attack_trees.html



Graph vs. List Thinking

"Defenders think in lists. Attackers think in graphs. As long as this is true, attackers win." *– John Lambert, Distinguished Engineer @ Microsoft Threat Intelligence Center*

Read CEO Email

Exchange Admin Password

Exchange Admin's Computer

Other Server

Other Server Admin's Computer

Desktop Admin Password

Desktop Admin's Computer

Reset CEO Password

Help Desk Admin

Help Desk Admin's Computer

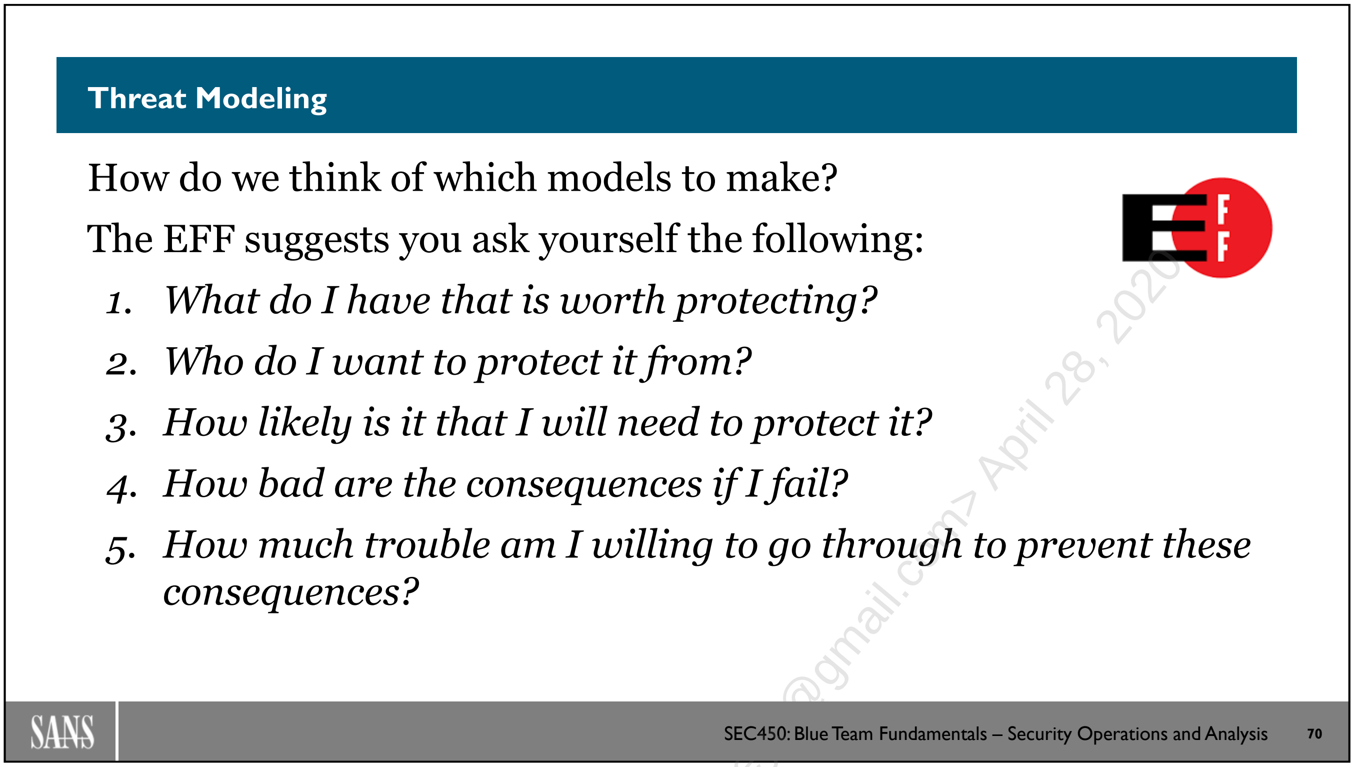
Graph vs. List Thinking

One insightful quote related to graphs from John Lambert, a Distinguished Engineer at the Microsoft Threat Intelligence Center, is "Defenders think in lists. Attackers think in graphs. As long as this is true, attackers win." What is meant by this? John explains in this article[1] that many operations get defense wrong from the start based on their incorrect notion of the way the cyber battlefield works. He says while defenders are busy obsessing over lists of controls and prioritizing assets, attackers see the network for what it truly is to them—a graph (of the graph theory, nodes and links nature) of security relationships. This is different than the previous slides graph of possible attacks. It refers to the literal nature of our networks, a graph of hosts and users connected via various relationships. When attackers succeed in the first step of their assault on our users and assets, they begin somewhere inside that graph. The point here is that the graph is our design—we choose which controls go where, how entities can interact with each other, and how much separation of accounts and hosts we have. Depending on how you have created your organization’s graph, attackers may have a tough or very easy time.

On the slide above, we have another attack tree with the attacker's true goal on the left side – reading the CEO's email. To get there, attackers must consider each possible path through the security relationship graph that would land them the capability to read the inbox, and then walk back further to find ways to get the previous step. This relationship graph is based on attacker thinking and knowledge of how credentials can be obtained in a Windows environment. It is something many blue teams don't consider nearly enough, and certainly one of the reasons we continue to see mega-breaches year after year.

In this example, we see there are at least three ways to get to the inbox: Stealing a Microsoft Exchange server administrator account, a desktop administrator account, or using the CEO's own actual account by resetting the password. Each of these steps is walked back to the next step it would take to acquire that information. For the exchange admin, their password can either be obtained from their desktop or a second server they administer. If another server is used, we may be able to steal someone's password who is a co-administrator of that second server and use their access to run Mimikatz on that server and get the Exchange admin's password as well. This same reasoning works for desktop or help desk admins as well. You can see how this can be walked further and further back to make multiple paths. The point here is to find a solution to obtaining data. Attackers use graph thinking; to have a chance of meeting their clever tactics, we must do the same and understand what they will try before they try it. In this case, we would continue filling out this graph with as many ideas as possible and then place our available and potential defenses against the tree to see if they line up. This is one way of getting ahead of attackers by anticipating their moves beforehand and a great way to decide where to lay traps they will fall into if this attack is attempted.

[1] https://github.com/JohnLaTwC/Shared/blob/master/Defenders%20think%20in%20lists.%20Attackers%20think%20in%20graphs.%20As%20long%20as%20this%20is%20true%2C%20attackers%20win.md



Threat Modeling

How do we think of which models to make?

The EFF suggests you ask yourself the following:

1. *What do I have that is worth protecting?*
2. *Who do I want to protect it from?*
3. *How likely is it that I will need to protect it?*
4. *How bad are the consequences if I fail?*
5. *How much trouble am I willing to go through to prevent these consequences?*

Threat Modeling

Though Threat Modeling can mean different things whether you are talking about software development threat modeling or general IT threats, the same type of questions applies to coming up with dangers that apply to you. You attempt to define what you're trying to protect and how adversaries will try to come after it so that you can come up with a clear plan to defend it. A great generic article that introduces the concept of threat modeling readable by any is the referenced Ars Technica post, "How I Learned to Stop Worrying Mostly and Love My Threat Model".[1]

A good set of questions that can be used for coming up with situations to model with an attack tree is the Electronic Frontier Foundation's (EFF) questions for surveillance self-defense.[2] Although the questions were designed for individuals to keep their personal data safe, they can be slightly tweaked (which was done for the slide) to apply to organizations in general.

1. What do I have that is worth protecting?
2. Who do I want to protect it from?
3. How likely is it that I will need to protect it?
4. How bad are the consequences if I fail?
5. How much trouble am I willing to go through to prevent these consequences?

An alternative but very similar set of questions designed for software threat modeling was put forth by Adam Shostack, author of *Threat Modeling: Designing for Security*.[3] As you can see, they are incredibly similar in their line of thought to the generic questions from the EFF.

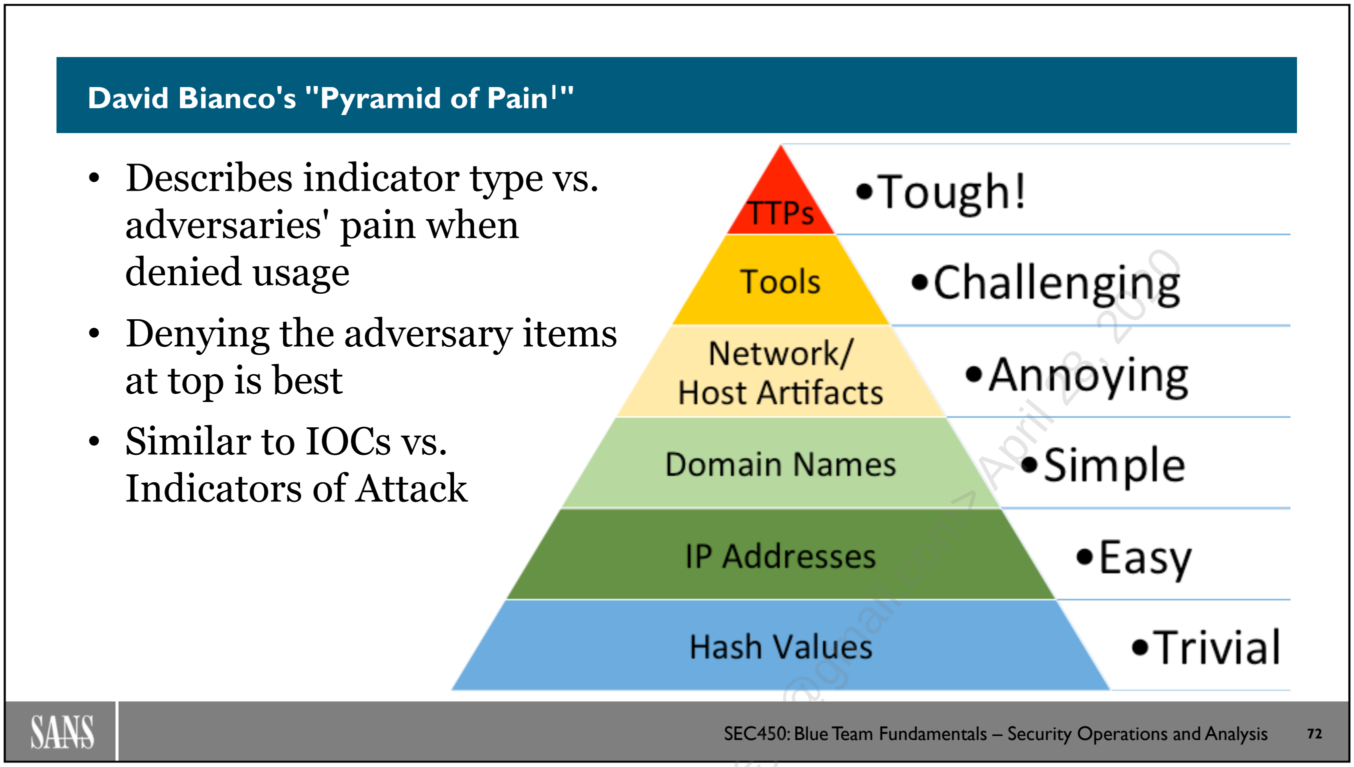
1. What are you doing? (and what info is involved)
2. What can go wrong? (consider all attack types, possibly using STRIDE model – Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service, Elevation of privilege)

3. What are you going to do about it? (identify improvements)
4. Did you do a good job? (re-assessment)

[1] https://arstechnica.com/information-technology/2017/07/how-i-learned-to-stop-worrying-mostly-and-love-my-threat-model/

[2] https://ssd.eff.org/en/module/your-security-plan

[3] https://www.wiley.com/en-us/Threat+Modeling%3A+Designing+for+Security-p-9781118809990



David Bianco's "Pyramid of Pain[1]"

- Describes indicator type vs. adversaries' pain when denied usage
- Denying the adversary items at top is best
- Similar to IOCs vs. Indicators of Attack

TTPs	•Tough!
Tools	•Challenging
Network/ Host Artifacts	•Annoying
Domain Names	•Simple
IP Addresses	•Easy
Hash Values	•Trivial

David Bianco's "Pyramid of Pain"

Another rather famous model for thinking of indicators that analysts may see and use in cyber defense is David Bianco's "Pyramid of Pain"[1]. David Bianco, a previous Hunt Team Lead at Mandiant and now Principal Cybersecurity engineer at Target, devised this graphic as a way of thinking about the best way to ruin an adversary's day once you know something about their operation. Throughout the course of your career, you will come upon an innumerable amount of hash values. Every virus has a unique one and changing them only requires modifying a single bit. Hash values sit on the bottom of the pyramid of pain because if, during incident response you decide to block an attack based on hash value, you can bet the attackers will be back soon with a modified version of that program with a different hash. It simply is too easy of a thing for them to get around to be a good long-term effective defense. That's not to say you shouldn't use them for block, just that it's not enough on its own. Moving up the pyramid IP addresses and domain names are slightly more annoying for them to change, but still ultimately only a matter of acquiring a new Amazon VPS or otherwise. Domain names can be acquired for free[2] so these do no present a significant roadblock either.

The best ways to ruin an attacker's day are higher-level items like network protocol-based blocks or host artifacts, or even the tools themselves. These require attackers to go back to the drawing board and at a minimum recompile their malware to act different so that it can't be spotted by these more broadly applicable protections. Finally, at the top of the pyramid, we have TTPs—tactics, techniques, and procedures. This is understanding the attackers' capability and style of attack in a deep way, and items at this level, such as the items enumerated in the MITRE ATT&CK matrix, force attackers to go back to the drawing board and devise entirely new methods for exploitation and post-exploitation, which is clearly the most effective way to slow them down in the long term. Keep this model fresh in your mind as it is a great reference to consider when going to design an analytic for an IDS, or to stop an in-progress intrusion.

Another perhaps more binary way of describing this same concept is referring to "indicators of attack" vs. indicators of compromise.[3] CrowdStrike draws this distinction in the referenced article by saying if you have seen someone in a red hat robbing your bank, that is an IOC that is out of date and can only lead to a reactive defense that can be easily changed by the thief the next time. If you can catch the perpetrator first casing the bank, sneaking into the back room, and hacking into the vault, these are real-time indicators of attack (or TTPs) that be proactively monitored for to catch subsequent intrusions.

[1] http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html

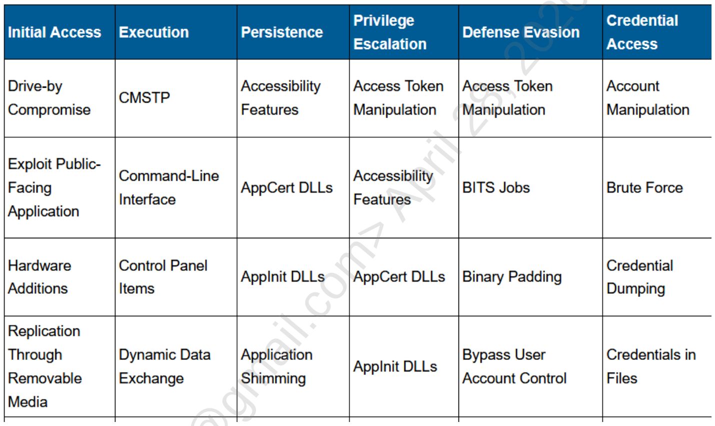
[2] http://dot.tk

[3] https://www.crowdstrike.com/blog/indicators-attack-vs-indicators-compromise/

MITRE ATT&CK

A list of "**A**dversary **T**actics, **T**echniques **&** **C**ommon **K**nowledge"

- A list of things at the **top of the Pyramid of Pain**
- **Tactics** on top row
- **Techniques** in column
- Blue Team Checklist!!!
- Becoming a popular standard framework
- Conference built for users
  - ATT&CKCON



Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access
Drive-by Compromise	CMSTP	Accessibility Features	Access Token Manipulation	Access Token Manipulation	Account Manipulation
Exploit Public-Facing Application	Command-Line Interface	AppCert DLLs	Accessibility Features	BITS Jobs	Brute Force
Hardware Additions	Control Panel Items	AppInit DLLs	AppCert DLLs	Binary Padding	Credential Dumping
Replication Through Removable Media	Dynamic Data Exchange	Application Shimming	AppInit DLLs	Bypass User Account Control	Credentials in Files

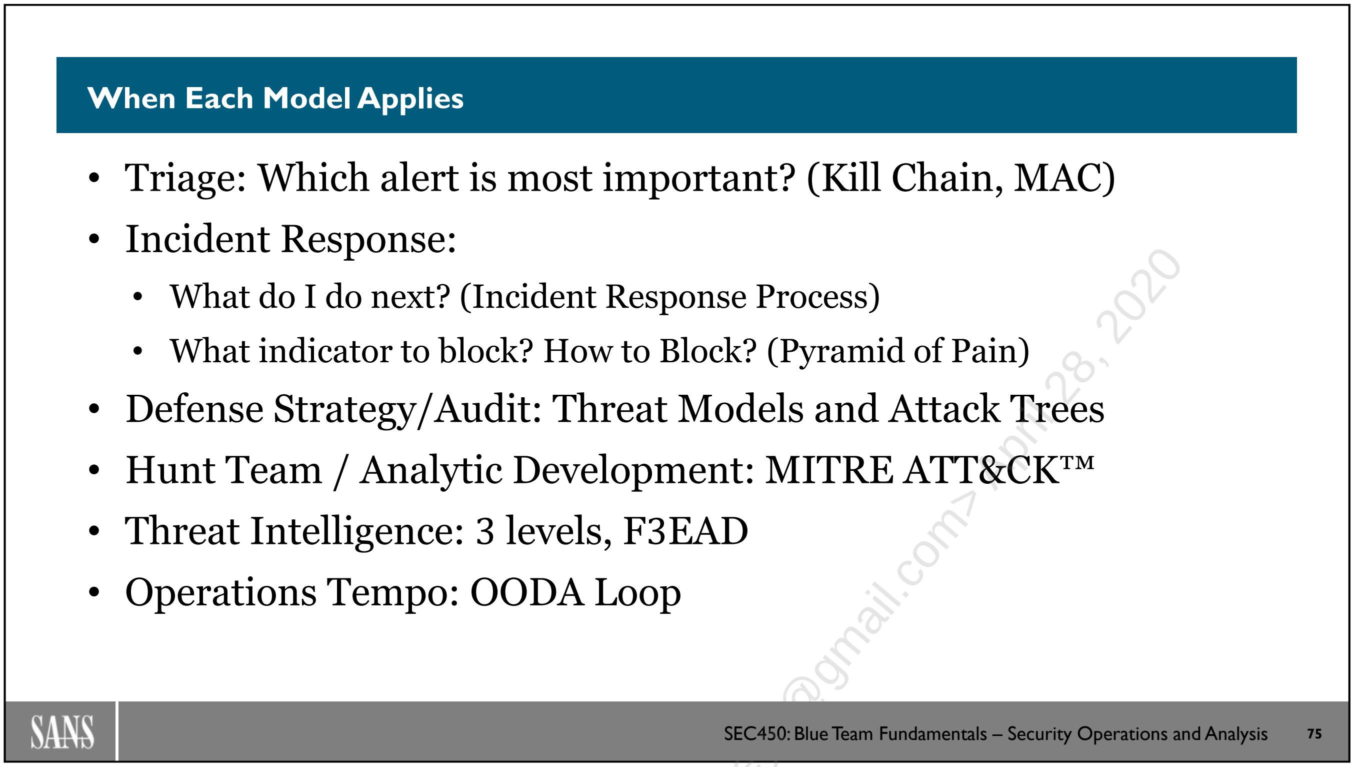
MITRE ATT&CK

We've mentioned the MITRE ATT&CK[1] matrix before a couple of times but haven't had a specific discussion about it yet. Simply put, this is a body of knowledge that you absolutely should be familiar with as a defender. MITRE, a U.S. government-funded research organization, has developed the framework as a way of trying to standardize and categorize all the common attack tactics and techniques seen in the wild. In this respect, the tactics is the item across the top on the blue bar and answer the question of "what" the attackers are trying to accomplish. Each item underneath in the column is a specific way to accomplish the tactic and answers the question of "how" it can be done. For example, the tactic of Credential Access can be accomplished through the techniques of account manipulation, brute force, credential dumping, or finding credentials in files.

The information itself is split into multiple matrices. There’s the ATT&CK Enterprise matrix, which focuses on post-exploit tactics (plus the new initial access section) seen in the context of the typical corporate network assets. This is further divided into Windows, Linux and macOS sections. There's also the Mobile ATT&CK and PRE-ATT&CK matrices, which are newer efforts to categorize both mobile platform attack techniques, as well as pre-exploit phase items respectively. Navigating to the website will bring up each matrix in which each item can be clicked for further detail. Each technique is helpfully listed not only with a description, but a list of known attackers who used it and the attack it was used in, as well as ways to mitigate and detect the technique. The ATT&CK matrix is a living document that is constantly being added to due to discussions with its creators on Twitter and has caught on in a big way with the industry. MITRE even held the first-ever ATT&CKCON in 2018 where representatives from various vendors and organizations all met up to discuss how they could use it and how it could be improved (videos available here[2]). Given how useful this model has been to so many people already, expect this trend to continue.

[1] https://attack.mitre.org

[2] https://infocon.org/cons/ATT&CKcon/ATT&CKcon%202018/



When Each Model Applies

- Triage: Which alert is most important? (Kill Chain, MAC)
- Incident Response:
  - What do I do next? (Incident Response Process)
  - What indicator to block? How to Block? (Pyramid of Pain)
- Defense Strategy/Audit: Threat Models and Attack Trees
- Hunt Team / Analytic Development: MITRE ATT&CK™
- Threat Intelligence: 3 levels, F3EAD
- Operations Tempo: OODA Loop

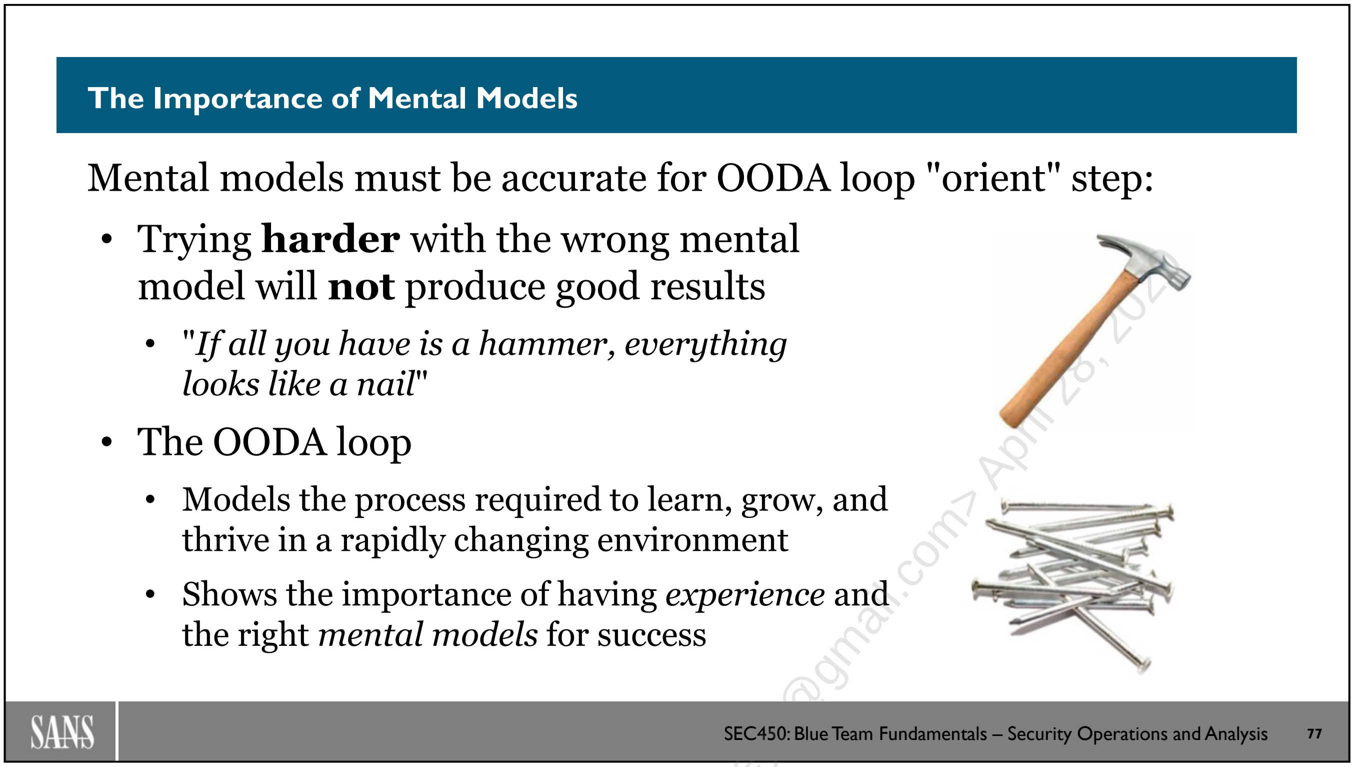
When Each Model Applies

Now that we've gone through several models, let's go through some examples of when and how you can use each one.

- Triage: At the triage stage, it's all about apparent risk. Remember, when you're looking at that list of alerts, have the Lockheed Martin cyber kill chain or Mandiant attack cycle in the back of your head. Ask yourself "which stage of the attack does this seem to be in?" The ones near the end of the line showing an active infection with unblocked, successful command and control channels should be dealt with first (unless of course, you sense active exfiltration, a thankfully much rarer occurrence).
- Incident Response: An incident has been declared. Quick, what is your first move? Refer here to the incident response cycle (PICERL). Since you've already prepared and set up the detections that have got you this far and know there is an active incident, you have made it through the "P" and "I" section. What the cycle then says you should do next is stop the bleeding. If you see anything actively going on, time to cut off communication (except in the case where you have decided an attack is worth the "watch and wait" risk, which we will cover later). Take the best action you can to prevent command and control from proceeding, whether it be a network firewall or proxy block, host or network IPS, and even a host-based firewall rule. Cutting off communication from the infected device to the attacker will at least temporarily prevent any damage that is being done. After further damage is prevented, you can move on to the next steps of eradicating and remediating the situation.
- Defensive Strategy and Auditing of Defense: If someone asks you what control you think is missing, or which would be the next most important addition, how do you decide? One great way is to go back to your threat model and your attack trees. If you know what could cause the biggest impact in your environment, and you know the steps to get there, it should be easy to align what to do next with the biggest risk that remains. Select the item that would have the biggest impact on the residual risk left over from uncovered attack paths.
- Hunt Teaming and Analytic Development: One of the goals of the hunting process is to perform an "IOCless" search based on hypothesis and known gaps in your defensive coverage. To do this, you need

to have a good list of what attack tactics exist so you can line them up with your present detection capabilities. MITRE's ATT&CK matrix can be a great place for the hunt team to start, hypothesizing that the attackers will likely be using one of those techniques, especially the ones you aren't yet looking for.

- Threat Intelligence: If you're thinking "what do I do with the indicators of all of these incidents?", or "how can I use outside information to bolster and focus my defensive work?" the three levels of threat intelligence and F3EAD process can be your guide. Remember that threat intel isn't for analysts only and must be a two-way street with a constant feedback loop. It is not simply a list of low-level atomic indicators coming into your organization that a vendor says is bad for some poorly documented reason.
- Operations tempo: When it comes to the ops tempo, remember the overall model of the OODA loop, which can give us a way to predict who will win any given battle. It is a reminder to stay nimble, aware, and on your feet—the one with the most capability and flexibility inherently has the advantage in any contest.



The Importance of Mental Models

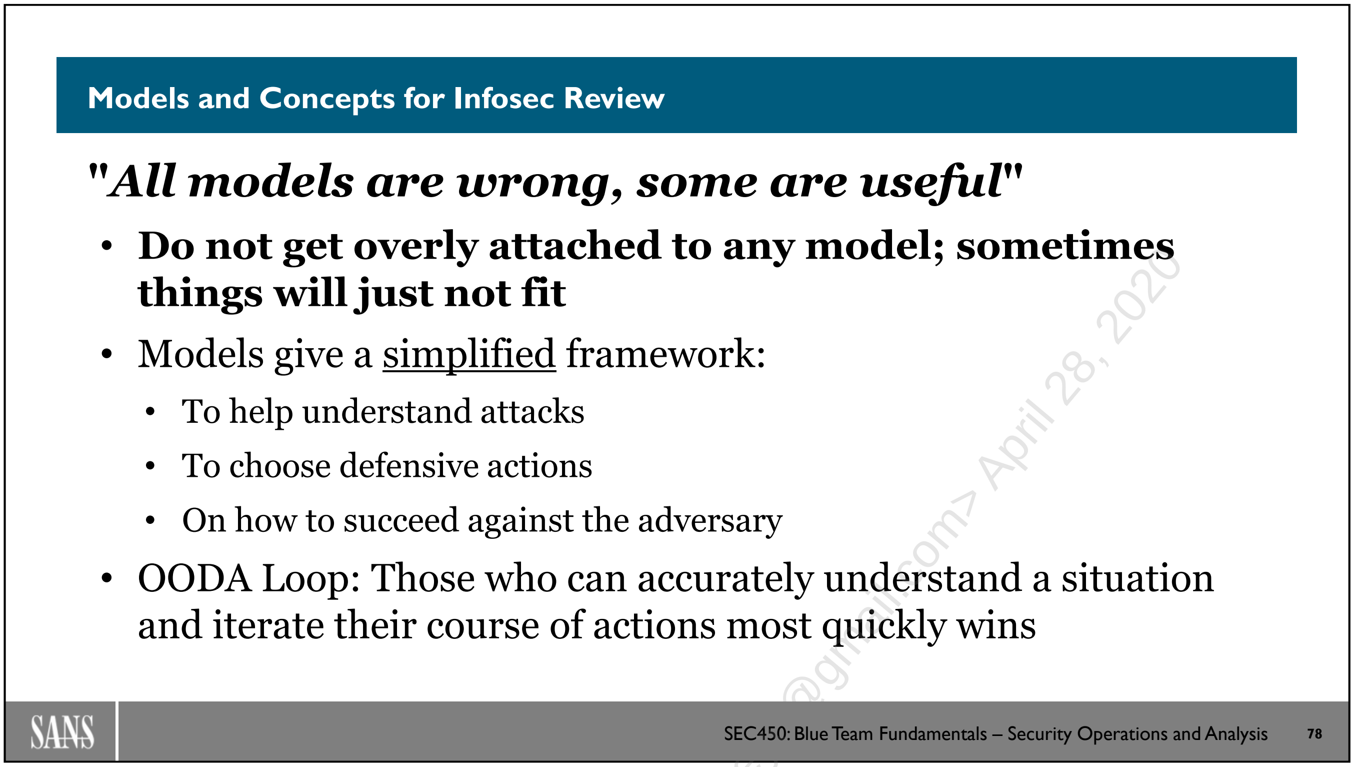
Mental models must be accurate for OODA loop "orient" step:

- Trying **harder** with the wrong mental model will **not** produce good results
  - *"If all you have is a hammer, everything looks like a nail"*
- The OODA loop
  - Models the process required to learn, grow, and thrive in a rapidly changing environment
  - Shows the importance of having *experience* and the right *mental models* for success

The Importance of Mental Models

The goal of this section is twofold. The first goal is to provide you with some of the industry standard models that you can use to conceptualize attacks that you may run into. Having the Kill Chain and Mandiant attack cycle committed to memory can be an incredibly useful framework for understanding the stage of an attack, and what may come next. Hopefully, understanding the three levels of threat intel, how threat intelligence should feed and be informed by the SOC, the Pyramid of Pain/DML model, and attack graphs, will inform your work as well by helping you *decompose* and *externalize* those concepts into something more manageable.

The second goal is introducing the idea of the OODA Loop because it underscores the importance of knowing these models. As the OODA process points out, in any rapidly changing environment (which cyber defense naturally is), the side that understands the battlefield and situation more clearly and can adjust that perception at a faster rate will have the strategic advantage. Since shifting models quickly require having numerous options to draw on and an understanding of how they relate, the goal of this section was to explain and connect these concepts to help kickstart or improve this capability. By framing some of the most popular defense, attack, threat intelligence models, we aim to improve your capability to Observe and Orient yourself so that you can rapidly Decide and Act on the information you gather during your investigations.



Models and Concepts for Infosec Review

"All models are wrong, some are useful"

- **Do not get overly attached to any model; sometimes things will just not fit**
- Models give a simplified framework:
  - To help understand attacks
  - To choose defensive actions
  - On how to succeed against the adversary
- OODA Loop: Those who can accurately understand a situation and iterate their course of actions most quickly wins

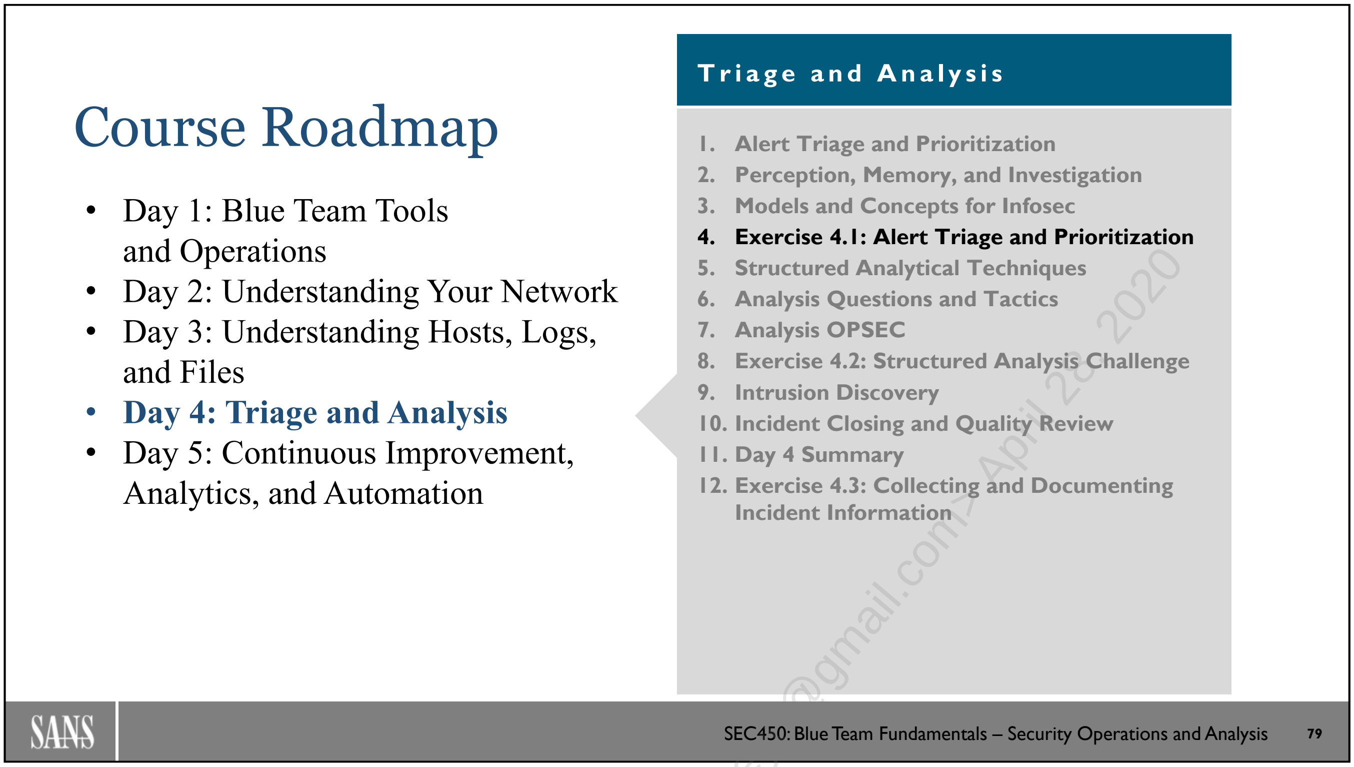
SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 78

Models and Concepts for Infosec Review

A final word on models—when it comes to them, remember this famous quote often attributed to the influential statistician George Box, "All models are wrong, some are useful."[1] By this we mean, don't try *too* hard to fit everything into the models we've presented. Earlier, we mentioned the concept of bounded rationality, which is what drives our necessity to simplify things, but simplifying by nature requires leaving out detail. Every attack won't fit perfectly into the Kill Chain or Mandiant Attack Cycle. The pyramid of pain and PICERL may not guide you to the most perfect decision 100% of the time, etc. Take them for what they are—a simplification that guides us in the right direction. These models give us a starting path and a skeleton to model off when approaching a problem and should not be used as a golden rule in every situation. The nuance of every incident will be slightly different and potentially require a different approach, which is why developing analytical and critical thinking skills are one of the most important goals for new analysts.

[1] https://en.wikipedia.org/wiki/George_E._P._Box



Course Roadmap

- Day 1: Blue Team Tools and Operations
- Day 2: Understanding Your Network
- Day 3: Understanding Hosts, Logs, and Files
- **Day 4: Triage and Analysis**
- Day 5: Continuous Improvement, Analytics, and Automation

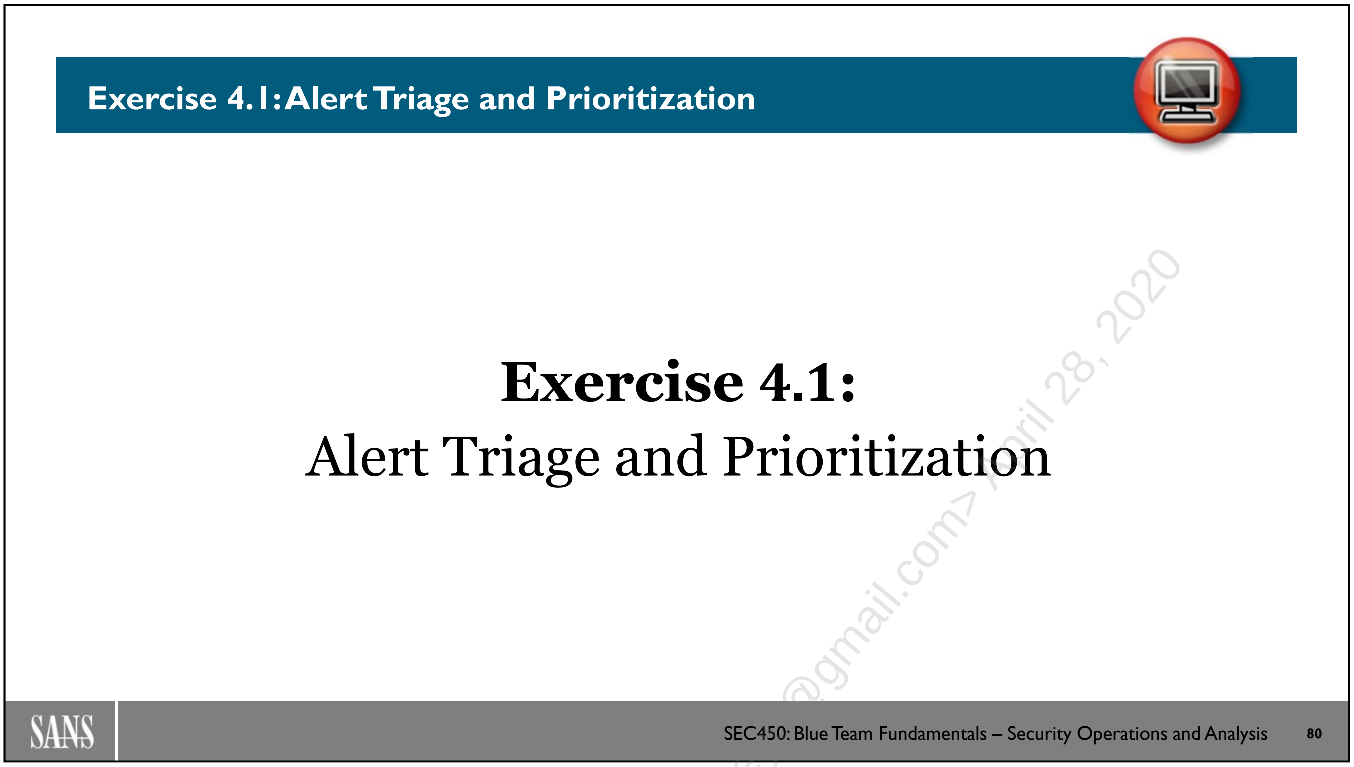
Triage and Analysis

1. Alert Triage and Prioritization
2. Perception, Memory, and Investigation
3. Models and Concepts for Infosec
4. **Exercise 4.1: Alert Triage and Prioritization**
5. Structured Analytical Techniques
6. Analysis Questions and Tactics
7. Analysis OPSEC
8. Exercise 4.2: Structured Analysis Challenge
9. Intrusion Discovery
10. Incident Closing and Quality Review
11. Day 4 Summary
12. Exercise 4.3: Collecting and Documenting Incident Information

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 79

This page intentionally left blank.

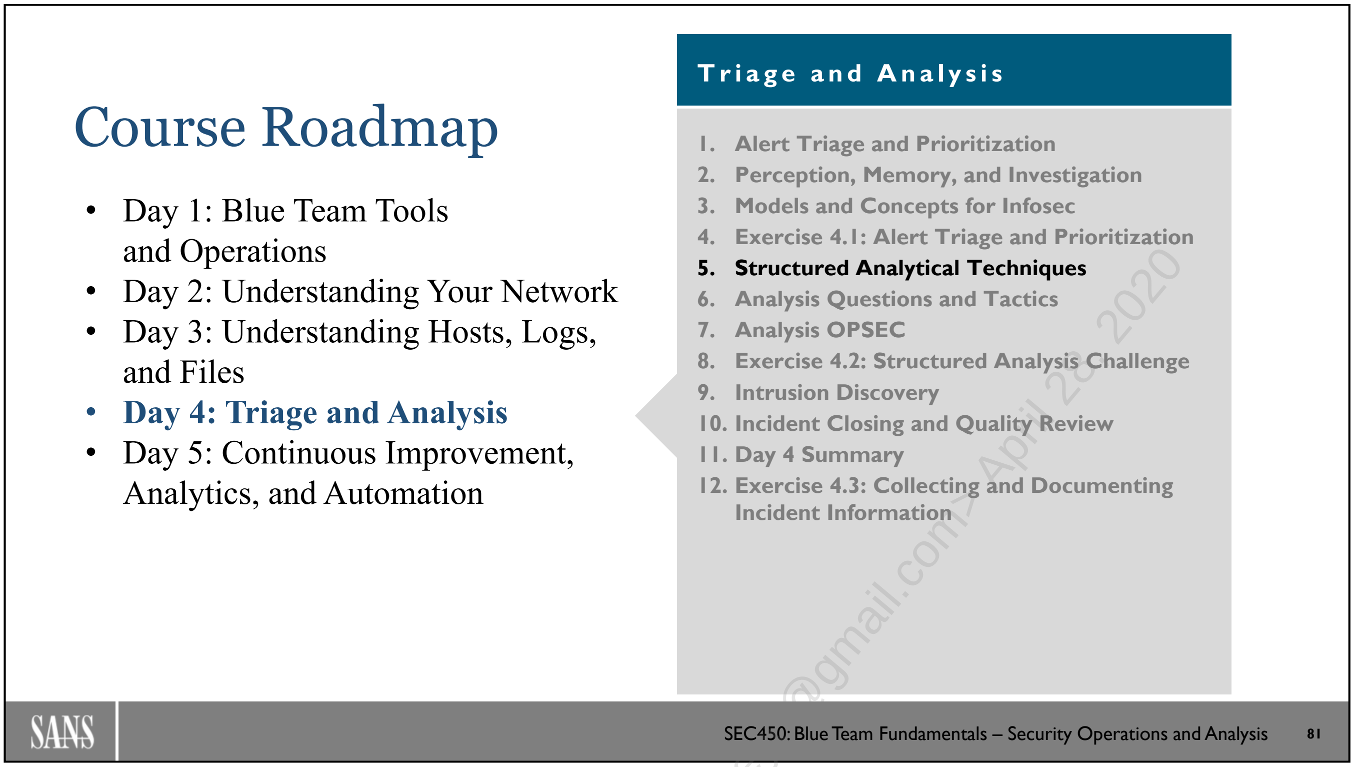


Exercise 4.1: Alert Triage and Prioritization

Exercise 4.1:
Alert Triage and Prioritization

Exercise 4.1: Alert Triage and Prioritization

Please go to Exercise 4.1 in the SEC450 Workbook or virtual wiki.



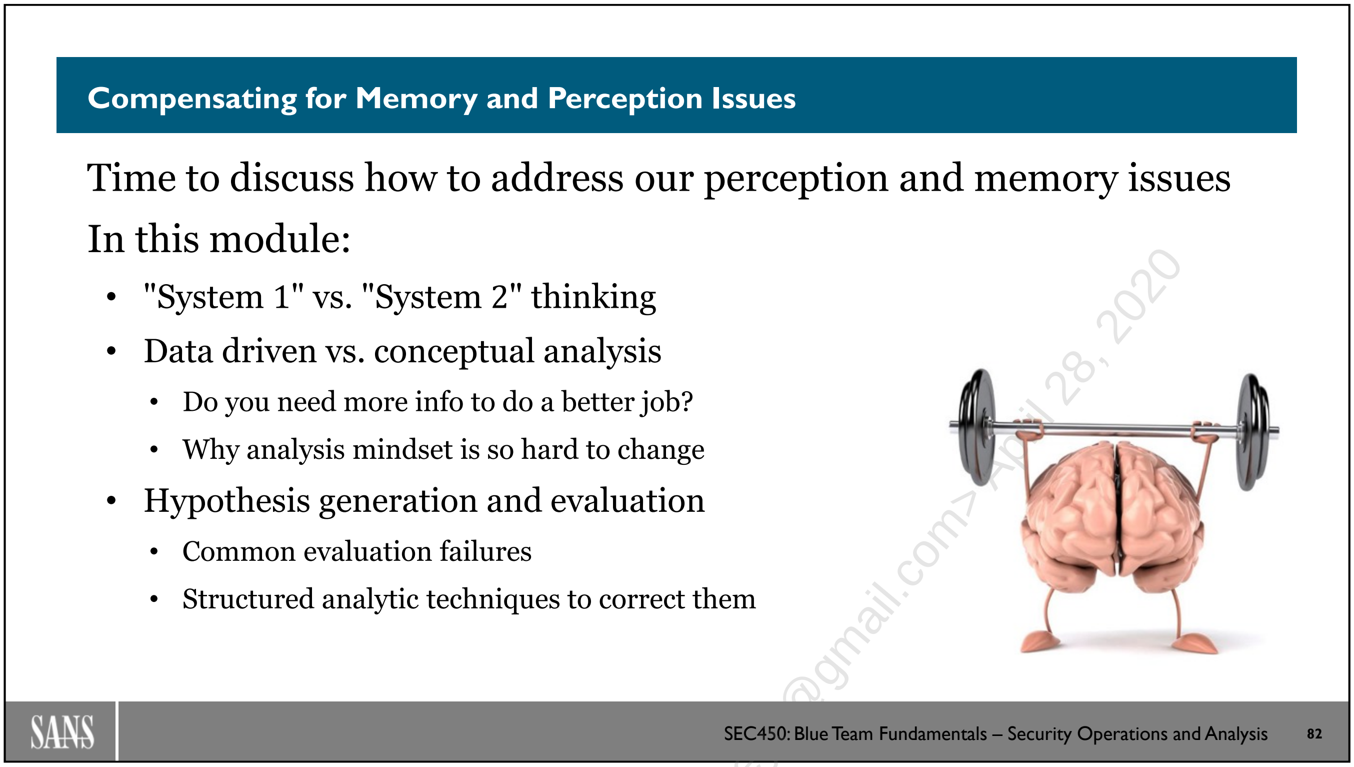
Course Roadmap

- Day 1: Blue Team Tools and Operations
- Day 2: Understanding Your Network
- Day 3: Understanding Hosts, Logs, and Files
- **Day 4: Triage and Analysis**
- Day 5: Continuous Improvement, Analytics, and Automation

Triage and Analysis

1. Alert Triage and Prioritization
2. Perception, Memory, and Investigation
3. Models and Concepts for Infosec
4. Exercise 4.1: Alert Triage and Prioritization
5. **Structured Analytical Techniques**
6. Analysis Questions and Tactics
7. Analysis OPSEC
8. Exercise 4.2: Structured Analysis Challenge
9. Intrusion Discovery
10. Incident Closing and Quality Review
11. Day 4 Summary
12. Exercise 4.3: Collecting and Documenting Incident Information

This page intentionally left blank.



Compensating for Memory and Perception Issues

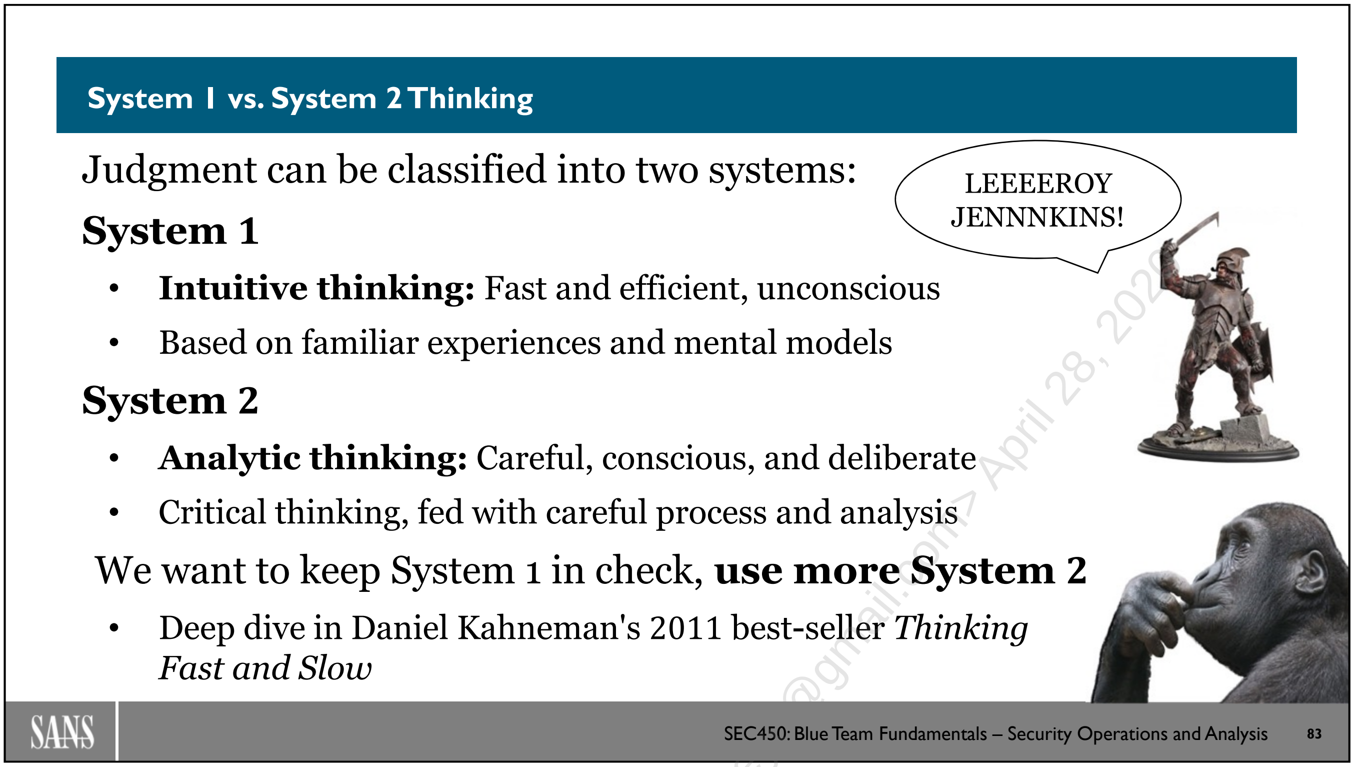
Time to discuss how to address our perception and memory issues

In this module:

- "System 1" vs. "System 2" thinking
- Data driven vs. conceptual analysis
  - Do you need more info to do a better job?
  - Why analysis mindset is so hard to change
- Hypothesis generation and evaluation
  - Common evaluation failures
  - Structured analytic techniques to correct them

Compensating for Memory and Perception Issues

Now that we have discussed the functional benefits and deficiencies of our perception and memory, as well as discussed some mental models for information security, it's time to go over how we can start to compensate for these issues in our analysis. Throughout this section, we will review analysis methods, the difference between intuitive and analytical thinking, data vs. conceptual analysis as it applies to alert triage and investigation, as well as strategies to perform better at hypothesis generation and evaluation.



System 1 vs. System 2 Thinking

Judgment can be classified into two systems:

System 1

- **Intuitive thinking:** Fast and efficient, unconscious
- Based on familiar experiences and mental models

System 2

- **Analytic thinking:** Careful, conscious, and deliberate
- Critical thinking, fed with careful process and analysis

We want to keep System 1 in check, **use more System 2**

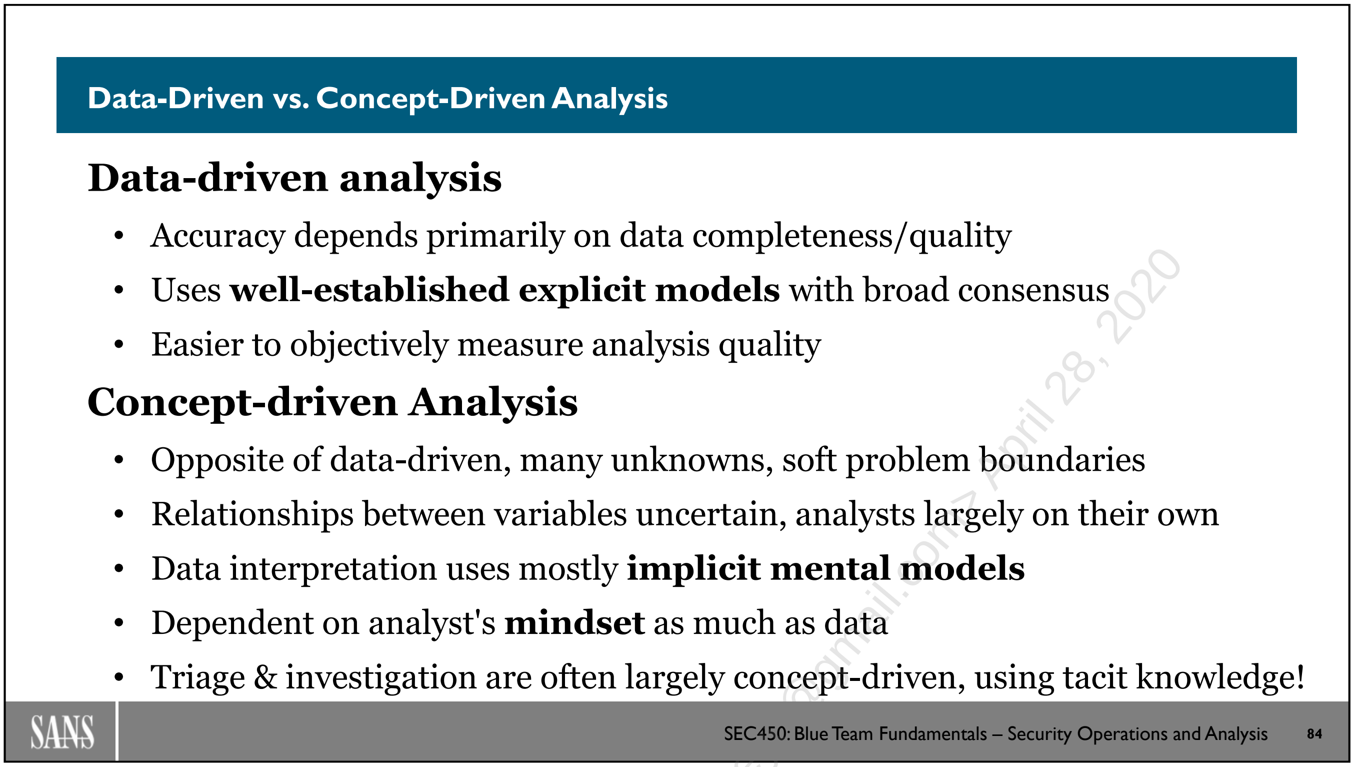
- Deep dive in Daniel Kahneman's 2011 best-seller *Thinking Fast and Slow*

System 1 vs. System 2 Thinking

Psychological research on human judgment has shown that we can think of decision-making being performed with 2 different systems. These have been labeled "System 1" and "System 2." Far from some obscure psychological theory, this model is the subject of one of the best-selling books of 2011, *Thinking Fast and Slow* by Daniel Kahneman, an expert on the topics of judgment and decision-making.[1]

As described, System 1 is described as intuitive thinking—the snap judgments you can make about many topics and questions based on past experience and your existing mental models. This mode of thinking is extremely efficient and works much of the time. The second system, System 2, is the opposite and is the realm of analytical thinking. This type of thinking is a very slow, deliberate, and conscious effort, and involves evaluating all the data and making a reasoned response. People tend to do this for big life purchases and decisions and other items where the stakes are high for a wrong answer. You can likely already see how this is going to relate to investigations. In this module, we'll discuss these two modes and the pros and cons of each, as well as ways to help make our analysis techniques better by moving away from system 1 and toward system 2 for those big, high stakes incidents, and anything else we want to be extremely clear and deliberate about.

[1] https://www.amazon.com/gp/product/B00555X8OA/ref=dbs_a_def_rwt_hsch_vapi_tkin_p1_i0



Data-Driven vs. Concept-Driven Analysis

Data-driven analysis

- Accuracy depends primarily on data completeness/quality
- Uses **well-established explicit models** with broad consensus
- Easier to objectively measure analysis quality

Concept-driven Analysis

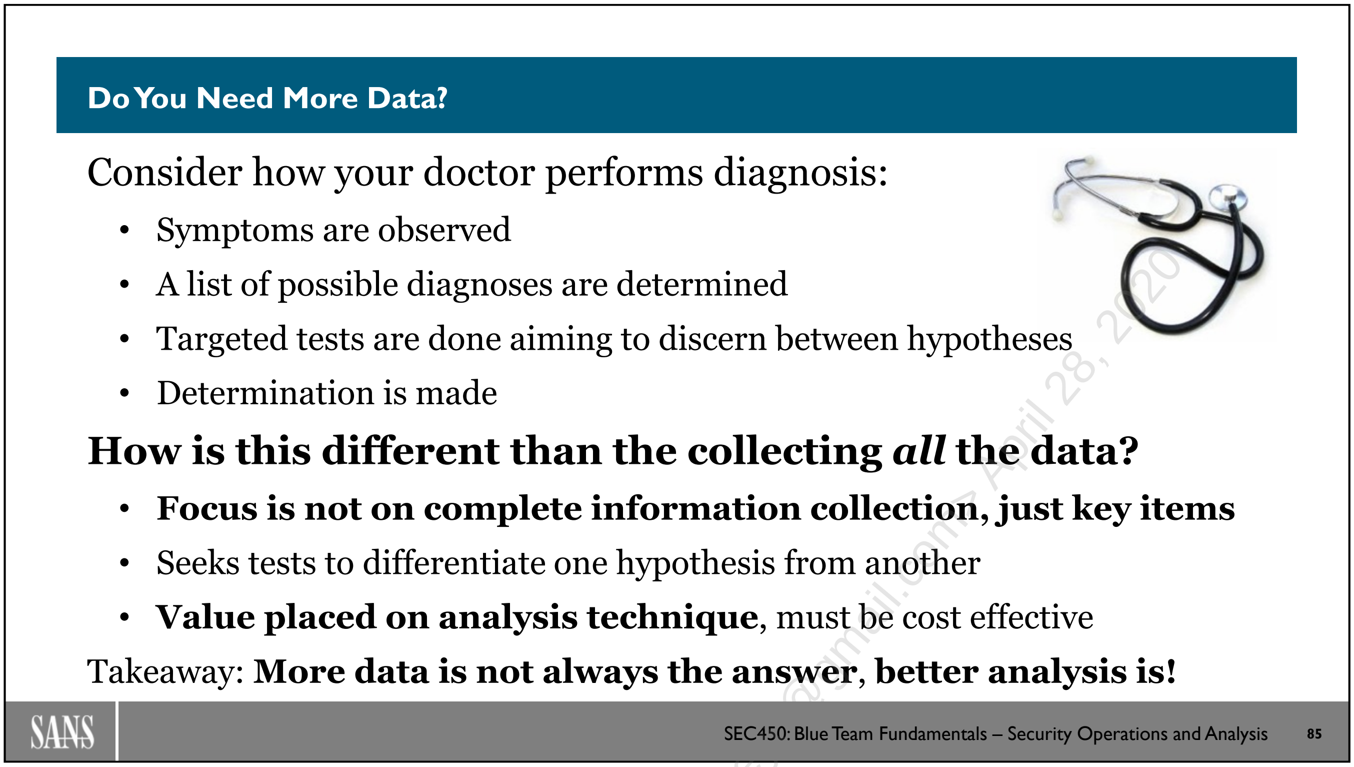
- Opposite of data-driven, many unknowns, soft problem boundaries
- Relationships between variables uncertain, analysts largely on their own
- Data interpretation uses mostly **implicit mental models**
- Dependent on analyst's **mindset** as much as data
- Triage & investigation are often largely concept-driven, using tacit knowledge!

Data-Driven vs. Concept-Driven Analysis

Considering the types of problems we can analyze and how gathering new information affects the outputs of that analysis, Heuer breaks analytical styles down into two main camps: Data-driven and concept-driven analysis.[1] On the one hand, we have data-driven analysis. These are the types of analysis that have been long established methods with a strong consensus on analytical frameworks. For these types, given the same information and capability to apply the consensus models, two people are highly likely to produce the same conclusion given the same data. This is because the analysis is based on well-established, explicit models that have been codified and shared throughout the profession. Because of the standardization, it is easy to measure and ensure quality of analytical output.

On the other hand, we have concept-driven analysis. This is the polar opposite of data-driven analysis in that the problem and data are much more open to interpretation. Although lots of data may be collected, the important relationships between items in the data might not be clear and different items might stick out as relevant to different analysts. Because of this, analysts tend to use implicit mental models that are harder to explain or judge. This is the type of analysis that is heavily influenced by mindsets and schemas, and the breadth of experience the analyst has had in the past to inform them. Unfortunately, much of alert investigation and incident response fall into the concept-driven analysis camp unless there is a very high degree of information captured. Even when there is, the limits of working memory can still make it difficult to use, and the items of data that are most important within that data may not be clear.

[1] Heuer,1999, pp. 59-62



Do You Need More Data?

Consider how your doctor performs diagnosis:

- Symptoms are observed
- A list of possible diagnoses are determined
- Targeted tests are done aiming to discern between hypotheses
- Determination is made

How is this different than the collecting *all* the data?

- **Focus is not on complete information collection, just key items**
- Seeks tests to differentiate one hypothesis from another
- **Value placed on analysis technique**, must be cost effective

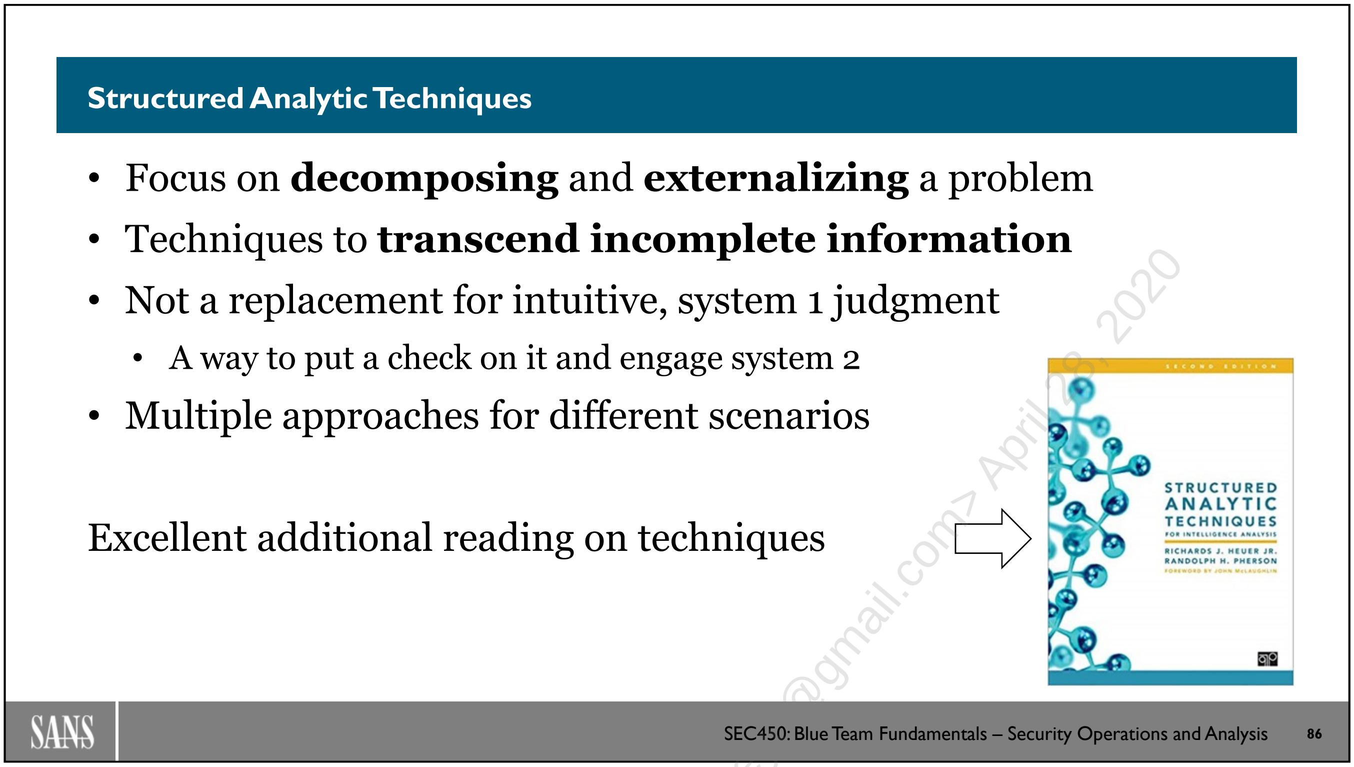
Takeaway: **More data is not always the answer, better analysis is!**

Do You Need More Data?

Is collecting all possible data necessarily the right approach to take or what will get you to the final answer? This alternative style of analysis would suggest no, and Heuer would agree. A whole chapter of *The Psychology of Intelligence Analysis* is "Do You Really Need More Information" and surprise, the conclusion is likely "no". Comparing the Mosaic style of analysis vs. Medical Diagnosis is one of way highlighting the reason why.

Instead of focusing on collecting that extra piece of data that may finally put the picture together, the approach a doctor uses to make a diagnosis takes a different route. Consider how a doctor's visit typically goes: You show up and tell the doctor your symptoms and the first move is considering all the options for what those symptoms may represent (hypothesis generation). Once some possible options are decided upon, the key difference between medical diagnosis method and mosaic theory of analysis shows up. Instead of doing every possible test that can be done, which would cost incredible amounts of money, the doctor attempts a more focused approach. While someone following the mosaic approach might say you can get to the answer by collecting all the test results possible, the doctor would likely focus on a few key tests that will discern one condition from another given the information that is known. This allows them to decide based on select test results and rule out the other diagnoses as cost effectively and efficiently as possible. This highlights the fact that while data sometimes *can* help the situation, in many cases, you might need as much as you might think and would be better served following a more thorough analysis technique.

[1] Heuer, 1999, p. 62



Structured Analytic Techniques

- Focus on **decomposing** and **externalizing** a problem
- Techniques to **transcend incomplete information**
- Not a replacement for intuitive, system 1 judgment
  - A way to put a check on it and engage system 2
- Multiple approaches for different scenarios

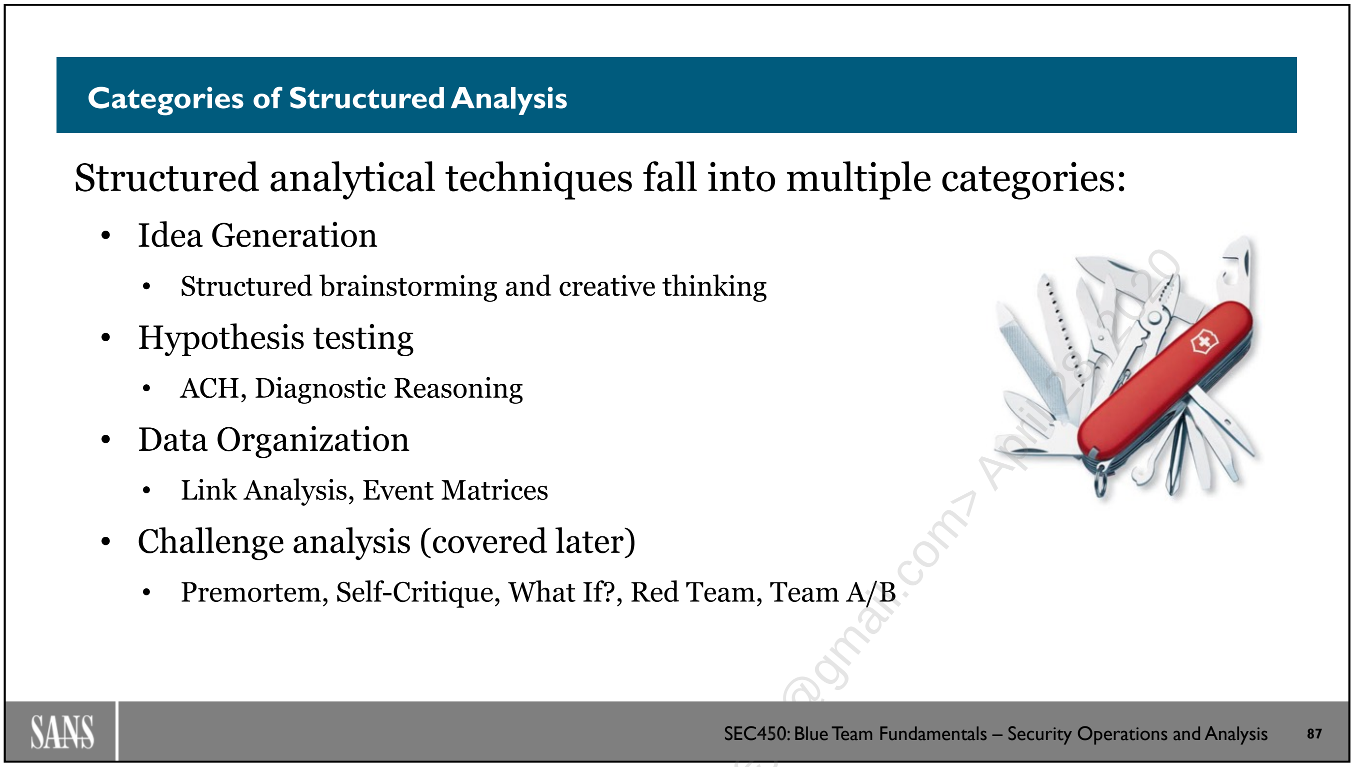
Excellent additional reading on techniques

Structured Analytic Techniques

While analysis in any information security realm will never meet the standard of science given the inherent unknowns and ambiguous nature of much of the data we deal with, we can do a much better job than you might expect. By using various structured analysis techniques, we can avoid the typical pitfalls and instead decompose and externalize issues to arrive at a clear, well thought out hypothesis. Structured analysis is not necessarily a replacement for the quick, intuitive system 1 thinking, but rather a check and balance we place on it to keep it on track.

If you'd like to take a deep dive into additional structured analysis techniques, the ACH and other methods in this chapter are more clearly defined in Heuer's second book with Randolph Pherson.[1] The book is a master class in clear thinking and, as Heuer says himself, "…the techniques described in this book have wide applicability to … law enforcement intelligence analysis, homeland security, business consulting, financial planning, and complex decision making in any field."

[1] Heuer, R. & Pherson, R. (2015). *Structured Analytic Techniques for Intelligence Analysis*. Washington, DC: CQ Press.



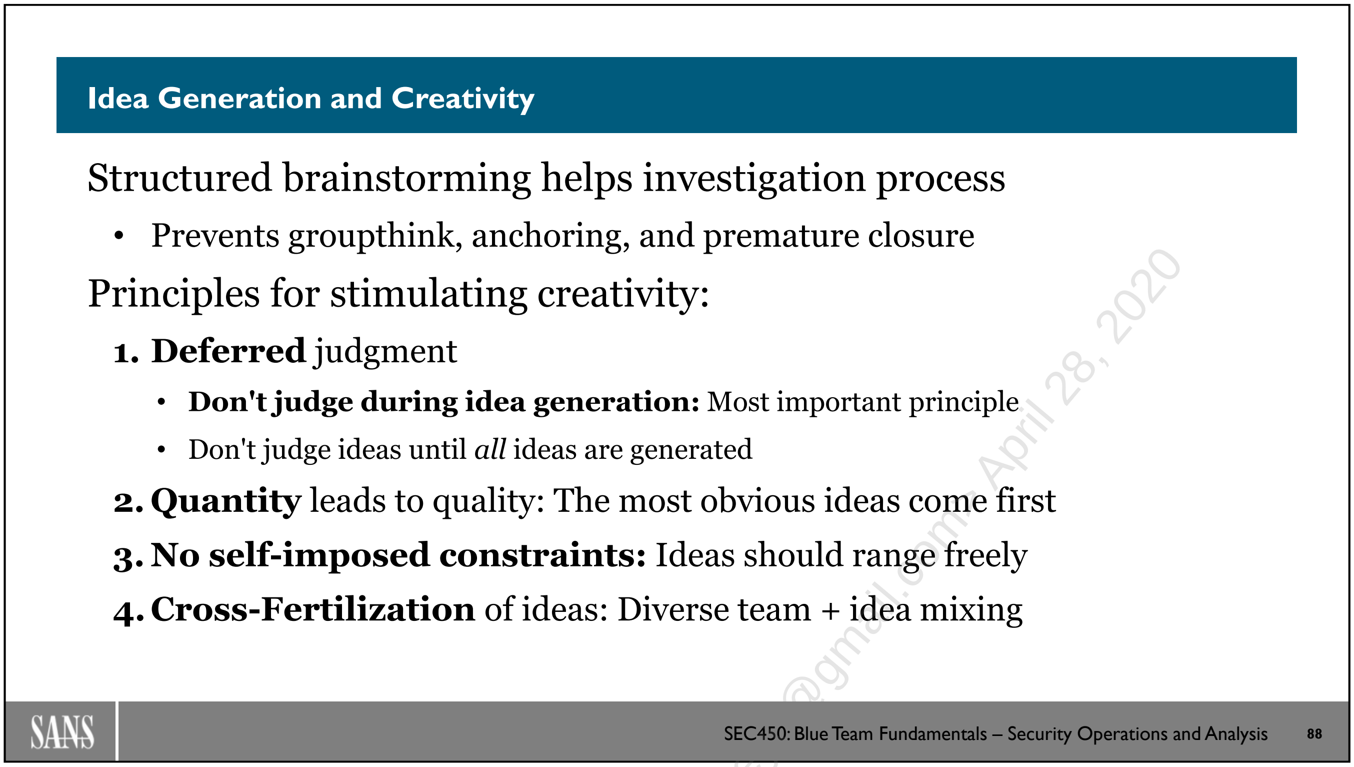
Categories of Structured Analysis

Structured analytical techniques fall into multiple categories:

- Idea Generation
  - Structured brainstorming and creative thinking
- Hypothesis testing
  - ACH, Diagnostic Reasoning
- Data Organization
  - Link Analysis, Event Matrices
- Challenge analysis (covered later)
  - Premortem, Self-Critique, What If?, Red Team, Team A/B

Categories of Structured Analysis

Structured analysis is not just one method. In fact, it covers multiple different categories and situations you may encounter. Throughout this module, we'll explore some of the techniques such as those for idea generation, and hypothesis testing, data organization and externalization. Later in the book, we'll also cover challenge analysis. All these methods can be used for problem solving in a variety of situations where you must make qualitative assessments with incomplete and ambiguous data.



Idea Generation and Creativity

Structured brainstorming helps investigation process

- Prevents groupthink, anchoring, and premature closure

Principles for stimulating creativity:

1. **Deferred** judgment
   - **Don't judge during idea generation:** Most important principle
   - Don't judge ideas until *all* ideas are generated
2. **Quantity** leads to quality: The most obvious ideas come first
3. **No self-imposed constraints:** Ideas should range freely
4. **Cross-Fertilization** of ideas: Diverse team + idea mixing

Idea Generation and Creativity

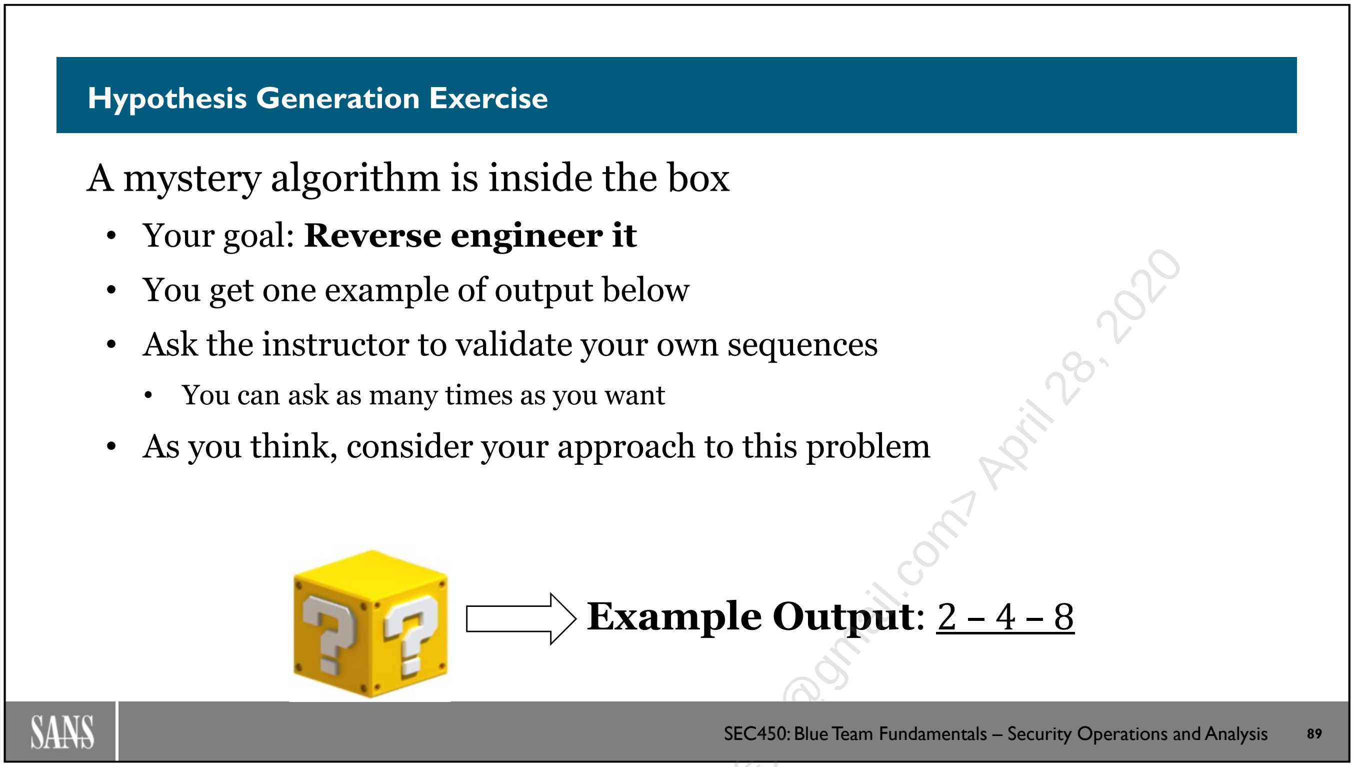
In performing analysis, analysts are required to generate new ideas, ask questions not yet considered, and find relationships between items that may have not been expected. This generation of new ideas requires creative thinking and an environment that fosters it. Structured brainstorming, the purposeful and methodical generation of ideas either by yourself or in a group, is a great tactic for creating inspiration and may be a common occurrence when trying to put together the pieces of a particularly difficult case. To be successful at structured idea generation, however, there are some principles outlined by Heuer that should be followed to ensure the environment in which you are trying to be creative doesn't sabotage your efforts.[1]

The first and foremost principle is deferred judgment. The separation of the evaluation of ideas from the stage where the ideas are generated is crucial for several reasons. Creativity and critical thinking are both necessary parts of the investigation process, but unfortunately the two do not mix well. Having team members shooting down ideas as they are generated leads to self-censorship and fear of criticism. The second principle is Quantity Leads to Quality. In order to come up with ideas that are truly unique, people must exhaust the list of obvious choices and simple explanations before starting to generate the genuinely unconventional material. Therefore, brainstorming must persist *at least* through the phase of all the obvious answers. Once these are eliminated, the new ideas will start to flow.

Another principle is not having any self-imposed constraints. This goes along a bit with the deferred judgment guidance. For the same reason no one should try to evaluate ideas on the fly as they are generated, you should not impose any limitations on yourself to bound ideas. Free flowing thought works best for creative idea generation. The final principle is cross-fertilization of ideas. The principle suggests that the options that are generated should be combined to form new thoughts and ideas. This naturally works best with more diverse ideas and, therefore, the diversity of the group generating the ideas will have an effect as well. Involving those who are not familiar with a case can inject fresh thoughts into the process.[2]

[1] Heuer, 1999. p. 76-78

[2] Ibid.



Hypothesis Generation Exercise

A mystery algorithm is inside the box

- Your goal: **Reverse engineer it**
- You get one example of output below
- Ask the instructor to validate your own sequences
  - You can ask as many times as you want
- As you think, consider your approach to this problem

Example Output: 2 – 4 – 8

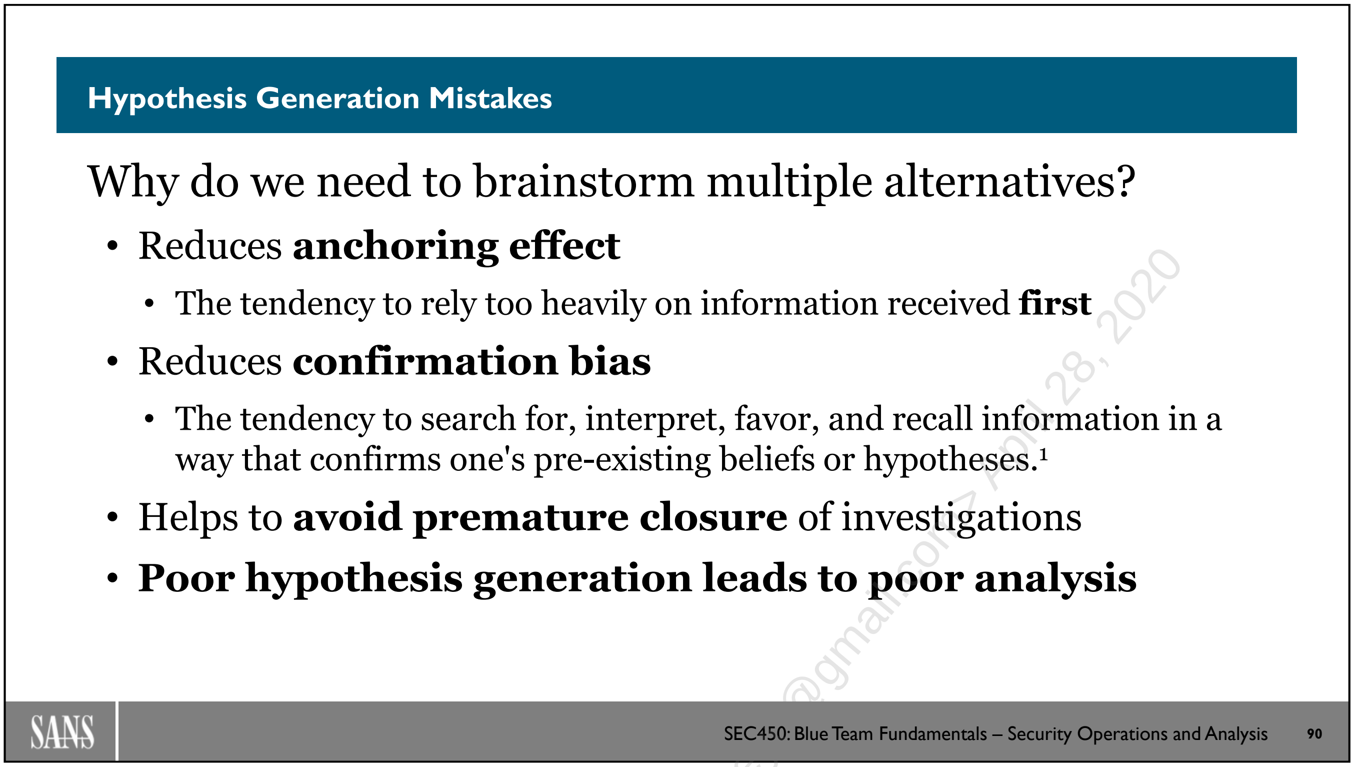
SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 89

Hypothesis Generation Exercise

Before we dive into the hypothesis generation section, let's do a warmup to check the process you use to come up with answers to a mystery. In the mystery box, we have a rule that can produce a set of numbers, your goal is to reverse engineer the algorithm to find out what it is. You can come up with as many of your own sets of three numbers to check if they are something the algorithm would produce or not and ask your teacher to verify them.

For those reading at home or OnDemand, you can do this without actually receiving the response. Just come up with an approach representing the line of questioning you *would* give to the instructor. The idea here is not necessarily about getting it right, but to analyze the method you take to narrow down the possibilities. How would you approach this problem? We'll describe the answer to this in a few slides…



Hypothesis Generation Mistakes

Why do we need to brainstorm multiple alternatives?

- Reduces **anchoring effect**
  - The tendency to rely too heavily on information received **first**
- Reduces **confirmation bias**
  - The tendency to search for, interpret, favor, and recall information in a way that confirms one's pre-existing beliefs or hypotheses.[1]
- Helps to **avoid premature closure** of investigations
- **Poor hypothesis generation leads to poor analysis**

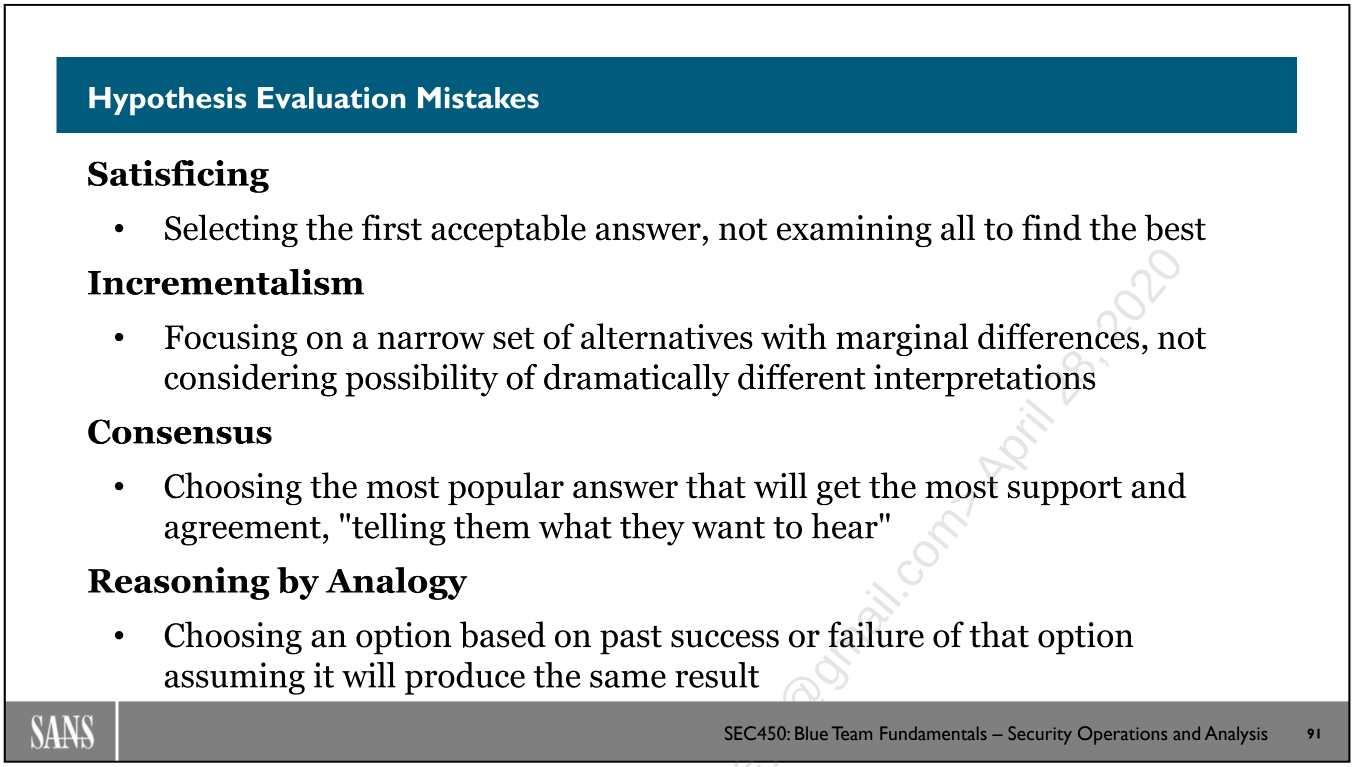
Hypothesis Generation Mistakes

Why is it so critical to produce multiple explanations to explain something instead of picking a favorite and running with it? Because there are two unconscious biases that particularly affect analysts when it comes to seeking truth: Anchoring effect and confirmation bias.

The anchoring effect is the tendency to rely too heavily on previous history of events or the first information that is received when judging a situation. An example of this phenomenon in the normal course of life often shows up in negotiations where whoever says a number first "sets the bar". All further discussions are judged relative to the initial numbers that were thrown out and the ending point tends to be highly dependent on where the initial number was set. For analysts, this bias shows up as a tendency to assume things are similar to alerts that have previously been seen or should match a hypothesis that was initially considered based on the first available evidence. We have a propensity to use initial information as an "anchor" point, and adjust from there, as opposed to re-evaluating information fresh, potentially leading to incorrect conclusions.

The second reason is confirmation bias. Although we have already mentioned confirmation bias previously, it is important to remember how it works since it is possibly the most relevant biases to the practice of analysis. This bias, combined with the anchoring effect, means we are most likely to come up with an initial theory, and give credit to evidence that confirms that theory while subconsciously ignoring or discrediting the information that might indicate our initial view was wrong. When these two effects combine, the results are premature ticket closure based on analysis with incorrect conclusions. Since our assessments can only be as good as the data provided and our analysis strategy allows, not letting the analysis strategy part of the equation get ruined by these cognitive biases is important, and that is what forcing yourself to generate multiple ideas accomplishes.

[1] Plous, S. (1993). *The Psychology of Judgment and Decision Making*. New York: McGraw-Hill, p. 233



Hypothesis Evaluation Mistakes

Satisficing

- Selecting the first acceptable answer, not examining all to find the best

Incrementalism

- Focusing on a narrow set of alternatives with marginal differences, not considering possibility of dramatically different interpretations

Consensus

- Choosing the most popular answer that will get the most support and agreement, "telling them what they want to hear"

Reasoning by Analogy

- Choosing an option based on past success or failure of that option assuming it will produce the same result

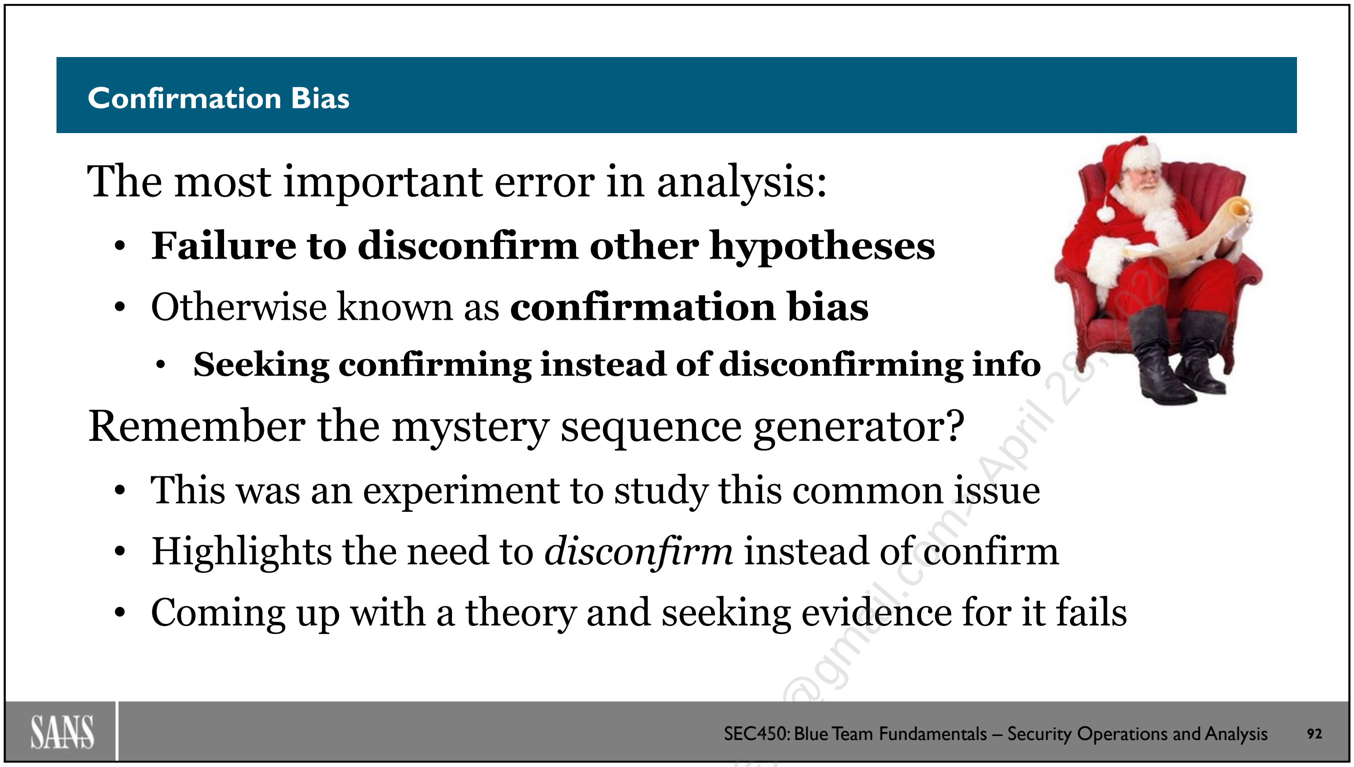
Hypothesis Evaluation Mistakes

According to Heuer[1], here are a few of the common strategies and mistakes made when performing analysis:

- Satisficing: We'll cover this one more in a bit because it is one of the most relevant to our use case. Satisficing is taking the first acceptable answer that one comes up with instead of doing a careful, reasoned analysis of all the options. This often goes hand in hand with confirmation bias because those who do it typically seek to find confirming evidence of that first acceptable theory, ignoring other options and the fact that the same evidence may work with other theories as well.
- Incrementalism: This is an error of being too narrow minded and only considering hypotheses that have minute differences. Those unwilling to entertain alternative but potentially wilder hypothesis may also totally miss the correct interpretation of data due to a hyper-focus on what they assume is going on.
- Consensus: This failure is choosing the answer that the analyst thinks will be the most acceptable and get the most support from others, instead of seeking to validate if an unpopular but true hypothesis may in fact be the right one.
- Reasoning by Analogy: This is based on a logical fallacy that since something has either worked or failed in the past, that taking the same route again will lead to the same conclusion. Of course, this *could* be true, but it does not necessarily logically follow. If this type of reasoning were sound, we might have never had companies like Tesla, because indeed most new car companies historically *have* failed. Elon Musk, however, is different (so far) and actually has specifically mentioned the failure of reasoning by analogy, stating that reasoning from first principles instead is one of the reasons for his success.[2]

[1] Heuer, 1999, p. 43

[2] https://www.businessinsider.com/elon-musk-first-principles-2015-1



Confirmation Bias

The most important error in analysis:

- **Failure to disconfirm other hypotheses**
- Otherwise known as **confirmation bias**
  - **Seeking confirming instead of disconfirming info**

Remember the mystery sequence generator?

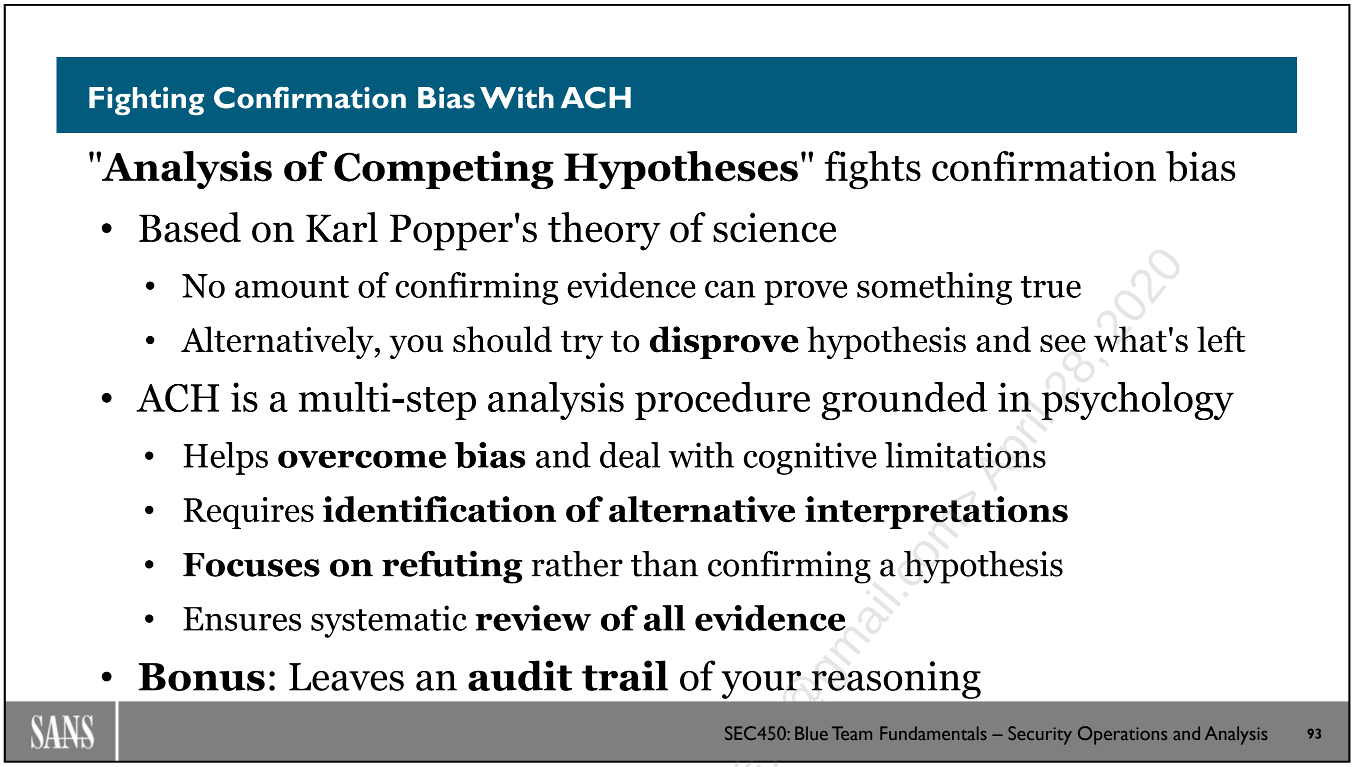
- This was an experiment to study this common issue
- Highlights the need to *disconfirm* instead of confirm
- Coming up with a theory and seeking evidence for it fails

Confirmation Bias

One of the biggest and most applicable failures to analysts, though, is the failure to consider and disconfirm other hypotheses that are also consistent with the data at hand. It's understandable why many do this. Without training in analytical methods, the natural approach many resort to is to consider a few hypotheses at the start of an investigation, pick the most likely one, then go forward seeking evidence to prove it is correct. Unfortunately, this falls into the trap of confirmation bias or, seeking evidence to prove what you already hope and expect to be true, instead of the better method of seeking to try to disconfirm it.

Now that we've discussed this as being one of the central issues in analysis, think back to the mystery algorithm a few slides ago. What was the answer? Any sequence that is three increasing numbers, that's it. This exact challenge was studied in psychological research and highlighted the fact that most people take a suboptimal approach when faced with a challenge like this. Since *so* many compatible sequences are possible, this scenario highlights perfectly the need to take the approach of disconfirming a hypothesis instead of finding evidence that *is* consistent with your first theory.[1] Go back and consider what your mindset and approach were for trying to figure this out. As you were guessing sequences, did you consider ways to prove a theory *was* true, or searching for ways to disconfirm ideas to take them out of the realm of possibility?

[1] Heuer, 1999, p. 46 - This exact challenge was studied by P.C. Wason in an article called "On the Failure to Eliminate Hypotheses in a Conceptual Task" published in *The Quarterly Journal of Experimental Psychology*, Vol. XII, Part 3, (1960).



Fighting Confirmation Bias With ACH

"**Analysis of Competing Hypotheses**" fights confirmation bias

- Based on Karl Popper's theory of science
  - No amount of confirming evidence can prove something true
  - Alternatively, you should try to **disprove** hypothesis and see what's left
- ACH is a multi-step analysis procedure grounded in psychology
  - Helps **overcome bias** and deal with cognitive limitations
  - Requires **identification of alternative interpretations**
  - **Focuses on refuting** rather than confirming a hypothesis
  - Ensures systematic **review of all evidence**
- **Bonus**: Leaves an **audit trail** of your reasoning

SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 93

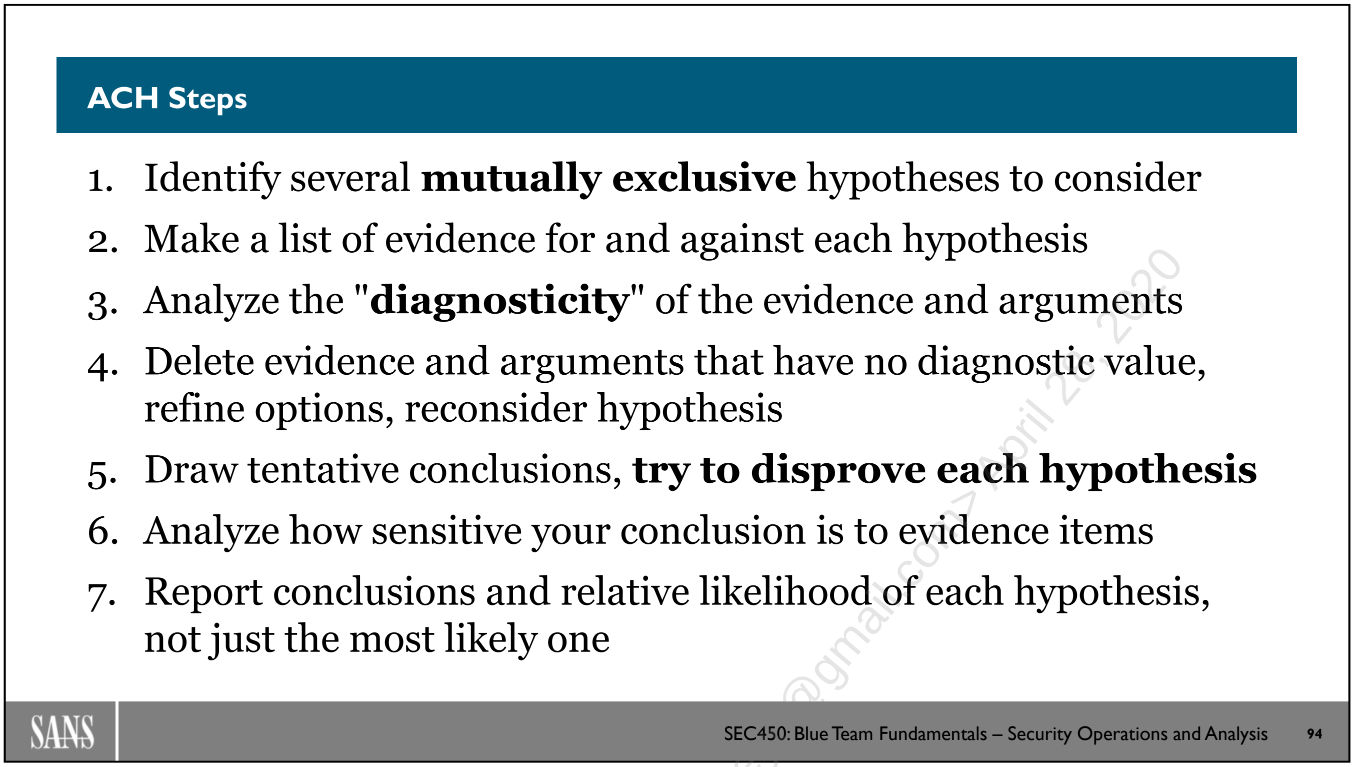
Fighting Confirmation Bias With ACH

One of the most well-known methods of structured analytical thinking, “Analysis of Competing Hypothesis", was made up by Heuer in his first book. Karl Popper, regarded as one of the 20th century's greatest minds in the philosophy of science, was a big proponent of "empirical falsification." In other words, when there are a set of possible hypotheses, designing specific experiments to try to *disprove* theories as a way of eliminating incorrect ones, acknowledging that no amount of confirming evidence can ever, in a strict sense, prove a theory true.

Inspired by this mindset, the Analysis of Competing Hypothesis can be viewed as Karl Popper's theory of science applied to the field of analysis. It is a multi-step procedure to help overcome common biases and involves identifying multiple mutually exclusive, alternative hypotheses and assessing each bit of evidence individually against each one in a matrix. In addition, it ensures a systematic review of all pertinent information, and leaves an audit trail that can be revisited if you need to explain your thinking to someone else in the future.

[1] Heuer, 199, pp. 95-110

[2] Heuer & Pherson, 2015, pp.180-191



ACH Steps

1. Identify several **mutually exclusive** hypotheses to consider
2. Make a list of evidence for and against each hypothesis
3. Analyze the "**diagnosticity**" of the evidence and arguments
4. Delete evidence and arguments that have no diagnostic value, refine options, reconsider hypothesis
5. Draw tentative conclusions, **try to disprove each hypothesis**
6. Analyze how sensitive your conclusion is to evidence items
7. Report conclusions and relative likelihood of each hypothesis, not just the most likely one

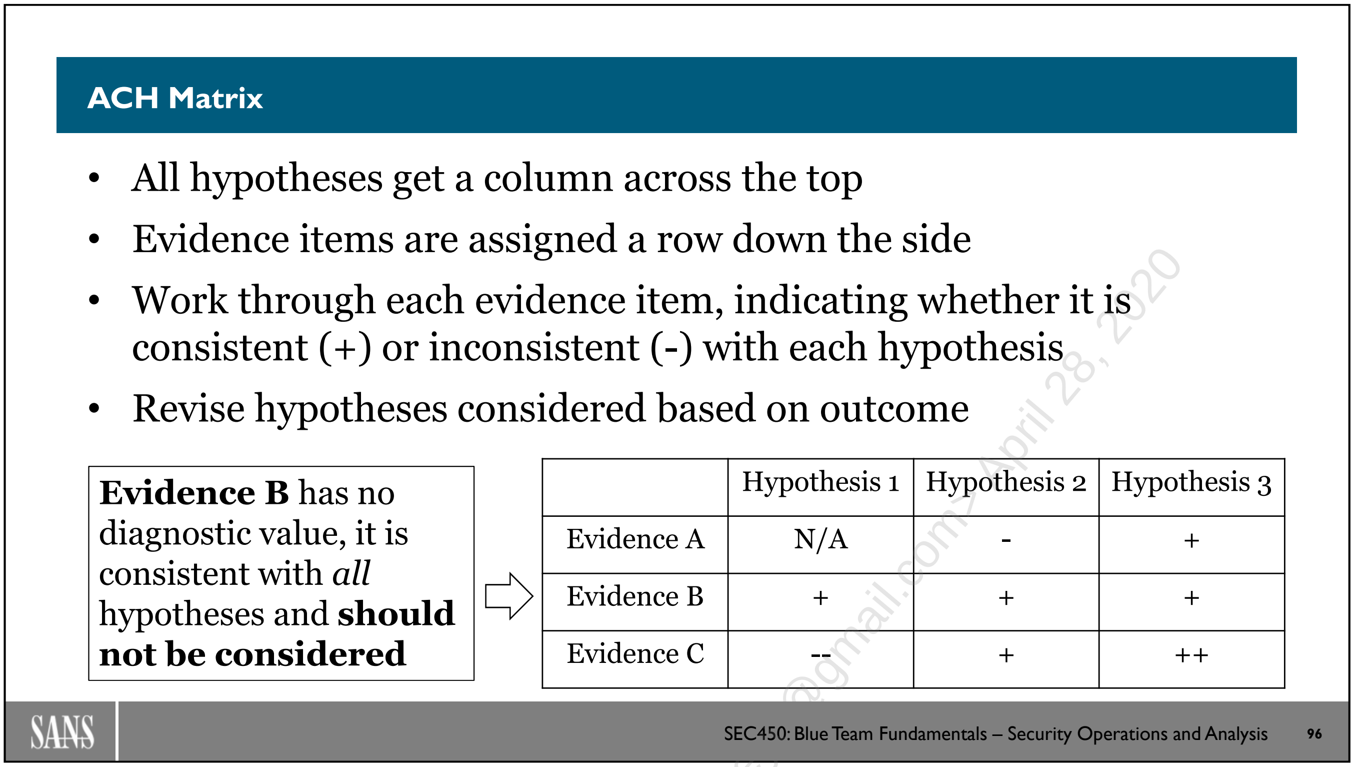
SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 94

ACH Steps

Here are the full steps of doing a complete Analysis of Competing Hypotheses. In daily life, will you really be doing all these steps for every single alert you take on? Likely not. It would take too much time to go into this level of depth for each alert. However, take the mindset here to heart and try to at least apply the principles of this process in each alert you do, whether it is an intuitive decision being made or not. Where you will likely use this full-on method is for alert and incidents that are high profile and need to have an extremely well-reasoned and auditable analysis performed.

1. The first step is to brainstorm several possible hypotheses. These should be mutually exclusive ideas such that if one is true, others must be false. It is best to come up with as many plausible options as possible, including a hypothesis with deception tactics, if such a thing is a possibility for the given situation.
2. List out all information relevant to evaluating each hypothesis. All evidence and assumptions should be included, including the absence of things that you would expect to see if a hypothesis were true. Assumptions can make a big difference in the judgment made, so they should be explicitly called out here so that others will know that it was included if the analysis is being reviewed in the future.
3. Place all hypotheses and evidence in a matrix (shown on the next slide) and run through each box to notate whether evidence is consistent or inconsistent with each hypothesis. If the answer is "it depends", the subsequent judgment for that column will be dependent on that "it depends", and this fact should also be noted.
4. Refine the matrix to eliminate data items that have no diagnostic value and adjust hypotheses as needed. If two need to be merged, or one should be removed, or added, this is the time to do so and rerun through each evidence item to update it for the changes.
5. At this point, tentative conclusions can be reached about which hypothesis is most likely based on which has the *least* amount of inconsistency scores for it. The ones with the most inconsistency scores are the least likely options. Remember these are not perfectly weighted rankings, so it will not be an exact science.

6. Analyze how the conclusions were reached. Are there any hypotheses that were ruled out based on a single item of evidence? How confident are you in that evidence? Is it an assumption? If so, it should be noted the conclusions are wholly dependent upon it.
7. Report conclusions on the likelihood of each hypothesis, not just the most likely one, especially if there are key assumptions or data the conclusion hinged on.



ACH Matrix

- All hypotheses get a column across the top
- Evidence items are assigned a row down the side
- Work through each evidence item, indicating whether it is consistent (+) or inconsistent (-) with each hypothesis
- Revise hypotheses considered based on outcome

Evidence B has no diagnostic value, it is consistent with *all* hypotheses and **should not be considered**

	Hypothesis 1	Hypothesis 2	Hypothesis 3
Evidence A	N/A	-	+
Evidence B	+	+	+
Evidence C	--	+	++

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 96

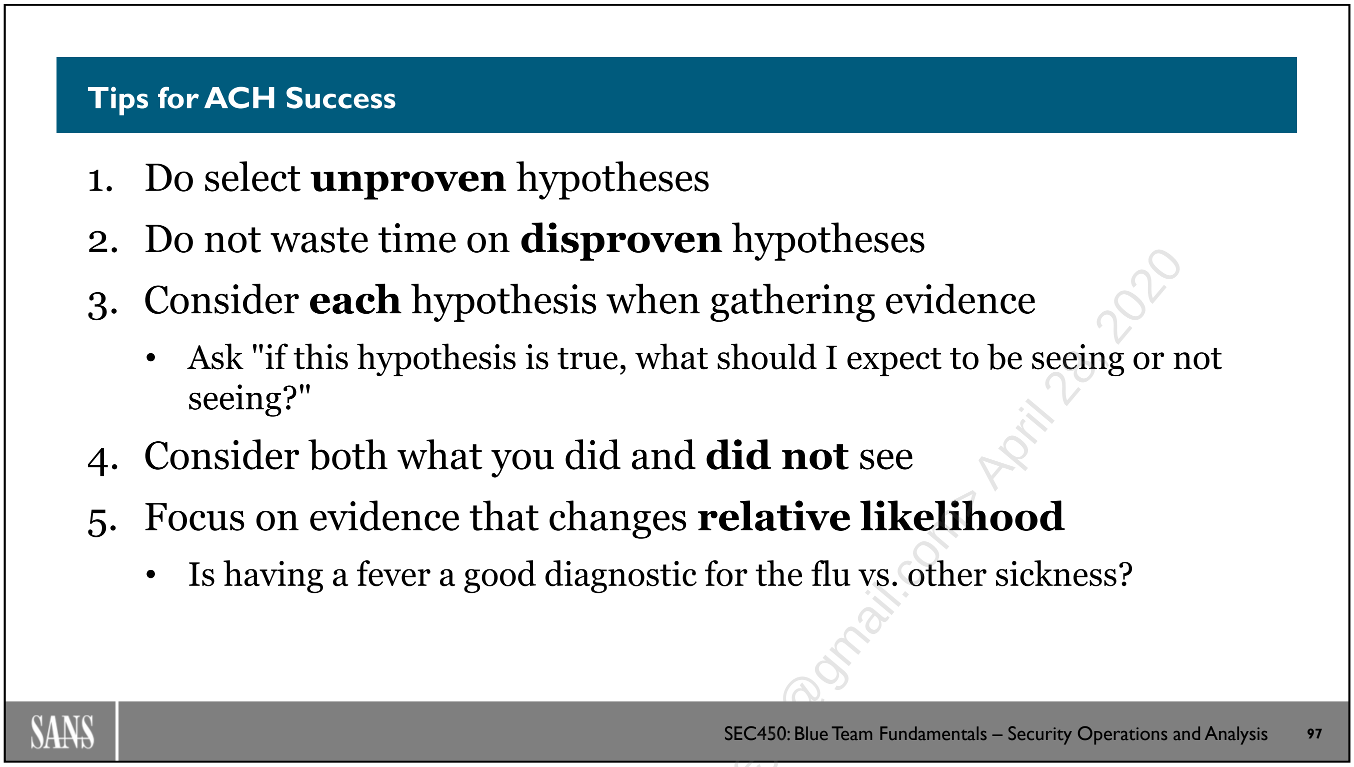
ACH Matrix

This slide shows the setup for a basic Analysis of Competing Hypotheses. We have 3 mutually exclusive potential explanations for what we are seeing, numbered 1, 2, and 3, and they are lined up across the top. Down the side, all the items of evidence have been placed, forming a matrix that will allow us to comment on each evidence item for each hypothesis. Afterward, each box must be filled in with notation that indicates whether that individual piece of evidence is consistent (+) with this hypothesis, is inconsistent (-), or not applicable. If a piece of evidence is particularly compelling in one way or another, -- and ++ can be used. Note that these can be done any way you see fit—it can be letters, symbols, a numeric order, or any other notation assuming it ranks the item properly.

Next, we must analyze the result. Let's look at the example results on the slide. This is where one of the most important and advantageous parts of this method comes into play—ranking the diagnosticity of each item of data. After filling out each item being consistent or not with each hypothesis, we can now see that Evidence B is consistent with all 3 hypotheses and, therefore, should not be considered in our analysis! This is the critical leap that is almost impossible to do with intuitive analysis. Even this simple 3x3 matrix is likely too complex for most to keep in their working memory. Given this information, we now know that we should only look at Evidence A and C. Reviewing the consistency of evidence A and B across the 3 hypotheses shows that the evidence is most consistent with Hypothesis 3 and, therefore, it is the most likely scenario. Though we can never prove a hypothesis to be correct, the single item of Evidence C can be enough to rule out hypothesis 1.

One thing we would want to note in our analysis is that Evidence C is a key piece of evidence for this case. It is the sole item that keeps Hypothesis 1 from being as valid as Hypothesis 3. Remember, it is not how many plusses a hypothesis has going for it—what is important is merely the lack of negatives. As Heuer and Pherson's book says, "A hypothesis that cannot be refuted should be taken just as seriously as a hypothesis that seems to have a lot of evidence in favor of it."[1]

[1] Heuer & Pherson, 2015, p.166



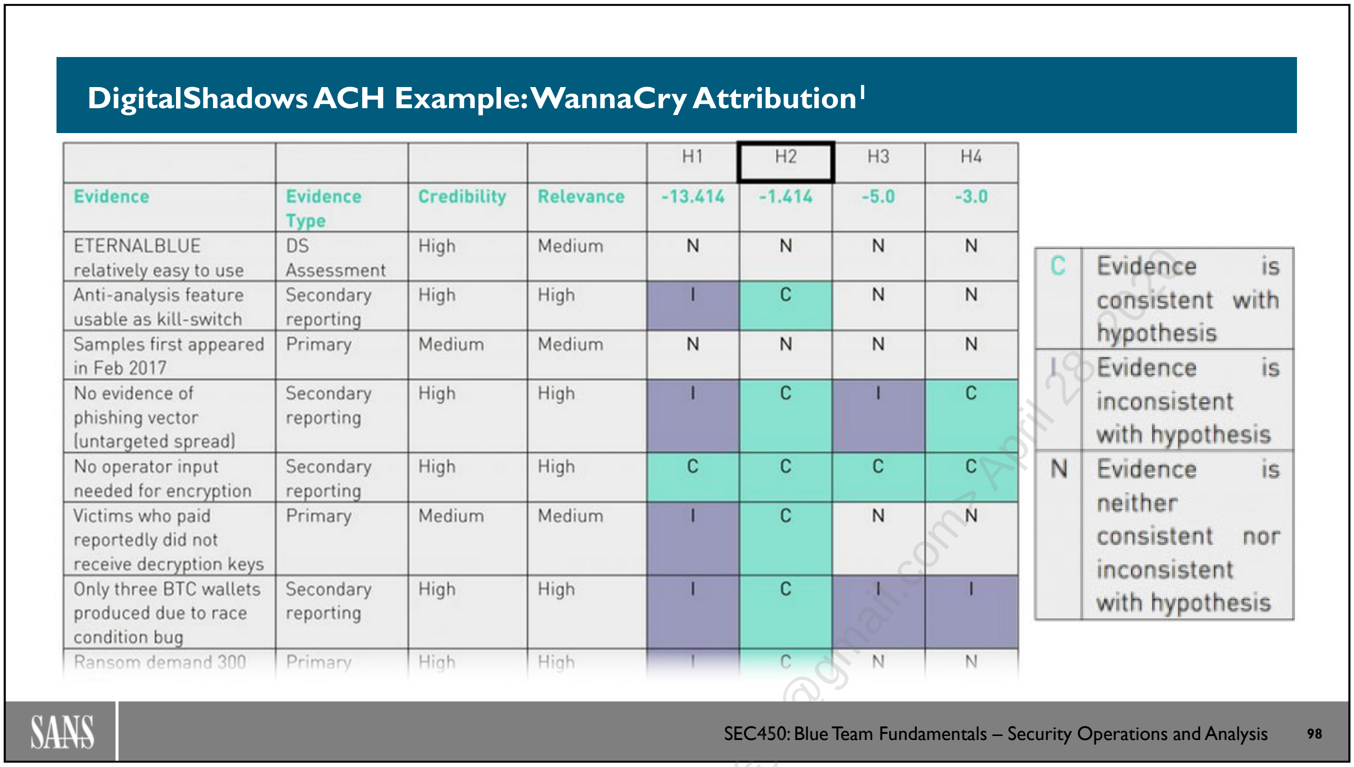
Tips for ACH Success

1. Do select **unproven** hypotheses
2. Do not waste time on **disproven** hypotheses
3. Consider **each** hypothesis when gathering evidence
   - Ask "if this hypothesis is true, what should I expect to be seeing or not seeing?"
4. Consider both what you did and **did not** see
5. Focus on evidence that changes **relative likelihood**
   - Is having a fever a good diagnostic for the flu vs. other sickness?

Tips for ACH Success

Here are some reminders for how to select and perform ACH that will help you be successful in your evaluations:

- When brainstorming a list of hypotheses, make a distinction between *disproven* and *unproven* hypotheses. One of the clearest examples of this is the deception theory. In most situations, you may have no reason to suspect deception, but it often is not ruled out either, meaning it is a valid hypothesis that should be considered. Chasing down wild ideas that are immediately obvious as inconsistent or *disproven* by the available evidence is likely not evaluating.
- When evaluating evidence, ask yourself, "If this hypothesis were true, what should I expect to be seeing or not seeing?" This can help you produce more entries for the ACH matrix that will further separate one hypothesis from another. Sometimes, *not* seeing something you would expect to see in a certain situation can be highly important.
- Focus on evidence that changes the *relative* likelihood of a hypothesis. It is the evidence that is both highly consistent with one hypothesis and highly inconsistent with another that will be able to influence your selection the most. Therefore, finding evidence that fits this description can improve accuracy. Bad evidence is evidence that is consistent with all selected hypotheses, making it near useless. An example of this would be someone going to the doctor with a fever claiming they have a rare tropical disease. Sure, a fever is consistent with that hypothesis, but it's also evidence for the flu and almost every other sickness on earth, meaning it's a poor piece of evidence to perform a diagnosis with. In other words, it has poor *diagnosticity*.



DigitalShadows ACH Example: WannaCry Attribution[1]

				H1	H2	H3	H4
Evidence	Evidence Type	Credibility	Relevance	-13.414	-1.414	-5.0	-3.0
ETERNALBLUE relatively easy to use	DS Assessment	High	Medium	N	N	N	N
Anti-analysis feature usable as kill-switch	Secondary reporting	High	High	I	C	N	N
Samples first appeared in Feb 2017	Primary	Medium	Medium	N	N	N	N
No evidence of phishing vector (untargeted spread)	Secondary reporting	High	High	I	C	I	C
No operator input needed for encryption	Secondary reporting	High	High	C	C	C	C
Victims who paid reportedly did not receive decryption keys	Primary	Medium	Medium	I	C	N	N
Only three BTC wallets produced due to race condition bug	Secondary reporting	High	High	I	C	I	I
Ransom demand 300	Primary	High	High	I	C	N	N

C	Evidence is consistent with hypothesis
I	Evidence is inconsistent with hypothesis
N	Evidence is neither consistent nor inconsistent with hypothesis

SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 98

DigitalShadows ACH Example: WannaCry Attribution[1]

If you'd like to see a great example of ACH as applied to the WannaCry incident, both DigitalShadows and Pasqual Striparo of the SANS Internet Storm Center created a matrix to evaluate the possibilities of who was behind the attack.[1] [2] The slide shows an excerpt from the DigitalShadows evaluation where they use the letters N, C and I as indicated to track consistency to each different hypothesis.

The hypotheses they used were as follows:

"*A sophisticated financially-motivated cybercriminal actor – H1*

An unsophisticated financially-motivated cybercriminal actor – H2

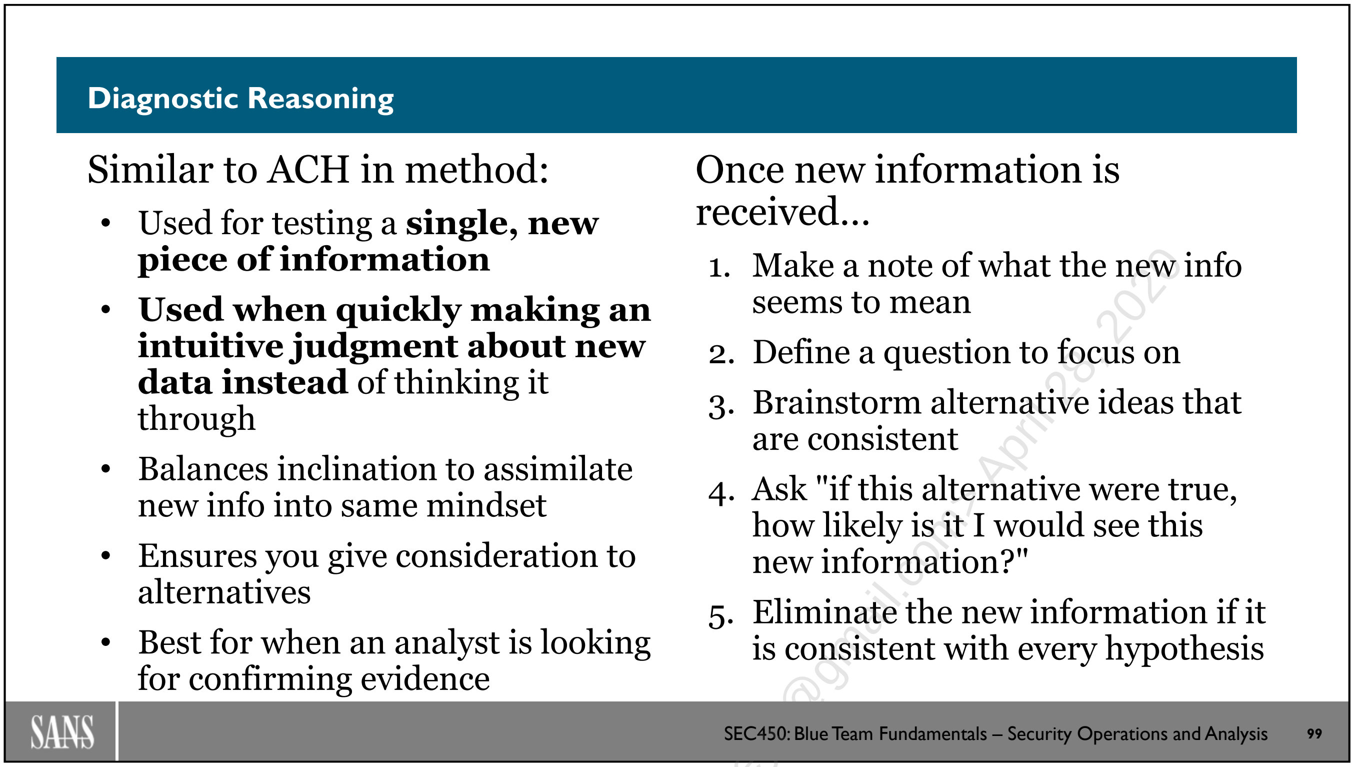
A nation state or state-affiliated actor conducting a disruptive operation – H3

A nation state or state-affiliated actor aiming to discredit the National Security Agency (NSA) – H4"

Their conclusion was that given all the evidence they had, H2—"an unsophisticated financially-motivated cybercriminal actor"—came out on top with H4— "A nation state or state-affiliated actor aiming to discredit the NSA"—close behind. Although many had attributed the attack to the claimed North Korean-based "Lazarus" group, this led DigitalShadows to ultimately come to the conclusion. "*At the time of writing, however, we assessed there to be insufficient evidence to corroborate this claim of attribution to [Lazarus] group, and alternative hypotheses should be considered.*" Of course, these conclusions are based on a rational, but still semi-subjective point ranking system.

[1] https://www.digitalshadows.com/blog-and-research/wannacry-an-analysis-of-competing-hypotheses-part-ii/

[2] https://isc.sans.edu/forums/diary/Analysis+of+Competing+Hypotheses+WCry+and+Lazarus+ACH+part+2/22470/



Diagnostic Reasoning

Similar to ACH in method:

- Used for testing a **single, new piece of information**
- **Used when quickly making an intuitive judgment about new data instead** of thinking it through
- Balances inclination to assimilate new info into same mindset
- Ensures you give consideration to alternatives
- Best for when an analyst is looking for confirming evidence

Once new information is received…

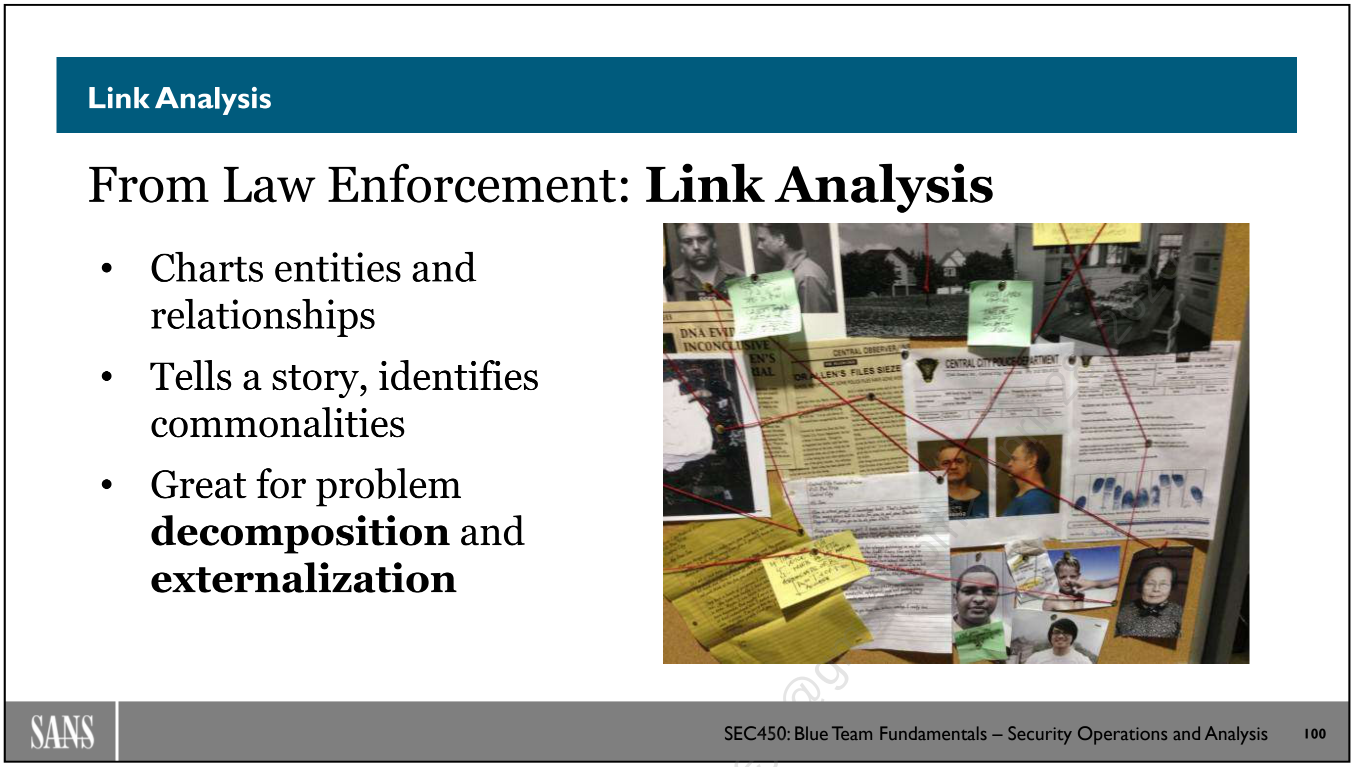
1. Make a note of what the new info seems to mean
2. Define a question to focus on
3. Brainstorm alternative ideas that are consistent
4. Ask "if this alternative were true, how likely is it I would see this new information?"
5. Eliminate the new information if it is consistent with every hypothesis

Diagnostic Reasoning

Another method of structured analysis is "diagnostic reasoning." Although it uses the same ideas as ACH, it is used not to select from many ideas, but to test the usefulness of a single, new piece of information that has been received. The goal is to eliminate the tendency to immediately assimilate the new information into your existing theory and force you to question it beforehand. It is best used when an analyst is looking for confirming evidence and potentially overlooking that the new bit of data might not be useful at all due to being consistent with other theories.

The steps are:

1. Once the information is received, make a note of what your first intuitive judgment was about it. What does it seem to mean and where does it seem to fit?
2. Define a question to focus on with the analysis such as "Is there a reason other than the lead hypothesis that…" that will help clarify the answer you are trying to tease out.
3. Brainstorm additional ideas where the new piece of evidence could conceivably be consistent with.
4. Ask yourself how likely you would be to see this new information if any of the alternatives in the previous step were to be correct.
5. If the new information seems to be consistent with many of the alternatives, it can potentially be dropped as it may not give any additional value over the information you already had. If it is *inconsistent* with alternatives, it may be used to rule those alternatives out.



Link Analysis

From Law Enforcement: **Link Analysis**

- Charts entities and relationships
- Tells a story, identifies commonalities
- Great for problem **decomposition** and **externalization**

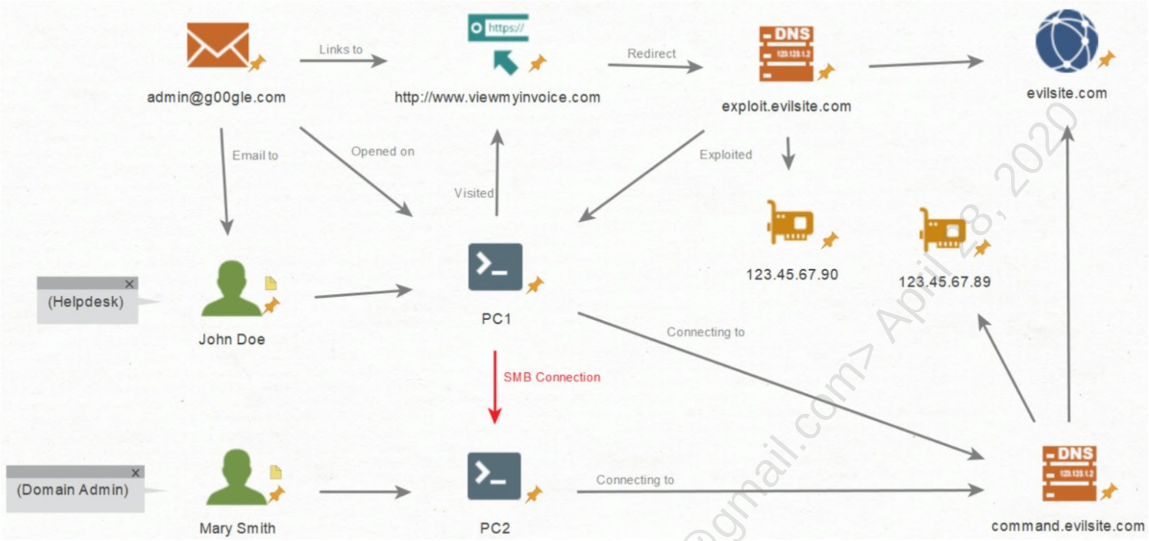
Link Analysis

One method we can draw on from the law enforcement world for data organization during analysis is Link or Relational Investigation.[1] In this method, all entities of various types related to the analysis problem at hand are put on a board, and connections between them are drawn. You can use the resulting graph to tell a story, identify commonalities among data items, and hypothesize connections you have not yet identified. In short, Link Analysis can help you organize and extract meaning from a set of data that is otherwise *far* too complex to keep in your working memory, making it an outstanding method for problem decomposition and externalization.

A link analysis chart can be created in several ways, depending on the tools or software you have available to assist you. The generic steps start with collecting all the data and constructing a matrix to enumerate the associations between all entities. This matrix or list can be used to either manually create, or ideally generate the graph with software. What comes out should assist you in interpreting and telling the story of what happened in your incident.

[1] https://www.unodc.org/documents/organized-crime/Law-Enforcement/Criminal_Intelligence_for_Analysts.pdf

Tools for Link Analysis: Maltego



Tools for Link Analysis: Maltego

Although there are certainly plenty of options for the link analysis method, there is one piece of commercial software called Maltego that is used very heavily for this within the information security world.[1] Maltego makes link analysis easy and allows "transforms" to be applied to entities within the chart, providing automatic enrichment such as IP and DNS lookups, Google searches, and *way* more. With the paid version of Maltego, tasks such as finding if 100 different domain names share and of the same IP addresses, ASNs, or netblocks is as simple as dragging a list of them to dump them all on the screen and right clicking to run the transforms for each enrichment. Maltego will automatically do all the resolutions and draw the IP addresses, ASNs, and netblocks as new entities, connecting them when there is overlap and making it easy to spot common infrastructure. As convenient as using it for analysis is, don't forget that outputs of link graphs also make great graphics to show management to explain the incident and for inclusion in incident reports. As shown in the example on the slide, they are a succinct way to show a complex set of relationships.

There are several different licensing levels from the free but limited Community Edition and Casefile, to the full featured "Classic" and "XL" licenses. At the time of writing, XL is $2,000 initially and $1,000 per year ongoing per seat, classic is half that. The Community edition can do many of the things the paid versions can do but will limit transform results to 12 each and graph save/output options. Casefile is free to use as many entities as you want and save items as you please, but transforms are totally disabled. For basic analysis, Casefile is a good place to start, since it will not restrict the graphs you draw.

For those who are looking for a Maltego alternative, a similar and completely free tool named yEd (pronounced "why Ed") can make very similar charts.[2] There's even a live web-based version of yEd that can be used for free without having to install the software!3 This can be useful for the occasional one-off graph, but, of course, it is incapable of doing the automatic "transforms" and data enrichment that Maltego can perform. Note that there is a lack of icons available by default when yEd is installed, but there is a built-in Palette management that has an easy one-click search and importing tool for finding additional icons online.

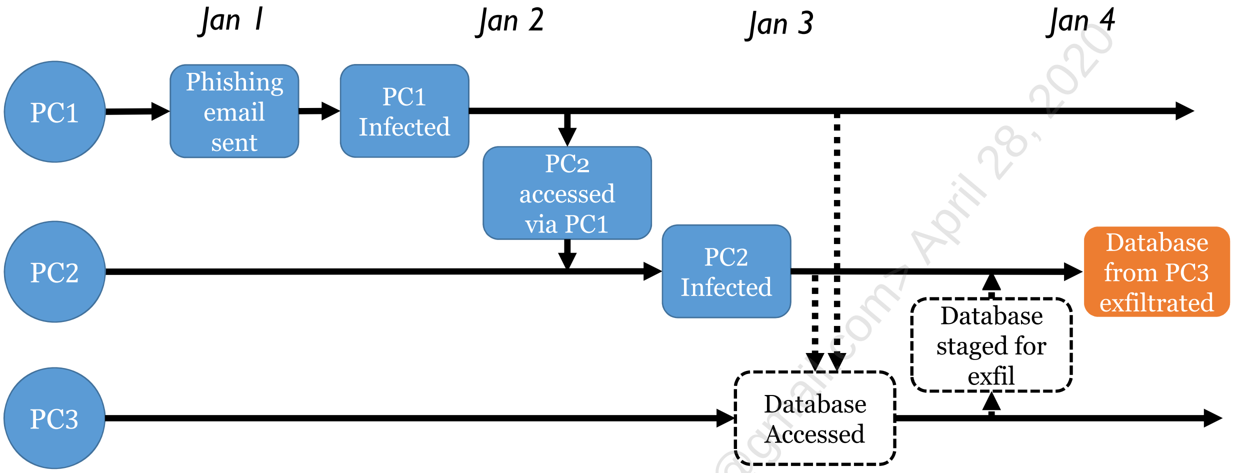
[1] https://www.paterva.com

[2] https://www.yworks.com/products/yed

[3] https://www.yworks.com/products/yed-live

Event Matrix Chart

Breach incident event chart:



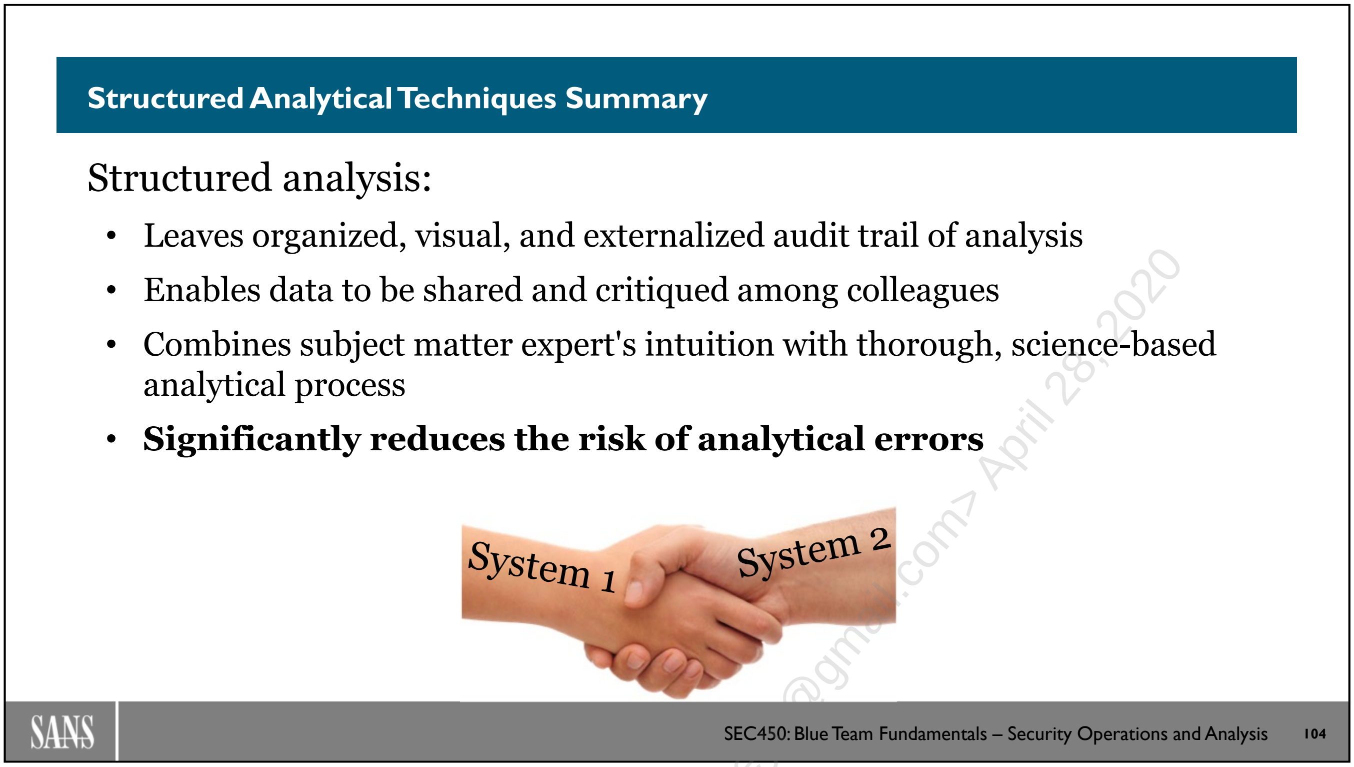
Event Matrix Chart

Event charting is a technique borrowed from law enforcement[1] where analysts lay out events on a chronological timeline. Although the chart itself may bear resemblance to a link analysis, it is laid out based on the timeline of events rather than the relationships between entities. In event charting, unlike link analysis, all entities must be the same object type (usually an event), and arrows show the progression of time as one event causes another. Multiple branches that merge or split may be used to show independent activity for different people or hosts and how they interact with each other or work together to cause some occurrence to take place.

This slide shows an event matrix style chart of a hypothetical breach incident. Each individual entity is represented on a horizontal line while dates are displayed along columns. Each event in the breach is aligned with the time it occurred and the asset it occurred on. Note how this type of chart can make sense out of a complex situation, and, also, potentially help guide where to look for key evidence. The series of events above the example show that both PC1 and PC2 were seen to be infected, then a database known to be saved on PC3 was exfiltrated. Even though it wasn't directly observed, we can guess the rough order of missing events and can predict that one of the two machines must have been used to access the PC3 to move the database back to PC2 for exfiltration. Given this, an analyst working the case might focus on looking for connections to and from PC2 and PC3 on January 3rd.

When the analysis first and foremost needs a timeline that is clear and common throughout all entities, this type of chart may be a good option. Don't forget that an event matrix can also make a great visual aid for final incident reports and updates to management as well.

[1] https://www.unodc.org/documents/organized-crime/Law-Enforcement/Criminal_Intelligence_for_Analysts.pdf



Structured Analytical Techniques Summary

Structured analysis:

- Leaves organized, visual, and externalized audit trail of analysis
- Enables data to be shared and critiqued among colleagues
- Combines subject matter expert's intuition with thorough, science-based analytical process
- **Significantly reduces the risk of analytical errors**

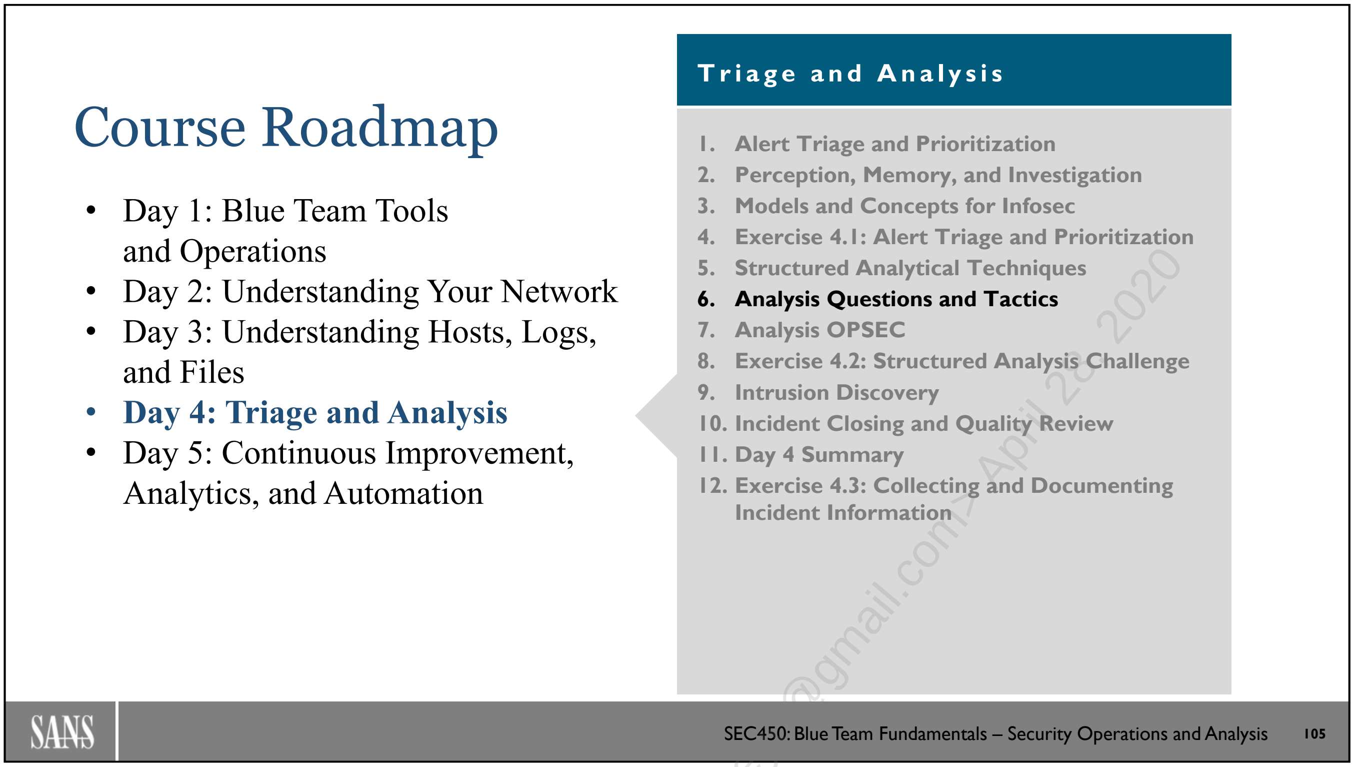
System 1
System 2

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 104

Structured Analytical Techniques Summary

While a big mindset change for some, learning structured analytical process and thinking can greatly improve your analytical capability as well as up the level of precision in investigations and documentation in the SOC. Thoroughly implemented, the ideas facilitate growth of so many positive skills such as science-based reasoning, brainstorming, critical thinking, and implementing regular peer feedback, that it's a shame it's not taught more often. While traditionally many of these techniques have been locked off behind government doors being applied only to the realm of intelligence analysis, we now have a wealth of publicly available knowledge and documented processes for running our own analysis in the same way. Combined with building subject matter expertise and the mindsets that go along with it, intuition and structured analytical thinking can go hand in hand to ensure our analysis is always efficient, logically sound, and well documented.



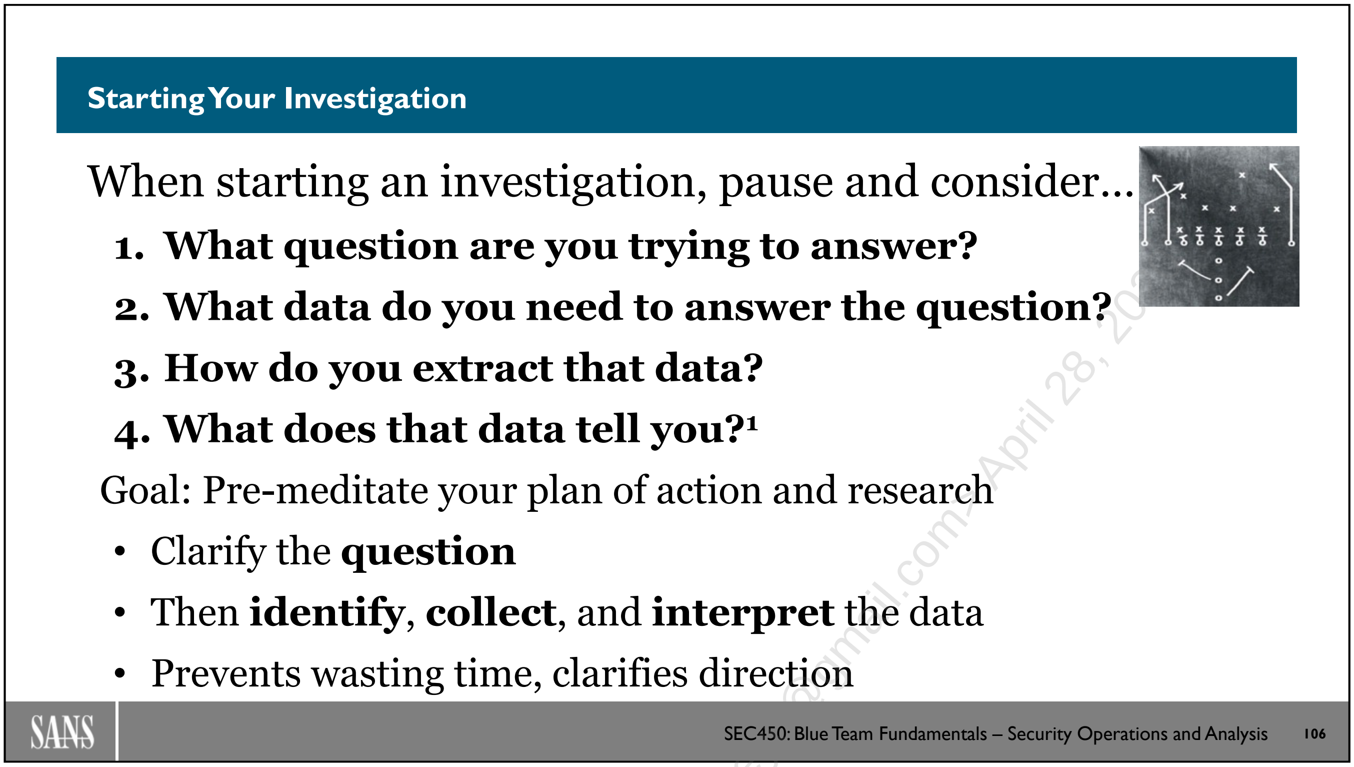
Course Roadmap

- Day 1: Blue Team Tools and Operations
- Day 2: Understanding Your Network
- Day 3: Understanding Hosts, Logs, and Files
- **Day 4: Triage and Analysis**
- Day 5: Continuous Improvement, Analytics, and Automation

Triage and Analysis

1. Alert Triage and Prioritization
2. Perception, Memory, and Investigation
3. Models and Concepts for Infosec
4. Exercise 4.1: Alert Triage and Prioritization
5. Structured Analytical Techniques
6. **Analysis Questions and Tactics**
7. Analysis OPSEC
8. Exercise 4.2: Structured Analysis Challenge
9. Intrusion Discovery
10. Incident Closing and Quality Review
11. Day 4 Summary
12. Exercise 4.3: Collecting and Documenting Incident Information

This page intentionally left blank.



Starting Your Investigation

When starting an investigation, pause and consider…

1. **What question are you trying to answer?**
2. **What data do you need to answer the question?**
3. **How do you extract that data?**
4. **What does that data tell you?**[1]

Goal: Pre-meditate your plan of action and research

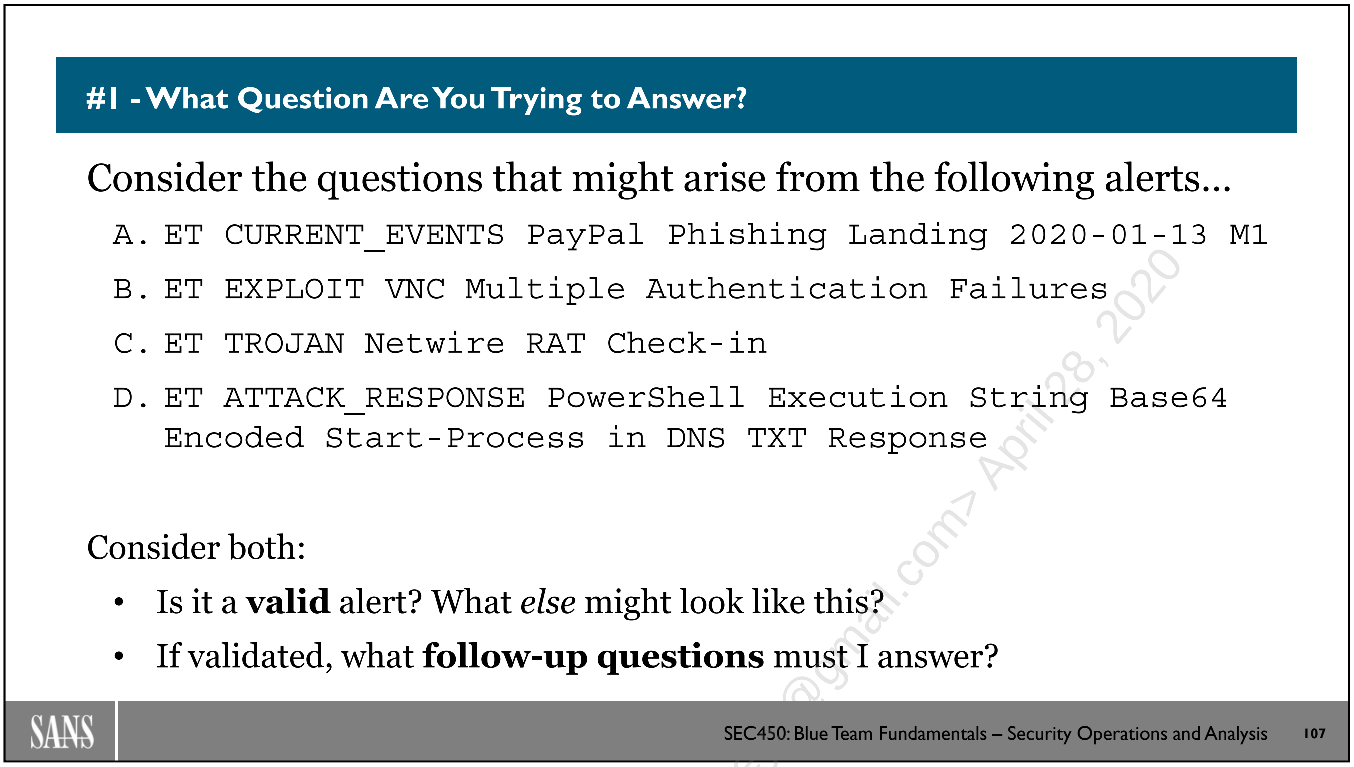
- Clarify the **question**
- Then **identify**, **collect**, and **interpret** the data
- Prevents wasting time, clarifies direction

Starting Your Investigation

Before you jump in too fast, though, consider exactly what data you're after and why. To do so, there's a nice framework in a blog post with set of questions called "The Alexiou Principle" (named after forensics expert Mike Alexiou, currently Director of Consulting at Elastic)[1] that you can ask yourself to make sure you have a crystal-clear vision of what you're trying to do. If you are finding you are unsure where to start in either a triage or analysis situation, ask yourself these smaller, more manageable, questions (notice this is *decomposition* of the problem):

1. What question are you trying to answer? Is it whether a device is compromised, what C2 site something is talking to, what type of malware it is?
2. What data do you need to answer the question? Consider the exact data you would need to answer that question in the most definitive way possible.
3. How do you extract that data? Where can you get that data? The host? The SIEM? Online OSINT research? PCAP? Go find it!
4. What does that data tell you? This question is what you will be answering through the subsequent analysis or triage work.

[1] http://thedigitalstandard.blogspot.com/2009/06/alexiou-principle.html



#1 - What Question Are You Trying to Answer?

Consider the questions that might arise from the following alerts…

```
A. ET CURRENT_EVENTS PayPal Phishing Landing 2020-01-13 M1
B. ET EXPLOIT VNC Multiple Authentication Failures
C. ET TROJAN Netwire RAT Check-in
D. ET ATTACK_RESPONSE PowerShell Execution String Base64
   Encoded Start-Process in DNS TXT Response
```

Consider both:

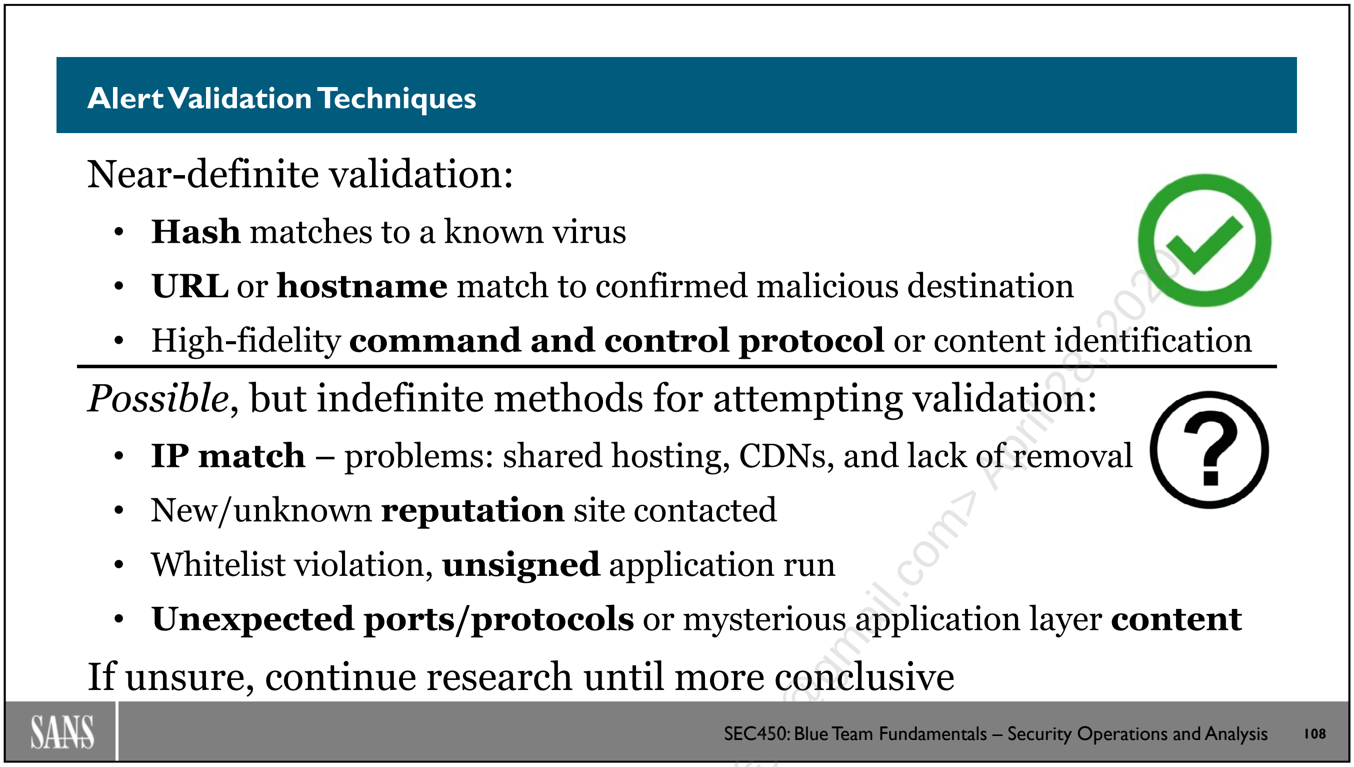
- Is it a **valid** alert? What *else* might look like this?
- If validated, what **follow-up questions** must I answer?

#1 - What Question Are You Trying To Answer?

Consider the alerts A-D and let's step through the proposed questions you might want to answer when faced with triaging these alerts. Step one is figuring out which questions we're trying to answer. One of the initial questions will always be "is this a valid alert?", after solving that question, consider the follow-on questions that would necessitate as well. Your playbooks should help guide you in these questions, but being able to produce them on your own is also important for those times you're outside the bounds of defined playbooks.

Here are some of the questions and follow-up items that would immediately come to my mind when faced with these items:

- ET CURRENT_EVENTS PayPal Phishing Landing 2020-01-13 M1
  - Did the user click a phishing link? From which email? Who else got it?
  - Did the user enter any information into the page?
- ET EXPLOIT VNC Multiple Authentication Failures
  - Was it really VNC? Do we allow VNC use? From this location? With this account?
  - Who was logged in at the time? What account was attempting to login?
- ET TROJAN Netwire RAT Check-in
  - Is this user infected with Netwire RAT? Is this traffic a check-in to a C2 site?
  - Was the traffic allowed/blocked? What process is running the virus?
- ET ATTACK_RESPONSE PowerShell Execution String Base64 Encoded Start-Process in DNS TXT Response
  - Did the user truly receive a PowerShell command in a DNS response? Is it the one the alert claims?
  - What process was started? What is the parent process that created this request? What site did it go to? Is anyone else talking to this site?



Alert Validation Techniques

Near-definite validation:

- **Hash** matches to a known virus
- **URL** or **hostname** match to confirmed malicious destination
- High-fidelity **command and control protocol** or content identification

Possible, but indefinite methods for attempting validation:

- **IP match** – problems: shared hosting, CDNs, and lack of removal
- New/unknown **reputation** site contacted
- Whitelist violation, **unsigned** application run
- **Unexpected ports/protocols** or mysterious application layer **content**

If unsure, continue research until more conclusive

Alert Validation Techniques

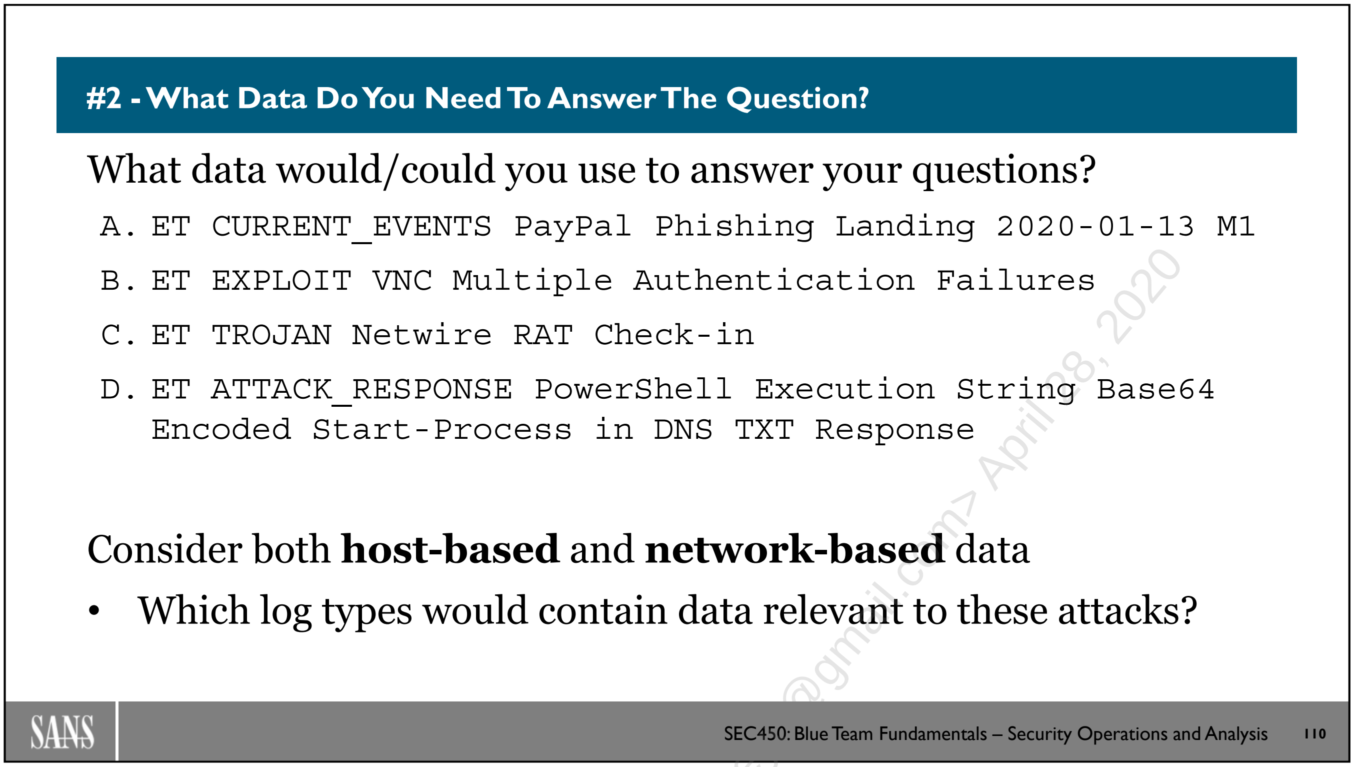
There are multiple quick ways to positively identify an alert as something worth further investigation. The best ways are the ones that provide a near proof-positive that something odd is happening.

- Alerts matching a known bad hash value are always worth turning into an incident for investigation. Since a hash is designed to uniquely identify a given file, any hash matches should be considered the highest level of confirmation that something bad did or is about to happen.
- Domains and URLs that are confirmed malicious through threat intel or OSINT can usually safely be turned into incidents as well. There are very few conceivable reasons other than an attempted attack that a computer would talk to a known bad domain.

Other signs aren't quite as clear but are usually worth pursuing as an incident. Often taking minor additional research steps can move these into the "definite problem" or false positive category:

- Known bad IP match: Depending on the IP address, these may be an immediate incident ticket, but unless the IP has only ever been used for bad and your threat intel indicates that, it usually takes some extra verification. The problem is many times threat intel vendors will find a malicious URL and mark it evil as well as the IP address that is hosting it. When a site is hosted on a shared hosting platform, it shares the same IP as many other websites, and writing an alert based on that IP means traffic to any of the hundreds or thousands of other sites on that IP gets caught in the net of the IP-based alert. Another potential issue is the number of sites that are behind sites like Cloudflare. Content delivery networks work by having site owners point DNS records to the CDN networks instead of the site server's true IP addresses so the CDN networks can perform their jobs, meaning any alerts written for a CDN network IP are going to also cause a *lot* of false positives. To make matters worse, many times IP addresses are never scrubbed from feeds meaning they just pile up over time making things worse and worse.
- Sites that are new and unknown by reputation services and OSINT sites alike are highly questionable. Unless a positive identification can be made, it is likely worth digging deeper into the interaction by opening an incident ticket.

- Unsigned, unknown applications may or may not be bad. But if your organization has taken steps to create a whitelist and block the downloading of additional executables in a dependable way, whitelist violations should not happen. If the application is unsigned, that makes it a bit more questionable as well.
- Unexpected port usage, unknown protocols and obfuscated or otherwise suspicious content inside known protocols should be considered a justified reason for investigation. If you find encoded content inside an HTTP POST or URL, or perhaps non-standard usage of other well-known protocols, it's likely worth making an incident ticket to investigate. Although there are good reasons you may see such a thing, the context of the destination of the traffic can be an easy differentiator whether it is of concern or not. Encoded traffic to a known business partner or software vendor may be OK, mysterious traffic to a mysterious domain is not.



#2 - What Data Do You Need To Answer The Question?

What data would/could you use to answer your questions?

A. ET CURRENT_EVENTS PayPal Phishing Landing 2020-01-13 M1

B. ET EXPLOIT VNC Multiple Authentication Failures

C. ET TROJAN Netwire RAT Check-in

D. ET ATTACK_RESPONSE PowerShell Execution String Base64 Encoded Start-Process in DNS TXT Response

Consider both **host-based** and **network-based** data

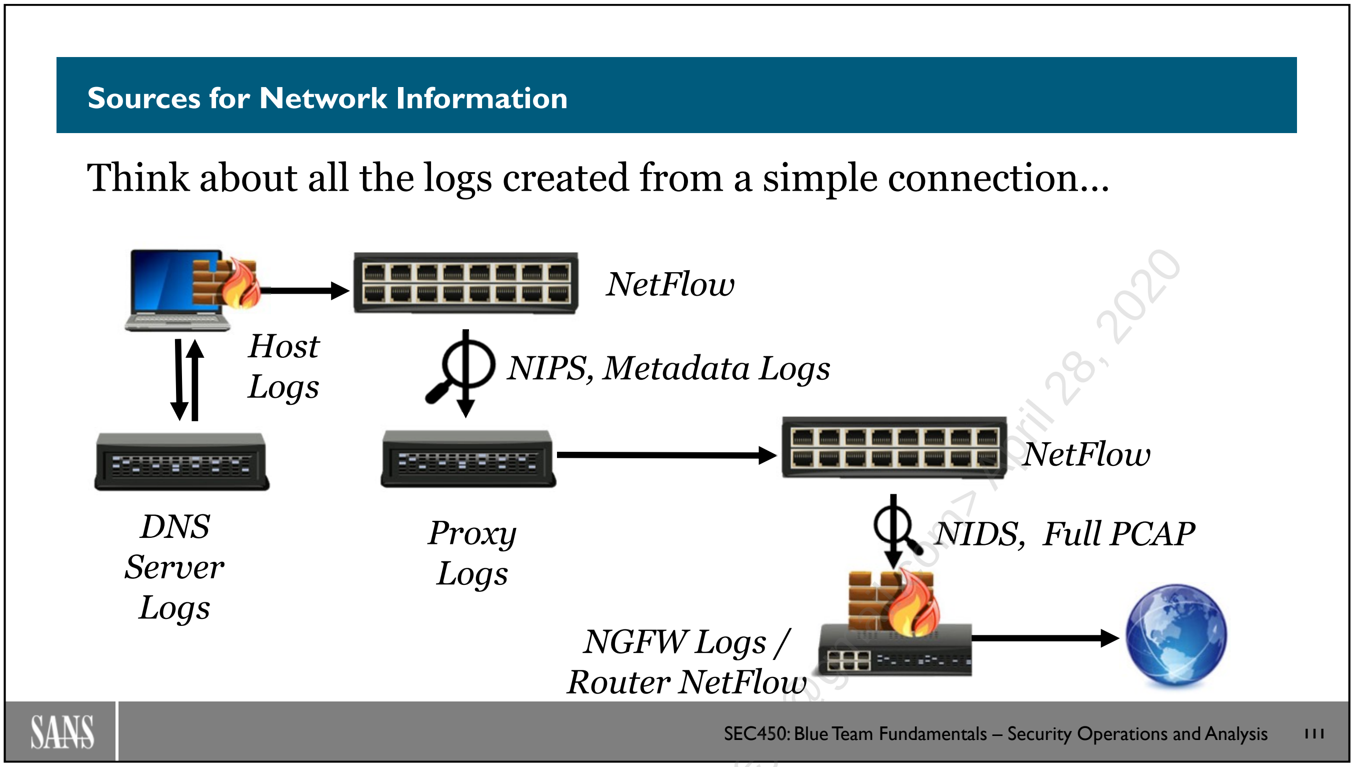
- Which log types would contain data relevant to these attacks?

#2 - What Data Do You Need To Answer the Question?

Considering alert A-D, what types of data could you use to first validate the alert, and then answer the follow-on questions you would generate if the alert were valid? This is where deeply understanding your data sources comes in handy. Being able to pinpoint exactly where the most useful source of data is will help you answer this part of the investigation quickly and find the answers you are looking for.

For the alerts, here are the data sources I would first think of:

- ET CURRENT_EVENTS PayPal Phishing Landing 2020-01-13 M1
  - HTTP(S) logs to confirm the user visited a malicious site
  - PCAP to see if the user typed anything into the phishing page
  - SMTP logs to find which email the link came from and who else received it
- ET EXPLOIT VNC Multiple Authentication Failures
  - PCAP / Zeek / NetFlow logs – Is it really VNC? Is it from an expected source? Was it truly a repeated failure, and how many were there?
  - Policy documents to check if VNC is allowed on the network
  - Authentication logs to see which account was used
  - User data to check details of the account failing to log in
- ET TROJAN Netwire RAT Check-in
  - PCAP/HTTP(S) logs to validate that the traffic does truly seem malicious
  - Process creation/whitelist/AV logs to locate the virus
- ET ATTACK_RESPONSE PowerShell Execution String Base64 Encoded Start-Process in DNS TXT Response
  - DNS Logs with responses to validate the alert
  - PowerShell logs to understand what commands were run



Sources for Network Information

Think about all the logs created from a simple connection…

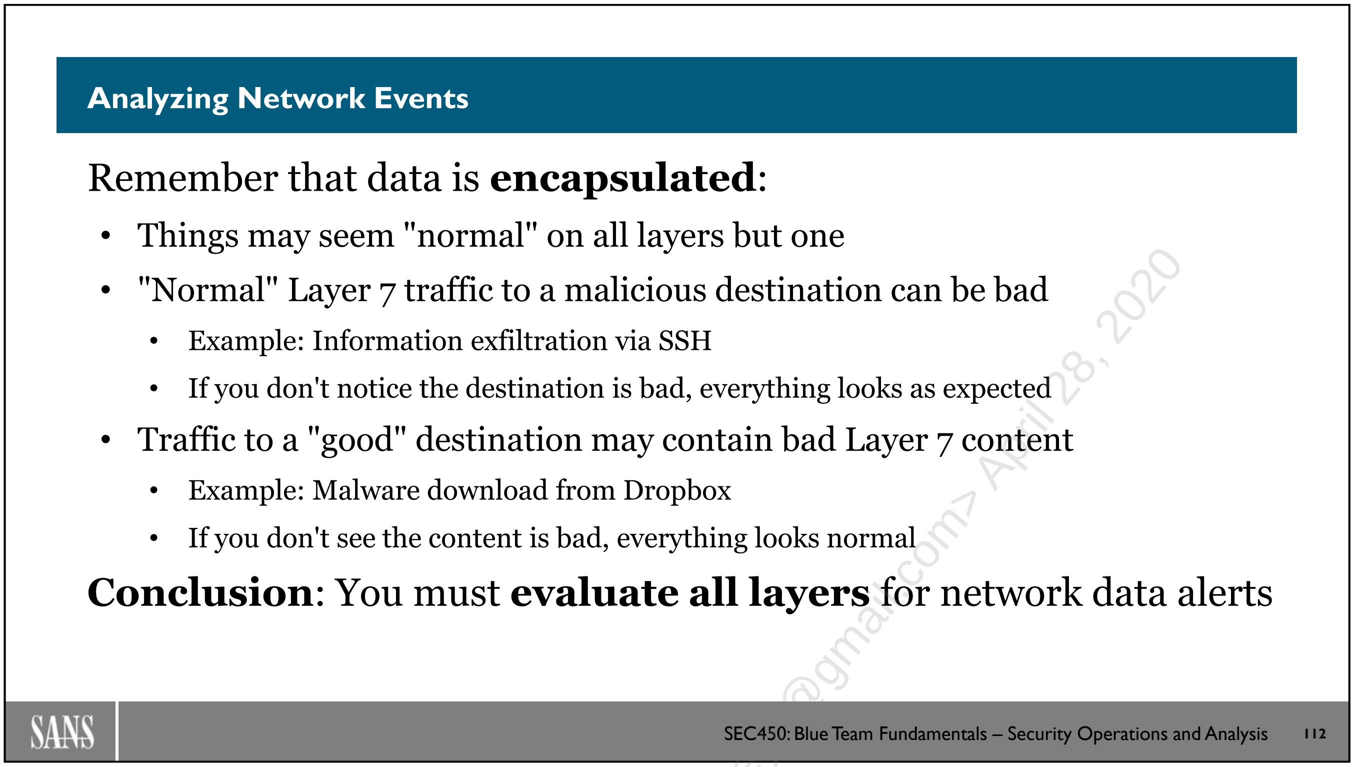
SEC450: Blue Team Fundamentals – Security Operations and Analysis 111

Sources for Network Information

Let's say you have a network-based alert that you need to chase down. How many options do you have for finding details on a device that connected a potentially malicious site.

1. URL is typed in browser (browser history is written, not centrally collected but works in a pinch)
2. DNS lookup is performed, logging the source, domain, and resolved IP (DNS cache written in Windows "ipconfig /displaydns")
3. Operating system logs connection in firewall logs, HIDS, EDR…
4. Traffic travels over the network to proxy, creating NetFlow, NIDS, or metadata logs
5. The proxy records the source and destination IP, and potentially the username as well.
6. Traffic exits the proxy and goes toward the firewall, where the connection is logged again
7. Full PCAP device records the whole transaction
8. Traffic leaves the network and NetFlow may be recorded again on the border router

That's a large list of places that are touched, leaving both intentional and unintentional evidence of the network connection and you could probably even come up with more in some scenarios! Although most people given this architecture may be inclined to go to the proxy as the one source of truth for internet connection logs, in the absence of that, there's still a wealth of ways information about the transaction can be obtained.



Analyzing Network Events

Remember that data is **encapsulated**:

- Things may seem "normal" on all layers but one
- "Normal" Layer 7 traffic to a malicious destination can be bad
  - Example: Information exfiltration via SSH
  - If you don't notice the destination is bad, everything looks as expected
- Traffic to a "good" destination may contain bad Layer 7 content
  - Example: Malware download from Dropbox
  - If you don't see the content is bad, everything looks normal

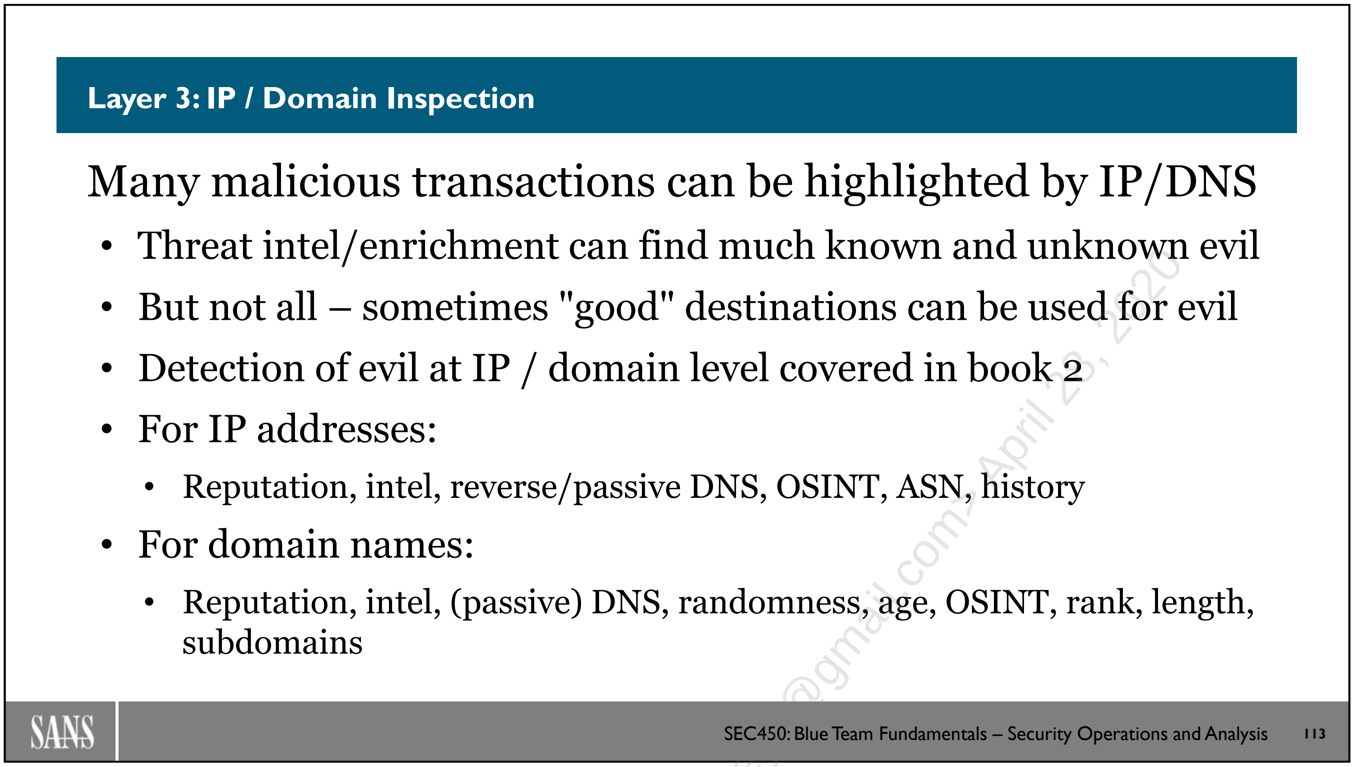
Conclusion: You must **evaluate all layers** for network data alerts

Analyzing Network Events

When analyzing any alert based on network traffic, remember that network traffic is highly *encapsulated* and might only present signs of being malicious on a single layer. For example, if an adversary is using SSH to exfiltrate data out of the network, you likely won't be able to decrypt it, so you won't know the Layer 7 contents is your stolen database. The port will likely be the standard 22, so how do you know anything bad is happening? Besides the "is SSH allowed and expected from this host" question, the network-based answer is the destination of that traffic—the Layer 3 content. If you know the SSH traffic is going to an IP address associated with an attacker or marked in a threat feed as having a bad reputation, this may be your only tip-off that something malicious is occurring. Finding this connection in your network metadata would be otherwise unlikely to raise suspicious if outbound SSH was allowed. It is only the association with the bad Layer 3 destination that gives us a clue.

On the flipside, sometimes things look completely normal all the way down through Layer 7, with the only anomaly being the content of the message the application layer protocol is transferring. For example, if a user is phishing with a link to malware hosted on a public Dropbox link, from your perspective, even with SSL decryption capability, this will be a connection to a good IP address and domain, using an expected port, using valid HTTP transactions with all normal HTTP headers. The only thing amiss is the fact that malware was transferred. If you were looking through network traffic and saw a Dropbox download and that is allowed under normal circumstances, the only way to identify the transaction as malicious is to examine the actual file that was downloaded.

The important takeaway here is that you never know which layer of network traffic the attacker will play games with. The encapsulated nature of traffic leaves many options open for using good protocols with bad intent or content, or good, protocol compliant transactions to bad destinations.



Layer 3: IP / Domain Inspection

Many malicious transactions can be highlighted by IP/DNS

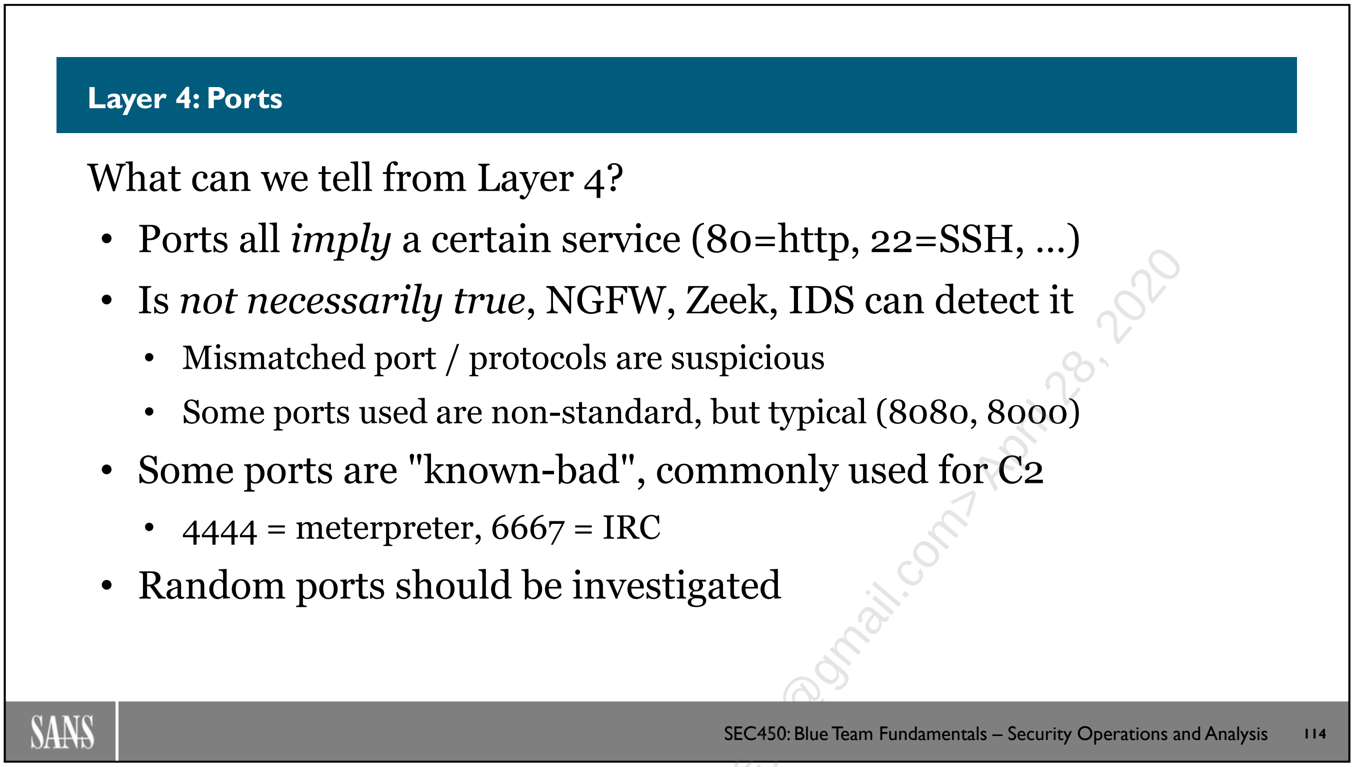
- Threat intel/enrichment can find much known and unknown evil
- But not all – sometimes "good" destinations can be used for evil
- Detection of evil at IP / domain level covered in book 2
- For IP addresses:
  - Reputation, intel, reverse/passive DNS, OSINT, ASN, history
- For domain names:
  - Reputation, intel, (passive) DNS, randomness, age, OSINT, rank, length, subdomains

SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 113

Layer 3: IP / Domain Inspection

We have already discussed the importance of IP and domain name enrichment and how a well-enriched log can easily highlight evil. Most of the time, attackers will like to work off their own infrastructure, giving them sure control of traffic to and from it. Utilizing a victim's server, DNS infrastructure, or file hosting leaves their operation subject to being shut down at any moment. Therefore, in many cases, the knowledge that an IP or domain name is bad might be all you need as a tip to start investigating traffic as malicious but doing this will rely on you having the IP/domain in a blacklist or detected as anomalous through enrichment. All SIEMs should have the capability to match against threat intel lists, but unfortunately they can't all easily perform the extensive enrichment required to find yet unknown evil sites, which means you will need to know how to do these lookups on your own. Beyond this, there are also ways for good services to be used for evil as well, meaning Layer 3 will look otherwise harmless, leaving you having to dig deeper to identify the malicious activity.

For IP addresses and domains, you should have a standard method for manual lookup of reputation, reverse DNS (ideally sourced from your own network as well), ASN, and the history of the indicator in your environment. The faster and easier these lookups can be done, the quicker you can decide if the interaction requires further investigation. Automation, threat intel, enrichment APIs, and one-click links can go a long way in this regard.



Layer 4: Ports

What can we tell from Layer 4?

- Ports all *imply* a certain service (80=http, 22=SSH, …)
- Is *not necessarily true*, NGFW, Zeek, IDS can detect it
  - Mismatched port / protocols are suspicious
  - Some ports used are non-standard, but typical (8080, 8000)
- Some ports are "known-bad", commonly used for C2
  - 4444 = meterpreter, 6667 = IRC
- Random ports should be investigated

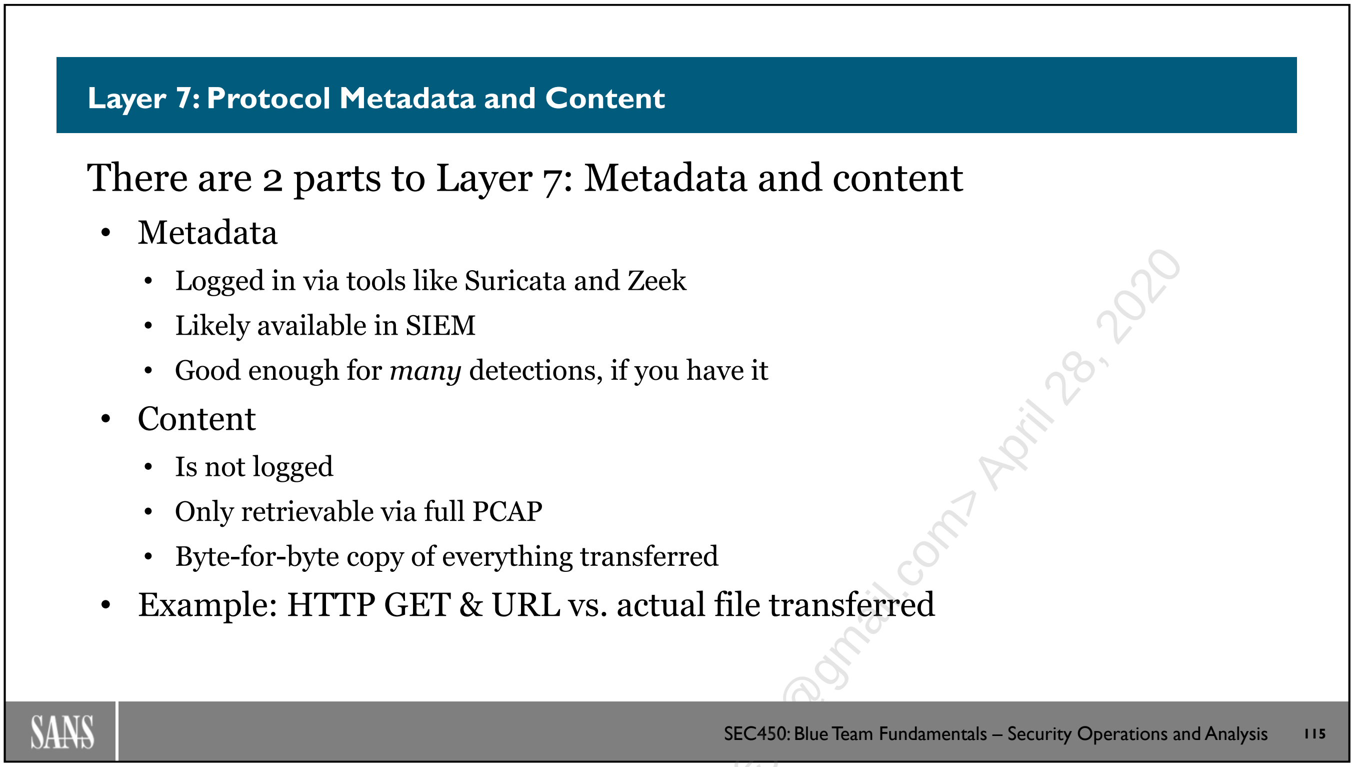
SANS | SEC450: Blue Team Fundamentals – Security Operations and Analysis 114

Layer 4: Ports

What if we see an alert using a port number that is atypical? There are a couple of ways this could break down, although port is only slightly correlated with badness overall. There are specific situations that may be higher fidelity than others. There are other ports that, under any circumstances, are likely to be some traffic you don't want. Port 4444 is an example of this. It is the default meterpreter port used with Metasploit. Port 6667 for IRC is another example of this. In almost all situations, organizations should be aware of usage of these ports as it's almost never repurposed for something else. Legitimate software authors know this would be a bad idea. These ports are generally safely assumed to be something that needs investigating when seen, but unfortunately things aren't always this clear.

Any time you see a port, an associated service is implied. When we see port 80, we assume HTTP, and so on. Knowing this is how many people (any security appliances) work, adversaries occasionally will run different protocols on known ports to slip them by. For example, if a malware author wanted to run an IRC- based command and control channel that still worked when an organization had outbound deny rules for port 6667 (IRC), they might run the same malware using port 80 instead. In almost all cases, port 80 will be allowed outbound and unless the organization can detect the anomaly, it is likely the command and control will go on unhindered. Therefore, although a port number *might* give us a clue as to what type of traffic something might be, we may need to look at Layer 7 to verify it. Tools such as IDSs and next-gen firewalls *should* point out non-conforming protocols on well-known ports to us so things like this can be spotted.

Finally, there is the potential occurrence of the use of a destination port that seems totally random. In this case, you should investigate. It may be a piece of software you are unaware of or malware that has decided to use a non-standard port for whatever reason. Keep in mind we're talking about the service-side listening port in most cases here. While the port an infected device uses as a source port sending traffic outbound will almost always be a random ephemeral port, it is the outbound traffic's *destination* port that will typically be the most interesting (unless the infected device is acting as the server). In summary, a bad thing can run on a "known-good" port, and a good thing can run on an unusual port, but not often a "known bad" port. In order to distinguish which is which, Layer 7 content will often be required, but port oddities and protocol mismatches can be a good tip-off as an investigation starting point.



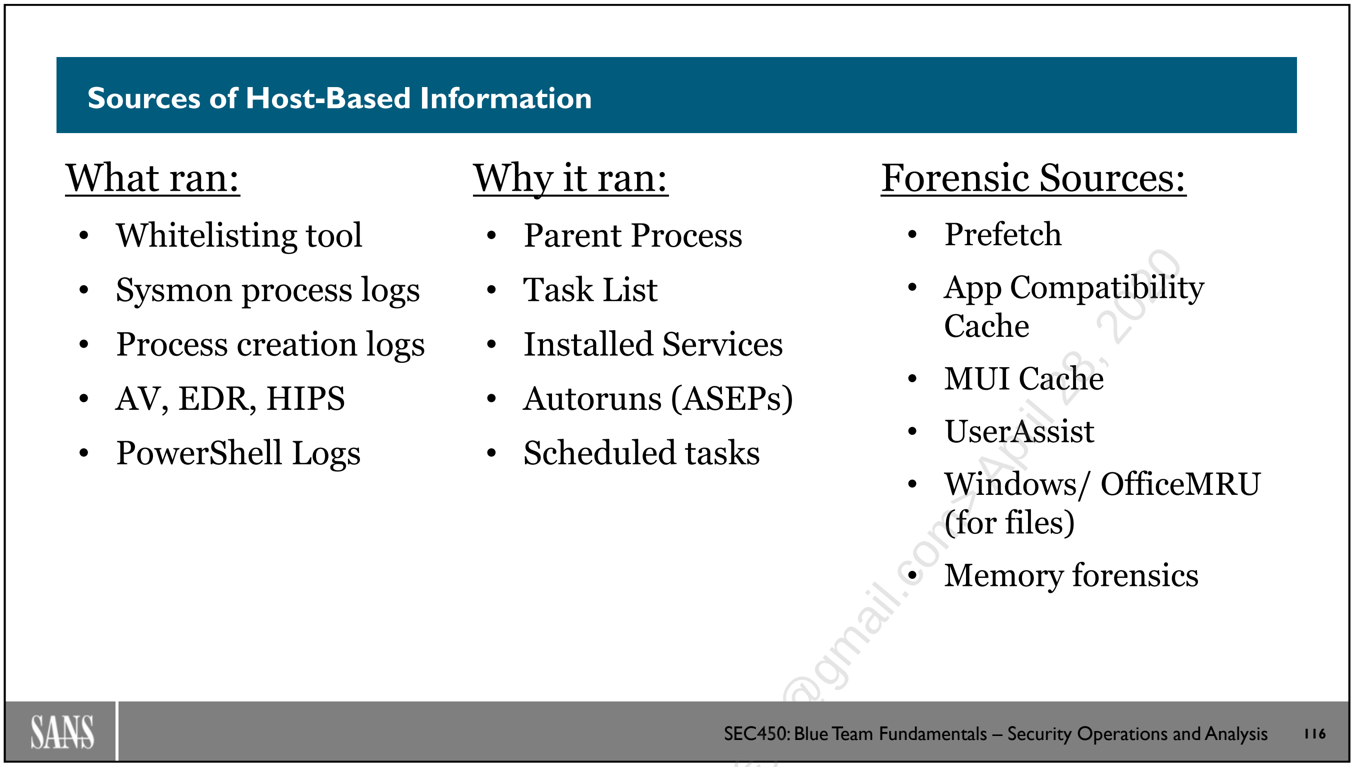
Layer 7: Protocol Metadata and Content

There are 2 parts to Layer 7: Metadata and content

- Metadata
  - Logged in via tools like Suricata and Zeek
  - Likely available in SIEM
  - Good enough for *many* detections, if you have it
- Content
  - Is not logged
  - Only retrievable via full PCAP
  - Byte-for-byte copy of everything transferred
- Example: HTTP GET & URL vs. actual file transferred

Layer 7: Protocol Metadata and Content

Getting all the way down to the bottom of the protocol stack, we finally hit the application layer. This is where you will see the HTTP methods, FTP commands, or other specifics of the protocol being spoken, regardless of the port or destination IP. If you have security tools such as Zeek and Suricata taking metadata for this layer, you will have outstanding levels of visibility giving you the information about file names transferred, user-agents, cookies, and any other fields your parser is able to extract. In addition, you will have any other layers between 4 and 7 that your tools can parse out (such as SSL certificates), adding additional detection options. This is a great place to start when looking at Layer 7 since it won't overwhelm you with the actual bytes of the traffic. You may find you can identify a known malicious user-agent or obvious command and control pattern in the URL. While having metadata for Layer 7 will provide an outstanding opportunity for detection, it *still* maybe will not be enough for the detection of evil. Sometimes, you will need to go all the way down the packet level to inspect the contents of the data transferred with the Layer 7 protocol. Back to our previous example of malware downloaded from "good" sites like OneDrive and Dropbox. Situations like this are often where the capability of the SIEM ends. You *may* be able to tell a file is suspicious from the filename downloaded in the URL, but to know for sure, you will be reliant on a decoded version of the full packet captured with a tool like Moloch.



Sources of Host-Based Information

What ran:

- Whitelisting tool
- Sysmon process logs
- Process creation logs
- AV, EDR, HIPS
- PowerShell Logs

Why it ran:

- Parent Process
- Task List
- Installed Services
- Autoruns (ASEPs)
- Scheduled tasks

Forensic Sources:

- Prefetch
- App Compatibility Cache
- MUI Cache
- UserAssist
- Windows/ OfficeMRU (for files)
- Memory forensics

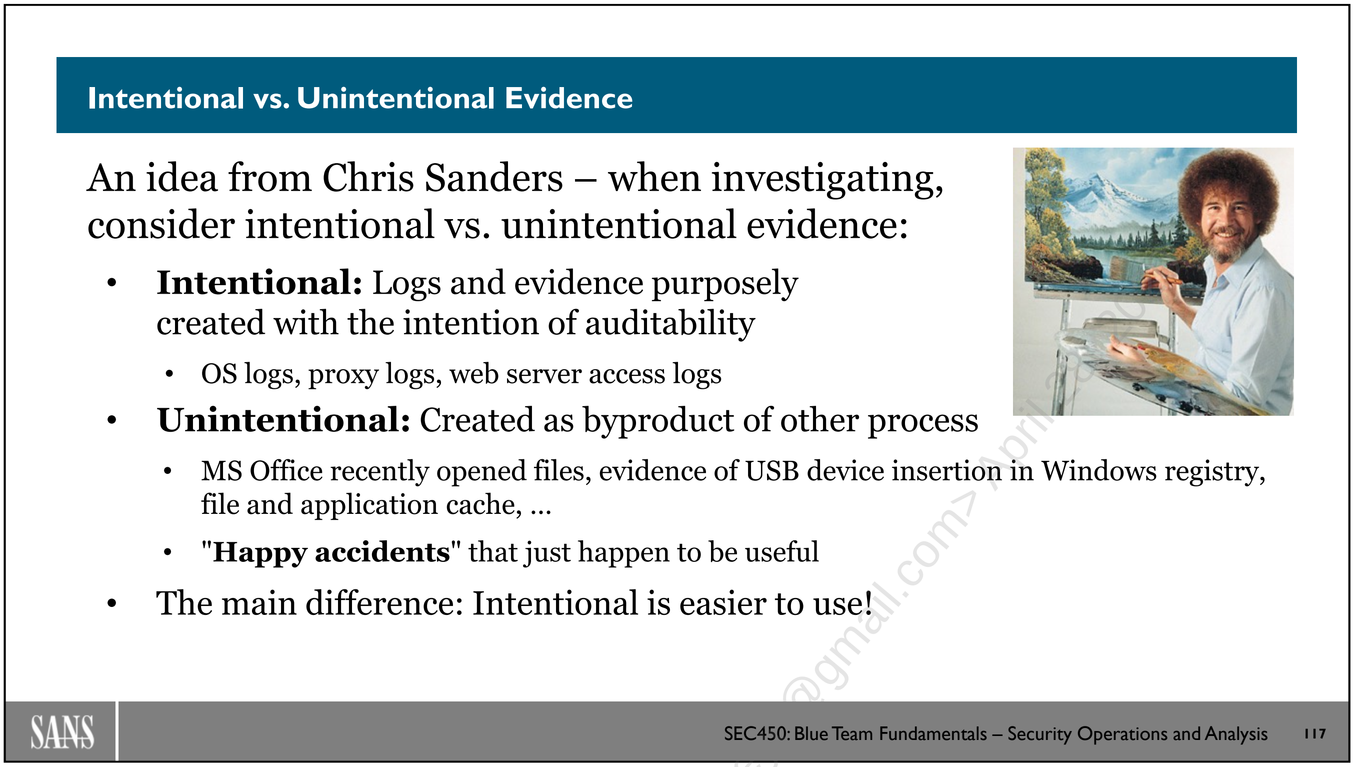
SEC450: Blue Team Fundamentals – Security Operations and Analysis 116

Sources of Host-Based Information

What about where to get information about a process that is running or file that was opened? There are an incredible number of intentional sources of information about running processes available on a Windows system. For example, any time a process executes, it may leave evidence in your whitelisting tool, AV, HIPS, EDR, Sysmon logs, process creation logs, and more. There is also a wealth of unintentional sources of data as well, such as the ones used for many forensic investigations. The forensic options listed on the slide, although meant to help Windows perform other function not related to process auditing, are an outstanding way to understand when the first time a virus was run and other information critical to some investigations. Although we don't have space to jump into forensic methods, there is a good article introducing what items are available from FireEye[1] and grabbing the information can be automated through tools like Dave Hull's Kansa PowerShell IR framework.[2] Aside from what ran, we can start to figure out why it ran by looking at tasks, services, autostart items (ASEPs) or even PowerShell logs. This can help us understand if something is a typical or new system service or task, and what its parent process or reason for running might be.

[1] https://www.fireeye.com/blog/threat-research/2013/08/execute.html

[2] https://github.com/davehull/Kansa



Intentional vs. Unintentional Evidence

An idea from Chris Sanders – when investigating, consider intentional vs. unintentional evidence:

- **Intentional:** Logs and evidence purposely created with the intention of auditability
  - OS logs, proxy logs, web server access logs
- **Unintentional:** Created as byproduct of other process
  - MS Office recently opened files, evidence of USB device insertion in Windows registry, file and application cache, ...
  - "**Happy accidents**" that just happen to be useful
- The main difference: Intentional is easier to use!

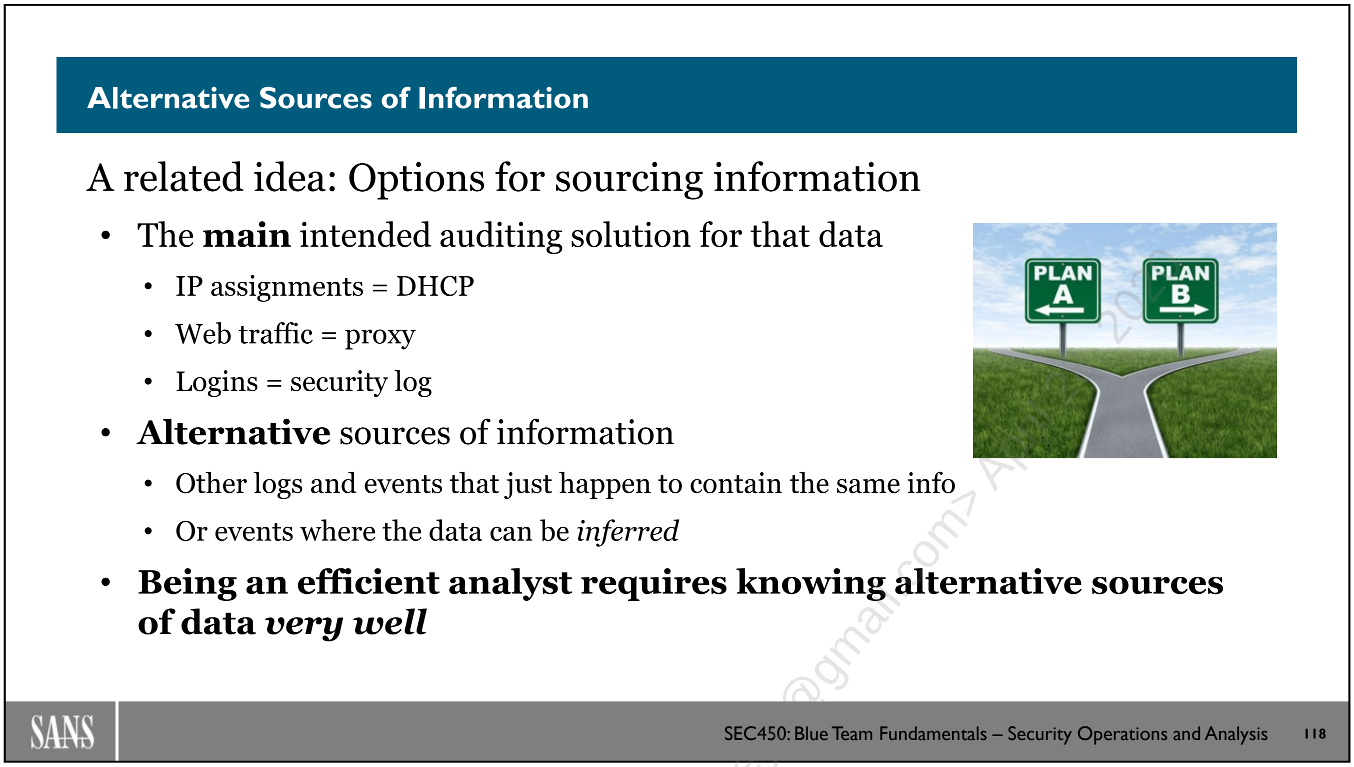
Intentional vs. Unintentional Evidence

One concept that is important for new analysts to understand is the role of evidence intention. This idea is described in a great blog post by Chris Sanders.[1] When doing an investigation, there are a multitude of sources you may consult to try to piece back together the set of events that led to a potential compromise. Some of these evidence sources are written by tools or programs that are writing the log with the full intention of that log being used as evidence. These are generally auditing functions built into operating systems, network security monitoring or endpoint software, or data loss prevention solutions.

On the other hand, there are also "unintentional" evidence sources—things that happen as a side effect of the user doing something that aren't intended to be used as evidence, but nonetheless still serve the purpose of aiding an investigation. For example, any time a new executable is opened in Windows, a file is written to the hard drive that shows the name of the file and has a timestamp showing when it was run. Although the purpose of this is to allow Windows to open the program faster the next time it is run, forensics investigators and incident responders love these files (called the "shim cache") because in the absence of other system monitoring software, it can identify when unexpected executables have been run. These files were not intended to be used for auditing or investigation purposes, but their nature lends them well to that purpose. They are more of what Bob Ross would call a "happy accident"!

Why is thinking about it this way important? Because in general intentional evidence is the more desirable source for information. It was purposely created to act as an audit trail and therefore is more likely to be trusted and accurate than something that just happened to work. It's also typically easier to gather, parse, and centralize than unintentional evidence. Windows doesn't have a mechanism to publish the previously mentioned shimcache to an event log where you can pick it up and send it to the SIEM, so to acquire unintentional evidence often involves live response or forensics.

[1] https://chrissanders.org/2018/10/the-role-of-evidence-intention/



Alternative Sources of Information

A related idea: Options for sourcing information

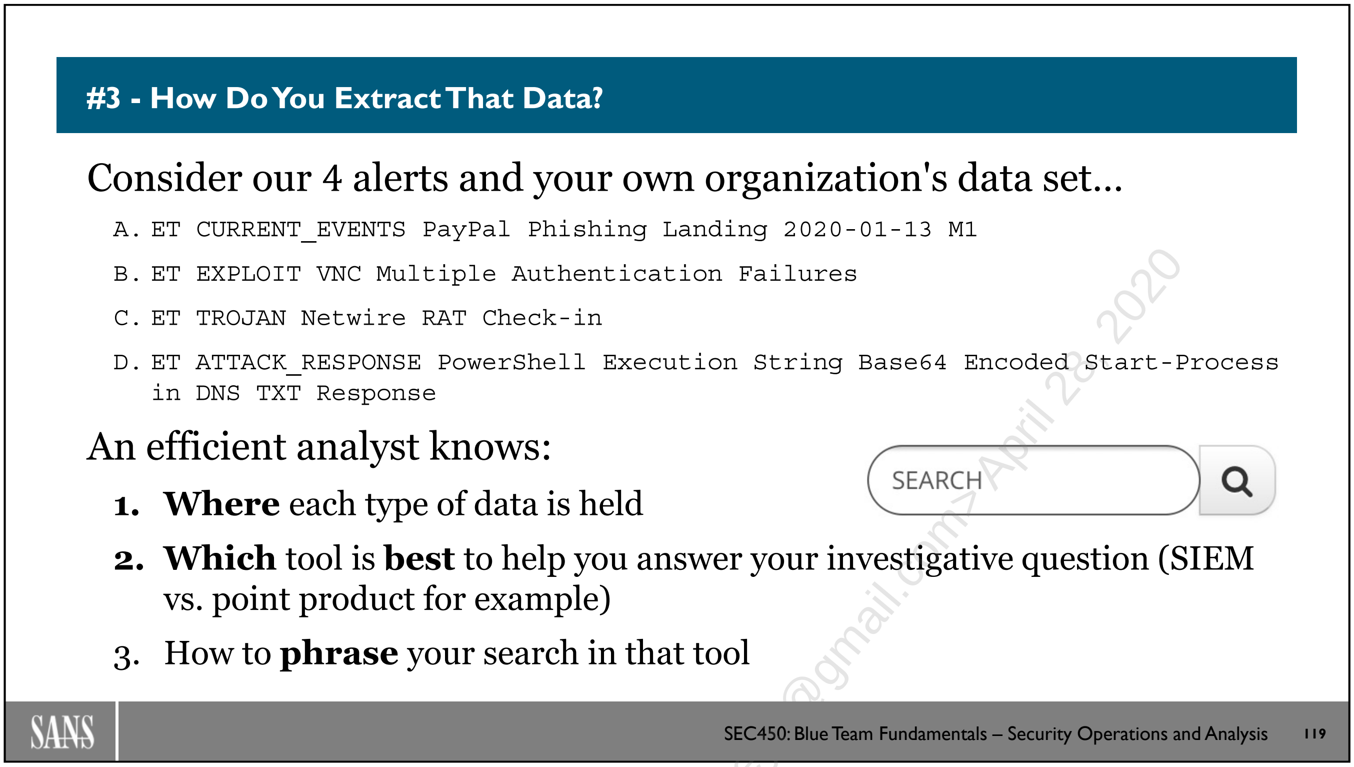
- The **main** intended auditing solution for that data
  - IP assignments = DHCP
  - Web traffic = proxy
  - Logins = security log
- **Alternative** sources of information
  - Other logs and events that just happen to contain the same info
  - Or events where the data can be *inferred*
- **Being an efficient analyst requires knowing alternative sources of data *very well***

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 118

Alternative Sources of Information

Being fast at alert investigation will require you being familiar with both intentional and unintentional data sources, but also what information is available from each source, even if it is not considered the primary source of that type of information. For example, you may normally figure out who went to a website based on the proxy log, but if someone took it away would you still be able to figure out the answer? Do you know which sources would collect both the source and destination IP of that transaction? There will plenty of times throughout your career you may be called upon to put a story together with much less than perfect information, which is why being familiar with the architecture of your network and the content of your logs is extremely important to your effectiveness.



#3 - How Do You Extract That Data?

Consider our 4 alerts and your own organization's data set...

```
A. ET CURRENT_EVENTS PayPal Phishing Landing 2020-01-13 M1
B. ET EXPLOIT VNC Multiple Authentication Failures
C. ET TROJAN Netwire RAT Check-in
D. ET ATTACK_RESPONSE PowerShell Execution String Base64 Encoded Start-Process
   in DNS TXT Response
```

An efficient analyst knows:

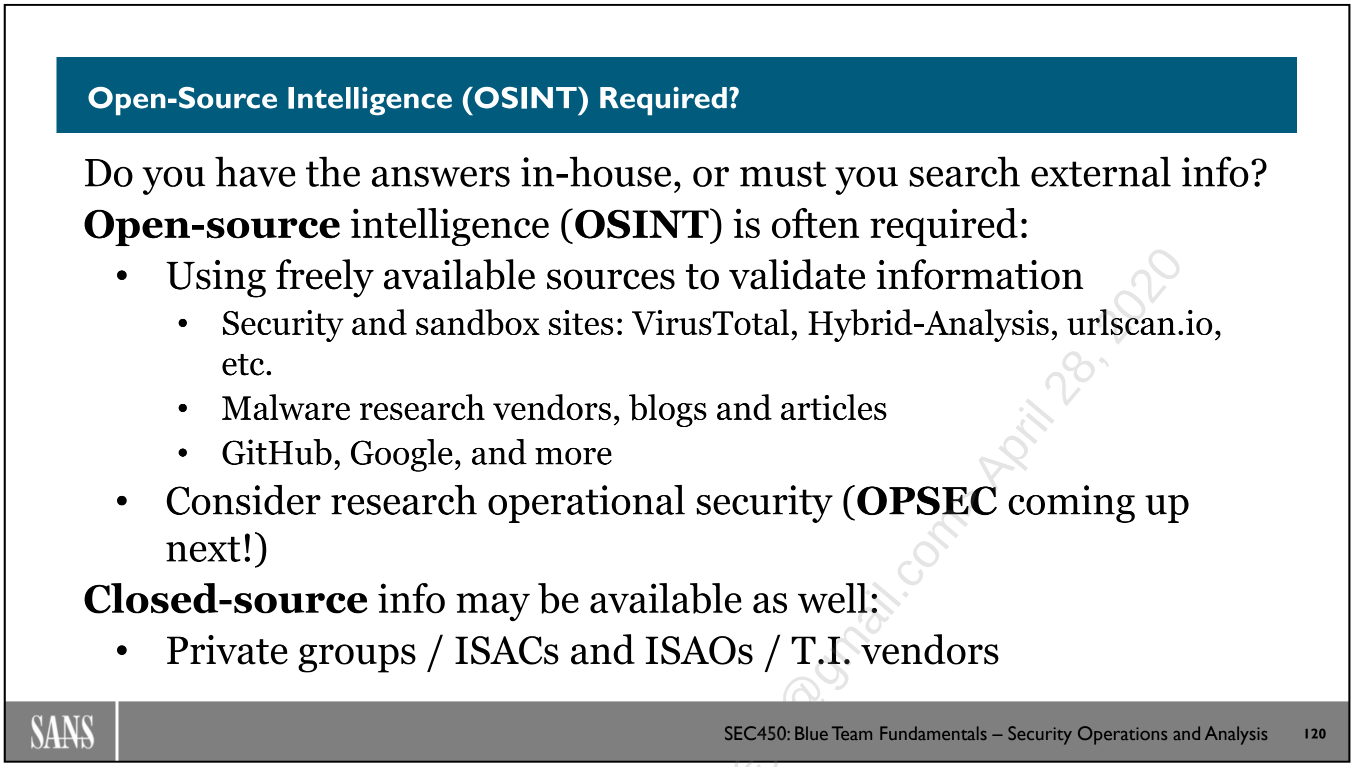
1. **Where** each type of data is held
2. **Which** tool is **best** to help you answer your investigative question (SIEM vs. point product for example)
3. How to **phrase** your search in that tool

#3 - How Do You Extract That Data?

Once the data *type* is identified (HTTP logs vs. PCAP, for example), the next step is to identify *where* that data is located and how to extract it. Keep in mind that many sources of data may be available in more than one location and that there is likely a *best* choice for where you should examine the data based on what you are looking for and the question you're trying to answer.

An example: Most intrusion detection systems will have a dedicated console that analysts could log in to for viewing details on the triggered alarms. As we discussed on day 1, it's also likely that the data from each alert is being passed to the SIEM for centralized correlation. If you are trying to investigate an alert, is it better to view the alert data in the SIEM or the IDS console? The answer depends on which additional data you will need to answer the question. If, for example, you're looking to examine packet captures of a triggered alarm, the SIEM is likely not going to be the best place to do that since most SIEMs do not directly support PCAP content browsing. If you're trying to correlate that information with other logs from the device however, the SIEM is your best bet.

A separate question is *how* you phrase that search. Each tool is likely to have its own search functionality implementing its own search language (i.e., Splunk SPL or Kibana KQL). If you had an IDS, packet capture solution, and SIEM, for example, that's three search languages you will need to be fluent in, or at least know how to use the interface to scope the search down to your intended data. This is another hurdle that new analysts must get over in order to be effective in their positions.



Open-Source Intelligence (OSINT) Required?

Do you have the answers in-house, or must you search external info?

Open-source intelligence (**OSINT**) is often required:

- Using freely available sources to validate information
  - Security and sandbox sites: VirusTotal, Hybrid-Analysis, urlscan.io, etc.
  - Malware research vendors, blogs and articles
  - GitHub, Google, and more
- Consider research operational security (**OPSEC** coming up next!)

Closed-source info may be available as well:

- Private groups / ISACs and ISAOs / T.I. vendors

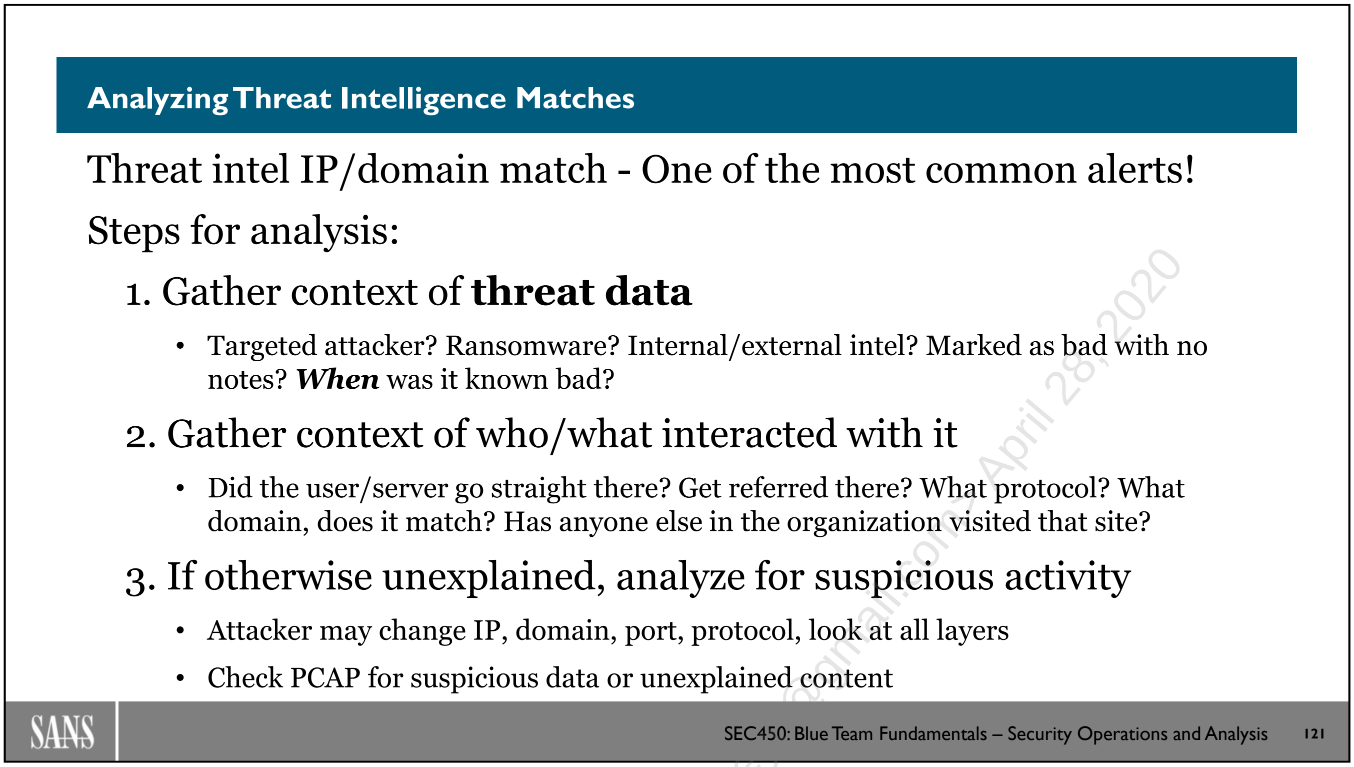
SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 120

Open-Source Intelligence (OSINT) Required?

It's great when you have all the answers you need within your SIEM and threat intelligence platform, but unfortunately, many times that will not be the case. Since there are an infinite number of potentially bad sites and files, it's likely that when you encounter a new potential IOC, you won't have any existing information on it. Whether you are trying to identify the reputation of an unknown file or new domain name, open-source intelligence, also called "OSINT" will often be required.

Open-source intelligence is taking the bits of information you've extracted from the situation and searching openly available, public sites for additional context on whether they are good or bad. This could take the form of VirusTotal searches, Googling IOCs, or reading blogs on past compromises and malware research. Any thing you can do that can connect the data item you have to a potential attack may help you understand whether the thing you are looking at is benign or a potentially malicious activity.

While open-source research is often quick and convenient, don’t forget to check any closed-source (non-public) data you have access to as well. If you are part of any threat intel sharing communities, have premium subscriptions to any sandbox websites, or have access to private reports from vendors, these sources may have the info you seek as well.



Analyzing Threat Intelligence Matches

Threat intel IP/domain match - One of the most common alerts!

Steps for analysis:

1. Gather context of **threat data**
   - Targeted attacker? Ransomware? Internal/external intel? Marked as bad with no notes? ***When*** was it known bad?
2. Gather context of who/what interacted with it
   - Did the user/server go straight there? Get referred there? What protocol? What domain, does it match? Has anyone else in the organization visited that site?
3. If otherwise unexplained, analyze for suspicious activity
   - Attacker may change IP, domain, port, protocol, look at all layers
   - Check PCAP for suspicious data or unexplained content

Analyzing Threat Intelligence Matches

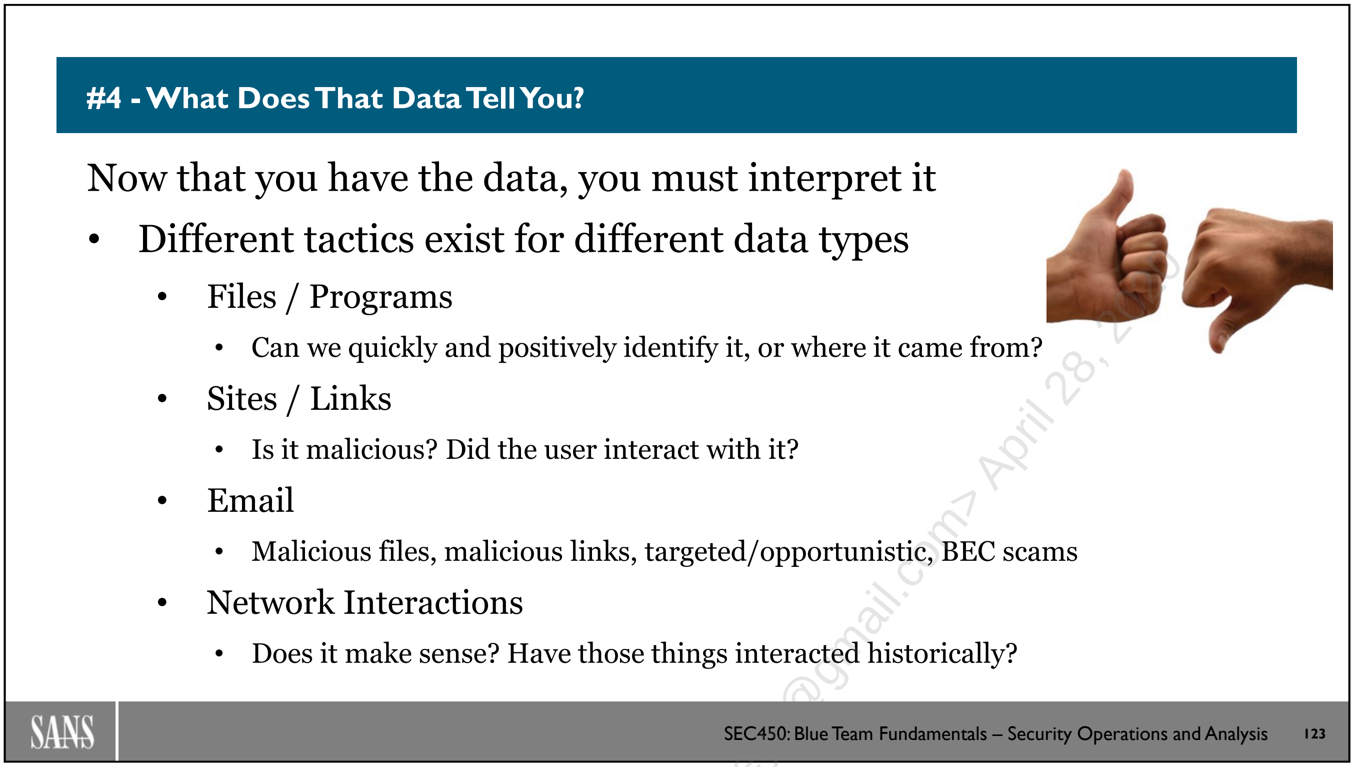
Given that your SIEM is constantly checking all the observables you've placed in your threat intelligence platform for matches with current files and network activity, threat intelligence matches are highly likely to be one of your most common alerts. Depending on how meticulous you are about putting entries into these systems, alerts based on threat intel matches may fall anywhere from low to high fidelity. Those who only trigger on matches for threat data they've seen in their own environment are likely to get a higher percent of true positives but may miss things not included in their own data. SOCs that take threat feeds from around the world and trust them to make alerts in their environment may find themselves inundated with threat intel match alerts with a higher false positive rate. Beware that everyone must decide on which set of data they will match against and where to set the bar for alerting.

Regardless of where the threat data is sourced, the process to triage these types of events is generally the same. The first thing that should be done if it isn't automatically included in the alert is to find the context in which the matched indicator was marked bad. Some indicator types are much higher fidelity than others—hashes tend to be the highest fidelity, domain names can be OK when they are purely used for attacks, and IP addresses are often the worst. Before you start to analyze the event, know how the indicator was "bad" in the past so you know what you're looking for. At this point, you may find that the indicator is an IP address of a CDN that was added to a list without any further explanation. In this case, you can already start to consider whether you are chasing a false positive.

Step 2 of analysis is to go to the actual data that triggered the alert. Look at the "story" around the situation. Was the user's PC doing nothing and suddenly reached out without explanation to a bad domain? That's highly suspicious compared to if they were browsing the web and were referred there via another site and seemed to close the window immediately without interacting with it. If there are details about the port or protocol used with a domain, see if the user was producing that type of traffic or something else. If the alert was for an IP match and the domain is on a shared hosting environment, it's entirely possible that the user just had the random luck to stumble into another site hosted from the same IP address or hidden behind the same CDN IP.

Step 3 is to dive in and check activity on the user's PC and do a full investigation of what happened. Consider all the possible options, both malicious and benign, that could have caused the match. Just because a user went to a malicious website, that does not mean they are necessarily infected. If a user went to a domain that served fake antivirus software but did not download it and closed the page, it is likely there is nothing to clean up, but the alert was "correct" nonetheless in that there was a threat.

It is hard to give a specific set of steps that applies in all cases for alerts like this, but the main takeaways are that you need to understand *why* the observable was marked bad in the first place and then look for a match to that exact thing. If it *doesn't* match, that doesn't mean the attackers aren't doing something new from old infrastructure but determining what exactly is going on will require putting the activity at the time of the alert back together.



#4 - What Does That Data Tell You?

Now that you have the data, you must interpret it

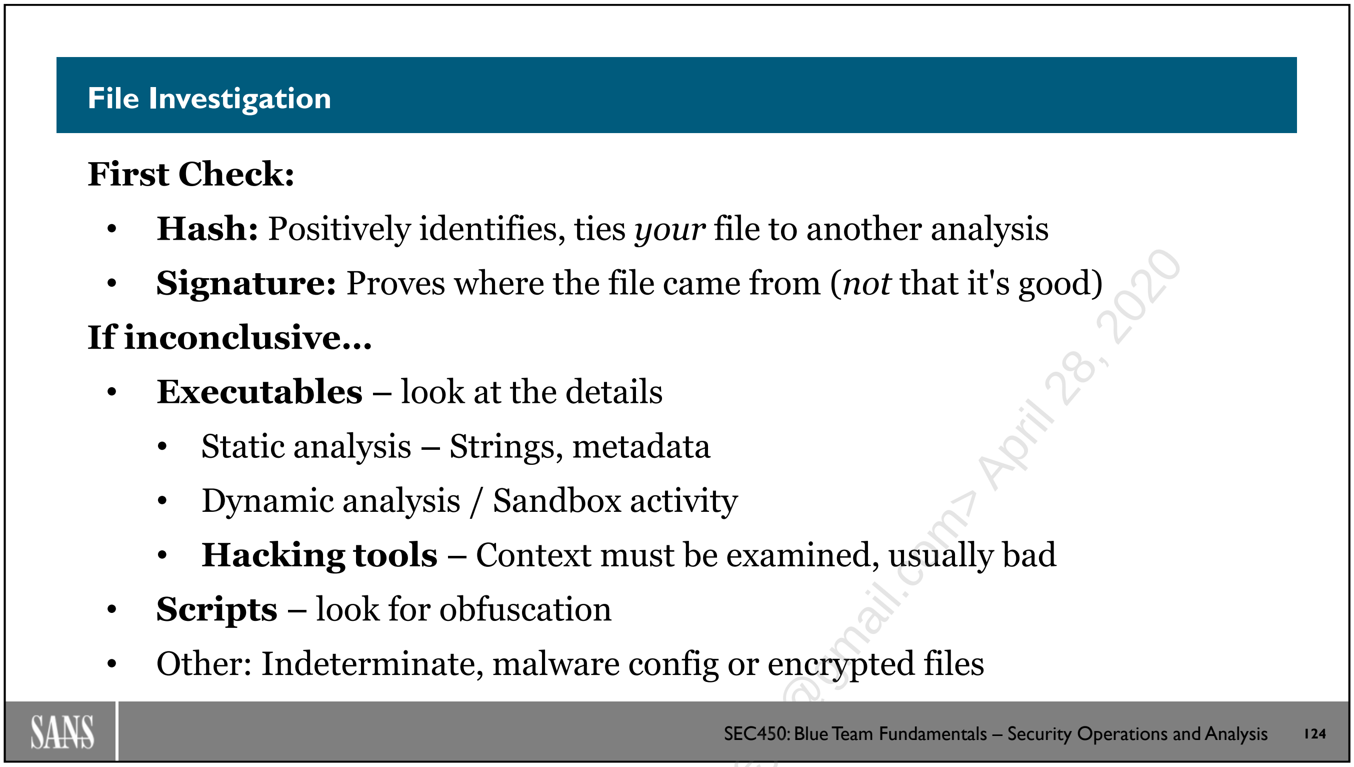
- Different tactics exist for different data types
  - Files / Programs
    - Can we quickly and positively identify it, or where it came from?
  - Sites / Links
    - Is it malicious? Did the user interact with it?
  - Email
    - Malicious files, malicious links, targeted/opportunistic, BEC scams
  - Network Interactions
    - Does it make sense? Have those things interacted historically?

SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 123

#4 - What Does That Data Tell You?

Finally, the ultimate question: “What does that data tell you?” In other words, interpreting the evidence you have located. Once you have the network traffic, files, or host activity logs in hand, it's time to decide what actually happened. This is where you should consider all the hypothesis that might be consistent with your data and ensure you aren't just chasing your favorite theory. Look at the information, ideally with a "fresh perspective", in totality, and try to see which theories are most consistent with the evidence, looking to disprove theories where possible.

The path you take to answer your questions will depend on the type of evidence you have collected. Over the next few slides, we'll walk through some specific common tactics that can be used to analyze files and programs, website traffic, and email to determine whether it is good or bad.



File Investigation

First Check:

- **Hash:** Positively identifies, ties *your* file to another analysis
- **Signature:** Proves where the file came from (*not* that it's good)

If inconclusive…

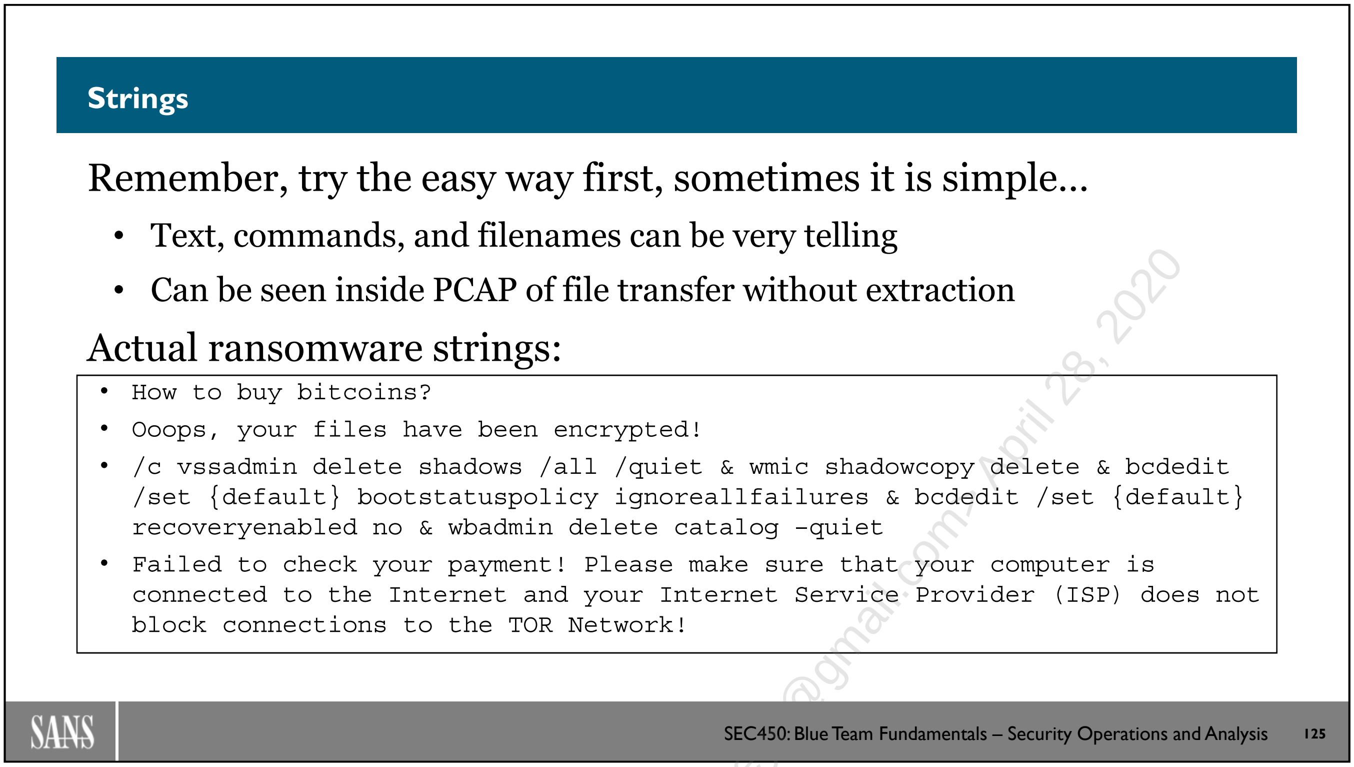
- **Executables** – look at the details
  - Static analysis – Strings, metadata
  - Dynamic analysis / Sandbox activity
  - **Hacking tools** – Context must be examined, usually bad
- **Scripts** – look for obfuscation
- Other: Indeterminate, malware config or encrypted files

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 124

File Investigation

We previously covered several ways of identifying files with signatures, hashes, and script content. There are other scenarios beyond this that are harder to identify and require more analysis than just purely a hash or signature, though. This includes hacking tools or tools used that *can* be used for good and therefore may not set off alarms but nonetheless allow attackers to make progress within the environment. While your first moves for identifying files should be the tactics we talk about in the previous books (hashes, signatures, file type, content etc.), there are some questionable areas for "gray area" executables, as well as artifacts of compromise where these checks might not be so clear. Let's discuss some of these additional situations you may find yourself in while performing file analysis and how to address them.



Strings

Remember, try the easy way first, sometimes it is simple…

- Text, commands, and filenames can be very telling
- Can be seen inside PCAP of file transfer without extraction

Actual ransomware strings:

```
• How to buy bitcoins?
• Ooops, your files have been encrypted!
• /c vssadmin delete shadows /all /quiet & wmic shadowcopy delete & bcdedit
  /set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {default}
  recoveryenabled no & wbadmin delete catalog -quiet
• Failed to check your payment! Please make sure that your computer is
  connected to the Internet and your Internet Service Provider (ISP) does not
  block connections to the TOR Network!
```

Strings

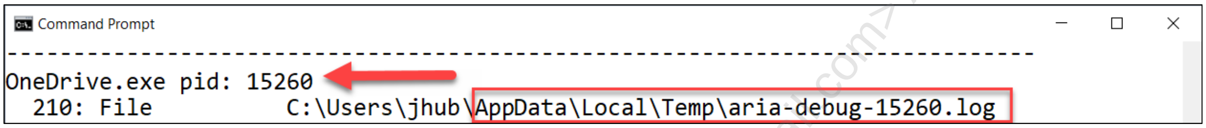
Remember to try the easy and sure things first as a method for fast investigation. Strings can be a shockingly easy way of identifying malware, although many programs will go lengths to obfuscate strings or otherwise hide them through packing, many samples do not. The good thing about non-hidden strings is that you don't even have to extract a file from a (non-encrypted) PCAP to see them. Simply using Wireshark or any other tool to "follow the session" and view the bytes of a malicious file transferred over the wire can sometimes expose messages such as the ones shown above, which give you a very clear indication that a file is malicious. These strings were taken from an actual sample on hybrid-analysis.com labeled as WannaCry.[1] Remember, malicious file identification is not always complicated, but understanding how to identify viruses in the most simplistic way possible is part of the battle.

[1] https://www.hybrid-analysis.com/sample/bf293bda73c5b4c1ec66561ad20d7e2bc6692d051282d35ce8b7b7020c753467?environmentId=100

Malware Configs and Threat Artifacts

Sometimes it's less clear, you may have a *piece* of data from malware

- Some malware uses external configuration files
- These may be encrypted or otherwise meaningless looking
- How do you associate them with the parent malware?
- One option: **Handle** from Sysinternals



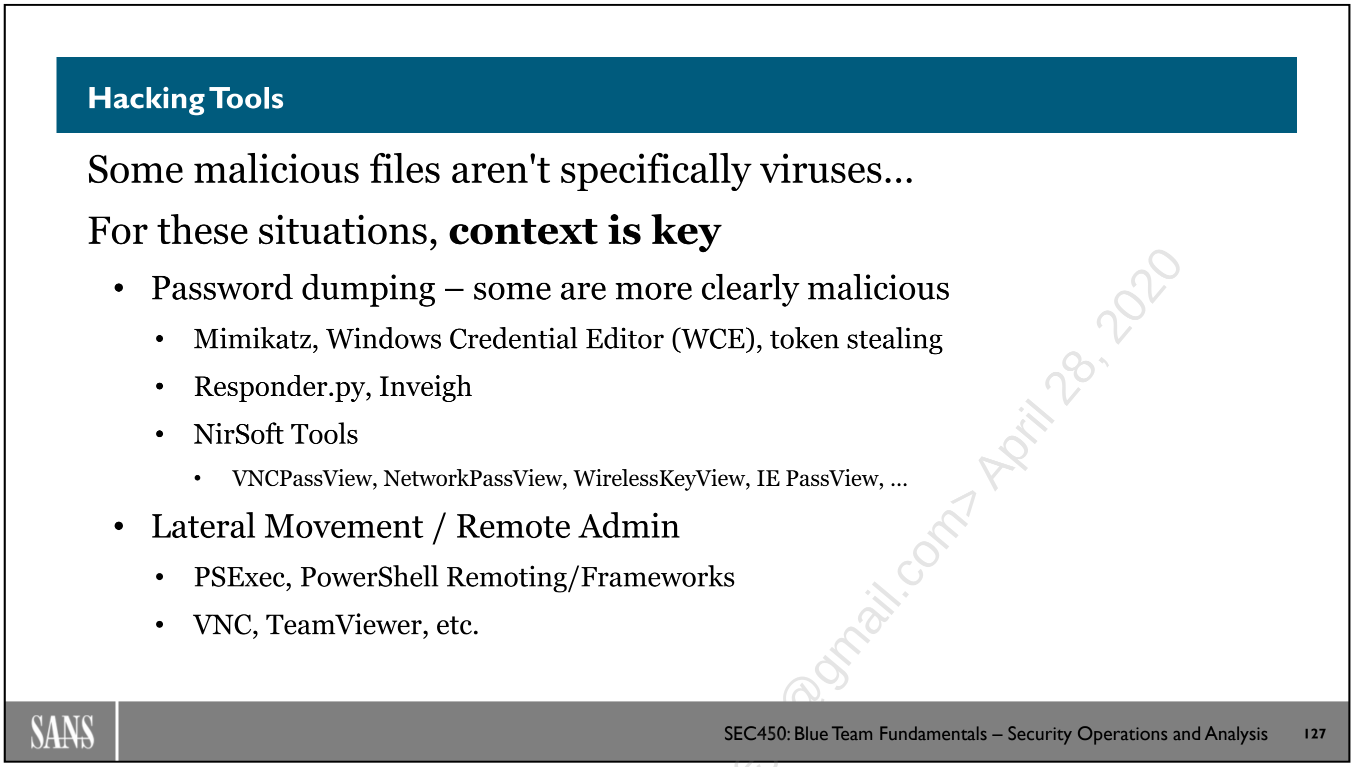
```
-----------------------------------------------------------------------------
OneDrive.exe pid: 15260
  210: File          C:\Users\jhub\AppData\Local\Temp\aria-debug-15260.log
```

Malware Configs and Threat Artifacts

Sometimes, you'll investigate a file and find it to be some simple text or a totally incomprehensible mess of encrypted content, leaving you with no idea of its purpose. This is another category of what are called "threat artifacts" by some antivirus vendors, and it can be hard to the situation back together. These files may be configuration items for malware executables, or logs from a benign program, but without some way to link them to the program that wrote them, it's hard to tell. While malware config files are not malicious on their own, they *are* artifacts left over from the running of malware, and an indicator that something *is* going on—you just haven't found the main file that is the source of the problem yet.

Many types of malware such as PlugX use this multi-file config technique. This type of malware is split up so that it can be modular and customizable for each victim. Instead of embedded command and control configuration compiled into the program itself, the virus is designed to look for an encrypted file in the same folder that has the setup parameters for how the virus should act, where it should communicate, and the protocols it should use. This allows the attackers to customize it for each deployment without having to reconfigure the executable itself and allows them to change the setup on the fly on the remote system if need be. Although it is difficult to spot one of these files, if you suspect you have found one or your antivirus suite pointed one out, there are some easy to use tools that will let you trace it back to the process that is accessing it. Assuming the malware is still active, these programs will lead you back to the process name and ID of the malware allowing you to remediate it. The "Handle" tool from Sysinternals is one of those programs. Running it on the potentially infected machine produces a list of all processes and the handles they have open (every file they are accessing). If you can identify a threat artifact, any process that has a handle open to that file is now suspect.[1] The slide shows an example of finding what process is writing an example log found in the appdata\local\temp folder called aria-debug-15269.log. According to Handle, the process that is writing it is OneDrive.exe.

[1] https://docs.microsoft.com/en-us/sysinternals/downloads/handle



Hacking Tools

Some malicious files aren't specifically viruses...

For these situations, **context is key**

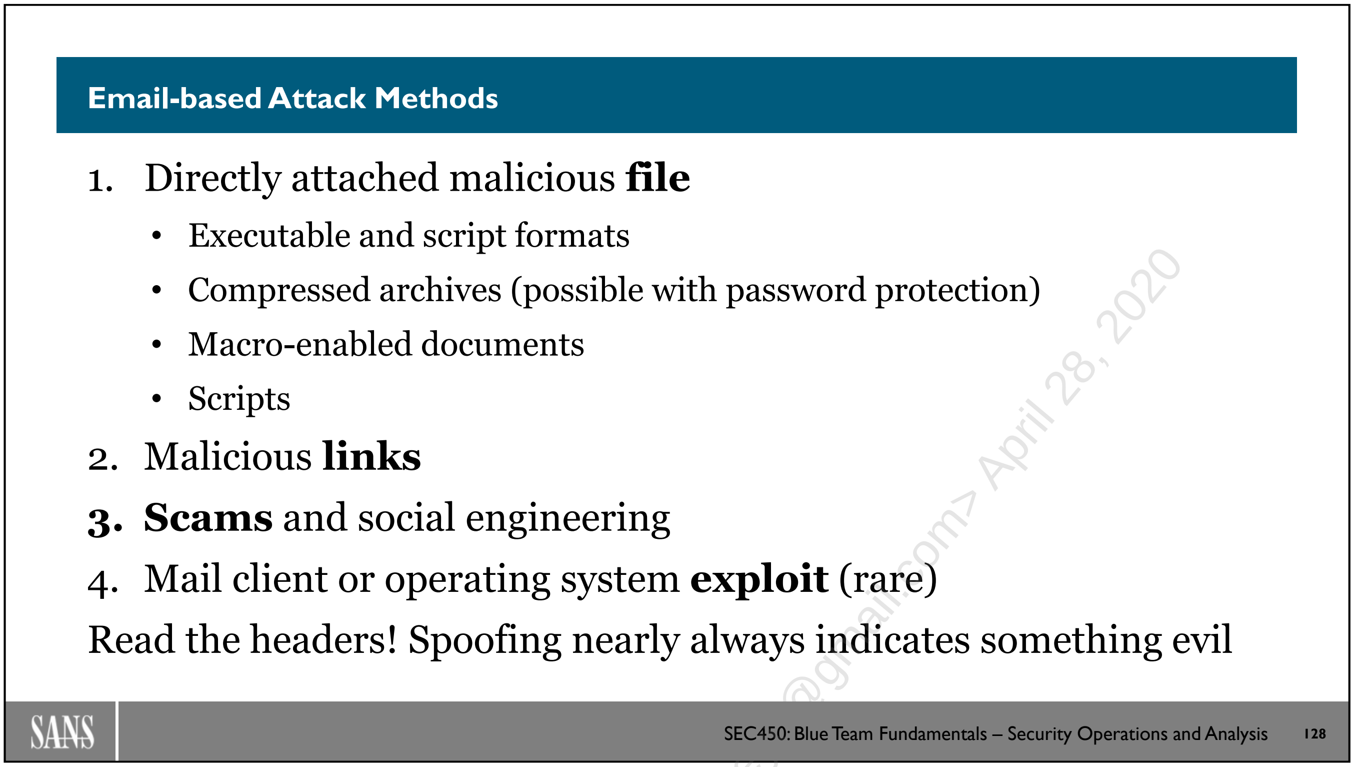
- Password dumping – some are more clearly malicious
  - Mimikatz, Windows Credential Editor (WCE), token stealing
  - Responder.py, Inveigh
  - NirSoft Tools
    - VNCPassView, NetworkPassView, WirelessKeyView, IE PassView, ...
- Lateral Movement / Remote Admin
  - PSExec, PowerShell Remoting/Frameworks
  - VNC, TeamViewer, etc.

Hacking Tools

When looking for malicious files and viruses, one problem is we sometimes run into the category of programs that some antivirus suites call "hack tools." These are programs that are not necessarily trojans or backdoors themselves, but are utilities commonly run by attackers as part of an active intrusion campaign for credential dumping or privilege escalation. While some of them such as Mimikatz, Windows Credential Editor, or any of the other attack-centric tools such as responder.py or Inveigh (used for privilege escalation on the local network) *should* set off antivirus alarms without questions, other programs that are more normal consumer-oriented may not.

Tools such as the utilities from the NirSoft website, which can be used "legitimately" for users to recover passwords, may also be used by attackers to gain credentials while staying under the radar.[1] Sometimes, it is only the context of their use that separates a user downloading one of these utilities vs. a targeted attacker doing it from their computer and pretending to be them. This situation can be even more difficult and volatile with some of the lateral movement/remote administration tools such as PSExec, PowerShell Remoting based frameworks, and utilities such as VNC and TeamViewer. Many of these tools are used by administrators for remote administration work and knowing a good from malicious use may come down to having a policy that strictly defines what is allowed and looking for deviations, as well as being able to effectively spot those deviations on all hosts. In many cases, if you see one of these tools downloaded, the best bet is either to talk with the user directly or check their browsing history immediately before the event to see if it seems they are trying to solve a specific problem and have settled on that tool as a solution. Seeing "how do I recover my Wi-Fi password" webpages in their proxy log history immediately before the download may be a clue this is a policy violating insider as opposed to an attacker.

[1] http://www.nirsoft.net/password_recovery_tools.html



Email-based Attack Methods

1. Directly attached malicious **file**
   - Executable and script formats
   - Compressed archives (possible with password protection)
   - Macro-enabled documents
   - Scripts
2. Malicious **links**
3. **Scams** and social engineering
4. Mail client or operating system **exploit** (rare)

Read the headers! Spoofing nearly always indicates something evil

Email-based Attack Methods

There are numerous ways email can be abused. The most common way is directly attaching evil files. Though many organizations have locked down the obvious method of directly sending in executable files, there are other, sneakier ways such as macro-enabled MS Office documents and scripts that can be executed by Windows. These are harder to eliminate with a file type check because people still need to send .doc and .xls files to each other for legitimate purposes, so filtering of these attachments takes more complex and expensive filtering technologies. Scripting files tend to have file extensions that can be blocked in a blanket manner by policy, but since these are slightly more "off the radar" than executable files, they might be more likely to make it through some spam filters.

Another method is sending compressed archives with malicious files inside them. This method is a bit cleverer as the files cannot directly be examined by email filters unless they are decompressed first, which puts additional overhead on the appliance to perform analysis. In addition, sometimes the compressed files are password protected. When password protection is applied, attackers must still include the password in the email body, but since it is hard to extract and use a password in an automated way, email filtering appliances tend to fail to decompress the files, leading to them potentially making it through the filter to the victim, who can successfully read the password and extract them.

Malicious links are another eternal favorite of attackers. They work well because it's harder to identify a malicious link in an automated way compared to a file, and since it's just text, there is no blanket policy that can easily eliminate all emails that contain them. Scams and social engineering are a third way of facilitating evil through email. These are sometimes the hardest to detect because there is no malicious content or links at all to be found, purely text from a scammer trying to impersonate someone else and perhaps insert themselves into a banking transaction or convince an employee to take an action they otherwise shouldn't.

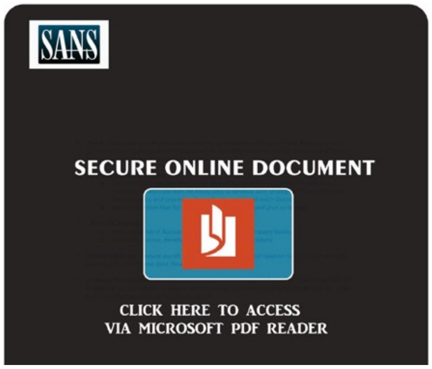
The final category is that of mail client and operating system exploits using file types delivered through email. These types of attacks are extremely rare in comparison but have been viable in the past. These types of attacks can be very dangerous when available because they, in effect, allow attackers to weaponize files that would normally be considered safe for email and use them to compromise a computer. One example of this was the Windows TIFF exploit that was released back in 2013.[1] This flaw in Windows rendering of the TIFF image format allowed attackers to gain control of a machine merely by having a user open an image file – something that is easy to do via email since people don't consider pictures to be malicious.

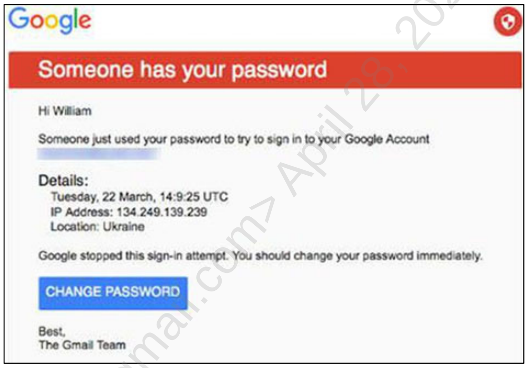
[1] https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/2013/ms13-096

Targeted vs. Opportunistic Attacks

One question you should **first** be asking yourself:

"Does this seem like a *targeted* attack?"




Google

Someone has your password

Hi William

Someone just used your password to try to sign in to your Google Account

Details:
Tuesday, 22 March, 14:9:25 UTC
IP Address: 134.249.139.239
Location: Ukraine

Google stopped this sign-in attempt. You should change your password immediately.

CHANGE PASSWORD

Best,
The Gmail Team

Targeted vs. Opportunistic Attacks

One of the questions you should be constantly asking yourself when you first see evidence of malicious activity is, "Does this seem like an attack specifically targeted at me or my organization?" Factors that can help you determine whether an attack is targeted or not are anything that shows the target has taken extra time to customize an attack on your organization or the people who work for it. This can take the form of using a lookalike domain name, a logo added to an email, spear-phishing content addressed only to a single individual employee by name, or even just the fact that the indicators associated with the attack seem to be unseen by anyone else thus far. On the left side of the slide, we have an actual spear-phishing email from a financially motivated attacker that included a company logo in the upper left corner (edited to the SANS logo). Although the document was customized in a very lame way, the fact remains that they are taking the time to craft specific emails per organization, meaning whatever they are after must be worth the extra effort.

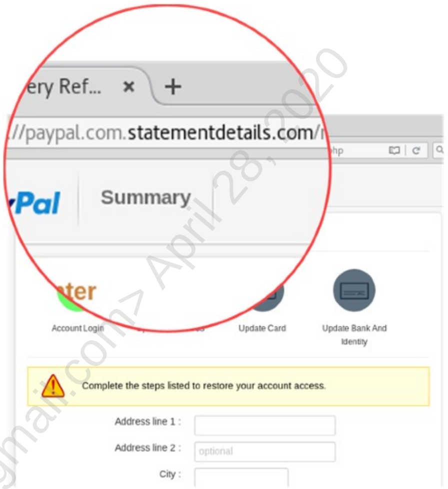
Other times, targeted attacks won't be so obvious on the surface. The picture on the right is a screenshot of the email that led to the hack of the Democratic National Committee in the US 2016 election. The change password button led to a bit.ly link which ultimately led to a fake password stealing page. If you were assigned this alert, how could you tell that this was a targeted attack? Later, we'll show a way to look at the forwarded URL and statistics associated with some shortened links by modifying the URL. Upon investigation of the link, the domain name it was forwarding to could have been researched and found to be unknown in open source intel, meaning it may have been created for them. Another option is the statistics of how many people had already clicked it. If the email were to have gone to a large group of potential victims, the statistics page for the link would show lots of activity.

Answering this question should be one of the main factors that drives the type of response actions taken. If an attack is clearly generic and contains only indicators that VirusTotal or other online services clearly show other individuals have already seen, the incident response can take a more casual approach. If, however, the attack is targeted, the SOC must carefully consider the angle that should be taken in the response—you don't want to tip the attackers off that you have noticed their attack. If you do, they may shift tactics and come back with another attack you might not notice. This is where the question of "immediate response vs. watch and wait" comes in, and careful operational security steps should be taken to not jeopardize the response. We will cover considerations for targeted attack response shortly.

Assessing Links

Most common tactics for phishing via links:

- Text differs from link target
- Lookalike domains (including IDNs)
- Lookalike subdomains
- Cloned login pages
- Open redirects
- "Good" site, bad content
- Hacked sites
- Shortened links

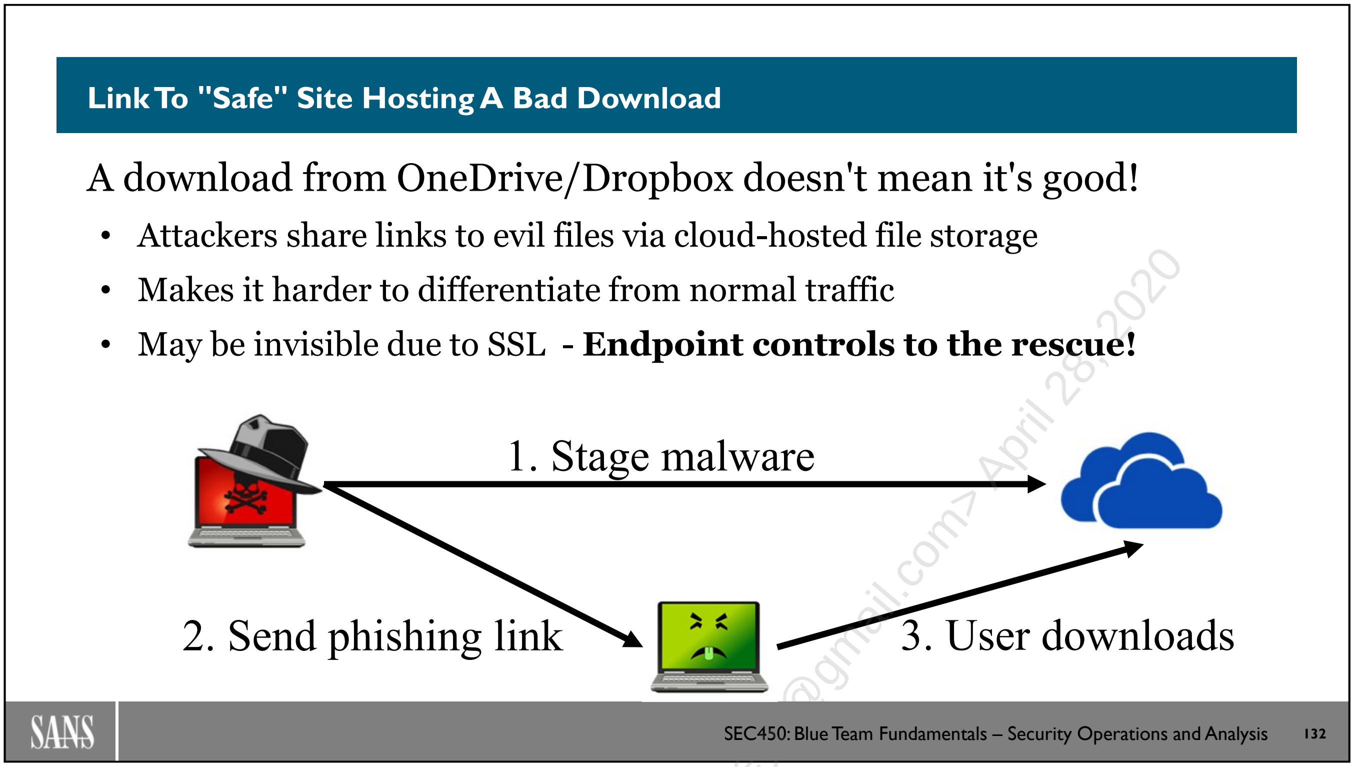


Complete the steps listed to restore your account access.

Assessing Links

There are lots of ways to trick users using phishing emails based on malicious links. Here are some of the most common tactics to look out for:

- Text differs from link target: One of the oldest and most common attacks, it unfortunately still works because people don't always hover over a link like they should before clicking on it.
- Lookalike domains: Even when the text and link match, that doesn't mean people are going to the site they think they are. IDNs or simple character switches such as "nn" instead of "m" can fool people.
- Lookalike subdomains: Domains that are "correct" but are not the parent-level domain. These prey on people who do not understand how domain names work.
- Cloned login pages: When the goal of the adversary is to collect names and passwords for the target environment, they often will use tools to clone webpages and link users there in hopes of getting credentials. Tools like the Social Engineering Toolkit make website cloning and hosting an extremely simple wizard driven affair.
- Open redirects: Acting sort of like link shorteners, every once in a while, attackers figure out how to abuse the features of a site to make it redirect visitors to a specific URL to an arbitrary secondary site.
- Good site, bad content: Another way of disguising a bad link is hosting something evil on a "known good" site. Attackers can upload malicious files to OneDrive and Dropbox and link users there through phishing. In logs, it won't trigger any alerts since people use these services all the time.
- Hacked sites: Domain shadowing or just plain hacked website servers can serve as a great way to hide malicious activity behind a "good" domain.
- Shortened links: Phishers often use shortened links to obscure where their link goes.



Link To "Safe" Site Hosting A Bad Download

A download from OneDrive/Dropbox doesn't mean it's good!

- Attackers share links to evil files via cloud-hosted file storage
- Makes it harder to differentiate from normal traffic
- May be invisible due to SSL - **Endpoint controls to the rescue!**

1. Stage malware

2. Send phishing link

3. User downloads

Link To "Safe" Site Hosting A Bad Download

Phishers and attackers love to host bad content on good sites. They do this by uploading a macro-enabled document, script, or even malicious executable to a folder in an account they own and share the folder to the public so that anyone with the link can get a copy. Then, they simply send out the typical phishing messages with a link to their malicious content hosted on a "trusted" site.

Attackers like to operate this way for multiple reasons. One is that they know you are much less likely to be suspicious of traffic going to Google Drive, Dropbox, OneDrive, or any other cloud-based file sharing platform. Another reason is that these sites tend to be SSL encrypted and easy to use, so an attacker's overhead for setup is very small, and unless your organization uses SSL decryption, it may be extra hard to identify. Using these services means virtual host field checks will pass and you will be left to detect their malicious file through some other type of methods such as endpoint controls (EDR, AV, process monitoring, etc). For defenders, this means we must either block these types of downloads by policy or be able to tell good downloads from bad downloads once they hit the endpoint. Appliances that sandbox file downloads initiated from clicks in email are one way to prevent this type of activity.

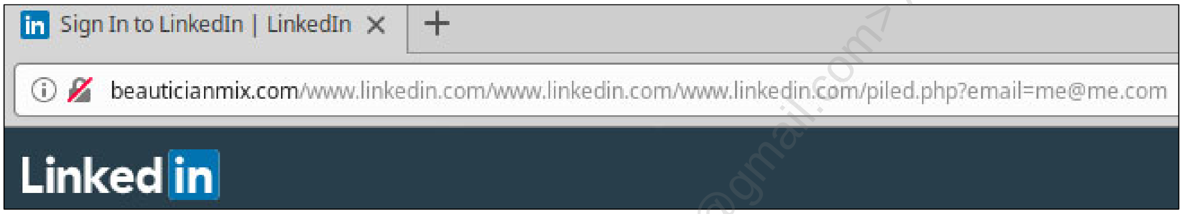
Link to a Hacked Site

If a domain *seems* ok, it *still* may have been hacked…

- Attackers love taking over vulnerable sites and hosting malicious content

Signs a normally good site might have been hacked:

- Odd **subdomain** of "good" parent domain (shadowing)
- Suspicious **folder name** under an otherwise normal domain
- Blog software – WordPress, Joomla, Drupal, etc.



Sign In to LinkedIn | LinkedIn

beauticianmix.com/www.linkedin.com/www.linkedin.com/www.linkedin.com/piled.php?email=me@me.com

Link to a Hacked Site

Sometimes, attackers like to piggyback off the good name a site has already made for itself and use it to deliver malware. When new exploits get released for popular content management systems such as WordPress or Drupal, you can bet that attackers will almost immediately start scanning the internet for servers they can compromise and flip to using for their own means. This means you may see an alert for an exploit attempt or malware download from a site that has a "good" reputation on many websites. Alternatively, if an adversary can phish a domain’s administrator and get access to their registrar, they may use an additional A record subdomain in their DNS zone pointed at their own IP to "host" malicious files under the good domain's name.

It's often easy to tell if this is the case because sites that have been evil from the start will not have good ratings with reputation engines or scanners like VirusTotal or URLScan. Hints such as running recently compromised software, an active malicious redirect, or a long-ago creation date for the domain points toward the theory of a hacked rather than an infected site. You may even be able to use the update times of the files in the HTTP server response to see if the one with the redirect was recently placed. One situation that may arise is finding a user went to one of these sites while it was infected, but before you had a chance to investigate it, the site got cleaned up. If you don't have a PCAP capture or other logs to put the interaction back together, sites like URLQuery.net or URLscan.io may have historical snapshots of the site in its infected state. You can use that information to see what the infection would have done or where it would've redirected the user if successful to see if your user seems to have been affected.

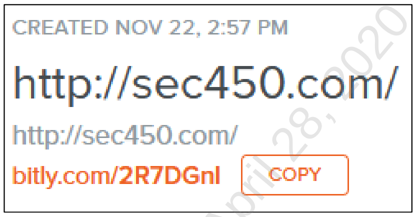
The thing you must consider when dealing with a hacked site is if you eventually want to unblock it. With a site created for evil, it's not a problem to permanently block a domain. For a hacked site, if it is a relatively unknown site, it may be OK to put on your block list forever, but occasionally bigger name sites become infected and need to be blocked for the time being. For situations like this, it may be worth considering how you can set a reminder to revisit it or set an expiration date on the block if possible.

What's That Shortened Link Hiding?

Shortener built-in methods:

- **Bit.ly**: Add "+" to end of URL
- **goo.gl**: Add "+" to end of URL
- **Tinyurl**: Add word "preview" as a subdomain
  - `http://**preview**.tinyurl.com/...`
- **Is.gd**: Add "-" to end of URL
- **Tiny.cc**: Add "~" to end

`https://bitly.com/2R7DGnl+`



CREATED NOV 22, 2:57 PM

http://sec450.com/

http://sec450.com/

bitly.com/2R7DGnl

Link expanding tools:

- Getlinkinfo.com
- Expandurl.net
- toolsvoid.com/unshorten-url

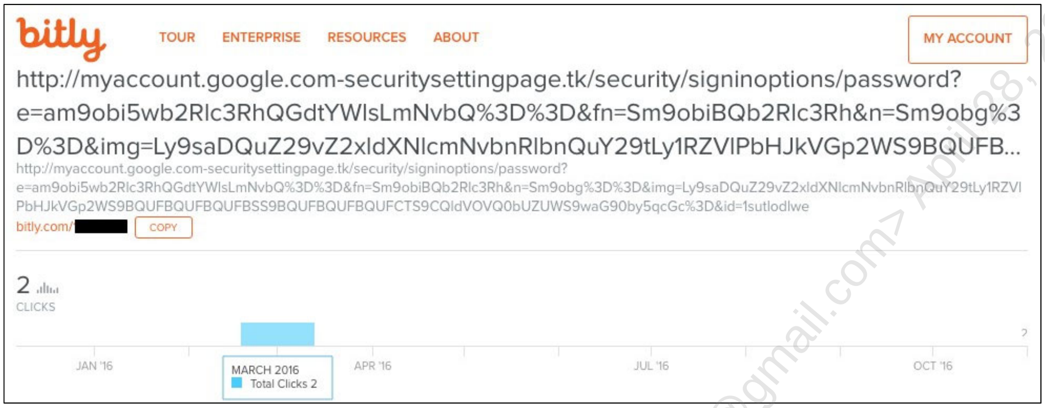
What's That Shortened Link Hiding?

Often, attackers will use shortened links for phishing and other malicious URLs to obfuscate the true link that will be visited. To counter this, analysts must be able to uncover the true URL in a safe way. One of the ways to do this is the built-in functionality that some link shorteners have. Bit.ly and goo.gl links, for example, will show where a link leads if a plus sign is added to the end of the URL. Most link shorteners offer this capability, but if you'd like to do all your link expanding in one place, there are third-party services that will expose a URL for you (listed on the right). Some of them, such as expandurl.net, will even show a screenshot of the destination as well.

Tracking Clicks to Shortened Links

Did anyone get phished with that shortened link?

Some services offer statistics pages that can be helpful!



http://myaccount.google.com-securitysettingpage.tk/security/signinoptions/password?e=am9obi5wb2Rlc3RhQGdtYWlsLmNvbQ%3D%3D&fn=Sm9obiBQb2Rlc3Rh&n=Sm9obg%3D%3D&img=Ly9saDQuZ29vZ2xldXNlcmNvbnRlbnQuY29tLy1RZVlPbHJkVGp2WS9BQUFB...

http://myaccount.google.com-securitysettingpage.tk/security/signinoptions/password?e=am9obi5wb2Rlc3RhQGdtYWlsLmNvbQ%3D%3D&fn=Sm9obiBQb2Rlc3Rh&n=Sm9obg%3D%3D&img=Ly9saDQuZ29vZ2xldXNlcmNvbnRlbnQuY29tLy1RZVlPbHJkVGp2WS9BQUFBQUFBQUFBSS9BQUFBQUFBQUFCTS9CQldVOVQ0bUZUWS9waG90by5qcGc%3D&id=1sutlodlwe

bitly.com/[redacted] COPY

2 CLICKS

Tracking Clicks to Shortened Links

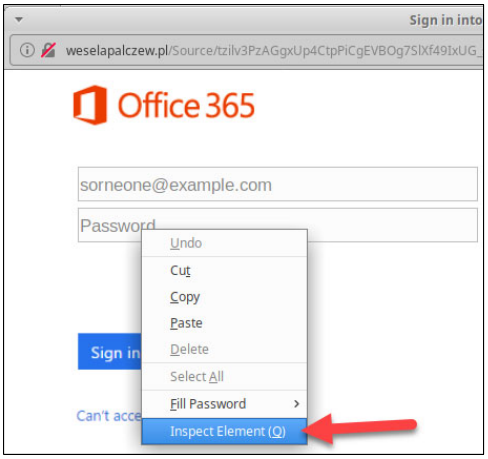
Some day, you will likely be hit with a targeted phishing wave that uses custom shortened URLs and you will need to answer the question "has anyone clicked this?" One of the ways you can assess this is the built-in statistics for some link shorteners.

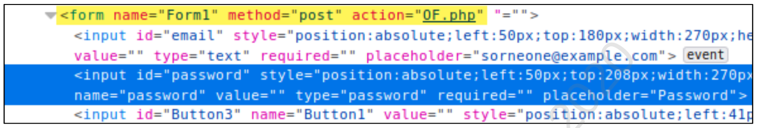
The picture on the screen is the bit.ly statistics for the shortened link used by the "Fancy Bear" APT group to phish John Podesta from the Democratic National Committee during the 2016 U.S. elections. Since the stats for the link are available, anyone with the link could see that in March 2016 the link was indeed clicked twice, likely leading to the compromise (also check out the clever lookalike attack domain – myaccount.google.comsecuritysettingspage.tk).[1] Note that although this does confirm the link was clicked, it doesn't mean it was clicked by the victim. The attackers themselves could have been testing it, but it is another data point that may be used for adding context to an investigation. This technique will also obviously only work if the link is used against a single person or organization. Checking the statistics on a link that went out in spam worldwide will not give any meaningful answers.

[1] https://motherboard.vice.com/en_us/article/mg7xjb/how-hackers-broke-into-john-podesta-and-colinpowells-gmail-accounts

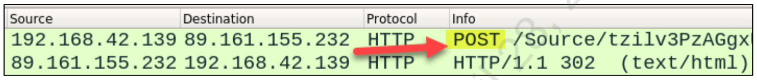
Where Does That Password Box Go?

Did a user enter their password? How can we tell?

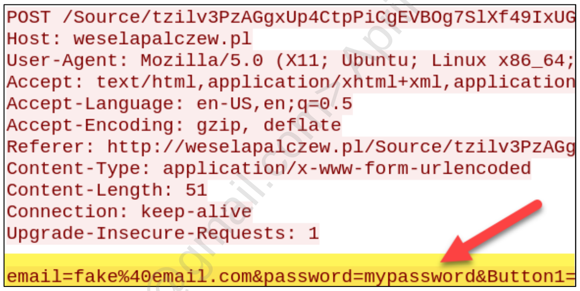




```
<form name="Form1" method="post" action="OF.php" "="">
  <input id="email" style="position:absolute;left:50px;top:180px;width:270px;he
  value="" type="text" required="" placeholder="sorneone@example.com"> event
  <input id="password" style="position:absolute;left:50px;top:208px;width:270px
  name="password" value="" type="password" required="" placeholder="Password">
  <input id="Button3" name="Button1" value="" style="position:absolute;left:41p
```



```
Source        Destination   Protocol Info
192.168.42.139 89.161.155.232 HTTP    POST /Source/tzilv3PzAGgx
89.161.155.232 192.168.42.139 HTTP    HTTP/1.1 302  (text/html)
```



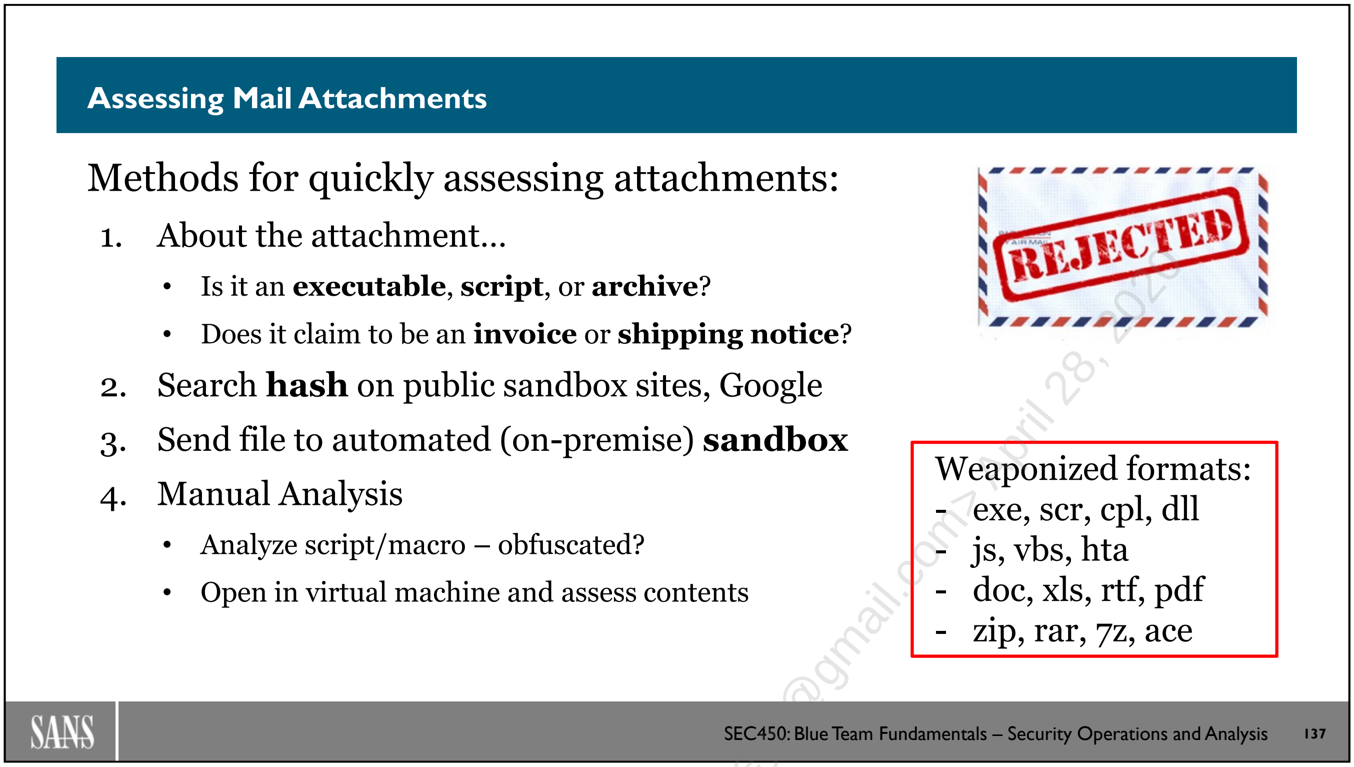
```
POST /Source/tzilv3PzAGgxUp4CtpPiCgEVBOg7SlXf49IxUG
Host: weselapalczew.pl
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64;
Accept: text/html,application/xhtml+xml,application
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://weselapalczew.pl/Source/tzilv3PzAGg
Content-Type: application/x-www-form-urlencoded
Content-Length: 51
Connection: keep-alive
Upgrade-Insecure-Requests: 1

email=fake%40email.com&password=mypassword&Button1=
```

Where Does That Password Box Go?

How can we tell if anyone at our organization has fallen for a phishing scam? If the traffic is not encrypted, we can investigate the site and find where the form posts the user's password if it is entered. In the image on the left side of this slide, the Firefox browser's "Inspect Element" functionality is used to pull up the HTML that runs the web form. In the top picture on the left, we see that if the password is entered, a POST method HTTP request will be sent to the file "OF.php".

Looking at traffic for the interaction in Wireshark shows that there is a POST request to the site. When the TCP session is followed, the contents of the filled-out form are visible, and this would help you identify the victim of the phishing (bottom right). For this example, the credentials of login "fake@email.com" and password "mypassword" were entered and submitted to the site.



Assessing Mail Attachments

Methods for quickly assessing attachments:

1. About the attachment…
   - Is it an **executable**, **script**, or **archive**?
   - Does it claim to be an **invoice** or **shipping notice**?
2. Search **hash** on public sandbox sites, Google
3. Send file to automated (on-premise) **sandbox**
4. Manual Analysis
   - Analyze script/macro – obfuscated?
   - Open in virtual machine and assess contents

Weaponized formats:
- exe, scr, cpl, dll
- js, vbs, hta
- doc, xls, rtf, pdf
- zip, rar, 7z, ace

Assessing Mail Attachments

The goal in assessing malicious attachments or any potentially malicious file is to make the determination as quickly and as confidently as possible. To that end, investigating these alerts in an efficient way then means using the quickest checks first and ideally also those with the highest fidelity. Although there are no absolute rules for doing this, the strategy employed by most organizations could roughly be summed up with the steps above.

First and foremost, if the file is an executable, archive, or script format, these are often *so* frequently found to be bad that they are outright blocked at the SMTP servers. If you do not have a block in place for sure files, this first rule is sure to be correct most of the time with the possible exception of known safe sources.

The first "real" check would be for the reputation of the hash of the attachment. Since hash values will be unique and there are so many APIs available for these types of checks, ideally this move can be either fully or semiautomated in your incident management system or SIEM. The best scenario would be a hash check against all incoming attachments for matches against a blacklist provided by either a threat intel or sandbox vendor. Since this can be a very high-volume proposition, at least checking hashes of known malicious filetypes could be a step back from fully automated checking, with the even less resource intense version being a "one-click" check for hashes entered in your IMS, such as how Cortex can be used for this in TheHive for an entered observable.

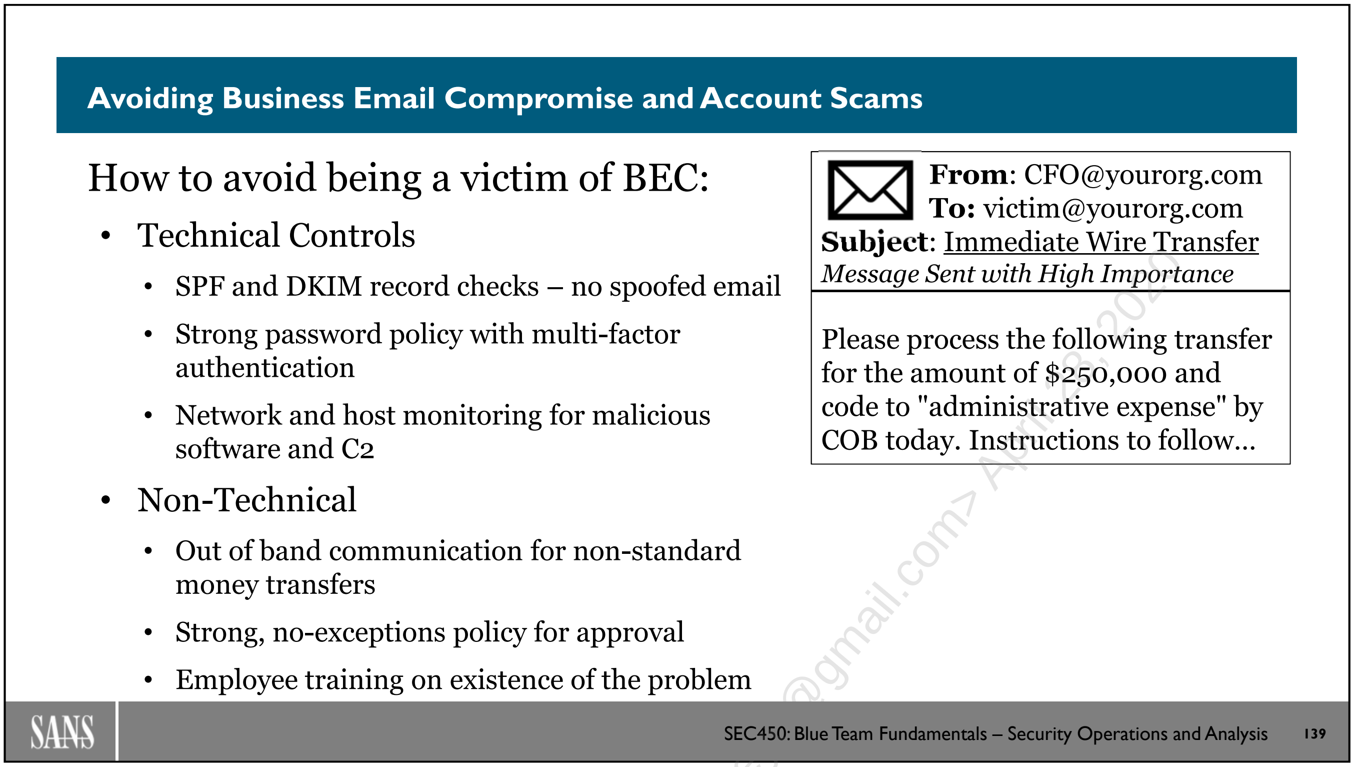
If the hash check comes back unknown, it means the file is either good, or unknown bad. The next best way to detect unknown bad would be to submit it to an automated sandbox. While this isn't necessarily the fastest answer (opening the file and looking at the contents could be faster), it can also be automated, so the actual analyst time required to do the check is minimal. Again, these checks should be ideally fully automated for files that reach this point, or at least a single-click submit away from being assessed. Since many sandboxes such as Cuckoo sandbox will produce a screenshot of the open file, you may get both methods wrapped into a single investigation action. Checking for odd activities such as weird processes spawning, the running of macros, or unexpected network activity from the file in the sandbox can be an easy check for evil.

Finally, if the automated sandbox and hash checks produce nothing, you may need to fall back to manual analysis. Although malware reverse engineering can be a big and exciting topic, there is not enough space to enumerate the full instructors to do it here. In summary, at this point, the next step would be to open the file in a virtual machine and see if you can find anything malicious about it, performing both static and dynamic analysis. Does a macro try to run? Does a program crash? If it is a script or macro, checking for obfuscation can be a quick and accurate verification.

There is an endless supply of filetypes that can be weaponized for phishing, but the types we most typically see tend to settle on a few categories.

- Directly executable files: These files are the "Portable Executable" standard file format used by Windows (you could, of course, send executables for Linux or MacOS as well, but spammers tend to play to the largest install base). Most organizations will have these file types blocked, but if you do not, users simply need to double-click the file for the damage to occur. The trickier formats such as .scr (screen savers) or .cpl (control panel items) tend to work better since they are more likely to slip by .exe filetype mail filters, and users don't tend to recognize them and thus may be less likely to think they are malicious.
- Scripts: Scripts are another common attack vector. Sending a JavaScript or vbs file can be just as malicious as any program. These scripts will be able to execute using the built-in Windows script interpreter (cscript.exe) with a double-click, just like any other file. HTA is a more obscure type of script but is equally as dangerous. HTA files are HTML Applications and are a format that is only compatible with Windows but allow attackers to run scripting languages the same as the other file types.
- Documents: Everyone is familiar with the macro-enabled Office document attack. It is one of the most common document-oriented phishing types out there and continues to be pervasive because it works so well. Attackers tend to use the pre-2007 (non-XML-based - .doc, .xls) file formats since the newer format makes it clear when a file has a macro (.docx – no macro vs. .docm – macro), making it easy for organizations to filter out macro-containing documents on the file name alone. There are other types of Office document-based attacks as well. The Dynamic Data Exchange method is one that was prevalent until Microsoft shipped a patch that turned it off by default in 2017. There are more obscure weaponizable file types, too, such as RTF and HTA files. RTF files are "rich text" format and when used for phishing, can either contain exploits for the RTF reader itself, or similar content to a normal word document. Depending on what the attacker chose, handle RTF documents with care. PDF documents are also commonly used, sometimes with exploits for the reader itself, and other times combined with social engineering tactics. The most common attacks outside reader exploits are embedding a malicious file, such as an executable inside the pdf so the user can open it from there or linking out to a malicious webpage.
- Compressed formats: Compressed documents are used not directly as an attack themselves, but as a way of covering for the other types of files. Attackers hope that by compressing, and possibly encrypting the files into an archive, they will be more likely to slip by detection and email filters. When a zip file is encrypted, the normal process an email filter would use to decompress and analyze the included files will not work since it doesn't have the password (even though it is written in the email, it cannot grab it in a reliable automated way). So, in many cases, this tactic will indeed work.

If you are interested in doing a deep-dive into malware sample reverse-engineering, check out SANS FOR610. It's an outstanding class that explains how to manually take apart many of these file formats to understand exactly what the attack will do.



Avoiding Business Email Compromise and Account Scams

How to avoid being a victim of BEC:

- Technical Controls
  - SPF and DKIM record checks – no spoofed email
  - Strong password policy with multi-factor authentication
  - Network and host monitoring for malicious software and C2
- Non-Technical
  - Out of band communication for non-standard money transfers
  - Strong, no-exceptions policy for approval
  - Employee training on existence of the problem

From: CFO@yourorg.com
To: victim@yourorg.com
Subject: Immediate Wire Transfer
Message Sent with High Importance

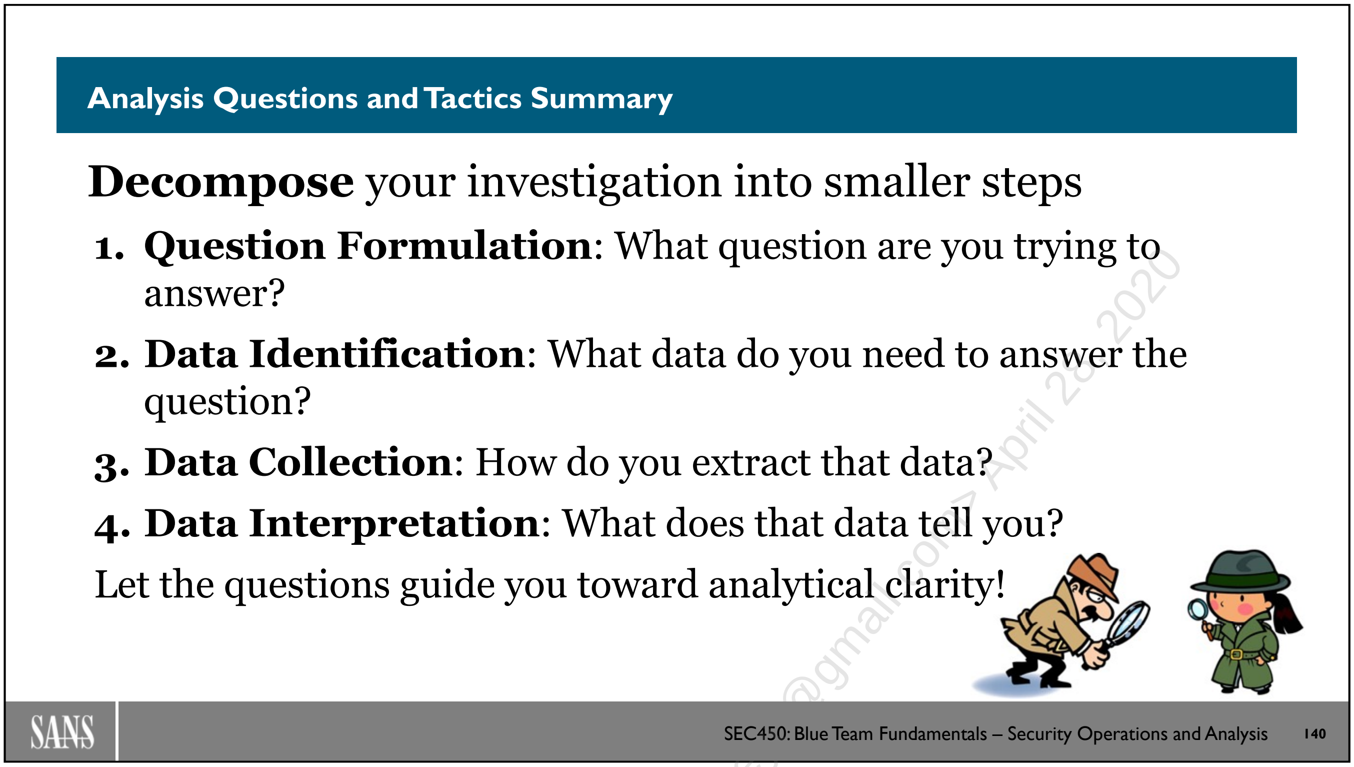
Please process the following transfer for the amount of $250,000 and code to "administrative expense" by COB today. Instructions to follow...

Avoiding Business Email Compromise and Accounting Scams

How then do we avoid these social engineering attacks? It's a considerably difficult problem and the tactic that might highlight the fraud attempt varies, depending on the tactics used. In the easiest case, simple checks like SPF record validation and DKIM can at least make sure email from trusted vendors is not passed through and spoofed. As an analyst, if you see an email you suspect to be a BEC attempt, it's worth looking at the headers in detail and checking if all the hostnames and IP addresses line up to form a valid chain, and if the SPF records of the organization sending the email align with what is expected. Keeping spoofed email out of the organization is one of the best defenses.

Beyond stopping fake emails from arriving in the first place, you can prevent your own organization's accounts from being part of a BEC scam by implementing a strong password policy and forcing the use of multi-factor authentication for those who can transfer money. That way, if their password is ever stolen with malware, the attackers hopefully will not be able to sign into them and use them to fool others. Monitoring of the network and traditional malware defense will also help with this.

Remember, in the worst-case scenario, these are emails that might be coming from the hacked account of someone the victim has corresponded with before, using legitimate infrastructure. How do we stop BEC in this case? The technical controls above must be teamed with a strong policy that does not allow "surprise" request transfers of money to new accounts. Out of band communication should be required to verify account numbers, especially when one changes, and a 2-man rule can be implemented for particularly risky transactions. Finally, employee awareness plays a huge role in stopping these kinds of compromises. If the potential victims knows that this is something that may be attempted and can spot the signs, they are way less likely to fall for it if it does happen.



Analysis Questions and Tactics Summary

Decompose your investigation into smaller steps

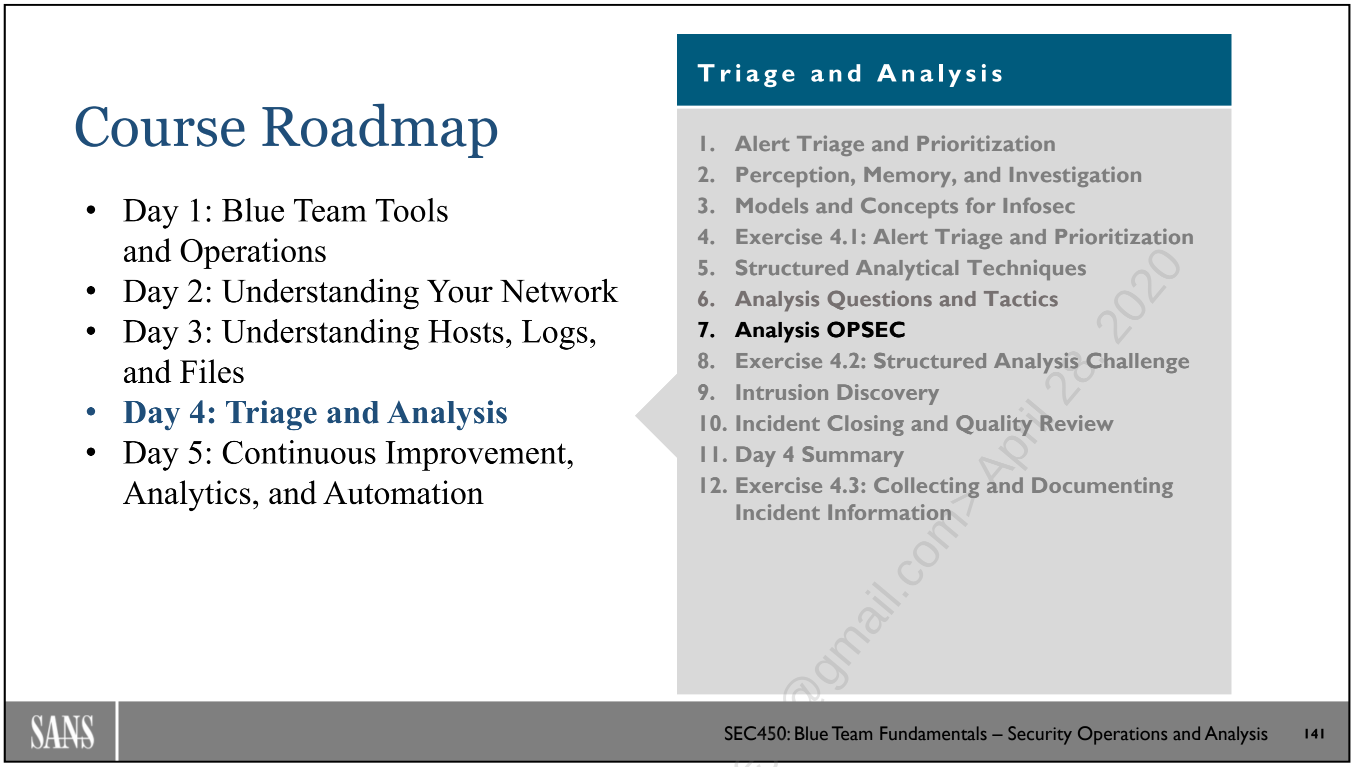
1. **Question Formulation**: What question are you trying to answer?
2. **Data Identification**: What data do you need to answer the question?
3. **Data Collection**: How do you extract that data?
4. **Data Interpretation**: What does that data tell you?

Let the questions guide you toward analytical clarity!

SANS
SEC450: Blue Team Fundamentals – Security Operations and Analysis
140

Analysis Questions and Tactics Summary

In this section we reviewed some of the questions that can help guide you through any investigation and some specific tactics that may come in handy. By breaking the larger "What happened here?" question down into smaller pieces, it's easy to gain clarity on exactly the steps the analysis must follow. The common steps of most investigations are 1. formulating a good question, 2. identifying what data is needed to answer that question, 3. efficiently sourcing that data from our tools, and 4. interpreting that data. Thinking of answering the larger question in smaller chunks helps give a model for how analysis should work and walks through the process in a way that is easily repeatable. If at any point you feel lost on what to do next, refer back to these questions as a guidance toward clarity.



Course Roadmap

- Day 1: Blue Team Tools and Operations
- Day 2: Understanding Your Network
- Day 3: Understanding Hosts, Logs, and Files
- **Day 4: Triage and Analysis**
- Day 5: Continuous Improvement, Analytics, and Automation

Triage and Analysis

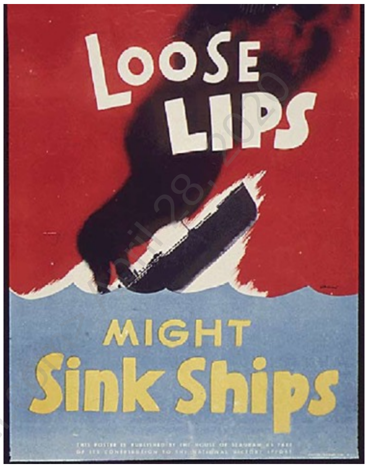
1. Alert Triage and Prioritization
2. Perception, Memory, and Investigation
3. Models and Concepts for Infosec
4. Exercise 4.1: Alert Triage and Prioritization
5. Structured Analytical Techniques
6. Analysis Questions and Tactics
7. **Analysis OPSEC**
8. Exercise 4.2: Structured Analysis Challenge
9. Intrusion Discovery
10. Incident Closing and Quality Review
11. Day 4 Summary
12. Exercise 4.3: Collecting and Documenting Incident Information

This page intentionally left blank.

A Key Investigation Concept: OPSEC

Operational Security (**OPSEC**)

NATO definition: "*The process which gives a military operation or exercise appropriate security, using passive or active means, to deny the enemy knowledge of the dispositions, capabilities and intentions of friendly forces.*"[1]



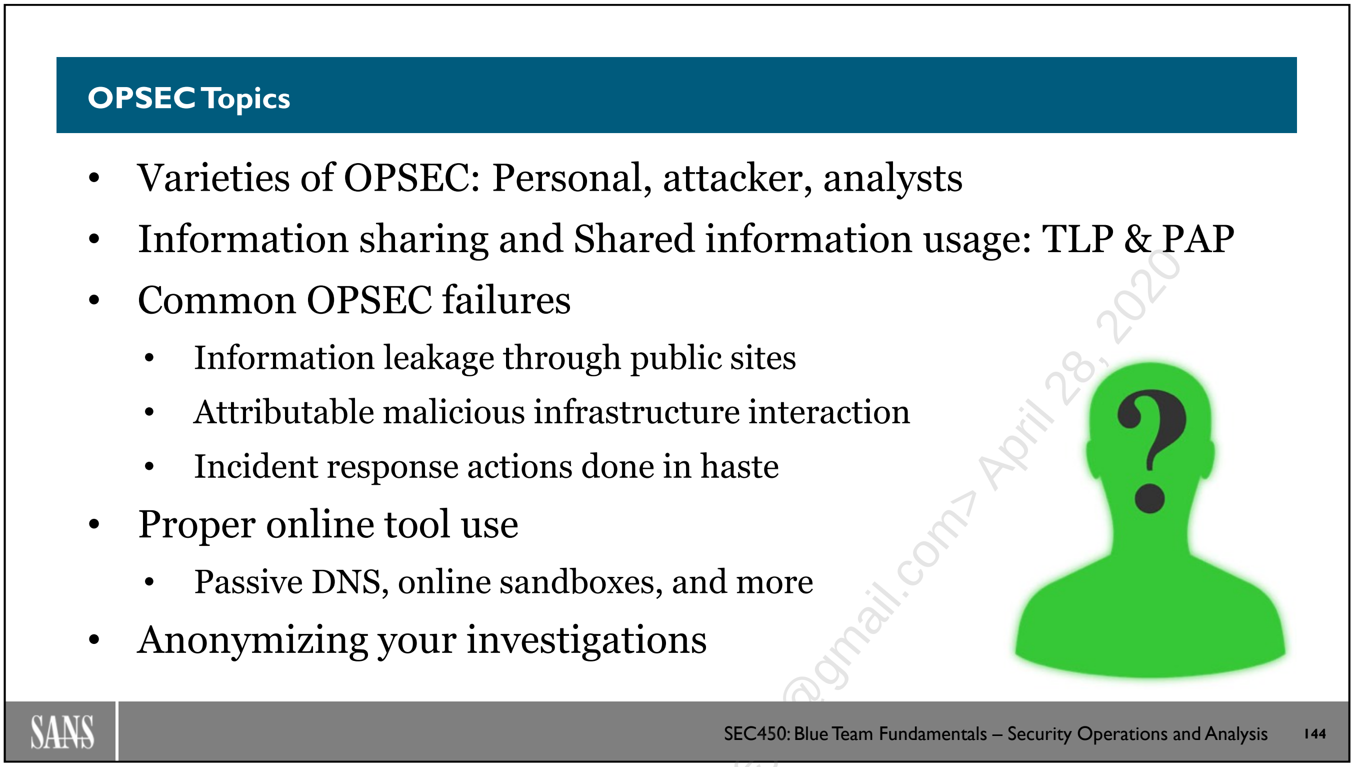
A Key Investigation Concept: OPSEC

An extremely important concept to understand for the blue team is how to maintain operational security during open-source intelligence gathering and data collection. According to the generic NATO definition, OPSEC is "*The process which gives a military operation or exercise appropriate security, using passive or active means, to deny the enemy knowledge of the dispositions, capabilities and intentions of friendly forces.*"[1] For InfoSec, the definition is pretty much identical. We also don't want to leak any information to the enemy about what we know. Considering the two broad camps that you can place attackers into, targeted and opportunistic attacks, OPSEC requirements for each will differ and drive the type of actions we might take after we identify an incident of either type.

In an opportunistic attack, adversaries spray spam and viruses all over the internet without even knowing who they're pointed at in an attempt to compromise anyone they can. In these cases, OPSEC is not as big of a concern. Teams around the globe will be investigating the infrastructure of adversaries like this. However, in a targeted attack scenario, things are different. Once we have identified an attack, we have a small tactical advantage. We can use the discovered info to stealthily analyze the indicators and files used for attack and pick apart the attackers' intentions without them knowing that they've been caught. This ideally buys the blue team time to watch, prepare, and plan a decisive remediation plan that can take effect all at once, locking the attackers out. This type of planning might not be possible if the attackers realize they have been spotted as this could inspire them to change tactics, forcing the blue team to start over on detection and analysis.

An example: Let's say you receive a phishing email with a targeted phishing link to a cloned version of your own website, a common tactic. You can bet that the adversary is watching the phishing site logs very closely for any interaction with the page. If they can tell that instead of victims visiting the page, it is a blue team member running it through a sandbox or other obvious investigation tools, they may immediately switch tactics to use another domain and virus—one that you don't know about, causing you to start the cycle over. For this reason, it is extremely important to ask yourself the question "could this be a targeted attack" before taking any action to actively investigate anything hosted externally or submitting data to publicly available websites. There is nuance to be aware of here and some of it only comes with being familiar with attacker techniques and defensive tools. We will dig into these non-obvious "gotchas" further throughout this module.

[1] NATO Glossary of Terms and Definitions: https://standards.globalspec.com/std/10275442/nato-aap-06



OPSEC Topics

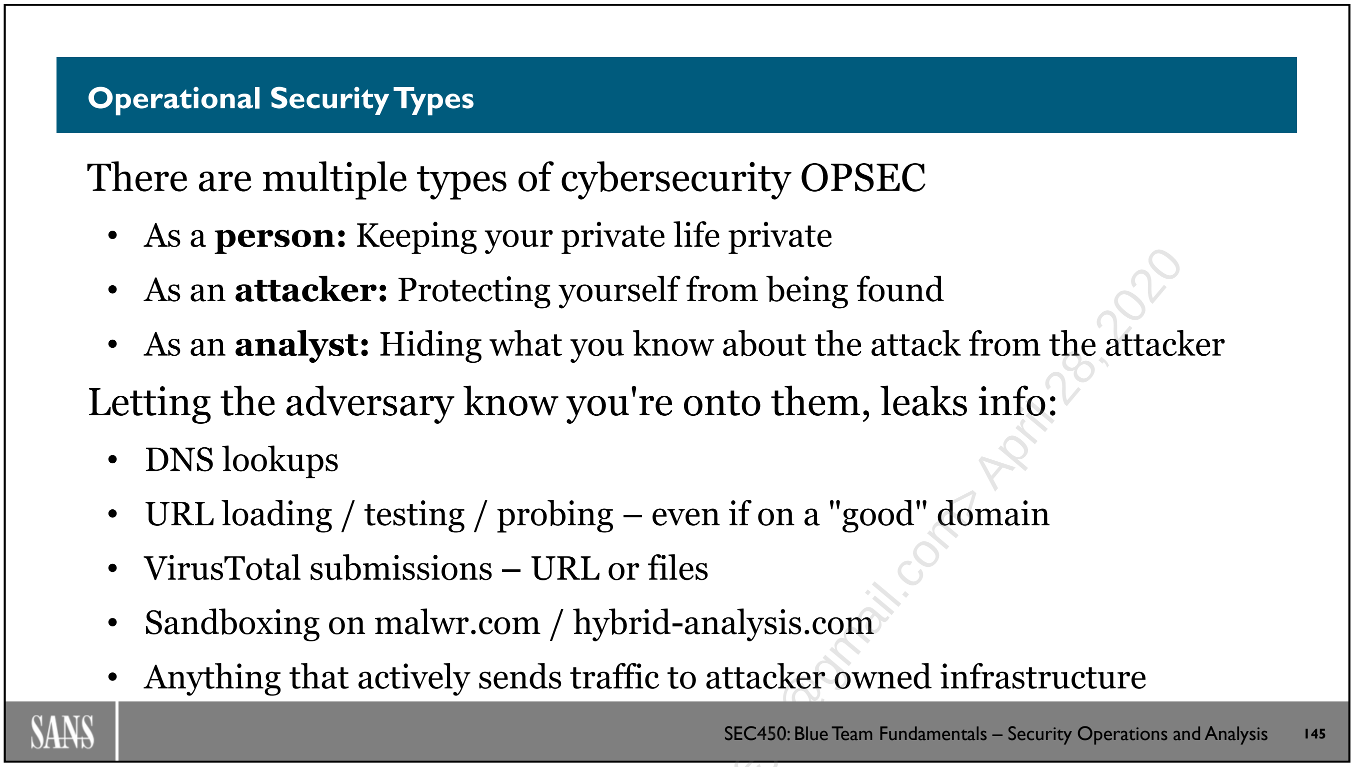
- Varieties of OPSEC: Personal, attacker, analysts
- Information sharing and Shared information usage: TLP & PAP
- Common OPSEC failures
  - Information leakage through public sites
  - Attributable malicious infrastructure interaction
  - Incident response actions done in haste
- Proper online tool use
  - Passive DNS, online sandboxes, and more
- Anonymizing your investigations

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 144

OPSEC Topics

OPSEC can be broken down into several topics. First, we will describe the type of OPSEC we're talking about as it pertains to the role as an analyst performing investigations. Afterwards, we'll describe some of the standards that have been developed that relate to it that can tell us how information can be shared and how shared information can be interacted with. We'll also discuss common failures and how to understand and avoid them with the proper and careful usage of the available tools and classifications systems like the Traffic Light Protocol (TLP) and Permissible Action Protocol (PAP). Finally, we'll discuss how to anonymize our activities in case we must actively investigate attacker infrastructure.



Operational Security Types

There are multiple types of cybersecurity OPSEC

- As a **person:** Keeping your private life private
- As an **attacker:** Protecting yourself from being found
- As an **analyst:** Hiding what you know about the attack from the attacker

Letting the adversary know you're onto them, leaks info:

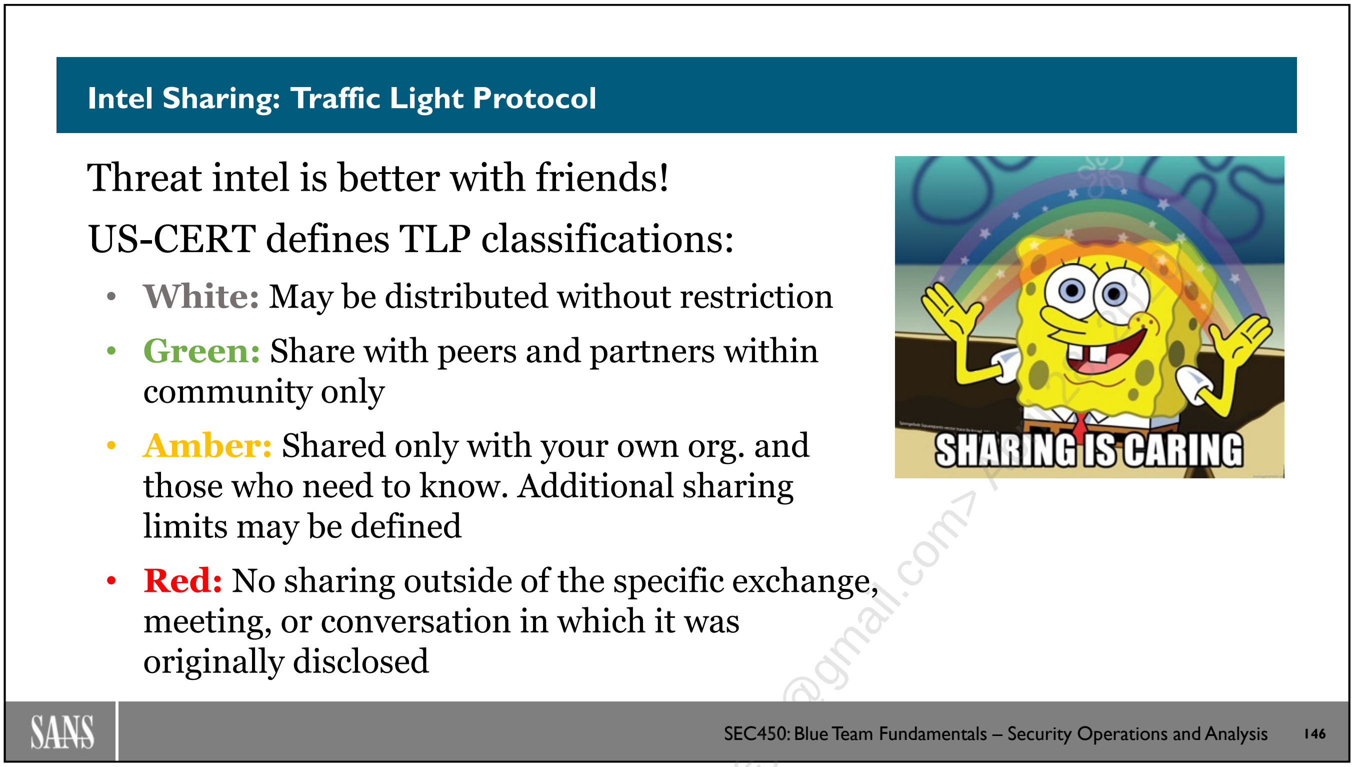
- DNS lookups
- URL loading / testing / probing – even if on a "good" domain
- VirusTotal submissions – URL or files
- Sandboxing on malwr.com / hybrid-analysis.com
- Anything that actively sends traffic to attacker owned infrastructure

Operational Security Types

You may have heard of operational security before, but it was likely in another context such as for attackers or for individuals. Most people have heard of the personal type of OPSEC such as not giving away too much information on social media, saying when and where you are going on trips and the like. You've also likely heard stories like these[1] of OPSEC failures from the attackers that left data or exploits around, exposing them to discovery or getting them caught. Our discussion of OPSEC is centered mainly around not letting *attackers* know what we know, so that they don't shift their tactics and make it more difficult for us to respond. The idea is that even if we have caught them, we want to let them continue thinking that we haven't for as long as possible.

There are lots of ways we can accidentally let an attacker know we are investigating them. One of the most common ways is by submitting searches for domains they use to public sandboxes and investigation sites like VirusTotal or urlscan.io. They can run searches the same way we can, and if their malware hash or domain suddenly shows up as being known, they can tell they've been caught and need to shift tactics. Another common method is by indiscriminately contacting their infrastructure to try to probe it for information, download malware samples, or resolve DNS records. Any time we actively send packets to attacker infrastructure, we run the risk of them seeing it in their own logs and taking action.

[1] https://arstechnica.com/information-technology/2019/01/researchers-discover-state-actors-mobile-malware-efforts-because-of-yolo-opsec/



Intel Sharing: Traffic Light Protocol

Threat intel is better with friends!

US-CERT defines TLP classifications:

- **White:** May be distributed without restriction
- **Green:** Share with peers and partners within community only
- **Amber:** Shared only with your own org. and those who need to know. Additional sharing limits may be defined
- **Red:** No sharing outside of the specific exchange, meeting, or conversation in which it was originally disclosed

Intel Sharing: Traffic Light Protocol

Like everything, threat intelligence gathering is better with friends, but that doesn't mean everyone can and will share everything. Sometimes sources and methods need to be protected and, therefore, a producer of threat intelligence will limit where and how the information given to you can be distributed. This is done by the Traffic Light Protocol standard, which is defined by US-CERT.[1] The definitions for each color level are defined as such.

"**TLP:WHITE:** Disclosure is not limited. Sources may use TLP:WHITE when information carries minimal or no foreseeable risk of misuse, in accordance with applicable rules and procedures for public release. Subject to standard copyright rules, TLP:WHITE information may be distributed without restriction.

TLP:GREEN: Limited disclosure, restricted to the community. Sources may use TLP:GREEN when information is useful for the awareness of all participating organizations as well as with peers within the broader community or sector. Recipients may share TLP:GREEN information with peers and partner organizations within their sector or community, but not via publicly accessible channels. Information in this category can be circulated widely within a community. TLP:GREEN information may not be released outside of the community.

TLP:AMBER: Limited disclosure, restricted to participants' organizations. Sources may use TLP:AMBER when information requires support to be effectively acted upon, yet carries risks to privacy, reputation, or operations if shared outside of the organizations involved. Recipients may only share TLP:AMBER information with members of their own organization, and with clients or customers who need to know the information to protect themselves or prevent further harm. **Sources are at liberty to specify additional intended limits of the sharing: These must be adhered to.**

TLP:RED: Not for disclosure, restricted to participants only. Sources may use TLP:RED when information cannot be effectively acted upon by additional parties, and could lead to impacts on a party's privacy, reputation, or operations if misused. Recipients may not share TLP:RED information with any parties outside of the specific exchange, meeting, or conversation in which it was originally disclosed. In the context of a meeting, for example, TLP:RED information is limited to those present at the meeting. In most circumstances, TLP:RED should be exchanged verbally or in person."[1]

US Cert also defines the following standards for labeling in email and files:

"How to use TLP in email

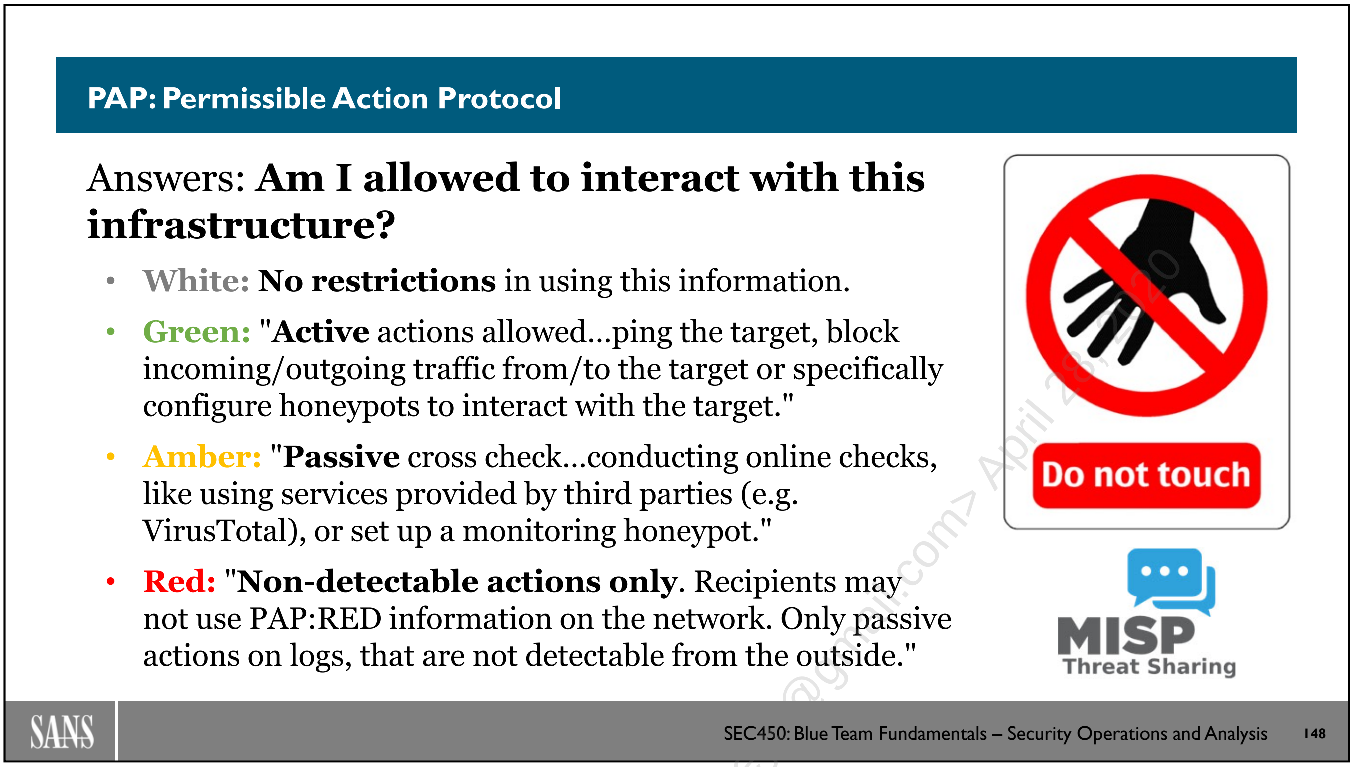
TLP-designated email correspondence should indicate the TLP color of the information in the Subject line and in the body of the email, prior to the designated information itself. The TLP color must be in capital letters:

TLP:RED, TLP:AMBER, TLP:GREEN, or TLP:WHITE.

How to use TLP in documents

TLP-designated documents should indicate the TLP color of the information in the header and footer of each page. To avoid confusion with existing control marking schemes, it is advisable to right-justify TLP designations. The TLP color should appear in capital letters and in 12-point type or greater."[1]

[1] https://www.us-cert.gov/tlp



PAP: Permissible Action Protocol

Answers: **Am I allowed to interact with this infrastructure?**

- **White:** **No restrictions** in using this information.
- **Green:** "**Active** actions allowed...ping the target, block incoming/outgoing traffic from/to the target or specifically configure honeypots to interact with the target."
- **Amber:** "**Passive** cross check...conducting online checks, like using services provided by third parties (e.g. VirusTotal), or set up a monitoring honeypot."
- **Red:** "**Non-detectable actions only**. Recipients may not use PAP:RED information on the network. Only passive actions on logs, that are not detectable from the outside."

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 148

PAP: Permissible Action Protocol

Have you ever seen an indicator from a neighbor organization and wondered, "What am I allowed to do with this?" Of course, you don't want to ruin someone's OPSEC and talk to an attacker server, giving away the information that someone is on to them, but without some way to know the secrecy level, it's hard to make this call. TLP can be used as a proxy for this type of decision, but it's not perfectly aligned for the purpose. To solve this problem, "Permissible Action Protocol" was developed and is built into MISP as another taxonomy that can be used for each individual indicator. It explains exactly what an analyst is allowed to do with a given atomic item of data and how they can interact with it, if at all.

The MISP taxonomy defines the levels in the same traffic light style colors as TLP, but with different meanings listed as follows:[1]

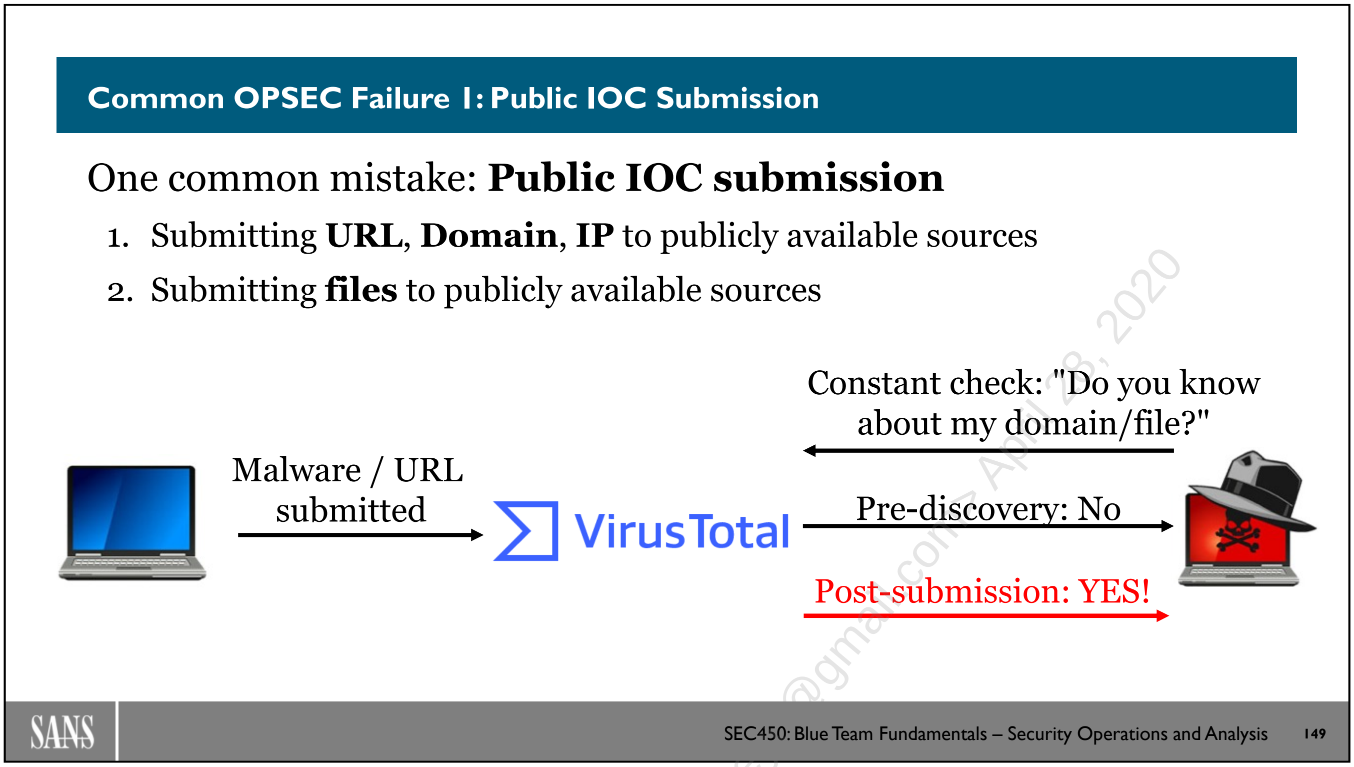
White: No restrictions in using this information.

Green: Active actions allowed. Recipients may use PAP:GREEN information to ping the target, block incoming/outgoing traffic from/to the target or specifically configure honeypots to interact with the target.

Yellow: Passive cross check. Recipients may use PAP:AMBER information for conducting online checks, like using services provided by third parties (e.g. VirusTotal) or set up a monitoring honeypot.

Red: Non-detectable actions only. Recipients may not use PAP:RED information on the network. Only passive actions on logs that are not detectable from the outside.

[1] https://www.misp-project.org/taxonomies.html#_pap



Common OPSEC Failure 1: Public IOC Submission

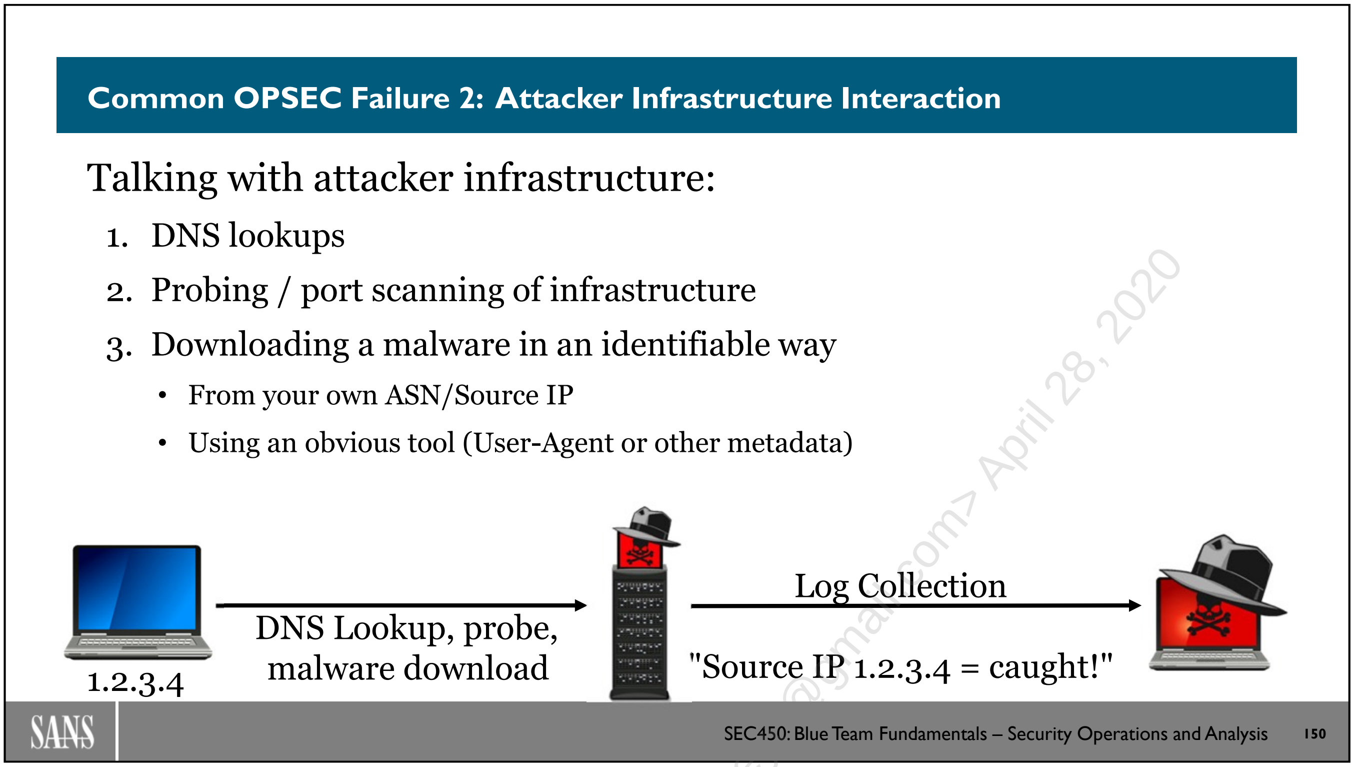
One common mistake: **Public IOC submission**

1. Submitting **URL**, **Domain**, **IP** to publicly available sources
2. Submitting **files** to publicly available sources

Common OPSEC Failures: Public IOC Submission

When it comes to analyzing and investigating potential targeted attack artifacts, there are several mistakes that are the most common. Fortunately, they are easily avoidable once you are aware of what not to do and why. Keep in mind that *targeted* attacks are the key word here. We aren't worried about this for opportunistic attacks since by definition they don't care who they are attacking.

First is submitting IOCs to publicly available sources. This includes everything from hashes to URLs, domains, IP addresses and anything else that could be unique to an attack. Submitting any of these items to a public sandbox that will then turn around and tell others that it has been seen is a beacon saying, "the attack was caught, or is being investigated." The same thing goes for files, but this can be even worse. If you submit files that are part of the attack to something like VirusTotal, not only will the adversary know that you found them, everyone else on Earth can then download evidence of your targeted attack forever. These are the most easily avoidable by ensuring you do research in a way that will not leave a lasting impression for someone else to find. There are too many tools to run over the operations of each in specific, but you *should* at least be aware of the ones you use in your SOC. If you don't know how the tools act, email support or do your own testing with fake data to figure out if they share the information not only with the public, but with other sites.



Common OPSEC Failure 2: Attacker Infrastructure Interaction

Talking with attacker infrastructure:

1. DNS lookups
2. Probing / port scanning of infrastructure
3. Downloading a malware in an identifiable way
   - From your own ASN/Source IP
   - Using an obvious tool (User-Agent or other metadata)

1.2.3.4

DNS Lookup, probe, malware download

Log Collection

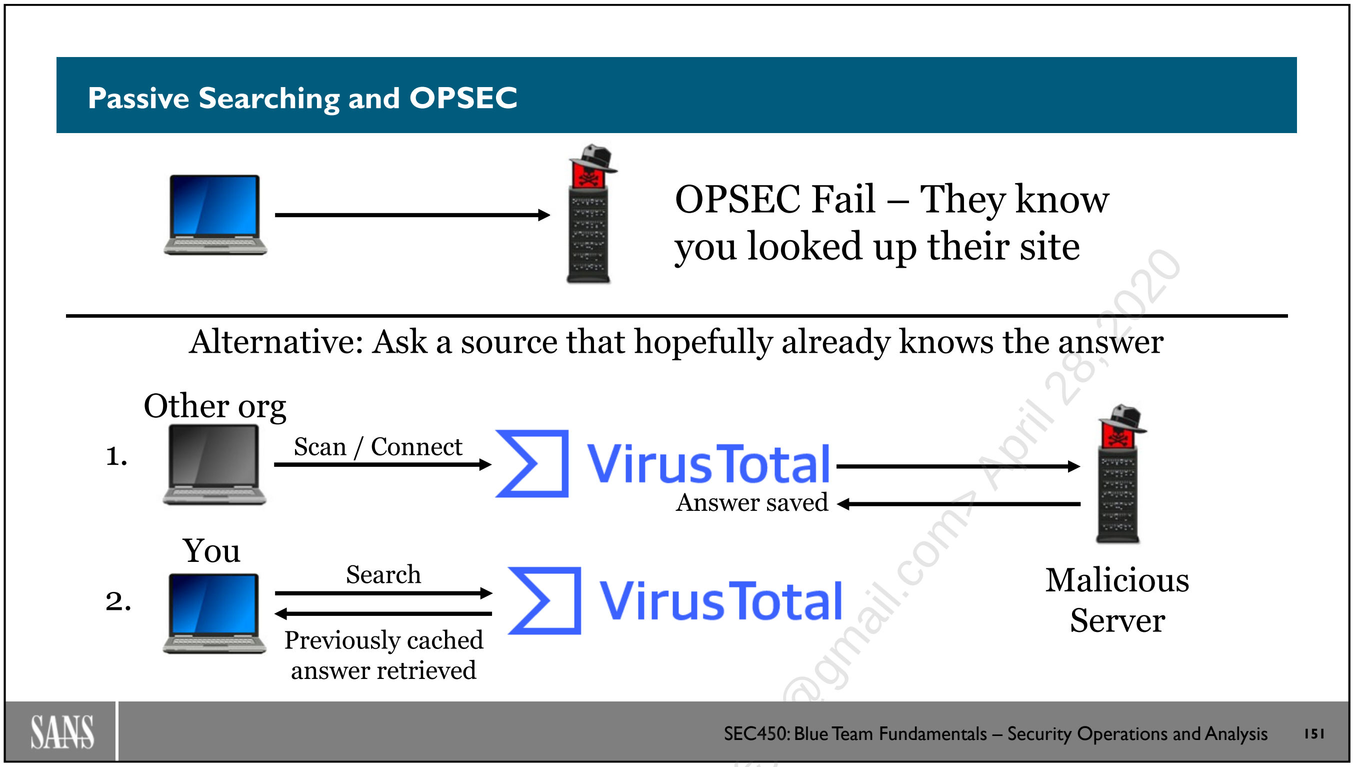
"Source IP 1.2.3.4 = caught!"

Common OPSEC Failure 2: Attacker Infrastructure Interaction

Another common operational security failure is interacting with attacker infrastructure in a way that informs them you are investigating. This is why the Permissible Action Protocol (PAP) system was developed, so that analysts at one company who have shared data don't accidentally have analysts from another company make a mistake and tip off the attacker. This can come in many forms, but a common example could be visiting the URL from a reported targeted phishing attack with a sandbox browser that is clearly not a real victim, doing DNS A record lookups by themselves, or trying to pull down a malware sample with something like wget instead of what the attacker would expect (they can tell from the user-agent unless it is faked). Since the attacker is running these services and seeing their own service logs, you should assume that a DNS request without an associated connection to the service afterwards would be suspicious to them. A connection to a malicious webpage with a Linux or wget User-Agent after an attempt to compromise a Windows machine could send a similar message. To get around this, consider what the adversary would expect a successful attack to look like or avoid it altogether and use passive data sources. For DNS, passive DNS sites like **VirusTotal** and **RiskIQ**.

(PassiveTotal) can be used for looking up information in a way that will not produce any traffic to the attacker. Of course, this assumes the attacker runs their own DNS infrastructure. If you can confirm that they do not own and operate their own DNS, then this is "safe." Sites such as **urlscan.io** and **urlquery.net** can be used as passive sources of information for HTTP sites. The takeaway is that you should *always* use passive sources for investigation whenever possible.

If you must interact with their infrastructure, it's best to do so in either an anonymous way, or in a way that duplicates the way they expect a successful attack to look. Although the tools and setup for this are beyond the scope of this class and tread into the malware analysis realm, a test run with the program they intended to compromise connecting to your own server can be used to craft an exact copy of the request they might expect, and therefore may cause them to serve the malware/exploit to you for capture. To the attacker interacting in this way, it looks from their perspective like the attack worked and was delivered but didn't produce the intended results (compromise with command and control). This will leave them wondering whether they were caught or not, and potentially avoid assuming the security team is actively investigating.



Passive Searching and OPSEC

OPSEC Fail – They know you looked up their site

Alternative: Ask a source that hopefully already knows the answer

1. Other org — Scan / Connect → VirusTotal → Malicious Server; Answer saved

2. You — Search → VirusTotal; Previously cached answer retrieved

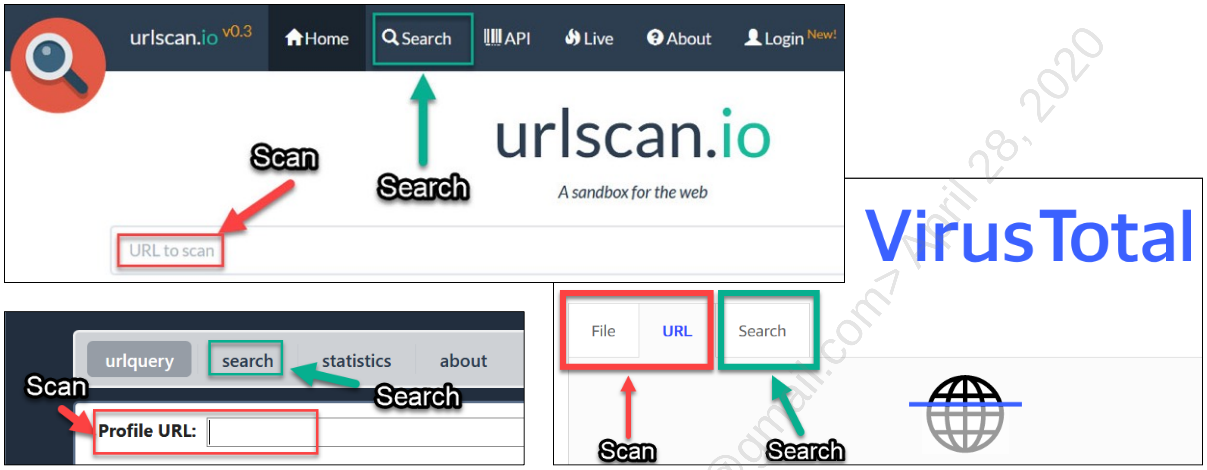
Passive Searching and OPSEC

When analyzing a potentially malicious domain, *especially* if it might be involved in a targeted attack, OPSEC is important to keep in mind. Since the idea is not to let the attacker know you're on to them, you should avoid sending any packets their way that aren't necessary, and this includes DNS requests. While sometimes this won't matter because attackers may share infrastructure amongst many attempted victims, sometimes it can, such as in the case of targeted attacks. Since it is unlikely you will be able to perfectly tell which is which, it's much better practice to always attempt to use passive information sources first, such as passive DNS captured from many malware research sites.

These sites allow anyone to put in websites to actively resolve and investigate, and when analysts do that, the answers to that research are saved. If you come along later and search for previous answers, instead of kicking off a new search for yourself, you can, in theory, safely find out what is known without causing any new traffic to be sent.

OSINT Scan or Search: Choose Wisely

This choice often has **serious** consequences – be careful!!



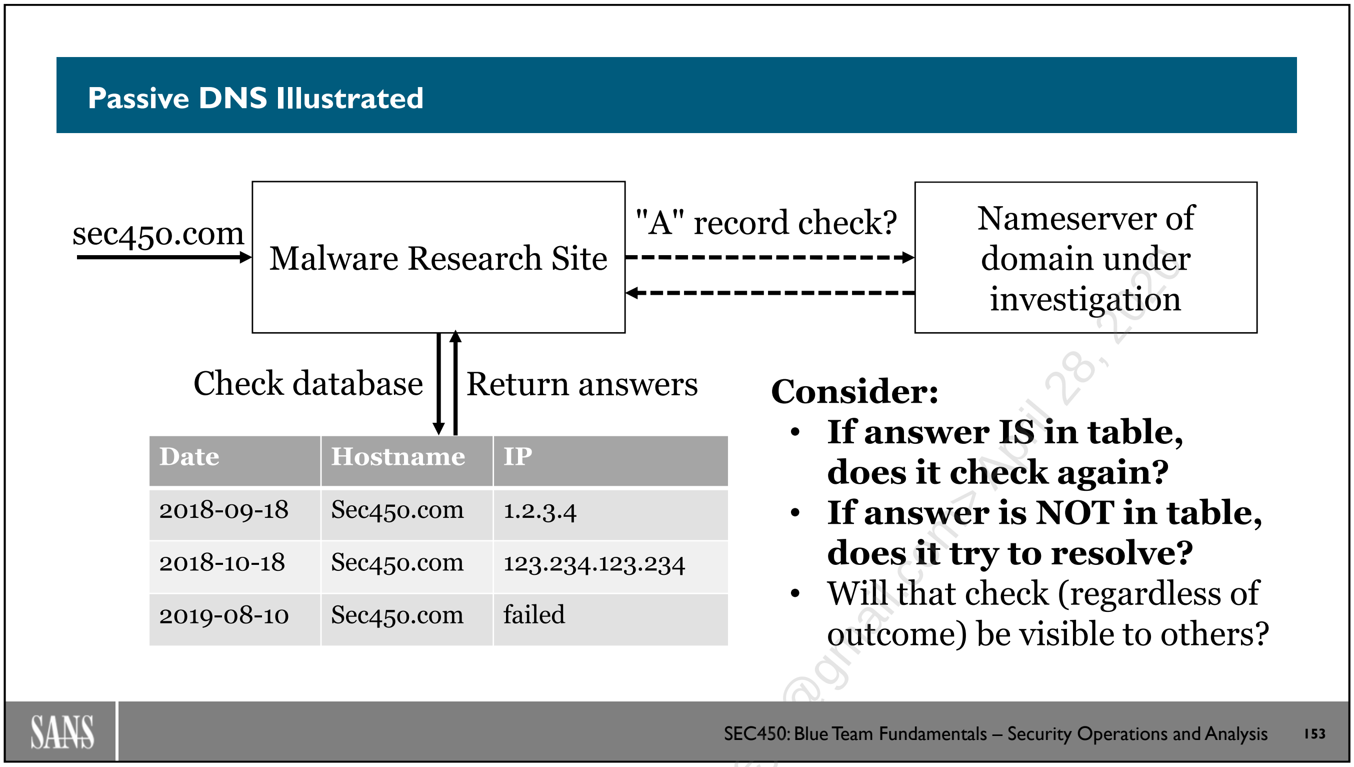
OSINT Tools Scan vs. Search: Choose Wisely

When it comes to using OSINT sites to check the reputation of something, there are usually two options: A passive "search" of the site’s data and an active "scan", and that difference is extremely important.

Although each site has its own nuance that you must test, generally, if you use the *search* button, these sites will check their databases for information about a domain that *already exists but* will not do any active checks. For VirusTotal, for instance, this is true except in the case of a domain VirusTotal has never heard of before. When a new domain is entered, it seems they *will* actively reach out and pull whois and DNS information, a nuance you can only find through testing. The search tab on these sites is usually much safer in terms of operational security and should be your first attempt to get information any time you use them.

If you decide to use the scan options, something very different will happen. If you pick file, you will be actively submitting its content to the site and they can and likely will save and share that file with other partners and researchers. Back to VirusTotal as an example, with access to the paid version of the site, anybody in the world who has a paid VirusTotal Enterprise subscription will be able to download that file afterwards. This is particularly dangerous in the unfortunately common case where an analyst thinks a pdf or word document is a phishing document and uses VirusTotal to check it, only to later realize they have actually submitted a real company invoice or other sensitive record. If you accidentally submit personal or sensitive business information, it is unlikely you will ever get them to take it down. Be aware of this common mistake and do not submit anything to these sites you are not ready for the whole world to potentially see!

If you pick URL scan, most OSINT will actively reach out to the URL and assess it with their suite of tools, saving the results and pages they receive in return. Doing this also will also show the next person who searches for that URL that it has already been scanned, and what the date was of the original and any subsequent analyses. Both file and URL submission are not a good idea in many cases as it can let the adversary know you've caught on to their attack. Exercise caution with VirusTotal and any site like it. In general, it will be safer to look for and ideally exclusively use the "search" functionality before submitting any documents or links of your own. Submitting URLs and files for examination is a trigger you can't unpull.



Passive DNS Illustrated

Date	Hostname	IP
2018-09-18	Sec450.com	1.2.3.4
2018-10-18	Sec450.com	123.234.123.234
2019-08-10	Sec450.com	failed

Consider:

- **If answer IS in table, does it check again?**
- **If answer is NOT in table, does it try to resolve?**
- Will that check (regardless of outcome) be visible to others?

Passive DNS Illustrated

Sites like VirusTotal[1], or the Community Edition of Risk IQ (formerly PassiveTotal)[2] can be used to look for previous hostnames to IP resolutions without sending any traffic in most cases. They do this by caching lookups that have already been done by others and putting the results in a database labeled with the date and IP the site resolved to at the time. Be aware that some sites may break OPSEC in that they may store that someone has attempted to look up a particular domain and tell the next searcher who asks, or cause the domain to be looked up from the service itself in an attempt to get a more recent date or add an answer to the database (remember, these services do have to populate the tables somehow after all).

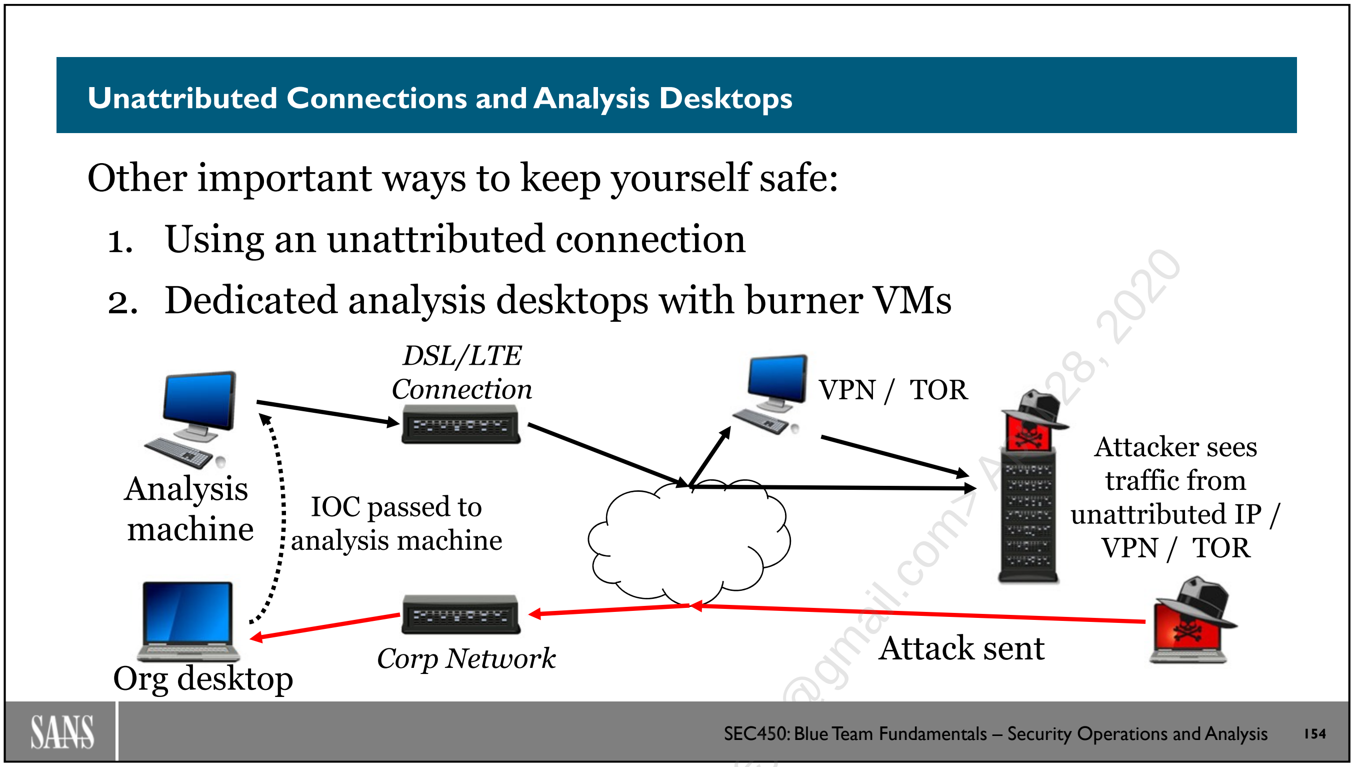
Here are the questions you should know about a service before using it for sensitive lookups.

1. If the answer is in the table, will it cause a new request to be made to refresh the entry?
2. If the domain has never been resolved by the service, will it try to do so for the first time? (this would break the whole point of doing this in the first place)
3. Will anyone else that visits the site know that this domain was looked up already?

How do you know which tool does what? Consult the site documentation and hopefully a description of the site's behavior will be there. If it isn't, you'll have to test it, put in a known non-existent domain once, and then a second time and see if it references that it had already been checked once. To see if traffic is being sent, you could register your own domain and set up a server under your control as the nameserver and check traffic hitting it at the time of the lookup. Hopefully, going to this extreme won't be necessary, but if you can't find a description, it might be necessary.

[1] https://virustotal.com

[2] https://community.riskiq.com/



Unattributed Connections and Analysis Desktops

Other important ways to keep yourself safe:

1. Using an unattributed connection
2. Dedicated analysis desktops with burner VMs

Analysis machine

DSL/LTE Connection

VPN / TOR

Attacker sees traffic from unattributed IP / VPN / TOR

IOC passed to analysis machine

Org desktop

Corp Network

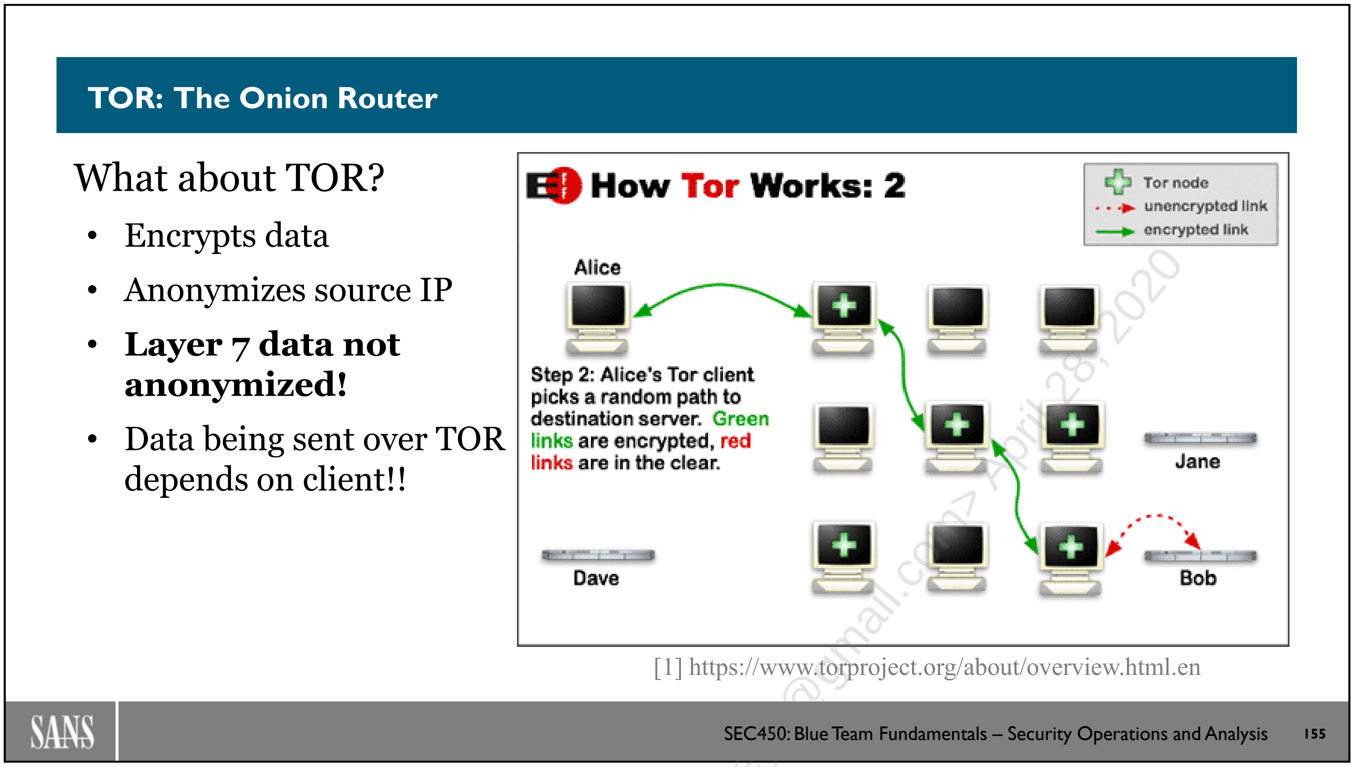
Attack sent

Unattributed Connections and Analysis Desktops

Another method to improve operational security is to use an unattributed line for service interaction with the hope that they used the same infrastructure to attack at least a *few* victims. This is an important caveat to this method. If your attacker has used dedicated and unique IOCs and infrastructure to attack you, *any* interaction with it whatsoever, regardless of the source, will inform them of your investigation. Regardless, it's considered a best practice to use an unattributed line for interaction with any attacker infrastructure as there's no good reason to give them any information in the first place, even if you don't suspect they're looking to use it. Often, this type of setup is implemented in the SOC as a separate DSL line or LTE hotspot that the attacker would have no way to associate with the organization they attempted to attack. It also makes it easy to share among multiple analysts. Beyond this, TOR or VPNs can be used as well to mask the true source IP of your interaction.

Along with the dedicated unattributed connection, another best practice is using virtual machines that can be thrown away or reverted from a dedicated "analysis machine" to perform the interaction. The idea is, if you get an alert that contains a link to a malware download or domain you want to investigate directly, you can bring that information over to another safe machine (via USB or other secure mechanism) on a separate internet connection and perform the analysis from there. This analysis machine should *absolutely not* be connected to the real corporate network at any time given that you are using it for probing known malicious URLs and handling malware. These VMs and the host itself should generally run either some flavor of Linux or a pre-made distribution for malicious file and URL analysis like REMnux[1] (of Lenny Zeltser's FOR610 class) to avoid even possibly being the victim of the grand majority of malware. The host should be kept secure while all malware analysis is done in virtual machines. This keeps you from needing to wipe the analysis machine if it accidentally (or purposefully) becomes infected. Keeping these desktops totally unassociated with the corporate network, accounts, and data allows them to be used for safe, unattributed investigation of any source on the internet.

[1] https://remnux.org/



TOR: The Onion Router

What about TOR?

- Encrypts data
- Anonymizes source IP
- **Layer 7 data not anonymized!**
- Data being sent over TOR depends on client!!

Step 2: Alice's Tor client picks a random path to destination server. Green links are encrypted, red links are in the clear.

[1] https://www.torproject.org/about/overview.html.en

TOR: The Onion Router

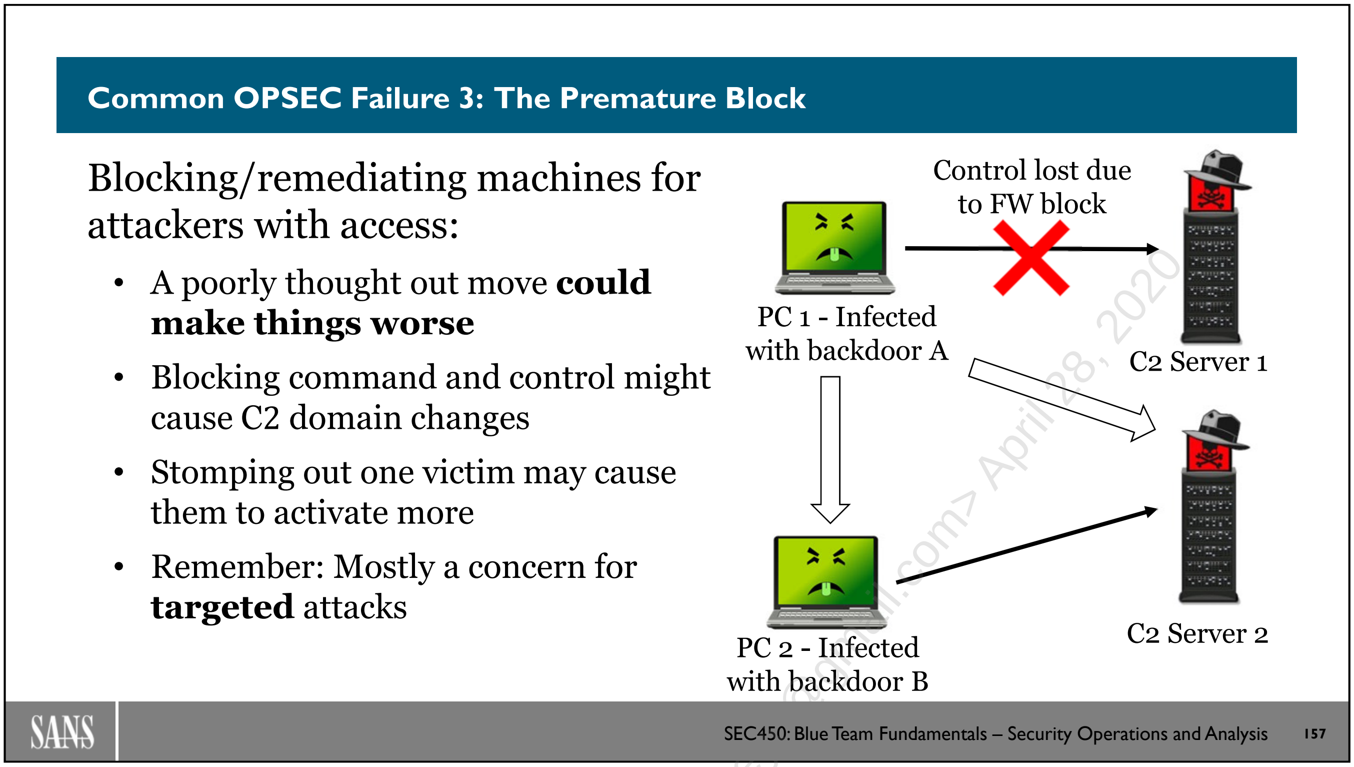
Another option for anonymization is using TOR, which stands for The Onion Router because of the way it works by wrapping data in multiple successive layers of encryption that are peeled back as the data traverses from the client to the destination. In the photo above provided by the EFF[1], each of the three relays the data goes through would have its own encryption key that is applied by Alice. She was using the last stops key first, then the second to last stop's key, then the initial relay's key. That way, as the data moves through the network, each server only knows where the packet came from, and where to send it based on decrypting its own layer. This keeps all nodes "in the dark" about as much as possible. As a result, the service that Bob is running will log the third relay as the sources of the connection, not Alice's computer. In *theory,* the only way to unmask the source of the network traffic is to control all TOR nodes in the whole chain so that you can put the source/destination puzzle back together, which no one should be able to do as long as there are enough TOR nodes run by volunteers and your path is chosen at random. In practice, timing attacks by governments and other OPSEC mistakes are possible, so do not fall into the trap of thinking the system is foolproof.

The caveat to TOR is that it *does not anonymize application layer data*, so if Alice is logging into Bob's service with her username and password, Bob can still tell that it's likely Alice on the other end of the connection. This means that if you want to use TOR for anonymization from attackers, not only do you have to use the service for *all* network interactions, DNS and all (which is not a given and depends on how you connect to the TOR network), but you also must avoid sending any identifying content in your packets. If an attacker sends your organization a unique link in a phishing email and you visit it via TOR, they still know you are investigating it—no one else could possibly have that unique URL! These are the less obvious things that newer analysts can sometimes look over, which is why it is important to fully understand your tools and how OPSEC can be blown with a simple misstep.

One of the *safest* (I would never call any method foolproof) ways of using TOR is the Tails Linux distribution. The idea is that the whole VM is set up to have no identifying information and is only capable of sending network traffic via the TOR network exclusively.[2] If you're going to investigate something using TOR, doing it from a Tails VM goes a long way to prevent simple mistakes, but it's still up to you to keep identifying information like unique URLs and cookies from being sent in your interactions.

[1] https://www.torproject.org/about/overview.html.en

[2] https://tails.boum.org/



Common OPSEC Failure 3: The Premature Block

Blocking/remediating machines for attackers with access:

- A poorly thought out move **could make things worse**
- Blocking command and control might cause C2 domain changes
- Stomping out one victim may cause them to activate more
- Remember: Mostly a concern for **targeted** attacks

Control lost due to FW block
PC 1 - Infected with backdoor A
C2 Server 1
C2 Server 2
PC 2 - Infected with backdoor B

Common OPSEC Failure 3: The Premature Block

As we will discuss in more detail in a later section, even a sudden block of malware or command and control connections could be enough of a tip-off to cause adversaries to change their attack. Though this isn't the case in many scenarios, for attackers that are already potentially on the inside of the network, acting hastily without considering the consequences and how it will look for the attack could also cause an issue. If the adversary has had constant command and control contact with a desktop for several days and suddenly it disappears, they can assume you likely found them. The question is, did they use the time with that access to provide themselves any secondary backdoors to use in this scenario, and with advanced attackers the answer is probably yes. Since you can assume this will be their line of thinking, advanced attackers must be investigated with care to not spook them into changing their tactics.

Here's an example of how this scenario could play out given the drawing on the slide:

1. The attacker breaks in and gets access to PC1 with backdoor A and uses it to infect PC2 with Backdoor B (a different backdoor with a different primary command and control server).
2. The SOC finds out about PC1 being infected and talking to command and control server 1 and decides to implement a block on that domain to contain PC1's infection.
3. Backdoor A falls back to its secondary command and control server, C2 Server 2, which may or may not use a totally different protocol and method of communication than what was done with C2 server 1.
4. The security team does not notice this, and the attacker can use C2 server 2 to re-establish themselves within the environment if need be.

Alternatively, perhaps command and control server 1 traffic and backdoor A are both found at the same time, completely stopping PC1 from communicating with the attacker. Since the attacker has thought ahead and already infected another machine with a totally different backdoor, they will still have access. In all likelihood, the security team will search all devices for backdoor A and signs of communication with C2 server 1, but since they did not find the other machine at the same time, the adversary stays present in the environment, and can use PC2 to re-establish themselves as needed.

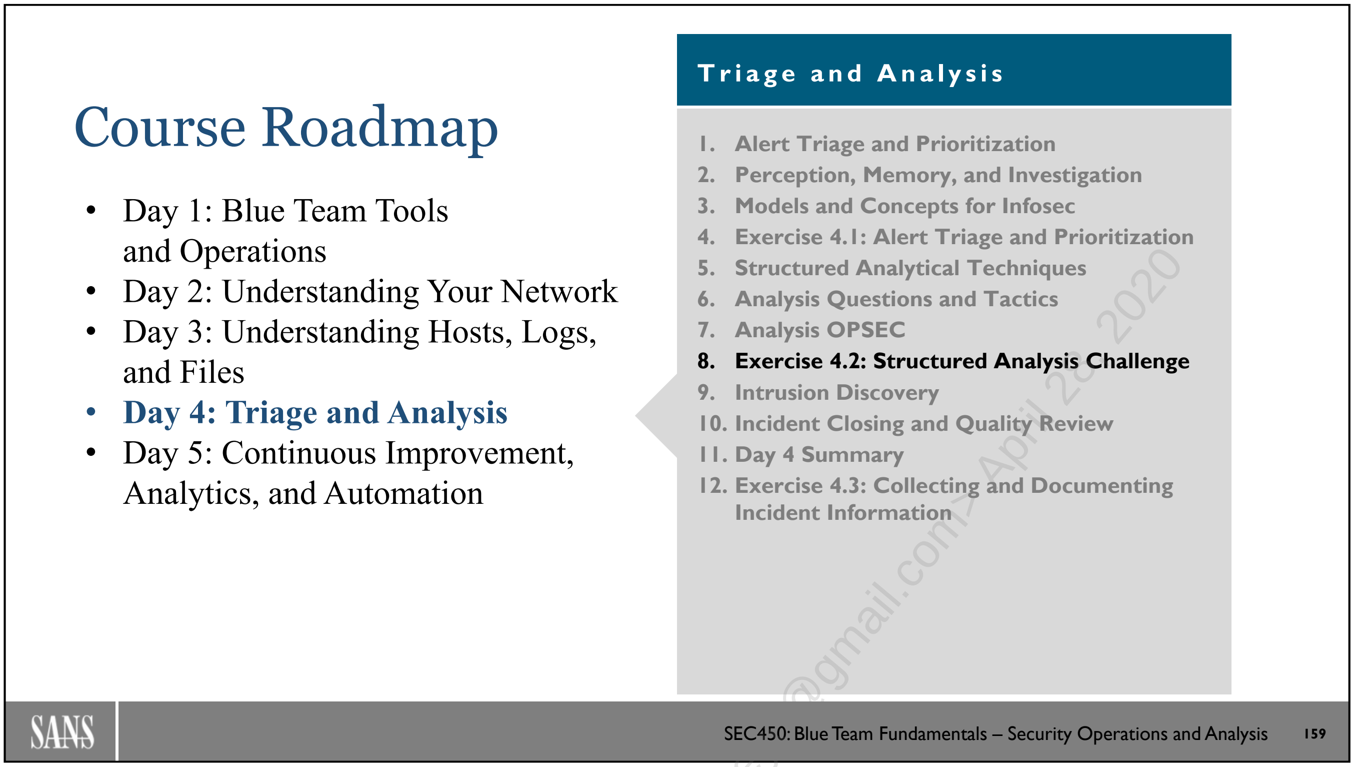
Analysis OPSEC Summary

Analysis has some pitfalls, but those can be easily avoided

- Do not prod attacker infrastructure without good reason
  - Use Passive DNS searches where possible
  - Utilize sandboxes with recorded responses for URL investigation
  - If you do interact with malicious infrastructure, use unattributed lines
- Keep IOCs/files to yourself unless told otherwise
- **TLP** tells you what you can share
- **PAP** guides what you can interact with

Analysis OPSEC Summary

Analysis involving malware samples and attacker infrastructure is sometimes necessary. When considering dealing with files and attacker servers, be sure to exhaust passive sources of information and information already submitted to OSINT sources before submitting IOCs to public sandboxes or talking with the actual malicious infrastructure. In the case of advanced attackers, this can "show your hand" and expose to attackers that you're on to their methods, likely inspiring them to change course and make your life more difficult. When you are touching attacker infrastructure, it's best to use anonymous connections or TOR and ensure that the packets you send to them look like the attack working, not like a defense team probing them. Whether or not you can do any of these activities or share information that you have been given from other organizations should be labeled with a TLP and PAP color, as should any information you disseminate out into the community.



Course Roadmap

- Day 1: Blue Team Tools and Operations
- Day 2: Understanding Your Network
- Day 3: Understanding Hosts, Logs, and Files
- **Day 4: Triage and Analysis**
- Day 5: Continuous Improvement, Analytics, and Automation

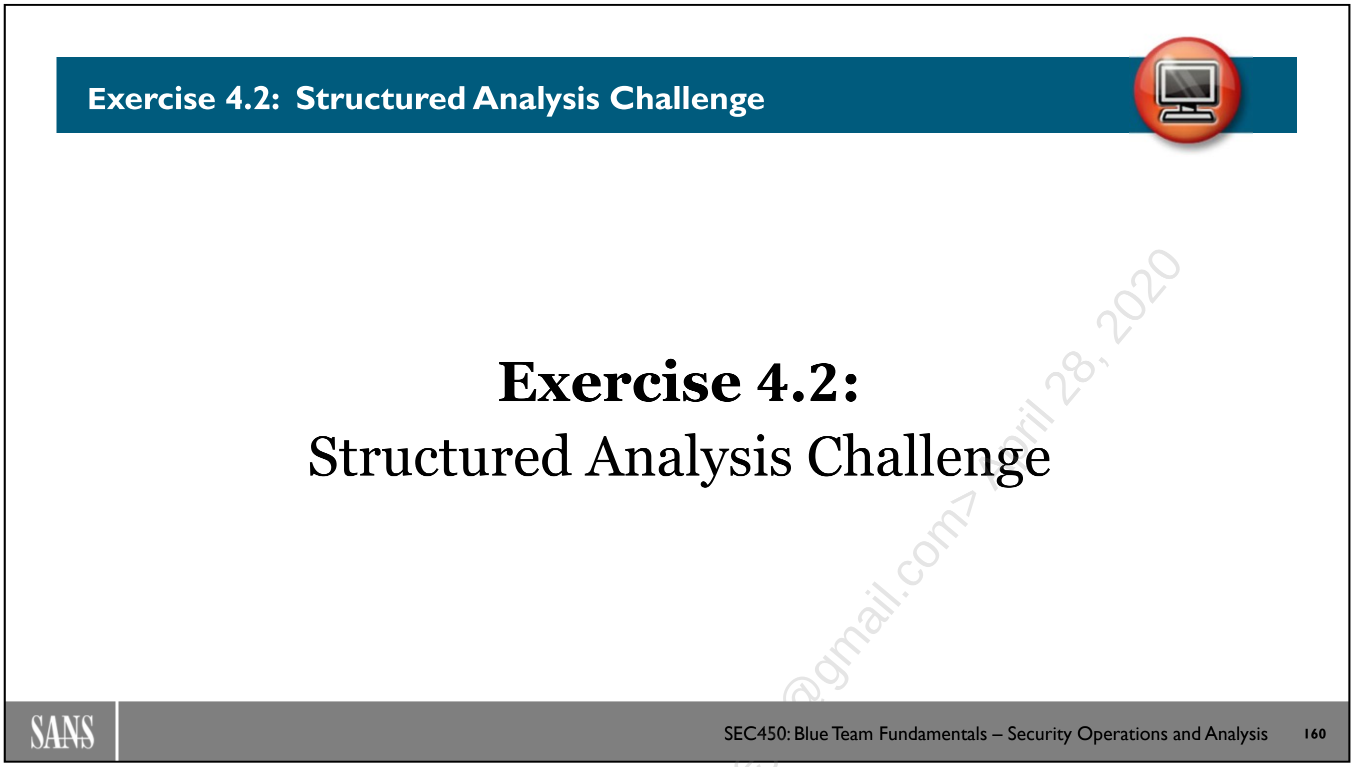
Triage and Analysis

1. Alert Triage and Prioritization
2. Perception, Memory, and Investigation
3. Models and Concepts for Infosec
4. Exercise 4.1: Alert Triage and Prioritization
5. Structured Analytical Techniques
6. Analysis Questions and Tactics
7. Analysis OPSEC
8. **Exercise 4.2: Structured Analysis Challenge**
9. Intrusion Discovery
10. Incident Closing and Quality Review
11. Day 4 Summary
12. Exercise 4.3: Collecting and Documenting Incident Information

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 159

This page intentionally left blank.



Exercise 4.2: Structured Analysis Challenge

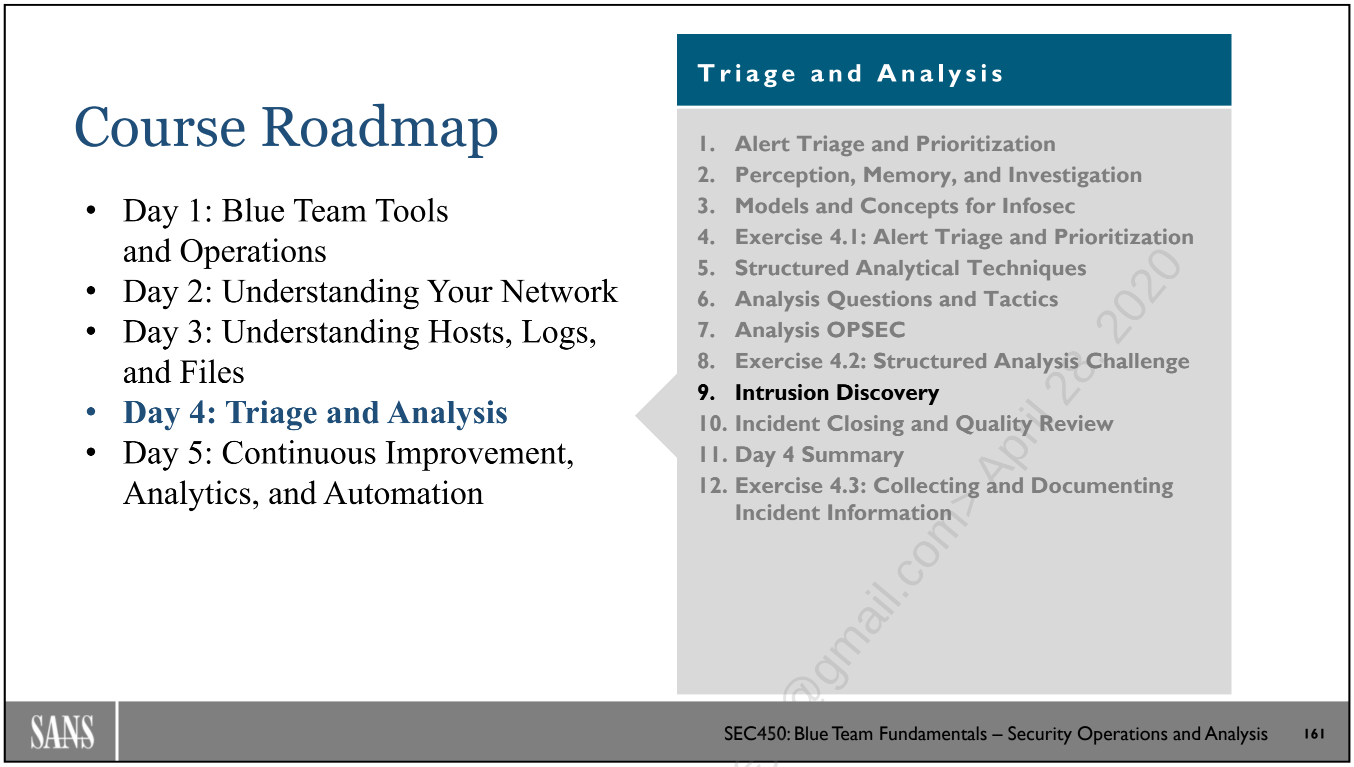
Exercise 4.2:
Structured Analysis Challenge

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 160

Exercise 4.2: Structured Analysis Challenge

Please go to Exercise 4.2 in the SEC450 Workbook or virtual wiki.



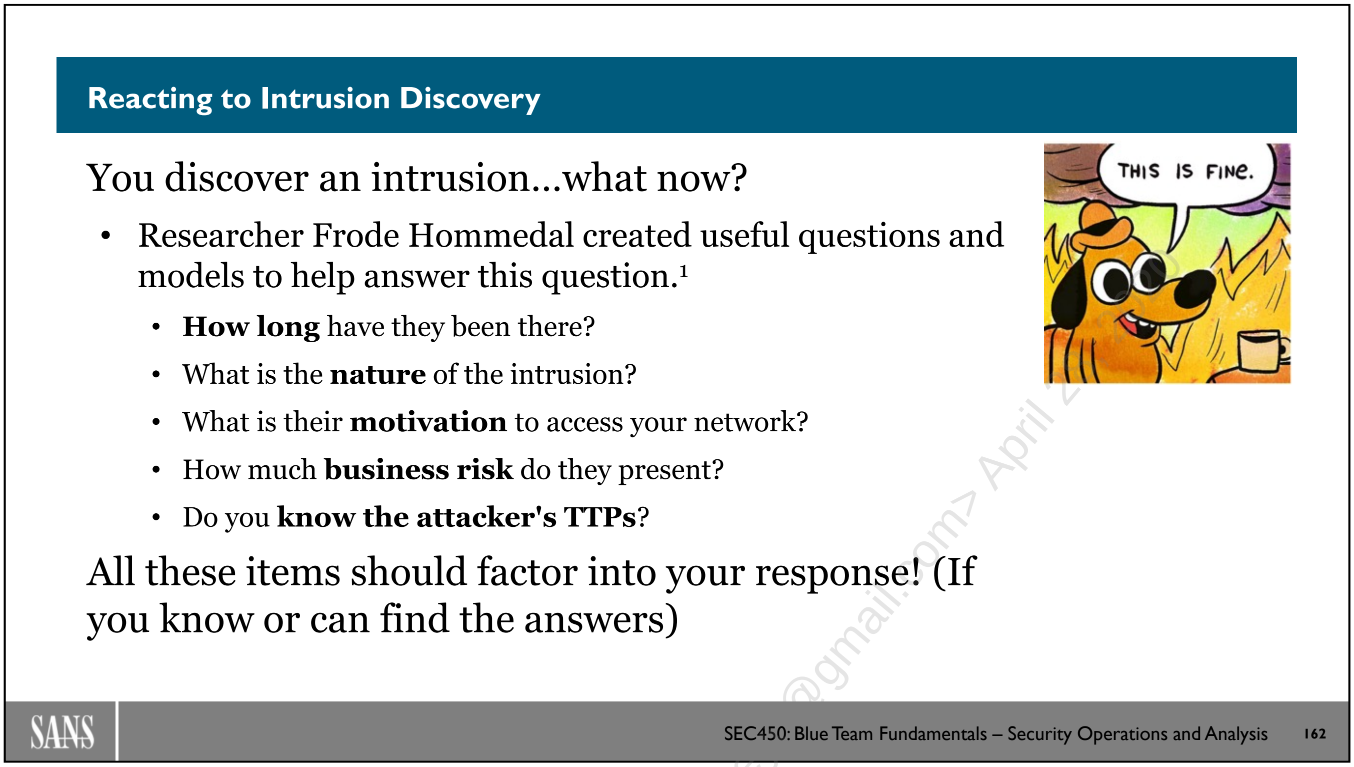
Course Roadmap

- Day 1: Blue Team Tools and Operations
- Day 2: Understanding Your Network
- Day 3: Understanding Hosts, Logs, and Files
- **Day 4: Triage and Analysis**
- Day 5: Continuous Improvement, Analytics, and Automation

Triage and Analysis

1. Alert Triage and Prioritization
2. Perception, Memory, and Investigation
3. Models and Concepts for Infosec
4. Exercise 4.1: Alert Triage and Prioritization
5. Structured Analytical Techniques
6. Analysis Questions and Tactics
7. Analysis OPSEC
8. Exercise 4.2: Structured Analysis Challenge
9. **Intrusion Discovery**
10. Incident Closing and Quality Review
11. Day 4 Summary
12. Exercise 4.3: Collecting and Documenting Incident Information

This page intentionally left blank.



Reacting to Intrusion Discovery

You discover an intrusion…what now?

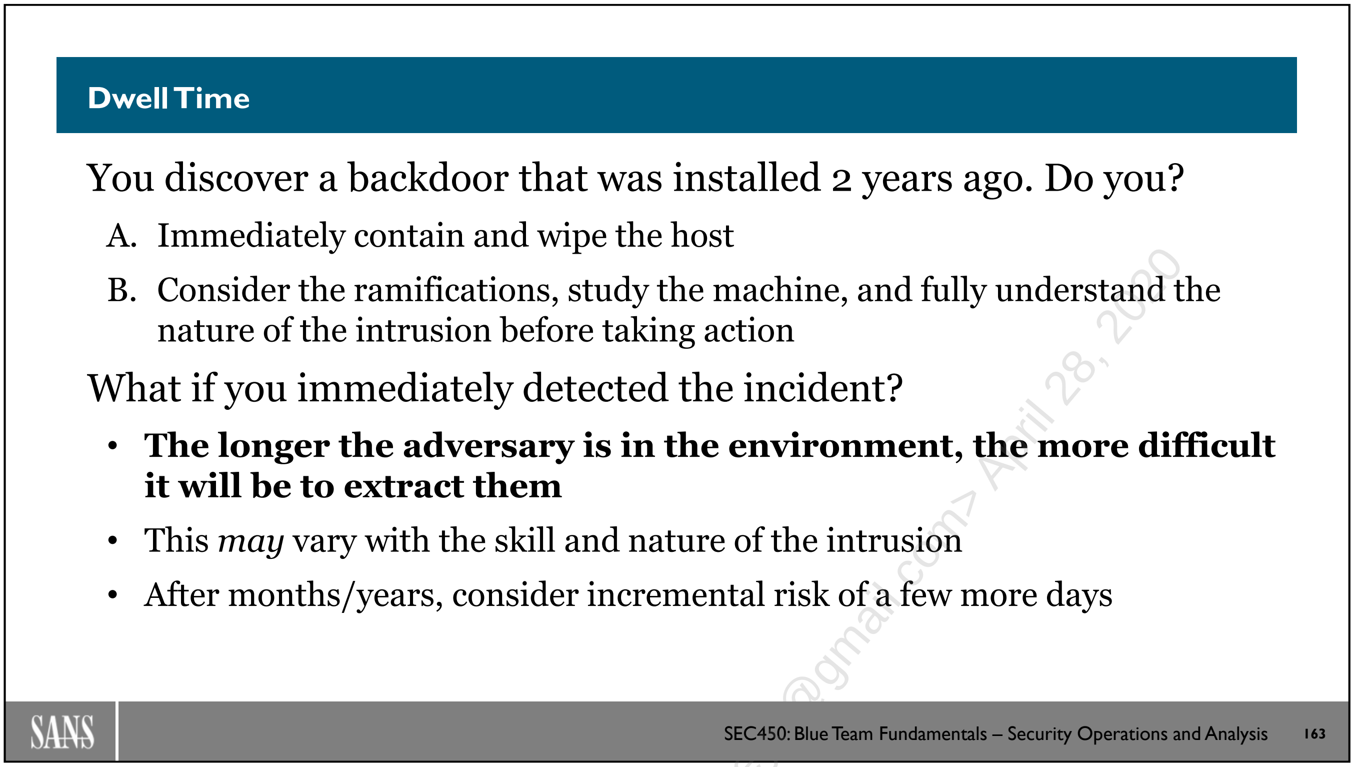
- Researcher Frode Hommedal created useful questions and models to help answer this question.[1]
  - **How long** have they been there?
  - What is the **nature** of the intrusion?
  - What is their **motivation** to access your network?
  - How much **business risk** do they present?
  - Do you **know the attacker's TTPs**?

All these items should factor into your response! (If you know or can find the answers)

Reacting to Intrusion Discovery

When you confirm there is an active incident in progress, what should you do next? Various response styles might be warranted depending on the nature of the attacker and the situation at hand. How can we decide the right path forward? To help answer these questions, Frode Hommedal of the Telenor CERT created a presentation, "Taking the Attacker Eviction Red Pill", where he describes some well-thought-out models and questions that should be asked that can guide us to the answer. The point of the presentation is that incident response decisions can be more complicated than it initially seems with multiple variables factoring in to the appropriate response. Some of the questions he says analysts should ask are on the slide above. In this section, we'll use these questions as a guide to discuss how each may influence your decision on what to do in this scenario.

[1] https://www.slideshare.net/FrodeHommedal/taking-the-attacker-eviction-red-pill



Dwell Time

You discover a backdoor that was installed 2 years ago. Do you?

A. Immediately contain and wipe the host

B. Consider the ramifications, study the machine, and fully understand the nature of the intrusion before taking action

What if you immediately detected the incident?

- **The longer the adversary is in the environment, the more difficult it will be to extract them**
- This *may* vary with the skill and nature of the intrusion
- After months/years, consider incremental risk of a few more days

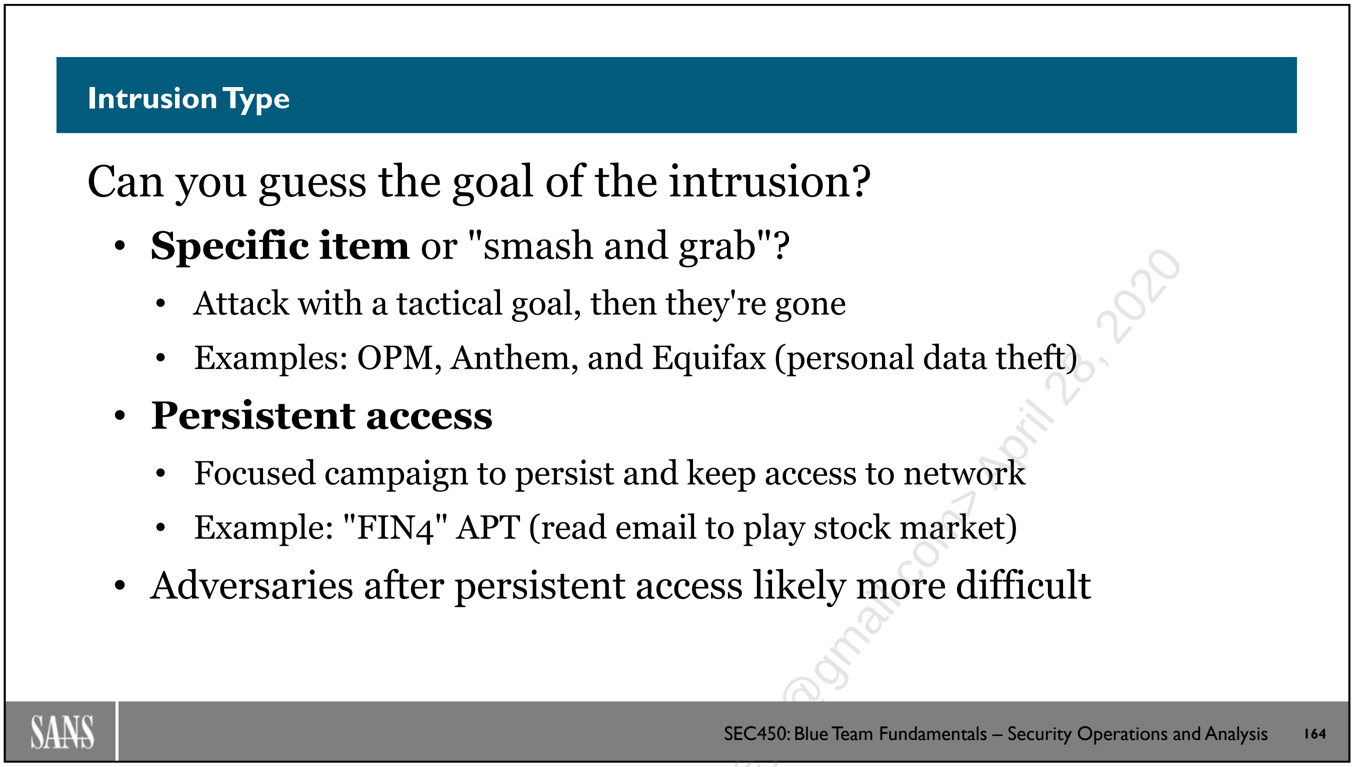
SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 163

Dwell Time

One of the factors you should highly consider when determining the speed of response is the dwell time—how long the adversary has already been in your environment. If you find a piece of malware that seems to have a timestamp of two years ago, you should likely consider a very different response than one that was detected immediately upon download.

Consider the malware that has been active for two years. If this is an APT-style attack, then it is likely the attacker has used this access over the period to fortify their position inside the network. If you immediately jump to remedy the single infection, it's unlikely to be very helpful given the breadth options for access they likely maintain after two years. With a situation like this, it's unlikely the incremental risk of letting the adversary go for a few more days while you gather intelligence will cause catastrophic harm. If you can sense them moving close to something big, you can always take smaller targeted actions that will slow them down without letting on that you know of their presence.



Intrusion Type

Can you guess the goal of the intrusion?

- **Specific item** or "smash and grab"?
  - Attack with a tactical goal, then they're gone
  - Examples: OPM, Anthem, and Equifax (personal data theft)
- **Persistent access**
  - Focused campaign to persist and keep access to network
  - Example: "FIN4" APT (read email to play stock market)
- Adversaries after persistent access likely more difficult

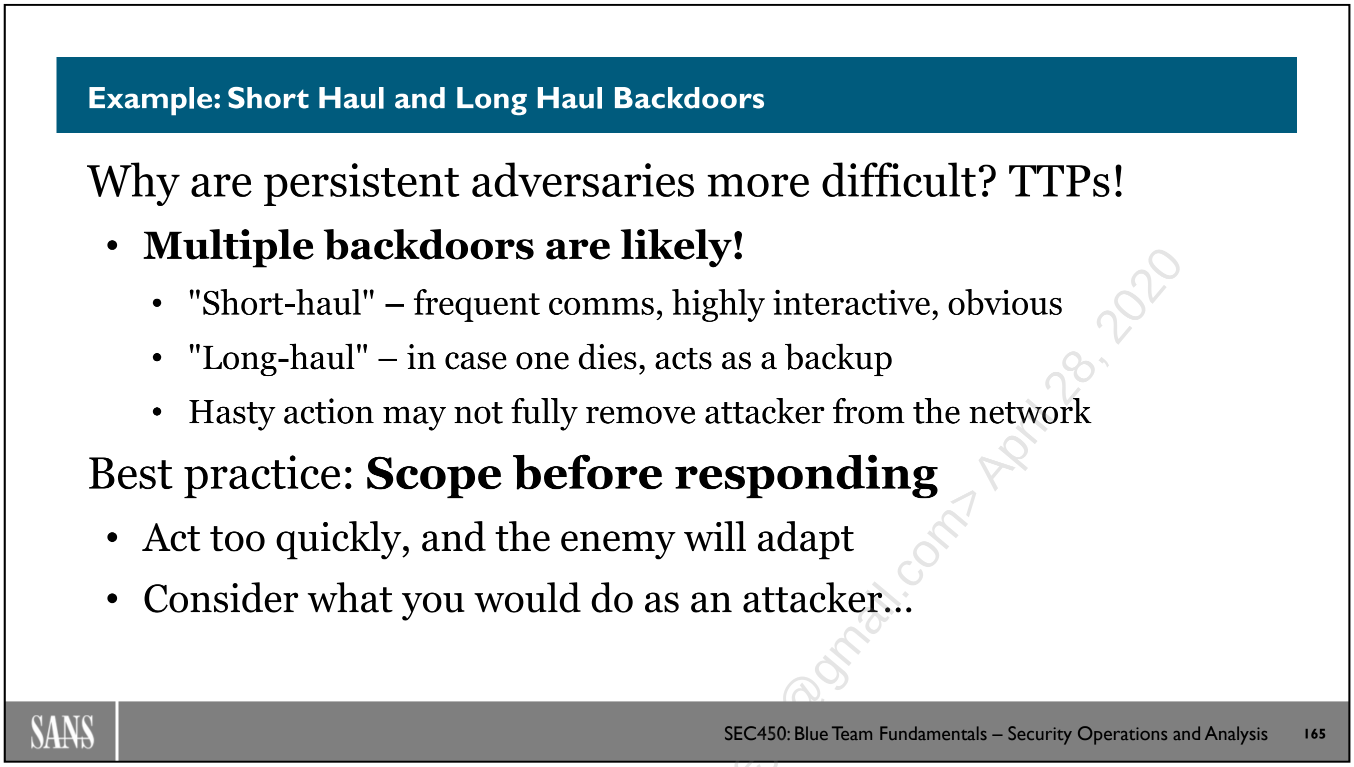
SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 164

Intrusion Type

Another factor for choosing a response will be the intrusion type. If you can tell, is the attacker after a single goal or piece of data such that once they obtain it, they will disappear? On the opposite side of the spectrum, maybe the evidence shows that the compromise is primarily about not being destructive or stealing data, but purely maintaining access to monitor the organization's activity. This type of compromise could potentially be identified if the adversary is stealing email, recording keystrokes over a long period of time, or up to any otherwise long-term monitoring activity.

How does this factor into response speed? If you are up against an adversary that has a goal to maintain long-term access to your environment, it is increasingly likely that the attackers have sunk their roots deep into the organization and getting them out will be a larger challenge. On the other hand, if the attack seems to be more of a smash and grab attempt for a single item, rejecting their attempts several times may frustrate them and turn them elsewhere. Quicker response times might be more appropriate in this scenario.

[1] https://www.slideshare.net/FrodeHommedal/taking-the-attacker-eviction-red-pill



Example: Short Haul and Long Haul Backdoors

Why are persistent adversaries more difficult? TTPs!

- **Multiple backdoors are likely!**
  - "Short-haul" – frequent comms, highly interactive, obvious
  - "Long-haul" – in case one dies, acts as a backup
  - Hasty action may not fully remove attacker from the network

Best practice: **Scope before responding**

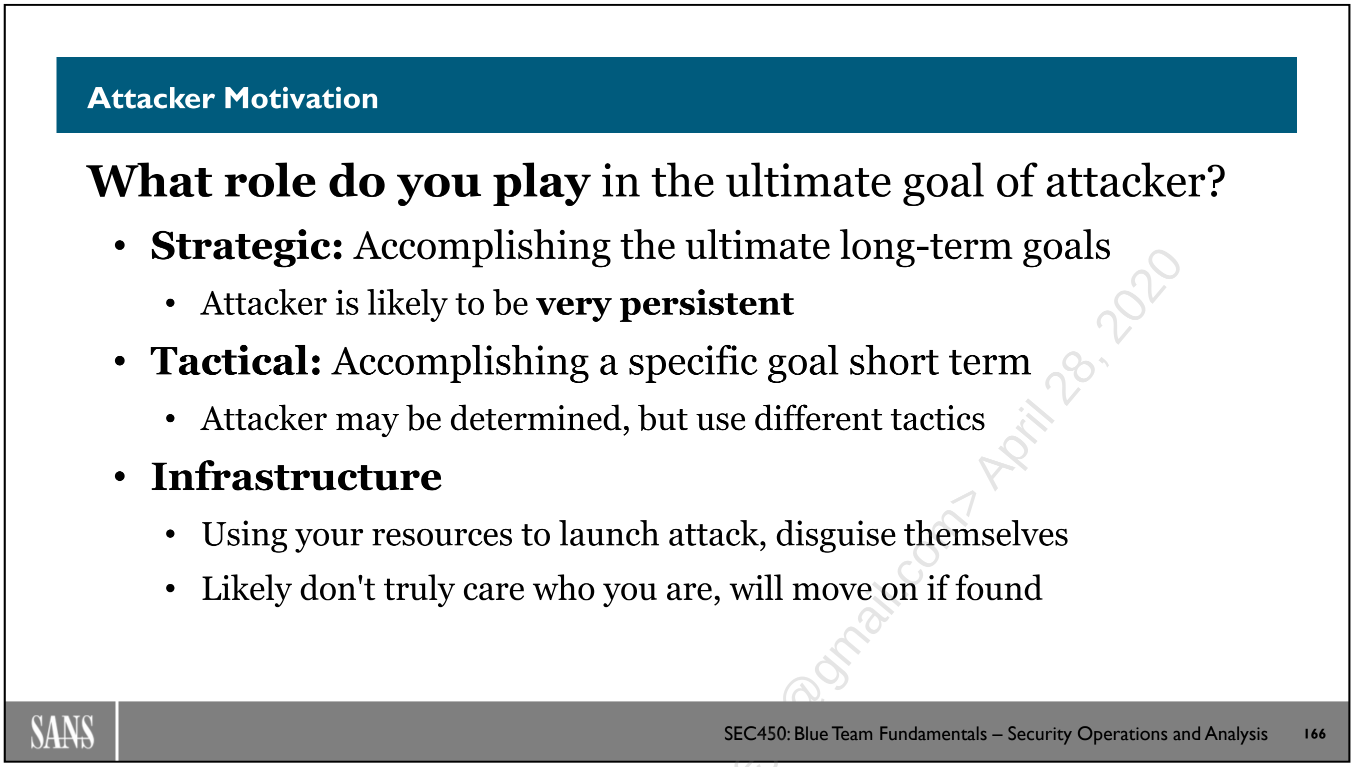
- Act too quickly, and the enemy will adapt
- Consider what you would do as an attacker...

Example: Short Haul and Long Haul Backdoors

Why do long dwell times and persistence-minded adversaries represent a more complicated case to evict from the network? One of the reasons is that the typical "best practice" for attackers of this type is to use multiple, wholly-unrelated backdoors! Consider if you were trying to stay persistent in a network—would you only maintain one point of presence with a single backdoor? Of course not. The more infected devices with unique malware they use, the more likely you are to miss one of them in a remediation, allowing them to let themselves back in.

To accomplish this, these types of backdoors are sometimes split into what's called "short-haul" and "long-haul" backdoors. The short-haul backdoor is the solution they will use day to day. It will facilitate highly interactive communication and may not have stealth as a priority. If you find an attacker's malware, it is most likely to be this type. Once the blue team catches a whiff of the short-haul backdoors and removes all copies of it from the network, the attackers must hope the long-haul malware is not found at the same time. These trojans often use much stealthier and less frequent communication, making them harder to find. As a defender going against an APT style threat, the idea is to strategically cut off their short-haul backdoor capability in such a way that will force them to use the long-haul option, and expose that, too, allowing you to remove all copies of that as well. Getting this right is part of the art and science of blue team.

[1] https://www.first.org/resources/papers/conf2016/FIRST-2016-108.pdf



Attacker Motivation

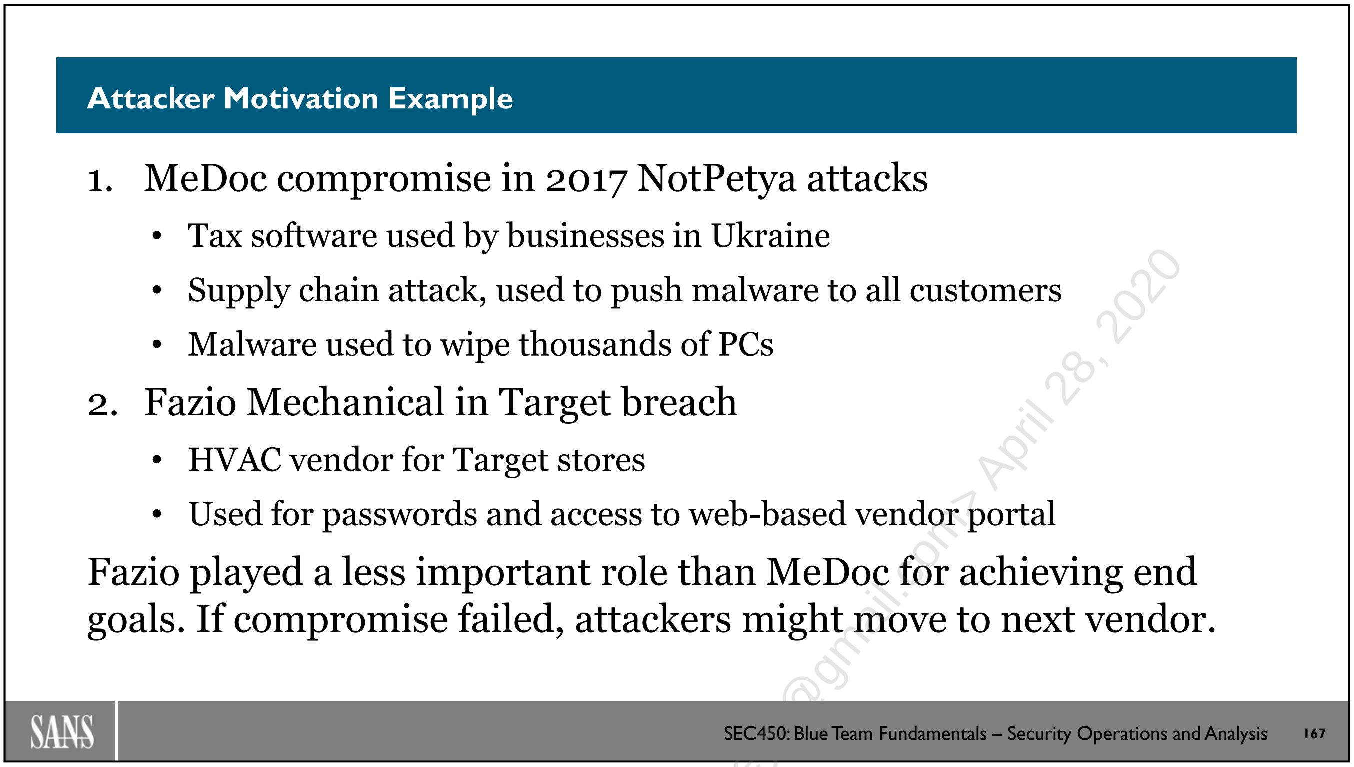
What role do you play in the ultimate goal of attacker?

- **Strategic:** Accomplishing the ultimate long-term goals
  - Attacker is likely to be **very persistent**
- **Tactical:** Accomplishing a specific goal short term
  - Attacker may be determined, but use different tactics
- **Infrastructure**
  - Using your resources to launch attack, disguise themselves
  - Likely don't truly care who you are, will move on if found

Attacker Motivation

A less obvious item to consider is the motivation for the attacker to want access to your network at a high level. This is like the intrusion style discussed earlier, but different in that you are to answer the question of what part you play in their grand scheme, not looking at the specific style of attack used as discussed in the previous section. Consider the options:

- Strategic usage: If you are part of the strategic goals of the attacker, having access to your network is likely part of a crucial long-term objective, and the attacker will throw everything at you to maintain access.
- Tactical objective: Perhaps the adversary needs access to your network, but only for a short period of time to obtain a piece of information, or piggy-back off the access you have to other resources. Once they obtain what they are looking for, they will move on. In this scenario, the attacker may be equally as determined to get into your network, or maybe not, depending on if there are other sources for the same data they can potentially exploit instead. This type of attacker will focus less on persistence and likely target fast execution of their objectives instead.
- Infrastructure / Operational use: Sometimes, an attacker wants control of your systems purely to get access to their target, or to shield themselves from exposure. Bot masters use PCs around the world to send spam, run DNS fast flux infrastructure, and perform other attacks with an organization's resources not because they care who they are, but purely because they are a PC with a CPU and an internet connection. This is like an opportunistic attacker, but you may still be playing a part in an APT campaign—you just don't have a crucial role, or the main target painted on you. You are a means to an end. If an attacker is using your infrastructure to facilitate compromise, it is likely much easier to frustrate them with simple, quick PC rebuild that will disconnect them and force them to move on to another option that may be more reliable.



Attacker Motivation Example

1. MeDoc compromise in 2017 NotPetya attacks
   - Tax software used by businesses in Ukraine
   - Supply chain attack, used to push malware to all customers
   - Malware used to wipe thousands of PCs
2. Fazio Mechanical in Target breach
   - HVAC vendor for Target stores
   - Used for passwords and access to web-based vendor portal

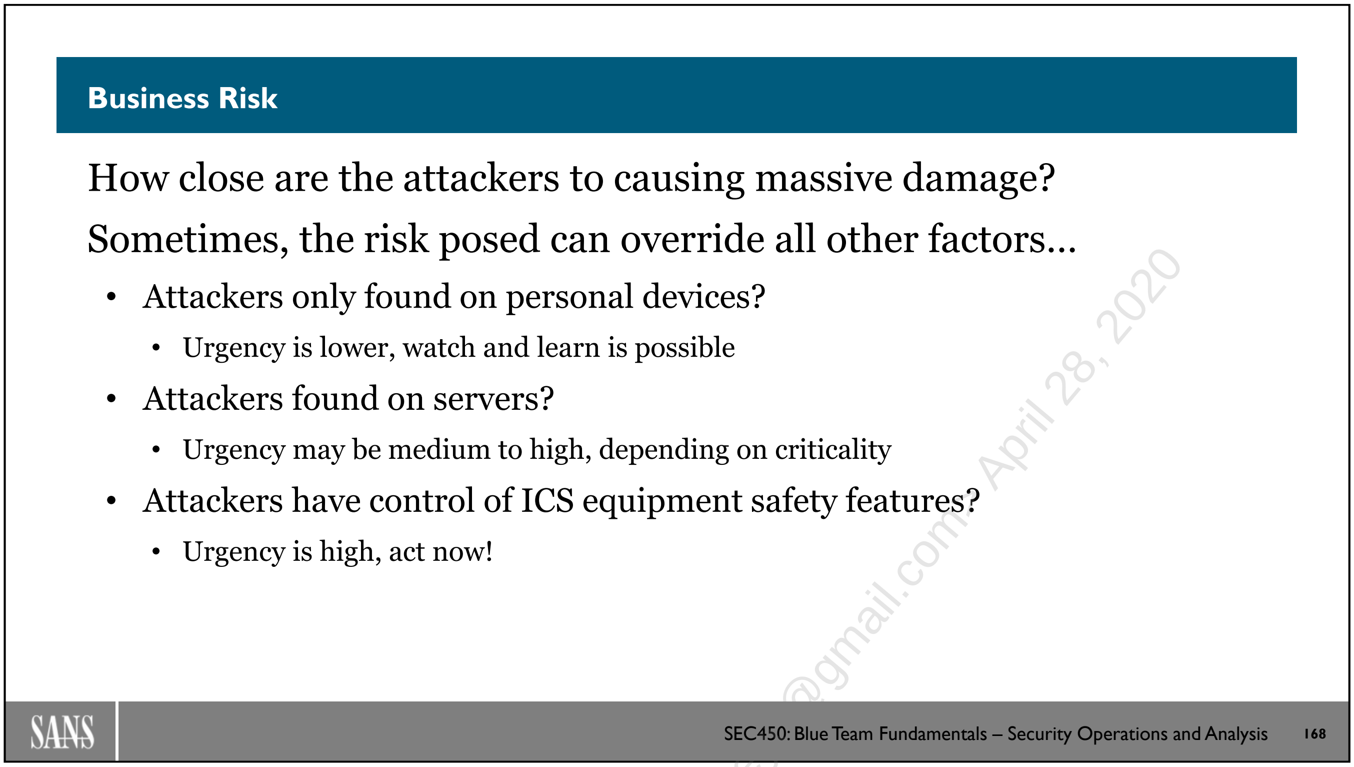
Fazio played a less important role than MeDoc for achieving end goals. If compromise failed, attackers might move to next vendor.

Attacker Motivation Example

To give an example of these situations, consider the role that the MeDoc (Ukrainian tax) software played in the NotPetya compromise of 2017 vs. the role that Fazio Mechanical, an HVAC vendor, played in the Target breach.

In the case of NotPetya, attackers were seemingly after Ukrainian companies and anyone who did business within the country and wanted to send that message via destructive malware. To accomplish this goal, they breached Intellect Service, the company that makes the software and custom compiled malware into a version of MeDoc that was later passed on to victims from the company's servers. The role that MeDoc and Intellect Service played in this breach was mostly a strategic one such that they were the main conduit for compromising the victims and controlling the backdoor. Had Intellect Service found the breach before NotPetya was activated, it's highly likely the attackers would have done everything in their power to regain their position of power within the network to continue with the mission.

In the Target case, Fazio Mechanical was breached due to the attacker's interest in their access to an externalized Target vendor portal. The attackers purportedly found Fazio's name through a list of vendors that Target had publicly available and, therefore, their specific participation in the breach may have not been all that important since in their absence, another vendor could have been leveraged in the same way. In this case, Fazio held a tactical or even operational-level role because had they fended off the initial attack, it is no doubt that the attackers might have moved on to the next name on the list. In this case, had Fazio done incident response, a swift removal of the attackers from the network might have been enough to take them out of the path used to ultimately compromise Target's systems.



Business Risk

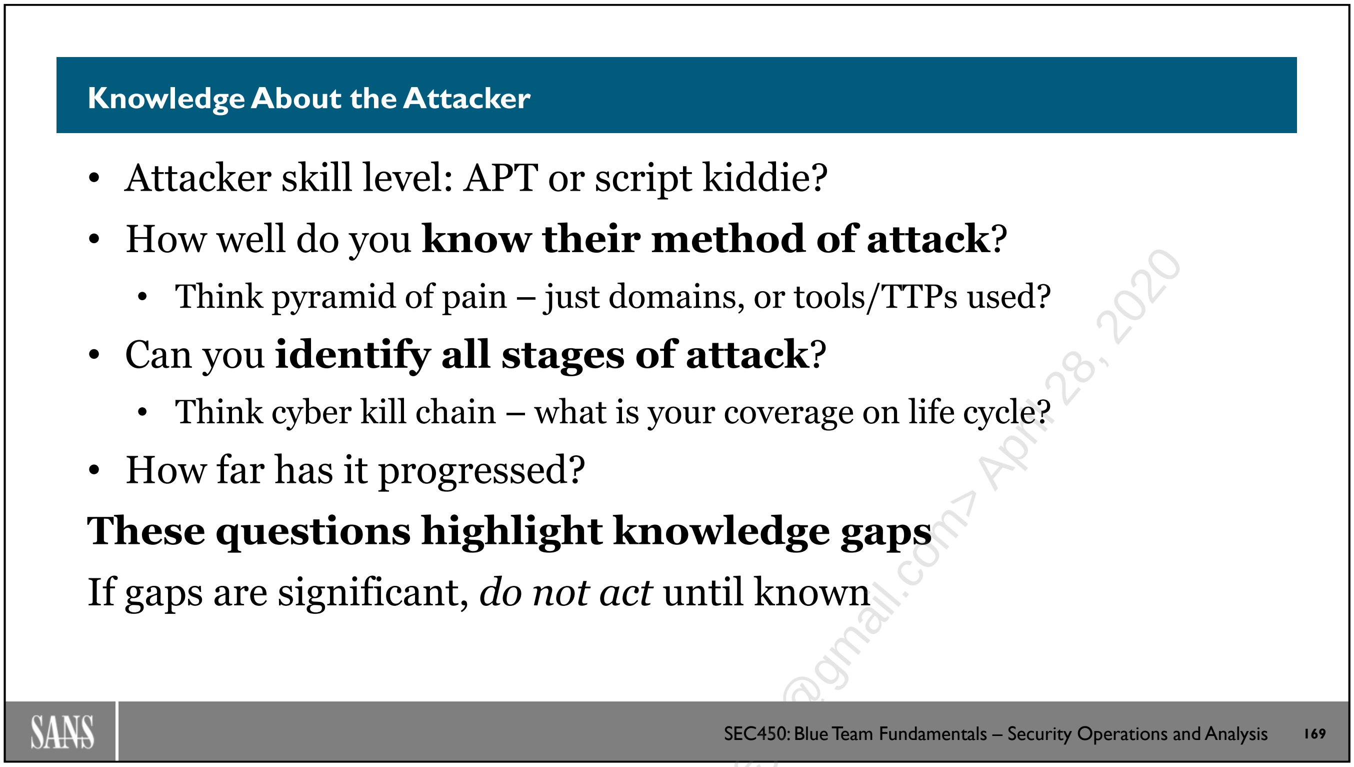
How close are the attackers to causing massive damage?

Sometimes, the risk posed can override all other factors…

- Attackers only found on personal devices?
  - Urgency is lower, watch and learn is possible
- Attackers found on servers?
  - Urgency may be medium to high, depending on criticality
- Attackers have control of ICS equipment safety features?
  - Urgency is high, act now!

Business Risk

There is one factor that can pull rank on all others, and that is the business risk posed by the situation. Regardless of the other factors, the position the attackers have established inside the network may singularly drive the response style if the situation is critical enough. Attackers that have taken over ICS safety equipment, for example, might cause an immediate threat to human life and those systems must be put on pause immediately, even if the consequences are an increase in remediation complication. By comparison, attackers that have only achieved a level of compromise affecting individual users' desktops may afford the blue team more time to craft a complete response compared to that of a compromise affecting servers.



Knowledge About the Attacker

- Attacker skill level: APT or script kiddie?
- How well do you **know their method of attack**?
  - Think pyramid of pain – just domains, or tools/TTPs used?
- Can you **identify all stages of attack**?
  - Think cyber kill chain – what is your coverage on life cycle?
- How far has it progressed?

These questions highlight knowledge gaps

If gaps are significant, *do not act* until known

SANS

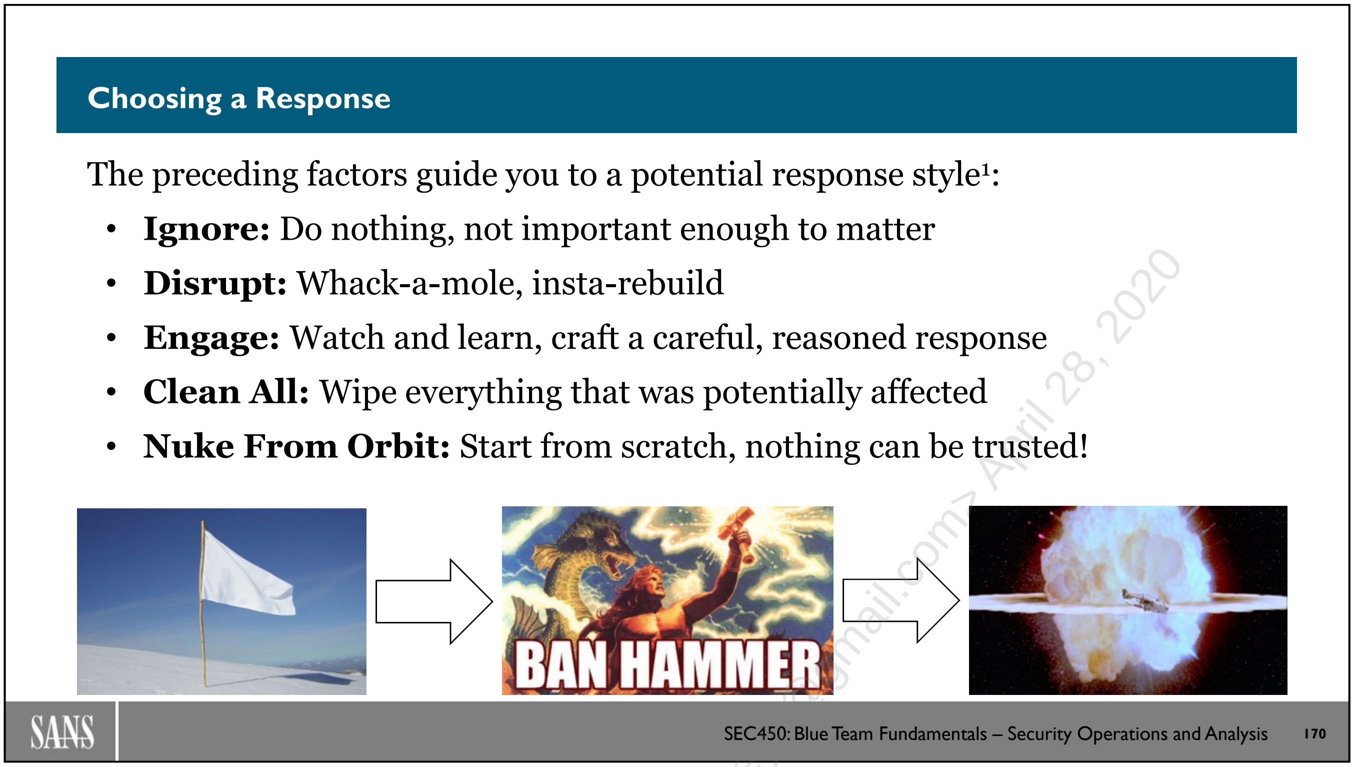
SEC450: Blue Team Fundamentals – Security Operations and Analysis 169

Knowledge About the Attacker

A final item to consider is your knowledge of the attackers' TTPs across the kill chain and whether you are instrumented to detect them or not. Frode Hommedal lays this information out on what he calls the Cyber Threat Intelligence Matrix (CTIM) and has a whole presentation purely on this idea.[1] The summary is that you should consider both the kill chain stages and the ideas behind the pyramid of pain as well as the similar "detection maturity level" idea laid out by Ryan Stillions.[2] You map your knowledge of the attacker against your defensive capability on a matrix with axis of attack life cycle stages and whether you know information about the adversaries high on the pyramid of pain like TTPs, or just low-level information like IP addresses and hashes. Doing so will highlight your knowledge gaps, and the presence of too many knowledge gaps means you are not yet ready to attempt advanced attacker eviction.

[1] https://www.slideshare.net/FrodeHommedal/the-cyber-threat-intelligence-matrix

[2] http://ryanstillions.blogspot.com/2014/04/the-dml-model_21.html



Choosing a Response

The preceding factors guide you to a potential response style[1]:

- **Ignore:** Do nothing, not important enough to matter
- **Disrupt:** Whack-a-mole, insta-rebuild
- **Engage:** Watch and learn, craft a careful, reasoned response
- **Clean All:** Wipe everything that was potentially affected
- **Nuke From Orbit:** Start from scratch, nothing can be trusted!

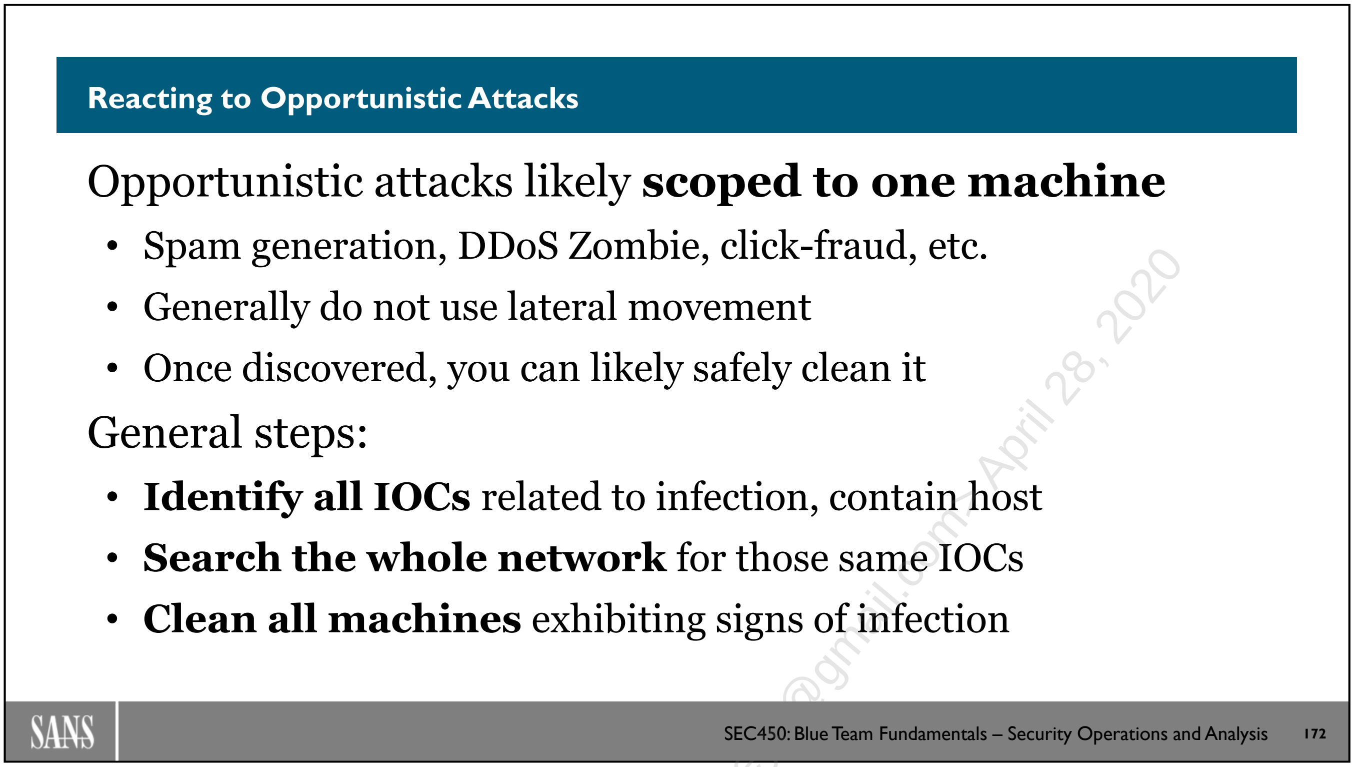
Choosing a Response

After the evidence has revealed an intrusion of some sort and you have considered the preceding factors, what should be your next step? Consider the above range of responses you could have. We have based our evaluation discussion around the factors identified in Frode Hommedal's presentation on attacker eviction.[1] That same presentation also brings what he calls "response patterns" that form a spectrum from "do nothing" to "start over from scratch", and uses the considerations previously discussed to help guide us to an option. The spectrum breaks down into the following high-level categories, some of which could be effective against single opportunistic attacks, and others that are more appropriate for advanced attacks that have been present in the environment for a long time.

- Ignore: Some opportunistic infections such as adware or "potentially unwanted programs" may post so little threat that it might not be worth your time to address them at all.
- Disrupt: Opportunistic threats can often be handled in this way—an immediate rebuild of the infected asset will take care of the program since these are not intrusions that use lateral movement and spread out throughout the environment.
- Engage: This is the "watch and learn" option. You leave the hosts online for a bit, investigate indicators on that machine and others and see how the whole campaign is comprised, what tools it uses and its method of communication. Once you understand the attack deeply, you move to perform an in-depth sweep to cut off the infection. In the meantime, it is advised that, if possible without tipping your hand to the adversary, you should tactically guard against further damage and incident escalation.
- Clean all: This option uses less watch and learn and moves to a mass cleaning and rebuild of all potentially affected assets. It is a more heavy-handed approach that uses less skill on the defensive side as TTPs don't need to be learned since everything is being rebuilt.
- Complete wipe: This option is an extreme only reserved for the worst of compromises or business disruption attacks. This is the option likely to have been used (or forced upon organizations) after large-scale issues such as the Sony breach, or at Merck/Maersk after NotPetya destroyed most of the machines in the environment.

Note that there is a spectrum of effectiveness, cost, and skill required of the team that varies across these options as well. The cheapest option is to ignore while the most expensive, at least in direct costs, will be the full environment rebuild. The least effective option is obviously at the top, and effectiveness at removing the adversary from the environment should increase moving down the list to the final items which ends with a totally new environment. The "easiest" options in terms of InfoSec skill are at the top and bottom of the list. Rebuilding machines or doing nothing don't take much from the blue team while engaging in battle with the enemy. The middle option takes the most advanced capabilities from the SOC.

[1] https://www.slideshare.net/FrodeHommedal/taking-the-attacker-eviction-red-pill



Reacting to Opportunistic Attacks

Opportunistic attacks likely **scoped to one machine**

- Spam generation, DDoS Zombie, click-fraud, etc.
- Generally do not use lateral movement
- Once discovered, you can likely safely clean it

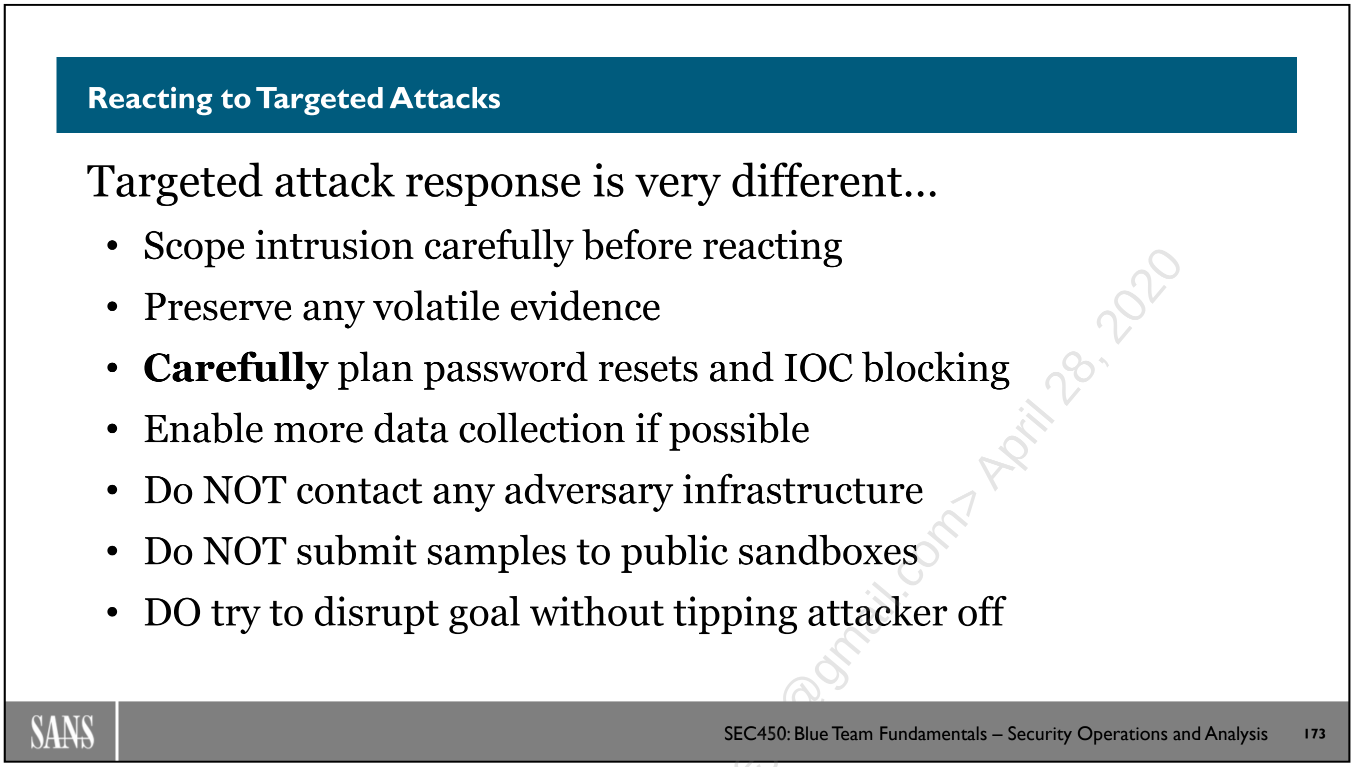
General steps:

- **Identify all IOCs** related to infection, contain host
- **Search the whole network** for those same IOCs
- **Clean all machines** exhibiting signs of infection

Reacting to Opportunistic Attacks

With these options in mind, when it comes to opportunistic attacks, how should we react? Since opportunistic attacks are generally single host-centric pieces of malware not designed to spread out across the network and give the attacker interactive command and control, these infections are relatively easy to deal with. In the case of something like a spam bot infection, click fraud malware, or even a banking trojan, it is likely that once that single machine is cleaned, the incident is finished. These types generally have no facilities to continue to spread on the inside of a network and, therefore, excising them from an organization is just a case of doing a thorough job on a single machine.

Generally, the steps are to identify the infection and contain the host so that the user cannot lose any sensitive data. Afterwards, the machine can either be reimaged, which is the safest way to remove malware, or AV/manual methods could be used to remove the infection. The infection's indicators should then be collected in either case under the assumption that if one machine picked it up, there could be other machines in the network with the issue as well. Take all domains and IP addresses the machine contacts, as well as any files it is known to drop and use any facilities available to search for these indicators anywhere else on the network. The idea is once you find a single infection, you should assume there could be more and leverage what is learned to ensure that *all* machines do not have the same issue. This should be done even though the single identified infection likely has nothing to do with any other copies of the malware inside the environment.



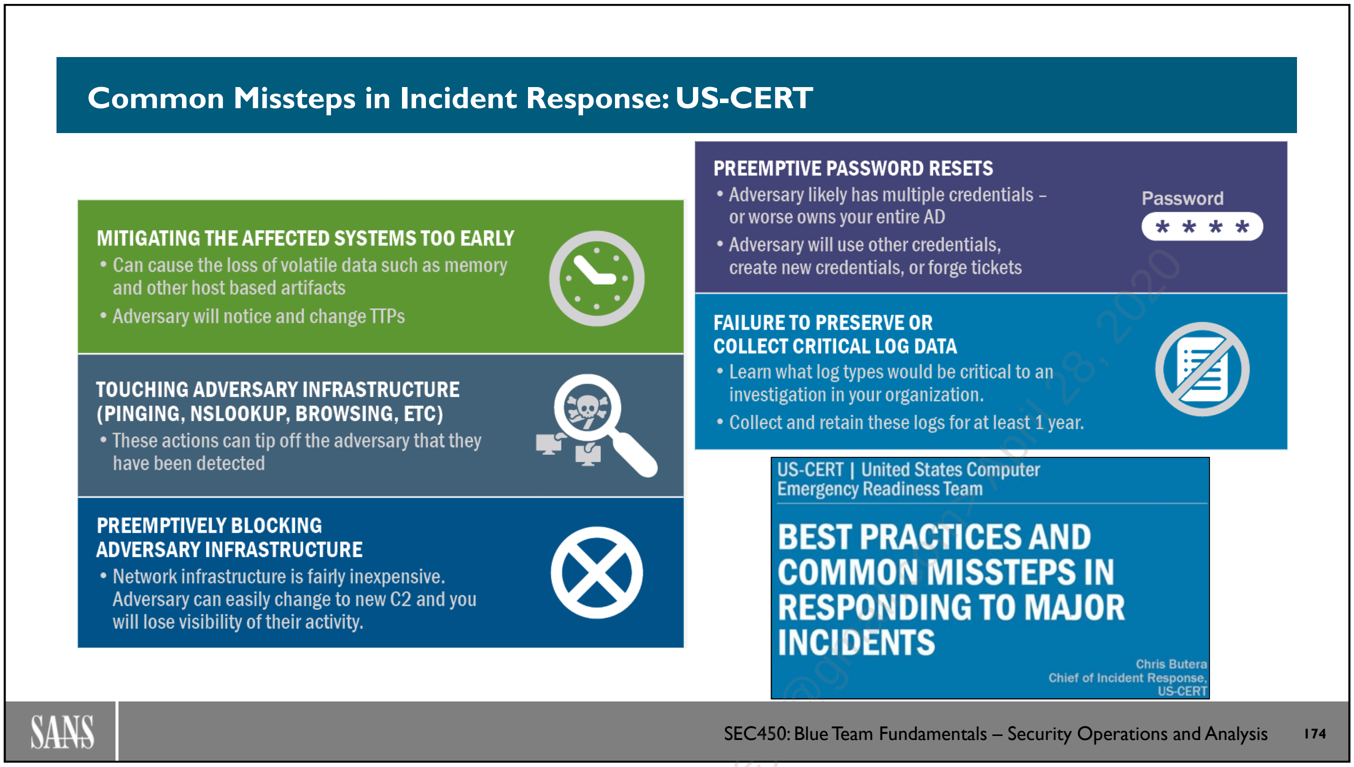
Reacting to Targeted Attacks

Targeted attack response is very different…

- Scope intrusion carefully before reacting
- Preserve any volatile evidence
- **Carefully** plan password resets and IOC blocking
- Enable more data collection if possible
- Do NOT contact any adversary infrastructure
- Do NOT submit samples to public sandboxes
- DO try to disrupt goal without tipping attacker off

Reacting to Targeted Attacks

We now know that targeted attacks are a different situation altogether. The goals of targeted attackers and the techniques they use to accomplish them drive the need to react very differently than an opportunistic attack. After an initial determination shows that a targeted attack is possible, the above guidance should be followed to ensure your reaction is appropriately considered and does not expose any unnecessary information to the adversary.



Common Missteps in Incident Response: US-CERT

MITIGATING THE AFFECTED SYSTEMS TOO EARLY

- Can cause the loss of volatile data such as memory and other host based artifacts
- Adversary will notice and change TTPs

TOUCHING ADVERSARY INFRASTRUCTURE (PINGING, NSLOOKUP, BROWSING, ETC)

- These actions can tip off the adversary that they have been detected

PREEMPTIVELY BLOCKING ADVERSARY INFRASTRUCTURE

- Network infrastructure is fairly inexpensive. Adversary can easily change to new C2 and you will lose visibility of their activity.

PREEMPTIVE PASSWORD RESETS

- Adversary likely has multiple credentials – or worse owns your entire AD
- Adversary will use other credentials, create new credentials, or forge tickets

FAILURE TO PRESERVE OR COLLECT CRITICAL LOG DATA

- Learn what log types would be critical to an investigation in your organization.
- Collect and retain these logs for at least 1 year.

US-CERT | United States Computer Emergency Readiness Team

BEST PRACTICES AND COMMON MISSTEPS IN RESPONDING TO MAJOR INCIDENTS

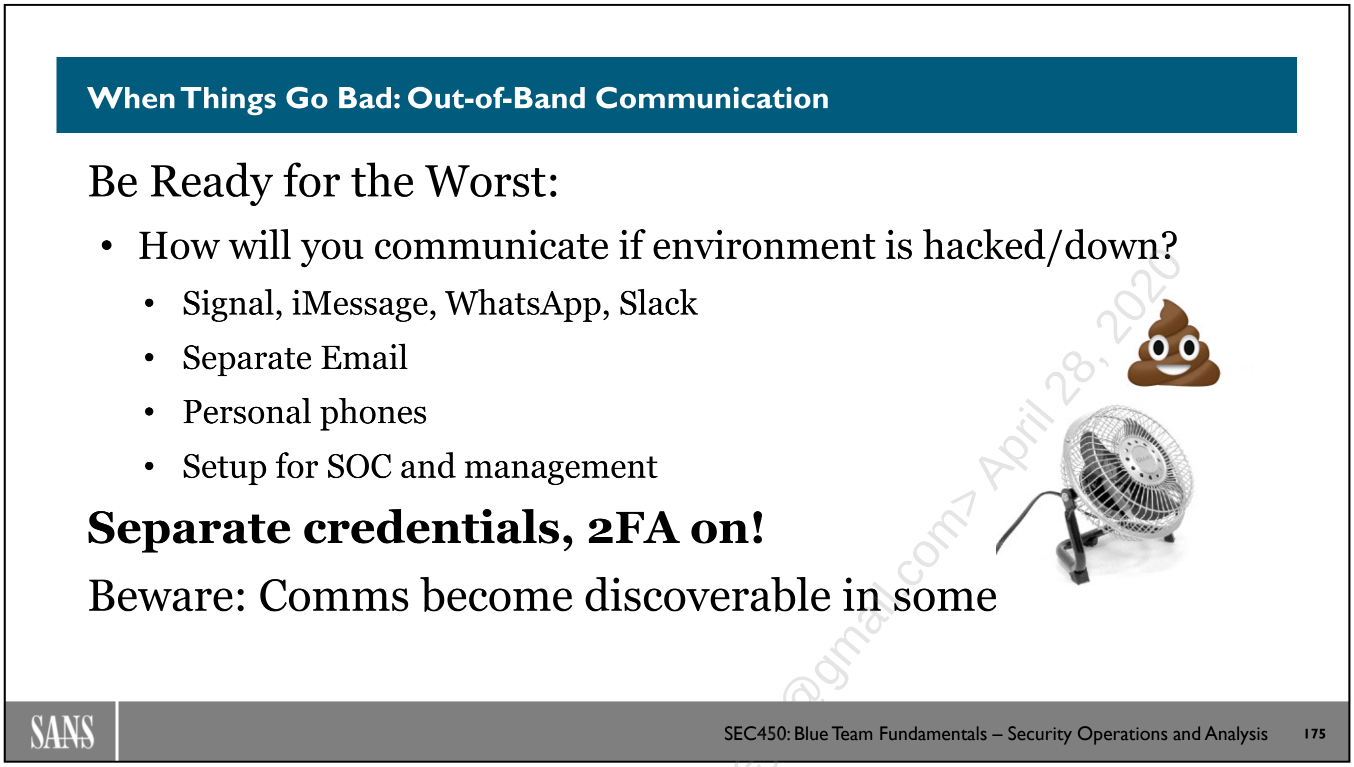
Chris Butera
Chief of Incident Response,
US-CERT

Common Missteps in Incident Response: US-CERT

Is the watch and learn method truly considered best practice? Yes! US-CERT agrees with the risk of reacting too fast. In the presentation "Best Practices and Common Missteps in Responding to Major Incident" the items above are called out as common mistakes. Notice the top item is mitigating systems too early!

Read through the rest of the advice. Targeted attacks require special, careful handling and resisting the instinct to immediately react. Later in the class, we will touch on some of these items such as OPSEC for analysis and when to apply enemy infrastructure blocking.

[1] https://www.first.org/resources/papers/conf2016/FIRST-2016-108.pdf



When Things Go Bad: Out-of-Band Communication

Be Ready for the Worst:

- How will you communicate if environment is hacked/down?
  - Signal, iMessage, WhatsApp, Slack
  - Separate Email
  - Personal phones
  - Setup for SOC and management

Separate credentials, 2FA on!

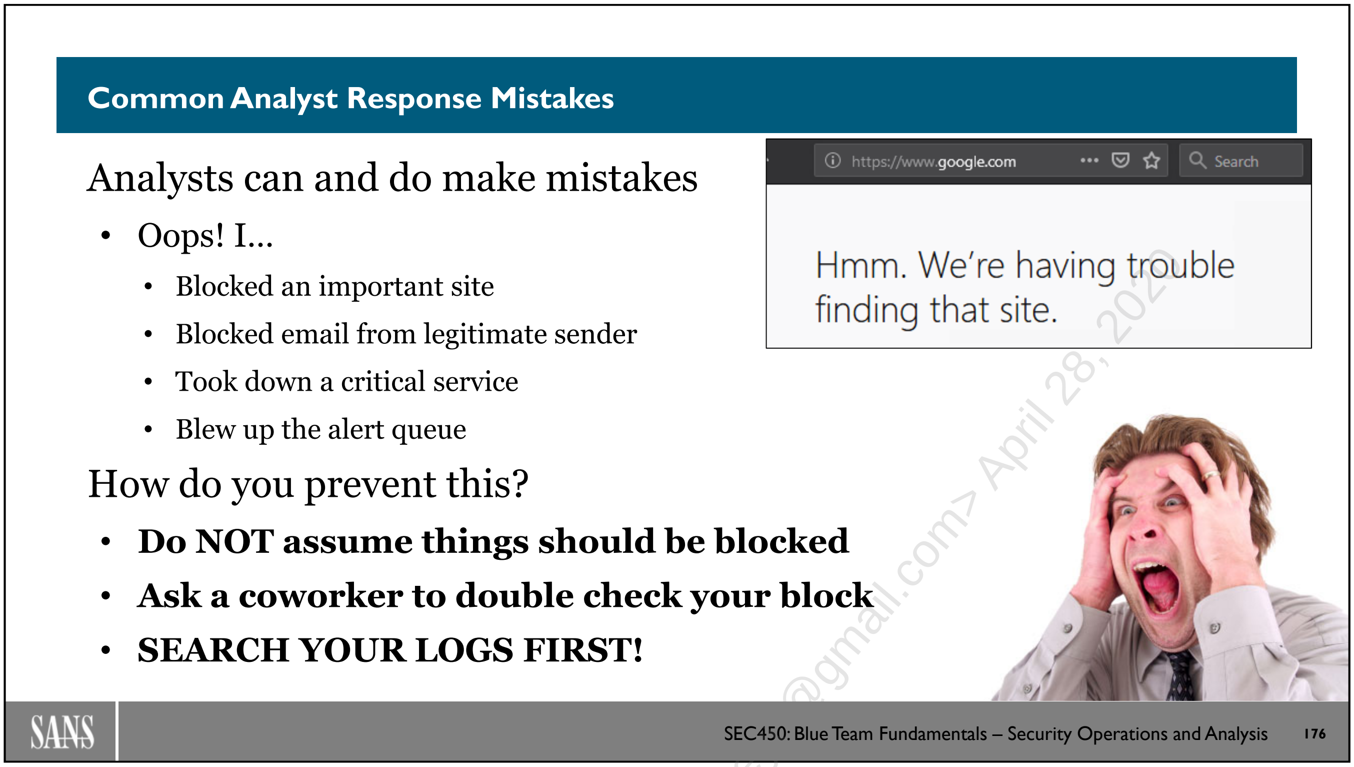
Beware: Comms become discoverable in some

SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 175

When Things Go Bad: Out-of-Band Communication

One of the strategies repeatedly employed by advanced attackers is monitoring the security team communications. If they can get access to SOC team member's communications and emails, they can stay permanently one step ahead of the recovery to ensure they always know what will come next from the defense. Though you may think this is unlikely, it is not. There are numerous reports of such activity across various incidents and industry sectors and is an obvious choice for attackers that wish to remain in the environment. All it takes to do this in many cases is access to a desktop administrative account such as the help desk or a domain admin. Since these types of accounts typically have privileges to log into all machines, the security teams' machines can easily be compromised as well. This is one of the arguments for keeping the security team infrastructure completely separate from the Windows domain. If there are no accounts in common, this attack becomes much more difficult.

With any incident that becomes bad enough (such as domain admin compromise) it may be advisable to assume your in-network communications are being monitored, regardless of whether you have seen direct evidence to support the theory or not. To solve this problem, every SOC should set up an out-of-band communication method pre-incident and ensure that it works. It could be something complicated like a separate email infrastructure or even something simple like having everyone install Signal on their mobile devices and starting a group chat (which is something that can be useful for out-of-hours team communication as well). The key point here is that it is 100% non-reliant on the organization's credentials, accounts, or infrastructure. That way, even in the case of a total network meltdown, team members can still stay in secure communication with each other. Methods for out-of-band communication should be established with both SOC team members, as well as management because they will certainly be interested in keeping up to date on any large-scale incident and will have their own coordination to do during that time as well. And as always, don't forget that multifactor authentication should be enabled for signing in to your out-of-band communication method. You don’t want attackers breaking their way into that as well.



Common Analyst Response Mistakes

Analysts can and do make mistakes

- Oops! I…
  - Blocked an important site
  - Blocked email from legitimate sender
  - Took down a critical service
  - Blew up the alert queue

How do you prevent this?

- **Do NOT assume things should be blocked**
- **Ask a coworker to double check your block**
- **SEARCH YOUR LOGS FIRST!**

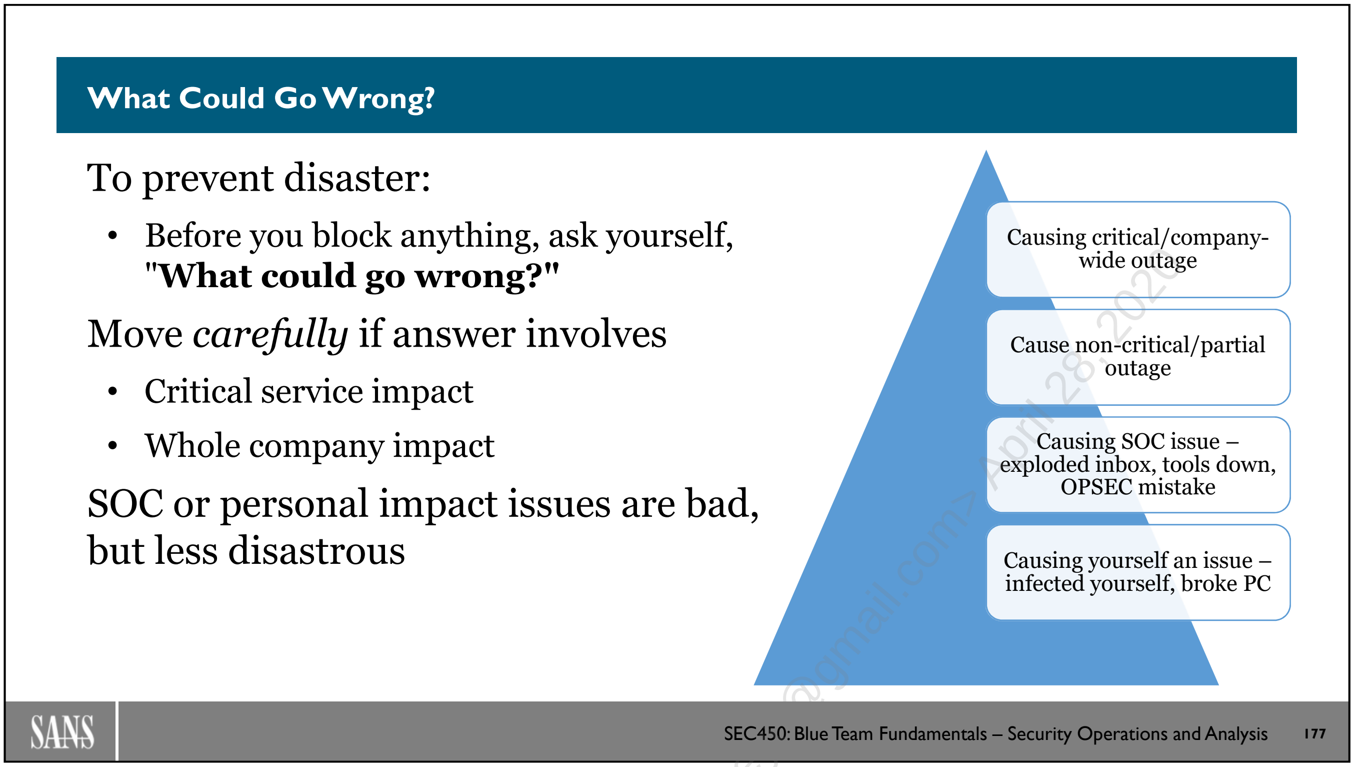
https://www.google.com
Search

Hmm. We're having trouble finding that site.

Common Analyst Response Mistakes

One of the most common mistakes that analysts can make is over-zealously applying a block for an indicator they found inside an alert. It's honestly a very easy thing to do given the power that many analysts must apply controls and the volume of indicators they see every day. While many of the problems come from prevention rules that can truly cause denial of service conditions, even detection rules can cause SOC issues such as an exploded alert queue.

What is the best answer to this issue? First, do not jump to conclusions that blocks on domain names, IP addresses, or hashes are the best idea. In *some* cases, this will certainly be true, but before you do anything that can take down a service, do a double check. Get an alert for legitbusinesspartner.com? Before you trust the alert and jump to the conclusion that it's trying to send attacks, do a search in your SIEM for any logs to this domain going back a few days or more. Do you see traffic to this domain in any other capacity? If there is constant traffic to it from all computers in the network, you can guess that it likely *is* a real partner organization and that blocking it would be a very bad idea. If you can go back a week and see that not a single item in the environment has ever spoken to that domain, (assuming you have the full logs), you now know that you are likely safe in blocking the domain and that it won't cause any significant impact. This same logic goes for hash files, email domains and anything else. As a general rule, before you apply blocks or any changes to detection rules, run a retroactive search to see what the impact may be.



What Could Go Wrong?

To prevent disaster:

- Before you block anything, ask yourself, "**What could go wrong?**"

Move *carefully* if answer involves

- Critical service impact
- Whole company impact

SOC or personal impact issues are bad, but less disastrous

Causing critical/company-wide outage

Cause non-critical/partial outage

Causing SOC issue – exploded inbox, tools down, OPSEC mistake

Causing yourself an issue – infected yourself, broke PC

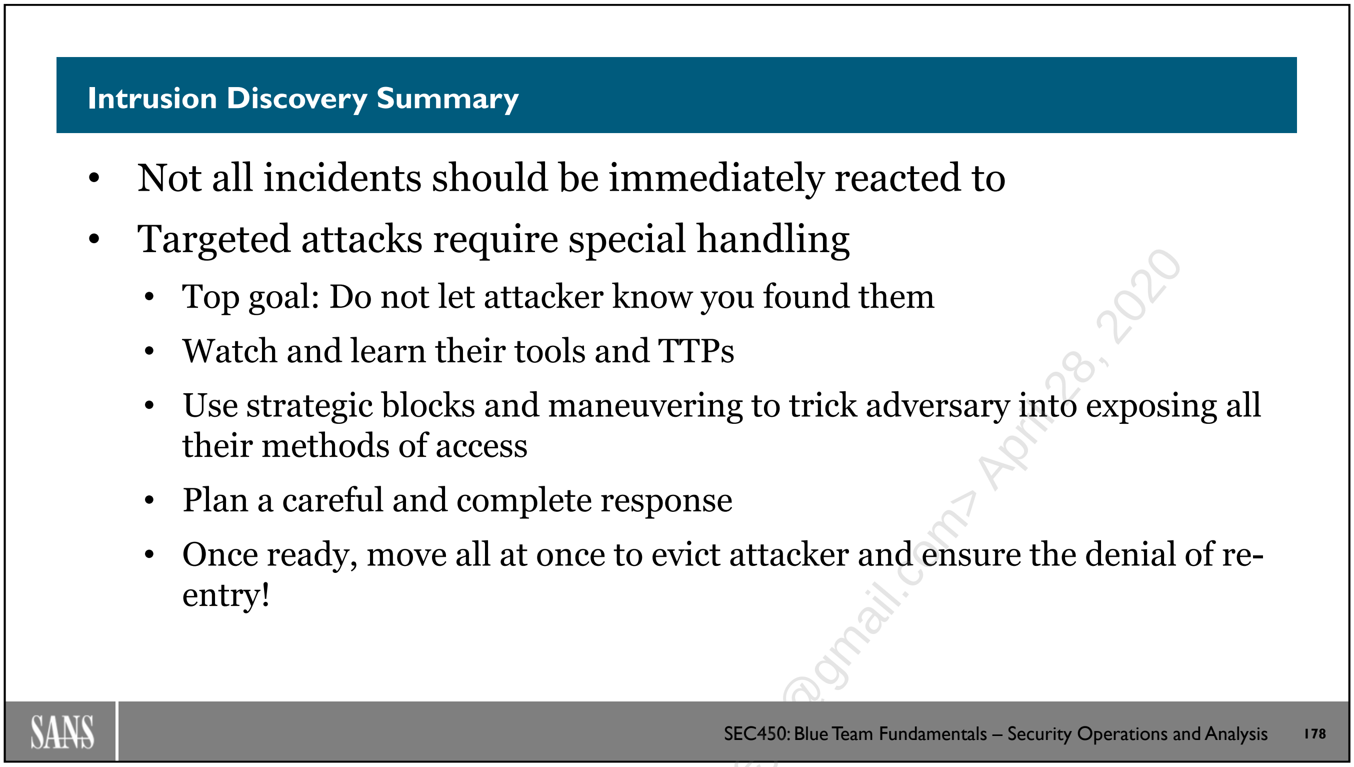
SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 177

What Could Go Wrong?

Even being careful, you will still make a mistake sometime in your career. The question is, how bad will it be? As with assessing risk, consider the range of possible mistake consequences and consider carefully what the worst apparent possible impact could be if you block something. If the alert has something to do with a machine running a critical service, it might be best to get buy-in from the business owner and SOC manager before taking any action. Don't freak out if you make a mistake. Just own up to it and remember you have just learned a tough lesson, but don't make it worse by making excuses. Everyone messes up at some point, and as they say, "failure is the best teacher." Expensive lessons are the best way to ensure you never make that mistake again. ☺

In the grand scheme of mistakes, most will not impact the whole company in a critical way, or even part of it. Many mistakes only involve the people in the SOC, and the tools used by them. A particularly inefficient SIEM search could lock it up, new threat data entered into an IDS could blow up the alert queue, you could infect yourself or others with a sample you accidentally ran. The best way to mitigate these kinds of mistakes is to make sure the recovery procedure for the appliance is available to everyone in case it happens off-hours. Of course, most mistakes won't be that serious and even the ones that are will very seldom result in lasting consequences for the analyst.



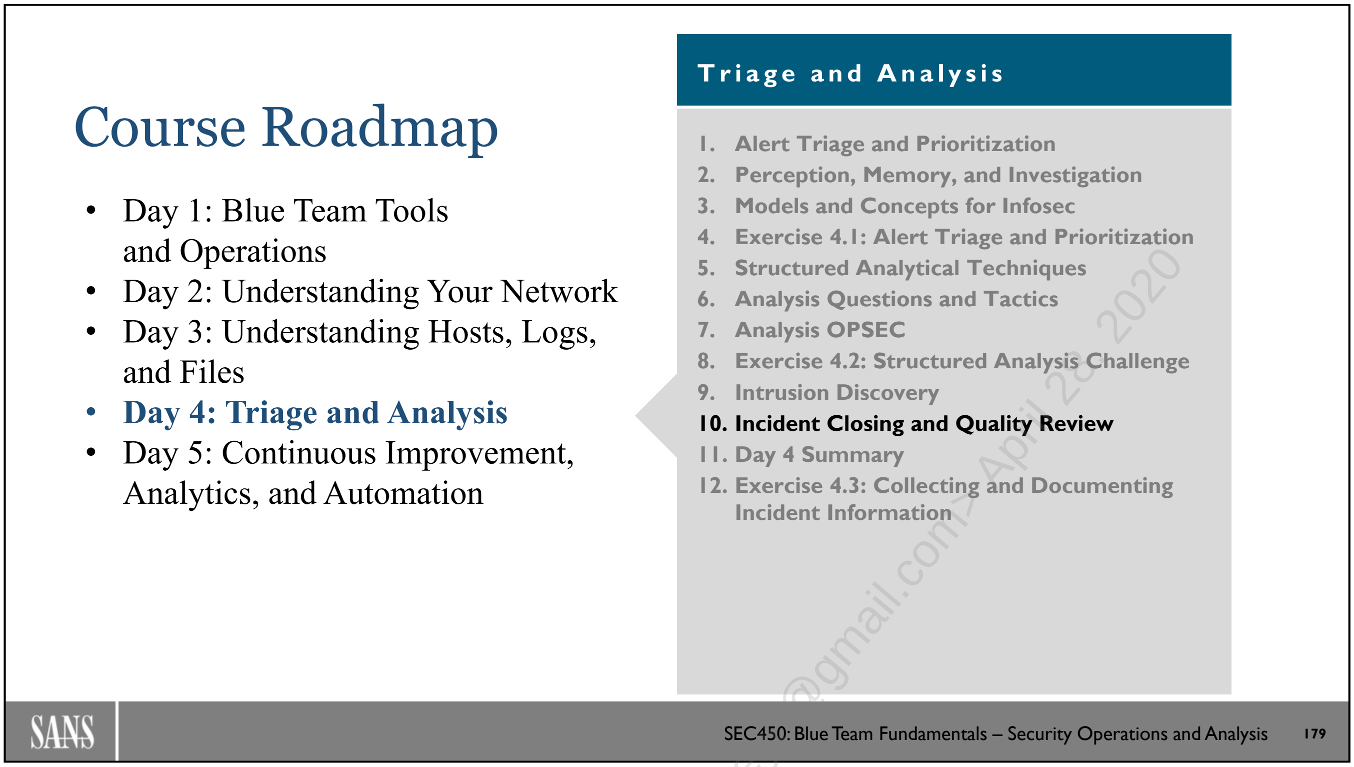
Intrusion Discovery Summary

- Not all incidents should be immediately reacted to
- Targeted attacks require special handling
  - Top goal: Do not let attacker know you found them
  - Watch and learn their tools and TTPs
  - Use strategic blocks and maneuvering to trick adversary into exposing all their methods of access
  - Plan a careful and complete response
  - Once ready, move all at once to evict attacker and ensure the denial of re-entry!

SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 178

Intrusion Discovery Summary

In this module, we discussed the options for incident response and how you can decide when to perform immediate action vs. when you should sit and wait on a more crafted eviction plan. Ultimately, this decision comes down to multiple factors and there is no single question that can answer which approach should be taken. This is a complicated game with no definite answers. There will always be a level uncertainty and, as defenders, the best thing we can do is use the structured models laid out for us to try to understand and reduce that uncertainty and ensure we are in position to make a complete response.



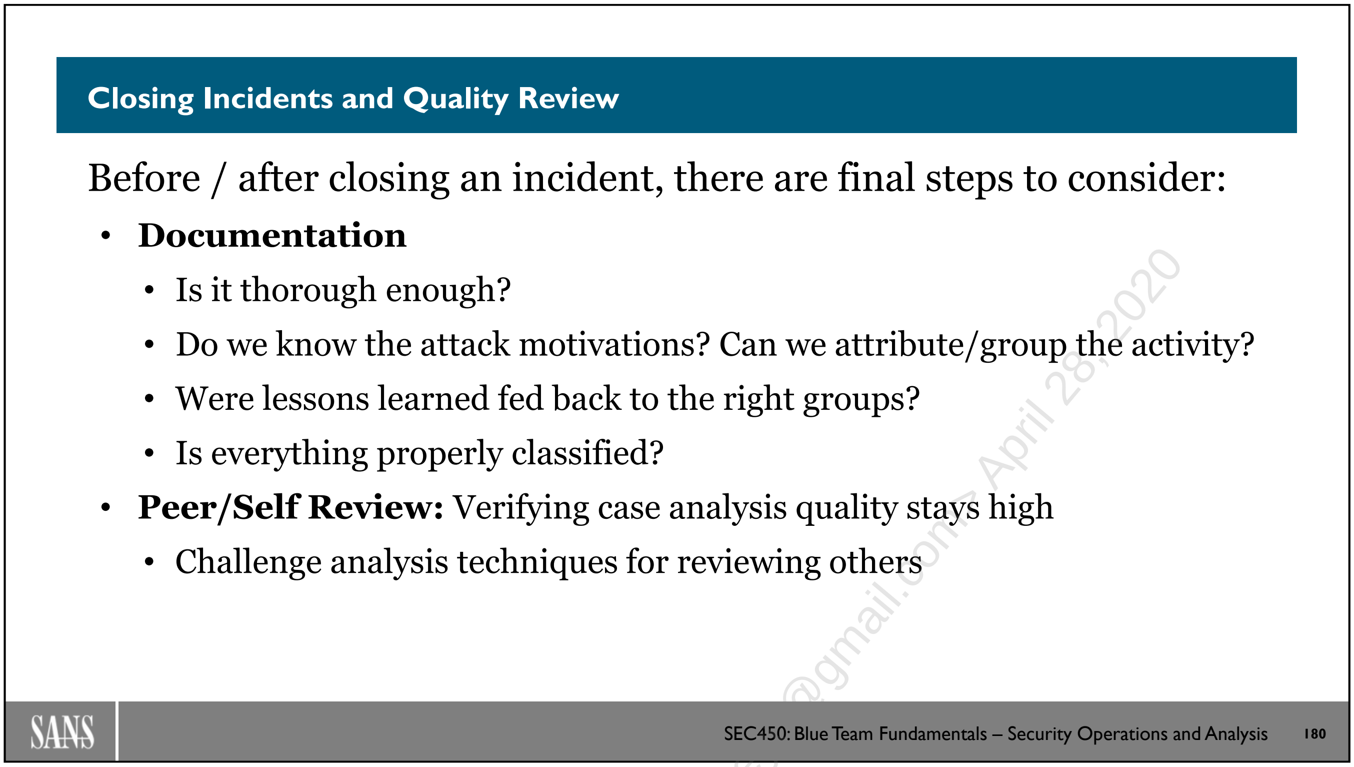
Course Roadmap

- Day 1: Blue Team Tools and Operations
- Day 2: Understanding Your Network
- Day 3: Understanding Hosts, Logs, and Files
- **Day 4: Triage and Analysis**
- Day 5: Continuous Improvement, Analytics, and Automation

Triage and Analysis

1. Alert Triage and Prioritization
2. Perception, Memory, and Investigation
3. Models and Concepts for Infosec
4. Exercise 4.1: Alert Triage and Prioritization
5. Structured Analytical Techniques
6. Analysis Questions and Tactics
7. Analysis OPSEC
8. Exercise 4.2: Structured Analysis Challenge
9. Intrusion Discovery
10. **Incident Closing and Quality Review**
11. Day 4 Summary
12. Exercise 4.3: Collecting and Documenting Incident Information

This page intentionally left blank.



Closing Incidents and Quality Review

Before / after closing an incident, there are final steps to consider:

- **Documentation**
  - Is it thorough enough?
  - Do we know the attack motivations? Can we attribute/group the activity?
  - Were lessons learned fed back to the right groups?
  - Is everything properly classified?
- **Peer/Self Review:** Verifying case analysis quality stays high
  - Challenge analysis techniques for reviewing others

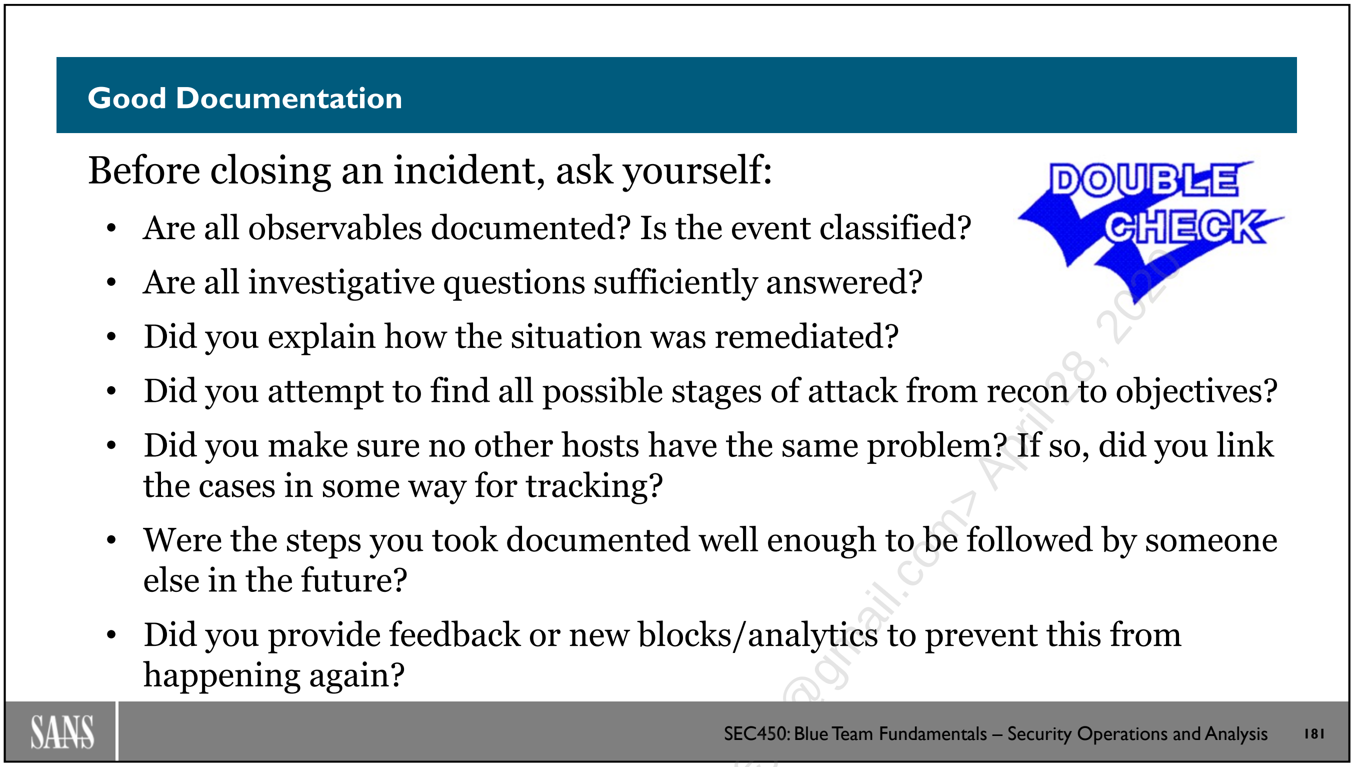
SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 180

Closing Incidents and Quality Review

You will likely work hundreds to thousands of incidents in your time as an analyst, which means you'll be leaving behind a small mountain of documentation through the cases you've worked. When it comes to closing out those cases and making sure your analysis stays high quality, there are a few things to remember, which we will review in this module. For documentation, you want to make sure you are being thorough and that others can follow your work. Cases you work contain lessons learned that need to be fed back into the collection, detection, and triage process, and part of being able to do that is understanding how the attack worked and what the goals were. We also need to classify closed items in a detailed-enough way that we can use the aggregated attack data as another source of feedback. This means we will need to leave behind a complete investigation record.

On top of that, we have discussed the need for systematic feedback to ensure that we are constantly learning and evolving our analytic capability. Without some type of peer review for yourself and others, the SOC will be unable to progress in analytical talent. Therefore, we will close this module with some additional structured analytic techniques that can be used to review yours or others' work, as well as some additional methods to challenge threat modeling and any other plans you may think are suboptimal.



Good Documentation

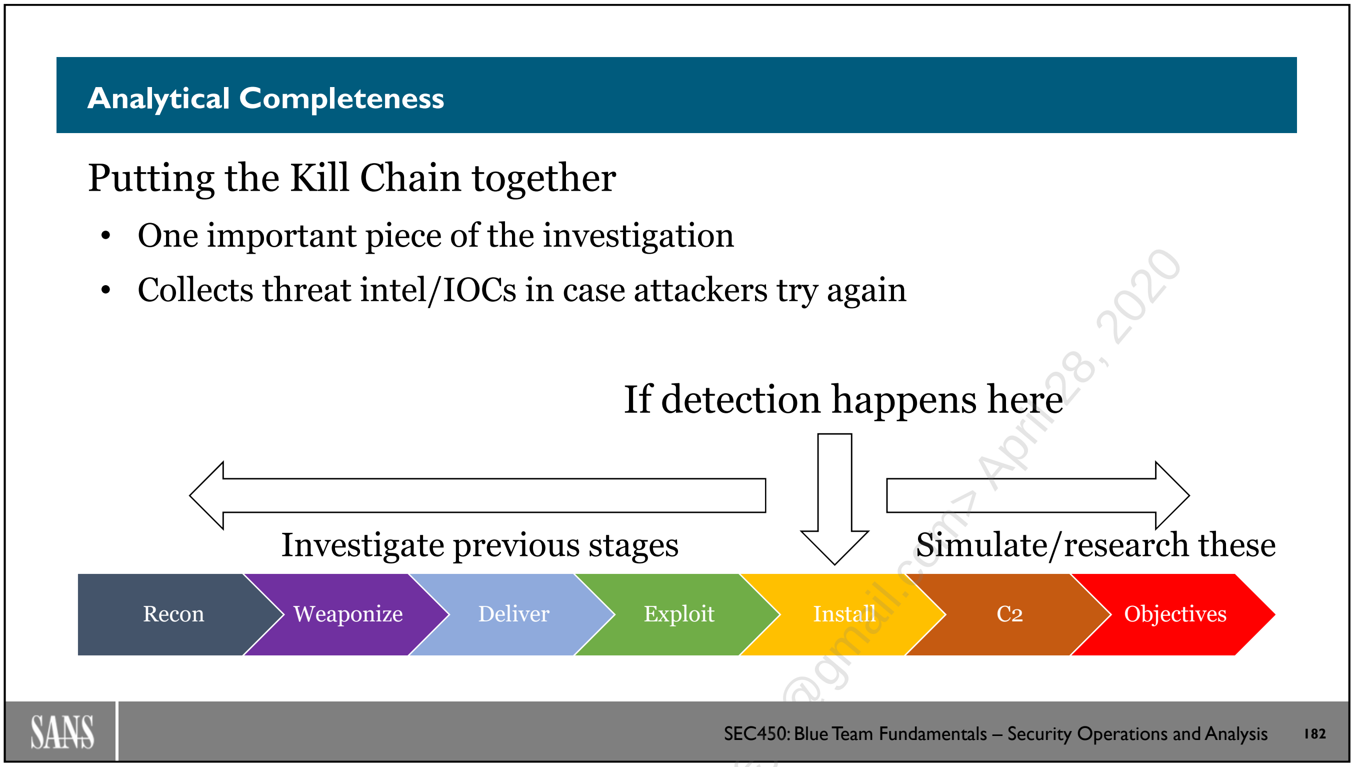
Before closing an incident, ask yourself:

- Are all observables documented? Is the event classified?
- Are all investigative questions sufficiently answered?
- Did you explain how the situation was remediated?
- Did you attempt to find all possible stages of attack from recon to objectives?
- Did you make sure no other hosts have the same problem? If so, did you link the cases in some way for tracking?
- Were the steps you took documented well enough to be followed by someone else in the future?
- Did you provide feedback or new blocks/analytics to prevent this from happening again?

Good Documentation

Before closing an incident, there are several questions you should ask yourself about the depth of your investigation. The goal of the notes you took as you worked through the case are to be thorough enough that someone can follow and reproduce the activity you took in case you were to see that type of malware or situation again, but not *so* detailed it becomes inefficient and painful.

The slide shows some questions that should be answered before closing a case. The list is not exhaustive, of course, but covers some of the big items like solving all questions about the tactics and motivations of the attack, as well as ensuring the situation has been remediated on both that host and all others and feeding back any information that was learned about defensive failures back to the appropriate groups. Note that not *all* cases need the same level of documentation, but in general, bigger incidents and targeted attacks require more detail since they are more likely to cause a large impact.



Analytical Completeness

Putting the Kill Chain together

- One important piece of the investigation
- Collects threat intel/IOCs in case attackers try again

If detection happens here

Investigate previous stages

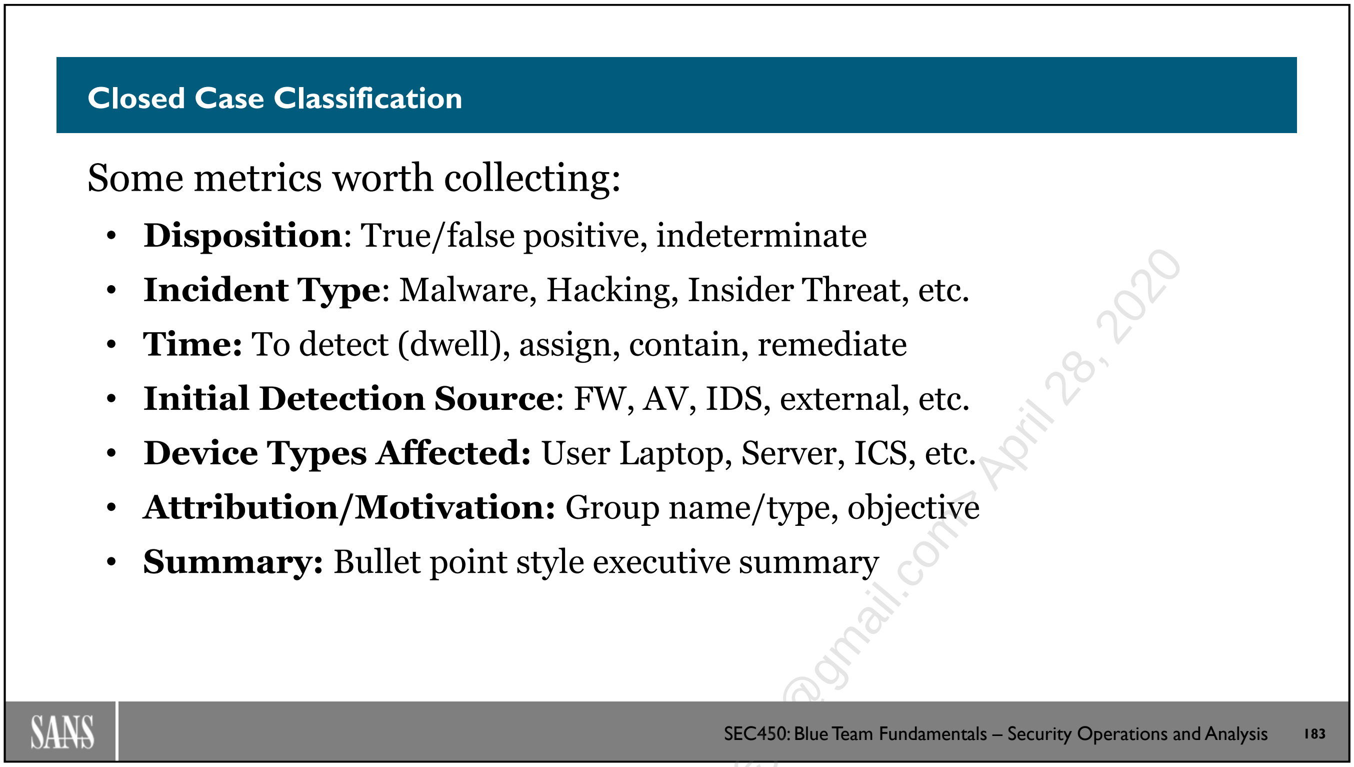
Simulate/research these

Recon → Weaponize → Deliver → Exploit → Install → C2 → Objectives

SEC450: Blue Team Fundamentals – Security Operations and Analysis 182

Analytical Completeness

One of the items to remember before an investigation is complete is putting the attacker kill chain back together where possible. In any investigation, you drop in at some point on the progression down the line. In an ideal world, for example, if you detect an intrusion with an antivirus detection, that starts you at the "installation" phase of the kill chain. From there, your goal should be to work backwards, finding the exploit, delivery, and even evidence of weaponization or targeting, if possible (even though it's often not). Additionally, you should try to move forward. If antivirus caught the file and it did not run, take a sample or look up a publicly available version of the virus run through a sandbox and see what sites it would've talked to for command and control and put those indicators into your investigation as well. This way, this information can get put into your threat intel database and will be flagged if the same infrastructure is used later for another virus that is not immediately caught.



Closed Case Classification

Some metrics worth collecting:

- **Disposition**: True/false positive, indeterminate
- **Incident Type**: Malware, Hacking, Insider Threat, etc.
- **Time:** To detect (dwell), assign, contain, remediate
- **Initial Detection Source**: FW, AV, IDS, external, etc.
- **Device Types Affected:** User Laptop, Server, ICS, etc.
- **Attribution/Motivation:** Group name/type, objective
- **Summary:** Bullet point style executive summary

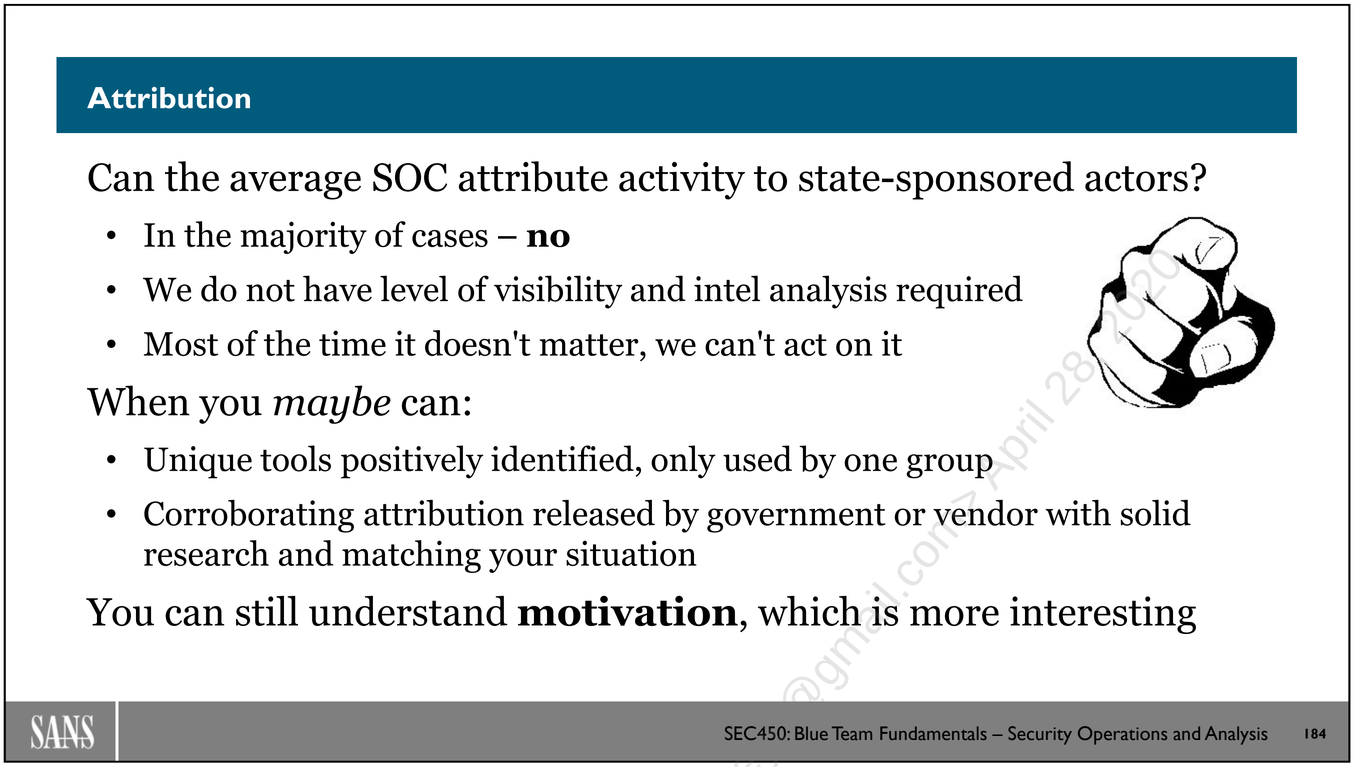
Closed Case Classification

The classification step of closing a case is important because it's the point in time where you can create metrics and trends about ongoing attack trends within your environment. The best and most relevant threat intelligence is the data you generate inside your organization based on the attacks you see, and the classification of cases is one of the big contributors to the dataset.

What type of data should you collect as you close a case? Anything you can think of that, when collected in aggregate, will help you continuously improve defenses in a meaningful way. Some of the examples listed on the slide are incident type, key points in time, detection sources, and the types of devices affected. With these bits of info, you can adjust your threat model, analyze your team's response speed, measure the usefulness of each individual security tool, and focus defenses on the population of devices that see the most attacks. Attribution and motivation of the attacks that you can identify can help validate your threat models' preconceptions about what types of threat actors are interested in you. The summary field, although not metadata, is important, so when someone goes back and views a ticket months or years later, they can get a quick bullet-point version of what happened without having to read pages of investigation notes. If your incident management system does not let you track these metrics, features like tags may allow you to wrench this data into the incident without it having an official spot in a form.

For those looking for additional ideas for incident classification, the previously mentioned VERIS framework is a great, highly detailed (perhaps too detailed if you filled everything out) framework for incident tracking.[1]

[1] http://veriscommunity.net/



Attribution

Can the average SOC attribute activity to state-sponsored actors?

- In the majority of cases – **no**
- We do not have level of visibility and intel analysis required
- Most of the time it doesn't matter, we can't act on it

When you *maybe* can:

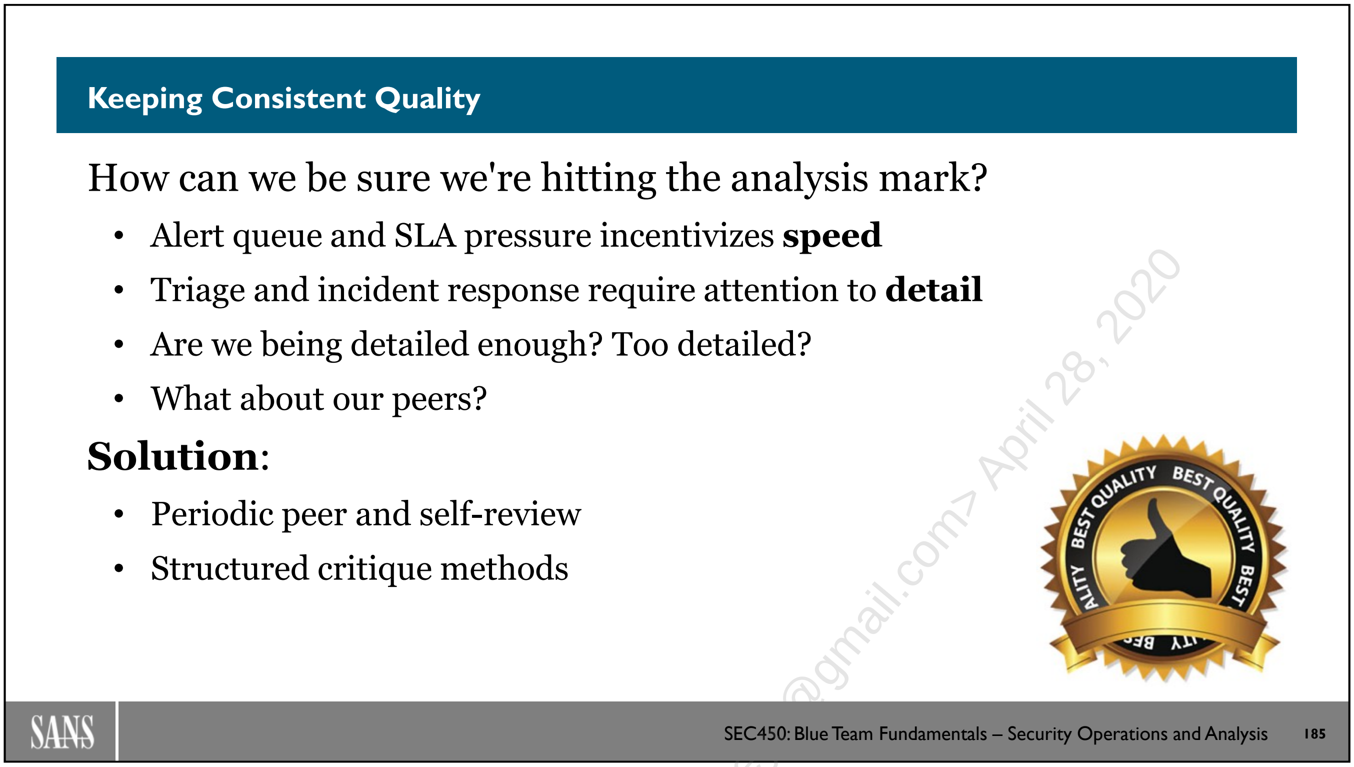
- Unique tools positively identified, only used by one group
- Corroborating attribution released by government or vendor with solid research and matching your situation

You can still understand **motivation**, which is more interesting

Attribution

When closing an incident, should the average SOC worry about attribution of a targeted attack? Can we even identify the perpetrators with confidence? Unfortunately, in most cases the answer is no. Attribution is *extremely* difficult as is, and once you add in consideration of purposeful deception, things get even murkier. The simple truth is that most average SOCs will not have the background and body of threat intelligence required to pin a given attack to a specific threat actor, let alone determine the difference between true attribution and attempted deception. The exception to this may be when specific tooling is found, positively identified, and is known to be associated with a single threat actor. In cases like this, you may see a report come out from a government CERT or threat intelligence vendor like FireEye or CrowdStrike after a period that names the tool, the campaign TTPs, and perhaps even the hash of a sample you found. Outside of this, it is unlikely that any civilian organization will have the breadth of knowledge required for proper attribution.

Is this something we should be disappointed about? Not necessarily. In the grand scheme of things, the SOC's job is to keep the organization safe. Aside from the satisfaction of knowing and feeding threat intelligence, having an attribution of Actor 1 is not going to drive any different reaction than Actor 2. The unfortunate truth is that most attackers go unpunished and organizations are not going to release indictments of state-sponsored actors, take them to court, or bring any repercussions to the attacker. Therefore, in most cases, it is better to focus on the motivation of the attacker and consider how you will better prevent that scenario in the future, regardless of the source of the attack.



Keeping Consistent Quality

How can we be sure we're hitting the analysis mark?

- Alert queue and SLA pressure incentivizes **speed**
- Triage and incident response require attention to **detail**
- Are we being detailed enough? Too detailed?
- What about our peers?

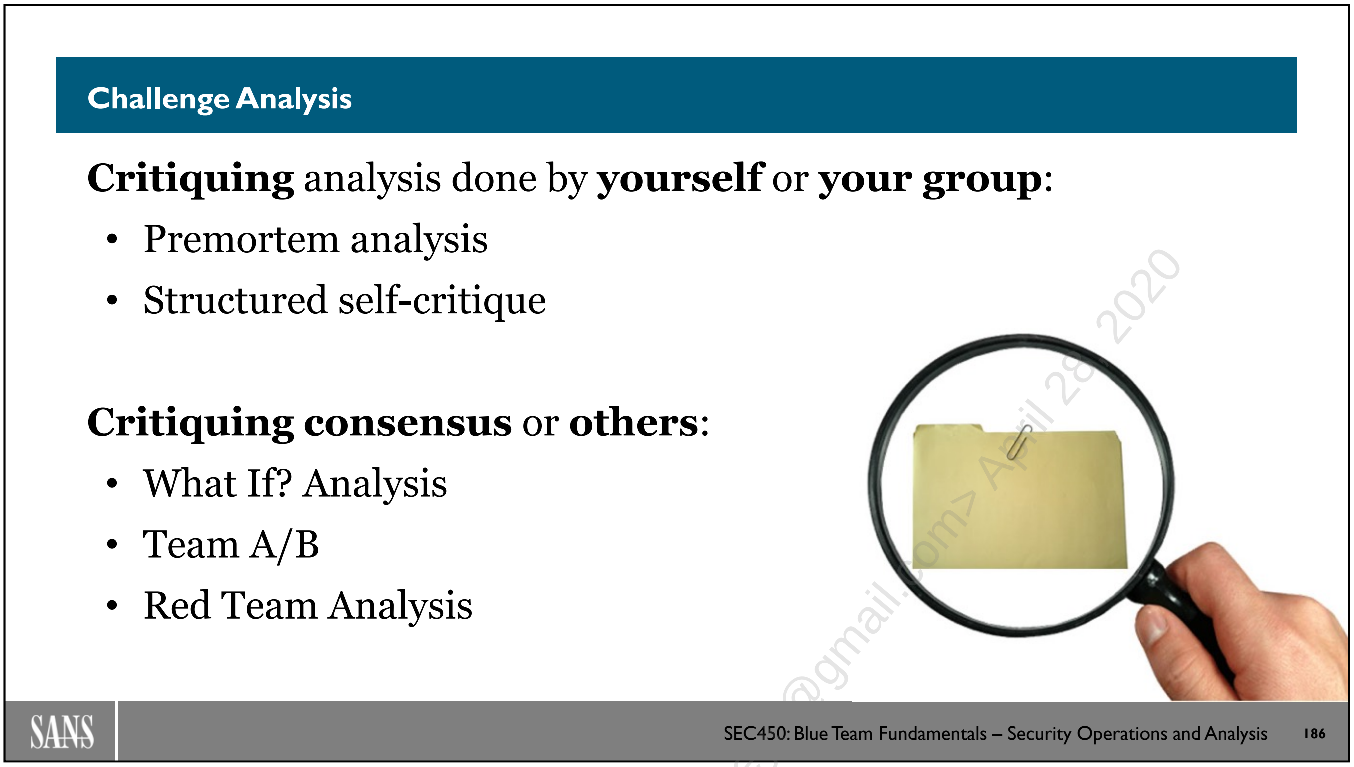
Solution:

- Periodic peer and self-review
- Structured critique methods

BEST QUALITY BEST QUALITY BEST QUALITY BEST

Keeping Consistent Quality

After triaging, investigating and closing alerts over the months and years, you might start to wonder how well you're doing. How can we get feedback on our capabilities and ensure we are first hitting the standard that we should be hitting for investigation quality in the first place and keeping it up over time? As we start to see the same situations repeatedly, we're more likely to develop the ability to use system 1 thinking, and SLAs and alert queues will tempt you to do so. While these things incentivize speed, triage and IR, in the long run, they require attention to detail and thorough investigation to ensure you aren't only partially removing attackers from the environment. Given these opposing pressures, how can we check ourselves and perhaps compare ourselves to our peers? One solid solution is periodic feedback provided through peer or self-review of your past investigations' structured critique methods.



Challenge Analysis

Critiquing analysis done by **yourself** or **your group**:

- Premortem analysis
- Structured self-critique

Critiquing consensus or **others**:

- What If? Analysis
- Team A/B
- Red Team Analysis

SANS

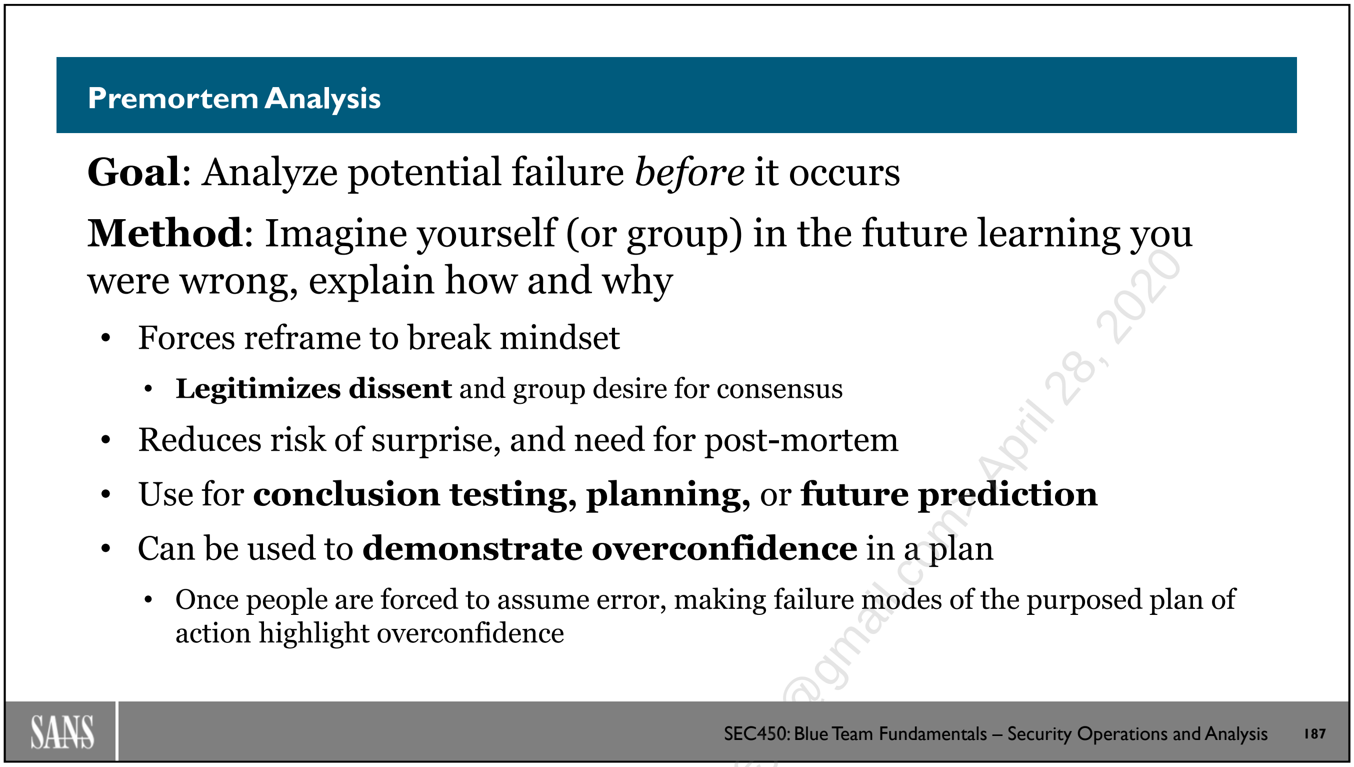
SEC450: Blue Team Fundamentals – Security Operations and Analysis 186

Challenge Analysis

A final type of structured analysis that we've put off discussing until now is the category Heuer and Pherson call "Challenge Analysis." Also sometimes called contrarian, alternative, competitive, or red team analysis, these terms share the goal of challenging an analytic consensus or model. According to Heuer and Pherson, past analytic failures within the intelligence community have often been the fault of a failure of imagination for alternative hypothesis or due to the lack of challenging consensus mental models. Since our mental models are created by the totality of everything we've seen and experienced in the past, they are highly influential on our judgment. They subconsciously tell us what to look for, which key pieces of evidence are the most important, and inform how to assemble the data we receive. The problem is that they are also slow to change; therefore, using these review techniques on ourselves and our peers can give us the systematic feedback required to keep analytical quality high and ever increasing. These methods use the technique of *reframing* to trick our minds into viewing an analysis from a new point of view, keeping our mind open and out of mental ruts.[1]

In order to fight the tendency to subconsciously and inappropriately write off new evidence, Heuer says implementing systematic feedback into our analysis process is required. To learn from experience, one must know the outcome of what they have done in the past. Therefore, without closing the loop via regular feedback, analysts will not know when their thinking went wrong, and it will not be possible to learn to make better judgments. According to Heuer, this feedback should include an assessment of "the accuracy of a judgment with the particular configuration of variables that prompted an analyst to make that judgment."[1] To this end, in this module, we will discuss the closing of incidents and how to implement periodic peer review into the SOC process to ensure analysts receive the feedback they need to constantly improve their analytic ability and mindsets.

[1] Heuer & Pherson, 2015, p. 232



Premortem Analysis

Goal: Analyze potential failure *before* it occurs

Method: Imagine yourself (or group) in the future learning you were wrong, explain how and why

- Forces reframe to break mindset
  - **Legitimizes dissent** and group desire for consensus
- Reduces risk of surprise, and need for post-mortem
- Use for **conclusion testing, planning,** or **future prediction**
- Can be used to **demonstrate overconfidence** in a plan
  - Once people are forced to assume error, making failure modes of the purposed plan of action highlight overconfidence

SANS

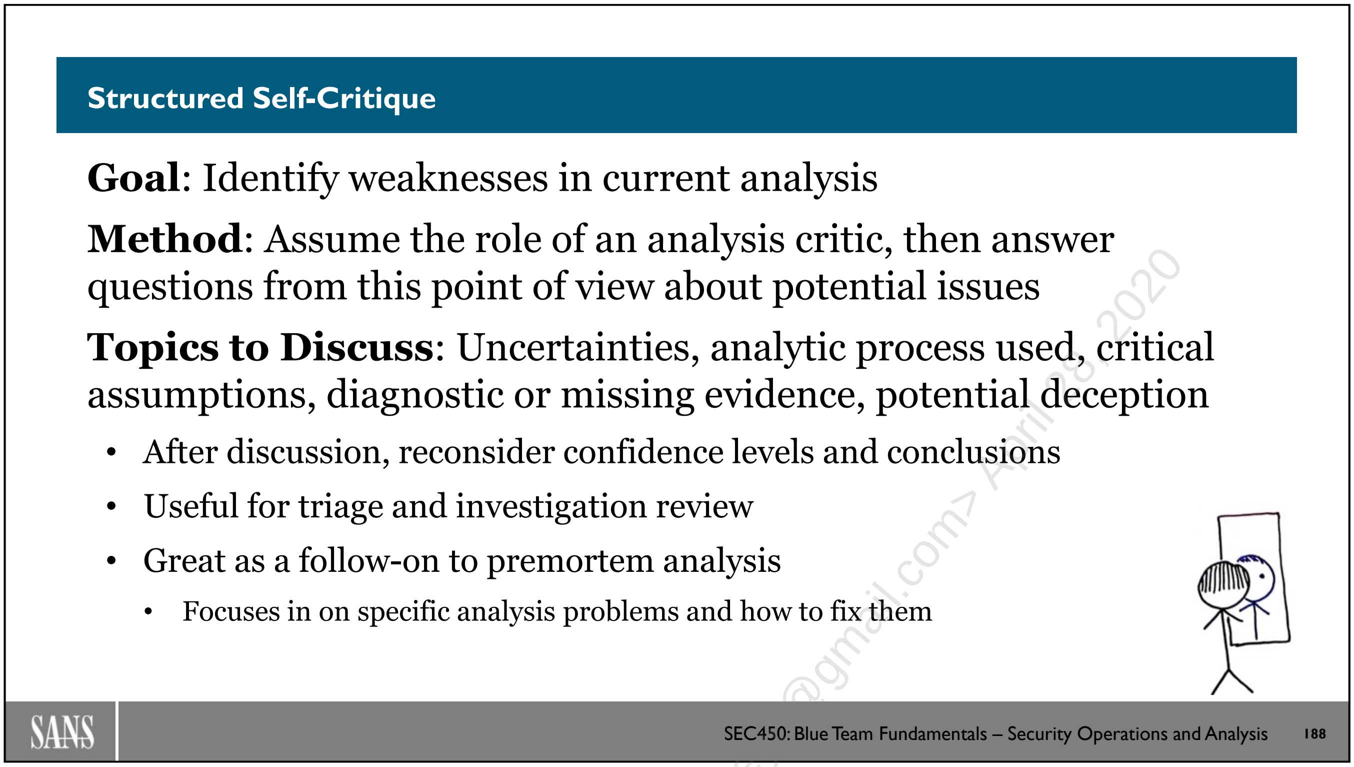
SEC450: Blue Team Fundamentals – Security Operations and Analysis 187

Premortem Analysis

One structured critique technique that can be used to test the strength of a proposed plan of action or analytical conclusions is the premortem analysis.[1] The goal of this technique is to analyze possible methods of failure *before* they occur to reduce the risk of future surprise, and the need to do a postmortem analysis because something *has* gone wrong. The method for this technique is to either by yourself, or in a group, have a meeting where you imagine yourselves at some point in the future learning that your conclusions or plan of action has gone wrong. Your job is to come up with *all* ideas of how and why that occurred, typically in a round-robin everyone speaks type of fashion. Doing this forces everyone to reframe the situation in their mind and breaks you out of mental ruts and groupthink. The expected outcome of this method is a more thorough understanding of the uncertainty of the situation as well as the identification of early warning signs that the plan is not going as anticipated.

This technique can be an outstanding way to break through the issue of groups desiring a fast consensus. Forcing everyone to take a dissenting opinion and suggest how failure could be possible legitimizes dissent that may have gone unspoken due to politics or other group dynamics and gives a voice to potential alternative viewpoints. It can also be used as an outstanding technique to highlight overconfidence in a plan. When people or groups come up with a plan of action, they are often extremely confident (overly so) in how well it will likely work. Once you perform a premortem analysis, forcing the reframe and the requirement to list potential methods of failure often highlights potential shortcomings of the plan and allows the creators to take premeditate action to prevent or detect the failure conditions coming to light, allowing them to better control the situation. Note that although premortem analysis does identify that potential problems exist, it does not necessarily highlight which problem or explain how to fix it. The next method, the structured self-critique, can do a more complete job of this part.

[1] Heuer & Pherson, 2015, pp.240-243



Structured Self-Critique

Goal: Identify weaknesses in current analysis

Method: Assume the role of an analysis critic, then answer questions from this point of view about potential issues

Topics to Discuss: Uncertainties, analytic process used, critical assumptions, diagnostic or missing evidence, potential deception

- After discussion, reconsider confidence levels and conclusions
- Useful for triage and investigation review
- Great as a follow-on to premortem analysis
  - Focuses in on specific analysis problems and how to fix them

Structured Self-Critique

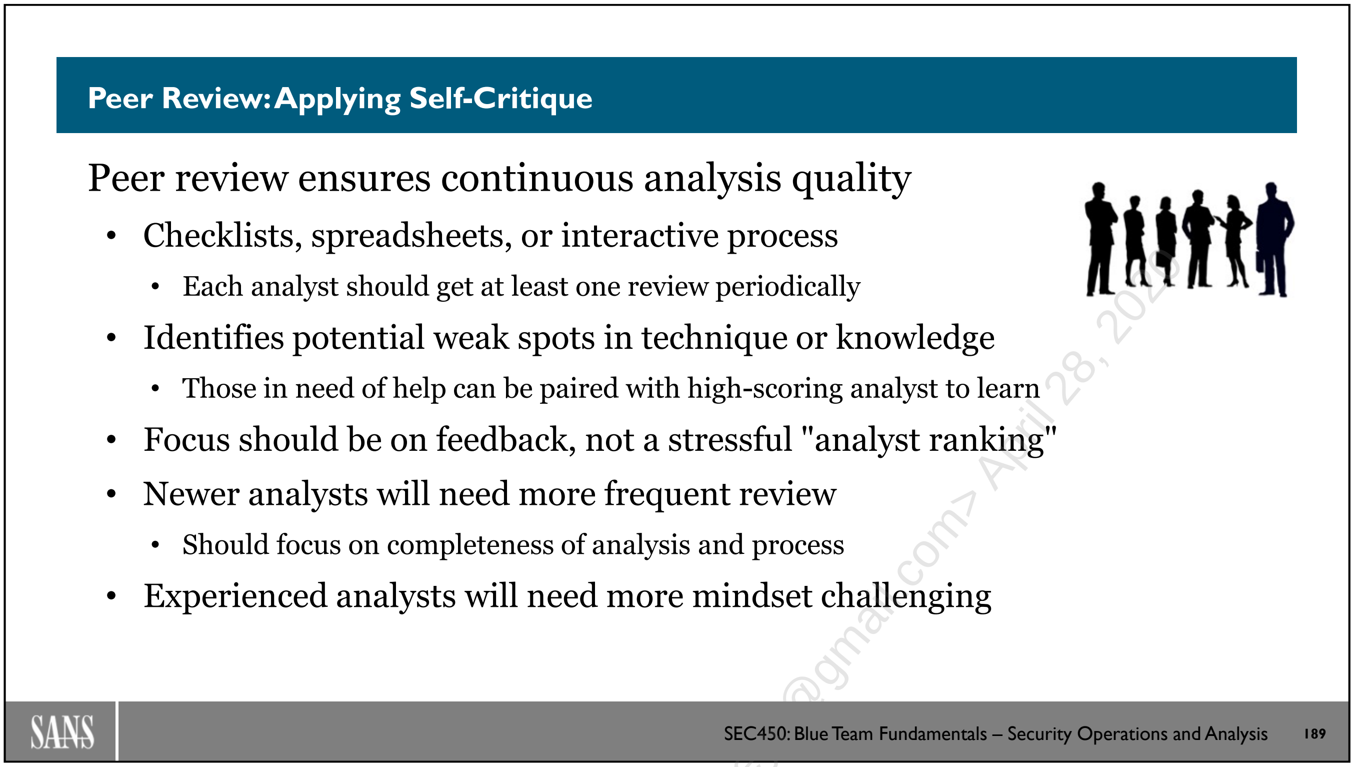
Another great self-assessment method is structured self-critique. This method is a great follow-on to a premortem analysis because it is more focused on finding the specific problems that may have occurred. For this technique, the method is to have everyone put on their pretend "black hat" and analyze a conclusion that has been reached by assuming a critical viewpoint and answering questions about the analysis. In this way, each person will be forced to pick apart the process, assumptions, evidence, and other aspects that led to the conclusion and come up with any reason for error.

Questions that should be asked should include things like:

- Were our key sources of evidence reliable?
- Was contradictory evidence ignored?
- Do we have an explanation for missing evidence?
- Were our key assumptions valid?
- Did we seriously generate and consider alternative hypotheses?
- Did the absence of information mislead us?
- Did deception go undetected?[1]

This technique is like the Devil's Advocacy technique where a single member of the team is designated to play the dissenting role. Although this method can work, Heuer and Pherson say that when only one person is dissenting, the team tends to get more defensive and the technique becomes less productive than having everyone do it.

[1] Heuer & Pherson, 2015, p. 244-247



Peer Review: Applying Self-Critique

Peer review ensures continuous analysis quality

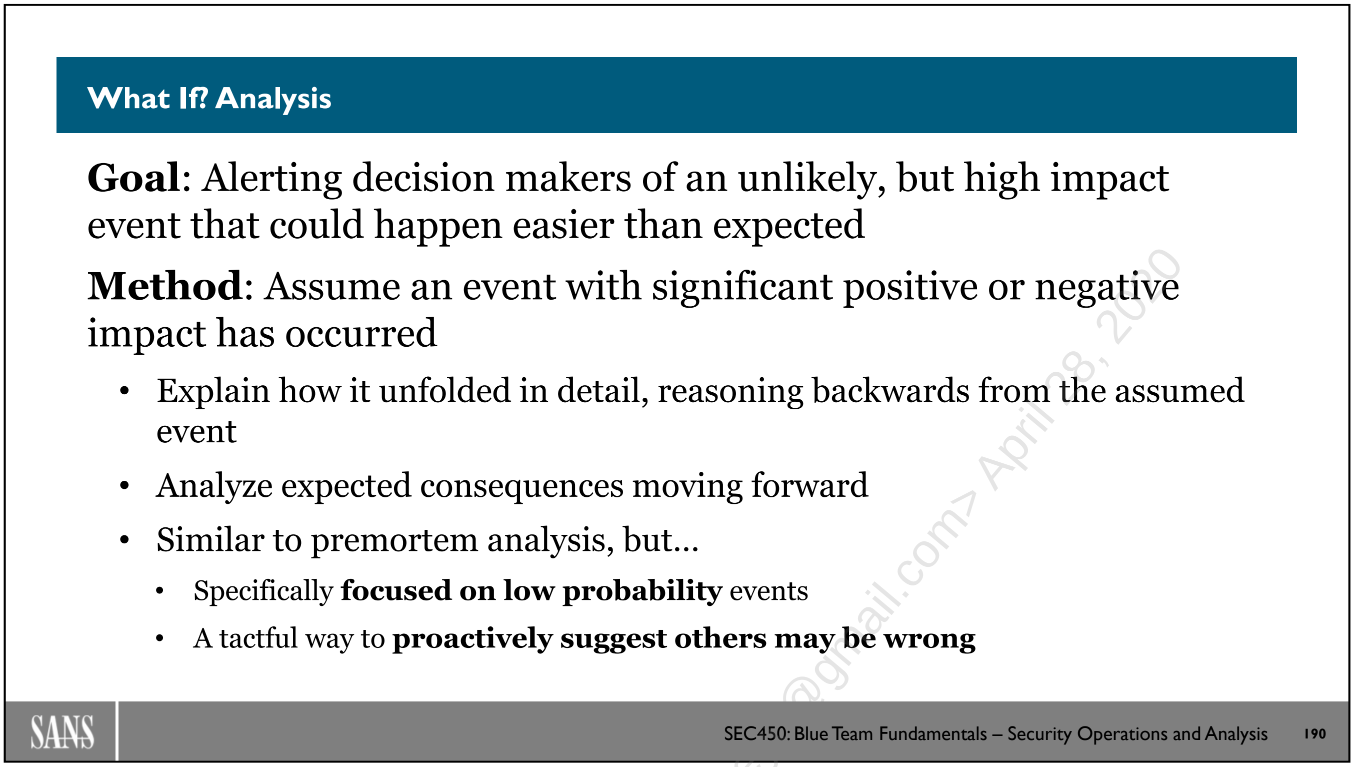
- Checklists, spreadsheets, or interactive process
  - Each analyst should get at least one review periodically
- Identifies potential weak spots in technique or knowledge
  - Those in need of help can be paired with high-scoring analyst to learn
- Focus should be on feedback, not a stressful "analyst ranking"
- Newer analysts will need more frequent review
  - Should focus on completeness of analysis and process
- Experienced analysts will need more mindset challenging

Peer Review: Applying Self-Critique

Given the structured self-critique method, how can we operationalize it so that it becomes part of the culture of the SOC? One way is to institute mandatory periodic reviews of a random sample case from each member of the SOC. This review can be once a week, month, or any period desired, but given that feedback is a key component of growth, it needs to happen at some interval. During this review, the case should be read through by a single analyst or group of other analysts from the SOC and notes should be taken about what was and was not done. The questions about analytical completeness and if all aspects of the attacker were found should be assessed as well as the analytic technique used. Did the person come up with multiple hypotheses and explain why they believed whatever conclusion they came up with? These reviews can be qualitative or point values can be assigned to each question to get a more objective measure, but if analysts hear back where they can improve from others, the objective should be met.

Aside from quality, there are other benefits of doing reviews. One is that newer analysts can see the technique and thinking process of the more experienced analysts and start to understand more quickly what tools and methods they should use in various situations. Another is that patterns of deficiencies in certain areas can be identified, and those with a need to learn a tool or technique can be paired with those who know it well for efficient on-the-job training.

One word of caution about peer review. The spirit of peer review should be kept light-hearted and purely focused on being a learning tool. If it becomes a stress-inducing "analyst ranking" system, the benefits of it may be overshadowed by the problems it causes. Ensure that all analysts know it is purely for their own learning and will not be used to punish them for anything they aren't yet exceeding at. To this end, you will probably find it beneficial to give newer analysts more frequent reviews than those who have been around longer, and the nature of those reviews will likely be different as well. Newer analyst reviews should focus on if their cases are analytically complete, the right tools were used, and the process they used to come up with a conclusion is solid. For more experienced analysts who understand the available tools and analysis process, reviews may be more of a "red team analysis" type exercise that challenges their conclusions and mindset and makes sure they didn't jump to conclusions based on mental ruts.



What If? Analysis

Goal: Alerting decision makers of an unlikely, but high impact event that could happen easier than expected

Method: Assume an event with significant positive or negative impact has occurred

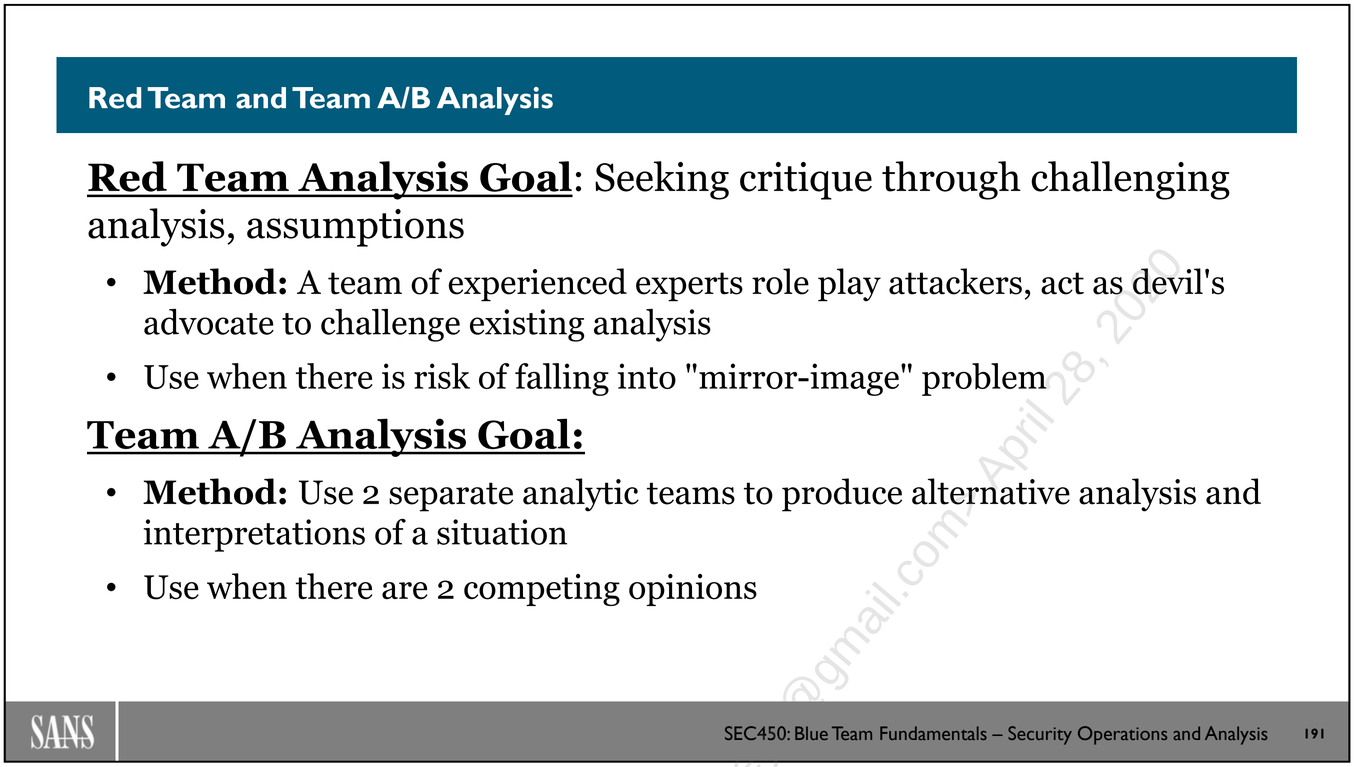
- Explain how it unfolded in detail, reasoning backwards from the assumed event
- Analyze expected consequences moving forward
- Similar to premortem analysis, but…
  - Specifically **focused on low probability** events
  - A tactful way to **proactively suggest others may be wrong**

What If? Analysis

What If? Analysis is another method for reframing that is built around the idea of imagining some very improbable event has occurred that will have a major positive or negative impact. The idea is using the benefit of this imagined "hindsight" to walk backward step by step and put back together the events that led to it and also what the consequences will be. Key pieces of this method are to develop at least one chain of events based on evidence and logic to explain how the event may occur, focusing on what must happen at each stage of the process. Each step should contain observables or indicators that things are leaning in that direction and the analysis should also include a list of consequences as well as how difficult and costly recovery would be for each scenario. The additional benefit of this technique is that this detail gives decision makers additional insight on what to do to prevent an undesired but low probability outcome from occurring and a list of indicators that the undesirable situation may be developing so that it can be proactively spotted.

This type of analysis is like the premortem analysis method in that we are assuming something in the future, but this method focuses on low probability high impact events with an analysis of consequences, as opposed to premortem, which just supposes an incorrect conclusion. Whereas one of the main goals of premortem analysis is to legitimize dissent and enable group members to speak up regardless of group dynamics, What If analysis is a potential way to tactfully suggest a consensus opinion may be wrong. It is best used in the following situations:

- When there is a well-ingrained mental model driving the consensus that a low likelihood event will not occur
- Where an issue is highly contentious, and no one has yet planned for activities assuming the event *did* occur
- When there is a perception that a legitimate possibility is not being given due consideration



Red Team and Team A/B Analysis

Red Team Analysis Goal: Seeking critique through challenging analysis, assumptions

- **Method:** A team of experienced experts role play attackers, act as devil's advocate to challenge existing analysis
- Use when there is risk of falling into "mirror-image" problem

Team A/B Analysis Goal:

- **Method:** Use 2 separate analytic teams to produce alternative analysis and interpretations of a situation
- Use when there are 2 competing opinions

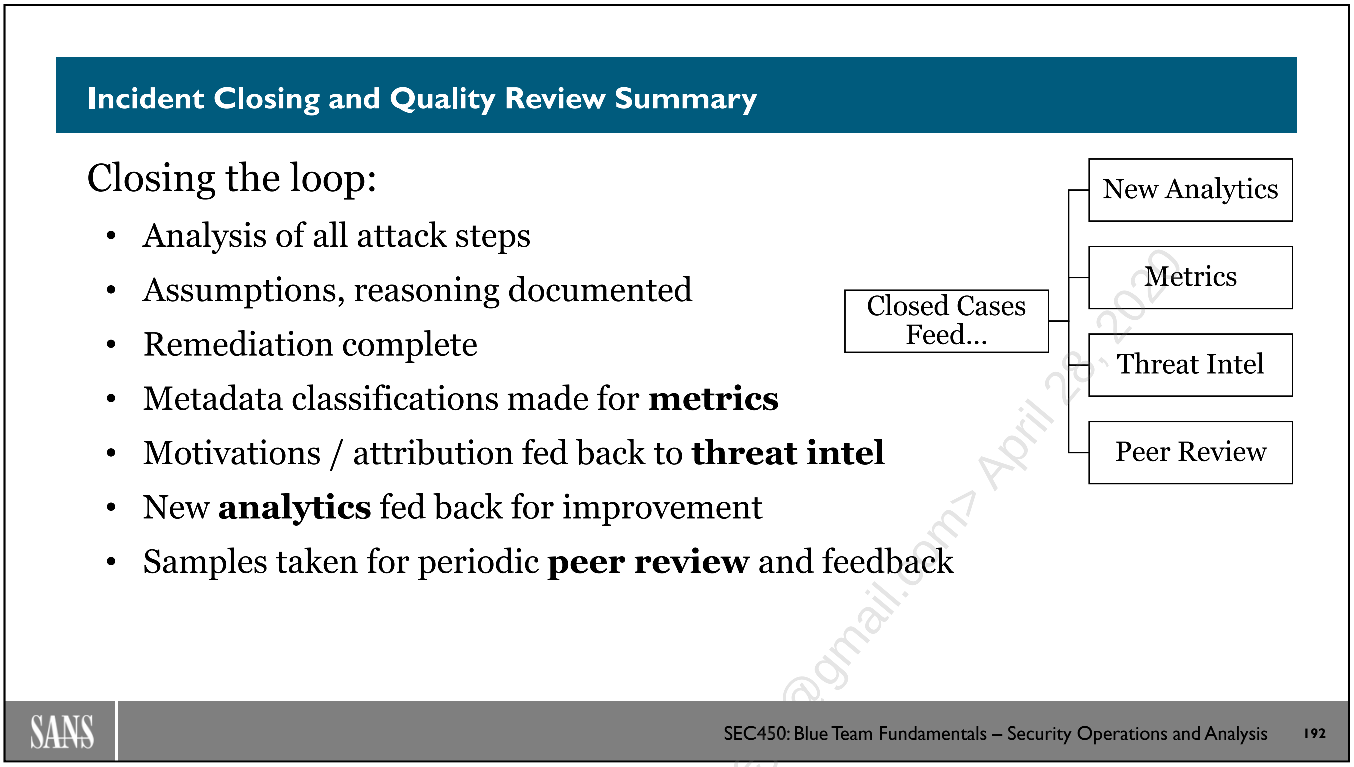
Red Team and Team A/B Analysis

Two final contrarian-style techniques that can be used to assess conclusions is a "red team analysis" and "team a/b analysis." Although the name is the same, red team analysis should not be confused with red teaming in the penetration testing sense. The goal of this method is to cause the author to evaluate their own technique by seeing someone else make the strongest possible case for an alternative explanation, or the case that the analysis is wrong. The idea is to play devil's advocate and see how well of a case can be made. If a strong counterargument cannot be conceived, that bodes well for the strength of the current analysis. This technique is best used when there is a risk of the "mirror-imaging" problem which is described as the tendency for analysts to apply their own motivations, cultural norms and expectations when analyzing a problem and forgetting their attacker's mindset may be *very* different. The red team analysis should include experts who have dealt with that threat group before and can best "put themselves in the attacker's shoes."

Team A/B analysis takes the same evidence and gives it to two different analytic teams to come up with independent conclusions that can be compared. This type of analysis can be useful when there is a division of opinion between analysts about what has occurred. Producing two well-thought-out and competing analyses can help each side understand and appreciate the other's viewpoint. This type of analysis might be done when there is an incident with very little data and there are multiple possible versions of what and how it happened that should both be explored in detail.

Additional detail on these methods, as well as other structured analysis techniques we have discussed, can be found in the free CIA analytic tradecraft primer below.[1]

[1] https://www.cia.gov/library/center-for-the-study-of-intelligence/csi-publications/books-and-monographs/Tradecraft%20Primer-apr09.pdf



Incident Closing and Quality Review Summary

Closing the loop:

- Analysis of all attack steps
- Assumptions, reasoning documented
- Remediation complete
- Metadata classifications made for **metrics**
- Motivations / attribution fed back to **threat intel**
- New **analytics** fed back for improvement
- Samples taken for periodic **peer review** and feedback

Closed Cases Feed...

New Analytics

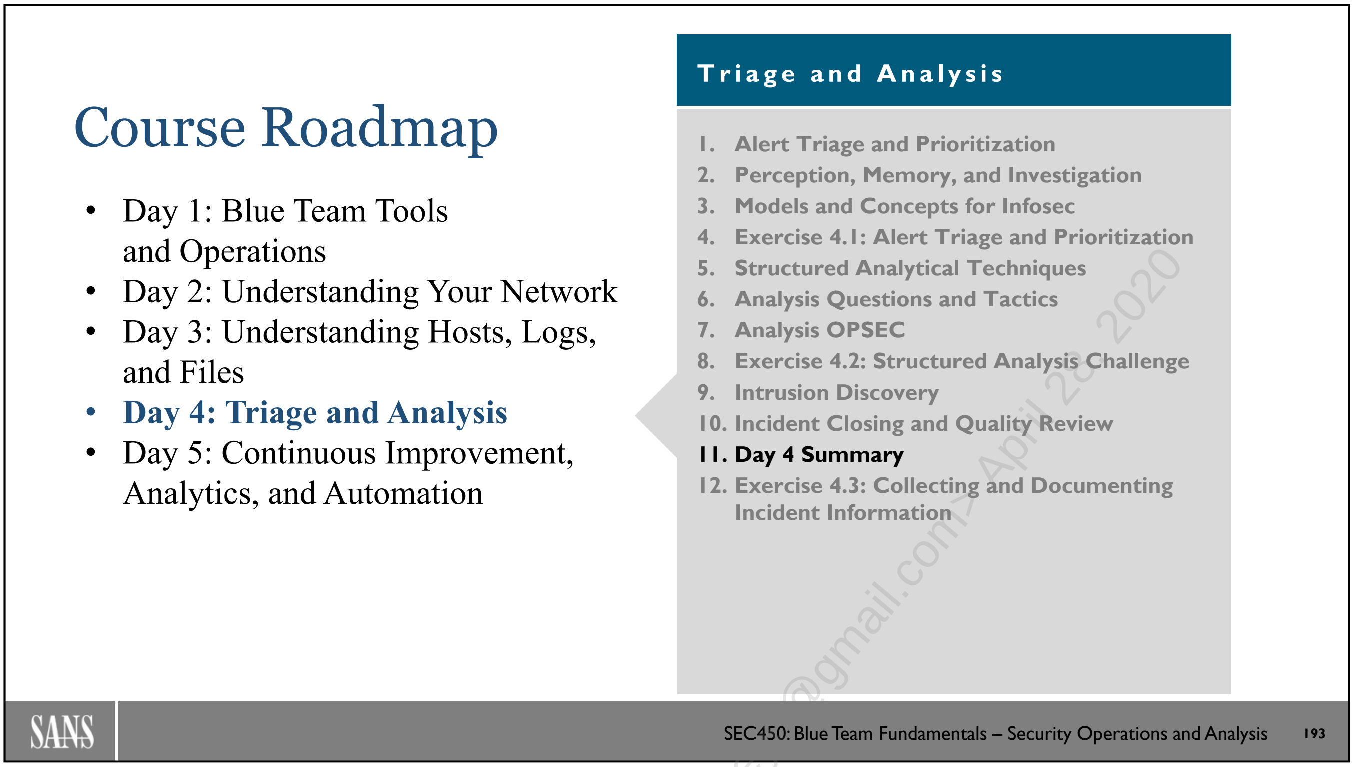
Metrics

Threat Intel

Peer Review

Incident Closing and Quality Review Summary

Closing an incident is a key point in time that allows us to collect important data that should feed multiple other processes. Lessons learned on how the attack was performed can be turned into new collection and data analytics rules, metrics can be used to drive future investment in controls and other security tools, and motivations and attributions can be used to inform the threat intelligence F3EAD cycle. Closed cases also form the library from which data for peer review can be sourced. Although the SOC is under constant time pressure to move on to the "next thing", taking the time to ensure analysis is complete, classified, and the important data is fed back into our processes will pay off in the long run.



Course Roadmap

- Day 1: Blue Team Tools and Operations
- Day 2: Understanding Your Network
- Day 3: Understanding Hosts, Logs, and Files
- **Day 4: Triage and Analysis**
- Day 5: Continuous Improvement, Analytics, and Automation

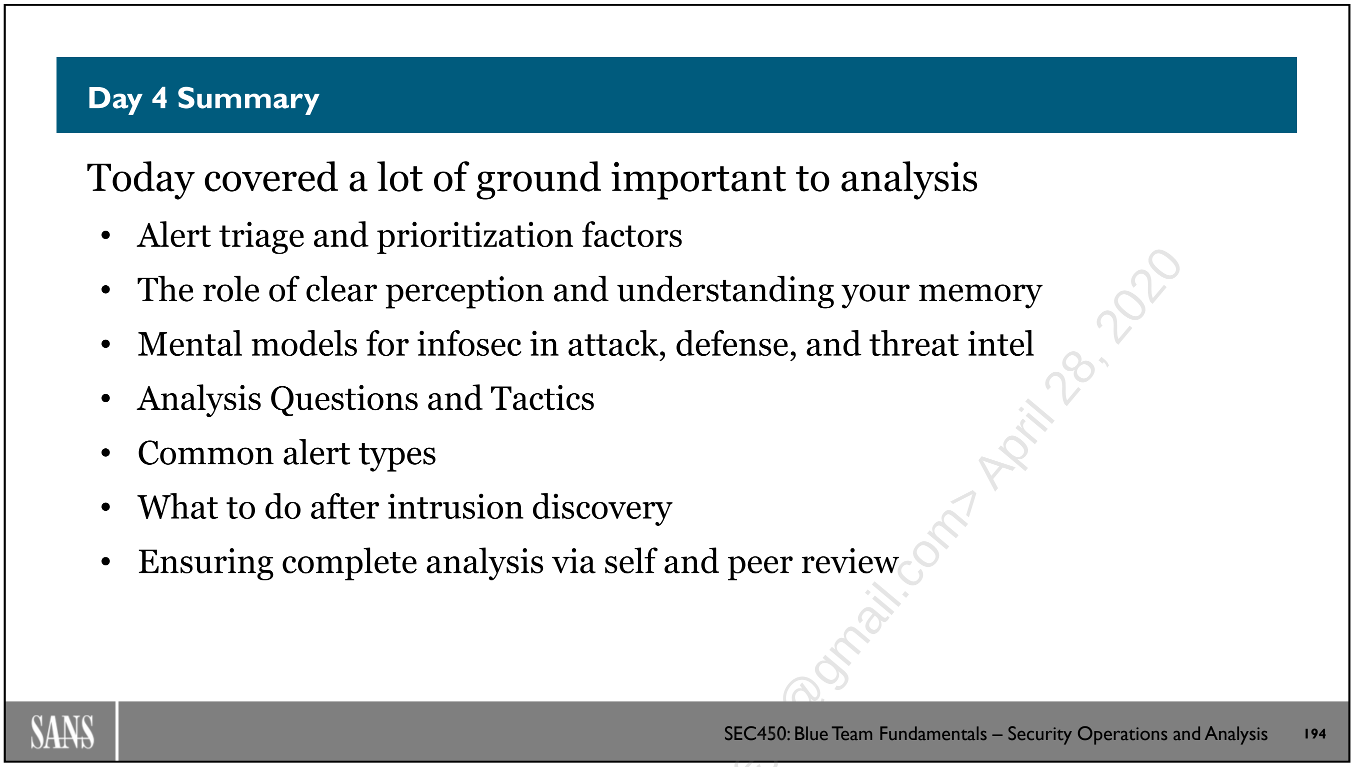
Triage and Analysis

1. Alert Triage and Prioritization
2. Perception, Memory, and Investigation
3. Models and Concepts for Infosec
4. Exercise 4.1: Alert Triage and Prioritization
5. Structured Analytical Techniques
6. Analysis Questions and Tactics
7. Analysis OPSEC
8. Exercise 4.2: Structured Analysis Challenge
9. Intrusion Discovery
10. Incident Closing and Quality Review
11. **Day 4 Summary**
12. Exercise 4.3: Collecting and Documenting Incident Information

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 193

This page intentionally left blank.



Day 4 Summary

Today covered a lot of ground important to analysis

- Alert triage and prioritization factors
- The role of clear perception and understanding your memory
- Mental models for infosec in attack, defense, and threat intel
- Analysis Questions and Tactics
- Common alert types
- What to do after intrusion discovery
- Ensuring complete analysis via self and peer review

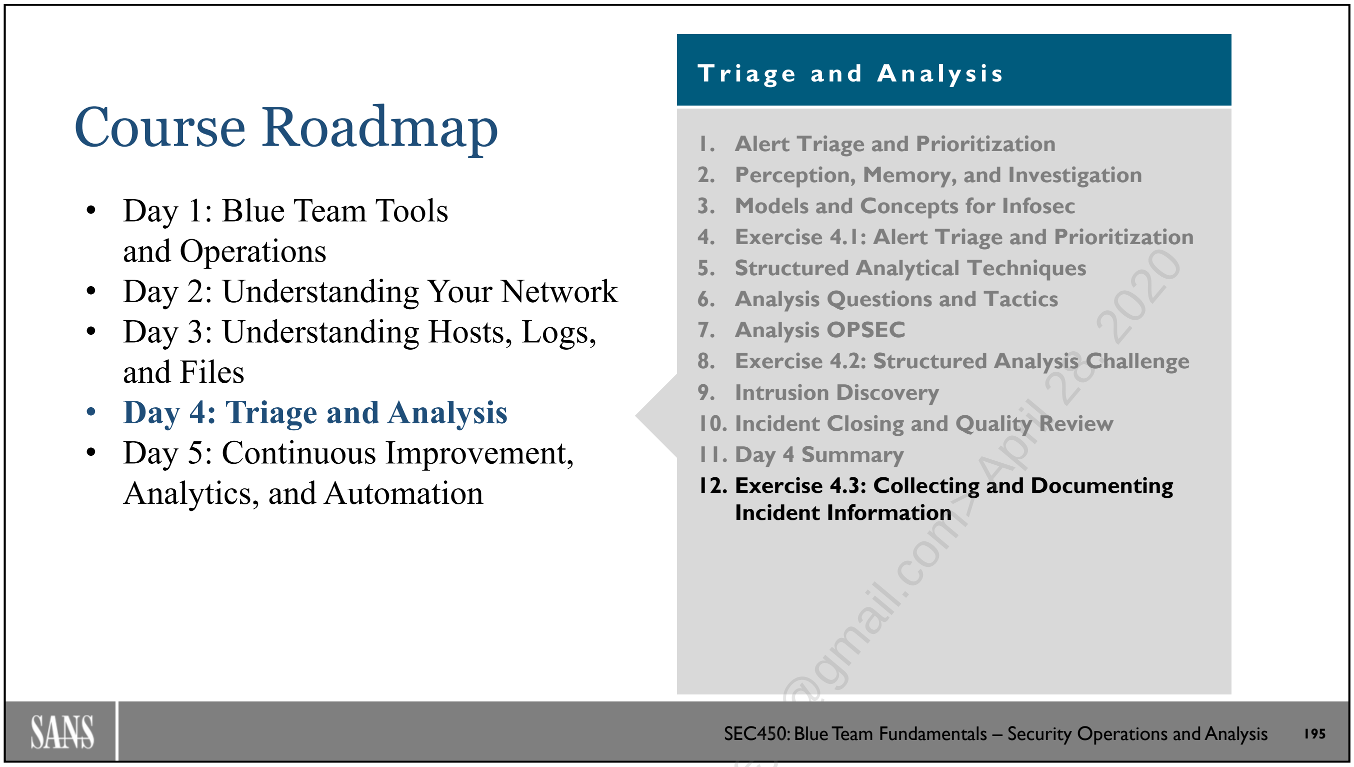
SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 194

Day 4 Summary

We covered a lot of ground in this book. Hopefully throughout our in-depth discussion of how perception, memory, and the brain work, you've had some revelations about how you've been doing analysis in the past and can see how to improve it going forward. As we previously mentioned, although analysts start out with the best of intentions, many do not realize the things they are overlooking until they step back and consider the pitfalls of quick, intuitive analysis. One of the main goals of today was to make you question your process and consider how you can start to look for alternatives and use the disconfirmation of hypotheses instead of trying to confirm your best option to produce more accurate investigations.

On top of the analysis technique information, we have gone through a lot of caveats and tricks that attackers can use to throw us off their trail. Triage, investigation, and closing of alerts can take a while to learn, but the goal in these sections was to give you a jump start on the ways to think and process the alerts you see in front of you, and how to pick and close out the most important ones. On Day 5, we will continue down this path further, looking for additional opportunities to make our day-to-day life in the SOC easier via alert tuning, optimization, and analysis optimization tools and close out with suggestions for further challenging yourself and continuing to develop your skills throughout your career on the blue team.



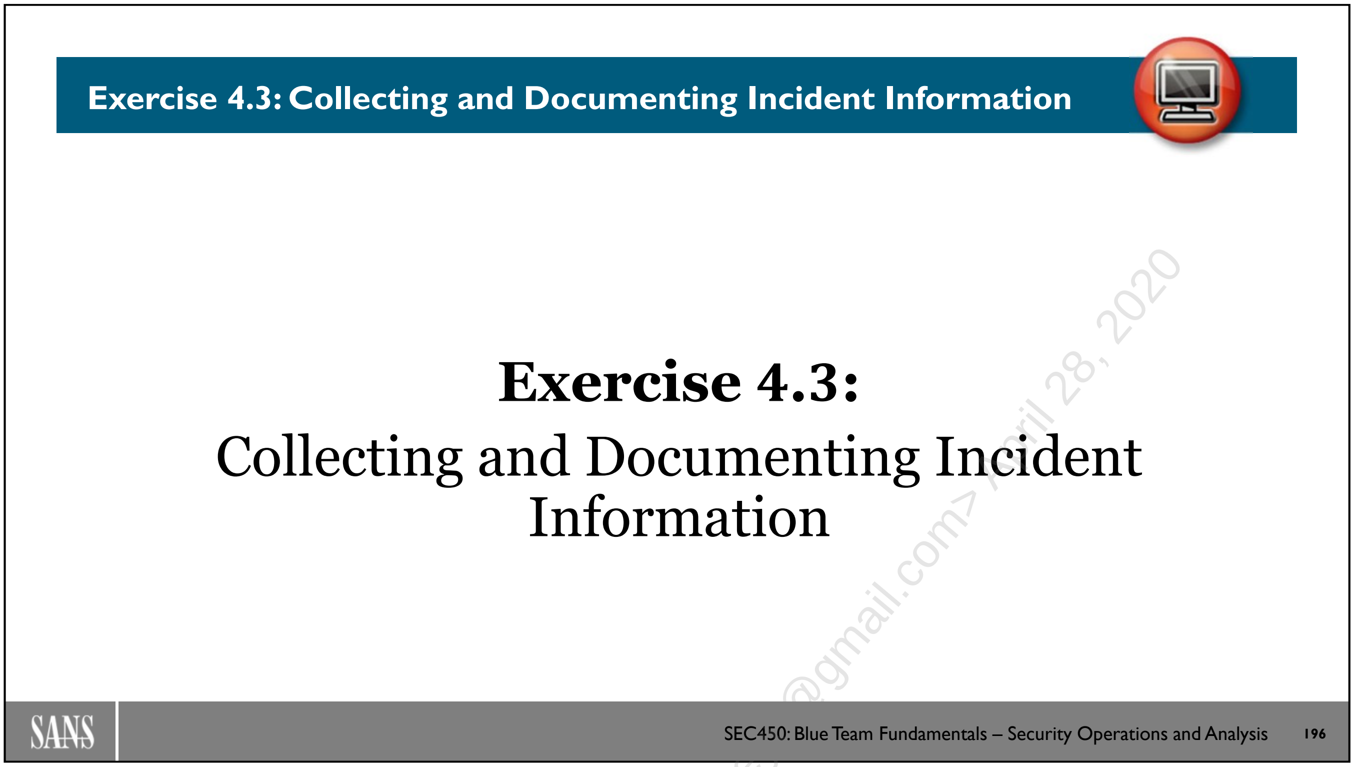
Course Roadmap

- Day 1: Blue Team Tools and Operations
- Day 2: Understanding Your Network
- Day 3: Understanding Hosts, Logs, and Files
- **Day 4: Triage and Analysis**
- Day 5: Continuous Improvement, Analytics, and Automation

Triage and Analysis

1. Alert Triage and Prioritization
2. Perception, Memory, and Investigation
3. Models and Concepts for Infosec
4. Exercise 4.1: Alert Triage and Prioritization
5. Structured Analytical Techniques
6. Analysis Questions and Tactics
7. Analysis OPSEC
8. Exercise 4.2: Structured Analysis Challenge
9. Intrusion Discovery
10. Incident Closing and Quality Review
11. Day 4 Summary
12. **Exercise 4.3: Collecting and Documenting Incident Information**

This page intentionally left blank.



Exercise 4.3: Collecting and Documenting Incident Information

Exercise 4.3:
Collecting and Documenting Incident Information

Exercise 4.3: Collecting and Documenting Incident Information

Please go to Exercise 4.3 in the SEC450 Workbook or virtual wiki.