SEC450 | BLUE TEAM FUNDAMENTALS: SECURITY OPERATIONS AND ANALYSIS

450.5

Continuous Improvement, Analytics, and Automation

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.

SEC450.5

Blue Team Fundamentals: Security Operations and Analysis

Continuous Improvement, Analytics, and Automation

Welcome to SANS Security 450.5 – Continuous Improvement, Analytics, and Automation

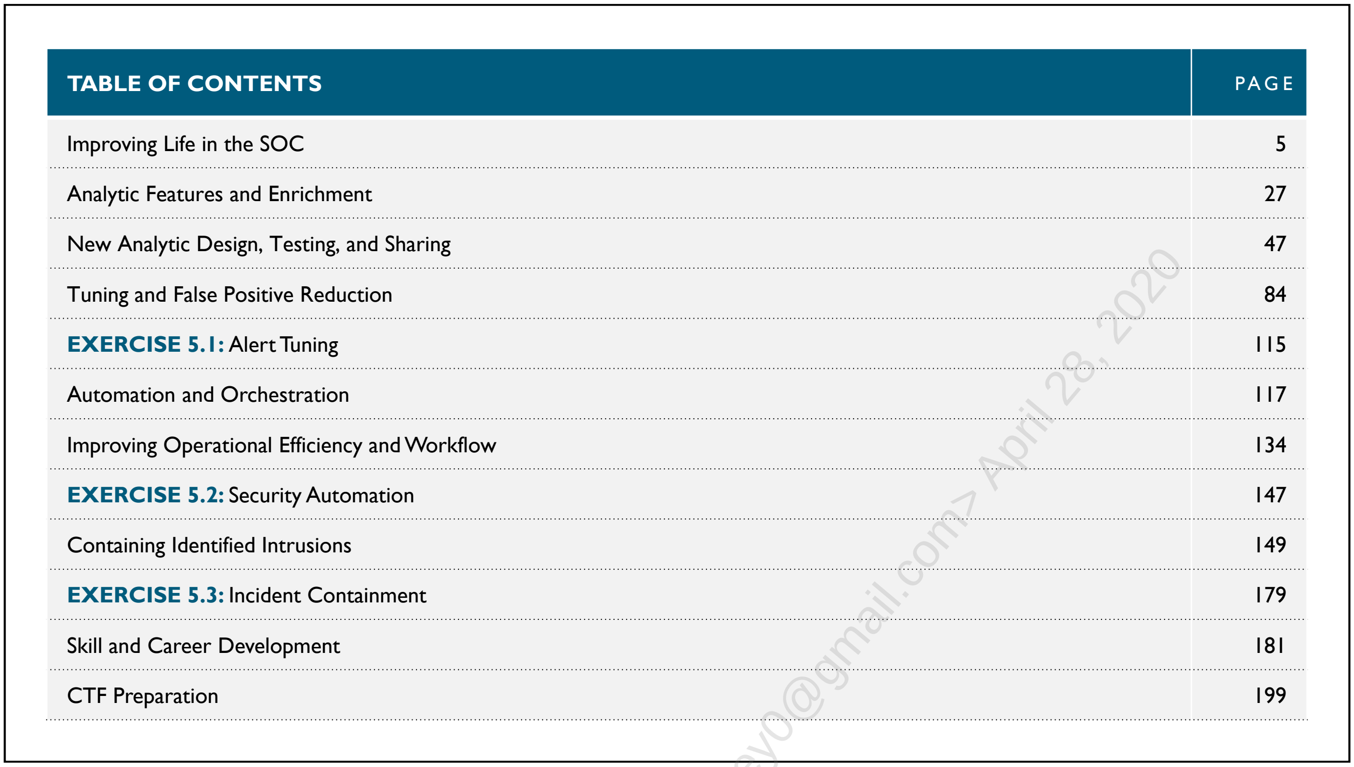
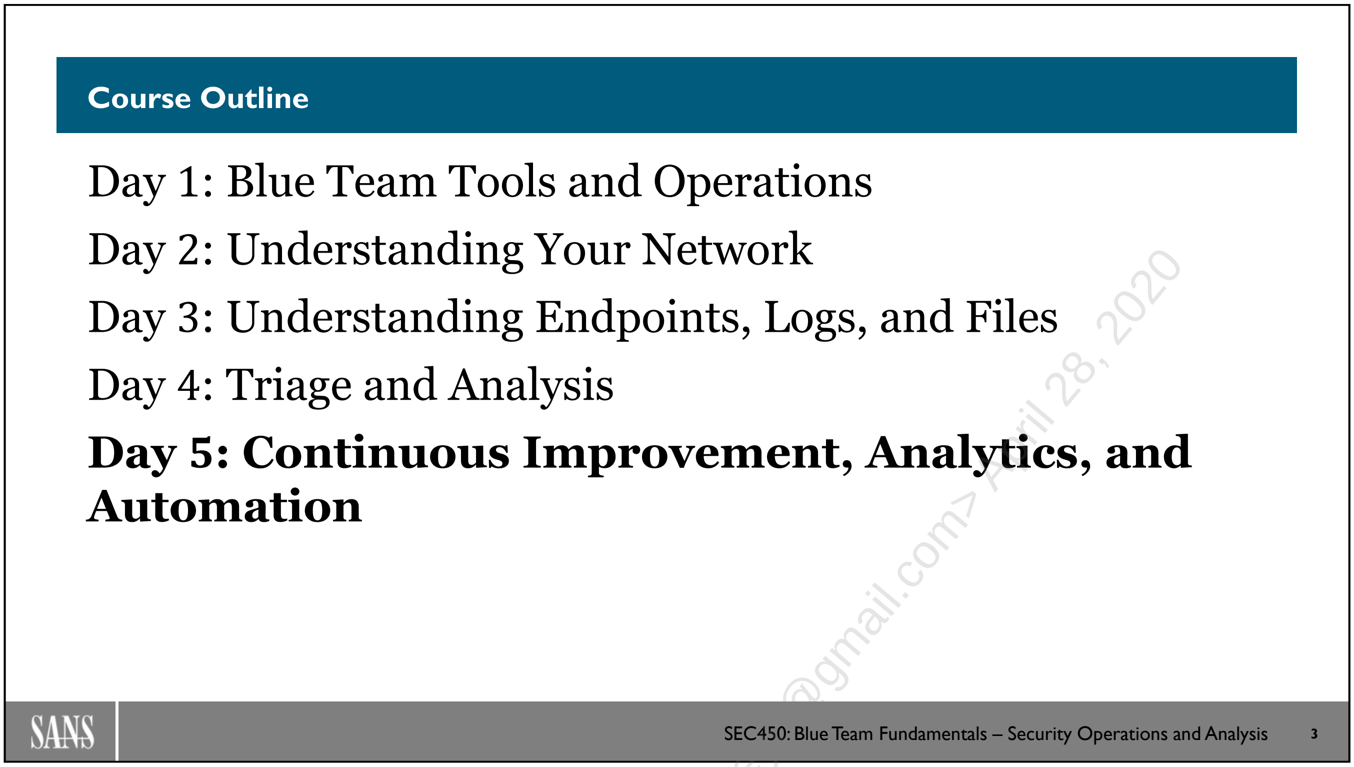


TABLE OF CONTENTS	PAGE
Improving Life in the SOC	5
Analytic Features and Enrichment	27
New Analytic Design, Testing, and Sharing	47
Tuning and False Positive Reduction	84
EXERCISE 5.1: Alert Tuning	115
Automation and Orchestration	117
Improving Operational Efficiency and Workflow	134
EXERCISE 5.2: Security Automation	147
Containing Identified Intrusions	149
EXERCISE 5.3: Incident Containment	179
Skill and Career Development	181
CTF Preparation	199

450.5 Table of Contents

This table of contents outlines the plan for 450.5.



Course Outline

Day 1: Blue Team Tools and Operations

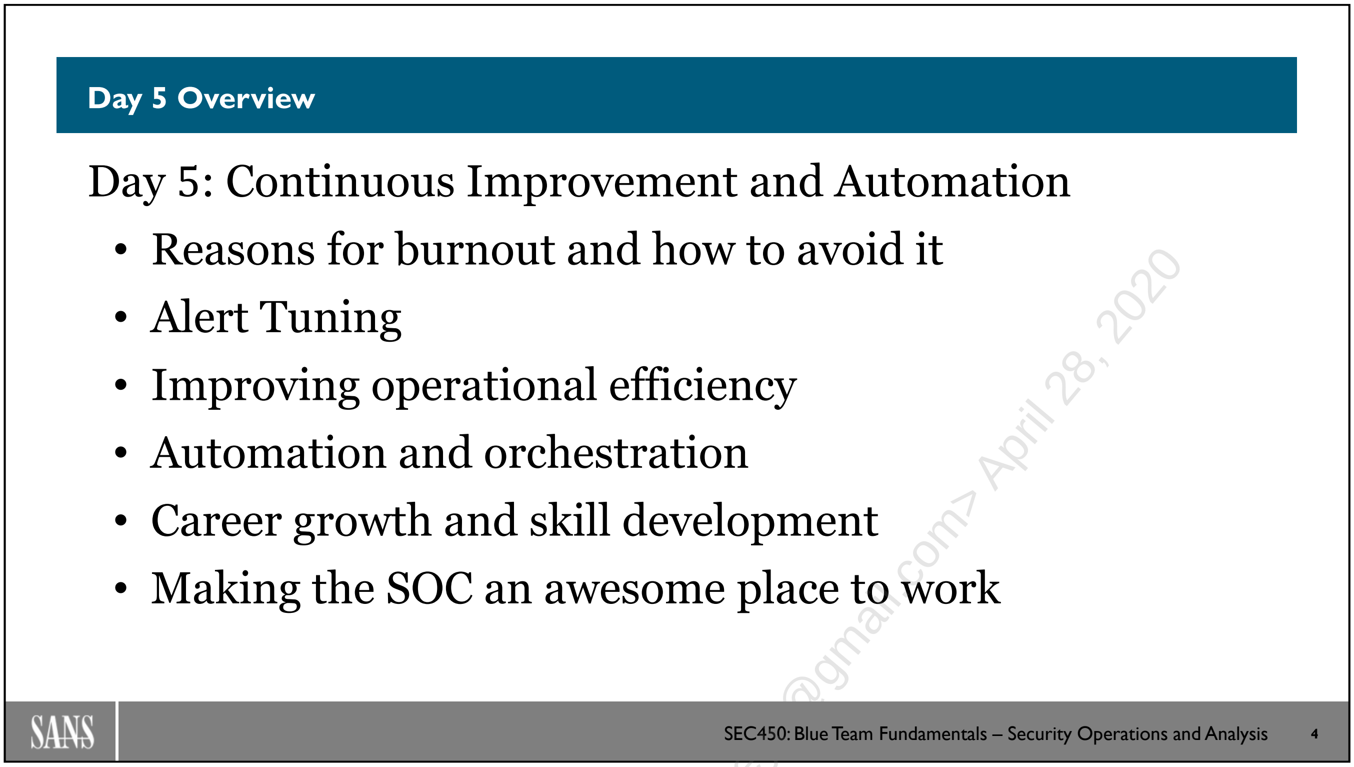
Day 2: Understanding Your Network

Day 3: Understanding Endpoints, Logs, and Files

Day 4: Triage and Analysis

Day 5: Continuous Improvement, Analytics, and Automation

This page intentionally left blank.

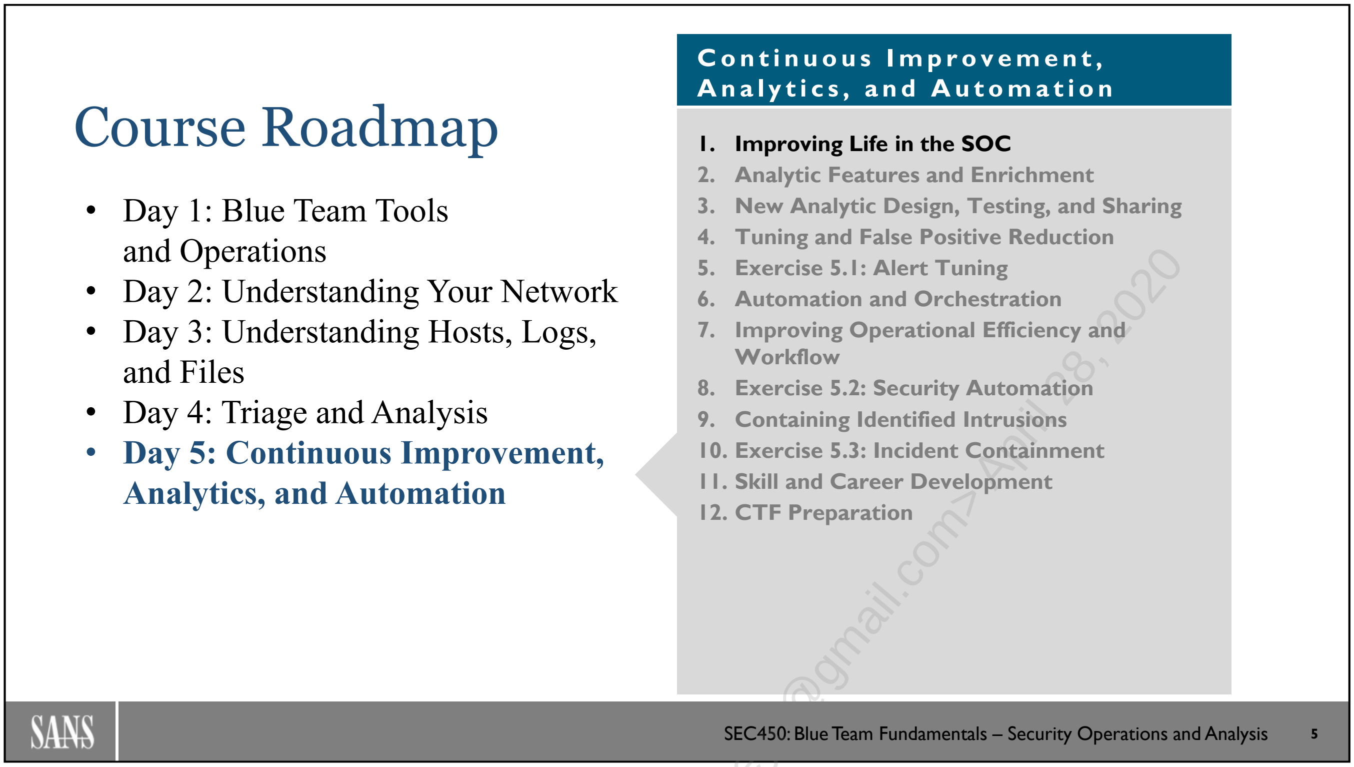


Day 5 Overview

Day 5: Continuous Improvement and Automation

- Reasons for burnout and how to avoid it
- Alert Tuning
- Improving operational efficiency
- Automation and orchestration
- Career growth and skill development
- Making the SOC an awesome place to work

This page intentionally left blank.



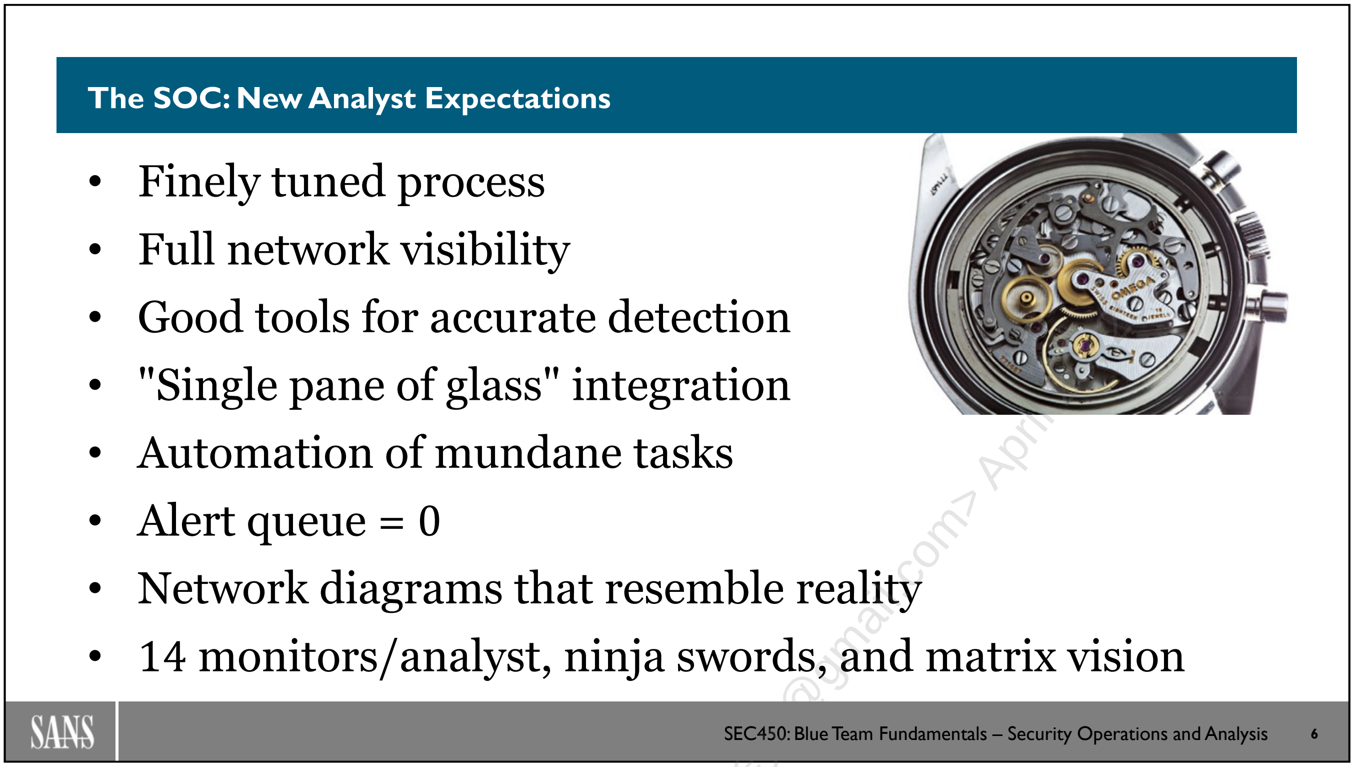
Course Roadmap

- 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**

Continuous Improvement, Analytics, and Automation

1. **Improving Life in the SOC**
2. Analytic Features and Enrichment
3. New Analytic Design, Testing, and Sharing
4. Tuning and False Positive Reduction
5. Exercise 5.1: Alert Tuning
6. Automation and Orchestration
7. Improving Operational Efficiency and Workflow
8. Exercise 5.2: Security Automation
9. Containing Identified Intrusions
10. Exercise 5.3: Incident Containment
11. Skill and Career Development
12. CTF Preparation

This page intentionally left blank.

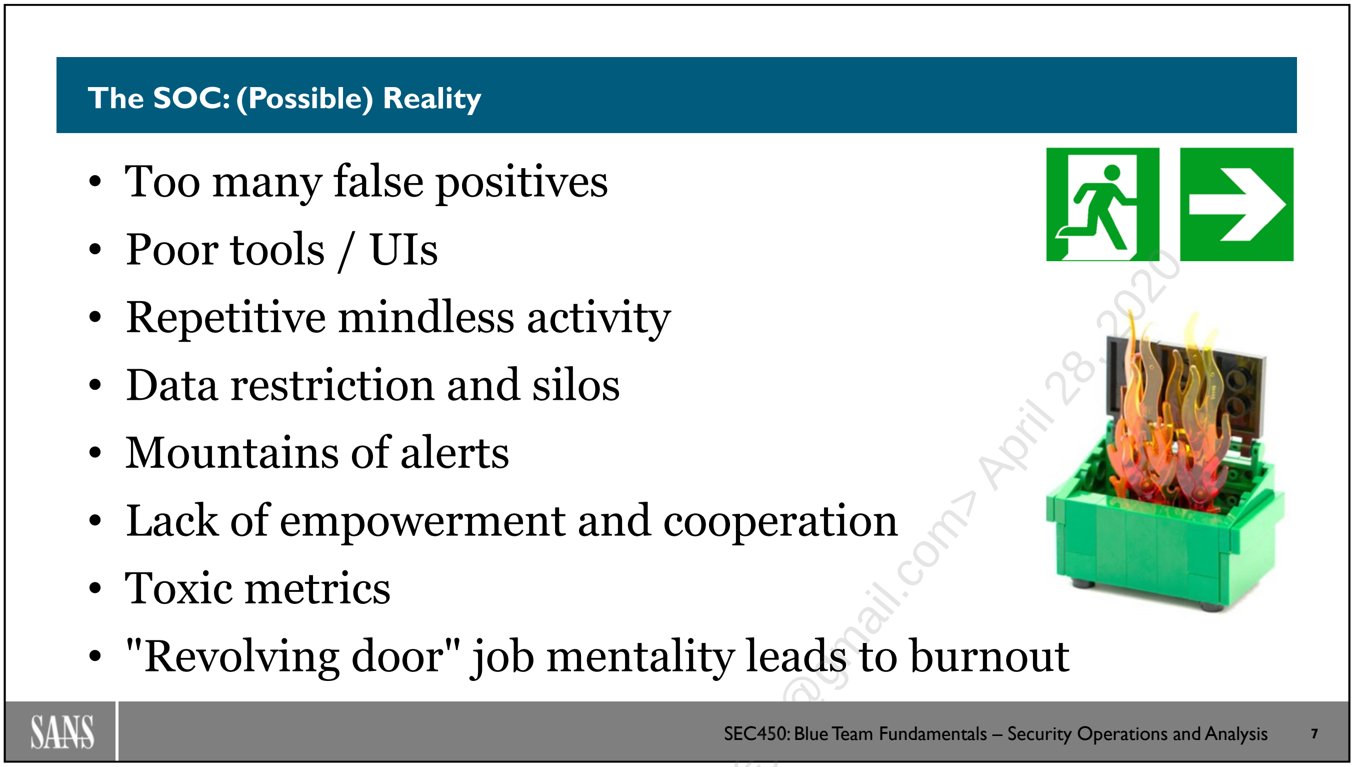


The SOC: New Analyst Expectations

- Finely tuned process
- Full network visibility
- Good tools for accurate detection
- "Single pane of glass" integration
- Automation of mundane tasks
- Alert queue = 0
- Network diagrams that resemble reality
- 14 monitors/analyst, ninja swords, and matrix vision

The SOC: New Analyst Expectations

You may have seen pictures of SOCs in movies, at conferences, or on YouTube before you joined one in real life. When you walked in on your first day, you probably had some reasonable expectations that they may be something like what you had been advertised. It would be reasonable to assume how put together they were, how well the tools and process worked, and you might have assumed they generally stayed on top of the alerts they received. In some SOCs, this is true. Unfortunately, this is not always the case. It takes a lot of time before many SOCs mature and find out what works best for them. In the best SOCs, data flow is highly dialed in, tools work together, the alert queue stays low while analysts stay engaged, learning and happy. These SOCs operate like a finely tuned watch, without too much or too little data, and able to effectively stay on top of their tasks most of the time. Although it can be hard to tell from the outside, some SOCs are quite the opposite.



The SOC: (Possible) Reality

- Too many false positives
- Poor tools / UIs
- Repetitive mindless activity
- Data restriction and silos
- Mountains of alerts
- Lack of empowerment and cooperation
- Toxic metrics
- "Revolving door" job mentality leads to burnout

The SOC: (Possible) Reality

Things are far from optimal in many SOCs. Even if you find a mature SOC with most processes and tools being well tuned, chances are high there's still one tool or workflow that you will highly dislike. Most SOCs will have some level of common issues:

- False positives
- Appliances that don't work with each other
- Inefficient process
- Data that doesn't flow well between tools
- Time pressure to complete alerts

In some cases, it gets so bad that a SOC can get the reputation as the place you do a "stint" in before you escape to another job. A job you need to "try to escape from" is clearly not going to be a fun place to work. Does it have to be this way? Compared to many jobs you could pick in this world, many SOCs offer a highly engaging, comfortable, positive environment geared toward those who like a challenge and lifelong learning. If your SOC environment has taken those benefits away by overshadowing them with negatives, it's time to make a change.

WE CAN REBUILD IT
WE HAVE THE TECHNOLOGY, BETTER, FASTER, STRONGER, THE SIX MILLION DOLLAR TEAM
imgflip.com

This Book's Focus: Making Things Better

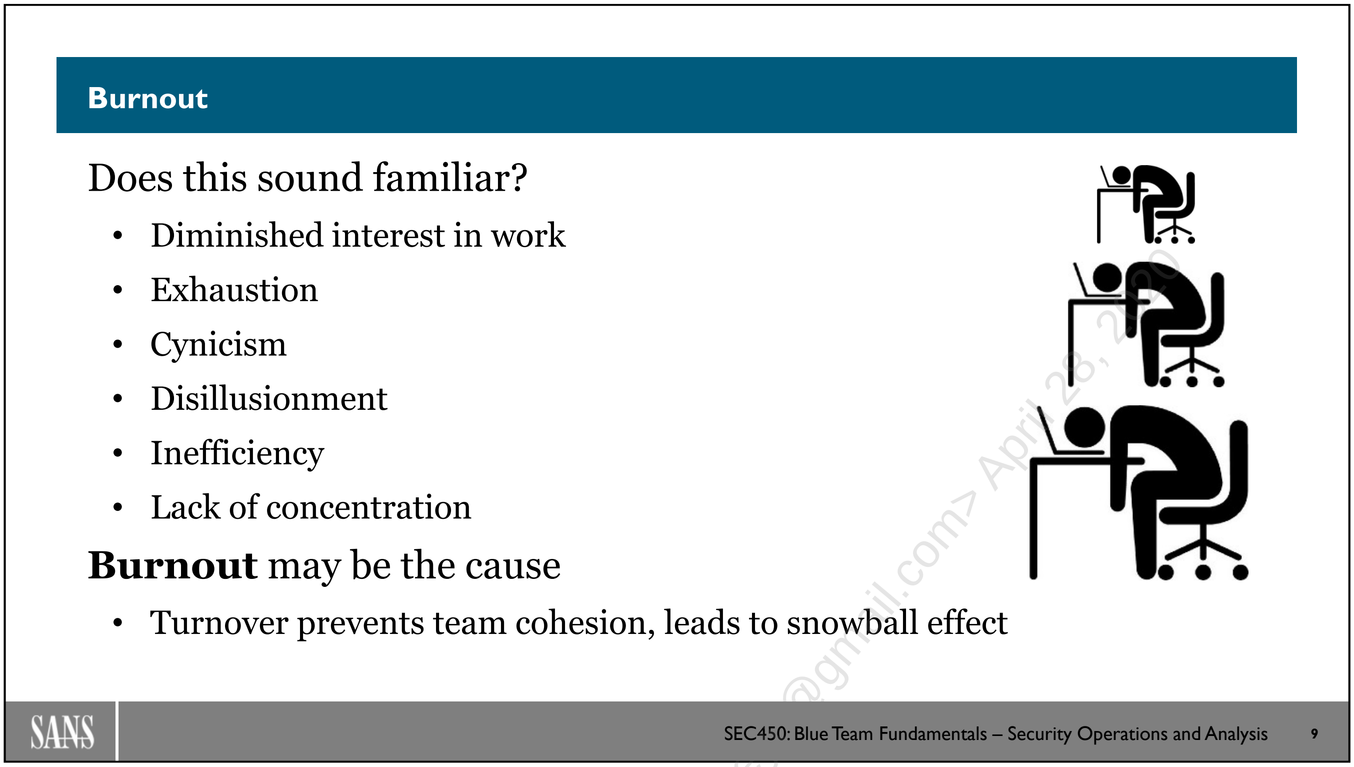
Things can be better, but it will take some effort

Today:

- What makes work stressful
- Causes of burnout in the SOC
- How do we fix it?
  - Alert tuning
  - Automation and Orchestration
  - Analyst empowerment
  - Incident Containment
  - Enabling creativity and growth

This Book's Focus: Making Things Better

Things will not get better without some work from you, however. To improve many of the common pain points of a SOC, it may take some extracurricular work: Learning how to script, how your data flow works, how your tools work (yes, this means potentially reading the product manual!), understanding their APIs, and digging into the depths of automation and orchestration tool capabilities. It will also take some cooperation from management, empowering and trusting analysts to come up with creative solutions to their problems. We didn't say it would be easy, but it *will* be worth it. Not only will going deeper on these tools make you the resident expert, further improving your value to your organization, but you will also learn a lot in the process. To top it all off, if all goes well, you'll also have eliminated the worst part of you and your fellow analyst's job, which they'll love you for doing. How awesome is that?



Burnout

Does this sound familiar?

- Diminished interest in work
- Exhaustion
- Cynicism
- Disillusionment
- Inefficiency
- Lack of concentration

Burnout may be the cause

- Turnover prevents team cohesion, leads to snowball effect

SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 9

Burnout

One of the biggest problems in SOCs is the extremely common issue of "burnout." According to HP, this is one of the main causes that lead to security analysts only staying in the job for 1-3 years before moving on to something else. Burnout causes us to make poor judgments, do an incomplete job, work inefficiently, and makes it hard to for us to concentrate. Furthermore, having our coworkers burn out and move on contributes to our likelihood of burnout. When team members are constantly coming in and out in a "revolving door" like situation, we never get the chance to bond with them and form a coherent team, which is one of the factors that can help improve and dissuade burnout. That means this situation is not only difficult to keep at bay, but it also feeds on itself, perpetuating the problem once it starts.

The clear message here is that we must architect our work environments to avoid burnout at all costs. Given that burnout tends to snowball and build on itself, the key to job satisfaction for yourself and others will be understanding the factors that lead to these feelings and avoiding them at all costs. Ideally, the improvements are supported and reinforced by management, but sometimes, we need to take our happiness into our own hands. In this book, we will go through the most prevalent factors that lead to burnout in a SOC so that you can understand them and avoid them. By taking alert design and tuning, automation, and containment into your hands, you can single-handedly make the SOC a better place to work.

Burnout and Stress Types

The two types of stress:

- **Good stress:** Perceived as things you can take on, challenges
- **Bad stress:** Persistent problems or things you feel you can't cope with or work around

Happy SOC: Learning new skills, challenging / creative problem solving

SOC Poison: Lack of control – Poor group cooperation, no visibility, no trust or empowerment to resolve issues, no time for reflection, false ceilings

Goal: Healthy amounts of good stress, elimination of bad stress

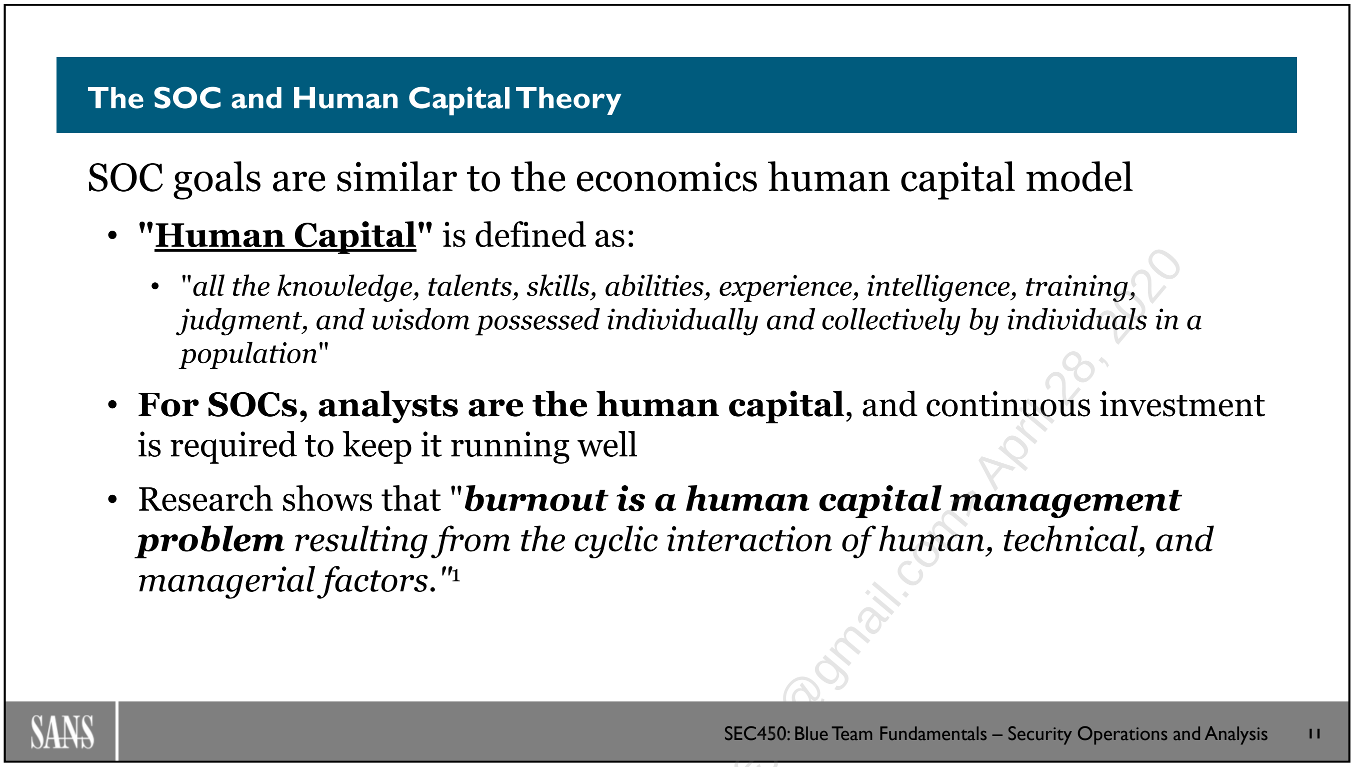
Burnout and Stress Types

While the word "stress" is usually used negatively, there is such a thing as good stress. Stress is not an emotion. It is a reaction to a stressor that causes adrenaline and anxiety. This *can* be bad in some situations, but in others it may be good, such as when it inspires us to get things done on time or take on new and difficult challenges. This type of stress is typically perceived as manageable and is not long-term persistent stress. Bad stress is the type that either won't go away or is so severe it cripples our ability to act, causing negative consequences. The ever-present looming of an uncontrollable situation is bad stress and is the type that can wear on people over time, eventually leading to burnout.

Another interesting fact about stress is that it is also tied to your perception. According to Psychology Today, "Stress is a perceived disconnect between a situation and our resources to deal with the situation. In other words, stress is a (real or imagined) threat that taxes our resources. The operative word here is perceived. Stress does not always arise from an actual threat; but if we perceive it to be a threat, then it's a threat."[1] Therefore, our stressors do not even have to meet any objective measure of dangerous; it is our perception of the stressor that is enough to give rise to the stress reaction. This means there are two main resolutions to reducing it: Removing yourself from the situation (changing it so you don't experience it in the first place), or changing your view about your ability to handle the given situation.

For SOCs, good stress can manifest itself as a challenging environment in which you are healthily pushed beyond your limits. You should have interesting problems to solve, be encouraged to learn and grow, and have an established capability and process to do so. Bad stress in a SOC often comes from the perception of the lack of control. Perhaps your SOC does not have enough authority or capability to cope with the issues it finds, struggles to get things done when communicating with other groups, or you have no time to take a breather to consider process improvement due to constant receiving alerts. These are problems that seem beyond your control in many cases, and with no apparent recourse to fix the situation, can become a persistent and toxic source of bad stress that leads to burnout.

[1] https://www.psychologytoday.com/us/blog/the-wide-wide-world-psychology/201601/why-stress-is-both-good-and-bad



The SOC and Human Capital Theory

SOC goals are similar to the economics human capital model

- "**Human Capital**" is defined as:
  - "*all the knowledge, talents, skills, abilities, experience, intelligence, training, judgment, and wisdom possessed individually and collectively by individuals in a population*"
- **For SOCs, analysts are the human capital**, and continuous investment is required to keep it running well
- Research shows that "***burnout is a human capital management problem*** *resulting from the cyclic interaction of human, technical, and managerial factors.*"[1]

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 11

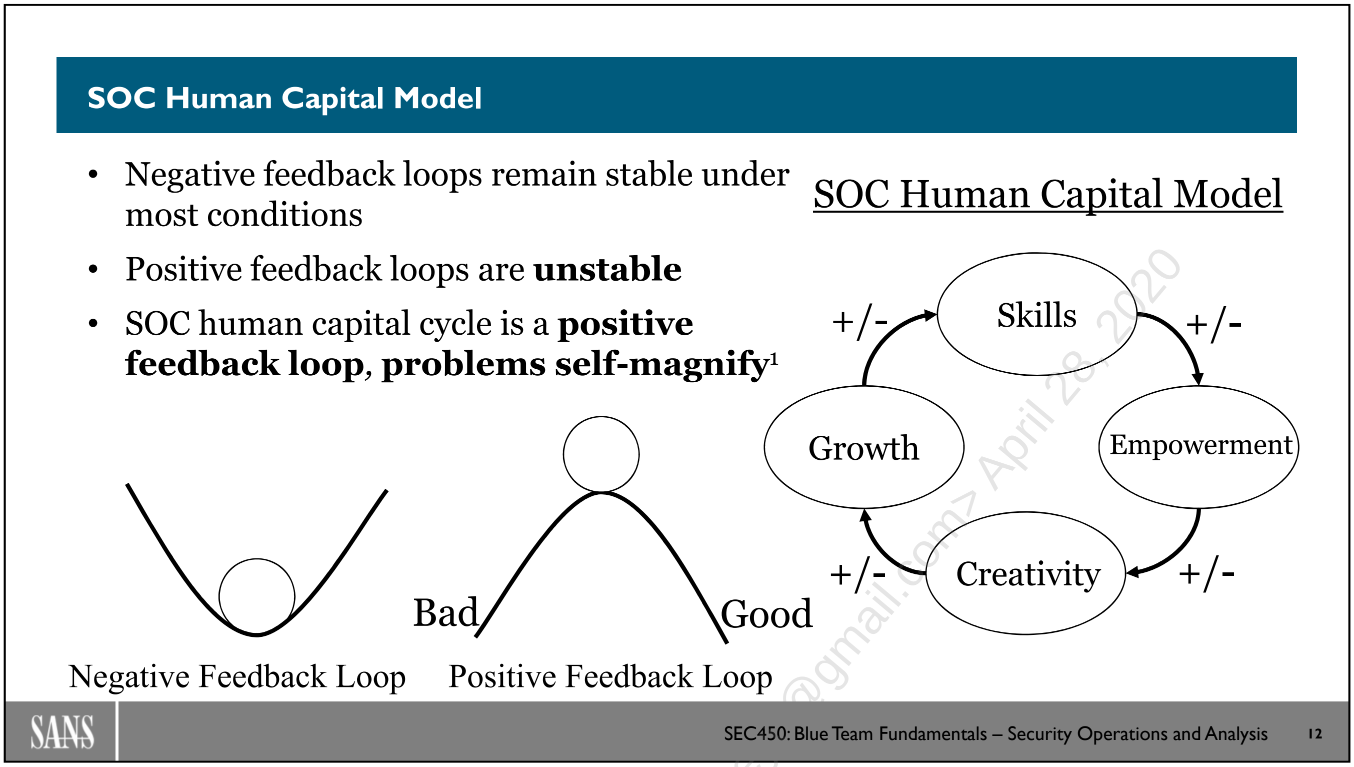
The SOC and Human Capital Theory

In a unique long-term research project funded by the National Science Foundation, Department of Homeland Security, and the Air Force Research Laboratory, a group of researchers performed an anthropological study on the culture of SOCs and what leads to the burnout phenomenon.[1] Their findings were published in the 2015 USENIX Symposium on Usable Privacy and Security in a paper titled "A Human Capital Model for Mitigating Security Analyst Burnout." After embedding one of the researchers within a large SOC taking daily detailed notes for six months, they concluded that burnout is a problem born of the mismanagement of "human capital." Human capital is an economic model popularized by Gary Becker, an economist at the University of Chicago and who has been described as "the most important social scientists in the past 50 years" by The New York Times.[1] Human Capital Theory's goal is to describe a collection of traits that express the capacity of a group of people to produce value, "all the knowledge, talents, skills, abilities, experience, intelligence, training, judgment, and wisdom possessed individually and collectively by individuals in a population." In the case of a SOC, analysts make up the human capital, and therefore a continuous investment is required to keep the group running efficiently.

In the study, using a sociological research methodology called "Grounded Theory," which is a way to construct a theory from qualitative data such as SOC observations, the researchers were able to come up with a human capital-based model for mitigating security analyst burnout.[2] Through this outstanding and enlightening research, we now have science-backed methods to describe how burnout occurs in analysts and how we can avoid it. Although their conclusions will likely not come as a surprise, one of the biggest benefits of this study was forming a model of the problem we can now use to *decompose* and *externalize* how a SOC functions, and the factors that lead to either burnout or job satisfaction. Over the next few slides, we'll explore this model as it will guide our discussions throughout the rest of the day and provide a science-backed framework for improving our own lives as well as the lives of our coworkers.

[1] https://www.nytimes.com/2014/05/06/upshot/how-gary-becker-transformed-the-social-sciences.html

[2] https://www.usenix.org/system/files/conference/soups2015/soups15-paper-sundaramurthy.pdf



SOC Human Capital Model

- Negative feedback loops remain stable under most conditions
- Positive feedback loops are **unstable**
- SOC human capital cycle is a **positive feedback loop, problems self-magnify**[1]

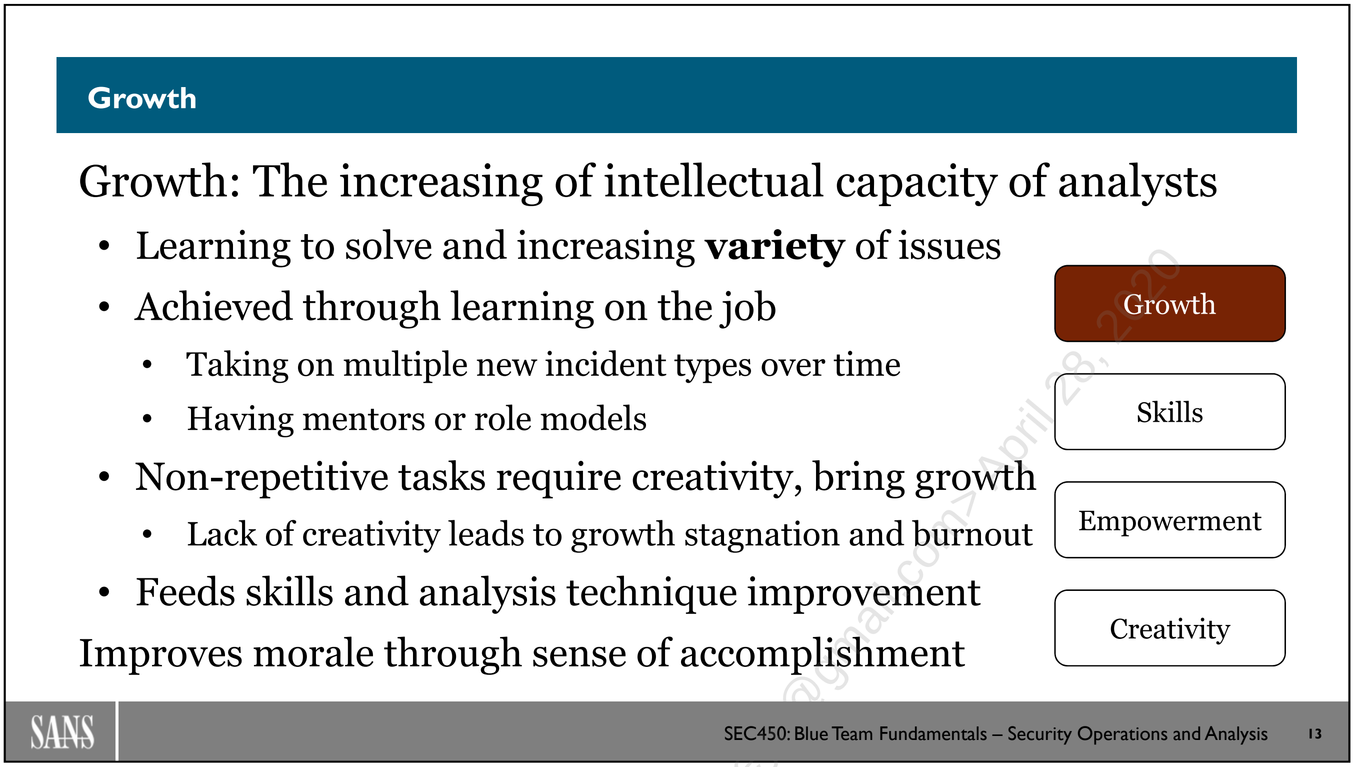
SOC Human Capital Model

The researchers identified four factors—skills, empowerment, creativity, and growth—as making up the SOC Human Capital Model, which influences the creation and maintenance of human capital in the SOC. For a SOC to thrive, you must cultivate these four items as the highest priorities. Since each item feeds into the next, these pieces form a positive feedback loop in that whatever direction one item goes, the next item is likely to follow (we use "positive feedback" in the engineering control theory sense here, not as in "good.") A positive feedback loop in our sense means that once the balance is disturbed in one direction, events following are likely to continue to speed up in that direction, such as a ball tipped off the top of a hill. This runaway effect is the opposite of a negative feedback loop that tends to take disturbances well and remain stable, like a ball in the bottom of a bowl.

Since the SOC human capital model was identified to be a *positive* feedback loop, we must be very careful to only push in the right direction. A disturbance of new policy, process, or technology can turn into either a *vicious cycle* where burnout and poor efficiency cause an accelerating downward spiral or *a virtuous cycle* in which the SOC reinforces a path toward continuous improvement. An example given in the research of a vicious cycle being born is management hiring entry-level, under-trained analysts due to budget constraints:

"These analysts will not be empowered enough as the managers do not trust the abilities of their analysts. This lower empowerment will lead to lower creativity, which in turn will lead to lower growth and skills. Since the skill level of analysts remains the same (very low), this will again lead to low empowerment, creativity, and growth. If this cycle continues, eventually the analysts will be burned out as they will start to feel that they are not accomplishing anything in their job—in other words, there is no growth, and the repetitiveness of the job exhausts the analysts."[1]

[1] https://www.usenix.org/system/files/conference/soups2015/soups15-paper-sundaramurthy.pdf



Growth

Growth: The increasing of intellectual capacity of analysts

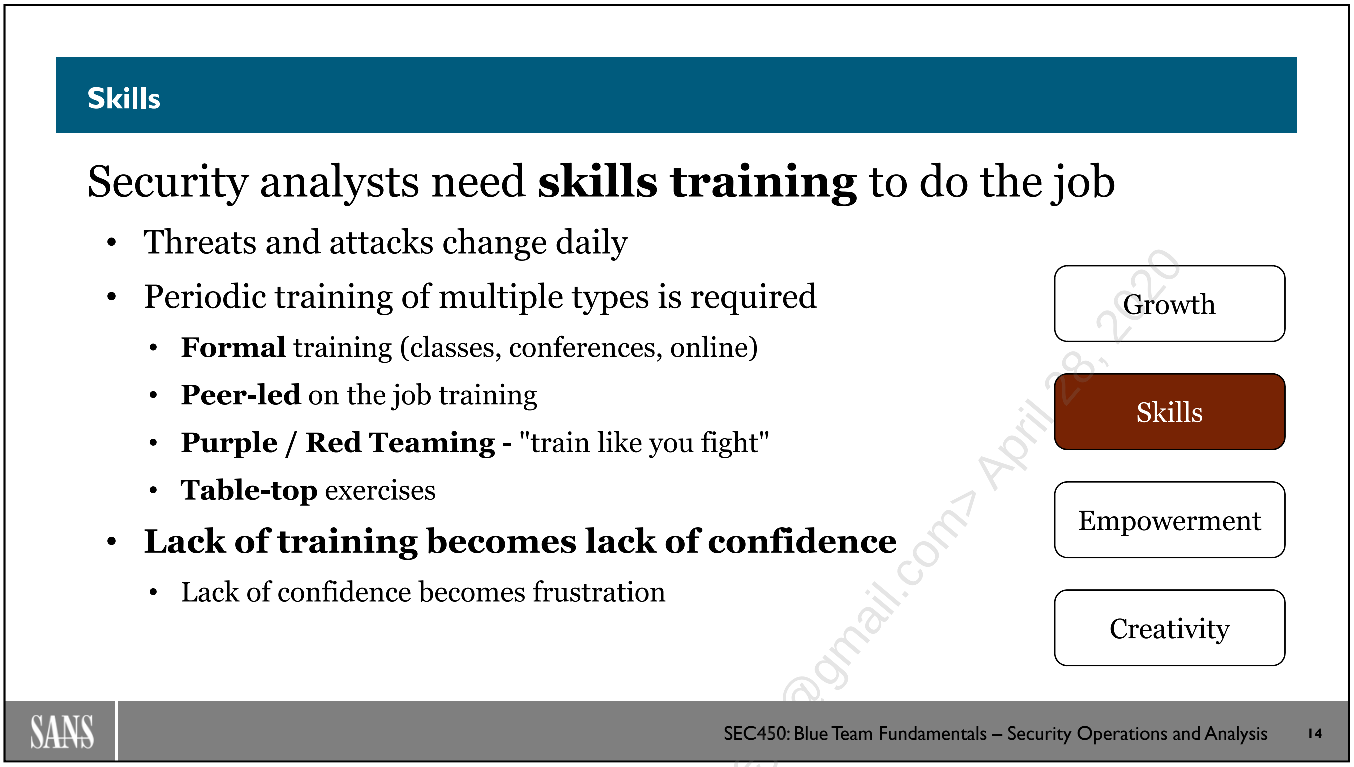
- Learning to solve and increasing **variety** of issues
- Achieved through learning on the job
  - Taking on multiple new incident types over time
  - Having mentors or role models
- Non-repetitive tasks require creativity, bring growth
  - Lack of creativity leads to growth stagnation and burnout
- Feeds skills and analysis technique improvement

Improves morale through sense of accomplishment

Growth

No one enjoys the feeling of stagnation and analysts are no exception. Therefore, it is important to ensure we feel that we're getting better over time to avoid burnout. Growth was defined by the researchers as the increasing intellectual capacity of an analyst as expressed by learning to handle *an increasing number of types of security incidents* over time. This growth should be taken on in multiple ways—by seeking mentors inside and outside of work and getting systematic feedback on how your analysis technique is progressing.

Growth is tied into the human capital cycle between creativity and skills because the ability to be creative feeds growth, and growth feeds new skill formation. Through avoiding repetitive tasks, an analyst is forced to use creative thinking to come up with solutions for the types of incidents they have not encountered before. Developing these new methods and seeing them succeed leads to a sense of purpose and accomplishment through this growth, improving morale. On the flip side, analysts who have no outlet for creativity may feel their growth has stagnated if they have not seen or accomplished anything new in a while. This growth, or lack thereof, will ultimately be reflected in analysts' skills over time and when things are improving, challenge analysts in a healthy way.



Skills

Security analysts need **skills training** to do the job

- Threats and attacks change daily
- Periodic training of multiple types is required
  - **Formal** training (classes, conferences, online)
  - **Peer-led** on the job training
  - **Purple / Red Teaming** - "train like you fight"
  - **Table-top** exercises
- **Lack of training becomes lack of confidence**
  - Lack of confidence becomes frustration

Growth

Skills

Empowerment

Creativity

SANS

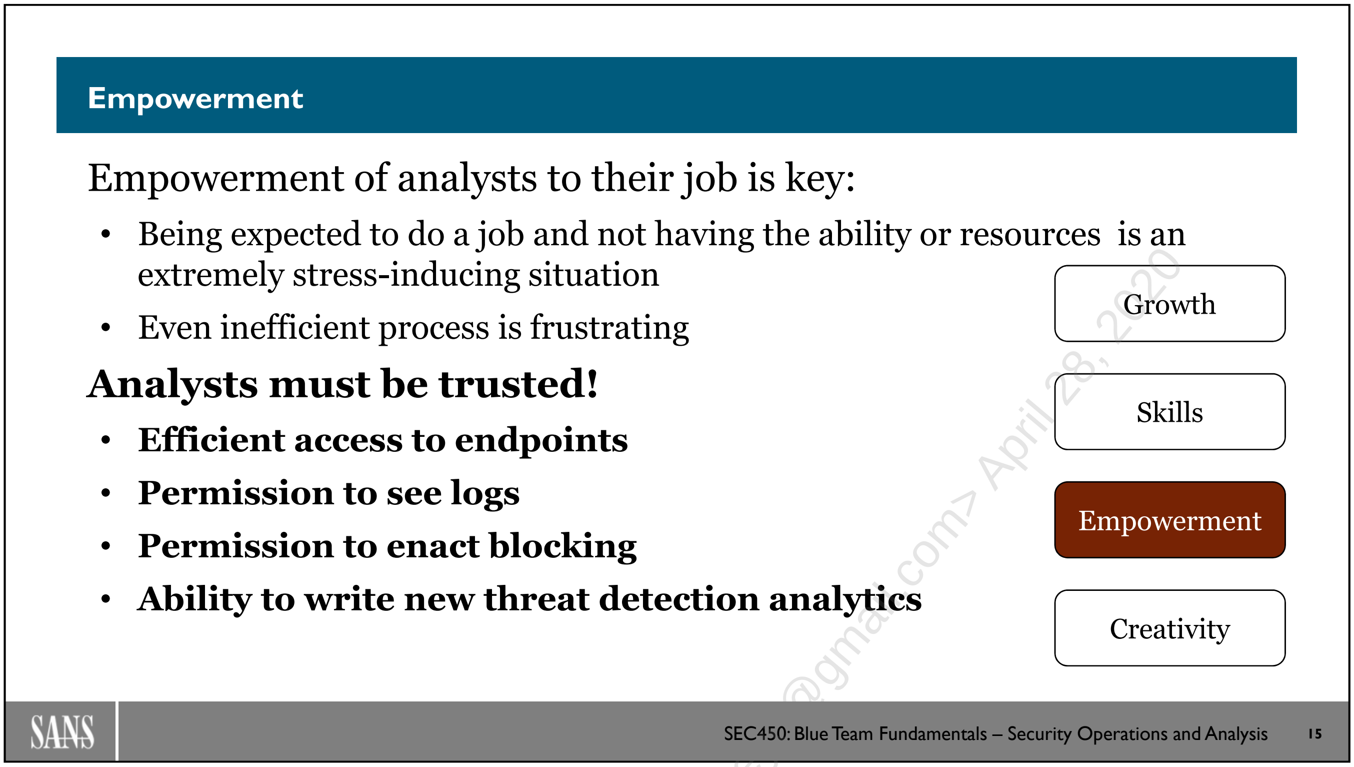
SEC450: Blue Team Fundamentals – Security Operations and Analysis 14

Skills

Skills training is a hot-button topic in many SOCs. It's one thing that analysts seem to constantly want more of but often fail to receive for various reasons. Without a doubt, training is a necessary part of the job. Not just for practical reasons, but also because it plays a large part in the human capital cycle. Mastery of a topic is one of the items that motivates many people in the information security industry, and if they do not feel they are gaining skill over time, they may start to succumb to burnout. Burnout happens because, without training, people lack confidence in what they are doing. The lack of confidence manifests as frustration and the inability or unwillingness to push the envelope in their daily lob, leading to frustration and low morale. With training, analysts gain skill and can earn the trust of management, which eases requests to be empowered to do more and more over time, making your job easier and your life happier.

Aside from human capital reasons, training is simply a necessity to keep up with the new attacks that are created daily. Although many of us love long-form, in-person training, keeping up with the times does not necessarily require travel, and some employers do not have the budget for it. Alternatively, online classes, capture the flag events, and conference videos posted to YouTube are a great secondary source of training that can be accessed anywhere. In addition, many cities have local conferences such as B-Sides that are very low cost and help connect people with the community in their area. We'll cover these sources in more detail later on in this book.

Lots of training can be performed on the job or at work, too, including mentor/peer-led training, purple/red teaming, and table-top exercises. Each one of these has its benefits. Purple teaming is one of the types that can be the most useful. Knowing a certain kind of attack is occurring and using your tools to identify it can be an outstanding way to learn how to catch the real thing. Red teaming can take this further to create an actual likely incident scenario, which will keep everyone trained and ready to react when the real thing occurs since they have already acted it out so many times. As they say, "train like you fight." If your mind has already gone through the motions of a simulated incident several times, the team will not run around scrambling for what to do when the real thing happens.



Empowerment

Empowerment of analysts to their job is key:

- Being expected to do a job and not having the ability or resources is an extremely stress-inducing situation
- Even inefficient process is frustrating

Analysts must be trusted!

- **Efficient access to endpoints**
- **Permission to see logs**
- **Permission to enact blocking**
- **Ability to write new threat detection analytics**

Growth

Skills

Empowerment

Creativity

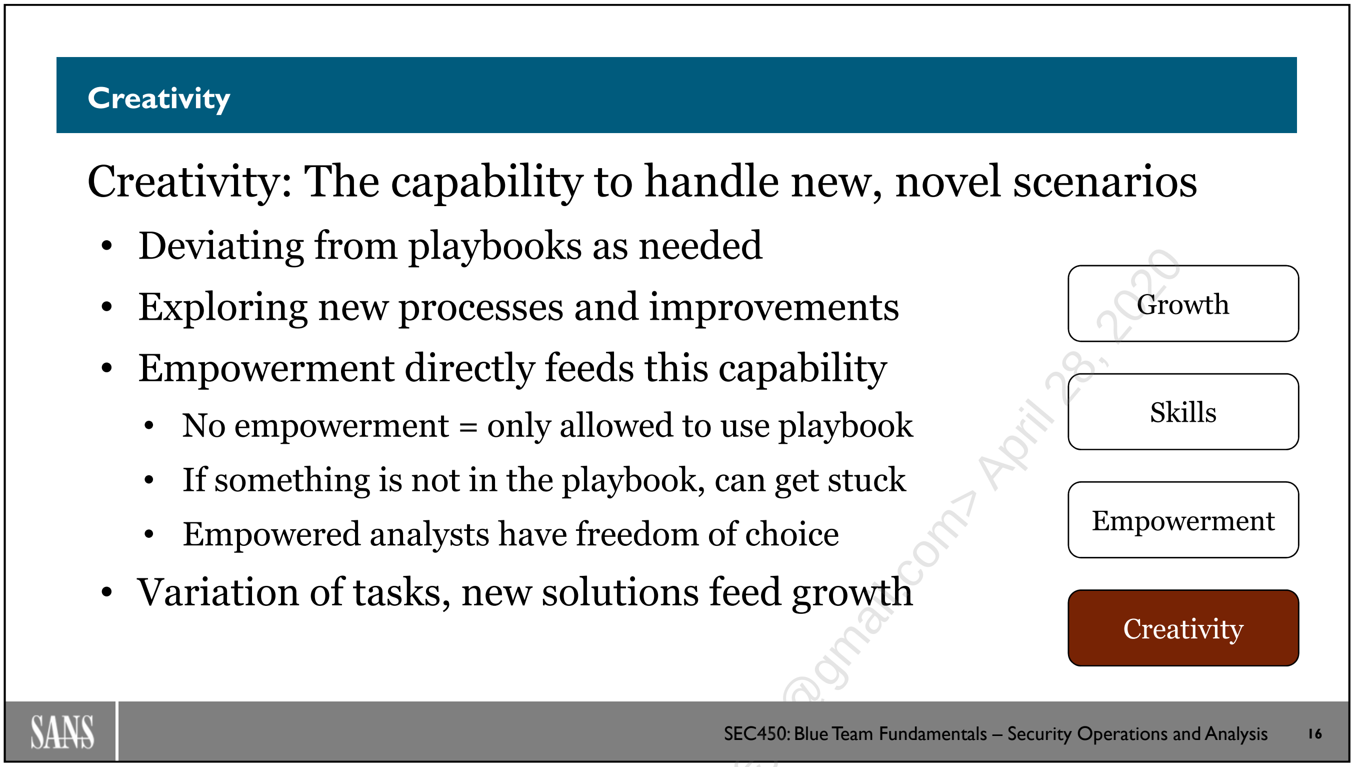
SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 15

Empowerment

Another common complaint from SOC analysts is the lack of empowerment. This can manifest itself in many ways but ultimately can be generalized as any speed bump in the way of getting the assigned job done. There's nothing more frustrating than being held accountable for doing a job at a certain rate of speed but also being denied the resources and access to do it. Even *perceived* inefficient processes could be enough to trigger stress from lack of empowerment.

The main problem with empowerment is the interplay between it and the previous item in the cycle—skills. Without skill, it's hard to justify why an analyst should be given the power and access that could potentially lead to mistakes or outages. While it is understandable and best practice to have the "least privilege" necessary for people to do their jobs, analysts must be trusted with certain powers to operate efficiently. How exactly this is done is environment dependent but, in general, analysts need efficient access to the systems they are defending. This includes permission to see logs related to alerts, permission to request or directly enable blocking actions, and the freedom to write new threat detection analytics. Although a phased approach aligned with skill may be needed to slowly ramp up permissions to make changes in the environment, empowerment is a key factor in morale, and its effects should not be ignored, especially with highly skilled analysts less likely to make a simple mistake. Empowerment is tied into the next factor of creativity because without being empowered, analysts may feel tied down and unable to be creative in their work.



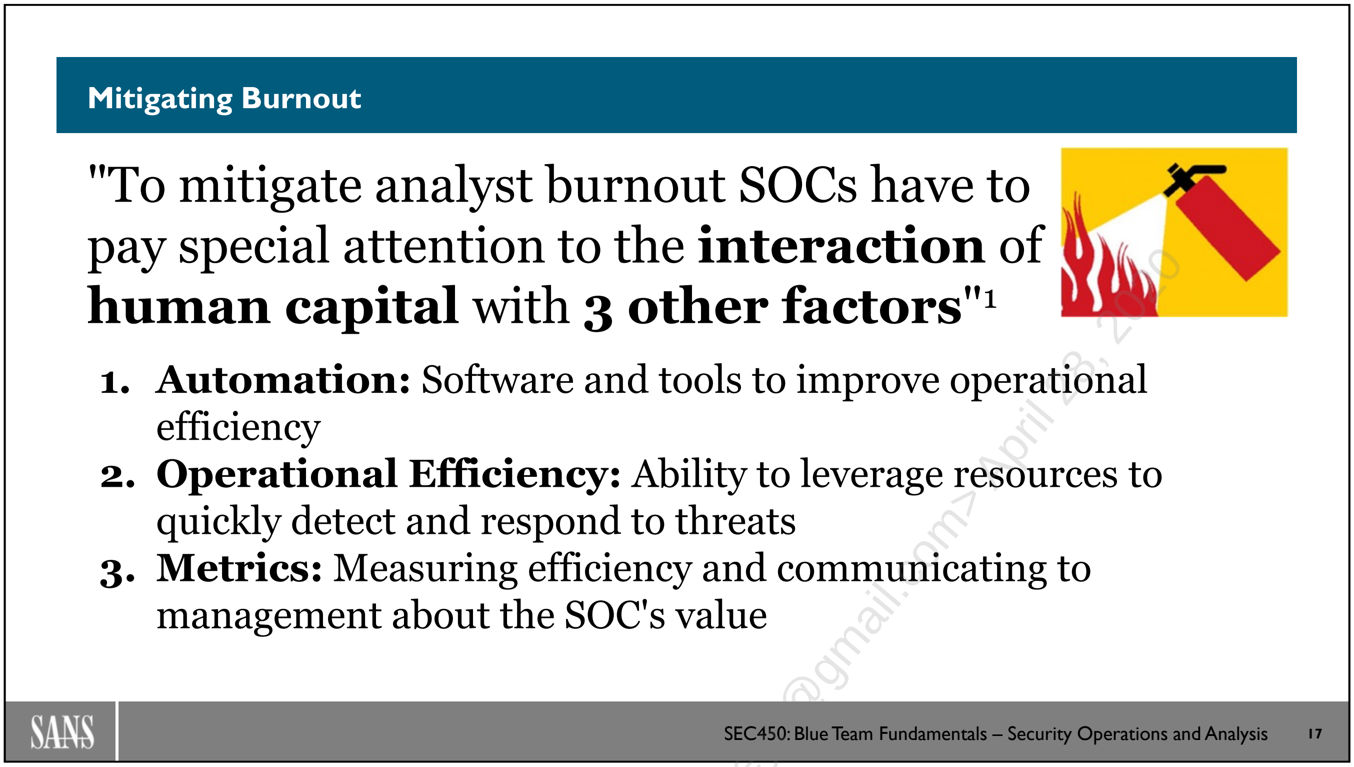
Creativity

Creativity: The capability to handle new, novel scenarios

- Deviating from playbooks as needed
- Exploring new processes and improvements
- Empowerment directly feeds this capability
  - No empowerment = only allowed to use playbook
  - If something is not in the playbook, can get stuck
  - Empowered analysts have freedom of choice
- Variation of tasks, new solutions feed growth

Creativity

The last step of the human capital to discuss is that of creativity. Creativity was defined as the *capability and freedom to handle new, novel scenarios in inventive ways*. Creativity is highly tied to empowerment because without the ability to deviate from playbooks and procedures in your daily job, your creative outlets will be severely restricted. Being stuck to a set procedure is a non-empowered state of affairs that brings creativity and morale down with it. Those who do have the freedom of choice to test out new and novel solutions to problems are by definition being creative, which then, in turn, allows them to grow, completing the cycle.



Mitigating Burnout

"To mitigate analyst burnout SOCs have to pay special attention to the **interaction** of **human capital** with **3 other factors**"[1]

1. **Automation:** Software and tools to improve operational efficiency
2. **Operational Efficiency:** Ability to leverage resources to quickly detect and respond to threats
3. **Metrics:** Measuring efficiency and communicating to management about the SOC's value

Mitigating Burnout

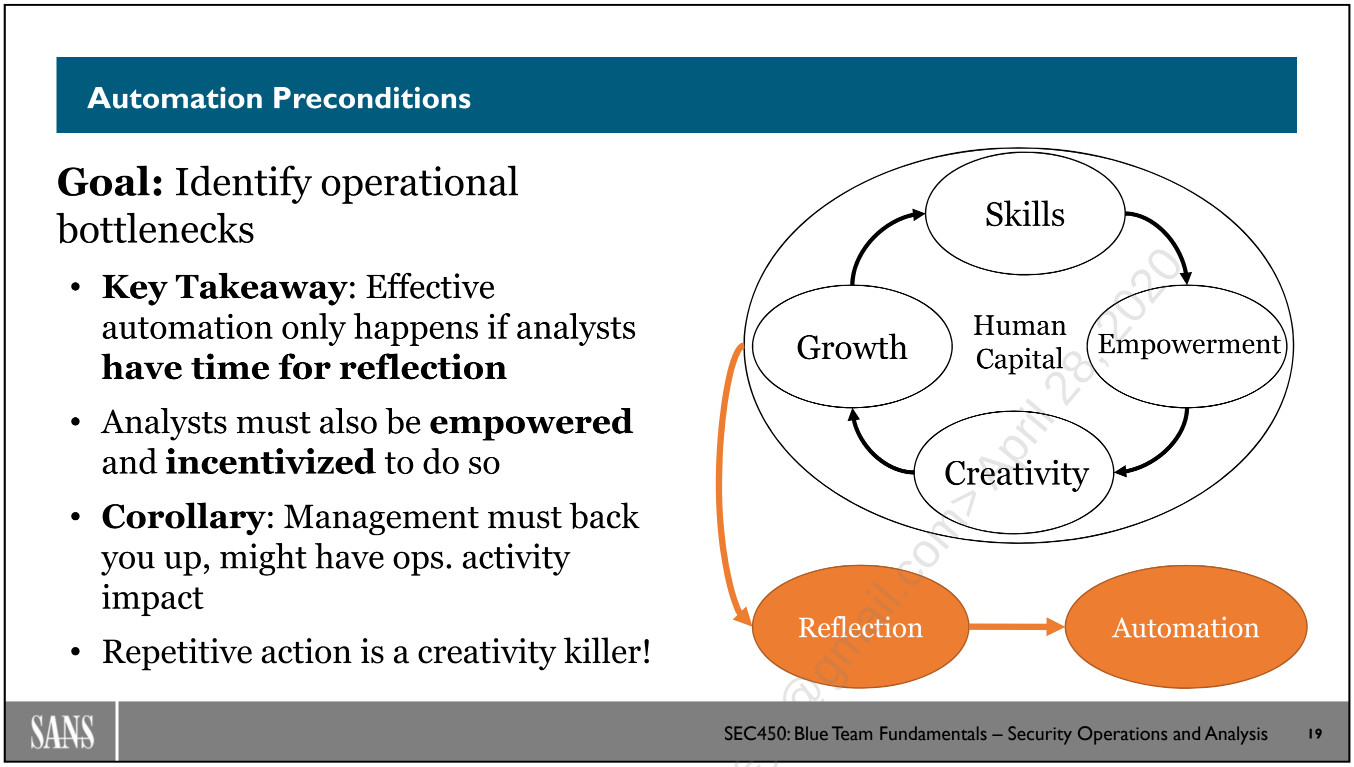
In addition to the four factors identified as part of the human capital, there are three additional external factors we must consider: Automation, Operational Efficiency, and Metrics.

Automation consists of all the tools we use in our daily lives that make threat detection, triage, and investigation faster and easier—everything from the SIEM down to the simple scripts you have running. The biggest benefit of these tools is that they remove repetitive tasks from our workload, and place them under the control of a computer, which is much better at it anyway. Unnecessary repetitive work leads to poor creativity and morale. Therefore, automation of everything possible is a huge boon for freeing up analysts to use their time for tasks better suited, and more interesting to humans.

Operational efficiency concerns the ability to use all available resources toward the job of detection and response. Since humans ultimately make the decisions and they are governed by the factors in the human capital model, it only makes sense that operational efficiency must affect AND be affected by human capital. Also, the automation capabilities of the group will have a direct impact on operational efficiency, performing tasks faster and more accurately than a manual process, and making analysts happier in the process. The implication here is that highly skilled and creative analysts will find ways to make operations more efficient, given automation tools and the capability to use them. Again, automation will form a positive feedback loop improving human capital as well, another virtuous cycle!

Metrics are one of the most complicated factors to nail down. Metrics are the link between analysts and management and should convey the value of what the SOC is accomplishing and the types of attacks it sees and is stopping, thus giving a clear value proposition to the business. They interact with human capital in that if the story is not told well or worse, if the metrics tracked tell the wrong story, analysts might find themselves in a position with less budget, less empowerment, or worse. Less budget means less training for the SOC. Less training, combined with restricted empowerment, means the growth factor of the human capital model veers into a vicious cycle from which it is hard to recover. Overbearing metrics tracking can also affect operational efficiency and employee morale, which can drive down creativity. As you can see, all these factors create a delicate balance that is hard to maintain, and even harder to fix when things do or have gone wrong in the past.

Throughout the rest of the book, we will dive deeper into some of the items that can be major drivers of the human capital cycle—alert design and tuning, automation, and containment. This way, we can start to get a grasp on the problem, recovering and reversing the cycle of burnout, and bringing the SOC back into a virtuous cycle of improvement.



Automation Preconditions

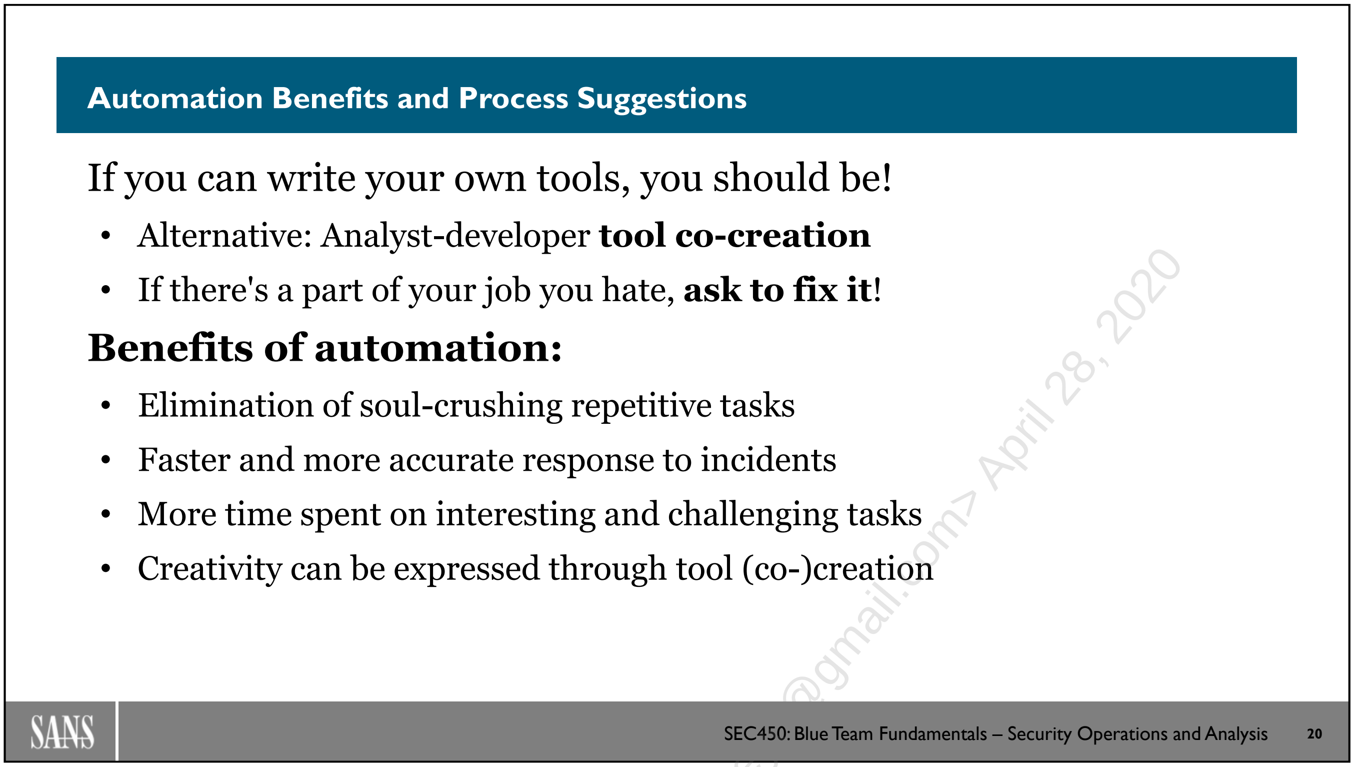
Goal: Identify operational bottlenecks

- **Key Takeaway**: Effective automation only happens if analysts **have time for reflection**
- Analysts must also be **empowered** and **incentivized** to do so
- **Corollary**: Management must back you up, might have ops. activity impact
- Repetitive action is a creativity killer!

Automation Preconditions

So, what do we need to get the state of automation we desire? The goal of automation is to remove operational roadblocks, but without time to sit back and consider where to best focus your energy, this effort will never get off the ground. Therefore, according to the research, effective automation can only take place if analysts are *given time for reflection* and can use that time to come up with the best plan of action to make their jobs better. If 100% of the day is spent fighting fires, this reflection that ultimately turns into automation can never occur. This leaves analysts with only years of repetitive action in their future—not a good prospect.

To ensure automation occurs in the SOC, analysts must be empowered and incentivized to spend time dreaming up and implementing automation. Structured brainstorming as a group can be of great benefit here as the group can come to a consensus about what the biggest problems are and the best ways to solve them. The SOC should do procedure review periodically and consistently look for these opportunities. Simultaneously, it should be giving analysts time to pursue the opportunities, even if it comes at the short-term cost of operational efficiency. The long-term payback of having a manual process become automated will more than outweigh the small backlog you may develop while someone takes a painful creativity-killing process off the table forever. In the worst-case scenario, you may have to do this work off hours to prove there is a return for the time invested. Of course, if you take this route, this effort should not go unmentioned once performance review time comes around!



Automation Benefits and Process Suggestions

If you can write your own tools, you should be!

- Alternative: Analyst-developer **tool co-creation**
- If there's a part of your job you hate, **ask to fix it**!

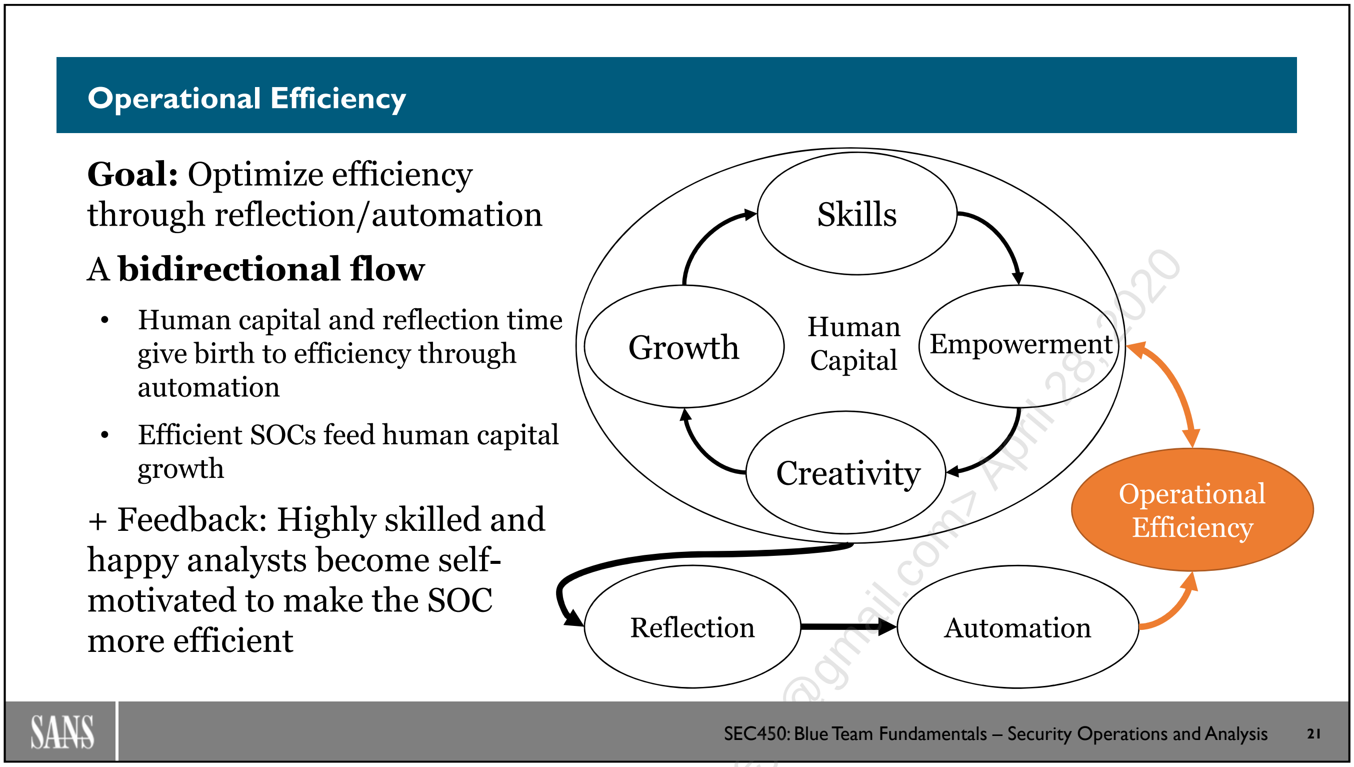
Benefits of automation:

- Elimination of soul-crushing repetitive tasks
- Faster and more accurate response to incidents
- More time spent on interesting and challenging tasks
- Creativity can be expressed through tool (co-)creation

Automation Benefits and Process Suggestions

The takeaway here is that everyone should attempt to squeeze some time for automation of tasks into their schedule. Although most managers will likely see the benefits, some work environments may be operated under strict circumstances that do not allow for this. In these cases, the researchers suggest the process they call "analyst-developer tool co-creation", a process where analysts write their process down in detail, draft a requirements list and pass it on to a software developer for creation. Once the first draft comes back, it can be tested and modified as necessary. This process gives analysts a creative outlet without having to pull them from operational activities.

The high-level benefits of automation should be clear at this point. Automation first and foremost removes some of the most painful parts of the job for an analyst. It eliminates the dreaded soul-crushing repetitive tasks that cause people to re-evaluate their decision to join a SOC and turns them into something that is done instantly and accurately via software. This frees up the people in your SOC to pursue the activities and investigations that challenge them and engage their minds, feeding positivity into the human capital model instead of negativity.



Operational Efficiency

Goal: Optimize efficiency through reflection/automation

A **bidirectional flow**

- Human capital and reflection time give birth to efficiency through automation
- Efficient SOCs feed human capital growth

+ Feedback: Highly skilled and happy analysts become self-motivated to make the SOC more efficient

Skills

Empowerment

Creativity

Growth

Human Capital

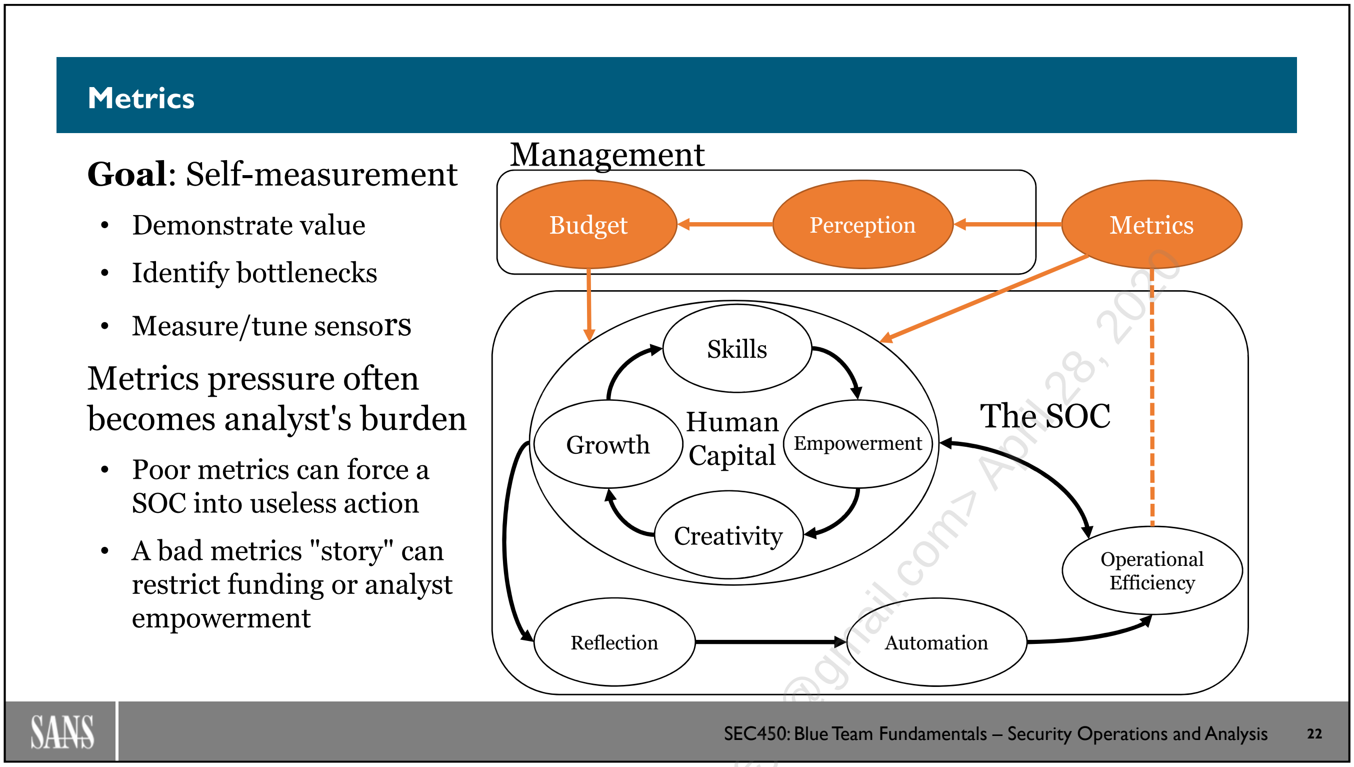
Reflection

Automation

Operational Efficiency

Operational Efficiency

The main feeder of the next external factor, Operational Efficiency, is Automation. The ability of analysts to reflect and automate processes leads directly to increased operational efficiency. Operational efficiency is the ultimate goal, as it is the state where all resources available are leveraged to quickly investigate and respond to incidents. The important point about operational efficiency is keeping it at a high level. Since it feeds back into the human capital of the SOC, it directly feeds the ability to accrue happy analysts. The researchers found that operational efficiency is not a one-way street. Although it is primarily fed through automation, there is also the effect that highly motivated, skilled and empowered analysts simply work more efficiently due to being more highly engaged. This is *another* bidirectional positive feedback loop connecting human capital, automation, and operational efficiency in an even bigger virtuous cycle!



Metrics

Goal: Self-measurement

- Demonstrate value
- Identify bottlenecks
- Measure/tune sensors

Metrics pressure often becomes analyst's burden

- Poor metrics can force a SOC into useless action
- A bad metrics "story" can restrict funding or analyst empowerment

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 22

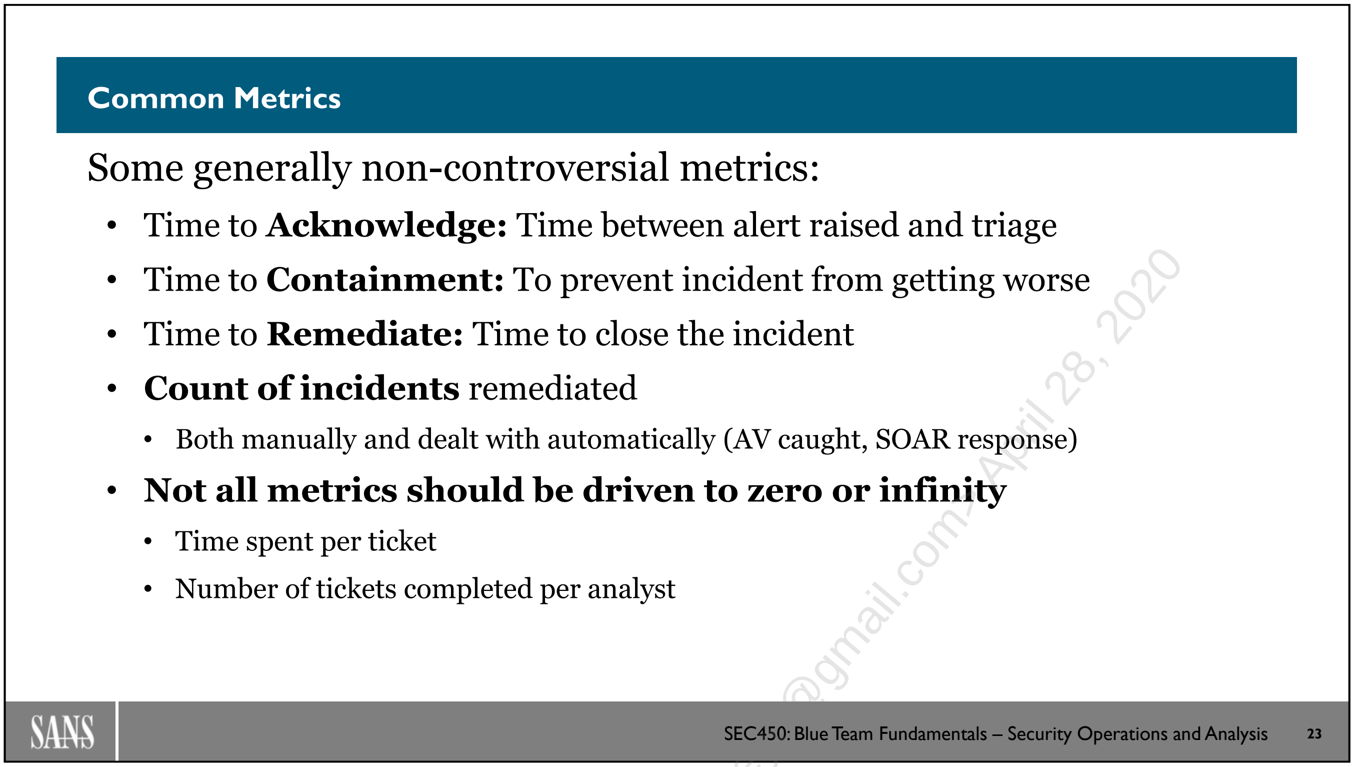
Metrics

SOC metrics are one of the most discussed topics in information security, precisely because they are just so difficult to get right. Metrics are often locally divided into "internal" and "external" metrics, numbers that will be used within the group or outside the group. External metrics should at a high level demonstrate the value the SOC brings to the organization as well as demonstrate the threats it faces. Internal metrics should be a feedback mechanism to the SOC allowing it to tune detection methods and measure itself for continuous improvement.

We must strike a fine balance—measuring many parts of the SOC may have a cost of analyst time and effort, which takes away from the main mission. Over and over, we hear stories about ticket management systems where an overbearing number of fields must be filled out for each alert, slowing analysts down and ruining their day with repetitive, mindless action. While some metrics can be derived automatically from SOC tools, automated metrics do not always allow for the nuanced situations of the real world and may blur what is going on through unclear classification or the inability to correctly time important points within the alert life cycle. Given that metrics are the face of the SOC that is presented to management to make decisions, analysts are rightly sensitive to make sure they are telling the correct story and accurate in their sampling.

Due to the control the metrics have on the outcome of the SOC and its individuals, you can imagine how it would directly affect the human capital aspect as well as the SOCs operational efficiency. While a good story can prop up the SOC and increase funding, metrics that don't tell the right story can frustrate or confuse management to the point where funding or empowerment can be taken away, leading the SOC down a dark path. Given this, the best way to handle metrics as an analyst is to accept them a necessary part of the job, but also push back if they become over-burdensome, inaccurate, or start driving behavior "for the metrics" instead of doing what is right. If any of these conditions are identified, it's best to hit the stop button as fast as possible and let management know the concerns so they can be worked out instead of letting the group get angry about it and bringing down morale. As Carson Zimmerman said in his 2018 SOC Summit presentation, "Metrics are like lightsabers, they can be used for good or evil."[1]

[1] https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1532960745.pdf



Common Metrics

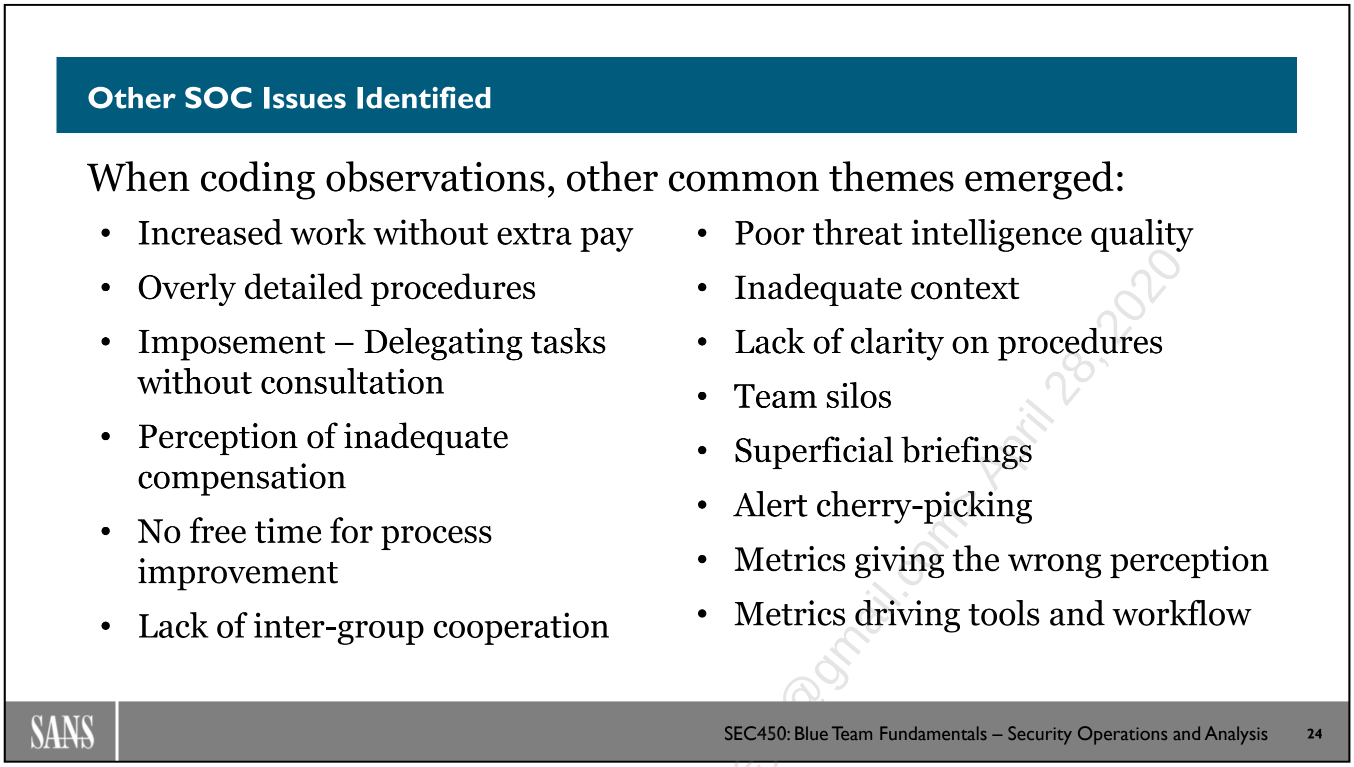
Some generally non-controversial metrics:

- Time to **Acknowledge:** Time between alert raised and triage
- Time to **Containment:** To prevent incident from getting worse
- Time to **Remediate:** Time to close the incident
- **Count of incidents** remediated
  - Both manually and dealt with automatically (AV caught, SOAR response)
- **Not all metrics should be driven to zero or infinity**
  - Time spent per ticket
  - Number of tickets completed per analyst

Common Metrics

Although there are an infinite set of possible metrics that can be dreamt up, the few listed on the slide above are great places to start as they are commonly used and tend to work non-controversially. As with any metric, ask yourself what will happen to your work as you try to drive it in the "better" direction. If the answer is "you will be tempted to do sloppier and less-thorough work to satisfy the number," take caution. Metrics such as "time to acknowledge, contain, and remediate" are generally safe as they are not judging the quality of the investigation at hand but more so the queue of work and response tempo of the SOC. Count of incidents is usually another safe metric as well, although it may need to be broken into incidents that had to be manually handled vs. issues that were primarily dealt with by automation and security tools. For example, you could say 500 viruses were identified in our systems, but only 10 of them needed manual action and remediation. The others were identified and deleted/prevented from running. Many metrics may need qualifying statements to ensure the content they're trying to express is clear.

One final statement about metrics, especially ones that involve a judgment of time and quality either directly or indirectly: Be very careful, as not all metrics should be driven to zero or as high as possible. Many SOCs measure tickets completed per analyst and time spent per ticket, for example. If analysts believe they are being judged on these numbers being as high/low as possible, it will be tempting to perform less thorough work, leading to improper remediation, and an overall bad situation for the SOC. If your SOC tracks these timing numbers, analysts should understand that they are for finding long-term averages only and that they shouldn't be driven to the maximum or minimum value in a way that will compromise the true mission of defense in the pursuit of metrics.



Other SOC Issues Identified

When coding observations, other common themes emerged:

- Increased work without extra pay
- Overly detailed procedures
- Imposement – Delegating tasks without consultation
- Perception of inadequate compensation
- No free time for process improvement
- Lack of inter-group cooperation
- Poor threat intelligence quality
- Inadequate context
- Lack of clarity on procedures
- Team silos
- Superficial briefings
- Alert cherry-picking
- Metrics giving the wrong perception
- Metrics driving tools and workflow

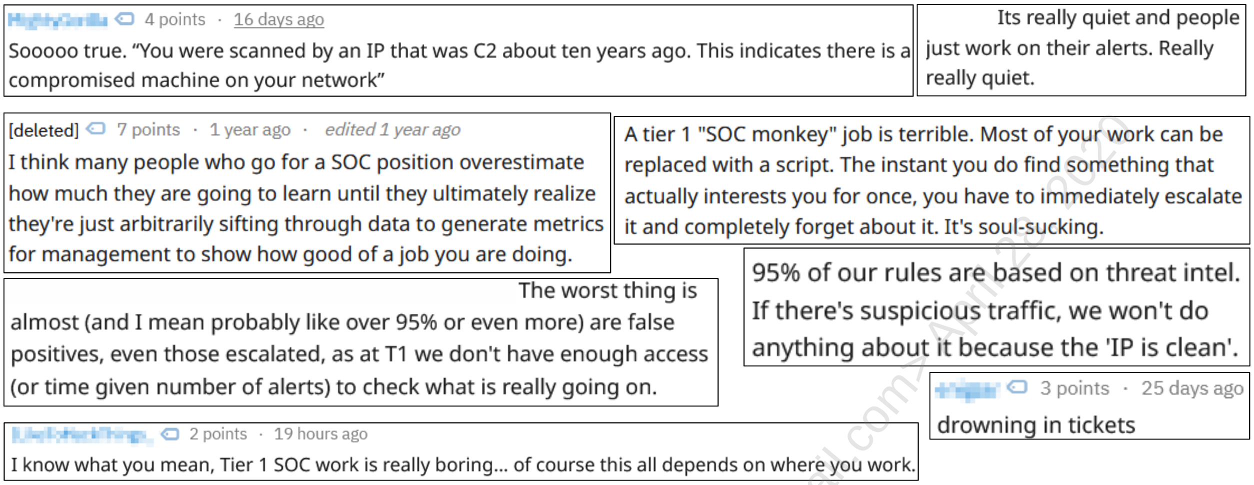
SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 24

Other SOC Issues Identified

Going into additional specifics—of the 85 pages of observations the researchers took and coded into groups, many similarities appeared. The slide lists some of the common themes and complaints that were heard across the six months they spent in the environment. These items should come as no surprise but may give you comfort in realizing that many organizations have the same problems. It is by no means an easy or self-evident thing to cultivate a positive, continuously improving SOC environment. The good news is that all these items fall under one of the higher-level categories we have already discussed, so an improvement in those top-level factors will likely encourage an improvement in these items as well. Throughout this book, we will dive into several of these specific topics as well.

Additional Comments Care of Reddit



[redacted] 4 points · 16 days ago
Sooooo true. "You were scanned by an IP that was C2 about ten years ago. This indicates there is a compromised machine on your network"

Its really quiet and people just work on their alerts. Really really quiet.

[deleted] 7 points · 1 year ago · *edited 1 year ago*
I think many people who go for a SOC position overestimate how much they are going to learn until they ultimately realize they're just arbitrarily sifting through data to generate metrics for management to show how good of a job you are doing.

A tier 1 "SOC monkey" job is terrible. Most of your work can be replaced with a script. The instant you do find something that actually interests you for once, you have to immediately escalate it and completely forget about it. It's soul-sucking.

The worst thing is almost (and I mean probably like over 95% or even more) are false positives, even those escalated, as at T1 we don't have enough access (or time given number of alerts) to check what is really going on.

95% of our rules are based on threat intel. If there's suspicious traffic, we won't do anything about it because the 'IP is clean'.

[redacted] 3 points · 25 days ago
drowning in tickets

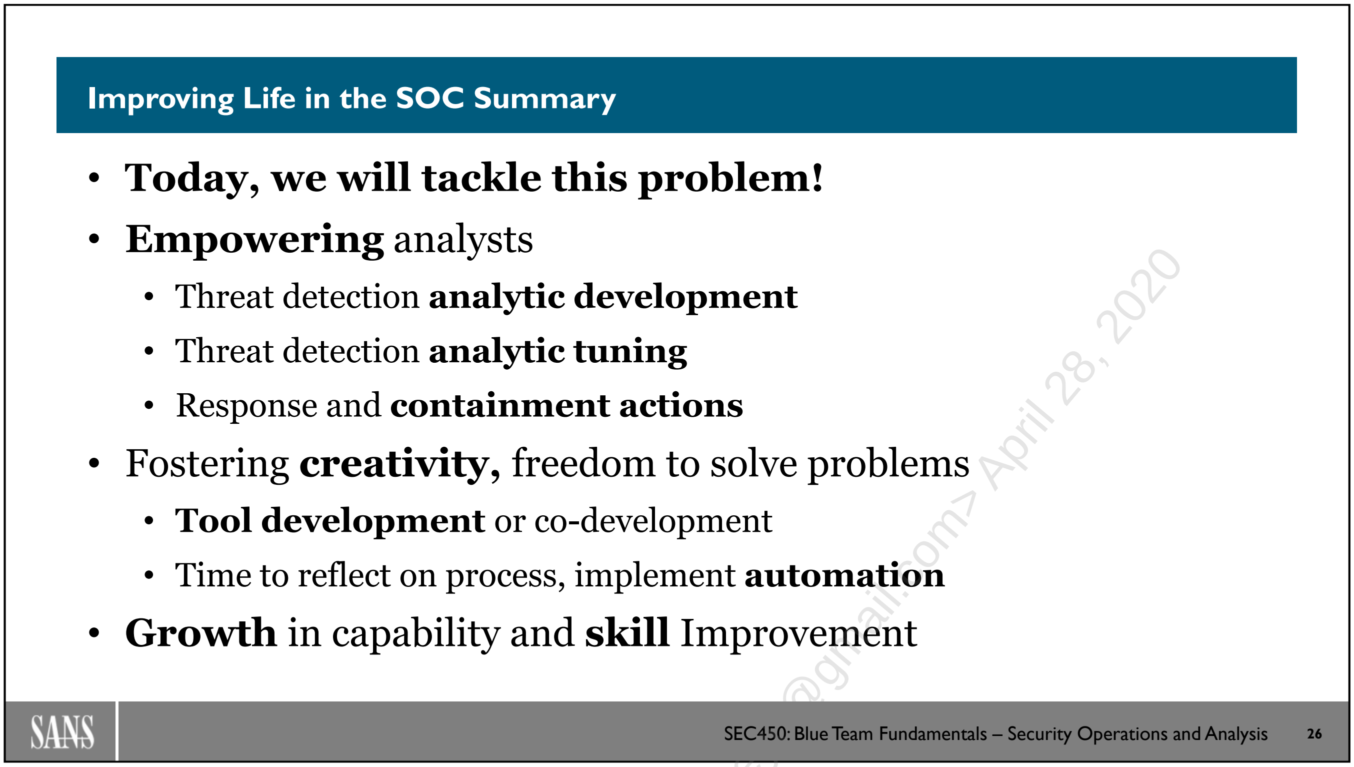
[redacted] 2 points · 19 hours ago
I know what you mean, Tier 1 SOC work is really boring... of course this all depends on where you work.

SOCs should **not** be this way!

Additional Comments Care of Reddit

There's no doubt that poor job situations are out there. Look at these random comments pulled from reddit.com/r/asknetsec threads about SOCs. Clearly, there's some bitterness. But sometimes, the fix is not as hard as you think. Just a few scripts and process modifications and you may be able to turn a grind of a job into something fun and challenging. If we can clearly understand the problem, the potential fixes, and the tools that will help us implement them, we are much more likely to turn what is an ugly, dysfunctional situation around in a very positive way.

While every job will have something to gripe about to *some* extent, in my experience, SOC situations can almost always be made significantly better with a little ingenuity and awareness of the possibilities. And that's what we'll be focusing on today—ways we can make everyday life better in very meaningful ways. With some creative thinking and instruction on the driving factors, some of the everyday pains can be removed from SOC life paving the way for our happiness, learning, and development.



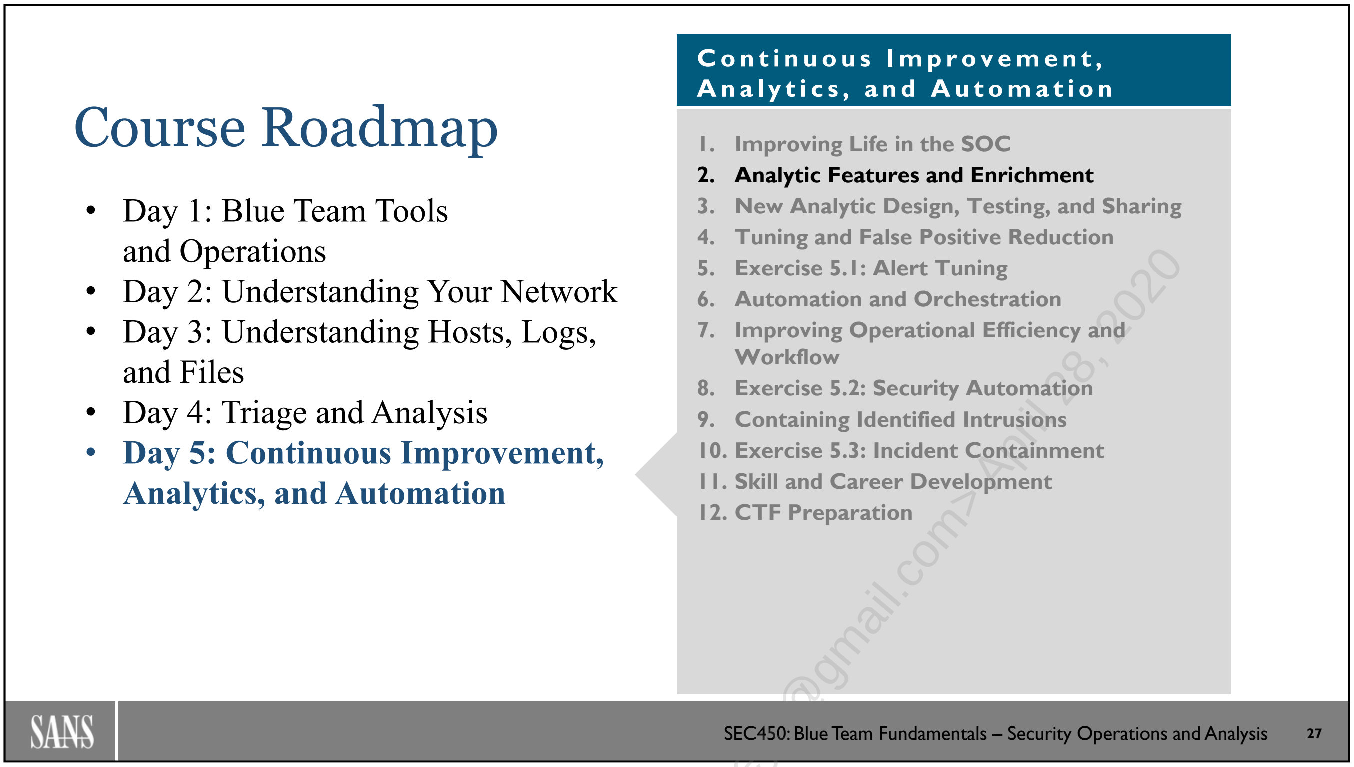
Improving Life in the SOC Summary

- **Today, we will tackle this problem!**
- **Empowering** analysts
  - Threat detection **analytic development**
  - Threat detection **analytic tuning**
  - Response and **containment actions**
- Fostering **creativity,** freedom to solve problems
  - **Tool development** or co-development
  - Time to reflect on process, implement **automation**
- **Growth** in capability and **skill** Improvement

SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 26

Improving Life in the SOC Summary

Today, we will focus on challenging ourselves and bettering both company defenses and our work life through continuous improvement of the SOC. Since it has been shown time and time again that analysts crave variety, empowerment, and the capacity to grow in their jobs, we will discuss methods to do just that, and how to convince management we need more of it when it is lacking. As the author of this course, I truly want to see everyone in a SOC enjoy their job and grow into a long and successful career on the blue team. For this reason, this book will hit these topics to round out the rest of the processes we have discussed, and focus on maximizing the factors identified in the research that keep burnout at bay to ensure a virtuous cycle of learning and growth within your SOC.



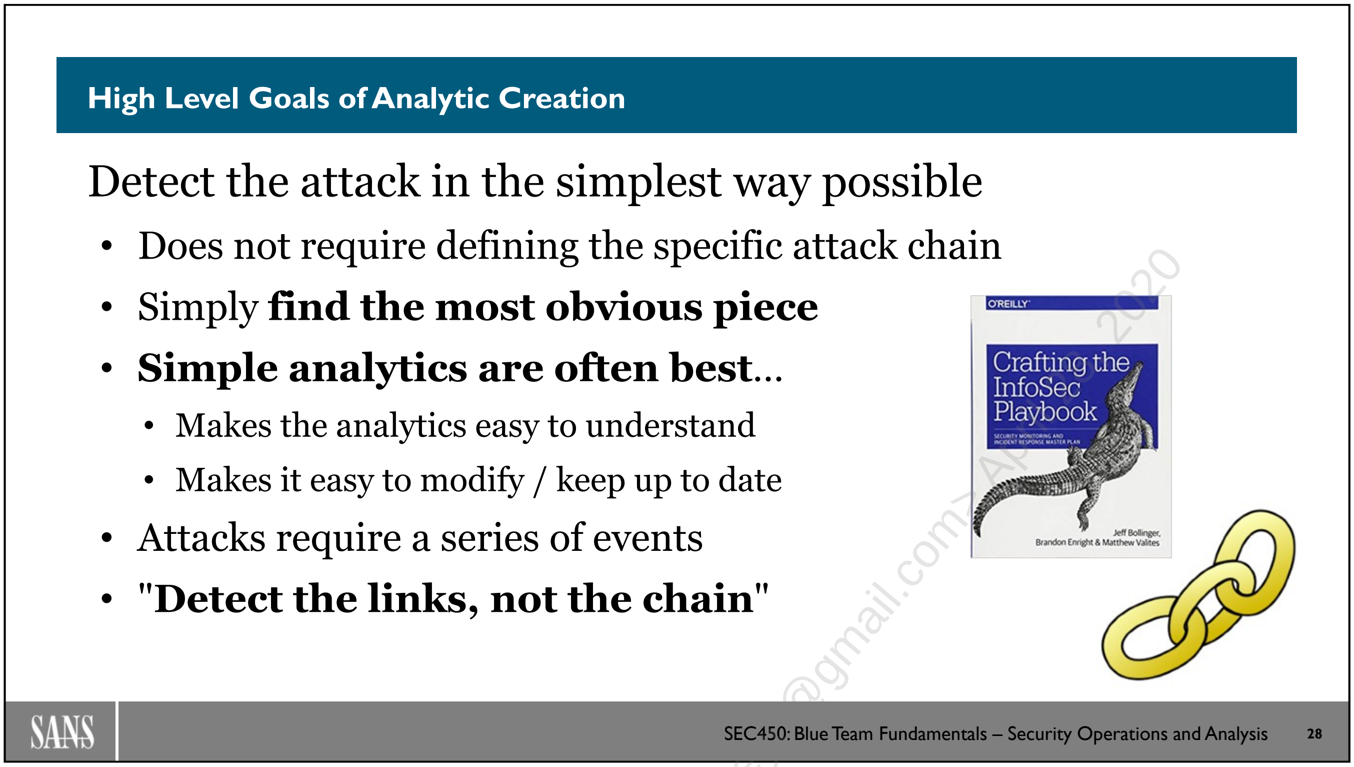
Course Roadmap

- 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**

Continuous Improvement, Analytics, and Automation

1. Improving Life in the SOC
2. **Analytic Features and Enrichment**
3. New Analytic Design, Testing, and Sharing
4. Tuning and False Positive Reduction
5. Exercise 5.1: Alert Tuning
6. Automation and Orchestration
7. Improving Operational Efficiency and Workflow
8. Exercise 5.2: Security Automation
9. Containing Identified Intrusions
10. Exercise 5.3: Incident Containment
11. Skill and Career Development
12. CTF Preparation

This page intentionally left blank.



High Level Goals of Analytic Creation

Detect the attack in the simplest way possible

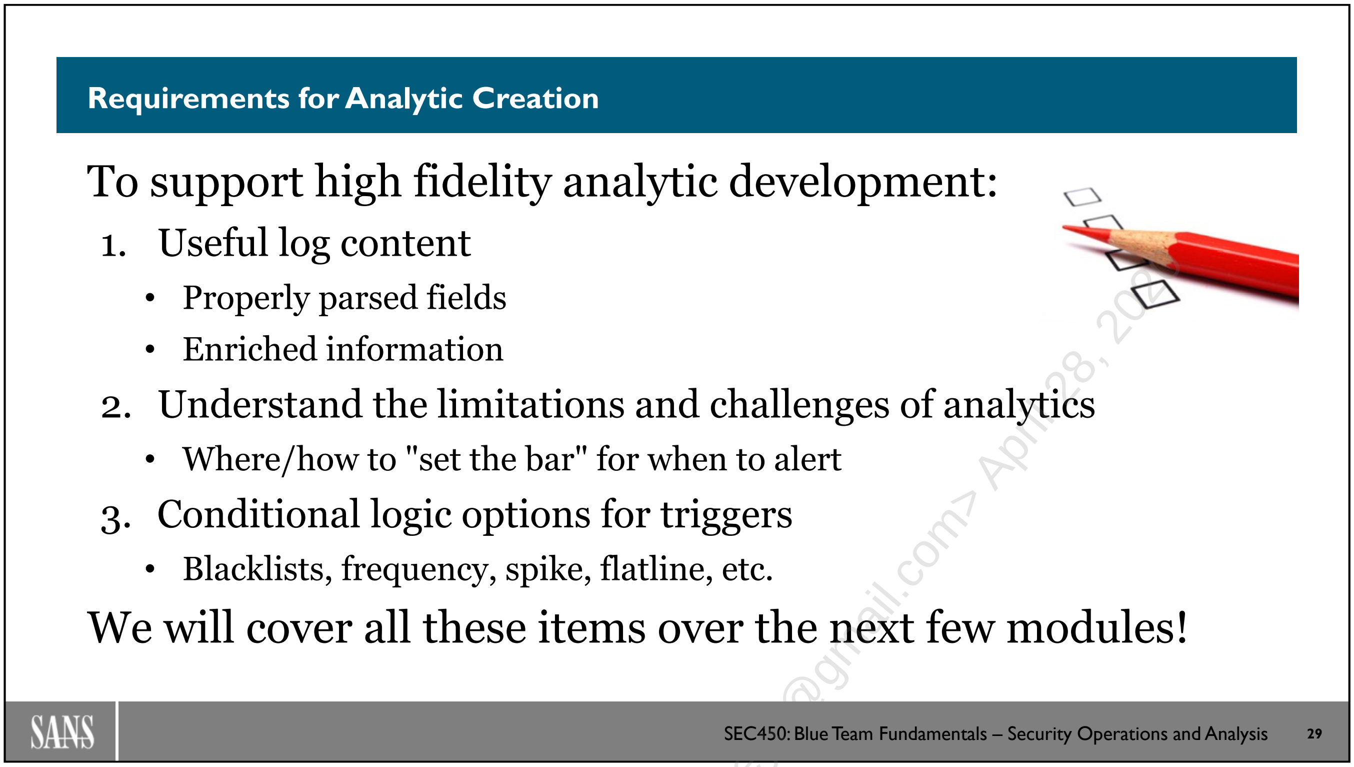
- Does not require defining the specific attack chain
- Simply **find the most obvious piece**
- **Simple analytics are often best**…
  - Makes the analytics easy to understand
  - Makes it easy to modify / keep up to date
- Attacks require a series of events
- "**Detect the links, not the chain**"

O'REILLY
Crafting the InfoSec Playbook
SECURITY MONITORING AND INCIDENT RESPONSE MASTER PLAN
Jeff Bollinger, Brandon Enright & Matthew Valites

High Level Goals of Analytic Creation

Taking a step back and looking at analytic creation, what should we be trying to detect? Should we write a complicated rule that can identify attackers entering the environment, searching around, compromising multiple systems and stealing data, or should we write atomic rules for each piece? A SIEM would let us to either. We could write a rule that says "brute force password attack, followed by an IDS alert for the same host, followed by an unknown application running…" but is that a good idea? As I'm sure you can intuitively tell, the answer is no. The best analytics are the simplest. To detect attacks with high fidelity, we do not need to come up with a complicated series of events that are highly specific. Most of the time, each piece of the chain will be from a grab bag of possible options anyway. The better way is to simply write your analytics to identify a *piece* of the attack in the easiest way possible and have coverage through many analytics that attempt to do that across the attack stages. This means writing analytics for malicious email, attempted exploits, running new programs, etc., all independent of each other. As stated in *Crafting the InfoSec Playbook*, which has several outstanding chapters on this concept—"detect the chain links, not the chain."[1] In this fashion, the malicious activity is most likely to get caught since it is not so highly dependent on other factors, and points us in the right direction to deduce the rest of the attack through other means. We do not need to know all the details at the start; we only need a reliable hint that something has occurred.

[1] https://www.amazon.com/Crafting-InfoSec-Playbook-Security-Monitoring/dp/1491949406



Requirements for Analytic Creation

To support high fidelity analytic development:

1. Useful log content
   - Properly parsed fields
   - Enriched information
2. Understand the limitations and challenges of analytics
   - Where/how to "set the bar" for when to alert
3. Conditional logic options for triggers
   - Blacklists, frequency, spike, flatline, etc.

We will cover all these items over the next few modules!

SANS

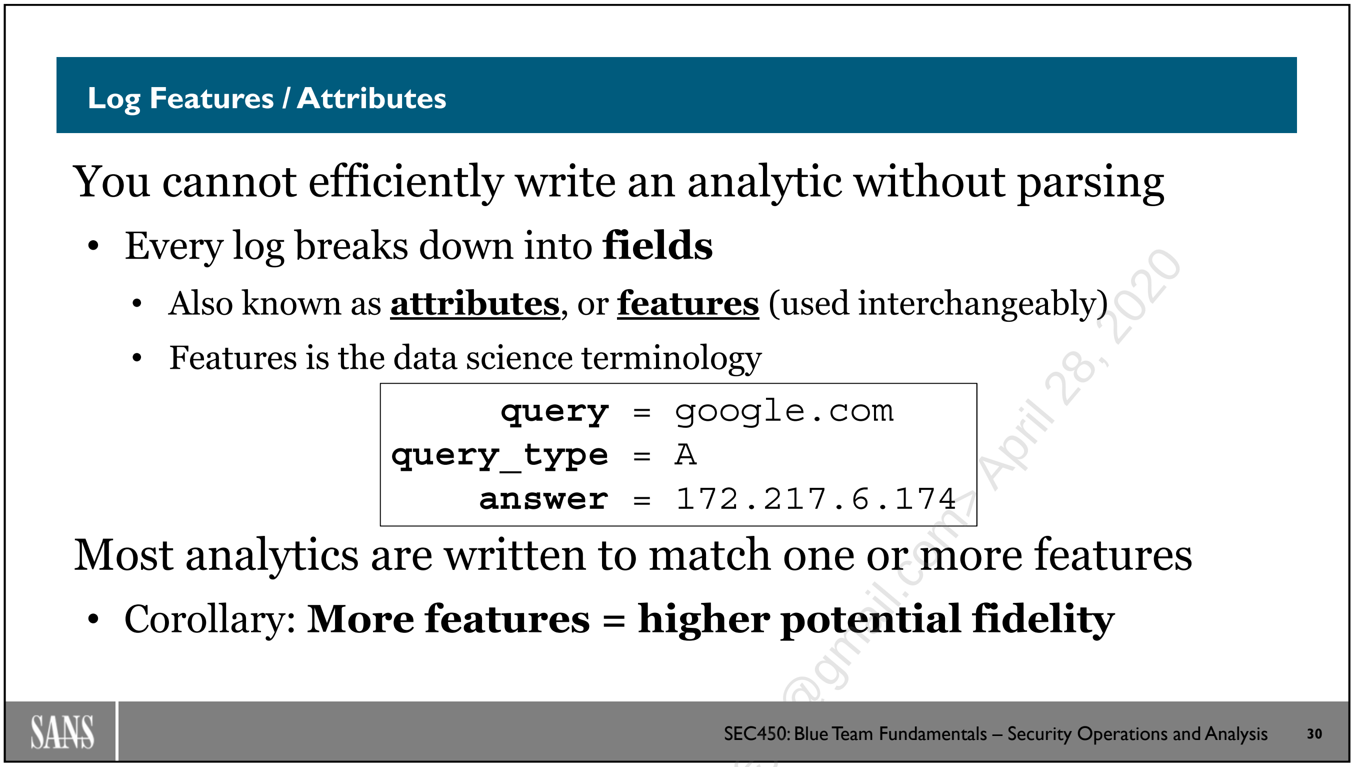
SEC450: Blue Team Fundamentals – Security Operations and Analysis 29

Requirements for Analytic Creation

If we're going to set off to write alerts on our own, we'll need a few items in place to do it correctly. First, our data must support the types of rules we're going to write. You can only write an analytic as good as the quality of data you're getting. If you only get log feeds from your devices but are unable to parse them fully or perform any additional enrichment with the SIEM, you are undoubtedly going to struggle compared to someone with the extra enrichment capability. As the old saying goes "garbage in, garbage out."

Second, we'll need to understand the condition we're trying to detect in the greater context of an attack. How willing are we to accept a false negative condition? Is this an analytic to catch an initial exploit, or some disastrous condition that only happens at the end of the kill chain? In other words, are we writing a rule for something that absolutely cannot be missed, or is it something we'd like to detect if possible, but the world won't stop if we miss it? If the rule isn't a life-or-death situation (as many are not) the analytic can be oriented toward minimizing false positives at the expense of the occasional false negative.

Finally, we need the ability of our SIEM to write a rule based on different types of triggering logic. Whether these are list, threshold, value, count, or sequence based, the more options we have, the more flexible our alerting can be.



Log Features / Attributes

You cannot efficiently write an analytic without parsing

- Every log breaks down into **fields**
  - Also known as **attributes**, or **features** (used interchangeably)
  - Features is the data science terminology

```
     query = google.com
query_type = A
    answer = 172.217.6.174
```

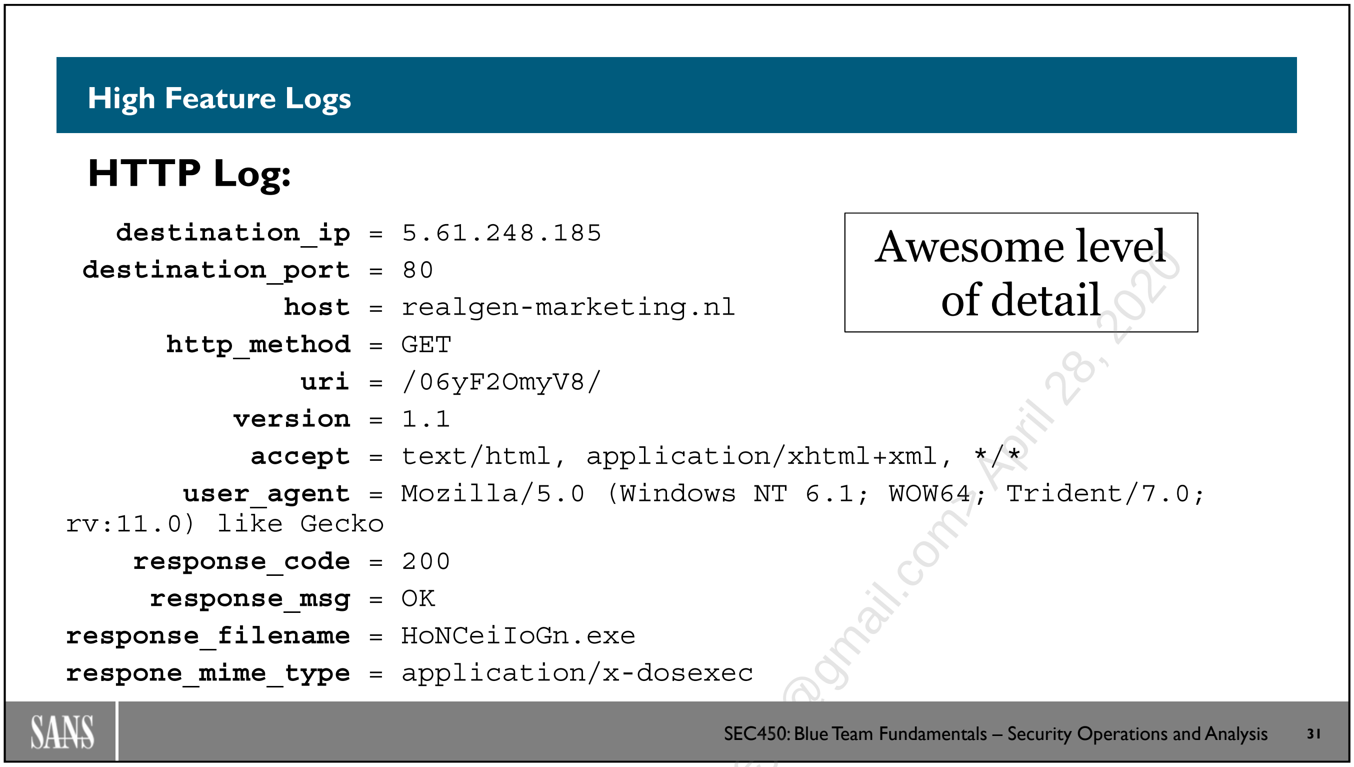
Most analytics are written to match one or more features

- Corollary: **More features = higher potential fidelity**

Log Features / Attributes

To create a simple analytic, the technique used is often content matching of the information inside the log. In any given log, there are a set of extractable fields like username or IP addresses that the SIEM will need to parse out. The fields (sometimes called attributes, used interchangeably for this class) referred to in data science terminology with the more general name "features" are a key driver in the fidelity of creating an analytic from a certain log.

In the slide, we have the most common features someone would extract from a typical DNS query—what the query was for (google.com), what type of query it was (A record), and what the response was (172.218.6.174). Based on these three features, there are already several options for analytics that could be created — looking for known bad IP addresses and domain names, but some of the more complicated malware may not be detectable from these features alone. The corollary here is that, in general, the more features a given type of log has, the better chance we have to key off one or more of them using logical tests to identify malicious conditions.



High Feature Logs

HTTP Log:

```
    destination_ip = 5.61.248.185
  destination_port = 80
              host = realgen-marketing.nl
       http_method = GET
               uri = /06yF2OmyV8/
           version = 1.1
            accept = text/html, application/xhtml+xml, */*
        user_agent = Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0;
rv:11.0) like Gecko
     response_code = 200
      response_msg = OK
 response_filename = HoNCeiIoGn.exe
 respone_mime_type = application/x-dosexec
```

Awesome level of detail

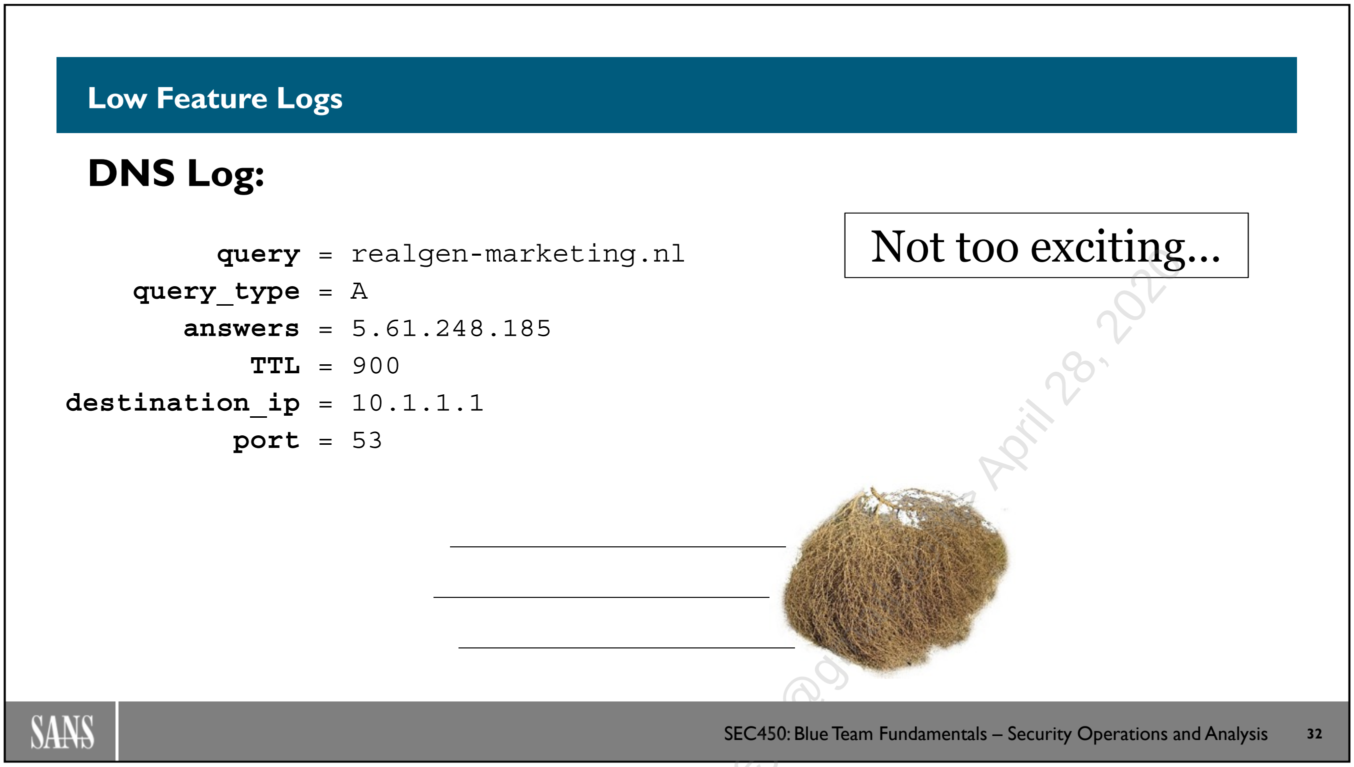
SEC450: Blue Team Fundamentals – Security Operations and Analysis 31

High Feature Logs

The example HTTP log above shows that an HTTP log is relatively great for writing analytics as they come with many fields to work with. HTTP has user agents, referers (not pictured here), methods, and all the other HTTP request and response headers that make for potentially useful data. What that *doesn't* mean, however, is that your analytic should contain conditions for *all* those features. We're merely saying it's better to have more *options*. Sure, you *could* write a very specific analytic, and sometimes you will need to base your alert off multiple features, but it's not a foregone conclusion that using a combination of conditions is the best method in every case. Doing so may cause you to be over specific, raising the specter of false negatives.

The HTTP request shown on this slide is a download by the Emotet malware as captured in January 2019, posted to malware-traffic-analysis.net.[1]

[1] http://malware-traffic-analysis.net/2019/01/21/index.html



Low Feature Logs

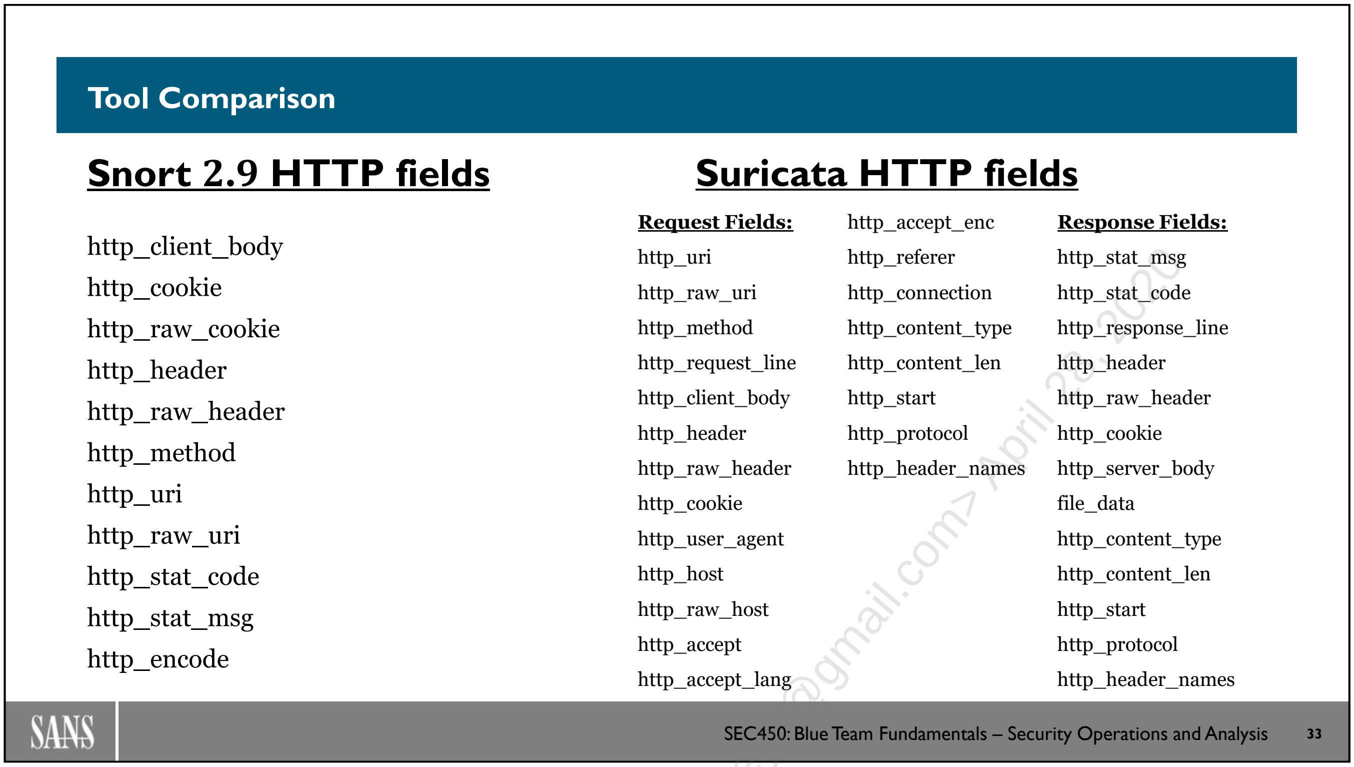
DNS Log:

```
         query = realgen-marketing.nl
    query_type = A
       answers = 5.61.248.185
           TTL = 900
destination_ip = 10.1.1.1
          port = 53
```

Not too exciting...

Low Feature Logs

When compared to the DNS lookup for the same exchange, it's clear that there is significantly less detail here. If you were trying to identify this as a malicious request based purely on the included log information below, unless the IP or domain was already on one of your threat intel lists, you likely wouldn't flag it. The HTTP request on the previous page, however, has plenty of things that might raise a flag—the Windows 7 user agent, the executable download, the mime type, a .nl TLD on the domain (which isn't inherently evil, but a non-traditional traffic destination for many organizations that may raise the suspicion a bit). As you can see, at least in a general sense, the more features there are in a log, the more chances you'll have to pick it out as suspicious.



Tool Comparison

Snort 2.9 HTTP fields

http_client_body
http_cookie
http_raw_cookie
http_header
http_raw_header
http_method
http_uri
http_raw_uri
http_stat_code
http_stat_msg
http_encode

Suricata HTTP fields

Request Fields:
http_uri
http_raw_uri
http_method
http_request_line
http_client_body
http_header
http_raw_header
http_cookie
http_user_agent
http_host
http_raw_host
http_accept
http_accept_lang
http_accept_enc
http_referer
http_connection
http_content_type
http_content_len
http_start
http_protocol
http_header_names

Response Fields:
http_stat_msg
http_stat_code
http_response_line
http_header
http_raw_header
http_cookie
http_server_body
file_data
http_content_type
http_content_len
http_start
http_protocol
http_header_names

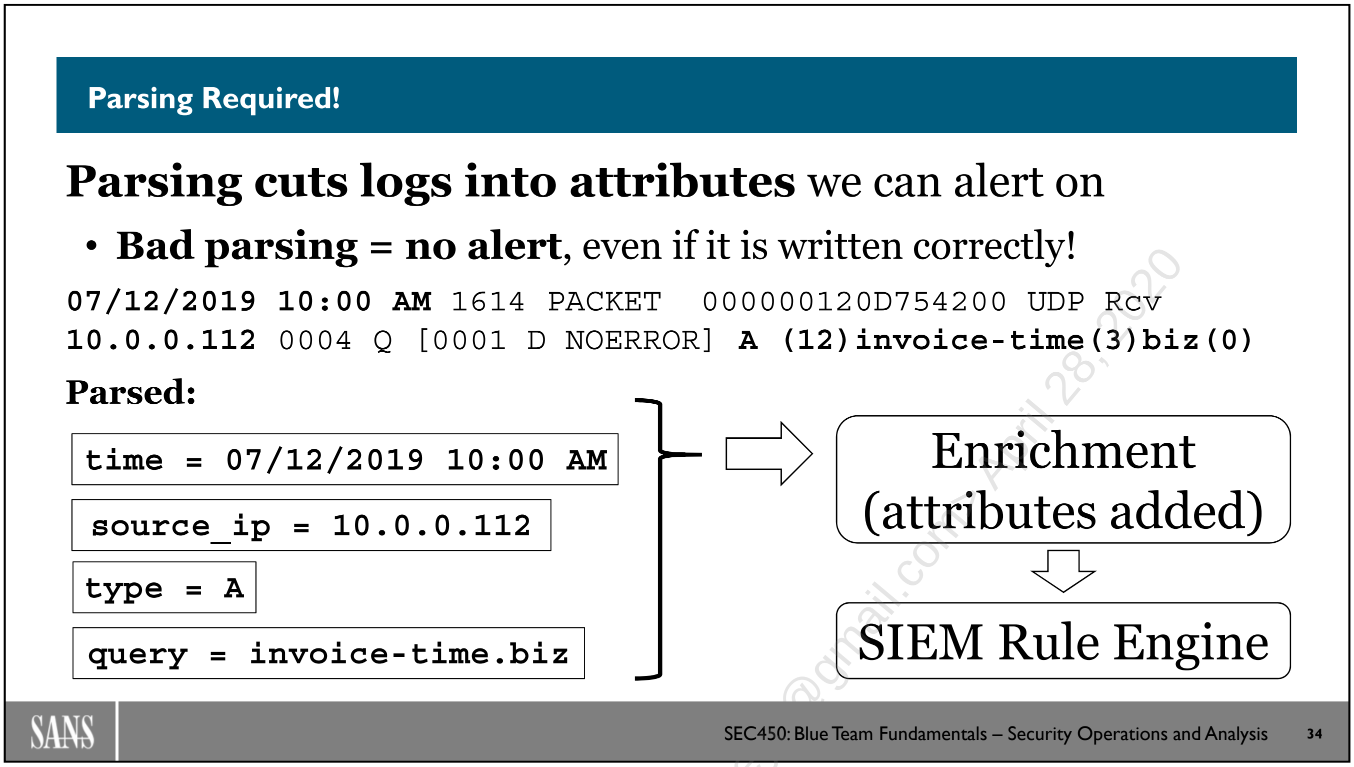
Tool Comparison

Given that more fields are generally better, consider now what data you are receiving from your tools. If you have the option of two different tools to purchase, one of which can extract very few features from the data it sees and another that can break the data down much more granularly, which one do you think will be capable of writing higher fidelity rules? The latter, of course! Perhaps the tools you already have are capable of giving more information as well—it's worth looking in the user manual to verify you are getting all security-relevant output that can be produced.

This slide shows an example of this; it shows a list of the fields both Snort and Suricata can use to reference fields in an HTTP transaction. Suricata clearly can break an HTTP transaction into much more distinct features—meaning, in most cases, you could write a higher fidelity rule for HTTP using Suricata compared to Snort.

[1] http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node32.html

[2] https://suricata.readthedocs.io/en/suricata-4.1.3/rules/http-keywords.html



Parsing Required!

Parsing cuts logs into attributes we can alert on

- **Bad parsing = no alert**, even if it is written correctly!

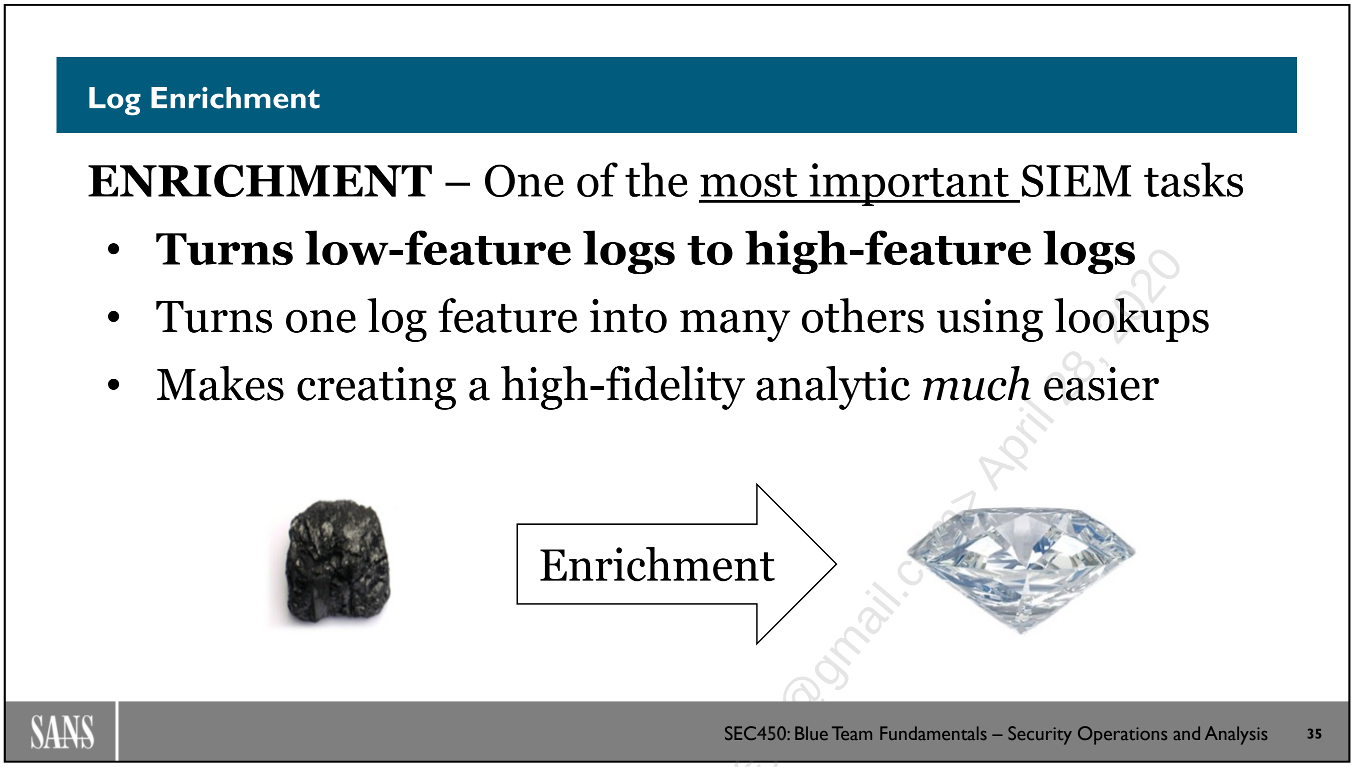
```
07/12/2019 10:00 AM 1614 PACKET  000000120D754200 UDP Rcv
10.0.0.112 0004 Q [0001 D NOERROR] A (12)invoice-time(3)biz(0)
```

Parsed:

```
time = 07/12/2019 10:00 AM
source_ip = 10.0.0.112
type = A
query = invoice-time.biz
```

Parsing Required!

We mentioned that parsing and indexing is one of the things that will affect search speed in a major way, but since analytics must key off attributes, it also drives the ability to alert as well. No matter how well you write your analytic rule, if the data coming in isn't parsed correctly, there's no way it will be able to fire. You can't match something against the source_ip field if there isn't a source_ip field! This slide has an illustration of what must happen to the raw log on top as it travels through and finally hits the analytic engine. First, it must be successfully parsed into all the relevant attributes we care about (notice it doesn't require parsing *all* possible fields, just the ones we care about). Then, optional (but highly encouraged) enrichment can be applied. Those enrichments could be resolving the source_ip to a hostname and user, making a passive DNS request for the domain name, or anything else that would add additional fields to the log before it enters the evaluation engine. After we have all the attributes inherent to the log parsed, and enriched with any additional information, *then* the rule engine runs the logic of the analytics we have created against it. If there are any matches, we send an alert. As you can see here, without proper parsing, everything downstream will fail. So, although we have said it multiple times, we cannot overemphasize the importance of that fact—if you want your SIEM to deliver value, we must parse the logs correctly!



Log Enrichment

ENRICHMENT – One of the most important SIEM tasks

- **Turns low-feature logs to high-feature logs**
- Turns one log feature into many others using lookups
- Makes creating a high-fidelity analytic *much* easier

Enrichment

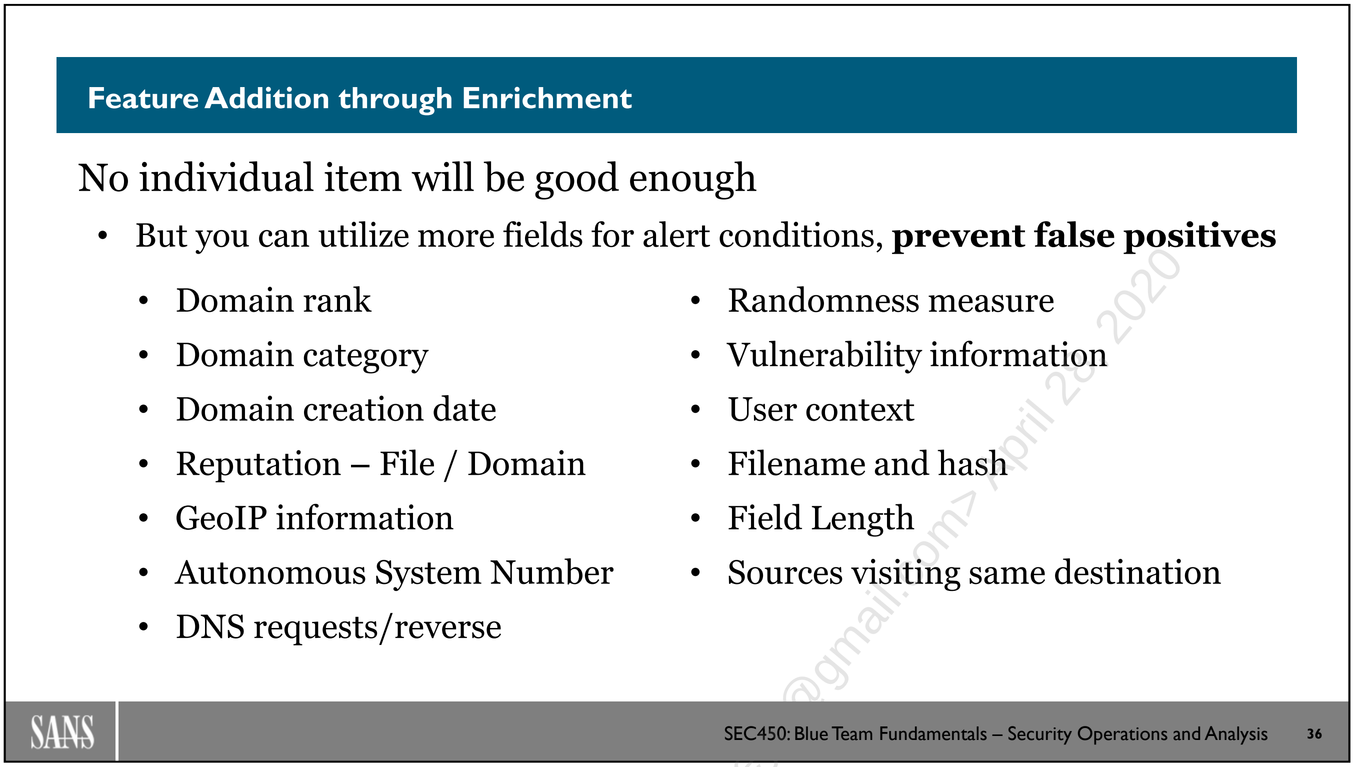
SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 35

Log Enrichment

Although we have alluded to the need to supplement your logs with additional info, we haven't yet specifically discussed what types of enrichment provide the best value for detection. When it comes to log enrichment, the SIEM is the tool that can and should be doing this job for you. If you are only accepting logs into the SIEM as they are, without using its capabilities to make those logs better, you are missing out on one of the main points of having a SIEM.

Enrichment is a key capability for writing high-quality analytics and every SIEM is designed to do it in multiple ways. The high-level view is that the SIEM will take the parsed fields from the log and perform additional steps based on that data, allowing you to understand better what that event is telling you. For example, consider a DNS log for a user looking up the domain unknownsite1234.com. Without enrichment, you know a source IP, domain, and A record answer. *With* enrichment features, the SIEM can look up that domain in threat intel sources, resolve past IP addresses, give you a reputation score and domain category, tell you if it's a top-ranked site, and much, much more. If you only had the DNS log, you would have to do a lot of extra work to decide if it's bad. The SIEM can automatically do it all for you with log enrichment, giving you the ability to potentially make the call on if it is malicious right away, and even use those extra enrichments in your analytics!



Feature Addition through Enrichment

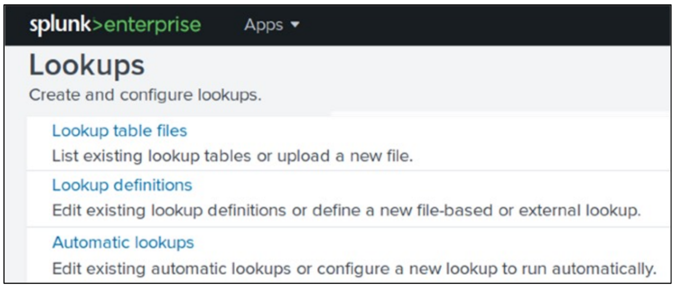
No individual item will be good enough

- But you can utilize more fields for alert conditions, **prevent false positives**
  - Domain rank
  - Domain category
  - Domain creation date
  - Reputation – File / Domain
  - GeoIP information
  - Autonomous System Number
  - DNS requests/reverse
  - Randomness measure
  - Vulnerability information
  - User context
  - Filename and hash
  - Field Length
  - Sources visiting same destination

Feature Addition Through Enrichment

What can you do to improve low-feature logs like DNS, whitelisting hits, or other logs that have a relatively minimal amount of info in them? Enrichment! Each SIEM does this a bit differently so you may need to do some digging to understand how to implement these lookups but given the ability to add enriched info based on the features present, your ability to sniff out suspicious events becomes significantly stronger.

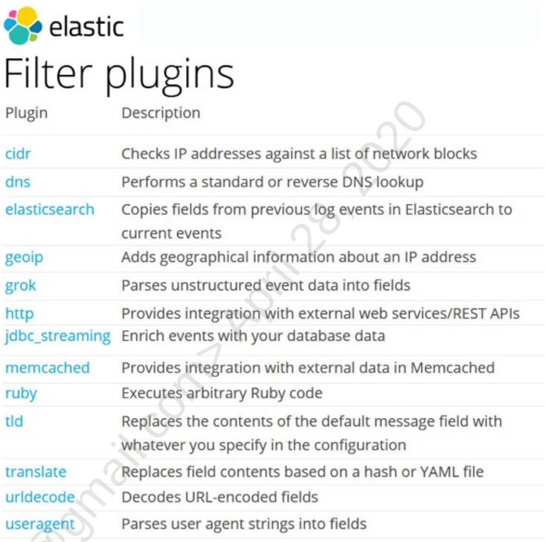
SIEM Enrichment Capabilities



Lookups

Create and configure lookups.

- Lookup table files
  List existing lookup tables or upload a new file.
- Lookup definitions
  Edit existing lookup definitions or define a new file-based or external lookup.
- Automatic lookups
  Edit existing automatic lookups or configure a new lookup to run automatically.



Filter plugins

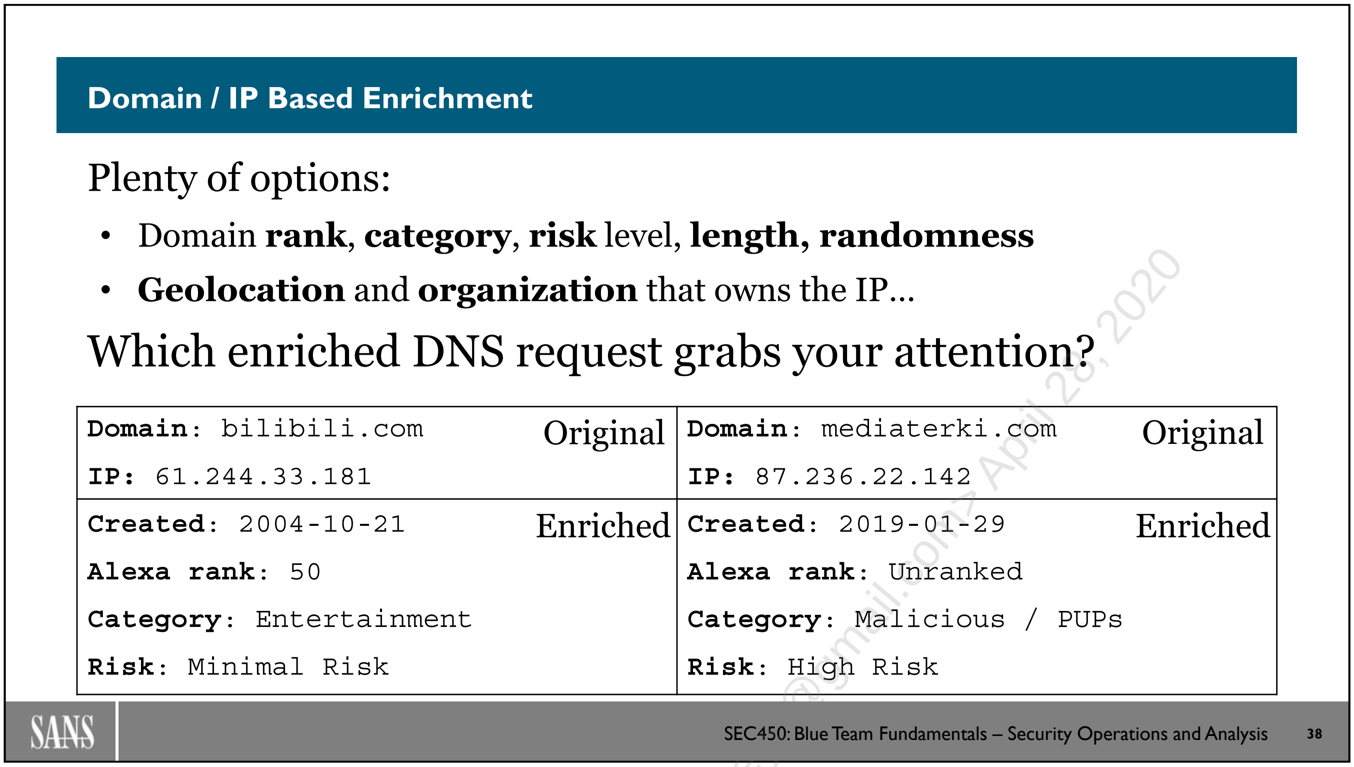
Plugin	Description
cidr	Checks IP addresses against a list of network blocks
dns	Performs a standard or reverse DNS lookup
elasticsearch	Copies fields from previous log events in Elasticsearch to current events
geoip	Adds geographical information about an IP address
grok	Parses unstructured event data into fields
http	Provides integration with external web services/REST APIs
jdbc_streaming	Enrich events with your database data
memcached	Provides integration with external data in Memcached
ruby	Executes arbitrary Ruby code
tld	Replaces the contents of the default message field with whatever you specify in the configuration
translate	Replaces field contents based on a hash or YAML file
urldecode	Decodes URL-encoded fields
useragent	Parses user agent strings into fields

Others:

- **QRadar:** Reference Sets
- **ArcSight:** Map Files
- **McAfee ESM:** Data Enrichment Sources
- **AlienVault USM:** Plugins

SIEM Enrichment Capabilities

Although all SIEMs will differently approach enrichment (and may use a different term for it), rest assured the capabilities are there in some respect. This slide shows how enrichment can be done via the "Lookups" feature in the Splunk, "filter plugins" in Logstash, and lists some of the equivalent capabilities for other SIEMS as well. You may have to get creative to implement certain enrichment if your SIEM doesn't have lots of options, but as they say, "where there's a will, there's a way," and SIEM enrichments are no exception. Whether you must create a REST API service implemented with python, have the SIEM execute a command, or load all the data into a lookup table, there is a solution available for nearly everything if you put your mind to it.



Domain / IP Based Enrichment

Plenty of options:

- Domain **rank**, **category**, **risk** level, **length, randomness**
- **Geolocation** and **organization** that owns the IP…

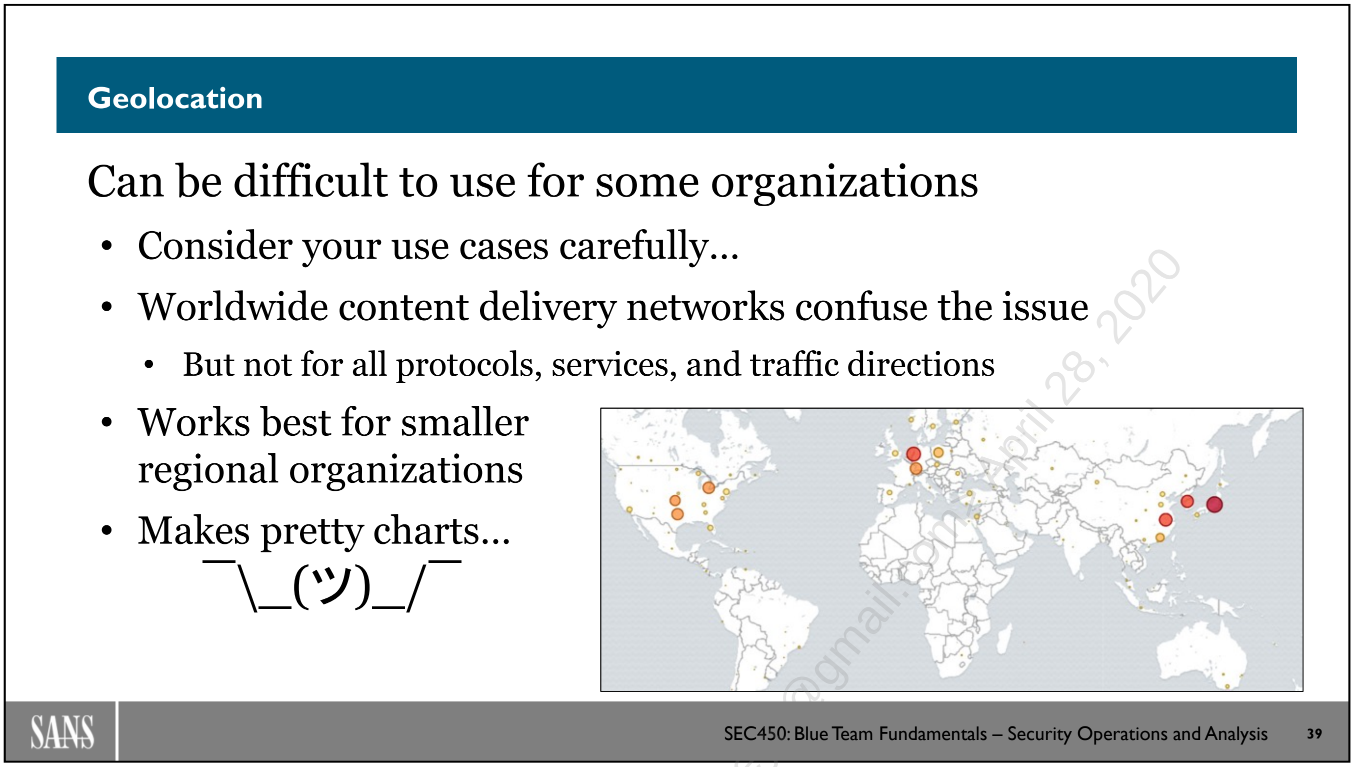
Which enriched DNS request grabs your attention?

Domain: bilibili.com **IP:** 61.244.33.181 Original	**Domain**: mediaterki.com **IP:** 87.236.22.142 Original
Created: 2004-10-21 **Alexa rank**: 50 **Category**: Entertainment **Risk**: Minimal Risk Enriched	**Created**: 2019-01-29 **Alexa rank**: Unranked **Category**: Malicious / PUPs **Risk**: High Risk Enriched

Domain / IP Based Enrichment

There are lots of ways to take a domain name or IP address and add context to make alerts much more useful. Some of the things you might do are automatically look up domain ranks in Alexa, the website category or risk level (if not included from a proxy already), you could calculate the length of the domain, or even measure the apparent randomness. All these factors independently might not be enough to raise an alarm, but when factored with something you have already identified as suspicious, it can make the difference between wasting time having to look up traffic and instantly knowing that something is malicious.

Take the simple example on this slide. Let's say you're looking at two different DNS requests, flagged because the IP addresses they resolved to were labeled "malicious" in one of your threat intel feeds—a common occurrence. The logs then come into your SIEM, and without enrichment, all you have is what is in the top box, the domain and IP address, and the fact that someone somewhere called them evil for some unknown reason. *With* enrichment done by the SIEM to look up creation dates, Alexa top site ranks, a category, and risk level associated with the domain, we add all the information in the second row. Now can you tell which one you might want to look at first? The answer is the right one. The site on the left is a Chinese video-sharing site that is the 50th most popular site on the internet according to Alexa. If you aren't Chinese, you may have never heard of it and therefore would have no clue what it is without the extra data. The site on the right was created much more recently and categorized as malicious. It was indeed a site associated with the Emotet malware. Enrichment gave us these answers immediately from low attribute count logs that would've otherwise required additional work!



Geolocation

Can be difficult to use for some organizations

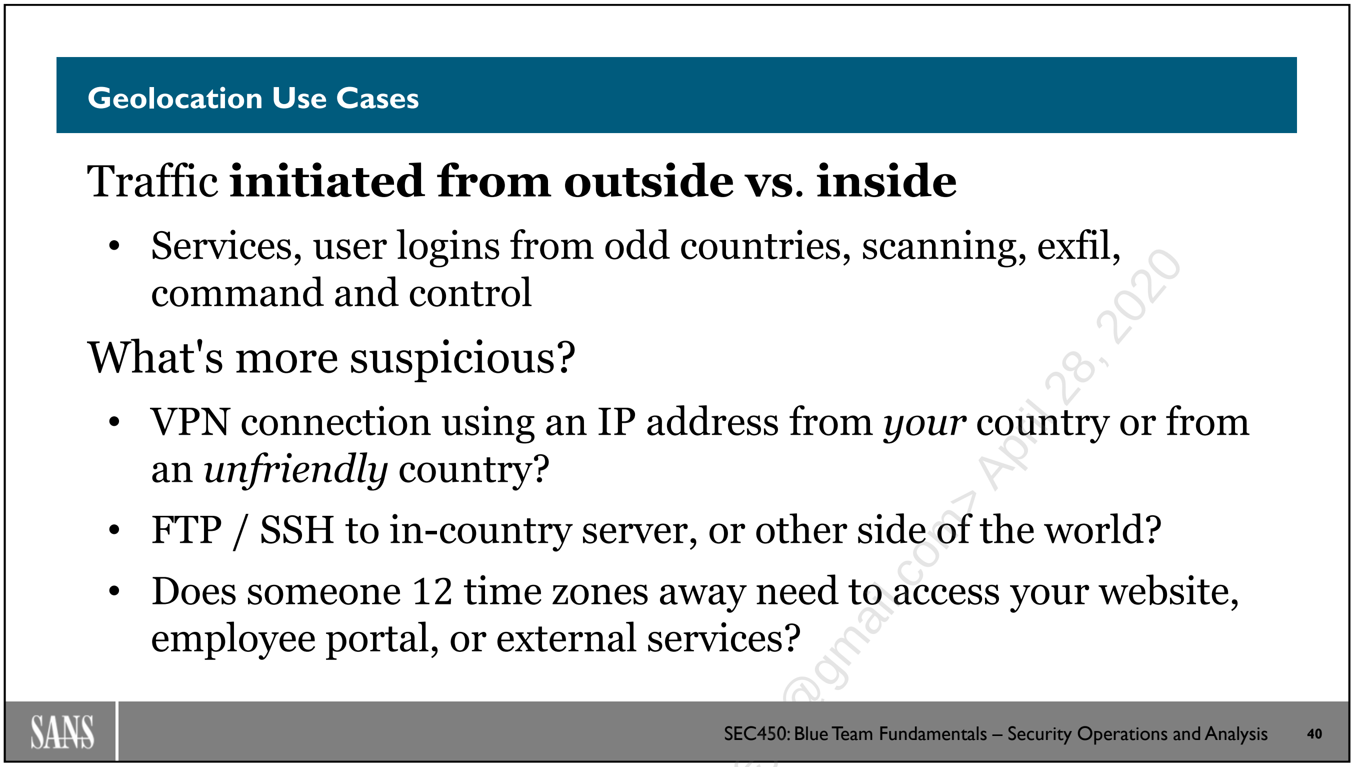
- Consider your use cases carefully…
- Worldwide content delivery networks confuse the issue
  - But not for all protocols, services, and traffic directions
- Works best for smaller regional organizations
- Makes pretty charts…

¯\_(ツ)_/¯

Geolocation

Enrichment of logs with the geolocation of an IP address is a standard feature within SIEMs, but the location of an IP address can be tricky to use. For some organizations that have a global presence, squeezing value out of where their traffic is originating from or going to can be a very complicated endeavor. On its own, it's unlikely to be written into any analytics, but it can be a factor to consider when traffic is already suspect feeling. Although at large you may not get any immediate value out of geolocation, you likely can think of specific cases where you should or should not see traffic from a particular location. Web traffic will likely occur to and from nearly everywhere, but what about other services?

One thing you can say about geolocation enrichment is that it *does* make visually appealing charts. Although your SOC cannot likely take any decisive action on a map of traffic, do not discount the ability of a chart like this to sell what you're doing to a SOC outsider, which is a vital part of keeping your group funded. Visualized data is extremely compelling, and people are naturally drawn to a well-designed dashboard, whether it is "functional" or not. Maps of traffic can make an easy to digest way of saying "we monitor this traffic that is going all over the world for attempted intrusions" and can be a good way of demonstrating, in a non-technical way, the scope of what the SOC is doing and the value the business is getting from it.



Geolocation Use Cases

Traffic **initiated from outside vs. inside**

- Services, user logins from odd countries, scanning, exfil, command and control

What's more suspicious?

- VPN connection using an IP address from *your* country or from an *unfriendly* country?
- FTP / SSH to in-country server, or other side of the world?
- Does someone 12 time zones away need to access your website, employee portal, or external services?

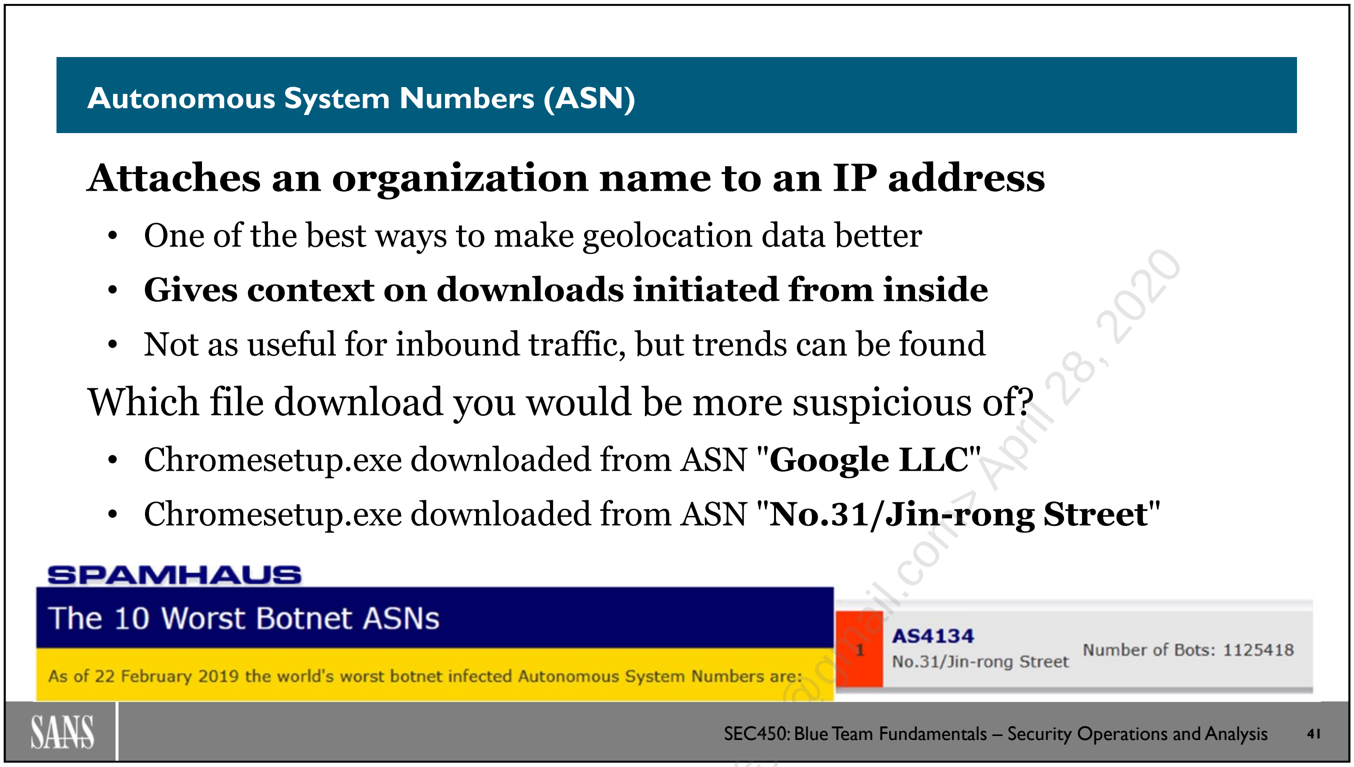
SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 40

Geolocation Use Cases

The great thing about geolocation is that nearly every data source includes an IP address of some source, so it can be nearly universally applied. For specific use cases, consider sorting traffic into where communications have been *initiated*. Connections initiated outside going to your organization will contain traffic to your external services, as well as scans and attacks. If you are a smaller regional organization, you may be able to block traffic to your services (or at least VPN, portals, and other tools meant for employees only) initiated from outside your region. There are very few reasons someone on the other side of the world will need to look at your local business website, for example. Just remember to watch out for accidentally locking people out while they're on travel! Looking at trends in your normal vs. attack traffic will likely highlight trends and blocking opportunities that will work for you.

Communications initiated from inside going and which country they're going out to may paint a different picture. While this traffic can be seemingly hard to wrangle due to content delivery networks and servers placed all over the world, further dividing traffic into types may help sort this out. Although you are likely to have HTTP traffic going everywhere in the world, no matter what type of organization you are, things like SSH, FTP, or other less traditional protocols headed to IP addresses in an unexpected country may be something you are more interested in.



Autonomous System Numbers (ASN)

Attaches an organization name to an IP address

- One of the best ways to make geolocation data better
- **Gives context on downloads initiated from inside**
- Not as useful for inbound traffic, but trends can be found

Which file download you would be more suspicious of?

- Chromesetup.exe downloaded from ASN "**Google LLC**"
- Chromesetup.exe downloaded from ASN "**No.31/Jin-rong Street**"

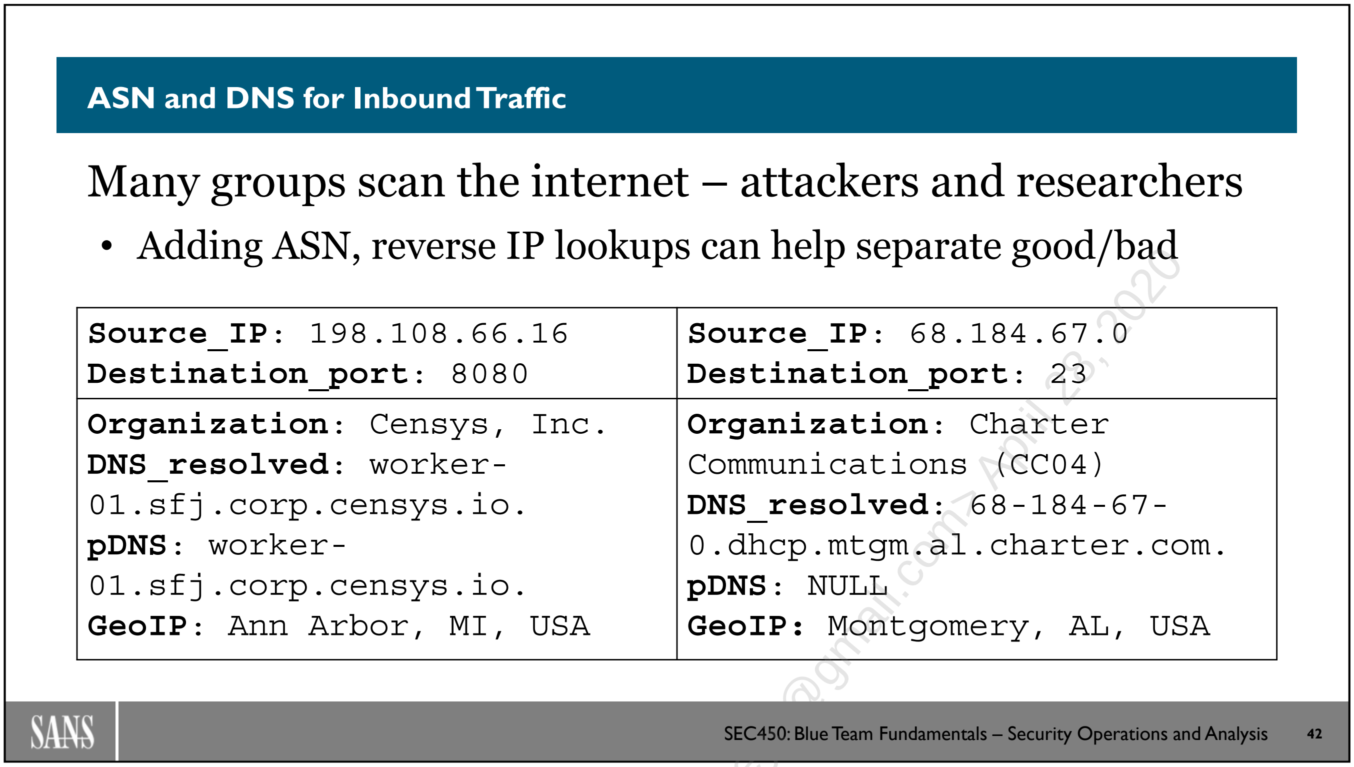
SPAMHAUS
The 10 Worst Botnet ASNs
As of 22 February 2019 the world's worst botnet infected Autonomous System Numbers are:
1
AS4134
No.31/Jin-rong Street
Number of Bots: 1125418

Autonomous System Numbers (ASN)

What about autonomous system numbers? Unfortunately, resolving IP addresses to ASN's or autonomous system numbers (the organization that owns the public IP space) is not as common as geographic mapping, but it can be equally or more useful. Like geoIP info, one direction is potentially more useful than the other. The inbound traffic sorted by ASN may not give you all that much actionable information. You will easily be able to spot trends in which organizations are attacking you but acting on that information might be difficult since many of the attackers will be coming from virtual private servers within each ASN (making it seem like DigitalOcean, Amazon, and Google are attacking you, which it's just their customers). In the outbound direction, however, it can give us useful context on certain types of alerts.

Let's say your IDS notices an executable download and alerts you. The alert includes the source IP of the downloaded and the filename – *Chromesetup.exe*. If you had the SIEM resolve the ASN of the download and the IP address belonged to "Google LLC", it might be easily classified as something not to worry about, especially if the domain name used in the transaction was google.com. On the other hand, if you see an alert for a "ChromeSetup.exe" download from the ASN "No. 31/Jin-rong Street", you might start to get suspicious. Which ASN is that? Why would they be serving Google's files? The answer is they probably wouldn't be, and it might be malware masquerading as Chrome. In fact, according to Spamhaus, this ASN is at the top of the 10 worst Botnet ASNs in the world![1] Given that bit of knowledge, once we implement ASN enrichment in the SIEM, you might even create an analytic to check for a "risky file download from top 10 bad ASN."

[1] https://www.spamhaus.org/statistics/botnet-asn/



ASN and DNS for Inbound Traffic

Many groups scan the internet – attackers and researchers

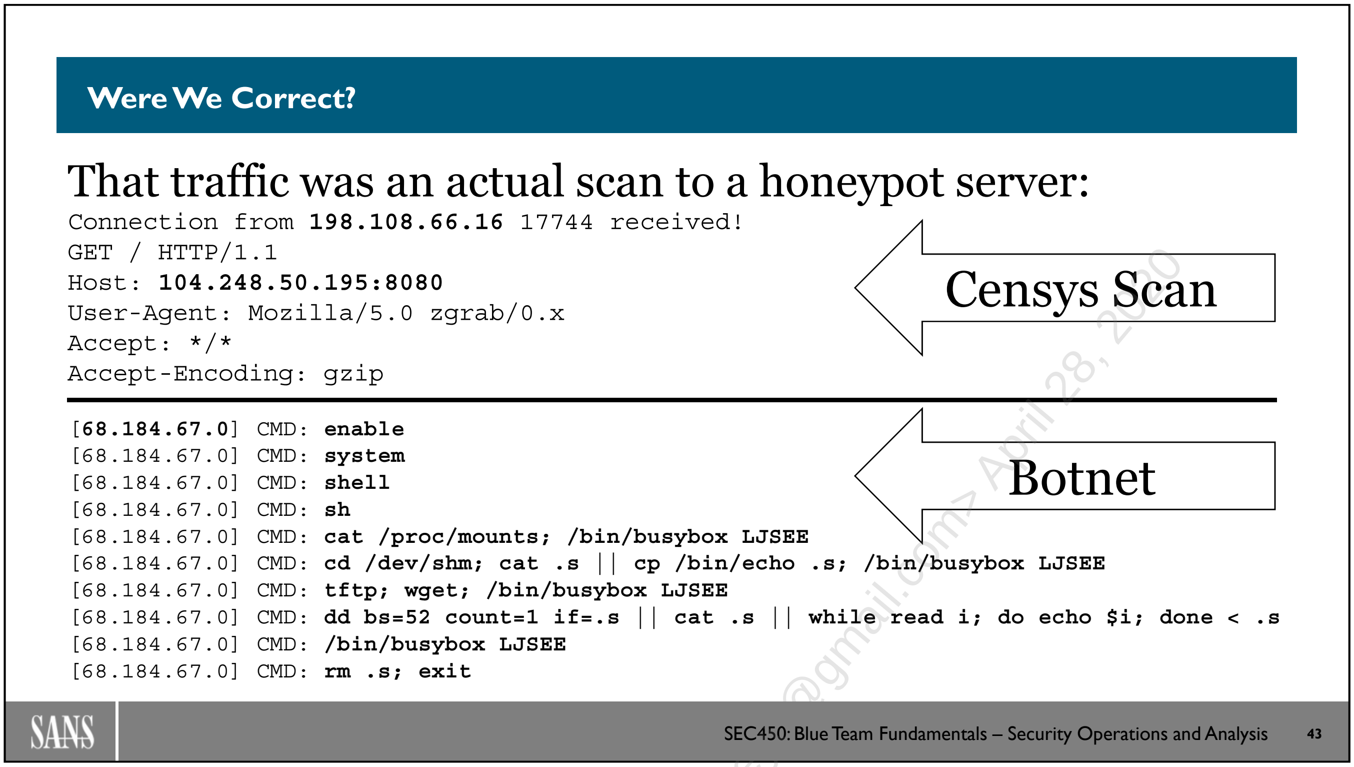
- Adding ASN, reverse IP lookups can help separate good/bad

Source_IP: 198.108.66.16 **Destination_port**: 8080	**Source_IP**: 68.184.67.0 **Destination_port**: 23
Organization: Censys, Inc. **DNS_resolved**: worker-01.sfj.corp.censys.io. **pDNS**: worker-01.sfj.corp.censys.io. **GeoIP**: Ann Arbor, MI, USA	**Organization**: Charter Communications (CC04) **DNS_resolved**: 68-184-67-0.dhcp.mtgm.al.charter.com. **pDNS**: NULL **GeoIP**: Montgomery, AL, USA

ASN and DNS for Inbound Traffic

Adding DNS lookups to your IP addresses, both from active PTR record lookups and passive DNS sources can also help illuminate good from bad inbound traffic. Let's say you received the above information in a log from your firewall. The box is the main information of use we would receive, where the traffic came from and what destination port it came from. With nothing other than an IP address and port to go on, there's not a whole lot else we can get out of this log. If we add enrichment with ASN and reverse/passive DNS however, we get a much better sense of who the two IP addresses attempting the connection came from. On the left side, we see the ASN resolved to Censys, Inc. (the group that constantly scans the internet for research purposes), and the reverse *and* passive DNS resolutions both line up with what we found in the ASN. In this case, we can safely conclude this was likely a research-related scan.

On the right side, we have a telnet connection attempt coming from a Charter Communications ASN (a U.S.-based residential internet service provider). When attempting to resolve the DNS address, we find the PTR record points back to what is likely someone's home internet connection and passive DNS sources seem to know nothing about the IP address. This is much more consistent with an attack than the information on the left. Although we likely wouldn't alert and take action on an individual scan, this shows this information can be used to make quick assessments about what is going on, even if we start with a log containing few fields. We could, however, use this type of information in aggregate over time if we find one IP source or ASN is continuously attempting to intrude.



Were We Correct?

That traffic was an actual scan to a honeypot server:

```
Connection from 198.108.66.16 17744 received!
GET / HTTP/1.1
Host: 104.248.50.195:8080
User-Agent: Mozilla/5.0 zgrab/0.x
Accept: */*
Accept-Encoding: gzip
```

Censys Scan

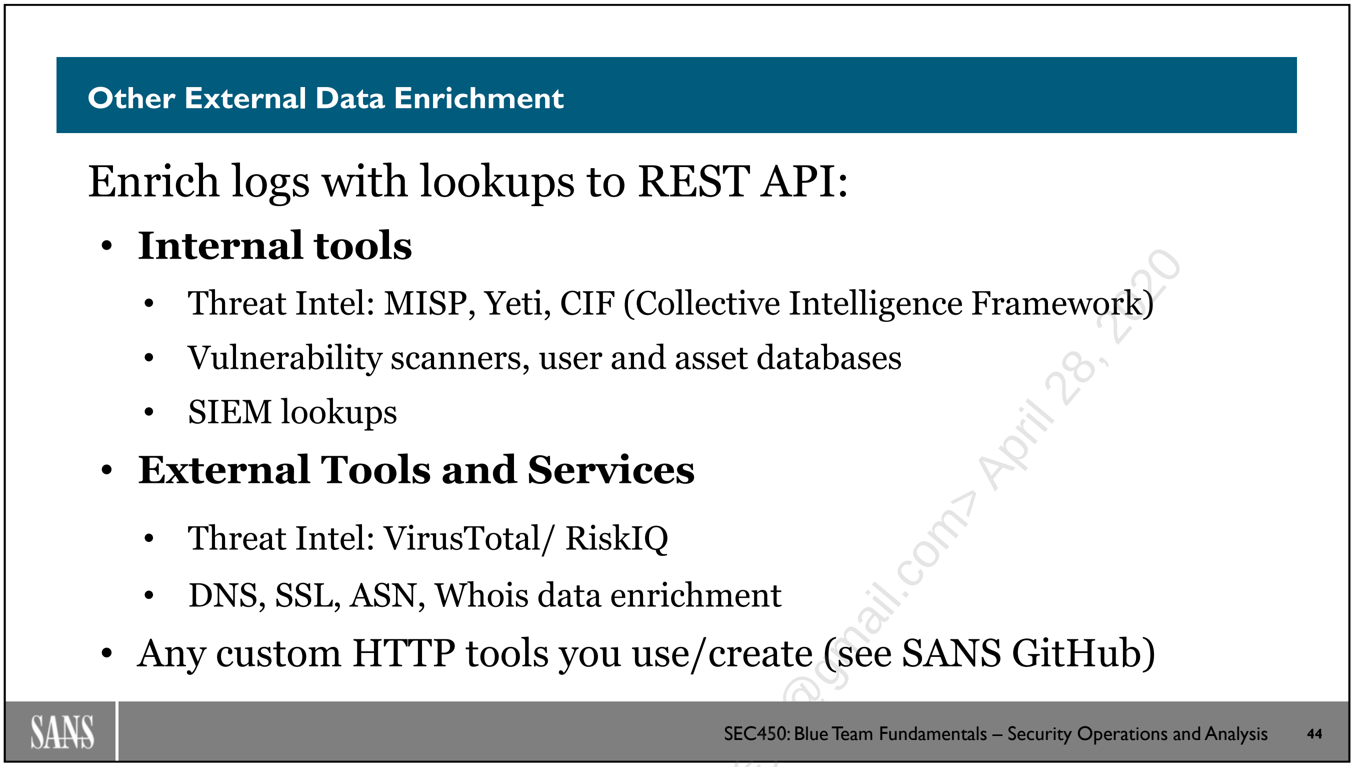
```
[68.184.67.0] CMD: enable
[68.184.67.0] CMD: system
[68.184.67.0] CMD: shell
[68.184.67.0] CMD: sh
[68.184.67.0] CMD: cat /proc/mounts; /bin/busybox LJSEE
[68.184.67.0] CMD: cd /dev/shm; cat .s || cp /bin/echo .s; /bin/busybox LJSEE
[68.184.67.0] CMD: tftp; wget; /bin/busybox LJSEE
[68.184.67.0] CMD: dd bs=52 count=1 if=.s || cat .s || while read i; do echo $i; done < .s
[68.184.67.0] CMD: /bin/busybox LJSEE
[68.184.67.0] CMD: rm .s; exit
```

Botnet

Were We Correct?

Was the assessment using the enrichment correct? Indeed it was. These were both real scans I received to a honeypot server that was not only recording the connection attempt but providing a fake service so we can capture the attacker input as well. In the case of the port 8080 scan, the HTTP request shown at the top of the page was issued after the connection was established. We can see that it is not an attack attempt, just a GET request for the root folder. The User-Agent also shows zgrab, which a quick Googling would turn up as a tool for mass scanning of networks (as Censys does).

The telnet connection, however, was much different. It consisted of the attacker connecting to my interactive telnet honeypot and issuing the commands shown on the bottom half of the page. Unfortunately, the trojan uses an obfuscation technique that caused the tool not to record the full list of commands, which is why the output doesn't quite make sense. Regardless of that issue, this is clearly an attempt to run code on devices that are susceptible. This scan was a botnet scanning the internet for open telnet ports and sending these commands in a scripted way to worm its way into as many devices as possible. Through a long-term collection of scan information with data enrichment, we can start gathering trends and stay on top of the current exploit of the day running wild on the internet.



Other External Data Enrichment

Enrich logs with lookups to REST API:

- **Internal tools**
  - Threat Intel: MISP, Yeti, CIF (Collective Intelligence Framework)
  - Vulnerability scanners, user and asset databases
  - SIEM lookups
- **External Tools and Services**
  - Threat Intel: VirusTotal/ RiskIQ
  - DNS, SSL, ASN, Whois data enrichment
- Any custom HTTP tools you use/create (see SANS GitHub)

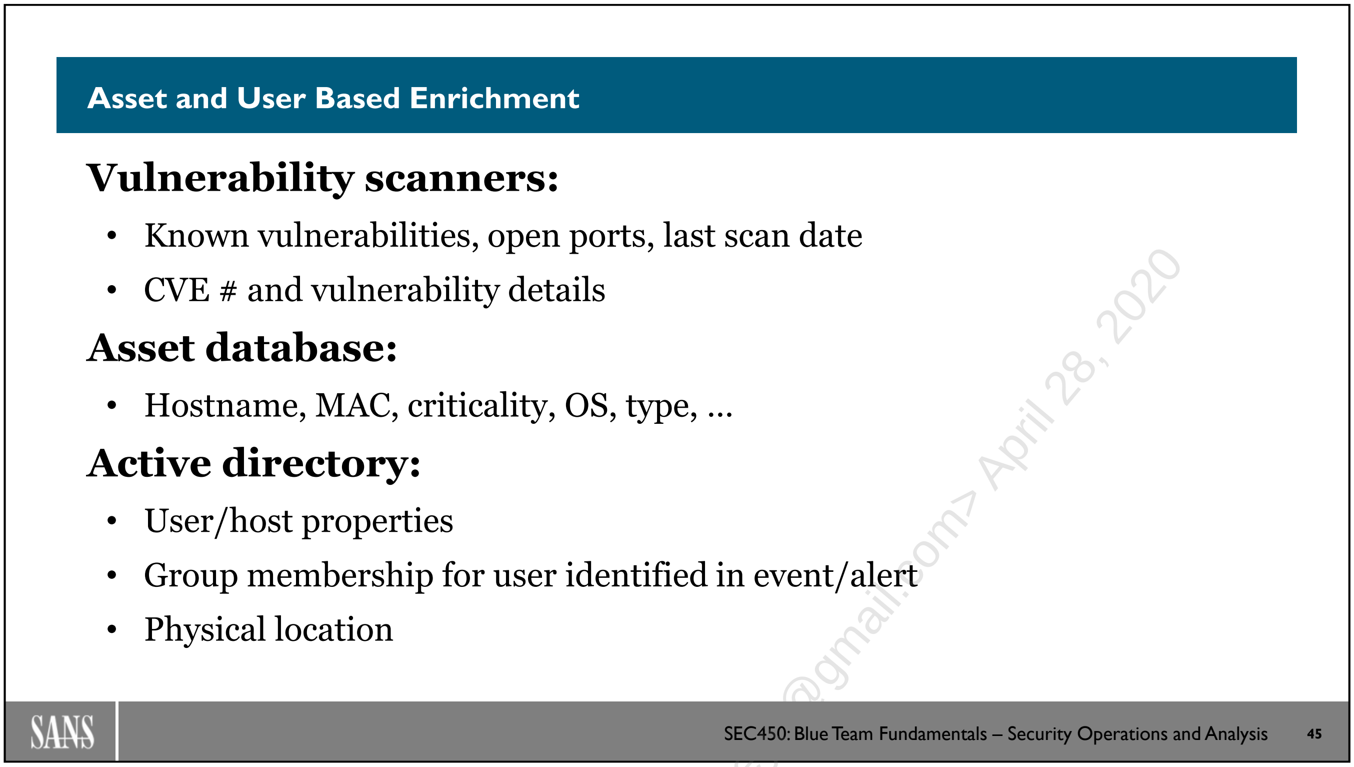
SANS
SEC450: Blue Team Fundamentals – Security Operations and Analysis
44

Other External Data Enrichment

There are plenty of other services you may want to integrate your SIEM with as well, but internally and externally. Internally, your threat intelligence platform is one obvious choice. If your threat intel platform is not actively dumping a list of malicious indicators for your SIEM to match on, you can potentially do the lookups live at search time as well. Another clever trick is to use the data already in the SIEM itself as a source to enrich your logs. While many SIEMs may do this by nature, others may need to have part of the log ingestion pipeline involve doing a SIEM query for matching data. An example of this: If you have your DNS logs (both query and response) in a database inside the SIEM, you can use it as a local passive DNS database. With the information available about every domain that has been looked up and the associated IP address, you could take IDS alerts or NetFlow that contain only an IP address and use the SIEM's knowledge of what domains have mapped to what IP address to attach a domain to it right off the bat. Every SIEM has a REST API that can be used to perform searches and return results, and most SIEMs have a way to enrich data with a web request. All you must do is marry these two capabilities together to have the SIEM look up things inside its databases.

This same technique could also work using external lookups, but care would have to be taken to ensure only a single domain was pulled back in the case of shared hosting or if it has changed over time. There are also ways to use REST API queries to enrich logs with randomness measures, creation dates from whois data, and Alexa ranks. For details on these methods, see the SANS Blue Team GitHub site for tools like freq.py and domain_stats, or classes like SEC511 and SEC555.

[1] https://github.com/sans-blue-team



Asset and User Based Enrichment

Vulnerability scanners:

- Known vulnerabilities, open ports, last scan date
- CVE # and vulnerability details

Asset database:

- Hostname, MAC, criticality, OS, type, …

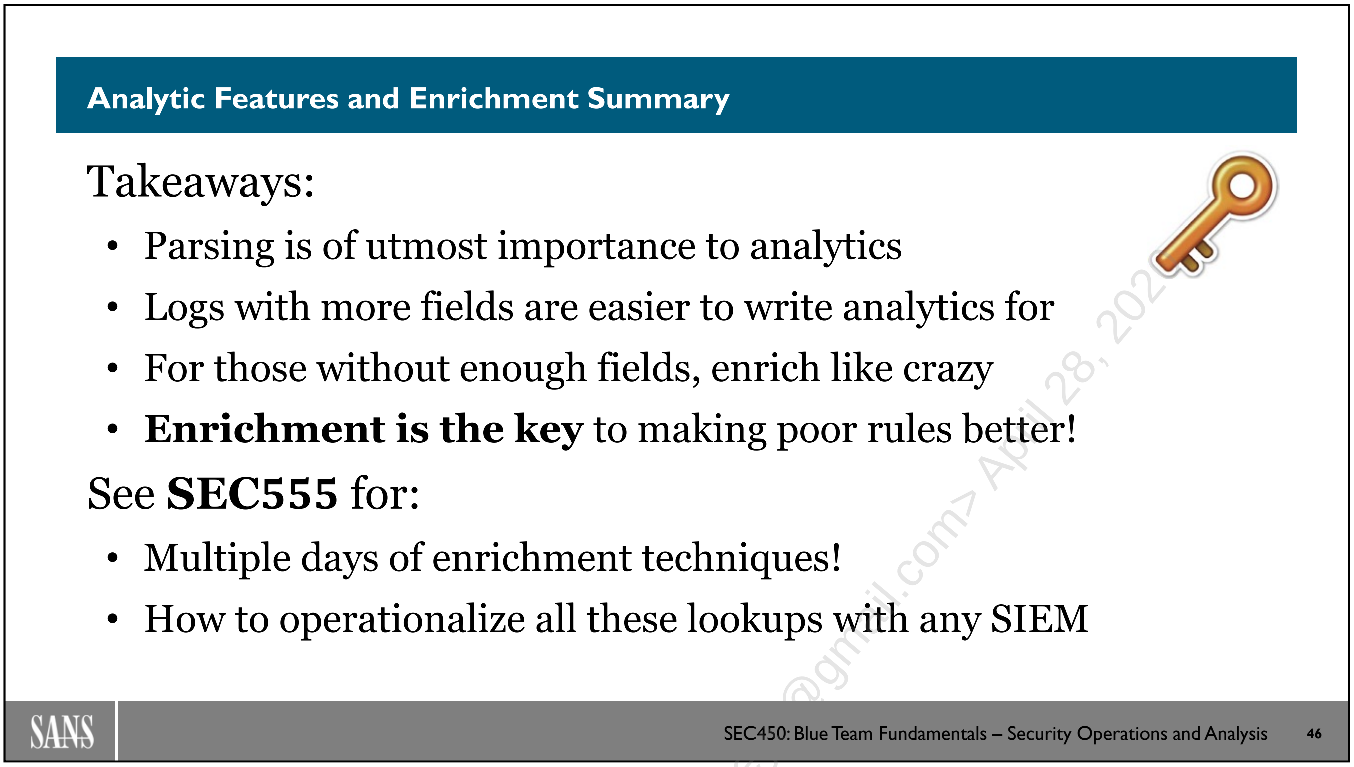
Active directory:

- User/host properties
- Group membership for user identified in event/alert
- Physical location

SANS
SEC450: Blue Team Fundamentals – Security Operations and Analysis
45

Asset and User Based Enrichment

Whether or not you store this type of data in the SIEM itself (some products will bring it in locally), your SIEM should also be enriching your logs with information about your assets and users. While it's great to have a username attached to a compromise, it's *much* better also to know what groups that person is in, where they sit, what type of host was compromised, the data from the last vulnerability scan, and how critical the item is to the business. Information like this, even if only looked up on the spot in an automated way, is a crucial part of correctly triaging alerts, and the SIEM should make access to it all very easy to put into your workflow. Most SIEMs will have built-in integration for information like this, but if not, the HTTP REST API is also a likely avenue to glue it together.



Analytic Features and Enrichment Summary

Takeaways:

- Parsing is of utmost importance to analytics
- Logs with more fields are easier to write analytics for
- For those without enough fields, enrich like crazy
- **Enrichment is the key** to making poor rules better!

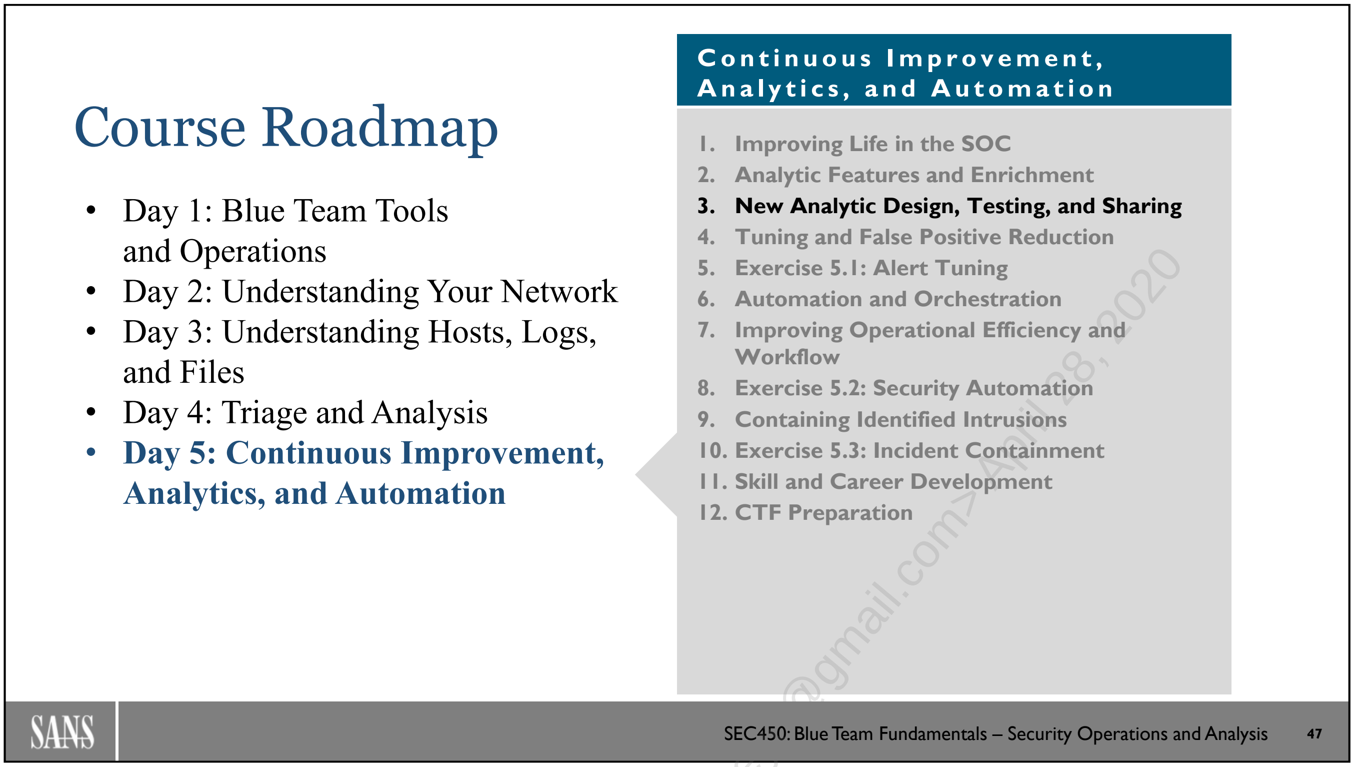
See **SEC555** for:

- Multiple days of enrichment techniques!
- How to operationalize all these lookups with any SIEM

SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 46

Analytic Features and Enrichment Summary

The takeaway here is that the first step in getting an effective analytic is extracting all the bits of information out of a log in a usable way, also known as feature extraction, or parsing the logs into its fields. Some logs will inherently contain more fields than others, giving them more information to inherently key off for high fidelity analytic writing. Other logs, however, may offer little detail and need some help. Whether high or low detail, almost all logs can benefit from enrichment of some sort. If you find yourself struggling to write a high-quality log, ask yourself "what piece of info, given the ability to have it inside this log, would help me tell true from false positive," then find a way to produce that as an enrichment. For those who want to go deep on these ideas, SEC555 is the course to head to. The author, Justin Henderson, has created days worth of content revolving around intelligence enrichment and analytics that can be written to use your SIEM in the best way possible.



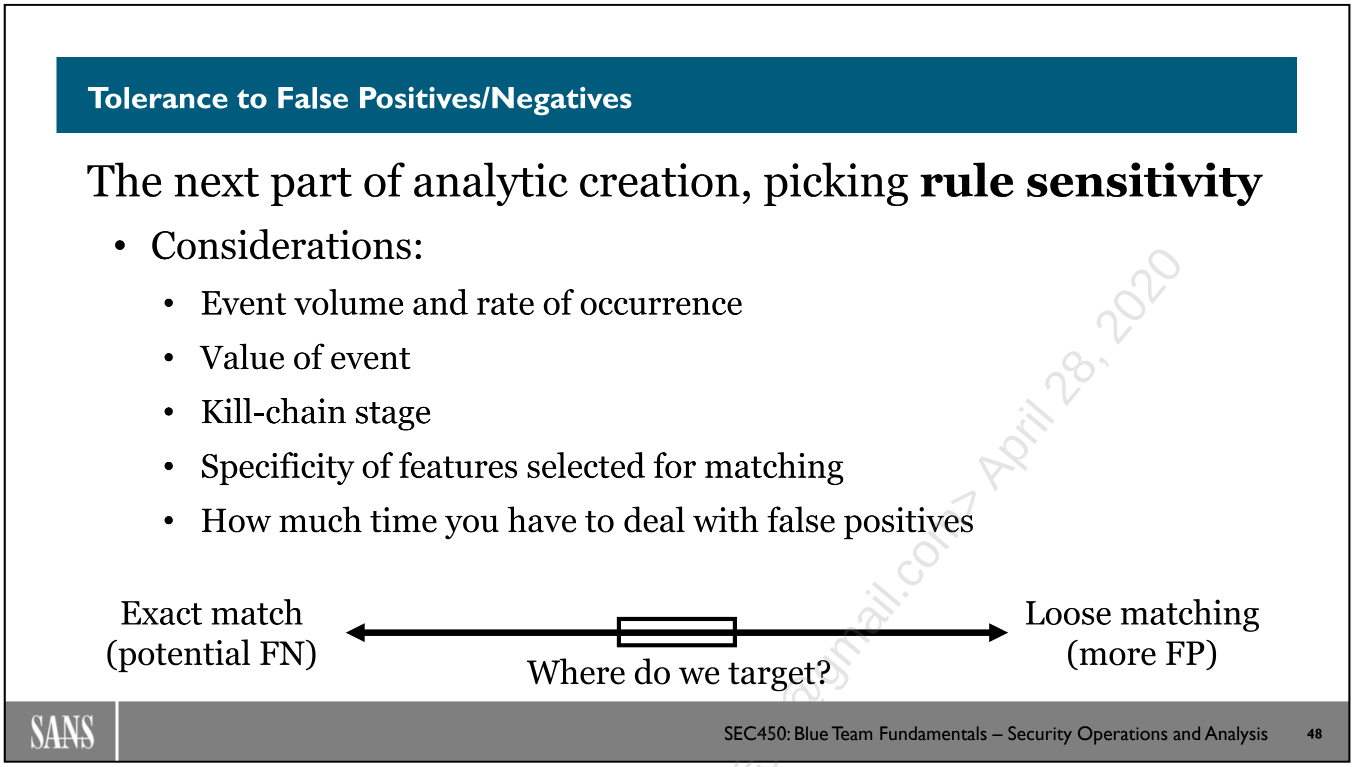
Course Roadmap

- 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**

Continuous Improvement, Analytics, and Automation

1. Improving Life in the SOC
2. Analytic Features and Enrichment
3. **New Analytic Design, Testing, and Sharing**
4. Tuning and False Positive Reduction
5. Exercise 5.1: Alert Tuning
6. Automation and Orchestration
7. Improving Operational Efficiency and Workflow
8. Exercise 5.2: Security Automation
9. Containing Identified Intrusions
10. Exercise 5.3: Incident Containment
11. Skill and Career Development
12. CTF Preparation

This page intentionally left blank.



Tolerance to False Positives/Negatives

The next part of analytic creation, picking **rule sensitivity**

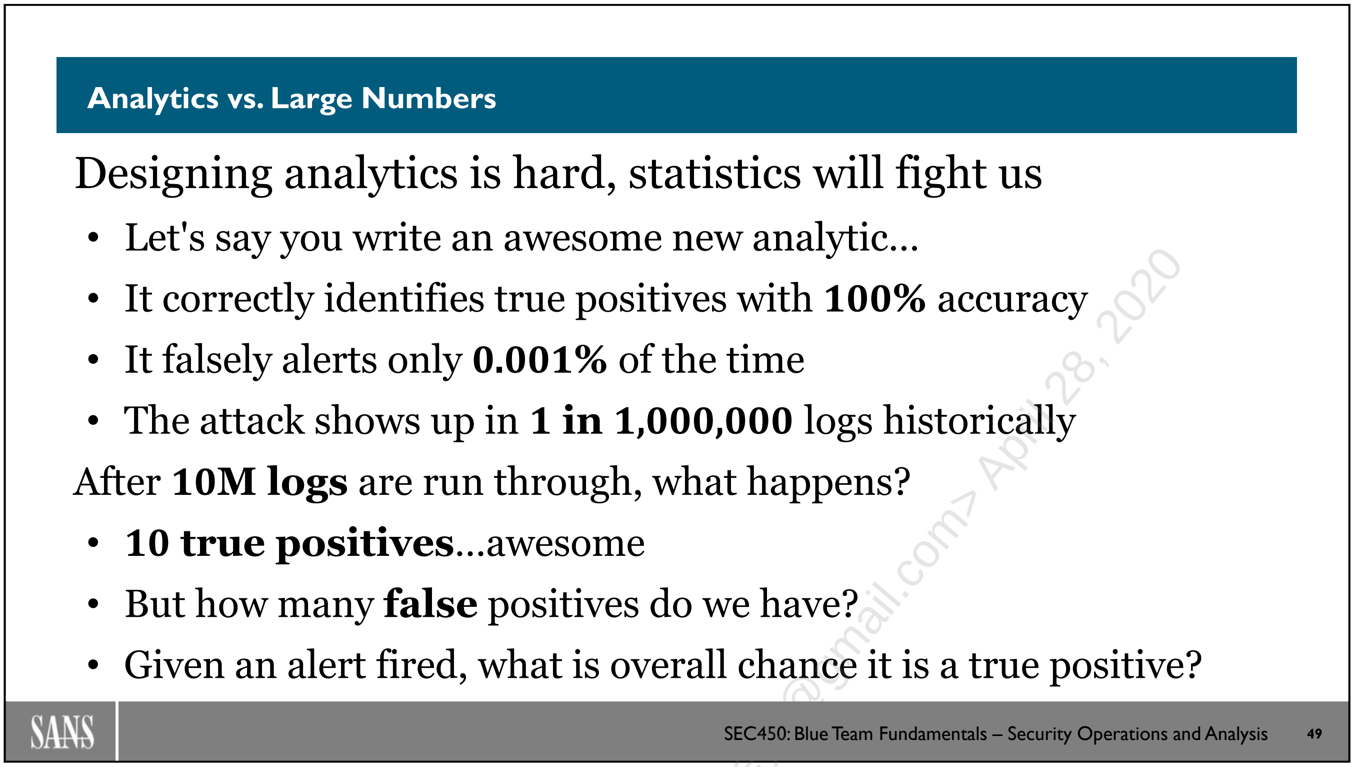
- Considerations:
  - Event volume and rate of occurrence
  - Value of event
  - Kill-chain stage
  - Specificity of features selected for matching
  - How much time you have to deal with false positives

Exact match
(potential FN)
Where do we target?
Loose matching
(more FP)

Tolerance to False Positives / Negatives

When designing an analytic, one of the things we must decide upon is its sensitivity. How we write a rule will heavily determine whether it is constantly going off and creating false positives, or perhaps rarely going off but missing things it could be catching. Given the spectrum of options, we must make a call on each rule based on several factors. One is the value of the event it is meant to catch. If the analytic is to catch a relatively inconsequential event like adware installation, we probably don't want to make it too easy to trigger because even when it does, it won't be the first thing we move to address. On the flip side, certain alerts are so important that we should be willing to accept the risk of false positives to ensure we catch them every single time. The event volume may play into this as well—if something that could be mistaken for a high-value event happens very frequently, although it's extremely important, we may need to be conservative in the rule log despite the high cost of missing the event.

How do we make this selection of sensitivity? Through the specificity and number of features we decide to match. We can write a rule that matches for only a specific condition, which will be great at that one thing, but what happens when attackers come along with a variation? Rules written to catch a single condition with laser focus may never create false positives, but this opens the potential for false negatives, which are worse. On the other side, if we match only a few general features, we will undoubtedly catch anything that resembles what we are looking for; the problem is we will also catch so much more in the process we may be unable to find the signal in the noise. The correct tuning lies somewhere between the extremes, and the considerations mentioned above will help guide you in deciding where to set the bar.



Analytics vs. Large Numbers

Designing analytics is hard, statistics will fight us

- Let's say you write an awesome new analytic…
- It correctly identifies true positives with **100%** accuracy
- It falsely alerts only **0.001%** of the time
- The attack shows up in **1 in 1,000,000** logs historically

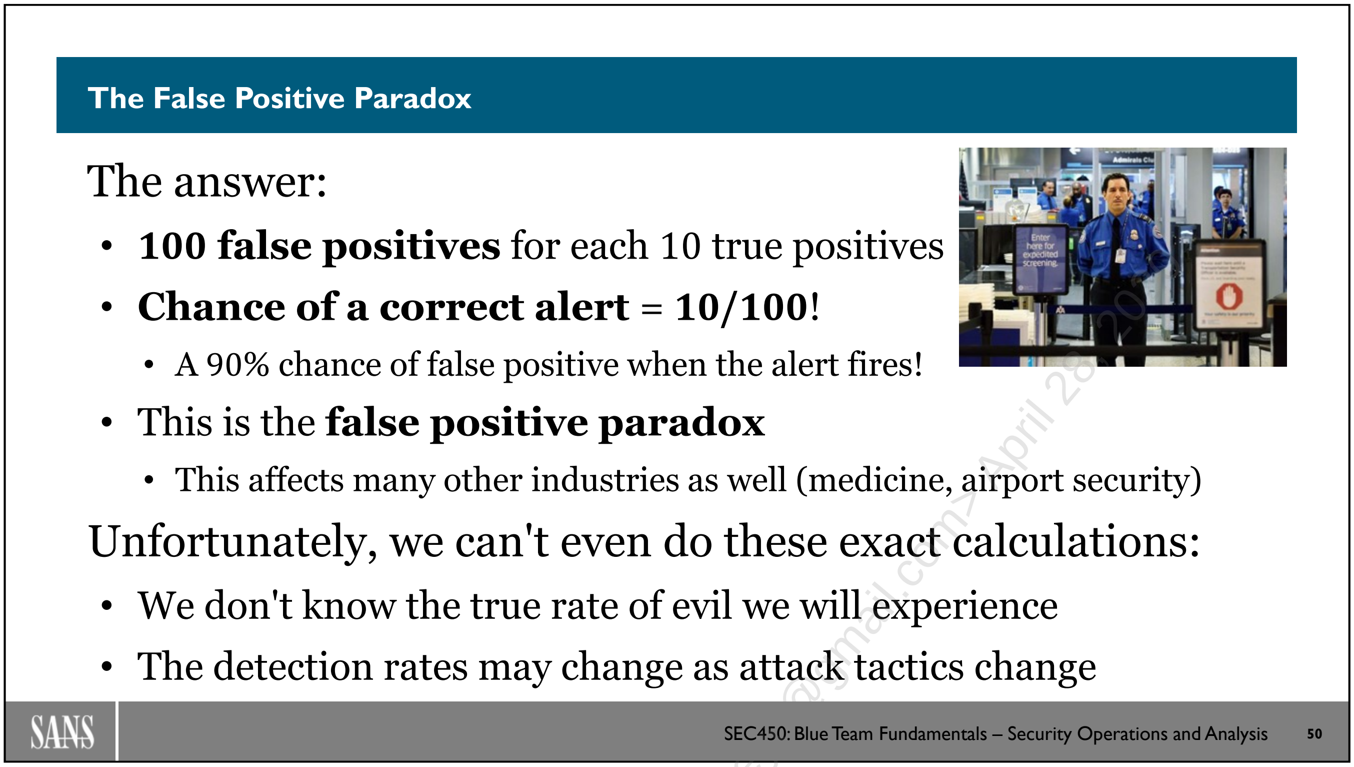
After **10M logs** are run through, what happens?

- **10 true positives**…awesome
- But how many **false** positives do we have?
- Given an alert fired, what is overall chance it is a true positive?

Analytics vs. Large Numbers

An interesting thing happens when writing tests for events that occur in a very small part of a large population. Even when you create a very strong test, the large numbers involved can overwhelm the quality of the test and lead to a condition where we create many more false positives than true positives. Here's an example: Let's say we set out to write an analytic to catch an attack that historically was found to occur at a rate of 1 in 1,000,000 of our logs. We write our new rule in a way that is 100% correct at identifying all *true positives*, which is great. We won't ever miss anything in a false negative; that's exactly what we want. Because of the lack of false negatives, the rule also requires that we accept a false positive rate of 0.001%—sounds like a pretty good rate, right?

Before we make it active, we take our new rule and do an exploratory search on our historical logs to ensure it's not going to explode the alert queue. We run our sample of 10M logs across the analytic and to ensure it will work, and what are the results? We'd get 10 true positives, which makes sense; 10M logs multiplied by a 1 in a million event = 10 true positives. What about the false positives, though?



The False Positive Paradox

The answer:

- **100 false positives** for each 10 true positives
- **Chance of a correct alert = 10/100!**
  - A 90% chance of false positive when the alert fires!
- This is the **false positive paradox**
  - This affects many other industries as well (medicine, airport security)

Unfortunately, we can't even do these exact calculations:

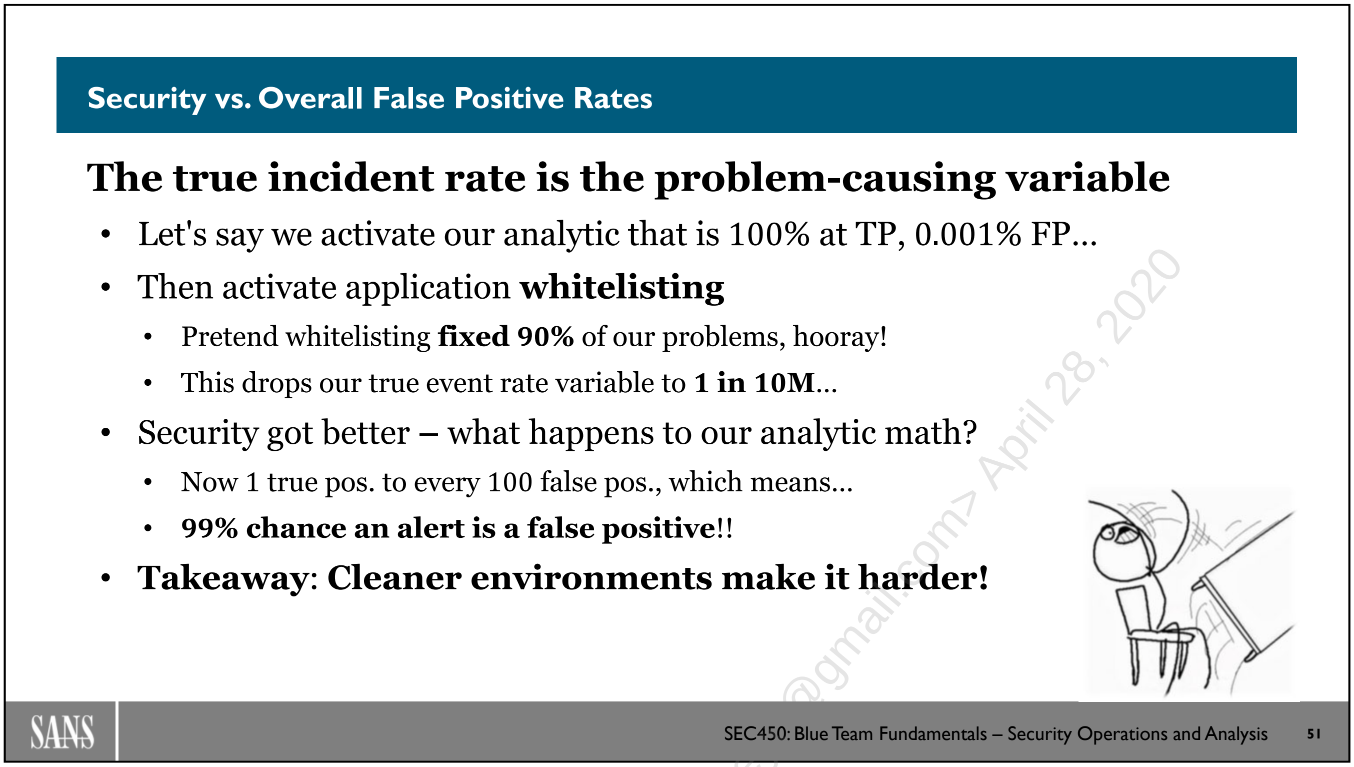
- We don't know the true rate of evil we will experience
- The detection rates may change as attack tactics change

The False Positive Paradox

The problem is we have also generated 100 false positives (0.001% of 10M) in the process! Looking at the totals of 10 true positives and 100 false positive means that overall 90% of the time the alert fires, it will be a false positive! Not such a great analytic after all…this is the false positive paradox, otherwise known as the base rate fallacy.[1]

This unfortunate situation is simply a consequence of the rates and large numbers involved. Detecting an extremely low probability event, even with a good detection analytic, is bound to make more false positives than true positives. We aren't the only industry that fights this; medicine has the same problem for diagnosing patients with rare diseases, and the TSA has the same issue screening people at the airport. An absurdly small fraction of the population is truly dangerous, but given the high number of flyers, even if they are 100% accurate at screening people with ill-intent and have a tiny false positive rate, that still means thousands of us will get incorrectly identified as potentially suspicious.

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



Security vs. Overall False Positive Rates

The true incident rate is the problem-causing variable

- Let's say we activate our analytic that is 100% at TP, 0.001% FP…
- Then activate application **whitelisting**
  - Pretend whitelisting **fixed 90%** of our problems, hooray!
  - This drops our true event rate variable to **1 in 10M**…
- Security got better – what happens to our analytic math?
  - Now 1 true pos. to every 100 false pos., which means…
  - **99% chance an alert is a false positive**!!
- **Takeaway**: **Cleaner environments make it harder!**

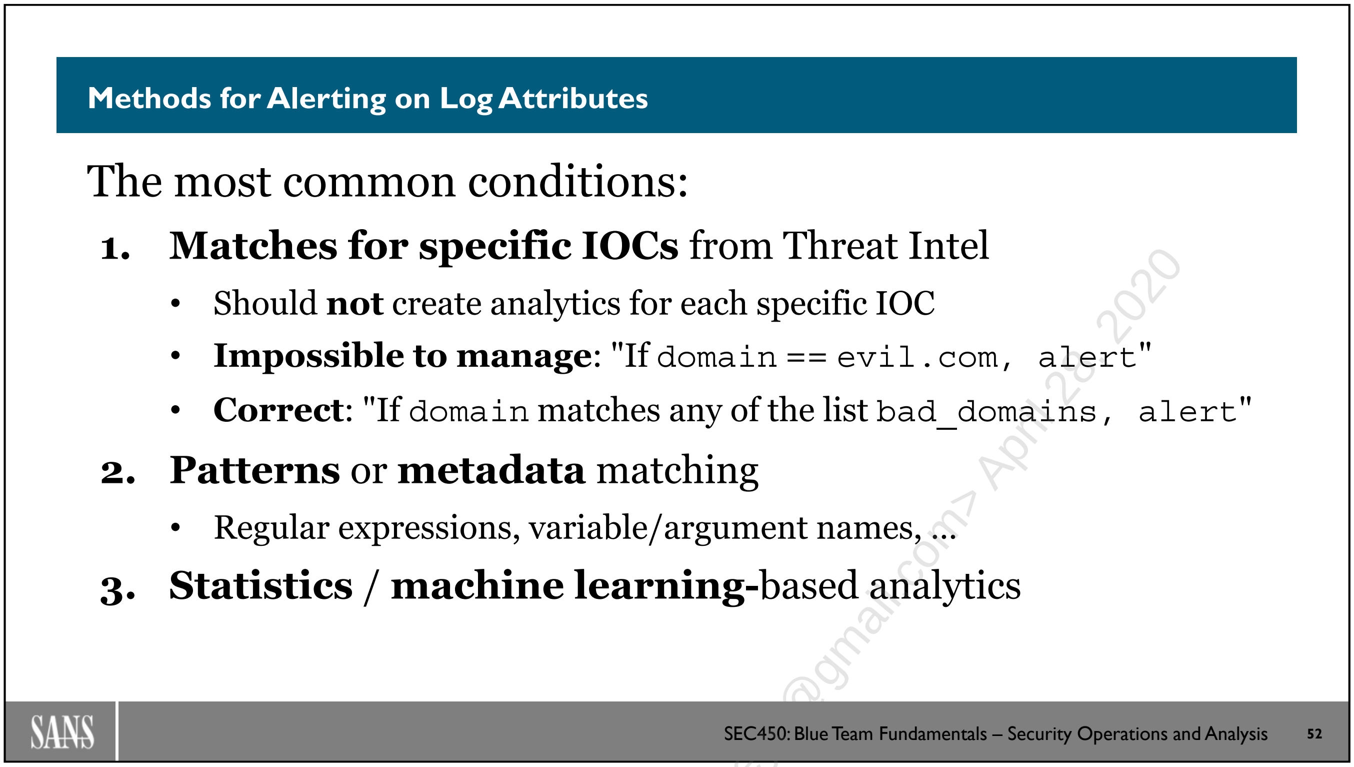
SANS
SEC450: Blue Team Fundamentals – Security Operations and Analysis 51

Security vs. Overall False Positive Rates

Not only is it difficult to truly understand the true rate of malicious logs in the environment and measure our true and false positive rates, we have the rate of occurrence variable fighting us as well. Ultimately, the SOC exists to prevent and detect security incidents, and to do this, occasionally, we may put in new preventive controls. Although a new preventive control will stop things from happening, what might it do to our analytic accuracy rates?

If we put better prevention controls in place, the count of successful attacks will drop. Let's say we put in an application whitelisting solution and have reduced our malicious application install rate by 10-fold. We have improved security undoubtedly, but using the numbers from the previous page, that means the true rate of malicious events occurring in the equation has now moved to 1 in 10M instead of 1 in 1M.

What happens to our overall chance of a true positive with this analytic? With 90% less true malicious events in the environment, we will now see one true positive for every 10M events. The problem is the false positive generation rate hasn't changed, meaning we still generate 100 false positives for 10M events. That means *improving security has brought our overall reliability of this analytic from 90% false positives to 1 in 100 or 99% false positives*!! The implication here is that the better you get at security, the harder it becomes not to generate a higher percentage of false positives—you can now see why false positives are such a problem and why they are so hard to eliminate.



Methods for Alerting on Log Attributes

The most common conditions:

1. **Matches for specific IOCs** from Threat Intel
   - Should **not** create analytics for each specific IOC
   - **Impossible to manage**: "If `domain == evil.com, alert`"
   - **Correct**: "If `domain` matches any of the list `bad_domains, alert`"
2. **Patterns** or **metadata** matching
   - Regular expressions, variable/argument names, ...
3. **Statistics / machine learning-**based analytics

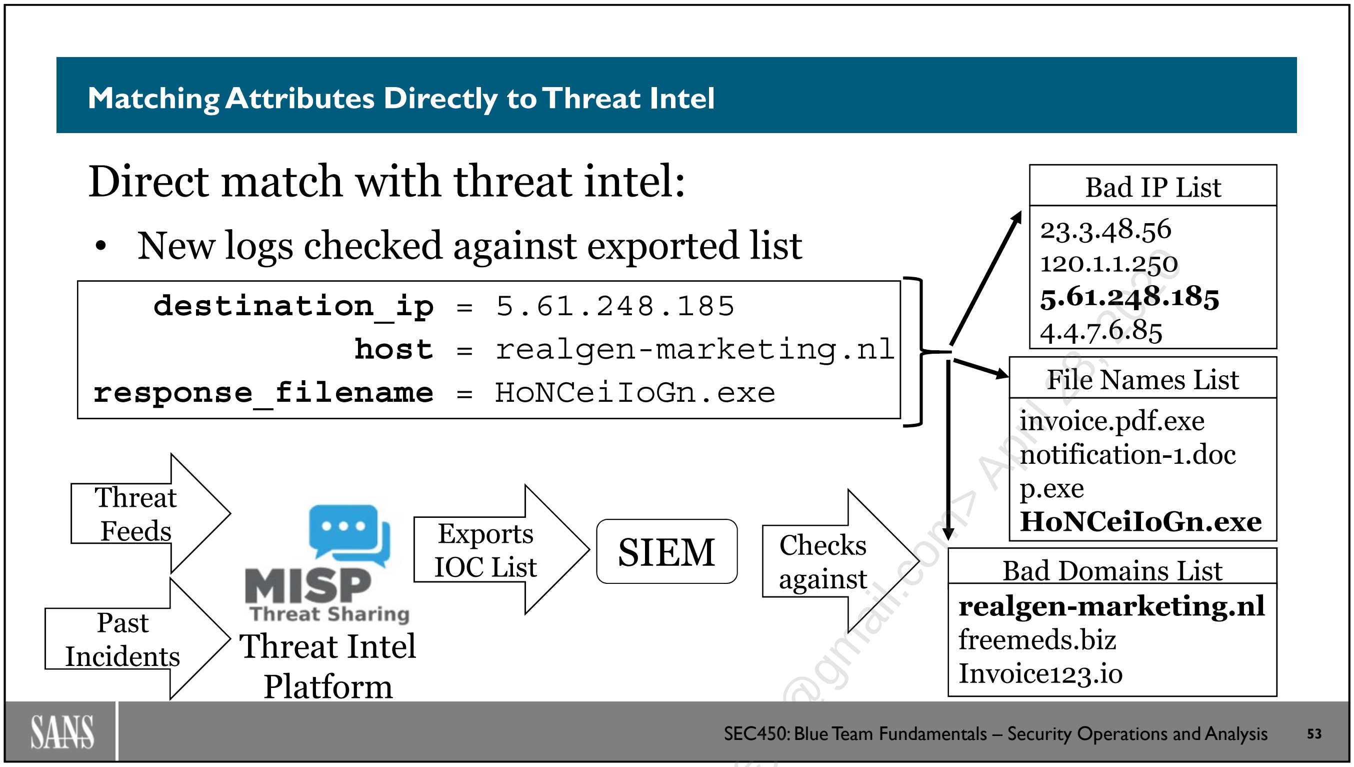
SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 52

Methods for Alerting on Log Attributes

While performing exact matches for the values in our log fields is probably the first type of analytic logic that comes to mind, there are many more types than that. With analytics for an exact match, there is a distinct "right" and "wrong" way to do it. The "wrong" way is to write a specific analytic looking for an IOC such as a rule that says "alert any time the domain evil.com is seen." This is wrong because, (think pyramid of pain), observables like this are rapidly changed by adversaries, and trying to keep with up atomic indicators that change this rapidly is a fool's bet. The right way of doing this type of detection is by applying each of the attributes for domain against a blacklist generated from your threat intelligence system. This way, your threat intel system manages what is actively good and bad. You do not have to change the rule, only write it to say, "if *domain* is ever on the list *bad_domains*, alert*."* This works based off the list brought in from MISP or whatever tool you use to collect malicious domain names.

Beyond the exact matching of values in a log field is the pattern or metadata-based match. These types of analytics don't look for an exact value in one of the fields, but instead, look for pattern match using regular expressions, or do some other type of meta-analyses such as counting the fields, looking at their order, or anything else of the sort. There are also statistics and machine-learning-based rules. These are the types that usually point out anomalies and although they may be great at pointing out anomalies, keep in mind that that doesn't necessarily mean they are evil. With some persistence, we can potentially use statistics and machine learning methods ourselves to manually come up with rules that can be applied to our traffic to find certain types of malware, although this type of investigation becomes complicated quite quickly.



Matching Attributes Directly to Threat Intel

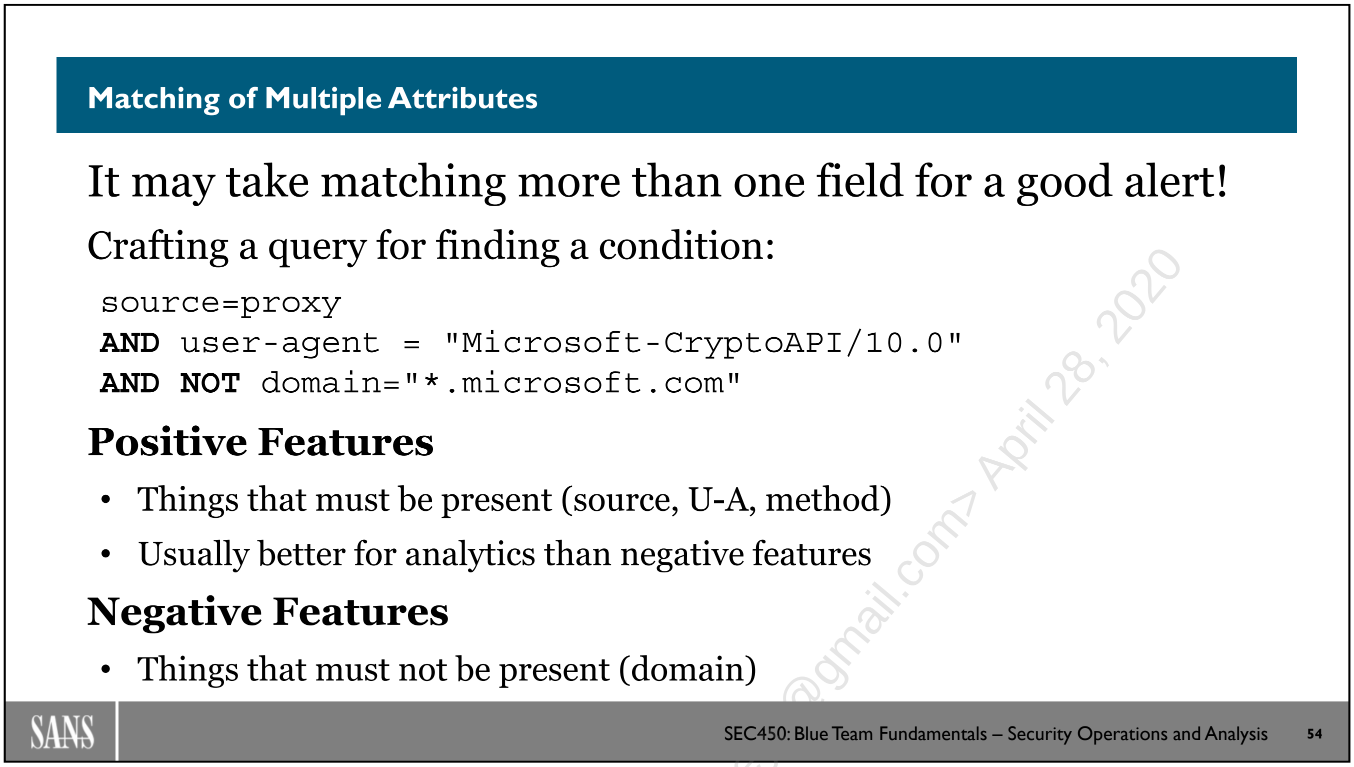
Direct match with threat intel:

- New logs checked against exported list

```
    destination_ip = 5.61.248.185
              host = realgen-marketing.nl
 response_filename = HoNCeiIoGn.exe
```

Matching Attributes Directly to Threat Intel

This slide has an illustration showing the information flow required to keep a robust rule for detection of specific values in your log fields. To start, a threat intel platform such as MISP (or maybe this feature is even built into your SIEM) collects what amounts to a list of known bad items. These can be domains, IP addresses, filenames, hashes, and everything else. However you do this, they must ultimately end up in a location where you can check every log coming into the SIEM that contains each of those attributes against the list. If you find anything on the list, an alert is sent telling the SOC that there has been a threat intelligence indicator match. That means if you bring in an HTTP log with the fields shown on the slide, you must check the IP against the list of known bad IP addresses, the filename against a list of known bad filenames, and the domain against any domain blacklist you have created. Assuming you have automated the process of feeding your threat intel platform with up-to-date indicators, and that you are feeding the information you find in previous incidents back into your TIP as well, these processes should be mostly self-running. You won't need to change the analytic itself, since it references the list that exports to the SIEM. You will not need to change the TIP because the feeds and incident management system are keeping the currently bad lists up to date for you. This process automates this loop as much as possible and keeps matching against known bad items maintainable.



Matching of Multiple Attributes

It may take matching more than one field for a good alert!

Crafting a query for finding a condition:

```
source=proxy
AND user-agent = "Microsoft-CryptoAPI/10.0"
AND NOT domain="*.microsoft.com"
```

Positive Features

- Things that must be present (source, U-A, method)
- Usually better for analytics than negative features

Negative Features

- Things that must not be present (domain)

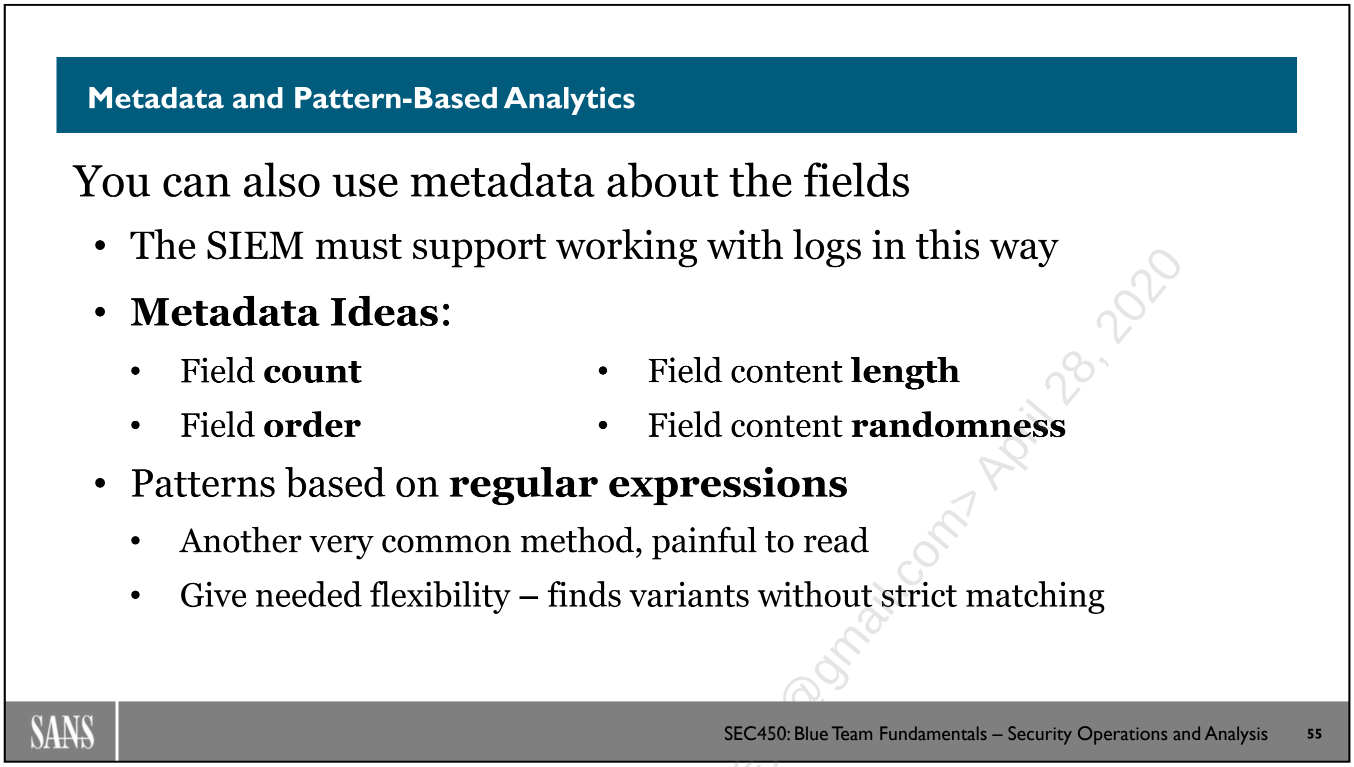
Matching of Multiple Attributes

One of the most basic types of rules is anything that matches a set of terms you can run as a search. Many times, however, one field will not be enough, so you will likely also run into analytics built off multiple log attribute matches.

For example, the search: source=proxy AND user-agent = "`Microsoft-CryptoAPI/10.0`" AND NOT domain="*.microsoft.com"

This pseudo-language search might find us all proxy traffic with a User-Agent of `Microsoft-CryptoAPI/10.0`, but NOT involving the domain *.microsoft.com. This type of rule might be whitelisting the one known good use of the `Microsoft-CryptoAPI/10.0` user-agent, and would flag on any malware trying to use the same user-agent when communicating with a malicious command and control domain. This would be a clever analytic, and a way to use what you know to stop malware from disguising itself.

These multi-condition searches are one of the most common methods for identifying traffic of interest on a network and consist of both positive and negative features. Positive features are the things that must be present (the source, user-agent, and method in this example). Negative features are things that must *not* be present (the specific domain name). In general, when writing an analytic, we want to be specific enough with our attribute identification that we will catch all occurrences of the malicious condition, but not *so* specific that slight variations will cause the analytic to miss something. For tuning this in, positive features tend to be more useful since defining what *should* be there is inherently more specific than defining what *shouldn't* be there. There are an infinite set of fields that are *not* present in every log.



Metadata and Pattern-Based Analytics

You can also use metadata about the fields

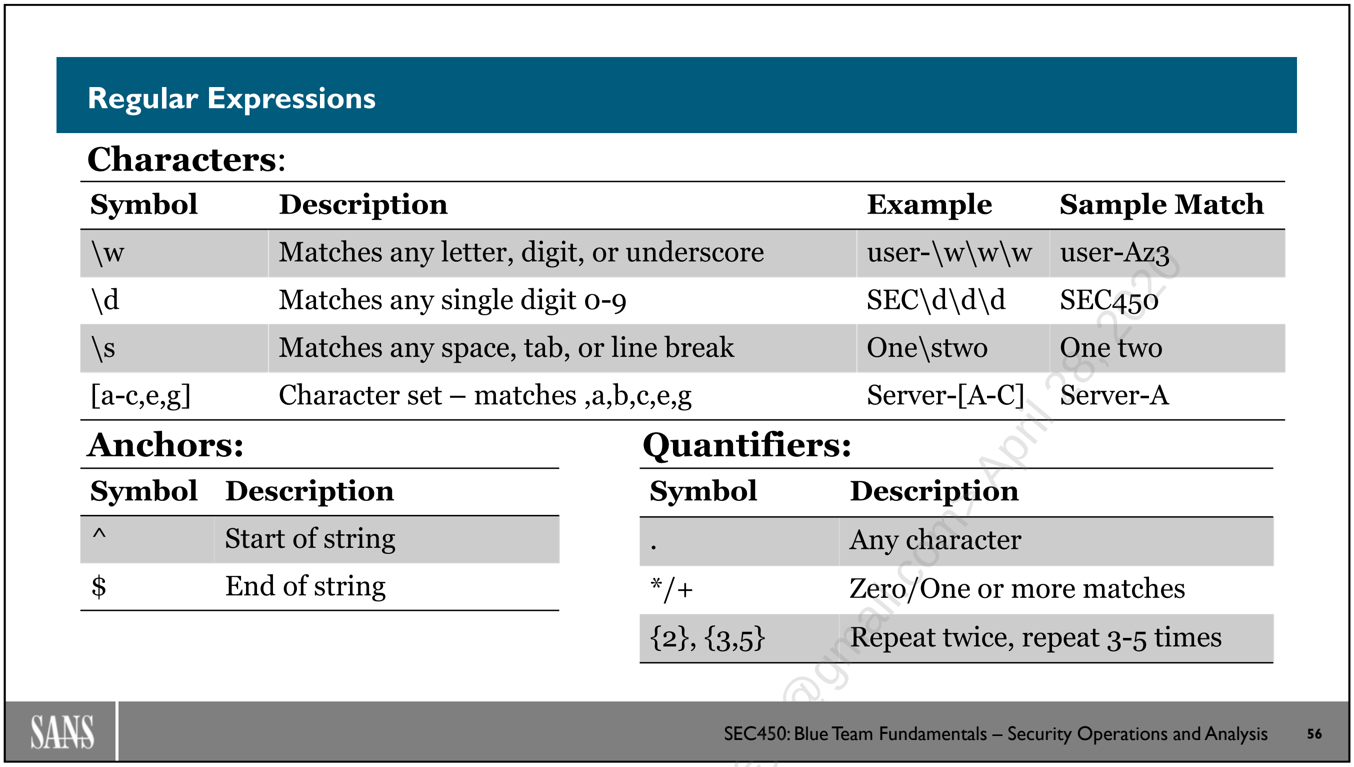
- The SIEM must support working with logs in this way
- **Metadata Ideas:**
  - Field **count**
  - Field **order**
  - Field content **length**
  - Field content **randomness**
- Patterns based on **regular expressions**
  - Another very common method, painful to read
  - Give needed flexibility – finds variants without strict matching

SANS
SEC450: Blue Team Fundamentals – Security Operations and Analysis
55

Metadata and Pattern-Based Analytics

One level of abstraction away from strict field content matching is moving into using metadata about the fields or writing patterns for their content. Metadata-based analytics don't rely on the actual content itself, but rather calculate some value or other meta-feature of the data. For example, measuring the **randomness** of a domain name recorded in a field, the **length** of a URL, a **count** of headers are in an HTTP transaction, or the **order** in which those headers appear. These can be slightly more difficult to implement depending on the type of SIEM you have.

There are also pattern-based alerts that do not look for exact content, but instead look for fields with characters that match a specific pattern. These types of rules make up an enormous amount of SIEM use cases as they are immensely powerful at finding multiple versions of an attack across time, even if the specific indicators have shifted. For example, malware often uses a specific callback pattern for command and control over HTTP. While each infection and version of the malware may use a new domain name, if you can instead write an analytic to catch the structure of the URL, then you will catch every version of that malware that exists until the authors decide to modify the code. This idea gets back to the pyramid of pain and how writing an alert based on tools is more of a pain for attackers than writing one based on easily changeable characteristics like a domain or IP address. The problem with regular expressions is that they can be confusing to write, and even *more* confusing to interpret what someone else wrote.



Regular Expressions

Characters:

Symbol	Description	Example	Sample Match
\w	Matches any letter, digit, or underscore	user-\w\w\w	user-Az3
\d	Matches any single digit 0-9	SEC\d\d\d	SEC450
\s	Matches any space, tab, or line break	One\stwo	One two
[a-c,e,g]	Character set – matches ,a,b,c,e,g	Server-[A-C]	Server-A

Anchors:

Symbol	Description
^	Start of string
$	End of string

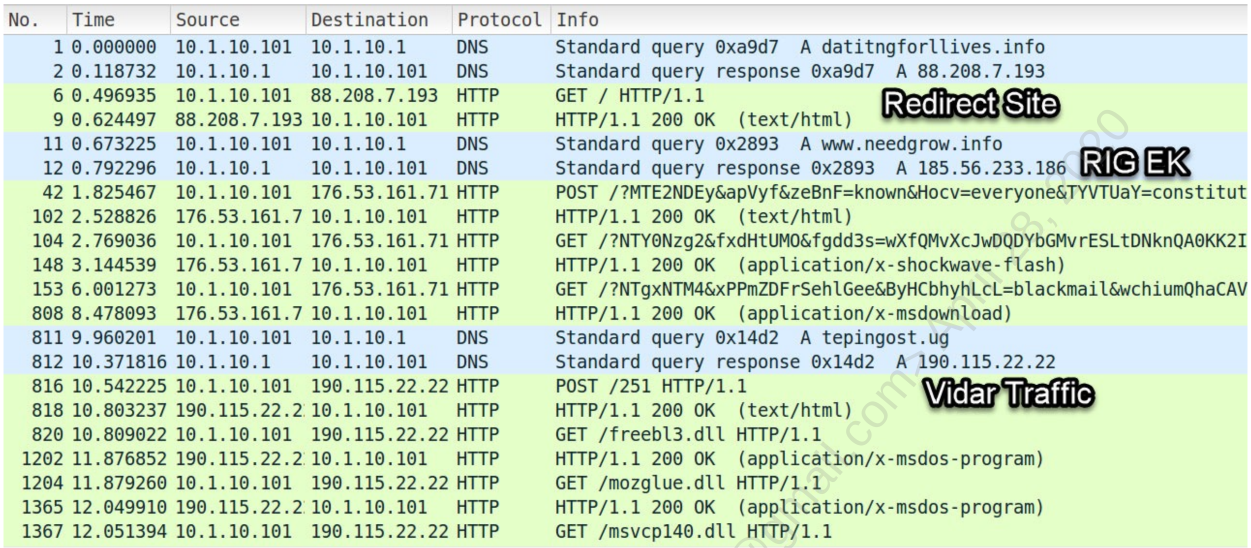
Quantifiers:

Symbol	Description
.	Any character
*/+	Zero/One or more matches
{2}, {3,5}	Repeat twice, repeat 3-5 times

Regular Expressions

To define parsers for unstructured logs as well as analytics for matching log content patterns, we use regular expressions. Here are some of the most common regular expression symbols you'll see if you look through any set of SIEM use cases or IDS signatures.

Vidar Infection Attribute Identification Challenge



```
 No.  Time      Source       Destination   Protocol Info
    1 0.000000  10.1.10.101  10.1.10.1     DNS      Standard query 0xa9d7  A datitngforllives.info
    2 0.118732  10.1.10.1    10.1.10.101   DNS      Standard query response 0xa9d7  A 88.208.7.193
    6 0.496935  10.1.10.101  88.208.7.193  HTTP     GET / HTTP/1.1
    9 0.624497  88.208.7.193 10.1.10.101   HTTP     HTTP/1.1 200 OK  (text/html)
   11 0.673225  10.1.10.101  10.1.10.1     DNS      Standard query 0x2893  A www.needgrow.info
   12 0.792296  10.1.10.1    10.1.10.101   DNS      Standard query response 0x2893  A 185.56.233.186
   42 1.825467  10.1.10.101  176.53.161.71 HTTP     POST /?MTE2NDEy&apVyf&zeBnF=known&Hocv=everyone&TYVTUaY=constitut
  102 2.528826  176.53.161.7 10.1.10.101   HTTP     HTTP/1.1 200 OK  (text/html)
  104 2.769036  10.1.10.101  176.53.161.71 HTTP     GET /?NTY0Nzg2&fxdHtUMO&fgdd3s=wXfQMvXcJwDQDYbGMvrESLtDNknQA0KK2I
  148 3.144539  176.53.161.7 10.1.10.101   HTTP     HTTP/1.1 200 OK  (application/x-shockwave-flash)
  153 6.001273  10.1.10.101  176.53.161.71 HTTP     GET /?NTgxNTM4&xPPmZDFrSehlGee&ByHCbhyhLcL=blackmail&wchiumQhaCAV
  808 8.478093  176.53.161.7 10.1.10.101   HTTP     HTTP/1.1 200 OK  (application/x-msdownload)
  811 9.960201  10.1.10.101  10.1.10.1     DNS      Standard query 0x14d2  A tepingost.ug
  812 10.371816 10.1.10.1    10.1.10.101   DNS      Standard query response 0x14d2  A 190.115.22.22
  816 10.542225 10.1.10.101  190.115.22.22 HTTP     POST /251 HTTP/1.1
  818 10.803237 190.115.22.2 10.1.10.101   HTTP     HTTP/1.1 200 OK  (text/html)
  820 10.809022 10.1.10.101  190.115.22.22 HTTP     GET /freebl3.dll HTTP/1.1
 1202 11.876852 190.115.22.2 10.1.10.101   HTTP     HTTP/1.1 200 OK  (application/x-msdos-program)
 1204 11.879260 10.1.10.101  190.115.22.22 HTTP     GET /mozglue.dll HTTP/1.1
 1365 12.049910 190.115.22.2 10.1.10.101   HTTP     HTTP/1.1 200 OK  (application/x-msdos-program)
 1367 12.051394 10.1.10.101  190.115.22.22 HTTP     GET /msvcp140.dll HTTP/1.1
```

Vidar Infection Attribute Identification Challenge

Here is an example of actual malicious traffic from January 2019 posted on the awesome malware-traffic-analysis.net blog. It contains a malicious redirect to a site hosting the "RIG" exploit kit, which then delivers "Vidar" malware, an info-stealing trojan[1]. If this traffic were turned into a log, consider the attributes you could extract to find similar occurrences of this attack.

Remember the chain of events here. In this single screenshot, we see delivery and exploitation can infer installation, and likely command and control. Here's a rough breakdown of where each occurs:

[Delivery]

- User hits a site with a malicious redirect (datitingforllives.info) that sends them to the next step
- User lands on domain hosting the RIG Exploit Kit (www.needgrow.info)

[Exploit]

- The exploit is delivered to the user (shockwave flash file, packet #148)

[Installation]

- If successful, the exploit sends malware for the user to run (x-msdownload mime type file, packet #808)

[Command and Control]

- Vidar malware runs and resolves its C2 server via DNS (Tepingost.ug)
- Malware starts performing command and control and additional downloads (GETs for .dll traffic, packets #820+)

How might we write analytics to catch these items given what we see here?

[1] http://malware-traffic-analysis.net/2019/01/10/index2.html

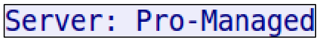
Feature Ranking Example: Which Are Best to Use?

Normal features:

- GET/POST requests, HTTP 1.1
- 200 OK Response

Semi-unique features:

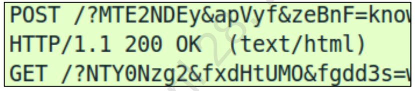
- **.info** and **.ug** domain traffic
- Shockwave **flash**, **exe** file mime type
- Long URIs with lots of parameters
- Host: Header shows IP, not domain
- "Server: Pro-Managed" header for Vidar



```
Server: Pro-Managed
```

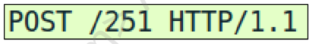
Potentially unique features:

- GET/POST matching regular expression pattern `"/\?\w{8}&\w{5,13}"`



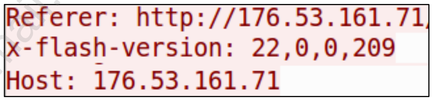
```
POST /?MTE2NDEy&apVyf&zeBnF=kno
HTTP/1.1 200 OK  (text/html)
GET /?NTY0Nzg2&fxdHtUMO&fgdd3s=
```

- POST to / with a number (Vidar)



```
POST /251 HTTP/1.1
```

- X-flash-version header, IP host and referer



```
Referer: http://176.53.161.71
x-flash-version: 22,0,0,209
Host: 176.53.161.71
```

Feature Ranking Example: Which Are Best to Use?

What might we be able to pull out of that traffic for detecting other RIG Exploit Kit delivery attempts or Vidar infections? The first step is to sort the features into what we think is and is not unique. The slide shows some of the things that may jump out when performing this task. Note that some items like the HTTP headers weren't shown on the previous page but would be available if you were performing this analysis and therefore are included here.

Under the list of normal items, we would put the fact that in terms of the HTTP method, version, and response, there is nothing atypical about the situation, so it's unlikely we would get any useful identifying info out of those features alone. We do have some potentially semi-unique features, though. One is the TLD of the traffic—how often do you visit .info or .ug (Uganda) domains? Alone it isn't enough to raise a flag on, but when combined with other details may be indicative. We also see the flash and executable mime types in the download. How often are you downloading an executable from a Uganda TLD? It depends on your organization, but I'm going to guess for most people, it's not very common. We then have the fact that all the URIs are extremely long and contain lots of parameters. If we had an enrichment capability to break up the GET parameters in a URI, and perhaps count how many there were, the exploit kit we may find to be atypical. If we look at the HTTP (which weren't shown on the previous page but are here), both the GET and POST request traffic use a Host: header of an IP address. Since most sites operate via domain names, this is also slightly odd. The final potentially semi-unique item spotted was the response from the server for the Vidar traffic showed "Server: Pro-Managed." This server is not one of the typical web servers commonly in use, so with some research, we may find this to be either semi or unique to this type of infection.

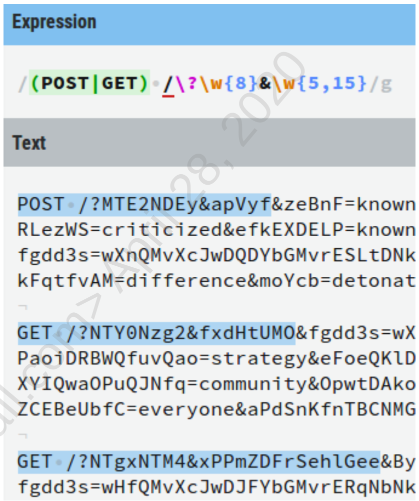
On the likely unique side, we can look a little deeper at the features we do have. For the POST requests, the URI format has a repetitive nature that we can write a regular expression to match. The one provided on the slide simply means a slash, followed by a question mark, followed by eight characters of 0-9, a-z, or A-Z, then an

"&" sign, followed by another 5-13 character of the same set. With additional samples of this exploit kit and some exploratory searching into the organization's URIs at large, we may find this is a great way to identify RIG exploit kit deliveries. Another odd-looking item is a POST request to a top-level folder with only a number. Websites do not usually have users POST to URI formats like this, so it as well could be used as a unique feature if a search of history proves this theory to be true. Finally, we have the presence of a flash version header. By itself, that's not surprising. All flash file transactions may or may not use this but combine that with the fact that we have a host header to a "naked IP" address instead of a domain name, and a referrer that was also an IP, and this situation quickly becomes unique. Although we don't know for sure, it's probably a safe bet that an analytic based on a flash download from a raw IP that was referred there by another raw IP address is not a usual situation, especially given the rapid phase-out of flash on the internet.

Testing Patterns with Regexr

If you haven't used regular expressions, time to learn!

- The best way: **regexr.com**
- Interactive regex interpretation
- Cheat sheets for reference
- Window for live highlighting
- Great for pattern testing with logs
- Testing GET/POST from RIG EK
  - It works!



Expression

```
/(POST|GET) /\?\w{8}&\w{5,15}/g
```

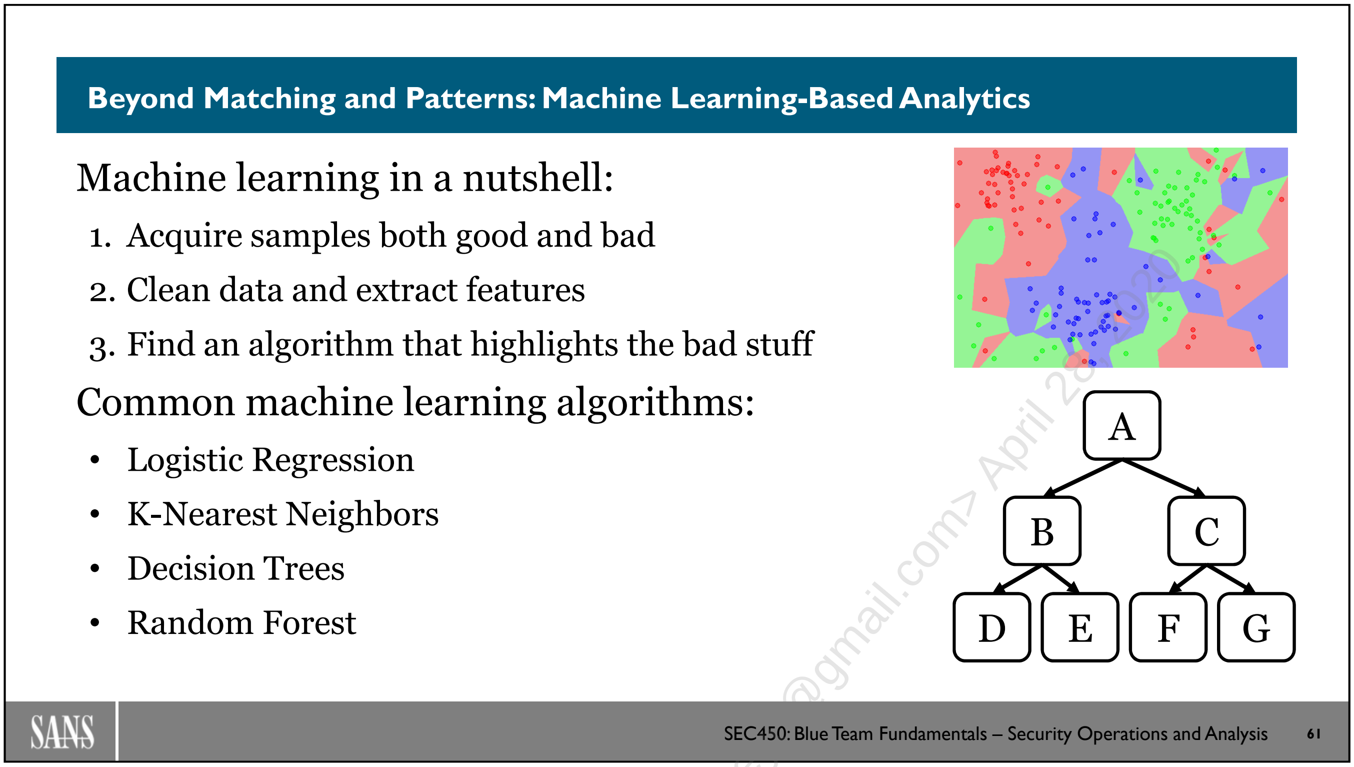
Text

```
POST /?MTE2NDEy&apVyf&zeBnF=known
RLezWS=criticized&efkEXDELP=known
fgdd3s=wXnQMvXcJwDQDYbGMvrESLtDNk
kFqtfvAM=difference&moYcb=detonat

GET /?NTY0Nzg2&fxdHtUMO&fgdd3s=wX
PaoiDRBWQfuvQao=strategy&eFoeQKlD
XYIQwaOPuQJNfq=community&OpwtDAko
ZCEBeUbfC=everyone&aPdSnKfnTBCNMG

GET /?NTgxNTM4&xPPmZDFrSehlGee&By
fgdd3s=wHfQMvXcJwDJFYbGMvrERqNbNk
```

Testing Patterns with Regexr

If we wanted to develop a regular expression with some assuredness that it will work when applied to the logs, the website **regexr.com** is an outstanding resource to do so. On regexr, there is a box for writing your regular expression and a text box where you can paste a large sample of logs. The idea is to take a sample of both known true positive logs you'd like to match as well as some regular logs that aren't a match and put them in the text box. As you write and develop your regular expression, the matching parts of your log samples will be highlighted live in the window below. If you aren't sure how to write the expression you need, there is a sidebar with cheat sheets, and hovering the mouse over what you have already written will spell out for you what each piece of the expressions is doing. Regular expression writing can be intimidating for beginners and confusing to understand when someone else has written the pattern; regexr.com is the solution to that problem!

A secondary option to regexr, which is also fantastic, is **regex101.com**.



Beyond Matching and Patterns: Machine Learning-Based Analytics

Machine learning in a nutshell:

1. Acquire samples both good and bad
2. Clean data and extract features
3. Find an algorithm that highlights the bad stuff

Common machine learning algorithms:

- Logistic Regression
- K-Nearest Neighbors
- Decision Trees
- Random Forest

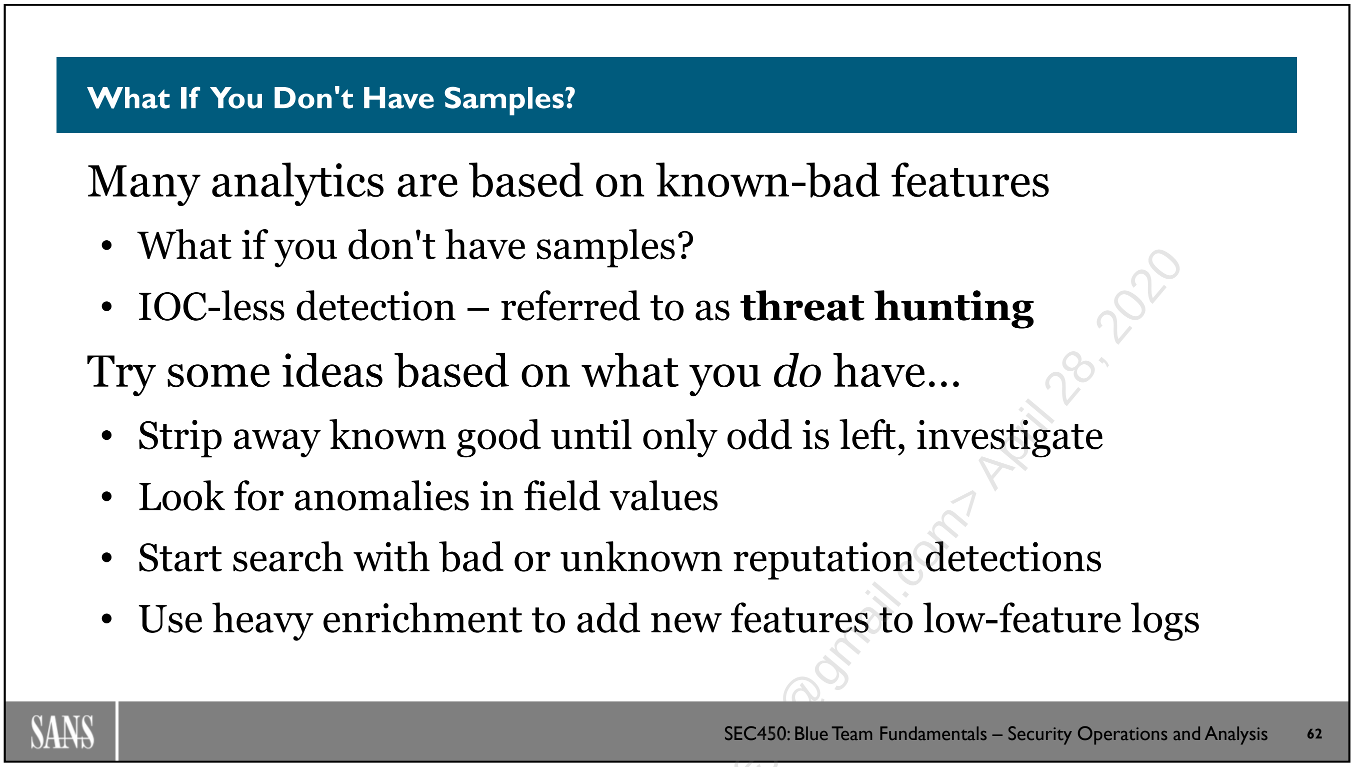
SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 61

Beyond Matching and Patterns: Machine Learning-Based Analytics

While way beyond the scope of this class, you will undoubtedly hear vendors bragging about their ability to detect malicious activity based on their proprietary, amazing machine learning algorithms. What is really going on with these algorithms? Is it truly the futuristic technology it's pitched as? The items we have already discussed give you the base knowledge to understand how we use machine learning. Machine learning is ultimately about extracting as many features out of a population of data as possible—whether that's logs, network traffic, or files. Once we extract those features, multiple mathematical algorithms and approaches can be taken to visualize and sort the data using different facets and combinations of the features until hopefully one is discovered that can usefully divide good from bad. When there is a gray area, the algorithms must still decide on whether they lean toward more false positives to drive down false negatives, or the other way around.

Is machine learning and AI a magic bullet destined to eliminate the entry-level analysts as the industry has been predicting? So far, we haven't quite seen that. I do think, however, that automation and AI are likely to shift the roles of those jobs in a positive direction. We have already discussed how automation and orchestration can make our lives not only easier but much more pleasant by eliminating repetitive manual work. Machine learning will no doubt be a contributing factor to this, driving the ability to understand easy to triage alerts in an automated way, leaving the more interesting stuff for analysts to manually work through.

If you're interested in diving deep into machine learning, keep your eyes open for a brand-new course offering from SANS … it is coming soon!



What If You Don't Have Samples?

Many analytics are based on known-bad features

- What if you don't have samples?
- IOC-less detection – referred to as **threat hunting**

Try some ideas based on what you *do* have...

- Strip away known good until only odd is left, investigate
- Look for anomalies in field values
- Start search with bad or unknown reputation detections
- Use heavy enrichment to add new features to low-feature logs

SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 62

What If You Don't Have Samples?

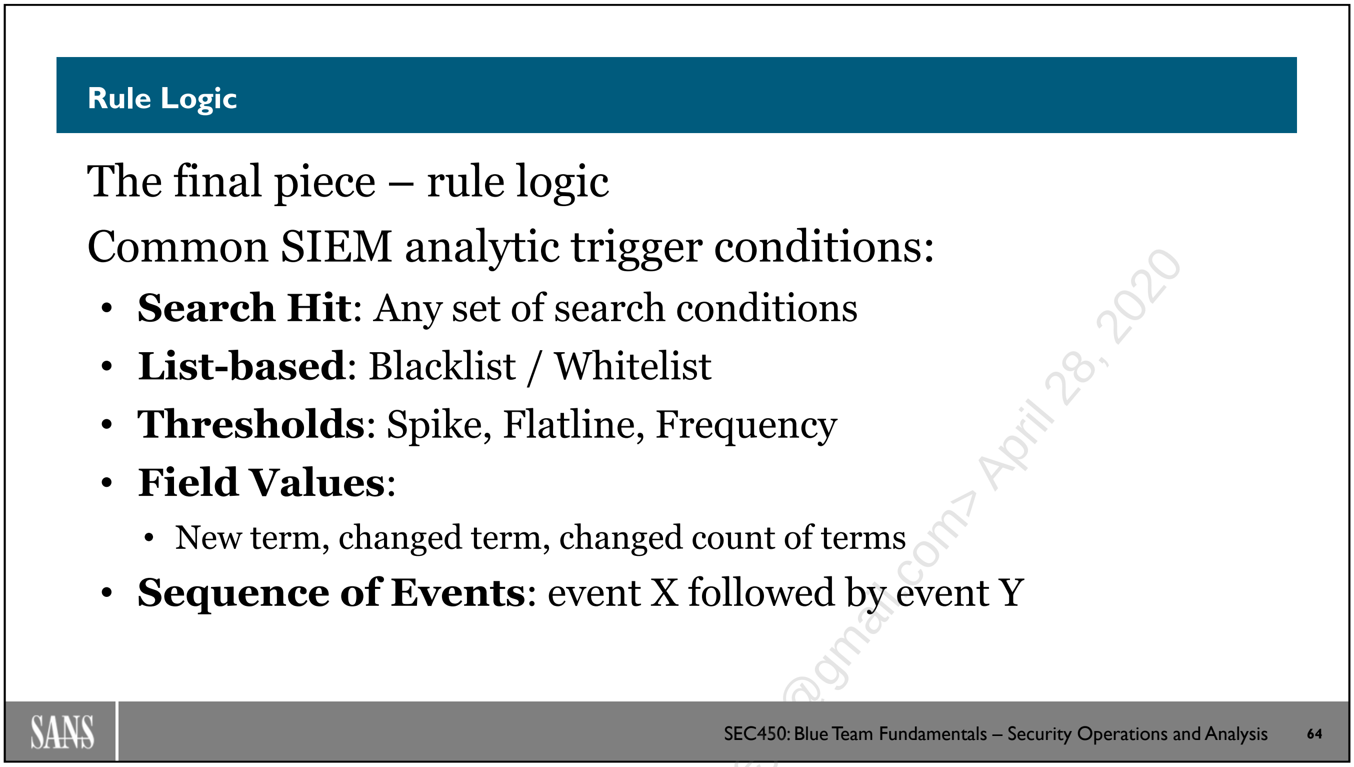
While we based many analytics on known-bad features, perhaps you're setting off to find new ways to identify malicious behavior and, therefore, don't have any specific samples to go off.

Setting off to develop new analytics from scratch is fun and is often referred to as "threat hunting," a way of doing detection in a manual capacity, not driven by IOCs. While doing this for the first time can give you a feeling of not knowing where to start, there is not necessarily a "wrong" way to go about it, just ways that are more efficient. As you see more and more malicious samples, your ability to intuitively come up with ideas will grow. Here are some things you might try to get a head start:

- Taking inspiration from current, working analytics and modifying it. A walkthrough your current Snort or other HIDS rules may be a great way to get inspiration for how you can tweak an existing analytic to find something new.
- Subtracting all known good and seeing what is left in the pile. This is anomaly identification and likely to lead you toward malware, but you will still need to separate odd from evil.
- Starting with items that already have some sort of pre-indication that they could be bad. Many proxy logs come with a reputation score for domains visited. While it would be crazy to try to investigate all hits with a less-than-stellar reputation, using that as a subset to start with, then applying additional filtering can more quickly lead to interesting items.
- Using enrichments to add additional fields to filter on. Though DNS and proxy logs won't necessarily come with rank, creation date, ASN, or other bits of data, if your SIEM can add them, this is another great way to pre-filter traffic down to what is likely more interesting.
- Looking for statistical anomalies. By sorting logs into groups based on the contents of fields, you may be able to discern items that stick out within a population by sorting by frequency and looking at the

bottom of the list. This method is also called "long tail analysis"—looking at the least common items in a data set, another method of anomaly detection. For example, looking at all user-agents seen on the network, and finding the least commonly occurring ones can highlight that single malware infection using a unique user-agent.

- Attack knowledge frameworks: If you do not know how attacks are performed, lists like the MITRE ATT&CK framework can be a great source of inspiration for this type of work.



Rule Logic

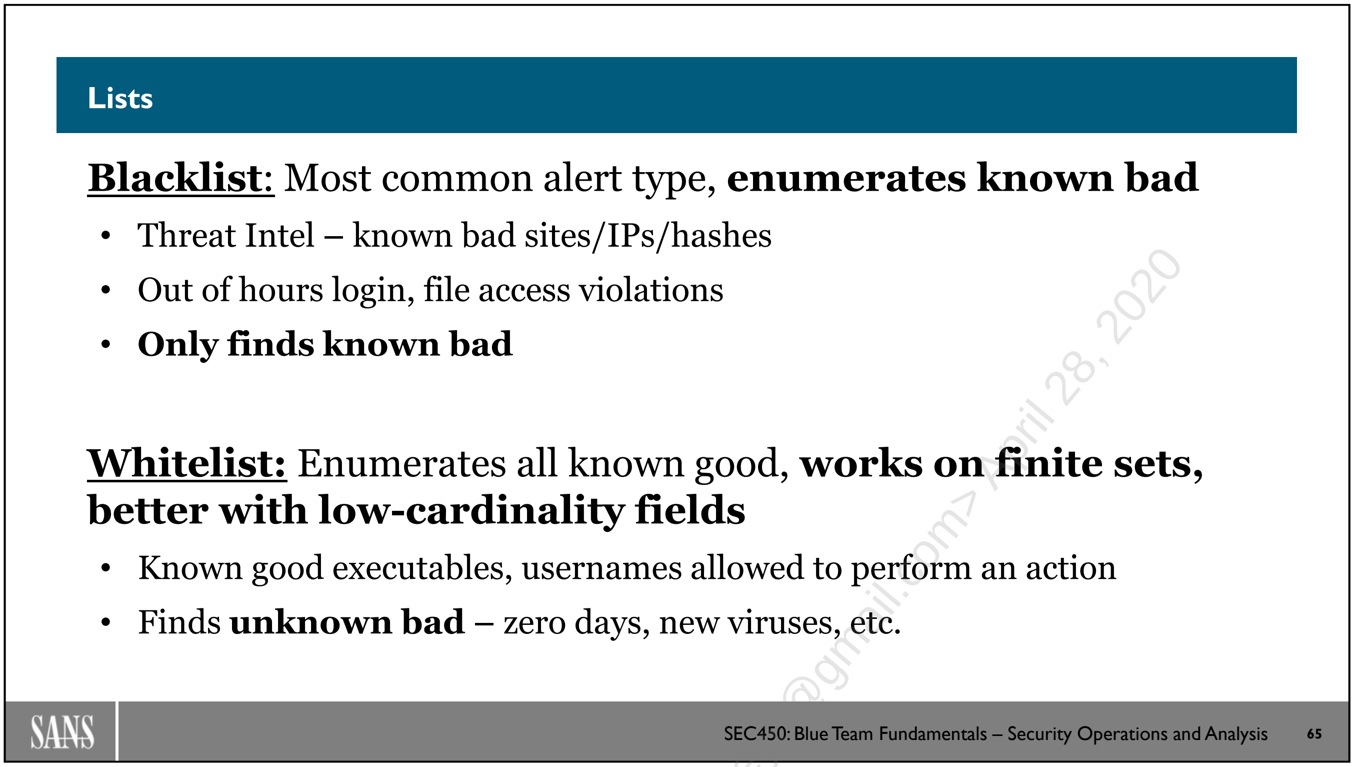
The final piece – rule logic

Common SIEM analytic trigger conditions:

- **Search Hit**: Any set of search conditions
- **List-based**: Blacklist / Whitelist
- **Thresholds**: Spike, Flatline, Frequency
- **Field Values**:
  - New term, changed term, changed count of terms
- **Sequence of Events**: event X followed by event Y

Rule Logic

We've discussed the requirement for parsing and enrichment as well as the pitfalls and statistical issues we must deal with in writing a high-fidelity alert. The final piece of the analytic writing puzzle is the higher-level categories that can be used to apply the matching or patterns we have created. Over the next few pages, we'll review these common ways of implementing analytics based on exact values or meta-feature of how the logs have been ingested (spike, x events in y time, or a sequence).



Lists

Blacklist: Most common alert type, **enumerates known bad**

- Threat Intel – known bad sites/IPs/hashes
- Out of hours login, file access violations
- **Only finds known bad**

Whitelist: Enumerates all known good, **works on finite sets, better with low-cardinality fields**

- Known good executables, usernames allowed to perform an action
- Finds **unknown bad** – zero days, new viruses, etc.

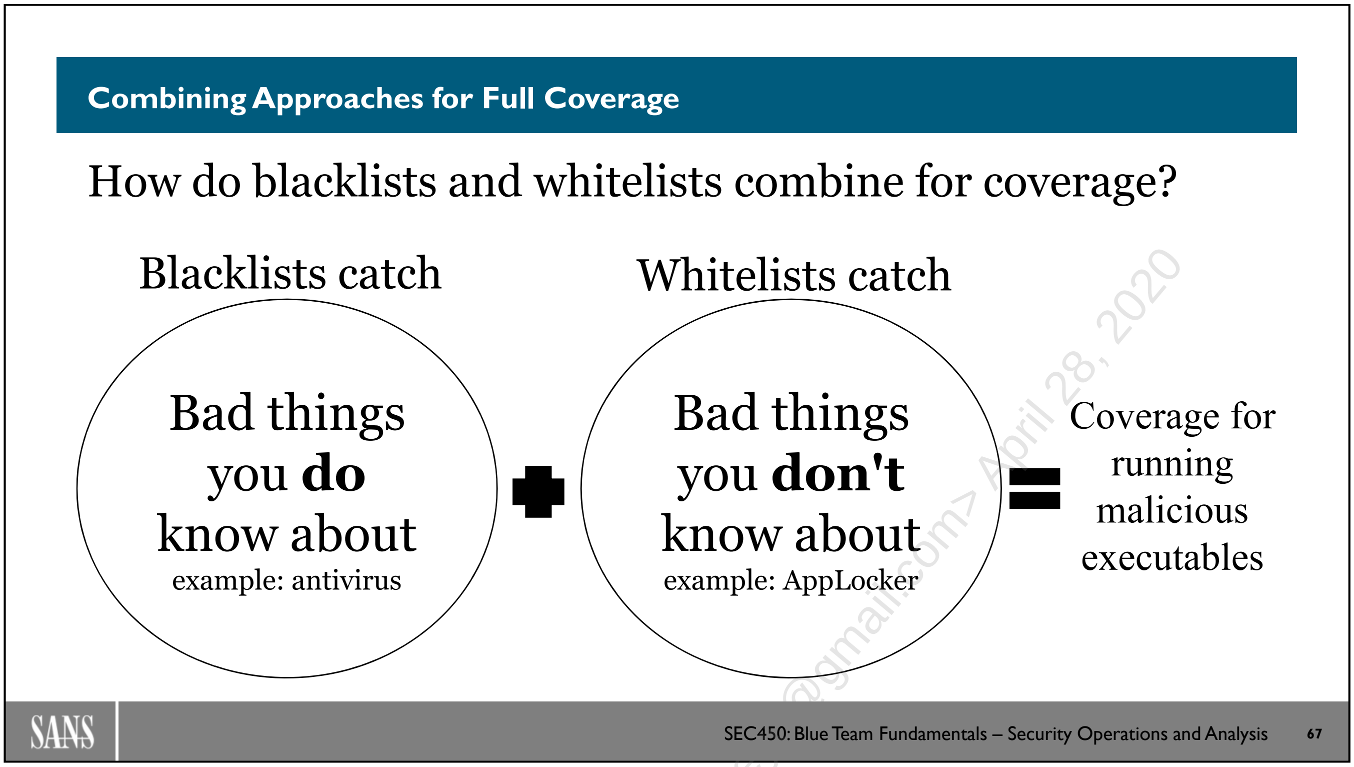
SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 65

Lists

One of the most basic types of analytic rules are those based on lists. An incredible amount of analytics in information security are based on this concept alone. Blacklist rules are simply a list of things we never wish to see, and therefore the selected fields in each log are checked against the list, and the alert fires if a match is found. This type of rule is used for matching against threat intel with program hashes checked against known bad programs, IP addresses, and domains check against known bad infrastructure and the like. It is, in general, anytime we are attempting to enumerate all known bad. As a policy, of course, this is impossible to do fully since what is bad changes constantly, but that doesn't mean it isn't worth doing as best possible. Clearly, antivirus, threat intelligence and plenty of other tools work on this concept and are quite successful. The downside of the blacklist is that it does not catch attacks from places we do not yet know are malicious, which is likely to include the most dangerous of operators. For this, use case rules based on a whitelist are better suited.

Whitelists are simply the opposite of the blacklist. Every log that is produced must match one of the terms on a list and if the term is *not* present the alert fires. The idea here is that we are enumerating *all known good.* Enumerating known good is a much easier task than enumerating all possible bad in almost every situation—we know which programs we should be running, what websites we typically connect to, and which hosts should talk to which other hosts. If we do not and cannot get a grasp on these traffic flows and interactions, we are boxing ourselves out of one of the most powerful analytic rules types available. This is why things like "Inventory of known hardware and software" consistently top the Center for Internet Security Top 20 list.[1] As long as we can keep tabs on what we expect to happen, even unknown threats will have trouble hiding. Whitelisting rules can be applied to files, paths, usernames, hostnames, and almost any other field in some capacity. The key piece is having the info to encode what is "normal" in the analytic. Should a domain administrator account be logging into random laptops all over the environment? No. Therefore, we can make a whitelist rule to alert us if the account is used anywhere but the pre-prescribed locations it should log in. Rules like this make it *much* more difficult for attackers to gain a foothold without alerting the SOC to their presence. Whitelist is the easiest to manage with low "cardinality" fields—fields that have fewer numbers for valid options. For example, whitelisting PC names would be manageable because in an organization of 100 PCs, the cardinality of the field is 100, which is feasible to keep updated. Whitelisting URLs on the internet, on the other hand, would be nearly impossible because the cardinality of a URL field is infinite.

[1] https://www.cisecurity.org/controls/



Combining Approaches for Full Coverage

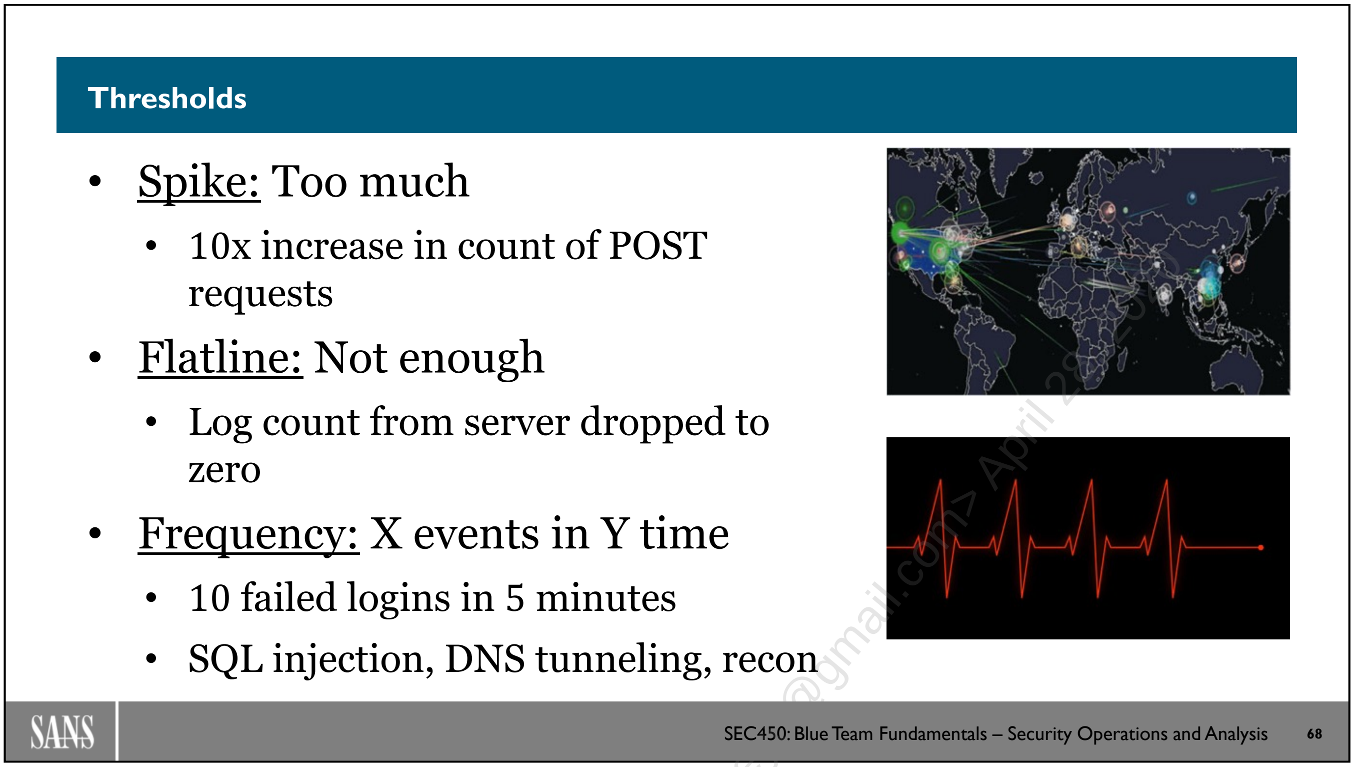
How do blacklists and whitelists combine for coverage?

Blacklists catch
Bad things you **do** know about
example: antivirus
Whitelists catch
Bad things you **don't** know about
example: AppLocker
Coverage for running malicious executables
SANS
SEC450: Blue Team Fundamentals – Security Operations and Analysis
67

Combining Approaches for Full Coverage

When should you use a blacklist-based approach as compared to a whitelist-based approach? The truth is you will probably need to use both to cover yourself from multiple kinds of threats. While you will make heavy use of signatures based on known malicious hashes, domain names, and other definitive IOCs, what about the malware no one has identified before? To catch the rest, whitelisting approaches can provide coverage. How does this look in reality? Traditional antivirus is a blacklist-based approach, comparing a set of known malicious files from a database to all files that are open and programs that are run. This provides good coverage for things we *already know* are bad. To supplement and further lock down the environment, many organizations employ application whitelisting on top of antivirus. This ensures that *all* programs that haven't been predetermined to be good will not run, covering the case of unknown malware not yet recognized by the antivirus engine. In a SOC, AV alerting is a near-certain indicator that something malicious (almost) happened. An application whitelisting hit isn't as sure to be an attack (it might be something you should've added to the whitelist but missed), but will almost certainly identify all executable-based viruses, whether anyone has ever heard of them before or not.

What if you want to take a whitelist approach but fear the noise it may cause? The "new term" type of rule, sometimes called "first contact" or the "dynamic whitelist" approach discussed in an upcoming slide, may provide a middle-ground option between blacklisting and whitelist.



Thresholds

- Spike: Too much
  - 10x increase in count of POST requests
- Flatline: Not enough
  - Log count from server dropped to zero
- Frequency: X events in Y time
  - 10 failed logins in 5 minutes
  - SQL injection, DNS tunneling, recon

SANS | SEC450: Blue Team Fundamentals – Security Operations and Analysis 68

Thresholds

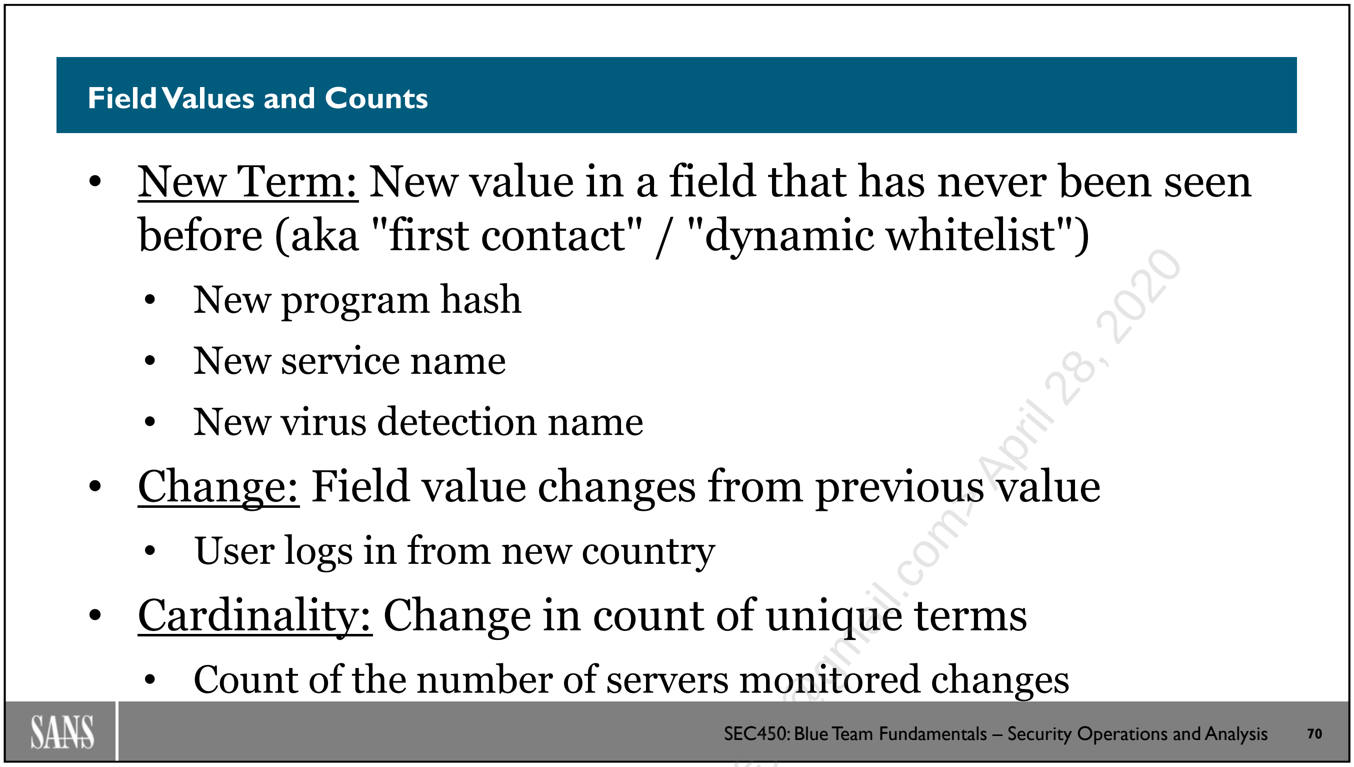
Threshold-based alerts are another common type of analytic. These alerts are slightly more advanced than the list-based options and have multiple ways of implementation. The first is the spike rule. A spike rule looks for an unexpected jump in activity that in theory could indicate a condition of interest. The spike could be based off multiple different reference points:

- An absolute value (ex: higher than 10GB traffic bandwidth)
- A multiplier relative to what it was a moment ago (ex: 5x in the last five minutes vs. the 5 minutes before that)
- A multiplier relative to the norm at this time of the day or this day of the week (10x jump vs. this 10 minute time slot last week/yesterday).

The flatline rule is just the opposite, looking for the value to either drop by some multiplier or go to zero vs. one of these reference points.

The tricky part of these types of rules is defining the thresholds at which they should alert. Making an absolute value-based rule ("alert if my outbound bandwidth goes above 1GB") doesn't account for the natural swing of many types of data. Bandwidth will go down during the night time and weekends, so if the idea is to catch exfiltration with a spike-based rule, will it fire if the exfiltration occurs over the weekend with a static bandwidth limit? Not if the attackers stay below the set limit, which is easy if no one is using the network. For this reason, triggering these types of rules relative to values in the past can be useful—a spike in bandwidth on Saturday this week compared to last week would successfully point out exfiltration. The problem here is that it also would potentially point out updates or patches occurring. The third type of reference point, triggering on sudden spikes compared to the last x amount of time, may be a useful option as well, only triggering when there is a drastic spike in activity, regardless of the absolute value, or what has occurred in the past. Machine learning algorithms can help keep track of the "norm" over time and apply those types of reference points, while absolute values or multiplier type rules can be researched with historical data and set by analysts.

Frequency rules are a bit different; their threshold is based on the count of activities within a certain window. These rules are straightforward to set up and are excellent at finding brute force attack types where something is attempted at high volume from a single source or generates too many of the same event. The obvious use for this type of rule is for someone attempting to brute force a password but uses go far beyond that. DNS tunneling (too many requests to the same domain in 10 minutes), SQL injection (too many requests to a single server in X amount of time), and plenty of other attacks can utilize this rule type as well.



Field Values and Counts

- New Term: New value in a field that has never been seen before (aka "first contact" / "dynamic whitelist")
  - New program hash
  - New service name
  - New virus detection name
- Change: Field value changes from previous value
  - User logs in from new country
- Cardinality: Change in count of unique terms
  - Count of the number of servers monitored changes

SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 70

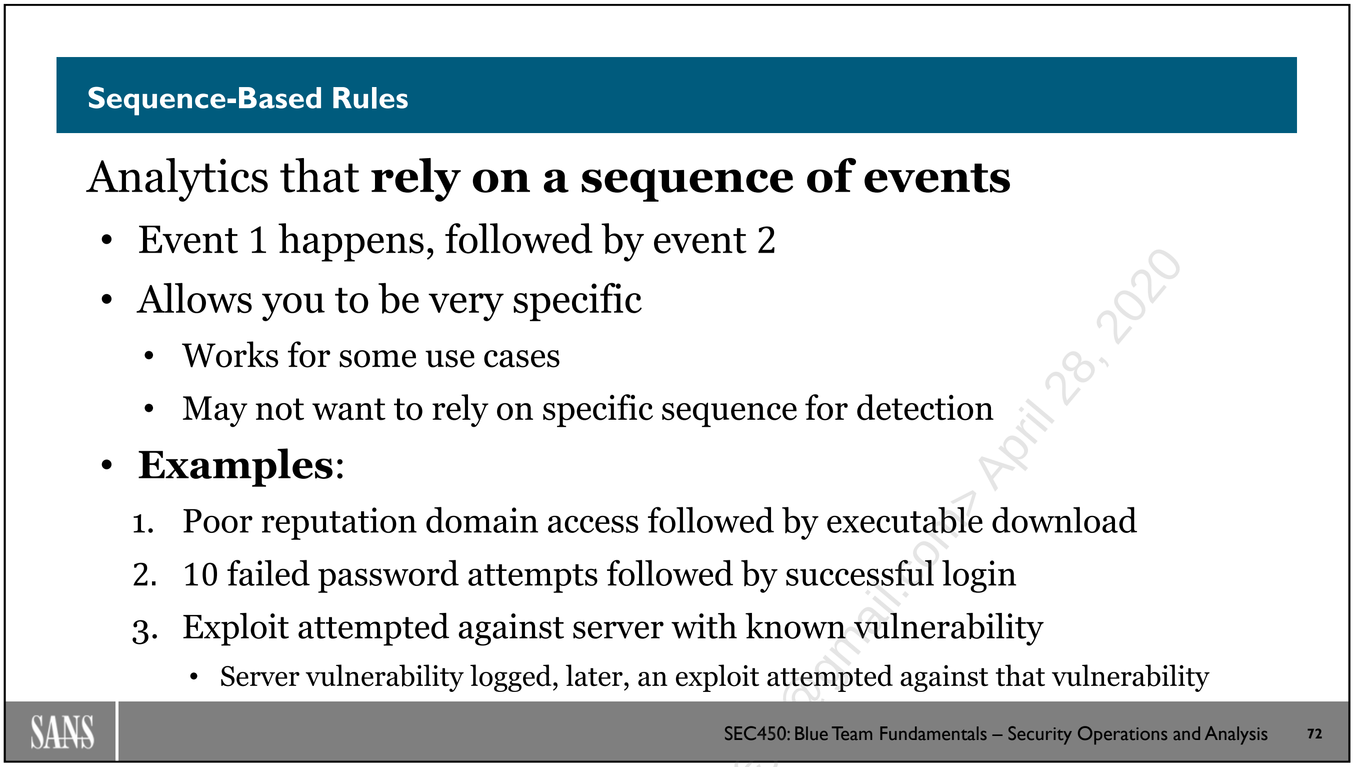
Field Values and Counts

Field value-based alerts are yet another type of alert logic, of which some could fall under the other rule types but are a bit more specific in how they work. Common rules of this type include looking for new terms in a pre-defined set (similar to whitelist, but not quite the same), the change of a specific field, or the change in the count of a field.

New term rules (or first contact / dynamic whitelist rules as they are sometimes called) are one of the best rule types and should be used much more frequently in the author's opinion. These rules have the benefits of a whitelist without the management overhead of keeping it updated. A new term or first contact rule will sample all the values seen in a specific field over the past X amount of time, and alert if any *new* items show up from then on. In effect, it is a whitelist, but instead of pre-determining what is good, you assume that the past is good and that the set shouldn't be changing. This means any time you see a new hash, new virus name, new host, new autorun item, or any other value you want to trigger on in the network you will still immediately know about it, but you don't have to go through the pain of finding all the valid ones first. Once we see the new term, the alert fires and that value is added to the list so that it will not cause any more alarms, which can be another nice feature for reducing noise since that initial tipoff is likely enough to get you on the case of a suspicious item.

Change rules are also a bit like a whitelist except the logic says that a value for a specific field should never change from what it is at the current moment. Change rules are used for situations where an account is *always* going to be performing some specific task, and you want to know if it differs, but do not want to manage re-writing the rule when it does change. This alert could be done with a whitelist as well, but same as with the "new term" rule. In the "change" rule, once the item has changed, the new value will be the "normal" value, and therefore it will only alert once. It is another form of a self-managing whitelist. For example, if you have an employee that rarely travels, you could use this rule to say any time the domain admin login happens from anywhere other than their home office location, then alert. If they went on a business trip, you would then get the alert that something different had happened but wouldn't continue to get alerts while they were away, only the single time as the value in the log changed.

Cardinality is a rule type that means "how many unique values exist for this field." The cardinality on the username field for a company with 10 employees would be 10, for example. If you find a situation where you do not care about the specific values of a set but would rather want to know if the *size* of the set grows, this is the alert logic type for you. Perhaps you have a specific part of your network where you know five systems will always be online, but which five systems will change, and therefore you cannot use the other types of rules. This rule could alert you if one of the systems goes offline or if a sixth is added by accident.



Sequence-Based Rules

Analytics that **rely on a sequence of events**

- Event 1 happens, followed by event 2
- Allows you to be very specific
  - Works for some use cases
  - May not want to rely on specific sequence for detection
- **Examples**:
  1. Poor reputation domain access followed by executable download
  2. 10 failed password attempts followed by successful login
  3. Exploit attempted against server with known vulnerability
     - Server vulnerability logged, later, an exploit attempted against that vulnerability

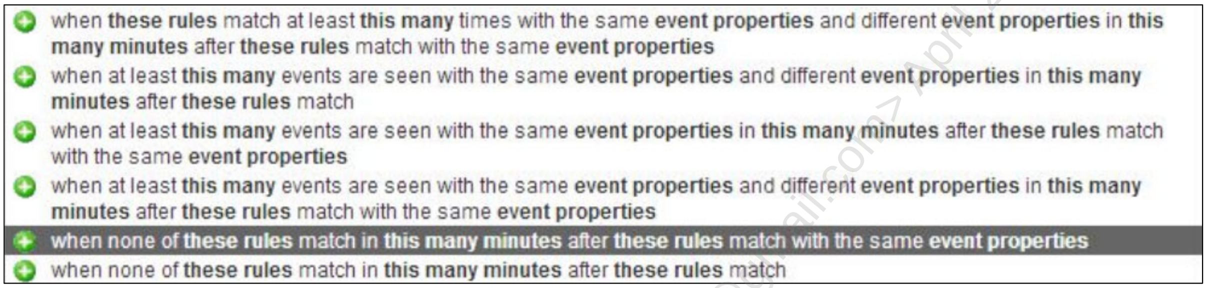
Sequence-Based Rules

Another way of creating analytic is by basing it on a sequence of events. Sometimes, these are referred to specifically as "correlation rules," and in other sources/products, the word correlation is used in a broader sense. So, for this slide, we're specifically talking about a sequence of events. Defining a sequence of events of interest allows us to write detections for things like attempted brute force attacks, exploit kit deliveries followed by executable downloads, and detecting an exploit attempt against a server that is specifically vulnerable to that exploit. They can be extremely useful, and the only way to detect certain attacks in some cases but should also be approached with caution due to their specificity.

Over-Specification

Using a *too* tightly defined sequence may lead to false negatives:

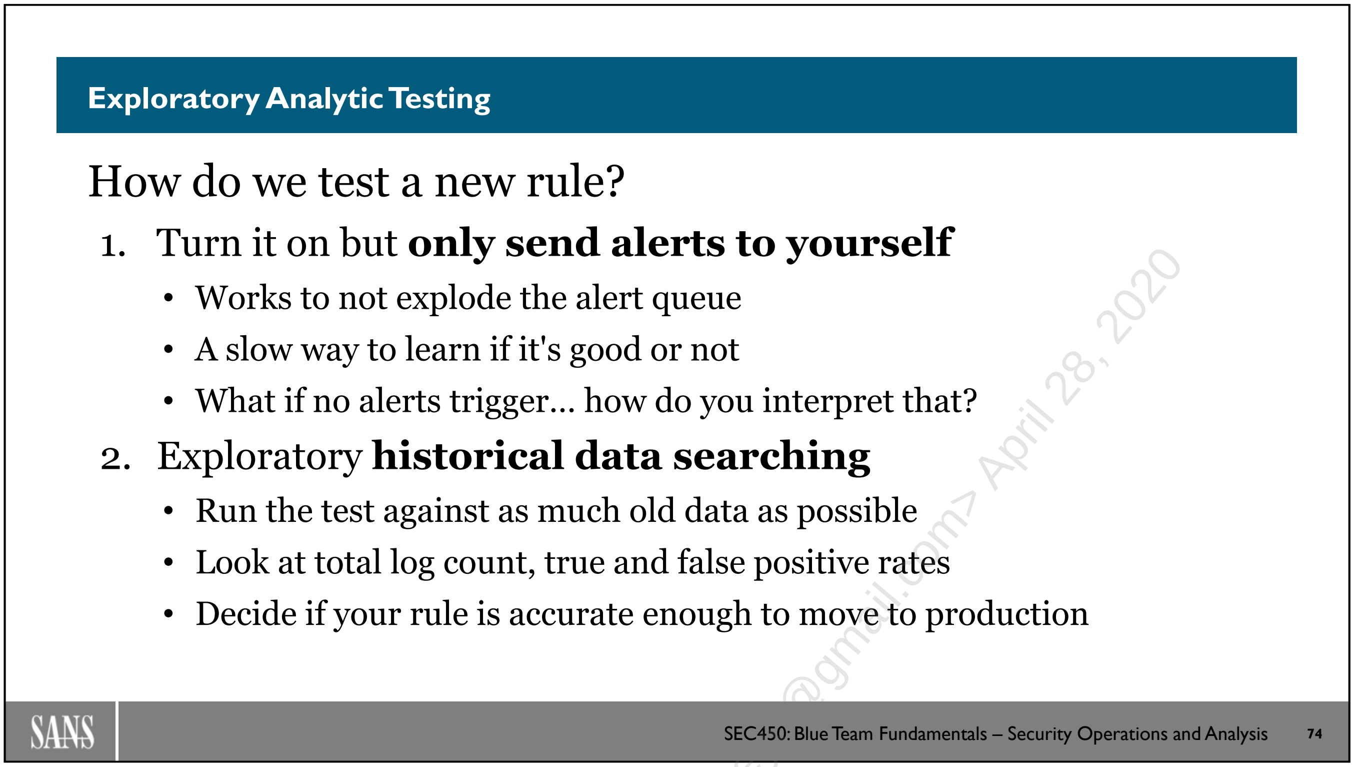
- Might have a very unique condition to catch
- If so, these types of rules can work
- Remember – simplicity is the name of the game



- when **these rules** match at least **this many** times with the same **event properties** and different **event properties** in **this many minutes** after **these rules** match with the same **event properties**
- when at least **this many** events are seen with the same **event properties** and different **event properties** in **this many minutes** after **these rules** match
- when at least **this many** events are seen with the same **event properties** in **this many minutes** after **these rules** match with the same **event properties**
- when at least **this many** events are seen with the same **event properties** and different **event properties** in **this many minutes** after **these rules** match with the same **event properties**
- when none of **these rules** match in **this many minutes** after **these rules** match with the same **event properties**
- when none of **these rules** match in **this many minutes** after **these rules** match

Over-Specification

Remember, with a sequence of events-based rules, that you can easily tread into "too specific", leaving out more general patterns that could allow you to identify slight variations on a previously seen attack. While rules with logic like the ones shown above (a screenshot showing how specific the rules in the QRadar SIEM rule wizard can get) can obviously be done in some systems, carefully consider whether you *need* to go to that level when a simple search, blacklist, or frequency rule might be able to accomplish the same thing.



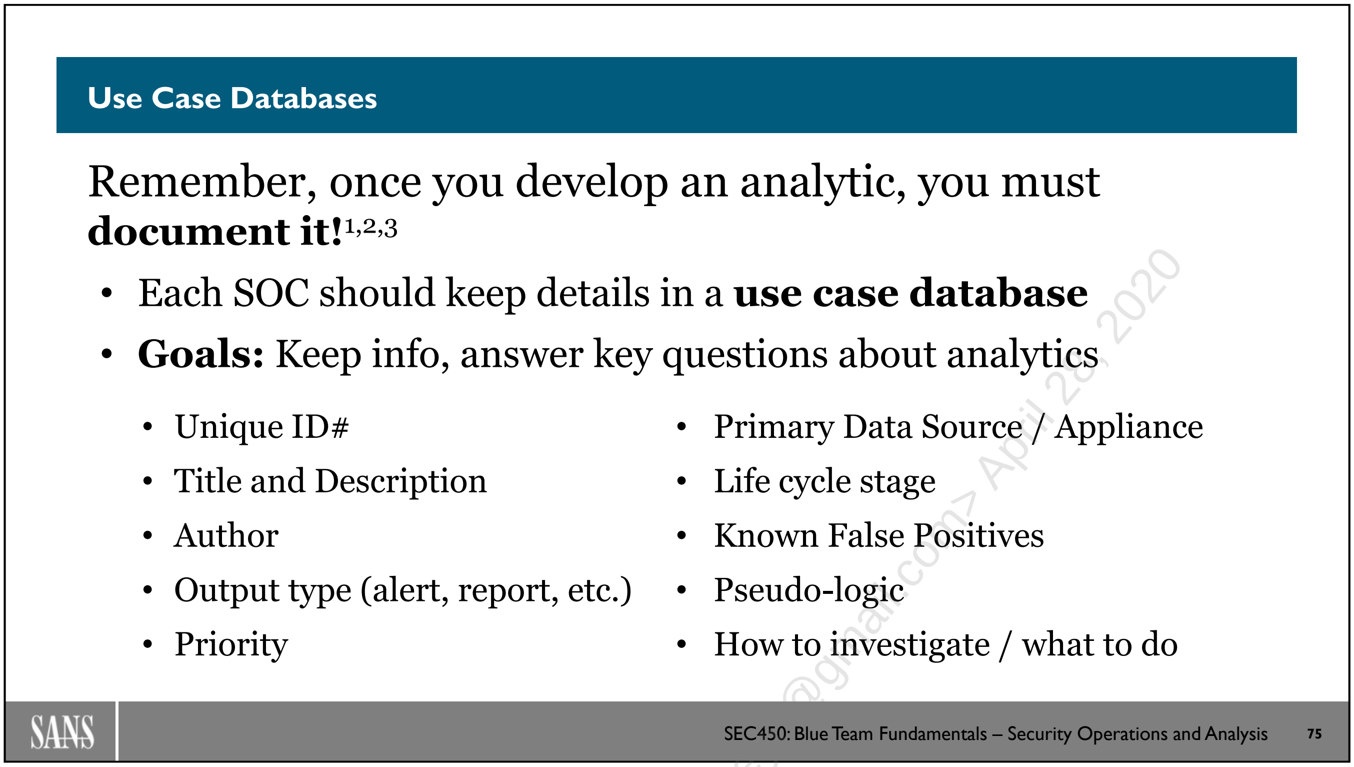
Exploratory Analytic Testing

How do we test a new rule?

1. Turn it on but **only send alerts to yourself**
   - Works to not explode the alert queue
   - A slow way to learn if it's good or not
   - What if no alerts trigger… how do you interpret that?
2. Exploratory **historical data searching**
   - Run the test against as much old data as possible
   - Look at total log count, true and false positive rates
   - Decide if your rule is accurate enough to move to production

Exploratory Analytic Testing

Once you have developed the logical conditions you plan to use with your rule, next you need to test it. While just turning it on without alerting sent to the main queue can be one way of soft-introducing the new analytic, a complimentary step is to test it retroactively against data you have already ingested. Using the past month or two of logs, run the logic you have created against it and see how many true and false positives you produce. If you've collected 10M proxy logs, for example, and your rule identifies 10 true positives and zero false positives, you have good reason to believe the accuracy of the rule will be high. If on the other hand, you find 50% true and false positives identified, you can look at the fields that are in common/different between the good and bad hits and modify your rule to accommodate. Running analytics against historical data should be considered a necessary step before deploying any new detection analytic.



Use Case Databases

Remember, once you develop an analytic, you must **document it!**[1,2,3]

- Each SOC should keep details in a **use case database**
- **Goals:** Keep info, answer key questions about analytics
  - Unique ID#
  - Title and Description
  - Author
  - Output type (alert, report, etc.)
  - Priority
  - Primary Data Source / Appliance
  - Life cycle stage
  - Known False Positives
  - Pseudo-logic
  - How to investigate / what to do

Use Case Databases

Now that you have designed an awesome, high-fidelity analytic, don't forget to record all the details about it. Every SOC should have a use case database of sorts that ensures that even if the author of an analytic leaves or is not present, everyone will have a reference for what the rule is about, and what to do if it fires. These can be as simple or as complicated as desired, but ultimately the information recorded should give a thorough introduction and explanation to what the rule is doing, why it was necessary, and any other relevant info about what to do when you see it alert. Options include everything from a big Excel chart, to wikis, to ticket systems like Jira or Redmine with nested parent/child relationships between analytics, and the tactics and techniques they attempt to discover. As mentioned in Day 1, for more outstanding information on how to make a thorough, well-documented use case, see the Blue Team Handbook volume 2 by Don Murdoch[1] or his 2017 Security Onion talk "Building Your Sec Ops Use Case".[2] Another option would be to use the ADS system developed by Palantir.[3]

[1] https://www.amazon.com/Blue-Team-Handbook-condensed-Responder/dp/1500734756

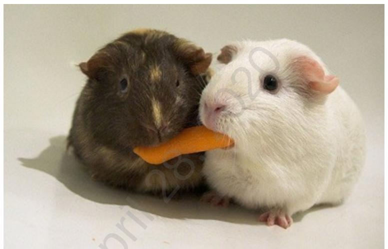
[2] https://www.youtube.com/watch?v=4ESQ0GfPHYY

[3] https://medium.com/palantir/alerting-and-detection-strategy-framework-52dc33722df2

Analytic Sharing

Have you ever received an analytic from another organization?

- Probably not… at least not a specific one

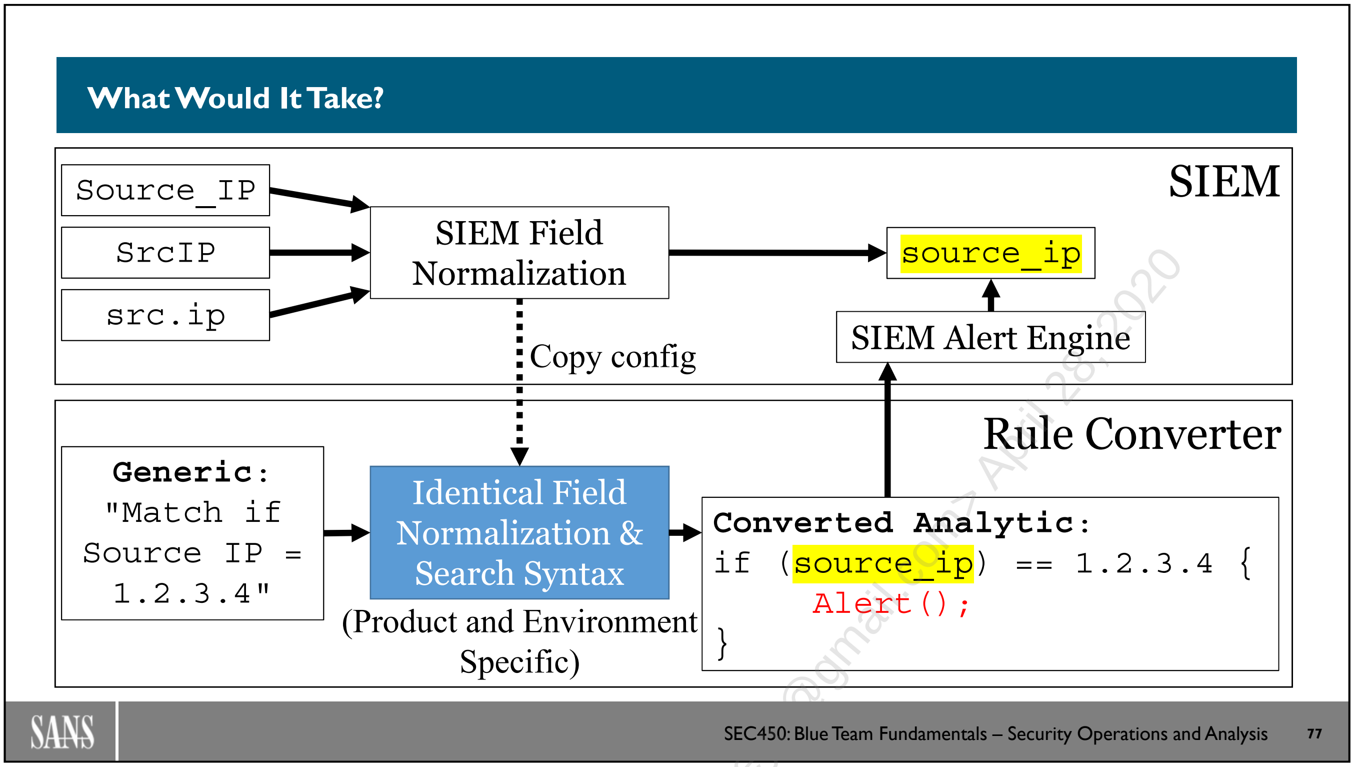


Why is it so difficult?

1. Even in deployments of same SIEM…
   - **Field names** may differ
   - **Data sources** differ
2. We collect in different log **formats**:
   - Windows logs – Syslog, JSON, XML
3. We've had **no common language** to specify analytics

Analytic Sharing

Now that you've developed an analytic that identifies a strain of specific malware or other malicious condition, wouldn't it be great if you could share it with the world? It sure would. Unfortunately, it's not that simple. For things like network traffic and files, this sort of sharing is very easy—passing out a Snort signature, or YARA signature is a common way of writing an analytic once in a way the whole world can use it, but there is no equivalent for events from log files. Why? Because even in the ideal case of two companies that have identical SIEMs, the field names and data sources may differ enough that trying to apply the same logic is likely to fail (for the SIEMs with a common information model that is strictly enforced, this may not be true). Within the data sources collected, what log agent you use and which format you send the data can also modify which field parsing and normalization. Then expand this to a world filled with different SIEMs and the distribution of analytics becomes even more difficult. Many SIEMs make rule export a pain, and of course, exporting from one SIEM and importing into another vendor system is never going to work. SIEM vendors have no motivation to make this work; it would make product switching too easy, and they want to lock you in.



What Would It Take?

Generic: "Match if Source IP = 1.2.3.4"

Converted Analytic:

```
if (source_ip) == 1.2.3.4 {
        Alert();
}
```

What Would It Take?

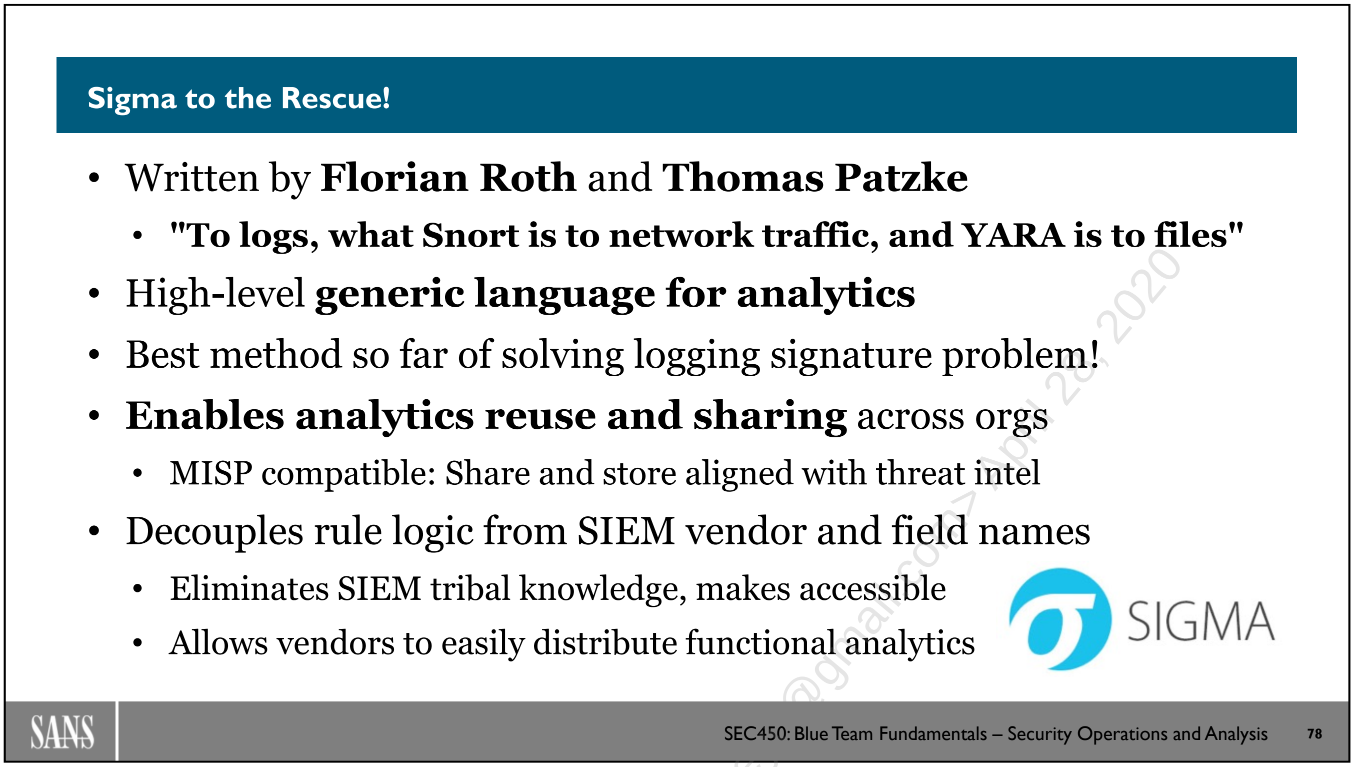
To accomplish this feat, consider what would need to happen. Across the top row in the above diagram, we see all the disparate devices in our network sending in logs with different field names. These get pushed through name normalization at the SIEM and may turn the field into a name called source_ip, which is all great. Once our analytic from someone else came along, though, we do not have any external system that can apply the same field normalization to the fields in the analytic, which means there's no chance of it ever matching, it's like our poorly parsed log problem but failing from the other side instead. If you write an analytic in company one looking for src_ip = 1.2.3.4 and apply it in an environment where that field is called "source_ip," it will never find anything.

To make this work, we would need the following steps to occur:

- Write analytics in some sort of generic format
- Apply the identical field normalization to that analytic that is used by our SIEM
- A conversion of the analytic search query to the syntax used by our specific SIEM

These processes are represented using the solid box in the diagram. If a process like this existed, we could write a generic rule, push it through that conversion process and spit out an analytic converted to the syntax and field names required in our specific SIEM implementation. That rule could be put into the SIEM alert engine using the correct field names and data sources as required.

The bottom line here is that up until this point, there has been no clear solution to this problem, and holding our breath waiting for vendors to create a solution is unlikely to work. Therefore, like so many other problems in information security, the community has come up with a solution. Enter Sigma…



Sigma to the Rescue!

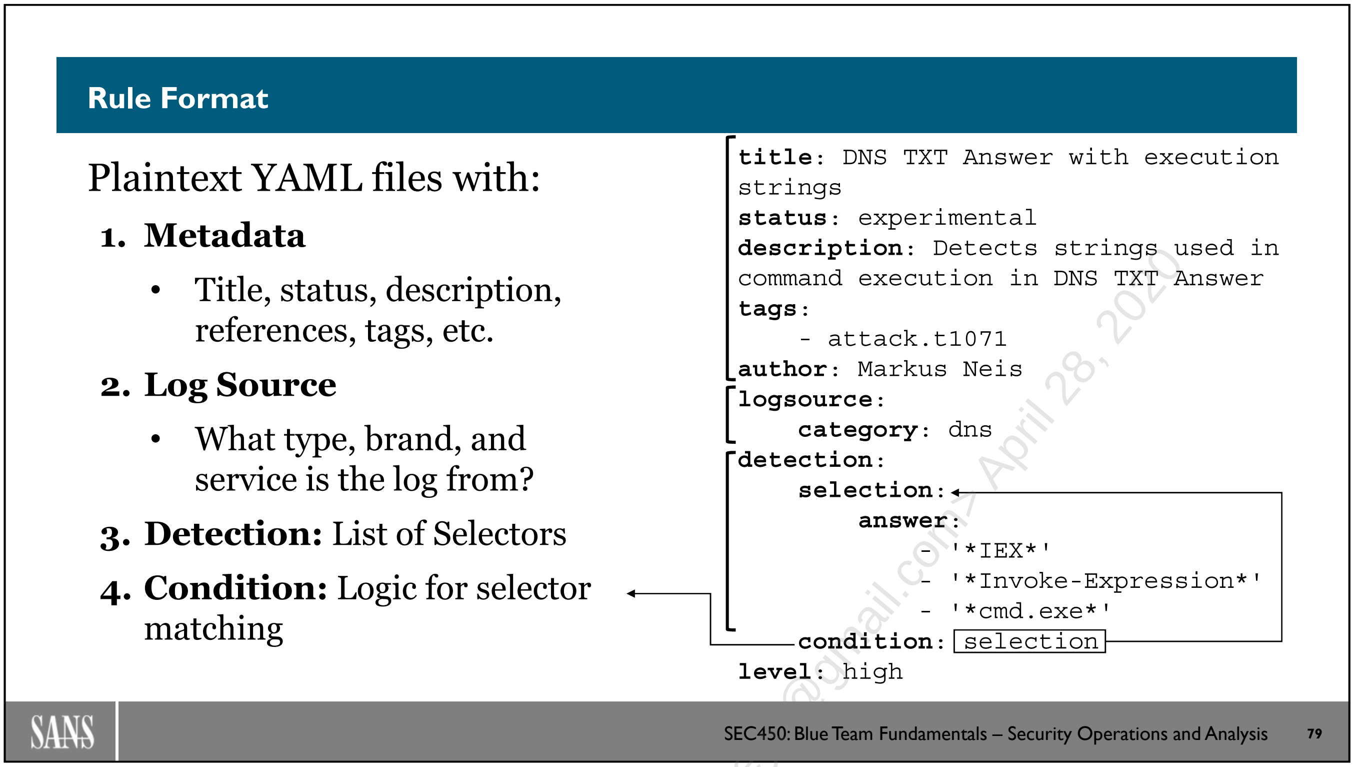
- Written by **Florian Roth** and **Thomas Patzke**
  - **"To logs, what Snort is to network traffic, and YARA is to files"**
- High-level **generic language for analytics**
- Best method so far of solving logging signature problem!
- **Enables analytics reuse and sharing** across orgs
  - MISP compatible: Share and store aligned with threat intel
- Decouples rule logic from SIEM vendor and field names
  - Eliminates SIEM tribal knowledge, makes accessible
  - Allows vendors to easily distribute functional analytics

Sigma to the Rescue!

Sigma, an open-source project on GitHub developed by researchers Florian Roth and Thomas Patzke, is emerging as the solution we've been looking for.[1] Sigma, in their own words, is "To logs, what Snort is to network traffic, and YARA is to files." Its goal is to be the lost method of writing an analytic that we can share with everyone in the same way we could push out a Snort and YARA signature. It is a high-level, generic language for analytics that will convert them to function with specific SIEMs and other software such as PowerShell and grep.

The community is excited about Sigma because if it were picked up as a standard, there would be a long list of outstanding side effects beyond purely analytic sharing. One of the most important is that it could help mitigate the effects of SIEM "tribal knowledge." By this, we mean the requirement to understand SIEM data sources and field names deeply before you can write a useful analytic. With Sigma, as long as one person in each SOC understood field names and the data ingest pipeline, that one person could configure Sigma for analytic conversion in that environment and all other analysts would only need to develop analytics in the high-level language while the Sigma converter took care of the implementation! While this is great at the individual organization level, it is also true across organizations. An analyst switching jobs from one company to another, both using Sigma, could walk in the door day 1 in the new environment and immediately be able to write analytics since the specifics are taken care of by the converter! It would also relieve the pain associated with extracting generic rules out of APT reports and having to figure out how to write them with our specific tools. In a future with Sigma, vendors could directly release the Sigma format generic rules and each organization could decide to convert and use them or not, no hassle required!

[1] https://github.com/Neo23x0/sigma



Rule Format

Plaintext YAML files with:

1. **Metadata**
   - Title, status, description, references, tags, etc.
2. **Log Source**
   - What type, brand, and service is the log from?
3. **Detection:** List of Selectors
4. **Condition:** Logic for selector matching

```
title: DNS TXT Answer with execution
strings
status: experimental
description: Detects strings used in
command execution in DNS TXT Answer
tags:
    - attack.t1071
author: Markus Neis
logsource:
    category: dns
detection:
    selection:
        answer:
            - '*IEX*'
            - '*Invoke-Expression*'
            - '*cmd.exe*'
    condition: selection
level: high
```

Rule Format

Whether or not we choose to use Sigma in our environment, it is a well-implemented system and studying the way it is organized will give us further clarity on how we should organize our use cases and analytics.

At the highest level, Sigma rules are broken down into four sections written in the YAML format—a text-based (which means Git repo controllable!), human-readable format that is simple to understand and modify as needed. As shown on the slide, the first section is the metadata about the rule. This section contains the title, status (active, disabled, experimental, author, etc.), description, tags, and more. This section is important because it helps use apply MITRE ATT&CK framework technique tags, or any other context we'd like, which makes sorting and counting our analytics based on kill-chain stages or anything else later on very easy. Can you answer the question "how many analytics do we have that trigger on email delivery?" With Sigma rule tags, producing that answer would be trivial.

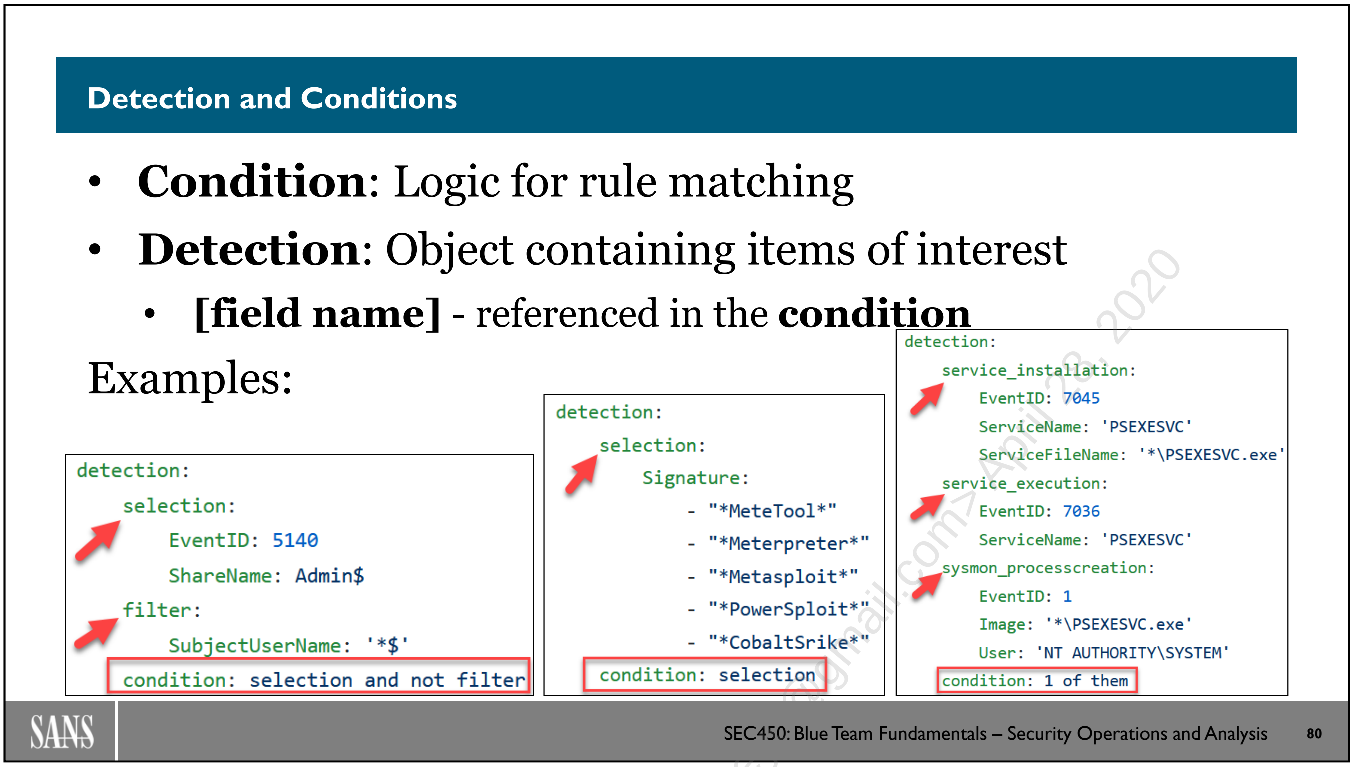
The second section is the log source information, and it describes at a high and low level what sources of data you are applying the rule to. Is it for Linux/Windows, a firewall, which brands of firewalls, which services on that firewall? All these questions are answered in this section by breaking them down into a category, product, and service. This categorization not only allows tracking but will affect how the Sigma rule conversion is done as well and is how sigma compensates for different tools and log sources in each environment.

category: proxy, firewall, AV, IDS - For all logs of a **group of products**

product: Squid, pfSense, Symantec, Snort, Windows - For all log outputs of **one product**

service: SSH, DNS, DHCP - For a **subset of a products logs** – sshd, named, dhcpd, …

The final two sections are the Detection and Condition sections. We use these to specify which fields need to contain certain values and the logic to use for detection.



Detection and Conditions

- **Condition**: Logic for rule matching
- **Detection**: Object containing items of interest
  - **[field name]** - referenced in the **condition**

Examples:

```
detection:
    selection:
        EventID: 5140
        ShareName: Admin$
    filter:
        SubjectUserName: '*$'
    condition: selection and not filter
```

```
detection:
    selection:
        Signature:
            - "*MeteTool*"
            - "*Meterpreter*"
            - "*Metasploit*"
            - "*PowerSploit*"
            - "*CobaltSrike*"
    condition: selection
```

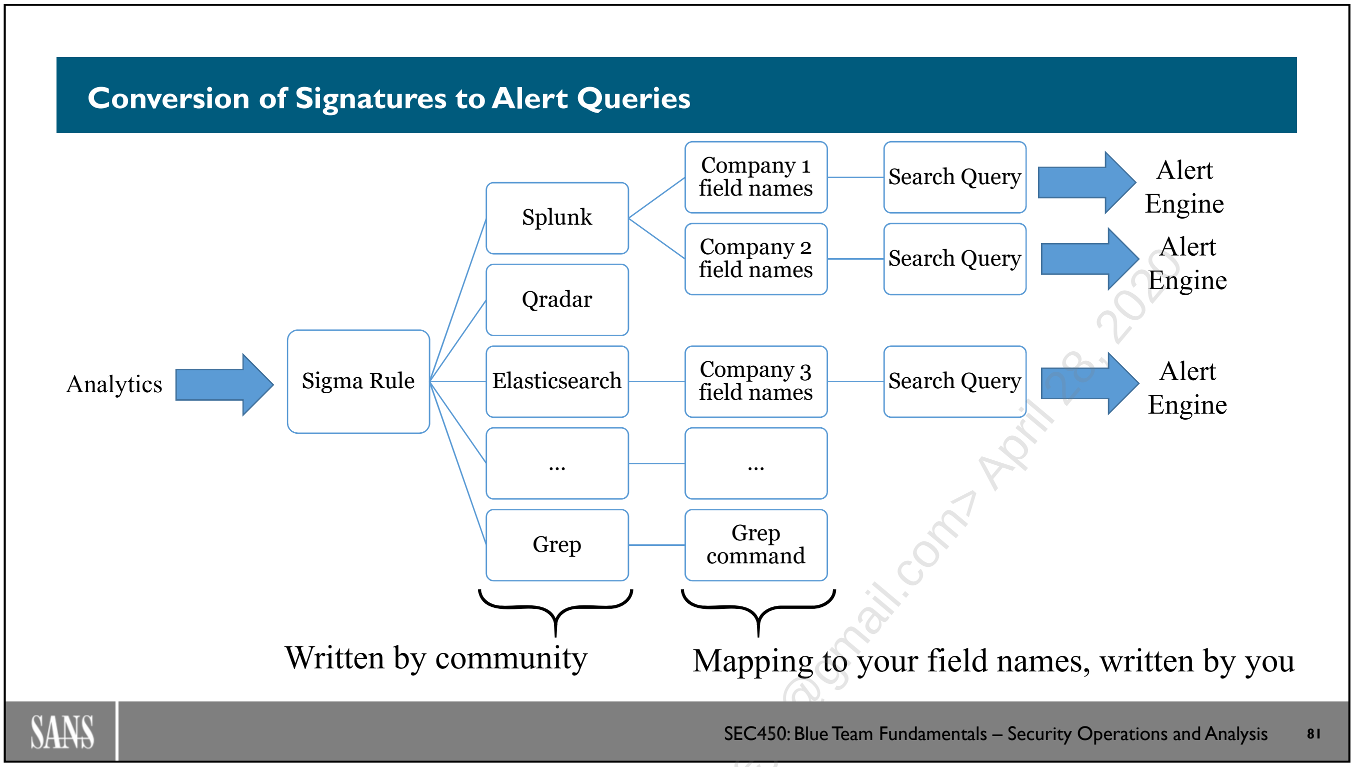
```
detection:
    service_installation:
        EventID: 7045
        ServiceName: 'PSEXESVC'
        ServiceFileName: '*\PSEXESVC.exe'
    service_execution:
        EventID: 7036
        ServiceName: 'PSEXESVC'
    sysmon_processcreation:
        EventID: 1
        Image: '*\PSEXESVC.exe'
        User: 'NT AUTHORITY\SYSTEM'
    condition: 1 of them
```

Detection and Conditions

Here are some additional examples of detections and conditional logic options. In the leftmost box, we are looking for "selection and not filter." The selection section specifies an EventID 5140, and a share name of Admin$, the filter is saying except when the username ends with a dollar sign. In effect, this rule would identify the mapping of any administrative shares, minus the computer accounts in active directory since those usernames end with a dollar sign.

The middle example is simply looking for any of the names of common hacking tool names inside the signature field for an antivirus scanner. It is using an OR condition so a match against any of the terms would be enough to match.

The rightmost example is looking for 1 of any of the detection conditions listed above. The first is a service installation event with EventID 7045 and a service name of PSEXECSVC with a service filename that matches. The next option looks for the execution of the same service using the name and a 7036 event ID (service entered the running state). The final is another option to catch the same thing but instrumented through a Sysmon Event ID 1, which records new process creation.



Conversion of Signatures to Alert Queries

Analytics

Sigma Rule

Splunk

Qradar

Elasticsearch

...

Grep

Company 1 field names

Company 2 field names

Company 3 field names

...

Grep command

Search Query

Search Query

Search Query

Alert Engine

Alert Engine

Alert Engine

Written by community

Mapping to your field names, written by you

Conversion of Signatures to Alert Queries

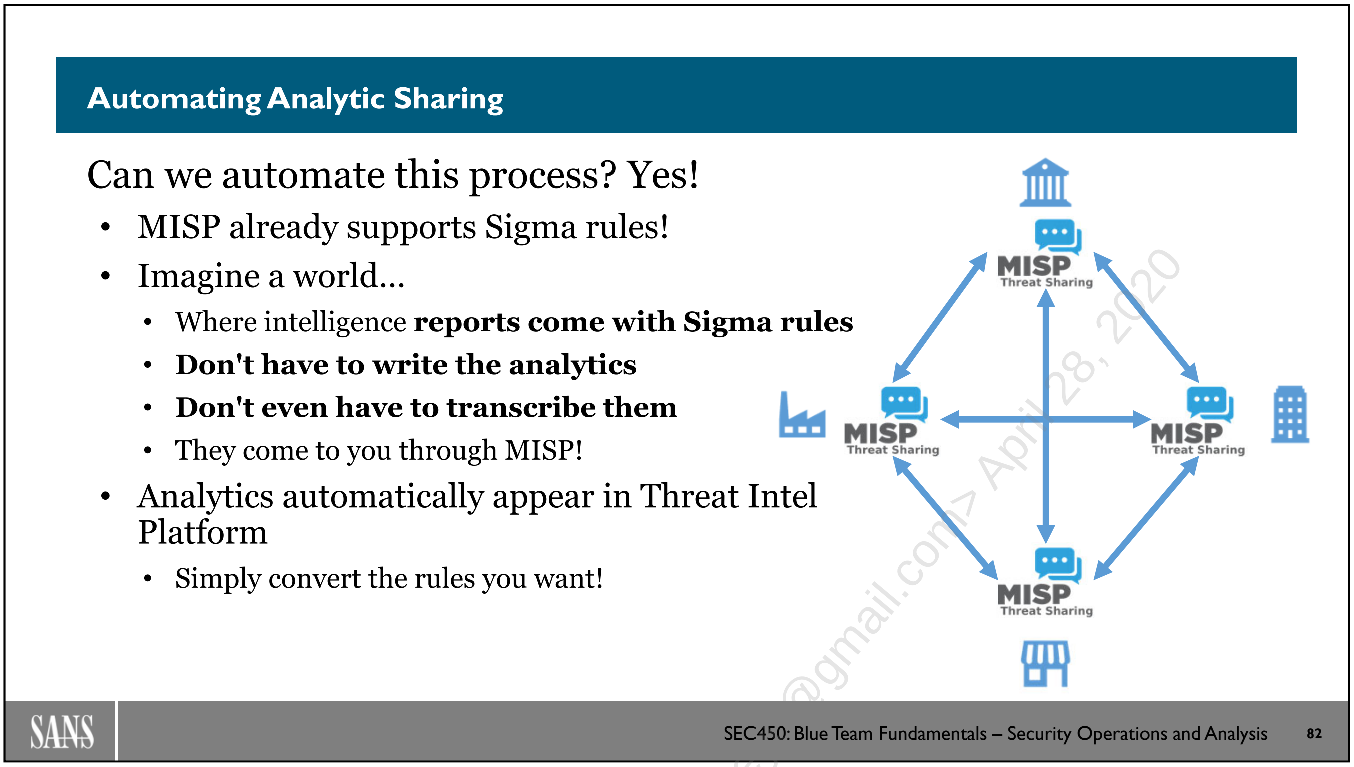
What happens when we go to transform a Sigma rule into an analytic? The first step is putting it into Sigma rule format as defined in the specification.[1] It is designed to be generic enough that it can be parsed and converted first by one of a set of plugins that can transform the fields and logic into a specific tool's required format, which is the second step of conversion. A plugin for Splunk or QRadar or Elasticsearch can be used to take the generic form and move the analytic into something that will be a valid input to these various tools. For many SIEMs as well as other tools such as PowerShell and grep, these plugins are already written by the community.

The next step is where you come in. Although the analytics are in a valid syntax for running a search, they will not yet have the correct field names. That means a third step with another conversion needs to occur. This third step will map the fields in the generic signature format to the names your organization uses in your particular deployment. This mapping is created inside an easy to format YAML file that is easy to modify as fields may be added or changed.[2]

Finally, after going through these two rounds, what comes out the other end is a full, ready-to-use search term that can be pasted directly into your SIEM search box. If everything worked as intended, the only step left is to tell the SIEM that you want to alert on any logs that match that search in the future and send out an alert with their contents.

[1] https://github.com/Neo23x0/sigma/wiki/Specification

[2] https://github.com/Neo23x0/sigma/wiki/Converter-Tool-Sigmac



Automating Analytic Sharing

Can we automate this process? Yes!

- MISP already supports Sigma rules!
- Imagine a world...
  - Where intelligence **reports come with Sigma rules**
  - **Don't have to write the analytics**
  - **Don't even have to transcribe them**
  - They come to you through MISP!
- Analytics automatically appear in Threat Intel Platform
  - Simply convert the rules you want!

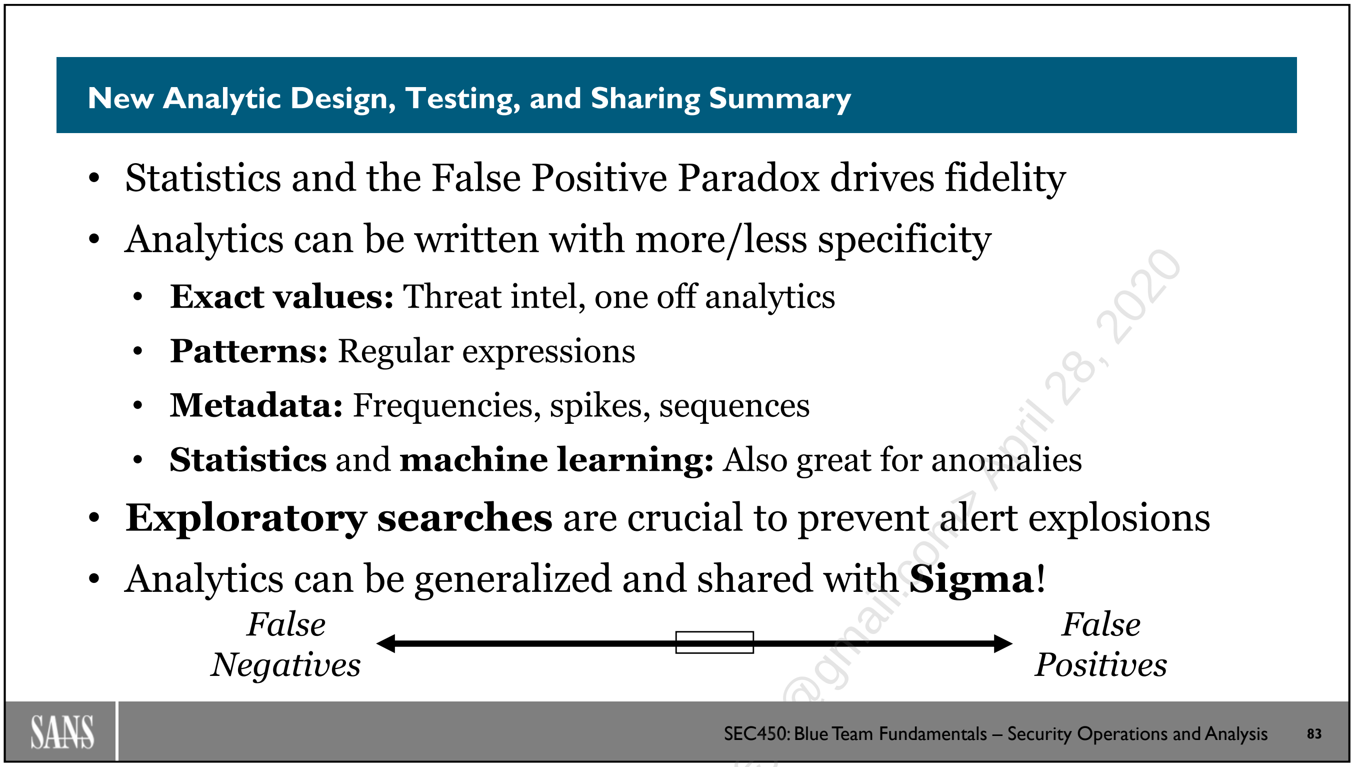
Automating Analytic Sharing

Things get even better when you bring other SOC tools into the picture. Not only does Sigma allow us to distribute rules generically through threat reports, GitHub, and other channels, we could also subscribe to, and automatically import rules through our Threat Intelligence Platform! Every threat intelligence platform can store and publish plaintext entries, and that is exactly what Sigma analytics are, which makes automated pushed distribution of analytic lists easy!

MISP has already jumped on this and put a specific "Sigma rule" object to enhance the compatibility of the project further![1] With the Sigma2MISP tool, you can push Sigma rules into any event in MISP. Once there, anyone who subscribes to the data you share will automatically receive a copy![2] On the receiving end, you only need to check through the new Sigma rules you receive and if you decide you want to implement them. You can pull them out with the MISP API and insert them through the Sigmac rule converter. The output will be an analytic that converts to the fields and syntax your SIEM uses, and can be implemented immediately!

[1] https://www.misp-project.org/tools/

[2] https://github.com/Neo23x0/sigma/blob/master/README.md#sigma2misp



New Analytic Design, Testing, and Sharing Summary

- Statistics and the False Positive Paradox drives fidelity
- Analytics can be written with more/less specificity
  - **Exact values:** Threat intel, one off analytics
  - **Patterns:** Regular expressions
  - **Metadata:** Frequencies, spikes, sequences
  - **Statistics** and **machine learning:** Also great for anomalies
- **Exploratory searches** are crucial to prevent alert explosions
- Analytics can be generalized and shared with **Sigma**!

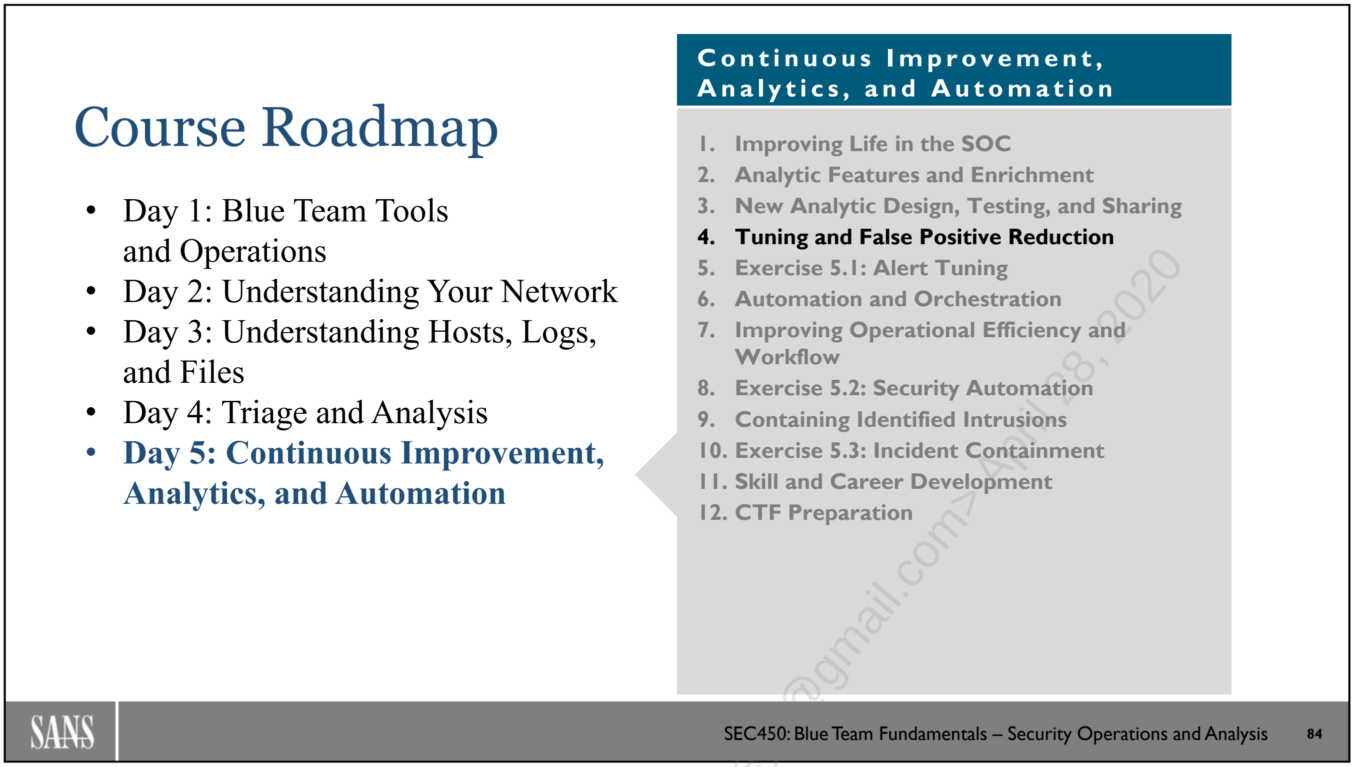
False Negatives ⟵ ▭ ⟶ *False Positives*

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 83

New Analytic Design, Testing, and Sharing Summary

In this section, we continued to cover some of the challenges and requirements of writing a strong analytic in order to avoid those false positives that will drag the team down. Hopefully, you now have a better appreciation for the factors and options that go into analytic design and tuning and what makes false positives so pervasive. Given that we operate in an environment that is friendly to the false positive paradox and making overall security better drives the likelihood of false detection up, our best chance is to enrich logs as much as possible and give analytic development the time it requires to get right. Once we do get it right, ensuring our logic is well documented is important to fight the tendency of the SOC to build up "tribal knowledge"—info that only a select few people have and is not written down.



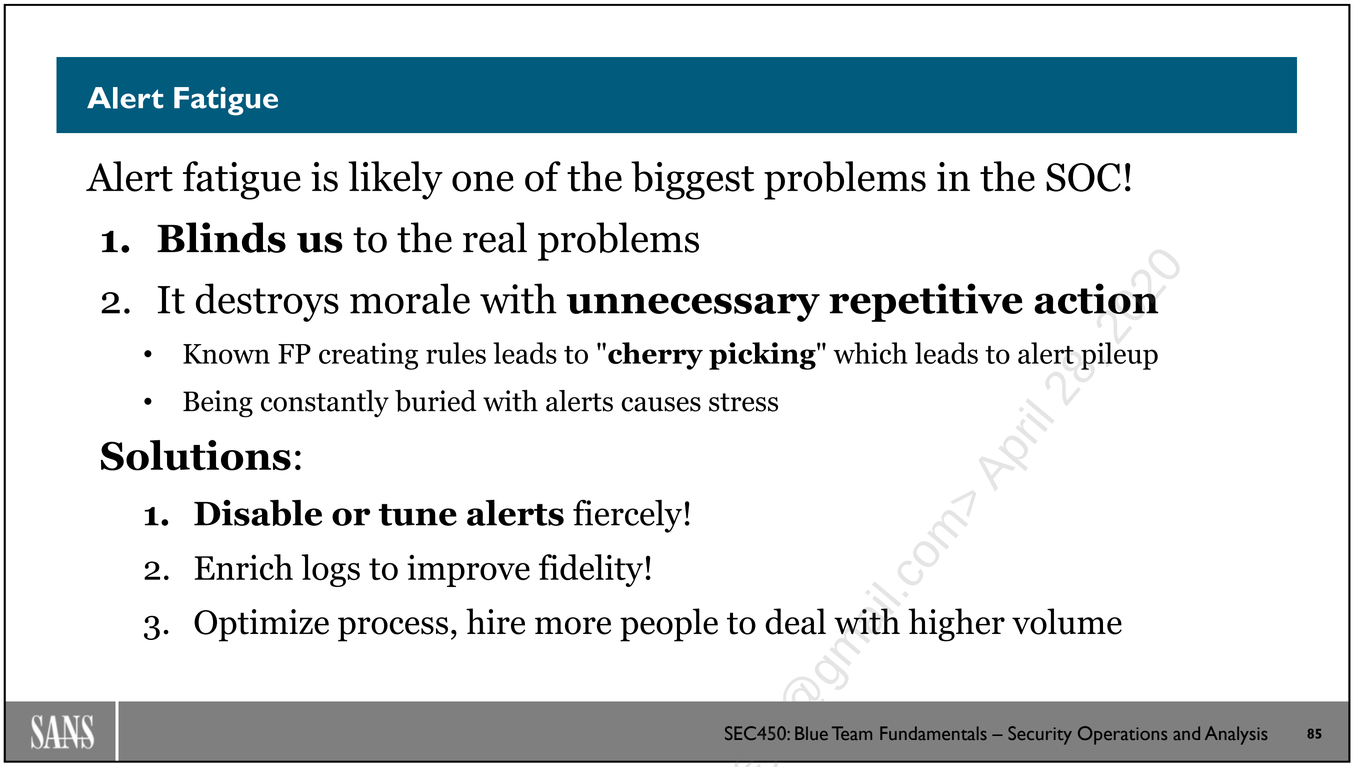
Course Roadmap

- 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**

Continuous Improvement, Analytics, and Automation

1. Improving Life in the SOC
2. Analytic Features and Enrichment
3. New Analytic Design, Testing, and Sharing
4. **Tuning and False Positive Reduction**
5. Exercise 5.1: Alert Tuning
6. Automation and Orchestration
7. Improving Operational Efficiency and Workflow
8. Exercise 5.2: Security Automation
9. Containing Identified Intrusions
10. Exercise 5.3: Incident Containment
11. Skill and Career Development
12. CTF Preparation

This page intentionally left blank.



Alert Fatigue

Alert fatigue is likely one of the biggest problems in the SOC!

1. **Blinds us** to the real problems
2. It destroys morale with **unnecessary repetitive action**
   - Known FP creating rules leads to "**cherry picking**" which leads to alert pileup
   - Being constantly buried with alerts causes stress

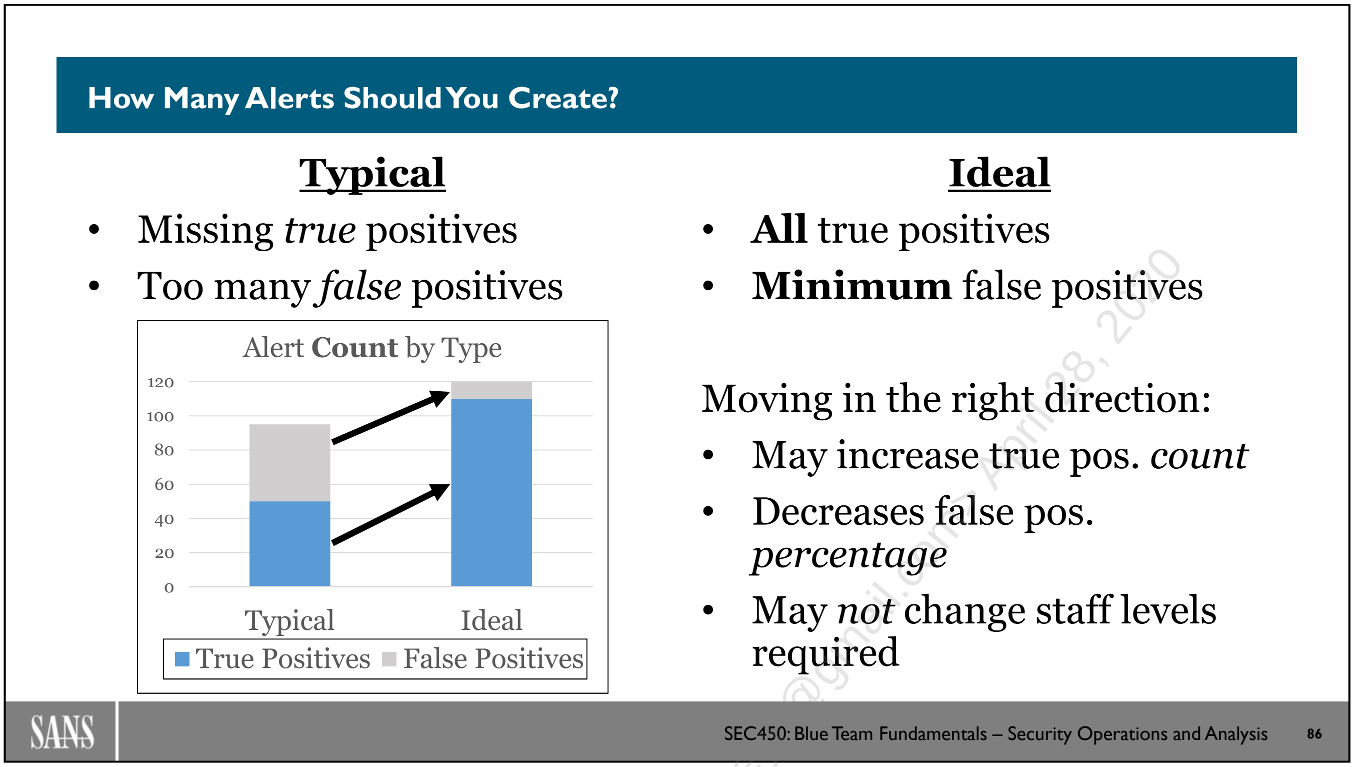
Solutions:

1. **Disable or tune alerts** fiercely!
2. Enrich logs to improve fidelity!
3. Optimize process, hire more people to deal with higher volume

Alert Fatigue

Unfortunately, alert fatigue is one of the facts of life in many SOCs. With every new security appliance we purchase, the detection capability, as well as the alert count, goes up. Since so many SOCs are constantly caught up in the whirlwind of daily operations, many can never take the time to stop and reconsider whether the data they're receiving makes sense to alert on, or even take a breather to tune their analytics.

Alert fatigue is a particularly dangerous problem because not only does it waste time, it makes analysts very unhappy and feeds the human capital cycle negatively. Here's how it happens; poorly written rules start generating a high number of alerts. Those alerts get picked up by analysts who find they are almost entirely false positives or inconsequential, and the rule gets the reputation for being low fidelity. As time progresses, if rules are not tuned, analysts start "cherry picking" the other alerts, since they know picking the low fidelity one is likely to be a waste of time. The low-fidelity alerts start to pile up, and everyone becomes more and more stressed because management is wondering why they aren't getting dealt with, but at the same time, no one wants to waste their time to clear them. Periodic and aggressive rule tuning is the way to fix this problem, but if there is no time allotted for the task, it can never get done, feeding the positive feedback loop of poor morale.



How Many Alerts Should You Create?

Typical

- Missing *true* positives
- Too many *false* positives

Ideal

- **All** true positives
- **Minimum** false positives

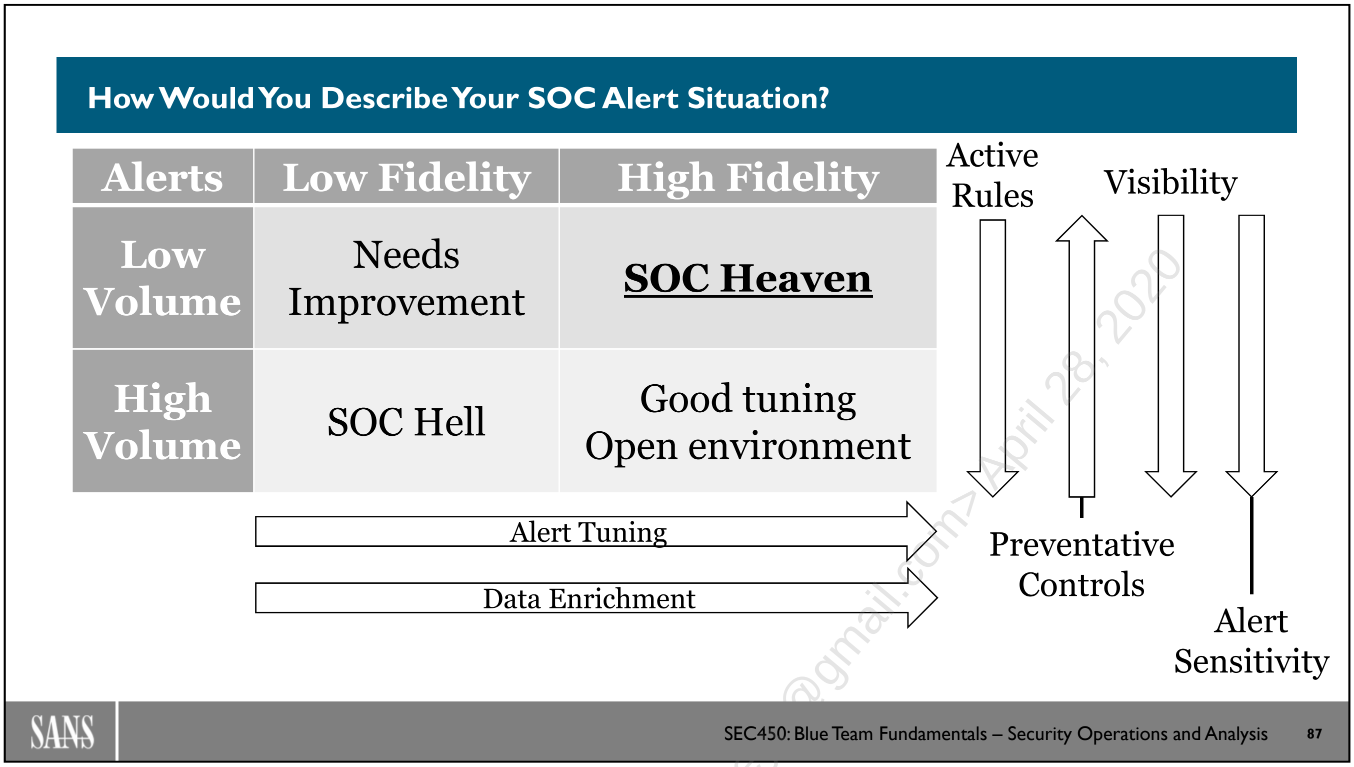
Moving in the right direction:

- May increase true pos. *count*
- Decreases false pos. *percentage*
- May *not* change staff levels required

How Many Alerts Should You Create?

Let's take a step back and think about how many alerts we *should* see. First, consider how many bad things are truly occurring in the environment. If we had perfect detection, all those things should be coming in as true positives, and seeing those alerts is a *good* thing. We also must add the inevitable small but non-zero amount of false positives, any more than that is a *bad* thing. As shown in the chart above, these two numbers added together represents the total alert *count* the SOC will see.

Notice though, to move in the right direction doesn't necessarily mean reducing the number of alerts you see. Seeing every attack in your environment means your alert count will now likely go up (because you are seeing more true positives). The good news is it may partially balance out with the reduction of false positive count, if nothing else, the percentage of alerts that are false positives will go down, which is always good.



How Would You Describe Your SOC Alert Situation?

Alerts	Low Fidelity	High Fidelity
Low Volume	Needs Improvement	**SOC Heaven**
High Volume	SOC Hell	Good tuning Open environment

Alert Tuning

Data Enrichment

Active Rules

Visibility

Preventative Controls

Alert Sensitivity

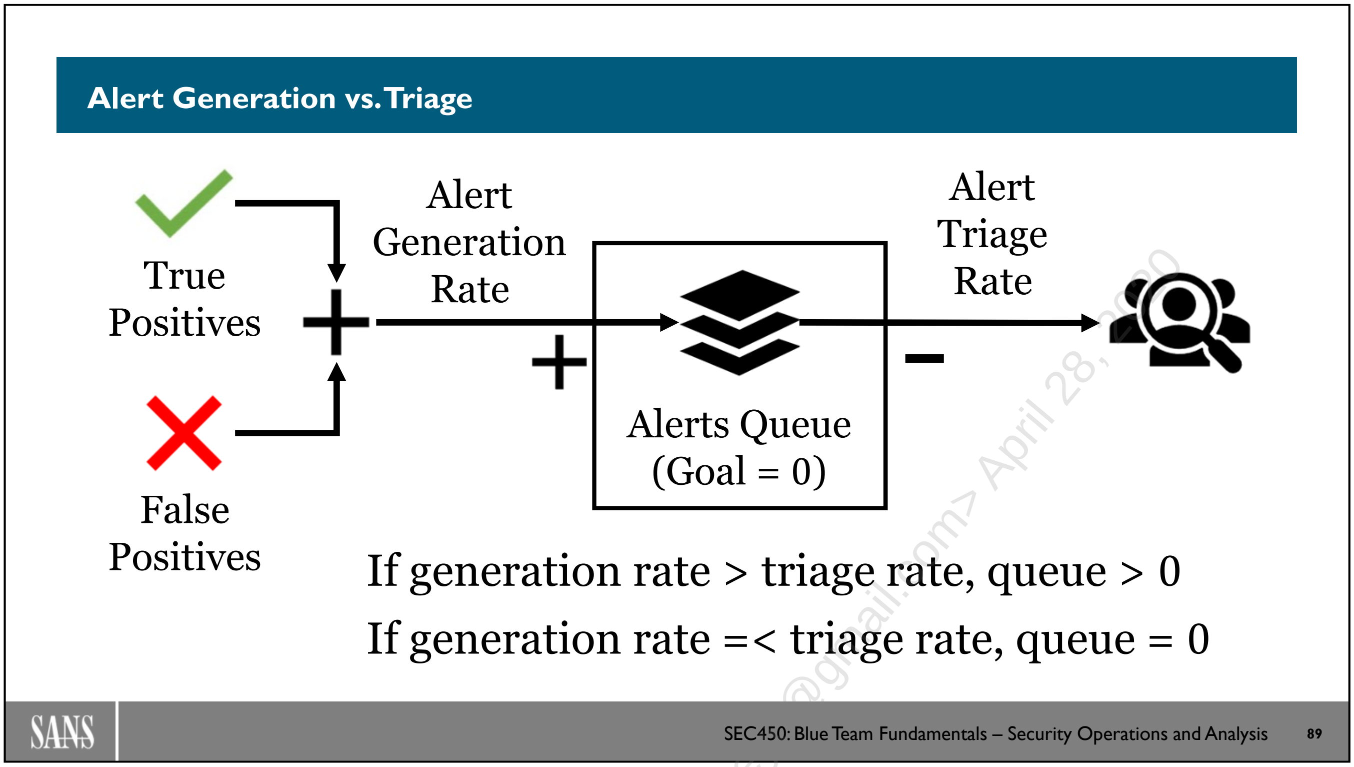
How Would You Describe Your SOC Alert Situation?

Now let's consider the variables behind those alerts, and where they place us on the chart above. If you were to think about the volume and overall average fidelity of alerts your SOC produces, where would you place it in the 4-box chart on this slide?

- **Low Fidelity, High Volume:** This is the worst place to be, and, unfortunately, a place many SOCs find themselves. In this situation, alerts are likely coming in faster than they can handle, and most of them turn out to be nothing. This is where burnout occurs.
- **High Fidelity, Low Volume:** This is the *best* place for a SOC to be. Alerts come in at a manageable pace because the business has used proper preventative security controls, and the SOC has taken the time to tune low-fidelity alerts. When alerts *do* fire, analysts can generally trust them, and this helps prevent burnout.
- **Low Fidelity, Low Volume:** Your SOC doesn't see much, but you also don't trust your rules. If you place yourself here, it may be a sign your SOC is either missing things, is new and doesn't have enough rules implemented yet, or otherwise. It's hard to say the true state of the environment with a SOC in this state since there are multiple variables that can cause it.
- **High Fidelity, High Volume:** If your SOC has tuned alerts, but *still* has a high volume of alerts, it's likely your organization needs to put in better security controls. If you have a relatively normal number of alerts, but you just don't have enough people to handle them, then perhaps more analysts are needed.

Consider which variables drive your SOC in the left, right, up or down in this chart. The arrows placed on this slide are meant to show what will happen to alert fidelity or volume as that variable is increased (moves towards the tip of the arrow). For example, increasing active ruleset will increase alert volume and increasing alert tuning will make alerts higher fidelity.

Which direction is the "right" way to go? In each arrow listed, following the arrow towards the tip is "better" for security as a whole, but also may have a negative impact on your alert rate or fidelity. When the "better security" option leads towards a negative impact on alert rate, one of the other factors will need to compensate if you need to keep things steady. For example, if you increase visibility by adding new network sensors, this will drive the alert rate up. This is not a "bad" thing – you can now see more, but to hold the alert level steady and manageable, you may need to decrease the alert rate in some other way such as adding more preventative controls (a good idea). Do NOT compensate by doing things that would have a negative impact on security like turning off active rules (intentionally blinding yourself) or decreasing alert sensitivity (potentially introducing false positives).



Alert Generation vs. Triage

True Positives

False Positives

Alert Generation Rate

Alerts Queue (Goal = 0)

Alert Triage Rate

If generation rate > triage rate, queue > 0

If generation rate =< triage rate, queue = 0

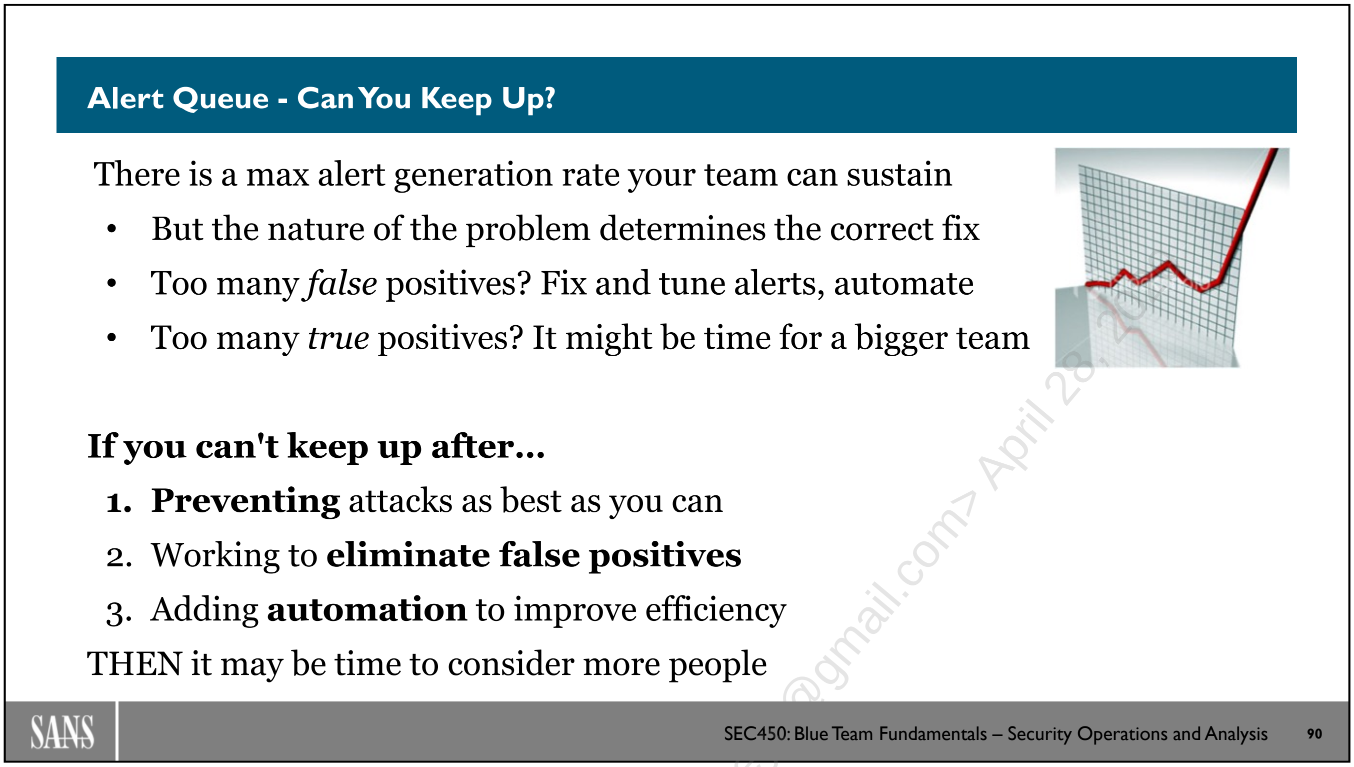
SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 89

Alert Generation vs. Triage

Once you have thought about the rate of alerts you are generating and why, consider whether that is in balance with the rate at which you can triage and investigate alerts. The alert generation rate is an independent variable when it comes to the SOC (there is a "correct" number as explained earlier). Whether or not you can keep up with that number is a different question.

We can think of this system as shown in the slide above. The inputs to the alert queue are the number of both true and false positives, the outputs from the alert queue is the rate at which you can deal with them. In a SOC, the goal is to keep the alert queue at an average size of 0, meaning you are always reviewing all alerts in a timely manner. If your alert generation rate is faster than the rate you can investigate them, you're going to run into a problem and will need to find the *correct* way to solve it. The correct way will depend on if the true positives or false positives are driving the overload of alerts.



Alert Queue - Can You Keep Up?

There is a max alert generation rate your team can sustain

- But the nature of the problem determines the correct fix
- Too many *false* positives? Fix and tune alerts, automate
- Too many *true* positives? It might be time for a bigger team

If you can't keep up after…

1. **Preventing** attacks as best as you can
2. Working to **eliminate false positives**
3. Adding **automation** to improve efficiency

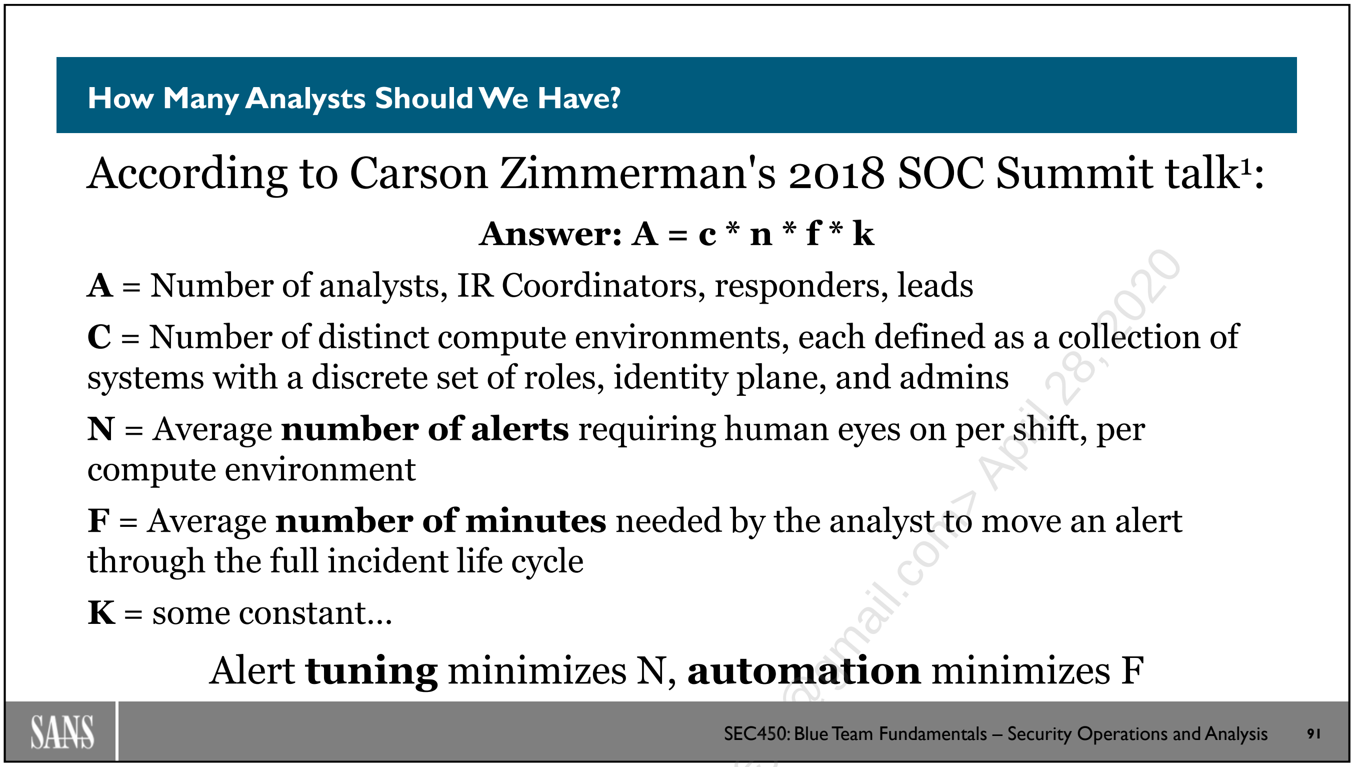
THEN it may be time to consider more people

Alerts Queue – Can You Keep Up?

SOCs often find themselves in the situation where the alert queue is getting away from them and the natural inclination is to jump to "we need more analysts!", but is that true? If your SOC finds that it has a significant number of false positives, adding more people to deal with them is the wrong solution – you'd just be addressing the symptoms, not the cause of the problem. If your SOC finds it has too many *true* positives to deal with, then it may be time to look at hiring, but only if you have explored other avenues to fix the issue first.

Let's look at what factors can help us increase the alert triage rate without adding additional people. If you find yourself unable to keep up, while there is no single answer that can always be applied, one of the following options may help you achieve balance without adding staff:

- Add additional preventative measures. This will reduce both the false positive and true positive alert generation rate, blocking more attacks and shrinking the workload the SOC needs to deal with. (Remember that some prevented attacks may still indicate an already in progress partial compromise and require investigation. Prevention alerts should still be used as a notification that you potentially need to kick off your incident response process.)
- Eliminating false positives via rule tuning. This reduces the generation rate in the best way possible, both reducing the count of alerts and increasing the percentage of true positives.
- Automation, if each person can deal with more alerts per day on average due to efficiency gains, then you may be able to get the queue under control.



How Many Analysts Should We Have?

According to Carson Zimmerman's 2018 SOC Summit talk[1]:

Answer: $A = c * n * f * k$

A = Number of analysts, IR Coordinators, responders, leads

C = Number of distinct compute environments, each defined as a collection of systems with a discrete set of roles, identity plane, and admins

N = Average **number of alerts** requiring human eyes on per shift, per compute environment

F = Average **number of minutes** needed by the analyst to move an alert through the full incident life cycle

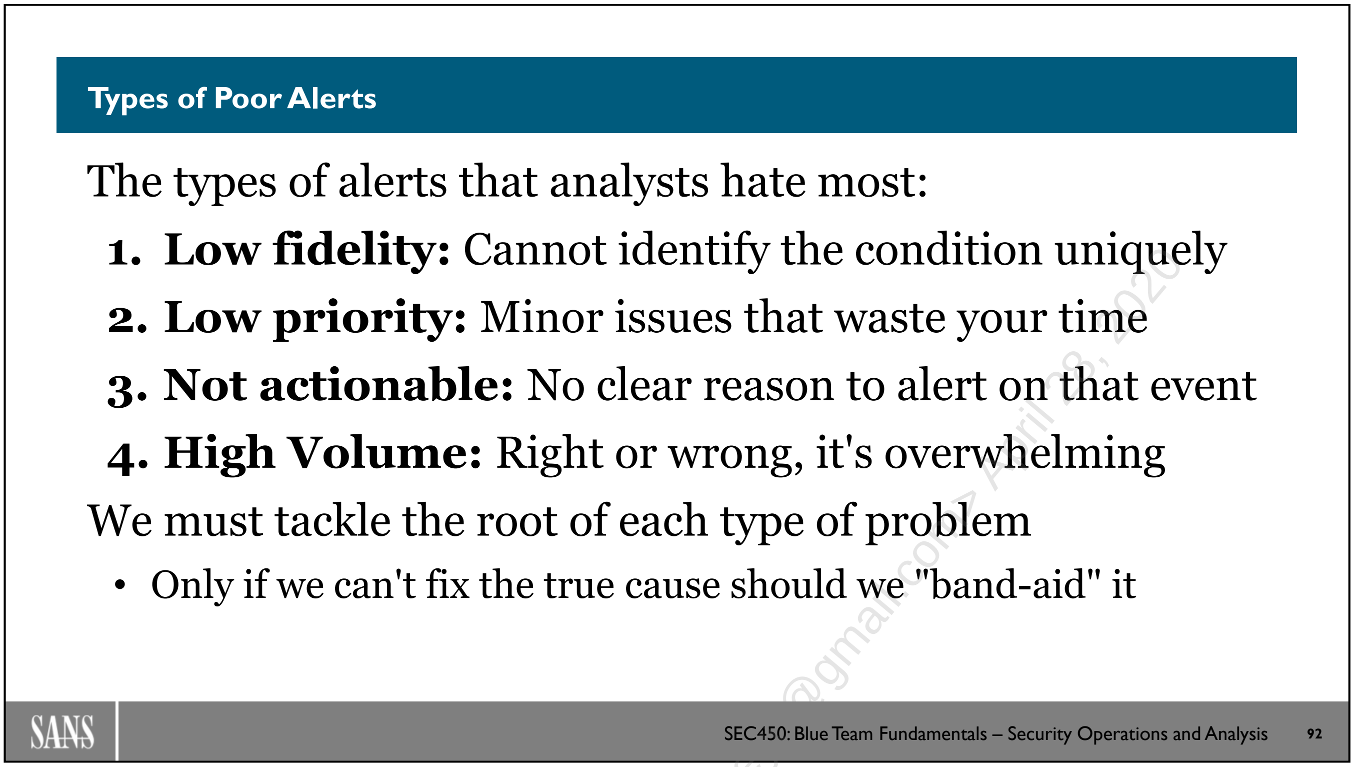
K = some constant...

Alert **tuning** minimizes N, **automation** minimizes F

How Many Analysts Should We Have?

Before we jump into alert tuning, what *is* the correct amount of analysts to have for a given environment? That is a very hard question to answer in a specific way. However, there have been attempts to create generic formulas based on the factors that have the most influence. This slide shows the calculation given as part of Carson Zimmerman's 2018 SANS SOC Summit talk in 2018.[1] Notice that the factors that make an appearance are the number of distinct computer environments that have unique roles, identities, and admins, the number of alerts the SOC receives *that require human eyes* (a key distinction vs. total count), the average time it takes to move the incident through the life cycle, and some unknown constant based on your organization. This is a very sensical approach to tackling the problem, but the main issue in coming up with a solid number will be the constant K. K must be derived from history and how alert queues have either kept up or fallen behind based on the variables that can be filled in and the count of people at the time. In this section, we will address alert tuning, which directly attacks the N factor of this equation followed by automation, which should minimize the F variable.

[1] https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1532960745.pdf



Types of Poor Alerts

The types of alerts that analysts hate most:

1. **Low fidelity:** Cannot identify the condition uniquely
2. **Low priority:** Minor issues that waste your time
3. **Not actionable:** No clear reason to alert on that event
4. **High Volume:** Right or wrong, it's overwhelming

We must tackle the root of each type of problem

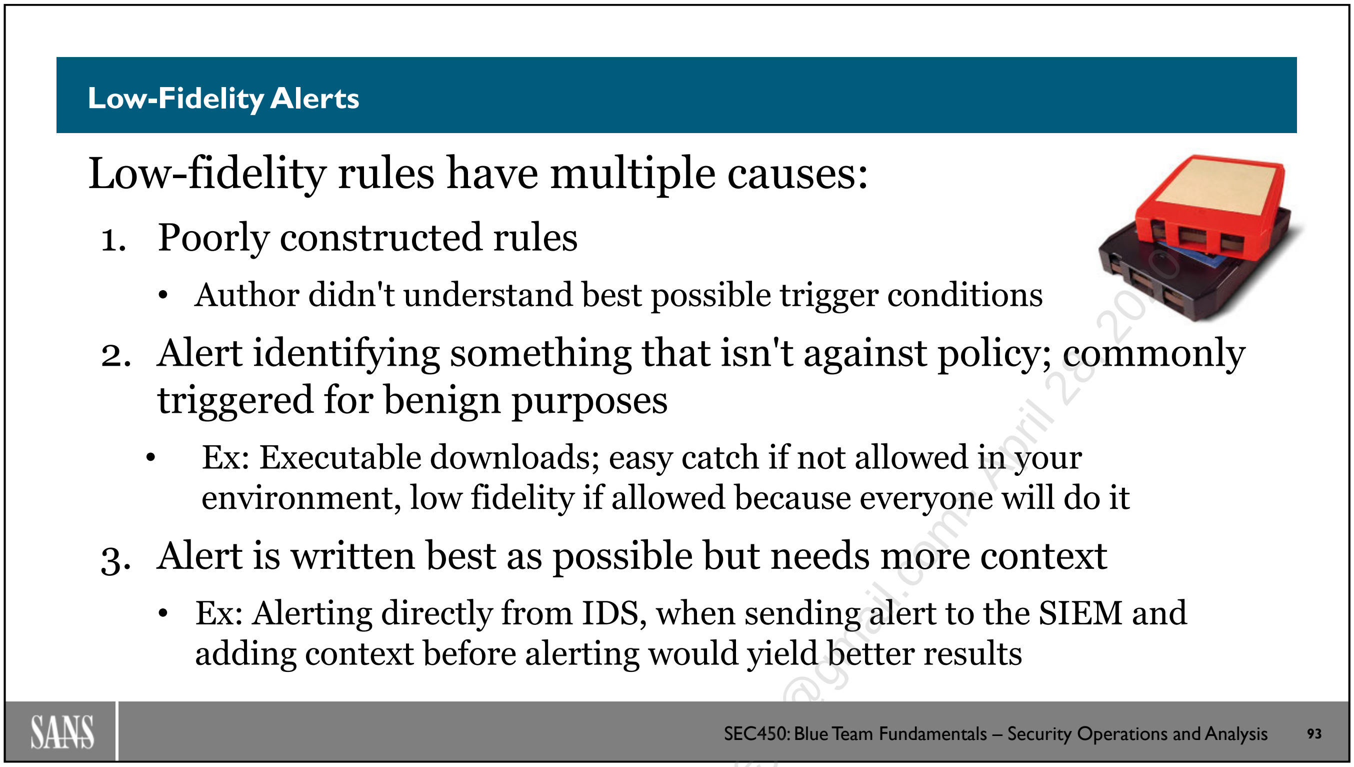
- Only if we can't fix the true cause should we "band-aid" it

SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 92

Types of Poor Alerts

When you examine the types of alerts that are analysts' least favorite, they often can be broken down into a few general categories:

- **Low fidelity:** Alerts that we *could* use if they were higher fidelity but currently cause too many false positives—these simply need to be tweaked for better logic or additional data enrichment and might become high-value
- **Low Priority:** Alerts that are accurate, but might be low priority and take up too much time for so little meaning that it's not worth the effort—policy violation rules, adware, and more often fall into this category
- **Not Actionable:** Alerts that are not actionable, or are too unclear as to why they might be bad to take investigative action on. This could be anything from mandated compliance alerting, to protocol anomalies, to out of order TCP handshakes, things that don't have any clear use case but somehow are still active as an alert in your appliances.
- **High Volume:** Alerts that, right or wrong, are constantly firing. In most cases things that fall into this category will also be in one of the others above, but being high volume makes it even more annoying. You shouldn't have any high-volume, high-fidelity alerts unless you are frequently being compromised—in which case, you know what the problem is. High-volume alerts, unless proven to be truly useful, should often be disabled or tuned.



Low-Fidelity Alerts

Low-fidelity rules have multiple causes:

1. Poorly constructed rules
   - Author didn't understand best possible trigger conditions
2. Alert identifying something that isn't against policy; commonly triggered for benign purposes
   - Ex: Executable downloads; easy catch if not allowed in your environment, low fidelity if allowed because everyone will do it
3. Alert is written best as possible but needs more context
   - Ex: Alerting directly from IDS, when sending alert to the SIEM and adding context before alerting would yield better results

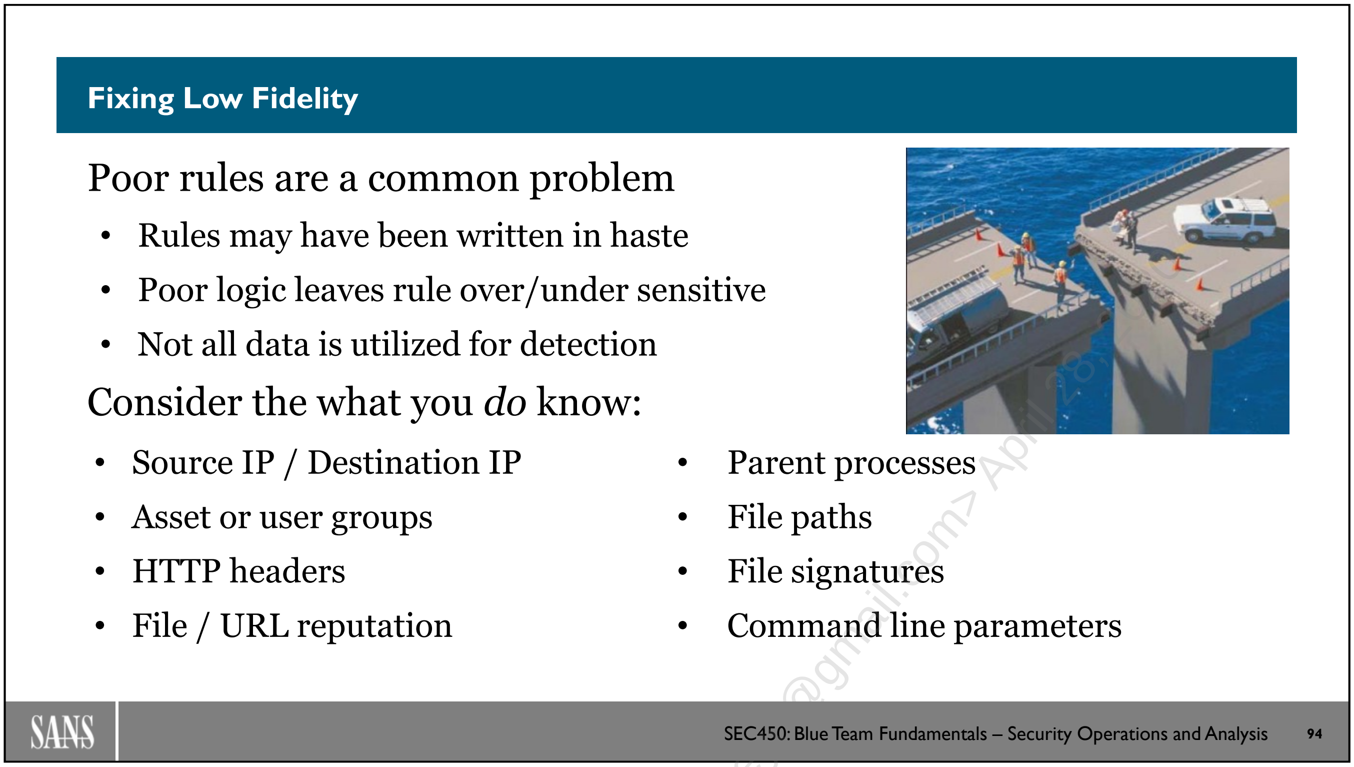
Low-Fidelity Alerts

Low-fidelity alerts are one of the big categories that nearly every SOC struggles with. There's a couple of different issues that could arise here.

One is that low-fidelity alerts may be the way they are because they were written by someone who didn't truly understand the problem and went overly broad with the detection logic. Fixing these types of alerts is a matter of discovering where the author went wrong and correcting the trigger conditions.

Another potential option is that it is simply too hard to separate good from bad in your environment due to a lack of policy defining what should and should not be happening. There may be an IDS alert for executable downloads, which is a rational thing to have in environments where it is not allowed. If your organization *does* allow them, however, you now have a condition where something that would otherwise indicate malicious activity with high fidelity will happen frequently. This means every time there's an executable download, most of them will be employees going about their business, therefore, making finding attacks based on that signature doomed to be low fidelity. The fix? Define a policy or add additional conditions to the rule so that it doesn't fire on every occasion.

The third common situation is that the rule is actually written to detect a condition in the best possible way given the information the source collects. The problem is just that the appliance doesn't truly have enough information to separate true from false positives. This situation is where enrichment and correlation can come to the rescue. Enrichment and correlation of logs is typically done at the SIEM after the appliance forwards it on. The purpose is that these additional data lookups based on the pre-existing log content may be able to further separate good from bad, turning a poor alert into a solid one!



Fixing Low Fidelity

Poor rules are a common problem

- Rules may have been written in haste
- Poor logic leaves rule over/under sensitive
- Not all data is utilized for detection

Consider the what you *do* know:

- Source IP / Destination IP
- Asset or user groups
- HTTP headers
- File / URL reputation
- Parent processes
- File paths
- File signatures
- Command line parameters

Poorly Constructed Analytic Rules

One of the possible causes of low-fidelity alerts are rules that are just poorly written. They can come from vendors as well as those on our own team, but regardless of the source, there's no doubt that this is one of the bigger categories for why we see false positives. It's understandable why it happens. Sometimes, analytic authors are either under time pressure to get *anything* that works even if false positives are included. It's also possible that they simply didn't have a big enough sample data set to verify what they wrote for false positives. The key is that once we know it is occurring, we must take steps to use our data to validate it further. Most alerts in this category will likely be on the overly sensitive side as opposed to being too conservative. That means to fix it we must look at our data in aggregate and find a pattern that can be used to improve fidelity. Be sure to consider all available data here; you never know when a field specific to your environment might be useful.

Low Priority Alerts

Consider the *lowest* priority alerts you currently act on

- Adware, toolbars, crypto-mining, file sharing, social media…
- No one likes meaningless alerts
- Can the alert response be **automated, auto-categorized**?
- Can it be made into a **weekly report** and bulk processed?

Low Priority Alerts

Low priority alerts are another common source of low morale in a SOC. These alerts are not necessarily *wrong* in their detections; it's just that they identify actions that will never make it to the top of any alert queue because they aren't interesting or high risk. These could be rules for potentially unwanted programs (PUPs) like adware toolbars, crypto-mining scripts on webpages, or policy violations like going to file sharing sites. These are items that although technically undesired, the group does not have a high interest in pursuing and won't ever pull rank on the true threats. The problem is if they never get addressed, they will simply pile up in the alert queue causing an issue because no one wants to clear them out.

There are a couple of different ways of dealing with alerts in this category and the route you take for each will depend on the nature of the alert. Some low-priority alert responses may be able to be automated. If the cleanup is something easy like adware or toolbar removal, consider creating a low-risk script that is pushed to the user's endpoint to fix the situation while the SOC stays hands off. For policy violations, an email could be sent informing the user of the noted infraction saying that it will be recorded in a report. This could be enough to get them to stop or uninstall the offending software themselves. Auto-categorization and closing is a path for alerts that you may want to know about but cannot take action on at the time. One example of this may be alerts for file sharing websites. It may not be the SOC's job to scold users for going to inappropriate sites, but the group that does may also find it to be low priority as well. What to do in this situation? Auto-categorize the alert and put the information that it occurred into a report that can be recalled later if further action or evidence is required.

Non-Actionable Alerts

As you triage alerts, ask yourself:

- "Is there a clear action to take when this fires?"
- "Do I truly care when this condition is detected?"
- "Do we run the software this is detecting attacks against?"
- For alerts with no clear actionable purpose, consider disabling

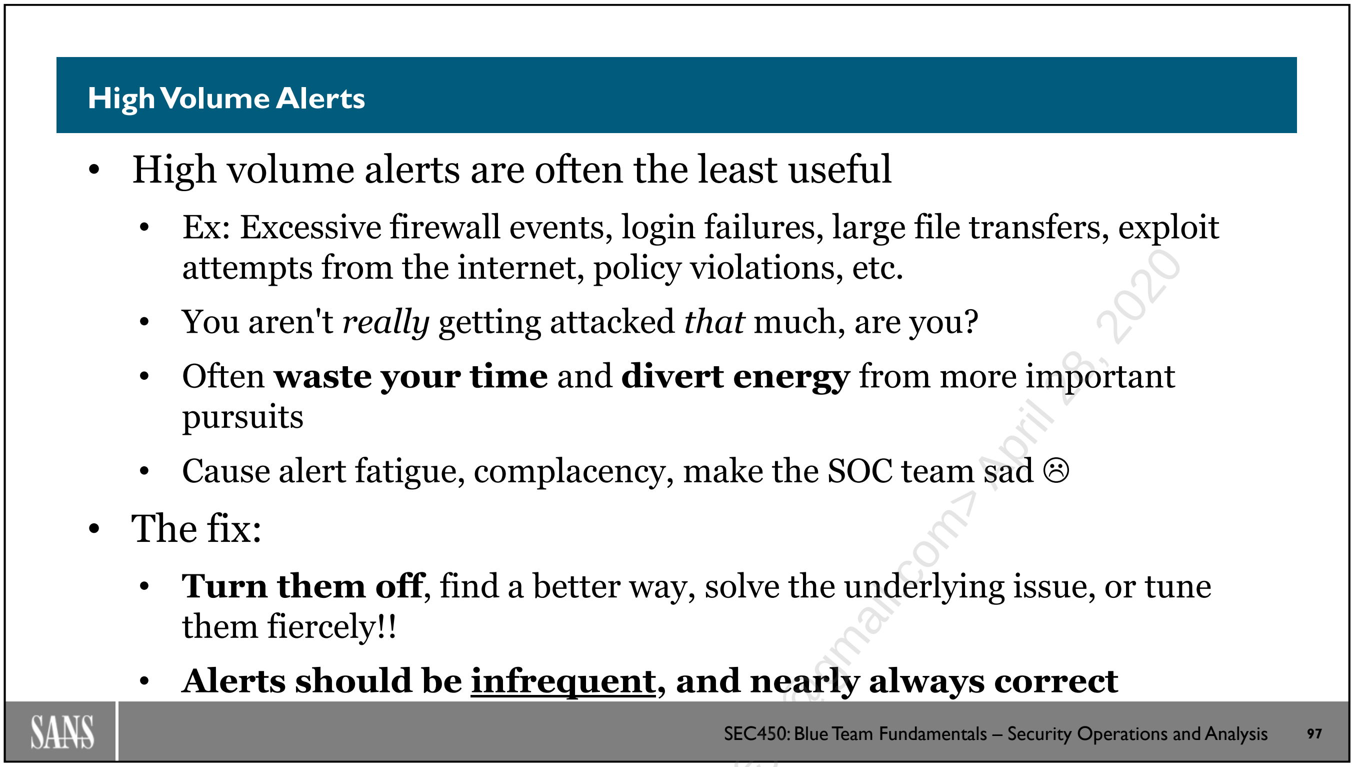
Identifying useless alerts:

- Take chartable metrics on which rule spawns each incident, are there any unused rules?
- Track true/false positive closure codes per rule

Non-Actionable Alerts

Intrusion detection systems of all sorts tend to come with hundreds or thousands of rules out of the box. These could include anything from specific evil domain names to alerts when a TCP SYN/ACK packet is sent without a preceding SYN. Some alerts will have a clear use and others may not. One of the first and easiest things we can do is check through the list of rules we have active and ask ourselves if we truly care about identifying those conditions. In many cases, the answer may be "no" either because the situation is too vague to pursue (out of order TCP handshakes), we don't care about the condition identified because it's within policy, or that it detects attacks against software that you do not use. All these items should be disabled at the source (not just ignored in the SIEM). Doing it this way conserves processing power on the device so it can focus on identifying the conditions you do care about.

The first step is to consider whether each rule should even be on in the first place. For example, do you care about WordPress exploit attempts to your DMZ web servers if they are not running WordPress? It *could* be interesting to know that people are trying to exploit WordPress, but in the pursuit of noise reduction, the alert is not actionable knowledge for you and is a creator of noise, so these types of rules are likely unneeded. This point is where the "tune up from zero" vs. "tune down from the default" decision will come into play. You are much less likely to have useless rules if you tune up from zero rules, only applying the items you know you are concerned with. Once irrelevant rules are gone, consider the "no clear action or path for investigation" rules as well. While back in the 1990s, it may have been interesting to know when something sent a TCP handshake out of order or a "Christmas tree" scan (sending packets with all flags true), tricks like that are unlikely to work these days and, therefore, are likely only another source of noise.



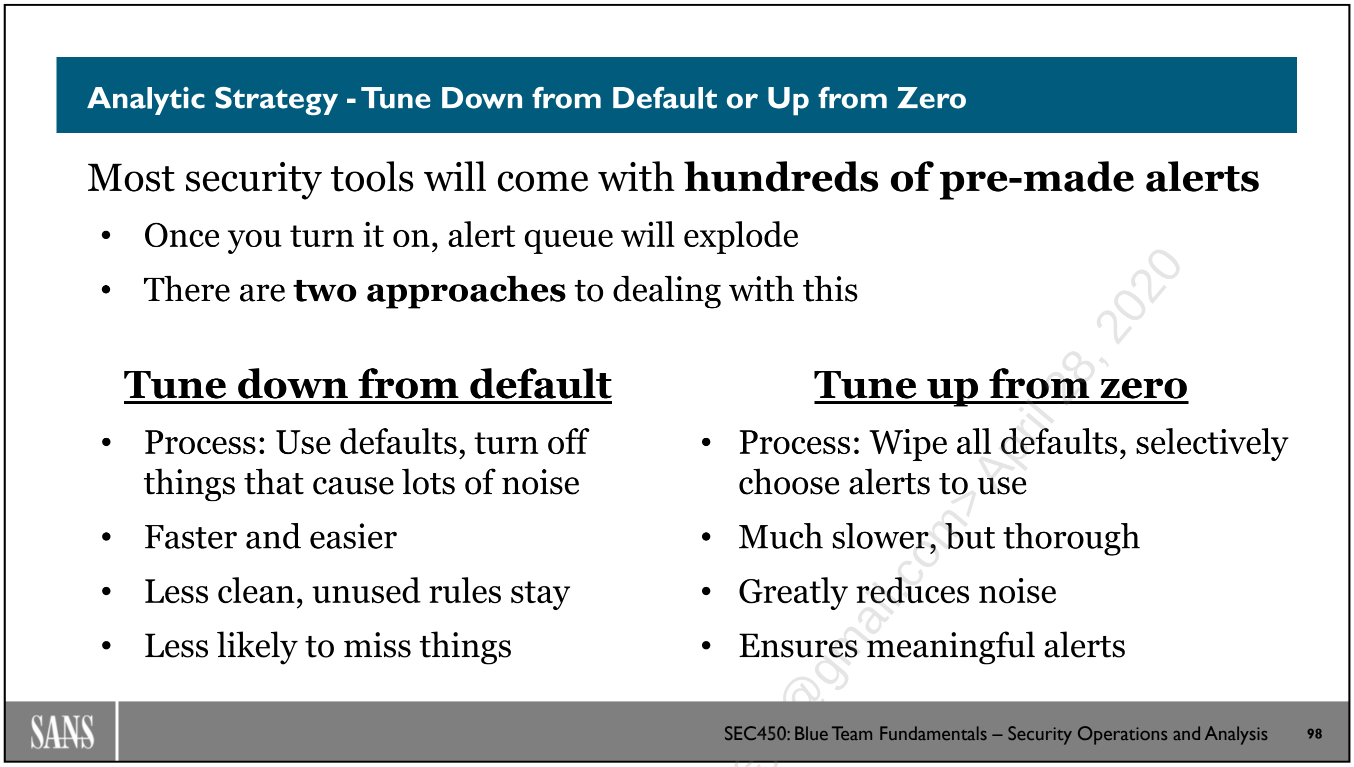
High Volume Alerts

- High volume alerts are often the least useful
  - Ex: Excessive firewall events, login failures, large file transfers, exploit attempts from the internet, policy violations, etc.
  - You aren't *really* getting attacked *that* much, are you?
  - Often **waste your time** and **divert energy** from more important pursuits
  - Cause alert fatigue, complacency, make the SOC team sad ☹
- The fix:
  - **Turn them off**, find a better way, solve the underlying issue, or tune them fiercely!!
  - **Alerts should be <u>infrequent</u>, and nearly always correct**

High Volume Alerts

There are certain alerts (and you know which ones they are for your environment) that are always going off, and almost never amount to anything worthwhile. As a rule, things that happen the most often are also the least likely to be useful or interesting, and that applies to alerts as well. This category represents those alerts that are going off every single day and are likely to be useless because, let's face it, you aren't seeing advanced attacks every single day.

The underlying problem with these alerts is that although they may technically work correctly, the situations they identify are often things that can be mistaken for everyday errors or are simply not all that interesting. These alerts are often more difficult to feel ok turning off, but consider this – if they aren't finding anything useful 99.9% of the time, taking the team's time, and contributing to analysts hating their job, do you *really* want those alerts turned on? For alerts that fall into this category, consider either turning them off outright, finding a better way to detect the same condition, solving the underlying issue that is causing them, or adding additional required conditions before they fire. Believe me, for the health of your team and your sanity, it's worth it!



Analytic Strategy - Tune Down from Default or Up from Zero

Most security tools will come with **hundreds of pre-made alerts**

- Once you turn it on, alert queue will explode
- There are **two approaches** to dealing with this

Tune down from default

- Process: Use defaults, turn off things that cause lots of noise
- Faster and easier
- Less clean, unused rules stay
- Less likely to miss things

Tune up from zero

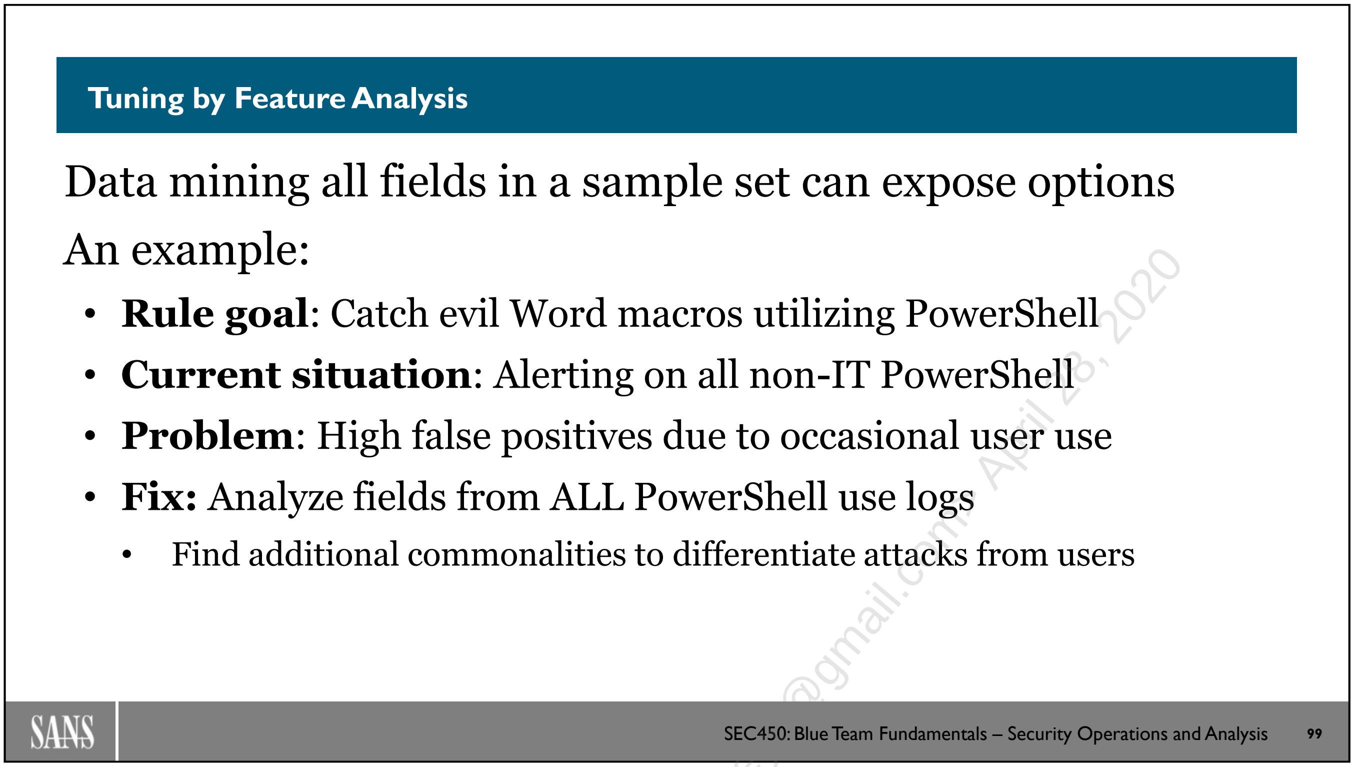
- Process: Wipe all defaults, selectively choose alerts to use
- Much slower, but thorough
- Greatly reduces noise
- Ensures meaningful alerts

Analytic Strategy - Tune Down from Default or Up from Zero

Vendors love to cram tons of analytic rules into their intrusion detection solutions and do this so they can give their customers a quick start and claim that when customers buy their solution, it will be "turn-key" easy to start and detecting evil from day one. While this may be true in some cases and well-intentioned, having an ever-increasing number of pre-created detections out of the box to keep up with competitors might not be a good thing for alert queues. The problem with tons of pre-made alert rules is that many of them will either not apply to your environment at all, detecting attacks for services and protocols that you don't run, and others may be written poorly by vendors who were more interested in chasing pre-made alert numbers they can use as "features" for sales. That means you regardless of what you buy, you will likely need to do heavy tuning of what comes out of the box.

When it comes to keeping unnecessary alerts at bay, there are two schools of thought—tuning down from default settings, and tuning up from zero. The "tune down from default settings" option is the "fix the mess" approach where you leave the rules on but aggressively suppress and tune the ones that cause problems. This method gets you going quickly but bombards you with noise while the tuning process is underway. On the one hand, it can be good in that you don't have to examine each rule and try to interpret if you need it or not. It can also be less optimal since you are likely to still leave on rules that are unneeded, but never fire, wasting potential processing potential that could be used for other rules.

The other school of thought is the "tune up from zero" approach, where you turn *all* rules off and then take the time to go through one by one, activating the ones you know are relevant. Although this approach certainly takes longer, is more painful, and requires staff that understands each rule to make the call, in the long run, it is likely better. It ensures your appliances are running at peak efficiency and reduces false positives due to irrelevant rules to near zero. With this approach, analysts don't have to wonder whether each alert is something they should care about or not because they know each item was individually selected as being useful.



Tuning by Feature Analysis

Data mining all fields in a sample set can expose options

An example:

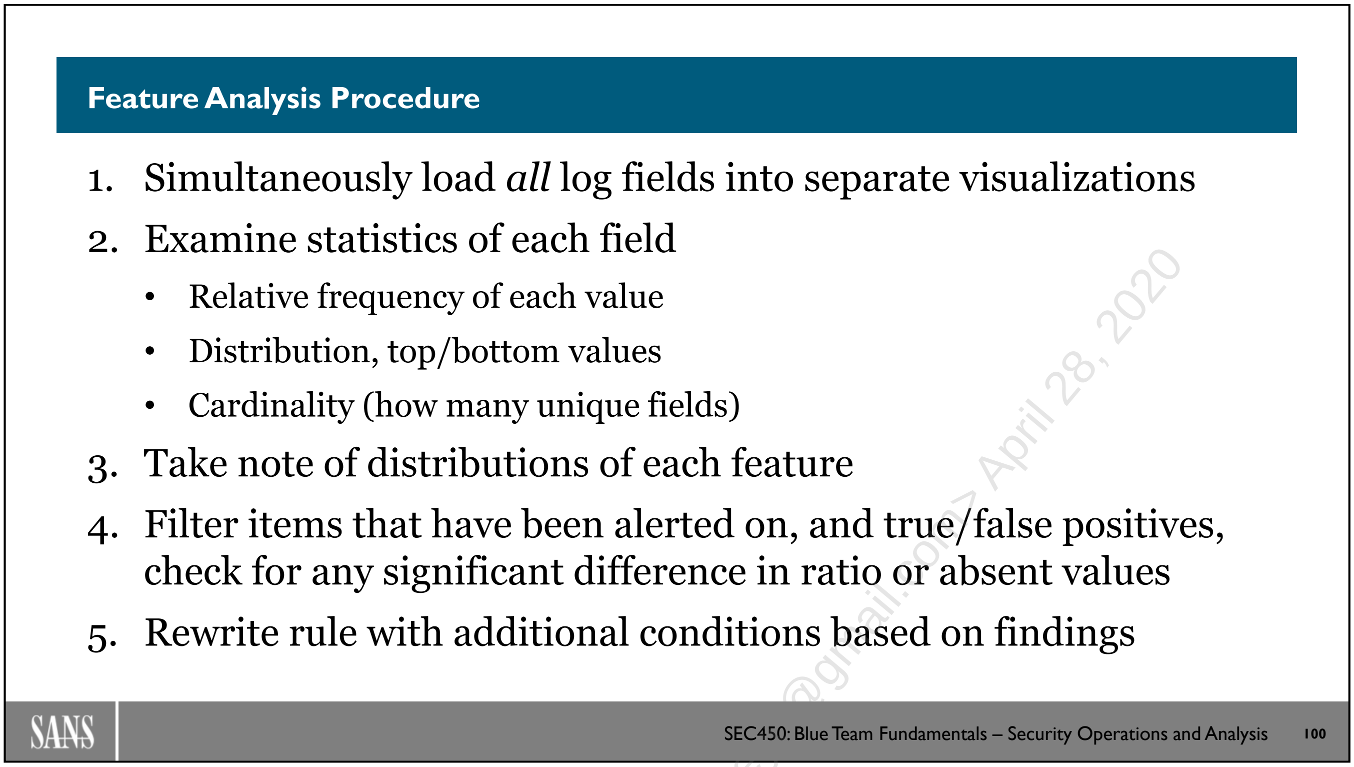
- **Rule goal**: Catch evil Word macros utilizing PowerShell
- **Current situation**: Alerting on all non-IT PowerShell
- **Problem**: High false positives due to occasional user use
- **Fix:** Analyze fields from ALL PowerShell use logs
  - Find additional commonalities to differentiate attacks from users

SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 99

Tuning by Feature Analysis

In every environment, there are suboptimal rules. The problem is it can be difficult to figure out how to make them better without adding additional info, enrichment or log correlation. While those are the best methods, sometimes there are easy wins to be found just by more closely analyzing the data set you have for additional patterns that were not noticed as relevant when the rule was first developed.

A simple example could be a SOC monitoring for suspicious usage of the powershell.exe process outside of the IT group. While many script-based pieces of malware utilize PowerShell and so do people in the IT department, the average user doesn't. Perhaps a rough rule was created to alert on any PowerShell processes for users that were not part of IT but has been causing false positives because the occasional user does still open PowerShell for some purpose. How can we tell the difference in more detail without using additional enrichment or correlation tricks? Look at all the PowerShell logs and compare the fields in the ones that are true positives vs. the ones that are false positives, and look for more fields we can add into the logic before an alert fires. Although this can be hard to do one field at a time, analyzing all fields at once with filtering capability can make this task much easier and encourage differences to come to light.



Feature Analysis Procedure

1. Simultaneously load *all* log fields into separate visualizations
2. Examine statistics of each field
   - Relative frequency of each value
   - Distribution, top/bottom values
   - Cardinality (how many unique fields)
3. Take note of distributions of each feature
4. Filter items that have been alerted on, and true/false positives, check for any significant difference in ratio or absent values
5. Rewrite rule with additional conditions based on findings

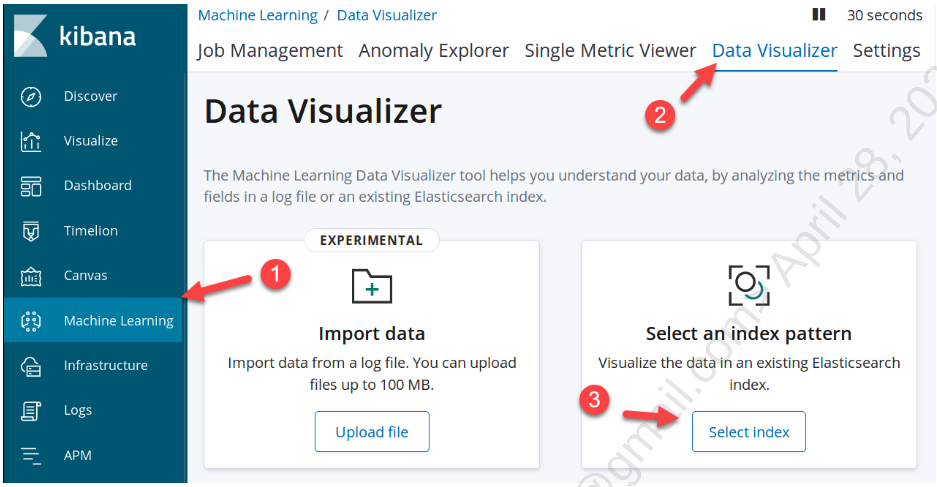
Field Content Analysis Procedure

The exact tools you use to perform this analysis is up to you. Some SIEMs or other analytic tools may have this functionality built in. Others may need to create the analysis manually, or even use something like Excel. In general, here are the steps:

1. Gather all logs in the population where you want to detect the malicious condition. If for example, you are interested in analyzing Windows login events, this could be a filter for all 4624 and 4625 events that have occurred in total.
2. Look at *all* the separate fields in the log recorded for that event by graphing them simultaneously. Your goal is to find the thing that is most specific to the condition you are trying to detect to use as an identified. For example, yes, all malicious connections may be TCP, but that would be a poor way to detect something, a domain name, port number, or field value is likely a much better option. For Windows logins, this might be the domain, username, source IP, and other key information. You can safely ignore fields that have pseudo-random data such as unique IDs generated for login sessions as it is clear fields like this will not have commonality. Key statistics to check are the relative frequency, population distribution, and least/most frequently occurring values in each field.
3. Save this information via screenshot, data export, or other for visual comparison or open another tab so that the next filtered set of data can be looked at simultaneously.
4. Apply the same filter that your alert is based on to the whole population of events and see which field statistics change. Obviously, the fields that you currently key off will stand out, but are there any other fields that change appreciably that aren't accounted for in your rule logic? What we're looking for are additional conditions that can be applied to the pre-existing rule that will help separate the true from false positives that hadn't been noticed. Since we are looking at statistics for *all* fields, filtering on additional items may help new options for conditions stand out.
5. As a pre-emptive check that our rule it will not fail spectacularly, filter historical data to give the new logic a test run. If all is good, you have a strong reason to believe you have improved your rule! Apply the findings of this analysis to your rule logic, rewrite or put in a request to rewrite the rule as needed, supplying the data for your analysis as a backup for justification.

Kibana Makes This Easy!

The Data Analysis tool in Kibana can do this!



Data Visualizer

The Machine Learning Data Visualizer tool helps you understand your data, by analyzing the metrics and fields in a log file or an existing Elasticsearch index.

Import data

Import data from a log file. You can upload files up to 100 MB.

Select an index pattern

Visualize the data in an existing Elasticsearch index.

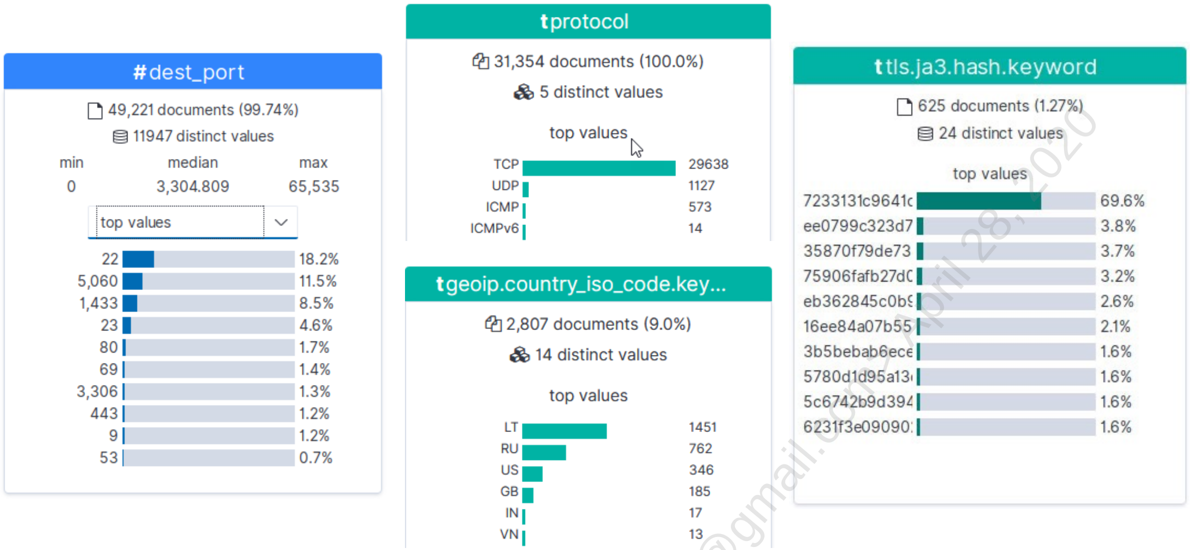
Kibana Makes This Easy!

Fortunately, the free “Basic” license version of Kibana[1] includes a feature called the Data Visualizer to perform exactly this type of analysis (the Apache 2.0 License free version included in the VM however, does not include this feature). Even if you do not use the Elastic Stack and Kibana in your SOC, setting it up in a virtual machine and using the "Import Data" feature shown on the slide can make it easy to import a sample of log data and analyze it. A docker-compose file for quickly and temporarily bringing up an Elasticsearch and Kibana instance is included in the class wiki.

To access the feature, click on the Machine Learning section of Kibana (which will only be present if you use Basic license version of Kibana), then select Data Visualizer at the top, and Select an index pattern. The index pattern is the name for a pre-imported set of data held in Elasticsearch. If you have previously used the Import data feature, you can set the index pattern at import time.

[1] https://www.elastic.co/subscriptions

Feature Analysis Example



#dest_port

49,221 documents (99.74%)

11947 distinct values

min	median	max
0	3,304.809	65,535

22	18.2%
5,060	11.5%
1,433	8.5%
23	4.6%
80	1.7%
69	1.4%
3,306	1.3%
443	1.2%
9	1.2%
53	0.7%

t protocol

31,354 documents (100.0%)

5 distinct values

top values

TCP	29638
UDP	1127
ICMP	573
ICMPv6	14

t geoip.country_iso_code.key...

2,807 documents (9.0%)

14 distinct values

top values

LT	1451
RU	762
US	346
GB	185
IN	17
VN	13

t tls.ja3.hash.keyword

625 documents (1.27%)

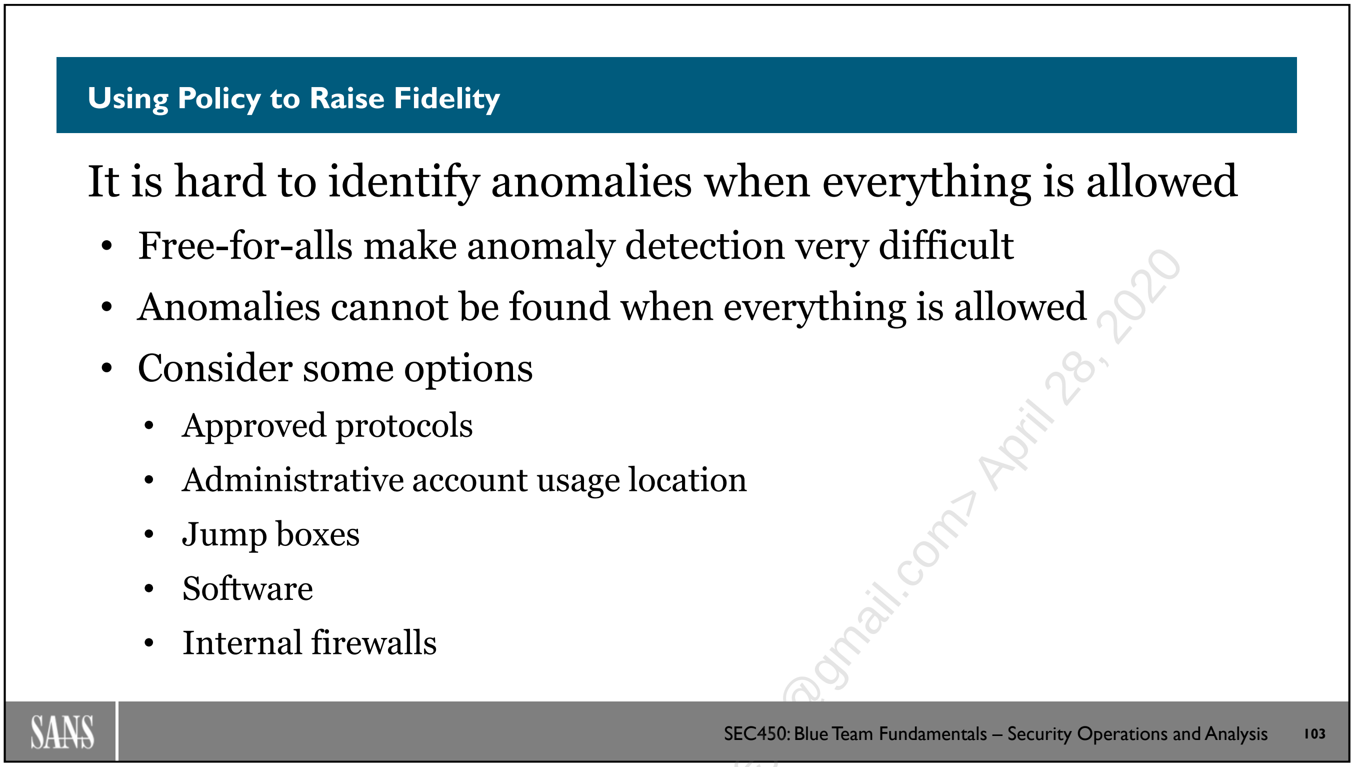
24 distinct values

top values

7233131c9641c	69.6%
ee0799c323d7	3.8%
35870f79de73	3.7%
75906fafb27d0	3.2%
eb362845c0b9	2.6%
16ee84a07b55	2.1%
3b5bebab6ece	1.6%
5780d1d95a13	1.6%
5c6742b9d394	1.6%
6231f3e09090	1.6%

Feature Analysis Example

This slide shows part of the results you would see when using the data visualizer built into Kibana. As you can see, the fields with terms have a distribution showing the count of each term in the dataset and the numeric values can be displayed either as a top value list (destination_port), or a histogram of the distributions. Both views can be useful at spotting patterns depending on the data. The great part about this feature in Kibana, though, is that all these graphs get created automatically for you with the tool and then you can use the built-in filtering search box to type in exploratory queries on your data and observe if distributions or counts change. Doing this can be a quick way to verify a hypothesis about features of interest within your logs when crafting a new analytic.



Using Policy to Raise Fidelity

It is hard to identify anomalies when everything is allowed

- Free-for-alls make anomaly detection very difficult
- Anomalies cannot be found when everything is allowed
- Consider some options
  - Approved protocols
  - Administrative account usage location
  - Jump boxes
  - Software
  - Internal firewalls

SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 103

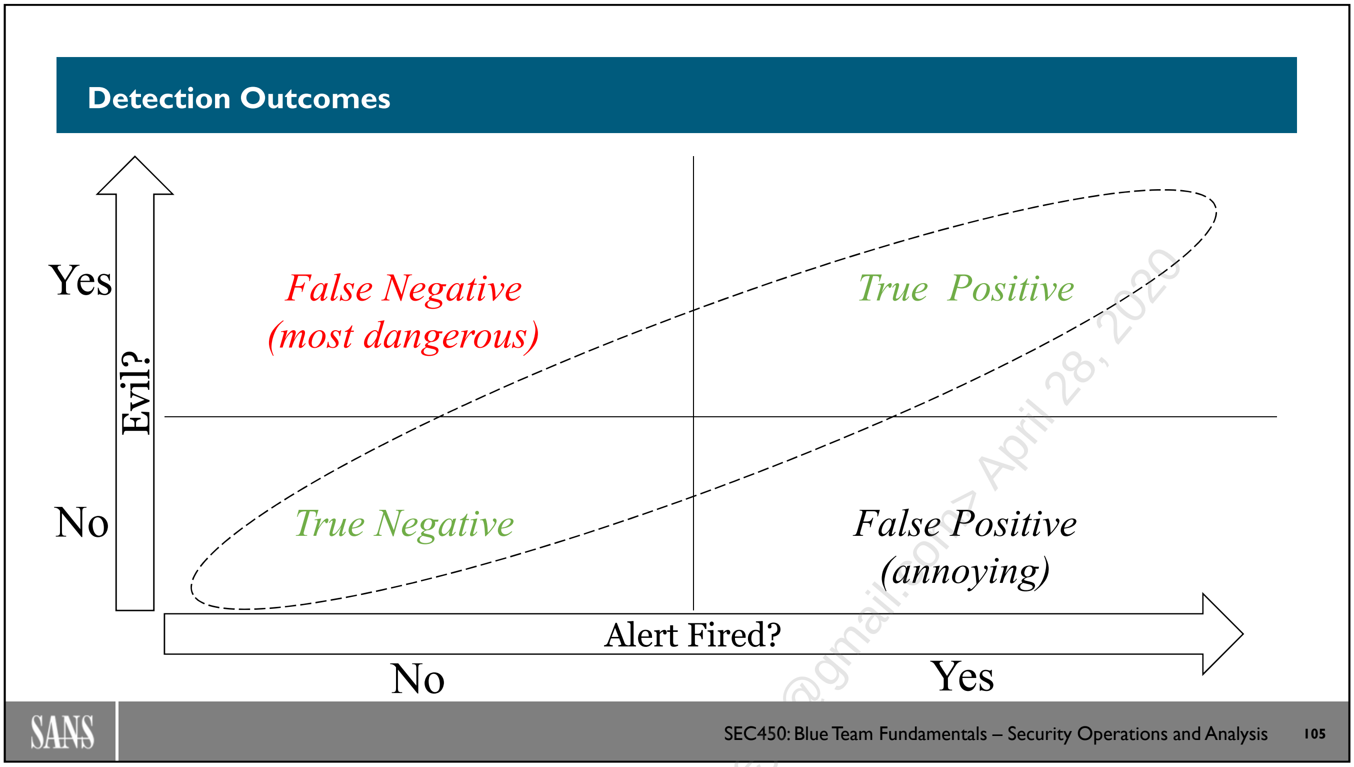
Using Policy to Raise Fidelity

One of the best things you can do to raise fidelity of alerts has nothing to do with the alerts themselves at all! Merely having a policy about which activities, software, and traffic types are allowed, especially in a whitelist type fashion, makes threat detection fidelity skyrocket. When you know that administrators must act a certain way, use certain tools, and send traffic in certain ways, you have created an amazing way to detect when something is out of place! That means one of the best ways to increase the fidelity of rules is to apply the logic of what your users are allowed to do (or create it if it doesn't exist) and monitor for violations of that which are commonly associated with attacks.

You are likely doing this to some extent already when it comes to policy violations for websites and software. But there are some specific policies that, when paired with alert rules, make it very difficult for adversaries to operate without violating one of them and tipping off the team. Here are some options to get the ideas flowing:

- Protocols: Not just which ones are allowed (can users use SSH, FTP, etc. or not), but *where* are they allowed to and from. Of course, just spotting an unallowed protocol like VNC may be a good tip-off that an attacker has installed a remote backdoor. But with a "where" definition, you can go further. For example, if you find that a server that isn't on the "FTP allowed" list attempts to contact the internet on those ports, it's highly likely that something suspicious could be going on.
- Administrative Account Usage: Administrative accounts need to be protected at all costs, and one of the ways of doing that is never using them on a machine that isn't the specific server or desktop they were created to administer. For example, domain administrators should not be logging on to any regular desktops or servers at *all*—only domain controllers. With this policy, not only can the password not be stolen with a tool like Mimikatz, but the defined containment of that account means that *any* attempt to use a domain administrator account outside of the norm (which is a go-to move for attackers) will immediately set off a high-fidelity alert.

- Jump Boxes: Similar to administrative account usage, this defines that all logins to a certain set of servers or other systems must pass through a "jump box" server first, and then log in from there. With a policy like this defined, all attempts by attackers to log in directly to servers will be inherently suspicious. Since it is unlikely attackers will know of the jump box's existence or the procedure to use it, they will reveal themselves by trying to login directly, which is a condition easy to create a high-fidelity alert for—"all login attempts for system x that do not come from source IP [jump box]."
- Software: If whitelisting is in play at your organization, the attempt to run any unknown executables, especially ones that have no reputation and aren't signed, can be an immediate high-fidelity alert that otherwise might not work without the whitelist policy.
- Internal Firewalls: Internal firewalls increase visibility and control where you might not otherwise have it and can also sense protocols being used from one subnet to another. An internal firewall can apply numerous detections, plus they are a fundamentally good idea for restricting lateral movement.

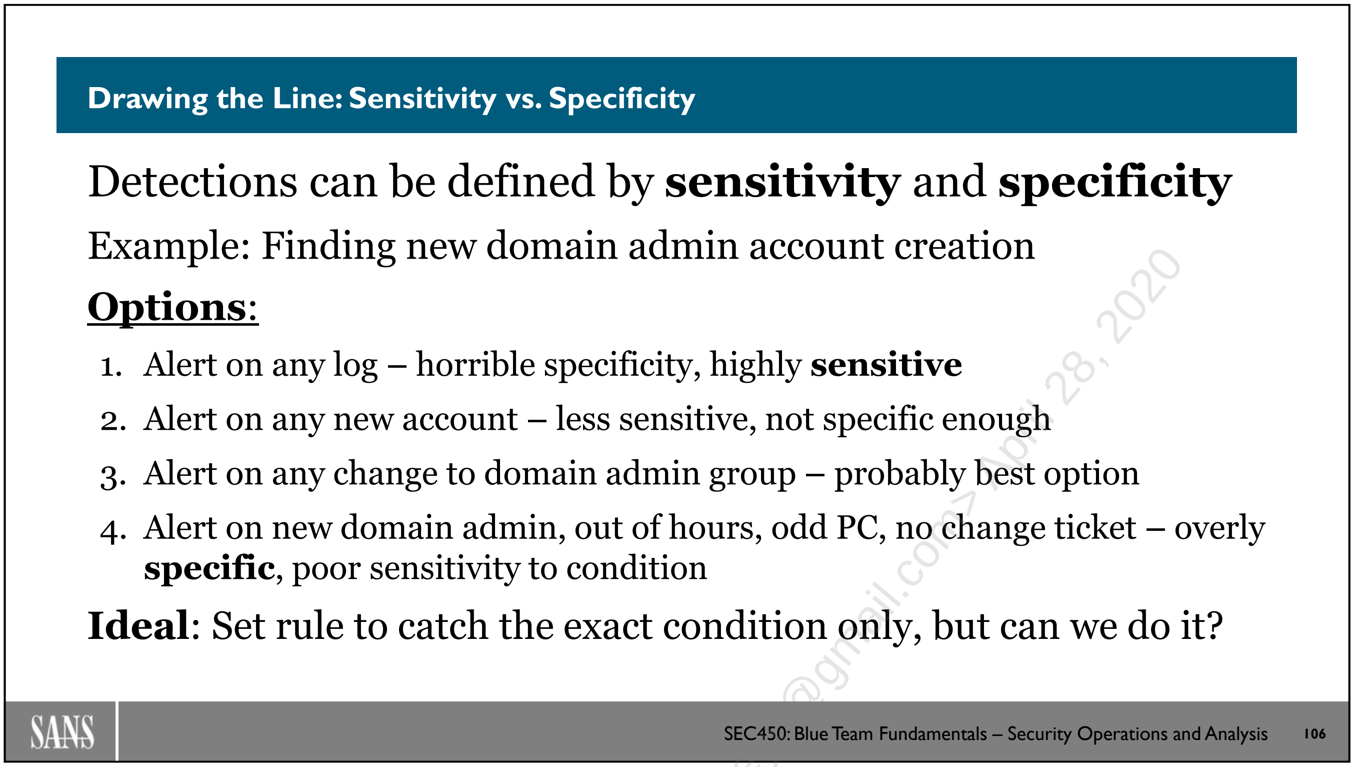


Detection Outcomes

Detection Outcomes

When it comes to alert validation, it is useful to consider the possible outcomes of any given situation. We have true negatives for when we shouldn't alert and don't, and true positives for when we should alert and successfully do identify the condition of interest. Ideally, everything falls into these two buckets, but unfortunately, this isn't reality, and it's the other two options that the SOC is in a constant battle to control. On the one hand, we have the ever-hated false positive that occurs when a rule is oversensitive and fires when it shouldn't. On the other hand, we have the less discussed and *worse* error—the false negative—when an alert doesn't fire, but it should.

If you consider how you would design your detection strategies in the real world, what we likely will pick is to make as many true positives and negatives as possible, but we will also likely tune our appliances to produce more false positives to avoid false negatives. Why can't we design for zero false positives or false negatives? Because these two are intrinsically linked in any detection system and bringing one down almost always necessitates the other going down. What this means is that as much as we want to have zero false positives, unless we are willing to risk missing things, achieving that is almost impossible without perfect knowledge of all attacks. What drives how much they are linked is how much data we have to separate the positive vs. negative condition.



Drawing the Line: Sensitivity vs. Specificity

Detections can be defined by **sensitivity** and **specificity**

Example: Finding new domain admin account creation

Options:

1. Alert on any log – horrible specificity, highly **sensitive**
2. Alert on any new account – less sensitive, not specific enough
3. Alert on any change to domain admin group – probably best option
4. Alert on new domain admin, out of hours, odd PC, no change ticket – overly **specific**, poor sensitivity to condition

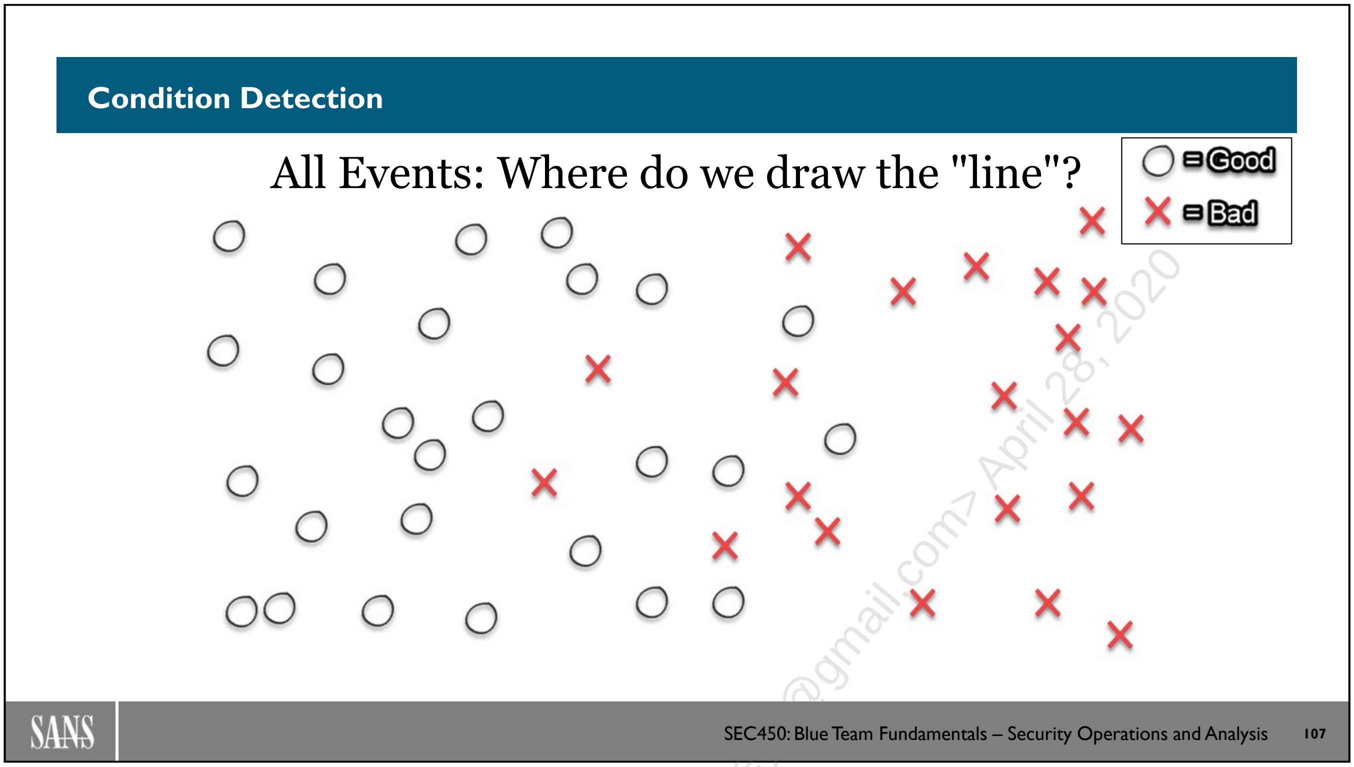
Ideal: Set rule to catch the exact condition only, but can we do it?

Drawing the Line: Sensitivity vs. Specificity

Alert rules can be thought of using two terms: Sensitivity and specificity. Sensitivity can be thought of as "how easy it is to trigger it," and specificity can be thought of as "how many conditions have to be true for it to trigger." As you may already see, these two items are inherently inversely correlated. Something that has more conditions necessarily is necessarily less sensitive since it will take a more specific situation to set it off.

As an example, consider if you want to make a rule to catch illegitimate new domain admins being created by attackers (a great rule that everyone should have.) To create the rule, you must pick the conditions that will set it off, therefore setting its specificity. Obviously, you want as few false positives as possible and ideally zero false negatives, so there are a couple of ways you could approach it. One (obviously ridiculous) way to be absolutely sure you will never have a false negative is to alert on *all* events. There's no doubt that you will indeed catch every instance of a new domain admin being created with this technique, but the number of false positives it would create would be outrageous—it's just too sensitive. More in the realm of possibility, you might enact a rule for alerting every time there is a new account made. This is more specific and, therefore, would also work and not miss anything but would still create a lot of unnecessary false positives. You only want to know about domain admins, not *everyone.* Maybe you decide you want to crank it up and make what is probably the most reasonable decision then—alert on all changes to elevated permission groups, such as domain admins. This method will meet your condition and produce a minimum of false positives since it is not a frequent event. Your organization likely will hire some domain admins over time though, so it is not a perfect rule, but it is still specific and still sensitive enough to catch everything. The final option would be the "zero false positive" option where you look at your threat model and think, an attack doing this would only do it in the middle of the night, on an unexpected PC without a change ticket, so you implement that rule. How will it work? Well, you sure won't get any false positives, but now you might be too specific and start producing the dreaded false negatives.

Why is it so hard to write a perfect rule? The short story is it is the lack of perfect knowledge of your environment and the context of the situation. In most cases, writing the perfect rule is strictly unattainable, so we must decide where to set the bar between accepting false positives and false negatives. Over the next few pages, we'll visualize this problem more closely.



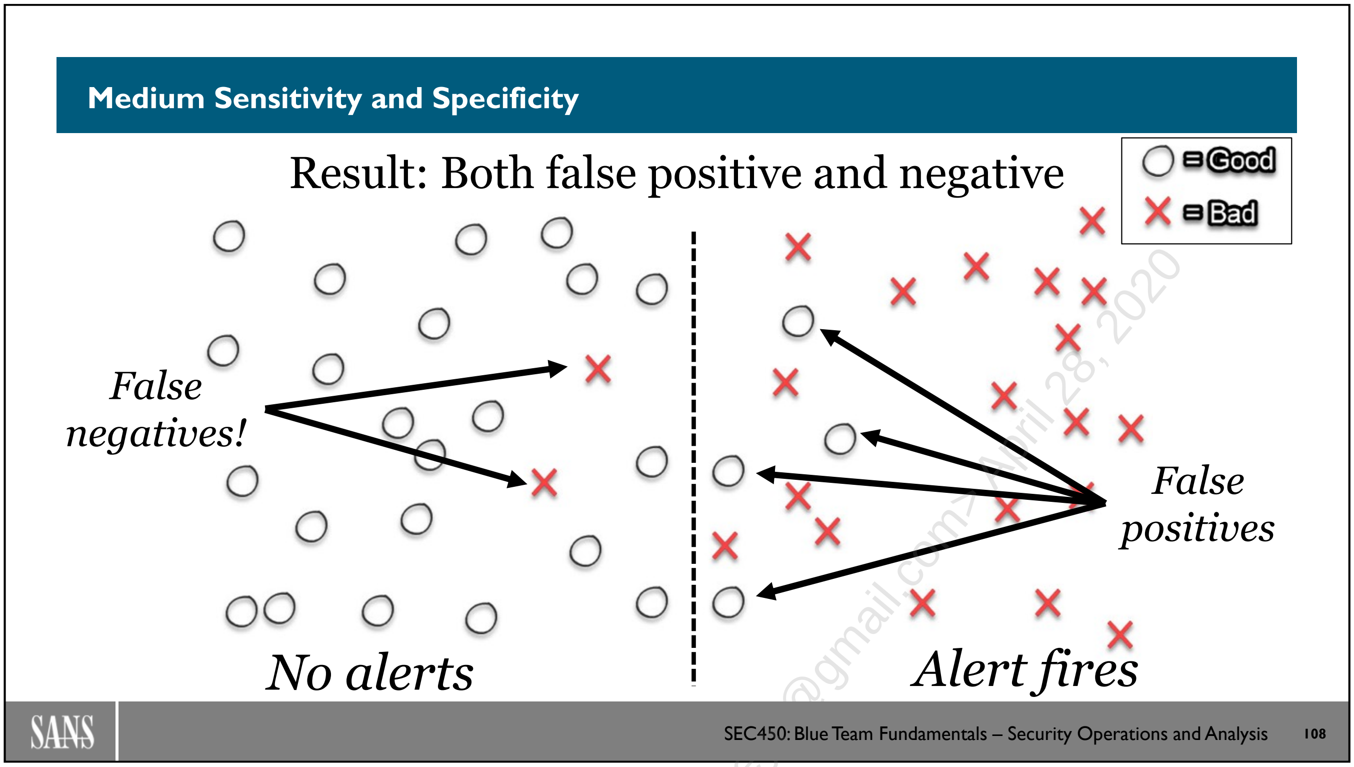
Condition Detection

All Events: Where do we draw the "line"?

○ = Good
✕ = Bad

Condition Detection

Let's use a visual to examine why getting rid of false positive and setting sensitivity and specificity is so hard. Imagine all the icons on the slide above are the population of events that happen in the environment. The circles are good events that we don't want to alert on, and the X's are events that represent attacks that we do want to alert on. If we must set our rule for alerting based on drawing a vertical line through the population of events, where should we draw it? The restriction of a vertical line represents the lack of complete knowledge we have in any given situation, and since we can't be 100% precise, we must make a less than perfect attempt at detection.

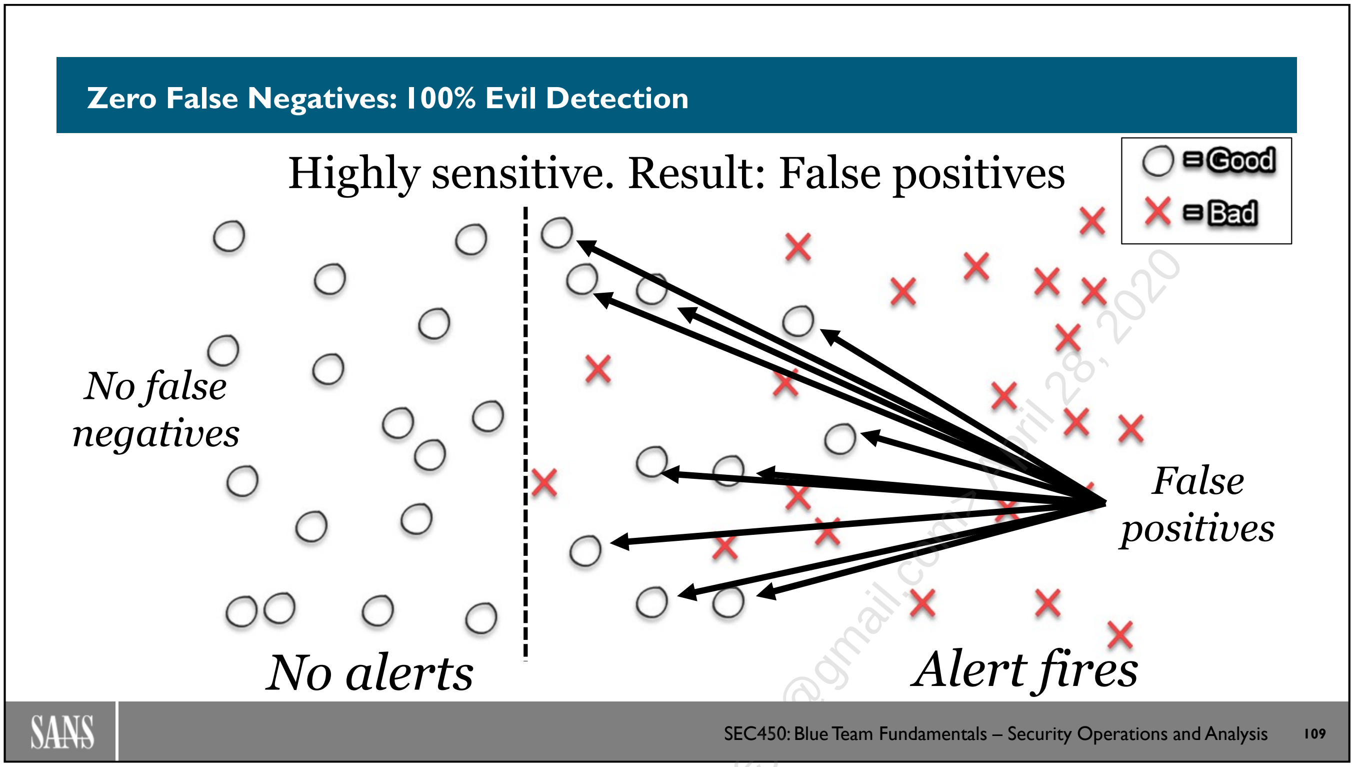


Medium Sensitivity and Specificity

Result: Both false positive and negative

Medium Sensitivity and Specificity

In this configuration, we find a middle ground between sensitivity and specificity. We are accepting both false positives and false negatives since the rule is not sensitive enough. We don't want false negatives. However, those can lead to breaches that go undetected until it's too late! What do we do then? We could move the line further to the left…



Zero False Negatives: 100% Evil Detection

Highly sensitive. Result: False positives

○ = Good
✗ = Bad

No false negatives

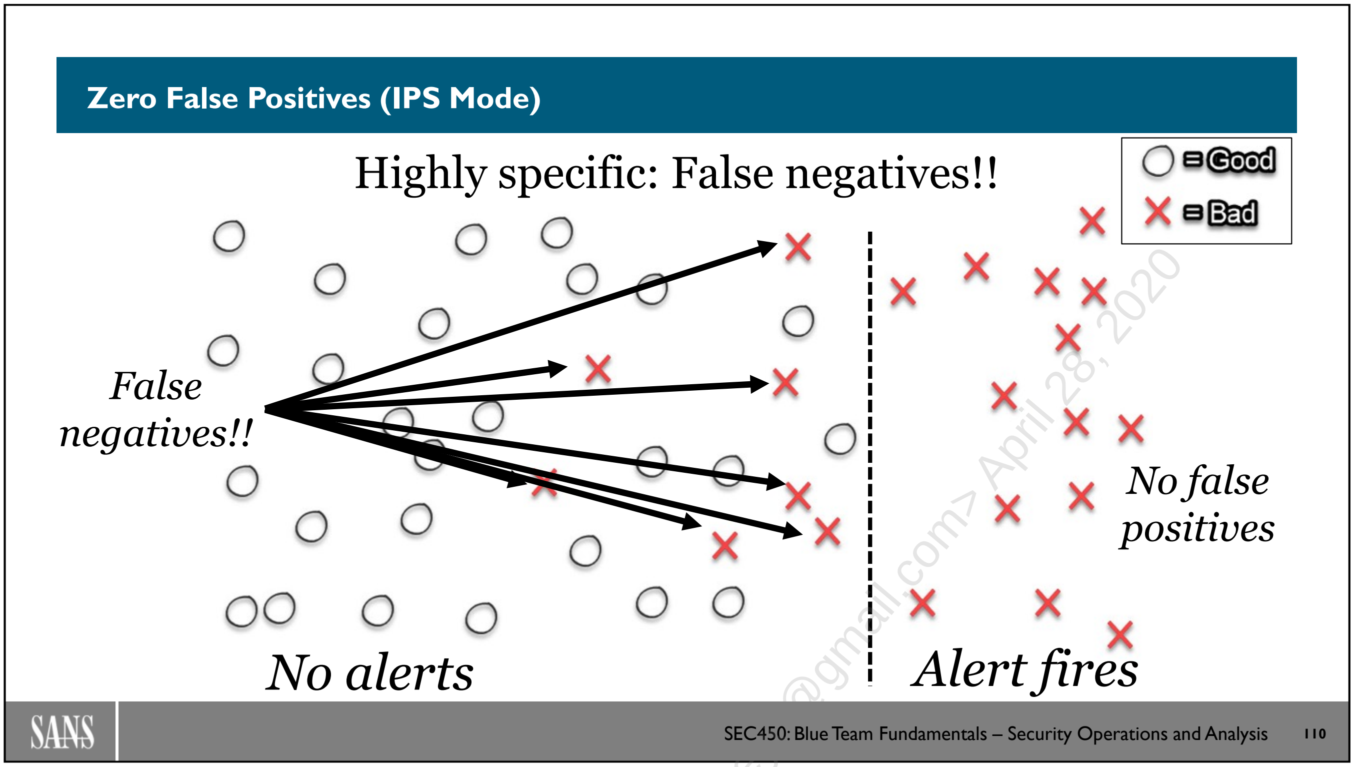
False positives

No alerts

Alert fires

Zero False Negatives: 100% Evil Detection

In this configuration, we are now catching every single evil thing that is happening—great right? Well, there is a cost associated with it. Look at what this has done to the number of false positives generated. We now have a lot more! That's because we can't move the position of the marks without changing our data set in some way; therefore, to get to complete detection, we will have to make our detection tools overly sensitive, resulting in false positives. What happens if we go the other way?

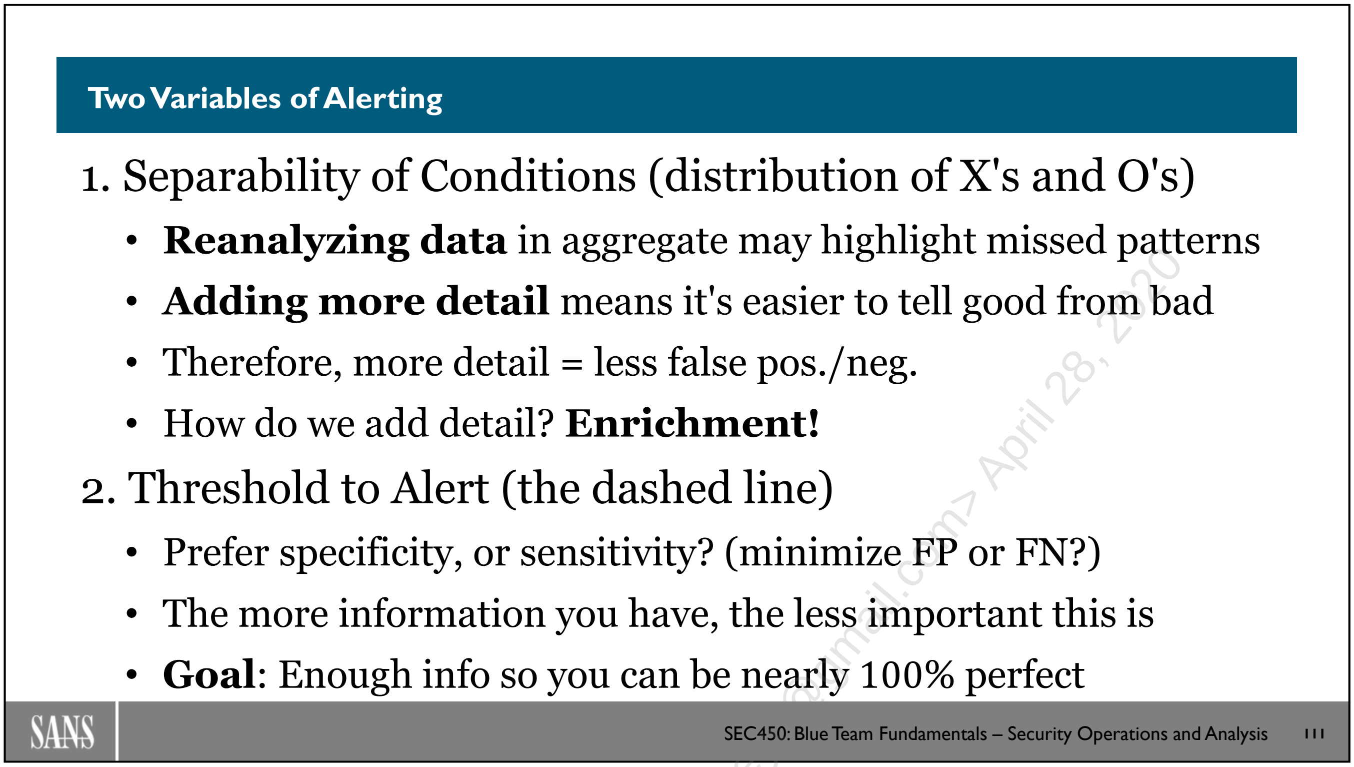


Zero False Positives (IPS Mode)

Zero False Positives (IPS Mode)

Here we can see that specificity is the inverse of sensitivity. When we tone down our sensitivity and make our rule more specific, yes, false positives will be eliminated, but now malicious events that are in the gray area will no longer be caught! This is the mythical land of "zero false positives." The problem is, to get there, without perfect knowledge, you will be forced to blind yourself from events "on the edge."

As a side note, this slide demonstrates the difference in mindset between IDS and IPS. An IPS *must* use signatures that work like this (no false positives) because if you are wrong with an IPS, you have a self-imposed denial of service. This inherently implies that you are accepting false negatives, and this is the reason we need both blocking and detection-only based signatures in an environment.



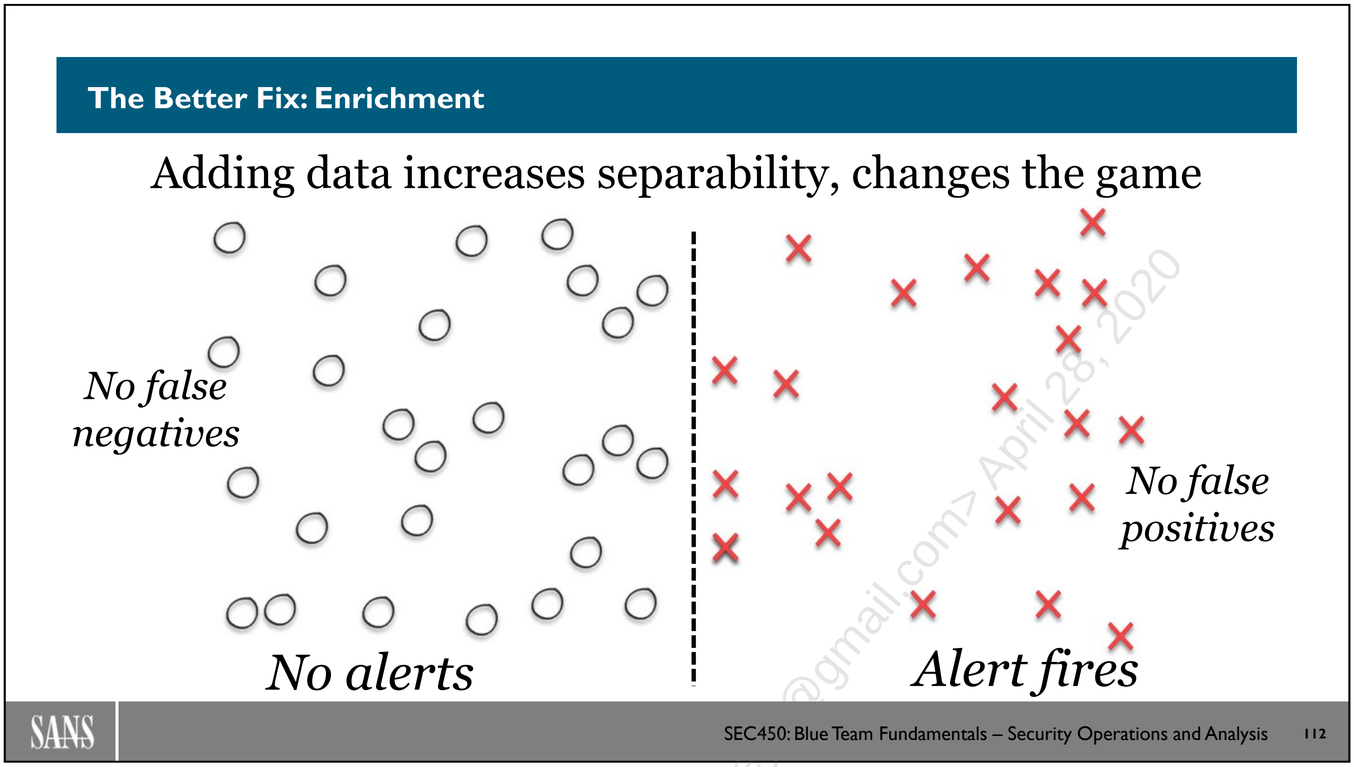
Two Variables of Alerting

1. Separability of Conditions (distribution of X's and O's)
   - **Reanalyzing data** in aggregate may highlight missed patterns
   - **Adding more detail** means it's easier to tell good from bad
   - Therefore, more detail = less false pos./neg.
   - How do we add detail? **Enrichment!**
2. Threshold to Alert (the dashed line)
   - Prefer specificity, or sensitivity? (minimize FP or FN?)
   - The more information you have, the less important this is
   - **Goal**: Enough info so you can be nearly 100% perfect

Two Variables of Alerting

How do we define the conditions that determine where the marks were placed on the previous slide vs. where the line is placed? In real data and alerting, what determines the "locations of the marks" is the amount of information you have on each event and how separable the good from the bad ones are. In the case of the example we used, this would be similar to data where some of the time you could tell good from bad, but not always, leading to the overlap area and the hard decisions we had to make about where to set the bar. There are two key methods that could add fidelity in this case. One is reanalyzing data in depth and finding the previously missed patterns. The second is adding more information to the data either from the source of the log, enrichment of the log with additional information at the SIEM, or correlation with other logs (another form of enrichment). While reanalyzing the data relies on previous mistakes and doesn't actually improve the log quality, adding more information about each even affects the separability of conditions, meaning we can likely write a higher fidelity detection.

The second variable is typically referred to as the alert threshold. It can be set arbitrarily by the user, but the output of the rule will depend on both variables, the threshold and the number of fields the data includes.



The Better Fix: Enrichment

Adding data increases separability, changes the game

No false negatives

No false positives

No alerts

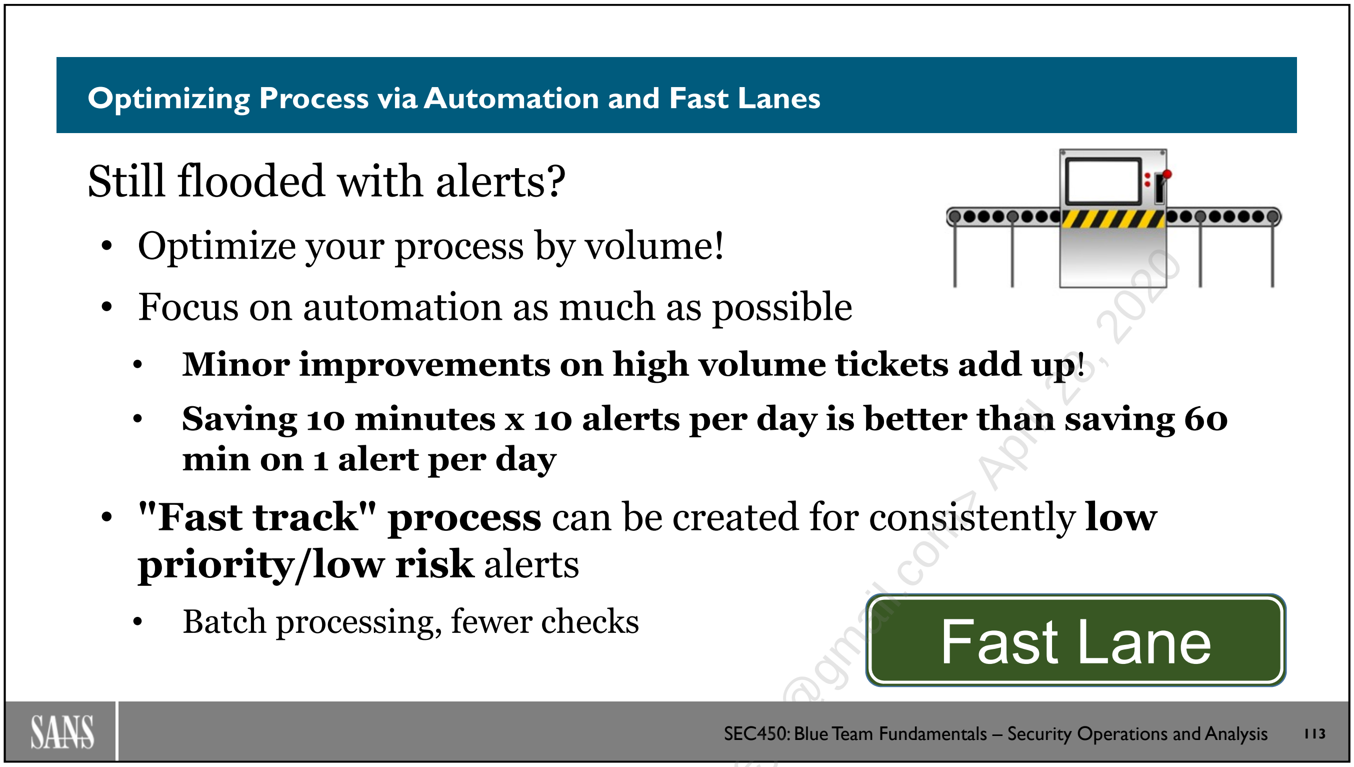
Alert fires

SEC450: Blue Team Fundamentals – Security Operations and Analysis 112

The Better Fix: Enrichment

When we add additional visibility, info, and enrichment to our logs, or use tools that are capable of correlating across multiple events, we can think of this as "rearranging" the problem space such that items can now be better separated. Consider going from a simple alert about an executable whitelist violation and the path of the file to an alert where the hash is present, has been checked against virus total, contains the reputation of the file according to your antivirus and more. With the simple alert, your only choice is to set the alert to fire on all whitelist violations, flooding you with alerts no matter if the file is a tool the user downloaded or truly a virus. It's too sensitive. By enriching the data, we can now separate the conditions of a virus vs. a benign file more accurately, and only alert when the file is new *and* has a bad reputation, or is known to be evil on VirusTotal, etc. This method increases the specificity of the alert quite a bit, but not to the point where many false negatives will appear, meaning you have eliminated almost all false positives and produced almost zero false negatives in the process by adding information.

The takeaway here is that *the solution to false positives is rule tuning and enrichment!* The more data you can use to drive the decision whether to alert or not, the better off you will be!



Optimizing Process via Automation and Fast Lanes

Still flooded with alerts?

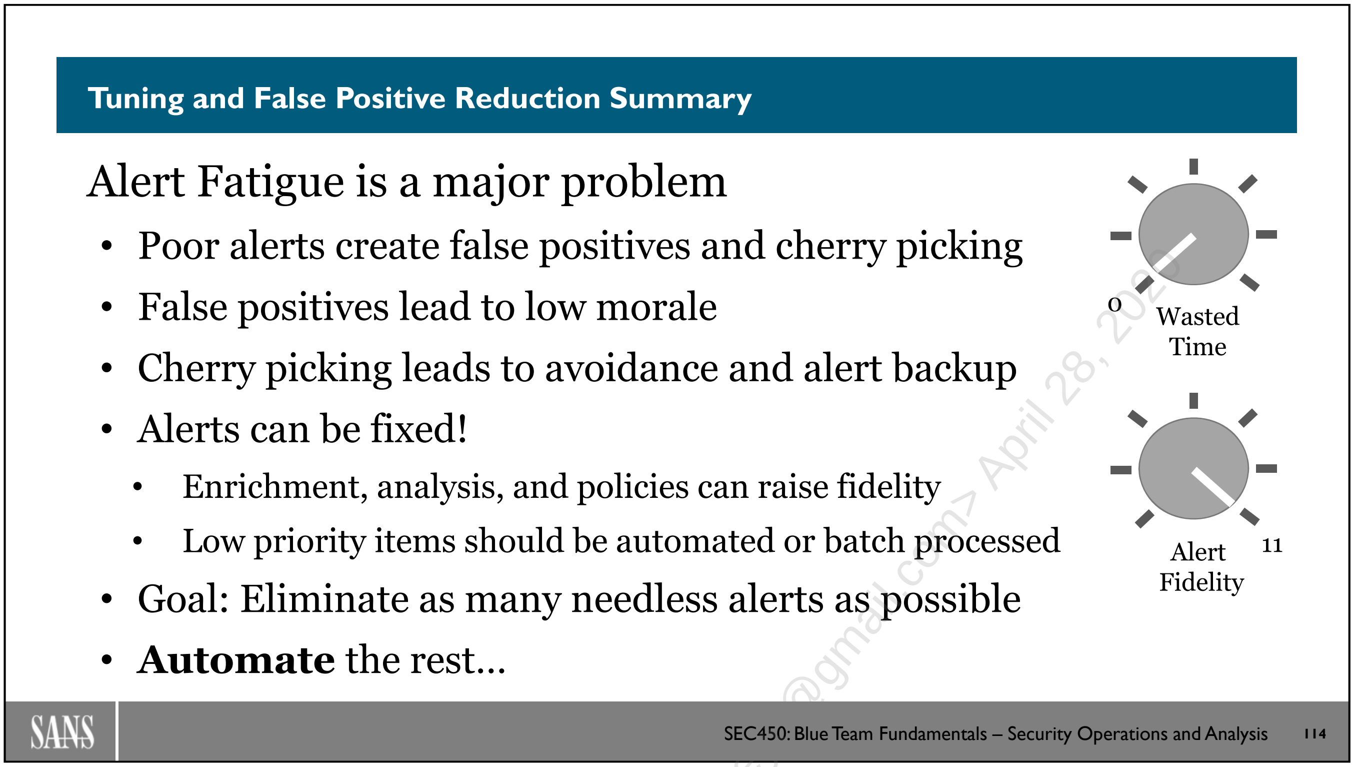
- Optimize your process by volume!
- Focus on automation as much as possible
  - **Minor improvements on high volume tickets add up!**
  - **Saving 10 minutes x 10 alerts per day is better than saving 60 min on 1 alert per day**
- **"Fast track" process** can be created for consistently **low priority/low risk** alerts
  - Batch processing, fewer checks

Optimizing Process via Automation and Fast Lanes

Perhaps you *still* have too many alerts to deal with after you have reduced as much as possible and are looking for the best way to deal with what is left. There are a few additional considerations for efficiency when deciding on what to automate and when.

First, although it may not be immediately intuitive, look at the volume of each type of alert you receive and find the highest ones. Is there are room for improvement, even if it's very small, on the process required to triage them? Although it may be tempting to automate an alert you see less often, a small improvement on a high-volume alert may be the better move. Consider an alert you get 10 times a day vs. an alert you see once a day. A 10-minute cutdown in time on the frequency alert will save you 100 minutes a day that saving even an *hour* on the infrequent once a day alert couldn't match. The best place to start considering automation is where you can make the most impact in terms of minutes saved times alerts per day.

Another option that can work in some circumstances is making a "fast lane" for tickets that have proven themselves to be consistently low priority and low risk. We can implement this in numerous ways, but its effectiveness relies on the fact that you are OK with the increased risk that comes with scrutinization of the data. Alerts that meet this criterion could either have a special, faster process they are subjected to with less rigorous investigation than an average alert or could even be batch processed together once per week if such a solution makes sense.



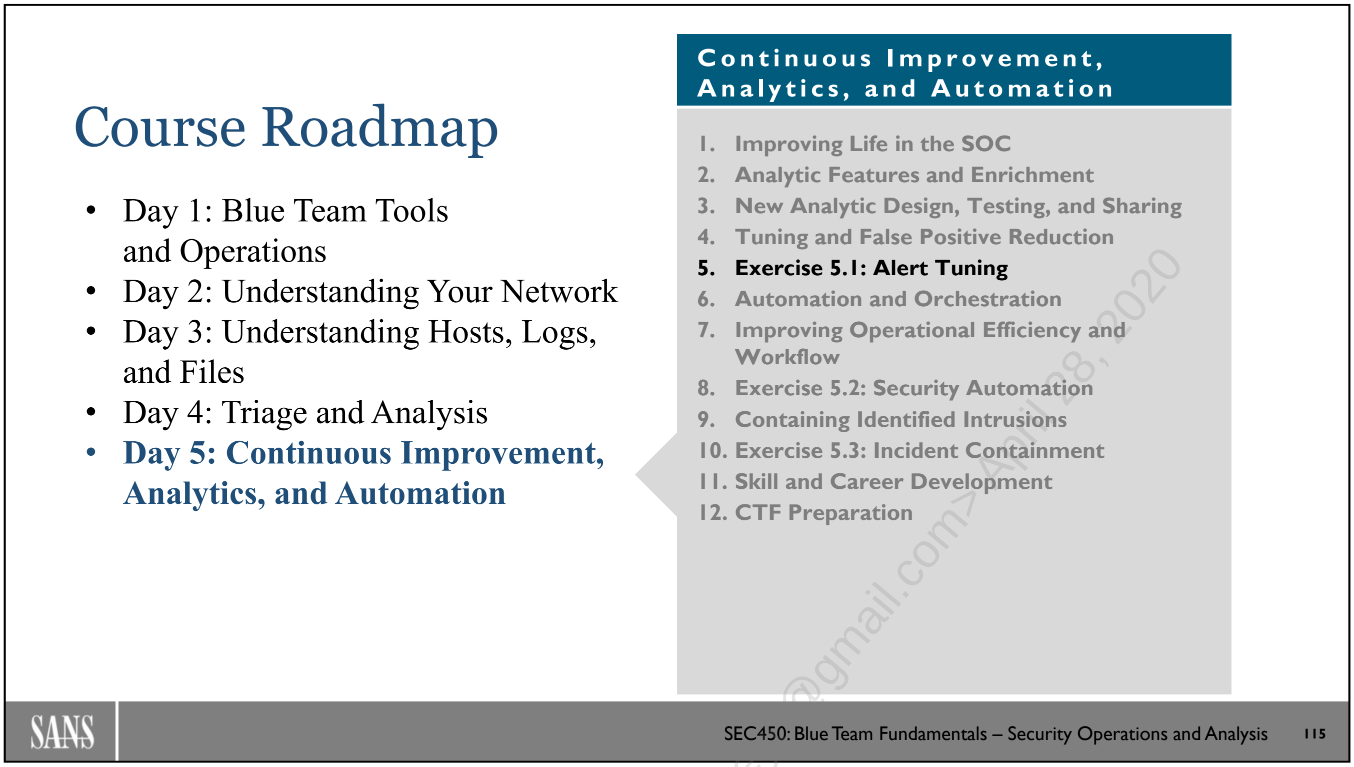
Tuning and False Positive Reduction Summary

Alert Fatigue is a major problem

- Poor alerts create false positives and cherry picking
- False positives lead to low morale
- Cherry picking leads to avoidance and alert backup
- Alerts can be fixed!
  - Enrichment, analysis, and policies can raise fidelity
  - Low priority items should be automated or batch processed
- Goal: Eliminate as many needless alerts as possible
- **Automate** the rest…

Tuning and False Positive Reduction Summary

While alert fatigue remains a major problem in many SOCs, do not fall into the trap of thinking it's something you cannot control, and you *must* control it. It will take time and effort, going through alerts logic is not easy, but you don't have to do it all at once. Start with your most frequent false positives and work your way down. As with anything, you will likely find a Pareto-like distribution. Most of the noise and false positives will be coming from a relatively few numbers of individual rules, so you can potentially make large progress with minor effort. What it comes down to is that leaving poor alerts to stay in the environment is just not an option. Find some time outside of the hours of fast-paced operation to work on it. A well-tuned set of rules pays off large dividends in team efficiency and morale.



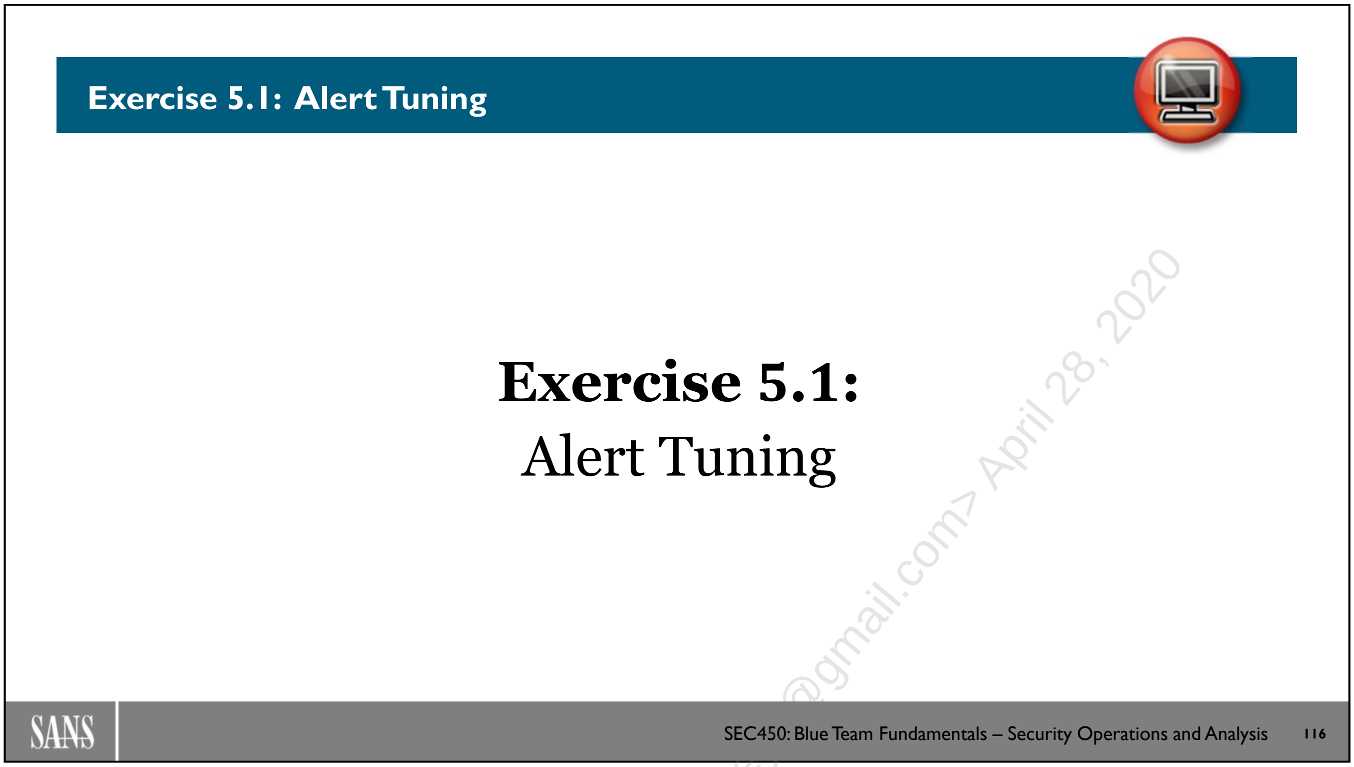
Course Roadmap

- 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**

Continuous Improvement, Analytics, and Automation

1. Improving Life in the SOC
2. Analytic Features and Enrichment
3. New Analytic Design, Testing, and Sharing
4. Tuning and False Positive Reduction
5. **Exercise 5.1: Alert Tuning**
6. Automation and Orchestration
7. Improving Operational Efficiency and Workflow
8. Exercise 5.2: Security Automation
9. Containing Identified Intrusions
10. Exercise 5.3: Incident Containment
11. Skill and Career Development
12. CTF Preparation

This page intentionally left blank.



Exercise 5.1: Alert Tuning

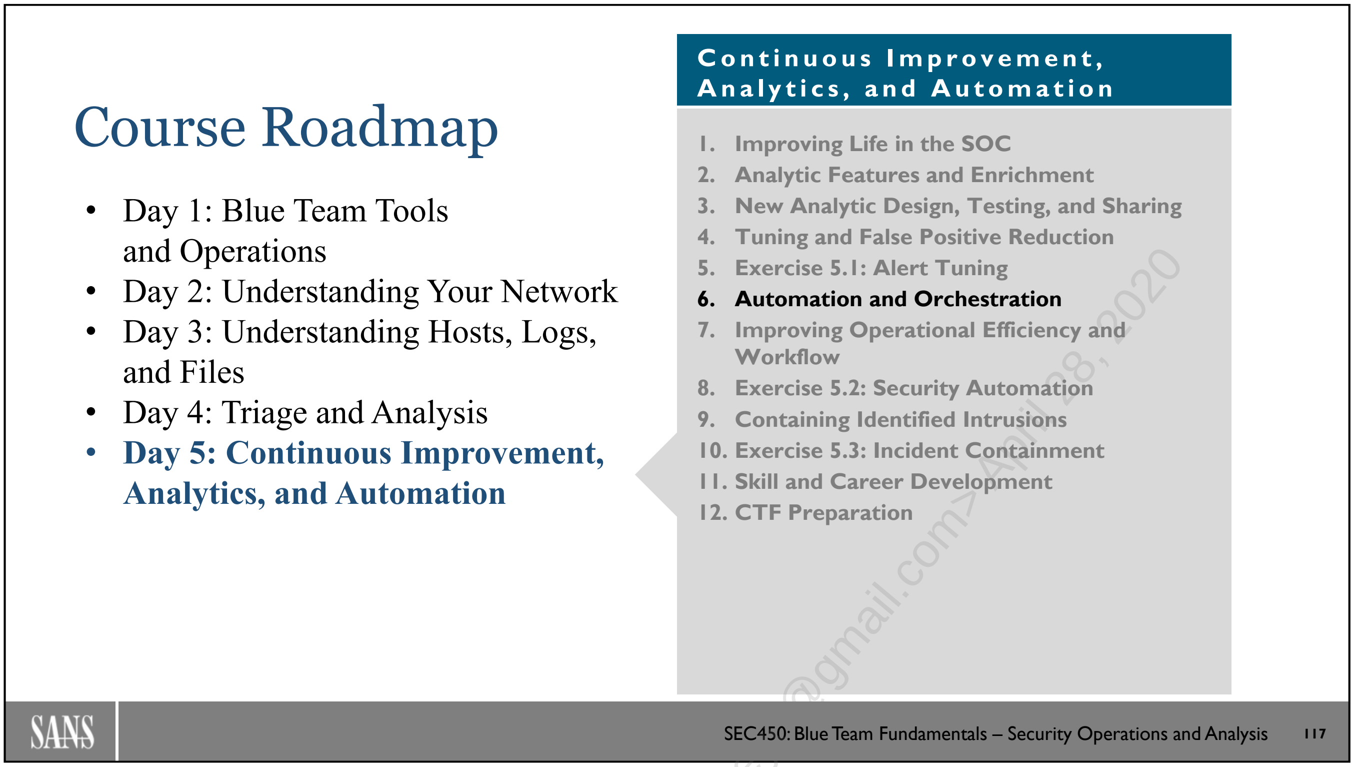
Exercise 5.1:
Alert Tuning

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 116

Exercise 5.1: Alert Tuning

Please go to Exercise 5.1 in the SEC450 Workbook or virtual wiki.



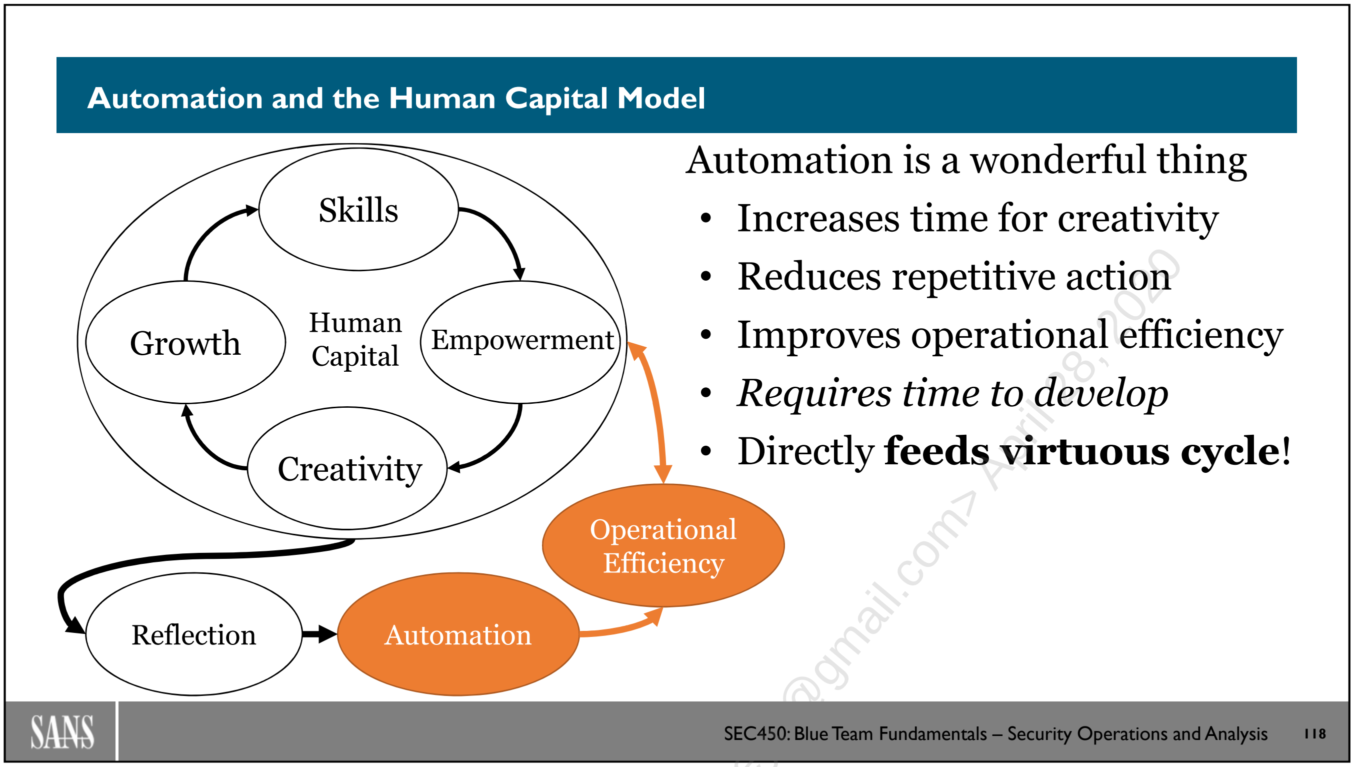
Course Roadmap

- 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**

Continuous Improvement, Analytics, and Automation

1. Improving Life in the SOC
2. Analytic Features and Enrichment
3. New Analytic Design, Testing, and Sharing
4. Tuning and False Positive Reduction
5. Exercise 5.1: Alert Tuning
6. **Automation and Orchestration**
7. Improving Operational Efficiency and Workflow
8. Exercise 5.2: Security Automation
9. Containing Identified Intrusions
10. Exercise 5.3: Incident Containment
11. Skill and Career Development
12. CTF Preparation

This page intentionally left blank.



Automation and the Human Capital Model

Human Capital

Skills

Empowerment

Creativity

Growth

Reflection

Automation

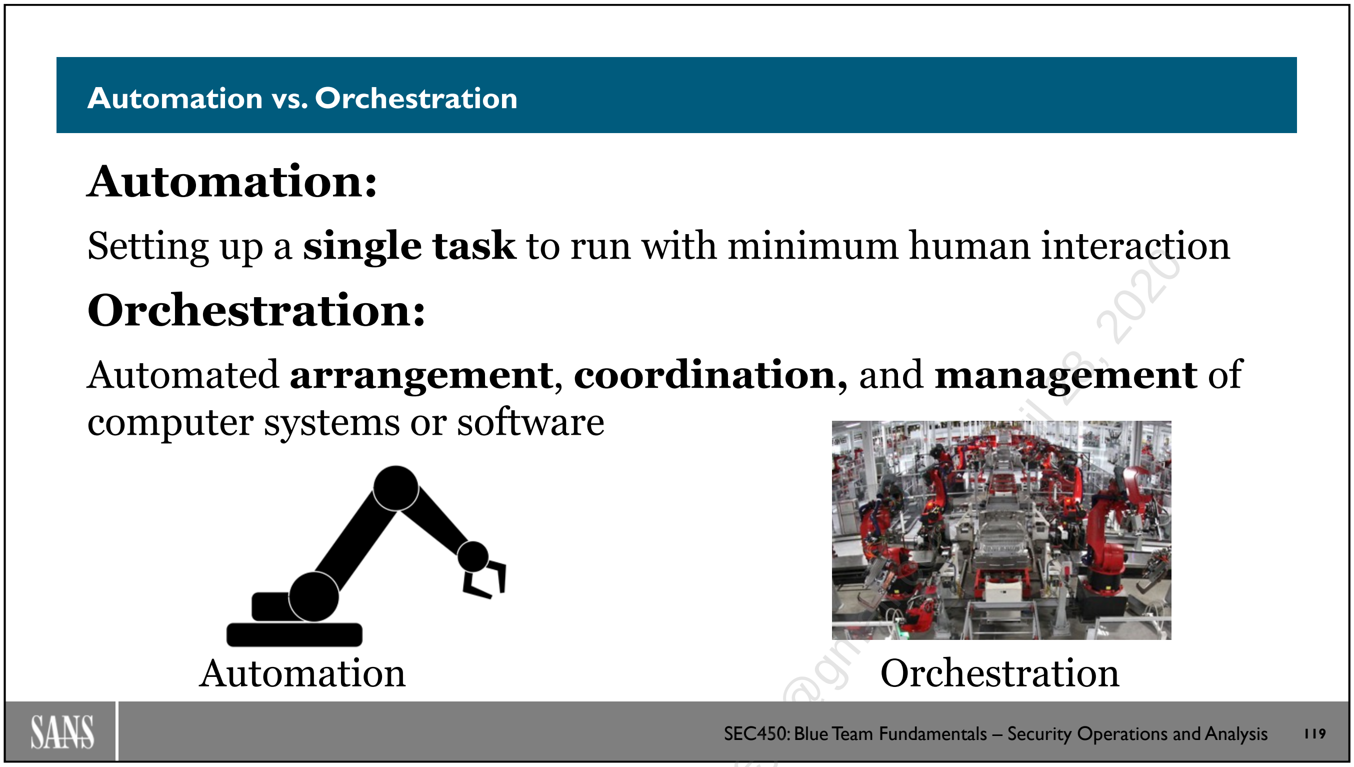
Operational Efficiency

Automation is a wonderful thing

- Increases time for creativity
- Reduces repetitive action
- Improves operational efficiency
- *Requires time to develop*
- Directly **feeds virtuous cycle**!

Automation and the Human Capital Model

It's now time to talk about automation in more detail. Remember the SOC human capital model—automation plays a key role in the feedback loop that drives operational efficiency. As the study found, a SOC that doesn't have the people or time to reflect on how processes can be automated is doomed to start down a dark path of discontent. When analysts are given time to make their jobs better, everyone wins. In this module, we'll cover multiple ways of using automation tools to ensure the continual improvement of the SOC.



Automation vs. Orchestration

Automation:

Setting up a **single task** to run with minimum human interaction

Orchestration:

Automated **arrangement**, **coordination,** and **management** of computer systems or software

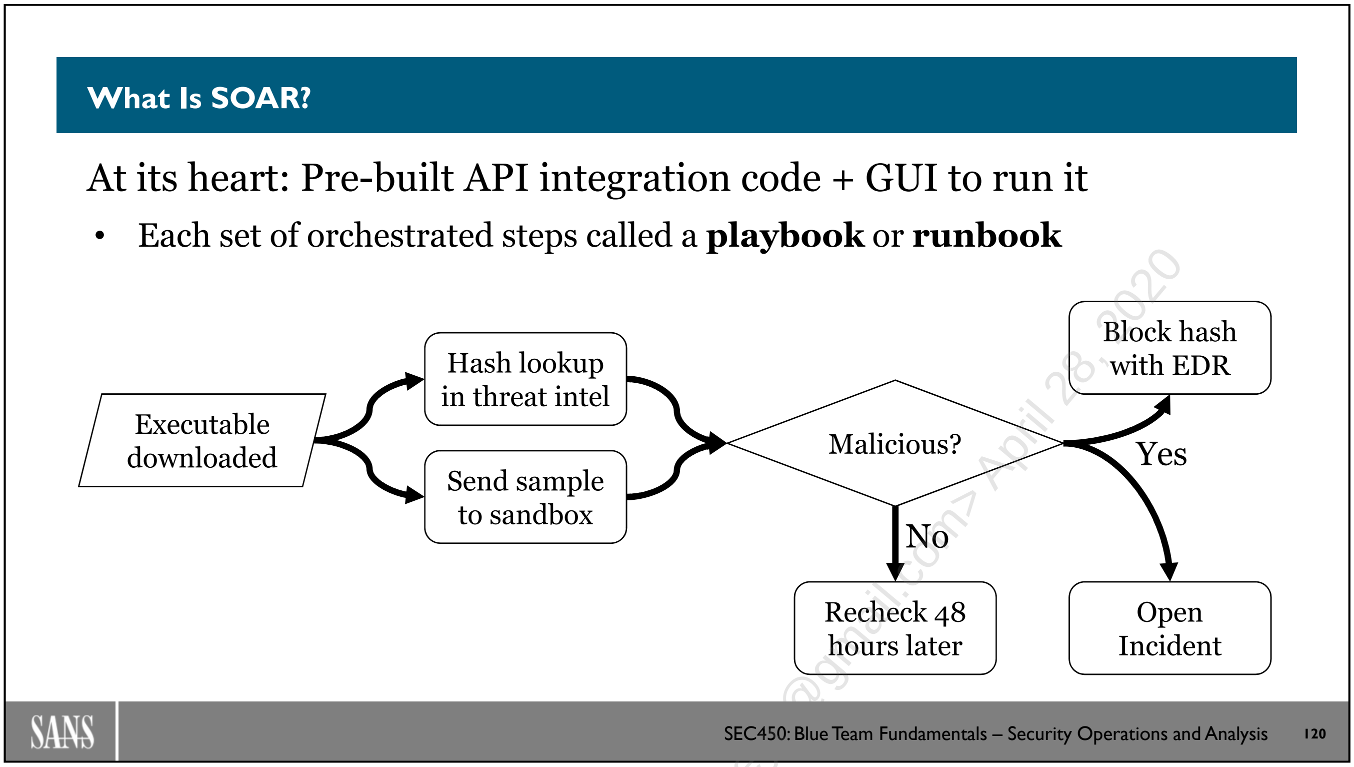
Automation

Orchestration

Automation vs. Orchestration

What is the difference between automation and orchestration? Although the two terms sound very similar, they are different things. Automation refers to taking a single task and reducing it to the minimal possible, or zero human interaction. Orchestration, on the other hand, is concerned with having multiple automated systems work together to get multiple steps of a larger process done.

For example, automation could be automating the looking up of an IP address on the abuseipdb.com database by the Cortex engine when using TheHive. Orchestration, on the other hand, might be something like pulling IP addresses that were used by confirmed malicious virus samples, inserting them into the threat intel platform with sourcing details, blocking the IPs on the firewall, then emailing the firewall team of the action taken. Let's look at how this is done, and some of the common use cases.



What Is SOAR?

At its heart: Pre-built API integration code + GUI to run it

- Each set of orchestrated steps called a **playbook** or **runbook**

Executable downloaded

Hash lookup in threat intel

Send sample to sandbox

Malicious?

Yes

Block hash with EDR

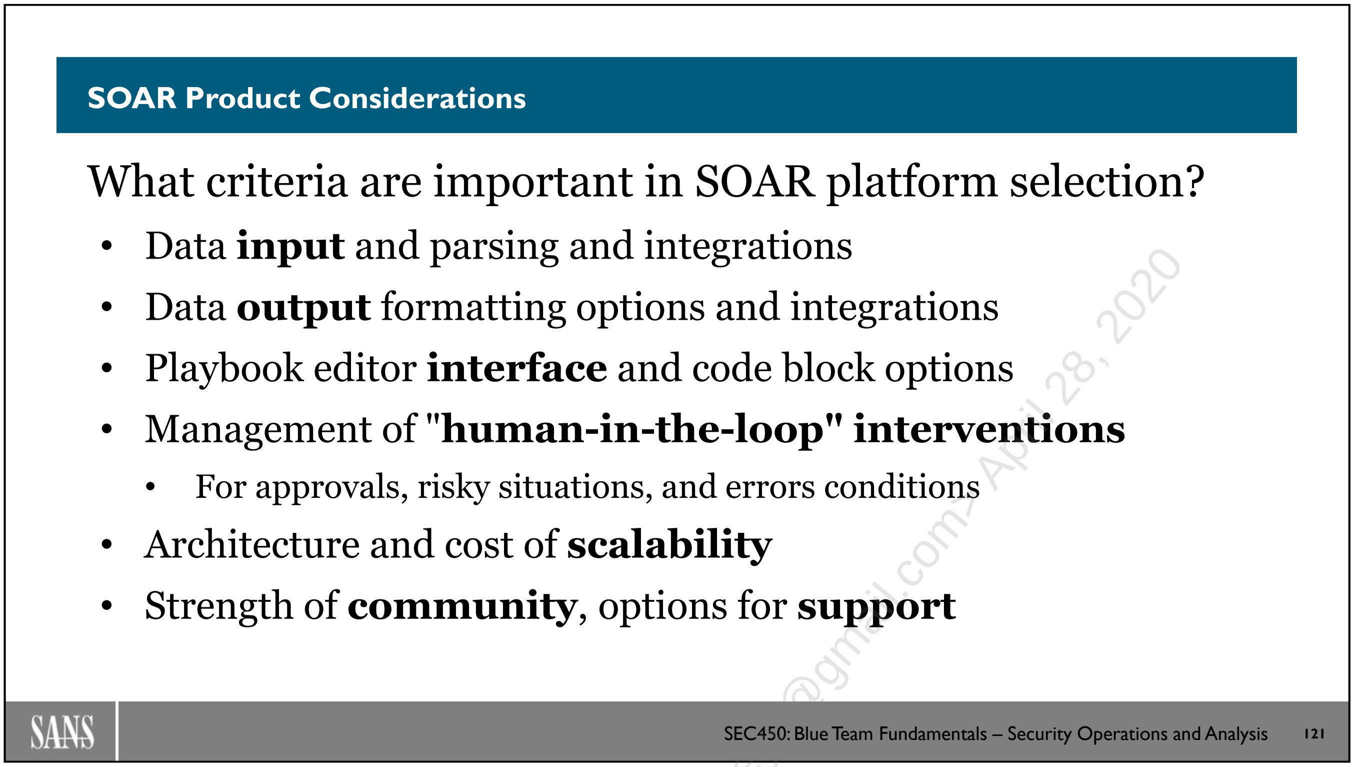
Open Incident

No

Recheck 48 hours later

What Is SOAR?

SOAR or Security Orchestration, Automation and Response is a recent product category for information security. These tools are centered around making workflows that help the SOC do a job as quickly and accurately as possible. The SOAR tool's claim to fame is "playbooks" (or "runbooks") such as the one shown on this slide. They are an *orchestrated* series of individual *automated* events that drive some investigation or response action. These playbooks keep humans out of the process when not necessary and ensure investigations and triage move along as quickly as possible. Given the natural resistance of analysts to do repetitive work, it's no wonder these tools caught on as quickly as they did. The best and most mature SOCs were likely already doing equivalent integrations through custom scripting, but the SOAR category added a nice GUI web front-end and a bunch of pre-written code to make it fast and easy for *all* SOCs to connect everything, regardless of if they had someone who could code it all from scratch.



SOAR Product Considerations

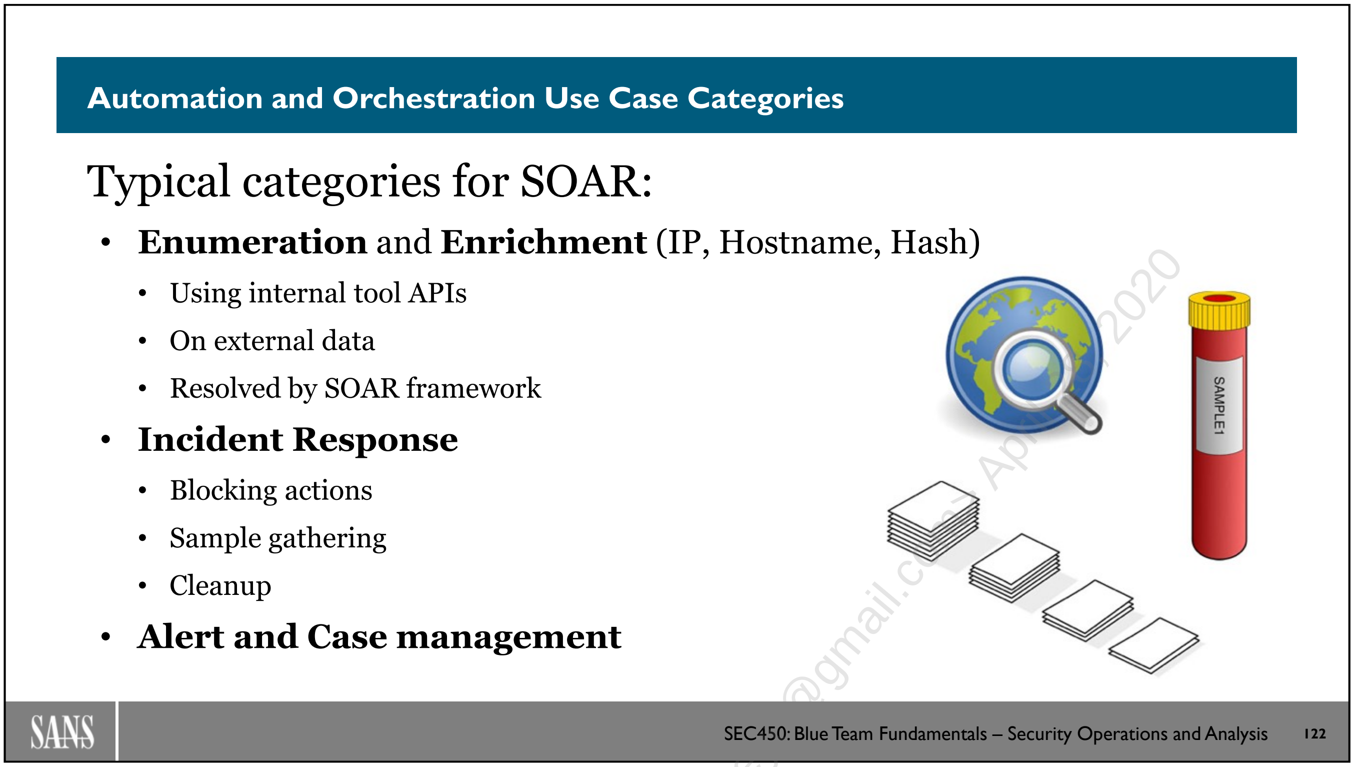
What criteria are important in SOAR platform selection?

- Data **input** and parsing and integrations
- Data **output** formatting options and integrations
- Playbook editor **interface** and code block options
- Management of **"human-in-the-loop" interventions**
  - For approvals, risky situations, and errors conditions
- Architecture and cost of **scalability**
- Strength of **community**, options for **support**

SOAR Product Considerations

What differentiates one automation and orchestration process from another? Some of the main factors you should consider when evaluating one product vs. another are listed on the slide above. Probably the most important criteria are the flexibility of input and output formats, and whether the tools you use in the SOC have pre-written code for them allowing you to get the job done. Since much of the benefit of these platforms is lowering the bar for automation, a solution that doesn't allow you to take in data from all your tools and push it back out to them in the format they need is not very useful.

Other considerations are how the actual interface works and if it is easy to diagram out a simple playbook, and how it handles supervised automation from analysts. Since not all tasks can or should be fully orchestrated, there will be workflows where prompting of an analyst will be necessary, and the way you do this should mesh with your current workflow. Also, how well does the solution scale? Constantly running multiple tasks in parallel with an ever-growing list of tasks for the orchestrator to perform can eventually max out the capabilities of a single machine, introducing the need to have more automation capacity. If you see an inevitable need for a capacity increase in the future, is it easy and affordable to do so? Finally, what is the strength of the community around the product? Well established solutions likely have a healthy third-party code base for when the vendor doesn't support a specific task. Alternatively, can you pay for support from the vendor to write you custom code modules, and if so, can you afford to do so if you cannot write them in-house?



Automation and Orchestration Use Case Categories

Typical categories for SOAR:

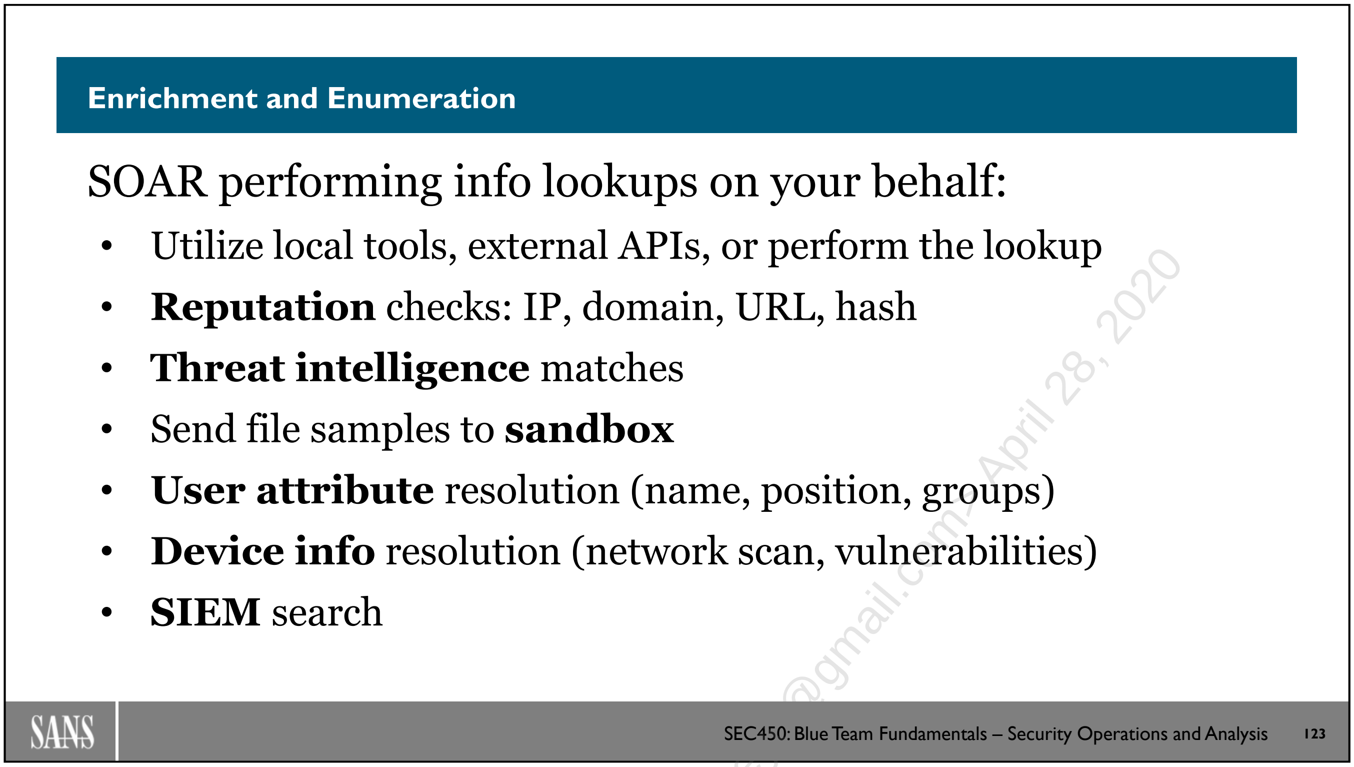
- **Enumeration** and **Enrichment** (IP, Hostname, Hash)
  - Using internal tool APIs
  - On external data
  - Resolved by SOAR framework
- **Incident Response**
  - Blocking actions
  - Sample gathering
  - Cleanup
- **Alert and Case management**

Automation and Orchestration Use Case Categories

For automation and orchestration, many of the use cases revolve around a small set of categories. One of those categories is enumeration and enrichment. In these use cases, the SOAR tool is using data already obtained to perform a lookup and pull in additional values. This could be an IP address associated with a domain, a virus check based on a hash, resolving a hostname via DNS inside the network or otherwise. Where these enrichment requests go can be broken down into lookups to internal tools and APIs, pulling information from external sites and data, and information the SOAR tool can make the request directly to resolve (such as DNS or NETBIOS lookups).

Another common category is response actions. The SOAR tool can be used to speed up ticket response and containment times by automatically grabbing malicious file samples or taking action to block hashes, domains, or IPs by interfacing with the organization's security tools directly.

The final category is alert and case management. Items in this category tend to be in the form of "moving text around and auto-submitting fields, so analysts don't have to." SOAR platforms may be used to correctly fill out observables, hostnames, notes, or other details in a ticketing system, or use that data to fill out tickets to other groups within the organization on different ticketing platforms. These actions usually revolve around moving data around instead of making analysts copy and paste it manually. In addition, some SOAR platforms contain their own dedicated alerting and case management systems! Whether or not it is acting in that capacity in your environment, the key item it should be solving is automating actions where you might ask yourself, "why do I have to do this manually?"



Enrichment and Enumeration

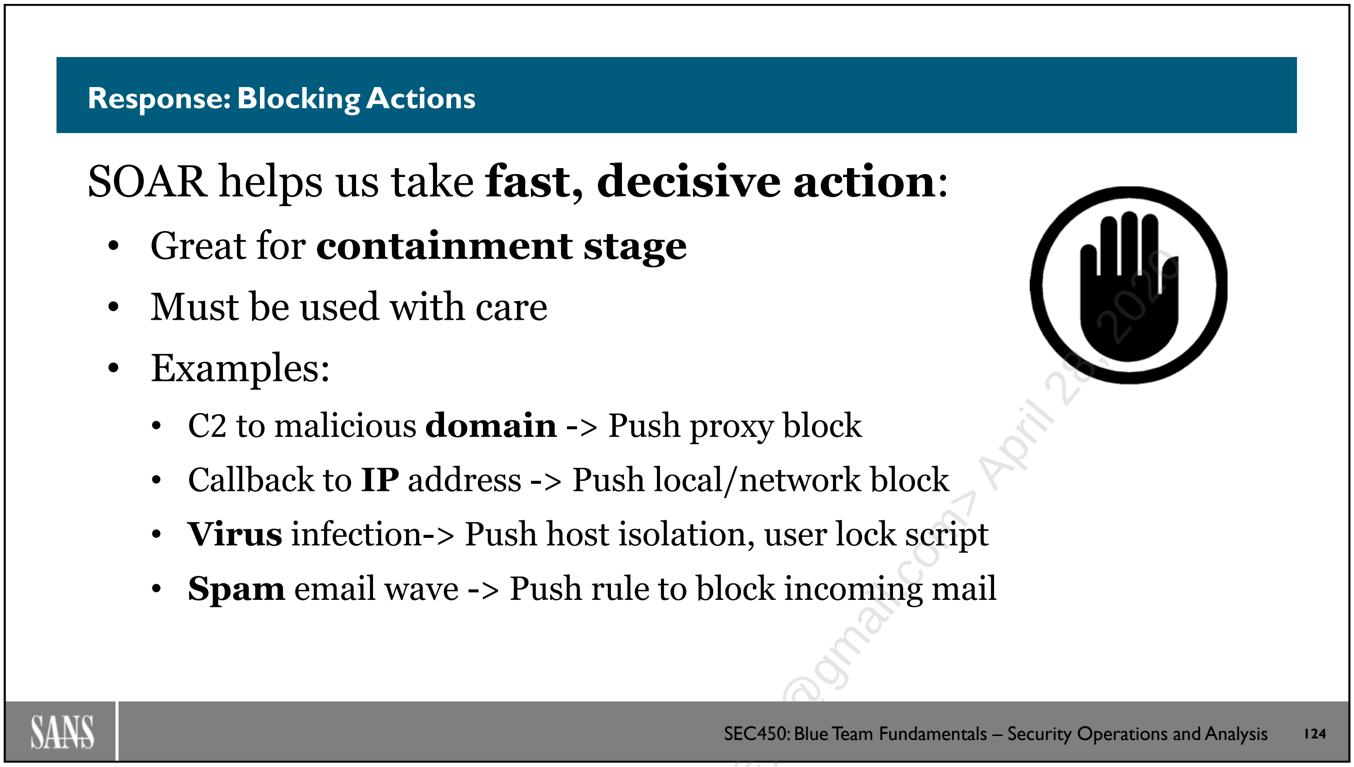
SOAR performing info lookups on your behalf:

- Utilize local tools, external APIs, or perform the lookup
- **Reputation** checks: IP, domain, URL, hash
- **Threat intelligence** matches
- Send file samples to **sandbox**
- **User attribute** resolution (name, position, groups)
- **Device info** resolution (network scan, vulnerabilities)
- **SIEM** search

Enrichment and Enumeration

SOAR is nearly always used for enrichment and enumeration purposes; it's the most obvious option. Why should you have to take an IP address and manually copy and paste it into one of your threat intel or reputation tools to lookup if you know anything about it when there's a perfectly good API available to use. In this mode, SOAR takes atomic observables of various types, parses them out of some input (likely an observable entered in an incident management system) and then can perform actions for looking up additional information on that observable in a multitude of tools. For example, a Snort IDS alert may come in saying an exploit kit was delivered from source IP 1.2.3.4, an analyst may have accepted the alert into their IMS, and the SOAR platform can grab the IP address from the system and run it through its various modules. This potentially evil IP address could be queried in *every* source of data the SOC has available to it automatically and all at once, and the results could then be placed back into the ticket, saving the analyst minutes per alert and lots of clicking. Uses for enumeration go far beyond threat data as well. You can look up user or device info, send file samples to a sandbox and pull back the verdict, or even turn around and query the SIEM for any related data.

The Cortex engine built into TheHive is one example of a tool built for enrichment automation. Once an observable is entered, an analyzer can run for multiple external services that will submit the indicator and read the output into a JSON file that can be rendered as a report. This workflow is an instance of automation (but not necessarily orchestration) built into an incident management system.



Response: Blocking Actions

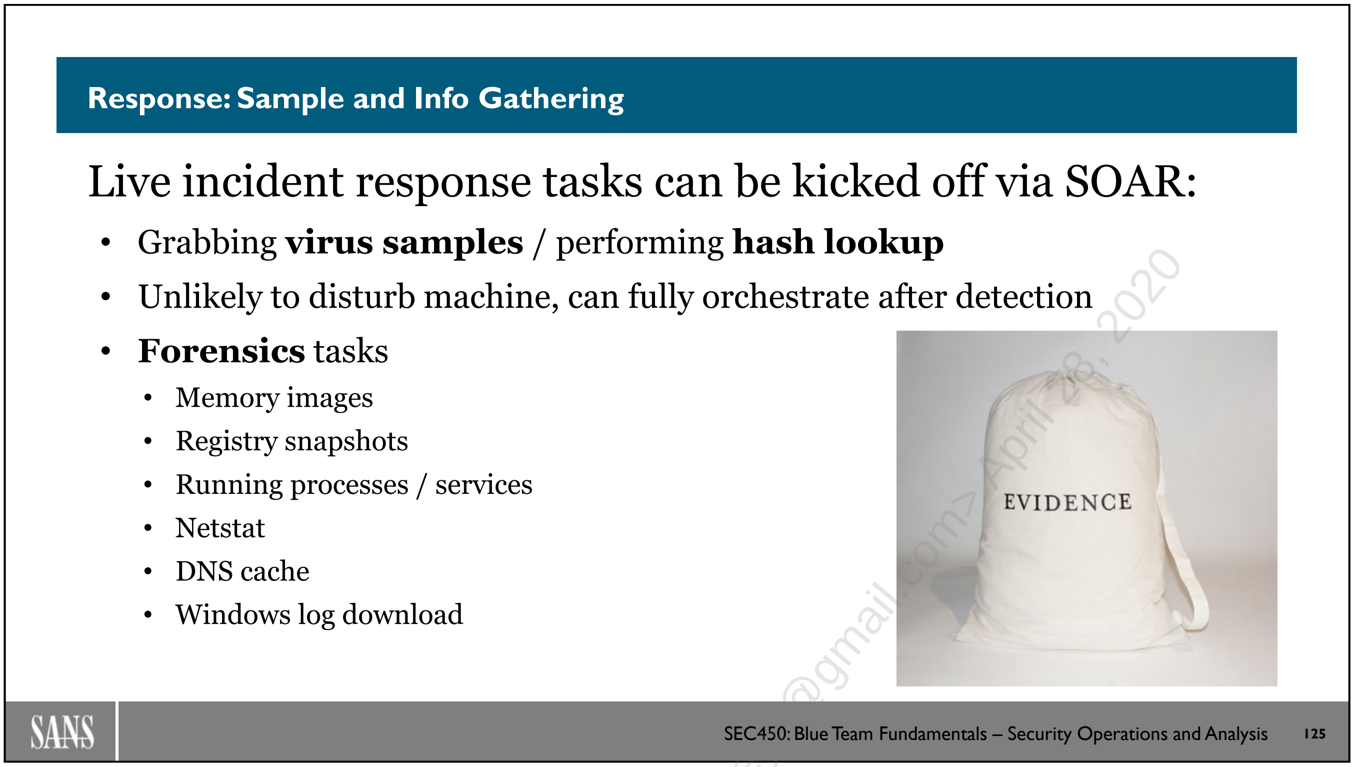
SOAR helps us take **fast, decisive action**:

- Great for **containment stage**
- Must be used with care
- Examples:
  - C2 to malicious **domain** -> Push proxy block
  - Callback to **IP** address -> Push local/network block
  - **Virus** infection-> Push host isolation, user lock script
  - **Spam** email wave -> Push rule to block incoming mail

Response: Blocking Actions

One of the best uses of SOAR is for analysts to quickly and accurately implement blocking actions. The ability to enter a domain, IP, email address, or otherwise for an automatic push to the tools to implement blocking is a great way to ensure quick responses. Blocking may or may not be directly allowed in some environments, many organizations implement careful change control request policies for things like this, and that's not necessarily a bad idea for uptime. While one could argue from a security perspective that it is crucial to be timely on these matters, the business may push back saying an accidental block on a business-critical tool could be very costly, and they aren't wrong about that either.

Every organization will need to come to their own conclusions about whether the SOC team should have the ability to implement observable blocks directly or not, but in the interest of response speed, gaining the trust to be delegated with a power like this is ideal. "Four-eyes" rules mandating multiple people must agree that a block won't cause a problem before the button is hit, and other SOC-based policies can help keep this power while preventing mistakes. Additionally, ensure all blocks implemented this way are traceable back to the user that pushed them, and the incident associated with the block. As someone who has been on a team with this power, traceability is key to maintaining the block list in the long term as well as looking back into blocking history. Analysts exercising this power should place a comment associated with each blocking action explaining their reasoning for doing so at the time.



Response: Sample and Info Gathering

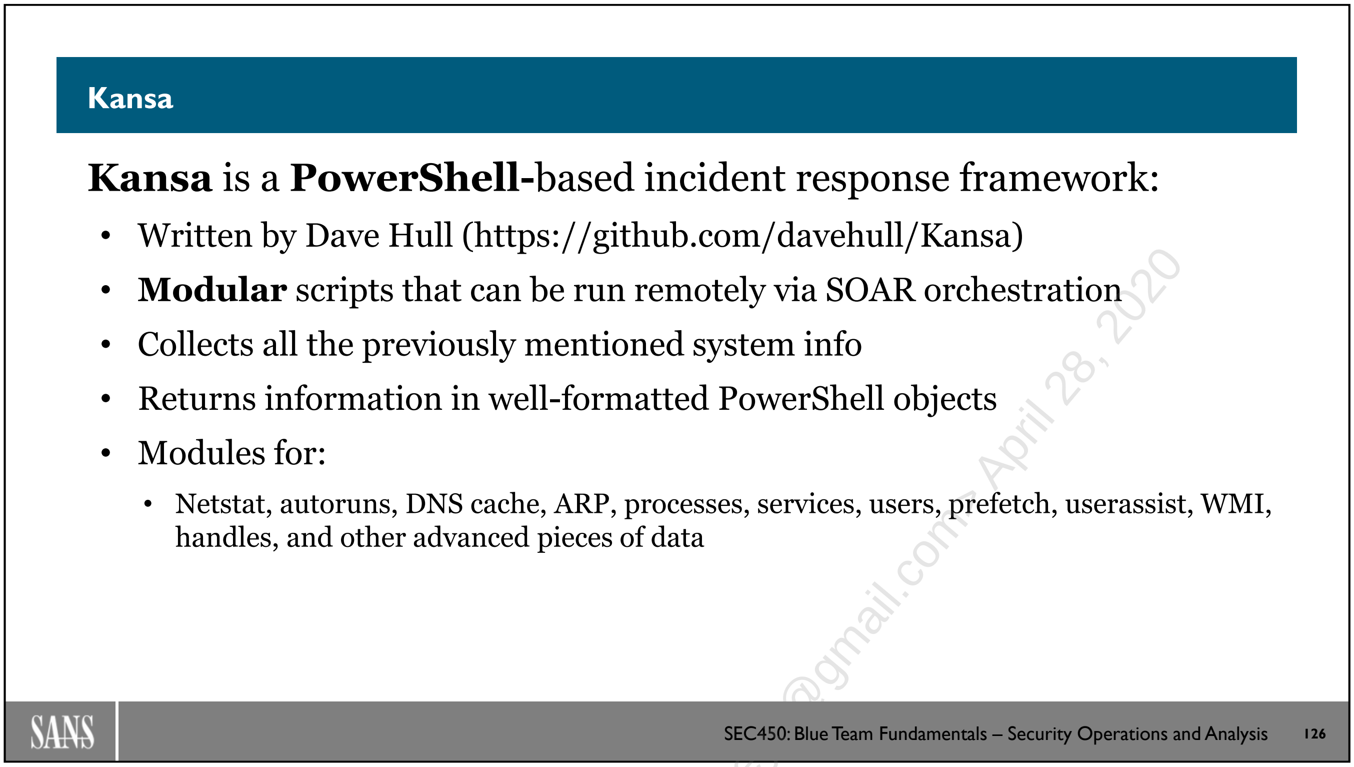
Live incident response tasks can be kicked off via SOAR:

- Grabbing **virus samples** / performing **hash lookup**
- Unlikely to disturb machine, can fully orchestrate after detection
- **Forensics** tasks
  - Memory images
  - Registry snapshots
  - Running processes / services
  - Netstat
  - DNS cache
  - Windows log download

Response: Sample and Info Gathering

Another good use of SOAR platforms is doing live incident response tasks such as sample and host information gathering. Information on an infected machine can be extremely volatile, so when something suspicious happens, quickly grabbing the state of the infected machine can make the difference between immediately understanding the problem and having to reverse engineer malware and search through PCAPs and logs to understand what happened.

The SOAR engine can be used to deploy custom scripts or enact EDR or other tools already present on the device to gather crucial information and send it back to the SOC. Things like autoruns, netstat entries, DNS cache, open files, running processes, memory images, virus samples, and other information can make the incident investigation go very quickly, but only if available and taken in a timely fashion. Better yet, since taking data and virus samples is unlikely to disturb the machine, this is a process that can likely be orchestrated to occur after any virus detection with minimal chance of unintended consequences.



Kansa

Kansa is a **PowerShell-**based incident response framework:

- Written by Dave Hull (https://github.com/davehull/Kansa)
- **Modular** scripts that can be run remotely via SOAR orchestration
- Collects all the previously mentioned system info
- Returns information in well-formatted PowerShell objects
- Modules for:
  - Netstat, autoruns, DNS cache, ARP, processes, services, users, prefetch, userassist, WMI, handles, and other advanced pieces of data

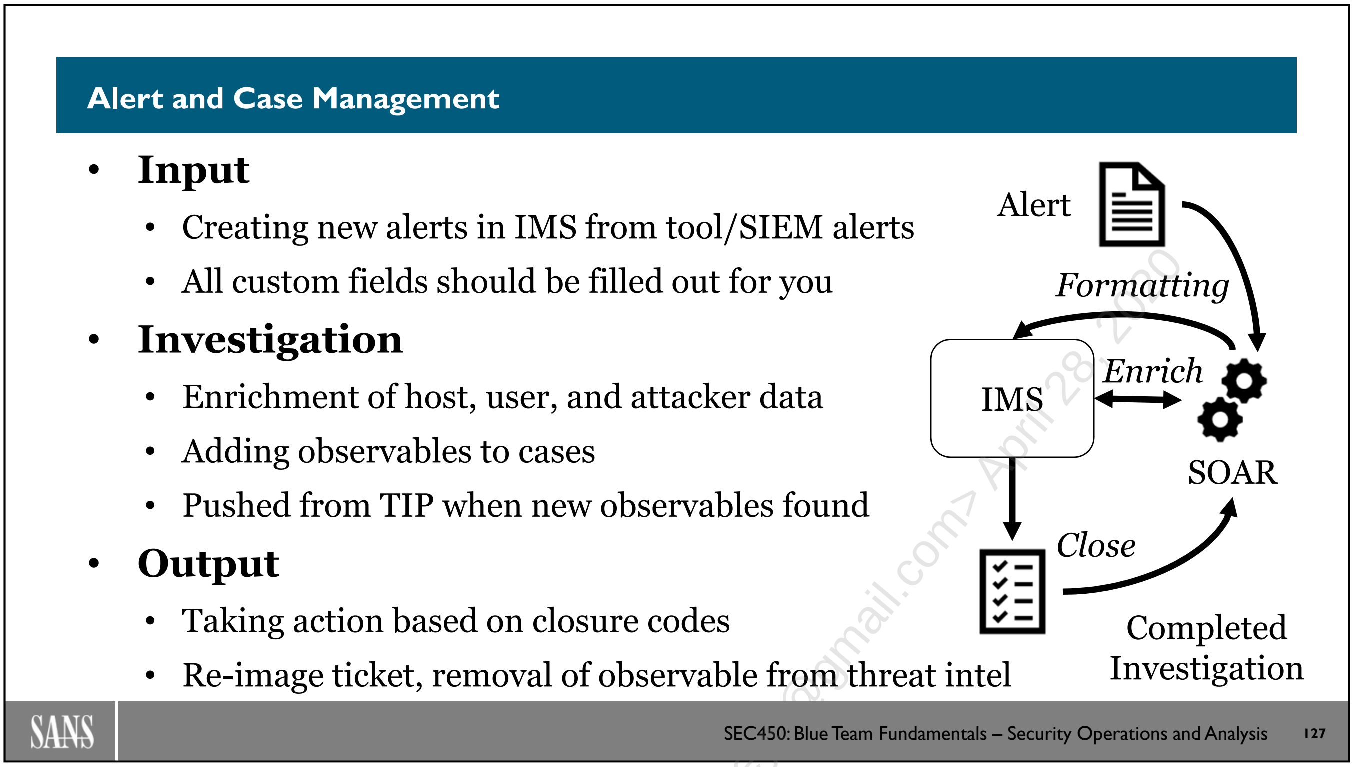
Kansa

If your SOAR framework doesn't have the capabilities to grab the samples or data that you're looking for, there are plenty of other projects out there that can be used instead. One of the most commonly referenced tools for data gathering for live incident response is Dave Hull's **Kansa** framework.[1] This is a live incident response data-gathering tool built completely out of PowerShell scripts that can be separately deployed by a SOAR engine to the device in question, pulling back the output for evidence. It even supports JSON output that could be pulled directly into your SIEM or ticketing solution with ease! There is no need to reinvent the wheel on many of these actions; your SOAR tool should simply be giving you a platform to easy chain a bunch of automated events together and to take action based on their output.

Although Kansa is a great project, it does require that PowerShell remoting access is available for the remote machine. If you cannot do PowerShell remoting, the **CIMSweep** project[2] is an alternative that uses Windows WMI commands instead.

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

[2] https://github.com/PowerShellMafia/CimSweep



Alert and Case Management

- **Input**
  - Creating new alerts in IMS from tool/SIEM alerts
  - All custom fields should be filled out for you
- **Investigation**
  - Enrichment of host, user, and attacker data
  - Adding observables to cases
  - Pushed from TIP when new observables found
- **Output**
  - Taking action based on closure codes
  - Re-image ticket, removal of observable from threat intel

Alert
Formatting
IMS
Enrich
SOAR
Close
Completed Investigation

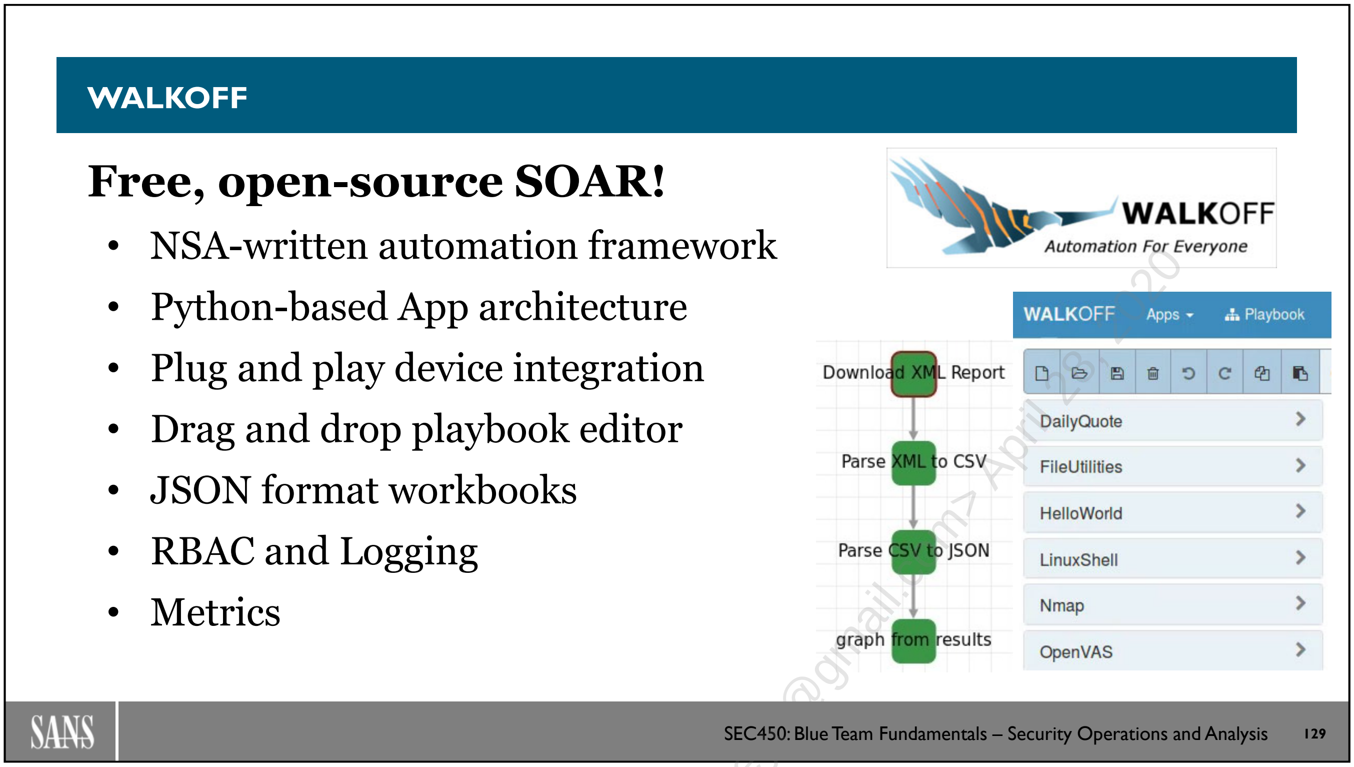
Alert and Case Management

SOAR can be used in a variety of ways to make interactions with your incident management system easier. Although exactly how you do it will depend on the specifics of the tool you use, there are a few tasks that apply to almost all incident management systems that we can discuss as general use cases. These use cases can be broken down into items that happen at the time of input, actions taken during investigations, and then things that happen during output.

For the input section, one of the items that should be considered a requirement is that any alert sent to the IMS for investigation should have all the data cleanly imported into all the correct fields. Alert logs may come from the SIEM in a variety of forms, syslog, CSV, key-value pairs and JSON, and all of them contain fields that need to be filled out in the incident management system for tracking. This is a perfect use case for automation because parsing fields and sending them in a well-formatted way between tools is where SOAR specializes. The ideal is that once the analysts switch to investigating a case in the IMS, all the user info, source, and destination IP addresses, and other asset information are not just in a lump of text in a description box but are filled out into all the custom fields the system likely has designated for them.

As part of the investigation, process analysts should be able to take additional observables and push them to the SOAR tool from the IMS for enrichment. Cortex analyzers in TheHive are a good example of this. If a new command and control domain is found, the analyst enters it as an observable in TheHive. From there, Cortex analyzers can be kicked off to look up additional contextual data from a variety of sources directly from inside the case. Any data the analysts will commonly access in this fashion should ideally be made available from inside the alert management system through automated means. In addition to enrichment, since incident management tools themselves are often playbook driven, any steps that the SOAR can undertake on behalf of the analyst can also help speed things up, especially if they are not dependent on other steps preceding it. If a playbook in the incident management system has three parallel paths of steps and two of them can be fully automated, the automatable ones should be scripted while the analysts run down the final investigation path manually.

On the output side of the system, a third option should be driving activity based on the closure code of an incident. If an investigation is found to be a true positive and a host must be re-imaged, having the SOAR platform extract the information and send the request to the help desk is an example of one way SOAR can help. In the case of a false positive, bad indicators could be tagged that way so in future alerts they will be seen as a lower fidelity indicator from the start. If there are enough closed alerts that turn out to be a false positive from the same domain/hash/IP, the SOAR tool could pick up on this and alert the team to take them out of the blacklist, so no further alerts are made.



WALKOFF

Free, open-source SOAR!

- NSA-written automation framework
- Python-based App architecture
- Plug and play device integration
- Drag and drop playbook editor
- JSON format workbooks
- RBAC and Logging
- Metrics

WALKOFF

Although there are not yet many free and open-source SOAR platforms, there is one called **WALKOFF** that has been released by the NSA.[1] [2] As with all automation and orchestration frameworks, it contains a GUI playbook editor, a collection of pre-made "apps" that can be used to create processes, and an easy python-based architecture that allows you to extend it in any way needed. Though it is still early in development and many apps are still being developed, the project shows promise and already works well with a variety of common tools. This class will use the WALKOFF tool in labs to demonstrate the basic process of creating an orchestrated workflow based on a series of atomic steps. Although it may not be the same interface as the SOAR tool you are using, the overall experience and workflow of using pre-made actions to link together with logical conditions mirror the usage of any SOAR app, regardless of vendor.

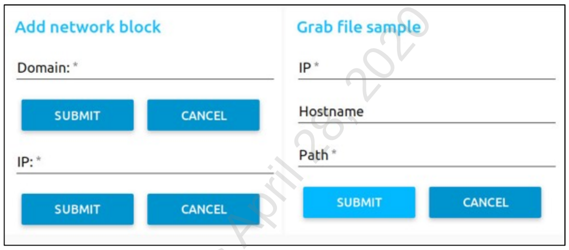
[1] https://github.com/nsacyber/WALKOFF

[2] https://nsacyber.github.io/WALKOFF/

IBM Node-RED

A flow-based programming and automation framework

- IOT focused, but *very* capable
- Easy to stand up with Docker
- Passes info as JSON data
- Many third-party modules



Add network block

Domain: *

SUBMIT CANCEL

IP: *

SUBMIT CANCEL

Grab file sample

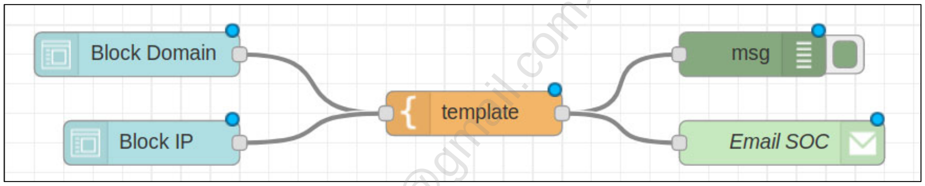
IP *

Hostname

Path *

SUBMIT CANCEL

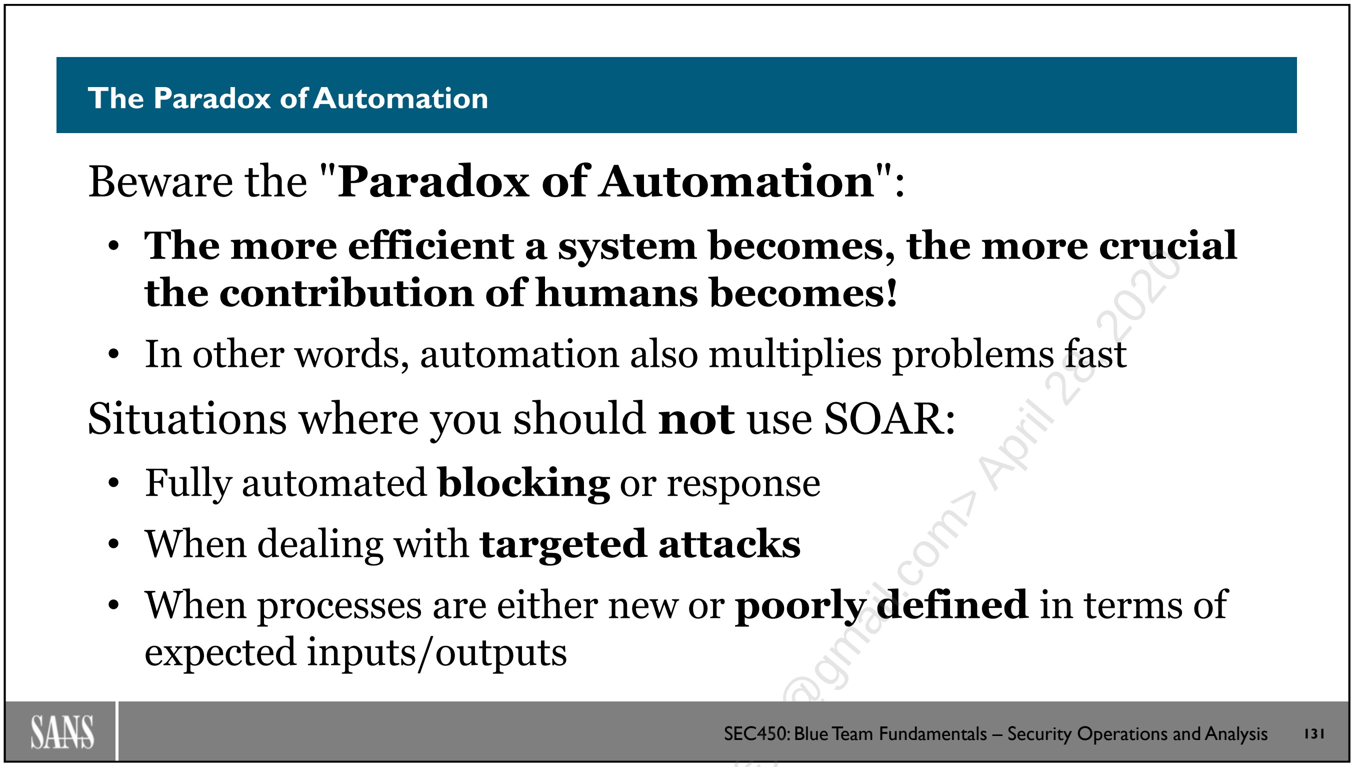
Node-RED



IBM Node-RED

Another free software option that works surprisingly well for SOAR tasks is IBM's Node Red.[1] IBM describes Node Red as "a programming tool for wiring together hardware devices, APIs, and online services in new and interesting ways." While not security focused, it has many built-in and third-party modules that can accomplish most of the tasks a SOC would need including HTTP API access, email, and SSH/SFTP. Even better is that it's extremely easy to try out, and with the provided Dockerized version, it's a single command in a Linux environment to have it up and running!

[1] https://nodered.org/



The Paradox of Automation

Beware the "**Paradox of Automation**":

- **The more efficient a system becomes, the more crucial the contribution of humans becomes!**
- In other words, automation also multiplies problems fast

Situations where you should **not** use SOAR:

- Fully automated **blocking** or response
- When dealing with **targeted attacks**
- When processes are either new or **poorly defined** in terms of expected inputs/outputs

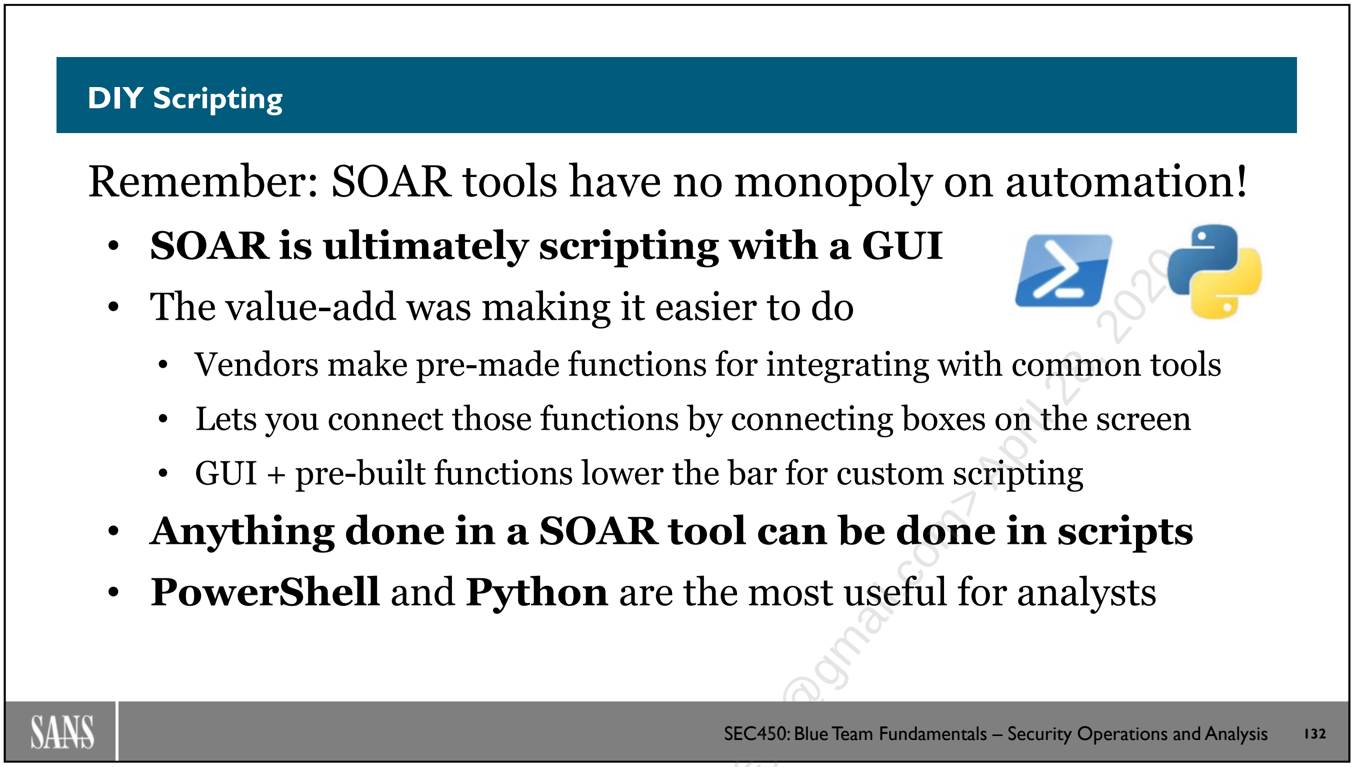
When NOT to Use Orchestration, or, "How to Automate a Self-DoS"

Although we have extolled the virtues of automation throughout this section, beware that as with many improvements, there is an associated cost. In this case, the cost comes in the form of the "paradox of automation."[1] This refers to the fact that the more a process becomes automated, the more important the human overseeing it becomes. This statement is not saying that the human will become *more* involved, merely that *when* they must become involved, it is likely because they are either averting or near a disaster.

Automation systems are fast and efficient at what they do, which is great when they're doing the right thing, but what if some unanticipated scenario comes up giving them the wrong input? Imagine a robot on a production line systematically ruining each product going through it due to an error in programming; this will continue until it is found, causing a massive expense. In these types of scenarios, highly orchestrated steps of events can run wild causing chaos at the output. Alternatively, they may continue to do something wrong over a period of time without much of a hint that anything is wrong until someone notices hours or days later, leaving a big mess to clean up. It is this double-edged sword we must be aware of when programming our orchestration playbooks, extreme attention to detail is necessary to ensure we have thought up all possible invalid inputs and other scenarios that may cause unanticipated behavior.

Given this issue, there are a few scenarios where you would be best advised to go light or forgo automation. One is in the response category for blocking malicious pages. If you created an action that automatically blocks a site the IDS detects an alert from, what happens that day where the IDS creates a false positive on Google or a critical business application? The same thing goes for targeted attacks. While automated remediation and research are great, it should likely still have a human in the loop to decide when it is and is not appropriate to use. Having automated cleanup or external data lookups going to adversary infrastructure may be enough to compromise your defense efforts.

[1] https://personalmba.com/paradox-of-automation/



DIY Scripting

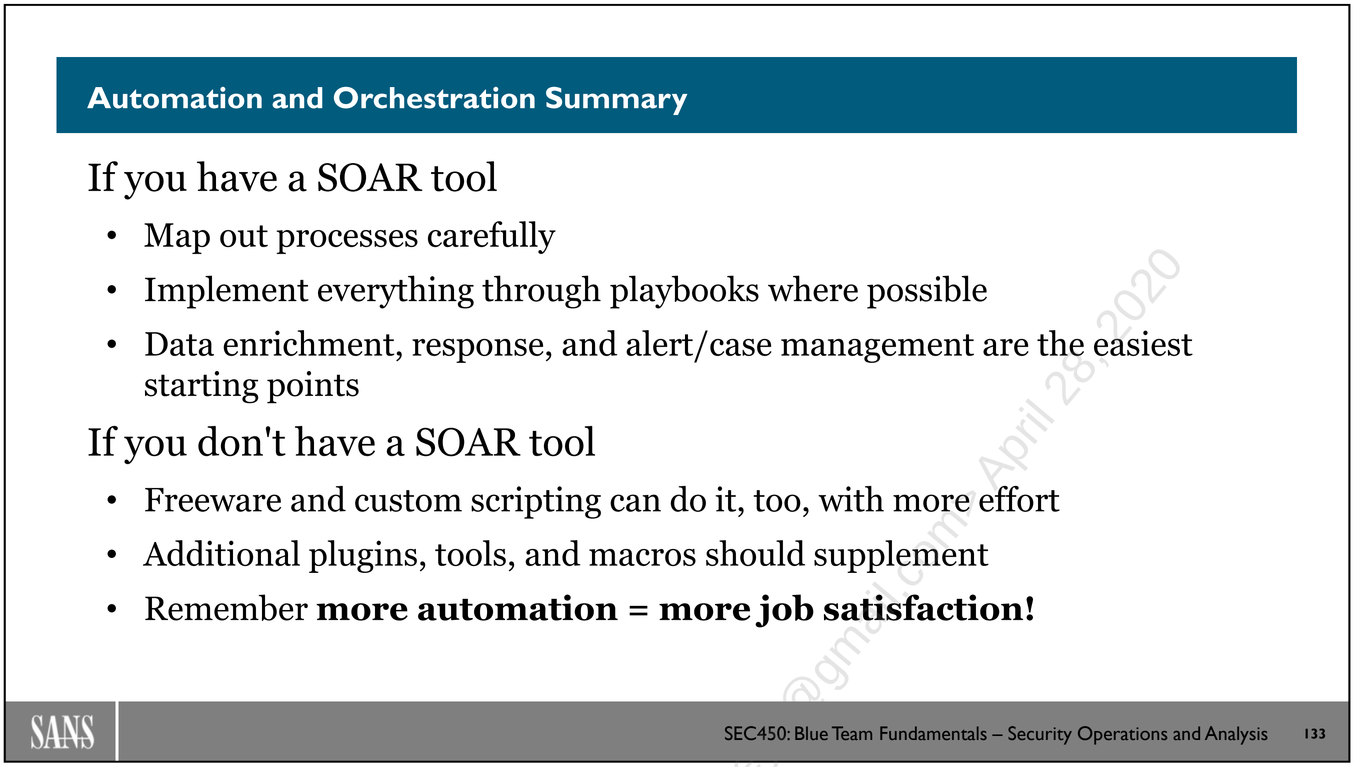
Remember: SOAR tools have no monopoly on automation!

- **SOAR is ultimately scripting with a GUI**
- The value-add was making it easier to do
  - Vendors make pre-made functions for integrating with common tools
  - Lets you connect those functions by connecting boxes on the screen
  - GUI + pre-built functions lower the bar for custom scripting
- **Anything done in a SOAR tool can be done in scripts**
- **PowerShell** and **Python** are the most useful for analysts

DIY Scripting

Remember, SOAR is not a revolution in technology. SOAR tools take what we've been doing for years and making it easier for the average person to implement. SOAR tools are ultimately running code in the background with logic behind it connecting when and which functions to run. The value-add that made it a "new" product category was simply pre-writing atomic functions for various vendor devices and allowing you to connect those functions by placing the boxes on a GUI screen with easy to design code flow control.

The message here should be clear – you do *not* need a SOAR tool at all to get any of the things done we have mentioned here; these tools simply make it faster and easier to do because the code is written for you. Do not discount the immense flexibility and ease of use available in PowerShell and Python. Although there is a learning curve associated with both, they are relatively friendly languages that have an enormous pre-written code base. Cobbling together something from pre-written snippets on Stack Overflow, GitHub, and other sites will likely be possible for an incredible number of common scenarios. Remember, Kansa is entirely written in PowerShell and does a lot of the same functions a SOAR tool might do. Not only will learning these two languages help you automate your job but since many tools are already written in these languages, it will help you understand and modify those tools as well.



Automation and Orchestration Summary

If you have a SOAR tool

- Map out processes carefully
- Implement everything through playbooks where possible
- Data enrichment, response, and alert/case management are the easiest starting points

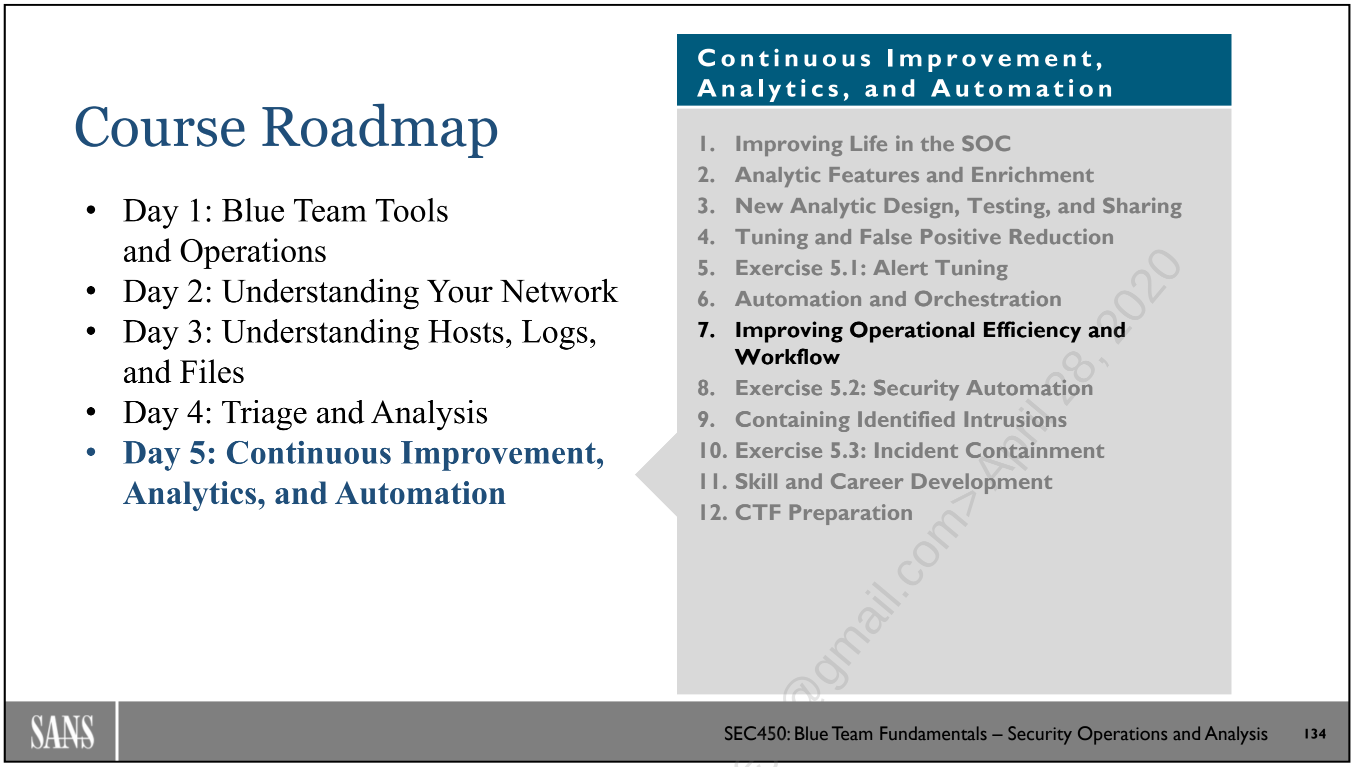
If you don't have a SOAR tool

- Freeware and custom scripting can do it, too, with more effort
- Additional plugins, tools, and macros should supplement
- Remember **more automation = more job satisfaction!**

Automation and Orchestration Summary

Ultimately, automation and orchestration should be part of your job whether you have an official SOAR tool or not. If you do, carefully map out all your common processes and find out where each step can be done with the tool, instead of manually by humans. Where a process can become risky, you should insert manual checks and interventions since the paradox of automation means humans become *more* important instead of less. If you don't have a SOAR tool, no reason to despair, projects like WALKFOFF are starting to show up as open source alternatives, and there is a treasure-trove of scripts in various languages waiting to be glued together on GitHub. It will just take more time to put it all together.

Regardless of your status on SOAR tools, there are opportunities for the smaller processes to be automated and made more pleasant as well. In the next section, we'll cover how to simplify email writing, form filling, and many other small repetitive tasks using browser plugins, macros, and templates, hopefully leaving you with more time to do the fun and engaging work.



Course Roadmap

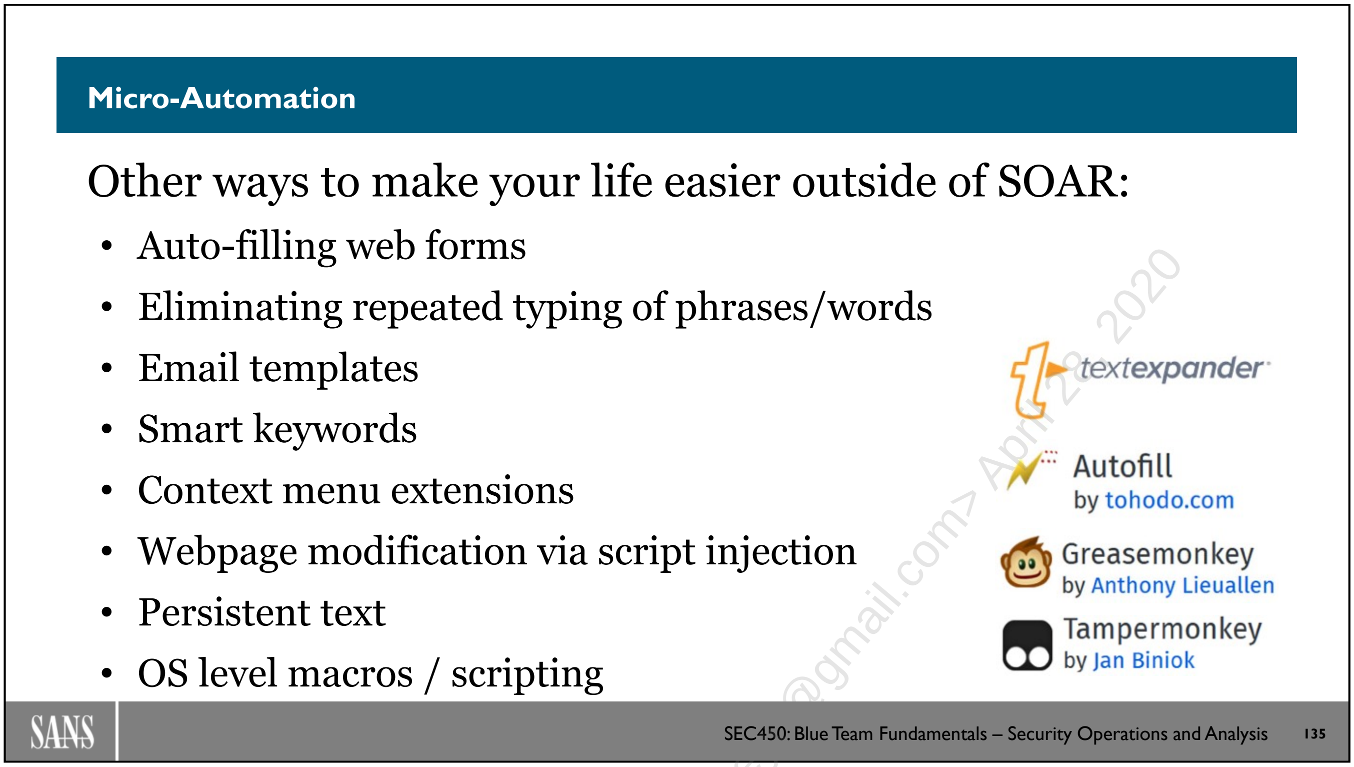
- Day 1: Blue Team Tools and Operations
- Day 2: Understanding Your Network
- Day 3: Understanding Hosts, Logs, and Files
- Day 4: Triage and Analysis
- **Day 5: Continuous Improvement, Analytics, and Automation**

Continuous Improvement, Analytics, and Automation

1. Improving Life in the SOC
2. Analytic Features and Enrichment
3. New Analytic Design, Testing, and Sharing
4. Tuning and False Positive Reduction
5. Exercise 5.1: Alert Tuning
6. Automation and Orchestration
7. **Improving Operational Efficiency and Workflow**
8. Exercise 5.2: Security Automation
9. Containing Identified Intrusions
10. Exercise 5.3: Incident Containment
11. Skill and Career Development
12. CTF Preparation

SANS

This page intentionally left blank.



Micro-Automation

Other ways to make your life easier outside of SOAR:

- Auto-filling web forms
- Eliminating repeated typing of phrases/words
- Email templates
- Smart keywords
- Context menu extensions
- Webpage modification via script injection
- Persistent text
- OS level macros / scripting

textexpander

Autofill
by tohodo.com

Greasemonkey
by Anthony Lieuallen

Tampermonkey
by Jan Biniok

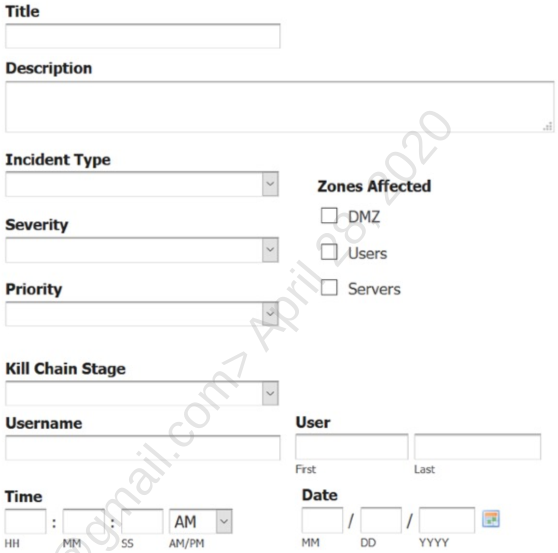
Micro-Automation

SOAR is a great option for many complicated tasks, but there are certain types of tasks they don't address. For all the other interfaces we use, programs we run, and things we type there are opportunities for what I've called "micro-automation" in this section. This section covers tasks like filling in web forms automatically, text expansion for common typed phrases and entities, smart keywords for browser bookmarks, JavaScript injection to change webpage behavior, and more. There are plenty of minor interactions we have with our tools throughout daily life that can be made easier through these tools as well. Therefore, in the next few slides, we will discuss how we can further optimize our workflow by typing as little as possible in every possible situation, save text in web forms so that it's not accidentally lost, and how we can modify any webpage with JavaScript to do almost any other customization we'd like! We'll also cover DIY scripting for creating your own orchestration and automation.

Form Filling in Browsers

Entering the same info over and over again?

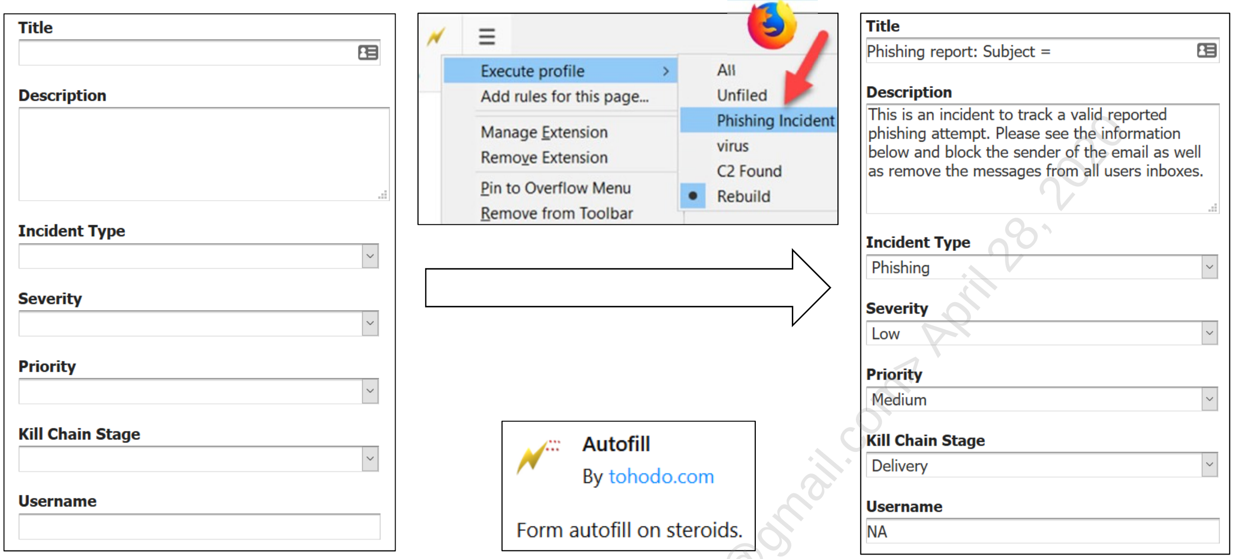
- This often happens with common alert types
- Time for a form filler!
- Fill out common items
- One click = form complete!



Form Filling in Browsers

One of the most grinding parts of any job based on filling out incidents in a ticketing system is repeated data entry. Invariably, much of the text we put in day to day will contain the same text in description boxes, the same checkboxes checked, and the same fields filled out for dropdowns. Why repeat entering this information over and over manually when it is not necessary? Browsers have form fillers built-in and available as plugins for exactly this reason. All it will take is some minor setup. Once you have determined some boilerplate text for the description boxes and the appropriate other settings, these can be saved and recalled, leaving you only to fill in specific details like user and hostnames.

Autofill Plugin Example



Title	
Description	
Incident Type	
Severity	
Priority	
Kill Chain Stage	
Username	

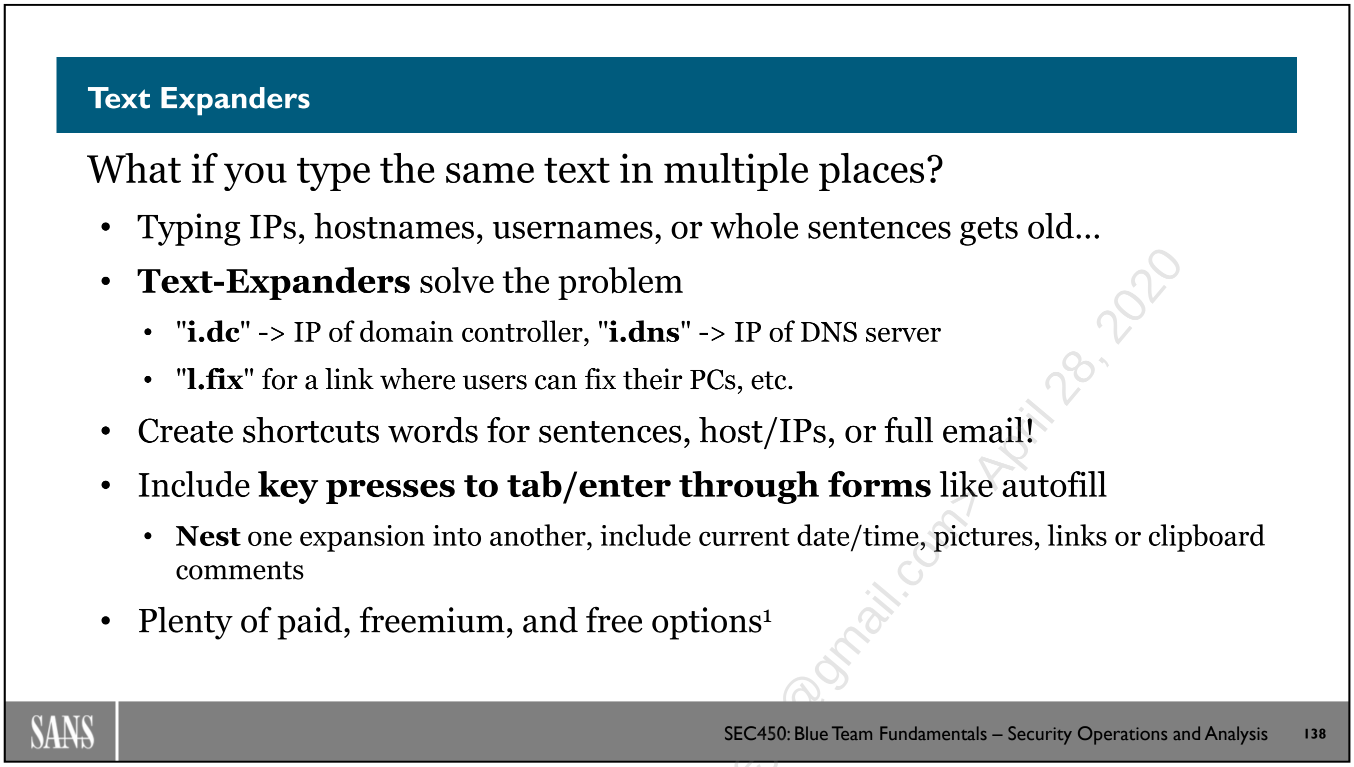
Title	Phishing report: Subject =
Description	This is an incident to track a valid reported phishing attempt. Please see the information below and block the sender of the email as well as remove the messages from all users inboxes.
Incident Type	Phishing
Severity	Low
Priority	Medium
Kill Chain Stage	Delivery
Username	NA

Autofill Plugin Example

Though there are many options for similar plugins depending upon which browser you use, this slide shows an example of what they should allow. On this slide, we show the "Autofill" plugin, available for Firefox and Chrome. Several different "profiles" have been created to fill in forms for various incident types.[1] Using this method, the base text for all fields can be filled in and then with the click of a button, Autofill will remember what is in every field. The next time you go to fill in a similar condition, you only must select the profile from the list and all the values previously saved will snap into place, saving you the time and mental anguish of typing it in each time by yourself.

One note on this specific plugin: If you decide to use Autofill, you must turn on manual fill mode in the settings. In its default state, fields will be filled with the default profile as soon as a page loads.

[1] https://addons.mozilla.org/en-US/firefox/addon/autofill-quantum/



Text Expanders

What if you type the same text in multiple places?

- Typing IPs, hostnames, usernames, or whole sentences gets old...
- **Text-Expanders** solve the problem
    - "**i.dc**" -> IP of domain controller, "**i.dns**" -> IP of DNS server
    - "**l.fix**" for a link where users can fix their PCs, etc.
- Create shortcuts words for sentences, host/IPs, or full email!
- Include **key presses to tab/enter through forms** like autofill
    - **Nest** one expansion into another, include current date/time, pictures, links or clipboard comments
- Plenty of paid, freemium, and free options[1]

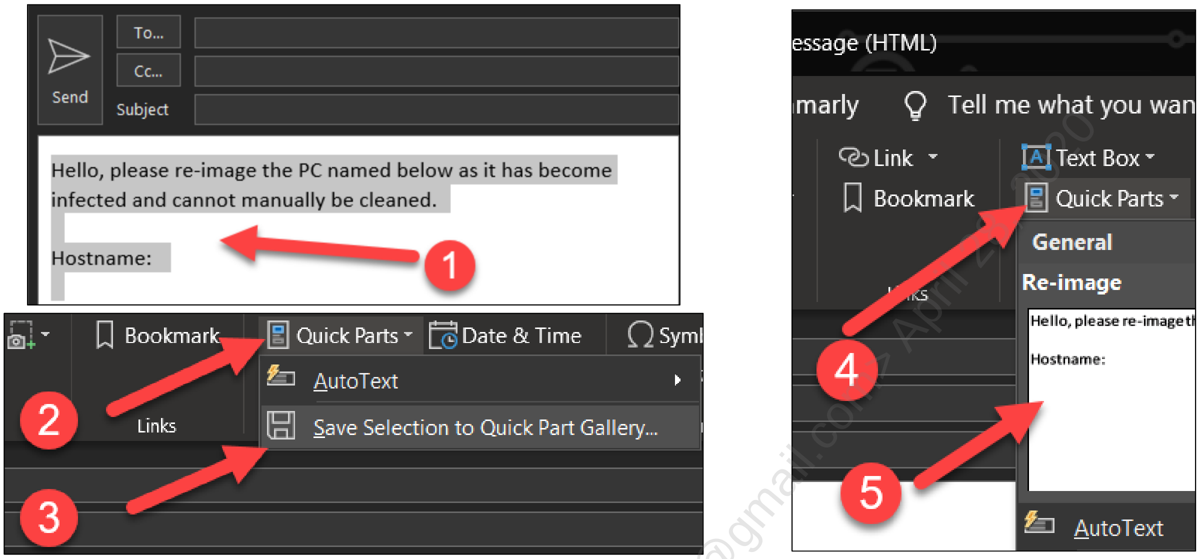
Text Expanders

To take this idea even further—for any long names, IPs, or even sentences and paragraphs you frequently use in disparate applications, text expanders might be your answer. These tools let you make shortcuts to make shortcut names for these common items and have them auto-expanded into full form. An example of this would be taking the sentence—"This computer has been identified as infected, please contact the user and have the machine re-imaged."—and making an alias such as "rmg" for it. With a text expander, you can now drop that sentence anywhere you need to on your PC—in a ticket, email, or instant message, simply by typing four characters. This capability is similar to auto-filling forms but allows the autofill to be cross-application. The key here is to pick a keyword to expand from that you won't type by accident and that you can remember. Using a well-defined keyword format can help you with this. For links, you could make a convention to have them all start with L for example, then perhaps make "l.home" for a homepage link, or "l.fix" for the link to instructions on fixing a machine. The same could be done with I and IP addresses—"i.dc" could be for a domain controller, and "i.dns" could be the IP for your DNS server, etc. This frees you from having to remember frequently used IP addresses and the pain of typing them repeatedly.

There are both commercial and freeware text expansion programs. Some offer features such as dynamic fields like the current date and time, sentence autocompletion, cross-device snippet sharing, and the ability to export shortcuts for coworkers. TextExpander is a commercial option that works with Windows, and Macs. At the price of $3.33 a month, it's not hard to justify if it's something you use multiple times a day, AutoHotKey is a free alternative that lets you create macros on top of text expansion and much more. There are plenty of other options as well.[1]

[1] https://techwiser.com/text-expander-apps-for-windows/

Outlook Quick Parts



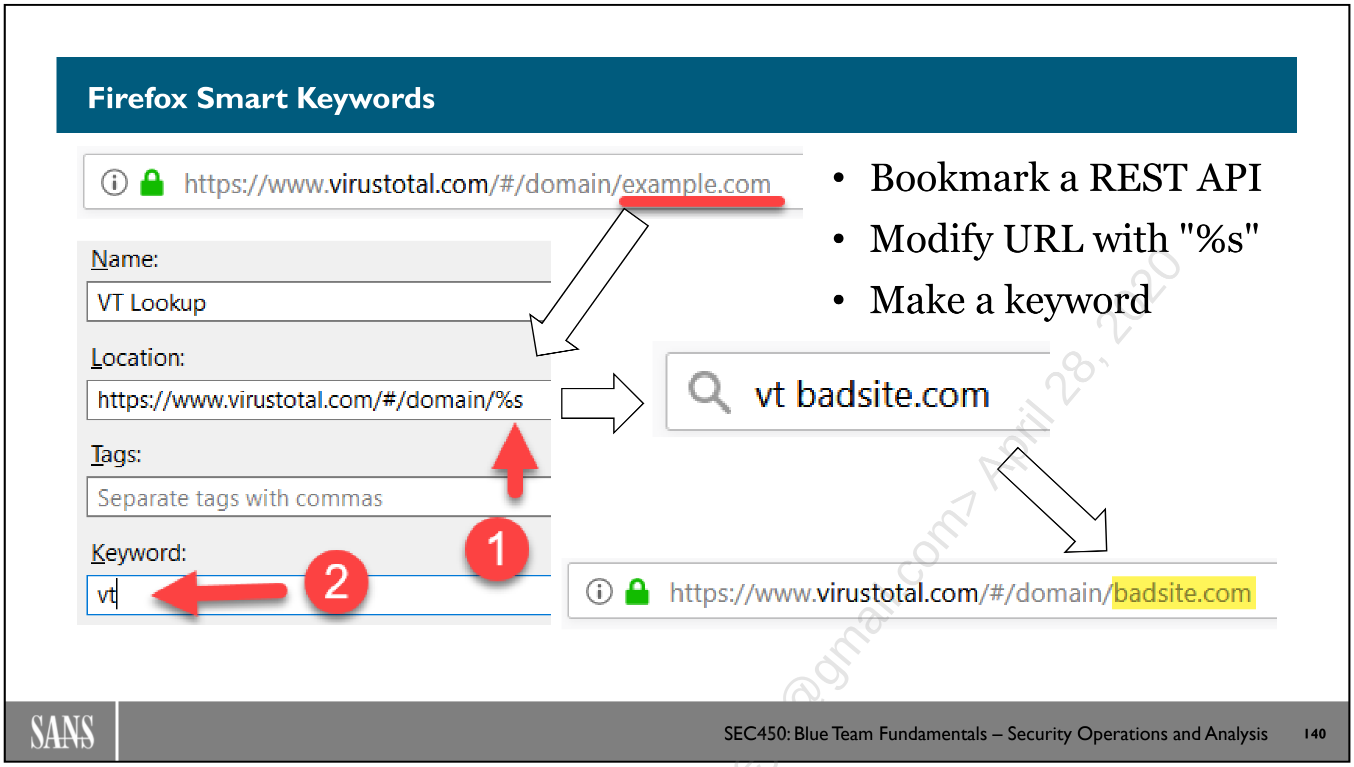
Hello, please re-image the PC named below as it has become infected and cannot manually be cleaned.

Hostname:

Outlook Quick Parts

There's another dedicated way to recreate emails in Outlook with ease, and that feature is called "Quick Parts."[1] To create a saved email template, simply go to the new email creation window and select the text you'd like to turn into a quick part. Then go to the text pane in the ribbon and select "Quick Parts," then "Save Selection to Quick Parts Gallery." Give it a name, (re-image) was used in this case. Then, the next time you open a blank email, you can click the Quick Parts button and see the name of any saved email forms you've created, making duplicating them a single-click affair.

[1] https://support.office.com/en-us/article/create-reuseable-text-blocks-for-email-messages-8fb6c723-c960-4c8c-9790-3e43ddc4b186



Firefox Smart Keywords

- Bookmark a REST API
- Modify URL with "%s"
- Make a keyword

https://www.virustotal.com/#/domain/example.com

Name: VT Lookup

Location: https://www.virustotal.com/#/domain/%s

Tags: Separate tags with commas

Keyword: vt

vt badsite.com

https://www.virustotal.com/#/domain/badsite.com

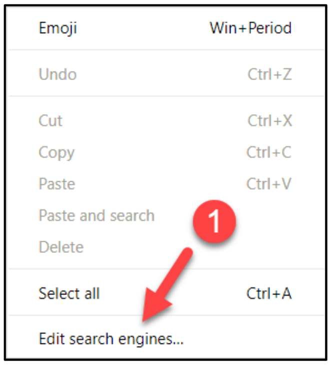
Firefox Smart Keywords

One of the most under-utilized features of all time in Firefox is smart keywords. Nearly every website-based lookup an analyst does goes to a REST API, meaning the URL is the same with only the specific item we are trying to look up as the changing part. Because of this, we can use Firefox's ability to create "smart keywords", or shortcuts with a variable that gets filled in, to take advantage of this, making lookups fast and easy!

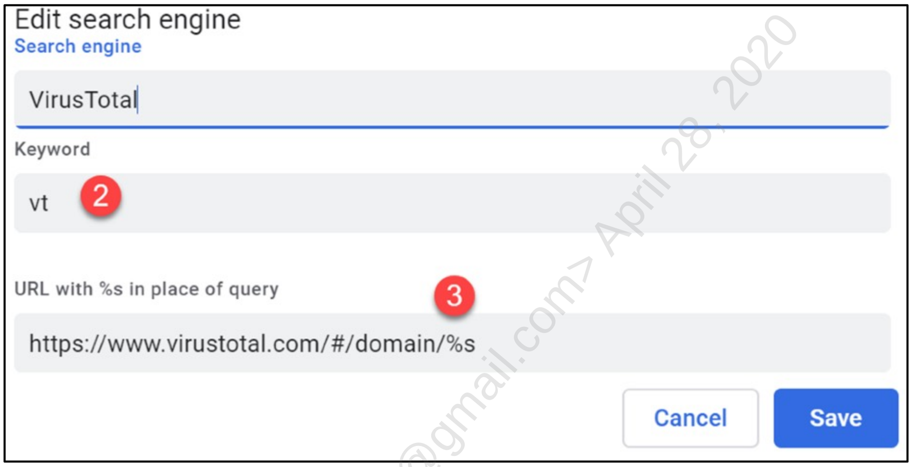
The slide has a demonstration of the simple process for dynamic bookmark creation. First, you must look at the URL on any given website you often visit and see if it has a simple format that can have a simple piece of text changed to take you to the related page for the item. If so, save it as a normal bookmark, then go into the bookmark editing menu (1), pick a "keyword," the shortcut you will use, then delete the specific piece of info and replace it with a %s. Rename the bookmark to something generic and then hit save. Now, to look up any item on the given website, all you must do is type "[keyword] [variable]" and press enter! (2) The dynamic bookmark will place your variable into the URL and whisk you off to the desired page without the need to load up the page first. This feature is outstanding for ticket system ticket numbers, threat intel websites, and all SOC tools in general.

Chrome Smart Keywords

- Use the search bar "Edit Search Engines" option



URL goes here:


Edit search engine

Search engine: VirusTotal

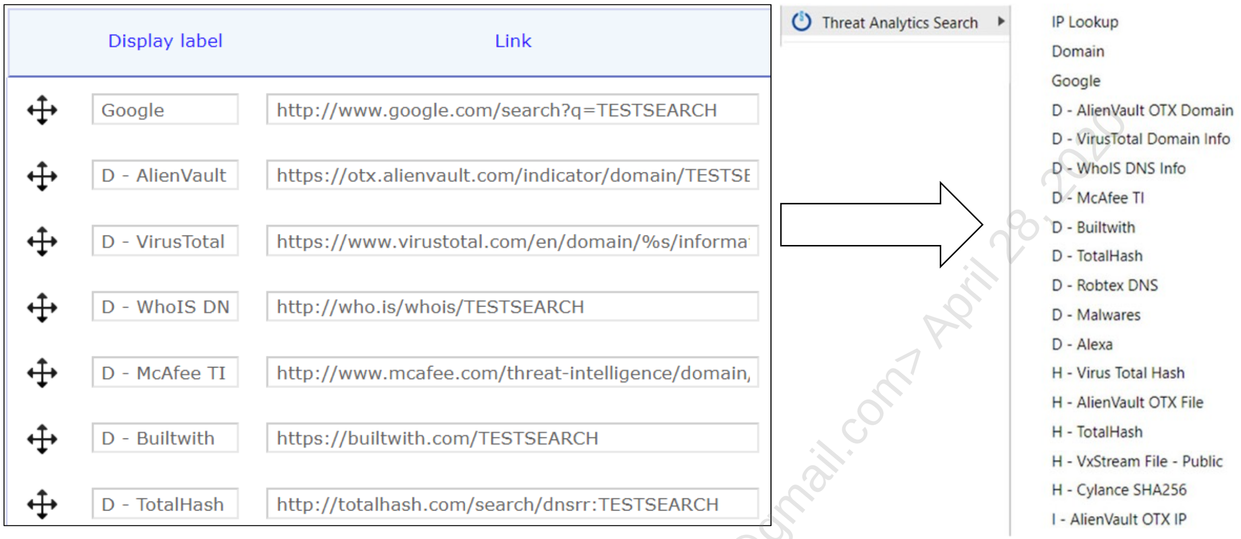
Keyword: vt

URL with %s in place of query: https://www.virustotal.com/#/domain/%s

Chrome Smart Keywords

Chrome can do the same trick, but it's not implemented through bookmarks like it is in Firefox. In Chrome, to set up a smart keyword style shortcut, you must right click on the search bar and hit "Edit search engines." Click "Add" to make a new option and paste the URL in the URL box with a %s placeholder, then pick a keyword, hit save, and you're ready to go!

For Chrome: Threat Analytics Search Plugin

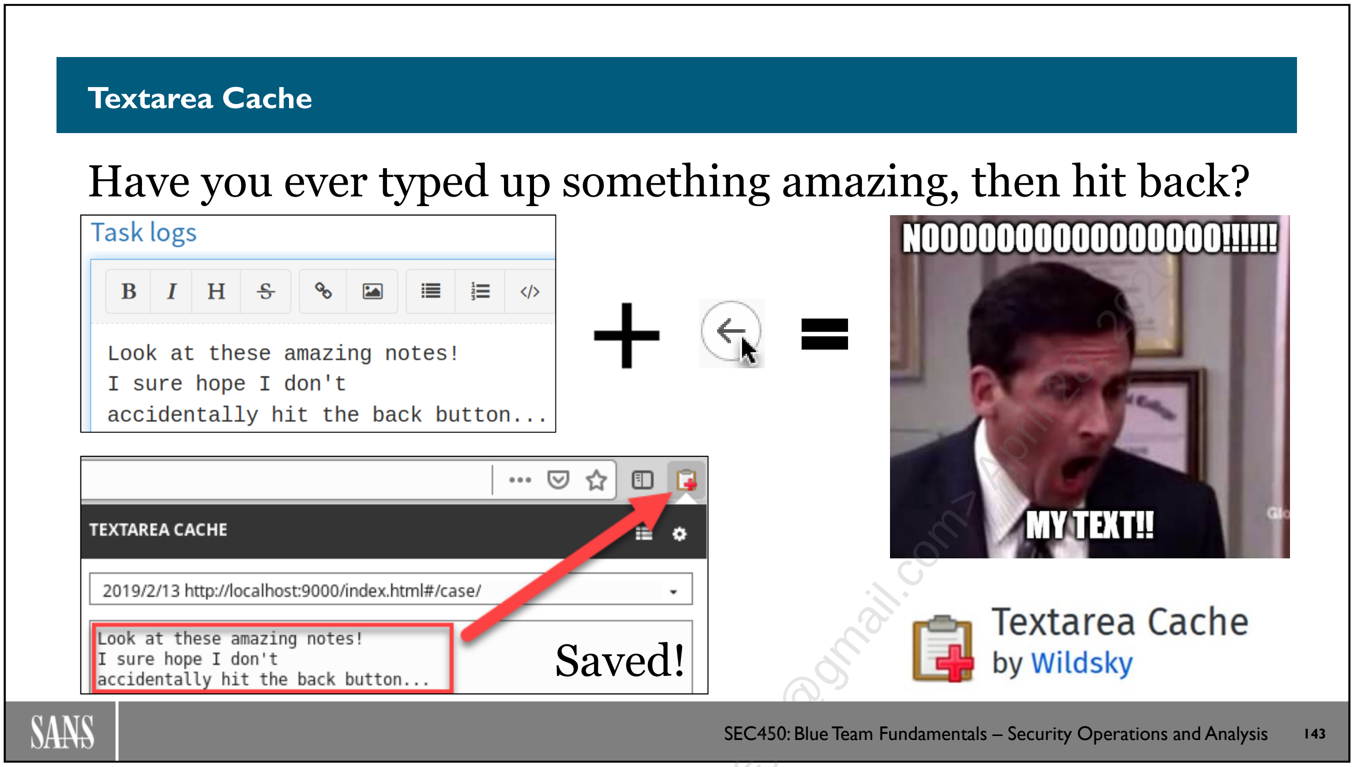


	Display label	Link
✥	Google	http://www.google.com/search?q=TESTSEARCH
✥	D - AlienVault	https://otx.alienvault.com/indicator/domain/TESTSE
✥	D - VirusTotal	https://www.virustotal.com/en/domain/%s/informa
✥	D - WhoIS DN	http://who.is/whois/TESTSEARCH
✥	D - McAfee TI	http://www.mcafee.com/threat-intelligence/domain,
✥	D - Builtwith	https://builtwith.com/TESTSEARCH
✥	D - TotalHash	http://totalhash.com/search/dnsrr:TESTSEARCH

For Chrome: Threat Analytics Search Plugin

Chrome user? You have another option in the "Threat Analytics Search" plugin as well. This plugin allows you to specify a REST-style URL pattern as shown on the slide and puts a right-click option in the context menu for it. Any time you highlight text on a page, you can use the Threat Analytics search context menu to perform a lookup of that item on any services you have set up.

[1] https://chrome.google.com/webstore/detail/threat-analytics-search/eliokoocofjemjjohafbmhmgjmedomko?hl=en-US



Textarea Cache

Have you ever typed up something amazing, then hit back?

Task logs

```
Look at these amazing notes!
I sure hope I don't
accidentally hit the back button...
```

2019/2/13 http://localhost:9000/index.html#/case/

```
Look at these amazing notes!
I sure hope I don't
accidentally hit the back button...
```

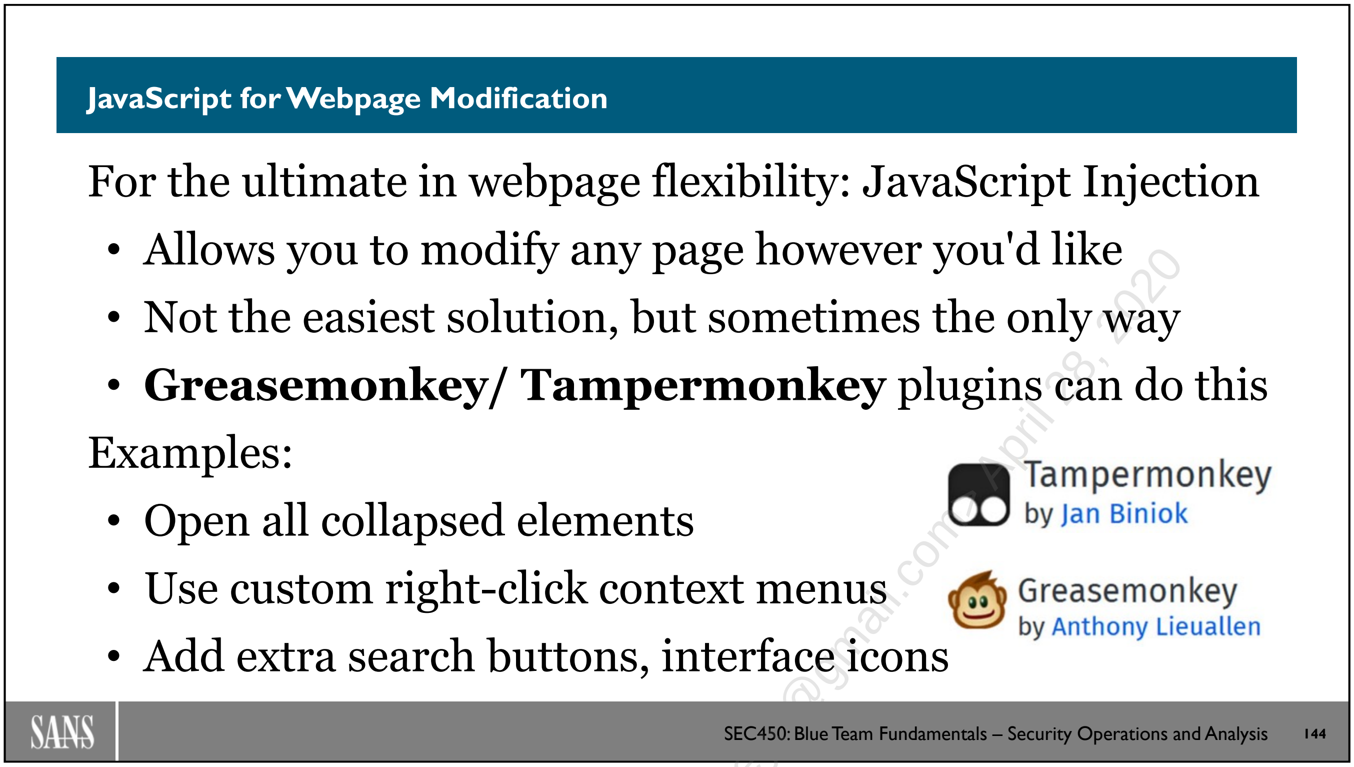
Saved!

Textarea Cache by Wildsky

Textarea Cache

Although it may seem like a minor problem, trust me that the time will come around when this plugin saves the day. Textarea Cache is a Firefox plugin that automatically saves anything you type into a web form so that if you accidentally navigate away from the window you were working in, you won't lose the text you typed.

Many ticketing systems will not automatically save what is entered into a text box (like TheHive task logs shown above) until you push the "save" button. In the case where you have taken a long time to fill out detailed notes and then accidentally hit the wrong button, without this plugin, you'll have to do it all over again. With Textarea Cache installed, you can simply right click and drop the text right back where it was. Amazing! Unfortunately, at the time of this writing, there does not seem to be a viable plugin for doing this with Chrome. Several have existed in the past but have become abandonware that is no longer compatible with the new versions.



JavaScript for Webpage Modification

For the ultimate in webpage flexibility: JavaScript Injection

- Allows you to modify any page however you'd like
- Not the easiest solution, but sometimes the only way
- **Greasemonkey/ Tampermonkey** plugins can do this

Examples:

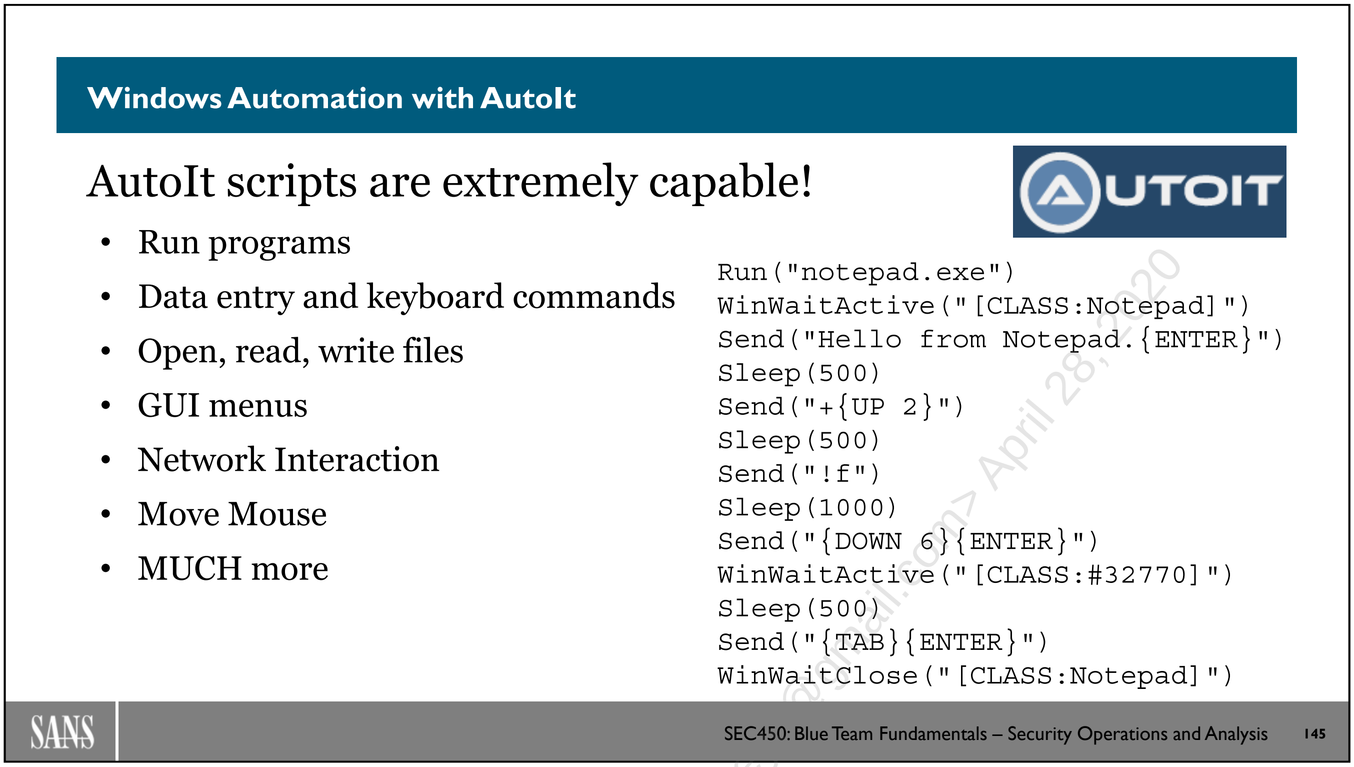
- Open all collapsed elements
- Use custom right-click context menus
- Add extra search buttons, interface icons

Tampermonkey
by Jan Biniok

Greasemonkey
by Anthony Lieuallen

JavaScript for Webpage Modification

Although this is one of the more advanced techniques in the list and requires that you understand at least a bit of JavaScript, it is also one of the most powerful. With plugins like Tampermonkey (multi-platform) and Greasemonkey (Firefox only), you can inject custom JavaScript code into chosen pages that can modify the resulting page in any way you'd like. One example of how I've seen this used in the past is in a ticketing system that opened each case with all notes collapsed, meaning you had to click a list of tiny triangles to read through what had happened. One of the analysts on the team finally got fed up enough with this and decided to fix it by creating a Greasemonkey script and distributing it to the team. Once installed, the analysts could simply click to run the script or not when they were on a case's notes page, causing all the notes to either expand or collapse, saving them 20+ clicks in the case of larger incidents. Scripts can do lots of other things as well, such as create custom right-click context menus, add extra buttons or functionality to a page, validate fields, and more. Unfortunately, we do not have the space to dive into this topic in detail, but it is an option worth knowing about for those extreme cases. For a set of pre-developed user scripts, check out greasyfork.org.



Windows Automation with AutoIt

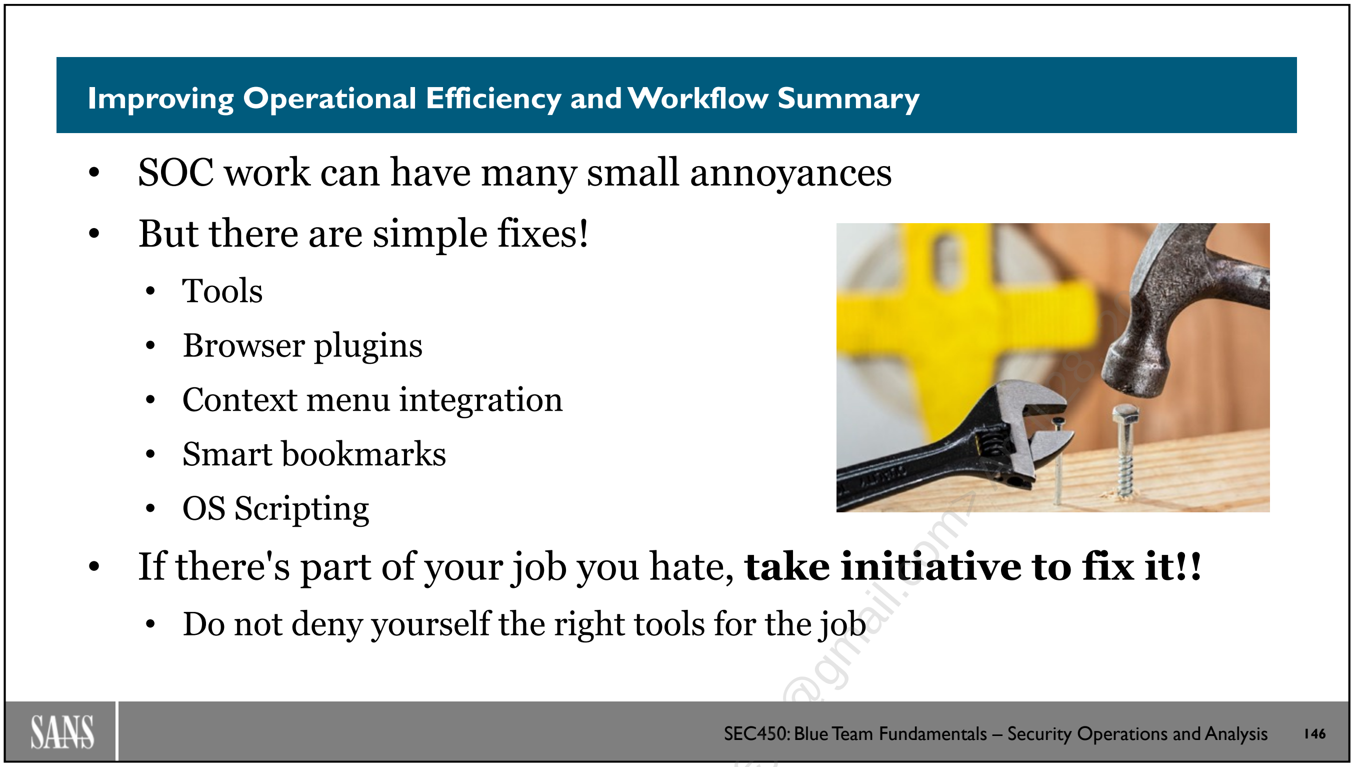
AutoIt scripts are extremely capable!

- Run programs
- Data entry and keyboard commands
- Open, read, write files
- GUI menus
- Network Interaction
- Move Mouse
- MUCH more

```
Run("notepad.exe")
WinWaitActive("[CLASS:Notepad]")
Send("Hello from Notepad.{ENTER}")
Sleep(500)
Send("+{UP 2}")
Sleep(500)
Send("!f")
Sleep(1000)
Send("{DOWN 6}{ENTER}")
WinWaitActive("[CLASS:#32770]")
Sleep(500)
Send("{TAB}{ENTER}")
WinWaitClose("[CLASS:Notepad]")
```

Windows Automation with AutoIt

Finally, if you can't get the job done with any of the previously mentioned tools, there's **AutoIt** – a fully-featured scripting language that can automate nearly anything both file access and GUI-wise. It has all the functions you could possibly need to accomplish automation, but it is also fairly complex. One of the demo scripts, for example, will open notepad, type a full multiline message out, highlight it, then navigate to the menu and quit out of Notepad without saving the file. They accomplish this action in only 13 lines of code, 4 of which are sleep statements, so although the scripts may look complex, you can get a lot done with very little. Tools like this could fully automate the process of opening a program that uses a thick client instead of a web browser, select options, and type all the information inside it for you. If you haven't come to this conclusion yet, the idea is no matter the task you need to automate, there is a tool out there that can help you do it.



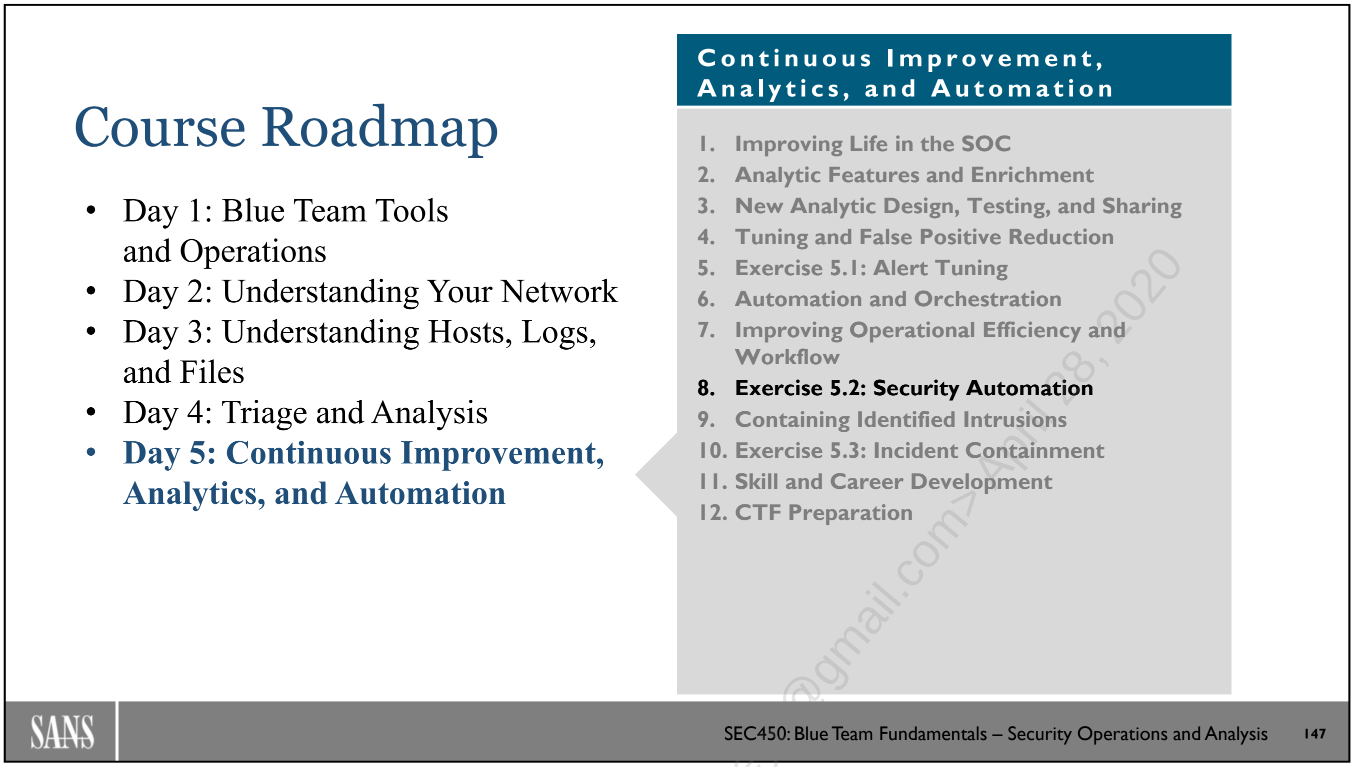
Improving Operational Efficiency and Workflow Summary

- SOC work can have many small annoyances
- But there are simple fixes!
  - Tools
  - Browser plugins
  - Context menu integration
  - Smart bookmarks
  - OS Scripting
- If there's part of your job you hate, **take initiative to fix it!!**
  - Do not deny yourself the right tools for the job

SEC450: Blue Team Fundamentals – Security Operations and Analysis 146

Improving Operational Efficiency and Workflow Summary

While there are many smaller routine tasks outside of alert workflow that we must attend to, there's no reason we can't make these items simpler as well. The tools are out there; we simply must have time to dream up a solution and implement it! Using custom context menus, smart bookmarks, and scripting can keep everything one click away instead of 10, making even those tiny tasks seem effortless.



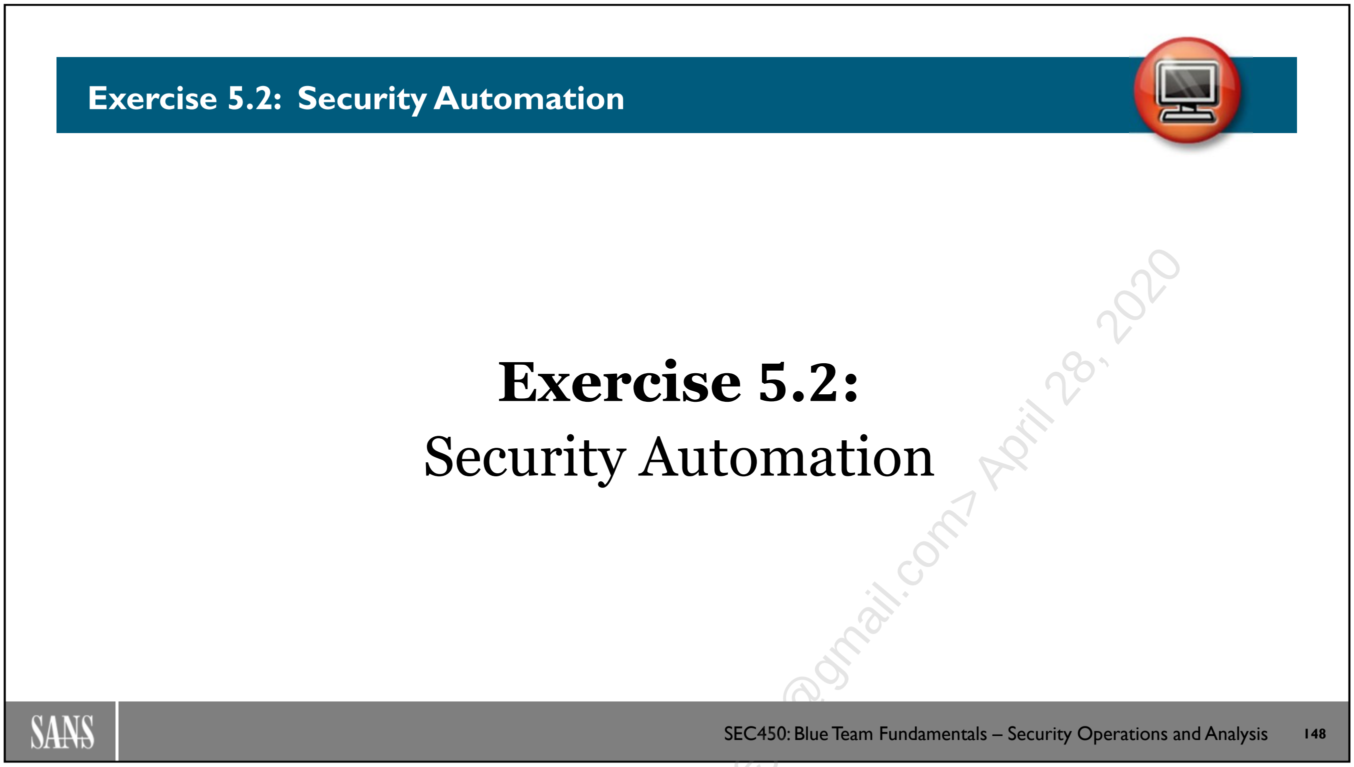
Course Roadmap

- 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**

Continuous Improvement, Analytics, and Automation

1. Improving Life in the SOC
2. Analytic Features and Enrichment
3. New Analytic Design, Testing, and Sharing
4. Tuning and False Positive Reduction
5. Exercise 5.1: Alert Tuning
6. Automation and Orchestration
7. Improving Operational Efficiency and Workflow
8. **Exercise 5.2: Security Automation**
9. Containing Identified Intrusions
10. Exercise 5.3: Incident Containment
11. Skill and Career Development
12. CTF Preparation

This page intentionally left blank.



Exercise 5.2: Security Automation

Exercise 5.2:

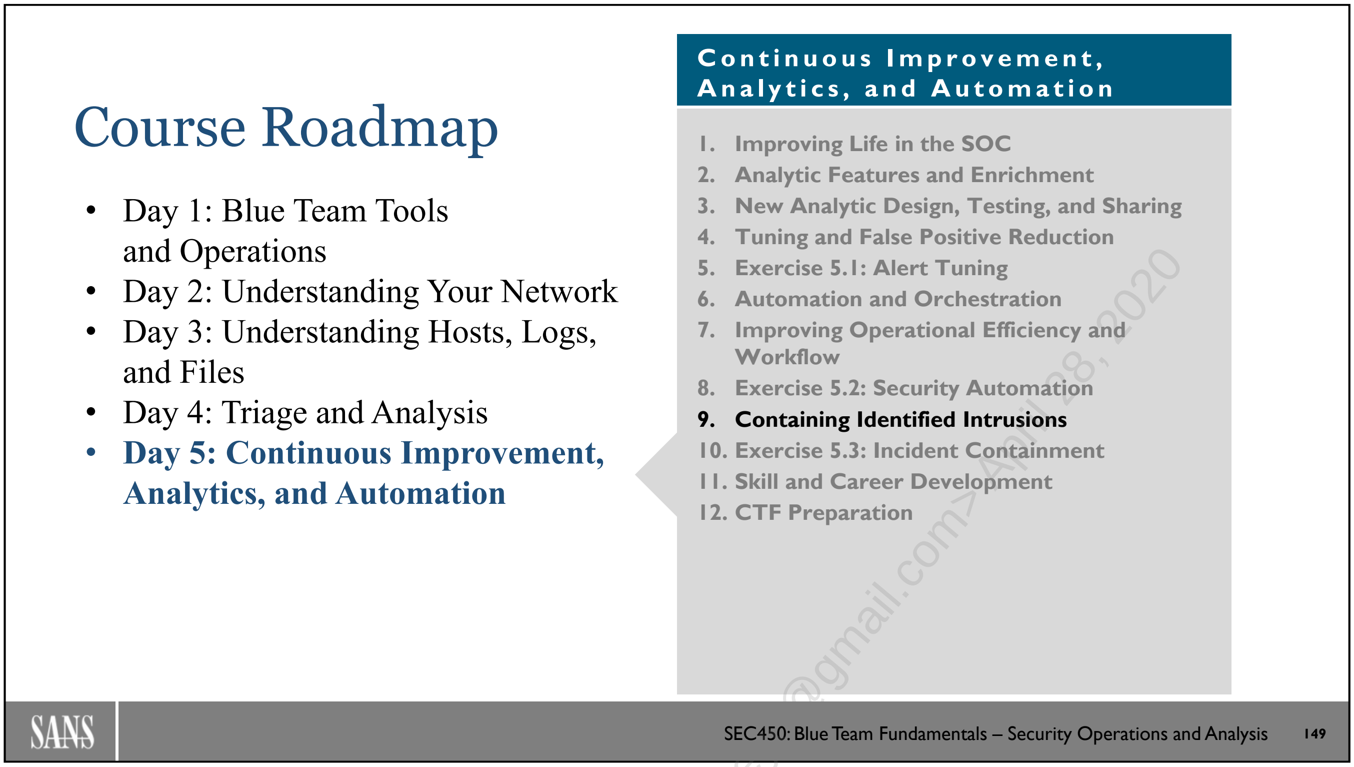
Security Automation

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 148

Exercise 5.2: Security Automation

Please go to Exercise 5.2 in the SEC450 Workbook or virtual wiki.



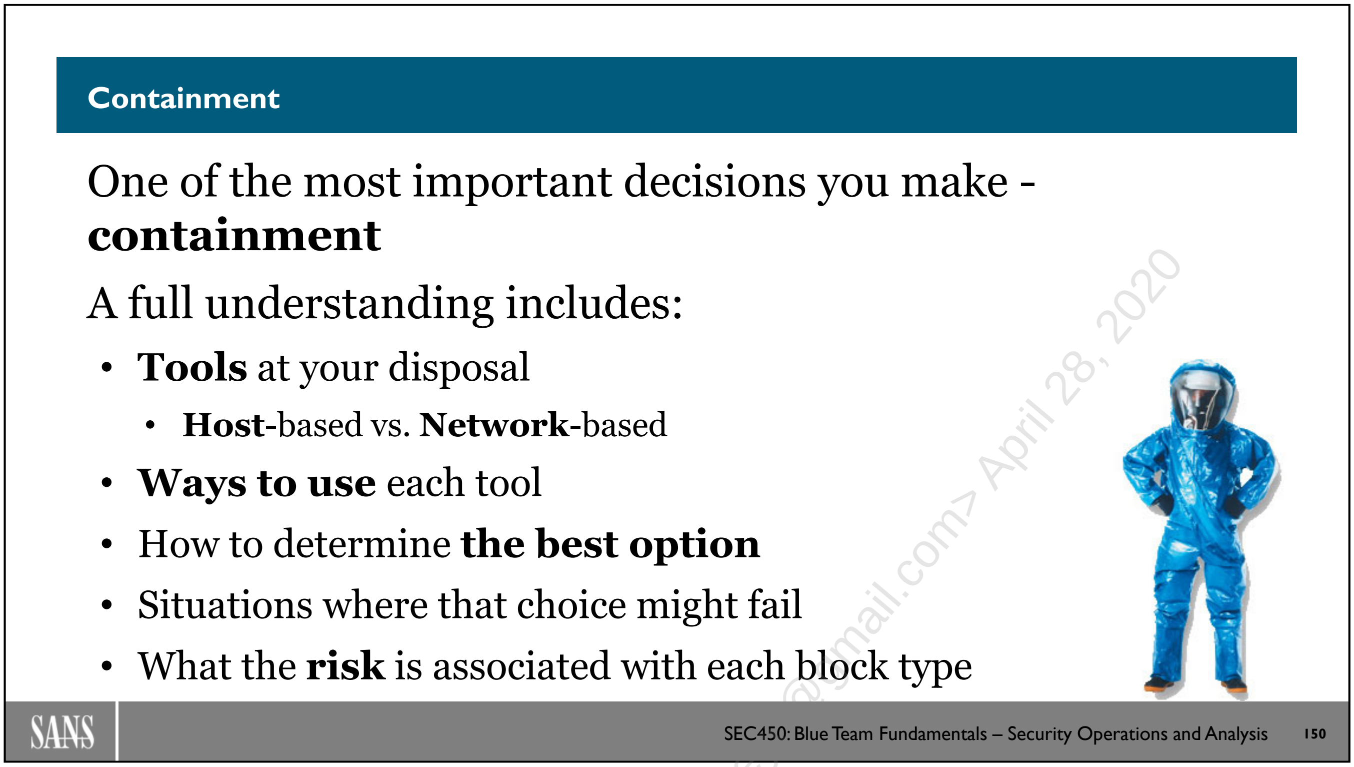
Course Roadmap

- 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**

Continuous Improvement, Analytics, and Automation

1. Improving Life in the SOC
2. Analytic Features and Enrichment
3. New Analytic Design, Testing, and Sharing
4. Tuning and False Positive Reduction
5. Exercise 5.1: Alert Tuning
6. Automation and Orchestration
7. Improving Operational Efficiency and Workflow
8. Exercise 5.2: Security Automation
9. **Containing Identified Intrusions**
10. Exercise 5.3: Incident Containment
11. Skill and Career Development
12. CTF Preparation

This page intentionally left blank.



Containment

One of the most important decisions you make - **containment**

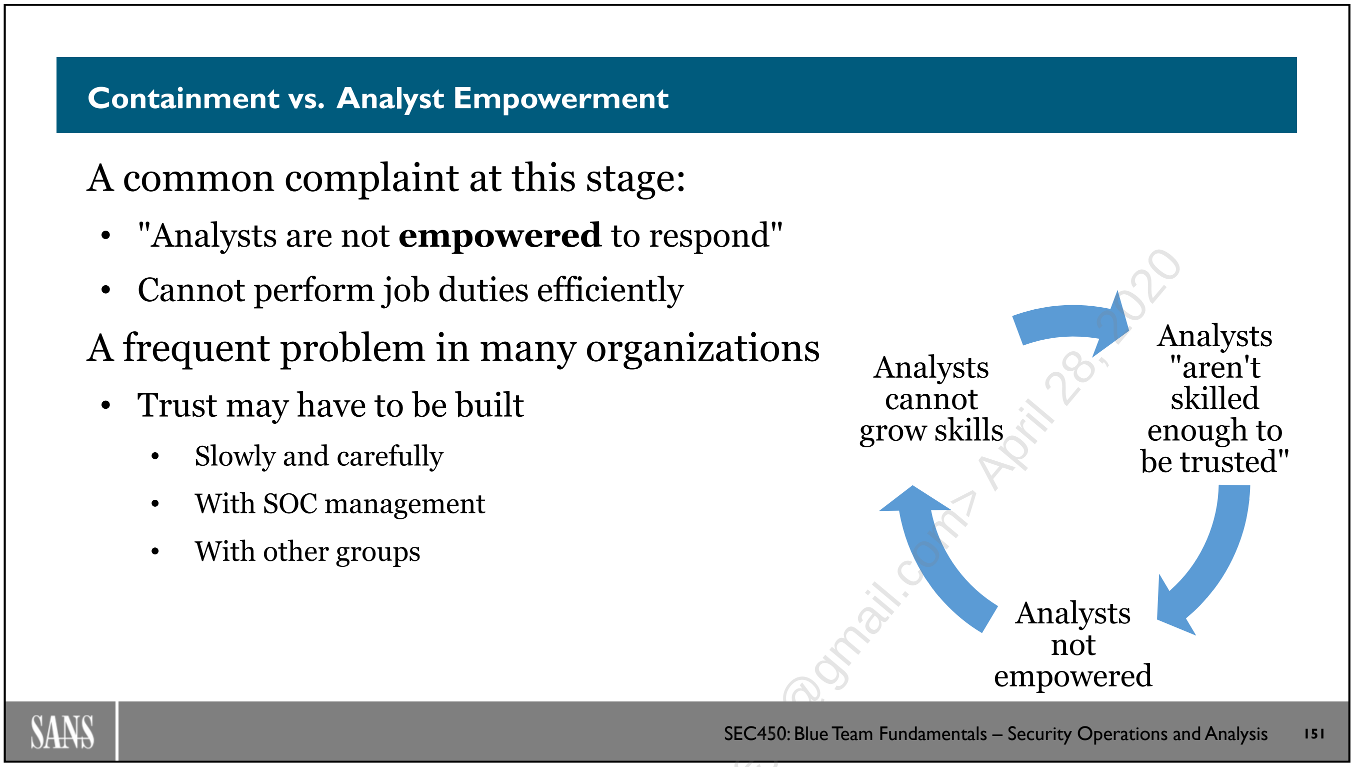
A full understanding includes:

- **Tools** at your disposal
  - **Host**-based vs. **Network**-based
- **Ways to use** each tool
- How to determine **the best option**
- Situations where that choice might fail
- What the **risk** is associated with each block type

SANS
SEC450: Blue Team Fundamentals – Security Operations and Analysis
150

Containment

After triage and investigation when a true positive is confirmed, containment is the next step. While containment can be approached in either short-term or long-term ways, the key pieces are a thorough understanding of your tools and having the power to do it. When walking through the steps for a containment plan, we must consider everything we know about the nature of the breach as well as the type of asset that was affected. Though being able to implement a firewall block to stop command and control is great, what happens when the employee takes the laptop home for the night and uses their home internet connection? On the flip side, perhaps you have implemented a host-specific block for a command and control channel on that laptop. What about the three other people in the environment that received the same virus via email and are about to open it themselves—are they protected? Considerations like this will drive your choice between host and network-based containment strategies.



Containment vs. Analyst Empowerment

A common complaint at this stage:

- "Analysts are not **empowered** to respond"
- Cannot perform job duties efficiently

A frequent problem in many organizations

- Trust may have to be built
  - Slowly and carefully
  - With SOC management
  - With other groups

Analysts cannot grow skills
Analysts "aren't skilled enough to be trusted"
Analysts not empowered

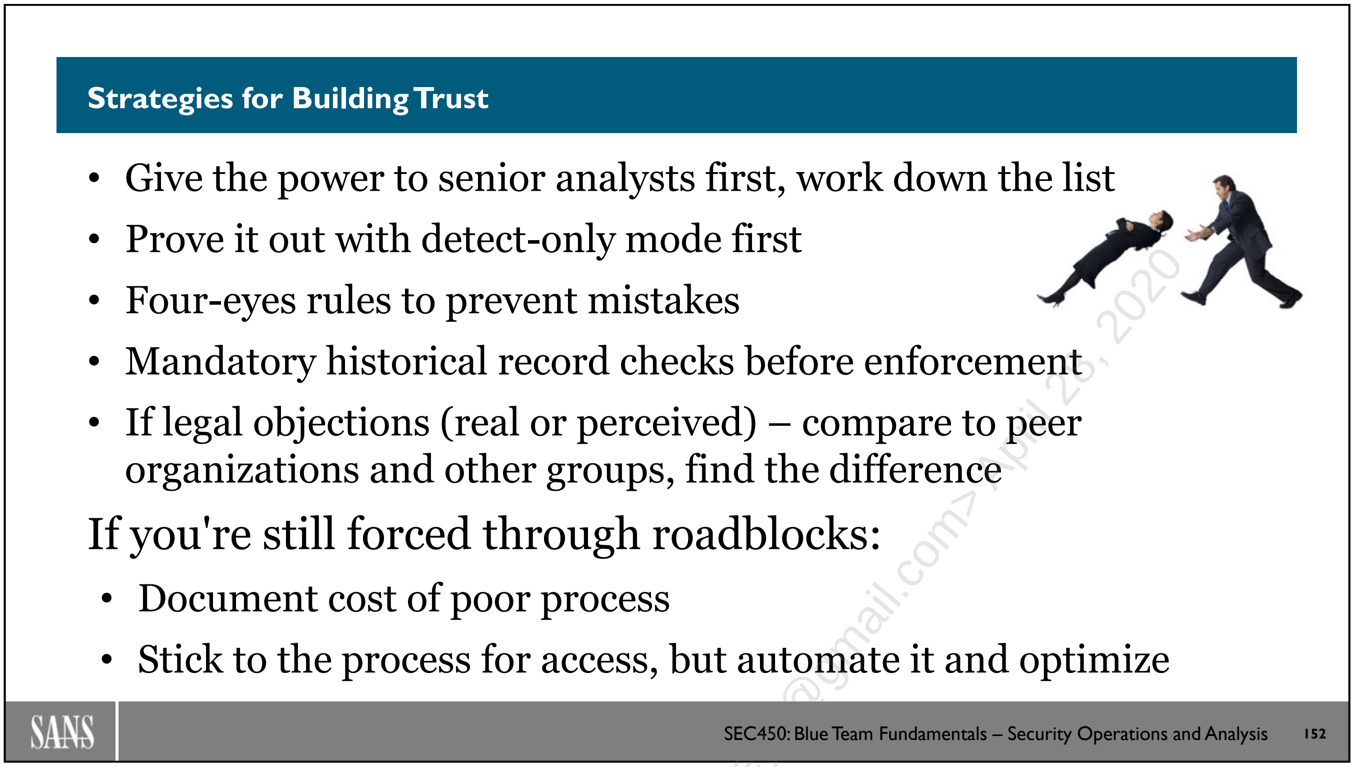
SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 151

Containment vs. Analyst Empowerment

While every network is different (the reason that understanding network architecture is important), there are tools that can be used in almost every environment to disrupt attacks. The assumption that goes with that, however, is analysts must be *empowered to use those tools*, either directly or through established communication with the groups that control them. Not only that, but there must be a process in place to implement the controls quickly. In the human capital model of analyst burnout, we saw that a lack of empowerment leads to a lack of creativity and ultimately a continuing downward spiral of enjoyment in the SOC. In the realm of incident response, this impact is multiplied by the fact the lack of empowerment means an attack goes on longer than it needs to, which no one enjoys.

Although we shouldn't blindly give trust to tools that can cause a business outage, every organization has *some* small group of people who are entrusted with this power. If the SOC is not one of those groups, something has gone wrong. SOCs that have never been given a chance to prove their worth may find themselves in a vicious cycle where management refuses to trust the SOC with preventive power because of a perception that something could go wrong. Because of that, the SOC is never given a chance to show they *can* handle it and doesn't have the chance to grow and learn how to safely take control of those procedures, further feeding the cycle. If this describes your situation, consider this one of the core items that you should address.



Strategies for Building Trust

- Give the power to senior analysts first, work down the list
- Prove it out with detect-only mode first
- Four-eyes rules to prevent mistakes
- Mandatory historical record checks before enforcement
- If legal objections (real or perceived) – compare to peer organizations and other groups, find the difference

If you're still forced through roadblocks:

- Document cost of poor process
- Stick to the process for access, but automate it and optimize

SANS
SEC450: Blue Team Fundamentals – Security Operations and Analysis
152

Strategies for Building Trust

There are plenty of SOCs that operate in organizations with tools to block attacks, but instead of a technical limitation issue, they may have a people/politics issue (sometimes jokingly referred to as Layer 8 and 9 of the network stack). Perhaps there have been mistakes in the past that left a bad impression, or people are reluctant to share control of the devices they administer with another group. While all of these may be legitimate concerns, you will never know unless you can convince them to try it out. To compensate for these risks, here are some ideas that I've seen turn out successful in the past:

Start slowly. You don't have to get full permission for everyone in the SOC to do something at once. Start with the most senior analysts to help build trust.

The same way you roll out whitelisting can be applied in this situation as well—use detect-only mode first. If you can simulate having certain capabilities and tracking how they would've turned out in reality while only generating alerts, this can help build confidence in the team's ability to take on the responsibility.

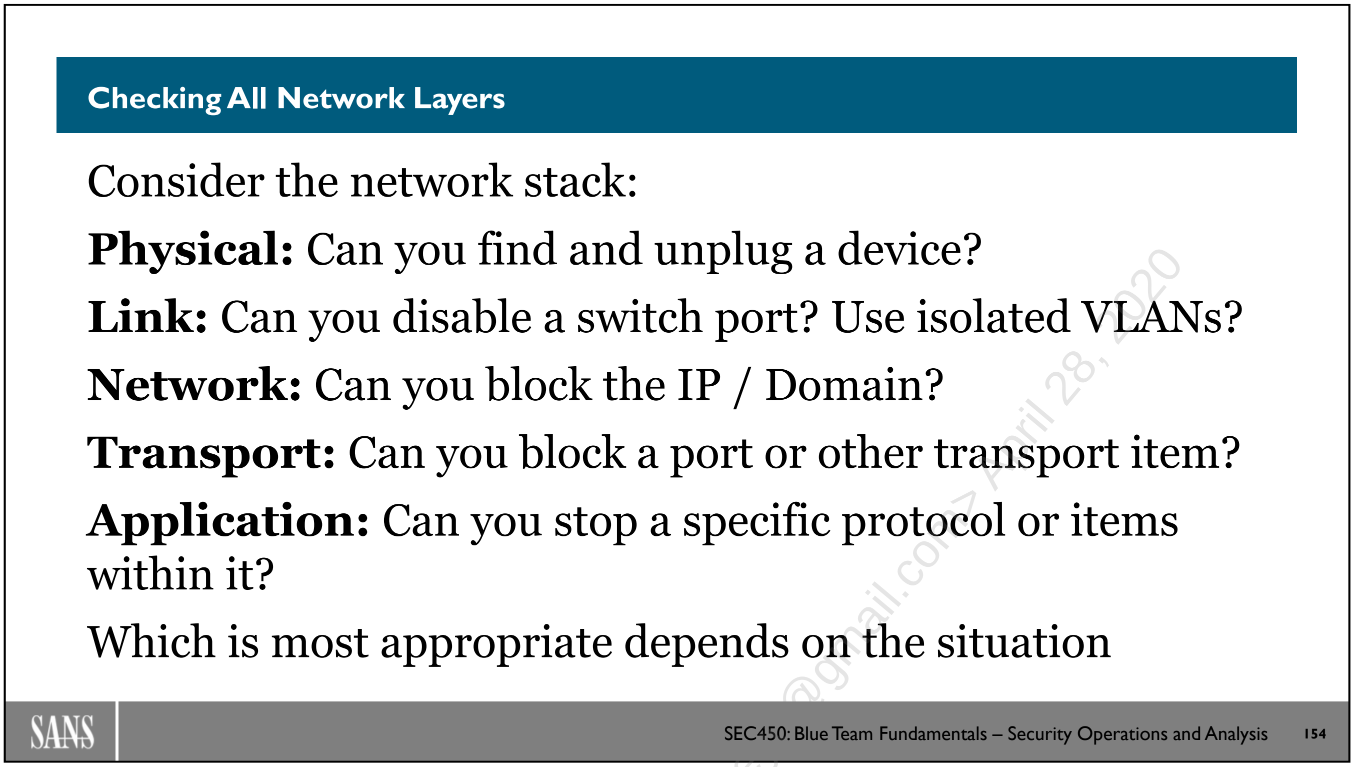
"Four-eyes" rules – in other words, a single person cannot implement a block without at least one other person agreeing that it's a good idea.

Running historical searches in the SIEM for anything you wish to block. In theory, this should prevent any major outages since a site critical for operations would appear in the logs extremely frequently, hinting to the analyst considering blocking it that it isn't a good idea.

Comparison with peer groups/organizations within your country. This method can sometimes work with perceived legal issues. Many companies have lawyers that disagree about what employees are allowed to see in terms of private data. Having experienced this issue in the past, carefully explaining what you need, why you need it, and what it tells you, may clear up these objections. If not, asking peers in your industry how they overcome the problems may help you work out a way to get access to data that may seem to be under privacy restrictions. With TLS encryption becoming the norm on most websites, this type of conversation will only become more important.

What if after all of these you still can't get permission to take the action you want, or see the data you need to perform your job? The best you can do is start to document the cost of the poor process and try to make the case after a while how it slows down your critical response actions. If you can work out a permission request process, work to make it as fast as possible, even if it ends up flooding the other group with requests. I've seen situations in the past where this helped point out the absurdity of asking permission to see data.

Ultimately, if you feel you have a lack of empowerment in your environment, consider it one of the top to-do list items. Without the ability to respond quickly to attacks with decisive action, your organization is merely relying on luck not to be affected by the next big WannaCry or NotPetya-style attack. Although it may take some relationship building, most people should be able to understand how not having the ability to act to do your job can be frustrating. In my experience, having them watch over you should help make the point clear and pave the way to new understanding.



Checking All Network Layers

Consider the network stack:

Physical: Can you find and unplug a device?

Link: Can you disable a switch port? Use isolated VLANs?

Network: Can you block the IP / Domain?

Transport: Can you block a port or other transport item?

Application: Can you stop a specific protocol or items within it?

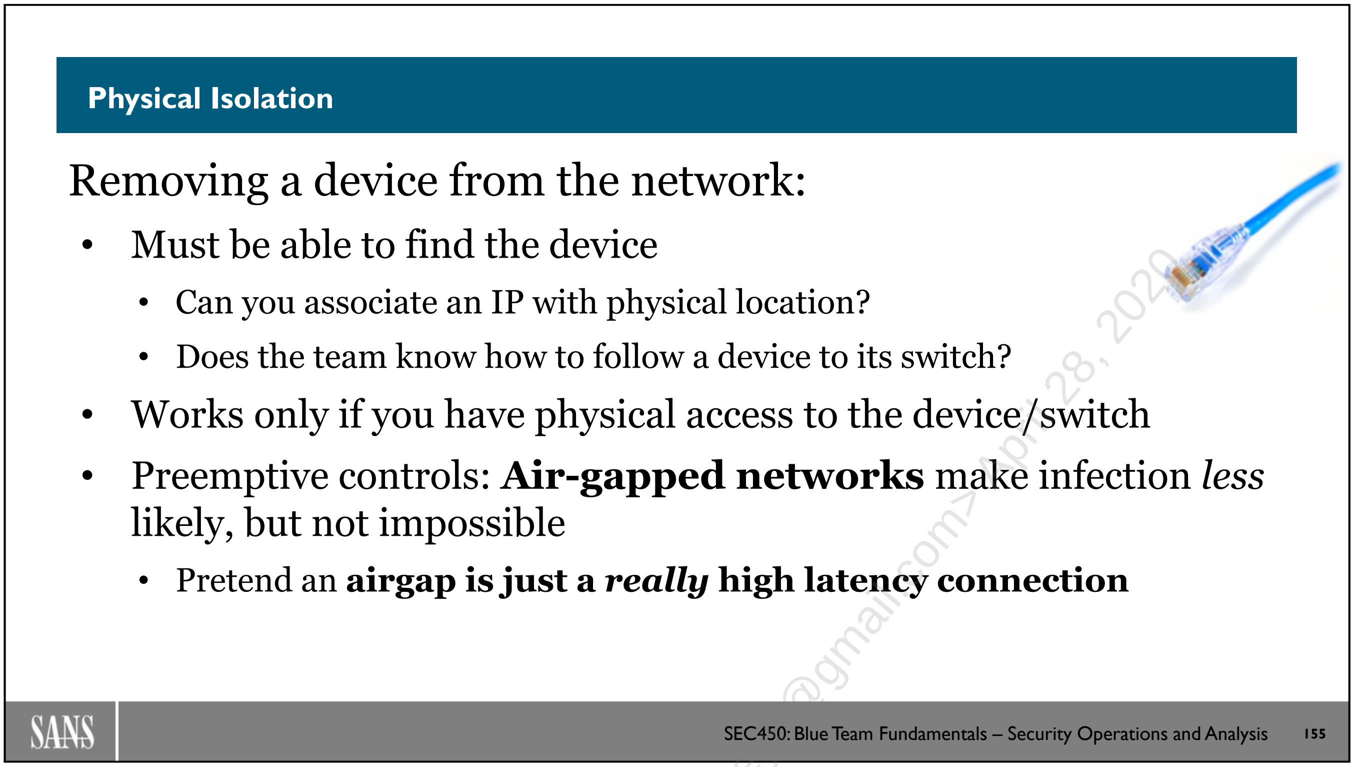
Which is most appropriate depends on the situation

SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 154

Checking All Network Layers

When considering a network-based block, there are multiple ways to go about it. The same way we consider the spectrum of the kill chain for assessing attacks, we should consider the network stack when considering network-based blocks. Though there's a way to stop almost every type of attack, knowing which tool in the toolbox to use is the challenge. Without a clear understanding of what layer an attack is operating on, we may pick the wrong option, leading to a suboptimal containment.

Though we have already discussed specifics about DNS, HTTP, and SMTP protocol in Day 2, we haven't yet covered all the nuances in how to disrupt attacks based on those protocols. In this section, we'll dive a bit deeper into the response options for the most common types of attack and help make it clear when to choose one option over another.



Physical Isolation

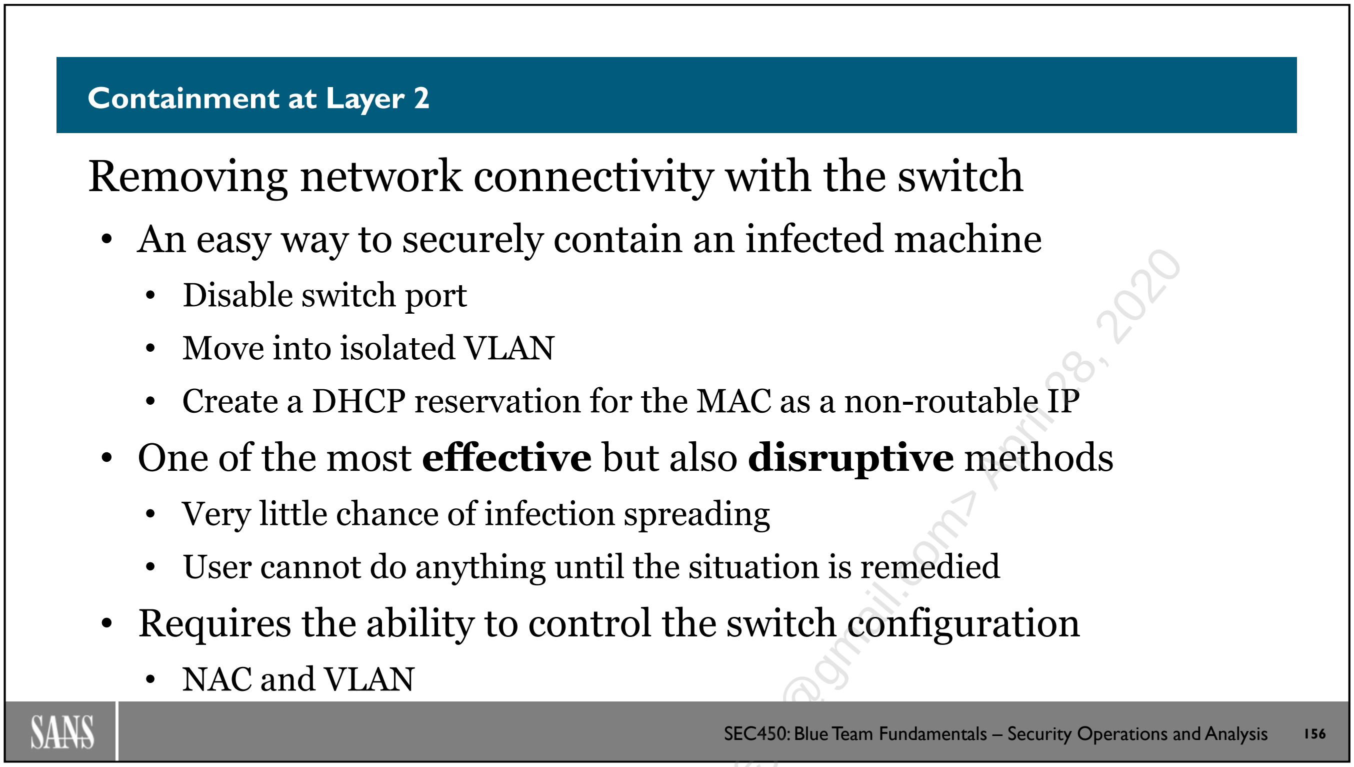
Removing a device from the network:

- Must be able to find the device
  - Can you associate an IP with physical location?
  - Does the team know how to follow a device to its switch?
- Works only if you have physical access to the device/switch
- Preemptive controls: **Air-gapped networks** make infection *less* likely, but not impossible
  - Pretend an **airgap is just a *really* high latency connection**

Physical Isolation

In some situations, infected devices may be able to be removed from the network completely through physically unplugging them. This is the most surefire method of containing an actively infected device and is typically the first step in collecting evidence for forensics as well. The requirement here is that you, or someone you can call, must be physically co-located with the device/switch, which is not always the case. If you *do* have access to all systems on the network, you also must ensure everyone on the team has both the information required to track down a device given its IP address, as well as the procedure and access required to do so. If you do not have a mapping of DHCP scopes to physical locations, labels on jacks to map back to switch ports, access to switches to read CAM tables, NAC or other logs such as DHCP to associate a device with a specific physical location, physical isolation can be difficult.

As a side note, if you do decide to remove an infected device from the network and plan on performing forensics on it, *do not* shut it down. There are lots of volatile items associated with proper forensic analysis that may be destroyed if the system is turn off. Therefore, the best approach is to keep it running as normal but to unplug it from the network. In the case of a laptop that must be shipped somewhere for forensics, using hibernate instead of shutdown can be a better idea because at least the state of RAM is written to the disk which can then later be used for memory forensics. If the system is normally shut down, this useful information will be lost.



Containment at Layer 2

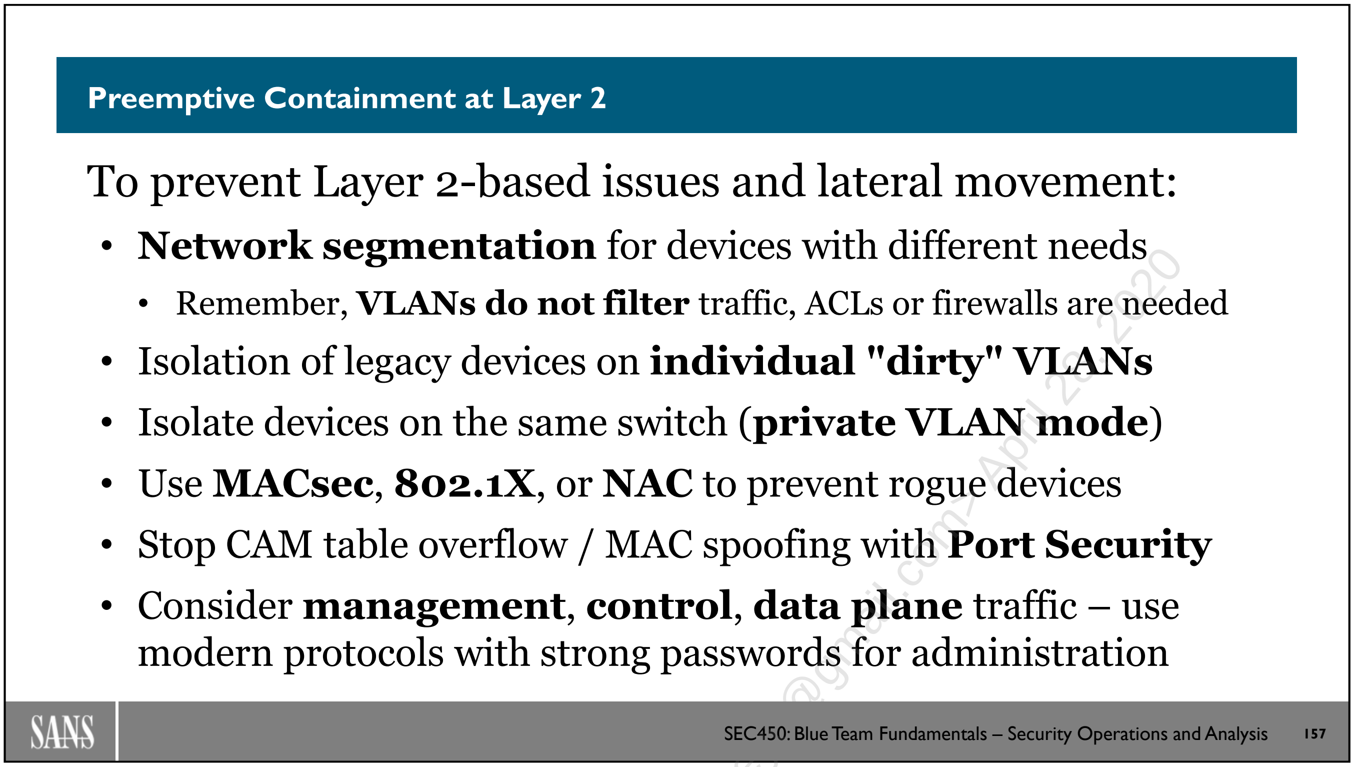
Removing network connectivity with the switch

- An easy way to securely contain an infected machine
  - Disable switch port
  - Move into isolated VLAN
  - Create a DHCP reservation for the MAC as a non-routable IP
- One of the most **effective** but also **disruptive** methods
  - Very little chance of infection spreading
  - User cannot do anything until the situation is remedied
- Requires the ability to control the switch configuration
  - NAC and VLAN

Containment at Layer 2

When you are not interested or cannot physically disconnect an infected device from the network, the next option is to limit access via Layer 2. This method can be equally as effective as physical isolation in most cases, assuming the attacker does not have a way of disrupting the switch. Empowered with the ability to log in to or modify the switch an infected device is connected to, the SOC can effectively fully isolate the device to any degree necessarily from a remote location. Options range from simply disabling the switch port to moving the port onto an isolated VLAN that can only talk with incident response tools and systems. You could also create a DHCP reservation for the MAC address as a non-routable IP address, this wouldn't technically isolate the machine, but would contain many problems.

Layer 2 isolation is one of the more complete options because it is incredibly effective and can be done remotely from any location given the ability to use the power. Since the device is no longer capable of talking with any other device on the network, the chance of increasing damage is minimal. One downside of this method, however, is that it is also maximally disrupting to the end user or service running from the device, it will remain completely out of commission while the response is in progress (potentially a good thing). The other downside is this isolation method relies on controlling a specific switch or Wi-Fi access point, a user who realizes their device is not working may take it somewhere else where the block is not implemented, continuing to expose the network to the threat. While there are preemptive solutions to this problem discussed on the next slide, this can be one reason that using higher network layer blocks can be a better choice in some scenarios where the device is mobile.



Preemptive Containment at Layer 2

To prevent Layer 2-based issues and lateral movement:

- **Network segmentation** for devices with different needs
  - Remember, **VLANs do not filter** traffic, ACLs or firewalls are needed
- Isolation of legacy devices on **individual "dirty" VLANs**
- Isolate devices on the same switch (**private VLAN mode**)
- Use **MACsec**, **802.1X**, or **NAC** to prevent rogue devices
- Stop CAM table overflow / MAC spoofing with **Port Security**
- Consider **management**, **control**, **data plane** traffic – use modern protocols with strong passwords for administration

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 157

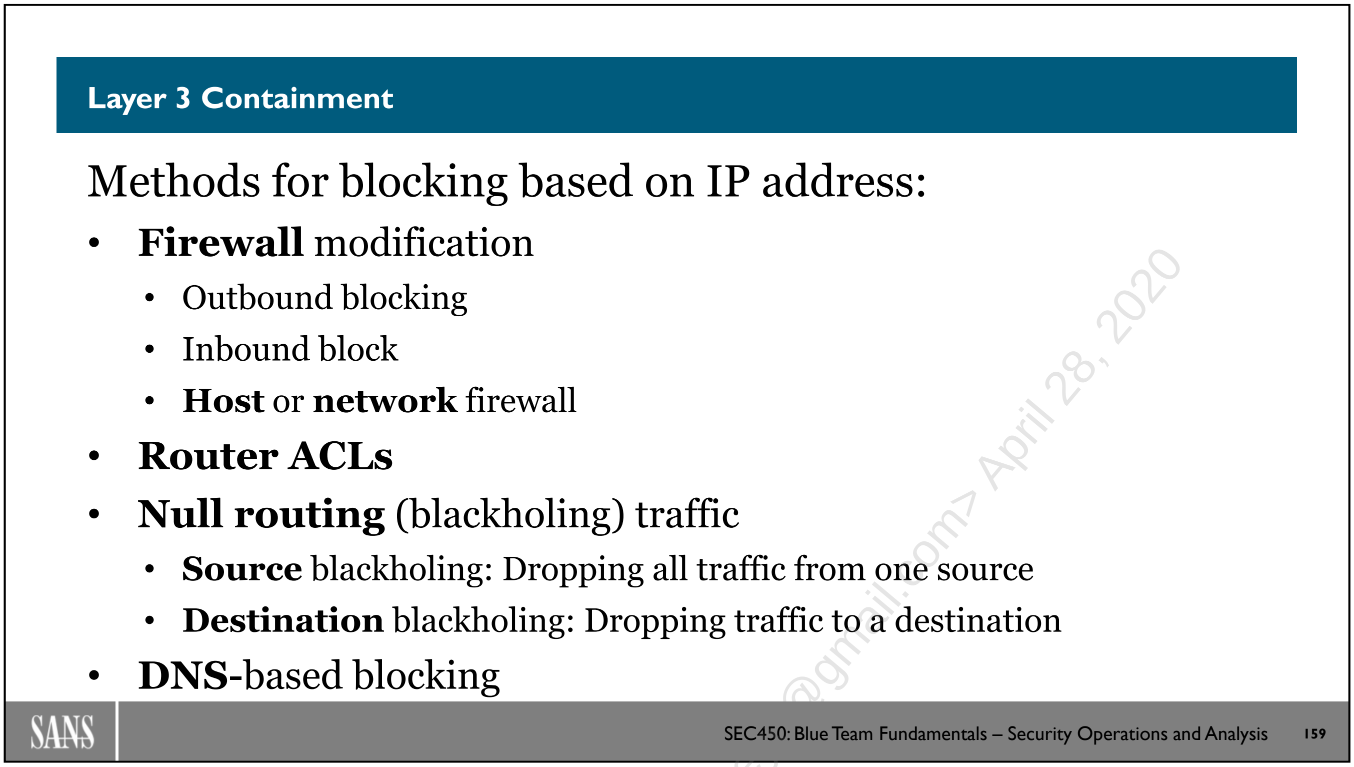
Preemptive Containment at Layer 2

Although being able to detect and stop an attack is important, it's obviously better to design your network to prevent issues from happening in the first place. Here are some industry best practices for network architecture at Layer 2 that will make attacker lateral movement difficult, prevent rogue devices from joining the network, and limit the options for other physical device-based attacks:

- Network Segmentation: This is one of the most important concepts to implement for a modern network. Flat networks fail catastrophically as we've seen with attacks such as NotPetya and WannaCry. Simple deconstruction of the network into zones with firewalls separating the security boundaries is one of the ways we can preemptively keep destruction contained and disrupt attacker's ability to move. This, of course, assumes we have internal firewalls separating our network segments. Remember, VLANs by themselves do not filter traffic, only form separate Layer 3 networks. With the techniques used for modern attacks, we should never assume traffic on the LAN can be trusted. Therefore, even on the smallest network with only a few devices, if there are both clients and servers, traffic should still be divided between servers and clients with filtering rules placed on traffic traversing the two zones.
- Dirty VLANs: For legacy devices with poor security features, a minimum level of security should be implemented by placing the device on *its own* "dirty VLAN" (*not* one dirty VLAN for each legacy device; everyone gets its own) with filtering rules that will prevent it from talking to other devices. Since the threat model for these devices should include a higher likelihood of compromise and use as a command and control point, least privilege principles should be considered a priority.
- Private VLAN Mode: Even with network segmentation, devices within a given zone may attack each other. While network segmentation helps stop one device from bringing down an *entire* organization, it doesn't stop devices from within the *same* zone from attacking each other. Using the Private VLAN (PVLAN) mode of switches allows us to separate all devices on the switches from talking to each other, allowing communication with the port that leads to the gateway only. This is a great defensive posture when such a setting can be enabled, as any worm that is released on the VLAN is unlikely to affect any other device.

- Rogue device prevention through switch-based security features will prevent multiple types of attacks before they can cause a problem. For one, this stops attackers from physically dropping a cellular connected Raspberry Pi or other remote access device on the network; secondly, it prevents users from plugging in their own wireless access points or other devices that may compromise the network. Features such as locking switch ports to a specific MAC address, 802.1X authentication, and Network Access Control are keys to preventing this type of attack.
- Port Security: This feature of switches can be used to lock the maximum number of MAC addresses that can be addressed to a specific physical switch port. By setting this number, CAM table overflow attacks (where attackers send thousands of MAC addresses, forcing the switch to go into hub mode) are prevented, instead disabling the switch if the attack is seen. It can also prevent attempts to snoop on traffic via MAC spoofing attacks.
- Consideration of traffic planes: Each device has three different "planes" of traffic. Each should be auditing to ensure that only up-to-date authentication protocols and versions of those protocols are being used, and extraneous services are disabled.
  - Management Plane: The protocols used to manage the device such as SSH and SNMP.
  - Control Plane: The traffic that controls where the data is sent. For switches, this is Cisco Discovery Protocol (CDP), Spanning Tree (STP), etc. For routers, this is EIGRP, OSPF, and similar.
  - Data Plane: The application and network data sent through the device. For switches, this is frame and for routers it is packets.

For more information on concepts like this, check out the new SANS "SEC530: Defensible Security Architecture and Engineering" class, which does a deep dive into designing security into your network at every layer.



Layer 3 Containment

Methods for blocking based on IP address:

- **Firewall** modification
  - Outbound blocking
  - Inbound block
  - **Host** or **network** firewall
- **Router ACLs**
- **Null routing** (blackholing) traffic
  - **Source** blackholing: Dropping all traffic from one source
  - **Destination** blackholing: Dropping traffic to a destination
- **DNS**-based blocking

SANS
SEC450: Blue Team Fundamentals – Security Operations and Analysis
159

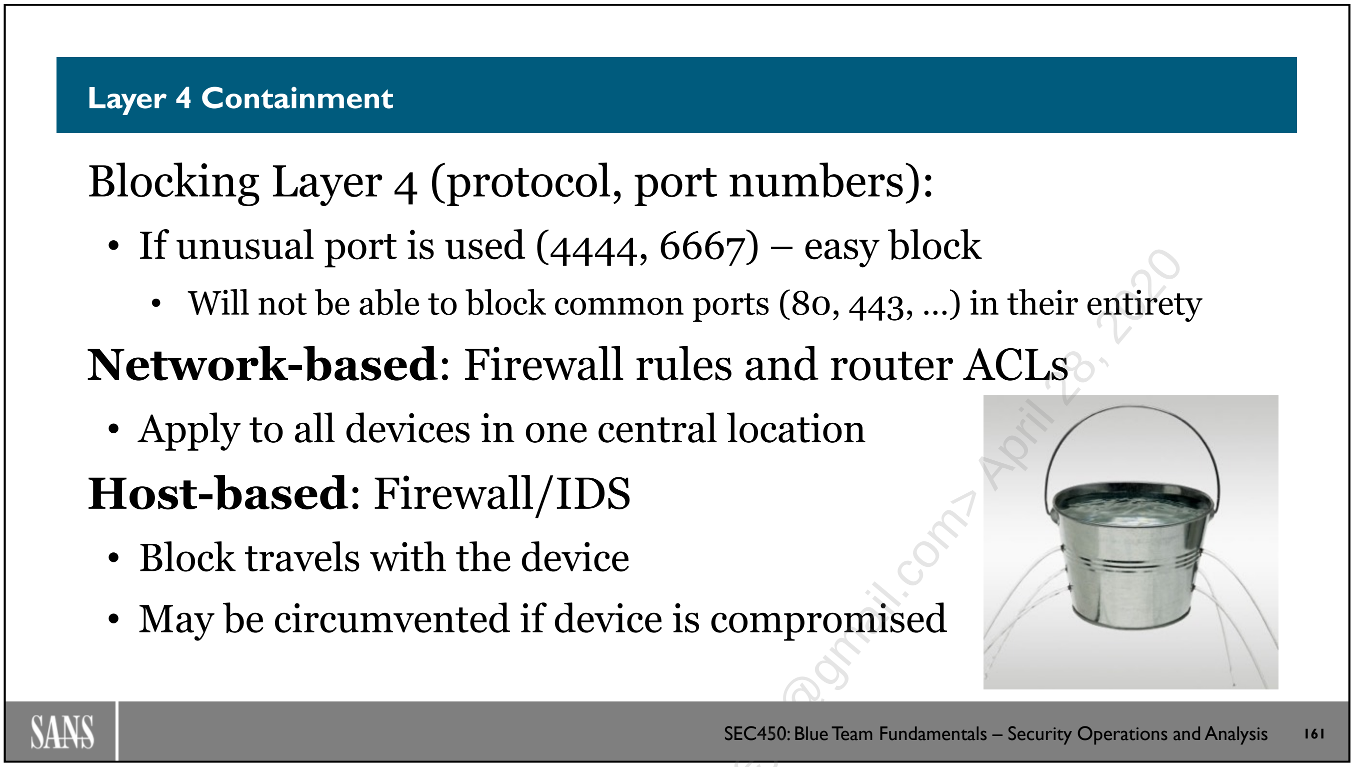
Layer 3 Containment

For containing an infection using Layer 3 devices, we also have several options depending on whether we want to isolate the device, or only stop traffic from a specific destination IP address. Firewalls are the most obvious choice for this type of block as they are perfectly suited to the task. A simple addition of a rule for either all or partial outbound traffic block from an infected internal IP can effectively isolate a device from the internet, but given the device could move or change IP addresses after a new DHCP lease, it might not be the best option for more than a very short-term solution. An inbound block for known malicious sites is a potentially better move since it does not rely on the device location. It *does, however,* rely on you being confident that you have a complete list of IP addresses related to the infection. Malware that uses a domain name with changing IP addresses will not be stopped for long with this method. For these situations, DNS-based blocks are potentially more appropriate.

Firewall-based blocks can be applied at the host-specific or network level; in most cases, both will be necessary. While a network-based firewall block is good because it takes care of *all* devices in the network with one centralized change, it will not protect devices that can roam off the network unless they are forced to VPN back to the network when outside of work. Host-based firewall blocks can also be used to surgically remove the ability for a single device to talk to a malicious destination and can be implemented through inbound or outbound rules. If a specific port or IP address destination is identified, pushing an inbound or outbound firewall block rule to a device's host firewall can be a fast way to stop malicious traffic from occurring. The problem here is that on a compromised device, the attacker may have the ability to turn off or revert the firewall change, thwarting your efforts. A device under attacker control cannot be trusted to implement its own safeguards for more than a short period; this method should only be used for a quick, tactical block. The good news about this type of block is that the host firewall or host IPS-based block travels with the device, protecting it whether it is on or off the network. It is also likely much easier to get permission to modify the firewall of a single device than that of the entire organization.

Another option is to kill traffic by using router ACLs or placing a "blackhole route" for unwanted traffic to the null0 interface, (which is essentially the /dev/null of a router). This method is used for source or destination blackholing to drop any traffic to or from a specific place. Adding blocks via a router may be riskier and more error-prone than using a firewall, so although this method is always an option, most security teams will probably choose the firewall-based Layer 3 blocking methods.

There is one other common option for disruption at Layer 3, and that is blackholing or redirecting traffic through modified DNS responses. We'll talk about this option in more detail in a few pages.



Layer 4 Containment

Blocking Layer 4 (protocol, port numbers):

- If unusual port is used (4444, 6667) – easy block
  - Will not be able to block common ports (80, 443, ...) in their entirety

Network-based: Firewall rules and router ACLs

- Apply to all devices in one central location

Host-based: Firewall/IDS

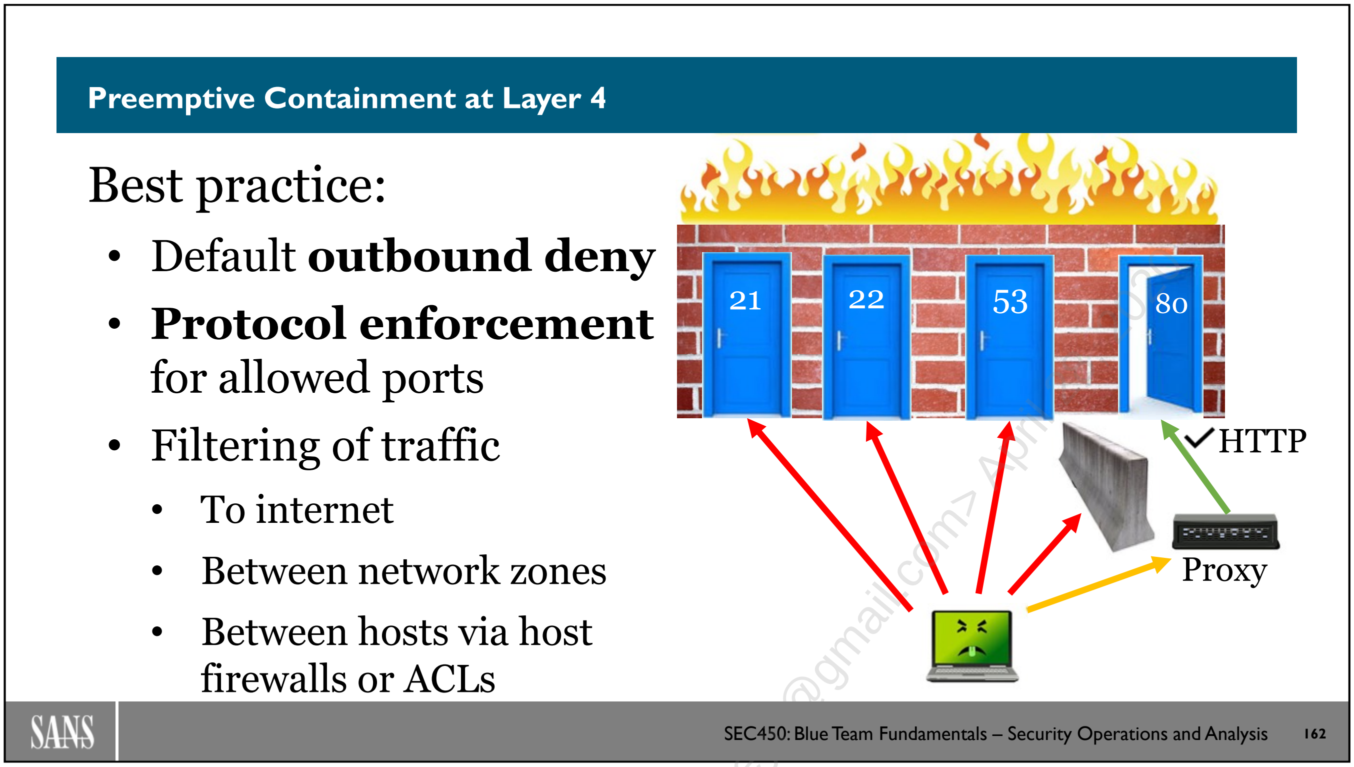
- Block travels with the device
- May be circumvented if device is compromised

Layer 4 Containment

Again, firewalls are another obvious choice for Layer 4-based blocking, but how to apply it will be a bit different than with IP addresses. Almost every organization on earth will have a default inbound deny rule for all port numbers except for the services they offer externally. It is the outbound deny piece that can be a bit more contentious. While it can contain an incident with a tactical block for a specific outbound port, it is *much* better to adhere to a default outbound deny rule for all ports in the first place, at least partially sidestepping this issue altogether (covered on the next page).

Modern routers can also be used to apply Layer 4 blocking based on ICMP types/codes as well as TCP/UDP ports. Inbound and outbound ACLs can be applied to traffic hitting the router that can drop it based on these characteristics, allowing them to effectively act as firewalls. Again, whether analysts will be trusted to change router configurations is another discussion altogether. Most organizations will probably want to keep a very careful watch on router configuration. Changes done in this manner are best achieved through the people who administer the routers to avoid costly mistakes.

Blocks that key off Layer 4 characteristics can easily be implemented on host-based firewalls and IDS solutions as well. Both Linux and Windows come with free, built-in firewalls that make these rules easy to implement via remotely administered commands. These types of blocks carry the same benefits and risks as Layer 3 containment methods. The containment method will travel with the device no matter where it is, but if it is already under attacker control, you may no longer trust it to faithfully apply the blocks as needed, especially if the malware is running with administrative privilege.



Preemptive Containment at Layer 4

Best practice:

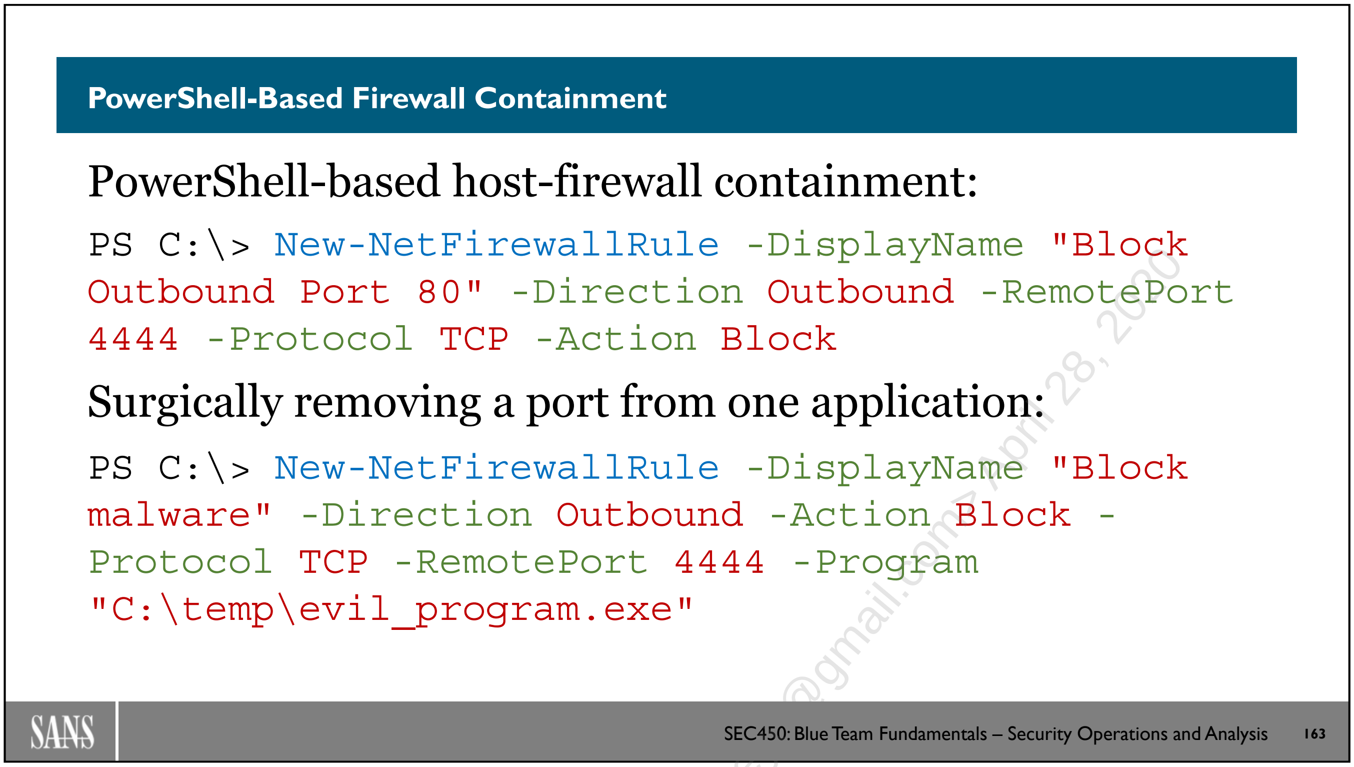
- Default **outbound deny**
- **Protocol enforcement** for allowed ports
- Filtering of traffic
  - To internet
  - Between network zones
  - Between hosts via host firewalls or ACLs

21
22
53
80
✓HTTP
Proxy
SANS
SEC450: Blue Team Fundamentals – Security Operations and Analysis
162

Preemptive Containment at Layer 4

By far, one of the best preemptive security policies you can put in place in your organization is a default deny outbound policy for all ports. Doing this effectively locks all traffic inside the network that doesn't go through pre-approved internet access paths (such as through a proxy). It also has the effect of killing most malware's ability to communicate and has the side benefit of producing a firewall deny when it does fail, alerting you to the malware's presence. Ideally, organizations should only allow the common ports outbound, but not just from any device, only from the source IP address of a proxy, meaning no traffic can directly reach the internet. Additionally, verifying the protocols being used on port 80 with a next-gen firewall can add additional safety since malware can run any protocol they want over port 80, hoping you don't notice. If you do not have a proxy or a default outbound deny rule, a short or even long-term port-based block may be your next best bet, assuming the port is one you know is not in use for any other reason. Ports like 4444 (meterpreter) and 6667 (IRC) should be easy to block on the whole for all devices for the long term without consequence. Malware that shares a port with good traffic may not be as easily blocked with this method unless it is targeted to a specific device.

This same concept *can* work for host firewalls as well, but outbound deny rules are not nearly as often due to the complication involved in getting the policy right for each device. At a minimum, we suggest low-volume tactical logging for inbound and outbound traffic on each device. Having this information can be an enormous help in chasing down lateral movement after an incident has occurred. It is, in effect, free internal traffic visibility.



PowerShell-Based Firewall Containment

PowerShell-based host-firewall containment:

```
PS C:\> New-NetFirewallRule -DisplayName "Block
Outbound Port 80" -Direction Outbound -RemotePort
4444 -Protocol TCP -Action Block
```

Surgically removing a port from one application:

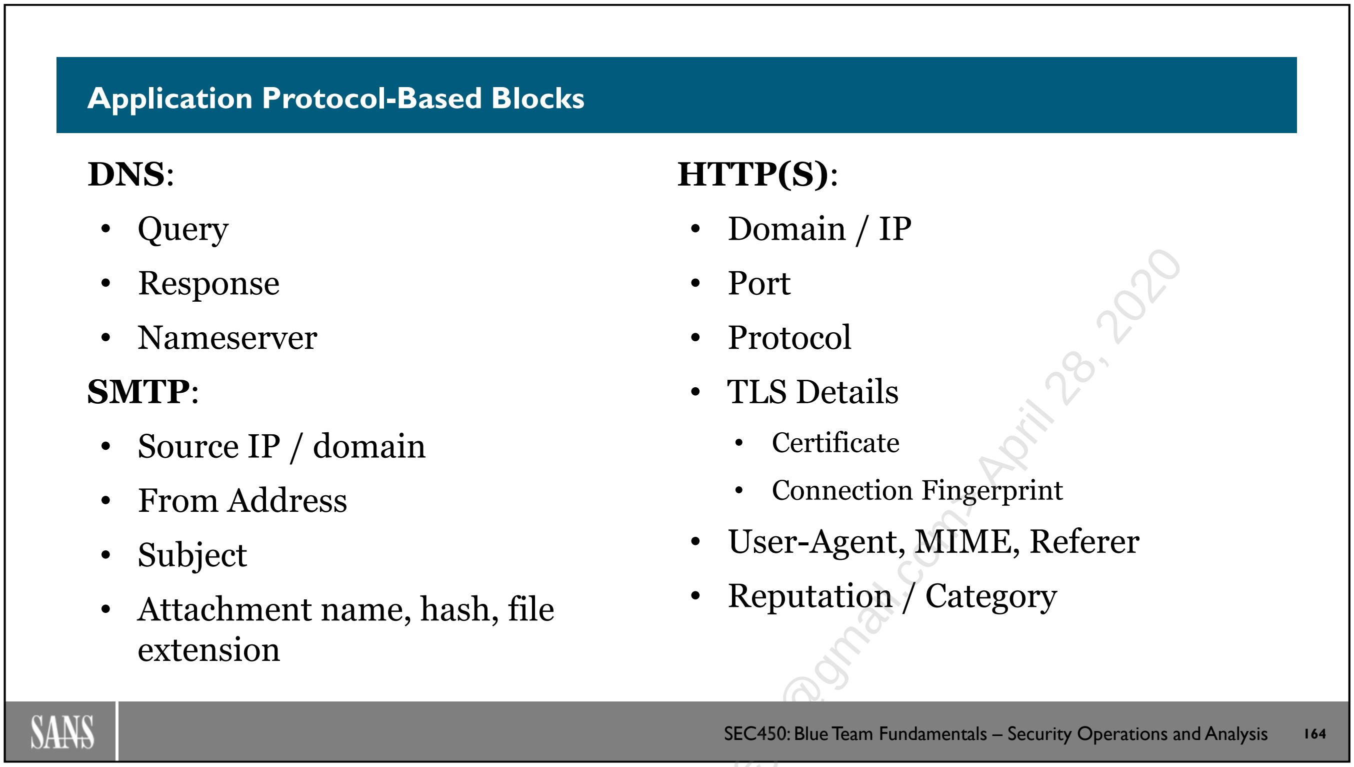
```
PS C:\> New-NetFirewallRule -DisplayName "Block
malware" -Direction Outbound -Action Block -
Protocol TCP -RemotePort 4444 -Program
"C:\temp\evil_program.exe"
```

SEC450: Blue Team Fundamentals – Security Operations and Analysis 163

PowerShell-Based Firewall Containment

Here are some example PowerShell commands that can be used (assuming you can run PowerShell remotely) to kick off tactical blocking of traffic at Layer 4. The first command blocks all outbound traffic to destination port 4444 from the host. The second commands would block it only for a single application. This method is what is required if the malware were to be using a common port like 80, for example. You likely do not want to block port 80 for the entire machine but given the ability to control the host-based firewall, you can put in the surgical containment of a single process running on the machine, a unique advantage of using a host-based block.

[1] https://docs.microsoft.com/en-us/powershell/module/netsecurity/new-netfirewallrule?view=win10-ps



Application Protocol-Based Blocks

DNS:

- Query
- Response
- Nameserver

SMTP:

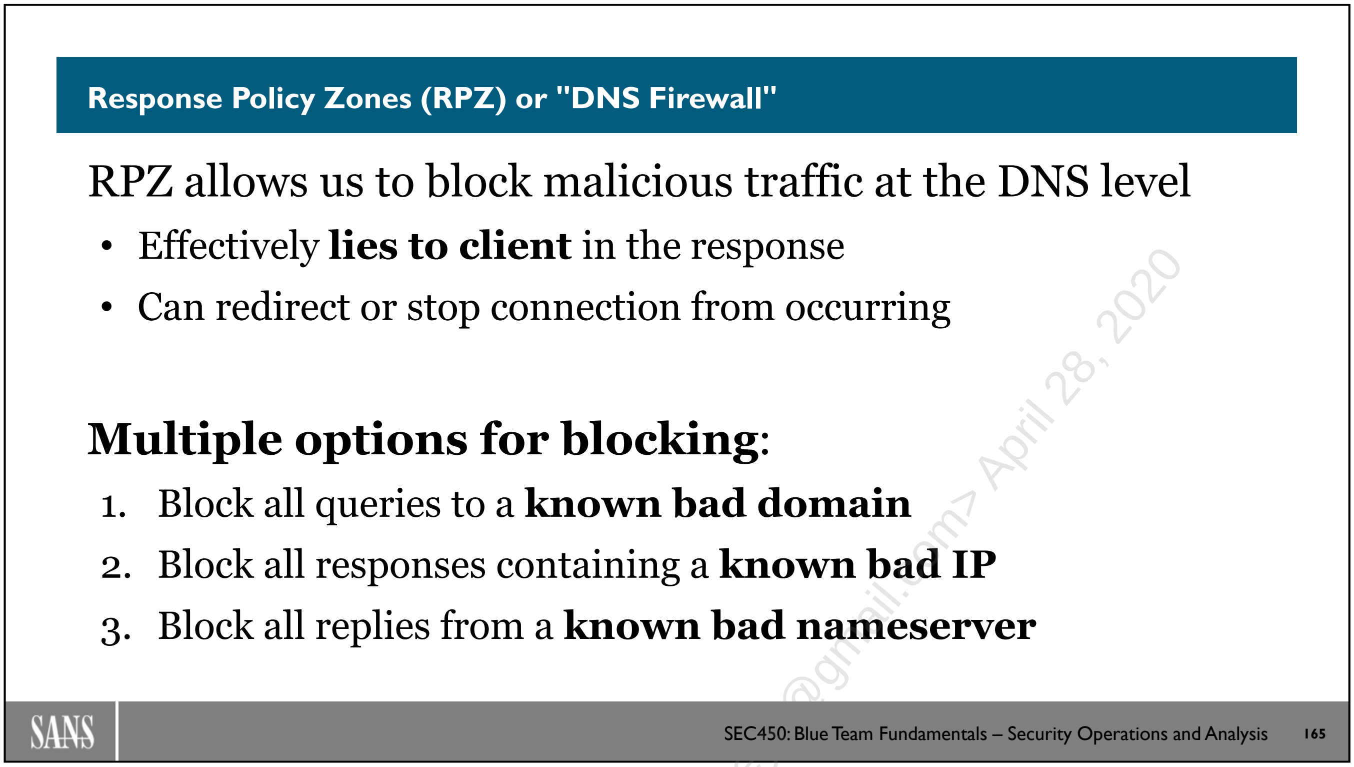
- Source IP / domain
- From Address
- Subject
- Attachment name, hash, file extension

HTTP(S):

- Domain / IP
- Port
- Protocol
- TLS Details
  - Certificate
  - Connection Fingerprint
- User-Agent, MIME, Referer
- Reputation / Category

Application Protocol-Based Blocks

When it comes to blocking activity at the higher layers of the networking stack, understanding the various ways to block DNS, SMTP, and HTTP traffic is the name of the game. For each of these protocols, consider the fields of interest that we use to find evil and how a block could be implemented for each of them. While Layer 3 and 4 blocks can easily be performed with a firewall, specifically blocking protocols based on requests and response field values can get a little more difficult. For some types of attacks, however, that is exactly what will need to happen. In DNS for example, you may want to block a domain name, but if a domain generation algorithm is used, that might only work for a short period. Perhaps you then switch to changing the block to the IP address, but if the malware is part of a fast-flux botnet where the IP addresses are changing, this may not work either. Clearly, there are some complicating factors here.



Response Policy Zones (RPZ) or "DNS Firewall"

RPZ allows us to block malicious traffic at the DNS level

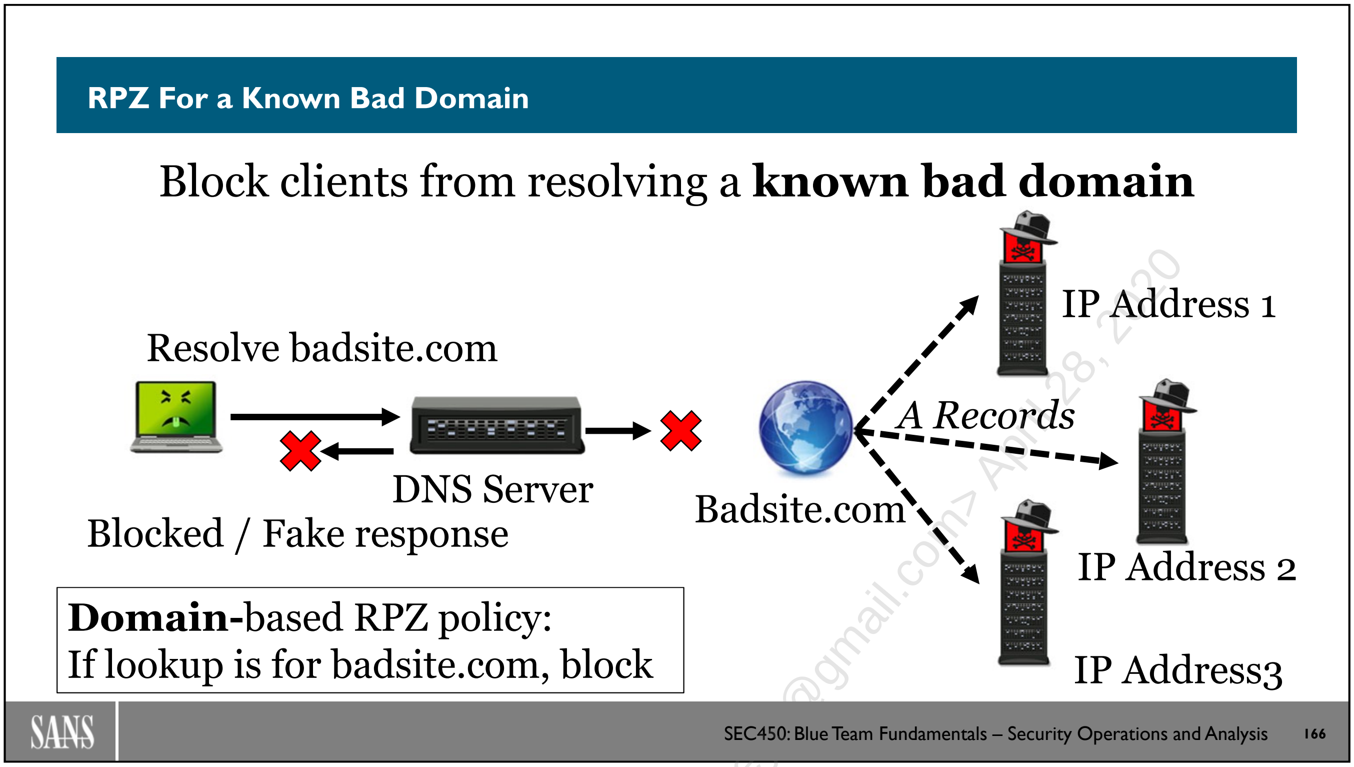
- Effectively **lies to client** in the response
- Can redirect or stop connection from occurring

Multiple options for blocking:

1. Block all queries to a **known bad domain**
2. Block all responses containing a **known bad IP**
3. Block all replies from a **known bad nameserver**

Response Policy Zones (RPZ) or "DNS Firewall"

A Response Policy Zone (RPZ), sometimes referred to as a DNS firewall, is a great go-to way of blocking access to known bad infrastructure. While blocking something with a traditional firewall based on IP *could* work, killing the whole domain, regardless of the IP that hosts it is a much more robust solution. There are other situations beyond a simple domain block a DNS firewall can cover as well, though. To fully utilize their capabilities, we first must understand what these situations are and when they apply. There are three main situations where an RPZ can be used—when you have a known bad domain name to block, when there is a known bad IP address to block, or when all the domains, regardless of IP, being resolved by the same nameserver should be blocked.



RPZ For a Known Bad Domain

Block clients from resolving a **known bad domain**

Resolve badsite.com

DNS Server

Blocked / Fake response

Badsite.com

A Records

IP Address 1

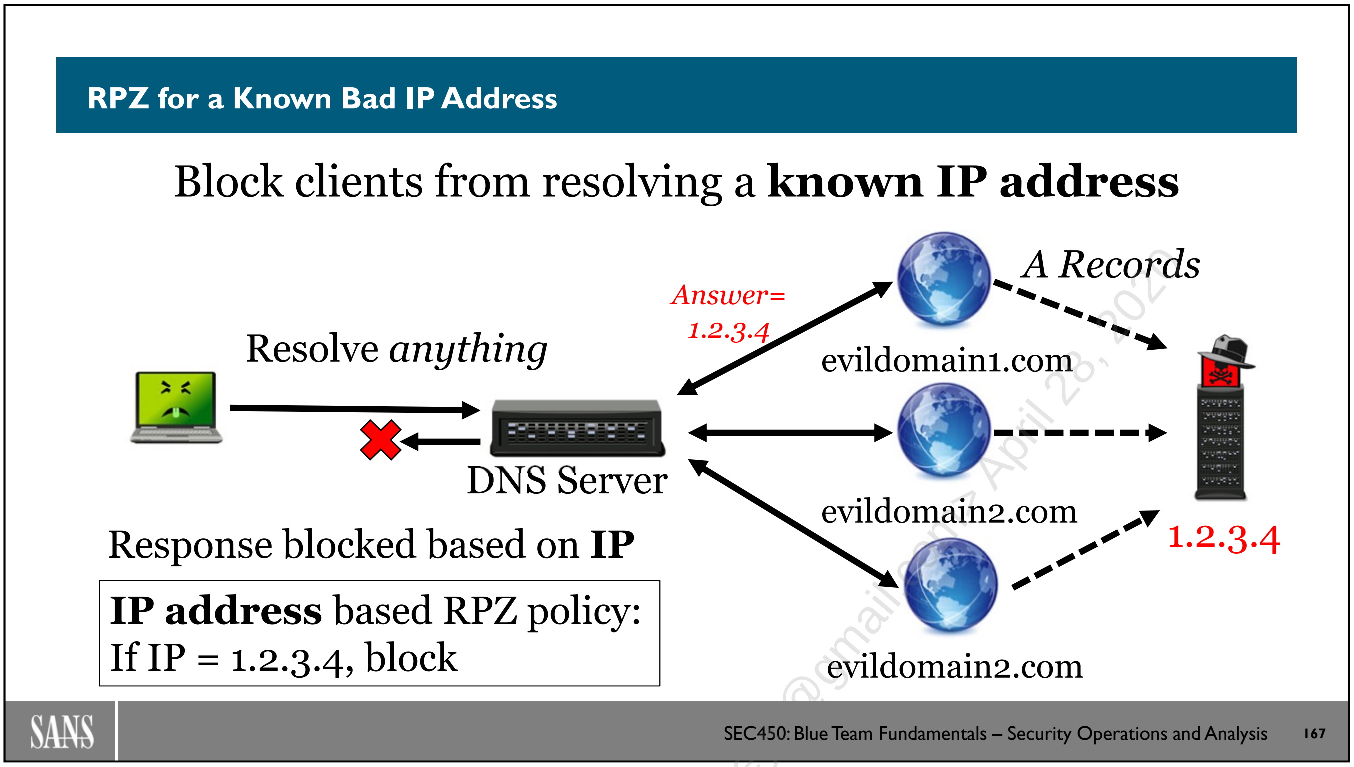
IP Address 2

IP Address3

Domain-based RPZ policy:
If lookup is for badsite.com, block

RPZ For a Known Bad Domain

One option for blocking with DNS is when you know the name of a specific domain that is evil. In this situation, you can make an RPZ policy that says any time a client attempts to look up that domain, instead of performing the lookup, provide a specified response. In the slide above, the SOC has identified an infected device that is trying to contact badsite.com, which could be hosted on three different IP addresses on the web (the attackers may do something like this for redundancy). Instead of having to block three separate IP addresses, a single RPZ policy can be put in on the server to say any time someone tries to resolve badsite.com, return the answer 0.0.0.0 (or any other response we choose). Note in this slide the DNS server is meant to signify the internal organization DNS server where the RPZ policy is implemented, not the attacker-controlled DNS nameserver.



RPZ for a Known Bad IP Address

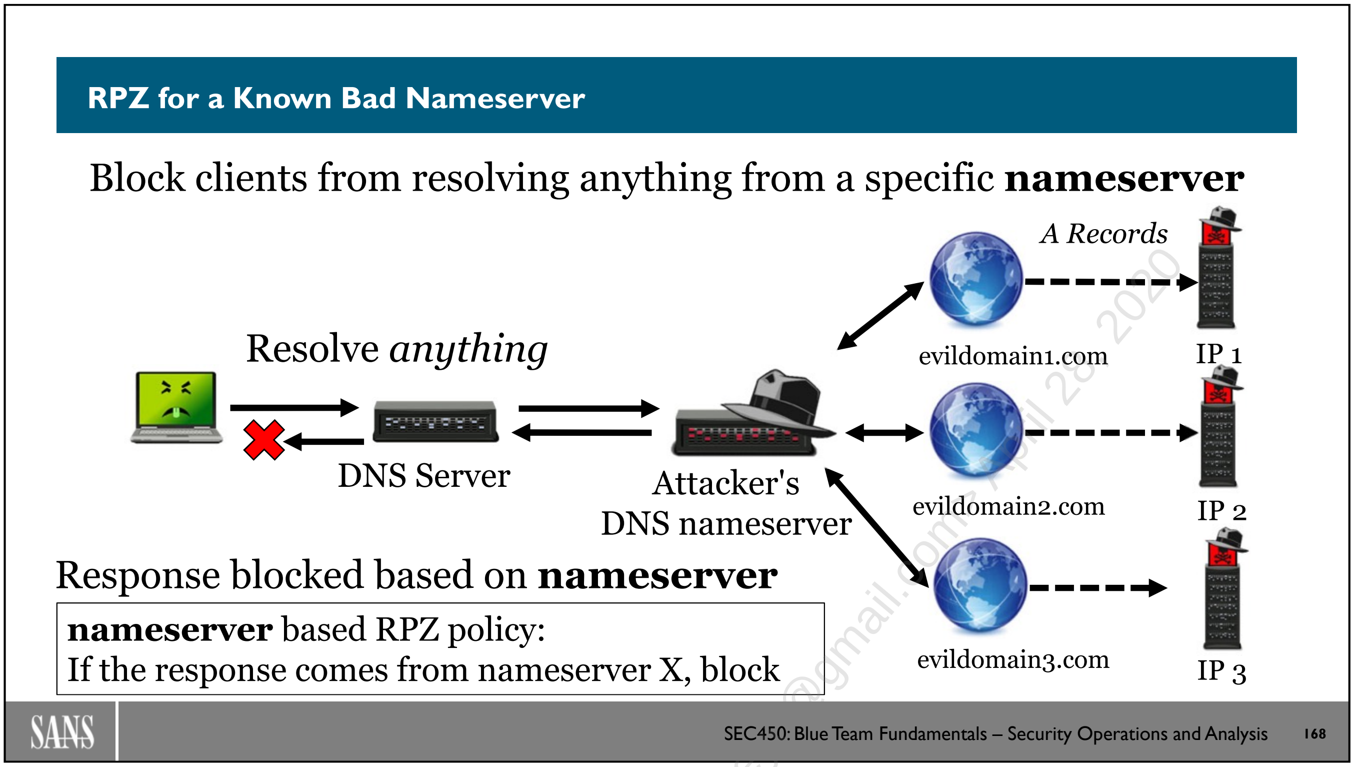
Block clients from resolving a **known IP address**

Response blocked based on **IP**

IP address based RPZ policy:
If IP = 1.2.3.4, block

RPZ for a Known Bad IP Address

An alternative situation for DNS-based blocking is that you may not know which or how many domains are involved in the malicious infrastructure, but you *do* have the information to say that certain IP addresses are bad. For example, say you've seen two infected devices communicating with evildomain1.com and evildomain2.com, but have reason to believe that there are likely more domains. Although you don't know how many other domains there are, you do know that both evildomain1.com and evildomain2.com both resolved to IP address 1.2.3.4. Therefore, regardless of the domain name someone looks up, you are interested in blocking all traffic to/from 1.2.3.4. You can do this with a firewall, but RPZ can be used for this as well with an IP address policy that says "do not return the answer for any query that returns an answer of 1.2.3.4." Note that while the domain-based policy can act based on the query packet alone, this policy relies on the response before it can act, so the DNS traffic will still occur.



RPZ for a Known Bad Nameserver

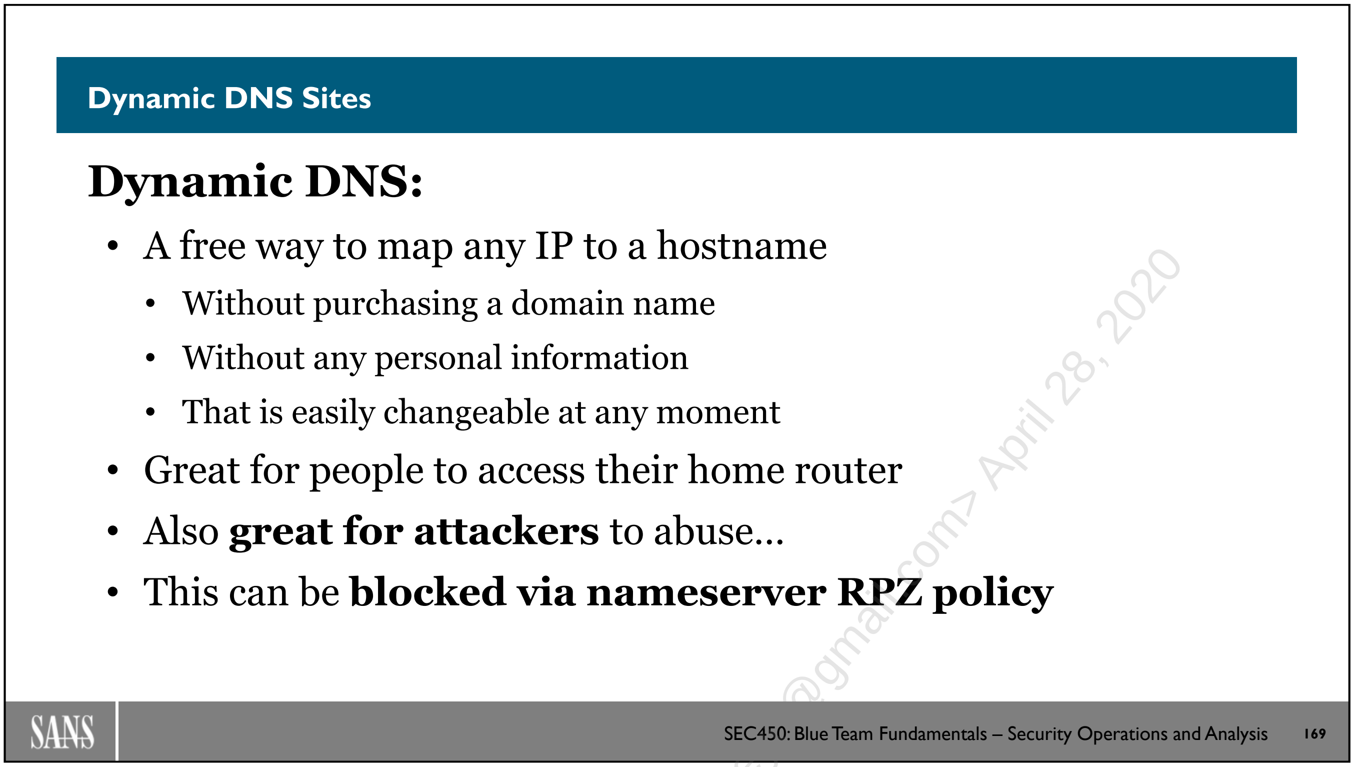
Block clients from resolving anything from a specific **nameserver**

Response blocked based on **nameserver**

nameserver based RPZ policy:
If the response comes from nameserver X, block

RPZ for a Known Bad Nameserver

A third situation is when there are both multiple IP addresses and multiple domain names in use by the attacker. We can accommodate this situation through RPZ policy as well, but the catch is all the domains must use the same (or a set of known) nameservers. The policy, in this case, can be for *your* DNS server to block anything that gets answered from the known bad attacker DNS nameservers. Since your DNS server knows where the lookup responses came from both by name and source IP address, you can choose either way.



Dynamic DNS Sites

Dynamic DNS:

- A free way to map any IP to a hostname
  - Without purchasing a domain name
  - Without any personal information
  - That is easily changeable at any moment
- Great for people to access their home router
- Also **great for attackers** to abuse…
- This can be **blocked via nameserver RPZ policy**

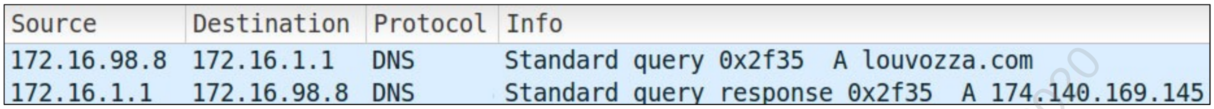
Dynamic DNS Sites

Where can we best use a nameserver-based block? There's at least one great condition. To block an entire nameserver's worth of sites, you must be confident that the nameserver is only servicing requests for sites that are evil, or at least sites that are not business critical. One of the most common use cases that meet such a condition is dynamic DNS services. You've probably seen or used dynamic DNS services in the past if you've ever set up a hostname for your router at home so that you can access your home internet without having to find your current IP address. These are services like DynDNS, No-IP, Duck DNS, and others. While these are great free services for home users to create ways to connect back to their IP at home, malware also uses them for command and control since they are free and easy to change.

Due to the nature of these services, almost no businesses and other legitimate, important sites will be hosted using them. Therefore, blocking the nameservers that service these hostnames will not only be unlikely to affect anyone in a way that affects business but also has the effect of ridding you of the risk of compromise via dynamic DNS-hosted websites.

Sinkhole Redirection

RPZ can also **redirect client to an internal sinkhole**

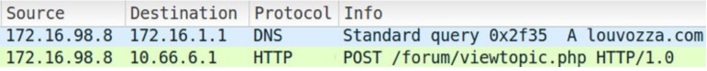


```
Source        Destination   Protocol  Info
172.16.98.8   172.16.1.1    DNS       Standard query 0x2f35  A louvozza.com
172.16.1.1    172.16.98.8   DNS       Standard query response 0x2f35  A 174.140.169.145
```

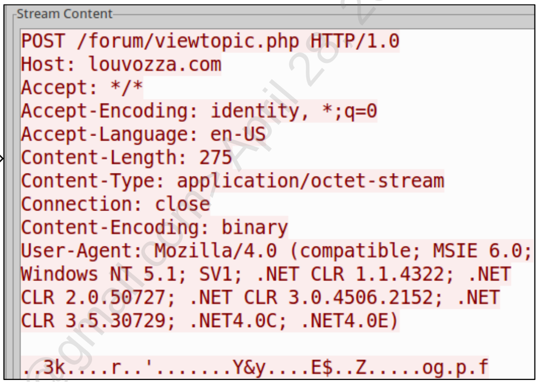
DNS only

Which is more useful…

DNS + full HTTP



```
Source       Destination  Protocol  Info
172.16.98.8  172.16.1.1   DNS       Standard query 0x2f35  A louvozza.com
172.16.98.8  10.66.6.1    HTTP      POST /forum/viewtopic.php HTTP/1.0
```



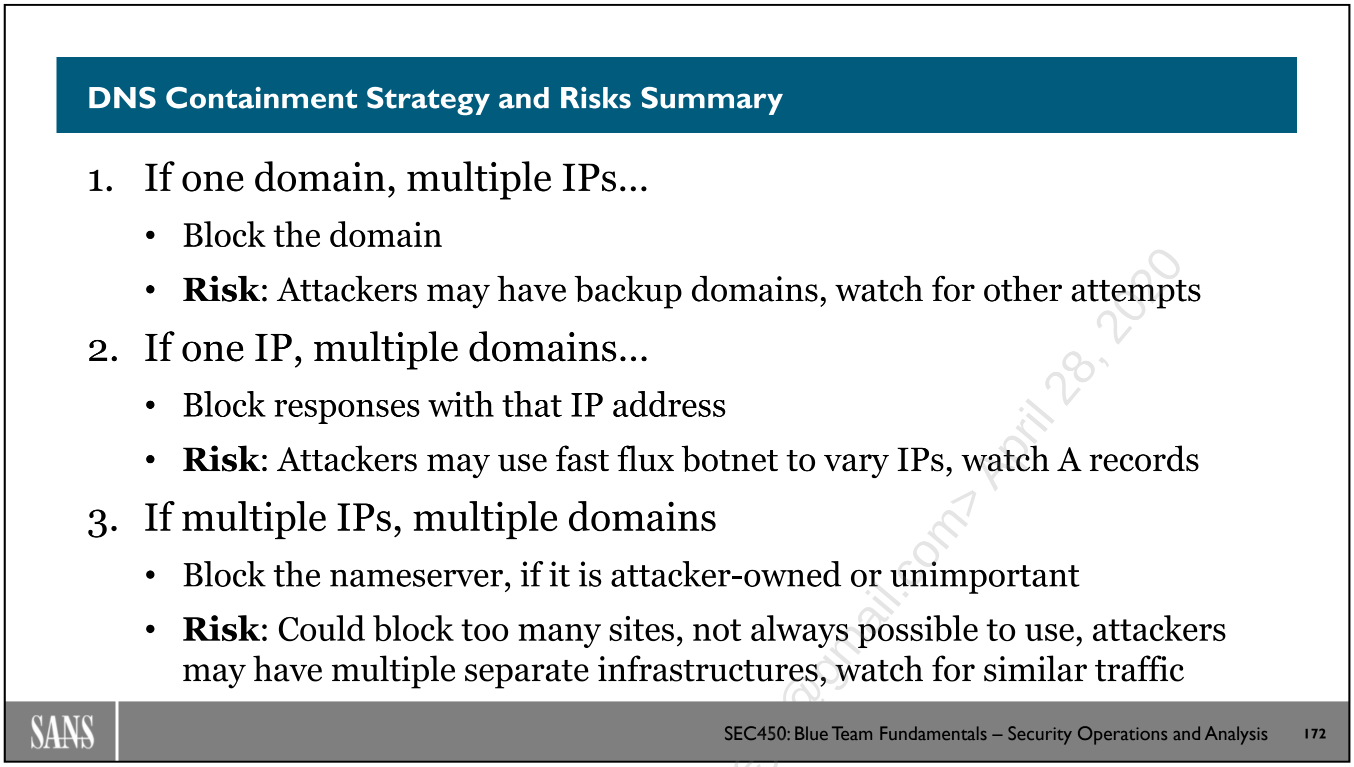
```
POST /forum/viewtopic.php HTTP/1.0
Host: louvozza.com
Accept: */*
Accept-Encoding: identity, *;q=0
Accept-Language: en-US
Content-Length: 275
Content-Type: application/octet-stream
Connection: close
Content-Encoding: binary
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0;
Windows NT 5.1; SV1; .NET CLR 1.1.4322; .NET
CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET
CLR 3.5.30729; .NET4.0C; .NET4.0E)

..3k....r..'.......Y&y....E$..Z.....og.p.f
```

Sinkhole Redirection

Another extremely useful tactic is instead of just blackholing a DNS request to 0.0.0.0, using RPZ for redirection to an internal listening server. In the slide above, the top picture shows a normal DNS request and the information we would receive from it. If we used RPZ to just blackhole it, instead of the IP 174.140.169.145 coming back as the response, the DNS server would "intercept" it and return 0.0.0.0. Although this effectively would block louvozza.com, doing this only gives us the information that 172.16.98.8 requested a DNS address for known bad domain. It doesn't tell us what it would have done had the IP resolved.

Instead of resolving the request to 0.0.0.0, we can instead set up an internal device listening on several common ports that can help give us more information in situations like this. Let's say we set up an internal listening sinkhole to IP address 10.66.6.1 and instead have RPZ redirect the traffic there. In this scenario, here's what would happen: The DNS request would go out for louvazza.com as shown in the bottom left photo, but the response would come back saying the site was located at 10.66.6.1 since the RPZ policy intercepted the request. Then, instead of trying to connect to 0.0.0.0 and that being the end, the infected device can now reach out to the device at 10.66.6.1. We will now see if it's trying to make a TCP, UDP or other connection, which port it's using, and, if it's using a well-known protocol, we can set up a fake service to "service" the request. Let's assume we've recorded a PCAP to our internal sinkhole and set up a web server assuming the malware will likely talk HTTP. Now what we would see is not only the port and protocol, but the information shown in the bottom right box. This shows the malware makes a POST request, includes all the HTTP headers such as a User-Agent we can use to search across the network for other infected devices, as well as the content of the POST request (which in this case seems to be only encrypted bytes). It gives us *much* more useful information that the blackholed request would alone, we now know Layer 4 and 7 information that can be further used to target the malware. This is the power of the RPZ + an internal listening sinkhole. The key piece of this technique is that the internal listening sinkhole must have a fake service running on the port the malware attempts to use. Tools like INetSim are made exactly for this purpose.[1] INetSim will listen on multiple popular ports and record a service log of everything that was said to it for use in this exact scenario. What if your malware is using a weird port that INetSim isn't prepared to accept? You can either modify the setup to listen on that port as well or, in a pinch, use netcat to listen on any port and accept whatever traffic is sent, printing it to the screen for your interpretation.

[1] https://www.inetsim.org/



DNS Containment Strategy and Risks Summary

1. If one domain, multiple IPs…
   - Block the domain
   - **Risk**: Attackers may have backup domains, watch for other attempts
2. If one IP, multiple domains…
   - Block responses with that IP address
   - **Risk**: Attackers may use fast flux botnet to vary IPs, watch A records
3. If multiple IPs, multiple domains
   - Block the nameserver, if it is attacker-owned or unimportant
   - **Risk**: Could block too many sites, not always possible to use, attackers may have multiple separate infrastructures, watch for similar traffic

SANS

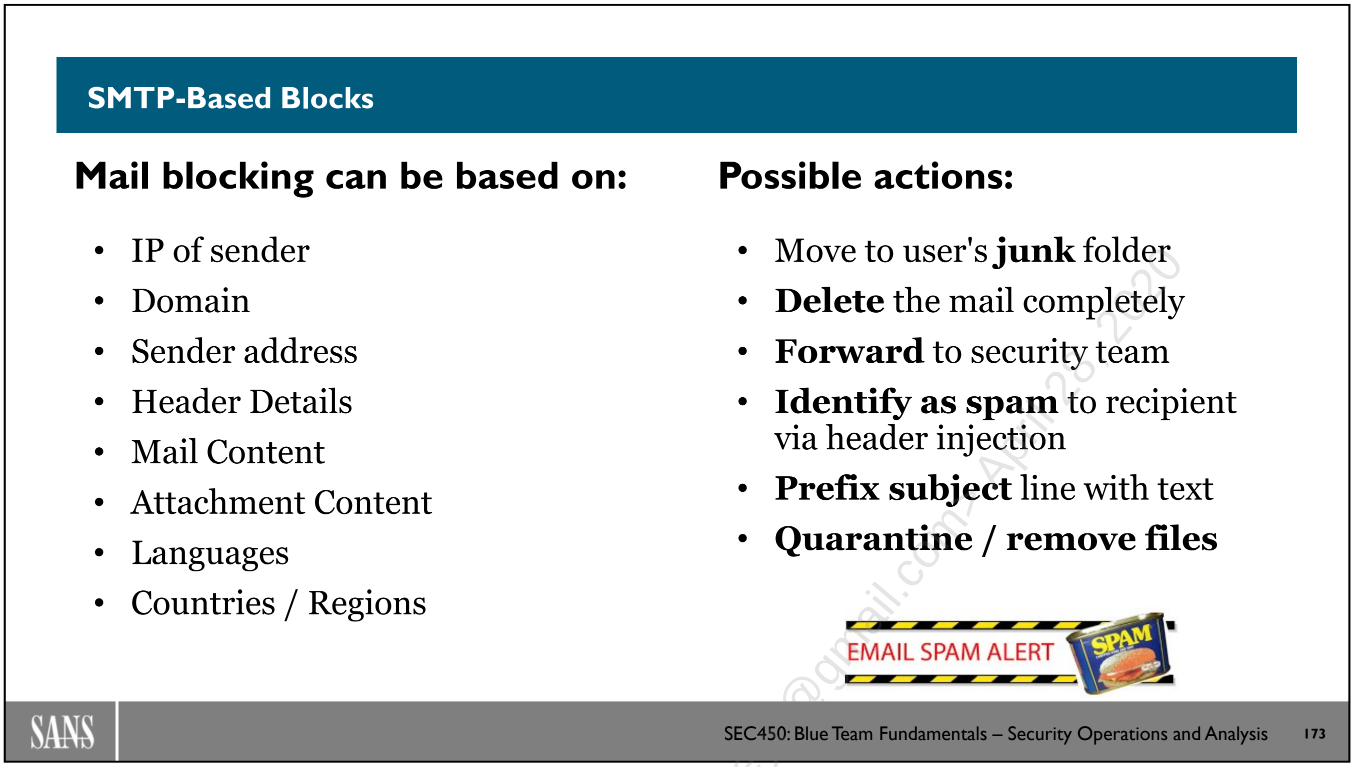
SEC450: Blue Team Fundamentals – Security Operations and Analysis 172

DNS Containment Strategy and Risks Summary

Given all these scenarios, here are the high-level rules to keep in mind:

- If you know a domain, block the domain.
- If you know an IP, block the IP in any response.
- If both are changing, you may need to rely upon a nameserver-based block if possible.

Of course, all these techniques rely upon you having complete information about what is being used. You may think you've blocked the only domain only to find that the malware falls back to a backup option that was programmed into the code once it finds it can't contact the primary site. For this reason, the activity from a host post-blocking action must still be watched, and you should reference OSINT / threat intel information to find if there are any other known IP addresses or domains associated with the information you do have. Doing this can prevent the situation where you think you have containment in place only to find out shortly after that it wasn't, in fact, effective.



SMTP-Based Blocks

Mail blocking can be based on:

- IP of sender
- Domain
- Sender address
- Header Details
- Mail Content
- Attachment Content
- Languages
- Countries / Regions

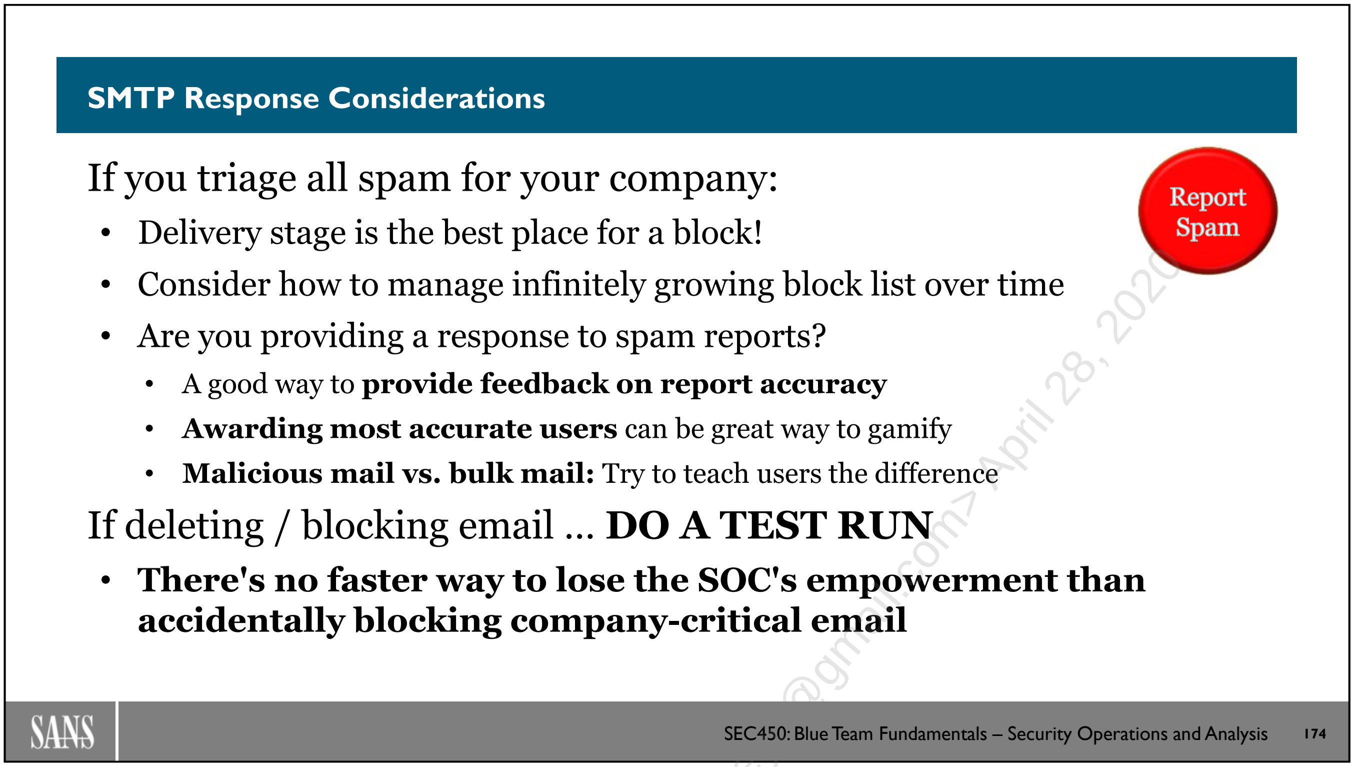
Possible actions:

- Move to user's **junk** folder
- **Delete** the mail completely
- **Forward** to security team
- **Identify as spam** to recipient via header injection
- **Prefix subject** line with text
- **Quarantine / remove files**

SMTP-Based Blocks

When containing an email-based attack, your SMTP server will determine the flexibility with which you can apply blocks. Most modern mail systems should have the ability to block senders' names, domains, IP addresses, header content, and more. The capability of blocking based on the email body is extremely important as well as many times spam waves will come from various addresses without any of the network-level indicators in common. If instead you can block or delete email based on the filename of an attachment or the presence of a certain a link or text, you will be in a much better position to remove spam with laser-like precision.

Email is not exactly like packets, which are either flat out rejected or not. Once you have identified the characteristics of the suspicious email, you'll have a set of options for actions to take. The obvious options are to immediately delete it, categorize it as junk, or inject a header to make the recipient's email client label it as malicious/spam. Other common options include either quarantining the entire email or removing the file attachments, making the user jump through hoops if they truly want to get to it. An alternative and potentially more useful and tactical option is forwarding identified malicious email to a SOC-controlled email address for analysis and trending over time. Depending on your level of certainty and the risk involved, you can strategically pick one of these options that balance security with the desire not to disrupt legitimate email.



SMTP Response Considerations

If you triage all spam for your company:

- Delivery stage is the best place for a block!
- Consider how to manage infinitely growing block list over time
- Are you providing a response to spam reports?
  - A good way to **provide feedback on report accuracy**
  - **Awarding most accurate users** can be great way to gamify
  - **Malicious mail vs. bulk mail:** Try to teach users the difference

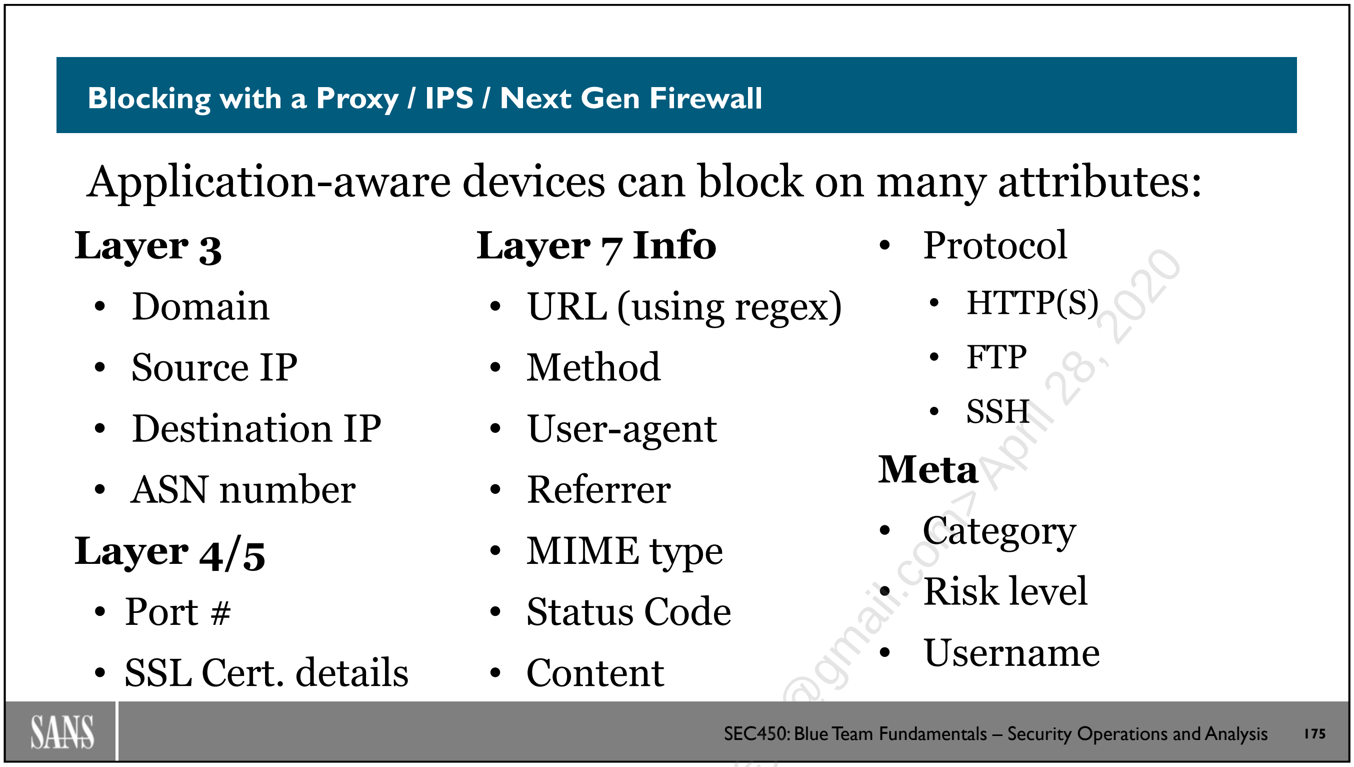
If deleting / blocking email … **DO A TEST RUN**

- **There's no faster way to lose the SOC's empowerment than accidentally blocking company-critical email**

SMTP Response Considerations

Email is a business-critical system, so taking risks with it is not advised. On the other hand, blocking an attack at the delivery stage is the ideal method since the attacker never even begins to gain a foothold in the environment. If you can apply email-blocking rules, reach into inboxes and remove things, or are on the receiving end of your company's "report spam" button, here are some things to consider.

- Responding to user spam reports can be a good way to train users as well as improve accuracy. Some commercial spam triage tools will even automate the responses based on how the email was triaged and track the accuracy of spam reports of users over time. Giving a monthly prize to the most accurate spam reporter can be a great way to encourage sending *real* malicious email, and not the non-malicious bulk email junk that many users think should be reported with that button.
- If you can delete emails that have already been delivered, or put in blocking rules, *do a test run first!* In the same way, we should run exploratory searches for new analytics. Testing is heavily advised for anything you plan on doing to current, or future emails. There's no quicker way to lose the power to perform a task than bringing down a major business critical system.



Blocking with a Proxy / IPS / Next Gen Firewall

Application-aware devices can block on many attributes:

Layer 3

- Domain
- Source IP
- Destination IP
- ASN number

Layer 4/5

- Port #
- SSL Cert. details

Layer 7 Info

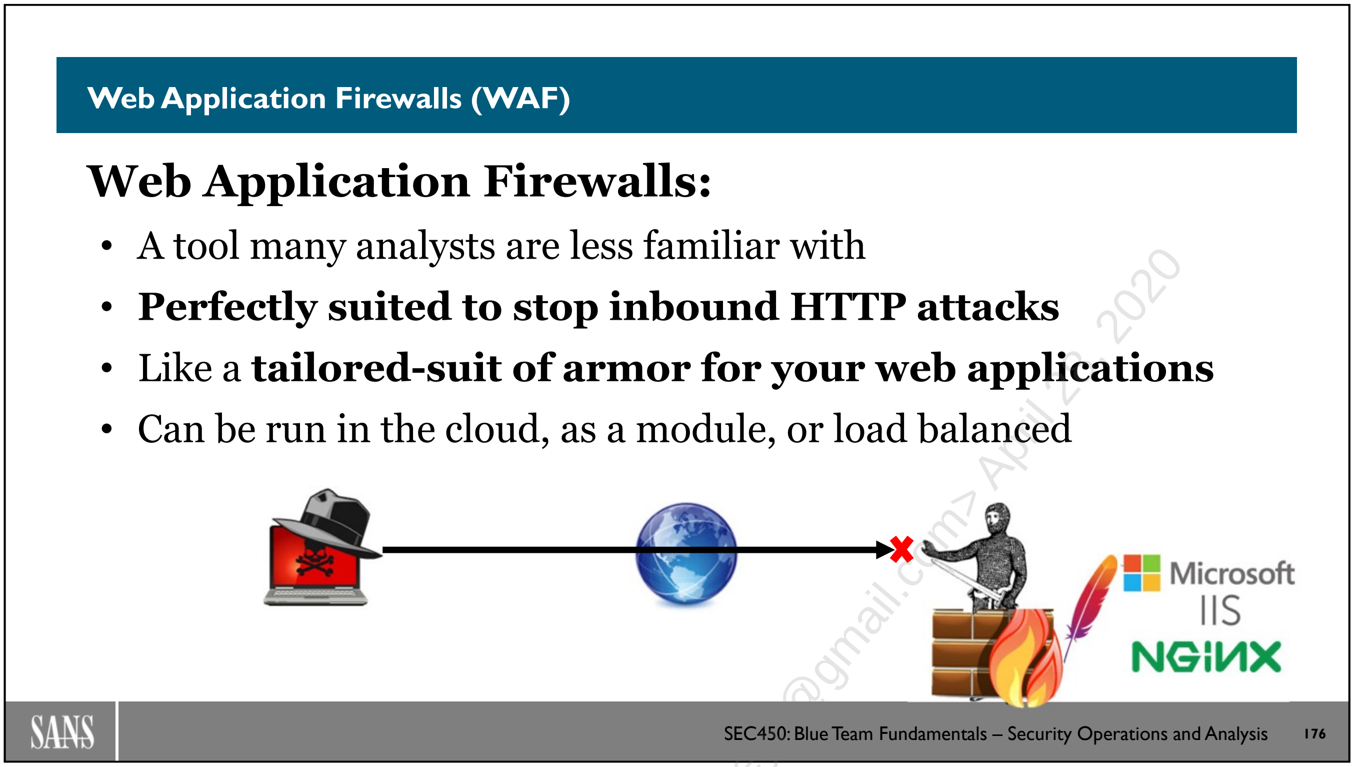
- URL (using regex)
- Method
- User-agent
- Referrer
- MIME type
- Status Code
- Content
- Protocol
  - HTTP(S)
  - FTP
  - SSH

Meta

- Category
- Risk level
- Username

Blocking with a Proxy / IPS / Next Gen Firewall

Looking beyond DNS RPZs, we still have many more options. Using devices that can see down to Layer 7, we can exert much more control of traffic. Proxies, network intrusion prevention systems, and next-gen firewalls can step in where DNS cannot, giving us the ability to block traffic based not only on Layer 7 content such as user-agents, mime types, referrers, and URL regex patterns, but also the domain category, risk level, usernames, and protocols as well. With access to one of these types of devices, you should have no technical issues blocking almost any type of traffic you come across. While not every environment will have proxies implemented, NGFW and IPS are much more common and detect many of the same fields and content items.



Web Application Firewalls (WAF)

Web Application Firewalls:

- A tool many analysts are less familiar with
- **Perfectly suited to stop inbound HTTP attacks**
- Like a **tailored-suit of armor for your web applications**
- Can be run in the cloud, as a module, or load balanced

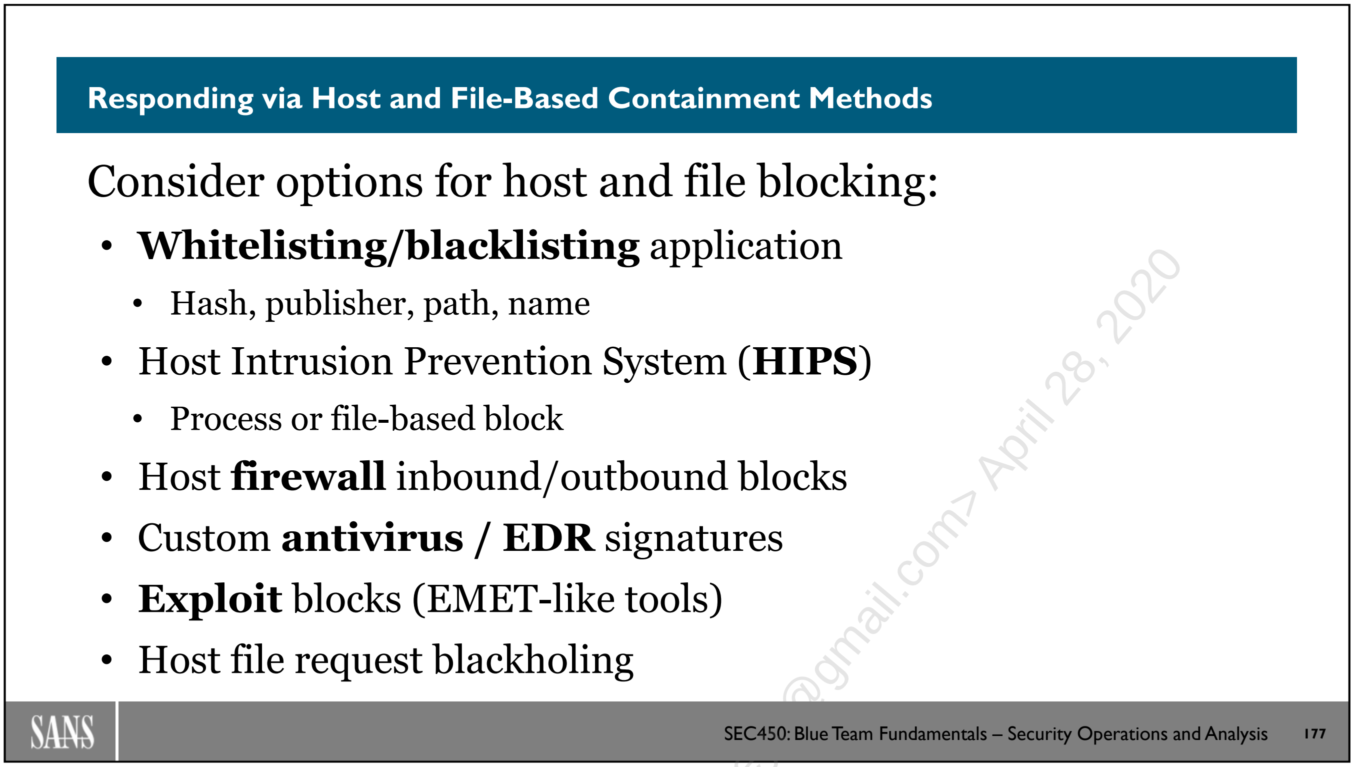
Microsoft
IIS
NGINX

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 176

Web Application Firewalls (WAF)

One final tool used for attack blocking and containment is the web application firewall. Although not a firewall in the normal sense, it is *much* more capable than even a next-gen firewall at stopping HTTP-based attacks going to your web servers. The firewall itself can be a separate appliance, deployed in the cloud forwarding clean traffic, or even installed as a module into the web server software itself. Regardless of the location, it can understand the HTTP protocol and filter the attempting input and output like no other tool can. If you find you do not have the fidelity you need to block an HTTP-based attack with a next-gen firewall or IPS, the web application firewall may be the right solution. See "SEC511 – Continuous Monitoring and Security Operations" for several labs and scenarios where a WAF may be the perfect answer to an attack. Examples include blocking data exfiltration utilizing honeytokens and preventing attempted SQL injections. WAFs can supply this type of visibility because, at least in their web server software module version, the WAF firewall can see all data after SSL/TLS decryption on the way in, as well as before encryption on the way back to the client.



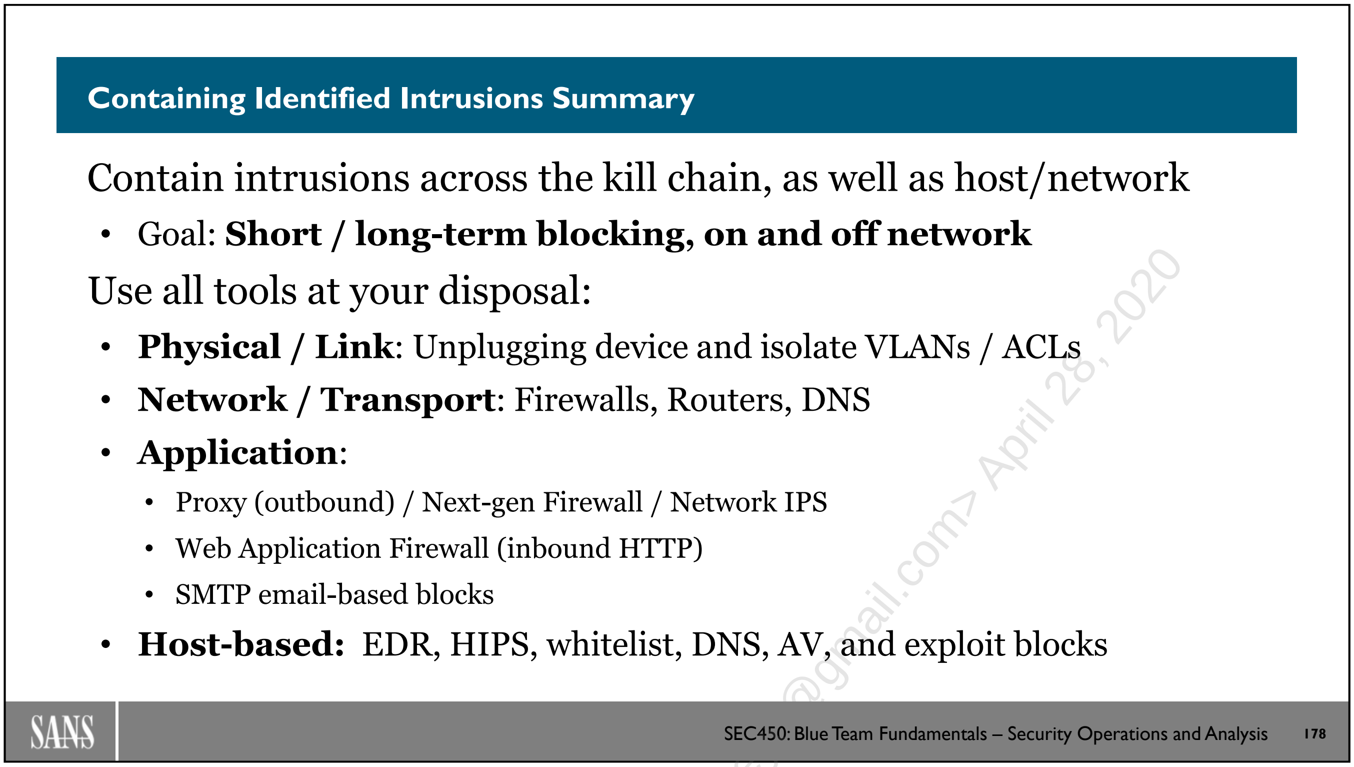
Responding via Host and File-Based Containment Methods

Consider options for host and file blocking:

- **Whitelisting/blacklisting** application
  - Hash, publisher, path, name
- Host Intrusion Prevention System (**HIPS**)
  - Process or file-based block
- Host **firewall** inbound/outbound blocks
- Custom **antivirus / EDR** signatures
- **Exploit** blocks (EMET-like tools)
- Host file request blackholing

Responding via Host and File-Based Containment Methods

For responding directly on the host, some methods involve attacking malicious processes, and others that can strike at the file level. For the process level, we can employ the use of whitelisting applications or host intrusion prevention systems to stop a system from running a file with a specific hash, path, or name. Using host-based methods in combination with network-based methods is highly recommended to continue protection for laptops when they leave the network.



Containing Identified Intrusions Summary

Contain intrusions across the kill chain, as well as host/network

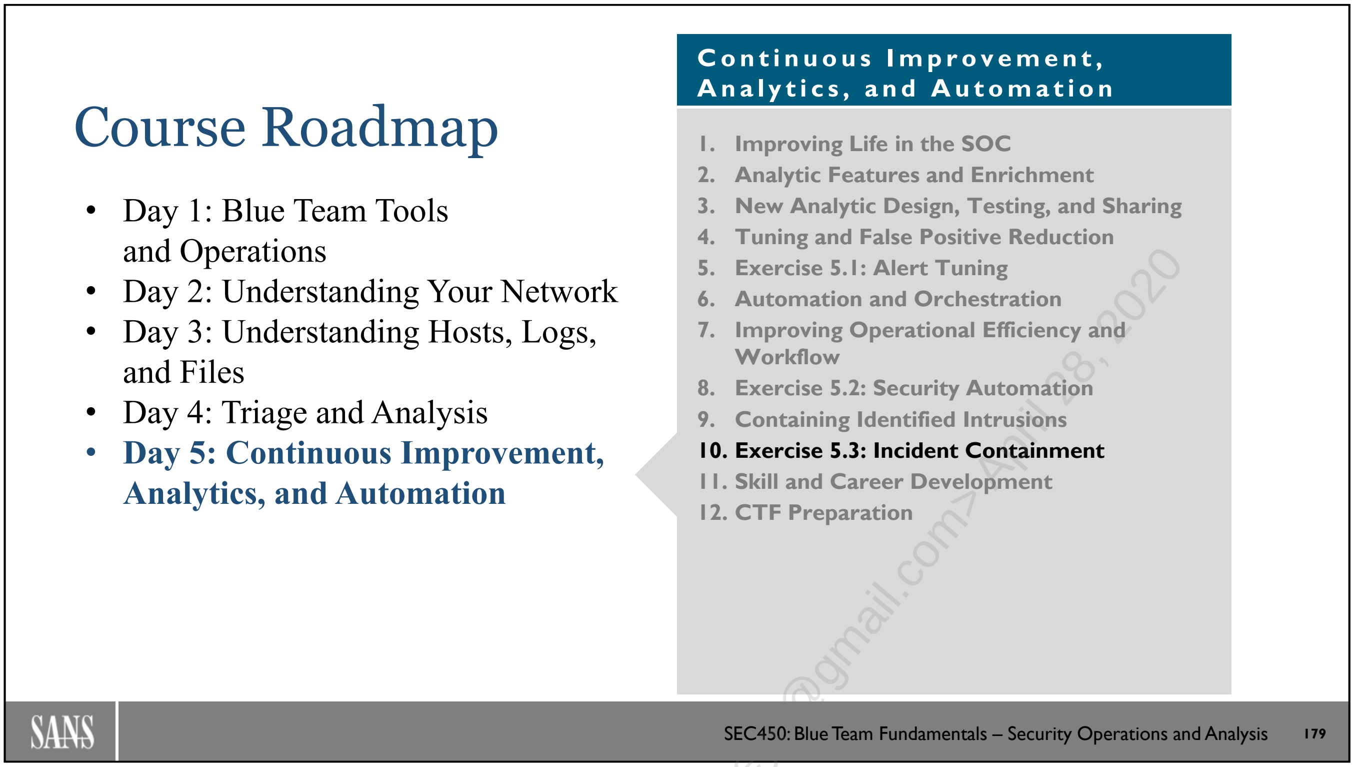
- Goal: **Short / long-term blocking, on and off network**

Use all tools at your disposal:

- **Physical / Link**: Unplugging device and isolate VLANs / ACLs
- **Network / Transport**: Firewalls, Routers, DNS
- **Application**:
  - Proxy (outbound) / Next-gen Firewall / Network IPS
  - Web Application Firewall (inbound HTTP)
  - SMTP email-based blocks
- **Host-based:** EDR, HIPS, whitelist, DNS, AV, and exploit blocks

Containing Identified Intrusions Summary

When confronted with an active intrusion, you will be called upon to decide the most effective way to contain an incident in both the short and long term. When you do this, consider the asset closely—whether it will be moving in and out of the network (user laptop), what else it can talk to on the network (to prevent lateral movement), and how urgent it is to isolate it from all other traffic. While some low-risk incidents may call for a simple domain block implemented through DNS or a proxy, other intrusions will undoubtedly be of the "go find and disconnect it NOW" type. Knowing the strengths, weaknesses, risks, and usability of each of these methods is important to providing an appropriate response to all the issues we will come across.



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**

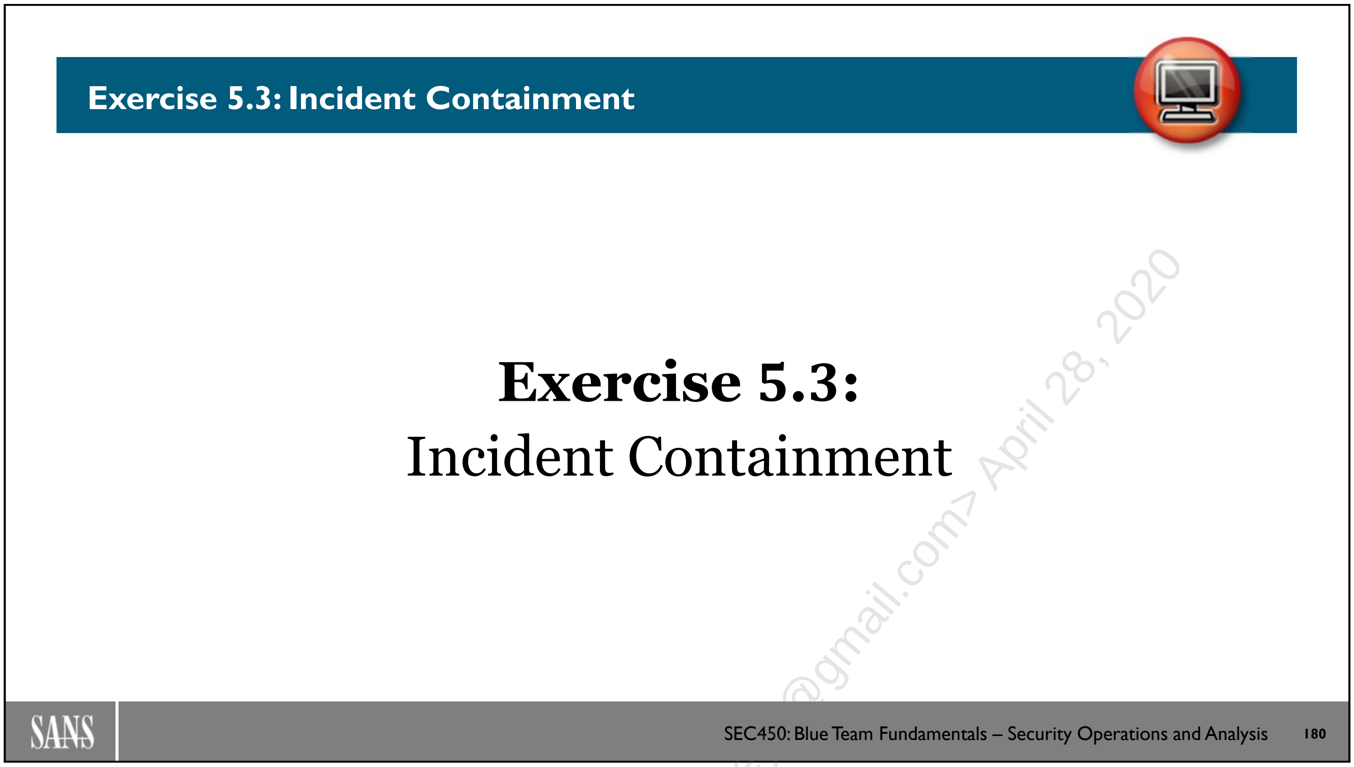
Continuous Improvement, Analytics, and Automation

1. Improving Life in the SOC
2. Analytic Features and Enrichment
3. New Analytic Design, Testing, and Sharing
4. Tuning and False Positive Reduction
5. Exercise 5.1: Alert Tuning
6. Automation and Orchestration
7. Improving Operational Efficiency and Workflow
8. Exercise 5.2: Security Automation
9. Containing Identified Intrusions
10. **Exercise 5.3: Incident Containment**
11. Skill and Career Development
12. CTF Preparation

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 179

This page intentionally left blank.



Exercise 5.3: Incident Containment

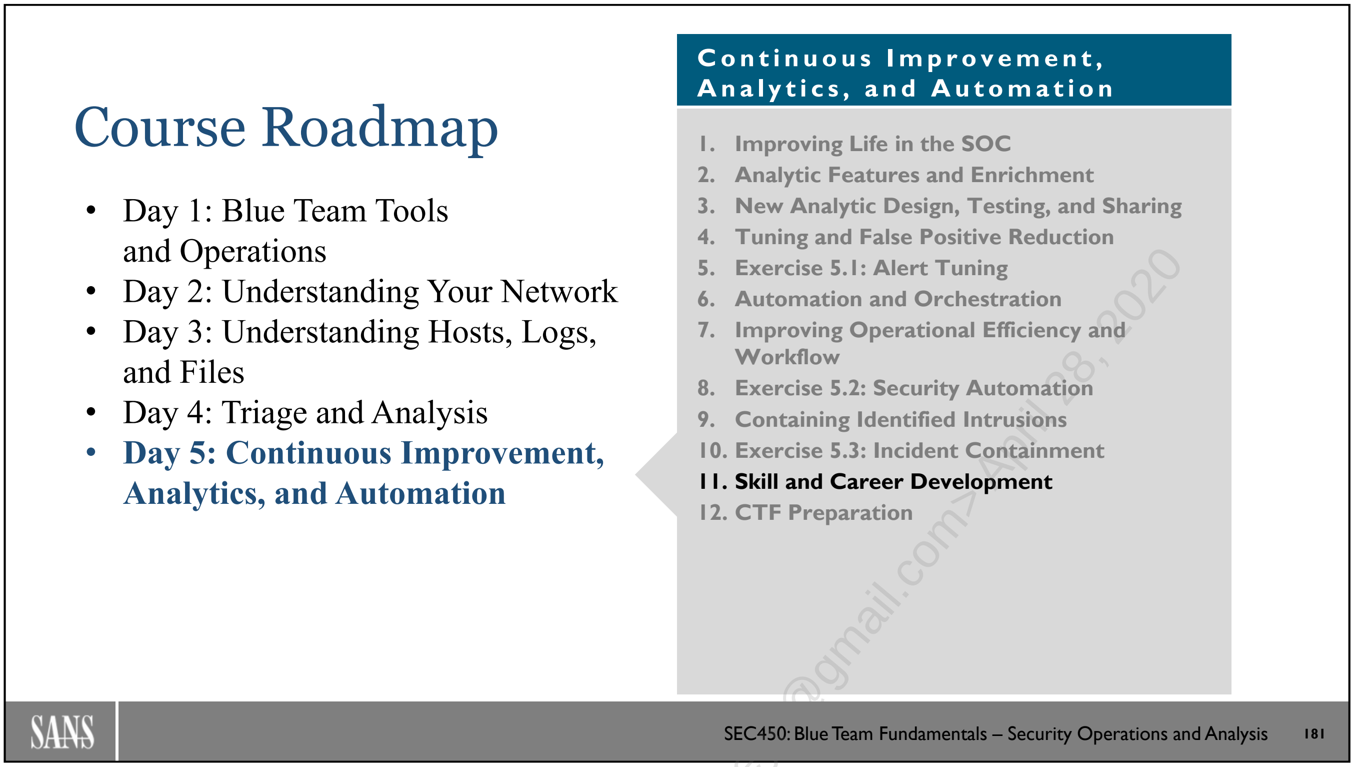
Exercise 5.3:
Incident Containment

SANS

SEC450: Blue Team Fundamentals – Security Operations and Analysis 180

Exercise 5.3: Incident Containment

Please go to Exercise 5.3 in the SEC450 Workbook or virtual wiki.



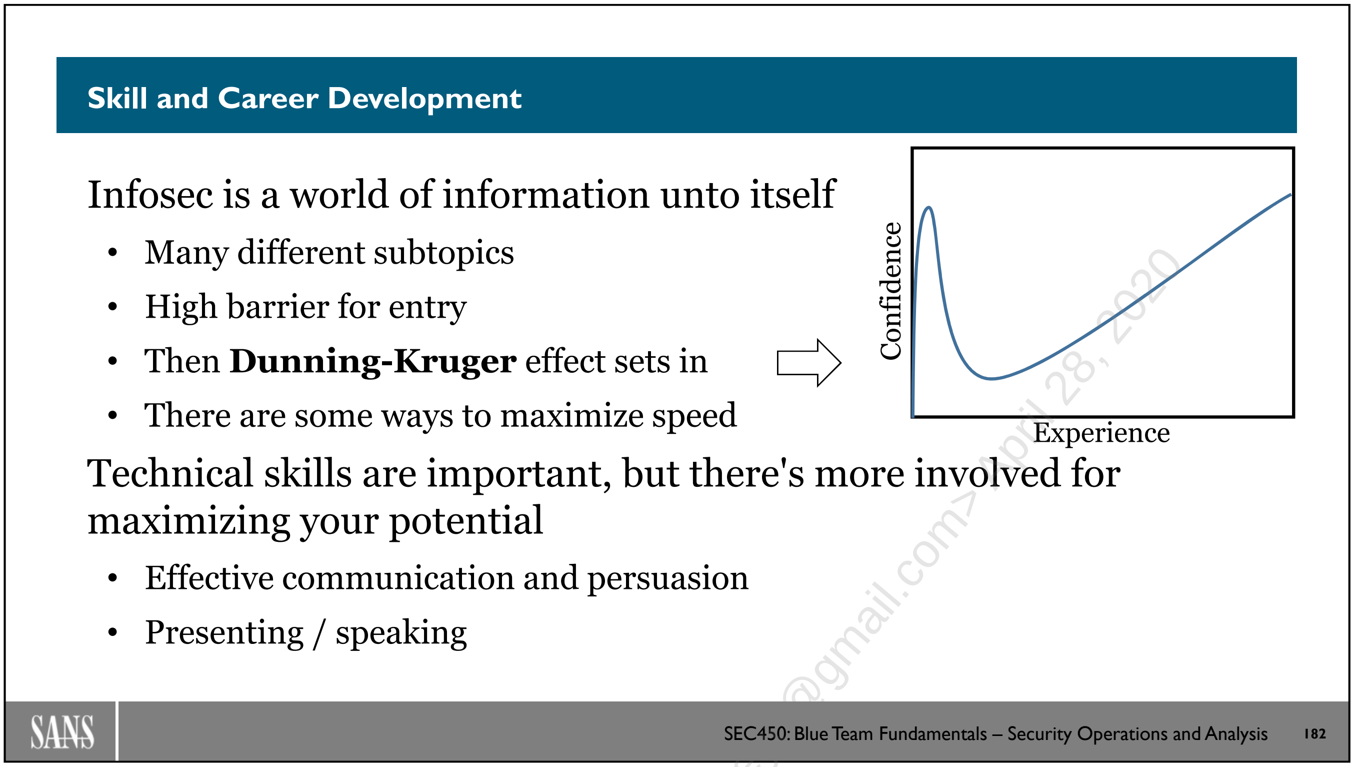
Course Roadmap

- 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**

Continuous Improvement, Analytics, and Automation

1. Improving Life in the SOC
2. Analytic Features and Enrichment
3. New Analytic Design, Testing, and Sharing
4. Tuning and False Positive Reduction
5. Exercise 5.1: Alert Tuning
6. Automation and Orchestration
7. Improving Operational Efficiency and Workflow
8. Exercise 5.2: Security Automation
9. Containing Identified Intrusions
10. Exercise 5.3: Incident Containment
11. **Skill and Career Development**
12. CTF Preparation

This page intentionally left blank.



Skill and Career Development

Infosec is a world of information unto itself

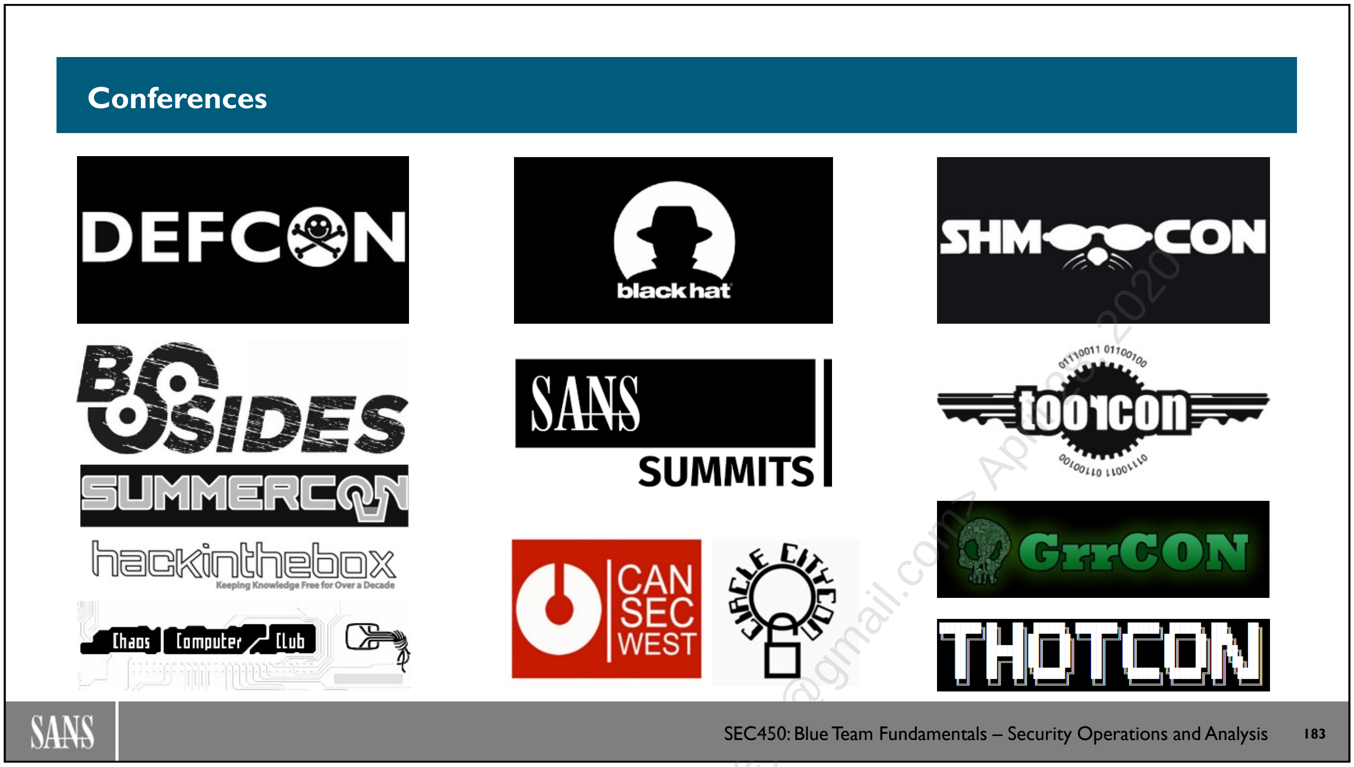
- Many different subtopics
- High barrier for entry
- Then **Dunning-Kruger** effect sets in ⇨
- There are some ways to maximize speed

Technical skills are important, but there's more involved for maximizing your potential

- Effective communication and persuasion
- Presenting / speaking

Skill and Career Development

Getting starting in information security can be stressful considering the vast breadth of knowledge required. To make things worse, learning a bit often gives way to an additional realization about how much you *don't* know, compounding the effect. This misguided confidence at low levels of experience is illustrated by the Experience/Confidence graph on the slide in a phenomenon called the "Dunning-Kruger Effect." While there is indeed an enormous amount of possible knowledge under the umbrella of information security, you in no way must or will be expected to know *all* of it—it's simply too vast. That being said, there *are* ways to rapidly improve the pace at which you can pick up new material and jump into the information security community. Those methods include going to conferences, participating in Capture the Flag (CTF) challenges, making time for deliberate practice, as well as things like improving soft skills and prioritizing your time effectively. In this section, we'll close the course with some advice on where to go from here and how to ensure continual growth.

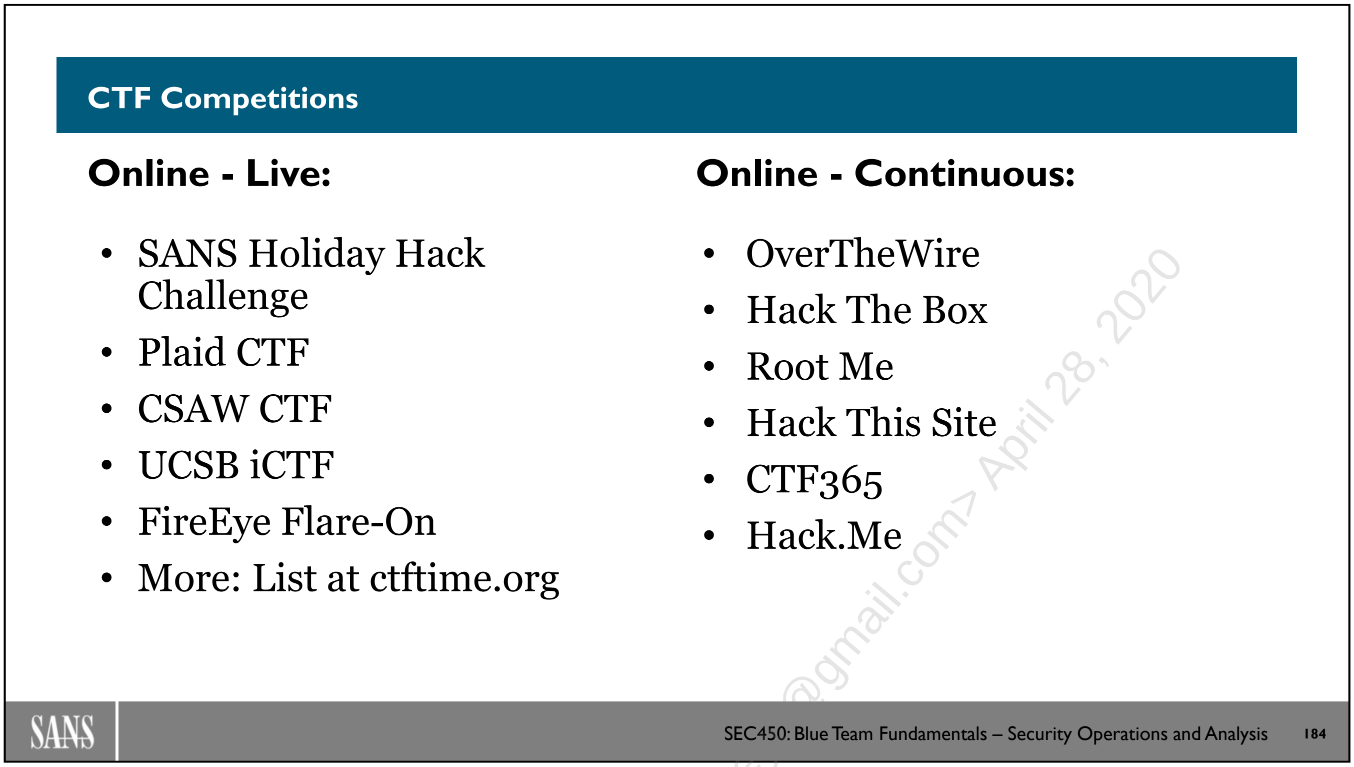


Conferences

SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 183

Conferences

There are *tons* of information security conferences to choose from all over the world. While Black Hat USA and DEFCON remain the biggest in terms of attendance, there are many options for regional events no matter where you are. Security BSides and SANS Summit conferences are the most prolific, taking place in cities all over the world. Outside of the mega conferences and BSides / SANS Summits held in various locations, there are regional conferences such as Shmoocon (Washington D.C.), Chaos Communication Congress (Germany), CanSecWest (Vancouver, BC), CircleCityCon (Indianapolis, IN), GrrCON, (Grand Rapids, MI), THOTCON (Chicago), toorcon (San Diego, CA), Hack in the box (various), SummerCON (Brooklyn, NY), Kiwicon (New Zealand), Pumpcon (Philadelphia, PA), and many, many more. With some quick searching, you're likely to find something near you that will not only teach you something new but connect you with your local information security community.



CTF Competitions

Online - Live:

- SANS Holiday Hack Challenge
- Plaid CTF
- CSAW CTF
- UCSB iCTF
- FireEye Flare-On
- More: List at ctftime.org

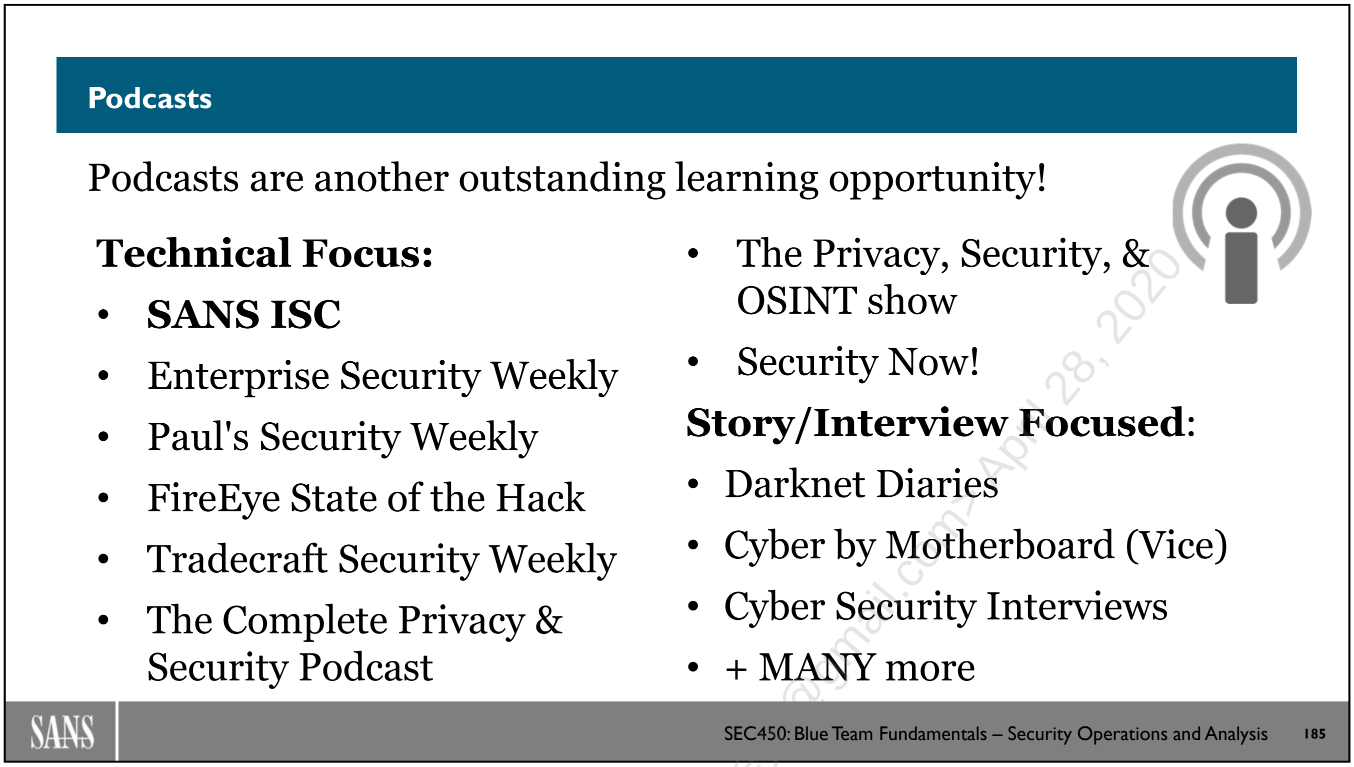
Online - Continuous:

- OverTheWire
- Hack The Box
- Root Me
- Hack This Site
- CTF365
- Hack.Me

CTF Competitions

Outside of conferences, there is ample opportunity for testing your skills online as well. Most online CTFs are "jeopardy style" with a set of data and questions to answer about it, while others are "team vs. team" type action. In addition, some online CTFs are more penetration testing focused and others have challenges aimed at defenders. Participating in either will undoubtedly help you build skill outside of work.

CTF challenge authors are a creative bunch and the things they come up with range from "totally realistic" to "I don't think everyone would ever do this in real life, but it's interesting that it's possible." Seeing both types of exercises can help open your eyes to some truly unique methods that attackers may use to hide or obscure data, and you never know when the knowledge will become useful in daily life.



Podcasts

Podcasts are another outstanding learning opportunity!

Technical Focus:

- **SANS ISC**
- Enterprise Security Weekly
- Paul's Security Weekly
- FireEye State of the Hack
- Tradecraft Security Weekly
- The Complete Privacy & Security Podcast
- The Privacy, Security, & OSINT show
- Security Now!

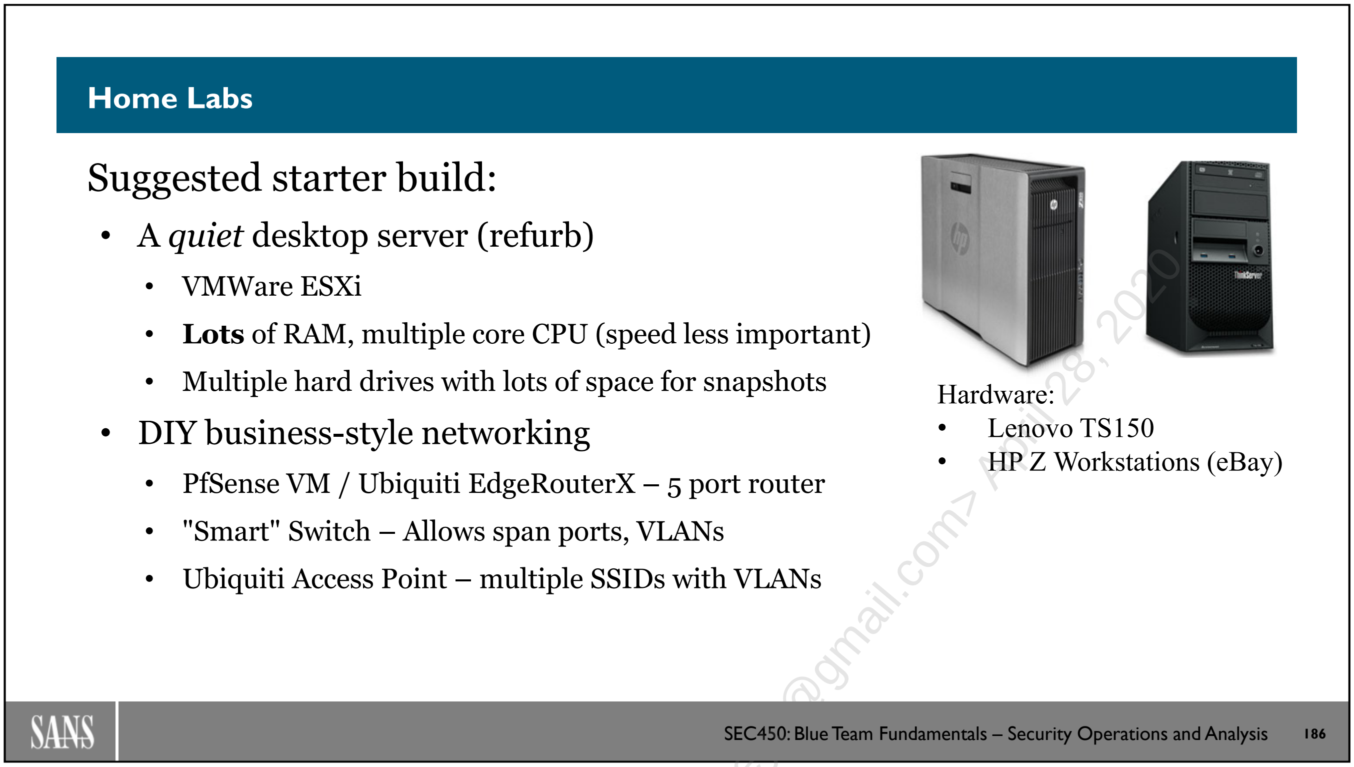
Story/Interview Focused:

- Darknet Diaries
- Cyber by Motherboard (Vice)
- Cyber Security Interviews
- + MANY more

Podcasts

Podcasts are another outstanding way to continue your skills development and make a great side task while doing other work. Podcasts make learning while commuting, cleaning your house or exercising easy and doubles your productivity! This slide lists some of the *many* security podcasts that are out there. I've personally listened to all these podcasts over time and can vouch for their educational content, but there's way more to choose from and new ones showing up every day. Depending on whether you are looking for news, interviews, or learning new concepts in defense, offense, OSINT, or threat intel, there are shows out there that are sure to grab your interest. Each show has its own attitude, release schedule, and level of polish, so give a few a try and see which may be a good match for you.

There's one show I recommend to everyone no matter what, and that is the daily SANS ISC podcast from Johannes Ullrich. It's 5 minutes long, easy to fit in on your way to work, and is great for getting the most important news of the day in as little time as possible. Give it a try at https://isc.sans.edu/podcast.html.



Home Labs

Suggested starter build:

- A *quiet* desktop server (refurb)
  - VMWare ESXi
  - **Lots** of RAM, multiple core CPU (speed less important)
  - Multiple hard drives with lots of space for snapshots
- DIY business-style networking
  - PfSense VM / Ubiquiti EdgeRouterX – 5 port router
  - "Smart" Switch – Allows span ports, VLANs
  - Ubiquiti Access Point – multiple SSIDs with VLANs

Hardware:

- Lenovo TS150
- HP Z Workstations (eBay)

Home Labs

One of the best ways to quickly begin to understand how monitoring works on a company scale is to emulate it at home. It takes a surprisingly little amount of hardware to simulate a multi-segment corporate environment completely with VLANs, full traffic capture capability, multiple SSIDs, and enterprise-grade virtualization. With a single decently equipped server or powerful workstation PC, and a couple of hundred dollars in networking gear, you're in business and can start doing network security monitoring at home!

Although by no means an exhaustive list, here is some of the hardware I've seen well reviewed and used with success for home labs, not just for its flexibility, but because of its lack of a requirement to buy a bulky rack to put it in. If you can buy a server rack and full-size equipment and hide it in a basement where you won't hear it, by all means, go ahead though!

Servers:

- Lenovo TS150: This series of desktop form-factor servers is extremely quiet and is relatively affordable, great for a starting home lab.
- Refurbished HP Z Workstations: There are *tons* of refurb versions available on eBay, many pre-equipped with 6+ core processors and lots of RAM. Check out the model year of the processor. Some are rather old, but will still work fine for a home lab if they have the VT-X and VT-d features built into the process (required for virtualization features).

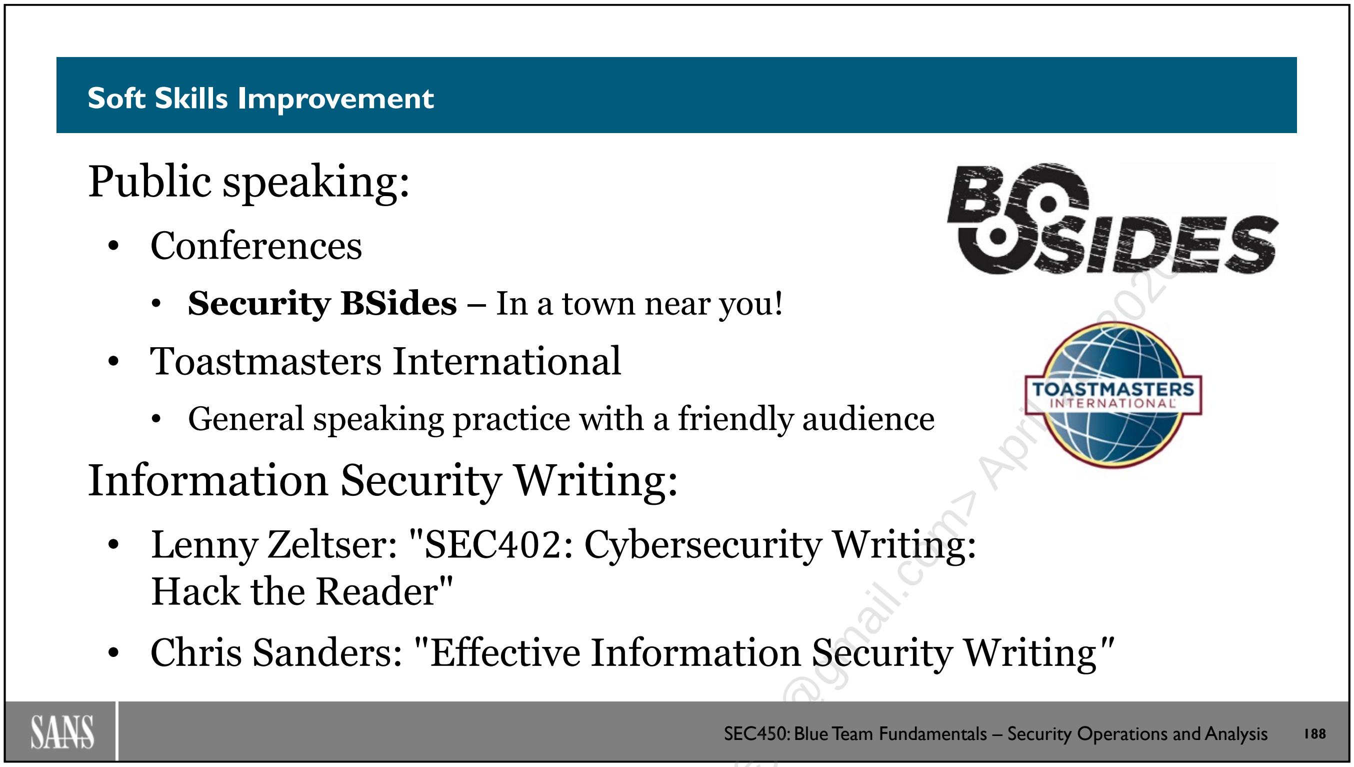
Networking Equipment:

- Ubiquiti: Enterprise features on a hobbyist budget.
  - EdgeRouterX: A tiny, fanless five port router that costs $50, a great way to get into "real" routers
  - Unifi Access Points: Enterprise-grade wireless features such as VLAN tagging and multiple SSIDs (multiple prices, depending on model)

- PfSense: A great, free, and easy-to-install router and firewall that can be used as a separate computer, or even as a virtual router within your server, but remember to have a backup for internet connectivity close by if the server goes down and you aren't home. ;)
- Switches: There are three types of switches—unmanaged, which you do NOT want; smart, which will probably do what you need; and fully managed, which are rather expensive but awesome if you can get one. Any smart switch that fits your budget and supports VLAN tagging and a mirror port should get the job done, and likely will not have a fan like many managed switches will.

For additional information, check out the r/homelab Reddit page and the wiki.[1]

[1] https://www.reddit.com/r/homelab/wiki/index



Soft Skills Improvement

Public speaking:

- Conferences
  - **Security BSides** – In a town near you!
- Toastmasters International
  - General speaking practice with a friendly audience

Information Security Writing:

- Lenny Zeltser: "SEC402: Cybersecurity Writing: Hack the Reader"
- Chris Sanders: "Effective Information Security Writing"

BSIDES
TOASTMASTERS INTERNATIONAL
SANS

Soft Skills Improvement

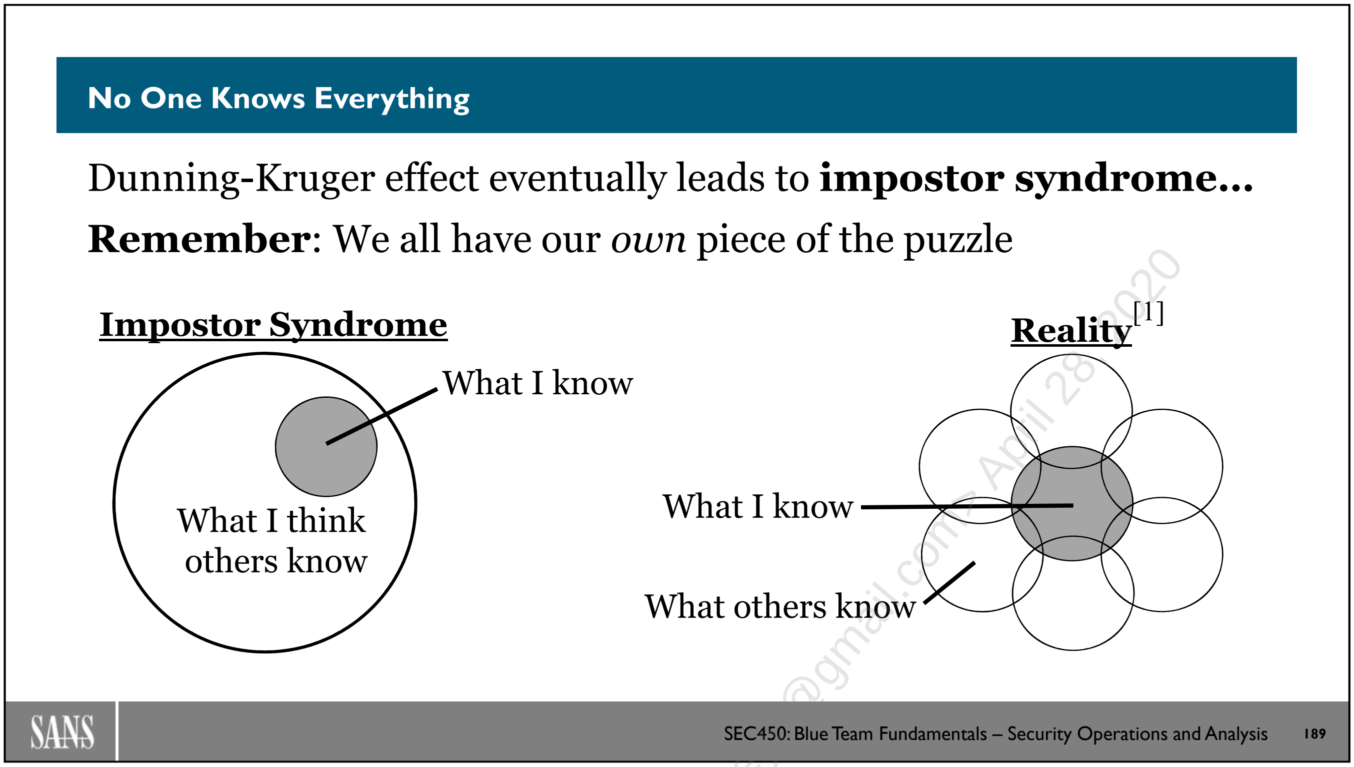
While technical skills are an absolute necessity, being an effective SOC member is not only about your technical capability. Being an effective communicator both in speaking and writing can help further propel your career as these skills will be needed when presenting incident results to management and trying to persuade others of why your suggested security changes or rules are necessary. This slide lists some resources for learning and practicing these types of soft skills.

On public speaking: While many may prefer to avoid public speaking, most people would probably like to be comfortable with public speaking for the times it becomes necessary. Practice with a friendly audience is the single best way to become comfortable and Toastmasters International is a club with branches all over the world designed around that fact, giving members a weekly supportive atmosphere to give simple, short talks on various topics to help practice and receive constructive feedback. To get started in information security speaking, Security BSides conferences are a great place to submit your first talk to give to your local community.

On the writing front, SANS is debuting a brand-new short course from Lenny Zeltser called SEC402 on effective cybersecurity writing. Chris Sanders has an online class available as well on Effective Information Security writing, as well as several other courses that would pair well with this class on threat hunting and analysis technique.[1 2]

In addition to the resources above, SANS Principal Instructor Ted Demopoulos wrote a book in 2017 based on his career in the industry with solid advice on various topics from time management to owning your online image titled, *Infosec Rock Star: How to Accelerate Your Career Because Geek Will Only Get You So Far*.[3]

[1] https://cyber-defense.sans.org/blog/2019/01/08/secrets-to-cybersecurity-writing-hack-the-reader
[2] https://chrissanders.org/training/writing/
[3] https://www.amazon.com/dp/B072KG7G4P/ref=dp-kindle-redirect?_encoding=UTF8&btkr=1



No One Knows Everything

Dunning-Kruger effect eventually leads to **impostor syndrome…**

Remember: We all have our *own* piece of the puzzle

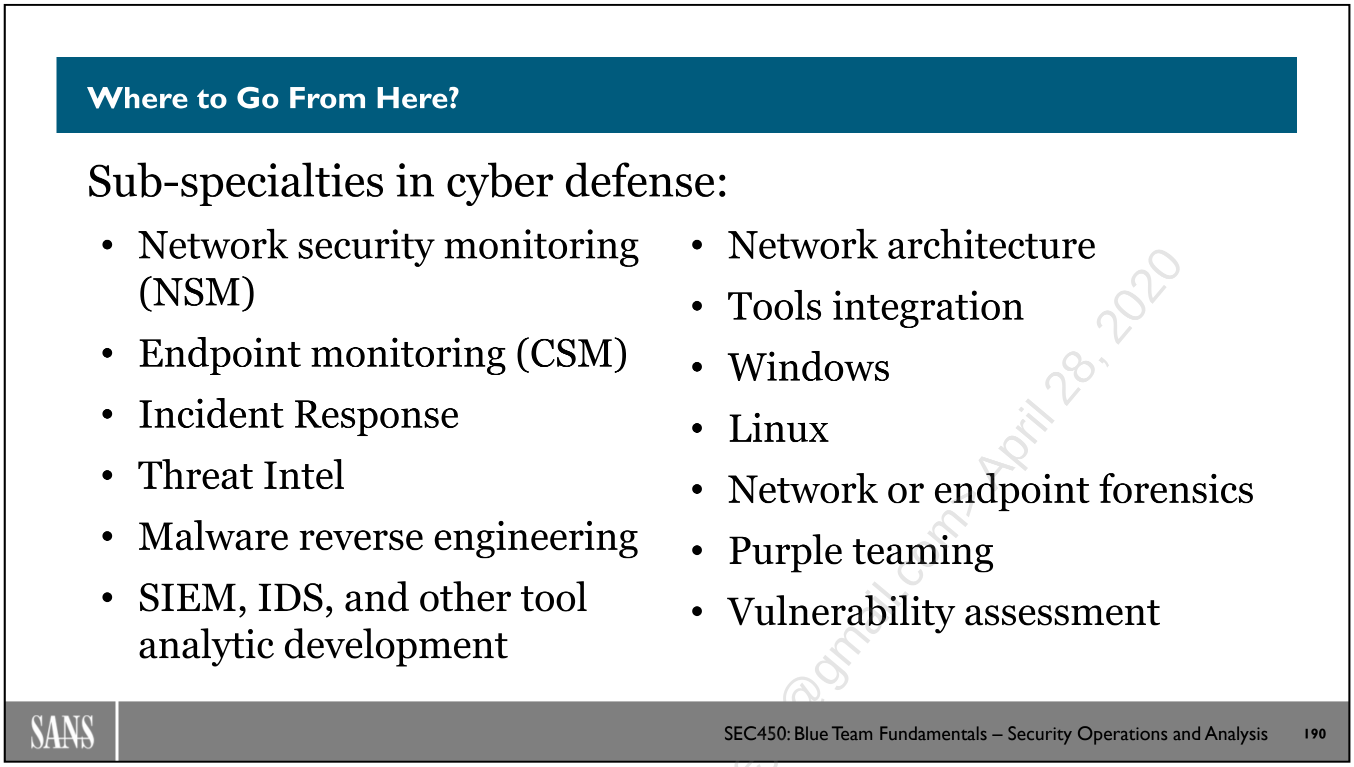
No One Knows Everything

As you progress through your career and dive deeper into information security, you will continue to learn the incredible amounts of information required to perform a cyber defense role. While getting a boost of information can initially lead to excitement and feelings of confidence, over the longer term, it often gives way to the realization of just how much you *don't* know. This phenomenon is part of the cycle called the Dunning-Kruger effect. Those who get an initial taste of success in a field can become overconfident at first, and then in short order come to realize there is a whole additional world of information for them to learn. This realization is useful but is also often followed by feelings of inadequacy, despair, and the tendency to feel like a fraud. This phenomenon has a name as well—Impostor Syndrome—and it can be prevalent in information security where we always hear about the newest amazing hack that proceeded to stun the world. Seeing this type of research and the subsequent media attention can lead us to believe that "we will never be as smart as x" or that "we could never figure out something like that / write such a useful piece of code." David Whittaker has wonderfully illustrated this effect a tweet he posted describe the topic, recreated on the slide above.[1]

Remember, we all know something that others don't, and we all have something unique to contribute. Although there are numerous very intelligent people in this field doing impressive things, everyone starts at zero and decides where to place their attention and focus. Keep in mind that no one knows everything. While some have focused on one piece of the knowledge puzzle, you too have unique knowledge that you could undoubtedly share that most people do not possess. Succumbing to those feelings of Impostor Syndrome keeps that knowledge inside and prevents you from sharing and helping the community grow. Micah Hoffman gave an excellent talk on overcoming impostor syndrome and the feelings of despair that can come with taking on a large topic such as information security. To further understand the phenomenon, check out the video from BSides DC 2017.[2]

[1] Recreation of David Whittaker's (@rundavidrun) image on Impostor Syndrome: https://twitter.com/rundavidrun/status/587671657193455616

[2] Micah Hoffman – Impostor Syndrome: I Don't Feel Like Who You Think I Am - https://www.youtube.com/watch?v=aDDCl_VQx7k



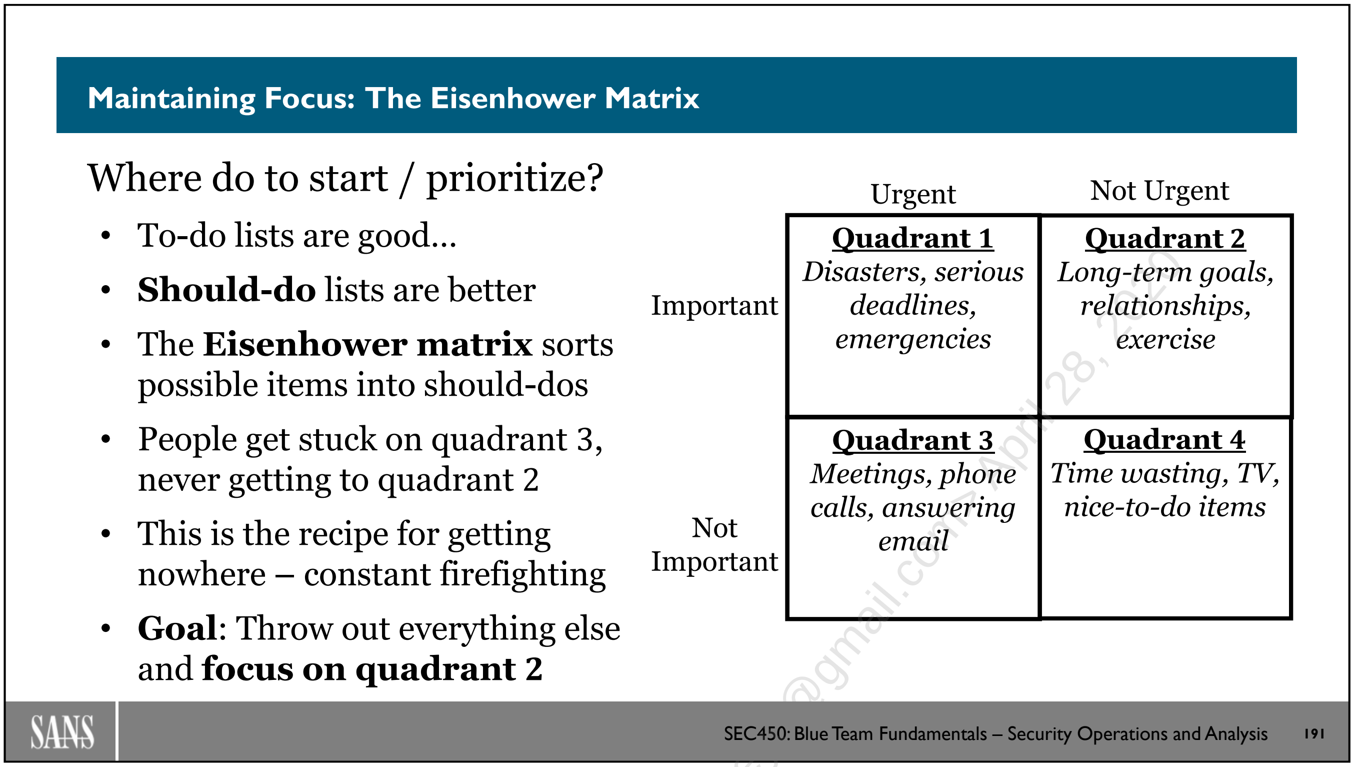
Where to Go From Here?

Sub-specialties in cyber defense:

- Network security monitoring (NSM)
- Endpoint monitoring (CSM)
- Incident Response
- Threat Intel
- Malware reverse engineering
- SIEM, IDS, and other tool analytic development
- Network architecture
- Tools integration
- Windows
- Linux
- Network or endpoint forensics
- Purple teaming
- Vulnerability assessment

Where to Go From Here?

Although you will likely want to hone your skills in the wide-exposure environment of the SOC for a while, after gaining general defense skill, most people get the urge to specialize. There are many different subtopics within cyber defense alone, each with more information than one can master in a lifetime. As SOC skills go, NSM and CSM, incident response, and malware reverse engineering, memory forensics, and threat intel tend to be the most popular activities. For a slight departure into system administration, there are also lots of opportunities to specialize in configuration hardening of Windows and Linux, integration of security tools, or SIEM/IDS/other analytic design. There are also opportunities in self-assessment activities such as vulnerability scanning, penetration testing, and red/purple teaming.



Maintaining Focus: The Eisenhower Matrix

Where do to start / prioritize?

- To-do lists are good…
- **Should-do** lists are better
- The **Eisenhower matrix** sorts possible items into should-dos
- People get stuck on quadrant 3, never getting to quadrant 2
- This is the recipe for getting nowhere – constant firefighting
- **Goal**: Throw out everything else and **focus on quadrant 2**

	Urgent	Not Urgent
Important	**Quadrant 1** *Disasters, serious deadlines, emergencies*	**Quadrant 2** *Long-term goals, relationships, exercise*
Not Important	**Quadrant 3** *Meetings, phone calls, answering email*	**Quadrant 4** *Time wasting, TV, nice-to-do items*

Maintaining Focus: The Eisenhower Matrix

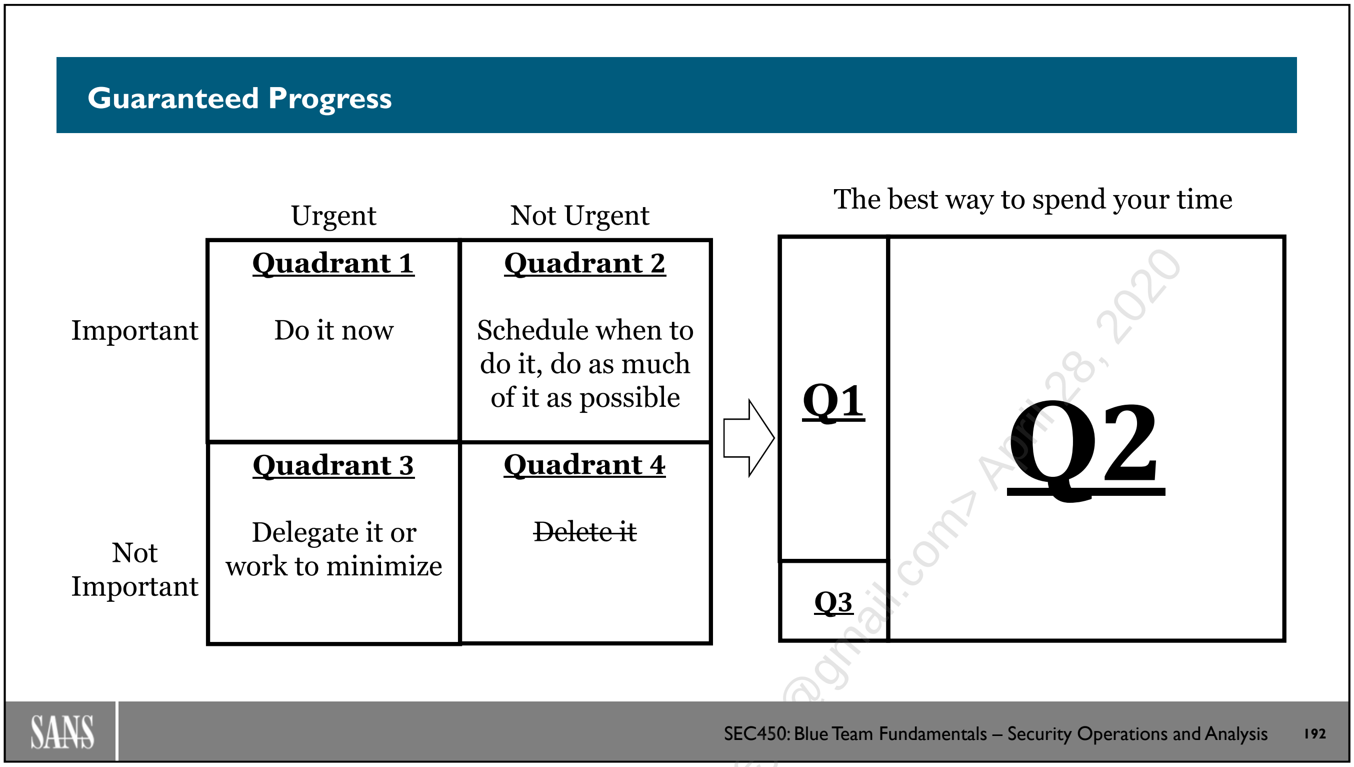
With our busy lives, there's never a shortage of things we *could* do. The problem is that most people put all these things in a to-do list and don't think hard enough about the best place to start. While it is tempting to always go for the item that is due next, it's not necessarily the approach that will lead you to the point you're trying to get. Your list should be looked at along two separate axes—what is *urgent* and what is important. The framework called the "Eisenhower matrix," named after U.S. President Dwight Eisenhower, who used it as his guiding prioritization system.[1] Important and urgent are up to your definition, but in the article quoted below, the following definitions are used:

"**Urgent** means that a task requires immediate attention. These are the to-do's that shout "Now!" Urgent tasks put us in a *reactive* mode, one marked by a defensive, negative, hurried, and narrowly-focused mindset.

Important tasks are things that contribute to our long-term mission—values, and goals. Sometimes, important tasks are also urgent, but typically they're not. When we focus on important activities, we operate in a *responsive* mode, which helps us remain calm, rational, and open to new opportunities."[1]

Since we're talking about career progression, let's assume important equates to deliberate practice or learning a new technical skill. If your personal goal is to work toward mastering PCAP analysis, for example, but your house is messy, you must ship a package, you need to respond to emails, and a new episode of your favorite TV show just got released, where do you start? Many people will immediately jump to the package and emails, seeing them as urgent and the most important. Others may want to take care of *all* urgent items on their to-do list before getting to the non-urgent items, thinking that it will help them clear their mind when they finally get around to it. The problem is that this often does not work. Since there is always something you can view as urgent, not kicking things out and making space for your real goals can cost you dearly over time.

[1] https://www.artofmanliness.com/articles/eisenhower-decision-matrix/



Guaranteed Progress

	Urgent	Not Urgent
Important	**Quadrant 1** Do it now	**Quadrant 2** Schedule when to do it, do as much of it as possible
Not Important	**Quadrant 3** Delegate it or work to minimize	**Quadrant 4** ~~Delete it~~

The best way to spend your time

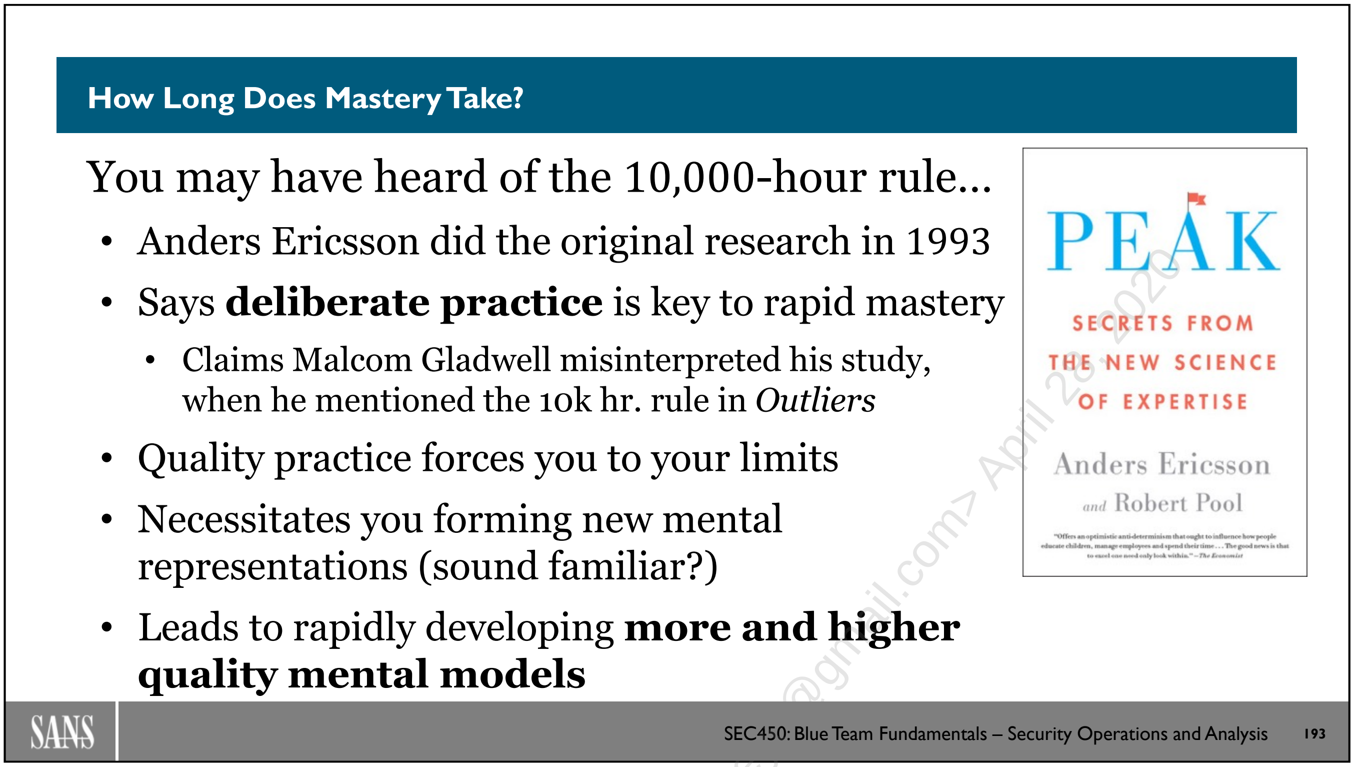
Guaranteed Progress

How *should* we operate? If we want to make incredible progress in a short time, we must focus all the attention on those items in quadrant 2 and not get pulled in by the false importance of quadrant 3, or the temptation of quadrant 4. In our example to-do list on the previous slide, this might mean dedicating the first few hours of your day toward that longer-term important, but non-pressing goal. If we can fit in at a minimum 1 hour of PCAP analysis per day, then deal with everything else we can in the remaining time, we are guaranteed at least a minimum level of effort on our truly important item. Doing this involves being very clear about what "important" means to you, and where each item truly sits. Doing this may mean letting certain to-do list items slide for a period, which can be uncomfortable, but in return, you will have guaranteed progress on the items that you have decided matter most. With a consistent level of effort over time (even if it is small), hours dedicated to the task add up and progress toward mastery is assured.

A more slightly different but also useful way of putting it is how Gary Keller and Jay Papasan, author of the best-selling book "The ONE Thing: The Surprisingly Simple Truth Behind Extraordinary Results" suggest you move toward success. They suggest asking yourself "the focusing question" each day, which is "What is the ONE thing I can do, such that by doing it everything will become easier and unnecessary."[1] Given that Gary Keller is the owner of Keller Williams Real Estate, the largest real estate company in the world, and Jay Papasan was named on the list of "Most Powerful People in Real Estate", I'm inclined to think their method, which bears a resemblance to the Eisenhower matrix, works quite well.

For those who tend to be pulled into procrastination and any other Q-4 task, the blog "Wait But Why?" has a great two-part series plus side article on the Eisenhower matrix with mental models of how and why procrastination happens that make for a great and entertaining read.[2][3][4]

[1] https://www.the1thing.com/
[2] https://waitbutwhy.com/2015/03/procrastination-matrix.html
[3] https://waitbutwhy.com/2013/10/why-procrastinators-procrastinate.html
[4] https://waitbutwhy.com/2013/11/how-to-beat-procrastination.html



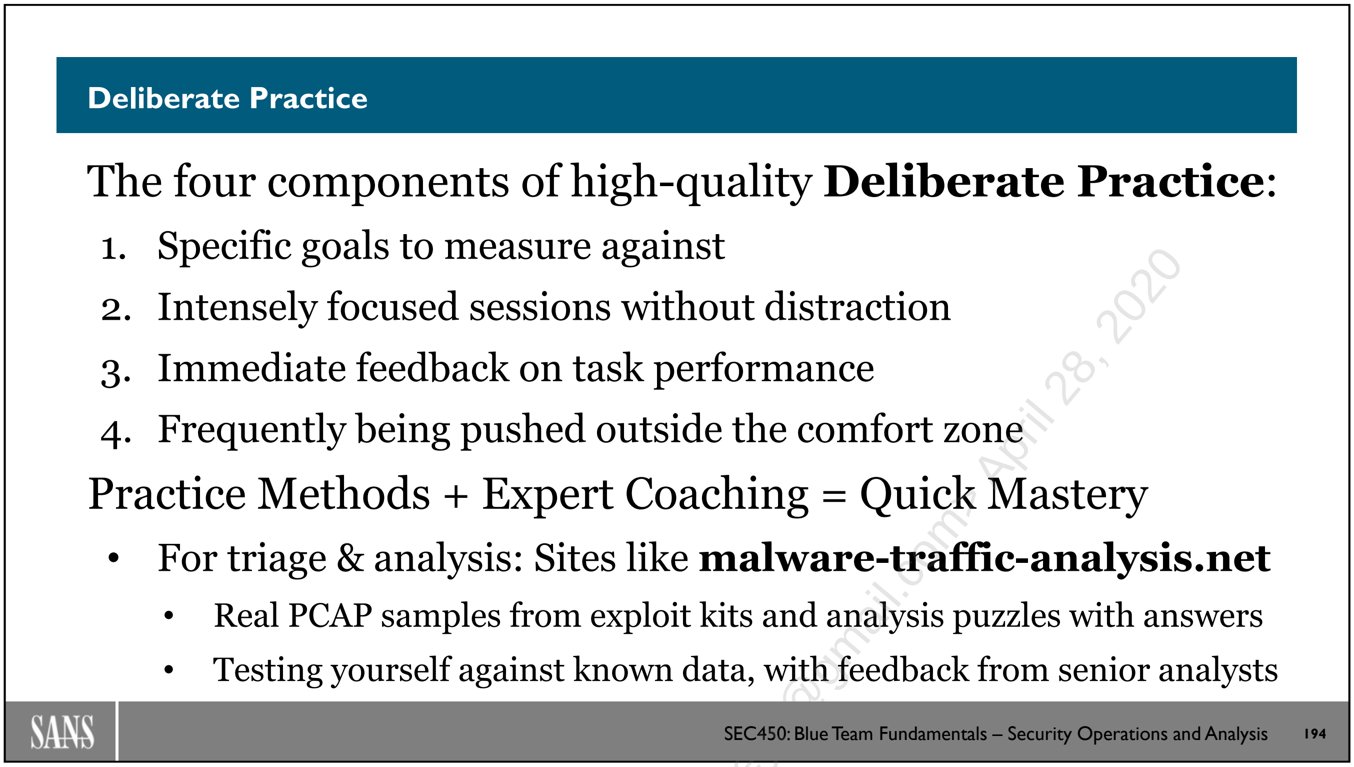
How Long Does Mastery Take?

You may have heard of the 10,000-hour rule…

- Anders Ericsson did the original research in 1993
- Says **deliberate practice** is key to rapid mastery
  - Claims Malcolm Gladwell misinterpreted his study, when he mentioned the 10k hr. rule in *Outliers*
- Quality practice forces you to your limits
- Necessitates you forming new mental representations (sound familiar?)
- Leads to rapidly developing **more and higher quality mental models**

How Long Does Mastery Take?

What specifically can you do, process-wise, regardless of the task to ensure you are getting better at a maximum rate? You may have heard of the 10,000-hour rule, popularized by Malcolm Gladwell in his book *Outliers*. The data was derived from a study Psychologist Anders Ericsson did in 1993. The kicker is, Ericsson says Gladwell misinterpreted what the study showed. Sure, the *average* for people who were masters was 10,000 hours, but that was just the average. Ericsson explains in his book, *Peak: Secrets From the New Science of Expertise,* that the range of time was highly varied. The factor that explained how some individuals took much less time to master a topic was the amount of time they spent in what he termed "deliberate practice." Deliberate practice forces people to the edge of their abilities and necessitates them coming up with new strategies and ways to solve problems or, in other words, forming new mental representations of the problem. Operating on the edge of your capability maximizes the speed at which you acquire additional mental models and further refine your present ones.



Deliberate Practice

The four components of high-quality **Deliberate Practice**:

1. Specific goals to measure against
2. Intensely focused sessions without distraction
3. Immediate feedback on task performance
4. Frequently being pushed outside the comfort zone

Practice Methods + Expert Coaching = Quick Mastery

- For triage & analysis: Sites like **malware-traffic-analysis.net**
  - Real PCAP samples from exploit kits and analysis puzzles with answers
  - Testing yourself against known data, with feedback from senior analysts

SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 194

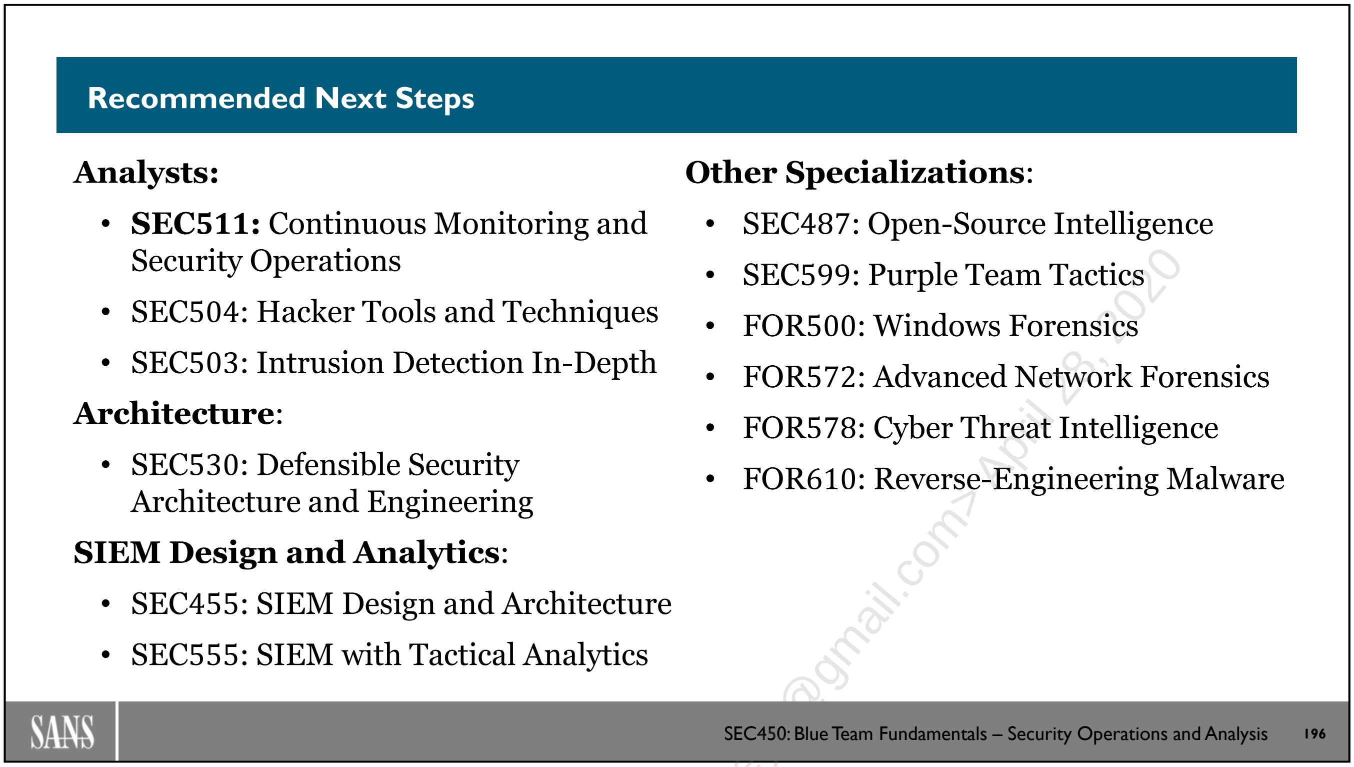
Deliberate Practice

What are the key components of deliberate practice? Number one is a specific and clear goal to reach for. It doesn't have to be a major goal, just an identifiable way of identifying progress. The second is doing highly focused practice sessions absent of distractions. The third is immediate feedback, ideally receiving an immediate qualitative assessment on your performance. Finally, and perhaps most importantly, frequently being pushed out of your comfort zone. Those who never push their boundaries are never forced to come up with ways to get better, and therefore they likely won't. Being pushed to the edge of what you are capable of (but not wildly beyond it) ensures you are forming new skills.

In his research, Ericsson studied this by testing if he could to train a research subject, who could at first only remember a few numbers (short term memory limits), to push far beyond the average 7 item capability. How did he do it? Through multiple sessions where, over time, the subject was able to devise increasingly better mnemonic devices successfully. These representations pushed the numbers into long term memory (sounds familiar from Day 4, doesn't it?) During many sessions, the subject would think they had hit a wall, get frustrated, and eventually had to come up with an idea of how to proceed. After enough focus and thought, sure enough, they were able to come up with a new method over and over. Eventually, Ericsson got the research subject to successfully remember a string of digits 82 numbers long, showing that surprising feats are possible in a relatively short period.

In the years since this study, many more people have attempted this same task, learning from the methods used by the first subject who was the record holder of the time. Today, the longest sessions of numbers remembered are up around 450 digits long! How are people doing it? Not only is the purposeful practice method being used, but now each successive person can learn from the last person's methods, further propelling them into higher numbers. This experiment shows the power not only of deliberate practice, but how having a coach who is an expert in your topic can take you farther, faster. According to Ericsson, deliberate practice plus a knowledgeable coach is the absolute ideal way to speed down the path to mastery of any subject.

How can we apply these principles to information security? For our tasks, the idea would be to break down the constituent parts of our job—parsing PCAPs, validating alerts, seeking data on hosts, and practicing how fast we can collect and interpret the data. If we can perform these tasks over and over in a practice format with known answers, we too can cultivate the type of deliberate practice that will rapidly develop mastery. One website that offers challenges exactly like this is malware-traffic-analysis.net. There are both real examples of malware and engineered challenges posted regularly to the site, and it would make a great starting point for this type of work.



Recommended Next Steps

Analysts:

- **SEC511:** Continuous Monitoring and Security Operations
- SEC504: Hacker Tools and Techniques
- SEC503: Intrusion Detection In-Depth

Architecture:

- SEC530: Defensible Security Architecture and Engineering

SIEM Design and Analytics:

- SEC455: SIEM Design and Architecture
- SEC555: SIEM with Tactical Analytics

Other Specializations:

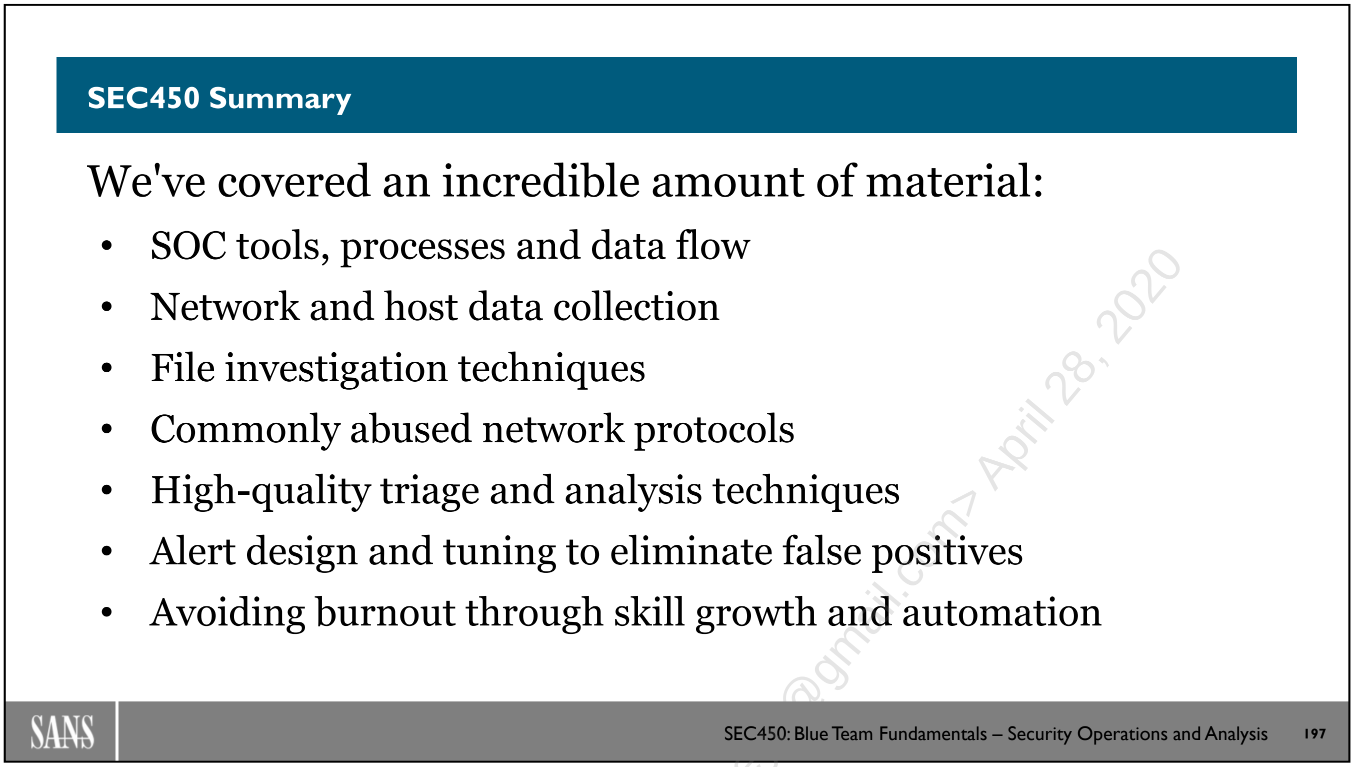
- SEC487: Open-Source Intelligence
- SEC599: Purple Team Tactics
- FOR500: Windows Forensics
- FOR572: Advanced Network Forensics
- FOR578: Cyber Threat Intelligence
- FOR610: Reverse-Engineering Malware

Recommended Next Steps

What classes would make good follow-ups to this course? That depends on which specialization you think you'd like to pursue! One of the best answers that is the next logical step after this course is "SEC511: Continuous Monitoring and Security Operations." Paring with the GIAC GMON certification, this course has been wildly popular over the last few years and is likely the best match for moving into Senior SOC analyst-type work. It teaches additional detail around NSM and CSM, security appliances, network and host hardening, and threat hunting. Another option may be "SEC504: Hacker Tools, Techniques, Exploits, and Incident Handling." A long-time popular class at SANS, SEC504 shows you both the attack and defense side of the coin and teaches incident handling techniques that will lead you in the direction of the GIAC GCIH certification. SEC503 is another long-time favorite of the SANS curriculum. If you love tearing deep into packets, SEC503 and the respective GIAC GCIA cert may be a great match for you.

Beyond that, SANS has several courses specializing in SIEMs—SEC455 which focuses on the use of Elasticsearch specifically and SEC555 which is a SIEM neutral course on writing high quality, tactical analytics. If you are looking at going into network architecture eventually, SEC530 would be a great choice to learn about building a defensible network at all layers of the networking stack and host. SANS also offers plenty of the more advanced and specialty courses. Whether you are interested in purple teaming, OSINT, Windows, Mac, or Memory forensics, Threat Intelligence, or Malware reverse engineering, there is a course that has you covered.[1] There are always new courses on the way, too, so keep watching into the future, because there are several additional blue team courses currently in development!

[1] https://www.sans.org/courses/



SEC450 Summary

We've covered an incredible amount of material:

- SOC tools, processes and data flow
- Network and host data collection
- File investigation techniques
- Commonly abused network protocols
- High-quality triage and analysis techniques
- Alert design and tuning to eliminate false positives
- Avoiding burnout through skill growth and automation

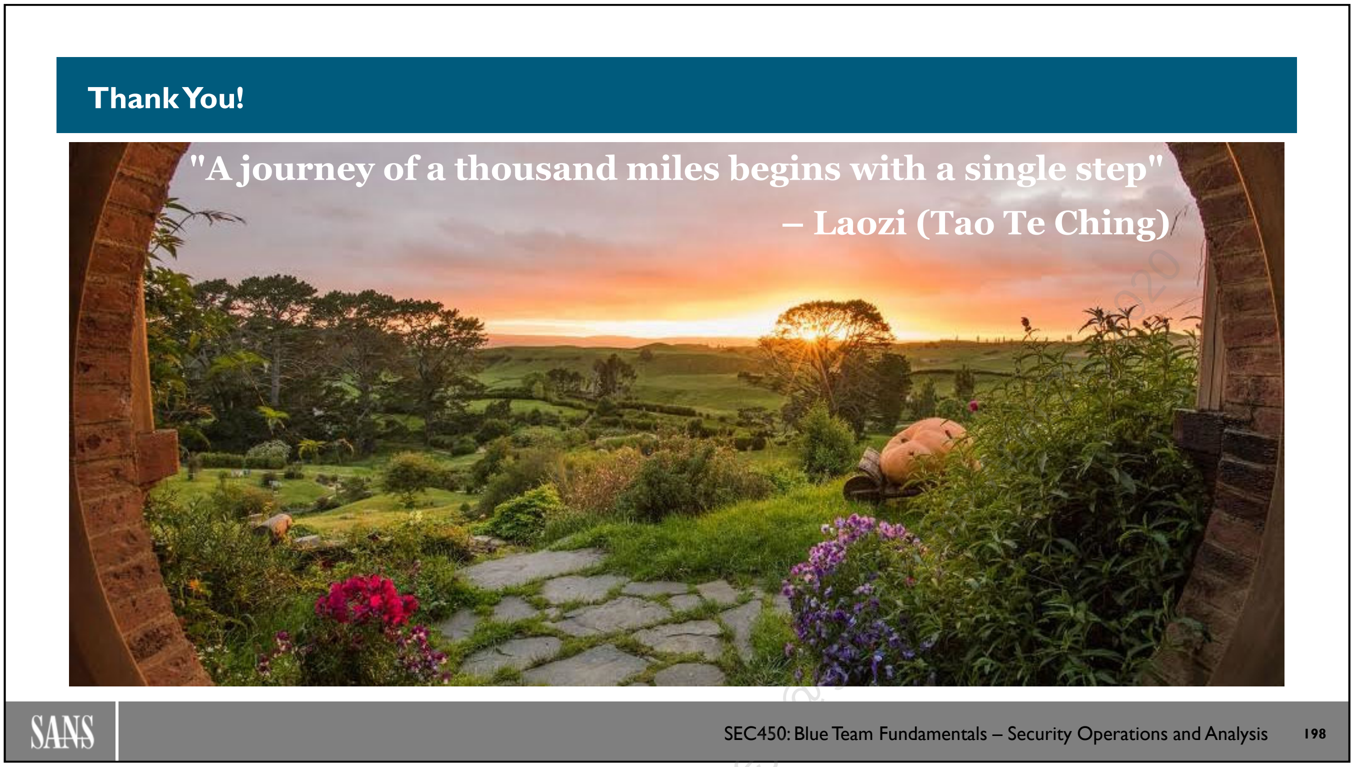
SANS SEC450: Blue Team Fundamentals – Security Operations and Analysis 197

SEC450 Summary

We've covered a *lot* of ground in this class. Throughout these five books, the goal was to give you the best-possible crash course in the tools and processes used in the SOC, and the types of data that are analyzed using them. We've covered the data flow between tools, network, and host monitoring techniques, network protocols, file dissection, high-quality analysis techniques, alert creation and tuning, and strategies to gear your day to avoid burnout. My goal in writing this class was to pass on the information I wish I knew coming into a SOC in an entry-level position years ago, as well as what I've learned along the way about staying engaged and happy in a blue team position. I'm driven to help see to it that everyone in a SOC enjoys their job and doesn't look it as a position to escape as soon as possible, which is why I hit the concepts on human capital today in such detail. While many analysts out there are unhappy with their jobs, they may not be able to put into words why or how to fix it, and it clearly is possible to do when implementing the concepts discussed today.

If nothing else, I hope this class can contribute to making your day-to-day life better by taking the human capital model of SOC operations home and sharing it with management. Done correctly, I believe a career on the blue team can be even more exciting and engaging than the red team and penetration testing jobs that typically gets all the attention. There is no shortage of attacks to stop, and they continue to change every day. Therefore, I have no doubt that the need for cyber defense jobs will grow into the foreseeable future. You've already got a great start, and if you stick with it, you have picked the career that will grow with you and keep you challenged and engaged for a lifetime!

I sincerely hope you've enjoyed the class and hope to see you back again for another blue team course in the future!

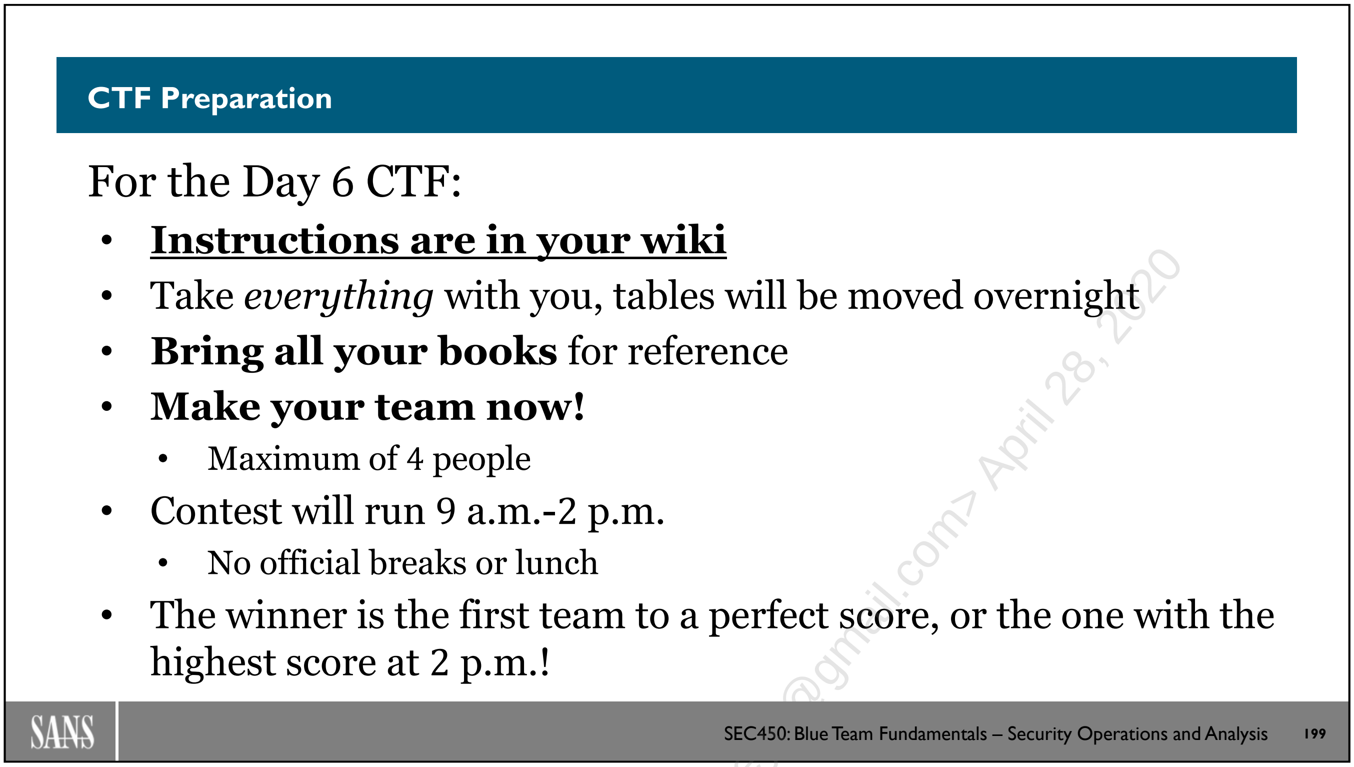


Thank You!

"A journey of a thousand miles begins with a single step"
– Laozi (Tao Te Ching)

Thank You

Thank you for attending SEC450! I hope you have learned new concepts and tools and feel more confident in your journey through information security. No matter where you are in your career, remember that the time dedicated to learning something new each day will sum up to impressive skill and accomplishments. It is this consistency over time, no matter how small, that is key to making long-term progress. As the saying goes "the journey of a thousand miles begins with a single step"…



CTF Preparation

For the Day 6 CTF:

- **Instructions are in your wiki**
- Take *everything* with you, tables will be moved overnight
- **Bring all your books** for reference
- **Make your team now!**
  - Maximum of 4 people
- Contest will run 9 a.m.-2 p.m.
  - No official breaks or lunch
- The winner is the first team to a perfect score, or the one with the highest score at 2 p.m.!

CTF Preparation

There is not a dedicated book for CTF instructions. All information on the CTF is located inside your class wiki in the "CTF Instructions" topic. Please be sure to read it before the contest tomorrow to familiarize yourself with the rules.

This page intentionally left blank.

Index

.acm	3:169
.ax	3:169
.cpl	3:169, 4:162
.dll	3:154, 3:169, 5:58
.doc	1:142, 3:151-152, 3:164, 3:171-173, 3:177, 4:160, 4:162, 5:54
.docm	3:171-172, 4:162
.docx	3:152, 3:164, 3:171-172, 4:162
.drv	3:169
.efi	3:169
.jar	3:169
.msi	3:169
.msp	3:169
.mui	3:169
.ocx	3:169
.scr	3:169, 4:162
.sea	2:95, 3:169
.sfx	3:169
.sys	3:169
.tsp	3:169
.xls	1:142, 3:171, 4:160, 4:162
.xlsm	3:171
.xlsx	3:171
802.1X	5:154-155

A

A Record	1:183, 1:186, 2:45, 2:49-54, 2:56-57, 2:59-62, 2:65-67, 2:72-73, 2:75, 2:84-87, 2:89, 2:91, 2:95-96, 2:112-113, 2:140, 2:182, 3:48, 4:134-135, 4:137, 4:157, 5:30, 5:35, 5:163-165, 5:169-170
A/B Analysis	4:205
AAAA Record	2:49
Accepting the Risk	1:33
Access Control Lists (ACLs)	2:16, 2:23, 5:154, 5:156-159, 5:176
access.log	3:55, 3:78, 3:124
ace	4:162
Actor Characterization	1:120

Advanced Persistent Threat (APT)	1:112, 1:140, 1:142, 1:177, 1:185, 2:27, 2:78, 2:149, 2:200, 3:104, 4:7, 4:9, 4:14, 4:20, 4:52, 4:169, 4:177-180, 4:183, 5:79
Air-gapped networks	5:152
airgap	5:152
Alert Fatigue	1:91, 5:86-87, 5:90, 5:111
Alert queue	4:190-191, 4:199, 5:6, 5:49, 5:75, 5:87-89, 5:92-93
Alert Triage	1:20, 1:88, 1:90, 4:2, 4:5-6, 4:15, 4:18, 4:26-27, 4:49, 4:86-89, 4:117, 4:143, 4:173, 4:175, 4:193, 4:207-209
Alert Validation	4:23-24, 5:102
Alexiou Principle	4:118, 4:121
AlienVault USM	5:37
Amazon AW	1:66, 2:82
Analysis of Competing Hypotheses (ACH)	4:95-96, 4:102-104, 4:106-109
Analytic Sharing	5:77, 5:79, 5:83
Anomali	1:2, 1:4, 1:22, 1:25, 1:50, 1:81-82, 1:90, 1:92-94, 1:96, 1:98, 1:100, 1:123, 1:125, 1:136, 1:150, 1:152, 1:175, 1:190, 2:41, 2:145, 2:202, 3:51, 3:97, 3:116-117, 3:119, 4:146, 5:53, 5:63, 5:84, 5:90, 5:100
anomaly-based alert	1:92-96
Anomaly-centric	1:96
Anonymous	1:138, 1:178, 3:31, 4:134, 4:142
AntiVirus (AV)	1:30, 1:32, 1:45, 1:62, 1:65, 1:67-68, 1:75, 1:96, 1:102, 1:112-113, 1:121, 1:158, 1:164, 2:28, 2:95-97, 2:140, 3:13, 3:22, 3:30, 3:39-40, 3:44, 3:66-67, 3:80, 3:102, 3:165, 3:167, 3:170, 3:175, 3:179-180, 3:182-183, 3:186, 4:12, 4:14, 4:16, 4:56, 4:121, 4:124, 4:144, 4:150-151, 4:159, 4:186, 4:196-197, 5:23, 5:66, 5:68, 5:80-81, 5:109, 5:175-176
Anunak	1:140, 1:177
Any.run	2:141
Apache	1:58, 1:68, 1:84, 1:186, 2:110, 2:112, 2:123, 3:55, 3:78, 3:97, 3:124, 3:168
Apache Struts	1:186, 3:168
Appliance Logging	3:79
Archer	1:104, 1:133, 1:185, 2:25, 2:79, 2:104, 3:104, 3:183, 4:38-40, 4:91, 4:128, 4:133, 4:176, 5:11-13, 5:20-21, 5:24, 5:42, 5:79

ArcSight	1:156, 3:130, 5:37
ASP	1:29, 1:35, 1:44, 1:134, 1:164, 2:115-116, 2:160, 3:19, 3:38, 3:50, 3:84, 3:96, 3:100, 3:104, 3:132, 3:189, 4:70, 4:154, 4:202-203, 5:18, 5:22, 5:66, 5:155
Attack Surface Reduction	3:35, 3:66
Attack Trees	4:50, 4:72, 4:82
Attacker Motivation	1:182, 4:180-181
audit.log	3:78, 3:91
Auditbeat	3:77, 3:88, 3:93
Auditd	3:77-78, 3:88, 3:90-93
Auth.log	3:75-78, 3:86, 3:92, 3:124
Authentication Server (AS)	3:108-109, 3:112-113
Authenticode	3:181-183
Auto Start Extensibility Points (ASEPs)	3:14, 4:124
AutoIt	5:142
Automation	1:2-4, 1:6, 1:8, 1:22, 1:25, 1:27, 1:46, 1:50, 1:81, 1:97, 1:100-102, 1:105-106, 1:110, 1:123, 1:125, 1:149-150, 1:152-153, 1:164-172, 1:174-175, 1:190, 2:3, 2:5, 2:30, 2:42, 2:74, 2:107, 2:109, 2:116, 2:134, 2:163, 2:165, 2:190, 2:203, 2:205, 3:3, 3:5, 3:29, 3:53, 3:69, 3:82, 3:105, 3:107, 3:120, 3:144, 3:146, 3:162, 3:188, 4:3, 4:5, 4:27, 4:49, 4:86, 4:88, 4:117, 4:143, 4:153, 4:173, 4:175, 4:193, 4:207, 4:209, 5:1-6, 5:8-9, 5:17-23, 5:26-27, 5:47, 5:62, 5:85, 5:87-89, 5:110, 5:112, 5:114-120, 5:124, 5:126-132, 5:142, 5:144-146, 5:177, 5:179, 5:195
Autonomous System Number	2:82, 5:36, 5:41
Autonomous System Number (ASN)	2:82, 2:141, 2:160, 4:134, 4:153, 5:36, 5:41-42, 5:44, 5:63, 5:173
Autonomous System Numbers (ASN)	2:82, 2:141, 2:160, 4:134, 4:153, 5:41-42, 5:44, 5:63, 5:173
Azure	1:66, 1:94

B

Backdoor	2:94, 2:146, 2:201, 3:17, 3:45, 3:47, 3:85, 3:101, 3:168, 4:54, 4:65, 4:141, 4:150, 4:177, 4:179, 4:181, 5:100

Bad stress	5:10
Banking Trojans	1:182, 2:157
Base64	2:146-149, 2:171, 2:174, 3:185, 4:11, 4:22
bash	3:99-100, 3:149, 3:164, 3:166, 3:170
bat	3:164, 3:170, 5:44
Bit.ly	4:166, 4:168-169
Blacklist	1:84, 1:92, 1:96, 1:99, 2:84, 2:87, 2:138, 2:160-161, 2:195, 3:39, 3:116, 3:136-137, 3:179, 4:153, 4:164, 5:29, 5:53-54, 5:65-66, 5:68, 5:74, 5:125, 5:175
blacklist-based alerts	1:92
Blue Team	1:1-6, 1:8-9, 1:11-12, 1:14-18, 1:20, 1:22, 1:25-32, 1:36, 1:38-39, 1:47, 1:49-50, 1:53, 1:57, 1:81, 1:91, 1:100, 1:102, 1:104, 1:123, 1:125, 1:132, 1:149-150, 1:152-153, 1:160, 1:163, 1:175, 1:180, 1:190-191, 2:1, 2:3, 2:5-6, 2:17, 2:19, 2:27-28, 2:30, 2:42, 2:74, 2:91, 2:105, 2:107, 2:109, 2:134, 2:143, 2:158-159, 2:162-163, 2:165, 2:190, 2:202-203, 2:205, 3:1, 3:3, 3:5, 3:11, 3:15, 3:23, 3:29, 3:53-54, 3:69, 3:82, 3:97, 3:105, 3:107, 3:120, 3:144, 3:146, 3:162-163, 3:188, 4:1, 4:3, 4:5, 4:27, 4:49, 4:73, 4:81, 4:86, 4:88, 4:91, 4:117, 4:125, 4:143, 4:173, 4:175, 4:179, 4:182, 4:185, 4:193, 4:207-209, 5:1, 5:3, 5:5, 5:26-27, 5:44, 5:47, 5:76, 5:85, 5:112, 5:114, 5:131, 5:144, 5:146, 5:177, 5:179, 5:194-195
Blue Team Truth	1:17-18
BMC Remedy	1:104
Bounded Rationality	4:85
Brute Forcing	2:76, 2:85, 2:87, 4:18
Burnout	1:6, 5:4, 5:7-14, 5:17-18, 5:26, 5:148, 5:195
Business Club	1:177
Byte Orders	3:157

C

Caching Server	2:63, 2:66, 2:98
Carbanak	1:140, 1:177
Carson Zimmerman	1:8, 1:41, 5:22, 5:89
Categorization	1:105, 1:118-120, 1:122, 2:78, 2:138, 3:139-

	140, 3:143, 5:80, 5:93
Centralization	1:51, 1:64, 1:75-76, 1:80, 1:87, 3:50
Channels of Interest	3:66
Chaos Computer Club	1:178
charter	1:29-30, 1:49, 5:42
China	1:177, 1:180
cipher	2:162, 2:173, 4:12
CIS Top 20	3:33
Click Fraud	1:182-183, 4:186
click-fraud	4:186
Client-side Exploits	3:9-10, 3:33, 3:35, 4:144
Closure	1:108, 1:112, 1:117, 1:127, 4:35, 4:75, 4:97, 4:99, 4:129, 5:91, 5:124-125
CNAME Record	2:45, 2:53, 2:56-57, 2:94
CNAME tunneling	2:94
Collaborative Research Into Threats (CRITS)	1:136
Collective Intelligence Framework (CIF)	1:136, 5:44
Command and Control (C2)	1:56, 1:83, 1:113, 1:145, 1:167, 2:79, 2:135-136, 2:142, 2:146-147, 2:152-153, 2:193, 2:202, 3:11, 3:15, 3:26, 3:38, 3:44, 3:159, 3:185, 4:9, 4:16-17, 4:24, 4:26, 4:52, 4:65, 4:82, 4:134, 4:141, 4:145-146, 4:151, 4:154-155, 4:186, 4:196, 5:40, 5:56, 5:58, 5:124, 5:147, 5:154, 5:166, 5:169
Comment Crew	1:185
Common Event Format (CEF)	3:130
Communication plan	1:44
Components of a SOC	1:39
Concept-driven Analysis	4:92-93
CONNECT	2:119
Content-Length	2:123, 2:127
Content-Type	2:114, 2:123, 2:127, 2:173-174
Continuous Security Monitoring (CSM)	1:51, 1:58, 1:64-68, 1:70, 1:76, 1:79-80, 4:58, 4:63, 4:66, 4:172, 5:188, 5:194
Controlled Folder Access	3:35, 3:66
Cookies	2:142, 2:146-147, 3:159, 4:140, 4:155
Cortex	1:109-110, 1:166, 1:170-171, 3:130, 4:164, 5:116, 5:120, 5:124
Courses of Action Matrix	4:55
cpl	3:164, 3:169, 4:162
Critical SOC Information	1:44
Cross-Sector Dependency	1:120

Crouching Yeti	1:185
Crypto mining	3:168
Cryptocurrency	1:182
CTFs	5:182
Cyber Kill Chain	1:129, 2:135, 3:7, 3:26, 3:167, 4:9-10, 4:17, 4:50, 4:52-53, 4:82, 4:183
Cyber Threat Intelligence Definition	1:129
CyberCPR	1:104

D

Dark Hotel	1:185
Data Breach Investigations Report (DBIR)	1:13, 1:34, 1:118
Data flow diagram	1:44
Data Loss Prevention (DLP)	1:68, 1:179, 3:25, 3:30, 3:49, 3:51, 4:11, 4:119
Data-driven analysis	4:92
David Bianco	4:77, 4:80
DDoS Zombie	1:182, 4:186
De-fang	1:134
deb	3:164
Defense In Depth	3:2, 3:5, 3:29-30, 3:52-53, 3:69, 3:82, 3:105, 3:107, 3:120, 3:144, 3:146, 3:162, 3:188, 4:50, 4:52, 4:55-56, 4:58, 5:88
defensible network	1:2, 1:4, 1:22, 1:25, 1:50, 1:52, 1:54, 1:79, 1:81, 1:100, 1:123, 1:125, 1:150, 1:152, 1:175, 1:190, 5:194
DELETE	2:119
Deliberate Practice	5:180, 5:189, 5:191-193
DeMilitarized Zone (DMZ)	1:57-58, 1:65, 1:120, 2:8-10, 2:12, 2:21, 2:23, 2:204, 4:17-18, 5:91
Demisto	1:166
Destover	4:12
Destruction clues	4:12
Detection Maturity Level	4:79, 4:183
DFLabs	1:166
Diamond Model	4:50, 4:60, 4:70-71
dll	3:164, 3:169, 4:162, 5:58
DMARC	2:180, 2:184, 2:186-189
DNS over HTTPS (DoH)	2:105
DNS over TLS (DoT)	2:105

DNS requests/reverse	5:36
DNS Tunneling	2:75, 2:93-97, 2:99, 4:11, 5:69-70, 5:169
DNSSEC	2:105
doc	1:142, 1:173, 2:64, 3:151-152, 3:171-173, 3:177, 4:160, 4:162, 5:54
Docuwiki	1:173
docx	3:147, 3:152-153, 3:164, 3:171-172, 4:162
Domain category	5:35-36, 5:173
Domain creation date	5:36
Domain Generation Algorithm (DGA)	2:80, 5:161, 5:169
Domain Keys Identified Mail (DKIM)	2:55, 2:180, 2:184-189, 4:171
Domain Name System (DNS)	1:52, 1:56, 1:68, 1:75, 1:84, 1:95, 1:167, 1:170, 1:181, 1:183, 2:2, 2:4-5, 2:30-31, 2:36, 2:42-55, 2:57-63, 2:65-67, 2:71-79, 2:82-100, 2:104-109, 2:111-113, 2:128, 2:133-134, 2:140, 2:154, 2:160, 2:163, 2:165, 2:168, 2:176, 2:178, 2:180-183, 2:185, 2:189-190, 2:203-205, 3:58, 3:136, 4:11, 4:15, 4:20, 4:24, 4:111, 4:123, 4:127-128, 4:133-137, 4:139, 4:142, 4:153, 4:157, 4:180, 5:30, 5:32-36, 5:38, 5:42, 5:44, 5:58, 5:63, 5:69-70, 5:80, 5:119, 5:122-123, 5:135, 5:151, 5:156-157, 5:161-167, 5:169-170, 5:173, 5:176
Domain Randomness	2:80
Domain rank	5:36, 5:38
Domain Shadowing	2:76, 2:81, 2:91-92, 4:167, 5:169
Dropbox	2:81, 2:150, 4:11, 4:152, 4:155-156, 4:167
Dumping	3:20-21, 3:36, 4:14, 4:81, 4:150, 5:44
Dunning-Kruger	5:180, 5:187
Dwell Time	1:14, 4:177, 4:179
Dynamic Host Configuration Protocol (DHCP)	1:68, 2:4, 2:36, 2:88, 2:194-195, 4:120-121, 5:80, 5:152-153, 5:156

E

EdgeRouterX	5:184
Eisenhower Matrix	5:189-190
Elastic Stack	1:2, 1:4, 1:22, 1:25, 1:50, 1:81, 1:100, 1:123, 1:125, 1:150, 1:152, 1:157, 1:175, 1:190, 1:193, 5:98
Elasticsearch	1:157, 3:74, 3:77, 3:93, 3:133, 5:82, 5:98,

	5:194
EldoS RawDisk driver	4:12
Electronic Frontier Foundation's (EFF)	4:75, 4:139
ELF	3:164
Email Headers	2:171-172, 2:175, 2:186
Email Spoofing	2:178
Emissary Panda	1:185
Encodings	2:100, 3:153, 3:155-157, 3:159, 3:161
Endpoint Detection and Response (EDR)	1:63, 1:65, 1:75, 2:28, 2:90, 3:48, 3:88-89, 3:93, 4:123-124, 4:147, 5:117, 5:122, 5:175-176
Enhanced Mitigation Experience Toolkit (EMET)	2:28, 3:34-35, 3:66, 5:175
Enrichment	1:19, 1:85, 1:88, 1:90, 1:97, 1:99, 1:101, 1:104-105, 1:109, 1:154-155, 1:167, 1:170, 1:174, 2:78, 2:106, 3:2, 3:4-5, 3:29, 3:53, 3:69, 3:82, 3:99, 3:105, 3:107, 3:120-122, 3:135-139, 3:143-146, 3:162, 3:188, 4:13, 4:23, 4:66, 4:111-112, 4:153, 5:2, 5:5, 5:27, 5:29, 5:32, 5:35-39, 5:41-47, 5:59, 5:63, 5:65, 5:85, 5:88, 5:90, 5:94, 5:96, 5:108-109, 5:111-112, 5:114, 5:119-120, 5:124, 5:130-131, 5:144, 5:146, 5:177, 5:179
Equation Group	1:185, 2:27, 2:191-192
Equifax	1:186, 1:188, 2:23, 4:18, 4:178
ETERNALBLUE	2:27, 2:193, 3:44, 3:168, 4:17-18
Event Charting	4:113-114
Event ID	1:97, 3:59, 3:61, 3:63, 3:68, 3:70, 3:83-85, 3:87, 3:89, 3:96-99, 3:101-103, 3:115-117, 3:119, 3:125, 3:140, 4:121, 5:81
Event Matrix Chart	4:115
example metrics	1:47
exe	1:62, 1:67, 1:84, 2:63, 2:160, 2:201, 3:18, 3:21, 3:36, 3:42, 3:44, 3:65, 3:91, 3:98, 3:147, 3:153, 3:164-165, 3:169-171, 3:176-177, 3:182, 4:12, 4:151, 4:162, 5:33, 5:41, 5:54, 5:59, 5:80, 5:96, 5:142, 5:160
Exfiltration	1:56, 1:186, 1:188, 2:21, 2:76, 2:93, 2:97, 2:135-136, 2:197-199, 3:6, 3:11, 3:24-27, 3:38, 3:49, 4:8, 4:11, 4:15, 4:26, 4:82, 4:145-146, 4:152, 5:69, 5:174
Exfiltration clues	4:11
Expandurl.net	4:168

Explicit Knowledge	4:91
Exploit Kits	1:92, 1:181, 2:135, 2:155-157, 4:144, 5:192
Exploit Protection	3:35, 3:66
eXtensible Markup Language (XML)	1:73, 1:78, 1:138, 2:201, 3:56, 3:60, 3:62-65, 3:68, 3:98, 3:125, 3:129, 3:134-135, 3:171-172, 4:162, 5:77

F

Falco	3:93
False Positive	1:37, 1:88, 1:90-94, 1:97-98, 1:112, 1:117, 1:139, 1:161-162, 1:179, 2:77, 2:80, 2:83, 2:96-97, 2:156, 3:15, 3:51, 3:137, 3:182, 4:12, 4:20, 4:23-24, 4:158, 4:197, 5:2, 5:5, 5:7, 5:27, 5:29, 5:36, 5:46-52, 5:62, 5:75-76, 5:84-87, 5:90-92, 5:94-97, 5:102-107, 5:109, 5:111-112, 5:114, 5:125, 5:128, 5:131, 5:144, 5:146, 5:177, 5:179, 5:195
Fancy Bear	1:185, 4:169
Fast Incident Response (FIR)	1:104
FBI Infragard	1:34
Field Length	5:36
File Analysis	2:142, 2:150, 3:4, 3:163, 4:148
File Integrity Monitoring (FIM)	3:30, 3:44-45, 3:47, 3:93
File Transfer Protocol (FTP)	1:32, 1:56, 1:68, 2:4, 2:22, 2:36, 2:198, 2:204, 3:31, 3:38, 3:97, 4:155, 5:40, 5:100, 5:173
flat network	2:23, 2:27-28, 5:154
Flatline	5:29, 5:65, 5:69
Forwarding Server	2:63, 2:65, 2:98
France	1:177
Frode Hommedal	4:176, 4:183-184
Functional Impact	1:120

G

Galaxy	1:138, 1:140, 1:146, 1:148
GeoIP information	5:36
Geolocation	5:38-41
Getlinkinfo.com	4:168
git	3:92

Global Information Security Workforce Study (GISWS)	1:7
globalHell	1:178
Glue Record	2:62, 2:66-67
Go-Audit	3:77
goo.gl	4:168
Good stress	5:10
Google Safe Browsing	2:138-139
Graph Thinking	4:72, 4:74
Greasemonkey	5:141
Grounded Theory	5:11

H

Hackers-for-Hire	1:179, 1:181
HackMD	1:173
Hacktivist	1:28, 1:178, 1:180
Hashing	1:137, 3:43, 3:179, 3:183
HEAD	2:119, 2:123
HIDS/HIPS	1:65, 1:68, 3:30, 3:45
Host Firewalls	3:30, 3:37-38, 3:94, 5:159
Host Intrusion Detection Systems (HIDS)	1:65, 1:68, 3:30, 3:44-45, 4:123, 5:63
Host Intrusion Prevention Systems (HIPS)	1:65, 1:68, 3:30, 3:44-45, 4:121, 4:124, 5:175-176
HP NetStream	1:73
hta	3:170, 3:173, 4:162
HTTP Response Codes	2:124
HTTP/2	2:105, 2:130-132
HTTP/3	2:132
Human Capital	5:11-14, 5:16-22, 5:86, 5:115, 5:148, 5:195
Human Capital Model	5:11-12, 5:17-18, 5:20, 5:115, 5:148, 5:195
Hybrid-analysis.com	1:67, 2:137, 2:141, 3:151, 3:159-160, 3:184, 4:128, 4:149
HyperText Transfer Protocol (HTTP)	1:56, 1:59-61, 1:68, 1:70, 1:84, 1:95, 1:168, 2:2, 2:4-5, 2:22, 2:30-31, 2:36, 2:38, 2:42, 2:44, 2:74, 2:105, 2:107, 2:109-114, 2:117-121, 2:123-136, 2:140-142, 2:146-149, 2:151-153, 2:155-156, 2:158, 2:163-165, 2:171, 2:174, 2:190, 2:198, 2:201, 2:203-205, 3:27, 3:37, 3:75, 4:15-16, 4:25, 4:51, 4:134, 4:152, 4:154-155, 4:157, 4:170, 5:31, 5:33-34, 5:40, 5:43-45, 5:54-56, 5:59,

	5:95, 5:127, 5:151, 5:159, 5:161, 5:167, 5:173-174, 5:176
HyperText Transfer Protocol Secure (HTTPS)	2:2, 2:4-5, 2:30, 2:42, 2:74, 2:105, 2:107, 2:109, 2:111, 2:134, 2:158-159, 2:163-165, 2:169-170, 2:190, 2:201, 2:203-205, 3:27, 3:37, 3:181, 4:121-122

I

IDS Alert Triage	4:15
IE PassView	4:150
IIS	2:110, 2:112, 2:123
Impostor Syndrome	5:187
Incident Management System (IMS)	1:2, 1:4, 1:22, 1:25, 1:50, 1:81, 1:83, 1:85, 1:89-90, 1:100-102, 1:104-106, 1:108-109, 1:111-112, 1:114, 1:117-118, 1:121-125, 1:132, 1:135, 1:137, 1:141, 1:149-150, 1:152, 1:165, 1:172, 1:174-175, 1:190, 4:34, 4:103, 4:147, 4:164, 4:197, 5:54, 5:120, 5:124
Incident Response Cycle	4:50, 4:58, 4:66, 4:68, 4:82
Incident Response Plan	1:44
India	1:177, 1:179, 5:181
Indicator Types	4:54, 4:158
Information Impact	1:120
Information Loss Prevention(ILP)	1:96
Infragard	1:34
Intelligence Definition	1:127, 1:129
Intentional Evidence	4:119, 4:123
Internationalized Domain Names (IDNs)	2:75, 2:100-104, 2:106, 4:167
Internet Control Message Protocol (ICMP)	2:4, 2:197, 2:204, 3:95, 5:158
Internet Relay Chat (IRC)	1:95, 2:36, 4:154, 5:159
Intrusion Detection System (IDS)	1:30, 1:58-59, 1:61, 1:68, 1:70, 1:72-73, 1:75-76, 1:83, 1:87-88, 1:90, 1:92, 1:95-96, 1:102, 1:112, 1:138, 2:26, 2:29, 2:36-37, 2:90-91, 2:128, 2:137, 2:141, 2:159, 2:195, 3:25, 3:93, 3:139, 3:141, 3:165, 3:167, 4:15, 4:56, 4:77, 4:154, 4:191, 4:197, 5:28, 5:31, 5:41, 5:44, 5:57, 5:80, 5:91, 5:107, 5:120, 5:128, 5:158, 5:188
Intrusion Discovery	4:2, 4:5, 4:27, 4:49, 4:86, 4:88, 4:117, 4:143, 4:173, 4:175-176, 4:192-193, 4:207-

	209
Intrusion Prevention System (IPS)	1:52, 1:68, 1:96, 2:28-29, 2:128, 2:159, 3:44, 4:56, 4:82, 5:107, 5:156, 5:173-176
Inveigh	4:150
IPVoid	2:138-139
Iran	1:177, 1:180
Is.gd	4:168
ISAC	1:34
Israel	1:177

J

Java	1:95, 2:110, 2:114-115, 2:117, 2:141, 2:155-156, 3:129, 3:147, 3:164, 3:166, 3:169, 3:174, 3:185, 4:162, 5:132, 5:141
JavaScript Object Notation (JSON)	1:73, 1:78, 1:138, 1:144, 1:157, 1:168, 3:65, 3:72, 3:77, 3:93, 3:124-125, 3:129, 3:131, 3:134-135, 5:77, 5:120, 5:123-124, 5:126-127
Jflow	2:31-32
js	1:166, 2:151, 3:153, 3:170, 4:162
JSP	2:115
Juniper's J-Flow	1:73

K

Kansa	4:91, 4:124, 5:123, 5:129
Kerberos	1:68, 2:36, 3:2, 3:5, 3:29, 3:53, 3:69, 3:82, 3:105, 3:107-120, 3:144, 3:146, 3:162, 3:188-189, 4:121
Kerberos Authentication Ticket (TGT)	3:109, 3:112-117, 3:119
Kern.log	3:75, 3:78, 3:124
Kernel Exploitation	3:19
Key Distribution Center (KDC)	3:108-114
Kibana	1:157-159, 5:98-99
Knowledge Database	1:101-103, 1:106-107, 1:173
Kolide	3:93
Komand	1:166
Krebs	1:34, 1:177, 1:180, 1:182-183

L

Last-Modified	2:123, 2:125
Layer 7	1:51, 1:58-62, 1:68, 1:79, 2:20-22, 2:31, 2:35-36, 2:41, 2:113, 2:132, 4:51, 4:139, 4:146, 4:152, 4:154-155, 5:173
Lazarus Group	1:185, 1:187, 2:150
Level Seven Crew	1:178
Levels of Threat Intelligence	4:60-61, 4:83
Link Analysis	4:96, 4:110-111, 4:113
Linux AppArmor	3:41
Linux Log Path	3:76
List Thinking	4:50, 4:73
Lizard Squad	1:178
lnk	3:164, 3:175-176
Location of Observed Activity	1:120
Log Agents	1:66, 1:78, 1:80, 3:60, 3:73, 3:124-126, 3:134
Log Enrichment	1:155, 3:2, 3:5, 3:29, 3:53, 3:69, 3:82, 3:105, 3:107, 3:120, 3:136, 3:144-146, 3:162, 3:188, 5:35, 5:108
Log Event Extended Format (LEEF)	3:130
Log flow diagram	1:44
Log Normalization	3:121, 3:139-140
Log Pipeline	3:121, 3:126, 3:128
Logging Facility	3:75
Logging Pipeline	3:121-122
Login Failures	3:87
LogRhythm	1:156, 3:125
Logstash	1:157, 3:93, 5:37
Long-Term Memory (LTM)	4:28, 4:36-38, 4:41-45, 4:47
Lulzsec	1:178

M

M-Trends	1:14, 1:34
macOS Gatekeeper	3:41
MACsec	5:154
Magic Bytes	1:62, 3:148, 3:150-152, 3:161, 3:169, 3:173, 3:178
Malicious Insiders	1:178-179, 3:49, 3:51
Maltego	4:111-112

Malware Information Sharing Platform (MISP)	1:2, 1:4, 1:22, 1:25, 1:50, 1:81, 1:100, 1:110, 1:123, 1:125, 1:133, 1:136-145, 1:147-148, 1:150-152, 1:171, 1:175, 1:185, 1:190, 4:131, 5:44, 5:53-54, 5:79, 5:83
Mandiant	1:14, 1:34, 4:9-10, 4:26, 4:77, 4:82, 4:84-85
McAfee ESM	5:37
Metadata	1:75, 1:144, 1:146-147, 2:31, 2:34-38, 2:41, 2:90, 2:120, 2:159, 2:172, 2:202, 3:48, 3:55, 3:57, 3:59, 3:70, 3:76, 3:88, 3:153, 3:159, 3:185, 4:22, 4:72, 4:123, 4:134, 4:147-148, 4:152, 4:155, 4:197, 4:206, 5:53, 5:56, 5:80, 5:84
Metrics	1:47-49, 1:105-107, 1:112, 1:117-118, 1:121-122, 1:163, 1:188, 4:197, 4:206, 5:7, 5:17-18, 5:22-24, 5:91, 5:126
Micro-Automation	5:132
Mimikatz	2:193, 3:21-22, 3:36, 4:74, 4:150, 5:100
Mirai	1:183
Mirror	2:17-19, 2:29, 2:35, 4:205, 5:126, 5:185
MISP	1:2, 1:4, 1:22, 1:25, 1:50, 1:81, 1:100, 1:110, 1:123, 1:125, 1:133, 1:136-148, 1:150-152, 1:171, 1:175, 1:185, 1:190, 4:131, 5:44, 5:53-54, 5:79, 5:83
MITRE ATT&CK	1:138, 1:161, 2:141, 2:149, 3:20, 4:77, 4:80-81, 4:145, 5:64, 5:80
Modbus	2:36
Modern Defense Mindset	1:15
Moloch	1:70, 1:73, 1:77, 2:35, 2:37-38, 2:40, 4:22, 4:147, 4:155
Mosaic Theory of Analysis	4:93-94
MS14-068	3:168
Museum Security	2:25-26
MX Record	2:48, 2:57, 2:166, 2:168, 2:182
MySQL	2:36, 3:74, 3:78
mysqld.log	3:78

N

NAC	1:180, 2:27, 2:79, 2:193, 3:168, 4:12, 4:35, 4:108, 4:121-122, 4:149, 5:15, 5:22, 5:103, 5:122, 5:150, 5:152-154

NetFlow	1:44, 1:58-60, 1:68, 1:70-71, 1:73, 1:75-78, 1:154, 2:26, 2:29, 2:31-36, 2:41, 2:90, 2:98, 3:25, 3:139, 4:11, 4:22, 4:123, 4:147, 5:44
NetStream	1:73, 2:31-32
NetWitness	1:73, 1:77, 1:166
NetWitness Orchestrator	1:166
Network Address Translation (NAT)	2:7, 2:18
Network diagram	1:44, 2:13, 5:6
Network Protection	3:35, 3:66
Network Security Monitoring (NSM)	1:38-40, 1:49, 1:51-54, 1:56-59, 1:61, 1:63-64, 1:68, 1:70, 1:76, 1:79-80, 1:130, 2:129-130, 2:132, 2:158, 2:162, 2:199, 2:202, 2:204, 4:22, 4:58, 4:63, 4:66, 4:119, 4:172, 5:184, 5:188, 5:194
NetworkPassView	4:150
nfcapd	2:33
nfdump	1:73-74, 2:32-33
NGINX	2:110, 2:112, 2:123, 2:125, 2:127, 2:143
NirSoft	4:150
Node Red	5:127
Norton Safeweb	2:138
NotPetya	1:144-148, 2:27, 2:146, 2:193, 4:12, 4:181, 4:184, 5:150, 5:154
NS Record	2:43, 2:46-48, 2:52, 2:55, 2:57, 2:60-62, 2:67, 2:91, 2:176, 2:181, 2:189, 4:24, 4:128
NSA WALKOFF	1:166
NT LAN Manager (NTLM)	2:36, 3:118
NXDOMAIN	2:51-52, 2:87

O

Observables	1:105, 1:109-111, 1:116, 1:122, 4:146, 4:158, 4:195, 4:204, 5:53, 5:119-120, 5:124
Observe Orient Decide Act (OODA)	4:50, 4:60, 4:62-65, 4:68, 4:71, 4:82-85
Observed Activity	1:120
Office of Personnel Management (OPM)	1:188, 2:78, 4:178
OneDrive	2:81, 2:150, 4:51, 4:151, 4:155-156, 4:167
OneNote	1:173
Operational	1:5-6, 1:11, 1:102, 1:130, 1:153, 1:180, 1:185, 2:23, 2:138, 3:50, 3:66-67, 3:89, 3:102-103, 3:110-112, 3:165, 4:60-61,

	4:69, 4:85, 4:125, 4:128, 4:133-135, 4:138, 4:166, 4:180-181, 4:203, 5:2, 5:4-5, 5:17-22, 5:27, 5:46-47, 5:85, 5:112, 5:114-115, 5:131, 5:143-144, 5:146, 5:177, 5:179
Operational Efficiency	5:2, 5:4-5, 5:17-19, 5:21-22, 5:27, 5:47, 5:85, 5:112, 5:114-115, 5:131, 5:143-144, 5:146, 5:177, 5:179
Operational Security (OPSEC)	1:180, 2:99, 2:138, 3:66, 4:2, 4:4-5, 4:27, 4:49, 4:65, 4:86, 4:88, 4:117, 4:125, 4:127-128, 4:131-136, 4:138-139, 4:141-143, 4:166, 4:173, 4:175, 4:188, 4:193, 4:207-209
OPTIONS	2:119
OSQuery	3:93

P

Page2Images.com	2:137
PassiveDNS	2:85, 2:98
PDF	1:95, 2:123, 3:34, 3:147, 3:149-152, 3:164, 3:166, 3:170, 3:174-175, 3:178, 4:29, 4:162
pdf	1:8, 1:41, 1:61, 1:83, 1:110, 1:130, 1:140, 1:163-164, 1:187, 2:17, 2:25, 3:104, 3:130, 3:147, 3:152, 3:159, 4:9-10, 4:22, 4:29, 4:53-55, 4:70, 4:110, 4:113, 4:133, 4:162-163, 4:179, 4:188, 4:205, 5:11-12, 5:22, 5:54, 5:89
Peer review	4:194, 4:200, 4:203, 4:206, 4:208
Perfect Forward Secrecy (PFS)	2:162
Permissible Action Protocol (PAP)	4:127, 4:131, 4:134, 4:142
PfSense	2:15, 5:80, 5:184-185
Phantom (Splunk)	1:156, 1:166, 3:125, 3:141, 5:37, 5:82
Phishing	1:108, 1:113, 1:115-117, 1:120-121, 1:142, 1:160, 1:167, 1:171, 1:179, 1:181-182, 1:184, 2:101-102, 2:104, 2:106, 2:115, 2:135-137, 2:150, 2:191, 2:193, 2:204, 3:8, 3:46, 3:151, 3:166-167, 3:173-175, 3:177, 4:18-19, 4:115, 4:125, 4:133-134, 4:139, 4:144, 4:146, 4:152, 4:156, 4:162, 4:166-170
Phishing Sites	1:182, 2:135, 2:137
Phishtank.com	2:137
PHP	1:61-62, 1:95, 2:115-116, 2:126-127, 2:145,

	2:153, 3:45, 3:100, 3:159, 4:170
PICERL	4:58, 4:68, 4:82, 4:85
Plug and Play	3:99, 5:126
PlugX	1:188, 4:151
Points of visibility	1:44, 2:6, 2:29, 2:204
Port 22	3:27, 3:86
Port 25	2:169, 3:27
Port 443	1:67, 2:132, 2:160, 3:27
Port 445	2:191, 2:193
port 587	2:169
port 5985	2:201
Port 80	2:111-113, 3:27, 4:154, 5:43, 5:159-160
port 993	2:170
Post-Exploitation	1:15, 2:202, 3:7, 3:11, 3:17, 3:28, 3:48, 3:189, 4:26, 4:77, 4:145, 4:172
Post-exploitation tactics	3:11, 3:48, 3:189
Potential Impact	1:120
PowerShell	1:74, 2:4, 2:201, 2:204, 3:12, 3:18, 3:22-23, 3:44, 3:46, 3:57-58, 3:63, 3:66, 3:88, 3:103, 3:125, 3:147, 3:164, 3:166, 3:173, 3:176, 4:124, 4:150, 5:79, 5:82, 5:96, 5:123, 5:129, 5:160
PowerShell Remoting	2:4, 2:201, 2:204, 3:23, 3:46, 4:150, 5:123
PowerUp	3:18
pre-classification	4:13
Pre-Exploitation	3:28
Privileged Access Workstations (PAWS)	3:30, 3:46
Proxy Log	1:61, 1:68, 1:70, 1:76, 1:108, 2:117-118, 2:128, 3:133, 3:141, 4:11, 4:119-121, 4:123, 4:150, 5:63, 5:75
ps1	2:151, 3:153, 3:170, 5:11-12
PSExec	2:193, 3:23, 4:150, 5:81
PTR Record	2:45, 2:51, 2:53-54, 2:73, 2:75, 2:82, 2:176, 5:42
Punycode	2:101-104
Purple Team	1:12, 1:162, 5:14, 5:188, 5:194
PUT	2:119
Pyramid of Pain	4:50, 4:77, 4:79-82, 4:84-85, 4:183, 5:53, 5:56
Python	1:111, 1:138, 2:115-116, 3:164, 3:166, 3:170, 5:37, 5:126, 5:129

Q

QRadar	1:156, 3:125, 3:130, 5:37, 5:74, 5:82
Quick UDP Internet Connections (QUIC)	2:132-133

R

RADIUS	2:36
Randomness measure	3:137, 5:36, 5:44
Ransomware	1:138, 1:144-145, 1:179-180, 1:182, 2:146, 3:35, 3:168, 4:149, 4:158
rapid response	1:15
rar	4:162
Recoverability	1:120
Recursive Server	2:63, 2:65
Red Team	1:12, 1:14, 1:39, 3:104, 4:96, 4:200, 4:203, 4:205, 5:14, 5:195
Redmine	1:104, 5:76
Referer	2:113, 2:120-122, 2:126, 2:133, 2:142, 2:150, 5:31, 5:33, 5:59, 5:161
regexr.com	5:61
Regular Expressions	1:158, 3:70, 3:72, 3:116, 3:128, 4:54, 5:53, 5:56-57, 5:61, 5:84
Remediation	1:91, 1:164, 1:167, 1:188-189, 3:11, 3:13, 3:48, 4:34, 4:125, 4:179, 4:182, 4:206, 5:23, 5:128
REMnux	3:174, 4:138
Remote Desktop Protocol (RDP)	1:56, 2:36, 2:200, 3:9, 3:23, 3:46, 3:83
Remote FrameBuffer (RFB)	2:36
Request Tracker for IR (RTIR)	1:104
Resilient	1:104
Responder.py	4:150
Response Policy Zones (RPZ)	5:162-167, 5:169
REST API	1:137, 2:116, 5:37, 5:44-45, 5:137
RFC 3954	2:32
RFC 7001	2:180, 2:184
RFC5424	3:72, 3:127
Richard Bejtlich	1:52, 1:54, 1:56, 2:31, 2:35
Risk Appetite	1:30-31
RiskIQ	2:78, 2:92, 4:134, 4:137, 5:44
rpm	3:164
Rsyslog	3:71, 3:74-76, 3:124, 3:126

rtf	3:150, 3:173, 4:162
RTF	3:150, 3:164, 3:166-167, 3:173, 4:162
Ruby on Rails	2:115
Rule Logic	2:20, 5:65, 5:79, 5:97
RunAs	3:85
Russia	1:177, 2:27
Ryan Stillions	4:79, 4:183

S

Sandboxing	2:136, 2:141-142, 2:154, 2:156, 2:159, 4:128, 4:147
Sandworm	1:185
Scheduled Task	3:13-14, 3:18, 3:98, 4:124, 4:145-146
scr	3:164, 3:169, 4:162
Sdelete	4:12
Secure SHell (SSH)	1:56, 1:66, 1:68, 2:4, 2:22, 2:36, 2:191, 2:199-201, 2:204, 3:23, 3:26-27, 3:37-38, 3:86-87, 3:97, 4:152, 4:154, 5:40, 5:80, 5:100, 5:127, 5:155, 5:173
Secure Sockets Layer (SSL)	1:95, 1:188, 2:36, 2:55, 2:119, 2:133, 2:158-162, 4:51, 4:54, 4:152, 4:155-156, 5:44, 5:173-174
Security Information and Event Management (SIEM)	1:2, 1:4, 1:22, 1:25, 1:45, 1:50, 1:57, 1:64-66, 1:68, 1:70-73, 1:75-78, 1:80-81, 1:83-90, 1:97-98, 1:100-102, 1:104-105, 1:111, 1:123, 1:125, 1:132, 1:135, 1:141, 1:150, 1:152-157, 1:160, 1:162-163, 1:172, 1:174-175, 1:190, 1:193, 2:32, 2:35-36, 2:41, 2:77-78, 2:87, 2:123, 2:143, 2:145, 2:152, 2:159, 3:38, 3:50, 3:58, 3:60, 3:65, 3:68, 3:71-72, 3:77, 3:79-80, 3:86, 3:89, 3:91, 3:94, 3:99, 3:102, 3:121-127, 3:129-133, 3:135-143, 3:189, 4:13, 4:15, 4:34, 4:118-119, 4:121, 4:147, 4:155, 4:158, 4:164, 4:190-191, 5:17, 5:28-30, 5:32, 5:35-38, 5:41, 5:44-46, 5:54, 5:56-57, 5:63, 5:65, 5:74, 5:77-79, 5:82-83, 5:91, 5:94, 5:108, 5:120, 5:123-124, 5:149, 5:188, 5:194
Security Operations	1:1, 1:5-6, 1:8-12, 1:15, 1:27, 1:41, 1:53, 1:101, 1:153, 1:163, 2:1, 3:1, 4:1, 4:61, 4:63, 4:68-69, 5:1, 5:174, 5:194

Security Orchestration, Automation and Response (SOAR)	1:102, 1:105, 1:107-109, 1:135, 1:141, 1:153, 1:165-170, 1:172, 1:174, 5:23, 5:117-130, 5:132
SecurityOnion	1:73
segmentation	2:16, 2:23, 2:27-28, 5:154
Sender Policy Format (SPF)	2:55, 2:166, 2:168, 2:178, 2:180-184, 2:186-189, 4:171
Sending spam	1:182
Sensitivity	1:32, 5:48, 5:103-105, 5:107-108
Sensory Information Storage (SIS)	4:37
Server Message Block (SMB)	1:56, 1:68, 2:4, 2:36, 2:151, 2:191-193, 3:9, 3:23, 3:31, 3:37-38, 3:44, 3:83, 3:168, 4:54
Service-side	1:184, 2:193, 3:9-10, 3:44, 4:154
ServiceNow	1:104
Session Initiation Protocol (SIP)	2:36
Set-Cookie	2:123, 2:127
Severity Levels	3:75
sFlow	2:31
Shamoon	4:12
Sharepoint+Word	1:173
ShellShock	3:168, 4:18
Short-Term Memory (STM)	4:37, 4:42-43
shred	4:12
SIEMplify	1:166
Sigcheck	3:182
Sigma	5:78-80, 5:82-84
signature	1:35, 1:41, 1:82, 1:88, 1:92-96, 1:99, 2:22, 2:37, 2:80, 2:148, 2:151, 2:159, 2:161, 2:180, 2:185, 2:187, 2:202, 3:39, 3:41-42, 3:44, 3:88, 3:97, 3:130, 3:137, 3:150-151, 3:163, 3:171, 3:178, 3:181-183, 3:186-187, 4:20, 4:58, 4:146-148, 4:172, 5:57, 5:68, 5:77, 5:79, 5:81-82, 5:95, 5:107, 5:175
Signature-Centric	1:96
Simple Mail Transfer Protocol (SMTP)	1:68, 2:2, 2:4-5, 2:30-31, 2:36, 2:42, 2:57, 2:74, 2:107, 2:109, 2:134, 2:147-148, 2:163, 2:165-166, 2:169-171, 2:173, 2:176-179, 2:186-187, 2:189-190, 2:203-206, 3:27, 4:15, 4:164, 5:151, 5:161, 5:171-172, 5:176
Sinkhole	4:15, 5:167
SmartScreen	3:35

Snoopy Logger 3:77, 3:92

Snort 1:70, 1:73, 1:78, 1:87, 1:95, 1:112, 1:138, 2:37, 2:156, 3:55, 3:124, 4:15-16, 5:31, 5:63, 5:77, 5:79-80, 5:120

SOA Record 2:59, 2:85

SOC Poison 5:10

Social Media 1:178, 1:182, 3:185, 4:128, 5:93

SOCKS 2:36

Software Defined Networking (SDN) 2:13

Sony Pictures 1:187

Source Code Repositories 1:102-103

Sources visiting same destination 5:36

Spam 1:30, 1:167, 1:181-182, 2:55, 2:57, 2:73, 2:79, 2:84, 2:137, 2:169, 2:172, 2:178, 2:189, 3:169, 3:171-172, 3:184, 4:125, 4:160, 4:162, 4:169, 4:180, 4:186, 5:41, 5:87-88, 5:121, 5:171-172

Spamming 1:182, 2:57

SPDY 2:130, 2:132-133

Specificity 3:81, 5:48, 5:73, 5:84, 5:103-105, 5:107-109

Spike 2:87, 5:29, 5:65, 5:69, 5:84

Splunk 1:156, 1:166, 3:125, 3:141, 5:37, 5:82

srm 4:12

SRV Record 2:58

SSL/TLS Decryption 2:159, 5:174

Start of Authority (SOA) 2:46, 2:48, 2:59, 2:85

steering committee 1:29-31, 1:49

Strategic 1:6, 1:13, 1:26, 1:39-40, 1:79, 1:102, 1:126, 1:184, 4:6, 4:53, 4:60-61, 4:69, 4:80, 4:84, 4:179-181, 4:192, 5:171

Stress Types 5:10

String Encoding 3:155

Strings 2:37, 2:46, 2:101, 2:148-149, 3:59, 3:148, 3:153-154, 3:156-161, 3:173-174, 3:178, 3:181, 3:185-187, 4:16, 4:148-149, 5:80

Structured Analysis 4:2, 4:5, 4:27-30, 4:49, 4:66, 4:86, 4:88, 4:95-96, 4:109, 4:116-117, 4:143, 4:173-175, 4:193, 4:200, 4:205, 4:207, 4:209

Struts 1:186, 3:168

Stub Resolver 2:63, 2:65

Stuxnet 3:43, 3:175, 3:183

Suricata 1:70, 1:73-74, 1:77, 1:138, 2:31, 2:36-37,

	2:98, 3:58, 4:22, 4:147, 4:155, 5:31
SWIFT	1:179, 4:181
Swimlane	1:166
Switched Port ANalyzer (SPAN)	2:17
Syria	1:177-178
Syrian Electronic Army	1:178
sys	3:164, 3:169
Sysdig Inspect	3:93
Syslog	1:70, 1:73, 1:75, 1:78, 1:157, 2:36, 3:50, 3:55, 3:70-79, 3:81, 3:86, 3:92, 3:95, 3:124-128, 3:130-131, 3:134-135, 5:77, 5:124
Syslog Disambiguation	3:71
Syslog-ng	3:71, 3:74, 3:76, 3:126
Syslogd	3:71, 3:74-75, 3:126
Sysmon	1:65, 1:84, 3:58, 3:67, 3:77, 3:88-89, 3:125, 4:124, 4:147, 5:81

T

Tacit Knowledge	4:91-92
Tactical	1:6, 1:39-40, 1:86, 1:102, 1:126, 3:37-38, 3:121, 3:137, 4:60-61, 4:69, 4:125, 4:178, 4:180-181, 4:184, 5:156, 5:158-160, 5:171, 5:194
Tactics, Techniques, and Procedures (TTPs)	1:12, 1:129-130, 1:140, 1:180, 4:53, 4:60, 4:78, 4:176, 4:179, 4:183-184, 4:192, 4:198
Tampermonkey	5:141
TeamViewer	3:9, 3:97, 4:150
Text-Expanders	5:135
Textarea Cache	5:140
The Dark Overlord	1:177
The Onion Router (TOR)	4:138-140, 4:142, 4:149
TheHive	1:2, 1:4, 1:22, 1:25, 1:50, 1:81, 1:90, 1:100, 1:104, 1:108-111, 1:113-117, 1:123-125, 1:137, 1:141, 1:150, 1:152, 1:166, 1:170-171, 1:175, 1:190, 4:164, 5:116, 5:120, 5:124, 5:140
Thinking Fast and Slow	4:90
Threat Definition	1:128
Threat Intelligence Platform (TIP)	1:2, 1:4, 1:22, 1:25, 1:50, 1:81, 1:100-102, 1:104-105, 1:109-110, 1:123, 1:125-126,

	1:131-137, 1:141-142, 1:147, 1:149-152, 1:171-172, 1:174-175, 1:190, 4:158, 5:44, 5:54, 5:83, 5:124
Threat Modeling	4:11, 4:50, 4:75, 4:194
Threat Quotient	1:136
ThreatConnect	1:136, 1:138
Thresholds	5:65, 5:69
Ticket Granting Service (TGS)	3:108-109, 3:112-114, 3:117-118, 4:121
tiered structure	1:9, 1:42
Tiny.cc	4:168
Tinyurl	4:168
TLP Taxonomy Library	1:144
TLS1.3	2:162
token stealing	4:150
toolsvoid.com	4:168
Top-Level Domain (TLD)	1:134, 2:60-62, 2:77, 2:79, 2:91, 2:96, 2:154, 3:137, 5:34, 5:59
Total.js	1:166
TRACE	2:119
Traffic Light Protocol (TLP)	1:110, 1:144, 4:127, 4:129-131, 4:142
Transport Layer Security (TLS)	1:95, 2:36, 2:105, 2:132, 2:158-162, 2:170, 2:173, 3:73, 5:150, 5:161, 5:174
true positive	1:37, 1:83, 1:91, 1:93, 1:139, 3:179, 4:158, 5:49-52, 5:61, 5:75, 5:96, 5:102, 5:125, 5:147
TrustedSource	2:78, 2:138-139
TShark	2:34, 2:37
Tuning	1:82, 1:94, 1:97, 4:20, 4:208, 5:2, 5:4-5, 5:8-9, 5:18, 5:26-27, 5:47-48, 5:55, 5:84-86, 5:89, 5:92, 5:96, 5:109, 5:111-114, 5:131, 5:144, 5:146, 5:177, 5:179, 5:195
Two Sides of Monitoring	1:54
TXT Record	2:48, 2:55, 2:73, 2:94-95, 2:178, 2:185
TXT record tunneling	2:94

U

Ubiquiti	5:184
ufw.log	3:78, 3:124
unemployment rate	1:7
Unicode	2:100-101, 2:103-104, 3:155, 3:157-158, 3:161

Unified2	1:73, 1:78
Unintentional Evidence	4:119, 4:123
United Kingdom (UK)	1:177
United States of America (USA)	1:177-178, 5:42, 5:181
unshorten-url	4:168
URL Reputation	2:136, 2:138, 2:141, 3:187, 5:95
URL Sandboxing	2:141, 2:156
Urlquery.net	2:78, 2:137, 2:141, 2:154, 4:134, 4:157
URLQuery.net	2:78, 2:137, 2:141, 2:154, 4:134, 4:157
Urlscan.io	2:137, 2:141, 4:14, 4:128, 4:134, 4:157
URLVoid	2:138-139
US-CERT Incident Categorization	1:120
User and Entity Behavior Analysis (UEBA)	1:179, 3:51, 4:11
User Behavior Analysis (UBA)	1:96, 1:179, 3:30, 3:51
User context	4:63, 5:36
User-Agent	1:62, 1:95, 1:139, 2:113, 2:120-122, 2:126, 2:133, 2:142-145, 2:149-150, 2:155, 2:167, 3:159, 4:134, 4:155, 5:43, 5:55, 5:64, 5:161, 5:167, 5:173
UTF-16	3:153, 3:155-157, 3:161
UTF-32	3:155-156
UTF-8	2:174, 3:153, 3:155-156, 3:161

V

vbs	2:151, 3:153, 3:164, 3:170, 4:162
Vietnam	1:177
Virtual Goods	1:182
Virtual Network Computing (VNC)	1:96, 2:36, 3:9, 3:23, 4:150, 5:100
Virtual Secure Mode (VBS)	3:36, 3:166
Virus detection	1:102, 1:167, 4:196, 5:71, 5:122
VirusTotal	1:62, 1:168-171, 2:51-52, 2:78, 2:82, 2:84-85, 2:92, 2:104, 2:116, 2:138-140, 2:151, 3:178-180, 3:182, 3:186, 4:14, 4:65, 4:128, 4:131-134, 4:136-137, 4:157, 4:166, 5:44, 5:109
VNCPassView	4:150
Vocabulary for Event Recording and Incident Sharing (VERIS)	1:118-119, 1:161, 4:197
Vulnerability information	1:79, 3:32, 5:36

W

WALKOFF	1:166, 5:126
WannaCry	1:180, 2:27, 2:193, 3:168, 4:12, 4:108, 4:149, 5:150, 5:154
Weaponized Linux formats	3:164
Weaponized Windows formats	3:164
Web Application Firewall (WAF)	1:32, 1:68, 1:96, 5:174, 5:176
Web-Exploit	1:167
Webcam Extortion	1:182
Whitelist	1:30, 1:32, 1:65, 2:55, 2:85, 2:102, 2:168, 2:178, 2:181-182, 2:195, 2:199, 2:202, 3:12, 3:15, 3:30, 3:35, 3:37, 3:41-43, 3:66, 3:88, 3:99, 3:116, 3:137, 3:170, 4:24-25, 4:124, 4:146, 5:36, 5:51, 5:65-68, 5:71, 5:100-101, 5:109, 5:149, 5:175-176
Whitelisting	1:30, 1:65, 2:182, 2:199, 2:202, 3:12, 3:30, 3:41-43, 3:88, 3:170, 4:124, 5:36, 5:51, 5:66-68, 5:101, 5:149, 5:175
Windows AppLocker	3:41
Windows Credential Editor (WCE)	4:150
Windows Defender	3:35, 3:66, 3:102, 4:147
Windows Event Forwarding (WEF)	3:125-127
Windows Events	1:73
Windows Exploit Guard	2:28, 3:35
Windows Firewalls with Advanced Security (WFAS)	3:94
Windows Management Instrumentation (WMI)	1:66, 3:23, 3:125, 5:123
Windows Remote Management (WinRM)	2:201
wipe	1:187, 4:7, 4:12, 4:58, 4:138, 4:177, 4:181, 4:184, 5:92
WirelessKeyView	4:150
Wireshark	1:62, 2:34, 2:37-40, 2:53, 2:55, 2:129-132, 2:151, 2:161, 2:170, 2:195-197, 3:73, 3:148, 4:149, 4:170
Working Memory	4:44, 4:47, 4:92, 4:106, 4:110
wsf	3:170

X

X-Forwarded-For	2:120
xls	1:142, 3:171, 4:160, 4:162
XML Event Log (EVTX)	1:73-74, 1:78, 3:55

Y

yEd	1:7, 1:49, 1:63, 1:91, 1:146, 1:158, 2:53, 2:102-103, 2:135, 2:148, 2:166, 2:171, 2:178, 2:180, 2:186-187, 3:32, 3:48, 3:60, 3:98, 3:108, 3:136, 3:159, 3:168, 3:189, 4:112-113, 4:115, 4:164, 4:181, 4:184, 4:189, 5:6, 5:99, 5:123, 5:152, 5:174, 5:195
Your Everyday Threat Intelligence (YETI)	1:136

Z

Zeek	1:68, 1:70, 1:73-74, 1:77, 2:31-32, 2:35-37, 2:41, 2:98, 3:58, 4:22, 4:147, 4:154-155
Zero-Trust	2:23-25
zip	1:87, 3:26, 3:150-153, 3:165, 3:171, 3:173, 4:11, 4:16, 4:51, 4:162-163
Zone transfer	2:59, 2:76, 2:85