SEC555 | SIEM WITH TACTICAL ANALYTICS
GIAC Certified Detection Analyst (GCDA)

555.2

Service Profiling with SIEM

SANS | GIAC CERTIFICATIONS

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

SEC555 | SIEM WITH TACTICAL ANALYTICS

GIAC Certified Detection Analyst (GCDA)

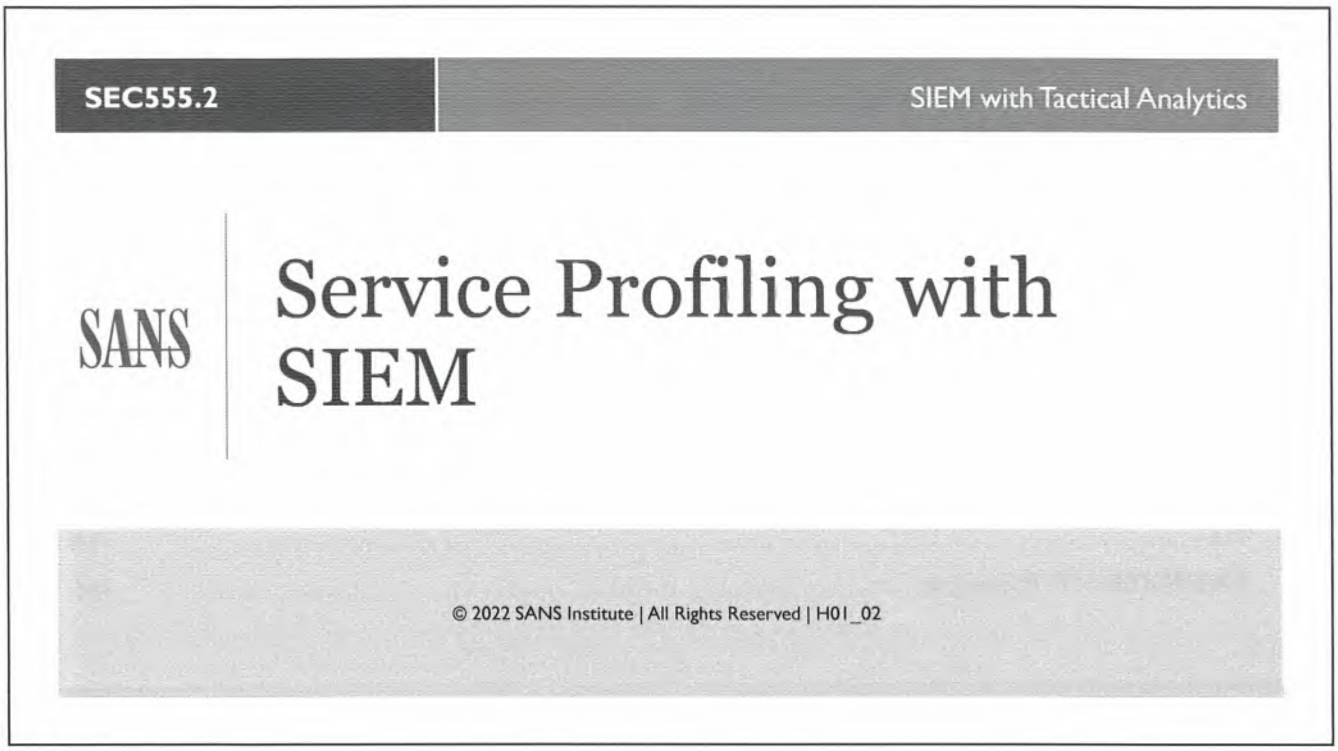
555.2

Service Profiling with SIEM

SANS | GIAC CERTIFICATIONS

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

All reference links are operational in the browser-based delivery of the electronic workbook.



SEC555.2

SIEM with Tactical Analytics

SANS

Service Profiling with SIEM

Welcome to SANS Security 555.2: Service Profiling with SIEM.

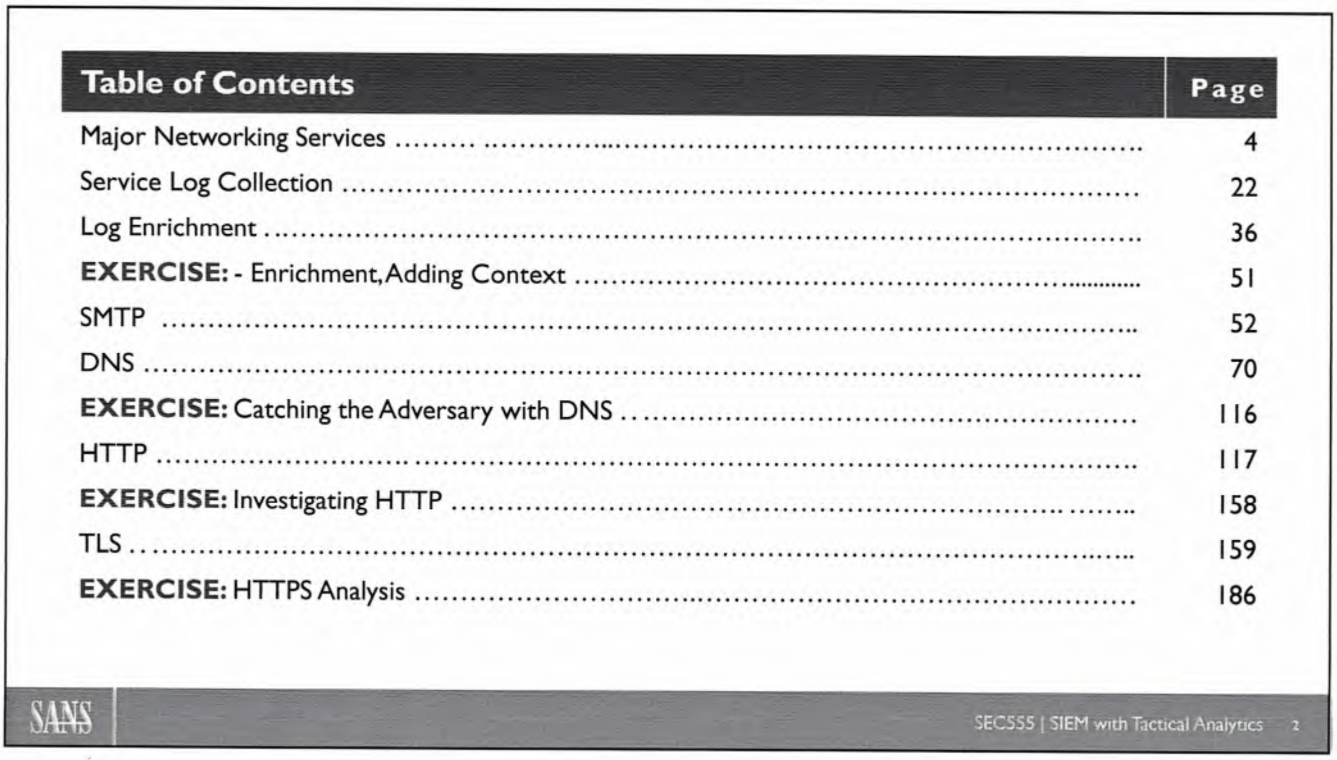


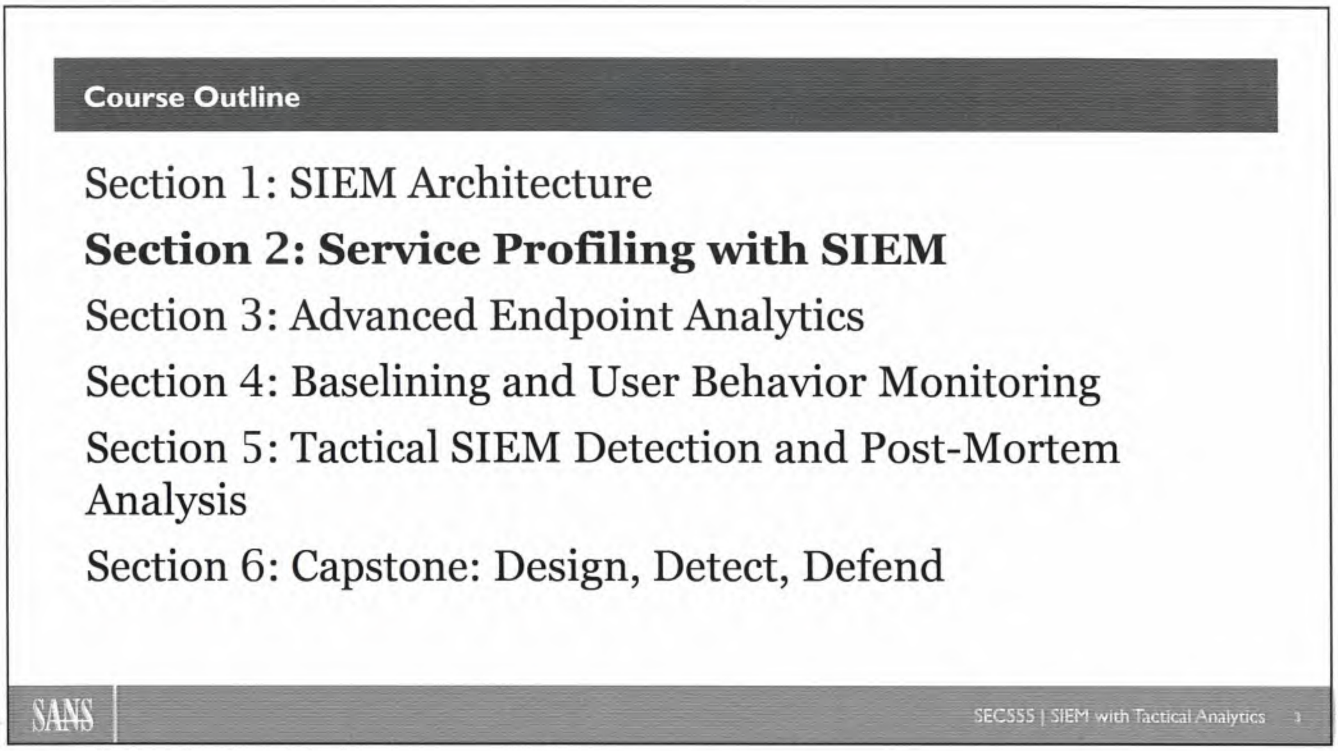
Table of Contents

SANS

SEC555 | SIEM with Tactical Analytics 2

555.2 Table of Contents

This table of contents outlines our plan for 555.2.



Course Outline

Section 1: SIEM Architecture

Section 2: Service Profiling with SIEM

Section 3: Advanced Endpoint Analytics

Section 4: Baselining and User Behavior Monitoring

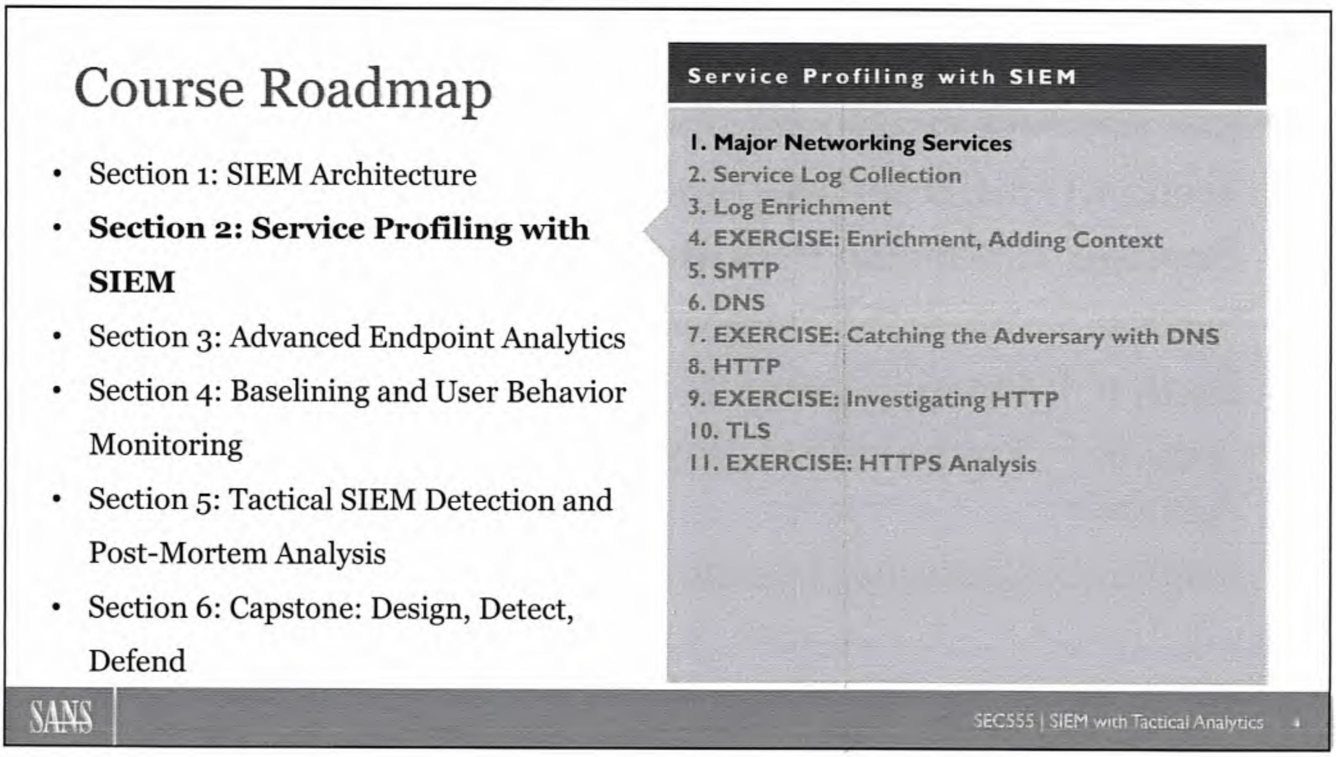
Section 5: Tactical SIEM Detection and Post-Mortem Analysis

Section 6: Capstone: Design, Detect, Defend

SANS

SEC555 | SIEM with Tactical Analytics 3

Welcome to Section 2: Service Profiling with SIEM.



Course Roadmap

- Section 1: SIEM Architecture
- **Section 2: Service Profiling with SIEM**
- Section 3: Advanced Endpoint Analytics
- Section 4: Baselining and User Behavior Monitoring
- Section 5: Tactical SIEM Detection and Post-Mortem Analysis
- Section 6: Capstone: Design, Detect, Defend

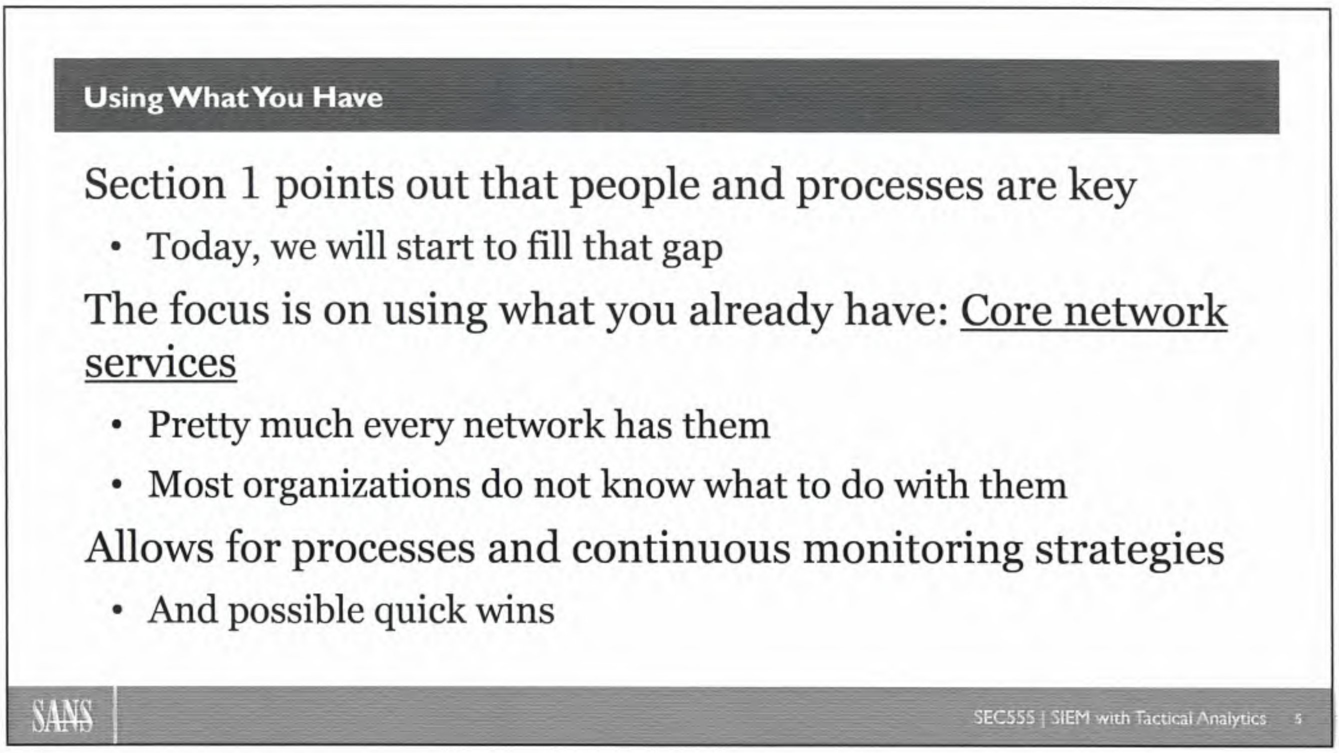
Service Profiling with SIEM

1. **Major Networking Services**
2. Service Log Collection
3. Log Enrichment
4. EXERCISE: Enrichment, Adding Context
5. SMTP
6. DNS
7. EXERCISE: Catching the Adversary with DNS
8. HTTP
9. EXERCISE: Investigating HTTP
10. TLS
11. EXERCISE: HTTPS Analysis

SANS

SEC555 | SIEM with Tactical Analytics 4

This page intentionally left blank.



Using What You Have

Section 1 points out that people and processes are key

- Today, we will start to fill that gap

The focus is on using what you already have: Core network services

- Pretty much every network has them
- Most organizations do not know what to do with them

Allows for processes and continuous monitoring strategies

- And possible quick wins

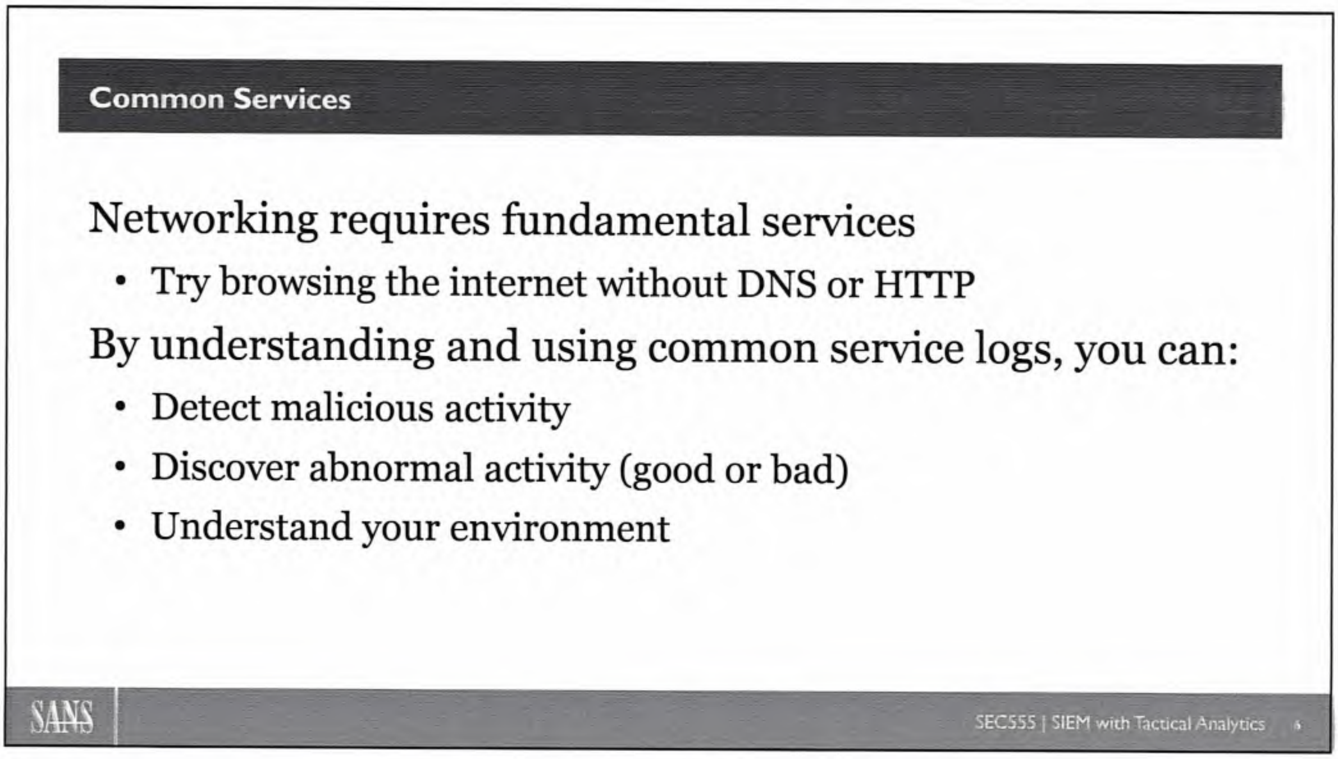
SANS

SEC555 | SIEM with Tactical Analytics 5

Using What You Have

Section 1 covers architecture and getting control over your logs. The rest of the sections focus on the actual how, what, and why. Section 2 will begin by using common service logs. Why? Everyone has key services in their environment and the material in this section will demonstrate how standard service logs can be incredibly useful.

These logs will be taken and transformed into continuous monitoring processes to find and discover evil. Hopefully, by the end of this section, common logs such as DNS and HTTP will no longer be boring. The other reason for focusing on logs such as these is they are easy to collect and allow for quick wins to justify your SIEM investment.



Common Services

Networking requires fundamental services

- Try browsing the internet without DNS or HTTP

By understanding and using common service logs, you can:

- Detect malicious activity
- Discover abnormal activity (good or bad)
- Understand your environment

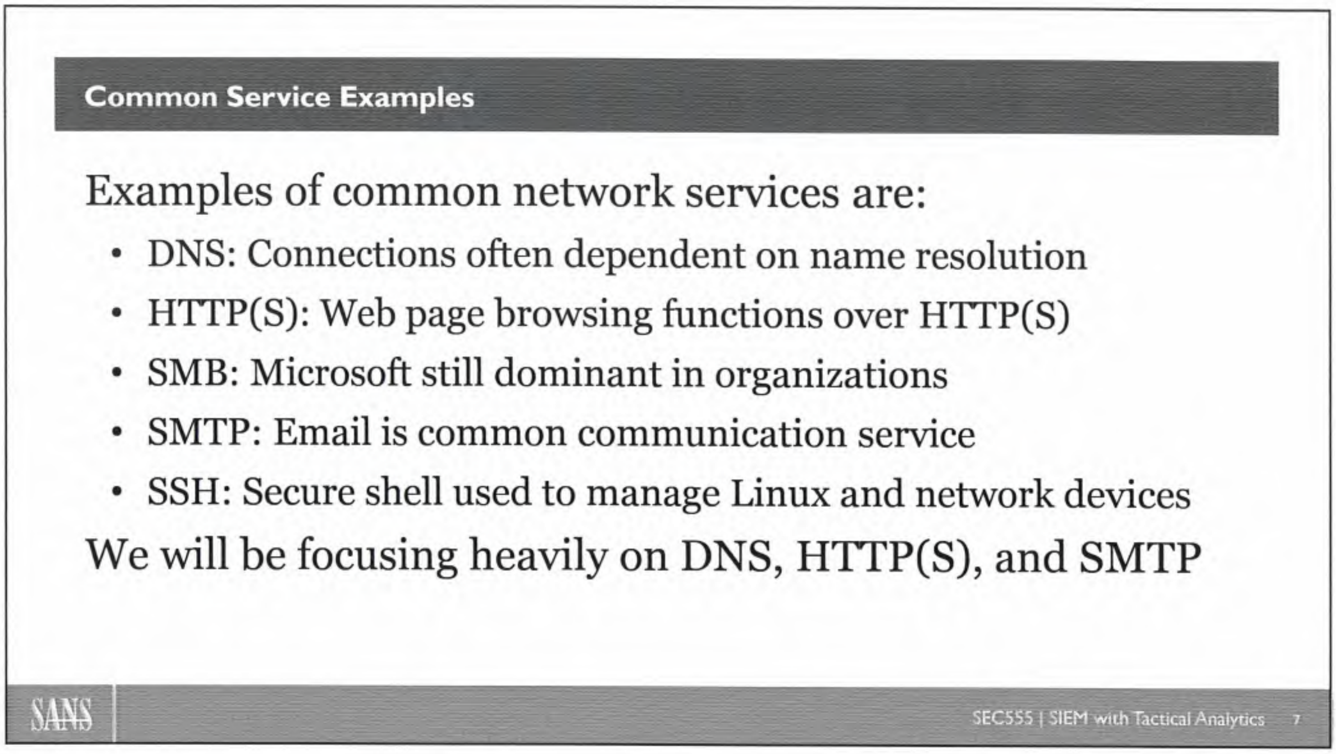
SANS

SEC555 | SIEM with Tactical Analytics 6

Common Services

Every environment requires common services such as DNS and HTTP. Technically, you can have networking without these applications, but what would be the point? Because of the Internet of Things, society has become dependent on quick and easy access to information and collaboration. The Internet of Things, in turn, generates insane amounts of service logs.

While it can be scary to try and collect and handle these types of logs, you will find that there is tremendous value in doing so. The dependency on core networking services means that our adversaries must use them as well, providing the opportunity for detection.



Common Service Examples

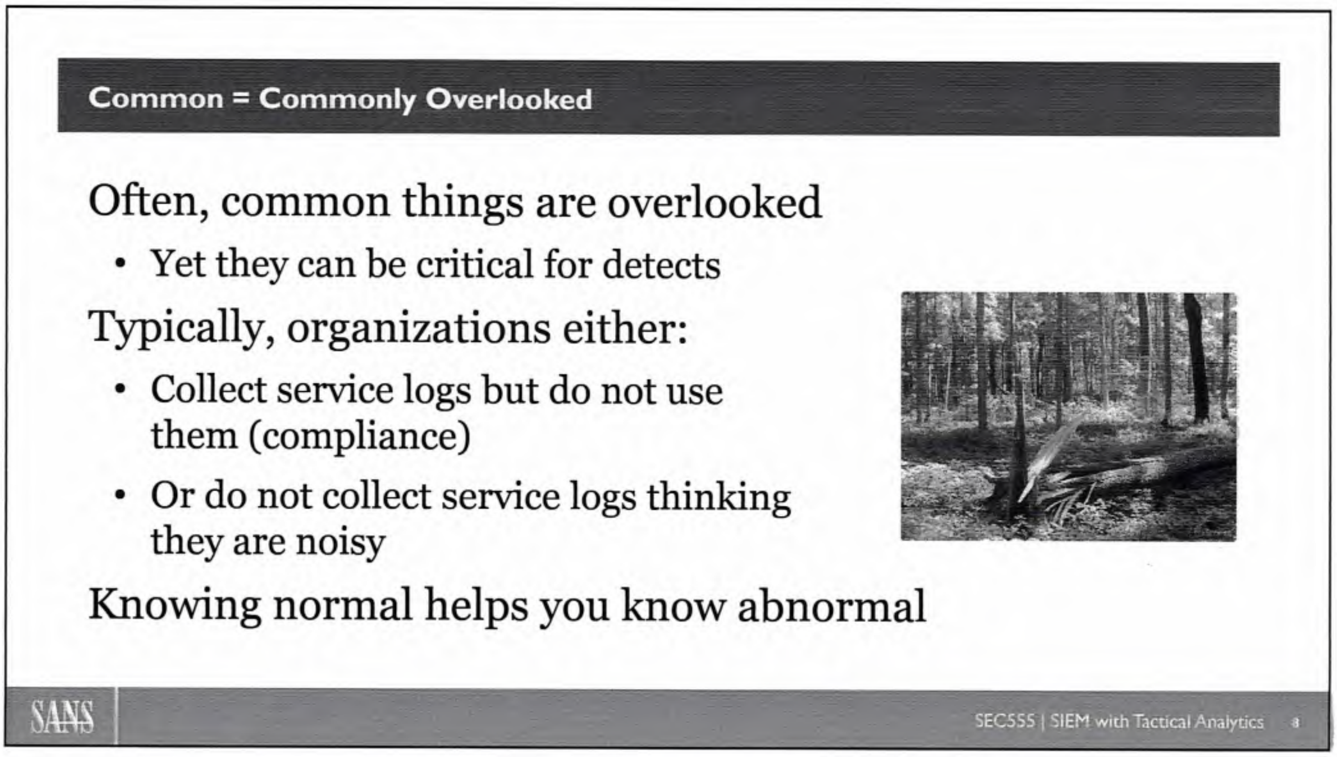
Examples of common network services are:

- DNS: Connections often dependent on name resolution
- HTTP(S): Web page browsing functions over HTTP(S)
- SMB: Microsoft still dominant in organizations
- SMTP: Email is common communication service
- SSH: Secure shell used to manage Linux and network devices

We will be focusing heavily on DNS, HTTP(S), and SMTP

Common Service Examples

While many services exist, the most common are DNS, HTTP, HTTPS, and SMTP. These are used by end users to surf the internet and receive email. As a result, we will be focusing heavily on these services since they are common attack vectors.



Common = Commonly Overlooked

Often, common things are overlooked

- Yet they can be critical for detects

Typically, organizations either:

- Collect service logs but do not use them (compliance)
- Or do not collect service logs thinking they are noisy

Knowing normal helps you know abnormal

SANS

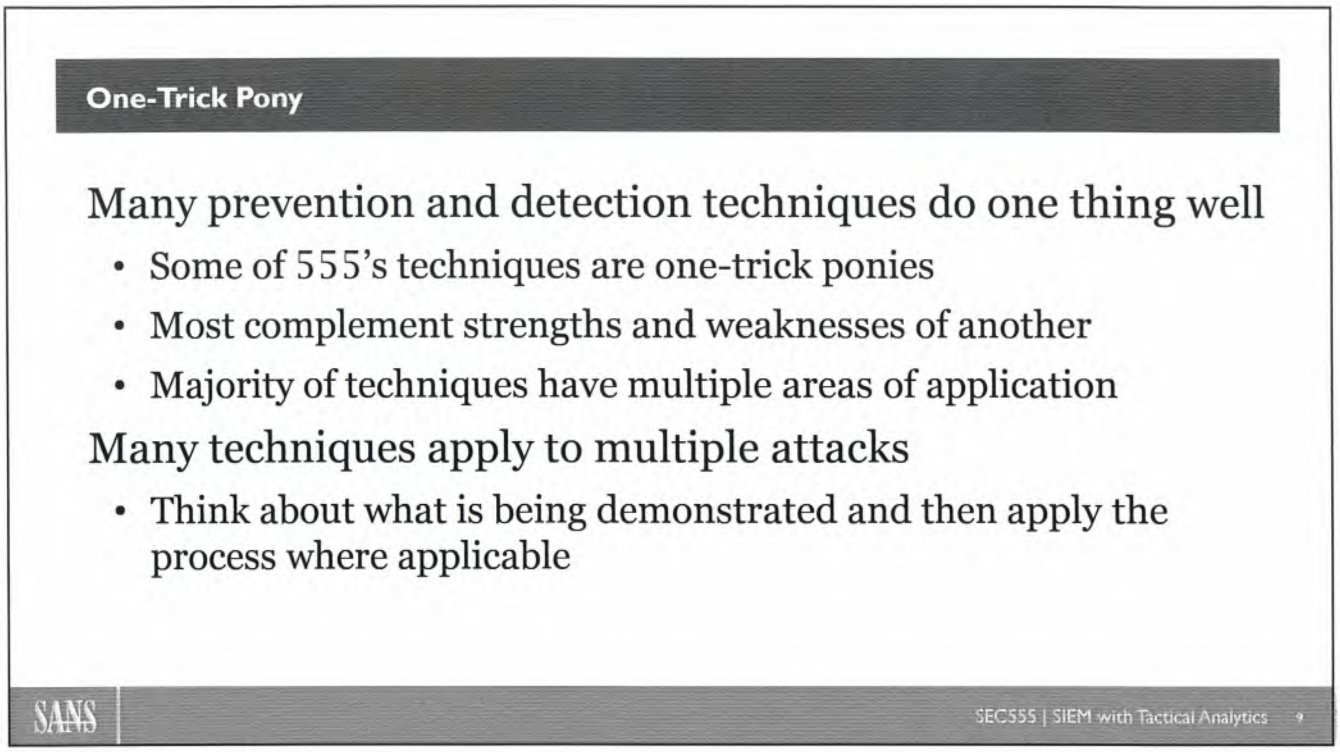
SEC555 | SIEM with Tactical Analytics 8

Common = Commonly Overlooked

Do not be like the nobles who wrinkle their nose at a passing commoner. Common logs compose the foundation of the interworking of your environment and can be critical for finding things. Sadly, most organizations either choose not to collect these types of logs or collect them but ignore them. The thought is that they serve no purpose so do not collect them, or that they have to be collected for compliance but, unless compromised, we do not need to look at them.

Have you ever seen a movie where characters are traveling through the woods, and someone stops and states, "Do you hear anything?" Everyone says no, and then that person says, "Me either. In fact, it's too quiet. Something is wrong." The same thing applies to common service logs. They typically surround us and have a common flow. The skill is tapping into this flow and finding out how to tell when something is wrong.

Another analogy would be a movie where someone has been chased into a field. The pursuer shoots a gun off, causing birds to fly into the air and then watches to see where they do not land. That spot is where the person is hiding. Today, you will be learning techniques to apply a similar process to find anomalies.



One-Trick Pony

Many prevention and detection techniques do one thing well

- Some of 555's techniques are one-trick ponies
- Most complement strengths and weaknesses of another
- Majority of techniques have multiple areas of application

Many techniques apply to multiple attacks

- Think about what is being demonstrated and then apply the process where applicable

SANS
SEC555 | SIEM with Tactical Analytics 9

One-Trick Pony

This course is not a one-trick pony show. Many techniques will be described in detail using a given scenario. However, the techniques shown will most likely apply to multiple scenarios and processes. The challenge for yourself is to study how the technique is being used and to see if 1) the technique applies in your environment and 2) can this technique be applied repetitively in your environment and, if so, where?

After all, this course is designed to fix the people and process issue. Therefore, the underlying theme is to broaden your depth of knowledge and identify areas where techniques can complement each other.

Lab Me, Inc.

Meet your victim company: Lab Me, Inc.

- Lab Me, Inc. is a fast-growing healthcare organization providing service to millions of patients

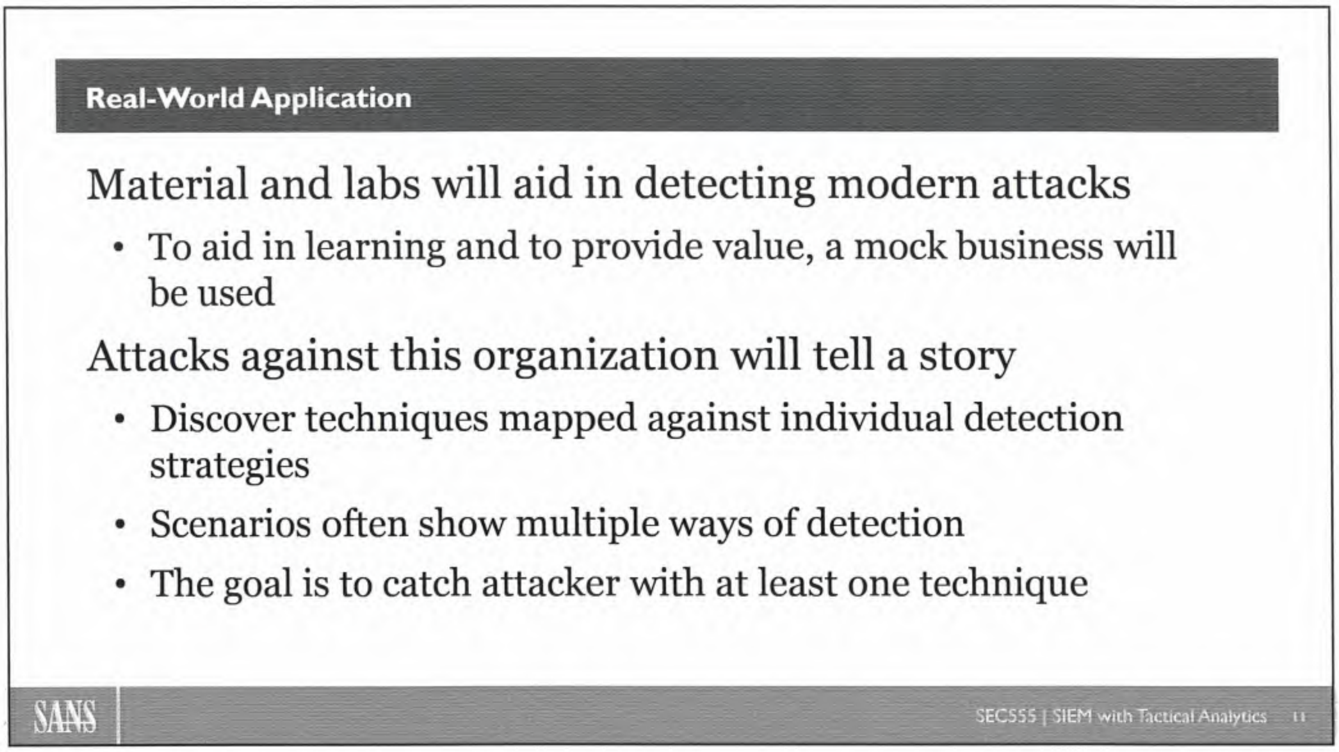
Site: www.labmeinc.com

Previous security professional left the company and you have been hired as a replacement

Lab Me, Inc.

Lab Me, Inc. welcomes you as their new information security super defender. Everyone is excited to see what you can do. Lab Me, Inc. has had tremendous growth. Due to the fast growth, as well as the previous information security position now being vacant, there is concern that security may not be sufficient to meet the needs of Lab Me, Inc.

Granted, the previous security administrator did a great job setting up defenses like firewalls, endpoint protection, and other common security controls. As far as you have been told, they are in full compliance with HIPAA and PCI. The last project this individual was working on was the implementation of a SIEM. Unfortunately, this now falls to you, and upper management would like to see this project finished before moving on to other projects.



Real-World Application

Material and labs will aid in detecting modern attacks

- To aid in learning and to provide value, a mock business will be used

Attacks against this organization will tell a story

- Discover techniques mapped against individual detection strategies
- Scenarios often show multiple ways of detection
- The goal is to catch attacker with at least one technique

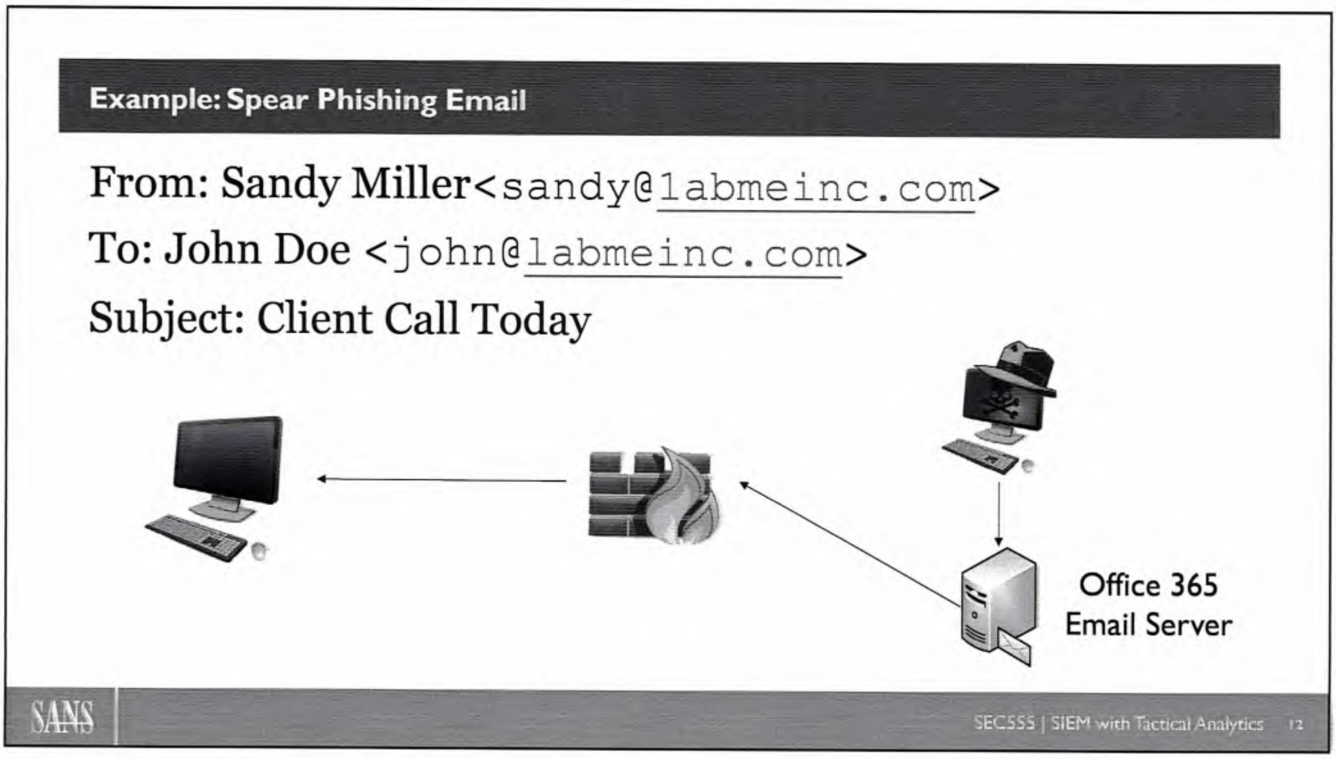
SANS

SEC555 | SIEM with Tactical Analytics 11

Real-World Application

To make learning fun, a mock business has been created. This mock business will be used to visualize and add understanding to techniques. This allows for mapping modern-day detection methods against "real-world" situations.

As this course progresses, this mock business will be used to explore "real-world" scenarios that will be broken up into smaller pieces, and techniques will be taught in regard to these pieces. This allows learning in small bites that is easy to understand. Know that even though this is a mock business, many of the attacks and detection scenarios given are based on real-world experiences of the author and other SANS instructors.



Example: Spear Phishing Email

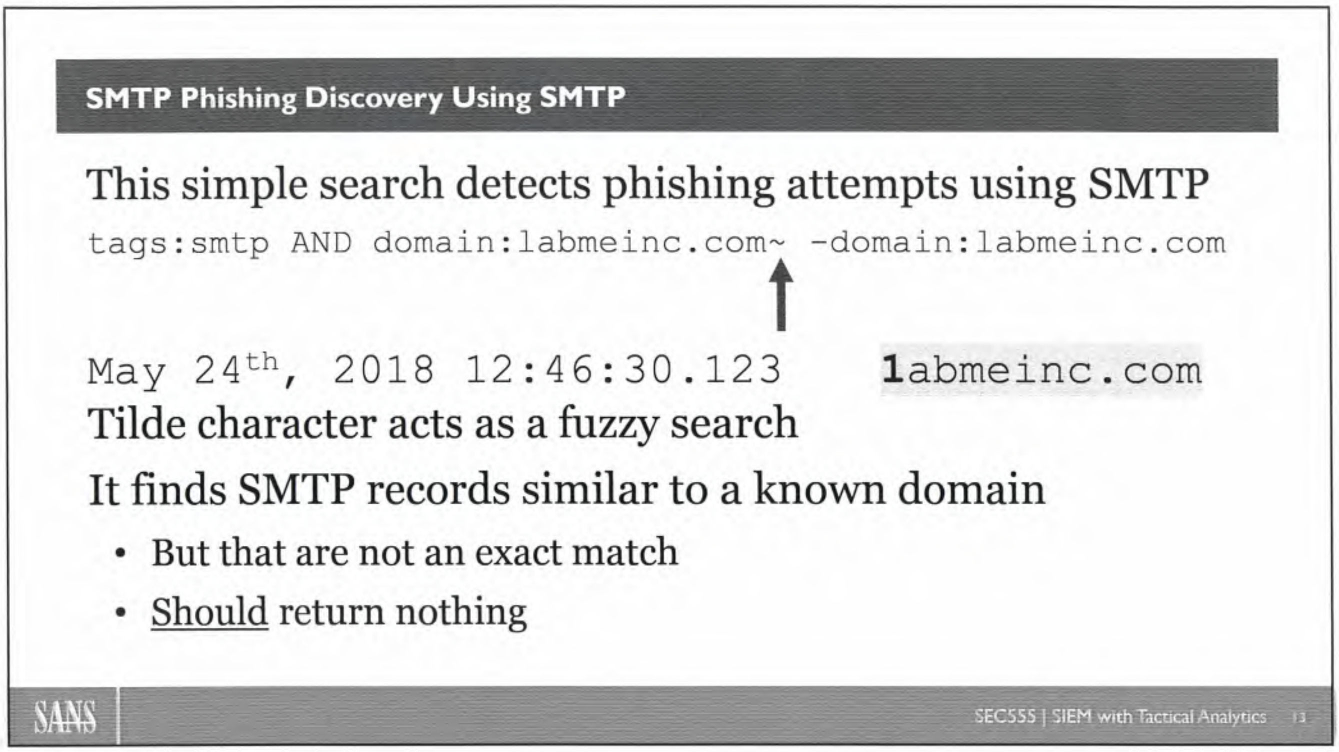
From: Sandy Miller<sandy@1abmeinc.com>

To: John Doe <john@labmeinc.com>

Subject: Client Call Today

Example: Spear Phishing Email

Take, for example, a simple spear phishing email. In this example, an adversary did their homework and discovered that Sandy Miller works with John Doe. Since John is the target, the attacker sent an email from sandy@1abmeinc.com to look similar to her real email address, which is sandy@labmeinc.com. Because there is only a single character difference, it is possible John will not catch that the email is not really from Sandy.



SMTP Phishing Discovery Using SMTP

This simple search detects phishing attempts using SMTP

```
tags:smtp AND domain:labmeinc.com~ -domain:labmeinc.com
```

```
May 24th, 2018 12:46:30.123    1abmeinc.com
```

Tilde character acts as a fuzzy search

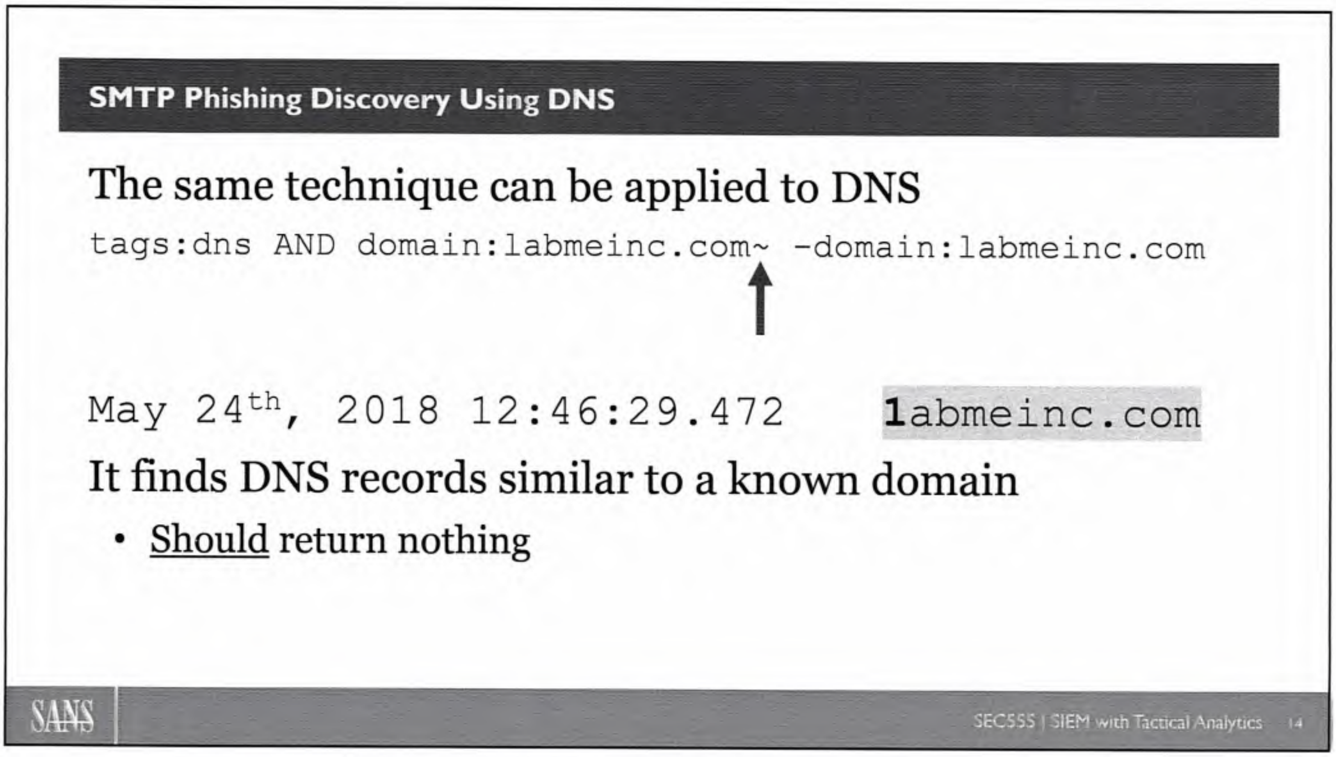
It finds SMTP records similar to a known domain

- But that are not an exact match
- Should return nothing

SMTP Phishing Discovery Using SMTP

Assuming that SMTP logs are being collected, discovering phishing attempts that look similar to an organization's real email domain is simple. In this example, the email domain in the SMTP log is fuzzy-searched for anything similar to labmeinc.com that is not an exact match to labmeinc.com. As a result, 1abmeinc.com would be found.

Levenshtein distance is a relatively simple discovery technique using fuzzy searches. A fuzzy search looks for things similar to whatever you are searching for by searching for values that are similar to a search string but off by a specified number of characters. Fuzzy searching works best with longer search strings. If you had a small domain such as sec555.com, it might still work, but it is possible it may create false positives.



SMTP Phishing Discovery Using DNS

The same technique can be applied to DNS

```
tags:dns AND domain:labmeinc.com~ -domain:labmeinc.com
```

```
May 24th, 2018 12:46:29.472     1abmeinc.com
```

It finds DNS records similar to a known domain

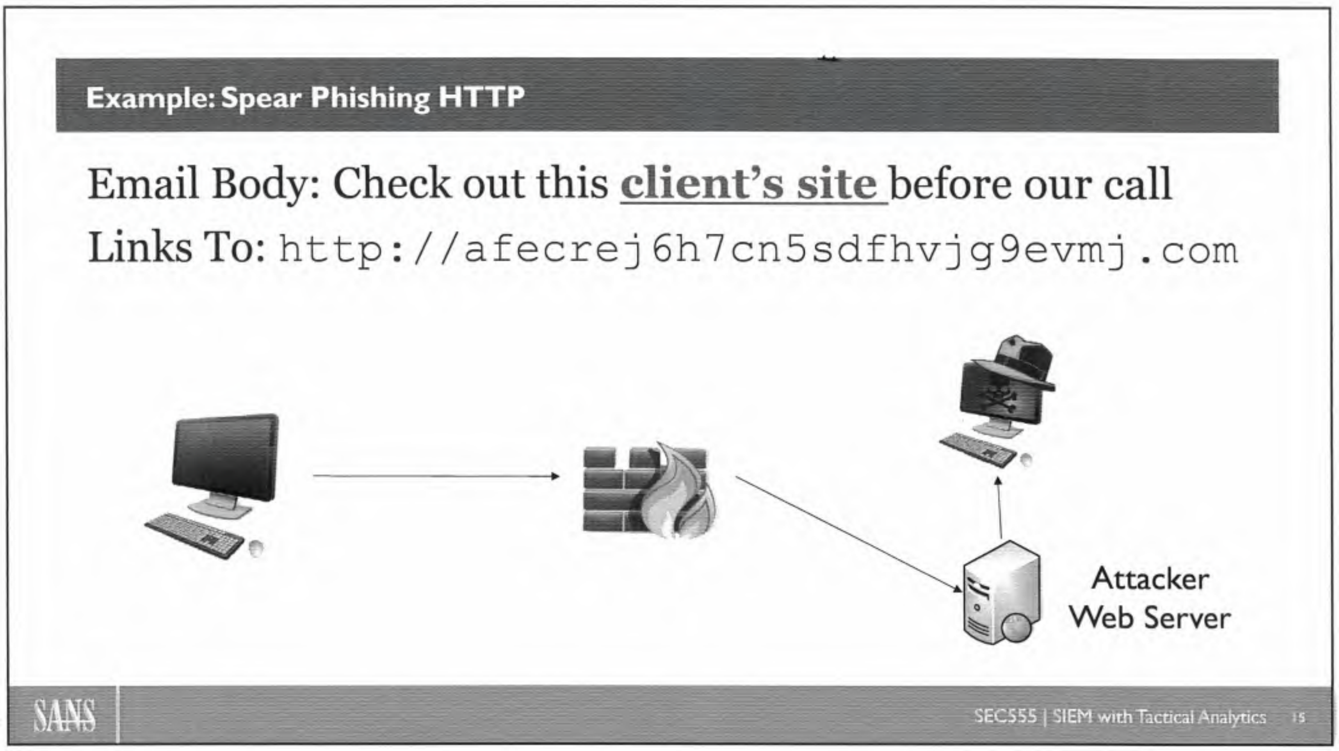
- Should return nothing

SMTP Phishing Discovery Using DNS

To demonstrate that a given technique can apply to multiple logs or scenarios, take the same inbound phishing email. Instead of analyzing the SMTP log this time, the example uses DNS logs. For example, fuzzy searching would discover the use of hasecuritysolution.com due to its similarity to hasecuritysolutions.com.

However, does the technique apply as successfully to DNS as it does to SMTP? Think about this for a second. What would happen if this email domain matched the corporate website, and someone tried to access the site but had a typo in their browser? A typo would end up creating a false positive. What would happen if an external user tried to send you an email but had a typo? The email would never arrive in the first place, so no false positive would be created.

Applying this technique for detecting phishing attempts could be used against DNS, HTTP, and SMTP but would likely create a lot of false positives for everything except SMTP. Again, think, understand, and apply.



Example: Spear Phishing HTTP

Email Body: Check out this **client's site** before our call

Links To: `http://afecrej6h7cn5sdfhvjg9evmj.com`

Attacker
Web Server

SANS

SEC555 | SIEM with Tactical Analytics 15

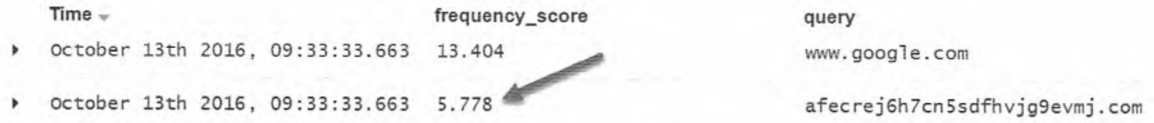
Example: Spear Phishing HTTP

Note: Link above and below is a fictious link and is not a real URL, it is solely for demonstration purposes.

Continuing the spear phishing example, let us assume that John opened the email and inside was a link. Upon clicking this link, his browser goes to http://afecrej6h7cn5sdfhvjg9evmj.com. The domain would show up in DNS logs, and it would show up in HTTP logs. But how is this helpful?

HTTP Phishing Discovery Using DNS

http://afecrej6h7cn5sdfhvjg9evmj.com seems random...



```
     Time ▾                             frequency_score                query
▸    October 13th 2016, 09:33:33.663    13.404                         www.google.com
▸    October 13th 2016, 09:33:33.663    5.778                          afecrej6h7cn5sdfhvjg9evmj.com
```

But frequency_score is not a field in my logs...

- Augmenting standard logs adds incredible power

Again, the same technique could be applied against HTTP, DNS, SMTP, and other event sources

SANS

SEC555 | SIEM with Tactical Analytics 16

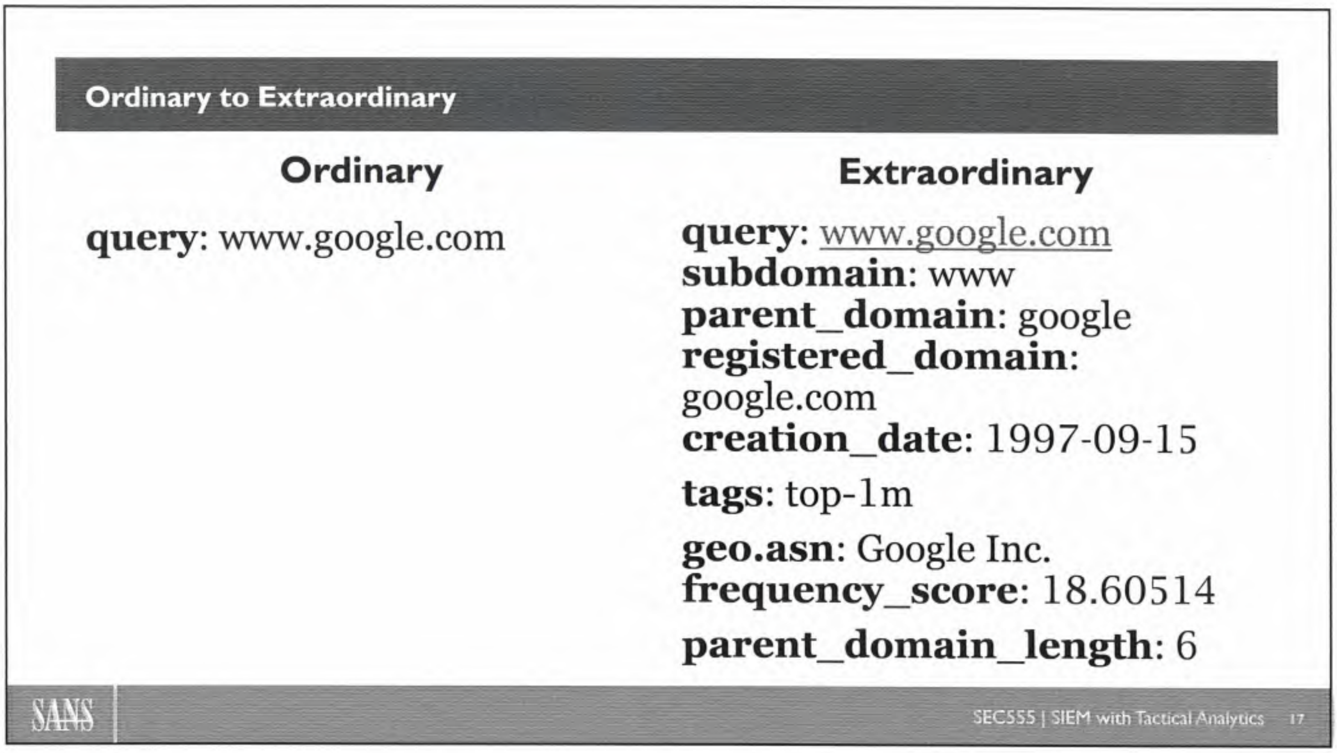
HTTP Phishing Discovery Using DNS

Because of security devices, attackers often have to use randomness to evade filters. In this example, the domain afecrej6h7cn5sdfhvjg9evmj.com is clearly random. This screenshot shows a DNS log for afecrej6h7cn5sdfhvjg9evmj.com with a frequency_score of 5.778. A normal DNS record for www.google.com has a score of 13.404. This massive gap in score points to afecrej6h7cn5sdfhvjg9evmj.com as being random.

Even though this example uses DNS, the same technique could be applied against HTTP or HTTPS. The example is because the randomness check is being applied to the website name. Therefore, the website name exists within a DNS log as well as within HTTP or event HTTPS if the site was redirected.

Now, I know you are thinking, "Where did the frequency_score field come from?" DNS logs do not generate this field. In this example, the DNS logs have been augmented by performing a randomness check against the DNS query and saving it into a field called frequency_score. This will be covered shortly. The point is a common service log can be augmented and made defensible regarding active detection.

The difference between a good meal and a great meal can be the addition of a single ingredient. This author hopes to share a few spices and seasonings that will add some flavor to your logs.



Ordinary to Extraordinary

Ordinary

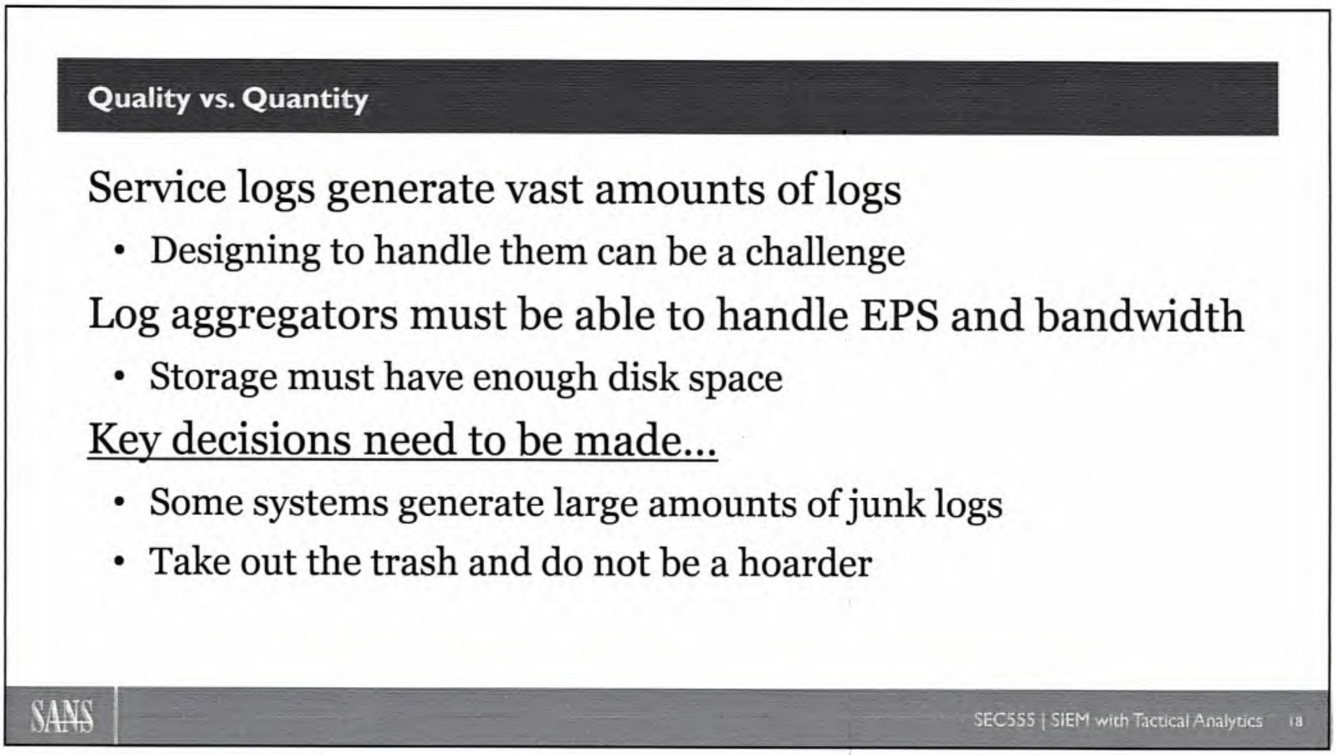
query: www.google.com

Extraordinary

query: www.google.com
subdomain: www
parent_domain: google
registered_domain: google.com
creation_date: 1997-09-15
tags: top-1m
geo.asn: Google Inc.
frequency_score: 18.60514
parent_domain_length: 6

Ordinary to Extraordinary

Your logs are here for log boot camp. The goal is to take them from ordinary to extraordinary, similar to how Steve Rogers started as a weak individual who later underwent an experimental procedure and came out as Captain America. Sometimes, it is OK to experiment with your logs because extraordinary results can happen.



Quality vs. Quantity

Service logs generate vast amounts of logs

- Designing to handle them can be a challenge

Log aggregators must be able to handle EPS and bandwidth

- Storage must have enough disk space

Key decisions need to be made...

- Some systems generate large amounts of junk logs
- Take out the trash and do not be a hoarder

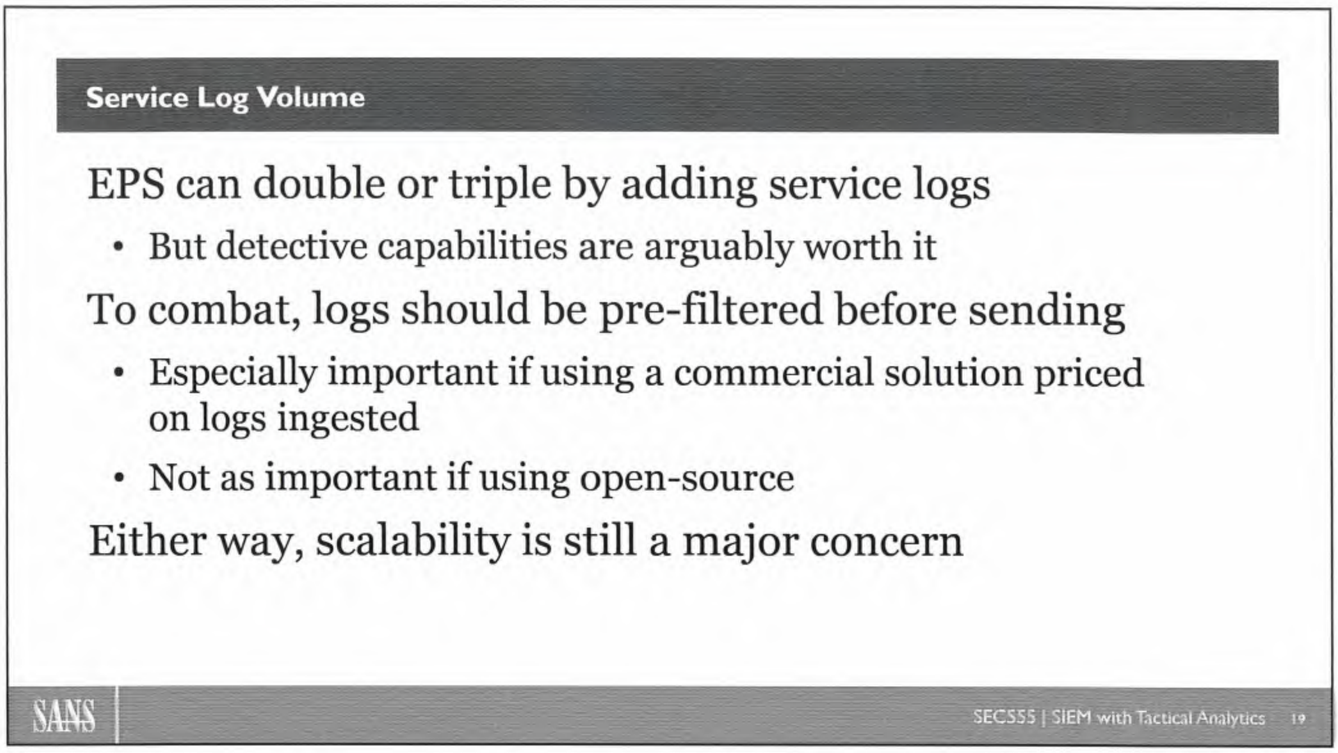
Quality vs. Quantity

Service logs are valuable, as we have already seen with the DNS and HTTP examples in the previous slides. They are actionable in detection. However, because key services like DNS and HTTP are used constantly, the number of logs generated is vast. Therefore, it is possible that some key decisions will need to be made.

For example, often 90%+ of DNS logs can be trimmed by filtering out requests to internal domains. Also, some systems generate large amounts of requests that are not actionable. This could be a system using DNS for security checks such as Team Cymru,[1] or it could be a web server constantly talking over an HTTP or HTTPS API to poll a cloud system. Trim the fat. If it is not actionable, then drop it like it is hot.

Reference:

https://team-cymru.com/community-services/mhr/, https://sec555.com/4g



Service Log Volume

EPS can double or triple by adding service logs

- But detective capabilities are arguably worth it

To combat, logs should be pre-filtered before sending

- Especially important if using a commercial solution priced on logs ingested
- Not as important if using open-source

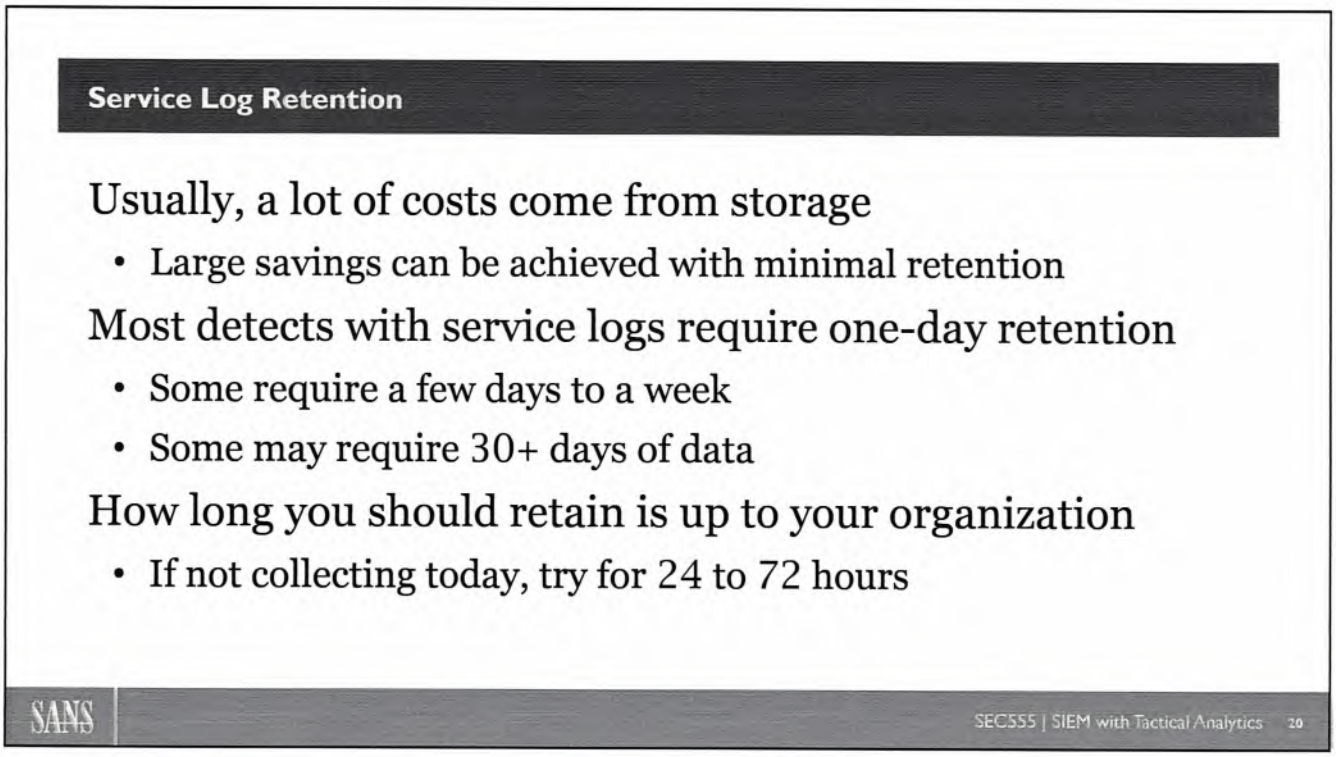
Either way, scalability is still a major concern

SANS | SEC555 | SIEM with Tactical Analytics 19

Service Log Volume

This author has experienced multiple times where environments that are not collecting service logs begin collecting service logs and quickly get overwhelmed. Too many logs are because your network communication relies on key application services. As such, service logs are frequently generated. Service logs can cause your EPS to double or triple if an organization previously only had basic logging on.

If an aggregator cannot handle the volume of logs thrown at it, consider pre-filtering service logs before they are shipped off. Pre-filtering can sometimes decrease the number of logs by as much as 80 to 90 percent. This decrease is extremely helpful if you are using a commercial solution that is priced based on logs ingested.



Service Log Retention

Usually, a lot of costs come from storage

- Large savings can be achieved with minimal retention

Most detects with service logs require one-day retention

- Some require a few days to a week
- Some may require 30+ days of data

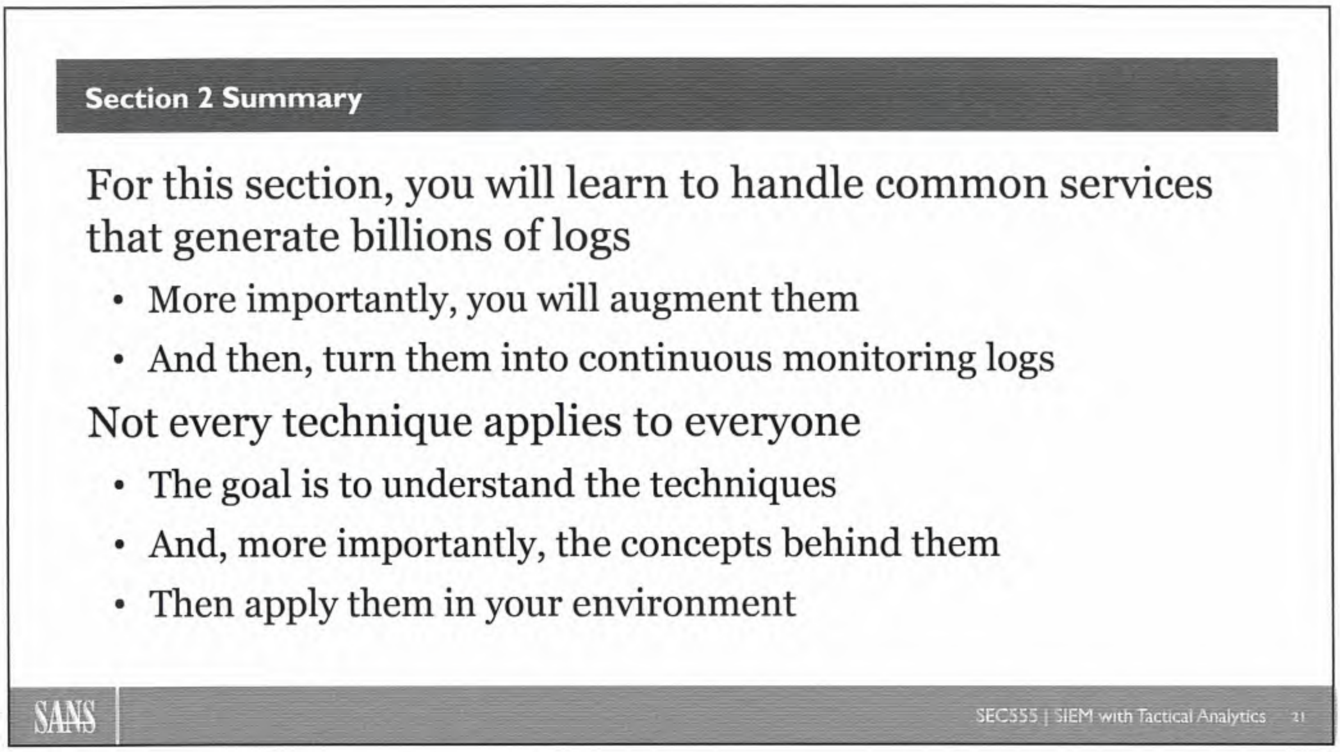
How long you should retain is up to your organization

- If not collecting today, try for 24 to 72 hours

Service Log Retention

Due to hardware and licensing costs, retention of service logs is a big deal. Put plainly, there are a lot of service logs, and as a result, they take up a significant amount of space. The good news is that a majority of the detects used against service logs only require a day's worth of logs. There are a few things that work best for at least a week's worth of data. One thing to note if you are looking into machine learning is that it is more accurate with more data.

If you currently are not collecting service logs, it can be intimidating to start. It also can be hard to get approval. If either of these is the case, you might consider doing a pilot using only one day's worth of service logs. Typically, this simple proof of concept drives home the effectiveness of service logs. Ideally, aim for three days. That way, analysts have time to review—even after a weekend or holidays.



Section 2 Summary

For this section, you will learn to handle common services that generate billions of logs

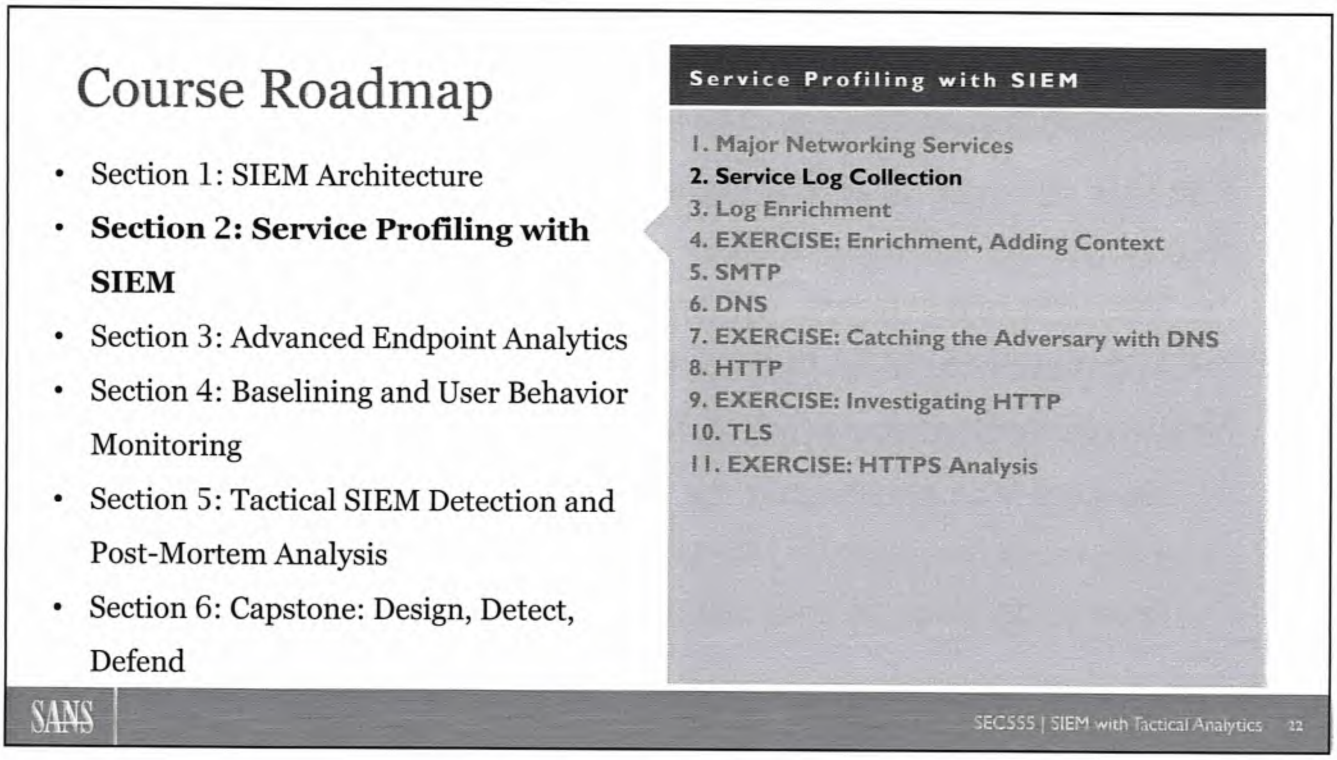
- More importantly, you will augment them
- And then, turn them into continuous monitoring logs

Not every technique applies to everyone

- The goal is to understand the techniques
- And, more importantly, the concepts behind them
- Then apply them in your environment

Section 2 Summary

In this section, we will take normal application service logs and transform them into tools for continuous monitoring and tactical defense. Your added mission is to take these processes and techniques and stretch them or even expand upon them. By studying the scenarios provided, you should be able to develop techniques that are tailored to your specific environment.



Course Roadmap

- Section 1: SIEM Architecture
- **Section 2: Service Profiling with SIEM**
- Section 3: Advanced Endpoint Analytics
- Section 4: Baselining and User Behavior Monitoring
- Section 5: Tactical SIEM Detection and Post-Mortem Analysis
- Section 6: Capstone: Design, Detect, Defend

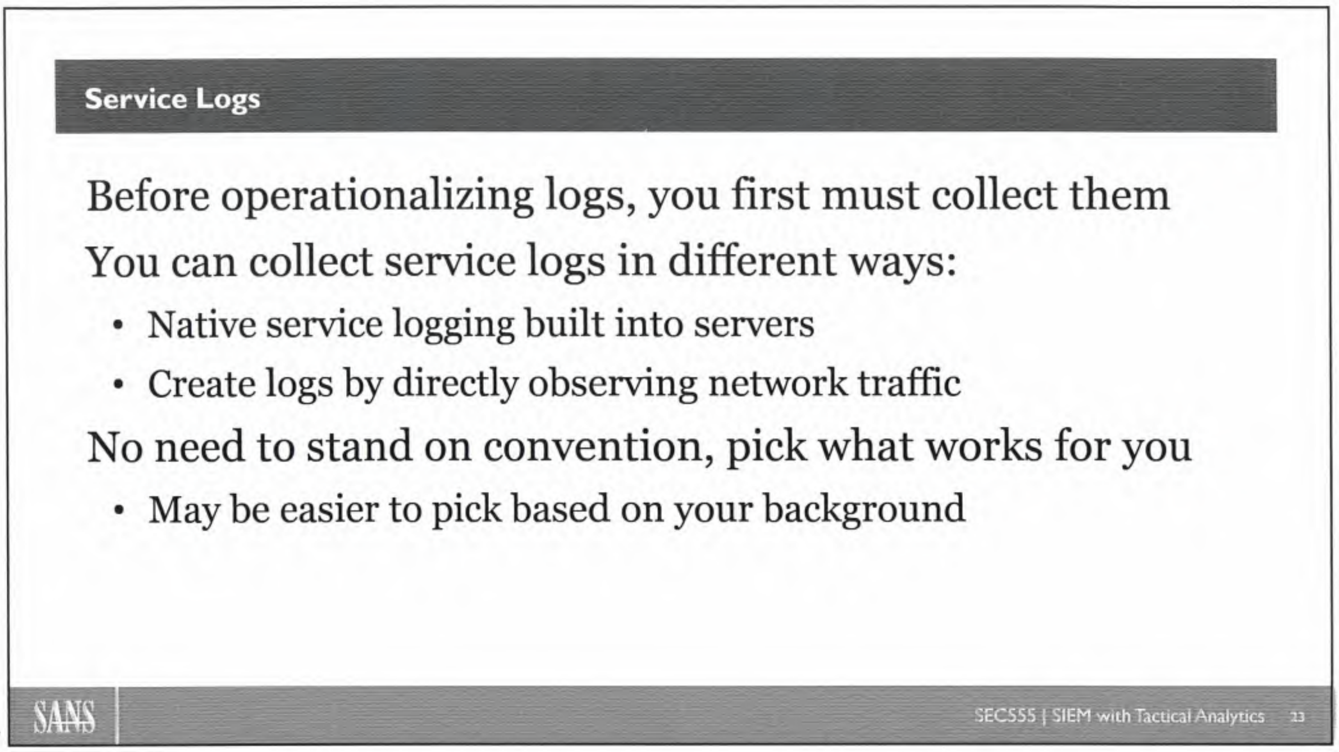
Service Profiling with SIEM

1. Major Networking Services
2. **Service Log Collection**
3. Log Enrichment
4. EXERCISE: Enrichment, Adding Context
5. SMTP
6. DNS
7. EXERCISE: Catching the Adversary with DNS
8. HTTP
9. EXERCISE: Investigating HTTP
10. TLS
11. EXERCISE: HTTPS Analysis

SANS

SEC555 | SIEM with Tactical Analytics 22

This page intentionally left blank.



Service Logs

Before operationalizing logs, you first must collect them

You can collect service logs in different ways:

- Native service logging built into servers
- Create logs by directly observing network traffic

No need to stand on convention, pick what works for you

- May be easier to pick based on your background

Service Logs

Before jumping straight into the awesome things, you can do with service logs, it is important to take a step back and discuss strategies for collecting them. The step back is important as there are some non-traditional methods that may be beneficial to your situation. For example, what if you are not allowed to reconfigure your DNS or proxy servers, but you need those logs? In this case, you can get the logs by tapping your network.

Regardless of the method, the goal is to point you in a couple of directions so that you can choose which path you think is best.

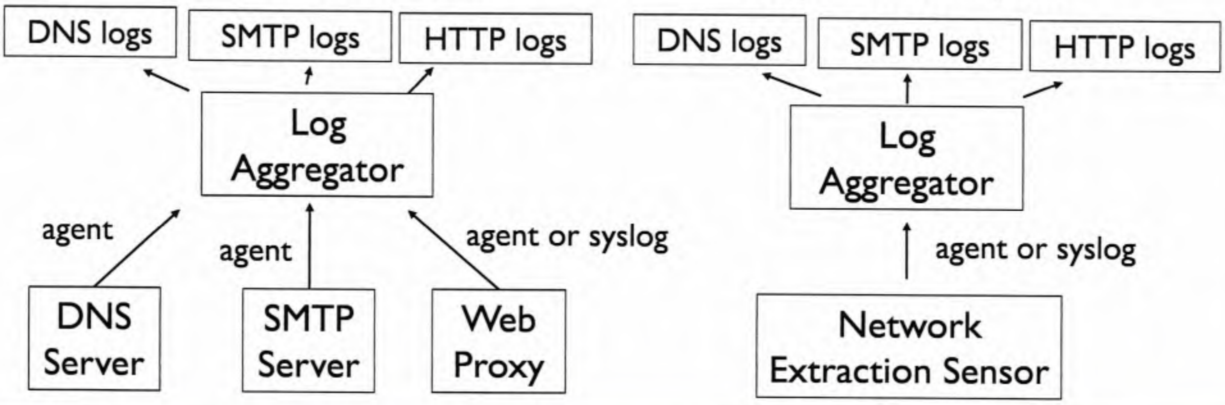
Traditional vs. Network Extraction

Traditional

Multiple collection points

Network Extraction

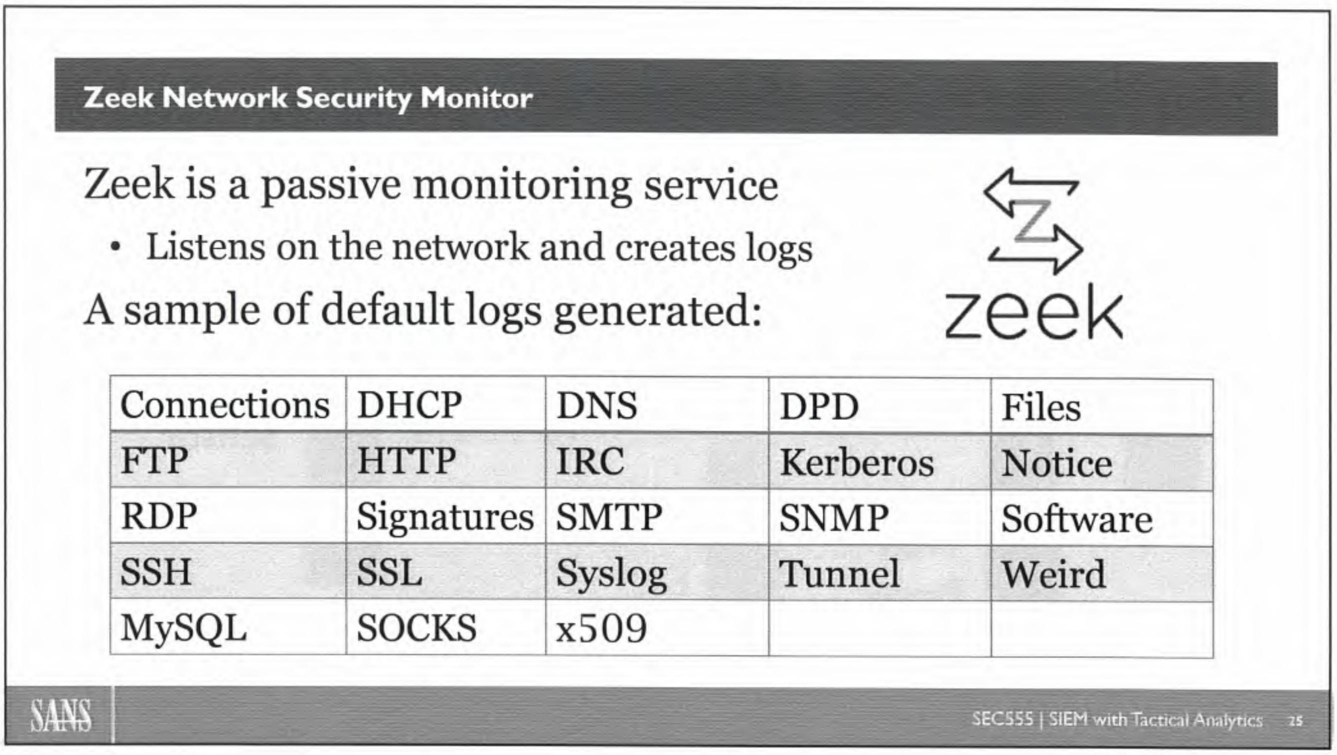
Single collection point



Traditional vs. Network Extraction

The normal method for collecting service logs is to install agents on all of your servers or enable syslog if you are using an appliance. Log agents make logical sense as you are going straight to the source of the service. For example, you might install a log agent on a DNS server and then point it to the file location and have it ship off logs as they are generated.

The flip side of this is that you can use a network monitoring system to look at packets and create logs as they are seen on the network. This approach is interesting: Instead of having multiple agents on various application servers, a single network monitoring host can often see and generate all the logs from one spot. The main drawback is that you first must give this system network visibility, such as through the use of a tap or port mirror.



Zeek Network Security Monitor

Zeek is a passive monitoring service

- Listens on the network and creates logs

A sample of default logs generated:

Connections	DHCP	DNS	DPD	Files
FTP	HTTP	IRC	Kerberos	Notice
RDP	Signatures	SMTP	SNMP	Software
SSH	SSL	Syslog	Tunnel	Weird
MySQL	SOCKS	x509		

SANS SEC555 | SIEM with Tactical Analytics 25

Zeek Network Security Monitor

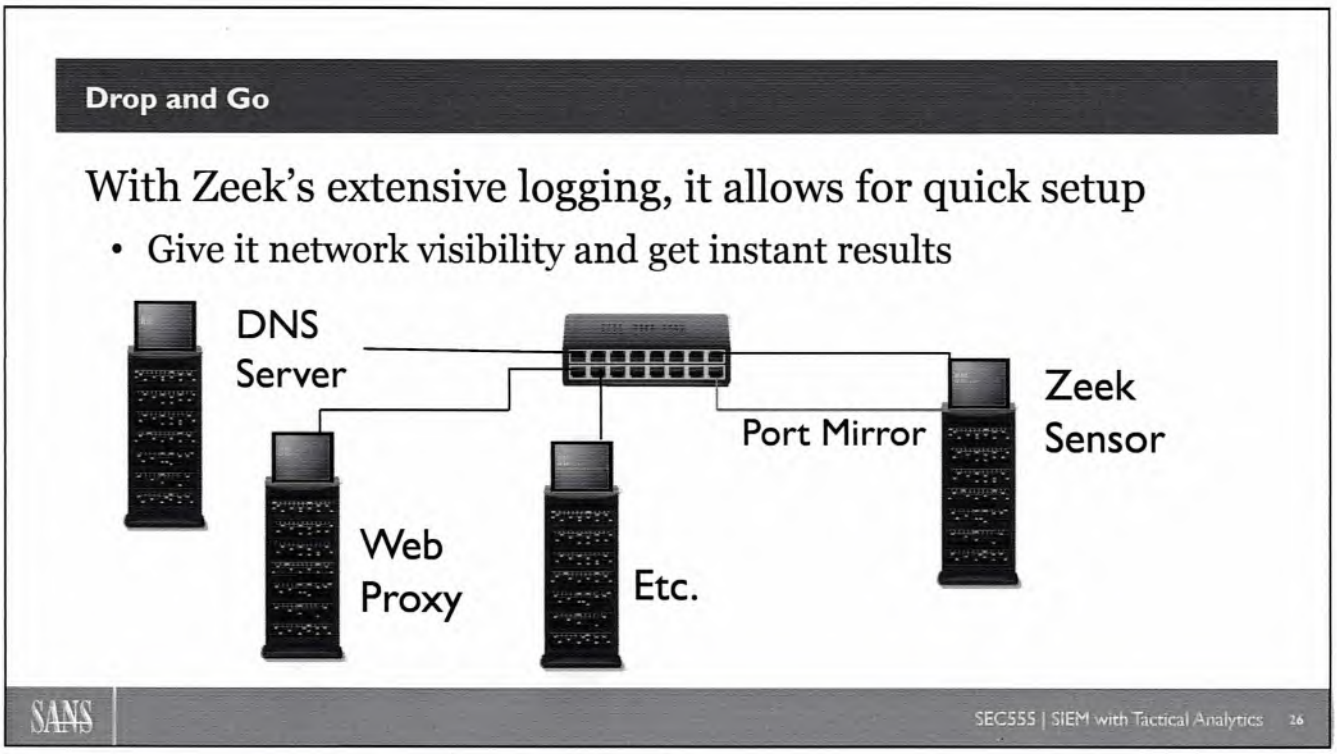
One such network extractor is Zeek Network Security Monitor. Zeek is an open-source project developed and maintained by the International Computer Science Institute in Berkeley, CA, and the National Center for Supercomputing Applications in Urbana-Champaign, IL. Its purpose is to listen on the network and extract metadata into logs. As you can see in the table above, Zeek default settings allow for the creation of pretty much all major service logs.

Zeek is currently being adopted by major universities, US military branches, research labs, and Fortune 500 companies. For a commercial version of Zeek, look into Corelight.

Referecnce:

https://www.zeek.org, https://sec555.com/4h

https://www.corelight.com, https://sec555.com/4i



Drop and Go

With Zeek's extensive logging, it allows for quick setup

- Give it network visibility and get instant results

SANS

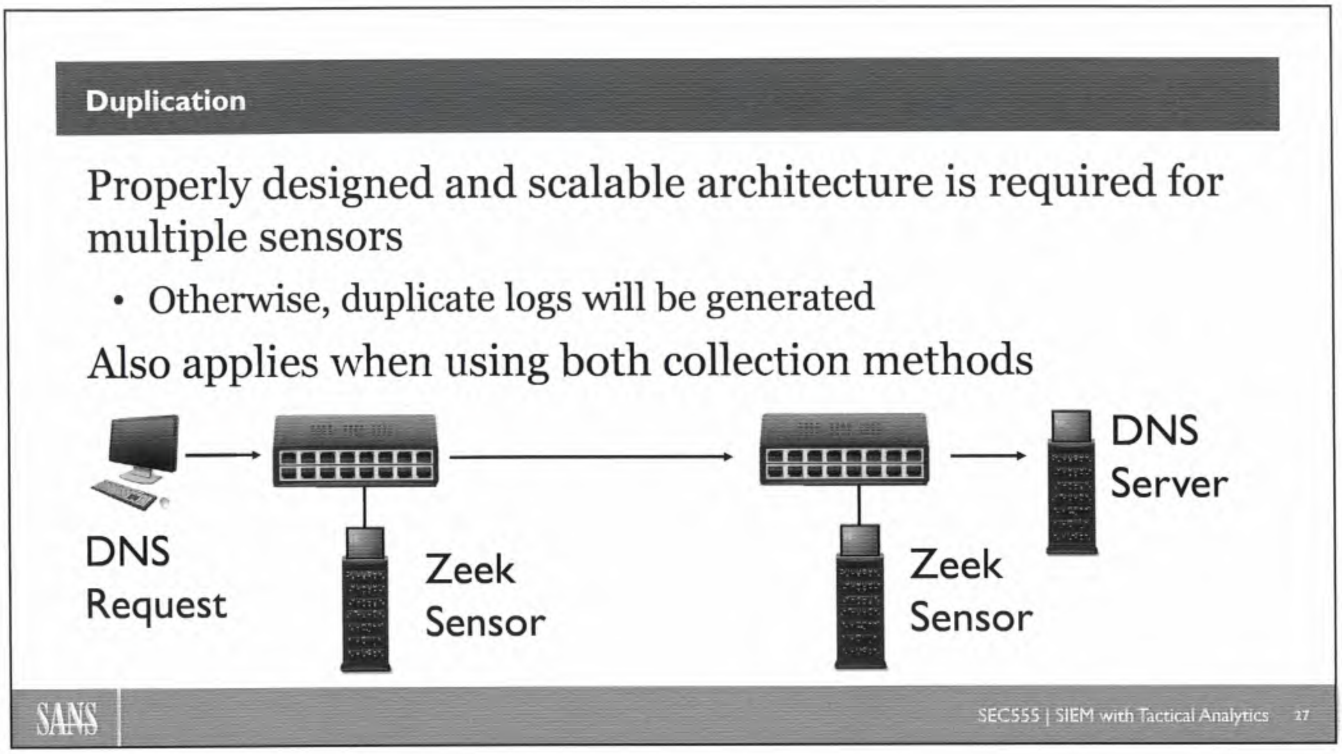
SEC555 | SIEM with Tactical Analytics 26

Drop and Go

Why would you want to deploy a network extractor like Zeek? First off, it is one of the quickest ways to start collecting service logs. Think of it as a mix where you "just add water." In this case, a port mirror or tap is the water that needs to be added to the recipe to be complete. One could argue that the meal Zeek delivers is more like a four-course feast than a quick and easy two-ingredient meal. It simply works. You do not need to build rules or change things. Right from install, it just works.

In this diagram, by adding a port mirror over to the Zeek sensor, you immediately will begin to have DNS, HTTP, HTTPS, etc. logs. Anything that Zeek can see, it will generate logs for. Zeek can even be a system directly browsing the internet that is not going through a proxy. As long as Zeek can see the traffic, it will generate a log for it.

Keep in mind when setting up port mirroring that it is half duplex. For example, if the three servers in this diagram were on a gigabit switch and were sending 550 Mbps and receiving 500 Mbps, then the port mirror would be unable to keep up—assuming it is monitoring both inbound and outbound traffic.



Duplication

Properly designed and scalable architecture is required for multiple sensors

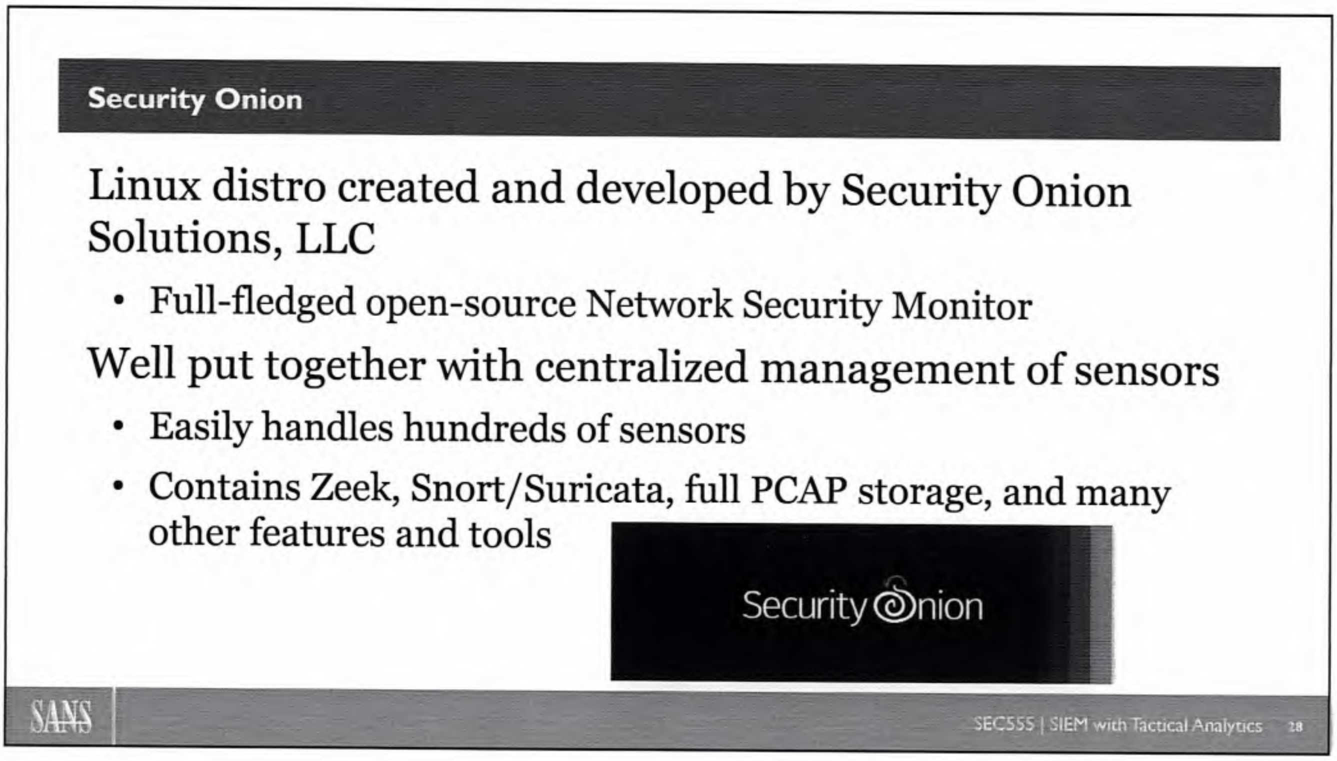
- Otherwise, duplicate logs will be generated

Also applies when using both collection methods

Duplication

When deploying network-based log creation, be careful as to where sensors are placed and what network data is being passed. Improper designs will cause duplicate logs. For example, in this diagram, a workstation is performing a DNS query to the DNS server on the right. The DNS query goes through both switches to reach the DNS server. In this example, each switch has a Zeek sensor monitoring traffic. Depending on how the sensors are set up, two DNS logs for the same request may be generated.

Note that the statement was it depends. It is possible that one of the Zeek sensors is configured never to see DNS packets or maybe it is configured to see all DNS packets except for certain destination IP addresses. Filtering packets can be done with software or hardware and is typically fairly easy.



Security Onion

Linux distro created and developed by Security Onion Solutions, LLC

- Full-fledged open-source Network Security Monitor

Well put together with centralized management of sensors

- Easily handles hundreds of sensors
- Contains Zeek, Snort/Suricata, full PCAP storage, and many other features and tools

Security Onion

Security Onion

Security Onion is one of the most well put together open-source distros out there. Kali is considered the red team distro while Security Onion is considered the blue team distro. In this author's opinion, Security Onion can stand up to—and even beat—many commercial solution alternatives. It supports centralized management of sensors and has been deployed successfully in environments with over 800 remote sensors. It also has gone toe-to-toe against the best-of-breed IDS solutions, and Security Onion has outperformed many of them.

If you are looking for a quick and easy way to deploy multiple Zeek sensors that natively support central management, then you probably are looking for Security Onion. On top of that, it natively supports running either Snort or Suricata, which are open-source IDS solutions as well as having the capability to save and rotate full PCAP storage should you want to store PCAPs. You can also purchase professional support or installation for Security Onion.

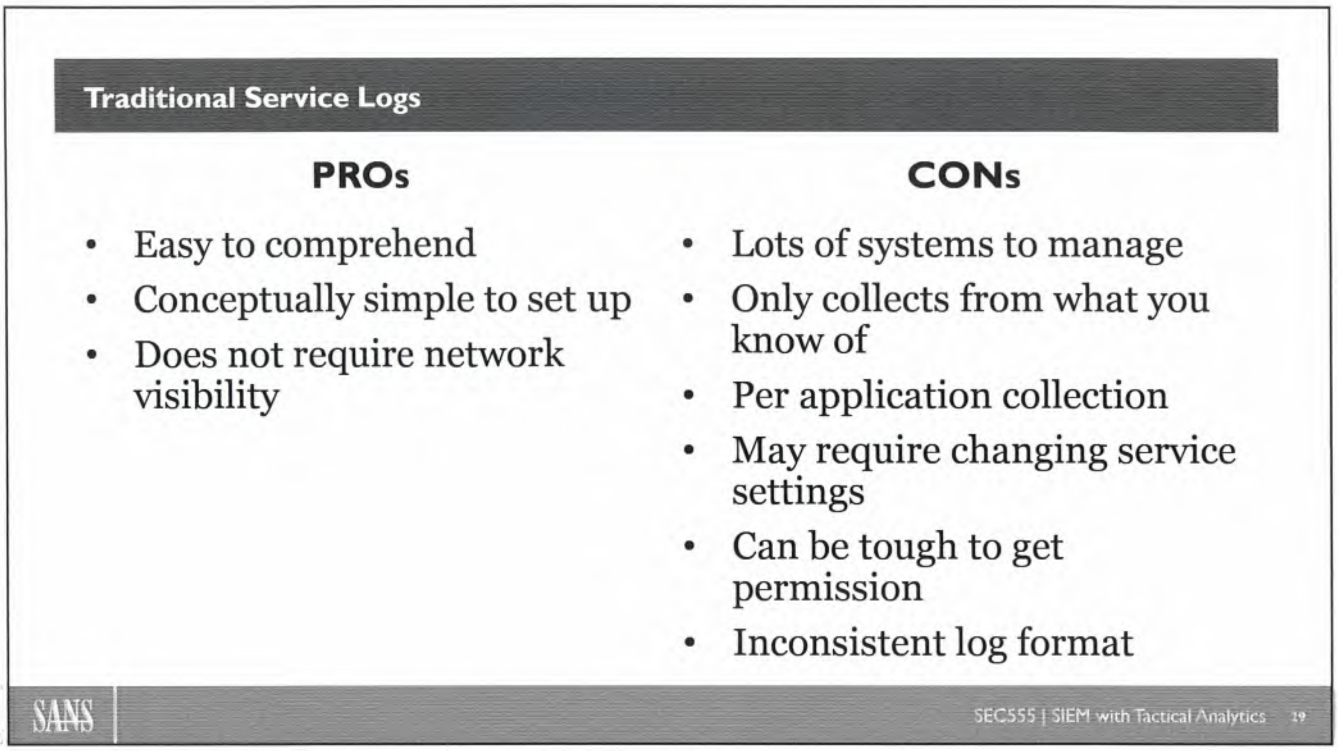
References:

https://securityonionsolutions.com/, https://sec555.com/4j

https://www.kali.org, https://sec555.com/4k

https://snort.org, https://sec555.com/4l

https://suricata-ids.org, https://sec555.com/4m



Traditional Service Logs

PROs

- Easy to comprehend
- Conceptually simple to set up
- Does not require network visibility

CONs

- Lots of systems to manage
- Only collects from what you know of
- Per application collection
- May require changing service settings
- Can be tough to get permission
- Inconsistent log format

SANS

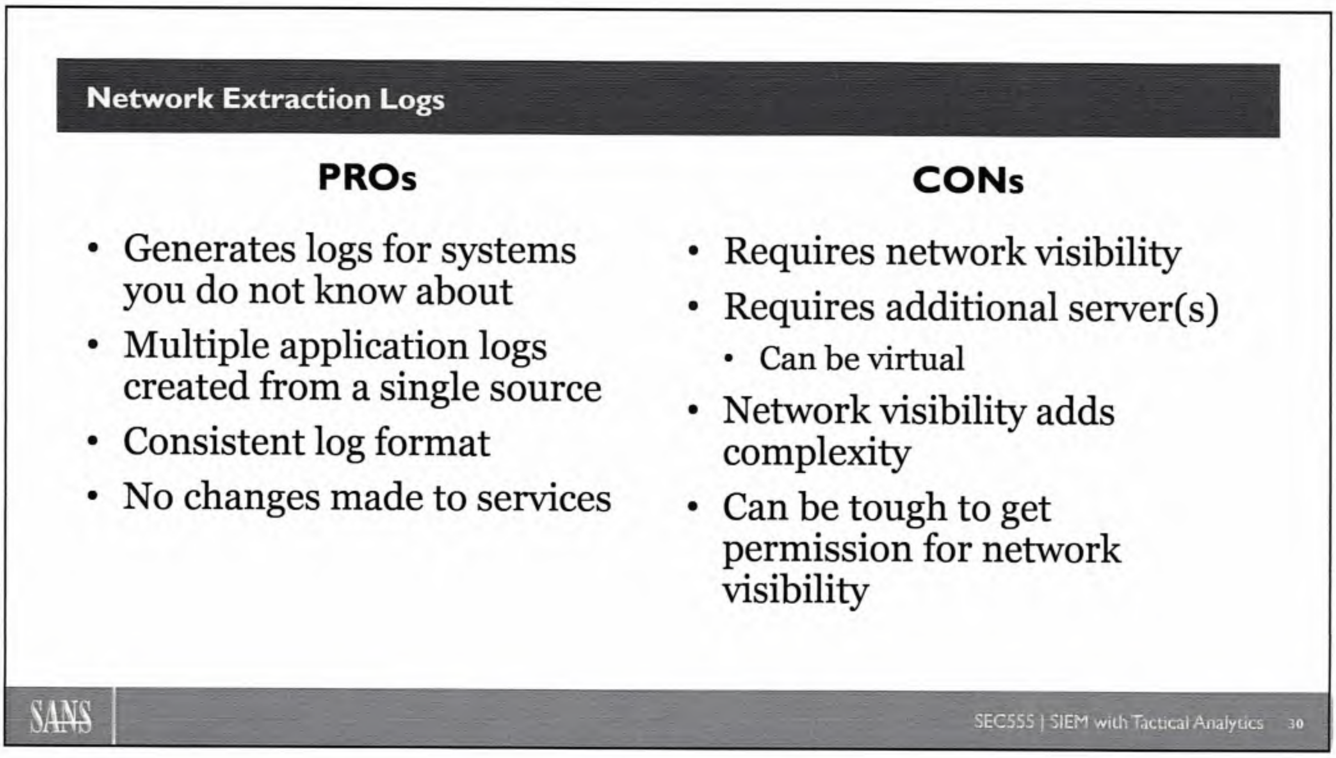
SEC555 | SIEM with Tactical Analytics 29

Traditional Service Logs

Traditional does not mean not effective. After all, collecting logs using an agent is very easy to comprehend, deploy, and use. Also, it does not require any form of network mirroring to work. If this method is simpler for you, then use it.

A drawback to traditional logging is that you have to deploy agents to many systems. As a result, configuration management can be difficult. Also, just because you have a log agent does not mean you can just collect logs. Changes may be required to services such as DNS before the log agent is able to pick up the log. Because of this, you may encounter some issues trying to collect certain logs using traditional agent/agentless methods.

With all that being said, the one main difference between traditional logging and network extraction is that traditional logging is blind to unknown or not yet configured systems. Consider this a nice-to-have setting and try not to justify your choice solely on this.



Network Extraction Logs

PROs

- Generates logs for systems you do not know about
- Multiple application logs created from a single source
- Consistent log format
- No changes made to services

CONs

- Requires network visibility
- Requires additional server(s)
  - Can be virtual
- Network visibility adds complexity
- Can be tough to get permission for network visibility

SANS

SEC555 | SIEM with Tactical Analytics 30

Network Extraction Logs

The main drawback to network extraction is finding the right network location(s) to deploy a sensor and getting permission to mirror traffic to the sensor. Sometimes, this can be a political fight. Keep in mind that depending on the device doing the mirroring, you could only mirror certain traffic, such as port 53 traffic for DNS. Ideally, you can unleash the hose, but some visibility is better than none. Also, note that Security Onion and Zeek can use software filtering not to collect or analyze certain traffic.

In virtual environments, you may be able to get away with deploying Security Onion or Zeek sensors on each hypervisor. Doing so would allow you to see all traffic into or out of your virtual environment. In the past, this author has successfully negotiated contracts to allow these sensors to be deployed and run using the "redundancy" capacity reserved for failures. Then, in the event of a hardware failure, the sensor, if needed, would be temporarily shut off until the systems were back in working condition. For example, if you have five hypervisors, you probably have 25% never used in case one hypervisor fails.

Once deployed, a collection of service logs can occur as long as network visibility is there. Service log collection means the sensor will generate logs of activity for unauthorized or not configured devices. Also, logs from a network sensor tend to follow a consistent pattern, such as tab delimited fields with consistent names, and the logs are centrally generated.

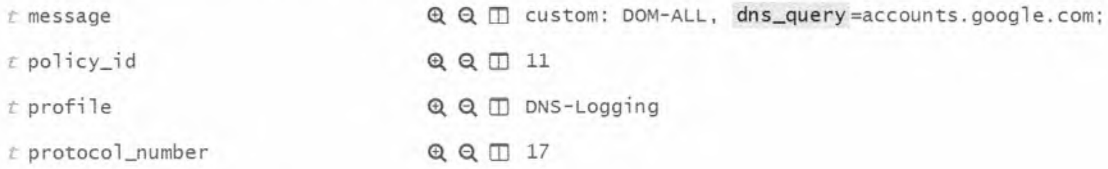
Other Service Log Sources

Outside of traditional and network extraction, there are other ways to collect service logs

- "Next-generation" firewalls can generate logs

Quality, service support, and fields vary dramatically

- But if that is all you have access to... use it

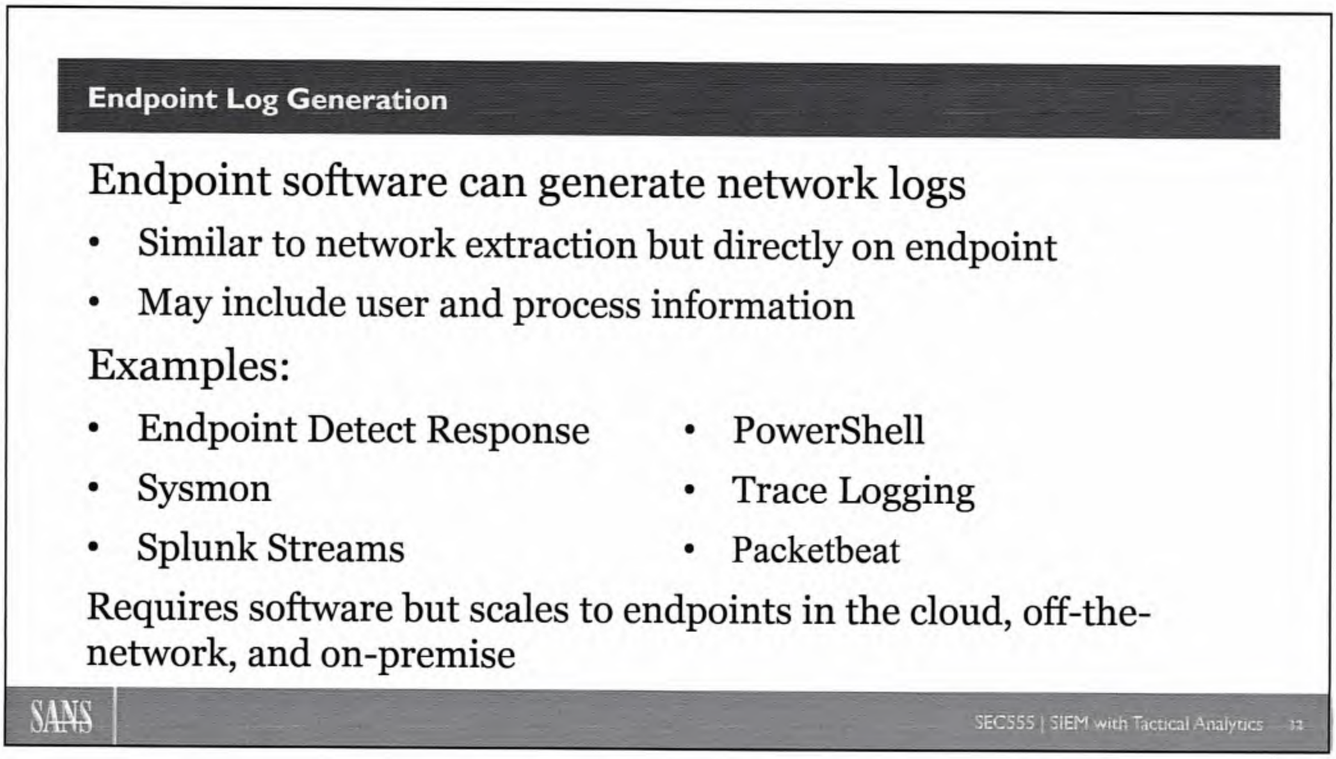


```
t message            custom: DOM-ALL, dns_query=accounts.google.com;
t policy_id          11
t profile            DNS-Logging
t protocol_number    17
```

Other Service Log Sources

In the event you are not able to collect service logs using traditional means, and you do not have permission to go the network extraction route, then it is time to get creative. If you are using devices that are classified as Next Generation, such as a Next Generation firewall, it is possible that you may be able to generate service logs by enabling advanced logging or creating special IPS rules.

In this author's opinion, this should be treated as a last-ditch effort as the quality is not there. For example, you may only be able to get limited data and fields per service. Typically, HTTP can get pretty close, but things like DNS, HTTPS, etc. are subpar.



Endpoint Log Generation

Endpoint software can generate network logs

- Similar to network extraction but directly on endpoint
- May include user and process information

Examples:

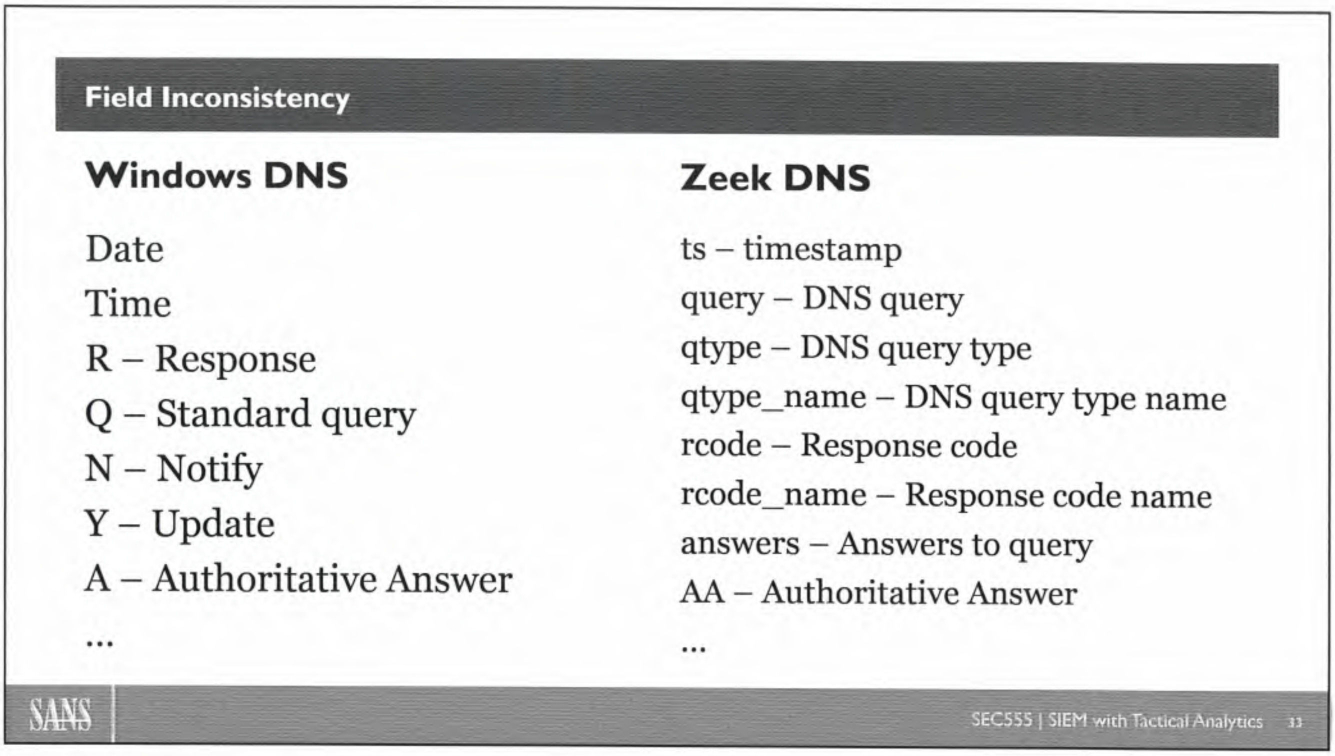
- Endpoint Detect Response
- Sysmon
- Splunk Streams
- PowerShell
- Trace Logging
- Packetbeat

Requires software but scales to endpoints in the cloud, off-the-network, and on-premise

Endpoint Log Generation

A strong alternative for network log collection is to generate them directly on your endpoints. Software exists to analyze network traffic and generate logs such as DNS, HTTP, or connection logs. Network logs are generated directly on the endpoint, and additional information such as what user or process are associated with the traffic may get logged as well.

By generating logs directly on the endpoint, organizations can scale-out log generation and collection to cloud environments, workstations off the network and still maintain traditional on-premise collection methods.



Field Inconsistency

Windows DNS

Date
Time
R – Response
Q – Standard query
N – Notify
Y – Update
A – Authoritative Answer
…

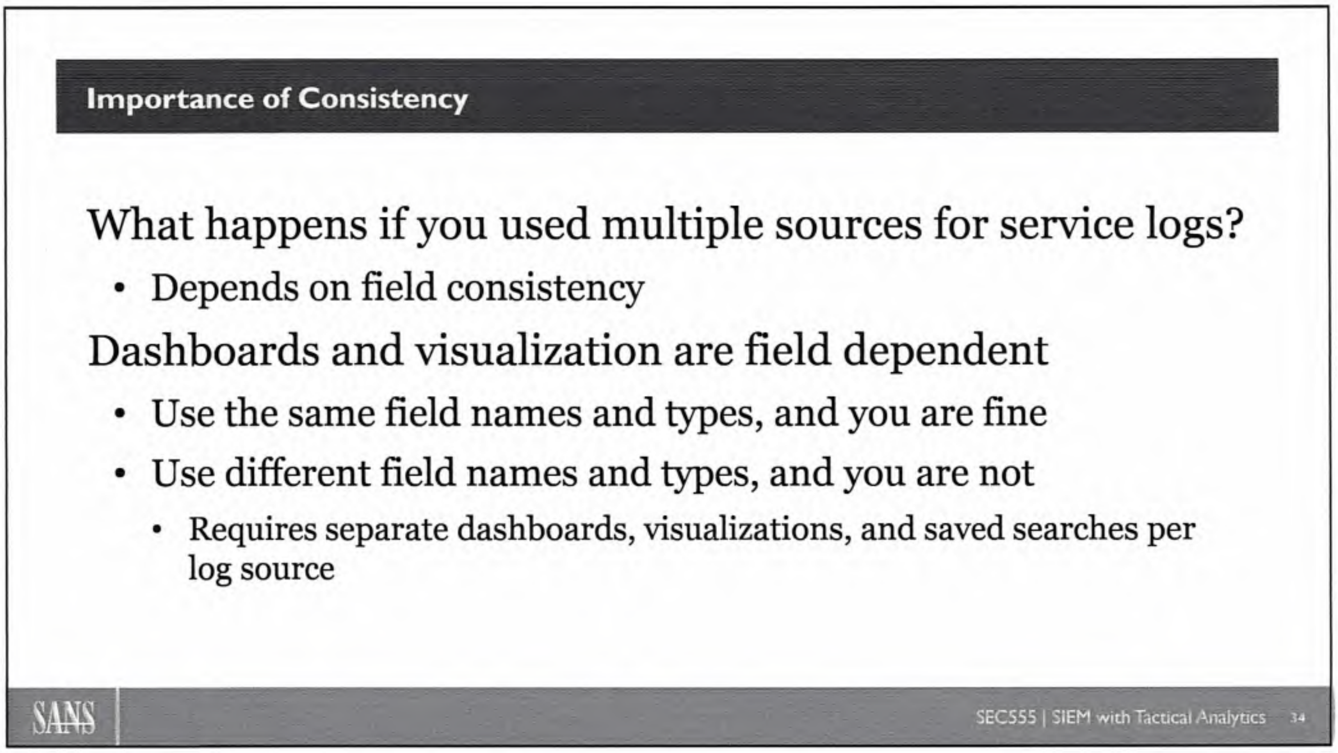
Zeek DNS

ts – timestamp
query – DNS query
qtype – DNS query type
qtype_name – DNS query type name
rcode – Response code
rcode_name – Response code name
answers – Answers to query
AA – Authoritative Answer
…

Field Inconsistency

Unfortunately, logs from various devices rarely are consistent. These logs pose the question, should you care? The answer is, it depends. If you know you are only ever going to use logs from a specific application or device and will never switch, then you can keep things the way they are. However, if you cannot, and want your dashboards, visualizations, searching, alerts, etc. to work consistently, then you better care.

This slide demonstrates the dramatic difference between one log source to the next.



Importance of Consistency

What happens if you used multiple sources for service logs?

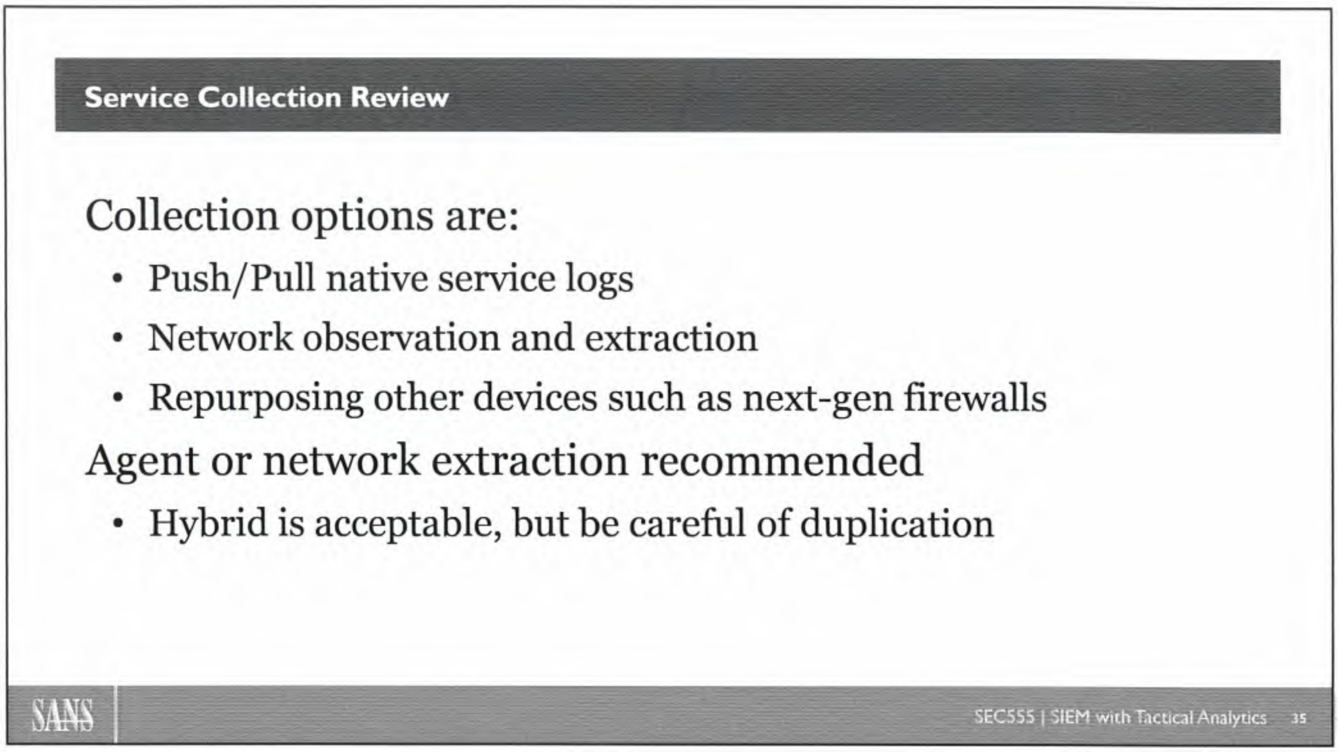
- Depends on field consistency

Dashboards and visualization are field dependent

- Use the same field names and types, and you are fine
- Use different field names and types, and you are not
  - Requires separate dashboards, visualizations, and saved searches per log source

Importance of Consistency

Ideally, logs would be modified to follow similar field names and consistent data values. Without this, you have chaos. With this, you can take multiple sources of the same types of logs and combine them for applicable use. For example, if an organization had two different web proxy products or was in the middle of migrating from one to another, all their data, dashboards, visualizations, alerting, etc. would function like normal as long as the two log sources were transformed into a single consistent fashion.



Service Collection Review

Collection options are:

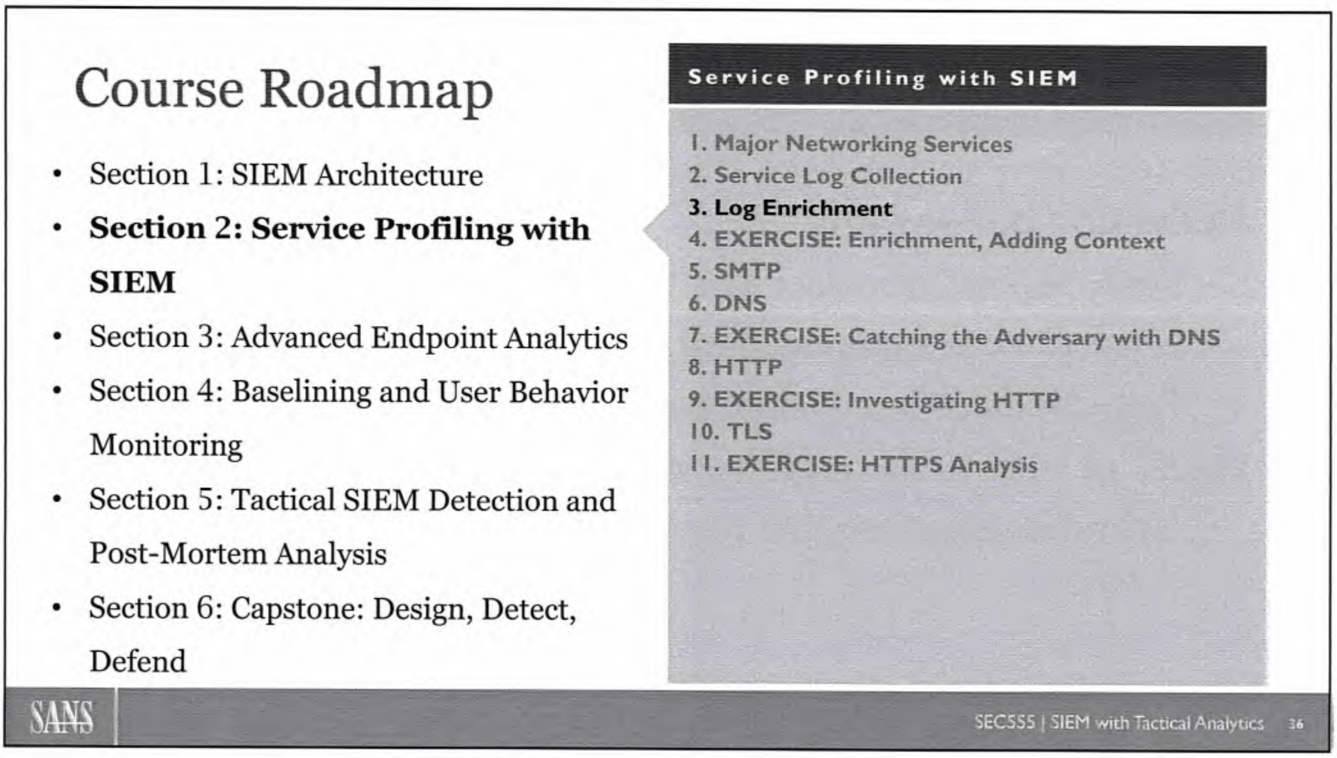
- Push/Pull native service logs
- Network observation and extraction
- Repurposing other devices such as next-gen firewalls

Agent or network extraction recommended

- Hybrid is acceptable, but be careful of duplication

Service Collection Review

It is important to choose one or more methods to collect network service logs. The recommendation is to use an agent, extract logs from the network, or a combination of the two. If using both, be cautious of duplicating logs.



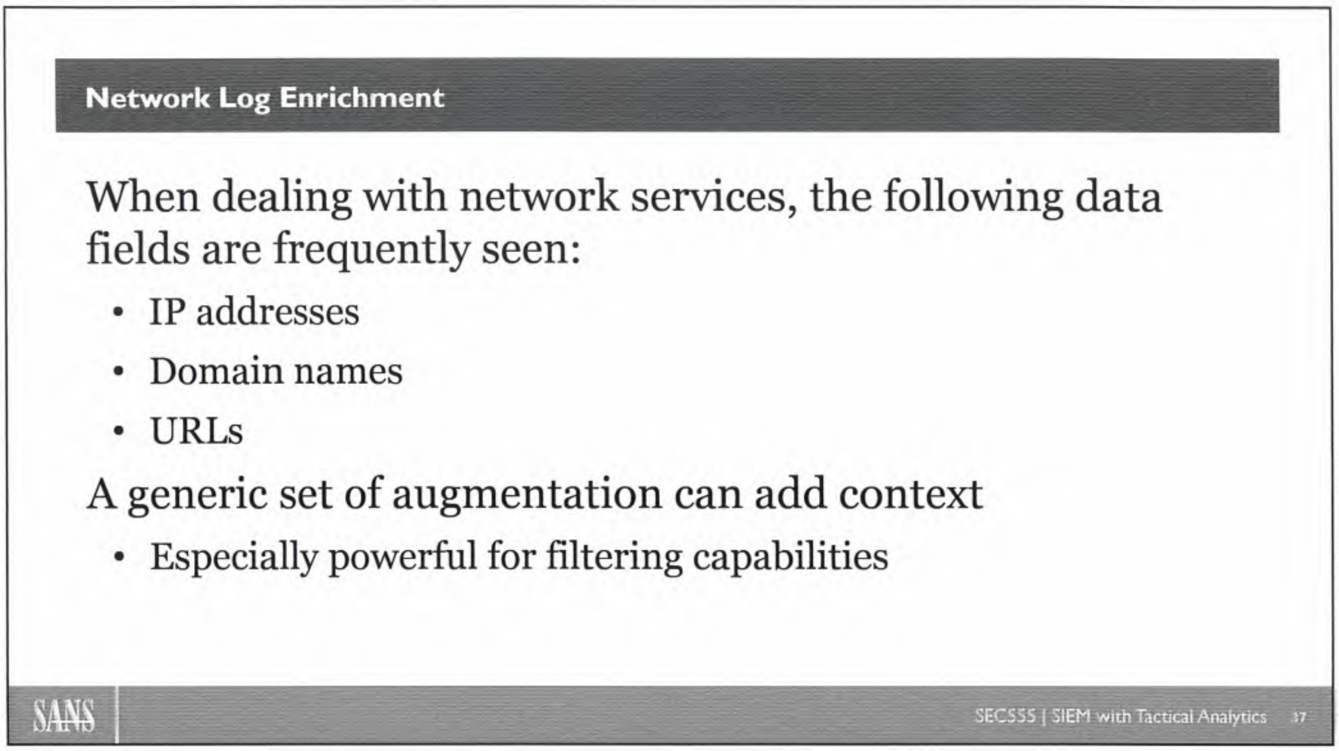
Course Roadmap

- Section 1: SIEM Architecture
- **Section 2: Service Profiling with SIEM**
- Section 3: Advanced Endpoint Analytics
- Section 4: Baselining and User Behavior Monitoring
- Section 5: Tactical SIEM Detection and Post-Mortem Analysis
- Section 6: Capstone: Design, Detect, Defend

Service Profiling with SIEM

1. Major Networking Services
2. Service Log Collection
3. **Log Enrichment**
4. EXERCISE: Enrichment, Adding Context
5. SMTP
6. DNS
7. EXERCISE: Catching the Adversary with DNS
8. HTTP
9. EXERCISE: Investigating HTTP
10. TLS
11. EXERCISE: HTTPS Analysis

This page intentionally left blank.



Network Log Enrichment

When dealing with network services, the following data fields are frequently seen:

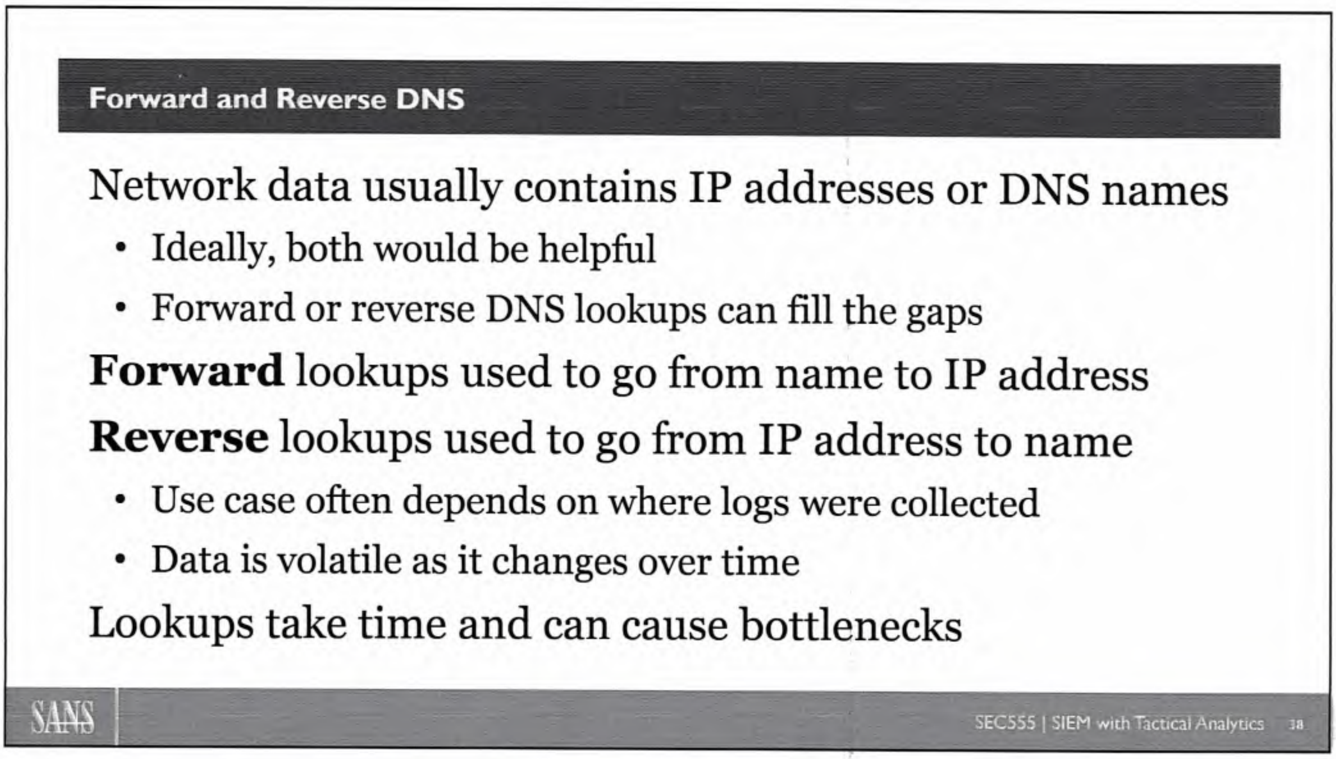
- IP addresses
- Domain names
- URLs

A generic set of augmentation can add context

- Especially powerful for filtering capabilities

Network Log Enrichment

When dealing with network logs, there are multiple things that can be done to augment them. Many of these are needed to filter out noise that would otherwise drown out a detection technique.



Forward and Reverse DNS

Network data usually contains IP addresses or DNS names

- Ideally, both would be helpful
- Forward or reverse DNS lookups can fill the gaps

Forward lookups used to go from name to IP address

Reverse lookups used to go from IP address to name

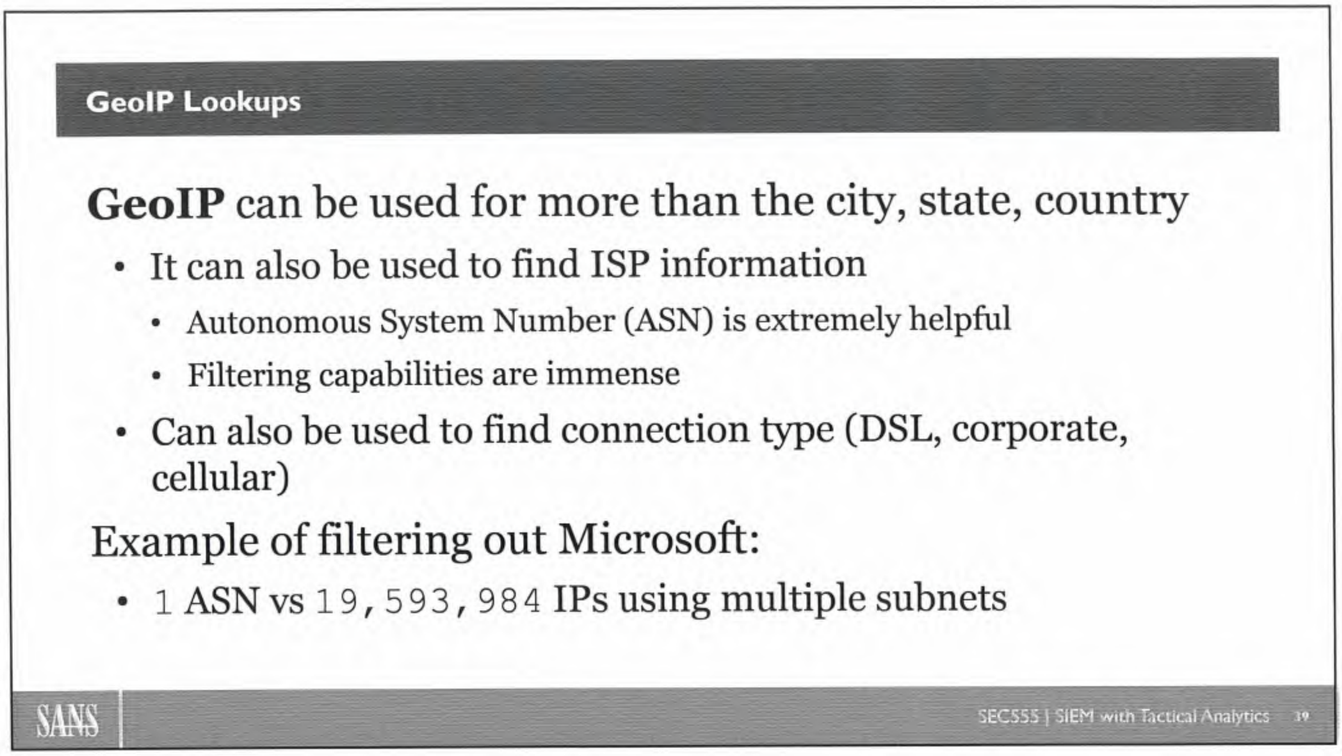
- Use case often depends on where logs were collected
- Data is volatile as it changes over time

Lookups take time and can cause bottlenecks

Forward and Reverse DNS

Knowing both IP addresses and DNS names are helpful for both filtering as well as to give context to an analyst. Forward lookups perform DNS requests for A records and return an IP address. Reverse lookups perform DNS requests for PTR records and return a DNS name.

If your DNS environment is not accurate, then beware. Using these lookups is for filtering and context. Incorrect responses can cause an analyst to interpret a log incorrectly.



GeoIP Lookups

GeoIP can be used for more than the city, state, country

- It can also be used to find ISP information
  - Autonomous System Number (ASN) is extremely helpful
  - Filtering capabilities are immense
- Can also be used to find connection type (DSL, corporate, cellular)

Example of filtering out Microsoft:

- `1` ASN vs `19,593,984` IPs using multiple subnets

SEC555 | SIEM with Tactical Analytics 39

GeoIP Lookups

While GeoIP lookups can be helpful for looking for traffic to other countries, this author finds them more useful in filtering. For example, Windows systems make many calls out to Microsoft using various IP addresses and domain names. However, this kind of traffic can be easily filtered out using ASN 8075 for the Microsoft Corporation. This works amazingly well when combined with various detection techniques.

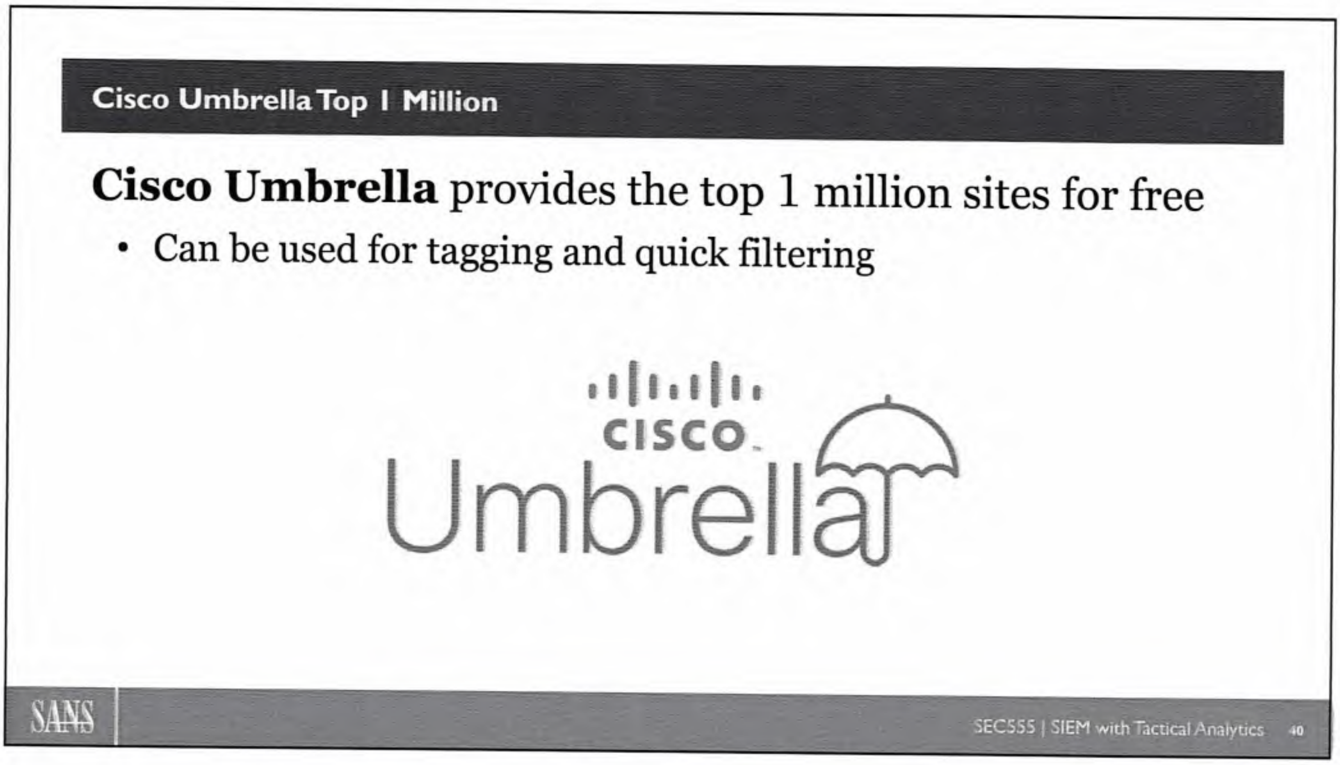
Which would you rather do, filter out almost 20 million IP addresses that spread out over many subnets or filter out a single ASN? Oddly enough, many commercial SIEMs do not provide ASN support.

MaxMind[1] provides GeoIP databases for city, country, and ASN that are free. The ASN database requires downloading the database in a legacy format. For more current information or to add additional GeoIP databases such as connection type, MaxMind offers low-cost subscriptions. In the author's opinion, the free databases have been accurate enough even though the subscriptions are not that expensive.

New versions of Logstash require the new MaxMind binary database format. There are free ASN number databases as well as more current commercial ones. As of 6/27/2019, a commercial license costs $1,470 annually. This is well worth the investment.

Reference:

https://dev.maxmind.com/geoip/geolite2-free-geolocation-data , https://sec555.com/4n



Cisco Umbrella Top 1 Million

Cisco Umbrella provides the top 1 million sites for free

- Can be used for tagging and quick filtering

cisco
Umbrella

SANS

SEC555 | SIEM with Tactical Analytics 40

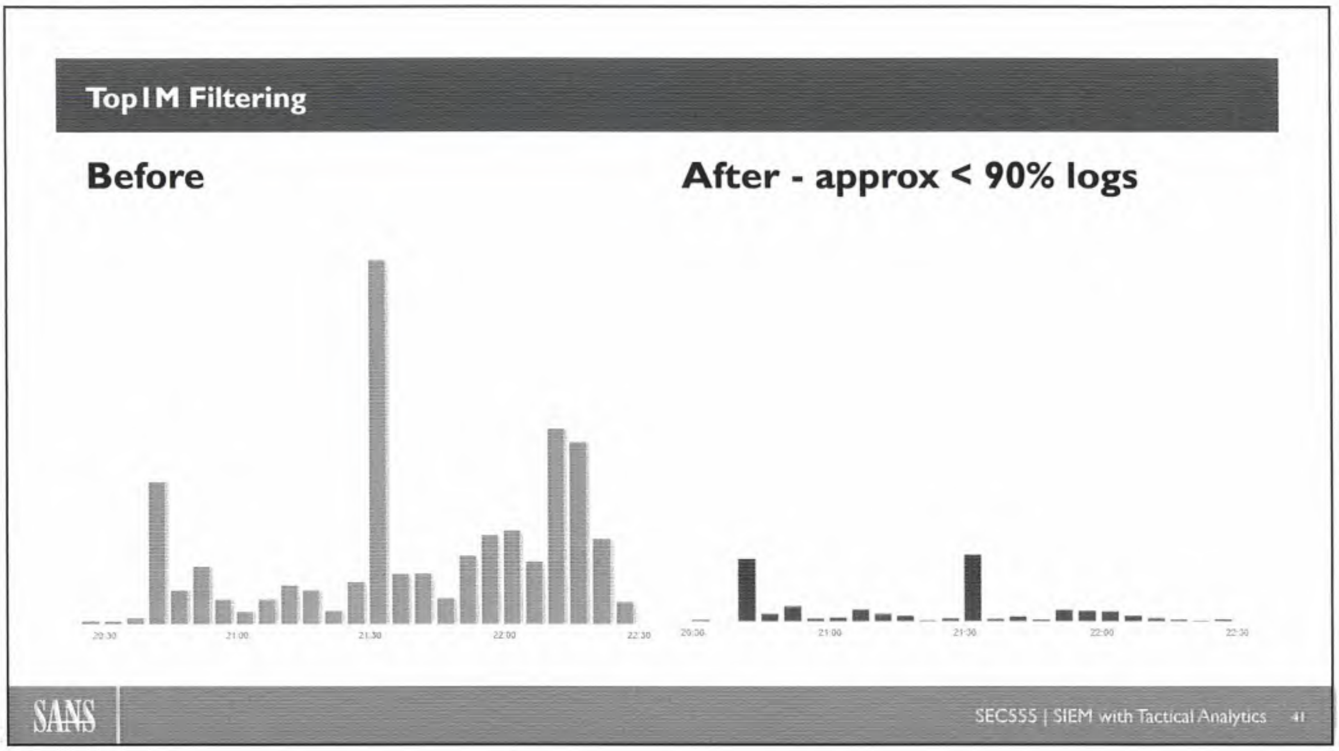
Cisco Umbrella Top 1 Million

The **Cisco Umbrella** Top 1 Million[1] is a listing of the most queried domains based on passive DNS usage across our Umbrella global network of more than 100 Billion requests per day with 65 million unique active users, in more than 165 countries. Unlike Alexa, the metric is not based on only browser based 'http' requests from users but rather takes in to account the number of unique client IPs invoking this domain relative to the sum of all requests to all domains. In other words, our popularity ranking reflects the domain's relative internet activity agnostic to the invocation protocols and applications where as 'site ranking' models (such as Alexa) focus on the web activity over port 80 mainly from browsers.

This type of list is great for filtering and context. For example, adding a tag of umbrella-to-umbrella domains allows filtering out domains quickly.

Reference:

http://s3-us-west-1.amazonaws.com/umbrella-static/index.html, http://sec555.com/105



Top1M Filtering

Before

After - approx < 90% logs

SANS

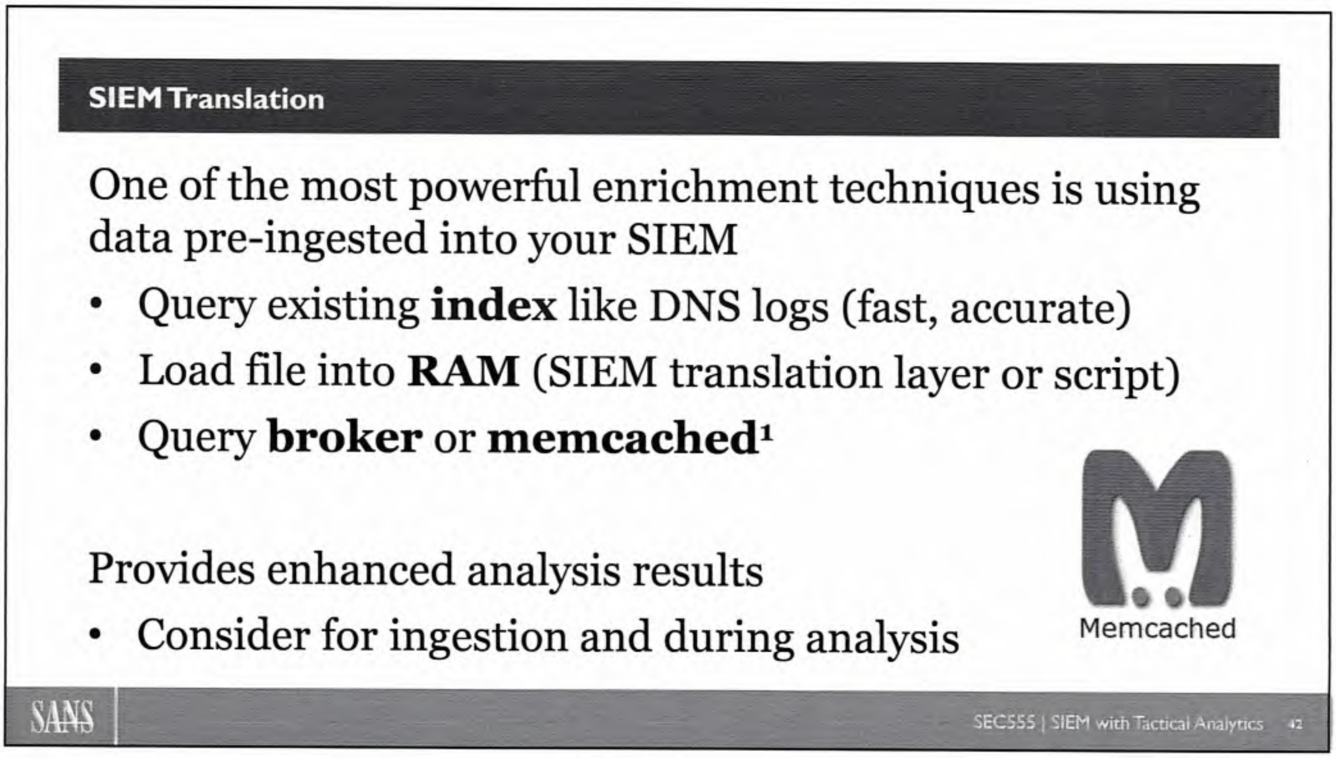
SEC555 | SIEM with Tactical Analytics 41

Top1M Filtering

On the left is a bar graph of DNS logs from 8:30 p.m. to 10:30 p.m. on a given day. On the right is the same timeframe and logs but filtering out anything with a tag of top1m. While this does not inherently find evil, it does shrink the amount of data you may need to analyze by almost 90%. This type of technique is fantastic for manual investigations or forensics. It also is useful for limiting intensive resource augmentation to only a subset of log data that it is needed on.

The top 1 million filtering is a tactical security decision and works under the assumption that the top 1 million sites are secure. Assuming the top 1 million sites are benign is not always true. Also, this does not account for corporate policy, as many sites are likely to be unauthorized by policy. However, it does help to filter and find events of interest tactically. Filtering out the top 1 million sites can be a starting point during an investigation. Then, if this does not find anything, the filter can be removed.

If push came to shove, you could also drop any logs that are from systems within the Cisco top 1 million instead of sending them to Elasticsearch.



SIEM Translation

One of the most powerful enrichment techniques is using data pre-ingested into your SIEM

- Query existing **index** like DNS logs (fast, accurate)
- Load file into **RAM** (SIEM translation layer or script)
- Query **broker** or **memcached**[1]

Provides enhanced analysis results

- Consider for ingestion and during analysis

SANS

SEC555 | SIEM with Tactical Analytics 42

SIEM Translation

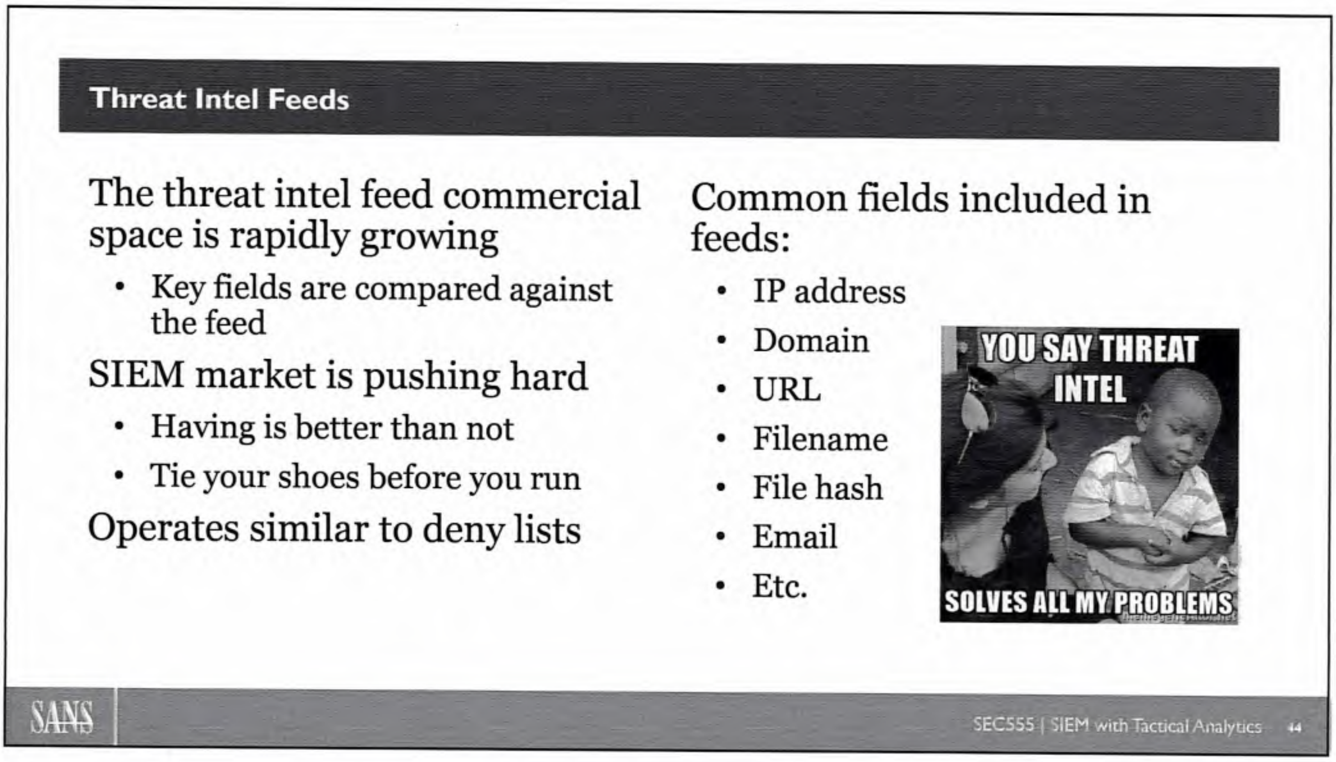
One of the most powerful enrichment techniques is pre-loaded data into your SIEM for querying. For example, DNS logs that already exist in your SIEM can be utilized to find which domain a given client queried regarding an IDS destination IP. Fortunately, this lookup technique can be used with all SIEMs as there is usually more than one way to do an automated lookup. Consider applying this technique during ingestion to help create more alerts as well as more actionable alerts, also while during analysis and investigations.

Regarding automating Alexa imports, you can create a simple script that downloads and extracts the top 1 million domains and then has Logstash pick it up and put it into Elasticsearch. In fact, here would be the Logstash config to do so:

```
input {
  file {
    path => "/lib/logstash_data/top-1m.csv"
    type => "top1m"
  }
}
filter {
  if [type] == "top1m" {
    csv {
      columns => [ "top1m_number", "site" ]
      separator => ","
      remove_field => [ "message" ]
    }
    mutate {
      convert => [ "top1m_number", "integer" ]
    }
  }
}
output {
  if [type] == "top1m" {
    elasticsearch {
      index => "logstash-top1m-%{+YYYY.MM.dd}"
    }
  }
}
```

A similar approach would work for ingesting the data into a SIEM index or a Memcached system.

Reference:
https://memcached.org/



Threat Intel Feeds

The threat intel feed commercial space is rapidly growing

- Key fields are compared against the feed

SIEM market is pushing hard

- Having is better than not
- Tie your shoes before you run

Operates similar to deny lists

Common fields included in feeds:

- IP address
- Domain
- URL
- Filename
- File hash
- Email
- Etc.

YOU SAY THREAT INTEL
SOLVES ALL MY PROBLEMS

SANS

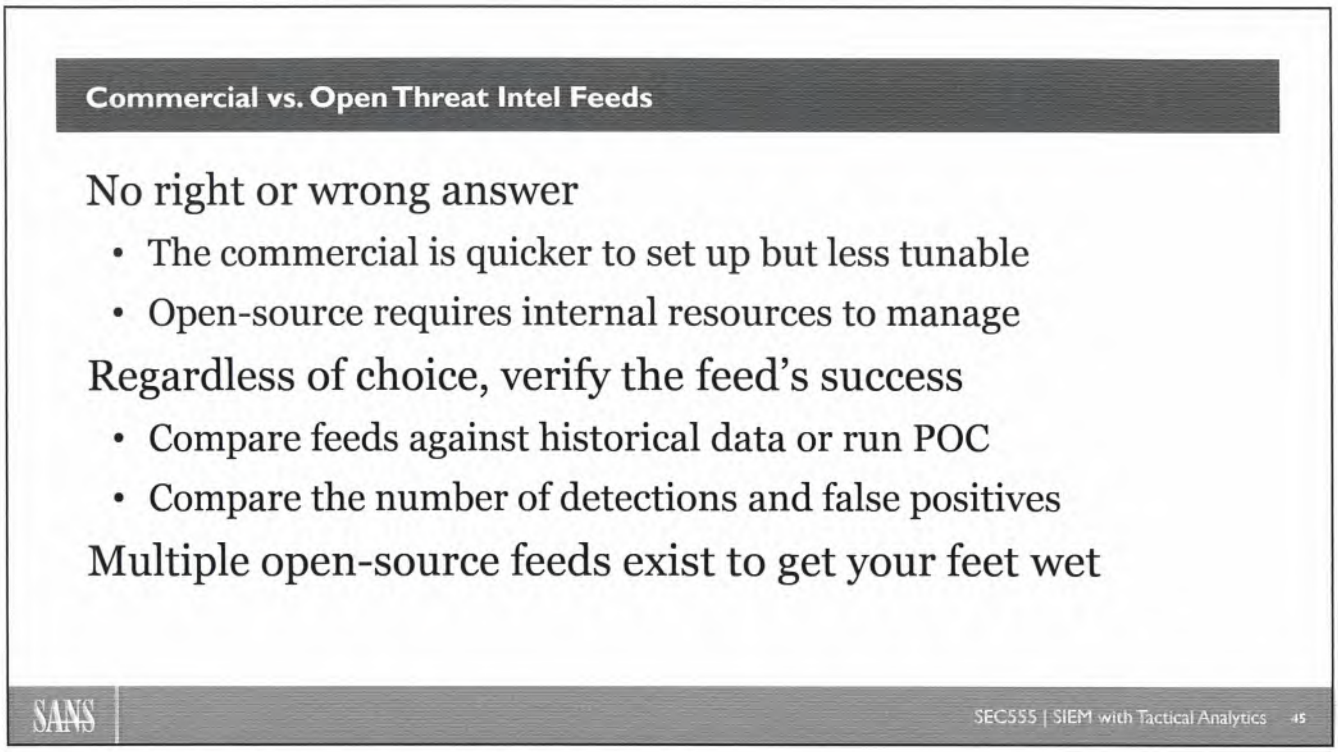
SEC555 | SIEM with Tactical Analytics 44

Threat Intel Feeds

The SIEM industry is pushing the concept of threat intel feeds with passion. This is very similar to how originally a firewall was just a Layer 4 device. Then came all the UTM functionality such as antivirus, network intrusion prevention, etc. While having a threat intel feed is obviously better than not having one, the concept is overrated.

Threat intel feeds are effectively a deny list of known bad actors. They cannot help with new attacks or targeted attacks. Granted, some may believe that they will never get targeted, thus a threat feed is perfect. However, even if you are not being targeted, malware will develop and attack faster than your intel feed.

The market push has been so strong that it has led consumers to have an unjustified sense of security. This author recommends focusing on good hygiene and people and processes. A threat intel feed is a great addition, but if it takes time away from proper hygiene and your people and processes, then change your focus until those are addressed.



Commercial vs. Open Threat Intel Feeds

No right or wrong answer

- The commercial is quicker to set up but less tunable
- Open-source requires internal resources to manage

Regardless of choice, verify the feed's success

- Compare feeds against historical data or run POC
- Compare the number of detections and false positives

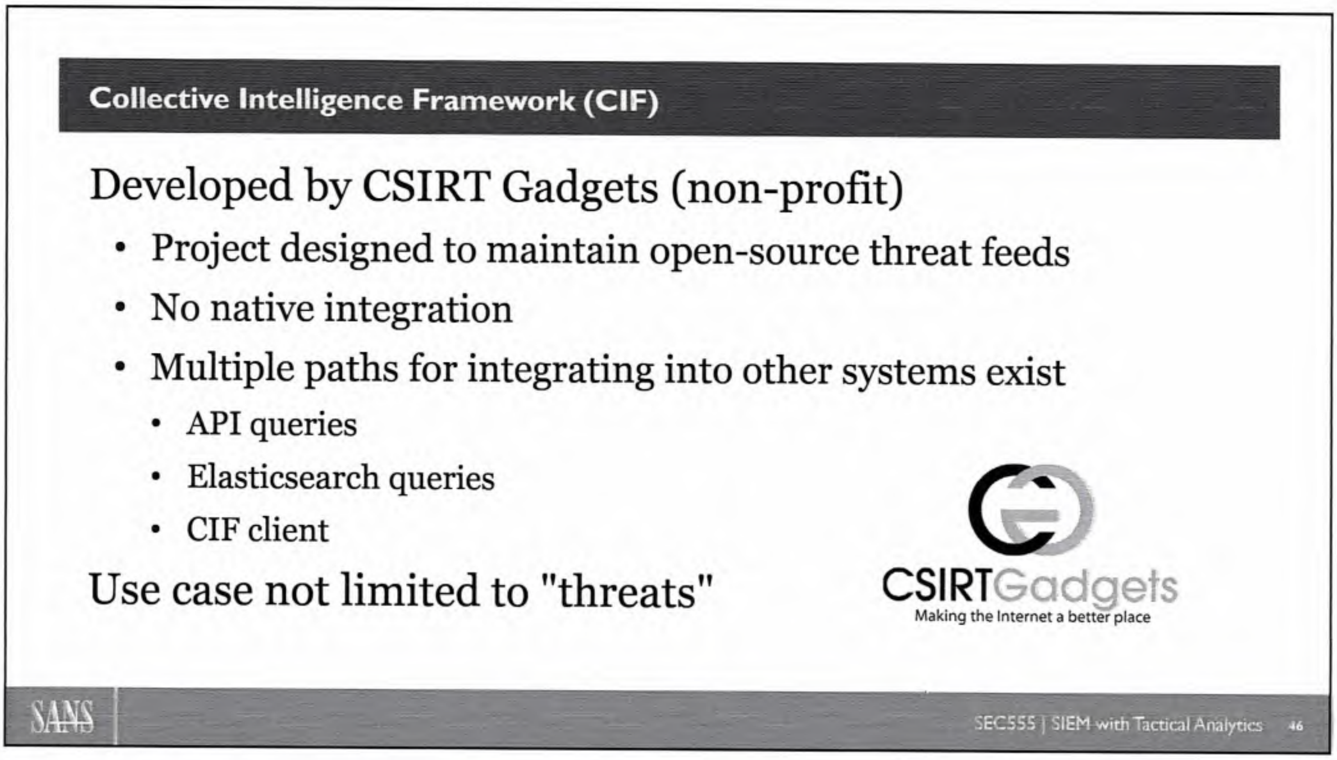
Multiple open-source feeds exist to get your feet wet

Commercial vs. Open Threat Intel Feeds

In response to which is better, the answer is simply to test and find out. From one commercial vendor to another can be highly different in this market space mainly due to the quality of the threat feed rather than the product delivering it. If you are interested in finding a strong threat intel feed, this author recommends trying a few open-source solutions and pitting them against one or more commercial solutions.

Threat intel feeds are interesting as many organizations follow the "set it and forget it" approach. Using a feed and measuring its success is quantifiable by monitoring how many times it is triggering and how many of those were false positives. If you are going to "set it and forget it," why bother in the first place?

In this case, "set it and forget it" is referring to having threat intel data in your logs that you never use. Many threat feeds can be set up and will automatically keep their feeds up to date.



Collective Intelligence Framework (CIF)

Developed by CSIRT Gadgets (non-profit)

- Project designed to maintain open-source threat feeds
- No native integration
- Multiple paths for integrating into other systems exist
  - API queries
  - Elasticsearch queries
  - CIF client

Use case not limited to "threats"

CSIRTGadgets
Making the Internet a better place

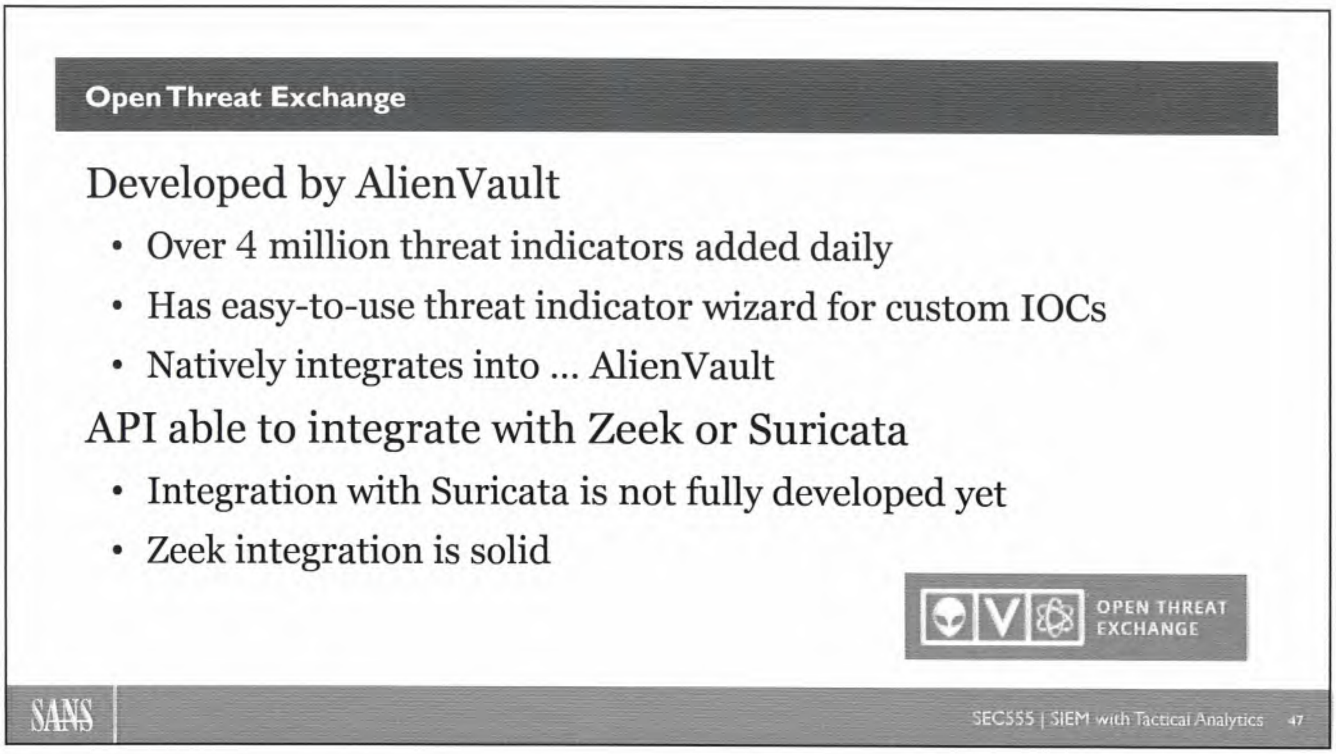
Collective Intelligence Framework (CIF)

CIF, or the Collective Intelligence Framework, is a framework for pulling in feeds from multiple locations and making them usable for other systems or devices. A default installation includes multiple open-source intelligence (OSINT) feeds that run every hour with a random offset of 30 minutes.

This framework includes threat feeds but also feeds used for allow lists. Effectively, it pulls down lists of known good sites such as Google and uses them to tag data. Lists of known good sites work great for filtering out the trusted noise. In fact, one of the default feeds is to pull in the Umbrella top 1 million. Adding custom feeds to CIF is fairly easy and requires putting new rule files in /etc/cif/rules.

Reference:

https://csirtgadgets.com/collective-intelligence-framework, http://sec555.com/110



Open Threat Exchange

Developed by AlienVault

- Over 4 million threat indicators added daily
- Has easy-to-use threat indicator wizard for custom IOCs
- Natively integrates into ... AlienVault

API able to integrate with Zeek or Suricata

- Integration with Suricata is not fully developed yet
- Zeek integration is solid

OPEN THREAT EXCHANGE

Open Threat Exchange

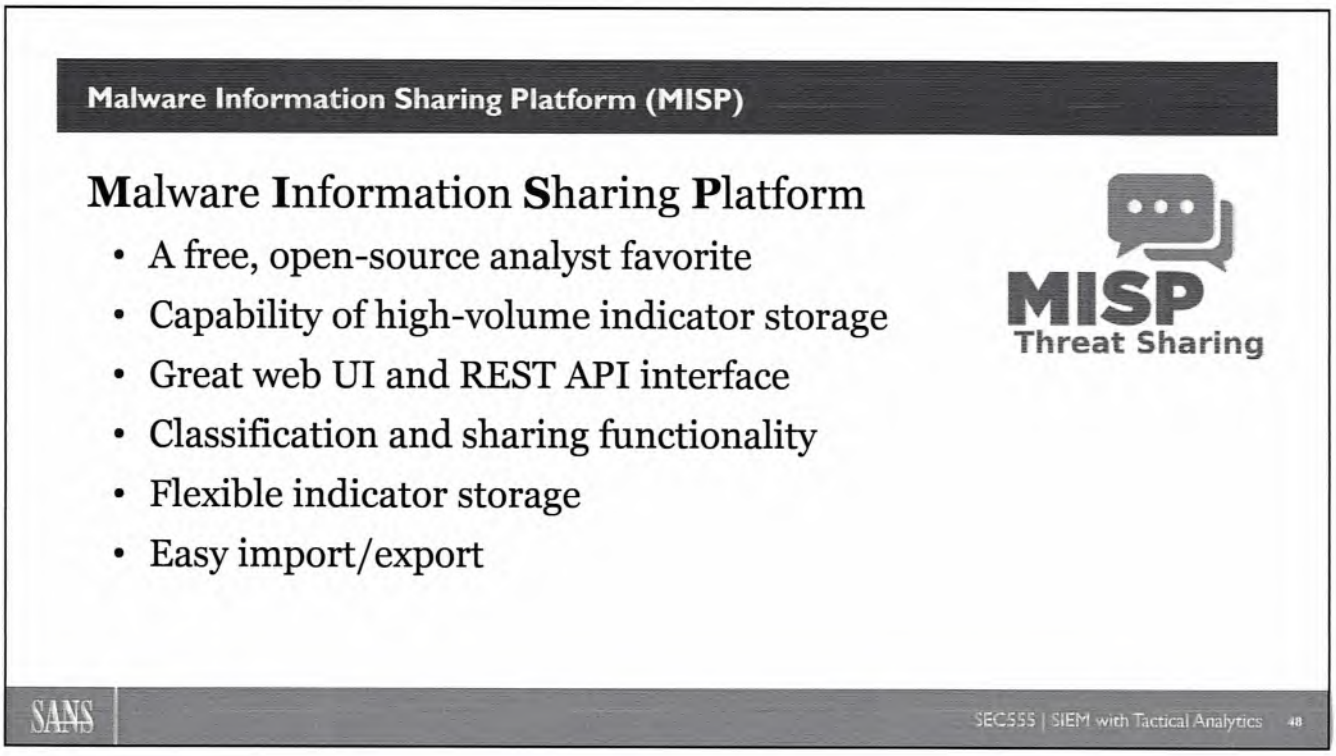
AlienVault[2] sponsors the Open Threat Exchange. This is a free threat feed in use by over 47,000 participants in over 140 countries. To integrate, sign up for a free account and get an API key. Then, subscribe to data feeds. After that, follow one of the integration guides such as for Zeek or Suricata to start using the feed.

This feed is fairly easy to set up, even for open-source solutions. The integration is best with AlienVault's product, but setup with other products is also supported.

References:

https://otx.alienvault.com/, https://sec555.com/4q

https://cybersecurity.att.com/products/ossim, https://sec555.com/4r



Malware Information Sharing Platform (MISP)

Malware **I**nformation **S**haring **P**latform

- A free, open-source analyst favorite
- Capability of high-volume indicator storage
- Great web UI and REST API interface
- Classification and sharing functionality
- Flexible indicator storage
- Easy import/export

MISP
Threat Sharing
SANS
SEC555 | SIEM with Tactical Analytics 48

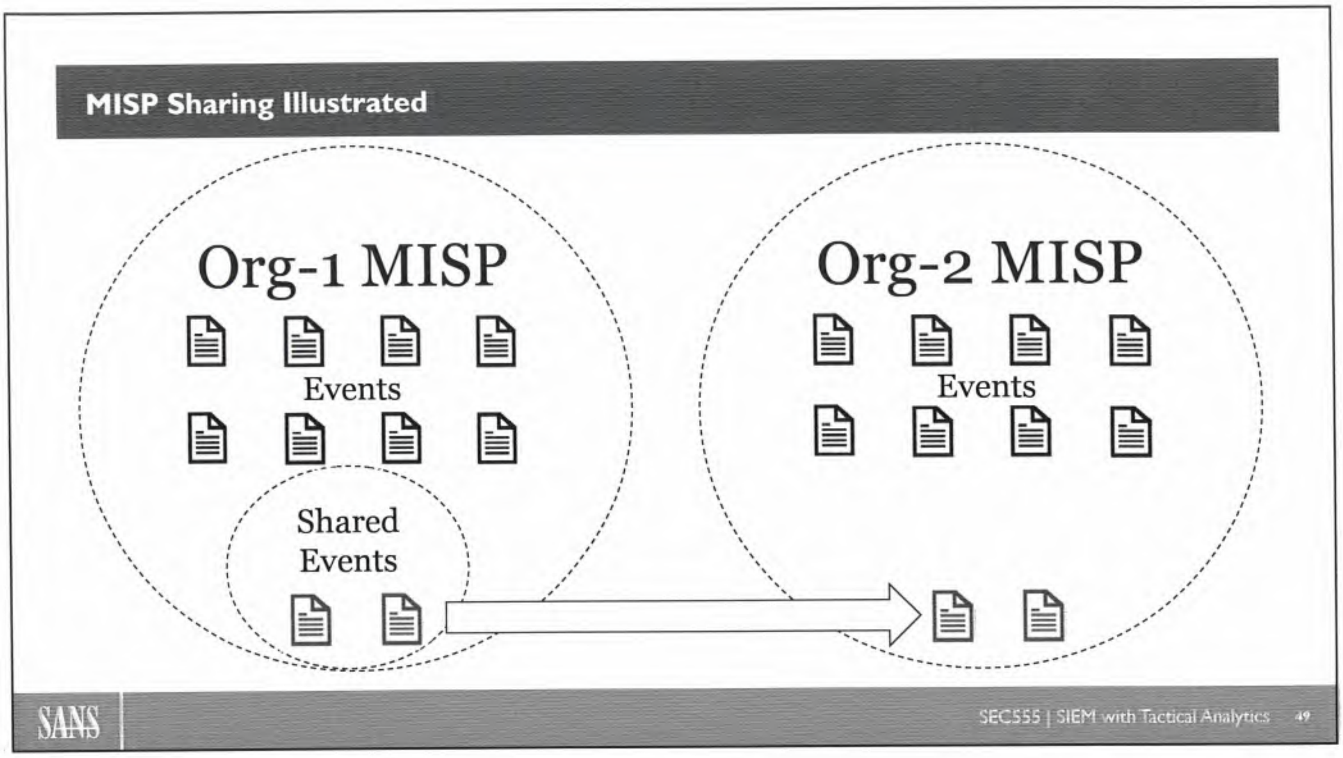
Malware Information Sharing Platform (MISP)

One of the most feature rich and trending threat intelligence sharing platforms available is the Malware Information Sharing Platform. MISP, by default, integrates with multiple free threat intelligence feeds, but it also natively supports custom feeds and sharing. For example, organizations can build their own list of indictors and then correlate them internally to find repeat offenders or adversaries changing tactics over time.

The sharing capabilities of MISP are a key capability. Organizations can choose which organizations they wish to push or pull feeds with as well as selectively share their own intelligence. For example, an organization can record indicators it identifies and then share them only with partnering organizations. Even then, sharing does not have to mean full access as MISP can granularly control who can see what and what information is shared.

Reference:

https://www.misp-project.org, https://sec555.com/4s



MISP Sharing Illustrated

Org-1 MISP

Events

Shared Events

Org-2 MISP

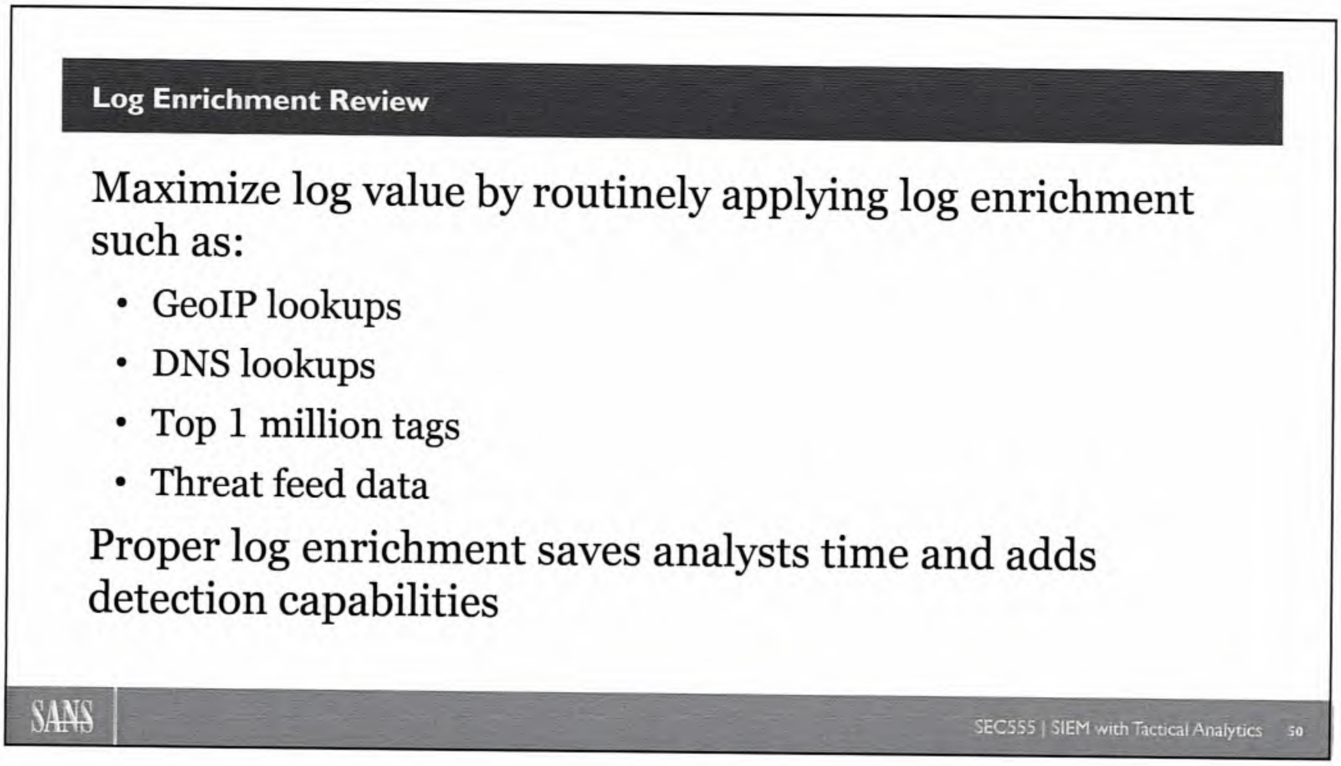
Events

SANS

SEC555 | SIEM with Tactical Analytics 49

MISP Sharing Illustrated

This slide visually represents sharing events or indicators from one organization to another. In the example on the slide, events from organization one are shared one-way to organization two.



Log Enrichment Review

Maximize log value by routinely applying log enrichment such as:

- GeoIP lookups
- DNS lookups
- Top 1 million tags
- Threat feed data

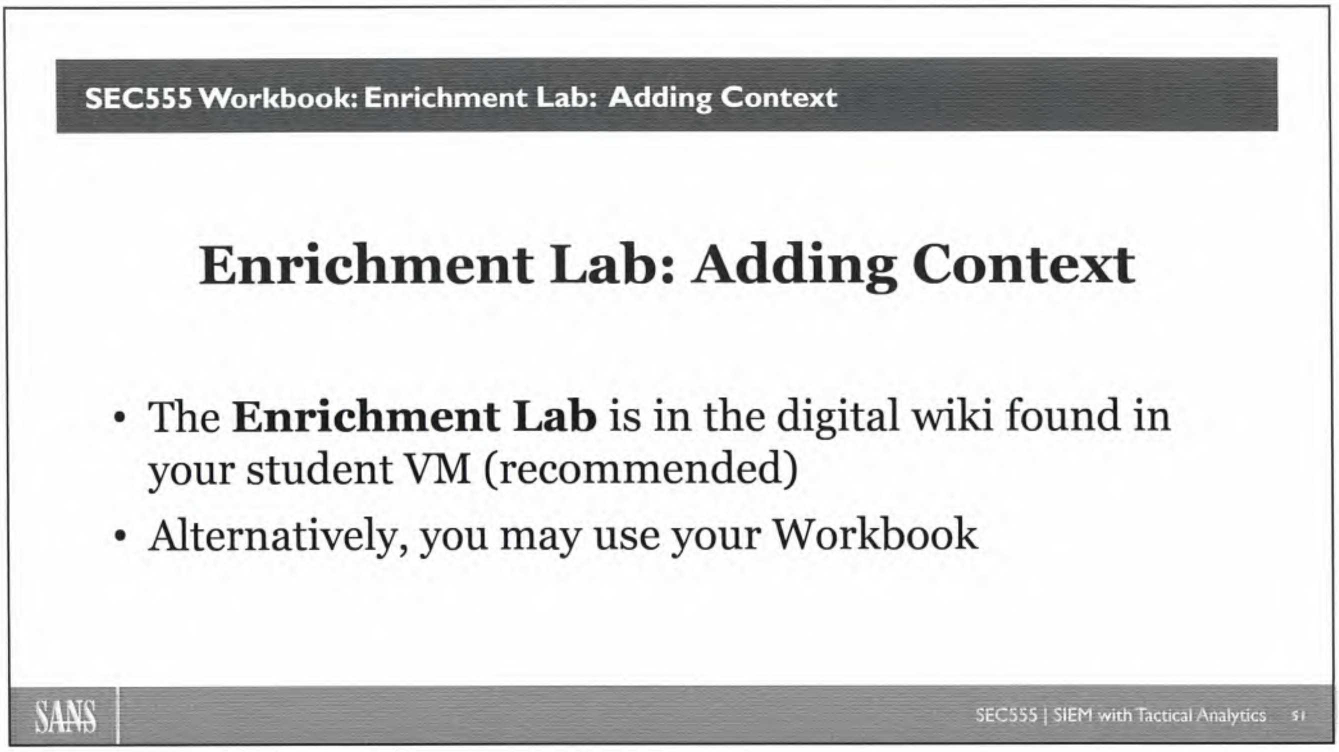
Proper log enrichment saves analysts time and adds detection capabilities

SANS

SEC555 | SIEM with Tactical Analytics 50

Log Enrichment Review

One of the most powerful things to consider when dealing with logs is proper log enrichment. A few minor additions to a log can greatly increase its value.



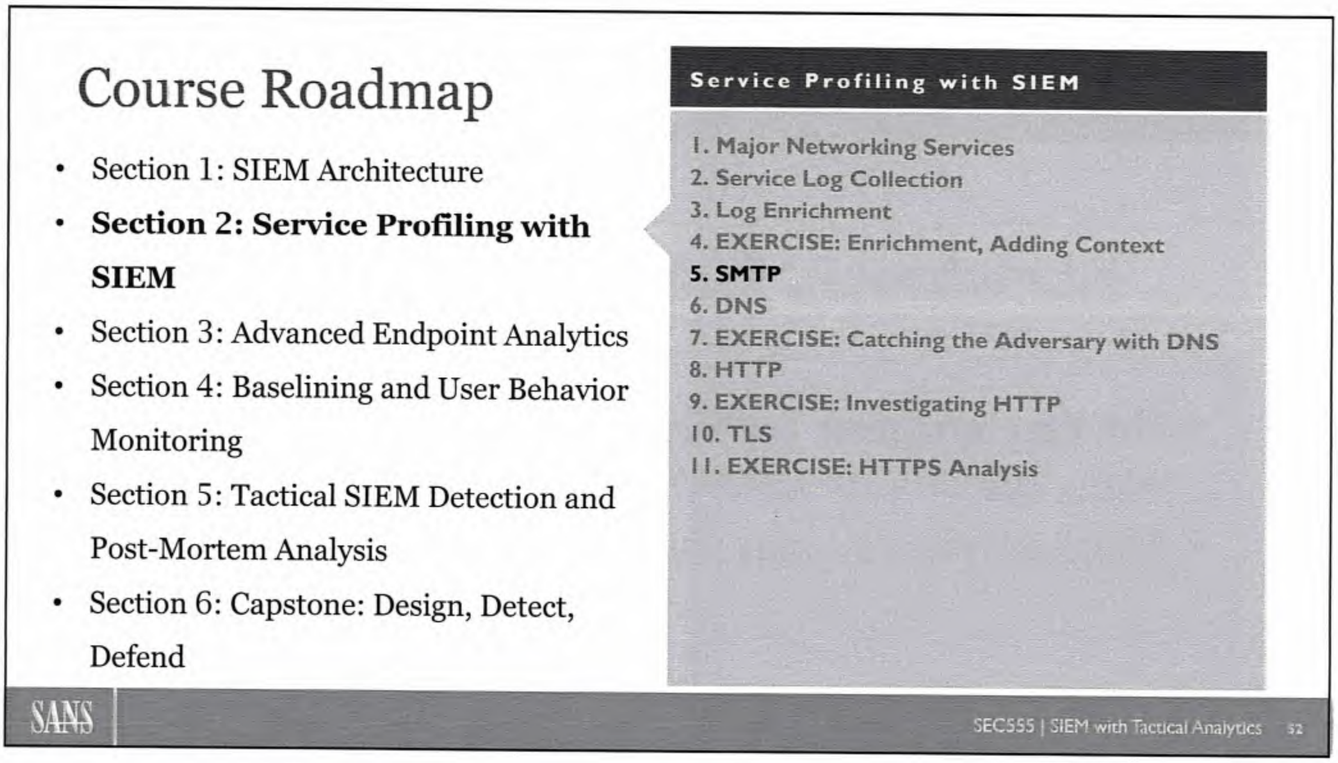
SEC555 Workbook: Enrichment Lab: Adding Context

Enrichment Lab: Adding Context

- The **Enrichment Lab** is in the digital wiki found in your student VM (recommended)
- Alternatively, you may use your Workbook

SANS

This page intentionally left blank.



Course Roadmap

- Section 1: SIEM Architecture
- **Section 2: Service Profiling with SIEM**
- Section 3: Advanced Endpoint Analytics
- Section 4: Baselining and User Behavior Monitoring
- Section 5: Tactical SIEM Detection and Post-Mortem Analysis
- Section 6: Capstone: Design, Detect, Defend

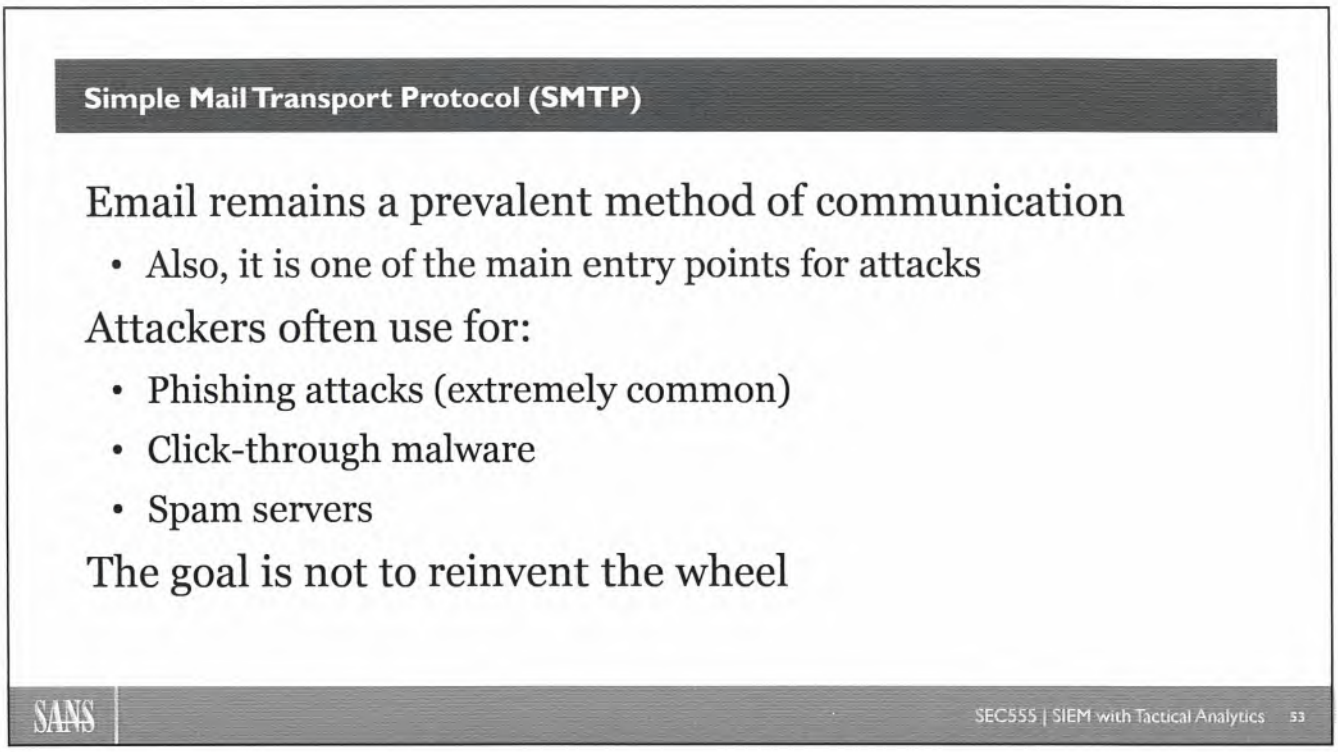
Service Profiling with SIEM

1. Major Networking Services
2. Service Log Collection
3. Log Enrichment
4. EXERCISE: Enrichment, Adding Context
5. **SMTP**
6. DNS
7. EXERCISE: Catching the Adversary with DNS
8. HTTP
9. EXERCISE: Investigating HTTP
10. TLS
11. EXERCISE: HTTPS Analysis

SANS

SEC555 | SIEM with Tactical Analytics 52

This page intentionally left blank.



Simple Mail Transport Protocol (SMTP)

Email remains a prevalent method of communication

- Also, it is one of the main entry points for attacks

Attackers often use for:

- Phishing attacks (extremely common)
- Click-through malware
- Spam servers

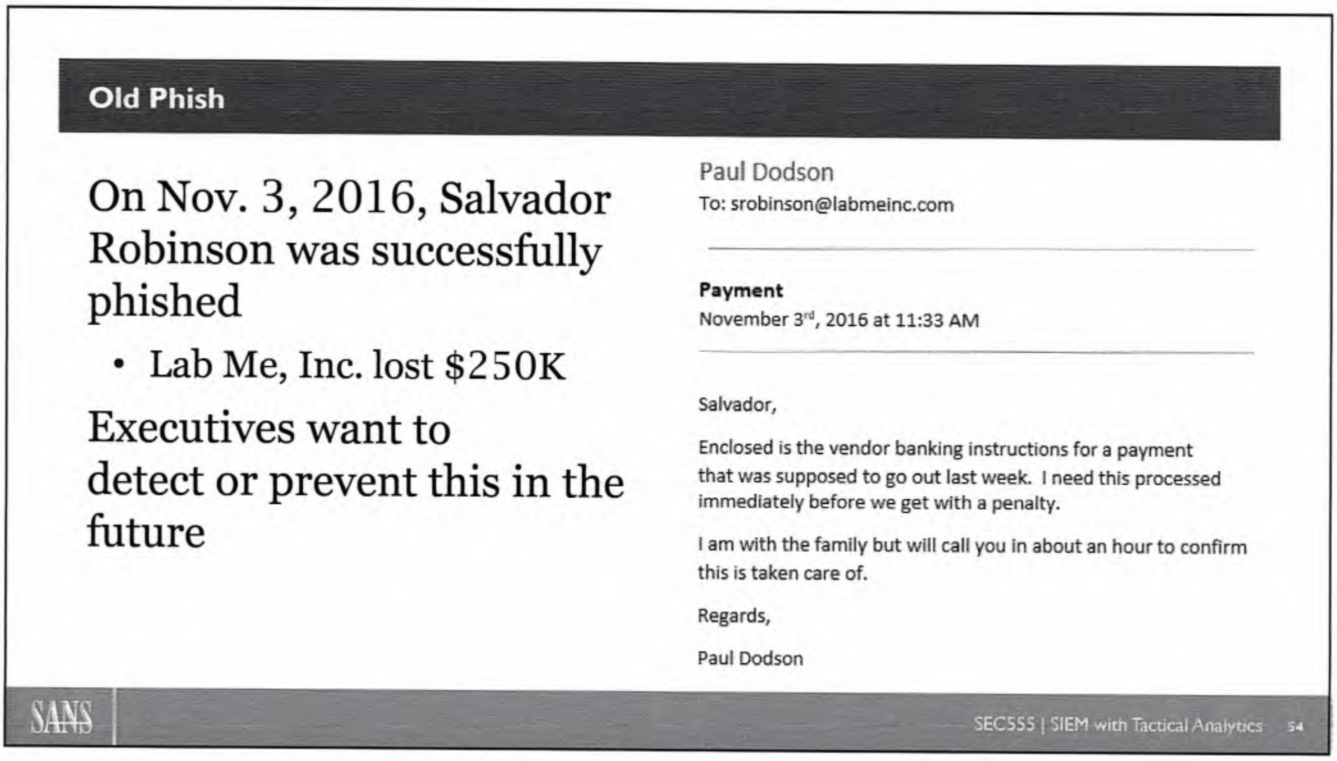
The goal is not to reinvent the wheel

SANS

SEC555 | SIEM with Tactical Analytics 53

Simple Mail Transport Protocol (SMTP)

Email is an extremely common entry point for attackers. Even with today's focus on end-user security awareness and phishing campaigns, email is a common target for automated malware, targeted attacks, and penetration testers.



Old Phish

On Nov. 3, 2016, Salvador Robinson was successfully phished

- Lab Me, Inc. lost $250K

Executives want to detect or prevent this in the future

Paul Dodson
To: srobinson@labmeinc.com

Payment
November 3rd, 2016 at 11:33 AM

Salvador,

Enclosed is the vendor banking instructions for a payment that was supposed to go out last week. I need this processed immediately before we get with a penalty.

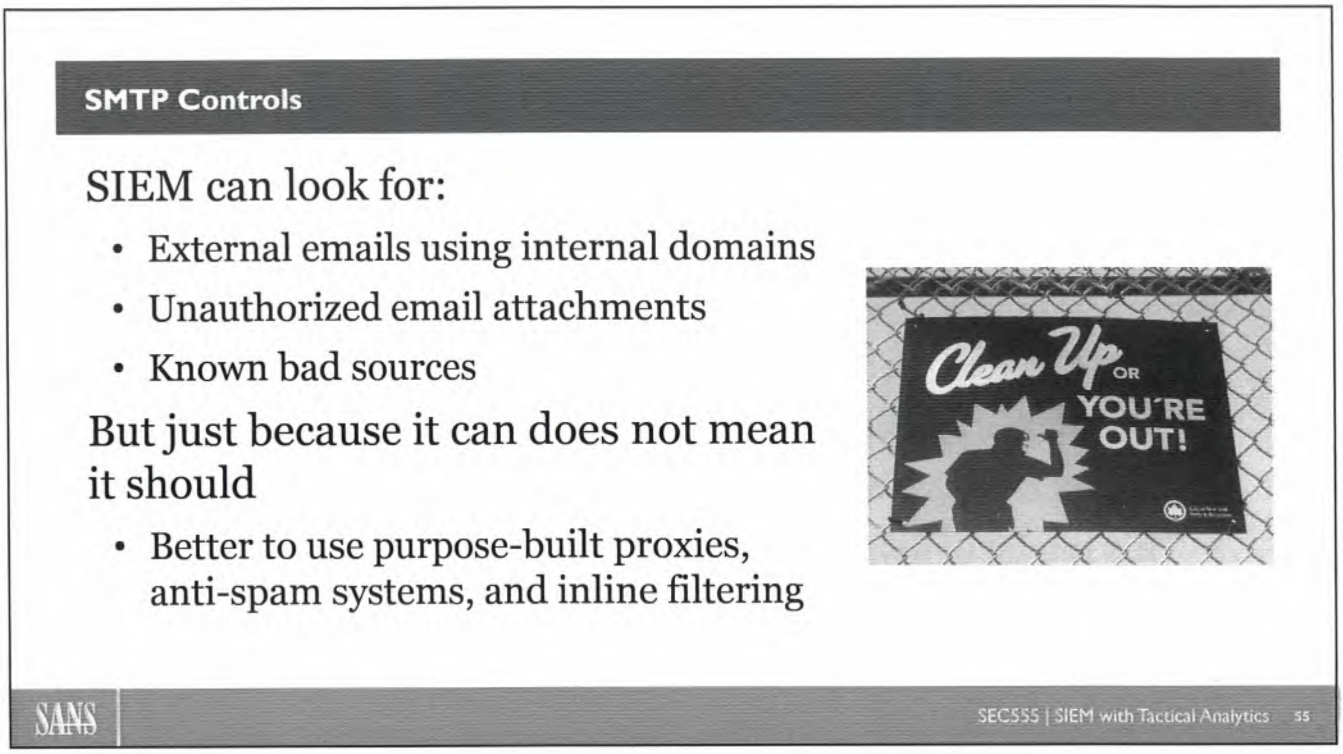
I am with the family but will call you in about an hour to confirm this is taken care of.

Regards,

Paul Dodson

Old Phish

This slide demonstrates a successful phishing attempt against Lab Me, Inc. back on November 3, 2016. This phishing example demonstrates the push for action and the urgency to perform it. It also follows a common trend of acting as a CEO or CFO.



SMTP Controls

SIEM can look for:

- External emails using internal domains
- Unauthorized email attachments
- Known bad sources

But just because it can does not mean it should

- Better to use purpose-built proxies, anti-spam systems, and inline filtering

Clean Up
OR
YOU'RE
OUT!

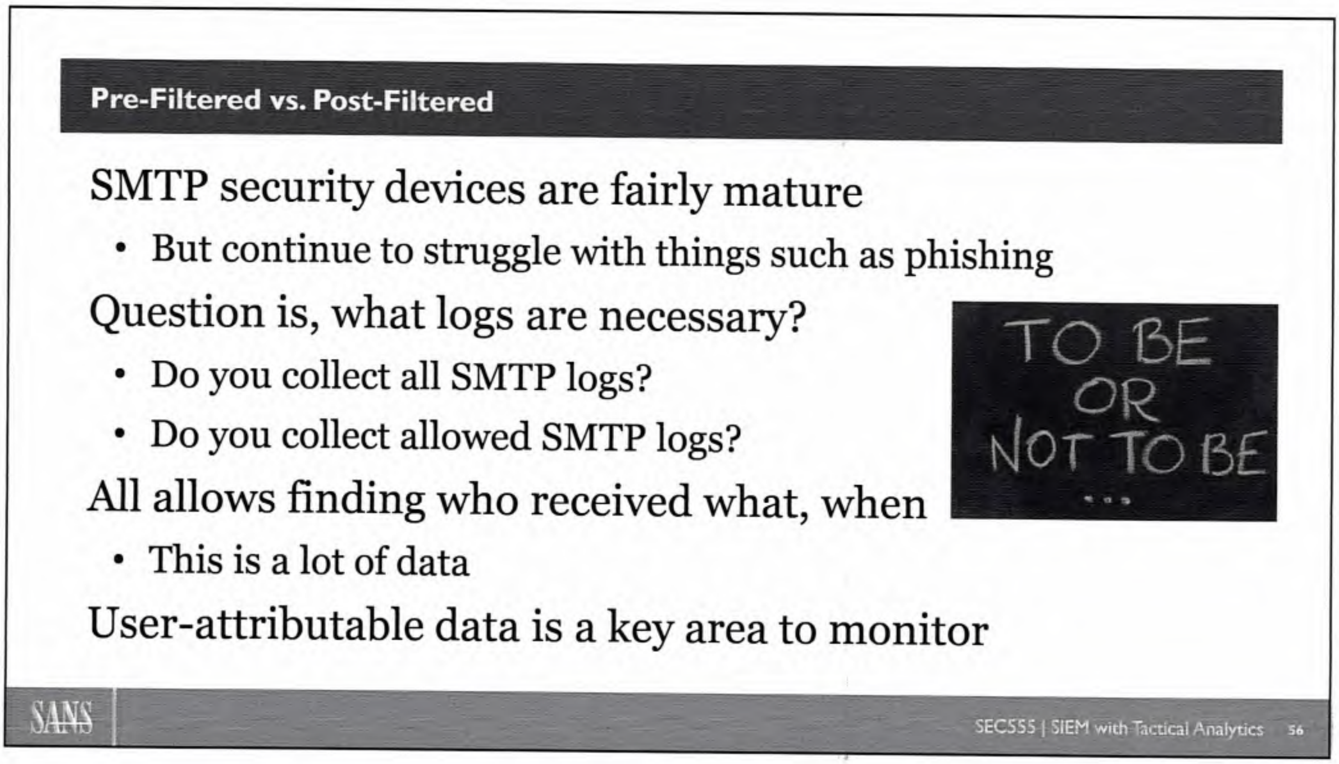
SANS

SEC555 | SIEM with Tactical Analytics 55

SMTP Controls

The SIEM is positioned so it could easily do things such as finding external emails coming in that use internal domains. However, there are better technologies that should be doing this. As a general hygiene rule, they should be used. If you want extra visibility, then go ahead and collect the logs and build out reports.

The point is, do not design your SIEM to do things that are better positioned elsewhere. For example, authorized SMTP servers should be the only thing allowed to send email outbound. Even then, they should ideally funnel through an SMTP proxy. This means detection of unauthorized email servers can be seen by monitoring firewall deny rules. On the flip side, you could use flow data, Zeek logs, etc., to monitor for unauthorized SMTP traffic. However, this gives you detection instead of prevention plus detection. If you are going to spend the time and effort, it may make more sense to spend the time to put in the proper firewall rules and then build your report using firewall logs.



Pre-Filtered vs. Post-Filtered

SMTP security devices are fairly mature

- But continue to struggle with things such as phishing

Question is, what logs are necessary?

- Do you collect all SMTP logs?
- Do you collect allowed SMTP logs?

All allows finding who received what, when

- This is a lot of data

User-attributable data is a key area to monitor

TO BE
OR
NOT TO BE
...

SANS

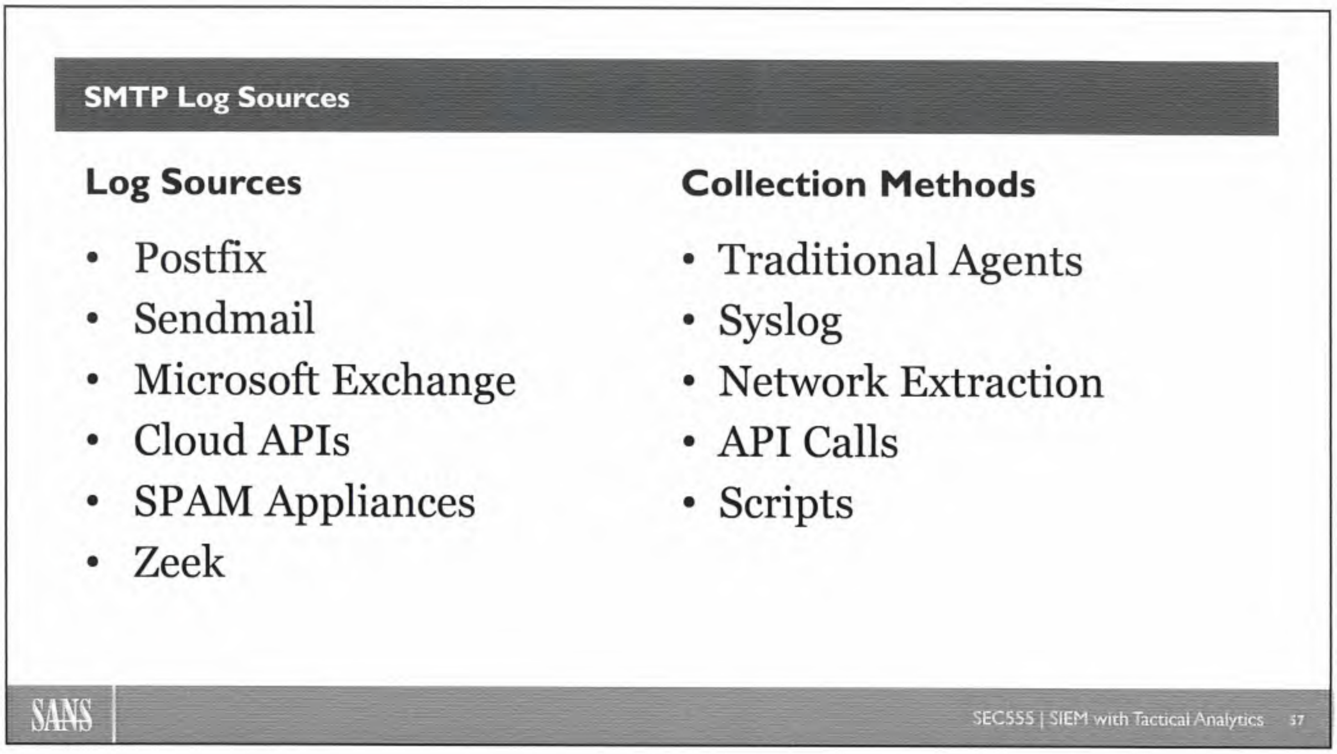
SEC555 | SIEM with Tactical Analytics 56

Pre-Filtered vs. Post-Filtered

A question you probably need to ask yourself is should logs be collected pre-filtered or post-filtered? Regarding SMTP, this would mean the collection of all logs versus only collecting SMTP logs for emails that made it through any SMTP filtering such as spam appliances. The answer depends on timing and use case.

If you are focusing on tactical enhancements to spam filters, then you only need post-filtered logs. If you want to be able to report on all emails including spam emails, then you need everything. The question then becomes centered on the cost to generate the report versus the value provided by the report.

In this author's opinion, the failure to ask this type of question is what causes some SIEM deployments to fail. Executives catch wind of how much money is going into the SIEM and pull the plug. The tendency to collect everything for reports slowly builds up and starts to nickel and dime your SIEM. Often, the data collected for a "must-have report" could have been generated directly from the device the logs came from in the first place (such as a spam appliance).



SMTP Log Sources

Log Sources

- Postfix
- Sendmail
- Microsoft Exchange
- Cloud APIs
- SPAM Appliances
- Zeek

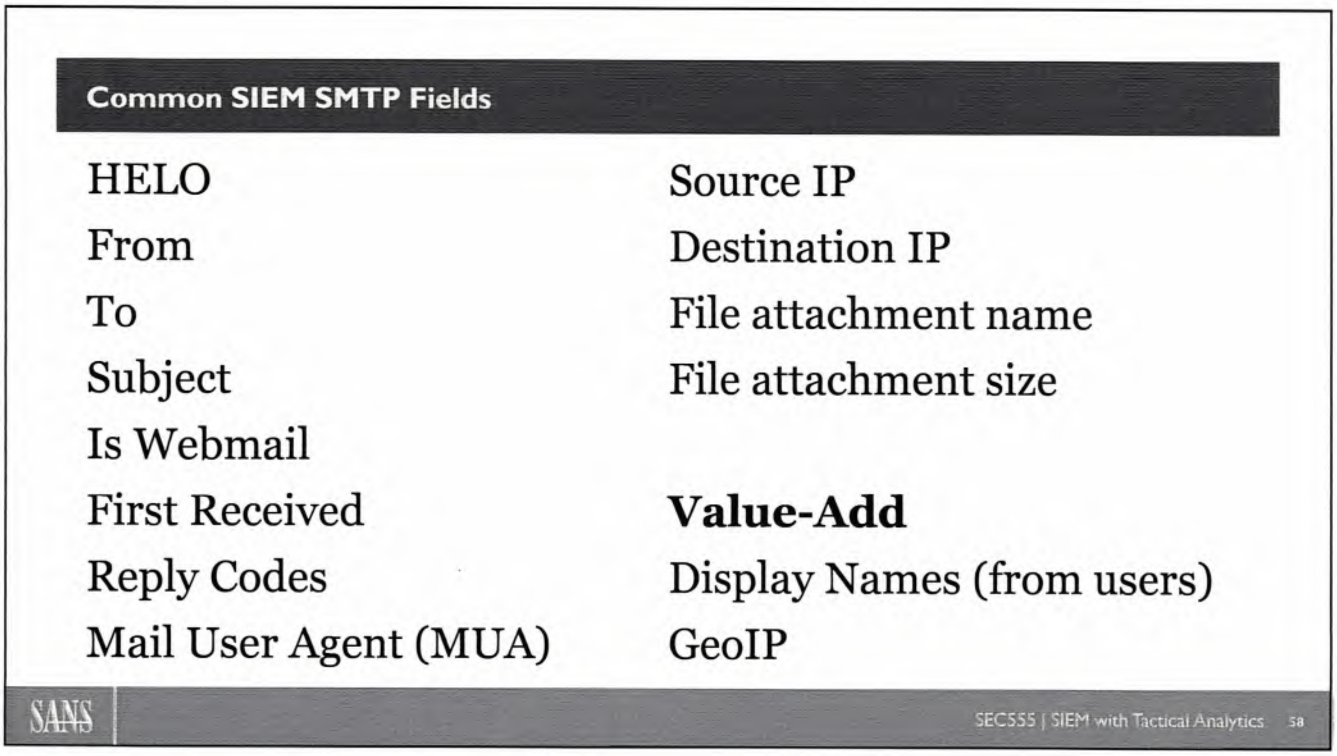
Collection Methods

- Traditional Agents
- Syslog
- Network Extraction
- API Calls
- Scripts

SMTP Log Sources

Of all the different types of logs, SMTP is probably one of the sources most tailored to an organization. This is because there are so many different email servers, email proxies, and other email security devices that an organization really should commit to which source it wants for mail. Unfortunately, many organizations decide to collect from all sources, which nets them lots of duplicate data. If you want to collect logs from your email server and a spam appliance, you should have a reason for pulling from both. It's possible that each has fields or values only available in one, but that is likely not the case.

Because of the wide range of sources logs reflected in this module, we'll reflect Zeek SMTP unless otherwise noted.



Common SIEM SMTP Fields

HELO

From

To

Subject

Is Webmail

First Received

Reply Codes

Mail User Agent (MUA)

Source IP

Destination IP

File attachment name

File attachment size

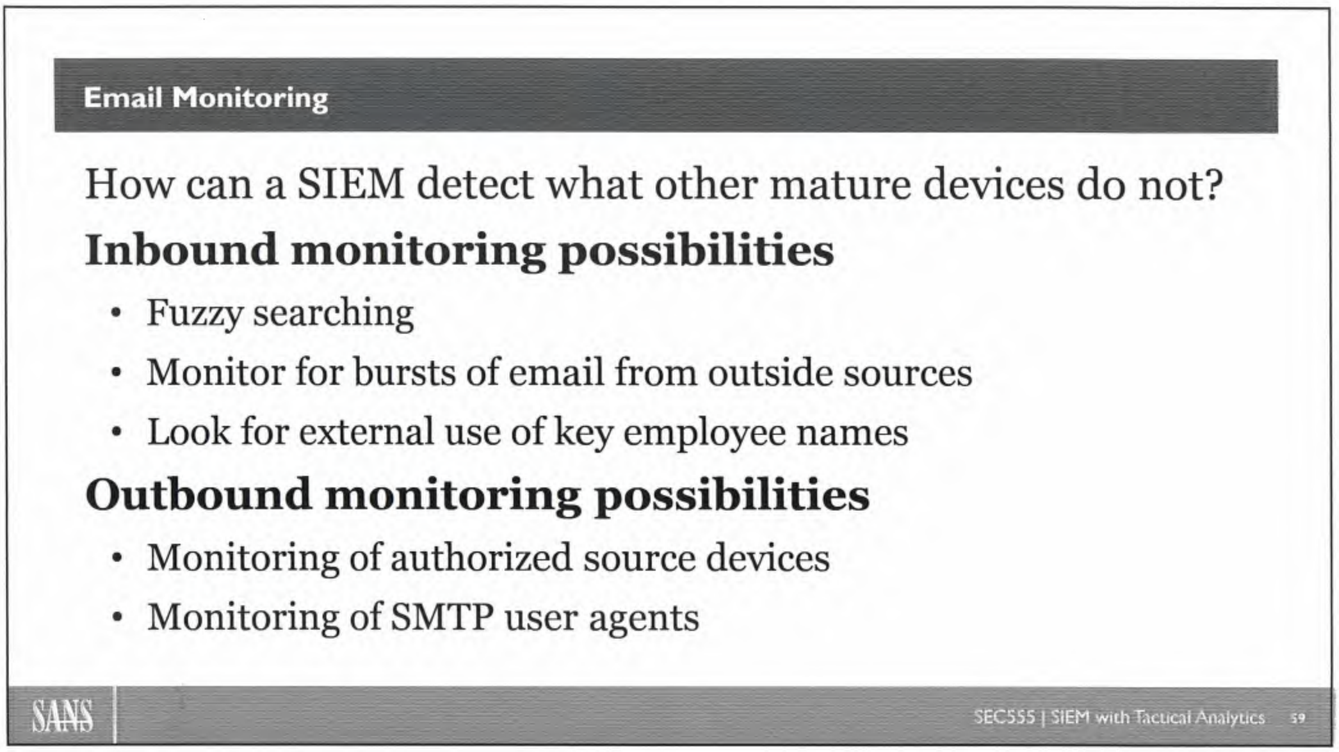
Value-Add

Display Names (from users)

GeoIP

Common SIEM SMTP Fields

This slide demonstrates some of the more common SMTP fields. Different sources log different information, and the field names are not always called the same thing.



Email Monitoring

How can a SIEM detect what other mature devices do not?

Inbound monitoring possibilities

- Fuzzy searching
- Monitor for bursts of email from outside sources
- Look for external use of key employee names

Outbound monitoring possibilities

- Monitoring of authorized source devices
- Monitoring of SMTP user agents

Email Monitoring

With so much technology pre-built around SMTP, how can a SIEM possibly help? If you stop to think about it, most of the existing technologies are designed to be prevention devices primarily. On top of that, they must be designed in a way that works across multiple environments. Thus, they are not tailored to your environment.

Some of the strongest capabilities lay in the fact that analysts can tune systems according to their environment.

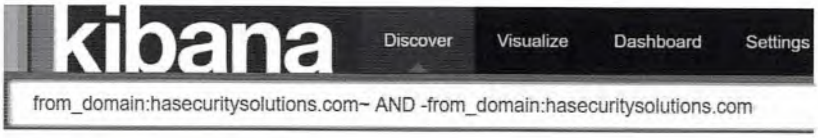
Fuzzy Phishing

Many SIEM techniques use insider information

- Such as fuzzy phishing searches

Take legitimate company domains and look for variants

- Extremely effective against phishing domains
- Best used in combination with email alerts or scripts
- Great for targeted attacks

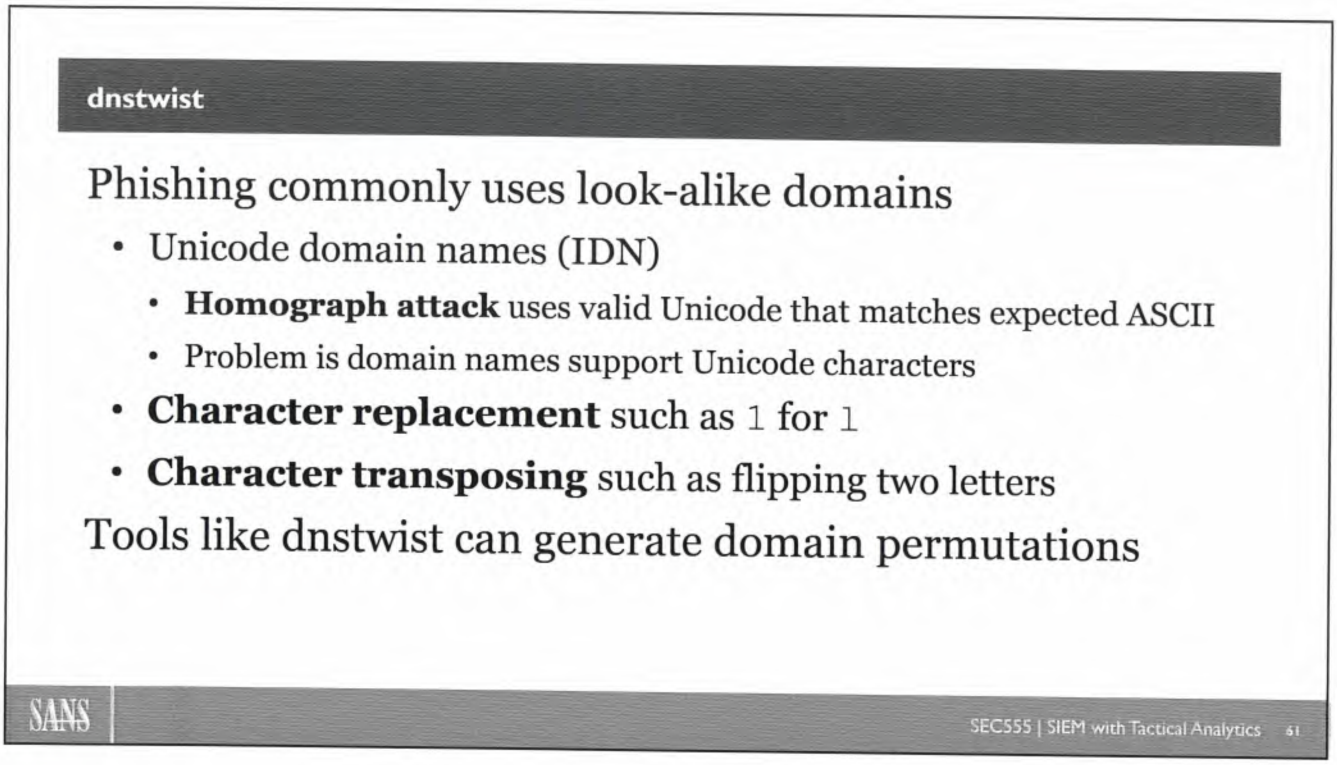


```
from_domain:hasecuritysolutions.com~ AND -from_domain:hasecuritysolutions.com
```

Fuzzy Phishing

The difference between a good analyst and a great analyst is a great analyst tries to understand his or her organization and uses this information to find and thwart the bad guys. By establishing this knowledge, an analyst can then turn around and use it as a weapon. For example, fuzzy phishing uses a list of internal domains to look for phishing attempts against it.

This technique works extremely well, but there is still a problem. The SIEM is excellent at detecting phishing domains, but phishing attacks happen quickly. However, all is not lost. The SIEM can be used to react. The dangerous thing would be to start blocking things, although it is something to consider. On the flip side, the SIEM could do things such as send out an email alert warning of suspicious activity.



dnstwist

Phishing commonly uses look-alike domains

- Unicode domain names (IDN)
  - **Homograph attack** uses valid Unicode that matches expected ASCII
  - Problem is domain names support Unicode characters
- **Character replacement** such as `1` for `l`
- **Character transposing** such as flipping two letters

Tools like dnstwist can generate domain permutations

SANS

SEC555 | SIEM with Tactical Analytics 61

dnstwist

The fuzzy phishing technique on the previous slide helps identify domains that are similar to but not exactly a given domain. However, using Levenshtein distance searches may not be as accurate or as comprehensive, as the default often only looks for a difference of up to two characters. The problem is phishing attacks can combine multiple techniques such as homograph attacks, character replacement attacks, and character transposing in which a phishing domain may have more than two characters changed. As an example, a homograph attack can use valid Unicode character sets such as Cyrillic, Greek, or Armenian to completely change every character in a domain with the domain still looking the same under a microscope.

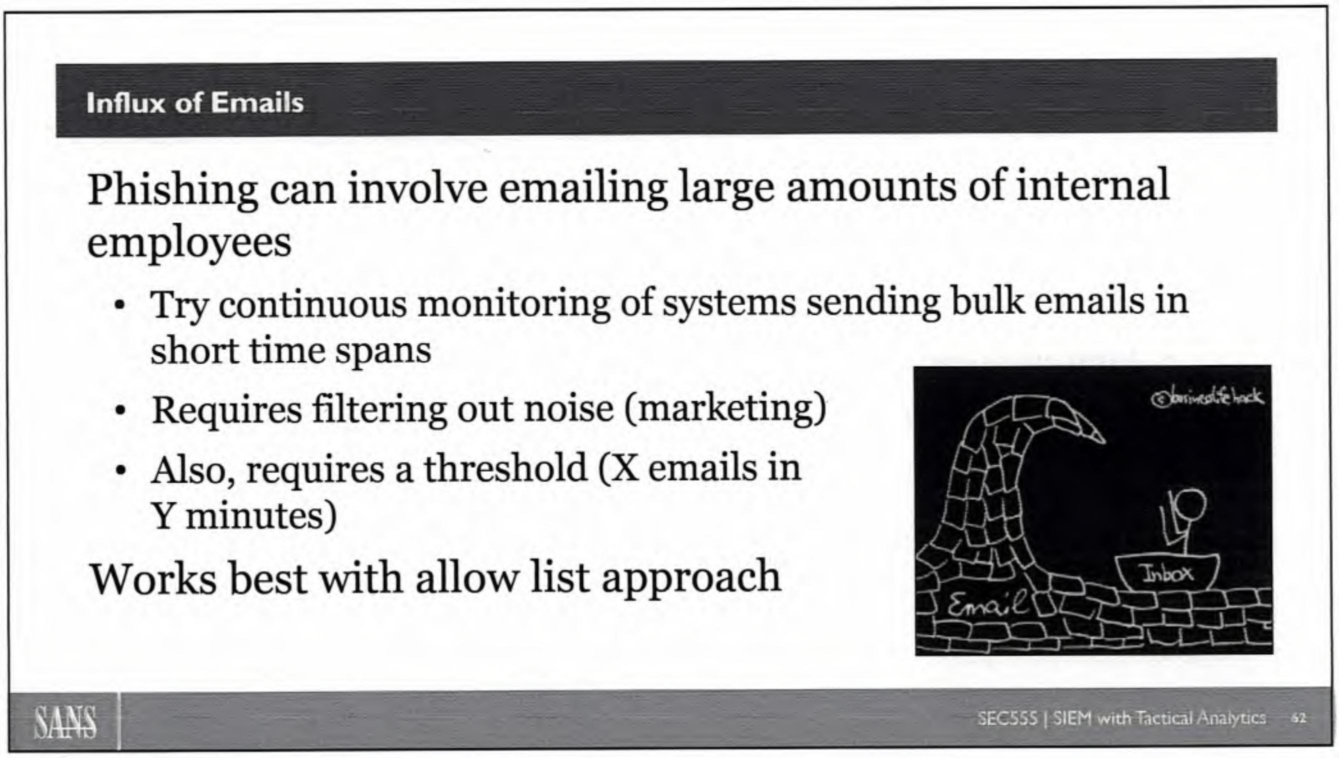
A different approach to detecting this is using a domain permutation engine to identify all the possible and valid permutations of a domain. One tool that calculates domain permutations is dnstwist[1]. Organizations can use dnstwist to generate possible permutations of a domain that may be used for phishing. The list of permutations can be fed to a SIEM as a lookup table.

Another option is the website implementation of dnstwist that does not require installation and can export the results as either json or csv.

References:

https://github.com/elceef/dnstwist, https://sec555.com/4t

https://dnstwist.it/, http://sec555.com/111



Influx of Emails

Phishing can involve emailing large amounts of internal employees

- Try continuous monitoring of systems sending bulk emails in short time spans
- Requires filtering out noise (marketing)
- Also, requires a threshold (X emails in Y minutes)

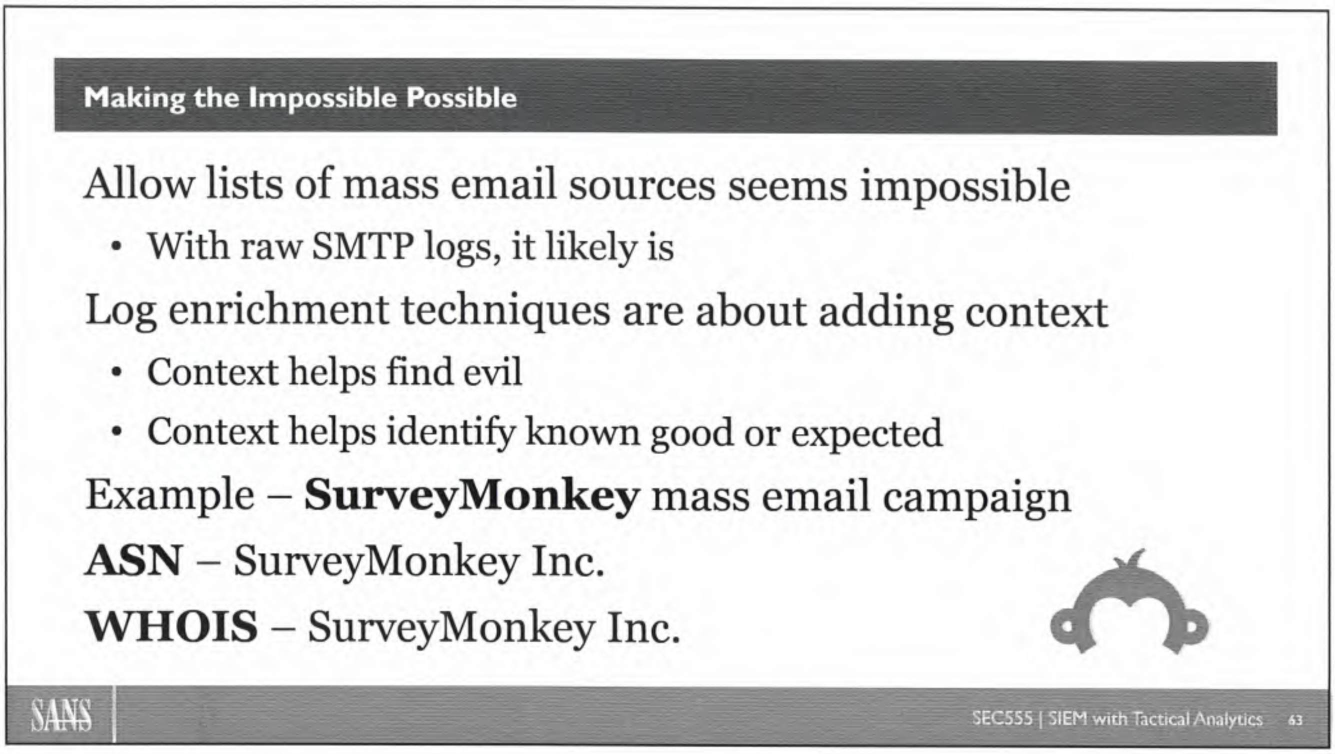
Works best with allow list approach

©businesslifehack
Inbox
Email
SANS

Influx of Emails

While many phishing attacks may be targeted spear phishing attempts, it is common to have a phishing email target a significant percentage of an organization. This method of attack has a higher chance of success even with a weakly-designed phishing message. Instead of crafting a targeted email against a handful of people, a general email can be sent to thousands or tens of thousands, and it only takes one person to click a link or open an attachment for the attacker to win.

Because of this, it is possible to monitor for large phishing campaigns by looking for bulk emails by source. This technique is guaranteed to have false positives. Marketing or partner services are likely to generate bulk emails to your organization. Therefore, the only way this method works is applying it with a allow list approach. Any authorized system that hits this list gets added to an allow list, and any source that is not authorized gets investigated.



Making the Impossible Possible

Allow lists of mass email sources seems impossible

- With raw SMTP logs, it likely is

Log enrichment techniques are about adding context

- Context helps find evil
- Context helps identify known good or expected

Example – **SurveyMonkey** mass email campaign

ASN – SurveyMonkey Inc.

WHOIS – SurveyMonkey Inc.

SANS

SEC555 | SIEM with Tactical Analytics 63

Making the Impossible Possible

Organizations often do not monitor for certain conditions such as who is sending mass email campaigns because it seems like a lesson in futility. In truth, many detection techniques are because raw logs do not provide sufficient information to filter out noise. Log enrichment adds context so that organizations can detect things they normally would not.

For example, if an organization appending geo information or WHOIS information about an email sender in SMTP logs, then filtering out known email companies is possible. In the slide, SurveyMonkey is provided as an example. An SMTP log would normally have the IP addresses of the sender, and that can be used to pull in information that the entity is actually SurveyMonkey. Therefore, an organization can use enrichment context to filter out mass marketing firms and partners.

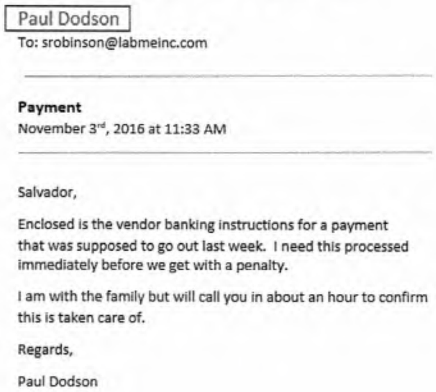
Hierarchical Spear Phishing

Many times, spear phishing impersonates executive staff to pressure a response

- Example: Fake CFO sends an email to accountant to transfer money

Possible to use a list of names to look for phishing attempts

- Look for display names of key staff and tag, alert, or trigger action



Paul Dodson
To: srobinson@labmeinc.com

Payment
November 3rd, 2016 at 11:33 AM

Salvador,

Enclosed is the vendor banking instructions for a payment that was supposed to go out last week. I need this processed immediately before we get with a penalty.

I am with the family but will call you in about an hour to confirm this is taken care of.

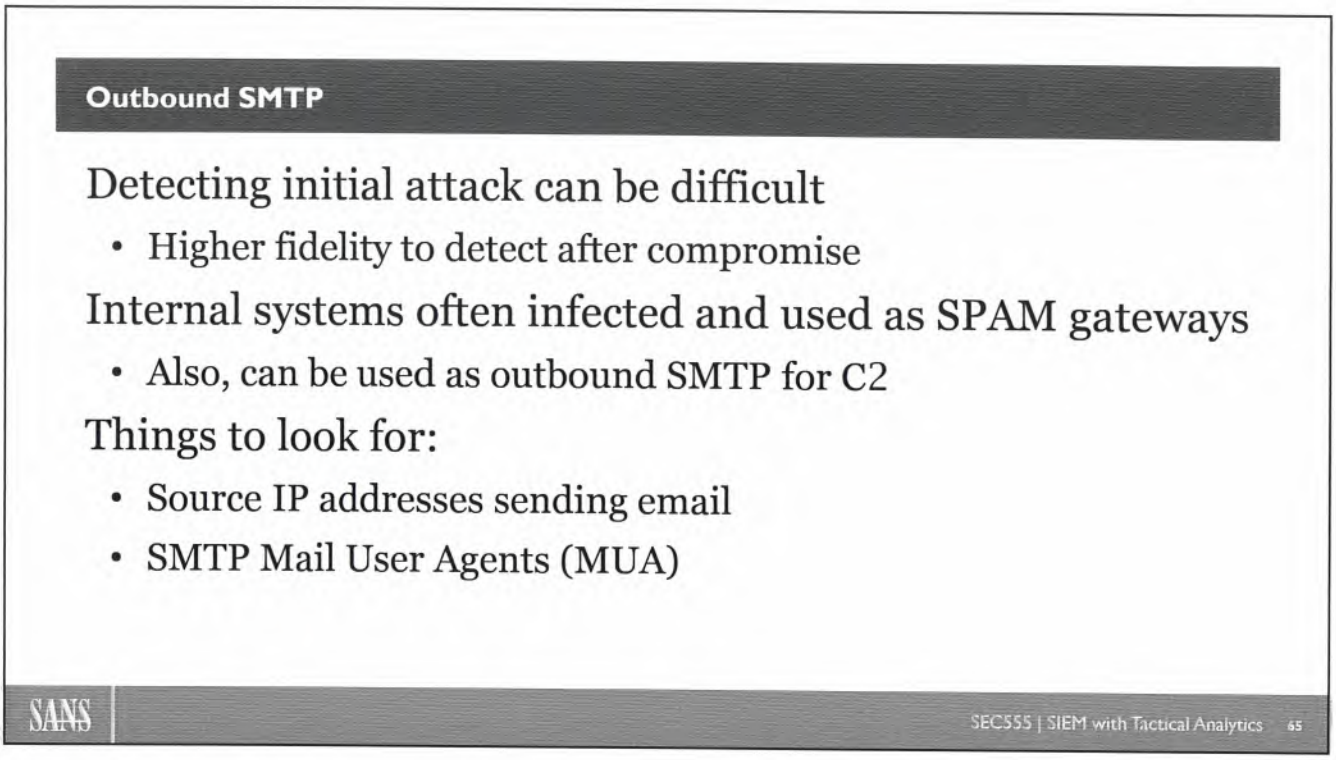
Regards,

Paul Dodson

Hierarchical Spear Phishing

Because many phishing attempts act as key executives, also referred to as whaling, it makes sense to monitor the display names for incoming emails. In a previous phishing email, the name of Paul Dodson, the CFO of Lab Me, Inc., was used against Salvador Robinson. By monitoring for incoming emails with a display name of Paul Dodson, but not coming from an email address associated with labmeinc.com, this could have been detected.

Occasionally, you may have a name collision, but likely this would not trigger often. To apply this technique, take the names of your executive team and start monitoring for inbound emails that use their names.



Outbound SMTP

Detecting initial attack can be difficult

- Higher fidelity to detect after compromise

Internal systems often infected and used as SPAM gateways

- Also, can be used as outbound SMTP for C2

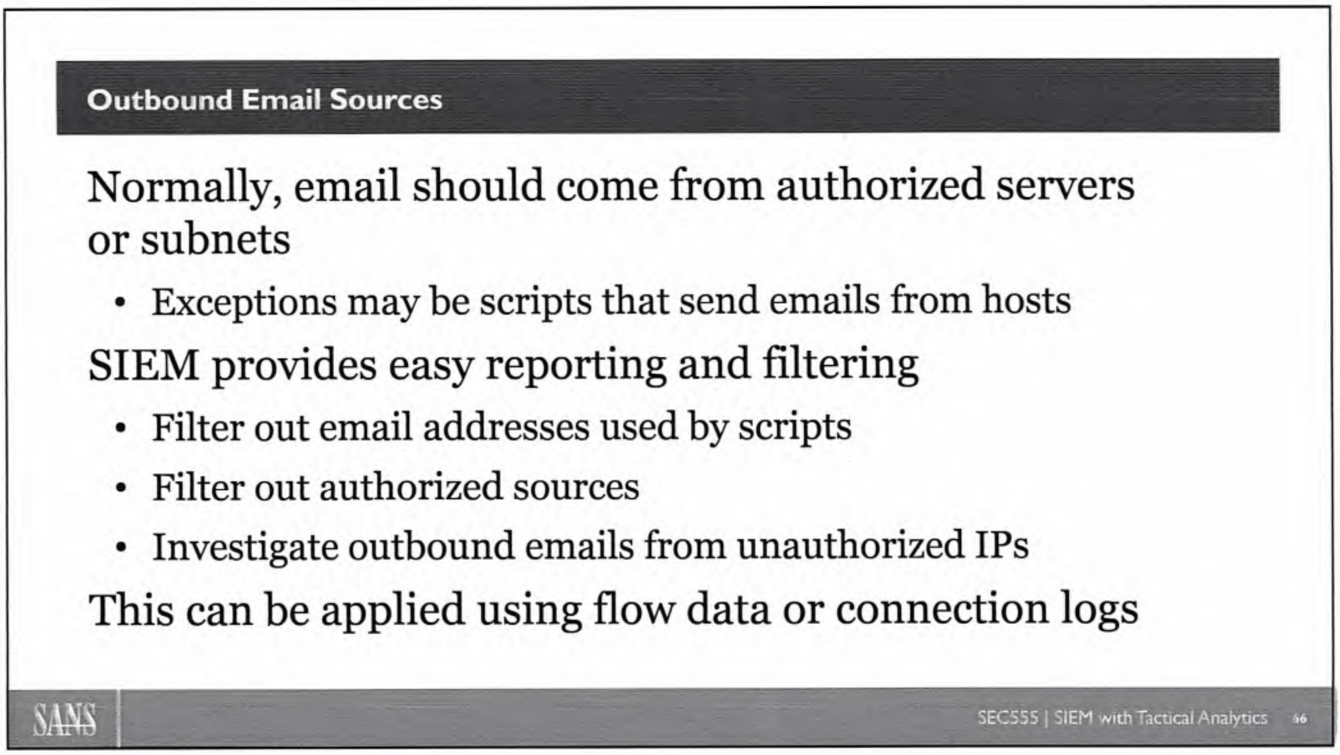
Things to look for:

- Source IP addresses sending email
- SMTP Mail User Agents (MUA)

Outbound SMTP

Exploitation can involve exploits that have never been seen before or that have been modified to bypass prevention controls. Because of this, it is often much easier to detect and attack post-compromise. In the case of SMTP, there are multiple historical cases where infected systems were set up as SPAM gateways or used SMTP for outbound command and control.

Monitoring of which systems are sending outbound email should be considered to detect this kind of behavior.



Outbound Email Sources

Normally, email should come from authorized servers or subnets

- Exceptions may be scripts that send emails from hosts

SIEM provides easy reporting and filtering

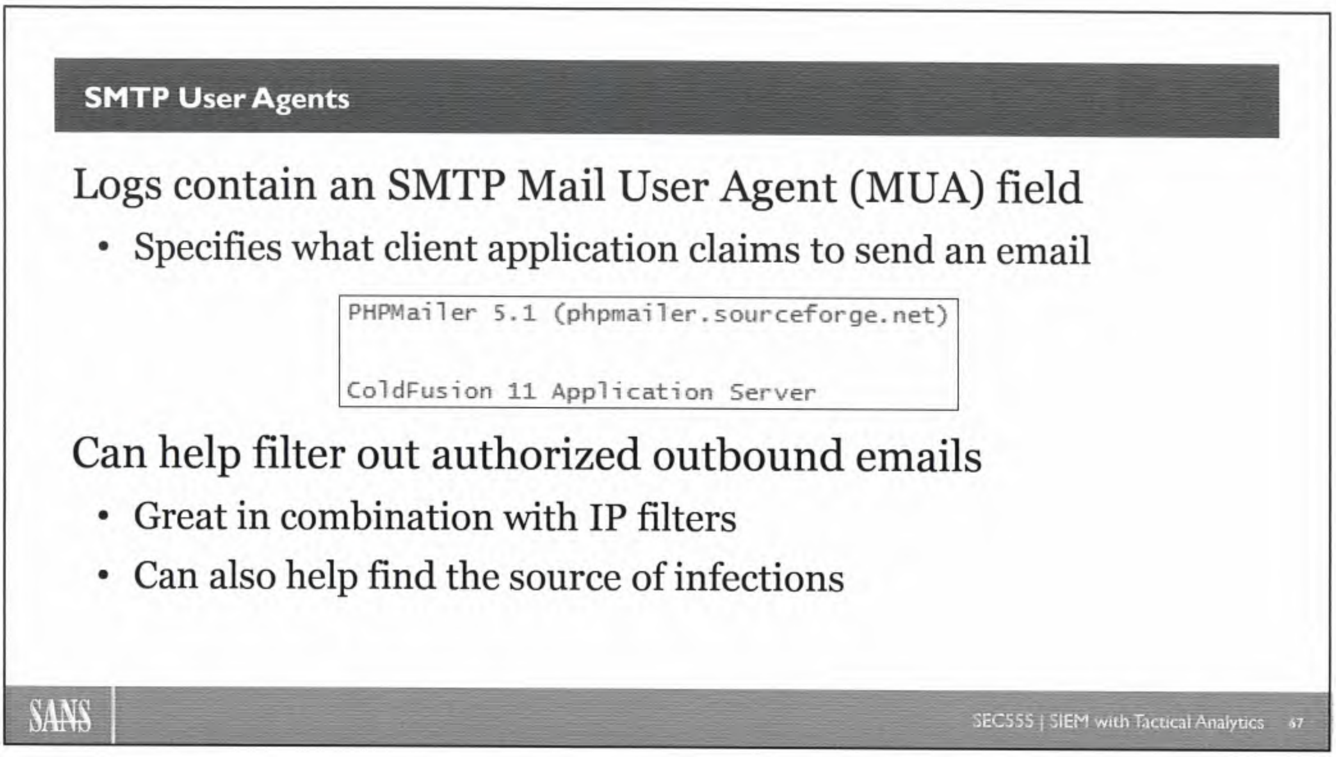
- Filter out email addresses used by scripts
- Filter out authorized sources
- Investigate outbound emails from unauthorized IPs

This can be applied using flow data or connection logs

Outbound Email Sources

Within a given organization, there should be a finite list of authorized email sources. Normally, when a desktop sends an email to someone, the email is actually generated and sent from an email server, not the desktop. Therefore, it is odd if SMTP traffic starts to occur in places it should not be.

This is another allow list approach to looking for email sources where they are not expected. Because most organizations have a limited amount of email servers and new ones are not regularly stood up, this is fairly low maintenance outside the initial setup.



SMTP User Agents

Logs contain an SMTP Mail User Agent (MUA) field

- Specifies what client application claims to send an email

```
PHPMailer 5.1 (phpmailer.sourceforge.net)

ColdFusion 11 Application Server
```

Can help filter out authorized outbound emails

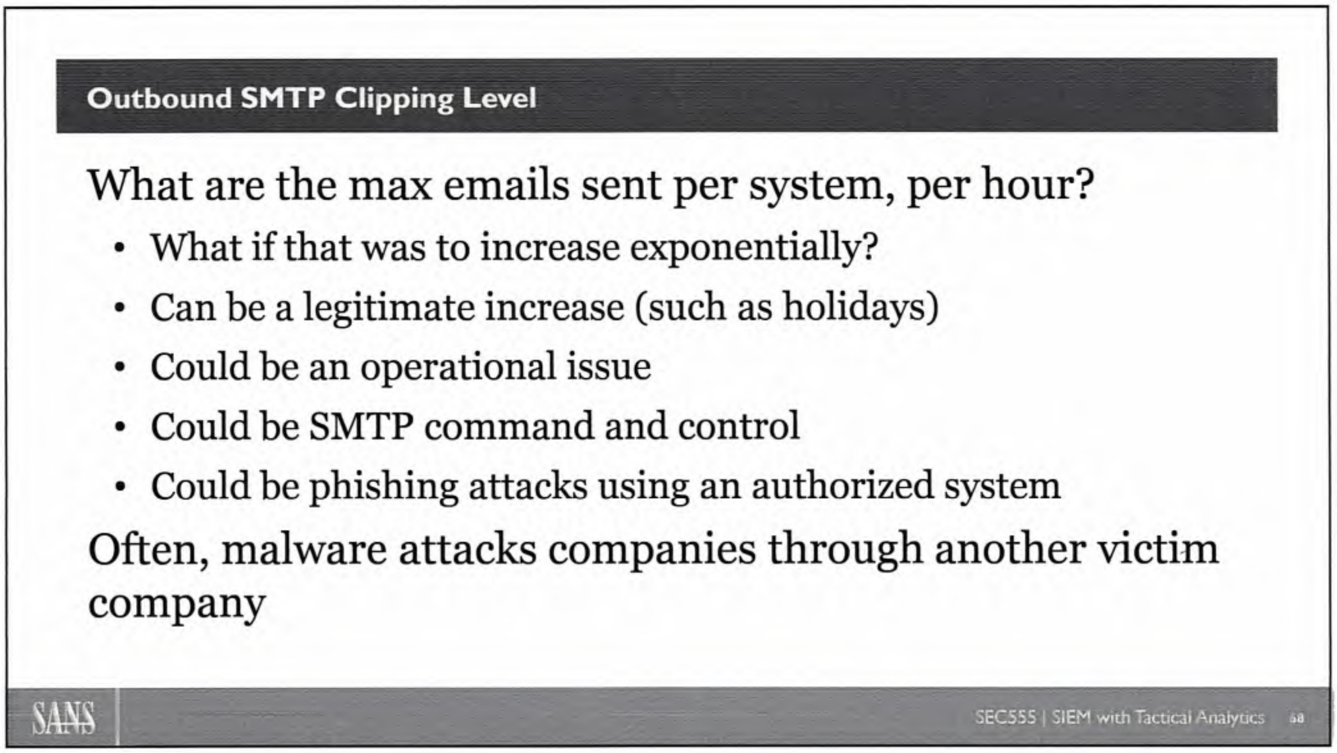
- Great in combination with IP filters
- Can also help find the source of infections

SMTP User Agents

An SMTP user agent describes the application being used to send email, such as PHPMailer 5.1 or ColdFusion 11 Application Server. The user agent can be used to help filter out authorized outbound emails or can be used as a separate internal allow list approach.

User agents can also be helpful if an authorized system is compromised and starts sending emails to a different user agent. For instance, if you had an authorized web server that normally sent emails using PHPMailer but then starts sending emails using Python, something is wrong.

The SMTP Mail User Agent is not required. Also, the contents can be set to anything. This means malware can mask as a legitimate email client.



Outbound SMTP Clipping Level

What are the max emails sent per system, per hour?

- What if that was to increase exponentially?
- Can be a legitimate increase (such as holidays)
- Could be an operational issue
- Could be SMTP command and control
- Could be phishing attacks using an authorized system

Often, malware attacks companies through another victim company

SANS
SEC555 | SIEM with Tactical Analytics 68

Outbound SMTP Clipping Level

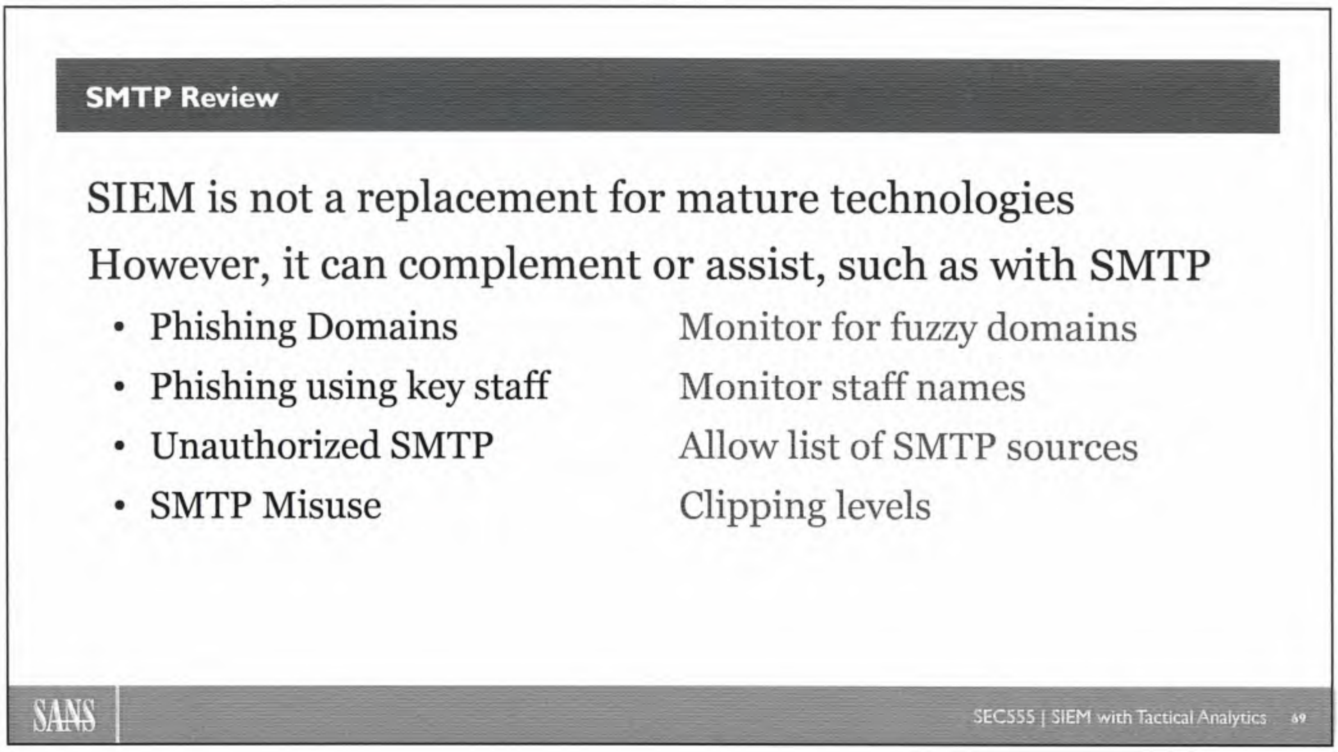
Knowing normal makes knowing abnormal easy. The hard part is knowing what is normal. Most organizations, if answering honestly, have no clue how many emails they send per hour, let alone how many per authorized system. Finding this amount and establishing a clipping level allows for finding abnormal spikes or unauthorized use. This is also an area where machine learning can help automate. However, this is something that is easy to implement by simply setting a high clipping level or using ElastAlert to monitor for excessive spikes.

This author has found this particular technique works in identifying a compromised DMZ web server that was authorized to send email. This system, once compromised, attempted to send massive amounts of phishing emails against a different target organization. An example where SMTP has been used for command and control is APT28.

References:

https://www.fireeye.com/blog/threat-research/2014/10/apt28-a-window-into-russias-cyber-espionage-operations.html

https://sec555.com/4u



SMTP Review

SIEM is not a replacement for mature technologies

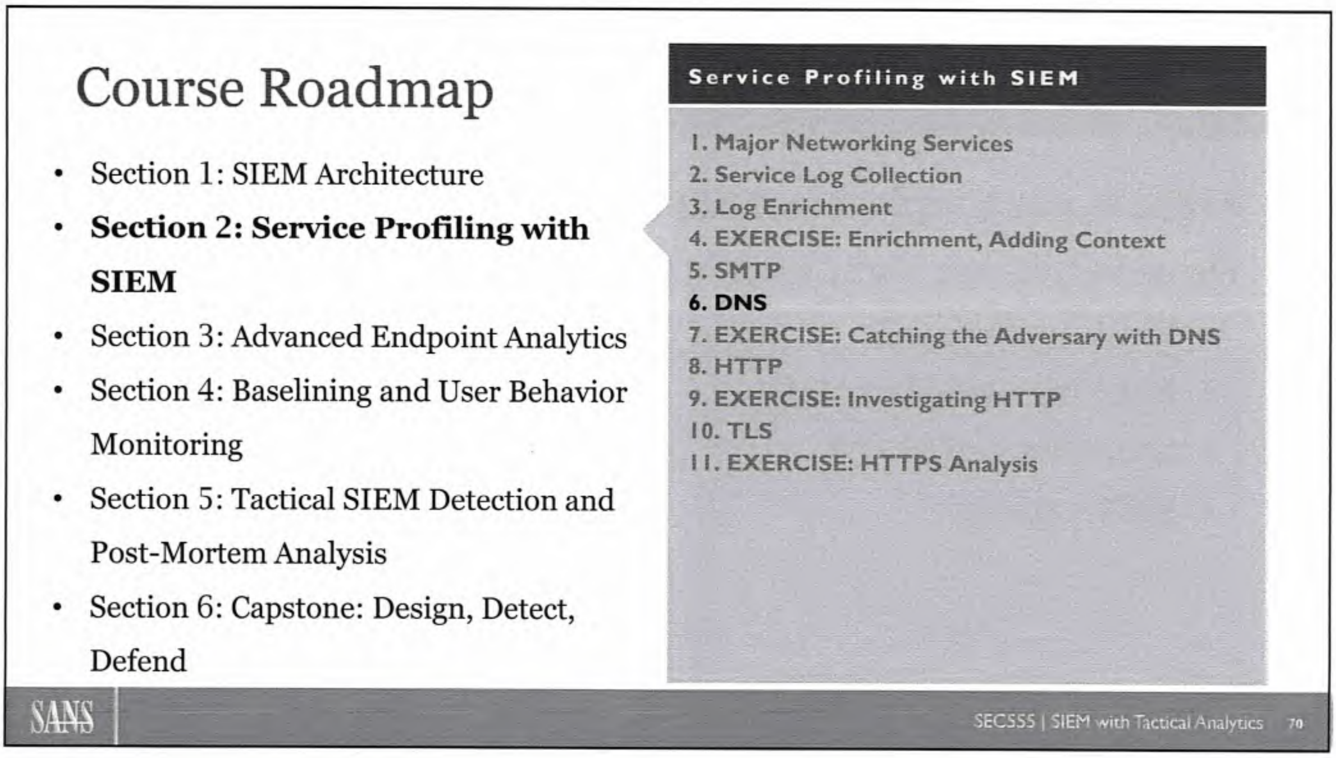
However, it can complement or assist, such as with SMTP

• Phishing Domains	Monitor for fuzzy domains
• Phishing using key staff	Monitor staff names
• Unauthorized SMTP	Allow list of SMTP sources
• SMTP Misuse	Clipping levels

SANS

SMTP Review

SMTP is a technology that has been around a long time. As a result, there are lots of security products that are more mature and more purpose-built around SMTP. However, there are still areas where a SIEM can assist these technologies to find things they simply cannot.



Course Roadmap

- Section 1: SIEM Architecture
- **Section 2: Service Profiling with SIEM**
- Section 3: Advanced Endpoint Analytics
- Section 4: Baselining and User Behavior Monitoring
- Section 5: Tactical SIEM Detection and Post-Mortem Analysis
- Section 6: Capstone: Design, Detect, Defend

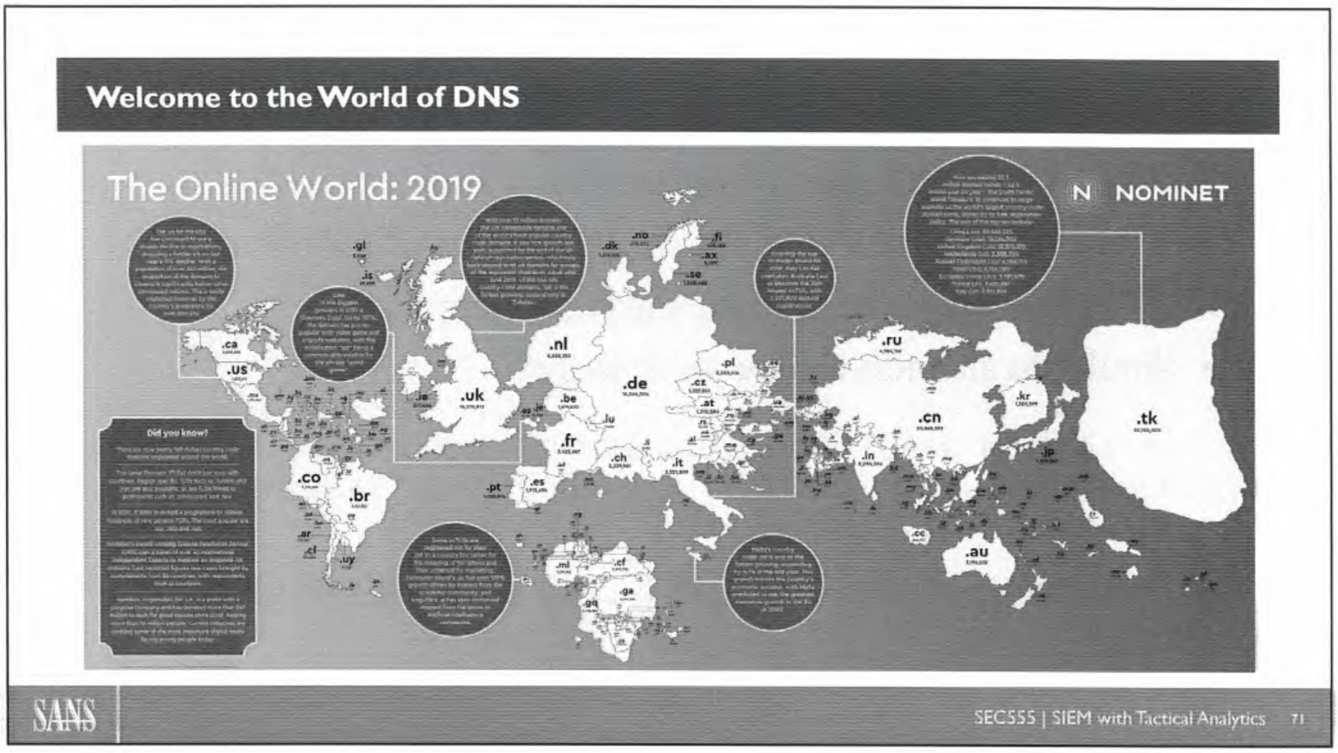
Service Profiling with SIEM

1. Major Networking Services
2. Service Log Collection
3. Log Enrichment
4. EXERCISE: Enrichment, Adding Context
5. SMTP
6. **DNS**
7. EXERCISE: Catching the Adversary with DNS
8. HTTP
9. EXERCISE: Investigating HTTP
10. TLS
11. EXERCISE: HTTPS Analysis

SANS

SEC555 | SIEM with Tactical Analytics 70

This page intentionally left blank.



Welcome to the World of DNS

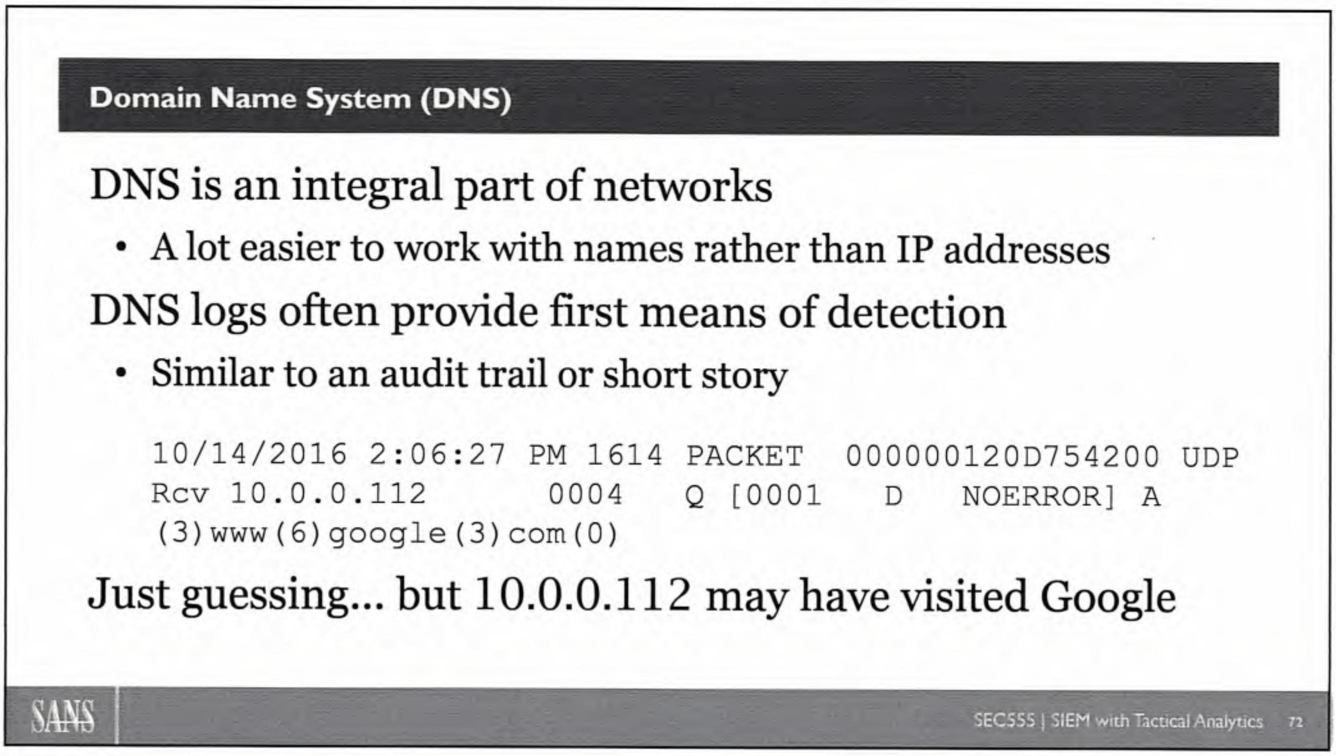
SANS

Welcome to the World of DNS

Nominet provides this world map. In this case, Nominet is visually representing the number of top-level domains (TLDs) by country code. In this map, tk is the largest country which maps to Tokelau. Tokelau is part of the island territory of New Zealand. Tokelau does not maintain their DNS nameservers. Instead, it is outsourced to another country that hosts .tk as a free domain service at dot.tk. Anyone with a valid email address can go to dot.tk and register for a free DNS name. Because of the ease in registering free DNS domains, .tk has become a cesspool of malicious domains. Malware authors love it as it can be fully automated even in conjunction with domain generating algorithms (DGAs).

Reference:

https://www.nominet.uk/the-shifting-world-of-country-codes/, http://sec555.com/112



Domain Name System (DNS)

DNS is an integral part of networks

- A lot easier to work with names rather than IP addresses

DNS logs often provide first means of detection

- Similar to an audit trail or short story

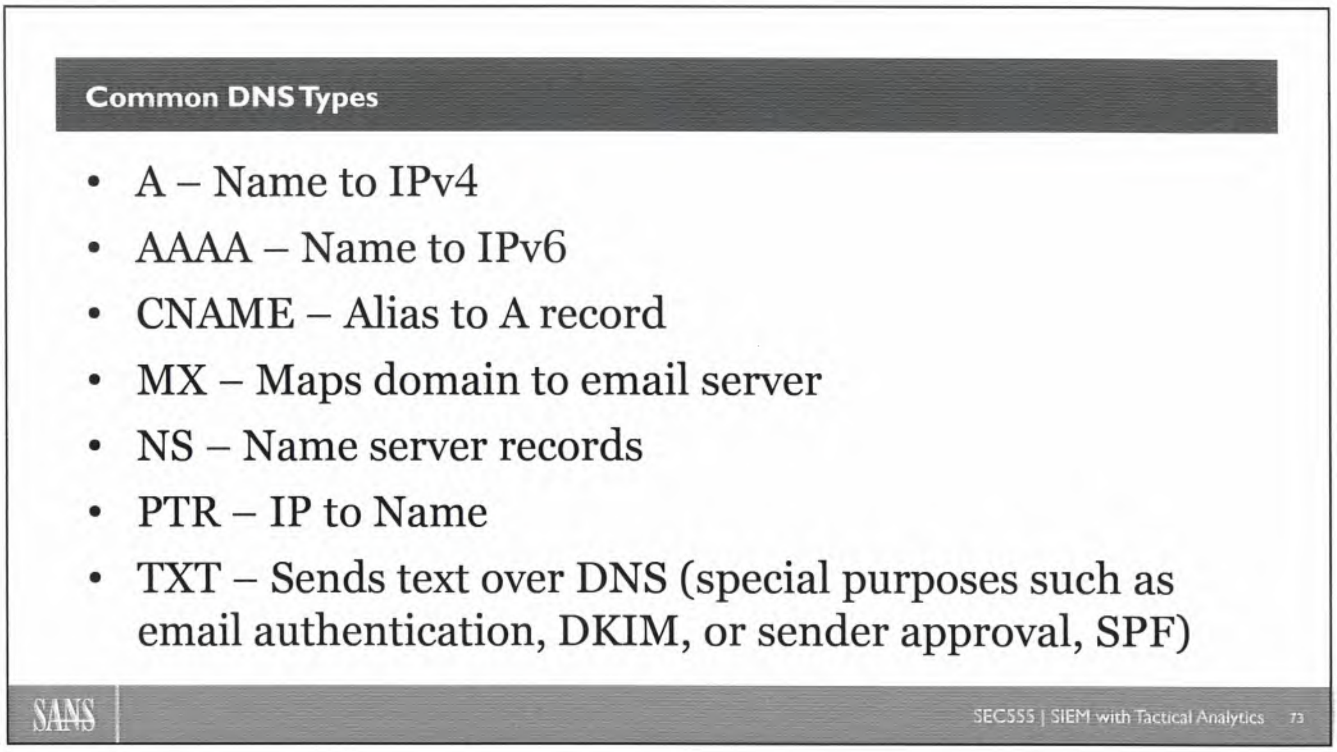
```
10/14/2016 2:06:27 PM 1614 PACKET  000000120D754200 UDP
Rcv 10.0.0.112      0004   Q [0001   D   NOERROR] A
(3)www(6)google(3)com(0)
```

Just guessing... but 10.0.0.112 may have visited Google

Domain Name System (DNS)

Today's networks are dependent on DNS to function correctly. End users need it to know which sites to go to and systems need it to update and transition connections to new systems. Almost every connection made uses DNS. Oddly enough, DNS logs come pretty close to telling the story of what was accessed.

In the above example, a Windows DNS debug log shows 10.0.0.112 requesting the IP address for www.google.com. If this is happening, there is a high likelihood www.google.com is about to be accessed.



Common DNS Types

- A – Name to IPv4
- AAAA – Name to IPv6
- CNAME – Alias to A record
- MX – Maps domain to email server
- NS – Name server records
- PTR – IP to Name
- TXT – Sends text over DNS (special purposes such as email authentication, DKIM, or sender approval, SPF)

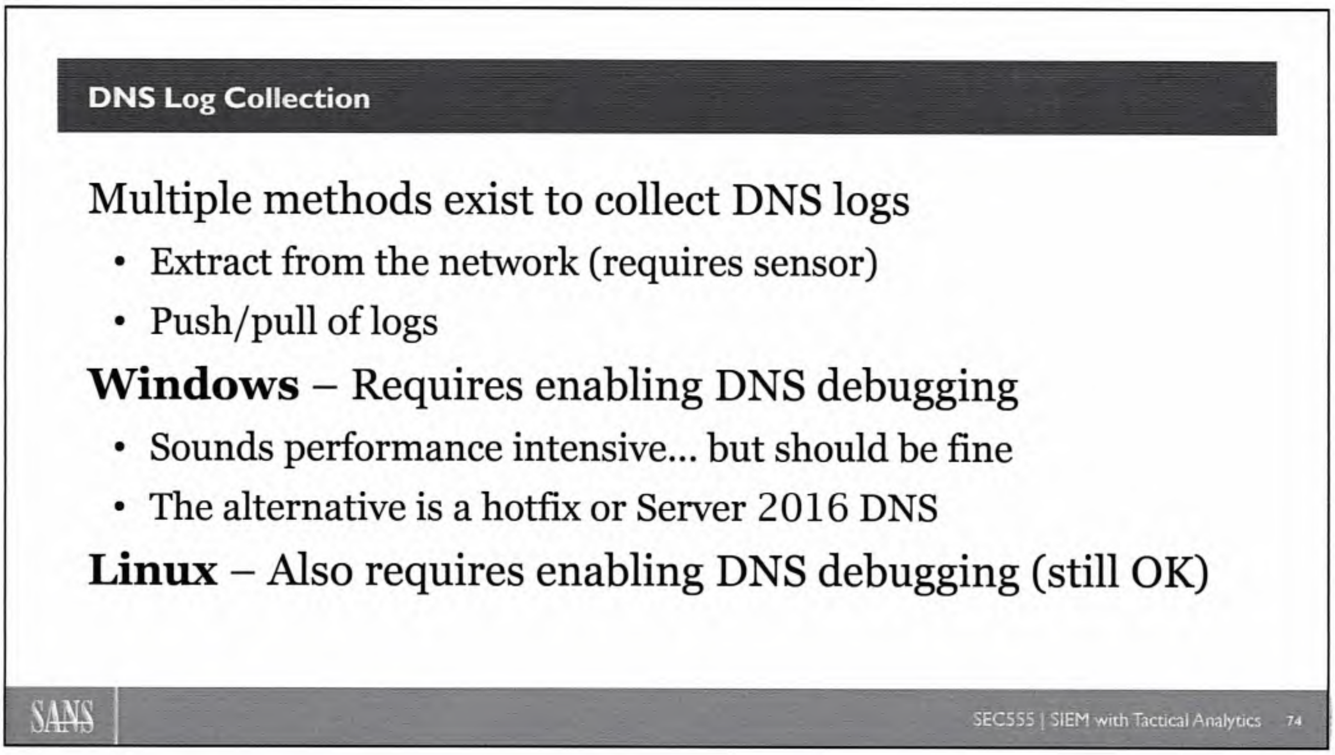
SANS

SEC555 | SIEM with Tactical Analytics 73

Common DNS Types

It is important to know the DNS types available and how they are used. Different techniques apply to different records. For example, when dealing with phishing attacks, MX records may be incredibly useful as they are involved with email routing. Looking for DNS tunneling? This most likely is being done by abusing TXT records.

This brings up the interesting dilemma of how skilled an analyst should be. In truth, an analyst should understand the systems he or she is overseeing. In the case of the SIEM, many different types of systems are involved. Many of the techniques discussed this week should help you understand systems and protocols more by studying what the technique is and, more importantly, why or how it functions.



DNS Log Collection

Multiple methods exist to collect DNS logs

- Extract from the network (requires sensor)
- Push/pull of logs

Windows – Requires enabling DNS debugging

- Sounds performance intensive... but should be fine
- The alternative is a hotfix or Server 2016 DNS

Linux – Also requires enabling DNS debugging (still OK)

SEC555 | SIEM with Tactical Analytics 74

DNS Log Collection

Before you can operationalize DNS, you must first collect the logs. This can be challenging, as enabling the level of logging needed often requires enabling debug logging. Simply mentioning the word debugging tends to rattle system administrators and they generally are inclined to say no. In truth, this does affect performance, but with today's hardware, it's not prone to be a problem.

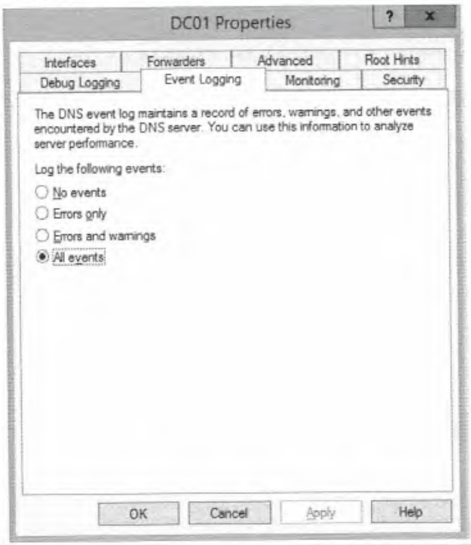
If you are running the newer Server 2012 R2 (with a specific patch) or 2016 or newer operating systems, Microsoft has released a new native DNS logging capability. As stated on TechNet:

"DNS server performance can be affected when additional logging is enabled; however, the enhanced DNS logging and diagnostics feature in Windows Server 2012 R2, and Windows Server 2016 Technical Preview is designed to have a very low impact on performance."[1]

The patch for Server 2012 R2 is KB2956577. This patch requires that KB2919355 is installed prior.

[1] https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/dn800669(v=ws.11), https://sec555.com/4v

Windows DNS



DC01 Properties

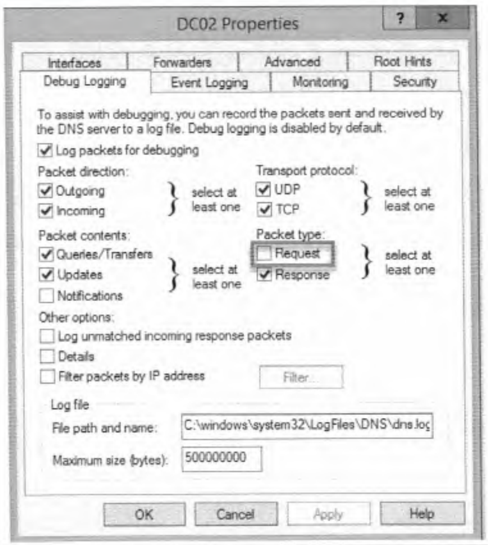
Interfaces | Forwarders | Advanced | Root Hints | Debug Logging | Event Logging | Monitoring | Security

The DNS event log maintains a record of errors, warnings, and other events encountered by the DNS server. You can use this information to analyze server performance.

Log the following events:

○ No events
○ Errors only
○ Errors and warnings
● All events

VS.



DC02 Properties

Interfaces | Forwarders | Advanced | Root Hints | Debug Logging | Event Logging | Monitoring | Security

To assist with debugging, you can record the packets sent and received by the DNS server to a log file. Debug logging is disabled by default.

☑ Log packets for debugging

Packet direction: ☑ Outgoing ☑ Incoming } select at least one

Transport protocol: ☑ UDP ☑ TCP } select at least one

Packet contents: ☑ Queries/Transfers ☑ Updates □ Notifications } select at least one

Packet type: □ Request ☑ Response } select at least one

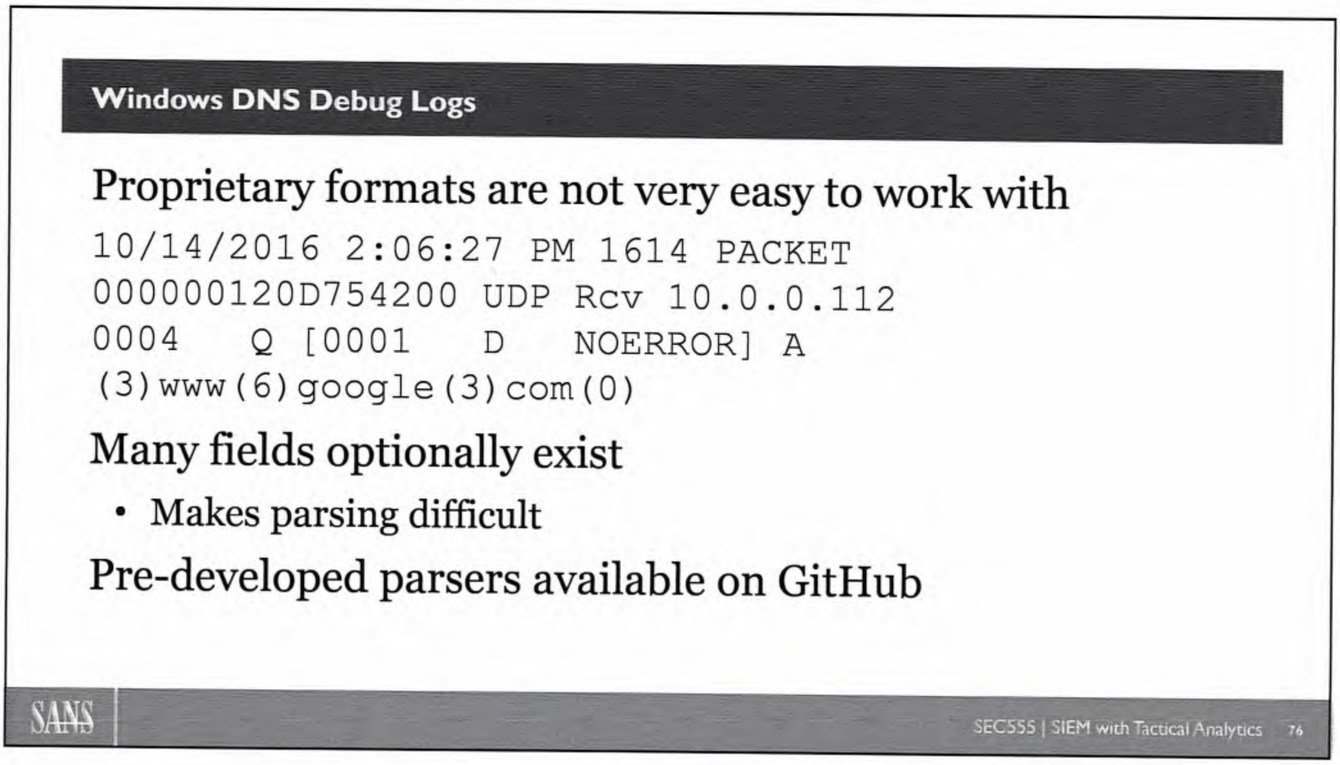
Other options:
□ Log unmatched incoming response packets
□ Details
□ Filter packets by IP address

Log file
File path and name: C:\windows\system32\LogFiles\DNS\dns.loç
Maximum size (bytes): 500000000

Windows DNS

Collecting logs from Windows DNS can be a little confusing. If you open the properties on the DNS server, you are confronted with two tabs: Event logging and debug logging. If you set event logging to all events in the Event Logging tab, you will receive logs of any DNS Server events, not query logging. These are logged in the traditional Windows Event format. However, what you want is query logs. To obtain query logs coming from clients, you have to enable debug logging on the debug logging tab.

Note that if you enable the Packet Type of Request, you will see all DNS queries before a response is generated. This creates unnecessary events and makes it appear that you have duplicate query records. Instead, keep Request unchecked and check Response. This will log queries along with the response. Even if the query fails, this will still generate a log with a response of NXDOMAIN.



Windows DNS Debug Logs

Proprietary formats are not very easy to work with

```
10/14/2016 2:06:27 PM 1614 PACKET
000000120D754200 UDP Rcv 10.0.0.112
0004   Q [0001   D   NOERROR] A
(3)www(6)google(3)com(0)
```

Many fields optionally exist

- Makes parsing difficult

Pre-developed parsers available on GitHub

Windows DNS Debug Logs

Microsoft's debug logs are not very pretty to look at, and the format is not much better. For example, sometimes certain letters or field data exists, and sometimes it does not. Even things like (3)www(6)google(3)com make the log look wonky. In case you are wondering what those numbers are, they are the length of the subsequent string: www has a length of 3, google has a length of 6, and com has a length of 3.

The good news is that the course author has many pre-developed parsers available at github.com/HASecuritySolutions/Logstash. Even if you are using a different SIEM solution, you should be able to reverse the logic of these parsers to meet your organization's needs.

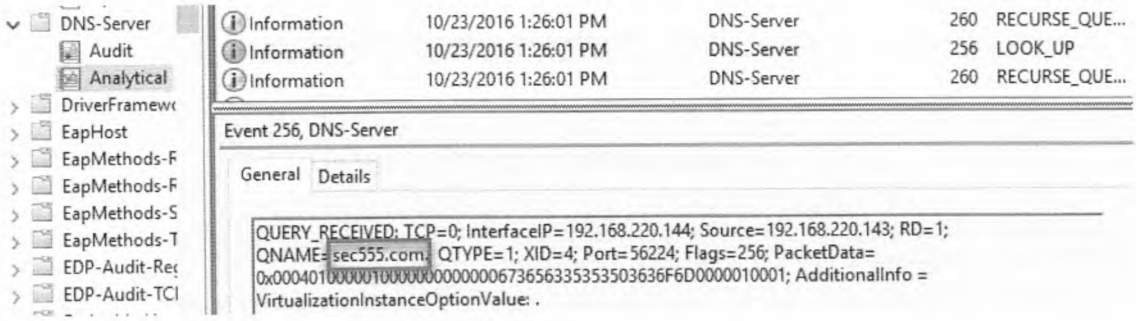
Reference:

https://github.com/HASecuritySolutions/Logstash, https://sec555.com/42

Windows DNS Analytical Log

Requires Server 2012 R2 with a hotfix or 2016+

- Requires log agent with the capability to read ETL files



```
Information    10/23/2016 1:26:01 PM    DNS-Server    260  RECURSE_QUE...
Information    10/23/2016 1:26:01 PM    DNS-Server    256  LOOK_UP
Information    10/23/2016 1:26:01 PM    DNS-Server    260  RECURSE_QUE...
```

Event 256, DNS-Server

```
QUERY_RECEIVED: TCP=0; InterfaceIP=192.168.220.144; Source=192.168.220.143; RD=1;
QNAME=sec555.com. QTYPE=1; XID=4; Port=56224; Flags=256; PacketData=
0x0004010000010000000000000673656335353503636F6D0000010001; AdditionalInfo =
VirtualizationInstanceOptionValue: .
```

Windows DNS Analytical Log

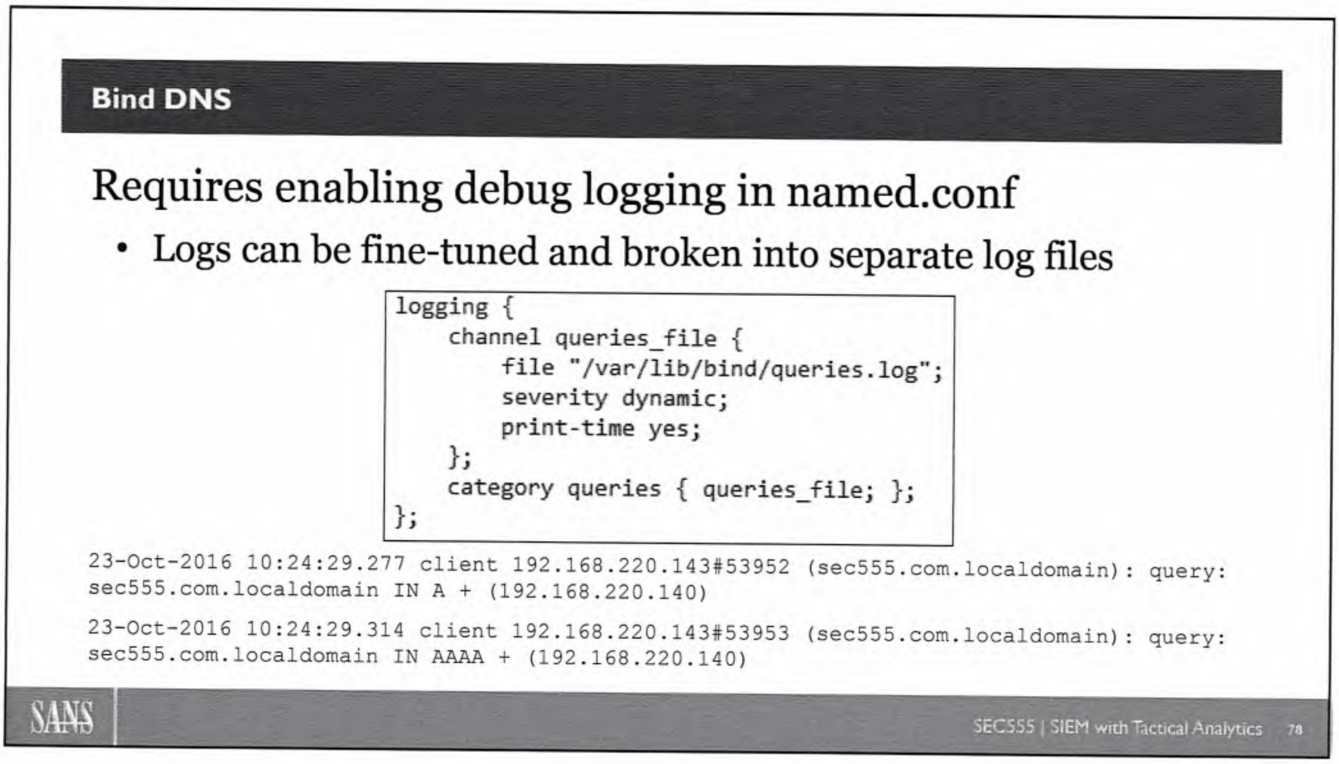
Starting in Server 2012 R2 and later, Windows has developed a new method for DNS logging called the DNS Analytical Log. This is a high-performance log system specific to DNS. This log is stored in an ETL (event trace log), so you must have a log agent that is capable of reading it to collect. One such agent is the NXLog Enterprise Edition.[1] Microsoft has released a statement that the performance impact of enabling this on a DNS server handling 50,000 queries a second is negligible.[2]

To enable this log, open Windows Event Viewer and browse to Application and Service Logs -> Microsoft -> Windows -> DNS-Server. Right-click on DNS-Server and click on View -> Show Analytical and Debug Logs. Then, right-click on Analytical and select Enable Log.

References:

https://nxlog.co, https://sec555.com/2y

https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/dn800669(v=ws.11), https://sec555.com/4v



Bind DNS

Requires enabling debug logging in named.conf

- Logs can be fine-tuned and broken into separate log files

```
logging {
    channel queries_file {
        file "/var/lib/bind/queries.log";
        severity dynamic;
        print-time yes;
    };
    category queries { queries_file; };
};
```

```
23-Oct-2016 10:24:29.277 client 192.168.220.143#53952 (sec555.com.localdomain): query:
sec555.com.localdomain IN A + (192.168.220.140)

23-Oct-2016 10:24:29.314 client 192.168.220.143#53953 (sec555.com.localdomain): query:
sec555.com.localdomain IN AAAA + (192.168.220.140)
```

Bind DNS

Another common DNS service is the open-source DNS solution called Bind. Similar to Windows, debug logging must be enabled to log client queries. Again, this is fairly low-impact but often scares the pants off system administrators.

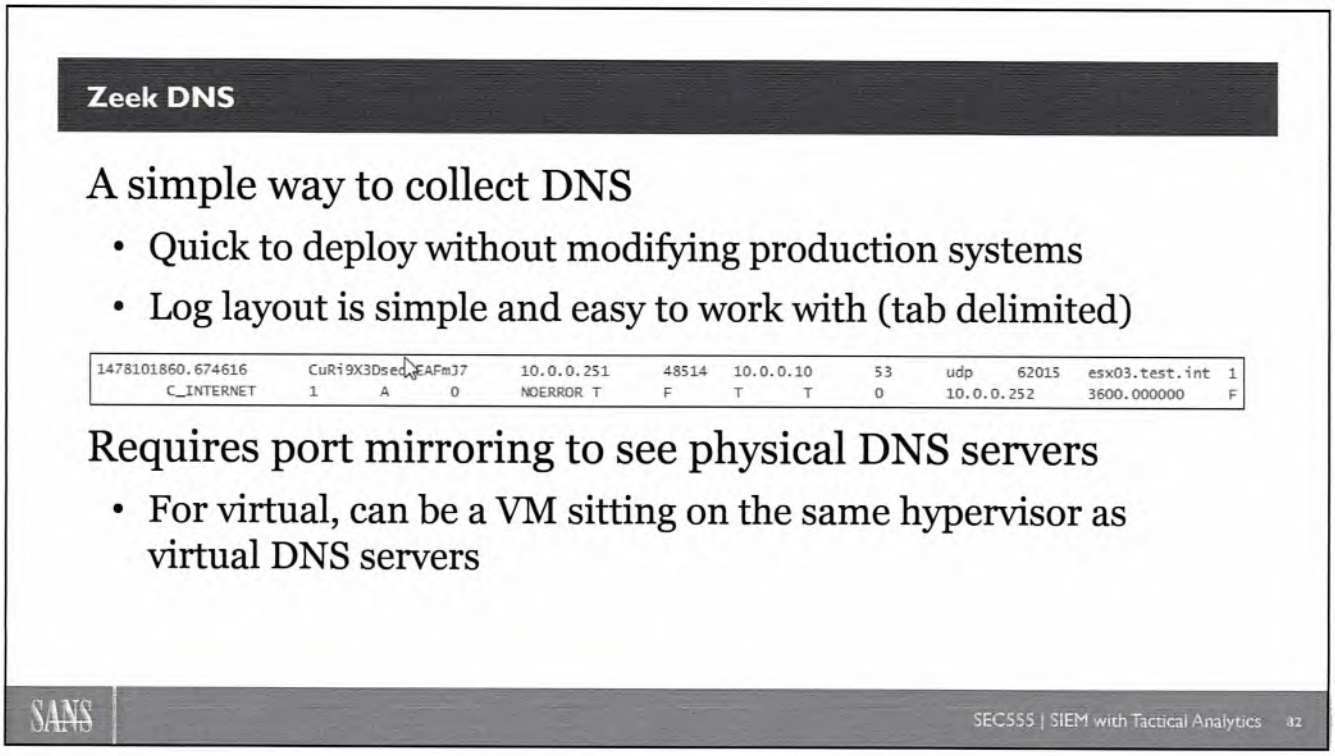
A nice feature of Bind is that category and log level can granularly log. Even the size and a few other options can be configured with Bind. This can be specified in the named.conf configuration file.

Below is an example of the logging portion of named.conf:

```
logging {
   channel default_file {
        file "/var/lib/bind/default.log";
        severity dynamic;
        print-time yes;
    };
    channel general_file {
        file "/var/lib/bind/general.log";
        severity dynamic;
        print-time yes;
    };
    channel database_file {
        file "/var/lib/bind/database.log";
        severity dynamic;
        print-time yes;
    };
    channel security_file {
        file "/var/lib/bind/security.log";
        severity dynamic;
        print-time yes;
    };
    channel config_file {
        file "/var/lib/bind/config.log";
        severity dynamic;
        print-time yes;
    };
    channel resolver_file {
        file "/var/lib/bind/resolver.log";
        severity dynamic;
        print-time yes;
    };
    channel xfer-in_file {
        file "/var/lib/bind/xfer-in.log";
        severity dynamic;
        print-time yes;
    };
    channel xfer-out_file {
        file "/var/lib/bind/xfer-out.log";
        severity dynamic;
        print-time yes;
    };
    channel notify_file {
        file "/var/lib/bind/notify.log";
        severity dynamic;
        print-time yes;
    };
    channel client_file {
        file "/var/lib/bind/client.log";
        severity dynamic;
        print-time yes;
    };
    channel unmatched_file {
        file "/var/lib/bind/unmatched.log";
        severity dynamic;
        print-time yes;
    };
    channel queries_file {
        file "/var/lib/bind/queries.log";
        severity dynamic;
        print-time yes;
    };
    channel network_file {
        file "/var/lib/bind/network.log";
        severity dynamic;
        print-time yes;
    };
    channel update_file {
        file "/var/lib/bind/update.log";
        severity dynamic;
        print-time yes;
    };
   channel dispatch_file {
        file "/var/lib/bind/dispatch.log";
        severity dynamic;
        print-time yes;
    };
    channel dnssec_file {
        file "/var/lib/bind/dnssec.log";
        severity dynamic;
        print-time yes;
    };
    channel lame-servers_file {
        file "/var/lib/bind/lame-servers.log";
        severity dynamic;
        print-time yes;
    };

    category default { default_file; };
    category general { general_file; };
    category database { database_file; };
    category security { security_file; };
    category config { config_file; };
```

```
category resolver { resolver_file; };
    category xfer-in { xfer-in_file; };
    category xfer-out { xfer-out_file; };
    category notify { notify_file; };
    category client { client_file; };
    category unmatched { unmatched_file; };
    category queries { queries_file; };
    category network { network_file; };
    category update { update_file; };
    category dispatch { dispatch_file; };
    category dnssec { dnssec_file; };
    category lame-servers { lame-servers_file; };
};
```

Reference:
https://www.isc.org/bind/, https://sec555.com/5o



Zeek DNS

A simple way to collect DNS

- Quick to deploy without modifying production systems
- Log layout is simple and easy to work with (tab delimited)

```
1478101860.674616      CuRi9X3DsedEAFmJ7      10.0.0.251      48514   10.0.0.10      53      udp     62015   esx03.test.int  1
        C_INTERNET     1         A       0     NOERROR T       F       T       T      0       10.0.0.252      3600.000000     F
```

Requires port mirroring to see physical DNS servers

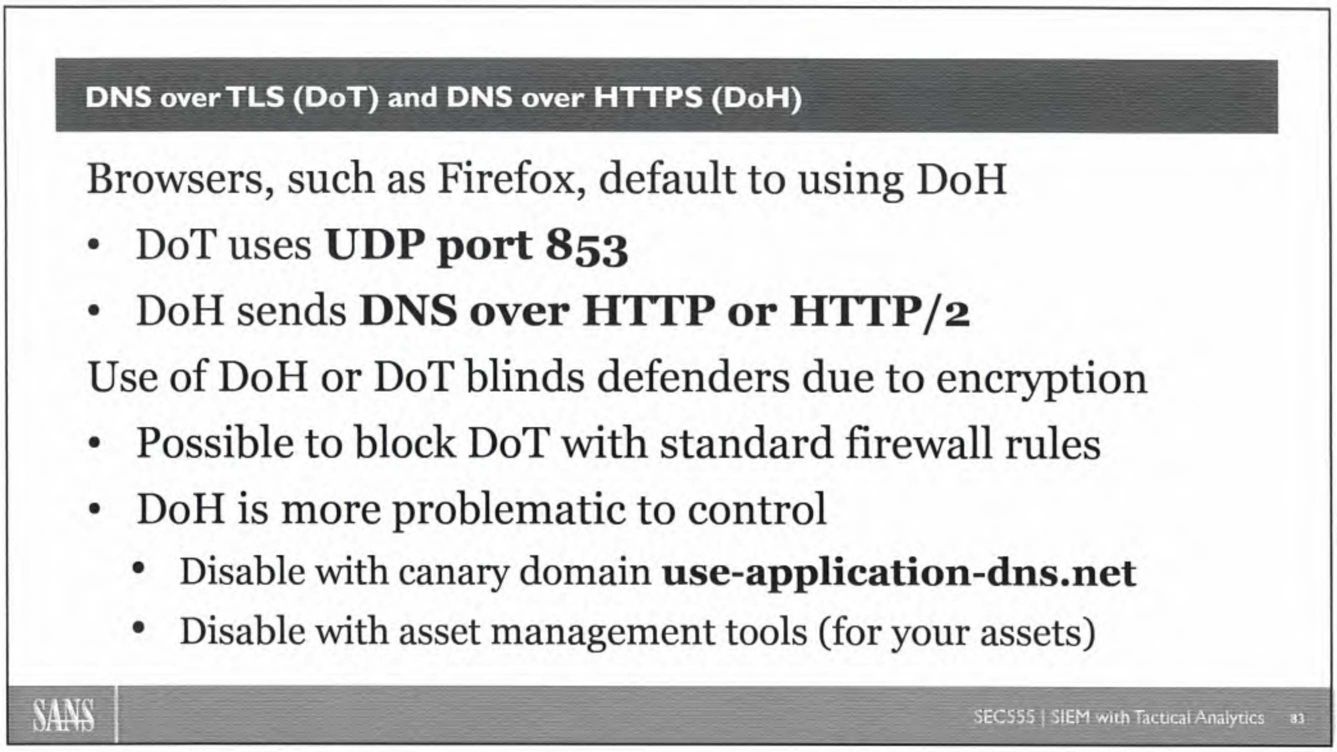
- For virtual, can be a VM sitting on the same hypervisor as virtual DNS servers

Zeek DNS

Possibly one of the easiest ways to start collecting DNS logs is to deploy Zeek. Instead of modifying the configuration of critical DNS systems or impacting their performance, you may opt to stand up a Zeek sensor and mirror the network traffic from your DNS systems to the Zeek sensor. This makes for quick DNS collection. Another huge benefit is that the Zeek DNS logs are in a much friendlier format than either Windows DNS or Bind. They default to tab delimited, and the default fields are great as is. Optionally, you can reconfigure Zeek to log in JSON.

Reference:

https://www.zeek.org, https://sec555.com/4h



DNS over TLS (DoT) and DNS over HTTPS (DoH)

Browsers, such as Firefox, default to using DoH

- DoT uses **UDP port 853**
- DoH sends **DNS over HTTP or HTTP/2**

Use of DoH or DoT blinds defenders due to encryption

- Possible to block DoT with standard firewall rules
- DoH is more problematic to control
  - Disable with canary domain **use-application-dns.net**
  - Disable with asset management tools (for your assets)

SANS

SEC555 | SIEM with Tactical Analytics 83

DNS over TLS (DoT) and DNS over HTTPS (DoH)

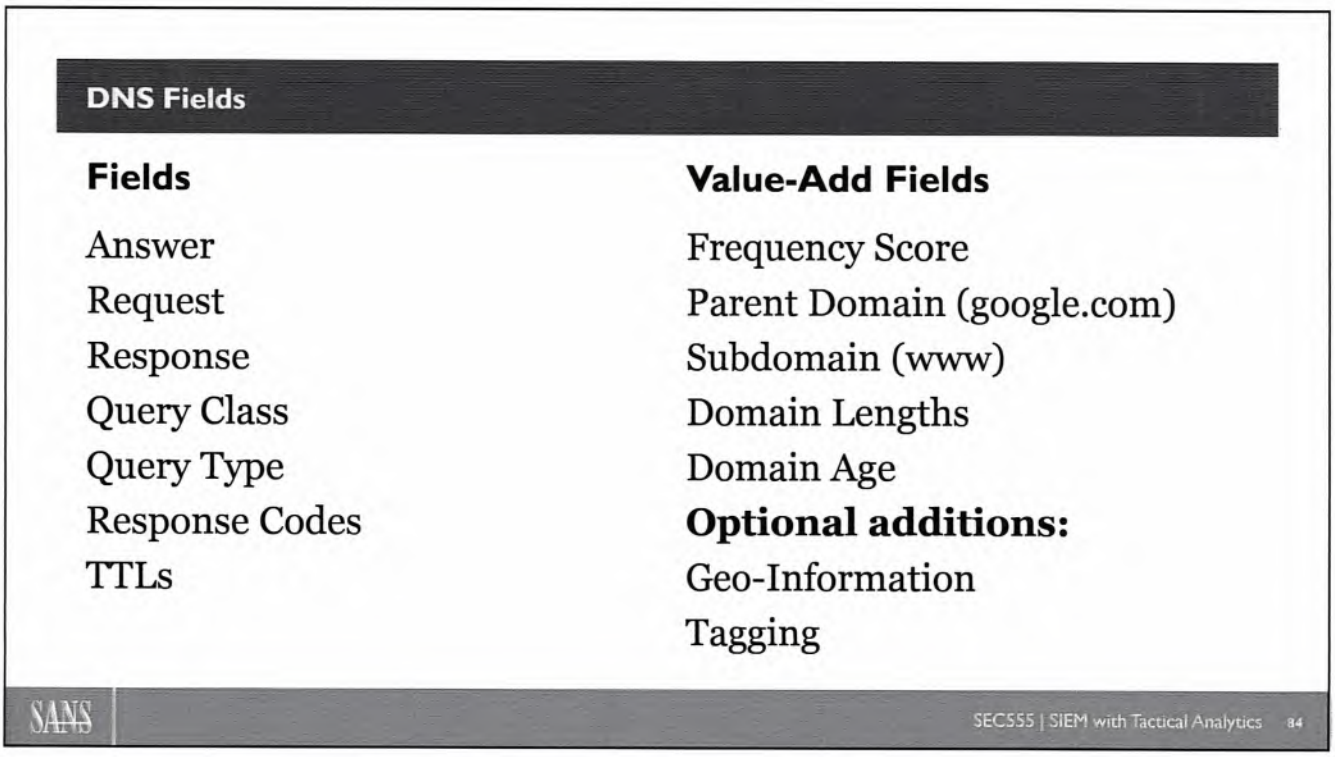
DNS over TLS (DoT) and DNS over HTTPS (DoH) is the modern implementation of DNS. DoH and DoT address privacy concerns as well as protect against machine-in-the-middle attacks. The problem is privacy, and enterprise visibility is not complimentary.

Adding encryption to DNS means organizations are unable to see which clients are requesting a given domain. Without this visibility, many effective DNS monitoring controls become obsolete. To address visibility issues, organizations may consider disabling DoT by blocking UDP port 853 with a router or firewall. DoH is more difficult. Organizations may try to prevent clients from using DoH by setting up a canary domain such as Firefox's use-application-dns.net1. If organizations configure their internal DNS servers to control the domain use-application-dns.net then the organization could disable or point clients to use their own internal DoH service. The downside to this approach is that if the browser is manually configured to point to a DoH server, then the default setting of checking the canary domain will not be enforced. For organization-controlled assets, registry keys, group policies, and configuration files can be utilized to disable DoH within browsers.

References:

https://support.mozilla.org/en-US/kb/firefox-dns-over-https

https://www.cloudflare.com/learning/dns/dns-over-tls/



DNS Fields

Fields

Answer
Request
Response
Query Class
Query Type
Response Codes
TTLs

Value-Add Fields

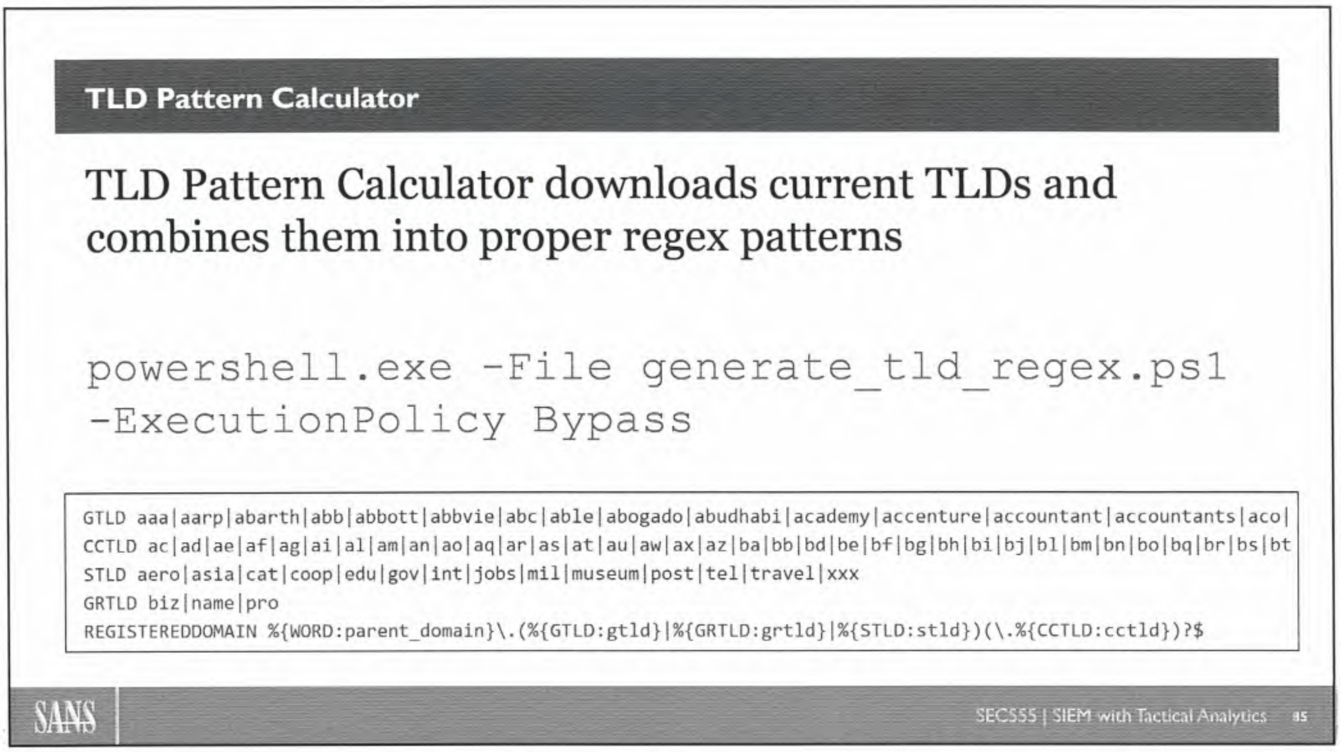
Frequency Score
Parent Domain (google.com)
Subdomain (www)
Domain Lengths
Domain Age
Optional additions:
Geo-Information
Tagging

DNS Fields

At a high level, DNS comes with a few key fields such as requests, responses, query types, and answers. To add more value and weaponize your logs, you should consider adding the following fields:

1. Frequency Score: Entropy test for how random a field is.
2. Parent Domain: High-level domain name (Example: would be google.com in mail.google.com).
3. Child Domain: Domains below parent domain (Example: would be mail in mail.google.com).
4. Domain Lengths
5. Domain Age

Other things that could be used to augment DNS logs would be GeoIP lookups and tagging.



TLD Pattern Calculator

TLD Pattern Calculator downloads current TLDs and combines them into proper regex patterns

```
powershell.exe -File generate_tld_regex.ps1
-ExecutionPolicy Bypass
```

```
GTLD aaa|aarp|abarth|abb|abbott|abbvie|abc|able|abogado|abudhabi|academy|accenture|accountant|accountants|aco|
CCTLD ac|ad|ae|af|ag|ai|al|am|an|ao|aq|ar|as|at|au|aw|ax|az|ba|bb|bd|be|bf|bg|bh|bi|bj|bl|bm|bn|bo|bq|br|bs|bt
STLD aero|asia|cat|coop|edu|gov|int|jobs|mil|museum|post|tel|travel|xxx
GRTLD biz|name|pro
REGISTEREDDOMAIN %{WORD:parent_domain}\.(%{GTLD:gtld}|%{GRTLD:grtld}|%{STLD:stld})(\.%{CCTLD:cctld})?$
```

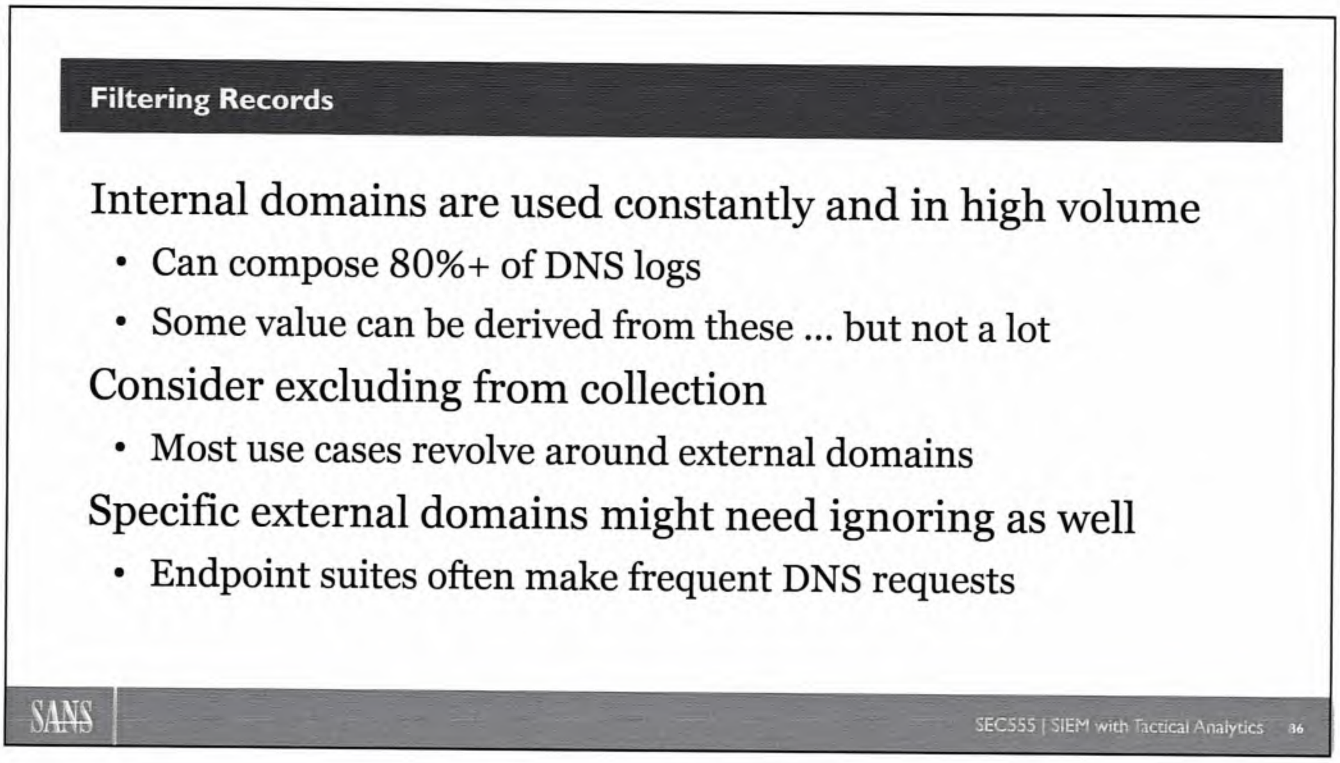
TLD Pattern Calculator

Oddly, SIEM solutions often do not have regex patterns for current top-level domains (TLDs). As the internet changes, so do the list of supported top-level domains. The problem is organizations need to know the latest TLDs to properly pull-out domain names.

Fortunately, there are projects online that help organizations parse out the latest TLDs. One such project is the TLD Pattern Calculator[1] GitHub project. The TLD Pattern Calculator contains a PowerShell script that pulls down a list of supported TLDs and then combines them into a regex pattern file. The GitHub project also contains the output of the top-level domain regex patterns from a former PowerShell execution.

Reference:

https://github.com/HASecuritySolutions/tld_pattern_calculator



Filtering Records

Internal domains are used constantly and in high volume

- Can compose 80%+ of DNS logs
- Some value can be derived from these ... but not a lot

Consider excluding from collection

- Most use cases revolve around external domains

Specific external domains might need ignoring as well

- Endpoint suites often make frequent DNS requests

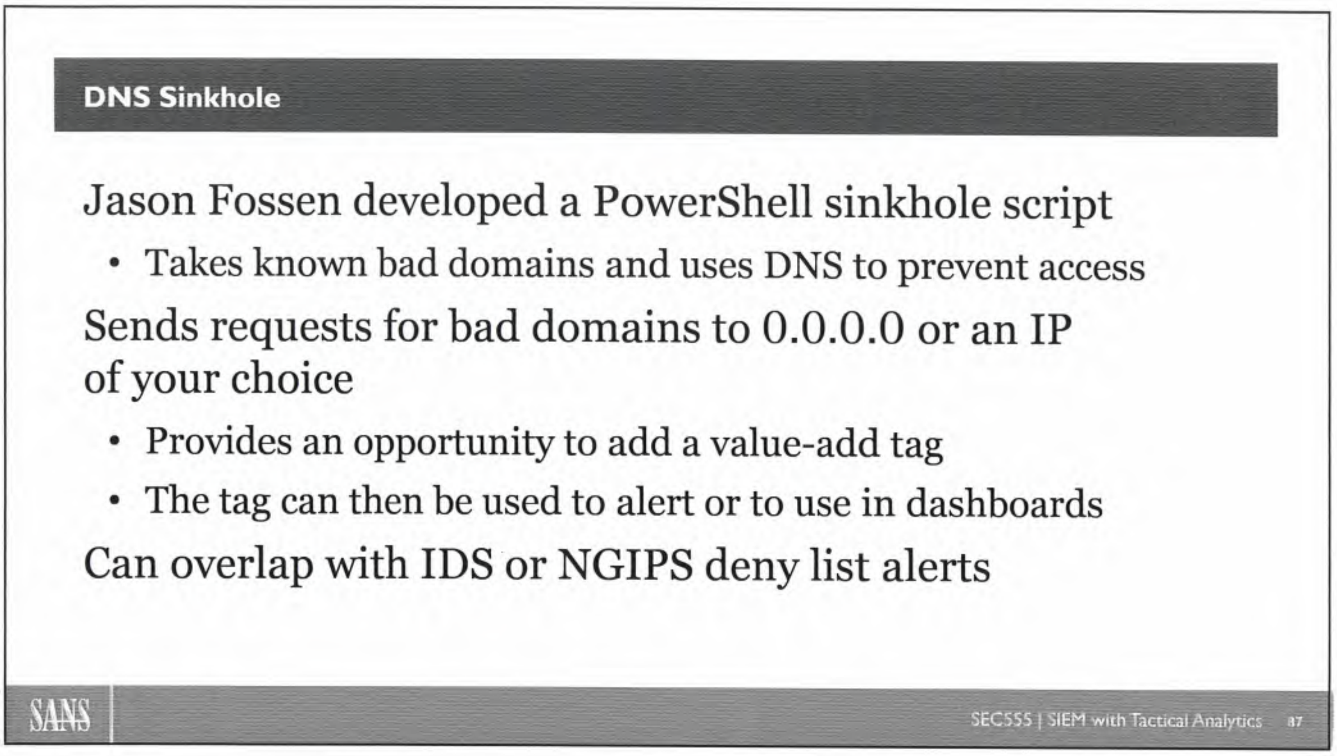
SANS

SEC555 | SIEM with Tactical Analytics 86

Filtering Records

A key decision needs to be made early on in DNS collection. Should you collect internal DNS requests or not? Internal systems are constantly talking to each other and, thus, an extremely large portion of DNS logs are from internal DNS domains. There may be some techniques that are designed around internal DNS queries, but the value-add and rate of return probably are not there. If you decide to ignore internal domain queries, it will probably shrink the amount of DNS logs collected by 80 to 90 percent. If cutting internal DNS queries, try to do so at the agent or set up DNS servers that exclusively handle DNS recursion, so they only are logging external domain requests.

Keep in mind that a majority of DNS analysis techniques revolve around looking for external domains. This is because malware either talks directly with IP addresses or, more commonly, through DNS.



DNS Sinkhole

Jason Fossen developed a PowerShell sinkhole script

- Takes known bad domains and uses DNS to prevent access

Sends requests for bad domains to 0.0.0.0 or an IP of your choice

- Provides an opportunity to add a value-add tag
- The tag can then be used to alert or to use in dashboards

Can overlap with IDS or NGIPS deny list alerts

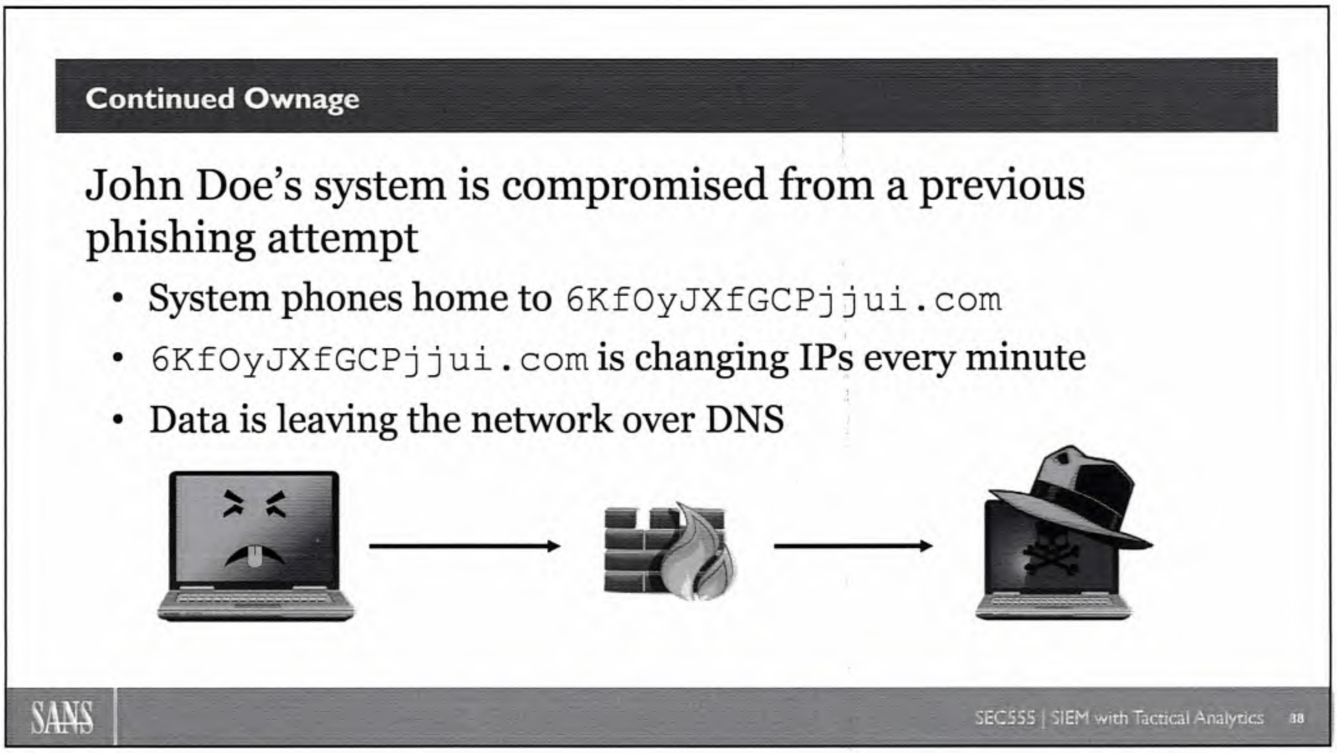
DNS Sinkhole

Sometimes, logging allows for enhancing pre-existing techniques. Jason Fossen, a SANS author and leader in everything Windows, developed a PowerShell script that blocks access to bad domains. The script takes a text file containing a list of bad domains and then uses Windows DNS to send all requests for these domains to either 0.0.0.0 or an IP address of your choice. This technique is considered a DNS sinkhole.

While it is great that it blocks access to bad domains, it is equally important that you are notified that a machine is attempting to access a known bad domain. Using DNS query logging and looking for the IP address being used to sinkhole domains gives you exactly that.

Reference:

https://www.sans.org/blog/windows-dns-server-sinkhole-domains-tool/, https://sec555.com/4w



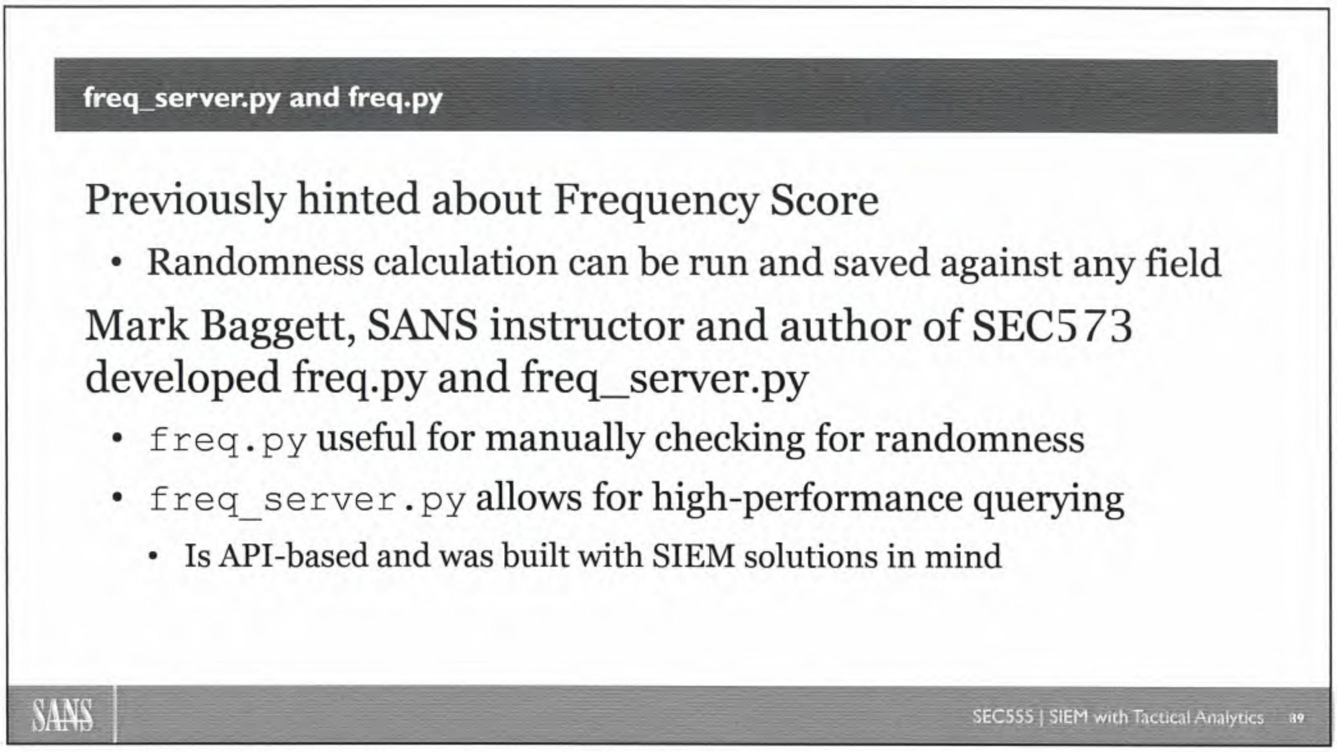
Continued Ownage

John Doe's system is compromised from a previous phishing attempt

- System phones home to `6KfOyJXfGCPjjui.com`
- `6KfOyJXfGCPjjui.com` is changing IPs every minute
- Data is leaving the network over DNS

Continued Ownage

Going back to the phishing email against John Doe, let's assume he clicked the link in the email and malware infected his box. Once the malware starts, it phones home to 6KfOyJXfGCPjjui.com. However, the IP address behind 6KfOyJXfGCPjjui.com changes every minute to avoid IP deny lists. Worst of all, the malware begins stealing data and is sending it out of the environment using DNS.



freq_server.py and freq.py

Previously hinted about Frequency Score

- Randomness calculation can be run and saved against any field

Mark Baggett, SANS instructor and author of SEC573 developed freq.py and freq_server.py

- `freq.py` useful for manually checking for randomness
- `freq_server.py` allows for high-performance querying
  - Is API-based and was built with SIEM solutions in mind

freq_server.py and freq.py

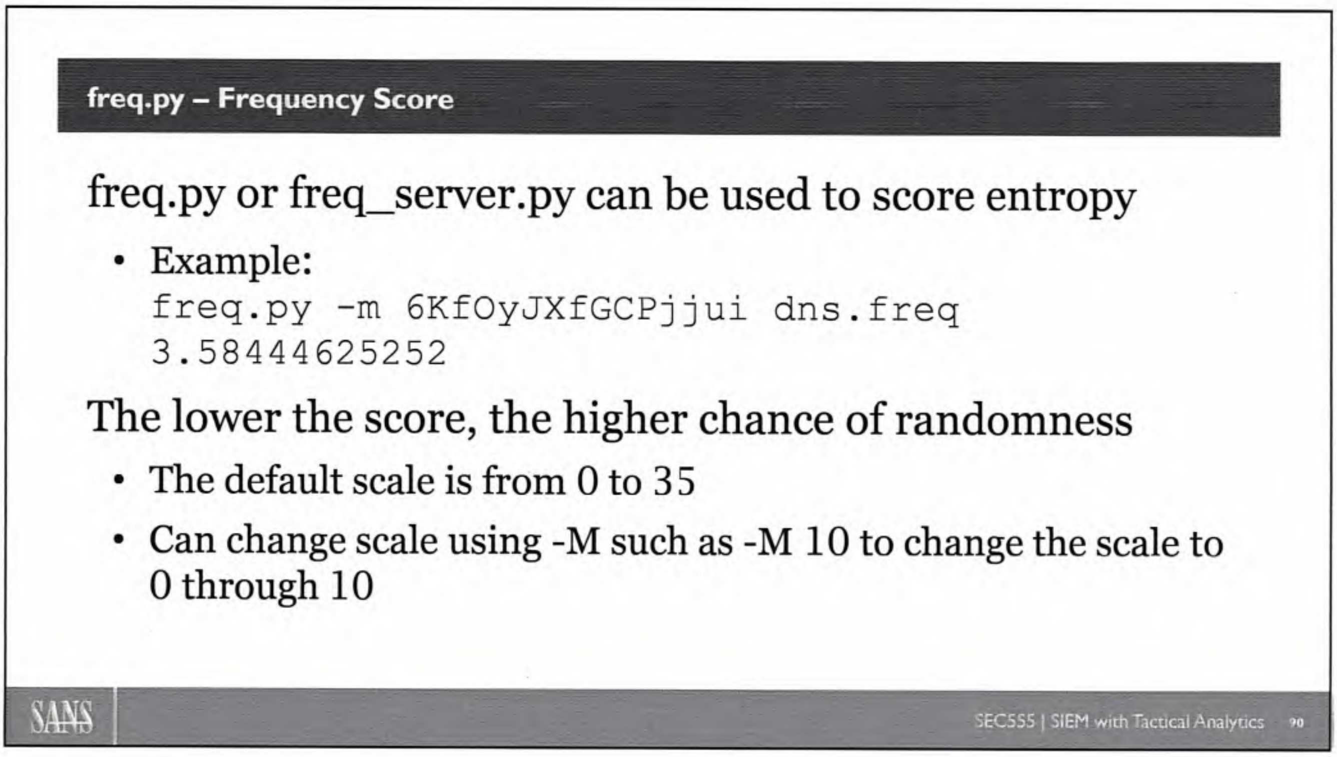
In previous slides, a field called frequency_score was introduced. This field contains a numeric float value that is used to express how random something is. This type of calculation is easy when using Mark Baggett's freq.py and freq_server.py scripts.[1] How this works is by mathematically calculating the frequency of occurrence of characters. For example, if the first character is the letter A, how likely will it be that the second character that follows is going to be a T and so forth. In the English dictionary, certain letters tend to follow other letters. If something is truly random, then this pattern will be broken.

The beauty of these scripts is that they allow creating your frequency tables. For example, you could have a frequency table based on the English dictionary as well as a frequency table of domain names generated by the Alexa top 1 million sites. This allows different applications for different use cases.

freq.py is useful for one-off manual checks or bulk checks against a text file. freq_server.py is designed as an API that can handle large-scale requests and is great for log augmentation.

Reference:

https://github.com/MarkBaggett/freq, https://sec555.com/4y



freq.py – Frequency Score

freq.py or freq_server.py can be used to score entropy

- Example:

```
freq.py -m 6KfOyJXfGCPjjui dns.freq
3.58444625252
```

The lower the score, the higher chance of randomness

- The default scale is from 0 to 35
- Can change scale using -M such as -M 10 to change the scale to 0 through 10

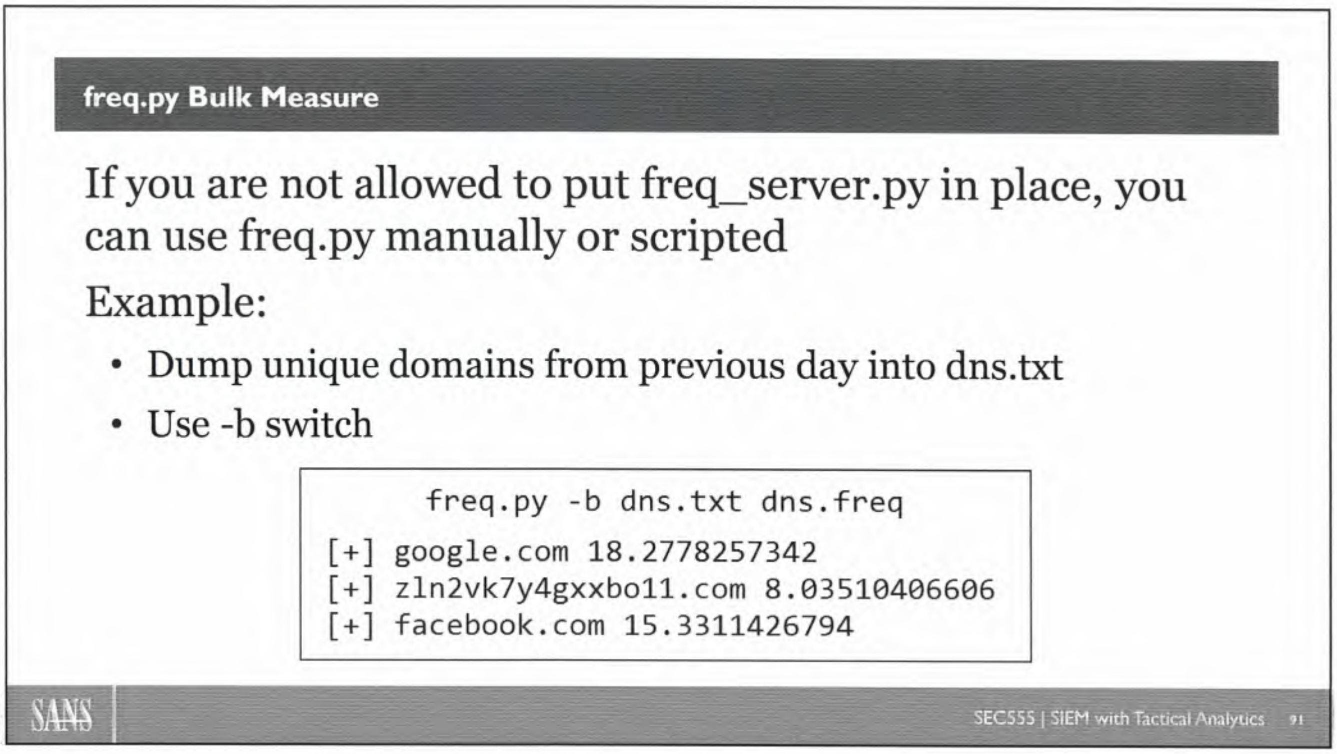
SANS

SEC555 | SIEM with Tactical Analytics 90

freq.py – Frequency Score

Both freq.py and freq_server.py are used to generate an entropy score indicating randomness. With these scripts, the lower the score, the higher the likelihood something is random. By default, the scale ranges from 0 to 35, but this can be overridden using the -M 10 flag to utilize a more traditional 0 to 10 scale.

In this example, dns.freq is a frequency table used for calculating entropy. Mark's tool supports building custom frequency tables. For example, the English dictionary can be fed through freq.py to generate a frequency table specific to words in the English dictionary. Then when a string is used to calculate entropy against this, the frequency score will be based on the rate of occurrence of characters compared to the English dictionary. This makes freq.py powerful, as custom tables can be generated for specific use cases.



freq.py Bulk Measure

If you are not allowed to put freq_server.py in place, you can use freq.py manually or scripted

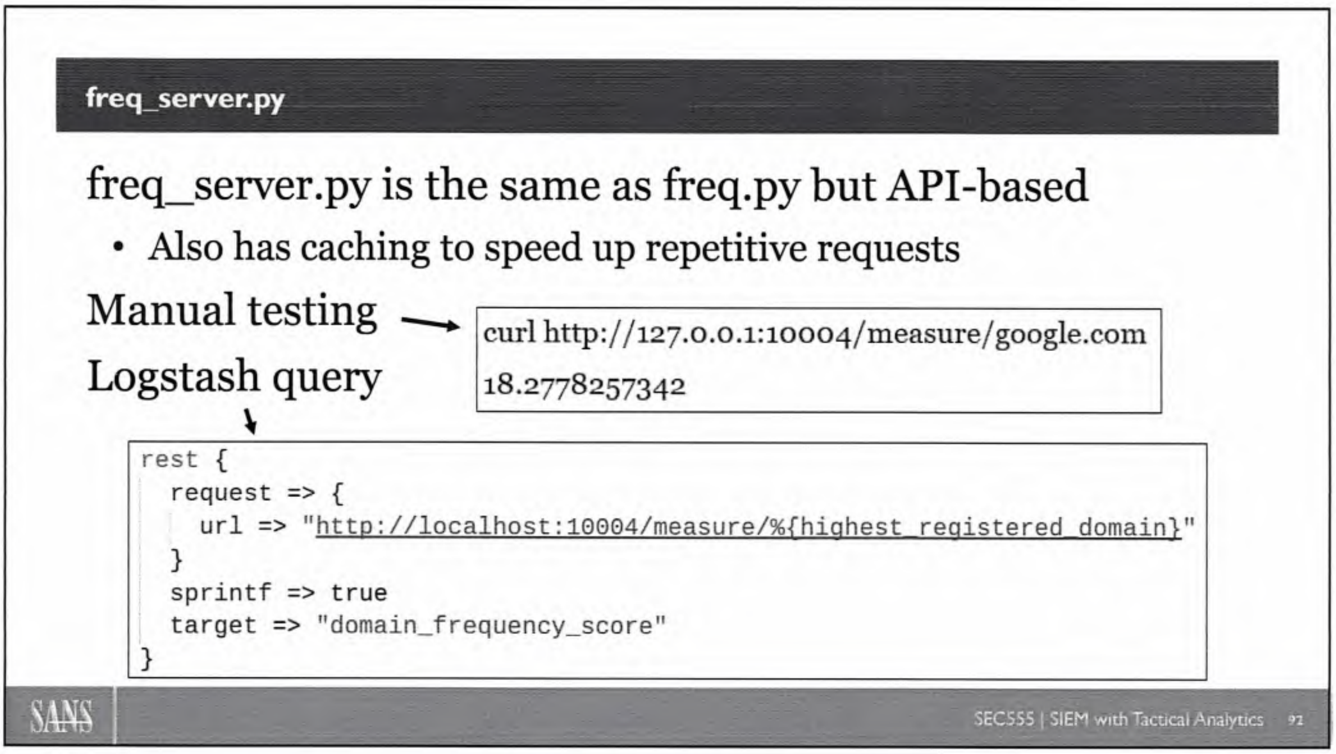
Example:

- Dump unique domains from previous day into dns.txt
- Use -b switch

```
      freq.py -b dns.txt dns.freq
[+] google.com 18.2778257342
[+] zln2vk7y4gxxbo11.com 8.03510406606
[+] facebook.com 15.3311426794
```

freq.py Bulk Measure

It is possible that your SIEM solution does not allow external API calls to freq_server.py or that you are not authorized to implement freq_server.py. In these cases, it may still be possible to apply the technique in an out-of-band method using freq.py. For example, if you dumped a list of domains to a text file (manually or scripted), you could then run the text file through freq.py using the -b (bulk measure) switch.



freq_server.py

freq_server.py is the same as freq.py but API-based

- Also has caching to speed up repetitive requests

Manual testing →

```
curl http://127.0.0.1:10004/measure/google.com
18.2778257342
```

Logstash query

```
rest {
  request => {
    url => "http://localhost:10004/measure/%{highest_registered_domain}"
  }
  sprintf => true
  target => "domain_frequency_score"
}
```

SANS

SEC555 | SIEM with Tactical Analytics 92

freq_server.py

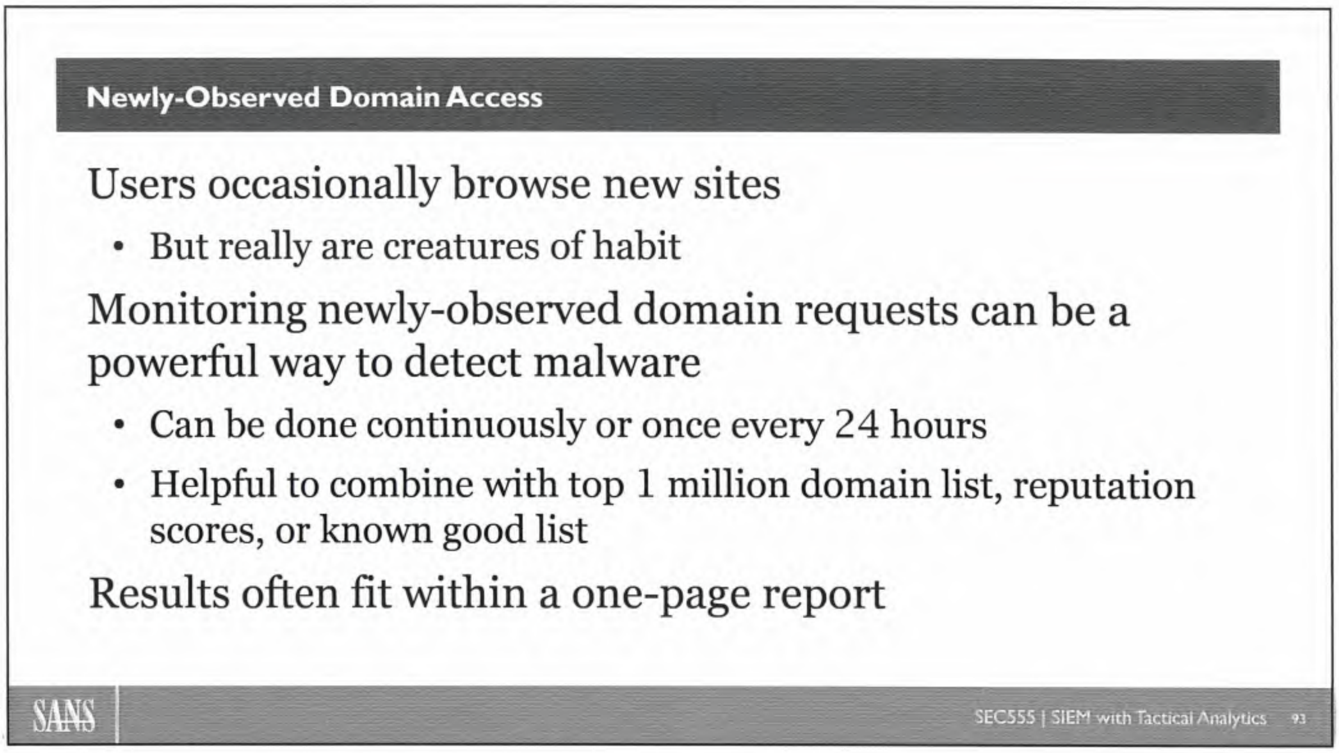
To perform entropy tests during log ingestion, you likely want to utilize freq_server.py rather than freq.py. It is intended to run as a service and accepts API calls to measure a string's entropy. Similar to freq.py, a custom frequency table can be used and run as a service. To support multiple frequency tables, start freq_server.py using a different port and different frequency table.

The curl command, in the top picture, reflects manually submitting the string "google.com" against freq_server.py. The resulting score was 18.2778257342. Curl can be invoked from Logstash, but a more appropriate method would be to use the rest community plugin[2] and query freq_server.py with it. In the bottom picture, the port 10004 is used, and the field value of highest_registered_domain within a log is passed. The result is then stored in a new field called domain_frequency_score.

References:

https://github.com/MarkBaggett/freq, https://sec555.com/4y

https://github.com/lucashenning/logstash-filter-rest, https://sec555.com/4z



Newly-Observed Domain Access

Users occasionally browse new sites

- But really are creatures of habit

Monitoring newly-observed domain requests can be a powerful way to detect malware

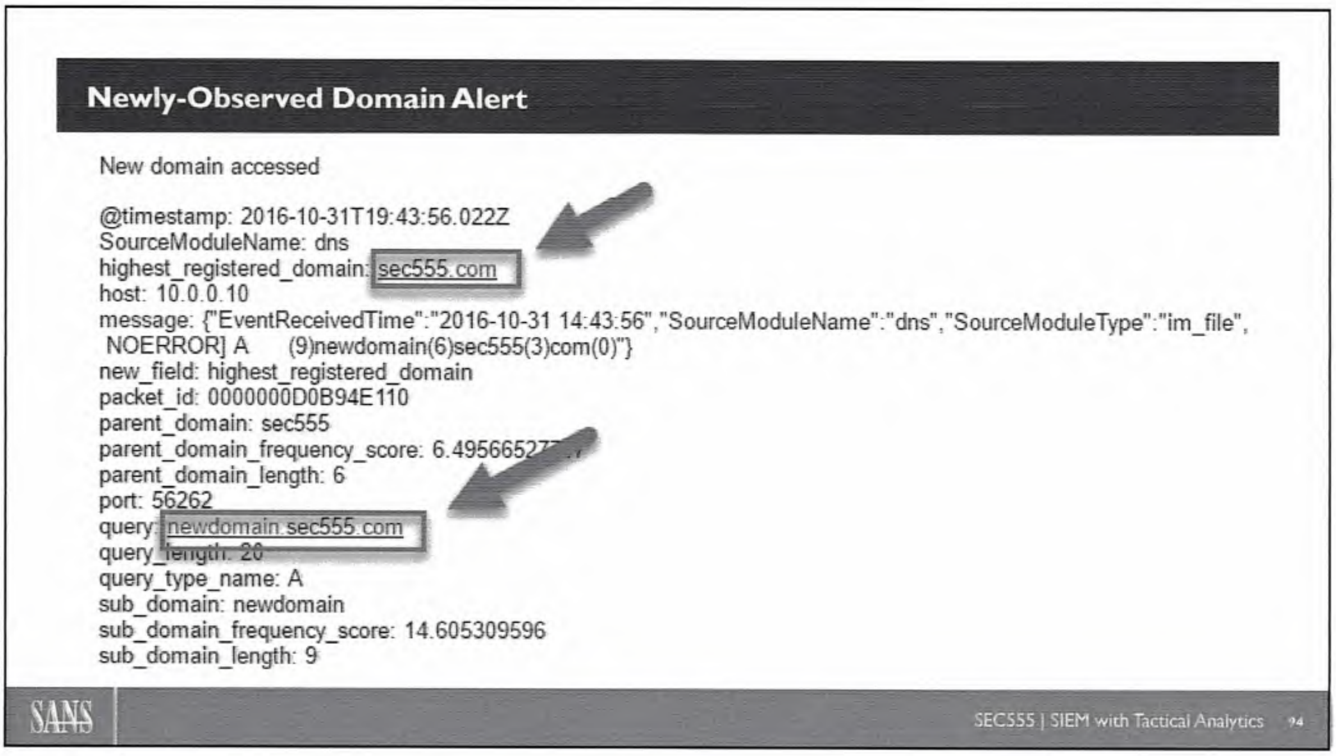
- Can be done continuously or once every 24 hours
- Helpful to combine with top 1 million domain list, reputation scores, or known good list

Results often fit within a one-page report

Newly-Observed Domain Access

If you think about it, network-based attacks occur either directly through an IP address or by using DNS names. Because of IP deny lists being commonplace, most of today's malware uses DNS names. This allows for another method of taking an opponent's strength and using it against them. If attacks are going to be coming in using DNS domains, then it is possible to catch them simply by inspecting new domains being used.

This can be done continuously using a SIEM alert engine or it can be done by once daily having someone investigate a list of new domains that appeared the previous day. This technique is easier to apply if you first filter out Alexa domains or a list of known good domains.



Newly-Observed Domain Alert

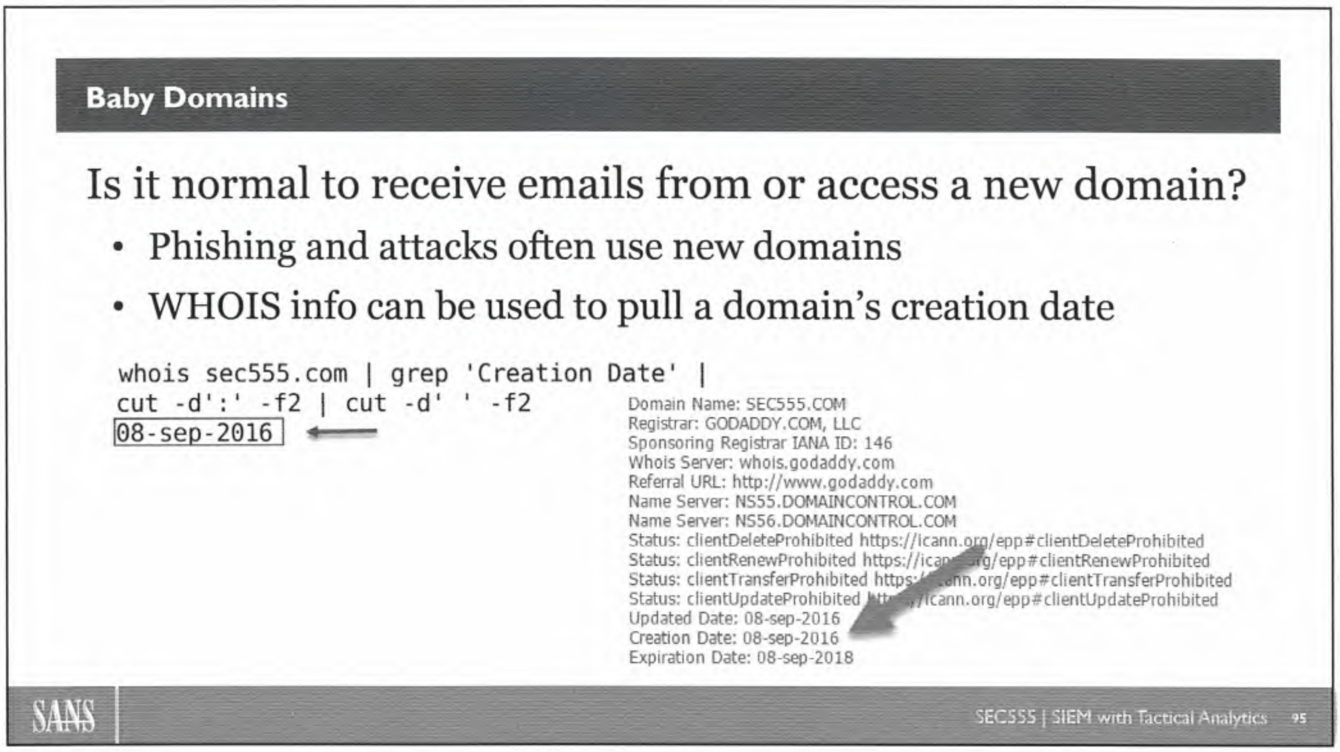
```
New domain accessed

@timestamp: 2016-10-31T19:43:56.022Z
SourceModuleName: dns
highest_registered_domain: sec555.com
host: 10.0.0.10
message: {"EventReceivedTime":"2016-10-31 14:43:56","SourceModuleName":"dns","SourceModuleType":"im_file",
 NOERROR] A     (9)newdomain(6)sec555(3)com(0)"}
new_field: highest_registered_domain
packet_id: 0000000D0B94E110
parent_domain: sec555
parent_domain_frequency_score: 6.4956652
parent_domain_length: 6
port: 56262
query: newdomain.sec555.com
query_length: 20
query_type_name: A
sub_domain: newdomain
sub_domain_frequency_score: 14.605309596
sub_domain_length: 9
```

Newly-Observed Domain Alert

This screenshot is a copy of an email generated by ElastAlert.[1] It is an open-source alert engine for Elasticsearch and, in this scenario, is being used to generate an alert every time a new domain is seen that has not been accessed in the last 90 days and is not a top 1 million domain. In this example, sec555.com was seen for the first time, so an alert was generated. Using ElastAlert allows for a near real-time monitor as it checks for new domains every minute with default settings.

Note that this technique is being applied against the high-level domain of sec555.com, not newdomain.sec555.com. The inclusion of subdomains would make this technique trigger on something every second.

[1] https://github.com/Yelp/elastalert, https://sec555.com/3x



Baby Domains

Is it normal to receive emails from or access a new domain?

- Phishing and attacks often use new domains
- WHOIS info can be used to pull a domain's creation date

```
whois sec555.com | grep 'Creation Date' |
cut -d':' -f2 | cut -d' ' -f2
08-sep-2016
```

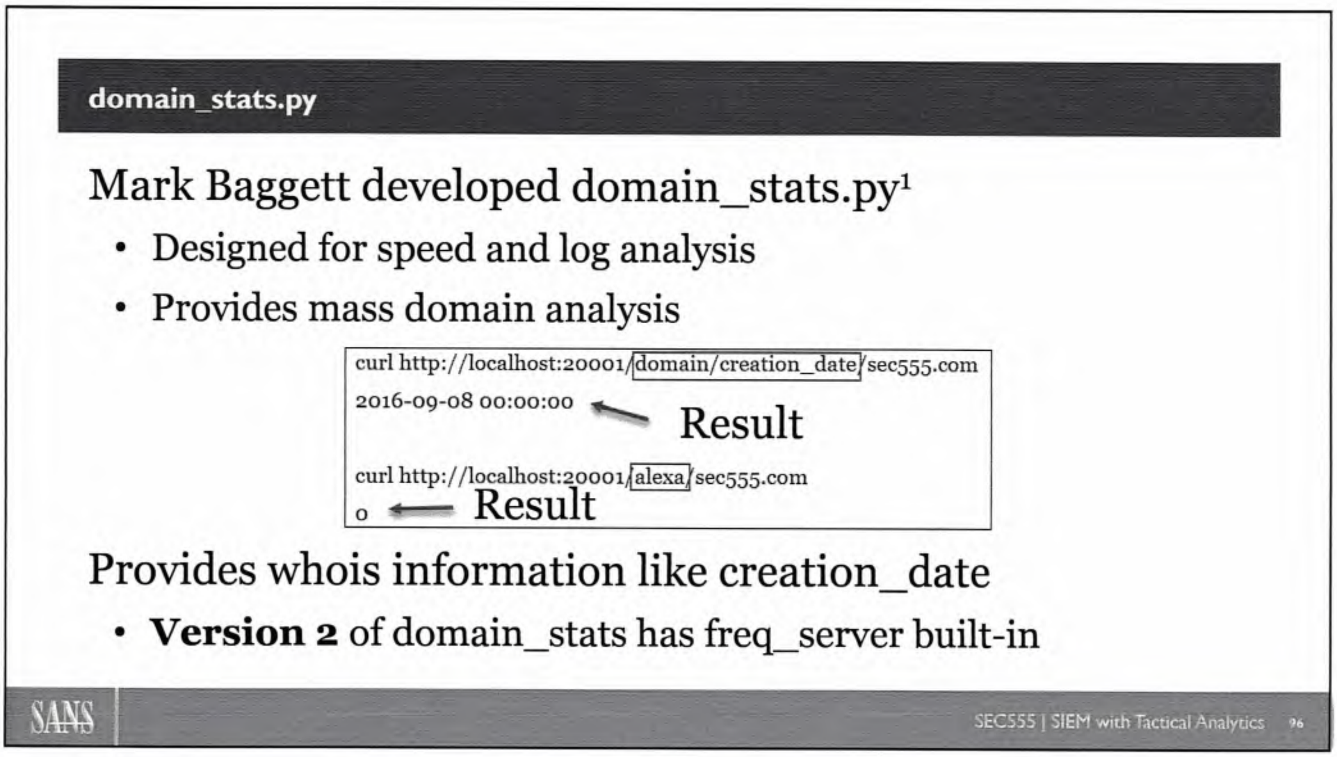
```
Domain Name: SEC555.COM
Registrar: GODADDY.COM, LLC
Sponsoring Registrar IANA ID: 146
Whois Server: whois.godaddy.com
Referral URL: http://www.godaddy.com
Name Server: NS55.DOMAINCONTROL.COM
Name Server: NS56.DOMAINCONTROL.COM
Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Status: clientRenewProhibited https://icann.org/epp#clientRenewProhibited
Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Updated Date: 08-sep-2016
Creation Date: 08-sep-2016
Expiration Date: 08-sep-2018
```

Baby Domains

Another technique that is interesting is the monitoring of domain age. Domains are registered with ICANN,[1] and part of this registration record keeps track of when a domain was created. This information can be retrieved by querying WHOIS lookups. For example, the Linux command "*whois sec555.com | grep 'Creation Date' | grep Z | cut -d':' -f2 | cut -d' ' -f2*" would return the creation date for sec555.com. This technique is not something that would scale well if applying against every DNS query log. Instead, it works best when filtering out Alexa domains and, if possible, only running it once against new domains.

Reference:

https://www.icann.org, https://sec555.com/50



domain_stats.py

Mark Baggett developed domain_stats.py[1]

- Designed for speed and log analysis
- Provides mass domain analysis

```
curl http://localhost:20001/domain/creation_date/sec555.com
2016-09-08 00:00:00
```

Result

```
curl http://localhost:20001/alexa/sec555.com
0
```

Result

Provides whois information like creation_date

- **Version 2** of domain_stats has freq_server built-in

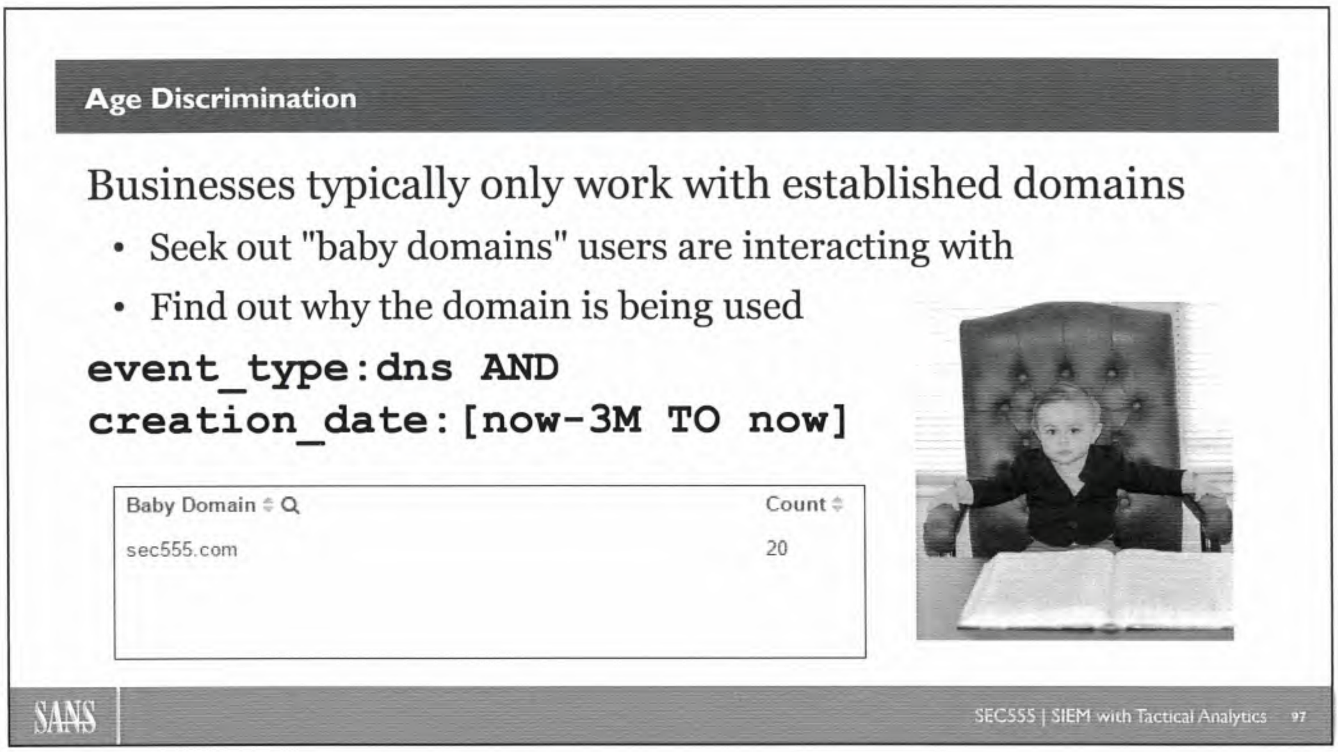
domain_stats.py

Providing domain creation_date lookups and even the Alexa top 1 million or Cisco top 1 million lookups at scale is a challenging problem. In order to combat this, Mark Baggett created domain_stats.py.[1] This script provides a web API for submitting domains to pull back specific results. In order to pull back domain information, /domain is used. If specific field information is required, then /domain/[field_name] is passed such as /domain/creation_date.

To find out if a domain is a member of the top 1 million sites, /alexa is used. Because the top 1 million sites were originally from Amazon Alexa and individuals are using this script in production, Mark has decided to keep /alexa named as is. However, the top 1 million sites from Cisco Umbrella works with this script. Because it is an API, any commercial SIEM can query it.

Reference:

https://github.com/MarkBaggett/domain_stats, https://sec555.com/51



Age Discrimination

Businesses typically only work with established domains

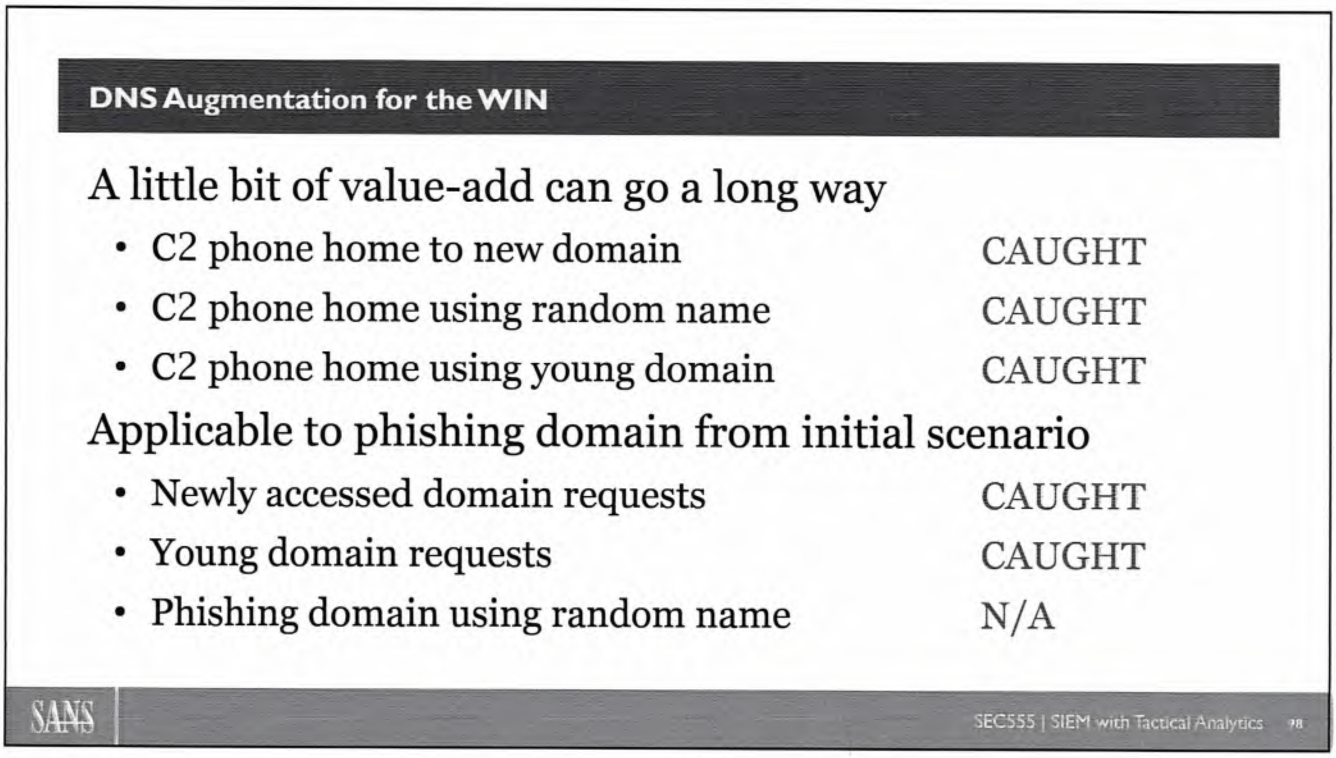
- Seek out "baby domains" users are interacting with
- Find out why the domain is being used

```
event_type:dns AND
creation_date:[now-3M TO now]
```

Baby Domain	Count
sec555.com	20

Age Discrimination

Ideally, the creation date would be used to monitor baby domains continuously. For example, a dashboard or table would be regularly reviewed to inspect why someone may be interacting with a baby domain. This technique works well to find phishing domains, as they are often used within the first three months of creation. This can be used for monitoring all baby domains or can be filtered down to specific use cases such as MX records for specifically looking for emails from new domains.



DNS Augmentation for the WIN

A little bit of value-add can go a long way

• C2 phone home to new domain	CAUGHT
• C2 phone home using random name	CAUGHT
• C2 phone home using young domain	CAUGHT

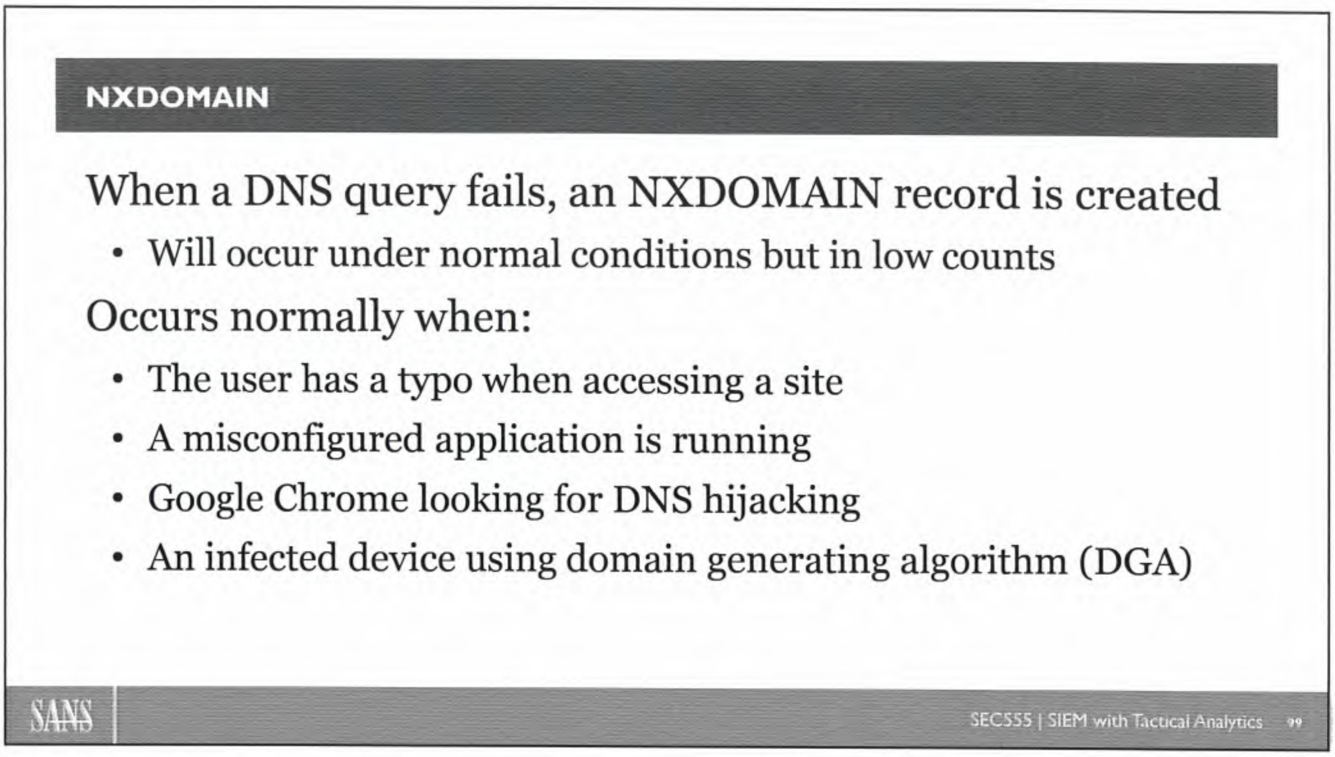
Applicable to phishing domain from initial scenario

• Newly accessed domain requests	CAUGHT
• Young domain requests	CAUGHT
• Phishing domain using random name	N/A

DNS Augmentation for the WIN

In the phishing campaign against John Doe, multiple techniques would have detected the attack if they were used. The malware phoning home would have been detected if you were monitoring for new domains, inspecting random domain names, or monitoring baby domains.

In fact, even the initial phishing email of 1abmeinc.com would have been caught by monitoring new domain names and baby domains.



NXDOMAIN

When a DNS query fails, an NXDOMAIN record is created

- Will occur under normal conditions but in low counts

Occurs normally when:

- The user has a typo when accessing a site
- A misconfigured application is running
- Google Chrome looking for DNS hijacking
- An infected device using domain generating algorithm (DGA)

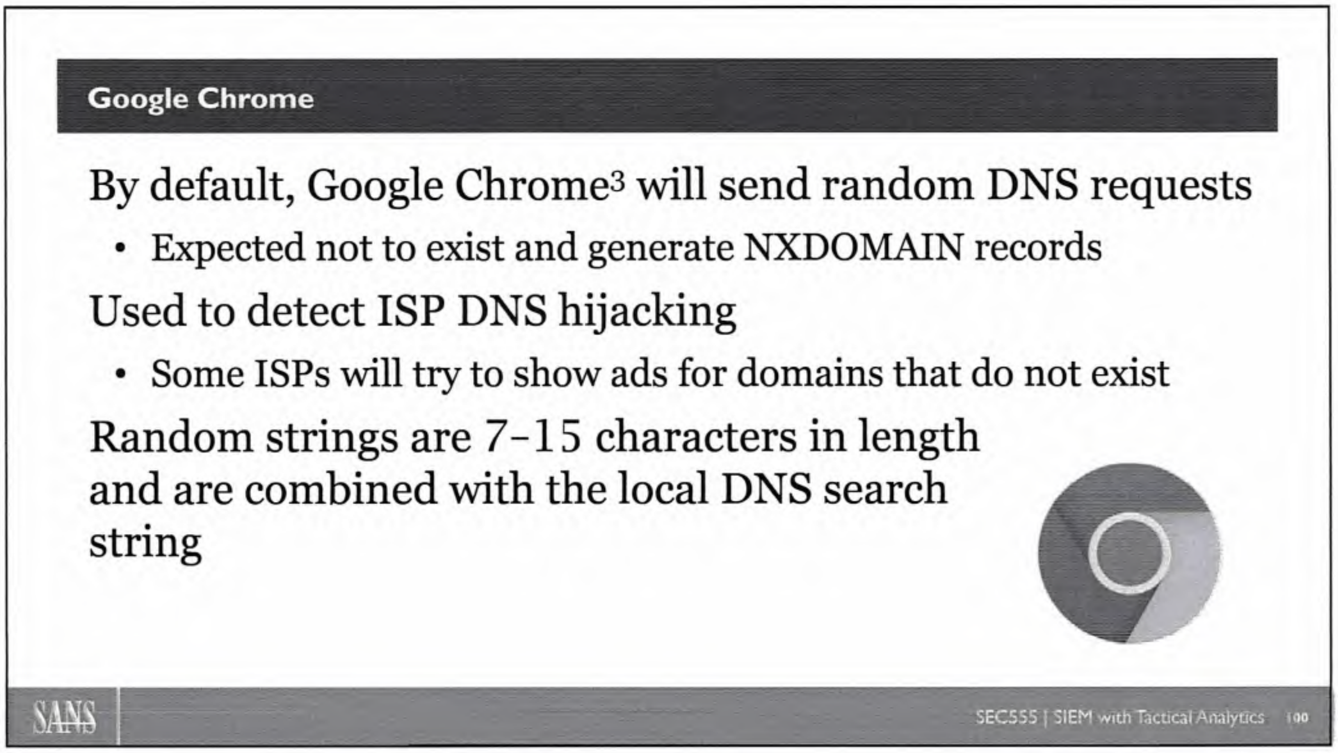
SANS

SEC555 | SIEM with Tactical Analytics 99

NXDOMAIN

When a DNS query is for a name that does not exist or for some reason cannot be found, an NXDOMAIN response is generated. This stands for a non-existent domain. An NXDOMAIN is a normal occurrence but tends to happen at a low frequency. This is most commonly seen due to user error and system misconfigurations.

However, if NXDOMAIN responses begin happening in high occurrences, this usually is a sign of misconfiguration or malware using a domain generating algorithm (DGA). Investigating these is typically quick, and the misconfigurations are easily resolved. For example, if you're using VMware vSphere and you join the hypervisor to the domain, it will start requesting Kerberos SRV records similar to _kerberos-master._udp.test.int with the test.int being the internal domain. By default, this SRV record does not exist, causing an NXDOMAIN response. This occurs many times per second and will light up an NXDOMAIN report.



Google Chrome

By default, Google Chrome[3] will send random DNS requests

- Expected not to exist and generate NXDOMAIN records

Used to detect ISP DNS hijacking

- Some ISPs will try to show ads for domains that do not exist

Random strings are 7-15 characters in length and are combined with the local DNS search string

Google Chrome

Another legitimate cause of NXDOMAIN requests is Google Chrome. By default, when Google Chrome is first launched, it will generate three HTTP HEAD requests to a 7- to 15-character random string using letters a–z only. In DNS, this often shows up as a random string with the local search string appended to it. These, by intention, are supposed to fail and respond with NXDOMAIN responses. However, if they return IP addresses instead, it means the internet provider is performing DNS hijacking to serve up ads in place of failed DNS searches.

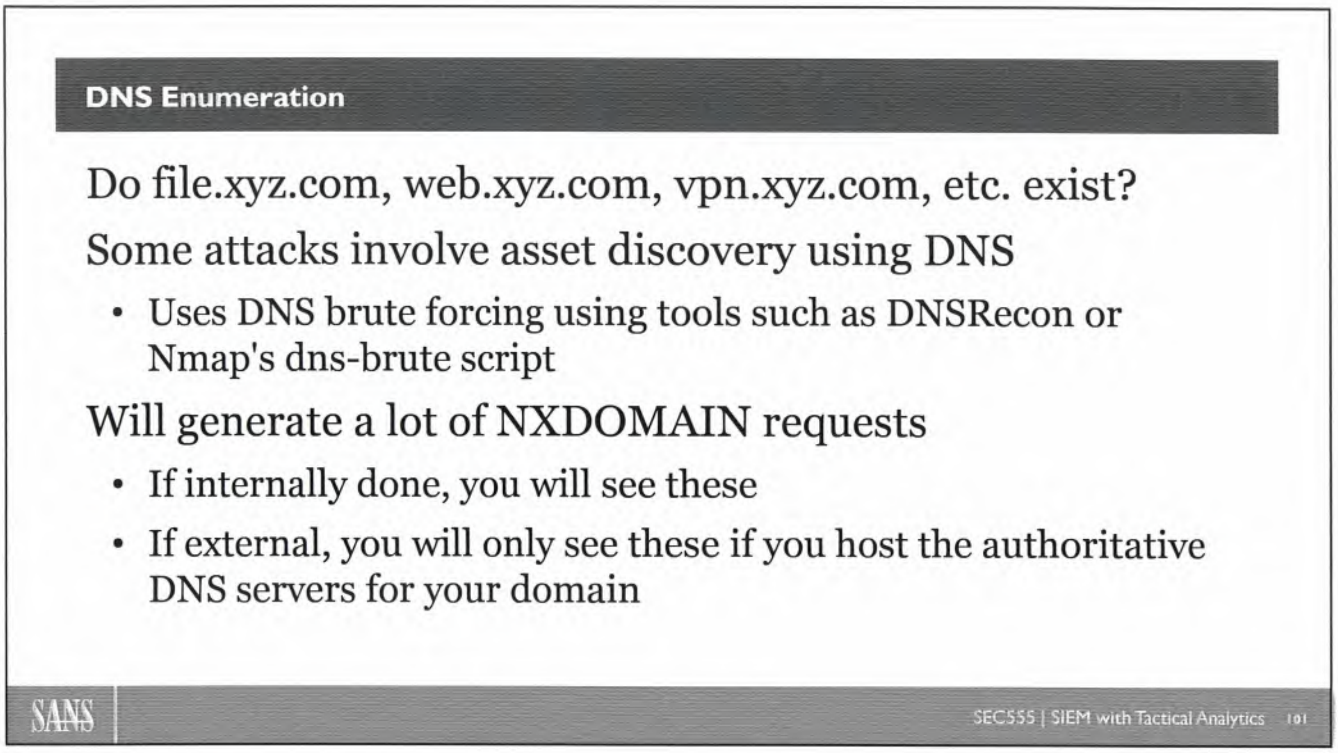
Fortunately, these are tied to whatever the local search domain is, so it is fairly easy to filter them out. Google does not see this behavior as an issue, so they have declined to add the capability to disable this feature.

References:

https://www.google.com/chrome, https://sec555.com/52

https://blog.apnic.net/2020/08/21/chromiums-impact-on-root-dns-traffic/

https://www.techtimes.com/articles/252002/20200824/googles-anti-hijacking-tool-blamed-for-50-of-root-traffic.htm



DNS Enumeration

Do file.xyz.com, web.xyz.com, vpn.xyz.com, etc. exist?

Some attacks involve asset discovery using DNS

- Uses DNS brute forcing using tools such as DNSRecon or Nmap's dns-brute script

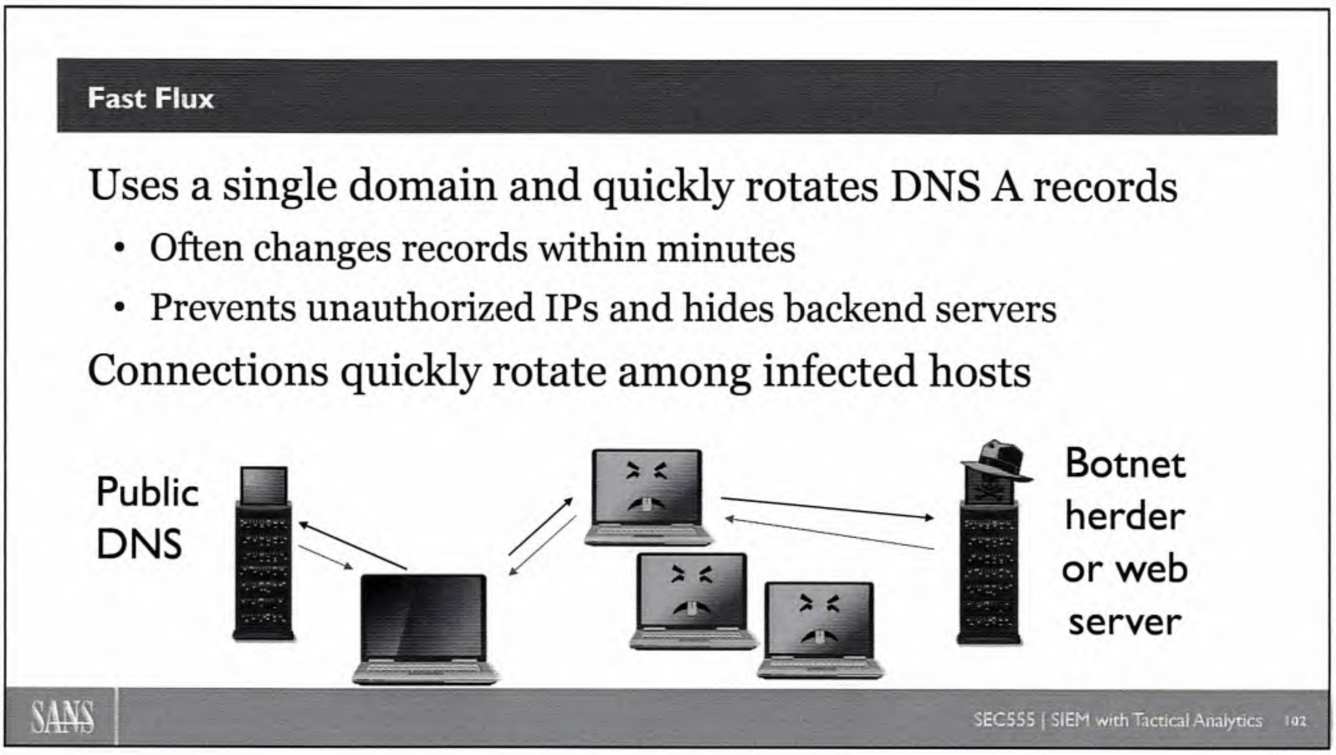
Will generate a lot of NXDOMAIN requests

- If internally done, you will see these
- If external, you will only see these if you host the authoritative DNS servers for your domain

DNS Enumeration

Sometimes, attackers will use DNS to discover what assets are available for attacking. For example, they may request the names file, Web, VPN, etc. against your external facing domains. By doing so, they can discover if you have any external facing servers such as a VPN appliance. Unfortunately, if this method of discovery is used externally, you would not be able to detect it unless the authoritative name server for your domain is maintained by your organization and capable of generating query logs.

On the flip side, if an attacker lands an initial foothold on your network, he or she may use DNS internally to discover assets. If so, this will generate a burst of NXDOMAIN requests. This is not a common practice for modern malware or adversaries. However, many tools, such as Nmap, default to using DNS servers for lookup unless explicitly disabled. This means that if an attacker or malware makes a mistake, a lot of DNS traffic and NXDOMAIN responses will be generated.



Fast Flux

Uses a single domain and quickly rotates DNS A records

- Often changes records within minutes
- Prevents unauthorized IPs and hides backend servers

Connections quickly rotate among infected hosts

Public DNS

Botnet herder or web server

SANS

SEC555 | SIEM with Tactical Analytics 102

Fast Flux

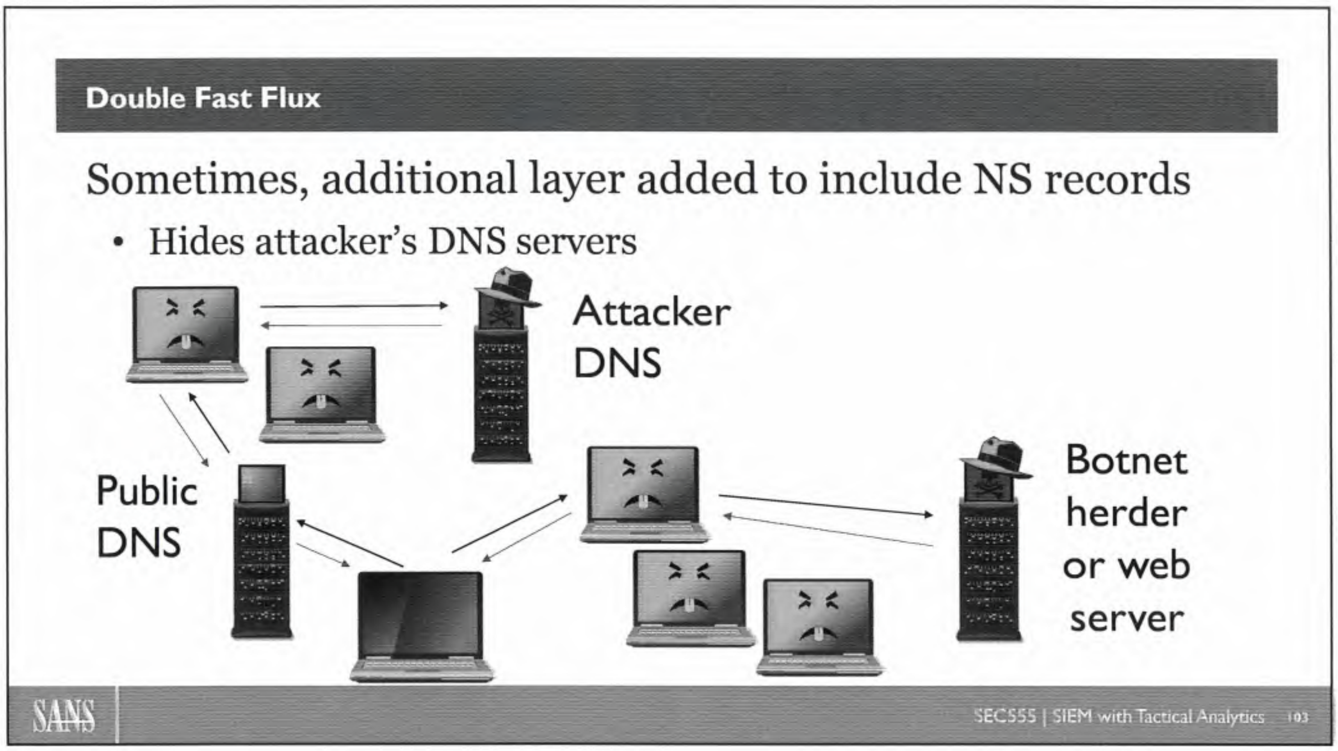
Since IP deny lists are commonly utilized, malware authors adapted and began using DNS to bypass filters. One such method is using fast flux, where a domain exists, but the IP addresses associated with the domain are constantly changing. This could be a single IP per A record that is changing every minute to a couple of minutes. Typically, this will involve multiple IP addresses associated with a DNS record that constantly rotate.

In the case of fast flux, the IP addresses associated with the changing DNS records are infected machines. The malware on these machines allows for self-registering against the fast flux system's domain. These machines then relay traffic to the backend attacker services. By using infected computers in this manner, it hides and protects the attacker's backend servers and services.

A study by Arbor Networks reports that, on average, fast flux domain lifespans are one week.[1] Because the IP addresses are rotating so quickly, this can also be difficult for network firewalls that can block based on DNS names to keep up. Eventually, defenders figured out a method to block fast flux attacks by blocking traffic to the IP addresses of a fast flux's designated authoritative name server. However, attackers then increased their game and came up with double fast flux.

Reference:

https://www.netscout.com/product/atlas-intelligence-feed-aif, https://sec555.com/53



Double Fast Flux

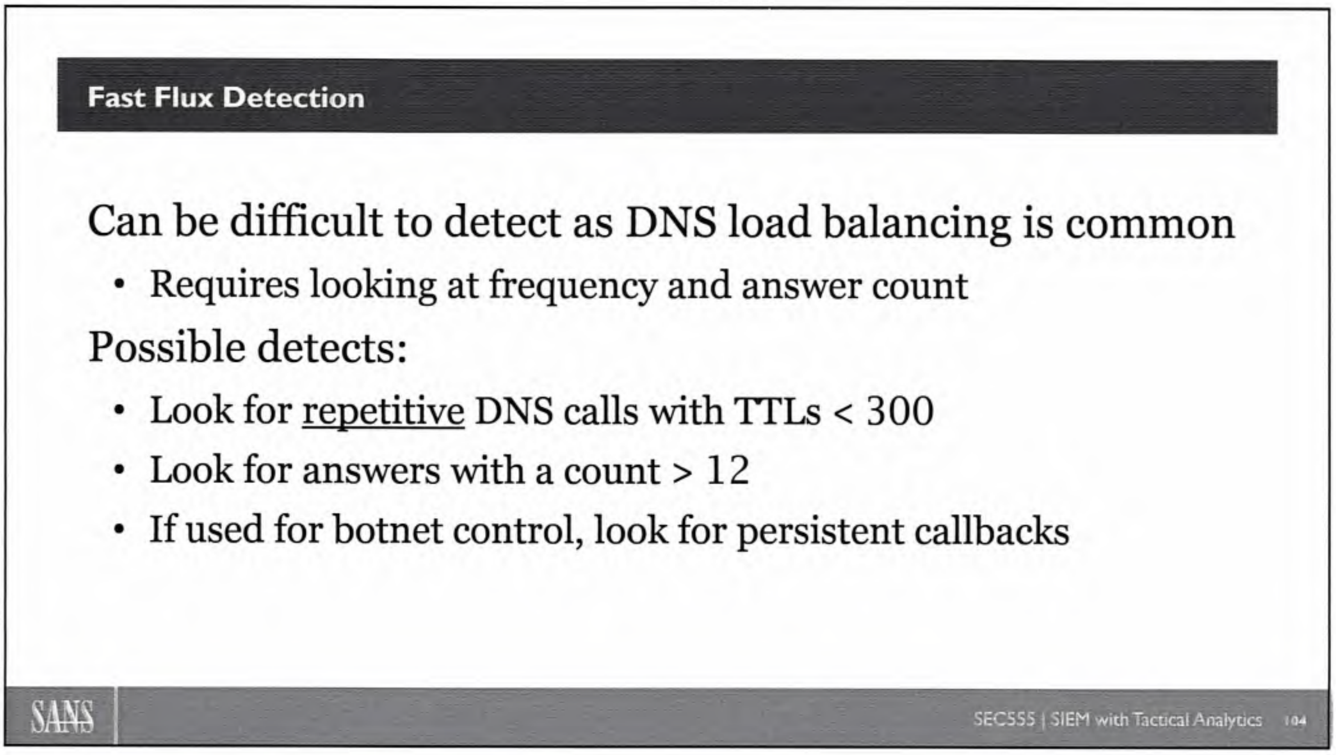
Sometimes, additional layer added to include NS records

- Hides attacker's DNS servers

Double Fast Flux

Double fast flux is fast flux but with the inclusion of additional layers. Typically, this extra complexity uses infected machines to handle A records like single fast flux and then another set of infected machines to handle name server records. Therefore, some machines are acting as DNS proxies to hide the attacker's DNS servers from being discovered, and another set of systems are acting as proxies for other services such as an attacker's web server or botnet control servers.

Again, defenders discovered a way to combat this: Use internal DNS servers to act as the authoritative name servers for the fast flux domain to prevent future connections. This, then, led attackers to switch to using domain generation algorithms (DGA). Regardless, today both DGA and fast flux are used by malware.



Fast Flux Detection

Can be difficult to detect as DNS load balancing is common

- Requires looking at frequency and answer count

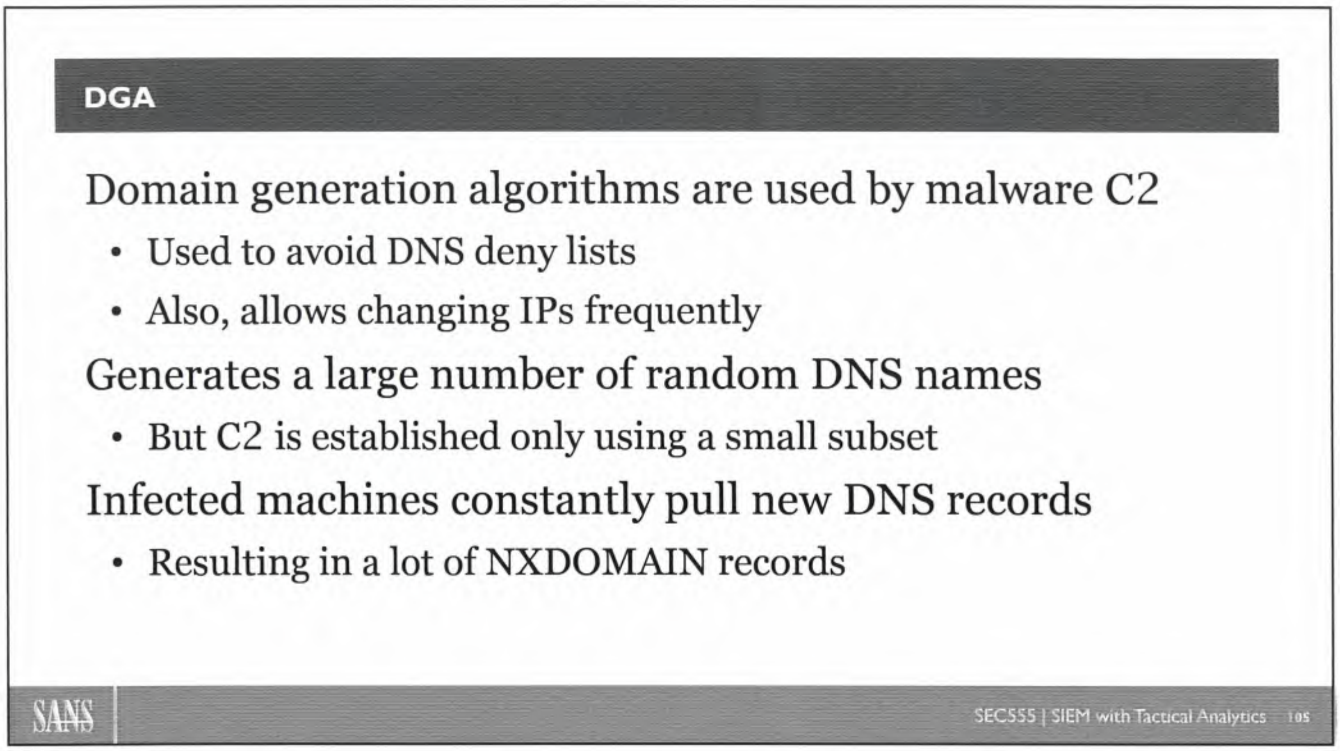
Possible detects:

- Look for repetitive DNS calls with TTLs < 300
- Look for answers with a count > 12
- If used for botnet control, look for persistent callbacks

Fast Flux Detection

In many ways, fast flux can look like normal DNS traffic as it closely resembles DNS load balancing or the use of content delivery networks. Both have widespread use on the internet. The main difference is the repetitive DNS queries and the possible use of higher-than-normal answer counts. While it is common for popular sites, such as social media, to have low TTLs and multiple calls throughout the day, it does not generate hundreds of DNS queries and responses and does not do so 24/7 as a C2 botnet commonly would.

Therefore, monitoring the number of DNS queries by source IP address, DNS queries with more than 12 answers provided, or monitoring DNS connections for persistent connections are all methods of detecting fast flux.



DGA

Domain generation algorithms are used by malware C2

- Used to avoid DNS deny lists
- Also, allows changing IPs frequently

Generates a large number of random DNS names

- But C2 is established only using a small subset

Infected machines constantly pull new DNS records

- Resulting in a lot of NXDOMAIN records

SANS

SEC555 | SIEM with Tactical Analytics 105

DGA

More modern use of DNS for hiding is the use of domain generation algorithms by malware. These are used to avoid IP and DNS deny lists and can be extremely difficult to catch. An algorithm is used to generate a large number of possible DNS domains with only a small subset of the domains having a system registered to the domain name. These domains, typically, have high entropy to bypass preventive measures. As stated before, randomness is an attacker strength that can become their weakness when used with frequency analysis.

Infected machines calling out to a DGA-protected system will generate a large amount of NXDOMAIN records.

References:

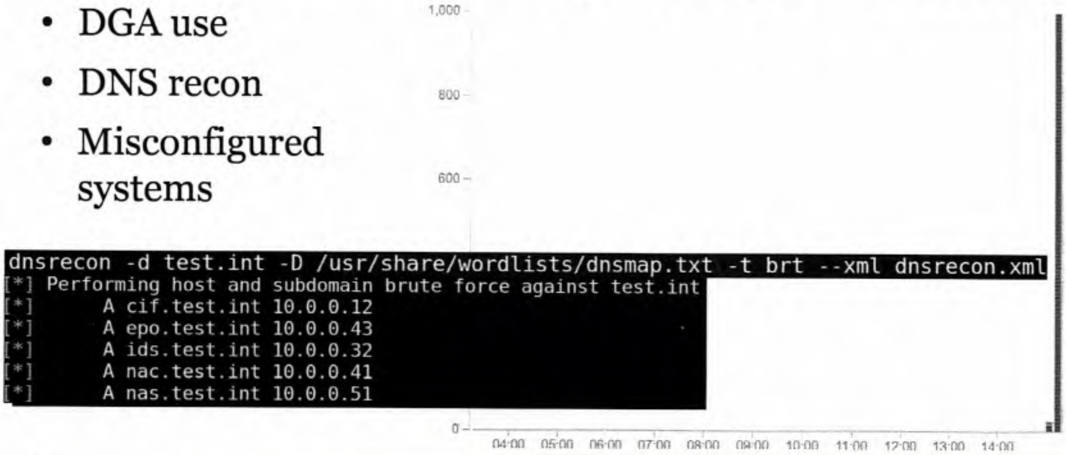
https://resources.infosecinstitute.com/domain-generation-algorithm-dga, https://sec555.com/54

https://isc.sans.edu/forums/diary/Detecting+Random+Finding+Algorithmically+chosen+DNS+names+DGA/19893, https://sec555.com/55

NXDOMAIN Detection

Monitoring NXDOMAIN responses by IP will detect:

- DGA use
- DNS recon
- Misconfigured systems



```
dnsrecon -d test.int -D /usr/share/wordlists/dnsmap.txt -t brt --xml dnsrecon.xml
[*] Performing host and subdomain brute force against test.int
[*]      A cif.test.int 10.0.0.12
[*]      A epo.test.int 10.0.0.43
[*]      A ids.test.int 10.0.0.32
[*]      A nac.test.int 10.0.0.41
[*]      A nas.test.int 10.0.0.51
```

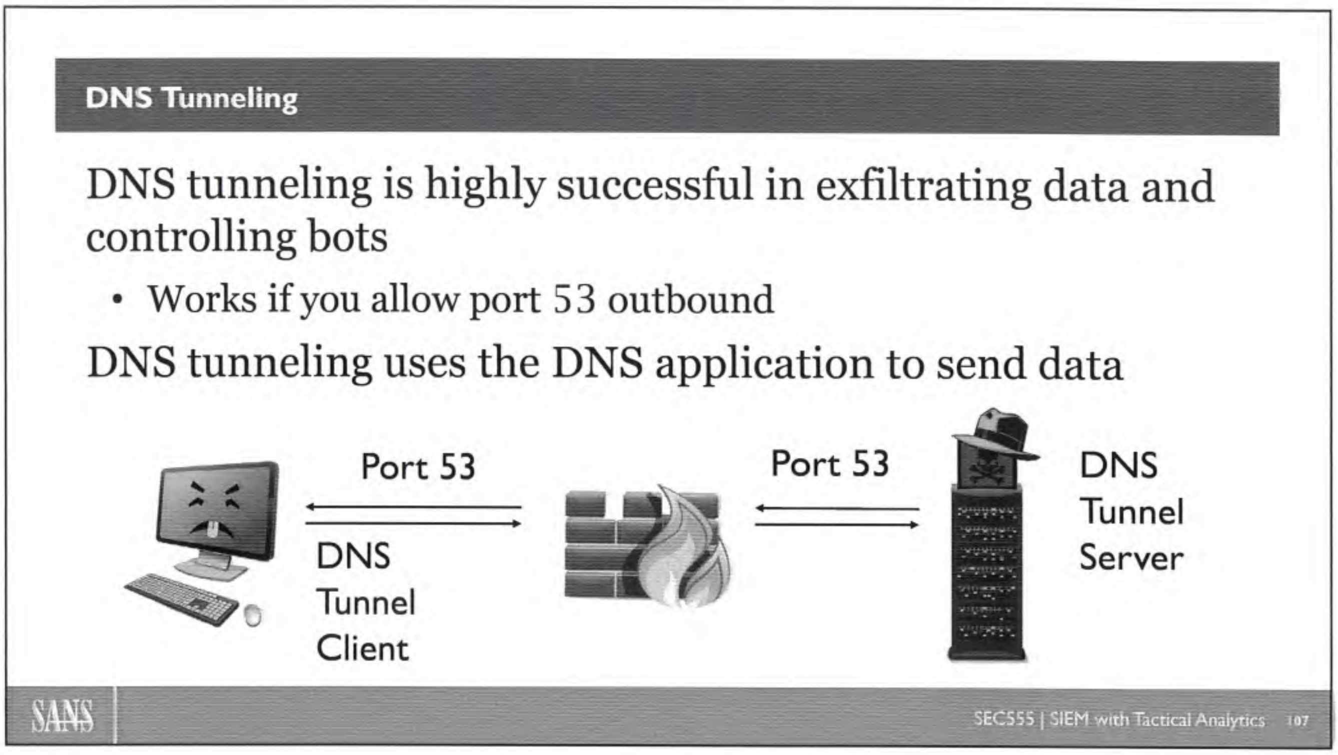
NXDOMAIN Detection

By simply monitoring NXDOMAIN responses by IP addresses, you will be able to uncover the use of DGA, DNS recon, and misconfigured systems. Again, NXDOMAIN is a normal occurrence, but the frequency of the occurrence is typically low. In this diagram, you can see that 10.0.1.6 (purple line) had a huge spike compared to other internal systems.

To simulate this, the tool dnsrecon[1] was used against the internal domain of test.int to brute force discover internal assets. To simply monitor NXDOMAIN responses, consider setting a threshold, such as show all systems that have generated more than 50 NXDOMAIN responses within a given period.

Reference:

https://github.com/darkoperator/dnsrecon, https://sec555.com/56



DNS Tunneling

DNS tunneling is highly successful in exfiltrating data and controlling bots

- Works if you allow port 53 outbound

DNS tunneling uses the DNS application to send data

Port 53

DNS Tunnel Client

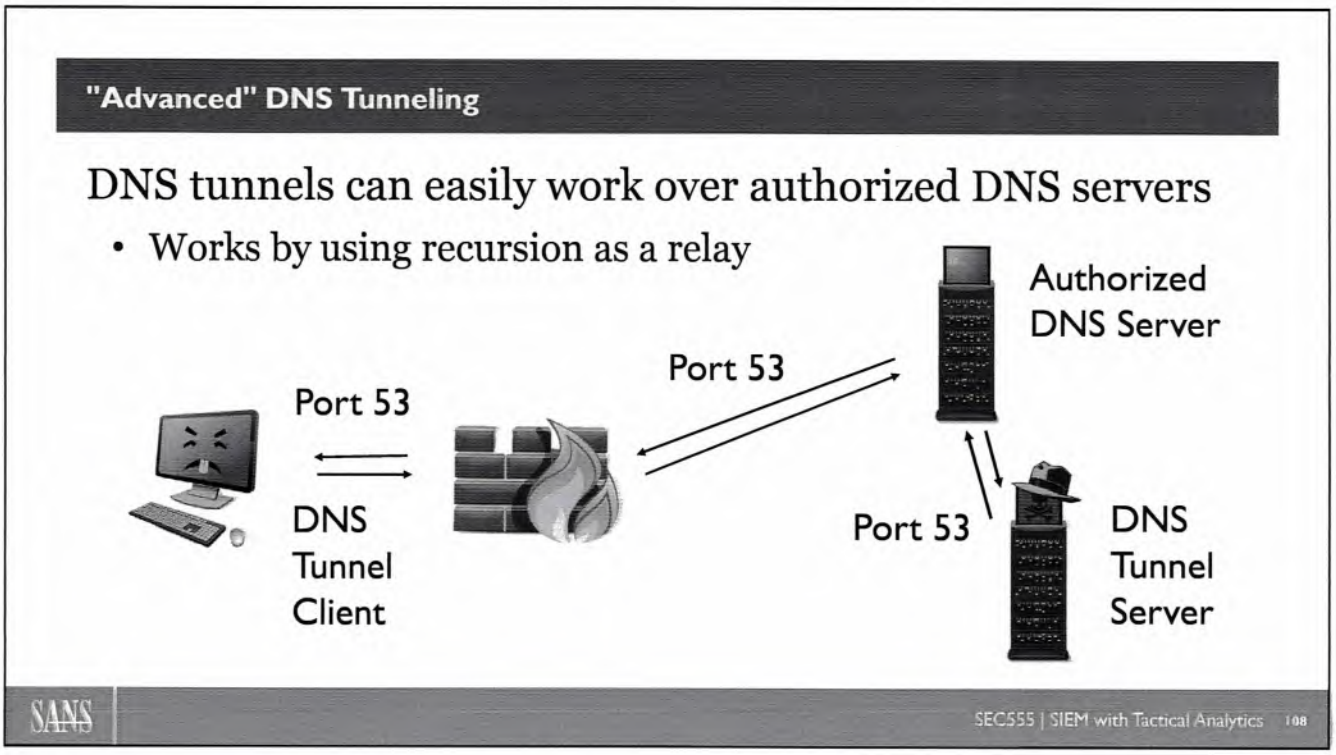
Port 53

DNS Tunnel Server

DNS Tunneling

In this author's opinion, DNS tunneling is one of the coolest methods to control systems or exfiltrate data covertly. Unfortunately, it is one of the methods that almost every organization fails to detect. In its simplest form, DNS tunneling uses the DNS protocol directly to an attacker's external DNS server to communicate.

This is simple to protect against and detect. Simply block all DNS outbound traffic, except authorized DNS servers to authorized external DNS servers, and then monitor for attempts to use DNS to unauthorized systems.



"Advanced" DNS Tunneling

DNS tunnels can easily work over authorized DNS servers

- Works by using recursion as a relay

Authorized DNS Server

Port 53

Port 53

DNS Tunnel Client

Port 53

DNS Tunnel Server

SANS

SEC555 | SIEM with Tactical Analytics 108

"Advanced" DNS Tunneling

More advanced use of DNS tunneling works if an organization allows external DNS recursion, even if only to authorized DNS servers. Under normal conditions, an internal DNS server will reach out to an authorized DNS server such as Google's 8.8.8.8 to perform recursive DNS lookups. This authorized system will then track down the name server responsible for a given domain and then interact with it for DNS queries it is responsible for. Because of this, malware can use DNS tunneling through legitimate and authorized servers. Put a different way, if you allow DNS externally, this attack will work.

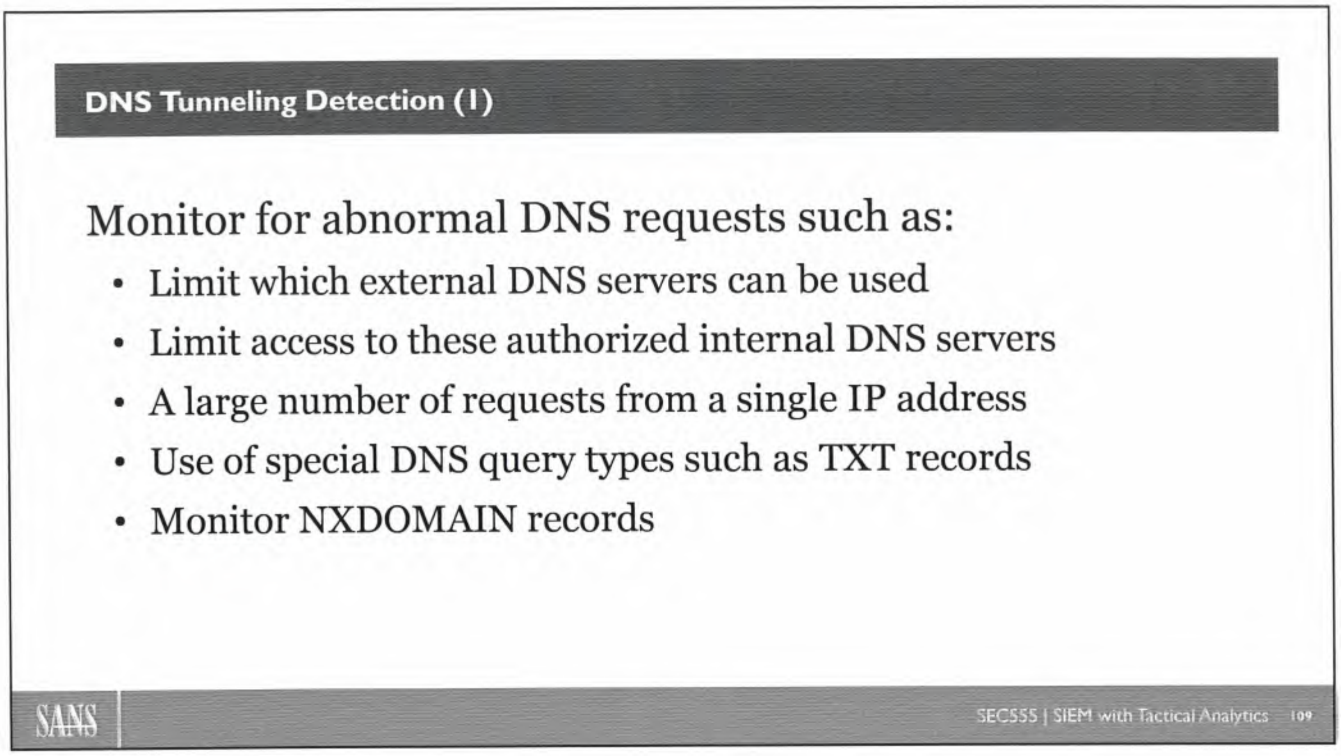
Lenny Zeltser has an awesome article demonstrating both DNS tunneling techniques and how to perform them. The most common tools available to create a DNS tunnel are dnscat2 and iodine.[3] They are commonly used by penetration testers to demonstrate an organization's lack of detection around DNS tunneling.

References:

https://zeltser.com/c2-dns-tunneling, https://sec555.com/57

https://github.com/iagox86/dnscat2, https://sec555.com/58

https://code.kryo.se/iodine, https://sec555.com/59



DNS Tunneling Detection (1)

Monitor for abnormal DNS requests such as:

- Limit which external DNS servers can be used
- Limit access to these authorized internal DNS servers
- A large number of requests from a single IP address
- Use of special DNS query types such as TXT records
- Monitor NXDOMAIN records

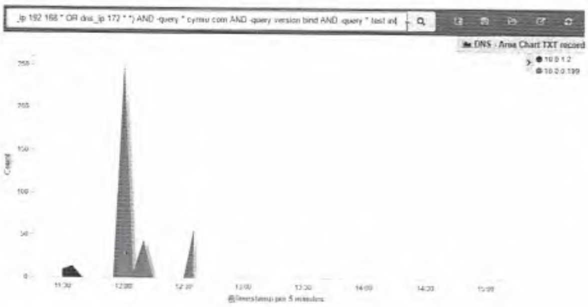
DNS Tunneling Detection (1)

Given that most organizations cannot detect DNS tunneling, you would figure it is difficult to find. However, it is not. Simply monitoring for large volumes of DNS requests from a given IP or monitoring NXDOMAIN responses by IP will catch DNS tunneling. Also, since the goal is to communicate or send large amounts of data, DNS tunneling is often performed over DNS TXT records. Monitoring the number of TXT records requested by a given source IP is a quick way to detect tunneling.

Ideally, organizations would limit external DNS from authorized internal servers to authorized external servers and then establish monitoring. This, combined with DNS query logging, will catch misuse of DNS.

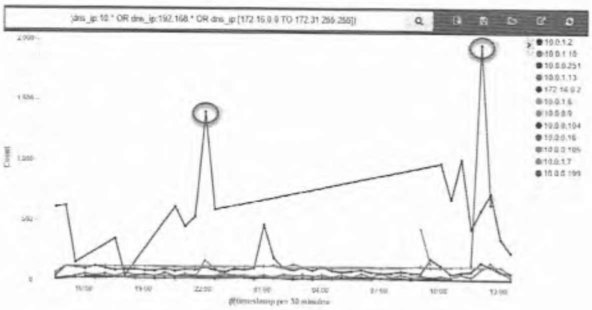
DNS Tunneling Detection (2)

Monitoring TXT queries



```
_ip 192 168 * OR dns_ip 172 * *) AND -query * cymru com AND -query version bind AND -query * test in(
```

Monitoring # of Queries



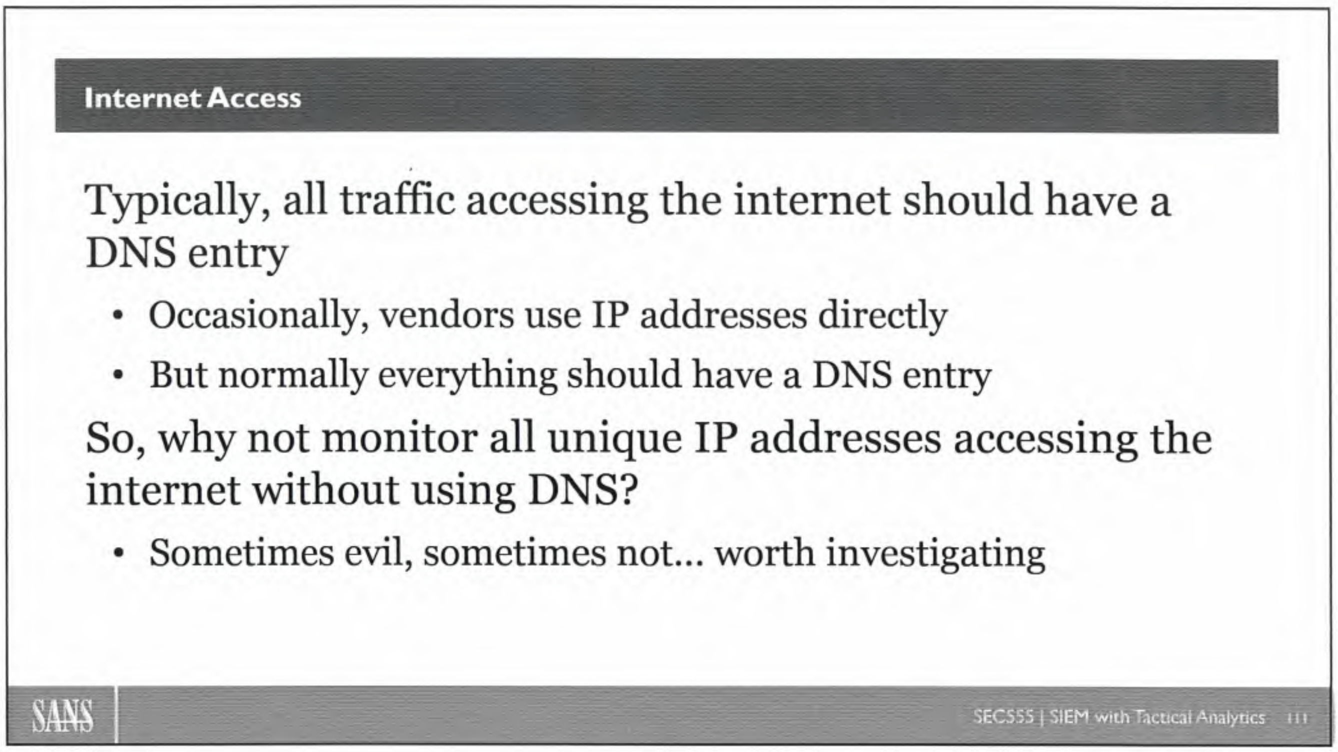
```
(dns_ip:10.* OR dns_ip:192.168.* OR dns_ip:[172.16.0.0 TO 172.31.255.255])
```

DNS Tunneling Detection (2)

On the left is an area chart visualizing DNS TXT record requests by IP address. Two IP addresses make the chart: 10.0.1.2 with a small number of TXT records and 10.0.0.199 with a large number. In reality, 10.0.1.2 was a system attempting to establish a DNS tunnel, but the request was getting mangled, so it failed. 10.0.0.199 was the second attempt to establish a DNS tunnel. It succeeded and transferred some data as well as established a shell connection back to 10.0.0.199. Put simply, TXT monitoring works.

Note that for this to work, you may need to exclude certain domains such as your client endpoint suite's domain. In this example, cymru.com was excluded as Security Onion uses TXT record requests to cymru.com to look for known bad file hashes.

On the right is a line chart plotting out the IP addresses with the highest number of external DNS requests. In this diagram, both 10.0.1.2 and 10.0.0.199 stand out. This is because internal systems tend to make less than a couple of hundred external DNS requests within a few hours. If needed, you could limit this to client networks or specific legs of your environment.



Internet Access

Typically, all traffic accessing the internet should have a DNS entry

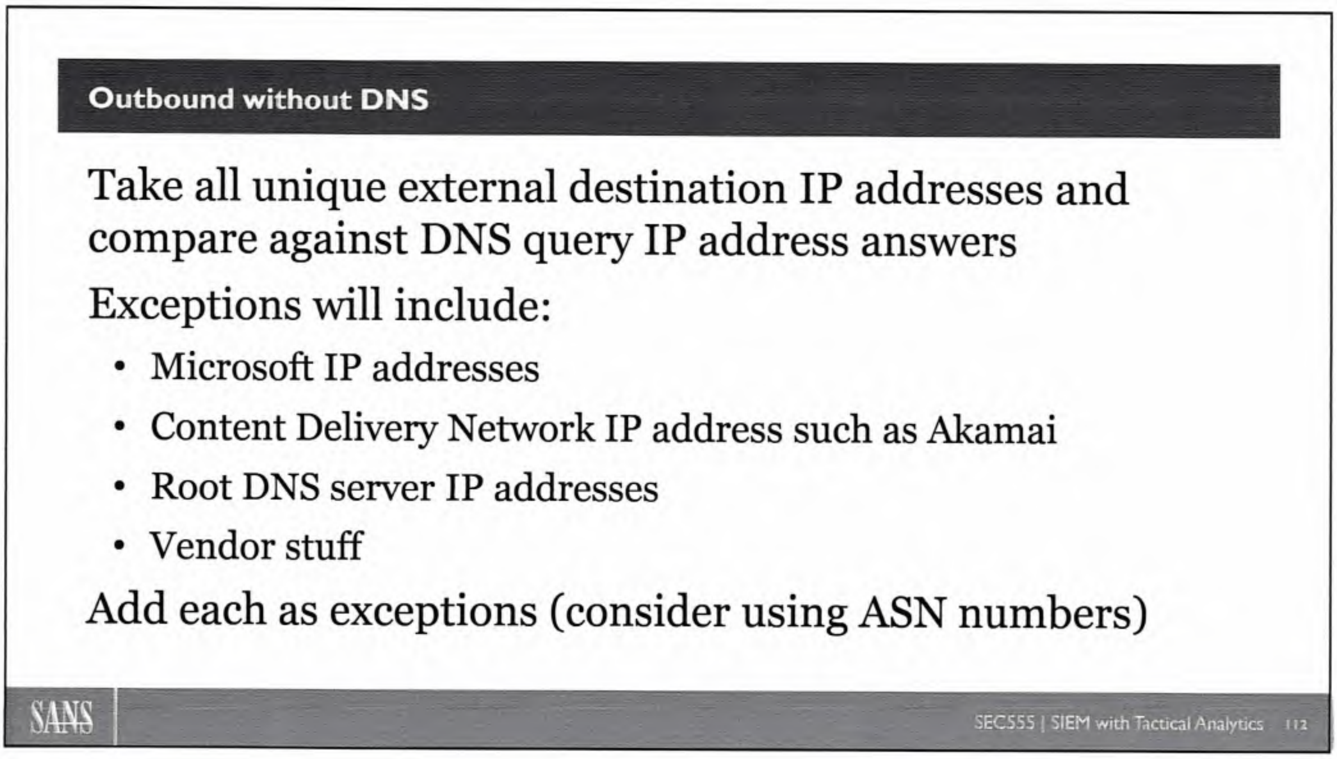
- Occasionally, vendors use IP addresses directly
- But normally everything should have a DNS entry

So, why not monitor all unique IP addresses accessing the internet without using DNS?

- Sometimes evil, sometimes not... worth investigating

Internet Access

Normally, everything that's accessing the internet does so using a name rather than an IP address. Flipping this statement, one would say anything that's accessing the internet using an IP address is abnormal. Vendors occasionally will hard code IP addresses in their hardware/software, and DNS servers themselves are directly communicated to with IP addresses. Outside of this, you often will find malware.



Outbound without DNS

Take all unique external destination IP addresses and compare against DNS query IP address answers

Exceptions will include:

- Microsoft IP addresses
- Content Delivery Network IP address such as Akamai
- Root DNS server IP addresses
- Vendor stuff

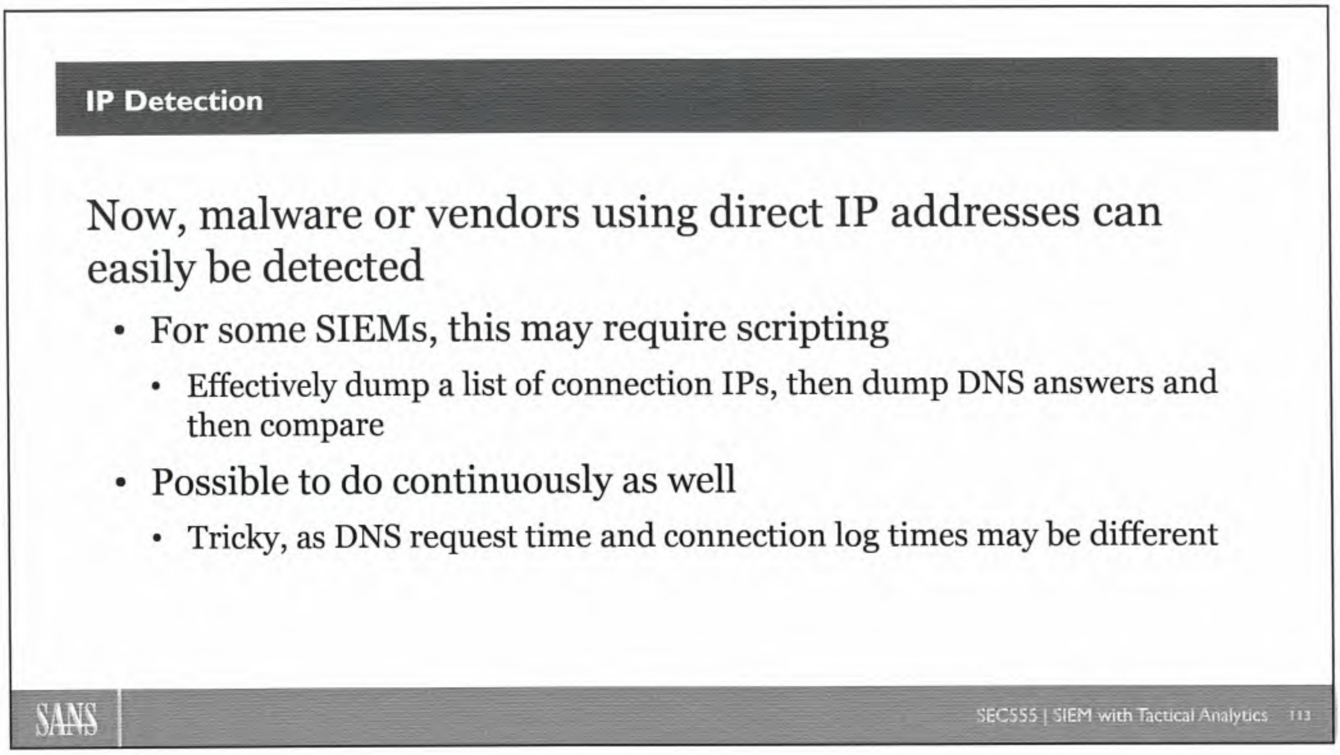
Add each as exceptions (consider using ASN numbers)

Outbound without DNS

Time should be taken to compare the list of external IP address destinations against any IP address answers given by DNS. On a normal network, the difference will include:

- Microsoft IP addresses
- Content Delivery Network (CDN) networks
- Root DNS server IP addresses
- Occasional vendor stuff
- Malware
- Policy violations

It is easy to filter out all of these so that all that is left are the malware and policy violations. To make this filtering easier, use ASN numbers for things like CDN networks.



IP Detection

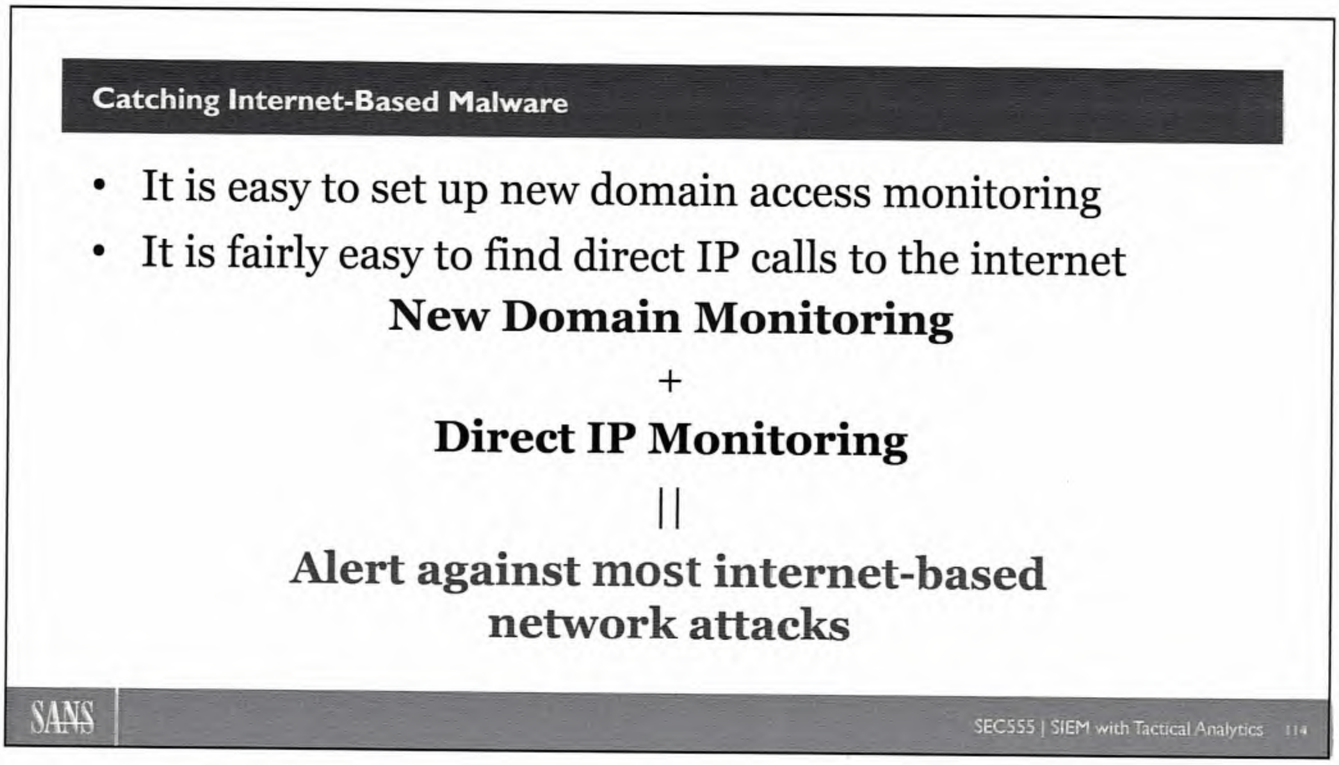
Now, malware or vendors using direct IP addresses can easily be detected

- For some SIEMs, this may require scripting
  - Effectively dump a list of connection IPs, then dump DNS answers and then compare
- Possible to do continuously as well
  - Tricky, as DNS request time and connection log times may be different

IP Detection

This technique sounds easy in principle, but how do you implement it? Many SIEM solutions do not support taking the results of one query and using them to filter against another. In this case, a script needs to be created that first saves a list of all unique external IP addresses and then saves a list of all IP address answers provided by DNS. Then, the script compares each external IP to see if it exists in the text export of DNS answers. This might be something that runs once a day and is manually reviewed.

If your SIEM supports using one query as a filter to another, then this is much simpler. Simply create a table view of any external IP address values that do not exist in DNS query answers.



Catching Internet-Based Malware

- It is easy to set up new domain access monitoring
- It is fairly easy to find direct IP calls to the internet

New Domain Monitoring

+

Direct IP Monitoring

||

Alert against most internet-based network attacks

SANS

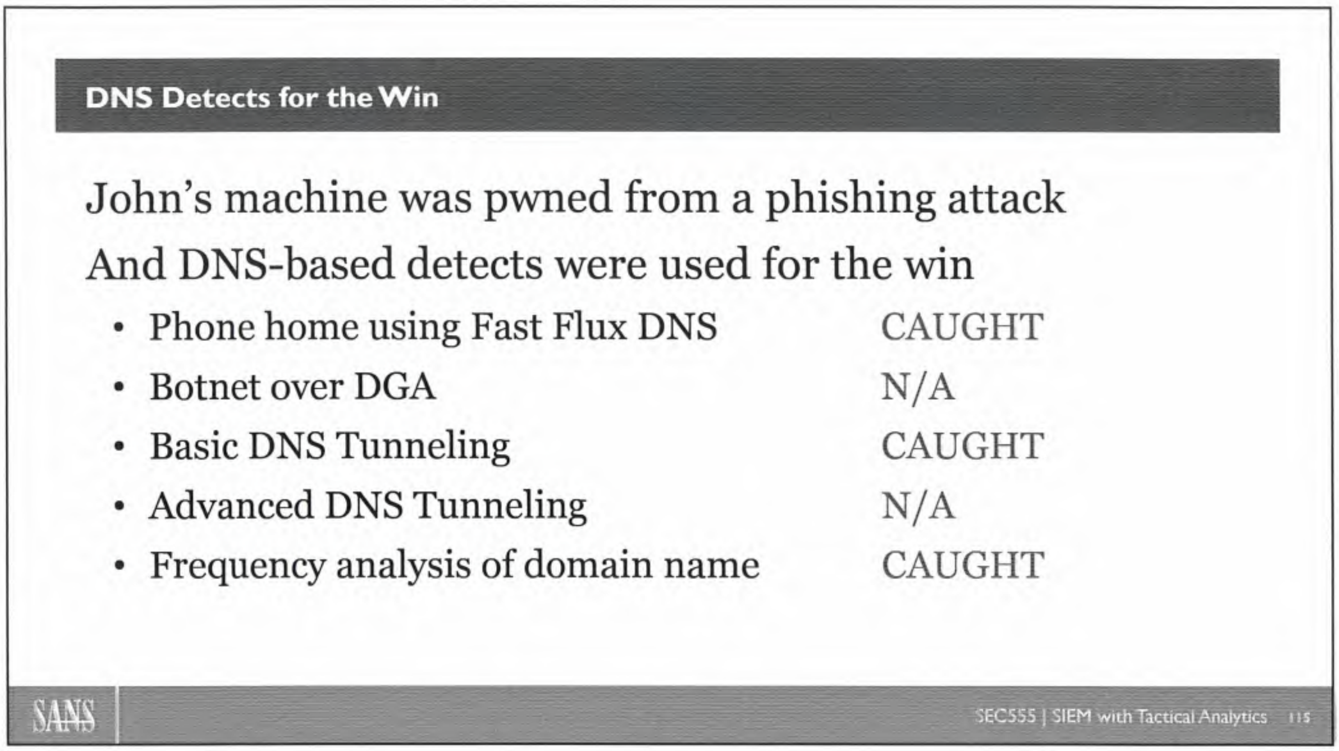
SEC555 | SIEM with Tactical Analytics 114

Catching Internet-Based Malware

If you take the time to implement both new domain monitoring and direct IP monitoring, you will effectively detect most internet-based network attacks. This is because every new connection, whether by DNS domain or by IP address, will end up on a dashboard for review. Does this take labor and time to review? Yes. Is it worth it? Most likely. It is worth trying out at a minimum.

What kind of internet-based attacks would this miss? These two techniques would not see traffic from known good domains. For example, modern malware occasionally will use things like social media sites for command and control. If these are authorized sites, these solutions would not see the attack. Basically, an attacker can use authorized sites to attack or control systems and these two techniques would be oblivious to it.

If you are trying to implement this, you will need to use key filtering information such as ASN, reverse DNS, and top 1 million.



DNS Detects for the Win

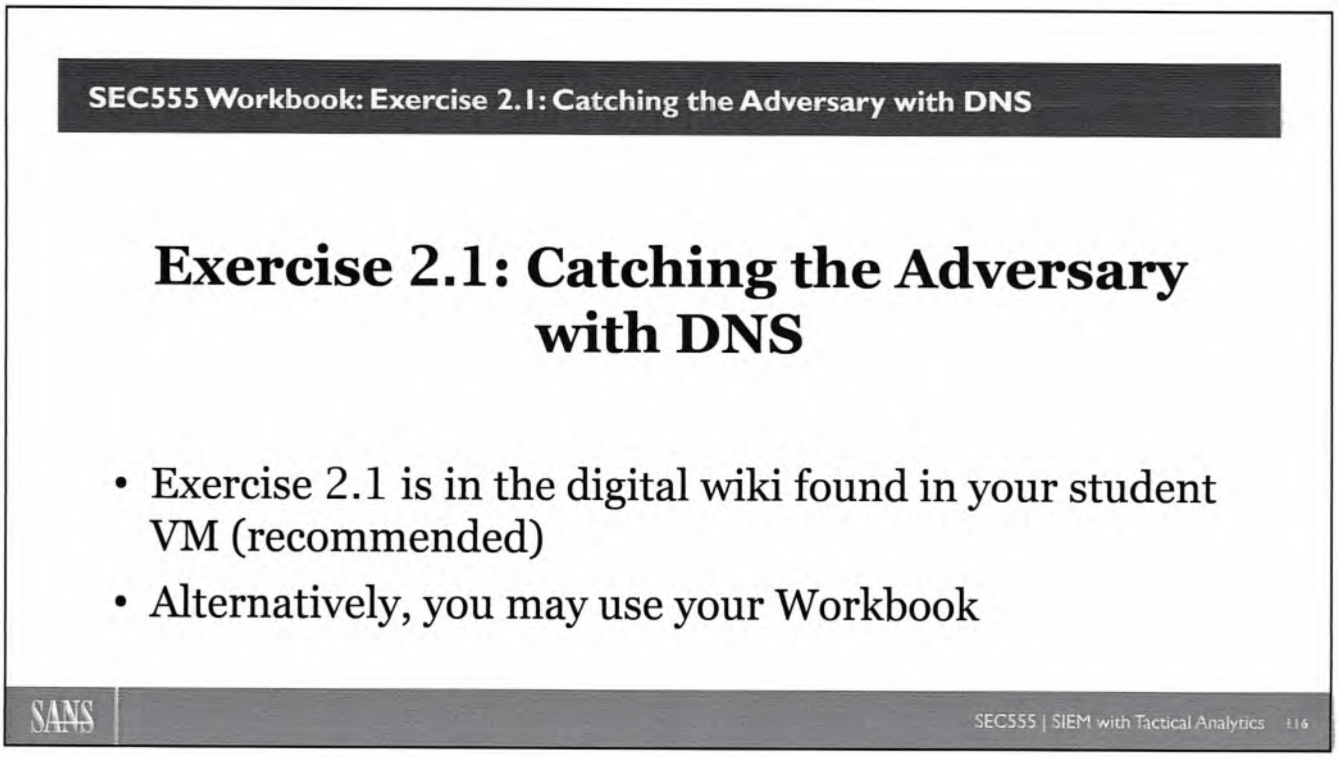
John’s machine was pwned from a phishing attack

And DNS-based detects were used for the win

• Phone home using Fast Flux DNS	CAUGHT
• Botnet over DGA	N/A
• Basic DNS Tunneling	CAUGHT
• Advanced DNS Tunneling	N/A
• Frequency analysis of domain name	CAUGHT

DNS Detects for the Win

Again, taking the phishing example against John Doe, we see that various monitoring techniques would have identified the use of fast flux DNS and the use of DNS tunneling. Had the attack been performed by a more sophisticated attacker using DGA or the more advanced DNS tunnel, they too would have been caught.



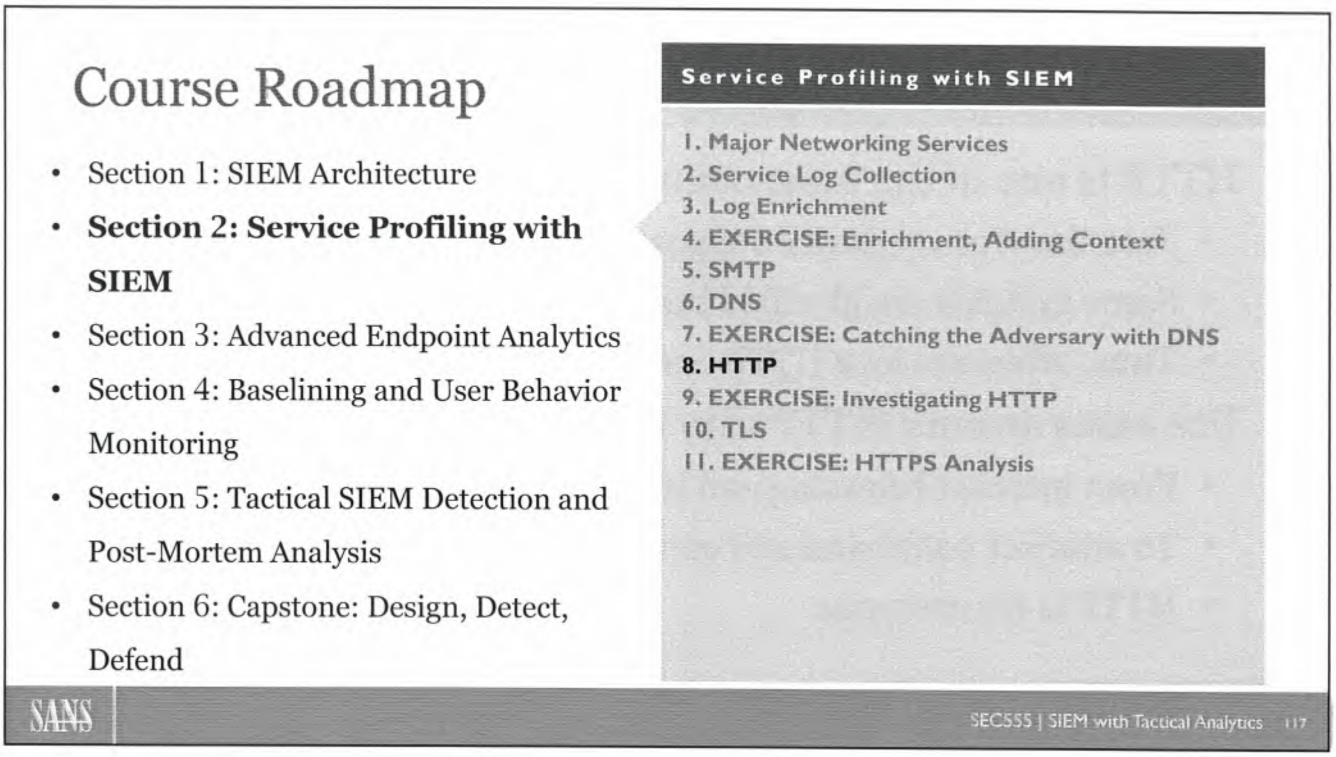
SEC555 Workbook: Exercise 2.1: Catching the Adversary with DNS

Exercise 2.1: Catching the Adversary with DNS

- Exercise 2.1 is in the digital wiki found in your student VM (recommended)
- Alternatively, you may use your Workbook

SEC555 | SIEM with Tactical Analytics 116

This page intentionally left blank.



Course Roadmap

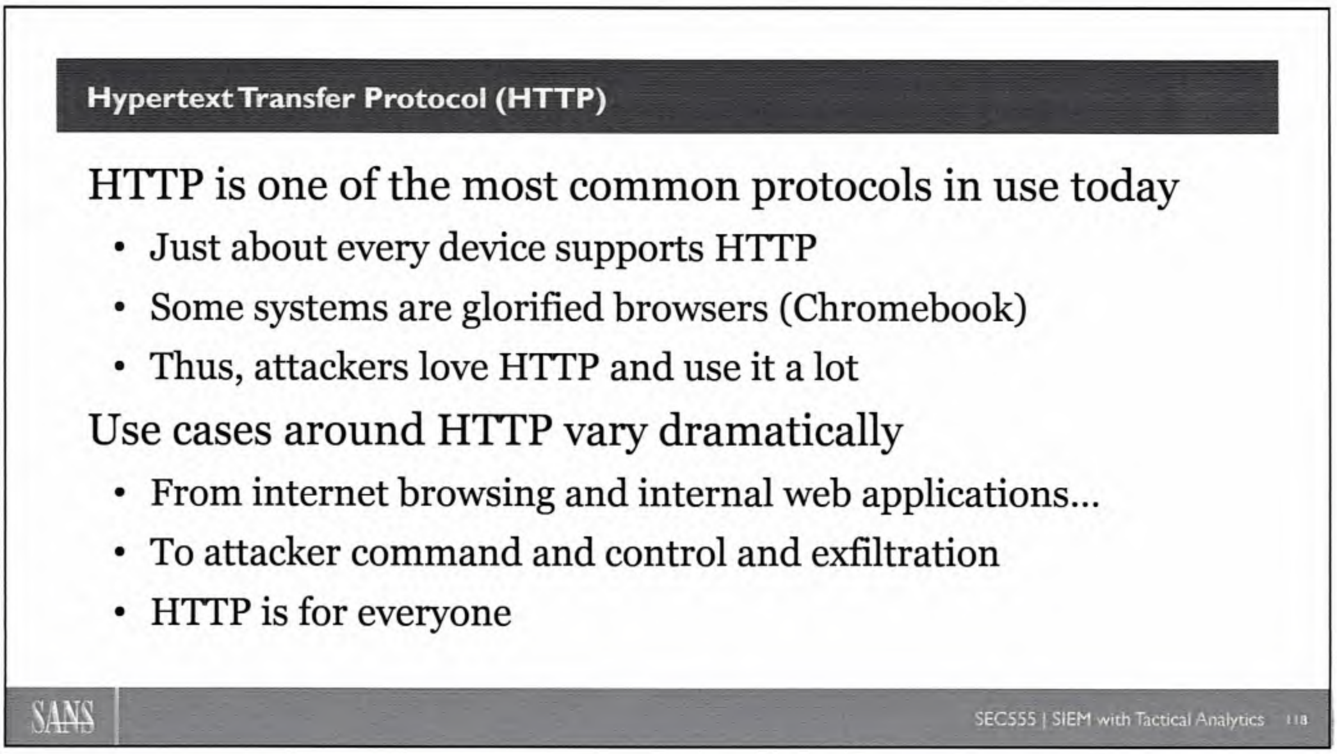
- Section 1: SIEM Architecture
- **Section 2: Service Profiling with SIEM**
- Section 3: Advanced Endpoint Analytics
- Section 4: Baselining and User Behavior Monitoring
- Section 5: Tactical SIEM Detection and Post-Mortem Analysis
- Section 6: Capstone: Design, Detect, Defend

Service Profiling with SIEM

1. Major Networking Services
2. Service Log Collection
3. Log Enrichment
4. EXERCISE: Enrichment, Adding Context
5. SMTP
6. DNS
7. EXERCISE: Catching the Adversary with DNS
8. **HTTP**
9. EXERCISE: Investigating HTTP
10. TLS
11. EXERCISE: HTTPS Analysis

SEC555 | SIEM with Tactical Analytics 117

This page intentionally left blank.



Hypertext Transfer Protocol (HTTP)

HTTP is one of the most common protocols in use today

- Just about every device supports HTTP
- Some systems are glorified browsers (Chromebook)
- Thus, attackers love HTTP and use it a lot

Use cases around HTTP vary dramatically

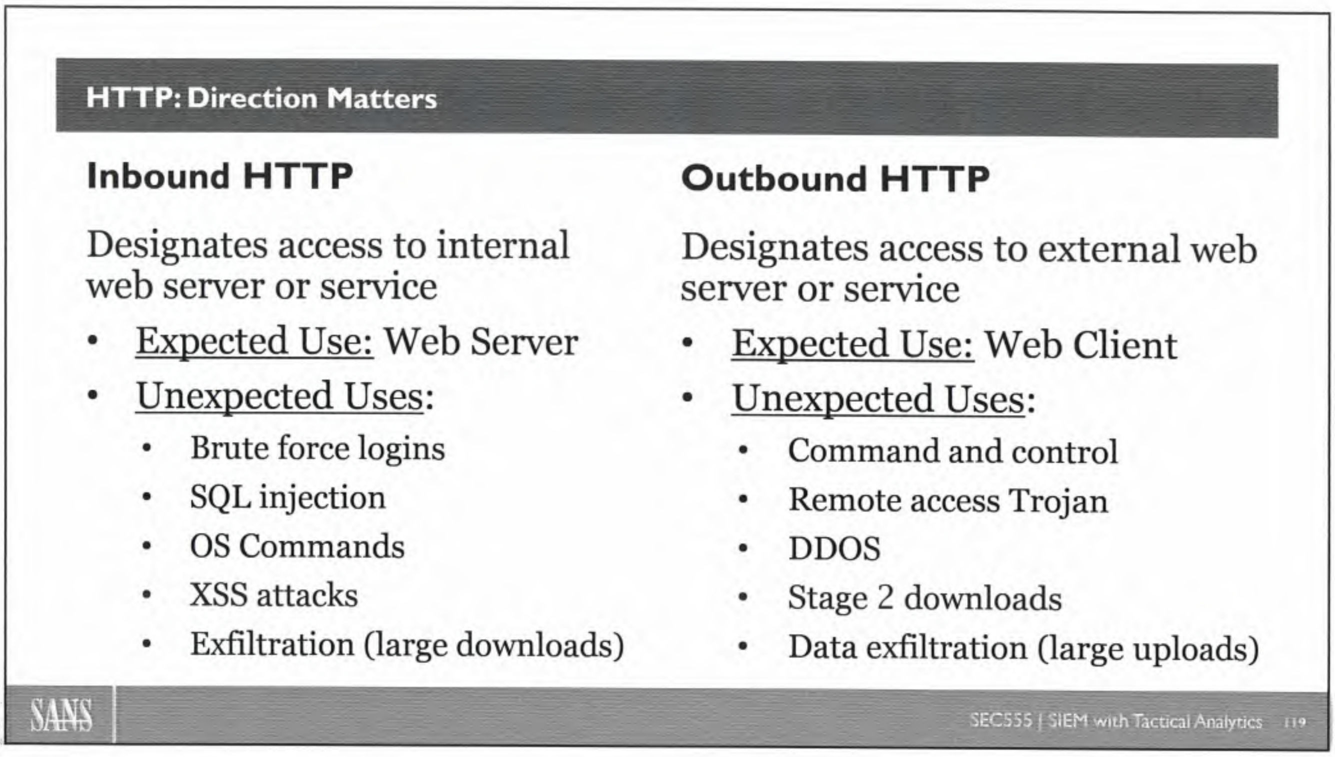
- From internet browsing and internal web applications...
- To attacker command and control and exfiltration
- HTTP is for everyone

Hypertext Transfer Protocol (HTTP)

One of the most important protocols to be familiar with is HTTP. Frankly, its importance stems from the fact that just about every device supports and communicates using HTTP. It has become so commonplace that some devices, such as a Chromebook, operate as glorified internet browsers.

Because of its widespread use, HTTP has become a favorite among attackers. Whether used to download additional tools after initial compromise, used directly for exploitation, or used for a command and control channel, HTTP is commonly used for nefarious purposes.

[1] https://www.google.com/chromebook, https://sec555.com/5a



HTTP: Direction Matters

Inbound HTTP

Designates access to internal web server or service

- Expected Use: Web Server
- Unexpected Uses:
  - Brute force logins
  - SQL injection
  - OS Commands
  - XSS attacks
  - Exfiltration (large downloads)

Outbound HTTP

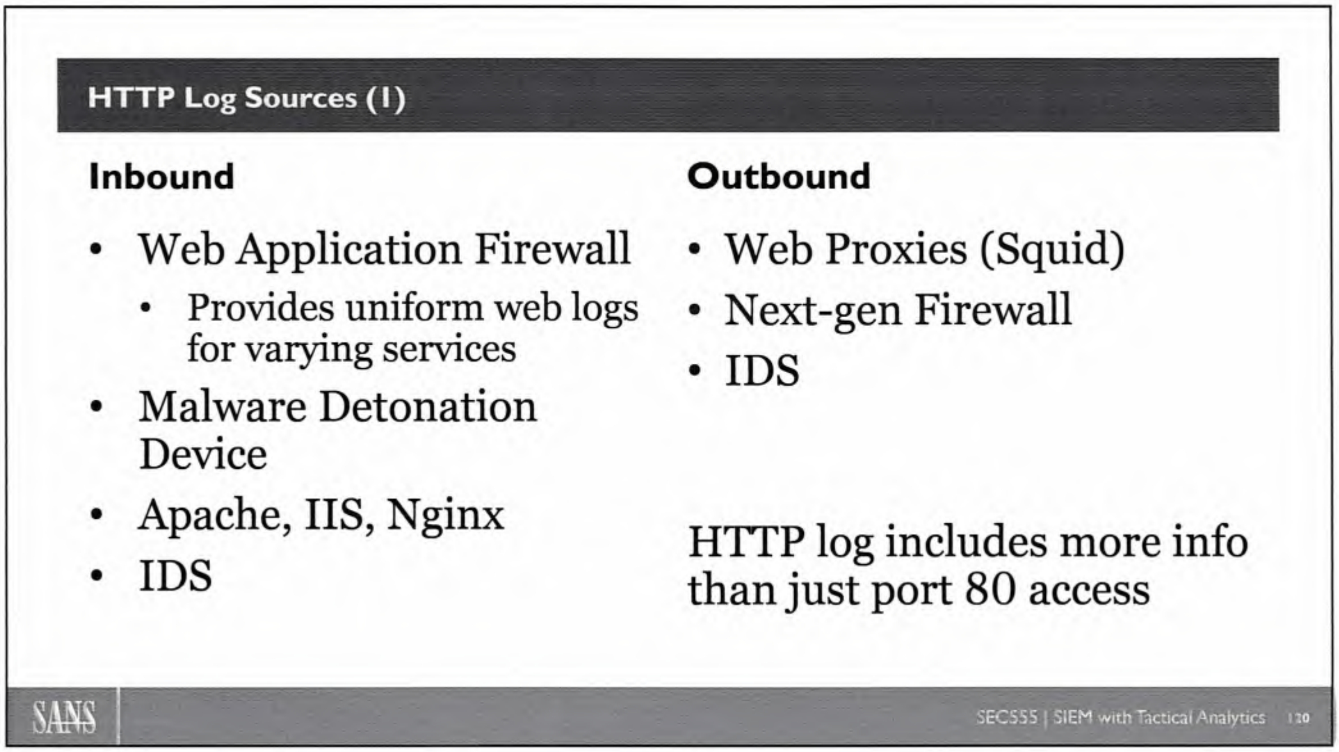
Designates access to external web server or service

- Expected Use: Web Client
- Unexpected Uses:
  - Command and control
  - Remote access Trojan
  - DDOS
  - Stage 2 downloads
  - Data exfiltration (large uploads)

HTTP: Direction Matters

One thing to note is that depending on the direction of an HTTP connection things end up being different. For example, if HTTP is being used from external system inbound into the network, then under normal conditions, this is most likely an HTTP connection into an organization's web server. If, however, an HTTP connection starts from the inside of the network and then reaches out to an external system, it is most likely a web client reaching out to an external web server such as an internal web browser reaching out to a website.

This ultimately means the techniques used around HTTP apply differently based on direction. Techniques revolving around attacks against an internal web server will apply to inbound connections while techniques around internal systems talking to the internet in an unauthorized manner will be applied to outbound connections. Some techniques apply to both inbound and outbound.



HTTP Log Sources (1)

Inbound

- Web Application Firewall
  - Provides uniform web logs for varying services
- Malware Detonation Device
- Apache, IIS, Nginx
- IDS

Outbound

- Web Proxies (Squid)
- Next-gen Firewall
- IDS

HTTP log includes more info than just port 80 access

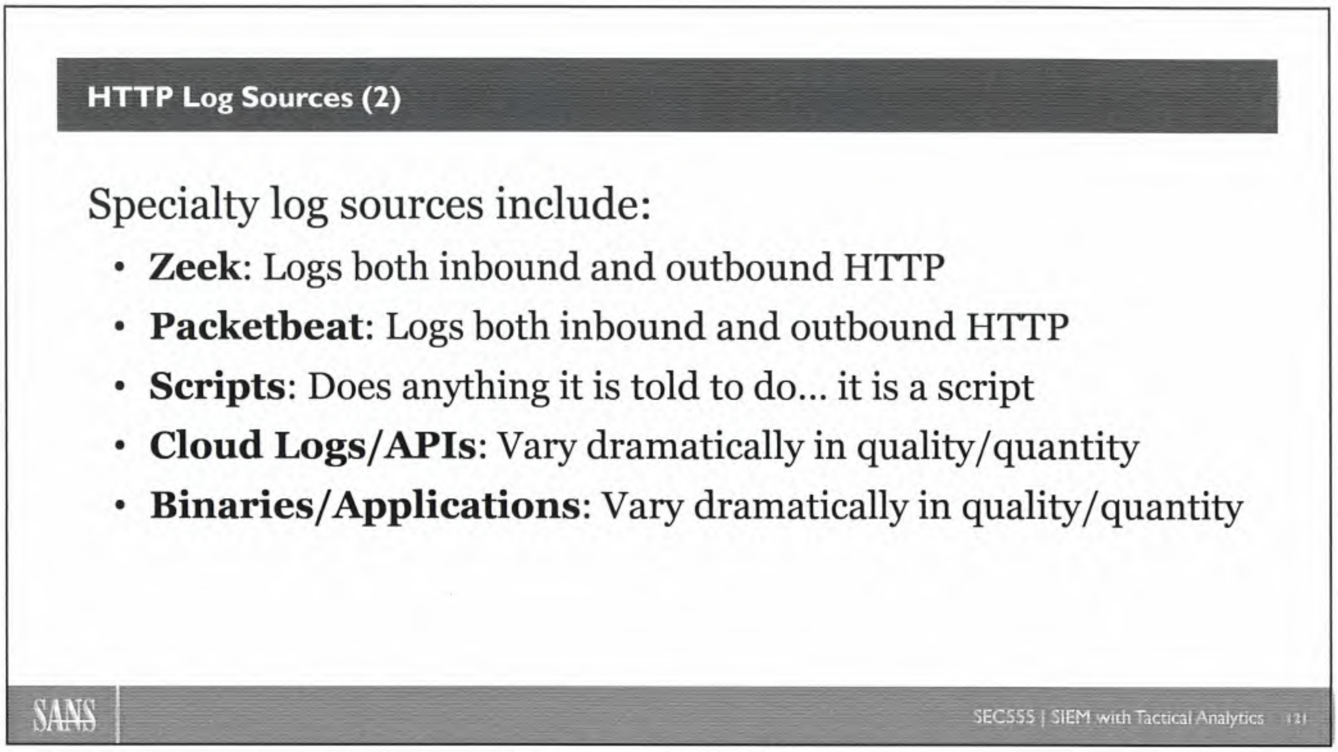
SANS

SEC555 | SIEM with Tactical Analytics 120

HTTP Log Sources (1)

Once again, before applying techniques against logs, the logs must first be acquired. Some of the most common sources for HTTP logs are web servers and web proxies. However, other sources for HTTP logs can include advanced firewalls, malware detonation devices, and network extraction of logs.

Notice that an HTTP log is not simply a log of who or what accessed port 80. HTTP logs are logs that include additional fields such as virtual host, user agent, and so on.



HTTP Log Sources (2)

Specialty log sources include:

- **Zeek**: Logs both inbound and outbound HTTP
- **Packetbeat**: Logs both inbound and outbound HTTP
- **Scripts**: Does anything it is told to do... it is a script
- **Cloud Logs/APIs**: Vary dramatically in quality/quantity
- **Binaries/Applications**: Vary dramatically in quality/quantity

HTTP Log Sources (2)

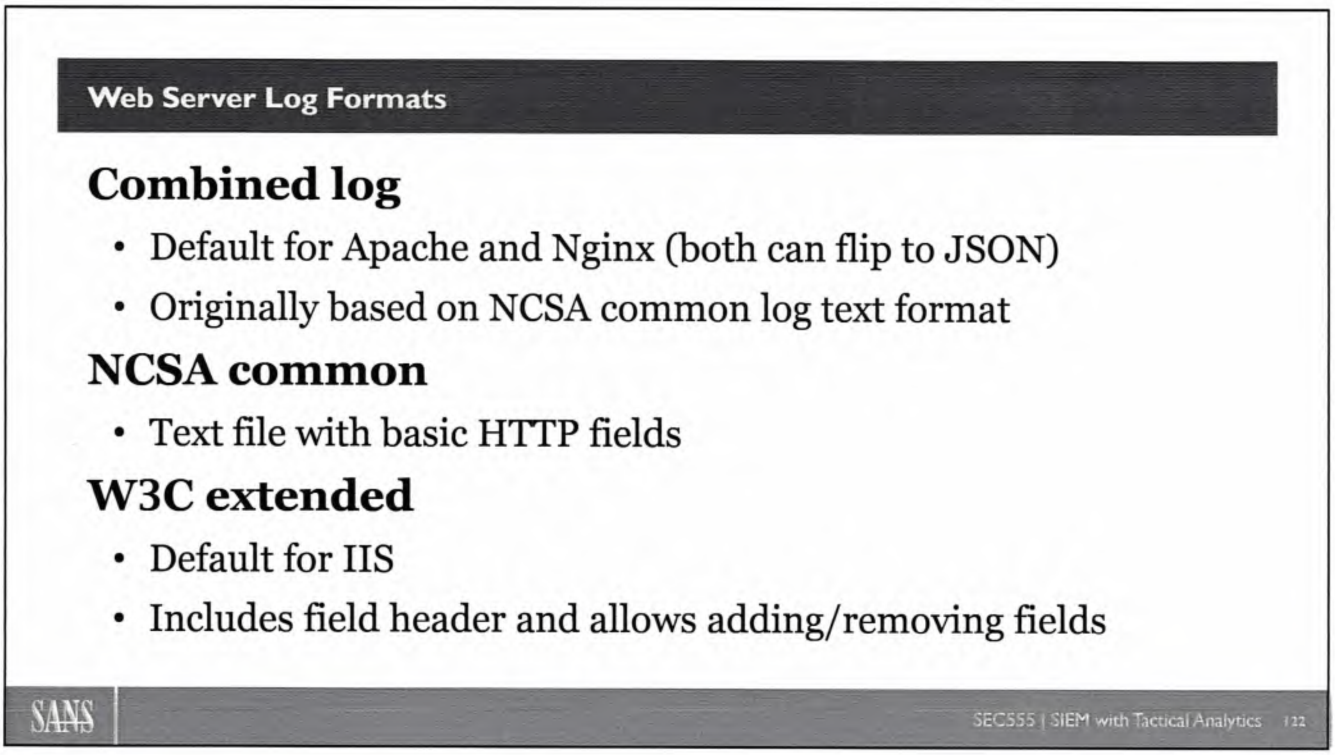
Because of its widespread use, HTTP also has alternative ways to collect logs. For network extraction, Zeek or Packetbeat can be used. Again, the advantage of using network extraction is it ends up being a single point for centralized logging that provides a consistent log format. In regard to HTTP, Zeek or Packetbeat also generate logs that would normally be generated on a web server as well as web clients. Scripts can also be used to generate or pick up logs.

One issue that is, unfortunately, common is that cloud solutions provide poor to no means of accessing logs. For example, most cloud solutions, such as Office 365, require making calls to APIs to retrieve logs. That would not be so bad, but often these logs retrieved via API do not have the same details that would be available if the service were being hosted on premises.

References:

https://www.zeek.org, https://sec555.com/4h

https://www.elastic.co/beats/packetbeat, https://sec555.com/5b



Web Server Log Formats

Combined log

- Default for Apache and Nginx (both can flip to JSON)
- Originally based on NCSA common log text format

NCSA common

- Text file with basic HTTP fields

W3C extended

- Default for IIS
- Includes field header and allows adding/removing fields

SEC555 | SIEM with Tactical Analytics 122

Web Server Log Formats

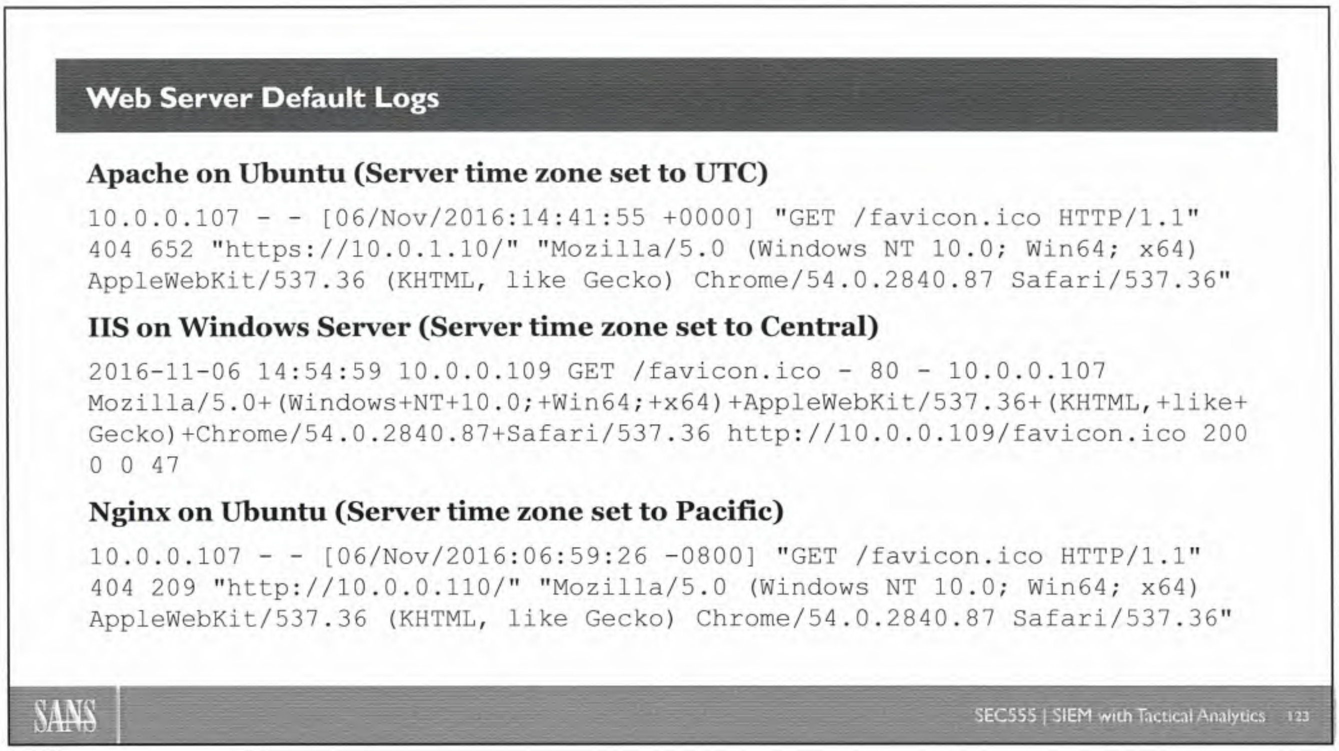
The three most common web server log formats are combined, NCSA, and W3C extended. Realistically, combined and W3C are the most prominent as Apache, Nginx, and IIS default to these formats.

NCSA common was an initial format that included the bare minimum fields. It included the following fields in a text-based file:

- host
- rfc931
- username
- datetime
- request
- statuscode
- bytes

Combined log format was based initially on the NCSA common log but was modified to allow many additional fields as well as customization of the logs. For example, the log can be saved in JSON format and the field format, such as a date field, can be modified to display in a certain way.

W3C extended format was proposed by W3C and designed to allow more granular control of web logs. It allows for inclusion or exclusion of fields, proxy logging, and more.



Web Server Default Logs

Apache on Ubuntu (Server time zone set to UTC)

```
10.0.0.107 - - [06/Nov/2016:14:41:55 +0000] "GET /favicon.ico HTTP/1.1"
404 652 "https://10.0.1.10/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.87 Safari/537.36"
```

IIS on Windows Server (Server time zone set to Central)

```
2016-11-06 14:54:59 10.0.0.109 GET /favicon.ico - 80 - 10.0.0.107
Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+
Gecko)+Chrome/54.0.2840.87+Safari/537.36 http://10.0.0.109/favicon.ico 200
0 0 47
```

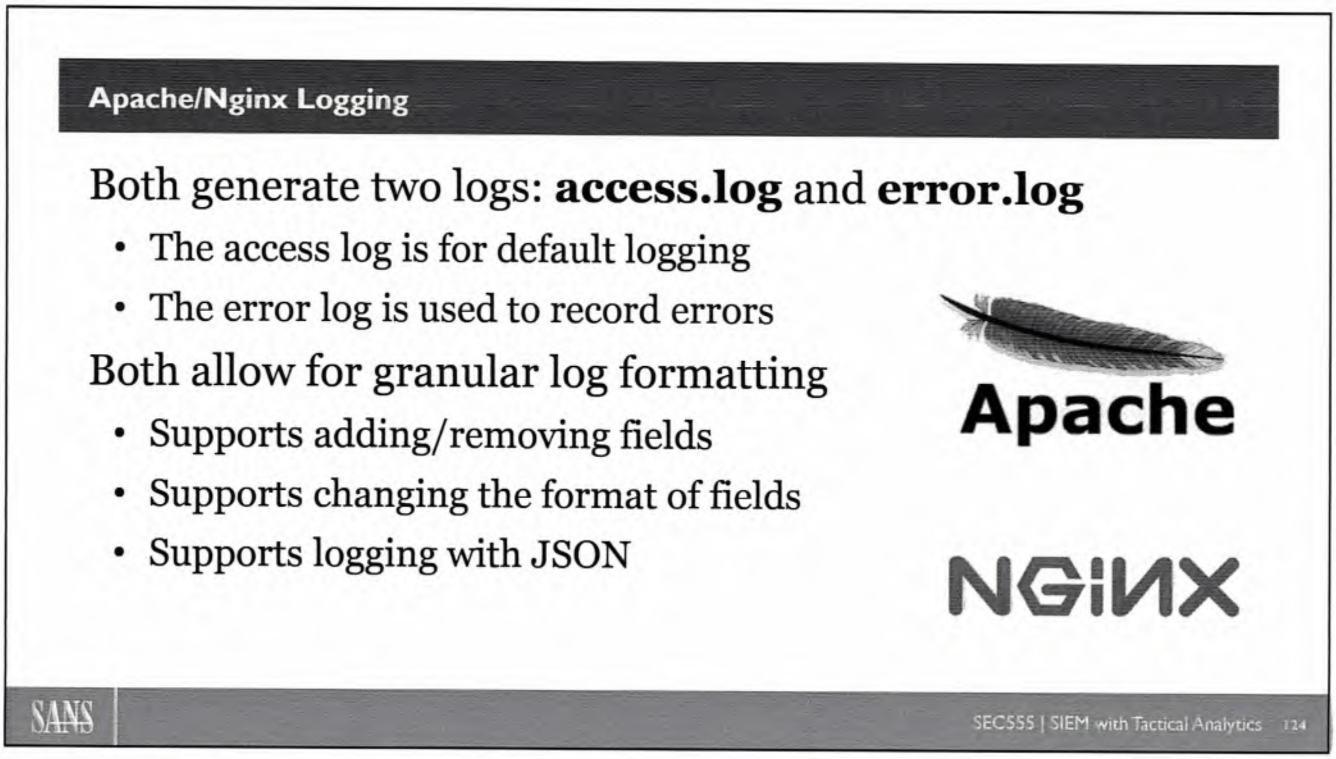
Nginx on Ubuntu (Server time zone set to Pacific)

```
10.0.0.107 - - [06/Nov/2016:06:59:26 -0800] "GET /favicon.ico HTTP/1.1"
404 209 "http://10.0.0.110/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.87 Safari/537.36"
```

Web Server Default Logs

This slide includes default logs for Apache, Nginx, and IIS for accessing the file favicon.ico. For Apache and Nginx, the logs came from an Ubuntu system, and the IIS logs came from a Windows Server box. Note that Apache and Nginx's default log format is the same while IIS is slightly different. While minor, once again, log inconsistencies mean more work for you.

In these logs, each system was set to a different time zone. For Apache and Nginx, the time zone is specified in the log by default. For IIS, the log does not specify the time zone, so be careful and know thy logs.



Apache/Nginx Logging

Both generate two logs: **access.log** and **error.log**

- The access log is for default logging
- The error log is used to record errors

Both allow for granular log formatting

- Supports adding/removing fields
- Supports changing the format of fields
- Supports logging with JSON

Apache
NGINX

Apache/Nginx Logging

Apache and Nginx default to creating two logs: access.log and error.log. The access.log is the default log of requests to the web server, and error.log is a record of errors or abnormalities discovered by the web servers. For HTTP analysis, access.log is what we are primarily analyzing.

Both web servers allow granular customization of logs. They support adding and removing fields, changing the format of fields, such as making dates follow YYYY-MM-dd format, and even tuning the output, such as making the logs output in JSON format.

For a sample of customizing Apache log formats, see Apache's site.

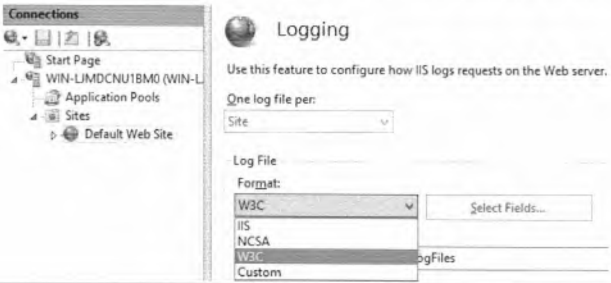
Reference:

https://httpd.apache.org/docs/current/mod/mod_log_config.html, https://sec555.com/5c

IIS Logging

Supports multiple log formats

- **IIS**: Comma-separated list of fixed field list
- **NCSA**: Log file with fixed set of basic fields
- **W3C**: Extendable log format (recommended and default)



Use this feature to configure how IIS logs requests on the Web server.

One log file per: Site

Log File

Format: W3C

Select Fields...

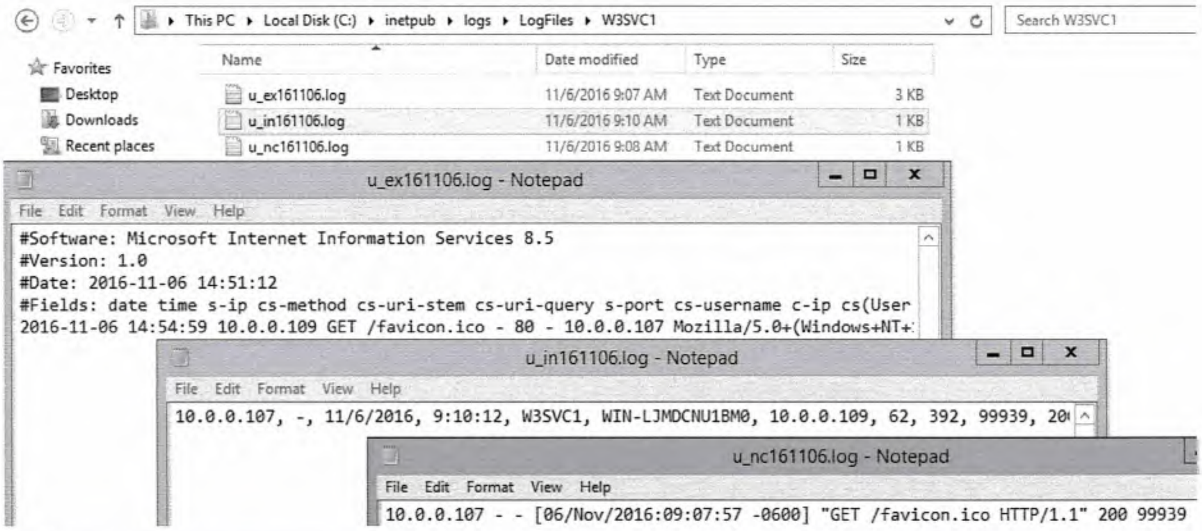
IIS
NCSA
W3C
Custom

ogFiles

IIS Logging

IIS defaults to W3C but has a few other logging options. The recommendation is to stick with W3C as it allows the most granular control, and the other options are fairly basic and limited. While not mentioned on this slide, IIS can also be configured to log to a database.

IIS Logs


This PC ▸ Local Disk (C:) ▸ inetpub ▸ logs ▸ LogFiles ▸ W3SVC1
Search W3SVC1
Favorites
Desktop
Downloads
Recent places
File Edit Format View Help

Name	Date modified	Type	Size
u_ex161106.log	11/6/2016 9:07 AM	Text Document	3 KB
u_in161106.log	11/6/2016 9:10 AM	Text Document	1 KB
u_nc161106.log	11/6/2016 9:08 AM	Text Document	1 KB

u_ex161106.log - Notepad

```
#Software: Microsoft Internet Information Services 8.5
#Version: 1.0
#Date: 2016-11-06 14:51:12
#Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(User
2016-11-06 14:54:59 10.0.0.109 GET /favicon.ico - 80 - 10.0.0.107 Mozilla/5.0+(Windows+NT+:
```

u_in161106.log - Notepad

```
10.0.0.107, -, 11/6/2016, 9:10:12, W3SVC1, WIN-LJMDCNU1BM0, 10.0.0.109, 62, 392, 99939, 20
```

u_nc161106.log - Notepad

```
10.0.0.107 - - [06/Nov/2016:09:07:57 -0600] "GET /favicon.ico HTTP/1.1" 200 99939
```

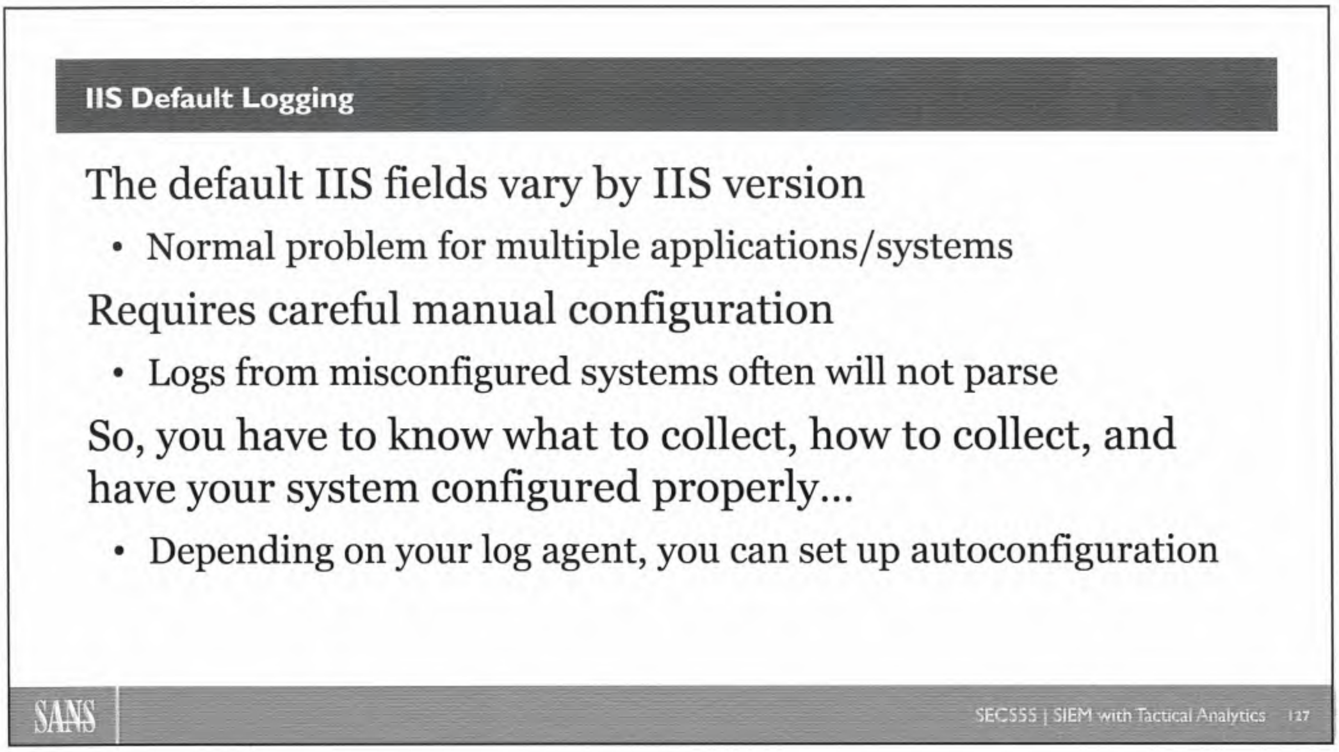
SANS

SEC555 | SIEM with Tactical Analytics 126

IIS Logs

This slide demonstrates the three main types of IIS logs: IIS, NCSA, and W3C. Below is the mapping of log type to filename for your reference. Each file represents a log requesting the same file, which is /favicon.ico.

W3C: u_ex161106.log
IIS: u_in161106.log
NCSA: u_nc161106.log



IIS Default Logging

The default IIS fields vary by IIS version

- Normal problem for multiple applications/systems

Requires careful manual configuration

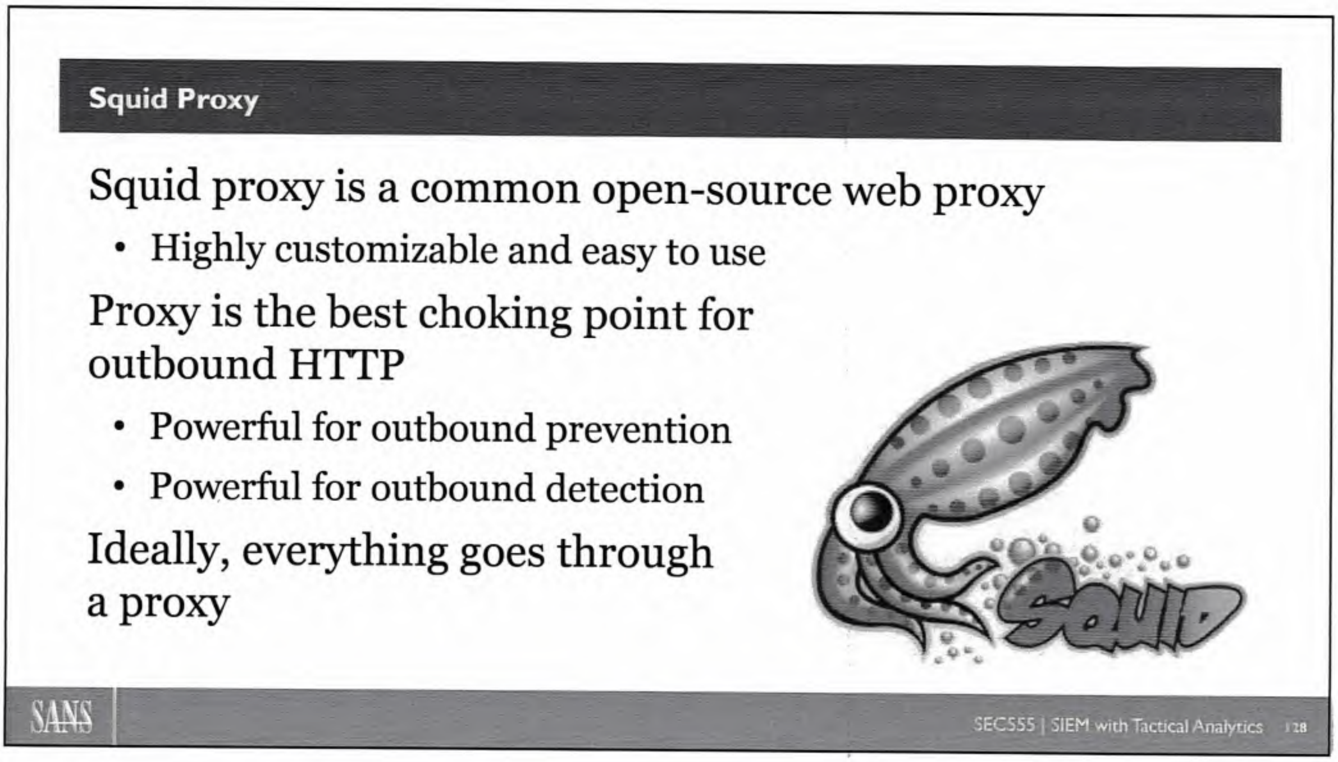
- Logs from misconfigured systems often will not parse

So, you have to know what to collect, how to collect, and have your system configured properly...

- Depending on your log agent, you can set up autoconfiguration

IIS Default Logging

A key challenge for SIEM solutions is collecting and managing logs from disparate systems. Unfortunately, this is true across version changes of IIS. For example, IIS on Server 2012 R2 defaults to including additional fields not collected on IIS on Server 2003. This means consistency must be accounted for in log collection as well as configuration. It is possible to alter IIS on the various versions, so the logs all include the same fields and format using Microsoft command line tools or PowerShell (only on newer IIS versions).



Squid Proxy

Squid proxy is a common open-source web proxy

- Highly customizable and easy to use

Proxy is the best choking point for outbound HTTP

- Powerful for outbound prevention
- Powerful for outbound detection

Ideally, everything goes through a proxy

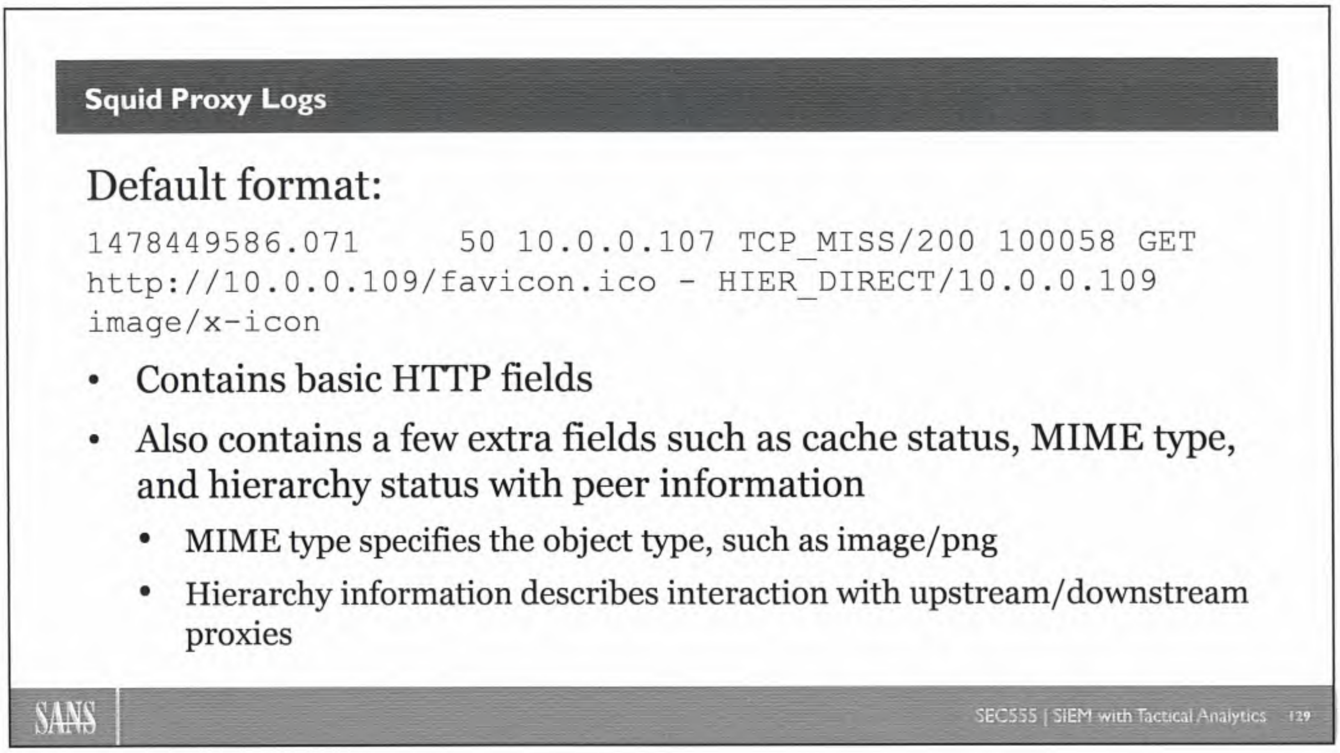
Squid Proxy

A high quality and free web proxy can be found with Squid, which is a highly customizable proxy that allows caching of content for lowering bandwidth requirements and improving latency but can also be used for content filtering and ICAP support. It can also be used to perform SSL man-in-the-middle for content filtering or antivirus checks against encrypted content.

Web proxies, in general, are a completely underrated prevention and detection technology. If outbound HTTP requests were only allowed from a proxy, then lots of malware would fail because they are not proxy aware. This, in turn, allows the detection of failed malware runs. Outside of that, tunneling all HTTP traffic through a proxy makes it an easy central location to collect and analyze logs. If you are not using a proxy… get one.

Reference:

http://www.squid-cache.org, https://sec555.com/5d



Squid Proxy Logs

Default format:

```
1478449586.071      50 10.0.0.107 TCP_MISS/200 100058 GET
http://10.0.0.109/favicon.ico - HIER_DIRECT/10.0.0.109
image/x-icon
```

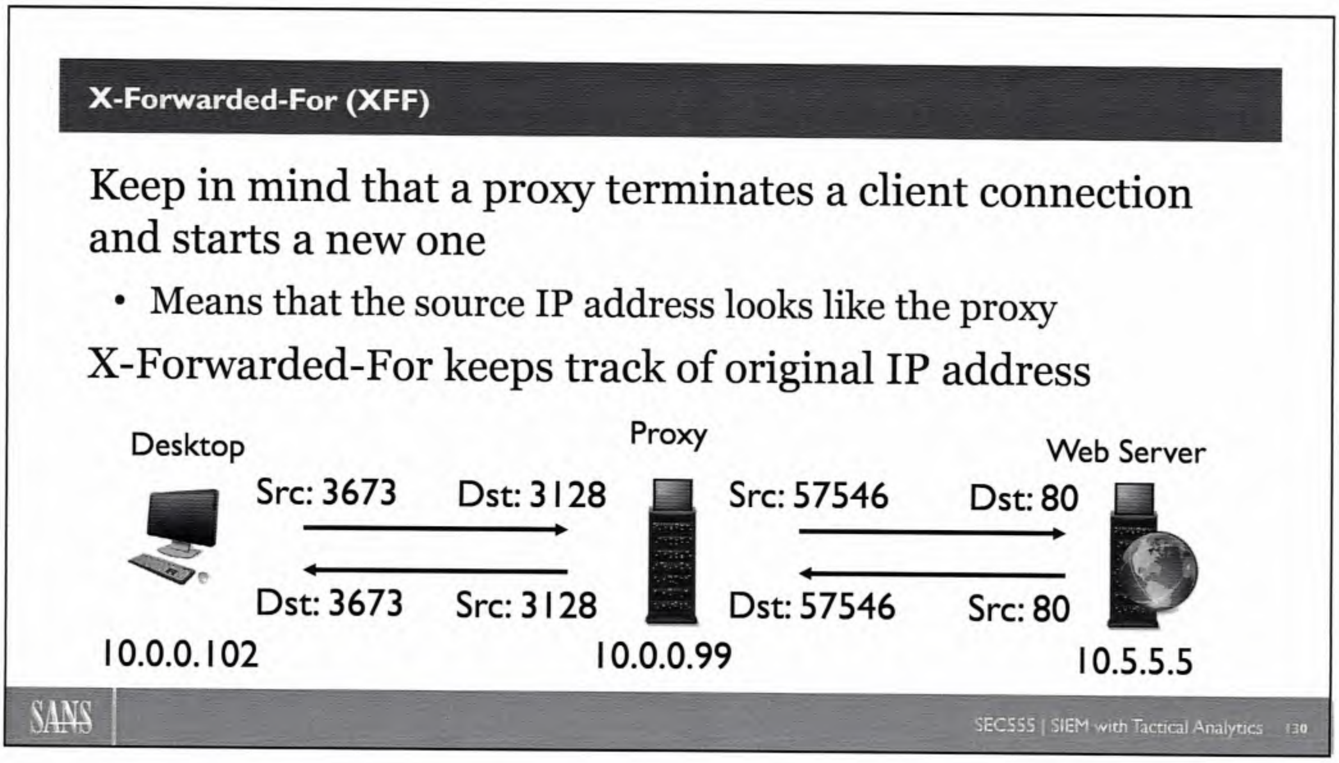
- Contains basic HTTP fields
- Also contains a few extra fields such as cache status, MIME type, and hierarchy status with peer information
  - MIME type specifies the object type, such as image/png
  - Hierarchy information describes interaction with upstream/downstream proxies

SANS

SEC555 | SIEM with Tactical Analytics 129

Squid Proxy Logs

The default format of a squid proxy log is different from web server logs but not by much. Because it is a proxy, a few additional fields exist that are proxy-specific. For example, there is typically a field that shows if an HTTP object was pulled from cache or not.



X-Forwarded-For (XFF)

Keep in mind that a proxy terminates a client connection and starts a new one

- Means that the source IP address looks like the proxy

X-Forwarded-For keeps track of original IP address

Desktop

Src: 3673 Dst: 3128

Dst: 3673 Src: 3128

10.0.0.102

Proxy

Src: 57546 Dst: 80

Dst: 57546 Src: 80

10.0.0.99

Web Server

10.5.5.5

SANS

SEC555 | SIEM with Tactical Analytics 130

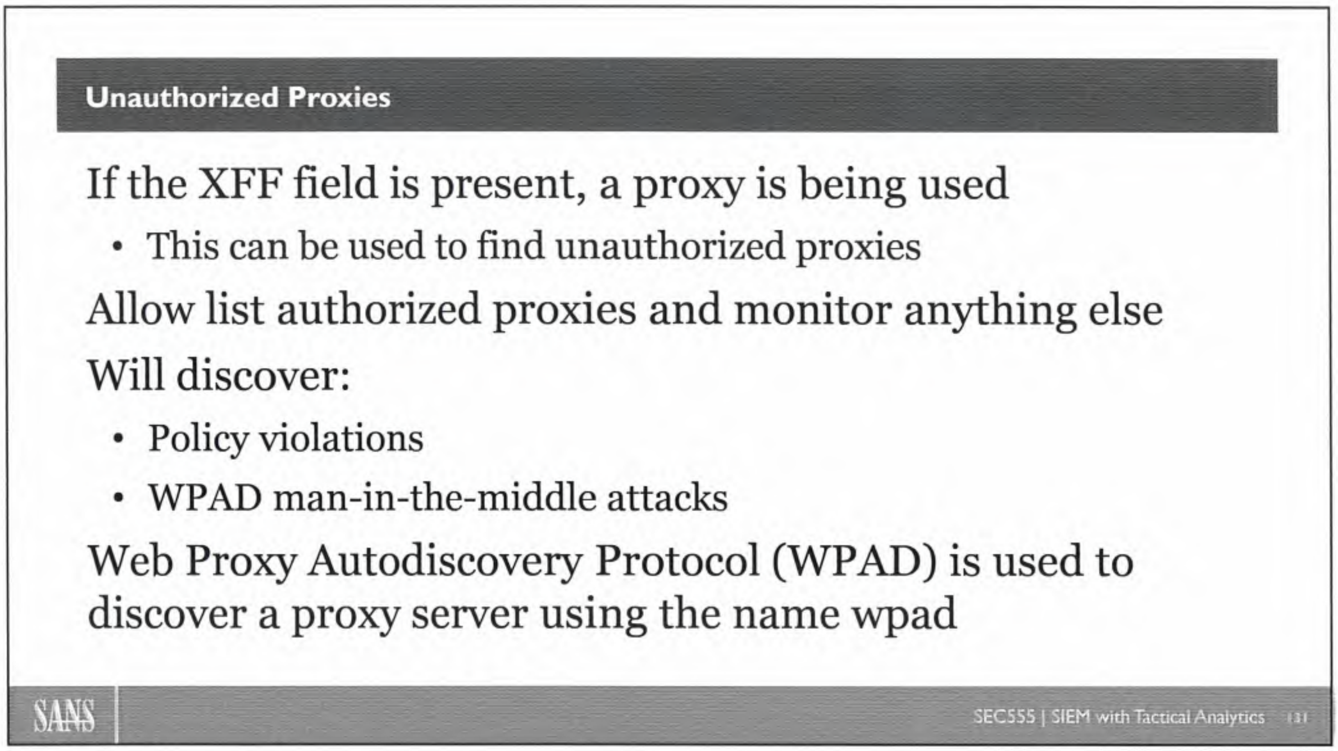
X-Forwarded-For (XFF)

Using a proxy can be great for detection and prevention, but it does add a layer of complexity when performing an analysis. For example, assume the desktop 10.0.0.102 is connecting to 10.5.5.5 through the proxy 10.0.0.99. In the scenario, the web server 10.5.5.5 will log that 10.0.0.99 is what accessed it, not 10.0.0.102. This is because a web proxy is an intentional man-in-the-middle. The desktop has a connection to the proxy, and the proxy has a separate connection to the web server. The means two separate sockets with their own IP addresses and ports are used to make any given connection.

So, if this is the case, how can you tell that 10.0.0.102 is the system actually accessing 10.5.5.5? The answer is X-Forwarded-For (XFF). This header field contains the source IP address of the system that started the request. Keep in mind that some systems require modification of logs to include the XFF field. Zeek, by default, will extract this. Web servers often will not. For instance, IIS requires enabling Advanced Logging and then adding the field X-Forwarded-For to log this.

This field, in particular, can be troublesome when tracking down connections. Some organizations take logs that include an XFF field and overwrite the source_ip address field with the XFF contents. This author is not a big fan of that as then the logs and ports do not match up. Another approach is to add all the source_ip, destination_ip, and XFF IP address fields to an array so you can search across all logs for particular IP addresses. Taking this a step further, you may also want to create a source_ips array and destination_ips array. In this case, the XFF IP address would be added to both the ips and source_ips array fields.

The X-Forwarded-For field is arbitrary and can be forged.



Unauthorized Proxies

If the XFF field is present, a proxy is being used

- This can be used to find unauthorized proxies

Allow list authorized proxies and monitor anything else

Will discover:

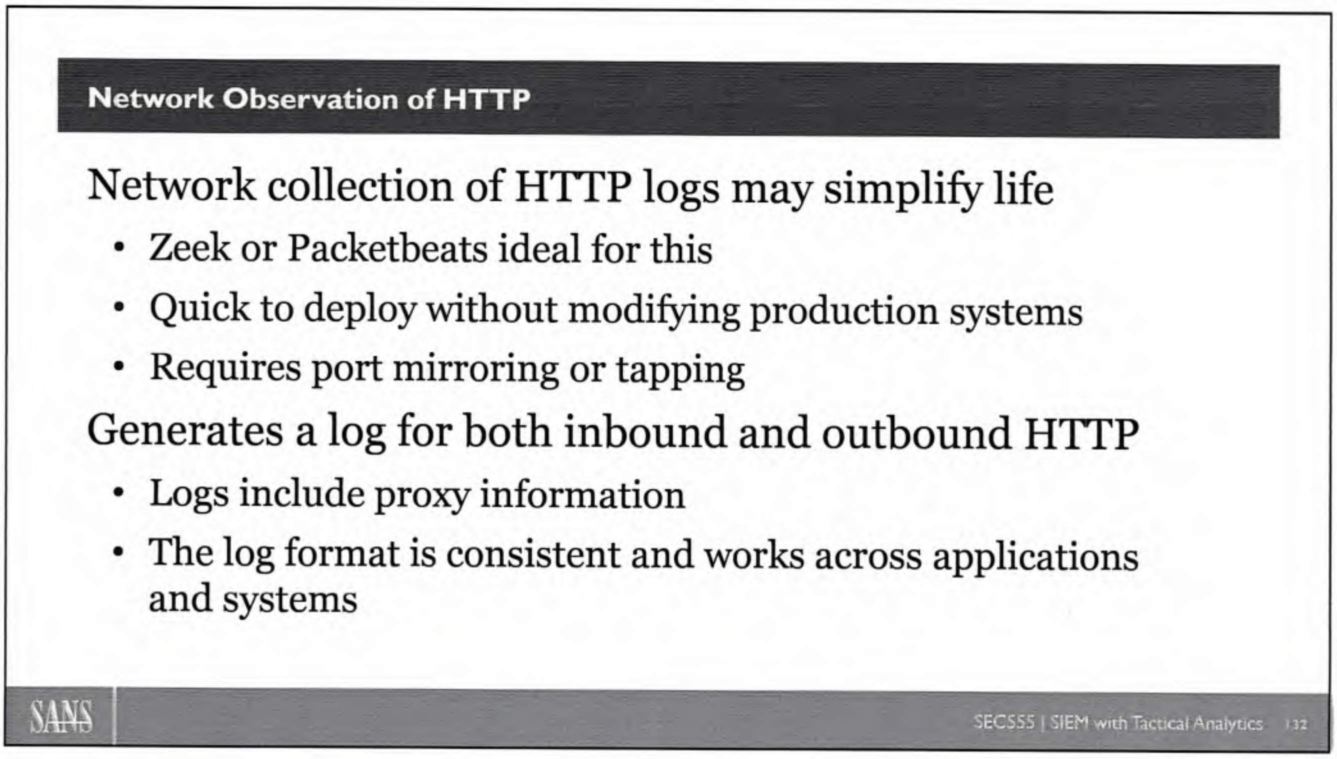
- Policy violations
- WPAD man-in-the-middle attacks

Web Proxy Autodiscovery Protocol (WPAD) is used to discover a proxy server using the name wpad

Unauthorized Proxies

A simple trick that most organizations are not doing is to look for unauthorized proxies. Ideally, HTTP access to the internet is restricted to only authorized proxies. However, this simply is not the case in most organizations. Regardless of if it is or is not, monitoring for proxy use over anything but known proxies is a quick way to find internal users trying to bypass internal security or malicious proxy use.

A well-known attack method is to gain control of systems using WPAD. If an attacker can use DHCP, DNS poisoning, or even the NetBIOS Name Service, then many victim systems will automatically connect to the attacker's proxy. This is because the default settings of most browsers and operating systems are to discover proxies using wpad automatically. By looking for unauthorized proxies, you would be able to catch this.



Network Observation of HTTP

Network collection of HTTP logs may simplify life

- Zeek or Packetbeats ideal for this
- Quick to deploy without modifying production systems
- Requires port mirroring or tapping

Generates a log for both inbound and outbound HTTP

- Logs include proxy information
- The log format is consistent and works across applications and systems

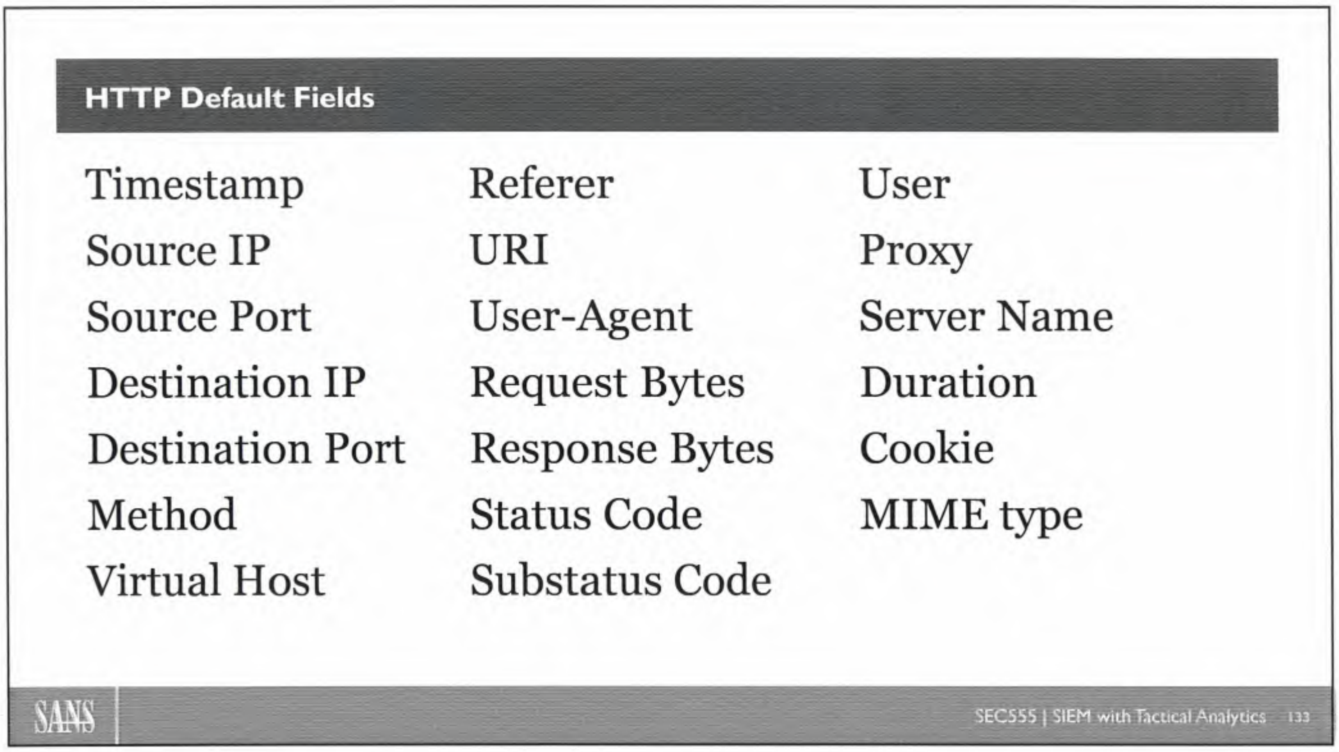
SANS

SEC555 | SIEM with Tactical Analytics 132

Network Observation of HTTP

Again, network extraction of logs is worth investigating. Let us assume your organization is using a squid proxy and has both IIS servers and Apache web servers. If you were to collect logs traditionally from these systems, you would need to separately tune and configure logging for the proxy, each Apache web server, and each IIS server. Varying versions of IIS etc. may require additional tuning and challenges. At a minimum, there would be three separate parsers required to extract fields properly. Another challenge for traditional logging is the necessity to identify and configure every asset for logging. What would happen if some systems were not going through the proxy or if someone stood up a web server but forgot to set up log collection?

Instead, a single network extractor, such as Zeek, could be deployed and then HTTP logs would be collected. Even if a system did not go through the proxy, as long as the network extractor had network visibility, it would generate an HTTP log. Granted… you could view a network extractor as a single point of failure, but in that case, you could stand up two if you truly needed the redundancy. And technically, each traditional agent is a single point of failure for its respective logs.

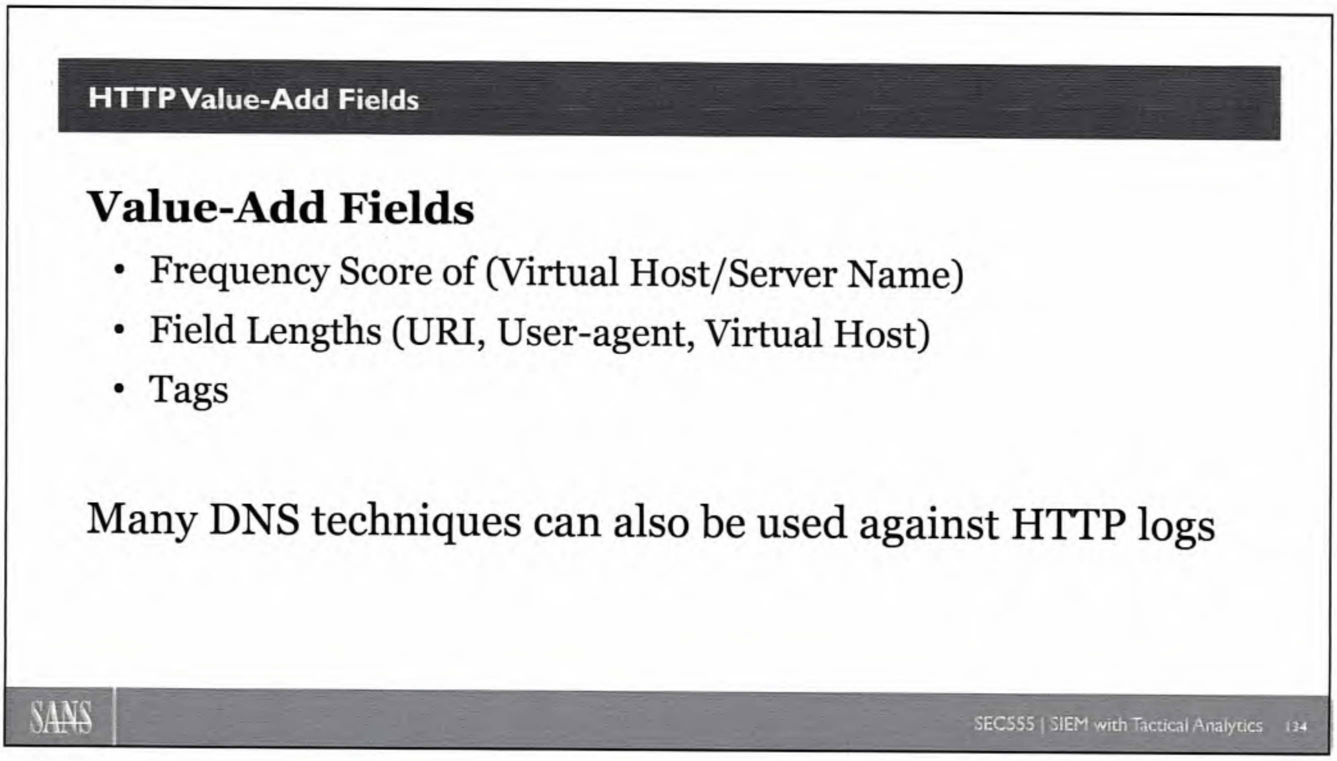


HTTP Default Fields

- Timestamp
- Source IP
- Source Port
- Destination IP
- Destination Port
- Method
- Virtual Host
- Referer
- URI
- User-Agent
- Request Bytes
- Response Bytes
- Status Code
- Substatus Code
- User
- Proxy
- Server Name
- Duration
- Cookie
- MIME type

HTTP Default Fields

This slide demonstrates the most common HTTP fields included in HTTP logs.



HTTP Value-Add Fields

Value-Add Fields

- Frequency Score of (Virtual Host/Server Name)
- Field Lengths (URI, User-agent, Virtual Host)
- Tags

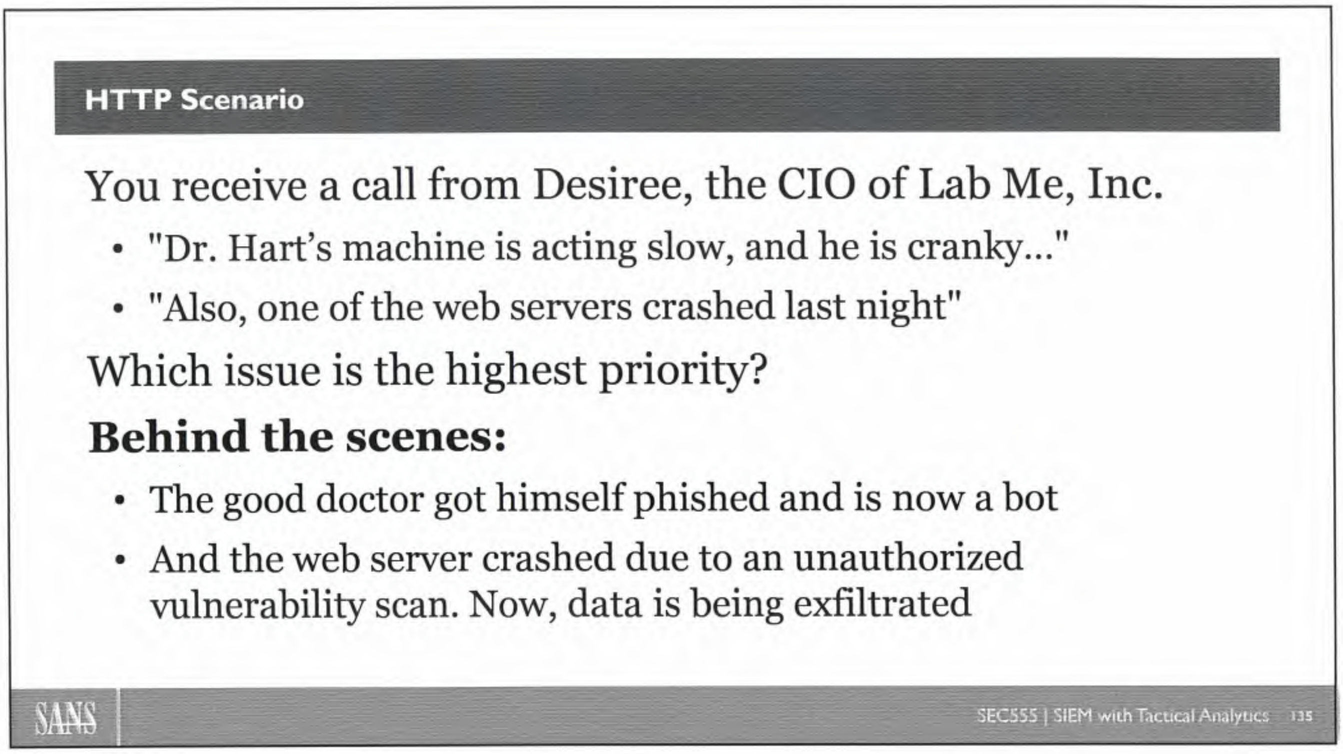
Many DNS techniques can also be used against HTTP logs

SEC555 | SIEM with Tactical Analytics 134

HTTP Value-Add Fields

To add a little spice to HTTP logs, consider adding frequency scores, field lengths, and tags. More information on this is included in the upcoming slides.

Technically, many of the log augmentation techniques applied to a DNS log would apply to HTTP. However, if you are analyzing DNS logs using those techniques, there is minimal value in duplicating that effort for HTTP unless you want some of the tags for filtering.



HTTP Scenario

You receive a call from Desiree, the CIO of Lab Me, Inc.

- "Dr. Hart's machine is acting slow, and he is cranky..."
- "Also, one of the web servers crashed last night"

Which issue is the highest priority?

Behind the scenes:

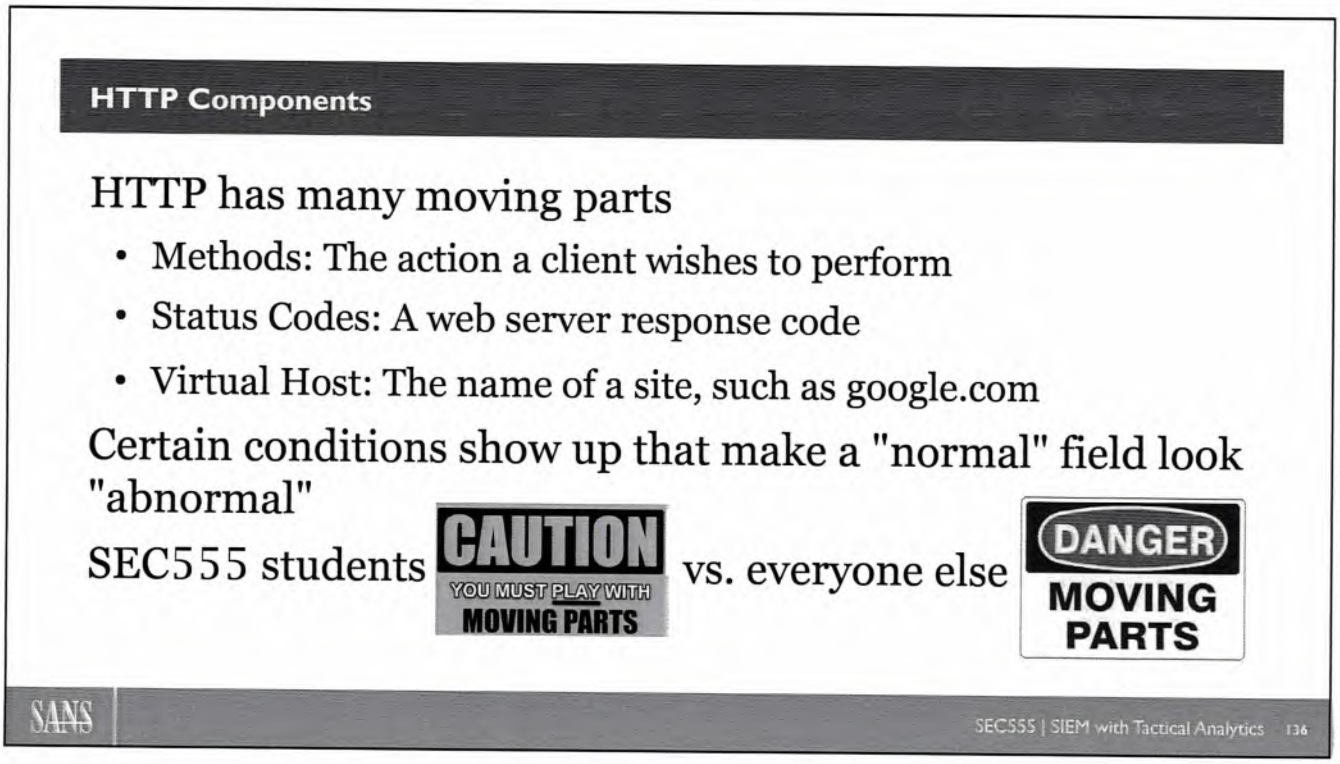
- The good doctor got himself phished and is now a bot
- And the web server crashed due to an unauthorized vulnerability scan. Now, data is being exfiltrated

SANS

SEC555 | SIEM with Tactical Analytics 135

HTTP Scenario

To help understand some of the upcoming techniques, the scenario in the slide has been created based on the author's real-life experiences. While the names have been changed to protect the innocent, this type of scenario is common.



HTTP Components

HTTP has many moving parts

- Methods: The action a client wishes to perform
- Status Codes: A web server response code
- Virtual Host: The name of a site, such as google.com

Certain conditions show up that make a "normal" field look "abnormal"

SEC555 students

CAUTION
YOU MUST PLAY WITH
MOVING PARTS

 vs. everyone else

DANGER
MOVING
PARTS

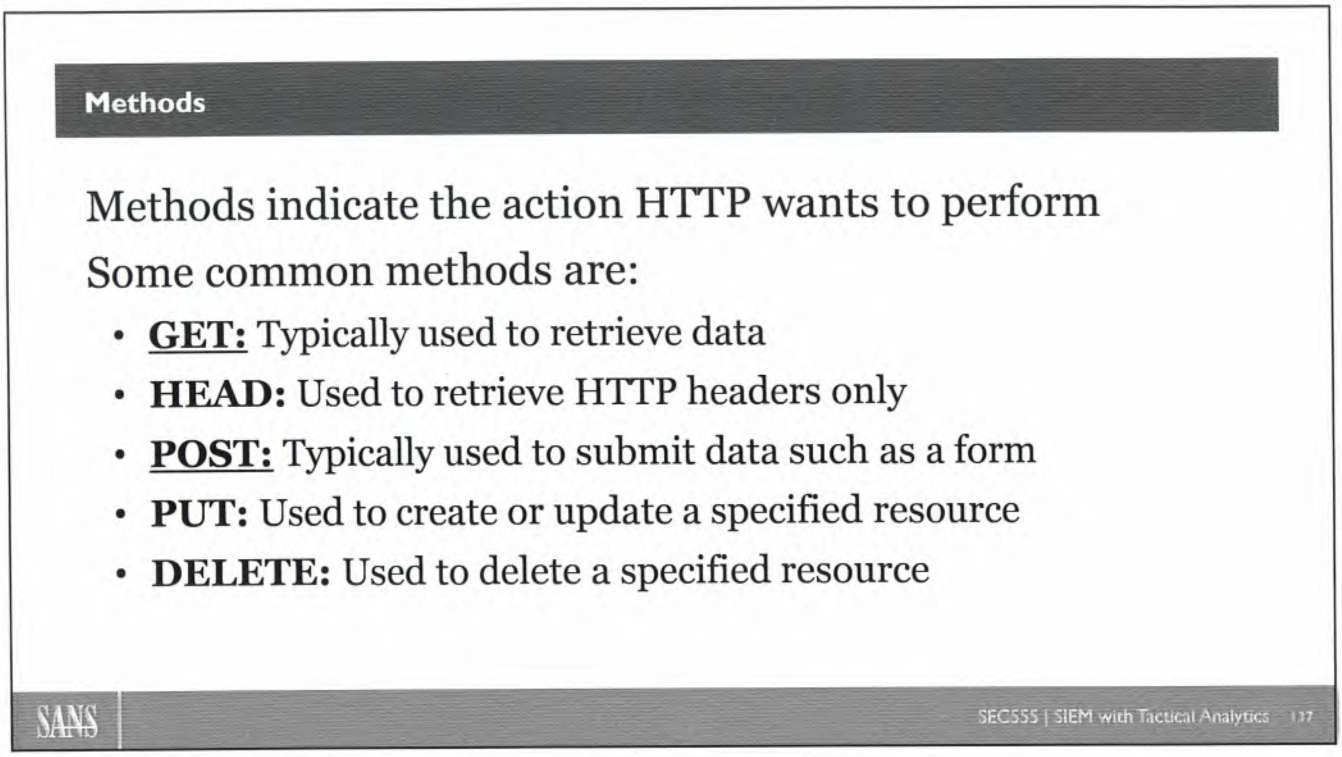
SANS

SEC555 | SIEM with Tactical Analytics 136

HTTP Components

HTTP consists of multiple fields, but a few stand out for some simple yet effective detection techniques. These are methods, status codes, and virtual hosts. A method is what a web client uses to tell the receiving system what action it wishes to perform. For example, internet browsers typically use the GET method to request a web page for viewing. Status codes are a numeric response from the receiving system, back to the web client, describing that status of the response. For example, if a page was requested that did not exist, a web server would still respond back to the client, but it would have a response code of 404. The response code would tell your browser to display Page Cannot Be Found. The virtual host field is used to specify what you are trying to access, whether it is an IP address or a website domain name.

Going back to the underlying theme of SEC555, if you never play with the data, you will miss the opportunity to develop a new detection technique.



Methods

Methods indicate the action HTTP wants to perform

Some common methods are:

- **GET:** Typically used to retrieve data
- **HEAD:** Used to retrieve HTTP headers only
- **POST:** Typically used to submit data such as a form
- **PUT:** Used to create or update a specified resource
- **DELETE:** Used to delete a specified resource

SANS

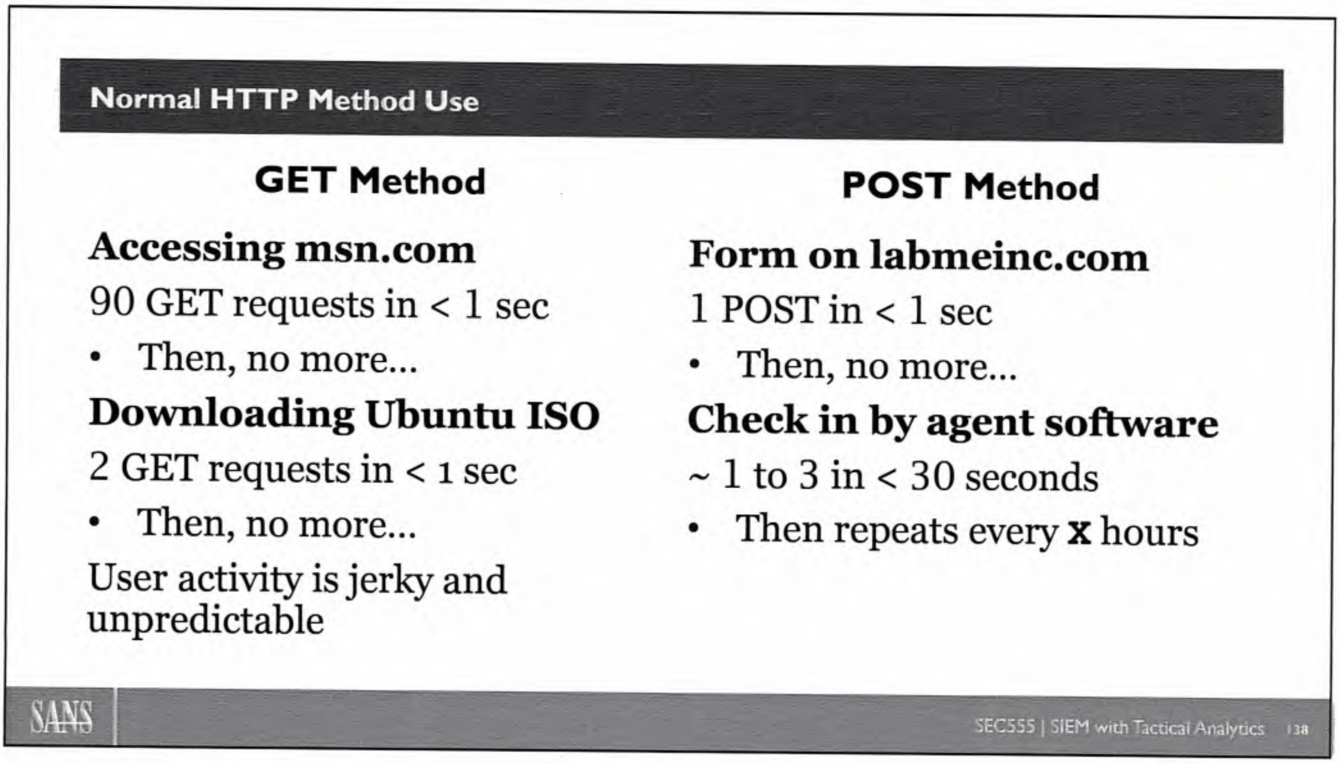
SEC555 | SIEM with Tactical Analytics 137

Methods

Methods are simply a way for a web client to tell its destination what action it wishes to perform.

Other common methods not mentioned in the slide:

- **TRACE:** Used to take input and send it back to the client for debugging.
- **OPTIONS:** Lists out all the VERBS supported for a given site.
- **CONNECT:** Allows tunneling a connection through another endpoint such as a proxy.
- **PATCH:** Allows for partially updating an object.



Normal HTTP Method Use

GET Method

Accessing msn.com

90 GET requests in < 1 sec

- Then, no more...

Downloading Ubuntu ISO

2 GET requests in < 1 sec

- Then, no more...

User activity is jerky and unpredictable

POST Method

Form on labmeinc.com

1 POST in < 1 sec

- Then, no more...

Check in by agent software

~ 1 to 3 in < 30 seconds

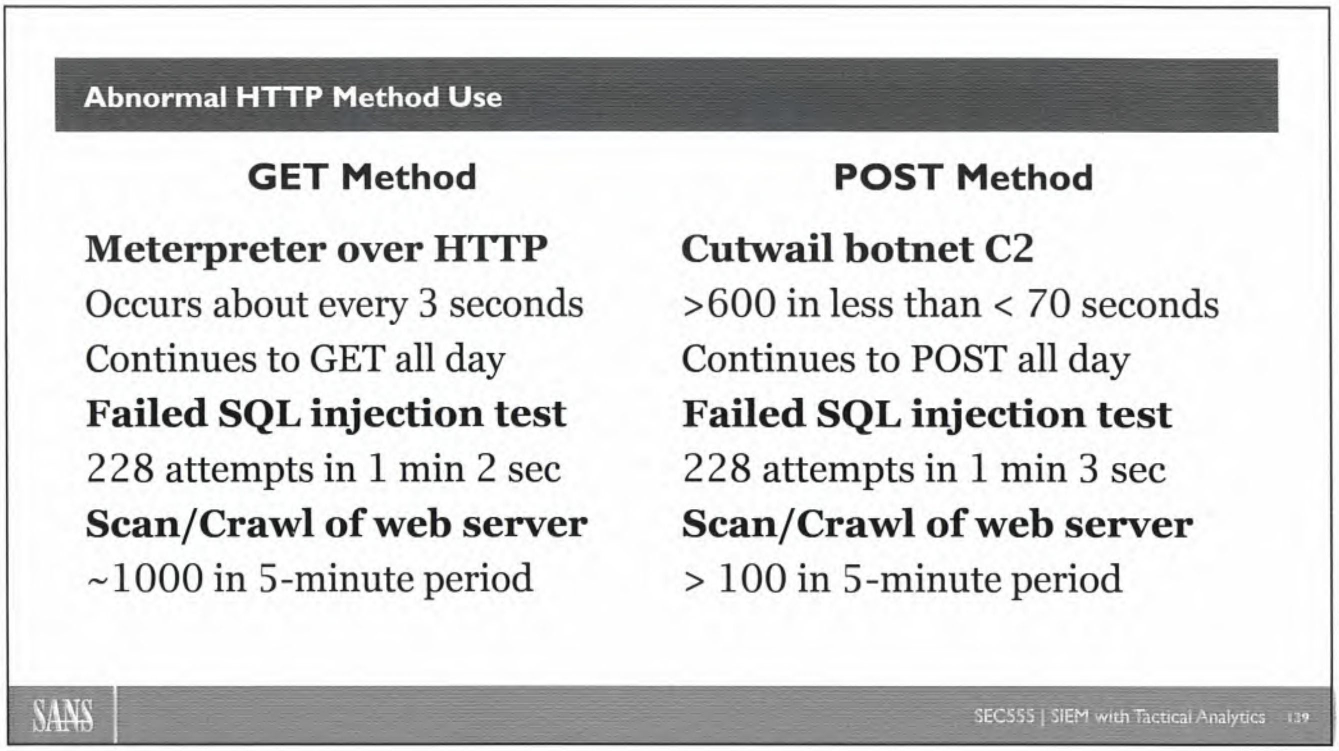
- Then repeats every **x** hours

SEC555 | SIEM with Tactical Analytics 138

Normal HTTP Method Use

A lot of vendors state they cannot share what alerts work because each environment is different. There is some truth to that, but the major failing point is that a SIEM is a ninja at slicing and dicing data. A SIEM is easily able to filter this out and make things actionable again. For example, HTTP GET, and POST tend to follow a pattern of activity. In this slide, a few examples of what a GET and POST would look like are given. If, for some reason, you started monitoring GETs but four systems break the pattern, you can easily tell the SIEM to ignore those four systems.

Be careful of common software update sites. For example, if you run updates on a Linux system, that system will end up having a burst of GET requests. However, if instead of giving up and saying that you cannot monitor methods because certain things like updates break alerts and dashboards, then disabling these tunes out any HTTP requests to update domains like *.microsoft.com or *.ubuntu.com.



Abnormal HTTP Method Use

GET Method

Meterpreter over HTTP

Occurs about every 3 seconds

Continues to GET all day

Failed SQL injection test

228 attempts in 1 min 2 sec

Scan/Crawl of web server

~1000 in 5-minute period

POST Method

Cutwail botnet C2

>600 in less than < 70 seconds

Continues to POST all day

Failed SQL injection test

228 attempts in 1 min 3 sec

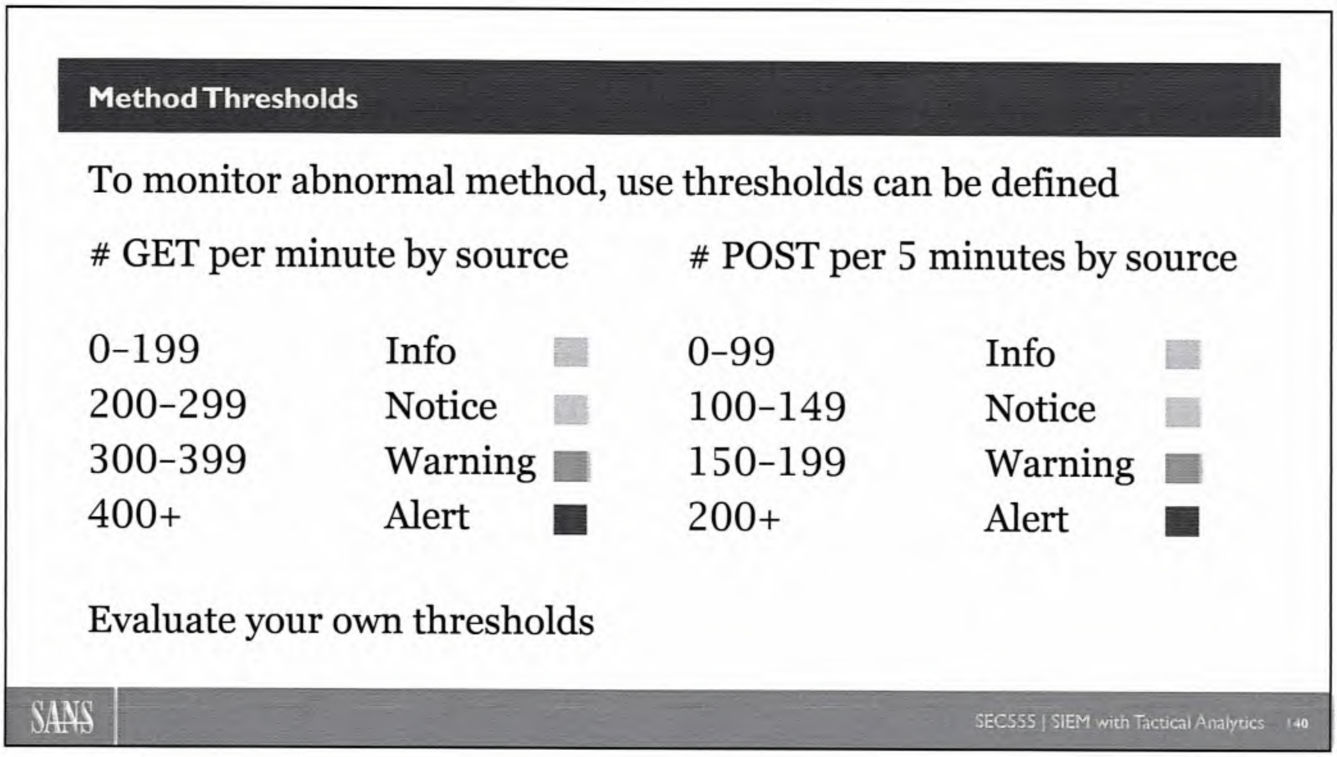
Scan/Crawl of web server

> 100 in 5-minute period

SEC555 | SIEM with Tactical Analytics 139

Abnormal HTTP Method Use

Unfortunately, there are many good things in life that people abuse. HTTP is one of those things. The examples in the slide demonstrate multiple things adversaries do that can cause abnormal HTTP usage.



Method Thresholds

To monitor abnormal method, use thresholds can be defined

# GET per minute by source		
0-199	Info	░
200-299	Notice	░
300-399	Warning	▒
400+	Alert	■

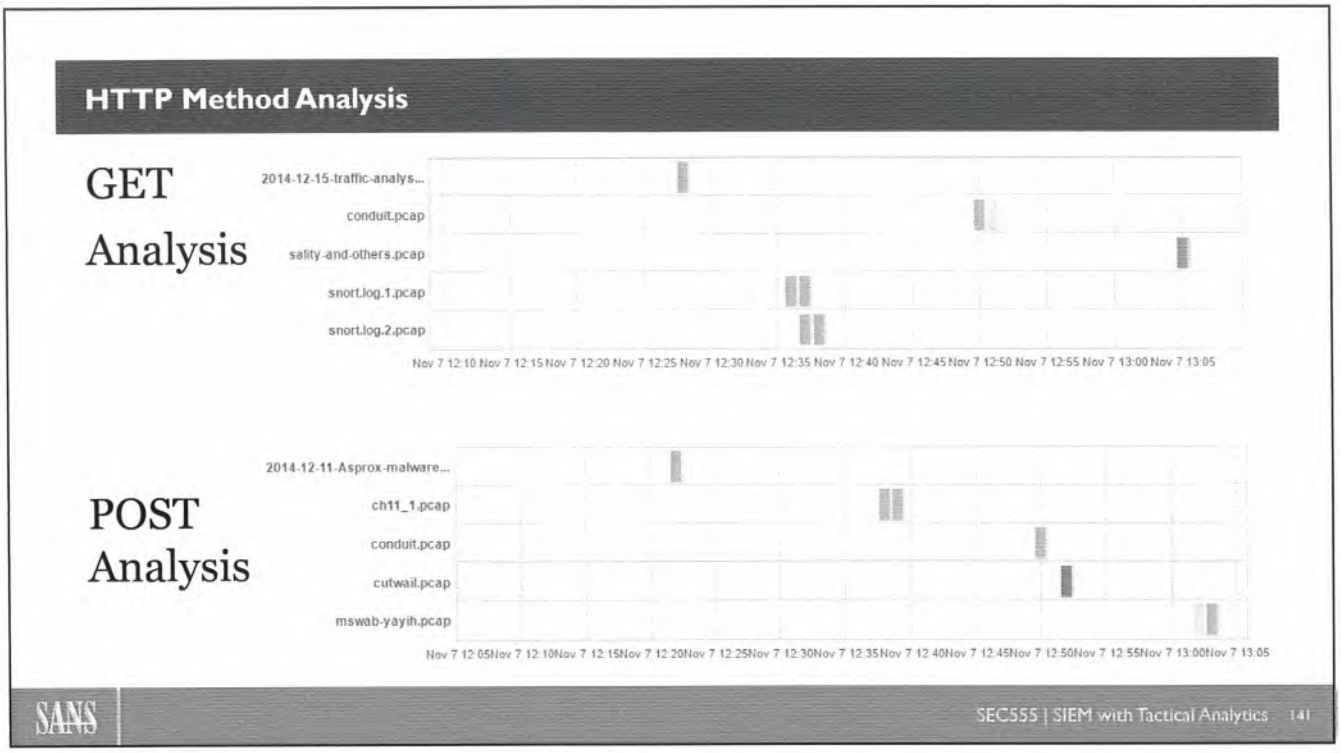
# POST per 5 minutes by source		
0-99	Info	░
100-149	Notice	░
150-199	Warning	▒
200+	Alert	■

Evaluate your own thresholds

SANS

Method Thresholds

A simple technique any SIEM can perform is to monitor a threshold. For instance, it is simple to set up a rule that states if more than a specific number of GET or POST requests are found by a given field, such as source IP address, alert. This technique applies very differently to workstation subnets versus server subnets. However, it can work across both if a high enough threshold is specified.



HTTP Method Analysis

GET Analysis

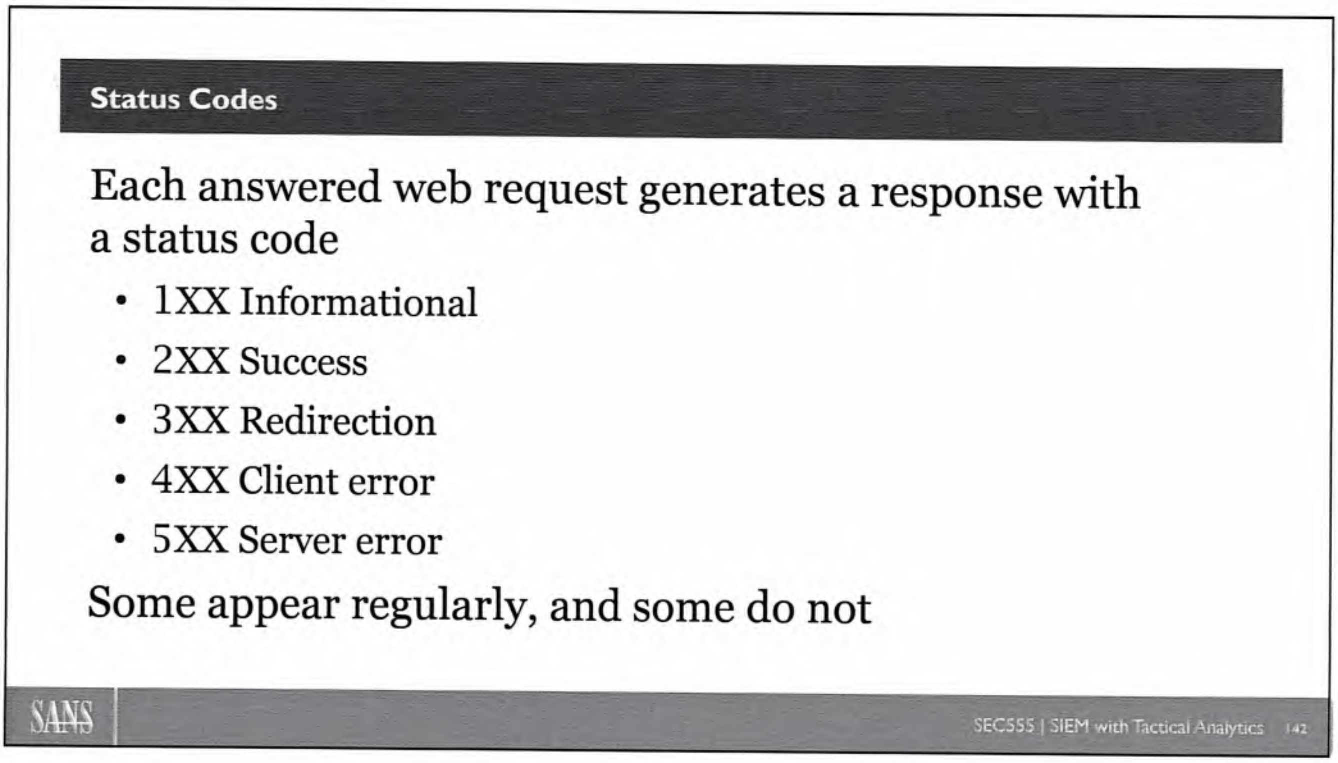
POST Analysis

HTTP Method Analysis

This slide demonstrates GET and POST method threshold monitoring. The pcap file names referenced are from Contagio[1] and contain known bad malware traffic. The course author uses this data, as well as malware samples, to test out techniques. Sometimes, this can be beneficial for looking at known abnormal or malicious traffic and comparing to your own environment. Just be careful to analyze malware and pcaps in a safe environment.

Reference:

https://contagiodump.blogspot.com, https://sec555.com/5e



Status Codes

Each answered web request generates a response with a status code

- 1XX Informational
- 2XX Success
- 3XX Redirection
- 4XX Client error
- 5XX Server error

Some appear regularly, and some do not

Status Codes

Another component of HTTP communication that is important to understand is HTTP status codes. When an HTTP request is made, the response to the request includes a status code indicating information about the request. For example, a successful web request will have a responding status code of 200 OK. A request to a non-existent page would receive a status code of 404 Not Found.

Reference:

https://www.restapitutorial.com/httpstatuscodes.html, https://sec555.com/5f

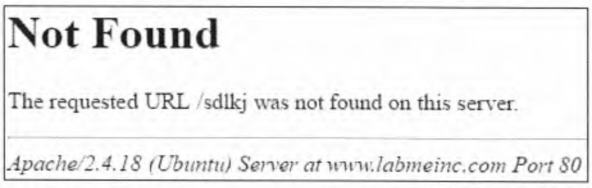
404 Not Found

Status code 404 is similar to DNS's NXDOMAIN record

- Occurs when a request is made for something that does not exist

Can occur due to typos but when monitored will discover:

- Web crawlers
- Vulnerability scanners
- Misconfigured websites



Not Found

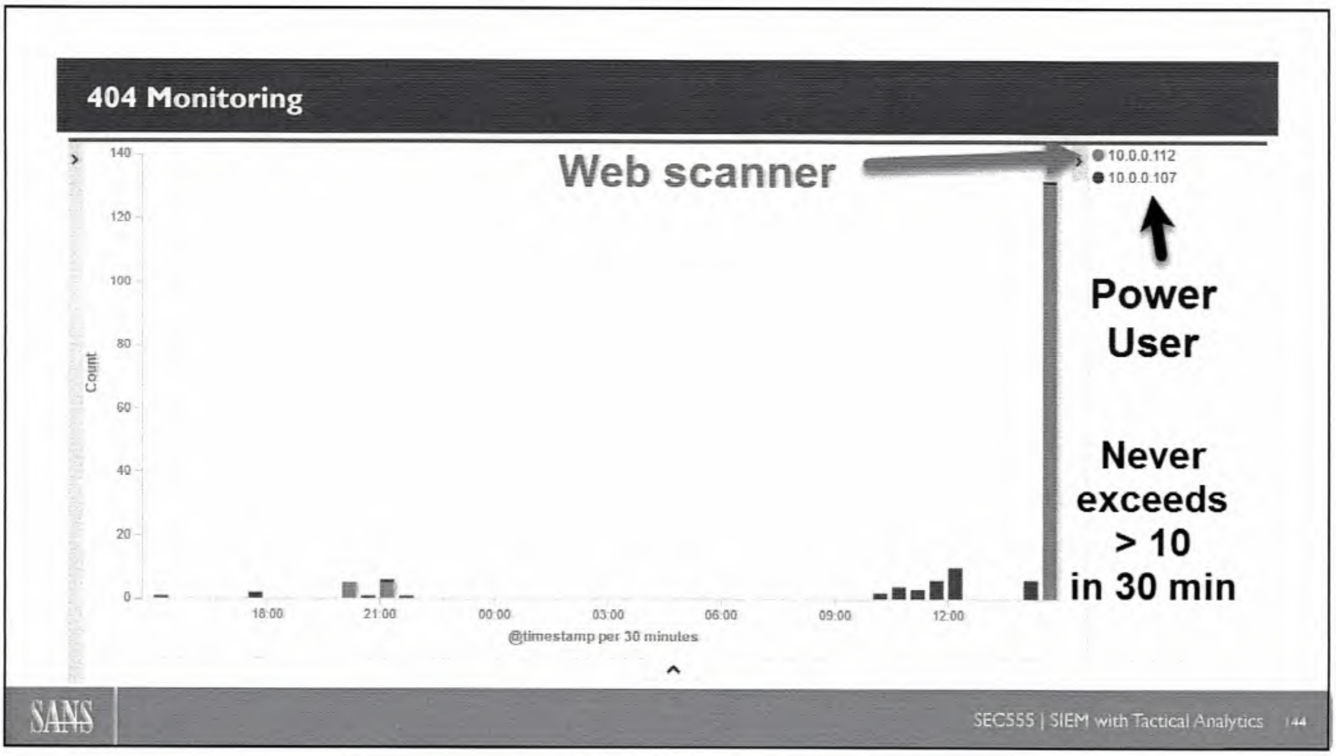
The requested URL /sdlkj was not found on this server.

Apache/2.4.18 (Ubuntu) Server at www.labmeinc.com Port 80

These generate lots of 404 messages

404 Not Found

The status code 404 is very similar to a DNS NXDOMAIN response. Both effectively mean something was requested that cannot be found. Monitoring for 404 errors will regularly point out misconfigurations such as dead links on sites. However, large amounts of 404 can point out more malicious activity such as from a vulnerability scanner or someone spidering a site.



404 Monitoring

Web scanner

Power User

Never exceeds > 10 in 30 min

404 Monitoring

This slide demonstrates basic monitoring of 404 status codes. In this example, 10.0.0.107 is a power user regularly browsing the internet. That user occasionally received 404 errors (behind the scenes) from web pages with bad links. Looking at 10.0.0.107 over the last 24 hours, it does not exceed ten 404 responses per 30-minute time slot.

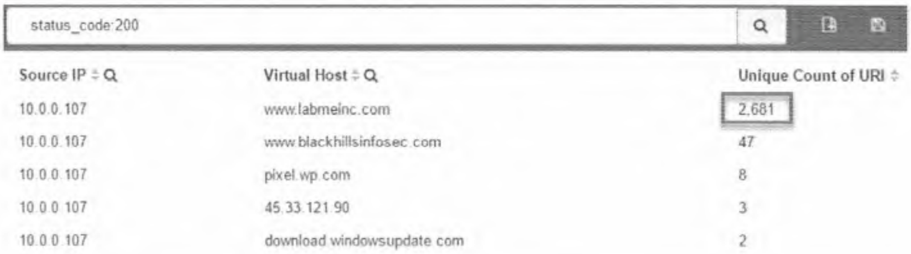
However, 10.0.0.112 exceeded more than a hundred 404 errors within 30 minutes. In this example, 10.0.0.112 is a system crawling a web server.

200 OK

Flip side of 404 is 200, which means that everything is OK

- Too much of a good thing... is a bad thing

If someone is crawling or scanning a site, there will be lots of unique URIs accessed with a status code of 200



```
status_code:200
```

Source IP	Virtual Host	Unique Count of URI
10.0.0.107	www.labmeinc.com	2,681
10.0.0.107	www.blackhillsinfosec.com	47
10.0.0.107	pixel.wp.com	8
10.0.0.107	45.33.121.90	3
10.0.0.107	download.windowsupdate.com	2

200 OK

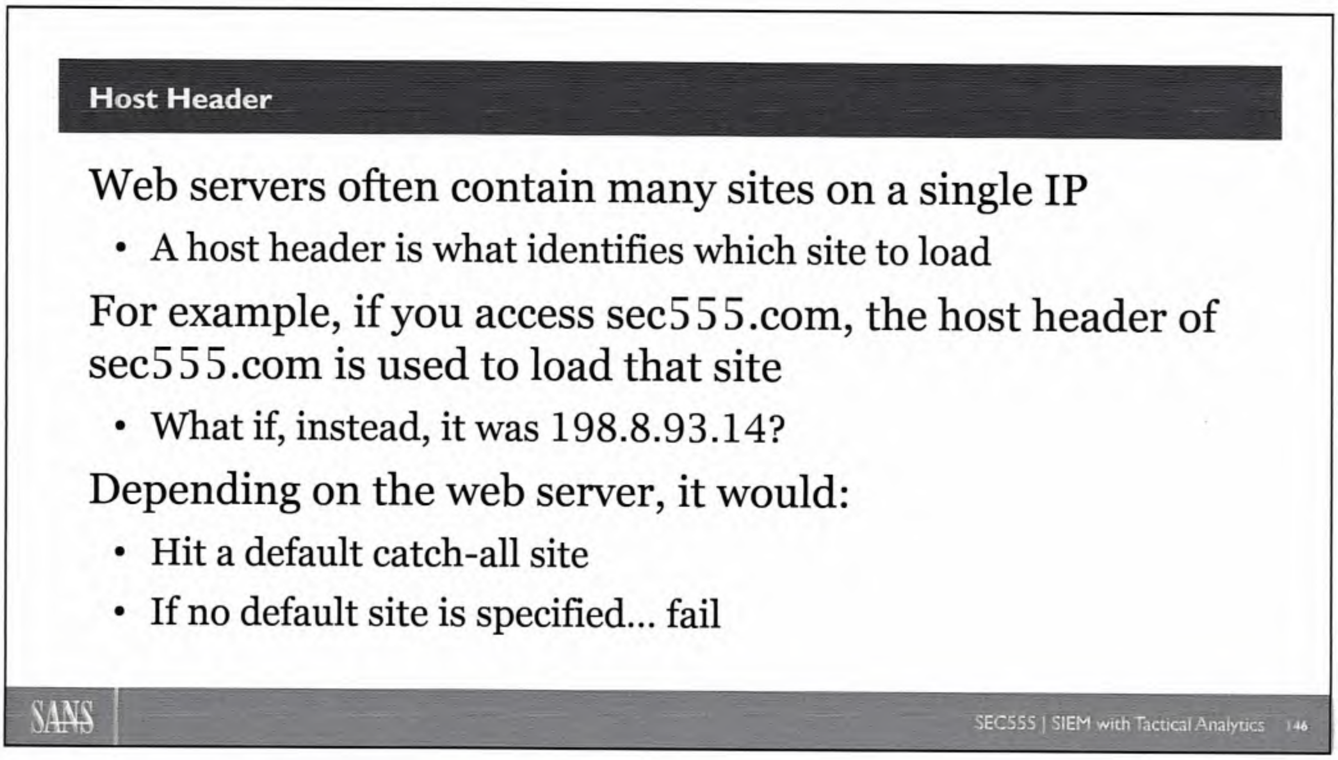
An age-old saying is "Too much of a good thing is a bad thing." This holds true—especially in SIEM analysis. Take, for instance, the status code 200 OK. This is absolutely a normal respond code given for the majority of HTTP requests. Yet if a system has an extremely high count of 200 OK responses, it may be odd. A bunch of 200 OK responses that belong to unique URIs is even odder.

In this slide, you can see that 10.0.0.107 has over 2,000 unique page requests against www.labmeinc.com. Now, some users are fast clickers, but not that fast. It is highly likely this is a spider scan against https://www.labmeinc.com/.

Oddly enough, www.labmeinc.com is actually a very small site with less than a hundred pages. Yet anything that scans it generates a lot of unique 200 OK status codes. The reason for this is the site deploys weblabyrinth,[1] developed by Ben Jackson and Mayhemic Labs. This is a simple PHP page that creates random URIs so anyone spidering a website will get stuck in an infinite scan loop. This type of defense has multiple benefits. First, the attacker's automated tools will not work without modification or being manually run. Second, it slows down the attacker. Last, it gives defenders an easy method to alert and respond.

Reference:

https://github.com/mayhemiclabs/weblabyrinth, https://sec555.com/5g



Host Header

Web servers often contain many sites on a single IP

- A host header is what identifies which site to load

For example, if you access sec555.com, the host header of sec555.com is used to load that site

- What if, instead, it was 198.8.93.14?

Depending on the web server, it would:

- Hit a default catch-all site
- If no default site is specified... fail

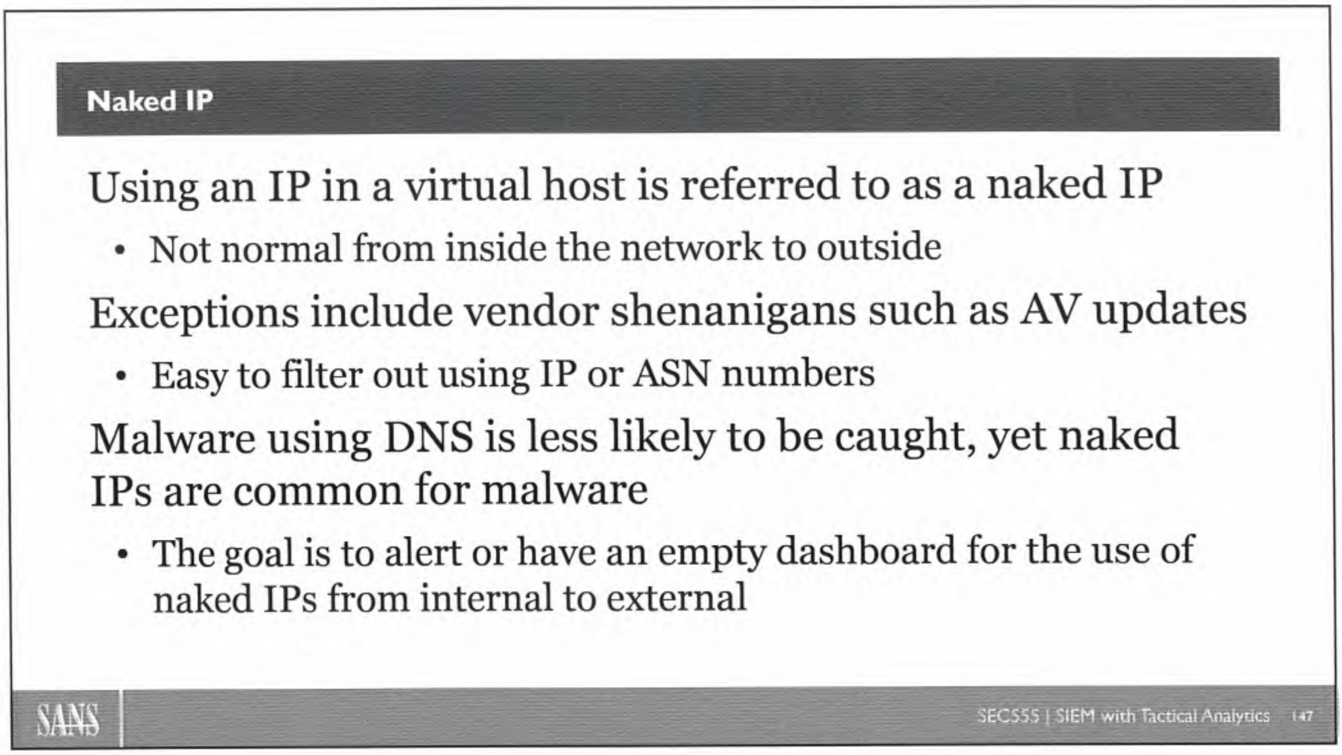
SANS

SEC555 | SIEM with Tactical Analytics 146

Host Header

In order for a single web server to host thousands of websites, the web server has to be able to tell which site a client is requesting. This is done using the host header specified in the request to load the right site. Usually, a domain name such as sec555.com is specified in the host header.

Note that sometimes a host header is referred to as the virtual host name. In logs, this is often written as virtual_host.



Naked IP

Using an IP in a virtual host is referred to as a naked IP

- Not normal from inside the network to outside

Exceptions include vendor shenanigans such as AV updates

- Easy to filter out using IP or ASN numbers

Malware using DNS is less likely to be caught, yet naked IPs are common for malware

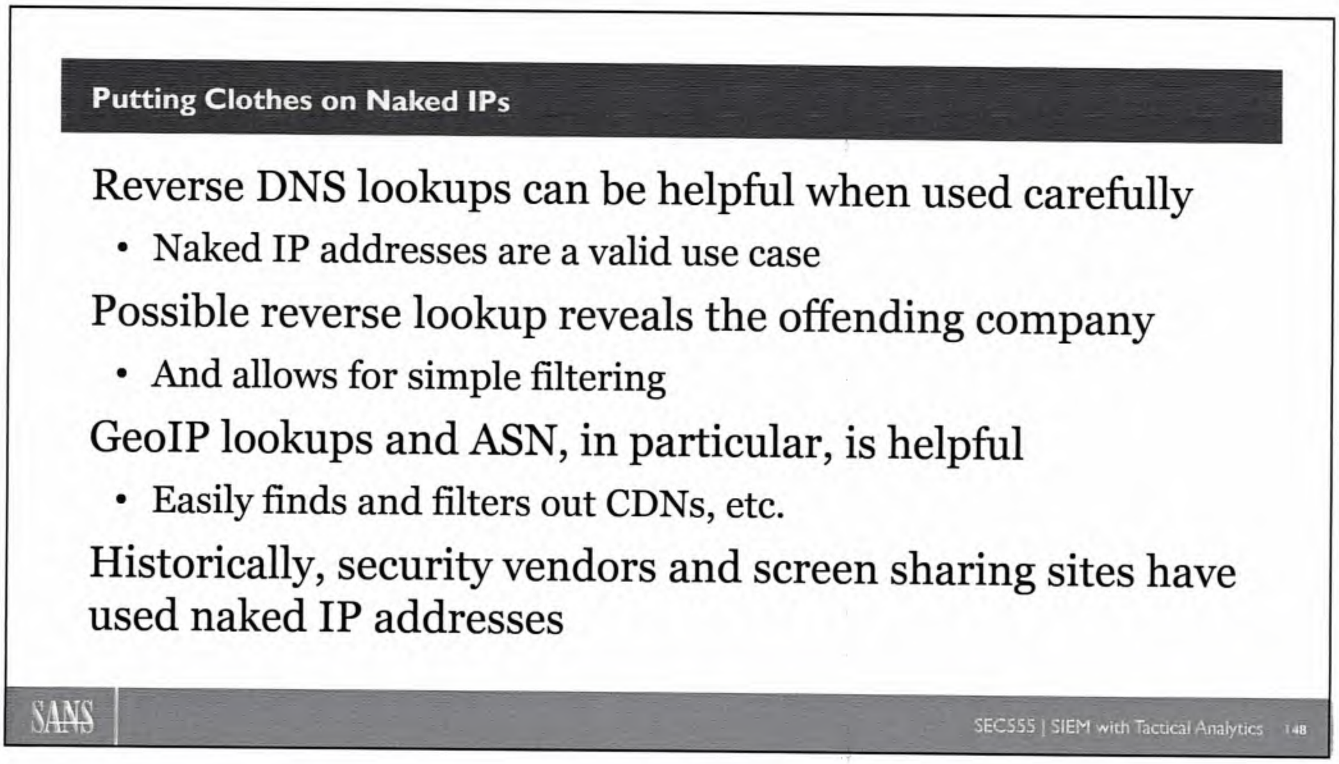
- The goal is to alert or have an empty dashboard for the use of naked IPs from internal to external

Naked IP

End users browse the internet using DNS names. Therefore, someone or something accessing the internet using an IP address is odd. Making an HTTP request using an IP address is sometimes referred to as a naked IP address request. Monitoring for naked IP address requests, typically, will point out a few vendor platforms. For instance, endpoint security systems have historically been notorious for downloading antivirus definitions or sending data over naked IP address requests.

Filtering out "valid" naked IP address requests is quite simple, especially if you can filter out the data based on ASN numbers. Oddly enough, a special filtering trick this author has found useful (in the event your system does not support ASN lookups) is using reverse DNS against naked IP addresses. Performing a reverse DNS lookup against an IP address specified in a naked IP address request will often point back to a culprit vendor or company and allow for easily filtering them out.

Even though a lot of malware uses DNS domains, it is still common to see malware use naked IP address requests. Thus, this technique works.



Putting Clothes on Naked IPs

Reverse DNS lookups can be helpful when used carefully

- Naked IP addresses are a valid use case

Possible reverse lookup reveals the offending company

- And allows for simple filtering

GeoIP lookups and ASN, in particular, is helpful

- Easily finds and filters out CDNs, etc.

Historically, security vendors and screen sharing sites have used naked IP addresses

SANS

SEC555 | SIEM with Tactical Analytics 148

Putting Clothes on Naked IPs

Oddly enough, reverse DNS can be extremely helpful in tracking down and filtering out IP addresses that are not mapped to a DNS request. For example, if a naked IP address is sent to a CDN such as Akamai, then performing a reverse DNS lookup of that IP will often result in something with Akamai in the DNS name. This can be great for filtering out the noise.

This works best when running conditionally. For example, if a naked IP address is discovered, then have the log pipeline add a reverse DNS field to the record. Tagging can also be used for better logic control. For example, if a naked IP comes in, you may want to add a tag of naked_ip. However, if a naked IP address comes in and a reverse DNS or ASN lookup shows that it is a CDN network, you could elect not to add the naked_ip tag or add naked_ip but with an extra tag such as cdn, noise, etc.

Using ASN is even more helpful as ASN can quickly and easily filter out CDN networks or other large IP blocks associated with certain businesses.

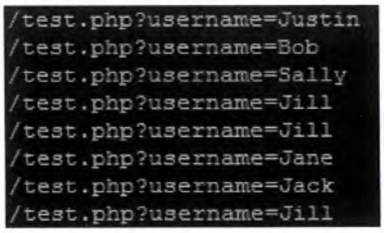
URL Lengths

Web page visits often fall within a short length

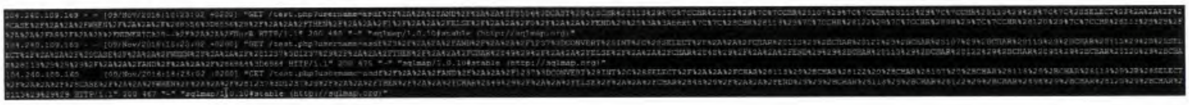
- Even with parameters, they tend to fall below a certain threshold

But under attacks, such as SQL injection, these URL lengths get very large

- Longest string within the right image is 23
- Longest string within failed SQL injection attack is 607



```
/test.php?username=Justin
/test.php?username=Bob
/test.php?username=Sally
/test.php?username=Jill
/test.php?username=Jill
/test.php?username=Jane
/test.php?username=Jack
/test.php?username=Jill
```

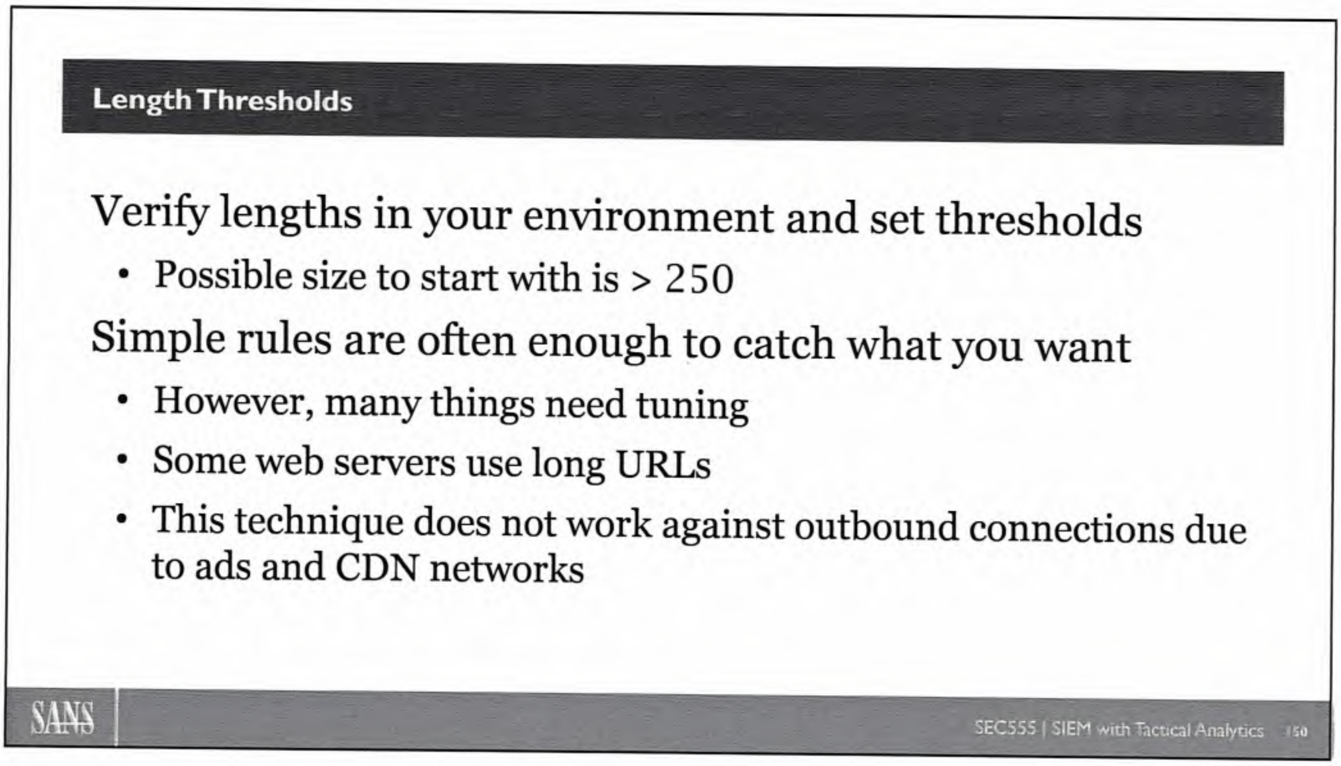


```
104.240.109.169 - - [09/Nov/2018:16:23:02 +0200] "GET /test.php?username=asd%2F%2A%2A%2FAND%2F%2A%2A%2F5559%3DCAST%28%28CHR%28113%29%7C%7CCHR%28122%29%7C%7CCHR%28107%29%7C%7CCHR%28113%29%7C%7CCHR%28113%29%29%7C%7C%28SELECT%2F%2A%2A%2F%2
8CASE%2F%2A%2A%2FWHEN%2F%2A%2A%2F%286556%3D6556%29%2F%2A%2A%2FTHEN%2F%2A%2A%2F1%2F%2A%2A%2FELSE%2F%2A%2A%2F0%2F%2A%2A%2FEND%29%29%3A%3Atext%7C%7C%28CHR%28113%29%7C%7CCHR%28122%29%7C%7CCHR%2898%29%7C%7CCHR%28120%29%7C%7CCHR%28113%29%29%2
9%2F%2A%2A%2FAS%2F%2A%2A%2FNUMERIC%29--%2F%2A%2A%2FHUcB HTTP/1.1" 200 488 "-" "sqlmap/1.0.10#stable (http://sqlmap.org)"
104.240.109.169 - - [09/Nov/2018:16:23:02 +0200] "GET /test.php?username=asd%2F%2A%2A%2FAND%2F%2A%2A%2F1237%3DCONVERT%28INT%2C%28SELECT%2F%2A%2A%2FCHAR%28113%29%2BCHAR%28122%29%2BCHAR%28107%29%2BCHAR%28113%29%2BCHAR%28113%29%2B%28SEL
ECT%2F%2A%2A%2F%28CASE%2F%2A%2A%2FWHEN%2F%2A%2A%2F%281237%3D1237%29%2F%2A%2A%2FTHEN%2F%2A%2A%2FCHAR%2849%29%2F%2A%2A%2FELSE%2F%2A%2A%2FCHAR%2848%29%2F%2A%2A%2FEND%29%29%2BCHAR%28113%29%2BCHAR%28122%29%2BCHAR%2898%29%2BCHAR%28120%29%2BCHA
R%28113%29%29%29%2F%2A%2A%2FAND%2F%2A%2A%2F%286984%3D6984 HTTP/1.1" 200 475 "-" "sqlmap/1.0.10#stable (http://sqlmap.org)"
104.240.109.169 - - [09/Nov/2018:16:23:02 +0200] "GET /test.php?username=asd%2F%2A%2A%2FAND%2F%2A%2A%2F1237%3DCONVERT%28INT%2C%28SELECT%2F%2A%2A%2FCHAR%28113%29%2BCHAR%28122%29%2BCHAR%28107%29%2BCHAR%28113%29%2BCHAR%28113%29%2B%28SELECT
%2F%2A%2A%2F%28CASE%2F%2A%2A%2FWHEN%2F%2A%2A%2F%281237%3D1237%29%2F%2A%2A%2FTHEN%2F%2A%2A%2FCHAR%2849%29%2F%2A%2A%2FELSE%2F%2A%2A%2FCHAR%2848%29%2F%2A%2A%2FEND%29%29%2BCHAR%28113%29%2BCHAR%28122%29%2BCHAR%2898%29%2BCHAR%28120%29%2BCHAR%2
8113%29%29%29 HTTP/1.1" 200 467 "-" "sqlmap/1.0.10#stable (http://sqlmap.org)"
```

URL Lengths

A technique that works fairly well on internal or DMZ facing web servers is simple URL length checks. Oftentimes, the length of URLs tends to fall below a certain threshold such as 100 characters in length. Yet, under attack, the URLs get bloated and large.

In the slide above, a failed SQL injection attack had a maximum length of 607. Note that this was a failed attack. A successful attack is likely to have even longer URLs. The technique of monitoring URL lengths is often used by Web Application Firewalls. However, many organizations do not have a web application firewall. For maximum effectiveness, a web application firewall must be properly configured. Those organizations deploying WAFs are often only protecting a subset of their web servers.



Length Thresholds

Verify lengths in your environment and set thresholds

- Possible size to start with is > 250

Simple rules are often enough to catch what you want

- However, many things need tuning
- Some web servers use long URLs
- This technique does not work against outbound connections due to ads and CDN networks

SANS

SEC555 | SIEM with Tactical Analytics 150

Length Thresholds

For URL length checks to be most effective, a threshold needs to be set and monitored. A possible starting size to monitor is 250 characters in length. This is because different sites have different use cases, and some may exceed 100 characters while others may always be less than 50 characters.

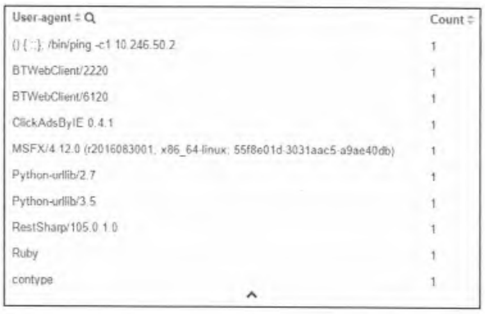
This technique is also intended for monitoring organization-controlled web servers. This could work for all HTTP connections, but online ads tend to have really long parameters for tracking end user activity. If you take the time to filter out URLs from ads, this technique could be applied across all HTTP URLs.

User Agents

User agents are a high value-add with little work

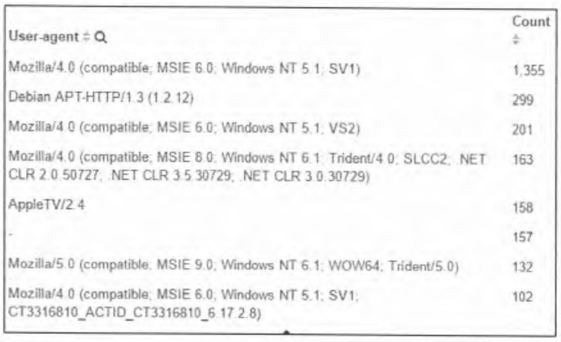
- Used to identify the client connecting to a web server

Malware often has typos or is blatantly obvious



User-agent	Count
() { :;}; /bin/ping -c1 10.246.50.2	1
BTWebClient/2220	1
BTWebClient/6120	1
ClickAdsByIE 0.4.1	1
MSFX/4.12.0 (r2016083001; x86_64-linux; 55f8e01d-3031aac5-a9ae40db)	1
Python-urllib/2.7	1
Python-urllib/3.5	1
RestSharp/105.0.1.0	1
Ruby	1
contype	1

VS.

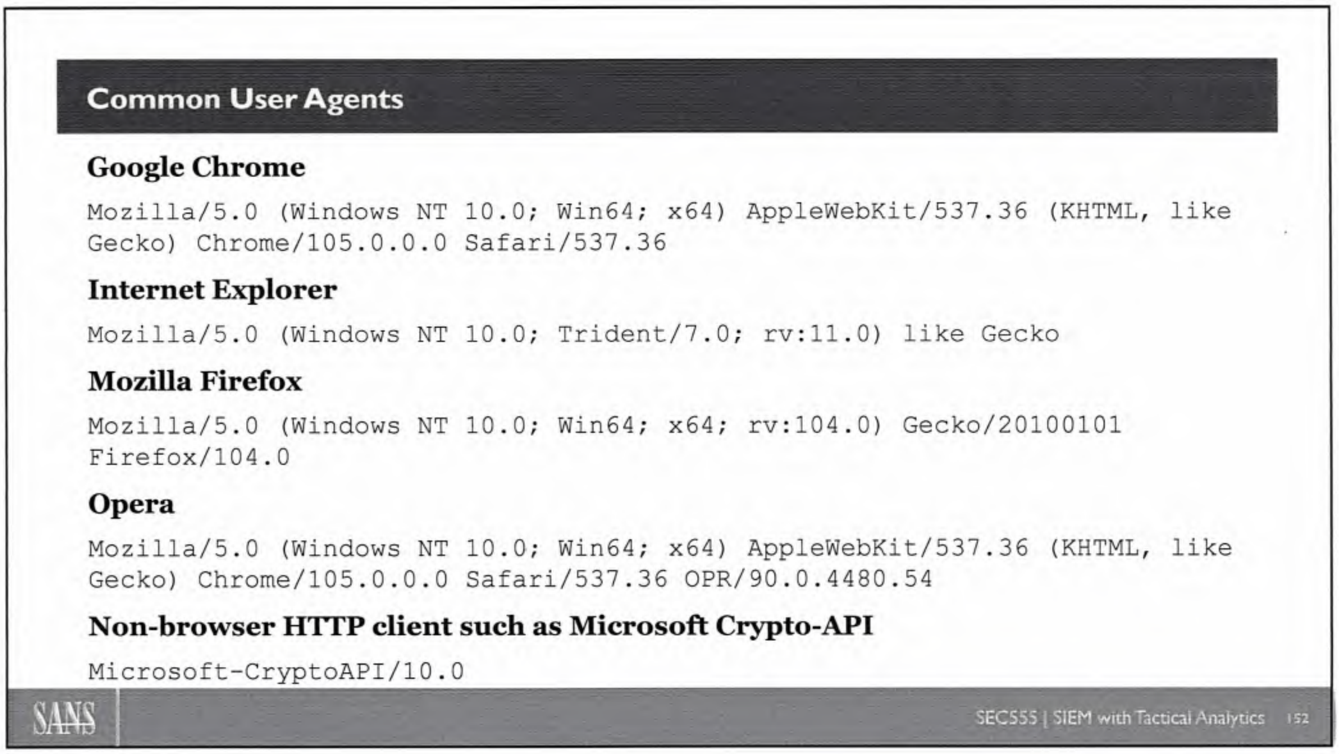


User-agent	Count
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)	1,355
Debian APT-HTTP/1.3 (1.2.12)	299
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; VS2)	201
Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729)	163
AppleTV/2.4	158
-	157
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)	132
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; CT3316810_ACTID_CT3316810_6.17.2.8)	102

User Agents

An HTTP user agent is used to identify the web client that is used to connect to a resource. It can be altered similarly to how a MAC address is supposed to remain constant; yet it can be changed in software. However, this still remains one of the best ways to catch malware. Yes, an advanced adversary may change this. However, this is often overlooked or, even when it is changed, it often has typos in it. Some malware is bold enough to put the name of the malware in the user agent.

The two tables in this slide show the most common user agents (on the right) and the least common (on the left).



Common User Agents

Google Chrome

```
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/105.0.0.0 Safari/537.36
```

Internet Explorer

```
Mozilla/5.0 (Windows NT 10.0; Trident/7.0; rv:11.0) like Gecko
```

Mozilla Firefox

```
Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:104.0) Gecko/20100101
Firefox/104.0
```

Opera

```
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/105.0.0.0 Safari/537.36 OPR/90.0.4480.54
```

Non-browser HTTP client such as Microsoft Crypto-API

```
Microsoft-CryptoAPI/10.0
```

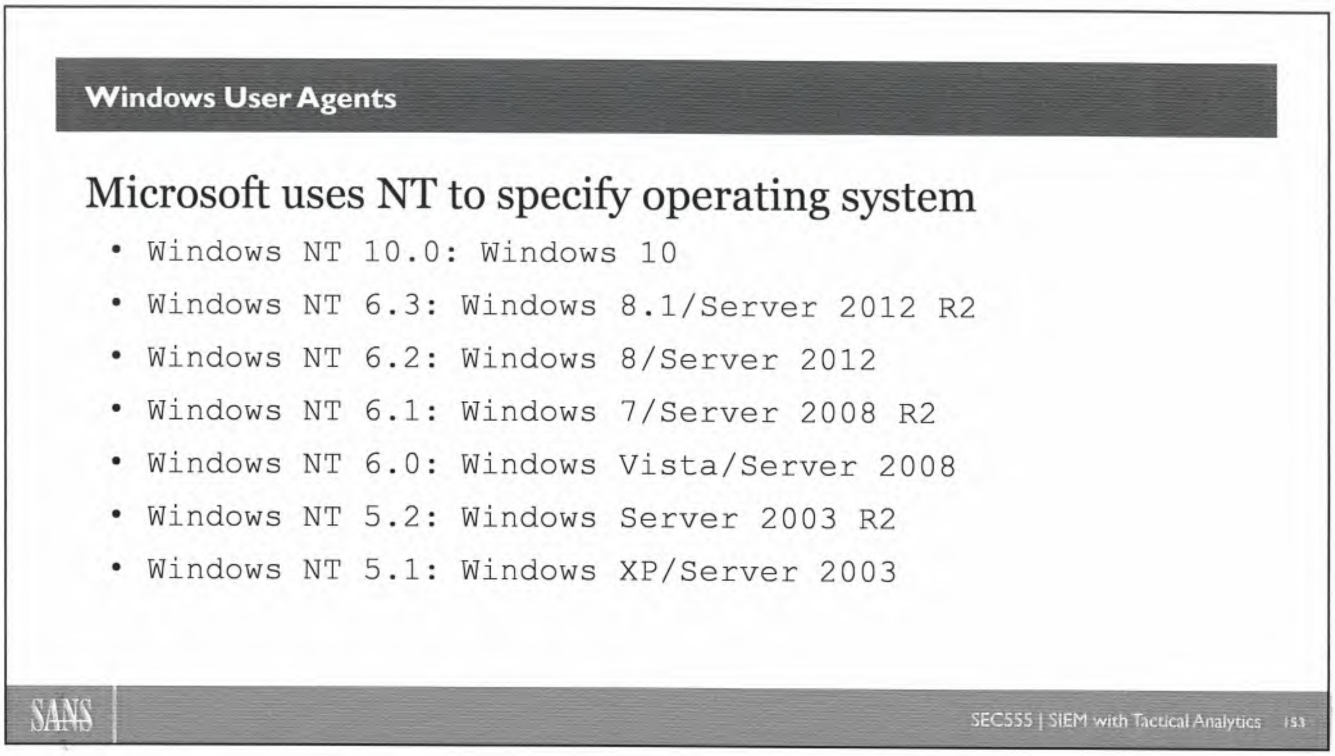
Common User Agents

This slide represents what user agents look like using modern internet browsers, as well as a common Microsoft user agent used for crypto functions. All of these were captured on a Windows 10 System.

User agents tend to be verbose in giving out information. For instance, Windows NT specifies the operating system version. Win64 or x64 specifies that the system is running a 64-bit operating system. Application versions such as Chrome/105.0.0.0 identify exactly what version of Chrome is being used. Some user agents will also include other details such as the .NET version a system has installed.

Reference:

https://www.whatismybrowser.com/guides/the-latest-user-agent/, http://sec555.com/114



Windows User Agents

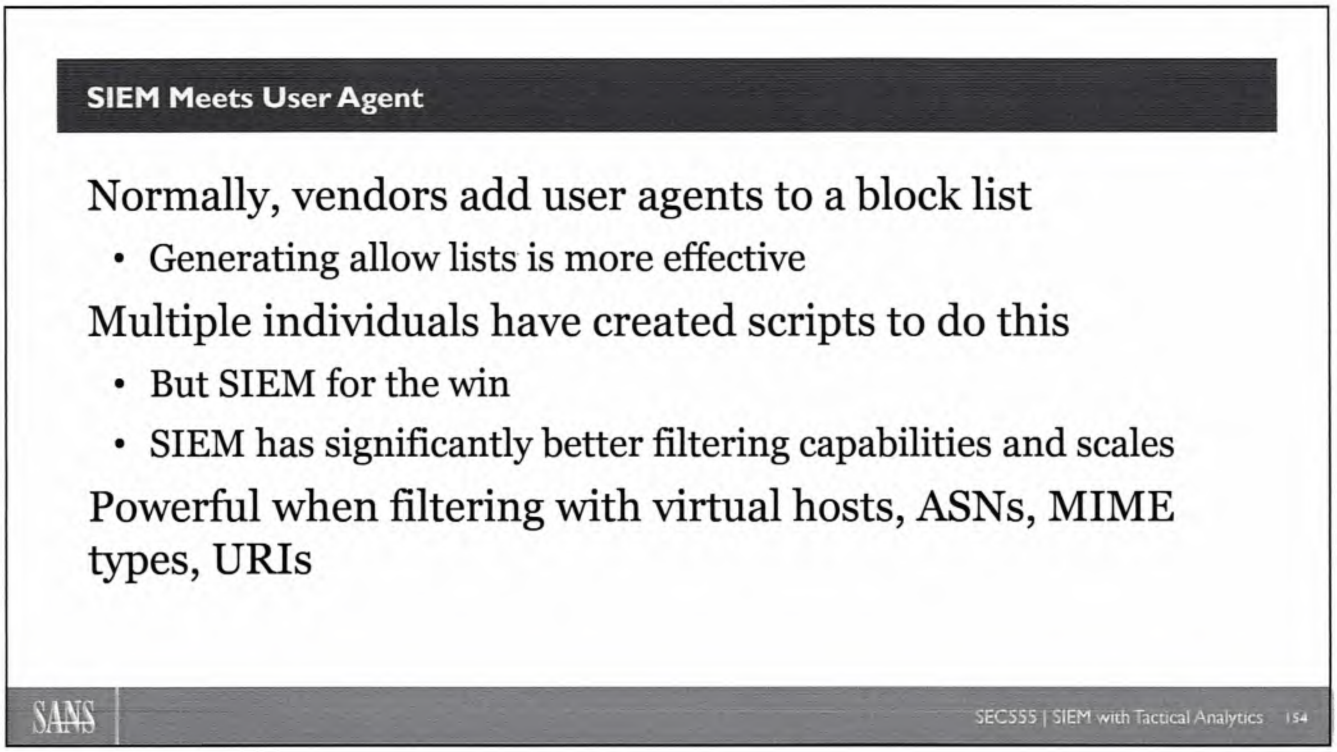
Microsoft uses NT to specify operating system

- `Windows NT 10.0: Windows 10`
- `Windows NT 6.3: Windows 8.1/Server 2012 R2`
- `Windows NT 6.2: Windows 8/Server 2012`
- `Windows NT 6.1: Windows 7/Server 2008 R2`
- `Windows NT 6.0: Windows Vista/Server 2008`
- `Windows NT 5.2: Windows Server 2003 R2`
- `Windows NT 5.1: Windows XP/Server 2003`

SEC555 | SIEM with Tactical Analytics 153

Windows User Agents

The Windows NT setting in a user agent identifies the version of Windows running. This can be helpful for multiple things. For example, if an attack that would work against XP ran against something with an NT version of 6+, then it most likely would not succeed. This can also be helpful for identifying unauthorized assets or assets that have not been upgraded yet.



SIEM Meets User Agent

Normally, vendors add user agents to a block list

- Generating allow lists is more effective

Multiple individuals have created scripts to do this

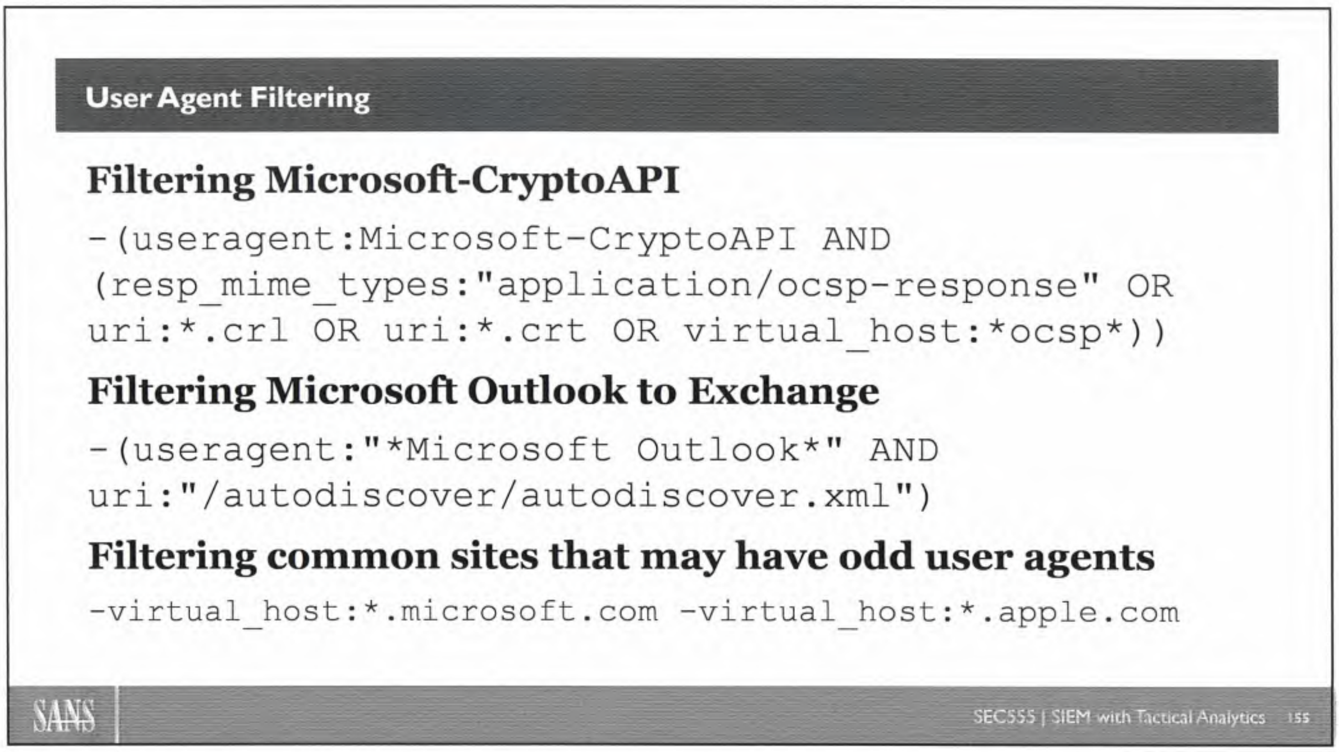
- But SIEM for the win
- SIEM has significantly better filtering capabilities and scales

Powerful when filtering with virtual hosts, ASNs, MIME types, URIs

SIEM Meets User Agent

Unfortunately, security products have a tendency to play whack-a-mole when evil is discovered. This is especially true surrounding user agents. If a bad user agent is discovered, it gets on the block list. However, a much more effective approach is generating an allow list of user agents.

The problem is that many organizations do not know how to generate an allow list. The fear is that it will be overly complex and hard to maintain. This is where the SIEM comes in for a win. A SIEM allows for easy slicing and dicing of data. Therefore, allow list techniques become easier. For example, you can exclude all user agents accessing certain ASN numbers or URIs.



User Agent Filtering

Filtering Microsoft-CryptoAPI

```
-(useragent:Microsoft-CryptoAPI AND
(resp_mime_types:"application/ocsp-response" OR
uri:*.crl OR uri:*.crt OR virtual_host:*ocsp*))
```

Filtering Microsoft Outlook to Exchange

```
-(useragent:"*Microsoft Outlook*" AND
uri:"/autodiscover/autodiscover.xml")
```

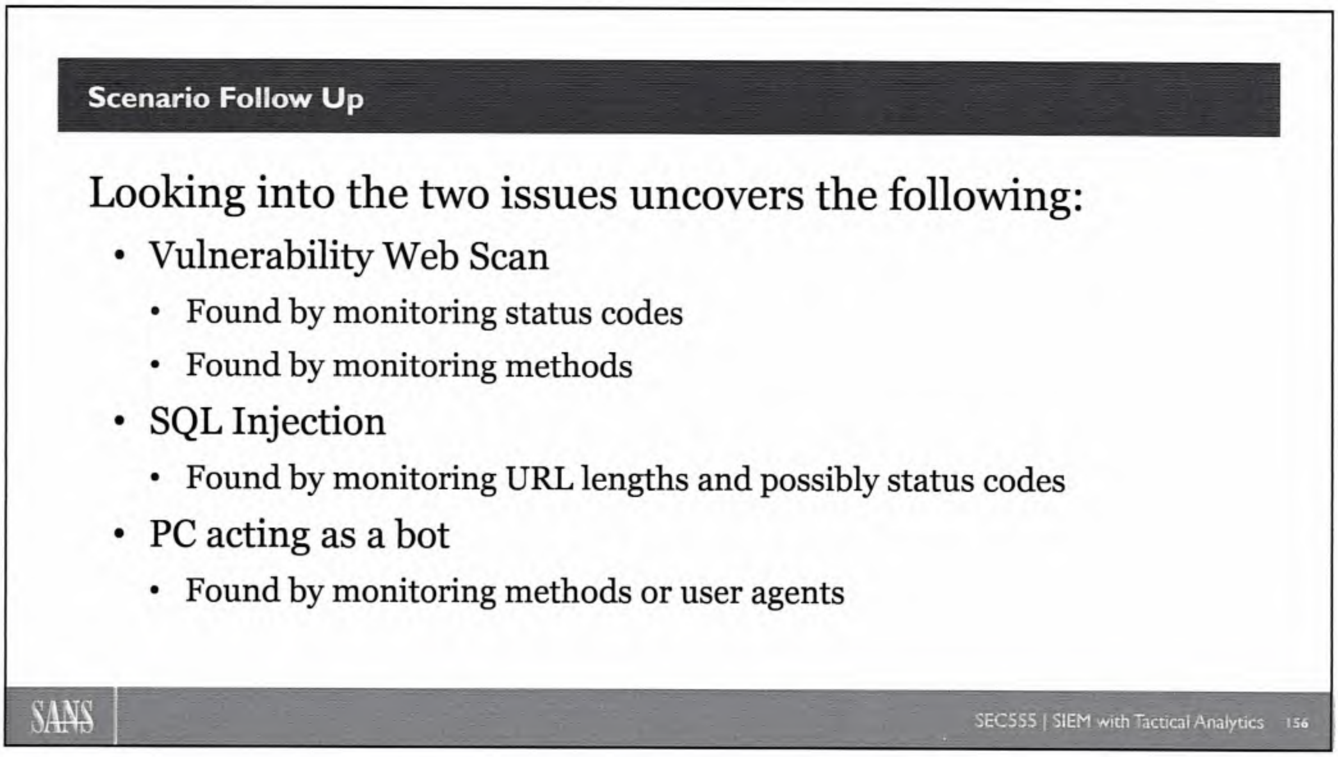
Filtering common sites that may have odd user agents

```
-virtual_host:*.microsoft.com -virtual_host:*.apple.com
```

User Agent Filtering

This slide represents a sampling of ways to help establish a successful allow list filter. It includes ignoring domains such as *.microsoft.com or *.apple.com so that any special user agents talking to those are ignored. It also includes specific use cases such as excluding Microsoft Outlook talking to Microsoft Exchange.

The more advanced use cases are allow listing a user agent itself. For example, the Microsoft-CryptoAPI user agent is used to perform crypto functions, such as looking up certificate revocation lists (CRL) and online certificate status protocol (OCSP) checks. Filtering out the ocsp-response MIME type, any URIs that end in CRL or CRT, and excluding all virtual_hosts with ocsp in the URL filters out Microsoft-CryptoAPI—but only for seemingly legitimate use cases. This means malware could be detected if it tries to hide as Microsoft-CryptoAPI.



Scenario Follow Up

Looking into the two issues uncovers the following:

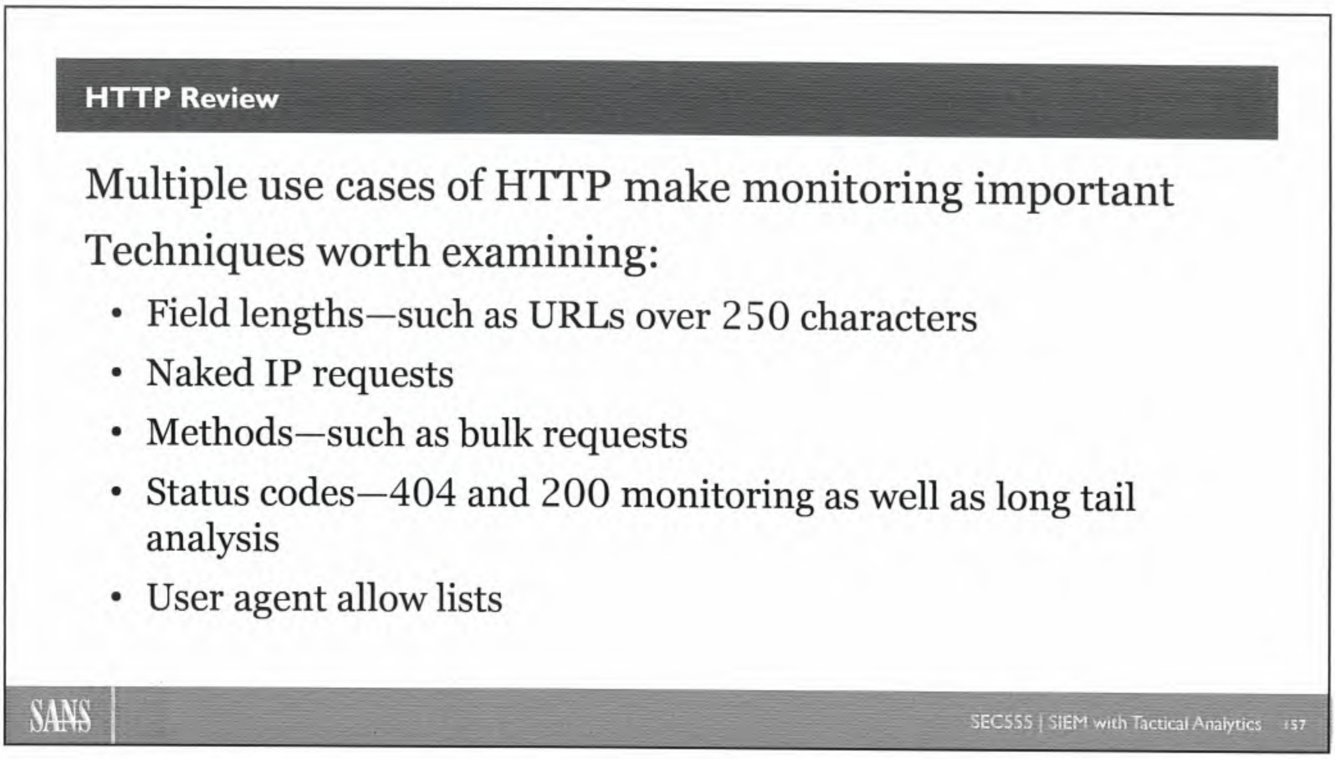
- Vulnerability Web Scan
  - Found by monitoring status codes
  - Found by monitoring methods
- SQL Injection
  - Found by monitoring URL lengths and possibly status codes
- PC acting as a bot
  - Found by monitoring methods or user agents

SANS

SEC555 | SIEM with Tactical Analytics 156

Scenario Follow Up

Previously, the CIO, Desiree, requested that Dr. James Hart's computer be looked into along with a crashed web server. Using the HTTP techniques supplied would have detected both the web server attack as well as the infection on Dr. Hart's computer. While in this scenario the detection was given more ad hoc, these techniques work best when monitored continuously. Doing so also would allow for earlier detection, plus an added bonus of making the security team look great to the CIO.



HTTP Review

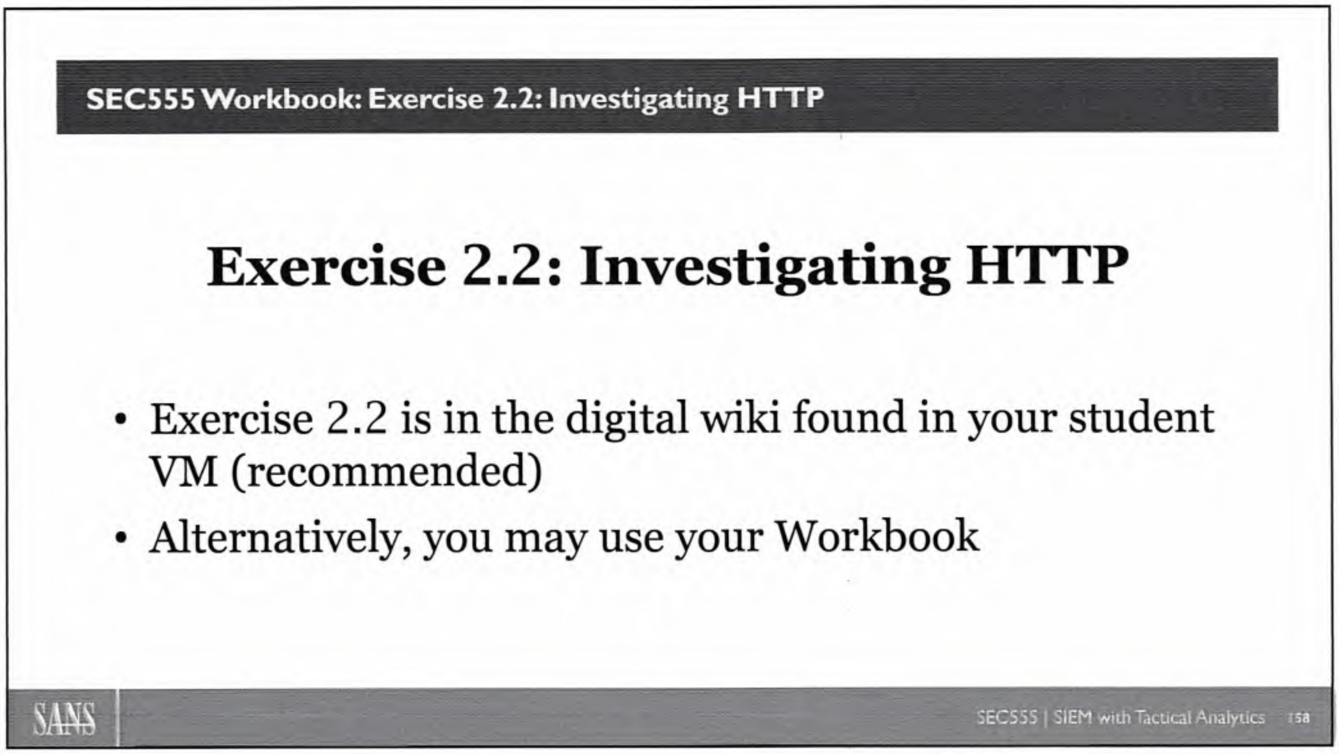
Multiple use cases of HTTP make monitoring important

Techniques worth examining:

- Field lengths—such as URLs over 250 characters
- Naked IP requests
- Methods—such as bulk requests
- Status codes—404 and 200 monitoring as well as long tail analysis
- User agent allow lists

HTTP Review

This slide contains a recap of some of the techniques listed in response to HTTP monitoring. In truth, many more techniques are applicable. This list is a short list to get your mind engaged.

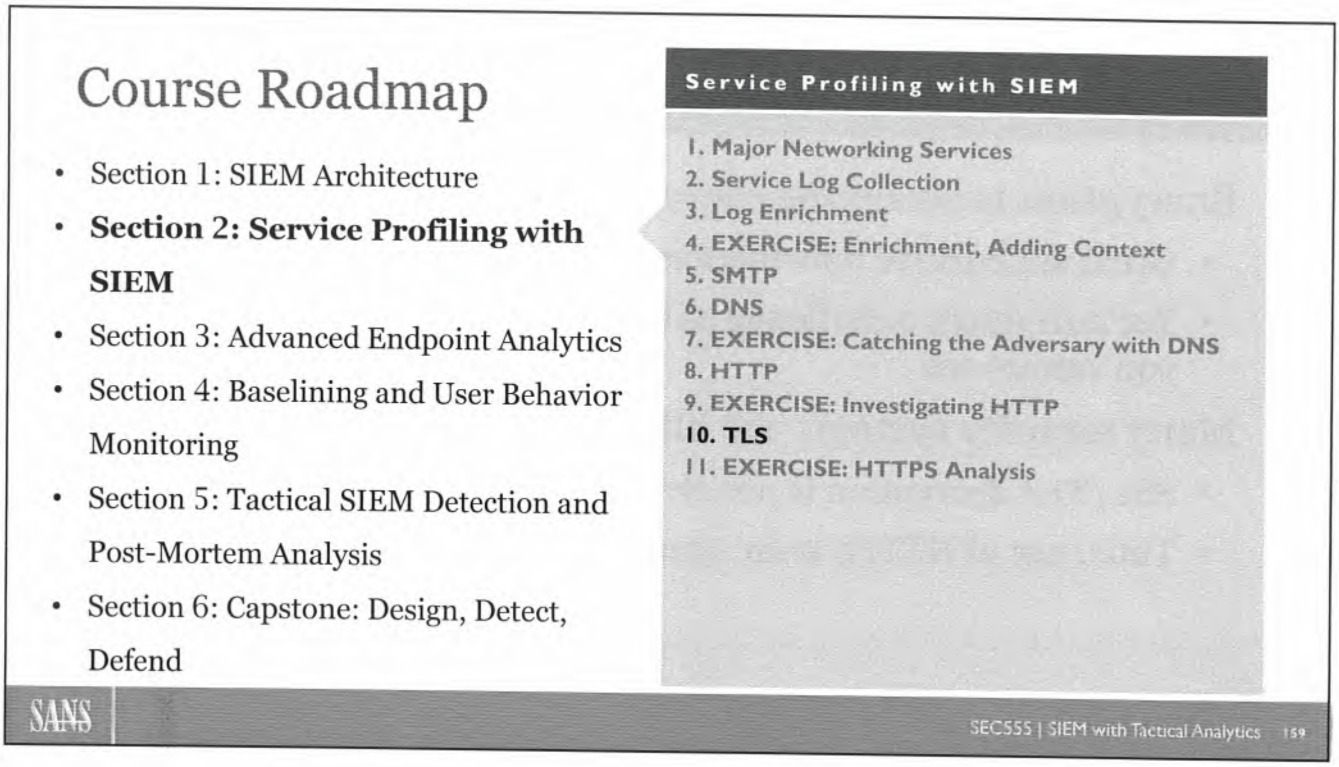


SEC555 Workbook: Exercise 2.2: Investigating HTTP

Exercise 2.2: Investigating HTTP

- Exercise 2.2 is in the digital wiki found in your student VM (recommended)
- Alternatively, you may use your Workbook

This page intentionally left blank.



Course Roadmap

- Section 1: SIEM Architecture
- **Section 2: Service Profiling with SIEM**
- Section 3: Advanced Endpoint Analytics
- Section 4: Baselining and User Behavior Monitoring
- Section 5: Tactical SIEM Detection and Post-Mortem Analysis
- Section 6: Capstone: Design, Detect, Defend

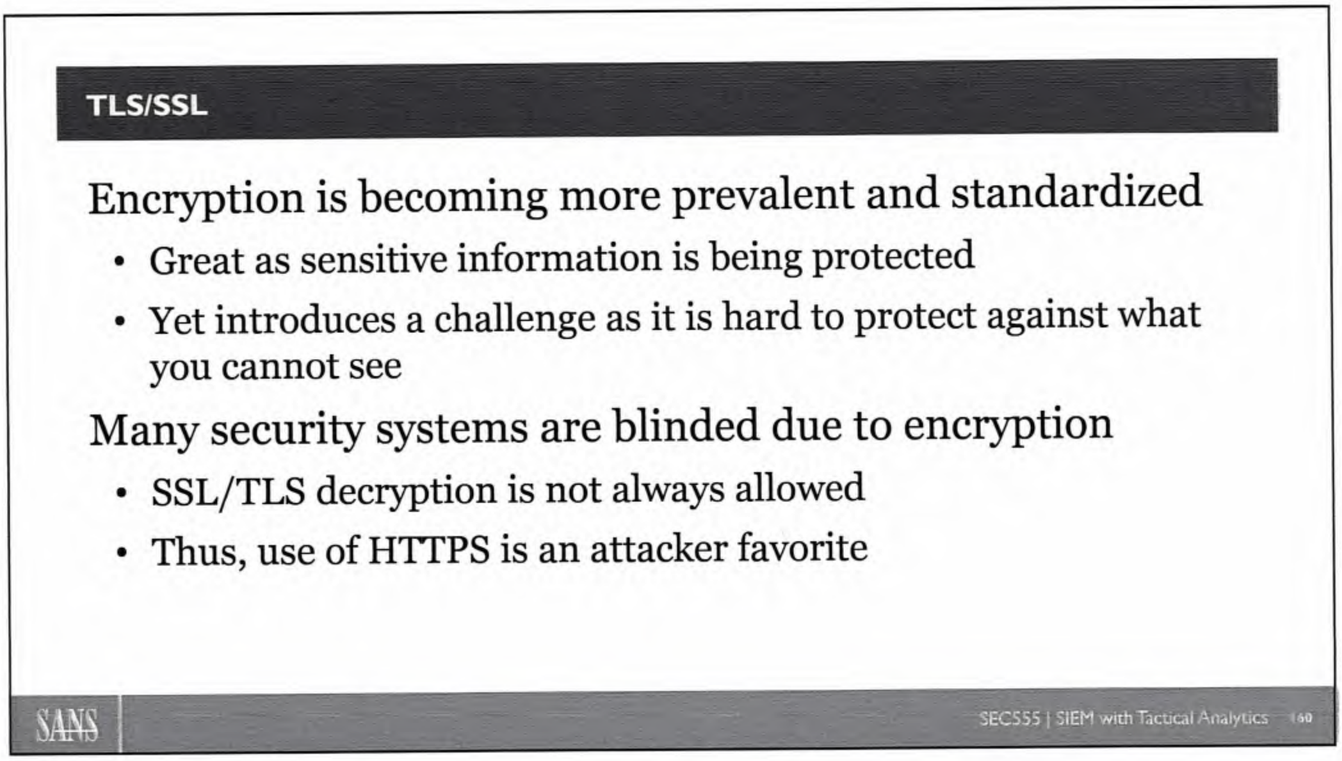
Service Profiling with SIEM

1. Major Networking Services
2. Service Log Collection
3. Log Enrichment
4. EXERCISE: Enrichment, Adding Context
5. SMTP
6. DNS
7. EXERCISE: Catching the Adversary with DNS
8. HTTP
9. EXERCISE: Investigating HTTP
10. **TLS**
11. EXERCISE: HTTPS Analysis

SANS

SEC555 | SIEM with Tactical Analytics 159

This page intentionally left blank.



TLS/SSL

Encryption is becoming more prevalent and standardized

- Great as sensitive information is being protected
- Yet introduces a challenge as it is hard to protect against what you cannot see

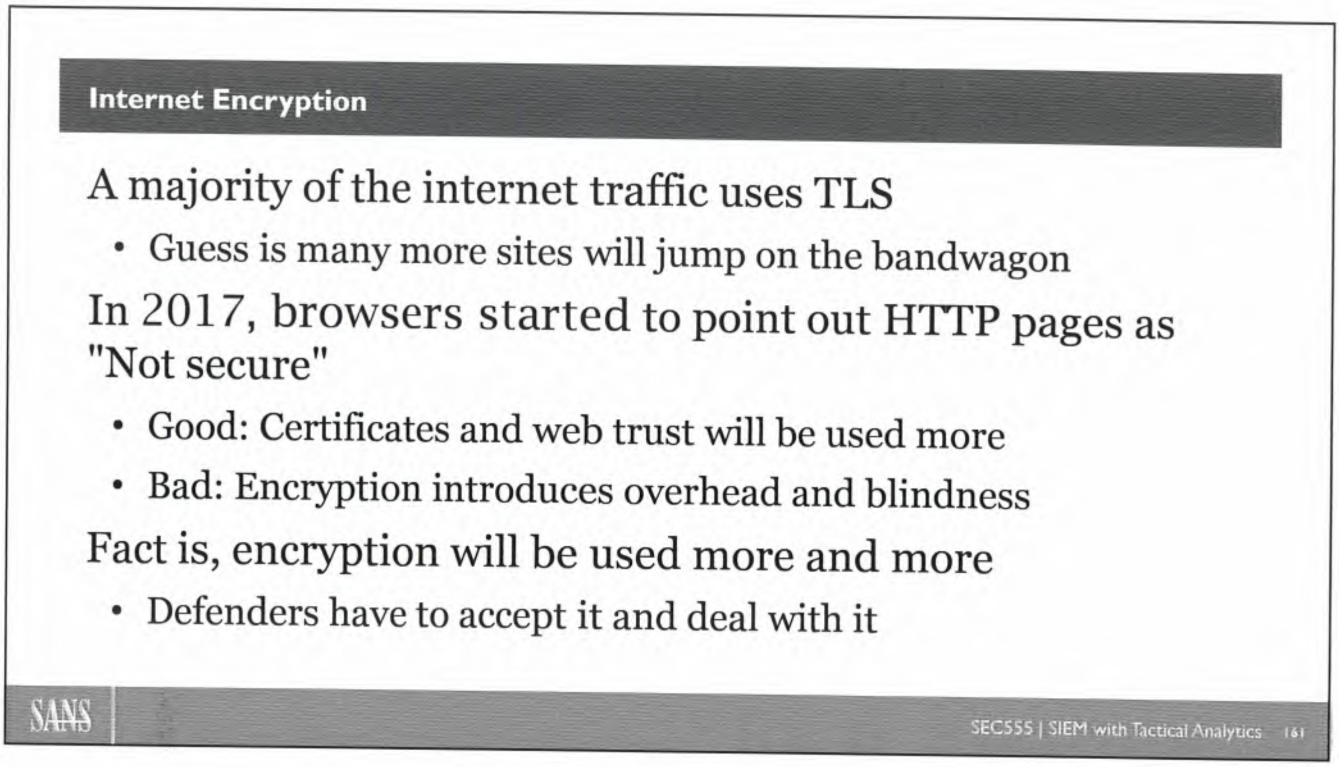
Many security systems are blinded due to encryption

- SSL/TLS decryption is not always allowed
- Thus, use of HTTPS is an attacker favorite

TLS/SSL

Many believe encryption is the end of all network security controls and the pushback to endpoint security. It definitely makes things more challenging, but it is not the end. Security is continuously shifting, and as a result, defenders have to come up with new strategies constantly.

This section is based on the fact that many organizations still do not have the authorization to perform SSL inspection. This can be due to a lack of senior management approval, or in some locales, legal restrictions. However, even if SSL inspection is disabled, these techniques still work and add further defense in depth.



Internet Encryption

A majority of the internet traffic uses TLS

- Guess is many more sites will jump on the bandwagon

In 2017, browsers started to point out HTTP pages as "Not secure"

- Good: Certificates and web trust will be used more
- Bad: Encryption introduces overhead and blindness

Fact is, encryption will be used more and more

- Defenders have to accept it and deal with it

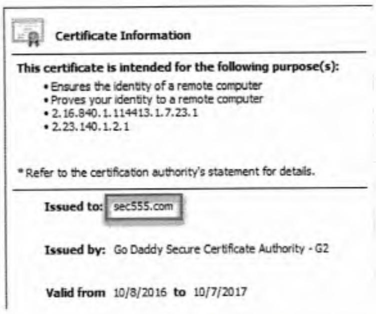
Internet Encryption

At this point, a majority of the internet uses encryption, and it is highly likely this trend will continue, though more aggressively. As defenders, we often do not get a decision in these matters, so we simply have to accept the change and figure out how it is going to affect our organizations. Encryption, like everything else, is a change. It is not the end of the world; it just simply requires adapting.

Certificate Details

With **TLS 1.2 and lower**, the payload is encrypted

- The certificate is not
- There are lots of details within a certificate



Certificate Information

This certificate is intended for the following purpose(s):

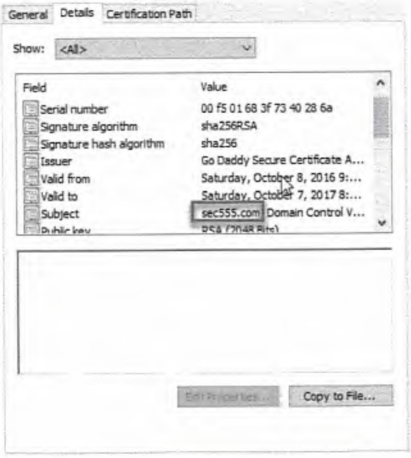
- Ensures the identity of a remote computer
- Proves your identity to a remote computer
- 2.16.840.1.114413.1.7.23.1
- 2.23.140.1.2.1

* Refer to the certification authority's statement for details.

Issued to: sec555.com

Issued by: Go Daddy Secure Certificate Authority - G2

Valid from 10/8/2016 **to** 10/7/2017

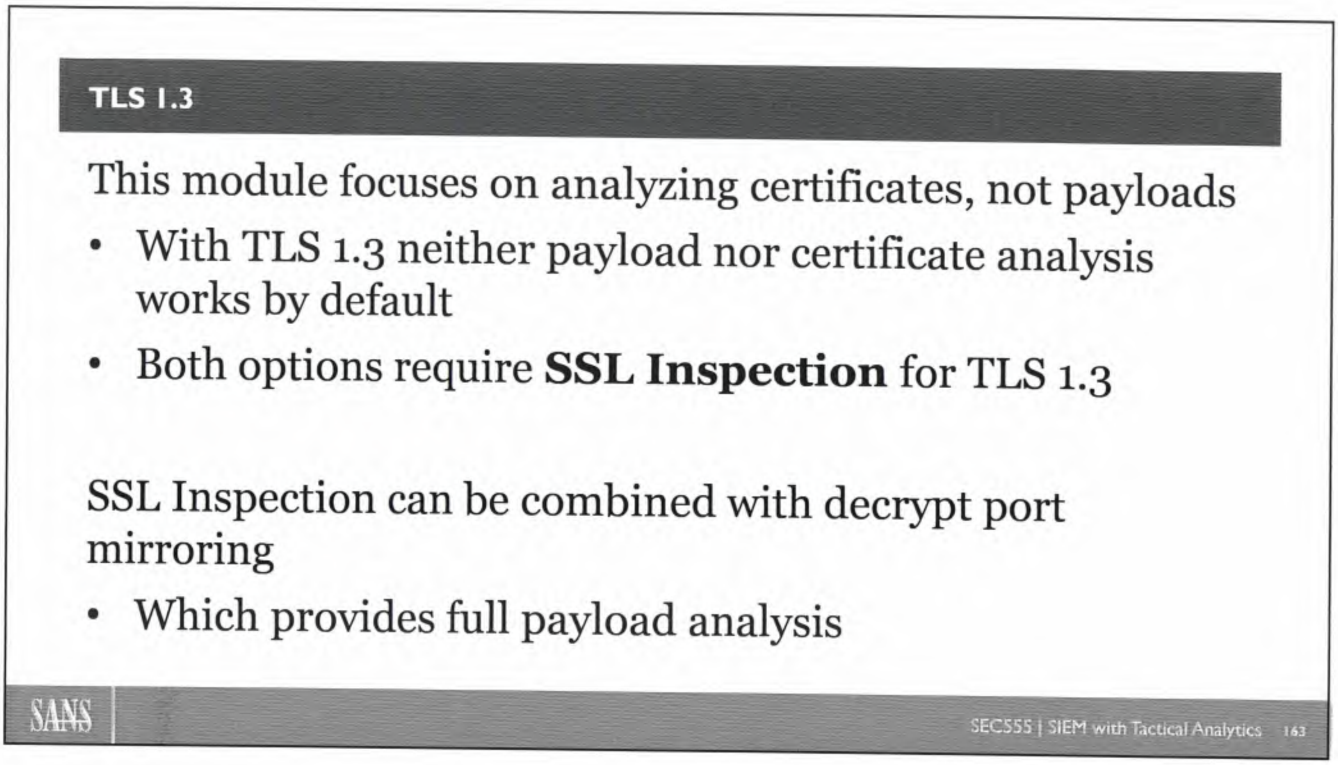


Show: <All>

Field	Value
Serial number	00 f5 01 68 3f 73 40 28 6a
Signature algorithm	sha256RSA
Signature hash algorithm	sha256
Issuer	Go Daddy Secure Certificate A...
Valid from	Saturday, October 8, 2016 9:...
Valid to	Saturday, October 7, 2017 8:...
Subject	sec555.com Domain Control V...
Public key	RSA (2048 Bits)

Certificate Details

When accessing a website using https:// using TLS versions 1.2 and lower, the contents being transported over the network are encrypted. However, the certificate information is not. This is by design and required for SSL or TLS to work. Because of this, a lot of information can be used within the certificates to identify malware.



TLS 1.3

This module focuses on analyzing certificates, not payloads

- With TLS 1.3 neither payload nor certificate analysis works by default
- Both options require **SSL Inspection** for TLS 1.3

SSL Inspection can be combined with decrypt port mirroring

- Which provides full payload analysis

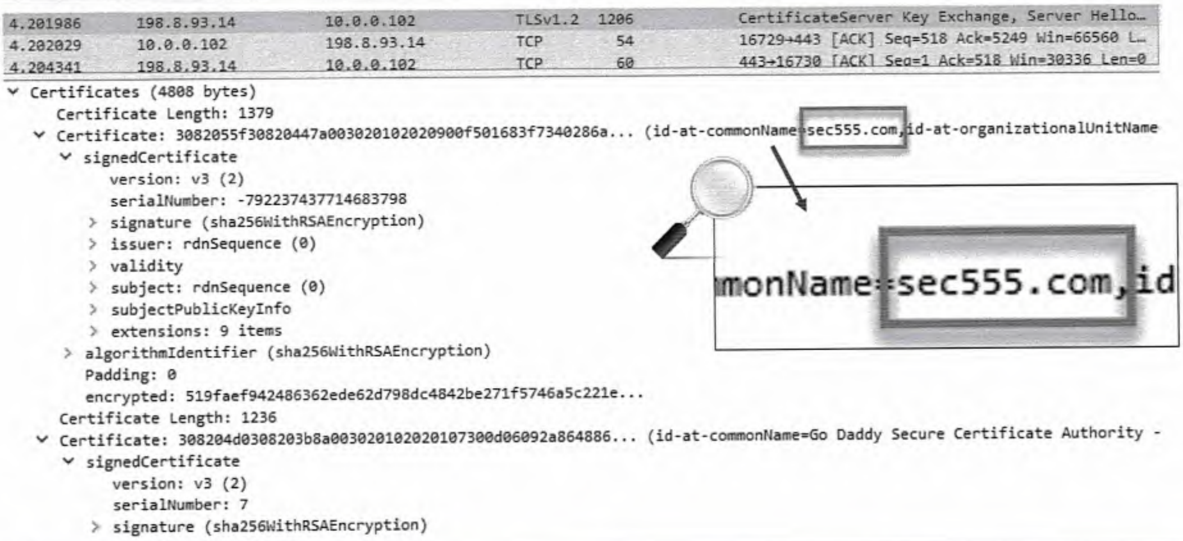
TLS 1.3

TLS 1.3 is a disruptive technology change. With TLS 1.3, the certificate information is no longer visible on the wire. Instead, TLS 1.3 provides confidentiality of payload information as well as the certificate information. By doing so, privacy is kept, and potential devices are unable to see where a user or machine is browsing. The problem is these blind organizations trying to analyze certificates to identify malware or other unauthorized use.

This module assumes that organizations are unable to use SSL Inspection, which provides the ability to see decrypted payload information. Due to privacy and compliance concerns, most organizations do not enable SSL Inspection. So, this module is based on analyzing high-level certificate information to find evil. However, this is not possible with TLS 1.3 traffic. With TLS 1.3, traffic organizations should strongly consider enabling SSL Inspection. Most commercial solutions do not require SSL Inspection to be enabled for all traffic. Instead, it can be allowed per IP or web category. For example, SSL Inspection can be disabled for banking but enabled for sites that are considered uncategorized or newly observed.

SSL Inspection can be further expanded by enabling decrypt port mirroring. This allows decrypted traffic to be mirrored to another device like a network security monitoring. By doing so, traffic can be fully inspected. For example, HTTPS traffic would generate HTTP logs if Zeek could see the decrypted traffic. The only caveat is you would need to tell Zeek that port 443 traffic is HTTP if it is set to ignore the port.

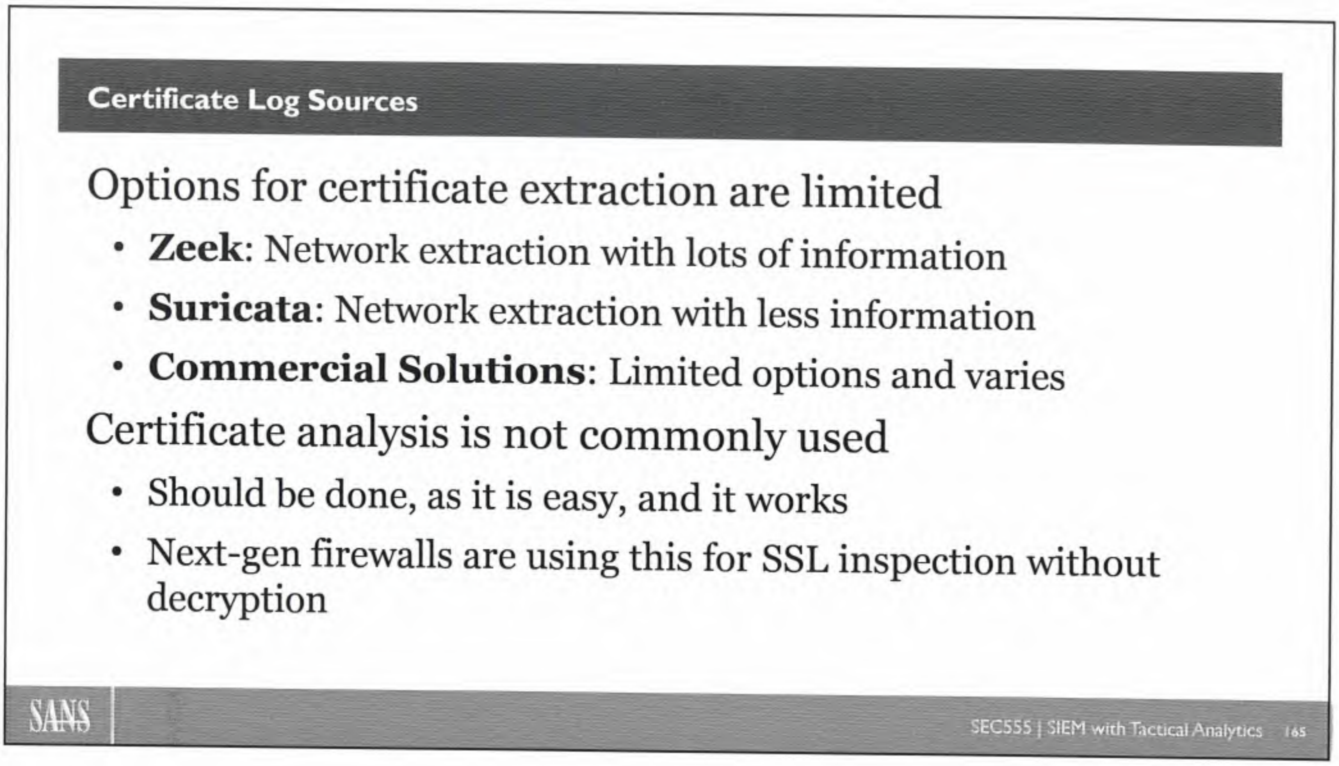
Certificates on the Wire



```
4.201986     198.8.93.14     10.0.0.102      TLSv1.2  1206     CertificateServer Key Exchange, Server Hello…
4.202029     10.0.0.102      198.8.93.14     TCP        54     16729→443 [ACK] Seq=518 Ack=5249 Win=66560 L…
4.204341     198.8.93.14     10.0.0.102      TCP        60     443→16730 [ACK] Seq=1 Ack=518 Win=30336 Len=0
v Certificates (4808 bytes)
    Certificate Length: 1379
  v Certificate: 3082055f30820447a003020102020900f501683f7340286a... (id-at-commonName=sec555.com,id-at-organizationalUnitName
    v signedCertificate
        version: v3 (2)
        serialNumber: -792237437714683798
      > signature (sha256WithRSAEncryption)
      > issuer: rdnSequence (0)
      > validity
      > subject: rdnSequence (0)
      > subjectPublicKeyInfo
      > extensions: 9 items
    > algorithmIdentifier (sha256WithRSAEncryption)
      Padding: 0
      encrypted: 519faef942486362ede62d798dc4842be271f5746a5c221e...
    Certificate Length: 1236
  v Certificate: 308204d0308203b8a003020102020107300d06092a864886... (id-at-commonName=Go Daddy Secure Certificate Authority -
    v signedCertificate
        version: v3 (2)
        serialNumber: 7
      > signature (sha256WithRSAEncryption)
```

Certificates on the Wire

This slide demonstrates looking at the certificate for sec555.com from within a packet capture using Wireshark. Because the certificate must be transmitted in cleartext, the certificate information is able to be seen fully on the wire.



Certificate Log Sources

Options for certificate extraction are limited

- **Zeek**: Network extraction with lots of information
- **Suricata**: Network extraction with less information
- **Commercial Solutions**: Limited options and varies

Certificate analysis is not commonly used

- Should be done, as it is easy, and it works
- Next-gen firewalls are using this for SSL inspection without decryption

Certificate Log Sources

The value of having certificate information is tremendous. Yet, there are not many options available for extracting log information on certificates. It is not that it is difficult or that it is computationally expensive. It is simply that techniques on analyzing certificates have not gained enough attention yet.

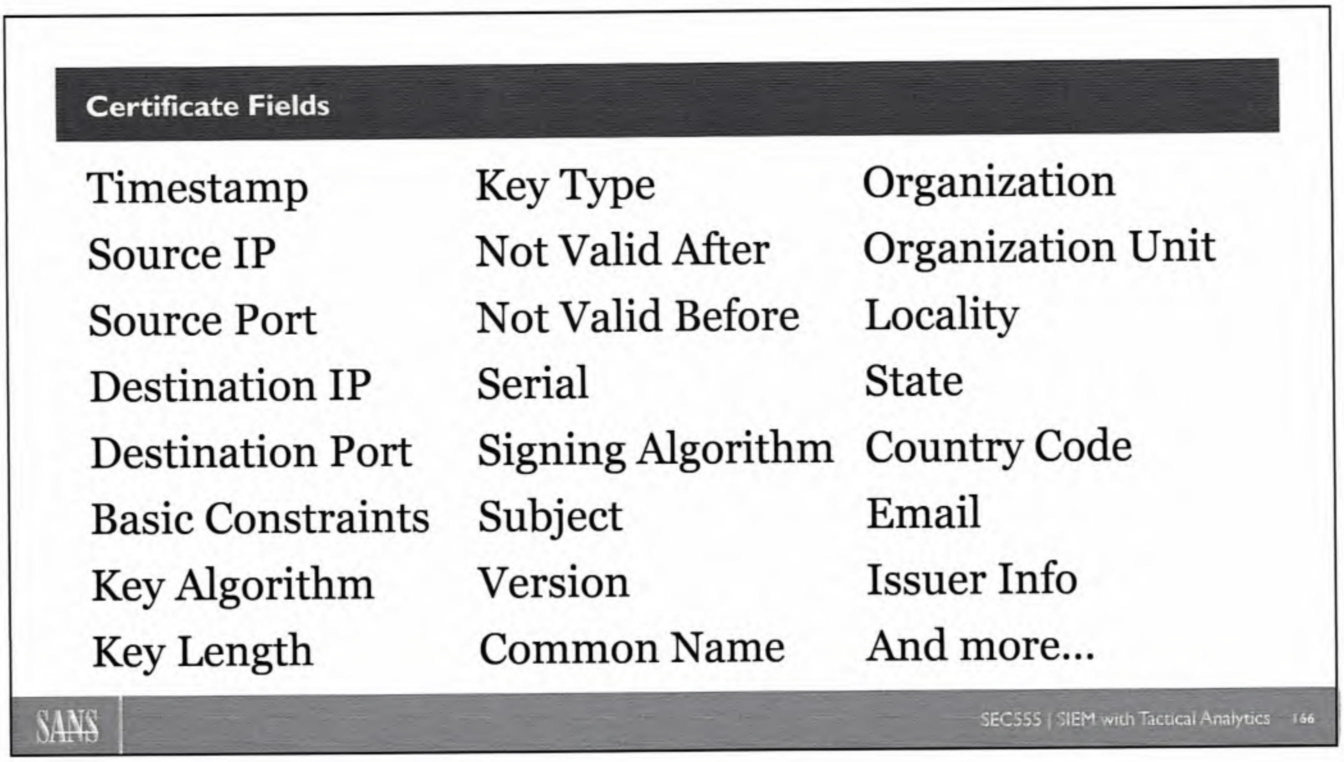
While it would be possible to log certificate information at a proxy or a next-gen firewall, these solutions typically do not offer that capability. Network extraction using Zeek or Suricata tend to be the primary ways to gather this information.

Next-gen firewalls are beginning to use this technique to catch malware. Certain SSL Inspection settings analyze the certificate information but do not decrypt the data.

References:

https://www.zeek.org, https://sec555.com/4h

https://suricata.io, https://sec555.com/4m



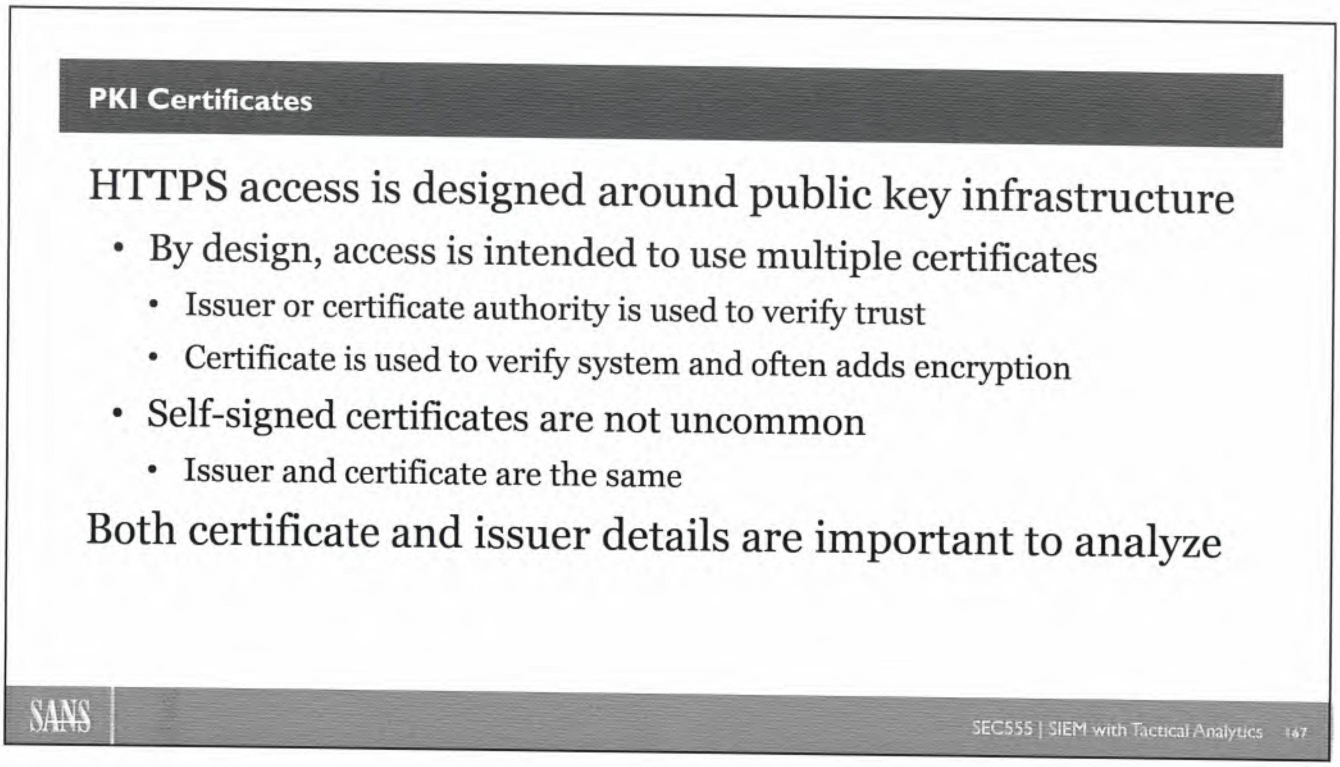
Certificate Fields

- Timestamp
- Source IP
- Source Port
- Destination IP
- Destination Port
- Basic Constraints
- Key Algorithm
- Key Length
- Key Type
- Not Valid After
- Not Valid Before
- Serial
- Signing Algorithm
- Subject
- Version
- Common Name
- Organization
- Organization Unit
- Locality
- State
- Country Code
- Email
- Issuer Info
- And more...

SANS SEC555 | SIEM with Tactical Analytics 166

Certificate Fields

This slide demonstrates some of the common fields included in certificates. All of these—and a little bit more—come with the default Zeek x509 logs.



PKI Certificates

HTTPS access is designed around public key infrastructure

- By design, access is intended to use multiple certificates
  - Issuer or certificate authority is used to verify trust
  - Certificate is used to verify system and often adds encryption
- Self-signed certificates are not uncommon
  - Issuer and certificate are the same

Both certificate and issuer details are important to analyze

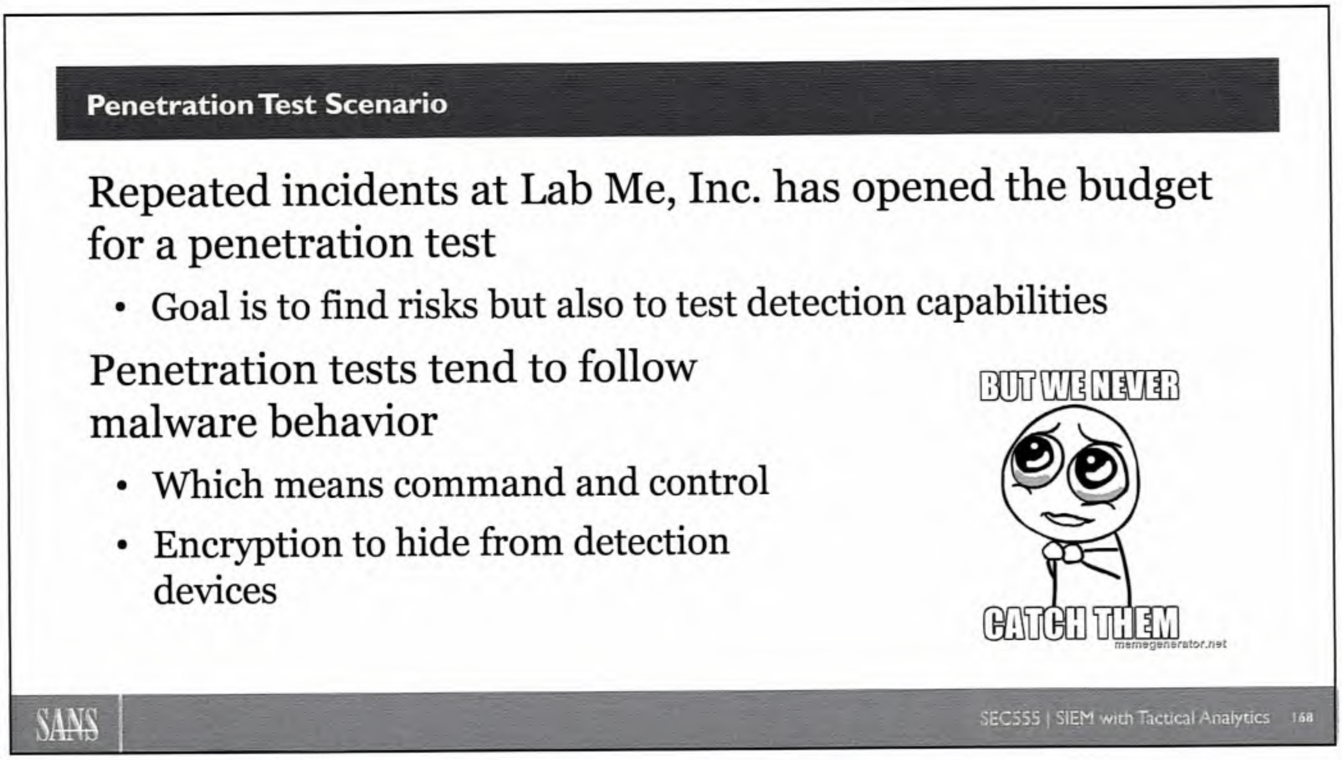
SANS

SEC555 | SIEM with Tactical Analytics 167

PKI Certificates

Public Key Infrastructure, or PKI, was intended to establish hierarchical trust. The intended use is for organizations to trust certain certificate authorities, and therefore, certificates issued by these certificate authorities will in turn be trusted. While this is the basic tenet of PKI, unfortunately self-signed certificates are common. In this case, the certificate is issued and signed by its own private key.

When dealing with certificates, the analysis is done by looking at two things: The certificate issuer and the actual certificate details.



Penetration Test Scenario

Repeated incidents at Lab Me, Inc. has opened the budget for a penetration test

- Goal is to find risks but also to test detection capabilities

Penetration tests tend to follow malware behavior

- Which means command and control
- Encryption to hide from detection devices

BUT WE NEVER
CATCH THEM
memegenerator.net

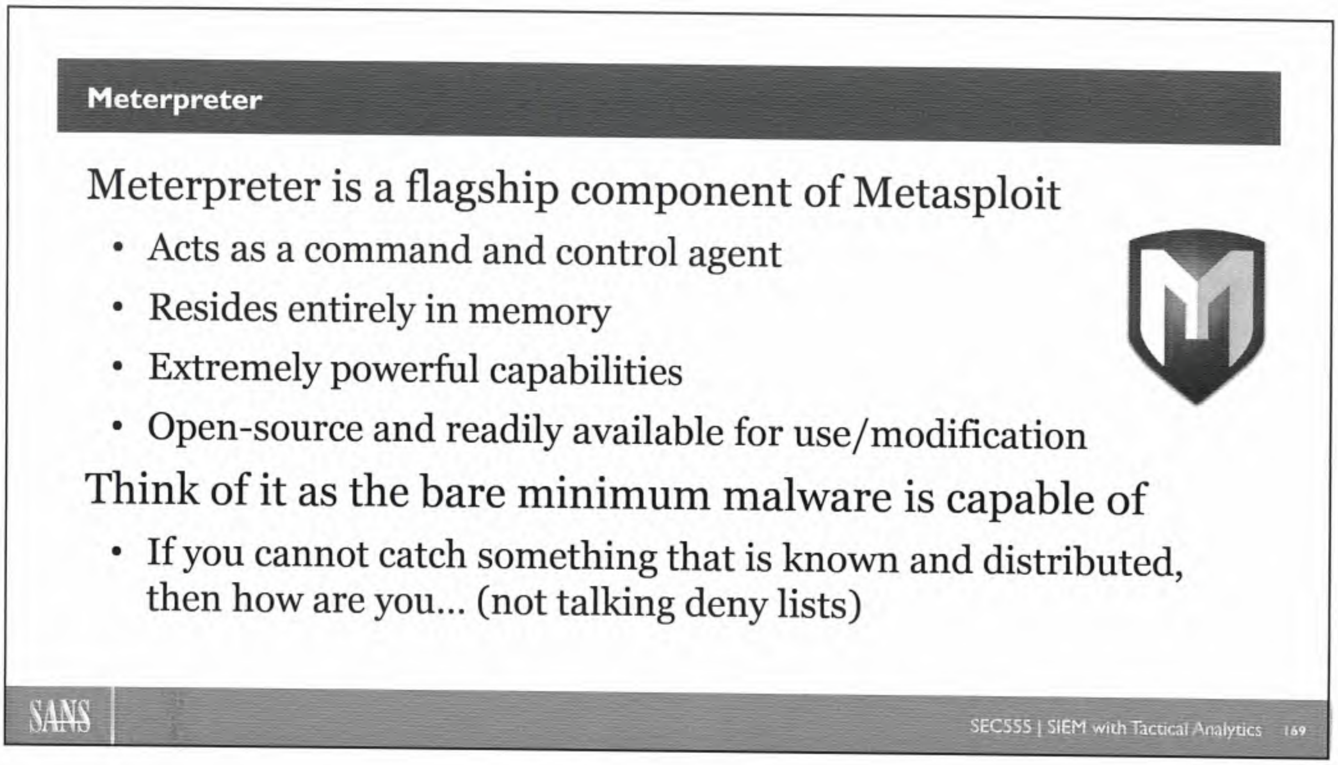
SANS

SEC555 | SIEM with Tactical Analytics 168

Penetration Test Scenario

Due to previous incidents, Lab Me, Inc. has decided to pay for an external penetration test. Penetration tests are great for finding and prioritizing risks to an organization but, if possible, should be used as an assessment of an organization's detection and response capabilities. This is an equally important piece of a penetration test that organizations sometimes overlook.

The end goal, whether or not the penetration testers were detected, is to lower risk. This includes increasing detection capabilities. While you may end up with a report of vulnerabilities and possibly recommendations to fix them, you should absolutely have recommendations to improve detection, such as "This is how you could have caught us." If a penetration testing company is not willing to share how you can catch them, pick a different company to partner with. Having a penetration test used as a lesson learned on detection capabilities is a great value-add.



Meterpreter

Meterpreter is a flagship component of Metasploit

- Acts as a command and control agent
- Resides entirely in memory
- Extremely powerful capabilities
- Open-source and readily available for use/modification

Think of it as the bare minimum malware is capable of

- If you cannot catch something that is known and distributed, then how are you... (not talking deny lists)

Meterpreter

A key tool used by penetration testers is Metasploit. Metasploit is a framework for penetration testing, and one of the most critical components is Meterpreter. Meterpreter is an exploit payload used for command and control and is extremely feature rich.

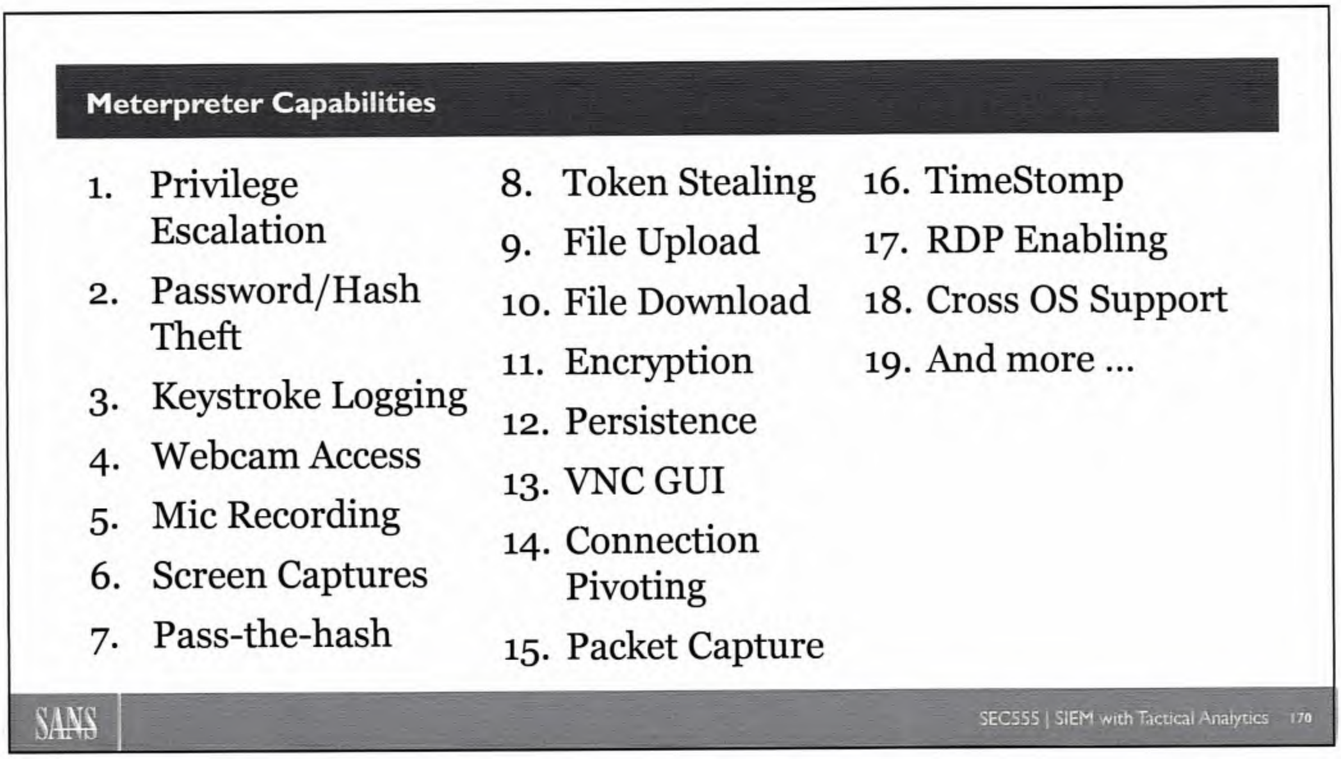
Meterpreter code is open-source, so anyone can freely borrow or use its code to mirror capabilities. This means malware can easily build off of functionality that is readily available. As a result, it should be treated as a minimum standard of what you should be able to catch. This means you should be familiar with the functionality and what occurs in order to catch it.

Vendors treat Meterpreter as a whack-a-mole game. This means they develop signatures that can detect a default Meterpreter agent. However, it is simple to modify it, so signature-based deny lists do not work. In this case, deny lists against Meterpreter using signatures almost hurts the security community more than it helps. Techniques for detection need to be more robust.

References:

https://www.offensive-security.com/metasploit-unleashed/about-meterpreter, https://sec555.com/5i

https://www.metasploit.com, https://sec555.com/5j

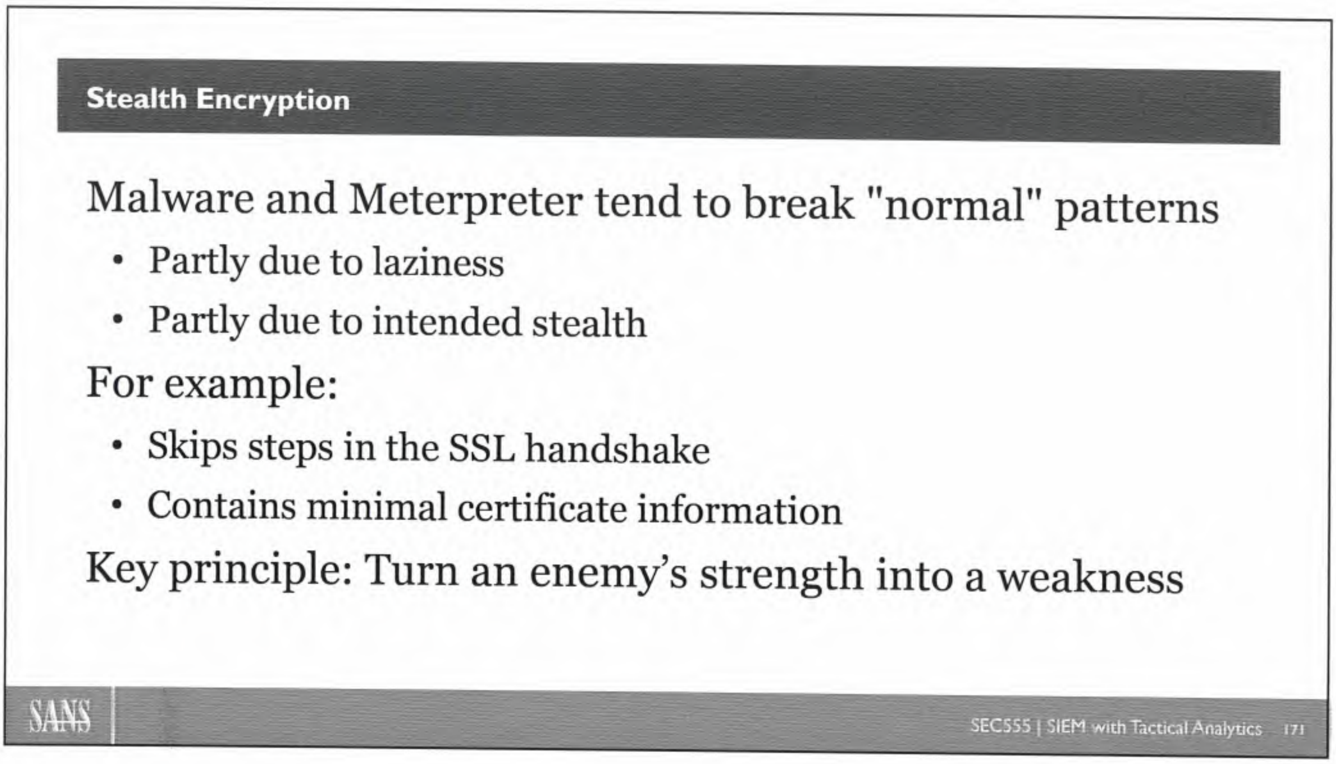


Meterpreter Capabilities

1. Privilege Escalation
2. Password/Hash Theft
3. Keystroke Logging
4. Webcam Access
5. Mic Recording
6. Screen Captures
7. Pass-the-hash
8. Token Stealing
9. File Upload
10. File Download
11. Encryption
12. Persistence
13. VNC GUI
14. Connection Pivoting
15. Packet Capture
16. TimeStomp
17. RDP Enabling
18. Cross OS Support
19. And more ...

Meterpreter Capabilities

Long story short… Meterpreter is awesome and packed full of vitamins and minerals. The flip side is that Meterpreter is terrifying as it is open-source. This means malware is capable of cloning and building upon these feature sets.



Stealth Encryption

Malware and Meterpreter tend to break "normal" patterns

- Partly due to laziness
- Partly due to intended stealth

For example:

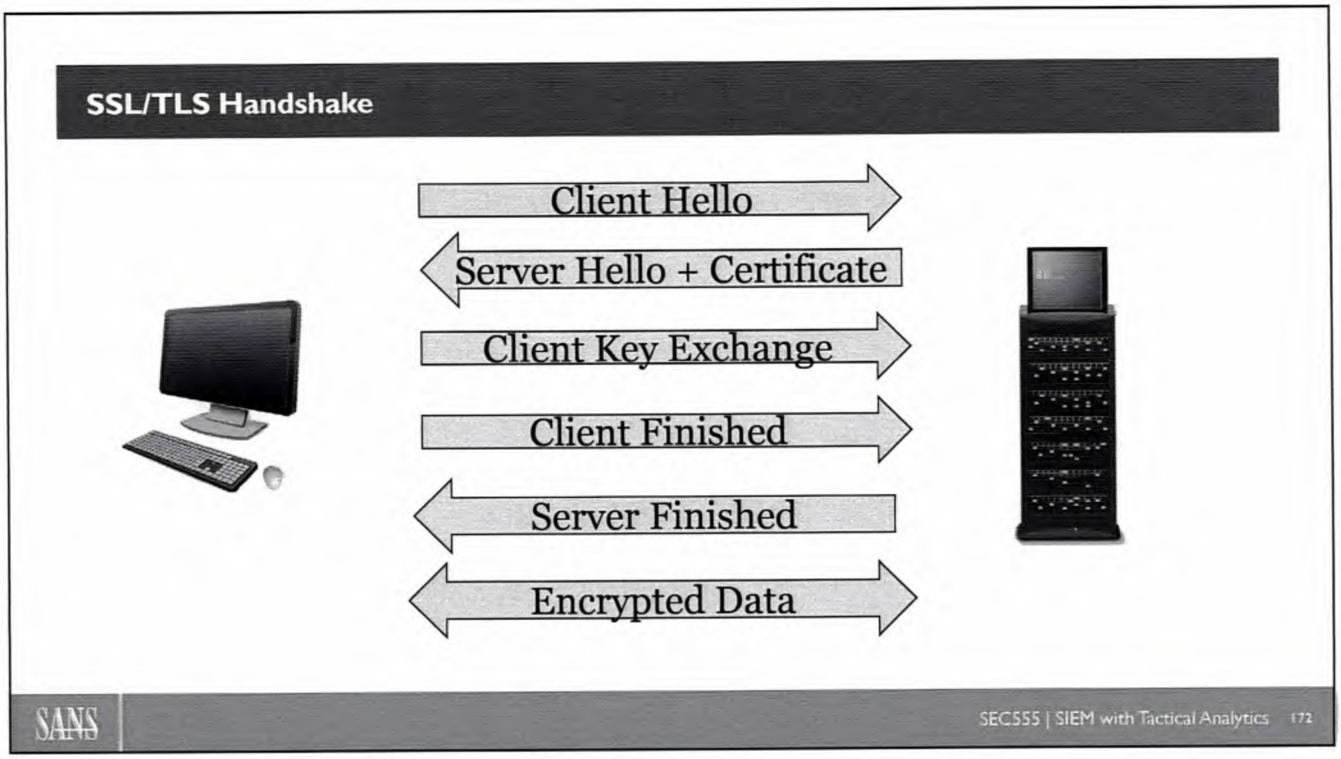
- Skips steps in the SSL handshake
- Contains minimal certificate information

Key principle: Turn an enemy's strength into a weakness

Stealth Encryption

Both Meterpreter and malware wish to avoid detection. To do this, they often will use encryption to blind security devices, but the methods used to encrypt do not always follow normal patterns. Identifying these differences and monitoring for them can provide opportunities for detection.

You may not know what happened, but you will at least know something did.



SSL/TLS Handshake

Client Hello

Server Hello + Certificate

Client Key Exchange

Client Finished

Server Finished

Encrypted Data

SANS

SEC555 | SIEM with Tactical Analytics 172

SSL/TLS Handshake

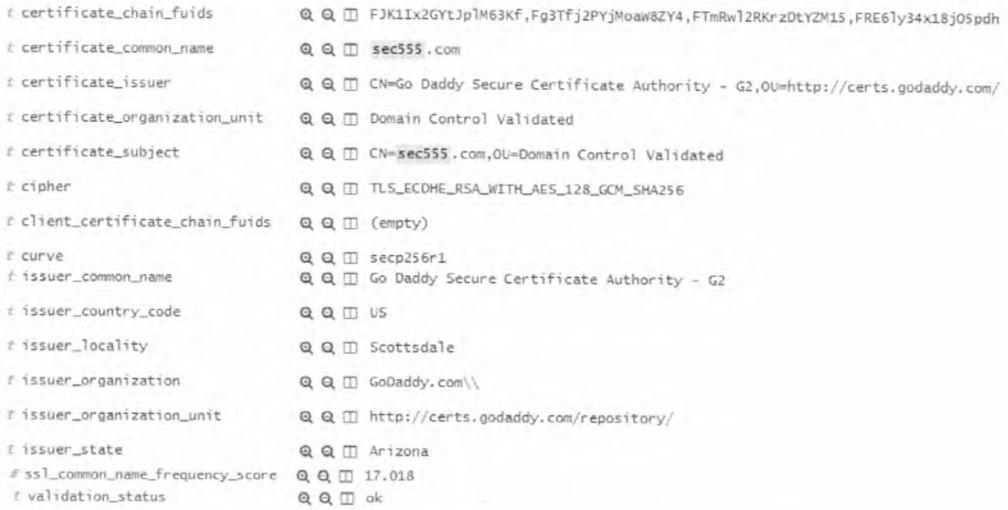
This slide represents an SSL or TLS handshake. This is what happens under normal conditions. Malware, on the other hand, may skip steps. If you are interested in a detailed breakdown of how this handshake works, multiple online articles are available such as that by Digicert[1].

One thing worth noting is that during the client and server hello, the supported version and cipher suites are exchanged. These can be used for asset discovery and baselining.

Reference:

https://www.websecurity.digicert.com/security-topics/how-does-ssl-handshake-work

SSL Log



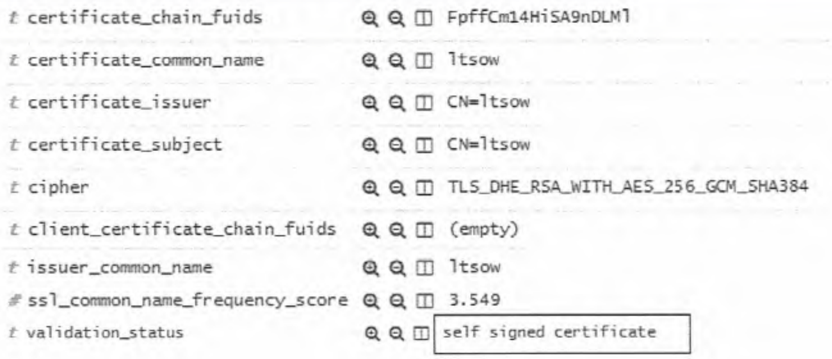
```
t certificate_chain_fuids          FJK1Ix2GYtJplM63Kf,Fg3Tfj2PYjMoaW8ZY4,FTmRwl2RKrzDtYZM15,FRE6ly34x18jO5pdh
t certificate_common_name          sec555.com
t certificate_issuer               CN=Go Daddy Secure Certificate Authority - G2,OU=http://certs.godaddy.com/
t certificate_organization_unit    Domain Control Validated
t certificate_subject              CN=sec555.com,OU=Domain Control Validated
t cipher                           TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
t client_certificate_chain_fuids   (empty)
t curve                            secp256r1
t issuer_common_name               Go Daddy Secure Certificate Authority - G2
t issuer_country_code              US
t issuer_locality                  Scottsdale
t issuer_organization              GoDaddy.com\\
t issuer_organization_unit         http://certs.godaddy.com/repository/
t issuer_state                     Arizona
# ssl_common_name_frequency_score  17.018
t validation_status                ok
```

SSL Log

This slide demonstrates a Zeek SSL log for accessing https://sec555.com. Note the frequency score is high, which in this case means that is probably not random. Also, the issuer certificate-specified an organization unit, organization, locality, state, and country code field and the certificate itself included an organization unit field along with the common name field.

Also, the certificate_chain_fuids field has four entries. This is because there are four certificates in the chain: sec555's certificate, two subordinate certificate authorities, and one root certificate authority.

Meterpreter SSL Log

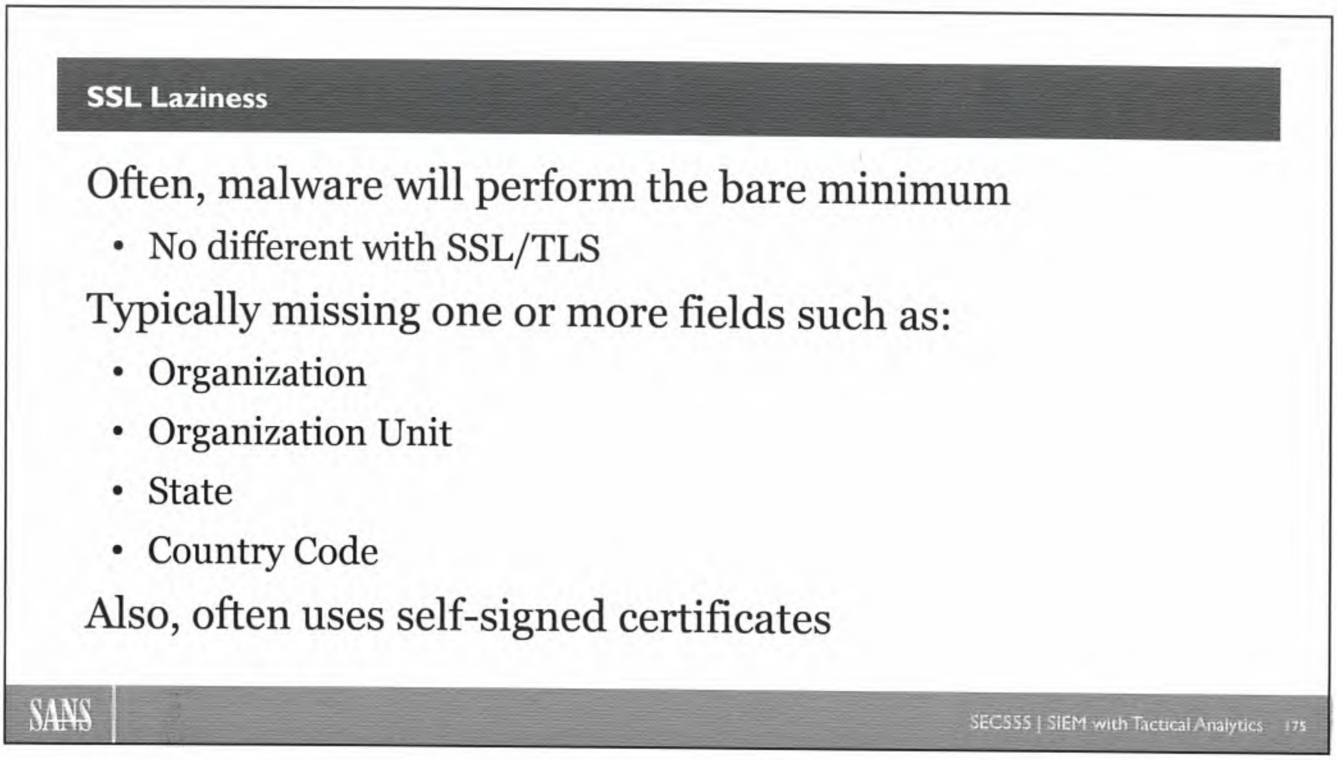


```
t certificate_chain_fuids              FpffCm14HiSA9nDLMl
t certificate_common_name              ltsow
t certificate_issuer                   CN=ltsow
t certificate_subject                  CN=ltsow
t cipher                               TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
t client_certificate_chain_fuids       (empty)
t issuer_common_name                   ltsow
# ssl_common_name_frequency_score      3.549
t validation_status                    self signed certificate
```

Meterpreter SSL Log

This slide demonstrates what an SSL log would look like for a reverse HTTPS Meterpreter connection. The use of a self-signed certificate with a random common name is generated. Also, many of the other fields normally in existence are missing.

In this case, certificate_chain_fuids only has one entry because a self-signed certificate is used.



SSL Laziness

Often, malware will perform the bare minimum

- No different with SSL/TLS

Typically missing one or more fields such as:

- Organization
- Organization Unit
- State
- Country Code

Also, often uses self-signed certificates

SANS

SEC555 | SIEM with Tactical Analytics 175

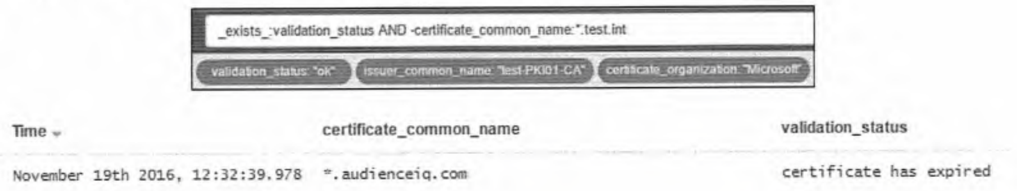
SSL Laziness

Similar to the previous Meterpreter SSL Log, malware often uses the bare minimum. This means key fields often used by SSL are left empty. Yes, malware can use properly formed certificates that use trusted root certificate authorities, but this is often expensive and time-consuming. Therefore, many do not.

Expired or Self-Signed

Accessing systems with expired or self-signed certificates is generally a bad idea

- Especially true for external access
- Monitor all expired or self-signed certificates
- Filtering required for internal monitoring



```
_exists_:validation_status AND -certificate_common_name:*.test.int
```

validation_status: "ok" | issuer_common_name: "test-PKI01-CA" | certificate_organization: "Microsoft"

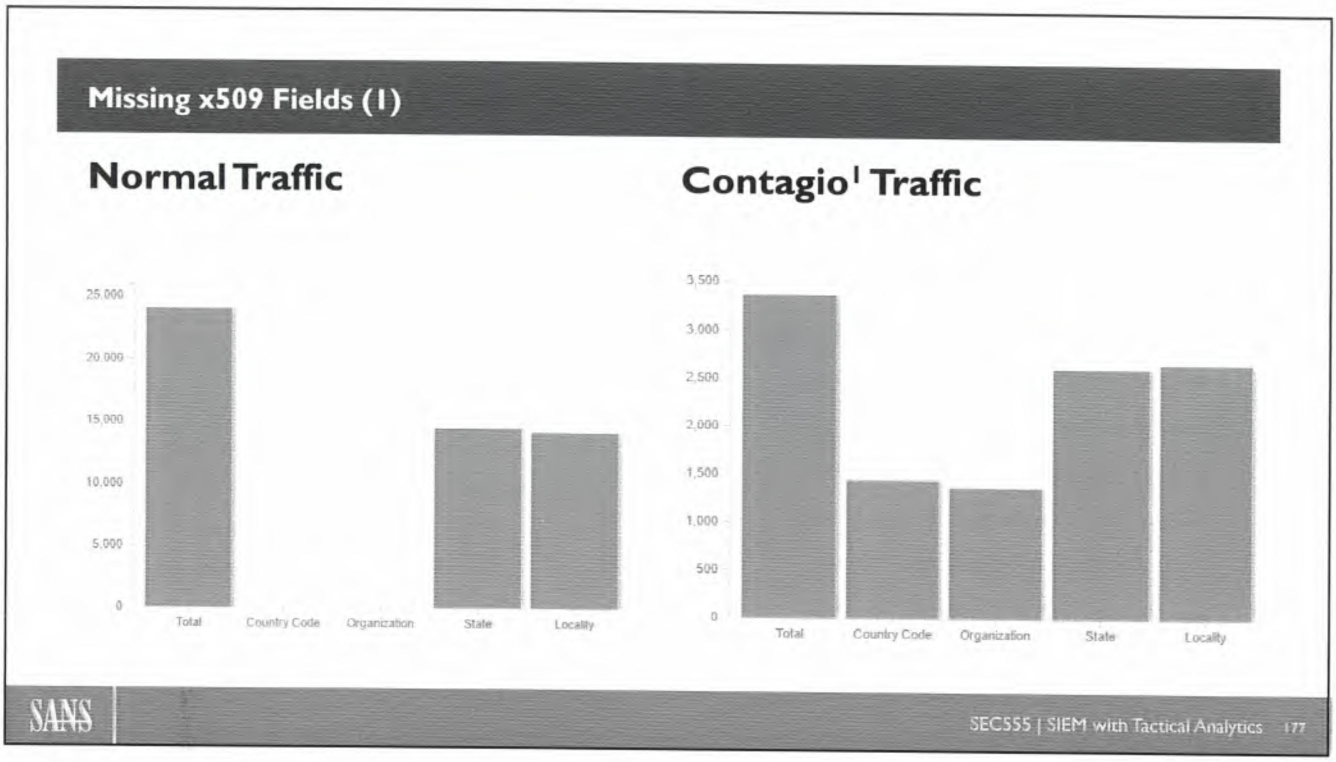
Time	certificate_common_name	validation_status
November 19th 2016, 12:32:39.978	*.audienceiq.com	certificate has expired

Expired or Self-Signed

Many next-generation firewalls support SSL decryption. When enabled, it allows for easy filtering of expired and self-signed certificates. Yet many organizations are not authorized to enable SSL decryption. Plus, a firewall is typically not positioned to see internal-to-internal certificates.

Collecting and analyzing certificates using something like Zeek allows visibility into internal-to-internal, external-to-internal, and internal-to-external certificate use. This opens up more opportunities to do things such as looking for self-signed or expired certificates being used.

If you are trying to filter out noise caused by self-signed certificates being used internally, try filtering on vendor-specific attributes. For example, most self-signed certificates used by vendors have the vendor's name or product specified. It is significantly better to filter them out than to state this technique will not work because we have internal self-signed certificates.



Missing x509 Fields (1)

Normal Traffic

Contagio[1] Traffic

SEC555 | SIEM with Tactical Analytics 177

Missing x509 Fields (1)

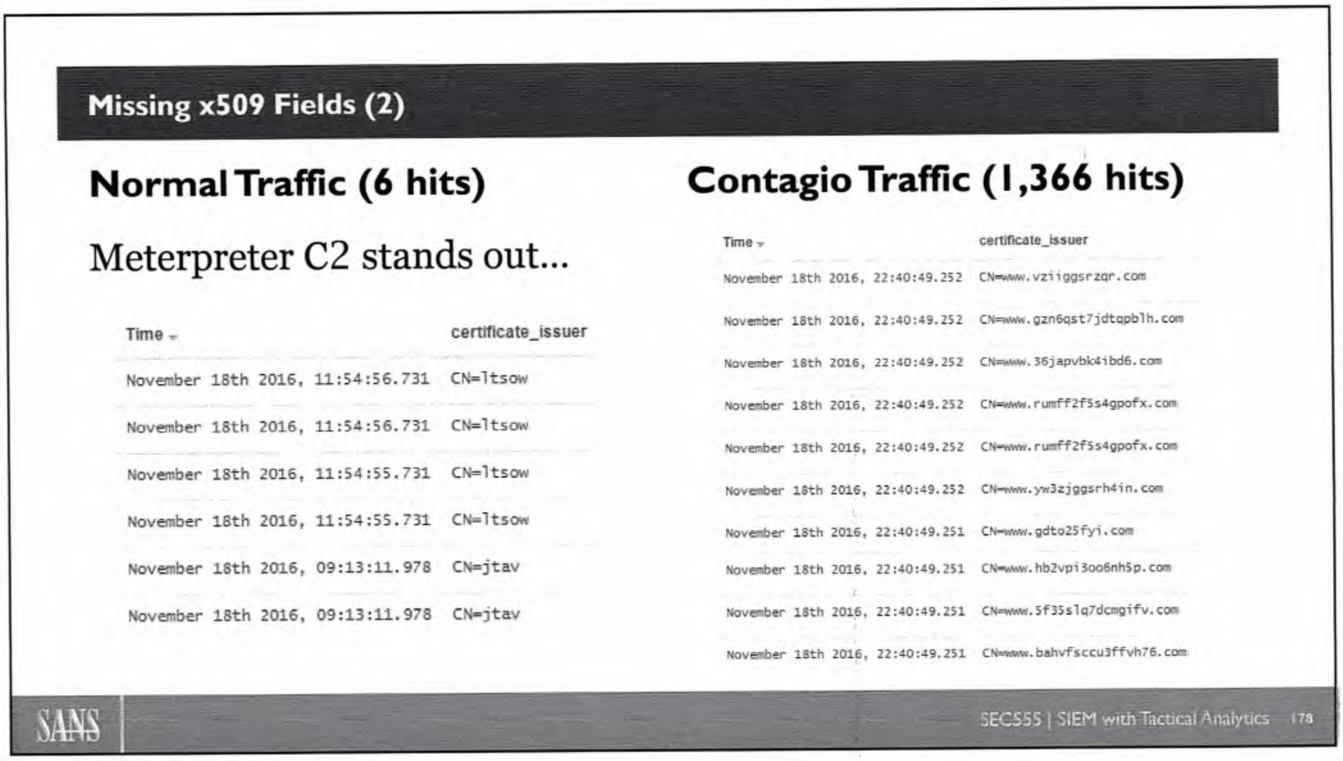
This slide represents analyzing x509 logs for missing fields. Both bar charts have a total record count, followed by four additional counts for country code, organization, state, and locality. These counts reflect whether the field is missing or not.

The image on the left is taken from a week's worth of normal traffic. Notice that country code and organization look empty; meaning that they are present in the x509 logs. State and locality, however, are missing about 50% of the time. Because organizations often have internal PKIs and self-signed certificates, this technique works best when filtering out things such as certificates using internal domain names such as -certificate_subject:*.test.int AND -san_dns:*.test.int. This assumes test.int was the internal domain.

The image on the right is taken from analyzing approximately 2 GB of malware traffic from Contagio. Notice that country code and organization are missing in almost 33% of the x509 logs and state and locality are missing in around 75% of the logs.

Reference:

https://contagiodump.blogspot.com, https://sec555.com/5e



Missing x509 Fields (2)

Normal Traffic (6 hits)

Meterpreter C2 stands out...

Time	certificate_issuer
November 18th 2016, 11:54:56.731	CN=ltsow
November 18th 2016, 11:54:56.731	CN=ltsow
November 18th 2016, 11:54:55.731	CN=ltsow
November 18th 2016, 11:54:55.731	CN=ltsow
November 18th 2016, 09:13:11.978	CN=jtav
November 18th 2016, 09:13:11.978	CN=jtav

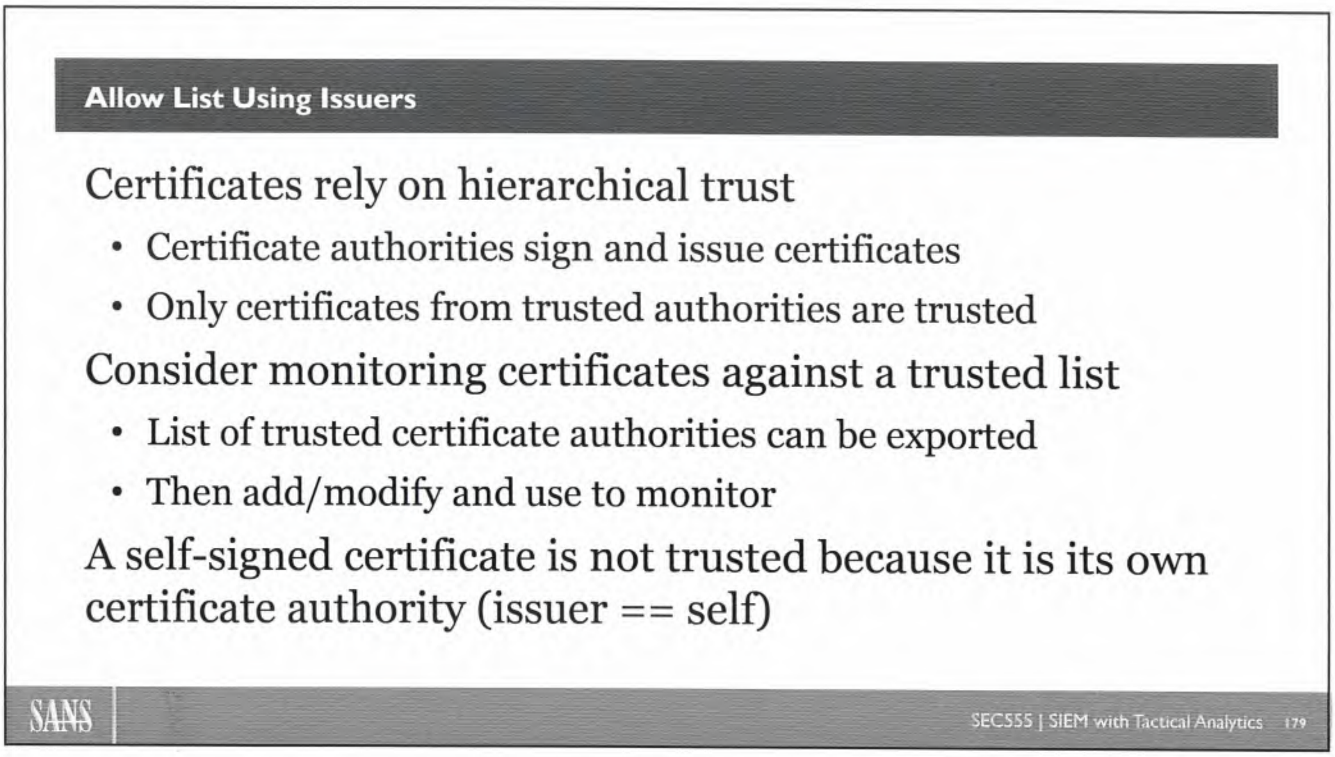
Contagio Traffic (1,366 hits)

Time	certificate_issuer
November 18th 2016, 22:40:49.252	CN=www.vziiggsrzqr.com
November 18th 2016, 22:40:49.252	CN=www.gzn6qst7jdtqpblh.com
November 18th 2016, 22:40:49.252	CN=www.36japvbk4ibd6.com
November 18th 2016, 22:40:49.252	CN=www.rumff2f5s4gpofx.com
November 18th 2016, 22:40:49.252	CN=www.rumff2f5s4gpofx.com
November 18th 2016, 22:40:49.252	CN=www.yw3zjggsrh4in.com
November 18th 2016, 22:40:49.251	CN=www.gdto25fyi.com
November 18th 2016, 22:40:49.251	CN=www.hb2vpi3oo6nh5p.com
November 18th 2016, 22:40:49.251	CN=www.5f35slq7dcmgifv.com
November 18th 2016, 22:40:49.251	CN=www.bahvfsccu3ffvh76.com

Missing x509 Fields (2)

The previous charts, on the last slide, make it look like the country code field and organization field are always present for normal traffic. This is because they normally are. In this case, there were actually six logs that did not have a country code or organization specified. All six logs belonged to Meterpreter command and control traffic from internally compromised systems. Looking for missing x509 fields simply works.

Looking for missing fields using known bad traffic also helps to vet the technique. As you can see in the Contagio traffic, looking for missing fields in x509 uncovers a lot.



Allow List Using Issuers

Certificates rely on hierarchical trust

- Certificate authorities sign and issue certificates
- Only certificates from trusted authorities are trusted

Consider monitoring certificates against a trusted list

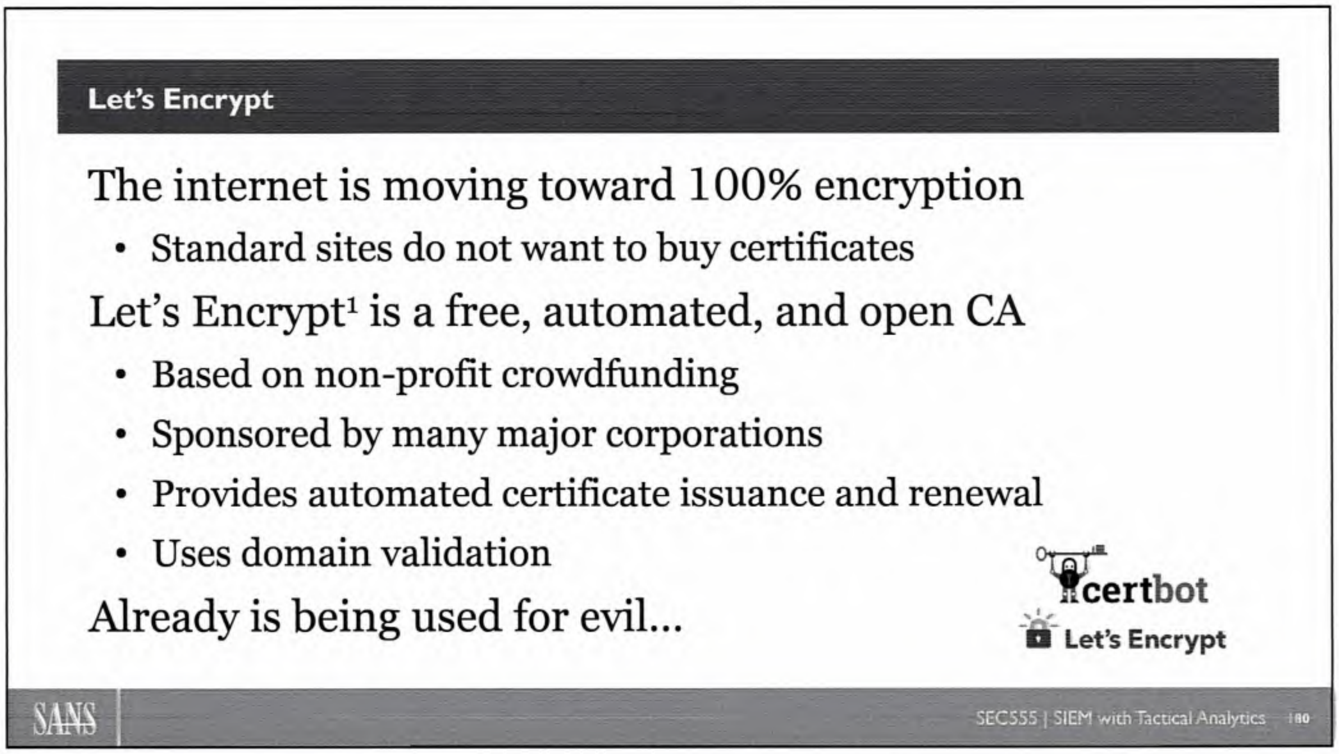
- List of trusted certificate authorities can be exported
- Then add/modify and use to monitor

A self-signed certificate is not trusted because it is its own certificate authority (issuer == self)

Allow List Using Issuers

Ideally, certificates should be generated by trusted root certificate authorities. Therefore, it is possible to export a list of trusted root certificate authorities and then have your SIEM look for SSL certificates that are not issued with your trusted list.

To make this usable, you may wish to only apply this to SSL certificates going into or out of the environment.



Let's Encrypt

The internet is moving toward 100% encryption

- Standard sites do not want to buy certificates

Let's Encrypt[1] is a free, automated, and open CA

- Based on non-profit crowdfunding
- Sponsored by many major corporations
- Provides automated certificate issuance and renewal
- Uses domain validation

Already is being used for evil...

certbot

Let's Encrypt

SANS

SEC555 | SIEM with Tactical Analytics 180

Let's Encrypt

One of the major holdbacks in shifting to 100% website encryption is the cost of a certificate. Previously, these ran about $30 a site. Let's Encrypt is an organization that was developed to eliminate this cost by providing free and automated certificates. It does this by allowing a client called certbot to issue and renew certificates automatically. These certificates are valid for exactly 90 days.

Arguably, this increases security as it introduces encryption and helps mitigate man-in-the-middle attacks. However, it also decreases security as it allows malware and phishing campaigns to use trusted certificates. Because of the automation, there is a high likelihood that malware and adversaries will start using free certificates such as the ones through this company. In fact, there already have been multiple occurrences of this that you can read about online.[2]

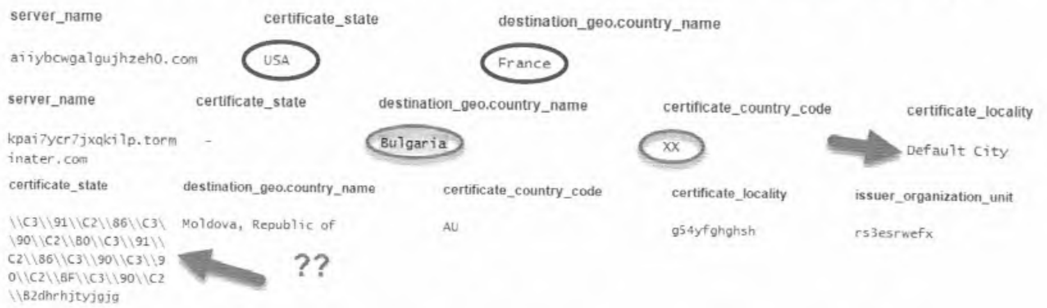
References:

https://letsencrypt.org, https://sec555.com/5m

https://www.trendmicro.com/tr_tr/research/16/a/lets-encrypt-now-being-abused-by-malvertisers.html, https://sec555.com/116

Certificate Analysis

Certificate analysis points out other items of interest



server_name	certificate_state	destination_geo.country_name
aiiybcwgalgujhzeh0.com	USA	France

server_name	certificate_state	destination_geo.country_name	certificate_country_code	certificate_locality
kpai7ycr7jxqkilp.torminater.com	-	Bulgaria	XX	Default City

certificate_state	destination_geo.country_name	certificate_country_code	certificate_locality	issuer_organization_unit
\\C3\\91\\C2\\86\\C3\\90\\C2\\B0\\C3\\91\\C2\\86\\C3\\90\\C3\\90\\C2\\BF\\C3\\90\\C2\\B2dhrhjtyjgjg	Moldova, Republic of	AU	g54yfghghsh	rs3esrwefx

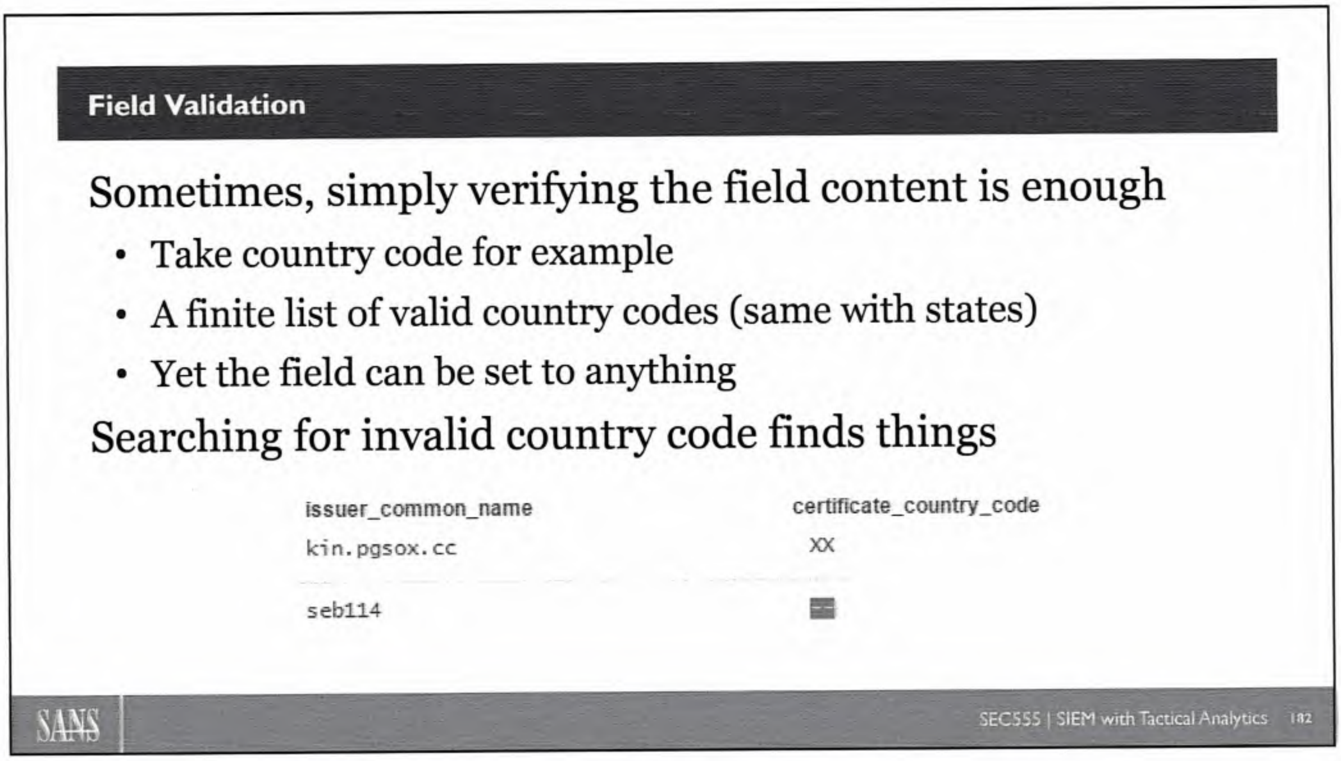
Not to mention the repetitive use of random values

Certificate Analysis

The goal of an analyst should include understanding their organization and identifying discrepancies or abnormalities. Things such as a certificate showing a certificate state of USA, yet having a destination Geolocation of France, are odd. A country code of XX with a city of Default city is just plain weird. And then, odd characters within any field, let alone the state field, is highly suspicious.

All of these are would-be examples of things worth looking into. Ideally, a SIEM will facilitate this level of analysis and provide the tool necessary to do so. However, it is ultimately up to the person on the other end of the tool to use it.

Techniques must be tried out and developed. For example, at first glance, looking for certificates with a country code of US where the destination Geo is not the US does not work. CDNs and sites associated with Facebook, Microsoft, and security suites often use global servers. Does that mean the technique will not work? It may not, but it might. Try filtering out things based on tags, ASN numbers, and other certificate information and see if the technique is applicable.



Field Validation

Sometimes, simply verifying the field content is enough

- Take country code for example
- A finite list of valid country codes (same with states)
- Yet the field can be set to anything

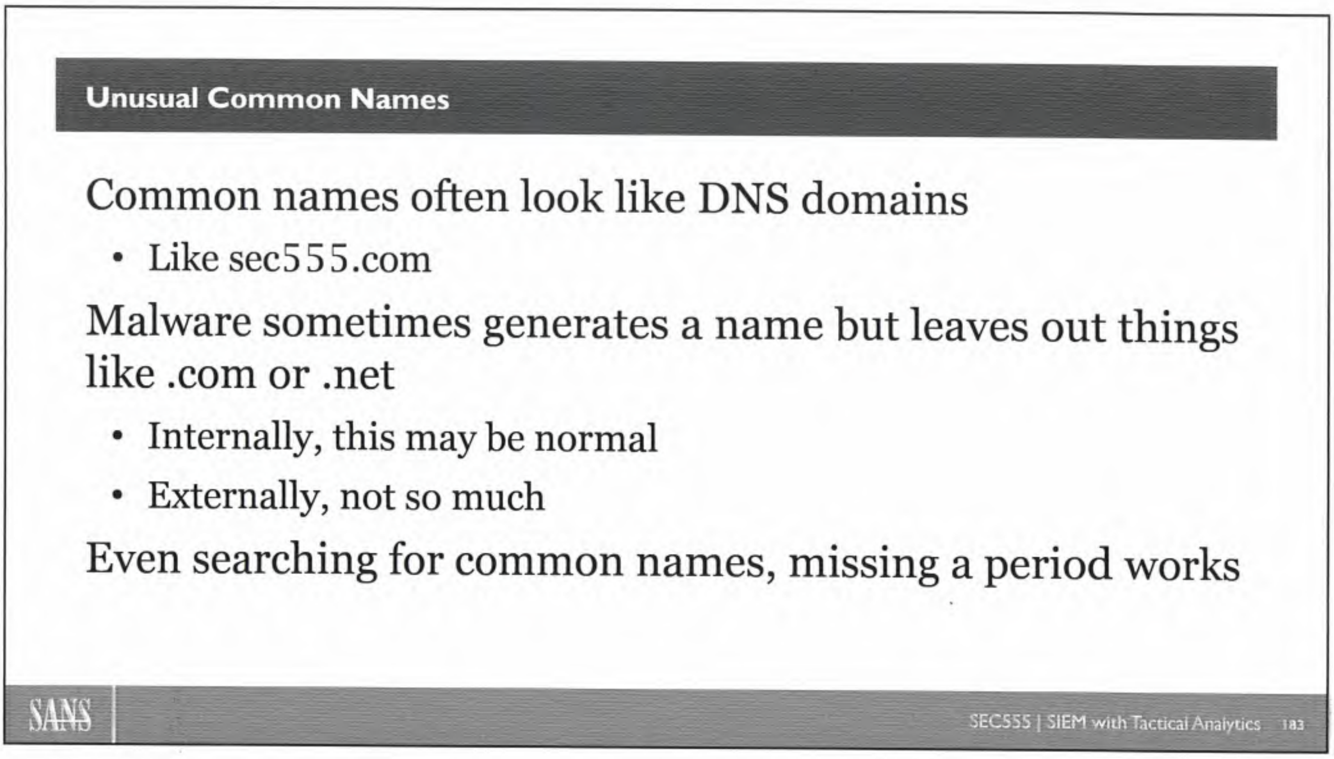
Searching for invalid country code finds things

issuer_common_name	certificate_country_code
kin.pgsox.cc	XX
seb114	▬

Field Validation

Sometimes, simple tricks work. For example, certificates often have a country code specified. Since there are only so many countries in the world, a simple search can be composed of looking for any certificates that do not match a valid country code. To test valid country codes, organizations need to understand how country codes are used in TLS certificates. For example, XX is a valid country code that means a TLS certification is intended to be used in multiple countries.

Testing this in a normal environment returns zero logs. Testing this against the Contagio malware analysis shows multiple logs such as those specified in this slide. This technique also works well for states.



Unusual Common Names

Common names often look like DNS domains

- Like sec555.com

Malware sometimes generates a name but leaves out things like .com or .net

- Internally, this may be normal
- Externally, not so much

Even searching for common names, missing a period works

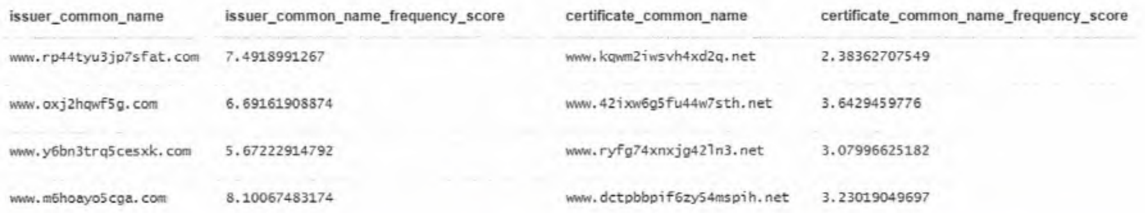
Unusual Common Names

Generally, HTTPS is used to access websites, and as a result, the certificate is expected to follow a domain name. Therefore, finding a certificate with a common name missing a period is abnormal. Yet malware, or even tools such as Meterpreter, are guilty of this.

Frequency Scores

Multiple fields are candidates for frequency analysis

- Missing fields find lazy malware
- Filled fields fall before the power of frequency analysis
- Combining techniques acts as a force multiplier

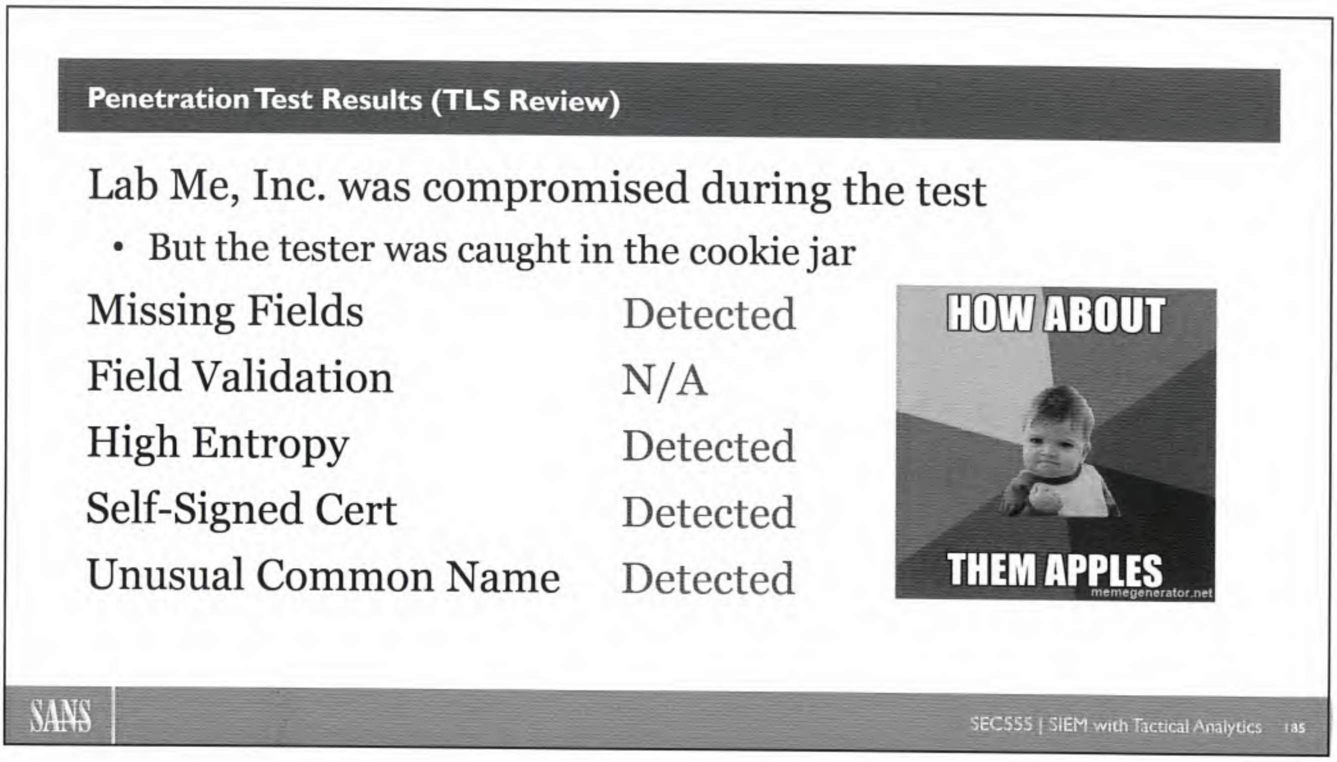


issuer_common_name	issuer_common_name_frequency_score	certificate_common_name	certificate_common_name_frequency_score
www.rp44tyu3jp7sfat.com	7.4918991267	www.kqwm2iwsvh4xd2q.net	2.38362707549
www.oxj2hqwf5g.com	6.69161908874	www.42ixw6g5fu44w7sth.net	3.6429459776
www.y6bn3trq5cesxk.com	5.67222914792	www.ryfg74xnxjg42ln3.net	3.07996625182
www.m6hoayo5cga.com	8.10067483174	www.dctpbbpif6zy54mspih.net	3.23019049697

Frequency Scores

Once again, frequency scoring of key fields is an amazing tool for finding randomly generated data where it is not expected. Certain techniques are complementary to each other. For example, missing fields will find malware that tries to fill out the bare minimum required. Yet, it does nothing for malware where fields are filled out. Frequency analysis will find randomly generated fields but does nothing for empty fields. Combining one technique with the other helps address the weaknesses in the other.

The end goal is to make evasion so difficult that adversaries cannot hide. Yes, any individual detection technique can be avoided. But ultimately, with the application of many techniques and processes, it is extremely difficult not to trigger at least one.



Penetration Test Results (TLS Review)

Lab Me, Inc. was compromised during the test

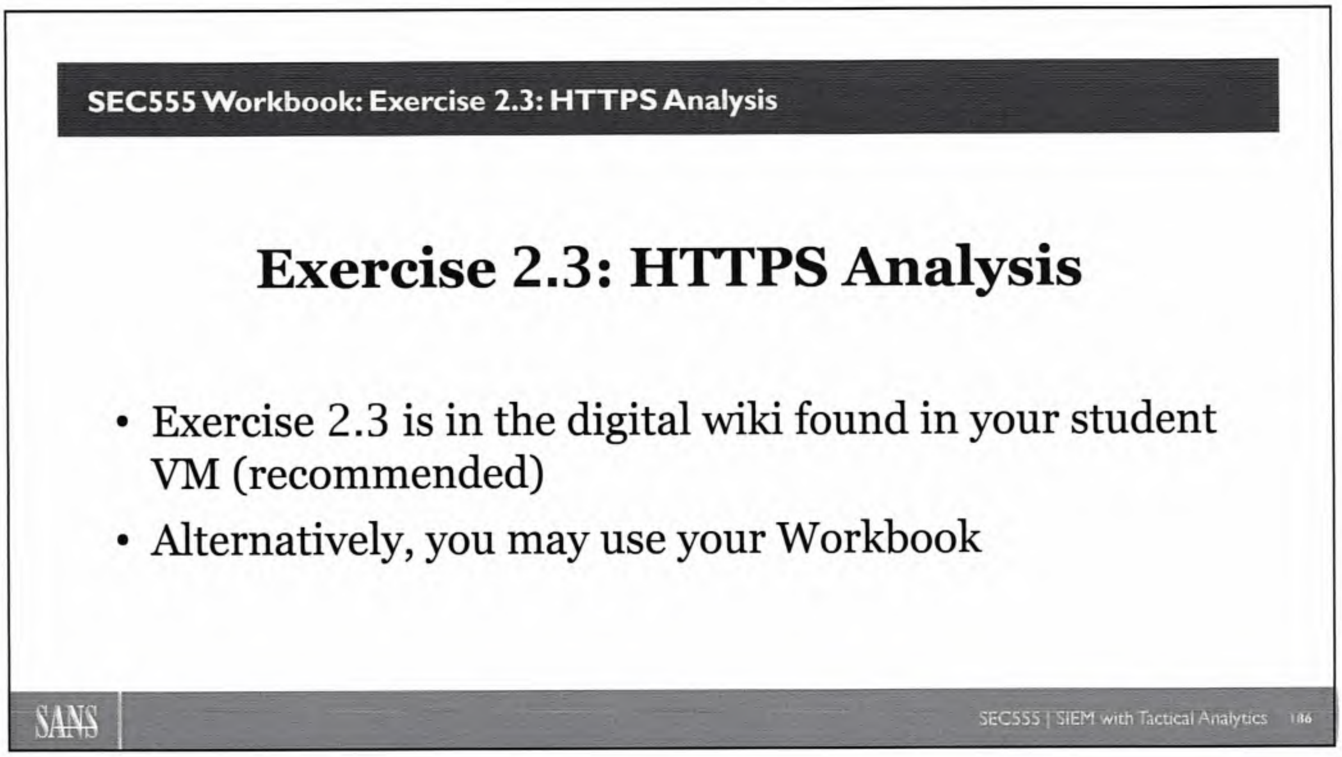
- But the tester was caught in the cookie jar

Missing Fields	Detected
Field Validation	N/A
High Entropy	Detected
Self-Signed Cert	Detected
Unusual Common Name	Detected

HOW ABOUT
THEM APPLES
memegenerator.net

Penetration Test Results (TLS Review)

As is the case in most penetration tests, Lab Me, Inc. was able to be successfully compromised. However, due to your awesome mad skills, you were able to detect them. Had this been a real attack, you would have been able to identify and respond to the threat quickly.

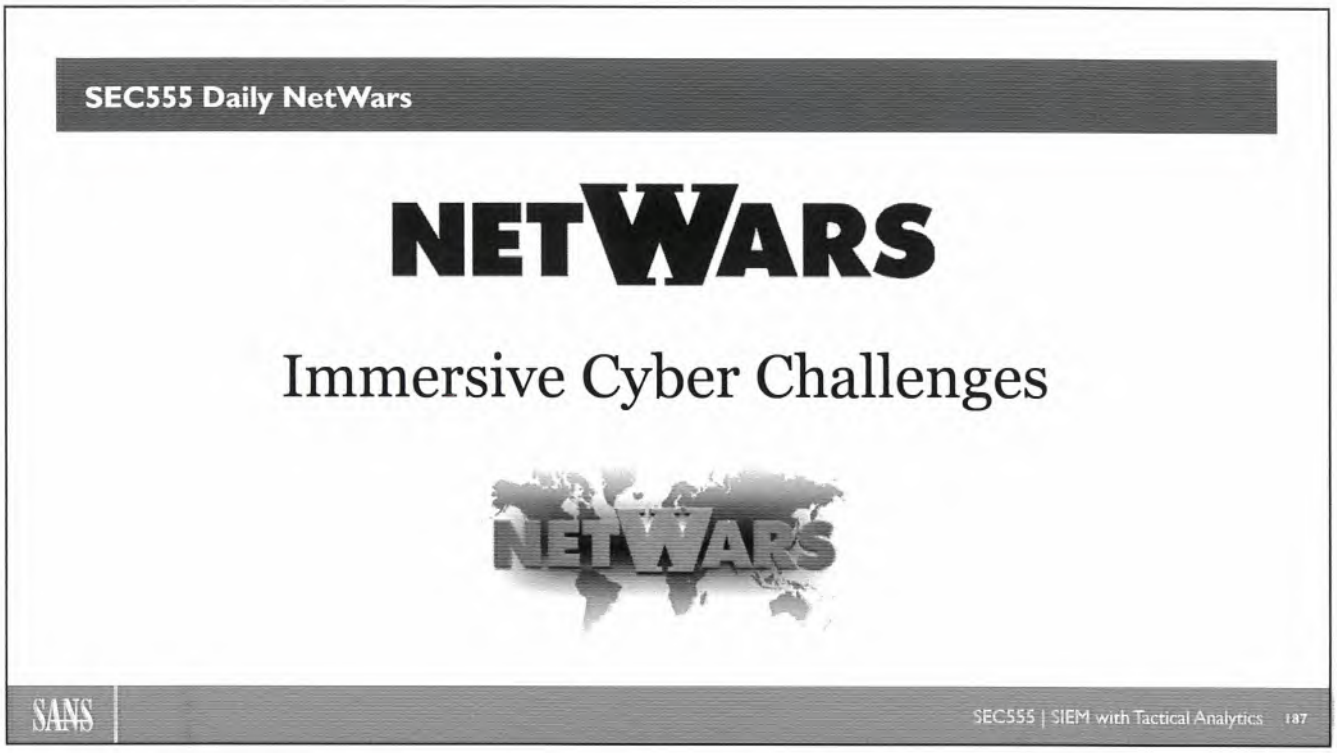


SEC555 Workbook: Exercise 2.3: HTTPS Analysis

Exercise 2.3: HTTPS Analysis

- Exercise 2.3 is in the digital wiki found in your student VM (recommended)
- Alternatively, you may use your Workbook

This page intentionally left blank.



SEC555 Daily NetWars

NETWARS

Immersive Cyber Challenges

Each section of SEC555: SIEM with Tactical Analytics ends with an immersive cyber challenge using the NetWars engine. A NetWars scoreboard will be utilized to provide questions that allow for significant hands-on experience in addition to prior labs.

If you are attending a live course, instructions will be provided for accessing the NetWars Bootcamp by your instructor.

If you are taking this class via OnDemand, instructions for connecting to your remote lab environment can be found by clicking on My Labs in your SANS portal and following the on-screen instructions.

Free Cybersecurity Resources

sans.org/free

SANS instructors and analysts produce thousands of free resources and tools for the cybersecurity community, including more than **150 free tools and hundreds of white papers authored annually**. SANS remains committed to providing free education and capabilities to the cyber communities we serve, train, and certify.

Free Cybersecurity Community Resources

- **Internet Storm Center** – Free Analysis and Warning Service
- **White Papers** – Community InfoSec Research
- **Blog** – Cybersecurity Blog
- **Newsletters** – Newsbites; @Risk; OUCH!
- **Webcasts** – Live and Archived
- **Posters** – Job-Focused Resources
- **SANS Holiday Hack Challenge**
- **Critical Security Controls** – Recommended Actions for Cyber Defense
- **Free Tools** – SANS Instructors have built more than 150 open-source tools that support your work and help you implement better security

Join the SANS alumni community online

> "As usual, SANS courses pay for themselves by Day 2. By Day 3, you are itching to get back to the office to use what you've learned."
>
> Ken Evans, Hewlett Packard Enterprise - Digital Investigation Services

Free Training and Events

- Test Drive 45+ SANS Courses
- Free SANS Summits & Forums
- Capture-the-Flag Cyber Challenges
- Cyber Aces

SANS | GIAC CERTIFICATIONS

www.sans.org