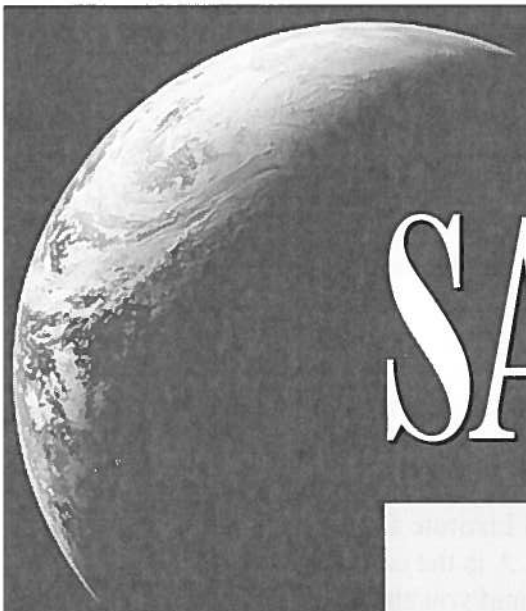


SANS

www.sans.org

SECURITY 511

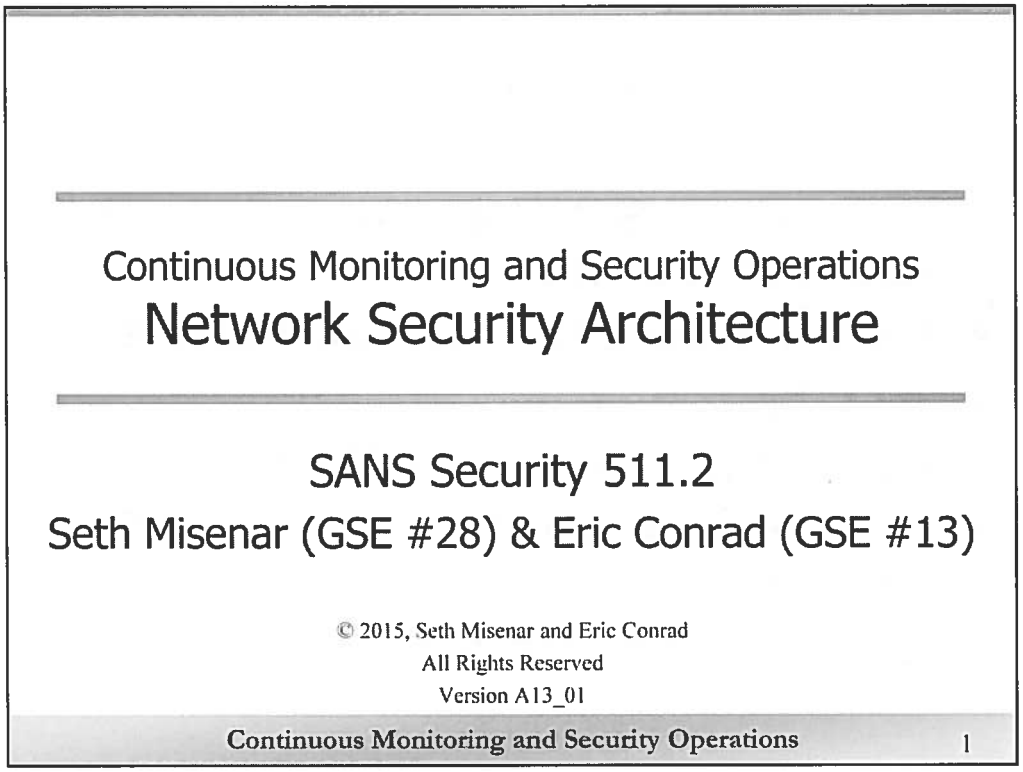
CONTINUOUS MONITORING AND SECURITY OPERATIONS

511.2

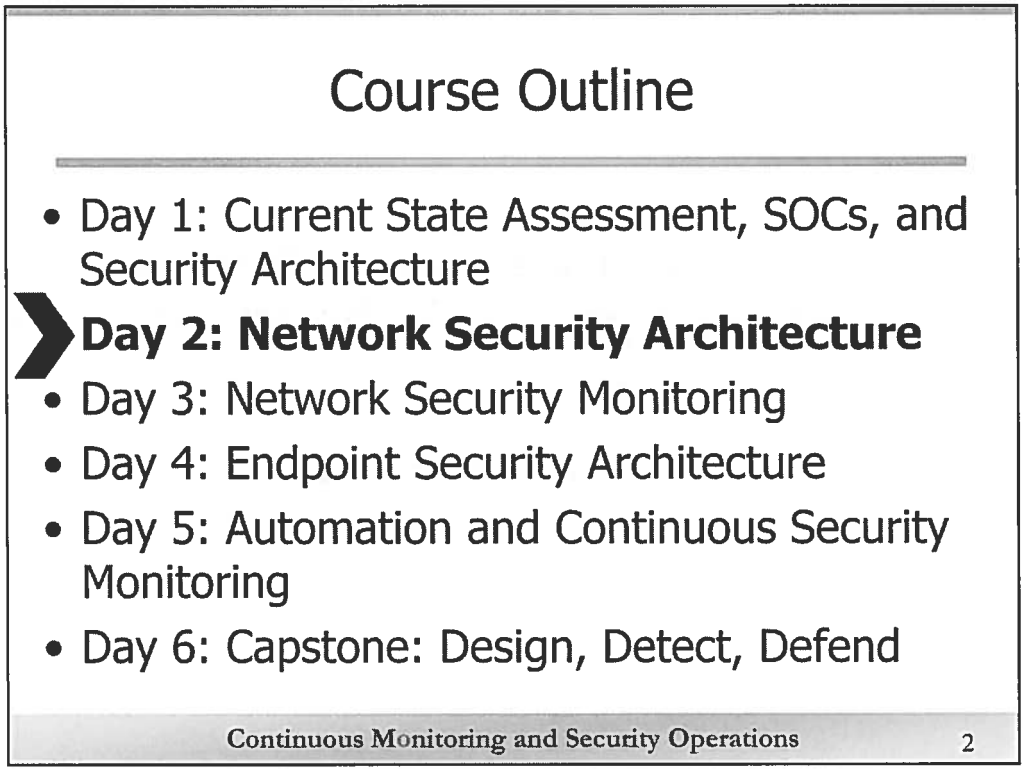
Network Security Architecture

The right security training for your staff, at the right time, in the right location.

Sec511_2_A13_01

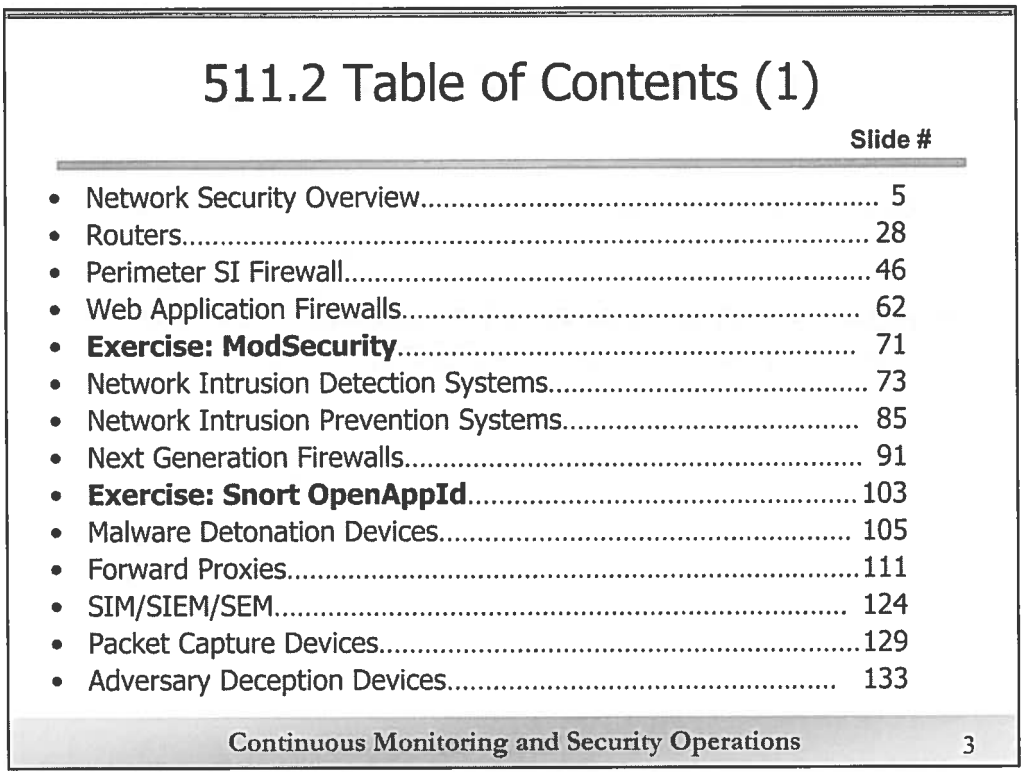


Welcome to Day 2, Network Security Architecture.



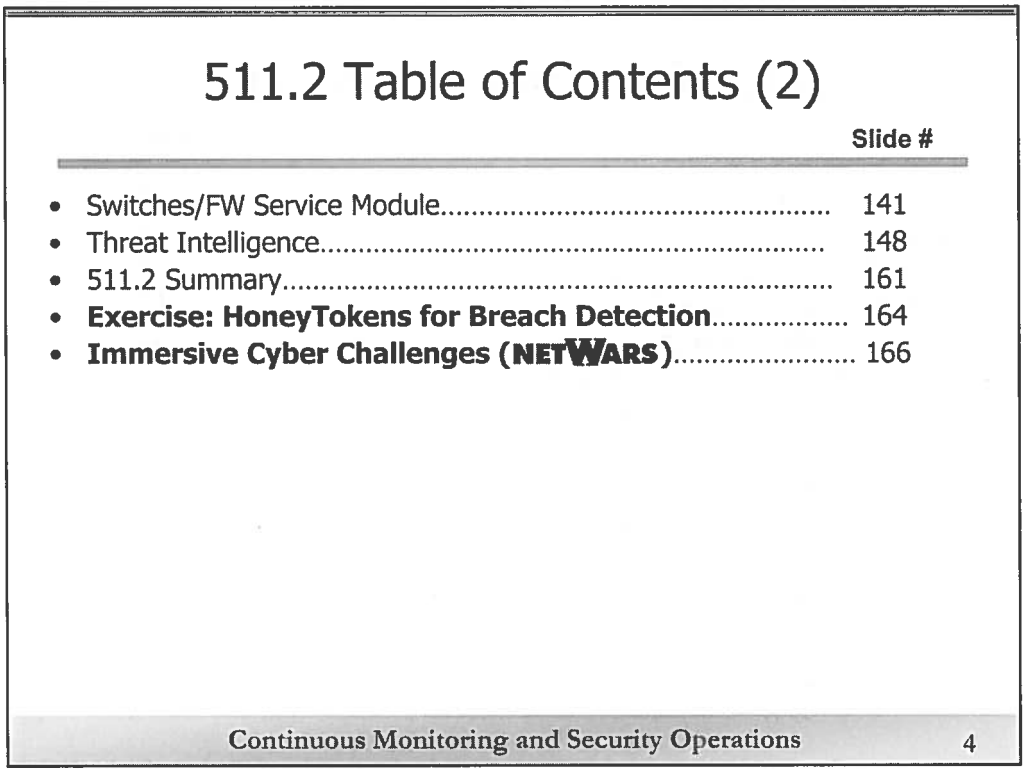
Course Outline

Now let's explore Network Security Architecture.



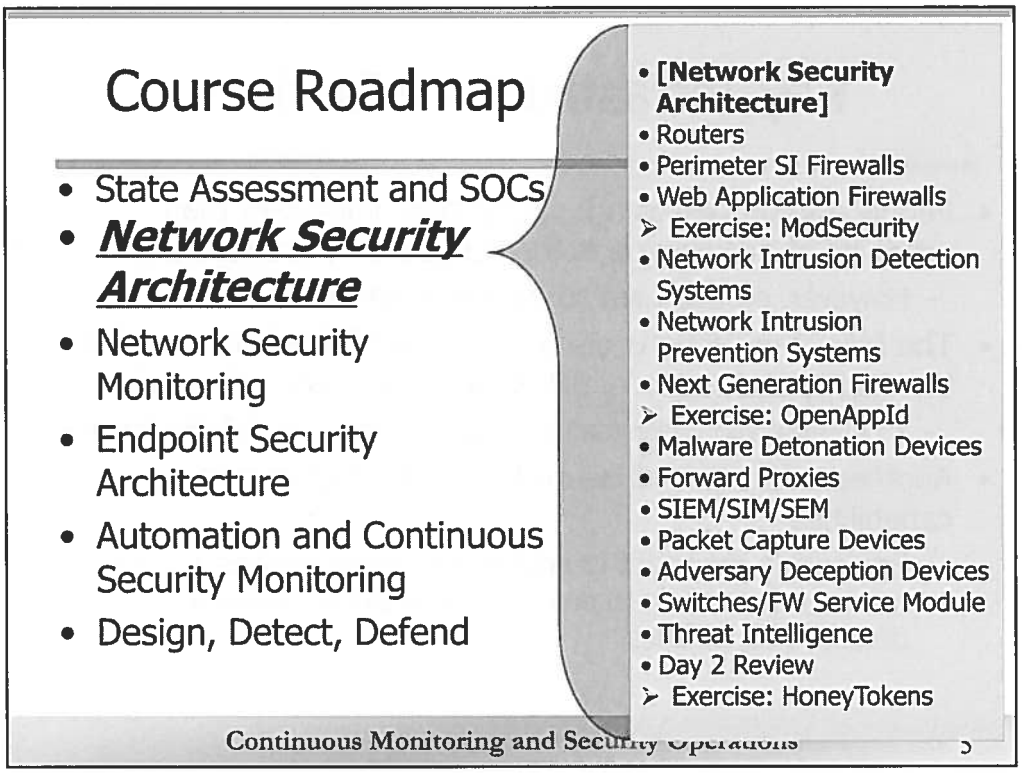
511.2 Table of Contents (1)

This table of contents outlines our plan for 511.2.



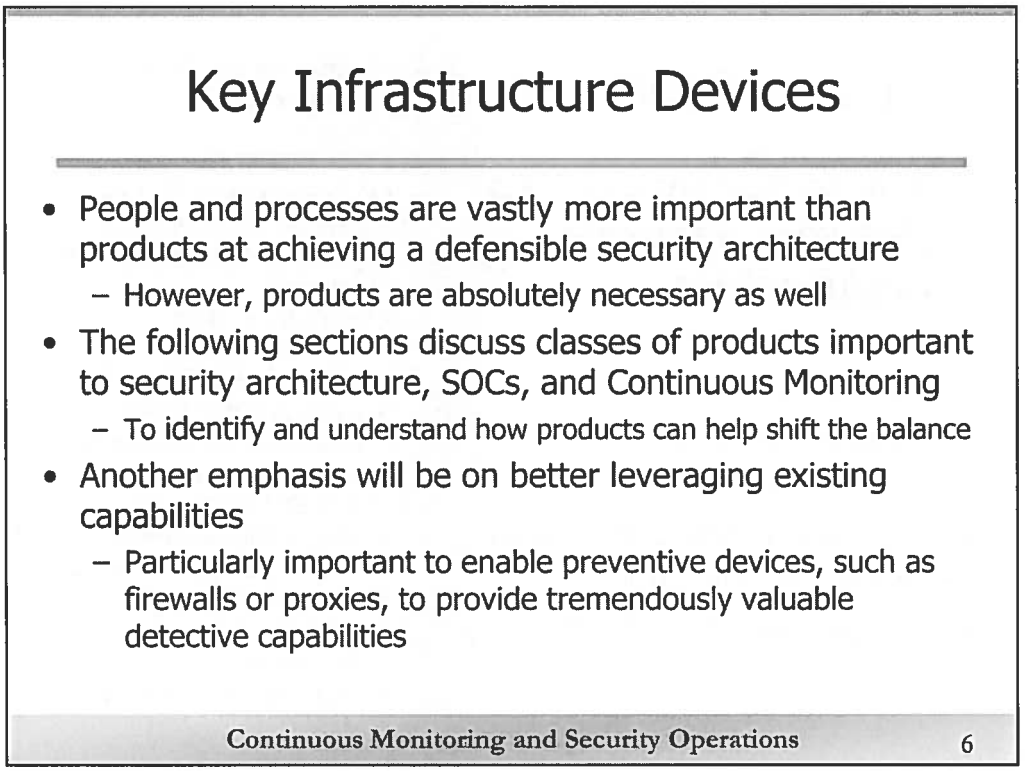
511.2 Table of Contents (2)

This table of contents outlines our plan for 511.2.



Course Roadmap

Let's begin with Network Security Architecture.

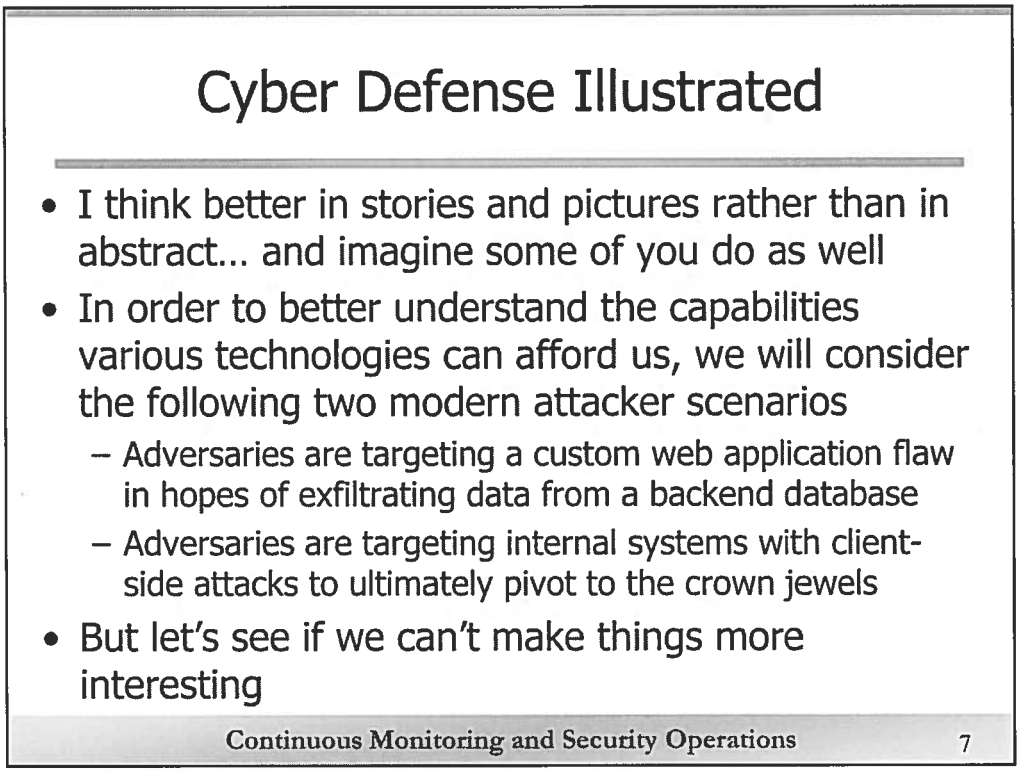


Key Infrastructure Devices

Though we submitted previously that people and process trump products and external services any day of the week, we also need the organization to be efficient. One of the major themes of SANS' Cyber Defense curriculum is the high level flow model Prevent → Detect → Respond.

Given the volume of malicious and benign data, products will almost certainly be a necessary component in our overall security paradigm. Otherwise we would likely not be operating with sufficient efficiency to enable rapid progression from detection to response.

Now, just because we are giving you license to lean on products does not mean that you should employ the typical model of third-party deployment, limited in-house expertise, third-party support/consulting services. No, we are going to focus not just on the basic idea of the device, but how it fits into our overall defensible security architecture that supports modern cyber defense principles.



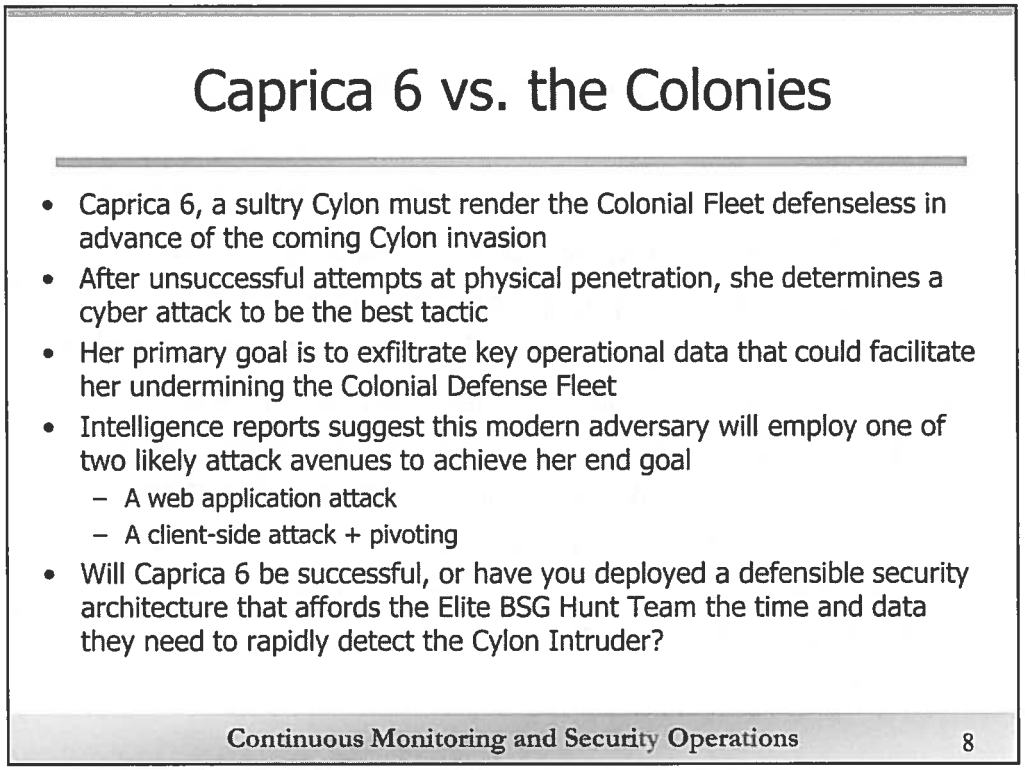
Cyber Defense Illustrated

We will be walking through how to best leverage a number of different devices to support our defensible security architecture. Some of products or techniques you might not have tremendous exposure to. To ensure that you can see how each device fits into the overall security architecture, we will employ two attack scenarios.

These two attack scenarios will help us better visualize the adversaries tactics as well as our own capabilities afforded us by the device under review.

At a high level the two scenarios are:

1. Adversaries are targeting a custom web application flaw in hopes of exfiltrating data from a backend database.
2. Adversaries are targeting internal systems with client-side attacks to ultimately pivot to the crown jewels.



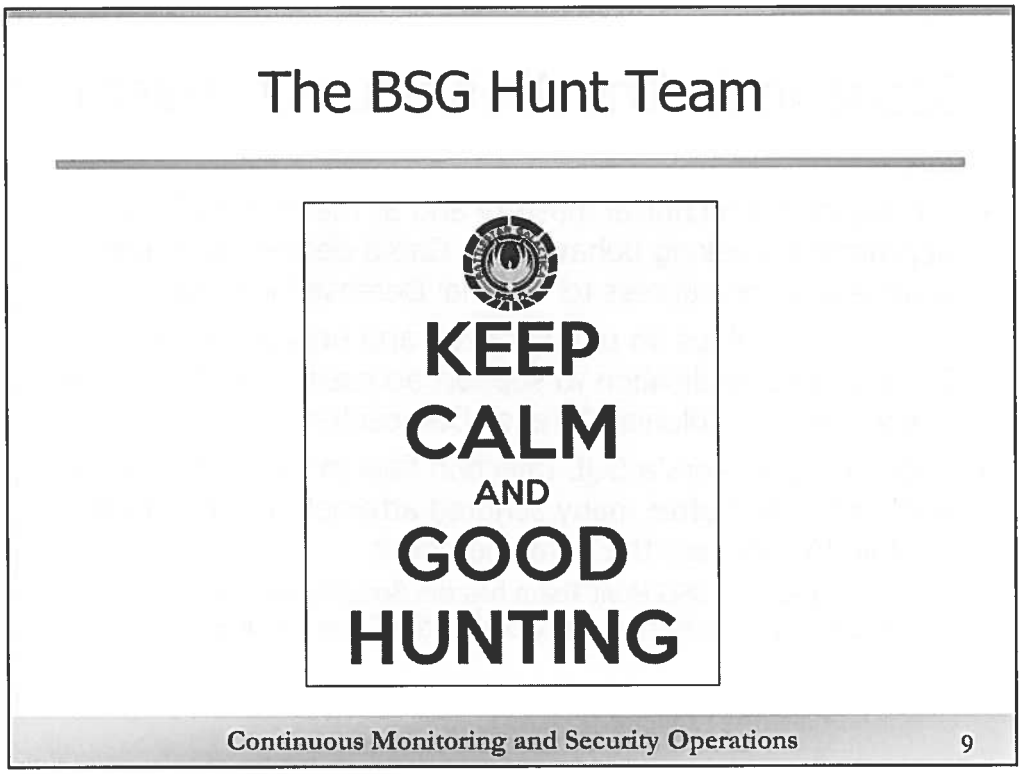
Caprica 6 vs. The Colonies

So let's make it more fun than just a generic adversary… let's make it a story.

We will present two different scenarios that emphasize different aspects of modern attacks that you will no doubt encounter at some time.

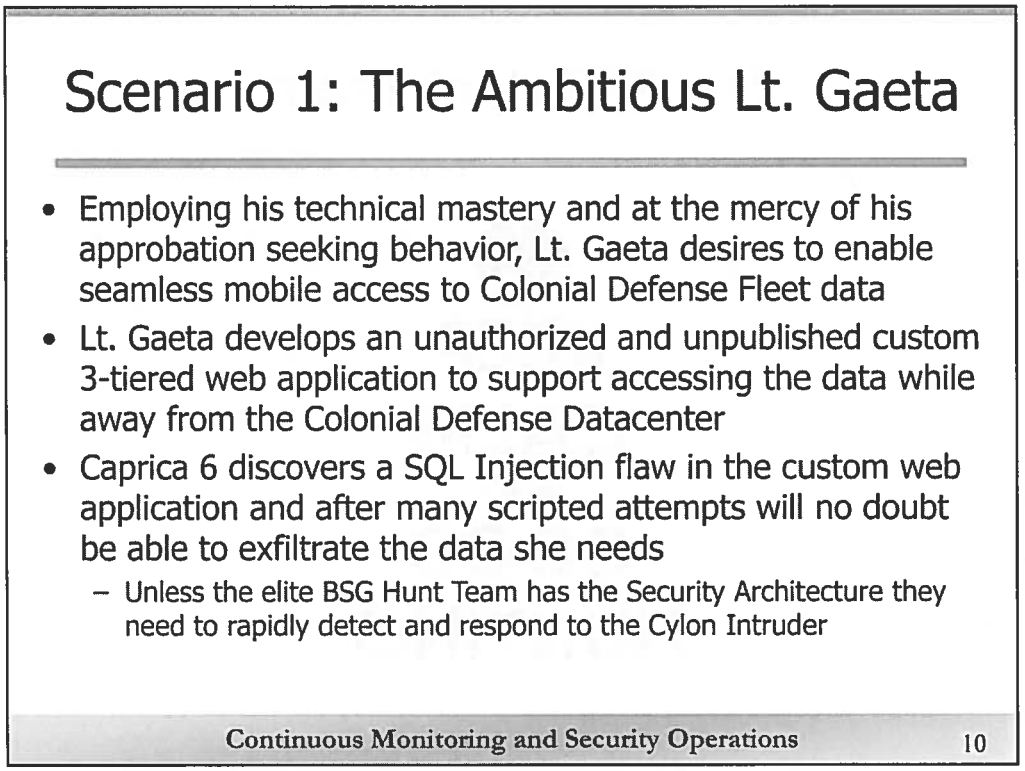
Caprica 6, a humanoid Cylon seeks to user her offensive cyber skills to render the Colonial Fleet defenseless before an upcoming kinetic assault. To achieve this, 6 seeks key sensitive data that will allow her to disable major defensive capabilities. So, ultimately the goal is rendering humans defenses useless, but the means to that end is data housed in the Colonial Defense Fleet's servers.

We will explore two scenarios: a custom web application attack; and a client-side attack + pivoting.



The BSG Hunt Team

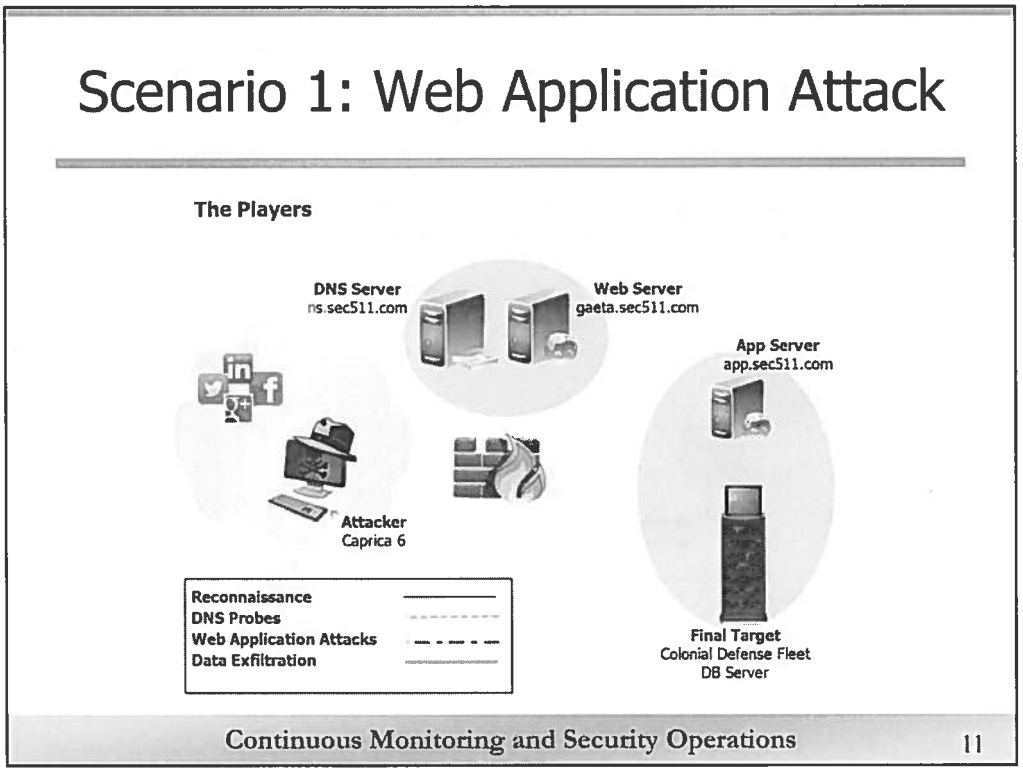
We are part of the Colonial Defense Fleet's BSG Hunt Team responsible for proactive and rapid detection of adversary activities that could cause substantial impact to the Colonies. Given the nature of our role, we need also to understand how to better enable detective capabilities of our infrastructure, and also how we could potentially prevent adversaries from achieving their own goals.



Scenario 1: The Ambitious Lt. Gaeta

The first scenario involves a custom web application developed by Lt. Gaeta to facilitate access to key data from the Colonial Defense's mobile devices. His praise seeking behavior leads him to develop this web application without authorization. To limit potential exposure, he deploys it without providing any public facing links to the test web server hosting the application.

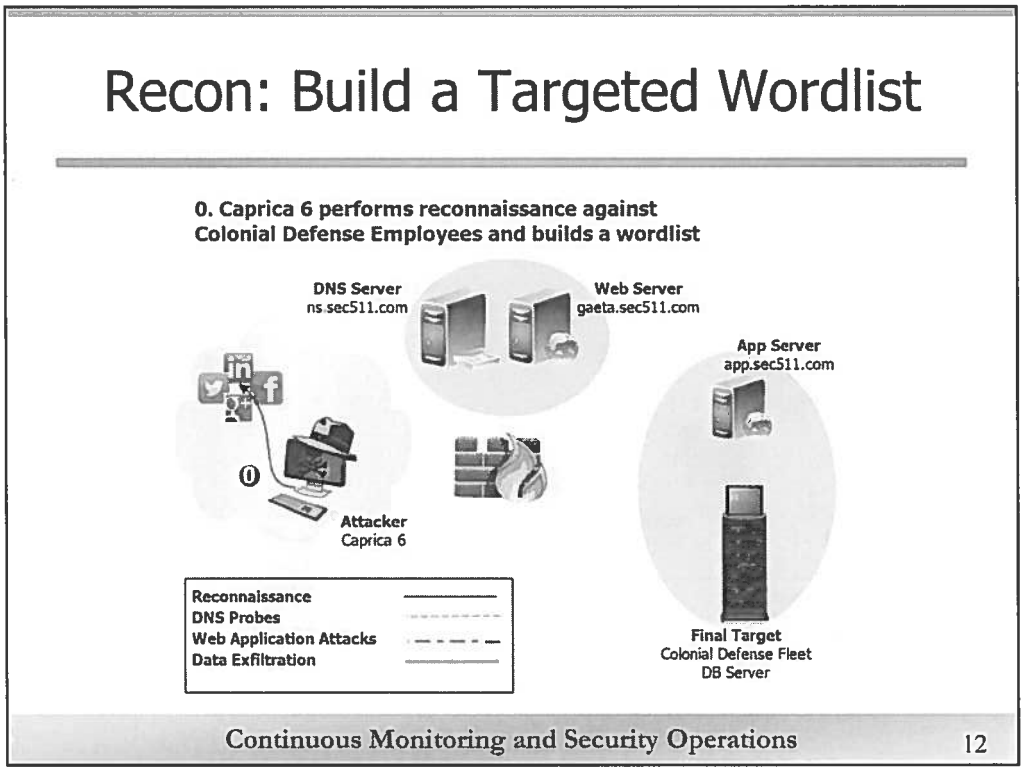
Though technically savvy, Gaeta inadvertently exposes key Colonial Defense data via poor input handling that an adversary can potentially access through exploitation of a SQL Injection flaw.



Scenario 1: Web Application Attack

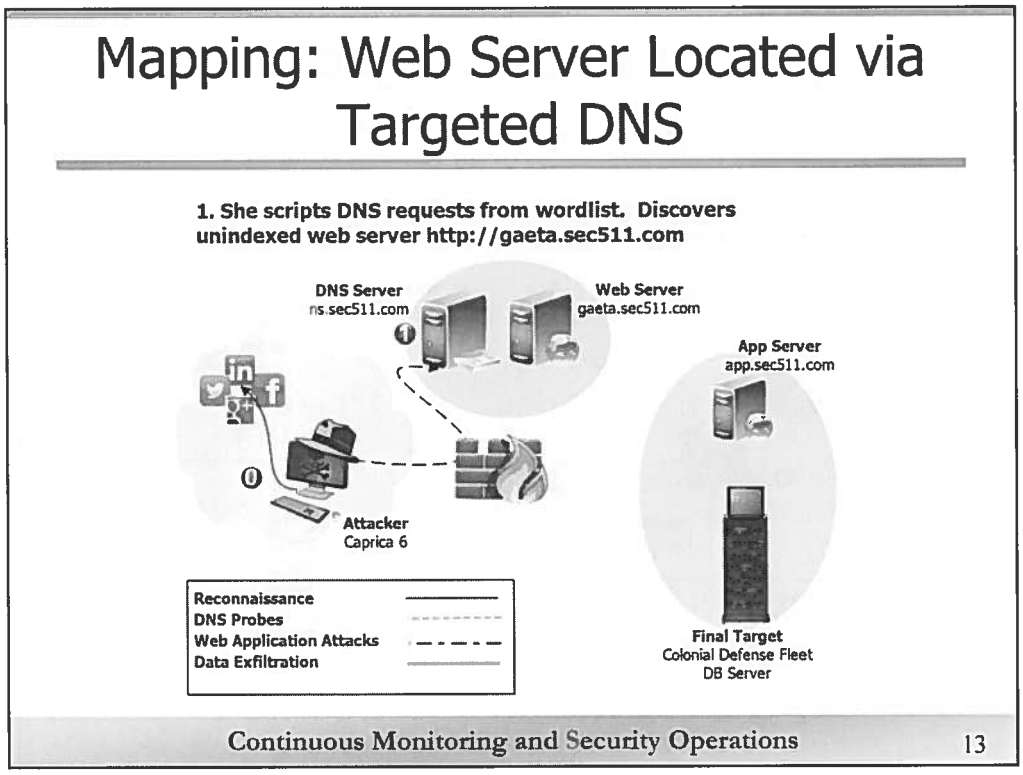
The graphic above shows the players in this scenario.

Adversary - **Caprica 6**
Final Target - **DB Server**
DNS Server - **ns.sec511.com**
Web Server - **gaeta.sec511.com** (no public links to the particular host)
App Server - **app.sec511.com**



Recon: Build a Targeted Wordlist

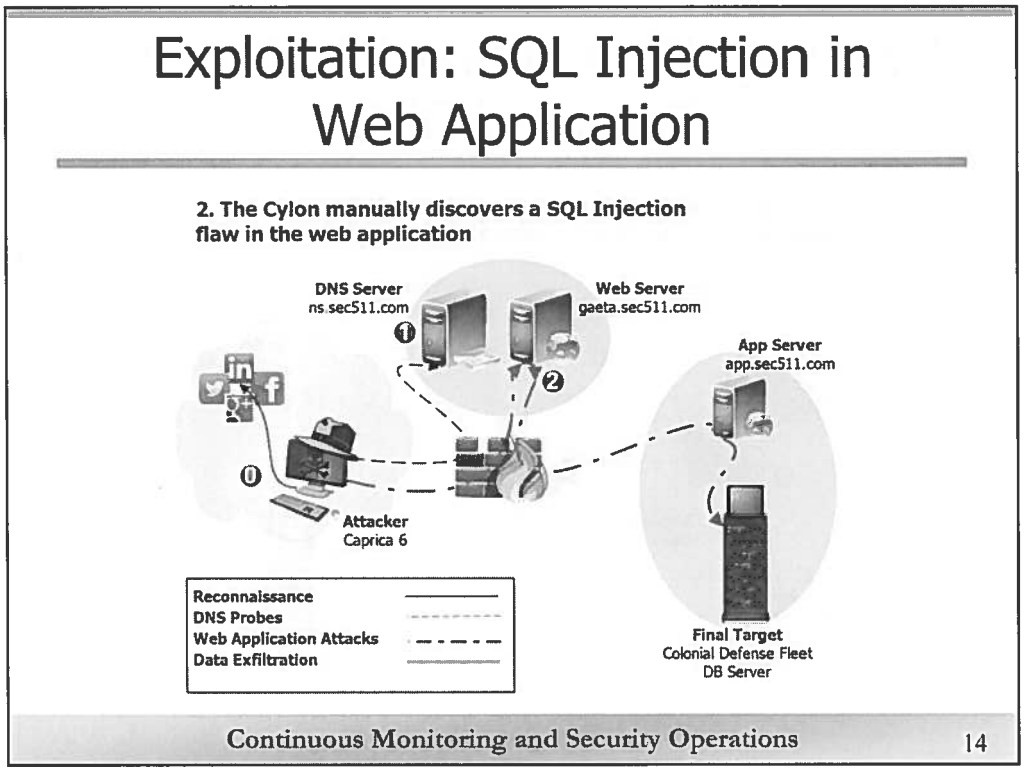
0. Caprica 6 performs reconnaissance against Colonial Defense Employees public facing information. She builds a wordlist that can be leveraged as potential usernames, passwords, etc.



Mapping: Web Server Located via Targeted DNS

1. After unsuccessful attempts at a zone transfer, 6 scripts DNS requests to brute force any potential unpublished hostnames. She leverages her recon wordlist and adds those words into the `namelist.txt` used by Carlos Perez's (@dark0perator) **dnsrecon**[1] tool. She discovers the unpublished web server at http://gaeta.sec511.com.

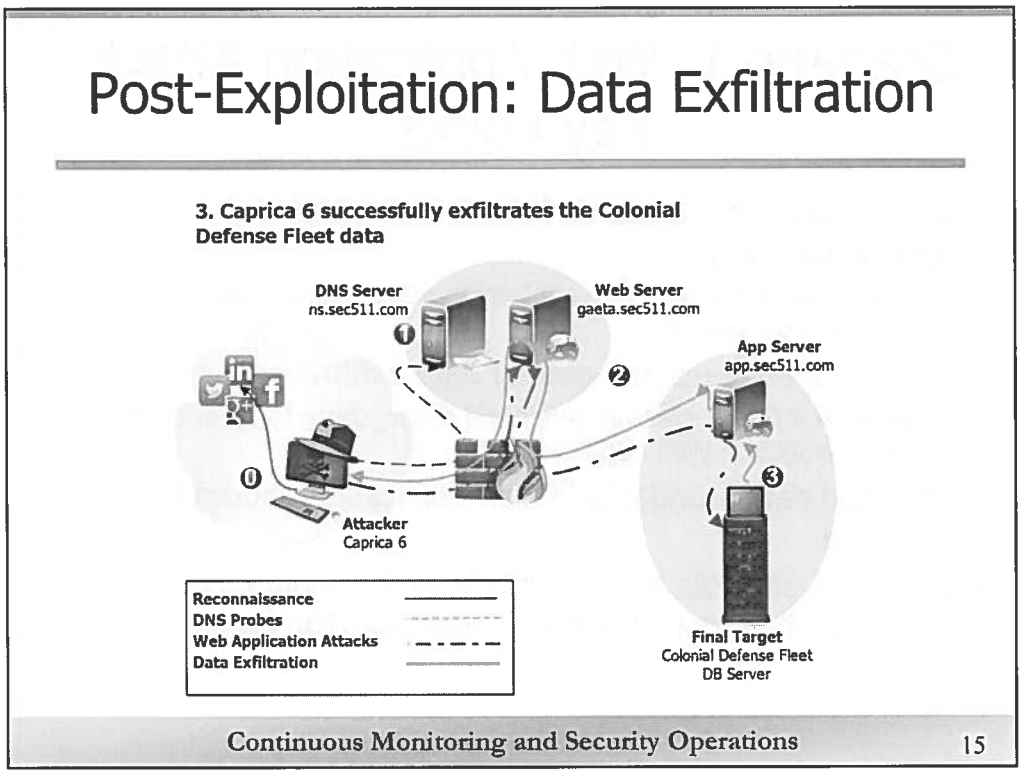
[1] https://github.com/darkoperator/dnsrecon (http://cyber.gd/511_210) **QR**



Exploitation: SQL Injection in Web Application

2. Using Daffyd Stuttard's (@portswigger) Burp Suite[1], Caprica 6 discovers a potentially exploitable SQL Injection flaw in the web application.

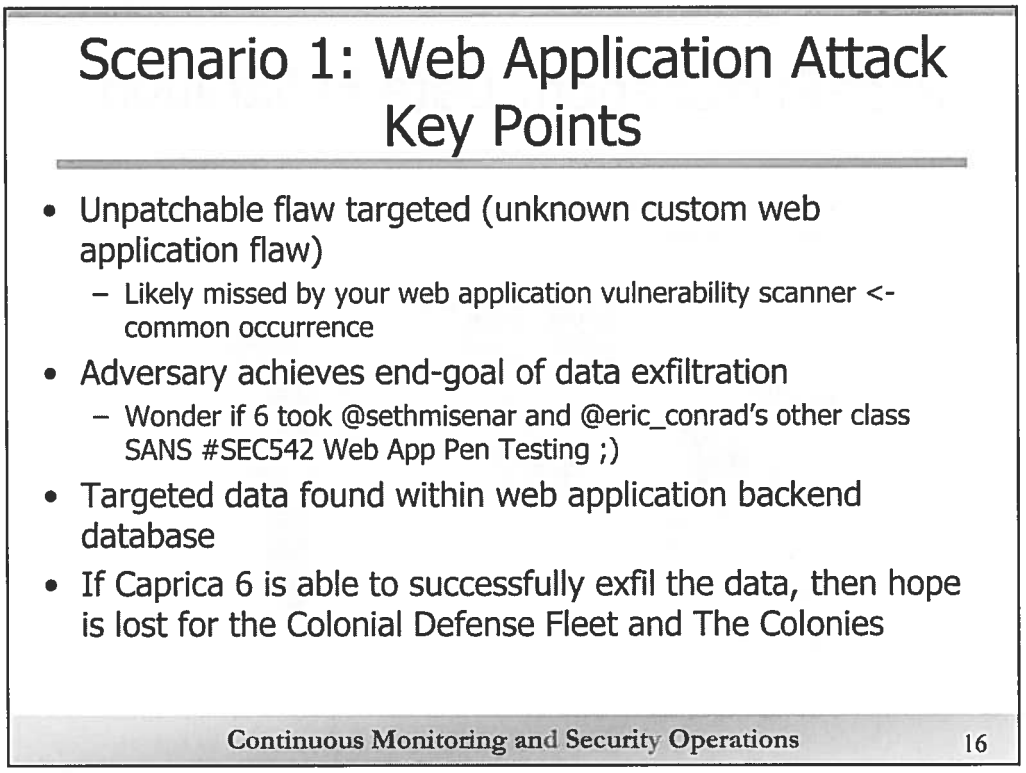
[1] http://portswigger.net/burp/ (http://cyber.gd/511_29) **QR**



Post-Exploitation: Data Exfiltration

3. After fuzzing the SQL Injection flaw using **Burp**, and subsequently leveraging **sqlmap**[1] for exploitation, the Cylon was able to exploit the SQL Injection flaw and dump key data and exfiltrate it back out the same patch used into the organization.

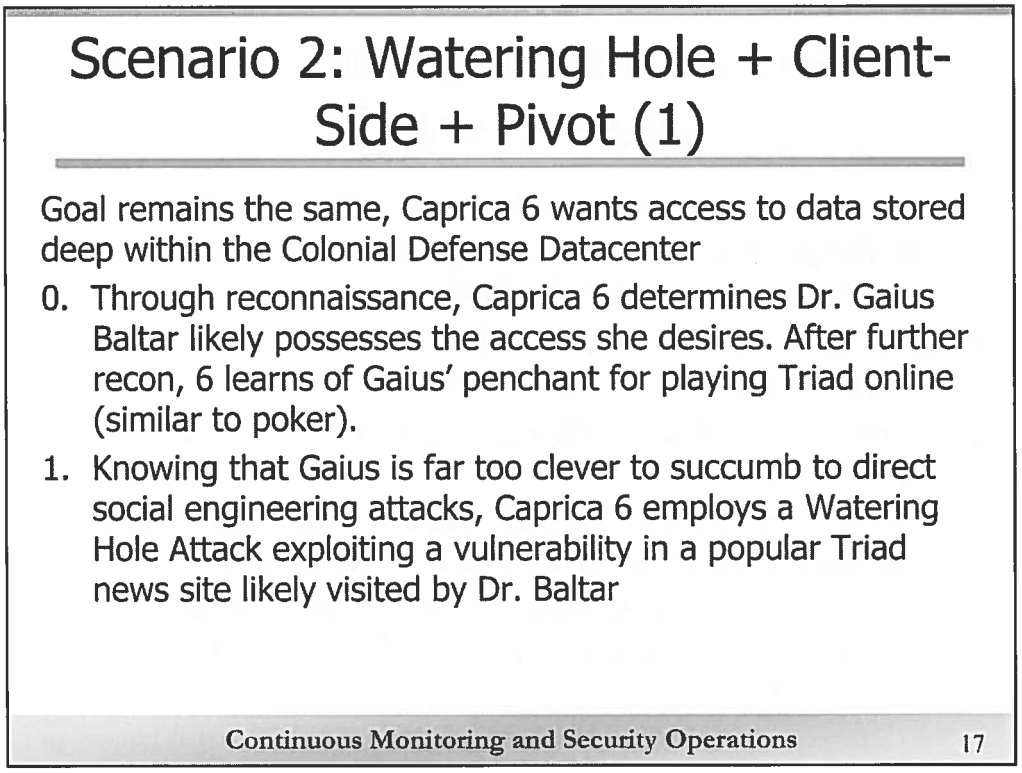
[1] http://sqlmap.org/ (http://cyber.gd/511_211) **QR**



Scenario 1: Web Application Attack Key Points

This scenario serves as an interesting case study for our architectural review due to the increasing likelihood that organizations not only have web applications, but also ones that ultimately might provide access to key business functionality or sensitive data. Note also that the SQL Injection attack yielding sensitive would have been more likely to have been part of an exploit chain that allowed for exploitation of an internal web application. However, for simplicity sake, and because the next scenario already hits on pivoting, we decided to make it conceptually a bit simpler.

Custom web applications are ubiquitous, and many have egregious flaws that go unnoticed for long periods of time because a vendor doesn't supply fixes/patches for your own personal busted code. This speaks to another central point, namely, that this scenario did not involve a patchable flaw. Yes, the code could be fixed, but there was no patch that was simply missing, there was no Critical/Level 5/CAT 1 vulnerability noticed by your vulnerability scanner.

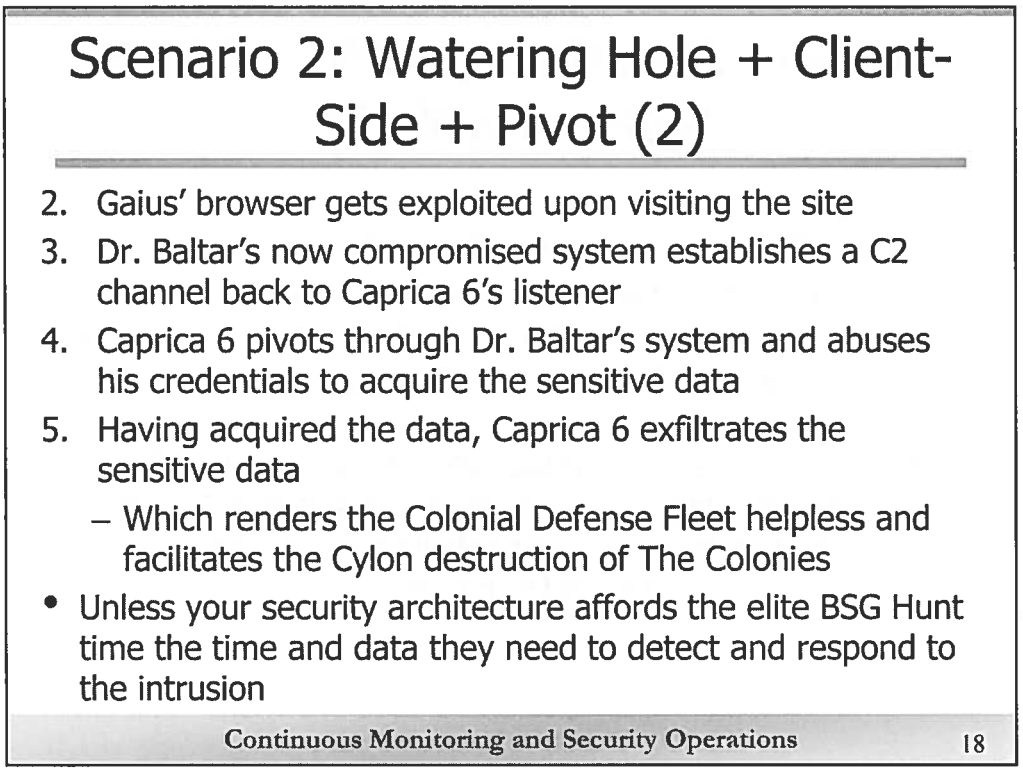


Scenario 2: Watering Hole + Client-Side + Pivot (1)

For the next scenario, Caprica 6's goal of exfiltrating sensitive data remains the same. This scenario will involve targeted client side exploitation as well as an internal pivot. Both of these activities are commonplace, and yet every enterprise still struggles with these types of activities.

Here is a text-based walk through the scenario:

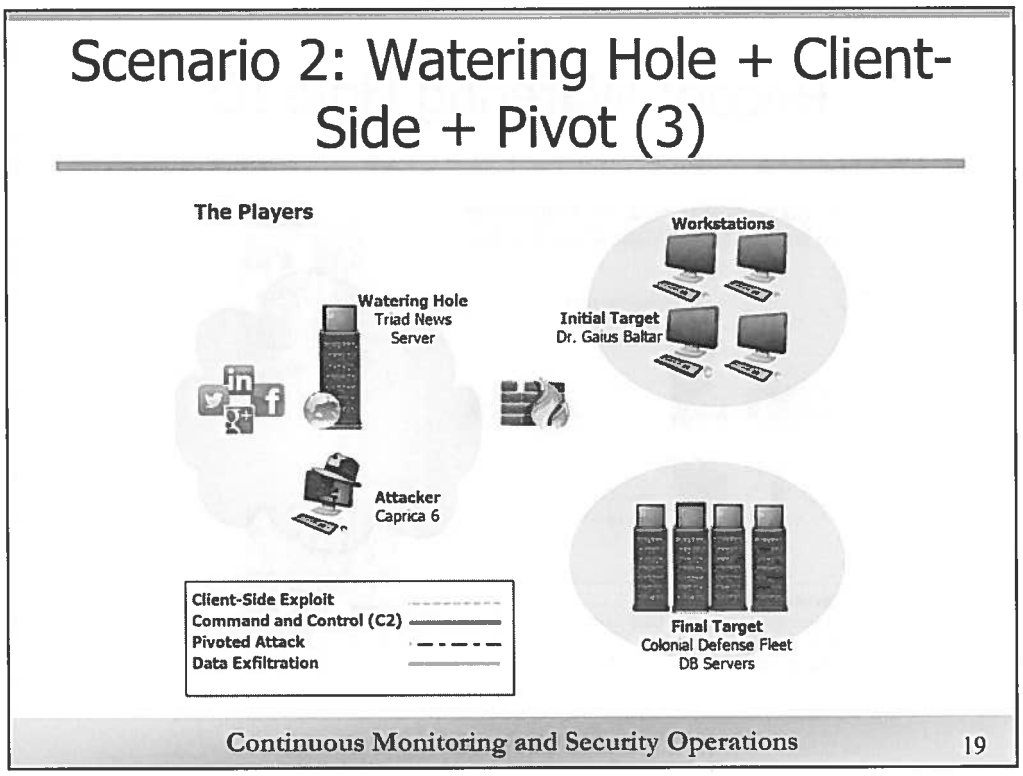
0. Through reconnaissance, Caprica 6 determines Dr. Gaius Baltar likely possesses the access she desires. After further recon, 6 learns of Gaius' penchant for playing Triad online (similar to poker).
1. Knowing that Gaius is far too clever to succumb to direct social engineering attacks, Caprica 6 employs a Watering Hole Attack exploiting a vulnerability in a popular Triad news site likely visited by Dr. Baltar.



Scenario 2: Watering Hole + Client-Side + Pivot (2)

Continuing our text-based walkthrough of the scenario:

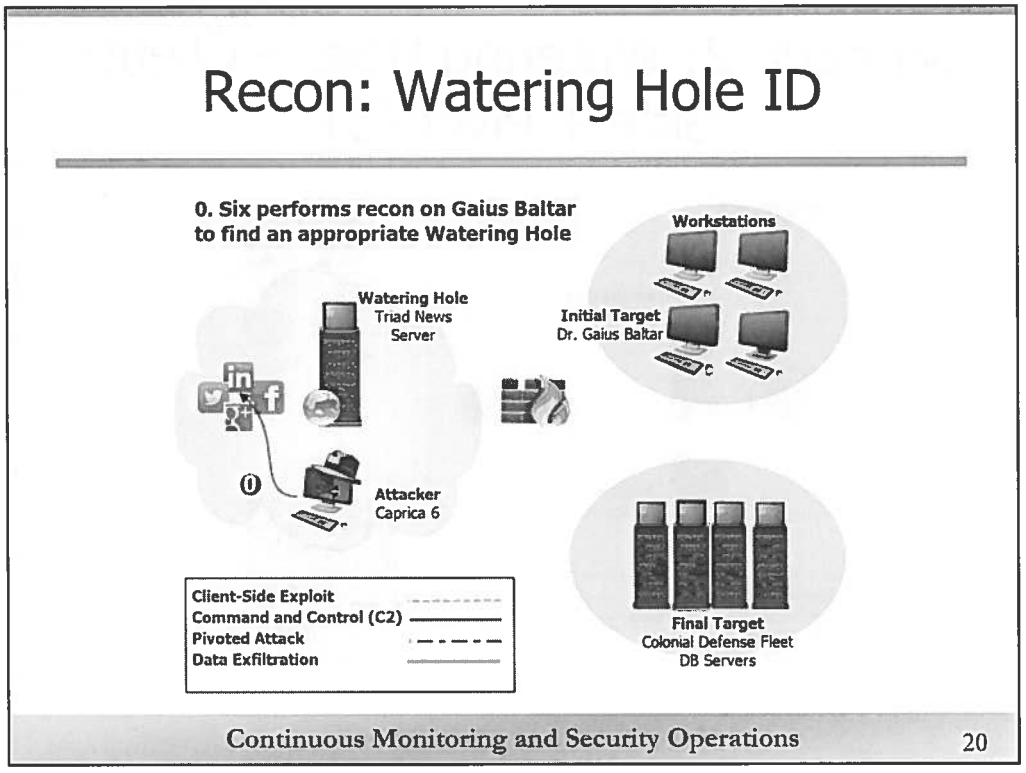
2. Gaius' browser gets exploited upon visiting the site.
3. Dr. Baltar's now compromised system establishes a C2 channel back to Caprica 6's listener.
4. Caprica 6 pivots through Dr. Baltar's system and abuses his credentials to acquire the sensitive data.
5. Having acquired the data, Caprica 6 exfiltrates the sensitive data.



Scenario 2: Watering Hole + Client-Side + Pivot

Players:

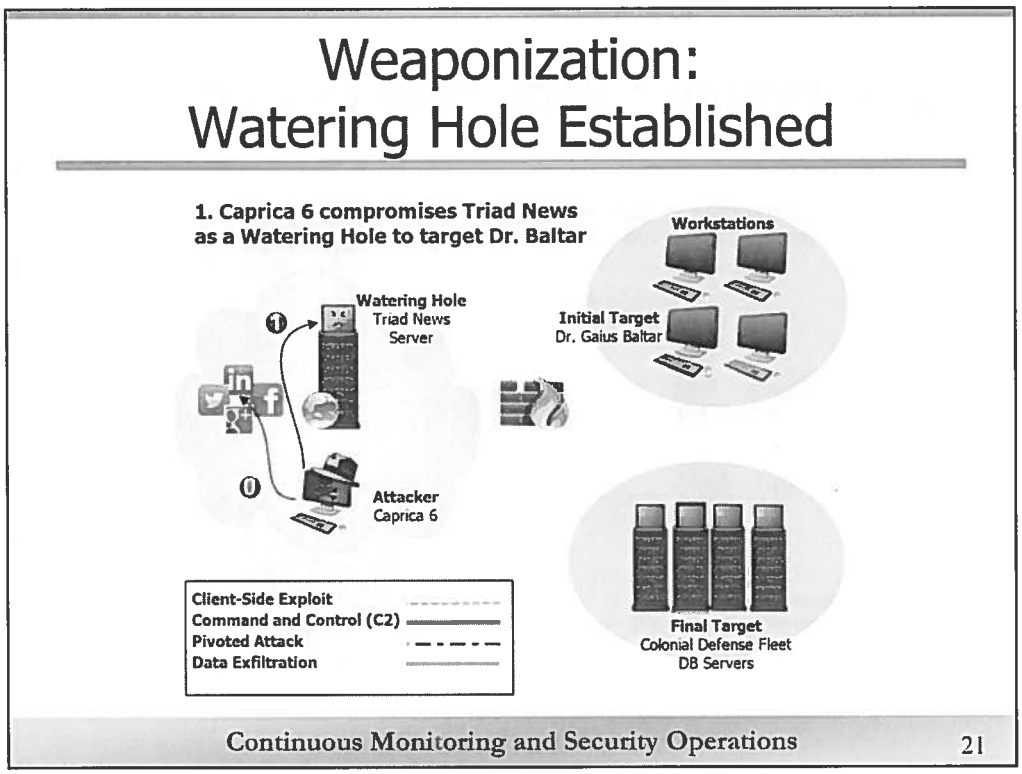
Adversary: Caprica 6
Watering Hole: Triad News Server
Initial Target: Dr. Gaius Baltar
Final Target: CDF Servers



Recon: Watering Hole ID

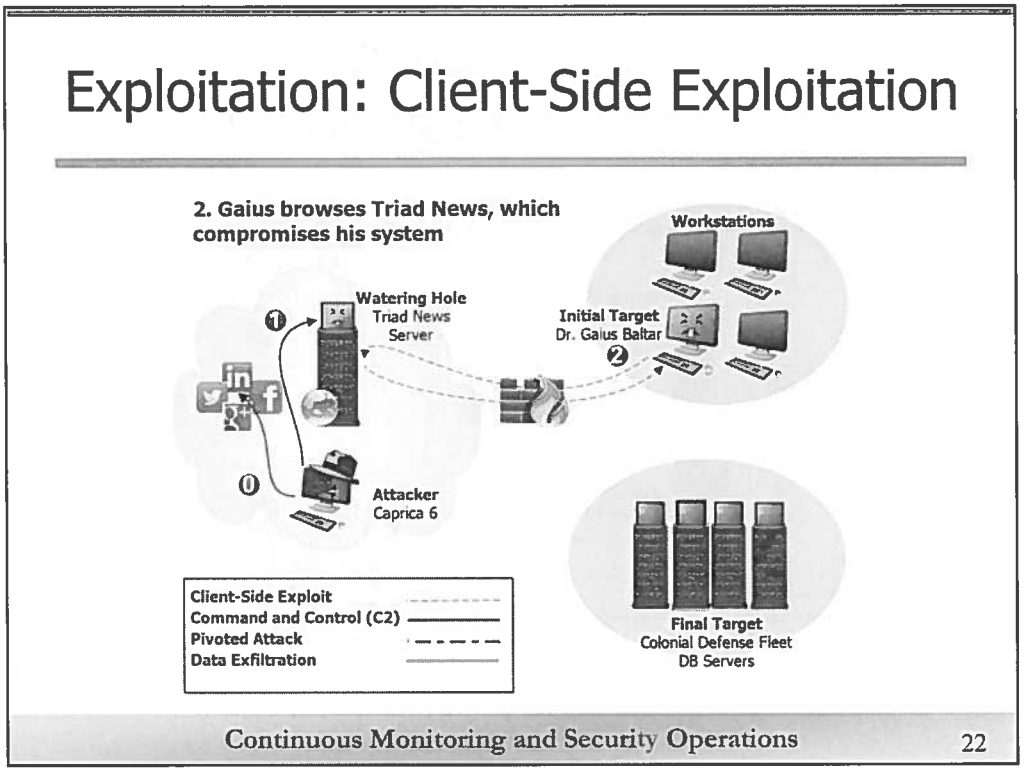
0. Six leverages recon-ng, written by friend and fellow SANS Instructor, Tim Tomes (@LaNMaSteR53), to determine that Dr. Gaius Baltar appears to be a likely victim. Further reconnaissance suggests a potential Watering Hole to allow for more subtle compromise of Baltar, which is warranted given his penchant for paranoia.

[1] https://bitbucket.org/LaNMaSteR53/recon-ng/ (http://cyber.gd/511_212) **QR**



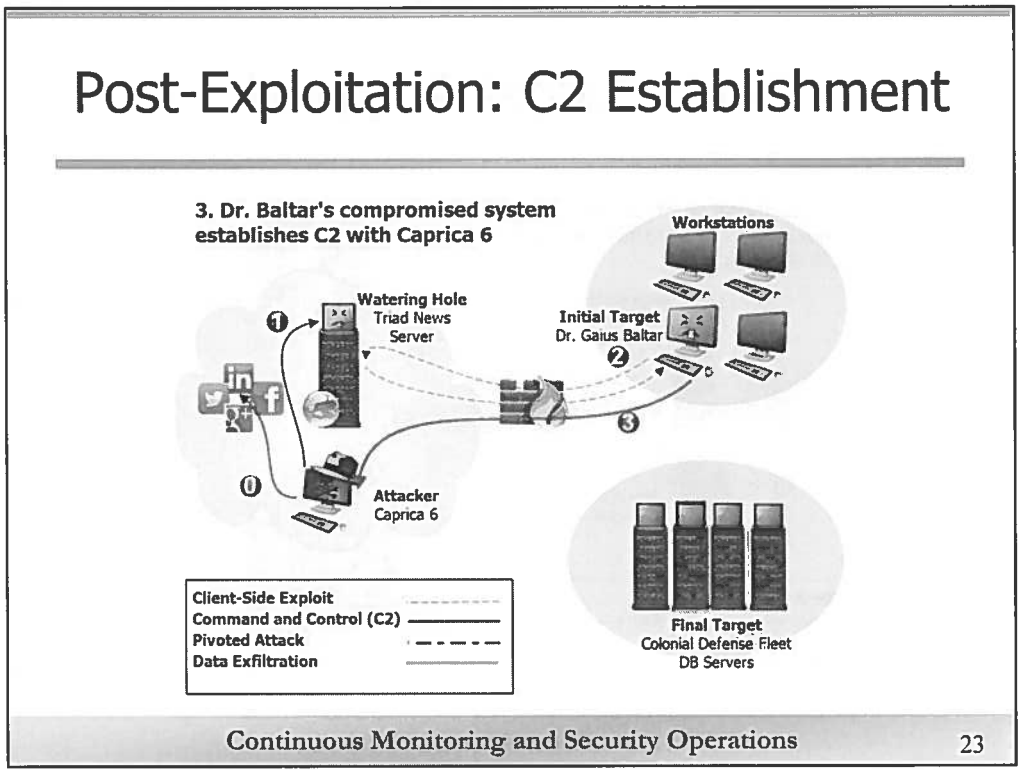
Weaponization: Watering Hole Established

1. Caprica 6 compromises the Triad News website. This site will unwittingly serve as the Watering Hole where 6 expects Baltar to eventually come for a drink (and a value added exploit).



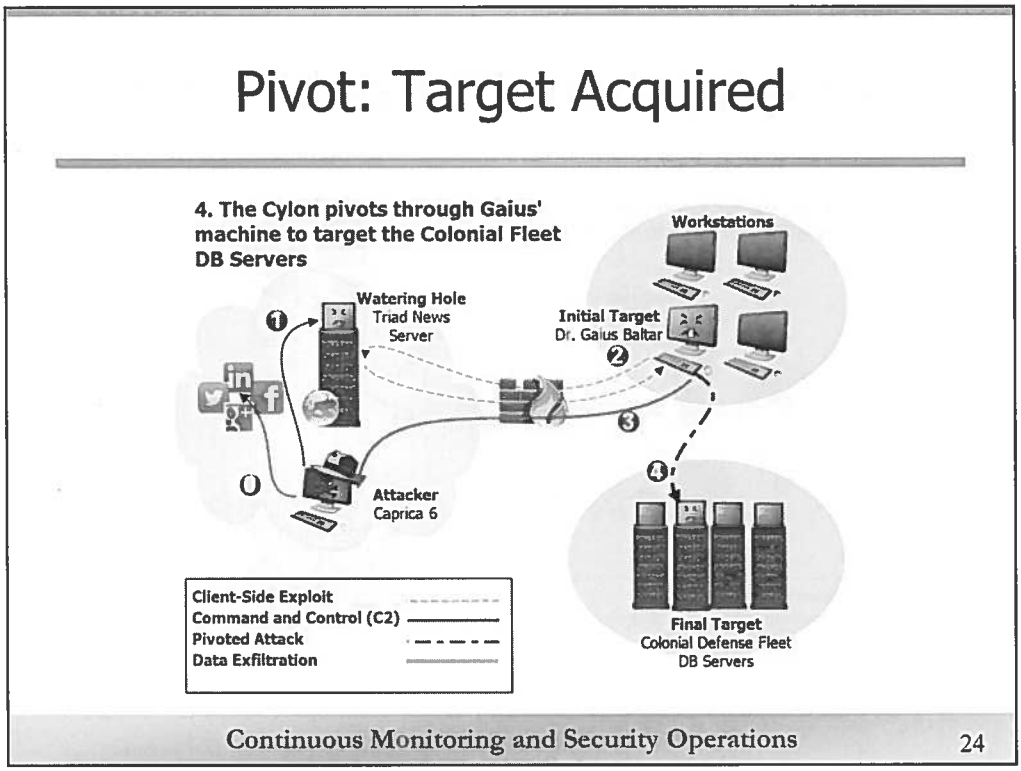
Exploitation: Client-Side Exploitation

2. Gaius ends up drinking from the Watering Hole, Triad News Server, and his system becomes compromised.



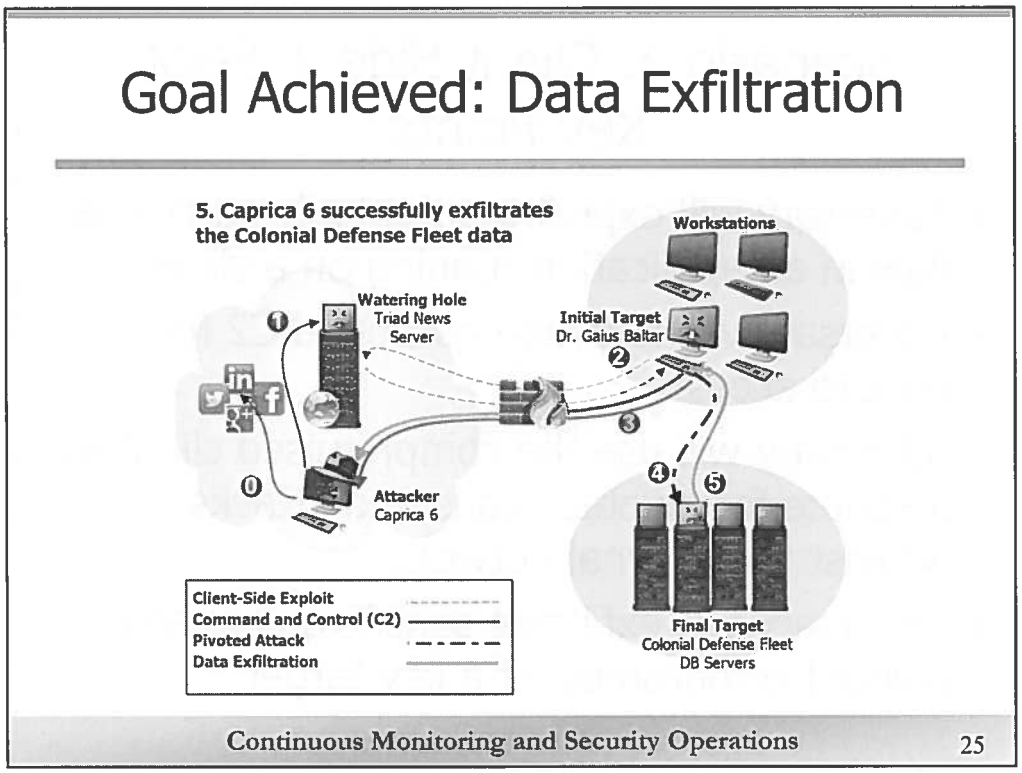
Post-Exploitation: C2 Establishment

3. Baltar's compromised machine initiates an outbound connection to Caprica 6's system establishing a C2, Command and Control, channel.



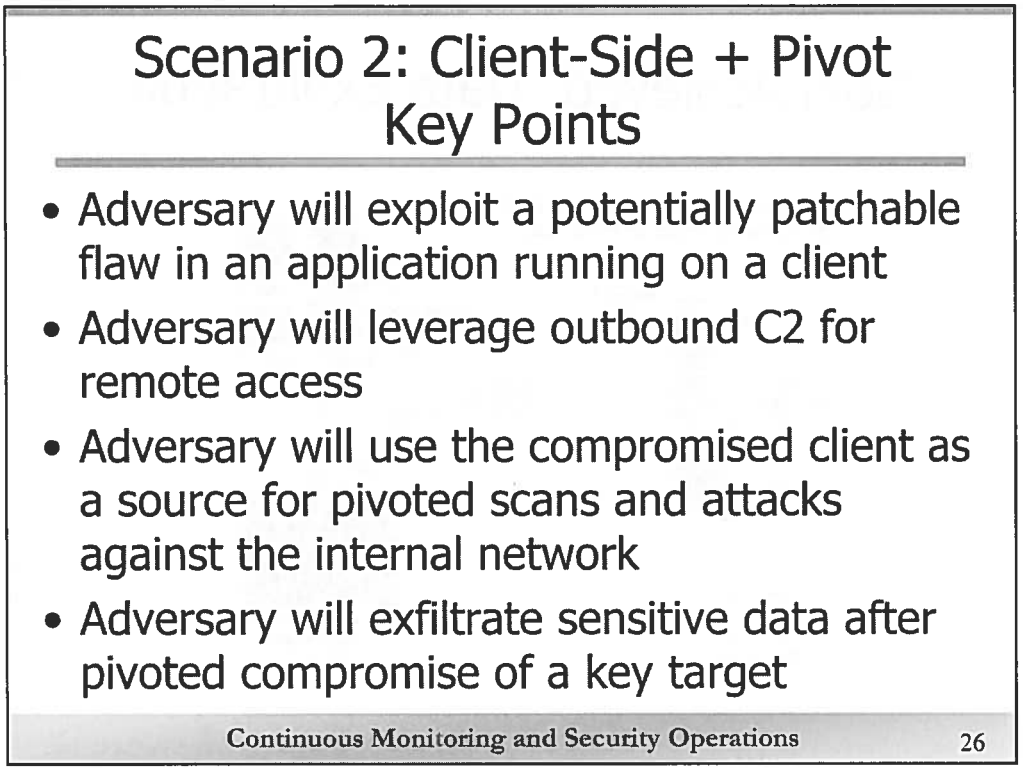
Pivot: Target Acquired

4. Six abuses Gaius' Access Token and successfully pivots to connect to the Colonial Defense Fleet servers.



Goal Achieved: Data Exfiltration

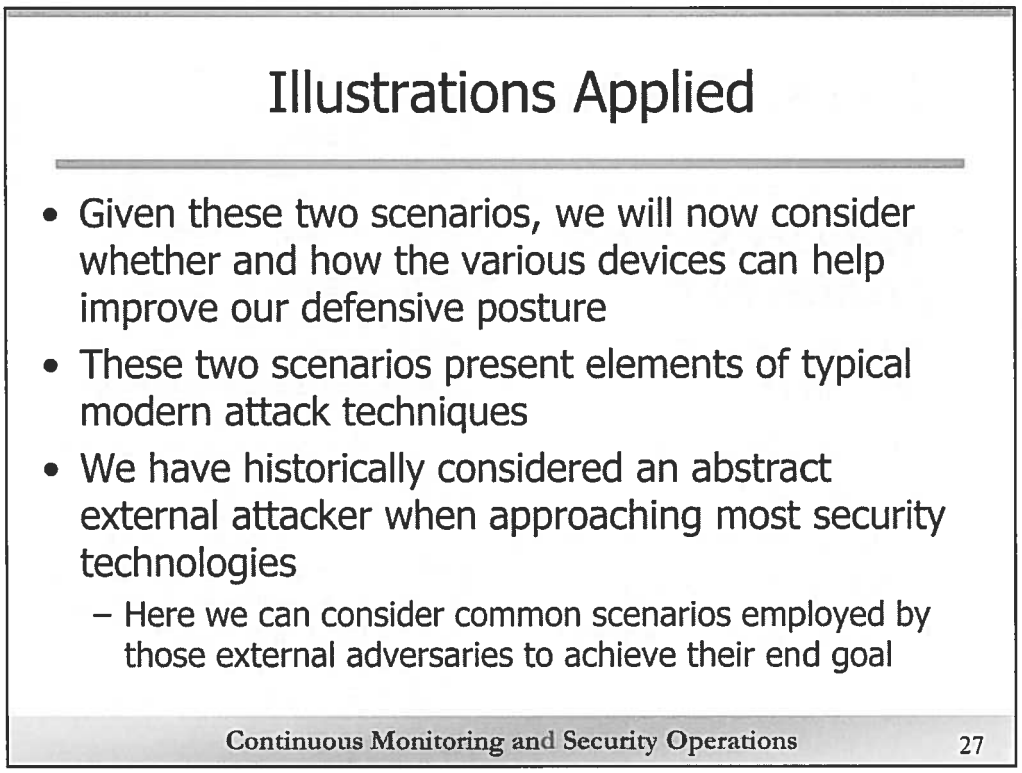
5. Caprica then exfiltrates data over her existing C2 channel.



Scenario 2: Client-Side + Pivot Key Points

Some of the key attributes of the 2nd scenario include the following. The adversary will exploit a patchable flaw in a client application. An outbound C2 channel will be leveraged to allow for successful command and control. This same channel will ultimately be used for exfiltration in this case. Leveraging the access on the compromised system, the adversary will pivot to scan and attack internal systems until finding the target portion of the network needed.

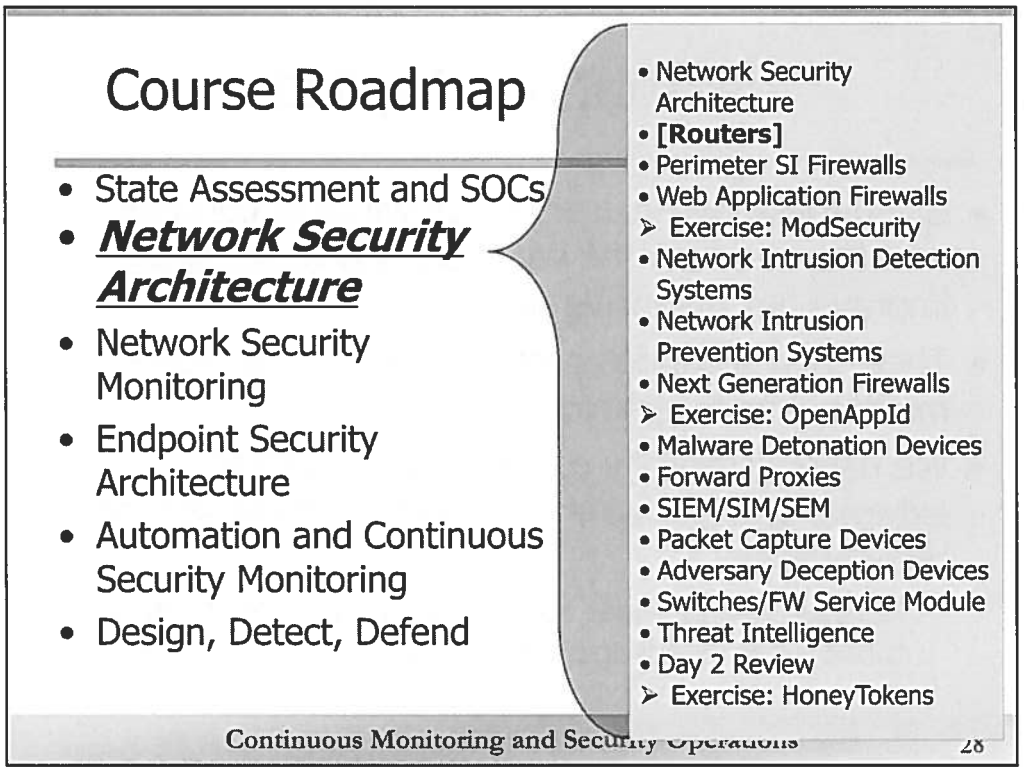
Though this may seem like a lot of moving parts, most compromises that result in breach are more complicated and involved than what is expressed here. Though the attack need not be more sophisticated in call cases, various elements could be more complex, surreptitious, or distributed.



Illustrations Applied

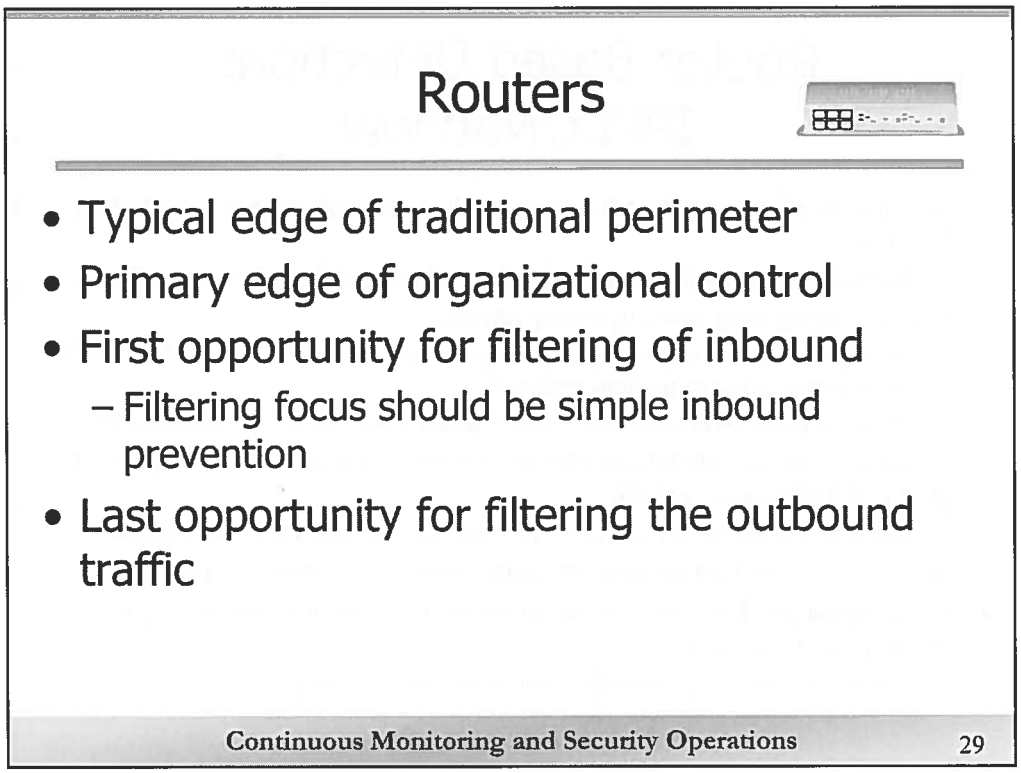
The purpose of these scenarios is to provide us a serviceable backdrop against which to juxtapose the various elements of our network security architecture. Though these two scenarios do not represent an exhaustive review of all adversary actions, they will provide a more than ample starting point for our discussions of the merits in both a preventing and detective capacity.

Web application attacks, client side exploitation, and pivoting are very commonly elements of modern cyber campaigns. They also happen to be two particular areas where many traditional technologies (and some newer ones) are wanting, particularly from the prevention of compromise vantage point.



Course Roadmap

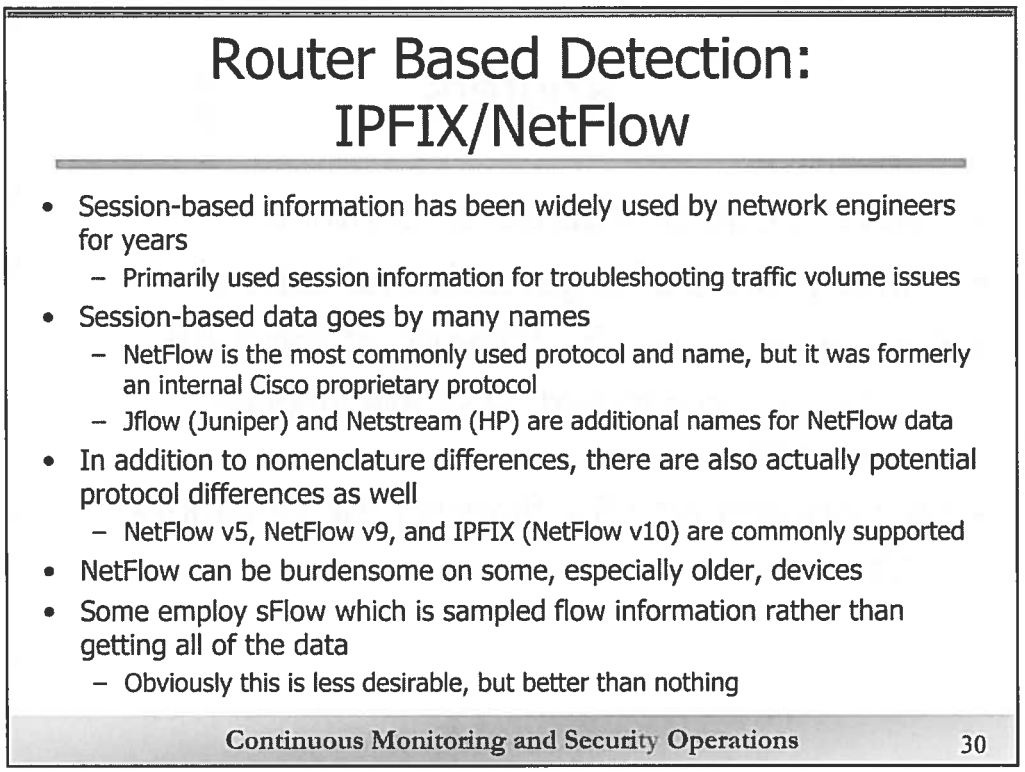
Now let's look at routers.



Routers

Though the router is not overtly a security device, given the location it resides makes it a device requiring consideration. Even though there are some overt router-centric security capabilities, our primary motivation for attending to the route is due to its position at the edge.

The router is typically the edge of a traditional perimeter. At the edge, the router represents the last opportunity for outbound filtering/monitoring, and also the first opportunity for inbound filtering/monitoring. Another reason to consider the router is due to the fact that it often represents the edge of our control and ownership (though in some smaller shops or remote offices, the company might merely lease the router).



Router Based Detection: IPFIX/NetFlow

The primary purpose of NetFlow[1] was initially to aid network engineers to better troubleshoot performance issues. Further, NetFlow better enabled rapid root cause analysis of the underlying problem leading to performance issues.

Prior to NetFlow the main built-in performance troubleshooting capability of network devices was simply to look at port statistics. With NetFlow, the engineer does not simply see mere port utilization, but rather can see some Layer 3 (IP) and Layer 4 (TCP/UDP) information. This allows for better understanding what particular application or service might be causing potential issues.

Though NetFlow has been widely used by network engineers for years, and is likely already enabled, many security practitioners are still unaware of this capability. Though, as we will discuss later, full packet captures are the gold standard in network traffic monitoring, especially for deep dive post-mortem review, NetFlow can enable rapid detection without the higher cost associated with full packet capture[2].

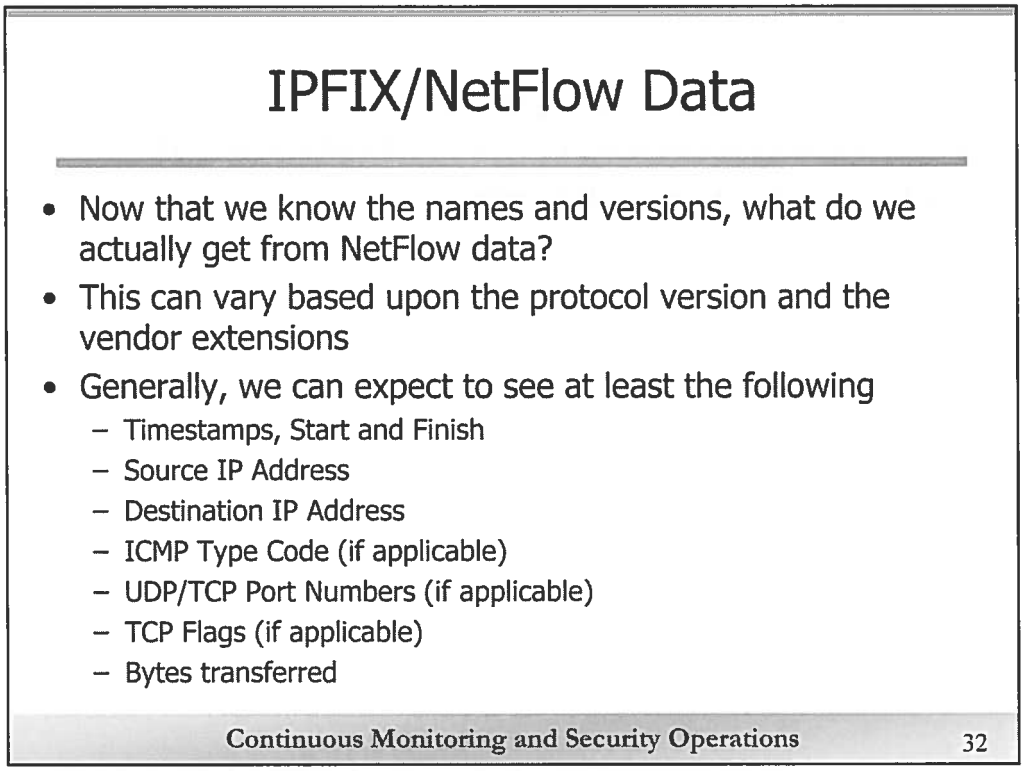
Though the term NetFlow is widely used in a generic way to refer to session-based logging capabilities of network devices, vendors other than Cisco often provide the same capabilities under a different name.

The public RFC is associated rather with IPFIX[3] (NetFlow v10), which was based on NetFlow v9.

[1] http://tools.ietf.org/html/rfc3954 (http://cyber.gd/511_213)

[2] http://www.first.org/global/practices/Netflow.pdf (http://cyber.gd/511_215) **QR**

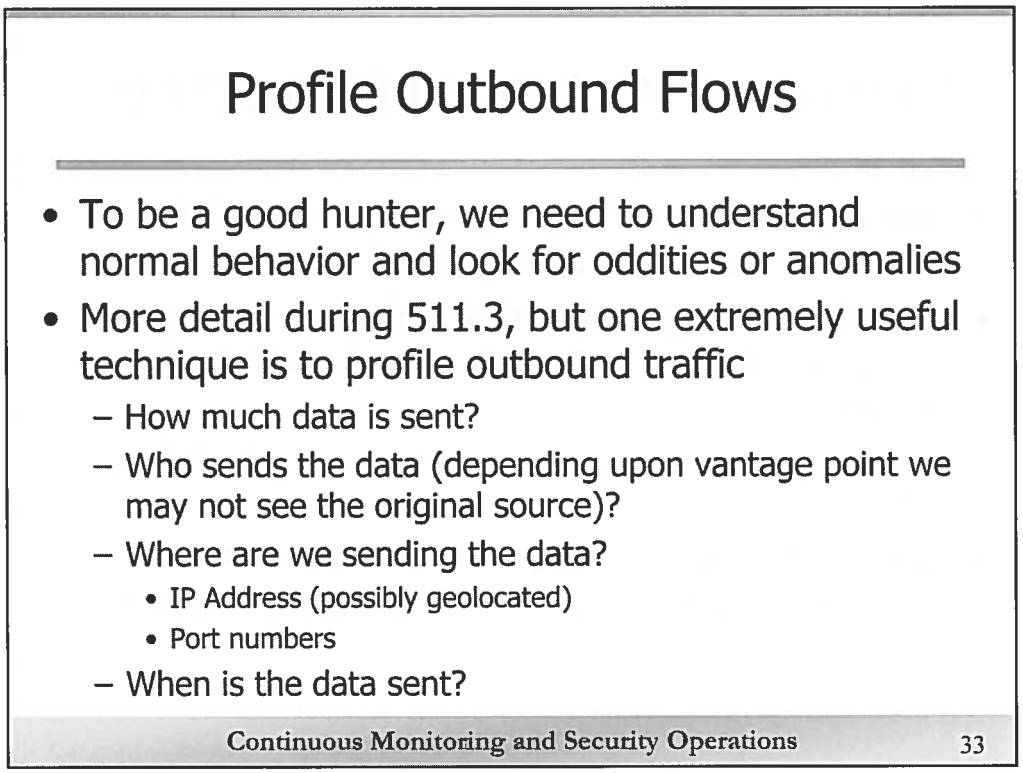
[3] https://tools.ietf.org/html/rfc7011 (http://cyber.gd/511_214)



IPFIX/NetFlow Data

The major versions of NetFlow (v5, v9, and v10/IPFIX) all provide session-based information. The more recent versions are more likely to include customizable user fields to be pulled. Generally, NetFlow records will all provide the following information:

- Timestamps, Start and Finish
- Source IP Address
- Destination IP Address
- ICMP Type Code (if applicable)
- UDP/TCP Port Numbers (if applicable)
- TCP Flags (if applicable)
- Bytes transferred



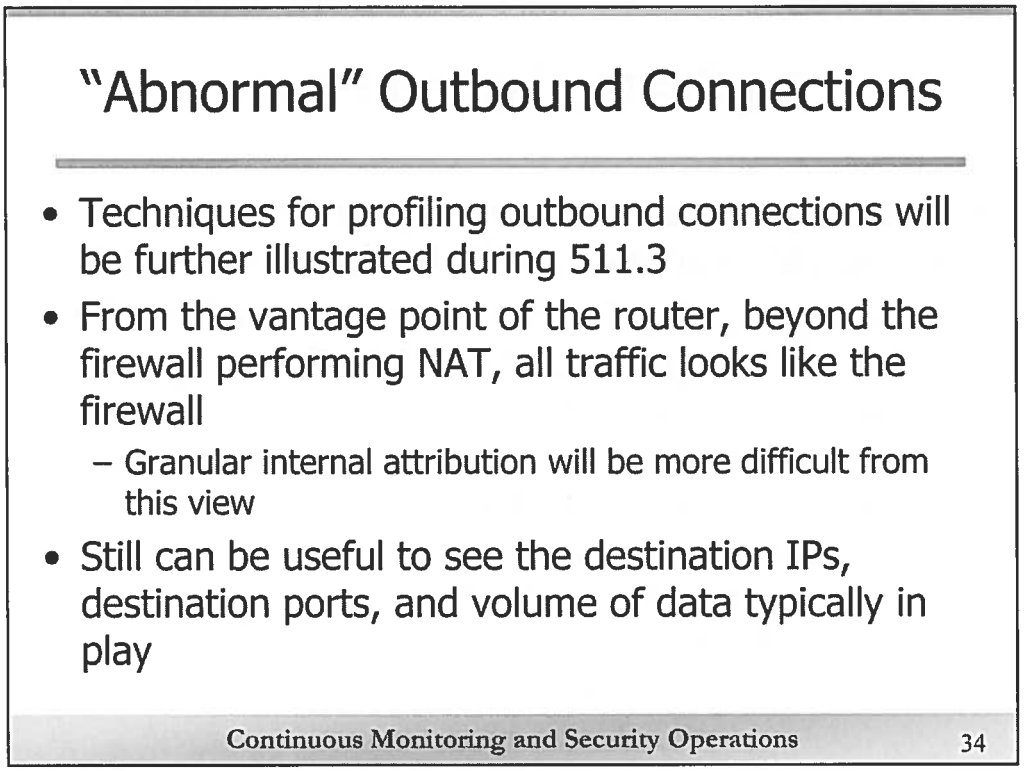
Profile Outbound Flows

NetFlow does not provide visibility into layer 7 payload data, for that we would require something like full packet capture. However, given even just the layer 3/layer 4 information, we can gain significant intelligence. Using NetFlow information, we can very quickly begin to characterize outbound traffic/flows.

Some items to consider that NetFlow can provide:

- The volume of data transferred;
- Who (IP address at least) sourced the data, which will very likely be just the firewall assuming it is performing NAT;
- Where in the world we are sending data (when the destination IP is coupled with GeoIP sources);
- What ports are being leveraged for communication;
- When is the data sent?

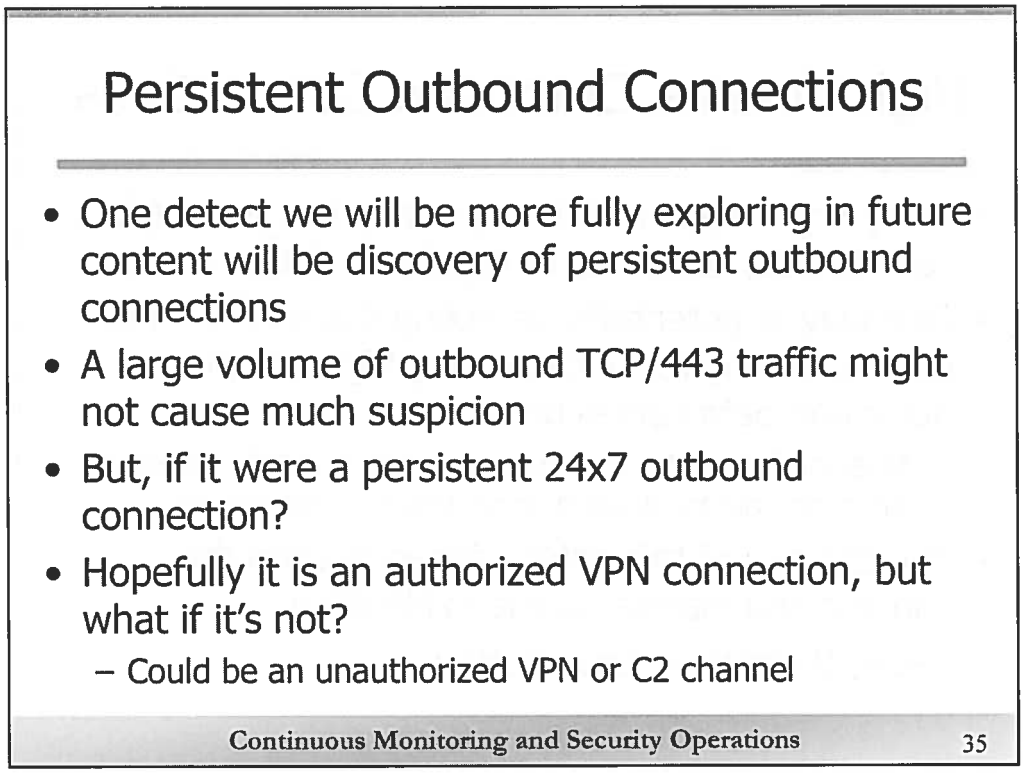
Answers to these questions can be hugely beneficial for profiling communications and looking for outliers with respect to outbound communication.



"Abnormal" Outbound Connections

We will be leveraging outbound connection profiling and looking for anomalous or overtly suspicious behavior during the discussion of Network Security Monitoring (NSM) in 511.3. As mentioned previously, though we can gain significant insight into outbound traffic, it could be difficult to determine the actual source of the traffic, depending upon the network architecture.

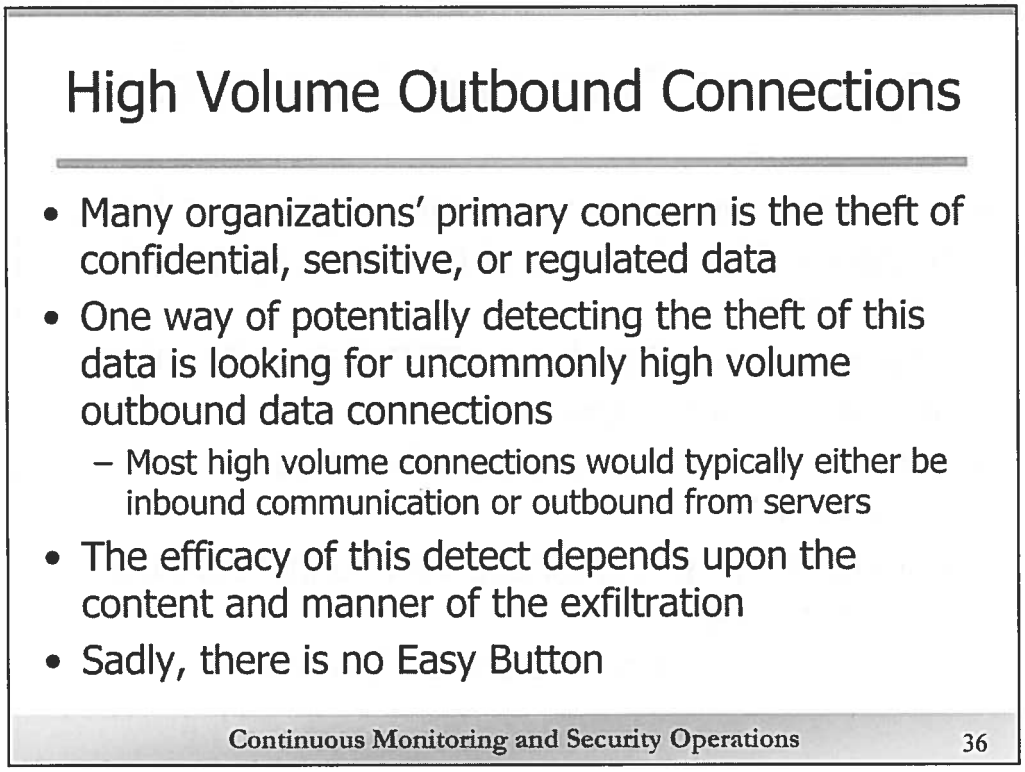
The router would likely only be able to attribute the traffic to the device performing NAT for outbound traffic, quite likely the firewall. This is unfortunate, but could still allow us to find issues that warrant further review.



Persistent Outbound Connections

A technique that we will be exploring more fully later in the course is identification and characterization of persistent outbound connections. Though you are likely to encounter some legitimate persistent outbound connections, site-to-site VPNs for example, you will often also find a number of unauthorized VPNs in the form of adversary C2 or perhaps even policy-violating insiders.

These are fairly straightforward opportunities to detect, that most organizations are already reviewing.



High Volume Outbound Connections

Data compromise represents many adversaries' primary goal, and likewise, many organizations' primary security concern. One simple attempt to do a little DIY DLP (Data Leakage Prevention), or at least detection, would be to monitor for abnormal high volume data being exfiltrated.

Stop and think about high volume connections to the outside world. Could be external client talking into our public facing servers and pulling lots of data? Is this typical? Does the volume of data being transferred make sense for the application? Effectively, these questions are trying to get you thinking about thresholds and clipping levels.

Another possible high volume communication could involve an internal client downloading lots of data (VM Images, streaming movies, etc.), but that would present as inbound high volume transfer not outbound. High volume inbound initiated from internal clients could be an AUP issue, but not especially likely to be malicious.

A third possibility involves a client initiating communication with an external systems and sending a large volume of data. Given the number of users in the modern enterprise, this has likely happened in an innocuous fashion as well as a malicious one. It could well be a successful client-side attack followed by a successful pivoted compromise of internal systems and subsequent exfiltration.

There is no Easy Button on advanced monitoring. The high volume detect can be a successful one, but it can also have you chase your tail for a while trying to figure out what if anything explains the volume. Clipping levels and determining baseline volume can make this a much more successful process.

Eric Cole (@drericcole) has a quick entry on detecting APT in which he discusses both a focus on outbound detection as well as on clipping levels[1].

[1] http://cyber-defense.sans.org/blog/2012/10/23/advanced-persistent-threat-apt-and-insider-threat (http://cyber.gd/511_216) **QR**

Unexpected Destinations

- Where do your outbound connections terminate?
  - Most likely to Alexa Top 500
- What transport protocol and port are employed for most connections?
  - Most likely TCP/80 and TCP/443
- So where does everything else go and how does it get there?

"Icons of the Web[1]"

Continuous Monitoring and Security Operations 38

Unexpected Destinations

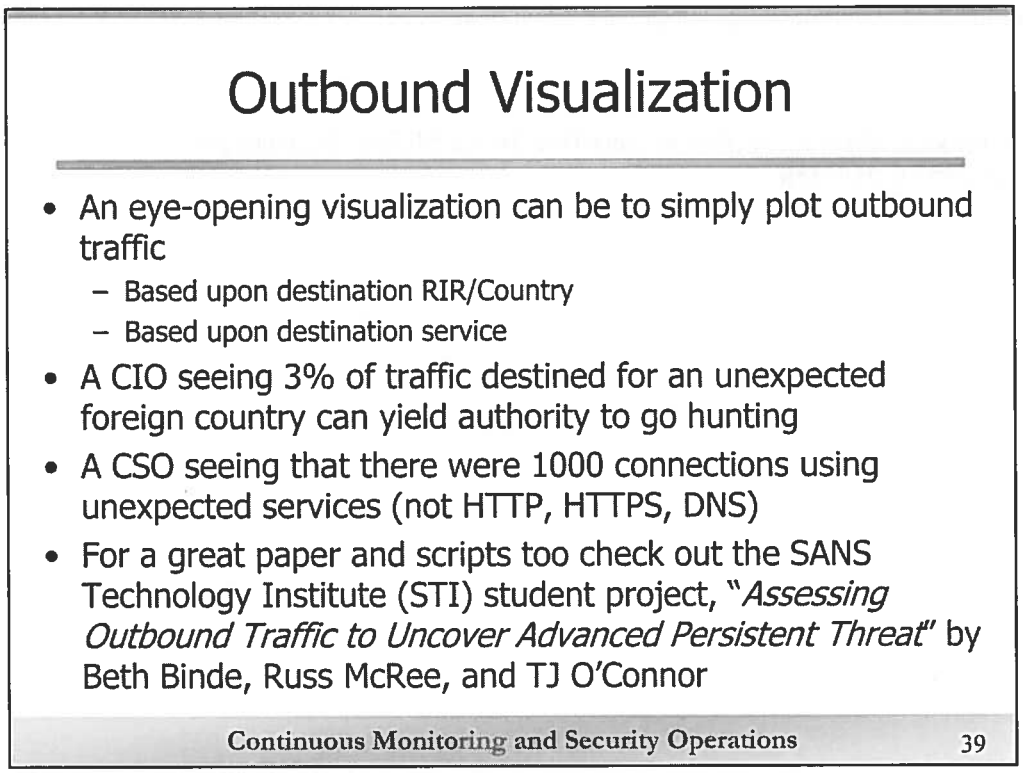
Where does traffic go when it leaves your network? Though you likely have some particular destinations that your users are more likely to hit due to your company, industry, etc., the likelihood is that a significant chunk of your traffic goes where the rest of the world's traffic typically goes.

The Alexa Top 500[2] represents the 500 most commonly hit sites based on the volume of traffic. While there will be sites your users frequent outside of these, they will likely be somewhat predictable.

For a fun and different way of consuming the list of top sites check out the seriously cool "Icons of the Web" project by Gordon "Fyodor" Lyon (@nmap)[1].

[1] http://nmap.org/favicon/ (http://cyber.gd/511_218) **QR**

[2] http://www.alexa.com/topsites (http://cyber.gd/511_217)



Outbound Visualization

One approach that I have seen used to significant affect is plotting/visualizing the outbound. This can be for show, but this can also be useful for analysis.

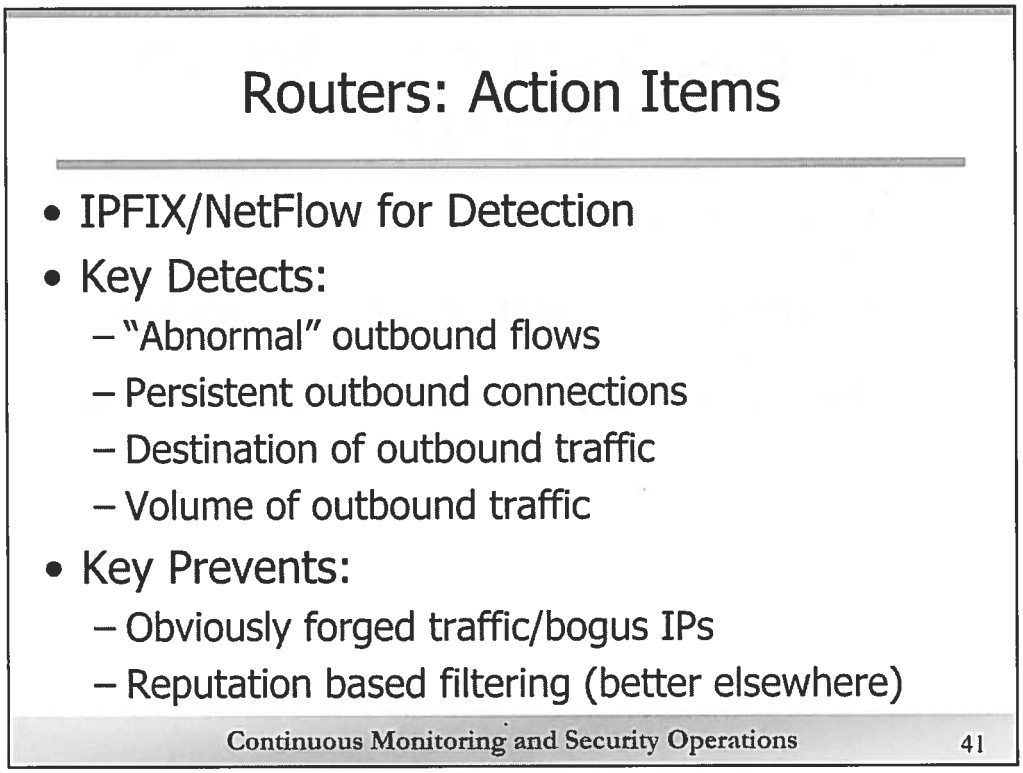
Some quick visualizations include plotting on a map the physical location of the "other end" of communications with the outside world. This is fairly straightforward and might not yield much pay dirt, but it can be a head scratching moment when you visually see that a relevant percentage of traffic goes to a foreign country where you have not clients/business partners. I have seen this exact visualization used to convince an organization that more monitoring capabilities were required. CIO asks the obvious questions: "Why does that much traffic go to $foreign_country?" and "What was actually sent to $foreign_country?" The analysts then indicated that they didn't have any additional details, but could gather those details with approval for additional monitoring capabilities. Oh, I see what they did there… ;)

Another quick and easy visualization would be to graph outbound connections based on the destination service ports. The overwhelming majority will typically be HTTP, HTTPS, and DNS. Are there others? If so, what are they? I have seen this visualization used when trying to get approval to move an organization, that was otherwise forward thinking on security, to a more restricted egress policy.

Definitely check out the SANS Technology Institute (STI) research paper from Beth Binde, Russ McRee, and TJ O'Connor, "Assessing Outbound Traffic to Uncover Advanced Persistent Threat."

One technique, and provided script, employs Python to analyze activity (in the form of a PCAP) by GeoIP[1].

[1] http://www.sans.edu/student-files/projects/JWP-Binde-McRee-OConnor.pdf (http://cyber.gd/511_219) **QR**

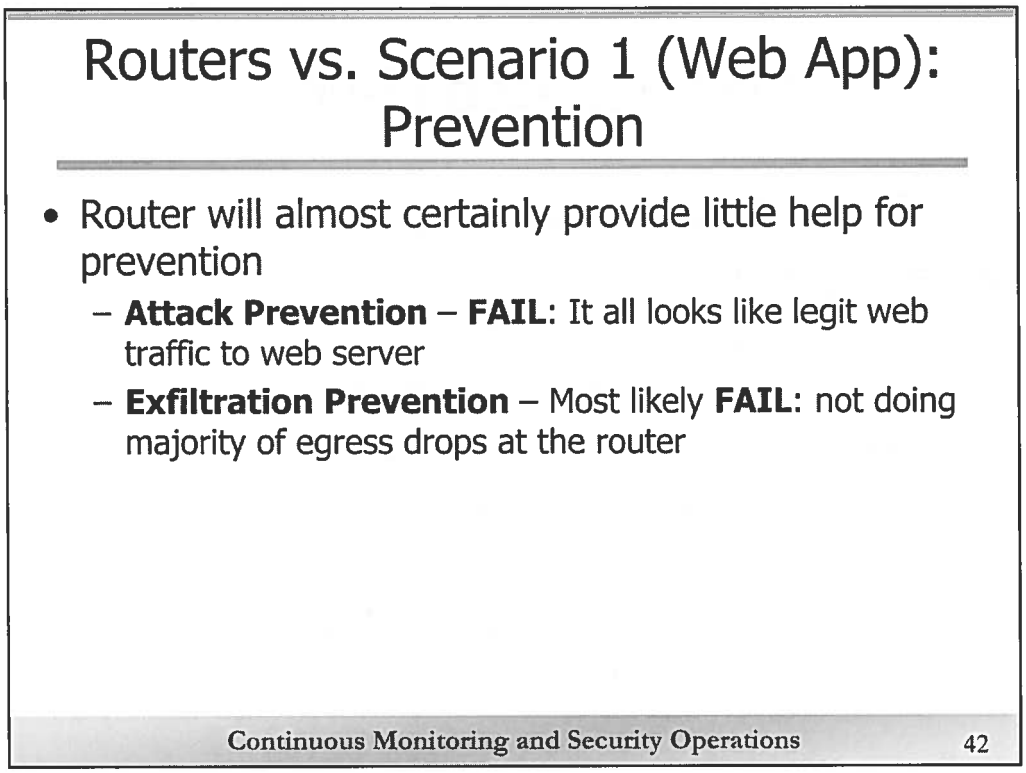


Routers: Action Items

Based on the data provided in this section, as well as a pointer to additional information, we have some potential action items related to routers that can beneficial to modern cyber defense.

On the detection front, the router is suitably positioned to help provide insight into our outbound traffic. Specifically, we recommend looking for "abnormal" connections (see previous slide for understanding abnormal). Also look at persistent outbound connections, the destination IP and service of outbound traffic, and also the volume of the traffic.

From a prevention standpoint, the router can do some very basic filtering such as blocking obviously forged packets, but more advanced prevents should likely be performed elsewhere.

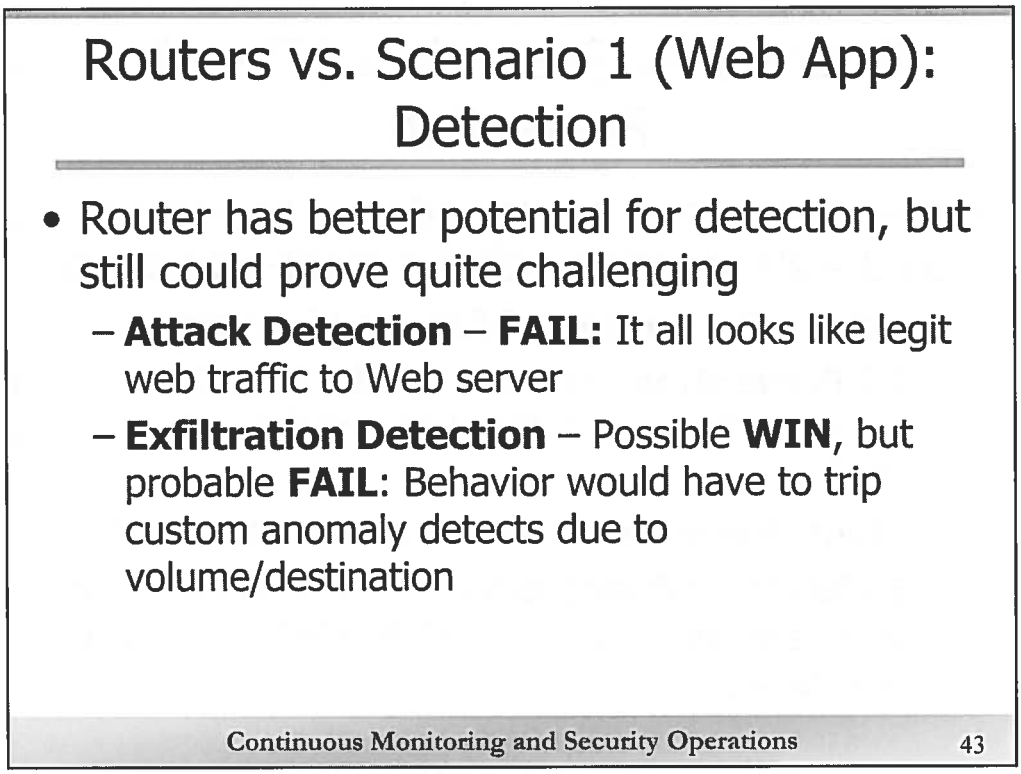


Routers vs. Scenario 1 (Web App): Prevention

Prevention in general is not, and should not be, the routers strong suit. The device is not intended to be doing much in the way of filtering.

For scenario 1, the web application campaign, the router will certainly fail on the prevention of the attack itself. The attack, from the router's perspective, will simply look like regular port 80 traffic.

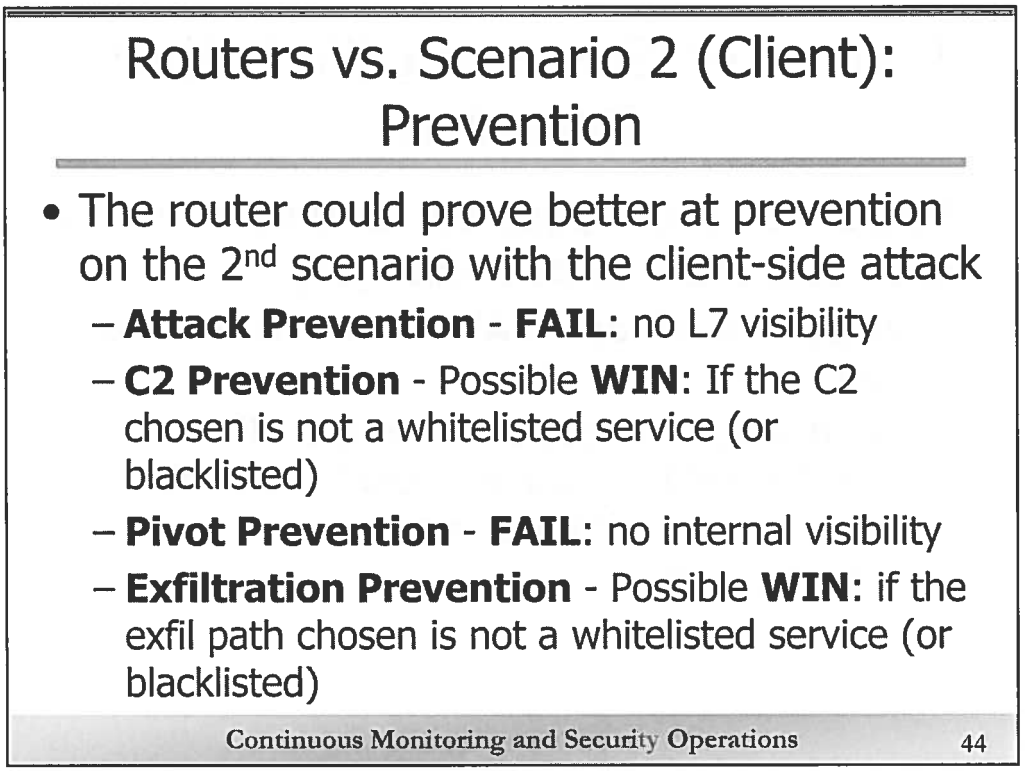
Preventing exfiltration too will provide difficult. Outbound blocking is unlikely to occur on the router, and certainly not to the extent that blocking return traffic from a web application interaction would be possible.



Routers vs. Scenario 1 (Web App): Detection

How does the router perform on the detection front for our web application campaign? Not much better than on the prevention front. Detection of the attack would be extremely unlikely as again it does not, and should not, be looking into Layer 7 data.

Detecting the exfiltration would also likely be unsuccessful. The only way that this could be detected would be if custom anomaly detects were instrumented based on the volume or destination of the data. These detects would really come from a separate process that was specifically looking at the router's log data.



Routers vs. Scenario 2 (Client): Prevention

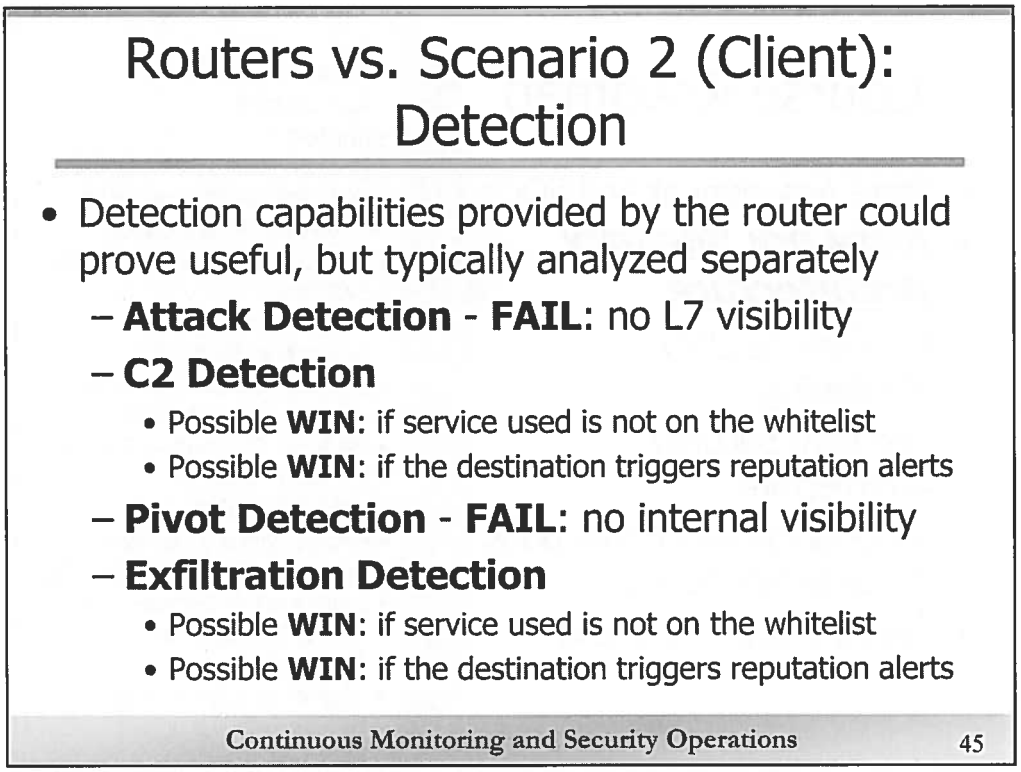
Let's see how the router can stack up against the client-side attack from the prevention standpoint.

The router will be unable to prevent the attack, as the attack was in Layer 7 in an allowed communication path (response to allowed outbound communication).

For the C2, command and control, the router might be able to block the traffic if it leveraged a service that is not explicitly whitelisted. This assumes that the organization has a strong security posture on their egress.

The router is wholly unhelpful regarding the pivot, as it is not suitably positioned to even see the traffic.

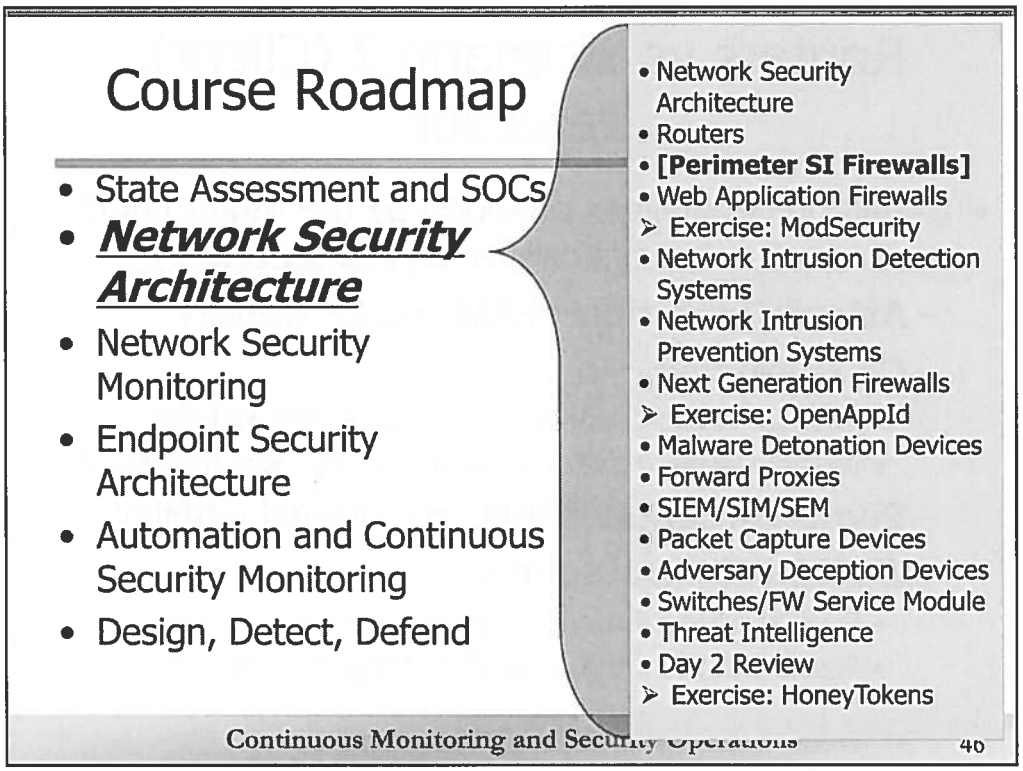
On the exfiltration front, we again have the same scenario as described for the C2. The router could possibly prevent the data if the communication path chosen by the adversary is not on the whitelist.



Routers vs. Scenario 2 (Client): Detection

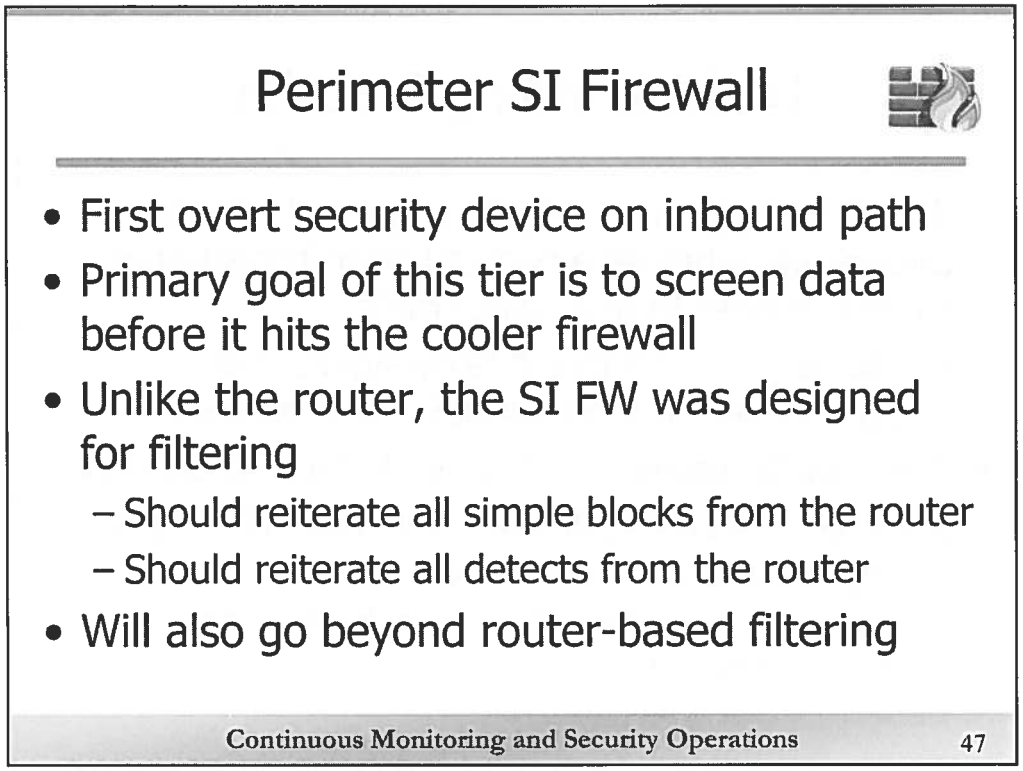
Detecting the client-side attack with the router feels very similar to the prevention discussion. The attack and pivot will be entirely lost on the router due to lack of Layer 7 and Internal visibility.

On the C2, command and control, and exfiltration front, the potential for detection would be due to either the adversary employing services not on the whitelist or perhaps sending the data to locations with a poor IP reputation.



Course Roadmap

Now let's take a look at Perimeter SI Firewalls.

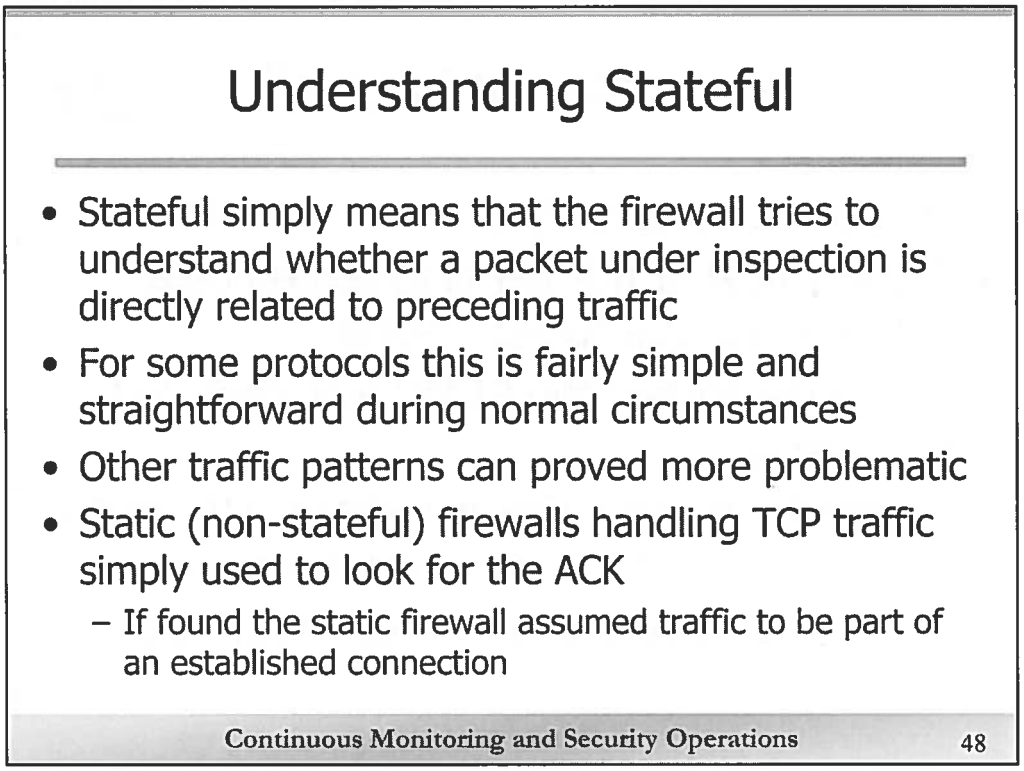


Perimeter SI Firewall

Though the router can prove helpful, primarily due to its location, the router is not an overt security product. The perimeter Stateful Inspection (SI) firewall is likely the first security tool to be encountered on the ingress and the last security tool to be seen for egress.

The primary focus of the perimeter SI firewall in the modern enterprise is to provide somewhat basic, but fast security screening. Even though we now have much more advanced firewalling capabilities, the increased features come at a price in terms of speed. Also, the cooler features also imply increased complexity, and therein vulnerability.

The perimeter SI firewall will also reiterate all prevention and detection capabilities afforded us by the router. However, it should be able to go beyond the most basic of filters employed by the router as this device actually operates as an intentional filter.

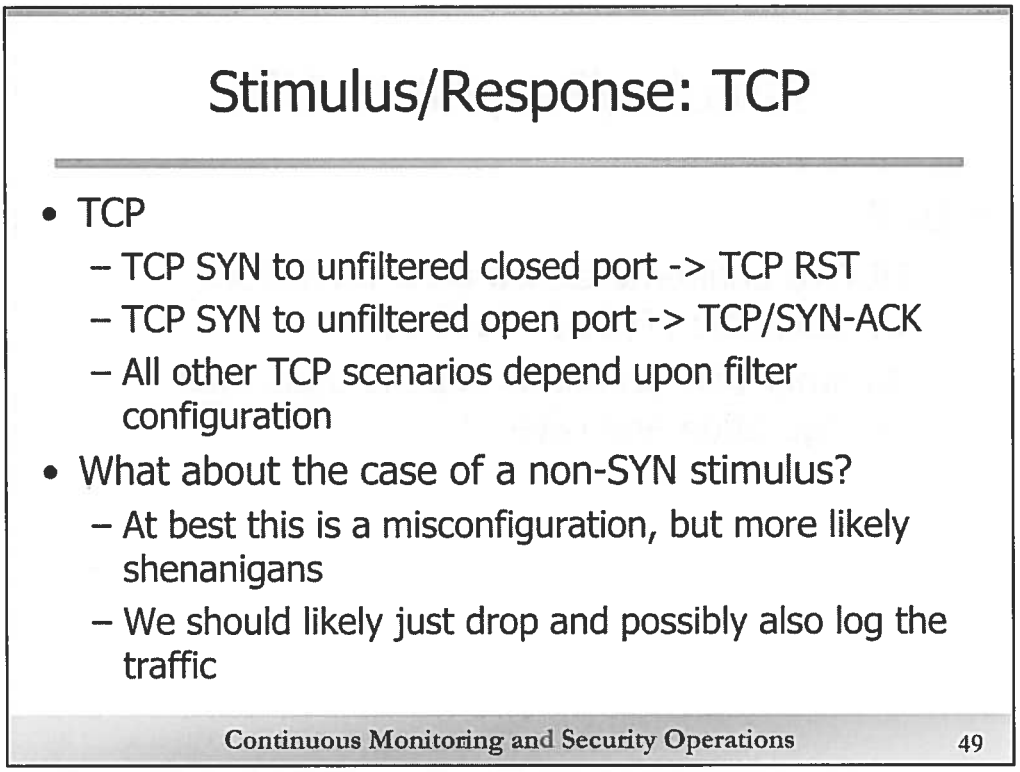


Understanding Stateful

So, what exactly does the S(tateful) in SI firewall mean. The term stateful is used to contrast this device with the older static firewalls. Static firewalls, also known as static packet filter firewalls made decisions about the final disposition of traffic based upon individual packets without any context. This poses a problem for building a comprehensive firewall rulebase.

Imagine a scenario where a client is initiating outbound HTTP traffic to http://www.google.com. The static packet filter and stateful inspection firewall both handle the initial outbound stimulus easily. Outbound traffic (TCP: SYN) destined for TCP/80 is allowed. In the case of the SI firewall an entry to the state table is made that corresponds to the initial traffic. When Google responds (TCP: SYN/ACK) the SI firewall sees that there is a corresponding entry in the state table and allows the traffic. The static packet filter has no state table and must decided based simply on this one SYN/ACK packet whether to allow or deny the traffic. One approach could be to allow all traffic sourced from TCP 80, assuming it to be legit response from a Web Server. Another, better, approach would be to look for the ACK flag and presume that this must be response traffic.

Merely looking for the ACK flag and allowing any communication is less than ideal, and TCP is actually the easiest to handle scenario, ICMP and UDP prove much more challenging.



Stimulus/Response: TCP

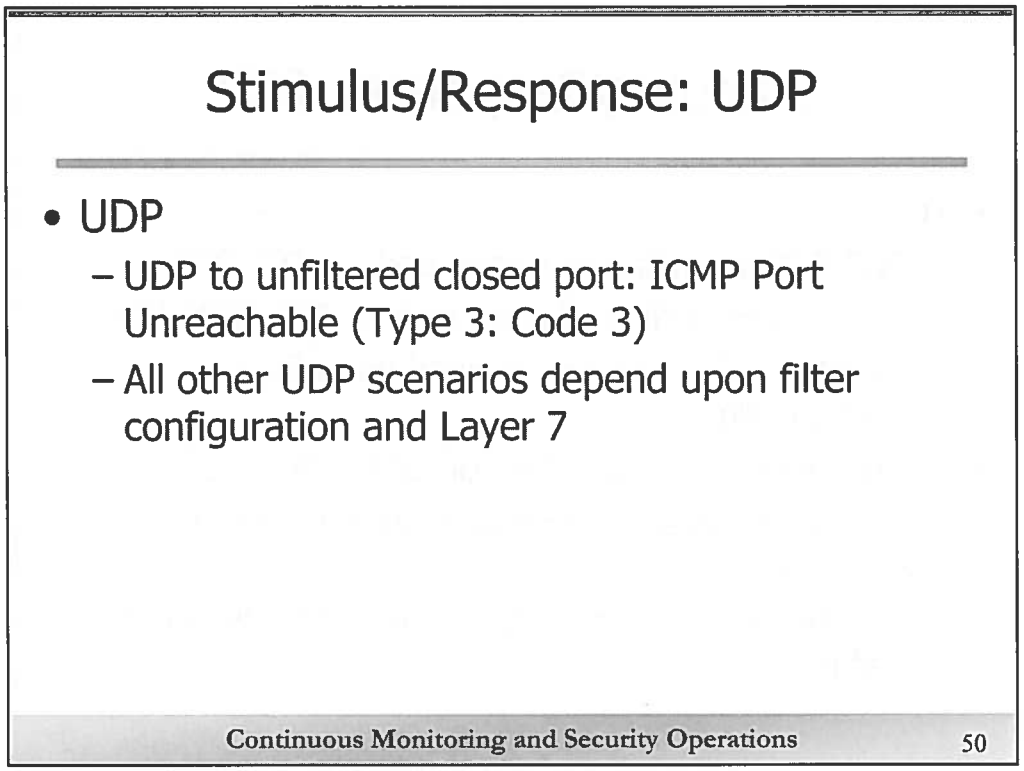
To understand state, ingress, and egress we need a quick review of how stimulus and response works for TCP, UDP, and some ICMP.

The TCP 3-way handshake involves a three packet exchange, (1) SYN; (2) SYN/ACK; (3) ACK, before any payload will typically be sent. Let's see how this can play out.

TCP SYN sent to an unfiltered closed port would result in a TCP RST packet response.

TCP SYN sent to an unfiltered open port would result in a TCP/SYN-ACK, the second leg of the 3-way handshake.

What happens when a SYN is sent to a filtered port? RFC suggests that a RST would be sent, but we don't necessarily care for our filters to be full stop RFC compliant in this case. Sending a response means divulging some, even if it is fairly limited and seemingly innocuous, information. Further, it will greatly increase an adversary's ability to rapidly scan our systems. Personally, I am not, intentionally at least, in the business of aiding adversaries.



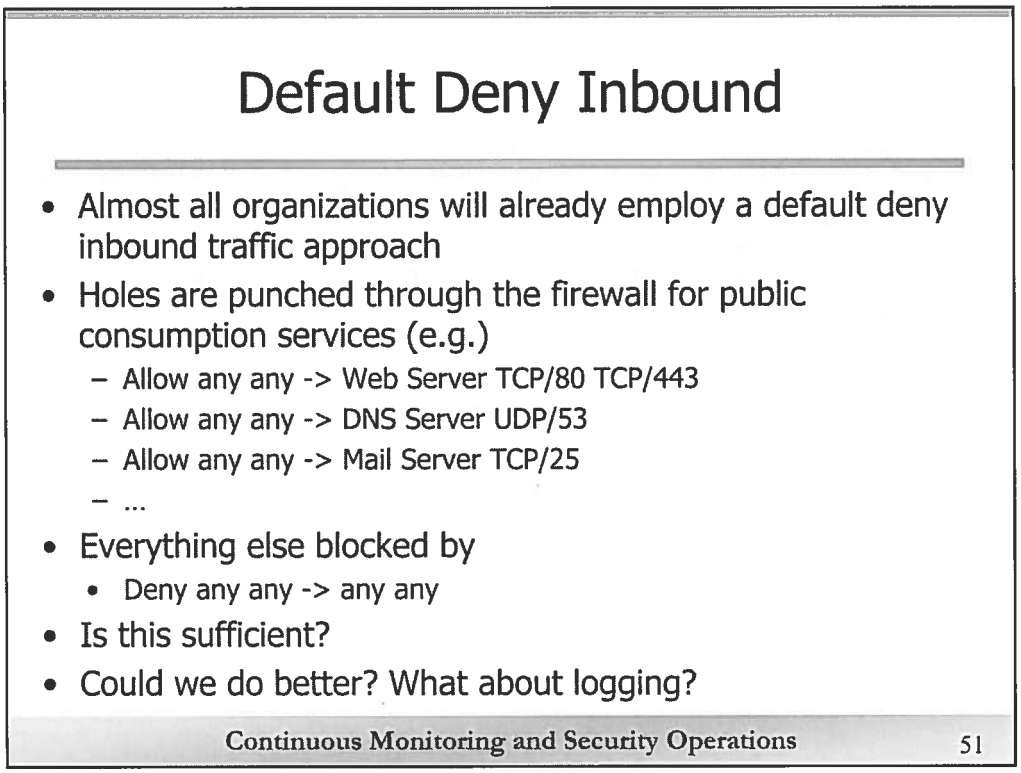
Stimulus/Response: UDP

UDP is a bit more challenging. The nature of UDP is to be first and foremost fast and simple. This emphasis on speed and simplicity means that UDP does not have native ways of handling unexpected stimulus. Also, even for expected stimuli, UDP does not require acknowledgement.

In the case of UDP stimulus being sent to an unfiltered closed port, the expected response is an ICMP Port Unreachable (Type 3: Code 3) message.

All other UDP stimulus/response is going to depending upon the configuration of the filter and the Layer 7 service being communicated with.

Static firewalls, and even some stateful firewalls have difficulties handling UDP due to its lack of being oriented for sessions.



Default Deny Inbound

Most organizations already employ a default deny rule for inbound traffic that is not explicitly allowed.

We create holes for any specific service that requires externally sourced communication. For example:

```
allow any any -> Web Server TCP/80 TCP/443
allow any any -> DNS Server UDP/53
allow any any -> Mail Server TCP/25
...
```

There is typically an implied **`deny any any -> any any`** at the bottom of the rulebase, so that anything not allowed before hitting the end gets blocked.

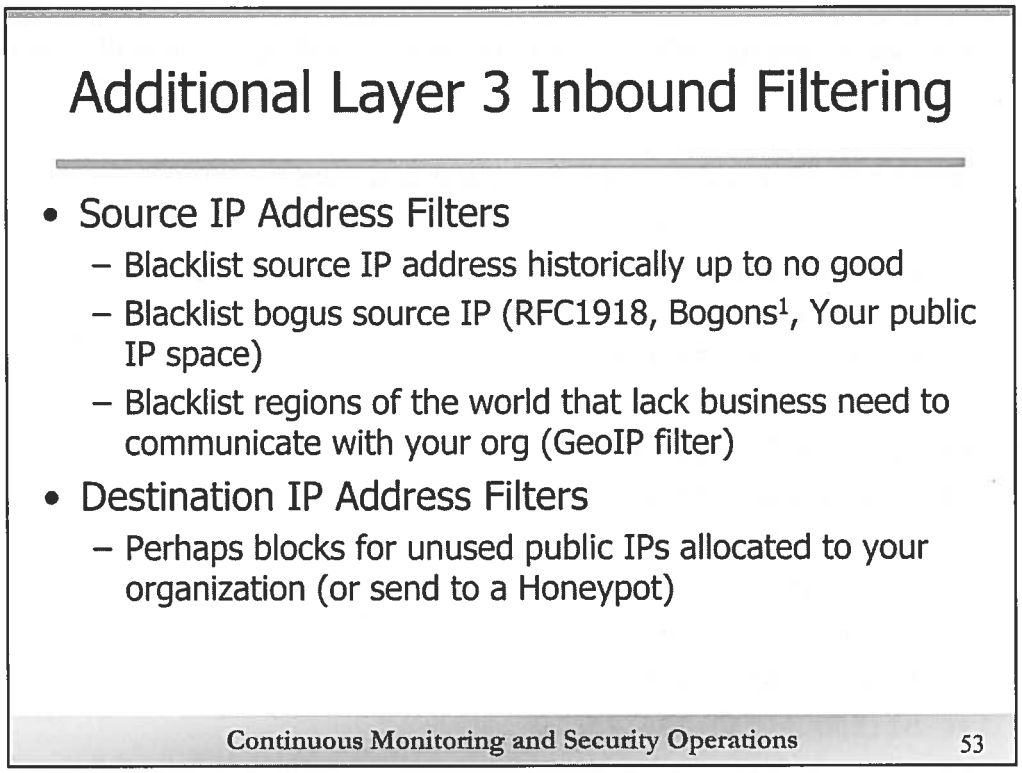
This seems to work fairly well, but can we improve upon it?

From a performance perspective, if you have a significant volume of traffic that has to be evaluated by a large rulebase before ultimately getting dropped, then it might be worthwhile to put an explicit block

above the allow rules. However, general performance tuning is not our primary concern. We want to achieve a more robust security posture.

One thing that we need to consider is the logging capabilities of the particular firewall. Do we get per rule logging options, like with iptables, or do we get packet logging regardless of the rule matched? There could be traffic that we do not really care to have logged as it is so tremendously high volume and we think the likelihood of abuse is sufficiently low. In these circumstances we might look into splitting out the high volume traffic to be blocked or allowed without any logging, again assuming per rule logging is an option.

Regardless of logging, we do have some additional filtering potential.



Additional Layer 3 Inbound Filtering

Beyond the implicit deny and the particular allowances, we could bolster the rulebase with some additional prevention/detection. Do you really want every system/IP in the universe to be able to talk to your website? Probably not, but you want all potential legitimate customers, clients, etc. to be able to interact with our public systems.

The trick is, how can we safely differentiate folks hitting our public consumption services for good from those hitting it for evil. Well, for a start if they are presenting with a known RFC1918, Bogon[2], or your own address space, then they are unlikely to be legitimate.

For some organizations it makes sense to perform geographical blocking, which is blocking based on the region or country the traffic is sourced from. Typically this is achieved with a GeoIP lookup database, like the ones available from MaxMind[3] (some of which, like GeoLite2[4] databases are free.)

While strange years back to consider blocking off chunks of the world, many of us, especially those that travel throughout the world, are not even a little surprised by this. Numerous streaming services are limited based upon country of origin. Note also, that GeoIP blocking can be very easily bypassed by even a moderately sophisticated adversary (e.g. tunneling traffic through a free Linux AWS Micro Server).

However, just because some can bypass the filter does not negate its value.

Naturally, with any sort of blacklist/blocklist, you need to be mindful that the data can change over time. Also understand that you run definitely run the risk of some blocking of potentially legitimate traffic.

Here is the Team Cymru dotted decimal bogon list (current as of December 2014):

- `0.0.0.0 255.0.0.0`
- `10.0.0.0 255.0.0.0`
- `100.64.0.0 255.192.0.0`
- `127.0.0.0 255.0.0.0`
- `169.254.0.0 255.255.0.0`
- `172.16.0.0 255.240.0.0`
- `192.0.0.0 255.255.255.0`
- `192.0.2.0 255.255.255.0`
- `192.168.0.0 255.255.0.0`
- `198.18.0.0 255.254.0.0`
- `198.51.100.0 255.255.255.0`
- `203.0.113.0 255.255.255.0`
- `224.0.0.0 240.0.0.0`
- `240.0.0.0 240.0.0.0`[1]

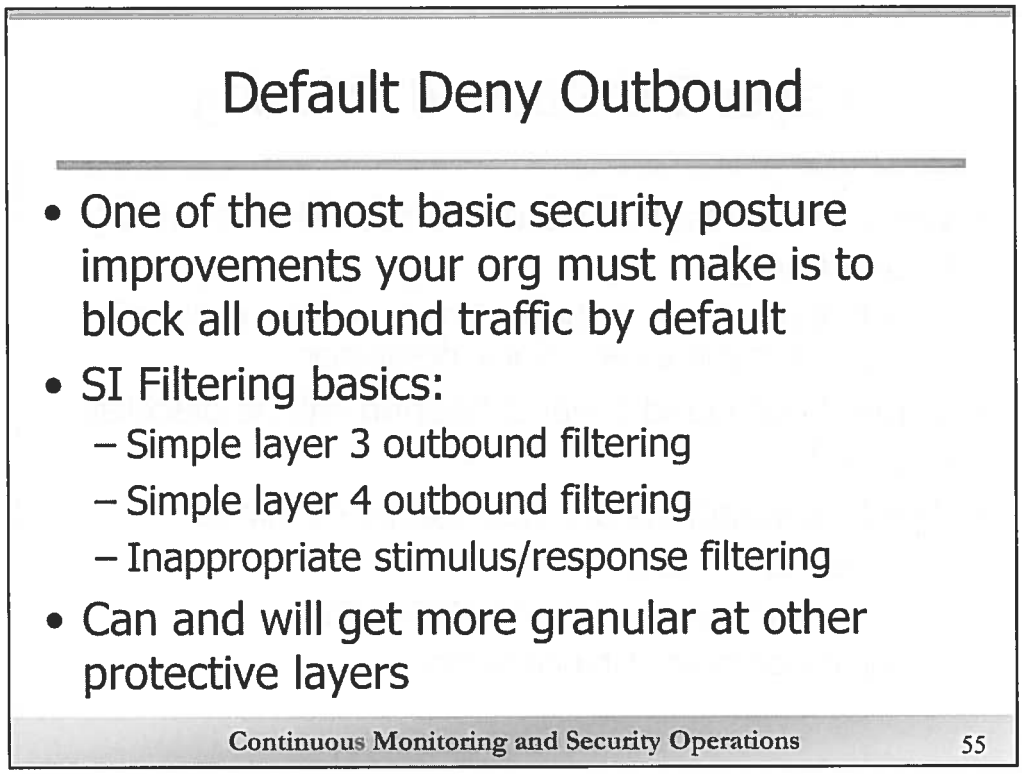
These source addresses should be dropped by the external interface of your external router or firewall. Also consider adding your internal IP addresses to this list (if they are not already listed, such as RFC1918 addresses), to prevent inbound spoofing.

[1] https://www.team-cymru.org/Services/Bogons/ (http://cyber.gd/511_220) **QR**

[2] Ibid.

[3] http://www.maxmind.com/en/home (http://cyber.gd/511_221)

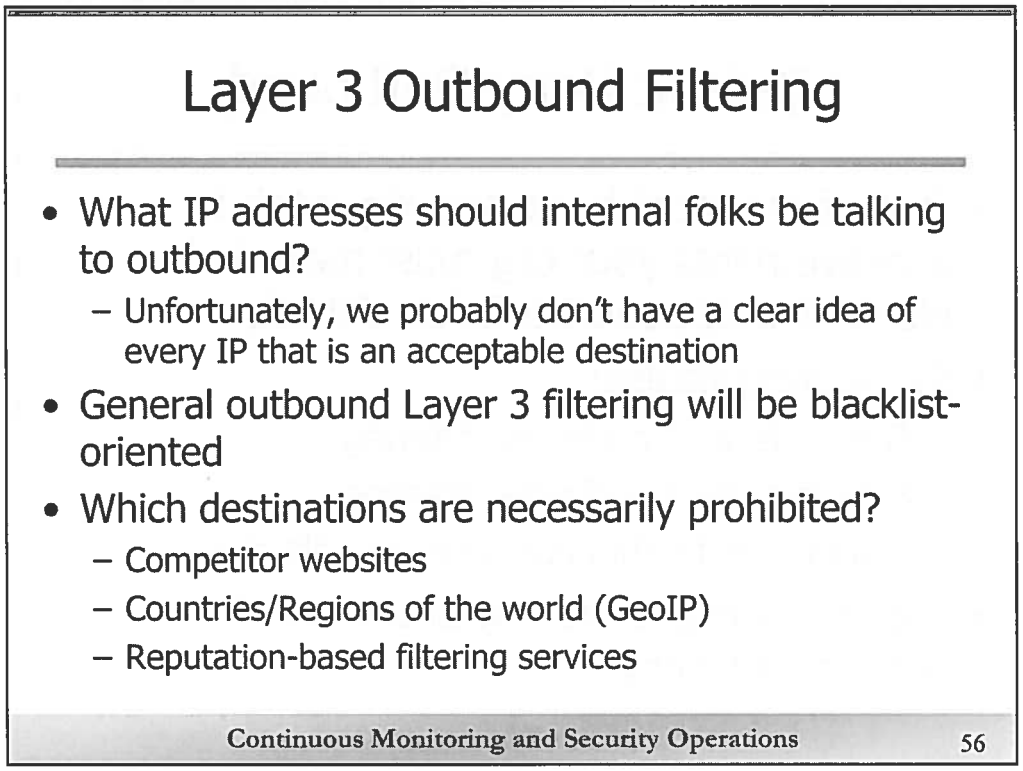
[4] http://dev.maxmind.com/geoip/geoip2/geolite2/ (http://cyber.gd/511_222)



Default Deny Outbound

The majority of organizations will employ a default block for all traffic originating form the outside. Then they punch specific holes for services intended for public consumption and other particular needs. Why do we not find that to be true also about outbound filtering? In the overwhelming majority of organizations the default outbound/egress policy is to allow that which is not explicitly denied.

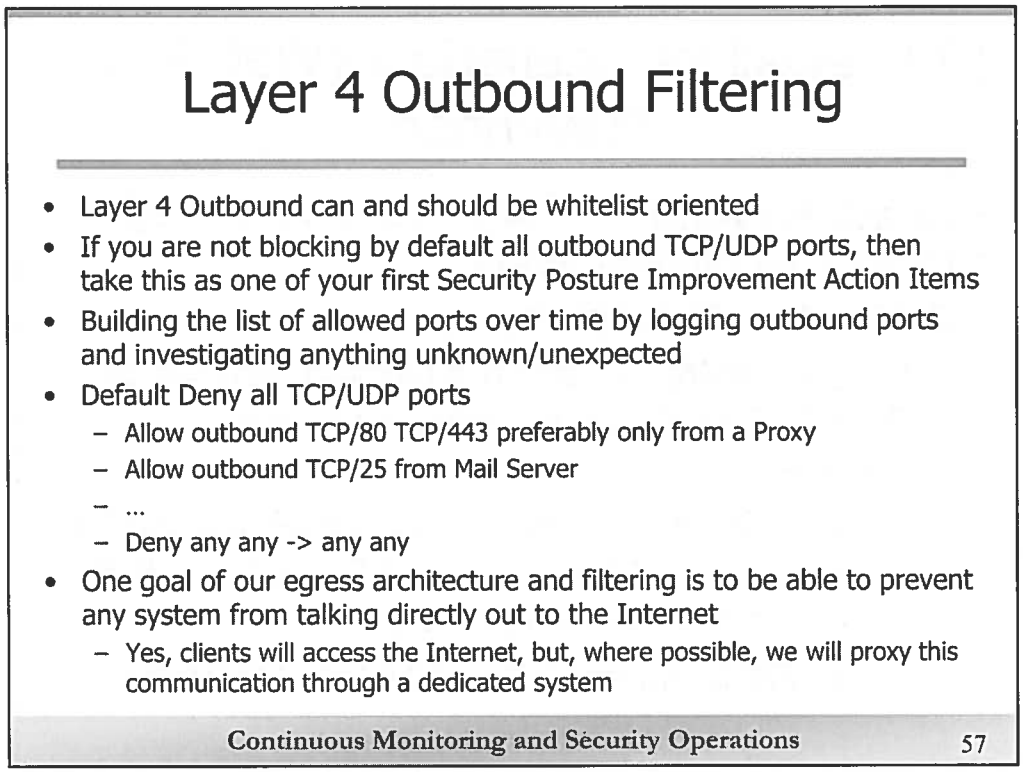
One of the most important security posture changes you can effect is to get your organization to a default deny outbound configuration.



Layer 3 Outbound Filtering

For the inbound firewall rulebase, we specified exactly the IP addresses that would be involved in a conversation. Unfortunately, it is unlikely that you will be able to build the same style of whitelist for outbound traffic. Could you enumerate all of the particular destination IP addresses you would like your folks to be able to reach? Didn't think so.

However, we do not have to give up on Layer 3 outbound filtering. We can still employ filtering, but it will be a blacklist rather than a whitelist. Not really talking here about individual IP addresses here. The most likely scenario would GeoIP-based or reputation based filtering.



Layer 4 Outbound Filtering

While we were only able to pull off a blacklist for our Layer 3 outbound filter, we should be able to pull off a whitelist for our Layer 4 outbound filter.

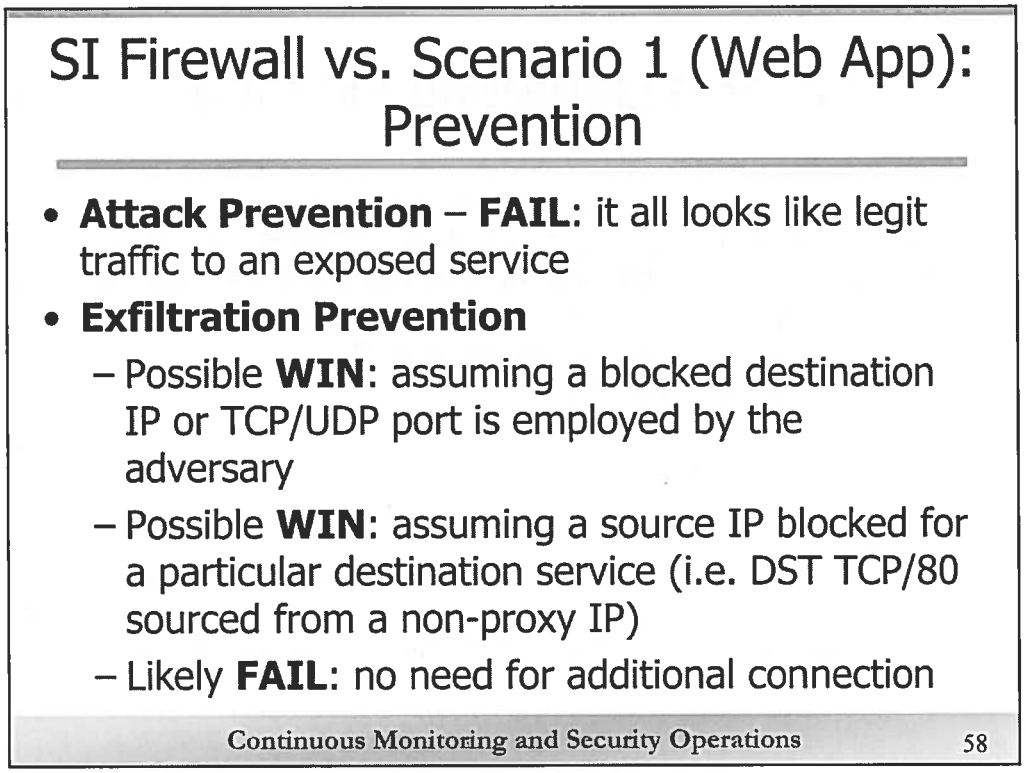
This is the big win for outbound filtering, and should easily be one of the first security posture improvements for your security architecture. What services/ports do internal folks need to access?

TCP/80 - from Proxy
TCP/443 - from Proxy
UDP/53 - from DNS Servers
TCP/25 - from Mail Servers
UDP/123 - from NTP Servers

Note that, by design, desktops/servers cannot talk directly out to the Internet. While this might not be achievable, it serves as a strong goal for us.

Dennis Distler, GSE #39 wrote a GIAC Gold paper on Egress Filtering in 2008 that is still relevant and worth a look[1].

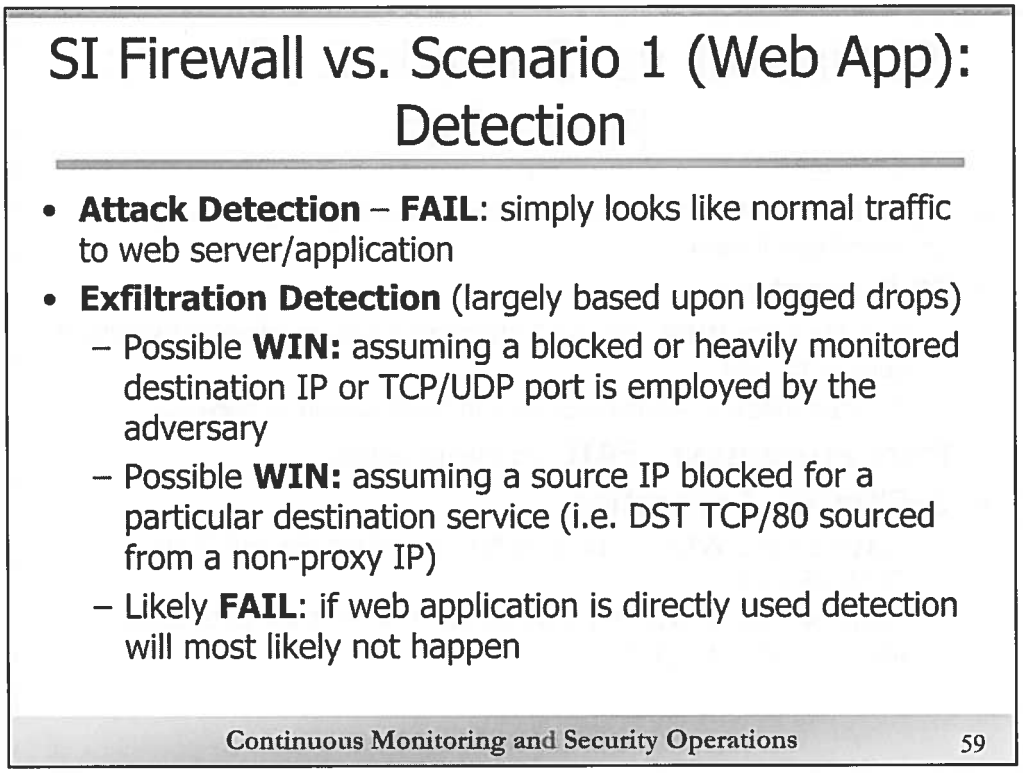
[1] http://www.sans.org/reading-room/whitepapers/firewalls/performing-egress-filtering-32878 (http://cyber.gd/511_223) **QR**



SI Firewall vs. Scenario 1 (Web App): Prevention

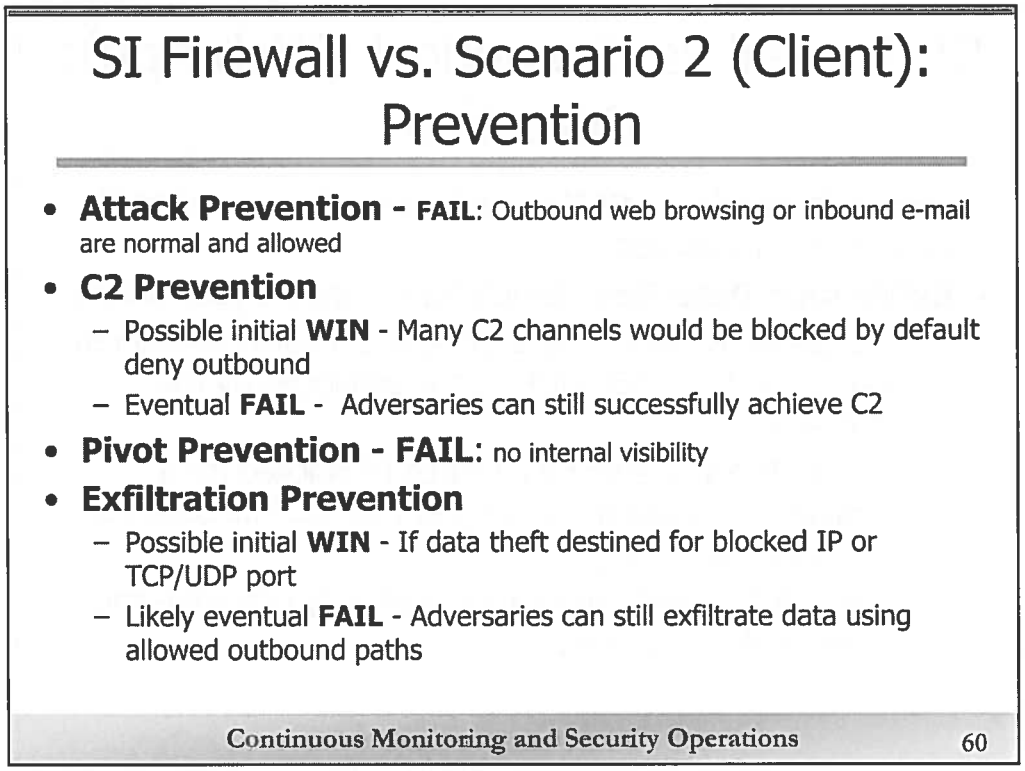
The Perimeter SI Firewall would not be able to prevent the web application attack from succeeding as it would look like normal traffic at Layers 3 and 4.

On the exfiltration front, the SI Firewall could prove successful, but this would only occur if the adversary employed an additional connection for the exfil, which is unlikely given the exfil could likely be just response traffic from the web application.



SI Firewall vs. Scenario 1 (Web App): Detection

Detecting the web application attack with an SI Firewall will be unsuccessful. We might be successful at detecting the data exfiltration if the adversary employs an IP or port that we are blocking. However, with the web application being the source of the data, it is unlikely that an additional IP/port would be employed.

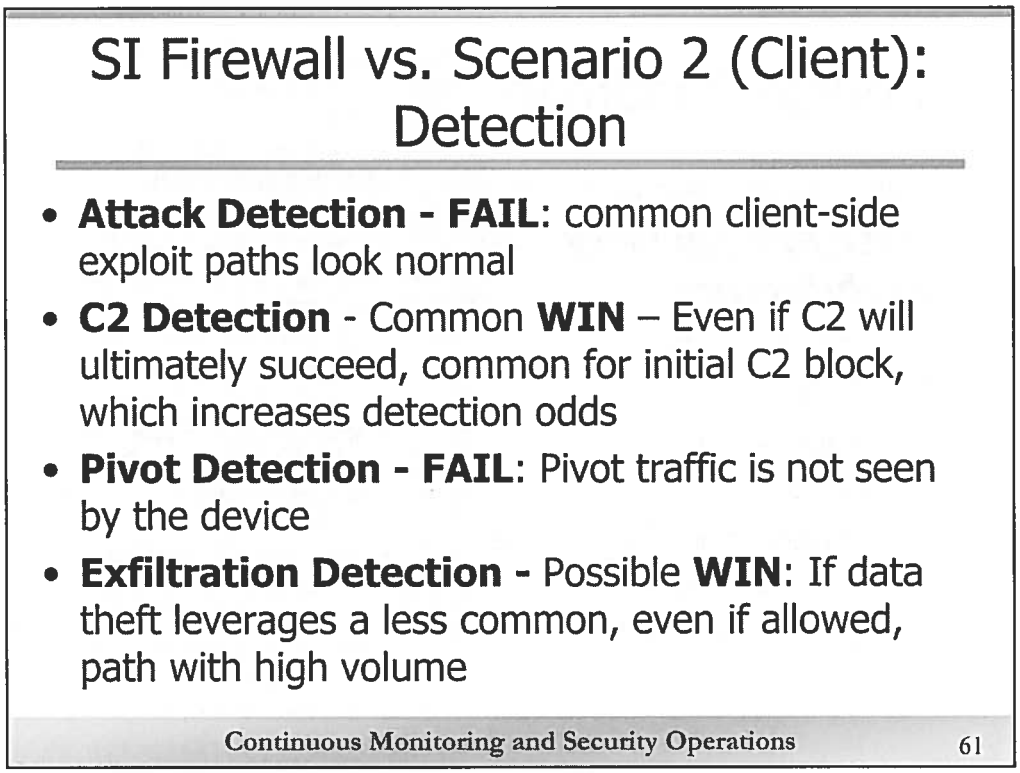


SI Firewall vs. Scenario 2 (Client): Prevention

The SI Firewall will likely perform a bit better against the client-side attack than the web application.

On both the attack and pivot prevention front, the SI Firewall will provide likely no benefit whatsoever.

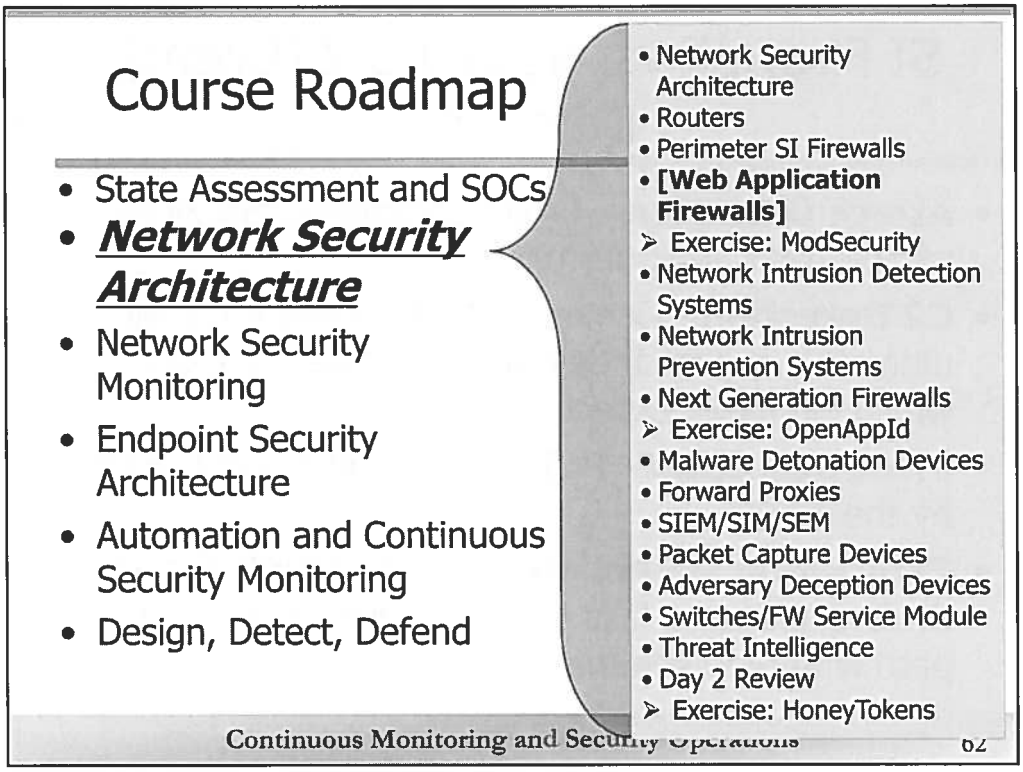
With respect to C2 and Exfiltration prevention, we could possibly achieve an initial bock due to our restricted egress, even though ultimately the adversary could likely prove successful at stealing the data.



SI Firewall vs. Scenario 2 (Client): Detection

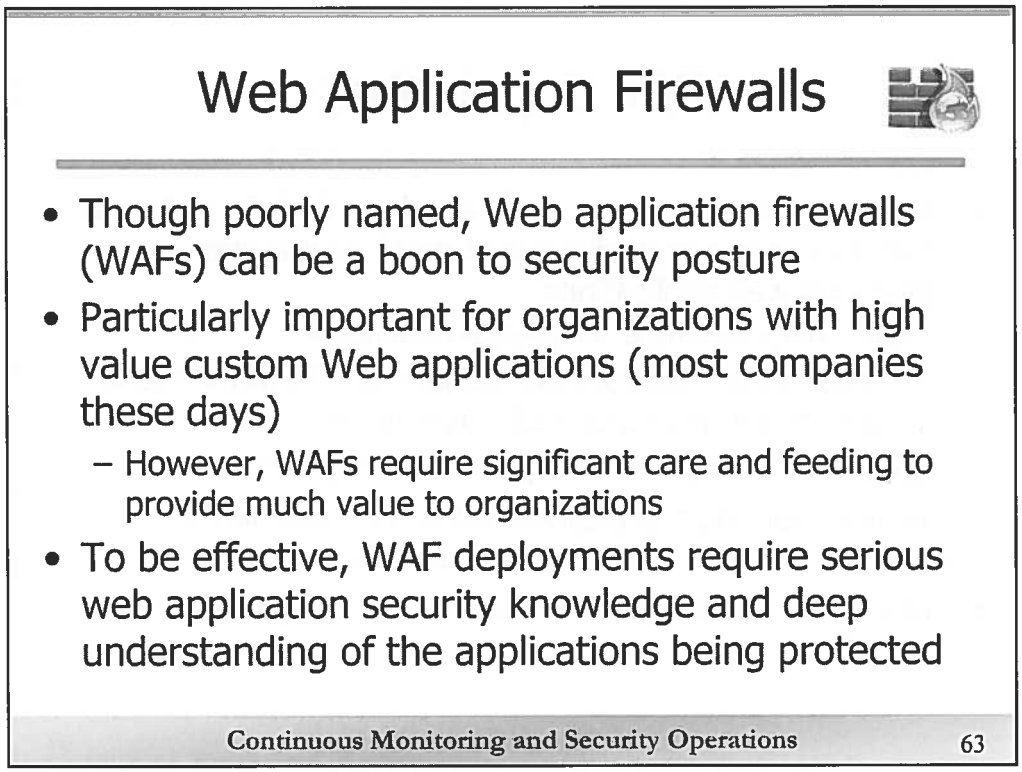
With respect to our potential detection of both the attack and the pivot, we are largely in the same position we were with the preventive capabilities; which is to say not expecting to be successful.

On the C2 and exfiltration detection, we very likely will fare much better. Through on the preventive front we indicated the potential for initial success, but likely a subsequent failure. On the detection front, we might very well catch the adversary making those initially blocked attempts, which provides us time to successfully detect and respond.



Course Roadmap

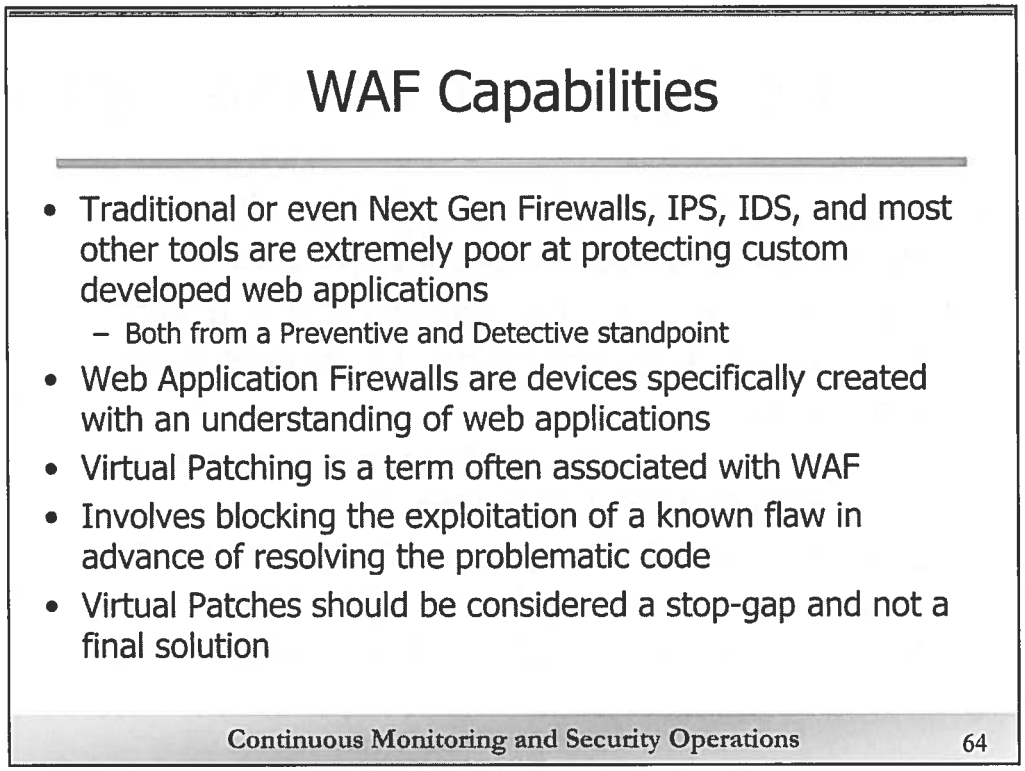
The next section is Web Application Firewalls.



Web Application Firewalls

The name, web application firewall (WAF), can cause many issues and misunderstanding for organizations. With the word firewall in the name, many folks walk away with some misconceptions. First, they expect the device to overtly serve in a preventive capacity. Another, larger issue, is that many people grossly underestimate the effort involved, thinking that, like their traditional firewall, they could simply drop WAF in front of web applications and derive tremendous security value.

WAFs, in order to provide significant security benefit, will require a tremendous amount of effort by someone (or a team) that has not only knowledge of web application security from both the attack and defense sides, but also a significant understanding of the particular web applications.



WAF Capabilities

Assuming the organization appreciates the level of effort involved and staffs accordingly, what could a WAF provide us? Traditional security devices, including NGFW, IDS, IPS, and Malware Detonation Devices, are rather poor when it comes to protecting custom web applications Web application firewalls are built with custom web applications in mind, and, with proper tuning, can be tailored to protect individual custom web applications.

Beyond just generally protecting web applications, WAFs can also provide another benefit that is referred to as virtual patching. Assuming you discover an exploitable flaw in your organization's custom web application, where do they go to get the patch? Oh, wait, there is no patch. The organization must fix their own code.

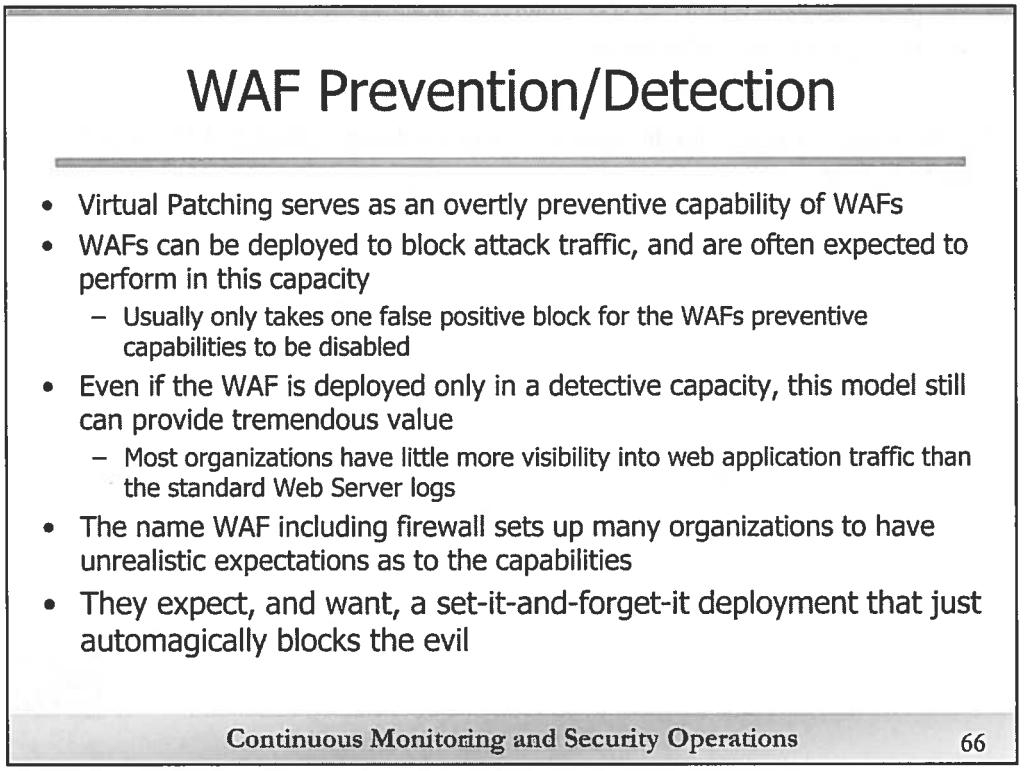
How long does fixing the code take? This can vary greatly, but WhiteHat's Website Security Statistics Report[1] can help shed some light on the issue. In WhiteHat's study, for .NET based web applications the average time to fix a discovered flaw was approximately 112 days[2]. Ouch, assuming this is a publicly exploitable flaw, you effectively have a 0-day vulnerability for 112 days. This flaw could be exploited as there is no patch.

WAFs can potentially help with the issue through Virtual Patching. Virtual Patching is a technique whereby the WAF can be used to attempt to thwart any attempts to exploit the flaw.

This is not a true patch, and the flaw should still be fixed in code, but it can mitigate the risk until such time as the code has been properly addressed.

[1] http://info.whitehatsec.com/rs/whitehatsecurity/images/statsreport2014-20140410.pdf (http://cyber.gd/511_224)

[2] Ibid.

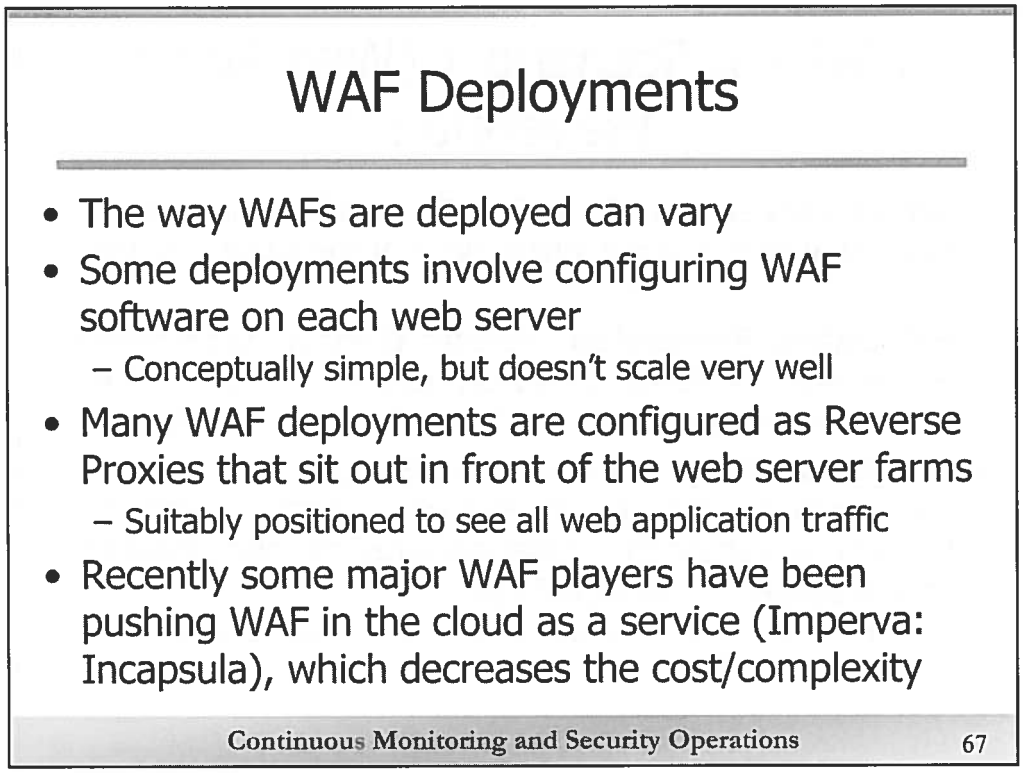


WAF Prevention/Detection

While Virtual Patching provides a primarily preventive capability, WAFs can be, and often are, used to provide significant detective capabilities.

Many organizations do not initially intend for the WAF to be a detective control. However, I have seen a large number of WAFs be employed without sufficiently skilled staff, and had false positives present in the WAF. Blocking a web application that is important enough to employ a WAF tends to get the preventive capabilities of the WAF scaled back considerably.

Often security teams view this as failed deployment. While on some levels I suppose it is, but the WAF can still be hugely beneficial on the detection front. Given the name, people have very mistaken impressions about the WAFs.

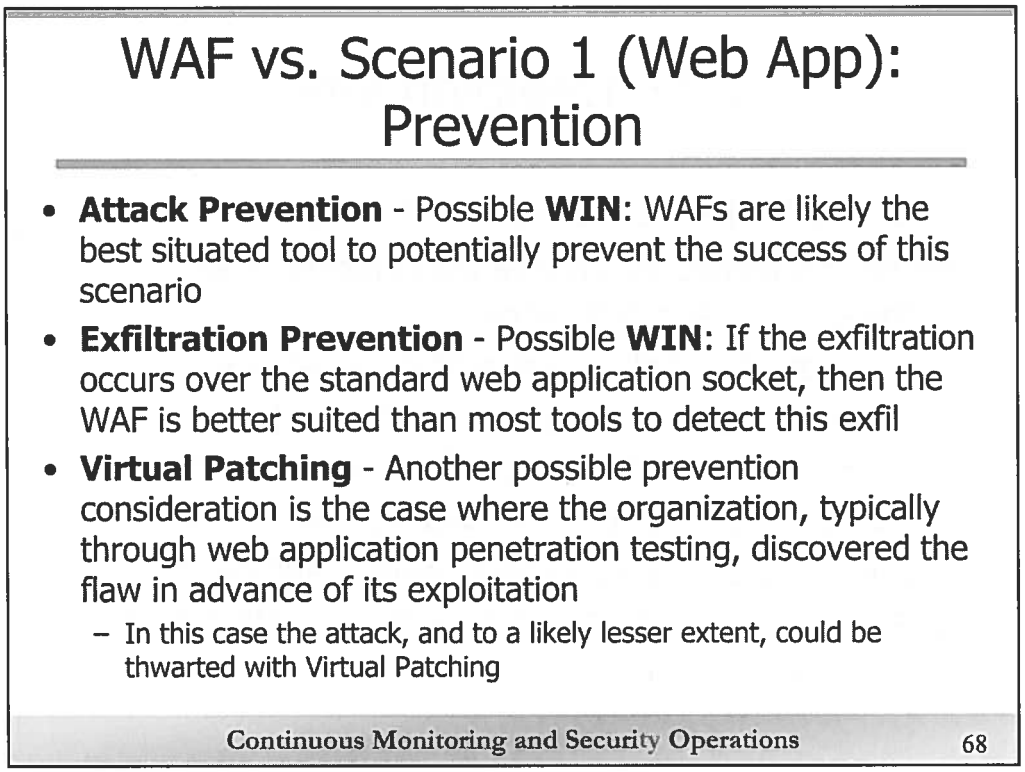


WAF Deployments

Architecturally, where does the WAF live, and how is it deployed? Necessarily the WAF needs to be in front of the web application(s) it is responsible for protecting. A conceptually simple approach is to employ the WAF as a module on the web server itself. While this approach has the benefit of being extremely simple conceptually, it does not scale well without a management infrastructure for the WAFs themselves. So, if you are protecting thousands of servers, then this might not be the best deployment model.

Beyond just general scale concerns, the module-based WAF deployment also has a weakness when it comes to web server farms where many, ostensibly identical, servers exist for load balancing purposes. In those cases, and in many others, one of the best deployment approaches could be as a reverse proxy that sits inline out in front of the web server farm. It should be said that the major load balancing appliances often can be extended to provide web application firewalling capabilities.

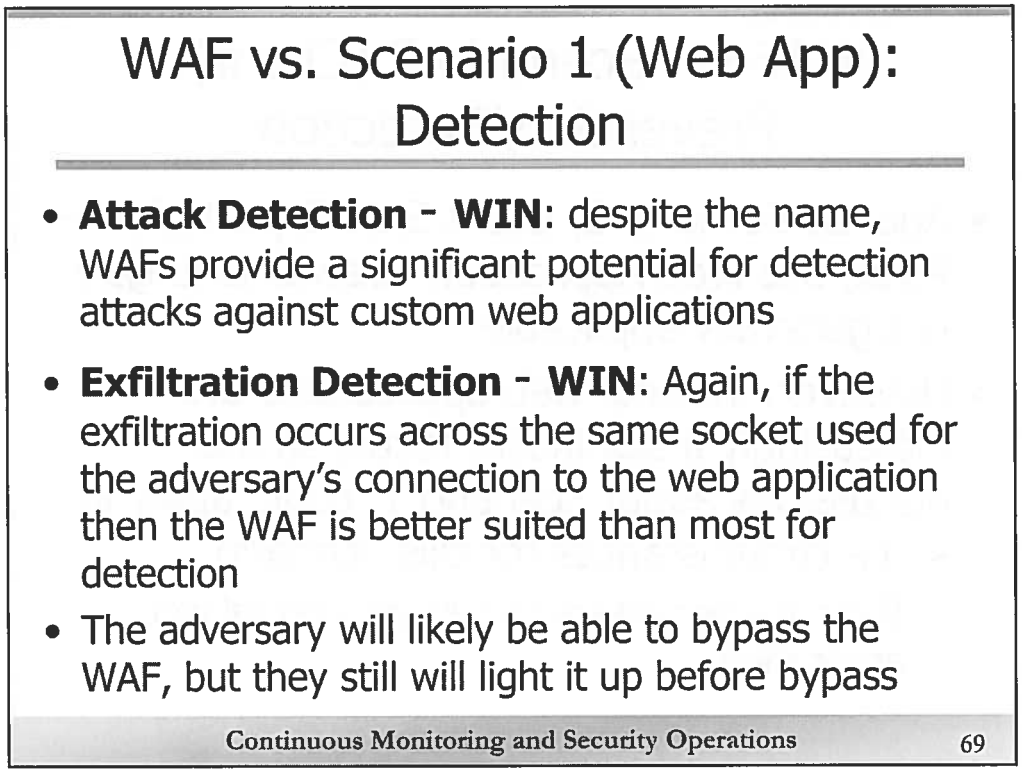
A final deployment model which has begun recently to be pushed by vendors is the WAF-in-the-cloud model. Effectively, much like the spam/mail filtering-as-a-service approach that is popular with many enterprises, the WAF would be in the cloud and your web application communications would go through the cloud. This would tend not to require any on premise device, or device management. Often there are also services that can be provided whereby you are effectively outsourcing a chunk of web application security capabilities to the vendor.



WAF vs. Scenario 1 (Web App): Prevention

For the web application scenario, on the prevention front, the WAF could possibly assist with the attack prevention and exfiltration prevention. While this doesn't sound like the high praise and high hopes that many organizations have for WAFs, it is realistic.

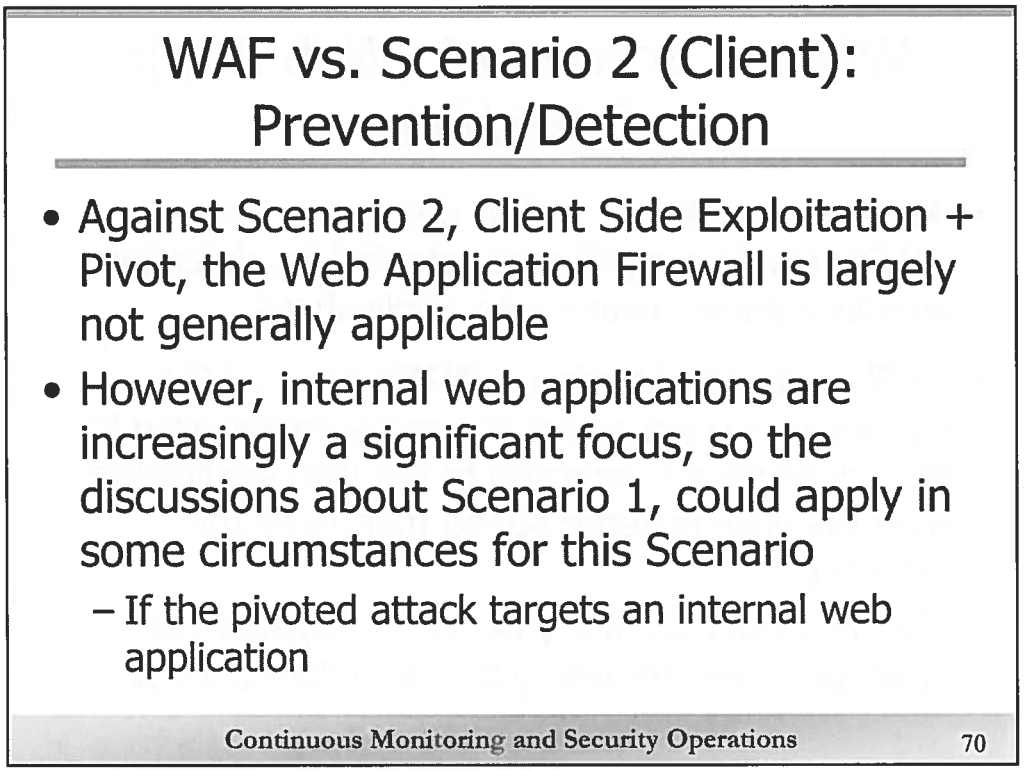
Also, realistically, we need to appreciate that most of our web applications are poorly secured from both a coding standpoint as well as from the external mitigation vantage point.



WAF vs. Scenario 1 (Web App): Detection

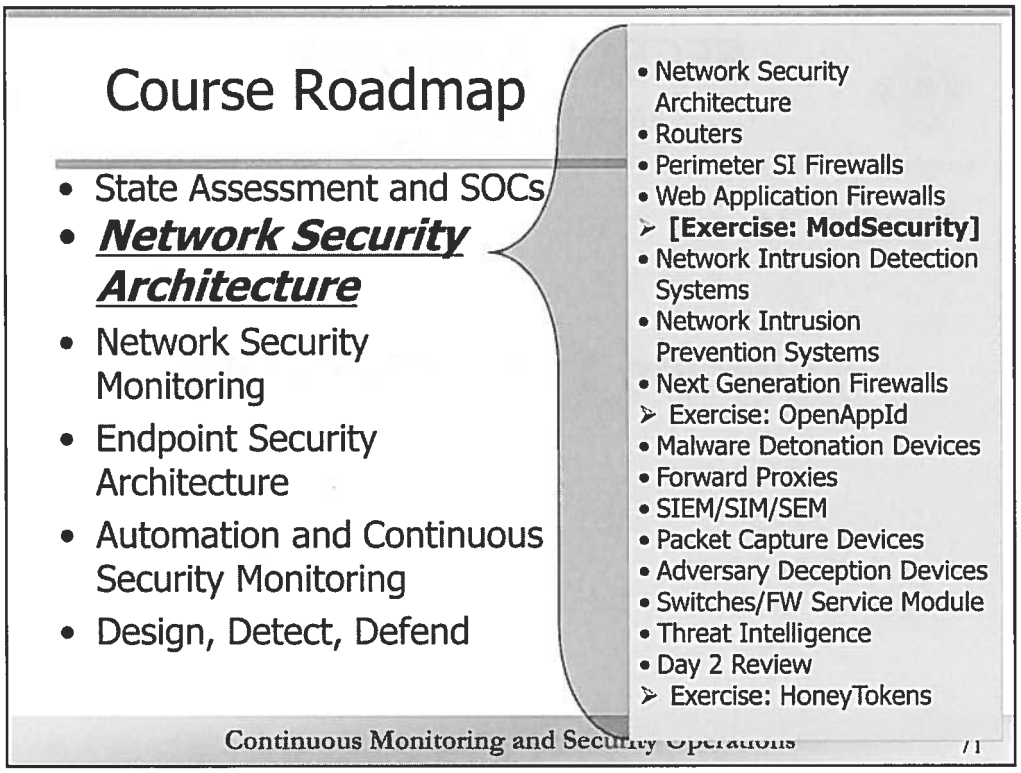
Though initially many organizations do not intend their WAF deployment to be primarily a detective control, but it often ends up being an overtly detective capabilities. I do not find this disconcerting at all. Our web applications have such poor supporting security infrastructure in most shops, we need all the help we can get on any front.

The WAF will almost certainly detect the attack and also the exfiltration attempt. Though WAF bypass is not often terribly difficult to achieve, the adversary would, even if successful at bypassing prevention, have been obvious in the web application firewall.



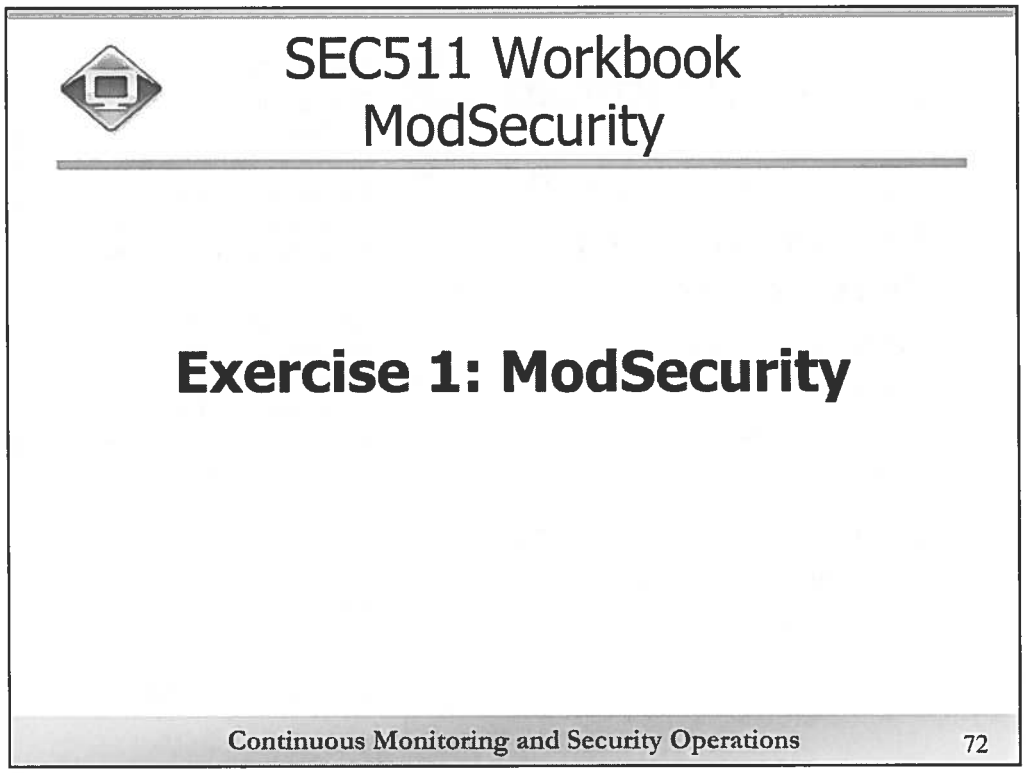
WAF vs. Scenario 2 (Client): Prevention/Detection

The WAF really has no capabilities with respect to scenario 2. No fault of the WAF, but it just does not work for the client side attack scenario.



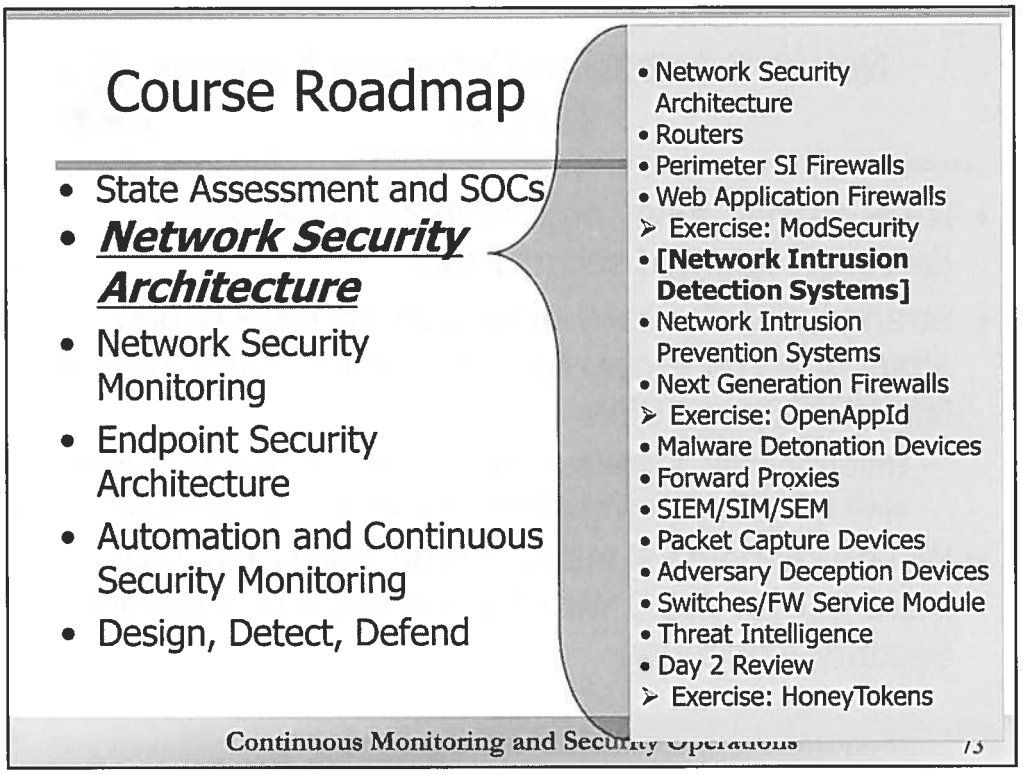
Course Roadmap

The next section is the ModSecurity exercise.



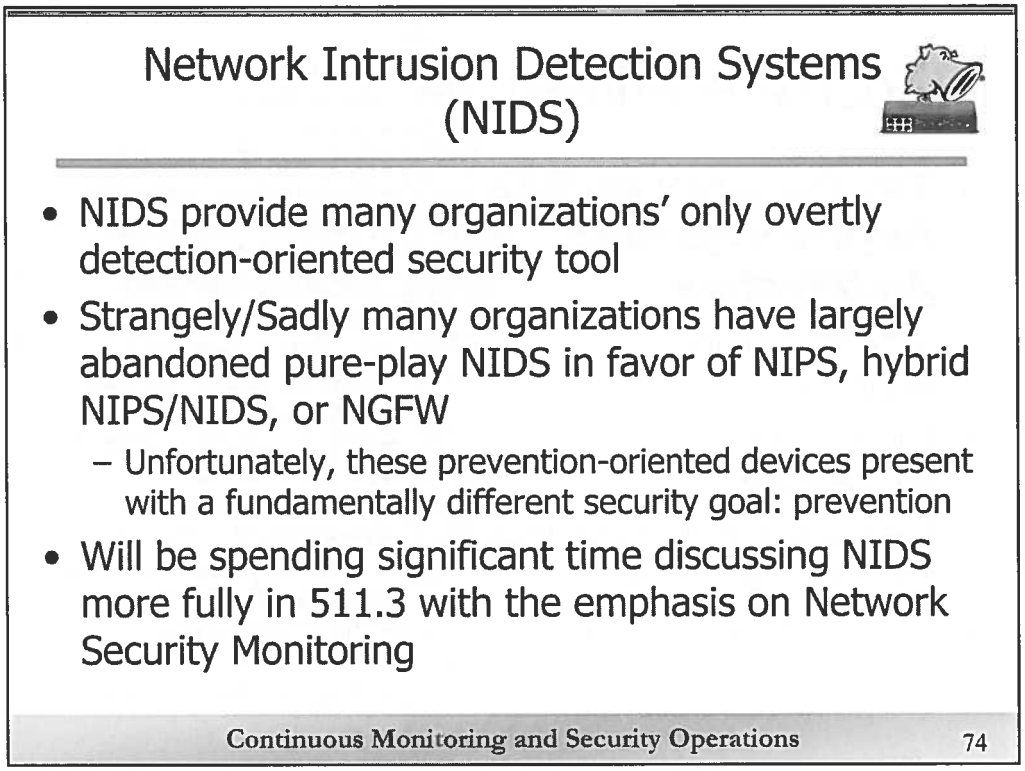
SEC511 Workbook ModSecurity

Please go to the 511 Exercise Workbook, section 511.2-1.



Course Roadmap

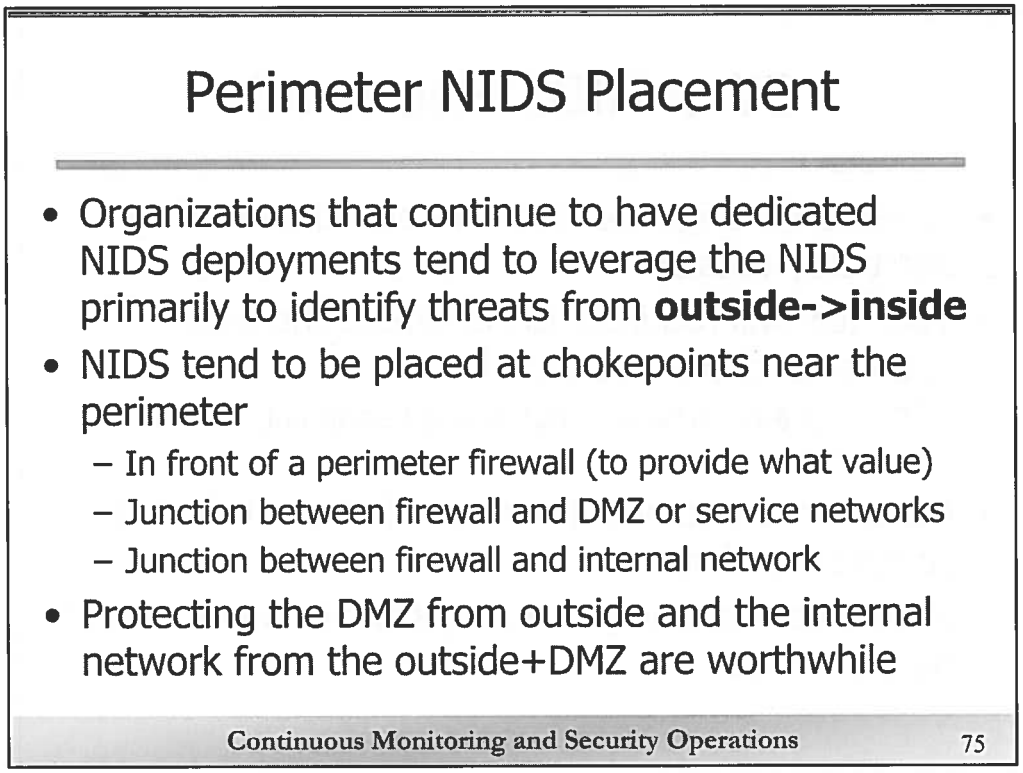
The next section is Network Intrusion Detection Systems.



Network Intrusion Detection Systems (NIDS)

Very often, the Network Intrusion Detection System is the only overtly detection-oriented device that many organizations have deployed. To make matters worse, many of them have plans, or have already, replaced their NIDS with a NIPS or even a NGFW.

Unfortunately, these prevention-oriented devices are fundamentally different than detection-oriented ones. This is true even if the NIPS is the same exact hardware appliance that can be used as a NIDS. Though it might make little sense that the same exact device can be drastically and fundamentally different, it is true due to the necessary configuration changes to support a prevention-oriented mindset.

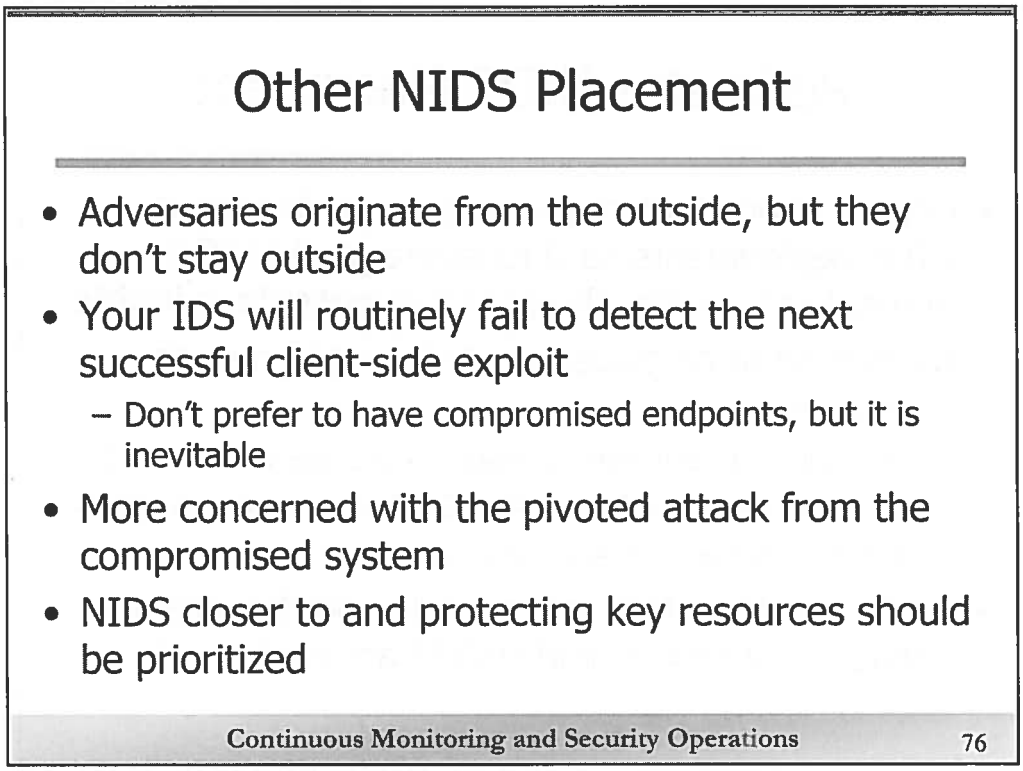


Perimeter NIDS Placement

Sadly, the perimeter-oriented NIDS could well be the only NIDS that exists. This NIDS commonly provides monitoring interfaces at a DMZ choke point and also a server chokepoint.

Monitoring data going from the firewall to the DMZ serves to protect the DMZ from external (to the DMZ) attackers. This means that not only would traffic being presented from the Internet be seen as potentially adversarial, but so too could traffic from the inside.

Another common location to situate a monitoring interface is where the firewall connects into the internal network. Like the DMZ sensor, this sensor would typically be configured to protect the internal network from external actors, which in this case is anyone not on the internal network.

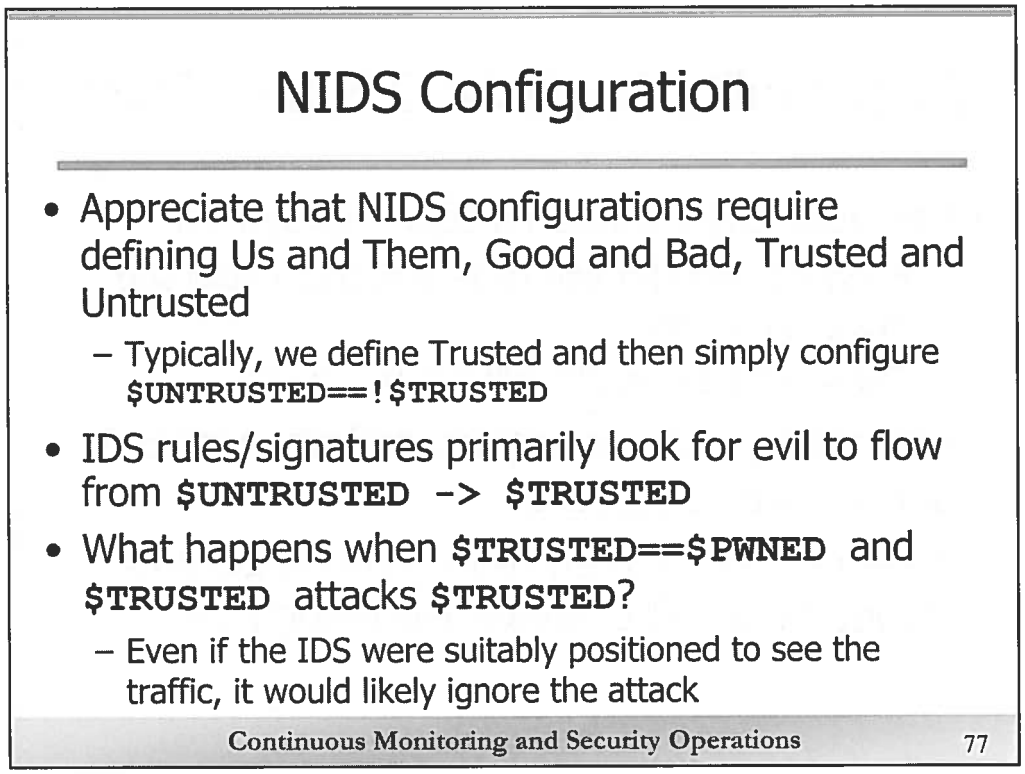


Other NIDS Placement

While the perimeter-focused NIDS is without question worthwhile, they are far from the only place that NIDS should live. Yes, it is true that the overwhelming majority of adversaries originate from the outside, but it is also trued that they do not stay outside for very long.

Once they bypass the external facing sensors with a cool client-side exploit adversaries will, almost without question, move laterally within the organization. Your external focused NIDS has zero visibility at that level.

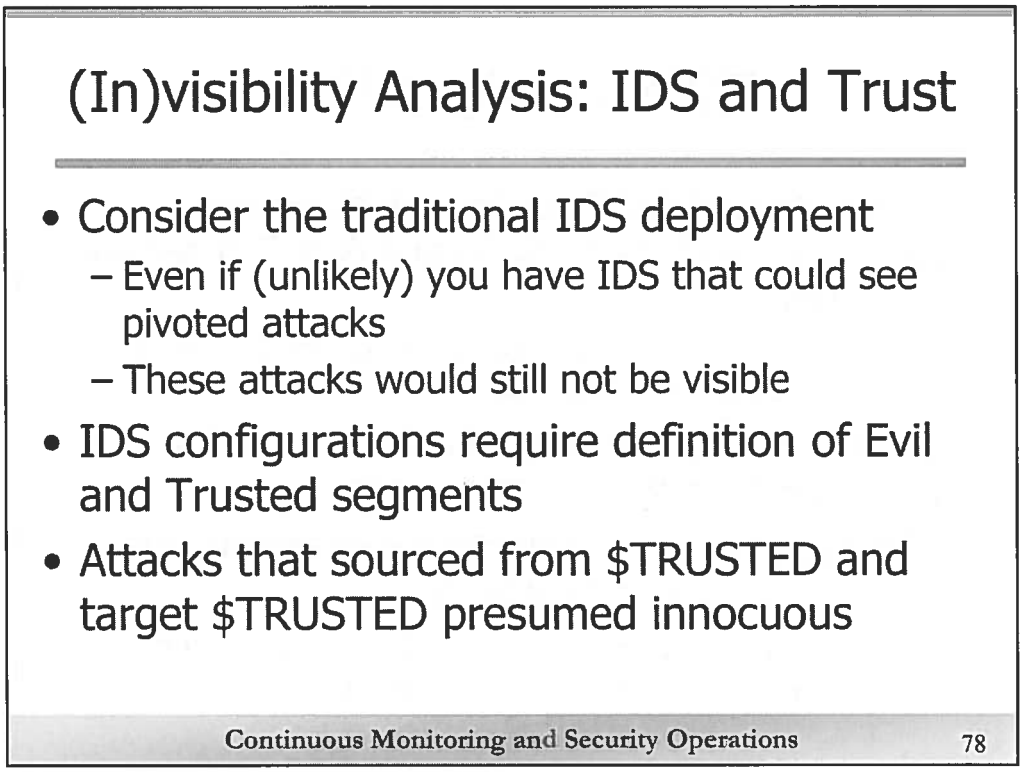
One major security posture improvement that every organization should consider is employing internal NIDS, especially in order to better protect key internal systems.



NIDS Configuration

One consideration that is lost on most folks that lack intimate knowledge of NIDS is to appreciate the configuration. The most basic configuration of a NIDS is to define what constitutes the $TRUSTED network. What are we trying to protect? Another common configuration would be to define the $UNTRUSTED, which most commonly is just defined by reference, !$TRUSTED.

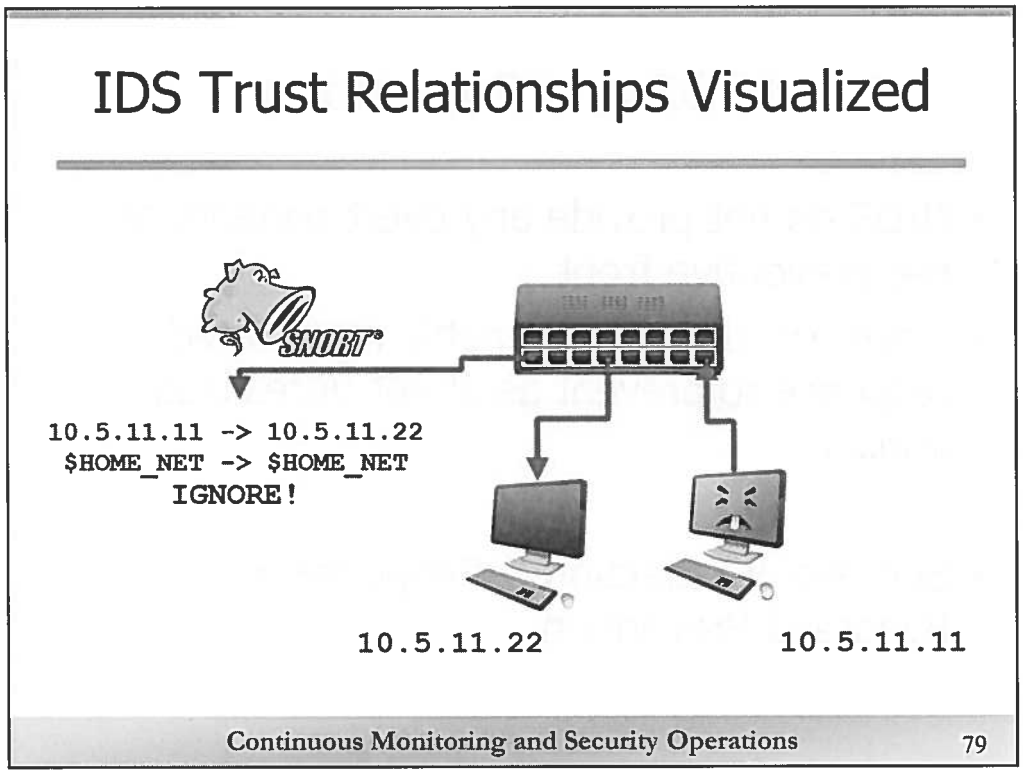
Most IDS are configured with rules/signatures that expect to find an $UNTRUSTED and a $TRUSTED. This is fine for some circumstances, but what happens when an internal $TRUSTED system becomes compromised. If $TRUSTED targets $TRUSTED, even in the unlikely event that the IDS is capable of seeing the traffic, it will often ignore even overt attacks launched with this communication path.



(In)visibility Analysis: IDS and Trust

An example that illustrates a common failing that many organizations do not even realize exists involves a typical IDS deployment.

Though unlikely, imagine an organization actually instrumented an IDS that could see internal to internal traffic. The most basic configuration of an IDS involves defining trusted and untrusted segments. In Snort speak these are referred to as HOME_NET and EXTERNAL_NET. Most of the signatures/rules look specifically for attacks to be sourced from the untrusted segment.

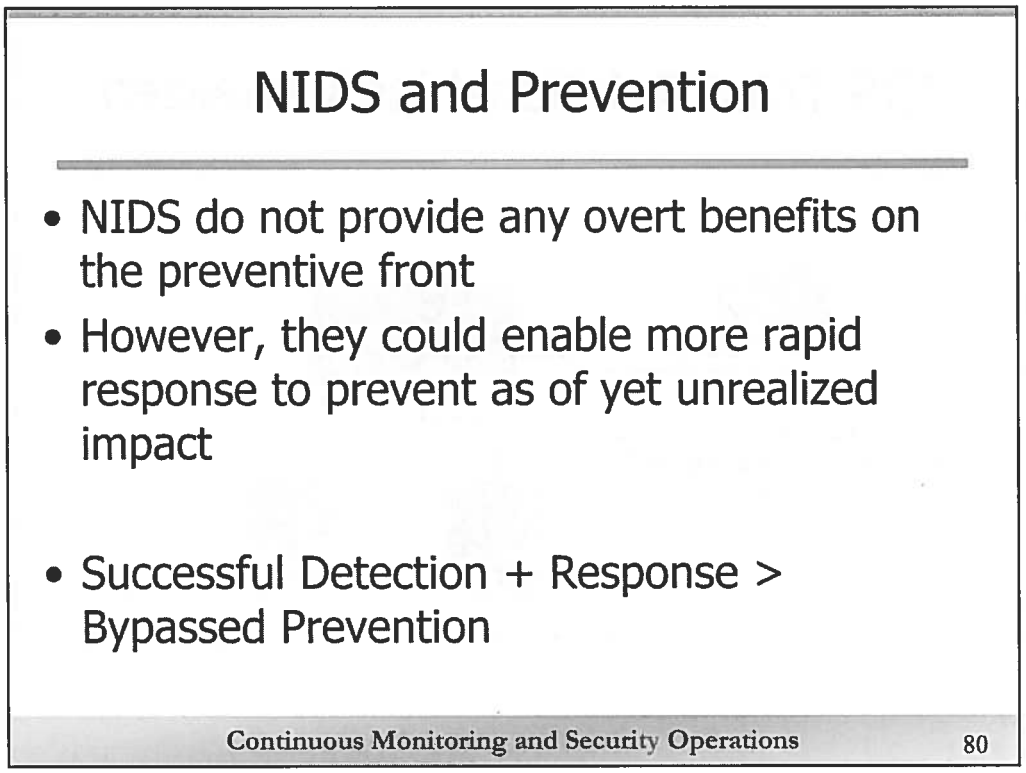


IDS Trust Relationships Visualized

The slide above illustrates the general lack of visibility for a pivoted internal attack. Here we see a compromised host (10.5.11.11) targeting a victim on the same subnet (10.5.11.22). An IDS is suitably positioned to see the traffic and ignores the traffic because the flow is from trusted segment to trusted segment, or `$HOME_NET->$HOME_NET`

We use the highly recognizable Snort Pig to represent the IDS in this slide. We will be learning more about and using Snort[1] in the class.

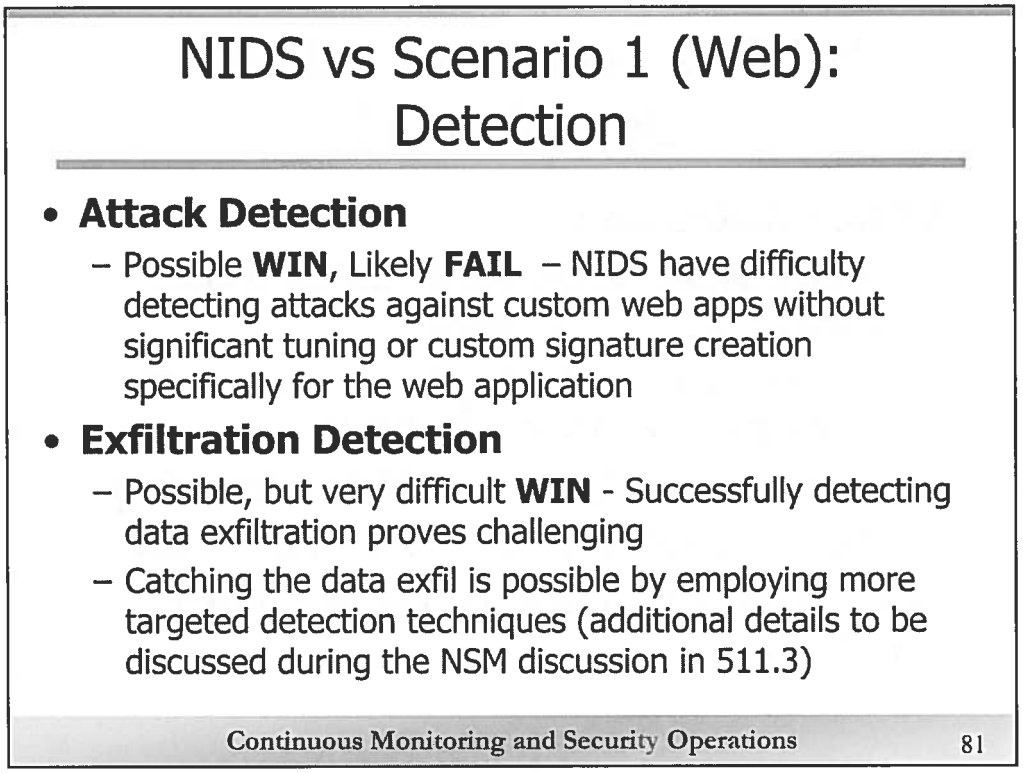
[1] http://snort.org (http://cyber.gd/511_128) **QR**



NIDS and Prevention

Should come as little surprise that the NIDS does not provide any direct prevention capabilities. That being said, we can absolutely better our preventive capabilities as a direct result of things we are seeing on the NIDS.

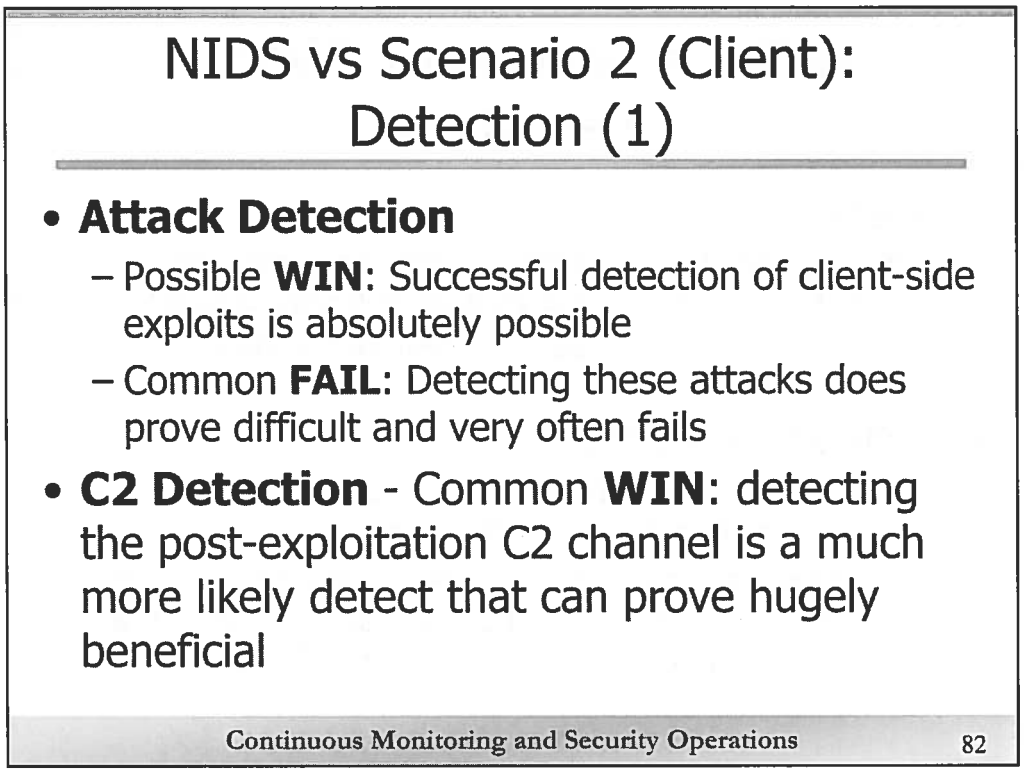
Also, and more importantly, the NIDS when properly tuned and staffed can be a great adjuvant to preventing compromise by affording us rapid detection, which can then be fed to response.



NIDS vs Scenario 1 (Web): Detection

NIDS are poor performing when it comes to detecting attacks against custom web applications. Generic signatures for web application attacks do exist that possibly could catch the web application attacks. However, these very often fail miserably or are extremely prone to false positive and are suppressed or ignored.

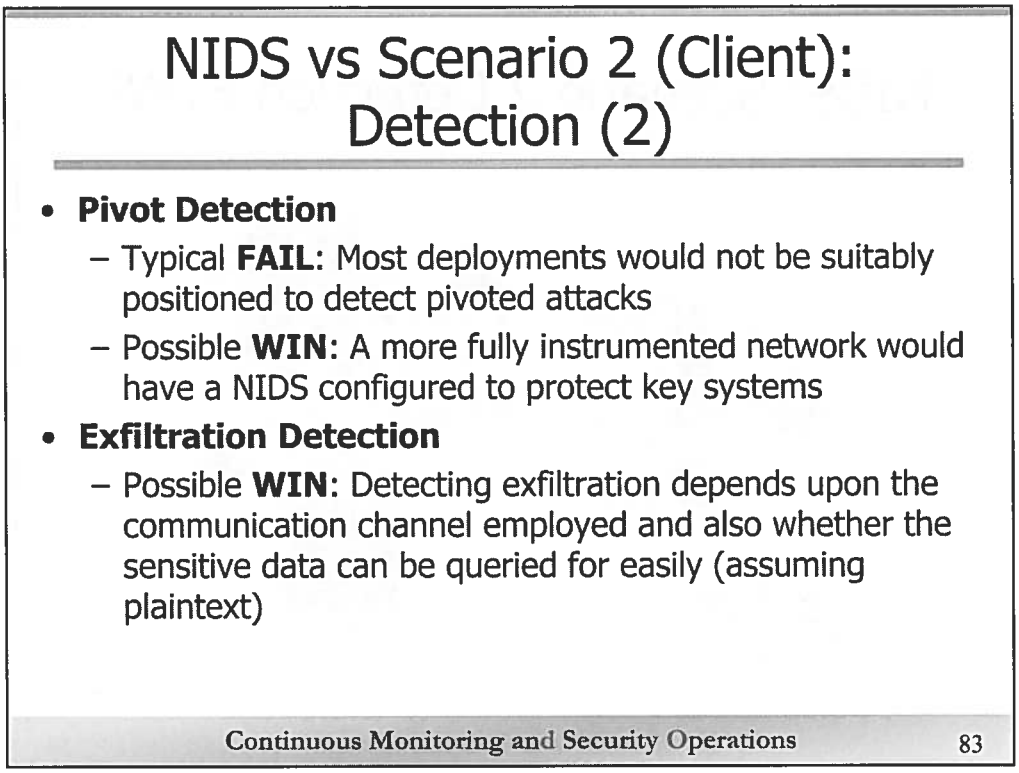
Detecting the exfiltration of data too can prove extremely difficult, but is possible. Naturally the success of the detect depends upon the data in question, whether the data was sent in plaintext, and the difference in volume of breach vs. normal traffic.



NIDS vs Scenario 2 (Client): Detection (1)

On the client-side exploitation the NIDS can prove significantly more helpful. Detecting client-side attacks happens regularly, however, the particular client-side attacks used change rapidly and often the detect can/will be bypassed.

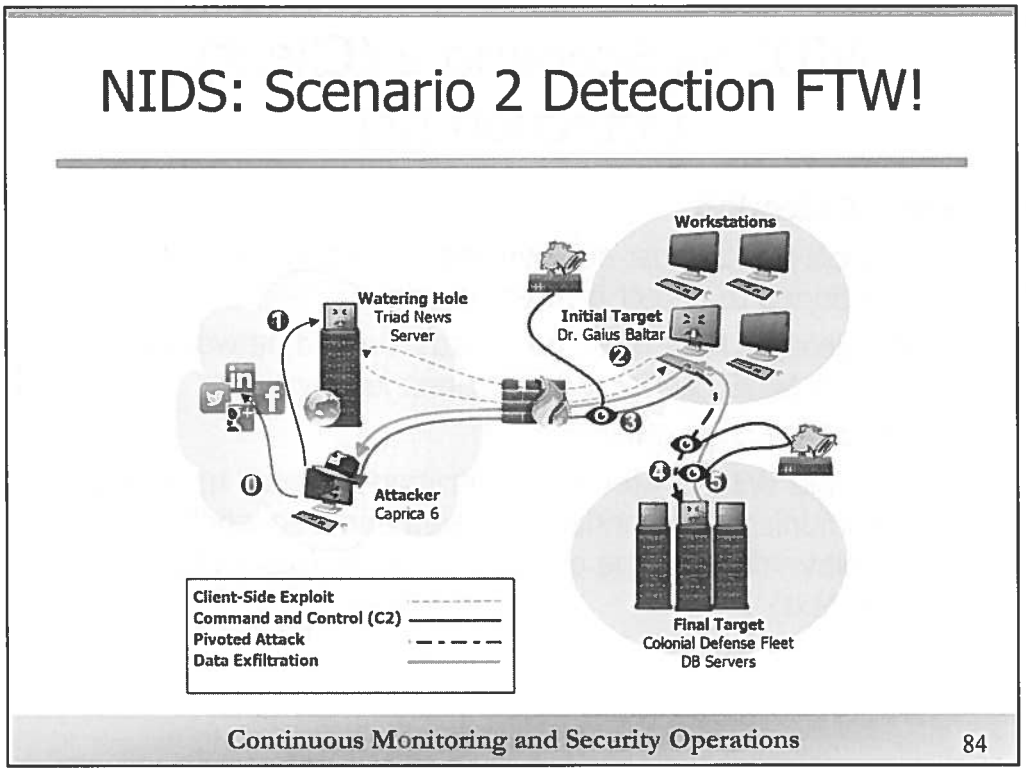
C2 detection is a big potential win for the NIDS. While it is true that adversaries can in fact employ C2 channels that would be fiendishly difficult to detect by the NIDS, they are still commonly either initially attempting or even simply employing C2 that is somewhat straightforward to detect, if the NIDS has been tuned appropriately.



NIDS vs Scenario 2 (Client): Detection (2)

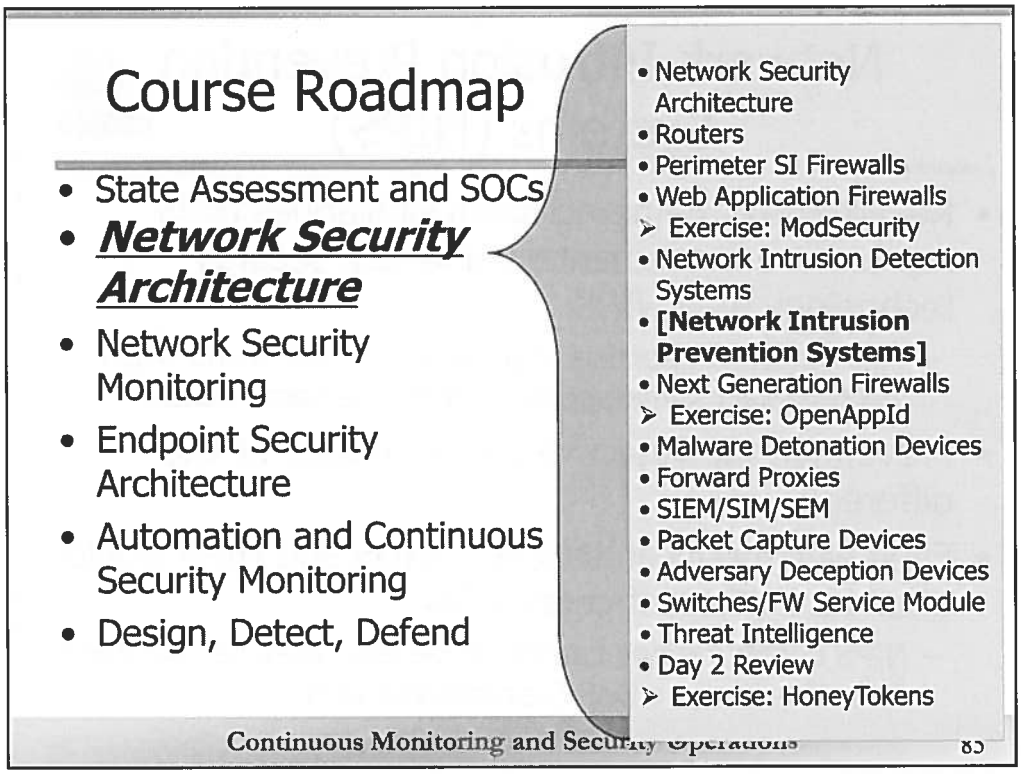
Detecting pivoted attacks is typically not a possibility for the majority of organizations' NIDS infrastructure due to the nature of the placement and configuration of the NIDS. However, if an organization moves to a more robust internal security architecture, then they will greatly increase the likelihood of detecting these pivoted attacks.

On the data exfiltration front, we are again rather dependent upon the nature of the data and the manner in which it was stolen to determine whether or not we would end up being successful.



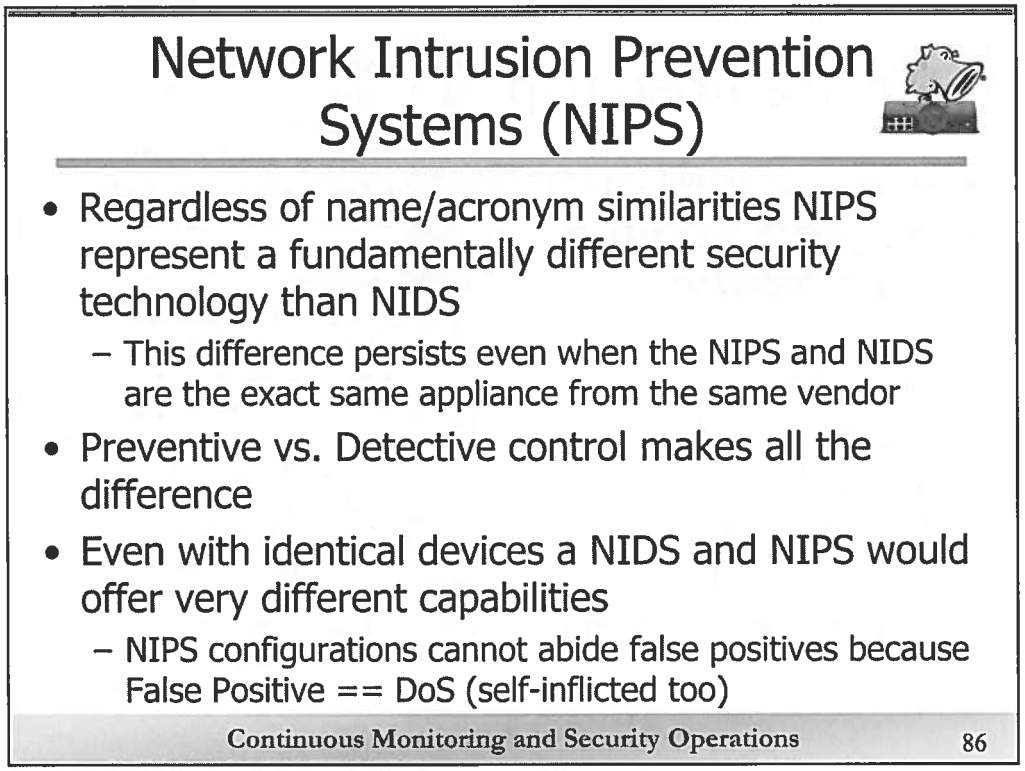
NIDS: Scenario 2 Detection FTW!

Above we see the successful detection of the NIDS illustrated. In particular, the NIDS is especially helpful at detecting C2 channels. Also, if internal NIDS are instrumented the possibility of detecting pivots and data exfiltration increases significantly.



Course Roadmap

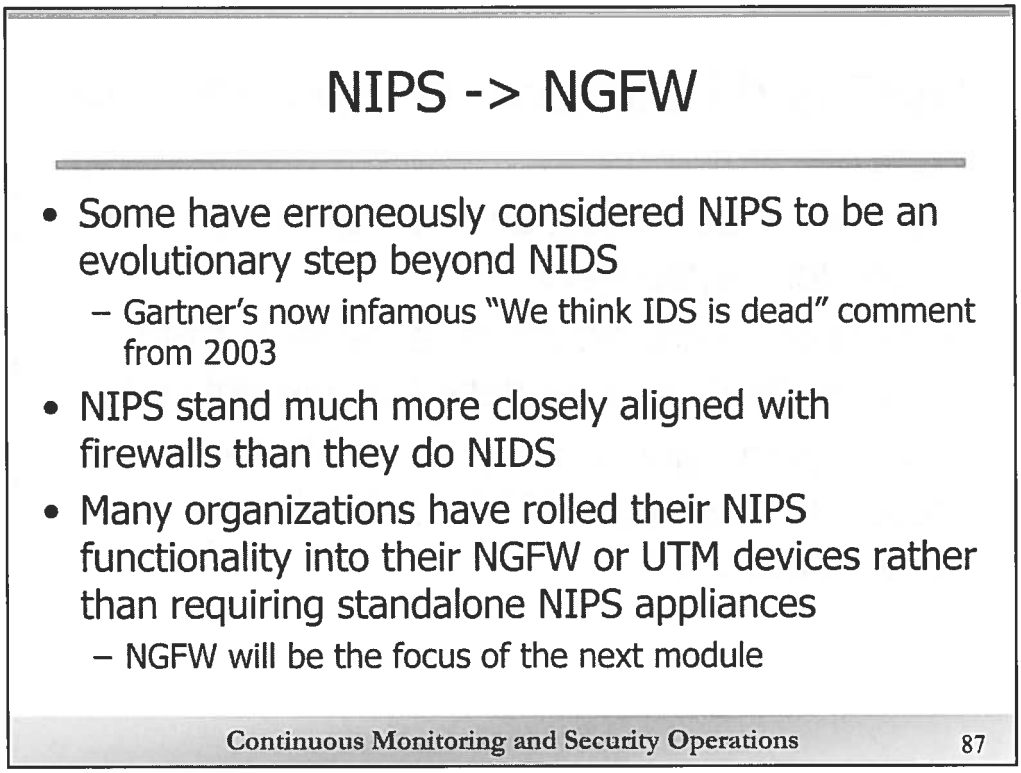
The next section is Network Intrusion Prevention Systems.



Network Intrusion Prevention Systems (NIPS)

Though the name and even hardware are extremely similar, NIDS and NIPS are materially different. Again this is true even if the exact same hardware can be used for both NIDS and NIPS (or a hybrid).

Fundamentally these are extremely different because the nature of the configuration required. The easiest conceptual distinction is with False Positives. A false positive on a NIDS is an annoyance to be sure, but does not cause business disruption. Whereas a false positive on an IPS causes service outages. Necessarily then the configuration of an IPS must be such that false positives cannot occur.

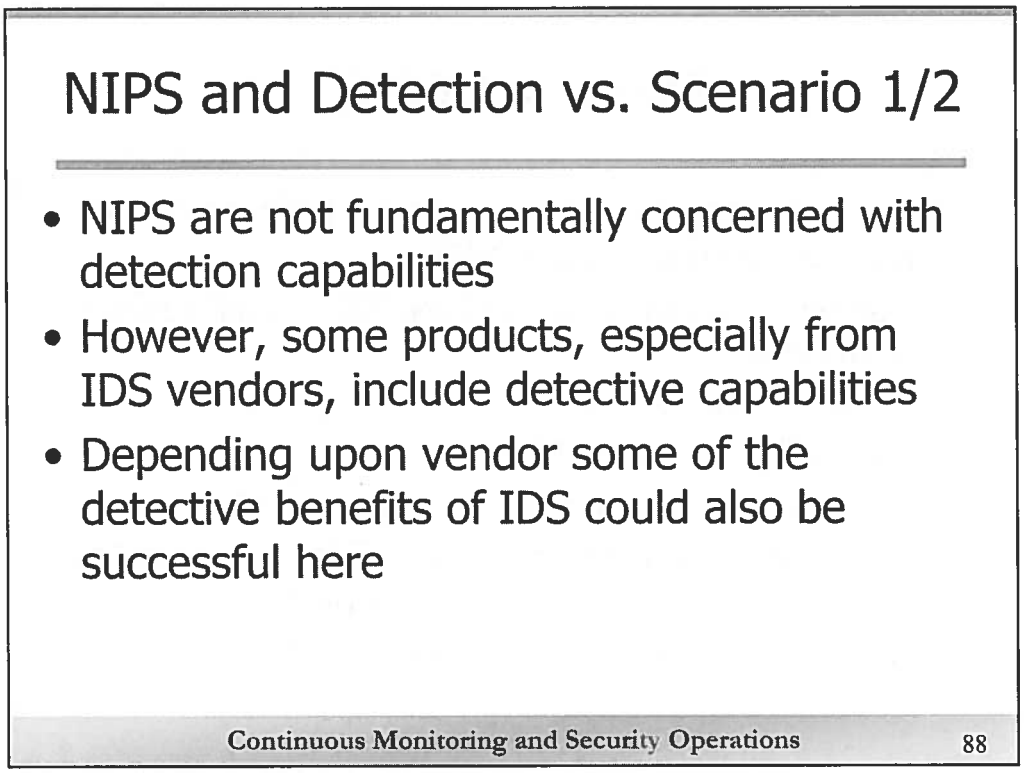


NIPS -> NGFW

Gartner is infamous for having stated, "We think IDS is dead" in 2003[1]. The suggestion had to do with the lack of significant benefit most IDS deployments were having at the time. In order to provide benefit there must be someone skilled on the other end of the IDS, whereas benefit can be derived from the IPS without direct interaction.

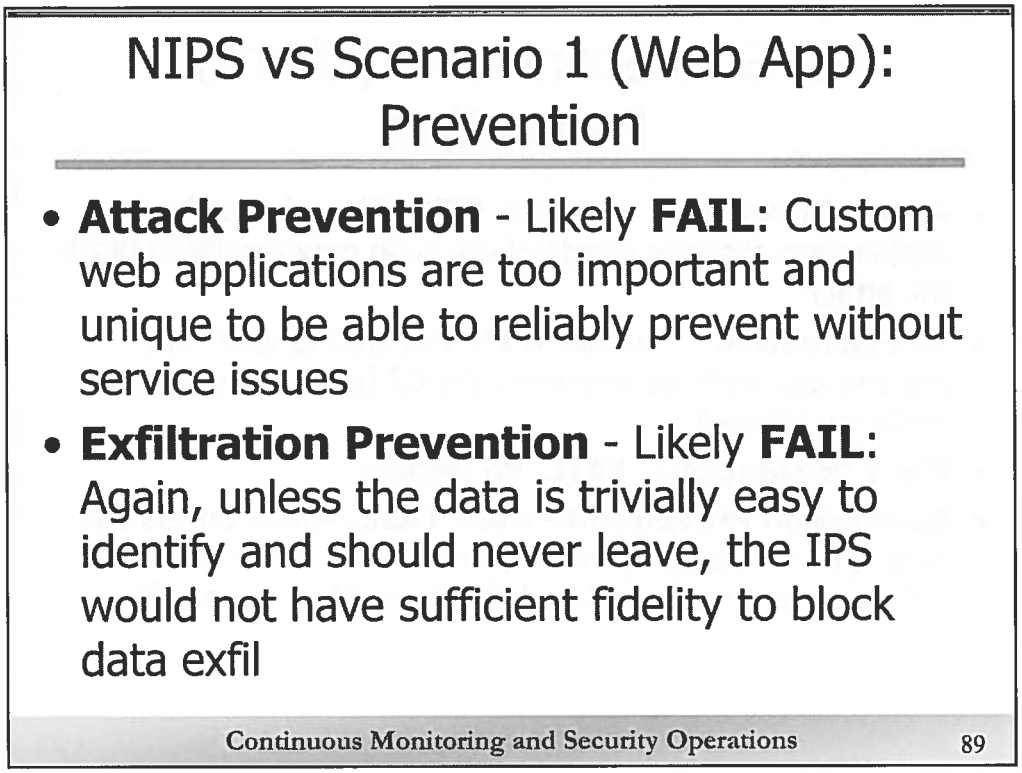
In truth, IPS are much closer to FW than they are to IDS. While I am by no means declaring IPS dead or suggesting you should abandon your IPS deployment, but there seems to be a lot of migration from pure IPS to NGFW. As an interesting example of this, both Sourcefire (before being acquired by Cisco) and TippingPoint, both of which are known for NGIPS also offer NGWF based upon very similar technology and underlying engines.

[1] http://www.businesswire.com/news/home/20030611005056/en/Gartner-Information-Security-Hype-Cycle-Declares-Intrusion#.U1al3vldW0g (http://cyber.gd/511_225) **QR**



NIPS and Detection vs. Scenario 1/2

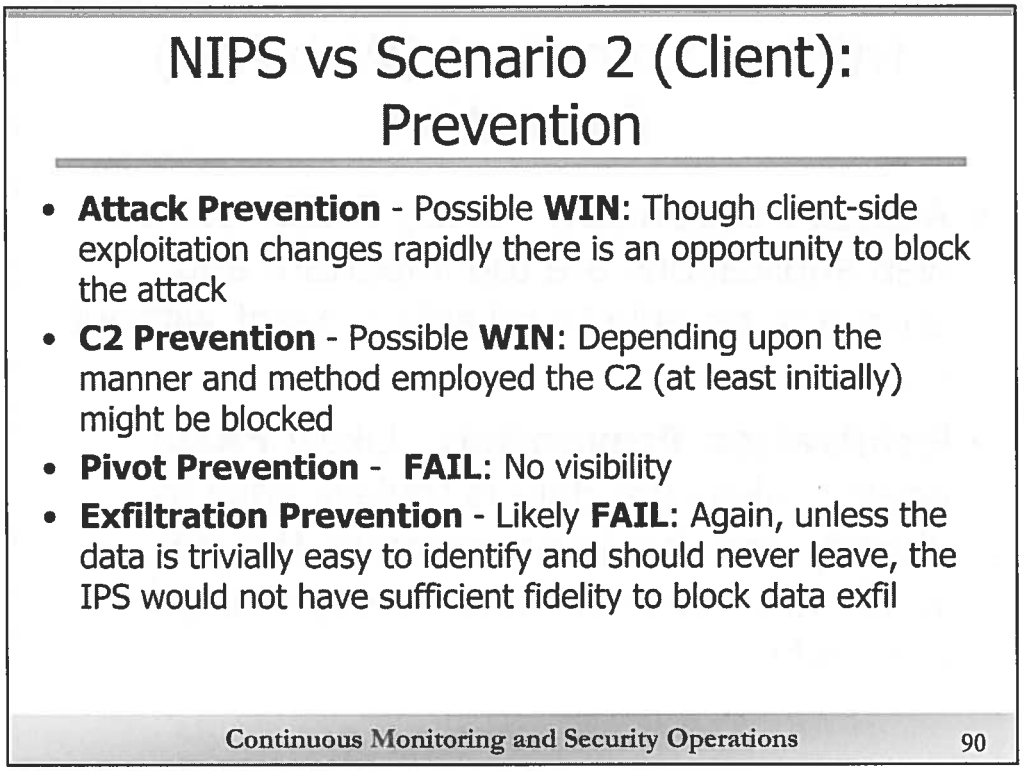
Network IPS are necessarily not intended primarily to be detective in nature. However, some products, especially if the vendor has roots in IDS, include detective capabilities as well. So, while not necessarily a stated benefit of NIPS, some products could potentially assist on the detection front.



NIPS vs Scenario 1 (Web App): Prevention

The nature of custom web applications is such that IPS would be hard pressed to have high enough fidelity blocks that would not also run the risk of service disruption.

On the data exfiltration front, again unless it could be made extremely clear the IPS would be unable to have high enough fidelity rules to block the exfiltration.

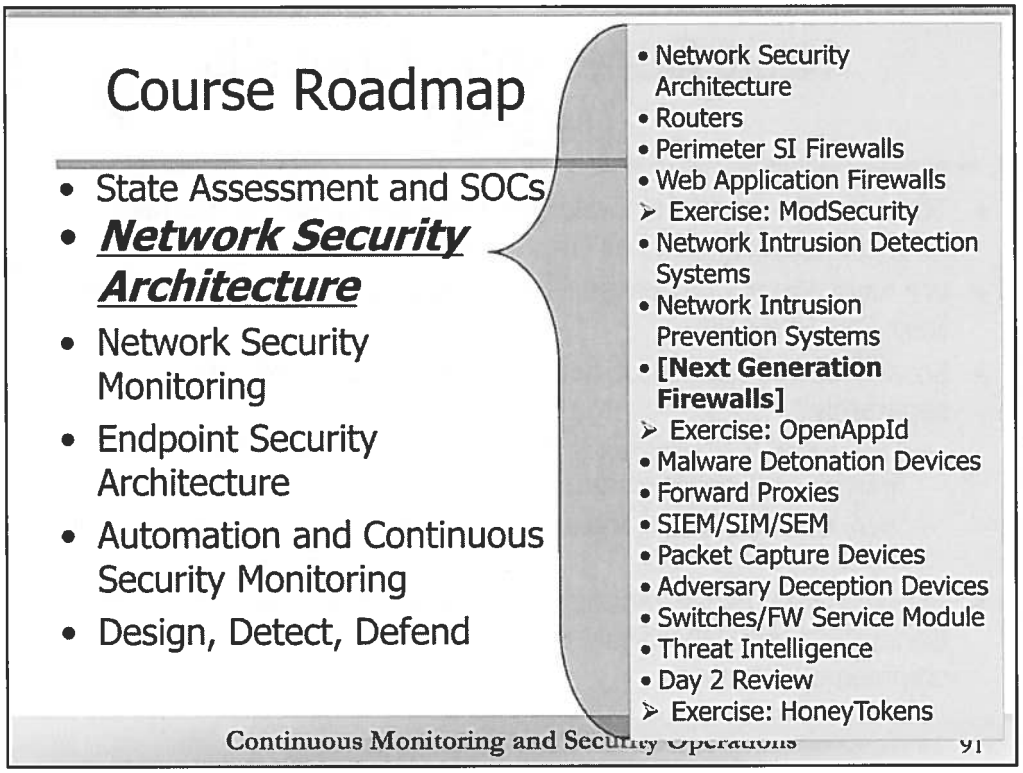


NIPS vs Scenario 2 (Client): Prevention

With respect to client-side exploitation the NIPS can fair a bit better. Commentary on exfiltration prevention remains largely the same as we found with the web application. Due to location, the NIPS has no visibility into the pivot.

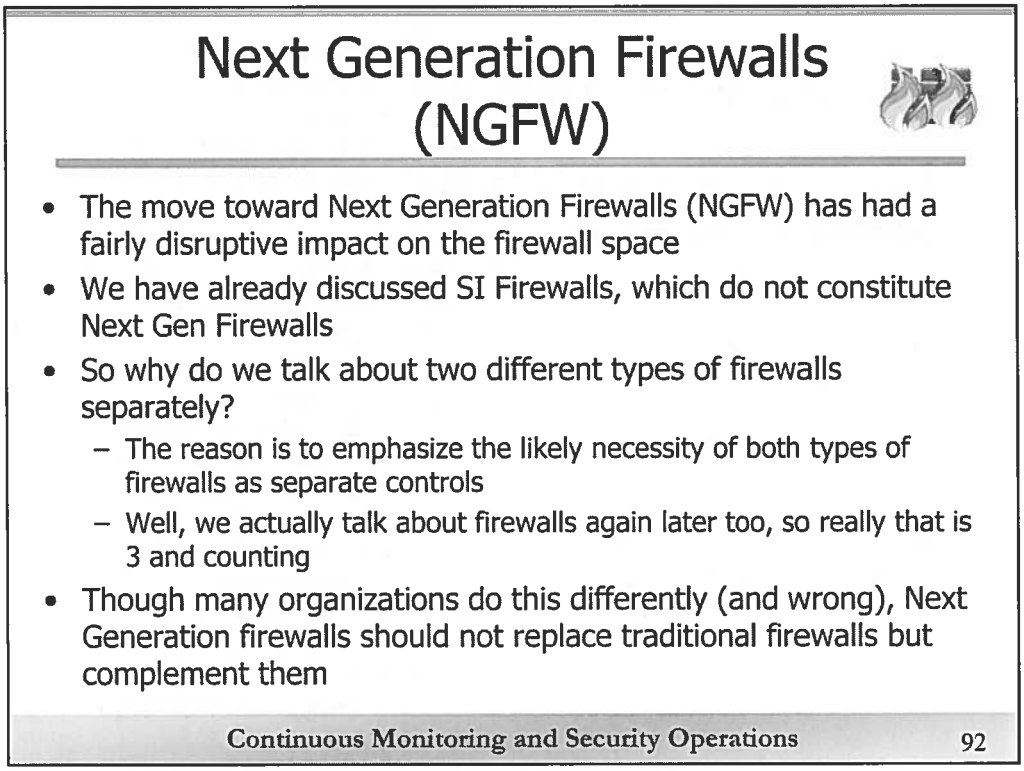
On the attack front, the NIPS does have potential to block the attack. This is especially likely in the case of exploitation of a known, but unpatched, vulnerability.

With regards to the C2, the NIPS could prove initially successful for some methods of C2. Though ultimately we would expect bypass to be possible.



Course Roadmap

The next section is Next Generation Firewalls.

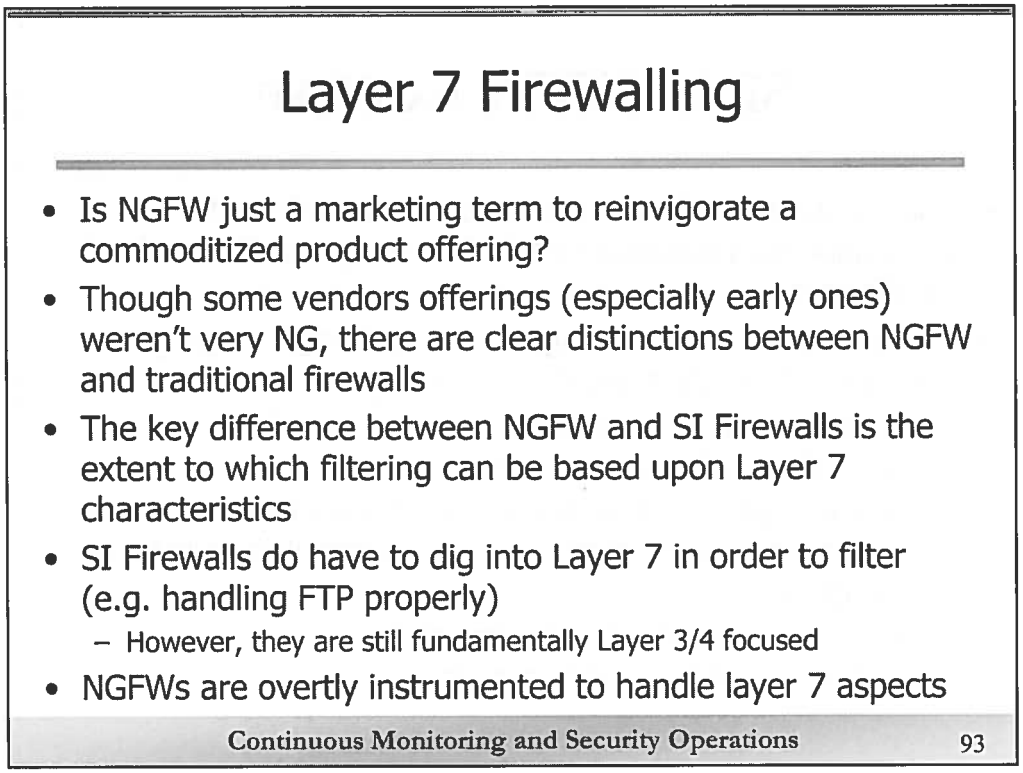


Next Generation Firewalls (NGFW)

Firewalls those old stalwarts of network security have changed quite a bit as of late. Though we have already talked about SI (Stateful Inspection) firewalls, now we can attend to a newer breed of firewall, NGFW.

Honestly, when I first started hearing the term NGFW bandied about, I thought it was utterly a marketing gimmick. Though I suppose there is some truth to the marketing angle, as NGFW is still fundamentally a firewall, NGFW do employ some specific tactics, distinct from SI, to achieve more robust capabilities warranted in today's threat landscape.

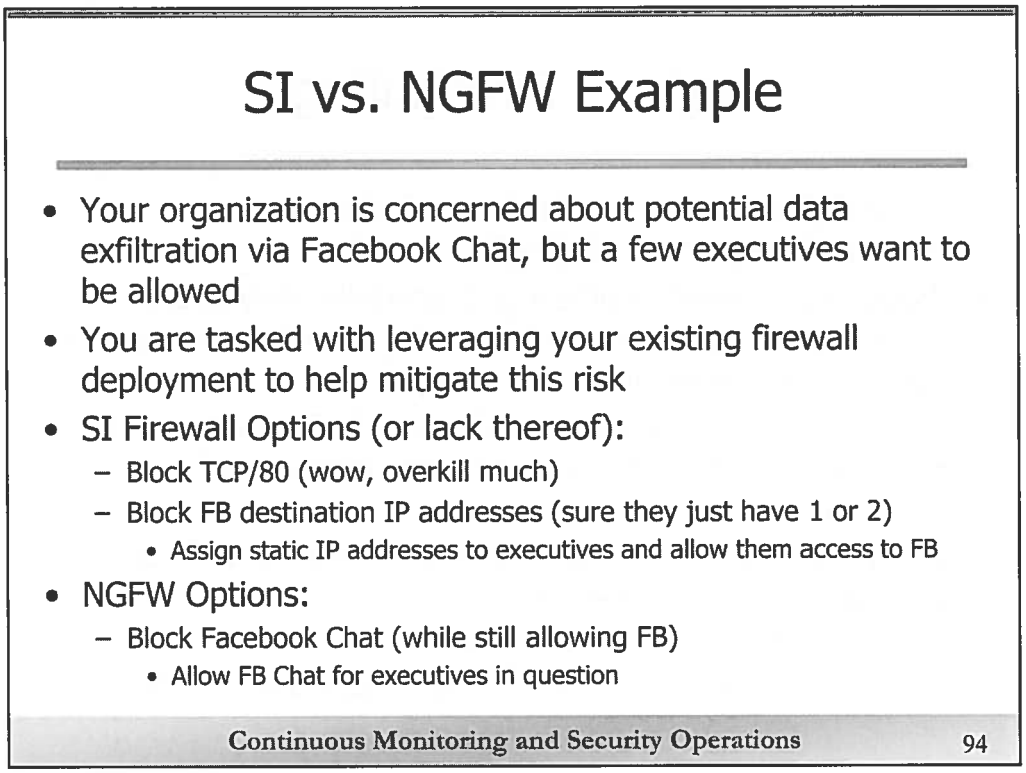
One point of order regarding NGFW, these devices, even though they are firewalls and cooler than SI firewalls, should not replace but complement the SI firewall deployment.



Layer 7 Firewalling

One of the most significant changes with the NGFW beyond more traditional firewalls is the capability and overt emphasis on Layer 7. Now, in truth, SI firewalls have historically dabbled a bit in Layer 7, but it was largely to better handle state more than providing overtly significant firewalling capabilities beyond Layer 3/Layer 4. At least initially that was the case.

NGFW have been built from the ground up with Layer 7 squarely in mind. This is a distinguishing characteristic that some traditional firewall vendors are absolutely having to play catch-up on.

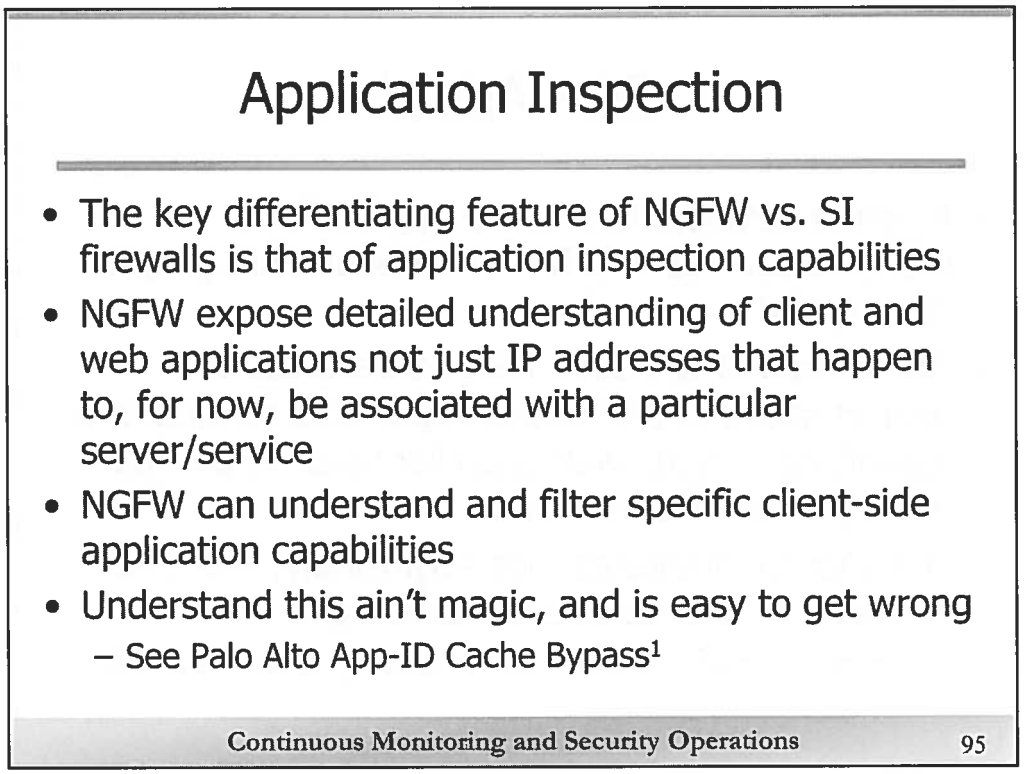


SI vs. NGFW Example

Let us consider a scenario to help illustrate some key differences between SI and NGFW. This can help you simply to better understand the offering and its capabilities. However, it is actually more important than that because every firewall is now a NGFW according to your vendors, whether this is actually true or not.

Consider that you are tasked with blocking the potential use of Facebook Chat due to its potential use as a means of data exfiltration. Now, the organization is generally intended to be allowed access to FB, but not to FB Chat. Oh, and there are a few executives that want to be able to access it in spite of the general ban.

Um, good luck pulling that off with a traditional SI firewall.

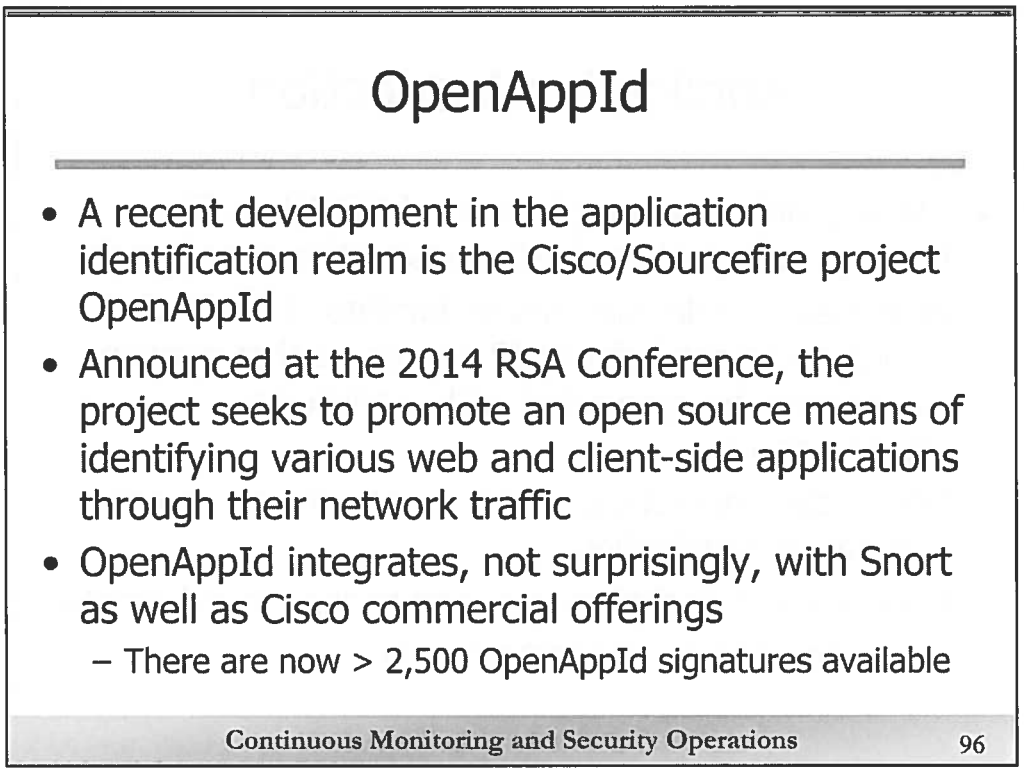


Application Inspection

One of the key differentiators between SI and NGFW is the ability for the latter to dig deep into Layer 7. We are not simply talking about having a simplistic understanding of what the RFC for HTTP or FTP or SSH looks like, though that is a need as well. No, NGFW very often go well beyond simple matters of protocols even to the extent of understanding particular, custom, and typically popular web applications.

This can be a significant boon in the world where everything is a web application or a mobile application, and the browser talking over HTTP is the conduit to almost everything. Going beyond simple Layer 3/Layer 4 filtering, and even beyond simple protocol understanding, as some SI vendors do, is necessary in the modern world.

[1] http://blog.anitian.com/tag/appid-cache-poison/ (http://cyber.gd/511_227) **QR**

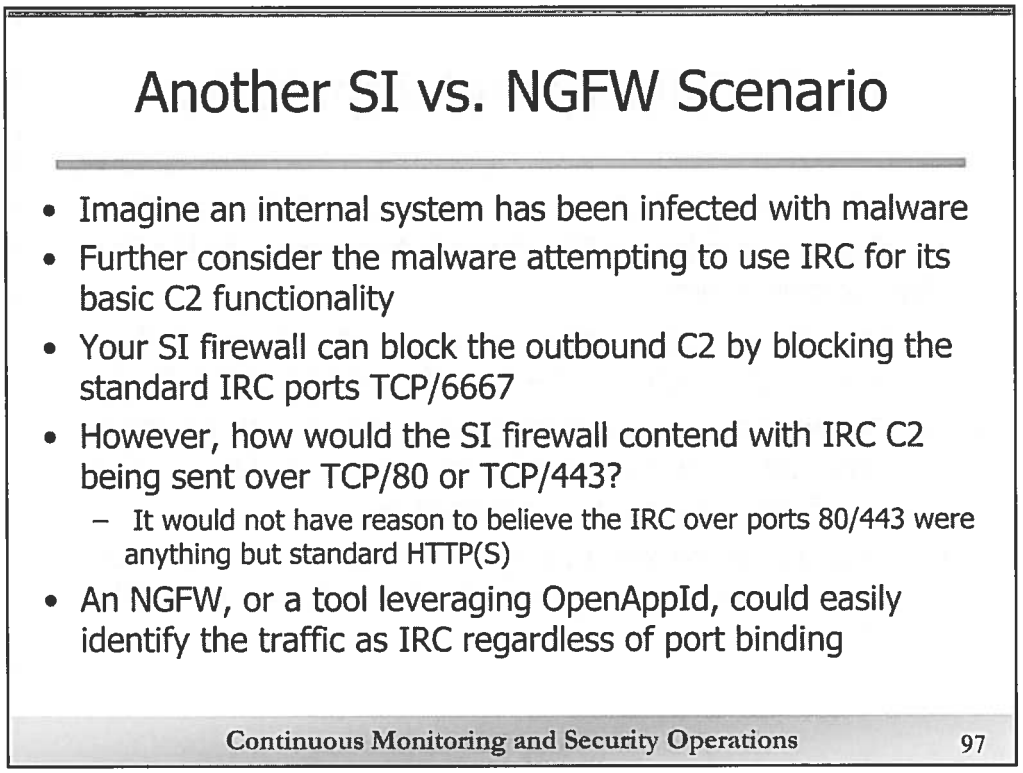


OpenAppId

A recent development in the Application Inspection/Identification space is OpenAppId. Sourcefire/Cisco released OpenAppId at RSA 2014. The project seeks to allow an open source framework for identification of particular applications. Again, we are not simply talking about, "Hey, that looks like HTTP…" but rather a much deeper understanding of the particulars of common web applications (though there are others web applications are very commonly a significant chunk of these).

Naturally, OpenAppId integrates with Cisco and Sourcefire offerings. One offering in particular though is quite interesting on that front, Snort. What this means is that the most popular IDS in the world, which happens to be open source, will gain an open framework for understanding and identifying applications.

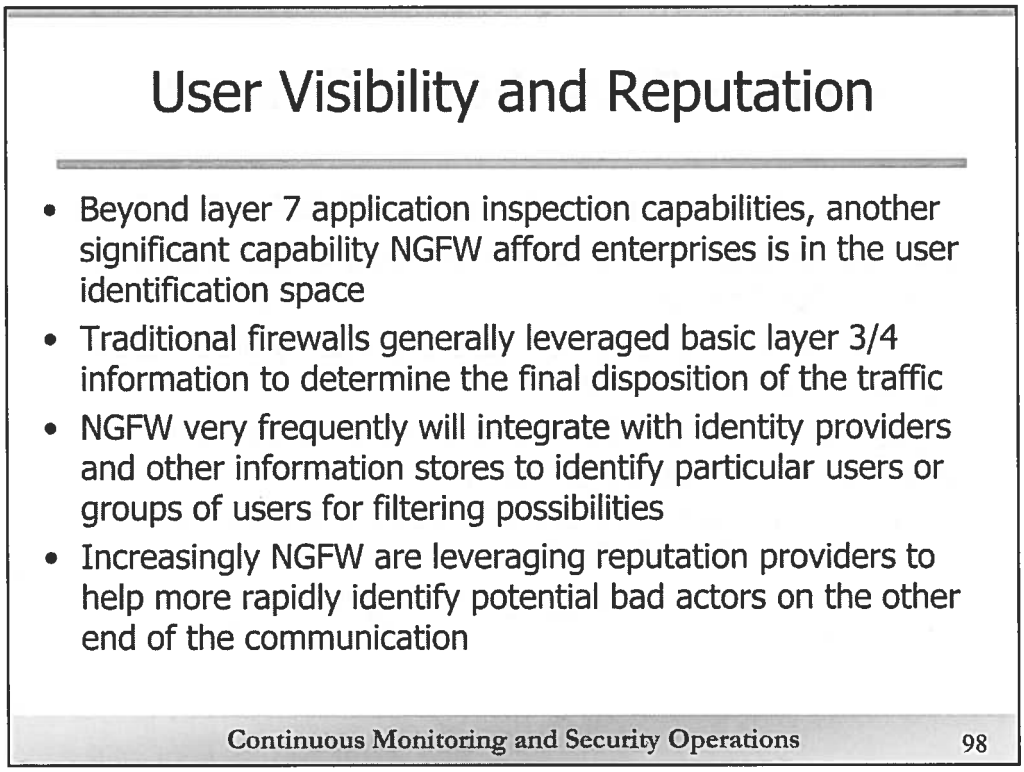
[1] http://blogs.cisco.com/security/cisco-announces-openappid-the-next-open-source-game-changer-in-cybersecurity/ (http://cyber.gd/511_226) **QR**



Another SI vs. NGFW Scenario

While the Facebook illustration was easy to understand and related to security, let us consider another scenario where application identification could have very significant impact.

Consider the scenario where an adversary, expecting that the target employs egress filtering, decides to perform their IRC-based C2 over TCP/80 or TCP/443 rather than TCP/6667. Whereas our traditional Layer 3/4 capabilities would pass this traffic as simply outbound traffic that matches the Layer 3 and Layer 4 requirements, a NGFW could potentially realize that the traffic in question is in fact IRC and block it as non-conforming.

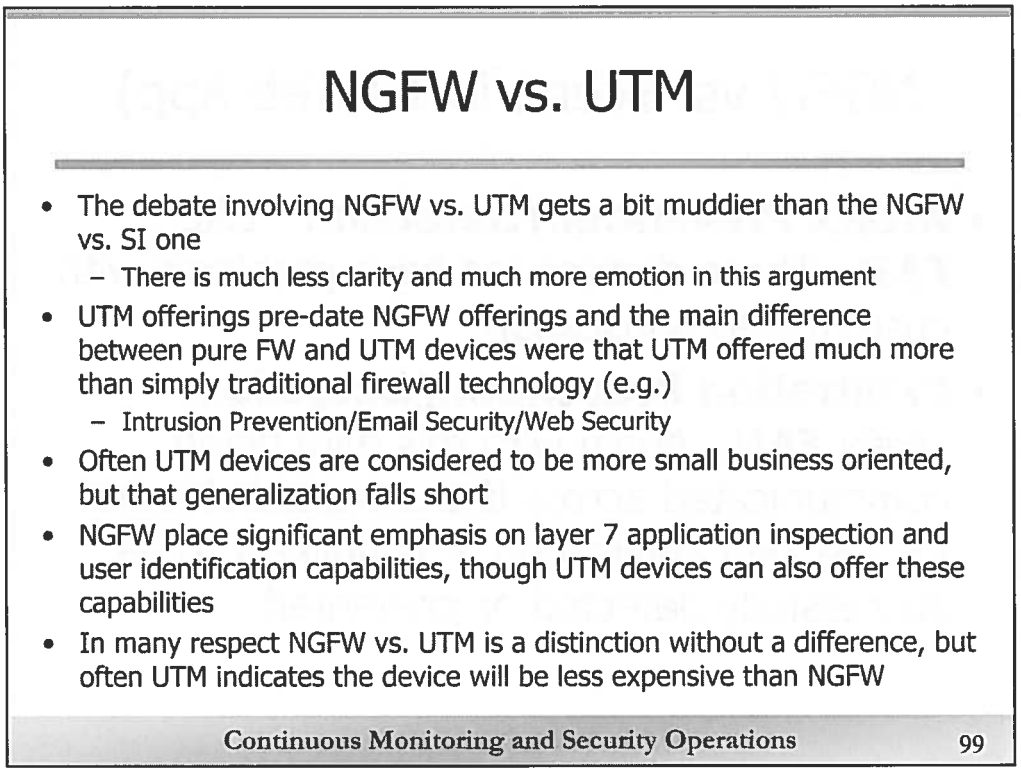


User Visibility and Reputation

Other characteristics of NGFW beyond traditional SI firewalls is the detailed tracking of users and the integration with reputation services.

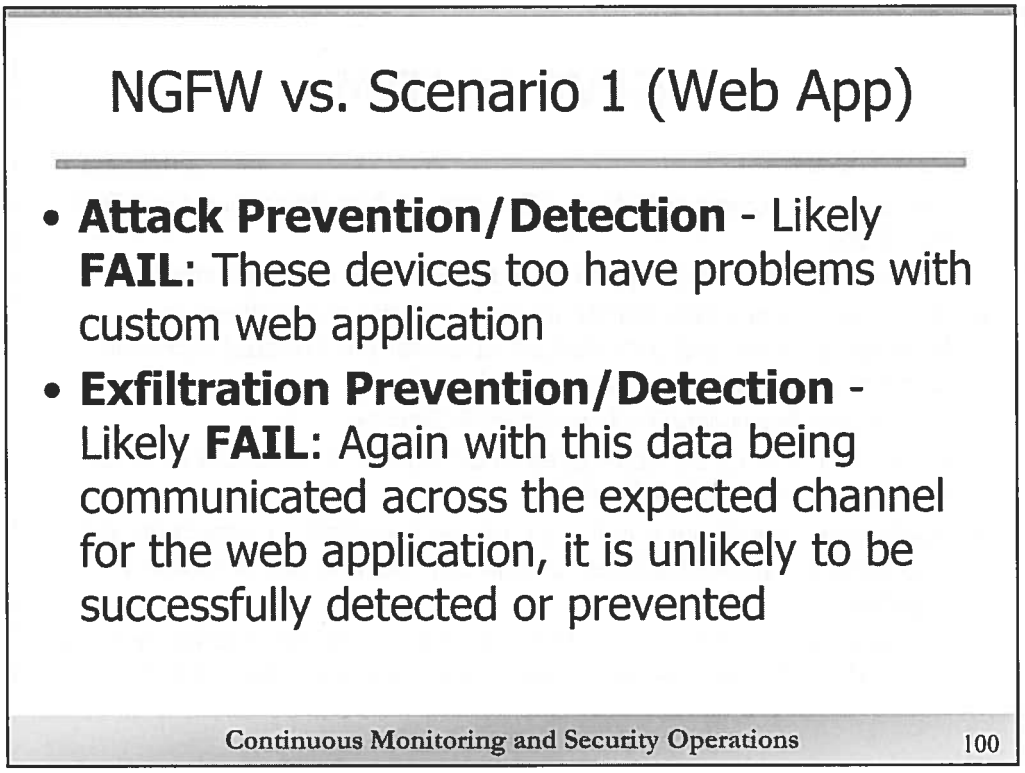
Historically, decisions about the disposition of traffic were based on simple IP address and Port information. However, with the common use of DHCP for clients, providing access to particular users or groups of users proved cumbersome. Typically to achieve this we have to isolate the users or groups of users to a particular VLAN so we would have a consistent IP address range to filter. Or, we configured a static IP address for the client in question so that we could provide appropriate filtering. NGFW typically have the ability to integrate with Identity Providers, such as AD, and necessary infrastructure to provide enhanced control down to the user level if needed.

Another common characteristic of NGFW is the increasing reliance on reputation-informed decisions. Typically this involves being linked up with a reputation service that helps to identify the security-relevant reputation of the system or network on the other end. We will be discussing reputation based information, and threat intelligence later.



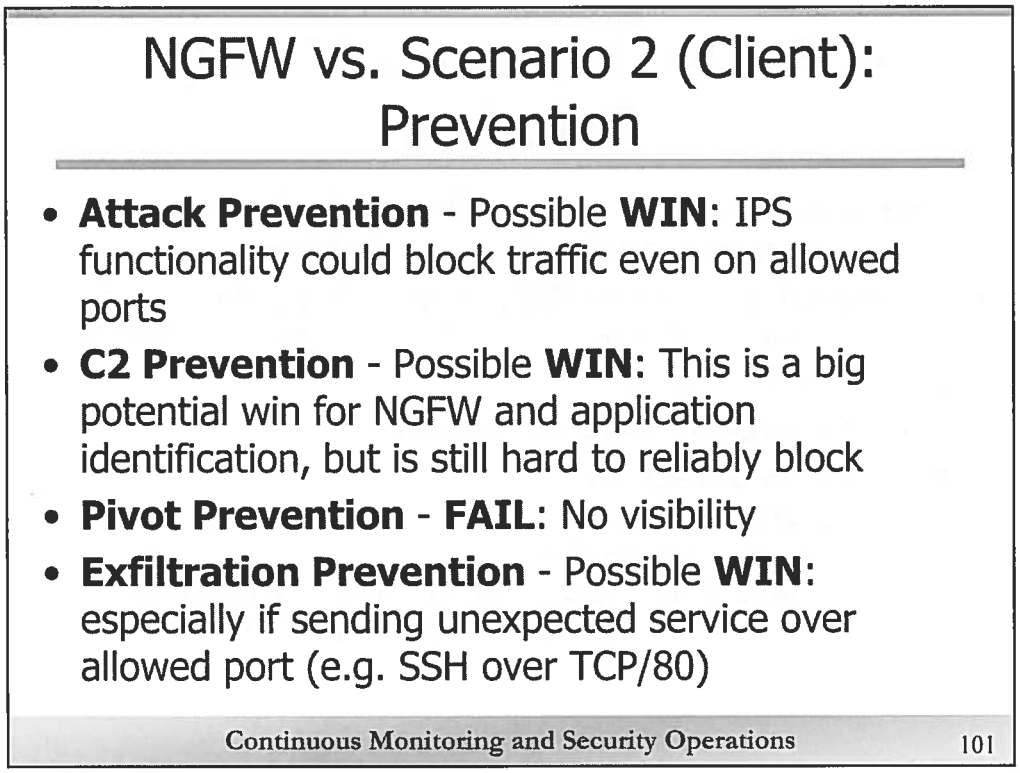
NGFW vs. UTM

While we have been contrasting NGFW and SI firewalls, some of you in the back of your head were thinking, what about UTM, or Unified Threat Management. UTM technologies predate NGFW, and the most compelling characteristic of the UTM device was that significant and bolstered services that had often been out of reach for many organizations were now available in one appliance.



NGFW vs. Scenario 1 (Web App)

As we have seen before the custom web application attack vector is actually proving the more difficult from a detection and prevention front. The NGFW too will fumble with the custom web application by and large. The attack will almost certainly not be blocked or detected by most NGFW. Likewise, the exfiltration, being across the expected web application channel, will also be unlikely to get caught or blocked.

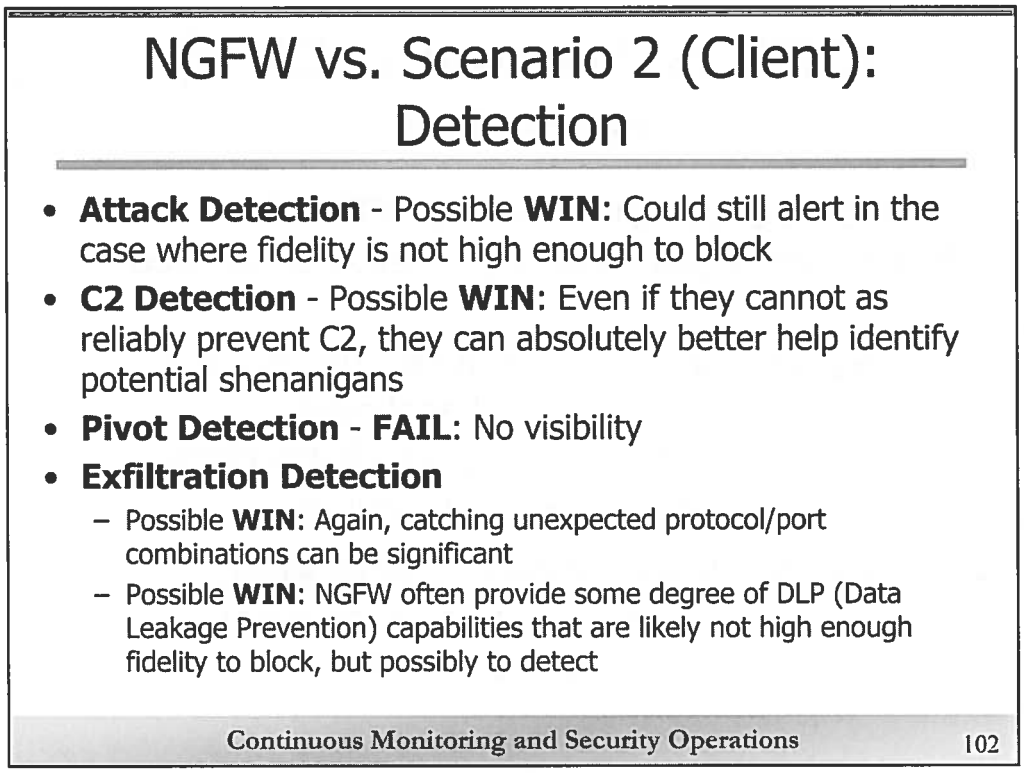


NGFW vs. Scenario 2 (Client): Prevention

The NGFW with its application identification/inspection capabilities can be extremely beneficial. The most significant security boon comes from the ability to potentially identify non-conforming Layer 7 traffic.

On the attack prevention front, the main capability comes from the IPS capabilities afforded us by the NGFW. Not much new is provided on this front beyond pure IPS functionality. The NGFW has no visibility into the pivot.

Data exfiltration prevention capabilities might prove helpful. The main approach would be identification of data being exfiltrated via a protocol over the wrong port, for example IRC over TCP/443 or SSH over TCP/80. Though many NGFW attempt to provide some degree of content-oriented DLP functionality, it likely would not prove high enough fidelity to actually block.

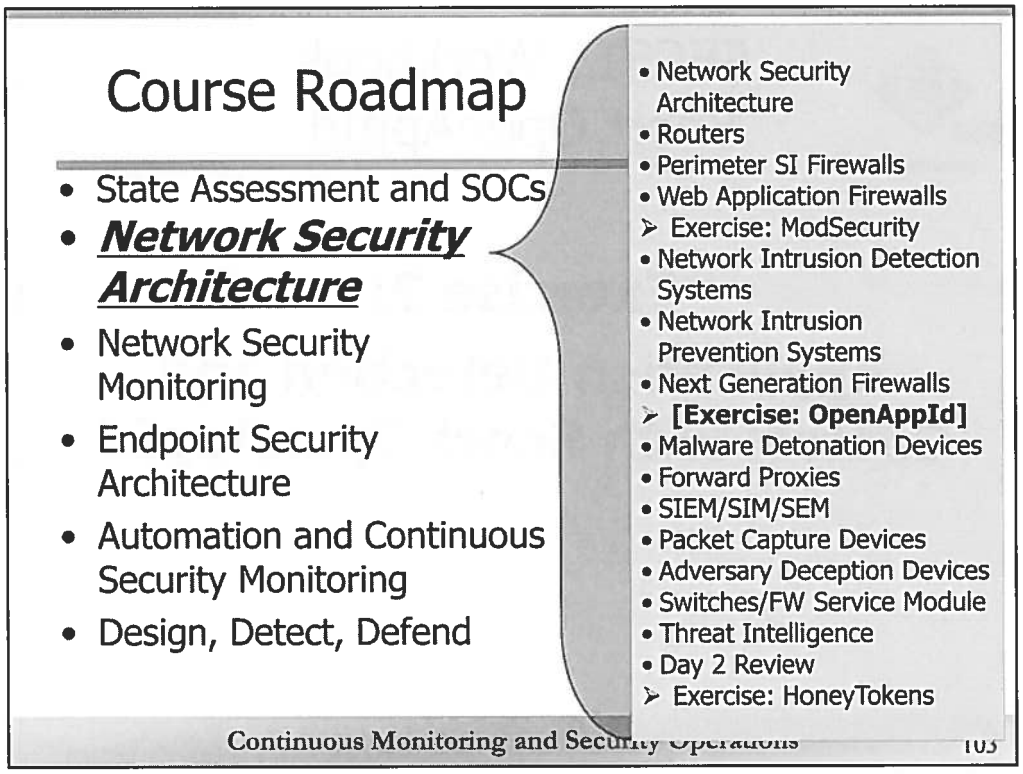


NGFW vs. Scenario 2 (Client): Detection

Again we naturally see that the NGFW provides no capabilities on the pivot front. On the attack detection, we again have capabilities provided by the IPS. However, we should also be able to detect more attacks than those that were prevented, as less high fidelity detects would only be willing to alert rather than block because of the IPS vs IDS impact of false positives (i.e. IPS + False Positive == self-inflicted DoS).

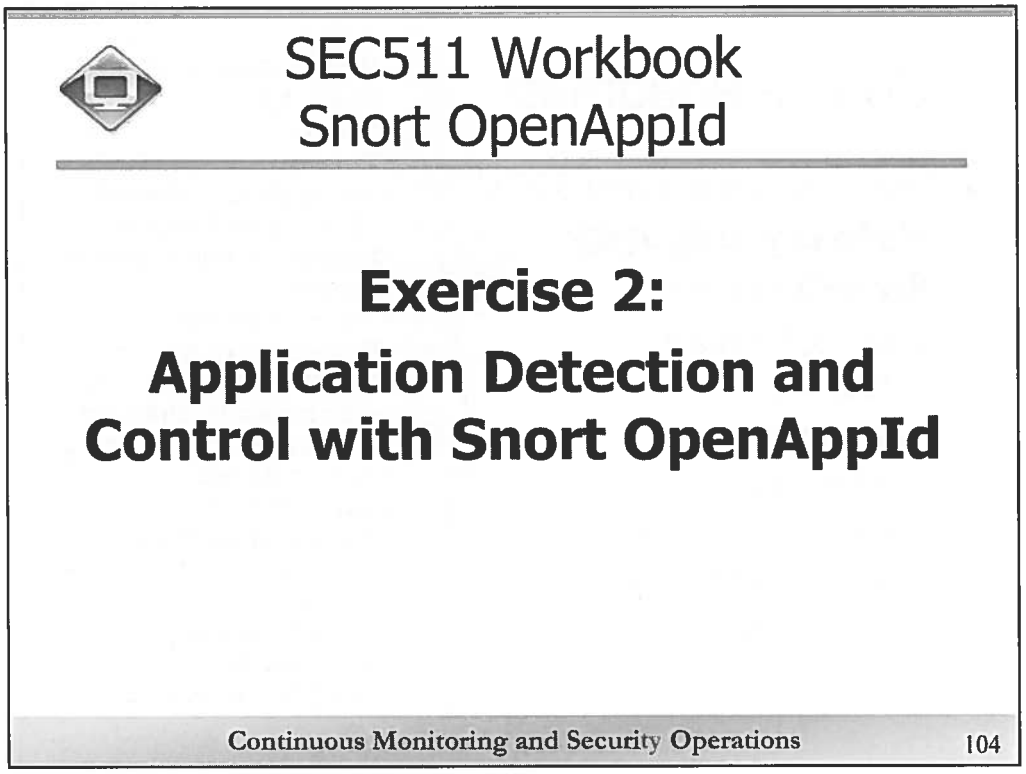
C2 detection again is a big potential win for the NGFW. Depending upon the way the vendor handles detection capabilities, there could be many potential issues that get noted indicating possible C2, but not with enough fidelity to actually block.

As regards to data exfiltration, the same capability mentioned on the prevention front exists, but we have added to it an indicator of DLP (Data Leakage Prevention) functionality that could prove helpful. While most DLP capabilities suggest they can ably prevent the loss of data, for most datasets differentiating legitimate traffic from exfiltration can prove fiendishly difficult. Again, (IPS + False Positive == self-inflicted DoS), which means we might be more likely to get a detect from this functionality even where a block is unlikely.



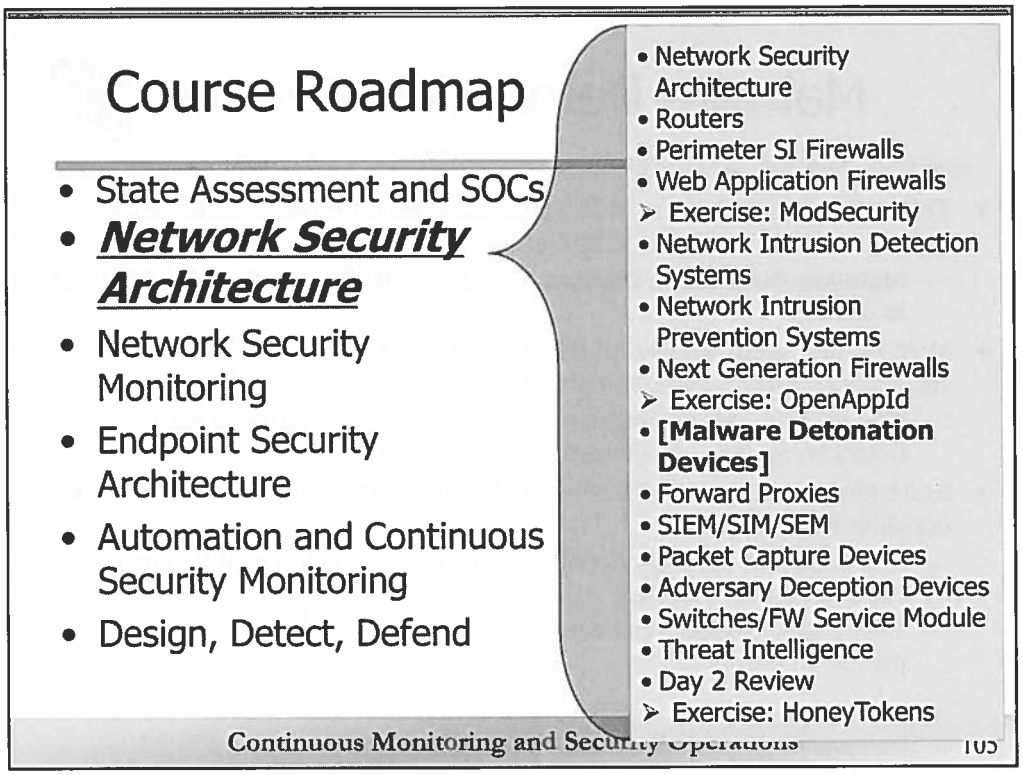
Course Roadmap

The next section is the OpenAppId exercise.



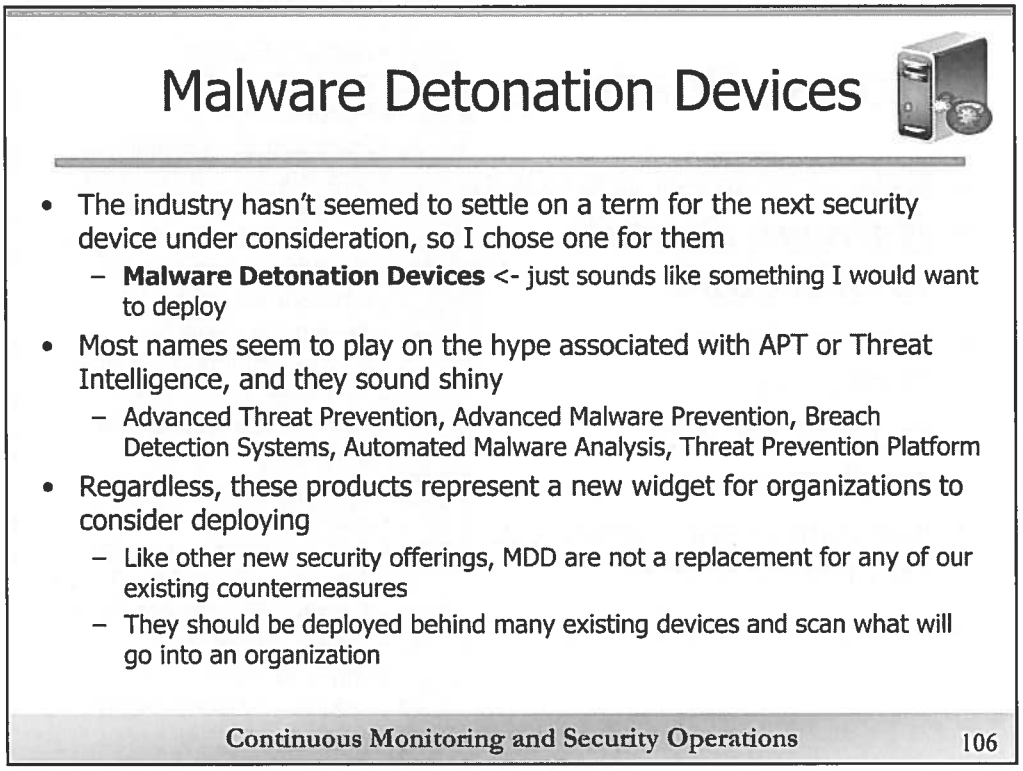
SEC511 Workbook Snort OpenAppId

Please go to the 511 Exercise Workbook, section 511.2-2.



Course Roadmap

The next section is Malware Detonation Devices.



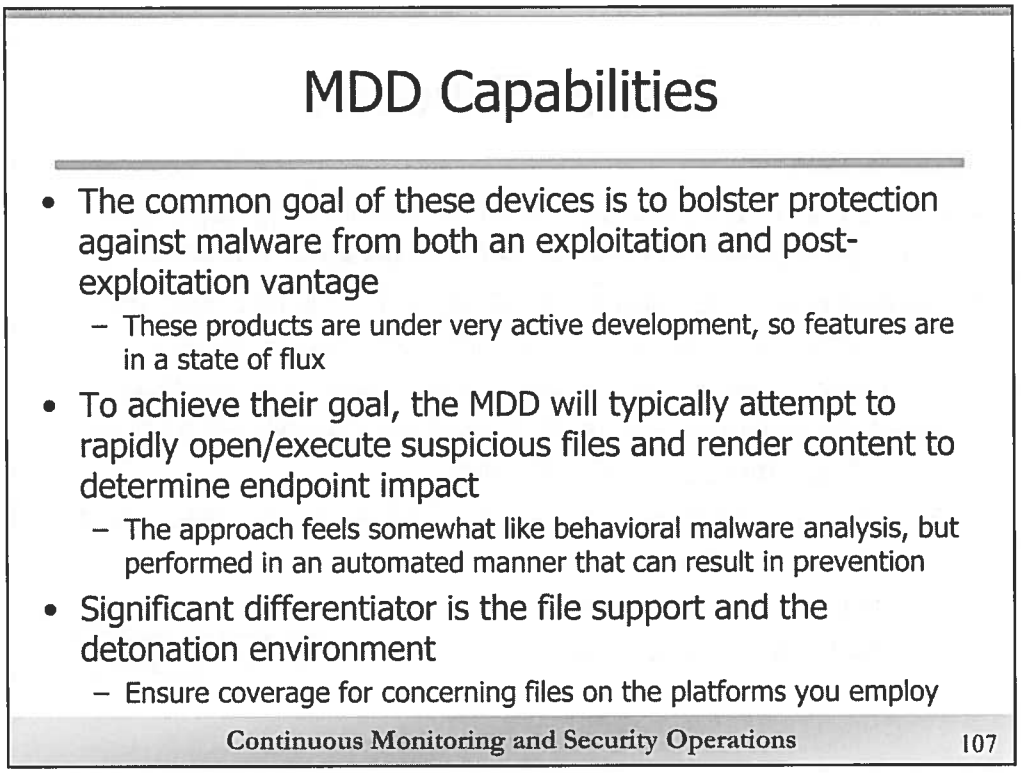
Malware Detonation Devices

One of the most recent devices to be added to the security landscape has yet to find its name, so I decided to give it my own that I think is awesome, and also illustrative: Malware Detonation Device (MDD). To my knowledge, none of the vendors are using this nomenclature, so we can't be accused of preferring a particular vendor. Other terms employed: Advanced Threat Prevention; Advanced Malware Prevention; Automated Malware Analysis; Breach Detection Systems; and more.

Regardless of the name, what does this new shiny device actually intend to do? The primary focus is on taking files and rendering/executing them in advance of passing them to the targets. A JAR file is downloaded. Could be perfectly legit, but it could also be evil. The MDD could, if JARs are supported, render the JAR and see what it actually does before giving it a thumbs up or down.

Please note that though the MDD are shiny and super cool and we have even seen some of them actually deliver on identifying 0-day exploits[1]; they are not a magic bullet that obviates the need for other security controls.

[1] https://isc.sans.edu/diary/FireEye+reports+IE+10+zero-day+being+used+in+watering+hole+attack/17642 (http://cyber.gd/511_229)



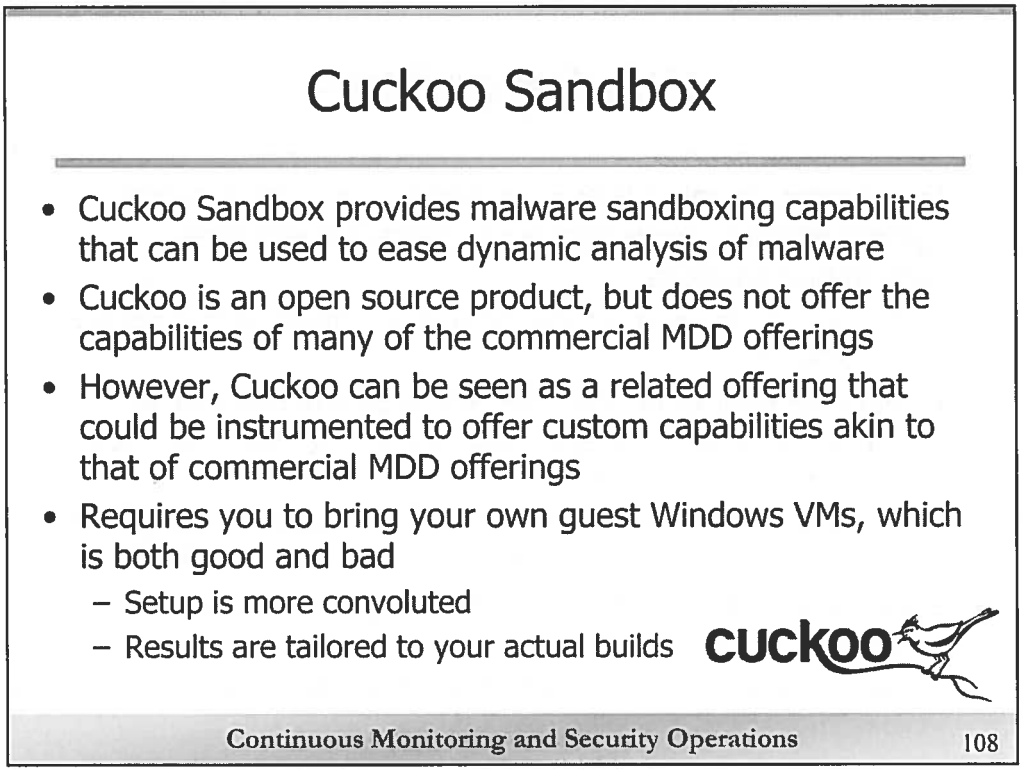
MDD Capabilities

The main emphasis of Malware Detonation Devices is automatically trying to render or execute files before passing them on, or perhaps simply providing a report after analysis.

Effectively MDD is an appliance (or cloud-enabled, big data, buzz word, buzz word) that automatically performs behavioral analysis. This approach has been employed for years in the forensics community, even in an automated fashion. Lenny Zeltser (GSE #2) has published a list of tools that perform automated malware analysis[1].

What makes MDD cool is the ability to perform the behavioral analysis in an automated, non-interactive fashion with potentially enough fidelity to determine whether there is a significant threat to the environment.

[1] http://zeltser.com/reverse-malware/automated-malware-analysis.html (http://cyber.gd/511_230) **QR**

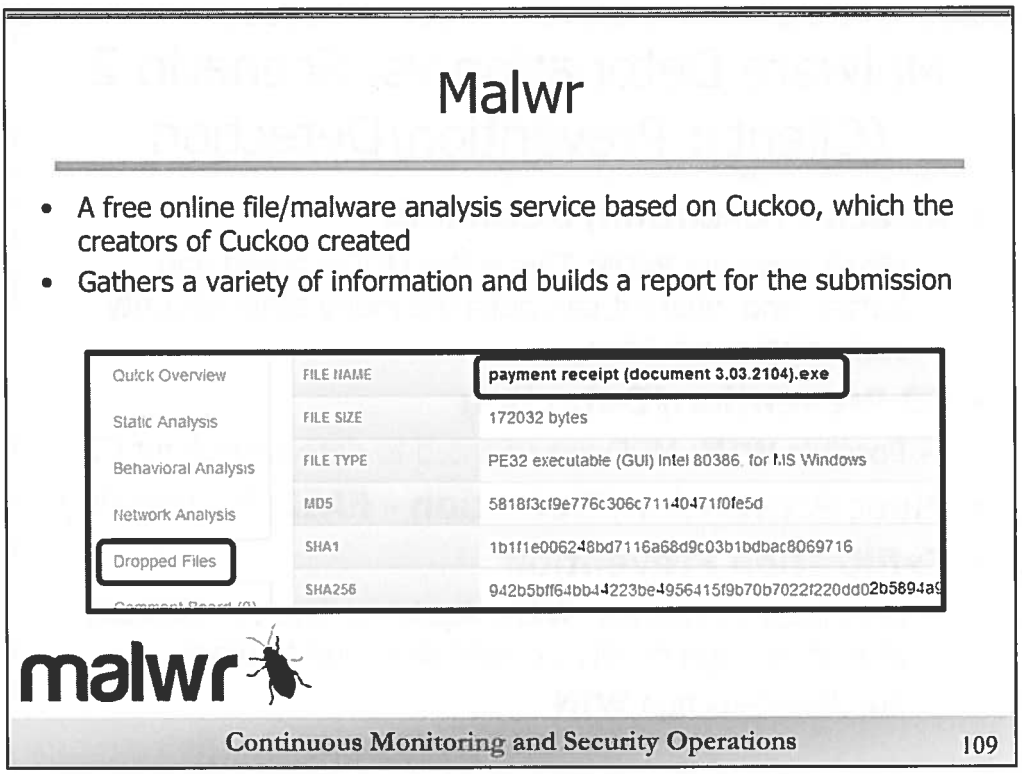


Cuckoo Sandbox

While not comparable to most commercial offerings, Cuckoo Sandbox[1] affords us an open source dynamic analysis platform. Before we had the big vendors, Cuckoo already existed to perform behavioral analysis and spit out reports for us.

There are a number of other free services for performing automated behavioral analysis of files that you upload, Cuckoo is especially interesting because it is open source and can be hosted in your organization.

[1] http://www.cuckoosandbox.org/ (http://cyber.gd/511_231) **QR**



Malwr

If you don't have the stomach for building your own Cuckoo right off the bat, or you want to get a sense for what it would look like once you were successful in creating it, then you can leverage Malwr. This service is provided for free online[1]. Note" it was taken down in July 2014 due to resource issues. A post on 8/22/2014 stated it was coming back online.

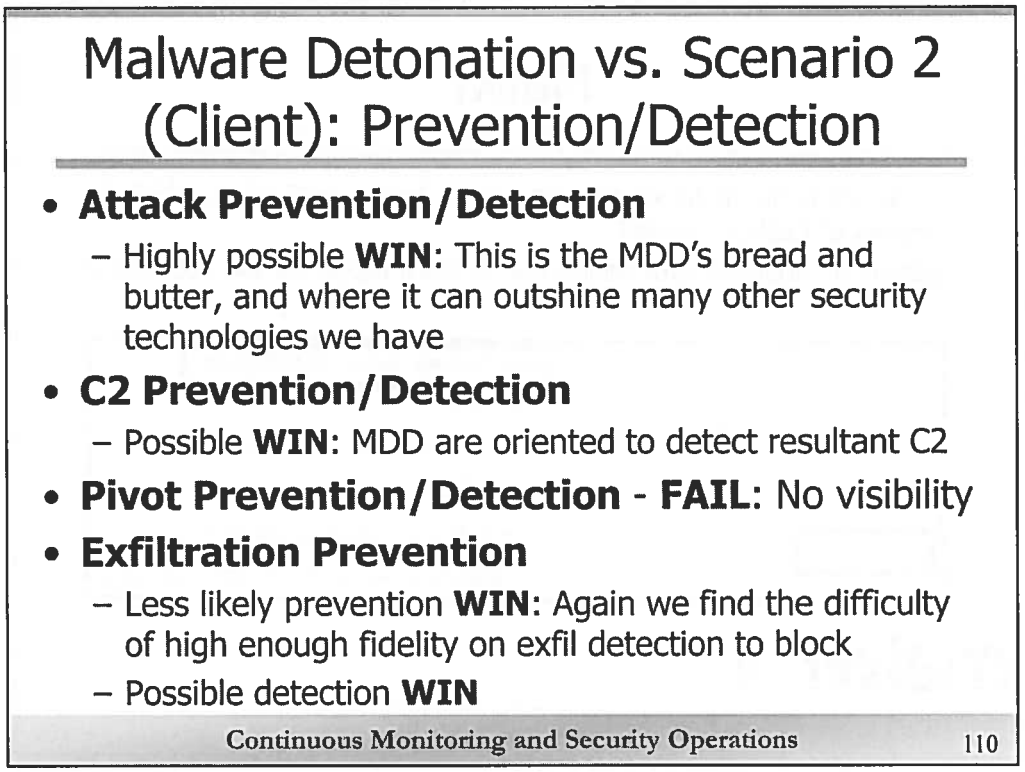
If you like what you see, then definitely check out the paper in the SANS Reading Room by Jim Clausing, GSE #26 (@jclausing), "Building an Automated Behavioral Malware Analysis Environment using Open Source Software[2]." Though his setup is based on Joe Stewart's Truman, the process will certainly put you in the right mindset even if you go with a Cuckoo based configuration.

Another recent paper that focuses on more than just dynamic analysis is from another GSE, they seem like a smart bunch ;), Wylie Shanks, GSE

[1] https://malwr.com (http://cyber.gd/511_232) **QR**

[2] http://www.sans.org/reading-room/whitepapers/tools/building-automated-behavioral-malware-analysis-environment-open-source-software-33129?show=building-automated-behavioral-malware-analysis-environment-open-source-software-33129&cat=tools (http://cyber.gd/511_233)

[3] https://www.sans.org/reading-room/whitepapers/incident/enhancing-incident-response-forensic-memory-analysis-malware-sandboxing-techniques-34540?show=enhancing-incident-response-forensic-memory-analysis-malware-sandboxing-techniques-34540&cat=incident (http://cyber.gd/511_234)

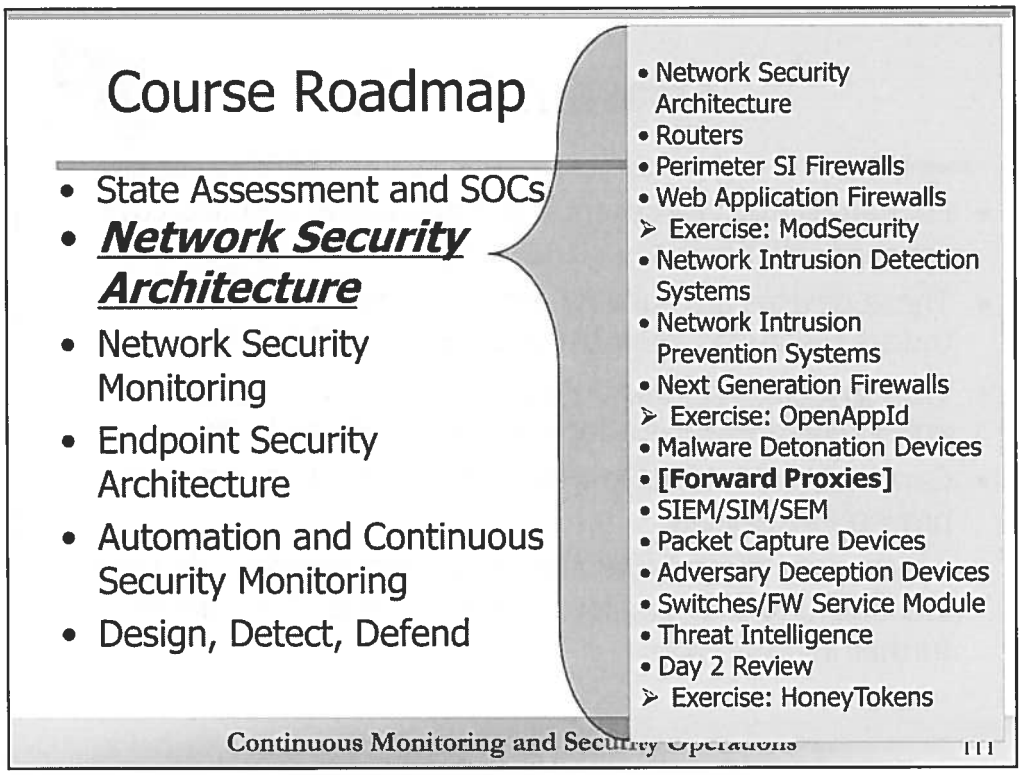


Malware Detonation vs. Scenario 2 (Client): Prevention/Detection

The MDD could significantly bolster prevention of client-side attacks that are otherwise quite difficult to prevent. One of the overt challenges of antimalware, and to a lesser extent IPS, is their reliance upon some reason to look for malicious activity in the first place, typically codified in the form of a signature.

C2 is another strong point for MDD, as part of the analysis intends to see whether there is any resultant activity that could be characterized as C2.

With regards to exfiltration, we again find a similar problem as discussed previously, which is that high fidelity detection of illicit data exfiltration is elusive in many cases. The difficulty means that devices are unlikely to automatically prevent the data exfiltration. However, they could still alert on the possibility, aiding detection.



Course Roadmap

The next section is Forward Proxies.

Forward Proxies

- Forward proxies represent a key preventive and detective capability that has been available for numerous years
- These devices are suitably positioned to see and potentially thwart client-side exploitation as well as C2 traffic
- They are also well-positioned to help identify rogue or policy-violating applications and abuse of privilege
- Can further be useful in a data exfiltration detection and prevention capacity
- Another significant potential use case of proxies is in the identification of anomalous traffic patterns that warrant further investigation

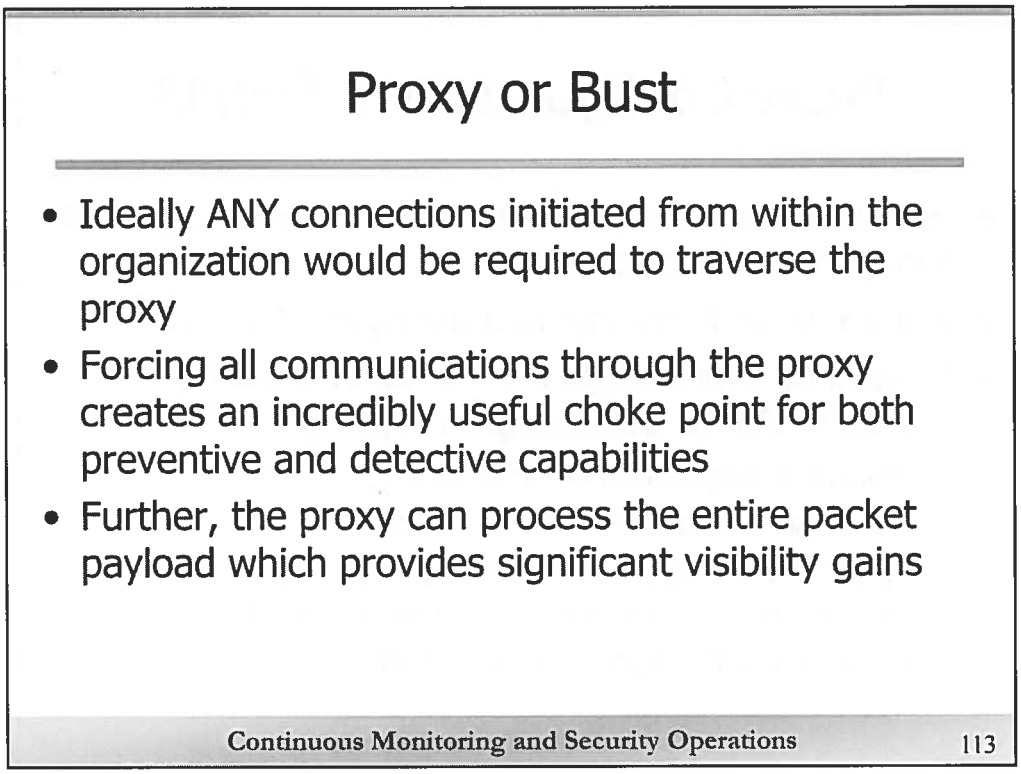
Continuous Monitoring and Security Operations 112

Forward Proxies

An essential construct for security has been that of the proxy. A proxy creates a choke point, whether it be a single appliance that fronts a web server farm (load balancer → reverse proxy) or single, possibly transparent, server/appliance that outbound traffic is funneled through.

While there are some potential performance benefits, especially in the case of the proxy being a purpose-built appliance, the primary security benefit comes from the choke point itself and the opportunities to perform serious inspection and access control at one location and have far reaching, perhaps enterprise-wide, impact.

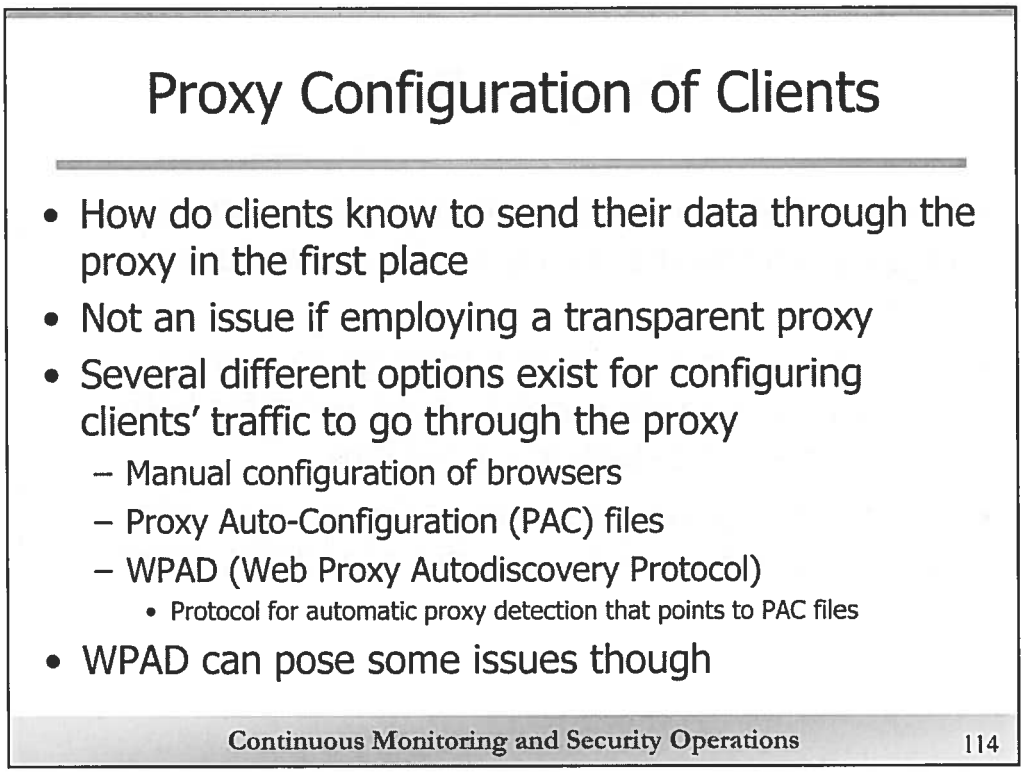
Configured properly, forward proxies, those acting as the upstream choke point for clients, are suitably positioned to scrutinize the majority of attacks and C2 traffic.



Proxy or Bust

We stated this goal earlier today when discussing our firewall rulebase. In particular, we were considering what an appropriate egress policy would look like. We suggested that all traffic moving from the internal network out to the Internet would be forced through a proxy of some kind to gain from the opportunities presented by the choke point.

Most importantly, we need to ensure that all clients must have any outbound communication proxied. This actually helps us on multiple fronts. The benefits of the choke point have already been discussed. However, an additional benefit is that if all outbound traffic from clients can be safely assumed to traverse the proxy, then how do we characterize traffic trying to reach outside directly from the clients themselves? At best , and early on the most likely answer, this is a misconfiguration. However, it could also be an indicator of compromise or a policy violation.

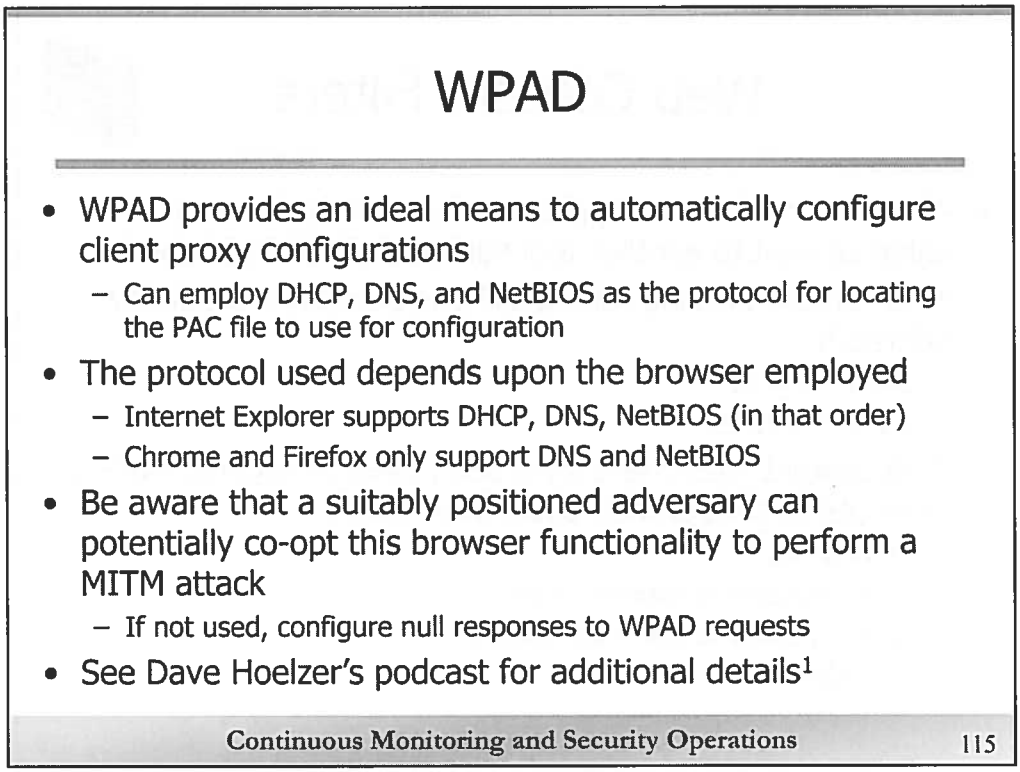


Proxy Configuration of Clients

In order to gain the security benefit of the forward or client proxy the browsers must either be forced through the proxy or configured to direct traffic through the proxy. There are several different options for configuration of the clients.

The most obvious approach to configuration is simply to manually configure browsers to point to the corporate proxy. While conceptually simple, this approach has some downsides. Most importantly, if the endpoint is a mobile device, it would likely require a different proxy configuration when connected to the enterprise network versus say a hotel network.

Another approach that is more scalable is to employ the use of PAC files. These are Proxy Auto-Configuration files that are written in JavaScript and can employ complex logic to easily support many varied configurations. WPAD, Web Proxy Autodiscovery Protocol, provides a means to have clients query the network to find out where a PAC file is to use.



WPAD

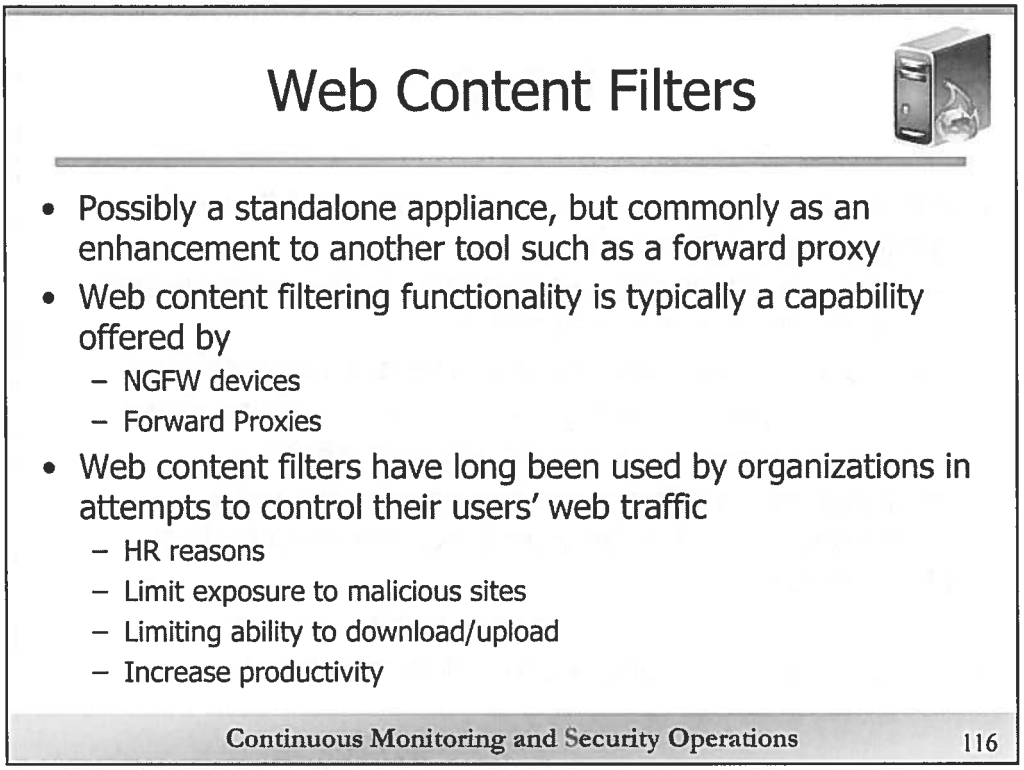
The clever WPAD functionality allows for automatic configuration of clients. This auto-configuration is achieved by having the browser ask the network where it should look for a PAC file. This network query is performed using DHCP, DNS, and NetBIOS, in that order, seeking a pointer to a PAC file. Whether each of these protocols is supported depends upon the browser being employed.

Internet Explorer supports all three methods of discovery. All browsers across operating systems will typically be able to leverage DNS. On Windows, Firefox and Chrome will employ DNS and NetBIOS, if NetBIOS is supported on the underlying OS.

Adversaries have developed a means to co-opt this WPAD functionality by providing their own response the WPAD requests if we do not provide our own. Using this method, suitably positioned adversaries could launch a MITM attack against clients.

Configuring WPAD DHCP/DNS/NetBIOS null responses if not actively being used is highly recommended.

[1] http://auditcasts.com/screencasts/17-man-in-the-middle-web-attacks-using-wpad (http://cyber.gd/511_244) **QR**

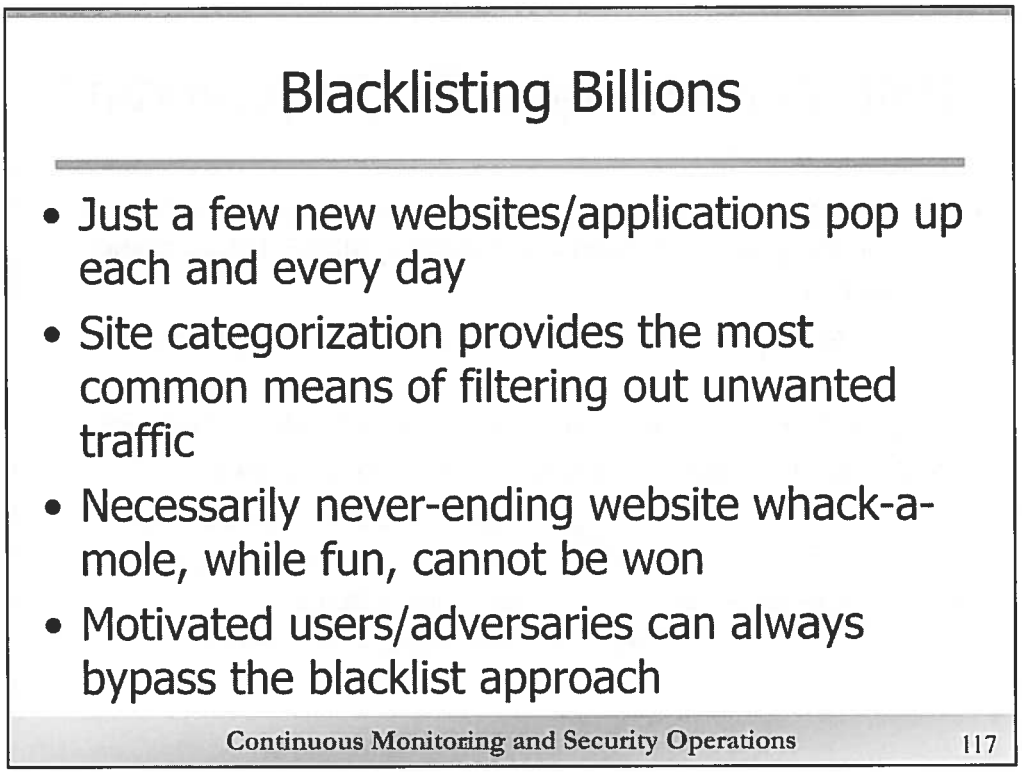


Web Content Filters

Though a forward proxy does not have to include web content filtering capabilities, they very often do. Note, however, that the web content filtering functionality could be a standalone device in its own right. Also, we see web filtering instrumented into UTM and NGFW devices as well.

Though we will consider primarily the cyber defense aspects of web content filtering, there are additional reasons that organizations employ web content filters. HR reasons and increased productivity are also additional potential benefits of this approach.

For our purposes the primary idea is to reduce the risk associated with users accessing content via the Internet, and to also gain significant visibility into potentially identifying compromised hosts.



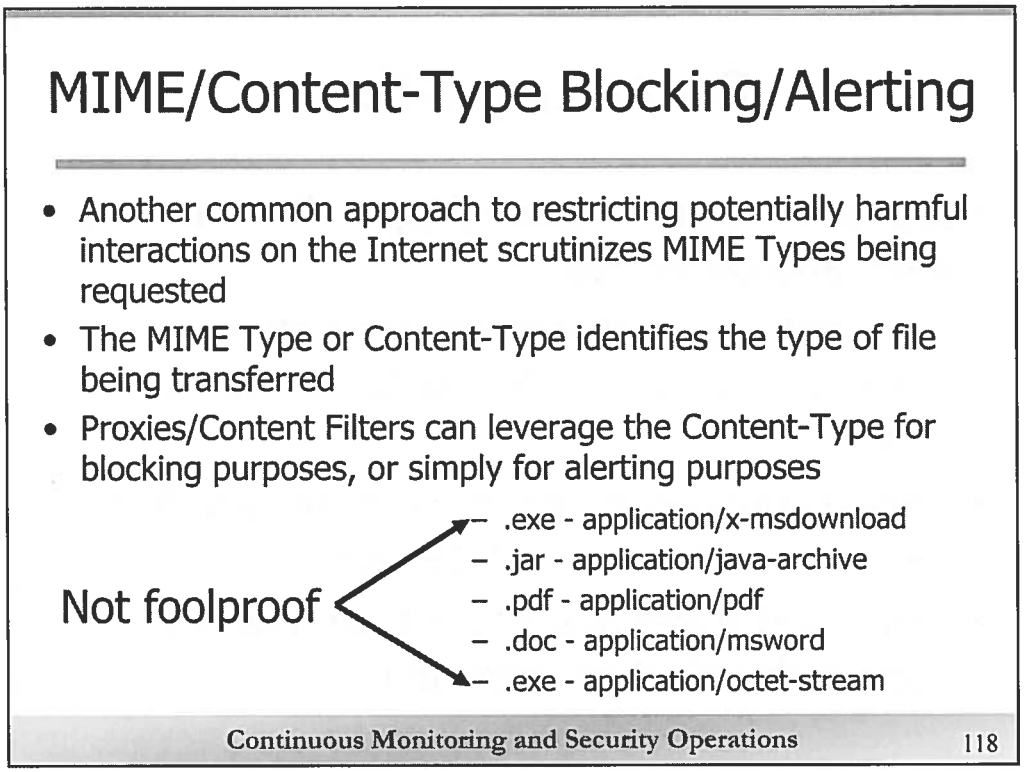
Blacklisting Billions

Most folks consider the primary benefit of the web content filter to be in blocking access to certain sites and categories of sites. Naturally blocking access to sites that would compromise systems could provide benefits, but additional categories such as adult sites, hate speech, etc. might be blocked due to the potential liability associated with what is sometimes termed a "hostile work environment."

Sounds great, but how do we actually pull this off. There are just a couple of new sites that popup each and every day. Can someone really categorize all of them? Not quickly that is for sure. This is necessarily a never-ending update process.

An additional question, how hard is it to bypass a blacklist for a motivated user or adversary[1]? Not that difficult at all.

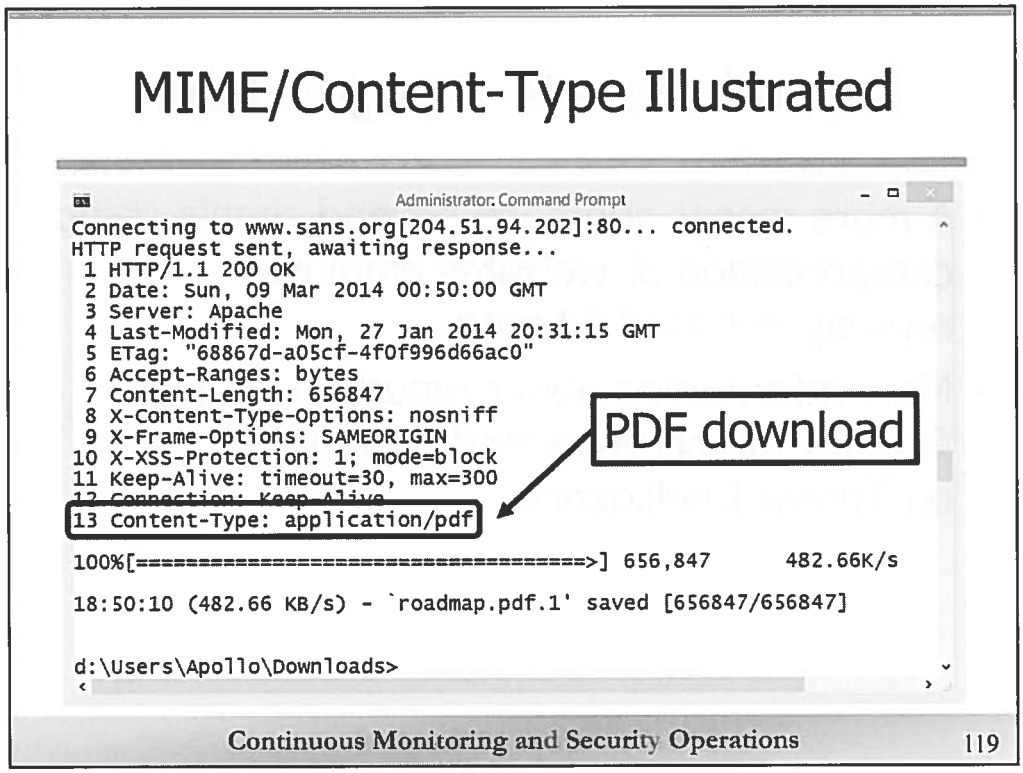
[1] http://lmgtfy.com/?q=proxy+bypass (http://cyber.gd/511_235)



MIME/Content-Type Blocking/Alerting

Beyond just blocking via URL and website categorization another approach to web content filtering is to block access based upon MIME or Content-Type. When downloading content via HTTP, a Content-Type header is provided that identifies the type of file being delivered. This concept originated with, and is still employed by, SMTP as a means of sending content other than straight ASCII plain text.

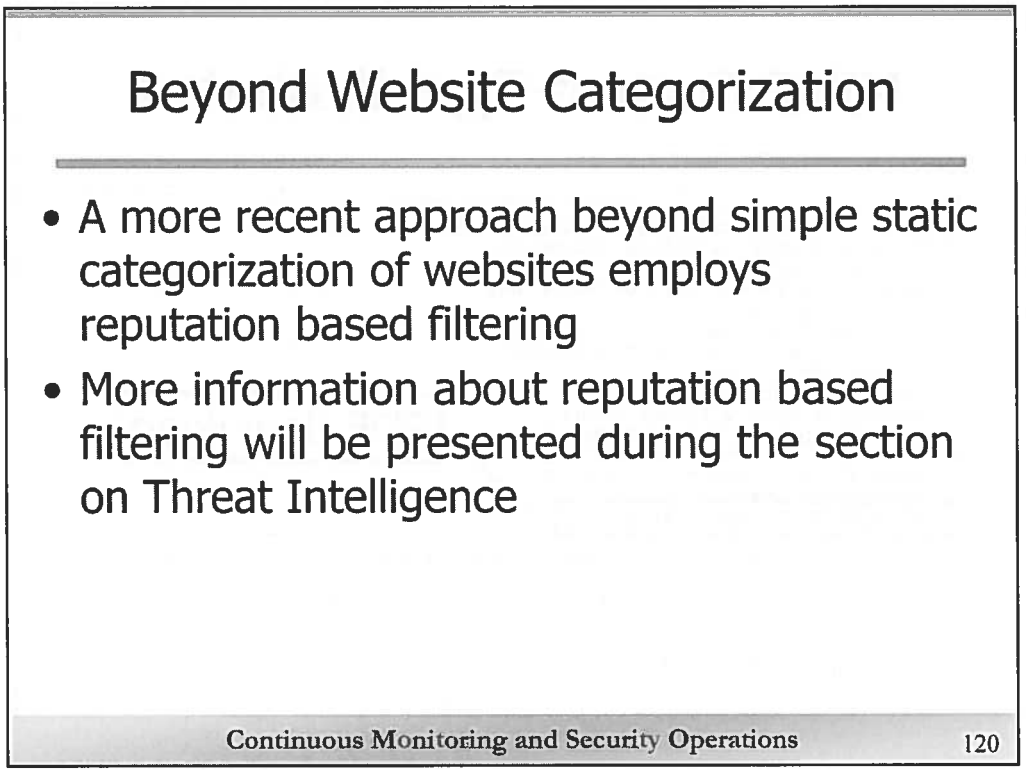
Proxies can look for these headers to identify types of content the might warrant additional scrutiny (in say an automated dynamic analysis sandbox) or just get blocked without scrutiny.



MIME/Content-Type Illustrated

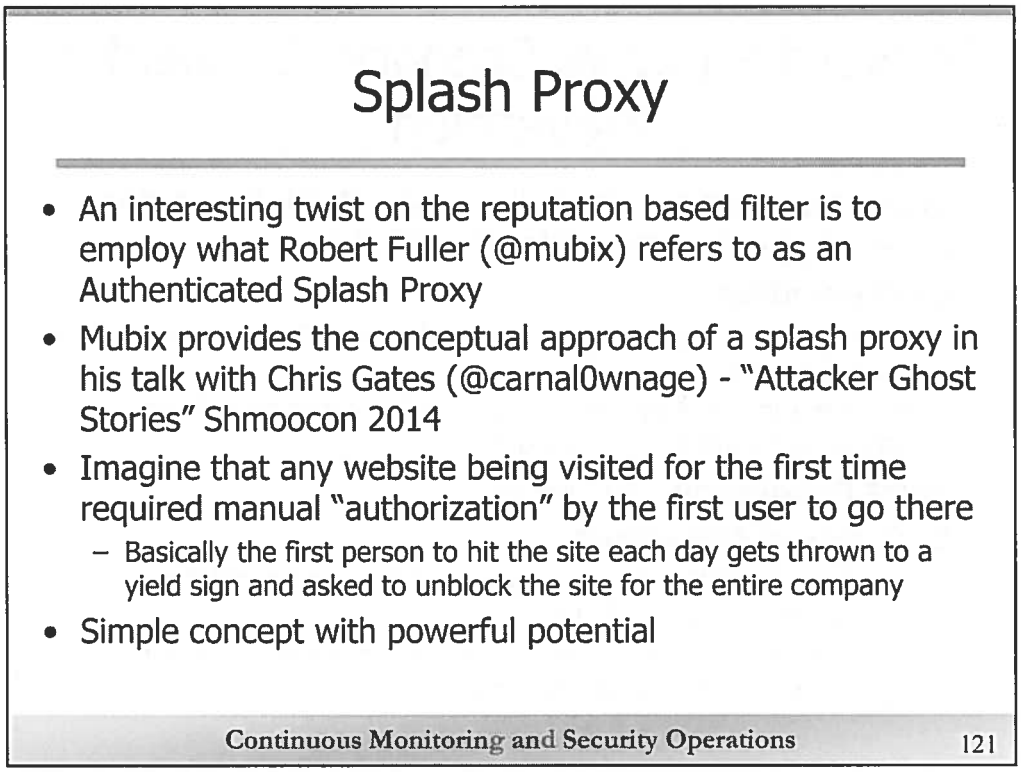
Above we see a sceenshot of using Wget to download a file and showing the headers. Here we see the Content-Type header indicates application/pdf. Not terribly surprising that the file then is roadmap.pdf. Numerous lists of known MIME/Content-Types are available[1], but be careful as many of them only include IANA defined MIME types rather all those that might be in wide use despite IANA.

[1] http://reference.sitepoint.com/html/mime-types-full (http://cyber.gd/511_22) **QR**



Beyond Website Categorization

Reputation based filtering is a recent approach that has started to find inclusion in a wide array of security products, including proxies and web content filters. Additional information will be provided on reputation based filtering during discussion of threat intelligence later.



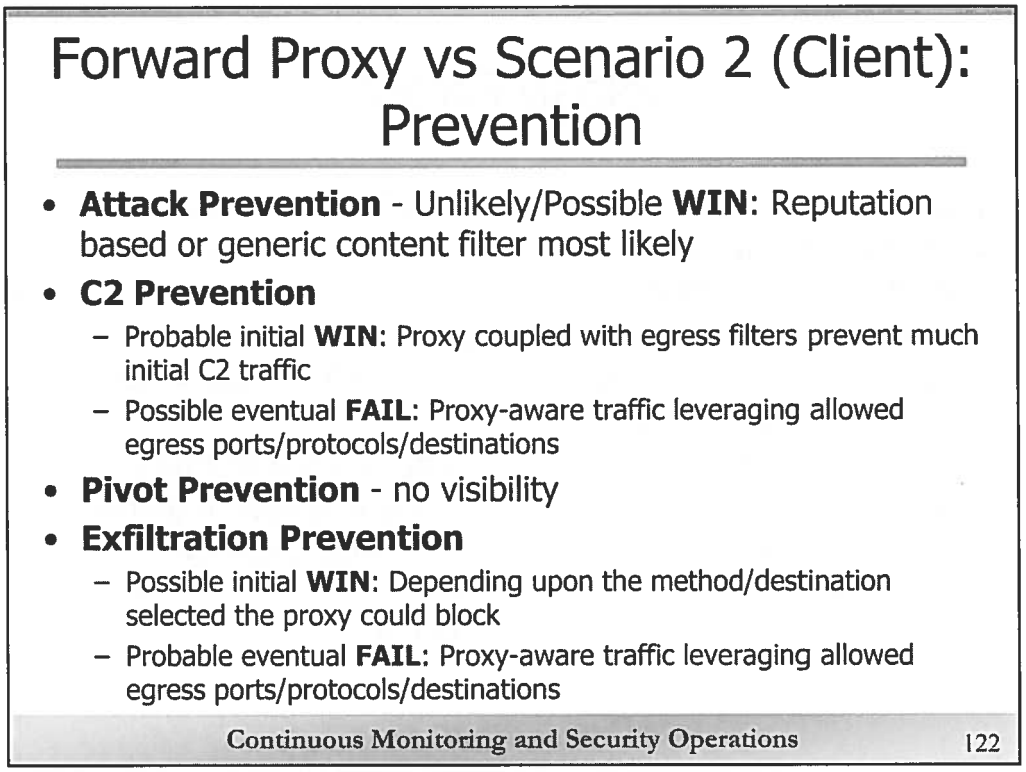
Splash Proxy

This is a quick proxy idea that I first heard about with Rob Fuller's (@mubix) 2014 Shmoocon 2014 talk, "Attacker Ghost Stories.[1]" The idea brings together the concepts of a captive portal and reputation filter together. In this case rather than sourcing externally a reputation source, you are leveraging your employees to provide their sense of reputation.

Basically his idea is the first time someone in the organization hits a site each day, the user would be required to submit a form, likely in the form of clicking a button, to tell the proxy that a site is ok. This would mean the first user to hit http://www.google.com would get a splash page requiring them to click the button to say this site is ok, for everyone in the organization.

This clever little shim would break a lot of C2 persistence mechanisms. Further, it will (hopefully) make users think twice before going to a less than reputable site. Further, if they are getting phished and clicked on a link that doesn't point where they thought, it could provide an undo button.

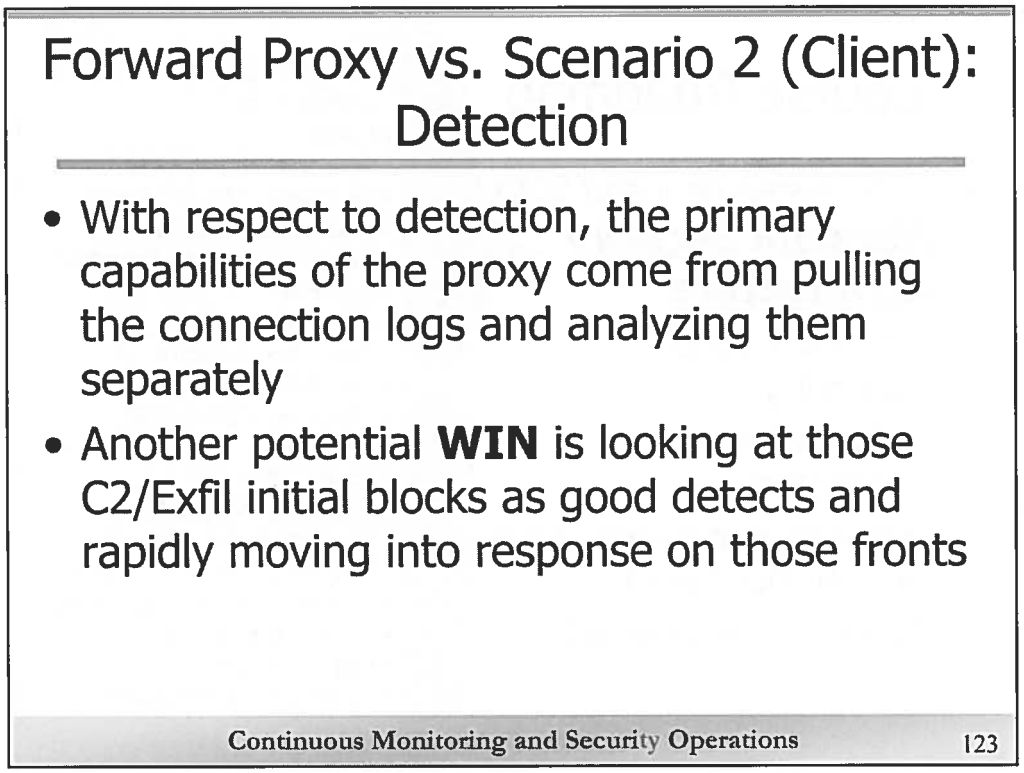
[1] https://archive.org/details/ShmooCon2014_Attacker_Ghost_Stories (http://cyber.gd/511_236) **QR**



Forward Proxy vs Scenario 2 (Client): Prevention

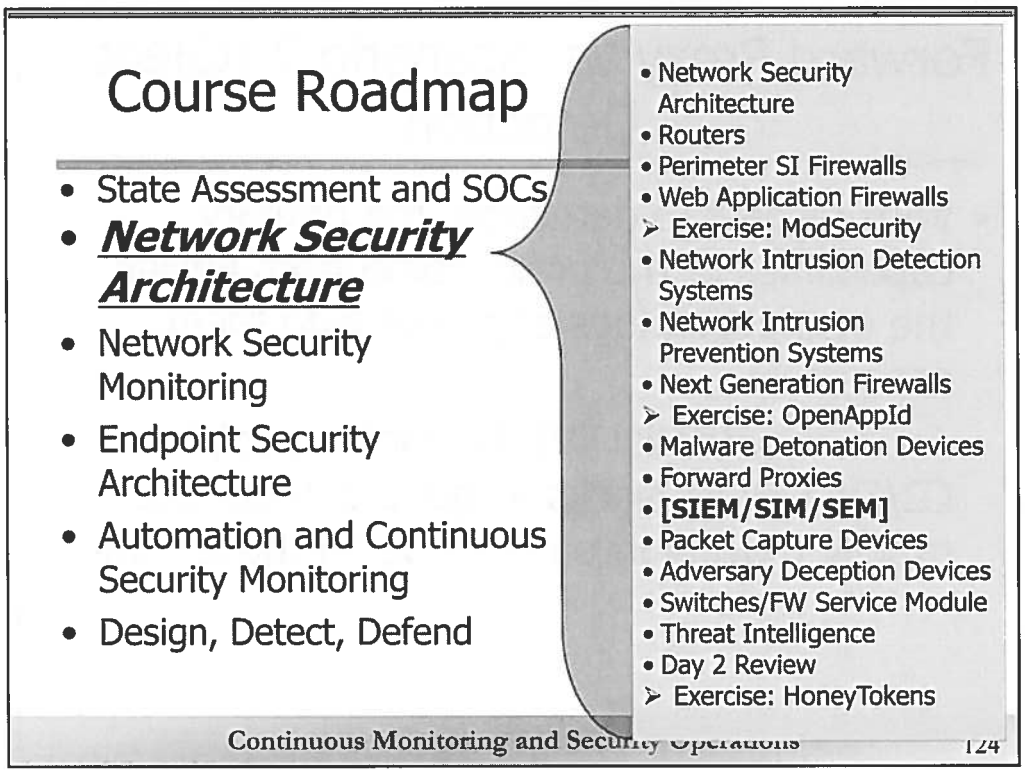
The proxy can be a significant adjuvant to security. Attack prevention could be viable primarily due to reputation or content based filtering of traffic.

If coupled with a strong egress policy, the C2 and Exfiltration prevention performance is better than the attack prevention capabilities. Very likely the initial C2 and exfiltration both could leverage ports/services that are not proxied, and destinations that are possibly blocked by reputation. So, when coupled with a strong egress policy, the proxy can prove quite effective.



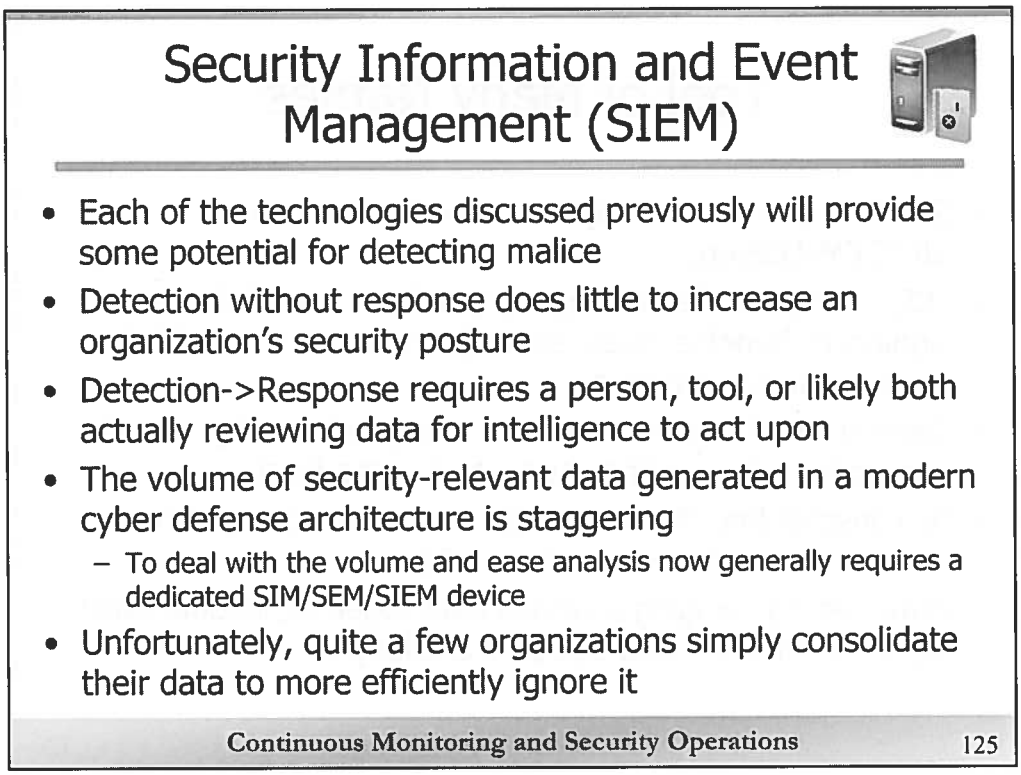
Forward Proxy vs. Scenario 2 (Client): Detection

Detection aspects of the proxy generally come from us parsing the information afforded by the choke point with another tool/analysis engine. However, another aspect that must be considered is leveraging the proxy blocks as potential detects that can lead into rapid response.



Course Roadmap

The next section is Security Information and Event Management.

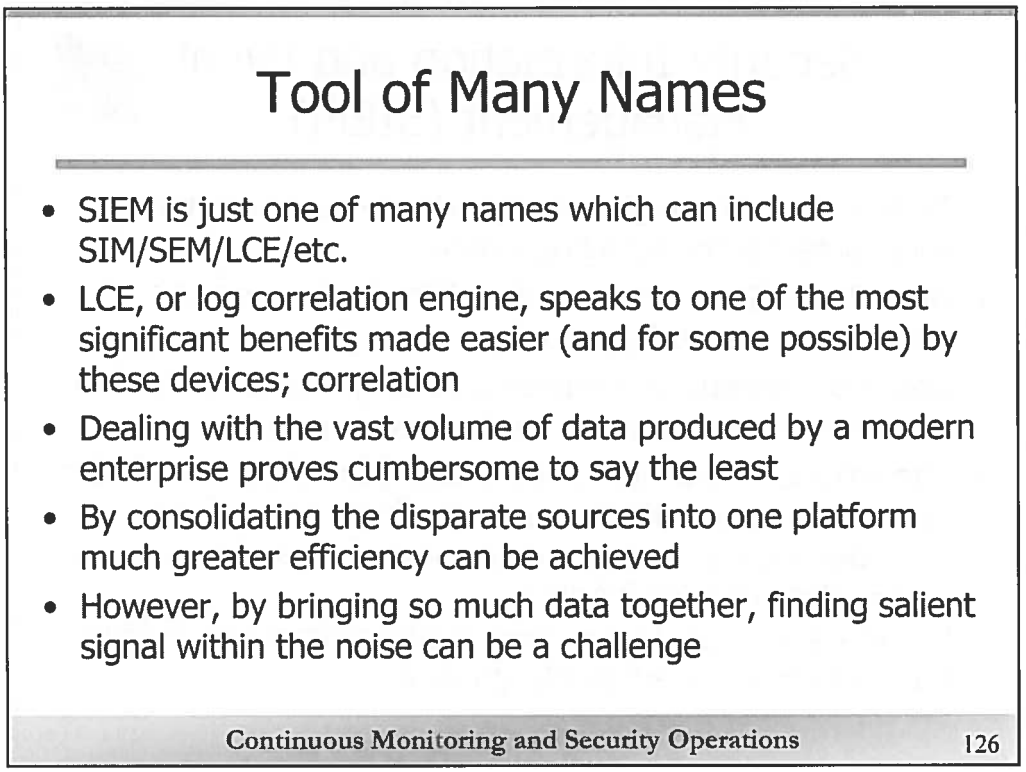


Security Information and Event Management (SIEM)

Many of the technologies discussed in today's content have provided some degree of detective capabilities, even if they were not overtly detective devices, as most were not. Just because those devices COULD allow us to detect the adversaries tactics does not mean that we WOULD detect them. Stop and think about when you have read details about an organization having been breached. We hear explanations about what happened, how it happened, and sometimes how long it was happening.

Or simply consider Mandiant M-Trends and Verizon DBIR, discussed on day 1, which routinely suggest that months often pass before an organization realizes that they have been compromised, usually because someone else tells them.

Consider for a minute what this means. How could Mandiant and Verizon determine how long an organization had been compromised? In each of the cases reviewed, there was sufficient evidence available for the IR/Forensics folks to effectively reconstruct events. This signals to me that the data necessary for detection was typically available, but ignored overtly or passively missed.

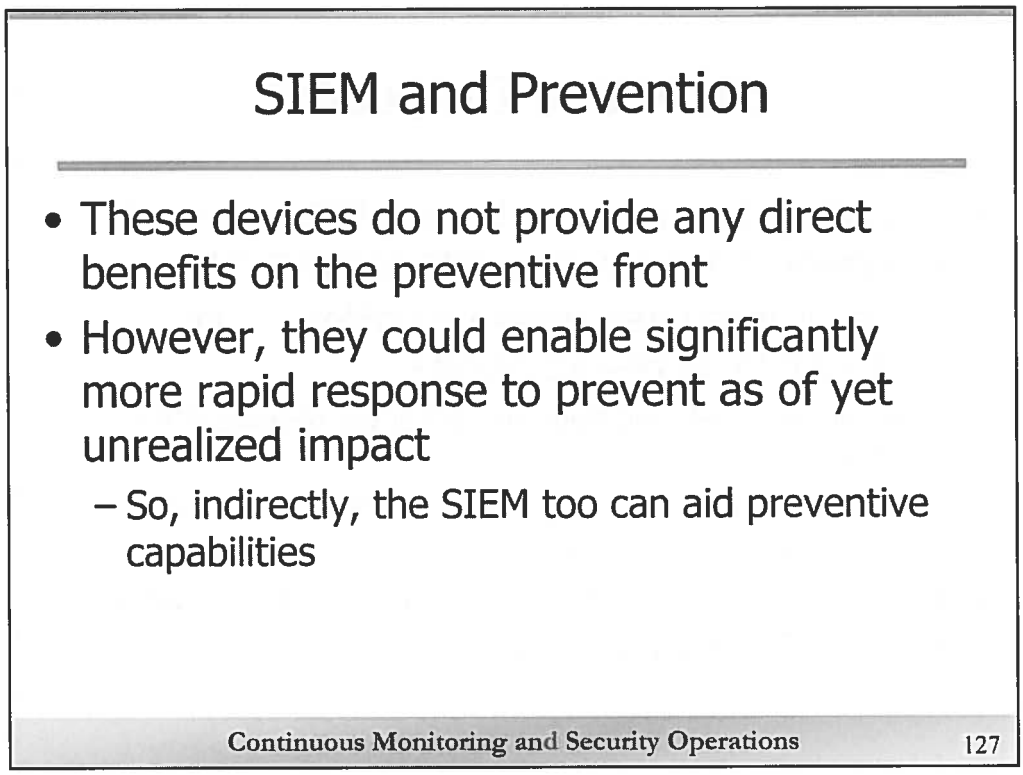


Tool of Many Names

The focus of this section is on leveraging a tool to ease the consolidation and correlation of data from multiple feeds. Be mindful that simply consolidating and correlating does nothing without a skilled analyst on the other end making sense of, prioritizing, and escalating data.

In addition to SIEM/SIM/SEM, you might also encounter LCE, or Log Correlation Engine, which hits on one of the most significant potential values derived by these tools, namely correlation. Generally when organizations are first going down this route their primary goal is to get all of the organization's data into one repository. However, this alone does little beyond allow us to more easily ignore data.

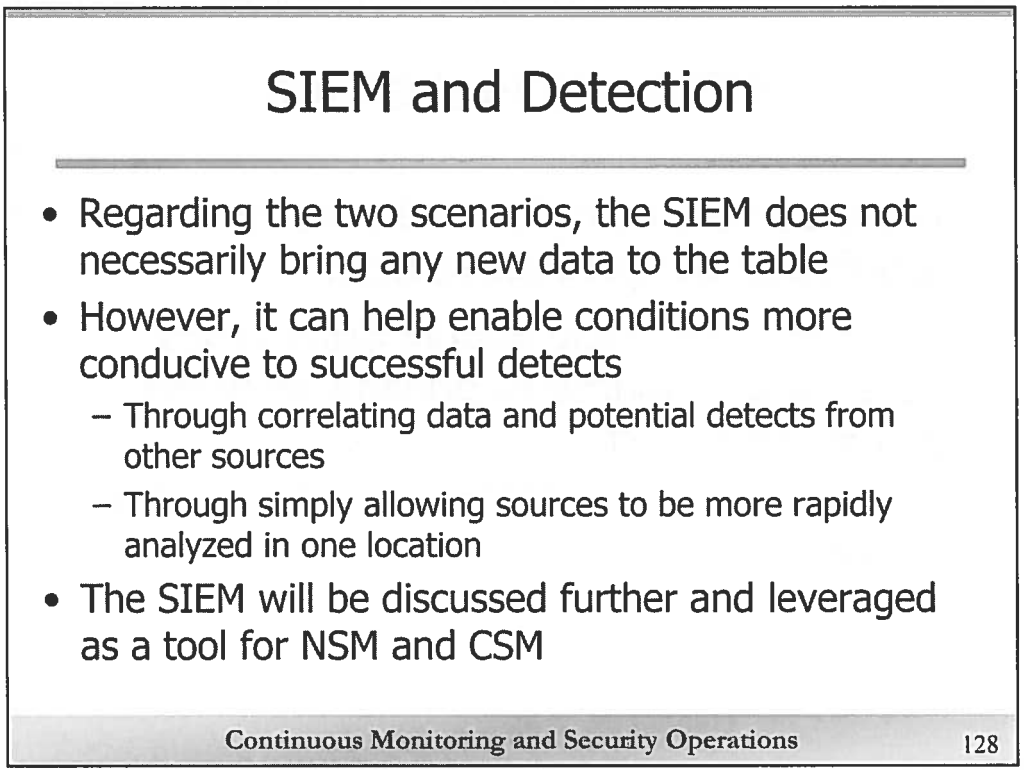
The Hunt Team can help divine signal from the noise that is the logs of the modern enterprise.



SIEM and Prevention

Certainly the SIEM does not provide direct preventive capabilities, as it is an overtly detective tool. However, preventive controls necessarily get bypassed, so we need not put all our efforts on that front.

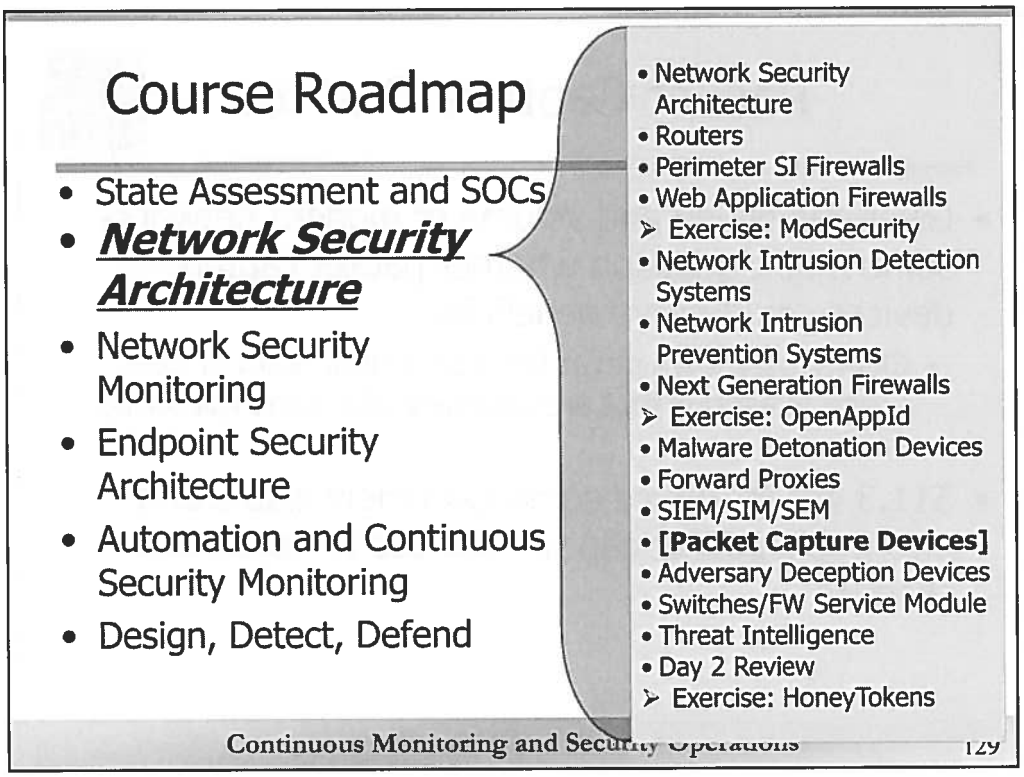
Though SIEM devices provide no direct preventive capabilities, they do indirectly provide substantial benefit at prevention. We are only able to help SIEMs achieve this feat by employing skilled analysts or, better yet, a dedicated Hunt Team to proactively detect and subsequently respond to attacks. Depending upon the nature and timeliness of these activities, we could well prevent future activities that would cause impact.



SIEM and Detection

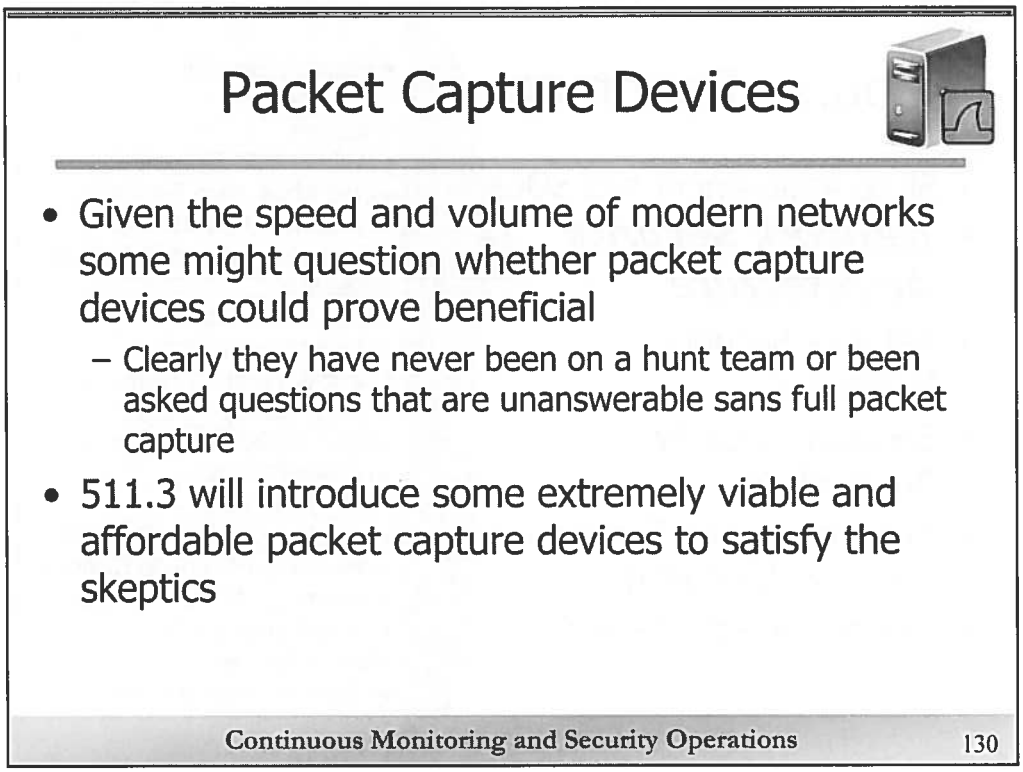
The natural sweet spot for SIEM is certainly oriented toward detection. With respect to our scenario, the SIEM does not bring any new or novel detect capabilities to us, but it could actually increase the likelihood of successfully detecting the data previously mentioned as potential detection WINs.

As stated previously, the SIEM is not the answer by itself. Too long organizations have neglected a key piece of the puzzle, the analysts that will sit on the business end of the SIEM and ultimately determine what, if any, value is gained from the SIEM.



Course Roadmap

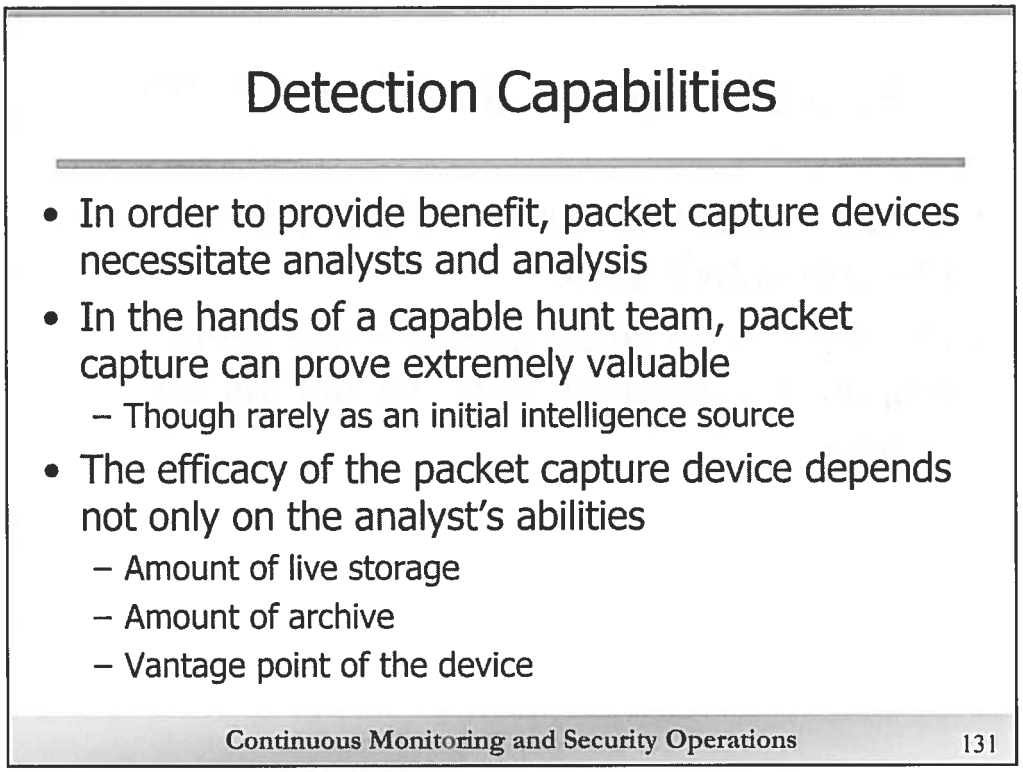
The next section is Packet Capture Devices.



Packet Capture Devices

Though we will discuss packet captures and packet capture devices when we attend to NSM in 511.3, we need to at least make basic mention of these tools.

Often times packet capture devices are dismissed or ignored because of worries about the volume of storage or other concerns. Let's table that consideration for now, as we will directly address it in 511.3. Let's just consider whether having full (or even partial) packet captures would potentially bolster our capabilities. If you have to wonder about the answer, then you have likely not done much analysis. Full PCAP can be hugely beneficial on multiple levels.

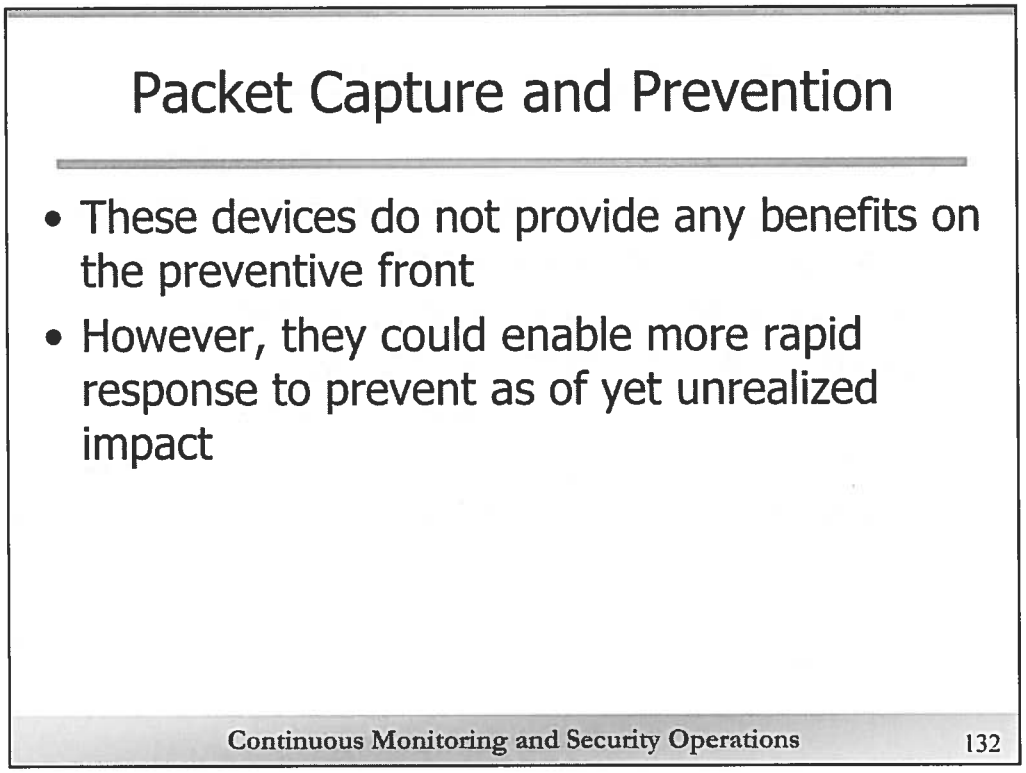


Detection Capabilities

The value of packet captures can rarely be seen without skilled analysts or a hunt team on the business end. The requirement for skilled analysts is typically even greater with packet captures than it is with SIEMs. Even with skilled analysts on staff, packet captures are seldom the starting point for analysis. We do not generally just go browsing around PCAPs and then start looking at our other tools.

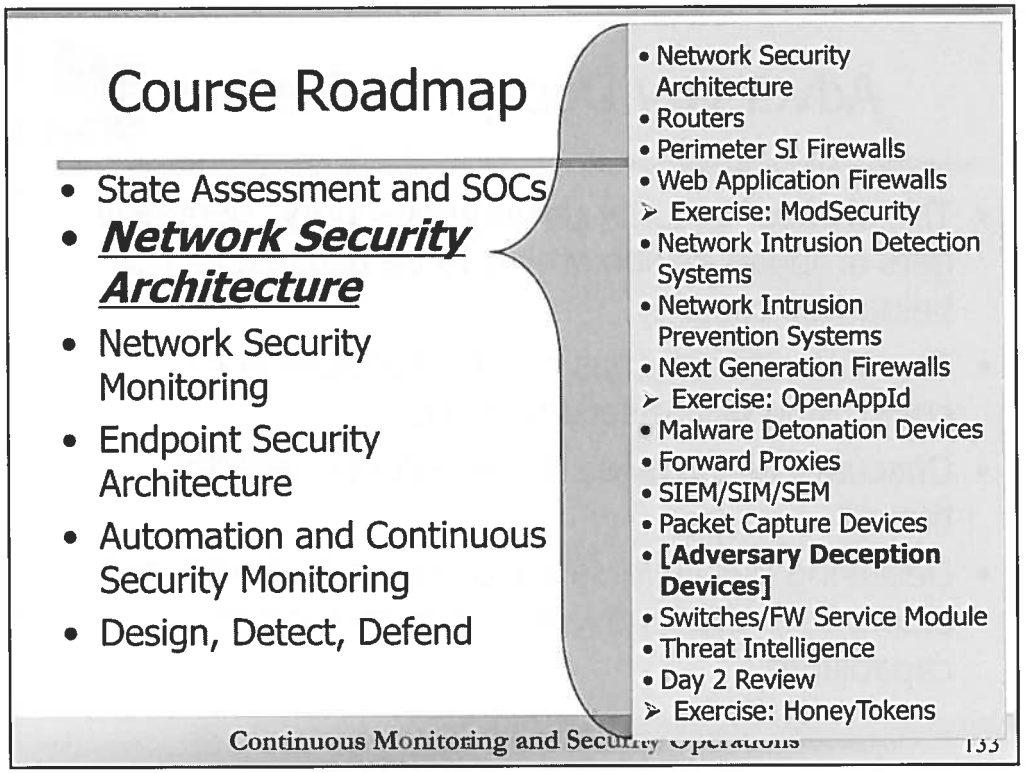
The PCAP is leveraged as the second opinion, and often a more complete picture. The benefit of the packet capture device depends necessarily on the availability of the resultant PCAPs, which in turn depends upon the amount of both online and offline storage.

Another critical determinant on the value of the PCAPs is the vantage point of the packet capture device. Where does the device sit, and what does it see.



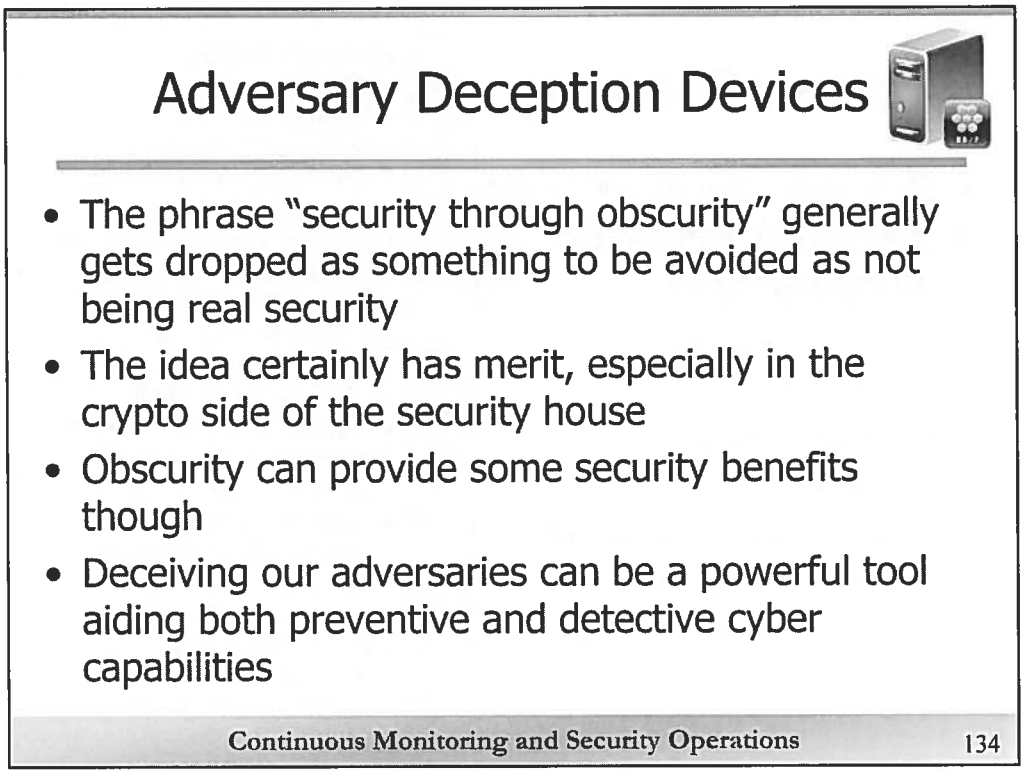
Packet Capture and Prevention

Much like with the SIEM, the packet capture device does not itself provide direct preventive benefit. However, rapid detection that moves to successful response can be a huge boon to the prevention of future impact.



Course Roadmap

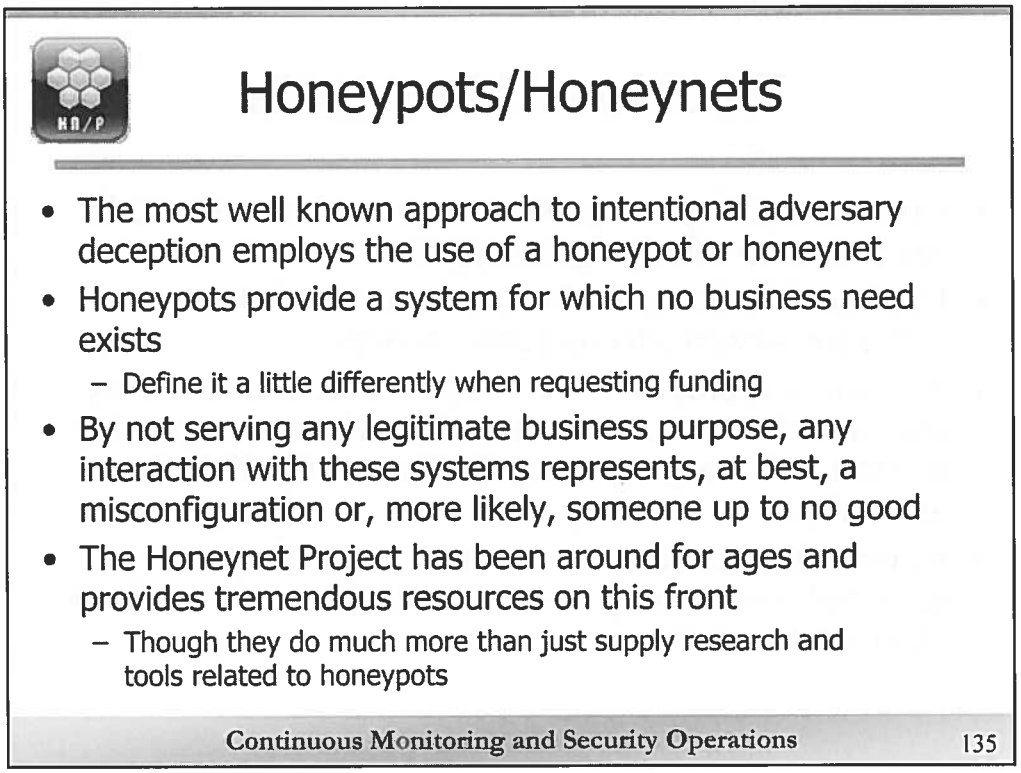
The next section is Adversary Deception Devices.



Adversary Deception Devices

Sometimes a dose of obscurity can be a significant boon to security. The phrase "security through obscurity" is usually meant derisively, but used appropriately obscurity can be a very good thing.

Also it is a lot of fun knowing that you are overtly deceiving your adversaries.

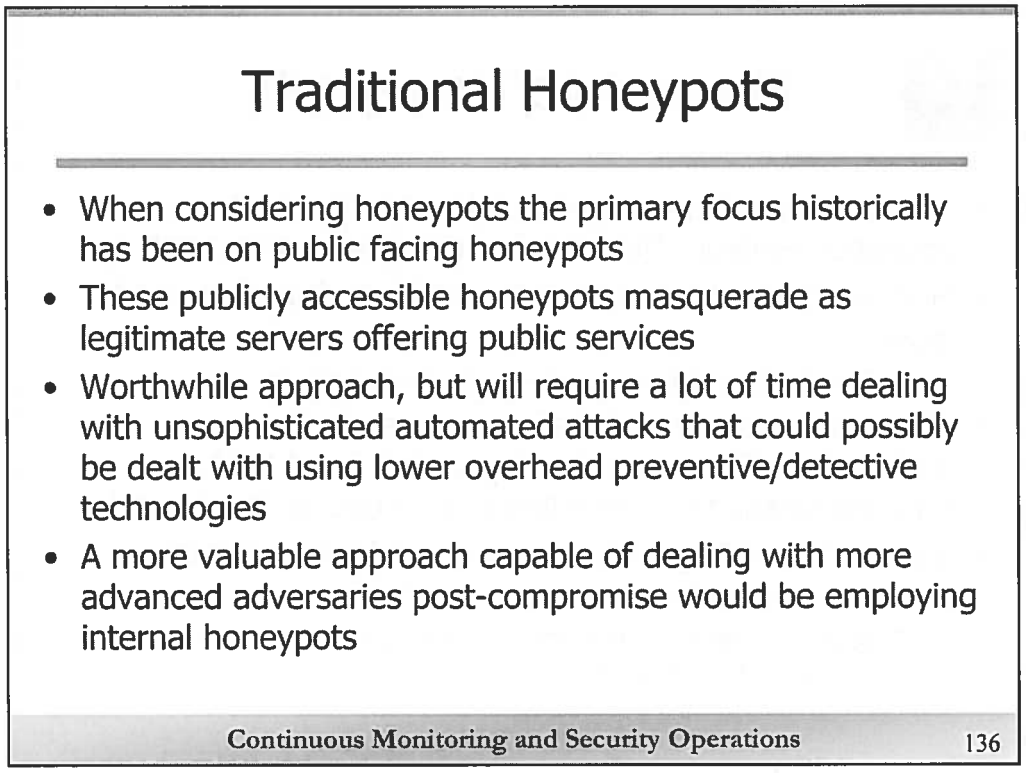


Honeypots/Honeynets

The Honeynet Project has been the most influential and visible organization in this space. The terms honeypot and honeynet are used to indicate deception devices. Honeypots are generally considered to be systems deployed that have no direct business need for interaction. The intent of the honeypot is primarily to serve as a trap for adversaries that mean to cause harm.

Because there is no legitimate use of a honeypot, any interactions with it are suspect. At best, a misconfiguration could lead to interaction with a honeypot, but the assumption is that any interaction is, at the very least suspicious.

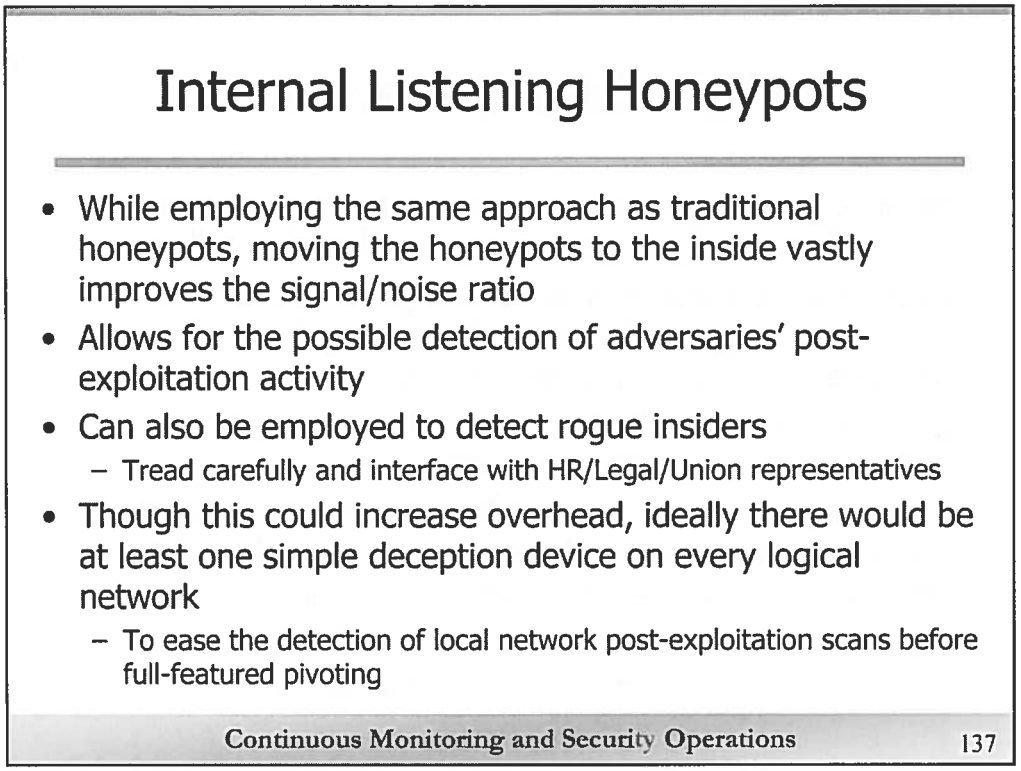
[1] https://www.honeynet.org/project (http://cyber.gd/511_21) **QR**



Traditional Honeypots

Historically the main emphasis on honeypots was to deploy these deception devices beside public facing systems/services. Effectively, now, in addition to your actual web server, you might have a honeypot web server that no one has any reason to know about/connect to as it is not offering legitimate business services.

While there is merit to these public facing honeypots, they tend to get hit with lots of automated scans and tools looking for very specific issues. While that can be valuable intelligence, the vast majority of the data simply points to unsophisticated attackers. And yet, to gain value from the honeypot requires actively leveraging the intelligence generated, which, in this case, can be fairly cumbersome.

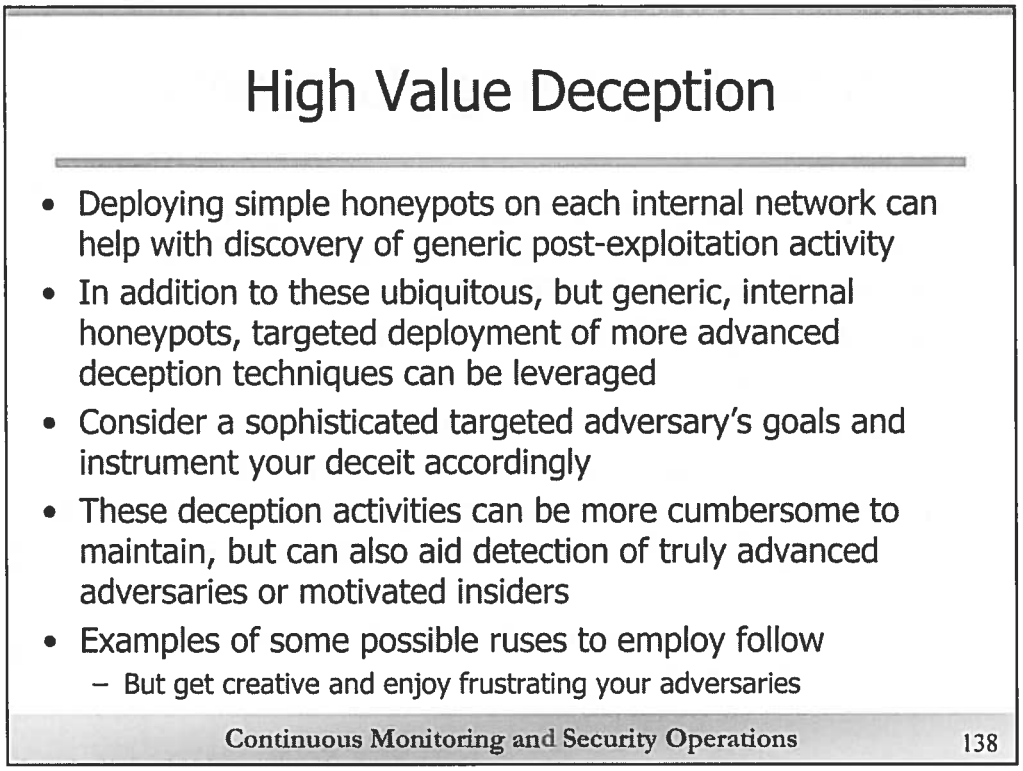


Internal Listening Honeypots

Rather than focusing all our deception devices on public segments could we benefit from pulling some of those back in-house? How could we use an internal honeypot and what would it look like? Further, what would be the goals?

Internal honeypots offer significant potential, but are not widely used at all. The goals of these honeypots are potentially twofold: detecting rogue insiders; and detecting pivoted post-exploitation activity. Tread very carefully when considering these as a tool for targeting potentially malicious insiders. Absolutely consult with HR, in-house counsel, and union representatives before going down that road.

Another, less controversial approach, targets the identification of compromised assets by looking specifically for pivoted post-exploitation. Simple, low-interaction honeypots could be leveraged and deployed on each and every internal network. If that proves easily manageable, then move to more sophisticated honeypots/honeynets or perhaps focus on high value deception.

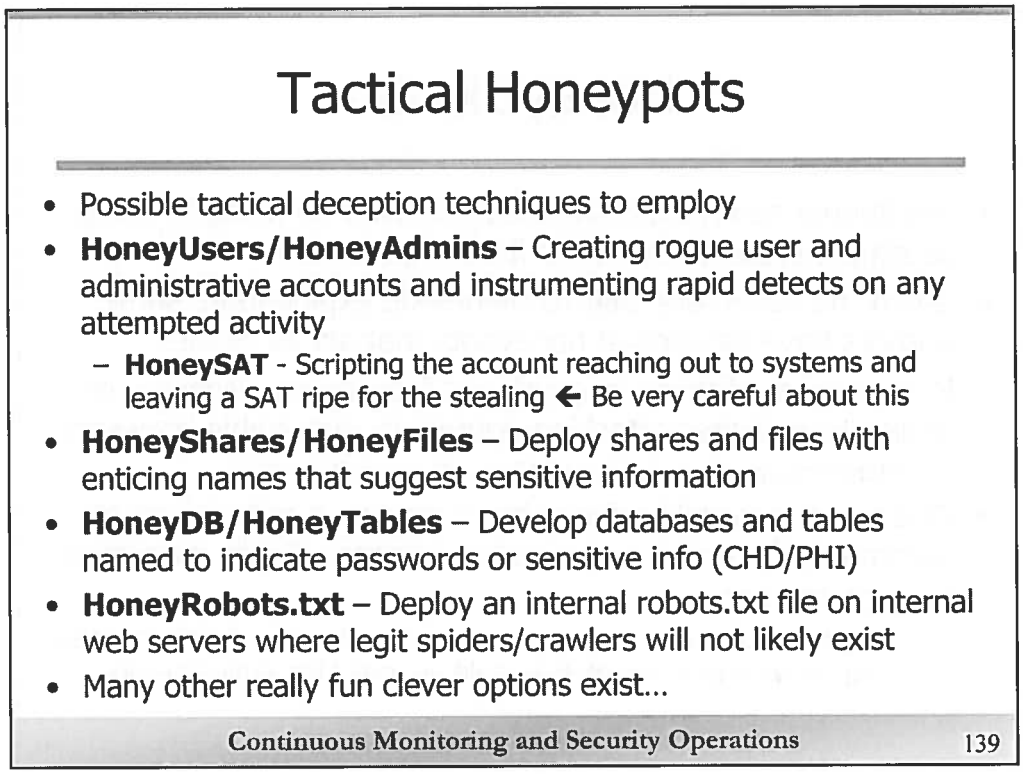


High Value Deception

Deploying simple low interaction honeypots on internal networks can prove a great boon to internal security's detection of basic pivoting and pivoted scans. However, we can gain even more value from honeypots by deploying them more tactically.

Now, the tactical internal honeypots can be a bit of a time sink, but they can also provide very significant and targeted value that little else is capable of providing. Consider your organization and what you are primarily concerned with protecting. Now consider ways in which someone would be able to compromise that data/system/application and think if there would be any way to potentially catch them before they could make it this far down that path.

Let's consider some generic examples, but keep in mind that the goal is to frustrate your adversary's ability to achieve theirs through more readily detecting their advances before they succeed with their end goal.

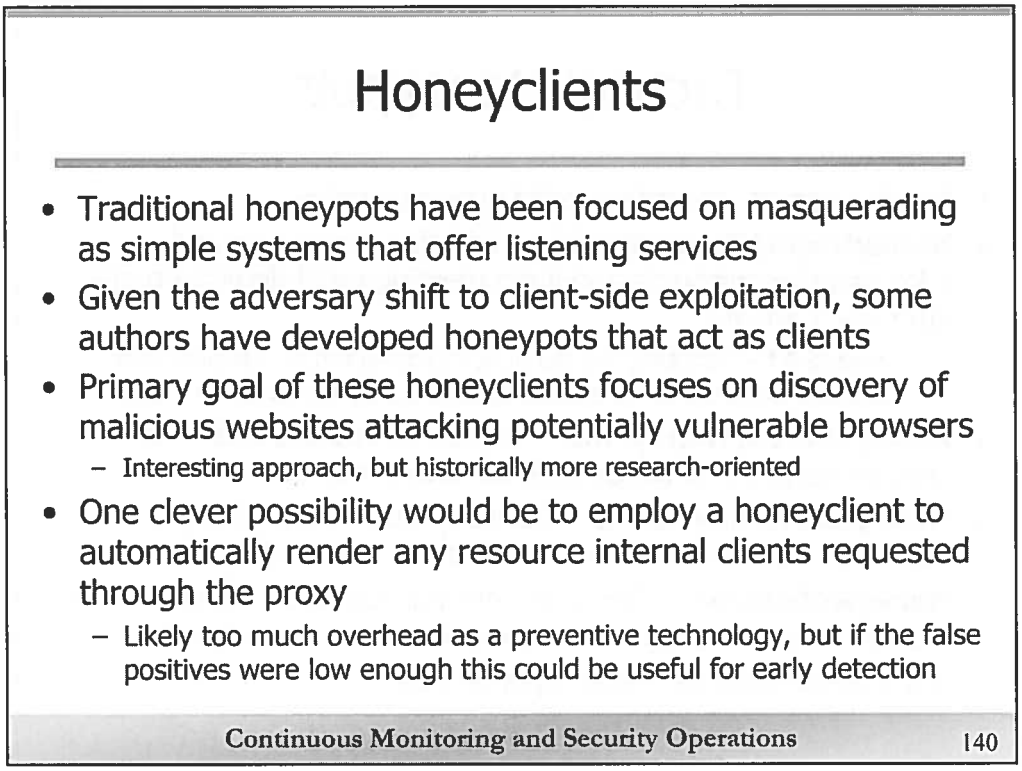


Tactical Honeypots

Some examples of tactical honeypots that could prove useful at both frustrating adversaries and also at potentially detecting internal shenanigans.

HoneyUsers/HoneyAdmins - this involves the creation of accounts, perhaps with names suggesting admin privileges. Do this not only for Windows/AD but also for other applications, databases, etc. How vulnerable you make yourself is open for discussion. Do you actually provide an easily guessable/crackable password? Could also get interesting actually have an account that routinely divulges its SAT (**HoneySAT**) to remote systems, but we lock it down and monitor it.

HoneyShares/HoneyFiles - These are simply shares and files meant to entice the adversary, but that are very closely monitored/alerted on any type of access.

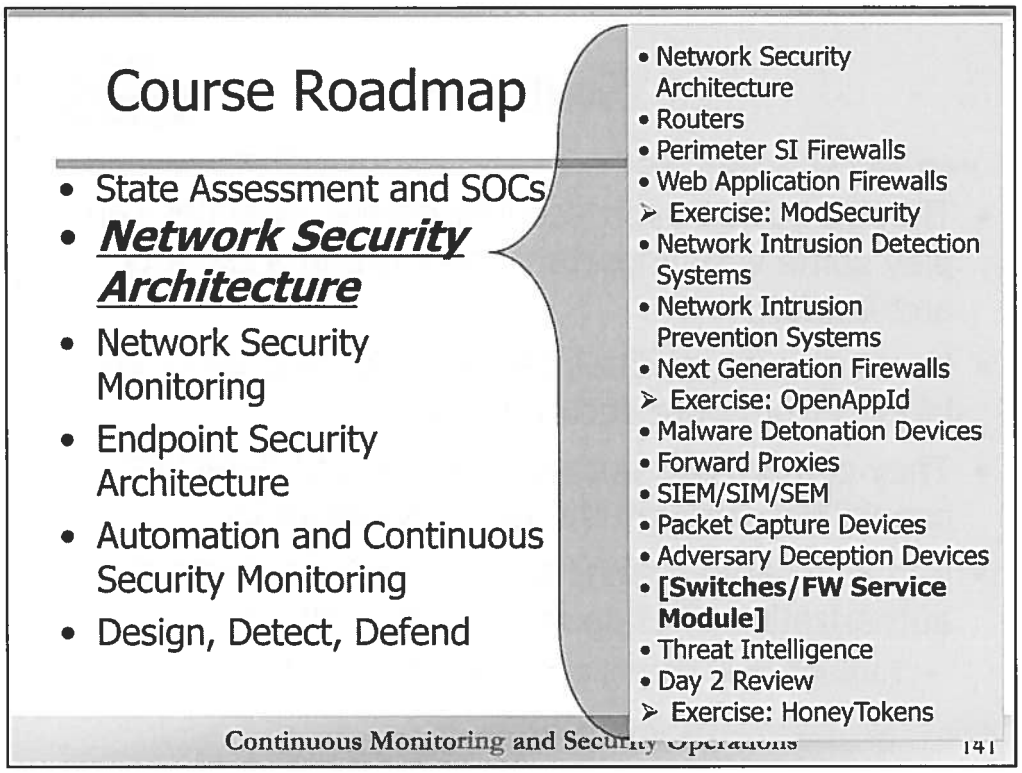


Honeyclients

Another consideration is employing client-oriented honeypots rather than listening honeypots. Though conceptually honeyclients have been around for years, their efficacy at providing operational security benefit has been rather poor. Primarily they have been used as research-oriented honeypots rather than production honeypots.

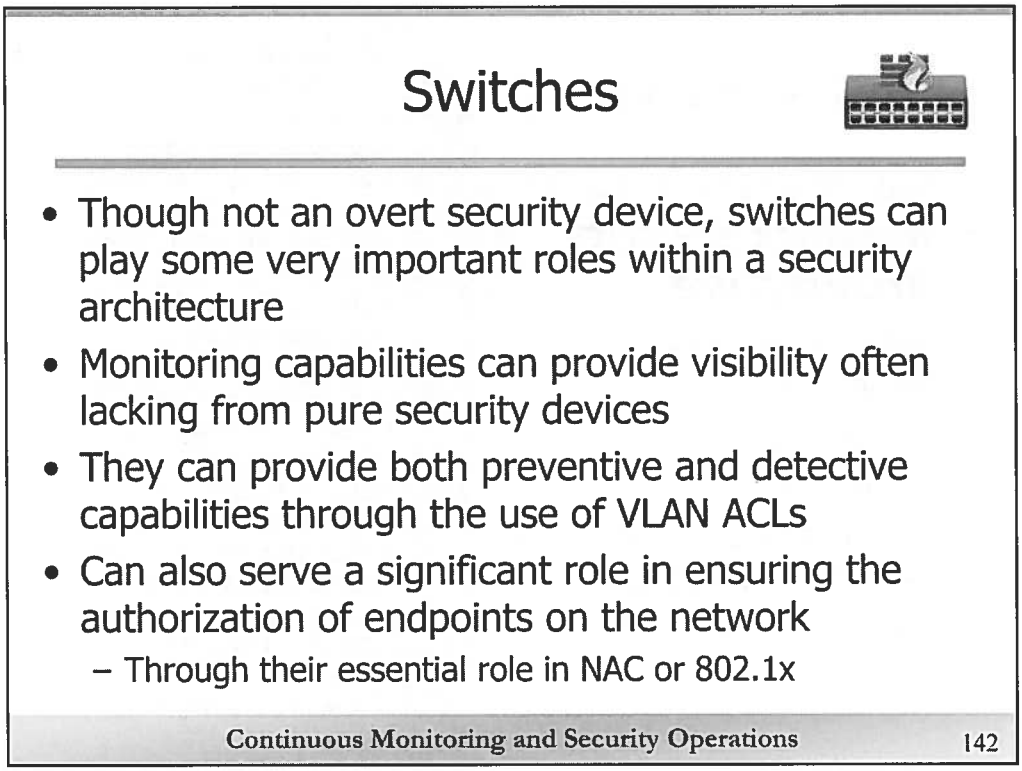
The goal of the honeyclient is to discover maliciously hosted content. Though this has been research-oriented in the past, the explosion of malware detonation devices could see these become decidedly more production-focused in the near term.

Probably way too much work and overhead, but could certainly see the ability of having a combination, for example, Squid proxy + Cuckoo analysis sandbox integrated to automatically have Cuckoo render everything the proxy is asked to fetch on clients' behalf.



Course Roadmap

The next section is Switches and Firewall Service Modules.



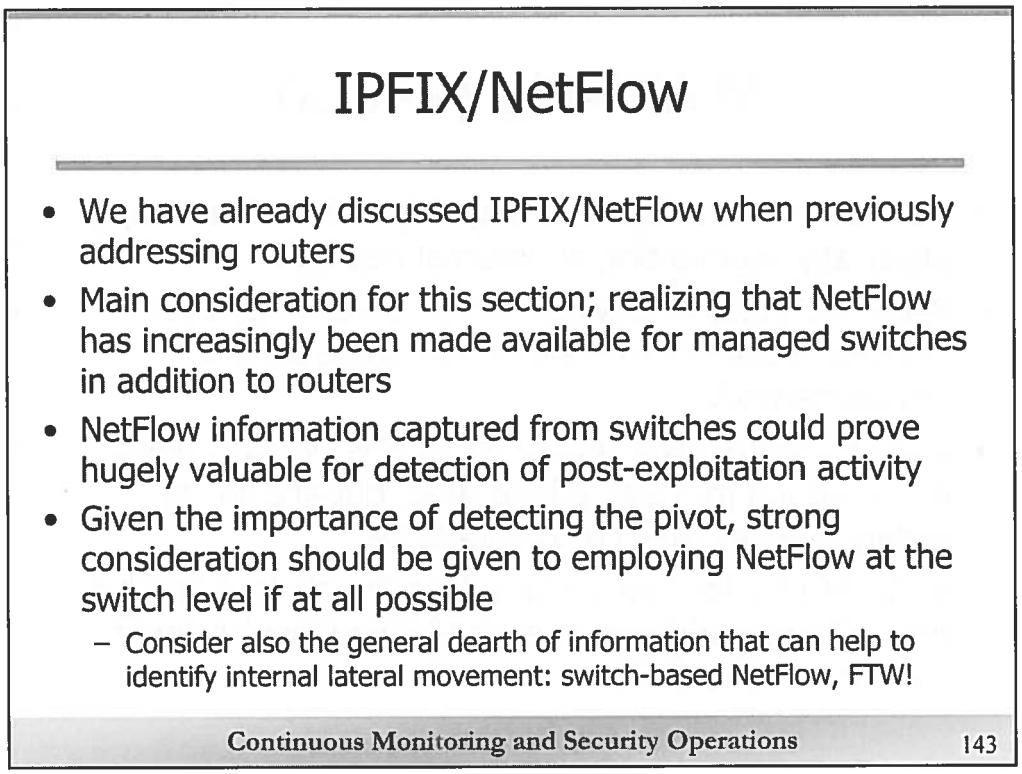
Switches

As you know, switches are not overtly a security device. However, nonetheless, they can play an important role with respect to security within the enterprise.

Formerly port statistics would have been considered the extent of monitoring capabilities afforded us by switches. Much more robust monitoring techniques have made it down to many, though not all, switches. This monitoring can play a vital role in helping provide visibility that is otherwise quite lacking.

Another key security aspect of switches is related to their ability to through VLAN ACLs provide preventive as well as detective capabilities that break up flat, at least from a security perspective, networks into something more securely segmented.

Though we will not delve into this aspect of switch security, the devices also play a vital role in endpoint authorization via NAC and 802.1x.



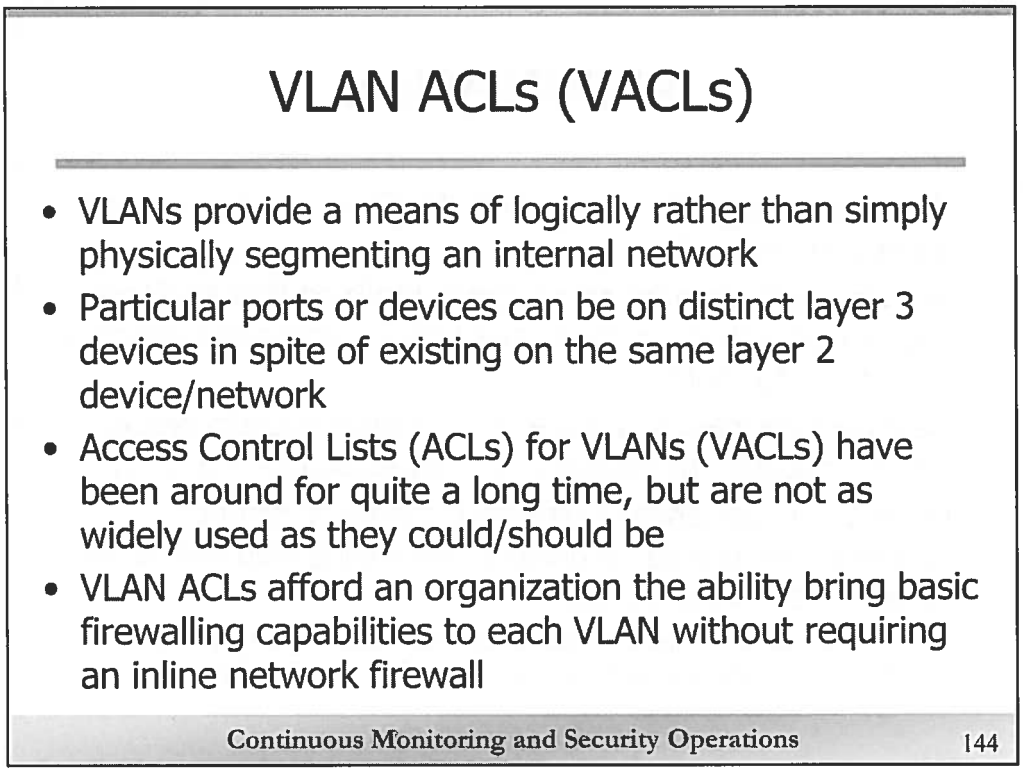
IPFIX/NetFlow

Though we have already discussed IPFIX/NetFlow previously (see the section earlier in the day on Routers for a refresher), it is important to realize that increasingly these capabilities are showing up with more regularity as a switch capability in addition to a router capability.

The configuration, type, and version of NetFlow supported, if any, can vary quite a bit even within the same vendor. Not surprisingly, Cisco seems to be the largest player in the space pushing NetFlow down to virtually every IOS device as of the 11.1 train[1].

NetFlow exports from switches greatly bolsters the security visibility within our networks.

[1] http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod_white_paper0900aecd80406232.html (http://cyber.gd/511_237) **QR**

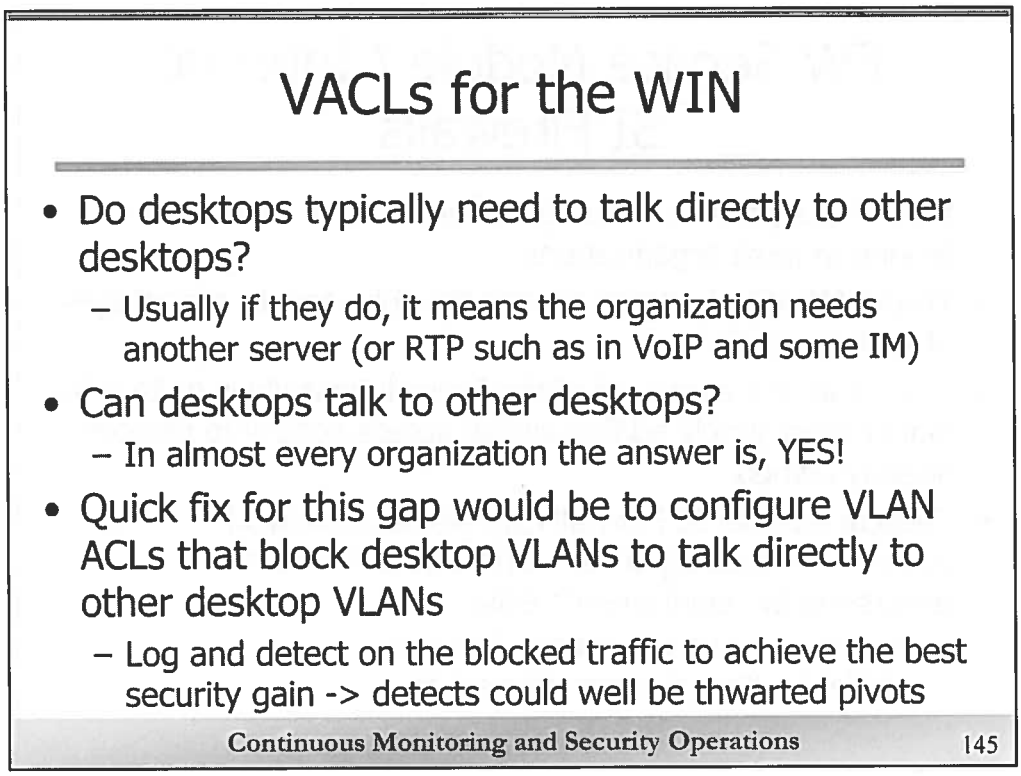


VLAN ACLs (VACLs)

While physical separation of every network would be a vastly more "secure" architecture, it would actually cause lots of little and some bigger self-inflicted Denial of Service attacks. While air gaps might be a gold standard for segmentation, it is absolute overkill, or at least to costly, for the majority of our networks.

VLAN ACLs are another means to achieve security segmentation, but without nearly the cost of air gaps. VLAN ACLs might be able to simply be bolted onto the existing VLAN implementation at your organization. Most organizations already employ VLANs, but they are typically only for performance and simple logical groupings. That can and should change.

Our internal security (our meaning the world's) is pretty poor, and a relatively simple cost-effective means to bolster internal security comes in the form of VACLs.



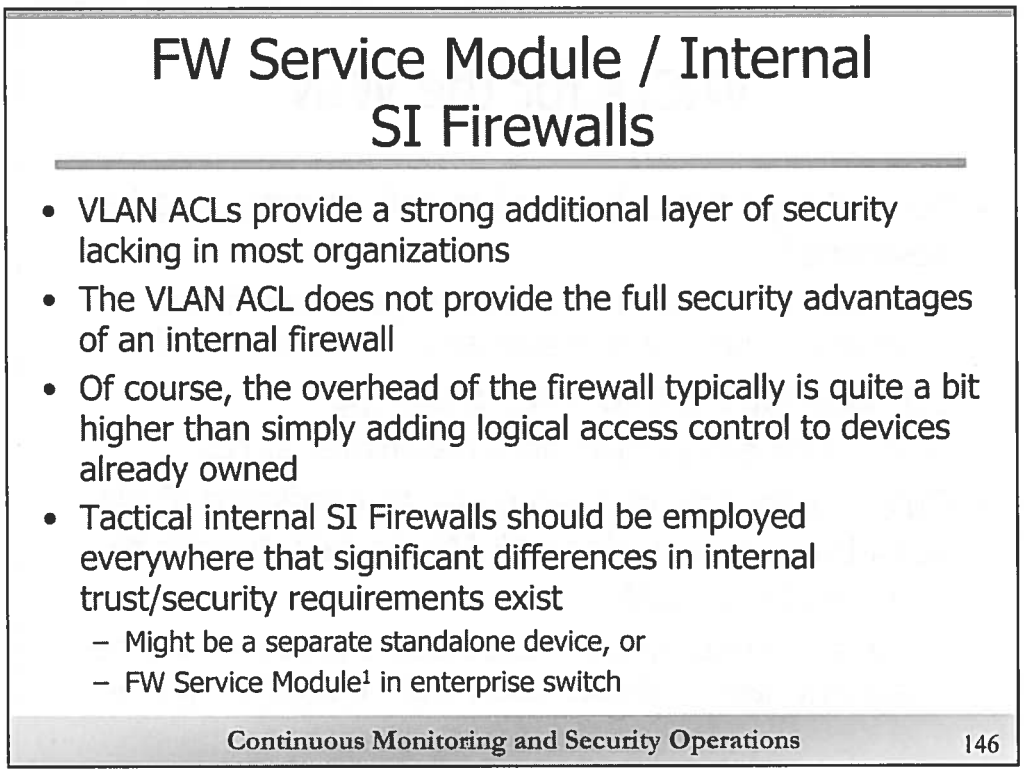
VACLs for the WIN

Let's consider an example of how a VACL can bolster internal security. Quick question, do desktops need to talk directly to one another? Perhaps for some RTP associated with VoIP or (e.g. Microsoft Lync Server). Even if there are others the list should be pretty darn short. What about unicast Windows networking protocols like SMB/CIFS? Should desktops talk to other desktops on SMB? I'd say no.

So what if they do, well you might have encountered a scenario I hadn't considered, or you might find someone poking around or perhaps even malware.

Regardless of should they be able to communicate, could desktops talk to other desktops in most enterprises? Yes, in almost every organization you encounter.

While we could try to bolster the endpoint firewall to do some blocking for us, and that is a sound idea, another approach would be to leverage our friends the VACLs to deny (and log) the traffic. The log piece is critical because these detects, after we get things filtered properly, will be invaluable at identifying scenarios we are concerned with.

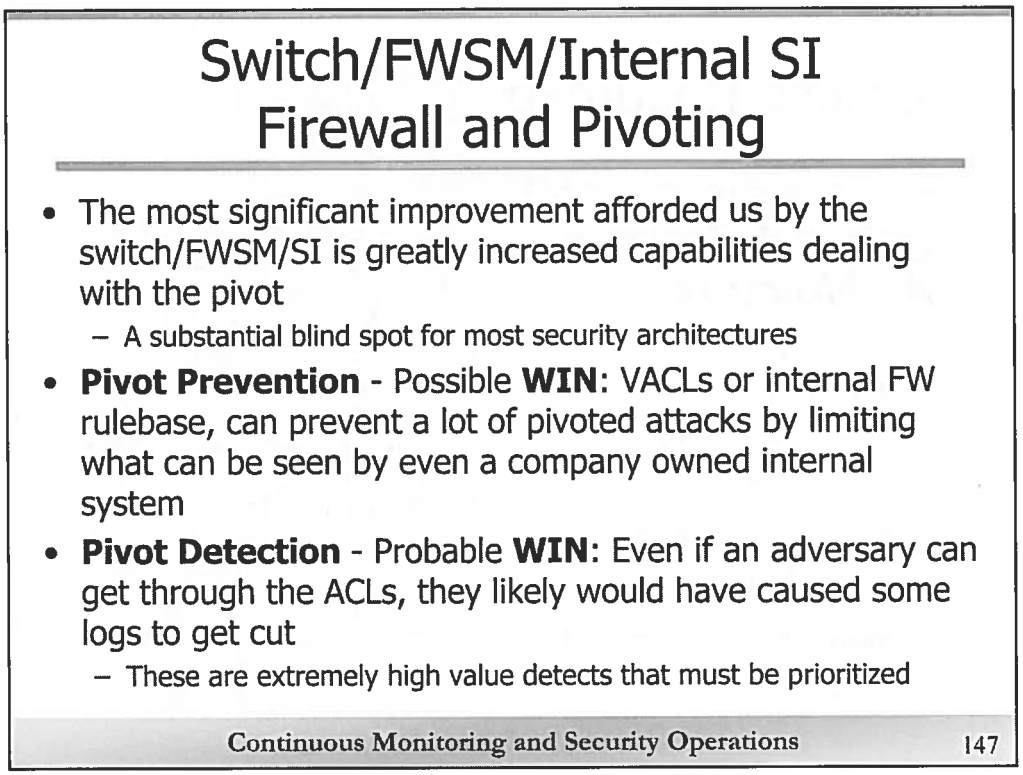


FW Service Module / Internal SI Firewalls

Though VLAN ACLs are a great boon to internal security, and the price is certainly right, for more sensitive segments of the organization internal network firewalls should be employed. VACLs are not a serviceable replacement for a firewall. Even full-featured IOS ACLs, supported in the L3 Switch, are not an acceptable replacement for a firewall.

My preference would be to employ a network full firewall, if possible. Understand that logistically this full SI firewall might well actually end up being a service module in an enterprise switch. In fact, the firewall service module approach would actually be preferred, in some respects, not because it represents a more robust firewall offering. It does not. However, the service module could actually be a better solution as it is more scalable and can, over time, be applied to more and more VLANs.

[1] http://www.cisco.com/c/en/us/products/interfaces-modules/catalyst-6500-series-7600-series-asa-services-module/index.html (http://cyber.gd/511_238) **QR**

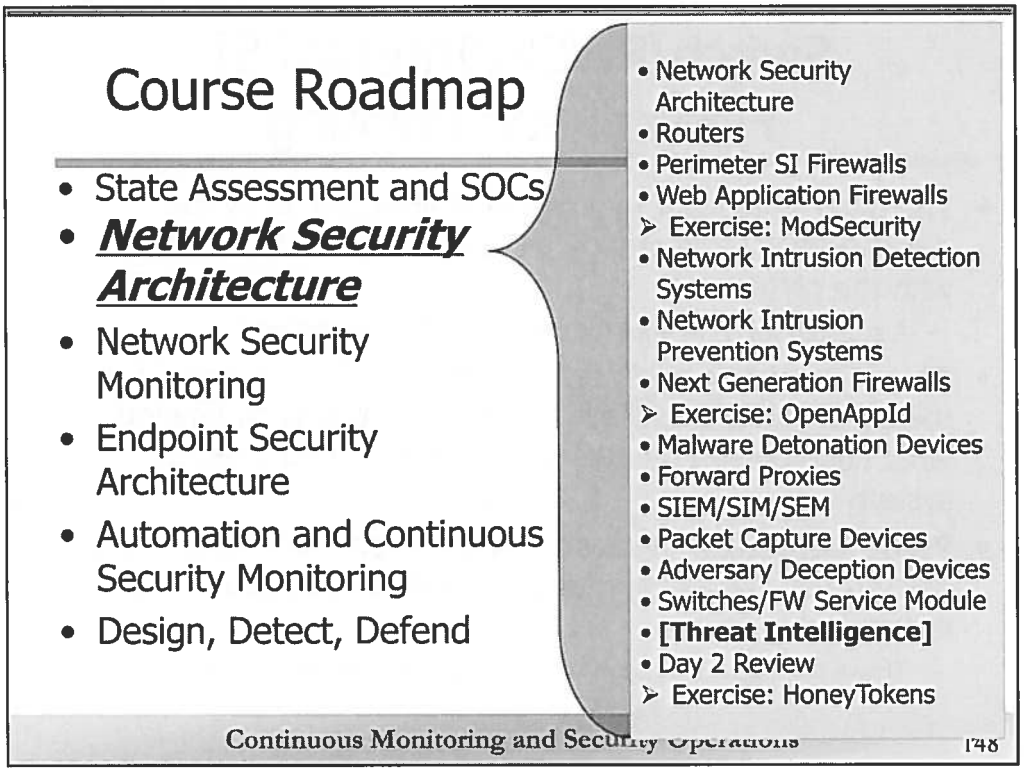


Switch/FWSM/Internal SI Firewall and Pivoting

The Switch, Firewall Service Module, or Internal SI Firewall offers tremendous ability that few other security tools, certainly network ones, can provide. Namely, these approaches can greatly increase an organization's ability to detect and possibly even prevent pivoted attacks.

As stated from the beginning of the course, lateral movement plays a key role for advanced adversaries. Anything we can do to better defend against this potential is a big win for us.

VACLs, and the like, can help prevent pivots by limiting what even fellow insiders might have access to on a given VLAN. Though possible to fully prevent successful pivots, an adversary might still be able to get through the prevention. However, their initial attempts would likely have resulted in VACL drops and logs. Those logs enable us to detect the attempted pivot. Needless to say, these logs afford us some extremely high value detects that absolutely must be prioritized for rapid review and response.



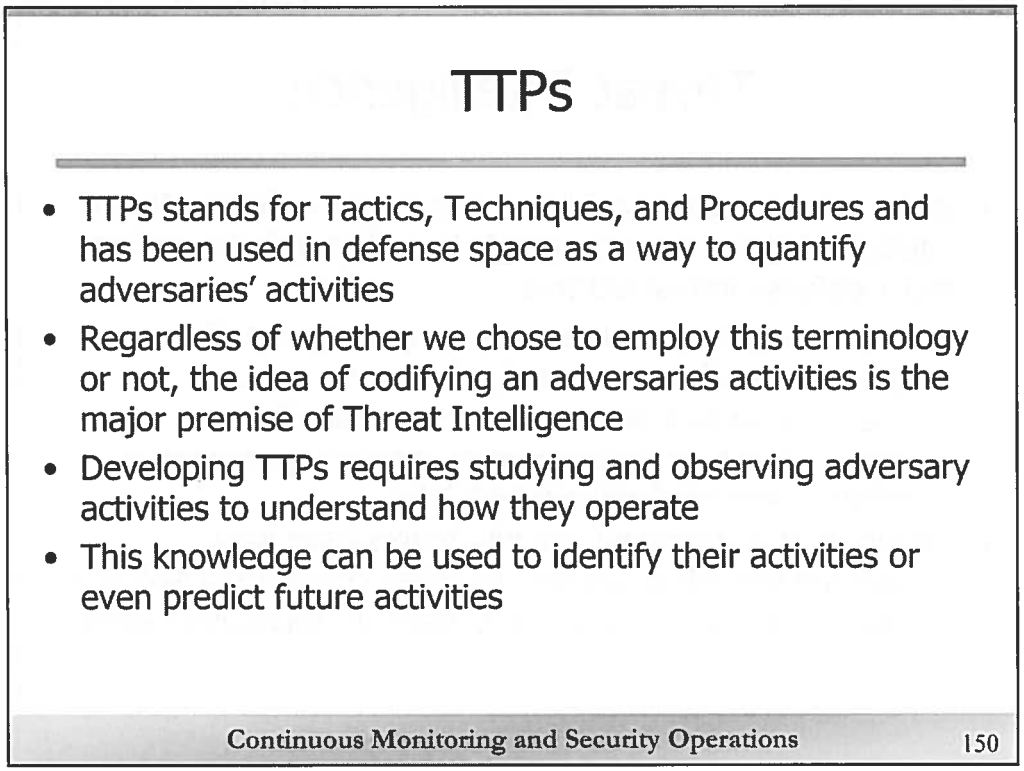
Course Roadmap

The next section is Threat Intelligence.

Threat Intelligence

Historically information security has emphasized the vulnerability side of the **Risk = Threat x Vulnerability**. The focus on vulnerabilities to ultimately reduce risk makes sense given that we generally have more control over the vulnerability side of the equation. Though our greater potential to impact vulnerabilities is no doubt true, this does not warrant a blindingly myopic focus on vulnerability alone.

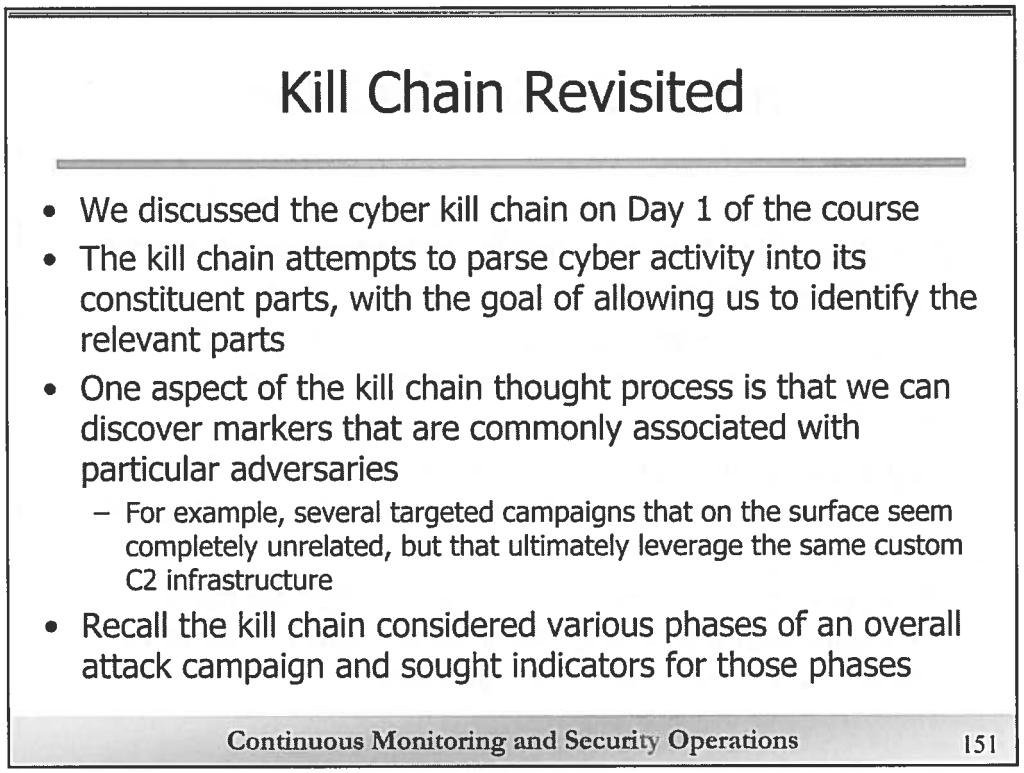
In recent years, enterprise information/cyber security has started to pivot towards greater emphasis on threats. The emphasis is not to the exclusion of vulnerabilities, but it is fueled by the understanding that offense can and should inform defense. The particular vulnerabilities that should be prioritized, the way in which they can potentially be exploited, the likelihood of capable adversaries, these all are best informed by threat intelligence.



TTPs

Governments and militaries throughout the world have quite a head start on the enterprise in considering threat intelligence. An acronym commonly employed to characterize particular adversaries activities is TTP, Tactics, Techniques, and Procedures.

We are not going to get incredibly formal with our treatment of TTPs, but this can serve as a threat intelligence touchstone. This allows us to have a bit of language that we can use internally when characterizing various adversaries and their activities.

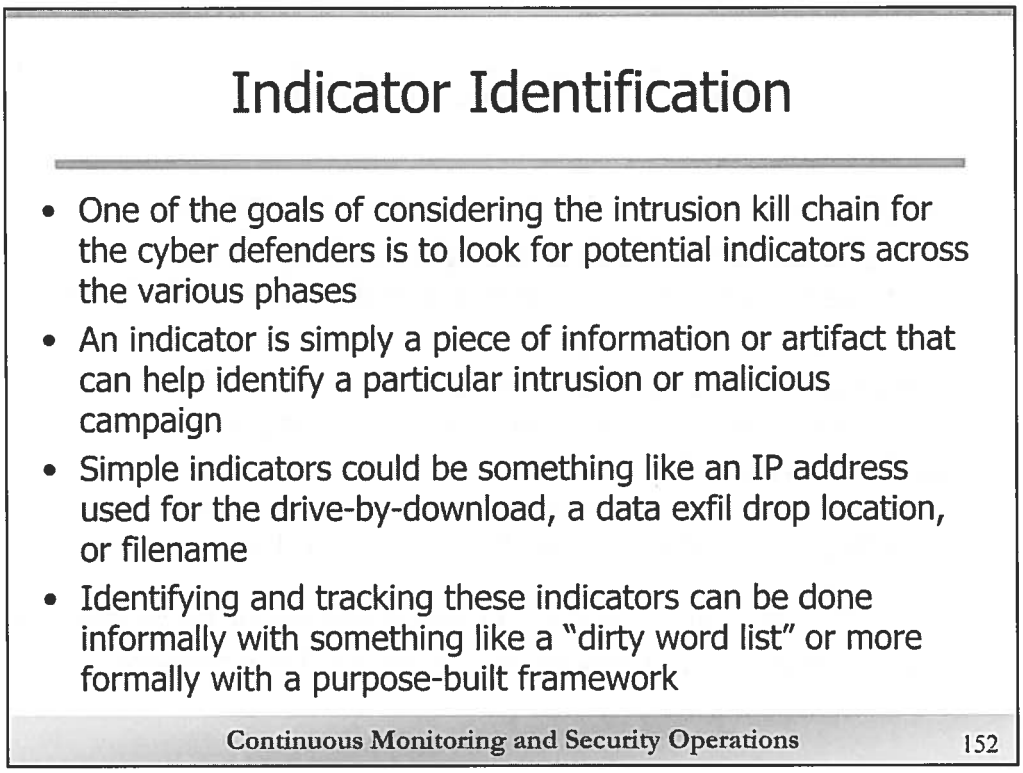


Kill Chain Revisited

Let us revisit the idea of the cyber kill chain that we discussed during day 1 of the course. In some respects we have been looking at pieces of the Cyber Kill Chain in today's material by considering various means of detecting and preventing adversary activities such as the exploitation, pivoting, and C2.

One of the primary emphases of the idea of the Kill Chain is to provide a model for considering various elements of a cyber intrusion. By codifying various phases and activities in those phases, the cyber kill chain provides a model for us to consider means to potentially detecting adversary activities within each phase. As we are reviewing particular incidents/intrusions consider how we could detect this activity in the future.

These detectable artifacts that we uncover can serve as indicators to detect future activities, and, depending upon the indicator in question, it could even point at a particular actor.

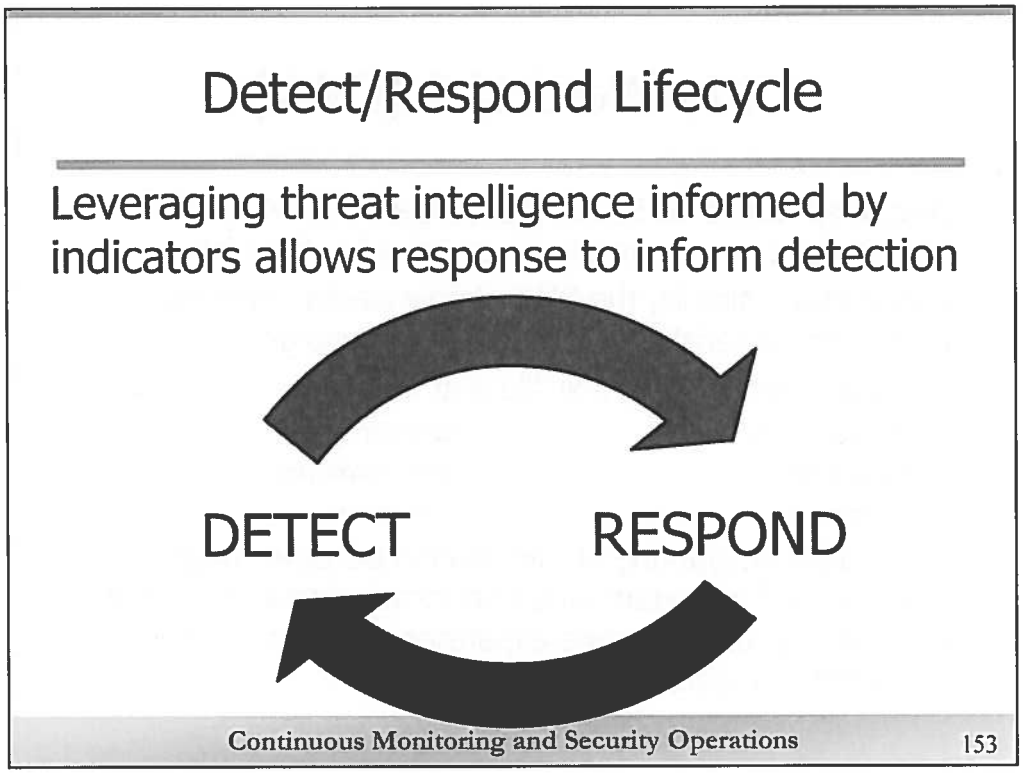


Indicator Identification

One of the primary emphases of the intrusion kill chain is identification of indicators. Indicators are simply information, sometimes termed an artifact, that can aid in the identification of a particular intrusion, malware campaign, or adversary's activities.

Indicators can vary in complexity. Some of the obvious and simple indicators include items such as IP addresses of mail servers delivering phishing e-mails, hostnames of websites hosting malware, or filenames, service names, usernames. More complex indicators are also possible, and might be less likely to be mutated by the adversaries. Examples of these types of indicators might include coding style, binary packers employed, exploitation techniques.

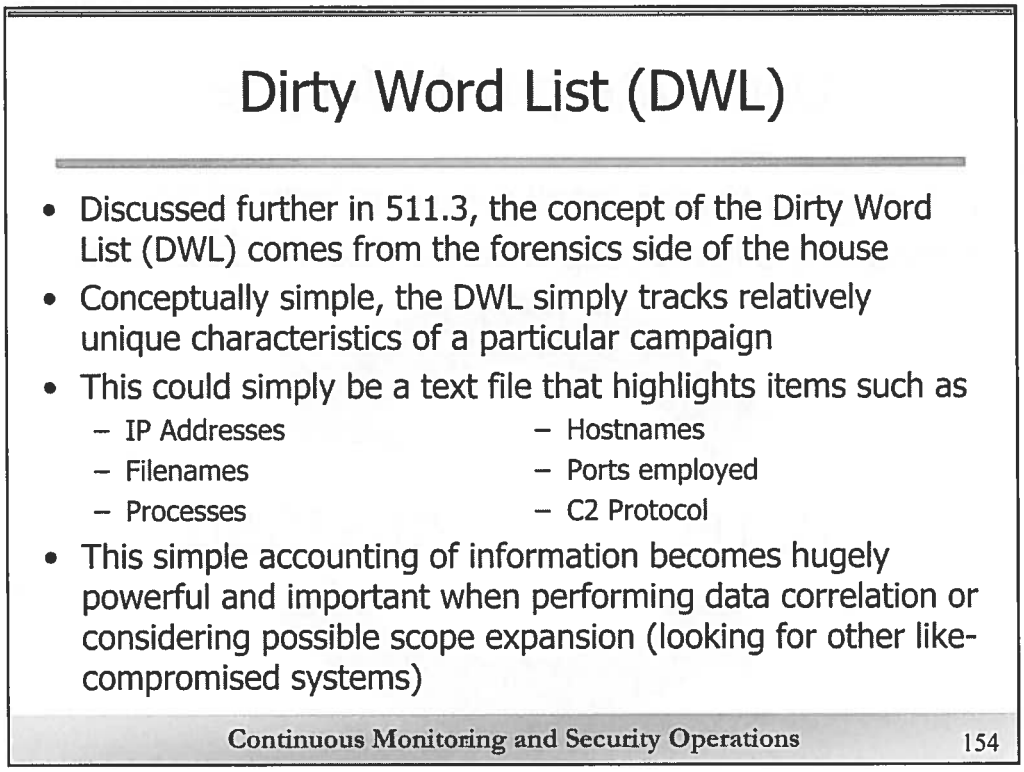
To leverage these indicators can be a simple process or a complex framework depending upon the need and maturity of the organization leveraging the indicators.



Detect/Respond Lifecycle

As mentioned before, detection must feed into response in order to actually make a meaningful impact on cyber security. However, response must also feed back into detection in order to make both detection and response more efficient and effective.

Indicators are created (or sourced) after having performed some analysis on a particular intrusion, which means that intrusion response often initially creates, or at least greatly increases the number and quality of, the indicators tracked.

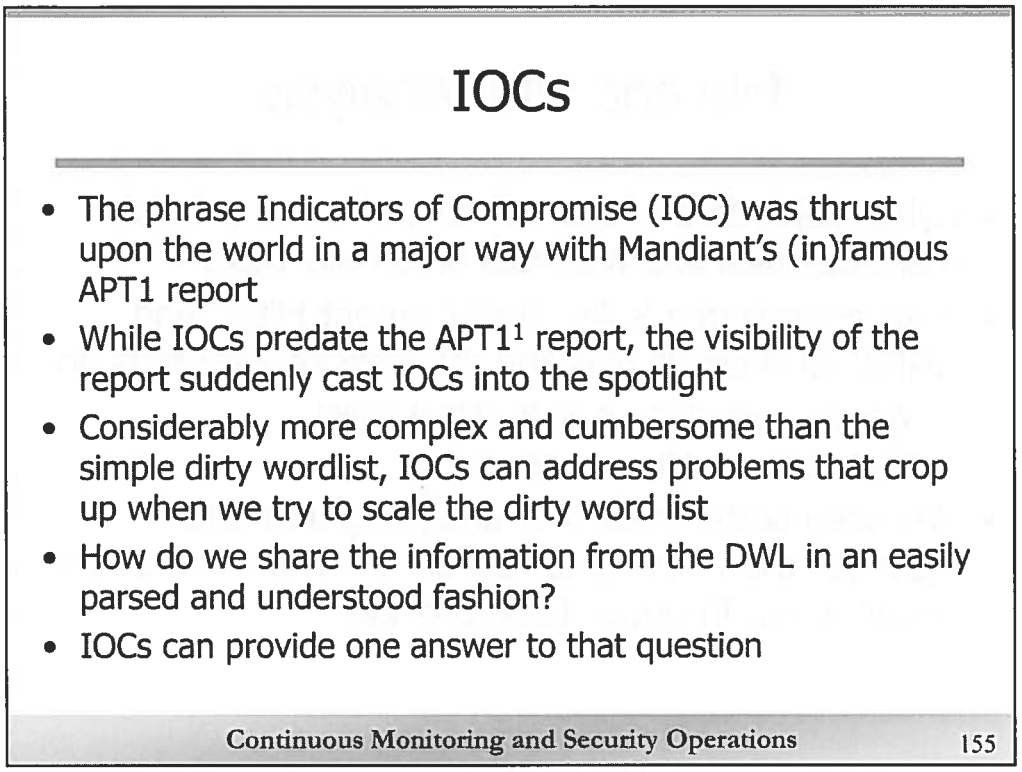


Dirty Word List (DWL)

While considering the Kill Chain we discussed the possibility of discovering artifacts of an intrusion that might allow us to uncover further activities that are occurring, have occurred, or possibly will occur. While the concept of indicators can be leveraged to build out extremely robust cyber TTPs for particular adversaries, we can also simply wield them in a less formal fashion.

To make this idea more approachable, I continue to use the less rigorous, but conceptually simple, idea of the dirty word list (DWL). The DWL can simply be thought of as a virtual scratchpad that you populate with key data that can identify an intrusion. Simplest case, we think a particular external IP address is evil, or simply somehow associated with this intrusion, so we add it to the DWL.

Conceptually simple, the DWL can be an incredibly powerful tool to look for other systems that might have been targeted or compromised by the same actor or in a similar fashion. This helps us with truly understanding the scope of the intrusion. In addition to looking at current data, we can also review historical data, if available, in the case that these same activities have occurred previously, but that we missed. We can also potentially turn this data into signatures in, for instance, our IDS infrastructure to help alert us to similar activities in the future, assuming they are relatively unique.

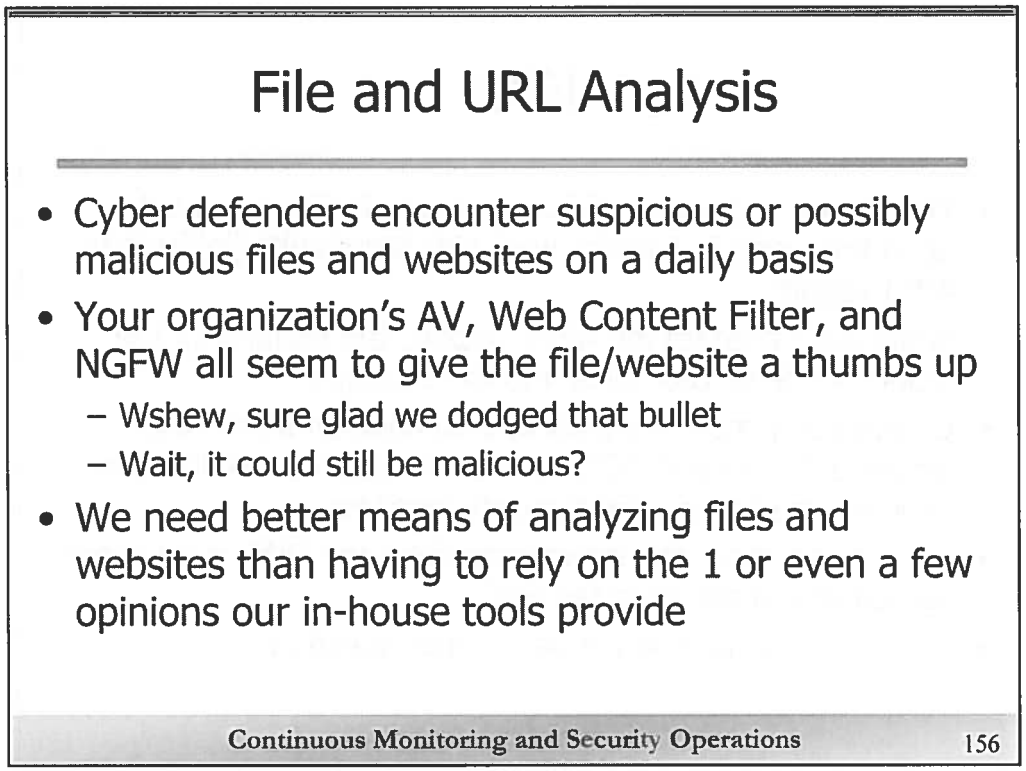


IOCs

The simple dirty word list (DWL) served the community quite well for many years, but unfortunately that simplistic text file approach does not scale well for larger teams. Further, the DWL does not allow for easy sharing of data in a predictable easily parsed fashion.

IOCs, or Indicators of Compromise represent a much more formal approach to documenting artifacts associated with intrusions and activities. The main benefit of IOCs over the simple DWL are its ability to scale for multiple analysts. Further, IOCs are built for information exchange, which allows for the easier sharing of intelligence.

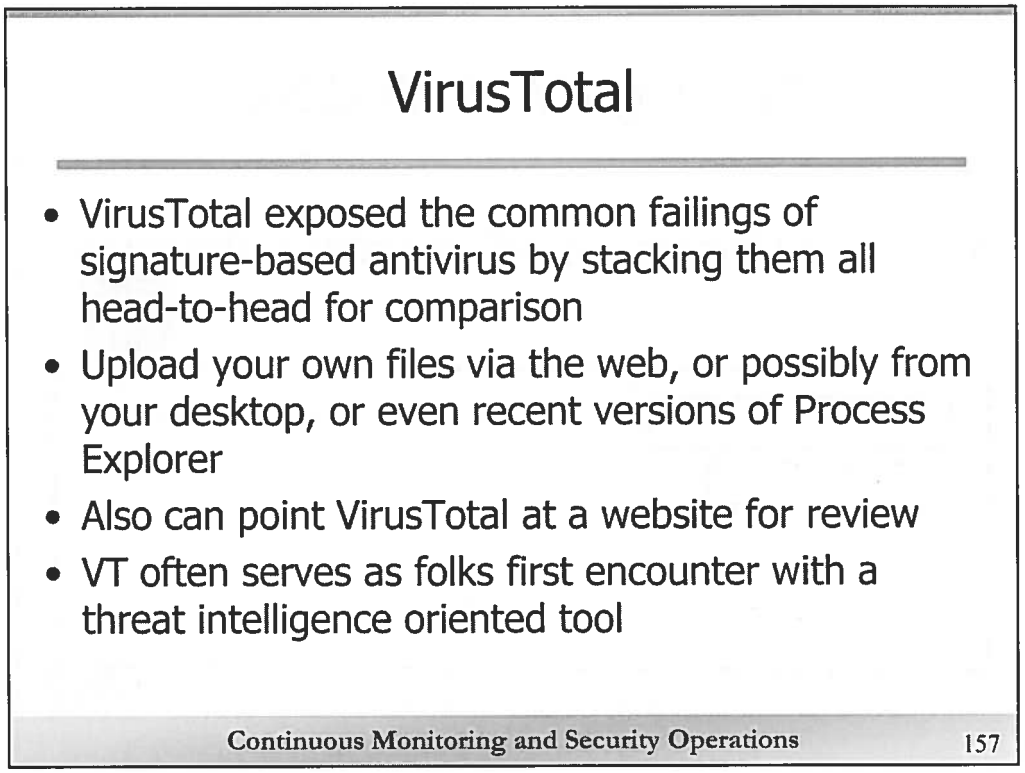
[1] http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf (http://cyber.gd/511_242) **QR**



File and URL Analysis

While conducting analysis and investigations we often encounter files and websites that we believe to be suspicious/malicious. How do we confirm or deny our suspicions? Well, if the file URL passed muster with all of our various devices, don't you think it could be trusted? Unfortunately, just getting through even our heavily instrumented architecture is no guarantee the file or URL is benign.

We need a better way of, at least on an ad-hoc basis, gaining further intelligence about files/URLs that we find suspicious. Merely passing muster with even multiple antivirus engines is no indicator of benign.

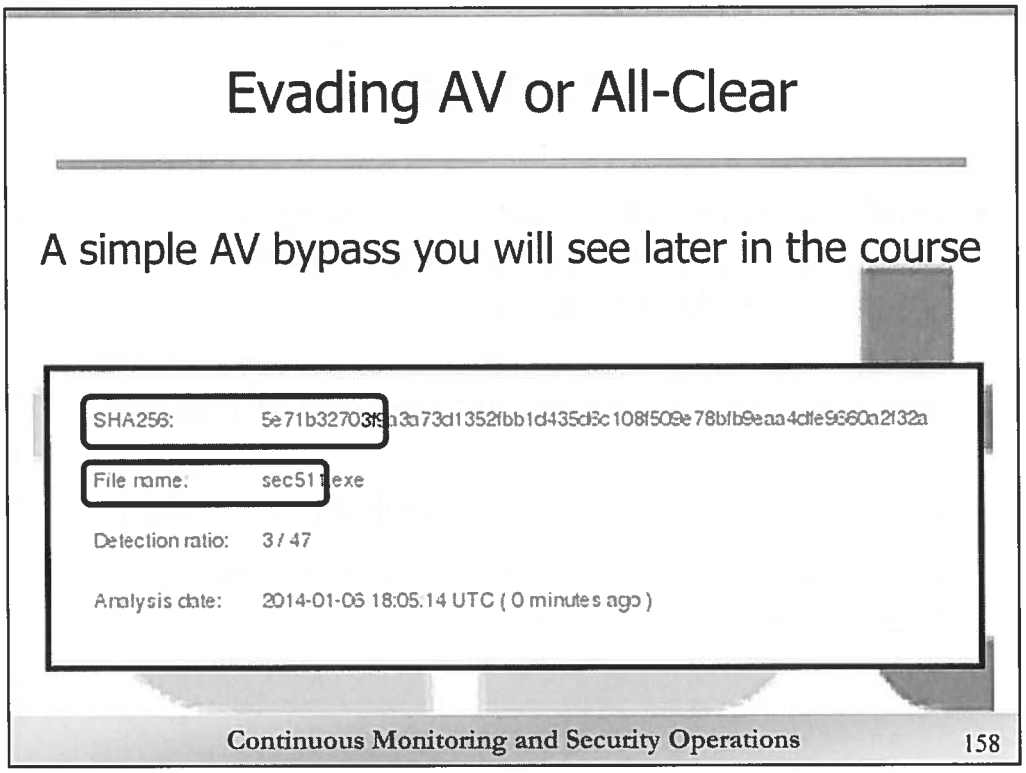


VirusTotal

Commonly the first threat intelligence oriented tool that many security professionals discover to perform some ad-hoc analysis of files is VirusTotal. The primary claim to fame of VirusTotal has been its free web interface that allows for uploading of files. These files will be run through, at present, 50 different antimalware engines.

Though VirusTotal is primarily known simply for file analysis with respect to Antivirus, it has more capabilities than just that. One of the most important additional features is the URL scanning functionality, which we will discuss shortly.

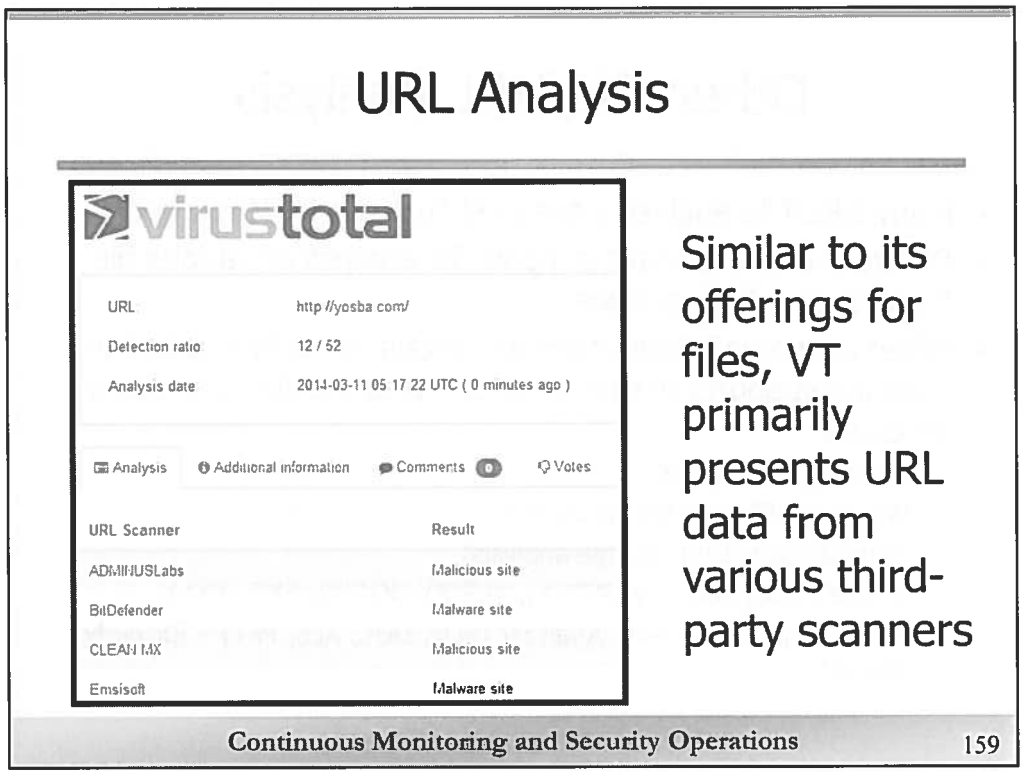
https://www.virustotal.com/ (http://cyber.gd/511_243) **QR**



Evading AV or All-Clear

Here we see the result of a VirusTotal scan against a file that was created for this course specifically. You will see it again later. So, does this mean that the file is clean or that AV has been successfully evaded? Very hard to tell, one thing that you will find that VirusTotal does in addition to basic AV scanning it creates, depending upon the type of file, a File Details and/or an Additional Information tab.

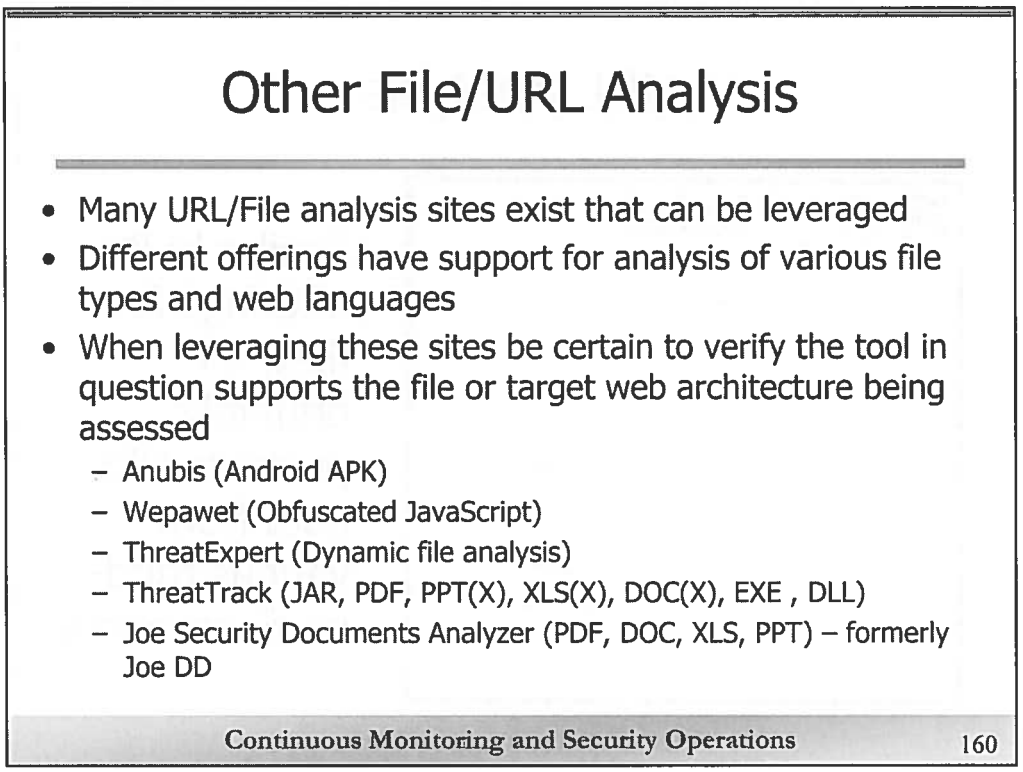
The File Details/Additional Information tabs can, depending upon the file type in question, provide a tremendous amount of information about the file itself. The actual content provided depends upon the type of file being analyzed.



URL Analysis

Another significant offering from VirusTotal is to run a URL through various third party scanners and present the results. In addition to the straight Analysis tab that indicates either Clean, Malicious, or Suspicious, VT also provides extremely useful data under the Additional Information tab.

Some examples of additional information will be common vendors' website categorization of the target as well as an indicator as to whether the site is known to have previously hosted malware, even if it does not currently.



Other File/URL Analysis

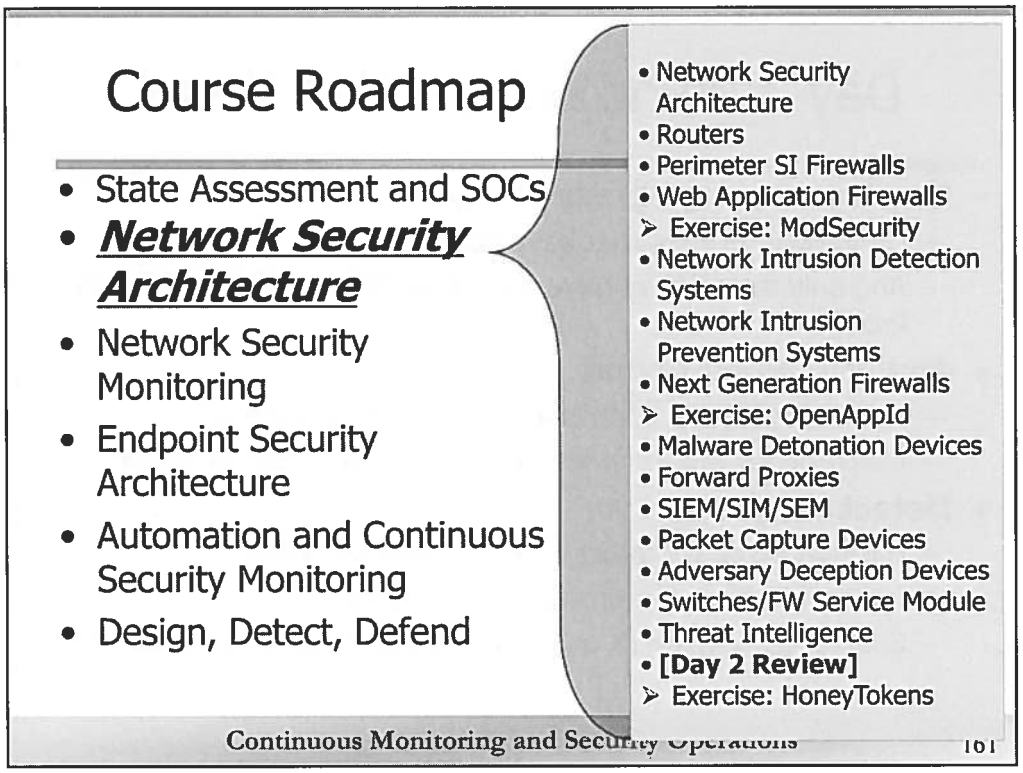
There are an increasing number of sites that will perform both static and dynamic analysis on files. Also there are a number of sites that will perform URL analysis by actually having a client interact with the sites.

These can be extremely powerful ways of gaining intelligence about the files and websites that are so frequently being created anew and updated. Lenny Zeltser, GSE #2 (@lennyzeltser) has a list of sites that will try to determine if websites[1] are malicious and a separate list for file[2] analysis capabilities.

Joe DD (from Joe Security) has become four sites: www.file-analyzer.net, www.apk-analyzer.net, www.document-analyzer.net and www.url-analyzer.net.

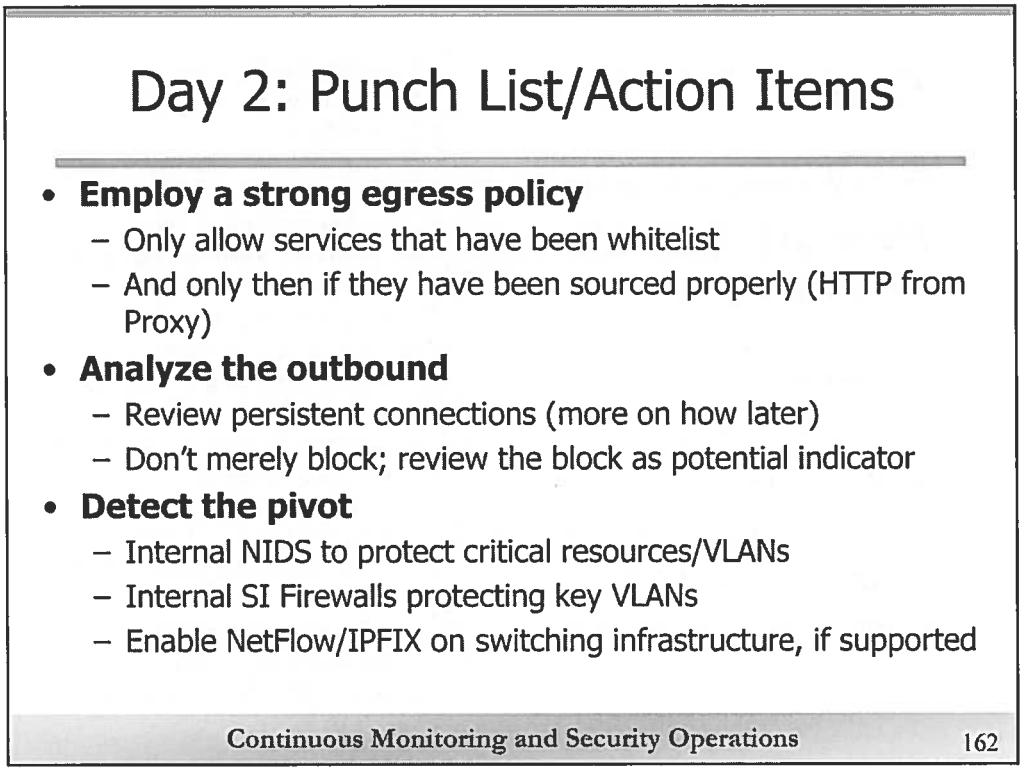
[1] http://zeltser.com/combating-malicious-software/lookup-malicious-websites.html (http://cyber.gd/511_239)

[2] http://zeltser.com/reverse-malware/automated-malware-analysis.html (http://cyber.gd/511_241) **QR**



Course Roadmap

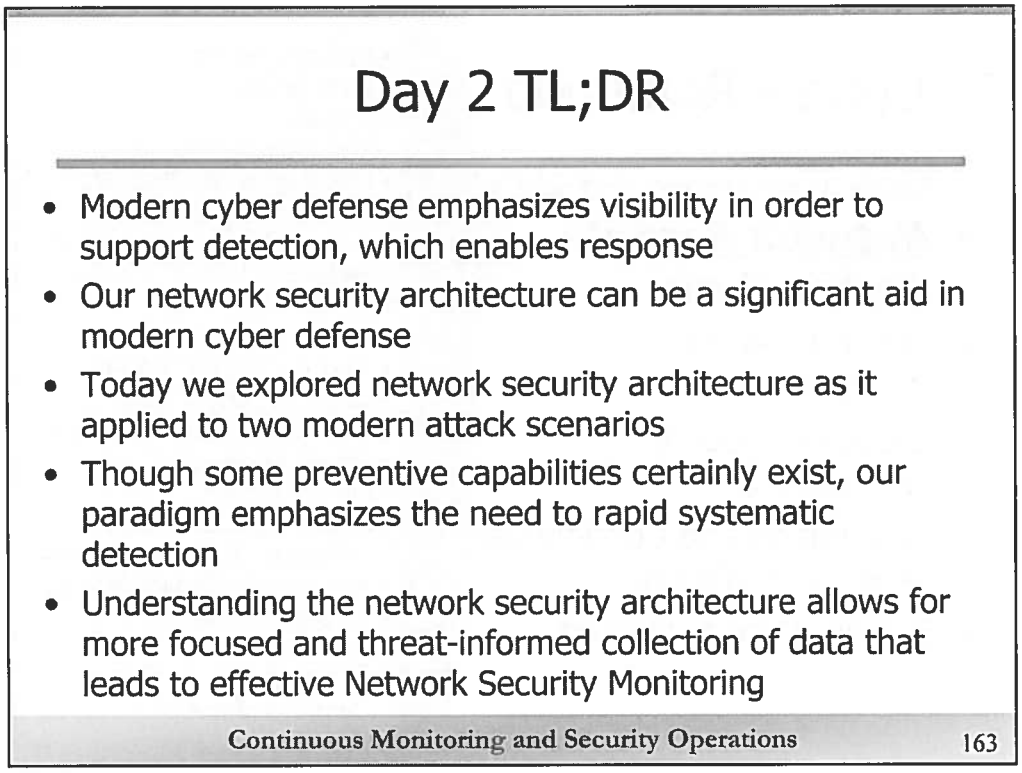
The next section is the day 2 review.



Day 2: Punch List/Action Items

Though there are, no doubt, many action items for you to implement at work, we want to make sure that at least these three are reiterated.

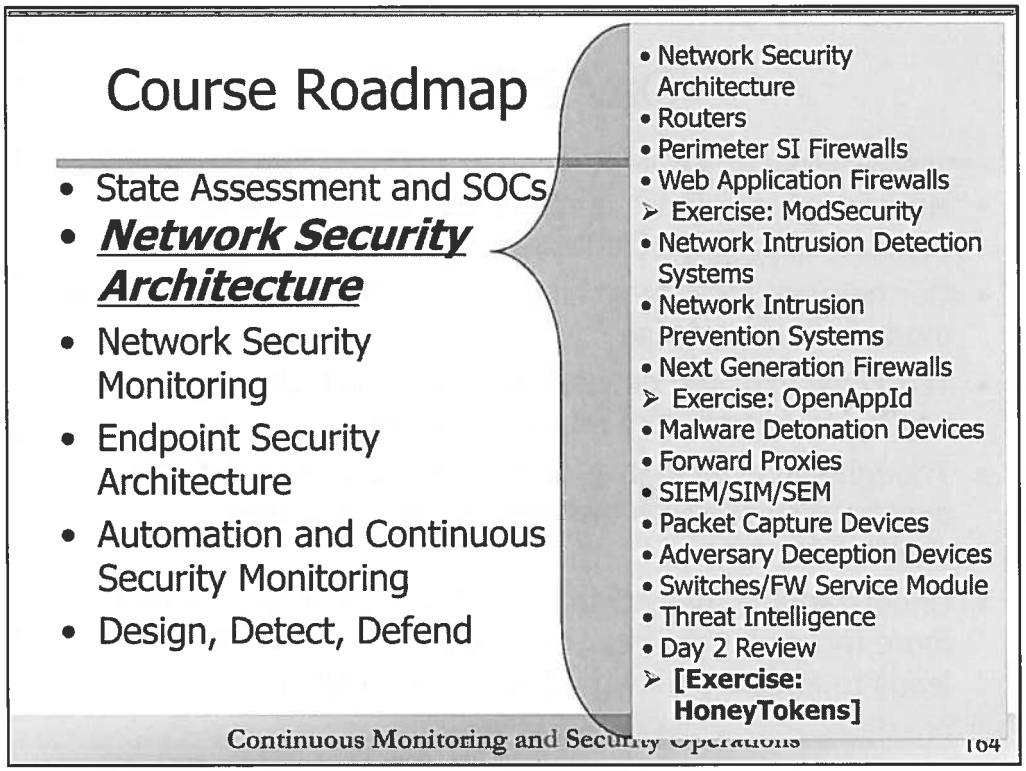
1) Employ a strong egress policy
2) Analyze the outbound
3) Detect the pivot



Day 2 TL;DR

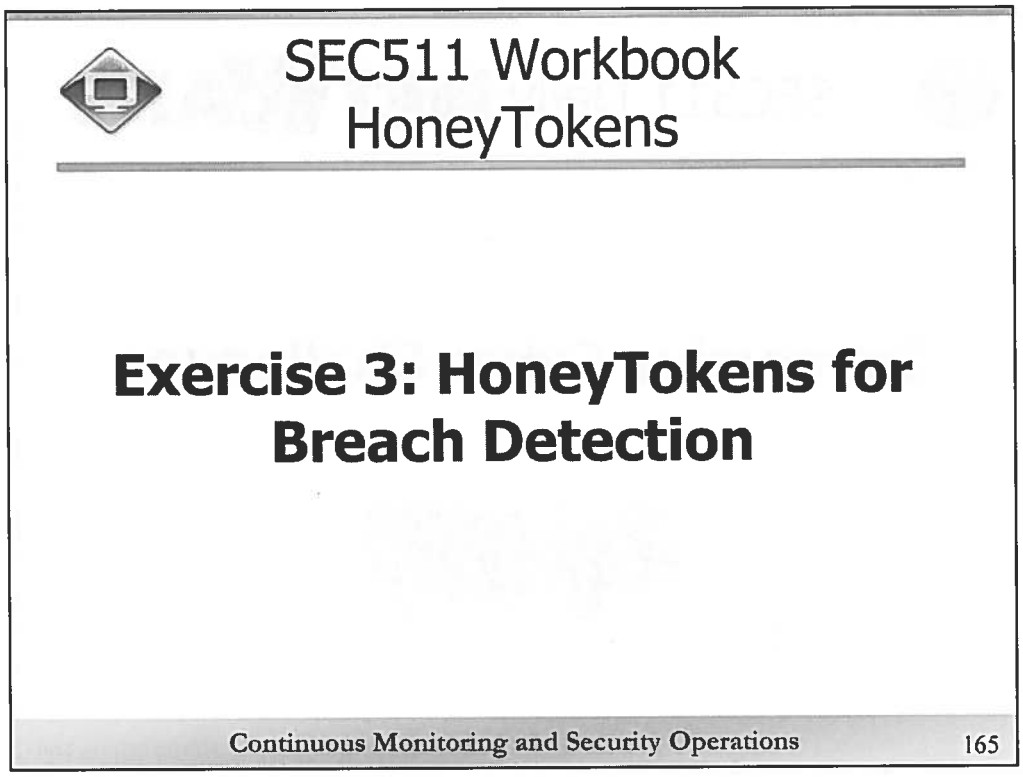
Network Security architecture is key to being able to effectively meet the modern threats currently being faced. A defensible network security architecture does not shy away from preventive capabilities, but will necessarily enable for robust detective capabilities.

Even if we adhere perfectly to principles of modern cyber defense and leverage a defensible network security architecture there is still significant work to be done. First, we will attempt to shore up some of the outstanding weakness that remain in spite of the network security architecture, namely, Endpoint Security Architecture. Then we will have some significant monitoring needs to be able to keep up with all this data, which will lead into NSM and CSM.



Course Roadmap

The next section is final exercise for day 2: HoneyTokens for Breach Detection.



SEC511 Workbook HoneyTokens

Please go to the 511 Exercise Workbook, section 511.2-3.

The instructor will remain in class to assist students. Advanced students may perform these steps on their own.

SEC511 Daily NetWars

Connect to the daily NetWars environment and continue working through the SEC511: Immersive Cyber Challenges.

Please see the first lab (511.1-0) in the SEC511 Workbook for details and instructions on configuring your system to connect to the NetWars environment.

Note: As indicated by the icon above, this lab leverages the class network. OnDemand, vLive, Simulcast, or other online students will need to connect to the SEC511A VPN to complete this lab.