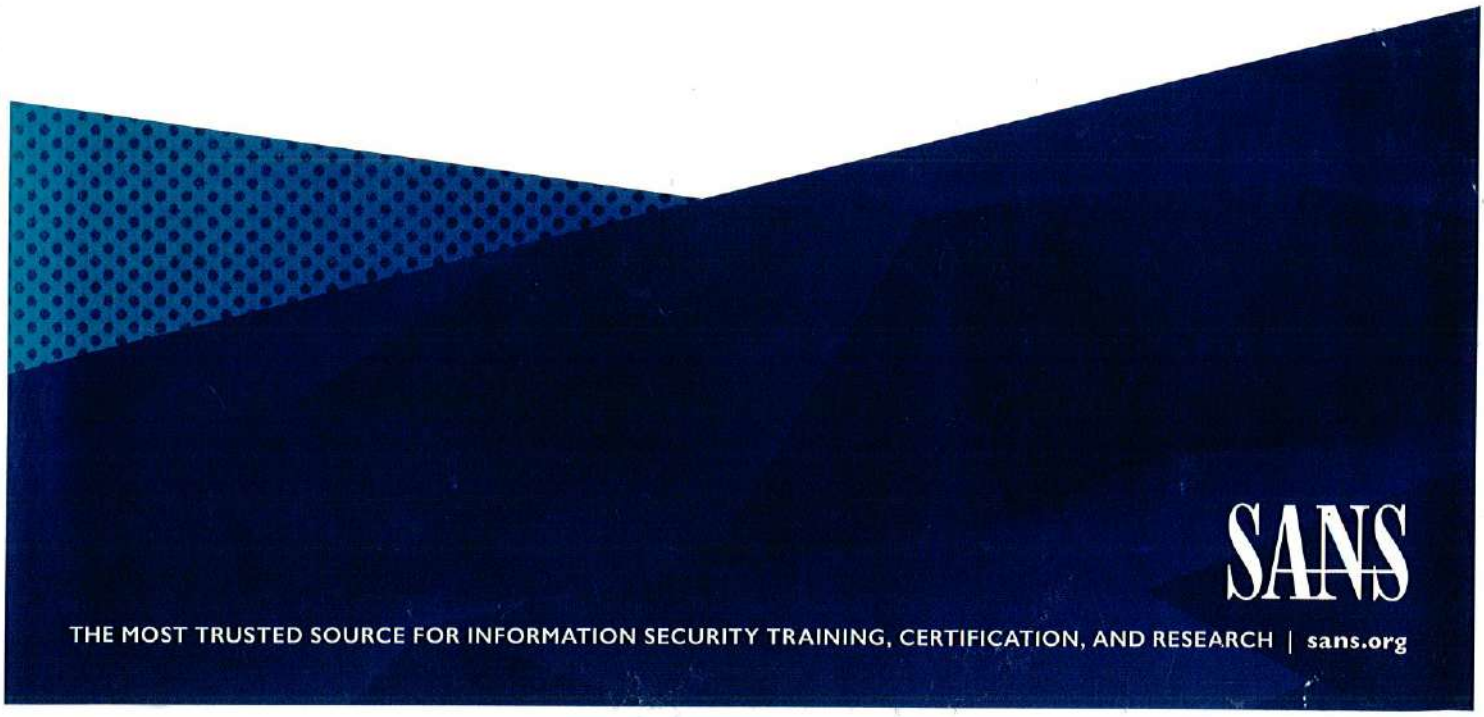
SEC401 | SECURITY ESSENTIALS BOOTCAMP STYLE

401.3
Threat Management



SANS

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

SEC401 | SECURITY ESSENTIALS BOOTCAMP STYLE

401.3
Threat Management

SANS

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

Copyright © 2017, Secure Anchor Consulting. All rights reserved to Secure Anchor Consulting and/or SANS Institute.

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

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

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

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

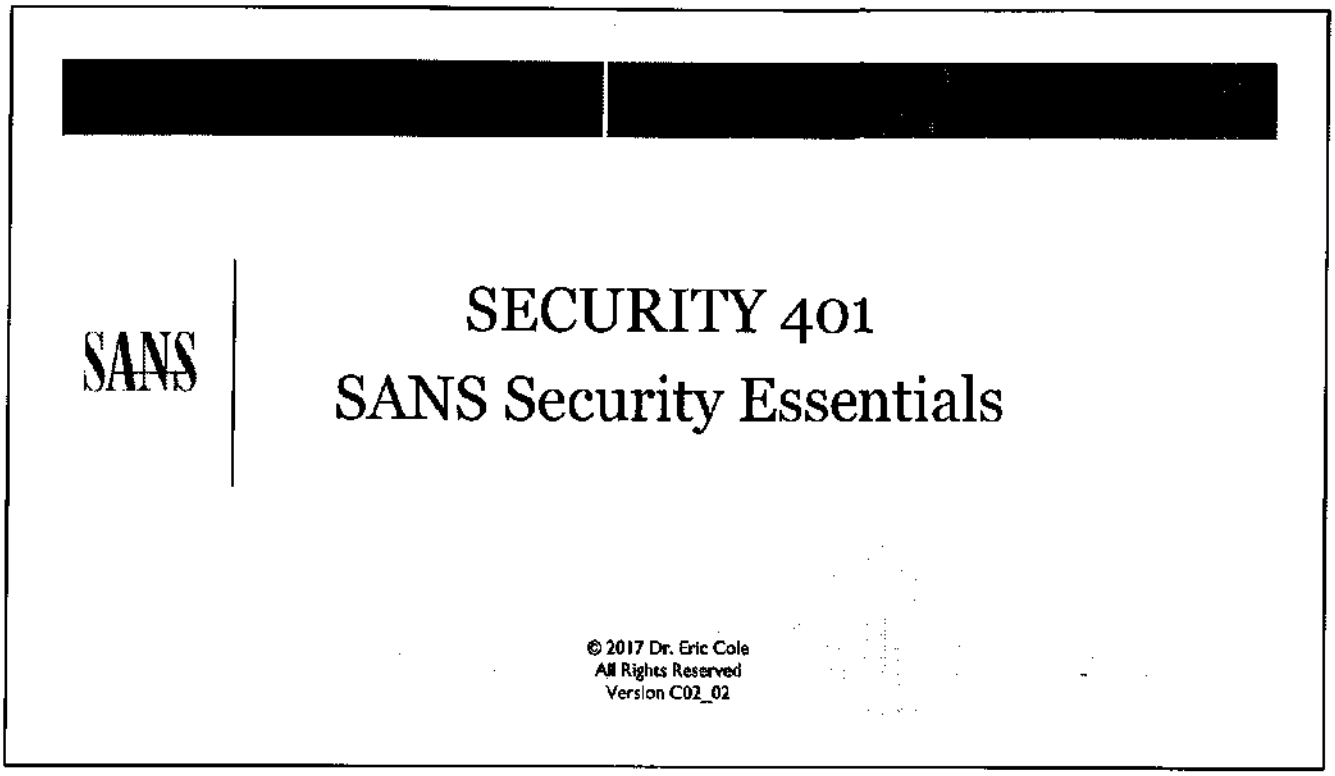
User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of the Courseware, in any medium whether printed, electronic or otherwise, for any purpose, without the express prior written consent of the SANS Institute. Additionally, User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

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

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

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

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



This page intentionally left blank.

SANS

Day 3
Threat Management

© 2017 Dr. Eric Cole
All Rights Reserved

This page intentionally left blank.

© 2017 Secure Anchor Consulting

SEC401 Day 3

- Vulnerability Scanning and Penetration Testing
- *Lab - nmap*
- Network Security Devices
- *Lab – Snort*

- Endpoint Security
- *Lab – hping*
- SIEM/log management
- Active Defense
- *Lab – Command Injection*

SANS | SEC401 | Security Essentials Bootcamp Style 3

This is the outline for Day 3 of SEC401.

© 2017 Secure Anchor Consulting

Security 401: Day 3

SANS

Module 13: Vulnerability Scanning and Penetration Testing

Module 13: Vulnerability Scanning and Penetration Testing

Whether targeting a specific system or just searching the Internet for an easy target, an attacker uses an arsenal of tools to automate finding new systems, mapping out networks, and probing for specific, exploitable vulnerabilities. This phase of an attack is called reconnaissance, and it can be launched by an attacker any amount of time before exploiting vulnerabilities and gaining access to systems and networks. In fact, evidence of reconnaissance activity can be a clue that a targeted attack is on the horizon.

Those in charge of system and network security cannot afford to be any less proficient in discovering and eliminating these vulnerabilities than the attackers are at finding and exploiting them. One strategy is to make full use of the very tools being used against you and to do it regularly. With security, proper visibility is critical. If you do not understand or know about vulnerabilities, this puts you at a disadvantage, especially based on the fact that the adversary is usually aware of these exposures. The more you know about your environment, the better you can protect it.

© 2017 Secure Anchor Consulting

Objectives

- Vulnerability management overview
- Network scanning
- Penetration Testing

This module covers technology, tools, and techniques used for information gathering, network mapping, vulnerability scanning, and the management application of mapping, scanning technology, including exploitation. First, let's set the stage in terms of the management expectation of such a program. Second, we define threat vectors and common sources of reconnaissance on your systems. Then, we examine some of the classic probing tools and their impact. We then show you how to use your own tools to find vulnerabilities before the attackers do. Finally, we will finish with showing you how to do a penetration test to verify and validate the security of your organization.

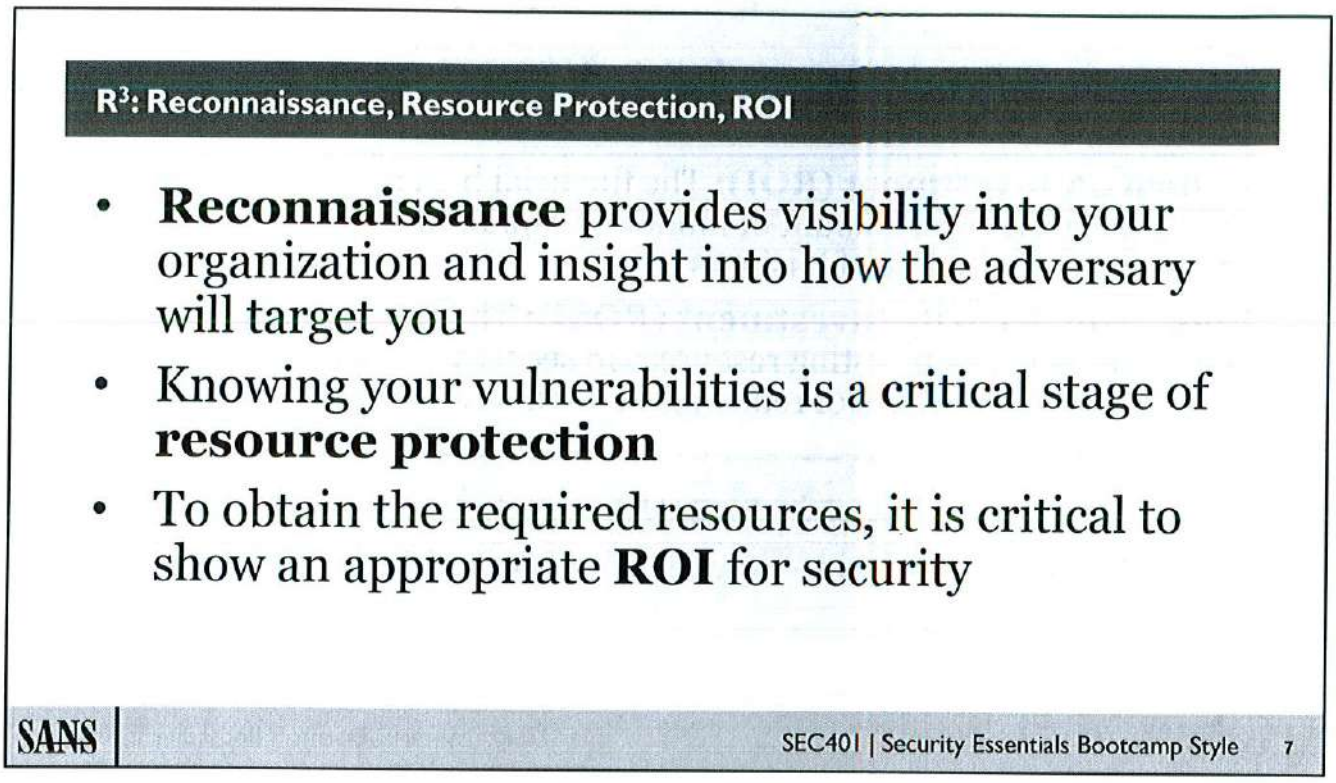
© 2017 Secure Anchor Consulting

Managing Vulnerabilities Overview

The student will understand the concepts and relationships behind reconnaissance, resource protection, risks, threats and vulnerabilities

This page intentionally left blank.

© 2017 Secure Anchor Consulting



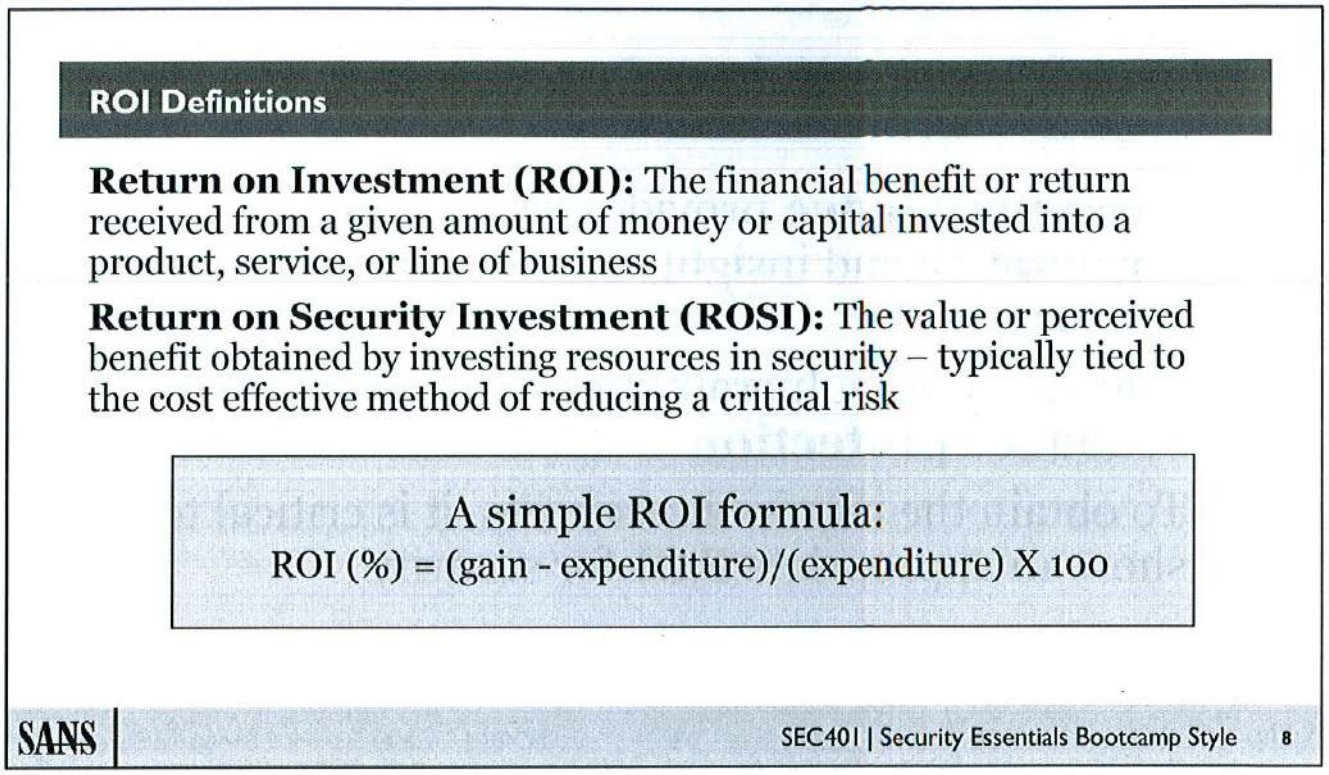
In performing security, visibility is key. If you do not understand how systems are configured, what services are running and where your weaknesses/vulnerabilities are located, how can you properly protect your organization? Knowledge is power. Unfortunately, in many cases, the knowledge and therefore all of the power lies with the adversary. If the adversary knows more about your environment and exposures than you do, you are going to lose. In order to join the winning team, you need to perform the activity that the adversary performs.

Thinking like the adversary starts with reconnaissance. An adversary, as one of the first steps in performing an attack, tries to find out as much information about the target that they can and most importantly start to identify weaknesses that can be exploited. Any successful security program starts with performing proper reconnaissance with the goal of finding and fixing exposures before the adversary can exploit them.

Performing reconnaissance is critical, but it is always important to remember that security is all about managing, controlling and protecting your critical resources. Therefore, in performing reconnaissance, it should be focused on the systems that contain and store your most critical information or resources. Always remember that the difference between a major breach and a minor breach is the data that was compromised. Monitoring, identifying and fixing vulnerabilities that expose critical resources should be a priority.

Once exposures are identified, they need to be addressed. While there are many options for reducing risk, it is always important to make sure that you are getting a proper return on any security investment. Also, once money is spent on security it is important to track and make sure that it provides the value that was anticipated in making the investment.

© 2017 Secure Anchor Consulting

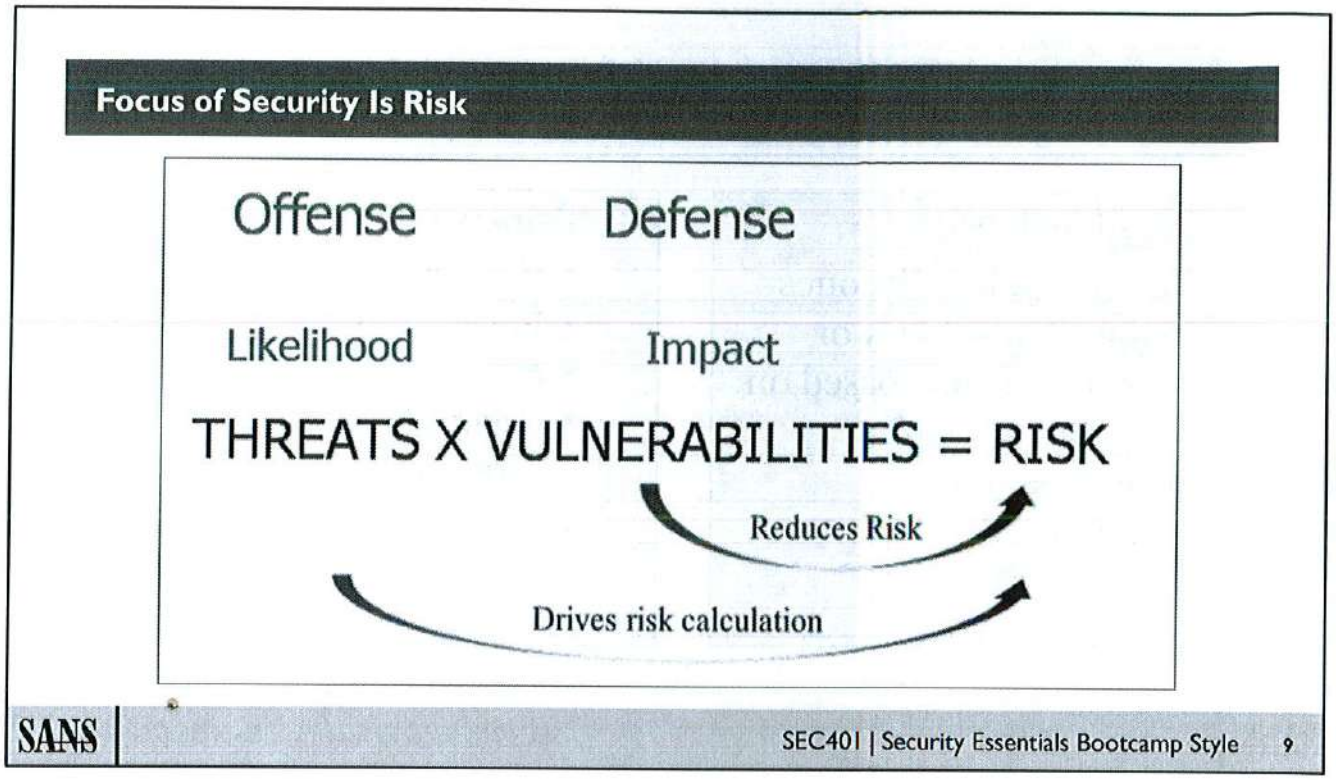


The Return on Investment (ROI) measurement we use in Information Technology and Information Security fields is typically calculated with the formula shown on the slide. When discussing ROI, be sure you share the same definition, because ROI can mean different things in different contexts. For example, Return on Capital Employed, Return on Net Worth, and Return on Equity can all be called "ROI" in financial and accounting circles.

Some common uses of ROI are for the development of a business case, for evaluating whether to go ahead with the purchase of a product, service, or line of business, or for predicting revenue. Also, remember that if a security expenditure in your company prevented damages comparable to what might have been suffered by another firm in your industry, this savings can be a factor in the overall understanding of ROI. Security can sometimes be considered a business enabler, but that is often difficult to capture on financial statements.

If you are not getting the resources you believe your security program requires, a question to ask yourself is: Are you showing the decision makers the correct information that justifies the ROI they will receive from the investment?

© 2017 Secure Anchor Consulting



As we discuss vulnerabilities and vulnerability management, it is important to put it in proper context. This slide provides an overview of the key components of risk and how they all fit together.

Risks, threats, and vulnerabilities are highly interrelated. Their relationship can be expressed by this simple formula:

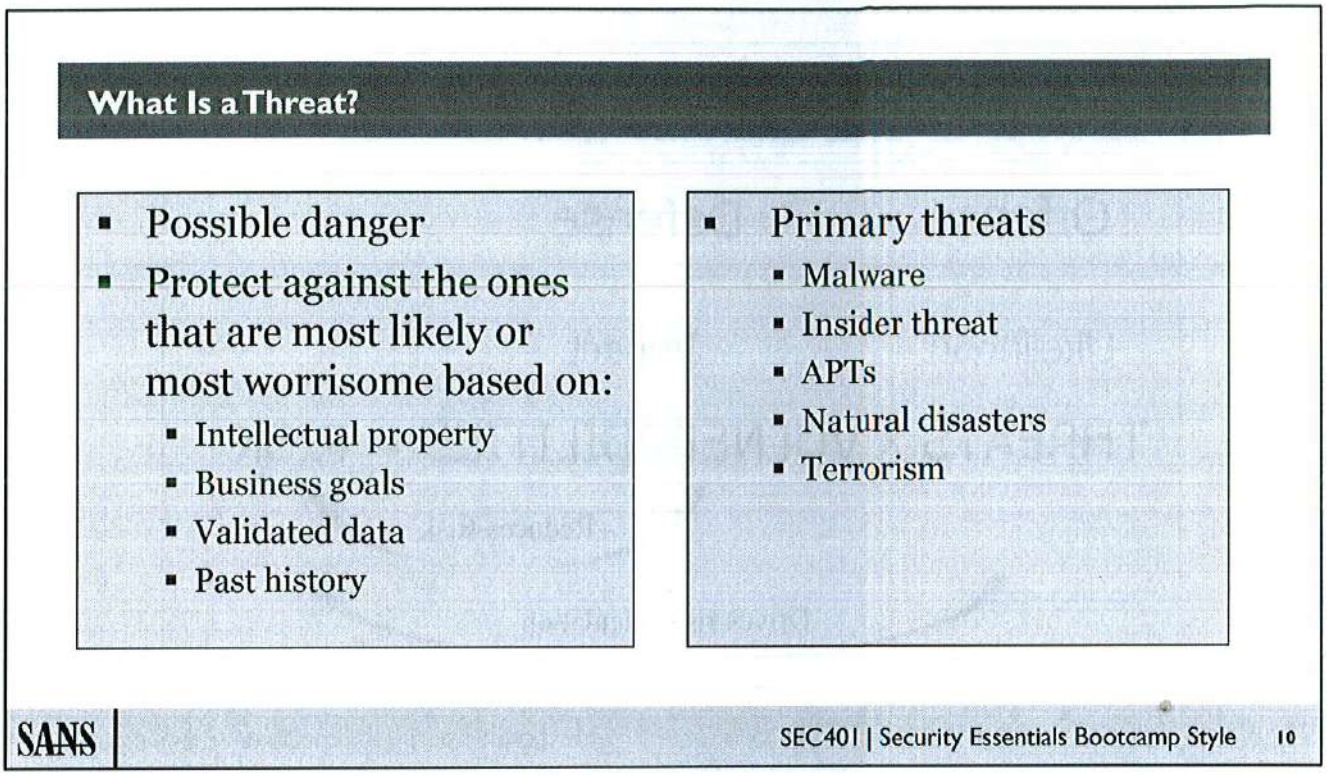
Risk (due to a threat) = Threat x Vulnerability (to that threat)

This formula shows that risk is directly related to the level of threat and vulnerability you, your systems, or your networks face. Here is how the formula works:

- If you have a very high threat, but a very low vulnerability to that threat, your resulting risk will be only moderate. For example, if you live in a high-crime neighborhood (thus, high threat) but you keep your doors and windows locked (so you have a low vulnerability to that threat), your overall risk is moderate.
- If you have a high vulnerability to a threat (by keeping your doors and windows unlocked), but the threat itself is minor (by living in a safe neighborhood), once again you have only a moderate risk factor.
- If, however, you have a high level of threat potential (a high crime area) and your vulnerability to that threat is very high (no locks), you have a very high-risk factor.

We must understand that it is impossible to completely eliminate all risk. Therefore, the job of the security professional is to constantly track, manage, and mitigate risk to an organization's critical assets.

© 2017 Secure Anchor Consulting



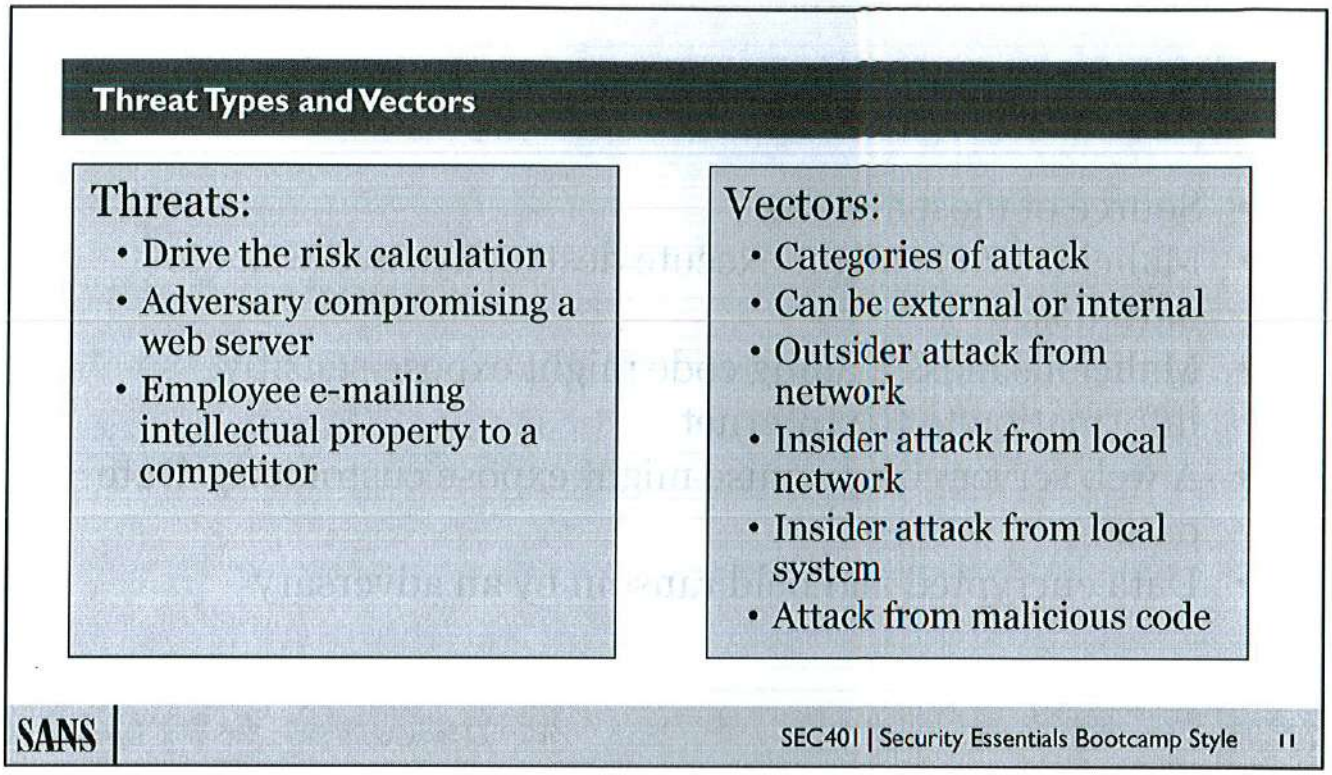
Not all the bad things that happen to computer systems are attacks per se. There are fires, water damage, mechanical breakdowns, accidental errors by system administrators, and plain old user error. But all of these are called threats. We use threat models to describe a given threat and the harm it can do if the system has a vulnerability.

In security discussions, you hear a lot about threats. Threats, in an information security sense, are any activities that represent possible danger to your information or operation. Danger can be thought of as anything that would negatively affect the confidentiality, integrity, or availability of your systems or services. Thus, if risk is the potential for loss or harm, threats can be thought of as the agents of risk.

Threats can come in many different forms and from many different sources. There are physical threats, such as fires, floods, terrorist activities, and random acts of violence. And there are electronic threats, such as hackers, vandals, and viruses. Your particular set of threats depends heavily on your situation: what business you are in; who your partners and adversaries are; how valuable your information is; how it is stored, maintained, and secured; who has access to it; and a host of other factors.

The point is that there are too many variables to ever protect against all the possible threats to your information. To do so would cost too much money and take too much time and effort. So, you need to pick and choose against what threats you will protect your systems. Security is as much risk management as anything. You start by identifying those threats that are most likely to occur or most worrisome to your organization.

© 2017 Secure Anchor Consulting



A key focus in vulnerability management is to enhance your understanding of controlling the threat. Obviously, this is a primary goal of vulnerability scanning and remediation. If there is no vulnerability, the threat cannot manifest itself. However, many classes of vulnerabilities are impossible to detect with a standard vulnerability scanner.

Once a vulnerability is discovered and prioritized, proper controls need to be put in place to mitigate the risk. In terms of controls, we hope detective, corrective, and preventive controls work together to increase our degree of resource protection. Detection is the action on the part of security personnel to track down the threat, and the response (corrective controls) is the countermeasure they use to fight threats once they are found. Prevention is another type of countermeasure (for example, a firewall which security personnel deploys to keep a threat vector from coming in contact with a vulnerability).

© 2017 Secure Anchor Consulting

External Threat Concerns

- Source of most attacks
- Malicious code might execute destructive overwrite to hard disks
- Malicious mass mailing code might expose sensitive information to the Internet
- A web server compromise might expose customer private data
- Data encrypted and held ransom by an adversary

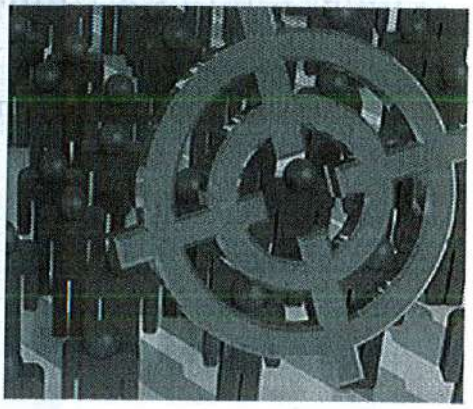
One of our close friends with many more years of experience than we have, told us once, "Get in touch with your fears." Perhaps if you stop to think about the things that you are afraid will happen in your organization, you will realize that these are not random, ungrounded fears. Many times, you are concerned about a particular tragedy because you have observed that safety or security practices are lax. Since threats drive the risk calculation, the more you can understand about the threats that have the highest likelihood, the more focus you can be on vulnerability remediation.

When most people think of threats, they automatically think about external threats because that is what the media tends to highlight: Foreign adversaries, competitors and organized crime targeting your organization. While this is partially true, it is important to remember that while the source of most threats is external, the cause of damage is often the insider threat…

© 2017 Secure Anchor Consulting

Internal Threat Concerns

- Often the cause of damage
- An insider is angry and sets off a logic bomb
- An insider feels "entitled" and sells company trade secrets
- An executive assistant is tricked via social engineering and leaks sensitive data
- An employee clicks an e-mail attachment that allows a hacker to penetrate systems



SANS | SEC401 | Security Essentials Bootcamp Style 13

Continuing from the previous page, a good way to prioritize which vulnerabilities to remediate is to focus on items that have the greatest likelihood and greatest impact on your organization. While threats drive the risk calculation, vulnerabilities drive the risk reduction. One of the problems with many organizations' approach to security is that they focus in on the external threats. As we pointed out, this is often the source, but in most cases, the actual damage and impact to an organization is caused by an insider threat.

When we talk about insider threat it is important to differentiate between the malicious insider and the accidental insider. Most people when they hear insider threat, they think about the deliberate malicious insider. Someone who is consciously aware of what they are doing and want to cause harm. While this will always be a challenge to an organization, the bigger threat in most cases is the accidental insider. Someone who is tricked or manipulated into doing something they normally would not do if they knew the true intent of their actions. Adversaries often target an accidental insider to perform some action, such as opening an attachment, to allow them an entry point to compromise a network.

Reference

1. Insider Threat Programs: 5 Easy Steps to Protect Your Company - http://www.gtscoalition.com/insider-threat-programs-5-easy-steps-to-protect-your-company/

© 2017 Secure Anchor Consulting

Vulnerabilities

- Vulnerabilities are weaknesses in a system
- Vulnerabilities are inherent in complex systems; they will always be present
- Many vulnerabilities are the result of poor coding practices
  - Lack of error checking
- Vulnerabilities are the gateway by which threats are manifested
- Vulnerabilities fall into various categories
  - Known
  - Unknown: zero-day (0-day)

In security terms, a vulnerability is a weakness in your systems or processes that could be exploited by a threat. However, simply having a vulnerability, by itself, is not necessarily a bad thing. It is only when the vulnerability is coupled with a threat that the danger starts to set in.

Suppose you like to leave the doors and windows to your house unlocked at night. If you live in the middle of the woods, far away from anyone else, this might not be a bad thing. In this case, the vulnerability of having no locks is present, but there really isn't any threat to take advantage of that vulnerability.

Now suppose you move to a big city full of crime. In fact, this city has the highest burglary rate of any city in the country. If you continue your practice of leaving the doors and windows unlocked, you have the same vulnerability as you had before. However, in the city, the threat is much higher. Thus, your overall danger and risk are much greater. Similar vulnerabilities exist on your computer systems.

Vulnerabilities are the gateways by which threats are manifested. Without vulnerabilities, threats do not pose a risk to the organization. Of course, vulnerabilities do not have to exist solely in software flaws. Vulnerabilities can be incorrect configurations, poor physical security, poor hiring practices, and so on. When we couple vulnerabilities with threats, we introduce risks to an organization. A zero-day (0-day) attack is an exploit that is not publicly available and the vendor is usually not aware of the flaw. As you can imagine, these are the most dangerous.

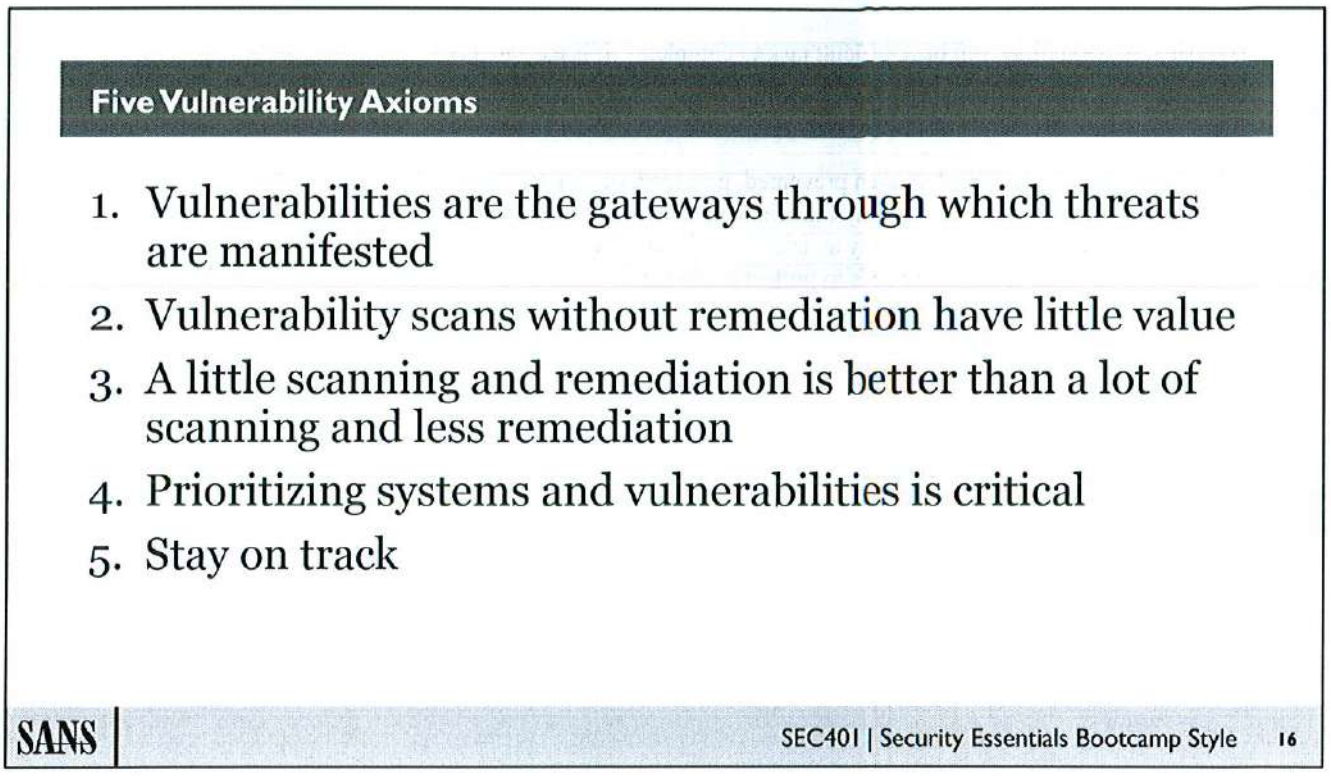
When we talk about identifying vulnerabilities, it is easy to focus on software vulnerabilities and the difficulty of implementing all-encompassing patch-management systems. However, we are introduced to vulnerabilities on a much broader scale, including electronic vulnerabilities from misconfigured software and problems introduced from the rapid deployment of software patches. We are also asked to manage vulnerabilities of the human type—accidental and intentional attacks against information and its storage, for example. When assessing safety, we are also concerned about vulnerabilities in our physical structures including fire, water, temperature extremes, toxins, and electrical vulnerabilities (loss of power).

© 2017 Secure Anchor Consulting

Assessing vulnerabilities can be a difficult task to complete—it is easy to assess obvious vulnerabilities, but a thorough assessment can take considerably more time. Only with a comprehensive vulnerability assessment can we accurately calculate our overall risk.

Vulnerabilities can be reduced or even prevented, provided, of course, that you know about them. The problem is that many vulnerabilities lay hidden, undiscovered until somebody finds out about them. Unfortunately, the somebody is usually an attacker. The attacker always seems to find out about vulnerabilities long before the organization that needs to protect against it.

© 2017 Secure Anchor Consulting



We need to clearly understand threats and vulnerabilities so that we can prioritize resource protection. It is easy to say that we need to remediate, but if people are not succeeding in doing that, then clearly it must be hard to do.

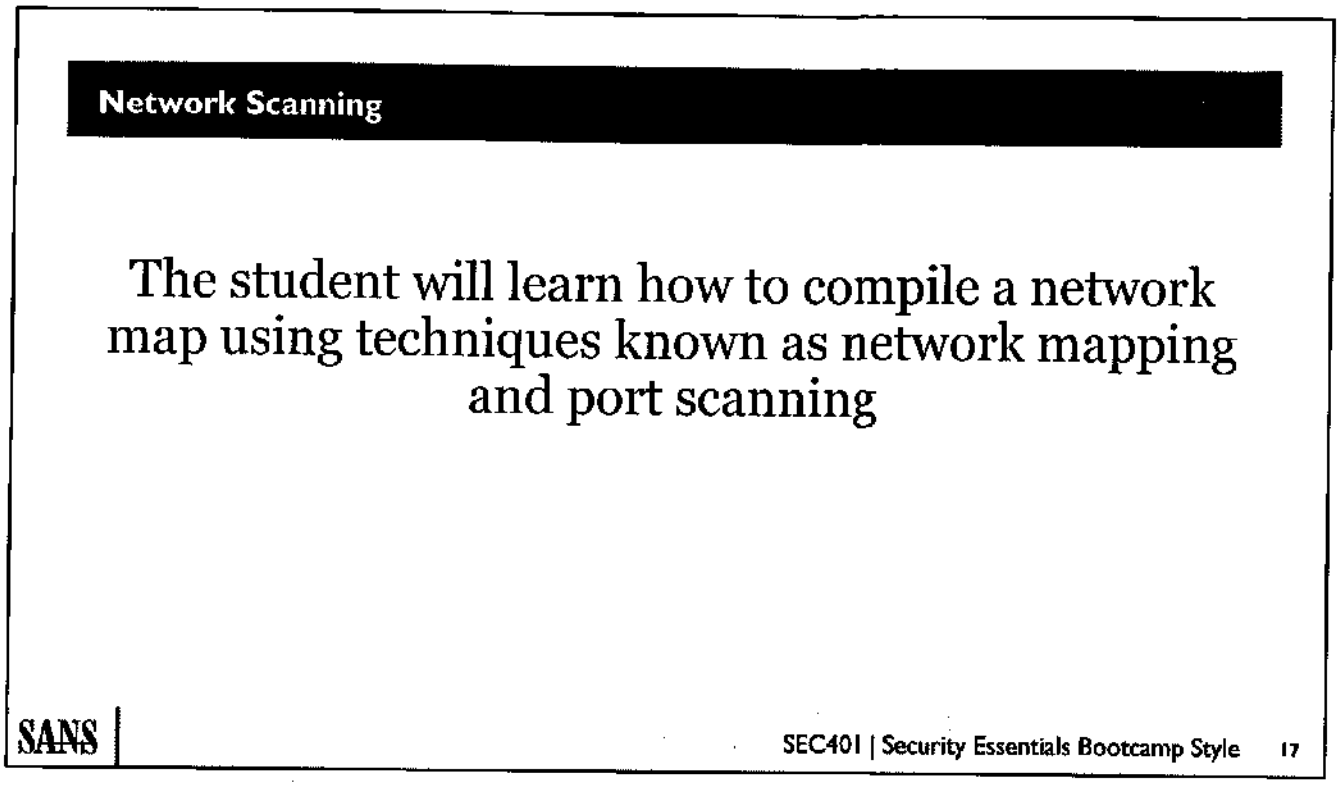
If other organizations have literally scored a negative 100% ROI, we learn not to despise the day of small beginnings. If we can start to get traction on any part of this, we are better off than making no progress at all.

As leaders, people depend on us to prioritize and hold them accountable to stay on track. The following are the 5 rules for staying on track:

1. Vulnerabilities are the gateways through which threats are manifested
2. Vulnerability scans without remediation have little value
3. A little scanning and remediation is better than a lot of scanning and less remediation
4. Prioritizing systems and vulnerabilities is critical
5. Stay on track

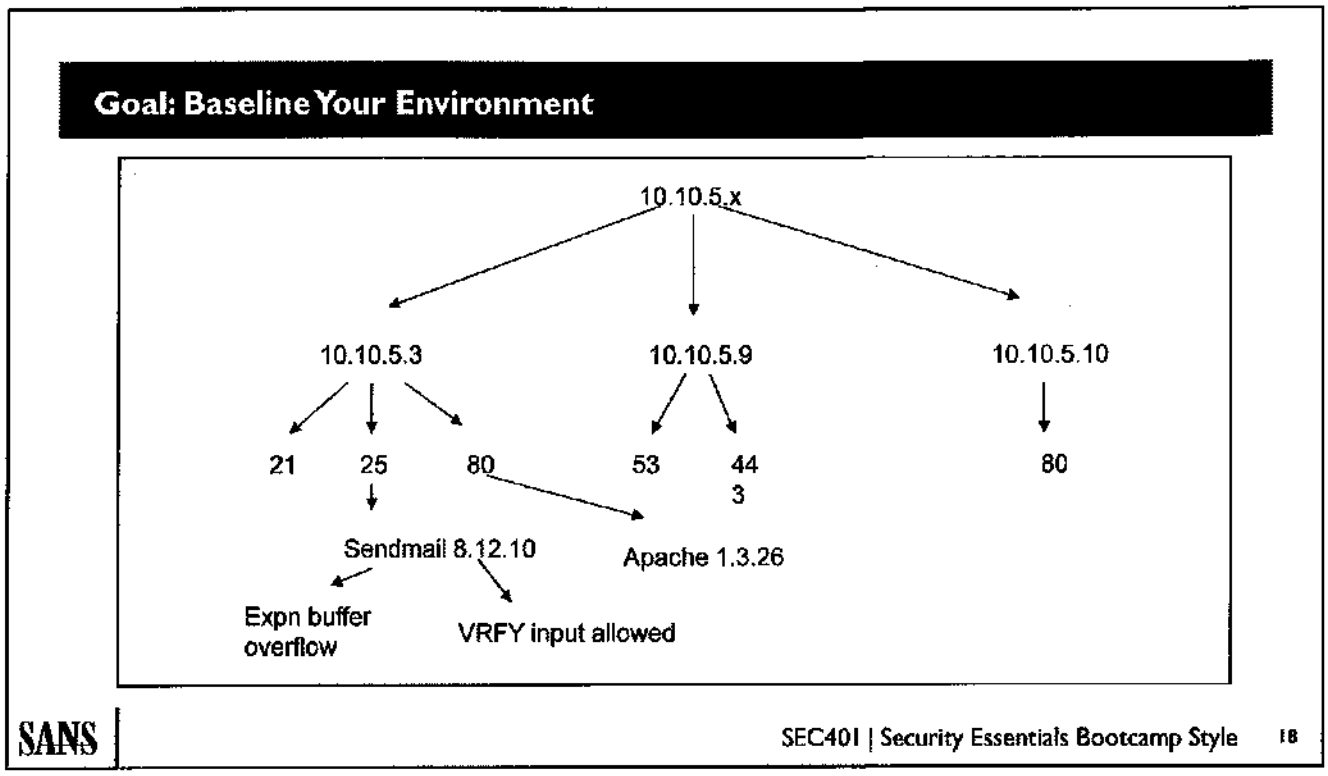
In order to win, the focus needs to be on remediation and creating clear metrics around remediation.

© 2017 Secure Anchor Consulting



Network Scanning
This page intentionally left blank.

© 2017 Secure Anchor Consulting



One of the many objectives of network scanning is to be able to baseline your environment to better understand an organization's exposure points. Only by knowing what the visibility of systems are, ports that are open, and vulnerable services can you properly assess your risk. This also provides a baseline that can be tracked and improved over time.

You must have visibility in order to protect and secure your environment. If you do not have asset inventory or configuration management, it makes it very easy for the adversary to compromise one vulnerability that you are not aware of. This section gives you the tools to create a network visibility map to provide better optics for reducing the attack service and remediating the vulnerabilities that really matter.

Note: This example uses private addresses to minimize the complexity of using a public address that could belong to another company.

Reference

1. SEC501 -- for consistency across courses

© 2017 Secure Anchor Consulting

Core Components of Visibility

To baseline an environment or exploit a system, 3 conditions are required

1. **Visible IP – hping3**
2. **Open Ports – Nmap**
3. **Vulnerability in a Service – Various scanners**

An optional 4th step that would be performed by penetration testers or attackers

4. **Exploit the vulnerable service - Metasploit**

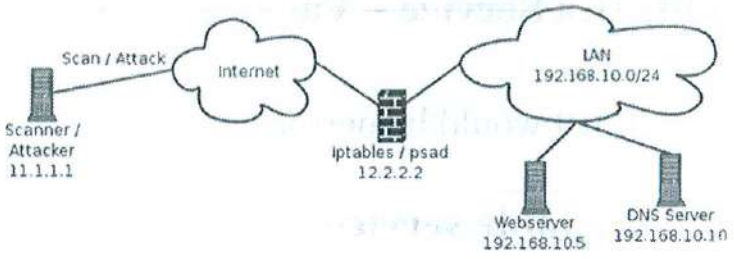
One of the core components emphasized throughout this course is the defense must know more than the offense or you will get compromised. If the offense knows about systems and vulnerabilities that you are not aware of, it is easy for them to exploit the system undetected. However, if you have awareness and can minimize the exposure or properly monitor those attack vectors, you have taken away the advantage from the adversary. The trick is you must have visibility and optics.

For a system to be compromised, it must be visible, have open ports and be running a service that has a vulnerability. The more you can understand which systems are visible and move them to a segmented network, find open ports and close them, or find vulnerable services and uninstall or fix them, the harder you will make it for the adversary to cause harm. This section will give you the tools needed in order to have the proper optics you need to make the right decisions for your organization's security.

© 2017 Secure Anchor Consulting

hping 3: Packet Crafting Port Scanner

- A TCP version of ping
- Sends custom TCP packets to a host and listens for replies
- Enables port scanning and spoofing simultaneously by crafting packets and analyzing the return



hping 3 is a network-analysis tool that fits ping's ICMP concept to TCP and UDP. An hping 3 user can craft packets with a customized destination and source port, window size, identification field, TCP flags (UAPRSF), and more. hping 3 returns results the way ping does: one line of output for every response.

One of hping 3's most useful (or most dangerous) abilities is to spoof the IP address of a third party, making the true origin of the scan hard to detect. This is done by hping 3 first finding a silent host, a host on the Internet that is currently idle. At any given time, many Internet hosts are up but not engaged in any communications. No packets are being sent or received. Although unattended, silent hosts still listen on the network and speak up if asked politely.

Here is how hping 3 finds silent hosts. With a repeated TCP ping of a host, followed by examination of that host's returned TCP sequence and IPID numbers, it is possible to tell whether that host is engaged in communication with any other host. After hping 3 finds a silent host, it starts the scan, crafting each TCP packet so the source address is that of the silent host.

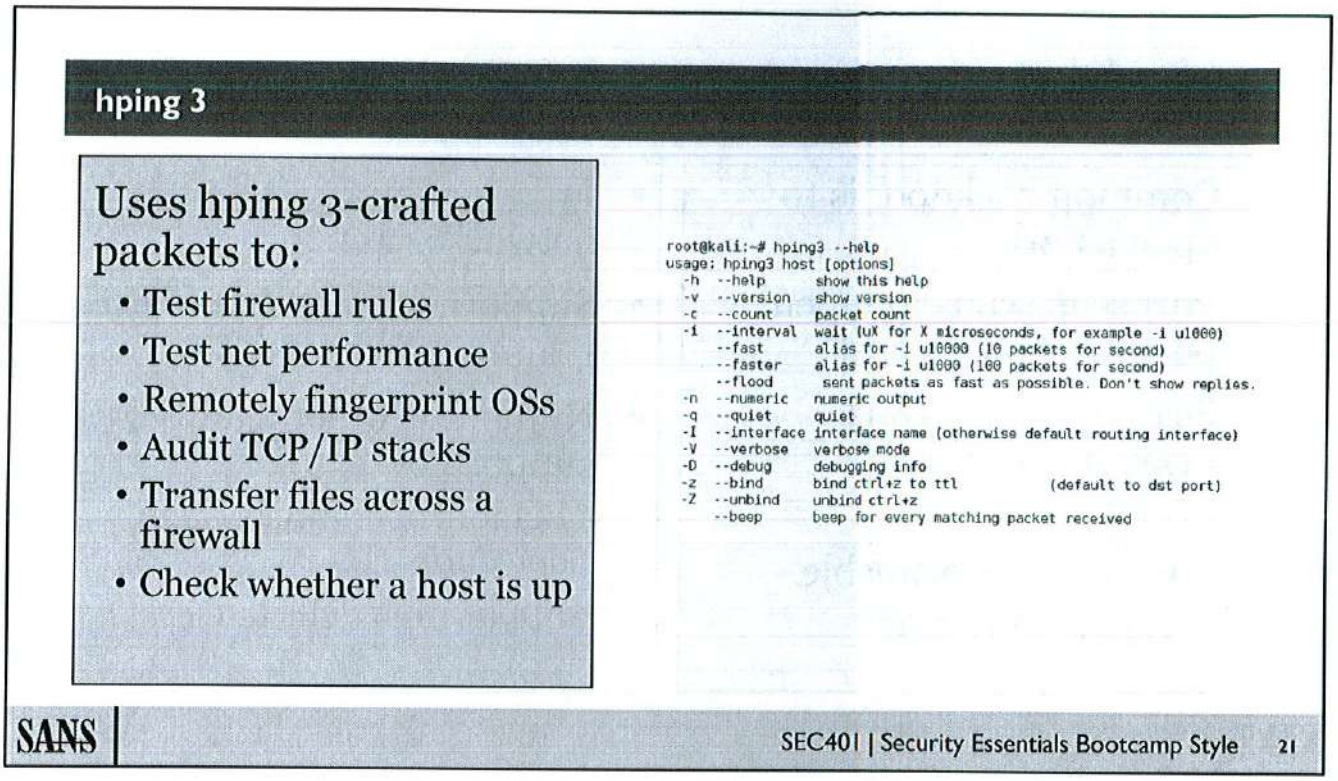
After sending these spoofed packets, hping 3 monitors the silent host to see whether it engages in any communications, again by sending it TCP pings and examining the returned sequence and ID numbers. If the silent host does appear to be talking to some other host, it probably means the target port is open and handshaking with the silent host. (Because the packets sent to the target host were spoofed as coming from the silent host, the target host responds to the silent host, not the host running hping 3.) If not, the target port is not listening or a firewall has intervened.

By using multiple silent hosts, an attacker can run a very stealthy port scan indeed: a distributed and spoofed port scan. Just like an Nmap decoy scan, this type of scan is difficult to track down because it looks like it is coming from multiple hosts.

Reference

1. Network Scanning with Hping3 - https://neelpathak.wordpress.com/tag/hping3-tutorials/

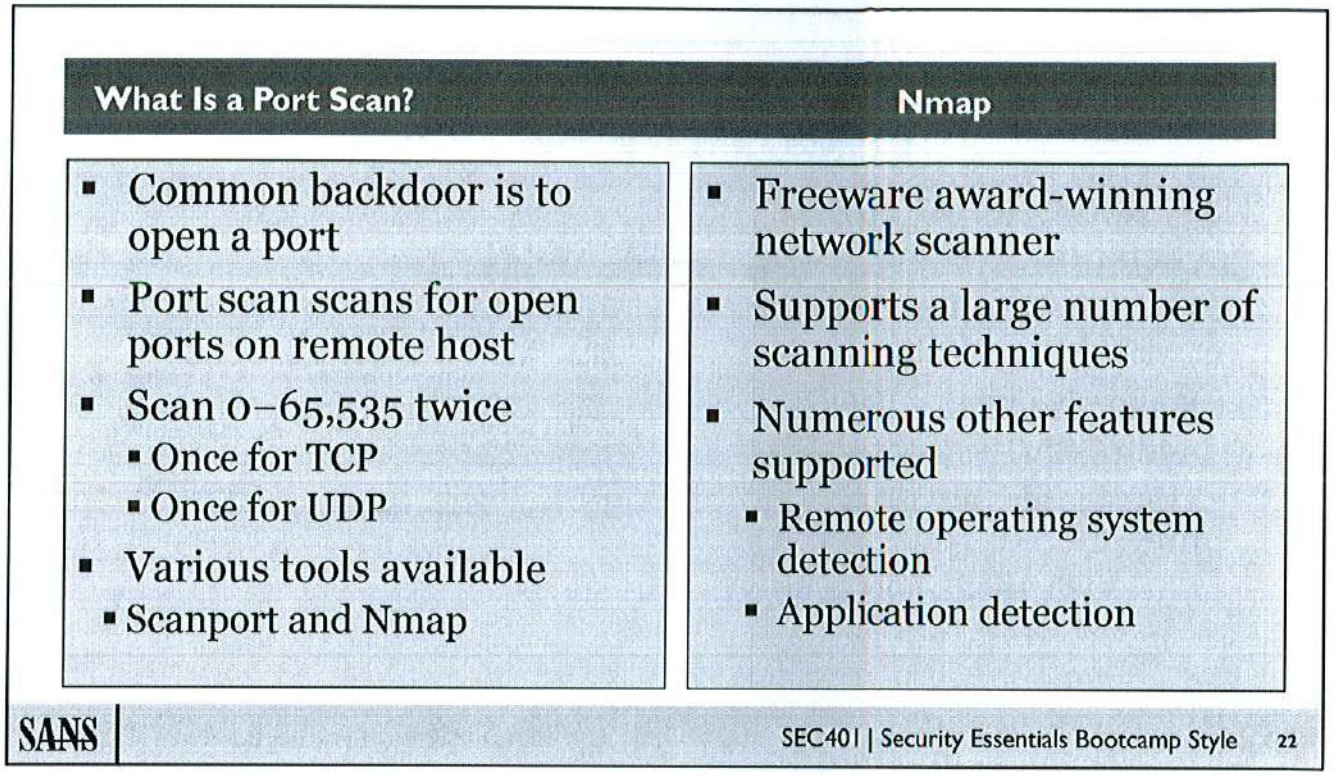
© 2017 Secure Anchor Consulting



One of hping 3's strengths is the control it gives the user over the packets it sends. It allows you to send specially crafted packets and gauge the target's response to various situations. This type of scan can reveal useful information about the target host. This section barely scratches the surface of the tool's features, but by the end, we'll be able to perform some sample scans.

The simplest use of hping 3 is analogous to the standard ping utility. If the remote host is reachable, you get a response for each packet transmitted to it. Unlike ping, hping 3 provides information specific to TCP, such as the TCP flags. It also displays useful IP quantities, such as the packet identification number. This information can be used to determine additional information about the remote host, such as which ports are open and even what operating system is running.

© 2017 Secure Anchor Consulting



Network mapping is simply the process of enumerating all hosts that respond on a given network. Port scanning goes a step further and tells you which ports on each machine have listener processes bound to them. Of course, the assumption is that if the port has a listener, it's probably a server process that's providing some sort of service to other machines on the network.

Nmap is a popular freeware network mapping tool. It supports a large number of scanning techniques, including port scanning (TCP and UDP), SYN, FIN, ACK, and ICMP ping sweeps. It also offers advanced features, such as remote OS detection, stealth scanning, and other functionality.

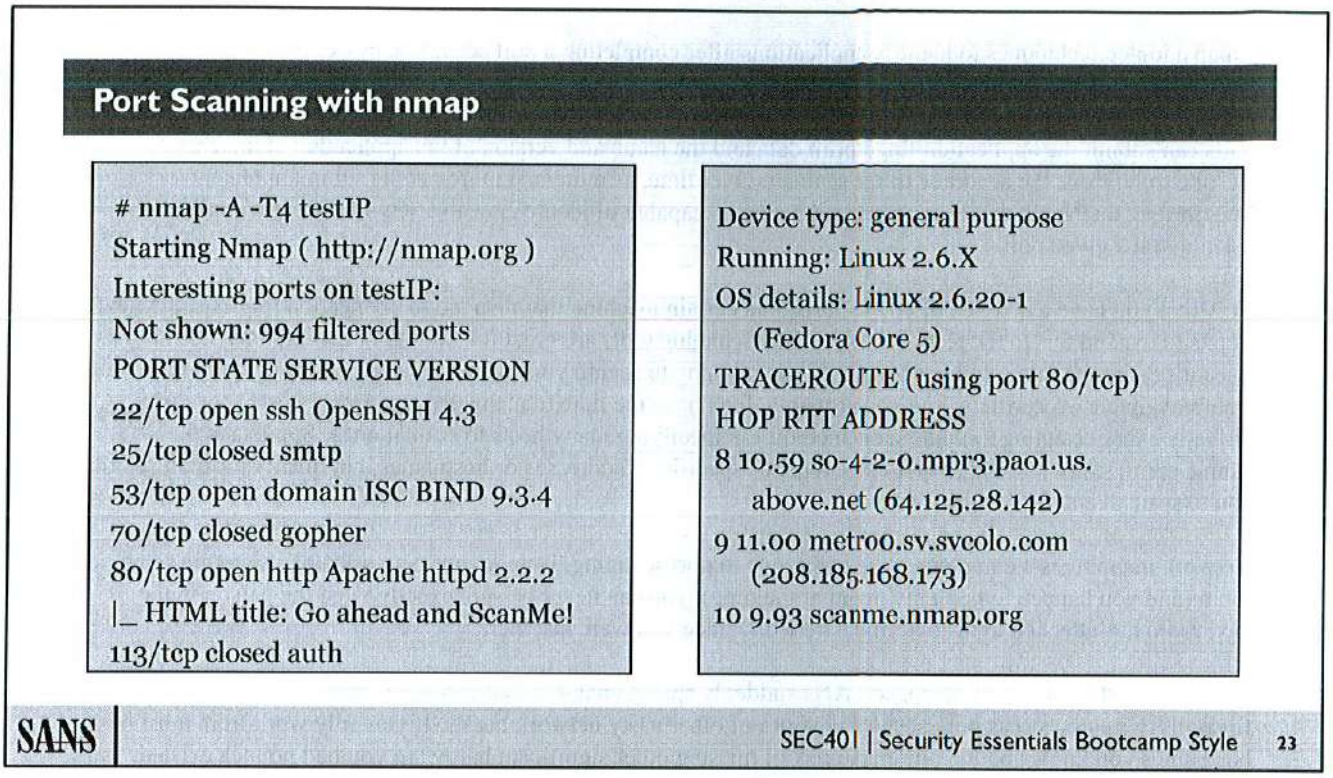
One interesting feature of Nmap is that it queries open ports to attempt to determine which application is running on a port. In some cases, it can even determine the version of the application being run on the port.

TCP vs. UDP Scans

The concepts we're about to impart are pretty straightforward and easy to understand. The principles apply to both the TCP and UDP protocols because they share the same concepts of what a host and a port is. There are some subtle differences, of course, but we'll look at those later. Forget them for now, and assume that we're talking about TCP scans, and that everything applies equally to UDP scans. We'll clear up any discrepancies later.

```
HOST = "samurai.sample.org"
for PORT_NUMBER in 1 to 65535 {
if(connect_to(HOST,PORT_NUMBER) == SUCCESS) {
print "Port PORT_NUMBER is listening"
close_connection(HOST, PORT_NUMBER)
```


© 2017 Secure Anchor Consulting



For many years, port scanners were simple tools that reported only whether a service was listening or not for a given port, just that something is there. For example, if the scanner found port 80 listening on samurai, you might guess that samurai is a web server because port 80 is known to be the standard port for HTTP. This is a good guess, but you still don't know for sure. The fact is, you can't tell exactly what type of service responded just from a simple port scan. The only thing you know for sure is that something is listening. Most scanners print a nice report that names the service running on each port, but this information can be misleading. This information is based on what service typically listens at that port and often comes from an /etc/services file on a UNIX host or the equivalent on other platforms. Basically, this is a text file that maps port numbers to their well-known service names. Printing the name makes the report look good, but the information isn't necessarily reliable.

This becomes even more confusing when you consider that some users actively try to trick you. Even legitimate users sometimes fall prey to this temptation. One common practice when deploying illicit services is to make them listen on port 80 instead of giving them their own ports. Why? HTTP is a vital protocol in lots of environments, and many routers and firewalls are configured to open up "holes" to let machines on one side talk to HTTP servers on the other side. If the administrator is sloppy, sometimes those rules are the equivalent of "permit any traffic going to port 80."

If a user wants to run P2P software on your corporate network or an attacker wants to make sure he can access his backdoor Trojan later, he might choose a port that's known to be associated with another service in the hopes that it slips through the cracks in your perimeter defenses. Similarly, it's common to offer well-known services on unusual ports to try to hide them from administrators. Successful attackers often leave backdoors like hacked SSH servers listening on arbitrary high-numbered ports so they can get back into your machines later.

Although we still can't determine the application that is running on the remote system with 100 percent accuracy, tools like Nmap have included functionality to decipher the responses returned after connecting to a port to identify the software that is listening, sometimes including version and patch level information.

© 2017 Secure Anchor Consulting

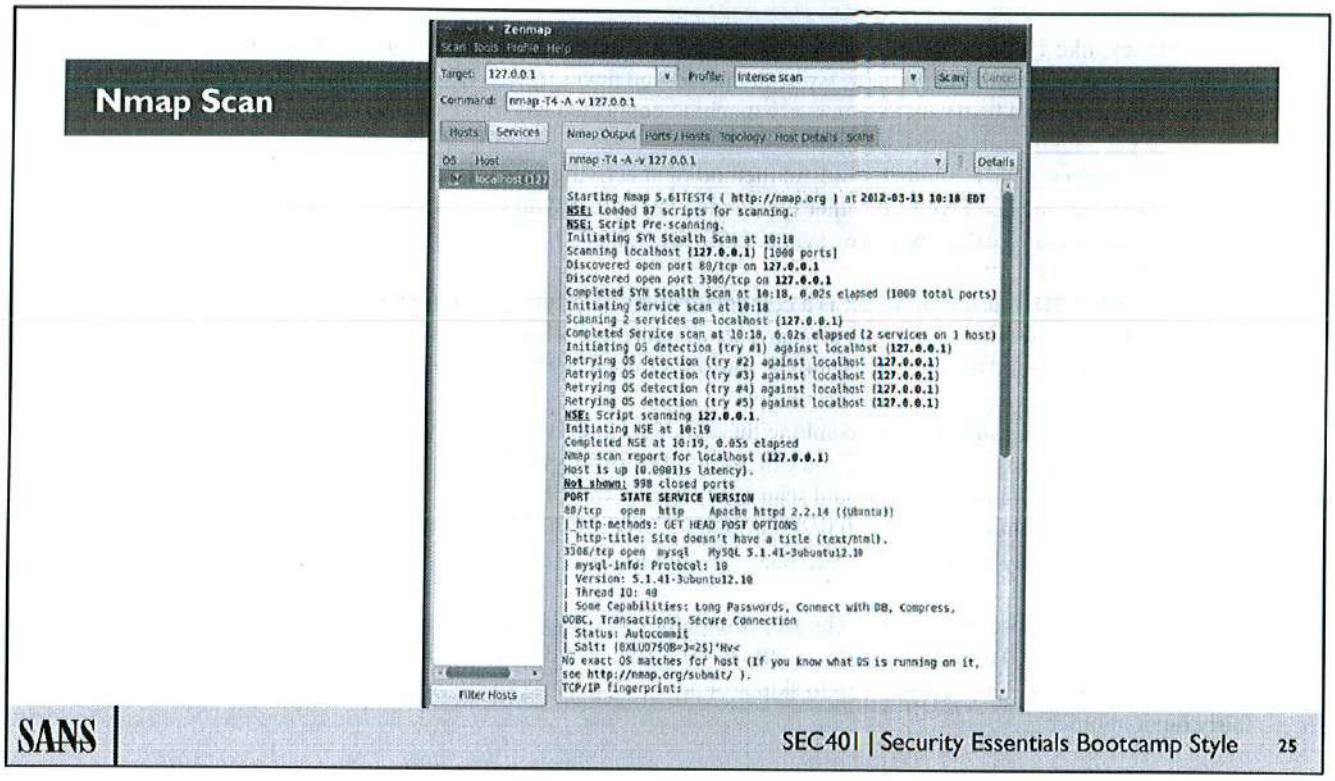
Nmap adopted techniques to identify applications after completing a port scan. The power of Nmap is in the large number of users who scan systems. When Nmap tries to identify an application and is unable to do so, it displays "application fingerprint" information. Someone running Nmap to identify applications listening on ports can submit the application fingerprint data and the name and version of the application at the Nmap website to improve the accuracy of the scanner. Over time, as more and more people submit application fingerprints, the Nmap database grows and becomes capable of identifying a variety of different applications listening on a given port.

Now that you have a nice list of all the ports on a certain machine that respond to connection requests, the next step is to expand this to the rest of the network. Scanning tools are great for finding out what's on a host, but as a security administrator, you need more. If you're going to secure your whole network, you've got to know what's available everywhere, and you probably don't have the time to manually scan every host. That's why virtually every scanning tool has some method for specifying many hosts to scan at once. Some take the simple approach of allowing you to give a list of specific IP addresses or hostnames, but quality tools allow you to scan an entire IP address space.

Network mapping is a complementary technique to port scanning. Unless your network all fits into the same room (and you happen to be in that room at the time), you can never be sure exactly what the person in the next room has attached to his part of the network since you were last there.

We've seen otherwise well-managed LANs suddenly sprout unauthorized consumer grade cable/DSL firewalls, wireless access points, and a plethora of other nasty network beasts. If you only work from a list of hostnames you know about, you run the risk of missing other significant hardware you had no idea existed. Scanning every possible IP address in a range helps ensure that you'll find out when such rogue devices appear.

© 2017 Secure Anchor Consulting



After Nmap finishes scanning the host and some brief scan statistics, it prints a list of the listening ports it found. Notice that there are two states indicated for each port on the target machine. The vast majority of ports are in the "closed" state, which means Nmap was able to determine with a fair amount of certainty that the port is not listening on that machine. Closed ports usually aren't that interesting, so Nmap doesn't report them individually. It simply tells you how many there were all together. The other state shown in this example is "open"; the meaning should be obvious: Something is listening on that port.

What is not so obvious is that there is a third possible state. We don't show any in this example, but a port also can be marked as "filtered." This is a kind of in-between state, which means that it might be listening, but Nmap can't tell for sure. The most common cause is that a firewall or filtering router is blocking access to that port from the scanning host, so Nmap can't get back any of the normal TCP protocol responses associated with either closed or open ports. Filtered ports are common when scanning a machine over the Internet, but not quite so common when scanning over LANs because those environments are less likely to impose network access restrictions on their internal users.

Remember what we said about how port scanners associate port numbers with the well-known services registered to run on them. If the Nmap output indicates that a port is "unknown," this means that the port is not a well-known port, so Nmap has no idea what is running on it.

Scanning a single host is great, but to be really effective, you need to be able to scan a group of hosts at once, maybe even an entire network. Nmap provides several convenient ways to tell it which hosts to scan.

- **Name several hosts on the command line.** This is the simplest way. Just provide more than one hostname or IP address on the command line, such as Nmap host1 host2... hostX:
- **Use wildcards**: This is another simple way, although it only works with IP addresses and not hostnames. When you specify an address to scan, you can substitute * for any of the octets, causing Nmap to scan addresses with all possible values of that octet. For example, 192.168.1.* would scan the entire 192.168.1.0/24 network (the so-called "Class C" address). You even can be somewhat

fancy, like 192.168.*.* or even 192.*.2.* or *.*.1.8, although it's somewhat doubtful how useful that last example would be. Finally, we recommend you never try *.*.*.* on any machine connected to the Internet. You'd be trying to scan every machine connected anywhere on the Internet and would be sure to get a lot of calls (at the least).

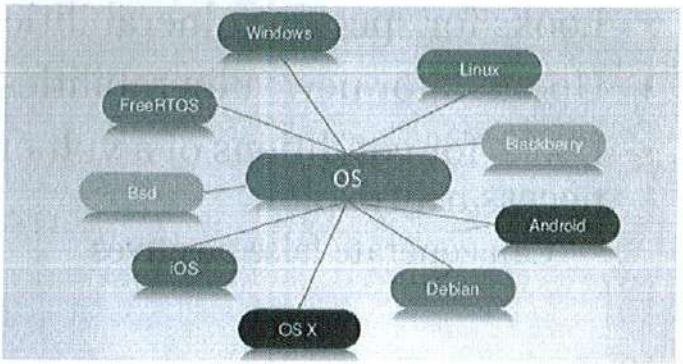
- **Use ranges**: This is another powerful method of specifying hosts. To scan just a subset of all possible values for one octet, you can either give a range of values with the - syntax (192.168.1.10-35), specify several specific values with the , syntax (192.168.3,4,5.10), or use both if you're feeling lucky (192.168.3,4,5.10-35).
- **Specify CIDR notation**: CIDR is a compact way of specifying an entire network without enumerating every possible address. In fact, to scan a set of addresses whose network numbers don't fall easily into 8-bit octet boundaries (if, for example, you want to scan 192.168.40.0/22), CIDR notation is the most convenient way to express this to Nmap.
- **Use a combination**: You can combine these methods into virtually any fashion that makes sense to best meet your needs. For example, you can ask to scan 192.168.0-5.* 192.168.10-254.*. For those of you playing along at home, that would scan every host in the 192.168.0.0/24 through 192.168.5.0/24 networks and in the 192.168.10.0/24 through 192.168.254.0/24 networks. In other words, it's a convenient way to say, "Scan 192.168.*.* except for 192.168.6-9.*."

No matter which of these methods you choose, Nmap does a pretty job of scanning all the hosts you asked for as efficiently as possible. To make your scan a little more difficult to detect, try giving the -randomize_hosts command-line option. This causes nmap to skip around the IP address space you gave it and scan the hosts in fairly random order.

© 2017 Secure Anchor Consulting

Operating System Identification

- Looks for subtle differences in target responses
- Develops a fingerprint
- Compares the fingerprint against a pre-built database of operating system fingerprints



SANS | SEC401 | Security Essentials Bootcamp Style 27

Maybe the most interesting of Nmap's features is its ability to probe a target and determine what operating system and version it runs. Think for a second how important this can be to attackers. If they know a particular machine is running Windows 10, for example, they can save themselves a lot of time by trying only Windows-oriented attacks and skipping all the UNIX and Linux exploits, plus those specific to other versions of Windows.

Strictly speaking, OS detection isn't a type of port scanning technique. Rather, it's an additional feature that happens to be an excellent complement to port scanning. If you know not only the services a target is offering but also the OS it runs, you know a great deal.

Nmap is pretty clever at OS detection. The basic idea is that there exists a set of simple but abnormal IP packets you can send to a remote host, which generates a particular set of responses that can be mapped to a particular operating system. One example of such a test would be to send a FIN packet to an open TCP port, pretending to tear down a session, which was never established in the first place. According to the Nmap website, some OSs respond to this with a RST packet, whereas others ignore it completely.

Of course, using just a single test doesn't tell you much. In the case of a FIN probe, for example, if Nmap receives a RST in response, it knows that the target must be Windows, BSDI, Cisco, or one of a few other OS types, but it doesn't know exactly which.

That's why Nmap performs several different tests and correlates their output. In many cases, Nmap can give you an eerily precise answer. The logic for figuring out which OS responds in which way to which probes isn't hard-coded into Nmap, either. It comes with a database of hundreds of "signatures" of different operating systems and networking equipment and relies on its user community to help keep it up-to-date by submitting new signatures as new operating systems become available.

Reference

1. Operating Systems Quizzes and Trivia - http://www.proprofs.com/quiz-school/topic/operating-system

© 2017 Secure Anchor Consulting

Technet24

Vulnerability Scanning

- Goes beyond mapping and port scanning
- Looks for specific vulnerabilities within a service
- Updated for new vulnerabilities
- Looks for conditions of a vulnerability, but does not necessarily exploit
  - Can generate false positives

Vulnerability scanning takes network mapping and port scanning one step further to find out even more information about the target. Not only will a vulnerability scanner look for live systems and ports and services, it can also look for vulnerabilities associated with those services. Some vulnerability scanners can even attempt to exploit the found vulnerabilities. Like virus scanners, vulnerability scanners are constantly updated with new tests.

A variety of both commercial and free software scanners exist. If you are shopping for a scanning toolkit, it is reasonable to assume that most of the major tools scan for the same number of vulnerabilities. They will all come up with false positives that have to be investigated manually. Before you invest your money (or time), you really want to consider the following four things:

- How is the product licensed? Is it flexible enough for your planned growth? Can it be upgraded easily?
- How interoperable is the product? Does it support the Common Vulnerabilities and Exposures (CVE) standard for cataloging vulnerabilities?
- Can you easily compare the results of a scan today with the results of one four weeks ago, or is it a manual process?
- Does your manager like the report output?

© 2017 Secure Anchor Consulting

Vulnerability Scanners

Only scan systems you own

- Scan
  - "Test for services"
  - Multiple ports on multiple machines
  - May have knowledge of vulnerabilities and test to see whether the vulnerability is present
- Report
  - Provide results in a clear, understandable fashion

Up to this point, this module has, hopefully, instilled in you a healthy respect for an attacker's ability to seek out and exploit vulnerabilities on your systems. Now, we show you how to beat those attackers to the punch.

The best way we've discussed to protect yourself against the threat vectors is to regularly audit your perimeter security. This is a fancy way of saying that you need to hack your own systems. The cardinal rule of scanning or vulnerability assessment is to be certain to only scan systems that you own or are authorized to scan. Otherwise, you will probably set off someone else's intrusion detection systems, which can get you in trouble.

We begin with a discussion of the general principles of scanning. Note that vulnerability scanning can be hazardous to your career. The difference between a penetration tester and an attacker is permission! Be sure that you have it. If you are just now coming up with a scanning policy in your organization, get written permission from the highest level possible in your organization (like your chief information officer).

© 2017 Secure Anchor Consulting

Technet24

How to Do a Vulnerability Scan

- Get permission, explain what you are doing, and "find our vulnerabilities before attackers do"
- Put out the word ahead of time, publish your phone number; people don't like surprises
- Click target selection, choose a system, tell it to expand to the subnet
- Heavy scan, but do not allow Denial of Service scan
- Only scan when you are in the office or by the phone
- Prioritize vulnerabilities based on your environment

You should also be sure to give people plenty of warning before starting your scan. Things can go very wrong when you are scanning. Scans often crash systems, and people will be a lot more forgiving if you warn them ahead of time and make sure it is easy for them to find you. If you are not in the office, or people do not know how to contact you, then you could create a serious problem for yourself and your organization.

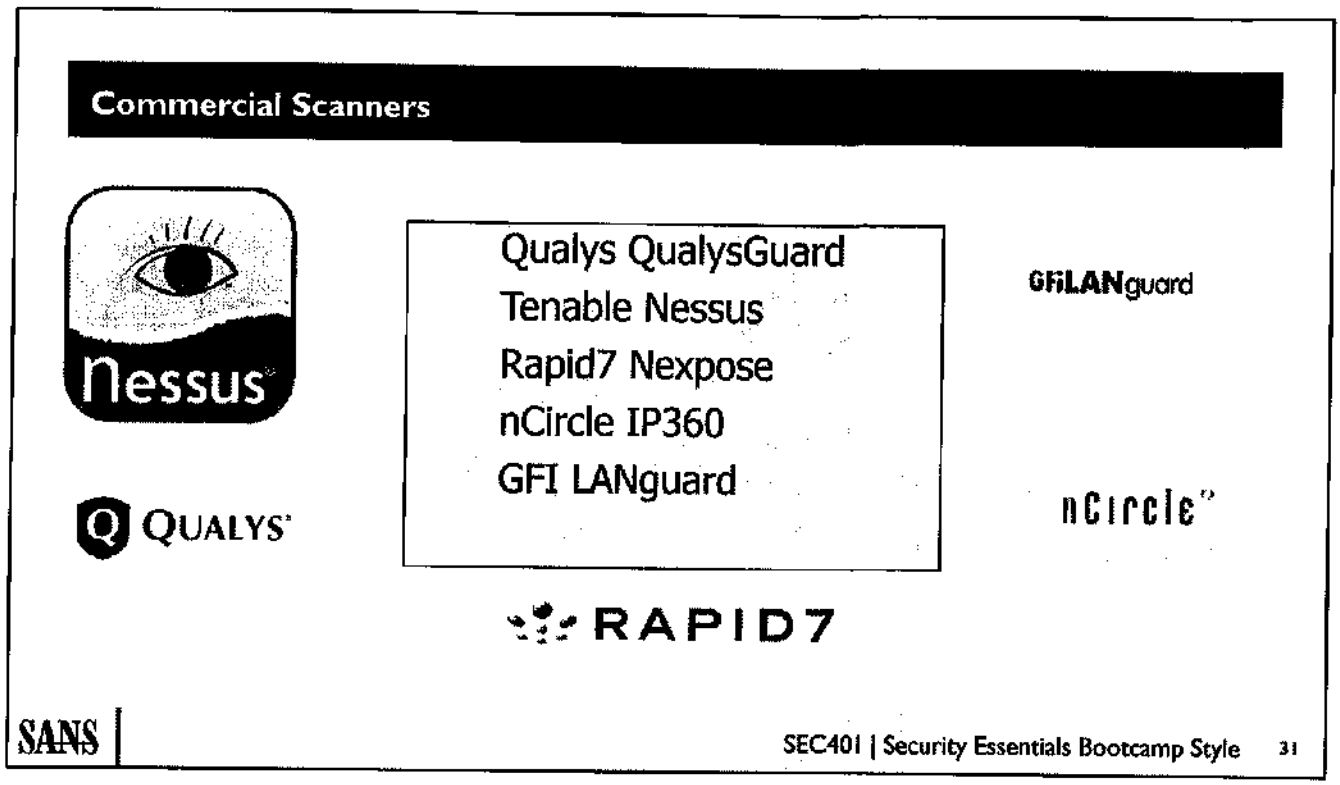
There is no point in configuring a scanner to hit all of your addresses unless your organization is small. Instead, scan a subnet at a time, a workgroup at a time, or whatever other subset of your network makes sense. This way, you won't bog down the network, and you won't have an overwhelming number of vulnerabilities to fix.

If you do scan the entire facility at once, you will have a huge list of problems, and although everyone can talk about fixing them, the work will never get completed. This approach can be dangerous. Consider the following scenario: You run a scan on a large scale and get a huge list of all the problems, each with its own risk level. You present the report to management and tell them life as we know it will end if they aren't fixed.

They agree. They create a task force. There are meetings, and everyone agrees to get things fixed. Great, so far. However, then you run into deadlines and emergencies, and no one ever gets around to fixing the vulnerabilities. Now, you cannot play that card again. If you run another scan, no one will take it seriously.

So, start small. Scan your own shop. Fix the problems, and then move on.

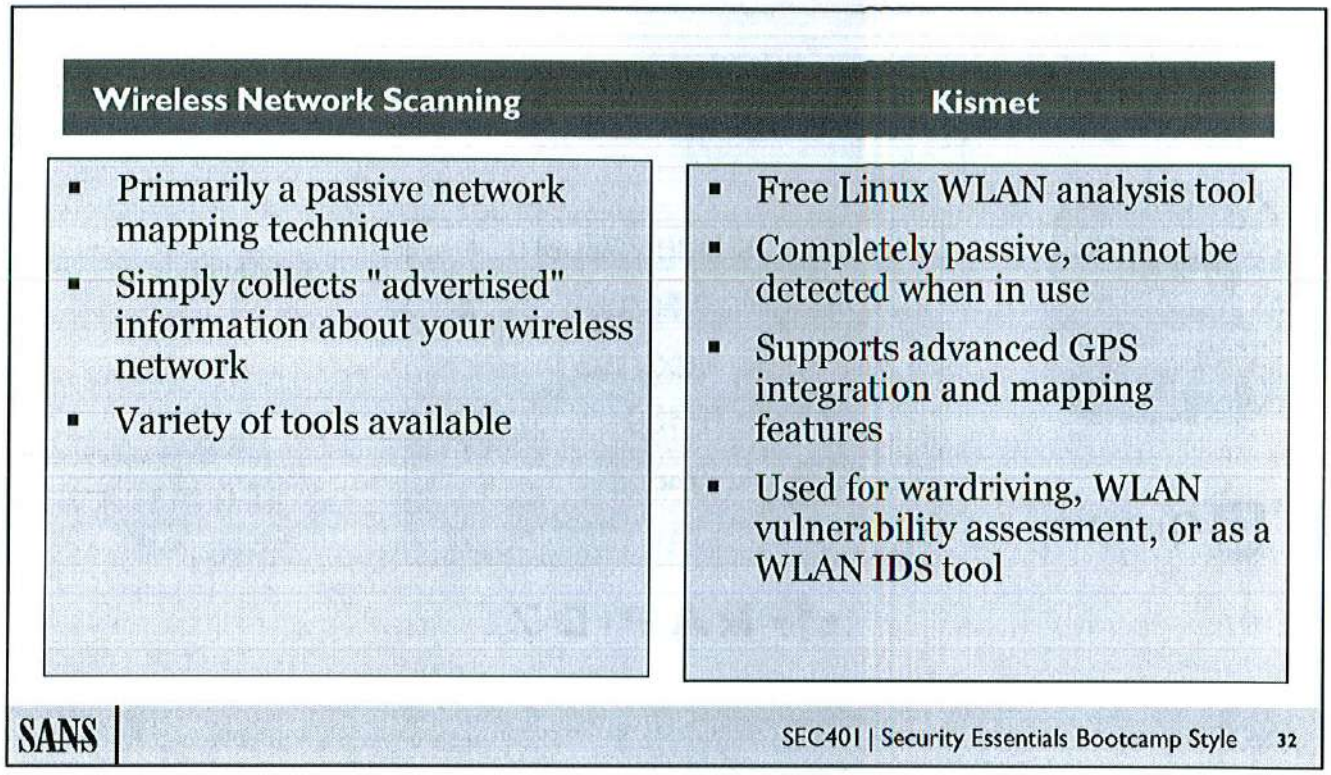
© 2017 Secure Anchor Consulting



This slide lists some of the best-known commercial vulnerability scanners on the market today. SANS doesn't endorse any one tool, so your mileage may vary with any of them. This slide is just meant to give you an overview of some of the various options, but is not meant to be a comprehensive list. Use this slide as a placeholder to research and find the vulnerability scanner that works best in your environment.

© 2017 Secure Anchor Consulting

Technet24



Wireless technology has become cheap and easy to deploy. This brings a new dimension to securing your network. Before wireless networks, you could protect your network by protecting your wiring closets and server rooms and physically protecting the facility. Wireless LANs and rogue access points can extend your network beyond the physical confines of your building.

Would-be attackers no longer need to defeat your perimeter defenses or compromise your remote access to access your internal network. An attacker can now drive into your parking lot with a laptop, a wireless NIC card, and a small antenna and gain access to your internal traffic. We don't have the luxury of thinking of our network perimeter as existing outside our firewall and in our DMZ—the network perimeter is any point where an attacker can gain access to the network, including wireless LANs. After an attacker has compromised the security of your wireless LAN, he has the same access as any other authorized user, despite the fact that he can be in any other location with physical proximity to your wireless network.

Kismet

Linux and BSD UNIX users have the option of using Kismet for mapping wireless networks. Kismet is designed to be a passive wireless sniffer, a wardriving tool, a wireless vulnerability assessment tool, and an intrusion detection tool. Kismet is made available under the GNU Public License, so it's free to download and use for private or commercial purposes. Kismet is available at http://www.kismetwireless.net.

Kismet is completely passive because it never transmits any frames. Instead, Kismet captures all the packets available on all monitored frequencies. What Kismet does with the data it receives is what makes it an amazing tool.

Using Kismet, we can determine an extraordinary amount of information about discovered wireless networks, organized into a format that is useful for ad-hoc analysis or detailed reporting. In addition to identifying located wireless networks, Kismet also identifies any clients that are on the network, cryptographically weak traffic, clear-text strings transmitted over the wireless network, factory-default access point configurations, and many more useful pieces of information. When equipped with a GPS, Kismet can produce detailed maps based off of vector maps (streets) or satellite photographs of an area, indicating the range and approximate center of wireless networks, as well as power interpolation diagrams indicating the exact shape and radiation of wireless signal from a central location.

© 2017 Secure Anchor Consulting

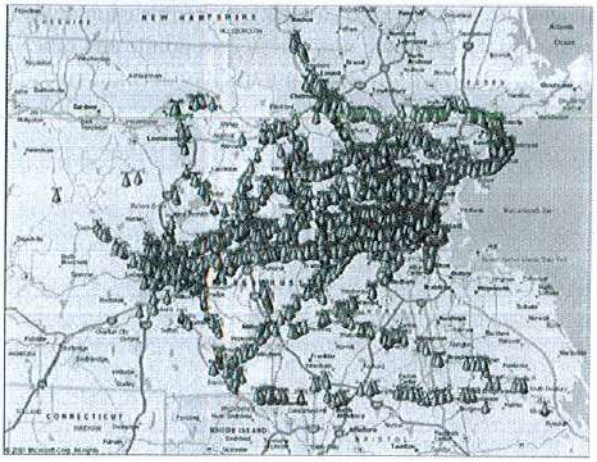
As an intrusion detection tool, Kismet can identify malicious activity on the wireless network, including many popular wireless network attacks, such as Denial of Service, man-in-the-middle attacks, and attacks against protocols. Kismet can be deployed in a client/server infrastructure, using the kismet_drone tool on a lightweight computer (small laptop, appliance or other embedded device) to monitor a wireless network for attacks. From a centralized location, the intrusion analyst can monitor all the activity on his wireless networks, including the presence of rogue access points and other events of interest. Kismet can also be integrated into Snort to analyze IP traffic using traditional Snort rules.

© 2017 Secure Anchor Consulting

Technet24

Mitigating Wireless Network Mapping

- Always employ strong encryption on wireless networks
- Follow best practices for securing wireless networks
  - Reduce signal strength
- Deploy monitoring mechanisms



SANS | SEC401 | Security Essentials Bootcamp Style 34

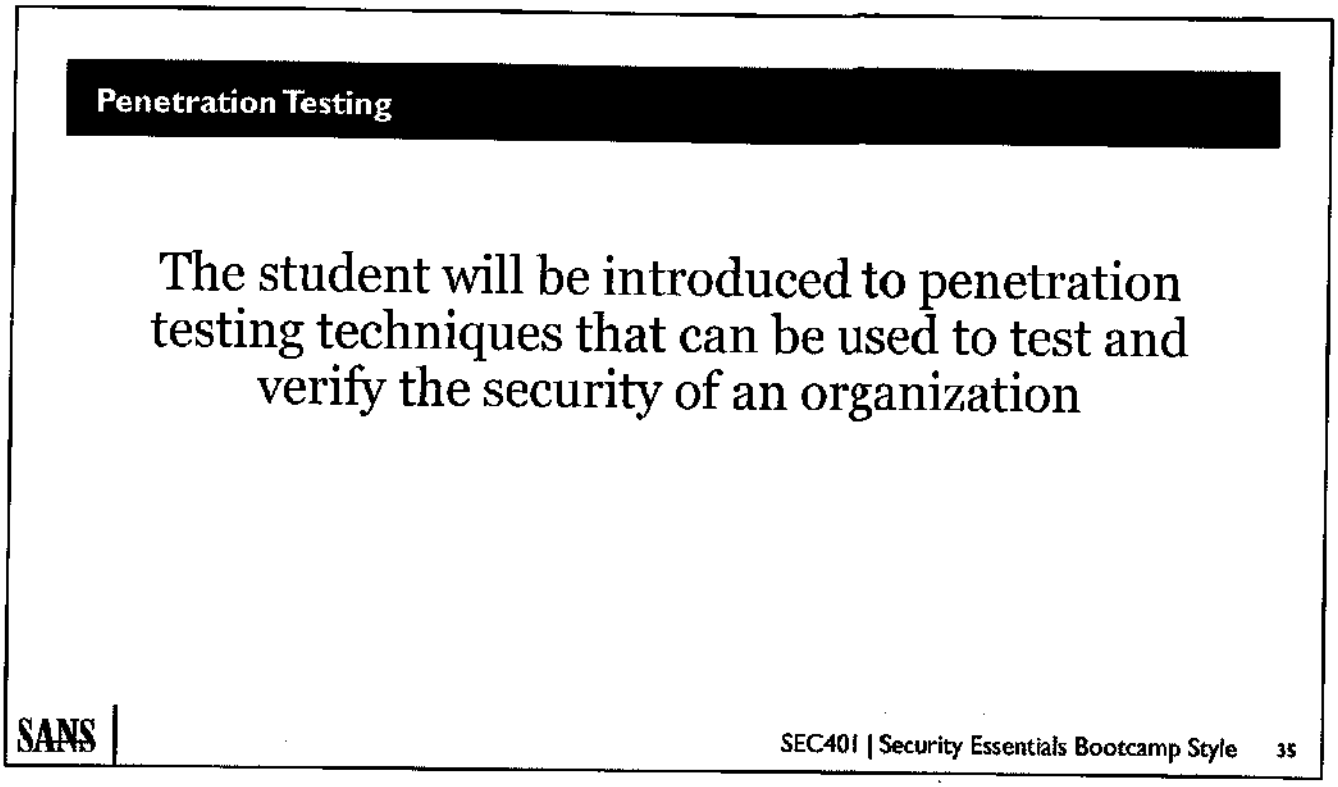
Because of the nature of wireless networking, there is little opportunity to avoid network mapping. If your wireless network is in a fairly populated area, you can expect your wireless network to be mapped frequently—sometimes by curious wardrivers, other times by people seeking free Internet access, and sometimes by people seeking to exploit weaknesses in your network. We can reduce the range from where an attacker can map a wireless network by reducing the signal strength of wireless equipment, using careful placement techniques to limit RF leakage outside a building, and by implementing RF barriers, such as metal screening inside exterior walls, but these measures are not foolproof and can be costly.

Instead, a strong authentication and encryption system is the best defense for a wireless network. Although an attacker can still locate your wireless network and possibly determine the type of authentication and encryption in use, we can rely on the strength of these protocols to limit an attacker's ability to gain additional access to the network.

Monitoring mechanisms are also important for protecting your wireless network. We often forget that monitoring the wireless network doesn't have to be done by monitoring the RF frequencies in use; instead, we can monitor logging information from access points and authentication servers (RADIUS, LDAP) to identify potential misuse. If we see the same username logged in multiple times from different source IP addresses, we can investigate the cause to determine whether there was a breach in network security.

Finally, scan your own network to see what an attacker can learn about your organization.

© 2017 Secure Anchor Consulting



This page intentionally left blank.

© 2017 Secure Anchor Consulting

Technet24

What Is Penetration Testing?

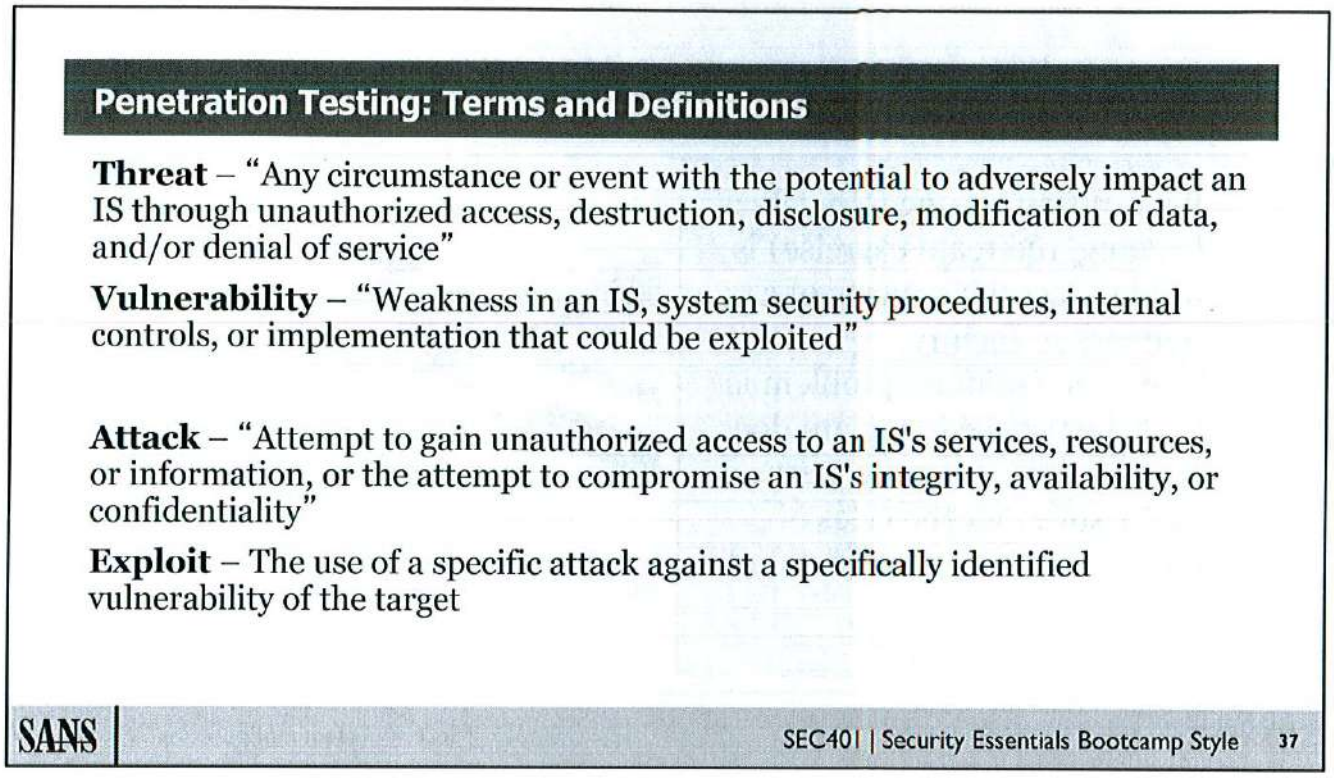
- It "is a method of evaluating the security posture of a computer system or network by simulating identified attacks by a malicious user, known as a hacker".
- The process involves an active analysis of the system for any potential vulnerabilities that may result from poor or improper system configuration, known and/or unknown hardware or software flaws, or operational weaknesses in process or technical countermeasures.
- Sometimes referred to as ethical hacking or red team exercises

In looking at a new area it is important to understand the scope of the topic and help define terms commonly used in this area of focus. First, a penetration test, as defined by Wikipedia, "is a method of evaluating the security of a computer system or network by simulating an attack by a malicious user, known as a hacker". The process involves an active analysis of the system for any potential vulnerabilities that may result from poor or improper system configuration, known and/or unknown hardware or software flaws, or operational weaknesses in process or technical countermeasures. This analysis is carried out from the position of a potential attacker and can involve active exploitation of security vulnerabilities. Any security issues that are found will be presented to the system owner, together with an assessment of their impact, and often with a proposal for mitigation or a technical solution. The intent of a penetration test is to determine the feasibility of an attack and the amount of business impact if a successful exploit is discovered.

References

1. Penetration Test - https://en.wikipedia.org/wiki/Penetration_test
2. SEC501 – for consistency across courses

© 2017 Secure Anchor Consulting



Let's take a moment to clarify and review some of the terminology we will be using in this section. Committee on National Security Systems (CNSS) 4009(1) – National Information Assurance Glossary and National Institute of Standards and Technology (NIST) IR 7298 Glossary of Key Information Security Terms, have several common terms and definitions that are key to understanding the use and scope of penetration testing, namely:

- Threat – As defined by CNSS 4009, a threat "is any circumstance or event with the potential to adversely impact an IS through unauthorized access, destruction, disclosure, modification of data, and/or denial of service."
  There are two primary types of threats: intentional and non-intentional. Usually, non-intentional threats are driven by natural occurrences, such as rain, thunder, and earthquakes. Intentional threats are usually man-made and are driven by an attack agent's malicious intent to exploit an information system.
- Vulnerability – Also as defined in CNSS 4009, a vulnerability "is a Weakness in an IS, system security procedures, internal controls, or implementation that could be exploited."

When working with anyone on security, it is always important to get a clear explanation of terms to make sure they have the same meaning. Many terms in cyber security mean different things to different people. Therefore if you do not have a consistent glossary it will be hard to communicate.

Reference

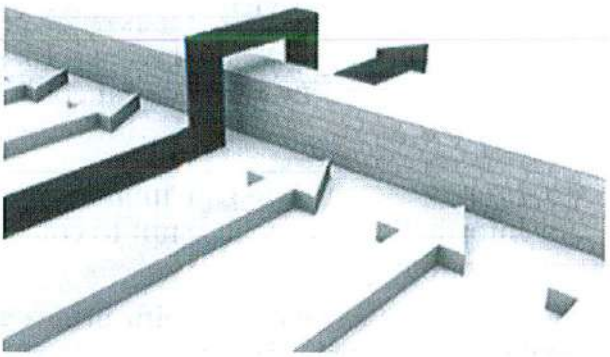
1. SEC501 – for consistency across courses

© 2017 Secure Anchor Consulting

Technet24

Managing Penetration Testing

- Penetration testing (ethical hacking, red team exercise) is used to test the security of a network or facility
- The most common problem is that the pen-testing team does not focus on the correct areas; make sure that the rules of engagement are clearly stated



Penetration testing is used to test the security of a network or security itself. The testing can take many forms: the perimeter security of any building, server, or network must be tested regularly, otherwise, attackers can do it for you.

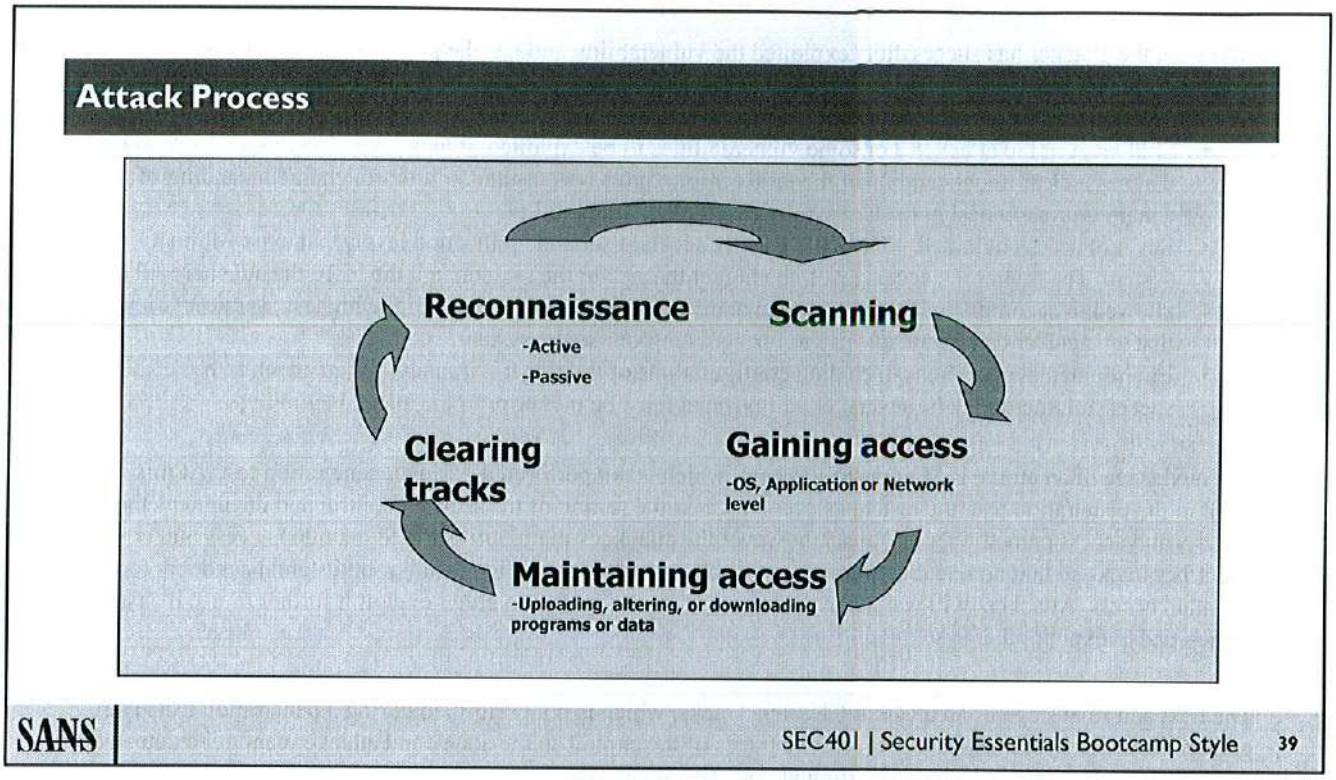
A penetration test sometimes is completed at the conclusion of a vulnerability scan and is used to determine the validity of any identified vulnerabilities. By actually attempting to exploit vulnerabilities found in the scan, a penetration test helps eliminate false positives.

Penetration tests are sometimes run in lieu of a vulnerability assessment and are conducted entirely from outside the network, from the perspective of a true outside attacker. The tests can evaluate the effectiveness of your security perimeter, including routers, firewalls, servers, and any other perimeter security devices.

Reference

1. Penetration Testing Summit - https://www.sans.org/event/pentest-berlin-2013

© 2017 Secure Anchor Consulting



It is important to understand the anatomy of an attack to see how penetration testing relates to a real attack - let's explore each step in the process.

The first step and preparatory stage of an attack is Reconnaissance. This is the same as the Information Gathering phase of the penetration testing methodology. During reconnaissance, the attacker gathers information about the target using active and passive mechanisms. This can include web searches, determining the IP address range, social engineering, and dumpster diving, among other techniques. This stage sometimes includes network IP scanning, which differs from the penetration testing methodology that puts all types of scanning in the Scanning phase. The goal of this phase is to obtain as much as possible from open sources and other methods to understand the target. The attack may use competitive intelligence to learn about the target. There are two primary categories of reconnaissance techniques: active or passive. Passive reconnaissance is when the attacker does not interact with the system directly. Active reconnaissance is used when the attacker believes that there is a low possibility of being detected by the system.

The Scanning stage is a pre-attack activity similar to the penetration testing Scanning phase. This activity is seen as a logical extension of active reconnaissance. This includes more advanced probing of the target for vulnerabilities that can be exploited. It can include network mapping, port mapping, vulnerability scanning, OS fingerprinting, war dialing, wireless scanning, and more. There is often a lot of overlap between the Reconnaissance and Scanning phases during an attack, and even during penetration testing. The attacker is attempting to assess the system to identify system vulnerabilities that could be easily exploited. As stated previously, an attack cannot be successful unless a vulnerability exists that can be exploited.

The next stage is Gaining Access, which parallels the Enumeration and Exploiting phase of penetration testing. Here the attacker exploits a vulnerability to gain access to the system. Techniques include buffer overflows, password cracking, Smurfing, and spoofing. This is usually considered the most important phase of the attack. At

© 2017 Secure Anchor Consulting

Technet24

this point, the attacker has successfully exploited the vulnerability and has the potential to cause damage to the system. As stated previously, there are several factors contributing to the success of the attacker in gaining access to the system. These factors include:

- Skill level of the attacker. For some vulnerabilities to be exploited, it may take a very skilled attacker that has a clear understanding of the target information systems and/or a thorough understanding of computer system design to successfully exploit the vulnerability.
- Level of access obtained – There are usually several user and application levels that exist within a system. The higher the access level, the higher the risk to the system, and the more the attacker will be allowed to accomplish. Root or a system administrator account is deemed the highest user access level that an attacker can obtain and use to further compromise the system.
- Environmental conditions, including configurations of the target – Usually, for an attack to be successful, there may be several conditions that must be met prior to the attack beginning.

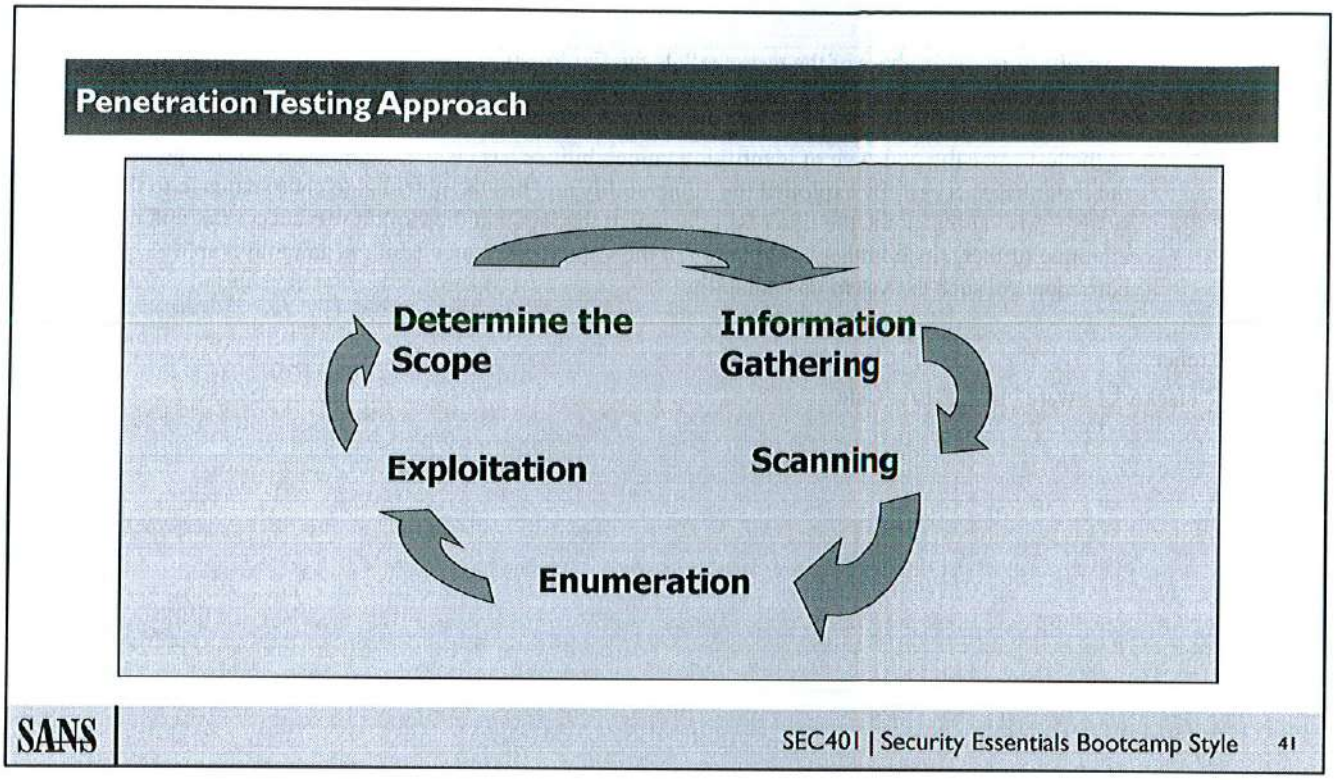
The next stage of an attack is Maintaining Access, which is not performed during a penetration test. At this point in the penetration test, the tester has accomplished the testing of the security control and documents the process and access gained. In a real attack however the attacker's main purpose is to maintain access and cover his or her tracks so that no one discovers the intrusion. Attackers will often install rootkits and backdoors to maintain access. Attackers will want to maintain access to the system to either exploit it further or use it as a launch pad to exploit other systems.

The next and final stage of an attack is Clearing Tracks, which is not performed during a penetration test. The goal of the attacker in this phase is to remove traces of the exploit so the attack and attacker cannot be detected and identified for penalization under criminal law. The attacker will remove or attempt to remove log entries and replace system binaries to cover their tracks and remain undetected on the system.

Reference

1. "Hackers Beware" by Dr. Eric Cole

© 2017 Secure Anchor Consulting



The following slides will present detailed descriptions of each step in the penetration testing approach, which is very similar to the attack steps described in detail in the previous slide. There exist many similarities between the attack process and penetration testing. At each step in the process, the tester takes detailed notes of which and when specific actions were taken as part of the test. The goal is to document and report vulnerabilities of the system in a systematic approach, keeping in mind the pre-approved constraints of the test. The steps involved with Penetration Testing include:

- **Determine the Scope** – Critical to any penetration test is to determine what the scope of the test will be. This includes identifying the target system, the types of testing authorized, tests that are not authorized, who the point of contact is for the test, acceptable testing time frame, and to clarify and define any other ambiguous terms of the test.
- **Information Gathering** – After receiving authorization to test and understanding the constraints of the test, this step is used to collect specific information about the target. This step is similar to the Reconnaissance step in the attack process. At this point, based on if you are performing black box, white box, or grey box testing, the penetration tester will seek to obtain additional information about the system. Each type of test will be described, in detail, in future slides.
- **Scanning** - The Scanning stage is a pre-test activity similar to the attack process Scanning phase. This includes more advanced probing of the target for vulnerabilities that can be exploited. It can include network mapping, port mapping, vulnerability scanning, OS fingerprinting, war dialing, wireless scanning, and more. There is often a lot of overlap between the Information Gathering and Scanning phases during an attack, and even during penetration testing. The tester is attempting to assess the system to identify system vulnerabilities that could be easily exploited. As stated previously, a test cannot be successful unless a vulnerability exists that can be exploited.
- **Enumeration** – This stage is a further extension of the scanning process to identify exploitable vulnerabilities of the system. The Enumeration phase attempts to list file permissions, user accounts, and open and idle ports, and other system items used for entry into the system.

© 2017 Secure Anchor Consulting

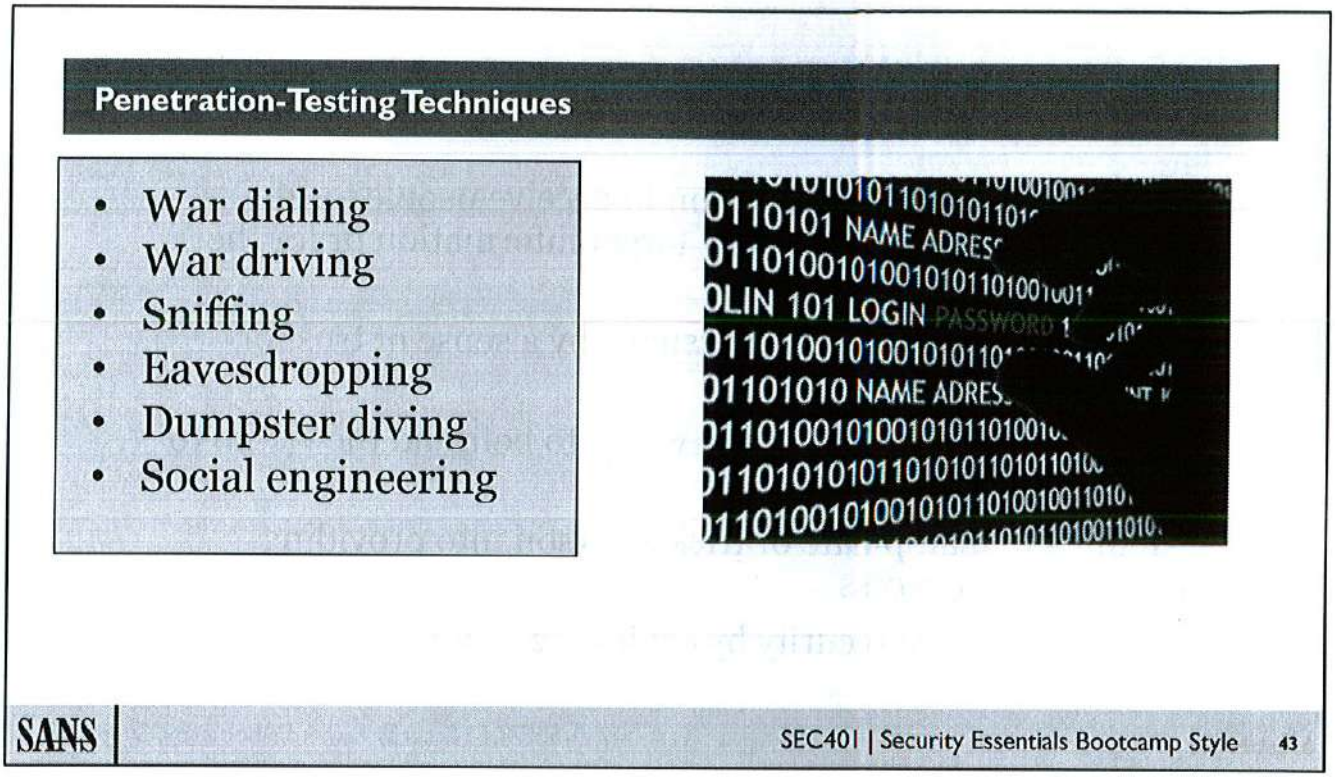
Technet24

- **Exploitation** – This phase of the test parallels the Exploitation phase of the attack process. Here the tester exploits a vulnerability to gain access to the system. Techniques include buffer overflows, password cracking, and spoofing. This is usually considered the most important phase of the test since the objective of identifying vulnerabilities has been accomplished. At this point, the tester has successfully exploited the vulnerability and has the potential to cause damage to the system. Also at this point, the tester will document the fact that the exploit was successful and not continue further, since further exploitation of the vulnerability may result in an unfavorable consequence, such as system unavailability.

Reference

1. "Hackers Beware" by Dr. Eric Cole

© 2017 Secure Anchor Consulting



Some common penetration testing techniques used by White Hats and Black Hats alike are

- War dialing
- Wardriving
- Sniffing
- Eavesdropping
- Dumpster diving
- Social engineering

Eavesdropping is one of the oldest tools in the security professional's handbook. It is simply listening to private conversations that might reveal information that can provide access to a facility or network. Eavesdropping can be done with a well-placed microphone or tape recorder. Or special monitoring equipment can be used to intercept cell-phone calls or other electronic communications.

Dumpster diving is wading through an organization's trash until you find enough information to give you access. Not a pleasant job, but it pays off once you find that sticky note with a valid password written on it.

Social engineering is the practice of using people rather than technology to obtain sensitive information or get access. Through dumpster diving, eavesdropping, or other means, we can find information and use it in conversation to convince someone to release information to us.

Think of the word "social" as representing the way someone might react to a request in polite society. An example is the "box theory." If you walk up to a facility holding what appears to be a heavy box, someone generally holds the door open for you, whether or not you are an employee with a valid identification badge!

Reference

1. What is a Pen Test and Why Would I Need One for My Organization? - http://www.forbes.com/sites/ericbasu/2013/10/13/what-is-a-penetration-test-and-why-would-i-need-one-for-my-company/#584fc29742da

© 2017 Secure Anchor Consulting

Technet24

Social Engineering

- Use of influence and persuasion to deceive people for the purpose of obtaining sensitive target information or for the victim to perform an action
- Enables access to the target, usually by a sense of trust or authority of the attacker
- Preys on human behavior and works to befriend the victim to obtain the information
- Attempts to manipulate or trick a person into providing information or access
- Bypasses network security by exploiting human vulnerabilities

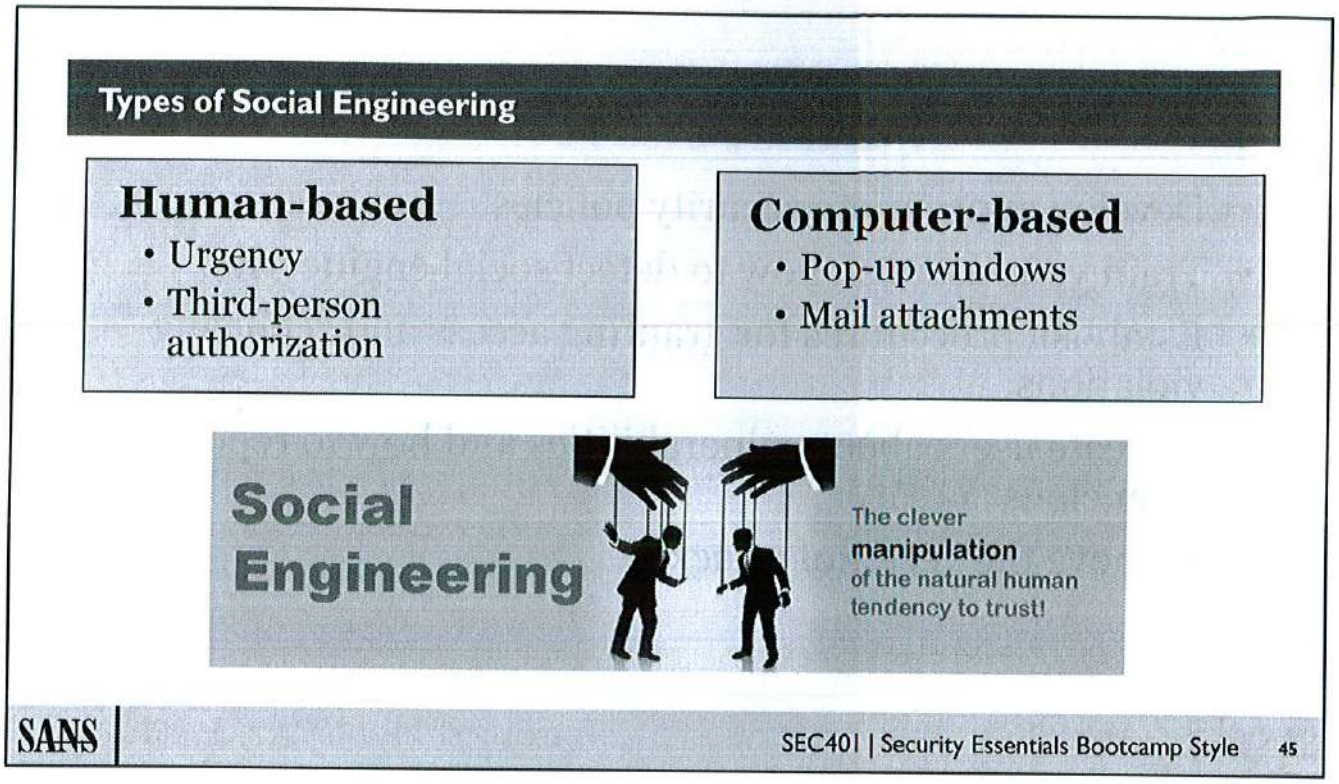
Social Engineering is defined as the use of influence and persuasion to deceive people for the purpose of obtaining sensitive target information or for the victim to perform an action enabling access to the target, with or without the use of technology, usually by a sense of trust or authority of the attacker. The penetration testing technique preys on human behavior and works to befriend the victim to obtain the information. In some cases, social engineering is an effective way to obtain access and information on the target system without using extensive resources to crack password files and other methods.

Often, people are the weakest links in an organization's security. All the technology in the world cannot protect your network from a user who willingly gives out her password or innocently installs malicious software.

Social engineering is the term used to describe an attempt to manipulate or trick a person into providing valuable information or access to that information. It is the process of attacking a network or system by exploiting the people who interact with that system.

Social engineering often preys on human nature, such as the desire to be helpful, the fear of getting in trouble, or the tendency to trust the people and computers with whom and with which we interact.

© 2017 Secure Anchor Consulting



Most social engineering is human-based, involving one person trying to get valuable information from another person. The best-known techniques are urgency, impersonation, and third-person authorization. Here is a classic example: A man calls the help desk: "Hello, this is Bob Smith, Vice President of Big Corporation. I'm traveling, and I forgot my password. Can you reset it so I can retrieve an important e-mail message for a meeting in 15 minutes?" Would your help desk question this request? Most people would give out the information without thinking, either because they want to be helpful or they are afraid of refusing the vice president's request, especially because he has an urgent meeting in 15 minutes.

Social engineering also can be computer-based. Consider this example: A user is browsing the web when she sees a pop-up window displaying her Internet connection has timed out, and she needs to re-enter her username and password to re-authenticate. Would the average user question this occurrence? It is a common means to steal password information. Also, most of the recent mail worms come with subject lines and body text designed to convince the reader to open the attachment, even though many users now know the danger of doing so. Exploiting human curiosity, gullibility, or greed, even automatically via the use of mass-mail worms, is pure social engineering at work. Phishing is a perfect example of how this can be used to cause harm.

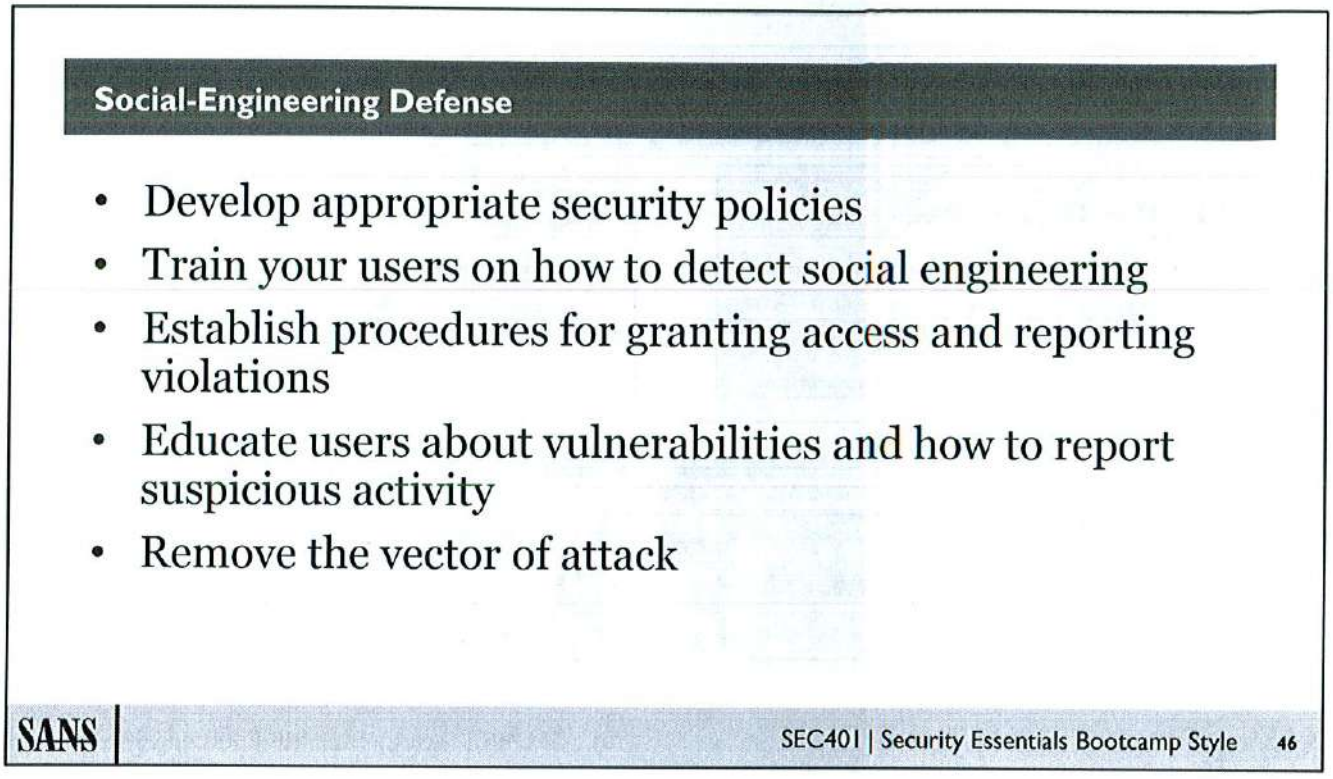
These examples show how human nature can make it easy for an attacker to walk right into your network. Why hack through someone's security system when you can get a user to open the door for you?

Reference

1. Social engineering - https://www.cybertec-security.com/what-we-do/social-engineering-attack/

© 2017 Secure Anchor Consulting

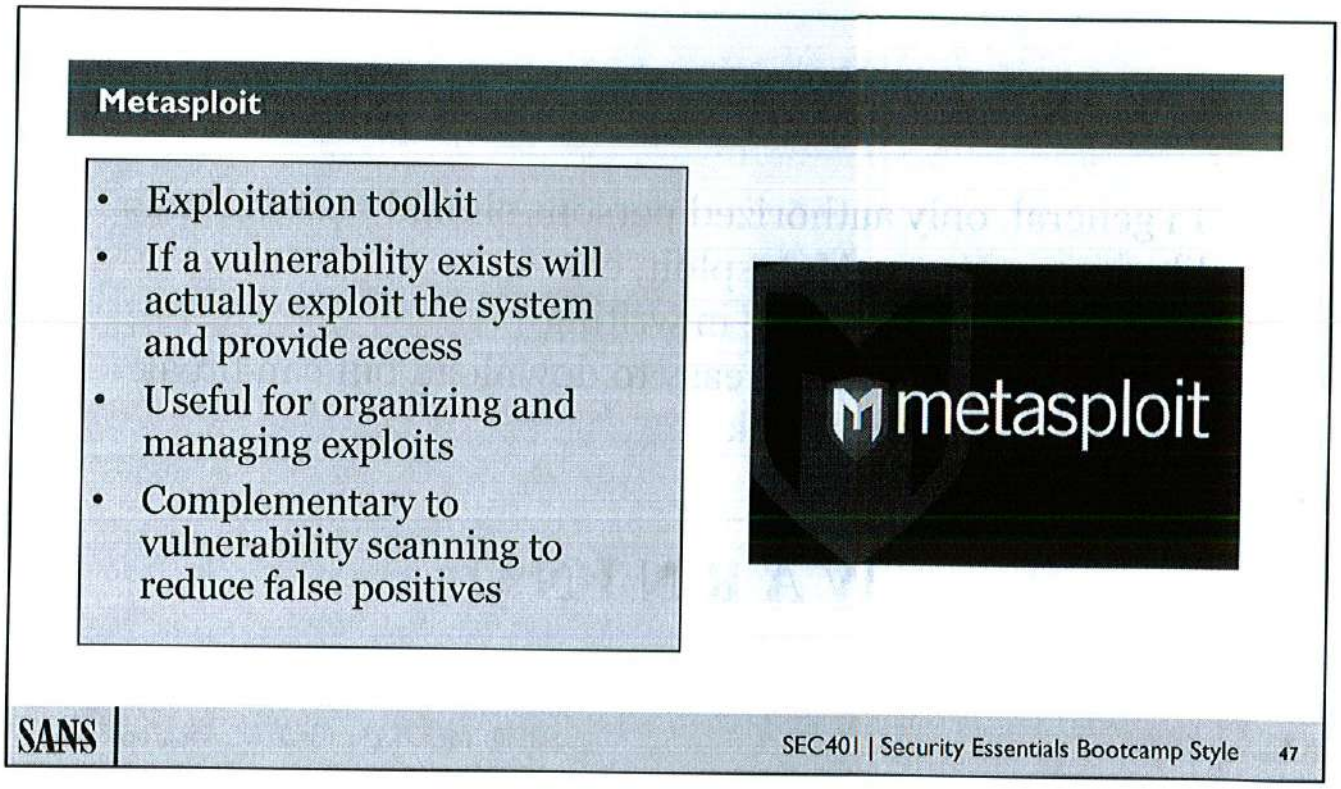
Technet24



Social engineering is one of the hardest classes of attack to defend against. The weakness is a human one: We want to help people. Technology, such as host perimeter defense products, can provide some protection (for example, antivirus software to guard against users who inadvertently run viruses or Trojan software). However, your best defense is to establish clear security policies and enforce them.

Security policies should establish such things as the types of access allowed, the people authorized to grant such access, and the circumstances under which exceptions might be granted. In addition to policy, you should define procedures for activating and deactivating accounts, changing or resetting passwords, and granting additional rights or privileges. Finally, educate your users about these types of threats. In most cases, users do not maliciously create security problems—they generally do so out of ignorance. If users are aware of the threats, they can properly guard against them.

© 2017 Secure Anchor Consulting



Vulnerability scanners are useful for finding vulnerabilities, but because they are often looking for conditions of a vulnerability, the can have false positives. The next evolution in tools is exploitation tools. These are tools that if a vulnerability exists will actually exploit the system and provide the person running the tool, access to the system. One very popular tool in this space is Metasploit. Metasploit will actually exploit a vulnerability, if it is present, and provide access to the system through various methods that can be specified in the tool.

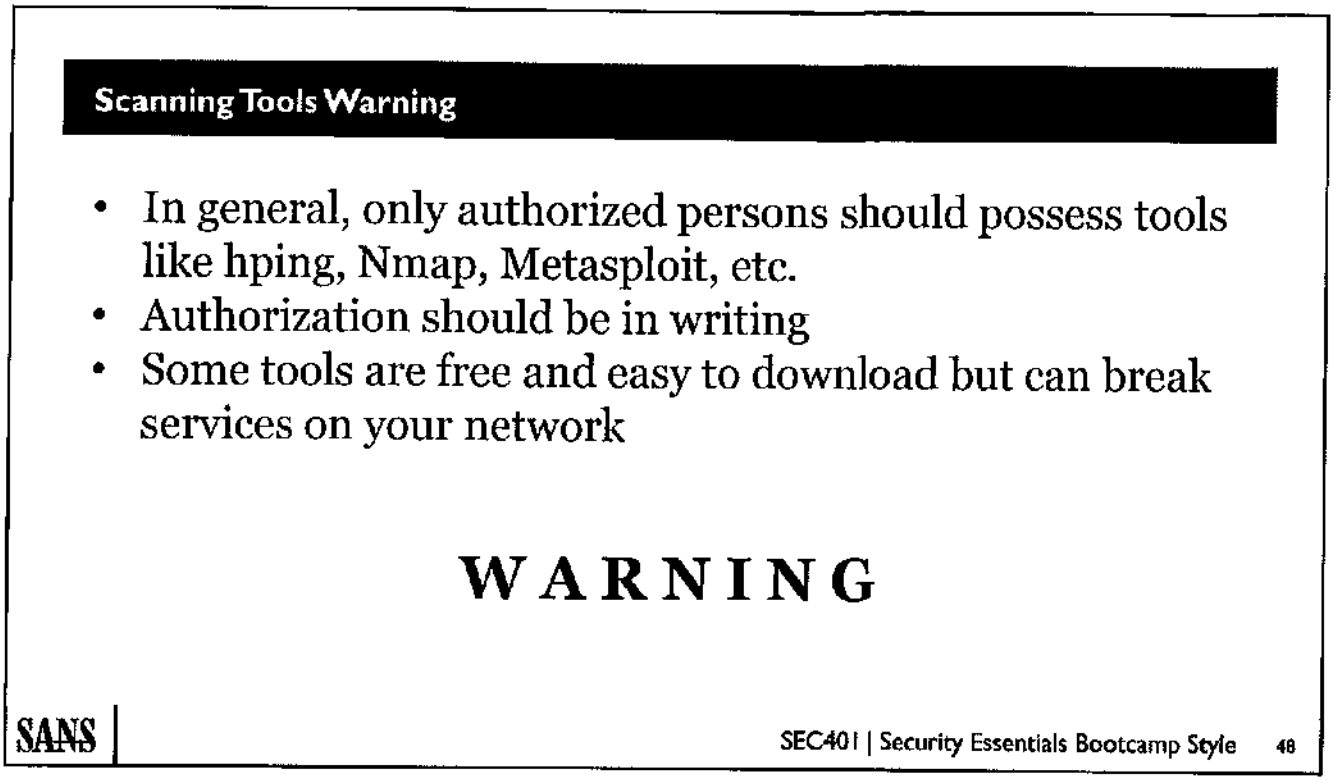
Metasploit is not only a useful tool for penetration testing, but it is also a helpful way to verify whether a vulnerability really exists on the system. If a vulnerability scanner finds an exposure it is easy for an administrator to say it is a false positive. However, if Metasploit actually exploits it, it is pretty hard to say it is a false positive!

Reference

1. How to Search Exploits with Metasploit - https://www.blackmoreops.com/2015/11/03/how-to-search-exploits-in-metasploit/

© 2017 Secure Anchor Consulting

Technet24

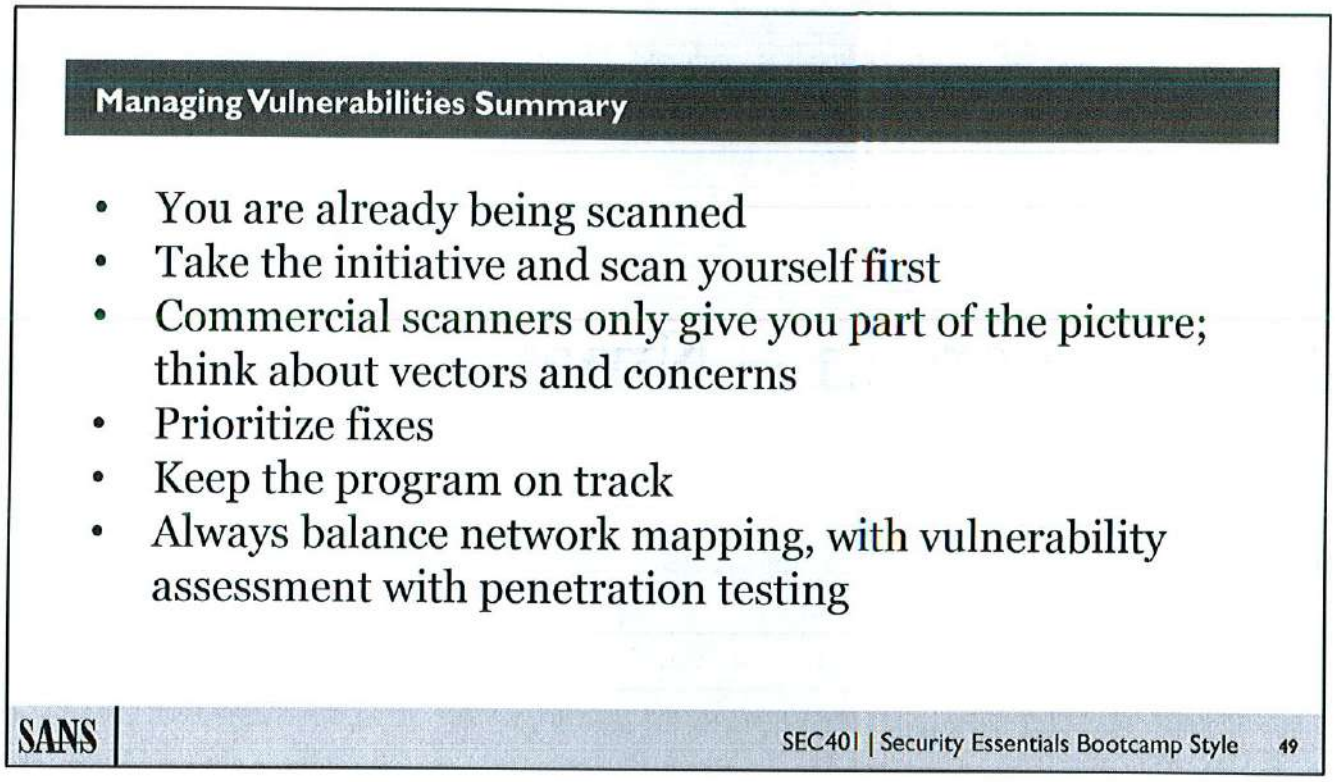


Scanning tools can be a valuable asset to system administrators for monitoring and assessing networks, but require a level of education about how the tools work to properly understand the impact to the systems and networks that are being scanned. When used by untrained personnel, these tools can cause damage to the systems being assessed, ranging from degraded to complete loss of service on the target networks and systems. It can also be illegal in some cases.

Authorization to use these applications on your network should be documented in writing and only granted to individuals who have demonstrated an understanding of how the tools work and the ramifications of improper use of the tools.

Intrusion detection systems (IDS) can be used to identify hosts that are running these potentially dangerous applications. They can be configured to only report use from unauthorized sources (for example, from a user who is not on the written authorization list). Organizations should consider logging all access to systems through these tools, even authorized scans, to maintain an audit trail of analysis to track how and when systems were assessed. This information is helpful when troubleshooting a problem that may have been caused as a result of a recent scan.

© 2017 Secure Anchor Consulting



Firewalls, no matter how well configured, are never perfect. They usually can be subverted by the actions of internal users. Attackers take advantage of firewall weaknesses using peer-to-peer file-sharing tools, wireless networking, and modem connections.

Attackers also have developed methods to take advantage of vulnerabilities within your network. They have developed Trojans, which your users unwittingly run, leaking important information onto the Internet and allowing attackers to gain access. Attackers also have developed scanners to look for open shares and services on your systems. The scanners often use packet crafting and source-address spoofing to circumvent firewalls. To avoid the effort involved with running scans of large chunks of the Internet, attackers also have developed worms that automatically scan for vulnerable hosts and propagate exponentially. The only sure way to combat these threat vectors is to make sure your operating system and application software are always patched and up-to-date.

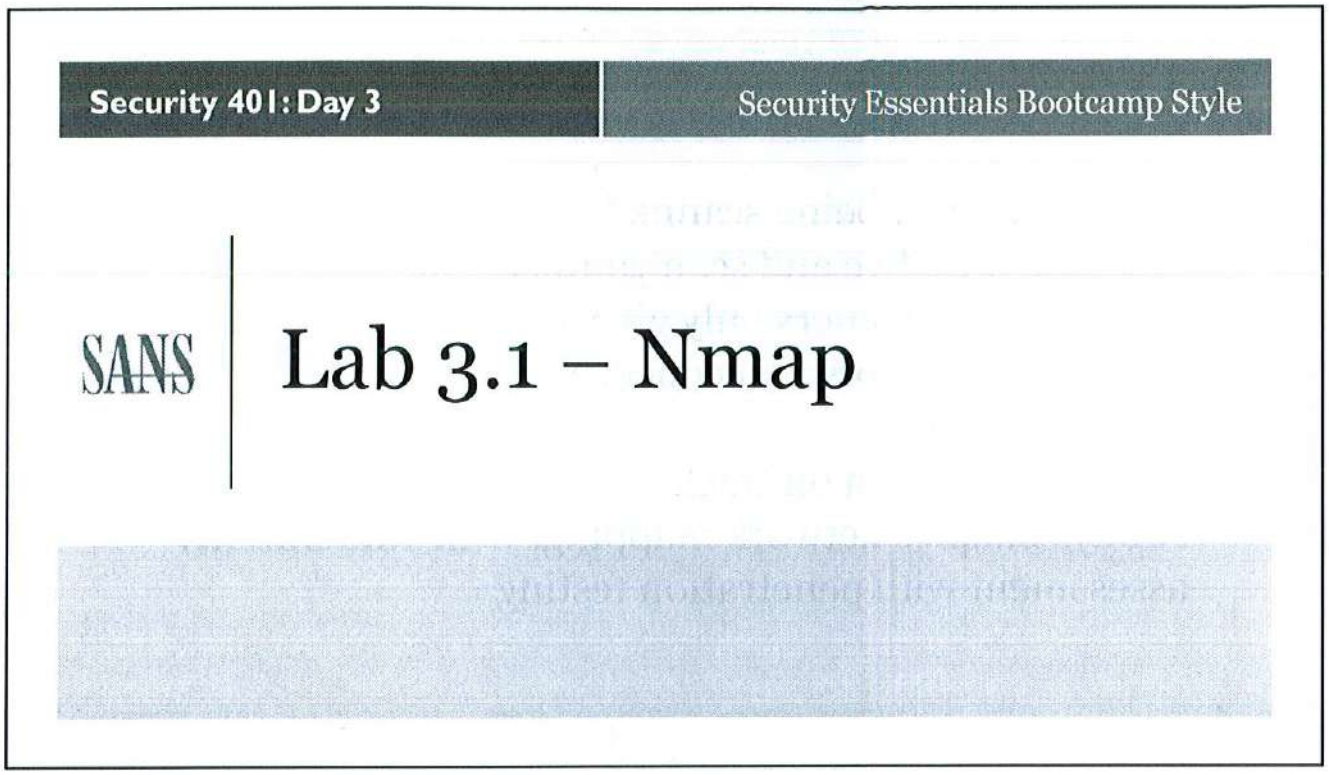
Because it is clear that attackers are continually performing reconnaissance scans of all Internet-connected systems, it makes sense to beat them to the punch by scanning your own systems and quickly correcting any vulnerabilities you find. Freely available tools, such as Nmap, can be useful in auditing your networks, but first be sure to get written permission from the appropriate official in your organization. In addition, because scans have been known to crash machines, it is also a good idea to alert system administrators before you run a scan.

It also makes sense to do your own wireless scanning in your facilities. Even if your company has not deployed its own wireless network, employees can cheaply and easily deploy their own rogue wireless access points. Finding and eliminating improperly configured wireless access points reduces the likelihood of an external attack against your internal network via your wireless network.

Penetration testing is the process of attempting to exploit actual vulnerabilities. It can be used in place of vulnerability scanning, but it is more effective when used to verify the results of a previous vulnerability scan. Most vulnerability scanners generate false positives, and a subsequent penetration test can verify whether those false positives can be ignored safely.

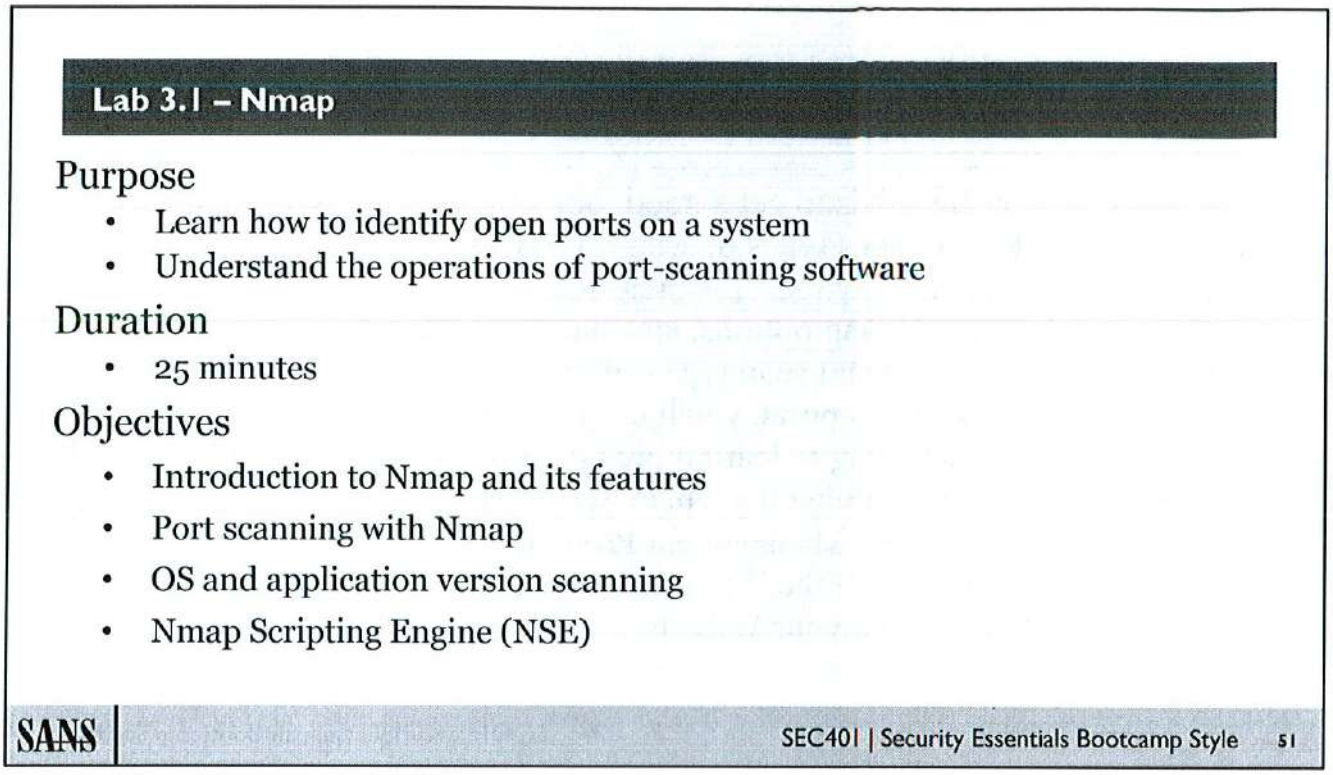
© 2017 Secure Anchor Consulting

Technet24



One of the tools covered in the "Vulnerability Scanning" module is Nmap. This powerful tool has the capability to perform many different types of network scans, including ping sweeping, TCP/UDP port scanning, OS fingerprinting, application version scanning, and even script execution with the Nmap Scripting Engine (NSE). Nmap was originally written by Gordon Lyon, who goes by the name Fyodor. It was originally released back in 1997 and has grown to the most popular open-source port scanner used today. The project is actively maintained and available at https://nmap.org/. A large number of freely available port scanners are available on the Internet; however, few would argue that Nmap is the best one and easily rivals any commercially available product.

© 2017 Secure Anchor Consulting



Purpose

- Learn how to identify open ports on a system
- Understand the operations of port-scanning software

Duration

- 25 minutes
- The estimated duration of this lab is based on the average amount of time required to make it through to the end. The duration estimate of this lab can decrease or increase depending on various factors, such as the booting of virtual machines, the speed and amount of RAM on your computer, and the time you take to read through and perform each step. All labs are repeatable both inside and outside of the classroom, and it is strongly recommended that you take the time to repeat the labs both for further learning and practice toward the GIAC Security Essentials Certification (GSEC).

Objectives

- Introduction to Nmap and its features
- Port scanning with Nmap
- OS and application version scanning
- Nmap Scripting Engine (NSE)

© 2017 Secure Anchor Consulting

Technet24

Lab 3.1 - Overview

Your objective for this lab is to get a good look at Nmap's power and ease of use along with its various types of scans. You'll first take a close look at some of the most commonly used features, such as the ability to save your scan results, changing timing options, and starting different types of scans. Next, you'll perform different scan types against your target Windows 10 VM. After identifying open ports, you'll use OS fingerprinting and application version scanning to learn more about the services running on the target. Finally, you'll utilize the Nmap Scripting Engine (NSE) to perform a Simple Network Management Protocol (SNMP) scan and enumerate information from the Windows system. You will use your Kali Linux VM for this lab with your Windows VM as the target.

Your objective for this lab is to get a good look at Nmap's power and ease of use along with its various types of scans. You'll first take a close look at some of the most commonly used features, such as the ability to save your scan results, changing timing options, and starting different types of scans. Next, you'll perform different scan types against your target Windows 10 VM. After identifying open ports, you'll use OS fingerprinting and application version scanning to learn more about the services running on the target. Finally, you'll utilize the Nmap Scripting Engine (NSE) to perform a Simple Network Management Protocol (SNMP) scan and enumerate information from the Windows system. You will use your Kali Linux VM for this lab with your Windows VM as the target.

© 2017 Secure Anchor Consulting

SANS

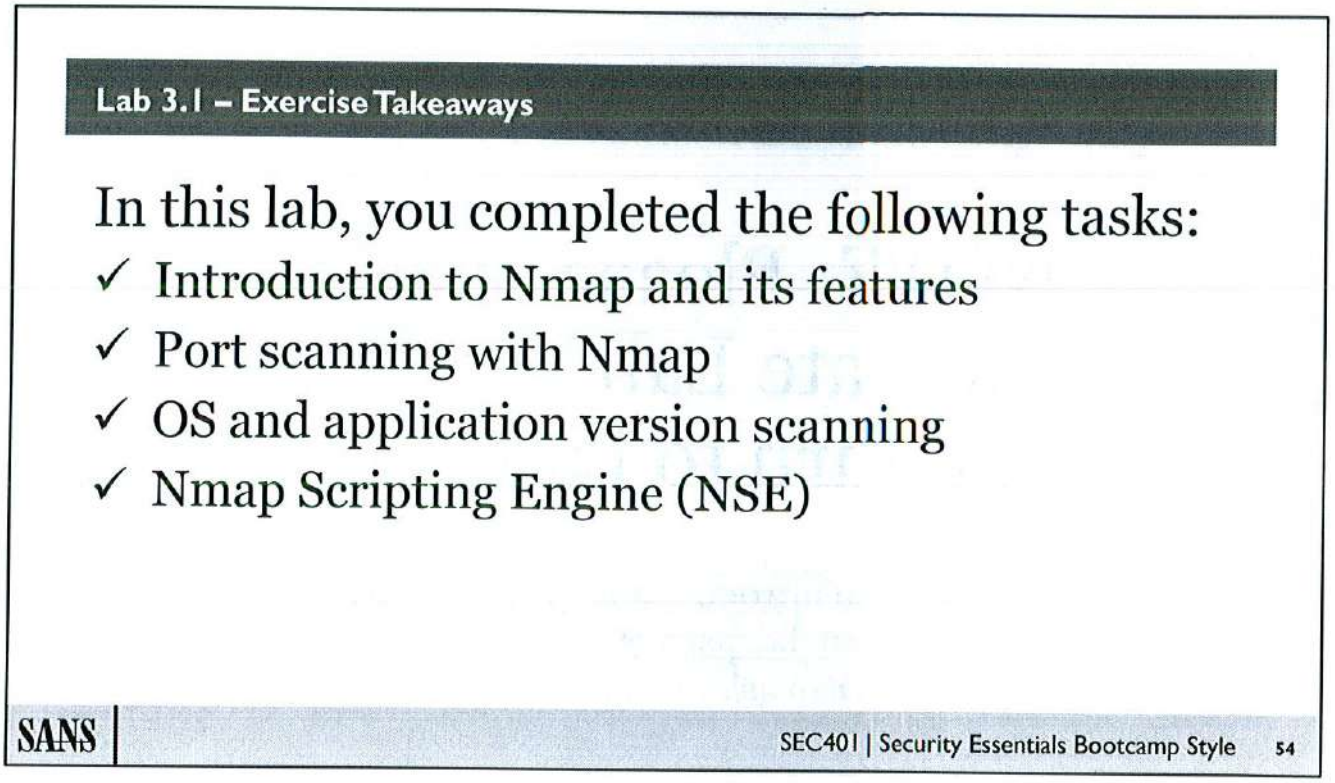
NOTE: Please open the separate Lab Workbook and turn to Lab 3.1

The instructor is going to introduce and go over the labs. Once the instructor is done, you will be instructed to work on the lab. If you have any questions, you can ask the instructor.

This page intentionally left blank.

© 2017 Secure Anchor Consulting

Technet24



In this lab, you completed the following tasks:

- ✓ Introduction to Nmap and its features
- ✓ Port scanning with Nmap
- ✓ OS and application version scanning
- ✓ Nmap Scripting Engine (NSE)

Nmap is by far the most widely used port scanner available today. As previously stated, it started off as a simple port scanner but has grown to much more, including OS and application identification and vulnerability scanning and exploitation with the Nmap Scripting Engine (NSE). As with all the tools used in this class, we must always have written permission prior to running them on a production network or system. Improperly used, tools such as Nmap have the ability to inadvertently set off IDS alerts, as well as negatively impact systems and networks. Always use caution.

© 2017 Secure Anchor Consulting

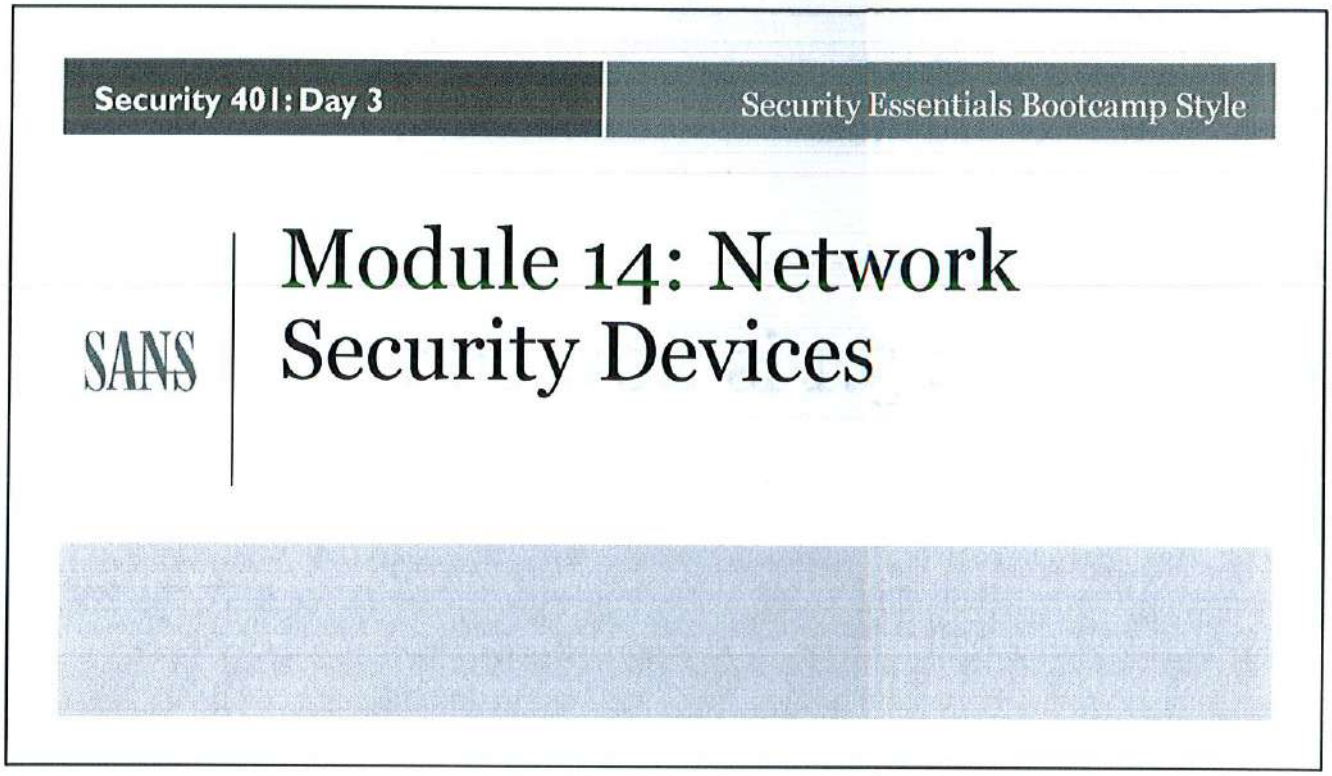
SANS

Lab 3.1 is now complete

This page intentionally left blank.

© 2017 Secure Anchor Consulting

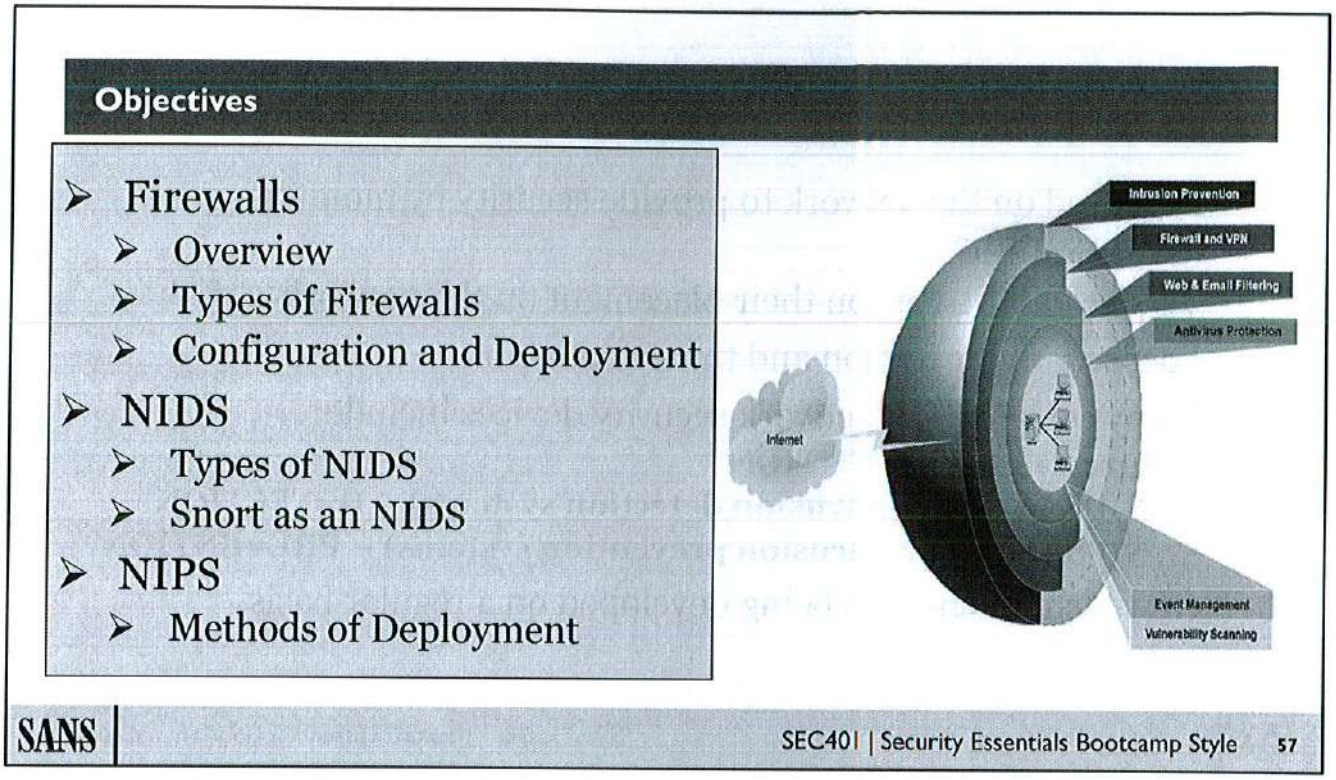
Technet24



Module 14: Network Security Devices

As more and more critical information is stored on computers, it is critical to have a proper network architecture. This encompasses network security devices placed at key areas in the network. Network security devices are critical to monitoring traffic and performing both prevention and detection on harmful or potential harmful traffic. By properly designing and configuring network security devices, can help in reducing the overall damage of an attack or even stopping an attack.

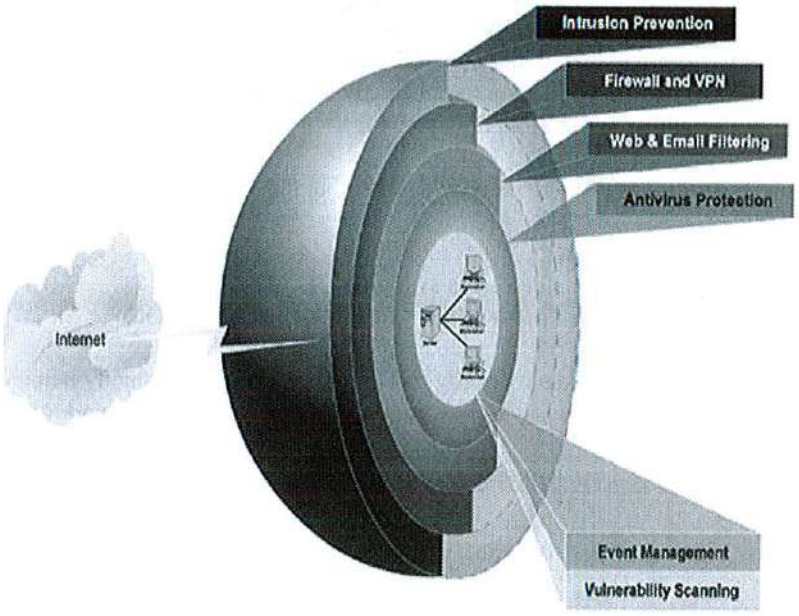
© 2017 Secure Anchor Consulting



This module will look at the 3 main categories of network security devices: firewalls, NIDS and NIPS. Together they provide a complement of prevention and detection capabilities. If implemented correctly, they will not only stop the adversary from causing damage but also allow for visibility to perform timely detection. The overall goal with any security including network security is to control and minimize the overall damage.

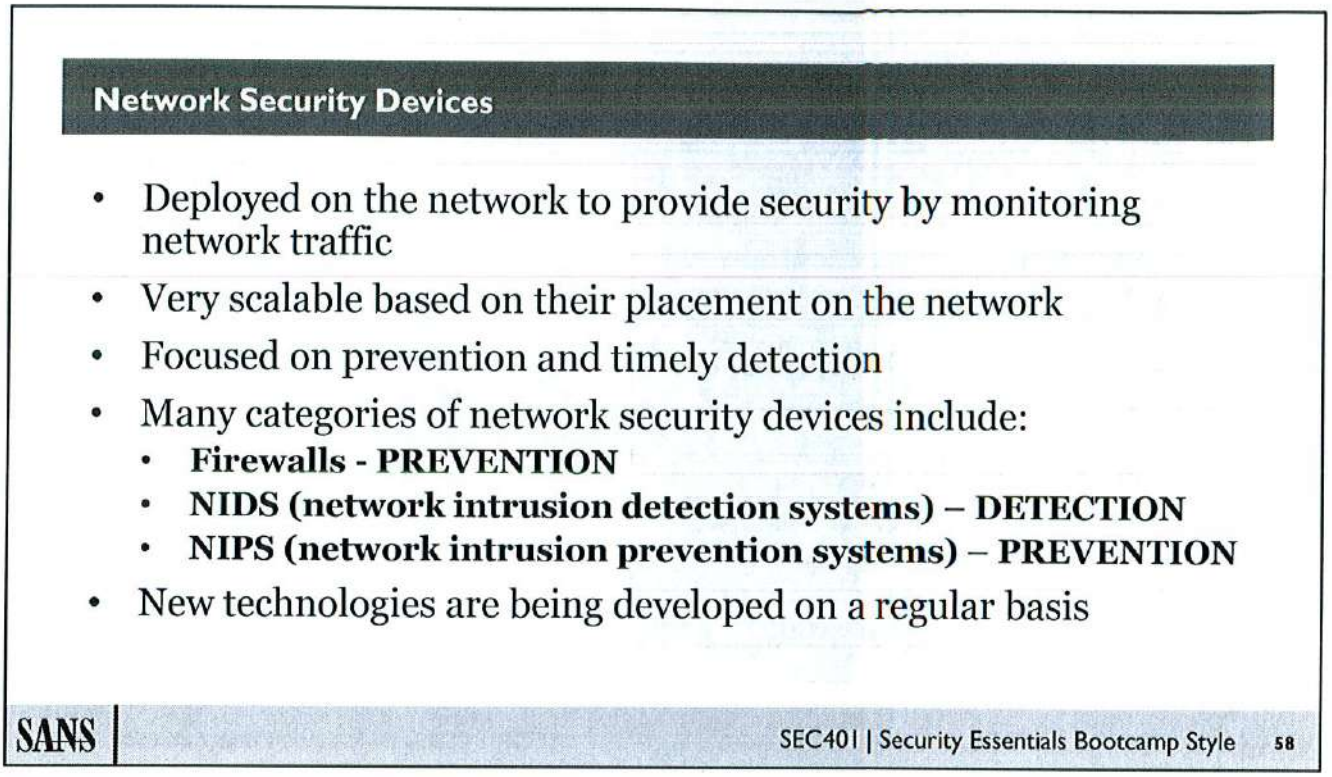
References

1. Intrusion Defense: A Layered Plan - http://www.brighthub.com/computing/smb-security/articles/2759.aspx
2. Security Layers - http://1.bp.blogspot.com/_jLYMd3eAswM/TK6Ez0_p8WI/AAAAAAAAAAk/gNZxLVFpcsY/s1600/security-layers.jpg



© 2017 Secure Anchor Consulting

Technet24



In most cases, an adversary is going to target an organization and break in over a network such as the Internet. The Internet allows for organizations to connect with someone anywhere in the world and have an international presence. However, remember that anything that can be used for good can also be used for evil. This means that the Internet also allows an adversary to connect to any organization in the world and cause harm. Therefore, it is important to ensure there are proper network security devices in place to protect critical information and make it difficult for the adversary to cause damage.

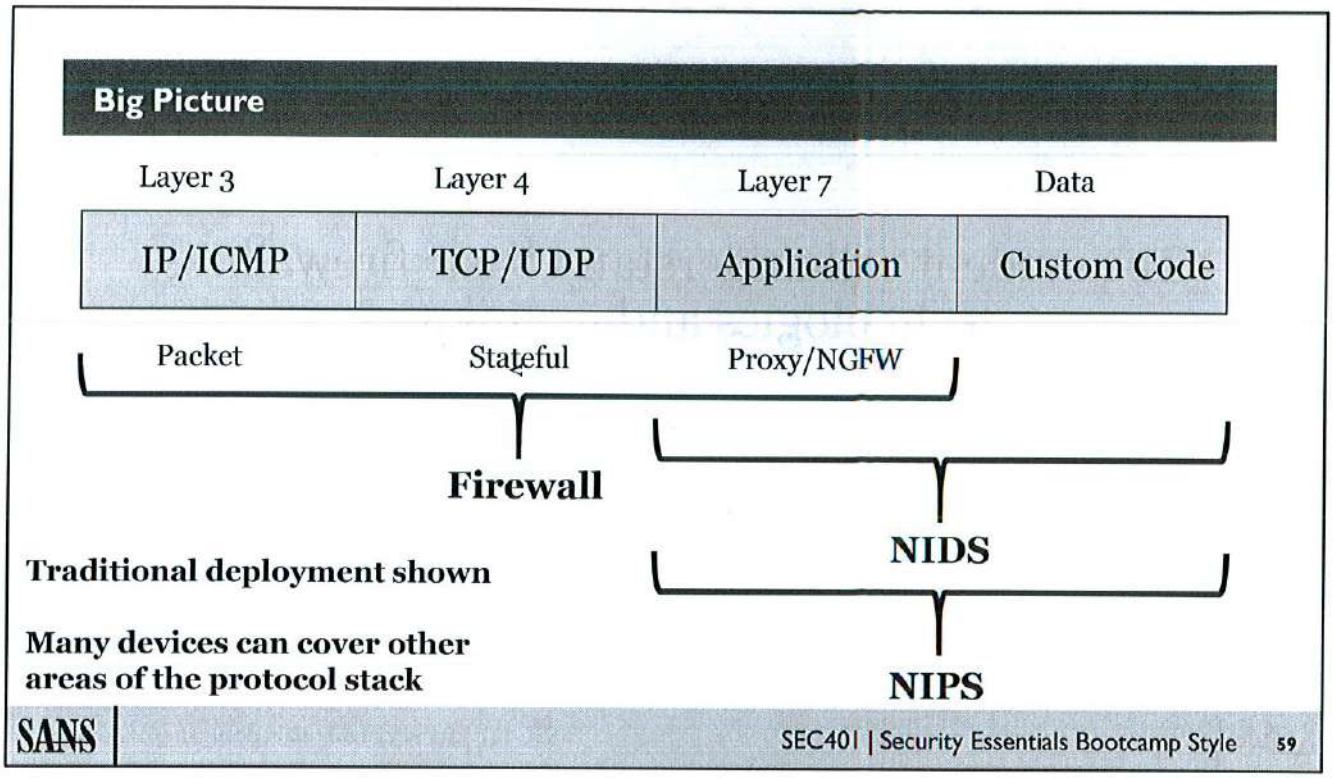
Network security devices allow us to balance the security vs. functionality pendulum. From a security perspective, the best way to protect any critical system is to completely air gap it from any network, including the Internet. An air gap means that there are no physical connections and a system is completely disconnected from any public networks. This provides 100% security and 0% functionality. The other extreme is to drive business and for all information to be fully accessible from the Internet. This provides 0% security and 100% functionality. As we can imagine, neither one of these options will work if we want to run a viable organization and stay in business. Network security allows for both functionality and security to be more carefully balanced by allowing some, but not all connectivity between trusted and untrusted networks.

There are many categories of network security devices, but the 3 main types of devices that we will look at in this section include:

- **Firewalls – PREVENTION**
- **NIDS (network intrusion detection systems) – DETECTION**
- **NIPS (network intrusion prevention systems) – PREVENTION**

As cyber security continues to be a top priority, new technologies are constantly being developed, but this section will provide a general overview of how prevention and detection can work together.

© 2017 Secure Anchor Consulting



As we have discussed throughout this course, a key premise of effective security is defense in depth. It is important to have multiple layers of protection, so if any one measure fails, effective security is still in place. Not only do you want multiple layers, but you also want to have comprehensive protection. This means you want to have a variety of preventive and detective technology deployed across the entire protocol stack. Utilizing firewall, NIDS and NIPS allows us to accomplish this, as you can see in the diagram.

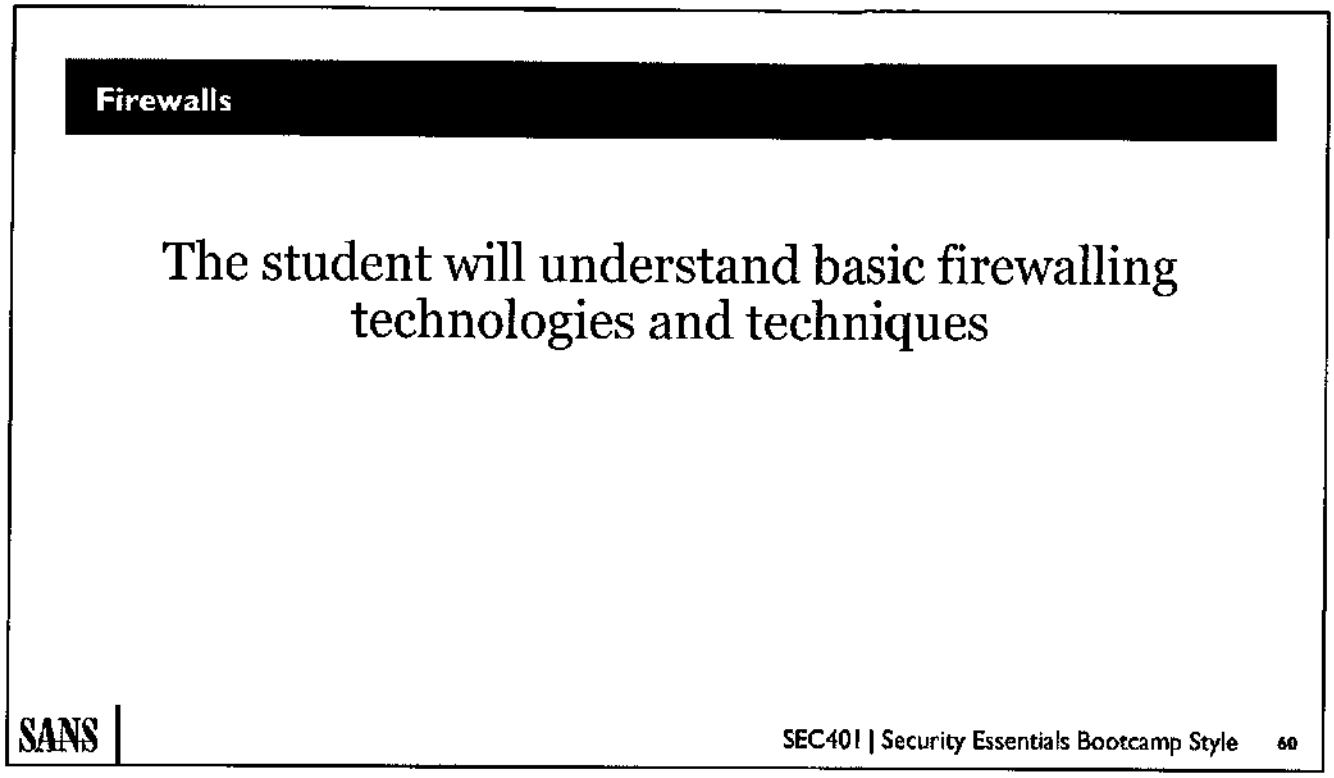
One important exercise to perform is a gap analysis, to understand where the exposures are in your network security architecture. If you are deploying a different set of network security devices or a subset of what is listed here, it is important to create a map like the above to see where additional protection is required. What makes security so exciting is that you are only as strong as your weakest link. You do not get credit for what you do right, you only get credit for what you do wrong. Therefore it is important to find your weakness because if you do not, you can be confident that the adversary will find and exploit them.

References

1. "Advanced Persistent Threat" by Dr. Eric Cole
2. "Hackers Beware" by Dr. Eric Cole

© 2017 Secure Anchor Consulting

Technet24

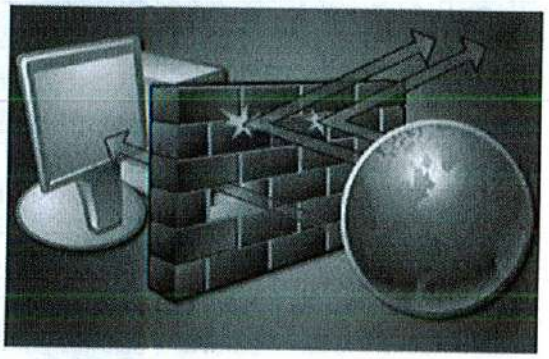


This page intentionally left blank.

© 2017 Secure Anchor Consulting

Why a Firewall?

- Preventive technology: A router with a filtering ruleset
- Reduces risks: Protects systems from attempts to exploit vulnerabilities
- Increases privacy: Makes it harder to gather intelligence about a site
- Enforces an organization's security policies



SANS | SEC401 | Security Essentials Bootcamp Style 61

Firewalls are some of the most versatile and important components in the information-security arsenal.

A firewall is a means to control what is allowed across some point in a network as a mechanism to enforce policy. It takes its name from the firewall that is meant to prevent the spread of fire from one portion of a structure to another within a building.

Firewalls are utilized at a variety of network locations, of which two are

- Between the public Internet and an organization's private internal network
- Between a PC's network interface card (NIC) and the rest of the PC

Firewalls can be implemented as

- Dedicated network appliances
- Hardware or software inserted into a network device, such as a router (that is primarily performing other duties)
- Software running on a general-purpose computer

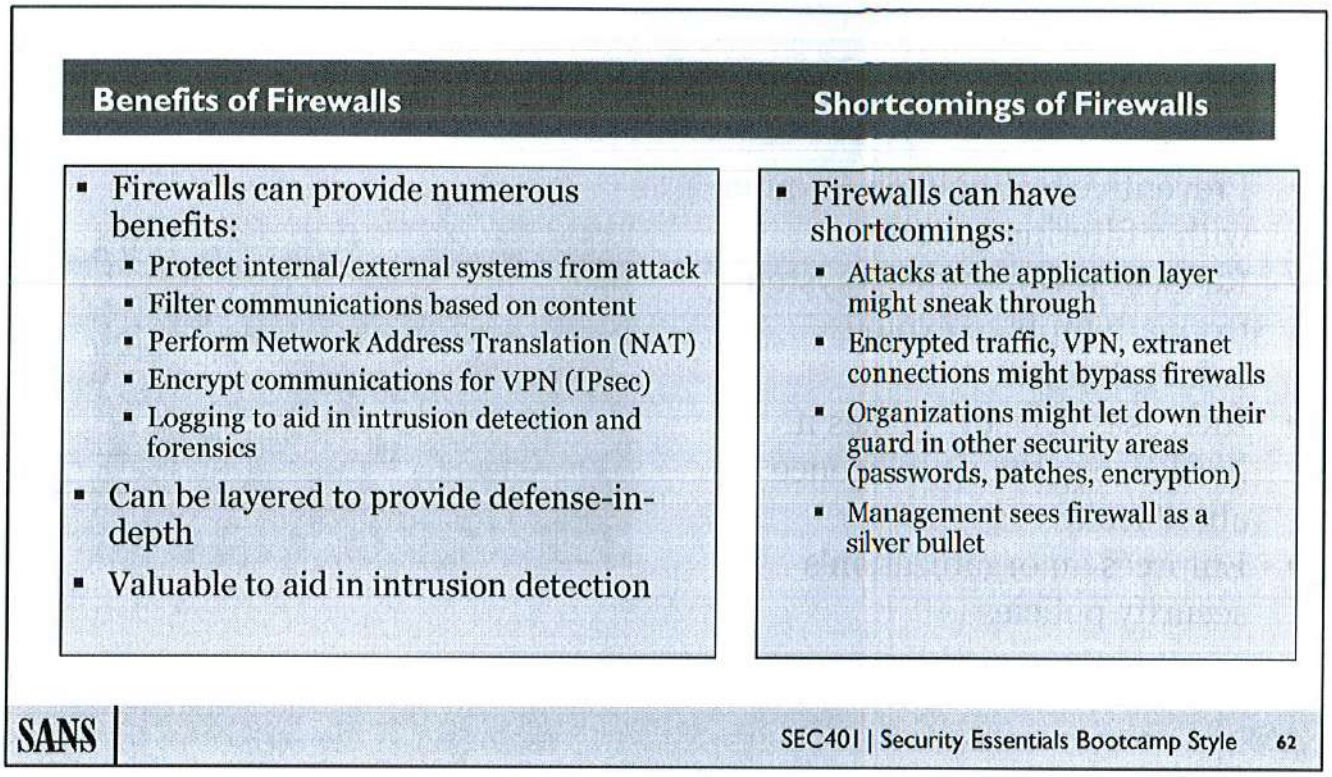
A firewall is most commonly deployed at boundaries between your site and the Internet. There is a point of demarcation where your Internet service provider's (ISP) network ends and yours begins.

Reference

1. The Role of a Firewall in Network Security - http://secure-steps.blogspot.com/2014/04/the-role-of-firewall-in-network-security.html

© 2017 Secure Anchor Consulting

Technet24



Firewalls are interesting in that they can play a variety of roles, each with significant benefits. Besides just enforcing an organization's security policies, firewalls can:

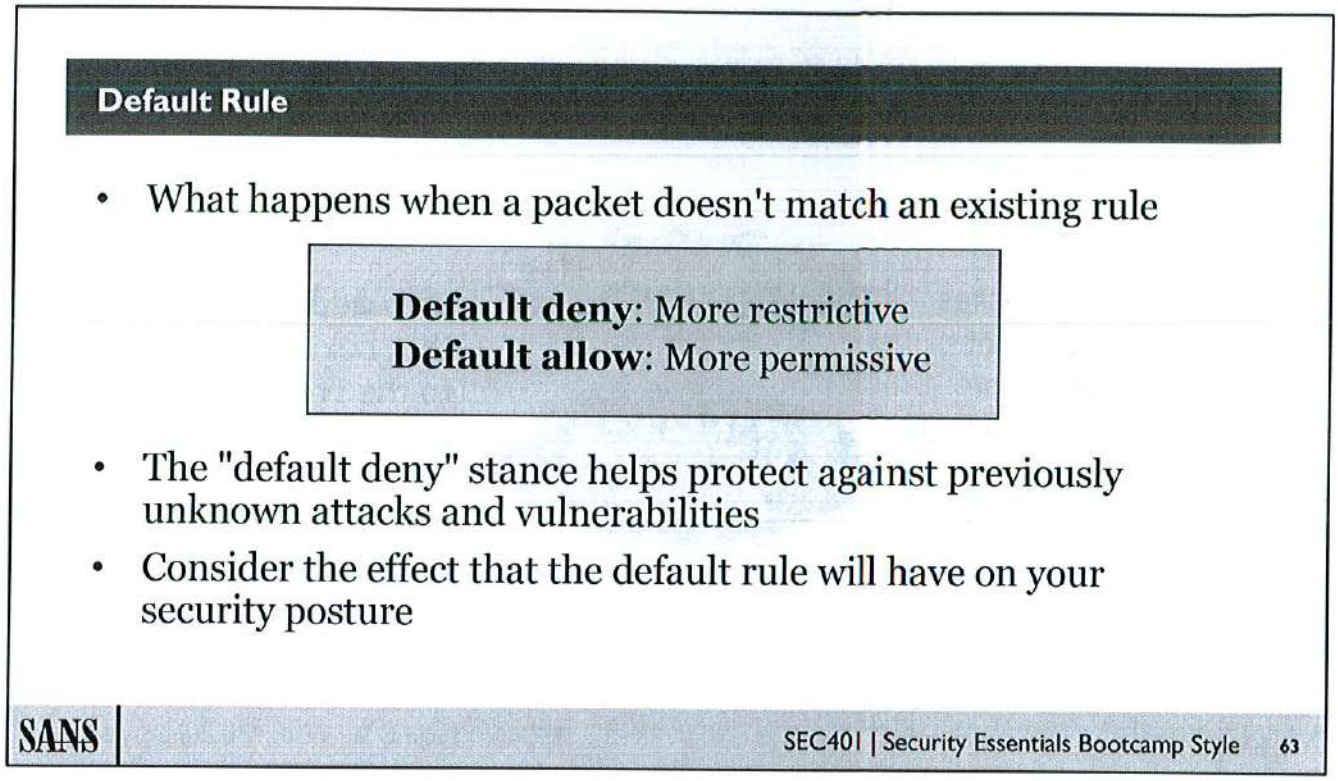
- Reduce risks by protecting systems from incoming and outgoing attempts to exploit vulnerabilities
- Increase privacy by making it harder to gather intelligence about a site
- Filter communications based upon content, such as offensive or malicious content coming in or proprietary content flowing out of an organization
- Encrypt communications for confidentiality
- Provide records concerning both successful and blocked network traffic, which might be critical for incident handling and forensics
- Serve as a "noise filter" and conserve bandwidth

Firewalls of different types can be cascaded effectively or otherwise applied in a myriad of network topologies. Some of the most secure networks intentionally use firewalls of different brands or types in series as part of a defense-in-depth strategy. Even with firewalls, defense-in-depth needs to be practiced.

With the value that firewalls offer, it can be tempting to think that they are cure-alls. They are not. Firewalls are not bulletproof. They do not stop all attacks. In fact, they can be attacked themselves and therefore have shortcomings.

Many people foolishly have blind faith in firewalls. You hear statements like, "We are behind a firewall. Why do we need to put patches on our systems or use access controls on our web servers?" One of the downsides of having a firewall is that an organization can become careless about other aspects of security. The best way to think of a firewall conceptually is like an umbrella. When you use an umbrella, it keeps a lot of the rain off you, especially your head. However, some of those raindrops get through the perimeter defense. In information warfare, we call these "leaks."

© 2017 Secure Anchor Consulting



Firewalls are designed with something called a "default rule:" If a packet doesn't match any of the other specific rules, the default rule would be implemented.

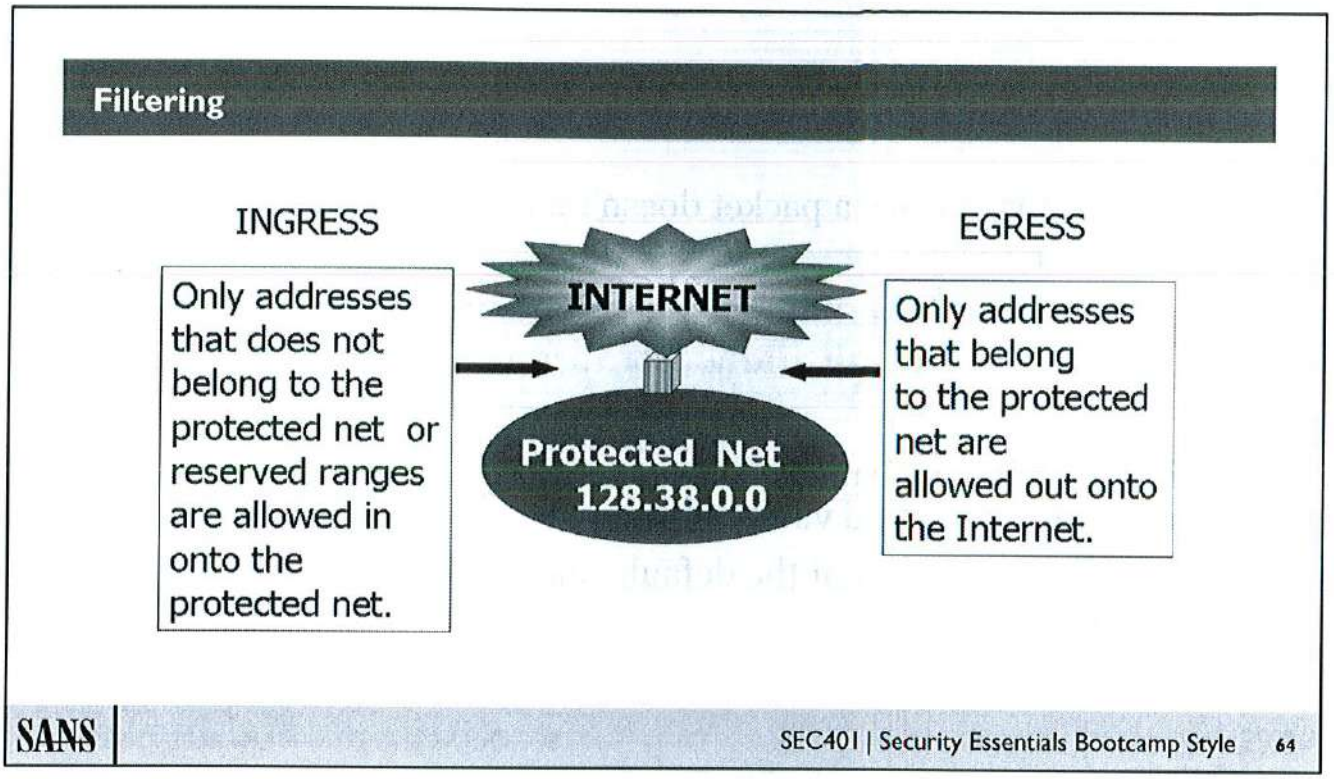
A traditional default rule is default deny. This is also referred to as deny all except that which is explicitly allowed. Firewall administrators who override this rule create an "allow all except that which is explicitly denied" policy.

This is one reason your security policy must be linked to your organizational policy. Either you make the detailed decisions necessary to establish firewall rules in accordance with the organizational policy or you make them arbitrarily. They likely will not withstand organizational pressure over time if they are arbitrary.

A default allow is very permissive and is normally not recommended. This is where anything that is not explicitly denied is allowed.

© 2017 Secure Anchor Consulting

Technet24



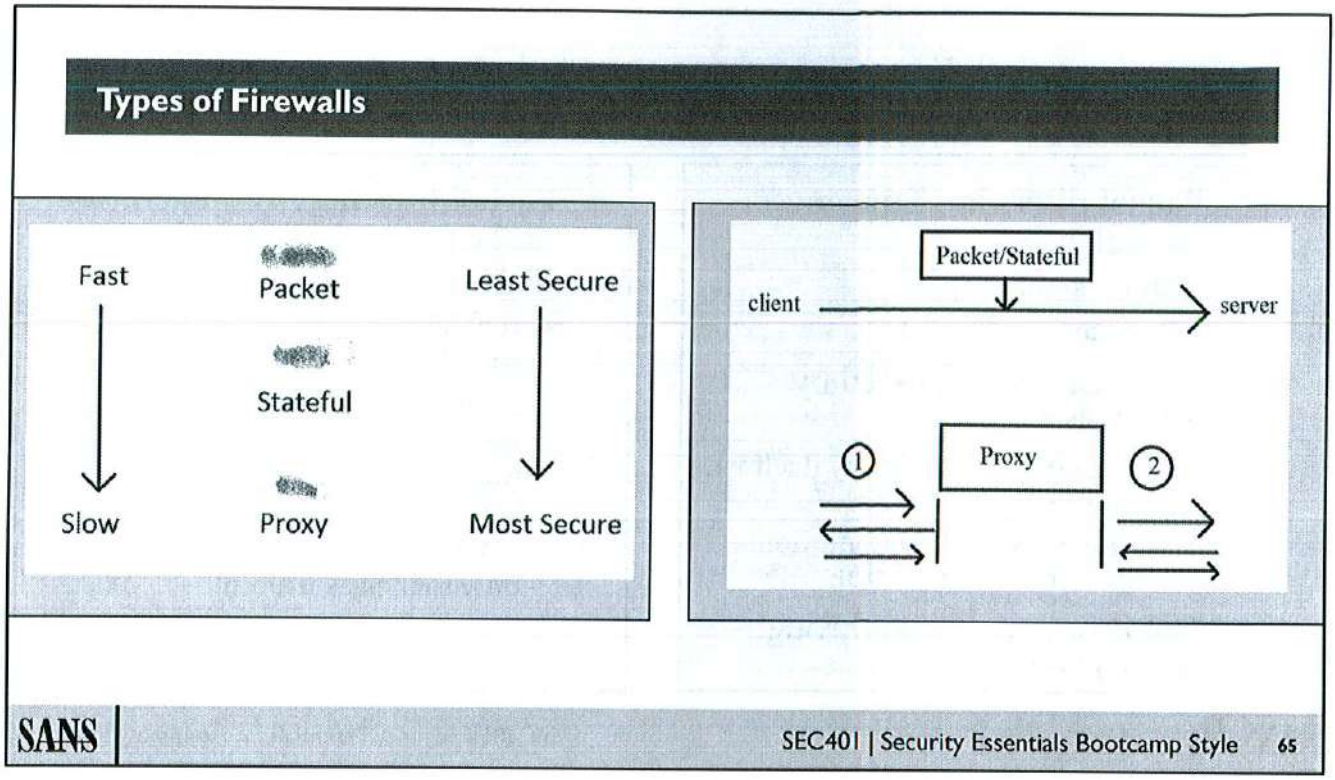
Ingress filtering refers to filtering applied to incoming traffic—from the perspective of your network. Generally, most of the firewall rules are applied to inbound traffic. Consider this simple example: All inbound packets should be dropped if they contain a source address from within the protected network address space. Whether these packets are the results of an attacker spoofing your address or a routing problem, they should not be allowed in. In the event that internal packets inadvertently have been routed to the public network, this rule will make both the routing error and the failure to block them with appropriate egress filtering conspicuous so that these errors can be corrected.

Egress filtering applies to filtering outbound traffic. The most basic egress filter simply provides filtering for addresses. Because of personal firewalls, egress filtering applies to individual computers and networks.

Flooding Denial of Service attacks often use a faked source address so that it is difficult to pinpoint the location of the attacking computer. These attacks are not elegant; they simply spew packets at the maximum rate possible. They can be launched by malicious users who are "playing" with their computer systems, but also they can be launched from compromised computers or systems infected with trojans or other malicious software. If your site applies egress filtering at the access point between your site and the Internet, you obviously are being a good neighbor (and being prudent with regard to downstream liability).

Egress filtering also is a wonderful intrusion-detection technique, utilizing your firewall log files. Suppose one of your internal machines has been infected with a malware. Indirectly, you can detect this by noting its attempts to spread through outbound traffic. Failure to detect this and take action raises issues of downstream liability.

© 2017 Secure Anchor Consulting



There are 3 general types of firewalls: packet, stateful and proxy (sometimes referred to as NGFW or next generation firewalls). It is important to remember as cyber security continues to be a big growth area, new hybrid solutions will continually emerge but in general, most firewalls fit into these categories. We will discuss each of these 3 types of firewalls over the next several slides, but just wanted to introduce them briefly.

Packet filtering examines an individual packet in order to make a decision of whether to drop or allow the packet. They are typically very fast but not very secure. Stateful filtering or stateful packet filtering keeps track of all of the packets in a state table and uses this to make a determination of whether to drop or allow a packet. Since stateful packet filtering stores and correlates information it is more secure, but this also requires resources and therefore is slower than a packet filtering firewall. With both packet and stateful, there is a single connection between the two systems that are communicating. In order to increase security, a proxy firewall is utilized which does not allow the two systems to directly communicate. As you can imagine, this increases the security but decreases the overall security.

In designing a referenced architecture we often use all 3. We place the packet filtering up front to filter out the noise. The stateful sits behind it but has less traffic to analyze because some of it was filtered out by the packet filtering device (which is illustrated with the red dots). Finally, the proxy is the slowest but only has to analyze a small amount of traffic, so the performance has a minimal impact.

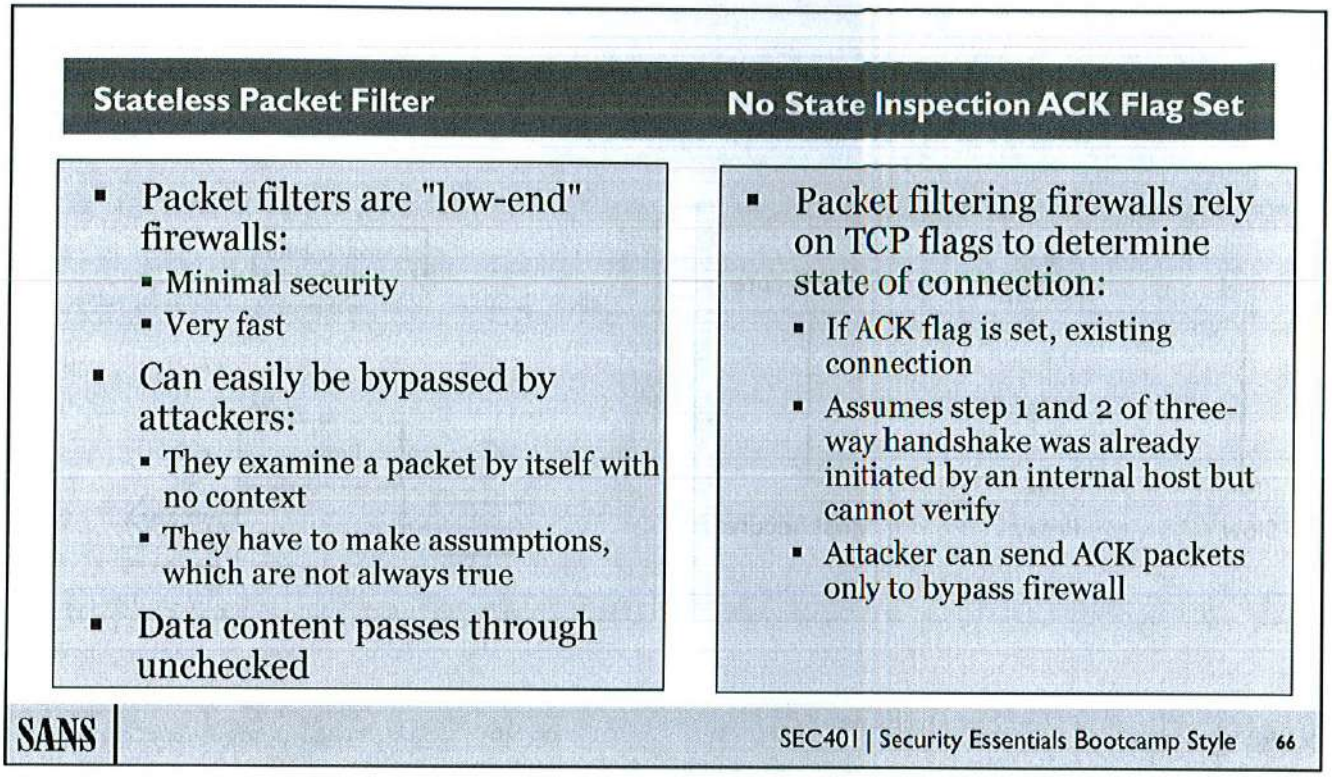
Firewalls vary in approaches and features, costs, and ease of management. In the following sections, we introduce you to the packet filter, stateful packet filter, and proxy or application gateway firewalls.

References

1. "Advanced Persistent Threat" by Dr. Eric Cole
2. "Hackers Beware" by Dr. Eric Cole

© 2017 Secure Anchor Consulting

Technet24



Stateless packet filters are low-end firewalls; they were the first to be deployed widely because they could be implemented with already existing network hardware, such as routers. Such hardware is adept at looking at fields in packets and can do so quickly, although you need to be sensitive to the load on the hardware and size it appropriately. Firewalls that use this technique are the fastest, often the cheapest, and make great noise filters ahead of more advanced types of firewalls.

The stateless packet filter's speed comes at a tradeoff, however, because this rather simplistic perimeter defense can be fooled easily; many techniques for doing this have been automated and are widely available. But, just because more sophisticated firewall technology exists and is usually needed, don't think that there's no longer a place in network security for stateless packet filters. Stateless filters have a useful role handling the simplest attacks at high speed before the packets are handed to the more stringent checks in a stateful firewall behind it.

The most significant limitation with simple packet filters is their lack of "state knowledge." As each individual packet arrives, packet filters must decide to forward it or discard it, and they must make this decision without considering any previous packets. Remember that nearly all network traffic is bidirectional. If an organization's policy is to only allow outbound traffic, what they really mean is to only allow traffic, which is initiated outbound. It is difficult for a packet filter to distinguish between inbound packets that are a consequence of an outbound-initiated connection, which must be allowed in, versus others that should be disallowed.

Because the packet filter isn't maintaining state information, it can make decisions only on whether to forward or drop traffic based on the content of the individual packets. No history information is available to determine whether this packet is the response to a previous SYN packet—the packet filter firewall can make only drop or forward decisions based on the information in the packet. Accordingly, the packet filter firewall uses the TCP flags field to determine whether the packet should be forwarded by looking for the ACK flag set. If the ACK flag is set, the firewall assumes the packet is following an initial TCP SYN packet from an internal host.

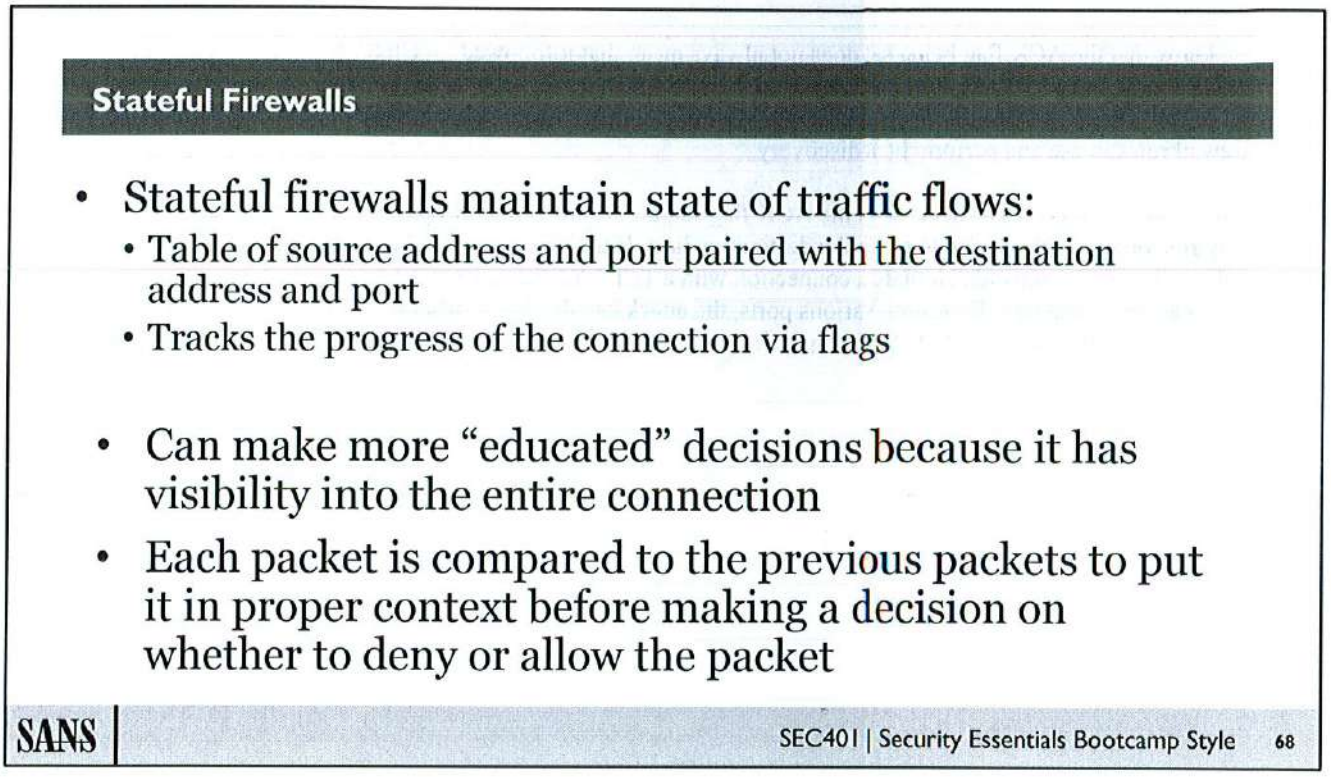
© 2017 Secure Anchor Consulting

We know that the ACK flag being set does not always mean that it followed a TCP SYN packet. An attacker can easily set any TCP flags they want and send them to a host protected by a packet filter firewall. Because the packet filter blindly passes any packets with the ACK flag set, the attacker has the opportunity to map the firewall rules in use and perform host discovery.

After a firewall receives a packet with the ACK flag set, the firewall either drops the packet if it matches a deny rule or allows the traffic to reach the destination host. If the destination host receives the traffic, it realizes that the source didn't initiate a connection with a TCP SYN packet, and typically generates a RST ACK packet in response. By testing various ports, the attack can determine what ports are being filtered by the firewall due to the lack of a RST ACK response. Ports that do respond are unfiltered. This is often referred to as an ACK scan.

© 2017 Secure Anchor Consulting

Technet24



A stateful firewall operates much like a packet filter firewall by focusing its attention on the IP and TCP header characteristics of traffic—stateful firewalls do not usually inspect any application data. Unlike a packet filter firewall, however, the stateful firewall keeps track of all the connections that are going through the firewall, so it cannot be fooled with an ACK scan. Instead, the stateful firewall uses a table of source address and source port, paired with the destination address and port information and a state flag. The state flag identifies the relationship between the source and destination address (and ports), and what the current status of the connection is. Possible values for state are as follows:

- **SYN_SENT**: A SYN packet has been sent from host A to host B, the first step in the three-way handshake
- **SYN_RECV** : A SYN ACK packet has been received from host B, the second step in the three-way handshake
- **ESTABLISHED**: The third step in the three-way handshake has been completed and the connection between host A and B is established
- **FIN_WAIT1**: One host has issued a FIN packet indicating the connection should be gracefully closed
- **LAST_ACK**: The other host has acknowledged the request to gracefully close the connection
- **FIN_WAIT2**: The other host has issued a FIN packet in response to the request to gracefully close the connection. Both sides are finished communicating
- **CLOSED**: No connection between the two hosts

By keeping track of the state of TCP connections, the stateful firewall can respond intelligently to packets out of order or packets that include malformed TCP flag combinations. This improves on the functionality of a packet filter firewall at the cost of additional overhead in maintaining a stateful tracking table for each connection through the firewall.

Because UDP and ICMP protocols do not have state flags that indicate when a connection is completed (unlike TCP, which has the RST and FIN flags), stateful firewalls rely on a timeout duration for these protocols to determine when they can be safely removed from the state table. Once the timeout duration is exceeded, traffic

© 2017 Secure Anchor Consulting

from the external host will be dropped until another UDP or ICMP packet for the source/destination address and port originates on the inside of the network.

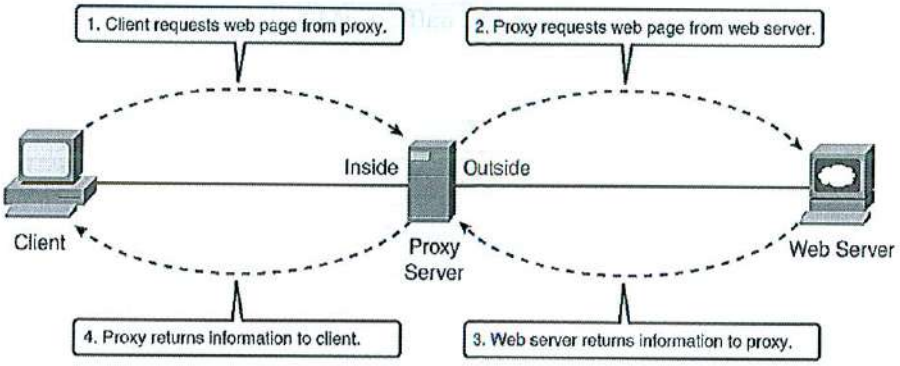
ICMP error packets are another issue for stateful firewalls. If an internal client connects over UDP to an external server that is not listening on the destination port, it is appropriate to issue an ICMP Port Unreachable error message in response. The stateful firewall isn't able to differentiate this legitimate use of Port Unreachable messages from an attacker sending this same traffic to an internal host, so it has to blindly accept the traffic.

Fortunately, there is another alternative for firewalls to improve on the functionality of a stateful firewall by incorporating the inspection of payload traffic in addition to maintaining a state table for stateful protocols.

Technet24

Proxy or Application Gateway

- Maintains complete TCP connection state and sequencing through two connections:
  - Session user to proxy
  - Proxy to destination server



Packet filters are fast, but they can be fooled; they trade speed for security. Proxy firewalls are at the opposite end of the spectrum. Among firewalls, they generally are the slowest in performance and the most inconvenient to manage, such as when a new protocol isn't yet supported; however, proxy firewalls usually provide the best security. In an environment that requires the high security of a proxy firewall, the default rule is always "deny if not explicitly allowed."

Proxy firewalls essentially tear down each packet layer-by-layer on one interface and build it back up on the opposite interface. From the perspective of the source, the traffic flows to the destination. But, the traffic actually is delivered to a virtual destination just inside the proxy firewall, on the input side, where it is disassembled and examined. If the policy being enforced allows this traffic through, it is regenerated (proxied on behalf of the source) on the output side of the proxy firewall. All of this effort results in poorer performance and cost, but tighter security.

The proxy firewall must maintain a complete TCP connection state and sequencing through two connections:

- Session user (the source) to the proxy
- Proxy to the destination server

Proxy firewalls use process tables to keep the connections straight.

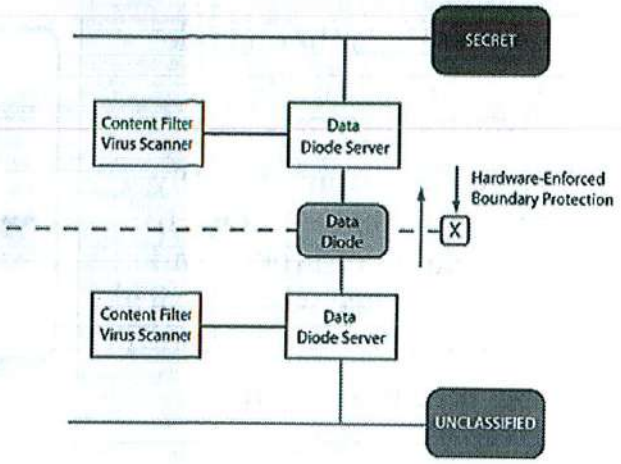
Reference

1. Chapter 1: Types of Firewalls - http://www.networkworld.com/article/2255950/lan-wan/chapter-1--types-of-firewalls.html

© 2017 Secure Anchor Consulting

Data Diodes

- A diode is a semiconductor device with two terminals, typically allowing the flow of current in one direction only
- A data diode typically references military technology that moves data into classified networks without the risk of leaking classified information



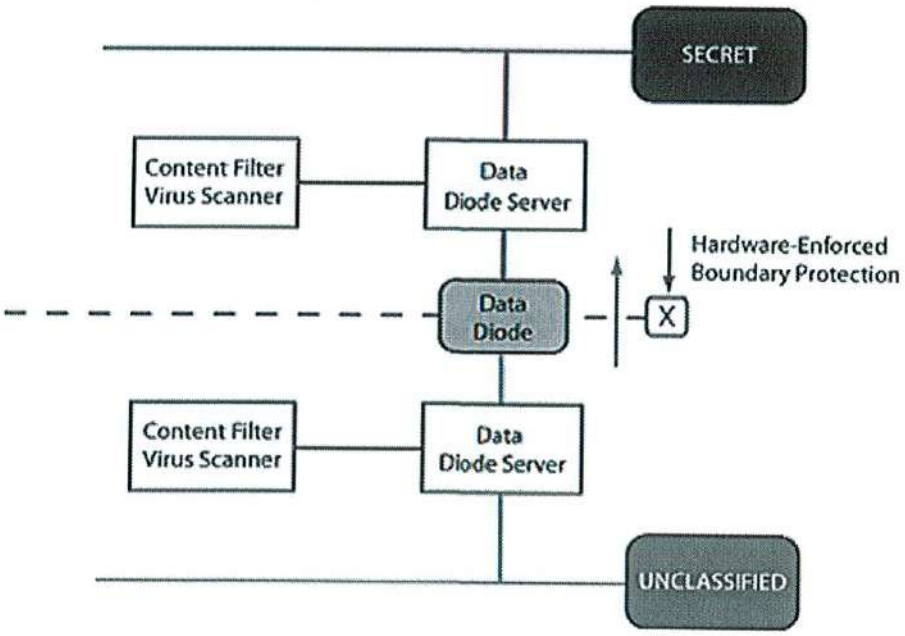
SANS | SEC401 | Security Essentials Bootcamp Style 71

A diode has little to zero electrical resistance to current flow in one direction, and high or infinite resistance to current flow in the other direction, thus creating an electric component that allows current flow in a single direction. Typically, a diode has a bar on one end indicating the terminal where current will flow out of and not in; the input side is referred to as the anode, and the output side is referred to as the cathode.

Data diodes are used routinely to protect secrecy in military and government networks. Data diodes are hardware-focused; software associated with diodes tends to be fairly primitive. In principle, you can turn diodes around to send data out of safety-critical networks instead of into confidentiality-critical networks.

Reference

1. SANS ISC410

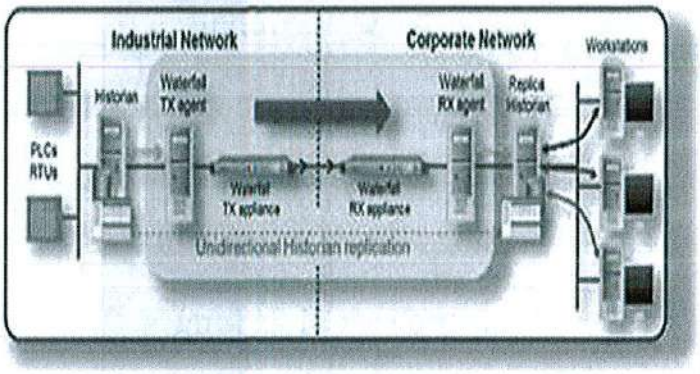


© 2017 Secure Anchor Consulting

Technet24

Unidirectional Gateways

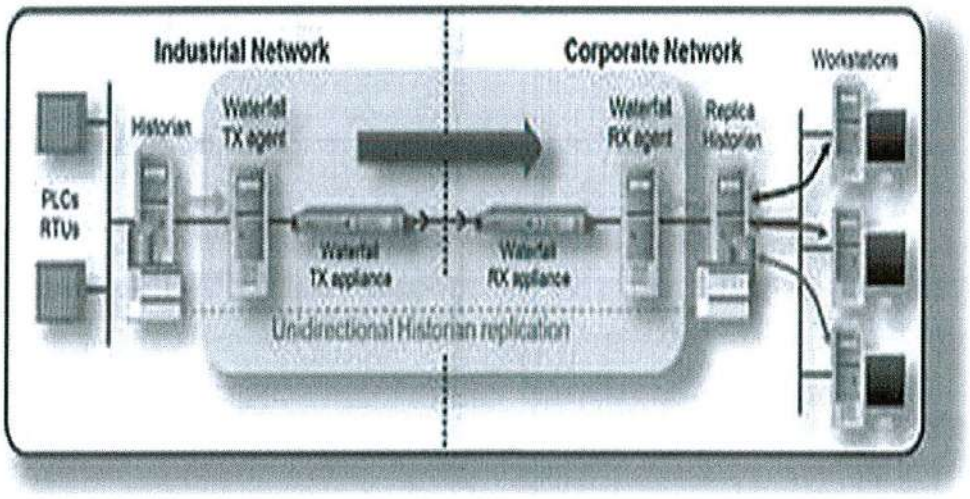
- Devices with multiple network cards that handle the data handoff with software controls
- Layered solutions that rely on software components to gather data, and then send to an appliance with physical one-way data flow capability
- Optical isolation is the industry standard.



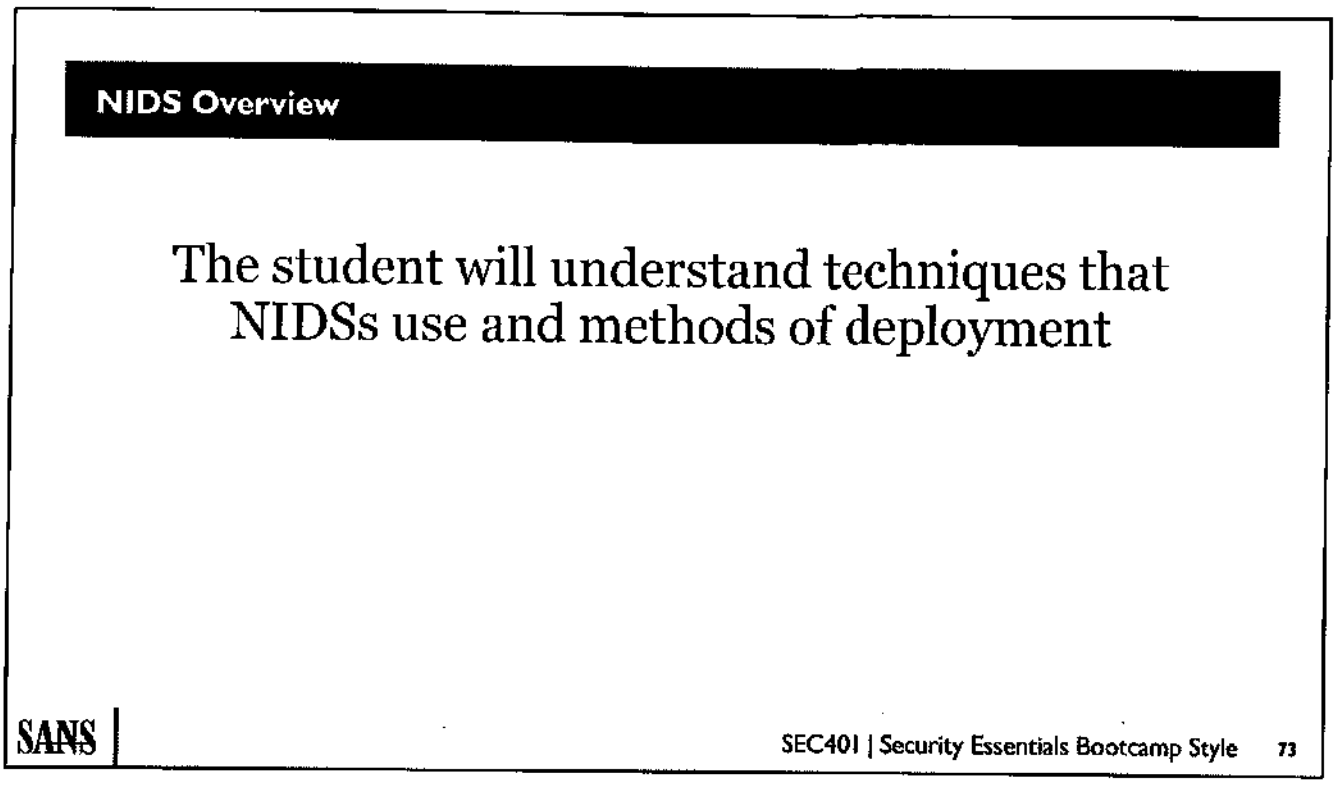
Unidirectional gateways is the term the ISA SP-99 / IEC 62443 standards are using to refer to the combination of hardware and software that allows information to flow out of a "secure" network unidirectionally. Numerous solution providers offer a method of solving this problem; some provide optical network interface cards, which are capable of only sending or receiving data. Others provide a network appliance with this kind of optical isolation under the hood. Still, others provide a pair of network devices, each with a copper and an optical interface, one appliance able only to transmit on the optical interface and the other appliance able only to receive on that interface.

References

1. How to Keep Nuclear Plants Safe? Let Me Count the Ways - http://www.power-eng.com/articles/print/volume-117/issue-11/features/how-to-keep-nuclear-plants-safe-let-me-count-the-ways.html
2. This diagram shows how unidirectional gateway networks work to keep networks safe from cyberattacks. Courtesy: Waterfall Security Solutions
3. SANS ISC410



© 2017 Secure Anchor Consulting



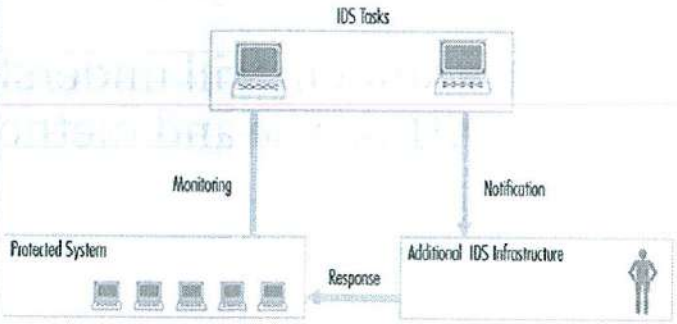
This page intentionally left blank.

© 2017 Secure Anchor Consulting

Technet24

What Is an IDS?

- Intrusion detection system (IDS)
- IDS reports attacks against monitored systems/networks
  - Alarm system
- Mature technology
- Requires monitoring, alerting, and reaction



Intrusion detection is the process of monitoring activity on a host or on the network, identifying clues that might indicate an attempted or successful security breach. An intrusion detection system (IDS) monitors activity that is known or suspected to be malicious in its intent, raising alerts to a human to be analyzed. The person who is responsible for responding to the alerts (incident handler) uses the information generated by the IDS to identify the intent of the suspicious activity and takes some action based on the analysis.

In this sense, an IDS is an alarm system for identifying undesirable activity on your network or hosts. Just like an alarm system doesn't stop a thief from trying to steal, the IDS doesn't provide any protection from attackers. Instead, the IDS alerts you when there is attack activity on your network, allowing the incident handler to respond to the activity according to the severity of the alerts. Organizations should not deploy IDS tools as a primary method or defense to protect their resources. Instead, IDSs should be utilized in conjunction with firewalls, antivirus software, vulnerability assessment and management, and patch management tools to support a defense-in-depth posture.

The technology supporting IDS is mature, having been actively in use for many years. Many organizations make good use of their IDSs in identifying attacks and several other positive measures, although many organizations with IDSs do not leverage their capacity for various reasons.

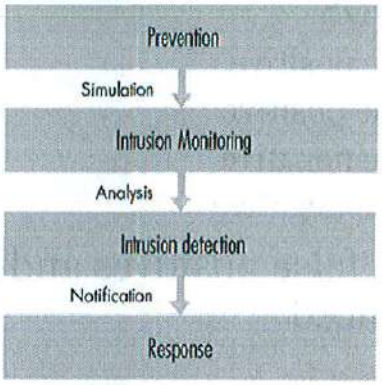
Reference

1. Intrusion Detection Systems(network intrusions; attack symptoms; IDS tasks; and IDS architecture) - http://www.ids-sax2.com/articles/IDS(tasks&architecture).htm

© 2017 Secure Anchor Consulting

What IDS Is Not

- Not a replacement for firewalls, strong policies, system hardening, timely patching, and other defense-in-depth techniques
- Not a low-maintenance tool
- Not an inexpensive tool
- Not a silver bullet



As mentioned in the previous slide, IDS technology is not a replacement for other security mechanisms that protect your network. Organizations should consider the deployment of IDS technology only after leveraging firewalls, strong policies, system hardening, and other defensive techniques. Remember the alarm system analogy? The IDS only informs you when you are being attacked. It does not prevent or stop attacks from being effective.

A common mistake in the deployment of IDS technology is to spend capital money on the acquisition and deployment of tools without a maintenance and utilization plan. The costs of maintaining and using the IDS far outweigh the costs of acquiring the tools. Depending on the configuration of your network, the placement of your IDS systems and the overall strength of your security policy, monitoring, and reacting to IDS events could easily be a job for an entire team of analysts. Furthermore, it takes a well-trained analyst to understand and correctly interpret the alerts generated by an IDS, which adds to the total cost of ownership for the organization.

Finally, an IDS simply isn't the silver bullet for organizations looking to secure their networks. Even the best-trained analysts can process only an alert to a specific point. Without comprehensive policies governing information security, and without a clear understanding of what actions are needed to protect the business assets, an analyst won't be able to react to alerts with any level of consistency.

Reference

1. Intrusion Detection Systems(network intrusions; attack symptoms; IDS tasks; and IDS architecture) - http://www.ids-sax2.com/articles/IDS(tasks&architecture).htm

© 2017 Secure Anchor Consulting

Technet24

IDS Alerts

- Alerts are generated from events of interest (EOI)
- An analyst must understand four types of events from the IDS
  - True positive
  - False positive
  - True negative
  - False negative
- Both false positives and false negatives must be balanced

SANS | SEC401 | Security Essentials Bootcamp Style 76

When an IDS sees activity, any kind of activity, an analyst must identify and classify the activity. This classification typically falls into two groups: positive for an event of interest (EOI) or negative for an EOI. An event of interest can be anything the analyst wants to identify with the IDS, including specific hacker tools, particular content or keywords in e-mail or instant messages, or even a specific filename being transferred between hosts. If the IDS sees activity that it believes is not an EOI, it classifies this as a negative event and continues without alerting the analyst. If the IDS sees activity that it believes does correspond to an EOI, it classifies this as a positive event and alerts the analyst.

Unfortunately, IDSs are not always correct and can make wrong decisions based on whether events are benign or malicious. The analyst inevitably has to work with four different classifications of events:

- **True positive**: In these cases, the IDS worked as intended and correctly flagged the activity as anomalous behavior that might be malicious. True positives generate alerts for the analyst to process.
- **False positive**: A false positive case is where the IDS generates an alert flagging hostile activity, which was benign. False positives generate alerts for the analyst to process, who then must decide how to handle the activity.
- **True negative**: A true negative event is what we want the IDS to see, the cases where data does not indicate any malicious activity, and the data is correct. In the case of a true negative, the IDS does not generate an alert for the analyst.
- **False negative**: A false negative event is when the IDS identifies data as benign when, in fact, it is malicious. A false negative does not generate an alert for the analyst.

In a perfect world, the IDS would generate only true positives and true negatives. Unfortunately, the nature of data analysis and the tactics of attackers make this task difficult, so we are forced to accept the reality of false positives and false negatives from our IDS systems.

If the IDS flags traffic as positive for malicious activity, the analyst gets an alert that he has to spend resources on to analyze. In some cases, false positives are only a nuisance for the analyst—cluttering the alerts from the IDS to the point where it becomes difficult to differentiate false positive from true positive attacks. In other

© 2017 Secure Anchor Consulting

cases, it is difficult to differentiate a false positive from a true positive, and the analyst has to perform an additional investigation to determine what really happened on the network or system in question. Fortunately, this problem is getting a lot of attention from vendors who are adding improved intelligence to reduce the number of false positives.

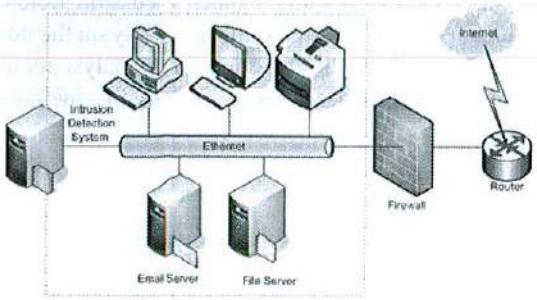
The false negative is a less popular discussion topic with IDS vendors because it signifies a weakness in their product to correctly identify attack activity on the network. A false negative is the worst-case scenario because it does not provide any information to the analyst about attacks against systems. A smart attacker might employ IDS evasion techniques designed specifically to generate false negative events on the system to avoid detection.

© 2017 Secure Anchor Consulting

Technet24

NIDS Overview

- Deployed as a passive sniffer/sensor at network aggregation points
  - Captures traffic
- Detects events of interest on the network
- Uses signature, anomaly, or application/protocol analysis



NIDS Overview

Let's begin our focus on the techniques used by network intrusion detection systems (NIDSs) to identify events of interest on the network. This variety of IDS collects packets from the network in a passive manner for analysis. Each packet that is collected is processed to identify events of interest and reported to the analyst accordingly.

To collect the necessary traffic information, the NIDS is deployed at traffic aggregation points in the network, typically with a network "tap" or mirrored interface that sends a copy of all the network traffic to the IDS. This way, the IDS can monitor all the traffic for downstream devices, up to the IDS's limitations in throughput and processing capacity.

A NIDS device can be a server or an appliance with a hardened operating system that makes it resistant to attack. Being in a position to monitor all the traffic on the network makes the IDS a valuable target for an attacker looking to capture information on the network. Vendors who produce IDS tools go to significant lengths to reduce the potential for attack on their IDSs by reducing the number of available services on the device, using strong encryption for any communication between the IDS and monitoring stations.

NIDS devices use a few different methods to identify events of interest on the network, including signature analysis, anomaly analysis, and application or protocol analysis.

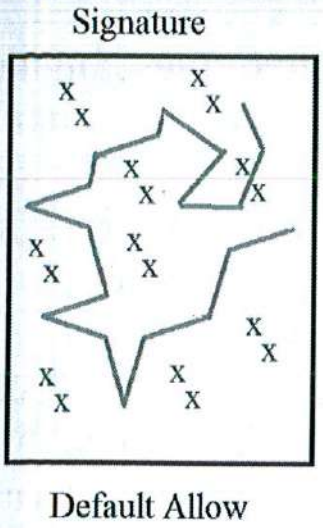
Reference

1. Intrusion Detection System - http://www.evoss.com.my/product/ids.html

© 2017 Secure Anchor Consulting

How Signature Analysis Works

- Performs pattern matching
- Rules indicate criteria in packets that represent events of interest
- Rules are applied to packets as they are received by the IDS
- Alerts are created when matches are found



Signature analysis is the most common method of identifying events of interest on the network. In its simplest form, a unique characteristic is identified for a particular EOI, and a signature is created to identify that characteristic, raising an alert from the IDS.

In practice, however, signature analysis can be a complex mechanism, requiring careful analysis and forethought when identifying the unique characteristics of hacker tools and other EOIs. Correctly identifying the unique qualities of the EOI is a critical component of the usefulness of any IDS; incorrectly identifying the "signatures" of hacker events ultimately results in false positives and false negatives from the IDS.

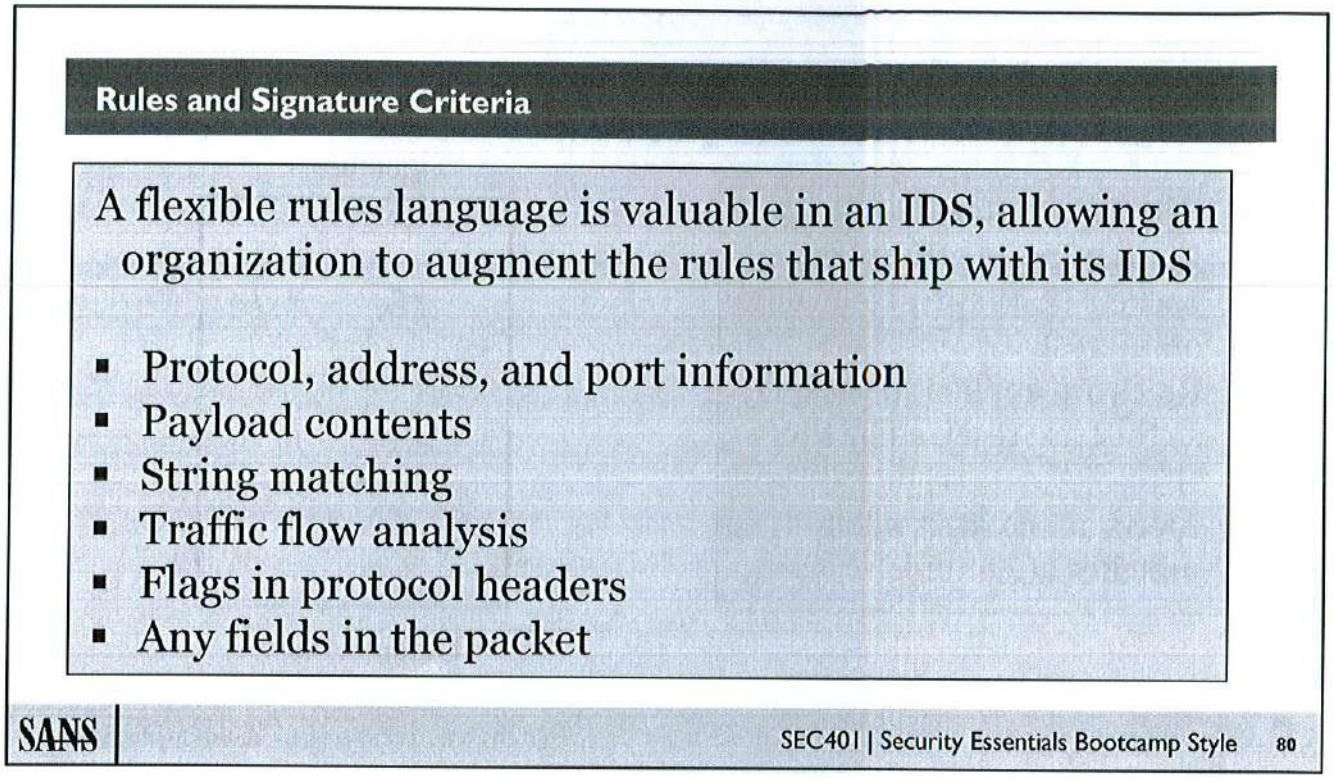
Most implementations of signature analysis utilize a series of rules, where each rule identifies a particular EOI. Each rule identifies the characteristics for the EOI using the criteria made available for analysis by the IDS. This might be the ability for the IDS to search for a particular string in a packet, or the checksum for a particular file, or more complex rules that include multiple characteristics for identifying events.

When the IDS starts, it reads through each of the rules it is configured to alert with and builds analysis and lookup tables that serve to optimize the analysis of data. As the IDS receives data to be processed, it references the lookup tables that consist of the configured EOI characteristics and generates alerts when matches are found.

This process is largely transparent to the analyst using the IDS who only has to understand the classification criteria and the alerts that are generated by the IDS. If an analyst wants to create custom rules, she must also understand the rules language to identify the desired EOIs.

© 2017 Secure Anchor Consulting

Technet24



A flexible rules language is valuable in an IDS, allowing an organization to augment the rules that ship with its IDS. With custom rules, an analyst has the ability to increase or decrease the granularity of monitoring for specific networks or hosts, and can quickly add rules to detect vulnerabilities, exploits, and virus and worm activity or other undesired activity on the network.

The language used to develop rules can be complex, because it addresses so many characteristics of patterns that might be sought after as an EOI. We investigate the rule language that is used by the network IDS tool Snort later in this module. Most IDS tools allow the analyst to examine and classify data on the following packet characteristics:

- **Protocol information**: Generating an EOI for specific protocols. This is usually Layer 3 protocols (IP, NetBIOS) or Layer 4 protocols (TCP, UDP, ICMP).
- **Address information**: Generating an EOI based on a specific source or destination address. This feature can be useful for monitoring traffic on an exclusive basis, generating an alert when traffic that does not match a list of internal IP addresses reaches a specific server IP address.
- **Port information**: Allows an analyst to generate an EOI on specific source or destination ports. This is commonly used to identify activity from worms or other malware that use specific TCP or UDP port assignments. This feature is also commonly used to identify ICMP type and code information in addition to the traditional port assignments used by TCP or UDP.
- **Payload contents**: One of the most common mechanisms for identifying specific EOIs, an analyst can identify specific payload contents in partiality or totality. For example, if an analyst were looking to identify buffer-overflow attacks, he might generate a rule to alert when the NOP instruction was found repeatedly in packets, generating an event when 20 or greater consecutive 0x90 values were found.
- **String matching**: Generating an EOI based on a specific string that is found in a packet. This feature can be used as a mechanism for identifying specific hacker techniques (such as looking for the "Login incorrect" string to identify failed login attempts against a UNIX system) or possibly as a mechanism to detect policy violations (such as looking for offensive keywords in e-mail or web-browsing activity). String matching rules are often free-form, where the IDS searches through the entire packet for a matching string instead of examining from a specific offset of the packet.

© 2017 Secure Anchor Consulting

- **Traffic flow analysis**: A rule feature that identifies when traffic flows from an outside to an inside source or vice versa. This feature is usually used in conjunction with other signature criteria to reduce the rate of false positive alerts by excluding traffic leaving the server as malicious. Although this is not always possible, it is a useful feature in many analysis scenarios.
- **Flags in protocol headers**: Generating alerts based on specific flags in protocol headers, notably the IP, TCP, UDP, and ICMP protocol headers. For example, an analyst might want to identify all the packets that are using a reserved portion of the IP header; they can create a rule that flags any traffic that has the reserved bit set. Of course, the IDS and the rules language must have support for the protocol to be assessed; otherwise, the analyst must make use of other rule signature criteria to identify the desired conditions.

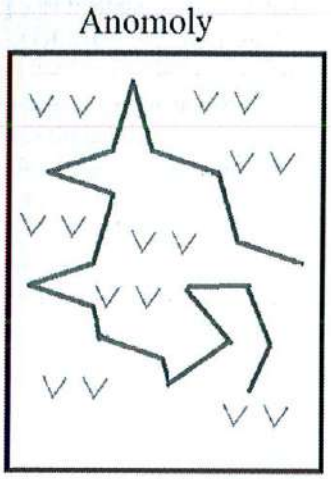
This list is a short representation of all the features available with modern IDS tools when developing rules for assessment. Now that we have a good understanding of how rule-based assessment works, let's move on to IDS anomaly analysis.

© 2017 Secure Anchor Consulting

Technet24

How Anomaly Analysis Works

- Baseline of network must be performed
  - Requires an understanding of what "normal" is
- Flags anomalous conditions in traffic on the network
  - Unexpected conditions are identified as suspicious
- Can catch zero-day exploits

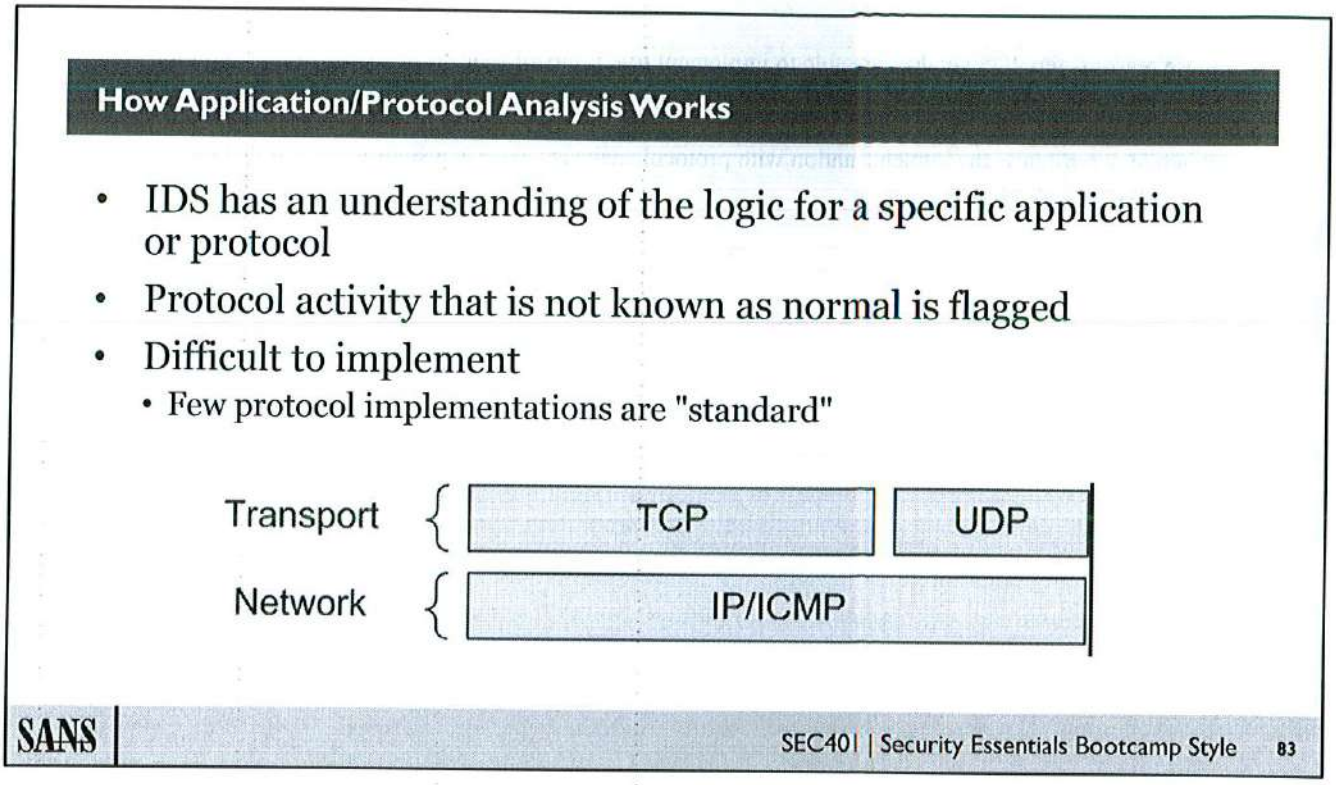


Anomaly analysis is another technique that IDSs use to identify events of interest on the network. Unlike specific signature analysis, anomaly analysis is based on events for specific protocols and applications that are outside the typical operating conditions. The IDS can leverage data from these unusual events to generate events to be analyzed by the intrusion analyst.

Anomaly analysis is usually made available from the IDS vendor for specific applications or protocols. A vendor identifies specific conditions that are not part of the normal functional behavior for the application and identifies conditions for which the IDS should generate an alert. This analysis is typically based on traffic analysis for the application but can lead to false positives. The vendor must establish an understanding of what normal behavior is for the application, typically based on baseline information including packet captures of expected conditions for the protocol and specifications that describe the intended protocol behavior.

This type of analysis is inclusive-based, meaning the IDS vendor identifies the conditions that are anomalous through its analysis of the protocol and its expected behavior. Only those conditions that are identified by the vendor will be reported by the IDS. The next analysis method, application/protocol analysis, makes use of an exclusive method for detecting undesirable events on the network.

© 2017 Secure Anchor Consulting



Where traditional anomaly analysis uses a set of conditions that the IDS looks for to flag an event, protocol analysis works by carefully examining the entirety of protocols and how they operate. Based on the implementation and specifications for protocols, the IDS develops the logic to understand how the protocol operates in its input, processing, and output phases. After the IDS has a complete understanding of the protocol and how it operates, it can use an exclusive method of identifying anomalous conditions on the network. Any use of the protocol that does not function within the definition of how the IDS understands the protocol is identified as anomalous activity, which generates an alert.

The use of protocol analysis as an IDS technique is powerful because it can detect both known and unknown attacks against a protocol. This can include Denial of Service attacks, buffer overflow attacks, misuse of a protocol, and other opportunities for abuse. Negatively, protocol analysis won't be able to detect the attacks that are successful within the bounds of the defined protocol, including attempts to exploit configuration deficiencies or upper-layer protocol weaknesses.

Furthermore, the implementation of this kind of a protocol is difficult, requiring the IDS vendor to overcome several hurdles for implementation:

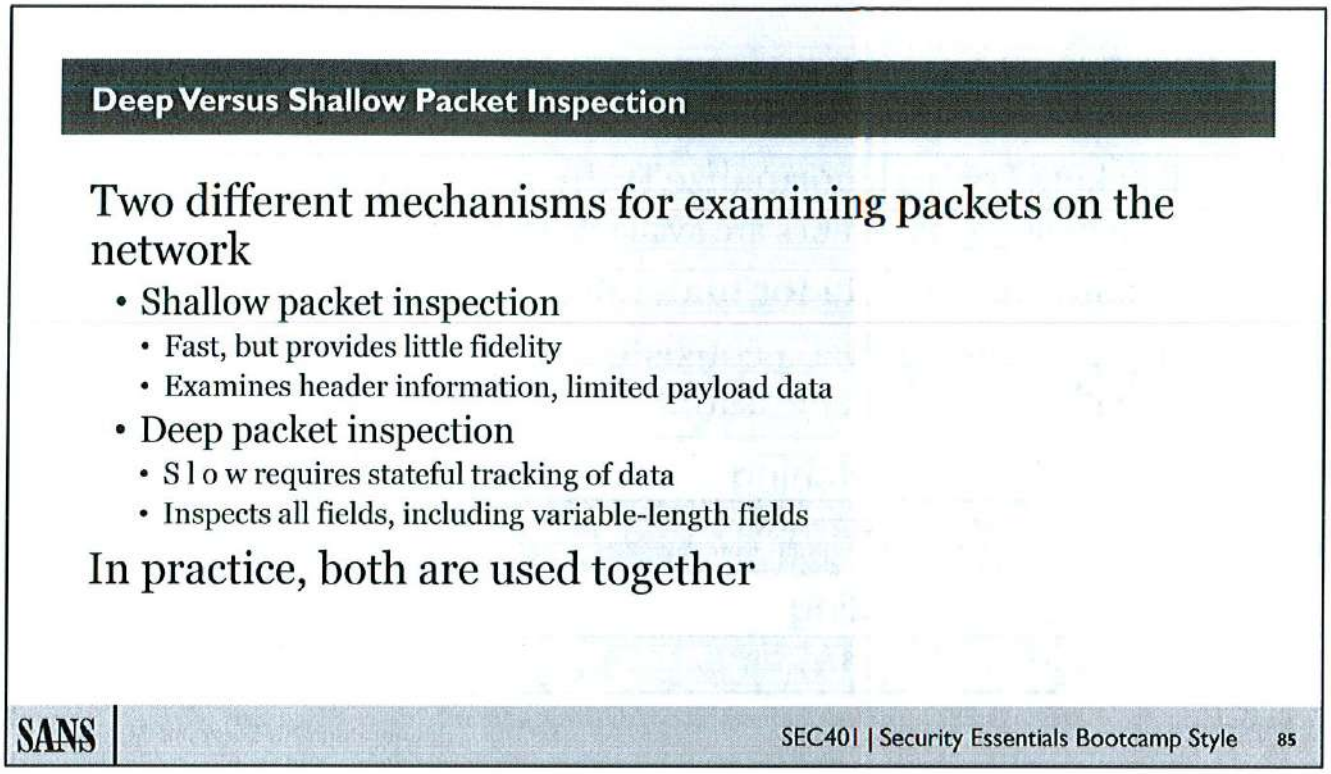
- **Standards definition**: The IDS vendor has to clearly understand the application and how it processes data to be able to identify what is "normal" behavior. This is a significant undertaking for the vendor, who has to devote significant resources toward the analysis and understanding of each protocol it wants to support.
- **Implementation nuances**: Even with a consistent design specification for a protocol, different manufacturers can implement protocols uniquely. The vendor-specific implementation of a protocol might be due to vague specifications that do not clearly understand how protocols work in all cases, or by adding "enhancements" to the protocol based upon requests from customers. In both cases, the IDS vendor has to analyze the protocol in utilization to understand its behavior.
- **Changes to a protocol**: As protocols are used, they are logically improved over time. Improvements imply changes to a protocol, which forces the IDS vendor to keep changing its understanding of the protocol over time. This can perpetuate the cost to a vendor implementing this method of intrusion detection.

© 2017 Secure Anchor Consulting

Technet24

For these reasons, few IDS vendors are able to implement true protocol analysis as a detection method for their IDS products. Vendors that do support this feature support a limited number of specific protocols that are well documented and consistently implemented. Although the ability to detect attacks that are not yet known is an attractive opportunity, the implementation with protocol analysis comes at a significant cost that must be carefully reviewed by vendors.

© 2017 Secure Anchor Consulting



From a network-based IDS perspective, vendors have implemented two different mechanisms of inspecting traffic, typically with rule-based analysis measures.

With shallow packet inspection, the IDS processes only a portion of the packet for analysis. This method of analysis extracts and evaluates the contents of a limited number of fields within the packet at predictable offset locations. The advantage of this method of analysis is that it is fast, it can be performed at near wire-speed with optimized hardware. Examples of events of interest that can be identified with shallow packet inspection include:

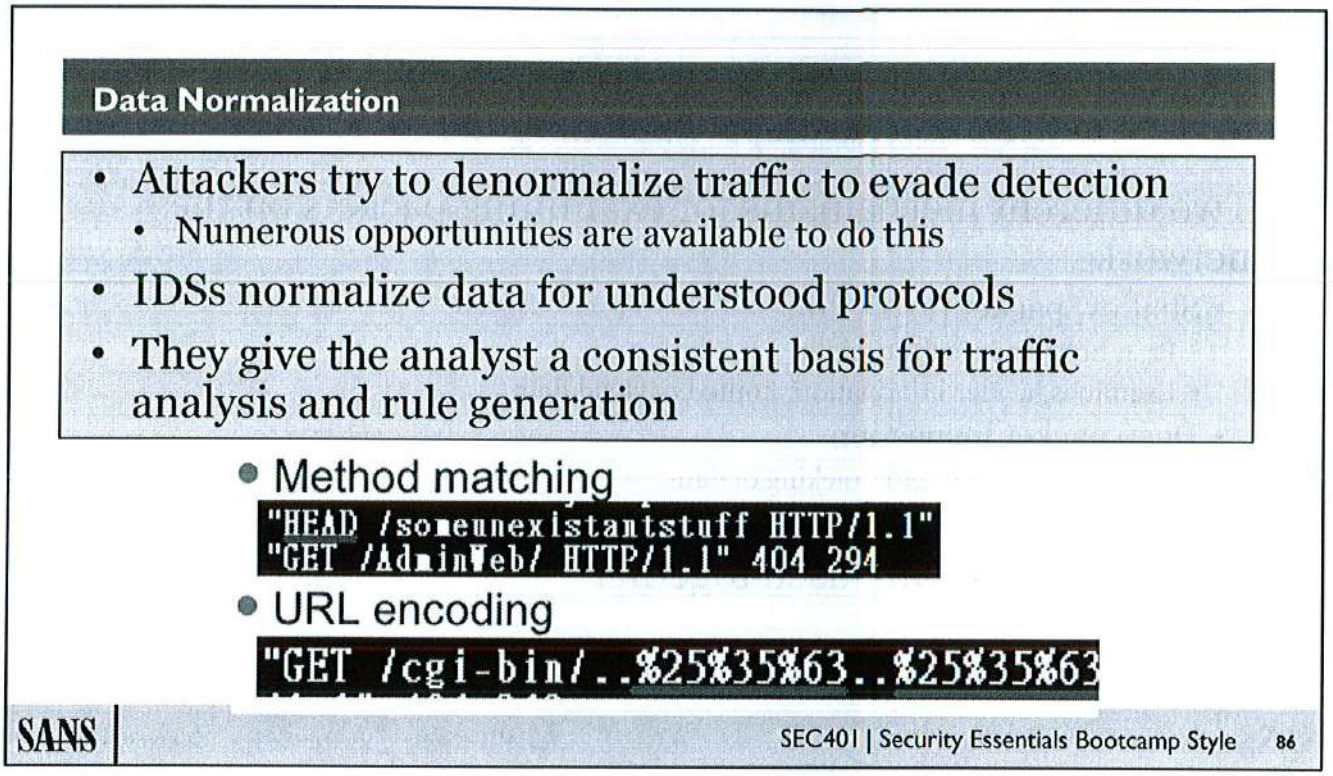
- Source and destination address and port information from the IP and upper-layer protocol headers
- Specific ICMP error messages
- Undesirable TCP flag combinations (for example, "SYN/FIN")
- Impossible fragmentation combinations (gaps or overlaps)
- Packet size information (for example, too-small UDP packets)

In opposition to shallow packet inspection, deep packet inspection performs a full analysis of the packet, including the evaluation of fixed-length and variable-length fields.

Deep packet inspection is traditionally deployed at an application-level firewall gateway, where the gateway has a complete understanding of the protocol and has the logic to follow the fields inside the packet. This method of packet inspection is more difficult than shallow packet inspection and much slower.

Modern IDSs typically deploy a combination of shallow and deep packet inspection. Most of the analysis in signature-based IDS engines is shallow with the ability to inspect beyond packet headers for string or payload content matching. In contrast, anomaly analysis and protocol analysis IDSs perform deep packet analysis by definition. Because it is, it must process the input data just as the intended application recipient would to identify anomalies or exception behavior.

Technet24



Another important topic in understanding IDS systems is the concept of data normalization. An IDS uses data normalization to take data and baseline it before analysis. To illustrate this topic, let's offer an example of why data normalization is important.

Attackers are certainly wary of IDS systems—the use of an IDS system and quick reaction from an incident response limits the attacker's ability to compromise valuable information and resources. An attacker tries to elude detection by the IDS using obfuscation techniques called "IDS evasion." By changing the characteristics of the traffic sent to exploit a particular vulnerability, the attacker manages to change what the IDS sees for analysis but can still exploit his intended target. For example, an attacker wanting to exploit a vulnerable Microsoft SQL Server/IIS with the xp_cmdshell vulnerability might request the following from the target site in order to add a user account with the name "hacker":

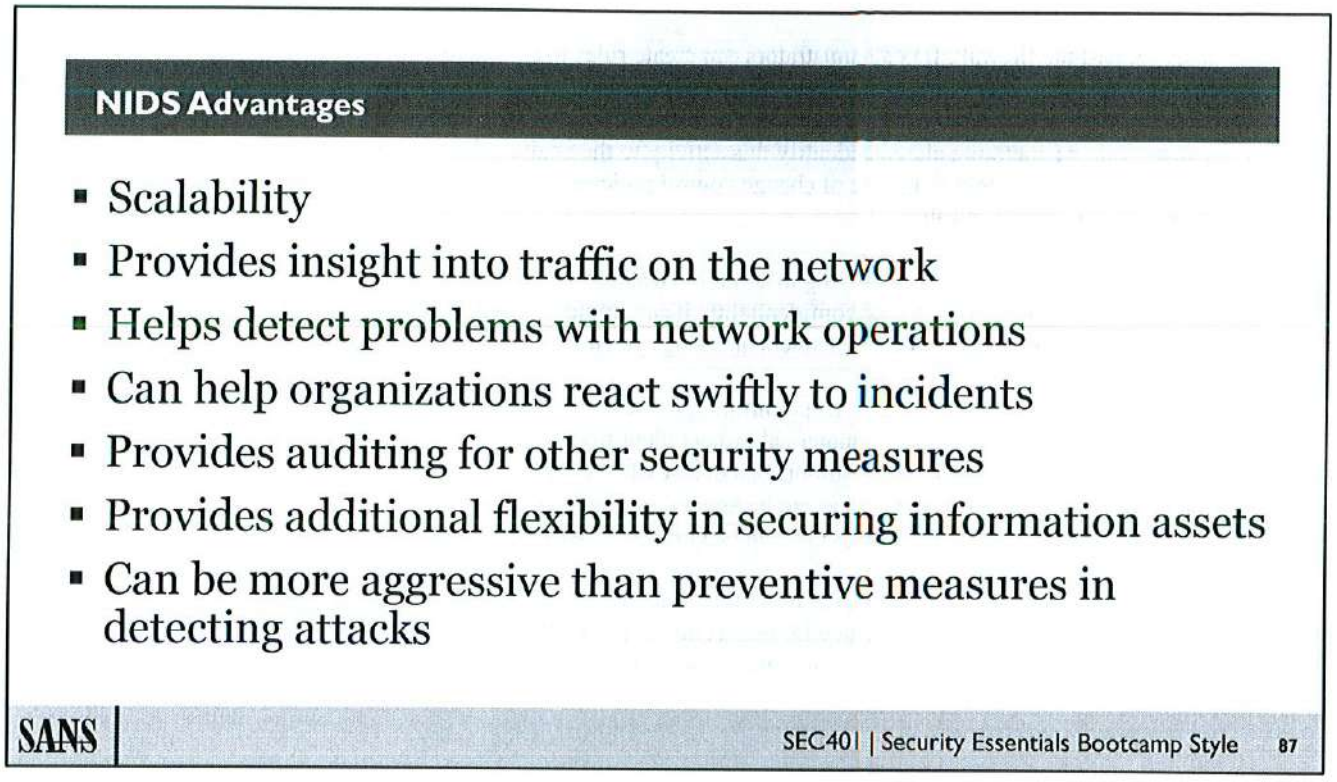
GET /scripts?0';EXEC+master..xp_cmdshell(cmd.exe+/c+net+user+hacker+password+/add);--

A NIDS would likely catch this attempt to exploit the vulnerable system, either by identifying an attempt to EXEC the stored xp_cmdshell procedure, through the cmd.exe string or through the use of net user.../add. The NIDS has several opportunities to identify the attack and generate an alert to the analyst.

Reference

1) Finding Web Vulnerabilities with Whisker - http://www.slideshare.net/Mike97/51710-kai2004-insa

© 2017 Secure Anchor Consulting



We've identified several features of NIDSs and how they operate. Clearly, NIDS tools offer many advantages to an organization in detecting events of interest on the network.

When asked, most people are unable to identify the nature of traffic on their networks. Many people speculate, but few implement mechanisms to monitor and analyze the consensus of traffic. Many IDS tools also provide summary reports that document the amount of bandwidth utilized and break the information down by protocol and application type. This information can be used to identify general misuse of the network (for instance, a high percentage of traffic being attributed to peer-to-peer applications) and for planning purposes when expansion is required to accommodate new facilities or new applications.

Without detailed insight into how network traffic functions, most administrators are unaware of misconfigured equipment that can affect performance and even the security of applications. Most network administrators are concerned about "Does it work?" instead of "Does it work correctly?" With the data generated by a NIDS, analysts can quickly identify patterns of activity on the network that are inefficient and not what the administrators intended when designing the network.

Conversely, the IDS can also act as an auditing mechanism to ensure that applications are operating as designed.

Of course, the primary intention for a NIDS is to detect hostile activity on the network and generate alerts for administrators to take action. By identifying probes and reconnaissance techniques from attackers, the IDS helps organizations better understand their risks and design appropriate system and network countermeasures. By identifying systems that have been exploited and compromised, the IDS helps organizations quickly respond to recover these systems to minimize downtime and loss.

Additional advantages of deploying NIDSs that are somewhat intangible are included here.

Many organizations rely on firewalls for their first line of defense against attacks, but few regularly review and audit firewall rules and log files. A NIDS tool can assist in the auditing process by identifying the traffic

Technet24

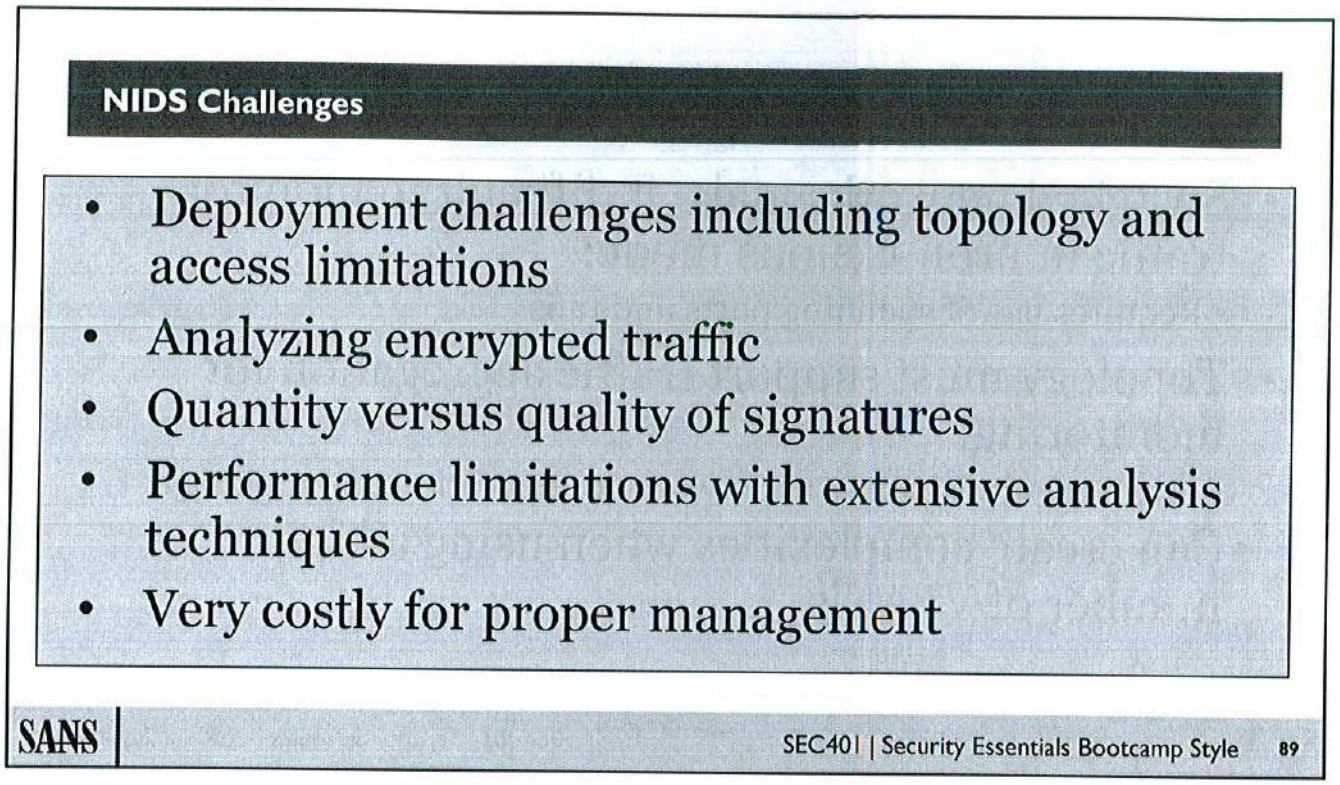
that does get past the firewall. IDS administrators can create rules to reflect current firewall policies for systems or networks, generating alerts for any traffic that does not match the approved rules. Should an administrator or attacker modify the firewall rules to allow different traffic to sensitive systems, the IDS immediately starts generating alerts to identify this activity to the analyst. This use of a NIDS allows organizations to clearly identify misuse of change control policies that should be in place to govern what access it allows past the firewall.

Use of a NIDS can contribute to identifying information leakage of sensitive information. If an organization has a project that requires secrecy and confidentiality, it can implement a honeytoken approach to identifying information that leaves its network, or outsiders querying systems for the protected information.

The approach of using a honeytoken to secure information involves labeling information with keywords that are unique, such as a project name or numerical project identifier. By configuring the NIDS to alert whenever this unique information is seen, administrators can detect when documents or e-mail messages traverse the network in unapproved mechanisms. This can be applied to word-processing documents or spreadsheets, project plans, and any other data sources that can be configured with specific keywords.

Organizations can extend this functionality of an IDS to identify sources of information leaks by planting documents that contain honeytoken data with fake content. Should the NIDS identify the honeytoken content, the analyst can use the source address information to identify the person who sent the document without authorization.

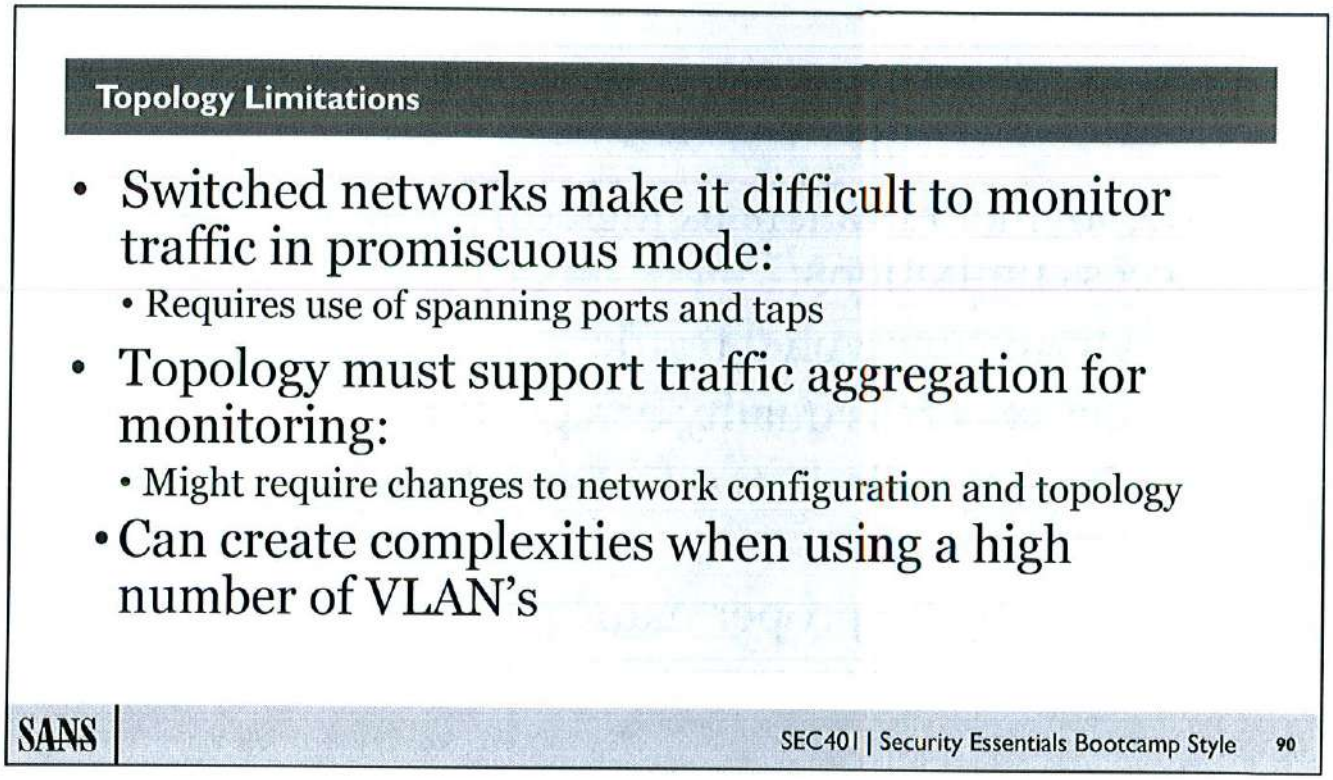
© 2017 Secure Anchor Consulting



In the next several slides, we look at some of the challenges in deploying NIDS. These include limitations in the design and configuration of topology to support NIDS, challenges with the analysis of encrypted traffic, the quality of signatures and detection techniques, performance limitations, and the overall cost in NIDS management.

© 2017 Secure Anchor Consulting

Technet24



To capture traffic for analysis, NIDS operate with network cards in promiscuous mode, a configuration that lets the card capture all the traffic that happens on the network, not just the traffic that is addressed to the NIDS station itself. This method of capturing traffic was perfectly legitimate when we were working with shared-hub segments in our networks, but it became much more challenging with the rapid adoption of switched networks.

In a switched network, the switch maintains an internal table of potential destinations that it understands and sends traffic to only those ports that are the intended recipients of the traffic. This poses a logical problem for the NIDS tool because they want to receive all traffic, despite whether it is addressed to the NIDS device or any other device on the network. To overcome this challenge, NIDS tools make use of spanning ports and network taps.

Most switches have a spanning port, which receives all traffic transmitted on the switch. The spanning port is generally used for debugging and other administrative tasks. A NIDS sensor can be placed on a spanning port to sample more traffic, but it often does not work very well in practice. There are many problems with spanning ports as a solution to support network-based IDSs.

Spanning can also adversely affect switch performance by dramatically increasing the traffic on the backplane because the normal port-to-port switched circuit traffic now needs to be replicated to the spanning port.

Another problem with using a spanning port is that frequent network changes can often disrupt spanning port settings. A network engineer who is unaware of the spanning port's purpose or the presence of the NIDS sensor can easily botch this configuration. Unfortunately, you won't notice the problem until you realize the sensor is not reporting any detects. Many current NIDS have this problem—unless connectivity to the management station is lost, they don't report a marked decrease in traffic as an error condition; they merely fail silently.

Switch vendors are more aware of the requirements of intrusion detection and in some cases are building network-based intrusion detection capabilities into the switch itself.

© 2017 Secure Anchor Consulting

Although it adds cost, another way to capture traffic from switched networks is to add network taps, hardware devices that hook directly into the coax, twisted pair, or fiber optic network cable. A tap sends a copy of the traffic that passes through it to one or more other networked devices. Because they have no impact on the performance of the switch and don't need to be reconfigured with every network change, taps can be an excellent way to instrument a network for network-based intrusion detection and are generally preferred if there is enough budget to support their use.

Although taps are easier to manage than spanning ports, they do have some issues of their own. To capture the network activity for all of your protected hosts, you need a tap for each one of them. This can get cumbersome, even if your network is not huge. If you try to reduce the number of taps required by moving them further upstream, you might run into some of the same bandwidth problems discussed previously.

© 2017 Secure Anchor Consulting

Technet24

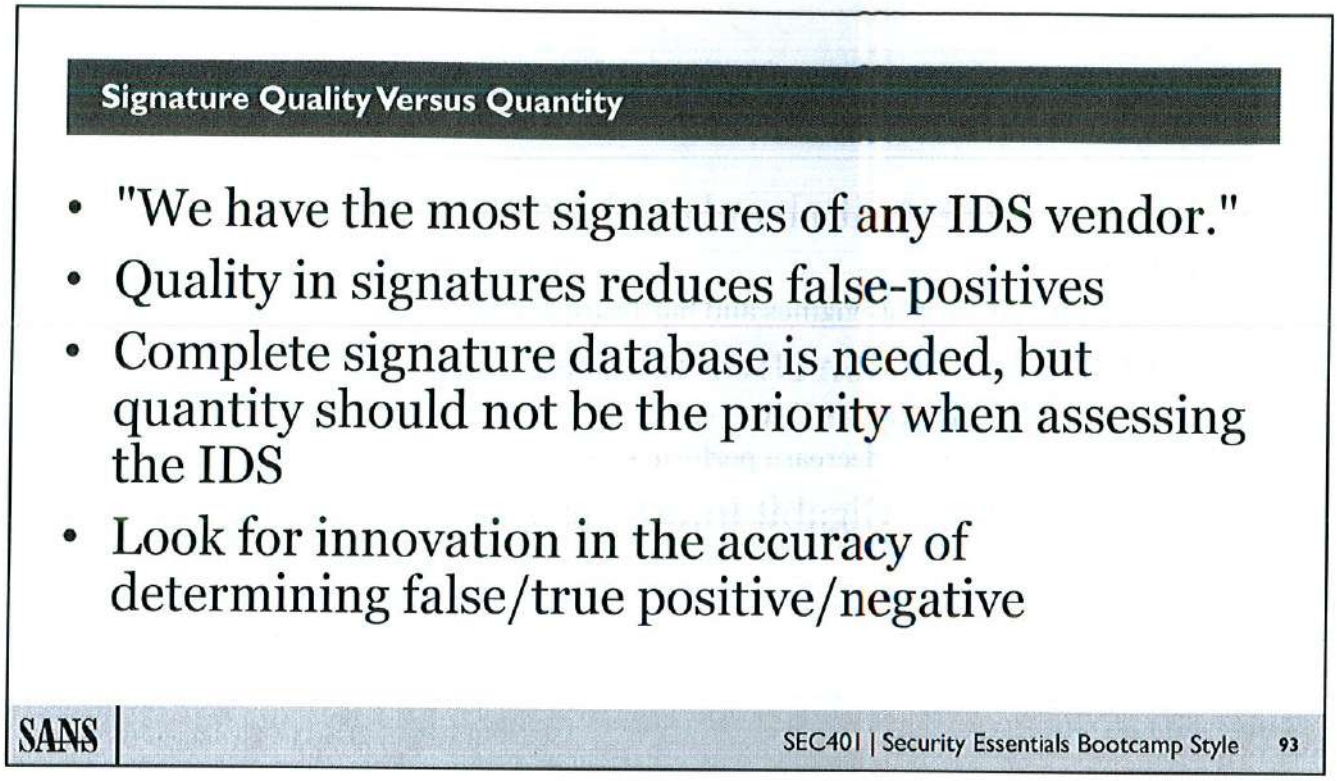
Analyzing Encrypted Traffic

- Encrypted traffic precludes many signature-based detection methods
- Decrypt traffic when possible
  - Will increase overhead on NIDS
- Use anomaly analysis on encrypted traffic
  - Will increase false-negatives
- Seek alternative IDS technology (HIDS)

We discussed the difficulties caused by the switched nature of networking hardware. Another barrier to network-based intrusion detection can be the traffic itself. If NIDS sensors cannot interpret the traffic it receives, it cannot analyze it. This problem only gets bigger as cryptographically secure protocols increase in popularity. Encryption of network traffic is definitely a worthwhile countermeasure, but it comes at the price of weaker network-based intrusion detection abilities. Attackers have caught on to this NIDS weakness and, therefore, often use encryption to hide their malicious actions.

Virtual private networks (VPNs) and other encrypted channels can make network-based intrusion detection nearly impossible. When cryptography is used heavily on the network, one useful place to put a sensor is on a host at either end of the encrypted channel, capturing the traffic before it is encrypted or after it is decrypted.

© 2017 Secure Anchor Consulting



A common mistake in evaluating IDS tools is using the number of rules or signatures as a guide for selecting a vendor's product. In the past, many vendors claimed to have more signatures than any other IDS vendor in an effort to convince consumers that their product is better than the competition. Although this practice has largely stopped, it is valuable to identify the issue of signature quality versus signature quantity.

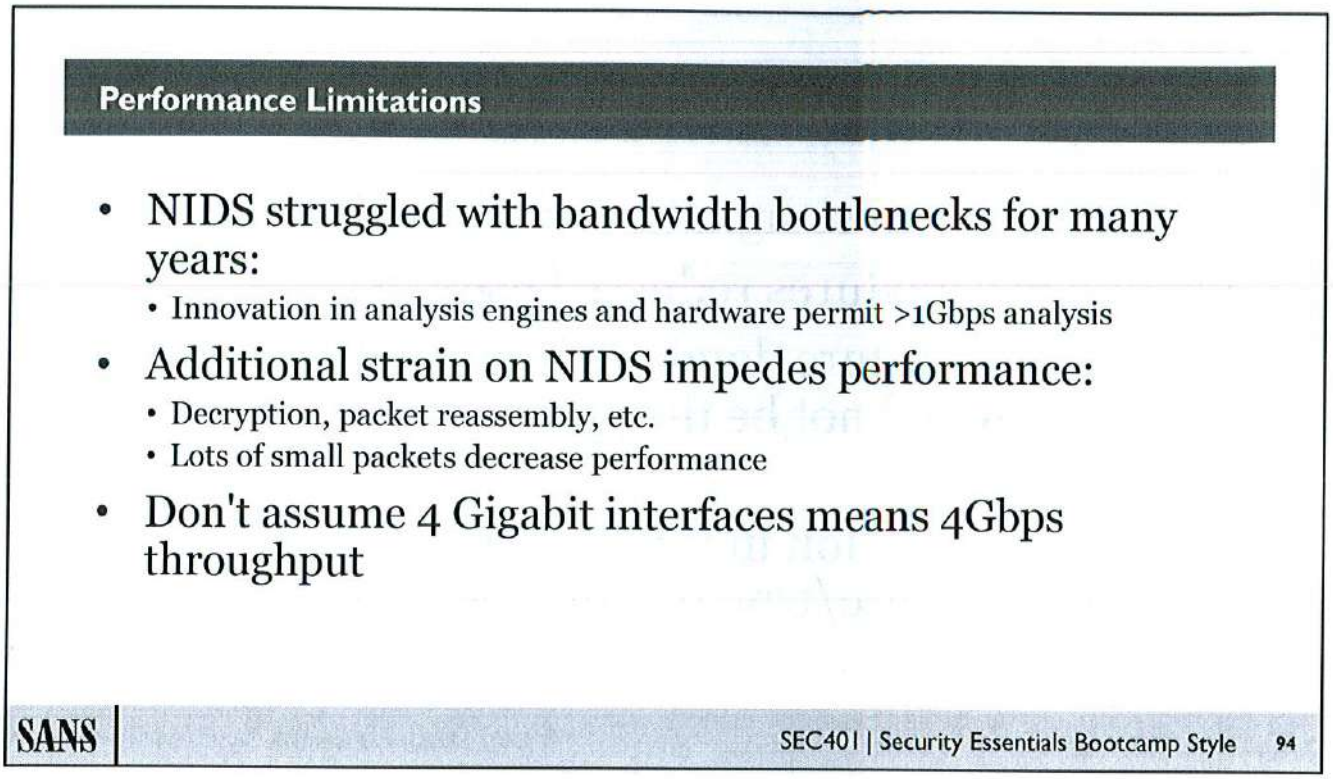
Every IDS needs to have a thorough database of signatures to be able to identify events of interest on the network. Supporting this database should be thorough testing and real traffic analysis of the signatures so they become fine-tuned—detecting events of interest reliably, without generating false-positives. How a vendor implements new rules into the IDS is important. Be sure to ask the IDS vendor about its testing procedure before adding new rules to its product.

Imagine a scenario where a new rule is added to the IDS through an auto-update procedure that generates hundreds of thousands of false-positive detects overnight. When an analyst comes in the next day, they will be forced to sort through all the alerts from the IDS to differentiate false-positive from true-positive detects. Or they might simply delete all the events and miss the actual hostile activity on the network.

Look for innovation from your vendors in how they are reducing the number of false-positive detects and increasing the true-positive alert percentage, and don't forget about identifying the cases of false-negatives. Make sure your vendor clearly understands these issues and can explain what its product can offer you.

© 2017 Secure Anchor Consulting

Technet24



High bandwidth requirements have been a major challenge for NIDS vendors. Historically, NIDS devices have been unable to keep up with the increasing throughput of networks while maintaining the thorough analysis techniques that are needed to identify some of the more advanced techniques used to compromise systems. Although there have certainly been significant improvements in the ability for NIDS tools to monitor high-bandwidth networks, the phrase "gigabit IDS" can be a little deceptive. Let's take a more detailed look at the actual achievements and limitations of this technology.

Although hardware and software limitations contribute to the restrictions for NIDS on high-speed networks, the most significant limitation is the NIDS processing capacity is in terms of packets per second. Many NIDS claim to be able to process traffic at speeds greater than 1Gbps, but this statistic can be misleading. NIDS systems that perform shallow packet inspection can likely inspect more than 1Gbps of traffic if the traffic consists mostly of large packets. When a NIDS is asked to inspect traffic that consists mostly of small packet sizes, it requires significantly more resources, which many IDS tools are unable to supply.

Another limitation on the NIDS is the ability to perform deep packet inspection. On standard Ethernet networks, packet size ranges from 60 to 1518 bytes in length. On Gigabit Ethernet networks, packet size ranges from 512 to 9216 bytes in length. Imagine the processing required to perform deep packet inspection on 9 K frames!

Additional processing burden on the IDS includes decryption of some traffic for analysis, packet and stream reassembly, data normalization, and other tasks that all contribute to the demand for resources. Combined, the feature-rich IDS that can detect every known attack on the planet will have to contend with processing requirements for these features versus the ability to perform limited inspection on high-throughput networks. Vendors must strike a balance between features and performance that doesn't leave opportunity for false-negatives on the IDS.

What happens when your IDS is overloaded with more throughput than it can handle? Once a NIDS bandwidth limit is exceeded, its performance tends to degrade rapidly, not just discarding excess packets, but thrashing from resource exhaustion. This means traffic is able to get past the sensor that would potentially be a

© 2017 Secure Anchor Consulting

an attacker trying to evade detection. Statistical sampling analyzing a manageable subset of the traffic is a desirable alternative to complete failure when there's too much traffic, but you're still not gaining access to all evidentiary traffic. To reduce an attack's chances of being detected, an attacker wanting to get past your NIDS would just have to flood your network while performing the exploit.

A response to the bandwidth limits of network sensors is to move the sensors downstream toward the leaf nodes of your network, increasing the number of sensors while decreasing the bandwidth required per sensor. By moving the NIDS device downstream, the tool has less aggregate bandwidth to process and will be less likely to become overloaded with requests for resources. Unfortunately, this introduces additional cost to the organization that must now invest in multiple sensors, and likely a management console to collect all alerts from all the NIDS throughout the organization.

© 2017 Secure Anchor Consulting

Technet24

NIDS Cost

- "I have $$$$ in the budget; I'm going to buy a NIDS."
- The real cost is in deployment and maintenance:
  - Always calculate total cost of ownership (TCO)
- Require trained administrators to analyze alerts
- Consider the disk space requirements for a Gigabit IDS and event correlation

The reality behind deploying an IDS is that the capital cost is a small fraction of the overall cost to an organization that uses its IDS

It is common for an organization to look at its budget and decide to use remaining capital dollars to invest in an IDS system. The shiny-new IDS gets delivered to the organization, maybe it gets set up and starts collecting data and generating events. What happens from there is dependent on the support for intrusion detection within the organization, and what the priorities are for the organization in terms of security.

In many organizations, the IDS just sits there. The reality behind deploying an IDS is that the capital cost is a small fraction of the overall cost to an organization that uses its IDS. What organizations quickly realize is that monitoring the IDS is more than a single person's full-time job. Managing updates to the IDS, customizing and adjusting IDS rules to reflect the traffic patterns in your organization, and maintaining and purging data sets are just some of the regular activities for managing the IDS. Still, these activities reflect a diminutive amount of effort compared to what an organization must do once it starts receiving alerts and detecting attack activity on the network including incident handling and response and designing network and system countermeasures to defend against the now-evident attacks.

To get the most benefit from an IDS system, organizations must have analysts that are well-trained in the necessary skills of intrusion analysis and incident handling. Without these skills, analysts will have a difficult time differentiating between false-positives and true-positives and will be unable to take the corrective actions necessary when responding to incidents.

The bottom line is that an IDS is much more expensive than just the capital acquisition costs. If an organization is serious about using an IDS to identify malicious activity on the network, it needs to be prepared for much greater costs in human resources, training, and associated equipment costs.

© 2017 Secure Anchor Consulting

Snort as a NIDS

The student will have a high level of understanding of network intrusion detection concepts and techniques and how Snort performs network intrusion detection

This page intentionally left blank.

© 2017 Secure Anchor Consulting

Snort as a NIDS

- Open-source tool, low-cost (free software, inexpensive hardware)
- Suitable for monitoring multiple sites/sensors
- Efficient detect system
- Low effort for reporting

Snort is billed as a lightweight network-based intrusion detection system. It was introduced to the open-source community in 1998 by its developer, Marty Roesch. Snort has quickly gained a reputation for being an extremely efficient, lightweight, and low-cost NIDS solution and owes its popularity and extensive features to a devoted team of core developers and an active user base.

Snort's design allows for easy integration into most networks, and it can be configured to monitor multiple sites, networks, or interfaces with relative ease. It has rules for packet content decodes and packet headers. This means that it can detect data-driven attacks, like buffer overflow errors and attacks on vulnerable URLs and scripts.

Because Snort is open-source and has such an active user community, it is an ideal system to learn how to analyze intrusions and experiment with different configurations.

© 2017 Secure Anchor Consulting

Snort Rule Flexibility

- One of the most significant advantages of Snort is the rule language.
- Administrators can create custom rules to detect any type of pattern match.
- Rules for new worms, exploits, or vulnerabilities are quick to be published by the user community.
- Rules can be developed to support honeytokens or any custom requirements.

Snort offers a powerful rules language for identifying events of interest on the network. Having the ability to develop rules based on almost any imaginable criteria for network traffic is a significant attractor for using Snort and one that few competitors are able to emulate with a high degree of success. Using Snort, an analyst can create rules to identify any measure of traffic on the network that can be uniquely characterized as an event of interest. Alerts are then classified, prioritized, and reported for analysis.

In practice, the Snort user community is consistently the first to develop new rules to detect the latest exploits, viruses, worm activity, or other attack techniques. This is advantageous to organizations deploying Snort as a NIDS tool because they can quickly identify new threats against systems. Although rule authors strive to ensure that the rules are well-tested and do not generate false positive detects, they do not undergo the same scrutiny as a commercial vendor would apply to its tools. There is a trade-off between fast rule development (higher potential for false positives or false negatives) and slow and methodical rule development (misses new attacks).

Snort rules are also well-suited for custom IDS techniques, such as the use of honeytokens, because the analyst can create rules to detect any kind of activity on the network. The next few slides examine the power of the Snort rules language in more detail by examining the contents of rules and a few specific examples.

© 2017 Secure Anchor Consulting

Technet24

Writing Snort Rules

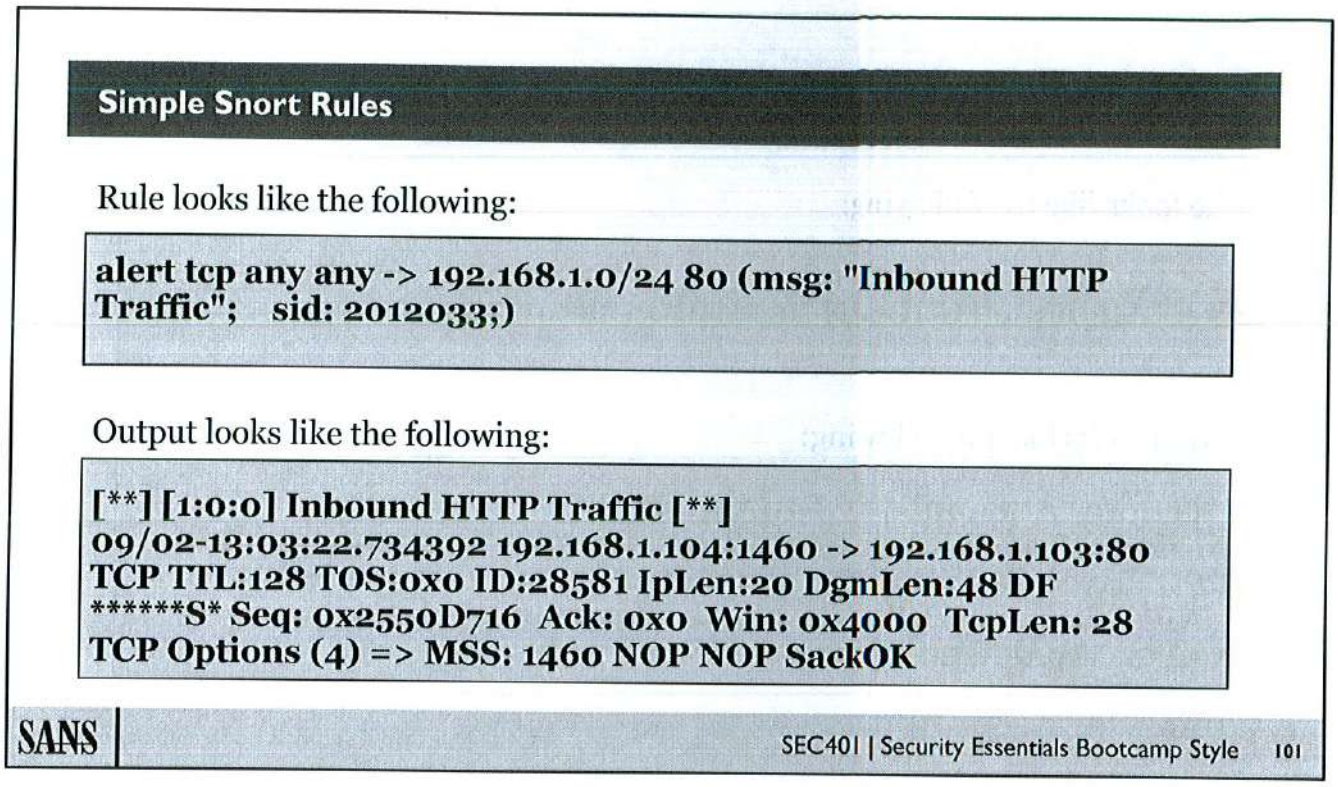
- Can create custom rules to filter on specific content.
- Pre-loaded with hundreds of rules (but you might need to create one or more custom rules).
- Simple to write yet powerful enough to capture most types of traffic.
- Options:
  - Basic (Pass, Log, Alert)
  - Advanced (Activate, Dynamic)

As discussed, Snort provides the capability to create custom rules to alert on specified content. The compiled source code provides hundreds of pre-written rules. However, there might be times when you need to create rules that are not included by default. Given the fast-paced world of intrusion detection and that new threats are released on a daily basis, the ability to quickly write custom rules can often make or break your career as an information security professional!

Snort rules are simple to write, yet powerful enough to capture most types of traffic. There are five options to keep in mind when writing rules:

- **Pass**: Ignore the packets and take no action.
- **Log**: Allows you to log the particular action to the location you specified in your snort configuration file (for example, snort.conf).
- **Alert**: Allows you to send alerts to a central syslog server, popup windows via SMB, or writing the file to a separate alert file. This alert file is commonly used with tools like Swatch (Simple Watcher) to alert the analyst to signs of intrusion or electronic tampering. Once the alert is sent, the packet is logged.
- **Activate**: Specifies that Snort is to send the alert and then activate another dynamic rule. For example, Snort can be configured to dynamically block ports based on various attack signatures, but this should be considered an advanced usage and extreme caution should be exercised when using this option.
- **Dynamic**: This rule remains idle until activated by another rule. Again, this is an advanced feature and should be used only by experienced intrusion detection professionals.

© 2017 Secure Anchor Consulting



This example is a simple rule, but it clearly illustrates the basics of creating custom Snort rules. Remember that you probably would not want to run a rule of this type on a production network with web servers unless you have a lot of disk space! As we can see, we told Snort to alert us on any traffic destined for port 80 (http) on the 192.168.1.0 network.

On the slide, you see a rule and below it an alert that was generated when the rule was matched. On this slide, the alert begins with [**] [1:0:0] Inbound HTTP Traffic [**] and that string was created by the message option in the rule shown on the slide. There are many potential options; they must all be separated by a semicolon. In the rule on the slide, there is only one option.

Now, we have an idea of what content is needed to create a rule, so let's look at the output of an event triggered by this rule. There is a lot of data logged by this rule, but it is all relevant information. We can see the message parameter followed by the date/timestamp, source IP address, and destination IP address.

In addition to basic source and destination information, we are presented with a detailed listing of TCP information to include which flags are set, window size, and options. This information can seem overwhelming, but having the added fidelity that Snort can provide is a powerful way to examine the techniques of hackers, and to sort false positives from true positives.

© 2017 Secure Anchor Consulting

Technet24

Advanced Snort Rules

Rule looks like the following:

```
alert tcp any any -> 192.168.1.0/24 80 (content: "/cgi-bin/test.cgi";
msg: "Attempted CGI-BIN Access!!"; sid: 2012033;)
```

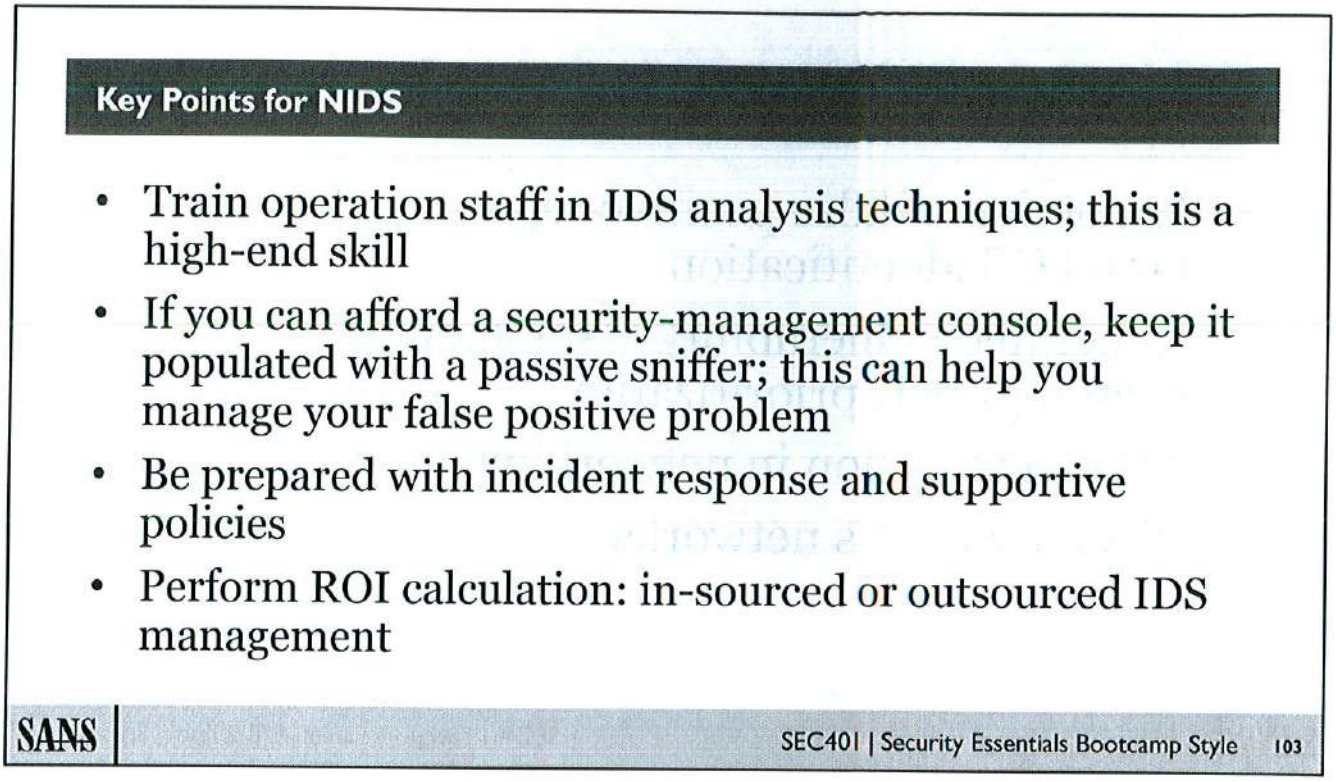
Output looks like the following:

```
[**] [1:0:0] Attempted CGI-BIN Access!! [**]
09/02-13:18:30.550445 192.168.1.104:1472 -> 192.168.1.103:80
TCP TTL:128 TOS:0x0 ID:29951 IpLen:20 DgmLen:466 DF
***AP*** Seq: 0x32D8E9C1  Ack: 0xB427699E  Win:
0x4470  TcpLen: 20
```


As you can see in this rule, we added a parameter called the "content field." We tell Snort to look for any signs of access to a file called "test.cgi" residing in the cgi-bin directory of a web server. If this type of access is detected, Snort sends an event-notification alert and logs the entire packet.

As stated previously, Snort allows for the creation of just about any type of rule imaginable.

© 2017 Secure Anchor Consulting



Whether you are considering the deployment of NIDS or are looking to leverage an existing deployment that is underutilized, follow these recommendations to make the most of your investment.

A well-trained analyst can configure the IDS to your network environment and respond to security alerts on the network. An analyst who is not sufficiently trained becomes overwhelmed with the amount of data being generated by the IDS and is unable to differentiate true positives from false positives.

A management console is used to combine the data from multiple sources of IDS and simplifies the job of data correlation from multiple sensors. A security information event management station (SIEM) allows organizations to combine data from multiple discreet sources, including IDS alerts, syslog messages, Windows event log data, and more. Having a single point of reference available helps the analyst understand the overall impact on a network and allows a more detailed and thorough analysis of the risks that are present.

At some point, you need to exercise your incident-response team to react to a large-scale event. A NIDS helps provide the necessary information to the incident-response team, who must have clear direction on the policies for computing in the organization. For example, if your NIDS detects that a home user is attacking internal hosts over VPN, does your organization have a policy on how the incident-response team can analyze the home PC to determine whether the activity was at the hands of the owner or some other malicious source?

Carefully examine the offerings from managed security companies. Although the initial costs might seem staggering for managed IDS services, consider the long-term costs to your organization to deploy an internally managed IDS (when done correctly) versus the potential losses from compromised systems that go undetected. Remember that outsourcing IDS is only part of the cost—organizations still need to invest in training for their analysts who receive calls from the monitoring company and must be prepared in the skills of incident handling.

© 2017 Secure Anchor Consulting

Technet24

Developments in NIDS

- Reduction of false positive reporting through target OS identification
- Integrated vulnerability assessment for threat profiling/alert prioritization
- NIDS integration in networking devices
- IDS for wireless networks

This slide covers some of the recent advances in NIDS technology and how they impact organizations that are planning to or have deployed NIDS systems.

We mentioned the problem of false positives in NIDS systems several times in this module, as have many organizations to their IDS vendors. In an effort to reduce the problem of false positive detects, vendors have started to implement passive OS fingerprinting in their products to eliminate alerts that trigger from rules but are not applicable to the target OS.

Passive fingerprinting is a technique that monitors network traffic and characterizes the nature of traffic to identify the host operating system. Characteristics such as the TCP window size, the IP header TTL counter, TCP flag combinations and order, and other criteria can be used to accurately identify the operating system from simply examining network traffic. Armed with the knowledge of the operating system for a remote host, the IDS vendor can classify an alert as applicable or inapplicable to the target host. For example, an organization might not want to generate alerts that exploit Windows flaws when it is targeted against a UNIX host. This way, the IDS has more intelligence about the network configuration and hosts on the network without additional configuration by the administrator to uniquely identify all the hosts on the network.

Another step to reduce the number of alerts from an IDS is to generate an alert only when an attacker attempts to exploit a system that is vulnerable to the exploit in use. This way, the IDS can be configured to alert only for an attempt to exploit a system that is indeed vulnerable and has a high likeliness of being compromised. This technique requires the IDS to maintain an inventory of known vulnerabilities for hosts on the network to be effective, which is where the products start to differentiate in their ability to characterize vulnerabilities. The early products supporting this new feature are classified as using passive or active vulnerability analysis.

Active vulnerability analysis uses traditional vulnerability-assessment tools to routinely scan systems to document their vulnerable services. This provides a complete assessment for the IDS to use in correlation to identified attacks against systems, but can be difficult for organizations to implement. Many organizations might want to avoid active vulnerability assessment on systems that are in production because the scans might

© 2017 Secure Anchor Consulting

disrupt legitimate services. In addition, the IDS is often unable to assess systems that are vulnerable to Denial of Service attacks because testing for the attack on a vulnerable system exploits the vulnerability and stops production services.

Passive vulnerability analysis uses a new method to identify vulnerable services on a host by passively monitoring traffic. The IDS attempts to identify the characteristics of an application that are exhibited only when the application is susceptible to a specific vulnerability. After this condition is identified, the IDS marks the resource as vulnerable and adds the information to its vulnerability database. This method is often incomplete in assessing vulnerabilities because it is not always possible to identify characteristics that uniquely identify a vulnerable application. Still, significant research is being devoted to this technology that will help build the IDS's knowledge of the vulnerabilities on the network without the adverse side effects of active vulnerability analysis.

We identified some of the challenges in deploying NIDS tools in modern networks, including the difficulties in monitoring switched networks and challenges in monitoring fast network segments. Network equipment manufacturers have developed modular functionality for chassis equipment to perform analysis of network traffic from the backplane of the switch. In this configuration, the "IDS blade" can simply insert into the switch a custom hardware that inserts into the chassis and monitors traffic as it traverses the switch, despite the port or interface that the traffic arrives on. Using custom hardware in this fashion also lets the IDS analyze traffic at fast speeds, as well as use custom "hardware buffering" that lets the switch hardware buffer packets for later delivery to the IDS when the processor is too high.

There are significant flaws in the operation of wireless networks that threaten the security of perimeter defenses. To combat this threat, vendors have developed wireless IDS tools that focus their monitoring activity on wireless-specific properties of traffic, leaving traditional IDS tools to analyze the upper-layer protocols such as IP. These wireless IDS tools (called "WIDS") are deployed to co-exist with wireless LAN deployments, or instead of wireless LAN deployments if an organization wants to identify "rogue" wireless LAN activity in unauthorized locations. Using specialized signature analysis and anomaly analysis techniques, these tools are able to detect hacker techniques that may be used to bypass intended wireless security mechanisms.

© 2017 Secure Anchor Consulting

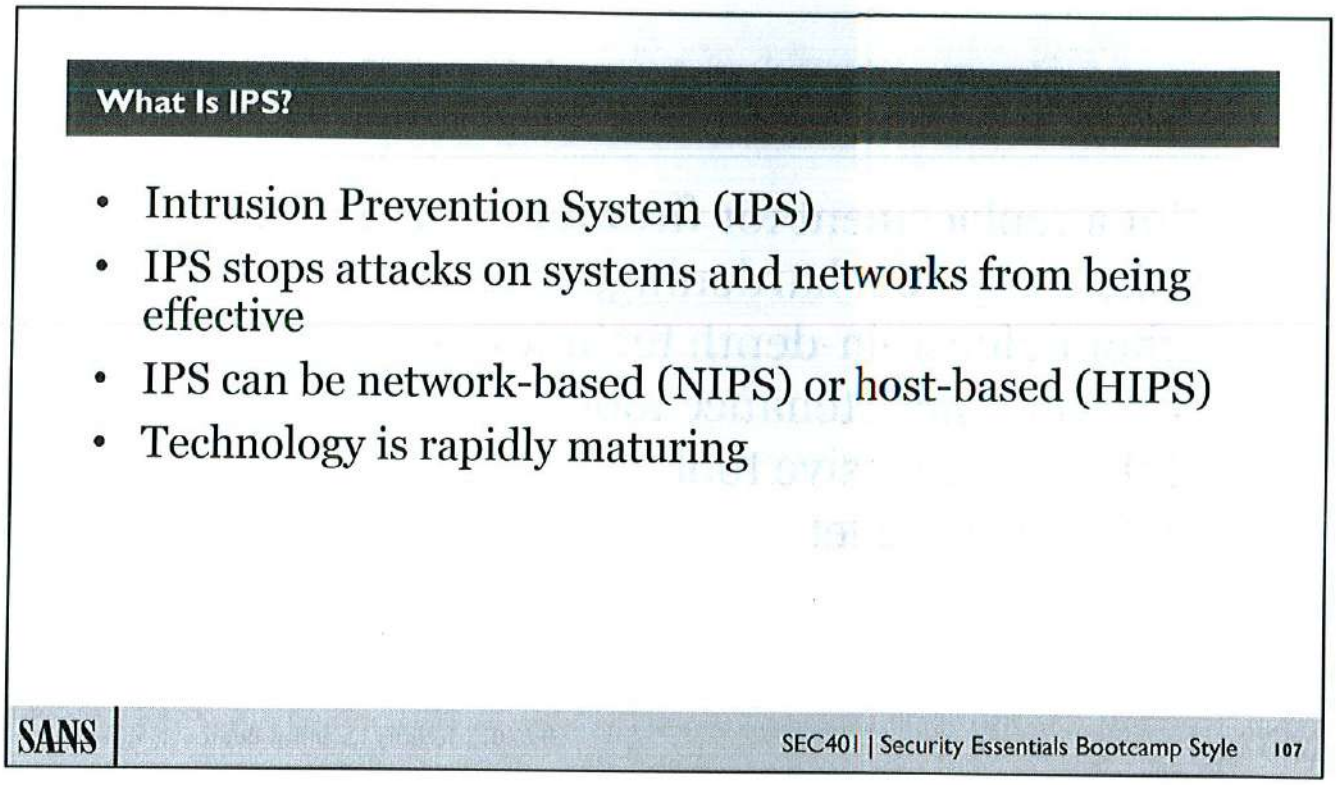
Technet24

NIPS Overview

The student will understand the technologies and techniques behind NIPS and how it can be applied

This page intentionally left blank.

© 2017 Secure Anchor Consulting



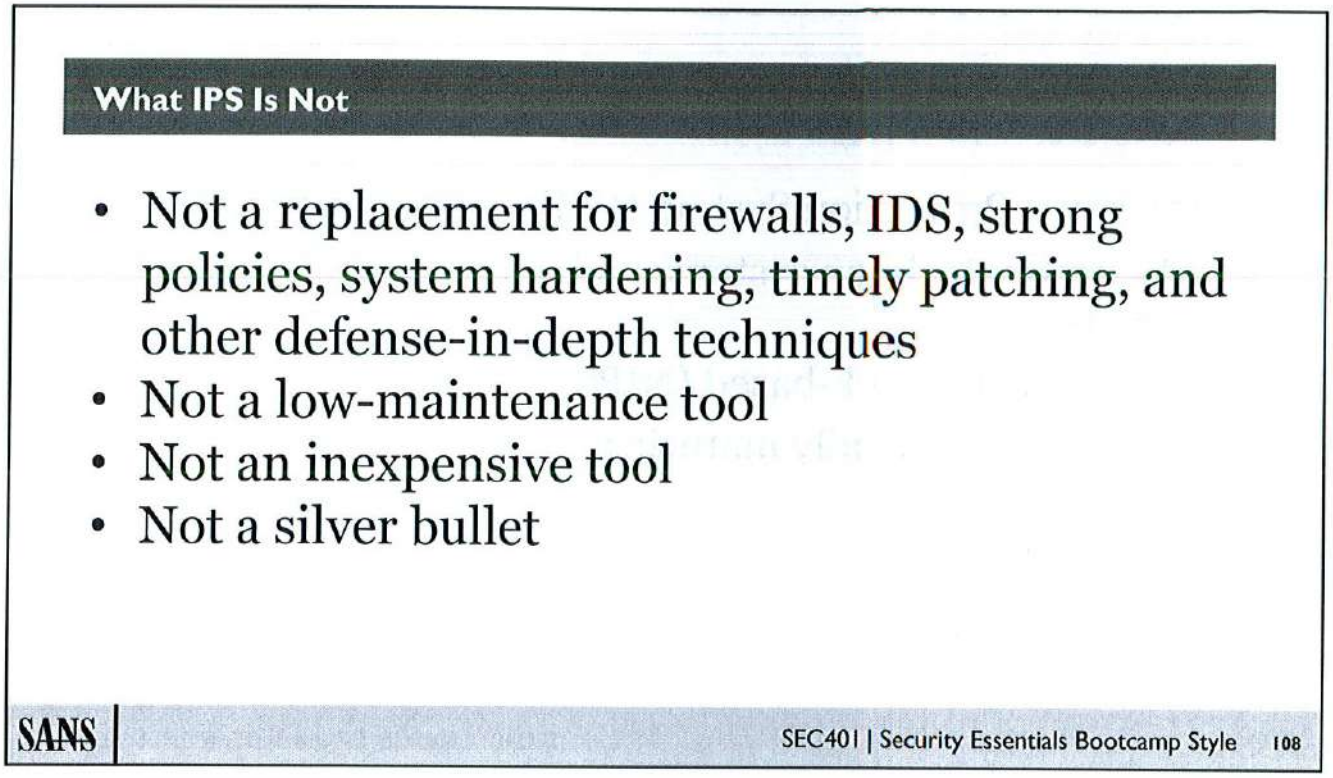
Simply stated, intrusion-prevention technology adds another layer of defensive measure to protect resources. Unlike intrusion-detection technology that reports only attacks against monitored systems and firewalls, which are utilized to permit or deny traffic based on source/destination IP addresses and ports, intrusion-prevention technology attempts to stop attacks before they succeed. How the IPS detects and stops the attack varies significantly between vendors, although each carries the same moniker of "intrusion prevention." In the past, IDS vendors who traditionally had a strong product offering (active IDS) decided to incorporate the feature of sending a TCP reset to a culprit host that was attempting to send malicious traffic to another host on a network. This ability was the industry's first attempt at intrusion prevention. Most vendors today use more advanced methods to prevent attacks. An active IDS stops an attack in progress, but a true IPS stops an attack automatically. In addition, an IDS is deployed passively, whereas an IPS is traditionally deployed inline.

Intrusion Prevention Systems (IPSs) can be generally classified into two vectors: network-based intrusion prevention (NIPS) and host-based intrusion prevention (HIPS). As their names indicate, NIPS products work at the network level, analyzing traffic much like a NIDS. HIPS products are installed on individual hosts and stop attacks at the operating system or application level.

Technology for IPS is making significant strides in reducing false positives, reducing the overall impact on network and server/host resources, and stopping unknown attacks against targets. Many organizations that have adopted IPS technology are able to mitigate the effects of different types of attacks against vulnerable systems, including attacks from hackers, worms, viruses, and other malware.

© 2017 Secure Anchor Consulting

Technet24



Unfortunately, IPSs won't solve all the security threats we face, but no technology does, and thus the reason for defense-in-depth strategies. Deploying an IPS is not a replacement for patch management and system hardening, but it does provide a valuable asset: time. Organizations using IPSs are often able to extend the amount of time they have to deploy patches to resolve operating system and application flaws, potentially delaying the deployment of fixes until several patches have accumulated and a window for scheduled maintenance of equipment is available. Still, organizations should rely on defense-in-depth instead of just an intrusion-prevention product to secure enterprise resources.

Although many vendors have "out-of-the-box" recommended system settings that will begin protecting organizations, systems upon deployment, like IDS and IPS, are not set-and-forget technologies. They require significant maintenance and monitoring to be effective defense tools. IPS is also not an inexpensive tool for enterprise-wide deployment.

© 2017 Secure Anchor Consulting

How NIPS Work

- NIPS are typically deployed at the perimeter in front and/or behind a firewall
- Deploying a NIPS between the firewall and ISP router ensures that the firewall and DMZ servers are protected
- Behind-the-firewall NIPS deployments protect the internal network from remote-access VPN users, but can also assist in locating infected internal hosts

Let's discuss how NIPS actually works. Traffic originates from the Internet and passes through the NIPS to your corporate firewall and beyond if it does not generate any alerts. Traffic that generates an alert is dropped by the NIPS and never delivered inside your network.

Most organizations commonly implement NIPS technology at the perimeter in two or more areas. There are benefits to implementing IPS technology in different locations. The advantages of deploying an IPS system in front of the firewall is the overall protection it gives the firewall itself, as well as the DMZ systems. Denial of Service (DoS) attacks aimed at your firewall can also be mitigated using this method.

Many attacks today are successful because of the lack of security implemented on corporate laptops that are used by traveling or work-from-home employees. To access the corporate network, users typically utilize a remote-access VPN, which can provide a backdoor into a corporate network if the laptop was to become compromised. Placing a NIPS behind your corporate firewall assists in protecting your internal network from such hosts.

If an internal system became infected and the IPS was deployed only in front of the firewall, the IPS logs would only present you the NAT'd IP address of the firewall or another external IP address you're using to NAT internal systems. Implementing an IPS behind the firewall allows you to narrow down your search of infected internal systems.

© 2017 Secure Anchor Consulting

Technet24

NIPS Detail

- Deployed inline at network aggregation points
- Uses custom ASICS to support high-speed analysis with complex inspection
- Uses data normalization and reassembly techniques on aggregate traffic
- Hierarchical rule classification schemes are used to classify and identify traffic
- Because of risk for false positives, NIPS cannot identify as many attacks as NIDS

From a network perspective, NIPS operates like a switch connecting the internal and external segments of your network. Unlike a switch, the NIPS device uses a variety of techniques to stop attacks from entering and leaving the network. By using many of the techniques employed by advanced NIDS tools, the NIPS device can identify events on the network that are hostile. Because of its position, inline with the traffic of your network, the NIPS device can stop the hostile activity from ever being delivered to the target system.

In order for NIPS devices to be deployed as reliable, effective devices, they must overcome several challenges.

NIPS devices must utilize the same techniques of traditional NIDS tools to reduce the risk of false negatives, but cannot generate false positives on the network. They must also use many of the same evasion resistance techniques employed by NIDS to reduce the threat of attackers obfuscating data in an effort to bypass the NIPS. This is a significant challenge for the NIPS to overcome, and most vendors are simply unable to include a detection engine as thorough as a NIDS. To detect the greatest number of attacks without false positives, NIPS tools use passive OS fingerprinting and vulnerability.

Because the NIPS is inline with network traffic, it represents a single point of failure for the network. NIPS devices must be as stable as a firewall or switch to gain market acceptance. They must also be resistant to malformed traffic and cannot break existing network protocols. This is a similar risk to that of false positives by the NIPS—if a NIPS cannot properly interpret traffic or should fail in any way, it causes a failure on the network and denies legitimate requests. These failures can be accidental (as in the case of hardware or software failures) or intentionally performed by an attacker looking to DoS a network.

Most NIPS vendors have built redundancy into their products so that an organization would not have to purchase a series of devices to increase maximum uptime. For example, many NIPS vendors have multiple power supplies in their products, as well as the ability to fail-open if the NIPS hardware is malfunctioning. Others offer a Zero Power High Availability (ZPHA) device that will re-route network traffic should the IPS device lose power. This option is extremely beneficial for organizations that simply cannot afford downtime.

© 2017 Secure Anchor Consulting

NIPS devices must be able to keep up with the network traffic that is being generated. To be practical for use in monitoring enterprise or internal networks, the NIPS device must be able to handle Gigabit Ethernet speeds.

Despite the requirements to use extensive analysis techniques on network traffic to identify attacks, the NIPS must also provide low latency for network traffic. Additional latency on traffic that is analyzed should be in the low millisecond range.

The NIPS device must be secured against compromise because a compromised NIPS would give an attacker the ability to establish a man-in-the-middle attack against all the traffic entering or leaving your network. This is typically performed by configuring the NIPS without IP or MAC addresses on data interfaces, using a hardened operating system that resists common attacks, and a secured management interface that strictly defines who is permitted to connect to and administer the system. Attackers will seek opportunities to break NIPS, DoS a network, or to circumvent the protection it provides, so the NIPS device must be able to withstand any direct attacks.

To meet the processing demands of identifying malicious traffic while using data normalization and reassembly techniques on high-speed traffic with low latency, NIPS vendors typically make use of custom ASIC hardware to perform parallel processing. Using specialized ASICS (much in the same way network switches use specialized ASICS), NIPS devices can meet the demands of performance and scalability, at the cost of limited flexibility. Where traditional NIDS devices operate on host-based operating systems, such as UNIX, Linux, and Microsoft Windows, NIPS devices require significantly more processing capacity and throughput to meet the demands of processing with low network latency.

Although not specifically an innovative advancement through NIPS technology, many NIPS vendors are looking for ways to properly classify and identify malicious activity with fewer demands on system processing and memory capacity. One technique is the use of a rule classification scheme to quickly sort through traffic in order to rapidly identify malicious events. Some vendors have coined the term "multiresolution filtering" for this technique, where simple analysis tests are first applied to traffic. The simple tests represent a portion of the overall detection capacity of the NIPS device where a packet that matches a simple test is then processed using the more thorough tests.

For example, a NIPS device may require traffic to have data on the payload of the device for analysis. If this simple test fails (overall length—packet header length = 0), the NIPS device does not attempt to further classify this packet and sends it onto the network. This way, the NIPS can reserve its available system resources for more complex analysis.

After applying the simple rules, the NIPS device proceeds to apply more rule sets of additional complexity including the examination of packet header information, transport layer session state information, application layer session state information, context-sensitive string matches against the packet payload, application-layer analysis and finally complex regular expression matching. The NIPS device is able to quickly and effectively classify traffic using only the required processing to complete the analysis, thereby allowing for the NIPS to process additional traffic.

© 2017 Secure Anchor Consulting

Technet24

NIPS Challenges

- Organizations (and NIPS vendors) can't afford false positives
  - A NIPS false positive drops legitimate traffic
- Throughput of NIPS device must be able to keep up with traffic demands
  - A NIDS would miss traffic, NIPS stops legitimate traffic
- NIPS tend to have a less-extensive rule base
  - More false negatives than IDS tools

Remember that a false positive in relationship to a NIPS device means that legitimate traffic is dropped, inflicting a Denial of Service. Organizations can't afford false positives with NIPS devices, so vendors make use of a combination of passive OS and vulnerability detection, network architecture identification, hierarchical rule classification, and fewer rules than traditional NIDS devices.

Meeting the throughput requirements for NIPS devices is a significant challenge because the NIPS device must perform complex processing and analysis in order to eliminate false negatives and false positives while examining traffic. Latency is also a significant issue because the amount of time that is required to process a packet before deciding to transmit or drop it adds directly to the latency of the network. Just like a false positive from the NIPS, if the NIPS is unable to keep up with traffic demands, it causes a DoS on the network when traffic is dropped.

Ultimately, NIPS devices cannot have as extensive of a rule-base for identifying attacks on the network as an IDS. Where a NIDS can rely on an analyst to decide whether an alert is a true positive or a false positive, the NIPS device does not enjoy the same luxury.

© 2017 Secure Anchor Consulting

Developments in NIPS

- Improved throughput and response times
  - Near-real-time analysis and forwarding
- Automated analysis/signature updates
  - Might be a good or a bad thing
- Environmental anomaly analysis
- Protocol "scrubbing," rate limiting, and policy enforcement
- Passive analysis

This slide covers some of the recent advances in NIPS technology and how they impact organizations that are planning or have deployed NIPS systems.

As NIPS products become more mature, throughput capacity is increasing with reduced latency. We expect to see increased performance with more complex detection techniques as NIPS products become more mature and accepted in the marketplace.

The technology for automated analysis or signature database updates has been around for various products for a while, with NIPS vendors touting this feature for the ability to quickly respond to new threats. The ability to respond to new threats is certainly desirable, but with it comes the risk of poor traffic identification patterns that lead to false positives on the network. Exercise caution when implementing these features, using organizational policy to dictate the trade-off between the risks of new threats and the risks for dropped traffic.

What is anomalous with a given application or protocol in one environment might not be anomalous in the next environment.

One organization might utilize a busy public web server farm, with hundreds of web requests per minute. Another organization might utilize a single internal-use-only web server for the finance department. If the finance department network receives hundreds of web requests per minute, that would be considered anomalous for their environment but not for the server farm.

NIPS tools can detect these kinds of anomalies through configuration by the analyst or administrator to help determine where appropriate thresholds should be set. Because the NIPS device is simultaneously tracking connection state for thousands or even millions of connections, it can take a "broad perspective" view to detect anomalies that involve many connections across an entire enterprise.

Sitting inline has some advantages that aren't always directly related to thwarting malicious attacks. In some cases, a NIPS device can be used to clean garbage from the traffic stream, reducing overall network load. For

© 2017 Secure Anchor Consulting

example, a server that is attempting to close a connection with a workstation that has shut down might continue to send packets to the destination waiting for a response to say, "I'm done." The NIPS tool can use intelligence to recognize that the conversation is finished, and either drop the traffic received from the server or send a spoofed packet to the server on behalf of the non-responsive workstation to stop the traffic altogether.

Another feature of NIPS devices is the capability to use rate limiting to apply QoS mechanisms to network traffic. The administrator can identify traffic on the network that should receive higher or lower priority than other traffic, or limit the total amount of traffic from a particular network, host, or specific application. This feature is particularly useful when trying to manage throughput on Internet connections, where the administrator can limit the ability for a single application or host to consume all available bandwidth for the organization.

© 2017 Secure Anchor Consulting

Passive Analysis

- Utilizes passive analysis techniques to reduce false positives
- Correlates OS and vulnerability information with identified attacks
- Supports "network learning" mode to identify network architecture, structure

To help the NIPS identify false-positive traffic, vendors make use of passive-analysis techniques to identify host operating systems, network architecture, and what vulnerabilities are present on the network. After this information is gathered, the NIPS can use it to classify attacks against internal systems based on their operating system and vulnerabilities.

Network learning architecture also permits the NIPS device to identify internal attacks based on internal source IP addresses and router hop counts.

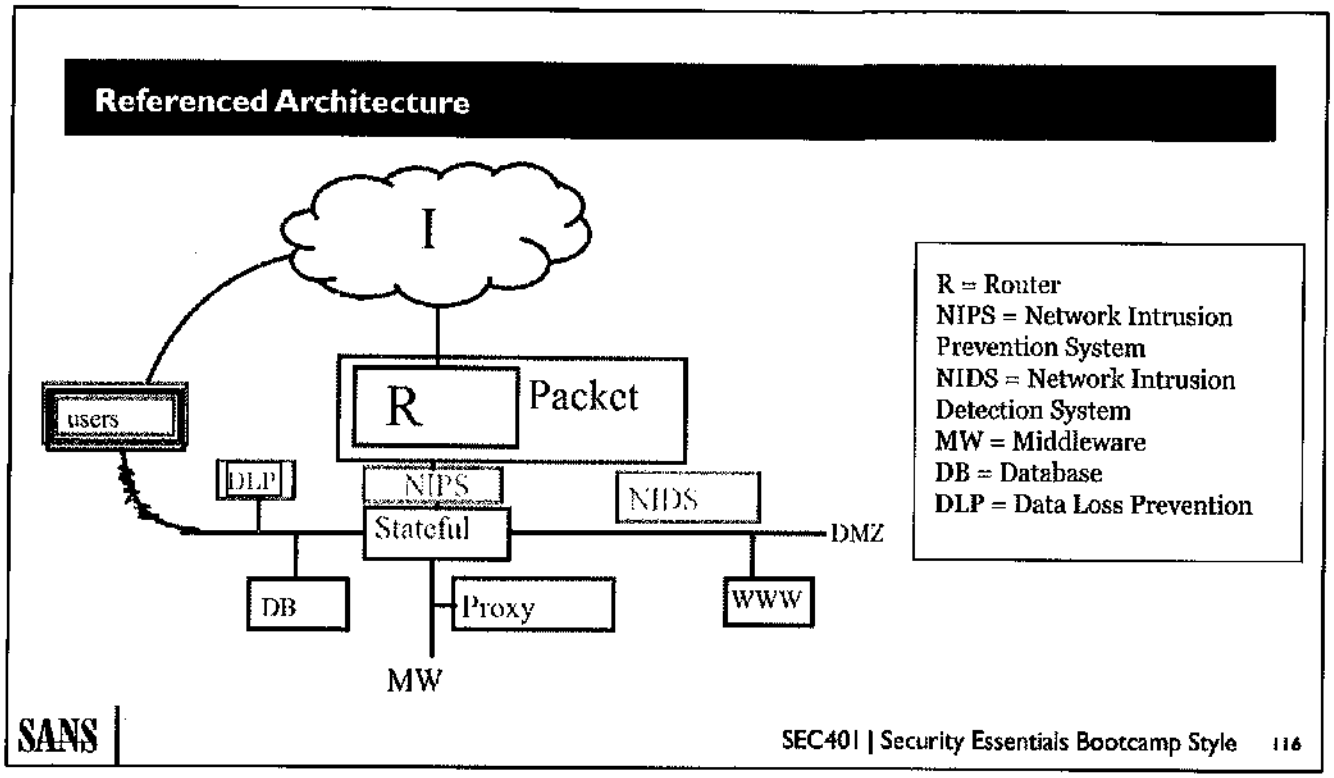
If the NIPS device sees traffic on its internal interface with an IP address that does not match a list of discovered internal networks, or if the TTL value is suddenly anomalous based on the history for the source IP address, the NIPS device can identify spoofed packets from the internal network, dropping them before they can exploit servers outside the organization.

Reference

1. Intrusion Prevention Systems, A Look Under the Hood – Tipping Point

© 2017 Secure Anchor Consulting

Technet24



In this section, we discussed several different network security devices and many different types. The question you might be asking is which one should we deploy. Hopefully, you recognize that the answer is going to be YES. You need to deploy all of these technologies in an integrated defense-in-depth solution. This is because any single technology has weaknesses, but deployed together, they can complement each other and provide an appropriate level of security.

In looking at a referenced architecture for network security devices, there are many correct answers. This slide is meant to show you one possibility. The trick is to make sure you have a proper balance of prevention and detection intermixed. Pen testing is a great way to verify and validate the robustness of your architecture.

© 2017 Secure Anchor Consulting

Network Security Device Summary

- Correctly placed network security devices can provide the proper balance between security and functionality
- Known bad attacks should be prevented but if there is a false positive, detection in a timely manner is a must
- Perform threat modeling and penetration testing to verify the overall security of your architecture
- Verify that least privilege is being implemented and adhered to

One of the areas that makes security so interesting and challenging is that you are only as secure as your weakest link. Therefore, in addition to host hardening, data protection, and encryption, it is critical to ensure proper network security devices are placed within your network architecture. While network security devices are useful, they only have value if they are properly designed and properly configured. Proper testing is required with proper tuning to make sure you have appropriate balance between prevention and detection.

© 2017 Secure Anchor Consulting

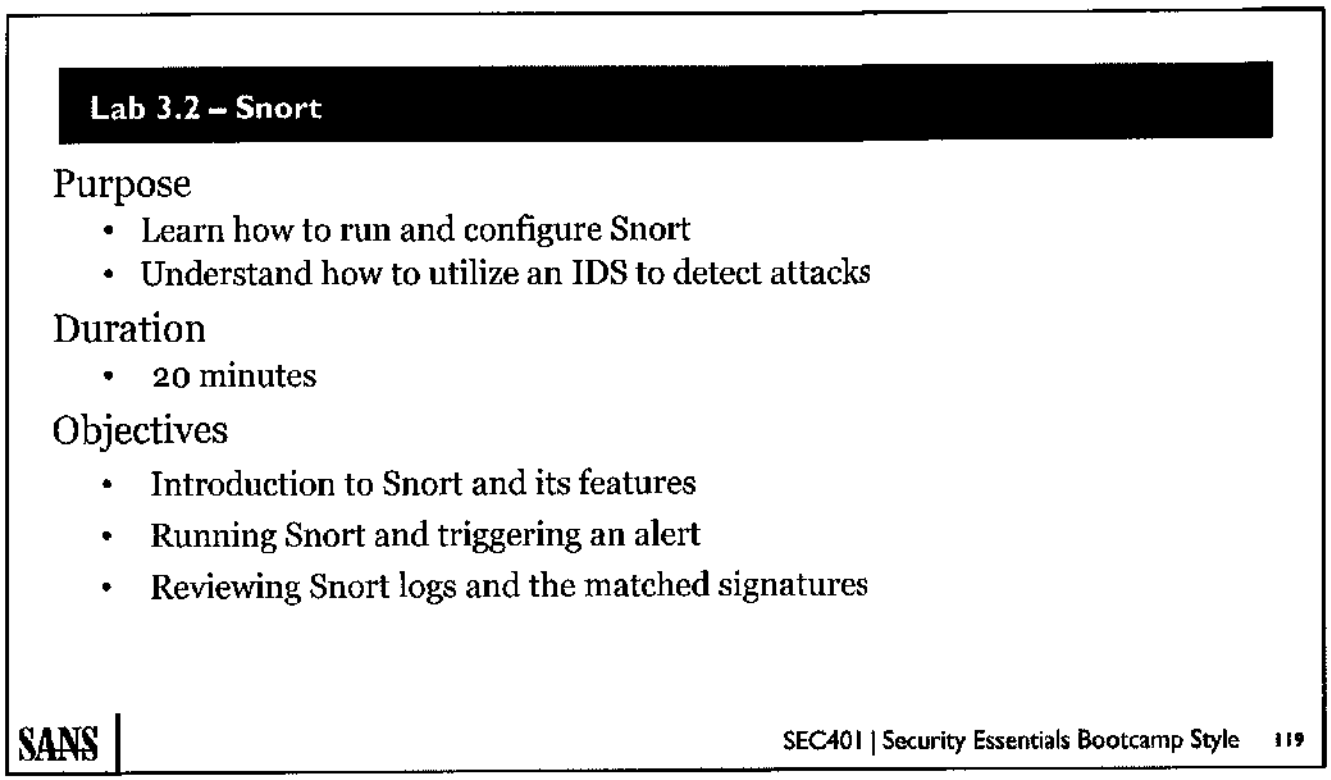
Technet24

SANS

Lab 3.2 – Snort

In the module, "Intrusion Detection Technologies (IDS)" you learned about the Snort intrusion detection system (IDS). Snort is a mature, widely used IDS with great community support. In October 2013, Sourcefire, the company who created Snort, was acquired by Cisco and is, fortunately, still freely available at https://www.snort.org/. Snort is already installed on your Kali Linux VM and ready to use. It comes with a large number of IDS signatures, each categorized into different groups such as exploit, DoS, VoIP, and scanning. As an analyst, you can choose which categories you want to include or exclude. Turning on all the default rules can generate a large number of false positives and, as such, an IDS requires ongoing tuning in order to get the most value. Snort is also extensible, meaning that you can write custom signatures and other functionality to meet your needs.

© 2017 Secure Anchor Consulting



Purpose

- Learn how to run and configure Snort
- Understand how to utilize an IDS to detect attacks

Duration

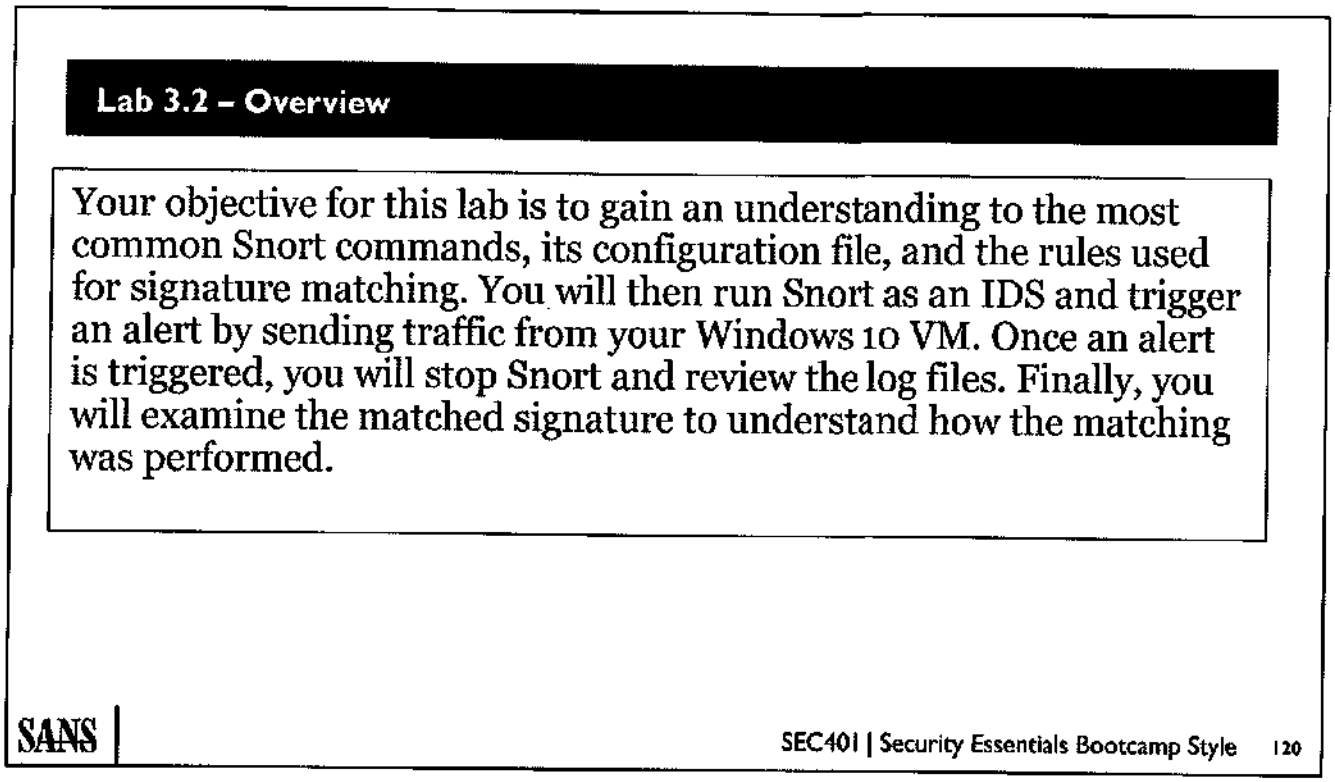
- 20 minutes
- The estimated duration of this lab is based on the average amount of time required to make it through to the end. The duration estimate of this lab can decrease or increase depending on various factors, such as the booting of virtual machines, the speed and amount of RAM on your computer, and the time you take to read through and perform each step. All labs are repeatable both inside and outside of the classroom, and it is strongly recommended that you take the time to repeat the labs both for further learning and practice toward the GIAC Security Essentials Certification (GSEC).

Objectives

- Introduction to Snort and its features
- Running Snort and triggering an alert
- Reviewing Snort logs and the matched signatures

© 2017 Secure Anchor Consulting

Technet24



Your objective for this lab is to gain an understanding to the most common Snort commands, its configuration file, and the rules used for signature matching. You will then run Snort as an IDS and trigger an alert by sending traffic from your Windows 10 VM. Once an alert is triggered, you will stop Snort and review the log files. Finally, you will examine the matched signature to understand how the matching was performed.

© 2017 Secure Anchor Consulting

SANS

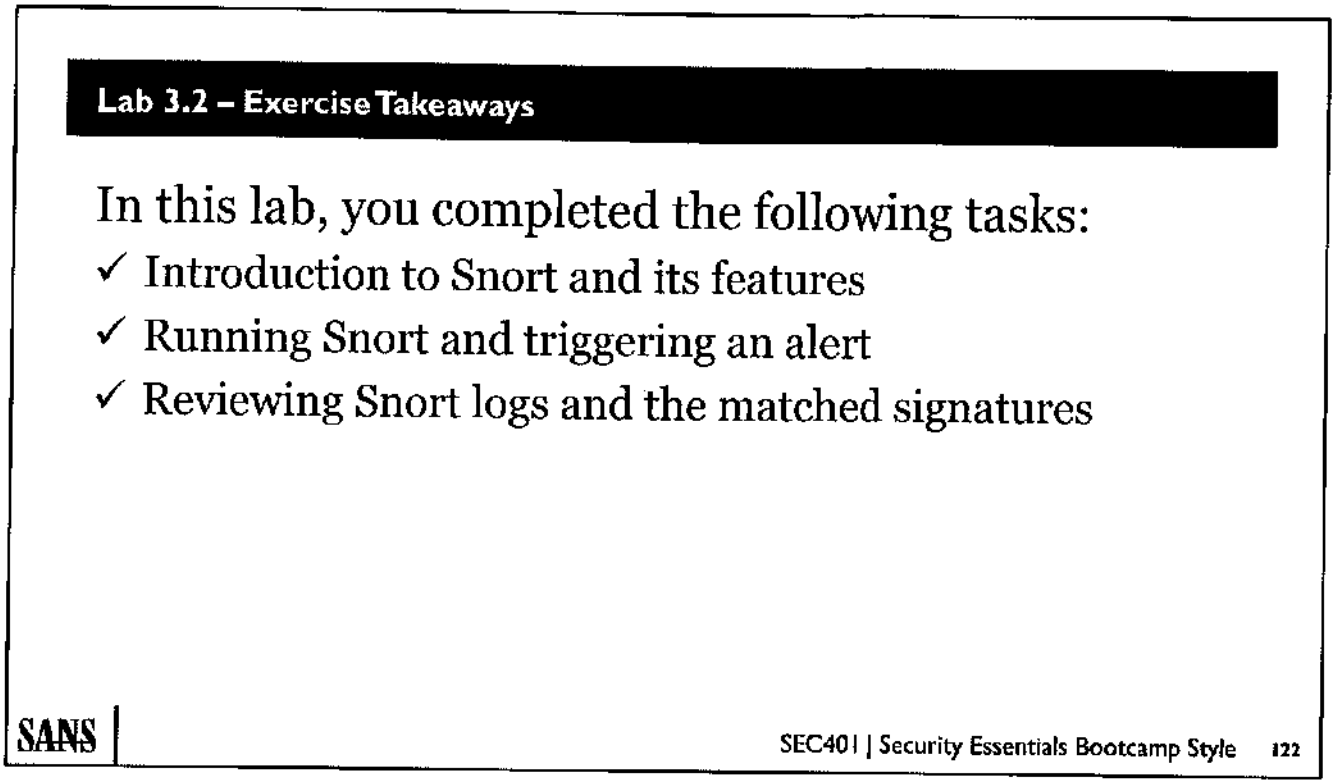
NOTE: Please open the separate Lab Workbook and turn to Lab 3.2

The instructor is going to introduce and go over the labs. Once the instructor is done, you will be instructed to work on the lab. If you have any questions, you can ask the instructor.

This page intentionally left blank.

© 2017 Secure Anchor Consulting

Technet24



In this lab, you completed the following tasks:

- ✓ Introduction to Snort and its features
- ✓ Running Snort and triggering an alert
- ✓ Reviewing Snort logs and the matched signatures

As you can see, Snort is a powerful and flexible IDS. We looked at some of the basic operations of Snort and its configuration file, as well as its logging locations. We then ran Snort, loading TFTP rules, and ran a script from our Windows 10 VM that was designed to trigger some signatures. Examining the alert file showed that we successfully triggered two signatures and were able to see what data was sent across in the "snort.log" file. Finally, we used Snort to run its checks against an existing PCAP file.

© 2017 Secure Anchor Consulting

SANS

Lab 3.2 is now complete

This page intentionally left blank.

© 2017 Secure Anchor Consulting

Technet24

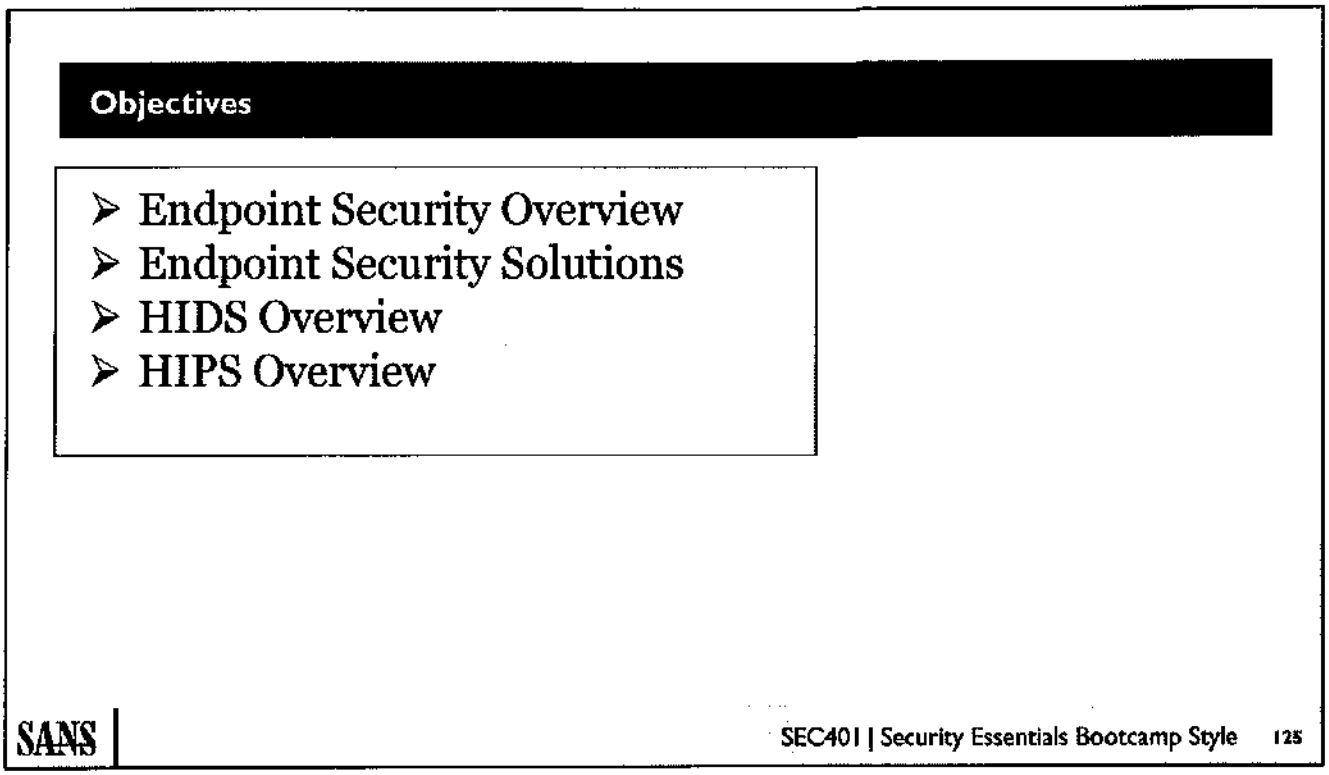
Security 401: Day 3 | Security Essentials Bootcamp Style

SANS

Module 15: Endpoint Security

Module 15: Endpoint Security
While attackers are going to typically come in over a network, ultimately they are going to compromise an endpoint to either maintain persistence, setup a pivot point, compromise sensitive data or allow the adversary access via a command and control channel. Therefore proper protection of the endpoint is critical to maintaining an effective level of security.

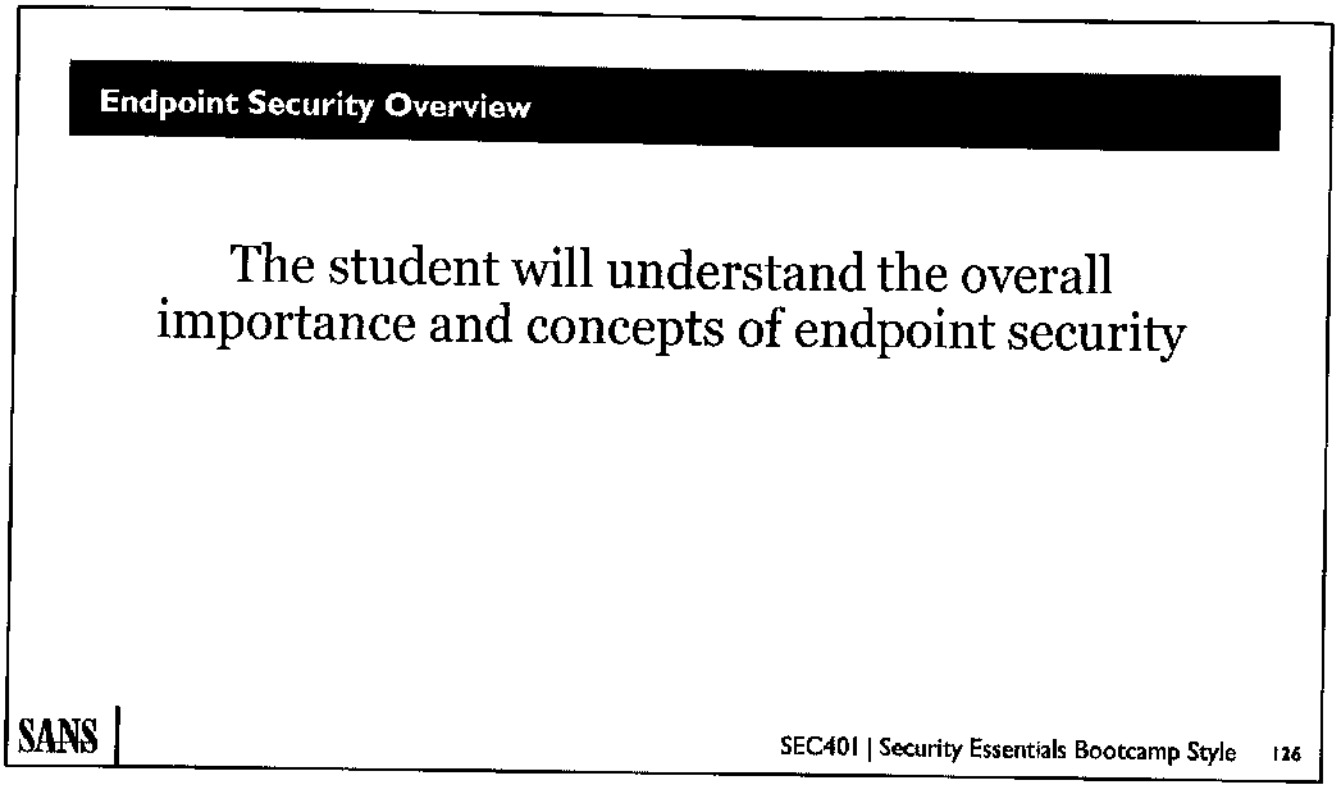
© 2017 Secure Anchor Consulting



In this module, we will examine some of the key components, strategies, and solutions for implementing endpoint security. This includes general approaches to endpoint security, strategies like baselining, and solutions like HIDS and HIPS. After this module, you will have an understanding of the importance of protecting the endpoint and ways to achieve this.

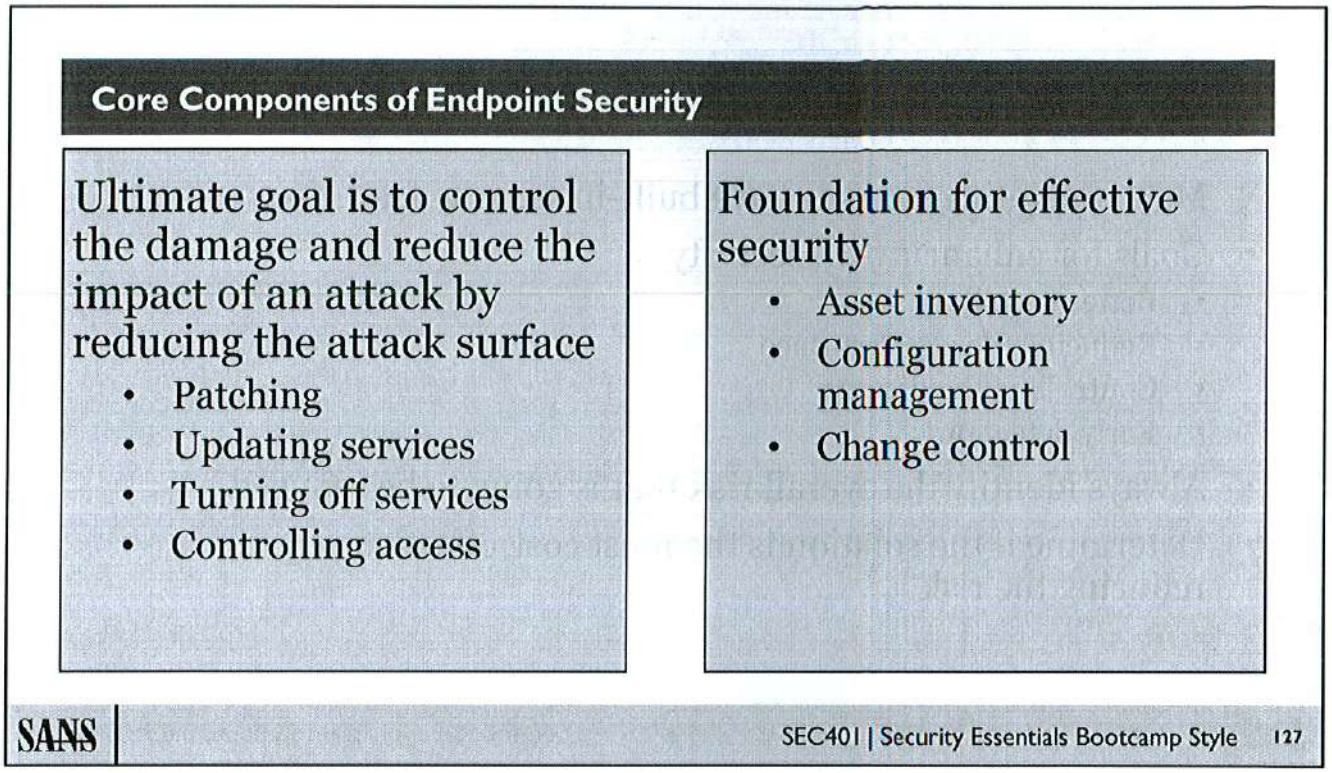
© 2017 Secure Anchor Consulting

Technet24



This page intentionally left blank.

© 2017 Secure Anchor Consulting



This advice has been said multiple times, but it is worth repeating: "Prevention is ideal, but detection is a must." When we look at securing and locking down endpoints, the goal is not solely prevention, but controlling the damage. Breaches are going to occur, so the key question to ask is, "If a system is compromised, how much damage can be caused to the system?" It is also important to remember pivot points. Often, when a system is compromised, it is used as a pivot point to break into other systems. Therefore, in answering the question, you have to look at what visibility the system has and how much damage can be caused to the entire network, not just the system that was initially compromised.

It is also important to make sure that any computer has a solid foundation which begins with proper hardening and securing of the underlying operating system.. People often focus on the latest technology, but if the system does not have a solid foundation, it will never be secure. It is critical to make sure that the system has clear asset inventory, configuration management, and change control. Without clear visibility into the vulnerability of a system, securing it will be hard.

© 2017 Secure Anchor Consulting

Technet24

Enhancing Endpoint Security

- Many operating systems have built-in security features
- Goals for enhancing OS security
  - Better visibility
  - Reducing the attack surface
  - Controlling the damage
  - Early detection
- Always identify the overall risk that is going to be reduced
- Determine if the solution is the most cost-effective way of reducing the risk

Many operating systems have built-in solutions and capabilities for implementing security. In utilizing any tool, it is important to remember what the goal of security is and, most important, what you are trying to accomplish. Often, engineers use and/or install a ton of tools, and when asked what the security goal is, they do not have a good answer.

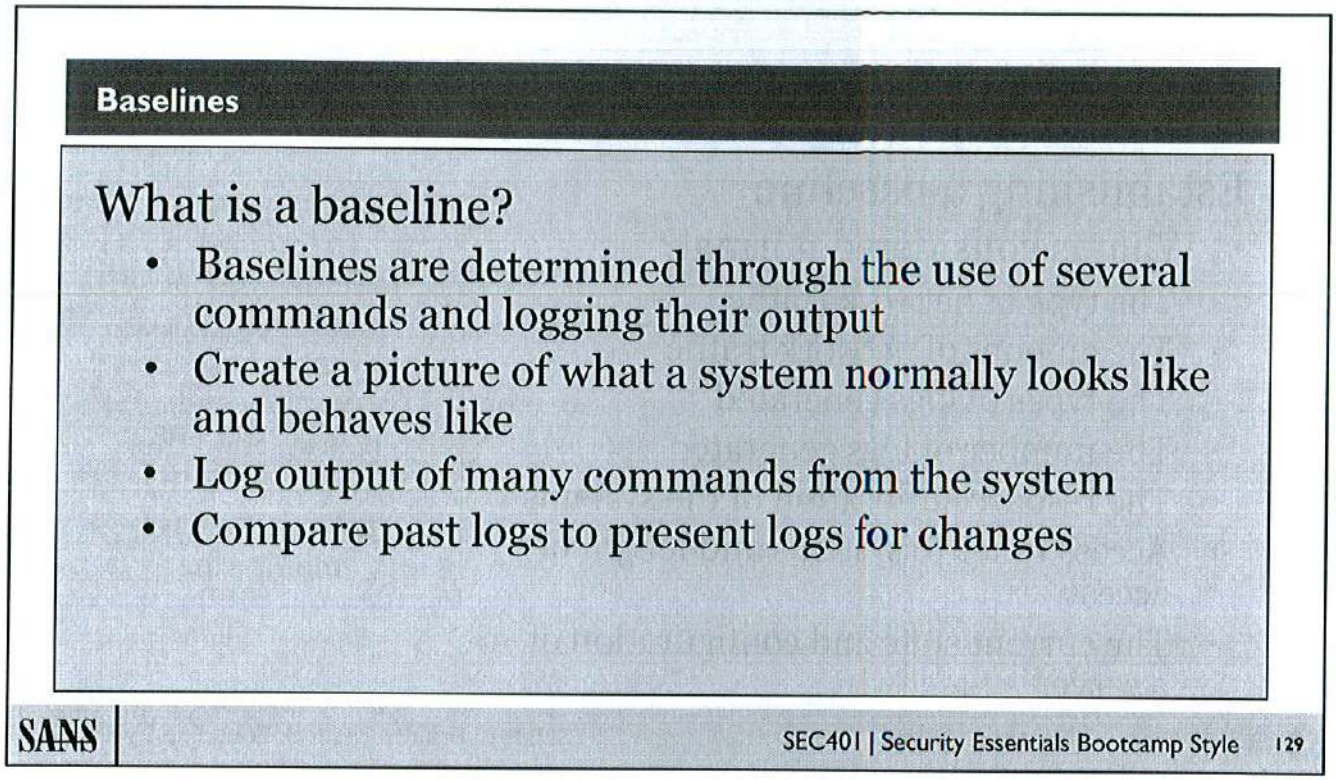
Organizations today are not doing a great job with detection. We hear all these stories in which organizations have been compromised for over a year and the entity had no idea. The reason is that they do not have proper visibility into what is happening on the system. Many security professionals are accustomed to visible attacks, and when the adversary goes into stealth mode, the attack is missed. Understanding what is happening on a system so that anomalies can be detected and investigated is a key part of security. The reason we want to do timely detection is to control the overall damage. We know that we cannot prevent all attacks, but if we can reduce the attack surface, we can make it harder for a compromise to occur.

Ultimately, security is all about reducing the risk to critical assets. Therefore, any time you are going to use a tool, you should always ask these questions:

- What is the risk?
- Is this the highest priority risk?
- Is this the most cost-effective way of reducing the risk?

Focus on reducing the highest priority risks with the most cost-effective measures, and you will go a long way to securing your organization.

© 2017 Secure Anchor Consulting

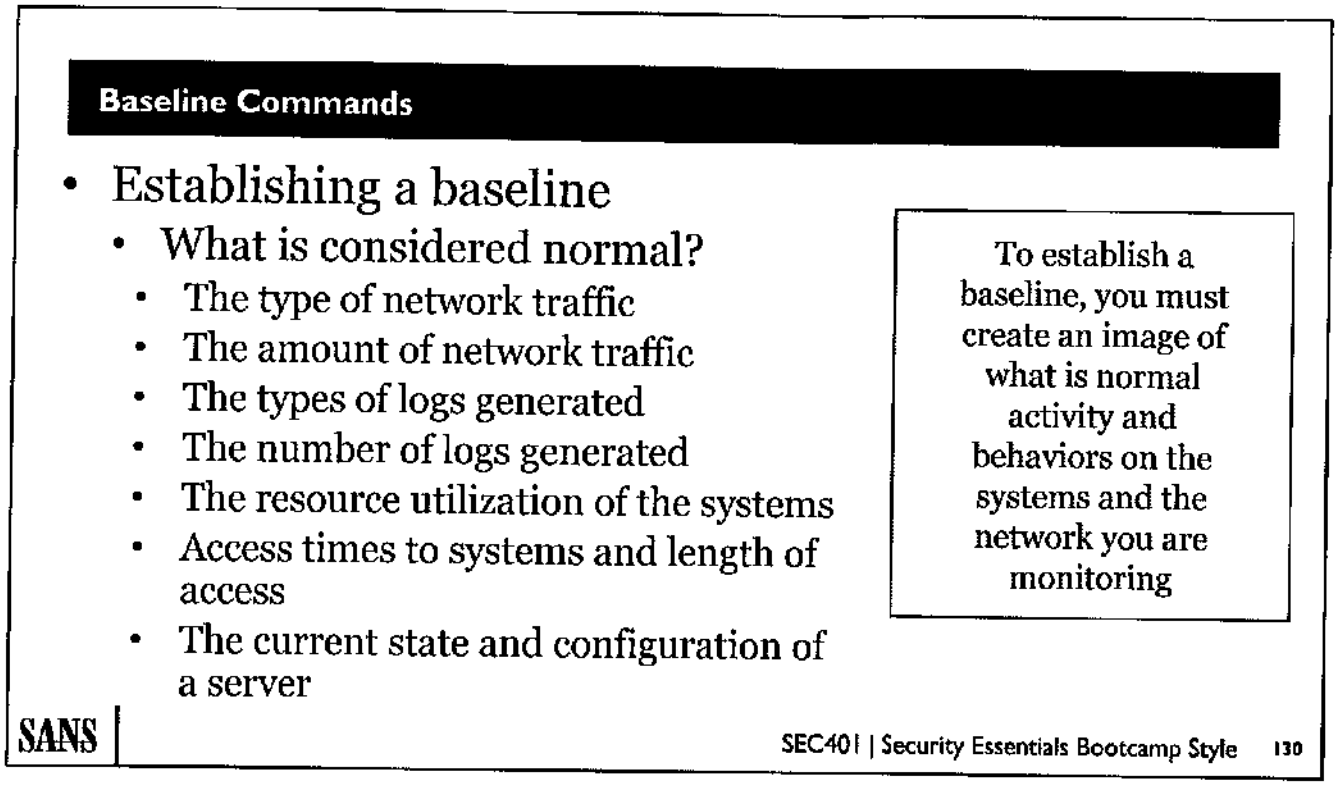


A seasoned network or system administrator who has been working on and in an environment for a long time will just innately know when something is wrong on the system or network. The reason is that the administrator has become aware of what normal operations and behavior are for the systems and network, and anything that deviates from that norm raises suspicion. This would be considered a form of baselining. Many system administrators have an informal baseline in place, and those administrators who have a formal baseline are just that much more likely to find anomalous behavior on their systems and network.

Baselines are determined through the use of several commands and logging their output from the system and then comparing the results of those same commands later to see if a difference occurs in the results. If there is a change that is not expected, an investigation should begin to determine if the change was a normal event or a malicious security event.

© 2017 Secure Anchor Consulting

Technet24



To establish a baseline, you must create an image of what is normal activity and behaviors on the systems and the network you are monitoring. It is always best to do this during a new network build, but usually, you don't have that luxury. Start logging all the network, system, and resource utilization information that you will find useful. The more you capture and trend, the more ways you can potentially identify outliers of activity to identify an issue, whether it be security related or just a system/network problem that needs to be fixed.

Common areas to monitor to create a normal baseline include the network traffic that is being transmitted, the protocol and ports used by systems, and the times they are being used. It is also good to know how much network traffic is normal at certain times of day from each system.

Baseline the kinds of logs generated by a system and the amount these systems are generated. Your baseline will generally follow a typical pattern, and your environment will deviate very little from that pattern.

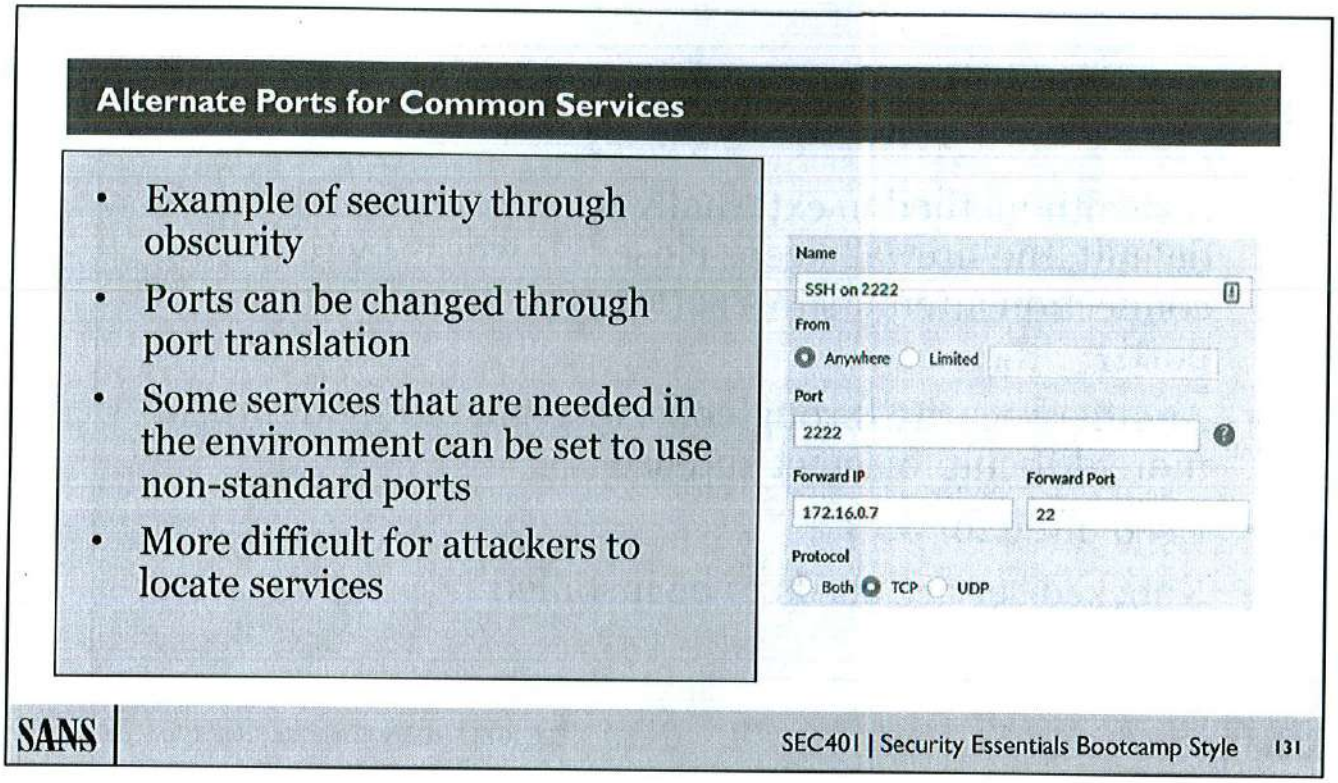
Systems will generally use the same amount of resources at certain times of day and on certain days of the week or month.

Users will generally only ever access systems at certain times from certain locations and rarely deviate. Some systems hardly ever get accessed.

Create a snapshot of the state of a server in its current configuration and the files that are on it. Do so by copying the configuration and monitoring for changes to the configurations and the files themselves through hashing.

All of this helps produce a picture of the pattern of normality over time on your network and can be used to help identify anomalies.

© 2017 Secure Anchor Consulting

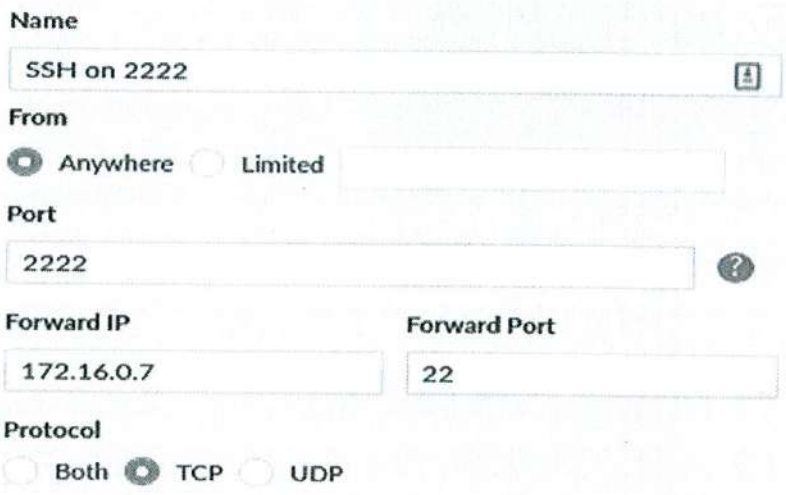


Even though this is an example of security through obscurity, it does assist to some degree. For some services that are needed in the environment, you can set programs to use new ports. This technique makes it that much more difficult for attackers to locate services on nonstandard ports. It has been demonstrated that a large percent of attacks are avoided by simply changing the standard port of the service. You must consider weighing the benefit of changing the port versus ease of use. For example, it may not be wise to change the standard port for your web server for your public website because most end users don't understand ports, and this will just cause issues in the general user population from accessing your web server.

Each service can be configured with different ports within the service configuration files.

Reference

1. UniFi - USG Port Forwarding Configuration and Troubleshooting - https://help.ubnt.com/hc/en-us/articles/235723207-UniFi-USG-Port-Forwarding-Configuration-and-Troubleshooting



© 2017 Secure Anchor Consulting

Technet24

Port Knocking

- A stealth method to externally open ports that, by default, the firewall keeps closed. It works by requiring connection attempts to a series of predefined closed ports.
- Another security through obscurity method, but much more difficult to detect and use
- Used often for SSH
- Knocked service needs to be installed

In port knocking, a service listens for a certain sequence of connection attempts on predetermined TCP or UDP ports; it does so by watching the firewall logs or having a packet capture on an interface. When a sequence of port connections matches the predetermined port connection combination, the service adjusts the iptable rules to allow the connection on an actual TCP or UDP port where the service is truly listening for use. This method is the very definition of security through obscurity, but it does an amazing job of making it difficult to impossible to determine whether a service is even listening on a system. It is not practical for publicly accessible and useable services such as web servers, but it is an excellent solution for remote administration services such as SSH.

Reference

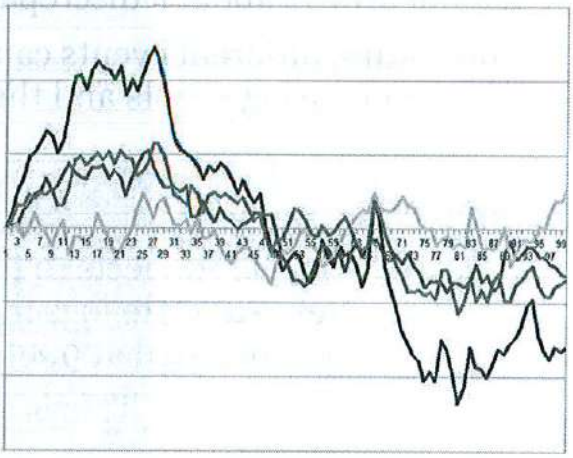
1. "How To Use Port Knocking to Hide your SSH Daemon from Attackers on Ubuntu", https://www.digitalocean.com/community/tutorials/how-to-use-port-knocking-to-hide-your-ssh-daemon-from-attackers-on-ubuntu

© 2017 Secure Anchor Consulting

Detecting Anomalies

Detecting anomalies

- Using tools to sift through the background noise enables you to find anomalies
- Visualizing all the changes can make it much easier to find outliers of system and network activity in your environment



SANS | SEC401 | Security Essentials Bootcamp Style 133

There is no such thing as an invisible adversary. Any adversary—even advanced stealthy adversary—is going to make changes to a system and/or alter the running state of a system. Therefore, creating a verified baseline of a system and rerunning it at set intervals is a great way to spot anomalies, which would enable you to identify changes on a system.

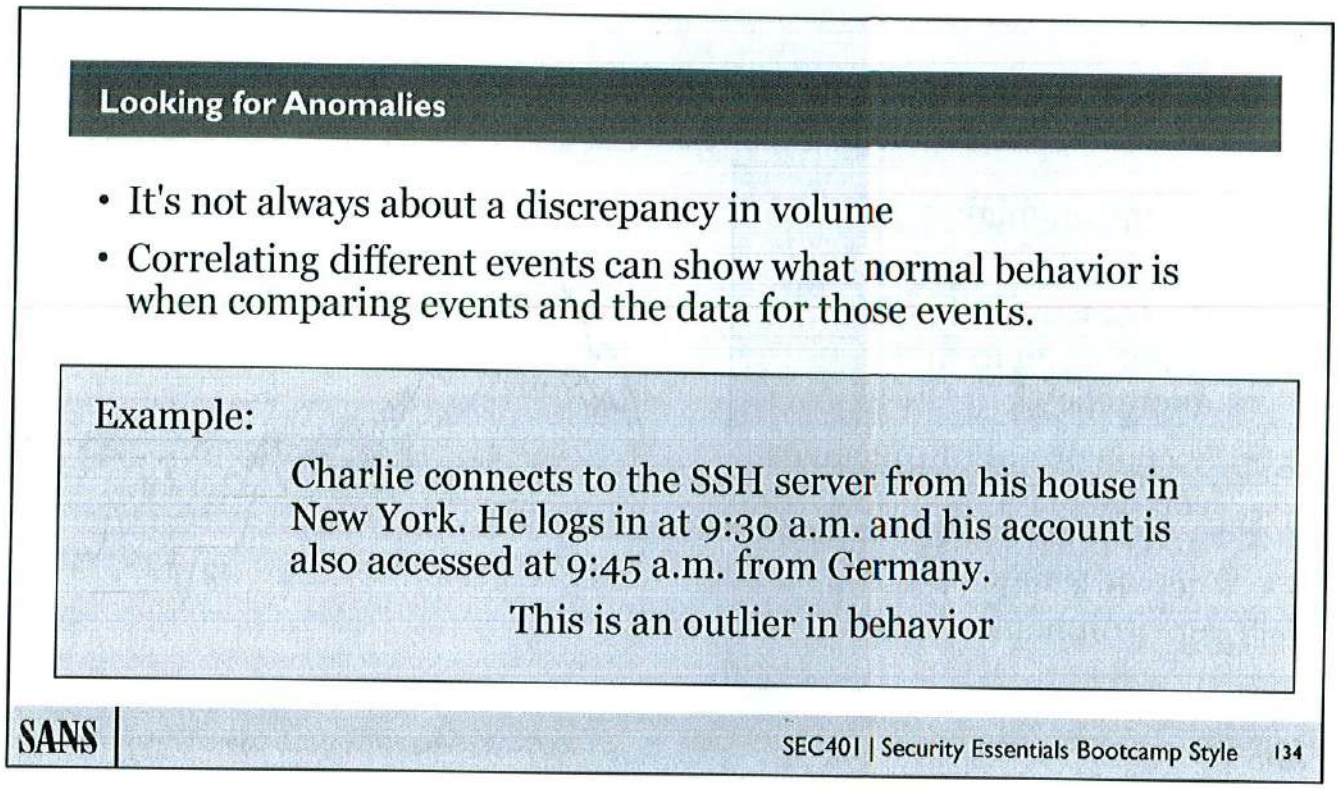
Sometimes it is quite easy to find anomalies in your systems or in your environment. Sometimes, though, it can be quite hard to discover anomalies, especially if you are using simple scripts to check the baselines of systems, one system at a time. Using file-checking tools is often quite helpful with a standalone server, and combining file checkers with a centralized log aggregation solution and graphical UI to visualize all the changes can make it much easier to find outliers of system and network activity in your environment.

Reference

1. Real-time anomaly detection - http://www.ibm.com/developerworks/library/bd-streamsanomalydetection/

© 2017 Secure Anchor Consulting

Technet24



When you create a baseline of a system or network, you gain a picture of the pattern of normality over time. This helps in identifying outliers in the pattern, especially when it comes to volume shifts that are easy to identify. Say you normally see 100 logon events per day, and now you see 5,000. In this case, you should investigate this volume-based discrepancy.

Not all outliers are volume-based. Often, the outliers are simply those that deviate slightly from the norm but do so in two or more areas at a time, which is where behavior-based anomaly detection comes in.

In this example, Charlie always logs in to the SSH server from home. He logged in today at 9:30 a.m. from his usual New York IP address, and then 15 minutes later, his account was accessed from an IP in Germany. You can combine the short time gaps between logins and the location from which the logins occurred. Therefore, it is unlikely Charlie logged in again 15 minutes later using a proxy that has him exit a German IP address, and it is even less likely he is actually in Germany, so you know this is highly likely to be nefarious-based behavior.

© 2017 Secure Anchor Consulting

Endpoint Security Solutions

The student will understand some of the various endpoint security solutions that can be deployed within their organization

This page intentionally left blank.

© 2017 Secure Anchor Consulting

Technet24

Endpoint Firewalls

- Firewalls are critical to manage and filter traffic
- Network firewalls provide a boundary defense
- Host-based firewalls are needed to complement network firewalls
- Provides additional protection from a defense in depth perspective.
- Allows for protection of traveling laptops that directly connect to the Internet from untrusted locations

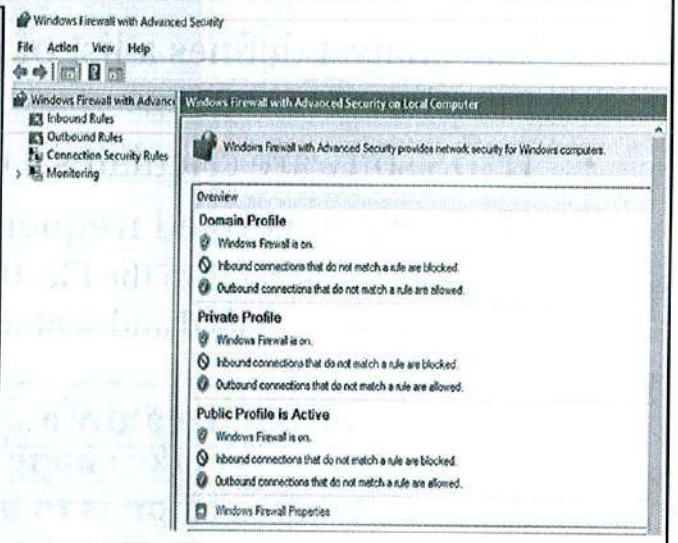
Even though firewalls cannot prevent all attacks and detection is required, a well-configured firewall is still a key part of an effective defense. In addition, a properly designed firewall can provide inbound prevention and outbound detection.

Network-based firewalls are ideal for setting up boundaries between different networks and/or to control the flow in and out from a sensitive network. However, network firewalls only protect systems behind the firewall. With traveling laptops, additional protection of the computer is needed via host-based firewalls. While host-based firewalls can be customized to protect a specific system, they are sometimes hard to manage in a large organization.

© 2017 Secure Anchor Consulting

Types of Endpoint Firewalls

- Packet Filter (Stateful)
  - Control only authorized services
  - Block unauthorized IP addresses
- Application Control and Operating System Control
- Endpoint security suites focus on desktop lockdown, which includes personal firewalls

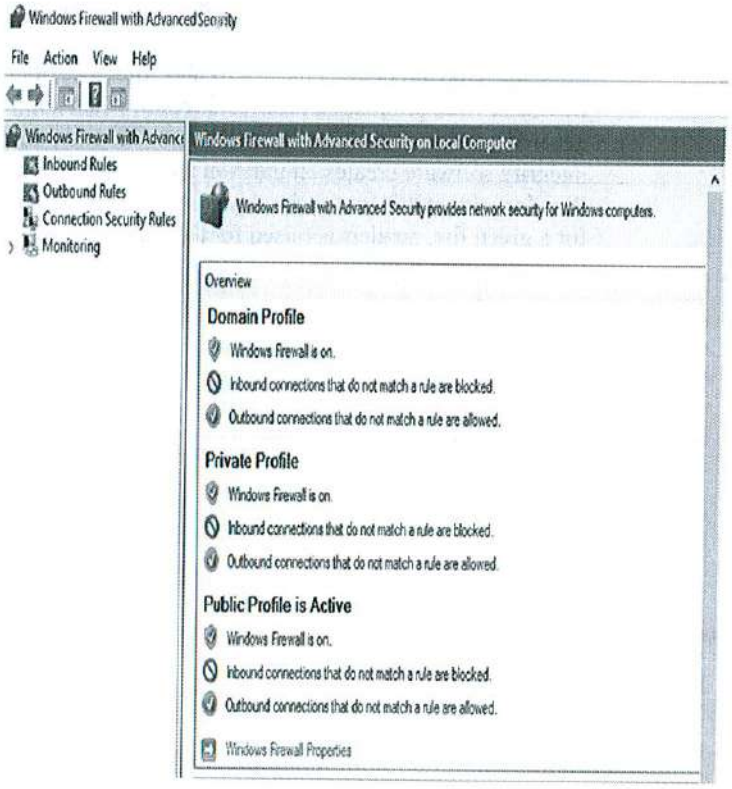


Now we're going to change the context from firewalls that control traffic on relatively large networks and protect many computers to firewalls that protect individual computers and most often exist as software within those computers. Although small, inexpensive firewall appliances exist, "personal" firewalls usually are thought of as software residing on an individual computer, often a PC.

The packet filter approach to personal firewalls looks at packets coming from the network to the PC. These tend to treat the PC as the trusted domain.

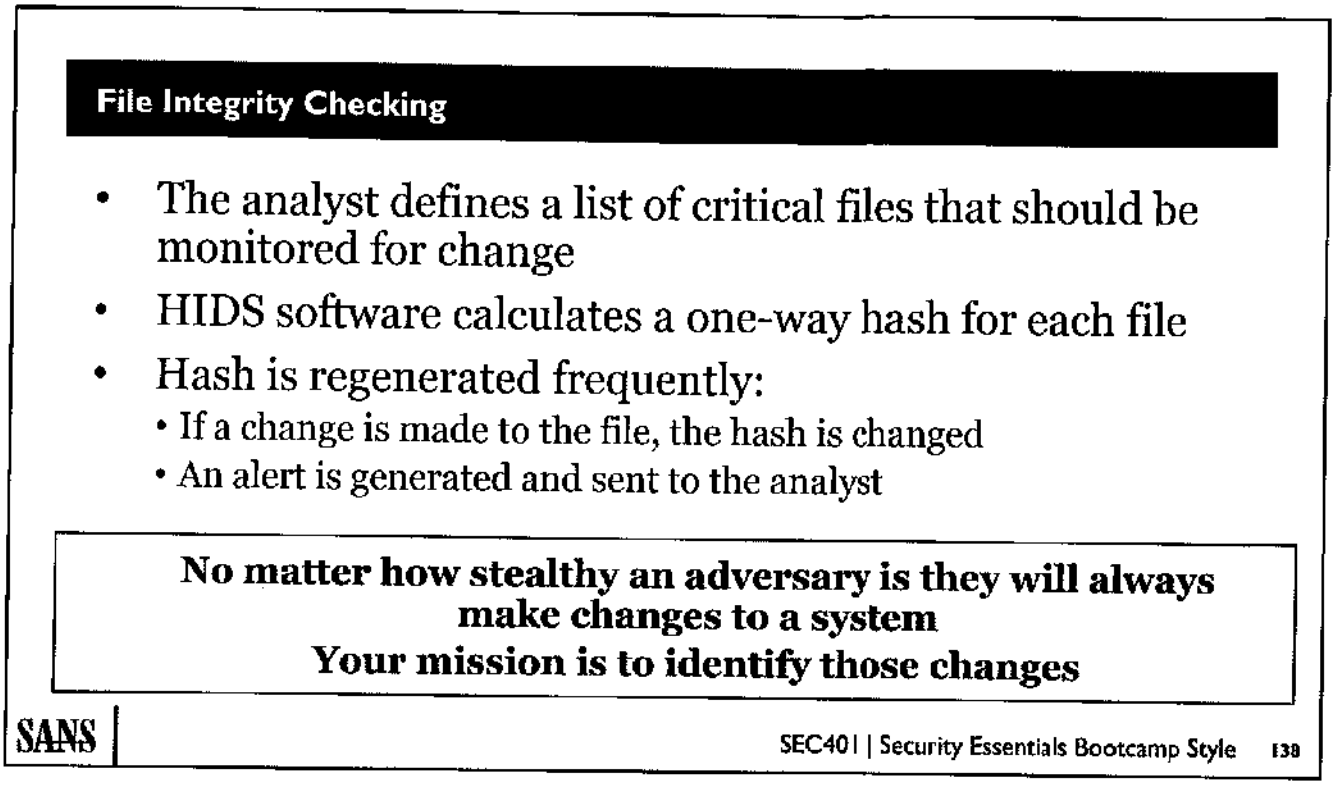
One of the most popular approaches to commercial personal firewalls is application control firewalls. These have the capability to screen incoming packets, but also keep a set of rules for applications. This allows the trust domain to be much more granular. For example, you could configure a rule to allow a specific application to connect to a specific IP address or a specific port. The significance of this as a user is that when an unknown application on the PC attempts to access the Internet, it is detected by the firewall, and the user is given an opportunity to approve the connection or refuse it.

The most flexible trust model of all is operating system control. These tools will not even allow a program to run, much less access the Internet, until it is approved.



© 2017 Secure Anchor Consulting

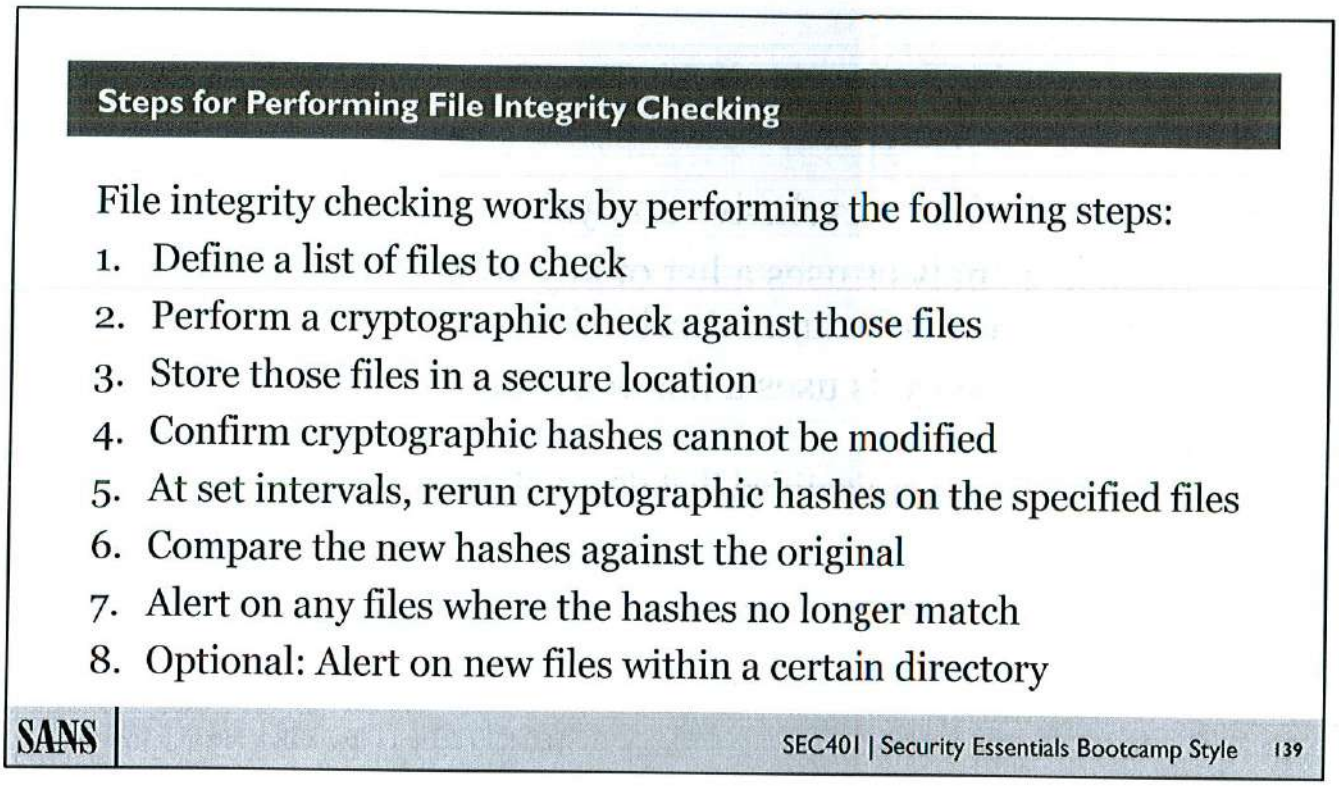
Technet24



File integrity checking is used to identify when any file has been changed on the host system. Any file that should be monitored can be defined by the analyst to monitor for unauthorized changes. This can be business documents, such as policies or business plans, or operating system files, or even web server content files.

To identify unauthorized changes to files, file integrity checking uses a mathematical function called a "one-way hash" that produces a hash-value result when applied to a monitored file. The hash algorithm always generates the same hash value on the same data file unless a change has been made to the file. The file integrity software creates an index of all the monitored files on the host with their associated hash values. Regularly, the HIDS software checks the hashes of monitored files and, if the previous hash does not match for a given file, an alert is raised for the analyst.

© 2017 Secure Anchor Consulting



The premise of file integrity checking(FIC) is to use a unique fingerprint of a file that will indicate whether a file has been altered. One powerful method of doing this is to use cryptographic hashing. Besides collisions, the only way the output of a hash will be the same is if the inputs are the same. Therefore, if a hash is the same, this means the file has not changed; and if the hashes for the same file are different, this indicates it has been altered.

FIC uses the following steps. First, define a list of the key critical files that should not change during the normal running of the system. The more aggressive, the more effective the tool, but the higher chance for false positives. The less aggressive, the less effective, which leads to a higher chance of missing attacks or a higher false negative.

Second, run cryptographic hashes on the list of key files and store them in a secure location. These hashes need to essentially be read only. The file integrity checking software needs to be able to read the files, but they cannot be modified. The reason is obvious: if someone can modify a file and also modify the hash, there would be no way to detect when an adversary modifies a file.

After the hashes have been captured, at set intervals the software would rerun the hash of each file and compare it to the value in the database. If they are different, an alert is triggered, indicating that a file has been modified. This would require someone from the security or incident response team to investigate.

Because adversaries often create new files instead of modifying existing ones, most file integrity software will also alert on new files that are created in a certain directory.

© 2017 Secure Anchor Consulting

Technet24

How Log Monitoring Works

- Uses inclusive or exclusive analysis
- Inclusive analysis uses a list of keywords to watch for
  - When a match is found, an alert is raised
- Exclusive analysis uses a list of events that can be ignored
  - When an event is identified that does not match the ignore list, an alert is raised

SANS | SEC401 | Security Essentials Bootcamp Style 140

Log monitoring is another mechanism of host-based intrusion detection that analyzes the log messages from operating systems and applications. This mechanism of monitoring uses inclusive or exclusive analysis to define events of interest from log files.

Inclusive Analysis
This measure of log monitoring uses a list of keywords or phrases that define the events of interest for the analyst. The keyword list might contain words from operating system log files, such as "login failed," or "unauthorized access," or more specific events, such as a web server exploit that would show up in web server log files.

The HIDS system takes a list of keywords to watch for and generates alerts when it sees matches in log file activity. For example, an analyst might configure the HIDS to generate alerts when the keyword "Failed" and expect to receive alerts as follows:

Feb 7 09:22:20 www sshd[2630]: Failed password for root from 10.9.1.88 port 1024 ssh2

Feb 7 13:52:47 www lbal[626]: LD-CRIT Real machine '192.168.77.149:0:0:tcp': Remains Failed

Inclusive analysis is only as effective as the keyword list used to generate alerts.

If there is a significant event in the log file that doesn't include any of the keywords configured by the analyst, the HIDS does not know to generate an alert. In some cases, it might be better to use the exclusive analysis technique for monitoring logs to overcome this limitation.

Exclusive Analysis
Like inclusive analysis, the analyst generates a file of keywords and phrases for use by the HIDS. Unlike inclusive analysis, the HIDS uses the keyword and phrase list to exclude log entries. The log entries that do not match the exclusive keyword list are raised as alerts to the administrators.

© 2017 Secure Anchor Consulting

This configuration works well for limited configurations where the information contained in log entries are predictable and repeated frequently. For example, an administrator that regularly reviews the logs from a firewall might exclude the message codes from known-benign events, and alert on all other events.

Consider the following logging entries:
%XXX-4-106023: Deny tcp src outside:10.9.200.54/6346 dst student:10.181.231.44/3156 by access-group "acl_outside"

%XXX-4-106023: Deny icmp src inside:10.112.4.17 dst outside:10.109.254.139 (type 8, code 0) by access-group "acl_inside"

%XXX-3-106011: Deny inbound (No xlate) tcp src outside:10.203.152.218/1665 dst outside:10.7.249.145/35945

%XXX-2-106017: Deny IP due to Land Attack from 10.181.224.13 to 10.181.224.13

%XXX-4-410001: Dropped UDP DNS reply from dmz:10.2.1.13/35100 to outside:10.16.224.101/53; packet length 564 %XXX-4-106023: Deny tcp src outside:10.6.200.4/80 dst student:10.181.239.86/3993 by access-group "acl_outside"

%XXX-4-106023: Deny icmp src inside:10.112.4.17 dst outside:10.109.254.141 (type 8, code 0) by access-group "acl_inside"

By using an exclusive filter matching the message codes XXX-4-106023, XXX-3-106011, and XXX-4-410001, the administrator can reduce the alerts from the HIDS to alert only on the following message:

%XXX-2-106017: Deny IP due to Land Attack from 10.181.224.13 to 10.181.224.13

Log monitoring is a powerful mechanism for host-based intrusion detection that offers the administrator a lot of flexibility. Log analysis is a measure that can be implemented by any organization at little cost. If you are collecting logging information, you can utilize open-source or commercial tools to analyze the log files for HIDS reporting.

© 2017 Secure Anchor Consulting

Technet24

Application Whitelisting

- Utilizes an approved list of applications
- Only those applications can run
- All applications and associated files are cryptographically verified
- Applications can only run if they are approved and all cryptographic hashes are valid
- Can run in passive (alerting) or active (blocking)
- Relies heavily on environments that have robust configuration management and change control processes

A great method for protecting, controlling and securing an endpoint is via application whitelisting. Application whitelisting creates a list of authorized files and only allows those files to execute on the system. This approved list of executable files is cryptographically hashed. Whenever an executable is going to run on the system, it is first checked to see if it is approved. If it is not approved, it is prohibited from running. If it is approved, the executable and all associated files are cryptographically verified to make sure they have not be altered or modified in any way. From an attacker's perspective, think about how you would attack the system. You cannot add, install or change an executable. This would make it very difficult to compromise an endpoint.

One of the challenges with application whitelisting is that it is difficult to install in most environments. Application whitelisting is based on the premise that you have configuration management with a robust change control process in place. Essentially you know what is needed to run and there are minimal to no changes required. Since most organizations are not very mature on configuration and hardening procedures, this creates a challenge for application whitelisting. The reason is simple. With application whitelisting it is difficult to install or update software during the normal running of a computer. If your users need to perform this activity on a regular basis, it will not be allowed with application whitelisting.

While deploying application whitelisting in blocking mode, is recommended, it cannot be done in some environments. In these cases we still recommend application whitelisting in passive or alerting mode. This is not ideal but at least will provide visibility to the SOC (security operations center) of potentially harmful activity.

© 2017 Secure Anchor Consulting

HIDS Overview

The student will have a general understanding of the techniques used by host-based intrusion detection systems

This page intentionally left blank.

© 2017 Secure Anchor Consulting

Technet24

HIDS Overview

- Provides much of the functionality of a NIDS, distributed to each host
- Can be more granular than NIDS, analyzing activity on the host
- Uses signature and anomaly analysis with unauthorized change monitoring, log monitoring, and network monitoring
- Local processing/alerting may be done, but data generally is sent to a central location for parsing

Host-based intrusion detection works on a single host, identifying events of interest that are configured by the administrator. Unlike a network-based IDS, host-based intrusion detection must be installed on the host it monitors, and it only monitors a single host. Therefore, most HIDS deployments target multiple or all hosts in an organization to collect events of interest at the host level.

A HIDS tool has the capability to provide additional granularity to the analyst because it can monitor more activity than what a NIDS can see. In addition to being able to monitor the network, a HIDS can monitor the use of the system altogether and any additional "backdoors" of access, including wireless connections.

For example, an organization might set a policy that forbids rogue software on company computers, but the policy would be unable to stop someone from downloading and installing an ActiveX component for a web browser that monitors the websites that are visited to deliver more effective marketing information. Using a HIDS system, administrators can define the addition of new software to a computer as an event of interest and identify people who violate the corporate policy.

Like a NIDS, host-based IDS tools utilize signature analysis and anomaly analysis to identify attacks on the network. In addition to these detection methods, HIDS tools utilize file-integrity checks, logfile monitoring, and individual network monitoring to identify events of interest on the network.

Although most HIDS tools allow you to install them for use on a single host, organizations can reap significant benefit by correlating the data across multiple sensors with centralized alerting. This way, an analyst can get an overall picture of the events on the network, instead of individually visiting and examining each HIDS tool.

Unfortunately, the deployment of this technology is not as simple as buying a product and installing it. To get a handle on this complexity, you must develop a plan that dictates which hosts should be a target for deployment and in what order.

Also, provisions for the consistent and uniform application of updates are needed. Your core servers, perimeter servers, firewalls, web servers, DNS servers, and mail servers are the obvious first choices for

© 2017 Secure Anchor Consulting

deployment. Although it would be desirable to roll out host-based intrusion detection to all systems throughout the organization, the costs are usually prohibitive for commercial HIDS.

The other issue influencing the deployment decision is that the more frequently a host is reconfigured, the more false positives the IDS generates.

© 2017 Secure Anchor Consulting

Technet24

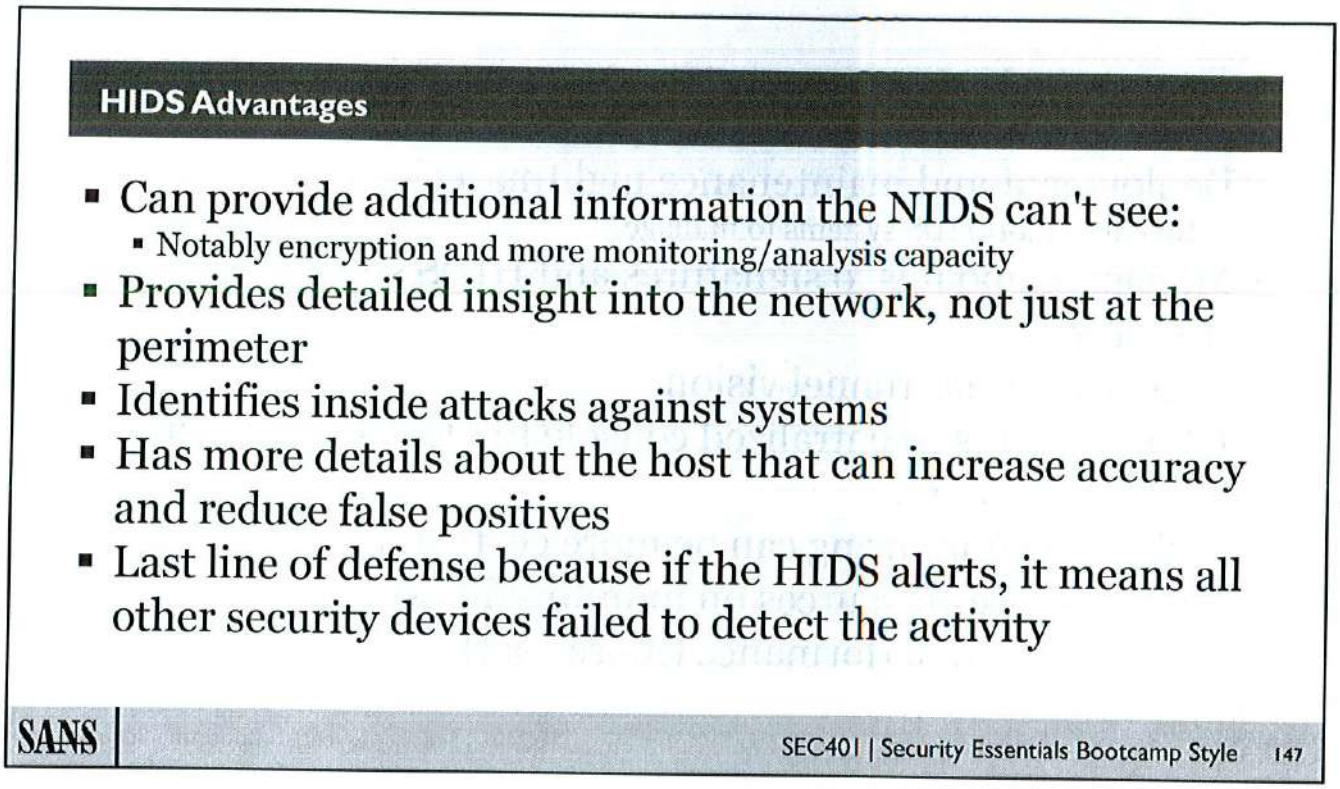
HIDS Network Monitoring

- Monitors network traffic to the host
- Typically listens on all interfaces
  - Ethernet, wireless, VPN
- Uses signature analysis to identify events of interest
- Much like a distributed NIDS
- Very powerful for not only monitoring inbound traffic but also outbound traffic
- Monitoring outbound traffic can be used to detect pivoting, internal reconnaissance, lateral movement and C2's

Network monitoring for HIDS systems works similar to that of NIDS systems. On a NIDS system, we utilize signature analysis to raise events of interest to the analyst for all the traffic that traverses the monitoring interface. With HIDS network monitoring, we monitor all the network traffic for the host on all interfaces. This has the advantage of being able to detect "backdoor" access to the host through wireless and VPN access.

HIDS network analysis is much like a widespread distributed NIDS model. Where we might distribute multiple NIDS sensors in a downstream network configuration to reduce the overhead on the NIDS, the network HIDS model takes this to the extreme by monitoring traffic at each node on the network.

© 2017 Secure Anchor Consulting



As shown, using HIDS systems offers many distinct advantages to the organization, including detailed analysis beyond what is capable by the NIDS. Notably, a HIDS system doesn't suffer from the same restrictions of the NIDS when processing encrypted traffic, because the HIDS can process the traffic after it is unencrypted by the host. Also, because the processing capacity of the HIDS is distributed so that it is only concerned with a single host, it can provide for extended analysis without adding significant overhead to the host it is monitoring.

The single biggest advantage to using HIDS systems is the additional insight it provides in detecting misuse of resources past the typical perimeter-monitoring mechanisms. Attacks against internal host resources are increasing with the increased worm activity that is introduced from a trusted user, attacks over the wireless network, or attacks over VPN tunnels from compromised systems. Using HIDS adds another measure of detection that is not easily accomplished with the use of traditional NIDS tools.

© 2017 Secure Anchor Consulting

Technet24

HIDS Challenges

- Deployment and maintenance nightmare:
  - Imagine 10,000 IDS systems to manage
- Managing updates to signatures and HIDS software can be complex
- Each sensor has tunnel vision
- HIDS requires a centralized console to identify trends and wide-scale events
- Full HIDS monitoring can be more costly than NIDS
- HIDS requires resources on monitored hosts and can impact system performance (more cost)

As with NIDS tools, the use of HIDS proposes several challenges that warrant careful planning and consideration. Even though a HIDS system might generate fewer alerts than a NIDS on a busy network, the management and handling of those alerts are a significant resource requirement for organizations, especially as the number of HIDS deployments increases throughout the network.

Some HIDS software offers a lot of features but requires frequent updates to software and analysis signatures. Managing the updates for software and signatures can be a difficult task because each update should be tested for quality assurance before deployment onto production resources. As with any updates to software, a risk is present because the update might cause instability on production equipment and must be evaluated and assessed before deployment.

Unlike a NIDS sensor, a single HIDS monitor suffers from a kind of tunnel-vision that comes from being limited to seeing attacks only on the local system. For example, a single HIDS monitor is unable to identify a wide-scale attack against multiple systems and is, therefore, unable to generate an alert for the analyst to convey the urgency of the event.

Fortunately, most commercial HIDS tools support a centralized monitoring console to identify attack trends and wide-scale events across multiple sensors. This lets the analyst perform analysis and assess the risks of events across all the monitoring systems that are available instead of narrowly focusing on a single event.

Although a limited deployment of HIDS tools on critical resources can be less expensive than a NIDS deployment, a wide-scale deployment of HIDS tools can be costly. Not only is the capital expense of software acquisition more significant for a wide-scale deployment of HIDS, but the associated costs for centralized monitoring stations, disk storage for all the alerts that are generated, and operational costs (maintaining the software, deploying updates, and so on) can be high.

Contributing to the overall cost of HIDS is the impact on available resources for servers and workstations that are monitoring for events. Although many vendors require "minimal" resources for their products, the

© 2017 Secure Anchor Consulting

aggregate cost of this overhead can reduce the life expectancy of servers and workstations. Organizations that have planned for a three-year amortization of hardware might be forced to shorten the lifecycle of hardware usefulness to meet the demands of improvements to HIDS software. This contributes to the cost of HIDS deployments, but is difficult to calculate because it is difficult to predict what the resource requirements for your selected HIDS product will be in future versions.

Finally, one of the challenges with HIDS is simply what to do with all the data that is generated by HIDS. Many organizations who deploy HIDS tools find they do not have the resources available to respond to all the events generated. Storing past events for future event correlation is a good idea, but the disk requirements can easily expand into requiring multiple terabytes to store all of the alerts gathered for a large deployment. Organizations need to clearly define their goals before deploying a HIDS solution, to clearly define the expected return on their investment.

© 2017 Secure Anchor Consulting

Technet24

Developments in HIDS and Recommendations

- Monitoring change at the application-level
- Protecting your website with HIDS
- Appliance platform support
  - File integrity monitoring for networking devices
- Moving from HIDS being a self-contained solution with a sensor and monitoring station – to HIDS becoming another data feed for the SIEM
- HIDS solutions are morphing into HIPS
  - Logical progression for mixed functionality coming together

This slide covers some of the recent advances in HIDS technology and how they impact organizations that are planning or have deployed HIDS systems.

More recent advancements in change monitoring tools permit an organization to monitor for change at the application level, which is often more beneficial and prone to fewer false positives. Newer application-integrity applications are able to monitor for changes at a higher-level, including monitoring for changes to a specified Windows group membership. These tools are also adapting themselves to integrity assessment of tools like database applications. By monitoring the contents of a database lookup table, for example, the application-integrity application might be able to detect changes that were the result of a SQL injection attack.

For many years, HIDS technology was available only on traditional host operating systems, such as UNIX, Linux, and Windows. Unfortunately, this isn't a comprehensive list of targets for an attacker; it is common for hackers to attack devices, such as routers, switches, and other infrastructure devices.

Fortunately, HIDS technology has been developing the ability to monitor the configuration and status of these networking appliances. The majority of these HIDS devices work from a centralized management server, polling configured devices for information about the configuration, status of cards and peripherals, software versions, running processes, and more. After establishing a baseline of operation, the appliance-HIDS generates alerts when unauthorized changes are made or if there are events of interest as defined by the analyst.

It is important to note that the line between HIDS software and host-based intrusion prevention software (HIPS) is becoming blurry. Personal firewall products stop some attacks by filtering traffic on the workstation and can be configured to perform centralized reporting for analysis. Antivirus products are starting to realize their contributory relationship to HIPS products by stopping a significant number of attacks from malware.

© 2017 Secure Anchor Consulting

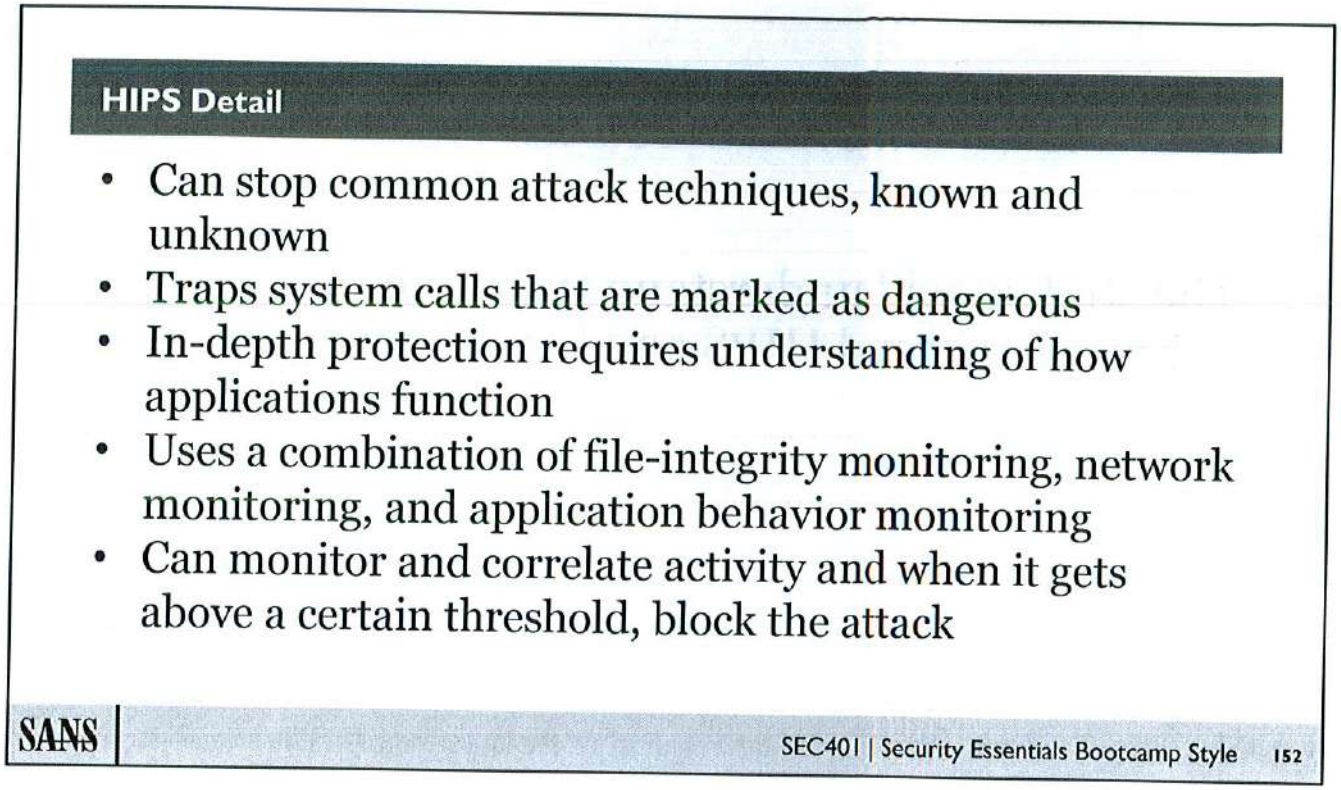
HIPS Overview

The student will understand the technologies and techniques behind HIPS and how it can be applied

This page intentionally left blank.

© 2017 Secure Anchor Consulting

Technet24

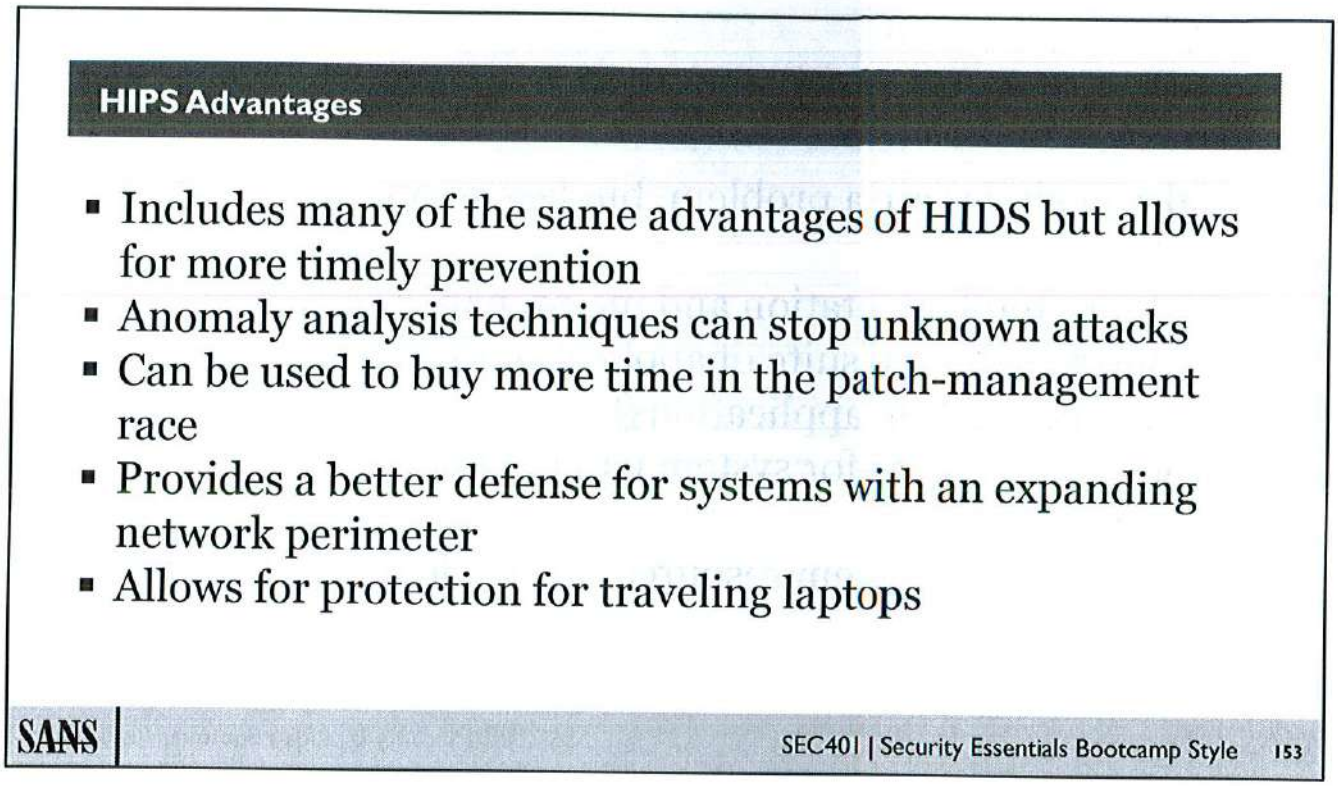


Let's investigate the technology supporting HIPS products. One of the major benefits of HIPS technology is the ability to identify and stop known and unknown attacks. This functionality lets enterprises have a wider window to deploy patches to systems because already deployed HIPS software is able to prevent common attack techniques.

To be effective at stopping attacks, HIPS software uses a technique called "system call interception," which is similar to what antivirus vendors have been doing for years. The HIPS software inserts its own processes between applications accessing resources on the host and the actual OS resources. This way, the HIPS software has the capability to deny or permit those requests based on whether the request is identified as malicious or benign.

A discussed, HIPS tools use a combination of signature analysis and anomaly analysis to identify attacks—this is performed by monitoring traffic from network interfaces, monitoring the integrity of files, and application-behavior monitoring.

© 2017 Secure Anchor Consulting



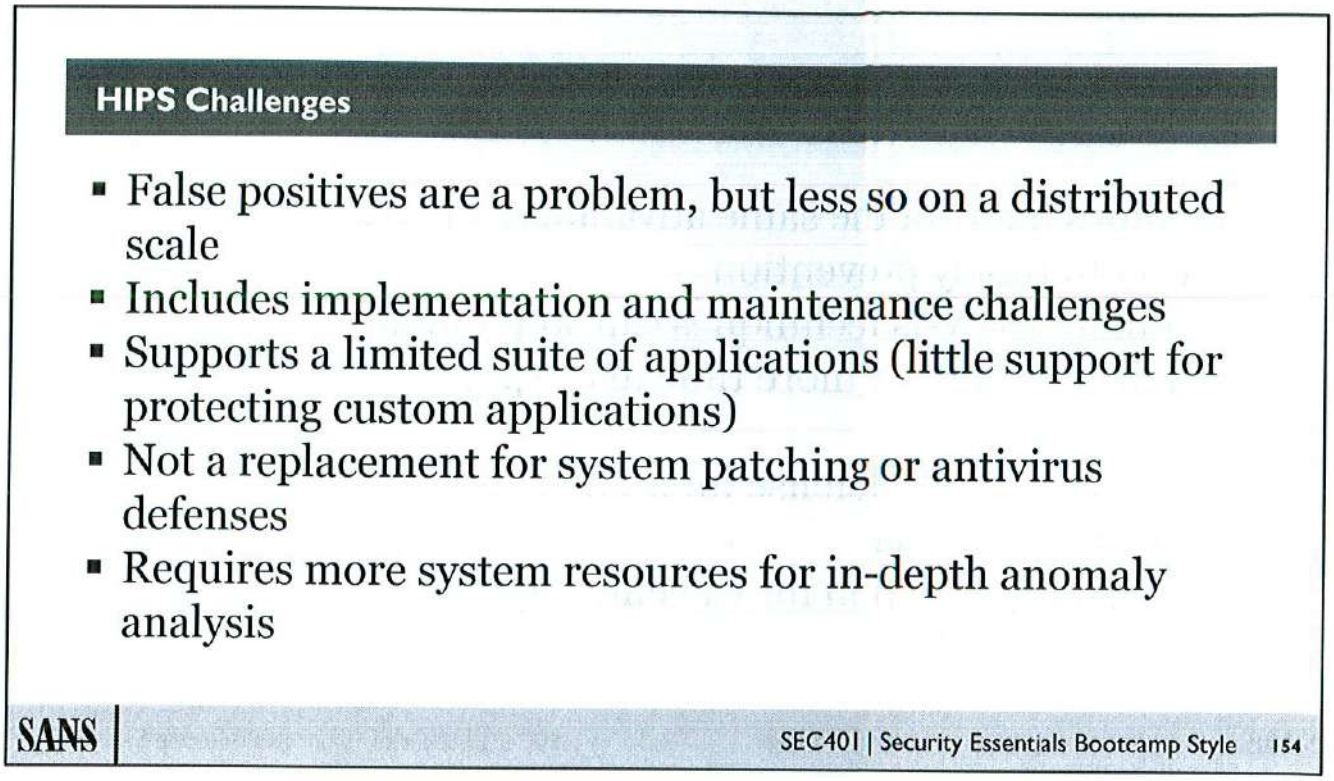
Now that we have a better understanding of how HIPS software functions and what it can do, let's look at the advantages of using HIPS.

Fortunately for us, the use of HIPS software includes nearly all the advantages of HIDS software. Identifying unauthorized change to files, monitoring network activity, and the ability to see the results of network-encrypted traffic are all advantages to HIPS software. The added benefit for HIPS, of course, is the capability to stop attacks from being successful. This is a welcome advantage for many organizations that struggle with patch-management challenges and the short window of time between when a vulnerability is announced and when it is actively being exploited. More and more organizations are implementing HIPS technology as this window rapidly continues to decrease.

Organizations are further challenged with an expanding network perimeter. Not too many years ago, we had to worry only about attacks from our Internet connections; now, attacks come from wireless networks, VPN connections, malware introduced by traveling users to our networks, and so on. HIPS software provides a better method of defending our perimeter when distributed throughout the enterprise than traditional tools allow.

© 2017 Secure Anchor Consulting

Technet24



IPS is not a silver bullet and, therefore, has its fair share of challenges. Plaguing HIPS deployments are implementation and maintenance challenges—testing updates, deploying updates, troubleshooting updates, and so on.

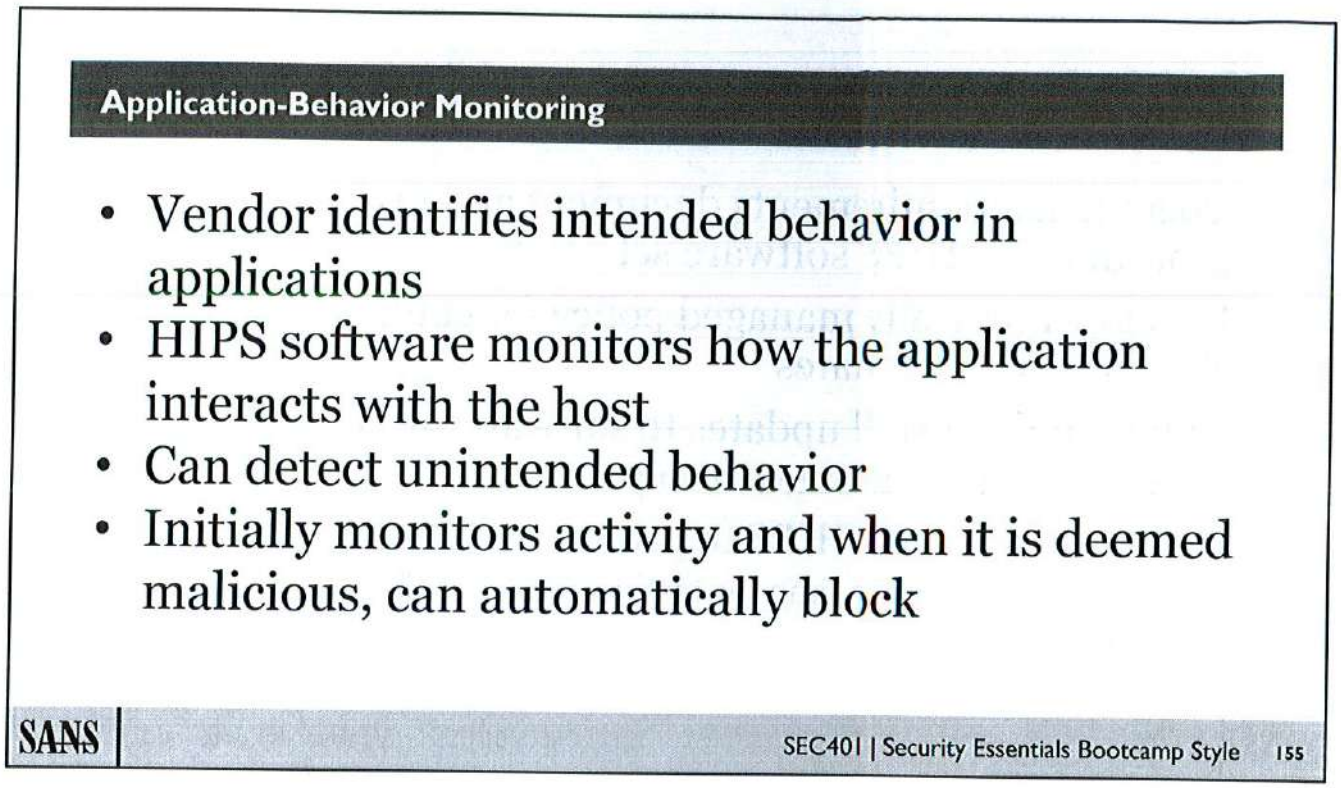
The capability to detect unknown attacks is a big advantage for IPS technology, but it is often tied to specific application functionality, such as IIS, Apache, or Exchange. The capability to monitor for anomalous behavior from applications is limited to those applications that are selected by your vendor, with limited support for protecting custom applications. Hardening operating systems and secure coding practices are still a good idea for protecting custom application software.

Despite the ability for HIPS software to identify and stop attacks, it is not a replacement for regular system patching or antivirus defenses. IPS software can still have weaknesses that have not been discovered and used to exploit this technology. It is best to use HIPS software as another piece of defense for your organization's security.

With all the advantages and detection techniques offered by HIPS software comes the additional burden of processing requirements on servers and workstations. This contributes to the total cost of ownership (TCO) of HIPS software, possibly reducing the lifecycle of your current server and workstation investments. Expect HIPS software to utilize between 10% and 20% of available CPU and memory resources, depending on configuration and analysis options.

Finally, the need for a management console to manage HIPS software throughout the organization is obvious, just as many organizations do to manage antivirus software updates and signature data files. Vendors are struggling with the extensibility of managing large numbers of nodes from a management console. If you are planning a large HIPS deployment, expect to make multiple investments in management consoles and the labor to replicate the management burden across multiple HIPS groups.

© 2017 Secure Anchor Consulting



Application-behavior monitoring is a feature of HIPS software where a manufacturer selects a supported application and records the intended functionality of the application in normal use. For example, if a vendor provided application-behavior monitoring for Microsoft Word, it would record how Microsoft Word interacts with the operating system and other applications, identifying all the product functionality. After collecting all the data about how an application should work, the vendor creates a database that details the functionality of the application to feed to the HIPS software. Once installed, HIPS software identifies and monitors the use of the supported application. If Microsoft Word opens a file from the file system and prints the document, the HIPS software would recognize that as intended functionality. If Microsoft Word starts parsing through each contact in the Outlook Contact Book to send repeated e-mails to each recipient, the HIPS software could recognize that as anomalous activity and shut down the application, generating an alert for the analyst.

Another example of application-behavior monitoring is on a web server product. If the HIPS software sees a request for GET /index.html, it would recognize that as intended functionality and let the web server respond to the request. If the HIPS software sees a request for ../../../../../ repeated 100 times, it could recognize the request as unintended functionality for the application and stop the request from being delivered to the application.

In practice, application-behavior monitoring is difficult to get right because applications are constantly changing functionality with updates and new releases. Most vendors are developing hybrid solutions that utilize a combination of application behavior monitoring and anomaly analysis, using a specified list of anomalous events that should not be allowed on the system.

© 2017 Secure Anchor Consulting

Technet24

HIPS Recommendations

- Document a requirements document and testing procedure for HIPS software selection
- Develop a centrally managed policy for controlling HIPS client rules and updates
- Don't blindly install updates to software without testing
  - Don't rely on the vendor to test for you
- Don't rely solely on HIPS to protect systems
  - Use them to buy more time for testing patches before a company-wide rollout

Here are some recommendations to keep in mind when evaluating or planning a HIPS deployment.

Carefully evaluate vendor products in a lab and production environment to ensure that they deliver the desired functionality without generating false positive detects. If a vendor's product requires significant troubleshooting and tweaking to get it working properly, record the time spent on this effort and add it to the TCO calculation for each application you want to use on hosts protected with the HIPS software.

Identify who should be responsible for managing updates to HIPS software and how often the software should be updated. Include information about how the organization should react to updates based on new Internet threats. Having this policy in place before a new attack threatens your organization will impact how well an organization can leverage the HIPS technology.

Despite the claims from manufacturers that they extensively test the updates to their products before deployment, they still can make mistakes and ship updates that render workstations and servers useless or severely impaired.

Establish a test environment for the supported workstation and server images for your organization and thoroughly test product functionality before approving the distribution of software.

Finally, use HIPS software to augment defense-in-depth techniques. Exclusively relying on HIPS software to protect systems is not a wise choice. Instead, use the extra time from the defenses provided by HIPS to carefully test and plan the delivery of patches to ensure that workstations are not vulnerable to the common exploits used by attackers.

© 2017 Secure Anchor Consulting

Developments in HIPS

- Vendors have some real-life scenarios to defend against
  - Proven protection against zero-day attacks
- Dynamic rule creation for custom applications based on observed behavior
- A new target for attackers
- Application shielding
- Like all security software, must be updated and properly maintained

This slide covers some of the recent advances in HIPS technology and how they impact organizations that are planning or have deployed HIPS systems.

HIPS products have seen unprecedented growth over the last few years. The organizations that have implemented HIPS are reporting their capability to protect against worms and other exploits, as well as zero-day threats. In the past, the typical cycle of when an exploit was released for a particular vulnerability was roughly 20 days. That number has significantly dropped, and we are now seeing exploits released the same day or before a vulnerability is announced. This leaves organizations vulnerable to attack by providing only a small window in which to patch their systems.

Zero-day protection features built into HIPS software is a concept vendors are employing when updating their products. Filters are now being written by studying not just the characteristic of the exploit, but the vulnerability itself. This is extremely beneficial in stopping exploits that have been slightly altered and attack the same vulnerability.

This feature has been introduced in many vendor products over the last year. Advanced Application Shielding essentially locks an application into a sandbox where it is not permitted to communicate with other applications. Many exploits tend to rely on an operating system's applications to launch attacks. If an application is locked down and prevented from communicating with other applications, you have essentially mitigated a big threat.

The popularity of HIPS software has started to get the attention of the attacker community, looking for ways to circumvent this technology. Some groups are focusing attention on attacking the management station and disabling HIPS software on clients throughout the organization centrally. Other groups are examining how HIPS software examines system calls and how the process might be circumvented.

© 2017 Secure Anchor Consulting

Technet24

Endpoint Security Summary

- With cloud-based environments and mobile devices, endpoint security is often the first and only line of defense
- Protecting the endpoint is critical to managing and minimizing pivot points
- Prevention and detection are both critical to stopping and minimizing the exposure of an attack

If your endpoint is compromised, how long would it take to detect it and how much damage would be caused?

In most cases, if an adversary is going to cause harm or damage to an organization, they are going to have to compromise the endpoint. There are a few exceptions with DOS attacks, but overall the adversary needs to get access to the endpoint so they can exfiltrate data or setup a pivot point to cause further damage. Therefore protecting and securing the endpoint is a key component of effective security.

While software security solutions have value, they must be installed on systems that are properly configured and managed. If a system has extraneous services and unpatched services, it will still be vulnerability regardless of the software that is installed. Remember an adversary only has to find one weakness to compromise your endpoint. Therefore proper hardening is required in addition to effective endpoint solutions. In addition, the endpoint solutions must also be properly configured and updated on a regular basis.

The bottom line is protecting and securing the endpoint goes a long way to minimizing the impact of a compromise.

© 2017 Secure Anchor Consulting

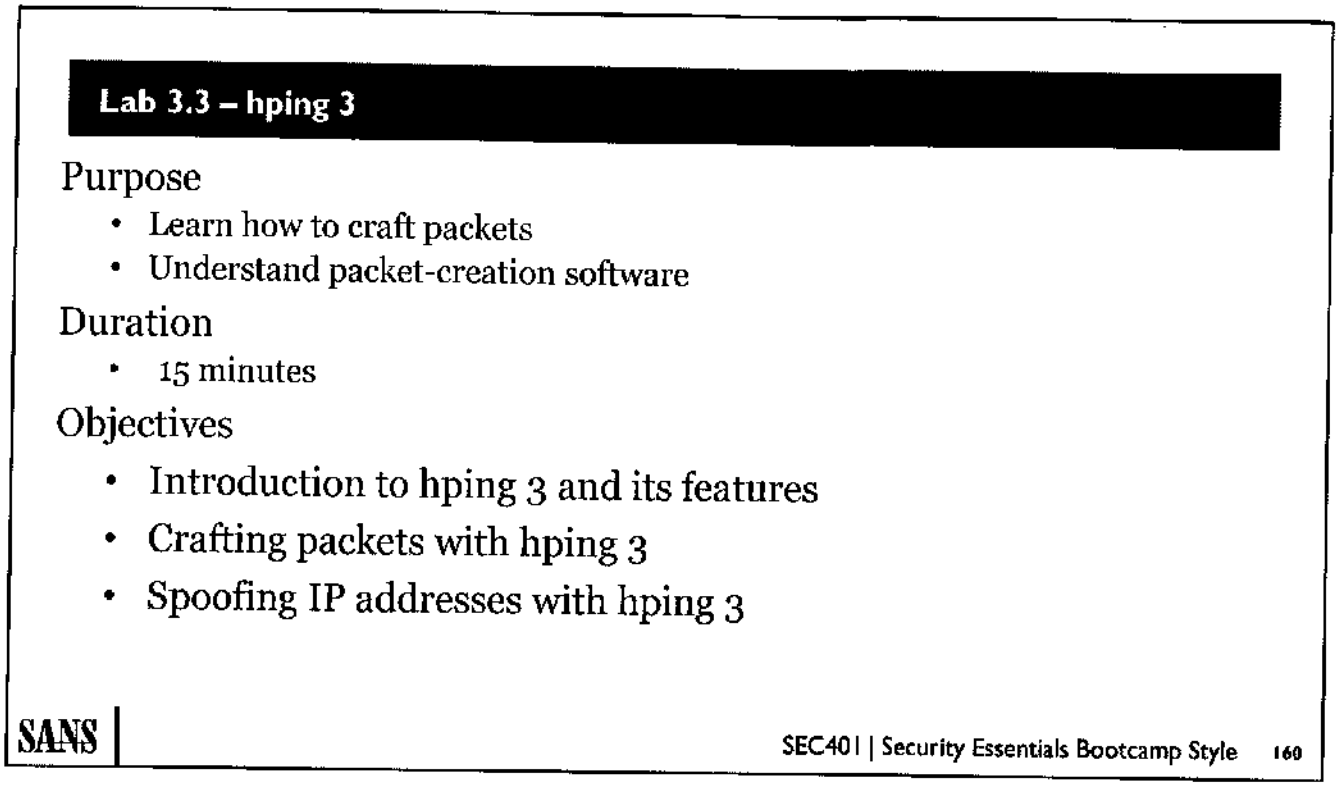
SANS

Lab 3.3 – hping 3

Another tool covered is hping 3. As stated on the official hping 3 website at http://hping.org, "hping is a command-line oriented TCP/IP packet assembler/analyzer." It can perform network scans similar to that of Nmap, but its real focus is on packet crafting to test firewall rules and technology, intrusion detection devices, advanced port scanning, and other uses. The lead developer of hping 3 is Salvatore Sanfilippo, although a lot of development has also been done by other contributors. Don't let the name fool you, hping 3 is not limited to ICMP by any means. It is fully capable of sending raw packets, TCP and UDP packets, as well as ICMP.

© 2017 Secure Anchor Consulting

Technet24



Purpose

- Learn how to craft packets.
- Understand packet-creation software.

Duration

- 15 minutes
- The estimated duration of this lab is based on the average amount of time required to make it through to the end. The duration estimate of this lab can decrease or increase depending on various factors, such as the booting of virtual machines, the speed and amount of RAM on your computer, and the time you take to read through and perform each step. All labs are repeatable both inside and outside of the classroom, and it is strongly recommended that you take the time to repeat the labs both for further learning and practice toward the GIAC Security Essentials Certification (GSEC).

Objectives

- Introduction to hping 3 and its features
- Crafting packets with hping 3
- Spoofing IP addresses with hping 3

© 2017 Secure Anchor Consulting

Lab 3.3 – Overview

Your objective for this short lab is to first take a look at some of the features of the hping 3 tool. Next, you'll craft various packets while sniffing them using tcpdump. This includes various TCP flag combinations. Finally, you will spoof the source IP address of a packet while examining it with tcpdump. We are only scratching the surface of the hping 3 tool in this lab. There are some advanced features of the tool, including fuzzing, which sends malformed packet data to a listening network service, and the capability to send files covertly over ICMP. For a great article written by Peter Kacherginsky, showing some of the more advanced usage, check out http://thesprawl.org/research/hping/. You will use your Kali Linux VM for this lab with your Windows VM as the target.

Your objective for this short lab is to first take a look at some of the features of the hping 3 tool. Next, you craft various packets while sniffing them using tcpdump. This includes various TCP flag combinations. Finally, you spoof the source IP address of a packet while examining it with tcpdump. We are only scratching the surface of the hping 3 tool in this lab. There are some advanced features of the tool, including fuzzing, which sends malformed packet data to a listening network service, and the capability to send files covertly over ICMP. For a great article written by Peter Kacherginsky, showing some of the more advanced usage, check out http://thesprawl.org/research/hping/. You use your Kali Linux VM for this lab with your Windows VM as the target.

© 2017 Secure Anchor Consulting

Technet24

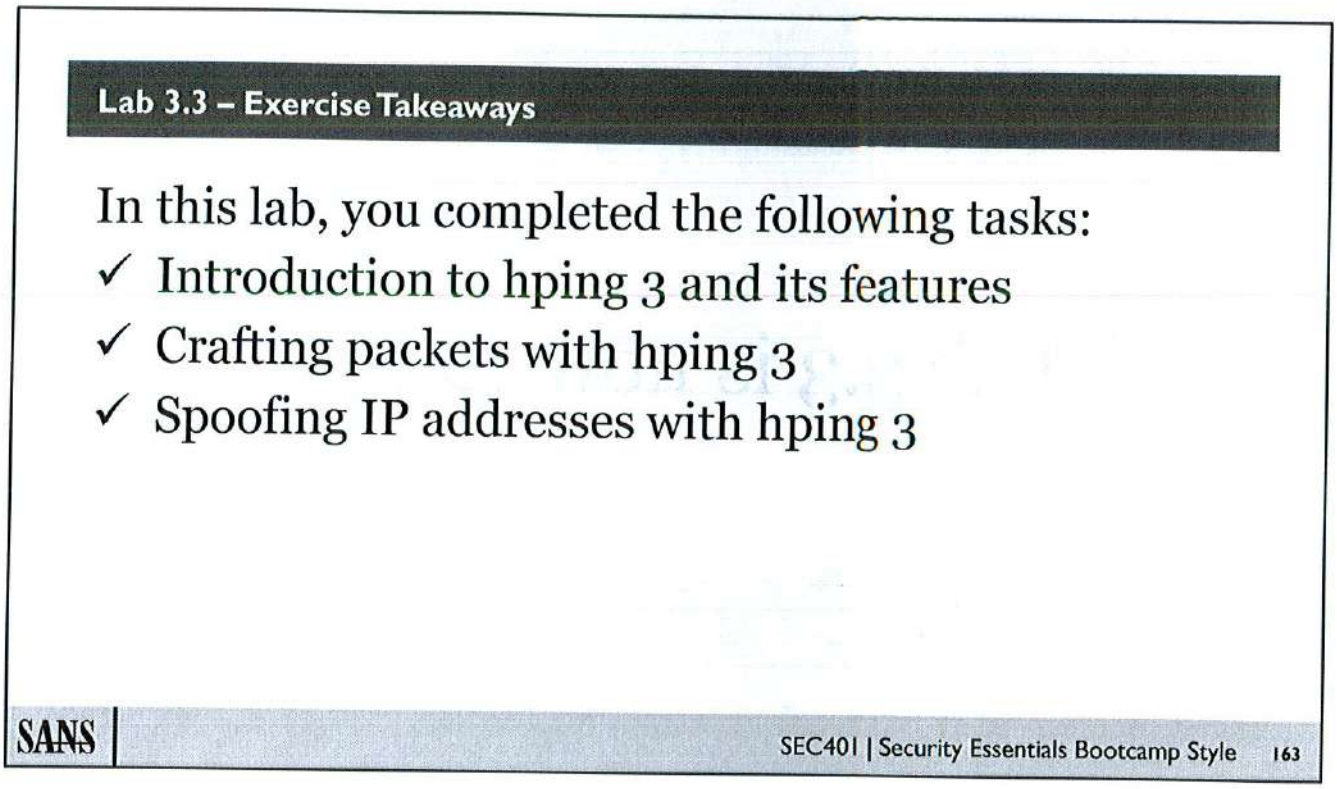
SANS

NOTE: Please open the separate Lab Workbook and turn to Lab 3.3

The instructor is going to introduce and go over the labs. Once the instructor is done, you will be instructed to work on the lab. If you have any questions, you can ask the instructor.

This page intentionally left blank.

© 2017 Secure Anchor Consulting



In this lab, you completed the following tasks:

- ✓ Introduction to hping 3 and its features
- ✓ Crafting packets with hping 3
- ✓ Spoofing IP addresses with hping 3

hping 3 is an easy-to-use command-line tool that allows us to craft packets. More advanced functionality can allow us to transfer files, create backdoors, and perform fuzz testing. It is commonly used to test devices such as firewalls and IDSs, as well as test access control lists on devices such as routers. We used the tool to perform a variety of tasks, such as spoofing source IP addressing and setting specific TCP flags in a packet.

© 2017 Secure Anchor Consulting

Technet24

SANS

Lab 3.3 is now complete

This page intentionally left blank.

© 2017 Secure Anchor Consulting

SANS

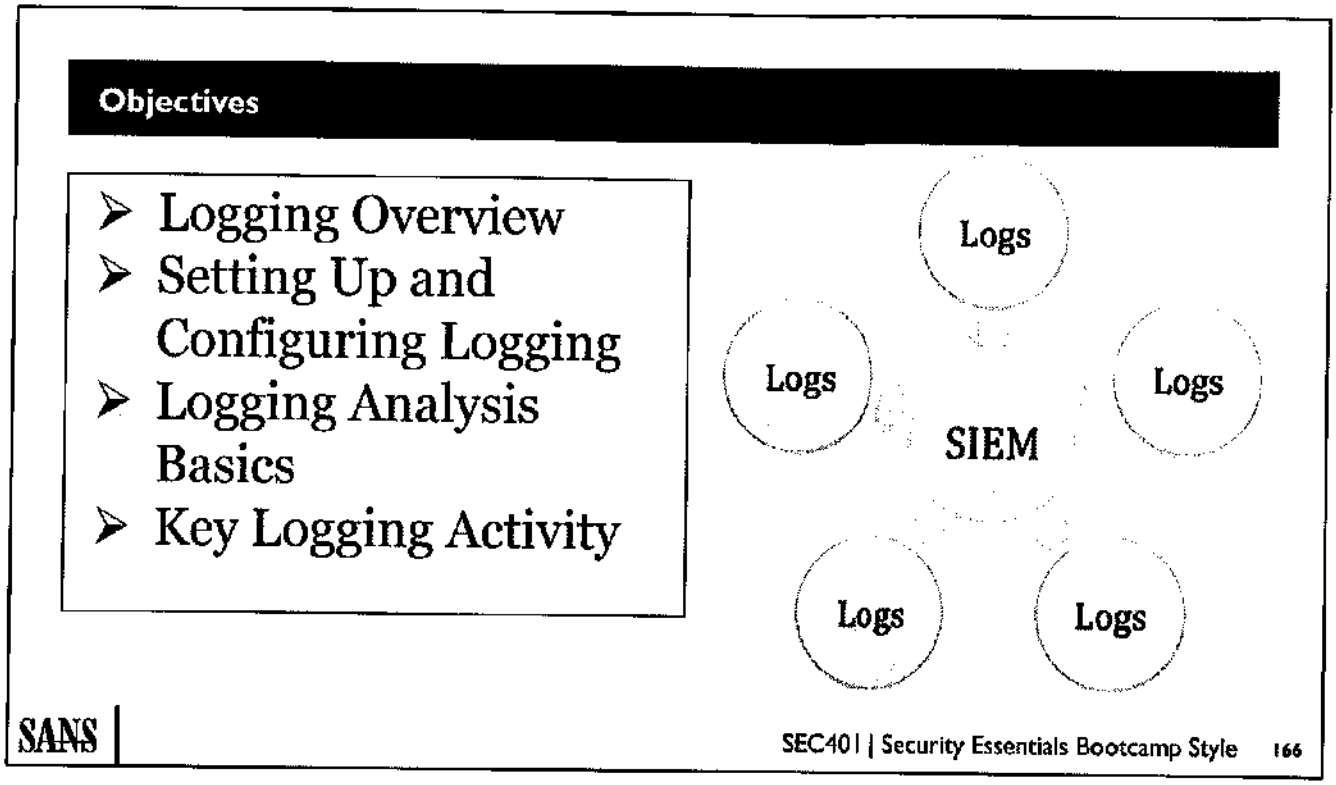
Module 16: SIEM/Log Management

Module 16: SIEM/Log Management

One of the challenges in cyber security is timely detection of attacks. Adversaries compromise systems for a long time before an organization detects it. While there are many reasons for this, one of the primary reasons is not having proper visibility to see what is happening. Logs contain events which ultimately is the evidence so you can track, find and monitor the adversary. Logs provide the optics that is required for timely detection. While turning on logging is important, it is just as important to make sure that you have the right logs. Too many organizations turn on logging, but it generates so many logs that they are not able to obtain any value. In this section, we will look at logging and the keys to a successful log management program.

© 2017 Secure Anchor Consulting

Technet24



In this section, we are going to cover the essential components of logging and how to properly manage it within your organization. The following are some of the topics that are going to be covered:

- Logging Overview
- Setting Up and Configuring Logging
- Key Logging Activity

Reference

1. What is a SIEM and What It Should Not Be? - http://binaryblogger.com/2015/12/01/what-is-a-siem-and-what-it-shouldnt-be/

© 2017 Secure Anchor Consulting

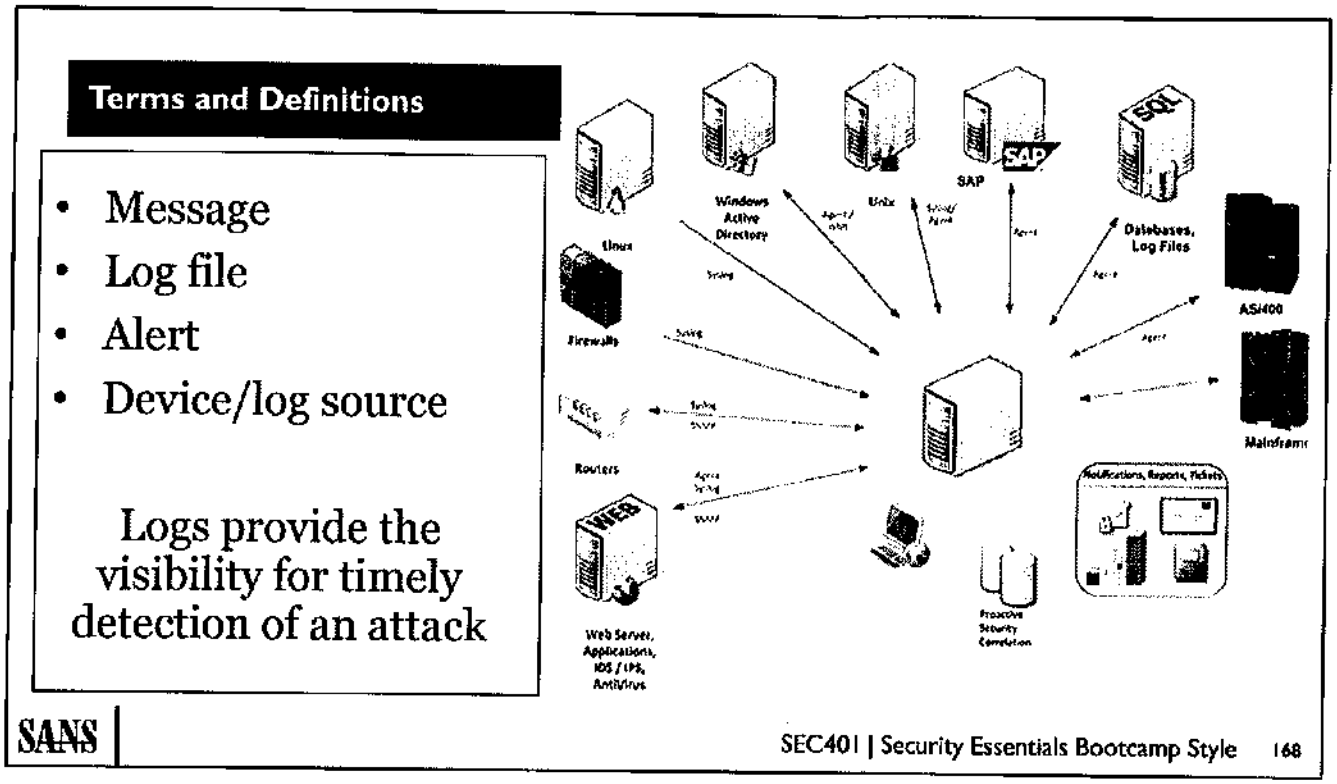
Logging Overview

The student will obtain a high-level understanding of what logging is and why it is important

This page intentionally left blank.

© 2017 Secure Anchor Consulting

Technet24



There are a lot of terms you will run across with logging: alerts, audit records, transaction logs, intrusion alerts, connection logs, system-performance records, other system messages, and so on. Every activity record on a system is a log.

Let's look at a few of the key terms:

- **Message**: Some system indication that an event has transpired
- **Log file**: Collection of the above log records
- **Alert**: A message usually sent to notify an operator
- **Device**: A source of security-relevant logs

There are plenty of other terms in logging:

- **Logging**: The act of creating a log
- **Auditing**: Sometimes used to name act of reviewing log records for the purpose of auditing
- **Monitoring**: Real-time or near-real
- **Alerting**: Sending a signal to an operator (the signal alert is treated as a type of log as well)
- **Debug traces**: Special logs specific to application debugging or tracing

Ultimately, logging is a key accountability mechanism across IT.

Reference

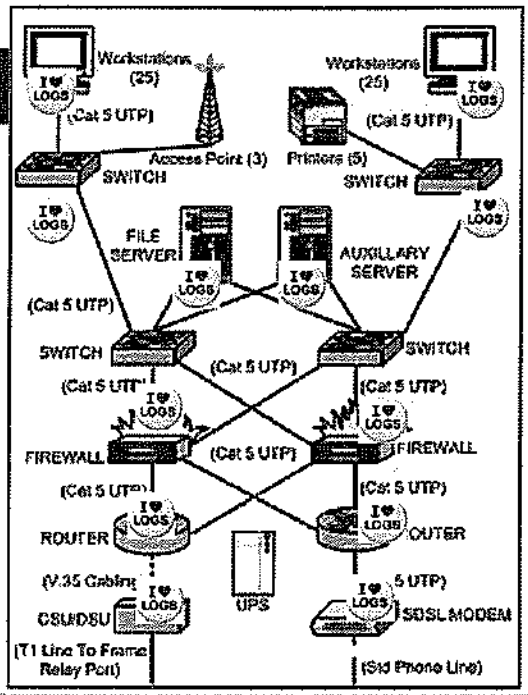
1. The CorreLog-Enabled Enterprise Architecture for SIEM - https://correlog.com/products.html

© 2017 Secure Anchor Consulting

Lots of Log Data Floating Around

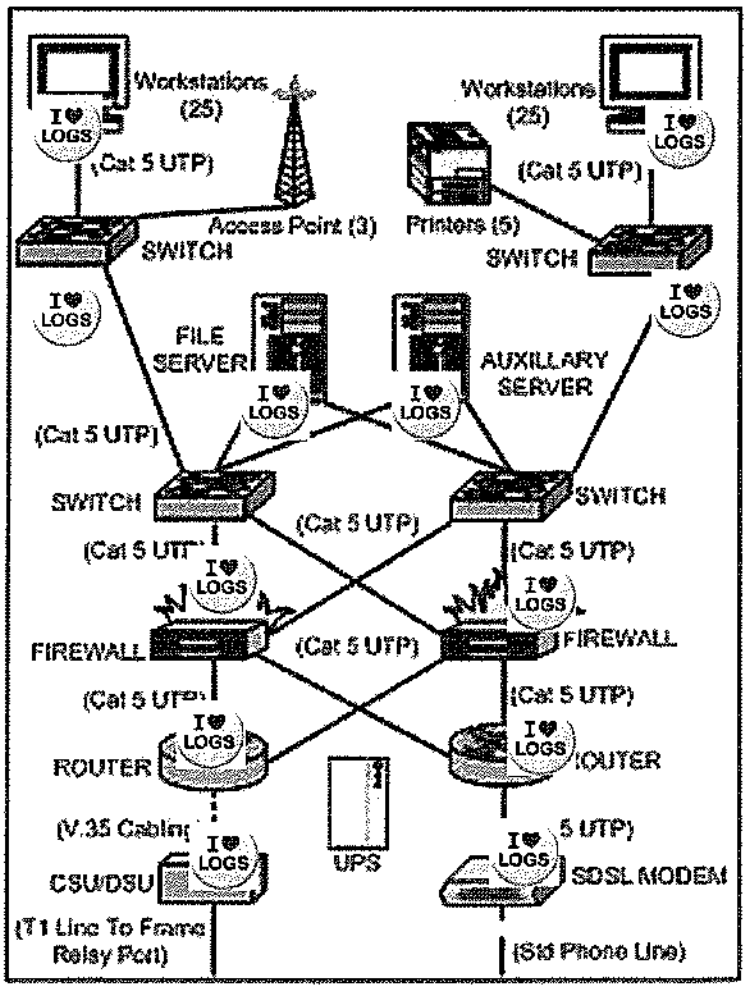
- Every machine has the capability to write logs
- More and more data to collect
- Each has a different format
- Logs can be overwhelming if it is not properly manage

They key to successful logging is to focus on quality of the logs not quantity



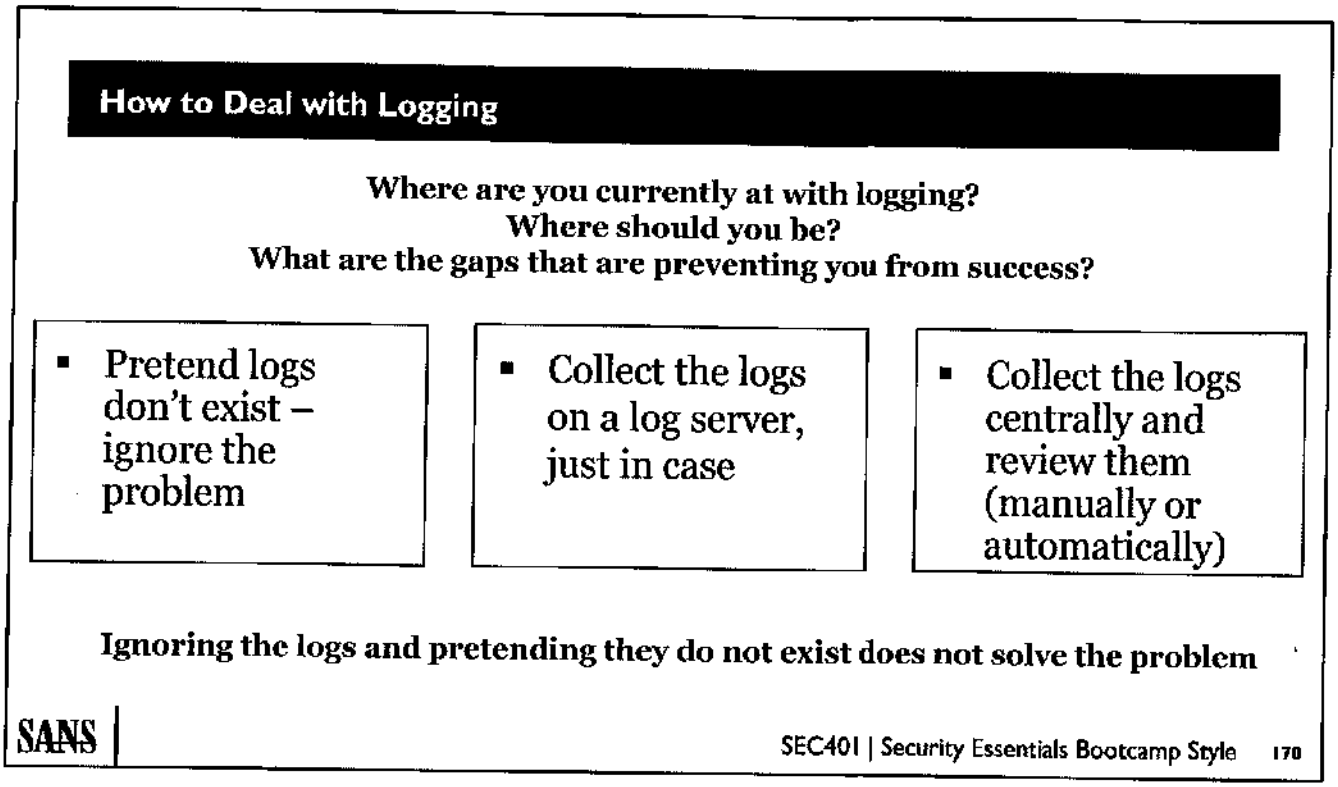
Here, we start painting the picture of the logging realm: lots of log sources, lots of logs, and more:

- **More and more data to collect:** This is due to both a growth in the log volume (people log more) and number of sources (people collect logs from more sources); each system, application, appliance, device may be writing a log or, often, more than one
- **Most systems will have its own language or log format:** Before the standards are developed and adopted, we have to face the multitude of log formats and transport mechanisms, as well as the level of recorded details
- **A large amount of enterprise data may be logs:** According to some estimates, a significant percent of all organization's data might be logs (as broadly defined).



© 2017 Secure Anchor Consulting

Technet24

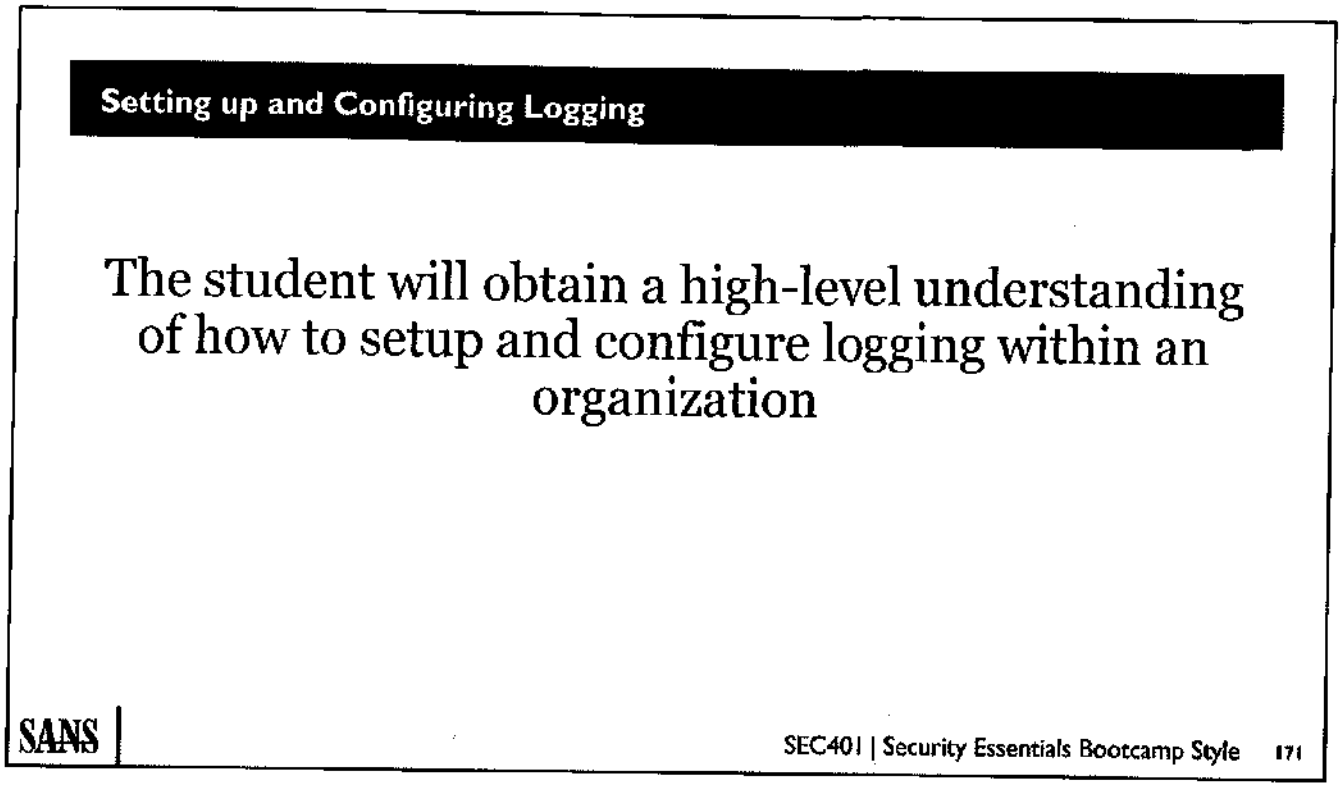


When people realize that there is so much log data floating around and then have a chance to think about it, they usually arrive at one of the following three choices:

- Pretend logs don't exist.
- Make an effort to collect the logs on a log server, just in case, but don't establish any process to review them.
- Make an effort to collect the logs on a log server and then review them (manually or automatically).

Keep these three choices in mind while we go through this class.

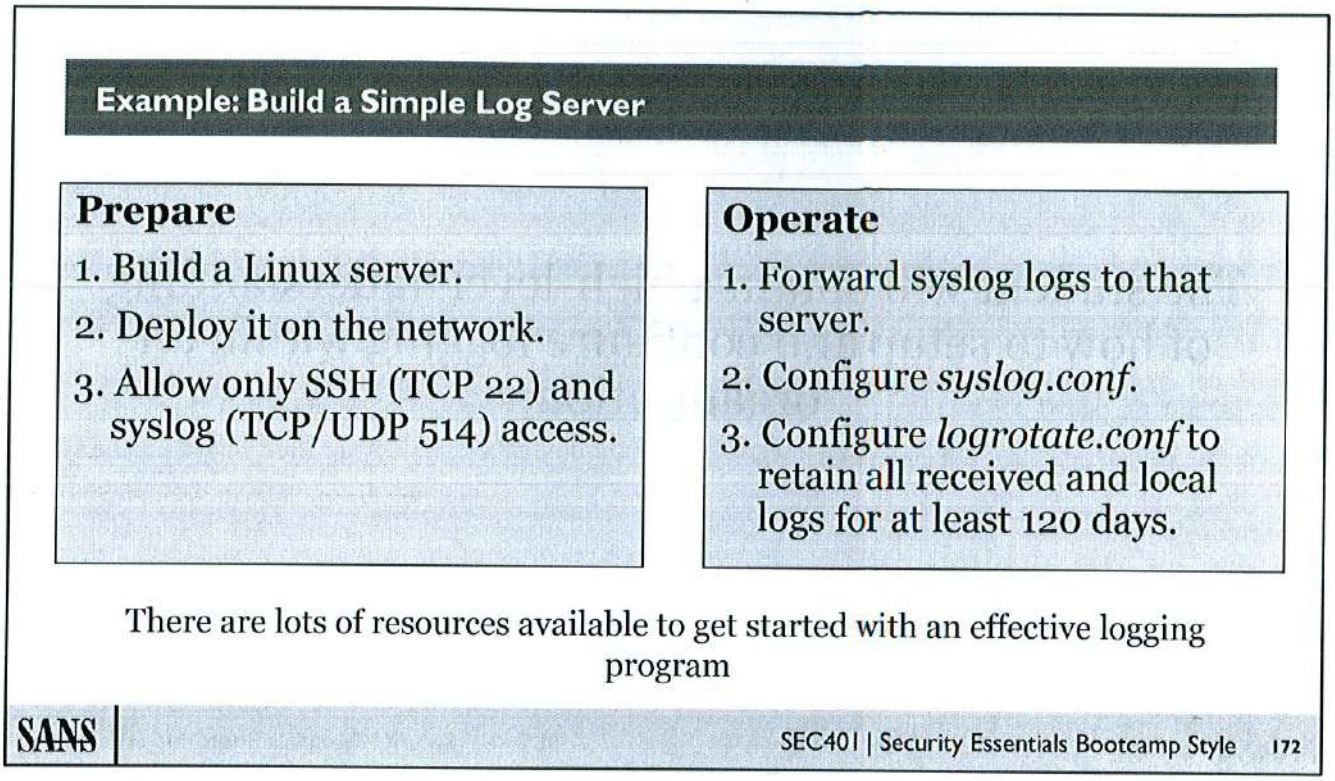
© 2017 Secure Anchor Consulting



This page intentionally left blank.

© 2017 Secure Anchor Consulting

Technet24



This slide shows and teaches you to build a simple log server:

First, prepare your environment:

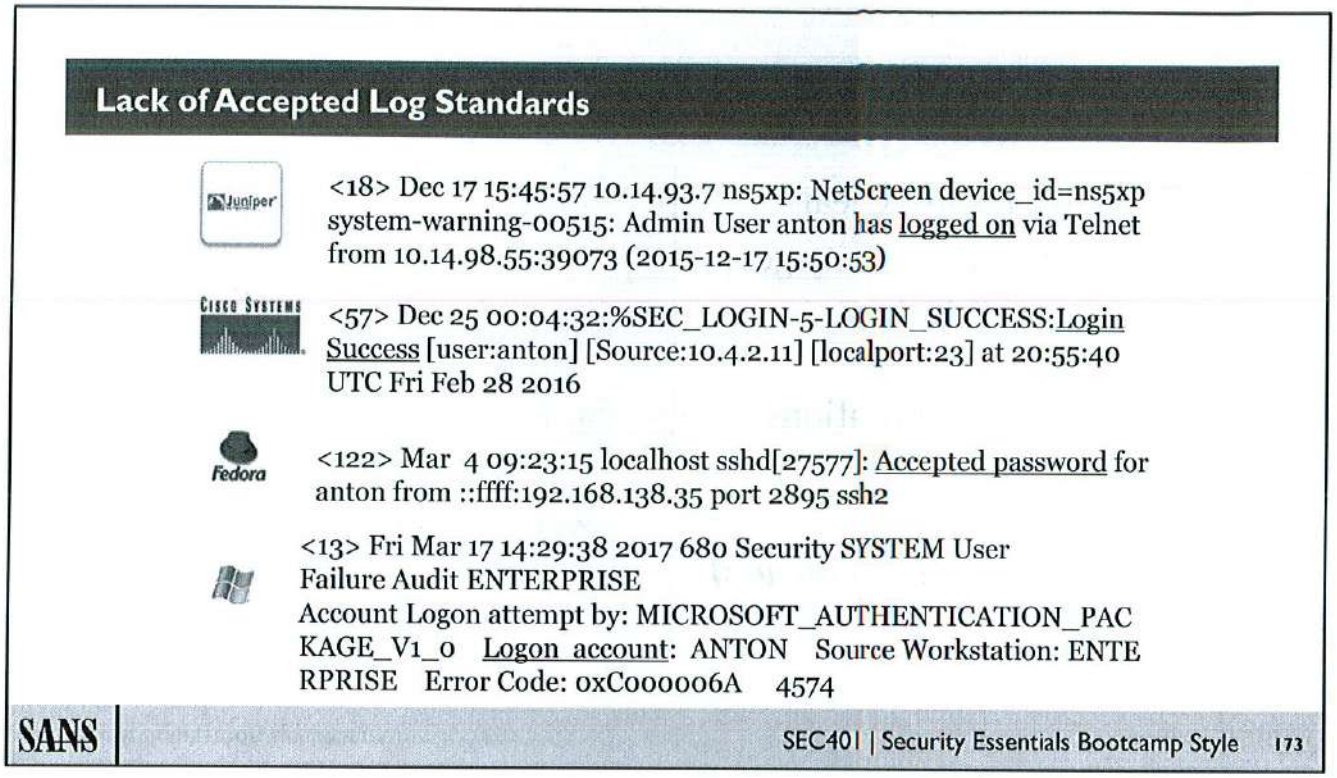
1. Build a Linux server; any reasonable hardware and any distribution (CentOS, Fedora, or anything else) will do.
2. Deploy it on the network where you can access it and make sure that it's close to the other server producing logs.
3. Only allow SSH (TCP port 22) and syslog (TCP/UDP port 514) access to the server.

Now you are ready to operate it:

1. Forward syslog logs from all UNIX and Linux devices to that server by modifying their /etc/syslog.conf files (on each log source).
2. Configure *syslog.conf* to send logs to files or other storage (on log server).
3. Configure *logrotate.conf* (or another log-retention management utility) on your new logserver to retain all received and local logs (on the log server).

Congratulations! You are on your journey to log management.

© 2017 Secure Anchor Consulting



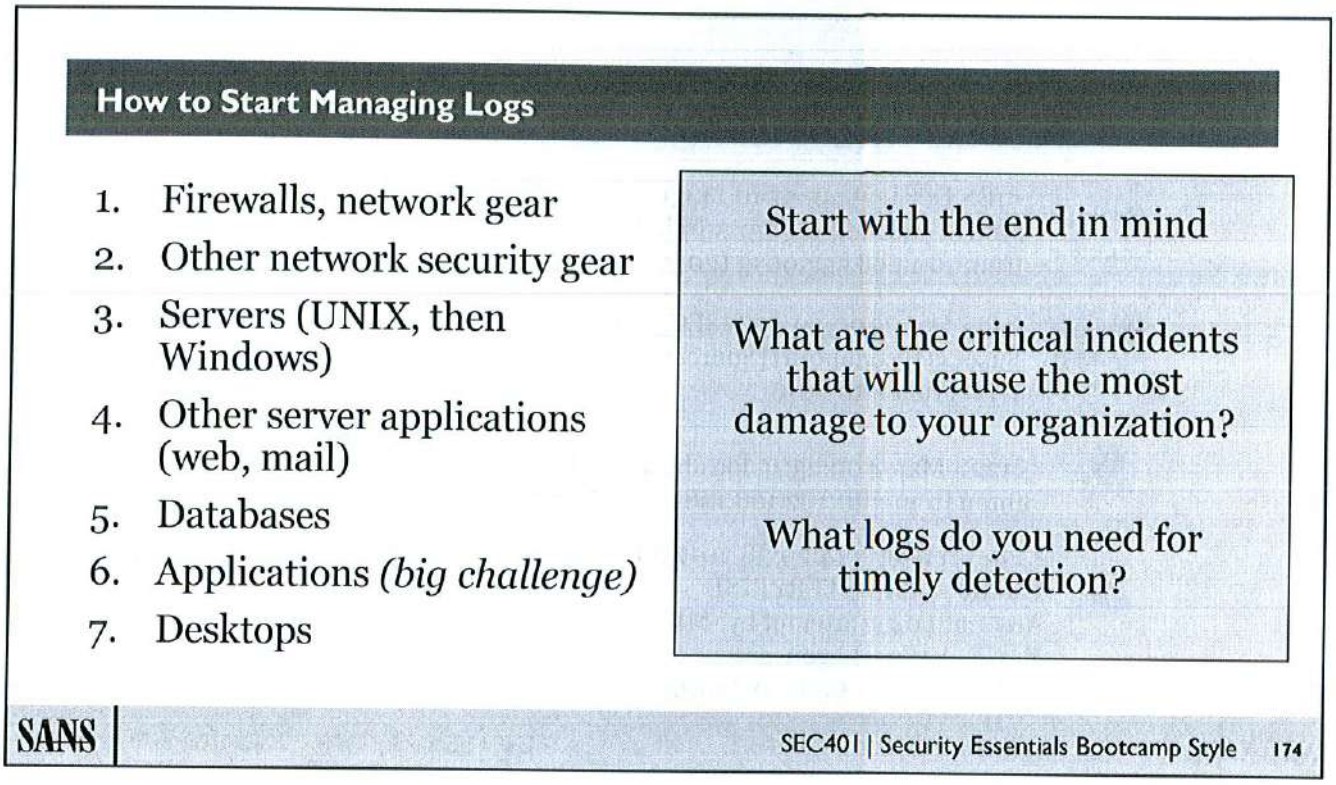
This slide shows a few of the so-called standard log messages (which are really not standard at all) from common log sources. All these logs from different sources are trying to represent a similar event (user logon) but are obviously in different formats and views. The slide is not meant to be readable, just to show how confusing the situation can be. As mentioned before, there is no standard for log formats at all, and this sad slide illustrates this point. All logs from different sources are trying to represent a similar event (user logon) in different formats and views.

What is interesting about it is that even a casual look at this slide reveals a few common things across the pictured logs: for example, source, destination, and protocol are all there for network logs. However, other fields, such as username, action, and so on, vary wildly.

To be fair, the Syslog RFC specifies a log standard format, of sorts. It consists of a header (with a timestamp, which itself varies) and a two-part message body. The problem is that this is far too general to fit all conditions AND it is not really being followed by syslog implementers. Current work on the Syslog RFC does have a chance to improve the uniformity of syslog messages by prescribing name=value pairs for the syslog messages. It remains to be seen whether it will be followed.

© 2017 Secure Anchor Consulting

Technet24



As is the case with how most companies deal with their logs, this is how you will likely progress as well, so understanding how to start managing logs is important. Progressing in this order takes into account both the need to analyze certain log types and typical available expertise.

Many firewalls log in standard syslog format, and such logs are easy to collect and review.

Reviewing Network IDS logs, for example, is extremely useful and is often a frustrating task because NIDSs would sometimes produce false positives. Still, NIDS log analysis often comes second after firewalls; the value of such info for security is undeniable and logs can, in most cases, be easily centralized for analysis.

Even though system administrators always knew to look at logs in case of problems, massive server operating system (both Windows and UNIX/Linux flavors) log analysis didn't materialize until more recently. Collecting logs from all critical (and many non-critical) Windows servers, for example, was hindered by the lack of agentless log-collection tools. On the other hand, UNIX server log analysis was severely undercut by a total lack of unified format for log content in syslog records.

Web server logs were long analyzed by the marketing departments to check on their online campaign successes. Also, most web server admins would review those logs. However, because web servers don't have native log-forwarding capabilities and most log files are stored on the web server itself, they often are not backed up and reviewed.

This material should give you pause: It is hard to justify why log management is still ordered this way given that we know that web-application attacks and database abuse are where most of the criminal activity happens. This order might not be the right order for today's reality, but it is still the most common order followed, because application logs and database logs are considered "too hard to deal with," at least initially.

© 2017 Secure Anchor Consulting

Key Points to Remember

- Logs are key organizational records
- Logs are produced by a huge number of components
- All logs look different since there is no log standard
- There is a certain logical sequence for tackling logs

SANS | SEC401 | Security Essentials Bootcamp Style 175

The key points to remember about logging are

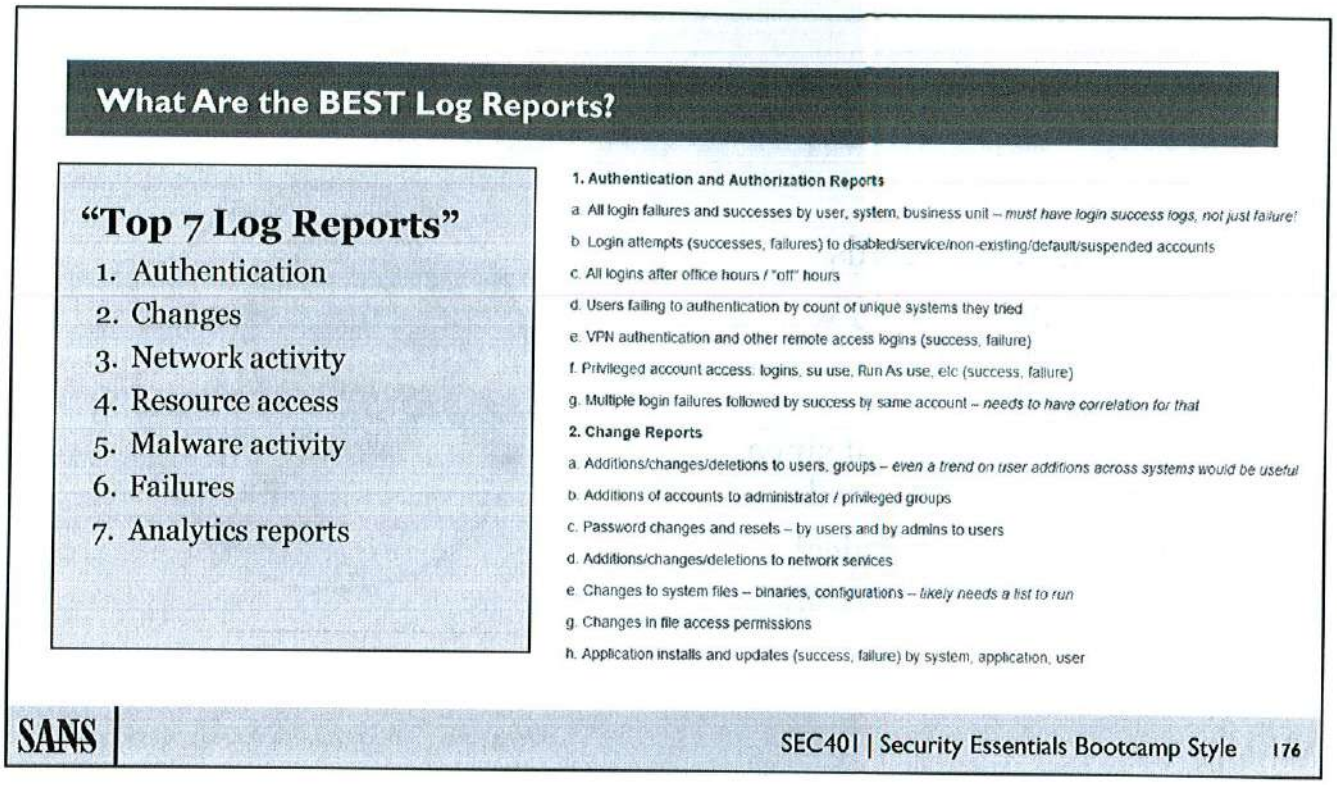
- **What are logs?**: Definition of logging terms that are in common use today
- **Where do they come from?**: In the near future, just about everything with a CPU will create logs (that we then need to understand and analyze)
- **What are they telling us?**: The type of information that is commonly logged: systems failures, authentication decisions, and so on
- **How to deal with logs?**: There is a certain logical sequence for tackling logs and the natural flow of log management

Reference

1. SIEM - http://www.telseccorp.com/siem

© 2017 Secure Anchor Consulting

Technet24



http://chuvakin.blogspot.com/2010/08/updated-with-community-feedback-sans_06.html

The following are the best logs to review:

1. Authentication and Authorization Reports

a. All login failures and successes by user, system, business unit (*must have login success logs, not just failure*)
b. Login attempts (successes, failures) to disabled/service/non-existing/default/suspended accounts
c. All logins after office hours/"off" hours
d. Users failing to authenticate by count of unique systems they tried
e. VPN authentication and other remote access logins (success, failure)
f. Privileged account access: logins, su use, Run As use, and so on (success, failure)
g. Multiple login failures followed by success by same account (*needs to have correlation for that*)

2. Change Reports

a. Additions/changes/deletions to users, groups (*even a trend on user additions across systems would be useful*)
b. Additions of accounts to administrator/privileged groups
c. Password changes and resets (by users and by admins to users)
d. Additions/changes/deletions to network services
e. Changes to system files, such as binaries and configurations (*likely needs a list to run*)
f. Changes in file access permissions
g. Application installs and updates (success, failure) by system, application, user

© 2017 Secure Anchor Consulting

3. Network Activity Reports

a. Log volume trend over days (*watch for both drops and increases in logging levels on systems*)
b. All outbound connections from internal and DMZ systems by system, connection count, user, bandwidth, count of unique destinations, hour of access (focus on "off hours")
c. Top largest file transfers (inbound, outbound) OR top largest sessions by bytes transferred
d. Web file uploads to external sites (*based on proxy logs*)
e. All file downloads by content type (exe, dll, scr, upx, and others) and protocol (HTTP, IM, and so on)
f. Internal systems using many different protocols/ports
g. Top internal systems as sources of multiple types of NIDS, NIPS, or WAF alerts
h. VPN network activity by username, total session bytes, count of sessions, usage of internal resources
i. P2P use by internal systems
j. Wireless network activity
    i. Rogue AP detection
    ii. Wireless network access by user
    iii. WIDS/WIPS alert activity

4. Resource Access Reports

a. General
    i. Access to resources on critical systems after office hours/"off" hours
b. Web
    i. Top internal users blocked by proxy from accessing prohibited sites (malware sources, pornography, and so on)
c. File
    i. File, network share or resource access (success, failure) (for specific audited resources)
d. Database
    i. Top database users, excluding known application access
    ii. Summary of query types, excluding known application queries
    iii. All privileged user access
    iv. All users executing INSERT, DELETE commands, excluding known application queries
    v. All users executing CREATE, GRANT, schema changes, and so on
    vi. Database backups
e. E-mail
    i. Top internal e-mail addresses by count of messages, byte volume
    ii. Top internal e-mail addresses sending attachments to public/hosted addresses
    iii. All e-mailed attachment content types, sizes
    iv. All internal systems sending mail, excluding known mail servers

5. Malware Activity Reports

a. All systems with AV events by user, system name, time trend
b. Detect-only events from antivirus tools (leave-alones)
c. All antivirus protection failures (crashes, unloads, update failures, and so on)
d. Internal connections to known malware IP addresses (*a public blacklist needed*)

6. Failures and Critical Errors

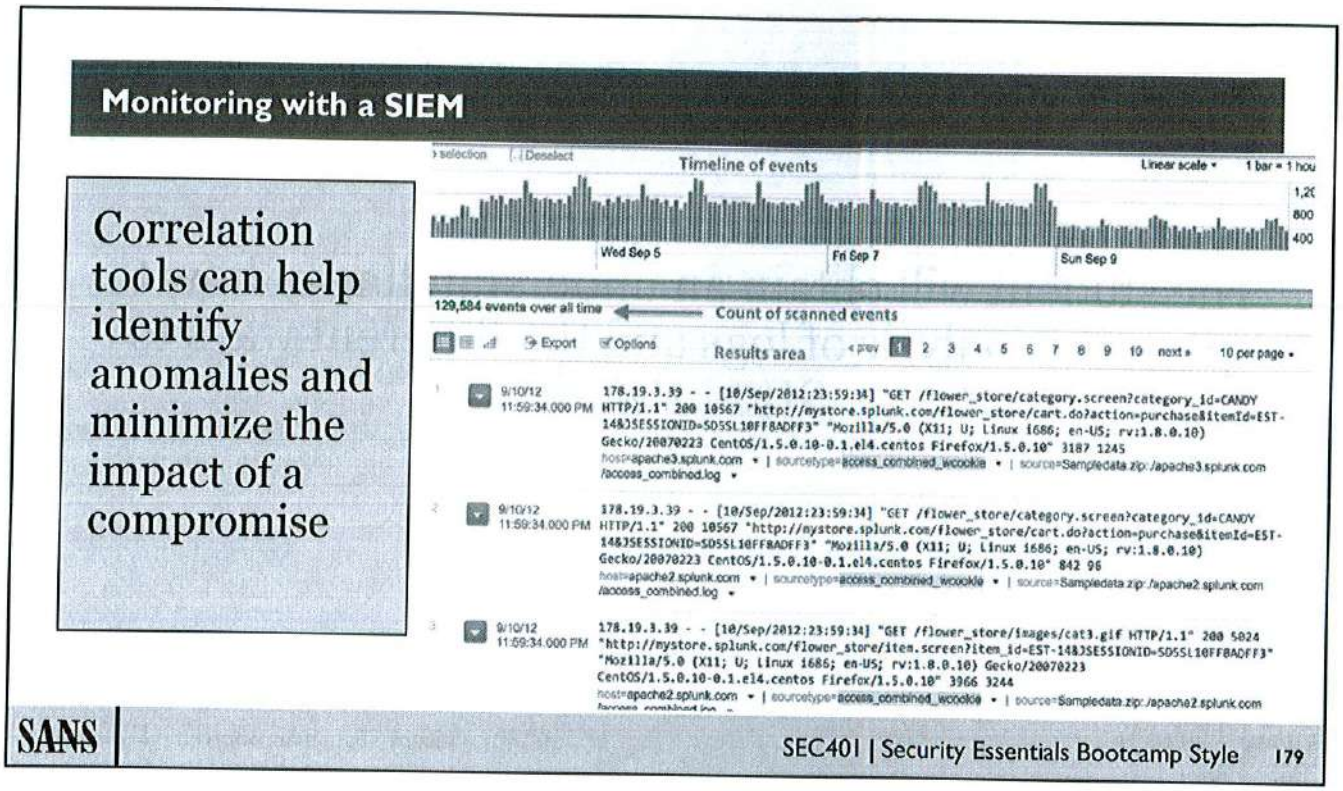
a. Critical errors by system, application, business unit
b. System and application crashes, shutdowns, restarts
c. Backup failures
d. Capacity/limit exhaustion (memory, disk, CPU, and so on)

© 2017 Secure Anchor Consulting

Technet24

7. **Analytic Reports, Mostly Using "Never Before Seen" (NBS)**
    a. (NBS) Log message types/event types
    b. (NBS) Users authenticating successfully
    c. (NBS) Sources that connected to systems using privileged accounts
    d. (NBS) Internal system connecting to external systems
    e. (NBS) External IPs connecting to NEW Entry Points (*not sure how to collect this*)
    f. (NBS) Ports accessed on internal systems
    g. (NBS) HTTP request types
    h. (NBS) Downloaded/uploaded content types
    i. (NBS) Query types on databases

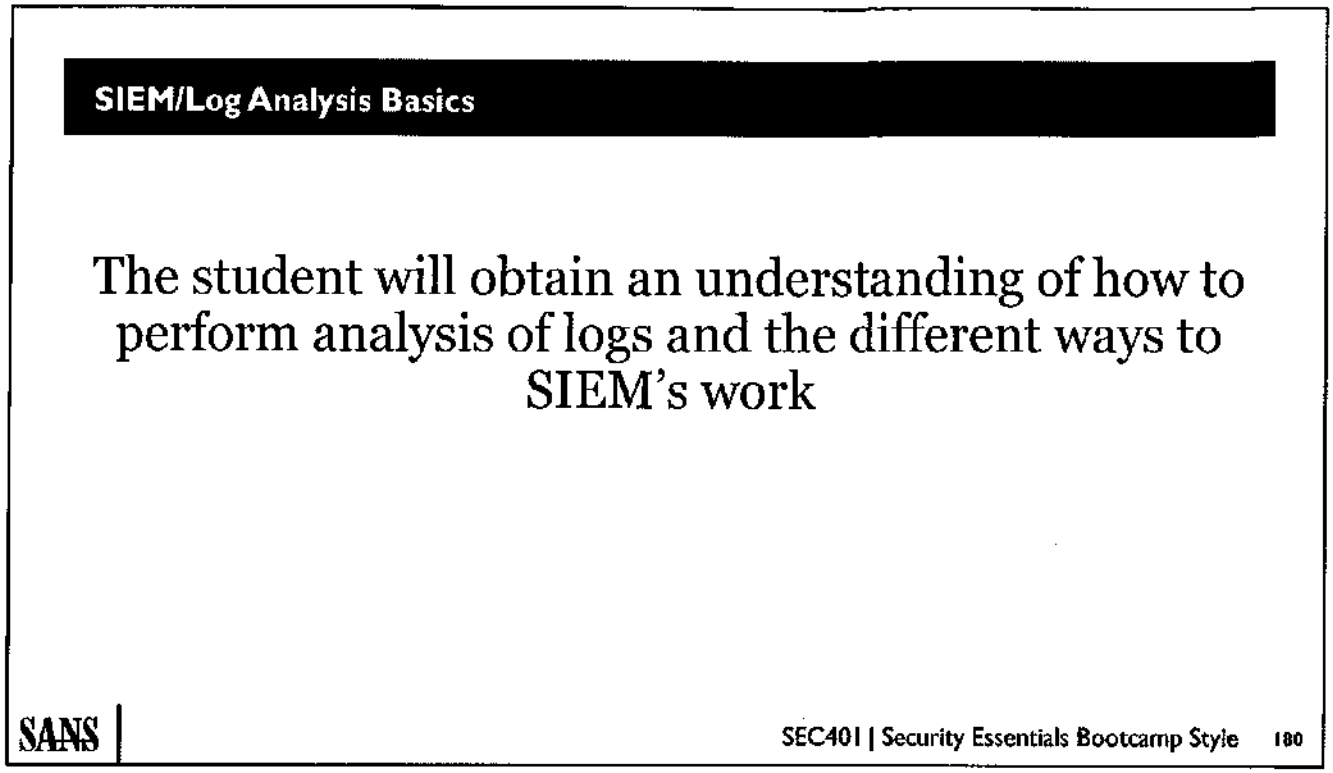
© 2017 Secure Anchor Consulting



Monitoring the performance of your application allows you to identify anomalous behavior, detect Denial of Service attacks, and ensure sufficient capacity to process user requests. By correlating the information across your web servers, firewalls, and other security devices, tools such as Splunk can help identify those anomalies. The key performance indicators to track for security purposes are latency (the time between making a request and seeing a result) and throughput (the number of items processed per time unit). Some specific attributes to measure are the latency and throughput of your network connections, page load times, application login, and transaction times. You first want to establish a baseline (what does the system look like under normal load?). This gives you something to compare to as utilization grows or when problems or incidents occur. Also, look for patterns and trends, such as how traffic varies with the time of day or day of the week. It can be beneficial to monitor the machine-load parameters, such as CPU or memory utilization of the different servers in the system; however, it is important to monitor your actual web performance and not just machine level load.

© 2017 Secure Anchor Consulting

Technet24

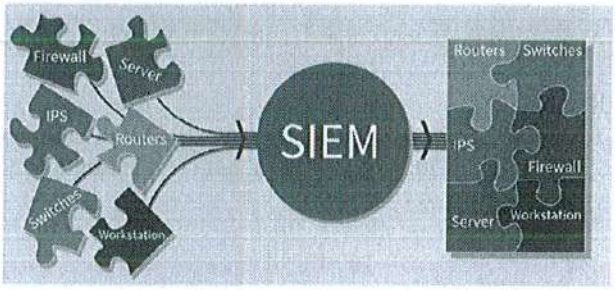


This page intentionally left blank.

© 2017 Secure Anchor Consulting

Select Log-Analysis Tools (SIEM)

- Log-collection tools
- Secure centralization
- Pre-processing
- Storage
- Analysis tools



There are many options when looking at log analysis. Let's review some of the tool choices:

Log-collection tools are numerous and solve the problem of getting logs from where they are produced to where we need them. Examples are syslog-ng (general purposes logging + reliable delivery + a lot of convenience features), kiwi, NTsyslog, Snare (for single machine Windows to syslog conversion), LASSO (for remote Windows to syslog mass conversion), DAD (all for Windows to syslog logging), Apache2syslog (for centralization of Apache logs via syslog), and more.

Secure centralization is the protection of logs in transit with encryption. Some examples include stunnel (SSL encryption), OpenSSH (with SSH), and IPsec VPNs offer this capability and can be used if log encryption is needed. However, many logging devices (such as network devices) might not support encryption "end to end," and, thus, this security measure becomes difficult.

Pre-processing is a task of converting logs from one format to another. For example, LogPP tool helps convert multi-line logs to a single line format.

Log storage is not a challenge when we are talking about megabytes, but multiple gigabytes and then terabytes present a non-trivial challenge. One can always store logs in flat files (such as all the standard files in /var/log/syslog). For an easy relational database option, MySQL is available. Designing your own storage is a choice needed for high-performance log processing.

Analysis tools: Many high power and popular tools exist in this space.

Some of the popular tool choices are

- SEC for rule-based correlation and analysis of logs in near-real-time; flexible but sometimes hard to use.
- OSSEC is a leading open-source tool for real-time log analysis; it is being actively developed and contains a large list of rules, usable out of the box.

© 2017 Secure Anchor Consulting

Technet24

- OSSIM is more ambitious projects to analyze and correlate logs; use them if your goals go beyond real-time alerting on logs.
- Swatch, logwatch, and logsentry are examples of literally hundreds of scripts that can look for specific strings in logs and then send an e-mail or other alert. Many people choose to write their own instead of using others, which leads to a proliferation of such tools.
- SLCT uses a simple clustering technique to make sense of stored logs; use it when you need to review a large amount of file logs.

Apart from these tools, there is always Perl, Python, and good old C compiler.

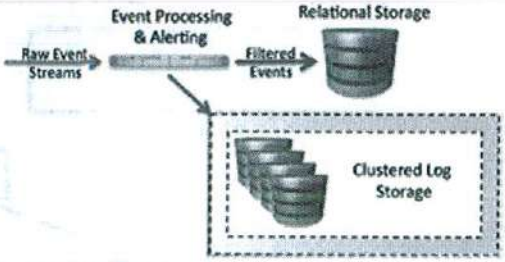
Reference

1. Benefits of Log Consolidation in a SIEM Environment - https://www.integritysrc.com/blog/122-benefits-of-log-consolidation-in-a-siem-environment

© 2017 Secure Anchor Consulting

How to Start Using the Tools

1. Collect logs
   Tools: Syslog-ng, standard syslog, & more
2. Store logs
   Tools: MySQL and others
3. Search logs
   Tools: grep, Splunk, LogParser, & more
4. Correlate and alert
   Tools: OSSEC, OSSIM, sec, nbs, logwatch, and others



Here, we learn how to start using the tool for taking control of your logs.

Start by collecting logs; use syslog-ng or whatever syslog variant is available on your systems. To combine these with Windows logs, use Snare or LASSO, which convert Windows logs to syslog.

Store logs in files (compressed or not) or in a database, such as open-source MySQL.

To start peeking at logs, use search logs like free "grep" or "Splunk" (previously mentioned).

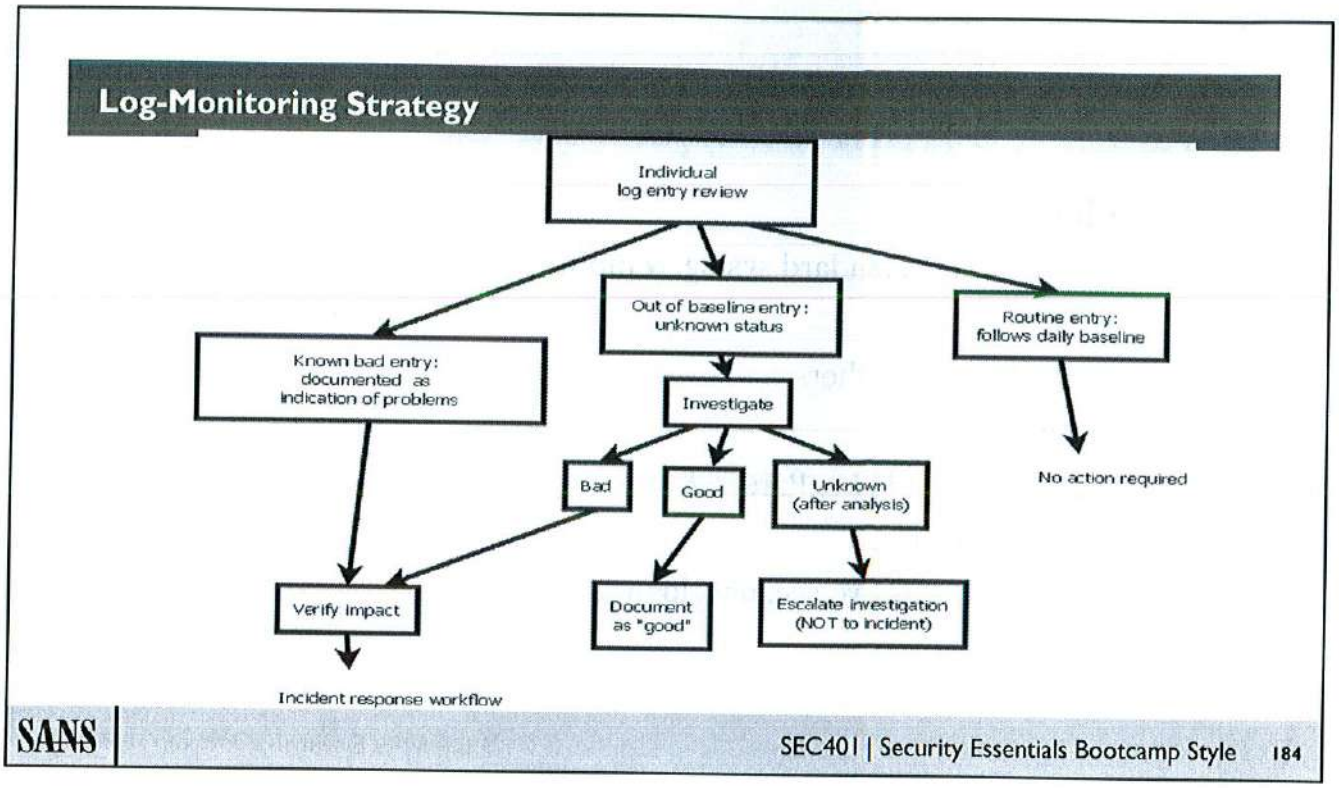
When ready to move to correlation and alerting, get OSSEC or other tools. At this point, you gain a degree of awareness of what is going on in your environment.

Reference

1. Security Analytics with Big Data: Integration - https://securosis.com/blog/security-analytics-with-big-data-integration

© 2017 Secure Anchor Consulting

Technet24



Log-Monitoring Strategy

This slide summarizes a sample monitoring strategy that reflects the actions of the log analyst who performs a daily log review. You can adopt a similar strategy for your organization. However, before we proceed, the issue of frequency of the log review needs to be addressed: It is unlikely that monitoring won't necessarily be daily. Notice that we don't start from bad or problematic. In the world of logs, we often don't know whether something is bad; we can often just say that it is interesting or suspicious.

This strategy helps us to answer questions such as

- How to plan a response strategy to activate when monitoring logs?
- Where to start the process?
- How to tune it for more efficient analysis?

So, it all starts from seeing something suspicious or interesting, or maybe some log entry that was associated with problems before. The first simple decision we make is "is this a known bad event?" If it is, we evaluate whether there is an incident in the making and we need to go into our incident response process. An example of that would be an IDS event that indicates a compromised system or a confirmed successful attack (for example, IRC traffic from your internal network, shell commands executed, or a firewall message showing your server connecting to a site from a known bad location.

Now, if we don't know that this is something "known bad," we need to do an additional assessment to move from the uncertainty of "it looks strange" to either "it's OK" or "it is bad and we possibly have an incident."

Finally, if something is suspicious but ends up being benign, we might have a false alarm (for example, a false positive when an IDS triggers on a benign activity and issues an alert). In this case, we end up needing to change the rules, adjust other controls, or do something else.

© 2017 Secure Anchor Consulting

Setting Up a Log-Monitoring Program – Phased Approach

- It is important to understand what information is needed to drive the setting up and configuration of the log program
- Log types to collect
- Networks/systems to collect logs from
- Add more users of log data, expand scope
- Address new use cases (as needed)
- Plan retention
- Plan response procedures
- Potential problem areas and how to avoid them

This slide looks at how to progress from "yes, we want to start monitoring our logs" to a fully established program for doing so across the whole organization. Phased approach is the only choice, since accomplishing this requires a lot of integration work to collect logs from various systems.

First, we decided which log types to integrate: firewalls, maybe network IDS, UNIX and Windows servers, then others. As we discussed before, a common strategy, which I call natural flow of log management, is:

1. Firewalls, network gear
2. Other network security gear
3. Servers (UNIX, then Windows)
4. Other server applications (web, mail)
5. Databases
6. Applications
7. Desktops

You need to adjust based on your needs and focus. For example, you might choose to go with servers first, since the focus might be on inside activities, not so much on outside.

Second, we decide which sections of the network, and over the IT systems overall, to connect first: maybe DMZ, then core, then other internal systems. The opposite is also common: especially when application audit is the focus.

Growth of your deployment will likely happen in several directions: more log sources, more network segments, and more users. For example, with the user community, it might grow like this: security team first, then other parts of IT or internal auditors. Indeed, approaching logs in silo-ed fashion, where every team that needs them deploys their own system, is extremely wasteful and inefficient.

What commonly happens next is to address new use cases for log data: from availability monitoring to incident response to compliance to others. This one is hard to encode in a standard; it is driven by the needs of the organization that deploys log management and monitoring.

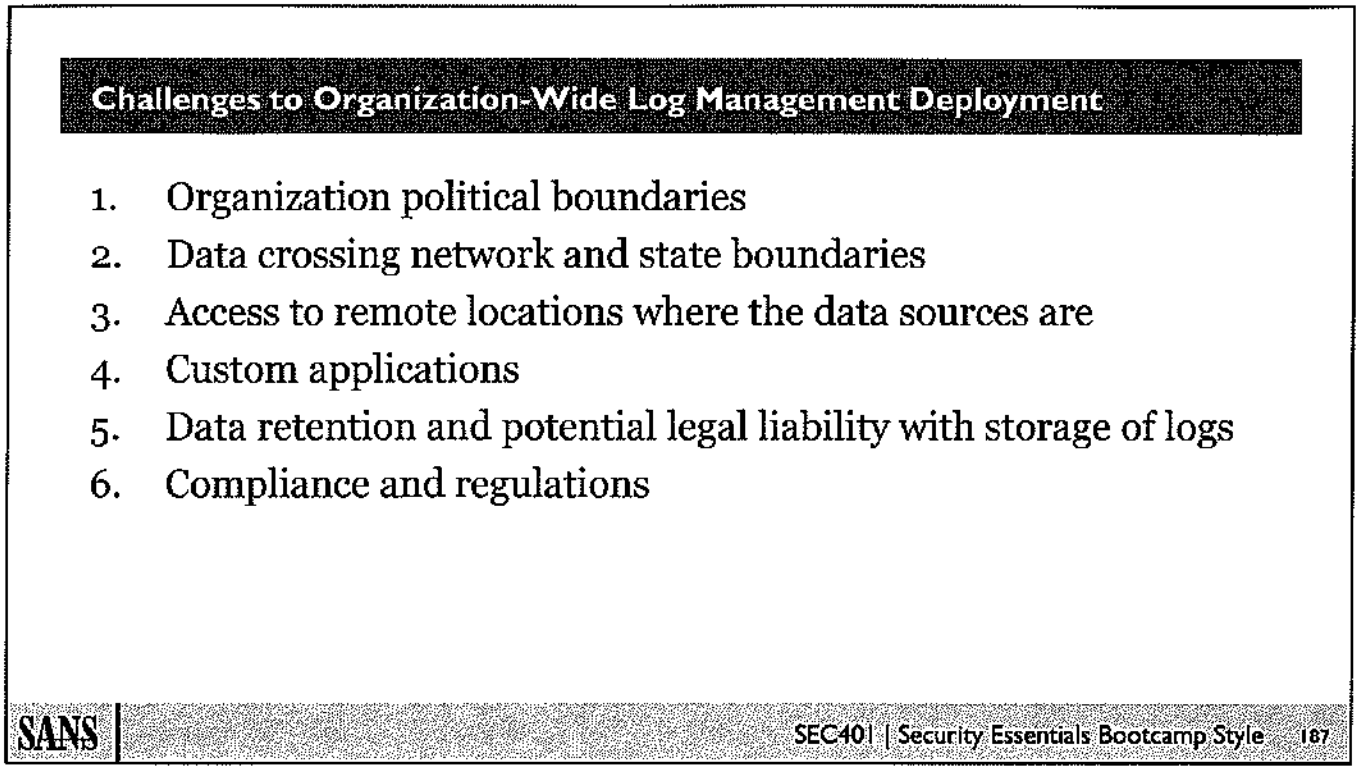
© 2017 Secure Anchor Consulting

Technet24

Reference

1. Log Management: A Pragmatic Approach to PCI DSS - http://www.isaca.org/Journal/archives/2012/Volume-1/Pages/JOnline-Log-Management-A-Pragmatic-Approach-to-PCI-DSS.aspx

© 2017 Secure Anchor Consulting



While deploying organization-wide log management and monitoring, a few challenges are common. Let's review them and help you inoculate yourself from them.

It might sound surprising, but the top challenge to log management and monitoring is not technical: It is an organization's political boundaries. They are inherent in any project involving integration; and log monitoring presents an ultimate integration project when many systems, owned by various groups, all send data to the same system for monitoring.

Data crossing network and state boundaries are also a challenge, especially for global deployment. For example, can an analyst located in the United States be monitoring logs that originate in France? Data privacy law and other cross-border concerns related to moving sensitive data (yes, logs are sensitive data!) also make deployment complicated and mandate the involvement of a legal team.

Access to remote locations where the data sources are located is also a challenge. Although remote management of log collection is common and can be controlled centrally from one or more management stations, remote installation is typically not possible. Thus, the team that plans and executes the log-management strategy might need to travel to remote locations where the log sources are or obtain the cooperation of system administrators from such locations.

Custom applications that produce unsupported and sometimes even undocumented log formats present a challenge, but not an unsolvable one. Leading commercial log-management vendors today can actually collect ALL logs, even if the vendor engineers didn't have a chance to develop support for such esoteric logs.

© 2017 Secure Anchor Consulting

Technet24

Key Logging Activities

The student will obtain a high-level understanding of the key logging activities and the frequency in which they should be performed

This page intentionally left blank.

© 2017 Secure Anchor Consulting

Real-Time Tasks

- Malware outbreaks
- Convincing and reliable intrusion evidence on key systems
- Serious internal network abuse
- Loss of service on critical assets
- Regulated data theft

This slide outlines a few things that you need to do as a part of your log-monitoring program. In case you detect any of the things from this slide, it is likely that you will initiate an immediate response.

Malware outbreaks, which can be detected by an antivirus, but also from firewall and IDS logs, clearly require immediate action in the form of quarantine and cleaning. Compared to a few years ago, antivirus solutions are MUCH more likely to miss the malware.

Convincing and reliable intrusion evidence usually comes from a combination of logs, such as correlation from IDS/IPS and firewall logs. In most organizations, such events also call for an immediate response (system shutdown or disconnect, if at all possible).

Serious internal network abuse or a serious policy violation, such as access to illegal content from the systems owned by your company, will likely trigger an immediate response as well because of massive legal liability concerns. Web proxy logs will likely be the source of such info.

Regulated data theft tops the chart for many organization and causes immediate response in some cases due to the "double whammy" of direct loss and regulatory fines.

Finally, loss of service on critical assets, whether because of security problems, human errors, or software bugs, also necessitates an immediate response. Server logs contain plenty of information on such potential problems.

Finally, log deletion and corruption detecting also need to happen ASAP, because it is a reliable indicator of an intrusion or insider abuse.

Reference

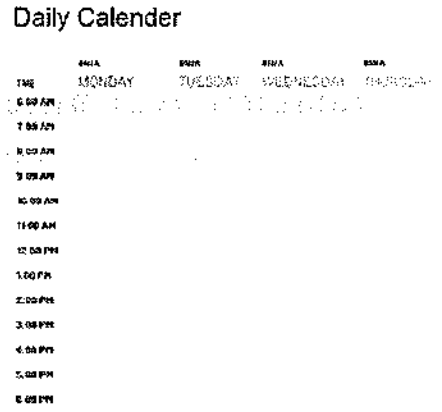
1. Real-time analytics have a 24/7 appetite for storage - https://www.tegile.com/blog/real-time-analytics-24-7-appetite-storage/

© 2017 Secure Anchor Consulting

Technet24

Daily Tasks

- Unauthorized configuration changes
- Disruption in other services
- Intrusion evidence
- Suspicious login failures
- Minor malware activity
- Activity summary



The next few slides go through sample tasks that result from a log-monitoring program. We outline a few things you would need to do as a part of your log-monitoring program.

In case you detect any of the things from this slide, it is likely that you initiate a response within a day or so. When monitoring logs, one needs to look for things presented here at least daily.

Unauthorized configuration changes are a major problem for many organizations. Now, we are not advocating submitting forms in triplicate to change a port in a firewall rule, but exercising control over system and network configuration change, thus, detecting when such changes without authorization is critical.

Disruption in other less critical services, such as a failure of the secondary DNS server, or other systems, will not immediately disrupt your business operations. If you see log messages indicating the system is experiencing errors that might cause it to fail, a response within the same day is desirable.

Intrusion evidence, which might not be as compelling as what we discussed on the previous slide, also falls in this category. For example, a large number of different attacks launched from one source (even with no indication of their success) calls for a prompt investigation.

Suspicious login failures, such as a large number of access attempts from one source across one or many systems, might point at either a hacker probing your system (if the logs come from the Internet-facing servers) or even at an insider attack (if logs come from internal systems). Response to such events needs to be swift, especially if a long string of login failures is followed by a successful login message.

Minor malware activity, such as machines infected by spyware (which are sadly common nowadays) or log records indicating antivirus protection failures (which might or might not be malware induced) also call for action within a day.

© 2017 Secure Anchor Consulting

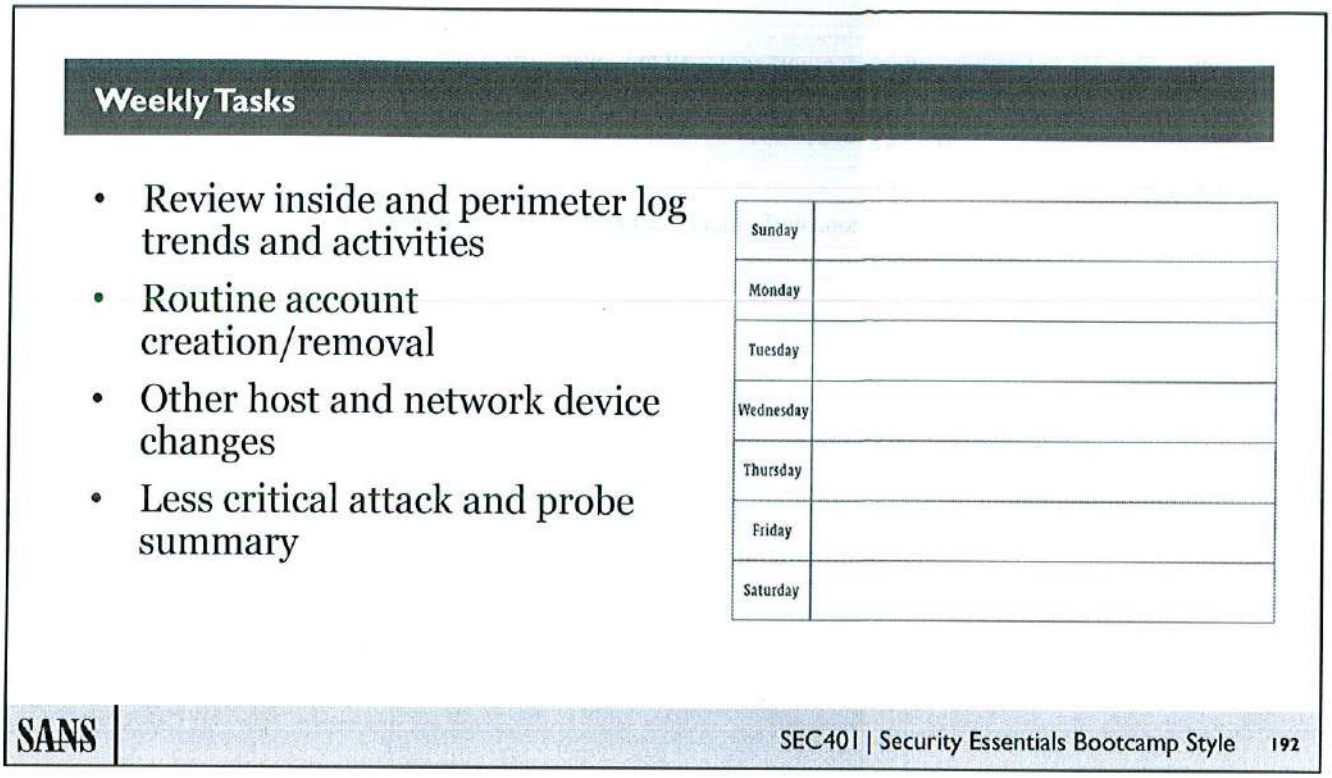
Running some kind of activity summary report across all logs every day is also a good idea, because it helps to take a quick look at everything that happened in your environment that day, malicious or not, and serves to improve your situational awareness.

Reference

1. Daily Calendar - http://www.shinzoo.com/category/calendar-02/daily-calendar.php

© 2017 Secure Anchor Consulting

Technet24



If you detect any of the things on this slide, for example, most organizations initiate a response within a week. When monitoring logs, one needs to look for things presented here at least weekly. A week is the smallest time period when some form of trend analysis becomes useful.

Review inside and perimeter log trends and activities by running a summary of events versus time. Here are some of the useful summaries to look at:

- System Failure Trend
- Attacks Trend
- Configuration Change Trend
- User Access Trend

If you collect and monitor logs by using a commercial log-management system, it typically has reports similar to these. If you are building your own, make sure that you can do things of that sort.

Routine account creation/removal is another thing to review weekly. The goal here is to look for new accounts created without proper authorization, as well as old accounts NOT being removed when employees are terminated (requires receiving the information from HR or another such information source). Ideally, one should be able to monitor account creation daily or even in real time (and correlated with other events) to detect accounts added by malicious parties.

Host and network device changes need to be reviewed at least weekly. The logic here is similar: Learn when/how the legit changes happen to detect the other kind.

Less critical attack and probe summary are probably performed during the summaries of the previously listed trends; however, it makes sense to mention this specifically, because changes in probes and attack floods might sometimes be significant.

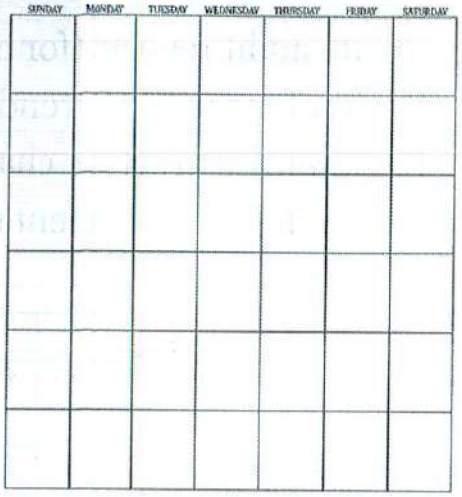
Reference

1. Weekly Calendar - http://www.shinzoo.com/category/calendar/weekly-calendar.php

© 2017 Secure Anchor Consulting

Monthly Tasks

- Review long-term network and system log trends
- Minor policy violation summary
- Various resource usage reports
- Security technology performance measurement



SANS SEC401 | Security Essentials Bootcamp Style 193

Here, we outline steps and go through sample tasks you need to perform as a part of your monthly log-monitoring program. When monitoring logs, one needs to look for things presented here at least monthly.

First, review monthly long-term networks and system log trends. A month is a time period when many of the normal network and system behaviors will be manifested, for example, a weekend decrease in system access and network usage or the end-of-month rush. In addition, some of the system maintenance tasks are scheduled monthly, and thus will be reflected in a monthly log review.

Minor policy violation summary, such as network acceptable use policy (AUP) or inappropriate web browsing (such as non-work-related sites reflected in web proxy logs), also call for a monthly review. Reviewing firewall, web proxy, and network IDS/IPS logs for such events can happen monthly.

Various resource usage reports, such as bandwidth utilization per server, user or application, or summary of resource exhaustion cases (for example, memory exceeded or system load too high conditions) is another summary report based on logs that need to be looked at monthly.

Logs can also help with the overall security technology-performance measurement. For example, a report on successfully blocked attacks or other indications of successful attack blocking or a report on denied connections from a firewall are examples of such monthly reports.

Reference

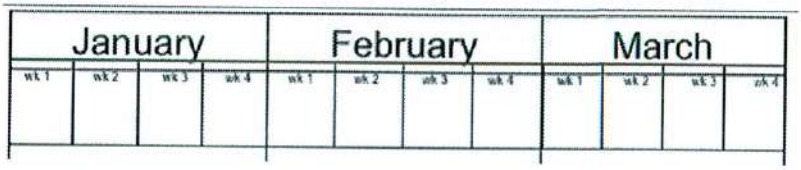
1. Monthly Calendar - http://www.shinzoo.com/category/calendar/monthly-calendar.php

© 2017 Secure Anchor Consulting

Technet24

Quarterly Tasks

- Create audit reports (for a quarterly audit)
- Review longer-term trends across log data
- Review infrastructure changes that affect log monitoring
- Review log-management system performance



Here, we outline steps and go through sample tasks you need to perform as a part of your quarterly log-monitoring program. When monitoring logs, one needs to look for things presented here at least quarterly.

Create audit reports (for a quarterly audit) that prove that other tasks (such as daily and weekly log reviews) were performed. For example, PCI DSS regulation mandates a daily review of logs, which means that an auditor wants to see a report that proves that logs were reviewed daily.

Review longer-term trends across log data in a manner similar to weekly and monthly trends. Some events on a network will occur quarterly, such as increased CRM system access by the business units to perform their quarterly reporting. Thus, a quarterly log review reveals them and helps you establish this final piece of a profile of normal activities. In addition, many of the quarterly summaries can be saved as evidence of log review in case of future auditor requests.

Reviewing infrastructure changes that affect log monitoring is critical for keeping your log-management program up-to-date; if there are new additions to the network, logs need to be collected from these new systems, and the logs need to be included in a normal profile of the activity. In addition, new regulation and mandates might require new, expanded scope for log collection and monitoring.

Finally, reviewing logs from your log management and monitoring system is needed to identify future problems with log monitoring.

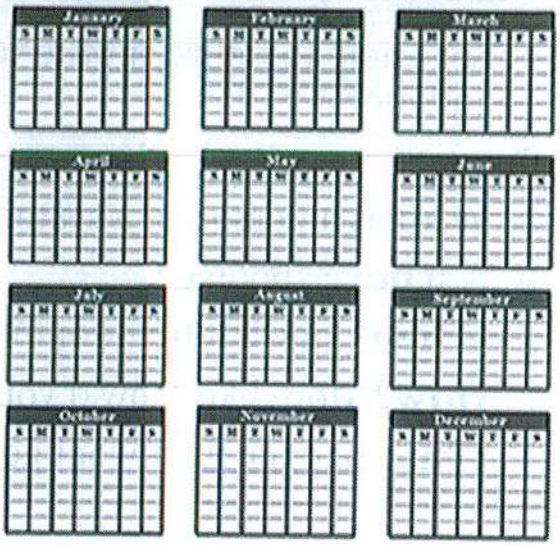
Reference

1. An In-Depth Guide To Developing And Planning Your Quarterly Editorial Calendar For Your Blog - https://www.blogmarketingacademy.com/editorial-calendar-planning-process-example/

© 2017 Secure Anchor Consulting

Annual Tasks

- Review log policy
- Verify log retention and archival
- Review longest-term trends across log data
- Use logs to analyze and prepare for next year's security budget
- Review possible new regulations that affect logging



SANS | SEC401 | Security Essentials Bootcamp Style 195

Here we outline steps and go through sample tasks you will need to perform as a part of your annual log-monitoring program. When monitoring logs, one needs to look for things presented here at least annually.

Review log policy and check whether you log too little or too much.

Verify log retention and archive at least once a year. Yes, the old sys admin joke, "The backups are perfectly fine, it's the restores we have problems with," still rings as ominous (and as true!) as ever. Verifying that you can read the archive logs, as well as that you do retain the logs for as long as you are committed, is a critical task. Now, it is not only the media decay that you are worried about. After all, hard drives likely won't decay. However, there are other issues to worry about: How about your log-management vendor updating its system so that the old backups can no longer be read?

Review of the longest-term trends across log data is a useful and enlightening task. Run the trend reports and other long-term views of the recorded events, from high-priority that you dealt with on a daily and weekly basis to normal audit events. Pay attention to any possible growth of log volume over the course of the year. Save such annual reports for the future, in case they are needed.

Few realize that we can use logs to analyze and prepare for next year's security and IT budget. The logs you collect contain plenty of information on various resource-exhaustion conditions. If you summarize such conditions appearing in logs over the course of the year, you can get some idea about the utilized system capacity. Similarly, summarizing bandwidth usage over time helps determine whether more network pipes need to be ordered.

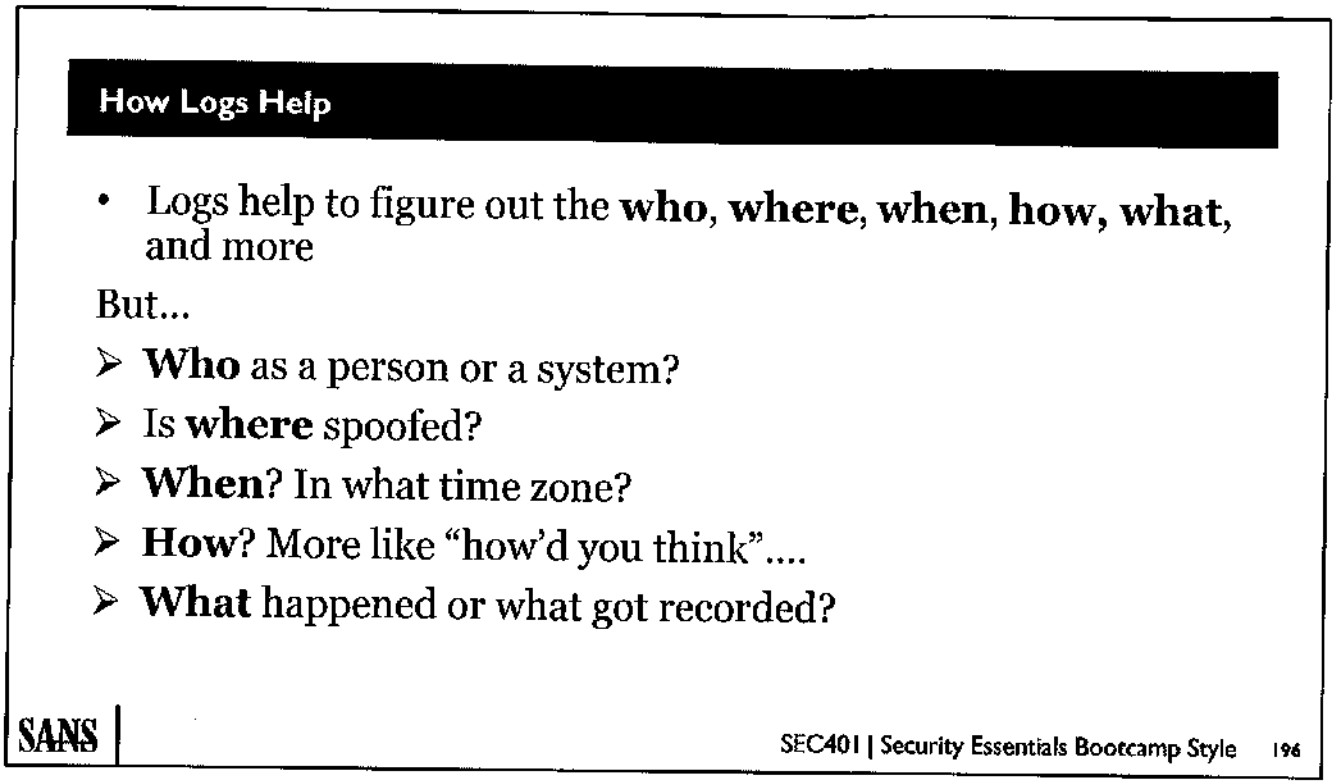
Also, every year, we need to review possible new regulations that affect logging. Admittedly, the new ones don't show up as often as new security problems, but changes to regulations do have logging implications.

Reference

1. Yearly Calendar - http://www.calendarlabs.com/blank-calendar/yearly-calendar/

© 2017 Secure Anchor Consulting

Technet24



Here, we learn that even though logs help us to figure out who, where, when, how, what, and more information objectively (or at least we like to hope so), challenges to that might hide in all the seemingly innocuous places. For example:

Logs tell us who did something, but is "who" a person or a system?

Is "where" spoofed? Is relying on logs for locating the perpetrator a bad idea? (It is!)

The question, "When?" brings up a long string of challenges. In what time zone? Is the time right?

Asking how something happened is often more akin to asking, "How do you think it happened?" which points at beliefs and not facts.

Similarly, the log should tell us "What happened?" but really shows, "What got recorded [in logs]?"

© 2017 Secure Anchor Consulting

Where Is It All Going?

- **Now**: collection, **then**: access, **next**: analysis
- **Compliance** mandates logging
- Moving up the stack to **applications**
  - Starts from databases
  - Cloud applications: HUGE challenges
- **More logs**!
- Log standards... finally!... maybe

What are the top three trends in the log-analysis industry?

There's a rapid increase in the breadth of log sources that people care for (and thus collect data from). It used to be just firewall and IDS logs, then servers, and now it is expanding to all sorts of log sources, such as databases, applications, and so on.

This might sound boring, but it is still a major trend: more regulations, as well as governance frameworks and standards will cover logs and logging.

There is also a trend toward auditing more access and more activity through logs; for example, few of the file server, storage, or database vendors cared about logging, but now they do. (Well, some do and some are starting to.) What used to be just about access to information is now evolving into auditable access info.

© 2017 Secure Anchor Consulting

Technet24

Conclusions

- Turn logging **ON**! You probably **have to** (for compliance)
- Plan your log-management strategy
- Decide how to deal with logs: build versus buy versus outsource
- Remember forensics challenges
- Avoid logging mistakes

Logging is all about the quality of the logs not the quantity – too many logs can make it difficult to find the activity that really matters and makes it easier for the adversary to hide.

In conclusion,

- Turn logging **on**! You probably **have to** (for compliance).
- Plan your log-management strategy.
- Decide how to deal with logs: **build** versus buy versus outsource.
- Remember forensics **challenges**.
- Avoid logging mistakes.

© 2017 Secure Anchor Consulting

SANS

Module 17: Active Defense

Module 17: Active Defense

As adversaries continue to get more advanced, traditional security technologies often can be bypassed. In addition, adversaries are so familiar with how they work that they do not necessarily slow down the adversary, thereby leading to significant damage from a compromise. The idea with active defense is to perform activity to make it harder for the adversary to break in and cause damage. In addition, active defense tries to find out more information about the adversary and what they are doing. Normally the defense is very passive, but in order to implement more effective security, the defense needs to be more aggressive and active in how they are breaking into systems.

© 2017 Secure Anchor Consulting

Objectives

- What is Active Defense?
- Active Defense Techniques
- Active Defense Tools
- Honeypots & Active Defense

In this module, we will explain what active defense is and how it can be used. You will get an appreciation for new ways to approach security and how to make your defensive solutions more active. The methods discussed in this model will help turn the tables on the adversary and take away the current advantage they have. Techniques and tools for implementing active defense will also be discussed.

© 2017 Secure Anchor Consulting

Active Defense Overview

The student will obtain a high-level understanding of what Active Defense is and how it can be used effectively within an organization

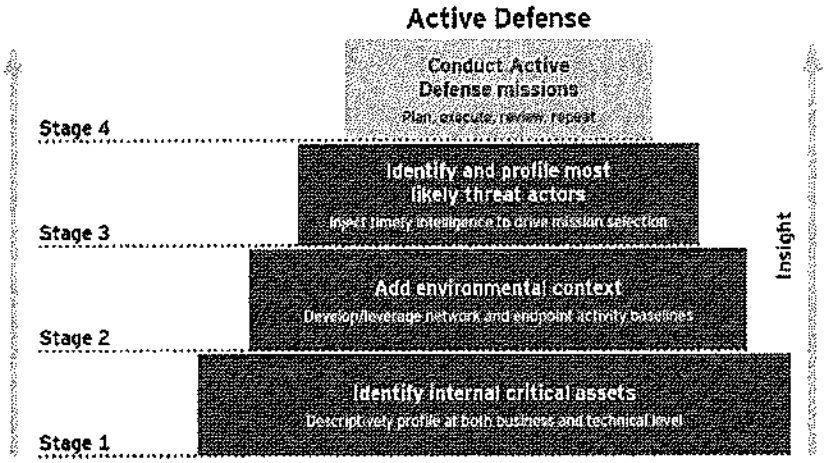
This page intentionally left blank.

© 2017 Secure Anchor Consulting

Technet24

What Is Active Defense?

- Activities performed by the defense that are more offensive oriented
- To make things more difficult for the offense
- Defensive activity that make it more expensive (both time and money) for the adversary to cause harm



A key motto in security is "offense informs the defense". The idea is that the more you can understand how the offense works and operates, the more effective an organization's defensive measures can be. Instead of focusing on fixing random vulnerabilities, focus on vulnerabilities in which there is a threat with a high likelihood and high impact of causing damage to critical assets. Active Defense takes this concept to the next level, instead of having the defense utilize the offense to guide them, what if the defense becomes more offensive oriented.

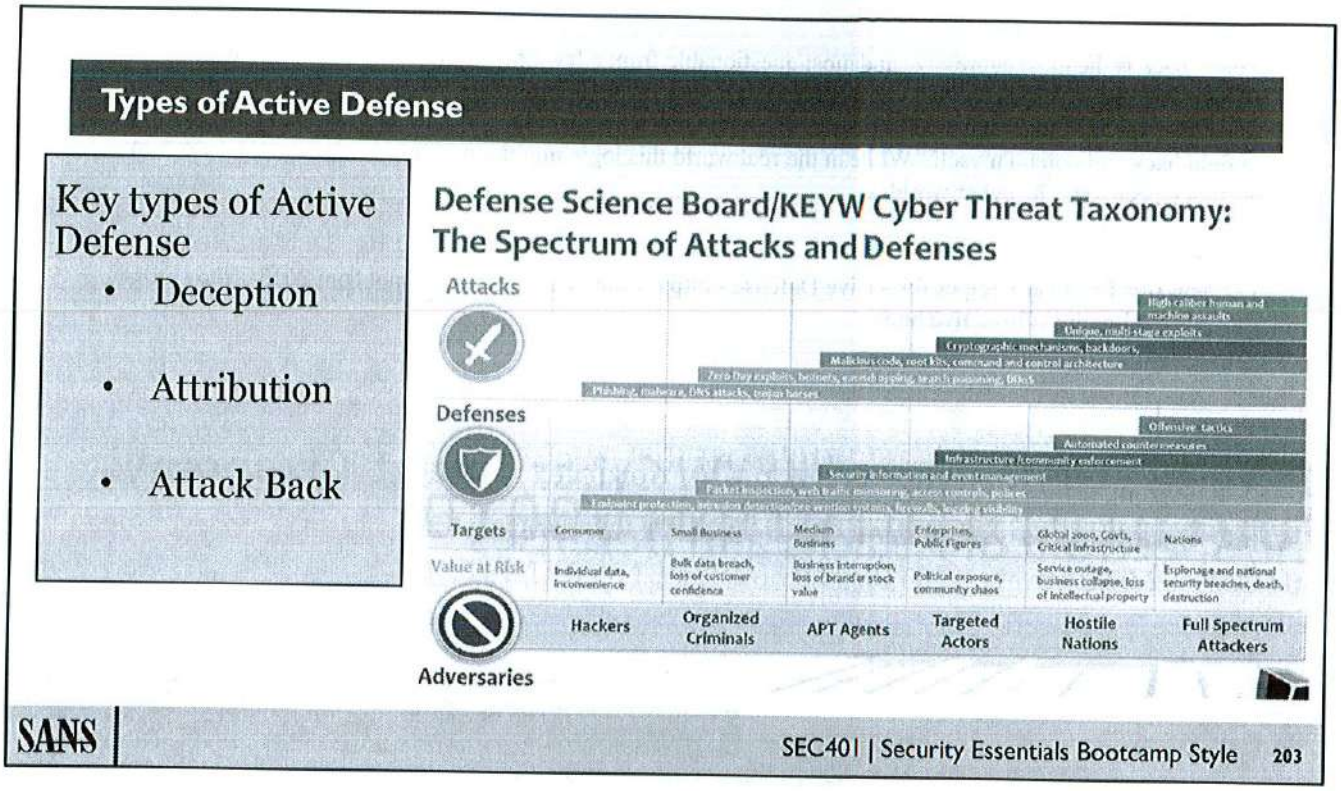
If the defense knows and understands how the offense works and operates, they can perform activity that makes it more difficult for the adversary. A great example is when an adversary performs scanning to determine activity hosts. What if a device, such as a firewall, responds that all IP addresses are active? This will slow down the adversary and make it much more difficult to cause harm.

The idea of active defense is not to replace existing technologies, but complement them to make it harder for an adversary to break in. The goal with Active Defense is to find out as much information about the adversary and slow them down. Not only will slowing down the adversary by making them use more resources, make it harder for them to break in, but ultimately will give us more time to respond.

References

1. How I Learned To Love Active Defense - http://www.darkreading.com/attacks-breaches/how-i-learned-to-love-active-defense/a/d-id/1321361
2. Enhancing your security operations with Active Defense - http://www.ey.com/gl/en/services/advisory/ey-cybersecurity-enhancing-your-security-operations-with-active-defense

© 2017 Secure Anchor Consulting



As the adversary continues to become more advanced and effective in their attacks, the defense needs to continue to up their game. While prevention and detection are very effective, they tend to be more passive, providing an avenue for the adversary to bypass or defeat the defensive mechanisms. Therefore one of the many reasons why the adversary causes so much harm and damage is because they are very aggressive. The problem is the defense needs to fight fire with fire. One of the problems with many defensive approaches is that we are bringing a knife to a gun fight. If you have watched any Hollywood movies, you quickly know how that ends. If you do not watch Hollywood movies, just look at the news at companies that have had significant breaches will millions of records stolen, you will get another perspective on how that ends.

For the defenders to level the playing field, we need to get more aggressive and active defense is one way of accomplishing this. Since active defense is a relatively new field there are a lot of perspectives on how it works and the best way to accomplish; however, in general, there are three general categories of active defense: deception, attribution, and attack back. While each of these will be discussed in details of the next several slides, let's quickly review the idea behind each of these types of active defense.

Deception focuses on misleading the adversary by providing false or inaccurate information to make their job more difficult and/or to slow them down. For example, if an adversary connects to a web server and thinks it is IIS, they will take a different approach than if it is Apache. However, if the actual web server is Apache, but they are under the belief that it is IIS, they might eventually figure it out, but it will slow them down and use up more resources than if they knew the truth. Also, depending on the sophistication of the adversary, they might actually get frustrated and move on to a new victim.

Attribution is focused on finding out who the adversary is and potentially what country they are coming from. This can be useful for both incident response and prosecution to understand the true identity of the adversary. Knowing who the adversary is can also be very valuable to understand their intent. A competitor, a foreign adversary or organized crime could all have different reasons for targeting or breaking into your organization.

© 2017 Secure Anchor Consulting

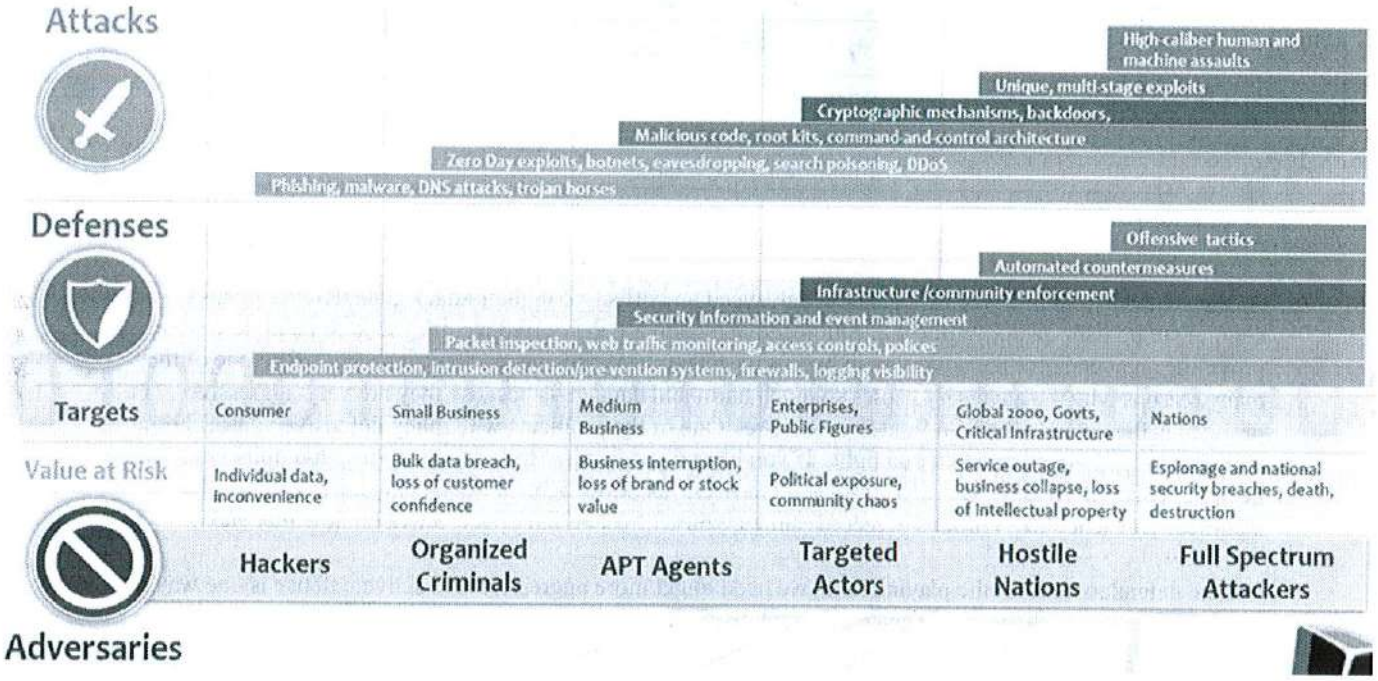
Technet24

Attack back is the most aggressive and most questionable from a legal perspective. In involves actively attacking the adversary and either causing harm or disrupting their ability to continue to attack you. It is sometimes referred to as cyber self-defense. If your throw a punch at me and start to hurt me, I have the right to fight back and defend myself. While in the real world this logic might work, it is not always the case from a legal perspective in the cyber world.

Reference

1. Towards Practical Recipes for Active Defense - http://security-architect.blogspot.com/2013/12/towards-practical-recipes-for-active.html

Defense Science Board/KEYW Cyber Threat Taxonomy: The Spectrum of Attacks and Defenses



© 2017 Secure Anchor Consulting

Active Defense: Deception

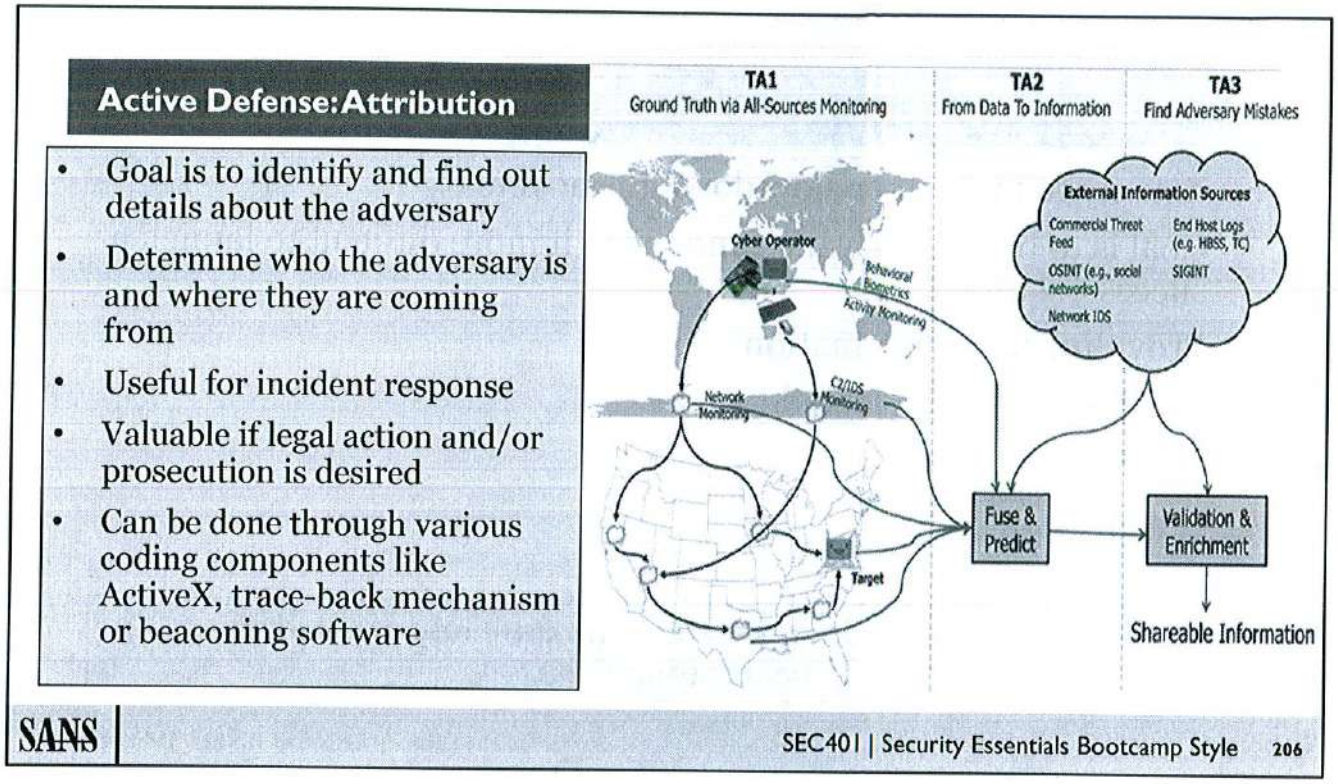
- Activity to mislead or slow down the adversary
- Goal is to be an annoyance and make it more difficult to launch a successful attack
- Providing false information
- Common methods include
  - Honeypots
  - Decoy servers
  - False DNS entries

Warning – can also increase time and difficulty of performing penetration tests

Deception is a very effective method of active defense and tends to be the one method that creates less legal concerns. With deception, you are providing false or misleading information to the adversary, but you are not actively doing anything to the adversary. The adversary is breaking into your systems and you are in essence providing them a false map with incorrect information. The main objective with deception is to slow down the adversary and make it harder for them to break in and/or cause it to take more time to cause harm. It also has an additional benefit of providing the defense more time to detect an attack and, in some cases, easier to detect an attack. If an attacker can launch a single exploit and break into a system, there are not a lot of packets or log entries and it could be hard for an organization to detect. However, if the adversary is provided false information so they have to launch many exploits and try many attack vectors, this greatly increases the amount of network traffic and log entries, giving the defense a better chance of finding the adversary sooner and controlling the overall amount of damage.

In the next section we will discuss active defense methods in detail, but to help understand how deception works we wanted to provide a few quick examples. Honeypots are a multi-versatile defense tool. Not only can they reduce the amount of false positives and provide insight into what an attacker is doing they can also be a deceptive tool. If a honeypot is set up correctly and properly configured, an adversary could actually believe it is a real system and spend a lot of time breaking in. They eventually will figure out that it was a honeypot, but this is typically after they spent considerable time and resources. Also, since a honeypot has no legitimate purpose, tricking an adversary into breaking into a honeypot makes it much easier to detect and respond. False DNS entries and decoy servers both focus in on making the adversary believe there are more systems than actually exist. Therefore the adversary will spend time breaking into a non-existent server.

© 2017 Secure Anchor Consulting



As we discuss the different types of active defense, not only does each one get more and more aggressive, but they also start to create potentially legal issues and concerns. The focus of the next active defense method is attribution, finding out who the adversary is. Understanding and finding out details about the adversary can be very valuable in understanding the intent and the reason behind an attack. Sometimes executives ask, why are we being attacked or what is their long term objective. By finding out who the adversary is can help to answer these questions.

Attribution is also useful in incident response. Understanding what country the attacker is coming from cannot only help with eradication, but can allow for more aggressive defensive measures to be implemented, such as blocking future attacks from that country. Also if legal action is desired, it is important to know who is attacking you to not only prosecute, but to see if prosecution is even worthwhile. For example, if an attacker is a nation-state attack, the probably of being able to prosecute a foreign country is fairly low.

From a government and military perspective, attribution is critical in order to retaliate. As attacks become more and more aggressive, some attacks could be considered acts of war. If you are going to declare war on a country, you better be confident you are correct in the actual source of an attack. Since relay attacks and spoofing IP addresses are very common, just because the source IP address is coming from a certain country, does not mean the attacker is in the country. They could either be relaying through another country to hide their true identity or even worse be framing another country. If you want to start a major international issue, convince one country than another country is actively attacking them. This misdirection could cause major issues if the wrong country is attacked. As cyber warfare moves to the forefront of discussions, attribution becomes a critical area to address.

Reference

1. DARPA calls for help to improve cyber attack attribution - https://www.helpnetsecurity.com/2016/05/10/darpa-cyber-attack-attribution/

© 2017 Secure Anchor Consulting

Active Defense: Attack Back

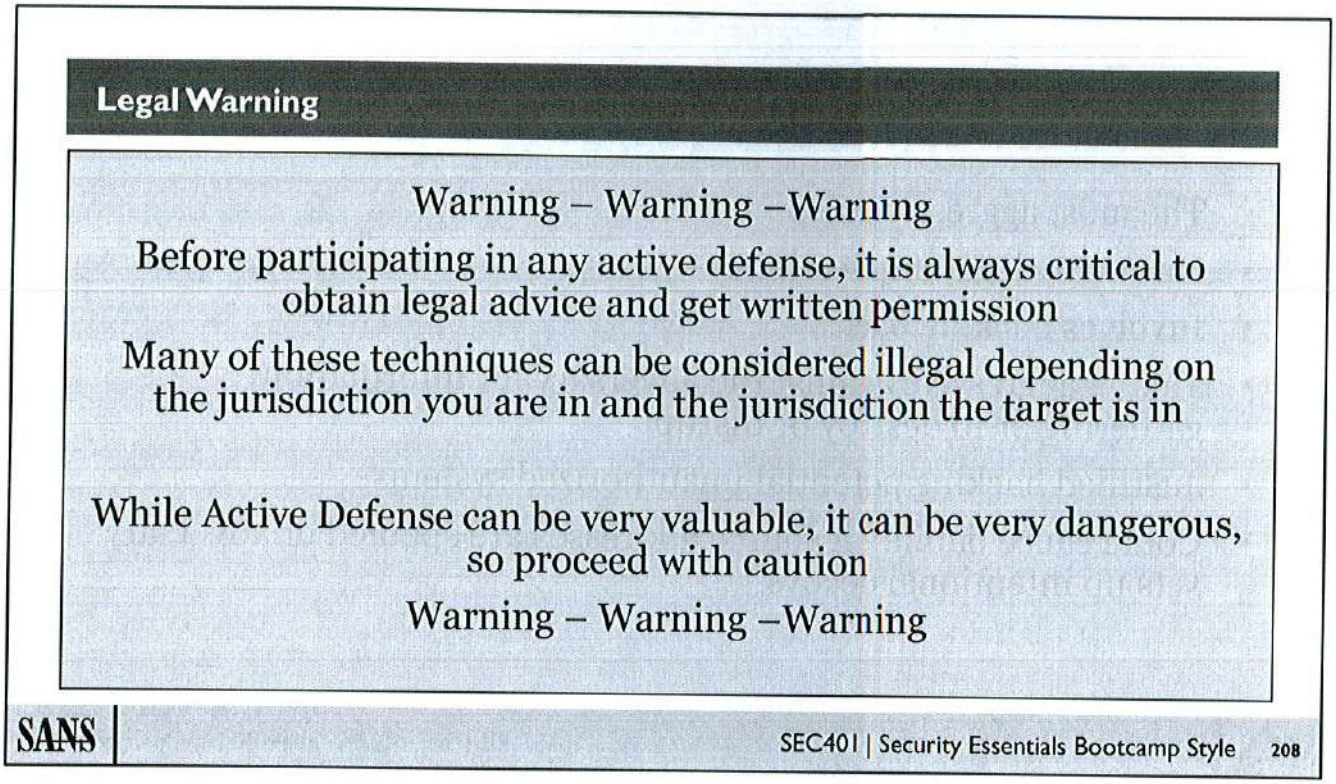
- The most aggressive step
- Has many legal implications
- Involves hacking back
- Taking aggressive against the adversary for information gathering or to destroy or disrupt
- Justified hacking potential unauthorized systems
- Could cause liability if source addresses are spoofed or adversary sets up intentional decoys

The most aggressive of all active defense mechanisms and the one with the most legal implications is attack back. As the name implies, this is where after you successfully identify who the adversary is and you actively attack them. This is sometimes referred to as cyber self-defense, which is based on the premise that if you start a fight with me, I can fight back. While this might logically make sense, from a legal perspective this does not always translate into proper justification or legal protection. In many countries, just because someone is attacking you, it does not justify you attacking back and causing harm to them.

With attacking back, proper attribution becomes critical. Just think of the legal implications of attacking back or in this scenario hacking the wrong entity. If you are not careful and an adversary knows that you are utilizing either automatic or manual attack back mechanisms, they can set you up. What if an adversary attacked you but spoofed their identity to be a major government or military organization. If you are not careful and reactively attack back, you potentially just hacked a major military, which not only could cause aggressive action on their part, but also raises serious legal implications.

Attack back could have value, but is still a very risky and questionable area in cyber security depending on how aggressive you become. Attacking back can be as straightforward as planting a tracking mechanism to passively monitor all activity of the adversary and reporting back to you. It could also include a phone home mechanism, where the adversaries system will beacon back to you, so you can track or monitor their activity. On the other extreme, it can be attacking back to destroy or steal information from the adversary system and cause equal or more harm than what they did to you.

© 2017 Secure Anchor Consulting



While hopefully it is obvious, we felt it was important to give a warning before talking about specific techniques and tools of active defense. Before participating in any active defense, it is always critical to obtain legal advice and get written permission. Many of these techniques can be considered illegal depending on the jurisdiction you are in and the jurisdiction the target is in. While Active Defense can be very valuable, it can be very dangerous, so proceed with caution. Also, remember, what is legal in one country might not be legal in another. Therefore, do not always consider where you are located but also take into consideration where the target is. Be careful…..

© 2017 Secure Anchor Consulting

Active Defense Techniques

The student will have an understanding of different methods and techniques for performing active defense

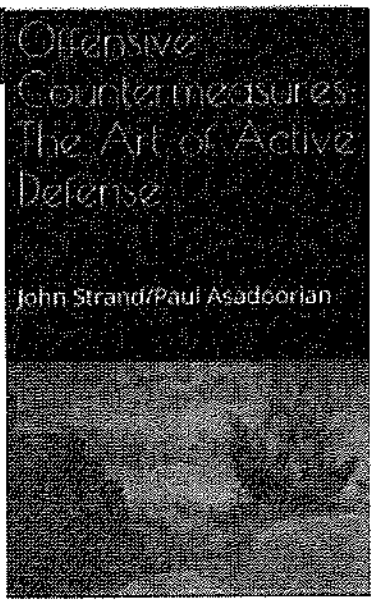
This page intentionally left blank.

© 2017 Secure Anchor Consulting

Technet24

Active Defense Techniques: Overview

- There are many methods of performing active defense
- Sometimes referred to as Offensive Countermeasures
- Ultimate goal is to slow down or mislead the adversary
  - Anything that accomplishes this goal can be used as part of active defense
- This section will cover some of the more common techniques
- Be creative and see what additional techniques you can deploy to make it more difficult for the adversary



There are a lot of methods for performing active defense and new ones are being developed all of the time. Anything that can be used to slow down or trick the adversary falls under this category. Who knows, as we go through this section and you go back to work to apply the knowledge, you might develop some new techniques of your own. It is also important to note that Active Defense is sometimes referred to as Offensive Countermeasures because we are implementing measures to counter the offense and the damage they are trying to cause.

A great book on the topic is Offensive Countermeasures: The Art of Active Defense and is a great read to augment your knowledge from this section. This section is meant to provide an overview of some of the more common topics, but with new cutting edge areas like Active Defense, that are still being developed, things change and advance very rapidly. Therefore it is important to constantly read and stay up on new developments.

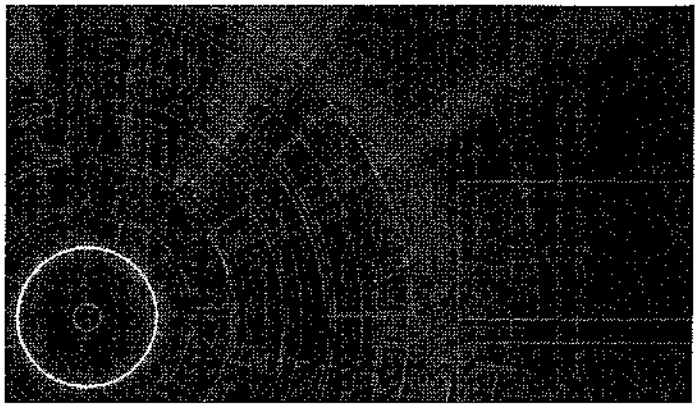
Reference

1. Offensive Countermeasures: The Art of Active Defense by John Strand and Paul Asadoorian

© 2017 Secure Anchor Consulting

Active Defense Techniques

- The following are some active defense techniques that can be used for deception:
  - Honeypots
  - Honeycreds
  - Jailed environments
  - False headers
  - Decoy IP's and ports
  - Tarpits
  - Bogus DNS entries



What is nice about many active defense techniques is that they are fairly inexpensive, easy to implement, and, in most cases, they utilize the technology you already have in place. Most modern security devices and operating systems support the ability to make configuration changes that are needed to perform various active defense measures. Typically, you do not have to purchase new hardware or software, you just have to properly implement what you already have in place.

The following are some of the active defense techniques that will be covered in this section:

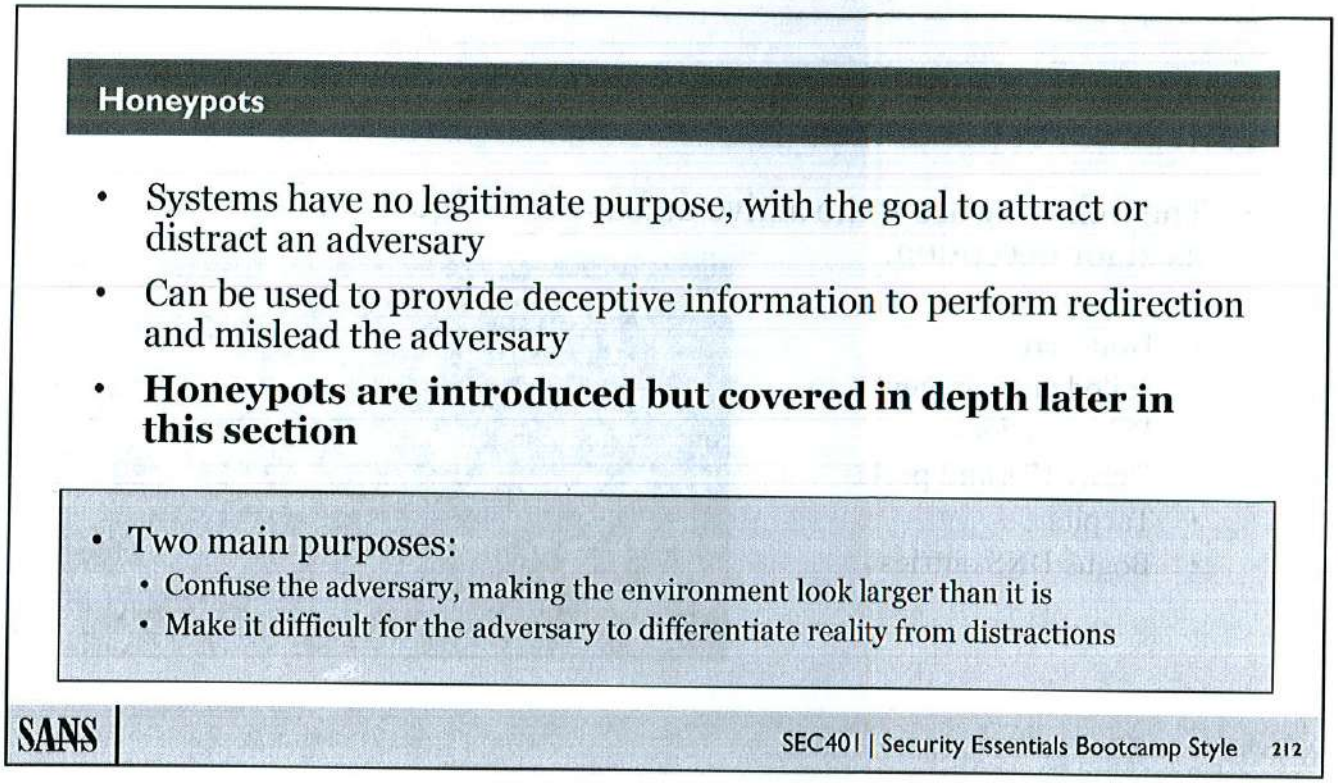
- Honeypots
- Honeycreds
- Jailed environments
- False headers
- Decoy IP's and ports
- Tarpits
- Bogus DNS entries

Active defense techniques are also closely tied to the applications and specific devices in use. Therefore it is important to make sure that these techniques are customized to the unique environments in which you operate and be creative in discovering new methods of defense. This list is meant to give you an idea but is not comprehensive or static, and as you implement active defense measures, it is important to make sure they are constantly kept up-to-date and evolve over time.

References

1. Implementing Active Defense Systems on Private Networks - https://www.sans.org/reading-room/whitepapers/detection/implementing-active-defense-systems-private-networks-34312
2. Is your board ready for a cyberattack this afternoon? - https://betterworkingworld.ey.com/better-questions/ready-cyber-attack

© 2017 Secure Anchor Consulting



Honeypots are a technique that has been around for a while, but continues to adapt in how it is used and configured. Honeypots are typically a device that has no legitimate purpose. Therefore, if anyone connects to it, it is unauthorized activity and indicative of an attack or a potential attack. Since no legitimate traffic should be accessing the honeypot, it is a valuable tool for quickly finding an adversary. If a normal server receives 10,000 log entries in an hour, determining good from bad is very difficult. With a honeypot, if the system receives 1,000 log entries, they are all bad and require further investigation.

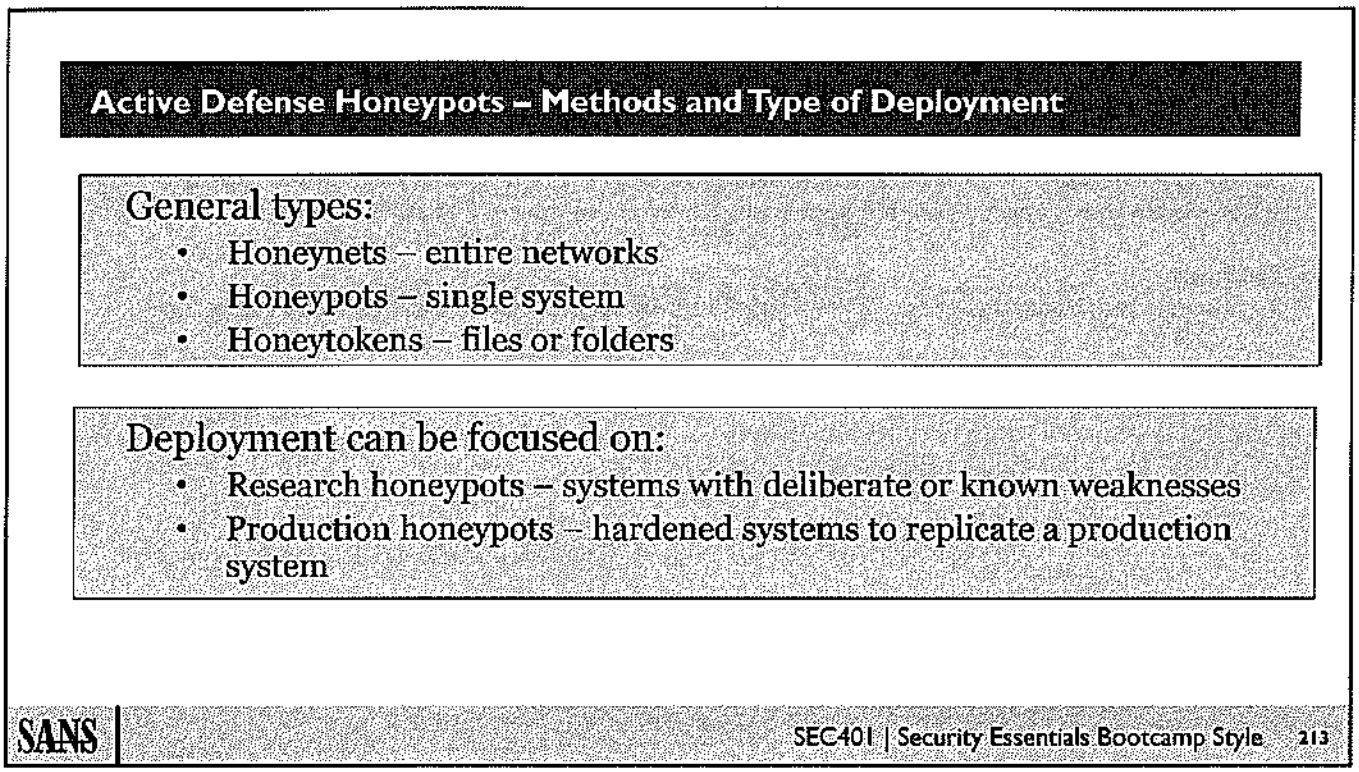
Depending on how honeypots are configured and used, they are sometimes referred to as sensors or canaries (to refer to the canaries that are used in the coal mines). These are pretty simplistic and are used to alert if any traffic or activity hits a system. They are clearly used in cases where traffic has to be carefully controlled and monitored. In ICS (industry control systems), like nuclear reactors, these types of sensors are valuable to alert on unauthorized activities. Though these simplistic honeypots may have a little value to active defense in slowing down basic host discovery, because they are merely sensors, the overall value is minimal.

Active fully-built honeypots are much more valuable in active defense and often take advantage of utilizing virtual machines. In this case, a full-blown system is set up to provide false information, provide misdirection, and perform high-volume activity that uses up a lot of the adversaries' resources and makes the adversaries' job much more difficult. Traditionally, this was fully-built servers running on detected hardware but this was very expensive and, therefore, was limited in its use. The advent of virtual machines has allowed honeypots as active defense tools to grow rapidly because almost any physical server can now run multiple guest OS's, allowing for multiple honeypots to confuse the adversary and making their job much more difficult.

Reference

1. Hunting with Honeypots - https://www.mwrinfosecurity.com/our-thinking/hunting-with-honeypots/

© 2017 Secure Anchor Consulting



Traditional honeypots have several ways of being deployed and these types of deployments can also be applied to active defense with some minor tweaks and changes. The main type of deployment that is consistent with the name honeypot is a single system that is deployed on a network to analyze and understand how the adversary works. With active defense we take a twist with honeypots by putting a lot of services, applications with misleading information and traps such as tarpits, to slow down the adversary. As mentioned with the use of virtual machines, this allows for a scalable way to deploy active defense honeypots. Now instead of having one server for the adversary to target, there are 11; 1 is good and 10 are bad. The adversary now has to do more work to find the real system and analyzing the 10 honeypots is time-consuming and utilizes excessive resources.

Honeynets are very similar to honeypots, but is an entire network set up to deceive and slow down the adversary. While having an entire network will utilize significantly more resources from the adversary, for this to work, it has to be designed and developed in such a way that it is not obvious it is a trap. This means it takes considerable resources from the defender to properly deploy and the additional resources are usually not justified. Typically, honeypots are valuable from an active defense perspective, but honeynets usually require too much time and resources to properly maintain which is why they are not the first choice of the defender.

Honeytokens are the easiest and most cost effective honey deceptive techniques because they are files and folders, placed on a system, that look legitimate. While they can be used for deceptive reasons, the number that would have to be deployed to significantly slow down the adversary would be very large. Therefore, they are often used for detection; if someone tries to access a honeytoken, they are a bad actor, instead of purely for active defense. The one exception is database files and encryption. If an adversary can be tricked into believing that a honeytoken is a critical database and is encrypted, they could use up a large amount of resources trying to break or read the information.

In traditional honeypots, the main methods of deployment are research and production. Research honeypots are systems with deliberate or known exposures that try to lure the adversary in so you can understand how they work and operate. These honeypots are typically unpatched and running older services to make it very

© 2017 Secure Anchor Consulting

Technet24

easy for an adversary to compromise. The focus is on analyzing what the adversary does after they break in. Production honeypots are hardened, locked-down systems that should be secure. The goal is to see if there is an unknown vulnerability and if compromised by an adversary, can be used to improve the security of other production systems. Active defense honeypots often take a hybrid approach where they have several hardened servers to lure in and slow down the adversary; they might also have a few vulnerable services to draw in the lower-end adversary, utilizing their resources trying to perform damage or find elevation attacks.

Reference

1. Hunting with Honeypots - https://www.mwrinfosecurity.com/our-thinking/hunting-with-honeypots/

© 2017 Secure Anchor Consulting

Honeycreds

- Decoy accounts to distract an attacker
- Unused accounts are setup on the system
- Often associated with privileged accounts
- Tied to legitimate credentials
  - Rename the built in administrator account and setup a bogus account with the name administrator
- Can be used for solely alerting if someone logs in
- Can be used to monitor activity and limit damage

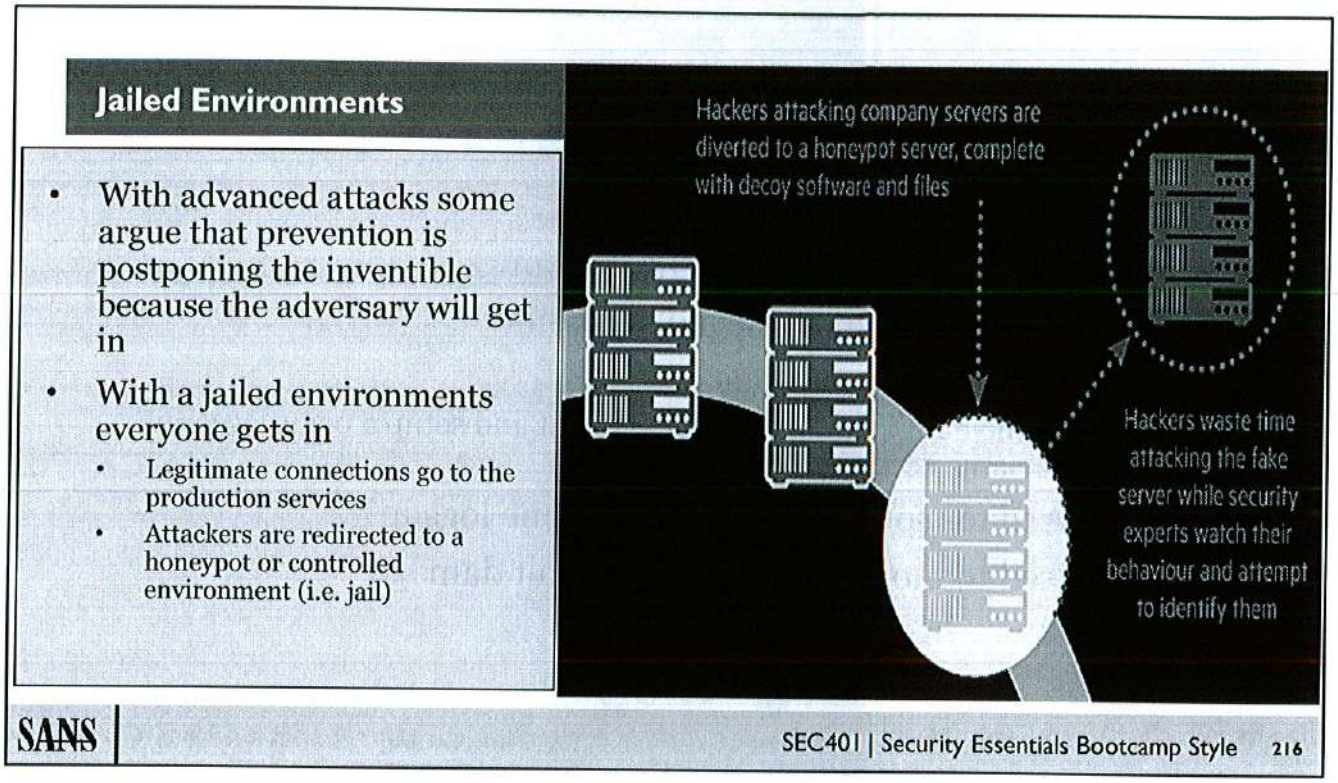
A new twist on the idea of honeypots is to set up decoy accounts on a system which are known as honeycreds. Typically the way an adversary gains access to a system is by identifying user ID's and trying to gain access by either brute force passwords or utilized known credentials that were gained through other attack methodologies. Since the goal of active defense is to slow down the adversary, additional credentials is another technique for making it more difficult for the adversary.

There are many different ways of setting up honeycreds. One method is to just set up decoy accounts to increase the attack vector for the adversary. This is very useful on systems that do not have a large number of accounts. If a system has only 5 accounts, performing targeted attacks and trying to brute force is relatively straightforward; however if 95 honeycreds are added, and now there are 100 accounts, it will take considerably longer for the adversary to perform the attacks. On larger systems that have 50,000 accounts, adding a few additional accounts is not going to make a significant difference in tripping up the adversary. In those cases, privileged or system honeycreds are more valuable.

Many operating systems and application have default system accounts that often have privileged access. Adversaries are familiar with these accounts and will often look for them as an initial target. This is because privileged access is more valuable than regular access, these are easier to find and target, and most times, system or privileged accounts passwords are not changed often, giving a longer window for someone to target and compromise a system. An example of this attack is many Windows systems have a built-in administrator account. A trick is to rename the administrator to a different name and create a decoy honeycred with the name administrator. Even though it has the name "administrator" it is not a privileged account because the original one has been renamed and has all of the privileges. An adversary would now spend considerable time trying to break into the account and even if they are successful, gain little value.

© 2017 Secure Anchor Consulting

Technet24



Many technologies that are deployed on networks today focus on prevention. The goal is to stop, or prohibit, an adversary from accessing a system or a resource. In the past, when adversaries used pre-written worms, if the worm was blocked by a firewall or prohibited access, they would move on to the next system and that attack would have been stopped. These worms typically went after the low-hanging fruit, trying to break into as many systems as possible, but not caring which ones they compromised. They knew they would not compromise every system and they did not care.

Today's advanced threats are quite different. Typically, with an advanced threat, they are targeting a specific organization and entity. Their sole focus is to break into that organization, so if their original attempt is prevented, they will keep trying. In many cases, they will keep trying until they are successful. Therefore, many people will say because advanced adversaries are very persistent, they will eventually get in because they will not stop trying. In these cases, people argue that prevention is just postponing the inevitable because the adversary will eventually find a way into your organization.

With these advanced threats, the idea is to allow everyone in but there is a twist. With the typically prevention model legitimate traffic is allowed in but attack traffic is blocked. With this new model, the legitimate environment that authorized people are allowed to access is complemented with a second environment known as a jail environment. This environment is configured to look just like the real environment, but is contained and controlled and does not have real data. Now when an adversary tries to make a connection, instead of being blocked, they are allowed into the jailed environment. The adversary believes they successfully got in and after spending time trying to cause harm might give up or recognize that there is nothing valuable and move on to a new target.

The success of this technique is how well the jailed environment is configured and how quickly the adversary is allowed in. For example, you still might want to block the adversary a few times before they are allowed into the jail so it simulates a real environment. If they are allowed in immediately, it could look suspicious. In addition, the more legitimate looking the jailed environment is, the more likely it will trick the adversary. As with any active defense technique, there is a balance between time and benefit. The more time you put in, the more likely you will deceive the adversary, but you always have to consider the time it takes for the benefit you receive.

© 2017 Secure Anchor Consulting

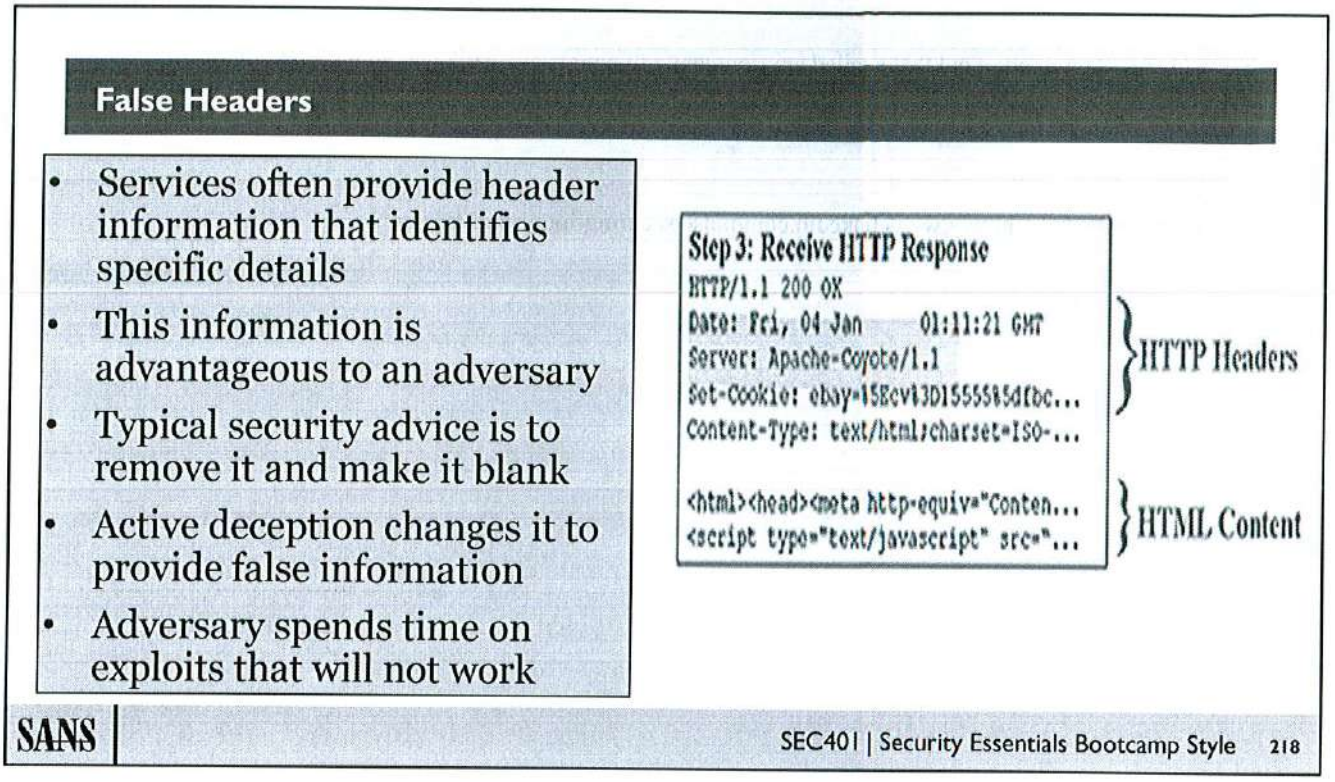
It is also important to point out that a jailed environment will not always work and with clever adversaries, they will eventually figure out what is happening; however, even in those cases, you still slowed down the adversary and made it take longer to break into the real systems.

Reference

1. Securing the IoT - https://www.linkedin.com/pulse/securing-iot-arvind-tiwary

© 2017 Secure Anchor Consulting

Technet24



When you do a default install, most servers will provide header information if you directly connect to a port and want to access a service. This header information often provides the name and version of the service often along with other details. From an adversary perspective, this information can be very valuable because it allows for a more focused attack. For example, if an adversary sees that TCP port 80 is open on a server, most likely it is a web server, but the adversary has no idea what version or what exploit to use. Therefore, they have to try to determine the service and version before they attack. Providing the detailed information makes life very easy for an adversary. Prior to Active Defense becoming a common field, the security advice we would always provide to clients was to remove that information—do not provide any details and keep it blank. This way, the adversary has no information, and you make them perform the extra work to figure out the service. While this is good advice, with Active Defense there is another option…

Since the goal of Active Defense is to make it more difficult, instead of leaving it blank, what if we put in false information? If the web server you are running is actually Apache, what if we change the field to say IIS 7.0. This will have no negative impact on the service or anyone connecting to it, but will now provide deceptive information to the adversary. If the adversary believes that it is actually IIS, they are going to try a series of exploits and attacks for IIS. Since the real service is not IIS, they will not work. Depending on the level of expertise of the adversary, they will get frustrated, give up and move on (likely, if they are a low-level attacker). For advanced adversaries, they will eventually figure out what is going on, but not until after they have wasted a lot of time trying the wrong attack. Either way, mission accomplished.

NOTE: Some students will have a question about how you can actually see the header information. In class, students will often say that when they connect to a website with their browser or use their e-mail client that they do not see any of this information. This is because the clients will shield this information. In order to see it, you need to make a raw connection to the service. While there are many tools for doing this, one of my favorites is netcat (nc.exe). This tool will let you make a raw connection, and then you will see all of the header information that is provided.

© 2017 Secure Anchor Consulting

Decoy IP's

- Host discovery is utilized to identify active hosts on a network
- Allows adversary to focus only on live IP's
- Decoy IP's provide feedback to adversary to have them believe that unused IP addresses are active
- The more "active" addresses the more scanning and the more resources an adversary has to use
- Can also be done with virtual machines, where decoy servers or honeypots are deployed

All attacks and all attackers are different, but, generally, in order for an attacker to break into a system and compromise it, there are 3 conditions that need to exist: a visible IP, an open port, and a vulnerable service. Therefore, one of the first things an adversary does is determine the IP address range that belongs to target. Once they know the range, they want to determine which systems are active by performing a variation of ping sweeps across the network. Hping is a tool for performing this type of activity. The more systems that are active the more work the adversary has to perform. Decoy IP's focuses on tricking the adversary into thinking there are more active IP's on the network than actually exist so they have to spend more time scanning.

With decoy IP's, there are two general approaches. One approach is having a gateway device, like a router or firewall, respond to all host discovery requests making the adversary think that every IP address has an active system connected to it. Most adversaries are smart enough to know that this is probably not the case and that they are being deceived; however, they still do not know which ones are active and which ones are not. The other variation is to utilize virtual machines, or other software, that will reply back on certain IP's making the adversary now think there are a lot more systems actually present and having to utilize additional resources to perform the scanning. With virtual machines, this takes a little longer to configure and set up, but is the ultimate deception with decoy IP's because they are actually systems. Therefore, even if the adversary does perform validation, they are actual systems and will still have to scan them. With some of the other decoy IP addresses schemes such as having a router impersonate a server, an adversary would potentially be able to figure out it is not a real system, but even if they do, it still required time and resources and slowed down the overall attack.

© 2017 Secure Anchor Consulting

Technet24

Decoy Ports

- Once an adversary finds an active host, they will typically perform a port scan to find open ports
- This will allow for focused attacks
- Decoy ports will make an adversary believe there are more ports open than actual exist
- Typically implemented through a firewall
  - When a SYN packet is sent to an IP, the firewall will always reply with a SYN-ACK whether a port is open or not

As mentioned on the previous slide, the second step that an adversary needs to perform to compromise a system is finding an open port. The more open ports, the more opportunity for compromise, but also the more work that has to be performed. The idea behind decoy ports is when an adversary performs a port scan, they receive a long list of open ports, in which only a few are active or real. It is important to give you a friendly warning because some clients are not careful in implementing active defense and actually create a larger vulnerability than what they are trying to protect against. Some clients will actually run services that open up additional ports to slow down the adversary. The problem is that by actually running the service that opens up the port, not only does it create additional work with patching, but it increases the attack surface and makes you more vulnerable. Plus, if there is an actual vulnerability, you have a created a new vector for the adversary to use to compromise your system. Therefore, we do not recommend running the actual service to create a decoy port.

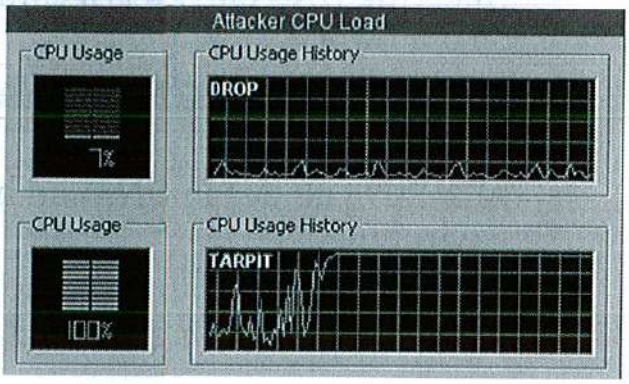
The two options that are recommended for creating decoy ports are having it implemented at a gateway device or setting up listening software on a server. Software can be installed on a server that causes ports to appear to be opened but are not connected to an actual service. To the adversary, this gives the impression that there are many entry points into the system. The benefit of this approach is that there is no gateway device required and can be implemented individually and customized on a host by host basis. The drawback is that you have to install and run software on a system and because the attacker believes the ports are open, will hit the server with additional packets. This could cause stability or reliability issues on the system. While the software approach does work in specialized cases, typically a network-based device is recommended.

Performing decoy ports with a network device, the device would intercept port requests and respond as if the port is open. This can be done by most modern firewalls and is fairly easy to configure. This is also very scalable in that it can be performed for all systems or servers sitting behind the firewall. The problem is most devices reply for all connection requests, so if an adversary is performing a full port scan it would look like 65,535 ports are open. Most adversaries would know that this is not true and that a deception technique is being used, but they still would not know which ports are open and which services to target.

© 2017 Secure Anchor Consulting

Tarpits

- Active form of decoys with the goal of drawing in the attacker and slowing down or using up all resources
- Typically done by manipulating the windows size within a TCP packet
- By setting a very small window size will keep a connection open utilizing resources
- When done across a high number of ports can have a big impact on adversary's resources



Most of the techniques that we have discussed so far are deceptive in that they provide false information to mislead the attacker; however, they are not aggressive in terms of actively doing anything to the attacker or their system. Tarpits is an aggressive technique that will send information to the adversary's computer causing it to use up all of its resources or behave in a very slow manner. The idea behind a tarpit is that if you are actively trying to attack my system, it will send information to cause your computer to get caught up in the "tarpit", which will utilize your system resources so it cannot perform any other functions.

The most common technique for tarpits is to manipulate the window size with TCP packets. Most people recognize that one of the benefits of TCP is confirmed receipt of information being sent. What many people fail to realize is that TCP can also do flow control. With flow control, each side tells the other side how much information they can send before receiving an acknowledgement. This communication is done through the use of a window size. The idea behind tarpitting is as soon as the adversary makes a connection to the target system, the target system will set a very small window size or periodically set the window size to zero. This means the adversary can only send a very small amount of data before it receives an acknowledgement, or will not be able to send any data if the window size is zero. By doing this, connections can be kept open for a long period of time, and the system will utilize significant resources maintaining the connections.

Reference

1. HTTP DDoS Attack Mitigation Using Tarpitting - https://www.secureworks.com/research/ddos

© 2017 Secure Anchor Consulting

Technet24

Bogus DNS Entries

- Querying DNS entries is a common tactic to find critical systems on a network
- Bogus DNS entries can mislead an adversary to what they believe might be a database system or a critical server
- Attractive names in DNS can be redirected to honeypots or jailed environments
- Can also be achieved with host files if adversary is setup up as a pivot and performing internal reconnaissance

Most organizations have domain names tied to servers to make it easier to connect to a system. Instead of having to remember the IP address of a server, the person would utilize a domain name. The client system would go to the local DNS server to resolve the domain name to an IP address. Since, internally, many domain names are descriptive names, this could provide information to an adversary of which system to target. For example, mail.samplecompany.com might be the domain name for a mail server at an organization. Since attackers will often query DNS servers to find out systems to target, bogus DNS entries can be another form of Active Defense.

Bogus DNS entries can also be combined with some of the other Active Defense methods that we discussed. For example, a honeypot in a jailed environment could be given an attractive domain name to look like a database system. Bogus DNS entries can also be tied to decoy IPs or decoy ports to lead the adversary on a treasure hunt where instead of finding a treasure, they find a lot of lost time looking for a vulnerability that does not exist.

Reference

1. Active Defense Through Deceptive Configuration Techniques - https://www.sans.org/reading-room/whitepapers/ActiveDefense/active-defense-deceptive-configuration-techniques-36692

© 2017 Secure Anchor Consulting

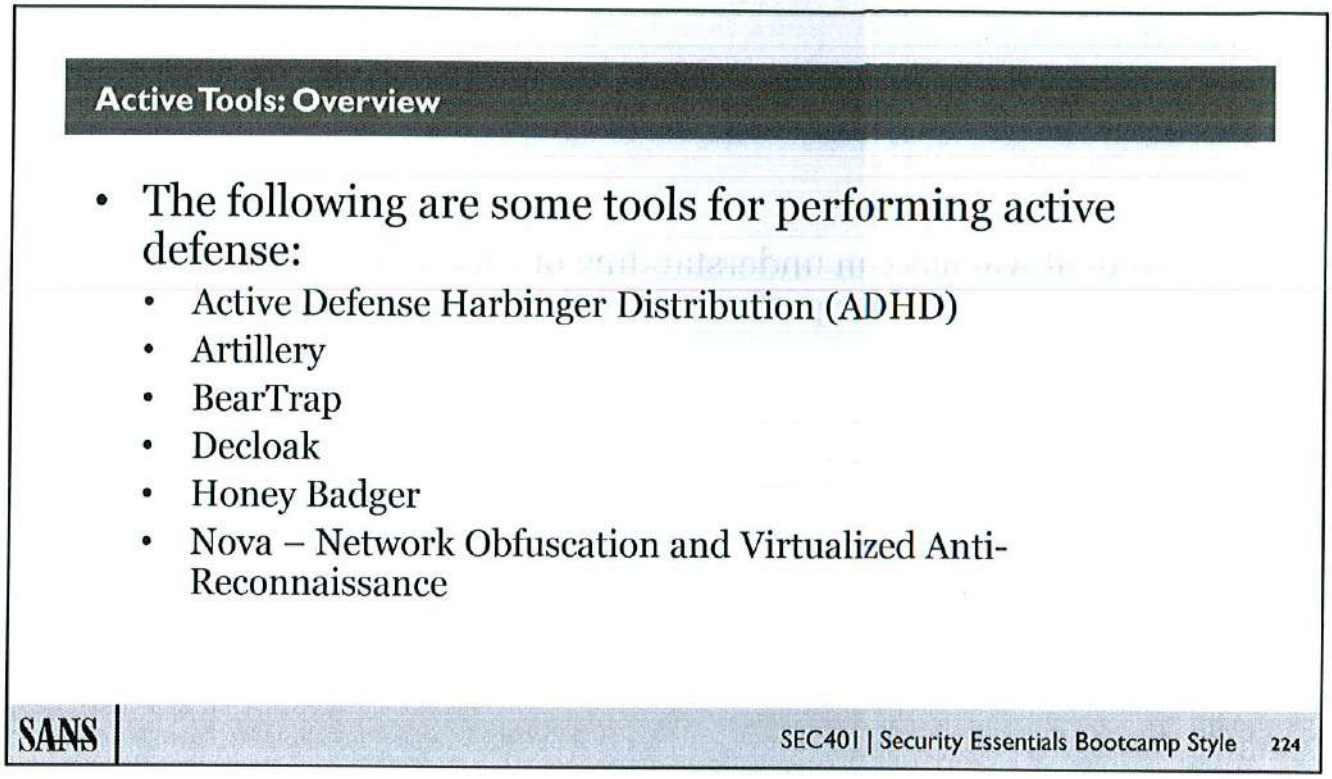
Active Defense Tools

The student will have an understanding of various tools that can be used for performing active defense

This page intentionally left blank.

© 2017 Secure Anchor Consulting

Technet24



As Active Defense and Offensive Countermeasures continue to become viable methods of cyber security, more and more tools are becoming available to help organizations implement these measures. It is important to remember that Active Defense can be as straight forward as adding in additional DNS entries or change the host header for a key service, which does not necessarily require additional tools. However, in some cases having tools can make it a lot easier to implement these countermeasures in a scalable and predictable manner.

It is important to note that there are a lot of new tools being developed and this is becoming a very active area for startups. While this section focuses in on some of the free tools available, it should not be taken as a complete list because new technologies are coming out on a regular basis. Also, there are more and more commercial companies being formed in this area so before making a purchase it is always important to do your research.

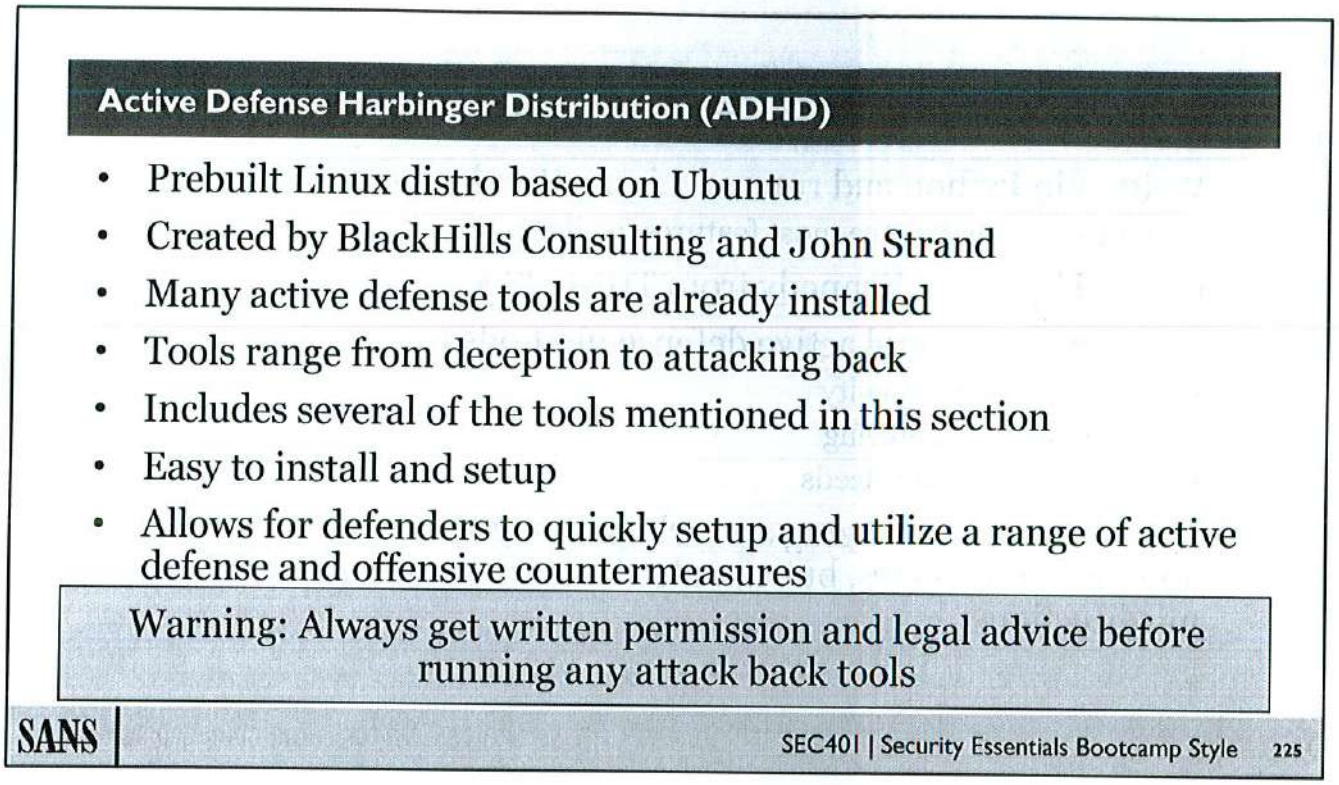
The following are some of the tools that will be covered in this section:

- Active Defense Harbinger Distribution (ADHD)
- Artillery
- BearTrap
- Decloak
- Honey Badger
- Nova – Network Obfuscation and Virtualized Anti-Reconnaissance

References

1. Implementing Active Defense Systems on Private Networks - https://www.sans.org/reading-room/whitepapers/detection/implementing-active-defense-systems-private-networks-34312
2. Free 'Active Defense' Tools Emerge - http://www.darkreading.com/attacks-breaches/free-active-defense-tools-emerge/d/d-id/1140109?

© 2017 Secure Anchor Consulting



One of the problems with many security tools is that they run on Linux and requires downloading, configuration and setup. This cannot only be time-consuming, but also be difficult and frustrating to get the tool properly installed. Prebuilt Linux distros (such as Kali) are becoming more and more popular. New distros are also being built for specialized purposes. Instead of having to research and find a lot of tools, if you are interested in a particular domain, you would download a distro that is focused on that domain, so you have all the tools you need installed and ready to go.

Our good friends from BlackHills Consulting and John Strand built the Active Defense Harbinger Distribution (ADHD), which is a distro focused on active defense and offensive countermeasure tools. It is not only very easy to use, but they have descriptions explaining how the tools work and how to run the tools on your system. If you want to use or play with a wide variety of active defense tools, it is highly recommended you download and run ADHD. The distro has a range of tools from deception to attack back capabilities.

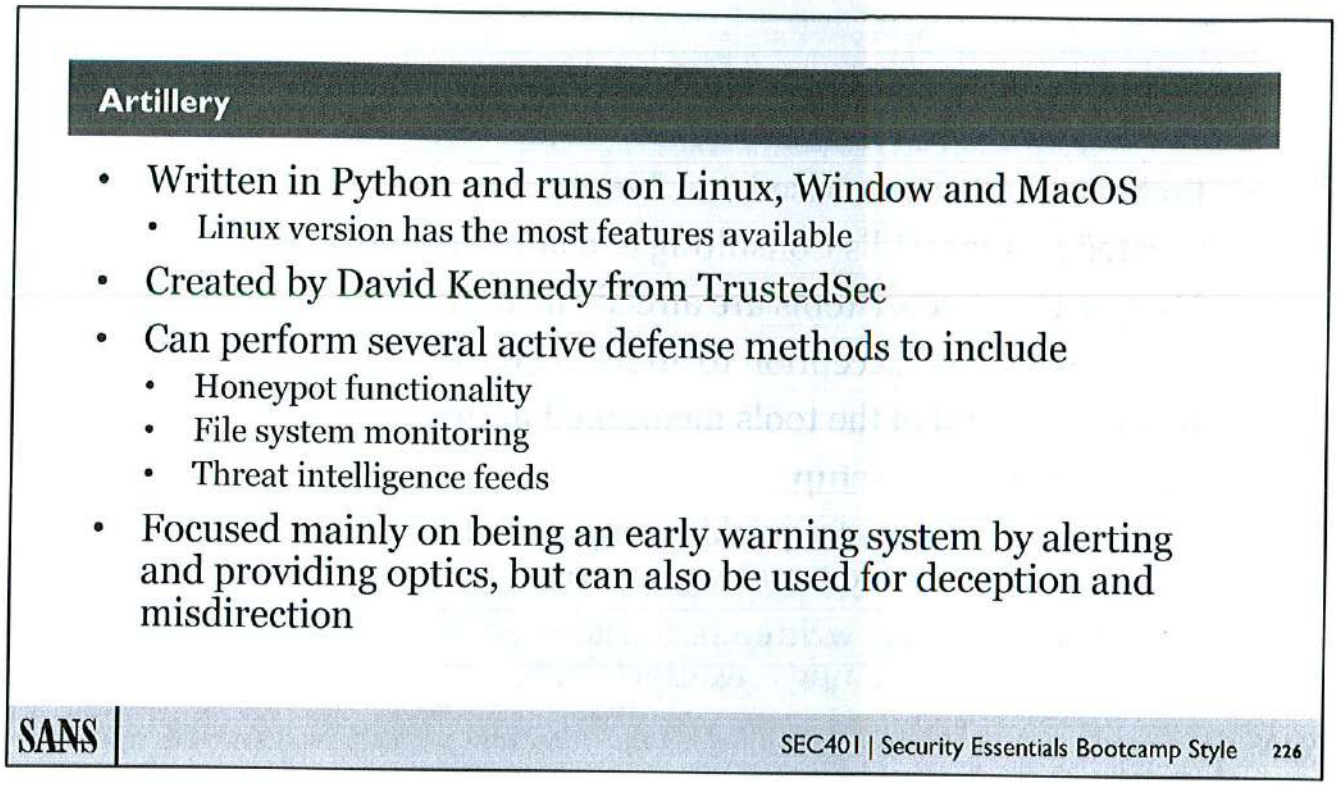
Since the ADHD does contain active tools, it is important to always get written permission and seek legal advice before actively running these tools against live targets. Even with permission, they should still be tested in a production environment.

Reference

1. http://www.blackhillsinfosec.com/?page_id=4419

© 2017 Secure Anchor Consulting

Technet24



Artillery is a tool based on providing an early warning system to administrators to give an indication of when someone is trying to break into a system and potentially how they are breaking in. One of the methods used by Artillery is that it will open up several ports on the system giving the adversary the perception that multiple ports and services are open on the system. While the main functionality is for monitoring, this can also be used as a deceptive technique to trick the adversary, fooling them into thinking a large number of ports are open on the system.

Some of the functionality that is supported by Artillery is:

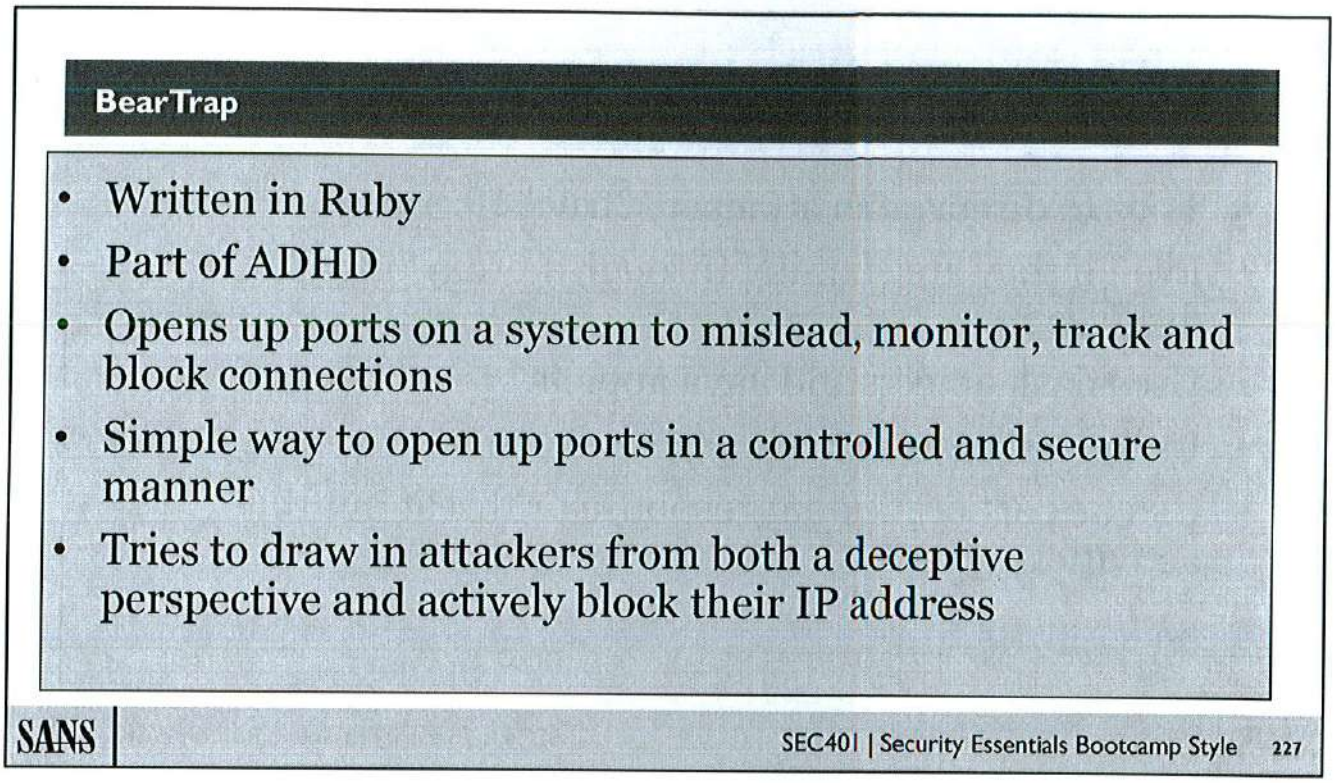
- Honeypot functionality – By opening up additional ports on the system, these additional ports have no legitimate function. Since they are honeypots, no one should be connecting to these ports. Therefore this can also be used as a decoy.
- File system monitoring – Artillery can monitor key files on the system and look for changes to the system. This is typically more of a detection capability than pure active defense but can be combined with honeytokens. The general concept is honeytokens will be created as a deceptive technique to mislead the adversary and Artillery can monitor for any potential activity on these files.
- Threat intelligence feeds – Artillery can also integrate with existing threat intelligence feeds to look for any previous attacks or addresses of attackers to provide more insight and optics into the adversary and what they are doing. This capability can be useful if active defense trace-back capability is desired.

Artillery is written in Python and while it does run on multiple operating systems, the Linux version has the most functionality and capabilities. It was created by David Kennedy from TrustedSec.

References

1. Implementing Active Defense Systems on Private Networks - https://www.sans.org/reading-room/whitepapers/detection/implementing-active-defense-systems-private-networks-34312
2. Artillery - https://www.trustedsec.com/artillery/

© 2017 Secure Anchor Consulting



BearTrap is a fairly simple tool, but still a very effective tool to have as part of your cyber security arsenal. One of the types of Active Defense that we discussed was decoy ports. We discussed the challenges of having open ports on a system but not running the real service because it could create a point of compromise. BearTrap is one option. BearTrap is written in Ruby and comes as part of the ADHD.

BearTrap opens up additional ports on a system to give the perception that additional ports are open. This provides a simple and scalable, but safe, way to open up ports or trick the adversary into thinking that the ports are open with a real service. This not only slows down the adversary, but confuses them into thinking that more ports are open than are actually legitimately open on the system.

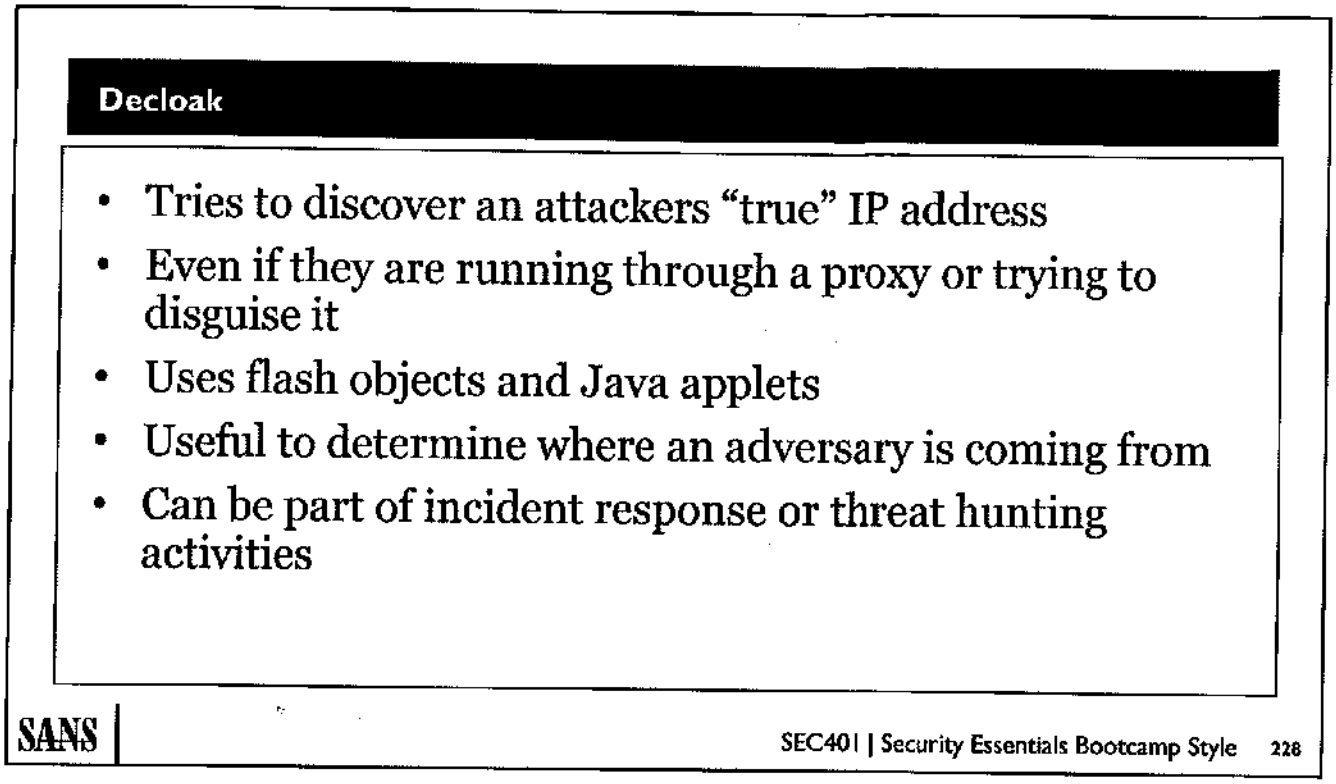
In addition, BearTrap also provides some active components to automatically block an IP address that connects to one of the unauthorized ports. Since these ports are not legitimate and should not be open, any connections can be deemed as potential hostile activity.

References

1. BearTrap - https://sourceforge.net/p/adhd/wiki/BearTrap/
2. Free 'Active Defense' Tools Emerge - http://www.darkreading.com/attacks-breaches/free-active-defense-tools-emerge/d/d-id/1140109?
3. The Offensive Approach to Cyber Security in Government and Private Industry - http://resources.infosecinstitute.com/the-offensive-approach-to-cyber-security-in-government-and-private-industry/#gref

© 2017 Secure Anchor Consulting

Technet24



For investigations and incident response purposes it is often important to know the "true" IP address of the attacker. As we know, looking at the source address in the IP header can be very dangerous because it can be spoofed, run through a proxy or an attacker can use an anonymizer that hides the real IP address. Decloak is meant to solve this problem by trying to identify the true address that is used by an adversary.

Once again it is important to remember that no tool is perfect but utilizing decloak can help to try and demystify the true identity of an adversary, which is important in understanding an attack and the true intent of an adversary.

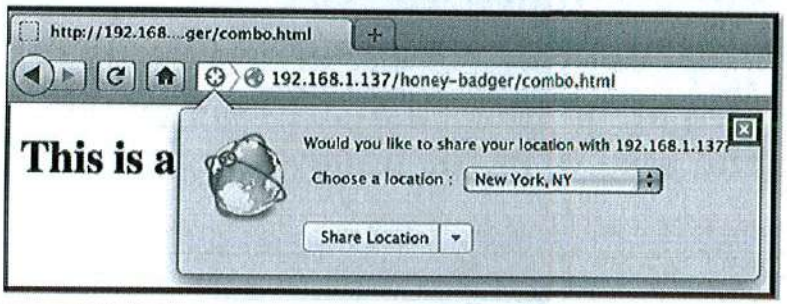
References

1. Decloak - https://sourceforge.net/p/adhd/wiki/Decloak/
2. DDOS Attacks - http://slideplayer.com/slide/7790066/

© 2017 Secure Anchor Consulting

Honey Badger

- Determines the physical location of a system
- Utilizes geolocation, Wi-Fi, and IP address
- Pretends to be an administrator interface
- Runs a java applet on the local machine



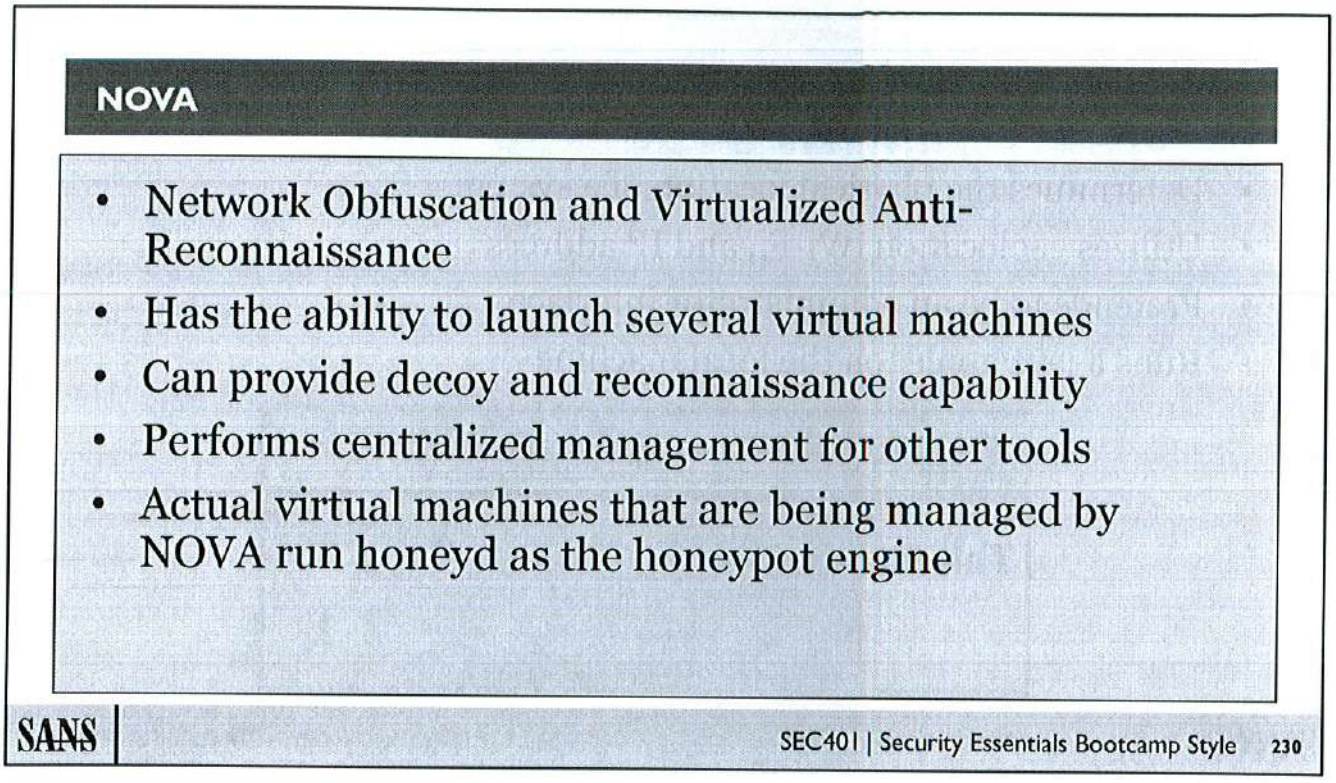
Honey Badger falls under the category of attack back with the goal of determining the physical location of a server. It will often do this by pretending to be an administrator interface to trick the adversary to connect. When the adversary connects, it runs a java applet on the attacker's system with the goal of trying to determine its physical location. Since code is actually dropped and run on the attacker's system, it could also technically crash or cause other problems. If someone is coming from a legitimate location and accidentally connects to the interface, honey badger could inadvertently cause harm. Honey badger is a powerful tool in your arsenal but as with any offensive tool, you need to be careful when and where you utilize it.

References

1. https://sourceforge.net/p/adhd/wiki/Honey%20Badger/
2. The Active Defense Harbinger Distribution - http://www.hacker10.com/other-computing/the-active-defense-harbinger-distribution/

© 2017 Secure Anchor Consulting

Technet24



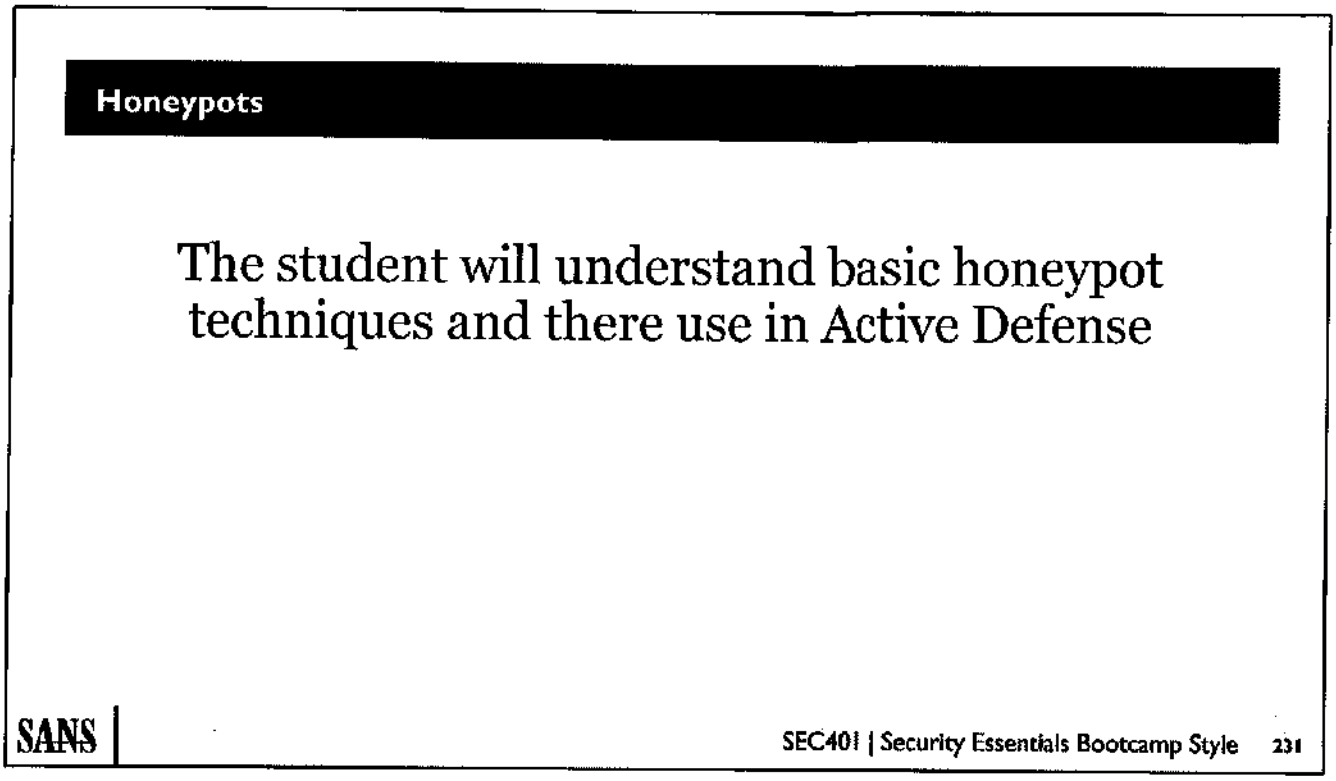
NOVA stands for Network Obfuscation and Virtualized Anti-Reconnaissance and has the ability to launch, manage and control several virtual machines. NOVA can act as the centralized management console for all of the virtual machines. The various virtual machines that are being managed are called haystacks. The actual virtual machines that are being managed by NOVA run honeyd as the honeypot engine.

NOVA does have deception capability by creating a labyrinth of honeypots, making it hard to differentiate between real systems and honeypots. However, this could use up significant resources and slow down the adversary. In addition to this benefit, NOVA is really focused on utilizing various machine learning techniques to identify unusual traffic across the various honeypots and provide an early warning system to detect attackers.

References

1. Implementing Active Defense Systems on Private Networks - https://www.sans.org/reading-room/whitepapers/detection/implementing-active-defense-systems-private-networks-34312
2. Network Anti-Reconnaissance Tool: Nova - https://n0where.net/network-anti-reconnaissance-tool-nova/

© 2017 Secure Anchor Consulting



This page intentionally left blank.

Technet24

What a Honeypot Is

- A honeypot is an information system resource that has no legitimate purpose or reason for someone to connect to it
- Two main purposes:
  - Draw in attackers to understand how they break into a system
  - Better determine what is attack traffic so defense measures can be improved
- It's an advanced technique that is usually deployed after other security measures have been implemented

There is no authorized activity on a honeypot. Any interaction with the honeypot is accidental or hostile in nature.

The ultimate goal of security is to reduce or eliminate risks to an organization's critical assets. Ideally, we prefer to do this by preventing attacks, but one of the key mottos of information security is, "Prevention is ideal, but detection is a must." We must realize that an organization's key resources will be attacked, and we have to be ready to detect the attack as early in the cycle as possible and take advantage of this when it does occur. One way of doing this is with honey-x technology, such as honeypots.

The functionality of honeypots is so diverse that it has been a challenge to define exactly what a honeypot is: Honeypots serve many different purposes for different organizations. Generally, a honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource. In fact, its value lies in its being misused. The information system resource might be

- Dedicated server
- Simulated system or state machine
- Service on a selected host
- Virtual server
- Single file with special attributes, which is sometimes called a "honeytoken"

The value in a honeypot is derived from the lack of any authorized activity to the resource. A honeypot resource is never meant for legitimate use; therefore, any use of the honeypot resource is illegitimate and accidental, or hostile, in nature.

When most people hear the term "honeypot," they think of a system that you unpatch, put on the Internet, and hope it gets broken into. Although this works well for pure research where a site does not have critical systems, it does not scale to a typical DMZ. You do not want your DMZ to be attacked or get compromised. If you have critical systems on your DMZ, you need to keep an attacker away. You do not want to draw them in with an un-patched system.

© 2017 Secure Anchor Consulting

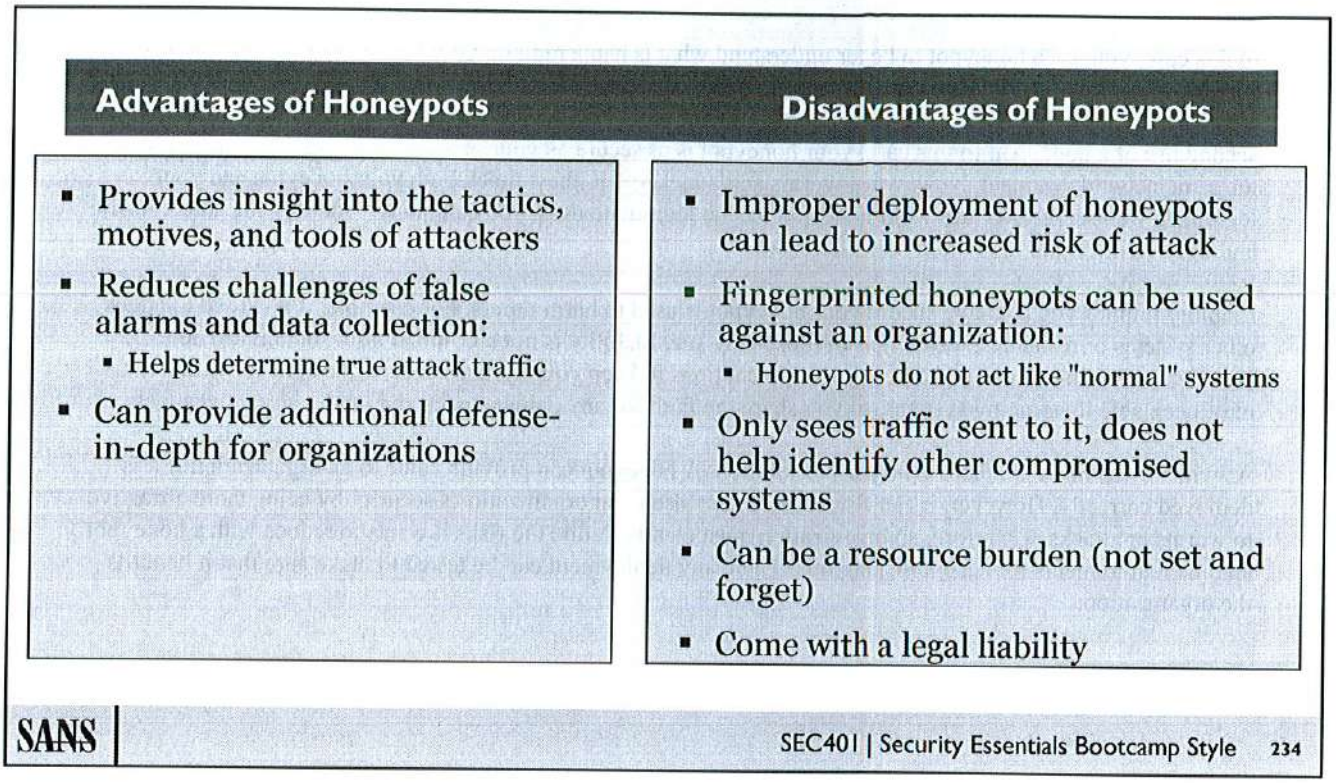
In this case, you use a honeypot to better understand what is happening on your key systems. A typical web server can get millions of hits a day. Attempting to identify the difference between legitimate connections and attackers is impossible. This is the case unless you have an easy way to discern attack traffic; thus, you have the second use of a honeypot. In this case, your honeypot is as secure as your production web server and is put on the same network segment. Now, when worms and attackers hit, they attack both your honeypot and your legitimate web server. Because your honeypot has no legitimate uses, you can quickly identify the attack traffic and use that information to build better defenses.

Liability implies you could be sued if your honeypot is used to harm others. For example, if it is used to attack other systems or resources, the owners of those may sue. Liability is not a criminal issue, but civil. The argument being that if you had taken proper precautions to keep your systems secure, the attacker would not have been able to harm my systems, so you share the fault for any damage occurred to me during the attack.

As with any technology, there is no perfect solution. A honeypot can provide value to an organization if it is deployed correctly. However, it can also cause a decrease in an organization's security by being more attractive to worms or attacks. Therefore, an organization must clearly define the risks it wants to reduce with a honeypot and the requirements for accomplishing this. Then, any deployment can be tested to make sure that it benefits the organization.

© 2017 Secure Anchor Consulting

Technet24



If deployed properly, honeypots play a critical role in the network security arsenal. There is no silver bullet or perfect solution when it comes to network security. Therefore, it is important to understand the advantages of a technology to make sure you deploy it correctly. Some of the advantages of honeypots are:

- Provides insight into the tactics, motives, and tools of attackers
- Reduces challenges of false positives, false negatives, and data collection by determining true attack traffic
- Provides additional defense-in-depth for organizations
- Method of active defense

One of the primary purposes organizations and researchers deploy honeypots is to learn about the tactics and motives of attackers. By utilizing honeypot technology and watching how attackers compromise systems and what they do after the system is compromised, we can identify the tools they use, their skill levels, and their motives for attacking systems. In addition, on large-volume networks, honeypots can help us focus on the attack traffic, providing a straightforward way to isolate the legitimate traffic.

The question of motive is often useful to organizations on a case-by-case basis. When we detect attempted (or successful) system compromises, we usually have little opportunity to determine whether the attack was the result of random selection or whether it was specifically targeted at the victim's organization.

Using a honeypot, researchers can watch the tactics of attackers after they compromise a system. Does the attacker start to scan for more systems to compromise; if so, do they use the same exploit that compromised the honeypot, or do they direct their analysis toward more valuable internal resources, such as database systems?

We mentioned one of the critical factors of honeypots is that the honeypot is deployed without authorized uses. By default, this makes any use of the honeypot accidental or hostile, but always unauthorized. In intrusion detection, we speak of the challenges of false positives (alerts on benign activity) and false negatives (the lack of alerts for hostile activity). When we deal with honeypots, we eliminate almost all of the risks of

© 2017 Secure Anchor Consulting

false positives and false negatives. All data associated with a honeypot (whether it is network traffic, application utilization, or use of the honeypot resource) is logged. Because we can log the data associated with the honeypot, we have a significant advantage over traditional signature-based IDS techniques. Signature-based IDSs generate alerts only for known hostile activity, whereas the honeypot system can capture and identify hostile activity from exploits that are not currently known.

One the most significant reasons organizations should not deploy honeypots is the risk of misconfigured honeypots, which increases the threat to other production networks. The last thing you want to give an attacker is a platform from which to extend the attack into the rest of your network. If an attacker can compromise a honeypot system and the rest of the production systems aren't sufficiently protected from the perspective of the honeypot, it is likely that the network is in significant jeopardy. An organization wants to keep attackers off its network, not give the attackers a foothold behind the firewall.

Another disadvantage of honeypot technology is the risk of honeypot fingerprinting. In general, you don't want the attacker to identify the target system as a honeypot. Although you hope this would make an attacker move on to a different network to avoid being detected by another honeypot they weren't able to identify, this is seldom the case. Attackers use the identity of a honeypot to throw off administrators by spoofing traffic from a legitimate system to the honeypot system, or worse, the attacker feeds the honeypot incorrect information about tactics and motives. Although administrators struggle to make sense of the attacks against the honeypot, the attacker might try to leverage the confusion generated and attack other production systems instead.

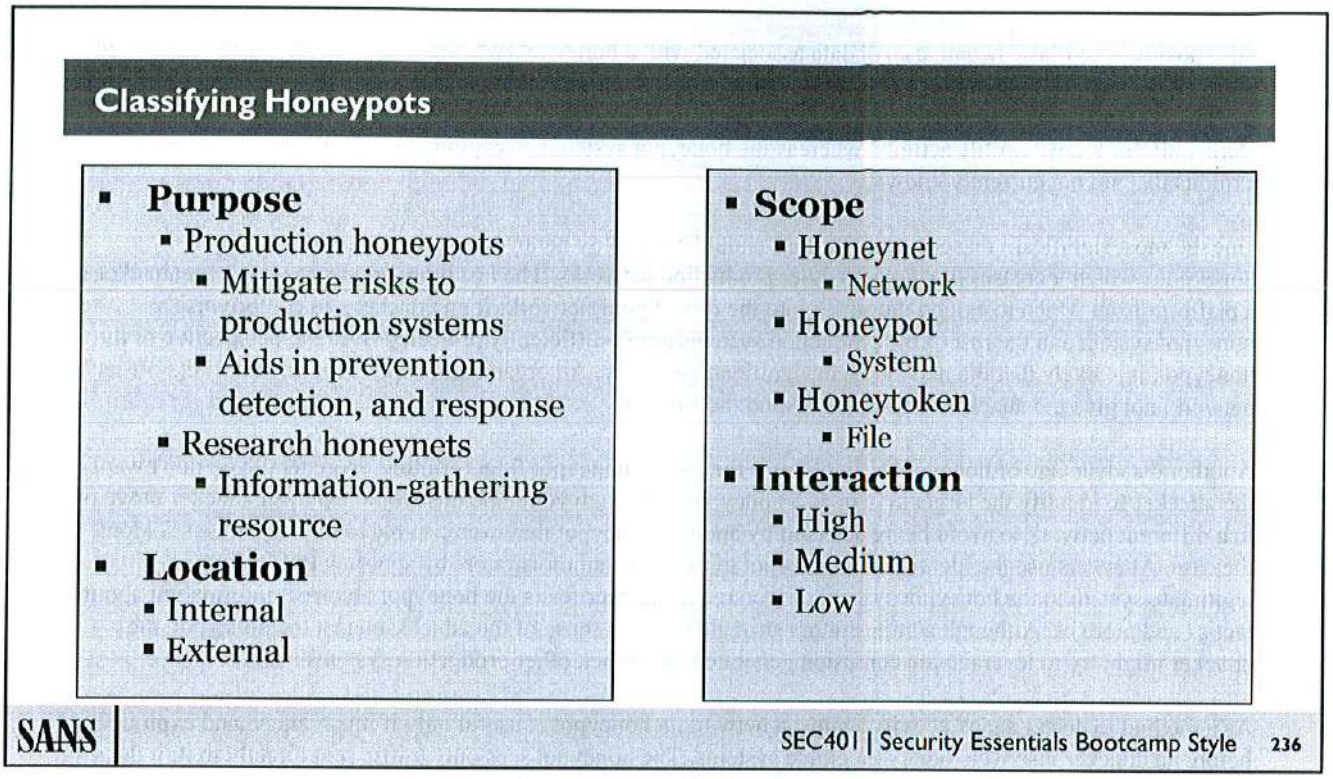
As a method to detect attack activity against a network, a honeypot is useful only if it is scanned and exploited before an attacker discovers other vulnerable systems. The honeypot sees only traffic sent directly to it; it does not identify other systems that might be compromised before the attacker reached the honeypot system. The honeypot is not an intrusion detection system (IDS) that sees all traffic. It sees only traffic going to that specific system. Just because a honeypot does not see attacker traffic, it does not mean the network is properly protected.

Honeypots can also be a resource burden for organizations. Honeypot technology is not a set-and-forget option; the deployment of honeypots requires constant monitoring, swift responses, and detailed analysis of attacks on compromised honeypot systems. If your organization is already overly burdened with other security measures, deploying honeypot technology does not help reduce that burden. Deploying limited-interaction honeypots limits the amount of resource burden that is required to maintain the honeypot, but it still requires resources for honeypot monitoring.

One of the more complex issues surrounding honeypot technology is the variety of legal issues that can affect organizations. Serious legal consequences can arise from the use of a honeypot. Organizations are encouraged to consult with legal counsel before deploying honeypots. In addition, if a honeypot is used by an intruder to attack other systems downstream, the operator of the honeypot might find himself embroiled in litigation by the downstream victims for facilitating the attack and failing to take steps to prevent use of the system. An operator also might face liability if he learns of attacks against others to whom the operator owes a duty of care but fails to notify the other victims. The honeypot operator also might find himself in the precarious position of having "stolen" information or other contraband stored on the system.

© 2017 Secure Anchor Consulting

Technet24



A honeypot is a general technology an organization deploys. However, depending on the classification of the honeypot, its role and function can significantly differ. There are four general categories for classifying a honeypot:

- **Purpose**: What is the reason for deploying a honeypot?
- **Location**: What is the visibility or location of the honeypot?
- **Scope**: What will the level of deployment be?
- **Interaction**: How much interaction is the attacker supposed to have with the system?

When referring to the purpose of a honeypot, we classify them into two major categories: production and research.

Production honeypots are used as a measure to increase the defense-in-depth measures for your network. An example might be a honeypot that monitors scanning activity from an attacker to multiple honeypot addresses that are otherwise not in use and blocking the source address before they get to production resources. Deploying production honeypots can give you considerable visibility into the approaches of attackers who target your network. In addition, the honeypot hopefully takes the brunt of attacks that otherwise might be directed against your production systems.

Research honeypots are used to study the techniques and motives of attackers. Although they can be used as a resource to aid in prevention, detection, and response, they are specifically designed to record the activities of an attacker and their tools.

Through research honeypots, we can determine the motives of attackers—whether they are socially motivated (such as participating in a group of attackers), financially motivated (collecting credit-card numbers, for example), or any other potential motives.

The configuration of a honeypot and the type of information you expect to receive is based on the visibility or location of the honeypot. The Internet is full of attackers, so it is not a surprise that an external honeypot gets a

© 2017 Secure Anchor Consulting

large number of connections. Typically, the purpose of external honeypots is to understand attack traffic and use this information to build better defenses.

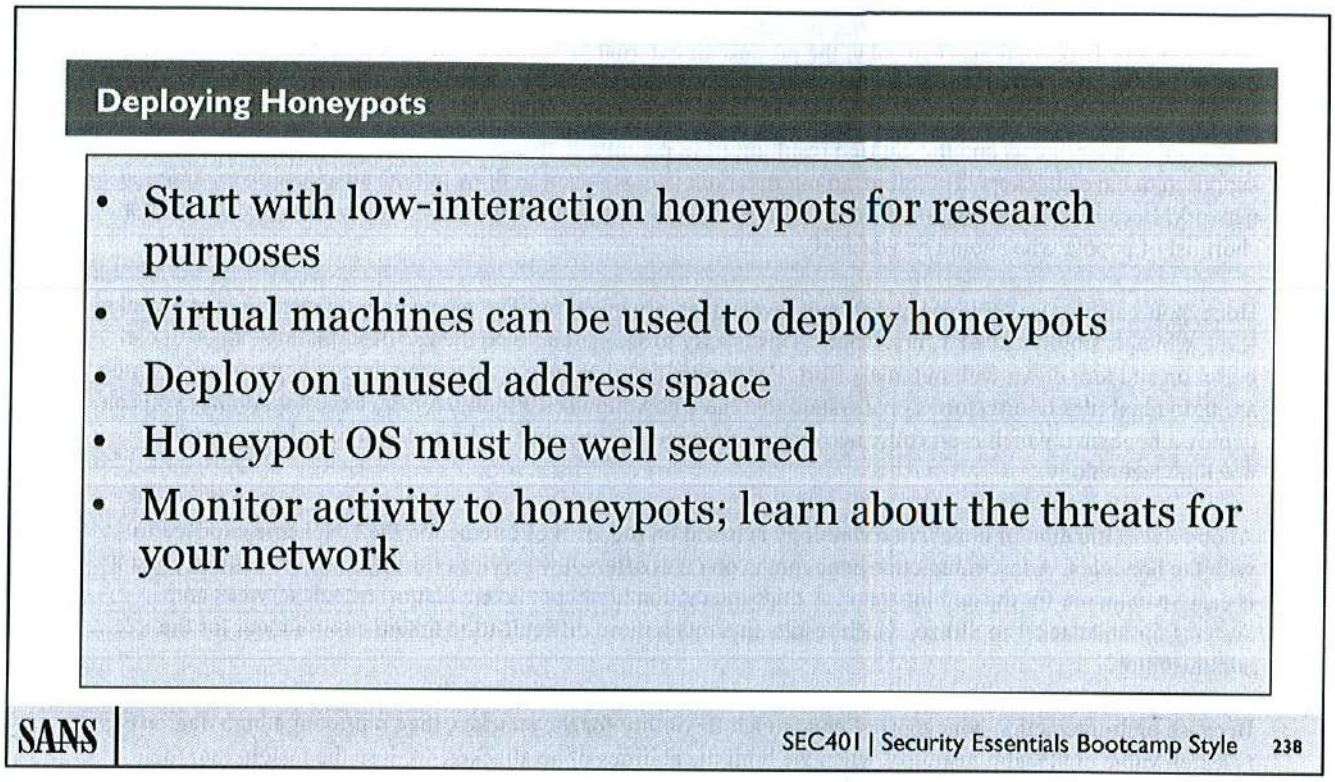
However, insider threats and the damage resulting from it continue to grow. An internal network is not supposed to have attackers. Therefore, an internal honeypot is meant to draw out the attackers on the internal network. Because there should not be a lot of internal attackers, this enables the security manager to create a short list of people who should be watched.

Honeypots can also be deployed at different levels in an organization. The most common method is at a system level, which is simply called a "honeypot." This is easy to set up, and with the use of virtual machines, it can be set up and torn down with minimal effort. For a more detailed analysis, you can deploy honeytokens, which are individual files or directories on a system that have no valid uses. Finally, for the largest visibility, you can deploy a honeynet, which is an entire network of honeypots. This can also be configured utilizing virtual machine technology.

Another classification of honeypot technology is based on the level of interaction the attacker is exposed to with the honeypot. A low-interaction honeypot is one that offers few services for the attacker to utilize, and it is easy to maintain for the administrator. A high-interaction honeypot is one that offers full services and systems for an attacker to utilize, which results in a much more difficult-to-maintain environment for the administrator.

Because high-interaction honeypots offer so much flexibility for the attacker, they represent a high degree of research value to the administrator. When we limit the abilities of an attacker, we lose the insight into their tools and techniques, as in the case of the low-interaction honeypot.

One of the challenges information security researchers face is the lack of information about the techniques of attackers. When systems become compromised, few organizations are willing to disclose the details surrounding the attack for fear of tarnishing their public image. Conversely, attackers readily share information about exploiting systems as a measure of status or to increase their skill level with mutual information exchange. Using honeypots, we can watch and examine the tools and techniques of attackers to determine how they compromise systems and what they do after a system is compromised.



When deploying honeypots, it's a good idea to start small and understand how they work. Carefully monitor the honeypots, and learn from the techniques and tactics of attackers. As your level of skill in working with honeypots increases, you can raise the level of interaction with attackers to keep learning more about attacker tactics. The following are the steps to take when deploying honeypots:

- Start with low-interaction honeypots for research purposes.
- Deploy on unused address space.
- The honeypot tool OS must be well secured.
- Monitor activity to honeypots and learn about the threats to your network.

The easiest and simplest honeypots to deploy are low interaction. With this, you gather information, but you do not interact with the attacker. The more you interact, the greater the complexity of the system and your potential liability. After you deploy a low-interaction honeypot and understand the information you receive, you can deploy additional honeypots to be more effective.

Tools such as Honeyd and LaBrea Tarpit take advantage of unused address space for production purposes. Use caution when deploying honeypots on used address space.

Honeypots that emulate the operating system of other hosts are advantageous because the attacker directs activity to the virtual host while the honeypot system remains unseen. It is still important to secure the host operating system of the honeypot controller to prevent it from being compromised and possibly manipulated by an attacker.

Finally, it is important to carefully monitor honeypot systems, ensuring that they are not manipulated for malicious purposes against other systems. Utilize an out-of-band monitoring mechanism that can alert you to activity on the honeypot, such as a modem and paging system. If you do not have the time or energy to monitor a honeypot, you should not deploy the technology.

© 2017 Secure Anchor Consulting

As with any technology, it takes a lot of work and energy to deploy a security solution. Therefore, it is critical that you plan properly and take the appropriate steps to make sure that the honeypots you deploy decrease your risks.

Reference

1. An Overview of Honeypots- https://www.blacksintechnology.net/an-overview-of-honeypots/

© 2017 Secure Anchor Consulting

Technet24

Honeypot Checklist/Summary

- Honeypot checklist:
  - Make sure that you have the resources needed to analyze the results
  - Validate whether a honeypot is the best solution
  - Determine whether a honeypot is increasing your risk
  - Identify what information you are trying to obtain from the honeypot
- Advanced technique: Do everything else first
- Honeypots are classified by interaction level and purpose (research, production)
- Capture and identify unknown threats
- Honeypots reduce complications with false positives, false negatives, and data collection
- Attackers can use the honeypot if they break the controls

To ensure an organization effectively deploys honeypots, you need to address the following key items:

- Make sure you have the resources needed to analyze the results.
- Validate whether a honeypot is the best solution.
- Determine whether a honeypot is increasing your risk.
- Identify what information you trying to obtain from the honeypot.

Honeypots are not a fire-and-forget technology. The value of a honeypot is that someone can analyze the data captured by the firewall and use it to increase security. Gathering a lot of information does not typically provide any value if no one is reviewing it; it just uses up valuable resources and increases the risk of compromise, which is not a good business decision. In addition to making sure that you have resources to analyze the results of a honeypot, make sure that it is the best use of that individual's time. For example, if your firewall rulesets have not been validated and your systems have not been hardened, it does not make sense to have administrators review honeypots; their time is better spent on higher risk tasks and critical security tasks.

Honeypots have a high "cool" factor, and many people want to deploy them. However, you need to make sure that a honeypot is the best solution for your problem.

For any high-risk situation, first, identify a list of possible solutions, perform a brief cost-benefit analysis, and then choose the most appropriate solution. For example, if a new worm propagates on the Internet, it is better to block it at the firewall than allow the traffic into the network, so that it can be captured by a honeypot.

The goal of security is to decrease or eliminate risks; you do not want to increase risks. Honeypots are meant to be scanned, connected, and compromised by an attacker, and a compromised system can potentially cause more problems than it solves. Therefore, to ensure its value, it is important that the honeypot is carefully designed and deployed. A honeypot without monitoring or analysis can quickly turn into a liability, especially if an attacker inflicts harm without your knowledge.

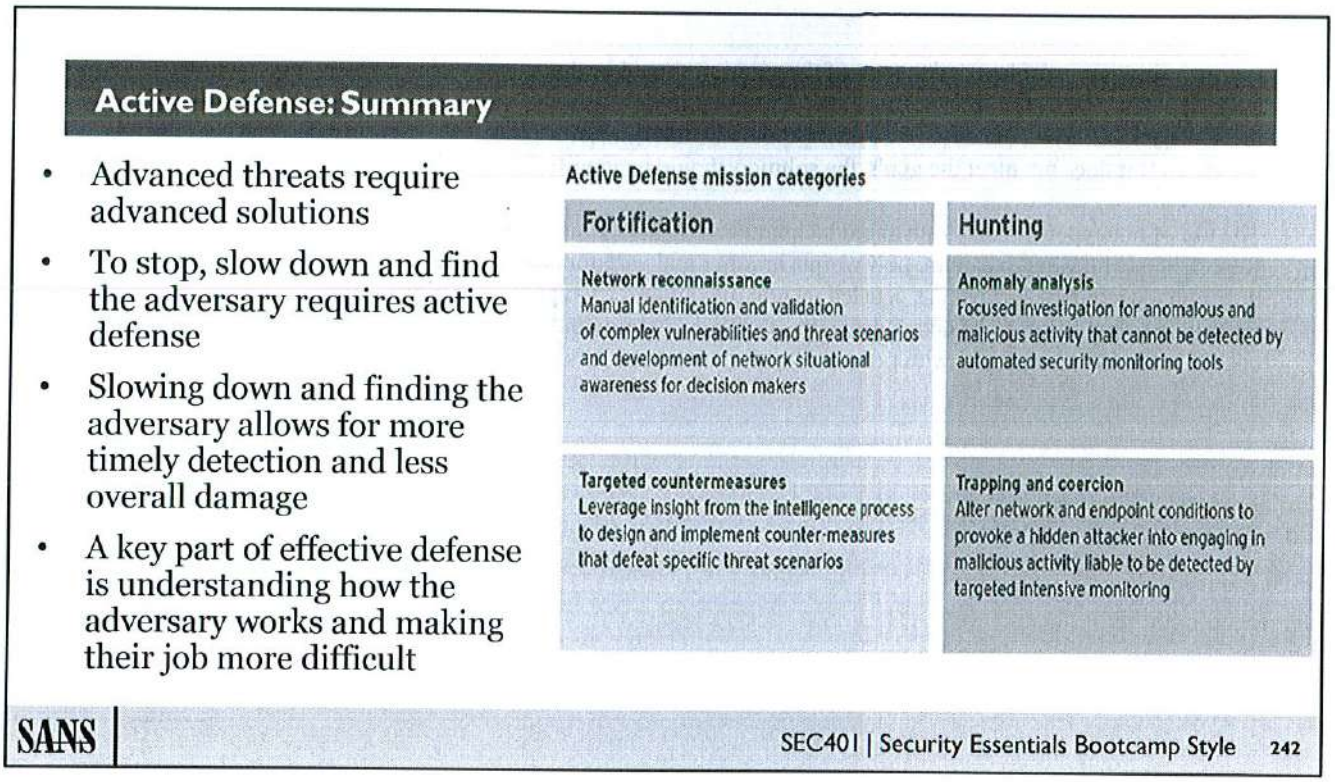
© 2017 Secure Anchor Consulting

Before a solution is deployed, the goals of the solution should be documented. Then, the goals should be mapped against a given risk. If you cannot map a solution to a risk, the solution should not be deployed. After the goals are validated, the deployed system should be tracked against the goals to ensure that it accomplishes the goals. If it does not meet the goals, the solution should be modified or eliminated.

The use of honeypots involves advanced techniques. Use honeypots after you apply other security techniques. Researchers who want to capture new worms or other malware for analysis use honeypots. For the rest of us, they provide a means to get more detailed insight into attacks attempted against our systems. Normally, firewalls prevent this type of detailed insight. It's available through logs of successful intrusions. Honeypots enable a visibility that comes with penetration, without compromising a production system.

© 2017 Secure Anchor Consulting

Technet24



If you watch the news or monitor any threat reports, it is pretty obvious that most companies are not winning when it comes to cyber defense. It seems the more time, money and resources an organization spends on security, the problem continues to get worse. This is because the adversary is adapting and changing how they attack, but many organizations are using the same methods and techniques to defend. If the adversary is fully aware of our defensive measures and knows how to get around them, continuing to take the same approach is not a recipe for success. If the adversary is advanced and adaptive than our security defenses must also be adaptive and advanced.

Active defense is focused on making things more difficult for the adversary through deception and misleading information. If the adversary has a false perception of how the network is designed and configured, the time to attack will be slower and the chance of success will be much lower. Active defense also includes methods that allow the defense to become more offensive oriented. One such method is attacking back. This is where instead of trying to prevent or stop the adversary, the defense is going to attack the adversary, find out their physical location, or actively harm their system to make it harder for them to continue to launch attacks. While there are significant legal issues associated with attacking back, as the world becomes more dangerous, this is a technique that you need to be aware.

A key motto of Dr. Cole is to stop making it easy for the adversary. Active Defense is a key component of getting more aggressive in defense and making it as difficult as possible for an adversary to cause harm.

© 2017 Secure Anchor Consulting

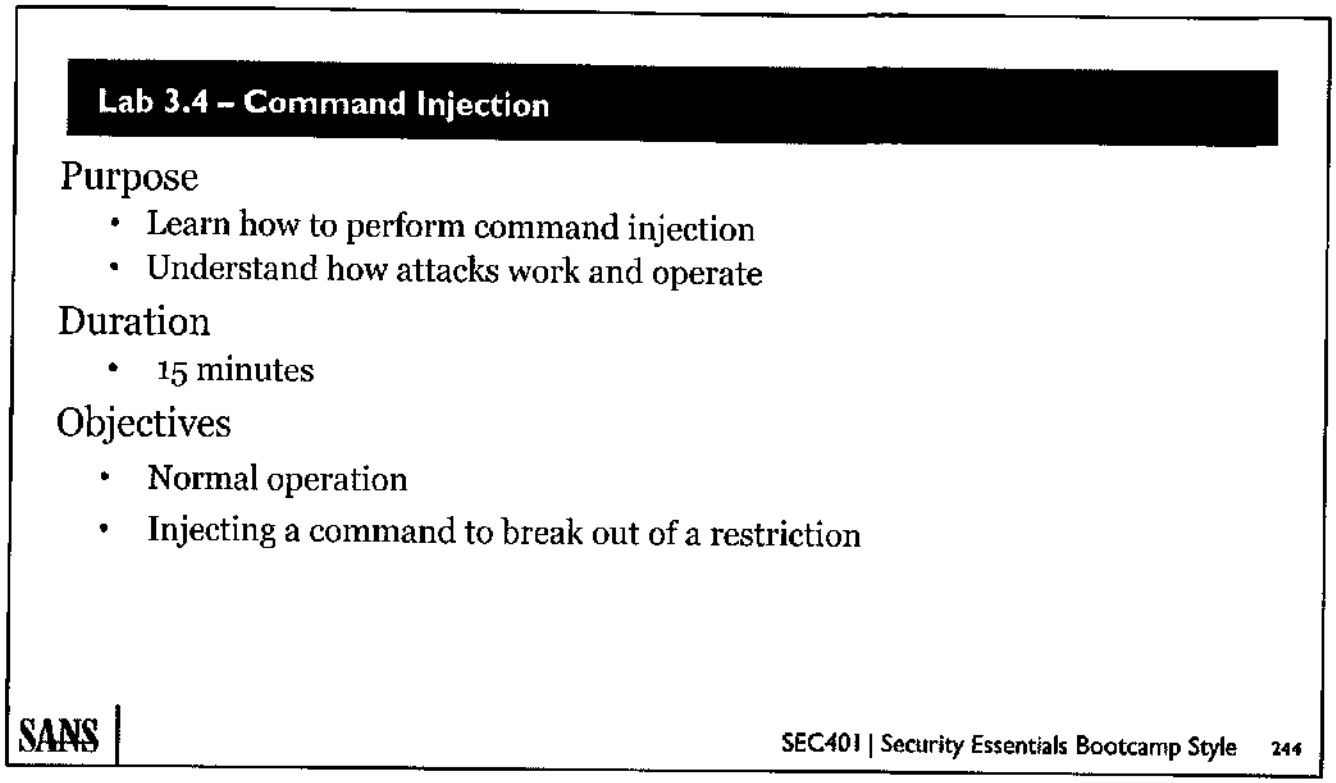
SANS

Lab 3.4 – Command Injection

You previously learned about various input attacks. One of these attacks, command injection, may allow an attacker to execute unauthorized commands on the target system if no input validation is being performed to filter potentially harmful characters. It is highly discouraged to execute code or scripts on the server side of a connection, especially if a user can potentially influence the data being processed. Each operating system, programming language, or scripting language have different characters that may allow an attacker to append additional commands to the end of a string, such as ";, &&, ;, |."

© 2017 Secure Anchor Consulting

Technet24



Purpose

- Learn how to perform command injection
- Understand how attacks work and operate

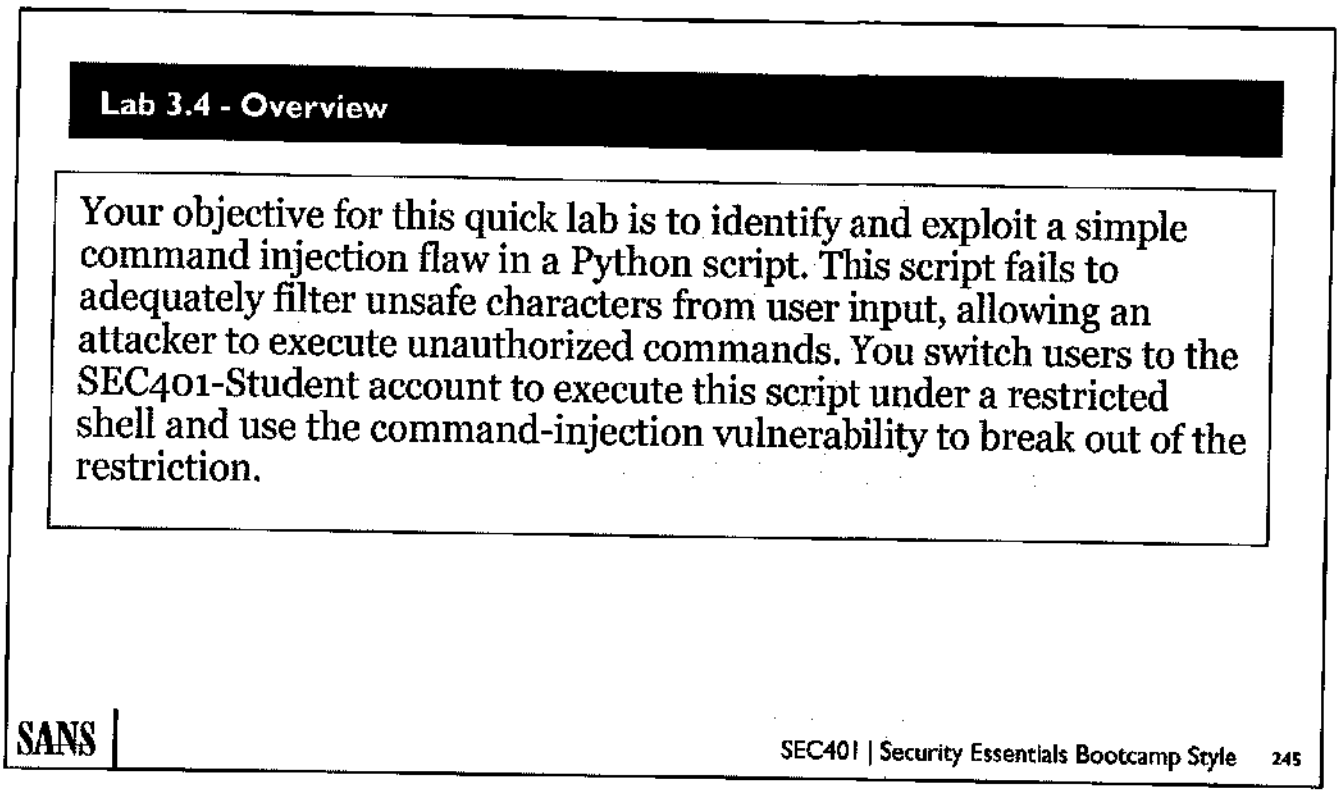
Duration

- 15 minutes
- The estimated duration of this lab is based on the average amount of time required to make it through to the end. The duration estimate of this lab can decrease or increase depending on various factors, such as the booting of virtual machines, the speed and amount of RAM on your computer, and the time you take to read through and perform each step. All labs are repeatable both inside and outside of the classroom, and it is strongly recommended that you take the time to repeat the labs both for further learning and practice toward the GIAC Security Essentials Certification (GSEC).

Objectives

- Normal operation
- Injecting a command to break out of a restriction

© 2017 Secure Anchor Consulting



Your objective for this quick lab is to identify and exploit a simple command injection flaw in a Python script. This script fails to adequately filter unsafe characters from user input, allowing an attacker to execute unauthorized commands. You switch users to the SEC401-Student account to execute this script under a restricted shell and use the command-injection vulnerability to break out of the restriction.

© 2017 Secure Anchor Consulting

Technet24

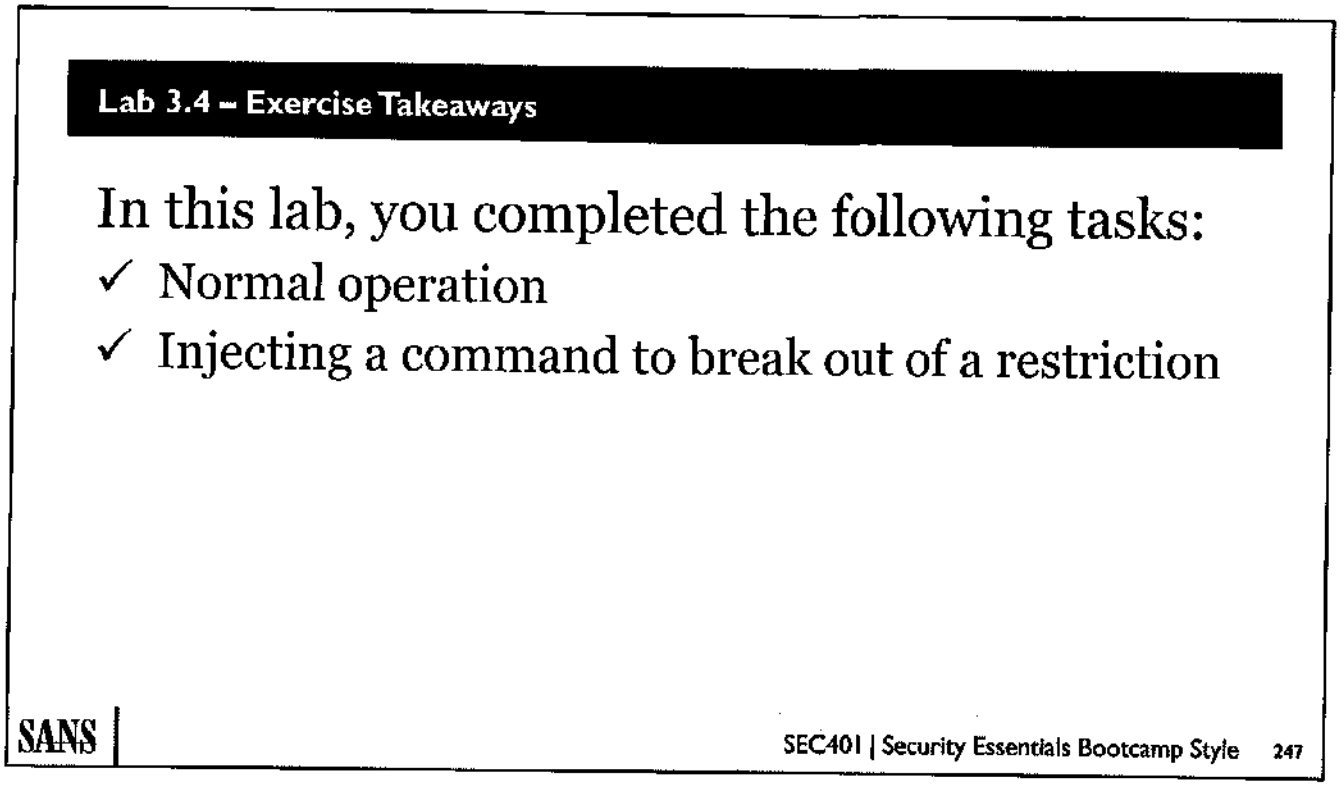
SANS

NOTE: Please open the separate Lab Workbook and turn to Lab 3.4

The instructor is going to introduce and go over the labs. Once the instructor is done, you will be instructed to work on the lab. If you have any questions, you can ask the instructor.

This page intentionally left blank.

© 2017 Secure Anchor Consulting



In this lab, you completed the following tasks:

- ✓ Normal operation
- ✓ Injecting a command to break out of a restriction

In this lab, you exploited a command-injection bug residing in a Python script. The script fails to properly filter out harmful characters that are meaningful to the system() function. You then used this vulnerability to escape from a restricted shell. Command-injection bugs are easy to fix once identified; however, they are also easy to miss, because the program still compiles. Proper source code review can help to find these bugs before they go into production.

© 2017 Secure Anchor Consulting

Technet24

SANS

Lab 3.4 is now complete

This page intentionally left blank.

© 2017 Secure Anchor Consulting

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

SANS Programs
sans.org/programs

GIAC Certifications
Graduate Degree Programs
NetWars & CyberCity Ranges
Cyber Guardian
Security Awareness Training
CyberTalent Management
Group/Enterprise Purchase Arrangements
DoDD 8140
Community of Interest for NetSec
Cybersecurity Innovation Awards

Search SANSInstitute

SANS Free Resources
sans.org/security-resources

- E-Newsletters
  NewsBites: Bi-weekly digest of top news
  OUCH!: Monthly security awareness newsletter
  @RISK: Weekly summary of threats & mitigations
- Internet Storm Center
- CIS Critical Security Controls
- Blogs
- Security Posters
- Webcasts
- InfoSec Reading Room
- Top 25 Software Errors
- Security Policies
- Intrusion Detection FAQ
- Tip of the Day
- 20 Coolest Careers
- Security Glossary

SANS Institute
8120 Woodmont Avenue | Suite 310
Bethesda, MD 20814
301.654.SANS(7267)
info@sans.org

Technet24