11.1 Student Guide: Introduction to Firewalls and Network Security

Overview

This week introduces you to network security through the principles of network defense. By the end of the week, you will have the foundations upon which to build and execute a network defense strategy.

Day 1 Introduces you to firewalls and defense in depth. Day 2 Introduces you to IDS systems and Network Security Monitoring. Day 3 starts by introducing you to the Cyber Kill chain then culminates with you exercising their threat hunting skills through the use of Enterprise Security Monitoring.

Class Objectives

By the end of the lesson, you will be able to:

Explain how open ports contribute to a computer's attack surface.

Use firewalls to protect a computer's open ports.

Decribe various types of firewalls and their use cases.

Explain the role firewalls play within a layered defense.

Gain hands-on experience with developing and implementing firewall policies using UFW and firewalld.

Slideshow

The lesson slides are available on Google Drive here: 11.1 Slides

01. Welcome and Motivation

Last week, we learned how cryptographic tools are used to maintain the confidentiality, integrity, and availability of data while at rest and in transit.

For example, malicious actors could attempt to hack into a hospital's web server in order to perform a ransomware attack.

In an effort to keep the private health information of their patients secure, the hospital protects secured access to its website using a 256bit cryptographic RSA VPN connection that provides authorized medical staff with access to patient records when responding to emergencies while away from the hospital. Emergency room physicians also use secured VPNs to host virtual appointments with patients who are unable to visit the hospital.

While these tools are important, cryptography is only a part of a multifaceted network defense ecosystem used to protect private information and critical network infrastructure.

While cryptographic applications help protect private data and critical network infrastructure from specific network attacks, other aspects of website security remain at risk.

For example, suppose the same website is the target of a denial of service (DoS) attack. Cryptography defenses aren't of much use.

Security practitioners need to implement network security protections like firewalls that provide an additional layer of defense.

Firewalls are a technical security control that distinguish between trusted and untrusted network traffic and network connections.

Defending against a wide variety of threats requires the knowledge and use of network security.

According to SANS Institute:

"Network Security is the process of taking physical and software preventative measures to protect the underlying networking infrastructure from unauthorized access, misuse, malfunction, modification, destruction, or improper disclosure, thereby creating a secure platform for computers, users and programs to perform their permitted critical functions within a secure environment." - SANS Network Security Resources

Firewalls are the first line of defense on the perimeter of the network's edge. However, they are merely the first layer upon which others preventive and protective layers are built.

This concept of layered security protections is known as the defense-in-depth methodology, which we touched briefly on in our GRC unit.

In the GRC unit, we covered how defense-in-depth plays a critical role in securing organizations. Throughout this unit, we will apply a DiD mindset to layer security measures that aims to slow an attack's progression, thus providing network defenders with enough time to respond:

Today, we will cover the benefits, limitations, for various types of firewalls. We will also explore defense-in-depth methodologies.

On Day 2, we will introduce intrusion detection systems (IDS) and network security monitoring (NSM). NSM is particularly useful for tracking an adversary through the network after a breach.

On Day 3, we will conduct advanced cyber threat hunting using Enterprise Security Management or (ESM). ESM expands upon the concept of NSM through the inclusion of endpoint telemetry. We'll discuss all these concepts in depth on day 3.

Today's lesson will explore:

How open ports contribute to a machine's attack surface and how firewalls are used to protect a computer's open ports.

Usage of different types of firewalls and their application.

The role firewalls play within a layered defense.

The development and implementation of firewall policies using UFW and firewalld.

Professional Application

Firewalls are the tools used to implement network access control. It is expected that professionals within the following roles are able to implement and execute firewall rules:

Help Desk/IT Specialist: Entry-level IT roles focus more on troubleshooting user issues than implementing network controls.

Being able to determine if and how firewalls affect user traffic can help troubleshoot issues like slow connections, an unexpected lack of a connection, and broken networked applications, such as Skype or Facebook Messenger.

System/Network Administrator: System and network administrators often determine who is allowed to access which devices on a network. These roles must be able to develop firewall policies and implement them using tools like UFW or firewalld.

SOC Analyst: perform threat hunting and alert triage.

They must be able to understand how firewalls on the network filter incoming and outgoing packets. They also need to understand intrusion detection systems in order to accurately interpret the traffic logs they monitor for incidents. Without this knowledge, it will be harder for them to identify abnormal network traffic.

It also enables them to direct IT personnel to modify their firewall policies in order to mitigate future attacks.

Penetration Tester: Penetration testers don't usually implement firewall rules themselves. However, launching a successful attack against a network requires them to determine whether a firewall sits between them and their target and, if so, identify which rules are enabled based on the results of their probe.

This allows them to identify ways to bypass the firewall and continue their test.

Emphasize that familiarity with today's content is essential knowledge for performing any of these roles.

Take a moment to address questions before proceeding.

02. Firewall Architectures

In Unit 8.2, we discussed how ports and services were the essential doorways into a network. Networks allow computers to communicate with one another by sending data to and from open ports on other machines.

Therefore, devices on networks must expose open ports in order to communicate with other machines on the network.

If a machine exposes an open port to a network. Anyone with the capability to discover a machine on a network can attempt to connect to it through that port. This is not a problem if connections are only made from trusted parties.

However, in practice, it is unwise to assume the only people who will connect to a device are those you trust. Malicious actors can and will exploit this assumption in order to access sensitive information on the network.

Restricting access to open ports is a fundamental skill for any technical security specialist. The most basic part of protecting a network is controlling who can access which services on which machines.

Let's see how we can secure network ports to allow only trusted traffic in.

Firewalls provide a layer of protection by analyzing data leaving and entering a network.

Firewalls are placed between application servers and routers to provide access control.

They protect trusted networks by isolating them from untrusted networks, such as the internet.

There are two types of firewalls: host-based firewalls and network firewalls.

Host-Based and Network Firewalls

Firewalls can be used to control access to either a single host (host-based firewall) or an entire network (network firewall).

A host-based firewall runs on the machine it is meant to protect and blocks traffic to and from that specific device.

A network firewall is often placed in front of a router in order to block malicious internet traffic from entering a private network.

These two firewalls work similarly by doing the following:

Intercepting traffic before it reaches its target host or router.

Inspecting the source/destination address/ports, TCP flags, and other features of the incoming packets.

Allowing packets that come from trusted sources and denying packets that don't.

Firewall Types and Architectures

A firewall is a multifunctional network security appliance that operates on multiple layers of the OSI. These firewalls can be broken down into four basic types:

MAC layer

Packet filtering

Circuit-level gateways

Application gateways

MAC Layer Filtering Firewall

Each device on a network has a unique hardware identification called a Media Access Control (MAC) address. This unique hardware ID is what allows all devices to communicate.

As its name suggests, this firewall operates at Layer 2 of the OSI model and filters based on source and destination MAC addresses.

MAC layer firewalls, also referred to as MAC filtering, are commonly found on Wi-Fi systems.

This firewall will first compare a device's MAC address against a statically approved list of MAC addresses. If there is a match, all traffic will be forwarded.

One advantage of MAC layer firewalls is that they may help secure a network from novice attackers.

One disadvantage is that this security control can easily be bypassed by MAC spoofing, the process of changing a MAC address to match that of another, typically one that is already allowed in the firewall.

Stateless Firewall

Stateless firewalls, also known as packet-filtering firewalls, operate at Layer 3 and 4 of the OSI model. These firewalls statically evaluate the contents of packets and do not keep track of the state of a network connection (aka Stateless).

Packet-filtering firewalls examine the network and transport headers closely for the following information:

Source and destination IP address: Source IP address is from a device that initiates a communication. The destination IP address is the intended receiving device.

Source and destination port information: When combined with an IP address, ports enable applications that are running on TCP/IP network hosts to communicate.

IP (Internet Protocol): Responsible for addressing devices on a network and ensuring the delivery of data from a source to a destination.

This delivery can be through connection-oriented TCP, meaning a successful three-way handshake must occur before the transmission of data.

This delivery can also be through connectionless UDP (User Datagram Protocol), which doesn't rely on a three-way handshake.

Ingress/egress interface: Interface of data being received (ingress) and data being sent out (egress).

Packet-filtering firewalls are the oldest type of firewall architecture.

It creates checkpoints within a router or switch and examines packet data as it is transported through a network interface.

If the information contained within the packet does not pass this inspection process, it is dropped.

This kind of filtering works by inspecting the contents of each individual packet. Stateless firewalls do not consider the state of the connection as a whole.

One advantage is that stateless firewalls are not resource intensive, meaning they are low-cost and do not significantly impact system performance.

One disadvantage is that they are easy to subvert compared to more robust firewalls. They are vulnerable to IP spoofing and do not support custom-based rule sets.

Stateful Firewall

Stateful firewalls operate at Layer 3 and 4 of the OSI model.

It uses a combination of TCP handshake verification and packet inspection technology to create stronger protection than either of the two previously discussed firewalls.

Rather than looking at individual packets, stateful firewalls examine the connection as a whole, looking at whole streams of packets.

This allows stateful firewalls to detect more information than stateless firewalls can. For example, they can determine:

If a packet is trying to establish a new connection. This is called a NEW state.

If a packet is part of an existing connection. This is called an ESTABLISHED state.

If a packet is not opening a new connection or belongs to an existing one. This is considered a Rogue packet.

Since stateful firewalls understand the context of the entire data stream, they can determine which application layer protocols are in use. However, they cannot actually understand application layer protocols, so they can't determine what the underlying traffic is doing.

For example: Stateful firewalls can identify that a connection is using HTTP, but cannot identify if the connection is being used to request an HTML file or a PNG image.

One advantage of stateful firewalls is that they offer transparent mode, which allows direct connections between clients and servers.

They can also implement protocol specific algorithms and complex models to ensure more secure connections.

One disadvantage is that they are resource-intensive systems that can slow the transmission of legitimate packets when compared to other solutions. If exploited, this can cause a DoS condition.

Circuit-Level Gateway Firewall

Circuit-level firewalls operate at Layer 5 of the OSI model.

Circuit-level gateways determine the legitimacy of TCP connections by observing the handshake process between packets.

Once the circuit is allowed to establish an end-to-end connection, all data is tunneled between the parties.

Circuit-level gateways work by verifying the three-way TCP handshake. TCP handshake checks are designed to ensure that session packets are from legitimate sources.

Circuit-level gateways use the following information to determine the legitimacy of network connections before forwarding traffic:

Unique session identifier

State of the connection (Handshake established, closed)

Sequencing information

One advantage of circuit-level gateways is that they quickly and easily approve or deny traffic without consuming a significant amount of computing resources. They are also relatively inexpensive and provide anonymity to the private network.

One disadvantage is that they do not check the contents of the packet itself.

If a packet contains malware but has the correct TCP handshake information, the data is allowed to pass through.

This is why using circuit-level gateways are not capable of fully protecting a network on their own.

Application or Proxy Firewalls

Application or proxy firewalls, often referred to as "application-level gateways" , operate at Layers 3 through 7 of the OSI model.

This firewall actually inspects the contents of the packet, including authentication and encryption components.

Proxy firewalls use deep packet inspection and stateful inspection to determine if incoming traffic is safe or harmful.

Proxy firewalls intercept all traffic on its way to its final destination, without the data source knowing. A connection is established to the proxy firewall, which inspects the traffic and forwards it if it's determined to be safe, or drops it if it's determined to be malicious.

Proxy firewalls create an extra layer of protection between the traffic source and its destination behind the network by obscuring the destination from the source creating an additional layer of anonymity and protection for the network.

One advantage is that it's more secure than other implementations and provides simple log and file audit management for incoming traffic.

One disadvantage is that it is resource intensive, requiring robust modern hardware and higher costs.

Summary

Firewall are modes of protection provided to an organization or user, that analyze data leaving or entering a network.

Packet filter firewalls operate between Layer 3 and Layer 4 of the OSI model.

These protect a network by examining source and destination IP address, port number, and packet type, without opening the packet to inspect its contents.

Circuit-level firewalls operate at Layer 5 of the OSI model.

These look at the header of a packet only. Once the circuit is allowed to establish an end to end connection, all data is tunneled between the parties.

Stateful firewalls operate at Layer 7 of the OSI model.

These use a combination of TCP handshake verification and packet inspection technology to create a greater level of protection.

Application or proxy firewalls operate at layer 3 through layer 7 of the OSI model.

These inspect the contents of the packet, which can involve authentication and encryption.

Firewalls are not immune to attacks.

We'll now demonstrate how to configure rules with an uncomplicated firewall or UFW in the next lecture.

03. UFW Overview and Demonstration

In this section, we'll introduce the standard Linux firewall: Uncomplicated Firewall (UFW).

UFW is a multifunction firewall that provides both stateless and stateful packet-filtering. It works on all kinds of network address and port translation, for example NAT (Network Address Translation) and NAP (Network Address Protection).

UFW has the following features:

Host-based: UFW is most commonly used on hosts.

Logging: UFW can generate multilevel logs based on specified rate limits. Rate limits tell the firewall how much bandwidth to allocate for logging functions. Logs are useful for providing insight into attacks.

Remote management: Firewalls can be remotely administered, for example through SSH via port 22. While this convenience can add risk, since port 22 is open, risk can be mitigated with a strong user password.

Rule sets for allow/deny: UFW operates between Layer 3 and 4 of the OSI model. It protects a network by examining source and destination IP address, port number, and packet type, without opening the packet to inspect its contents.

Additional security: UFW also operates at layer 7 of the OSI mode and uses a combination of TCP handshake verification and packet inspection technology to create stronger protection.

Rate-limiting: UFW supports rate-limited connections to protect against brute force attacks.

For example: UFW will deny a connection that attempts to initiate five or more connections within a time span of one minute.

UFW provides an easy-to-use interface for those who are unfamiliar with more complex firewall concepts.

UFW Demo Setup

Log into Azure then launch the Ubuntu UFW VM.

This demonstration will show how to use UFW in the following scenario:

The IT department is hosting a website that requires the use of both normal and encrypted web traffic.

Your CISO has released a security advisory authorizing the use of secured remote firewall administration.

Because of this, we need to open ports 22, 80, and 443.

The tasks we need to complete are:

Use sudo ufw reset to reset all UFW rules to factory defaults.

Use sudo ufw status to check the current status of the firewall.

Use sudo ufw enable to start the firewall and update rules.

Use sudo ufw reload to reload the UFW firewall.

Use sudo ufw default deny incoming to block all incoming connections.

Use sudo ufw default allow outgoing to allow all outgoing connections.

Use sudo ufw allow to open specific ports.

Use sudo ufw deny to close specific ports.

Use sudo ufw delete to delete rules.

Use sudo ufw disable to shut down the firewall.

Enabling UFW

Let's remove all settings on UFW by resetting all rules to their factory defaults. This will allow us to customize UFW with our own rule sets.

Type the following command:

sudo ufw reset

The reset command option changes configuration back to the defaults.

Output should look similar to below:

Resetting all rules to installed defaults. Proceed with operation (y|n)? y

Backing up 'user.rules' to '/etc/ufw/user.rules.20191210\_144807'

Backing up 'before.rules' to '/etc/ufw/before.rules.20191210\_144807'

Backing up 'after.rules' to '/etc/ufw/after.rules.20191210\_144807'

Backing up 'user6.rules' to '/etc/ufw/user6.rules.20191210\_144807'

Backing up 'before6.rules' to '/etc/ufw/before6.rules.20191210\_144807'

Backing up 'after6.rules' to '/etc/ufw/after6.rules.20191210\_144807'

UFW isn't started by default, so administrators have to enable it themselves.

Type the following command:

sudo ufw status

The output should look similar to below:

Status: inactive

UFW commands require sudo rights, because manipulating the firewall affects all users on the system.

Next, we need to enable UFW to start.

Type the following command:

sudo ufw enable

Output should look similar to below:

Firewall is active and enabled on system startup

This tells us that the firewall has successfully started.

Next we need to check the status of the firewall again, now that it's enabled.

Type the following command:

sudo ufw status

Output should look similar to below:

Status: active

Setting Default Rules

It's best practice to deny all incoming and outgoing traffic by default. This way, all we need to do is allow required ports.

Type the following commands:

sudo ufw default deny incoming

sudo ufw default deny outgoing

default: Applies settings to all traffic.

deny: Blocks traffic.

incoming: Refers to traffic coming in (inbound) to the network interface.

outgoing: Refers to traffic going out (outbound) from the network interface.

Output will look similar to below:

Default incoming policy changed to 'deny'

(be sure to update your rules accordingly)

Allow and Deny Rules

Denying traffic by default breaks networked applications, since it prevents them from sending or receiving data. Next, we'll need to explicitly allow traffic to and from ports 80, 443, and 22.

Type the following commands:

sudo ufw allow 80

sudo ufw allow 443

sudo ufw allow 22

sudo ufw allow 110

Output should look similar to below:

Rule added

Rule added (v6)

allow: Opens the port to allow inbound and outbound traffic.

80, 443, 22, 110: The ports to be opened.

Now, let's check the status of the firewall again.

Type the following command:

sudo ufw status

This time, the output will appear as follows:

Status: active

To Action From

-- ------ ----

80 ALLOW Anywhere

443 ALLOW Anywhere

22 ALLOW Anywhere

110 ALLOW Anywhere

80 (v6) ALLOW Anywhere (v6)

443 (v6) ALLOW Anywhere (v6)

22 (v6) ALLOW Anywhere (v6)

110 (v6) ALLOW Anywhere (v6)

Now we can see all the ports that we had just allowed.

Note: IPv6 traffic is identified with (v6). All others are IPv4 traffic.

Port 110 was not one of the ports that we decided to use in our scenario. Let's deny port 110.

Type the following command:

sudo ufw deny 110

deny: Closes the port indicated in the command.

110: The port to be closed.

Now we need to check the status of the firewall again to make sure our changes took effect.

Type the following command:

sudo ufw status

Output should now look like:

Status: active

To Action From

-- ------ ----

80 ALLOW Anywhere

443 ALLOW Anywhere

22 ALLOW Anywhere

110 DENY Anywhere

80 (v6) ALLOW Anywhere (v6)

443 (v6) ALLOW Anywhere (v6)

22 (v6) ALLOW Anywhere (v6)

110 (v6) DENY Anywhere (v6)

We can see that port 110 is now blocked.

if we want to verify if these rules to take effect, we need to run: sudo ufw status

Deleting Rules

You can delete firewall rules when they are no longer needed.

Type the following command:

sudo ufw delete deny 110

Output should look similar to below:

Rule deleted

Rule deleted (v6)

delete: Removes rule.

deny 110: The rule to be removed.

We need to check the firewall status again to ensure our changes have taken effect.

Type the following command:

sudo ufw status

Output should now look like:

Status: active

To Action From

-- ------ ----

80 ALLOW Anywhere

443 ALLOW Anywhere

22 ALLOW Anywhere

80 (v6) ALLOW Anywhere (v6)

443 (v6) ALLOW Anywhere (v6)

22 (v6) ALLOW Anywhere (v6)

The deny rule for port 110 is gone.

There's another command we can use that displays more information about the status of the UFW firewall.

Type the following command:

sudo ufw status verbose

verbose: Displays extra information.

Output should look similar to:

Status: active

Logging: on (low)

Default: deny (incoming), deny (outgoing), deny (routed)

New profiles: skip

To Action From

-- ------ ----

80 ALLOW IN Anywhere

443 ALLOW IN Anywhere

22 ALLOW IN Anywhere

80 (v6) ALLOW IN Anywhere (v6)

443 (v6) ALLOW IN Anywhere (v6)

22 (v6) ALLOW IN Anywhere (v6)

In the example above we can see extra information such as:

The logging level is set to low.

Default deny is active on the following interfaces: incoming, outgoing, and routed.

New profile creation has been skipped.

To verify if the new changes are active we'll need to run sudo ufw status.

Type the following commands:

sudo ufw status

When troubleshooting firewall rules, sometimes it's helpful to completely stop it and restart it:

Firewall stopped and disabled on system startup

disable shuts down or turns off the firewall.

sudo ufw disable

Output should look similar to below:

Firewall is active and enabled on system startup

Run: sudo ufw enable

Output should look similar to below:

Firewall is active and enabled on system startup

reload loads all the current rules without stopping the firewall.

disable shuts down or turns off the firewall.

enable starts and loads all the current rules.

UFW Demo Summary

What command would I use to check on my firewall?

sudo ufw status

What command would I use to enable my firewall?

sudo ufw enable

What is the default port for SSH?

port 22

What is the default port for HTTP?

port 80

What is the default port for HTTPS?

port 443

Knowing these concepts will help answer questions on the Security+ exam.

04. Configuring UFW Activity

Activity File: Configuring UFW

05. Review Configuring UFW Activity

Solution File: Configuring UFW

06. firewalld Introduction

Every device that connects to the public network must have a unique hardware address (a MAC address) before it can send and receive data. Depending on the size of the network, there might be hundreds of devices, each requiring its own set of rules.

Firewalld is very similar to UFW, but it's important to understand how both firewalls work in order to be prepared for your future job enviroment. Firewalld is a bit more complicated than UFW, but provides greater flexibility and does not disrupt services when managing firewall updates.

firewalld Overview

firewalld is a dynamically managed firewall. It allows the user to set up and configure multiple firewall type options, which allows the user to block or allow incoming or outgoing traffic.

It is a bit more complicated than UFW, but provides greater flexibility and does not disrupt services when managing firewall updates.

firewalld uses the concept of zones to divide network interfaces into groups of that share trust levels. The zones are assigned sets of rules depending on the needs and restrictions of each zone's interface.

Zones are the organization of rules. Each zone can contain several rules.

Through this division of zones, firewalld can manage rule sets dynamically without breaking existing sessions, disrupting services and bringing down the entire network.

firewalld supports network and firewall zones that define trust levels of networked connections or interfaces.

Rules and configurations can be tested and evaluated in runtime environments.

Runtime configurations are only valid until the next system reboot or service reload. This means we can create settings that are active for a limited amount of time. Runtime configurations can also be used to test new configurations. They can then be seamlessly saved to permanent environment if they work well.

Permanent configurations are loaded with each reboot. These become the active runtime environment until new runtime configurations are made.

Different interfaces may requires different firewall rules due to the various services that may be in use.

In addition to zones, firewalld uses services as shortcuts to configuring firewall rules for common services.

firewalld enables you to designate which services to allow, and automatically opens the ports associated to those services.

For example, if you enable the SSH service in a zone, firewalld opens port 22 without requiring you to specify the port number explicitly.

Services can be predefined, making it easy to configure the firewall rules most commonly required by most servers. Services can also be custom defined.

firewalld Demo Setup

We'll demonstrate how to use firewalld with the following scenario:

An IT administrator is bringing a new Microsoft Active Directory server online. It will serve several new hosts on the third floor at the main office, which is serviced by the eth1 on the firewall.

The administrator requested that this new network not be able to transmit or receive data from the Fifth Street office location, which uses an IP address of 10.10.0.10.

Lastly, the administrator asked you to block all ICMP pings on that same interface as an extra level of protection.

Note the following:

eth0 = first physical interface on a device

eth1 = second physical interface on a device

wlan0 = first WiFi interface

wlan1 = second WiFi interface, etc.

The steps we need to take are:

Use sudo /etc/init.d/firewalld start to start the firewall.

Use sudo firewall-cmd to create, modify, and delete rules.

Use --list-all-zones to list the currently configured zones.

Use --zone=work --change-interface=eth1 to bind zones to physical interfaces.

Use --list-all to list all active rules in a zone.

Use --get-services to list the currently configured services.

Use --add-rich-rule= to configure rules with more detailed options.

Use --add-icmp-block= to block ping requests.

firewalld Demo

Starting firewalld

Since firewalld is dynamic, we can start it and make changes without having to restart or reload it. We start firewalld using the following command:

sudo /etc/init.d/firewalld start

/etc/init.d/firewalld: File path to the firewalld program.

start: Starts the program.

Output should look similar to below:

[ ok ] Starting firewalld (via systemctl): firewalld.service.

Firewalld service is located in the etc/init.d/firewalld directory.

firewalld Zone Views

With firewalld, we can set a zone to an interface and configure each individual zone setting. We can use firewall-cmd to list zones and services, associate zones with interfaces, and configure new firewall rules.

Run the following command

sudo firewall-cmd --list-all-zones

firewall-cmd: Option that tells firewalld to create, modify, and delete rules.

--list-all-zones: Command option that lists all currently configured zones.

Your output should reflect the following default zones that come preconfigured:

block, dmz, drop, external, home, internal, public, trusted, and work.

Binding Zones to Physical Interfaces

Since we know that the zone will be work-related and the physical interface is located on the third floor is eth1, we need to bind the work zone to the eth1 interface.

Type the following command:

sudo firewall-cmd --zone=work --change-interface=eth1

--zone=work: Displays information regarding the work zone.

--change-interface=: Command option used to bind an interface to a different zone.

eth1: The interface used to change to.

Output should read:

success

The process of binding eth1 to the work zone causes the eth1 interface to inherit all rules from the work zone.

Zone and Service Verification

Let's verify the binding of the work zone to the physical interface of eth1.

Type the following command:

sudo firewall-cmd --zone=work --list-all

--list-all: Lists all settings for a specific zone.

Output should look similar to:

work (active)

target: default

icmp-block-inversion: no

interfaces: eth1

sources:

services: ssh dhcpv6-client

ports:

protocols:

masquerade: no

forward-ports:

source-ports:

icmp-blocks:

rich rules:

We can see which services are allowed. If the service isn't listed, it will be blocked.

Next we will list all currently running services inside firewalld.

Type the following command:

sudo firewall-cmd --get-services

--get-services: Returns a list of all currently running services.

Output should look similar to:

RH-Satellite-6 amanda-client amanda-k5-client bacula bacula-client bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc ceph ceph-mon cfengine condor-collector ctdb dhcp dhcpv6 dhcpv6-client dns docker-registry docker-swarm dropbox-lansync elasticsearch freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp ganglia-client ganglia-master git high-availability http https imap imaps ipp ipp-client ipsec irc ircs iscsi-target kadmin kerberos kibana klogin kpasswd kprop kshell ldap ldaps libvirt libvirt-tls managesieve mdns minidlna mosh mountd ms-wbt mssql murmur mysql nfs nfs3 nrpe ntp openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole pmcd pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy proxy-dhcp ptp pulseaudio puppetmaster quassel radius redis rpc-bind rsh rsyncd samba samba-client sane sip sips smtp smtp-submission smtps snmp snmptrap spideroak-lansync squid ssh synergy syslog syslog-tls telnet tftp tftp-client tinc tor-socks transmission-client vdsm vnc-server wbem-https xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-server

Iit's important to know which serivces are running on your system.

The --get-services command option provides insight into which services are running.

Based on your needs, disable the ones that are not critical to business operations. This is a form of system hardening.

Next, we will block all traffic coming from the Fifth Street location. The IP address associated with that location is 10.10.0.10.

Run the following command:

sudo firewall-cmd --zone=work --add-rich-rule='rule family="ipv4" source address="10.10.0.10" reject'

--add-rich-rule=: The option to add a new rich rule.

rule family="ipv4": Limits the rule to the IPV4 protocol.

source address="10.10.0.10": The source IP address.

reject: The option to reject IPV4 addresses from the source address.

Run: sudo firewall-cmd --zone=work --list-all

Output should look similar to:

work (active)

target: default

icmp-block-inversion: no

interfaces: eth1

sources:

services: ssh dhcpv6-client

ports:

protocols:

masquerade: no

forward-ports:

source-ports:

icmp-blocks:

rich rules:

rule family="ipv4" source address="10.10.0.10" reject

As seen in the output above, the new "rich rule" has been successfully added to our work zone and will be applied to the binded interface of eth1.

Note: The reject option in our rich rule will block all traffic from the 10.10.0.10 network.

Rules are erased when the firewall reboots. To save the configuration permanently to the database, add the --permanent option to the end of the rich rule.

We will also block ICMP pings from entering this network, in order to mitigate against DoS attacks. A ping request is also a technique that malicious actors use to acquire information about their targets.

Run the following command:

sudo firewall-cmd --zone=work --add-icmp-block=echo-reply --add-icmp-block=echo-request

--add-icmp-block: Command option used to block ICMP protocols.

Run sudo firewall-cmd --zone=work --list-all

Output should look similar to:

work (active)

target: default

icmp-block-inversion: no

interfaces: eth1

sources:

services: ssh dhcpv6-client

ports:

protocols:

masquerade: no

forward-ports:

source-ports:

icmp-blocks: echo-reply echo-request

rich rules:

rule family="ipv4" source address="10.10.0.10" reject

We can see that icmp-blocks will block echo-reply and echo-request.

07. firewalld Configuration

Activity File: firewalld Configuration

08. firewalld Configuration Activity

Solution Guide: firewalld Configuration

09. Testing Firewall Rules with NMAP

In the final section of this class, we will cover the security implications of an improperly configured firewall.

Security professionals are faced with the never-ending task of defending networks against attacks. They are also expected to know, at all times, what is running on their networks, and any vulnerabilities that exist.

While there are many monitoring utilities available for performing network scans and security audits, nothing beats the versatility and useability of Nmap, the industry standard in performing network scans.

Suppose the following scenario:

You're a security analyst, and you just discovered that logs and files are missing from your system. You also notice that timestamps on logs and files have been manipulated.

At the end of the incident response effort, it was discovered that an attacker had entered your network through port 3389, or Remote Desktop Protocol (RDP).

A network scan with Nmap would have identified that the port was open. In fact, Nmap would have identified every port that was open.

The following is a short list of what kind of information attackers can gather by performing network scans:

Name and version of operating system (OS fingerprinting).

All open and closed ports.

All filtered ports (ports behind a firewall).

Types of services running on a specific port (service and daemon names).

Firewalking is a type of reconnaissance that uses network security analysis to determine which Layer 4 protocols a specific firewall will allow.

Nmap Demo Setup

In this demo, we'll use our firewalld VM to perform scans against our UFW firewall.

We will use the following scenario:

Your security manager has installed a brand new, fully configured firewall and would like you to test it using Nmap.

You've decided to use various fingerprinting techniques to reveal the type of operating system, services, daemons, and protocols that are currently running. You will also test to see which ports are open, closed, and filtered.

We'll need to complete the following steps:

Use nmap -O -p 1-500 --osscan-guess to perform OS fingerprinting, and scan ports 1 through 500.

Use uname -a to print the OS type and version.

Use nmap -sV to enumerate service type.

Use nmap -A -T4 to perform OS fingerprinting using fast execution.

Use nmap -sO to perform an IP protocol scan.

Use nmap -sU -F to perform device and port enumeration.

Use nmap -sA to enumerate the type of firewall in use.

Basic Nmap Scans

First, we will set up our test environment. We'll establish a basic UFW configuration.

Type the following commands:

sudo ufw reset

sudo ufw enable

sudo ufw default deny incoming

sudo ufw default deny outgoing

sudo ufw allow 80

sudo ufw allow 22

sudo ufw allow 443

sudo ufw disable

We now have a fully functional firewall for our test environment. Next, we'll disable it so we can observe the differences when performing Nmap scans.

Operating System Fingerprinting

Now we'll simulate an OS scan, which will enumerate OS names and versions. With this information, attackers know which specific vulnerabilities exist then attack those.

Type the following command from inside the firewalld VM:

sudo nmap -O -p 1-500 --osscan-guess 172.17.18.72

nmap: Program to perform network scan.

-O: Enables OS detection.

-p 1-500: Scans ports 1 through 500.

172.17.18.72: IP address to perform network scan against.

The output should resemble:

Starting Nmap 7.60 ( [https://nmap.org](https://nmap.org/) ) at 2020-03-18 14:42 EDT

Nmap scan report for 172.17.18.72

Host is up (0.00071s latency).

Not shown: 492 closed ports

PORT STATE SERVICE

21/tcp open ftp

22/tcp open ssh

25/tcp open smtp

80/tcp open http

110/tcp open pop3

139/tcp open netbios-ssn

143/tcp open imap

445/tcp open microsoft-ds

MAC Address: 00:15:5D:00:04:00 (Microsoft)

Device type: general purpose

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux\_kernel:3 cpe:/o:linux:linux\_kernel:4

OS details: Linux 3.2 - 4.8

Network Distance: 1 hop

OS detection performed. Please report any incorrect results at <https://nmap.org/submit/> .

Nmap done: 1 IP address (1 host up) scanned in 4.69 seconds

In the above example, we can see that nmap returned several open ports along with the protocols they are running.

The nmap scan returned the MAC address of the machine.

It also returned the best guess of a version of OS (linux):

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux\_kernel:3 cpe:/o:linux:linux\_kernel:4

OS details: Linux 3.2 - 4.8

From within the UFW VM, type the following command:

uname -a

uname: Prints operating system information.

a: Returns all results.

Output should look similar to below:

Linux ubuntu 4.15.0-70-generic #40~18.04.1-Ubuntu SMP Thu Nov 14 12:06:39 UTC 2019 x86\_64 x86\_64 x86\_64 GNU/Linux

Our version of OS is 4.15.0-70-generic which is a bit off from the scan.

Note: Your results may be different, implying that the OS scan was not able to accurately guess the OS version. Regardless, it will properly guess that we are using a Linux distribution.

From within the UFW VM, enable the UFW firewall and run the test again.

Return to the firewalld VM and try the command one more time.

Type the following command:

sudo nmap -O -p 1-500 --osscan-guess 172.17.18.72

Output should look similar to:

Starting Nmap 7.60 ( [https://nmap.org](https://nmap.org/) ) at 2020-03-18 14:52 EDT

Nmap scan report for 172.17.18.72

Host is up (0.00079s latency).

Not shown: 497 filtered ports

PORT STATE SERVICE

22/tcp open ssh

80/tcp open http

443/tcp closed https

MAC Address: 00:15:5D:00:04:00 (Microsoft)

Aggressive OS guesses: Linux 3.10 - 4.8 (97%), Linux 3.2 - 4.8 (96%), Linux 2.6.32 - 3.13 (95%), Linux 3.16 - 4.6 (94%), Linux 2.6.22 - 2.6.36 (93%), Linux 2.6.39 (93%), Linux 3.10 (92%), Linux 2.6.32 (92%), Linux 2.6.32 - 3.10 (91%), Linux 4.4 (91%)

No exact OS matches for host (test conditions non-ideal).

Network Distance: 1 hop

OS detection performed. Please report any incorrect results at <https://nmap.org/submit/> .

Nmap done: 1 IP address (1 host up) scanned in 16.84 seconds

With ufw enabled, even though nmap performed an "Aggressive OS guesses", the scan returned a lot more possible versions of Linux.

We can see that nmap is also (96%) sure that the host's Linux version is 3.10 - 4.8. That's a bit further off from our version of OS which is 4.15.0-70-generic.

Service and Daemon Name Scans

Next we'll perform an Nmap scan to enumerate specific services and daemons that are running on open ports.

Type the following command from within the firewalld VM:

nmap -sV 172.17.18.72

-sV: Command option that enables version detection.

Output should look similar to:

Starting Nmap 7.60 ( [https://nmap.org](https://nmap.org/) ) at 2020-03-18 14:59 EDT

Nmap scan report for 172.17.18.72

Host is up (0.00097s latency).

Not shown: 997 filtered ports

PORT STATE SERVICE VERSION

22/tcp open ssh OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)

80/tcp open http Apache httpd 2.4.29 ((Ubuntu))

443/tcp closed https

MAC Address: 00:15:5D:00:04:00 (Microsoft)

Service Info: OS: Linux; CPE: cpe:/o:linux:linux\_kernel

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .

Nmap done: 1 IP address (1 host up) scanned in 23.65 seconds

The scan returned the versions of active services on all open ports.

Attackers can use this information to look for vulnerabilities in the Common Vulnerabilities and Exposure website, and develop strategies to exploit them. This is why having a comprehensive patch management process is a critical part of risk management.

Service Type and OS Detection with Fast Execution

Next we will perform a scan that attempts to fingerprint the type of operating system running on a host by enumerating information gathered through open ports.

Type the following command from within the firewalld VM

nmap -A -T4 172.17.18.72

-A: Enables OS detection, version detection, script scanning, and traceroute.

-T4: Command option that performs fast detection.

Output should look similar to:

Starting Nmap 7.60 ( [https://nmap.org](https://nmap.org/) ) at 2020-03-18 15:03 EDT

Nmap scan report for 172.17.18.72

Host is up (0.00074s latency).

Not shown: 997 filtered ports

PORT STATE SERVICE VERSION

22/tcp open ssh OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)

| ssh-hostkey:

| 2048 42:a7:96:c1:11:0a:f9:14:14:78:3e:4f:75:a4:13:11 (RSA)

| 256 fc:77:2e:b7:15:d4:d7:9d:5e:9a:9d:3b:9c:54:2e:4e (ECDSA)

|\_ 256 e0:f0:3d:4f:5c:97:bd:0c:75:6e:59:19:4a:46:59:1d (EdDSA)

80/tcp open http Apache httpd 2.4.29 ((Ubuntu))

|\_http-server-header: Apache/2.4.29 (Ubuntu)

|\_http-title: Apache2 Ubuntu Default Page: It works

443/tcp closed https

Service Info: OS: Linux; CPE: cpe:/o:linux:linux\_kernel

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .

Nmap done: 1 IP address (1 host up) scanned in 24.71 seconds

We can see that the Nmap scan performed a deeper analysis to enumerate the type of operating system installed on the victim's host.

Type the following command from inside the UFW VM:

apache2 -v

apache2: Command for the Apache program.

-v: Displays version.

Output should look similar to below

Server version: Apache/2.4.29 (Ubuntu)

Server built: 2019-09-16T12:58:48

This Nmap scan returned accurate results.

The attacker was able to perform an accurate OS fingerprint as revealed in the output:

80/tcp open http Apache httpd 2.4.29 ((Ubuntu))

The attacker was able to enumerate the OS type through port 80 of the HTTP web server, which returned the result: (Ubuntu).

It's difficult to obfuscate this information, because the Apache web server is a public-facing web server designed primarily for public access.

In this case, security professionals must rely on other security controls to ensure that their networks remain protected.

Device and Port Enumeration

Attackers use different methods to determine the true state of a port, i.e., if a port is actually open or closed. Nmap has some tools for determining the true state of a port. With the UFW firewall enabled, perform the following actions:

Type the following command from within the firewalld VM:

sudo nmap -sU -F 172.17.18.72

-sU: Performs a UDP scan.

-F Performs a fast scan, scanning fewer ports than the default number.

Output should look similar to:

Starting Nmap 7.60 ( [https://nmap.org](https://nmap.org/) ) at 2020-03-23 17:23 EDT

Nmap scan report for 172.17.18.72

Host is up (0.00063s latency).

Not shown: 98 open|filtered ports

PORT STATE SERVICE

80/udp closed http

443/udp closed https

MAC Address: 00:15:5D:08:05:00 (Microsoft)

There are 100 ports that are open,filtered. This occurs when the scanned port doesn't give a response. This is most typically an indication that the firewall dropped the probe by not responding to it.

Type the following command with the firewalld VM:

sudo nmap -sA 172.17.18.72

-sA: ACK scans probe packet that only has the ACK flag set.

Output should look similar to below

Starting Nmap 7.60 ( [https://nmap.org](https://nmap.org/) ) at 2020-03-18 15:11 EDT

Nmap scan report for 172.17.18.72

Host is up (0.00083s latency).

Not shown: 997 filtered ports

PORT STATE SERVICE

22/tcp unfiltered ssh

80/tcp unfiltered http

443/tcp unfiltered https

MAC Address: 00:15:5D:00:04:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 18.19 seconds

Now we can see that 997 ports are filtered.

This typically indicates that either a hardware firewall, host-based firewall, or router is installed and configured on the host.

Filtered ports are a major obstacle for attackers, because scans provide little information about them.

We can also see that ports 22, 80, and 443 are unfiltered.

This means that the scan determined that the ports are in fact accessible, but is unable to determine wether the ports are open or closed. This indicates that a stateful firewall is being used.

The ACK scan is used to determine if a firewall is stateful, meaning it will only accept connections to previously established connections.

With this information, an attacker will know that they need to perform source routing in order to bypass the firewall's security controls.

Source routing allows attackers to craft packets that specify the route the packet must take through a network.

The goal of this exercise was to test how our firewall interacts with various scan types. Although we as security professionals cannot eliminate threats, we can mitigate risks by hardening our systems with properly configured firewalls.

10. Testing Firewall Rules with Nmap

Activity File: Testing Firewall Rules with Nmap

11. Testing Firewall Rules with Nmap Activity

Solution Guide: Testing Firewall Rules with Nmap

Class Summary

Today we talked about the relationship between ports and services.

We discussed the difference between stateful and stateless firewalls.

Stateful firewalls monitor full data streams and are able to determine whether packets belong to an existing connection, are attempting to open a new connection, or are rogue.

Stateless firewalls allow or reject traffic on a "per-packet" basis.

We also used and experimented with UFW and firewalld, two different firewalls that are essential parts of an IT professional's skill set.

We used UFW in order to create new rules for ports or services with simple syntax.

UFW requires us to restart the firewall every time we want to add or delete a rule change.

We used firewalld to create zones for different network interfaces and assign each zone their own firewall rules.

firewalld will immediately start using updated rules, without requiring a restart, and therefore limits protection downtime.

© 2020 Trilogy Education Services, a 2U Inc Brand. All Rights Reserved.

11.2 Student Guide: Introduction to Intrusion Detection, Snort, and Network Security Monitoring

Overview

Last class, you learned how firewalls play a critical role in establishing a perimerter defense using access controls by allowing or denying traffic from trusted and untrusted networks. You also learned the limitations of firewalls as in the case of the stolen laptop scenario when confidentiality has been compromised.

Today, you will build upon these concepts by utilizing an alternative defense in depth methodology that involves instrusion detection systems such as, Snort IDS systems. You will learn how to use Network Security Monitoring (NSM) and the Snort IDS engine to analyze indicators of attack (IOA), indicators of compromise (IOC), perform network forensics, and acquire intelligence and situational awareness of their networks providing them with the necessary information required to defend against network attacks.

Class Objectives

By the end of today's class, you will be able to:

Interpret and define Snort rules and alerts.

Explain how intrusion detection systems work and how they differ from firewalls.

Use Security Onion and its suite of network security monitoring tools to trace the path of network attacks.

Collect and analyze indicators of attack and indicators of compromise using NSM tools.

Apply knowledge of NSM, Snort rules, and Security Onion to establish situational awareness within a network.

Slideshow

The lesson slides are available on Google Drive here: 11.2 Slides

01. Security Onion Setup

Before we get started, we will all log into Azure and launch an instance of Security Onion. This will generate alert data that we'll use to complete the labs.

It's an important skill for cybersecurity professionals to have the ability to replay PCAP files. Remember, PCAPs are simply snapshots of live traffic frozen in time.

You have already replayed PCAPS using wireshark. This is accomplished by simply loading a PCAP into Wireshark, in other words, when you import a PCAP or click on a PCAP files on your desktop and Wireshark launches, you are essentially replaying a PCAP. The only difference is that in this case, it's a single file.

Security Onion uses the command sudo so-replay to replay multiple PCAPS stored in the /opt/samples directory, which stores hundreds of PCAPs. It's essential, that network defenders have the capability to replay network traffic using PCAPS in order to analyze and triage alert data as we'll explore in today's lesson.

Note that it's an important skill for cybersecurity professionals to have the ability to replay PCAP files. Remember, PCAPs are simply snapshots of live traffic frozen in time.

You have already replayed PCAPS using wireshark. This is accomplished by simply loading a PCAP into Wireshark, in other words, when you import a PCAP or click on a PCAP files on your desktop and Wireshark launches, you are essentially replaying a PCAP. The only difference is that in this case, it's a single file.

Security Onion uses the command sudo so-replay to replay multiple PCAPS stored in the /opt/samples directory, which stores hundreds of PCAPs. It's essential, that network defenders have the capability to replay network traffic using PCAPS in order to analyze and triage alert data as we'll explore in today's lesson.

Activity File: Security Onion Setup

02. Welcome and Overview

Network Security Recap

In last class, you learned how firewalls play a critical role in establishing a perimeter defense through the use of access controls that either allow or deny traffic from trusted and untrusted networks. Students also learned the limitations of firewalls as in the case of the stolen laptop scenario when confidentiality has been compromised.

Today, you will build upon these concepts by utilizing an alternative defense in depth methodology that involves instrusion detection systems such as, Snort IDS systems.

You will learn how to use Network Security Monitoring (NSM) and the Snort IDS engine to analyze indicators of attack (IOA), indicators of compromise (IOC), perform network forensics, and acquire intelligence and situational awareness of their networks providing them with the necessary information required to defend against network attacks.

Let's begin by reviewing the basic concepts:

Firewalls protect networks by making decisions based on rules that are set by administrators. Firewalls are designed to allow traffic from trusted sources and block traffic from untrusted sources.

Firewalls do have their limitations. Advanced hackers can easily fool them through packet manipulation.

For instance, an attacker can send malicious data through a firewall by hijacking or impersonating a trusted machine. This is why it's crucial to have a strong defense in depth methodology to help protect sensitive data.

Firewalls are only one layer of defense.

Note the following:

The first half of the day we'll begin with the introduction to Intrusion Detection & prevention systems and Snort IDS, how to physically interconnect IDS systems, and how to read, write, and interpret Snort rules.

The second half of the day will introduce you to Security Onion and the role NSM tools play within the realm of network security.

The day will culminate by exploring the concepts of threat hunting and the role it plays in Network Security Monitoring through a hands-on lab.

Intrusion detection systems (IDS) are tools that can both analyze traffic and look for malicious signatures. An IDS is similar to a firewall but has addtional capabilities such as, reads the data in the packets it inspects, issues alerts/alarms, and blocks malicious traffic if configured to do so.

There are many different types of intrusion detection systems but today's lesson will focus on Snort, the world's most popular open-source solution.

Snort adds an additional layer of defense designed to protect networks.

Network security monitoring (NSM) is the process of identifying weaknesses in a network's defense. It also provides organizations with situational awareness of their network.

Security Onion is a specific Linux distribution that's derived from Ubuntu. Security Onion uses the Snort IDS engine solely as its event-driven mechanism.

Cyber Threat Hunting is a cyber defense methodology and important skill of a cybersecurity analyst used to provide network security through network defense processes.

It's a process that proactivly and interatively searches through networks to identify and contain threats designed to evade existing security controls.

Provides the securyt analyst with a high level overview of the tactics, techniques, and procedures or TTPs used by adversaries.

An iterative process which must be continuously performed in a loop that all starts with a hypothesis, typicaly an IOA or IOC.

Explain that Cyber Threat Hunting follows three basic methodologies:

Analytics Driven The process of hsing indicators to derive a hypothesis upon which to investigate.

Intelligence Driven Uses threat intellegence reports, network forensics, and malware anlysis to establish adversarial TTPs.

Situational Awareness Driven Uses risk assesments to remediate and mitigate threats.

03. Intro to Intrusion Detection Systems and Snort

Today, we'll explore the world of intrusion detection systems and how they generally differ from firewalls.

Recall that a firewall is a device used in network security designed to filter ingress and egress traffic (inbound and outbound traffic respectively), based upon a set of predetermined administratively defined rules.

Firewalls make decisions to either allow or block traffic based on the following:

Source and destination IP address

Source and destination oort number

Protocol type

Firewalls do have their limitations. They can easily be fooled through packet manipulation by clever hackers.

For instance, attackers can send malicious data through a firewall by hijacking or impersonating a trusted machine.

Unlike firewalls, intrusion prevention and detection systems monitor, detect, and alert about an attack depending upon the configuration.

Intrusion detection systems (IDS) are tools that can both analyze traffic and look for malicious signatures. An IDS is like a firewall that also reads the data in the packets it inspects, issues alerts/alarms, and blocks malicious traffic if configured to do so.

Explain that there are many varieties of intrusion detection systems, but today's class will focus on Snort, the world's most popular open-source solution.

Network security monitoring (NSM) is the process of identifying weaknesses within a network's defense. It also provides organizations with situational awareness of their network.

Note that Security Onion is a specific Linux distribution that's derived from Ubuntu. Security Onion uses the Snort IDS engine as its event-driven mechanism.

Intrusion Detection System Overview

Intrusion detection systems (IDS) are passive devices that perform packet captures of all traffic that passes through a network interface.

Intrusion detection systems are not designed to respond to an attack, but rather to document and log attacks for future analysis.

Intrusion detection systems help organizations enforce the cyber kill chain by establishing situational awareness of their adversaries, which may include intent and end objectives. Organizations can use this information to harden their defenses.

IDS Types

There are two primary types of IDS:

Signature-based IDS compares patterns of traffic to predefined signatures.

Good for identifying well-known attacks.

Requires regular updates as new attack signatures are released.

Vulnerable to attacks through packet manipulation that tricks the IDS into believing malicious traffic is good.

Unable to detect zero-day attacks.

Anomaly-based IDS compares patterns of traffic against a well-known baseline.

Good for detecting all suspicious traffic that deviates from the well-known baseline.

Prone to issuing false alerts.

Assumes normal network behavior never deviates from the well-known baseline.

Excellent at detecting when an attacker probes or sweeps a network.

Intrusion Detection Architecture

Intrusion detection systems have two basic architectures:

Network intrusion detection system (NIDS) filters an entire subnet on a network.

Matches all traffic to a known library of attack signatures.

Passively examines network traffic at the points it is deployed.

Relatively easy to deploy and difficult to detect by attackers.

Host-based intrusion detection system (HIDS) runs locally on a host-based system or user’s workstation or server.

Acts as a second line of defense against malicious traffic that successfully bypasses a NIDS.

Examines entire file systems on a host, compares them to previous snapshots or a baseline, and generates an alert if there are significant differences between the two.

Image Source

Intrusion Prevention System

An intrusion prevention system (IPS) does everything that an IDS can do, but can also respond to attacks. An IDS doesn’t alter or react to packets as they enter the network. An IPS does this by blocking malicious traffic and preventing it from being delivered to a host on the network.

The two main differences of an IPS and IDS:

IDS physically connects via a network tap or mirrored port or SPAN.

Network tap (Test Access Port) is a hardware device that provides access to a network. Network taps transit both inbound and outbound data streams on separate channels at the same time, so all data will arrive at the monitoring device in real time.

SPAN (Switched Port Analyzer), also known as port mirroring, sends a mirror image of all network data to another physical port, where the packets can be captured and analyzed.

IDS requires an administrator to react to an alert by examining what has been flagged.

IPS physically connects inline with the flow of data. An IPS is typically placed in between the firewall and network switch.

Requires more robust hardware due to the amount of traffic flowing through it.

Automatically takes action by blocking and logging a threat, thus not requiring administrative intervention.

An IDS generates an alert when a Snort rule detects malicious traffic that matches a signature. An alert is a message that’s created and sent to the analyst’s console as an indicator of attack (IOA).

There are two primary types of indicators:

Indicators of attack indicate attacks happening in real time.

Proactive approach to intrusion attempts.

Indicate that an attack is currently in progress but a full breach has not been determined or has not occurred yet.

Focus on revealing the intent and end goal of the attacker regardless of the exploit or malware used in the attack.

Indicators of compromise (IOC) indicate previous malicious activity.

Reactive approach to successful intrusions.

Indicate that an attack occurred, resulting in a breach.

Used to establish an adversary's techniques, tactics, and procedures (TTPs).

Expose all of the vulnerabilities used in an attack, giving network defenders the opportunity to revamp their defense as part of their mitigation strategy, and learn from an attack so it won't happen again.

Snort

Snort is a freely available open-source network intrusion detection\prevention system. It can perform real-time traffic analysis and log packets on a network. Snort is used to detect a wide variety of attacks.

Snort adds additional layers of defense that can be applied at various layers of the defense in depth model, including:

Perimeter IDS and IPS architecture

Network IDS and IPS architecture

Host IDS and IPS architecture

Configuration Modes

Snort can operate in three modes:

Sniffer Mode: Reads network packets and displays them on screen.

Packet Logger Mode: Performs packet captures by logging all traffic to disk.

Network Intrusion Detection System Mode: Monitors network traffic, analyzes it, and performs specific actions based on administratively defined rules.

Most Snort deployments use all three modes of operation.

Snort Rules

Snort uses rules to detect and prevent intrusions. Snort operates by:

Reading a configuration file.

Loading the rules and plugins.

Capturing packets and monitoring traffic for patterns specified in the loaded rules.

When traffic matches a rule pattern, generating an alert and/or logging the matching packet for later inspection.

Rules can direct Snort to monitor the following information:

OSI layer: Watches for IP vs. TCP data.

Source and destination address: Where the traffic is flowing from and to.

Byte sequences: Patterns contained in data packets that might indicate malware, etc.

Consider the following Snort rule:

alert ip any any -> any any {msg: "IP Packet Detected";}

This rule logs the message "IP Packet Detected" whenever it detects an IP packet.

Another example:

alert tcp any 21 -> 10.199.12.8 any {msg: "TCP Packet Detected";}

This rules triggers an alert whenever a TCP packet from port 21, with any source IP address, is sent to the IP 10.199.12.8. With each alert, it will print the message "TCP Packet Detected."

Rule Header

alert: The action that Snort will take when triggered.

tcp: Applies the rule to all TCP packets.

any: Applies the rule to packets coming from any source IP address.

21: Applies the rule to packets from port 21.

->: Indicates the direction of traffic.

10.199.12.8: Applies the rule to any packet with this destination IP address.

any: Applies the rule to traffic to any destination port.

Rule Option

{msg: "TCP Packet Detected";}: The message printed with the alert.

Snort provides many additional actions and protocols, which can be combined to design rules for almost any type of packet.

04. Intrusion Detection Systems and Snort Activity

Activity File: Intrusion Detection Systems and Snort

05. Review Intrusion Detection Systems and Snort Activity

Solution Guide: Intrusion Detection Systems and Snort

06. Instructor Do: Network Security Monitoring and Security Onion

Network Security Monitoring Overview

On November 24, 2014, a hacker group called Guardians of Peace released confidential information from Sony Pictures that contained personally identifiable information for all its employees, including full names, home addresses, social security numbers, and financial information.

It was later discovered that the assailants had been lurking in Sony's network for 17 months prior to the discovery of the breach.

As a result, a number of executives and upper management were fired, all employees had their PII exposed, and the company suffered massive damage to its reputation. Sony was also forced to pay large fines for violating federal regulations.

If Sony Pictures had put a network security monitoring program in place, they would have discovered the attack much sooner, perhaps within hours.

NSM would have allowed Sony to stop the attack immediately, while gaining a good understanding of the tactics, techniques, and procedures (TTPs) used by the adversary to penetrate the network.

This could have been accomplished by adding additional layers of defense in the form of an NIDS, NIPS, and HIDS as part of an NSM program.

Security monitoring highlights the failures of existing security controls through the use of data analysis tools. NSM is most useful when the front-end layers of defense are compromised.

It takes time for intruders to achieve their objectives. In many cases, infiltrators spend hours, weeks, months, or even years inside of a network before achieving their final objectives. It’s during this critical time that NSM can work to slow and/or eliminate threat-based attacks.

NSM is threat-centric. Its primary focus is the adversary, not the vulnerability.

NSM is focused on visibility of an attack, not the response to the attack.

NSM also reveals statistical data related to specific IOAs and IOCs from attacks.

NSM Strengths

NSM can only protect against the adversarial tactics that it can detect. This detection process takes place when collected data is inspected and irregularities are identified.

NSM allows organizations to:

Track adversaries through a network and determine their intent.

Acquire intelligence and situational awareness.

Be proactive by identifying vulnerabilities.

Be reactive through incident response and network forensics.

Provide insights related to advanced persistent threats (APTs).

Uncover and track malware.

NSM Weaknesses

It’s important for security administrators to know the limits of their defenses so they can better prepare new ones. NSM capabilities are extremely limited in the following situations:

Encrypted traffic and VPNs: NSM and IDS do not have the capability to read encrypted traffic. Adversaries will often use this tactic to bypass security defenses.

Underpowered hardware: NSM and IDS require adequate amounts of processing and memory to function well. Larger networks have more traffic, requiring more powerful hardware requirements, and larger expenses.

Mobile communication platforms: Adversaries often use mobile radio communications to obfuscate their attacks because it's difficult for NSM and IDS to monitor radio transmission waves.

Legal and privacy issues: NSM is an invasive process that monitors and records all network data as it passes through. Therefore, there may be legal implications regarding certain types of data collected by an NSM.

Limited access to network taps: The placement of an NSM sensor can be limited at certain areas of the network.

NSM Stages and Processes

NSM operates under two distinct stages, each with two processes.

Detection: In this stage, an alert is first generated in the Sguil analyst's console. (Sguil, which you'll learn about in a moment, is a tool that collects alert data from Snort. )

Collection: The event is observed and the data is stored in a PCAP file.

Analysis: The alert data is identified, validated, documented, and categorized according to its threat level.

Response: In this stage, a security team responds to a security incident with two processes:

Escalation: All relevant individuals are notified about the incident.

Resolution: The process of containment, remediation, and any additional necessary response.

NSM Sensor Connectivity

Intrusion detection systems are generally placed at strategic locations in a network where traffic is most vulnerable to attack. These devices are typically placed next to a router or switch that filters traffic.

An IDS can be physically connected to a network in two ways:

Mirrored port or SPAN: A SPAN port is a function of an enterprise-level switch that allows you to mirror one or more physical switch ports to another port. A mirror image of all data will flow across both ports equally. This is what allows the IDS to perform packet captures on all inbound and outbound traffic within a network.

Network Test Access Point (TAP): TAPs allow us to access our network and send that data in real time to our monitoring systems. One example of a TAP is known an aggregated TAP, in which a cable connects the TAP monitor port with the NIC on the sensor. This specific placement allows traffic to be monitored between the router and switch.

Security Onion

Today we'll work with Security Onion, a network security monitoring platform that provides context, intelligence, and situational awareness of a network.

Security Onion is an Ubuntu-based, open source Linux distribution that contains many NSM tools used to protect networks from attacks. Security Onion adds multiple layers of defense and helps enforce the cyber kill chain.

We'll be using a few NSM tools to help us with an incident detection and response routine that will simulate a real world situation.

The tools we will be using are:

Sguil: Pulls together alert data from Snort. It provides important context for alerts, which we can use to complete more detailed analysis of the data.

Transcript: Provides us a view of PCAP transcripts that are rendered with tcpflow, the equivalent to following TCP streams in Wireshark.

NetworkMiner: Performs advanced network traffic analysis. Extracts artifacts from PCAP files and provides an intuitive user interface to analyze them with. Allows the analyst to analyze, reassemble, and regenerate transmitted files and certificates from PCAP files.

Alert Data

Snort watches and interprets network traffic and creates a message when it sees something it is programmed to report. These alerts are based on patterns of bytes, counts of activity, or even more complicated options that look deeply into packets and streams.

Sguil

Sguil has six key functions that help NSM analysts with their work:

Performs simple aggregation of alert data records.

Makes available certain types of metadata and related data.

Allows queries and review of alert data.

Allows queries and review of session data.

Allows easy transitions between alert or session data and full content data, which is rendered as text in a transcript or in a protocol analyzer like Wireshark.

Exposes features so analysts can count and classify events, enabling escalation and other incident response decisions.

Sguil is made up of four main sections:

Alert Panel: Displays detailed alert data, including:

Source and destination IP

Source and destination port

Alert ID/severity

Event message (message generated by Snort rule option)

Snort Rule: The Snort rule that generated the alert, obtained from the IDS engine.

Packet Data: PCAP file showing header and payload information of the data.

IP Resolution: Displays reverse DNS lookup information.

Sguil's Alert Panel

As shown in the screenshot, the Snort IDS generated the alert GPL ATTACK\_RESPONSE id check returned root. The analyst must decide if this is benign or malicious. This demonstration will focus on how to obtain data and use tools and process to make this decision.

The alert panel has four fields that we should look at:

ST or Status: Colors indicate severity levels of "real-time" or "RT" events.

Red: Critical, possible data breach in progress. Must be resolved immediately.

Orange: Moderate, high potential for data breach. Requires immediate review.

Yellow: General, low potential for data breach. Requires review.

Alert ID: A randomly generated numerical ID created by Sguil to itemize alert data.

Source IP: IP address of the source identified by the alert.

Event Message: The message generate by the Snort rule option.

Sguil's Snort Rule and Packet Data Sections

The next screenshot is an example of the Snort rule set in Sguil that activated this alert.

Snort Rule: In the top portion of the window, we see the Snort NIDS engine that generated the alert data when traffic matched one of its rules.

Alert data is an indicator of attack. An analyst may have to determine if it represents benign or malicious activity.

Alert data from the Snort NIDS stores entries in the Event Messages column that begin with text like "ET" (for Emerging Threats, an IDS rule source).

Packet Data: The lower, more colorful part of this window is the portion of Sguil that performs network packet analysis.

The packet analyzer shows a detailed view of the data capture. It includes packet header information and data streams presented in hex and text form.

Sguil's IP Resolution Section

This section of Sguil's analyst console provides reverse DNS lookup information. This is used to reveal identifying information about the attacker. This includes their domain name registries and IP addresses.

Other information may include the country of origin, and, ideally, the names, email addresses, and/or phone number of the DNS registrants.

Analysts can use the data obtained from IP resolution to formulate attacker profiles.

07. Security Onion and NSM Overview

Activity File: Security Onion and NSM Overview

08. Review Security Onion and NSM Overview Activity

Solution Guide: Security Onion and NSM Overview

10. Instructor Do: Alert - FTP File Extraction

There will be many times when an alert requires an analyst to do some data mining. A security analyst must have a thorough understanding of how NSM tools are integrated in order to do this. These skills help speed up incident and response efforts.

In the following walkthrough, we will explore the Security Onion interface, using Sguil as the starting point for learning other NSM tools for security investigations.

Security Onion Demo

The first thing we will do is search using a filter for the IP address from the indicator of attack (IOA).

Launch an instance of Security Onion. Do the following steps:

Click the Sguil desktop icon and launch the application.

When prompted, select both networks to monitor.

Click Query in the top toolbar.

Click Query by IP in the dropdown menu.

Next, input the IP address that we obtained from the IOA alert:

Enter the IP address obtained from the alert: 128.199.52.211.

Click Submit.

Now, we will only see information related to our filtered alert data, as seen below.

Alert information:

NSM sensor that triggered the alert.

Source and destination IP.

Source and destination port.

Date and time of the alert.

Event message (defined in the Snort IDS rule option used to generate the alert).

Reverse DNS lookup information.

Snort rule that triggered the alert.

Server response message in the packet data section.

Since we've now compiled critical information from the attack signature, we have a partial picture of the attack profile. Let's continue our network forensic investigation.

From the information we've gathered so far, we can conclude the following:

This attack occurred as the result of a drive-by attack that used the HTTP protocol.

A drive-by attack is when a user navigates to a webpage that has built-in malicious scripts running in the background.

Drive-by attacks are dangerous because the user doesn't need to click anything on the webpage to launch the attack. The mere act of opening the webpage creates a session in the background without the user knowing, which prompts malicious software downloads.

Now that we have this knowledge, we know we must search for any files that may have been downloaded to this particular host (the victim).

Next, we'll introduce a new forensic tool that can extract any files that were installed on the user's machine, and provide us with an attacker profile.

NetworkMiner

NetworkMiner is an NSM tool that's included as part of the Security Onion NSM suite of tools.

NetworkMiner performs advanced network traffic analysis (NTA) of extracted artifacts, and presents them through an intuitive user interface.

From our Sguil window, we will switch to NetworkMiner by following the steps below.

Instructor Note: Details may vary slightly from screenshot but the concepts still apply.

Sort the alert IDs from low to high by clicking on Alert ID at the top of the column. Right-click on the first Alert ID at the top.

Click on NetworkMiner in the dropdown menu.

Now we are presented with NetworkMiner's interface. Pay attention to the tabs at the top. We'll focus on the Files tab next.

Select the Files (4) tab as seen in the screenshot. This tells us that NetworkMiner was able to extract and reconstruct the four files used in the attack, from Security Onion's PCAP captures.

Source port and protocol used (Box 3).

Protocols used during transmission (Box 4).

Right-click on one of the files.

Click on Open folder.

This will open the folder on the local hard disk where NetworkMiner stores the recompiled files.

Next, open the Chromium Web Browser, navigate to [www.virustotal.com](http://www.virustotal.com/), and select File.

These are parts of the malware. We can click and drag on any one of the files and get results. Drag the d10.octet-stream file to the Choose file box in the VirusTotal window.

This opens the VirusTotal search tool, which will attempt to match these files to any known malware signatures.

VirusTotal returned the number of matches it discovered for well-known virus engines against this particular file.

VirusTotal hashes the files, which establishes a malware signature used to look for a match and determine the common name for the malware.

VirusTotal lists all of the common names for this specific malware.

Security professionals, especially security researchers, use this website frequently when performing malware analysis and establishing the tactics, techniques, and procedures used by adversaries to infiltrate networks. This information contributes to what is commonly referred to in the industry as an "attacker profile." These help us know our enemies in order to better defend against future attacks.

Summary

Computer Incident Response Teams can use the vast amount of information accumulated by NSM to formulate the tactics, techniques, and procedures used by an adversary.

NSM allows organizations to establish situational awareness within their networks by enforcing the cyber kill chain.

11. Alert - FTP File Extraction

Activity File: Alert - FTP File Extraction

12. Review Alert - FTP File Extraction

Solution Guide: Alert - FTP File Extraction

11.3 Student Guide: Enterprise Security Management (ESM)

Overview

In today's class, students will advance their network security knowledge by learning enterprise security management (ESM) and how host-based OSSEC IDS technology plays a critical role in endpoint telemetry. Students will expand their investigations of threats using Security Onion's Elastic Stack and the web-based data analytics visualization tool, Kibana using a process known as "Cyber Threat Hunting".

Class Objectives

By the end of today's class, you will be able to:

Analyze indicators of attack for persistent threats.

Use enterprise security management to expand an investigation.

Use OSSEC endpoint reporting agents as part of a host-based IDS alert system.

Investigate threats using various analysis tools.

Escalate alerts to senior incident handlers.

Slideshow

The lesson slides are available on Google Drive here: 11.3 Slides

01. Security Onion Set Up

Activity File: Security Onion Setup

02. Overview and Alert - C2 Beacon Setup (0:20)

Network Security Recap

On Day 1, we covered how firewalls protect a network. On Day 2, we expanded our layers of network security to cover IDS and IPS systems.

Today, we will turn our focus to learning how an adversary conducts network security attacks. Then, through a process known as cyber threat hunting, we will use advanced network security tools, such as Security Onion and ELK, to gain a deeper understanding and situational awareness of a network's security posture.

C2 Alert Beacon Set Up

In the first activity, we'll explore how command and control (C2) servers are used to create a specific type of alert against attacks that use persistence as part of its attack campaign.

NSM plays a critical role in implementing a defense in depth approach, serving as an additional layer of protection when an adversary bypasses defenses.

Attacks against these servers make infected hosts call back to C2 servers. These callbacks, referred to as "keep alives", serve as beacons that keep the back channel open, therefore enabling access in and out of the network at all times.

These keep alive beacons activate a specific alert. In the screenshot below, we see an alert identified as a C2 beacon acknowledgement. Note it includes the text CnC Beacon Acknowledgement in the Event Message.

There is a reference URL specified within the Snort rule option.

Sometimes, writers of Snort rules will put links in their rule options to help network defenders establish TTPs.

With this information, network defenders can form mitigation strategies to help improve their security posture.

03. C2 Beacon Activity

Activity File: C2 Beacon

04. Review C2 Beacon Activity

Solution Guide: C2 Beacon

05. Enterprise Security Monitoring

Now that we've learned about the benefits of using firewalls and NSM, we must move from traditional network-based IDS engines, such as Snort, to the more all-encompassing enterprise security monitoring (ESM), which includes endpoint telemetry.

OSSEC

Firewalls and NSM cannot see inside of encrypted traffic. This is major limitation because:

In most cases, malware will be transmitted from attacker to victim in an encrypted state, in order to hide its presence and intent. This also serves as a method of obfuscation to bypass IDS detection engines.

Since malware cannot activate in an encrypted state, it must be decrypted. This can only happen after it is installed on the victim’s machine. This is where ESM and, more specifically, endpoint telemetry become relevant.

ESM uses OSSEC to provide visibility at the host level, where malware infection takes place after it's decrypted.

OSSEC is the industry's most widely used host-based IDS (HIDS). It has many configuration options and can be tailored to the needs of any organization.

Endpoint telemetry as host-based monitoring of system data.

OSSEC agents are deployed to hosts and collect syslog data. This data generates alerts that are sent to the centralized server, Security Onion.

Security administrators can then use Security Onion to form a detailed understanding of the situation and reconstruct a crime.

Elastic Stack

OSSEC monitors all of the syslog data that it sees. However, not every syslog entry will generate an alert. Security admins will need to switch to other tools to fully analyze packet captures.

These other tools are known as the Elastic (ELK) Stack, the engine that operates within Security Onion. It consists of three important components:

Elasticsearch is considered the heart of the Elastic Stack. It is a distributed, restful search and analytics engine built into Security Onion that is capable of addressing thousands of data points seen within network traffic. It helps security administrators locate the expected and uncover the unexpected.

Logstash is an open-source, server-side data processing pipeline built into Security Onion. It ingests data from many sources at the same time by transforming it and sending it to designated log files, known as stashes.

Kibana is a browser-based visualization interface. It uses thousands of data points from the Elastic Stack as its core engine.

These tools work together with OSSEC to make a comprehensive alert data process:

OSSEC agents generate an alert.

OSSEC sends alert data gathered from syslog to Security Onion's OSSEC server.

The OSSEC-generated syslog alert is written to Logstash for storage.

Log data is ingested into the Elasticsearch analytics engine, which parses hundreds of thousands of data points to prepare for data presentation.

Users interact with the data through the Kibana web interface.

Investigation, Analysis, and Escalation Demo

In this demo we will discuss using several tools in the ELK stack. We will focus on how these tools work, and not a specific attack.

We will also focus on the process of escalation within a Security Operations Center:

A junior analyst working in a Security Operations Center will belong to a multi-tier group of analysts. Junior analysts typically perform the initial triage of alerts and then escalate these events to senior incident responders.

This process and the tools involved will be our focus.

We'll begin our investigation with a new tool called Squert:

Click on the Squert desktop icon and enter the same credentials you used for you Sguil login.

After logging in, we may need to change the date range to ensure we see all the alert data in our system. Click on the date range as illustrated below.

The default view shows alerts from today. In order to show older alerts, click INTERVAL, then click the two right arrows to set your custom date.

In this example we'll change the year to 2014 in the START field. This date range should cover all alerts used in the PCAPs.

Click on the circular arrows to reload the web page and refresh the alert data for the newly selected date range.

Next, click on the word QUEUE to arrange the priorities from the highest count to the lowest. It may require two clicks.

Clicking on a red number box will drop down that line and reveal several important items.

We can see URL links to two websites that provide additional insights into the attack.

As security administrators, we can use this research later in our incident investigations. It's encouraged to accumulate information from several different resources.

The screenshots below show the articles found at the links.

The articles, written by two different security researchers, provide incident responders with different insights into the same attack.

Click on the Views tab at the top.

The Views tab displays traffic as it flows between a source and destination IP.

Scroll down to see more.

Thicker bands indicate higher volumes of traffic.

This visualization indicates to security administrators potential problem areas that may require further investigation.

Hover the mouse over a band and a window will pop up displaying the flow of traffic between source and destination IP using directional arrows.

The number of transmissions that have occurred is also displayed.

Next we'll use Elastic Stack's data analytics engine through Kibana's web-based visualization tool.

Minimize the Squert window and open Sguil.

Launch Kibana by doing the following:

Right-click on any IP address.

A dropdown menu will appear. Select Kibana IP Lookup, and then select either the destination (DstIP) or source IP (SrcIP).

After Kibana launches, you may be prompted to log in. If so, log in with the same credentials used for the Sguil client.

Once the web browser launches, it's best practice to verify that Kibana is using the correct filter, as seen in the screenshot below.

The IP address in the Kibana filter should match the one that we right-clicked to pivot from the Sguil client. In this case, it matches the IP that we used for this pivot, which is good.

We have now started using the powerful Elastic Stack data analytics engine.

Elastic Stack is the heart of Security Onion's enterprise security monitoring capabilities. Kibana is the interface that provides insight into endpoint telemetry by interpreting the OSSEC agent syslog data.

In this next example, we'll begin our investigation by scrolling down to the HTTP - Destination Ports section.

At this point in an investigation, we are looking for non-standard HTTP ports.

For example, if we saw port 4444 indicated here, that would be a clear indicator that a Metasploit Meterpreter session was in progress.

Scroll down to the MIME word cloud.

MIME (Multipurpose Internet Mail Extension) types (not a file extension), are used by browsers to determine how URLs are processed.

Therefore, it is important that web servers use the correct type of MIME.

Incorrectly configured MIME types are misinterpreted by browsers and can cause sites to malfunction and mishandle downloaded files.

Attackers use this to their advantage.

In the graphic below, the Elastic Stack data analytics engine is displayed through Kibana's MIME - Type (Tag Cloud) visualization window.

The more a MIME type is discovered, the larger it appears in the word cloud.

If we scroll down a little further, we see HTTP - Sites, which lists the number of times particular websites have been visited. This is a good place to look for suspicious websites.

The HTTP - Sites Hosting EXEs section lists websites that were used to either download or search for an EXE. Again, anything that looks malicious will require further investigation.

To investigate a malicious website, we apply a filter by hovering our mouse over the count and clicking the + sign. This will filter out all other websites.

In the graphic below, the arrow pointing to the left will drop down the contents of the selected log, revealing its contents.

The arrow pointing to the right is the hyperlink to the PCAP file.

Clicking on this link will launch the PCAP in another window and display the TCP conversation using either TCP or HTTP flow.

Now, using the image above as an example, click on the triangle arrow pointing to the right.

This will drop down the log file and reveal its contents, as shown in the screenshot below.

In our screenshot example, we can see the message A Network Trojan was detected, as indicated in the classification field.

We can also see that the event\_type is indicated as snort.

An event type is the source of the alert, i.e., the application that generated the alert.

This entry is the result of an endpoint Snort IDS engine alert.

Scroll up and click the link under \_id. We can see the PCAP pivot. We can learn three facts from this view:

The configuration is set to IDS and not operating in IPS mode. A download could have occurred.

We can see the HTTP response SRC: Connection: Close, meaning it closed when the victim got to this page.

We can also see an error 302, meaning the website had moved.

From this we can assume the following likely happened:

The victim clicked on a malicious link.

The link opened a window which downloaded or attempted to download the trojan.

The window quickly closed itself.

Further analysis is required.

Once we determine an alert needs further analysis, we will escalate the event to a senior incident handler for further review.

Return to your Sguil window.

Right-click RT in the status column.

Select Update Event Status.

Select Escalate.

Add a comment: "Trojan may have been downloaded."

Click Okay.

Note: This will move the alert from the Real Time alerts queue to the Escalation queue.

We can verify the escalated event by:

Selecting the Escalated Events tab.

Right-clicking on the event.

Selecting Event History.

Verifying the note that was entered by the junior analyst.

Summary

This demonstration covered how to conduct investigations using various threat hunting techniques. We focused on a few of the many ways to start an investigation.

ESM (enterprise security monitoring) includes endpoint telemetry, host-based monitoring of system data that uses OSSEC collection agents to gather syslog data.

To investigate network-based IDS alerts, security administrators must use enterprise security monitoring, which includes visibility into endpoint OSSEC agents.

IDS alerts are snapshots in time. They raise questions that need answers. With the use of Security Onion, security admins can use PCAPs to reconstruct a crime.

07. Investigation, Analysis, and Escalation

Activity File: Investigation, Analysis, and Escalation

09. Review Investigation, Analysis, and Escalation Activity

Solution Guide: Investigation, Analysis, and Escalation Activity

10. Threat Hunting - Cyber Threat Intelligence

Threat intelligence is important at every level of government and public sector organizations, which use it to determine acceptable risk and develop security controls that inform budgets.

Malicious actors have various motivations. For example:

Hacktivist organizations are politically motivated.

Criminal hackers are financially motivated.

Cyber espionage campaigns, most typically associated with nation states, steal corporate secrets.

Knowing the motivations for attacks against your organization will help you determine the security measures necessary to defend against them.

Threat Intelligence Cards

As a member of the Computer and Incident Response Team (CIRT), one of your responsibilities is to establish a threat intelligence card, which documents the TTPs used by an adversary to infiltrate your network.

When handling a large-scale intrusion, incident responders often struggle with organizing intelligence-gathering efforts.

Threat intelligence cards are shared among the cyber defense community, allowing organizations to benefit from the lessons learned by others.

Cyber threat intelligence centers on the triad of actors, capability, and intent, along with consideration of TTPs, tool sets, motivations, and accessibility to targets.

These factors inform situational aware decision making, enhanced network defense operations, and effective tactical assessments.

11. Threat Hunting - Cyber Threat Intelligence

Activity File: Threat Hunting - Cyber Threat Intelligence

12. Review: Threat Hunting - Cyber Threat Intelligence Activity

© 2020 Trilogy Education Services, a 2U, Inc. brand. All R