# Introduction to Firewalls and Network Security

Cybersecurity

Network Security Day 1

# Class Objectives

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

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

Use firewalls to protect a machine's open ports.

Develop and implement firewall policies with UFW and firewalld.

## Last week:

We learned how cryptographic tools help maintain confidentiality, integrity, and availability of data at rest and in transit.

# Limits of Cryptography

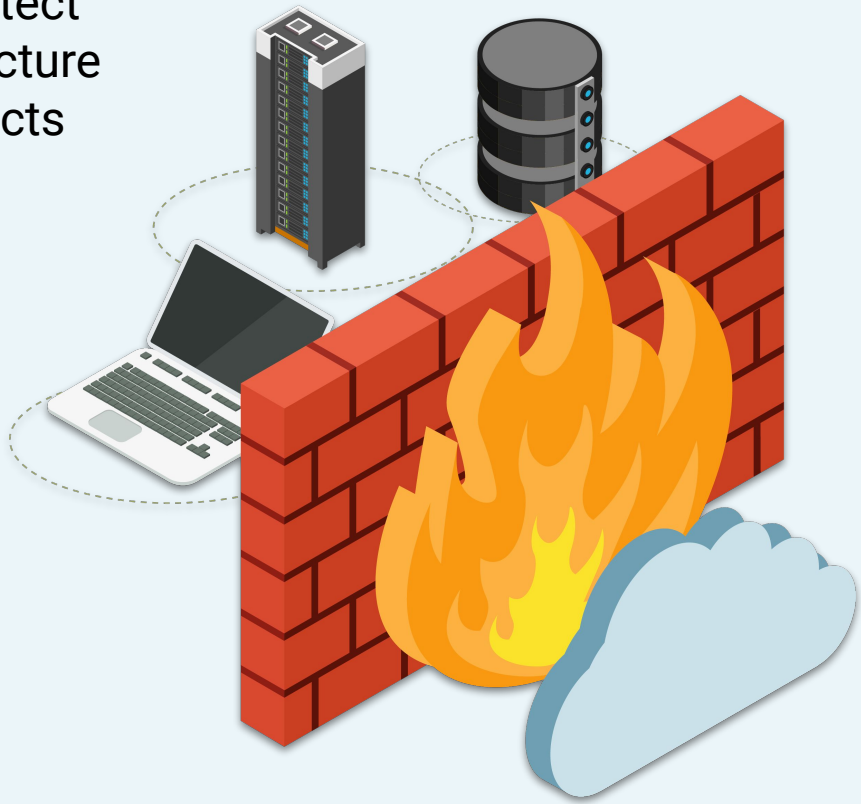
Suppose a malicious actor attempted to hack into a hospital's web server in order to perform a ransomware attack.

- In an effort to keep the private health information of their patients secure, the hospital protects secured access to its website using a cryptographic RSA VPN connection.
- While these tools are important, cryptography is only a part of a multifaceted network defense ecosystem that's used to protect private information and critical network infrastructure.

# The Need for Network Security

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

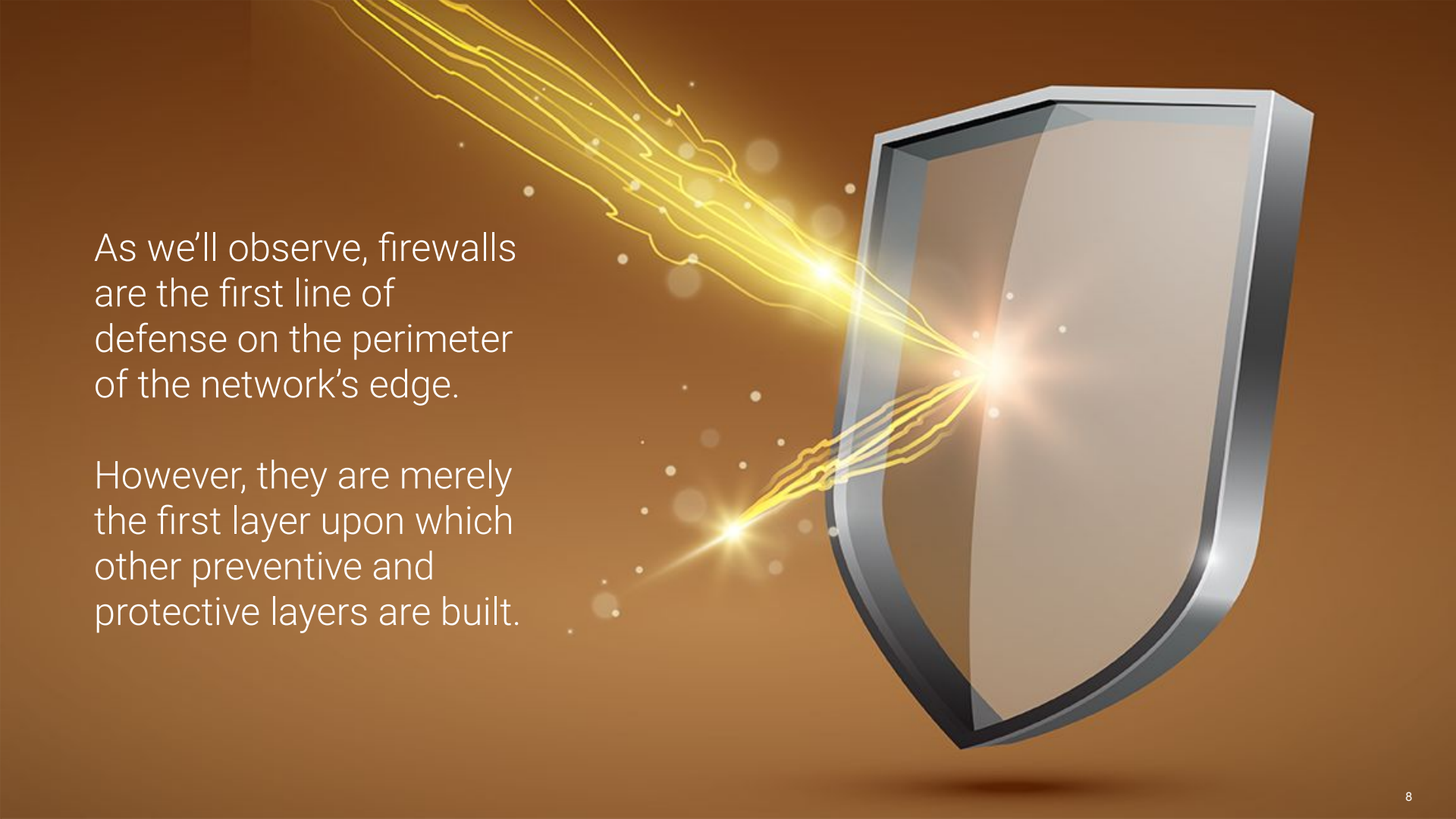
- For example, suppose the same website is the target of a denial of service (DoS) attack. Cryptography defenses aren't of much use.
- Therefore, security practitioners need to implement **network security** strategies like **firewalls** that provide an additional layer of defense.



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

**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**



As we'll observe, firewalls are the first line of defense on the perimeter of the network's edge.

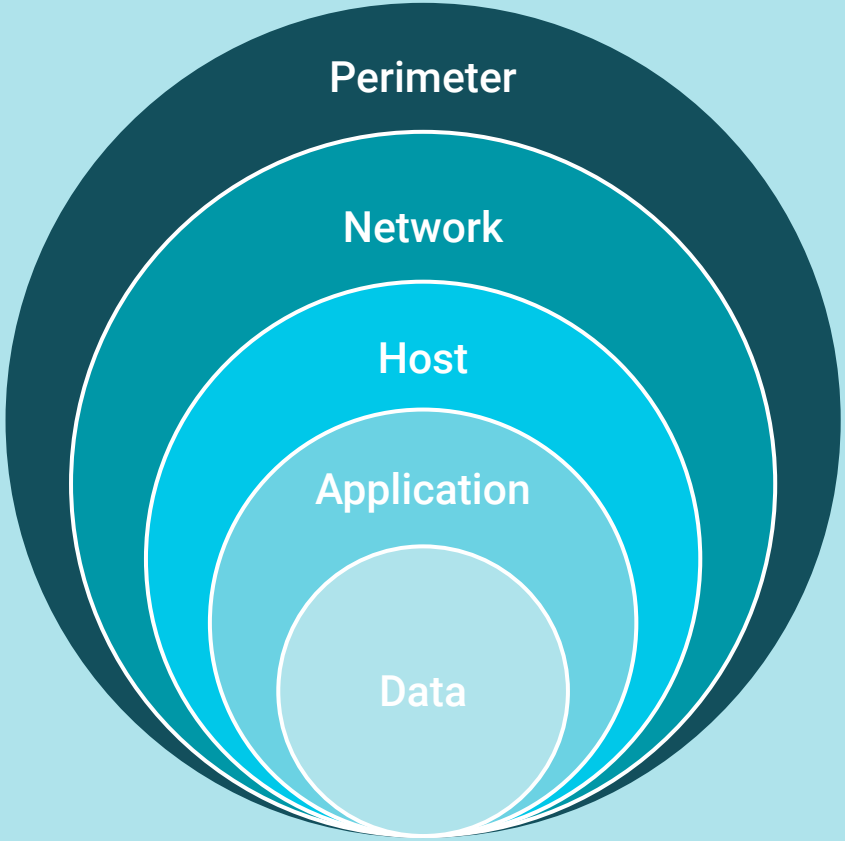
However, they are merely the first layer upon which other preventive and protective layers are built.

# Defense in Depth

In the GRC unit, we covered how defense in depth (DiD) plays a critical role in securing organizations.

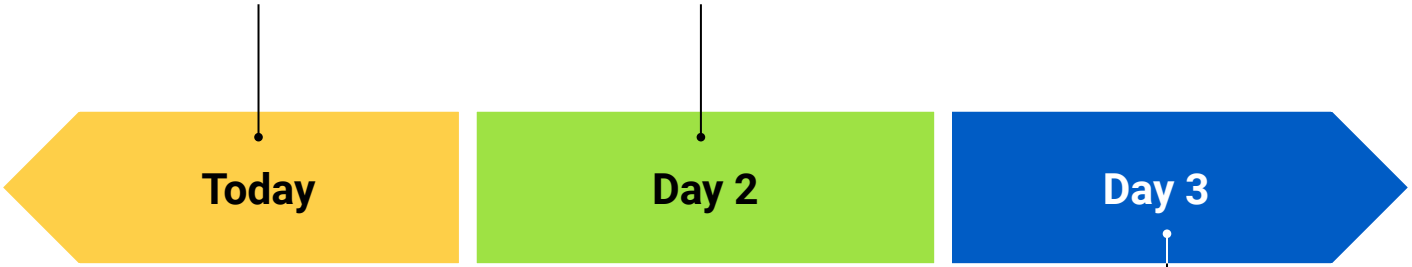
**Throughout this unit:**

- We will apply a **DiD mindset** to layer security measures that aim to slow an attack's progression, thus providing network defenders with enough time to respond.
- We will learn how to apply a **DiD methodology** to create expansive layers of network security.



**We will cover:**
the benefits and limitations of various types of firewalls.

**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.




**We will conduct:**
advanced cyber threat hunting using Enterprise Security Management (ESM), which expands upon NSM through the inclusion of endpoint telemetry.

# Today's Class

Today's class will progress as follows:

01 How open ports contribute to a machine's attack surface.

02 How firewalls are used to protect a computer's open ports.

03 Usage of different types of firewalls and their application.

04 The role firewalls play within a layered defense.

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

# Network Security

Knowledge of computer networking is essential for the following technical roles:

| | |
|---|---|
| **Help Desk/IT specialist** | Knowing if and how firewalls affect user traffic can help with troubleshooting issues like slow connections, lack of connection, and broken networked applications, such as Skype or Facebook Messenger. |
| **System/network administrator** | Often determine who is allowed to access devices on a network. These roles must develop firewall policies and implement them, using tools like UFW or firewalld. |
| **SOC analyst** | Need to understand how firewalls on the network filter incoming and outgoing packets in order to accurately interpret traffic logs. |
| **Penetration testers** | Must know if a firewall is between them and their target, as well as which rules are enabled, in order to launch a successful attack against a network. |

# Defense in Depth

# Firewall Architecture and Defense Methodologies

# Ports Recap

Networks allow computers to communicate with one another by sending data to and from open ports on other machines.

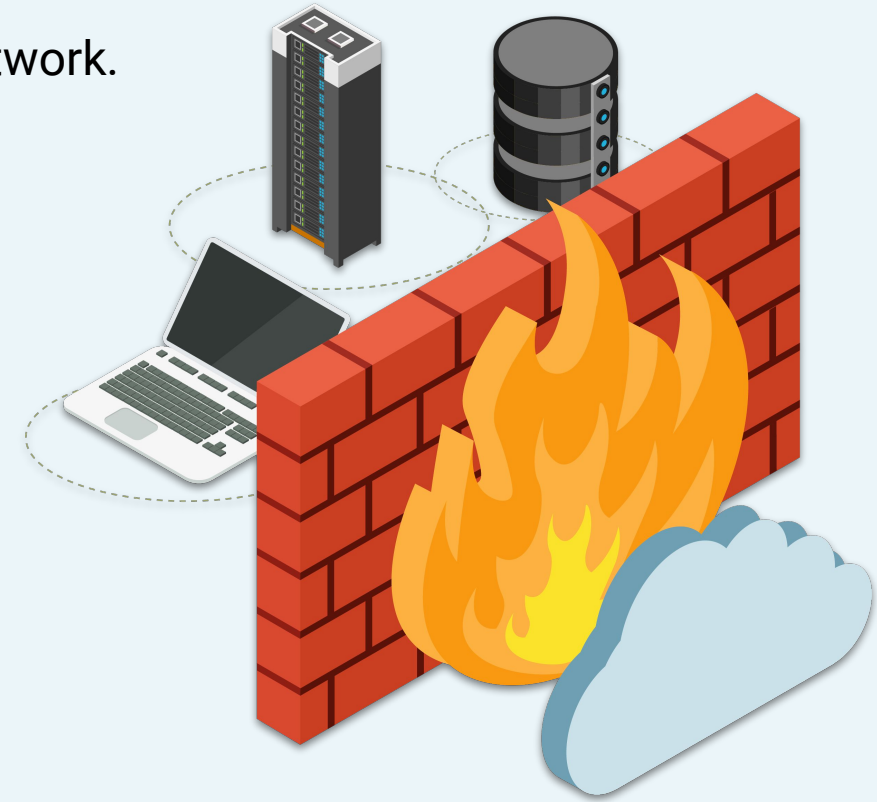
Devices must expose open ports in order to communicate with other machines on the network.

- We can’t assume that the only people who will connect to a device are those we trust.
- Malicious actors will exploit this assumption to access sensitive information.
- Restricting access to open ports is a fundamental skill for any technical security specialist.

# Firewalls

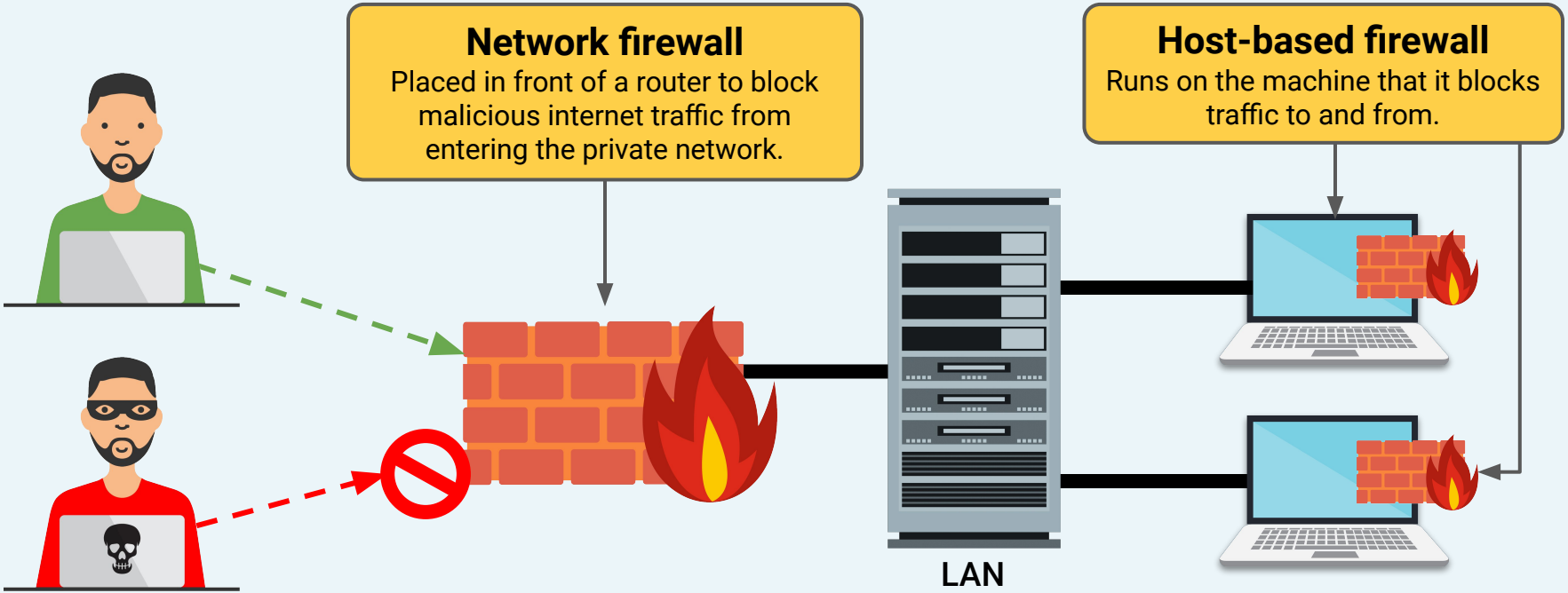
Firewalls provide a layer of protection by analyzing data leaving and entering a network.

- They are placed between application servers and routers to provide access control.
- Protect trusted networks by isolating them from untrusted networks, like the internet.



# Firewalls

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



# Firewalls

Network-based and host-based firewalls work in the same way:

Intercept traffic before it reaches its target host or router.

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

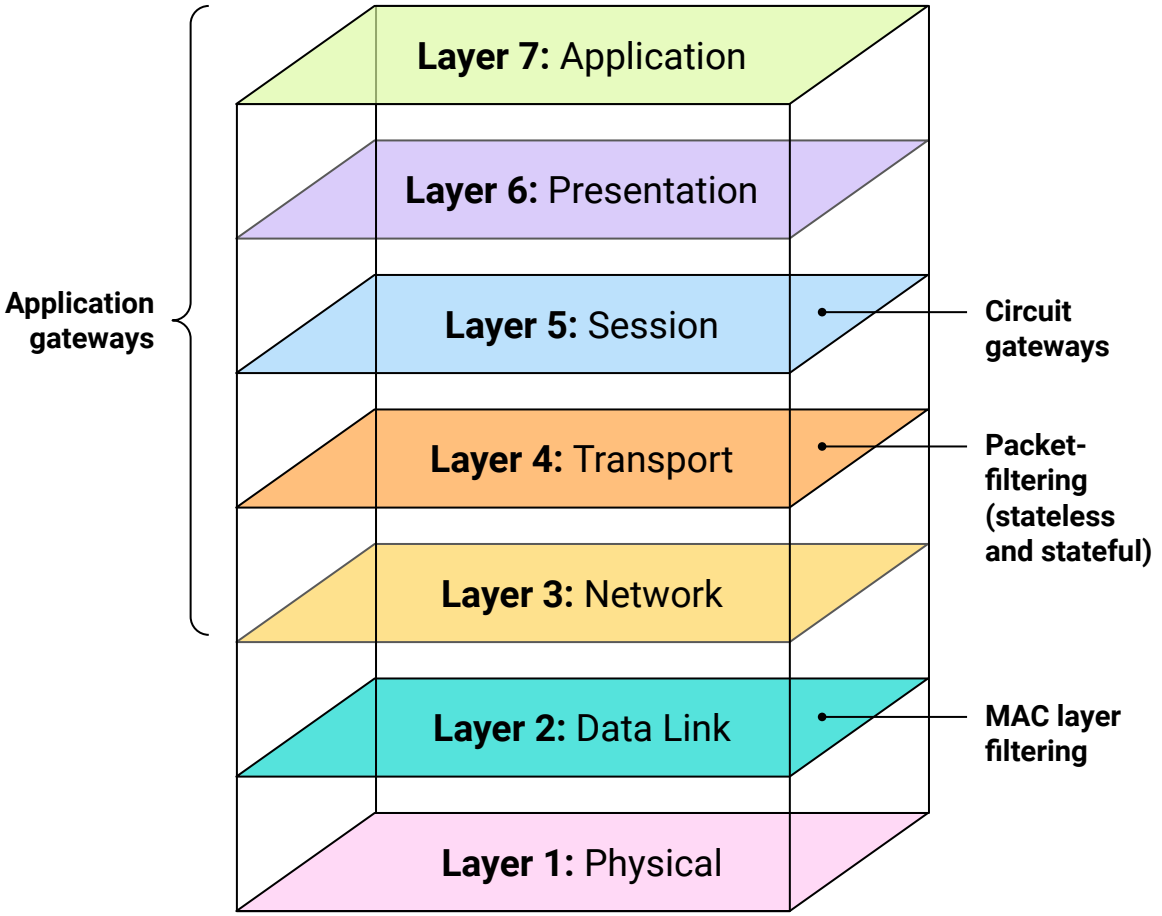
Allow packets that come from trusted sources and deny packets that don’t.

# Firewalls on the OSI

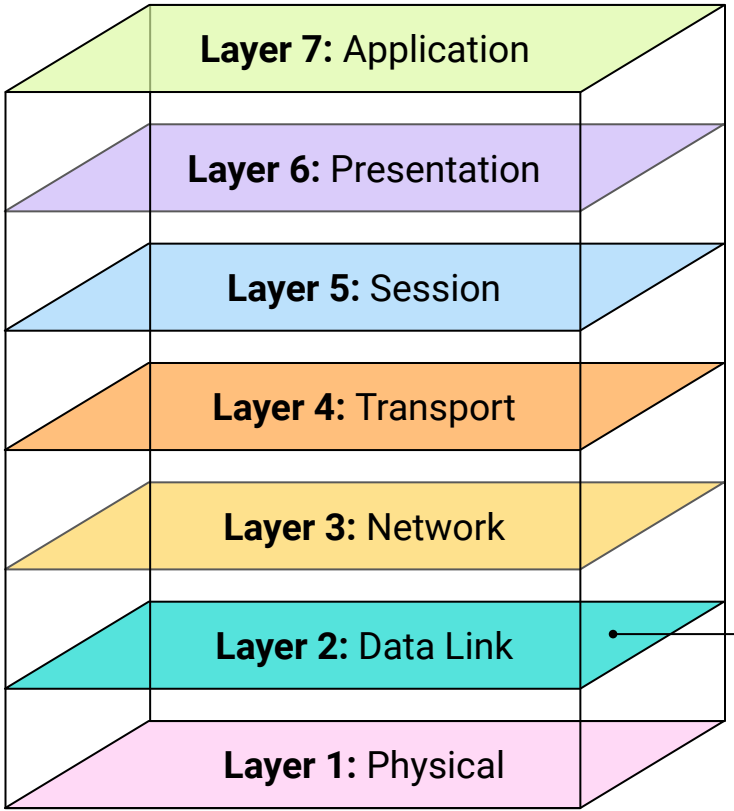
Firewalls are multifunctional network security appliances that operate at multiple layers of the OSI model.

**They can be broken down into four basic types:**

- Application gateways
- Circuit-level gateways
- Packet filtering
- MAC



# MAC Layer Firewall




**REMEMBER**

The media access control (MAC) address is a unique hardware ID that helps devices communicate with each other.

MAC firewalls operate at **Layer 2** of the OSI and filter based on source and destination MAC addresses.

# MAC Layer Firewall

Routers compare the MAC address of a device against an approved list.
If there is a match, the traffic is forwarded to that device.

## Advantages

Can secure the network from novice attackers.

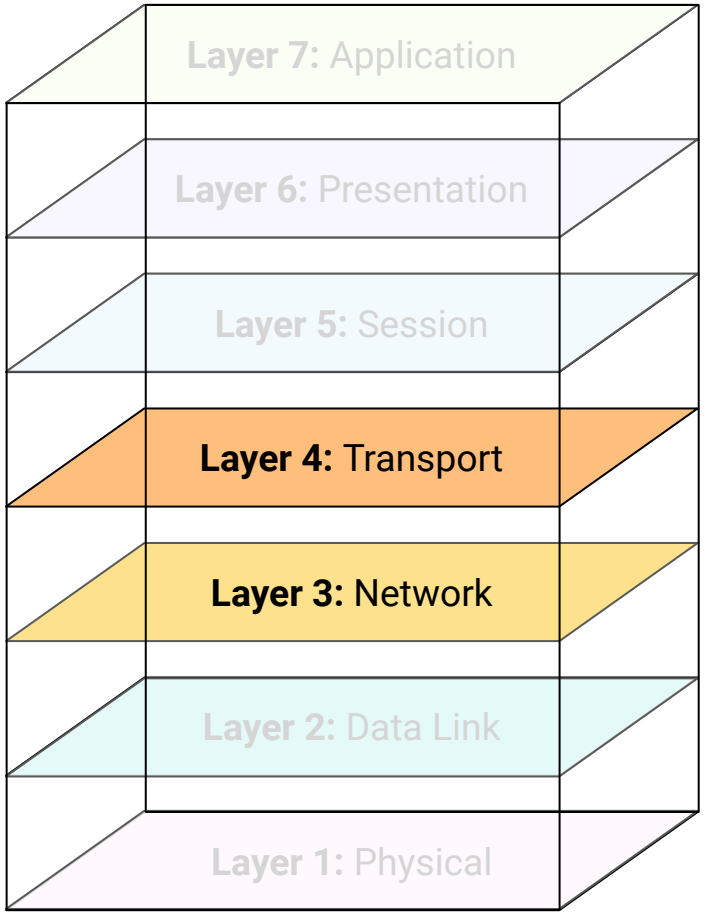
## Disadvantages

Can be easily bypassed by MAC spoofing.

# Packet-Filtering Firewalls (Stateless)

Stateless packet-filtering firewalls operate at **Layer 3** and **Layer 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).




**Use rules based on individual packets:**

- Source and destination IP address
- Source and destination port information
- IP protocols
- Ingress and egress interface

# Packet-Filtering Firewalls (Stateless)

Stateless packet-filtering firewalls create checkpoints within a router and examine packets as they progress through an interface. If the information does not pass the inspection, it is dropped.

## Advantages

Not resource intensive, meaning they are low-cost and do not have a significant impact on system performance.

Work best with small networks.

## Disadvantages

Easy to subvert compared to more robust firewalls.

Only operate at the network layer.

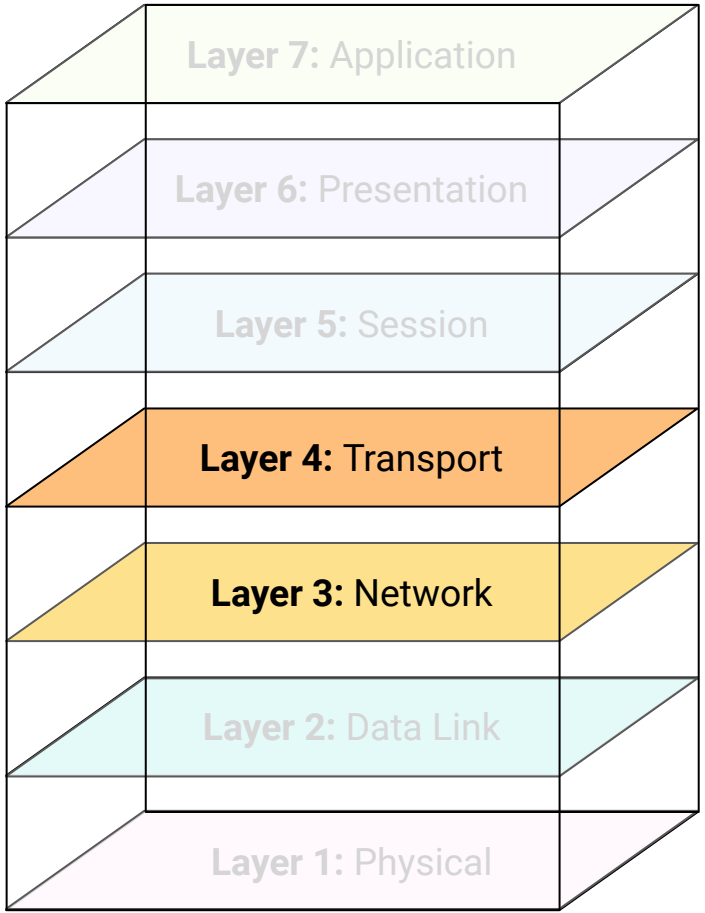
They are vulnerable to spoofing and do not support custom based rule sets.

# Packet-Filtering Firewalls (Stateful)

Stateful packet-filtering firewalls operate at **Layer 3** and **Layer 4** of the OSI model.

**Stateful firewalls can determine if a packet is:**

- Trying to establish a new connection, known as a **new** state.
- Part of an existing connection, known as an **established** state.
- Is neither a new nor an existing connection, known as a **rogue** packet.




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

They inspect the packets' conversation and routing tables and use a combination of TCP handshake verification and packet inspection technology.

# Packet-Filtering Firewalls (Stateful)

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, and therefore can't determine what the underlying traffic is doing. While they can identify that a connection is using HTTP, they cannot identify if the connection is being used to request an HTML file or a PNG image.

## Advantages

Offer transparent mode, which allows direct connections between clients and servers.

## Disadvantages

Are resource-intensive systems.

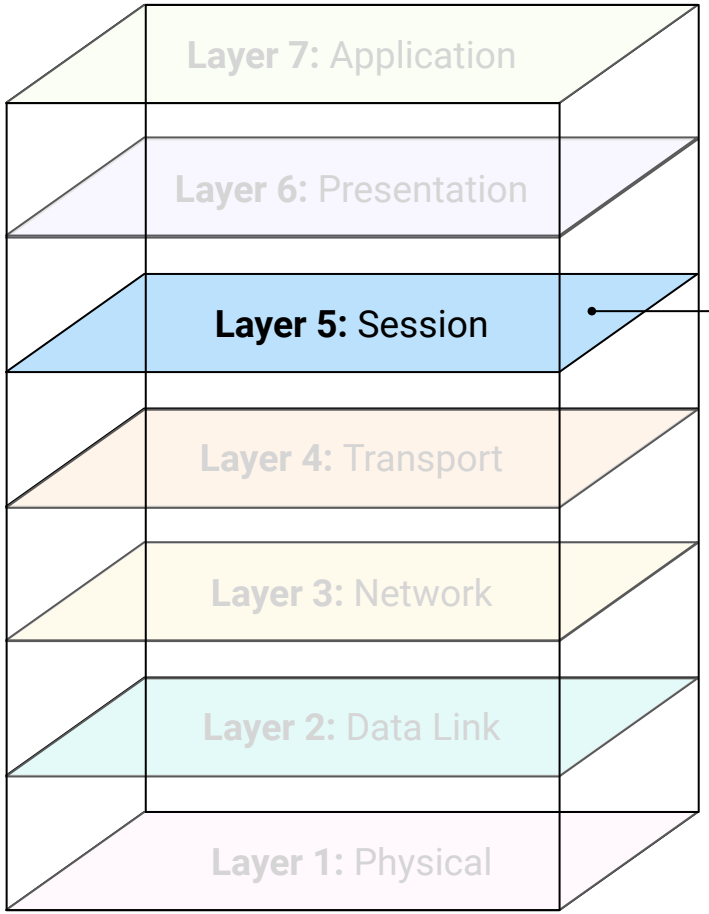
# Circuit-Level Firewalls

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

Circuit-level gateways work by verifying the three-way TCP handshake.

**TCP handshake checks can verify the following about a source:**

- Unique session identifier
- State of the connection (handshake established, closed)
- Sequencing information




These firewalls only look at the header of a packet.

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

# Circuit-Level Firewalls

By verifying the three-way TCP handshake, they ensure that session packets are from legitimate sources.

## Advantages

Quickly and easily approve and deny traffic without consuming significant computing resources.

Relatively inexpensive and provide anonymity to the private network.

## Disadvantages

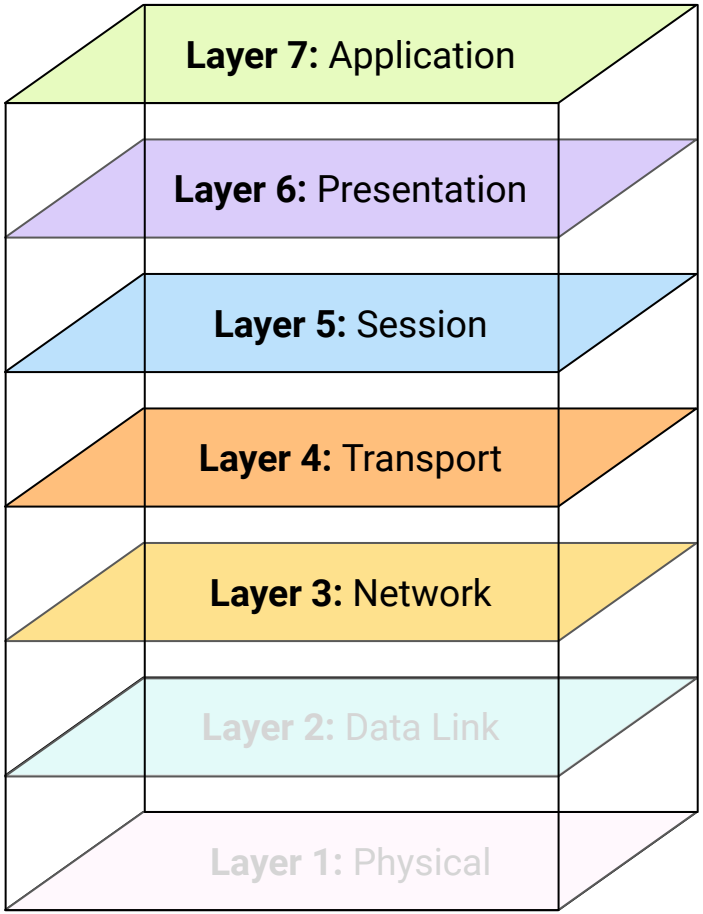
Do not check the contents of the packet.

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

# Application (Proxy) Firewalls

Application, or proxy, firewalls operate at **Layer 3** through **Layer 7** of the OSI model.

They inspect the actual 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.

They 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.

# Application / Proxy Firewalls

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.

## Advantages

More secure than other implementations and provide simple log and file audit management for incoming traffic.

## Disadvantages

Resource intensive. Requires robust modern hardware and high costs. Bypassed with encryption.

# UFW

# Uncomplicated Firewall (UFW)

UFW is a multifunctional firewall that provides stateless and stateful packet filtering.

It works on all kinds of network address and port translation, such as Network Address Translation (NAT) and Network Address Protection (NAP).

It is a standard Linux firewall, available by default in the latest Ubuntu installations.

# Uncomplicated Firewall (UFW)

UFW provides the following features:

| Term | Definition |
|---|---|
| **Host-based** | UFW is most commonly used on hosts. |
| **Logging** | UFW has the ability to generate multi-level logs, providing great insight into attacks. |
| **Remote management** | Firewalls can be remotely managed. For example, through SSH via port 22. (Security concern?) |
| **Rules for allow / deny** | Examines source and destination IP addresses, port numbers, and packet types, all without opening the packet. Also uses TCP handshake, packet inspection. |
| **Rate-limiting** | Supports rate-limited connections to protect from brute force attacks. |

# UFW Demo

In the next demo, we will work with the following scenario:

The IT department is hosting a website that requires the use of both typical 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.

# Instructor Demonstration

## UFW

# Recap

| | |
|---|---|
| `sudo ufw reset` | to reset all UFW rules back to factory defaults. |
| `sudo ufw status` | to check the current status of the firewall. |
| `sudo ufw enable` | to start the firewall and update rules |
| `sudo ufw reload` | to stop, and restart the UFW firewall. |
| `sudo ufw default deny incoming` | to block all incoming connections. |
| `sudo ufw default allow outgoing` | to allow all outgoing connections. |
| `sudo ufw allow` | to open specific ports. |
| `sudo ufw deny` | to close specific ports. |
| `sudo ufw delete` | to delete rules. |
| `sudo ufw disable` | to shut down the firewall. |

# Activity: Configuring UFW

In this activity, you will use UFW to harden your system and ensure that your organization complies with PCI DSS (Payment Card Industry Data Security Standard).

Suggested Time:

20 Minutes

# Time's Up! Let's Review.

# Questions?


Countdown timer

15:00

(with alarm)

# Firewalld

# Introducing Firewalld

While UFW allows us to manage multiple networks devices over the command line, it has a major drawback:

Before firewall rules can be changed or modified, all firewall services must be stopped and restarted.

**This can be extremely disruptive to an organization's operations.**

**For example:**

If you just brought a new host online, you would have to interrupt service on hundreds of other devices before creating a rule for this new host.

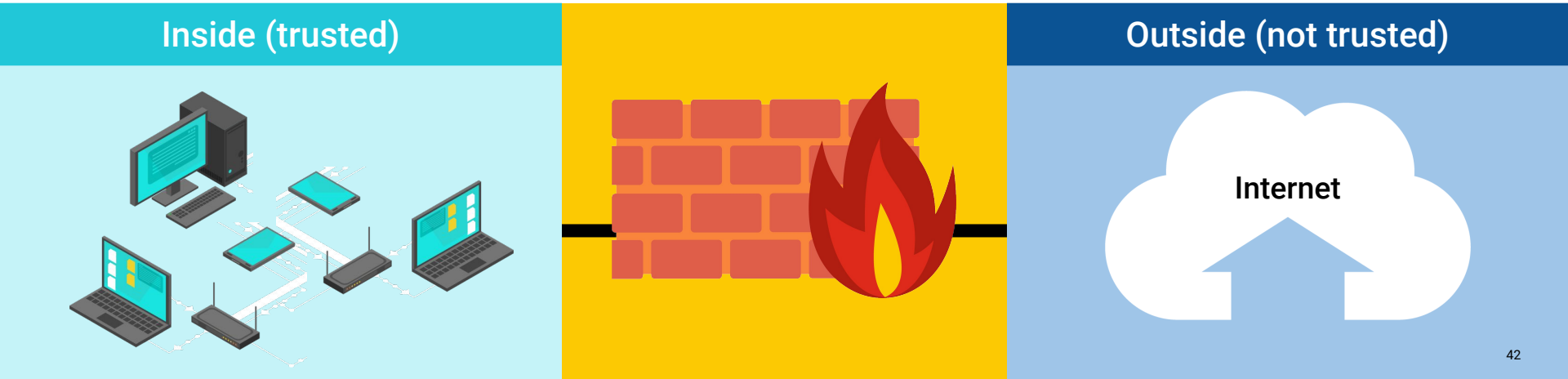
**firewalld**

Provides similar functionality to UFW but does not require the disruption of services when implementing firewall services.

# Introducing Firewalld

firewalld is a dynamically managed firewall that uses **zones** to divide network interfaces into groups of shared trust level. The zones are assigned sets of rules depending on the needs and restrictions of each zone's interfaces.

- Zones are the organization of rules and each zone can contains several rules.
- Through this division of zones, firewalld can manage rulesets **dynamically**, without breaking existing sessions, disrupting services, and bringing down the entire network.



# Testing Rules with Firewalld

Rules and configurations can be tested in runtime environments.

## Runtime configurations

Valid until the next system reboot or service reload. Allows us to create settings that are active for a limited amount of time.

Can be used to test new configurations. Can then be seamlessly saved to permanent environment if deemed good and working.

## Permanent configurations

Loaded with each reboot and reload and become the current runtime environment until new runtime configurations are made.

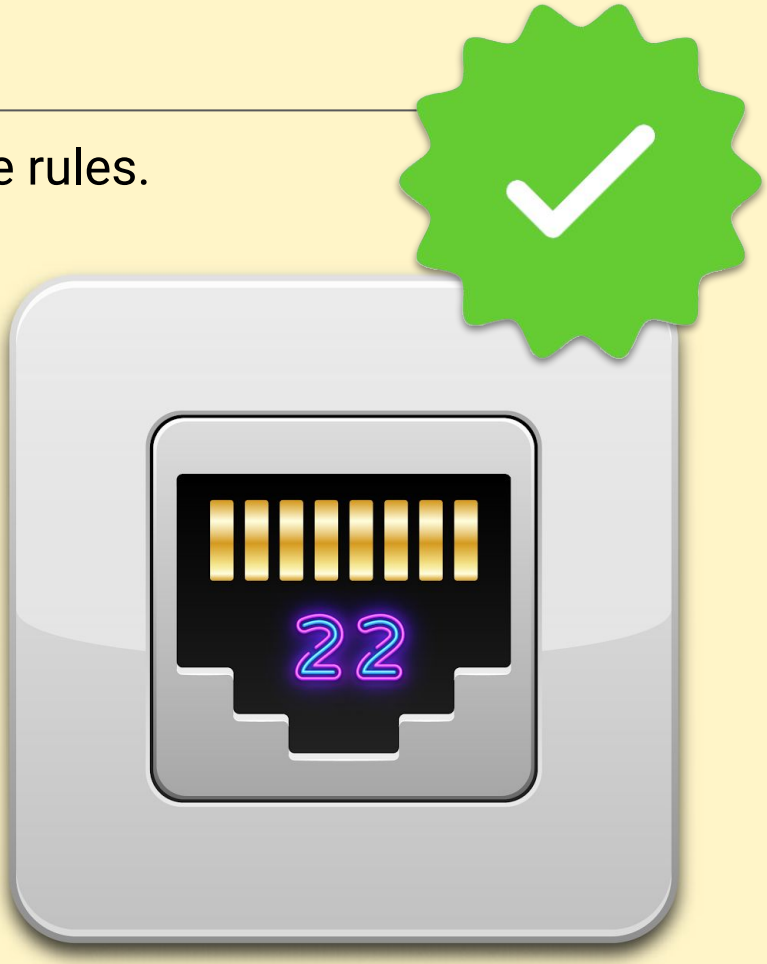
# Firewalld and Services

Firewalld also uses services to easily configure rules.

By designating which services you want to allow, firewalld will automatically open the ports associated to those services.

**For example:**

- If you enable the SSH service in a zone, it will open 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.



# Firewalld Demo

We will demonstrate firewalld using 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 `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`.

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

# Instructor Demonstration

## Firewalld

# Recap

| | |
|---|---|
| `sudo /etc/init.d/firewalld start` | to start firewalld. |
| `sudo firewall-cmd --list-all-zones` | to list all current zones. |
| `sudo firewall-cmd --zone=home --change-interface=eth0` | to bind together interfaces. |
| `sudo firewall-cmd --get-services` | to list currently configured services. |
| `sudo firewall-cmd --zone=home --list-all` | to list all currently configured services for a specific zone. |
| `sudo firewall-cmd --zone=home --add-rich-rule=` | to add specific rules to specific zones. |
| `--add-icmp-block=echo-reply --add-icmp-block=echo-request` | to block pings for specific zones. |

# Activity: Firewalld Configuration

In this activity, you will use firewall to add rules to various zones.

Suggested Time:

20 Minutes

# Time's Up! Let's Review.

# Questions?

# Testing Rules with Nmap

# Nmap

Nmap is the industry-standard network scanner.

- 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.


(nmap)

# Nmap

Attackers can get the following from 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** allows attackers to perform network analysis to determine which Layer 4 protocols a specific firewall allows.

# Nmap Demo

In the following demonstration, we'll perform scans against our UFW firewall. Consider the following scenario:

01

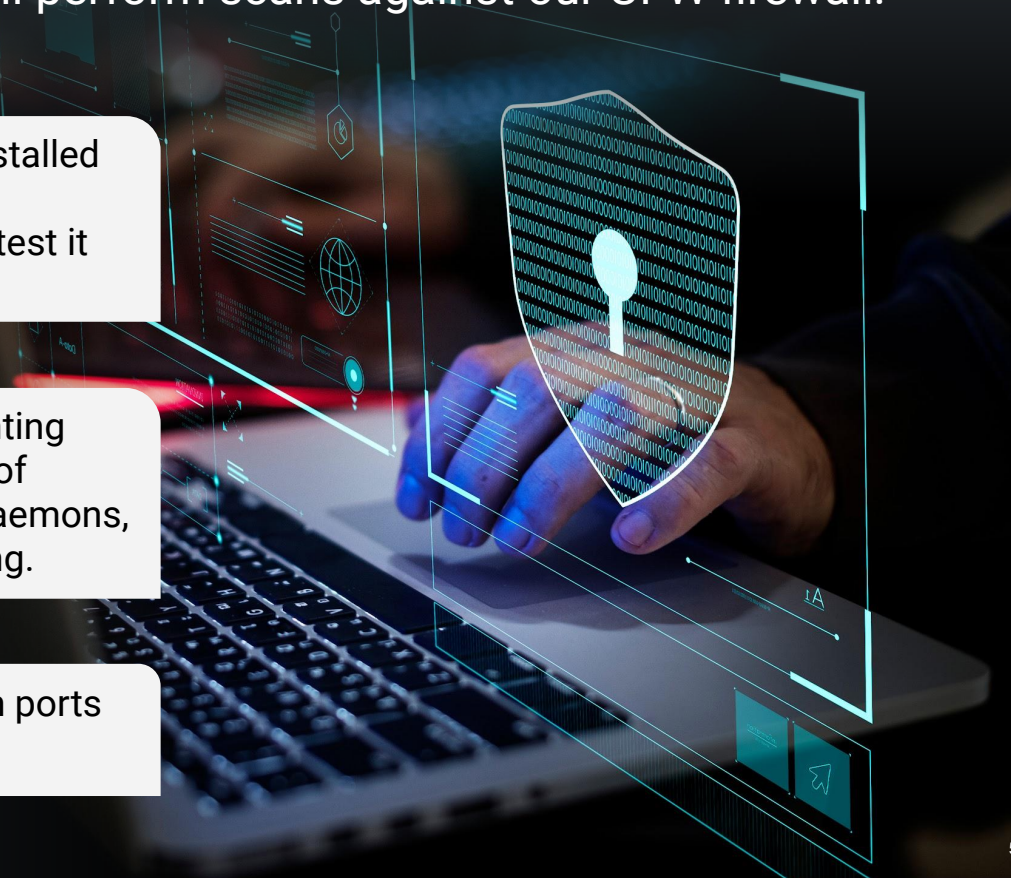
Your security manager has installed a brand new, fully-configured firewall and would like you to test it using Nmap.

02

You will use various fingerprinting techniques to reveal the type of operating system, services, daemons, and protocols currently running.

03

You will also test to see which ports are open, closed, and filtered.



# Instructor Demonstration

## Nmap

# Recap

| | |
|---|---|
| `nmap` | to perform network scans. |
| `nmap -sO` | to perform an IP protocol scan. |
| `nmap -sV` | to enumerate a service type. |
| `nmap -A -T4` | To perform OS fingerprinting using fast execution |
| `uname -a` | To print the OS type and version. |
| `nmap -sA` | To enumerate the type of firewall in use |
| `nmap -sU -F` | To perform a UDP fast scan |

# Activity: Testing Firewall Rules with Nmap

In this activity, you will perform various network scans to test firewall integrity.

Suggested Time:

10 Minutes

# Time's Up! Let's Review.

# Questions?


The End