



Purple Team

FA2025 • 2025-09-16

# Network Security & Active Recon

Ronan Boyarski

ctf.sigpwny.com

**sigpwny{SYN;SYN\_ACK;ACK;}**



# Table of Contents

- Overview of network security
- Networking intro
- Active Recon
  - Port scanning
  - Service-specific recon techniques
  - Edge cases (proxies & UDP services)
- Gaining Access
  - Services & misconfigurations
  - Exploit a known vulnerability (n-day)
    - Exploitdb, searchsploit, GitHub, Metasploit
  - User Enumeration, Password Brute Force & Password Spray
- Live Demo (Infra-dependent)
  - Port scan, service recon, password attack & exploit!

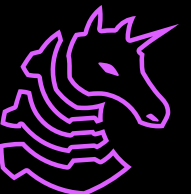


# Infrastructure Update



# Infraaaaahhh

- Servers physically moved to ACM rack (3 floors up!)
- New pwnyos site is: <https://pwnyos.purple.sigpwny.com:443>
- Cyber range longer accessible outside of **IllinoisNet**
  - Quirk of Illinois IP space
  - We may change this later with tunneling
- You should be able to access PwnyOS anywhere

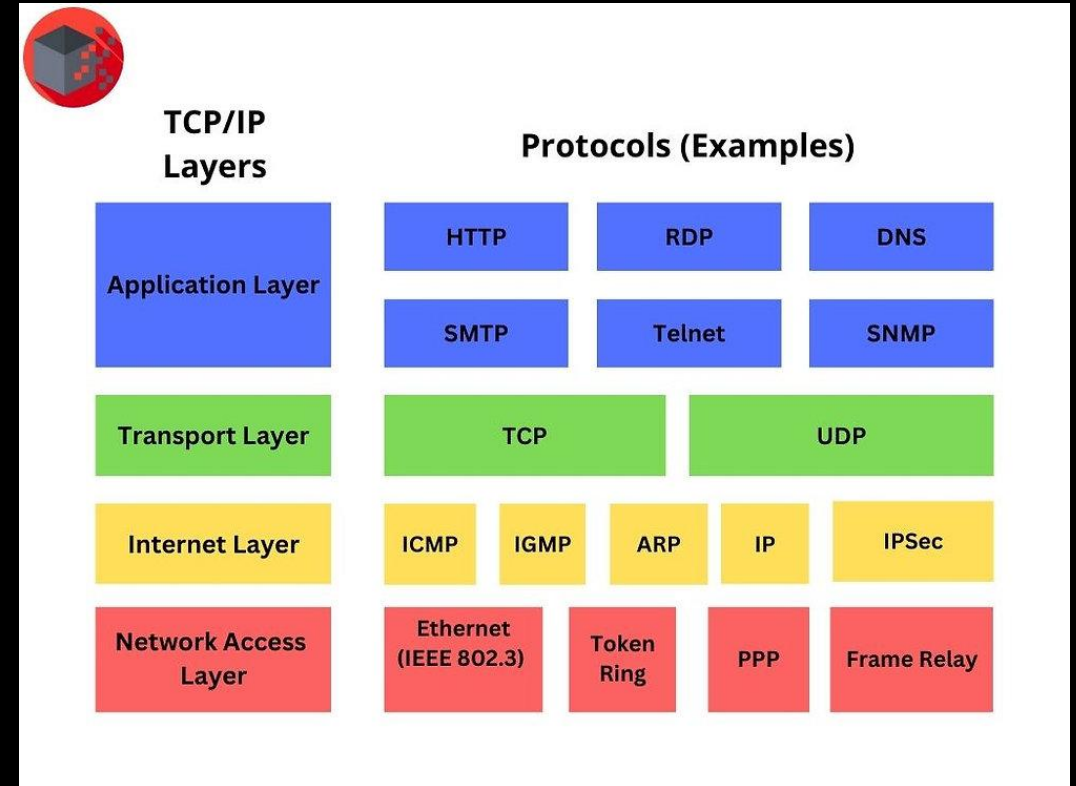


# Network Security Overview



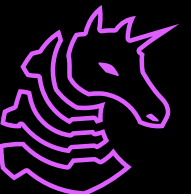
# TCP/IP model

- TCP/IP model offers a very simplified view of networking
- It consists of 4 layers of network, encapsulating one above.
- In purple team, we are only concerned with Application and Transport Layer



# Services

- Services serve content with specific Application protocols
- For example, an HTTP server is a service that serves web content with HTTP protocol
- Network Security concerns the security of **services & trust relationships** that occur in networked environments
- Common exploitable services include HTTP/S servers, SMB, SSH, NFS, \*\*SQL, WinRM, and many more





# How to network 101

- Internet layer offers a way to address machines (IPv4 and IPv6)
- Transport layer offers end-to-end communications (TCP and UDP) between computers, with 65536 different ports each to run different connections on
- Application layer offers client-server communication without worrying about underlying implementation
- To fully understand a service, we must know the address, TCP/UDP and port, as well as application layer protocol
- e.g. <https://sigpwny.com> -> 172.66.x.x, TCP, port 443, HTTPS



# TCP vs UDP

- TCP has a state machine to ensure reliability and speed.
- We will talk about the hand-shake process to initiate a connection, but during the connection it uses sequence number and acknowledgement number to make sure data is received reliably
- UDP is “best-effort”, data reliability is not guaranteed but has very low data overhead.
- Ideal for cases where speed matters over data integrity, like video streaming

Most services you will see are going to be TCP!



# Transmission Control Protocol

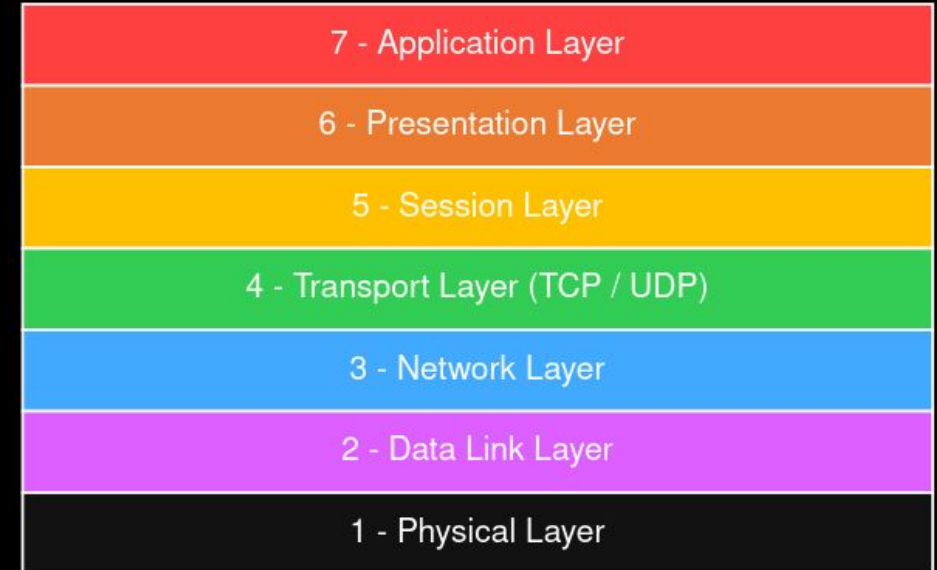


Client

How do these sync?



Server



# Transmission Control Protocol



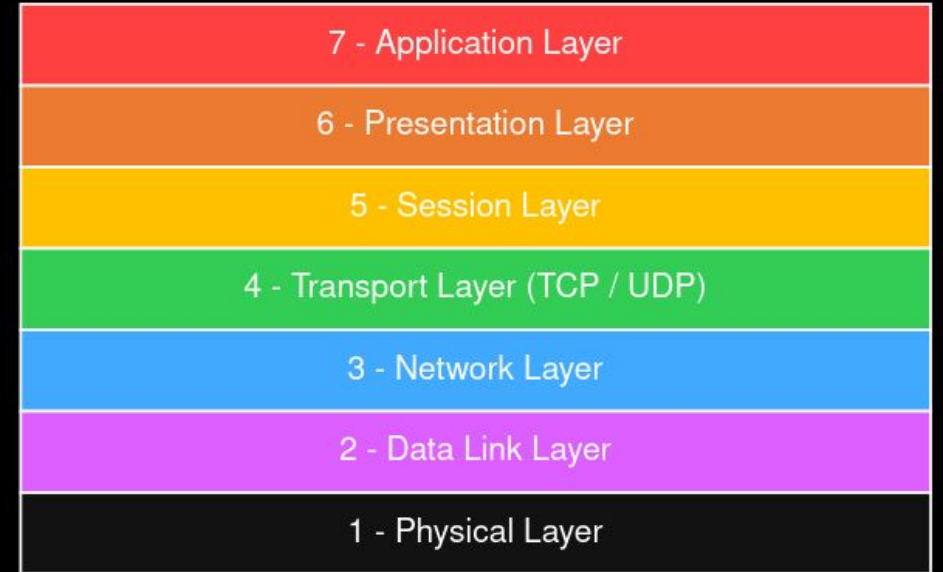
TCP 3-Way Handshake



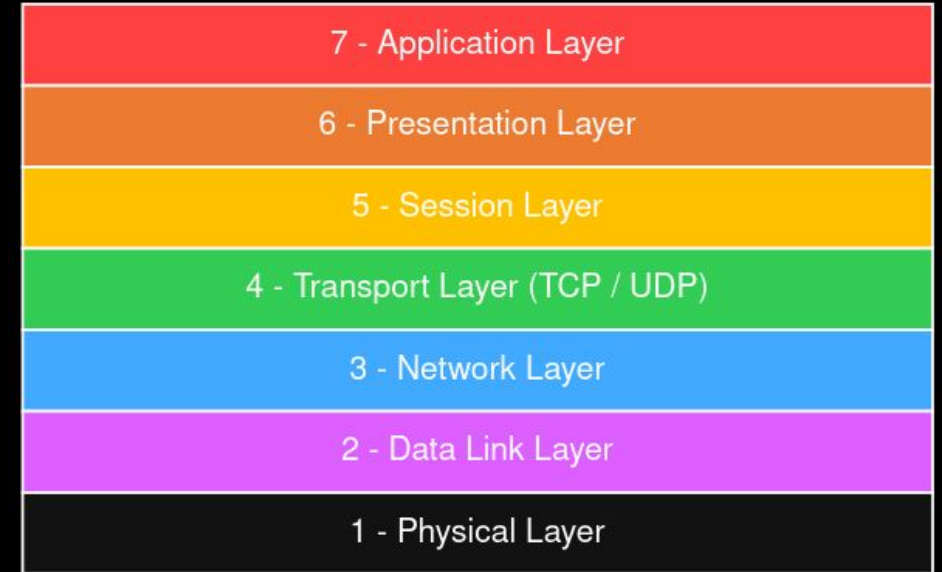
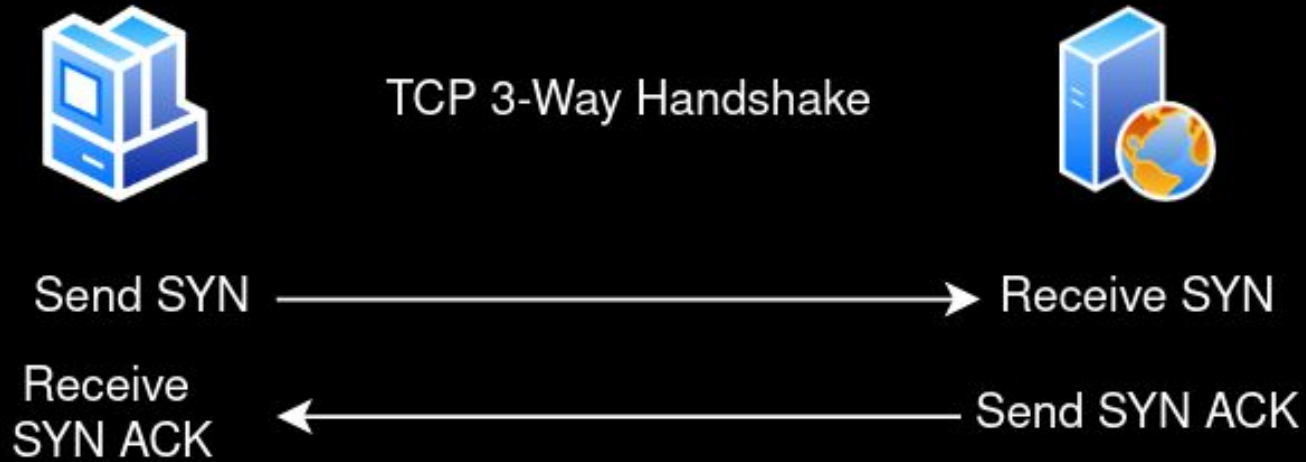
Send SYN



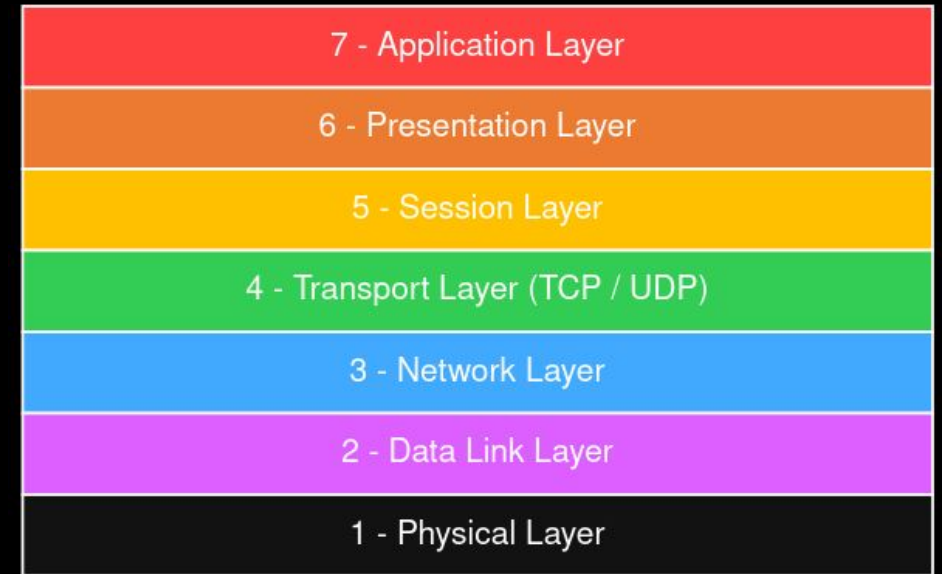
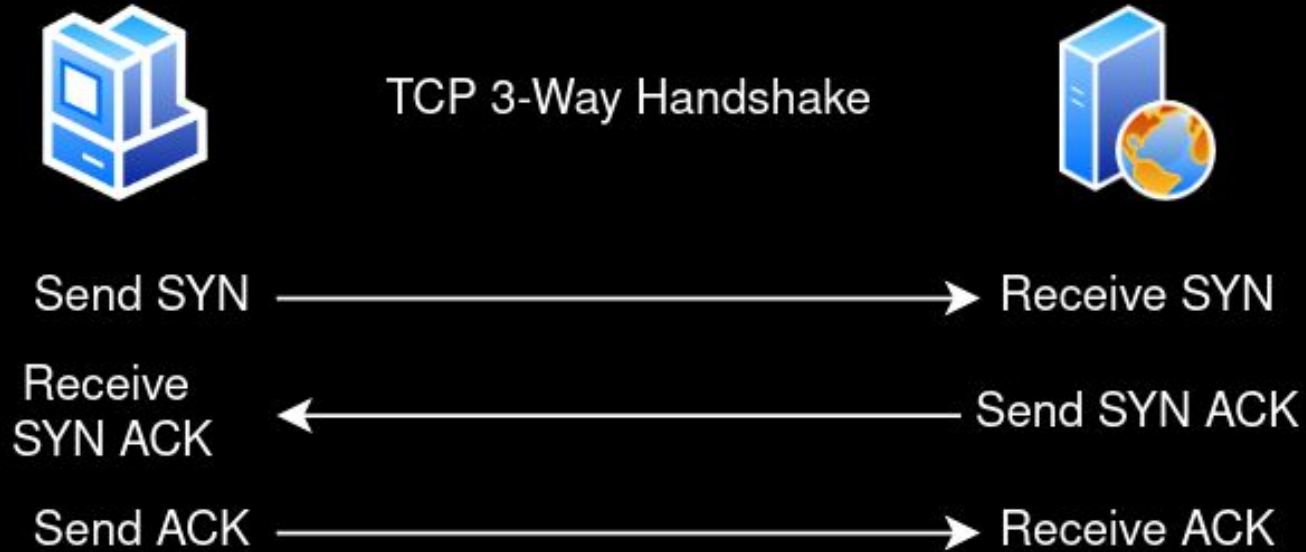
Receive SYN



# Transmission Control Protocol



# Transmission Control Protocol



# Sorry, We're Closed



TCP 3-Way Handshake (Closed)



Send SYN

Receive SYN

Give up

Send RST



TCP 3-Way Handshake (Closed)



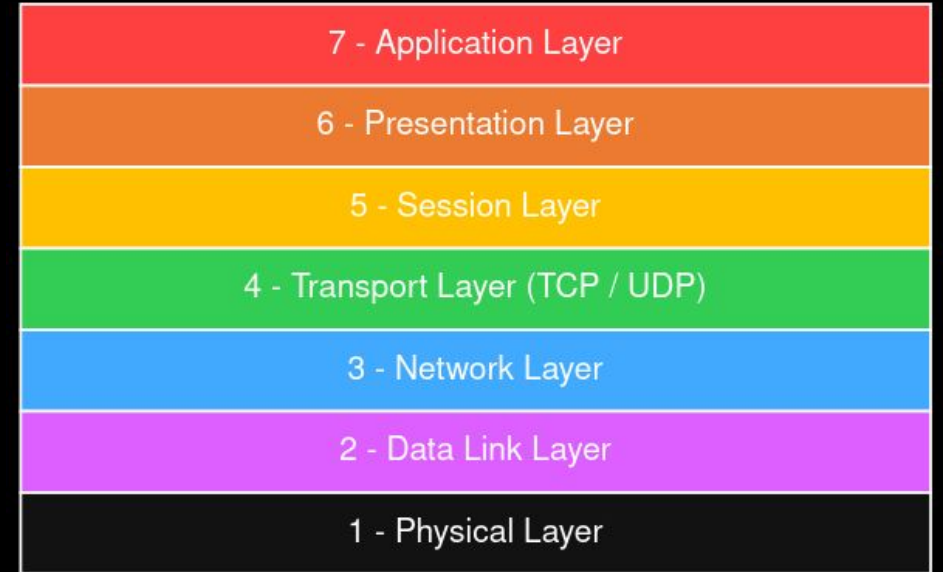
Send SYN

Receive SYN



Give up slowly

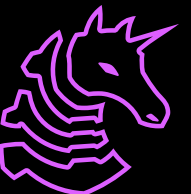
Ghost Client



# Example Services

- FTP (port 21)
- SSH (port 22)
- Telnet (port 23)
- DNS (port 53)
- HTTP (port 80)
- HTTPS (port 443)
- SMB (port 445)
- MSSQL (port 1433)
- NFS (port 2049)
- RDP (port 3389)

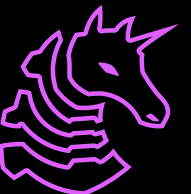
All the ports above are the default ports, assigned by IANA!





# Example Services

- FTP (port 21)
- SSH (port 22)
- Telnet (port 23)
- DNS (port 53)
- HTTP (port 80)
- HTTPS (port 443)
- SMB (port 445)
- MSSQL (port 1433)
- NFS (port 2049)
- RDP (port 3389)
- **Non-Default HTTP (31337)**



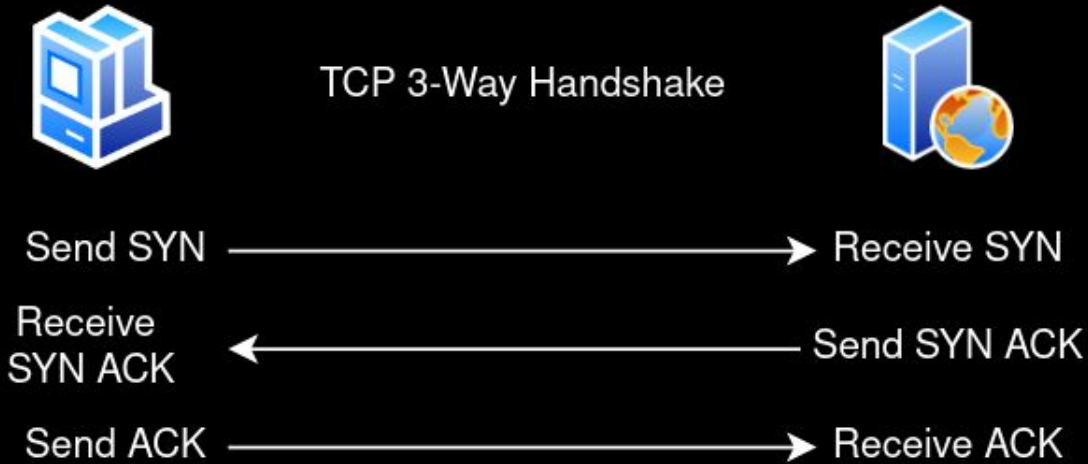
# Example Services

- FTP (port 21)
- SSH (port 22)
- Telnet (port 23)
- DNS (port 53)
- HTTP (port 80)
- HTTPS (port 443)
- SMB (port 445)
- MSSQL (port 1433)
- NFS (port 2049)
- RDP (port 3389)
- **Non-Default HTTP (31337)**

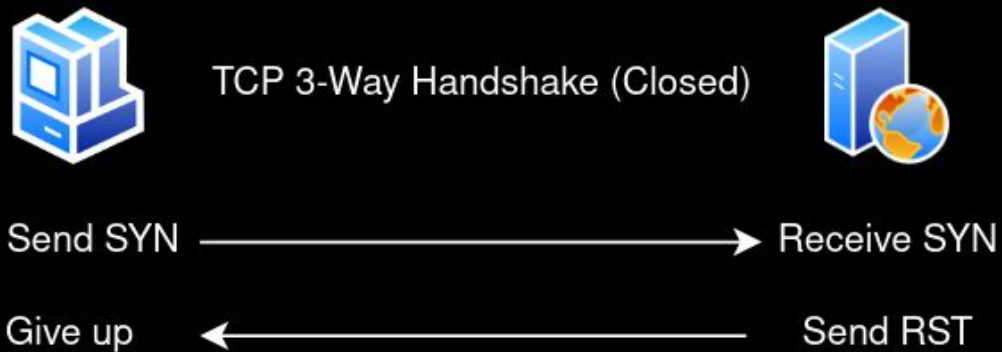
How do we tell **which services** are running on **which ports** for a given IP?



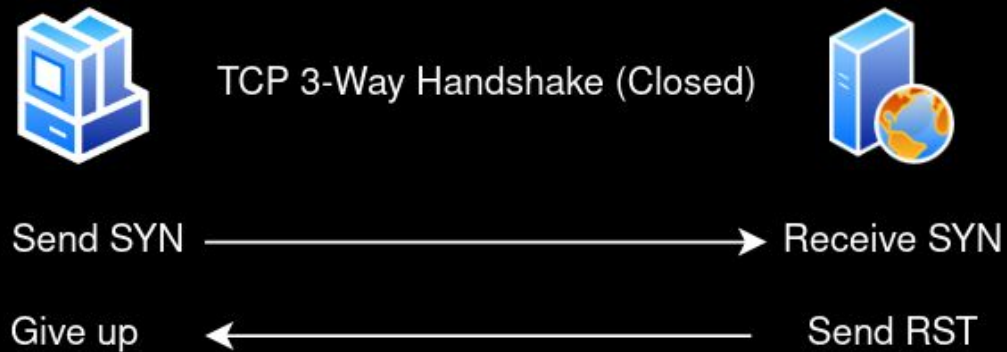
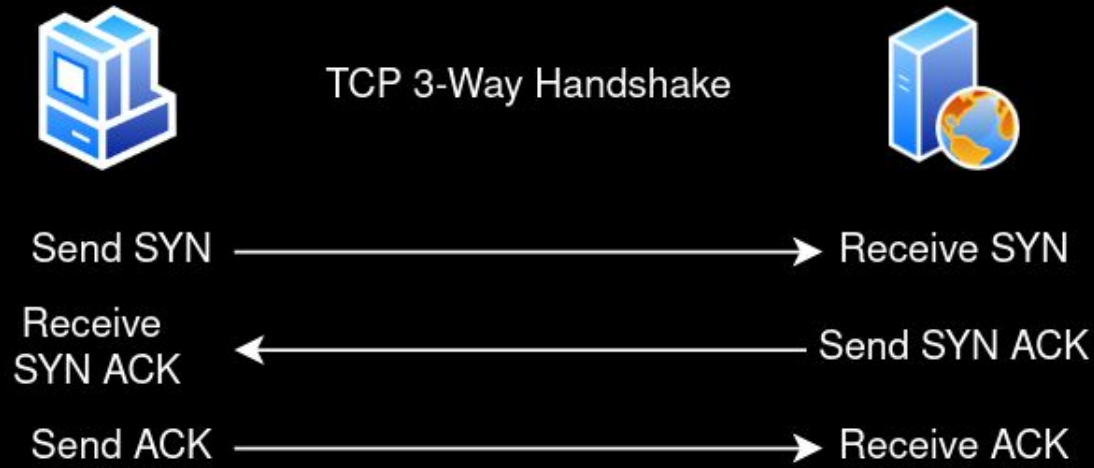
# Transmission Control Protocol



- 65536 TCP ports
- What if we do this handshake on port 0, 1, 2, up to 65535?



# Transmission Control Protocol

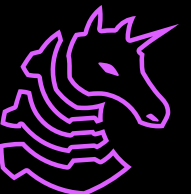


- 65536 total TCP ports
- What if we do this handshake on port 0, 1, 2, up to 65535?
- **Easy to detect**, but we can randomize the order
- SYN scan: just send the SYN and see if we get anything back (bypasses firewalls!)



# Port Scanning

- This process is **port scanning**, the most important active recon technique
- Port scanning tells us which **ports** are open for a given **IP**
- Once we know this, we can then check each port to see what **protocol** they are talking
- Once we know that, we can check them by protocol to see what **service** they are offering
- Optionally, we can check each **service** by cross-referencing its version with a large database of all known vulnerabilities



# Disclaimer: Passive Recon

- When engaging a real target, there is a lengthy **passive recon** phase before this
- You would scope out owned IPs, domains, employees, and tech stack
- Advanced adversaries would also do their covert infrastructure and malware preparation here
- None of these are relevant for this point in the year



# Active Recon



# Port Scanning

- Port range: 0-65535, TCP & UDP
- `sudo nmap -Pn -F -sV -vv $IP -oN fast.txt`
- `sudo nmap -Pn -A -sV -p- -vv $IP -oN full.txt`
- `-Pn` skips the ping check (Windows does not respond)
- `-A` means that nmap will run scripts and OS fingerprinting (Aggressive)
- `-sV` will have the scan perform version checking
- `-p-` will scan every single port from 1-65535
- `-vv` will enable very verbose output
- `-oN` saves the result to a text file so you don't re-scan





# Port Scanning - Edge Cases

- Don't forget UDP services like SNMP!
- `sudo nmap -Pn -F -sU -vv $IP -oN udp.txt`
- `-sU` will have the scan check UDP ports
- `-F` will scan top 1000 ports (UDP scanning is **SLOW**)
- If you're scanning through a SOCKS proxy, you can only scan TCP ports, and should use the `-sT` flag
  - This does a TCP scan rather than SYN scan
- If you're in a network, do a very fast scan using IP range
- General workflow tip: make a directory for each target



# Port Scanning Alternative - Rustscan

- rustscan is a modern, insanely fast alternative to **nmap**
- Can scan all 65535 TCP ports in **as fast as 3 seconds**
- **Not stealthy AT ALL**, does not bypass firewalls
- Great for situations where the only thing that matters is speed
- Integrates with nmap for service scanning and script execution
- Generally fewer features

We recommend rustscan for practice like hackthebox!



# Service Scanning: SMB

- Server Message Block runs by default on all Windows computers
- If you know the password, you can view remote file shares
- If the target is running Windows Server or is AD joined, and you have Administrator credentials, **remote code execution is a feature**
- Windows computers prior to Windows 7 SP 6.1 are vulnerable to MS17-010 (**SYSTEM** Remote Code Execution)
- Depending on the target configuration, you can potentially read/write files



# Service Scanning: Other services

- FTP: can be used to upload files or download sensitive files if left unsecured
  - This is especially potent if chained with a web server w/LFI vuln
- SSH: if you have a password or key, you can login and get a shell
- Telnet: like SSH, but without the secure part (yikes)
- SNMP: Simple Network Management Protocol, allows viewing all of the running processes, usernames, and software versions, including command-line arguments (UDP port 161)
- SMTP: Simple Mail Transfer Protocol, runs email server
- MSSQL: Microsoft SQL server, can sometimes **run commands**
- Redis: Database, can **gain RCE as a feature**



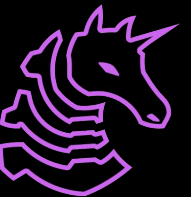
# Service Scanning

- You won't know every service
- Get in the flow of understanding unfamiliar services quickly and think in terms of primitives (what does the service let me do)
- <https://book.hacktricks.xyz/> has some good preliminary steps for interacting with and attacking unfamiliar network services
- Other really important or common services (like web servers & active directory) will be covered individually
- It is very common to see new and unfamiliar services when attacking a network

When you don't know something, Google it!



# Gaining Access



# Gaining Access: Exploitation

- Sometimes, when attacking vulnerable software, it's as easy as running **searchsploit** or the relevant metasploit module
- Other times, custom exploit development is necessary
  - This is where time spend doing traditional CTF is helpful
- Example workflow:
  - nmap -> port 80 is open -> feroxbuster -> find gitlab instance
  - searchsploit gitlab
  - run exploit, hopefully get shell
- **ALWAYS** read exploit code before running it!



# Gaining Access: Misconfiguration

- Sometimes, services are set up with really stupid permissions
- For example, an file server that lets you write to anything in a web server or a user's home directory would be a huge problem
- Example Workflow:
  - nmap -> port 21 & 80 -> unauthenticated FTP server with access to /var/www/html -> put webshell -> browse to port 80 -> get shell
- There are way too many possible misconfigurations to cover here
- Get in the habit of thinking about what access is appropriate



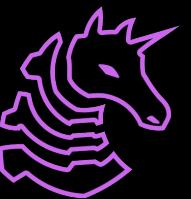


# Gaining Access: Password Attack

- Lots of common software, like WordPress, doesn't rate-limit authentication, so you can go through an obscene amount of login attempts
- **Hydra** is a fantastic general-purpose password attack tool
- Example workflow:
  - nmap -> port 443 -> feroxbuster -> /wp-admin
  - hydra -l Admin -P /usr/share/wordlists/rockyou.txt 10.10.230.209 http-post-form "/wp-login.php:log=^USER^&pwd=^PWD^:The password you entered for the username" -t 30
- Use admin login to upload PHP reverse shell (feature)
- Hydra can be used to attack many other services as well

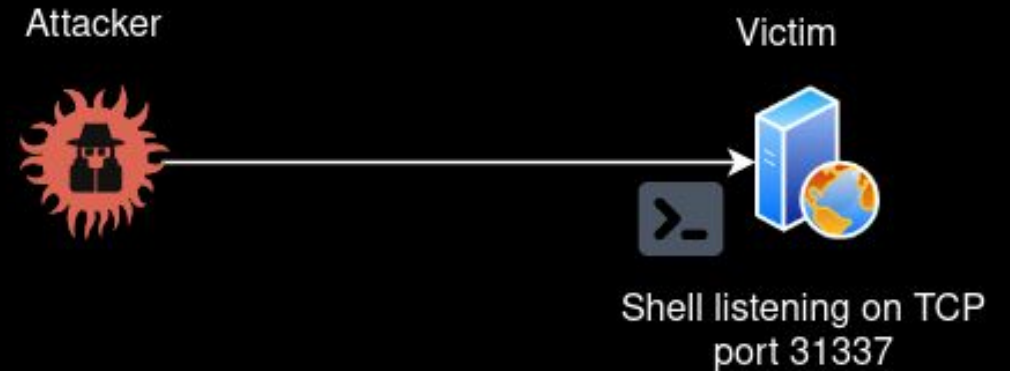


# Welcome to Shell



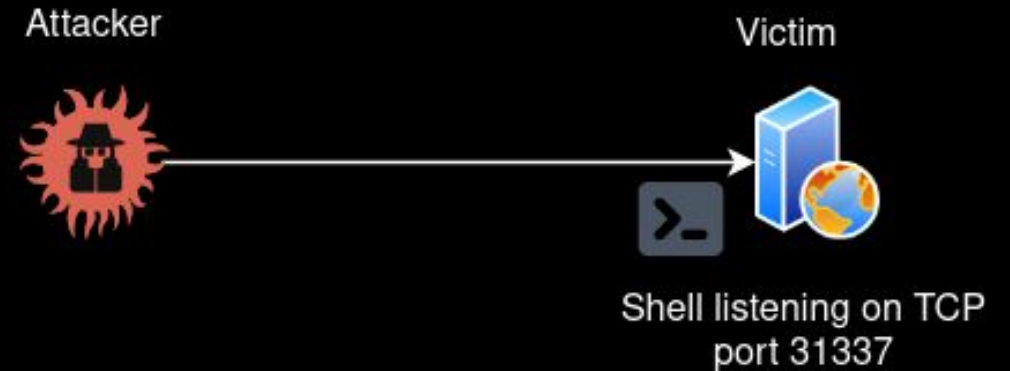
# Bind Shells

- Run a command or program to run a shell as a service
  - Binds to a port on the victim
- Connect **forward** into the shell
- Generally uncommon. Why?



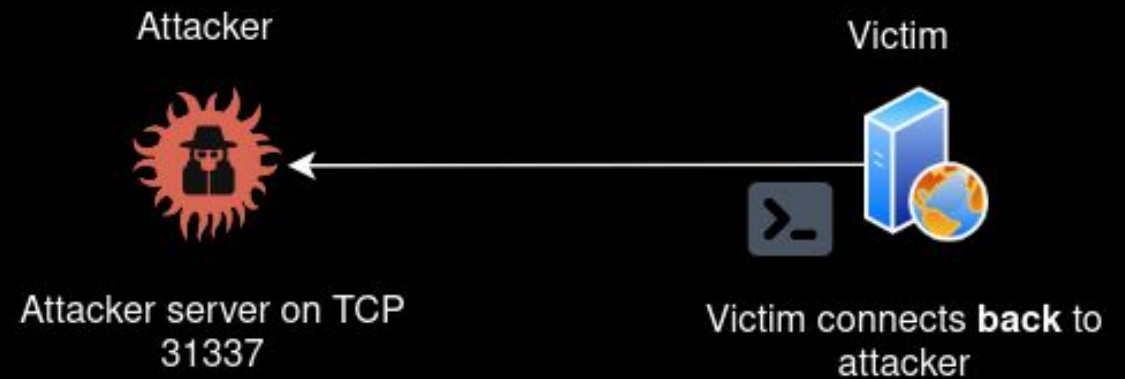
# Bind Shells

- Run a command or program to run a shell as a service
  - Binds to a port on the victim
- Connect **forward** into the shell
- Generally uncommon due to **poor security and stealth**
- Anyone can connect to this!



# Reverse Shells

- Shell connects **back** to an attacker server
  - In this case, the attacker runs the "service" to accept the connection
- Harder to detect (most software clients generate EGRESS TCP traffic)
- More secure - only the attacker gets access

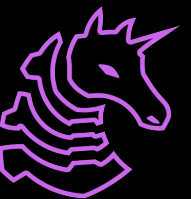


# Food for thought

- What happens here?
- The answer will be revealed in about a month...



# Assembling the Pieces



# Network Recon & Attacks

- Begin with a quick sweep of all in-scope IPs to see which ones you can reach
- Continue to port scan each of them, preferably including a version scan and maybe even vulnerability scan
- Recon each service with further tools
  - SMB? NetExec, enum4linux, smbclient
  - FTP? FTPclient
- Figure out any weaknesses (insecure credentials) or vulnerabilities
- **Recon phase ends here**
- Exploit the vulnerability
- Control the computer via a bind or reverse shell





# Hypothetical

- Attacker needs to discover vulnerable devices on this network to attack them

Attacker



Target: 192.168.10.0/24



# Hypothetical

- The attacker will first discover computers on the network with a quick scan
- No need to scan all ports when most computers will not be up

Attacker



Target: 192.168.10.0/24

```
sudo nmap -p22,80,135,139,445  
192.168.10.0/24
```



192.168.10.150
135
139
445



192.168.10.151
135
139
445



192.168.10.152
22

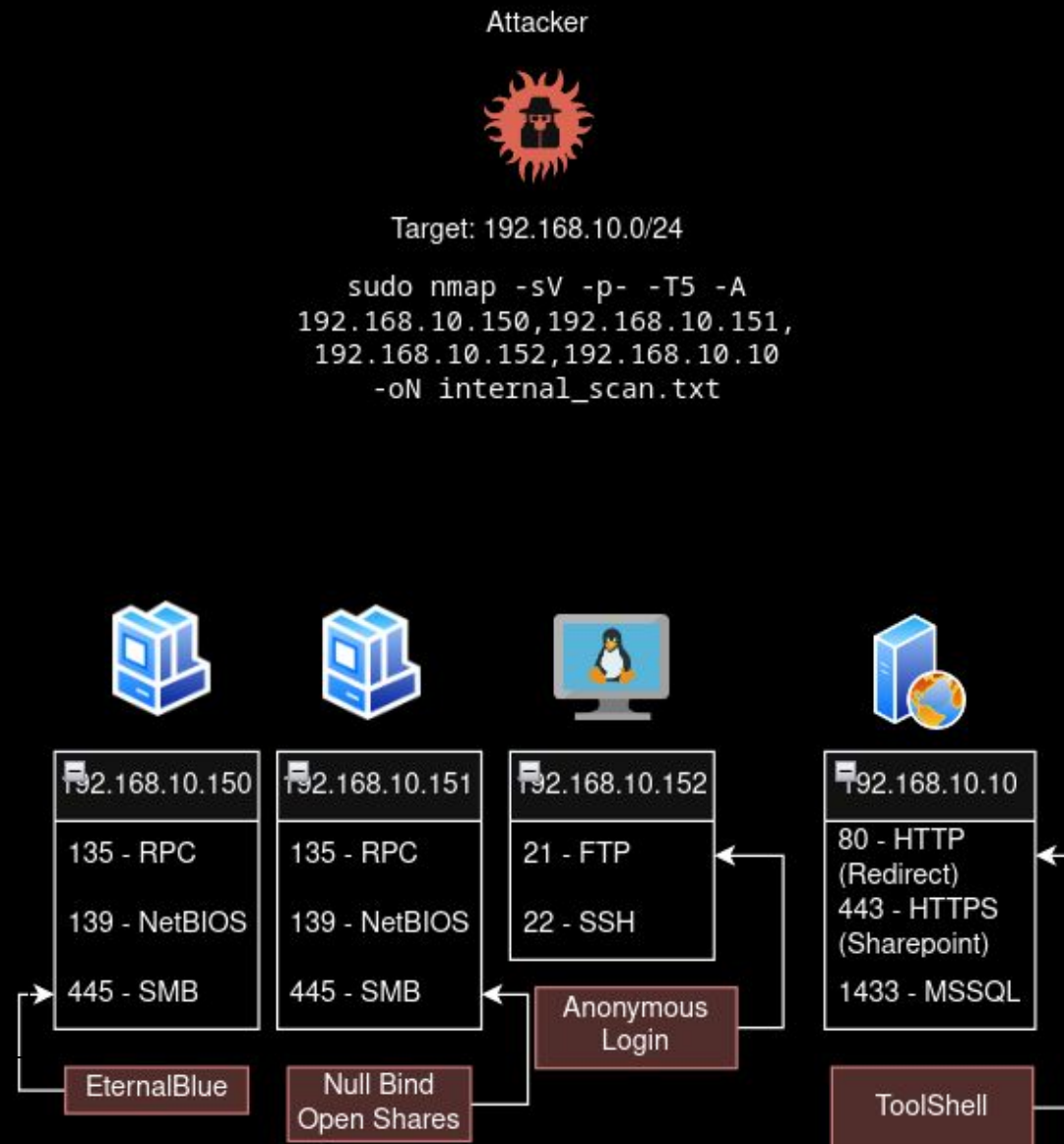


192.168.10.10
80



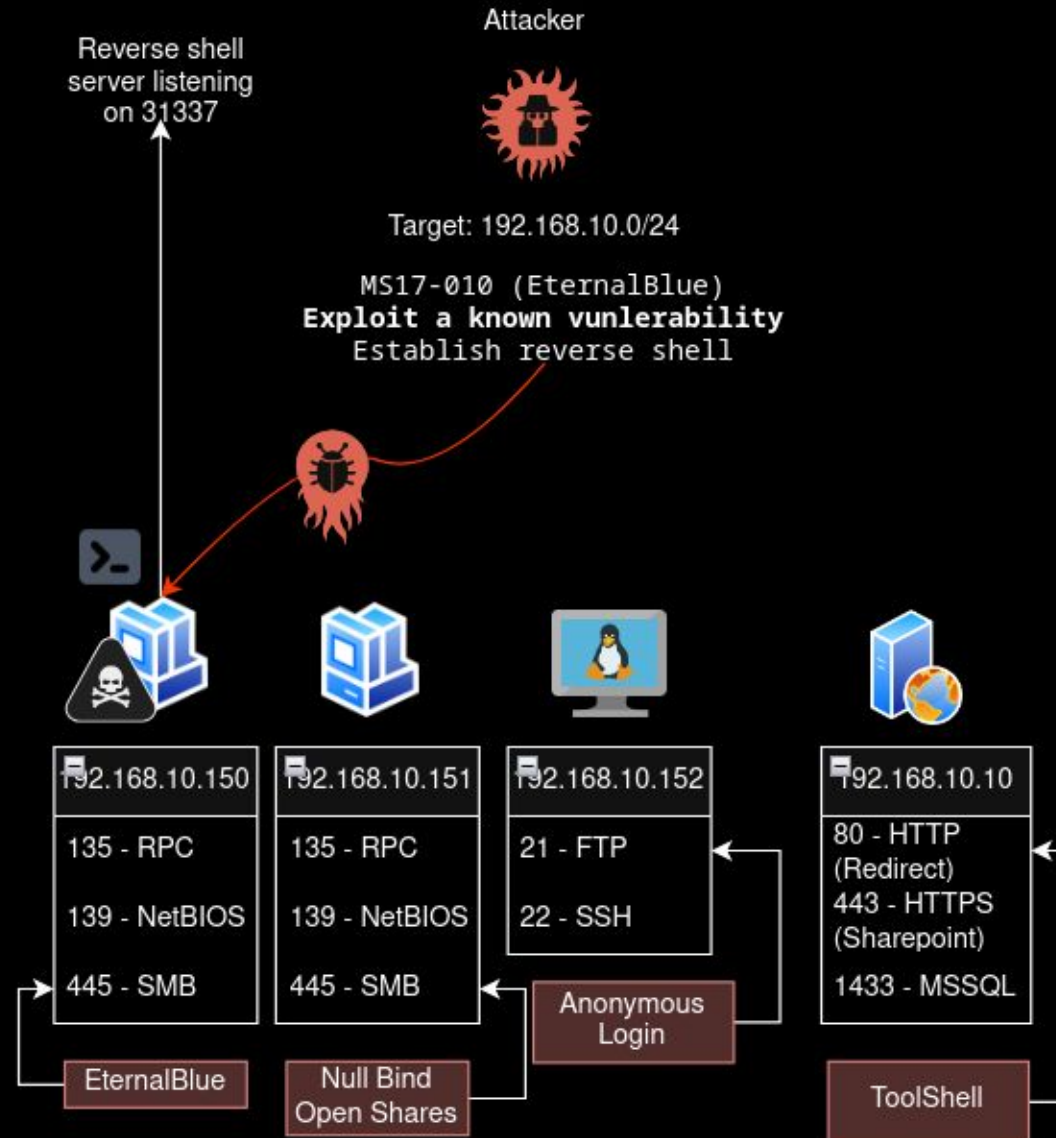
# Hypothetical

- The attacker will now map out specific services, hunting for information and vulnerabilities



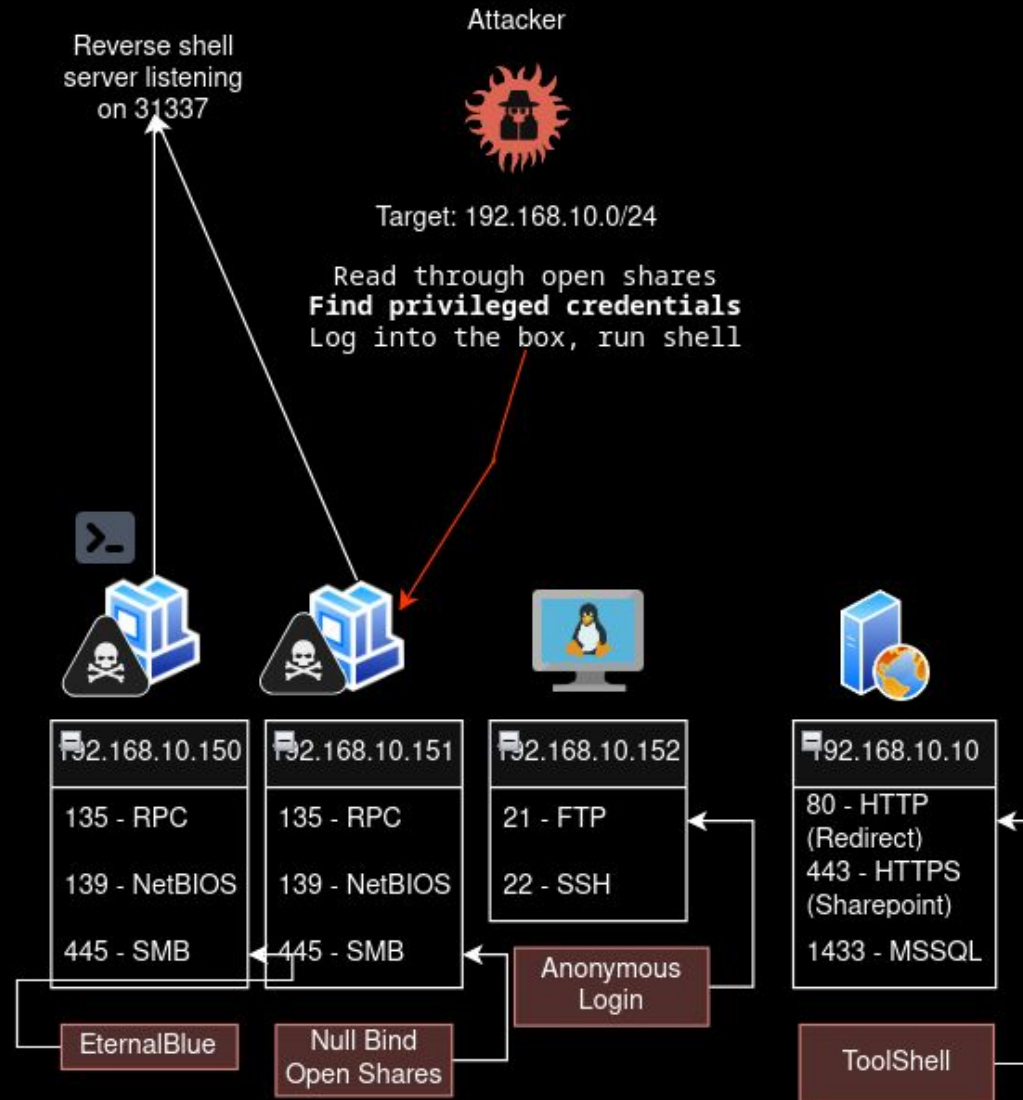
# Hypothetical

- Exploiting known vulnerabilities will often net an easy compromise
- A common method is to run a command to gain a reverse shell
- This is the simplest and most ubiquitous type of compromise



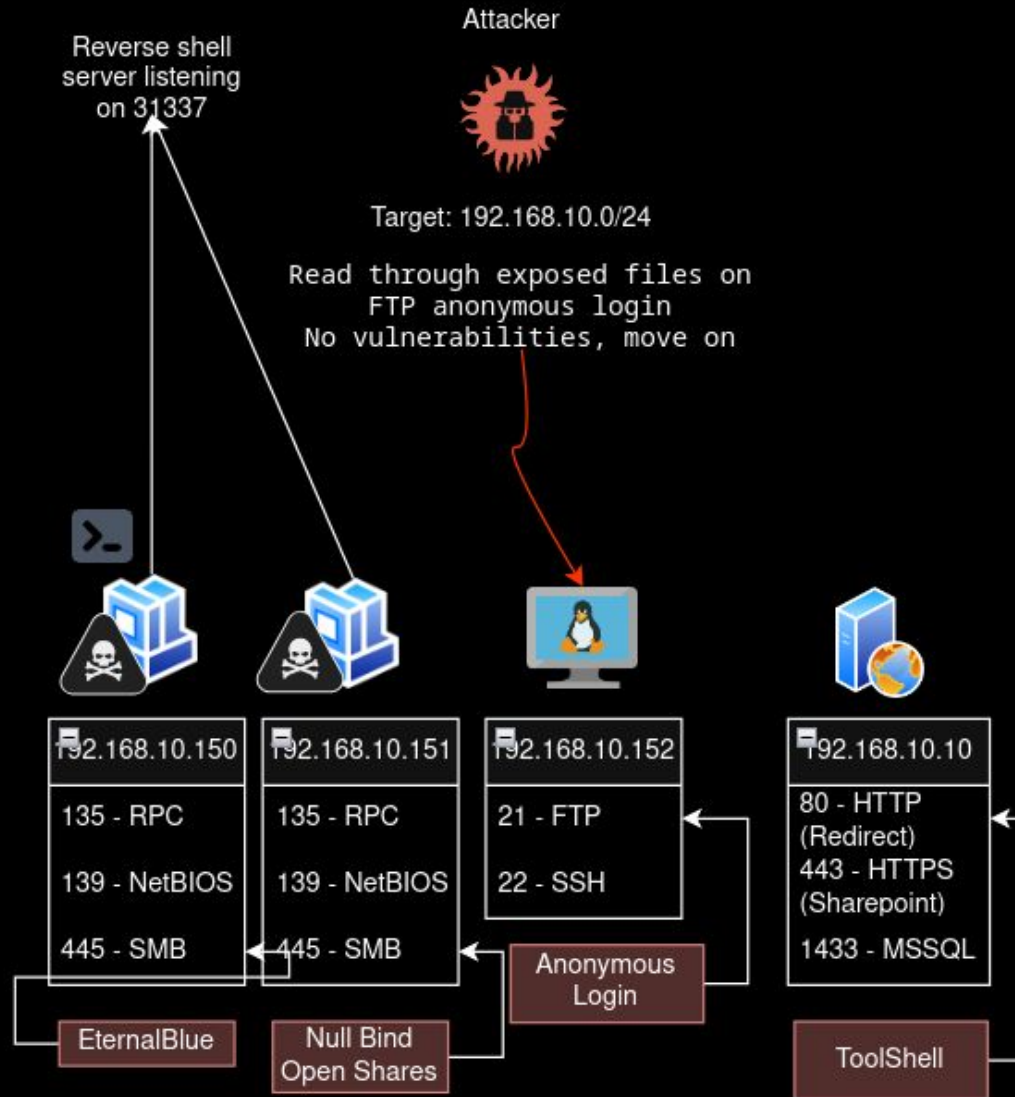
# Hypothetical

- Sometimes, we will find **valuable information** exposed
- What may not be a vulnerability in theory can lead to compromise in practice
- Especially if an admin leaves their password in the open...



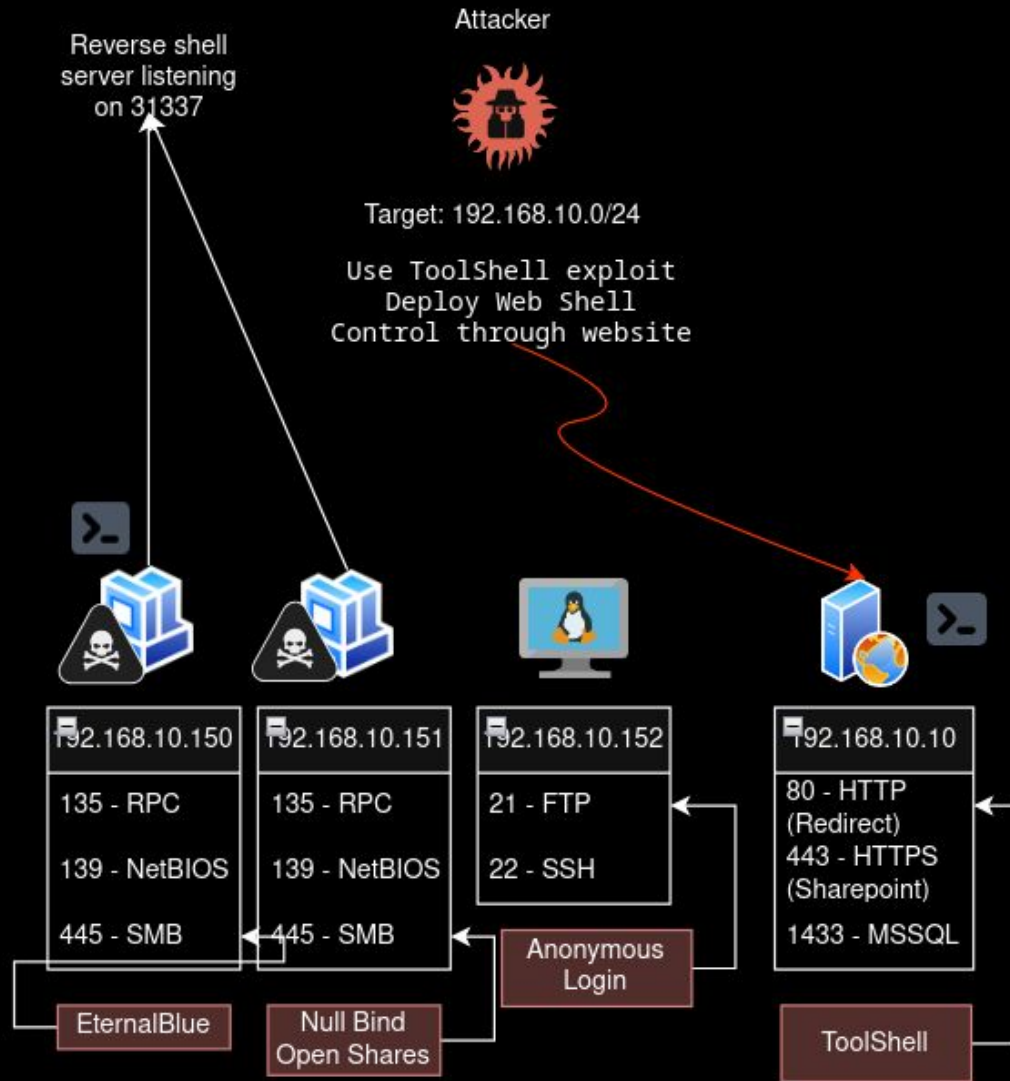
# Hypothetical

- Not all boxes will be vulnerable!
- Many times, you will need to compromise one to get to another
- "Six degrees of separation"



# Hypothetical

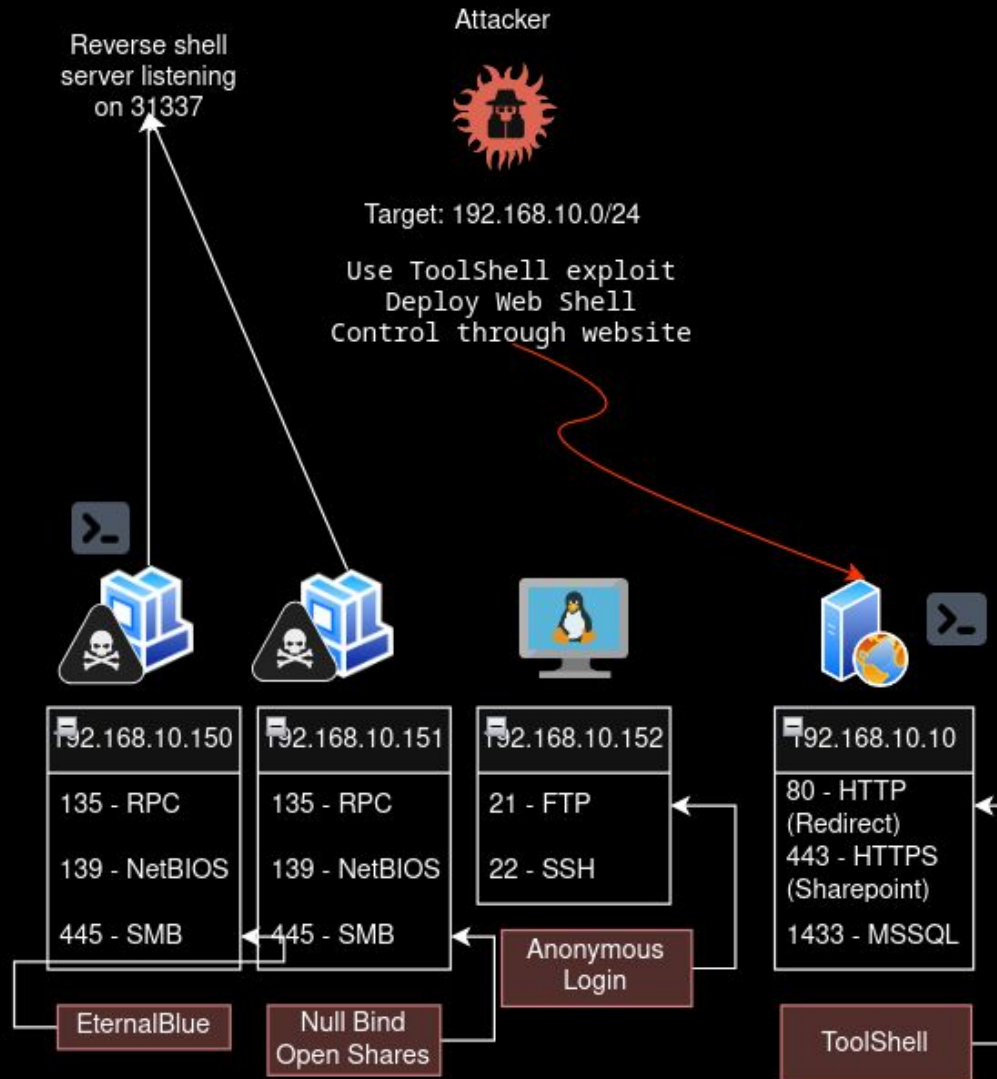
- Deploy a web shell
  - Special bind shell
  - Lives on a website
  - Access it by visiting the site





# Hypothetical

- Deploy a web shell
  - Special bind shell
  - Lives on a website
  - Access it by visiting the site
- Now, we own 75% of the network!
- Realism depends on environments, open shares are common, EternalBlue is not





# Next Meetings

## 2025-09-18 • This Thursday

- Wireshark & Detecting Lateral Movement
- We'll go over the basics of defensive network security

## 2025-09-23 • Next Tuesday

- Practical Web Hacking
- Learn some different web hacking techniques that we see in the wild!



ctf.sigpwny.com

sigpwny{SYN;SYN\_ACK;ACK;}

Meeting content can be found at  
[sigpwny.com/meetings](https://sigpwny.com/meetings).

