Understanding the penetration

penetration often refers to penetration testing, which is a method used to evaluate the security of a computer system, network, or web application by simulating an attack from a malicious actor. The goal is to identify vulnerabilities that could be exploited by attackers.

Objective:

The primary objective of penetration testing is to assess the security posture of a system and identify potential weaknesses before malicious hackers can exploit them.

Process:

Penetration testing involves a controlled and authorized attempt to exploit vulnerabilities in a system. This is typically done by a skilled and ethical cybersecurity professional or a team. The process often includes information gathering, vulnerability analysis, exploitation, post-exploitation analysis, and reporting.

Types of Penetration Testing:

- **Black Box Testing: The tester has no prior knowledge of the system being tested.**
- **White Box Testing: The tester has full knowledge of the system, including architecture and source code.**
- **Gray Box Testing: The tester has partial knowledge of the system, simulating the perspective of an insider or a trusted use**

Common Techniques:

- ➢ Scanning and Enumeration: Identifying active hosts and services on a network.
- ➢ Vulnerability Assessment: Identifying and assessing vulnerabilities in systems and applications.
- ➢ Exploitation: Attempting to exploit identified vulnerabilities to gain unauthorized access.
- ➢ Post-Exploitation: Assessing what an attacker could do after gaining access and identifying potential impacts

. **Reporting**:

A detailed report is generated at the end of the penetration testing process, outlining the vulnerabilities discovered, the methods used to exploit them, and recommendations for improving security.

Benefits:

Helps organizations understand their security weaknesses.

Provides insights into potential real-world attack scenarios.

Aids in prioritizing and remedying identified vulnerabilities.

CVE (Common Vulnerabilities and Exposures) and CWE (Common Weakness Enumeration) are two key resources in the field of cybersecurity that help in identifying and classifying vulnerabilities and weaknesses in software and systems.

CVE (Common Vulnerabilities and Exposures):

Definition:

CVE is a dictionary or catalog of known vulnerabilities in software and hardware. Each vulnerability is assigned a unique identifier, known as a CVE ID.

Purpose:

The primary goal of CVE is to provide a standardized naming convention for vulnerabilities, making it easier to share data across different vulnerability databases and security tools.

Format:

A CVE ID is typically in the format "CVE-YYYY-NNNN," where YYYY is the year the CVE ID was assigned, and NNNN is a unique identifier.

Example:

CVE-2022-12345

Usage:

Security professionals, vendors, and organizations use CVE IDs to reference and discuss specific vulnerabilities. It simplifies communication and coordination when addressing security issues.

CWE (Common Weakness Enumeration):

Definition:

CWE is a community-driven project that provides a standardized taxonomy for describing common software security weaknesses in a vendor-neutral manner.

Purpose:

CWE aims to provide a common language for discussing, identifying, and dealing with software security weaknesses across different tools and organizations.

Format:

CWE entries are identified by a CWE ID, which is in the format "CWE-XXX," where XXX is a unique identifier.

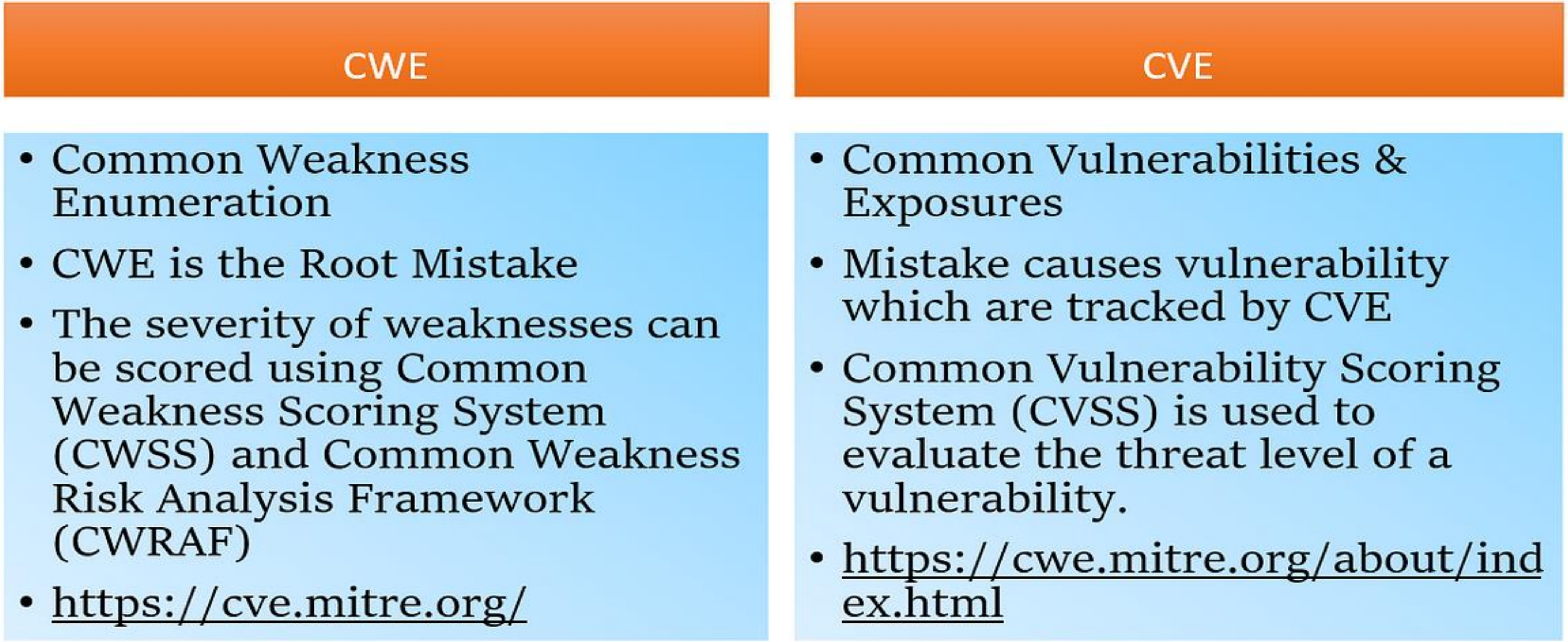
Example:

CWE-119 (Improper Restriction of Operations within the Bounds of a Memory Buffer)

Usage:

Developers, security analysts, and researchers use CWE to categorize and describe the different types of vulnerabilities and weaknesses that can be present in software.

Relationship between CVE and CWE



CWE

- Common Weakness Enumeration
- CWE is the Root Mistake
- The severity of weaknesses can be scored using Common Weakness Scoring System (CWSS) and Common Weakness Risk Analysis Framework (CWRAF)
- https://cve.mitre.org/

CVE

- Common Vulnerabilities & Exposures
- Mistake causes vulnerability which are tracked by CVE
- Common Vulnerability Scoring System (CVSS) is used to evaluate the threat level of a vulnerability.
- https://cwe.mitre.org/about/index.html

free hands-on labs for penetration testing, there are several platforms that offer practical exercises and challenges to help you enhance your skills. Here are some recommendations:

Hack The Box (HTB):

Website: Hack The Box

Description: HTB is a popular platform that provides a variety of virtual machines with different difficulty levels. Users can exploit these machines to practice penetration testing skills. It has an active community and offers both free and premium content.

TryHackMe:

Website: TryHackMe

Description: TryHackMe is an online platform that offers a wide range of virtual environments and challenges for learning and practicing cybersecurity skills. It has free rooms that cover different topics, including penetration testing and ethical hacking.

OverTheWire:

Website: OverTheWire

Description: OverTheWire provides various war games and challenges that focus on different aspects of cybersecurity, including penetration testing. The games are designed for different skill levels, from beginners to advanced users.

PentesterLab:

Website: PentesterLab

Description: PentesterLab offers hands-on exercises and labs that cover a wide range of topics in web penetration testing. While some content is paid, there are free exercises available for learning and practicing various skills.

VulnHub:

Website: VulnHub

Description: VulnHub is a platform that provides a collection of vulnerable virtual machines for penetration testing practice. Users can download these machines and attempt to exploit them in a safe and controlled environment.

OWASP WebGoat:

Website: OWASP WebGoat Project

Description: WebGoat is a deliberately insecure web application designed to teach web application security lessons. It is maintained by OWASP (Open Web Application Security Project) and is available for free to help users practice their skills in a controlled environmen virtual machine for cybersecurity and penetration testing purposes, you have several options for virtualization software. Here are a few commonly used virtualization platforms:

Oracle VM VirtualBox:

Description: VirtualBox is a free and open-source virtualization platform developed by Oracle. It supports a variety of guest operating systems, making it a versatile choice for creating virtual machines. VirtualBox is user-friendly and suitable for beginners.

VMware Workstation Player:

Description: VMware Workstation Player is a free virtualization software that allows you to run, evaluate, and test various operating systems in a virtual environment. It is a popular choice for both professionals and enthusiasts.

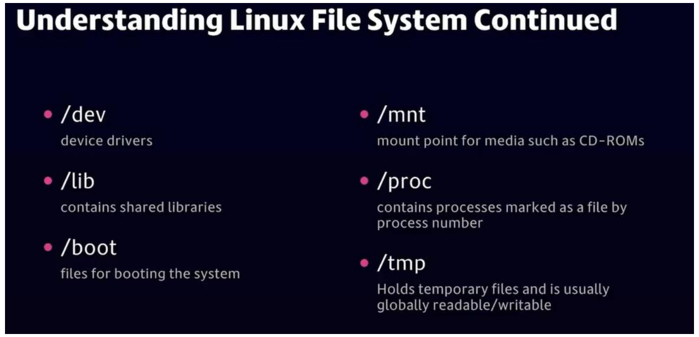
VMware Fusion (for macOS):

Description: VMware Fusion is a virtualization software specifically designed for macOS. It enables users to run Windows and other operating systems on their Mac machines. It's a paid software but offers a free trial.

Hyper-V (for Windows):

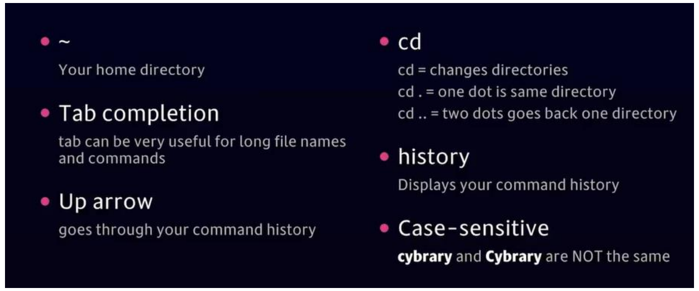
Description: Hyper-V is a virtualization platform provided by Microsoft for Windows users. It is available as a standalone product (Hyper-V Manager) or as a feature within Windows operating systems. Hyper-V is commonly used in enterprise environments.

Kali Linux Downloaded : https://www.kali.org/get-kali/#kali-installer-images : installation video : kali installation :
https://www.youtube.com/watch?v=cT1wxs6rNrE&t=2

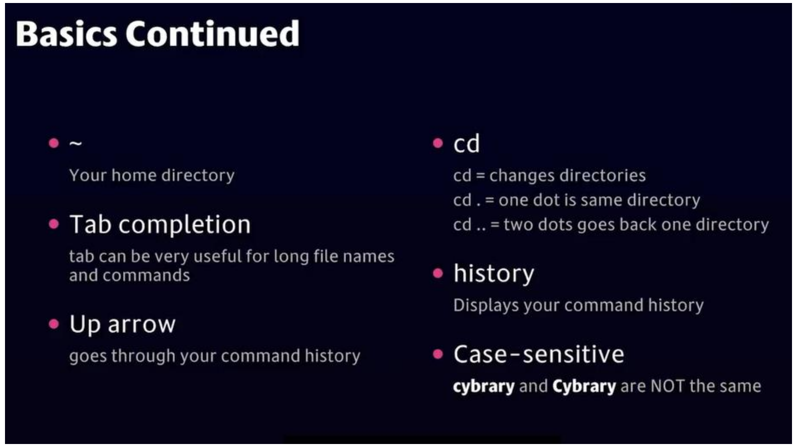


Understanding Linux File System Continued

- /dev
  device drivers
- /lib
  contains shared libraries
- /boot
  files for booting the system
- /mnt
  mount point for media such as CD-ROMs
- /proc
  contains processes marked as a file by process number
- /tmp
  Holds temporary files and is usually globally readable/writable

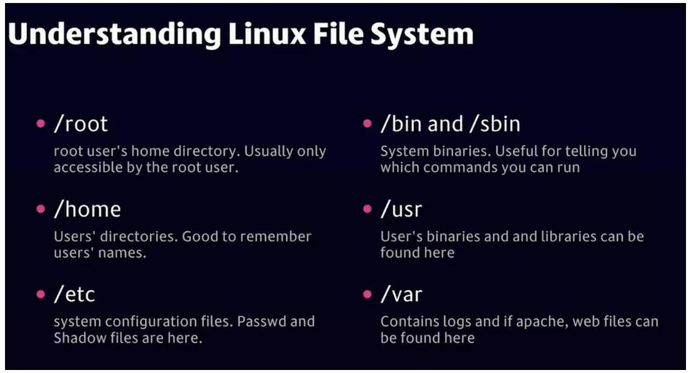


- ~
  Your home directory
- Tab completion
  tab can be very useful for long file names and commands
- Up arrow
  goes through your command history
- cd
  cd = changes directories
  cd . = one dot is same directory
  cd .. = two dots goes back one directory
- history
  Displays your command history
- Case-sensitive
  cybrary and **Cybrary** are NOT the same



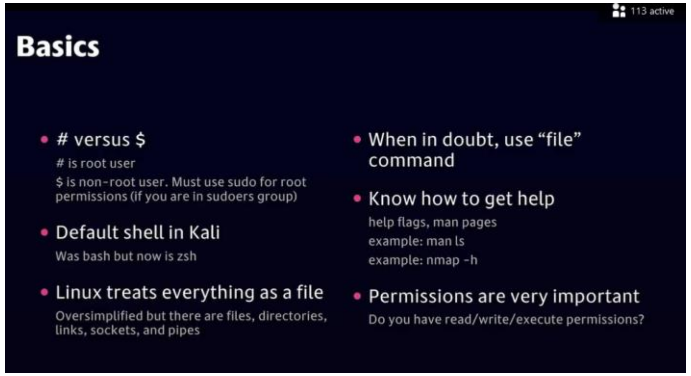
Basics Continued

- ~
  Your home directory
- Tab completion
  tab can be very useful for long file names and commands
- Up arrow
  goes through your command history
- cd
  cd = changes directories
  cd . = one dot is same directory
  cd .. = two dots goes back one directory
- history
  Displays your command history
- Case-sensitive
  cybrary and **Cybrary** are NOT the same



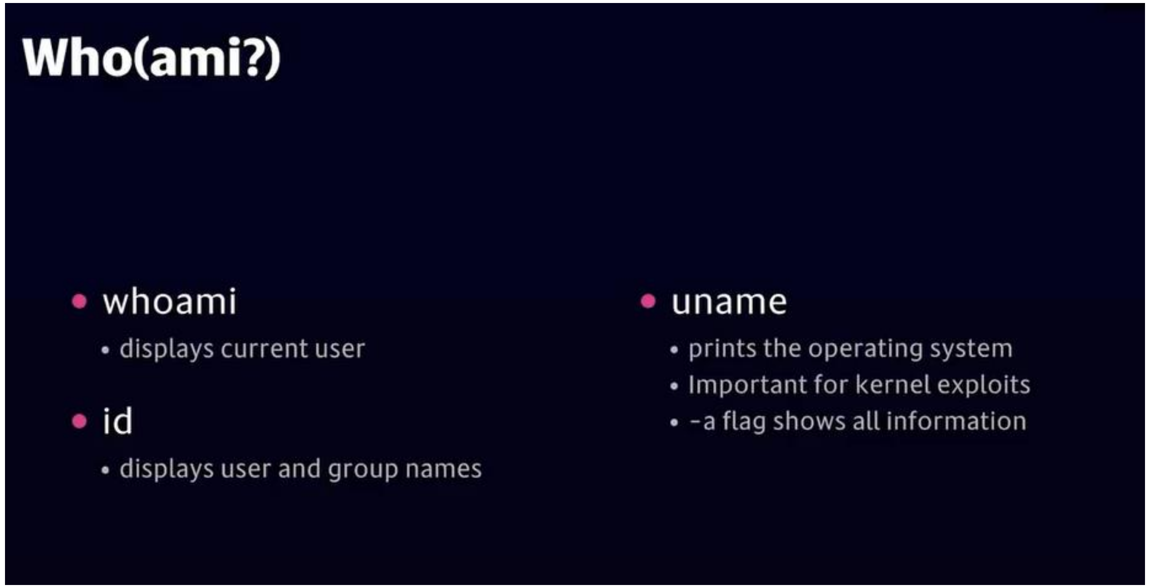
Understanding Linux File System

- /root
  root user's home directory. Usually only accessible by the root user.
- /home
  Users' directories. Good to remember users' names.
- /etc
  system configuration files. Passwd and Shadow files are here.
- /bin and /sbin
  System binaries. Useful for telling you which commands you can run
- /usr
  User's binaries and and libraries can be found here
- /var
  Contains logs and if apache, web files can be found here



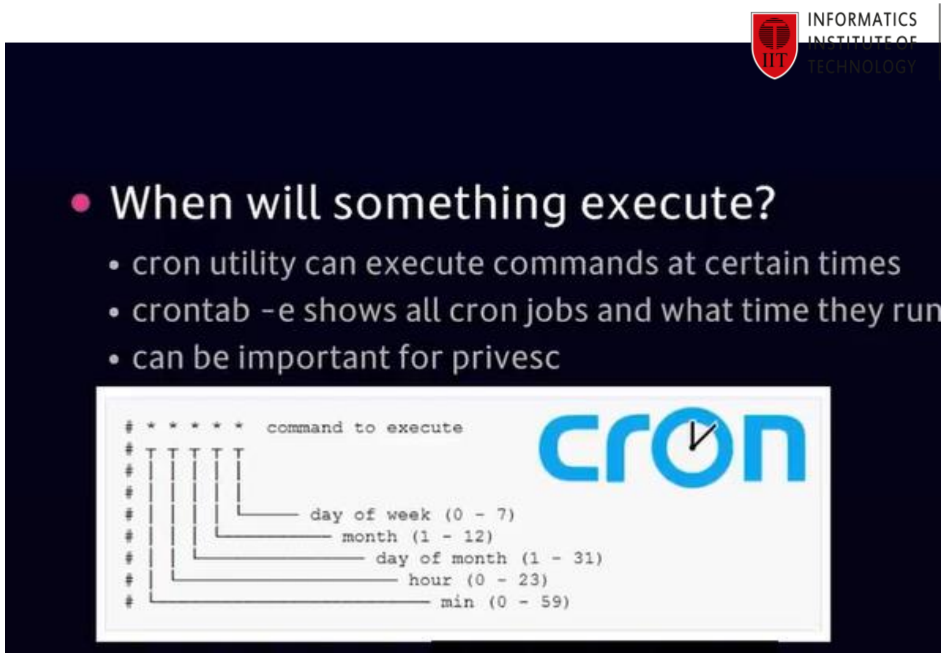
Basics

- # versus $
  # is root user
  $ is non-root user. Must use sudo for root permissions (if you are in sudoers group)
- Default shell in Kali
  Was bash but now is zsh
- Linux treats everything as a file
  Oversimplified but there are files, directories, links, sockets, and pipes
- When in doubt, use “file” command
- Know how to get help
  help flags, man pages
  example: man ls
  example: nmap -h
- Permissions are very important
  Do you have read/write/execute permissions?



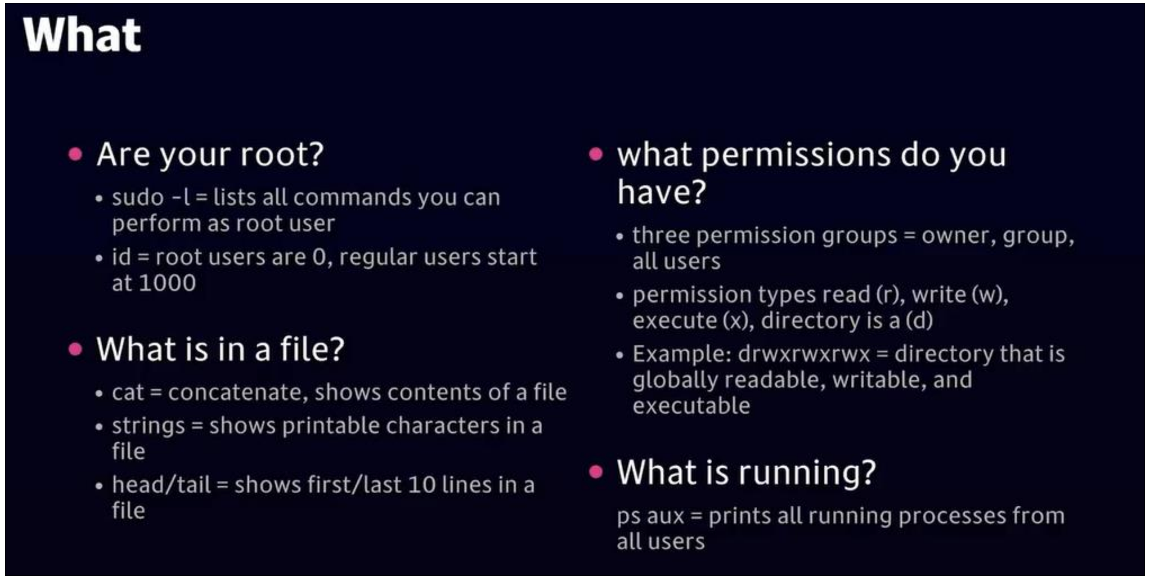
Who(ami?)

- whoami
  - displays current user
- id
  - displays user and group names
- uname
  - prints the operating system
  - Important for kernel exploits
  - -a flag shows all information



- When will something execute?
  - cron utility can execute commands at certain times
  - crontab -e shows all cron jobs and what time they run
  - can be important for privesc

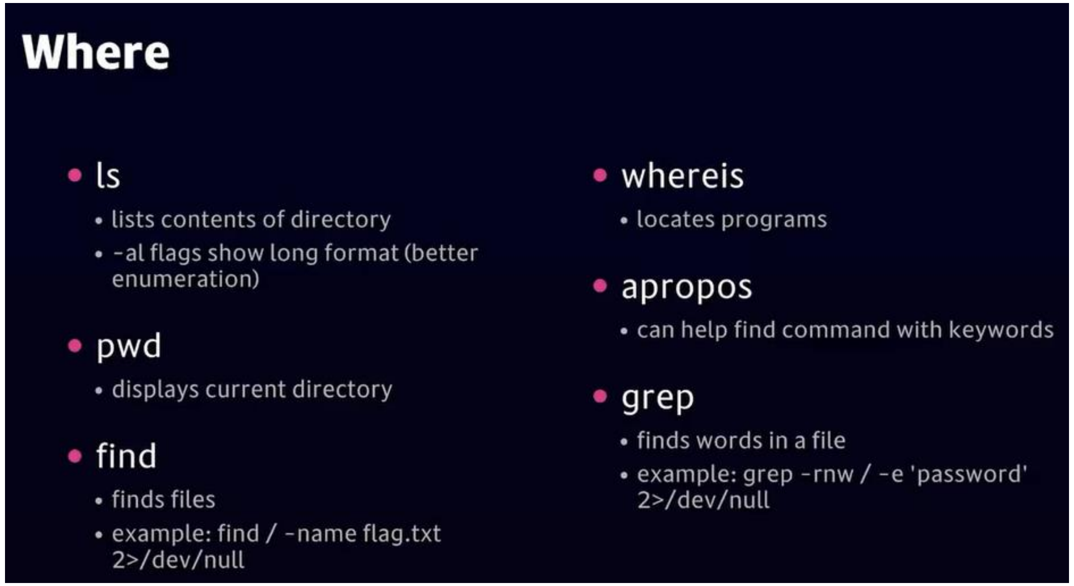
```
# * * * * *  command to execute
# ┬ ┬ ┬ ┬ ┬
# │ │ │ │ │
# │ │ │ │ │
# │ │ │ │ └──── day of week (0 - 7)
# │ │ │ └────── month (1 - 12)
# │ │ └──────── day of month (1 - 31)
# │ └────────── hour (0 - 23)
# └──────────── min (0 - 59)
```



What

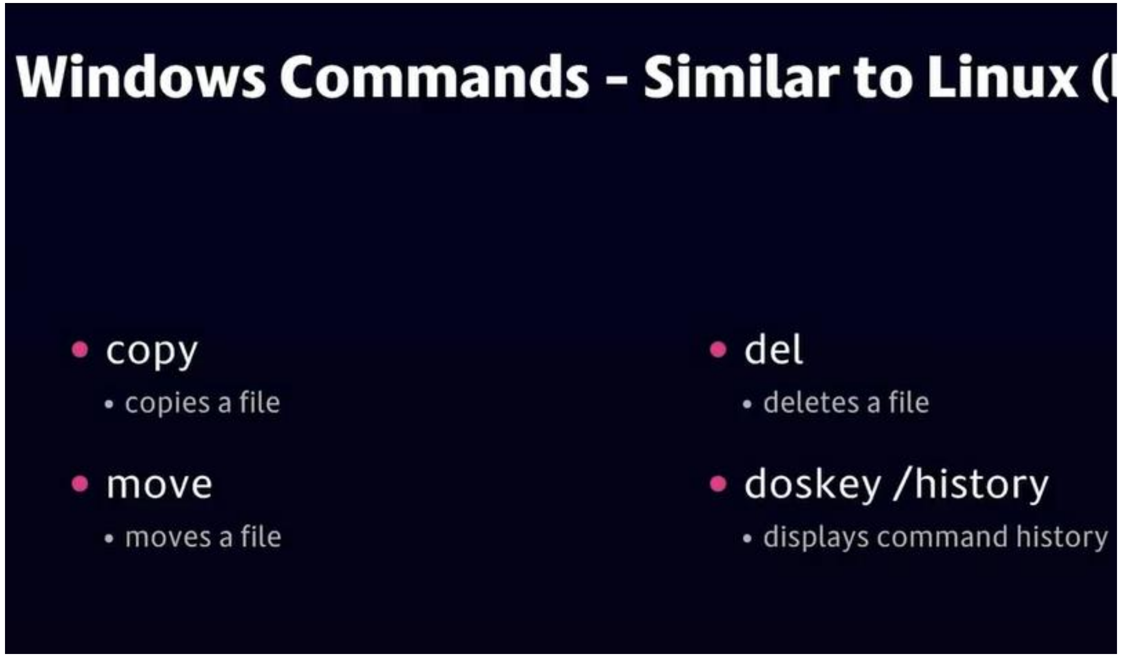
- Are your root?
  - sudo -l = lists all commands you can perform as root user
  - id = root users are 0, regular users start at 1000
- What is in a file?
  - cat = concatenate, shows contents of a file
  - strings = shows printable characters in a file
  - head/tail = shows first/last 10 lines in a file
- what permissions do you have?
  - three permission groups = owner, group, all users
  - permission types read (r), write (w), execute (x), directory is a (d)
  - Example: drwxrwxrwx = directory that is globally readable, writable, and executable
- What is running?

  ps aux = prints all running processes from all users



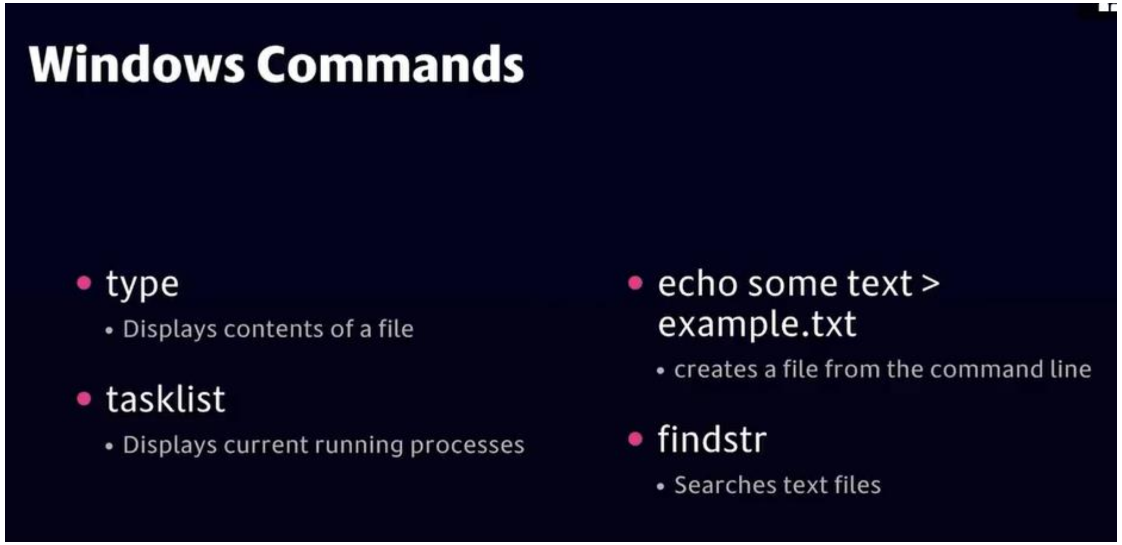
Where

- ls
  - lists contents of directory
  - -al flags show long format (better enumeration)
- pwd
  - displays current directory
- find
  - finds files
  - example: find / -name flag.txt 2>/dev/null
- whereis
  - locates programs
- apropos
  - can help find command with keywords
- grep
  - finds words in a file
  - example: grep -rnw / -e 'password' 2>/dev/null



Windows Commands - Similar to Linux (

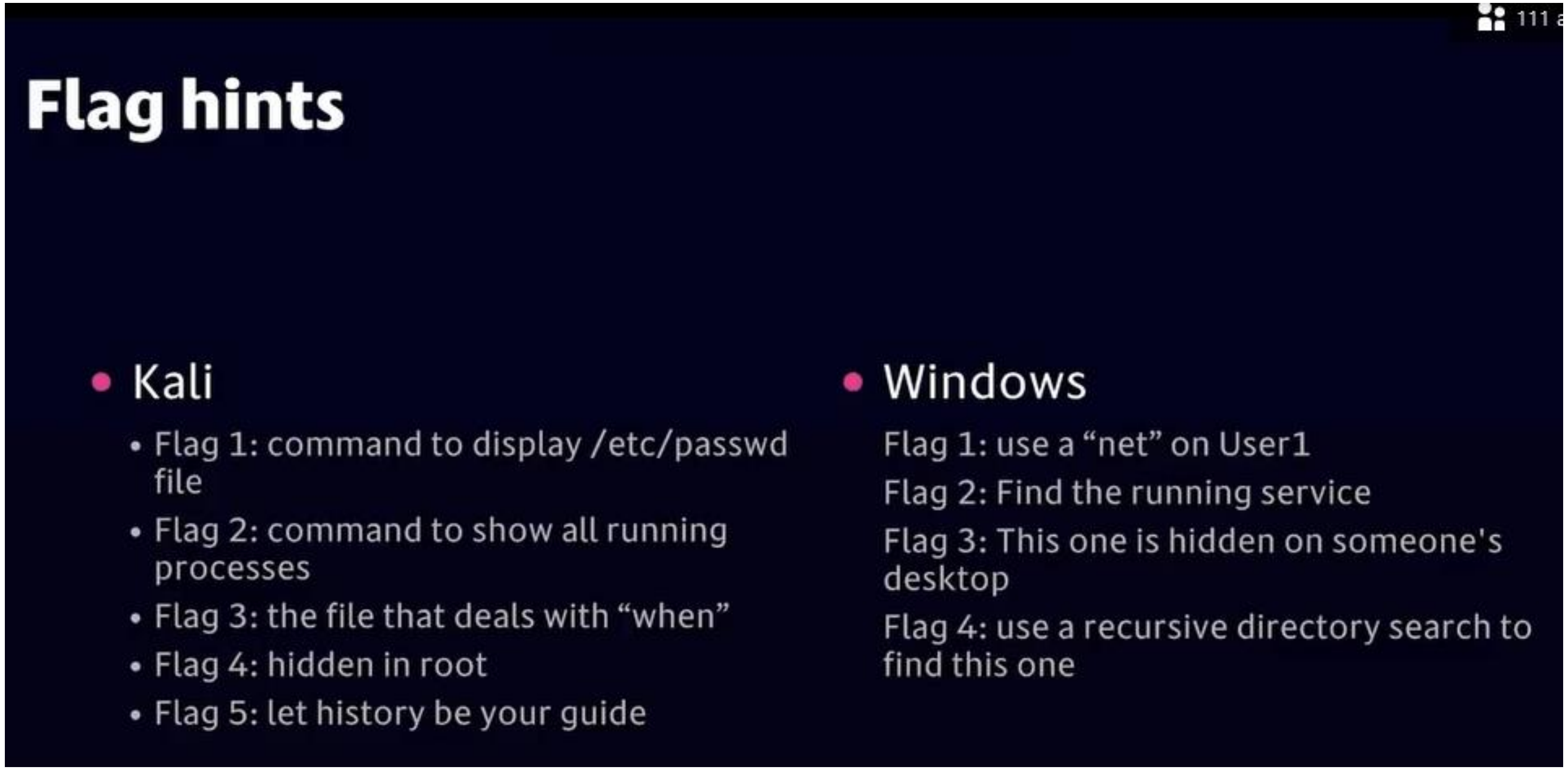
- copy
  - copies a file
- move
  - moves a file
- del
  - deletes a file
- doskey /history
  - displays command history



Windows Commands

- type
  - Displays contents of a file
- tasklist
  - Displays current running processes
- echo some text > example.txt
  - creates a file from the command line
- findstr
  - Searches text files

Command line labs



Flag hints

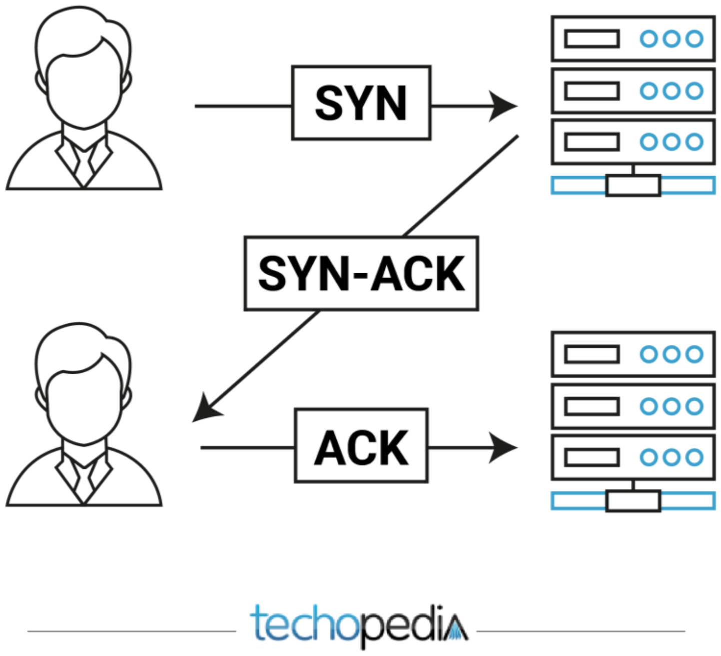
- Kali
  - Flag 1: command to display /etc/passwd file
  - Flag 2: command to show all running processes
  - Flag 3: the file that deals with “when”
  - Flag 4: hidden in root
  - Flag 5: let history be your guide
- Windows

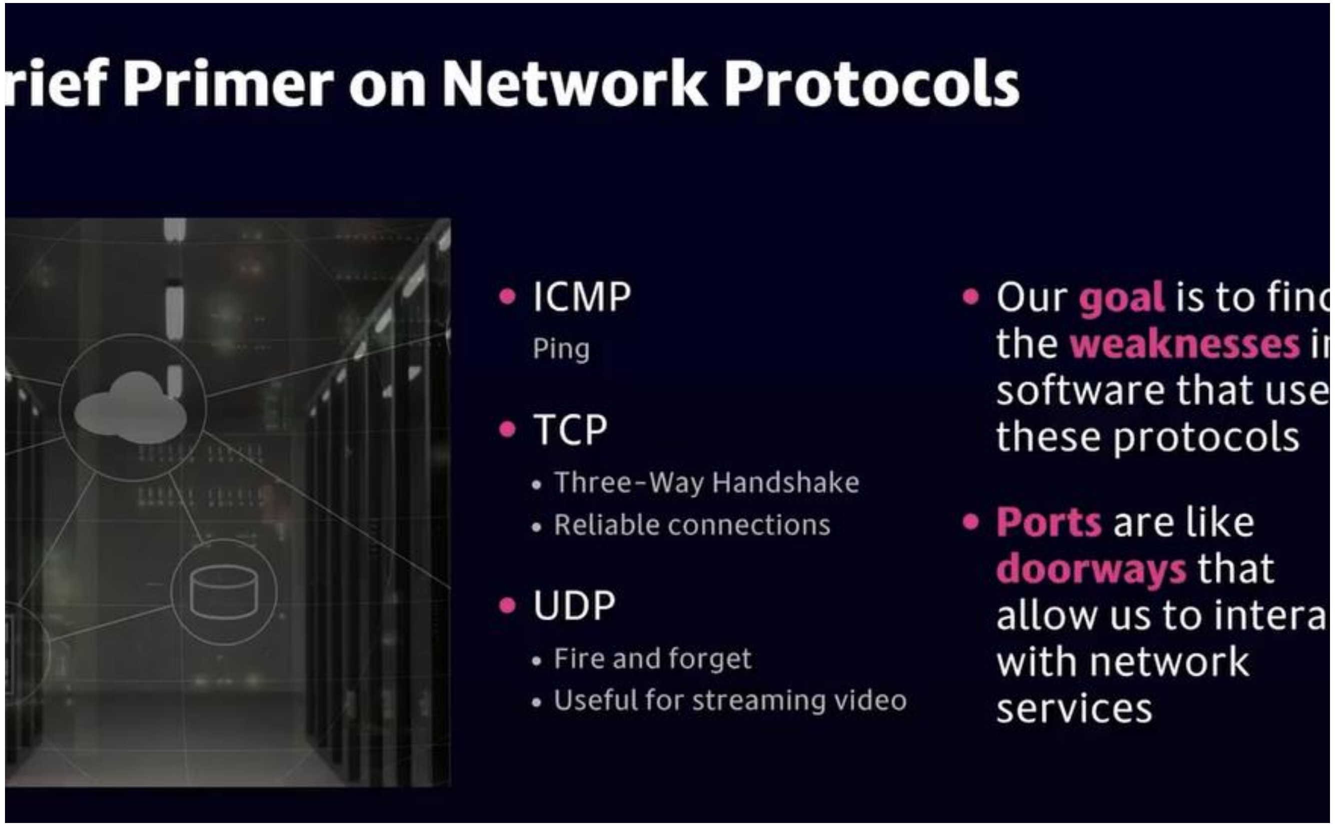
  Flag 1: use a “net” on User1

  Flag 2: Find the running service

  Flag 3: This one is hidden on someone's desktop

  Flag 4: use a recursive directory search to find this one





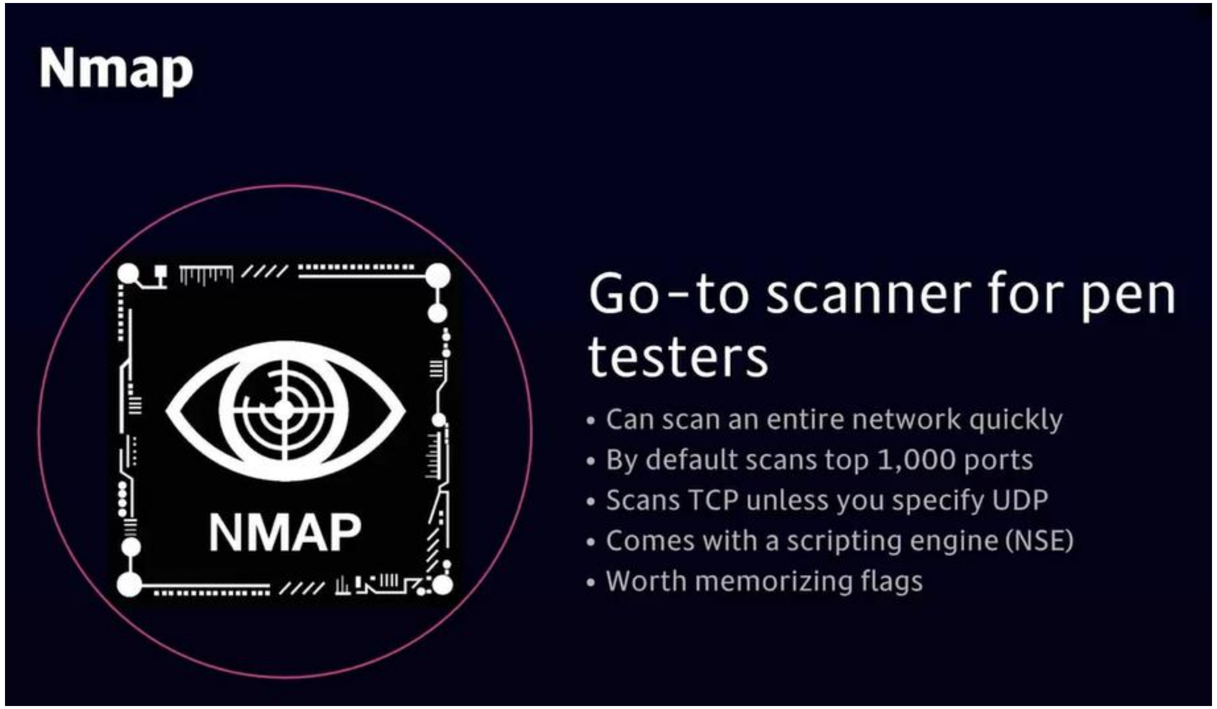
rief Primer on Network Protocols

- ICMP
  Ping
- TCP
  - Three-Way Handshake
  - Reliable connections
- UDP
  - Fire and forget
  - Useful for streaming video

- Our **goal** is to finc the **weaknesses** ir software that use these protocols
- **Ports** are like **doorways** that allow us to intera with network services

understanding network protocol

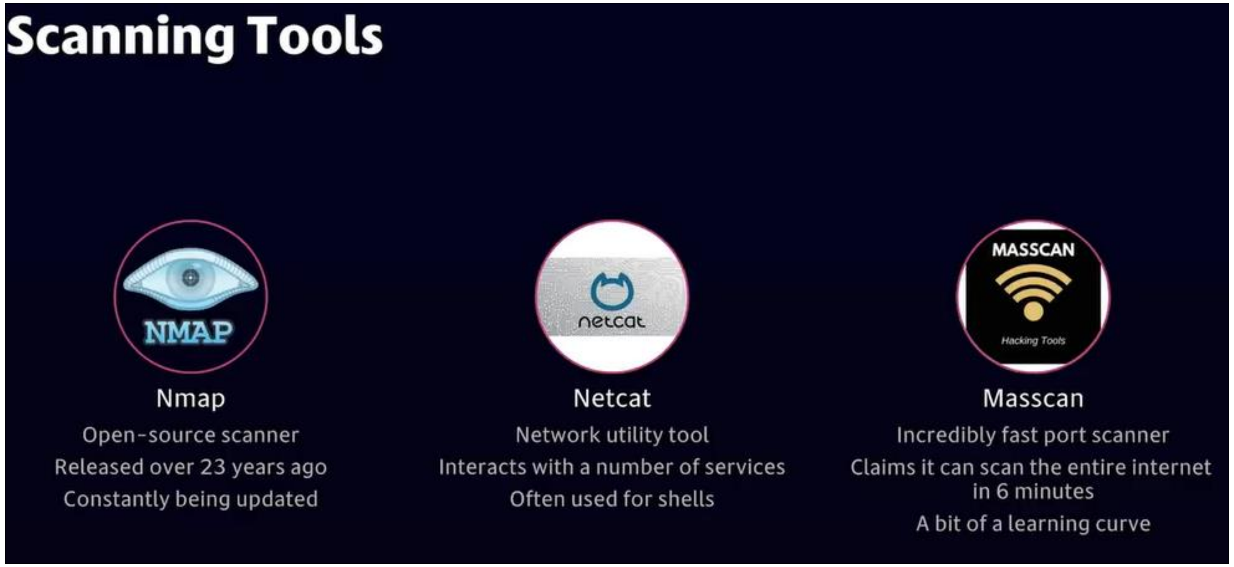
Scanning Nmap



Nmap

Go-to scanner for pen testers

- Can scan an entire network quickly
- By default scans top 1,000 ports
- Scans TCP unless you specify UDP
- Comes with a scripting engine (NSE)
- Worth memorizing flags

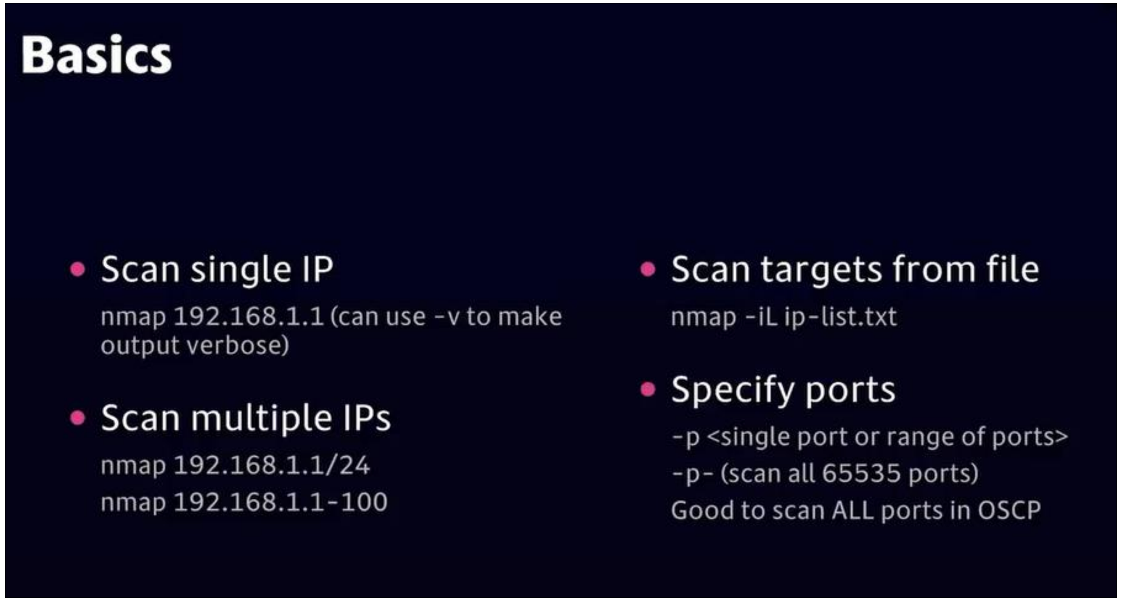


Scanning Tools

Nmap
Open-source scanner
Released over 23 years ago
Constantly being updated

Netcat
Network utility tool
Interacts with a number of services
Often used for shells

Masscan
Incredibly fast port scanner
Claims it can scan the entire internet in 6 minutes
A bit of a learning curve



Basics

- Scan single IP

  nmap 192.168.1.1 (can use -v to make output verbose)

- Scan multiple IPs

  nmap 192.168.1.1/24

  nmap 192.168.1.1-100

- Scan targets from file

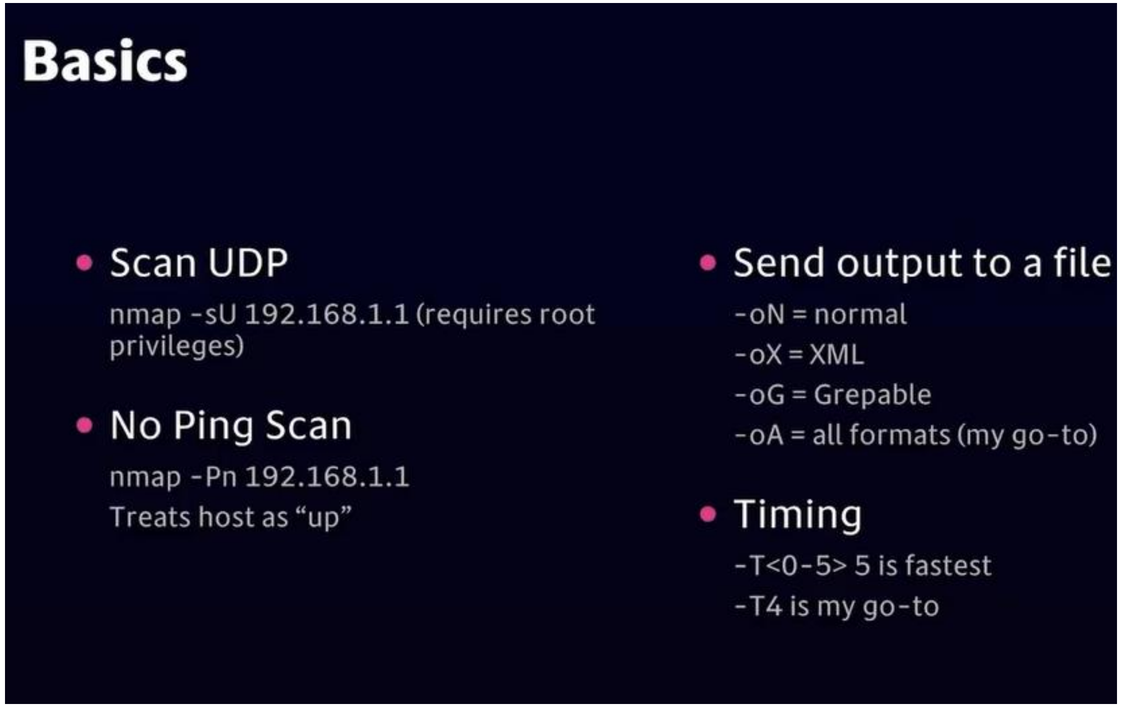
  nmap -iL ip-list.txt

- Specify ports

  -p <single port or range of ports>

  -p- (scan all 65535 ports)

  Good to scan ALL ports in OSCP



Basics

- Scan UDP

  nmap -sU 192.168.1.1 (requires root privileges)

- No Ping Scan

  nmap -Pn 192.168.1.1

  Treats host as “up”

- Send output to a file

  -oN = normal

  -oX = XML

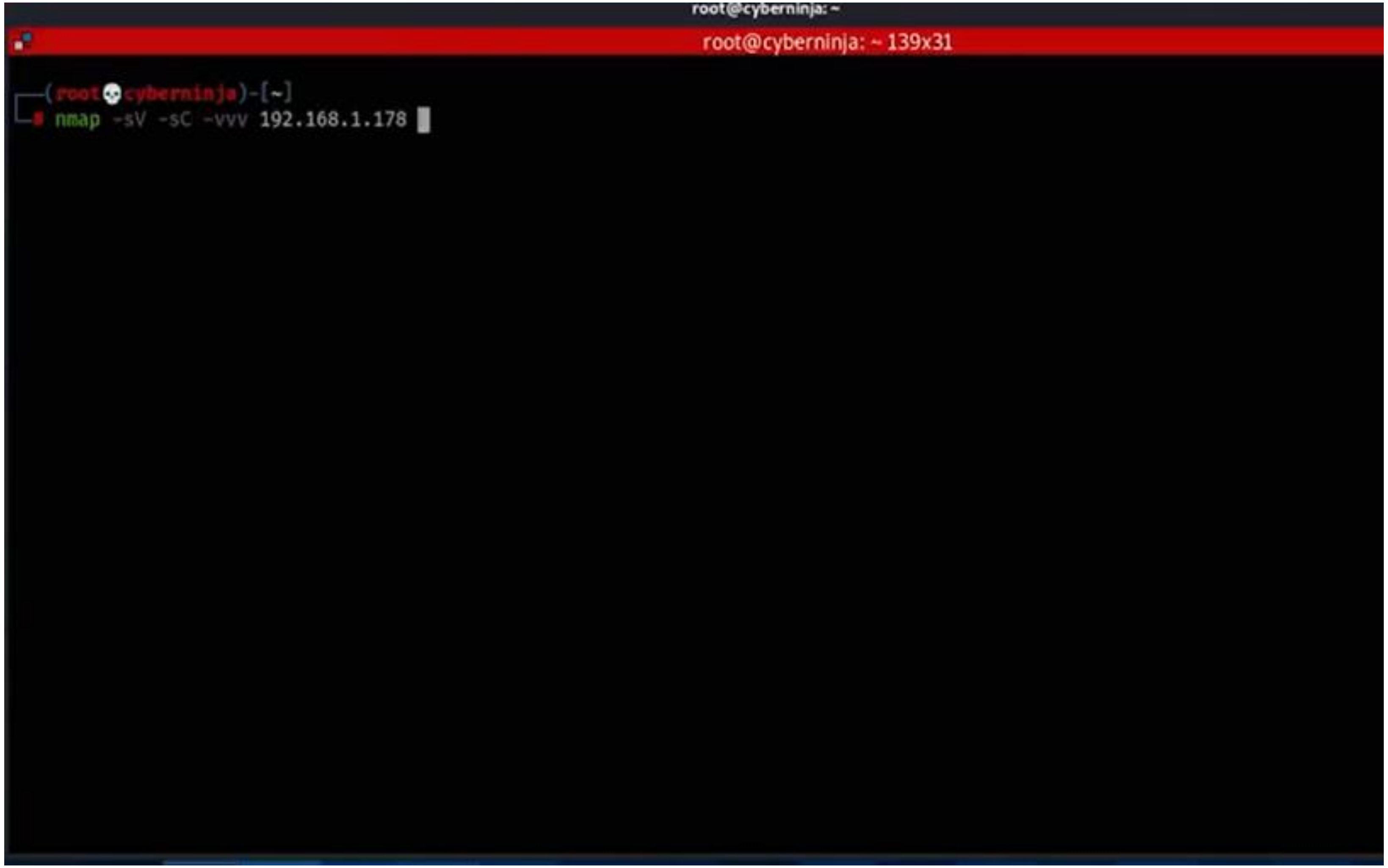
  -oG = Grepable

  -oA = all formats (my go-to)

- Timing

  -T<0-5> 5 is fastest

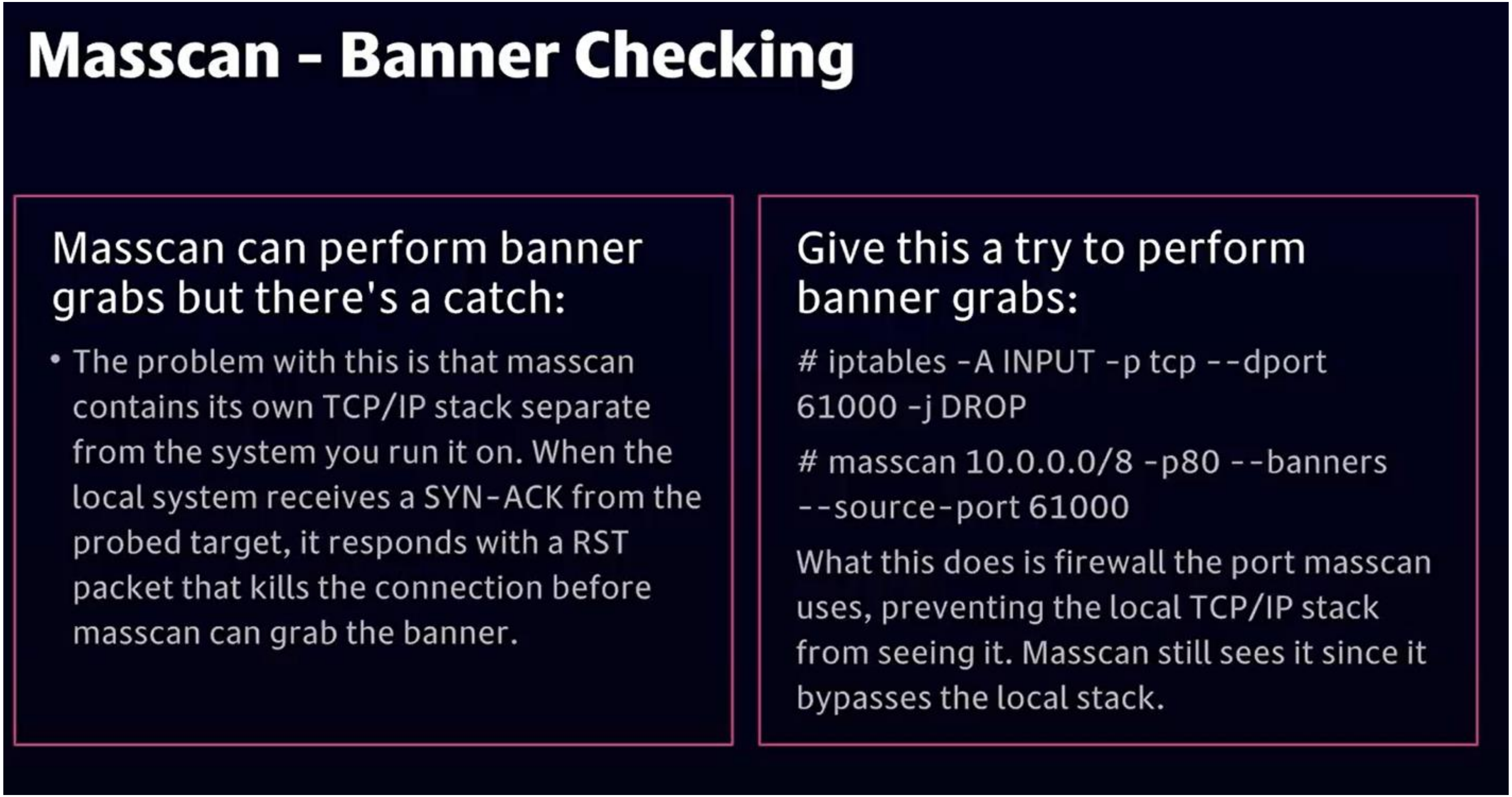
  -T4 is my go-to



```
┌──(root💀cyberninja)-[~]
└─# nmap -sV -sC -vvv 192.168.1.178
```

Masscan

Fast port scanner which comes pre-installed in Kali



Masscan - Banner Checking

Masscan can perform banner grabs but there's a catch:

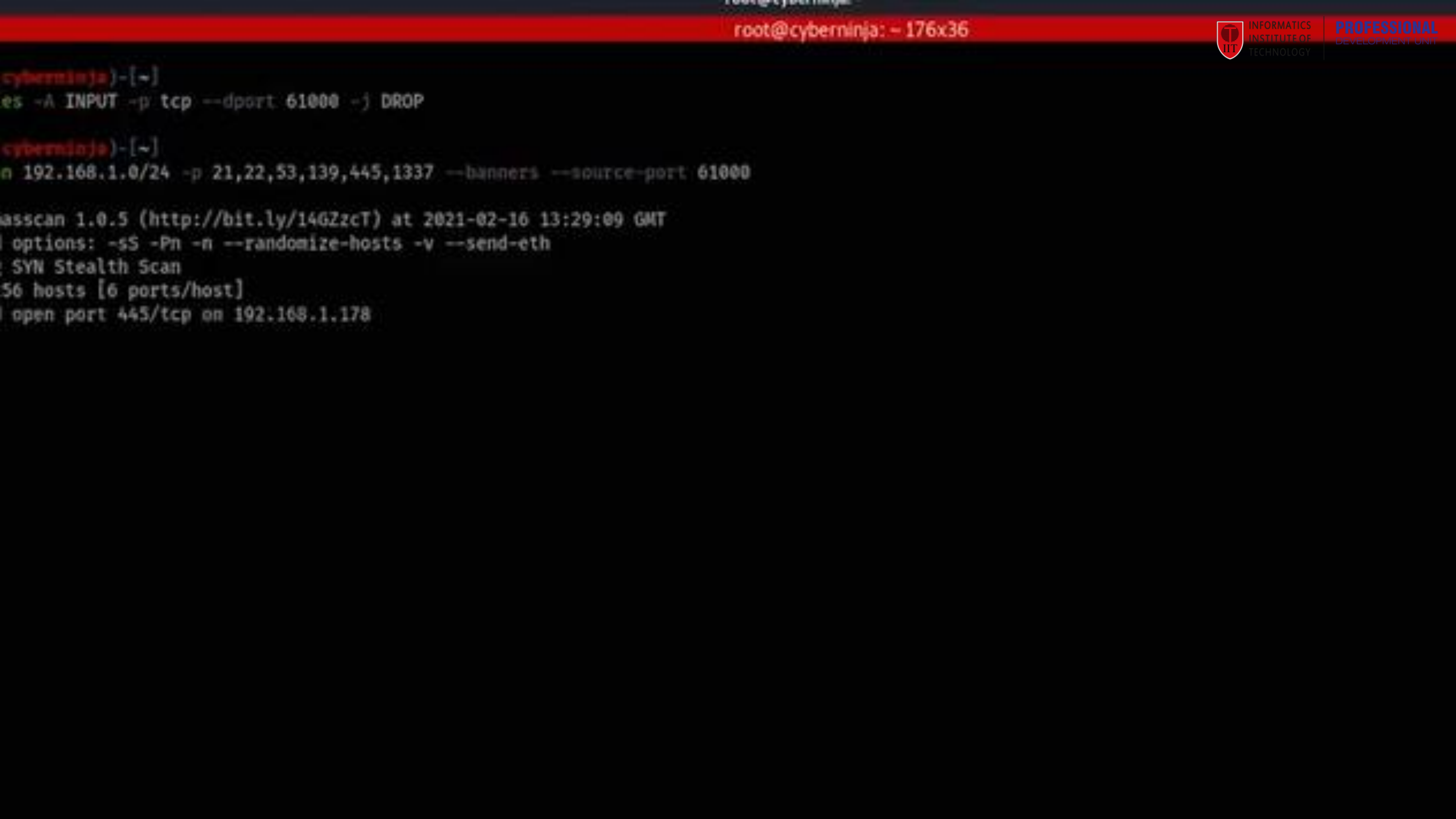
- The problem with this is that masscan contains its own TCP/IP stack separate from the system you run it on. When the local system receives a SYN-ACK from the probed target, it responds with a RST packet that kills the connection before masscan can grab the banner.

Give this a try to perform banner grabs:

```
# iptables -A INPUT -p tcp --dport 61000 -j DROP
```

```
# masscan 10.0.0.0/8 -p80 --banners --source-port 61000
```

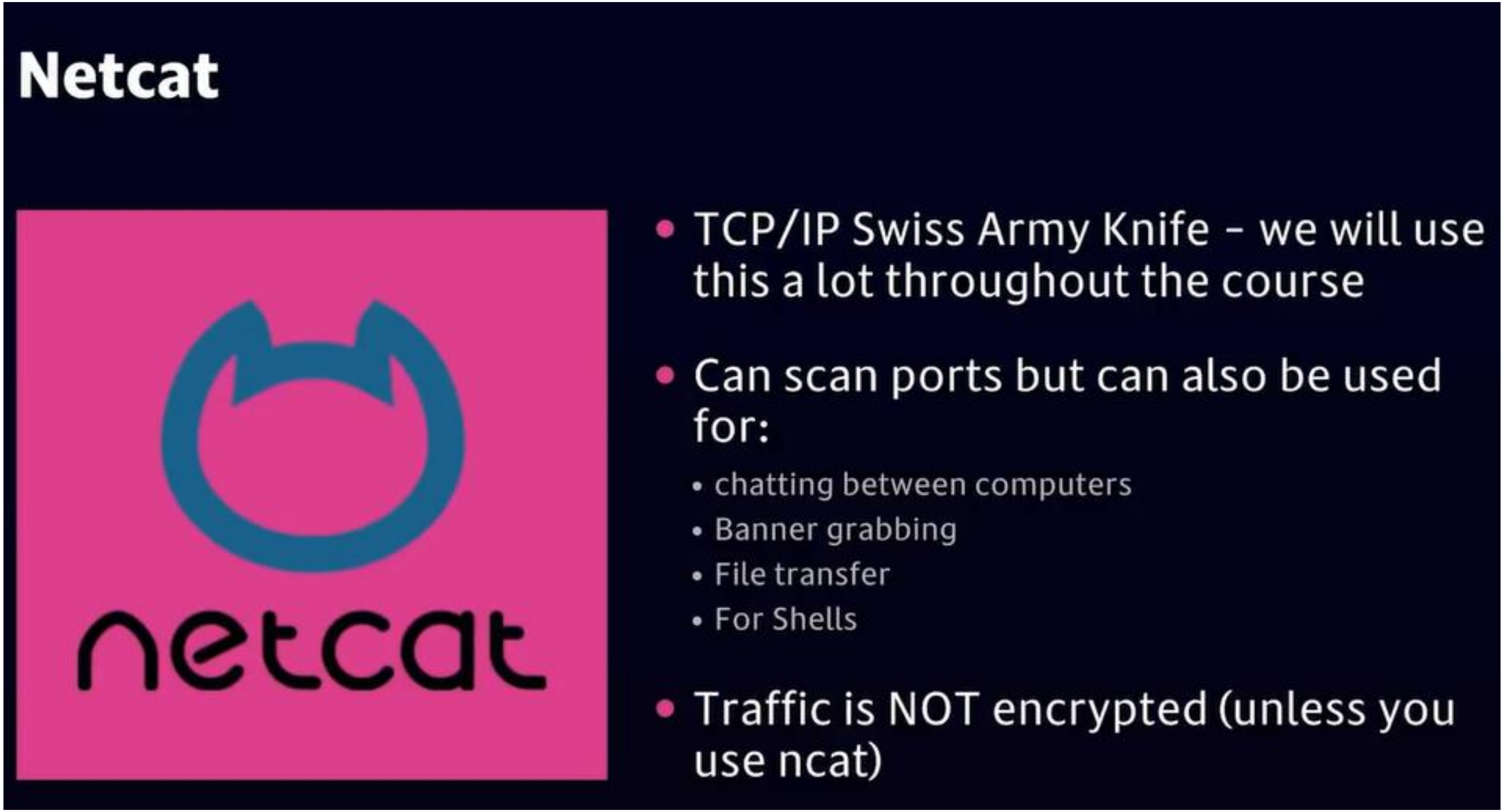
What this does is firewall the port masscan uses, preventing the local TCP/IP stack from seeing it. Masscan still sees it since it bypasses the local stack.



```
cyberninja)-[~]
es -A INPUT -p tcp --dport 61000 -j DROP

cyberninja)-[~]
n 192.168.1.0/24 -p 21,22,53,139,445,1337 --banners --source-port 61000

asscan 1.0.5 (http://bit.ly/14GZzcT) at 2021-02-16 13:29:09 GMT
 options: -sS -Pn -n --randomize-hosts -v --send-eth
 SYN Stealth Scan
56 hosts [6 ports/host]
 open port 445/tcp on 192.168.1.178
```



Netcat

- TCP/IP Swiss Army Knife – we will use this a lot throughout the course
- Can scan ports but can also be used for:
  - chatting between computers
  - Banner grabbing
  - File transfer
  - For Shells
- Traffic is NOT encrypted (unless you use ncat)

- Syntax: nc -nv -w 1 -z <target-ip> <port range>

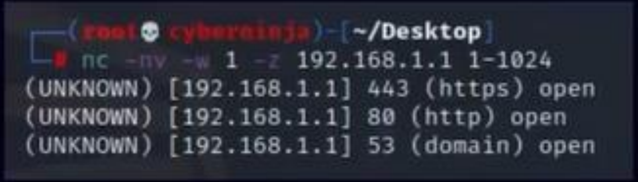
  -nv = doesn't resolve DNS and prints verbose output

  -w 1 = sets timeout to 1 second

  -z = specifies a port scan

  -u = UDP mode (can be unreliable)

- Can be slow to scan, but may find things nmap doesn't



```
┌──(root💀cyberninja)-[~/Desktop]
└─# nc -nv -w 1 -z 192.168.1.1 1-1024
(UNKNOWN) [192.168.1.1] 443 (https) open
(UNKNOWN) [192.168.1.1] 80 (http) open
(UNKNOWN) [192.168.1.1] 53 (domain) open
```