Pentesting Linux

By Kyri

What is Pentesting?

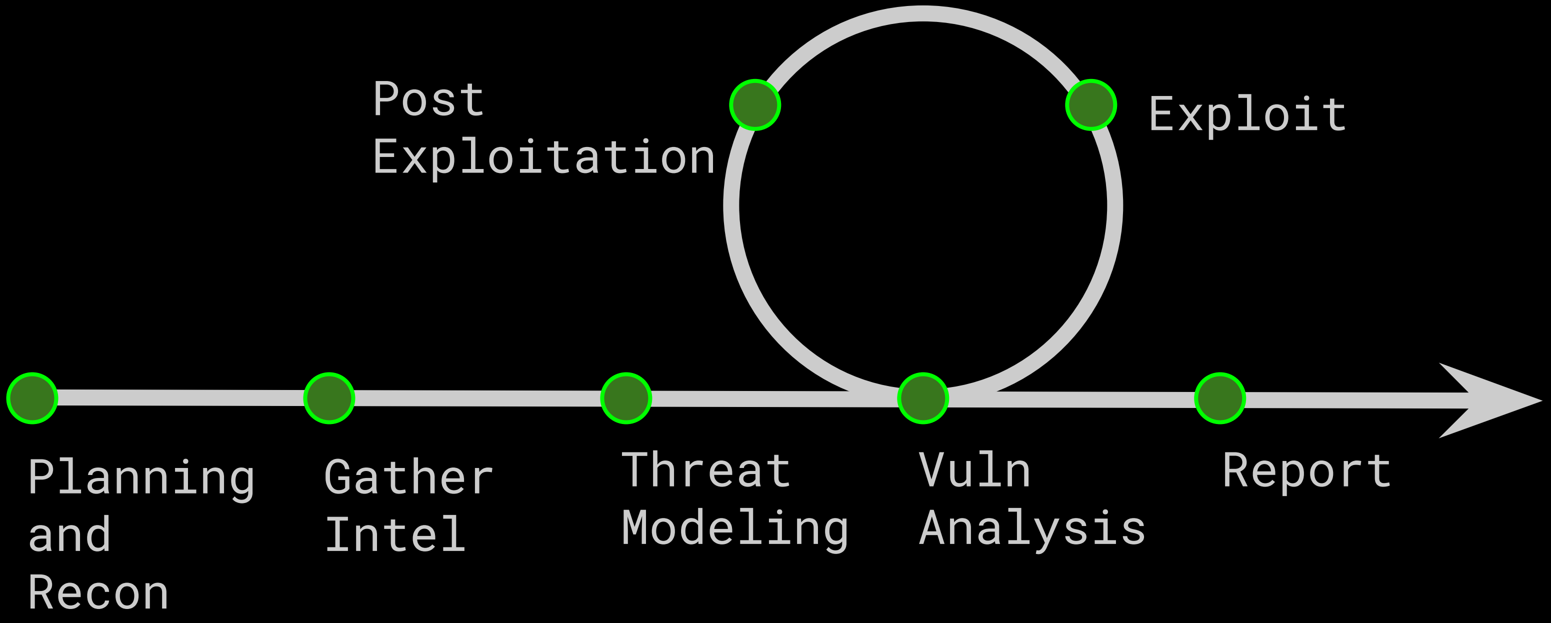
Penetration testing - simulating an attack against a computer or network to identify vulnerabilities.

- Also called ethical hacking
- Not the same as Red Teaming

Find the weak points before a real attacker

Provide remediation recommendations and impacts on CIA for the client

Methodology



Types of Tests

Internal and External Network

Web

Mobile

Physical

Wireless

Goals

We want to answer

- What vulnerabilities are present?
- What services are running?
- What data can we access?
    - Sensitivity of data
- What role does this server play in overall environment?

Not just about getting a shell/root

Enumeration and Info Gathering

Scanning

Ping Sweep

```
nmap -sn <ip/CIDR> -T<1-5>
```

TCP Scan - Find open ports

```
nmap -p- <ip/CIDR>
```

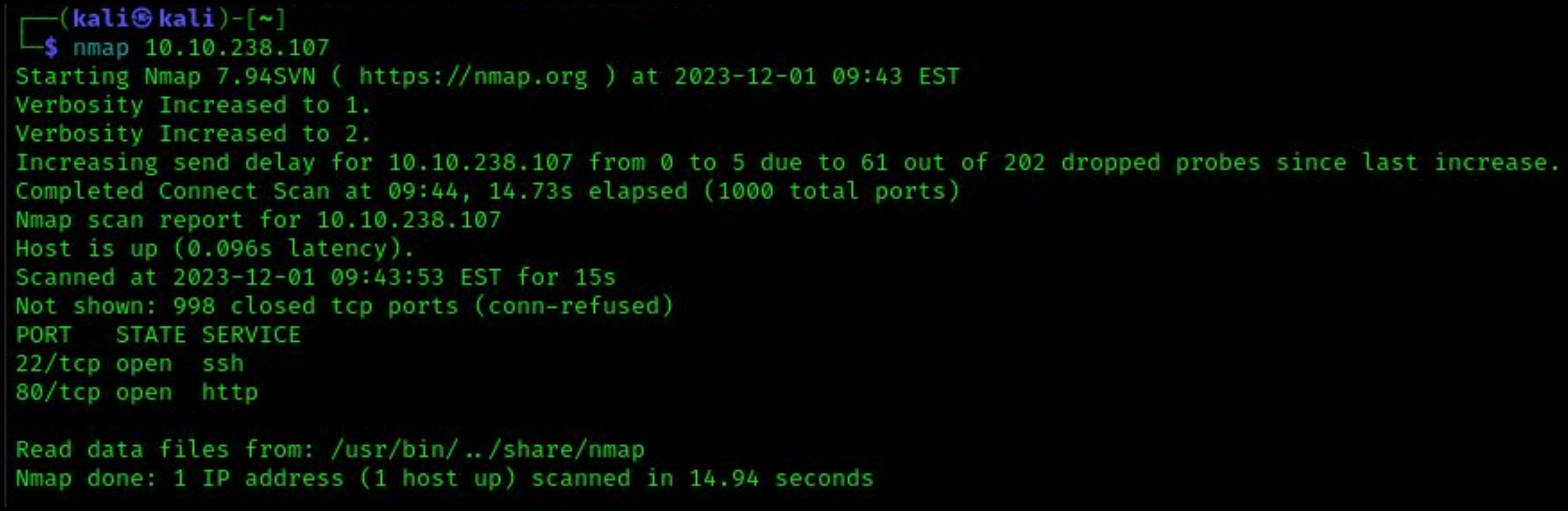
TCP Scan - Details

```
nmap -sC -sV -p <open_ports> <ip/CIDR>
```

UDP Scan

```
nmap -sU <ip/CIDR>
```

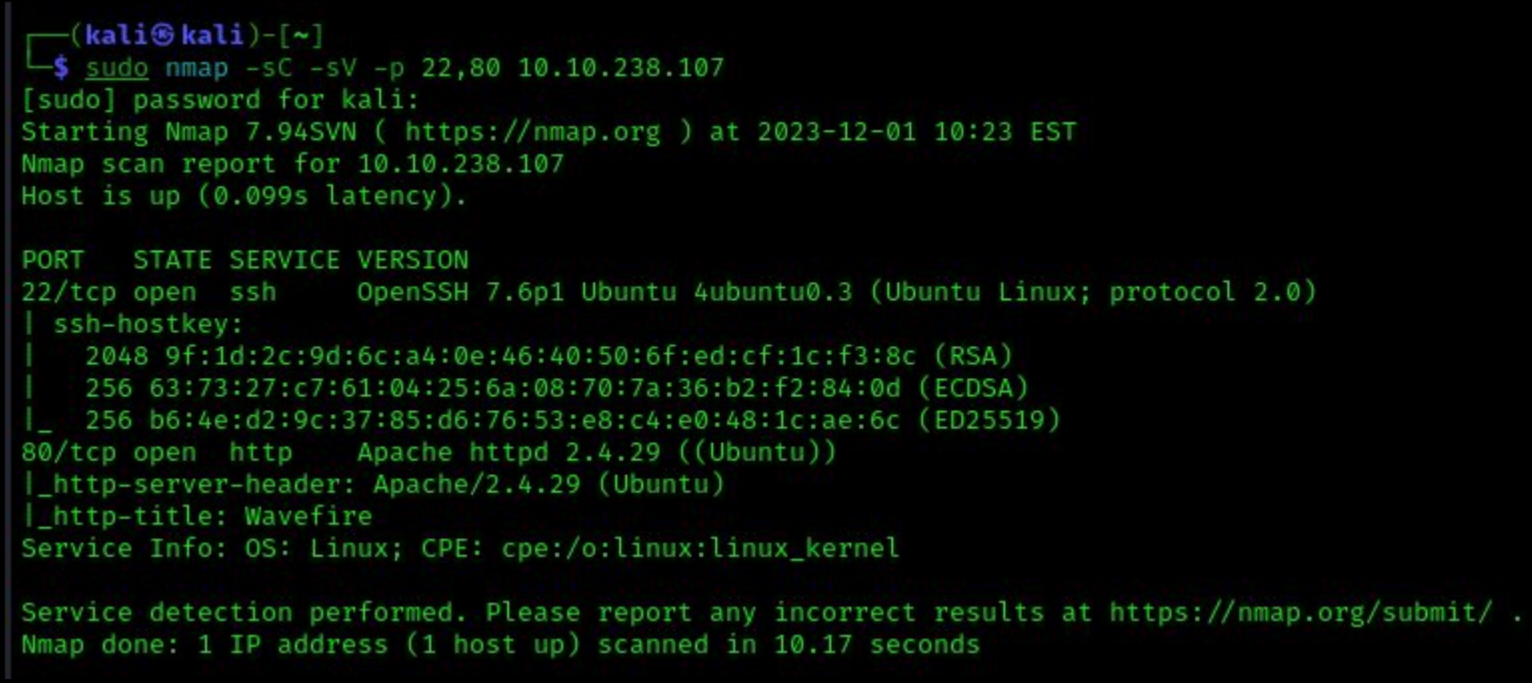
Nmap



```
┌──(kali㉿kali)-[~]
└─$ nmap 10.10.238.107
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-01 09:43 EST
Verbosity Increased to 1.
Verbosity Increased to 2.
Increasing send delay for 10.10.238.107 from 0 to 5 due to 61 out of 202 dropped probes since last increase.
Completed Connect Scan at 09:44, 14.73s elapsed (1000 total ports)
Nmap scan report for 10.10.238.107
Host is up (0.096s latency).
Scanned at 2023-12-01 09:43:53 EST for 15s
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 14.94 seconds
```

Nmap



```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sC -sV -p 22,80 10.10.238.107
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-01 10:23 EST
Nmap scan report for 10.10.238.107
Host is up (0.099s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 9f:1d:2c:9d:6c:a4:0e:46:40:50:6f:ed:cf:1c:f3:8c (RSA)
|   256 63:73:27:c7:61:04:25:6a:08:70:7a:36:b2:f2:84:0d (ECDSA)
|_  256 b6:4e:d2:9c:37:85:d6:76:53:e8:c4:e0:48:1c:ae:6c (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Wavefire
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.17 seconds
```

Public Exploits

Use software and version numbers from nmap service/version scan.

- Exploit-DB
- Blog Posts
- Metasploit

Credentials

Weak/default passwords

- admin:admin, admin:password, root:root, root:toor, etc.

Passwords stored in /etc/passwd

- Everyone can read
- Weak encryption, can be cracked
- Useful for finding additional usernames

Password Wordlists - cracking and brute force

- Rockyou

Guest and Anonymous Login

Automated Tools

LinPEAS - searches for possible paths for privilege escalation

- So cute, and has Mac and Windows versions

LinEnum - enumerates the system, providing user information, service configs, default passwords, etc.

Metasploit

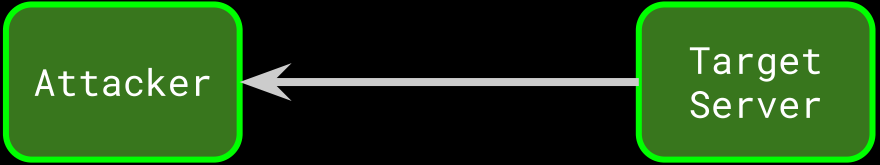
Framework for cross-platform exploitation

- Public exploits
- Shell handling
- Payload generation
- Auxiliary - port scanning, fuzzing

Anyone can create new modules for new vulnerabilities

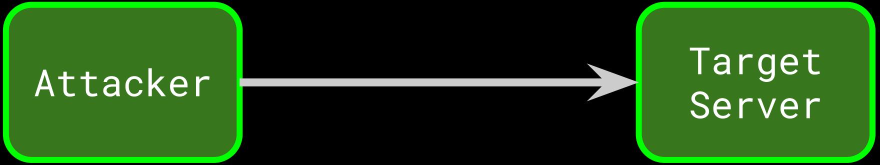
Types of Shells

Reverse Shell



Attacker acts as server.
Attacker port open.
Do not need to know target IP address.

Bind Shell



Target acts as server.
Target port open.
Need to know target IP address.

Services - FTP

Port 21

Anonymous Login
- file/information disclosure

Read/Write Permission
- Arbitrary file upload/download
- Upload and run executables

chroot Disabled
- Access to all files

Services - SSH

Port 22

Hydra - Brute Force Passwords

```
hydra -L <users> -P <passwords> <ip/CIDR> ssh -vV
```

Test passwords and private keys found elsewhere

Stable way to have a shell

Services - DNS

Port 53

Useful for network pentests

Dnsrecon - Brute-force subdomains

```
dnsrecon -n <nameserver> -d <domain> -D <wordlist> -t brt
```

Check /etc/hosts for more hostnames

Services - HTTP/HTTPS

Port 80/443

Web servers are often Linux, good place to target

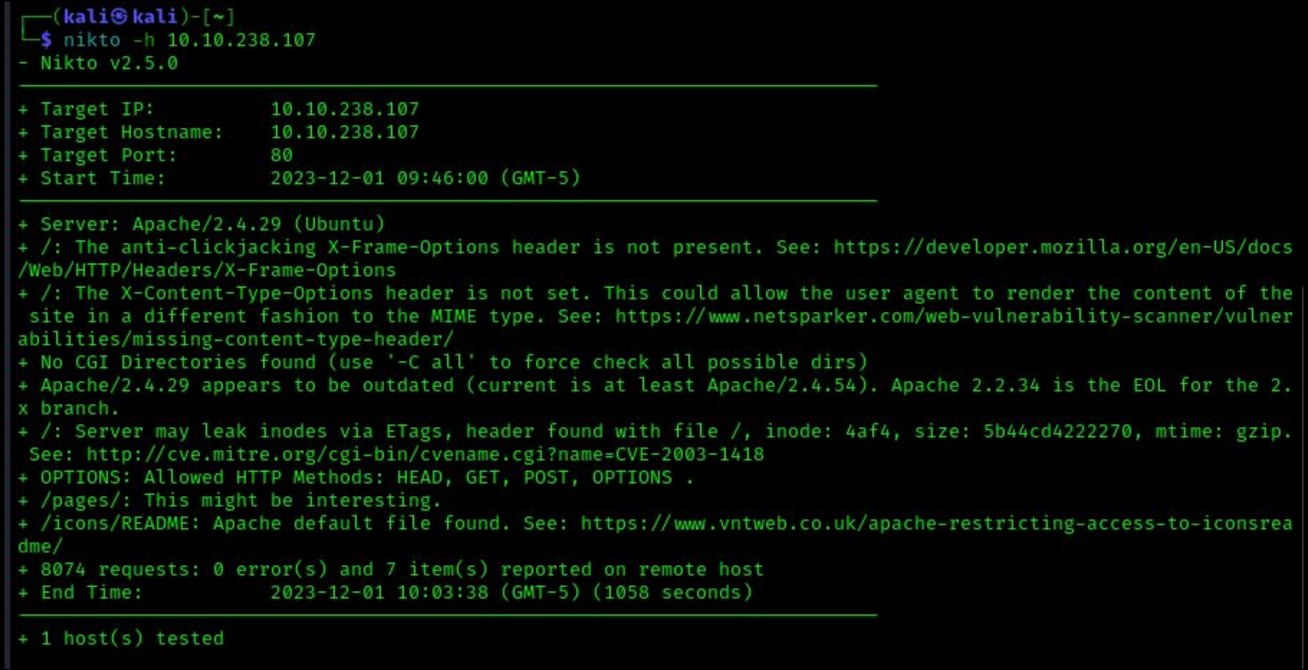
Look at the website, understand intended functionality

Inspect, View Source, robots.txt

- Hidden pages, usernames/passwords/sensitive information in comments
- Software and version

Services - HTTP/HTTPS

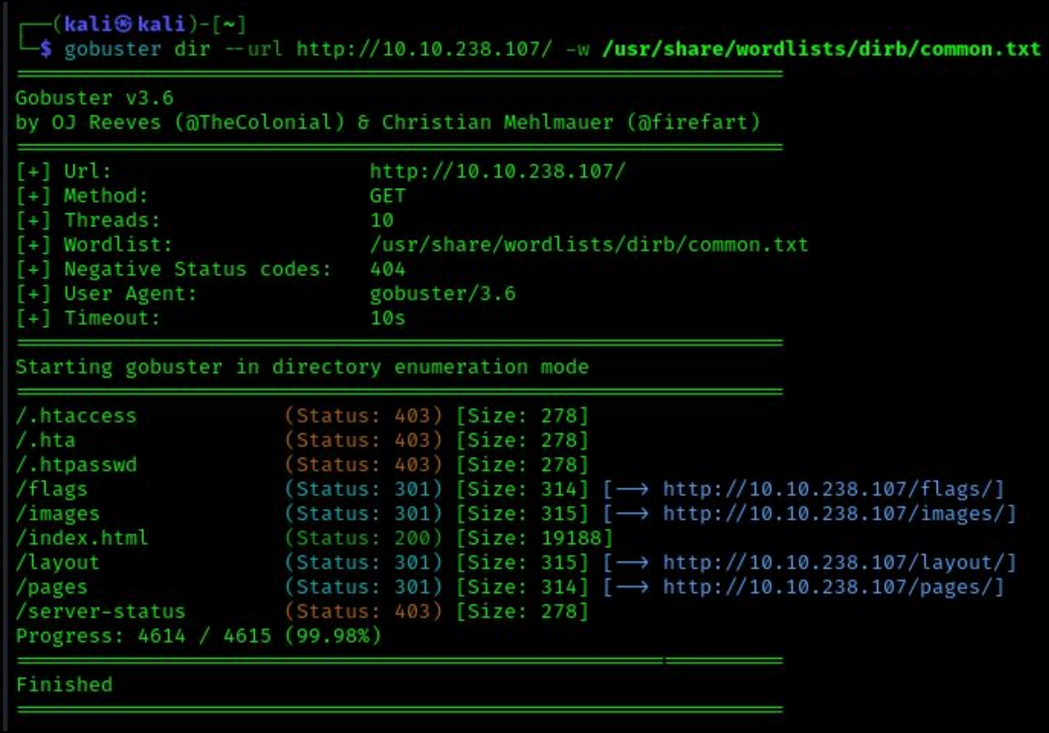
Nikto - Scan for basic weaknesses, default pages, out of date software, etc.



```
┌──(kali㉿kali)-[~]
└─$ nikto -h 10.10.238.107
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          10.10.238.107
+ Target Hostname:    10.10.238.107
+ Target Port:        80
+ Start Time:         2023-12-01 09:46:00 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.29 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs
/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the
 site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulner
abilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.
x branch.
+ /: Server may leak inodes via ETags, header found with file /, inode: 4af4, size: 5b44cd4222270, mtime: gzip.
 See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ OPTIONS: Allowed HTTP Methods: HEAD, GET, POST, OPTIONS .
+ /pages/: This might be interesting.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsrea
dme/
+ 8074 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time:           2023-12-01 10:03:38 (GMT-5) (1058 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Services - HTTP/HTTPS

Gobuster/Dirb - Directory bruteforce



```
┌──(kali㉿kali)-[~]
└─$ gobuster dir --url http://10.10.238.107/ -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.238.107/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.htaccess            (Status: 403) [Size: 278]
/.hta                 (Status: 403) [Size: 278]
/.htpasswd            (Status: 403) [Size: 278]
/flags                (Status: 301) [Size: 314] [--> http://10.10.238.107/flags/]
/images               (Status: 301) [Size: 315] [--> http://10.10.238.107/images/]
/index.html           (Status: 200) [Size: 19188]
/layout               (Status: 301) [Size: 315] [--> http://10.10.238.107/layout/]
/pages                (Status: 301) [Size: 314] [--> http://10.10.238.107/pages/]
/server-status        (Status: 403) [Size: 278]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================
```

Services - HTTP/HTTPS

Local File Inclusion - Escaping the web server directory to see other files on the system

- http://web.com/page.php?param=../../../../../etc/passwd

Command Injection - When user input is run as an OS command

- Characters like ;, &&, ||

SQL Injection - Gather information, maybe even log in

And so, so many more

Privilege Escalation and Exploitation

Privesc - Shell Escape Sequence

If a user is able to run some commands with sudo, they can likely find a way to escalate privileges to root.

Check what sudo commands a user has access to:

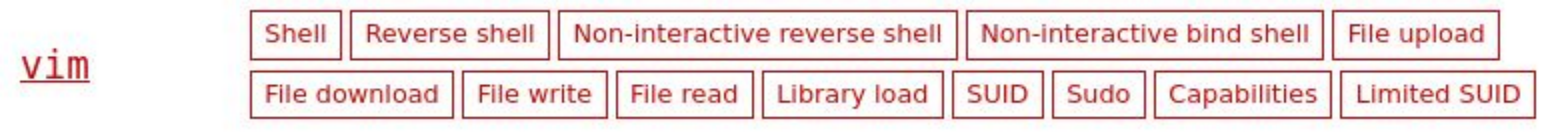
```
sudo -l
```

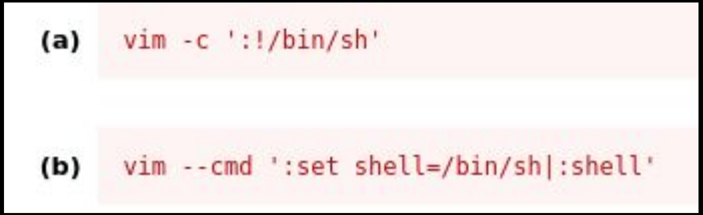
A reference for how to escalate to root with each command:

- https://gtfobins.github.io/

Privesc - Shell Escape Sequence

Example: Vim




(a)

```
vim -c ':!/bin/sh'
```

(b)

```
vim --cmd ':set shell=/bin/sh|:shell'
```

Privesc - LD_PRELOAD and LD_LIBRARY_PATH

Sudo can inherit environment variables from the user. Check the env_keep options with sudo -l

LD_PRELOAD - when a program is run, load this shared object first

LD_LIBRARY_PATH - A list of where the system searches for shared libraries

Privesc - LD_PRELOAD and LD_LIBRARY_PATH

If LD_PRELOAD is inherited:

Create a malicious shared object file that does anything you want, for example, spawn a shell

Run a program with sudo and set this environment variable to your shared object

```
sudo LD_PRELOAD=/path/to/newobj.so command
```

Privesc - LD_PRELOAD and LD_LIBRARY_PATH

If LD_LIBRARY_PATH is inherited:

Check what libraries are used by the command you are running

```
ldd command
```

Name your .so file the same name as one of these

Run a program with sudo and set this environment variable to the folder your file is in

```
sudo LD_LIBRARY_PARH=/path/of/file/ command
```

Privesc - SUID

SUID bit allows a script to be ran in the context of the owner.

```
> ls -la
drwxr-xr-x  qu3ri qu3ri .
drwxr-x---  qu3ri qu3ri ..
.rwSr--r--  root  root  script.sh
```

Find files with the SUID bit set:

`find -type f -perm -4000 2>/dev/null`

Some files are supposed to have this bit set, look for weird ones

Crontabs

See what cronjobs are running on the system.

`crontab -e`, `cat /etc/crontab`, `cat /var/spool/cron/crontab/user`

If any cronjobs run scripts, check if the script is writable

- If it is a root crontab, create reverse shell

Reporting

Report

Value for the client

Sections

- Executive Summary
- Overview - Scope, Methodology, Impact and Risk Charts
- Response Plan
- Attack Narrative/Timeline
- Findings

Components of a Finding

Basic information - Title, Risk, CVSS, MITRE ATT&CK

Description - What is the vulnerability, environment specific

Impact - If exploited, what could happen?

Recommendation/Remediation - Suggestions to mitigate

Replication - Details so client can validate

Questions?