

# **Final Engagement**

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

**PRESENTED BY: SAHEED LASISI, SAM SUR AND JOE JONES**

# Table of Contents

---

This document contains the following resources:

01



02

## Table of Contents

- **Offense**
- **Defense**
- **Network**

03





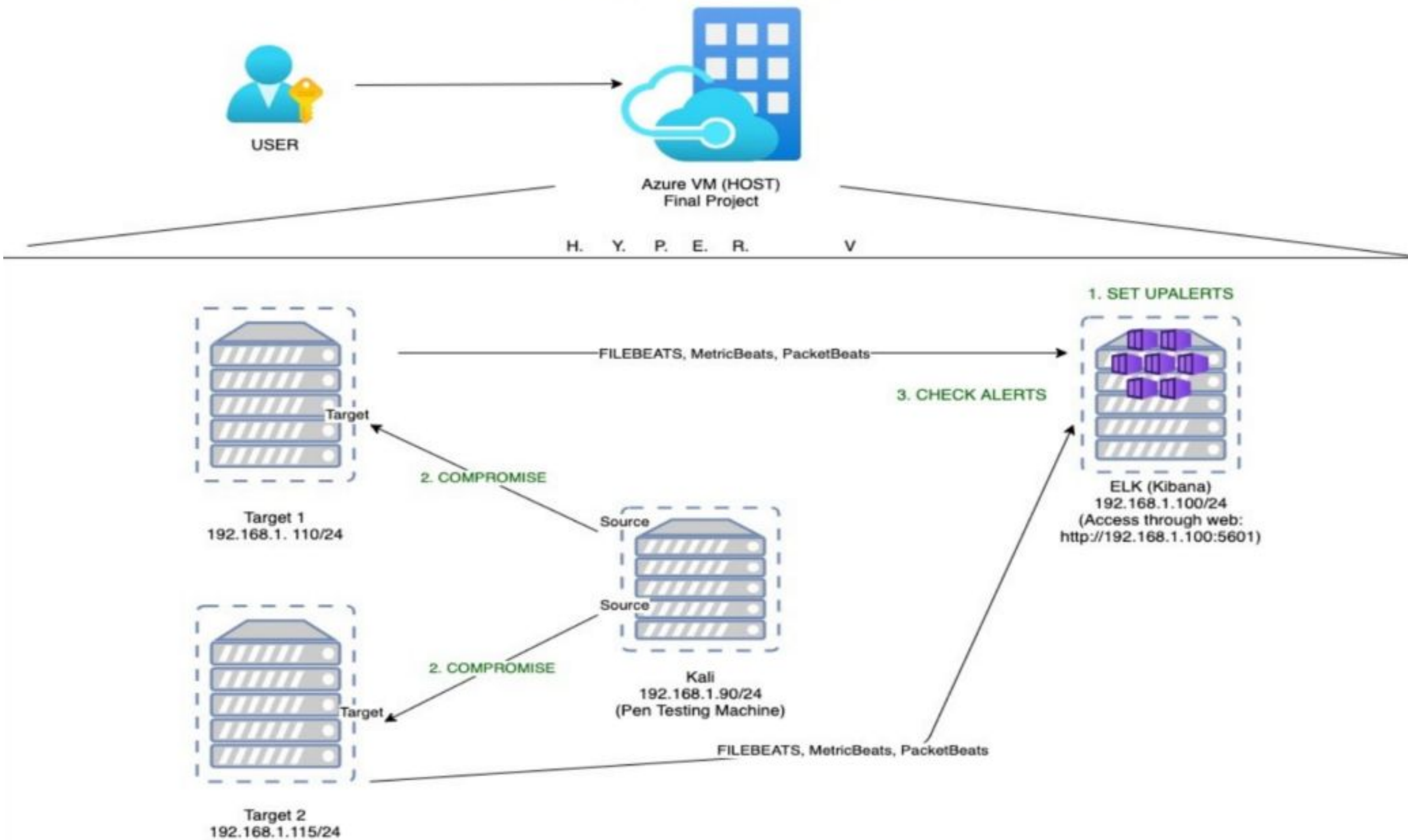
Offensive

# Network Topology & Critical Vulnerabilities



# Network Topology

Final Project - 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: Linux  
Hostname: ELK

IPv4: 192.168.1.90  
OS: Linux  
Hostname: Kali

IPv4: 192.168.1.110  
OS: Linux  
Hostname: Target 1

IPv4: 192.168.1.115  
OS: Linux  
Hostname: Target 2

# Critical Vulnerabilities: Target 1

---

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

Vulnerability	Description	Impact
Exposed WordPress Credentials	Running a scan can enumerate usernames.	Reveals user's credentials
Weak Passwords	Users that have weak passwords can give access to malicious actors	lateral Movement
WordPress XML RPC Ping API Pingback locator	Allow attackers to send request to internet servers for port-scanning attack	One server can expose all the internal server composition

# Critical Vulnerabilities: Target 2

---

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

Vulnerability	Description	Impact
WordPress User Enumeration	WPScan detect the list of users with the specific options used (-u)	Anyone can brute force the authentication for the system
open port 22 SSH & Weak Password	Having port 22 SSH open anyone with the username and password can get into the system	Anyone can brute force the authentication for the system
Python sudo privileges	User is given access to sudo privileges via python	Attacker can escalate to root privileges easily gaining access to the system



# Exploits Used



# Exploitation: Wordpress User Enumeration

- WPScan was used to find the 2 users: Michael and Steven wpscan
  - -url <http://192.168.1.110/wordpress> -eu

```
Shell No. 1
File Actions Edit View Help
:01
[+] 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 https://wpvulndb.com/users/sign_up
[+] Finished: Tue Sep 7 15:28:45 2021
[+] Requests Done: 64
[+] Cached Requests: 4
[+] Data Sent: 12.834 KB
[+] Data Received: 17.287 MB
[+] Memory used: 133.434 MB
[+] Elapsed time: 00:00:02
root@Kali:~#
```



# Exploitation: Open Port 22 SSH & Weak Password

---

Hydra was used to get Michael's password

```
File  Actions  Edit  View  Help
root@Kali:~# hydra -l michael -P /usr/share/wordlists/rockyou.txt
.168.1.110
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in milit
cret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 202
5:54:08
[WARNING] Many SSH configurations limit the number of parallel tas
recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login
:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://192.168.1.110:22/
[22][ssh] host: 192.168.1.110  login: michael  password: michael
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 202
5:54:15
root@Kali:~#
```



# Exploitation: Sensitive Data Exposure

- Discovered the wordpress directory there is a wp-config.php file. found the username is root and the password is R@v3nSecurity

```
michael@target1: /var/www/html/wordpress
File Actions Edit View Help
* * Database table prefix
* * ABSPATH
*
* @link https://codex.wordpress.org/Editing_wp-config.php
*
* @package WordPress
*/

// ** 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');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');
```



# Avoiding Detection



# Stealth Exploitation of (Wordpress User Enumeration)

---

## Monitoring Overview

- this alert detection was used this exploit:
- WHEN count () GROUPED OVER top 5  
"http.response\_status\_code IS ABOVE 400 FOR THE LAST 5 minutes"
- packetbeat-http.response.status\_code metrics was used
- above 400 FOR THE LAST 5 MINUTES threshold was fired at.

## Mitigating Detection

- You can disable User Enumeration in Wordpress by using free plugin WP" Hardening
- install and activate plugin> 'Security Fixers' tab> stop user enumeration
- Underline factor is to educate the employees to use the stronger passwords for security purposes. (not using the same pw as id like michael)
- company should also imply a policy for password change every 3-6 months.

# Stealth Exploitation of [Open Ports 22 SSH]

---

## Monitoring Overview

- SSH Login alert would detect this exploit
- Monitor SSH Port for unauthorized access
- Triggers when user attempts to access system over Port 22

## Mitigating Detection

- the best mitigation for this exploit would be closing the port 22
- However there might be times when you would need Port 22 open.
- if this is the case we can create whitelisting of IPs so that only authorized can access the Port 22

# Stealth Exploitation of [Sensitive Data Exposure]

---

## **Monitoring Overview**

- Alert when SQL Database has gained access with unauthorized personnel.
- Triggers external/unauthorized IP connections that are made to the SQL database or any other sensitive files (like wp-config.php file)

## **Mitigating Detection**

- Have all the sensitive data encrypted, not in plaintext
- this will have the data not easily accessible
- only whitelist the IPs that are being logged in for SQL database (ex. Administrators)



# Defensive



# Alerts Implemented

# HTTP Request Size Monitor

- http.request.bytes
- Is above 3500 bytes for the last 1 minute
- Detect HTTP request smuggling

### Edit HTTP Request Size Monitor

Send an alert when your specified condition is met. Your watch will run every 1 minute.

**Name**  
HTTP Request Size Monitor

**Indices to query**  
packstbeat-\* x

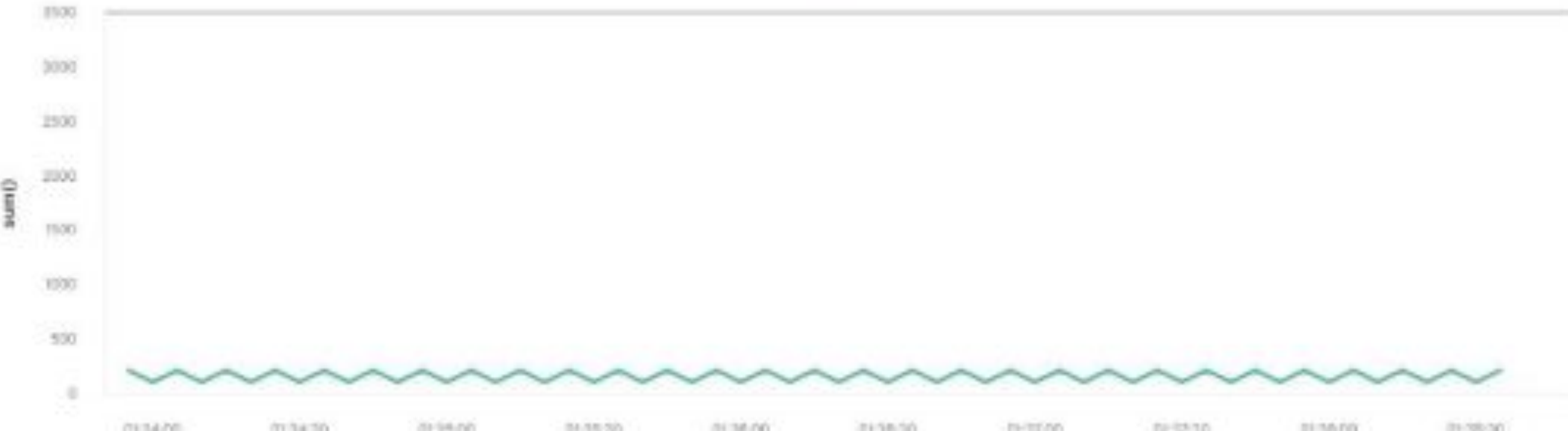
**Time field**  
@timestamp

**Run watch every**  
1 minute


Use \* to broaden your query.

Match the following condition

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



Perform 1 action when condition is met

>  Logging

✓ Save alert

Cancel

Show request

# CPU Usage Monitor

- `system.process.cpu.total`
- Is above 50% for the last 5 minutes
- DoS Attack or detection of software causing excessive system usage

### Edit CPU Usage Monitor

Send an alert when your specified condition is met. Your watch will run every 5 minutes.

Name

CPU Usage Monitor

Indices to query

metricbeat-\* \*

Time field

@timestamp

Run watch every

5

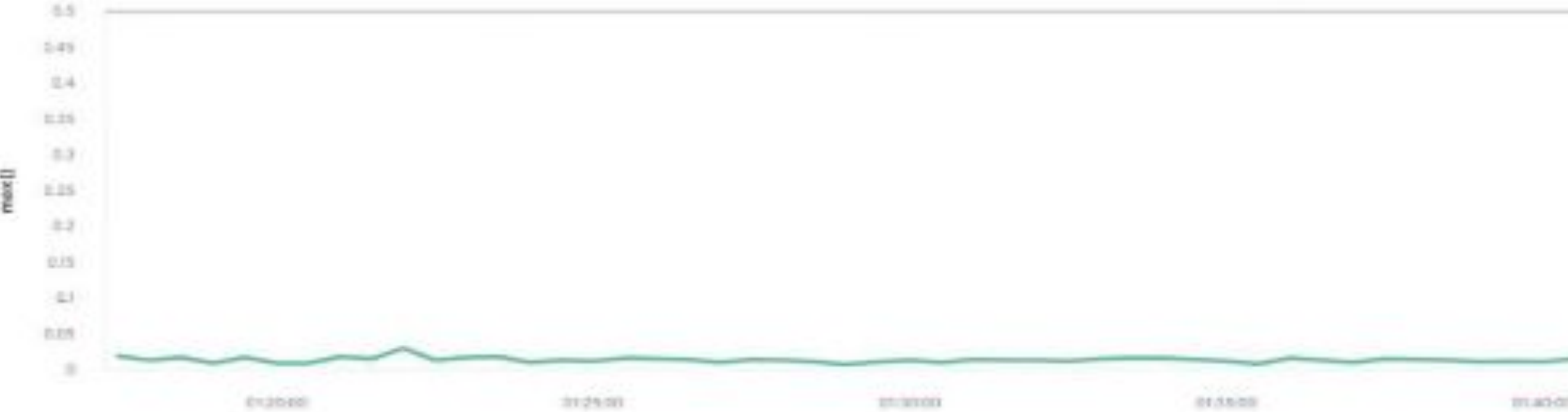
minutes

Use \* to broaden your query.


Match the following condition

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

max()



Perform 1 action when condition is met

>  Logging

✓ Save alert

Cancel



# Excessive HTTP Error Codes

- `http.response.status_code`
- One of the top 5 status codes is above 400 for the last 5 min
- Brute Force Attack detection  
DoS Attack  
Attacks that would influence number of hits on the website

## Create threshold alert

Send an alert when your specified condition is met. Your watch will run every 5 minutes.

Name

Excessive HTTP errors

Indices to query

packetbeat-\* X

Use \* to broaden your query.

Time field

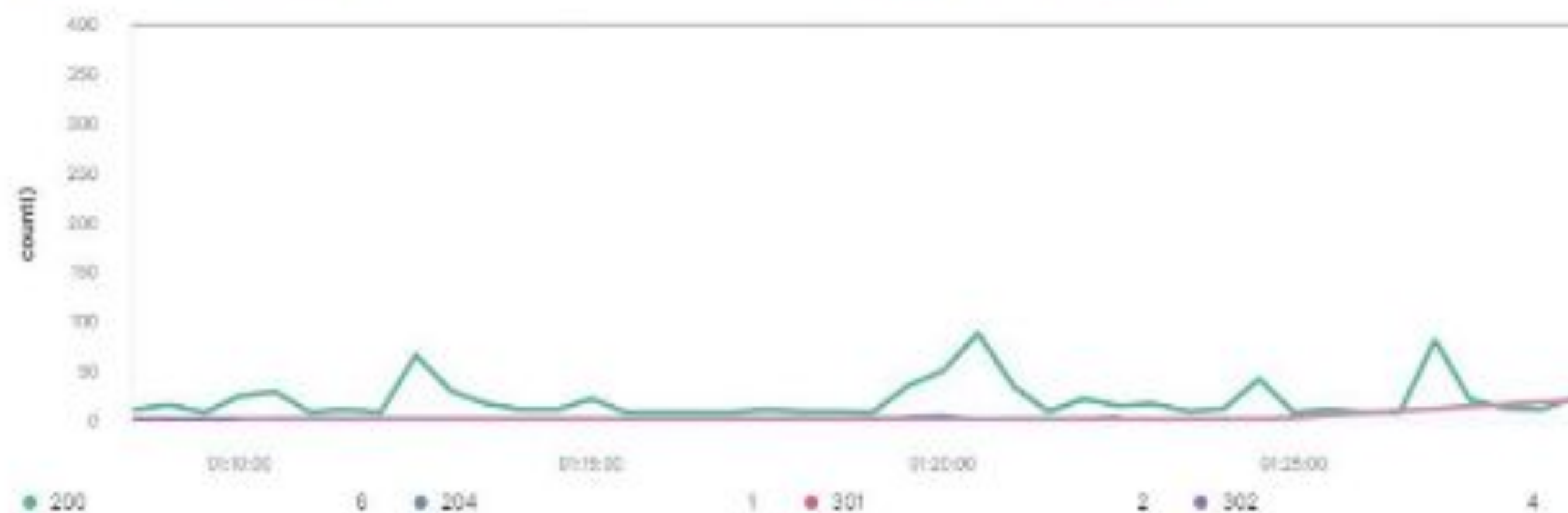
@timestamp

Run watch every

5

Match the following condition

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



Perform 0 actions when condition is met

✓ Create alert

Cancel



# Hardening

# Hardening Against Weak Passwords on Target 1

---

If users are forced to create more complex passwords, they could easily be orders of magnitude more difficult to guess or brute force, effectively stopping brute force attacks from being a viable attack vector.

Executing the following steps will force users to make more complex passwords.

1. Use admin account
2. run "sudo apt-get install libpam-cracklib"
3. run "sudo nano /etc/pam.d/common-password"
4. There will be a line that reads "password requisite pam\_cracklib.so retry=3 minlen=8 difok=3"
5. Edit it to "password requisite pam\_cracklib.so try\_first\_pass retry=3 minlength=12 credit=1 credit=1 credit=1 credit=1 reject username"
6. This will require passwords to be 12 characters long and include, 1 uppercase letter, 1 lowercase letter, 1 digit, and 1 other character. Also, the password can no longer be the same as the username.



# Hardening Against Privilege Escalation via Python on Target 1

If Stevens sudo privileges related to python are removed, malicious actors won't be able to gain access to the root shell via python after accessing Stevens account.

Executing the following steps will force users to make more complex passwords.

1. Use admin account
2. Run "sudo visudo"
3. Delete the line that reads "steven ALL=(ALL) NOPASSWD: /usr/bin/python"
4. Save and exit the document with "ctrl + x"

```
GNU nano 2.2.6      File: /etc/sudoers.tmp

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL) NOPASSWD:ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d

Steven ALL=(ALL) NOPASSWD: /usr/bin/python
```



# Hardening Against MySQL Data Breach on Target 1

Michael was able to read the wp-config.php file to learn the password. By restricting read and write access from “other users”, we can prevent unauthorized accounts from using credentials to access the MySQL database.

Executing the following steps will force users to make more complex passwords.

1. Use admin account
2. Run “sudo chmod 660 /var/www/html/wordpress/wp-config.php”

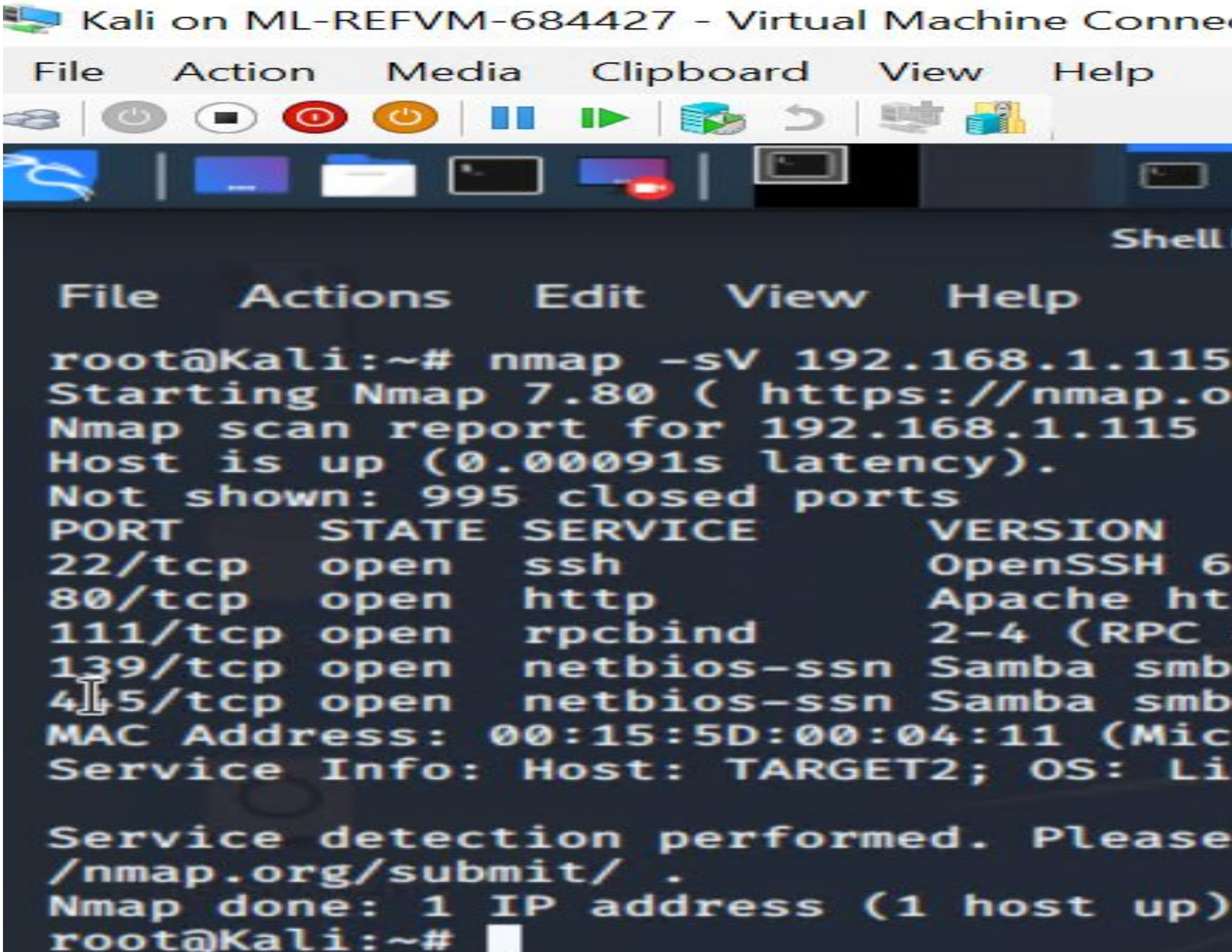
```
michael@target1:/var/www/html/wordpress$ ls -l
total 192
-rwxrwxrwx 1 root root 418 Sep 25 2013 index.php
-rwxrwxrwx 1 root root 19935 Aug 13 2018 license.txt
-rwxrwxrwx 1 root root 7413 Nov 23 13:35 readme.html
-rwxrwxrwx 1 root root 6864 Nov 23 13:35 wp-activate.php
drwxrwxrwx 9 root root 4096 Jun 15 2017 wp-admin
-rwxrwxrwx 1 root root 364 Dec 19 2015 wp-blog-header.php
-rwxrwxrwx 1 root root 1627 Aug 29 2016 wp-comments-post.php
-rw-rw-rw- 1 www-data www-data 3134 Aug 13 2018 wp-config.php
-rwxrwxrwx 1 root root 2853 Dec 16 2015 wp-config-sample.php
drwxrwxrwx 6 root root 4096 Nov 25 12:49 wp-content
-rwxrwxrwx 1 root root 3286 May 24 2015 wp-cron.php
drwxrwxrwx 18 root root 12288 Jun 15 2017 wp-includes
-rwxrwxrwx 1 root root 2422 Nov 21 2016 wp-links-opml.php
-rwxrwxrwx 1 root root 3301 Oct 25 2016 wp-load.php
-rwxrwxrwx 1 root root 34347 Nov 23 13:35 wp-login.php
-rwxrwxrwx 1 root root 8048 Jan 11 2017 wp-mail.php
-rwxrwxrwx 1 root root 16200 Apr 6 2017 wp-settings.php
-rwxrwxrwx 1 root root 29924 Jan 24 2017 wp-signup.php
-rwxrwxrwx 1 root root 4513 Oct 16 2016 wp-trackback.php
```

```
root@target1:/var/www/html/wordpress# sudo chmod 660 /var/www/html/wordpress/wp-config.php
root@target1:/var/www/html/wordpress# ls -l
total 192
-rwxrwxrwx 1 root root 418 Sep 25 2013 index.php
-rwxrwxrwx 1 root root 19935 Aug 13 2018 license.txt
-rwxrwxrwx 1 root root 7413 Nov 23 13:35 readme.html
-rwxrwxrwx 1 root root 6864 Nov 23 13:35 wp-activate.php
drwxrwxrwx 9 root root 4096 Jun 15 2017 wp-admin
-rwxrwxrwx 1 root root 364 Dec 19 2015 wp-blog-header.php
-rwxrwxrwx 1 root root 1627 Aug 29 2016 wp-comments-post.php
-rw-rw---- 1 www-data www-data 3134 Aug 13 2018 wp-config.php
-rwxrwxrwx 1 root root 2853 Dec 16 2015 wp-config-sample.php
drwxrwxrwx 6 root root 4096 Nov 25 12:49 wp-content
-rwxrwxrwx 1 root root 3286 May 24 2015 wp-cron.php
drwxrwxrwx 18 root root 12288 Jun 15 2017 wp-includes
-rwxrwxrwx 1 root root 2422 Nov 21 2016 wp-links-opml.php
-rwxrwxrwx 1 root root 3301 Oct 25 2016 wp-load.php
-rwxrwxrwx 1 root root 34347 Nov 23 13:35 wp-login.php
-rwxrwxrwx 1 root root 8048 Jan 11 2017 wp-mail.php
-rwxrwxrwx 1 root root 16200 Apr 6 2017 wp-settings.php
-rwxrwxrwx 1 root root 29924 Jan 24 2017 wp-signup.php
-rwxrwxrwx 1 root root 4513 Oct 16 2016 wp-trackback.php
```



# Hardening Against Unsecured/Open Ports on Target 2

- Whitelist known, safe IP's for SSL  
Using firewall or similar programs will allow for ease in implementing
- Close port 80 HTTP and open port 443 HTTPS for more secure/encrypted traffic



```
Kali on ML-REFVM-684427 - Virtual Machine Connected
File Action Media Clipboard View Help
root@Kali:~# nmap -sV 192.168.1.115
Starting Nmap 7.80 ( https://nmap.org )
Nmap scan report for 192.168.1.115
Host is up (0.00091s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 6.6p1
80/tcp    open  http         Apache/2.4.18
111/tcp   open  rpcbind      2-4 (RPC)
139/tcp   open  netbios-ssn  Samba smbd
445/tcp   open  netbios-ssn  Samba smbd
MAC Address: 00:15:5D:00:04:11 (Mikrotik)
Service Info: Host: TARGET2; OS: Linux

Service detection performed. Please see the Nmap project page at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up)
root@Kali:~#
```



# Hardening Against Directory Listing on Target 2

- 1) Securing the directories prevents unauthorized users from accessing any sensitive information stored here.
- 2) Store the data internally and not directly to the site to avoid easier
- 3) using authentication to access said files
- 4) change privilege to a Roll-Based privilege system

```
View Help
http://192.168.1.115 /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
dir -u http://192.168.1.115 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
=====
onial) & Christian Mehlmauer (@_FireFart_)
=====
http://192.168.1.115
0
usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
00,204,301,302,307,401,403
obuster/3.0.1 2018-08-13 07:56 26K
0s
=====
starting gobuster
=====
01) 2018-08-13 07:56 13K
2018-08-13 07:56 2.3K
2018-08-13 07:56 6
```



	<a href="#">Parent Directory</a>	
	<a href="#">LICENSE</a>	2
	<a href="#">PATH</a>	2
	<a href="#">PHPMailerAutoload.php</a>	2
	<a href="#">README.md</a>	2
	<a href="#">SECURITY.md</a>	2
	<a href="#">VERSION</a>	2
	<a href="#">changelog.md</a>	2
	<a href="#">class.phpmailer.php</a>	2
	<a href="#">class.phpmaileroauth.php</a>	2
	<a href="#">class.phpmaileroauthgoogle.php</a>	2
	<a href="#">class.pop3.php</a>	2
	<a href="#">class.smtp.php</a>	2
	<a href="#">composer.json</a>	2
	<a href="#">composer.lock</a>	2
	<a href="#">docs/</a>	2
	<a href="#">examples/</a>	2
	<a href="#">extras/</a>	2
	<a href="#">get_oauth_token.php</a>	2
	<a href="#">language/</a>	2
	<a href="#">test/</a>	2
	<a href="#">travis.phpunit.xml.dist</a>	2



# Hardening Against Vulnerable Software on Target 2

---

- Updating to a more recent and more stable version of the software.
- Newer software updates include know patches for exiting vulnerabilities.

syntax: apt-get update <software\_name>

apt-get upgrade <software\_name>



```
root@kali:~# apt-get update && apt-get upgrade
```

# Implementing Patches



<https://blog.dbi-services.com/automating-linux-patching-with-ansible/>

Ansible playbooks for upgrading and patching would scan installed packages for upgrades and patches, then run upgrade, patches, and print any errors from upgrade/patching.

```

1  ---
2  - name: Get packages that can be upgraded
3    become: yes
4    ansible.builtin.dnf:
5      list: upgrades
6      state: latest
7      update_cache: yes
8      register: reg_dnf_output_all
9      when: ev_security_only == "no"
10
11 - name: List packages that can be upgraded
12   ansible.builtin.debug:
13     msg: "{{ reg_dnf_output_all.results | map(attribute='name') | list }}"
14   when: ev_security_only == "no"
15
16
17 - name: Get packages that can be patched with security fixes
18   become: yes
19   ansible.builtin.dnf:
20     security: yes
21     list: updates
22     state: latest
23     update_cache: yes
24     register: reg_dnf_output_secu
25     when: ev_security_only == "yes"
26
27 - name: List packages that can be patched with security fixes
28   ansible.builtin.debug:
29     msg: "{{ reg_dnf_output_secu.results | map(attribute='name') | list }}"
30   when: ev_security_only == "yes"
31
32
33 - name: Request user confirmation
34   ansible.builtin.pause:
35     prompt: |
36
37       The packages listed above will be upgraded. Do you want to continue ?
38       -> Press RETURN to continue.
39       -> Press Ctrl+c and then "a" to abort.
40   when: reg_dnf_output_all is defined or reg_dnf_output_secu is defined
  
```

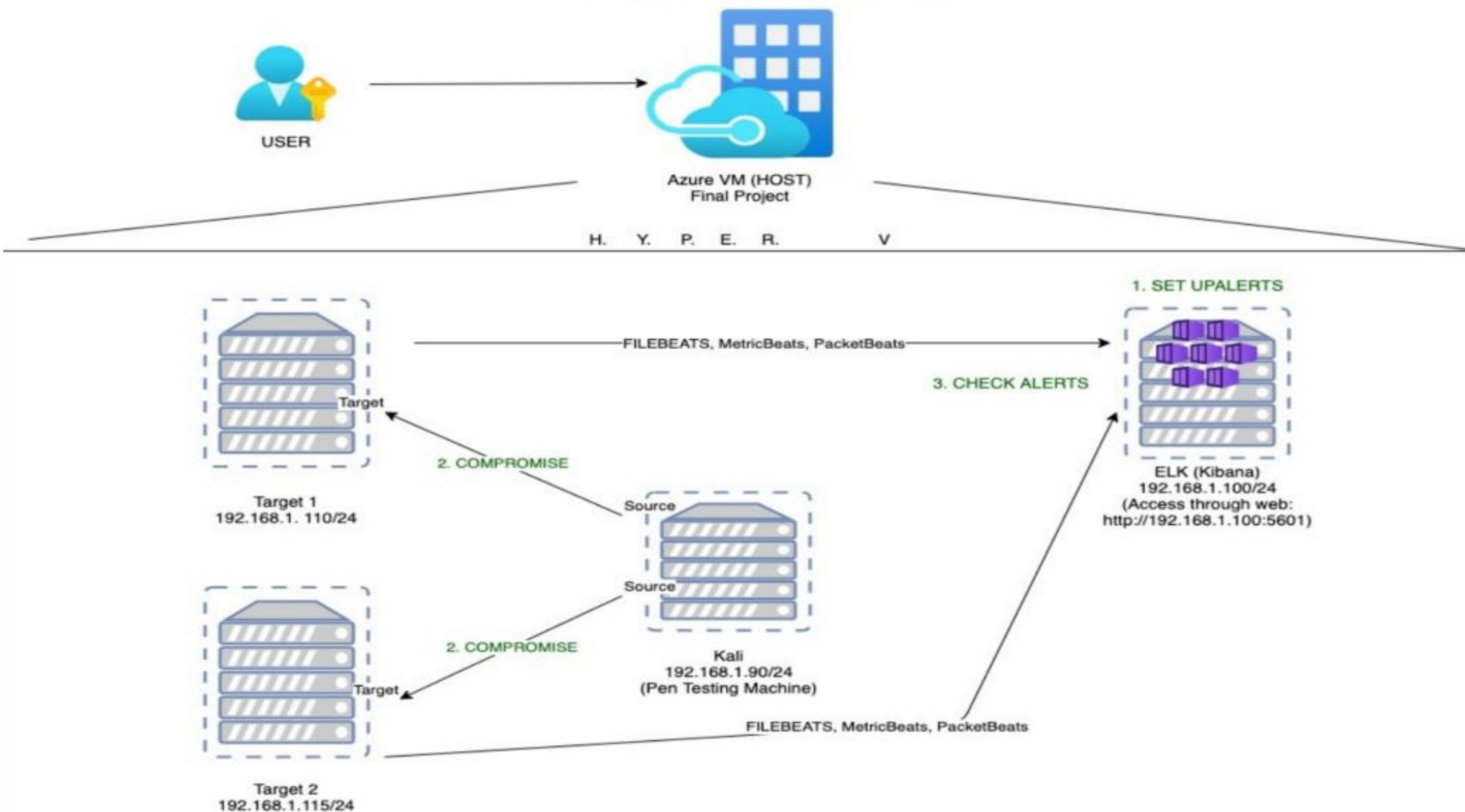


# Networking

# Network Topology & Critical Vulnerabilities

# Network Topology

Final Project - 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: Linux  
Hostname: ELK

IPv4: 192.168.1.90  
OS: Linux  
Hostname: Kali

IPv4: 192.168.1.110  
OS: Linux  
Hostname: Target 1

IPv4: 192.168.1.115  
OS: Linux  
Hostname: Target 2

# Critical Vulnerabilities: Target 1

---

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

Vulnerability	Description	Impact
Open Access to ports	Easy access to open 22 and 80	unsecure common ports facilitate brute force attack
enumerating usernames in Wordpress	Easy identification of usernames on system	Credentials provide easy acces for attackers
Simple password	passwords lacking variation or length	passwords can easily be brute forced or guessed
root vulnerability with python	Allow local users to gain privilege via python script	permits full administrative access



# Critical Vulnerabilities: Target 2

---

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

Vulnerability	Description	Impact
unsecured/open ports	SSH(22), HTTP (80)	Remote access via SSH and unsecured traffic between requests.
Directory Listing	Public Facing Directories	Access to sensitive information
Un-patched/Vulnerable software	Outdated vulnerable software is being used	Contains vulnerabilities patched by newer versions

# Traffic Profile

# Traffic Profile

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

Feature	Value	Description
Top Talkers (IP Addresses)	172.16.4.205 49.36% 185.243.115.84 29.16% 10.0.0.201 18.74% 	Machines that sent the most traffic.
Most Common Protocols	TCP 88.5% UDP 11.2% ARP 0.2% 	Three most common protocols on network.
# of Unique IP Addresses	IPv4 808 IPv6 2 	Count of observed IP addresses.
Subnets	172.16.4.0/24 185.243.115.0/24 10.0.0.0/24	Observed subnet ranges.
# of Malware Species		Number of malware binaries identified in traffic.
	2	



# Behavioral Analysis

## Purpose of Traffic on the Network

Users were observed engaging in the following kinds of activity.

### “Normal” Activity

- Watching Youtube
- Browsing medical information
- Shopping for toys and records

### Suspicious Activity

- Creating of server on corporate network
- Torrenting
- Malware transmission



# Normal Activity



# Streaming Video Analysis

Streaming Video from Frank-n-Ted.com  
UDP port 53

10.6.12.0/24 and udp.port == 53

Packet list

Narrow & Wide

☐ Case sensitive

String

frank

Find

Cancel

Name	Source	Destination	Protocol	Length	Info
0-06-30 17:04:22.856430400	10.6.12.157	10.6.12.12	DNS	132	Standard query 0x79df SRV _ldap._tcp.Default-First-Site-Name._
0-06-30 17:04:22.859599900	10.6.12.12	10.6.12.157	DNS	198	Standard query response 0x79df SRV _ldap._tcp.Default-First-Si
0-06-30 17:04:23.162542400	10.6.12.157	10.6.12.12	DNS	117	Standard query 0xde86 SRV _ldap._tcp.Default-First-Site-Name._
0-06-30 17:04:23.165134200	10.6.12.12	10.6.12.157	DNS	183	Standard query response 0xde86 SRV _ldap._tcp.Default-First-Si
0-06-30 17:04:23.395691900	10.6.12.157	10.6.12.12	DNS	88	Standard query 0x4133 A ygrvqkgouzou.frank-n-ted.com
0-06-30 17:04:23.398331200	10.6.12.12	10.6.12.157	DNS	165	Standard query response 0x4133 No such name A ygrvqkgouzou.fra
0-06-30 17:04:23.492265800	10.6.12.157	10.6.12.12	DNS	90	Standard query 0xde17 A Frank-n-Ted-DC.frank-n-ted.com

132 bytes on wire (1056 bits), 132 bytes captured (1056 bits) on interface eth0, id 0

Src: Intel\_68:42:d3 (00:11:75:68:42:d3), Dst: Dell\_2a:f7:e5 (98:40:bb:2a:f7:e5)

on: Dell\_2a:f7:e5 (98:40:bb:2a:f7:e5)

Intel\_68:42:d3 (00:11:75:68:42:d3)

4 (0x0800)

Protocol Version 4, Src: 10.6.12.157, Dst: 10.6.12.12

m Protocol, Src Port: 50198, Dst Port: 53

rt: 50198

on Port: 53

8

0xdfea [unverified]

a Status: Unverified]

ndex: 1150]

ps]

System (query)

on ID: 0x79df

0100 Standard query

: 1

ts: 0

r RRs: 0

l RRs: 0

\_tcp.Default-First-Site-Name.\_sites.ForestDnsZones.frank-n-ted.com: type SRV, class IN

In: 55607]



# Searching for Record Information

Browsing [www.vinylmeplease.com](http://www.vinylmeplease.com) for record information

HTTP port 80

http.request.uri contains "vinyl"

Packet list   Narrow & Wide   ☐ Case sensitive   String   warped   Find   Cancel

No.	Time	Source	Destination	Protocol	Length	Info
46007	2020-06-30 17:03:00.448490800	10.11.11.200	13.33.255.37	HTTP	502	GET /magazine/guide-to-flattening-warped-vinyl-records/ HTTP/1.1
50145	2020-06-30 17:03:50.147885200	10.11.11.200	13.33.255.31	HTTP	486	GET /widgets/configs/site-c34415b4-vinylmeplease.com.json HTTP/1.1
51091	2020-06-30 17:03:57.290983500	10.11.11.200	172.217.9.134	HTTP	614	GET /activityi;src=8704410;type=retar0;cat=vmp_r0;ord=6577035978844;gt HTTP/1.1
51433	2020-06-30 17:04:00.216086700	10.11.11.200	13.33.255.31	HTTP	486	GET /widgets/configs/site-c34415b4-vinylmeplease.com.json HTTP/1.1

Frame 46007: 502 bytes on wire (4016 bits), 502 bytes captured (4016 bits) on interface eth0, id 0

- Ethernet II, Src: Dell\_8a:50:a9 (84:8f:69:8a:50:a9), Dst: Cisco\_97:4b:f0 (00:01:c9:97:4b:f0)
  - Destination: Cisco\_97:4b:f0 (00:01:c9:97:4b:f0)
  - Source: Dell\_8a:50:a9 (84:8f:69:8a:50:a9)
  - Type: IPv4 (0x0800)
- Internet Protocol Version 4, Src: 10.11.11.200, Dst: 13.33.255.37
- Transmission Control Protocol, Src Port: 49198, Dst Port: 80, Seq: 1, Ack: 1, Len: 448
- Hypertext Transfer Protocol
  - GET /magazine/guide-to-flattening-warped-vinyl-records/ HTTP/1.1\r\n
  - Accept: text/html, application/xhtml+xml, \*/\*\r\n
  - Accept-Language: en-US\r\n
  - User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko\r\n
  - Accept-Encoding: gzip, deflate\r\n
  - Host: www.vinylmeplease.com\r\n
  - DNT: 1\r\n
  - Connection: Keep-Alive\r\n
  - Cookie: \_gcl\_au=1.1.949876142.1573510598; \_ga=GA1.2.2082810095.1573510598; \_gid=GA1.2.1906447370.1573510598; \_\_zlcmid=vDigFj7IpbARDG\r\n\r\n
  - [Full request URI: <http://www.vinylmeplease.com/magazine/guide-to-flattening-warped-vinyl-records/>]
  - [HTTP request 1/3]
  - [Response in frame: 46078]
  - [Next request in frame: 46081]



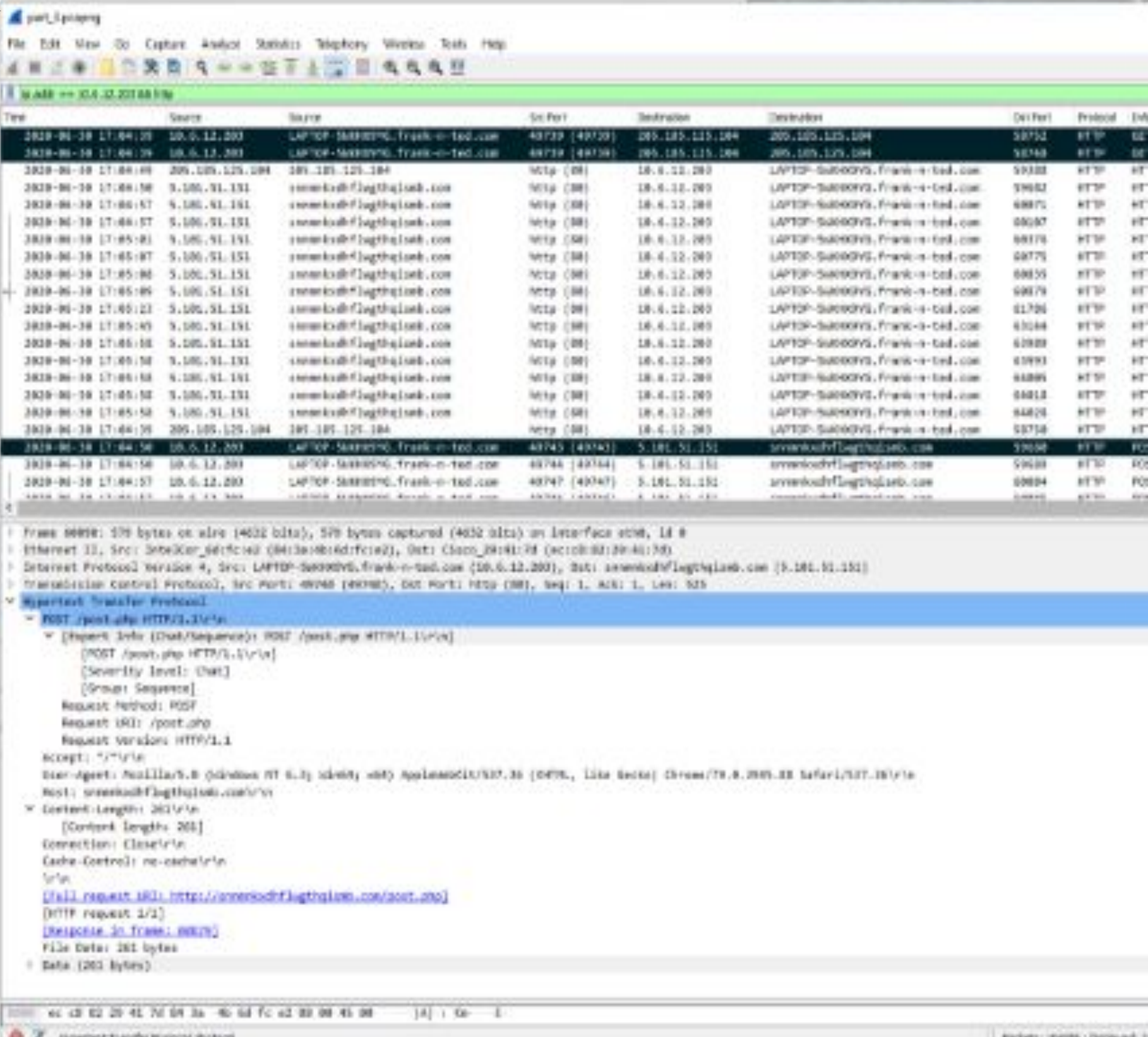
# Malicious Activity

# Zloader RAT Download

[frank-n-ted.com](http://frank-n-ted.com) (10.6.12.203) downloaded a file  
malicious file from 205.185.125.104  
HTTP GET requests were made for:  
pQBtWi  
june11.dll

This is commonly associated with an Excel macro  
june11.dll is a RAT  
It POSTS to the host [snnmnkxdhflwgthqismb.com](http://snnmnkxdhflwgthqismb.com)  
(5.101.51.151)

[snnmnkxdhflwgthqismb.com](http://snnmnkxdhflwgthqismb.com) is a C2 site for the  
ZLoader RAT



The image shows a Wireshark packet capture of an HTTP POST request. The packet list on the left shows a POST request from 10.6.12.203 to 5.101.51.151 on port 80. The packet details pane on the right shows the structure of the POST request, including the method, URI, host, and content length. The packet bytes pane at the bottom shows the raw data of the request.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.6.12.203	5.101.51.151	HTTP	151	POST /post.php HTTP/1.1
2	0.000000	5.101.51.151	10.6.12.203	HTTP	151	200 OK

Packet 1: POST /post.php HTTP/1.1

Request Method: POST  
Request URI: /post.php  
Request Version: HTTP/1.1

Host: snnmnkxdhflwgthqismb.com

Content-Length: 261

Connection: close

Cache-Control: no-cache

File Data: 261 bytes



# NetSupport RAT Download

## This is a Remote Access Trojan - NetSupport RAT

Visit <http://green.mattingsolutions.co> (185.243.115.84)

This is a known infected site

Likely a fake web browser update

POST request to 185.243.115.84 included

501 ASCII hexadecimal data files

empty.gif

2 screen shots of the infected user's desktop

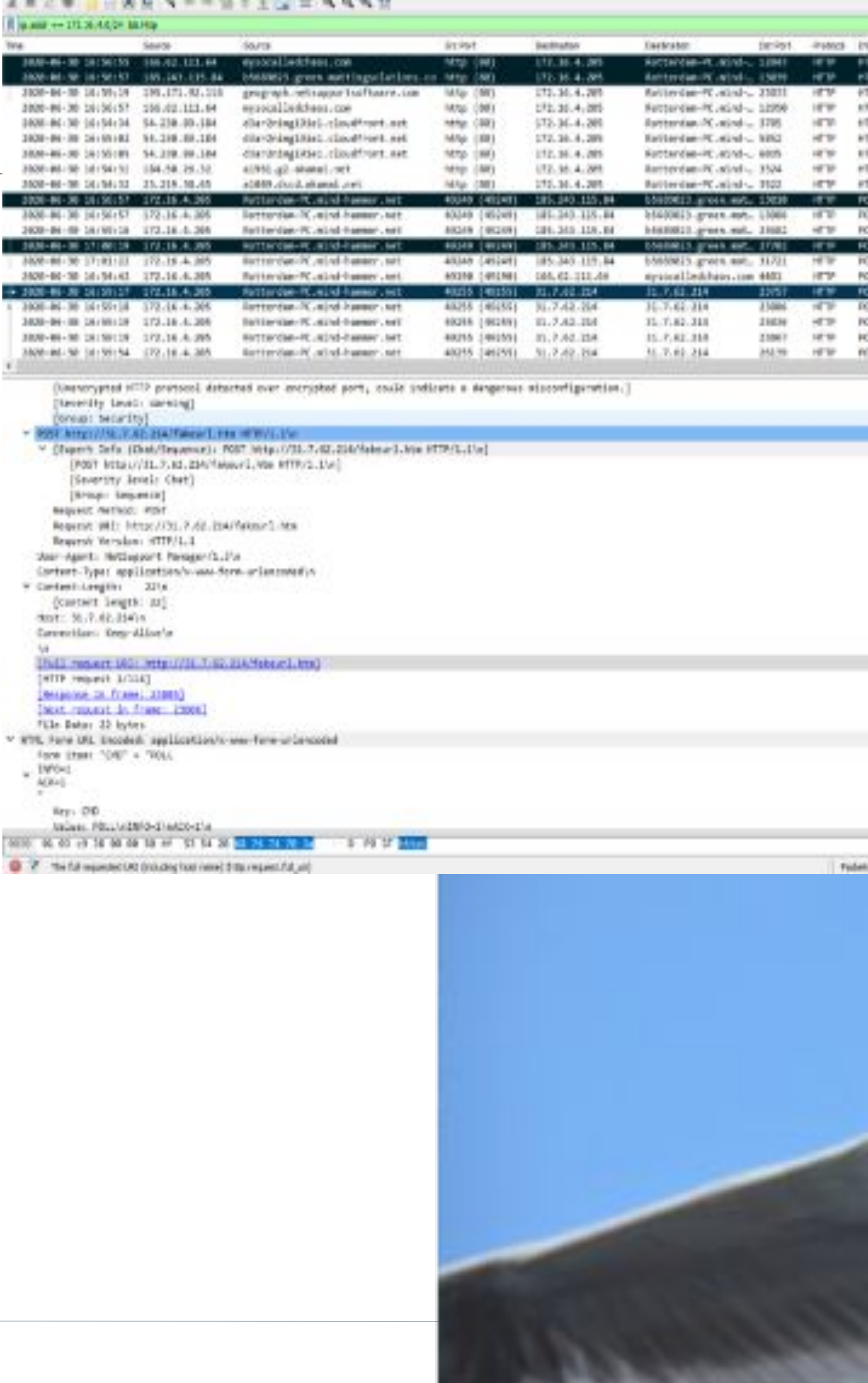
empty.gif?ss&ss1.img and empty.gif?ss&ss2.img

POST requests to <http://31.7.62.214/fakeurl.htm>

114 application/x-www-form-urlencoded

fakeurl.htm

This file name is associated the the Netsup  
RAT





# The End