Network Vulnerability Assessment
"Its the End of the Assessment as We Know It, and I Feel Fine"

Phase 1: "I'd like to Teach the World to *Ping* "

You have been provided a list of network assets belonging to RockStar Corp. Use ***fping*** to ping the network assets for *only the Hollywood office.*

-Determine the IPs for the Hollywood office and run ***fping*** against the IP ranges in order to determine which IP is accepting connections.

-RockStar Corp doesn't want any of their servers, even if they are up, indicating that they are accepting connections.

-Use ***fping <IP Address>*** and ignore any results that say "Request timed out".
-If any of the IP addresses send back a Reply, enter Ctrl+C to stop sending requests.

Create a summary file in a word document that lists out the ***fping*** command used, as well as a summary of the results.

Your summary should determine which IPs are accepting connections and which are not.

Also indicate at which OSI layer your findings are found.

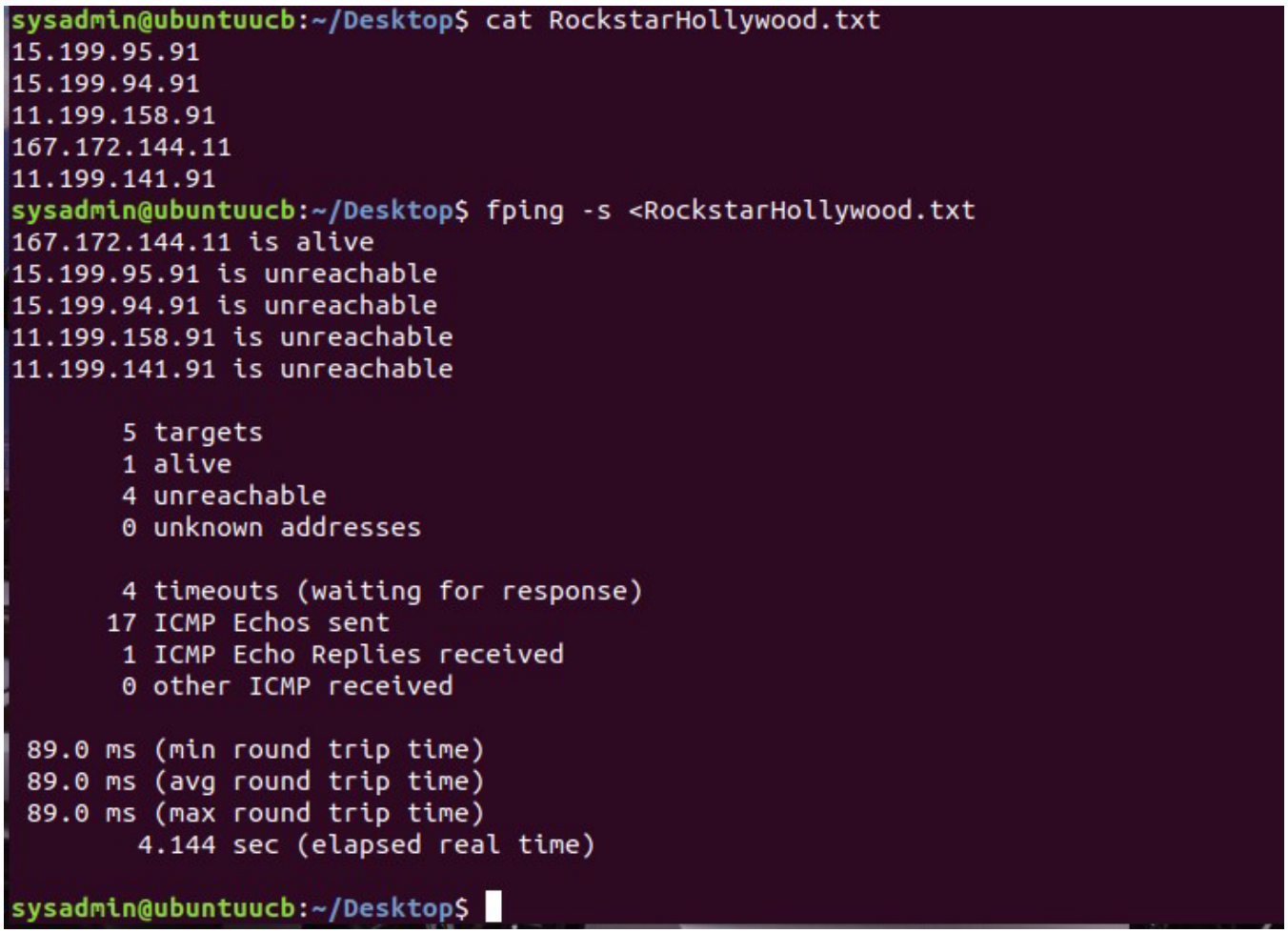
Determined the IP ranges to scan from the list were 167.172.144, 15.199.95.91, 15.199.94.91, 11.199.158.91, and 11.199.141.91, then ran fping with the -s option for a stats summary.

```
fping -s <IP Addresses typed out with stats summary>
```

```
fping -s 15.199.95.91 15.199.94.91 11.199.158.91 167.172.144.11 11.199.141.91
```

```
sysadmin@ubuntuucb:~/Desktop$ fping -s 15.199.95.91 15.199.94.91 11.199.158.91 167.172.144.11 11.199.141.91
167.172.144.11 is alive
15.199.95.91 is unreachable
15.199.94.91 is unreachable
11.199.158.91 is unreachable
11.199.141.91 is unreachable

       5 targets
       1 alive
       4 unreachable
       0 unknown addresses

       4 timeouts (waiting for response)
      17 ICMP Echos sent
       1 ICMP Echo Replies received
       0 other ICMP received

 93.4 ms (min round trip time)
 93.4 ms (avg round trip time)
 93.4 ms (max round trip time)
        4.126 sec (elapsed real time)

sysadmin@ubuntuucb:~/Desktop$
```

This can also be done with a text file containing the list of IP addresses to run in case you need to update or modify the list for any reason. In doing so, I will be using the same “fping -s” command (with stats summary option) and reading input from a text file of IP addresses in order to see ones that are reachable and unreachable. See the following screenshot.



Determined a potential vulnerability that the Hollywood Application Servers with IP 167.172.144.11 is alive and responding after running the scan.

Since Rockstar Corp doesnt want any of their servers indicating that they are accepting connections, this is a vulnerability for them.

Recommend to restrict allowing ICMP (Internet Control Message Protocol) echo requests against IP 167.172.144.11 to prevent successful responses from PING requests.

This occurred on layer 3 (Network Layer) of the OSI model as Ping uses IP addresses and IPs are used on the Network Layer.

Phase 2: "Some *Syn* for Nothin`"

The most popular SYN scan is used with the Nmap tool using the -sS option for a TCP SYN Scan. Since the scan type requires root privileges, I will be using the added “sudo” command.

`sudo nmap -sS 167.172.144.11`

```
sysadmin@ubuntuucb:~/Desktop$ nmap -sS 167.172.144.11
You requested a scan type which requires root privileges.
QUITTING!
sysadmin@ubuntuucb:~/Desktop$ sudo nmap -sS 167.172.144.11
[sudo] password for sysadmin:

Starting Nmap 7.60 ( https://nmap.org ) at 2022-02-09 01:15 PST
Nmap scan report for 167.172.144.11
Host is up (0.0053s latency).
Not shown: 997 filtered ports
PORT     STATE  SERVICE
22/tcp   open   ssh
1720/tcp closed h323q931
3389/tcp closed ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 19.53 seconds
sysadmin@ubuntuucb:~/Desktop$
```

From the SYN scan, port 22 is open for ssh.

This occurred on layer 4 (Transport Layer) of the OSI model as Ports and TCP for Transmission Control Protocol as it puts data onto the network.

Phase 3: "I Feel a DNS Change Comin' On"

Port 22 for SSH was open from the last excersize, so I will be using the given credentials to login to the active IP 167.172.144.11.

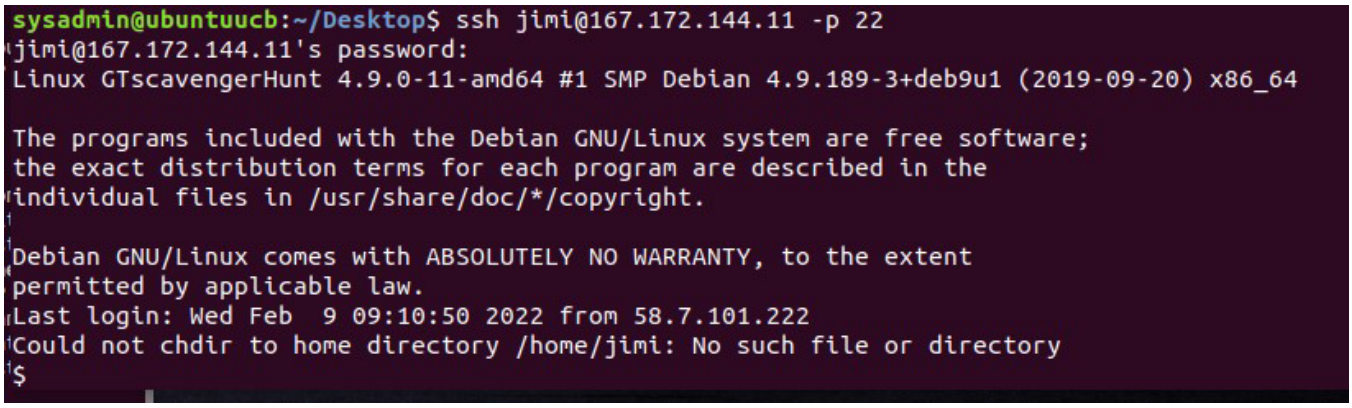
`ssh jimi@167.172.144.11` using default SSH port 22 or specify `ssh jimi@167.172.144.11 -p 22`

```
sysadmin@ubuntuucb:~/Desktop$ ssh jimi@167.172.144.11 -p 22
jimi@167.172.144.11's password:
Linux GTscavengerHunt 4.9.0-11-amd64 #1 SMP Debian 4.9.189-3+deb9u1 (2019-09-20) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Feb  9 09:10:50 2022 from 58.7.101.222
Could not chdir to home directory /home/jimi: No such file or directory
$
```

Phase 3: "I Feel a DNS Change Comin' On"

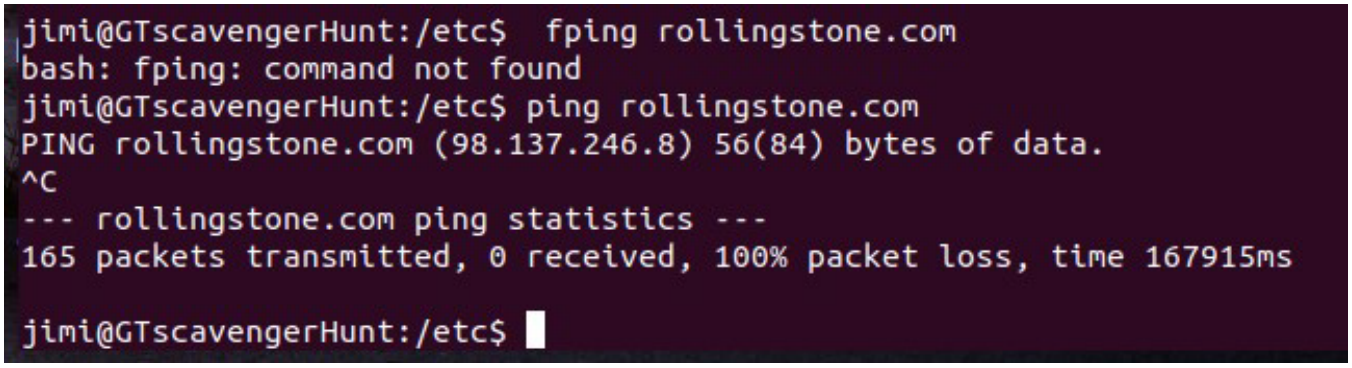
Port 22 for SSH was open from the last excersize, so I will be using the given credentials to login to the active IP 167.172.144.11.

```
ssh jimi@167.172.144.11 -p 22
```



Since Rockstar Corp reported that they were unable to access rollingstone.com, I ran a ping to see what IP address returns when trying to access the website. The command and results are as follows:

```
ping rollingstone.com
```



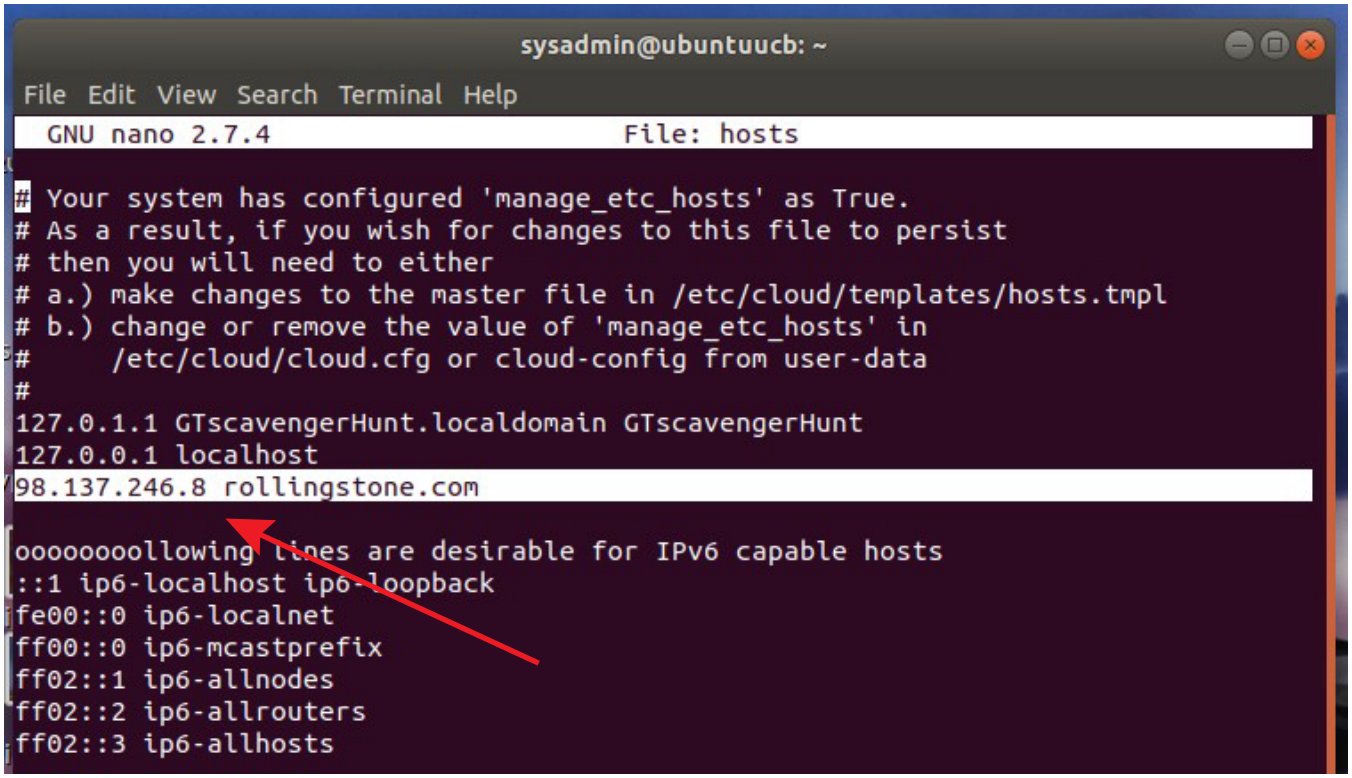
The results returned the IP address 98.137.246.8 in my findings. Since the linux “hosts” file is used by the operating system to translate specific hostnames and IP-addresses, we will double check it to determine if something was modified. Something might be changed there for why the website is not working properly on the particular system.

The information in the /etc/hosts file was modified by someone with the website for rollingstone.com going to 98.137.246.8 instead. This was viewed with the nano editor.

```
nano /etc/hosts
```

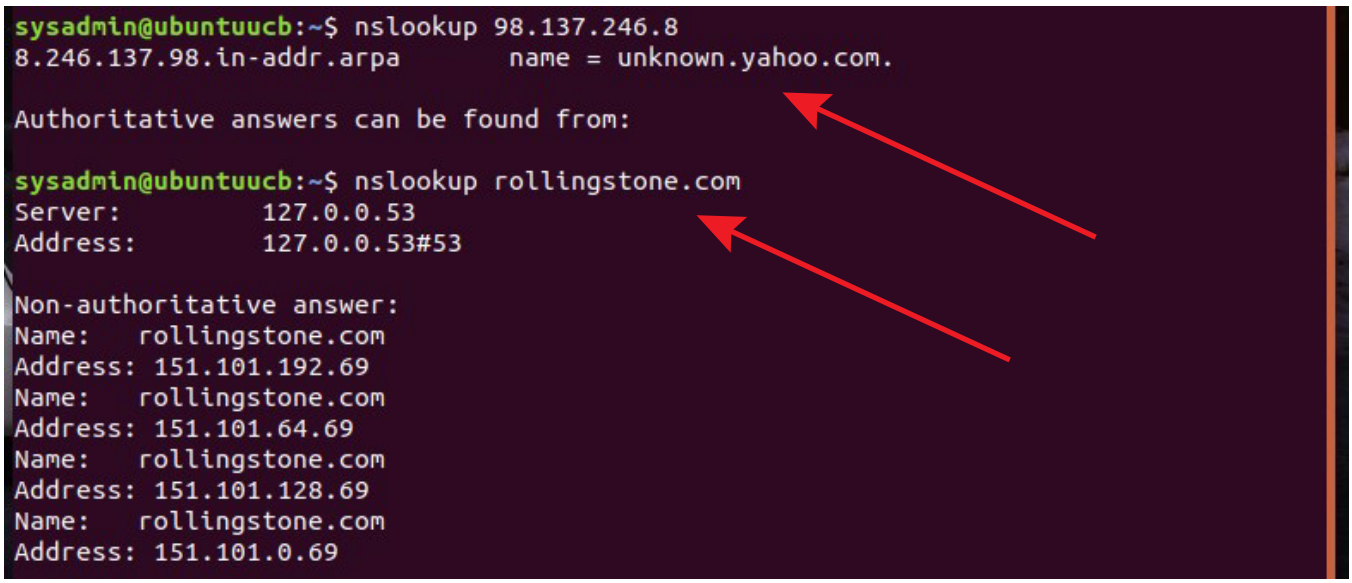
User privileges should be restricted to avoid unauthorized edits to the hosts file that would cause DNS issues that could lead to more malicous outcomes.

The results of the modifcations are on the following page.



Using the nslookup utility, I did a look up with the IP address that the website is listed with and found that it lead to an unknown.yahoo.com domain. The real rollingstone.com website produced a different IP address than the one listed in the hosts file.

`nslookup 97.137.246.8` `nslookup rollingstone.com`

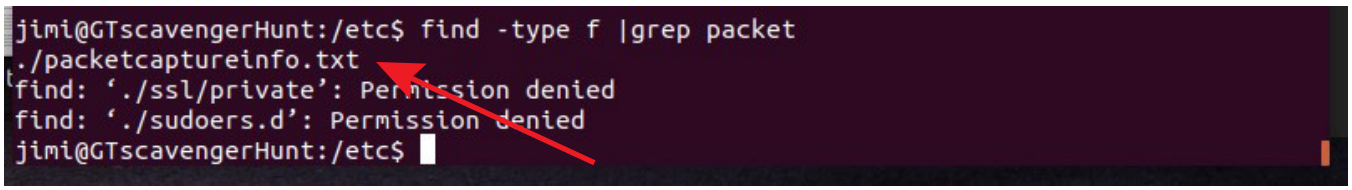


This occurred on layer 7 (Application Layer) of the OSI model as the redirect for for the website IP address was a DNS issue that was determined with the ping utility and confirmed with the nslookup utility.

Phase 4: "Sh *ARP* Dressed Man"

Since the hosts file was in the /etc directory, the directory was reviewed for anything that might not belong or indicate a note regarding “packet captures.” The file “packetcaptureinfo.txt” was found with the find command and the word “packet” put into the grep command.

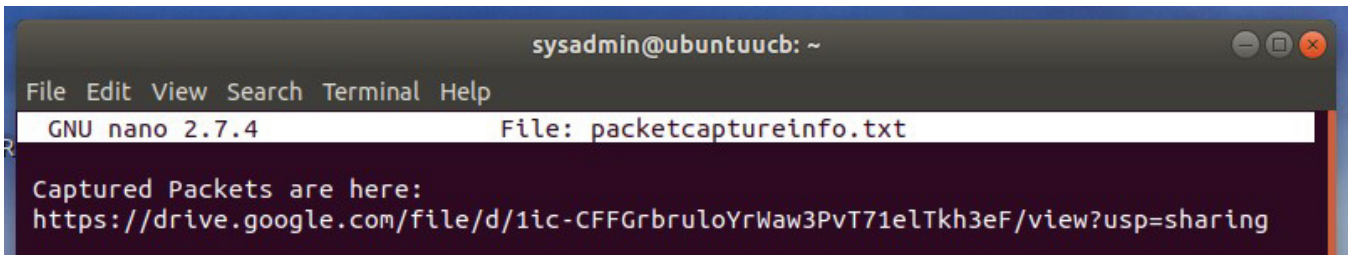
```
find -type f | grep packet
```



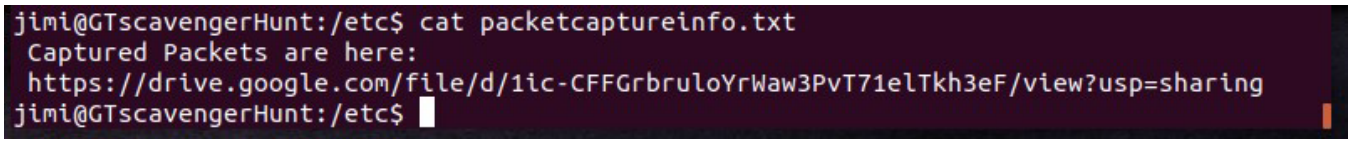
Viewing this file with the nano editor or cat resulted in the following link and info:

Captured Packets are here:
https://drive.google.com/file/d/1ic-CFFGrbruloYrWaw3PvT71elTkh3eF/view?usp=sharing

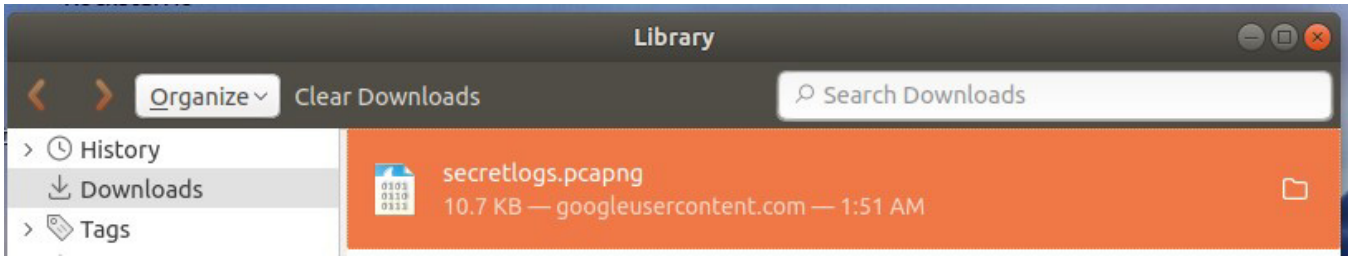
```
nano packetcaptureinfo.txt
```

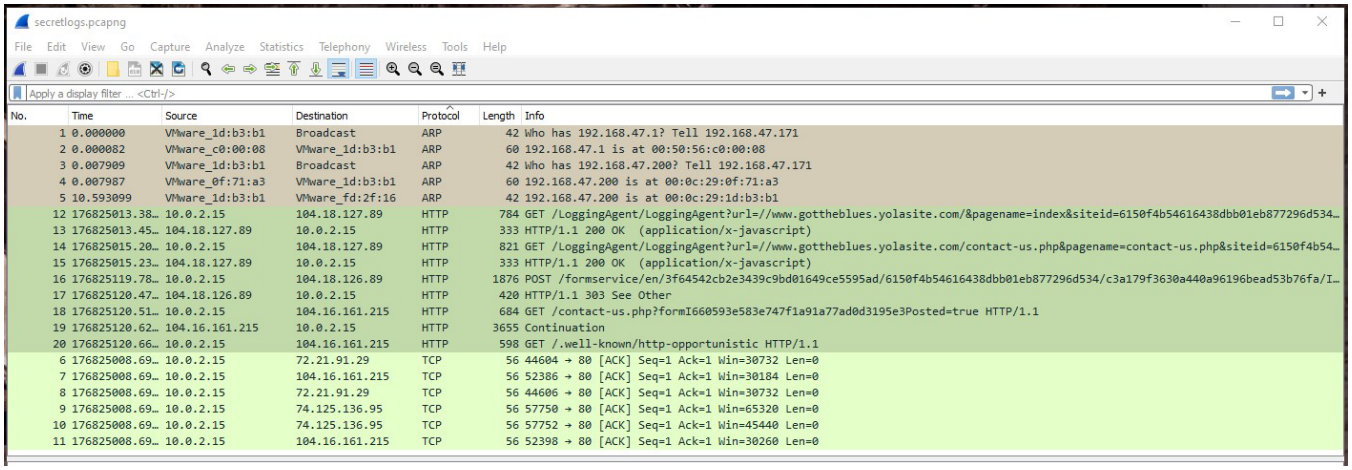



```
cat packetcaptureinfo.txt
```



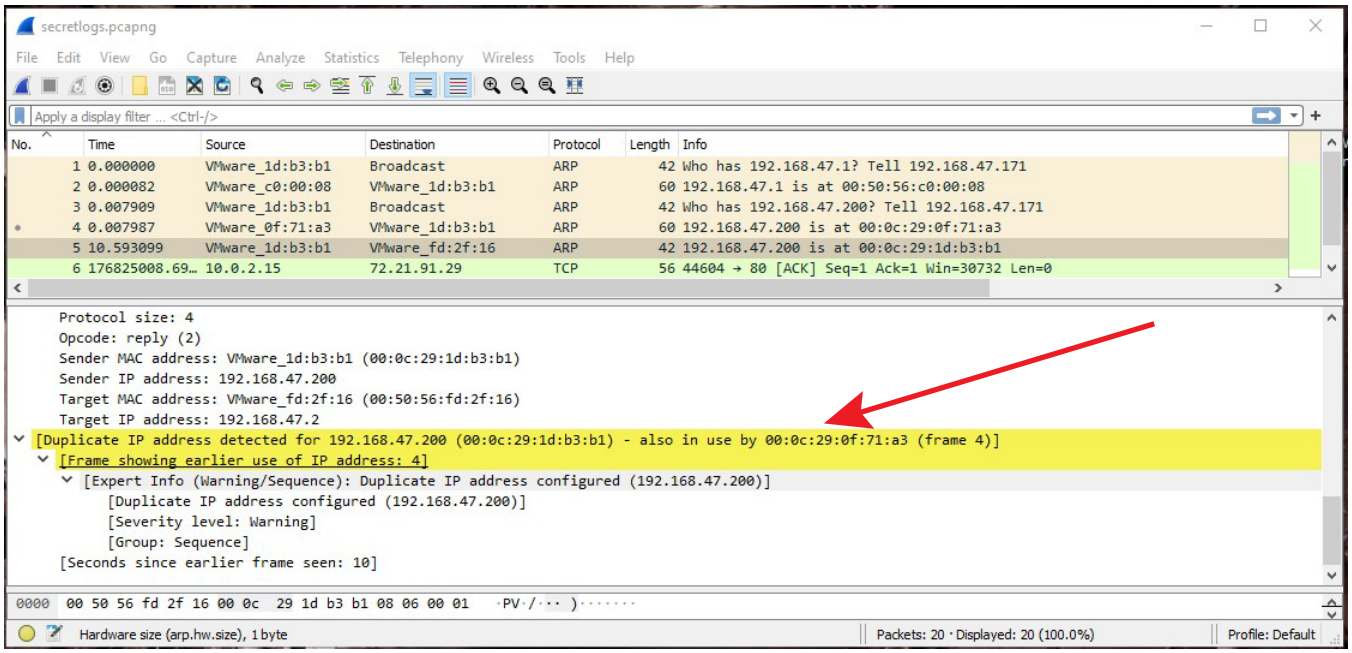
A file called secretlogs.pcapng was downloaded from the Google Drive link.



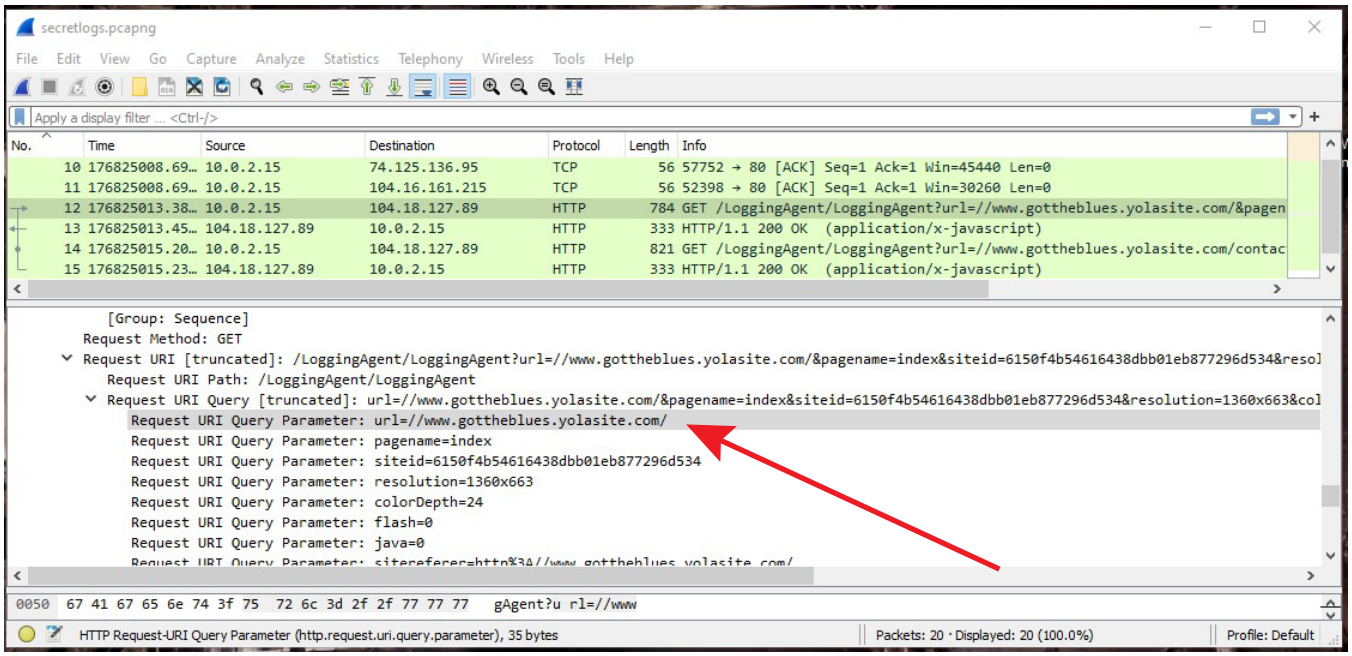


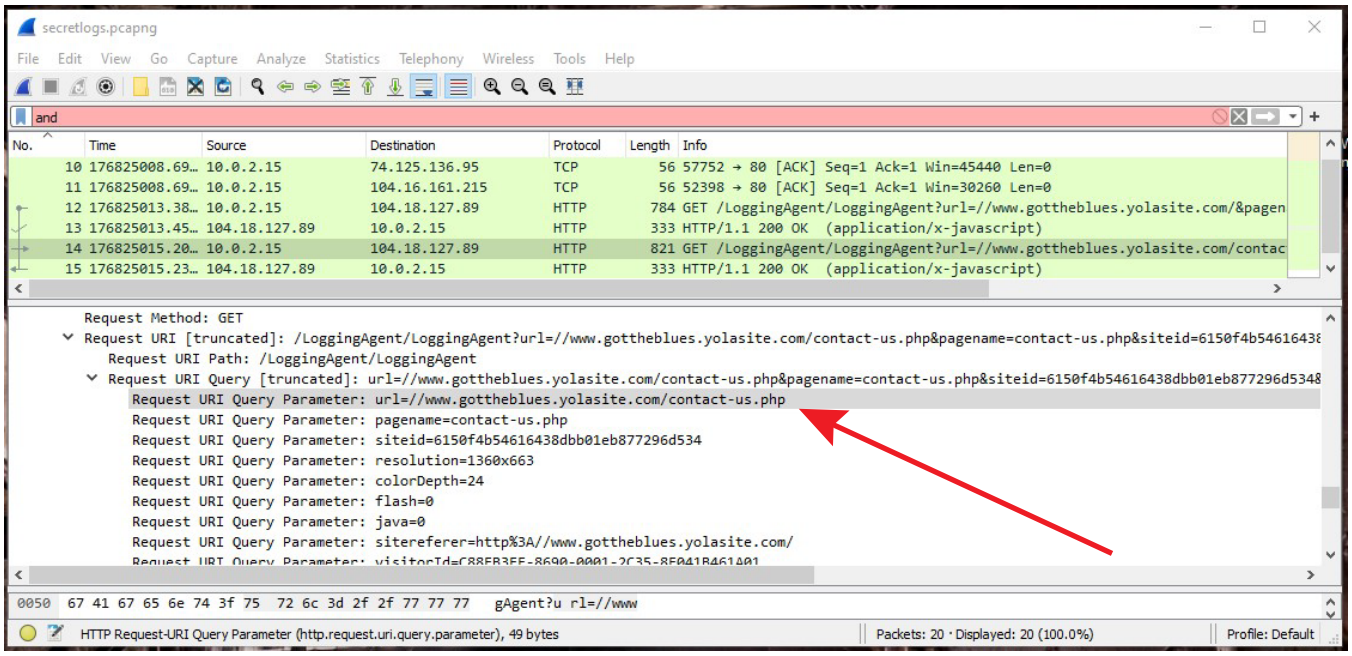
In the pcap file that was downloaded from the recovered link, I have determined that the bad actor was sending ARP (Address Resolution Protocol) requests from the IP 192.168.47.171 (MAC 00:0C:29:1d:b3:b1) while inside the Rockstar Corp Server(s) to find a specific machine IP 192.168.47.200 (MAC 00:0c:29:0f:71:a3).

Once they found the target machine, they cloned their IP address from 192.168.47.171 => 192.168.47.200 or spoofed his MAC address to be able to also get information meant for 192.168.47.200 as both MAC addresses were using the same IP.



The bad actor then accessed a competitor website, navigated to their “Contact Us” page to send an inquiry to the other company with the web page contact form as shown in the following screenshots on the next page.

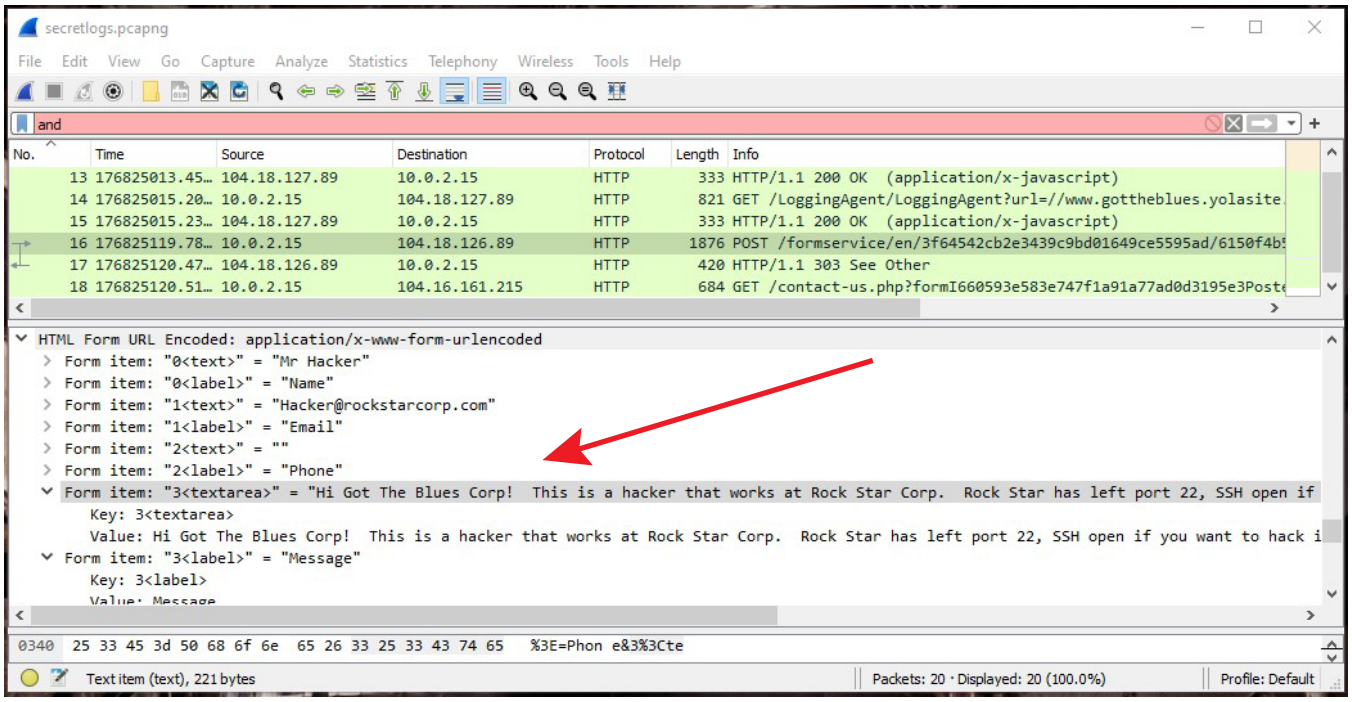




They then used the contact form to send a message out to “Got The Blues Corp” that he is a hacker at Rock Star Corp, telling them about a vulnerability with the server, and for a hefty reward fee, they can release the login information to access the server. The message reads as follows:

"Hi Got The Blues Corp! This is a hacker that works at Rock Star Corp. Rock Star has left port 22, SSH open if you want to hack in. For 1 Milliion Dollars I will provide you the user and password!"

The following screenshot shows the message and contact information that was left with “Got The Blues Corp” in the form.



The bad actor “Mr Hacker” was communicating that the open SSH port 22 was a vulnerability to a competitor company, and also willing to provide credentials if he was given the reward that was requested.

I recommend patching this vulnerability by changing the default port 22 for SSH to another port that is not commonly used as well as do away with passwords to login to the server. Instead, Rock Star should implement the use of authentication key pairs with public/private rsa key pairs generated for each user to use for logging in. This protects the server from access via stolen username/password credentials and avoid possible brute force password attacks on the server with known login usernames of employees.

This occurred on layer 7 (Application Layer) of the OSI model as the IP address or MAC address was cloned on the server after being compromised. ARP requests were then made via layer 3 (Network Layer) when the bad actor was searching the local network of the server for specific IP address. This went back to layer 7 (Application Layer) when the bad actor “Mr Hacker” made HTTP requests to access the “Got The Blues Corp” website’s contact form to submit and attempt to sell compromised information from “Rock Star Corp.”