Todd Wenker

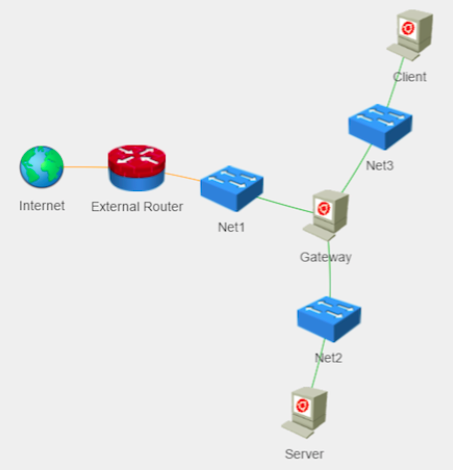
ASU ID: 1206233882

Lab Assignment 2

Sniffing and Scanning

**Summary:** The goal of the lab is to use Wireshark to view the incoming and outgoing packets from a machine and then analyze their contents. The command *nmap* will be used to scan the networks of different hosts.

**Network Setup:**



The two virtual machines used are Gateway and Client (Server was not used).

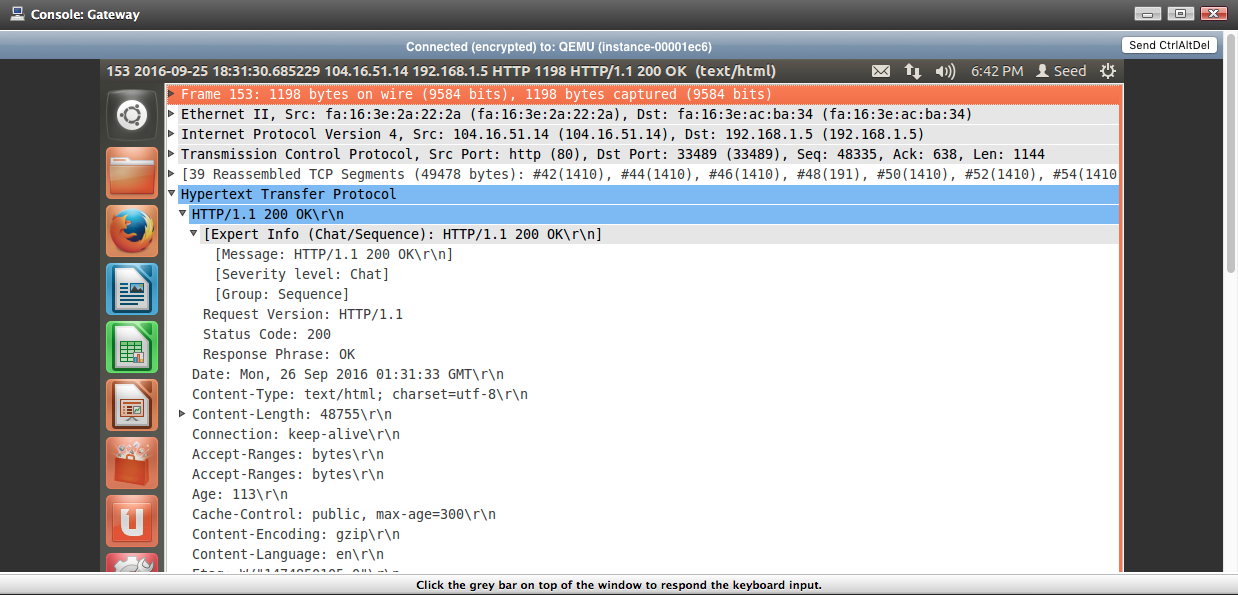
**Software Packages Used:**

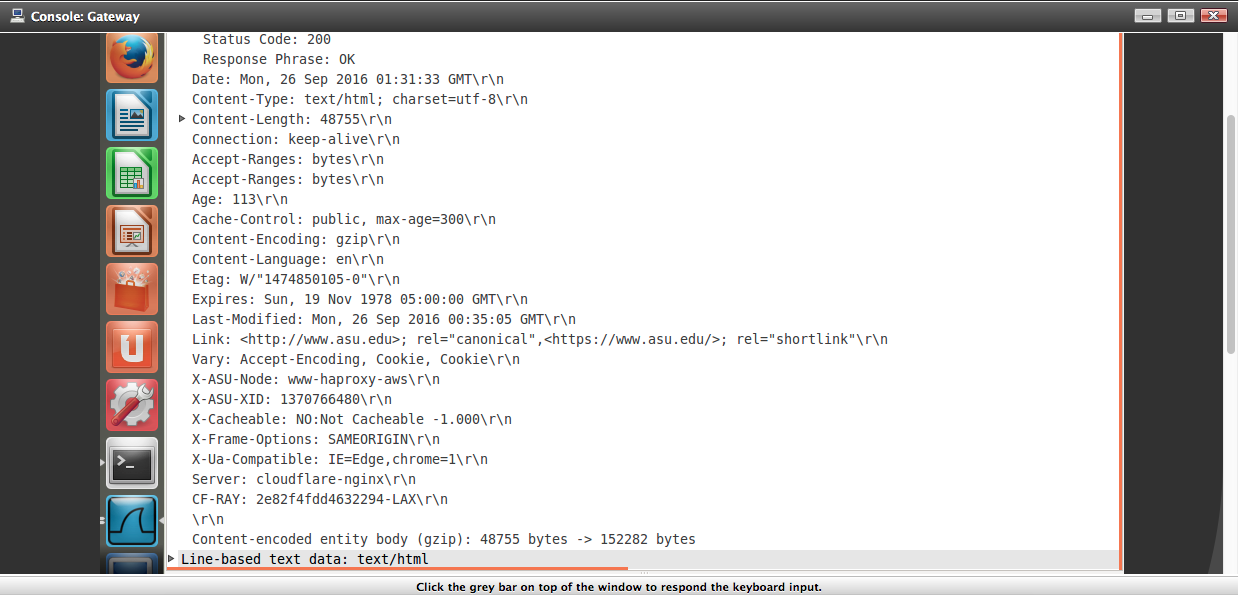
* Wireshark
* nmap
* traceroute

**Task1 --- Sniffing Web Traffic**

On the Gateway machine, the application Wireshark is opened and a capture is set up on the eth0 interface so that any traffic that enters or leaves from that interface will be captured. After the capture is started, Firefox is opened and the address *http://asu.edu* is entered into the address bar. The capture can then be ended. The HTTP request and response packets are shown below.

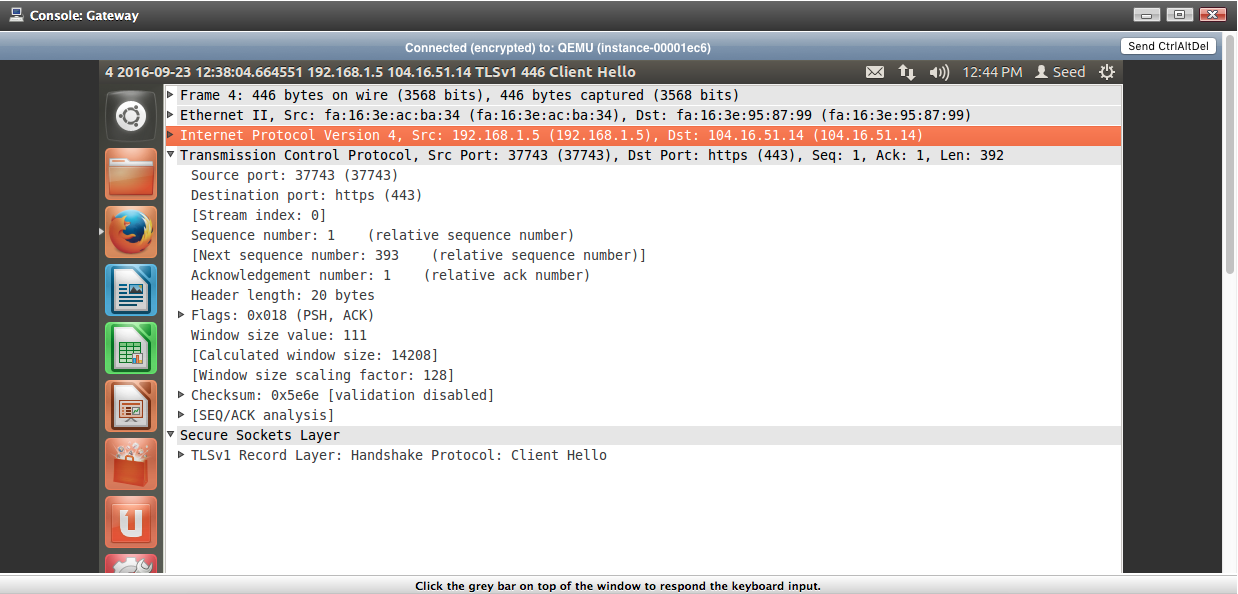
Above is the HTTP GET request sent by the Gateway Machine. The packet contains the host being contacted (asu.edu\r\n), attributes of the system sending the request, the language, and the encoding.

****

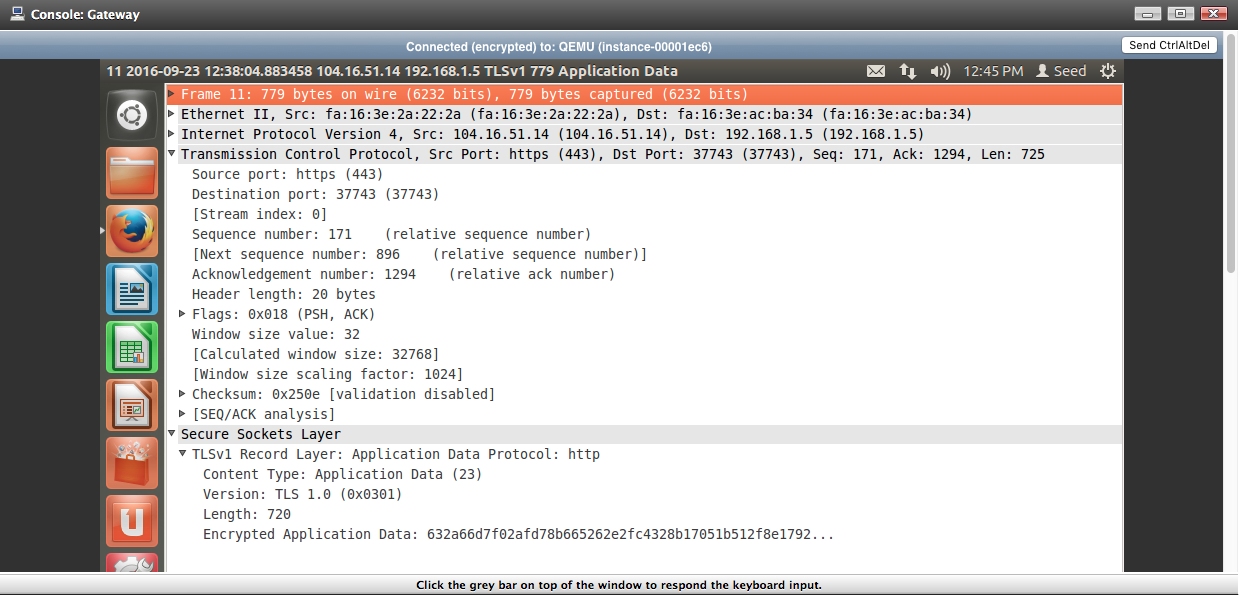
****

The above pictures shows the HTTP Response sent by the asu.edu server. At the bottom, the actual HTML file wanted by the client is there.

Next, the same webpage is accessed but instead of entering *http://asu.edu, https://asu.edu* is entered instead. The Request and the Response packets are shown below:

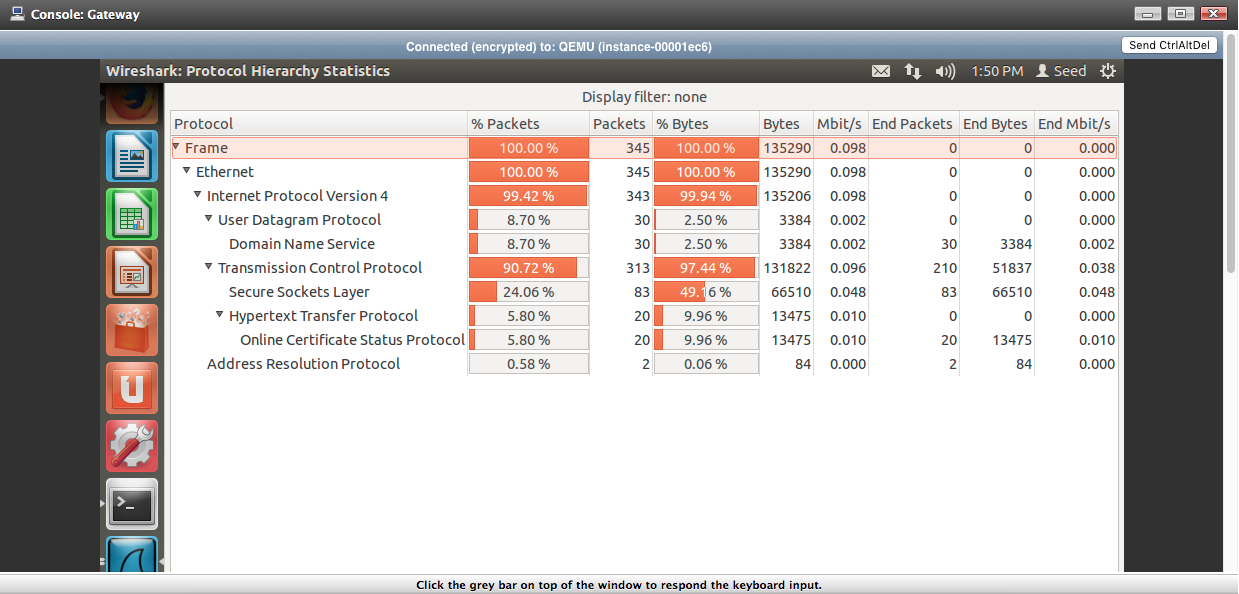


The HTTPS request is shown above. Instead of an HTTP field, the packet has an SSL (Secure Sockets Layer) field.



The HTTPS response packet is shown above. The major difference between the HTTP and HTTPS packets is that HTTPS chooses a key between machines and encodes the data so that if it is intercepted, it cannot be easily read.

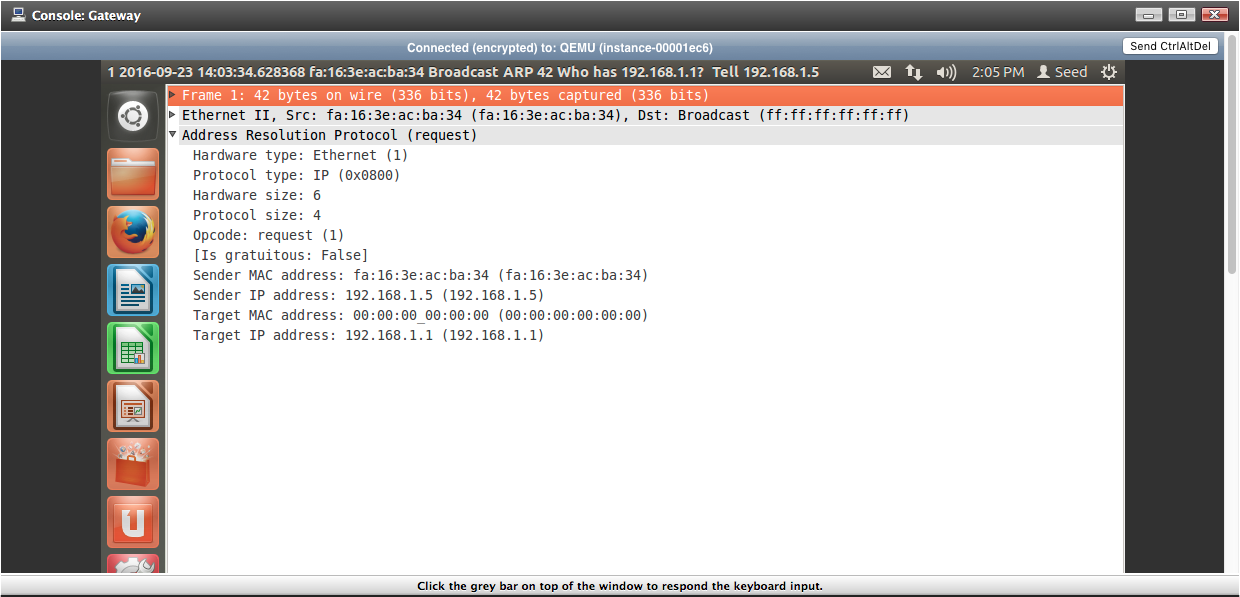
Wireshark can also show the capture’s protocol statistics, as shown below.



Noticeably, TCP is used much more than UDP on the Network layer.

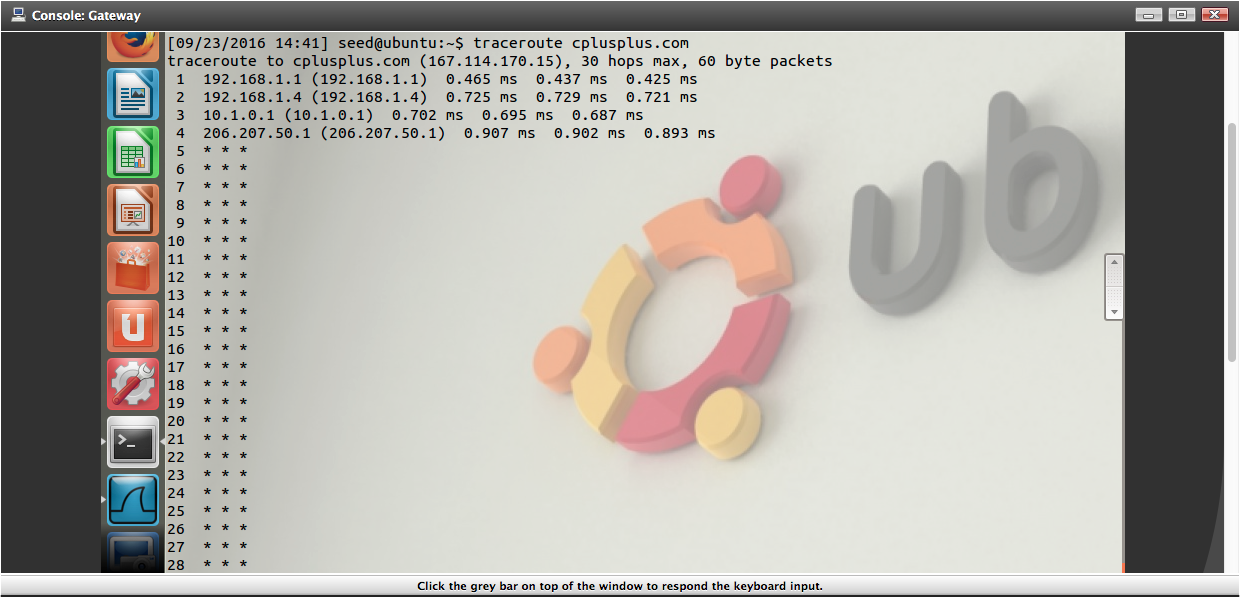
**Task 2 --- Capturing an ARP and an ICMP Packet using Wireshark**

ARP (Address Resolution Protocol) is used to map an IP address, which is a network level address, to a physical address like a MAC address. To clear the ARP cache, the command *sudo ip –s –s neigh flush all* will remove all the entries in the table. From here, a new capture is started on eth0 and Firefox is opened long enough for the homepage to load before the capture is ended. Browsing through the capture, we can find ARP requests, shown below:

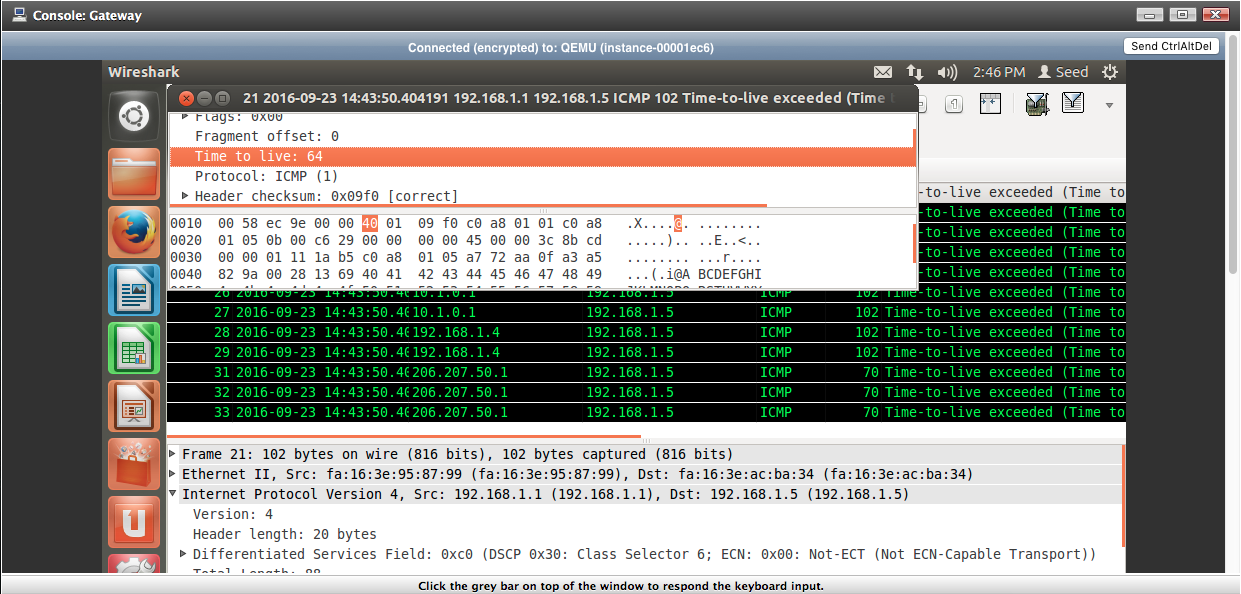


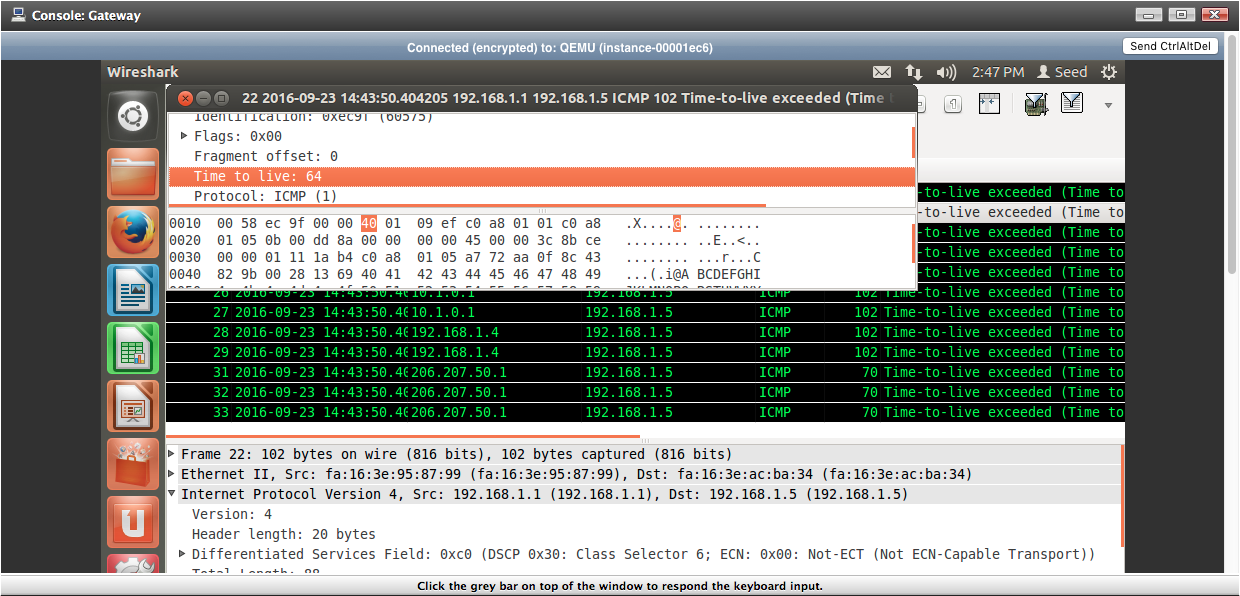
As shown above, the ARP request is sent by 192.168.1.5, the IP address of the Gateway machine, to find the MAC address of IP address 192.168.1.1. The 00:00:00:00:00:00 in the Target Mac Address shows that the Sender machine does not have the MAC address of the target IP address, hence the sending of the ARP request to begin with.

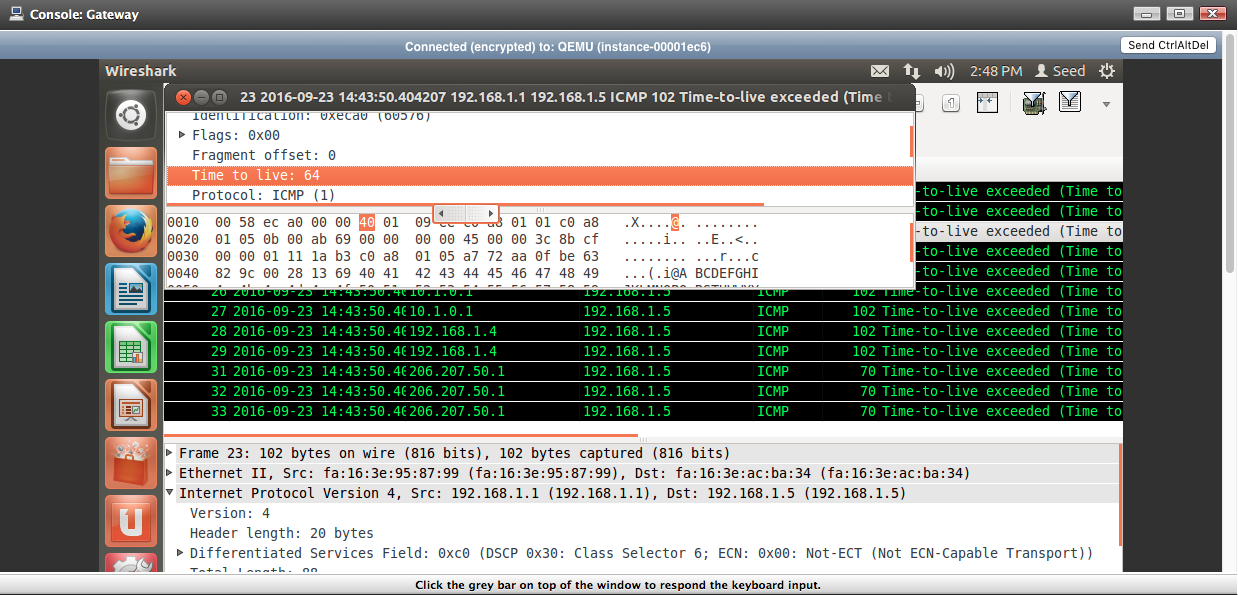
To generate ICMP (Internet Control Message Protocol) requests, the command *traceroute* needs to be used so that our machine can receive the messages from the IP addresses en route to the target IP address. First, a capture is started to record the results of *traceroute.* By entering the command *traceroute cplusplus.com,* we can record how our request reaches the specified target.

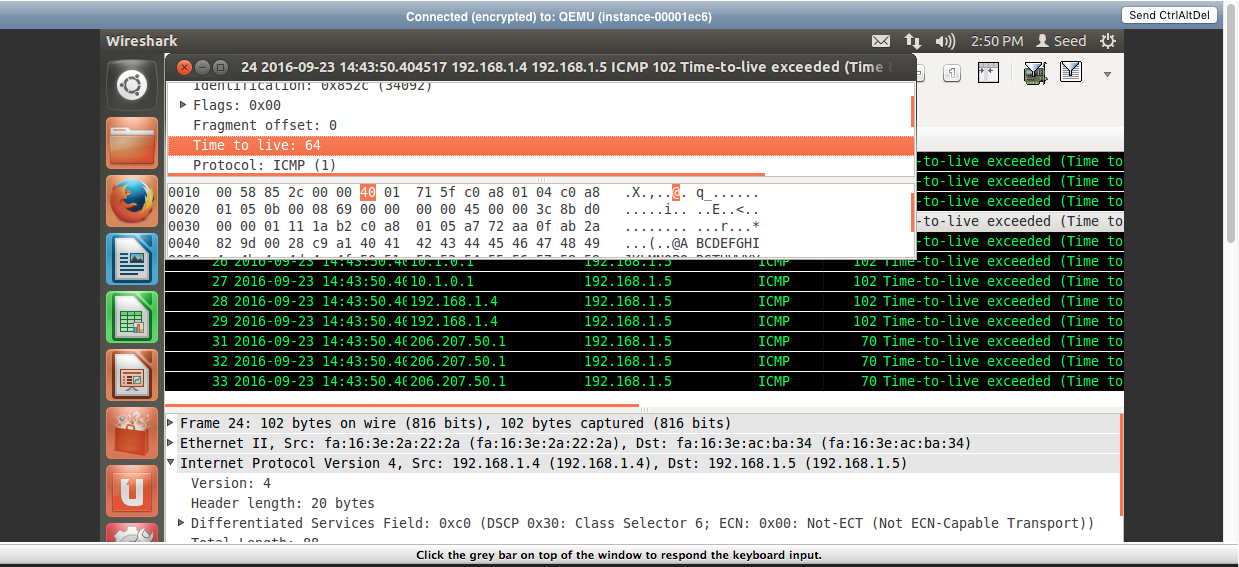


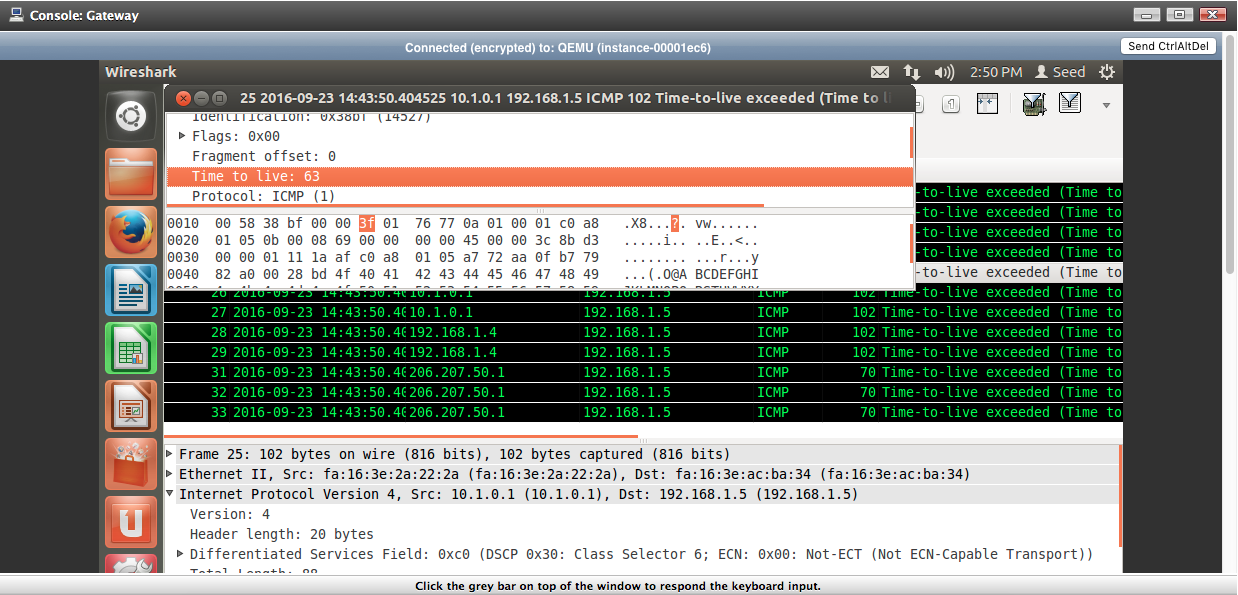
Above is the result of the command *traceroute cplusplus.com*. The ICMP requests in the capture are as follows.

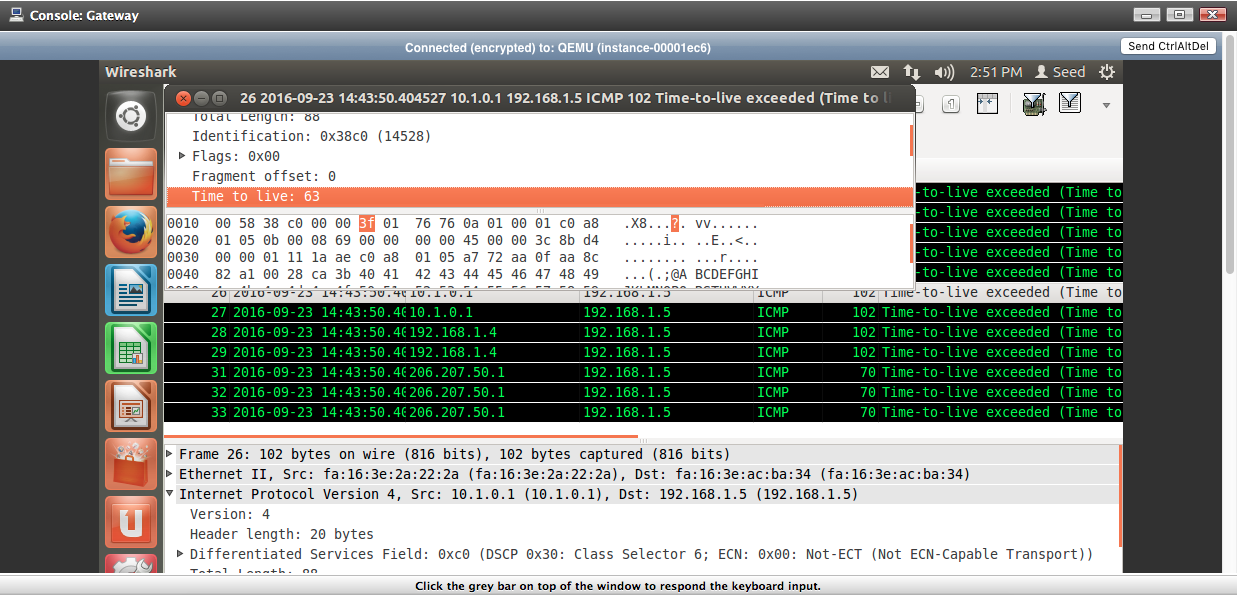


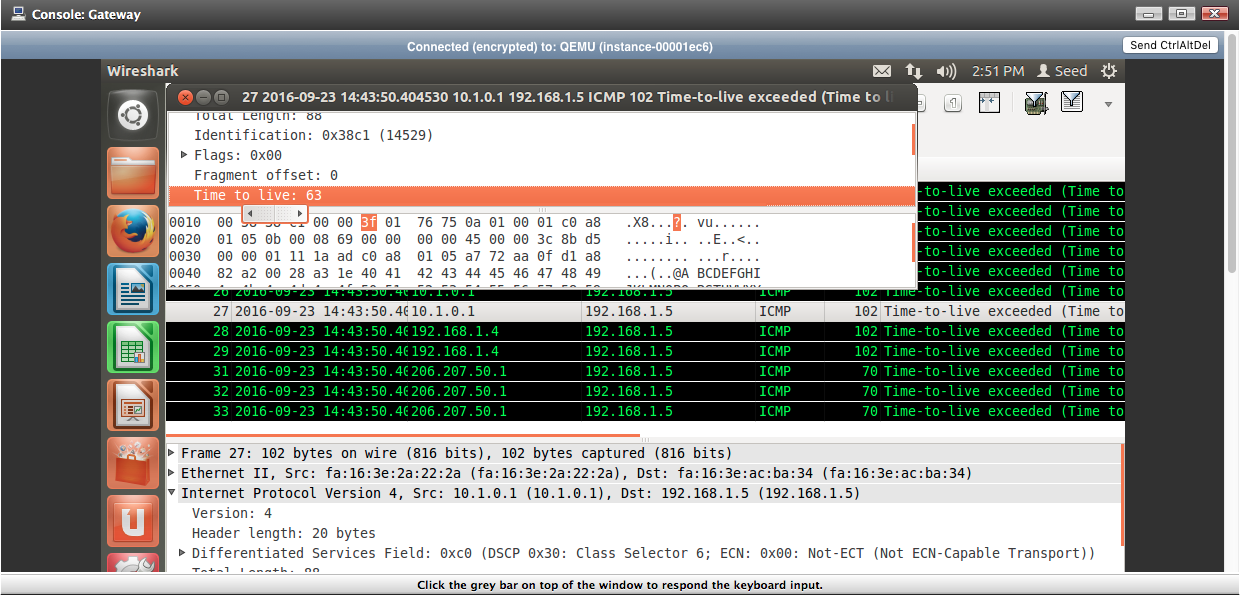


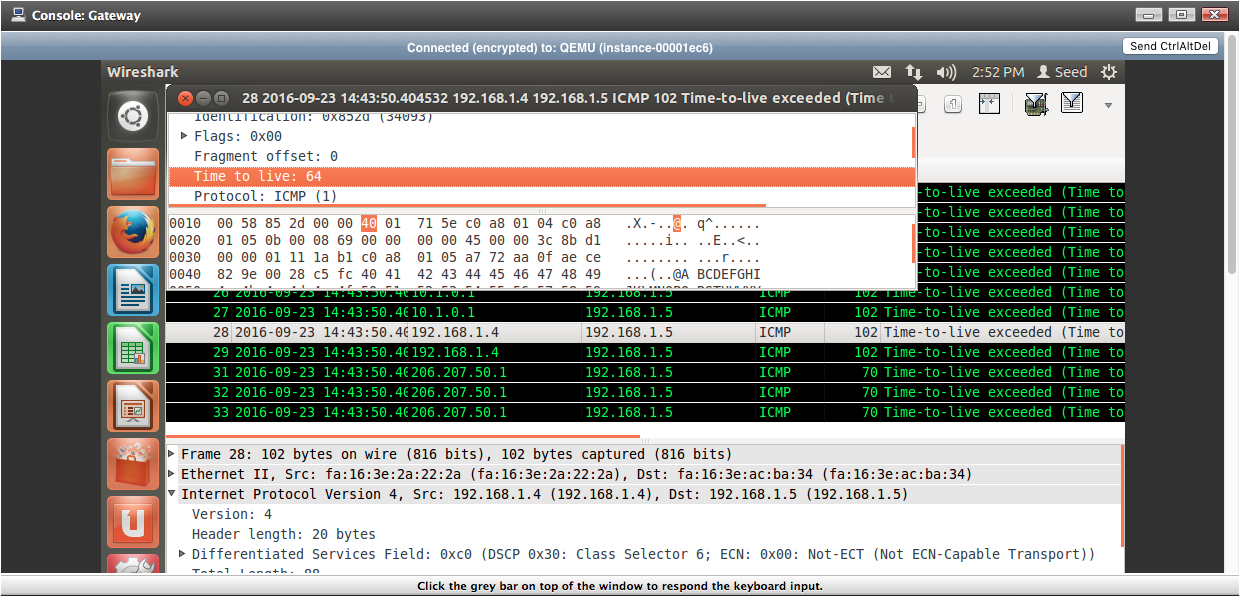


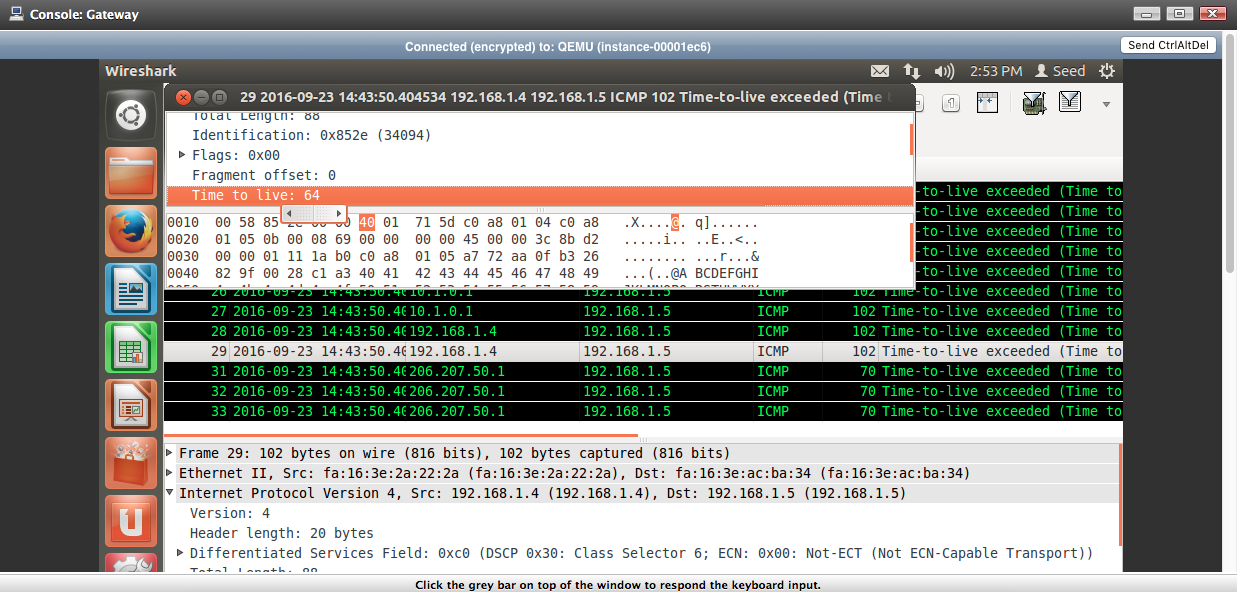


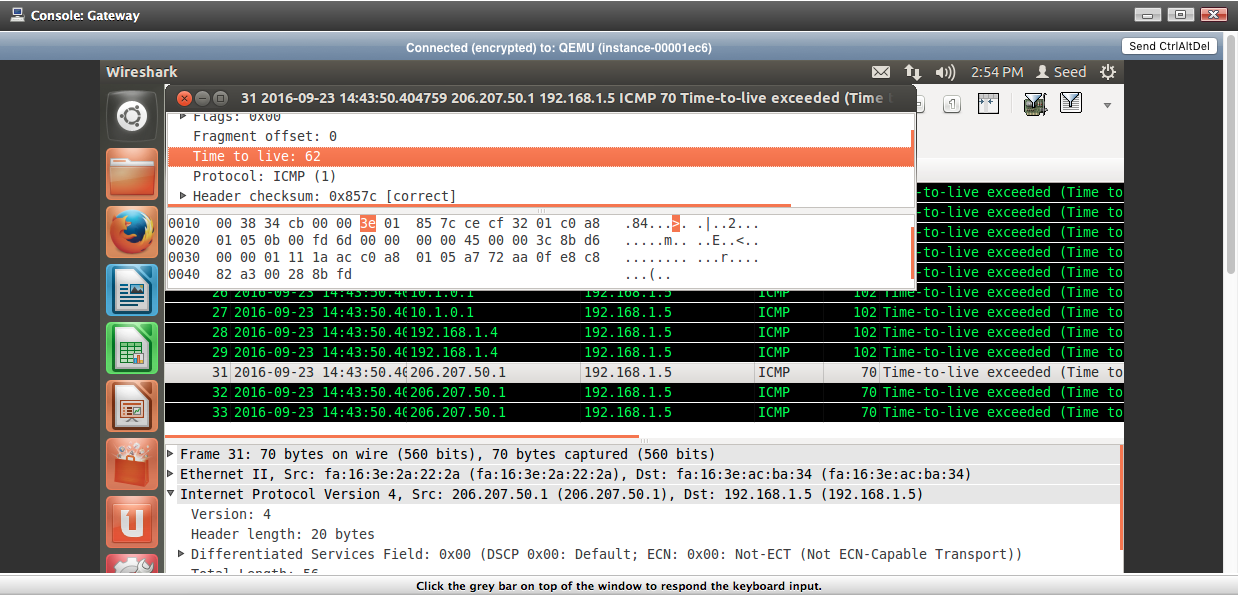


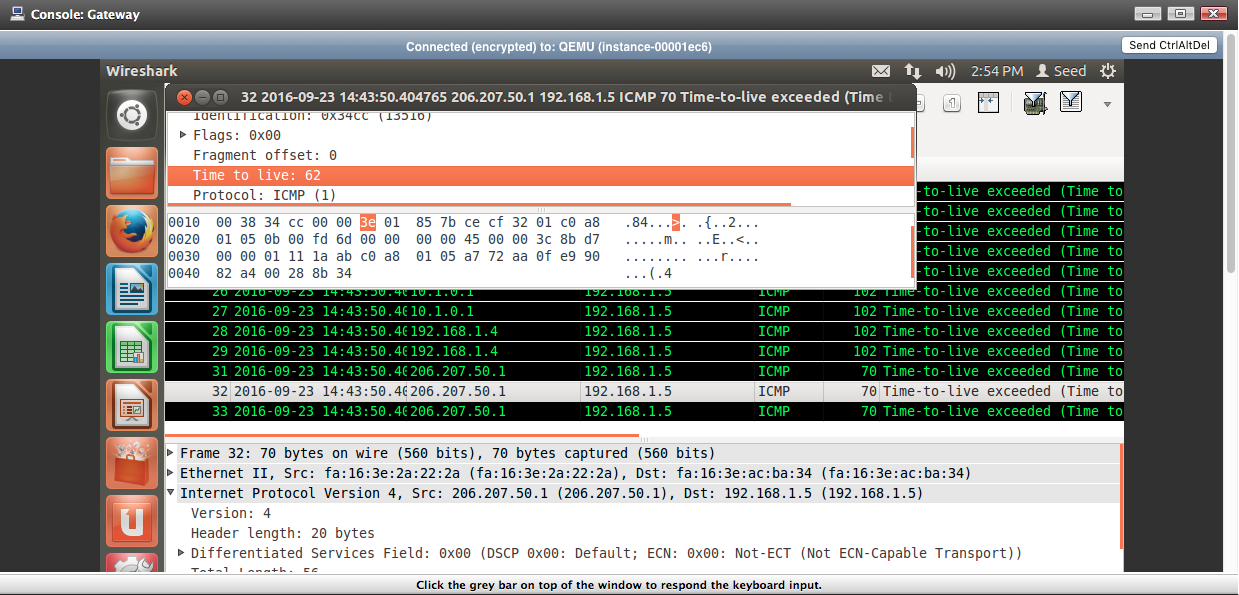


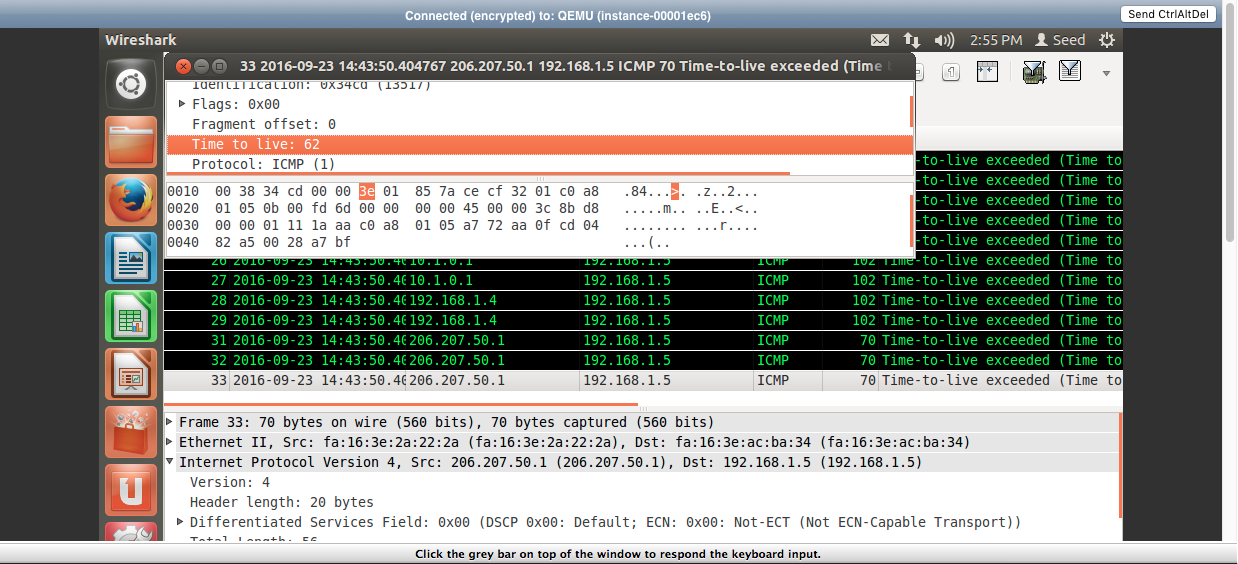








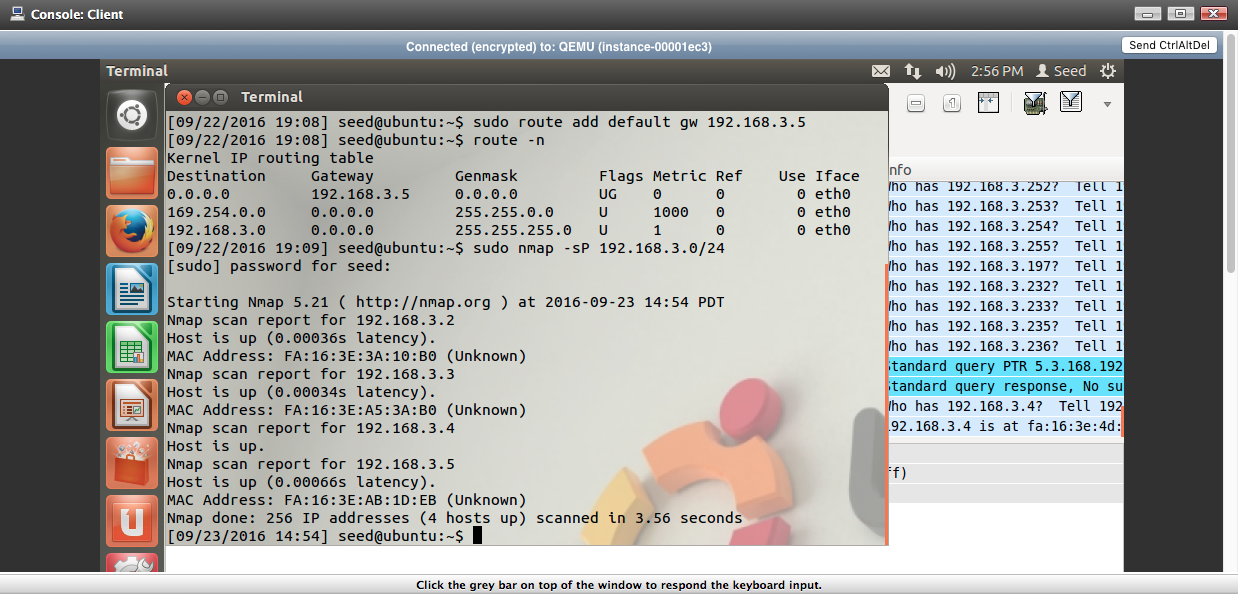




Each IP address loops and sends multiple ICMP messages to the host machine. However, TTL (Time to Live) does not decrease when the source IP address sends duplicate messages. The TTL only decreases when the messages are sent from outside of the 192.168.1.0 network.

**Task 3 --- Network Scanning Using nmap on Client**

The command *nmap* can be used to scan a network and find any connected hosts and any ports on said hosts. By using the command *sudo nmap –sP 192.168.3.0/24,* we can scan the 192.168.3.0 network. The –sP flag denotes that the search will only ping each host and see if they are online instead of a more thorough scan.

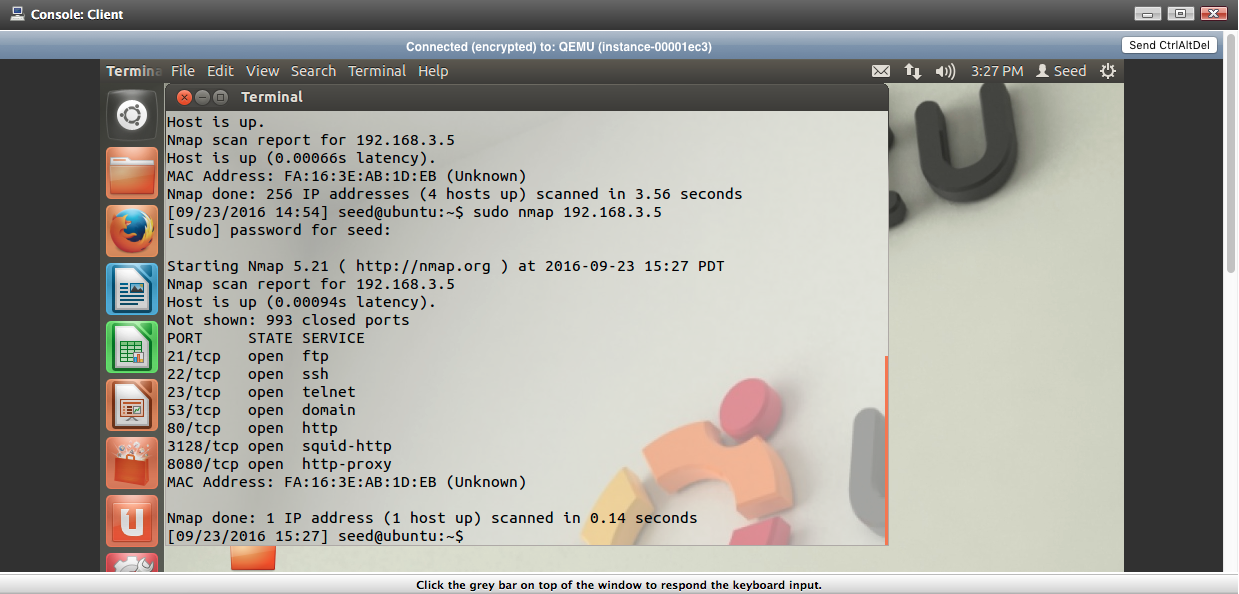


Above are the results for the command *sudo nmap –sP 192.168.3.0/24* run on the Client machine*.* The scan captures the IP of active hosts and their MAC address. Before the command was run, a capture was started on the eth0 interface.



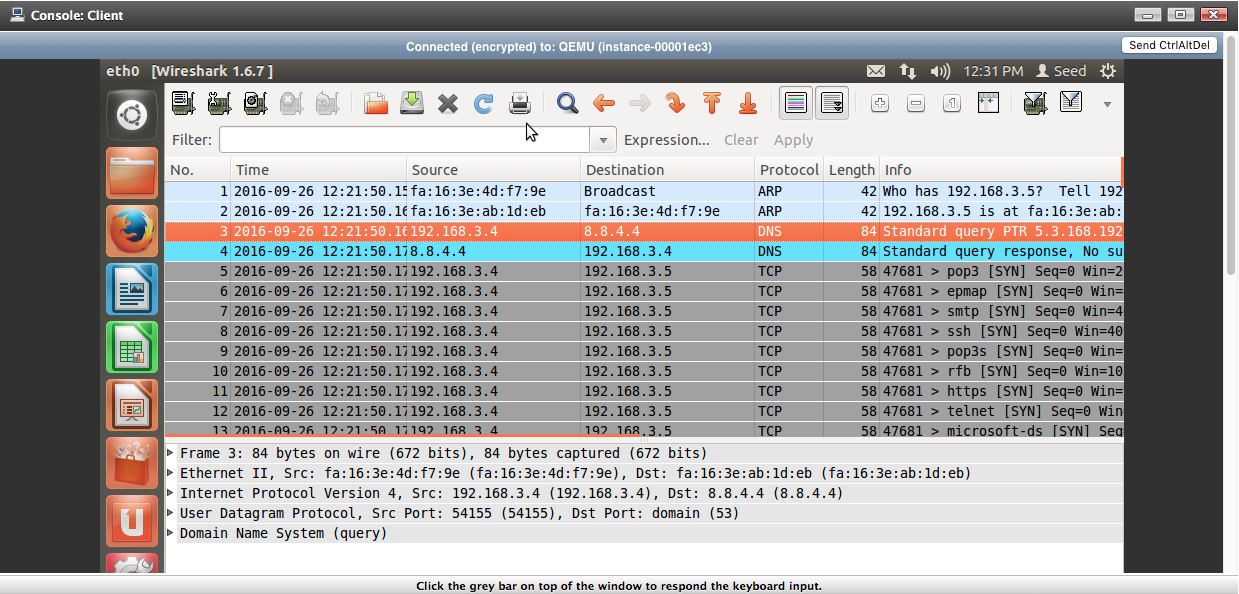
The results of the capture are above. It consisted of numerous ARP requests sent to every single possible IP address on the 192.168.3.0/24 network. This means that an ARP request was sent to all the IP address 192.168.3.0 through 192.168.3.255.

*Nmap* is also capable of scanning individual IP addresses and scanning their ports to determine whether they are open or not.

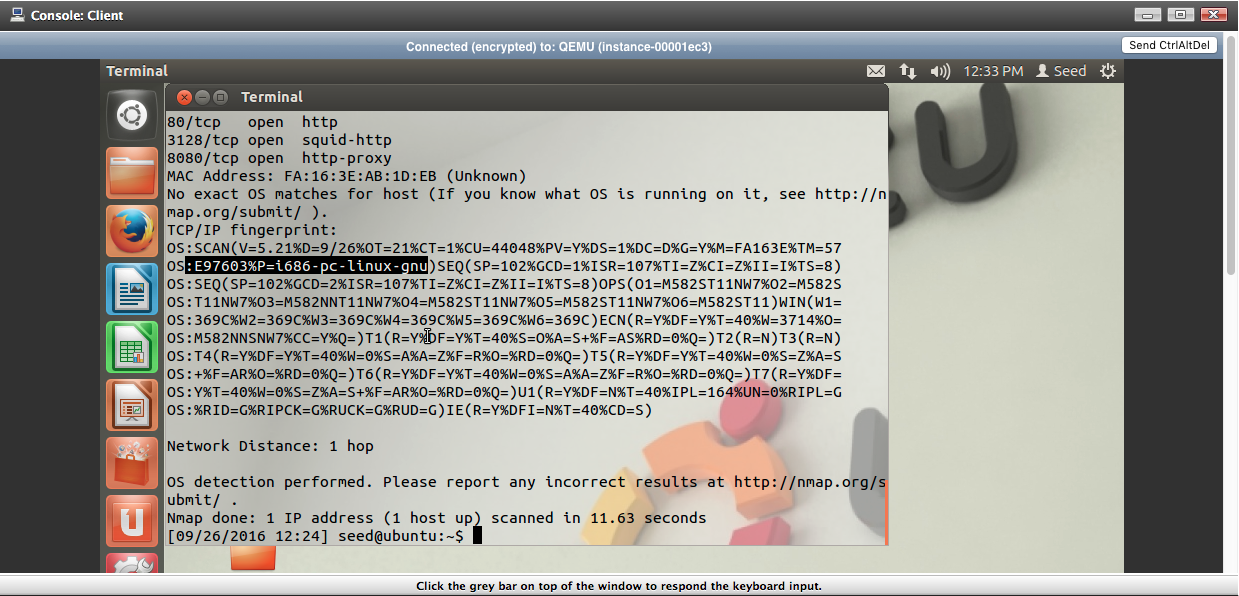


The command *sudo nmap 192.168.3.5*, used on the Client machine, will do a port scan of the Gateway machine. As the picture shows above, Gateway has 993 closed ports and seven open TCP ports. A total of 1000 ports were scanned in total.

The command *sudo nmap –O 192.168.3.5* will do a port scan like before and do an OS scan as well on the target IP address. The capture file of this command looks as follows.



The capture first sends an ARP request to the target IP address to learn its MAC address before sending a series of TCP messages with the SYS flag set.



The OS scan states that there is no exact match for the OS. However, the scan does state that the host is running Linux (highlighted above).

**Conclusion:**

By using Wireshark, users gain the ability to view the packets sent inside a network. By clearing the ARP cache, we can capture ARP messages and by using *traceroute,* we can generate ICMP messages on other machines. Network and port scanning can be done with the *nmap* command, revealing active hosts on a network and any ports on them.