CS 313 Final Project Documentation

Group Members: McKenna Galle, Vina Le, Andrew Hutson

# Objectives

* Create a man-in-the-middle attack using ARP poisoning (also known as ARP spoofing) on a Raspberry Pi.
* Explore other operating systems/languages to use instead of Linux.

# The Path We Took

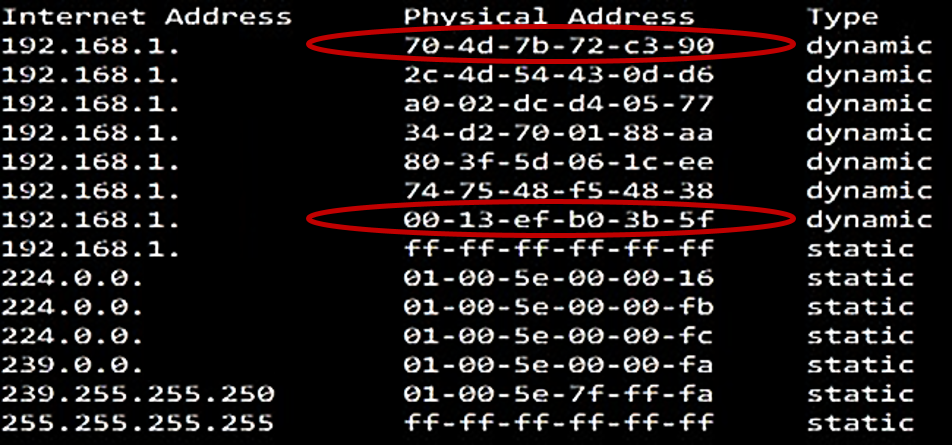
We used a Raspberry Pi running Ubuntu Mate to perform a man-in-the-middle attack on McKenna’s house network. Due to legal and ethical issues, the project was unable to be carried out on Whitworth’s campus. With the user being exposed to any (and all) IP address of all computers on the network, and in part able to ARP poison all said computers, the poisoner is able to cause havoc or reroute the victim into unethical territories. Since we were essentially hacking a network, the group decided that McKenna would do the actual practical portion at her family home – with permission from the parental unit. The actual project is carried out without difficulty – except trying to connect to the raspberry pi’s operating system. The first part was to look up arp functions and use “arpspoof” to carry out the request. By using arpspoof, we were able to replace the router with the raspberry pi as the man-in-the-middle that intercepts packet requests from the victim. Refer to the screenshots found at the end of the document.

# What We Learned

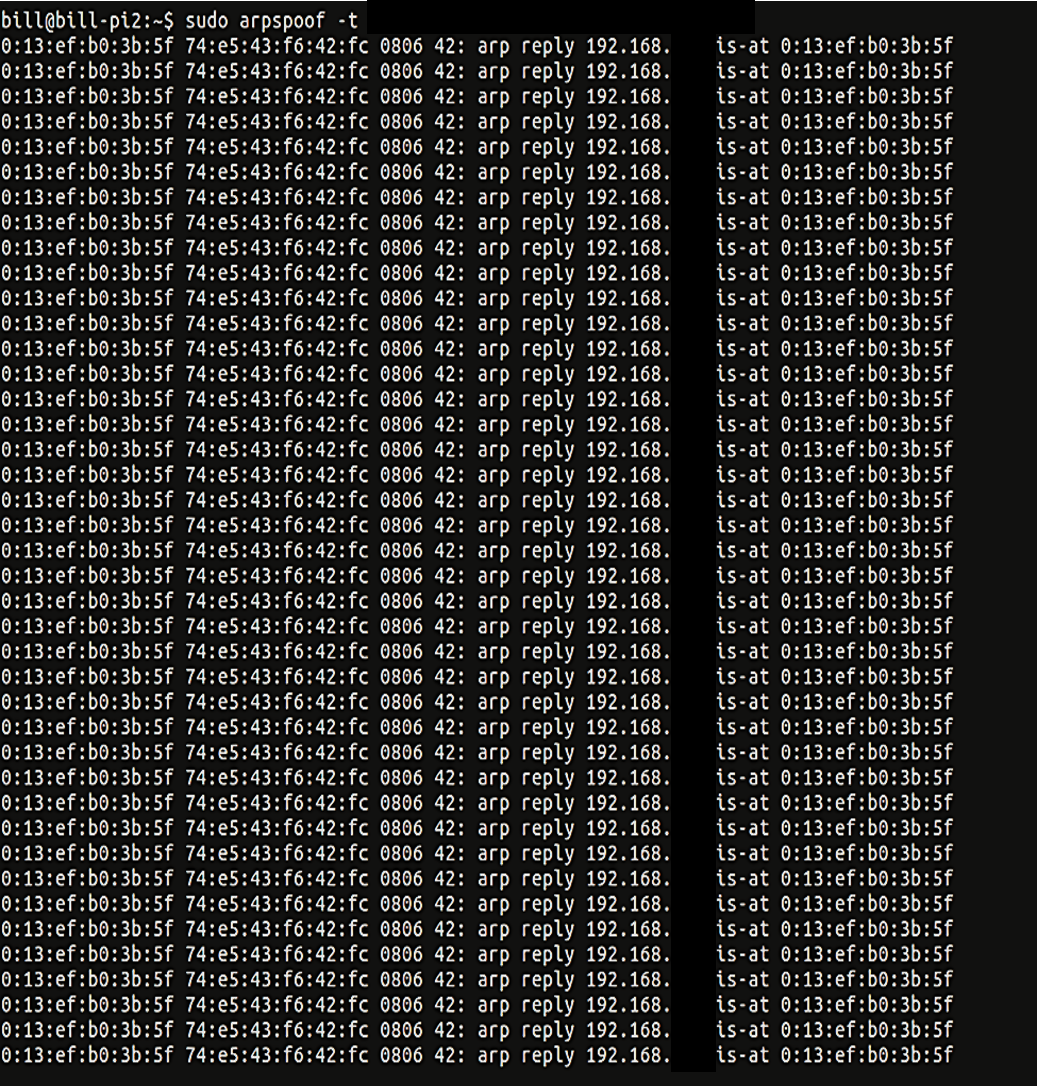
* ARP poisoning is ridiculously easy using Linux – all the commands are there; you just have to know how to use them.
* Relating to the previous point, Linux is the ideal choice for a host OS – again, all the commands are there; no extra code or setup is required.
* When we were deciding which OS to use as a host, we looked at C++ on Windows. We then decided to use Linux since the C++ samples we found online looked ridiculously complicated.
* You can ARP spoof individual devices; you don’t have to spoof the whole network at once.

# Where the Project Could Go Next

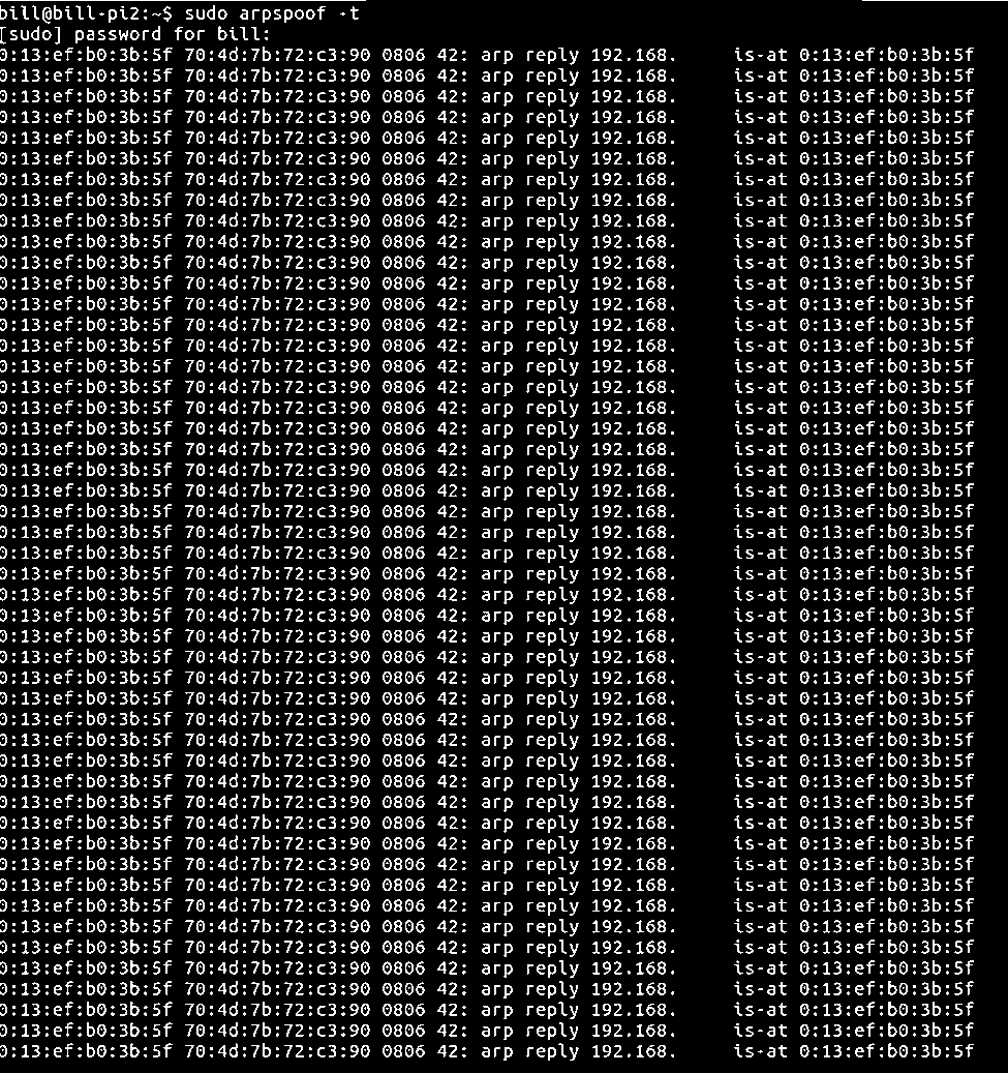
* What we have learned could be used to help develop techniques to prevent ARP spoofing.
* Or, on the opposite end of the moral spectrum, it could be used to do a larger scale attack.

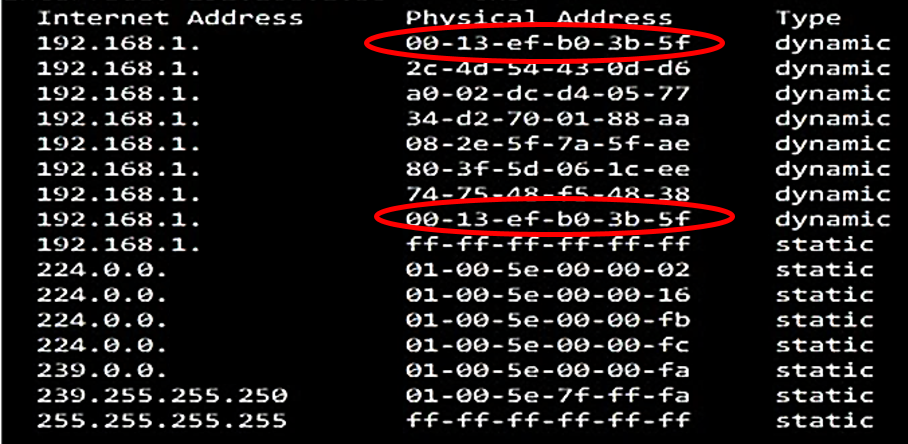


This screenshot displays the network’s internet address (with the IP addresses at McKenna’s house blacked out for security purposes) that displays the gateway’s physical address (the first red circle) and the raspberry pi’s physical address (second red circle). This provides the poisoner with information when the pi is connected to the network.

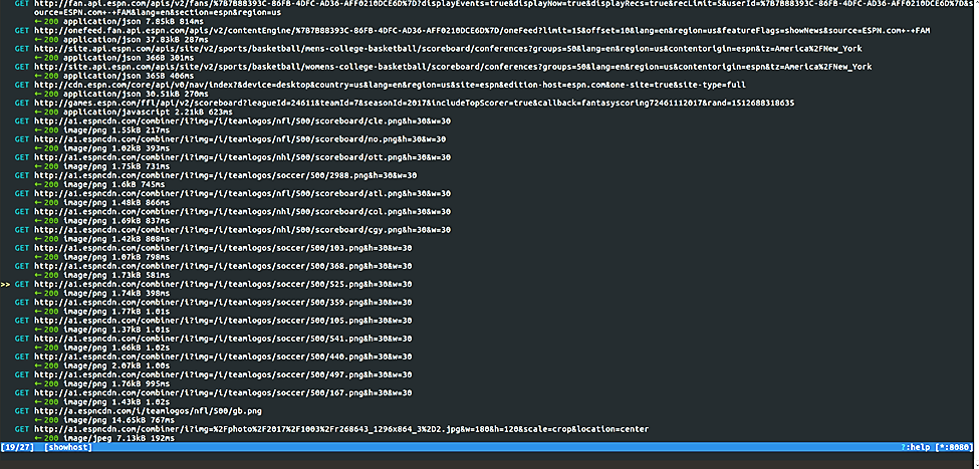
This is the display found on the rasperry pi’s running program. The IP address of the network is blacked out due to privacy reasons.

This is displayed to the poisoner describing the connection made from the victim’s computer.

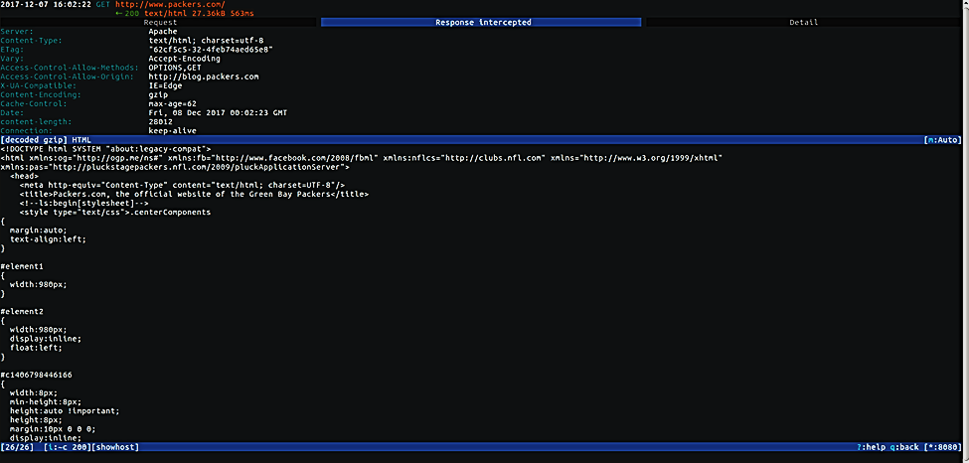
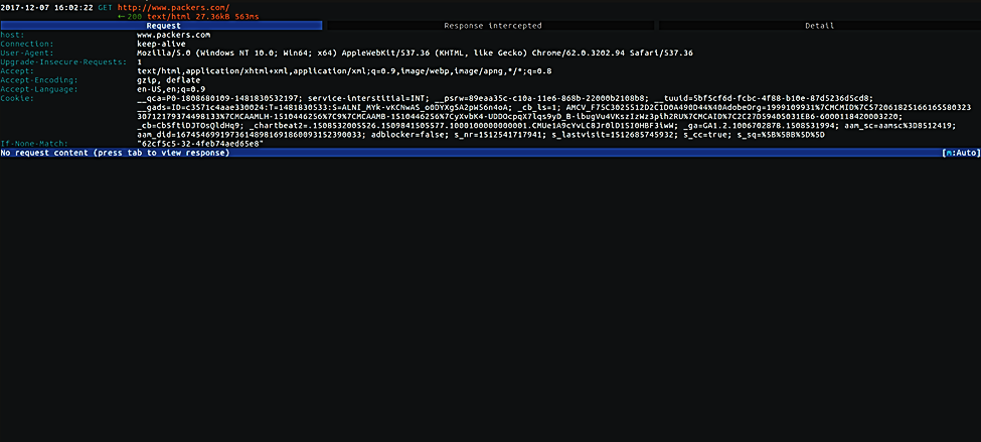
This is the display found on the rasperry pi’s running program. The IP address of the network is blacked out due to privacy reasons. This is displayed to the poisoner describing the reply from the pi to the victim’s packet request.



The view from the poisoner’s perspective that displays the home network’s internet and physical address gateway router (circled first) and the physical address of the raspberry pi (circled second). This message is displayed when the victim has successfully connected to the network and mistaken the raspberry pi (man in the middle) as the router that can help it connect to the internet.

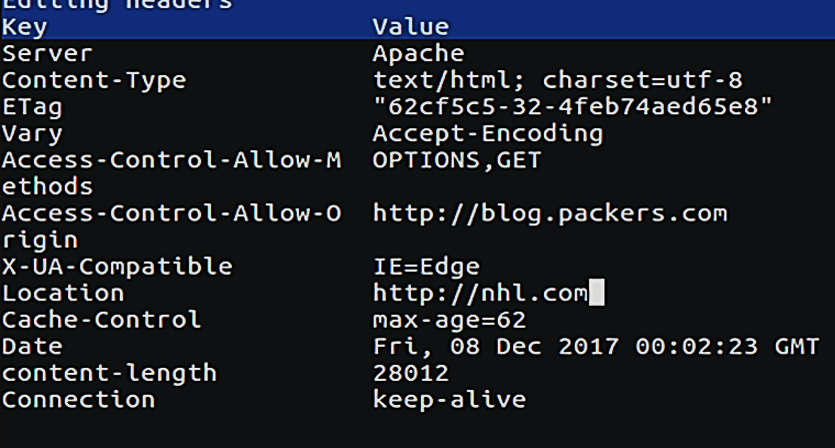
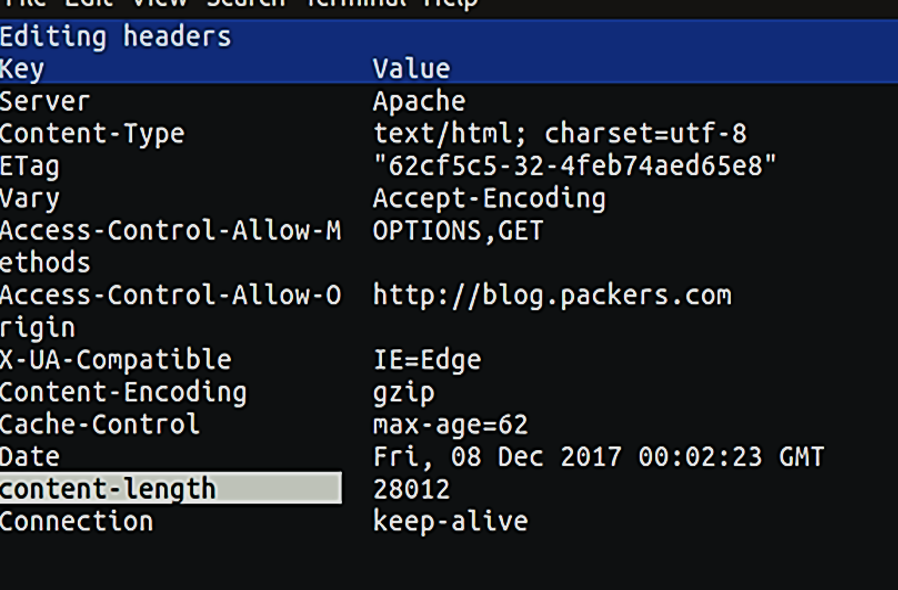


Linux has a program called mitmproxy that allows the poisoner to view the GET commands of all request packets that the victim is trying to access on the network – with interception from the raspberry pi.



The two screens display to the poisoner the header of the GET commands the victims is using. The packet requests are not going through to the router, but instead are intercepted by the raspberry pi.

The poisoner is able to look at the response packet and all web addresses that the victim is trying to access when they think that the router was unable to process that request (in truth, it can’t).



In the GET response, the poisoner is able to modify the content and website the victim is trying to access by replacing Content-Encoding with the Location in which reroutes the victim from going to blog.packers.com and guide the victim towards nhl.com upon their connection to the internet.