CSCD 434-040 - Network Security

Lab 2 - Scapy MITM - Due, October 6, 2025

October 1, 2025

Overview

Sometimes we need to make our own tools because the tools we need don't exist yet. In this lab you will use `scapy` to create a Man In The Middle (MITM) attack on unencrypted HTTP traffic! This lab setup will be similar to what you would use to intercept traffic if you were hosting a network to do nefarious things, and it is close to the paper we read earlier.

Use `scapy` to create a packet sniffer that prints out every time a page is visited using port 80. Take a screen shot of the code running while you visit http://httpforever.com/ or http://10.101.68.89/books in a web browser.

Forward and sniff

First, forward all traffic destined for port 80 to port 8080. We do this so that we know where to look for traffic to intercept. Make sure to put the following command on one line.

```
sudo iptables -t nat -A OUTPUT -p tcp -m owner ! --uid-owner root
--dport 80 -j DNAT --to :8080
```

Using `scapy`, `sniff()` all traffic and pick out just the traffic on destination port `8080`. Once you've done this you can then make a new packet, but change the destination port to `80`. Don't forget the payload, and make sure you don't copy the packet directly because that probably won't work.

Reminder: you'll need to drop reset packets (RST). Use the following command to drop RSTs that would otherwise cause your attack fail.

```
sudo iptables -I OUTPUT -p tcp --tcp-flags ALL RST,ACK -j DROP
sudo iptables -I OUTPUT -p tcp --tcp-flags ALL RST -j DROP
sudo iptables -I INPUT -p tcp --tcp-flags ALL RST -j DROP
sudo iptables -I INPUT -p tcp --tcp-flags ALL RST,ACK -j DROP
```

Questions:

1. Why can't one just change the port with 'scapy' to make it work?

2. What type of proxy did you create (e.g., regular, reverse, SOCKS5)? Hint: it's not one of the examples given in the previous sentence.
3. What in the first iptables command makes it such that our scapy created packets bound for port 80 are not mapped to 8080?
4. What would your program need to do to handle 443/HTTPS traffic?

Nefarious

Now that you can intercept traffic, let's do something nefarious. Silently drop any packets with a particular keyword ("frankenstein") in the `GET` request. You may need to try a few times with the same payload before you get censored, also try using `curl`.

Limitations:

You may only import `scapy` and use `scapy` functions.

Extra credit for all (3 pts)

Modify the original GET request to retrieve a different page. You may have to use `curl` to retrieve the page.

If you do the extra credit, make sure to have command line options to enable or disable the extra credit code. This should be explained in your README

Turn in

You'll create a tar file to turn in your lab. Make sure to include:

1. A PDF write up answering the questions about. Be sure to include any screen shots and explanations requested above.
2. Source code. Your python file **must** be named `lab2.py` and your code code must run. Put your name in the comments at the top of the file.
3. Include a bash or make file to add and remove the necessary `iptables`.
4. A detailed `README` so I now how to run your code and what it does. Include **all** `iptables` commands and when to use them in your `README`.