Assignment 3 **Tyler Wilding**
COSC 4426 Due Date: 02/10/16
Prof. Biocchi No Source Code
TA: – No Website

Packet Capturing

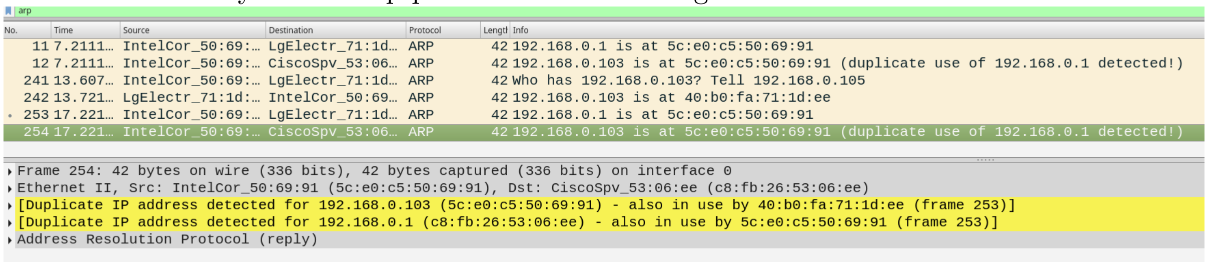
Packet capturing is very useful and not used for just gathering sensitive information, it is also used for testing many things that are networking related. As data is transferred amongst the network, the software picks up and logs the information and could even potentially decode it to show it in a more human readable format. This can especially be used to see how the data is while in motion, if it is encrypted or not, if it is being intercepted along the way.

ARP Poisoning

However, one of the issues when trying to capture packets in this assignment is that just simply looking at the packets on even a public WiFi with no encryption will not allow you to intercept information with just simple packet capturing software like wireshark. This is because networks are much more complex now, information is not sent directly between clients for the most part, instead data first travels to the router or switch which will redirect the data to its final destination or to the next hop. The router is not going to broadcast the information out to everyone on the network, only the relevant people. However, this might also be due to the limiting nature of wireless cards on laptops and phones.

This is where ARP (Address Resolution Protocol) poisoning comes in handy. This is a man in the middle attack that allows you to spoof to addresses in the network that you are the actual router. No one else in the network as a user sees a noticable difference, the packets are simple transferred to your computer first and then to the router, hence the man in the middle attack. Now that you have all packets going through your device you can use a program like wireshark to capture them.

We can actually see the arp poison in action through wireshark.



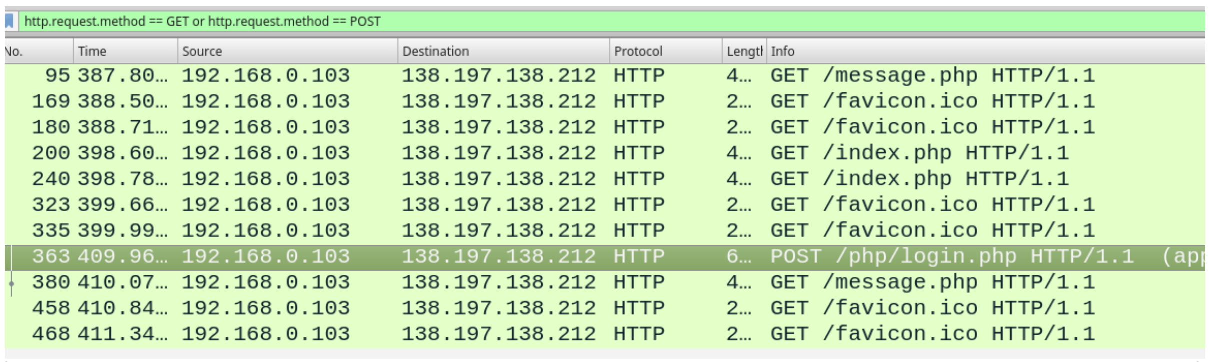
```
arp
No.  Time       Source              Destination         Protocol  Lengtl Info
  11 7.2111…   IntelCor_50:69:…   LgElectr_71:1d…    ARP       42 192.168.0.1 is at 5c:e0:c5:50:69:91
  12 7.2111…   IntelCor_50:69:…   CiscoSpv_53:06…    ARP       42 192.168.0.103 is at 5c:e0:c5:50:69:91 (duplicate use of 192.168.0.1 detected!)
 241 13.607…   IntelCor_50:69:…   LgElectr_71:1d…    ARP       42 Who has 192.168.0.103? Tell 192.168.0.105
 242 13.721…   LgElectr_71:1d:…   IntelCor_50:69…    ARP       42 192.168.0.103 is at 40:b0:fa:71:1d:ee
 253 17.221…   IntelCor_50:69:…   LgElectr_71:1d…    ARP       42 192.168.0.1 is at 5c:e0:c5:50:69:91
 254 17.221…   IntelCor_50:69:…   CiscoSpv_53:06…    ARP       42 192.168.0.103 is at 5c:e0:c5:50:69:91 (duplicate use of 192.168.0.1 detected!)

Frame 254: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
Ethernet II, Src: IntelCor_50:69:91 (5c:e0:c5:50:69:91), Dst: CiscoSpv_53:06:ee (c8:fb:26:53:06:ee)
[Duplicate IP address detected for 192.168.0.103 (5c:e0:c5:50:69:91) - also in use by 40:b0:fa:71:1d:ee (frame 253)]
[Duplicate IP address detected for 192.168.0.1 (c8:fb:26:53:06:ee) - also in use by 5c:e0:c5:50:69:91 (frame 253)]
Address Resolution Protocol (reply)
```

In this assignment I used ettercap to ARP poison and wireshark to capture the packets. I used Will's website Link for a non-SSL website, and mine, Link for a website with SSL.

Packet Output

Wireshark Configuration:
Capture Filter: host 192.168.0.103 (the ip of my phone)
Display Filter: http.request.method == GET or http.request.method == POST



```
http.request.method == GET or http.request.method == POST
No.  Time       Source          Destination      Protocol  Lengtl Info
  95 387.80…    192.168.0.103   138.197.138.212  HTTP      4…     GET /message.php HTTP/1.1
 169 388.50…    192.168.0.103   138.197.138.212  HTTP      2…     GET /favicon.ico HTTP/1.1
 180 388.71…    192.168.0.103   138.197.138.212  HTTP      2…     GET /favicon.ico HTTP/1.1
 200 398.60…    192.168.0.103   138.197.138.212  HTTP      4…     GET /index.php HTTP/1.1
 240 398.78…    192.168.0.103   138.197.138.212  HTTP      4…     GET /index.php HTTP/1.1
 323 399.66…    192.168.0.103   138.197.138.212  HTTP      2…     GET /favicon.ico HTTP/1.1
 335 399.99…    192.168.0.103   138.197.138.212  HTTP      2…     GET /favicon.ico HTTP/1.1
 363 409.96…    192.168.0.103   138.197.138.212  HTTP      6…     POST /php/login.php HTTP/1.1  (app
 380 410.07…    192.168.0.103   138.197.138.212  HTTP      4…     GET /message.php HTTP/1.1
 458 410.84…    192.168.0.103   138.197.138.212  HTTP      2…     GET /favicon.ico HTTP/1.1
 468 411.34…    192.168.0.103   138.197.138.212  HTTP      2…     GET /favicon.ico HTTP/1.1
```

Source	**Destination**	**Protocol**	**Info**	**Data**
192.168.0.103	138.197.138.212	HTTP	POST /php/login.php...	Frame 363: 609 bytes on wire (4872 bits), 609 bytes captured (4872 bits) on interface 0 Ethernet II, Src: LgElectr_71:1d:ee (40:b0:fa:71:1d:ee), Dst: IntelCor_50:69:91 (5c:e0:c5:50:69:91) Internet Protocol Version 4, Src: 192.168.0.103, Dst: 138.197.138.212 Transmission Control Protocol, Src Port: 38175 (38175), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 543 Hypertext Transfer Protocol HTML Form URL Encoded: application/x-www-form-urlencoded **Form item: "userInput" = "heywill"** Key: userInput Value: heywill **Form item: "passwordInput" = "installssl"** Key: passwordInput Value: installssl Form item: "submit" = "" Key: submit Value:

Source	**Destination**	**Protocol**	**Info**	**Data**
192.168.0.103	192.229.162.211	TLSv1.2	Encrypted Message	Frame 419: 85 bytes on wire (680 bits), 85 bytes captured (680 bits) on interface 0 Ethernet II, Src: LgElectr_71:1d:ee (40:b0:fa:71:1d:ee), Dst: IntelCor_50:69:91 (5c:e0:c5:50:69:91) Internet Protocol Version 4, Src: 192.168.0.103, Dst: 192.229.162.211 Transmission Control Protocol, Src Port: 52823 (52823), Dst Port: 443 (443), Seq: 1, Ack: 1, Len: 31 Secure Sockets Layer TLSv1.2 Record Layer: Encrypted Alert

As we can see with the above results, the non-SSL website has all its data sent unencrypted through the HTTP protocol, but the SSL secured website has everything encrypted over the Secure Socket Layer protocol and no data is available.