Introduction

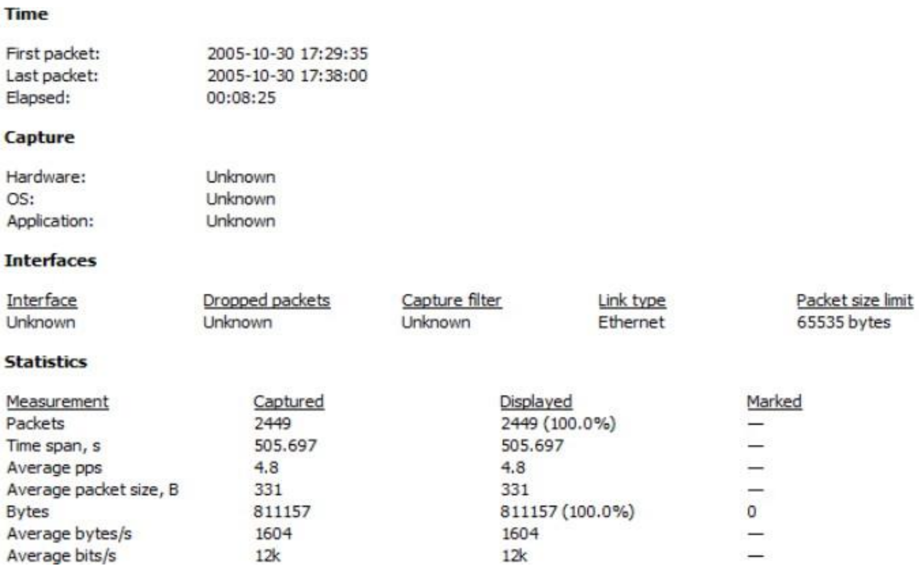
In this lab, a .pcap file was provided in which the tools Wireshark, Snort, and NetworkMiner can be used to perform intrusion detection on a professor’s home network.

Objective

The objective for this lab is to detect any sort of intrusion in accordance with the basic intrusion detection framework on the local network depicted in the packet capture file, along with identifying the type and extent of the attack.

Report

For this lab it is wise to start analysis with a general view of the system, much like the general practice of intrusion detection and investigation, which in this case the system is presented to us in a packet capture file. A general idea of traffic can best be surmised by analyzing basic capture statistics, such as session length and number of packets sent in this time. Provided in figure 1 is this basic information as presented by the statistics panel in Wireshark. The session lasts about eight and a half minutes, and with a total of 2449 packets, meaning that about 5 packets are sent per second on average. This infers a small internal network, supported by the fact that the total amount of bytes is 811157, which when divided by the time the capture file was taken over comes to about 1600, and further comes to 5 when divided by 331, the average packet size in bytes(*How many Packets per Second per port are needed to achieve Wire-Speed?*). As seen in figure 1, this information takes averages over all types of packets, so to get an even clearer image of normal and expected traffic for this network, we must investigate the types of protocols shown in the packet capture.

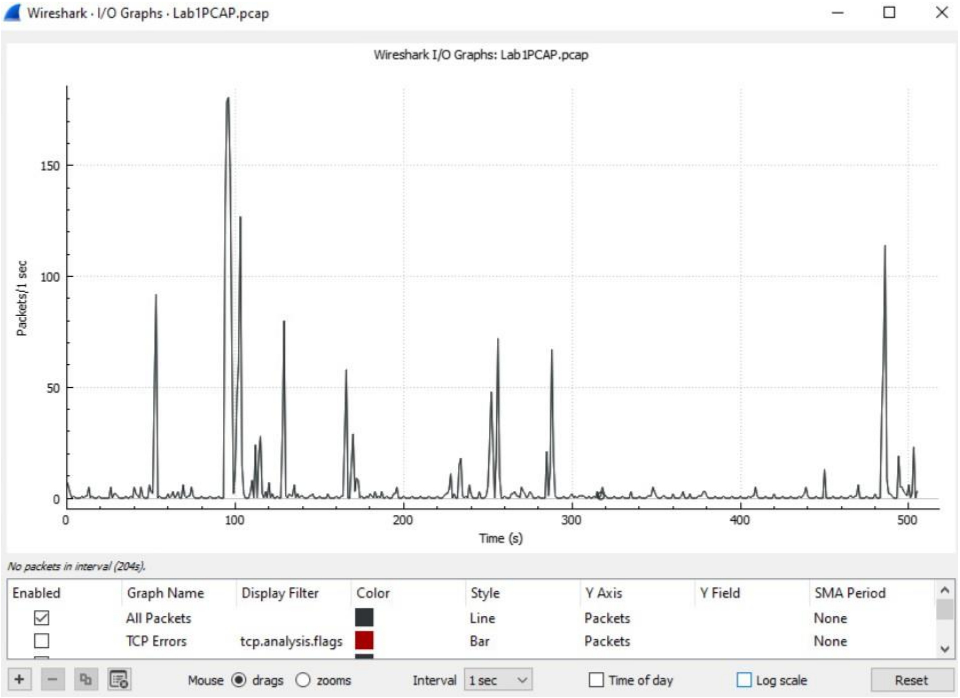


```
Time

First packet:          2005-10-30 17:29:35
Last packet:           2005-10-30 17:38:00
Elapsed:               00:08:25

Capture

Hardware:              Unknown
OS:                    Unknown
Application:           Unknown

Interfaces

Interface          Dropped packets      Capture filter      Link type         Packet size limit
Unknown            Unknown              Unknown             Ethernet          65535 bytes

Statistics

Measurement              Captured            Displayed             Marked
Packets                  2449                2449 (100.0%)         —
Time span, s             505.697             505.697               —
Average pps              4.8                 4.8                   —
Average packet size, B   331                 331                   —
Bytes                    811157              811157 (100.0%)       0
Average bytes/s          1604                1604                  —
Average bits/s           12k                 12k                   —
```

Figure 1: Capture file statistics

Wireshark allows sorting by protocol, and when searching explicitly for certain protocols, we can get a percentage of the specific protocol of all the traffic. The presence of ARP, DNS, HTTP, TCP, DHCP, and BROWSER protocols infers web connections through a router (*Network protocols: ARP, DNS, DHCP, HTTP and FTP*). With the presence of MDNS, NBNS, SSLv2, and TiVoConnect protocols, we can also infer that this network uses legacy wlan connection methods and services (*NetBIOS/NBNS*). This is further supported by the usage of FTP, where the host dials on, or rather, anonymous and automatically logs into, a linux-wlan server. This FTP server provided drivers and utilities for older wireless network adapters, meaning that the network in the packet capture used legacy wireless protocols and hardware, in which case signal leakage and unintended user connection are likely (*Linux-WLAN-ng*). At first the FTP server connections seemed anomalous, as without sensible time for user input a password is given to connect to the directory. Given that FTP is insecure and unencrypted, it seemed that these connections were IOCs, but looking at the further connections and network traffic patterns, it is simply part of the users ISP wireless connection system. To further understand normal network activity, we must also investigate user actions and connections. Much like anomaly-based IPS systems, to identify unusual activity one must identify normal activity, and while some IOCs are ubiquitous like signatures, to follow an intruder means to follow the host and their activity.

Looking at the I/O graph on Wireshark gives the first indicator that there is anomalous traffic. A spike in traffic occurs at packet 237 as shown in figure 2, and unlike the minor spikes where the user connected to a website briefly, such as the Microsoft home page of the browser, the number of packets sent in these 20 seconds is unlike the rest of the graph. The graph tells us that the network is mostly idle save for when the host connects to different web servers. However, packet 237 tells us that the spike occurs specifically during a connection to rbfcu.com, a bank website. This indicates to us that this connection may have been a target, as it would make sense as compared to all the other connections the host made, the information the bank would carry would be most desirable for an outside attacker unless other indicators tell us otherwise. Analyzing this connection tells us that the spike is due mainly because of extra ACK TCP segments sent by the host and Dup ACK segments sent back by 216.166.24.20, the IP address of rbfcu’s web server. The TCP Dup ACK and TCP Out-Of-Order gives us the first IOC of an attack, specifically session hijacking (*What is duplicate ack when does it occur?* 2018). Ideally, the host will send a SYN packet to the server, the server sends back a SYN/ACK, and the host will send back an ACK to complete the three way TCP handshake to establish a connection. If one could send an ACK statement before the host in the manner the server is expecting, an adversary can steal the session and have access to whatever information the server intended to send to the legitimate user as well as access to permissions granted to that session. Further analysis as shown in figure 3 gives us that the Dup ACK segments sent back by the server account for 5.6% of all packets, which also accounts for the host sent ACK statements that generated these server responses, giving reason as to the traffic spike during this connection.

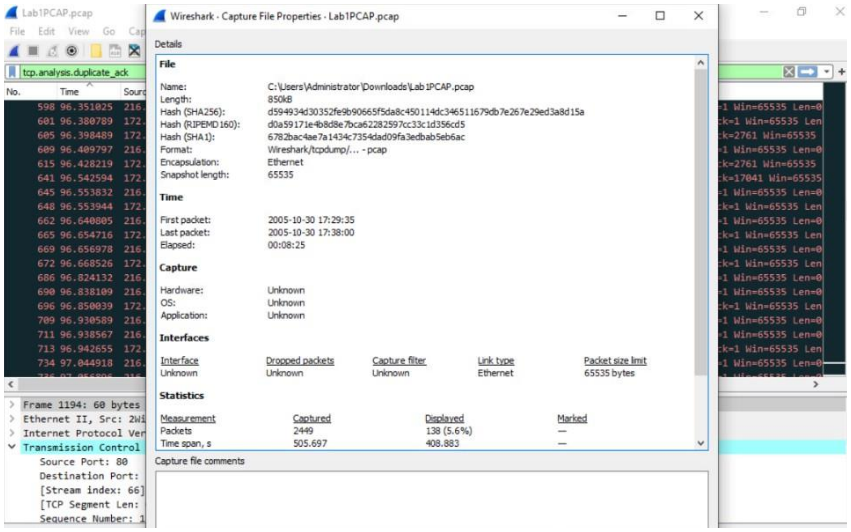


No packets in interval (204s).

Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Period
☑	All Packets		■	Line	Packets		None
□	TCP Errors	tcp.analysis.flags	🟥	Bar	Packets		None

Mouse ● drags ○ zooms Interval 1 sec □ Time of day □ Log scale

Figure 2: Wireshark I/O graph



tcp.analysis.duplicate_ack

```
No.    Time       Sourc
  598 96.351025  216.
  601 96.380789  172.
  605 96.398489  172.
  609 96.409797  216.
  615 96.428219  172.
  641 96.542594  172.
  645 96.553832  216.
  648 96.553944  172.
  662 96.640805  216.
  665 96.654716  172.
  669 96.656978  216.
  672 96.668526  172.
  686 96.824132  216.
  690 96.838109  216.
  696 96.850039  172.
  709 96.930589  216.
  711 96.938567  216.
  713 96.942655  172.
  734 97.044918  216.
```

```
=1 Win=65535 Len=0
ck=1 Win=65535 Len
ck=2761 Win=65535
=1 Win=65535 Len=0
ck=2761 Win=65535
ck=17041 Win=65535
=1 Win=65535 Len=0
ck=1 Win=65535 Len
=1 Win=65535 Len=0
ck=1 Win=65535 Len
=1 Win=65535 Len=0
ck=1 Win=65535 Len
=1 Win=65535 Len=0
=1 Win=65535 Len=0
ck=1 Win=65535 Len
=1 Win=65535 Len=0
=1 Win=65535 Len=0
ck=1 Win=65535 Len
=1 Win=65535 Len=0
```

```
> Frame 1194: 60 bytes
> Ethernet II, Src: 2Wi
> Internet Protocol Ver
v Transmission Control
    Source Port: 80
    Destination Port:
    [Stream index: 66]
    [TCP Segment Len:
    Sequence Number: 1
```

Wireshark · Capture File Properties · Lab1PCAP.pcap

Details

File

Name:	C:\Users\Administrator\Downloads\Lab1PCAP.pcap
Length:	850kB
Hash (SHA256):	d594934d30352fe9b90665f5da8c450114dc346511679db7e267e29ed3a8d15a
Hash (RIPEMD160):	d0a59171e4b8d8e7bca62282597cc33c1d356cd5
Hash (SHA1):	6782bac4ae7a1434c7354dad09fa3edbab5eb6ac
Format:	Wireshark/tcpdump/... - pcap
Encapsulation:	Ethernet
Snapshot length:	65535

Time

First packet:	2005-10-30 17:29:35
Last packet:	2005-10-30 17:38:00
Elapsed:	00:08:25

Capture

Hardware:	Unknown
OS:	Unknown
Application:	Unknown

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
Unknown	Unknown	Unknown	Ethernet	65535 bytes

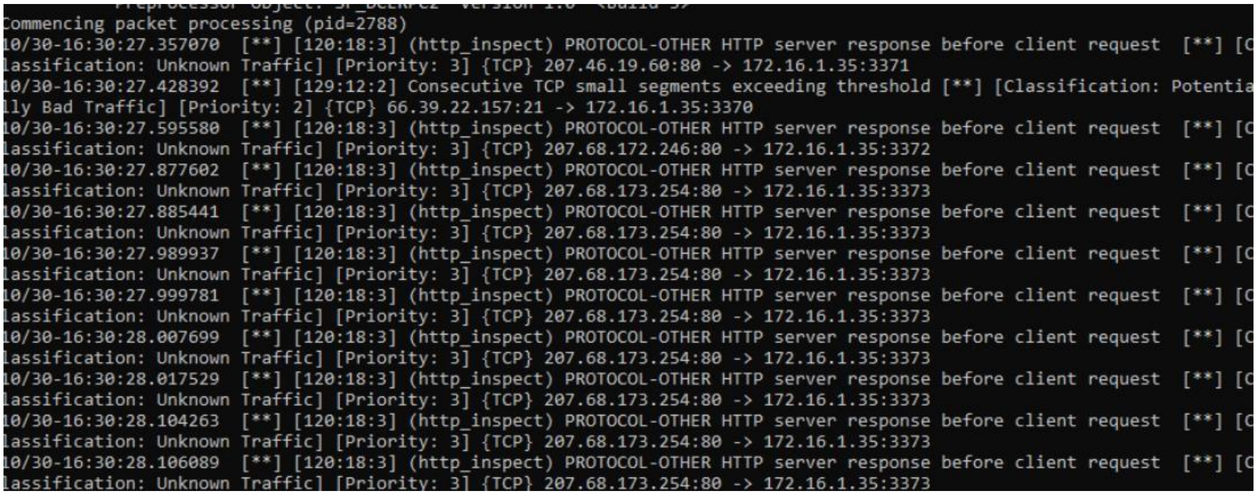
Statistics

Measurement	Captured	Displayed	Marked
Packets	2449	138 (5.6%)	—
Time span, s	505.697	408.883	—

Capture file comments

Figure 3: Dup ACK percentage

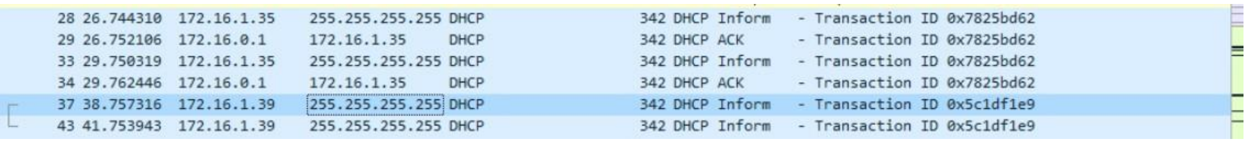
When providing the capture file to snort, many alerts are generated, and as seen in figure 4 all the errors relate to out of order TCP segments or rapid segments. While practically confirming the presence of session hijacking, the specific form of this attack is likely segment number guessing. To further investigate the possibility of TCP sequence number guessing, we can run the search command “tcp.seq && !tcp.ack && frame.time_delta > 0.01” which finds unexpected sequence number jumps, which may be indicative of sequence number guessing (*The TCP sequence prediction attack*). 20% of the packets show this behavior, and along with the steady rate of Dup ACK packets at about 12 per second during the main connection and the cluster of RST flags at packet 1209, this form of session hijacking is likely. With this, we have identified the event in the basic model of intrusion, with the action being stealing the connection by ACK segments and the target being the connection to the bank web server. To ascertain an attack, we must find the tool, vulnerability, and result of the event. To do this we have to understand what tools can be used on this network, and how someone could enter the network to exploit these vulnerabilities in the TCP connections.



```
Commencing packet processing (pid=2788)
10/30-16:30:27.357070  [**] [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP server response before client request  [**] [C
lassification: Unknown Traffic] [Priority: 3] {TCP} 207.46.19.60:80 -> 172.16.1.35:3371
10/30-16:30:27.428392  [**] [129:12:2] Consecutive TCP small segments exceeding threshold [**] [Classification: Potentia
lly Bad Traffic] [Priority: 2] {TCP} 66.39.22.157:21 -> 172.16.1.35:3370
10/30-16:30:27.595580  [**] [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP server response before client request  [**] [C
lassification: Unknown Traffic] [Priority: 3] {TCP} 207.68.172.246:80 -> 172.16.1.35:3372
10/30-16:30:27.877602  [**] [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP server response before client request  [**] [C
lassification: Unknown Traffic] [Priority: 3] {TCP} 207.68.173.254:80 -> 172.16.1.35:3373
10/30-16:30:27.885441  [**] [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP server response before client request  [**] [C
lassification: Unknown Traffic] [Priority: 3] {TCP} 207.68.173.254:80 -> 172.16.1.35:3373
10/30-16:30:27.989937  [**] [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP server response before client request  [**] [C
lassification: Unknown Traffic] [Priority: 3] {TCP} 207.68.173.254:80 -> 172.16.1.35:3373
10/30-16:30:27.999781  [**] [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP server response before client request  [**] [C
lassification: Unknown Traffic] [Priority: 3] {TCP} 207.68.173.254:80 -> 172.16.1.35:3373
10/30-16:30:28.007699  [**] [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP server response before client request  [**] [C
lassification: Unknown Traffic] [Priority: 3] {TCP} 207.68.173.254:80 -> 172.16.1.35:3373
10/30-16:30:28.017529  [**] [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP server response before client request  [**] [C
lassification: Unknown Traffic] [Priority: 3] {TCP} 207.68.173.254:80 -> 172.16.1.35:3373
10/30-16:30:28.104263  [**] [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP server response before client request  [**] [C
lassification: Unknown Traffic] [Priority: 3] {TCP} 207.68.173.254:80 -> 172.16.1.35:3373
10/30-16:30:28.106089  [**] [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP server response before client request  [**] [C
lassification: Unknown Traffic] [Priority: 3] {TCP} 207.68.173.254:80 -> 172.16.1.35:3373
```

Figure 4: Snort warnings

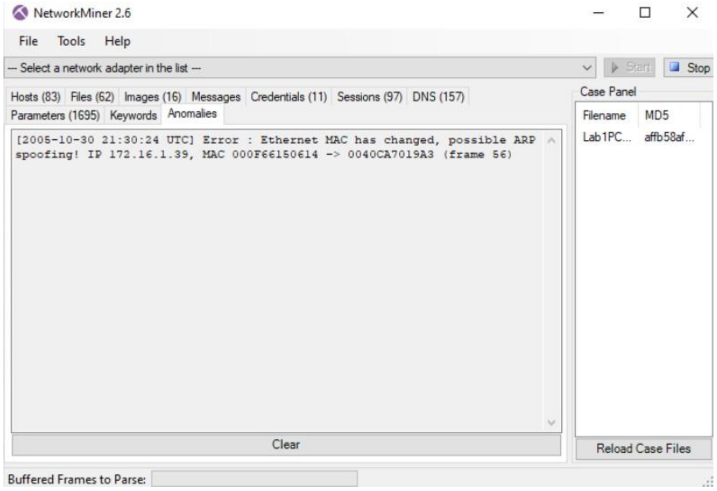
A key to finding this information is the observation that a user that is not the host would have to have the same IP address to hijack the session in the way we see in the packet capture. To investigate this, we must find more information on the home network and configurations of internet access. To do this, we can look at the very beginning and very end of the session, where we find that the main service provider is AOL with the account KaufmannUpstairs at americaonline.aol.com, though he accesses his accounts at other ISPs such as yahoo.com and msn.com. The fact that the session begins and ends with communication to the wlan FTP server and AOL services tells us that this network uses wireless connection and has one gateway router that manages the wireless connection to all devices on the subnet that access the internet. This is important to us because this means that the intruder must be on the wlan in the same subnet as the host behind the router to perform his attack. DHCP is used to assign IP addresses to new devices to a network, so searching for this protocol at the very beginning of the session can tell us about any active devices that were booted during the session (Fisher, *What is DHCP? (Dynamic Host Configuration Protocol)* 2020). As seen in figure 5, we identify an IP address of 172.16.1.39 and 172.16.1.35 as having this property. This means that the attacker can only infiltrate using one of these addresses from the outside of the network without creating another DHCP request.



```
28 26.744310  172.16.1.35    255.255.255.255 DHCP               342 DHCP Inform   - Transaction ID 0x7825bd62
29 26.752106  172.16.0.1     172.16.1.35     DHCP               342 DHCP ACK      - Transaction ID 0x7825bd62
33 29.750319  172.16.1.35    255.255.255.255 DHCP               342 DHCP Inform   - Transaction ID 0x7825bd62
34 29.762446  172.16.0.1     172.16.1.35     DHCP               342 DHCP ACK      - Transaction ID 0x7825bd62
37 38.757316  172.16.1.39    255.255.255.255 DHCP               342 DHCP Inform   - Transaction ID 0x5c1df1e9
43 41.753943  172.16.1.39    255.255.255.255 DHCP               342 DHCP Inform   - Transaction ID 0x5c1df1e9
```

Figure 5: DHCP of network devices

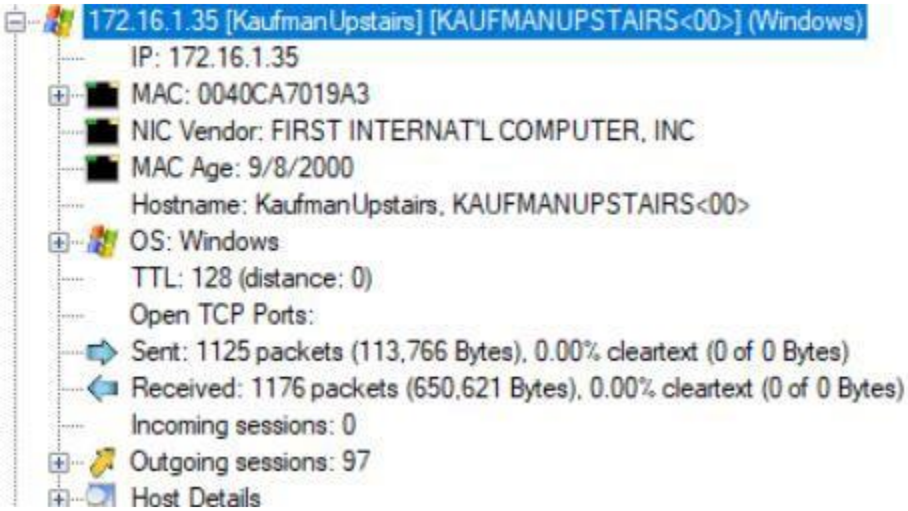
Using Network Miner to resolve end users and identify anomalies, we are alerted that there is possible arp spoofing due to the changing of the mac address for this device. In figure 6, we can see that the mac address changes from 00:0f:66:15:06:14 to 00:40:ca:70:19:a3, where now the mac address of the .39 IP is the same as the of 172.16.1.35, the host. This means that at the next arp query at packet 225 will result in the IP address of the host being assigned to the .35, so after this moment there are two machines with the same IP address. Along with this the host information including the device name is resolved to two devices which as seen in figure 7, is KaufmanUpstairs. This accounts for the Dup ACK packets if two machines with the same IP are attempting to connect to the same server at the same time, which makes sense if we apply it to a traditional session hijacking attack, where one machine attempts to intercede on another’s tcp handshake. To further map out the attack, we need to figure out the vulnerability and tools used to exploit it.



```
[2005-10-30 21:30:24 UTC] Error : Ethernet MAC has changed, possible ARP
spoofing! IP 172.16.1.39, MAC 000F66150614 -> 0040CA7019A3 (frame 56)
```

Filename	MD5
Lab1PC...	affb58af...

Figure 6: Mac address change

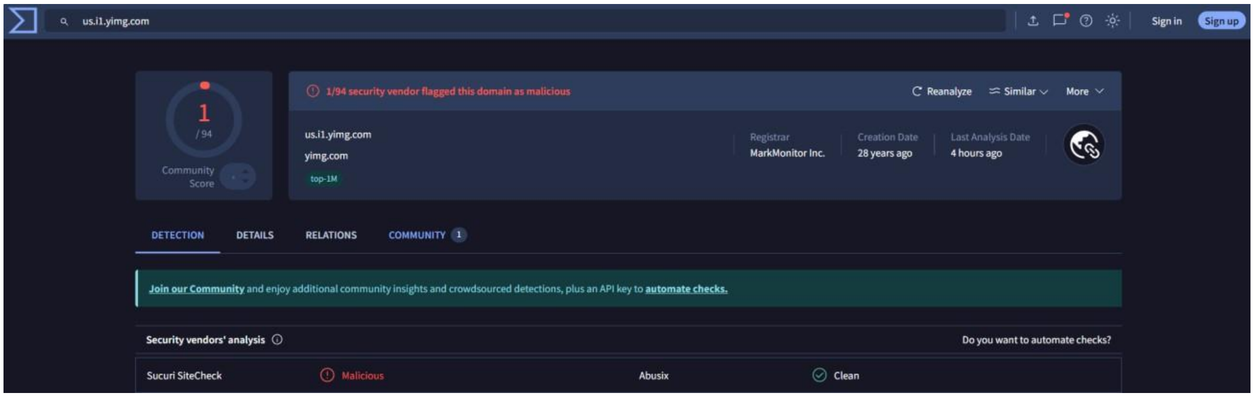


```
172.16.1.35 [KaufmanUpstairs] [KAUFMANUPSTAIRS<00>] (Windows)
    IP: 172.16.1.35
    MAC: 0040CA7019A3
    NIC Vendor: FIRST INTERNAT'L COMPUTER, INC
    MAC Age: 9/8/2000
    Hostname: KaufmanUpstairs, KAUFMANUPSTAIRS<00>
    OS: Windows
    TTL: 128 (distance: 0)
    Open TCP Ports:
    Sent: 1125 packets (113,766 Bytes), 0.00% cleartext (0 of 0 Bytes)
    Received: 1176 packets (650,621 Bytes), 0.00% cleartext (0 of 0 Bytes)
    Incoming sessions: 0
    Outgoing sessions: 97
    Host Details
```

Figure 7: Host information

The vulnerability is a configuration vulnerability, where both an attacker can connect without authorization and a non-static arp table. While not entirely dangerous on its own, both vulnerabilities combine to create an open pathway for attackers to arp spoof on the wlan. With such a small network, it would not be burdensome to have a static arp table, meaning that mac address spoofing would never be resolved to an IP change (*Configuring static ARP table entries for mapping IP addresses to MAC addresses*). However, many times arp tables need to be dynamic, so more security on initial connection is needed during the initial DHCP connection and from unauthorized devices connecting to the wlan(*How to secure a network from DHCP attacks*). With both configurations, the router at 172.16.1.1 allows an attacker to easily begin an action against a target, but the tool used to charge the mac address is vital information to finish mapping out the attack.

Due to the presence of carriage returns in http connections and the first home page the user connects to being msn.com, we can infer that the host operating system is a version of Windows (Freitag, *What is the difference between ASCII CHR(10) and CHR(13)* 2023). This is supported by NetworkMiner, that attributed Windows to being the host OS. For the attacker, NetworkMiner attributes Linux to the 70.245.59 subnet where the attacker likely resided before spoofing his way to the network as seen in figure 8, meaning that the tool used to change the arp address is likely a linux command. It only takes a simple command in a gui to change a mac address on both linux and windows, and this command that the attacker likely used is within a downloadable toolkit like macChanger or ettercap (McKay, *How to permanently change your MAC address on linux* 2023). Now that we know this, the last part to the attack framework to analyze is the scope of damage, or rather, the unauthorized result.



1 / 94

1/94 security vendor flagged this domain as malicious

us.i1.yimg.com
yimg.com
top-1M

Registrar: MarkMonitor Inc.
Creation Date: 28 years ago
Last Analysis Date: 4 hours ago

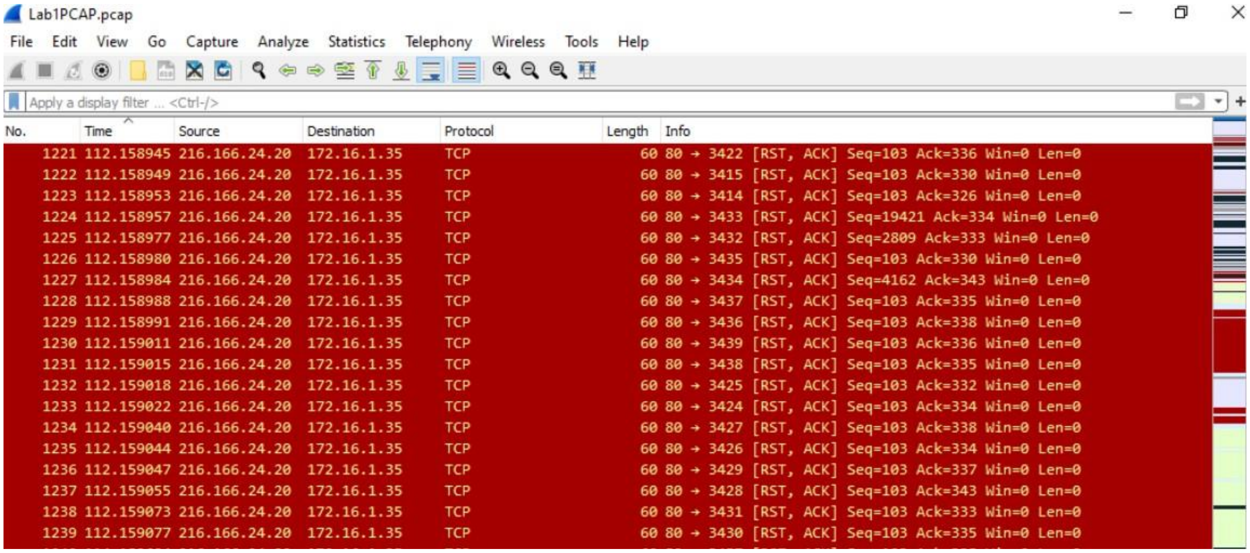
Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Security vendors' analysis — Do you want to automate checks?

Sucuri SiteCheck	Malicious	Abusix	Clean

Figure 8: VirusTotal malicious activity report of linux subnet

We can follow the attacker’s efforts by tracking when the Dup ACK packets were sent and knowing that most of these were sent to rbfcu.com during the data spike, we must look towards the end of this connection to see if the hijacking was successful. After the connection at frame 1209 shown in figure 9, there is a series of RST flags once the host navigates away from rbfcu.com. This long wall of RST flags ties to the connections made during the rbfcu connections, and the initial tcp after the DNS query to the website also has an RST in this cluster. This along with the lack of a FIN flag shows that none of the attacker sessions where successfully completed as the RST is sent when a connection is made or ended abnormally (Cao, *Understanding TCP flags syn ack rst fin urg psh* 2023). The fact that all these RST flags are all together is because once the user navigates off their connection normally, the server has many unaccounted-for ACK statements that need to be closed. After this connection, there does exist more Dup ACK statements albeit more sparsely, showing that the intruder is still attempting the same attack but missed the window of opportunity to complete a session with the bank website. During the SSLv3 communications, the attacker was also unable to session hijack as it is password protected and encrypted by negotiation with the true host. The lack of get requests over HTTP after the secure session also means there is no data exfiltration. Now to complete an analysis of the full incident, we must follow the attacker to objective over the network.



```
No.   Time       Source        Destination  Protocol  Length Info
 1221 112.158945 216.166.24.20 172.16.1.35  TCP       60 80 → 3422 [RST, ACK] Seq=103 Ack=336 Win=0 Len=0
 1222 112.158949 216.166.24.20 172.16.1.35  TCP       60 80 → 3415 [RST, ACK] Seq=103 Ack=330 Win=0 Len=0
 1223 112.158953 216.166.24.20 172.16.1.35  TCP       60 80 → 3414 [RST, ACK] Seq=103 Ack=326 Win=0 Len=0
 1224 112.158957 216.166.24.20 172.16.1.35  TCP       60 80 → 3433 [RST, ACK] Seq=19421 Ack=334 Win=0 Len=0
 1225 112.158977 216.166.24.20 172.16.1.35  TCP       60 80 → 3432 [RST, ACK] Seq=2809 Ack=333 Win=0 Len=0
 1226 112.158980 216.166.24.20 172.16.1.35  TCP       60 80 → 3435 [RST, ACK] Seq=103 Ack=330 Win=0 Len=0
 1227 112.158984 216.166.24.20 172.16.1.35  TCP       60 80 → 3434 [RST, ACK] Seq=4162 Ack=343 Win=0 Len=0
 1228 112.158988 216.166.24.20 172.16.1.35  TCP       60 80 → 3437 [RST, ACK] Seq=103 Ack=335 Win=0 Len=0
 1229 112.158991 216.166.24.20 172.16.1.35  TCP       60 80 → 3436 [RST, ACK] Seq=103 Ack=338 Win=0 Len=0
 1230 112.159011 216.166.24.20 172.16.1.35  TCP       60 80 → 3439 [RST, ACK] Seq=103 Ack=336 Win=0 Len=0
 1231 112.159015 216.166.24.20 172.16.1.35  TCP       60 80 → 3438 [RST, ACK] Seq=103 Ack=335 Win=0 Len=0
 1232 112.159018 216.166.24.20 172.16.1.35  TCP       60 80 → 3425 [RST, ACK] Seq=103 Ack=332 Win=0 Len=0
 1233 112.159022 216.166.24.20 172.16.1.35  TCP       60 80 → 3424 [RST, ACK] Seq=103 Ack=334 Win=0 Len=0
 1234 112.159040 216.166.24.20 172.16.1.35  TCP       60 80 → 3427 [RST, ACK] Seq=103 Ack=338 Win=0 Len=0
 1235 112.159044 216.166.24.20 172.16.1.35  TCP       60 80 → 3426 [RST, ACK] Seq=103 Ack=334 Win=0 Len=0
 1236 112.159047 216.166.24.20 172.16.1.35  TCP       60 80 → 3429 [RST, ACK] Seq=103 Ack=337 Win=0 Len=0
 1237 112.159055 216.166.24.20 172.16.1.35  TCP       60 80 → 3428 [RST, ACK] Seq=103 Ack=343 Win=0 Len=0
 1238 112.159073 216.166.24.20 172.16.1.35  TCP       60 80 → 3431 [RST, ACK] Seq=103 Ack=333 Win=0 Len=0
 1239 112.159077 216.166.24.20 172.16.1.35  TCP       60 80 → 3430 [RST, ACK] Seq=103 Ack=335 Win=0 Len=0
```

Figure 9: Cluster of RST packets

By following time stamps and sessions, we can tell the story of the actions that led to the observed traffic. However, before doing so we need to have a full understanding of the environment that the actions are taking place in. As seen in figure 10, we have a small personal network capable of wlan connection through a central router that also provides name resolution capabilities. There were 3 host addresses on this network initially with two communicating with the gateway router, but now that an attacker has connected to the wlan and arp spoofed, the same number of devices are communicating but the router being a network layer device only transmits traffic as if it were one device.

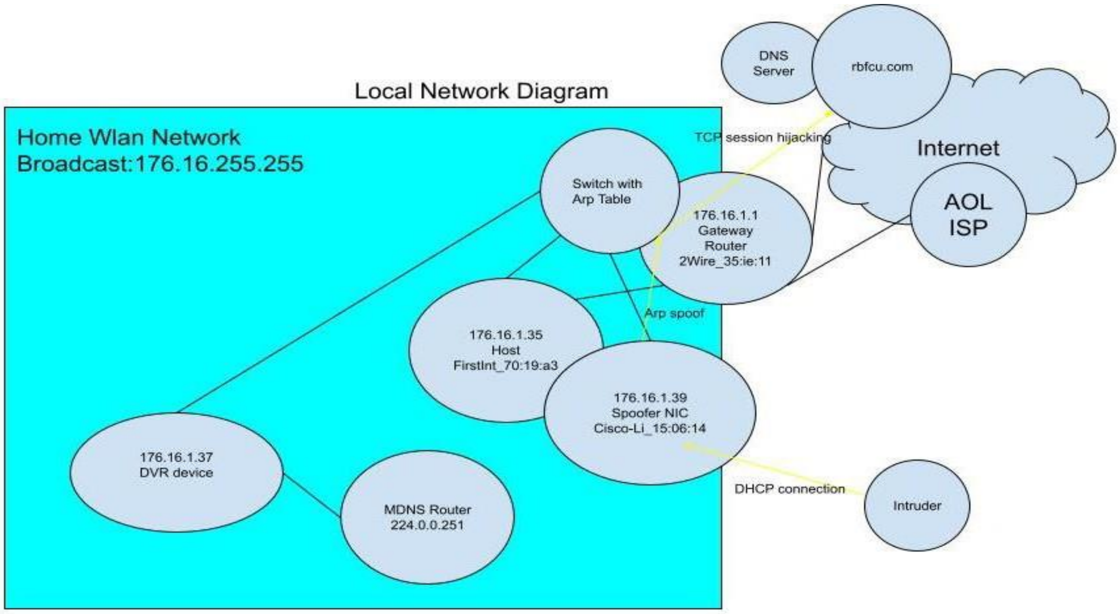


Figure 10: Home network diagram

With this, we can follow from start to finish the story of this incident. The attacker likely sees TiVoConnect packets from the nearby wlan network as these packets are beacons. From this he connects to the router and spoofs or is assigned the IP address of 172.16.1.39, the address of one of Kaufmann's NIC by means of the DHCP protocol. He then spoofs his mac address by using linux commands through a toolkit, changing his mac address along with the device name that NBNS resolves along with ARP to be equivalent to the host device, being assigned the IP address of 172.16.1.35. During this process the host connects to his ISP for wlan service and connects to the main page of his browser. The host then connects to his bank website to the home page. The attacker in the hopes of gaining access to an authorized session tries to hijack a tcp connection so that he can take advantage of the host having already provided his credentials. However, he is unsuccessful, generating many out-of-order warnings from the server. The user, meanwhile, after spending time on the home page logs into his account. The website refreshes and now a connection is made using SSL i.e. HTTPS unlike the earlier HTTP used for the home screen. The attacker is unsuccessful at guessing the password to this encrypted session, but sometime later when the host navigates to utsa.com, he loses his window of opportunity and RST flags are sent to terminate all the abnormal connections. The user navigates to some mail sites after logging into utsa.com (where he changes his Ialab4(Fall05).doc file and his picture), but the attacker is hardly interested and disconnects. The user connects to yahoo for email services with his email of robkaufmaniii@sbcglobal.net. The host then terminates his AOL ISP connection and ends his packet capture. With this, we can identify the full incident, from attacker to attack and to objective. As seen in figure 11, this packet capture contains a malicious event and can be modeled using standard practices.

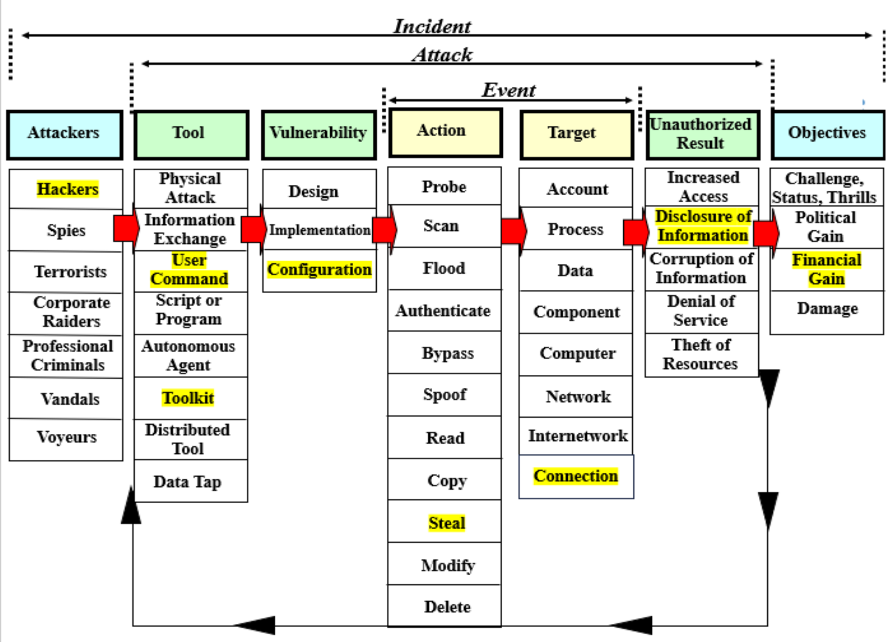


Figure 11: Basic intrusion model application

Conclusion

Performing intrusion detection on this packet capture file required usage of industry standards for attack modeling such as the basic model for computer incidents. By focusing first on the event, the incident becomes clearer when also considering the environment. It is always easier to collect information first and sketch things out, then dial in on the anomalous activity. When performed on this lab, the attack becomes clear from start to finish, including the story of the local attacker trying to gain access to a bank.

Bibliography

Bernatavičius, M. (2018, January 11). *What is duplicate ack when does it occur?*. Stack Overflow. https://stackoverflow.com/questions/48148820/what-is-duplicate-ack-when-does-it-occur

Cao, D. (2023, September 21). *Understanding TCP flags syn ack rst fin urg psh*. howtouselinux. https://www.howtouselinux.com/post/tcp-flags

Configuring static ARP table entries for mapping IP addresses to MAC addresses. Configuring Static ARP Table Entries For Mapping IP Addresses to MAC Addresses | Junos OS | Juniper Networks. (n.d.). https://www.juniper.net/documentation/us/en/software/junos/multicast-l2/topics/task/interfaces-configuring-static-arp-table-entries.html

Fisher, T. (2020, December 13). *What is DHCP? (Dynamic Host Configuration Protocol)*. Lifewire. https://www.lifewire.com/what-is-dhcp-2625848

Freitag, P. (2023, December 7). *What is the difference between ASCII CHR(10) and CHR(13)*. Pete Freitag. https://www.petefreitag.com/blog/ascii-chr-10-chr-13/

How many Packets per Second per port are needed to achieve Wire-Speed?. CEC Juniper Community. (n.d.). https://supportportal.juniper.net/s/article/How-many-Packets-per-Second-per-port-are-needed-to-achieve-Wire-Speed?language=en_US

How to secure a network from DHCP attacks. CIOReview. (n.d.). https://www.cioreview.com/news/how-to-secure-a-network-from-dhcp-attacks-nid-15482-cid-21.html

Linux-WLAN-ng. linux-wlan-ng - Debian Wiki. (n.d.). https://wiki.debian.org/linux-wlan-ng

McKay, D. (2023, March 31). *How to permanently change your MAC address on linux.* How. https://www.howtogeek.com/880124/how-to-permanently-change-your-mac-address-on-linux/

NetBIOS/NBNS. NetBIOS/NBNS - Wireshark Wiki. (n.d.). https://wiki.wireshark.org/NetBIOS/NBNS

Network protocols: ARP, DNS, DHCP, HTTP and FTP. - MCSI Library. (n.d.). https://library.mosse-institute.com/articles/2022/05/network-protocols-the-foundation-of-digital-communication-arp-dns-dhcp-http-and-ftp/network-protocols-the-foundation-of-digital-communication-arp-dns-dhcp-http-and-ftp.html

The TCP sequence prediction attack. (n.d.). https://www.idc-online.com/technical_references/pdfs/data_communications/TCP_Sequence_Prediction_Attack.pdf