Active Sniffing

Technical Report

Mark Ennis
November 17th, 2022

Introduction

Active sniffing is an attack where specially crafted packets are sent to different machines to exfiltrate data, and set up a Machine-in-the-Middle attack. In a switched environment, it is more difficult for an attacker to intercept or manipulate traffic that is not meant for them. The switch directs traffic from the sender to the receiver, as opposed to a hub which broadcasts all traffic to all hosts. The switch, and hosts on the network, keep track of where other hosts are by maintaining a list of MAC or Ethernet addresses, and the IP address they are paired to. These pairings are communicated to other machines on the network using Address Resolution Protocol (ARP) messages. Local copies of these pairings are stored in an ARP cache. An attacker can use techniques such as MAC flooding to begin receiving broadcast messages. MAC flooding is the process of sending an extremely large number of messages with non-existent MAC addresses. Eventually, the ARP cache of the switch will fill up. In some implementations, this causes the switch to begin broadcasting messages. Essentially, this transforms a switch into a hub. This report shall focus on two exploits that rely on ARP poisoning and a penetration testing tool called Ettercap. It shall also cover various tools and techniques for identifying ARP poisoning and spoofing.

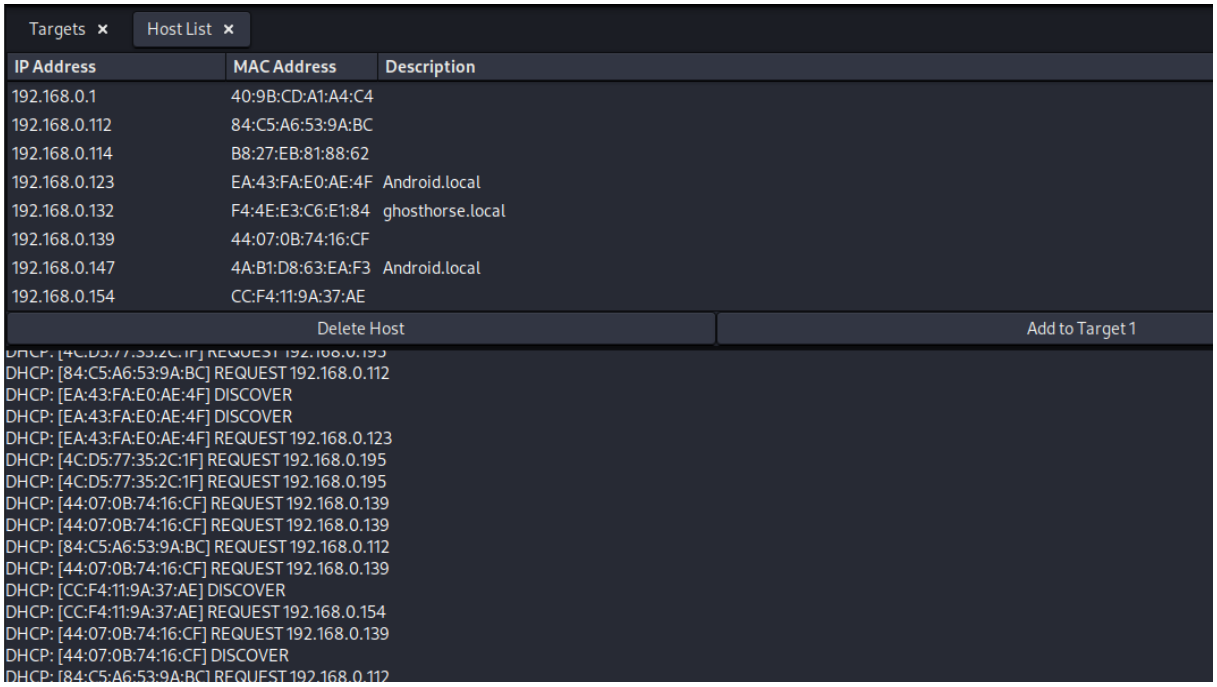
A detailed list of the files contained within the report submission can be found in Appendix A.

Body

ARP poisoning is the process of tricking machines on a network into associating the wrong MAC address with an IP address. This is accomplished by broadcasting a large number of ARP requests telling other machines on the network that one or more IP addresses are located at the attackers' MAC address. Since ARP does not provide an authentication mechanism for requests on the network, a machine's local ARP caches are updated and they will begin to direct traffic to the attacker's machine. Once an attacker begins receiving this traffic, they can easily perform several different attacks. The attacker can operate as a Machine-in-the-Middle (MitM). Any unencrypted traffic will be able to be read by the attacker. This is useful for credential harvesting and data exfiltration. An attacker can also drop packets that are being sent to it, which allows ARP poisoning to be used in Denial of Service (DoS) attacks.

Ettercap

Ettercap is a traffic sniffer that can use ARP poisoning and other techniques to perform credential harvesting, DNS spoofing and MitM attacks on a local network. It has a graphical user interface, and allows an attacker to perform ARP spoofing in a few clicks. Using both active and passive techniques it can map a network and identify running machines.



The host list can be built by passively listening to DHCP and ARP traffic on the network, or it can perform an ARP scan.

Once Ettercap has a list of hosts, it can begin to select targets. Ettercap puts targets into two groups: Target 1 and Target 2. Once an attack is launched, Ettercap acts as the MitM between the hosts in the two targets. For example, imagine one host (192.168.1.1) has been added to Target 1 and another host (198.168.1.2) has been added to Target 2. Once Ettercap performs an ARP poisoning attack, all the traffic between 192.168.1.1 and 192.168.1.2 will go through Ettercap. If either of those hosts to 192.168.1.3, it would not go through Ettercap.

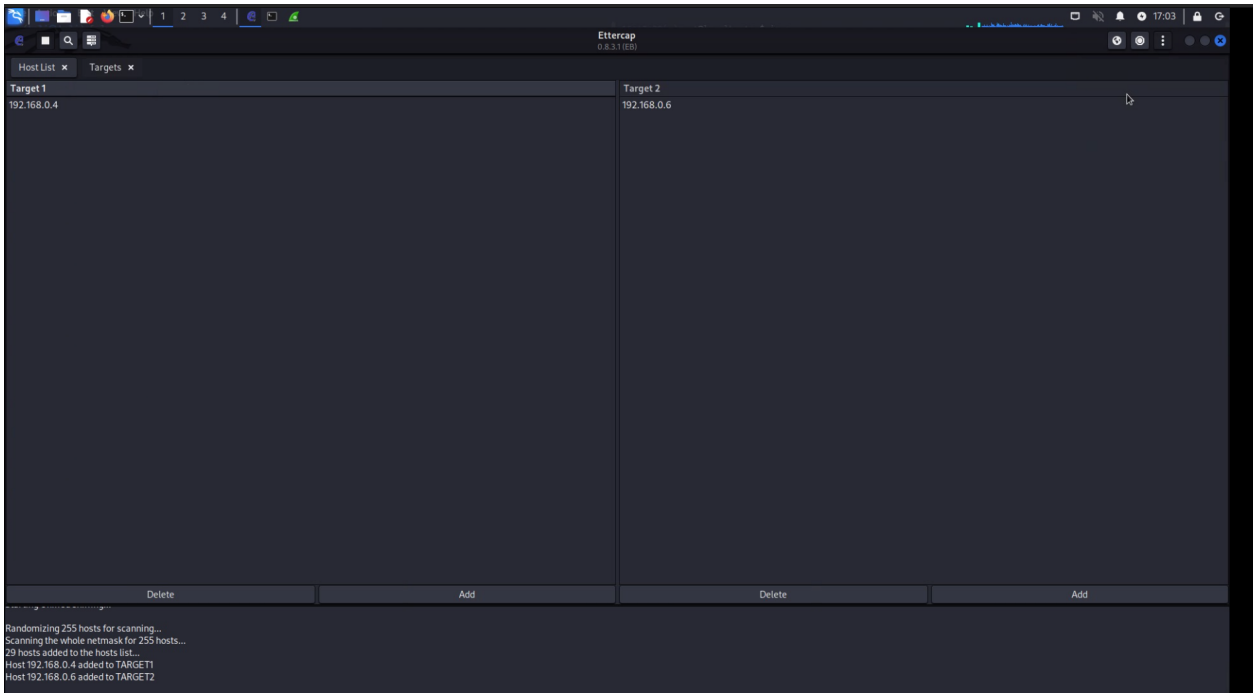
By default, all hosts are in both targets. This means that all traffic will go through Ettercap. If a target contains a network gateway, all traffic entering or exiting the network would also go through Ettercap.

Credential Harvesting Attack

Given the ease with which a MitM attack can be launched using Ettercap, credential harvesting is fairly straightforward. Ettercap has to identify two groups of hosts on the networks as targets, and then perform an ARP poisoning attack. Then, whenever a host in the Target 1 group sends a request to a host in the Target 2 group, it passes through Ettercap. This allows Ettercap to capture and inspect all the traffic passing between these two groups.

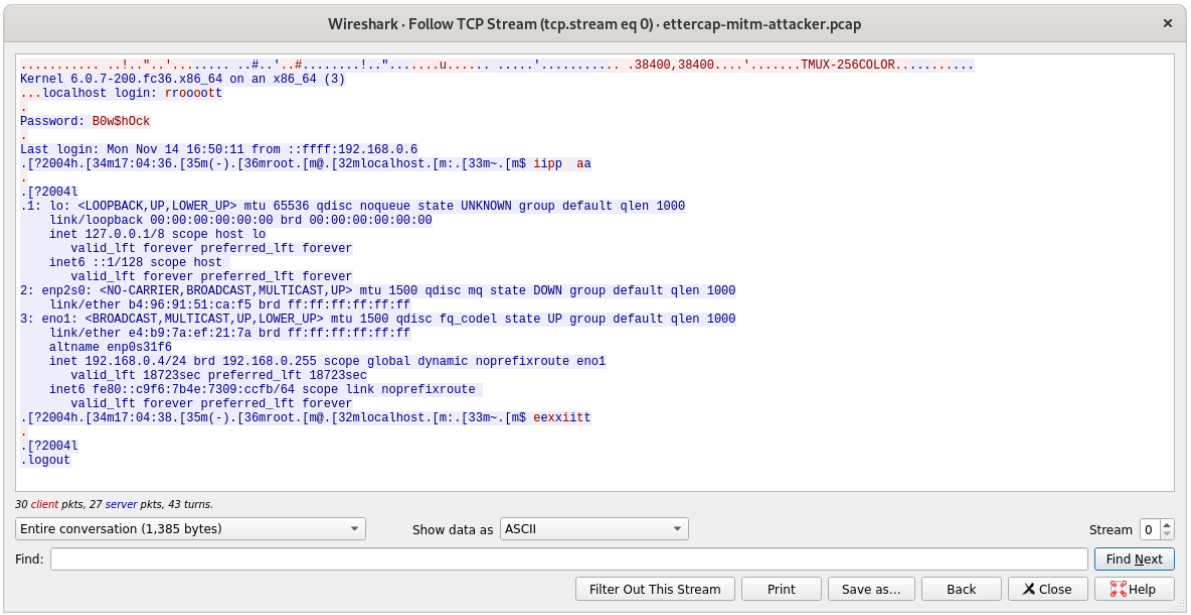
To being this attack, launch Ettercap. Ettercap requires root privileges and the graphical interface can be launched from the command line by running: `sudo ettercap -G`. From there, an active network ARP scan can be performed by selecting Hosts > Scan for hosts from the menu. The hosts can be viewed by then selecting Hosts > Hosts list from the menu. Hosts can be added to the target groups by right-clicking on them and selecting Add to Target 1 or 2. Add

at least one host to each target, then verify the targets by selecting Targets > Current Target from the menu.



After selecting targets, Ettercap can run an ARP poisoning attack on the machines by selecting Arp Poisoning… from the MitM menu. ARP messages are sent to all the hosts in Target 1 that spoof the MAC address of all the hosts in Target 2 to match the machine running Ettercap. The same is then done for Target 2. At this point, all traffic will pass through Ettercap.

Capturing traffic on the Ettercap machine shows the traffic passing between the two. If it is unencrypted, it can be viewed in the packet capture.



It is worth noting that if the network gateway is a host in either of the target groups, Ettercap will intercept all traffic going out of the network from the other group. The machine running Ettercap will be able to view any unencrypted traffic.

DNS Spoofing Attack

Ettercap provides several modules that extend its base functionality. One of these modules, `dns_spoof`, enables the redirection of HTTP/S traffic by performing DNS spoofing. DNS spoofing is when an attacker uses fraudulent Domain Name records to send a victim to the wrong IP address.

The first step to perform this attack is to add the configuration to the `/etc/ettercap/etter.dns` configuration file. Each domain name that is being spoofed has three to four parameters separated by whitespace. The three parameters are:

- Domain name - The domain name being spoofed
- DNS Query type - The type of DNS record being spoofed. This could be an alias record, of type A or AAAA to support IPv6, a PTR record for reverse DNS lookup, or other less common query types.
- Redirect address - The IP address where the record will point now.
- Time to live - An optional TTL value.

```
## Redirect facebook to this machine
facebook.com            A       192.168.0.5
*.facebook.com          A       192.168.0.5
www.facebook.com        PTR     192.168.0.5
```

Once DNS spoofing configuration is done, Ettercap can be started and the DNS spoofing plugin needs to be activated. The plugin page can be found by clicking on the menu in the top right-hand corner, and selecting Plugins > Manage Plugins. Then, find the `dns_spoof` entry in the table, right click on it, and select Activate. An asterix should appear to the left of the plugin entry in the table.

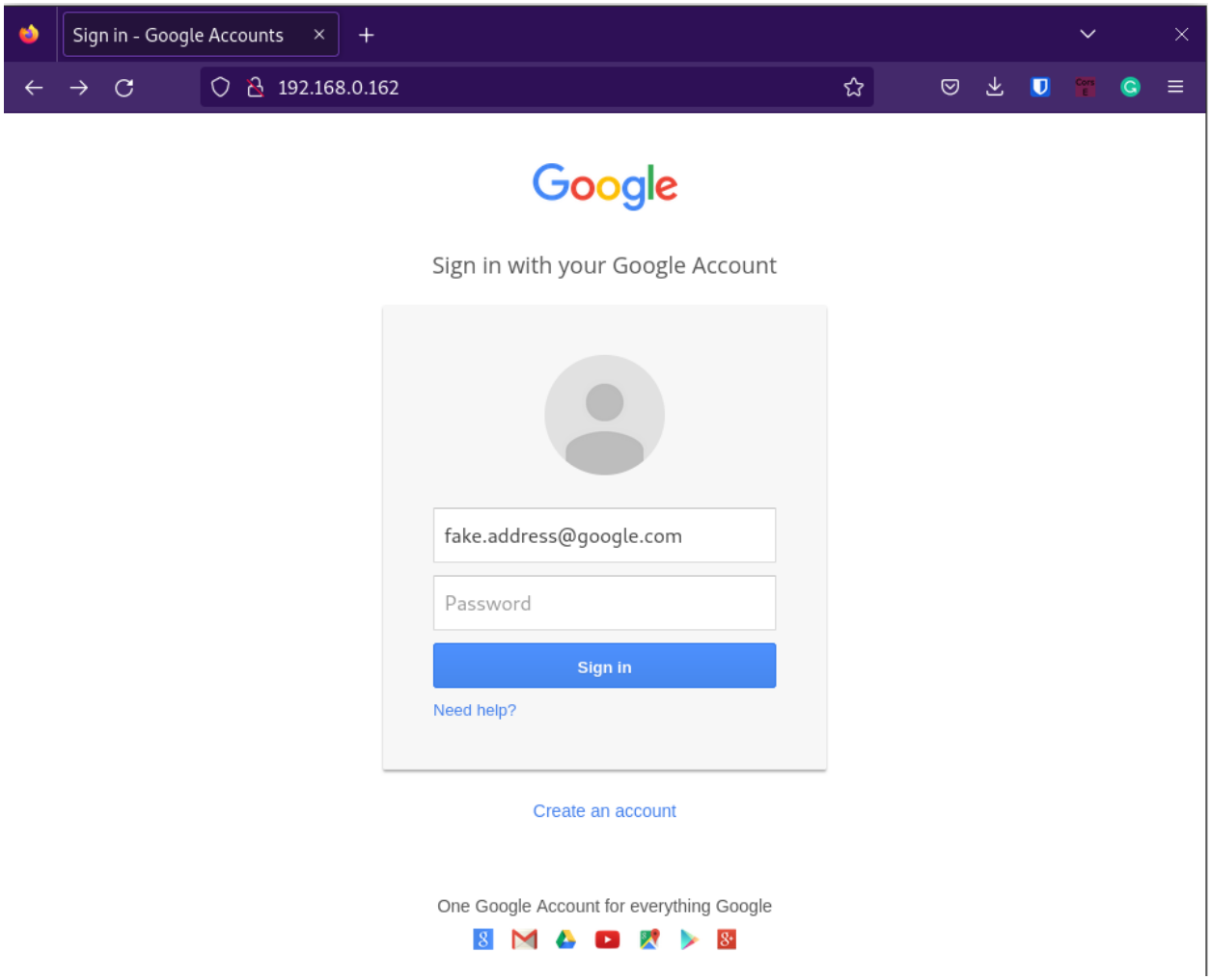
```
    chk_poison      1.1     Check if the poisoning had success
*   dns_spoof       1.3     Sends spoofed dns replies
    dos_attack      1.0     Run a d.o.s. attack against an IP address
    dummy           3.0     A plugin template (for developers)
```

The next step is to identify hosts on the network, similar to the previous attack. From the menu, select Hosts > Scan for hosts. This will initiate an ARP scan. Given enough time listening passively, this step could be ignored. Once the hosts have been gathered, the entire LAN can have its ARP cache poisoned. This can be done by not specifying any targets before selecting the MitM menu, and selecting the Arp poisoning… option. Now, Ettercap can intercept all DNS traffic. If the DNS query does not match any of the configured spoof records, the DNS request is forwarded as normal. If the query does match the configuration, the spoofed IP address will be returned instead. Then, when a machine on the network attempts to visit the spoofed domain, they will making HTTP connections to the wrong endpoint.

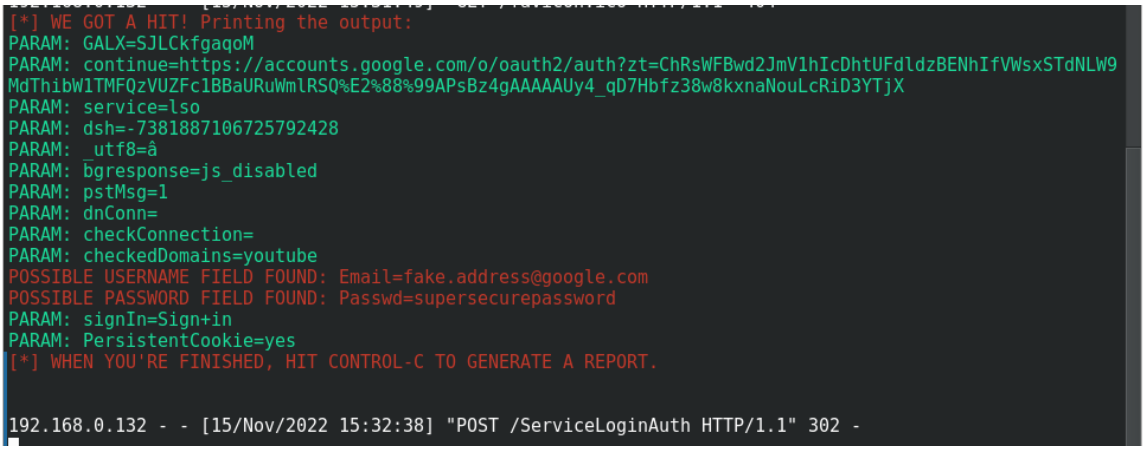
```
Frame 9271: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits)
Ethernet II, Src: Dell_ef:21:7a (e4:b9:7a:ef:21:7a), Dst: Dell_ef:57:fe (e4:b9:7a:ef:57:fe)
Internet Protocol Version 4, Src: 192.168.0.4, Dst: 192.168.0.5
Transmission Control Protocol, Src Port: 60674, Dst Port: 80, Seq: 1, Ack: 1, Len: 76
Hypertext Transfer Protocol
    GET / HTTP/1.1\r\n
    Host: facebook.com\r\n
    User-Agent: curl/7.82.0\r\n
    Accept: */*\r\n
    \r\n
    [Full request URI: http://facebook.com/]
    [HTTP request 1/1]
    [Response in frame: 9273]
```

Looking at the request in Wireshark, this curl request to facebook.com is being sent to a local IP address.

DNS spoofing can be combined with other malicious tools to trick an end user. The Social-Engineer Toolkit (SET) ships with Kali Linux and provides several tools for building believable attack vectors in short pages. It includes prebuilt credential harvesting functionality. A login page can be generated using either a prebuilt template can be used, or a page can be harvested from a link. Here is an example Google page generated from a template:



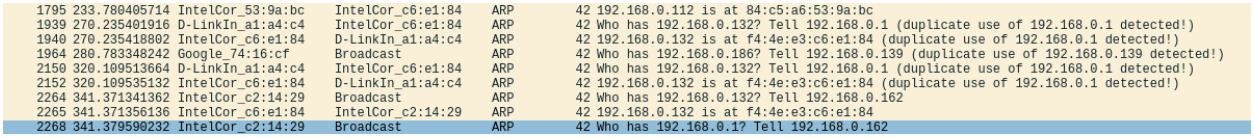
Any credentials entered into this page are captured by the SET and printed out in plaintext before the user is then rerdirected to google.



Combined with DNS spoofing, a credential harvesting page can catch an inattentive user and they could surrender their credentials.

Detection

Detecting ARP spoofing is not particularly difficult. It is relatively simple to keep track of the IP address to MAC address mapping of a local network. Given that ARP messages are broadcast over a local network, any machine should be able to detect the change. When capturing ARP traffic, changes in IP to MAC address mapping can be detected by Wireshark.



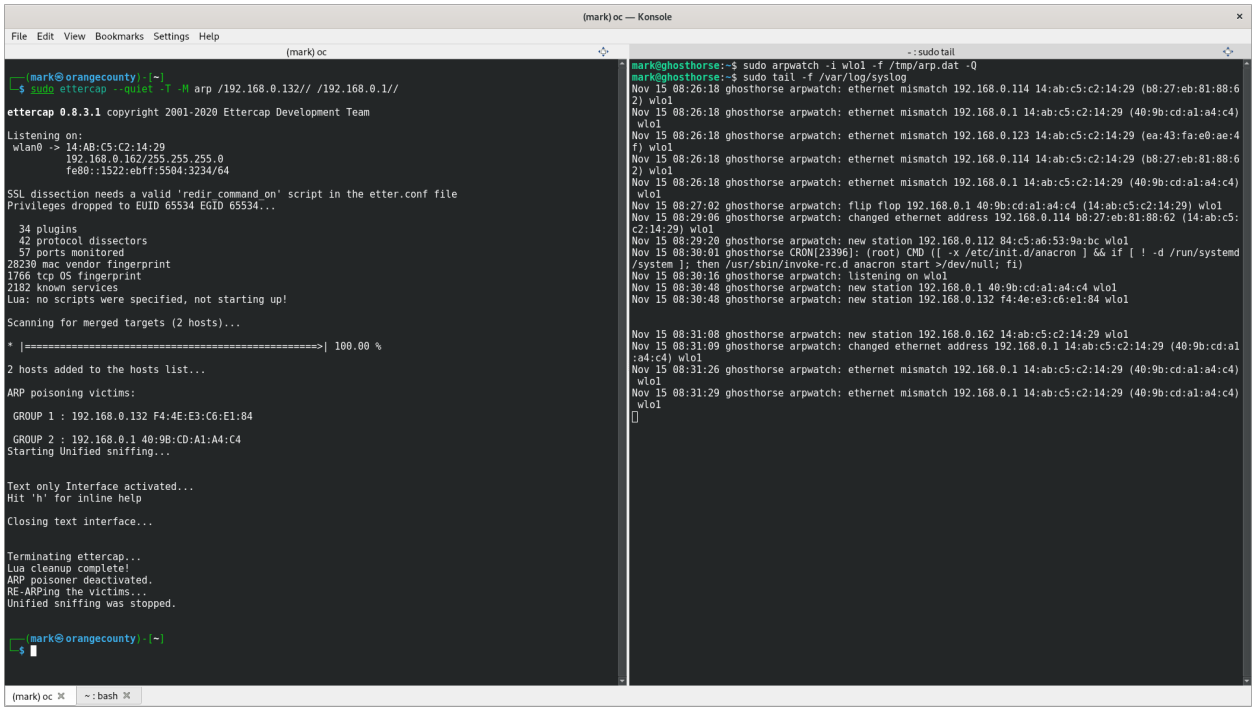
However, there are legitimate use cases for the MAC address assigned to an IP address to change. Machines could be swapped out, or they could have their network interface cards changed. In certain settings, where many devices are connecting and disconnecting from a network frequently, the mapping could change often. The challenge then becomes how to limit the number of false positives. If there are too many alerts generated for legitimate reasons, the alerts become useless.

One of the other challenges is that many Cybersecurity tools focus on traffic at the Network and Application layers and ARP traffic is at the link-layer. Tools like netfilter and Snort are not well suited to filtering or inspecting ARP traffic.

arpwatch

Arpwatch is a tool that uses pcap to listen for ARP packets on an ethernet interface and keeps track of MAC address to IP address pairings. This tool reports when it sees a new address pair, a new MAC address on the network, when a MAC address changes, and when a MAC address

reverts to a previously used address. Reports can be configured to send emails to an administrator. The reports, and additional events, are logged using syslog.



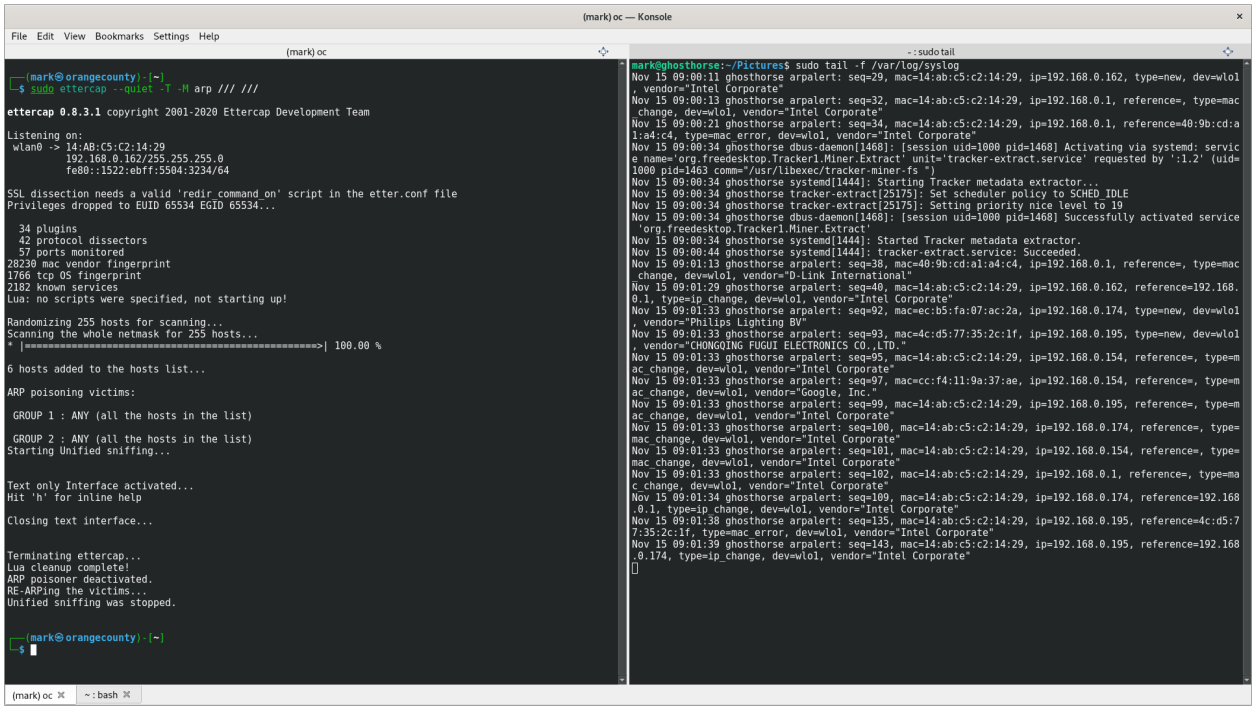
Arpwatch is a useful tool in largely static networks. In a network that has many new or changing devices connecting, a huge volume of alerts would be generated. In a largely static network, reports could easily be viewed and potentially treated as an event for an administrator to look into. Additionally, arpwatch can be used to alert administrators when a new device connects to a network.

arpalert

Similar to arpwatch, arpalert is an ARP traffic monitoring tool. Arpalert allows for more detailed configuration and provides more functionality than arpwatch. Some of the additional features that arpalert provides are:

- MAC Address filtering, either allow or block listing.
- Custom scripts can be run when alerts are generated.
- Devices vendors are identified based on their Organizationally Unique Identifier (OUI)
- ARP Flood alerting. When the number of requests exceeds a predetermined limit, a flood event is logged and events are ignored for an interval to avoid sending too many messages.

Even with more functionality and control than arpwatch, arpalert still functions best in a largely static environment. The ability to alert on message flooding does make it more useful for identifying spikes in ARP traffic. Although there will be legitimate spikes in ARP traffic, they should not occur often and should be easily explainable.



Snort arpspoof

The IDS Snort does have a preprocessor called arpspoof that should detect ARP spoofing attempts. In the configuration file, an administrator can define IP address and MAC address combinations that should not change.

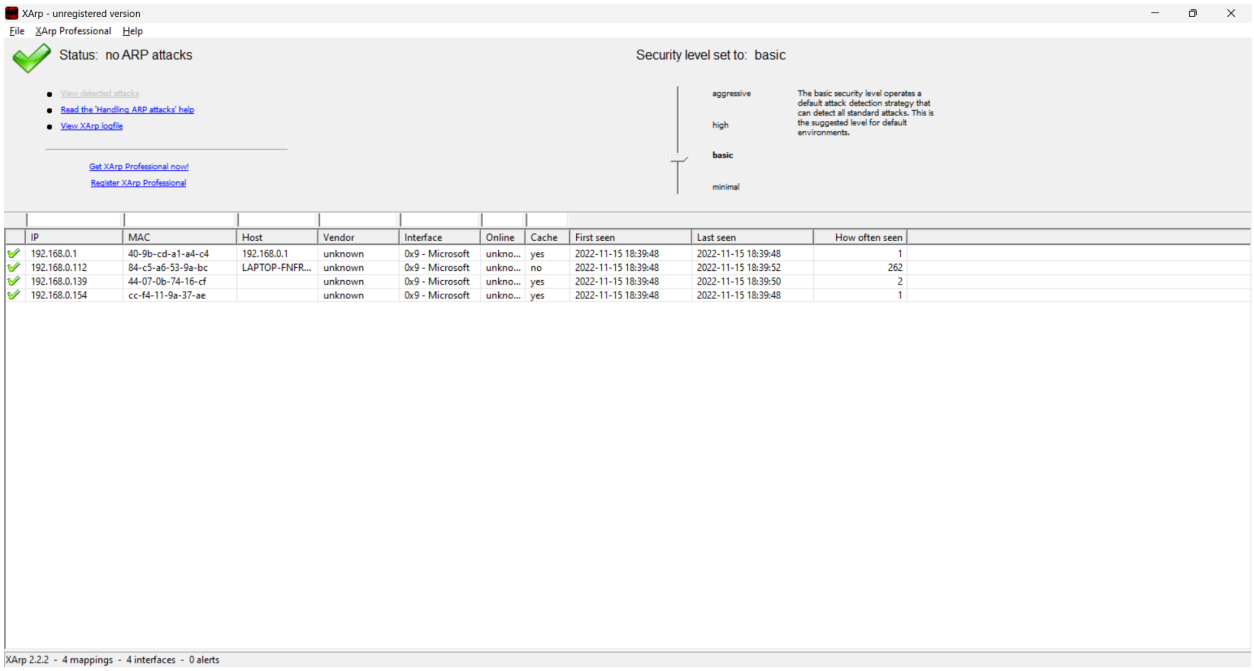
```
preprocessor arpspoof
preprocessor arpspoof_detect_host: 192.168.0.1   00:aa:bb:cc:dd:ee
```

If an ARP request indicates that the MAC address is different than the one in the configuration file, this could indicate that an ARP spoofing attack is occurring. A snort alert is generated.

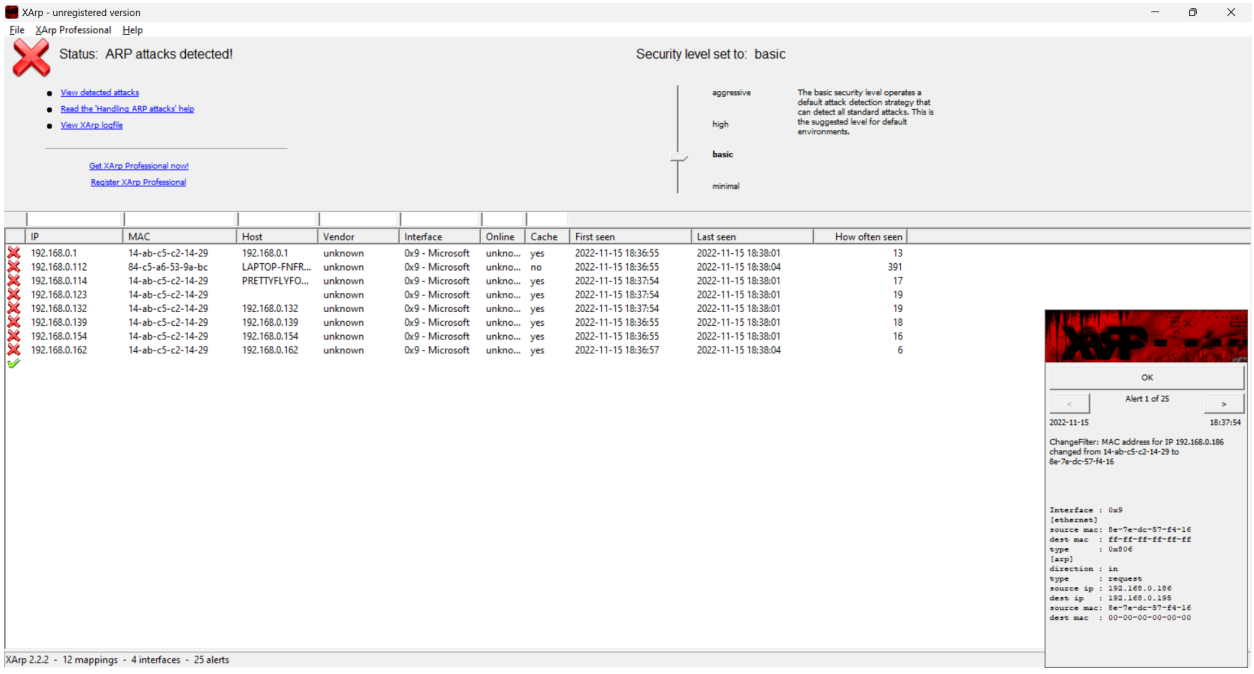
This preprocessor is marked as experimental, and the student was unable to get the preprocessor working as part of this assignment.

XArp

XArp is a Windows application that detects ARP poisoning. It passively gathers information about machines on the network by listening to ARP broadcasts.



Details on how XArp works are sparse. Through testing, it appears to operate in a similar fashion to arpalert and arpwatch. When it identifies changes in the IP address to MAC address pairing, it raises an alert.



Conclusion

Detecting and preventing an ARP spoofing attack can be challenging. There is no authentication mechanism built into the protocol itself. Many of the existing tools and technologies for monitoring and controlling network traffic operate at a higher level than Ethernet traffic. Most of the free tools that exist for detecting ARP spoofing attacks are most effective on static networks that do not see devices connecting or changing often. In situations where new devices are added often, or when IP addresses get reused, these tools are likely to trigger many false positives. Given the control an attacker can gain over a network once they have access to a single host, and how easy it is for these attacks to be launched, further techniques and tools for preventing and detecting these attacks should be researched and developed.

Appendix A - Submission File Structure

Directory	File	Description
./report/	Report.pdf	This report.
./videos/	ettercap-dns.ogv	A demo video of DNS spoofing using Ettercap
	ettercap-cred-harvesting.ogv	A demo video showing credential harvesting using Ettercap and Wireshark
./data	ettercap-dns-attacker-filtered.pcap	A packet capture taken from the victim of a DNS spoofing attack
	ettercap-dns-victim-filtered.pcap	A packet capture taken from the attacker of a DNS spoofing attack
	ettercap-mitm-attacker-filtered.pcap	A packet capture taken from the MitM udring credential harvesting
	ettercap-mitm-telnet-client-filtered.pcap	A packet capture taken from a telnet client durling credential harvesting
	ettercap-mitm-telnet-server-filtered.pcap	A packet capture taken from a telnet server durling credential harvesting