Global Information Assurance Certification Paper

Address Resolution Protocol Spoofing and Man-in-the-Middle Attacks

Practical Assignment GSEC Version 1.2f (amended August 13, 2001)

Robert Wagner

Abstract

The classic Man-in-the-Middle attack relies upon convincing two hosts that the computer in the middle is the other host. This can be accomplished with a domain name spoof if the system is using DNS to identify the other host or address resolution protocol (ARP) spoofing on the LAN. This paper is designed to introduce and explain ARP spoofing. The term Man-in-the-Middle is used from a historical usage, this does not imply that only men can use these attacks. Perhaps Teenager-in-the-Middle or Monkey-in-the-Middle may be more accurate terms.

Throughout the Internet, IP is used for communications. Once an IP packet comes into a Local Area Network (LAN) it must be converted into a packet that the LAN can understand. An IP packet is encapsulated into an Ethernet frame for local handling. Ethernet is the most common type of network and other types of networks handle communications differently. Ethernet does not communicate via IP addresses, but uses the hardware address instead. Ethernet uses the Address Resolution Protocol (ARP) to resolve IP addresses into hardware addresses. This hardware address is also called the Media Access Controllers (MAC) addresses. Once the destination's MAC address is determined, the encapsulated IP packet can be transmitted to the host. Every network device must have an unique MAC address for the Ethernet LAN to function correctly.

Within Ethernet ARP there are four types of messages.

ARP request – This is a request for the destination hardware address. It is typically sent to all hosts.
ARP reply – In response to a request, this tells the host the hardware address of the destination host.
RARP request – Known as Reverse ARP request. This requests the IP address of a known MAC address.
RARP reply – The response for a RARP request. This gives the IP address from a requested hardware address.

All Ethernet hosts and switches keep a list of MAC and IP address in a table and will request this information from the network when a new IP address is requested or a table entry expires. On a network connected with a hub, sniffing packets to gather MAC addresses is easy. This is because all packets are broadcast to all hosts. Sniffing data is as easy as plugging into any open port. Systems connected with switches present a different problem. Traffic is no longer broadcast to every host. There are a few solutions to this problem, some are in the user's manuals while others are hacks. The first is to connect to an administrative port on the switch and set it to broadcast mode. The administrative port will now receive all traffic. Some switches allow one to choose the administrative port in a software setup, while others restrict it to one particular physical port. The second method is to send such a large number of ARP entries to the switch so it fails open and just sends all packets to all ports. This may vary depending upon the manufacturer. The final method is to run software that tells every host on the network to use an attacker's MAC/IP address in order to get to any other host. The software then creates a table to forward the traffic to the correct host after one has had an opportunity to sniff it.

Many of these techniques and attacks involve tricking a system into modifying its MAC table and sending the data to the attacker instead. This paper is going to look at several tools and methods for gathering ARP information and how to defend the network against it. The result of an attack on a system's MAC table can vary. The basic problem is data will not go to the correct host and this can result in a denial of service if the MAC address does not exist on the network. Another result may be session hijacking where the attacker acts as a Man-in-the-Middle to each host. Similar to this is where the attacker actually takes over the MAC and IP address of one of the hosts and now may act as a trusted host for stealing more data.

Arpoison **http://web.syr.edu/~sabuer/arpoison/**
Arpoison is a simple command line tool that creates a custom ARP Reply packet. This software is effective because many machines will gladly accept the ARP reply and update their MAC table. One notable exception is the Solaris operating system. Solaris will only accept an ARP Reply after the MAC table has reached a timeout value. The default value for requested ARP entries is twenty minutes, the value for unrequested entries is five minutes. In reviewing the source, we see the command executed called by:

```
arpoison -i <device> -d <dest IP> -s <src IP> -t <target MAC> -r <src MAC>
```

This allows an attacker to specify the source and destination IP and MAC addresses. In order to update all hosts on a LAN the packet could simply be sent to the broadcast MAC address of ff:ff:ff:ff:ff:ff.

The ***arp –a*** command is used to view the MAC table and ***arp –s*** can set static MAC entries. There may be slight changes to this command depending on the operating system. Attacks of this kind are typically difficult to detect. Some suspicion should be raised during a massive attack aimed at poisoning the ARP table on all hosts. Another identifier would be a denial of service attack on one host. A method for securing the system would be to manually maintain the MAC table and setup monitors and alerts for any ARP traffic. This would method would be cumbersome in a fluid environment and adding a new host would require a variety of changes to all hosts.

Part of the security relies upon knowing the correct MAC for all hosts. Many sniffers will track this level of information, but when trying to identify a MAC to a physical computer, many systems administrators are lost. Please note the fact that in order to inject this traffic into a LAN, one must have a physical connection to the LAN. Attacks from the outside will need to compromise a system first to get onto the LAN. Attacks from the inside are much easier because they typically have access to the LAN. Deploying Network and Host based Intrusion Detection Systems and keeping an eye on unusual activity will help.

Ettercap **http://ettercap.sourceforge.net/**
Ettercap is an enhanced sniffer for Unix based systems. The software claims to allow the user to collect data and/or passwords from a variety of protocols including SSH1 and HTTPS. The program allows the user to poison the ARP cache on systems and sniff switched LANs as well. Once this is accomplished, acting as a Man-in-the-Middle is an easy act. Because packets are being sent through the attacker's computer, injecting malicious commands into an existing session is automated through this software. Another feature is the ability to drop packets. This is particularly problematic with protocols such as UDP (like remote logging) where the session may not be expecting the packet and lost packets do not raise an alarm.

This software is limited to the number of different operating systems it supports. Sun Solaris and Microsoft Windows are currently under development. The program sets up a basic framework where additional modules and protocols it can attack can be added easily.

Here is an explanation of the SSH1 Man-in-the-Middle attack to show the ingenuity of the programmers. First one must poison the cache to have the traffic routed through the attacker's machine. Once the client requests an SSH1 session from the server, the attacker intercepts the key generated by the host and forwards its own key to the client. The client, thinking they are communicating with the secure server, now sends authentication and session information to the server. The attacker decrypts this information and encrypts it using the server's original key and passes it to the server. Once in the middle, the attacker has access to all traffic. It's easy to see that this software can be used anywhere on the LAN.

Security no longer can look at just protecting single hosts. They must know and understand how the entire LAN functions. It is only secure as its weakest machine. Insider attacks will be easier to commit as software such as this makes it easy to gather passwords and data from users and sessions. Prevention is similar to *ARPoison* prevention where MAC tables are setup statically. Another method for monitoring this would be to run a list of MAC addresses on a regular basis and look for duplicate addresses (unless one is running virtual IP addresses from a single machine).

Parasite **http://packetstormsecurity.org/sniffers/parasite-0.5.tar.gz**
Parasite is another sniffer with the ability to bypass basic security on most switches. While allowing the attacker to sniff data, its most powerful feature may be in performing denial of service attacks. This software is designed to wait until the switch sends out ARP Request packets to poison the cache. It will slowly poison the cache without a large flood of ARP packets that may setoff alarms. Even if removed, systems will take time to recover. Systems administrators must be aware of this possibility if they are battling a slow or non-responsive network.

Dsniff **http://www.monkey.org/~dugsong/dsniff/**
The final tool being shown is *dsniff*. Like the others, this program provides password sniffing and "monkey-in-the-middle" attacks for SSH and SSL. This program is touted as a vulnerability checker that can pull passwords from Telnet, FTP, SMTP, HTTP, POP, poppass, NNTP, IMAP, SNMP, LDAP, Rlogin, RIP, OSPF, PPTP, MS-CHAP, NFS, VRRP, YP/NIS, SOCKS, X11, CVS, IRC, AIM, ICQ, Napster, PostgreSQL, Meeting Maker, Citrix ICA, Symantec PcAnywhere, NAI Sniffer, Microsoft SMB, Oracle SQL*NET, Sysbase and Microsoft SQL. It can be utilized to identify if these protocols are occurring over the network and if passwords are being sent. Similar to *Ettercap*, this program can use ARP or DNS poisoning to trick a host into communicating through it. Both the HTTPS and SSH Man-in-the-Middle attacks are performed through DNS poisoning, thus allowing it to occur outside a single LAN (although the race condition against a local DNS server may be tricky).

This last tool is mentioned because it ties several ARP and DNS attack methods together showing the progress these tools have made in becoming easier to use and more flexible for the attacker. One can easily see that we are at the third generation of attack tools. Man-in-the-Middle attacks that were once thought of being too complex to realistically occur are now just a click away.

Securing Against ARP Attacks and Poisoning

It should be apparent at this time that knowing the MAC addresses of each machine should not be left up to the automatic resolution used in the protocol. Unless one controls the ARP process, poisoning and sniffing can occur. The manifestation of this problem could range from slightly slow network response to the complete disruption in network traffic. This may be found only through using another sniffer to see what is happening at the MAC level or looking at the ARP tables on both the client and server machines.

At this point, it is necessary to identify the MAC address of every computer and gateway on the LAN. Running an **ipconfig /all** on Microsoft Windows NT computer will show the MAC address of the Ethernet card. It will also show the MAC addresses of VPN virtual adapters. Sun Solaris and Linux use the **ifconfig –a** command, while FreeBSD uses the **dmesg** command. Microsoft Windows 9x machines use the **winipcfg** command.

Armed with the list of MAC addresses, the systems can now be secured. It is recommended a login script that automates the **arp –s** command to store all host IP and their MAC addresses. Replacing broken network cards will now require additional changes on every host. Systems such as printers may not have the ability to store a list of static MAC addresses. Most switches have a port security option that allows only one MAC address per port (thus reducing the sniffing). SSH has a S*trictHostKeyChecking* option. This option specifies whether or not SSH will automatically add new host keys to its known hosts file. With this set to YES, that attacking computer will not be allowed into the middle unless they are already a trusted host. Having a static HOST file listing DNS names will also reduce DNS based attacks. Consider using both for high security. It should be pointed out that some versions of Microsoft Windows would accept and use dynamic routes even if static routes have been set. This problem is noted in Sean Whalen's "An Introduction to ARP Spoofing". Test all systems to ensure this is a valid solution.

With the hosts now locked down, it will be important to monitor ARP and DNS traffic. If everything is hard coded into the machines, no ARP requests should be occurring and DNS traffic should be limited to external domains. Please be aware that most IDS systems monitor traffic at the IP level, not the Ethernet level. Here are some examples:

SNORT **http://www.snort.org**
Due to the high popularity of *Snort* I would like to mention it here. *Snort*'s User Manual states that *Snort* only supports TCP, UDP, ICMP, and IP; not ARP or routing protocols such as IGRP, OSPF. It is mentioned to warn people that rely on *Snort* that it does not support ARP and should be supplemented with another monitoring tool.

ARPWATCH **http://www.redhat.com/swr/i386/arpwatch-2.1a4-29.i386.html**
Arpwatch contains two utilities designed to monitor the IP/MAC table and e-mail any changes. This is a very simple and straightforward piece of software. Here are some samples of what will show up in the /var/log/messages file.

```
Sep 20 12:36:11 myhost arpwatch: new station 192.168.a.b 0:50:94:d7:ca:d5
Sep 20 12:35:07 myhost arpwatch: changed ethernet address 192.168.a.c
0:10:a4:bf:b1:c9 (0:0:86:45:32:fa)
```

The first line shows a new IP/MAC address combination. This will continue for every host on the LAN. The second line shows that the MAC address has changed for host 192.168.a.c. The new MAC address is 0:10:a4:bf:b1:c9. The previous address was 0:0:86:45:32:fa. This should cause the system administrator to pause and review some basic information about the host. If this is a dedicated server, then the address shouldn't change without switching the hardware. This host may also be using DHCP. If one host leaves the network and a separate host picks up its IP address, this change may be appropriate. Please note, by using the hardware address to identify the vendor, one may notice that the change MAC address changed from a Xircom to Gateway Communications (bought by Megahertz and then 3Com). This could also alert one to hardware that is outside of their LAN inventory.

When investing in an IDS system, take time to understand if it will monitor below the IP level. Ask the engineers if ARP traffic is monitored and how changes are noted. If not, one can always monitor the machine manually and fix them manually if needed.

Cisco IOS Commands

Like many switches, Cisco's IOS has a command called: **Set Port Security**. Enabling this feature will restrict the switch such that only one (default) MAC address is allowed per physical port. This command allows one to configure the action that will take place upon a hardware address change. By limiting the number of hardware addresses per port to one, a host cannot change their hardware address on the fly or try and map multiple MAC addresses to route traffic out one port. This will not analyze the MAC/IP table and take action during changes. It will not have any affect on DNS spoofing. An attacker could use this as a denial of service tool by forcing hardware address changes on a host.

Summary

As one can see, higher level protocols such as IP rely on a good base protocol foundation. Security measures such as switches, hard coded host tables, and coding IP addresses instead of DNS can be subverted through ARP attacks. Security mechanisms like HTTPS and SSH can also be tricked when someone is able to get into the LAN. A security analyst must review the entire LAN for any weakness, as a single compromised host can be a gateway for other attacks and intelligence gathering. If a system is already acting as a gateway then the attacker's job is much easier because traffic to passing through the system. These Man-in-the-Middle attacks bring to question the security of HTTPS and SSH where a single certificate is handed out to the supposed end user. Even more advanced asymmetric encryption algorithms may not be helpful unless we can truly verify someone is not operating in the middle and handing fake keys back to the user. The concept of knowing and trusting keys handed out by hosts and people becomes increasingly important.

Warning about using these utilities

It should be obvious that unless a system's MAC table is locked down, it is easy to mess it up.

Using these utilities on an unprotected system will cause problems. Do not run these on a production network. Recovery from an ARP poison will occur when timeout values are reached or the tables are manually cleared.

References and For Further Information

Hewes, Douglas "I Can See you Behind Layer 2… Overcoming the difficulties of Packet Capturing on a Switched Network" (14 September 2000)
http://www.sans.org/infosecFAQ/switchednet/layer2.htm

Watson, Keith and Noordergraaf, Alex "Solaris Operating Environment Network Settings for Security" (December 2000) http://www.sun.com/blueprints/1200/network-updt1.pdf

Fermilab – Data Communications and Networking Group "How to find your MAC address" (05 February 2001) http://www-dcn.fnal.gov/DCG-Docs/mac/

Mourani, Gerhard "Securing and Optimizing Linux: RedHat Edition – A Hands on Guide" (2000)
http://www.linuxdoc.org/LDP/solrhe/Securing-Optimizing-Linux-RH-Edition-v1.3/chap15sec121.html

Roesch, Martin "Snort Users Manual, Snort Release: 1.8.1" (10 August 2001)
http://www.snort.org/docs/SnortUsersManual.pdf

EcoPRO "Windows NT, PC Networking – Windows NT 4.0" (1999)
http://www.ecoproimaging.com/nt_networking.htm
ARP table information for Windows NT

Whalen, Sean "An Introduction to ARP Spoofing" (April, 2001)
http://packetstormsecurity.org/papers/protocols/intro_to_arp_spoofing.pdf

Fairhurst, Gorry "Address Resolution Protocol (arp)" (01 January 2001)
http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html

Ethernet Codes Master Page (26 October 1998) – Used to match MAC address to hardware vendor. http://www.cavebear.com/CaveBear/Ethernet/

Cisco "IOS Commands – Set Port Security"
http://www.cisco.com/univercd/cc/td/doc/product/lan/cat5000/rel_5_4/cmd_ref/set_po_r.htm#xtocid573819