An Introduction to ARP Spoofing

Sean Whalen
http://node99.org/projects/arpspoof

April, 2001

Revision 1.82

P U R P O S E

This paper deals with the subject of ARP spoofing. ARP spoofing is a method of exploiting the interaction of IP and Ethernet protocols. It is only applicable to Ethernet networks running IP.

The subject will be addressed such that anyone with basic networking experience can understand key points of the subject. Knowledge of the TCP/IP reference model is vital to full understanding, as is a familiarity with the operation of switched and non-switched networks. Some background will be presented in the "Introduction" section, but experienced readers may wish to skip to "Operation".

I N T R O D U C T I O N

A computer connected to an IP/Ethernet LAN has two addresses. One is the address of the network card, called the MAC address. The MAC, in theory, is a globally unique and unchangeable address which is stored on the network card itself. MAC addresses are necessary so that the Ethernet protocol can send data back and forth, independent of whatever application protocols are used on top of it. Ethernet builds "frames" of data, consisting of 1500 byte blocks. Each frame has an Ethernet header, containing the MAC address of the source and the destination computer.

The second address is the IP address. IP is a protocol used by applications, independent of whatever network technology operates underneath it. Each computer on a network must have a unique IP address to communicate. IP addresses are virtual and are assigned via software.

IP and Ethernet must work together. IP communicates by constructing "packets" which are similar to frames, but have a different structure. These packets cannot be delivered without the data link layer. In our case they are delivered by Ethernet, which splits the packets into frames, adds an Ethernet header for delivery, and sends them down the cable to the switch. The switch then decides which port to send the frame to, by comparing the destination address of the frame to an internal table which maps port numbers to MAC addresses.

When an Ethernet frame is constructed, it must be built from an IP packet. However, at the time of construction, Ethernet has no idea what the MAC address of the destination machine is, which it needs to create an Ethernet header. The only information it has available is the destination IP from the packet's header. There must be a way for the Ethernet protocol to find the MAC address of the destination machine, given a destination IP.

This is where ARP, the Address Resolution Protocol, comes in.

O P E R A T I O N

ARP operates by sending out “ARP request” packets. An ARP request asks the question, “Is your IP address x.x.x.x? If so, send your MAC back to me.” These packets are broadcast to all computers on the LAN, even on a switched network. Each computer examines the ARP request, checks if it is currently assigned the specified IP, and sends an ARP reply containing its MAC address.

To minimize the number of ARP requests being broadcast, operating systems keep a cache of ARP replies. When a computer receives an ARP reply, it will update its ARP cache with the new IP/MAC association. As ARP is a stateless protocol, most operating systems will update their cache if a reply is received, regardless of whether they have sent out an actual request.

ARP spoofing involves constructing forged ARP replies. By sending forged ARP replies, a target computer could be convinced to send frames destined for computer A to instead go to computer B. When done properly, computer A will have no idea that this redirection took place. The process of updating a target computer’s ARP cache with a forged entry is referred to as “poisoning”.

A T T A C K S

S N I F F I N G

Switches determine which frames go to which ports by comparing the destination MAC on an frame against a table. This table contains a list of ports and the attached MAC address. The table is built when the switch is powered on, by examining the source MAC from the first frame transmitted on each port.

Network cards can enter a state called “promiscuous mode” where they are allowed to examine frames that are destined for MAC addresses other than their own. On switched networks this is not a concern, because the switch routes frames based on the table described above. This prevents sniffing of other people’s frames. However, using ARP spoofing, there are several ways that sniffing can be performed on a switched network.

A “man-in-the-middle” attack is one of these. When a MiM is performed, a malicious user inserts his computer between the communications path of two target computers. Sniffing can then be performed. The malicious computer will forward frames between the two target computers so communications are not interrupted. The attack is performed as follows (where X is the attacking computer, and T1 and T2 are targets):

- X poisons the ARP cache of T1 and T2.
- T1 associates T2's IP with X's MAC.
- T2 associates T1's IP with X's MAC.
- All of T1 and T2's IP traffic will then go to X first, instead of directly to each other.

This is extremely potent when we consider that not only can computers be poisoned, but routers/gateways as well. All Internet traffic for a host could be intercepted with this method by performing a MiM on a target computer and the LAN's router.

Another method of sniffing on a switched network is MAC flooding. By sending spoofed ARP replies to a switch at an extremely rapid rate, the switch's port/MAC table will overflow. Results vary by brand, but some switches will revert to broadcast mode at this point. Sniffing can then be performed.

B R O A D C A S T I N G

Frames can be broadcast to the entire network by setting the destination address to FF:FF:FF:FF:FF:FF, also known as the broadcast MAC. By sweeping a network with spoofed ARP replies which set the MAC of the network gateway to the broadcast address, all external-bound data will be broadcast, enabling sniffing.

If a host were to listen for ARP requests and generate a reply containing the broadcast address, potentially crippling amounts of data could be broadcast on large networks.

D O S

Updating ARP caches with non-existent MAC addresses will cause frames to be dropped. These could be sent out in a sweeping fashion to all clients on the network in order to cause a Denial of Service attack. This is also a side effect of post-MiM attacks, since targeted computers will continue to send frames to the attacker's MAC address even after they remove themselves from the communication path. To perform a clean MiM attack, the target computers would have to have the original ARP entries restored by the attacking computer.

H I J A C K I N G

Connection hijacking allows an attacker to take control of a connection between two computers, using methods similar to the MiM attack. This transfer of control can result in any type of session being transferred. For example, an attacker could take control of a telnet session after a target computer has logged in to a remote computer as administrator.

C L O N I N G

MAC addresses were intended to be globally unique identifiers for each network interface produced. They were to be burned into the ROM of each interface, and not be changed. Today, however, MAC addresses are easily changed. Linux users can even change their MAC without spoofing software, using a single parameter to "ifconfig", the interface configuration program for the OS.

An attacker could DoS a target computer, then assign themselves the IP and MAC of the target computer, receiving all frames intended for the target.

T O O L S

A R P O I S O N

http://web.syr.edu/~sabuer/arpoison/

ARPoison is a command-line tool for UNIX which creates spoofed ARP replies. Users can specify the source and destination IP/MAC addresses.

E T T E R C A P

http://ettercap.sourceforge.net/

Ettercap is a powerful UNIX program employing a text-mode GUI, easy enough to be used by "script kiddies". All operations are automated, and the target computers are chosen from a scrollable list of hosts detected on the LAN.

Ettercap can perform four methods of sniffing: IP, MAC, ARP, and Public ARP. It also automates the following procedures:

- Injecting characters into connections
- Sniffing encrypted SSH sessions
- Password collection
- OS fingerprinting
- Connection killing

P A R A S I T E
http://www.thehackerschoice.com/releases.php

Parasite is a daemon which watches a LAN for ARP requests, and automatically sends spoofed ARP replies. This places the attacking computer as the MiM for any computer that broadcasts and ARP request. Eventually, this results in a LAN-wide MiM attack and all data on the switch can be sniffed.

Parasite does not do a proper clean up when stopped. This results in a DoS of all poisoned computers because their ARP caches are pointing to a MAC address that is no longer forwarding their frames. Poisoned ARP entries must expire before normal operation can resume.

D E F E N S E S

There is no universal defense against ARP spoofing. In fact, the only possible defense is the use of static (non-changing) ARP entries. Since static entries cannot be updated, spoofed ARP replies are ignored. To prevent spoofing, the ARP tables would have to have a static entry for each machine on the network. The overhead in deploying these tables, as well as keeping them up to date, is not practical for most LANs. Also of note is the behavior of static routes under Windows. Tests found that Windows still accepts spoofed ARP replies and updates the static entry with the forged MAC, sabotaging the purpose of static routes.

MAC cloning can be prevented by a feature found on high-end switches called Port Security (also known as Port Binding or MAC Binding). Port Security prevents changes to the MAC tables of a switch, unless manually performed by a network admin. It is not suitable for large networks, or networks using DHCP. Port Security does not prevent ARP spoofing.

Aside from these two methods, the only remaining defense is detection. Arpwatch is a free UNIX program which listens for ARP replies on a network. It will build a table of IP/MAC associations and store them in a file. When the MAC address associated with an IP changes (referred to as a flip-flop), an email is sent to an administrator.

Tests showed that running Parasite on a network caused a flood of flip-flops, leaving the MAC of the attacker present in Arpwatch's emails. Ettercap caused several flip flops, but would be difficult to detect on a DHCP-enabled network where flip flops occur at regular intervals.

MAC cloning can be detected by using RARP (Reverse ARP). RARP requests the IP address of a known MAC address. Sending a RARP request for all MAC addresses on a network could determine if any computer is performing cloning, if multiple replies are received for a single MAC address.

If a MAC flood is performed and the switch reverts to broadcast mode, a computer will have to enter promiscuous mode to examine the broadcast frames. Many methods exist for detecting machines in promiscuous mode. These can be found in the Sniffing FAQ, at http://www.robertgraham.com/pubs/sniffing-faq.html. Note that you can perform ARP spoofing without being in promiscuous mode since redirected frames will be routed to your MAC.

It is important to remember that Operating systems have their own TCP/IP stacks, and Ethernet cards have their own drivers, each with their own quirks. Even different versions of the same operating system have variations in behavior. Solaris is unique in its treatment of ARP replies. Solaris only accepts ARP updates after a timeout period. To poison the cache of a Solaris box, an attacker would have to DoS the second target machine in order to avoid a race condition after the timeout period. This DoS may be detected if the network has an Intrusion Detection System in place.

C L O S I N G

ARP spoofing is one of several vulnerabilities which exist in modern networking protocols, which allow a knowledgeable individual free reign over a network. IP spoofing, TCP sequence prediction, and ICMP redirects are just a few examples of other current weaknesses in these protocols. It is unlikely that these problems will be addressed until they are abused on a wide enough scale to force a change in the status quo. The problem is poised to grow as broadband Metropolitan Area Networks are implemented using Ethernet as the protocol of choice.

Information in this paper was heavily influenced by the Ettercap and Parasite projects. Proof of concept tests were performed with the tools mentioned in this paper, against Linux, Windows NT, and Windows 2000 machines.