A low-cost embedded IDS to monitor and prevent Man-in-the-Middle attacks on wired LAN environments

Jorge Belenguer, Carlos T. Calafate*
Department of Computer Engineering
Polytechnic University of Valencia
E-mail: jorbefer@fiv.upv.es, calafate@disca.upv.es

Abstract

A Man-in-the-Middle (MitM) attack is, in the scope of a LAN, a technique where an attacker is able to redirect all traffic between two hosts of that same LAN for packet sniffing or data manipulation, without the end hosts being aware of it. Usually these attacks exploit security flaws in the implementation of the ARP protocol at hosts.

Up to now, detecting such attacks required setting up a machine with special-purpose software for this task. As an additional problem, few Intrusion Detection Systems (IDS) are able to prevent MitM attacks. In this work we present a low-cost embedded IDS which, when plugged into a switch or hub, is able to detect and/or prevent MitM attacks automatically and efficiently. Since our system is limited to a micro-controller and a network interface, it can be produced at a very low cost, which is attractive for large scale production and deployment.

1 Introduction

The detection and prevention of attacks, as well as repairing the damage inflicted, is an essential requirement for all organizations with security constraints. To solve such problems there are worldwide efforts seeking to alleviate the flaws of existing technologies, as well as to propose new technologies and systems which are more robust and secure.

Security issues in wired LAN environments are frequently under-estimated since wired LANs, when compared to their wireless counterparts, are considered to be inherently secure. However, many years after the Internet protocol stack was proposed, several security flaws are still present which put LAN users at risk, especially from inside attackers. An example of such flaws is related to the Address Resolution Protocol (ARP) [2], where operating systems rarely filter-out fake messages.

ARP-based attacks may be used to achieve Denial of Service (DoS) and packet sniffing/manipulation. DoS attacks usually cause a host to lose Internet connectivity. This can be achieved either through ARP cache poisoning or ARP cache overflow. In the former the physical address of the gateway router is changed to a non-existing address, causing all messages to be lost. In the latter the ARP cache is filled-up with spurious data, causing the system to turn-off the network interface.

Concerning sniffing attacks, these are more worrying since they allow an eavesdropper to gain access to all the packets sent and received by a host, which may include sensitive data such as passwords, credit card and bank account numbers, etc. Since such activity relies on passive monitoring, the end-to-end connection suffers no changes and so the host being attacked is often unable to detect it. In legacy LANs, the use of hubs made packet sniffing a trivial task since it could be achieved by setting the network interface card to promiscuous mode and running monitoring software. Wireless LANs suffer from the same problem, though new standards such as IEEE 802.11i [4] already offer the possibility of having per-host encryption keys. Current switch-based LANs require more elaborate attacks, such a the Man-in-the-Middle (MitM) attack, though there are free cracking tools that allow inexpert users to mount such attacks simply and quickly.

Hardware-based IDSs have been proposed as an efficient solution to protect hosts against network-based attacks. Examples of such works are [7, 9], which use FPGAs to achieve high-performance protection.

In this paper we propose a low-cost embedded system to achieve automatic detection and quick restoration of links suffering MitM attacks in wired LAN environments. The small size and the Plug & Play nature of the embedded system allows its massive deployment in Cyber Cafes and other low budget businesses, representing a significant improvement in terms of user privacy.

*This work was partially supported by the Ministerio de Educación y Ciencia, Spain, under Grant TIN2005-07705-C02-01.

0-7695-2989-5/07 $25.00 © 2007 IEEE
DOI 10.1109/SECURWARE.2007.8

The paper is organized as follows: in the next section we refer to some of the currently available tools that allow launching Man-in-the-Middle (MitM) attacks. In section 3 we detail the MitM type of attack we propose to solve. Section 4 presents an overview of the proposed embedded system, including two system prototypes: one offering reactive security and another one offering proactive security. An analysis of the performance and limitations of the proposed system is made in section 5. Finally, in section 6, we present our conclusions.

2 Software tools to perform and detect MitM attacks

There are several software applications to perform ARP cache poisoning. One of them is Arpoison [8], a command line utility that allows generating and sending ARP packets with user-defined values. Through this utility an attacker can launch DoS attacks or MitM attacks (if combined with packet forwarding).

Cain & Abel [5] is another tool targeting ARP cache poisoning. Its main purpose is poisoning all the existing ARP caches to monitor the activity of the entire network. It automatically activates packet forwarding on the eavesdropping station so that all packets arrive to the destination. Moreover, this tool scans all incoming traffic to obtain passwords and other confidential information, relying on brute force to extract them when necessary.

SwitchSniffer [3] is another tool which also combines ARP cache poisoning and packet forwarding to monitor traffic. Its simple user interface allows selecting one or more target links to perform a MitM attack. This utility also reacts to ARP repair packets, immediately restoring the attack.

In terms of detection tools, Snort [1] is an example of a widely used Intrusion Detection System (IDS) that is used to protect LAN environments from attacks to security and privacy. Typically, IDSs are able to detect several types of attacks, including Denial-of-Service (DoS) attacks, IP spoofing, etc. However, their effectiveness against MitM attacks is known to be very limited.

3 ARP-based MitM attack in LANs

In a Man-in-the-Middle attack (MitM) the attacker is able to read, insert and modify at will, messages between two parties without either party knowing that the link between them has been compromised.

In switched LAN environments such attack is usually accomplished through ARP exploits. This usually involves creating a new fake entry or updating existing entries in the ARP tables of the target hosts. Figure 1 is an example that helps to understand the MitM attack through ARP cache poisoning.

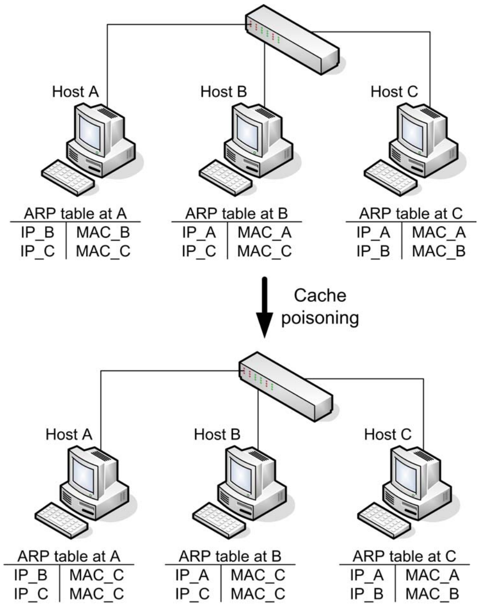

Figure 1. ARP cache mappings before (top) and after (bottom) cache poisoning.

Initially hosts A, B and C have the correct IP to MAC address mappings. If host C wishes to sniff on-going communication between hosts A and B it must poison the caches of both of them, as shown in figure 1. To achieve this it generates fake ARP messages to both hosts; such ARP messages are either ARP Request or ARP Reply Messages. ARP Request messages have the advantage that they are able to both update and create new ARP entries on all operating systems, while ARP Reply messages are typically limited to updating preexisting ARP entries.

For our example scenario, the messages generated by host C are those shown in figure 2.

Notice how, through an ill-defined ARP header, host C causes hosts A and B to send packets to itself every time they wish to send a packet to one another. If host C enables packet forwarding, all packets reach their destinations, allowing host C to sniff all the messages interchanged in a straightforward manner. Message manipulation is also possible with little additional effort.

4 Overview of the embedded IDS prototype

In this paper we propose using a low-cost embedded IDS to detect and prevent ARP-based MitM attacks in switched LAN environments. We have designed a prototype based on a 18F4620 micro-controller connected to a 10BaseT Ethernet interface that is equipped with a Realtek RTL8019AS

Ethernet header	
Source MAC address	MAC(C)
Dest. MAC address	MAC(A)
Ethernet Type	ARP
ARP message	
Message Type	ARP Request
Source MAC	MAC(C)
Source IP	IP(B)
Destination MAC	-
Destination IP	IP(A)

a) Host A cache poisoning

Ethernet header	
Source MAC address	MAC(C)
Dest. MAC address	MAC(B)
Ethernet Type	ARP
ARP message	
Message Type	ARP Request
Source MAC	MAC(C)
Source IP	IP(A)
Destination MAC	-
Destination IP	IP(B)

b) Host B cache poisoning

Figure 2. Fake ARP Request packets generated.

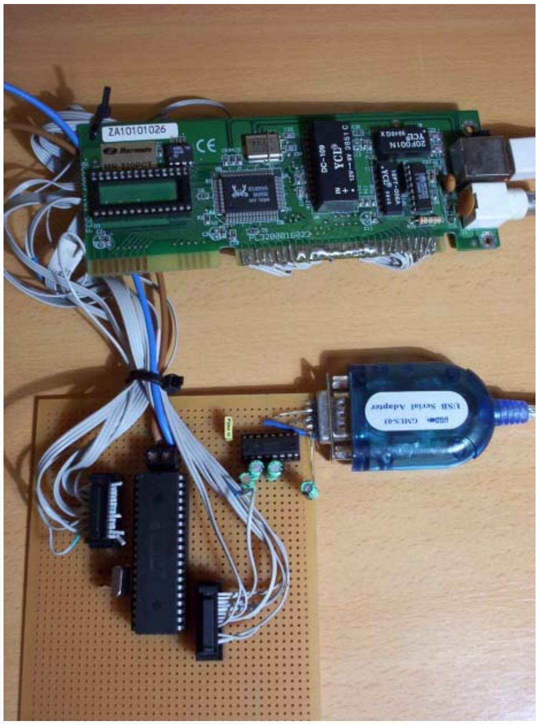

Figure 3. Picture of the prototype developed.

controller, usually known as NE2000. In figure 3 we show a picture of our prototype.

An RS-232 serial port is also available, which is used to send warnings to administrators; it also allows us to check the correct operation of the prototype since all activity is logged through this port.

Based on the same hardware prototype, we have developed two firmware variants - reactive and proactive - that offer different functionality; both aim at detecting and repairing ARP-based MitM attacks. The two firmware versions create and maintain ARP cache tables that store MAC to IP mapping information obtained by listening to ARP packets. This table will be used to detect fake ARP packets and also to repair the ARP cache of victims.

Each registry in the ARP table has a lifetime associated. Stall entries are refreshed through a *ping* or an *arping*, which refer to the generation of an ICMP or a directed ARP Request packet. If no reply is received within a reasonable time the entry being refreshed is considered to be down, and is removed from the ARP cache table.

We now proceed to detail the functionality of the two security modes developed.

4.1 Reactive security mode

Upgrading LANs from Hub-based to Switch-based networks hardens the packet sniffing process. On the other hand, Network IDSs also have difficulties in promiscuously monitoring their networks. To overcome this problem most switches now include the option of replicating the data from all ports or VLANs onto a single port. Depending on the manufacturer, such function is known as either Port Mirroring, Monitoring Port, Spanning Port, SPAN (Switch Port ANalyzer) or Link Mode port.

Activating this function is quite useful since an IDS may scan all the packets flowing through the network. However, switches usually disallow bidirectional traffic on the port, which means that they can receive but not send (unless they are equipped with a second Ethernet interface).

In the scope of our prototype, the reactive security mode was designed to be attached to such special ports of switches. If an on-going MitM attack is detected it may proceed to warn the system administrator through the serial port. If bidirectional communication on the Ethernet port is allowed it can proceed to repair the poisoned ARP caches.

This reactive security mode may also be used in the scope of hub-based LANs not to avoid packet sniffing, but to avoid Denial of Service attacks or attacks to message integrity. In the case of switch-based LANs, an embedded IDS device is required per switch for full network monitor-

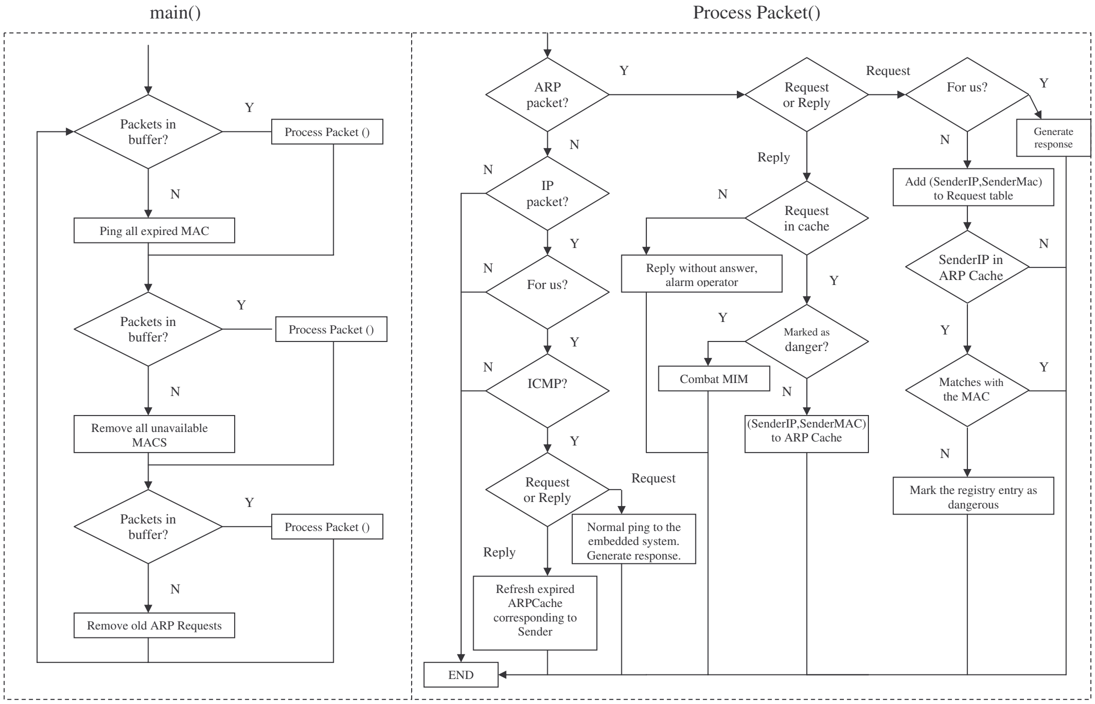

Figure 4. Flowchart for the reactive version of the firmware.

ing; if only hubs are used, a single embedded IDS device is enough.

To support reactive security we developed a piece of firmware for the micro-controller. Figure 4 shows the flowchart describing the code we developed.

The main loop is responsible for checking if there are packets in the input buffer of the Ethernet interface. If any is available, the *ProcessPacket* function is called. Otherwise, the system proceeds to ping all known hosts to check if they remain active. This is done by maintaining a list of ARP cache entries along with a lifetime. An ICMP *Echo Request* message is sent to all the entries in that list whose lifetime has expired; the default lifetime used by our prototype is of 10 seconds, though it can be adjusted. All the hosts in the LAN that do not answer to the test ICMP message are deleted from the ARP cache list. Finally, the system proceeds to delete pending ARP Requests states if the respective ARP Reply messages were not received.

In the scope of the *ProcessPacket* function, an additional table is maintained to store all ARP requests. This enables us to detect those ARP replies which are not related to an ARP request; there are high chances that these are being used to launch MitM attacks. Task *CombatMitM* is especially relevant since it is in charge of repairing poisoned ARP cache entries of victims. Taking our example, if we want to repair the cache on host B by telling it the correct IP to MAC mapping for host A, our embedded IDS would generate an ARP Reply message such as the one shown in figure 5.

Ethernet header	
Source MAC address	MAC(IDS)
Dest. MAC address	MAC(B)
Ethernet Type	ARP
ARP message	
Message Type	ARP Reply
Source MAC	MAC(B)
Source IP	IP(B)
Destination MAC	MAC(A)
Destination IP	IP(A)

Figure 5. Fake ARP Reply packet generated to correct on-going MitM attacks.

4.2 Proactive security mode

The proactive security mode applies to switched LANs where no monitoring port is available. So, our prototype is connected to a regular port, with the advantage that only a single device is required for a same VLAN, independently of the number of switches available. It operates by periodi-

Ethernet header	
Source MAC address	MAC(IDS)
Dest. MAC address	MAC(G)
Ethernet Type	ARP
ARP message	
Message Type	ARP Request
Source MAC	MAC(A)
Source IP	IP(A)
Destination MAC	-
Destination IP	IP(G)

Figure 6. Proactive ARP Request packets generated to correct possible MitM attacks.

cally refreshing the IP to MAC mappings of all active hosts to repair ARP-based MitM attacks. To avoid overloading the network with such messages, only the most significant ARP entries of hosts are refreshed. An example of a significant ARP entry is that of the Internet gateway router. To refresh the ARP entries related to the host-gateway link, a single corrective ARP Request Packet is generated.

Supposing that we want to refresh both the cache of a host A and gateway G, the IDS impersonates host A by sending a unicasted ARP Request to gateway B as shown in figure 6; such packet corrects the IP to MAC mapping for host A at the gateway. The gateway will then send an ARP Reply to host A, correcting G's ARP entry at that host. On-going MitM attacks on the link between A and G are avoided this way.

If the attacker uses a broadcasted ARP Request message instead of unicasting, the proactive security system detects it and acts immediately to correct the problem.

For the proactive security mode we also developed the corresponding piece of firmware. The flowchart for the code developed in shown in figure 7.

The main loop acts similarly to the one in the reactive version, except that no checking is performed in terms of ARP Requests that timed out since ARP Replies (unicasted) are no longer received. The *RefreshCaches* function is responsible for periodically sending ARP Requests to the gateway router, impersonating all the active hosts on the LAN to correct possible poisoned caches as explained earlier.

5 System performance and limitations

To evaluate the system we have tested both security modes in a LAN with an Internet gateway and two hosts. To test the first security mode a monitoring port was used, while a regular switch port was used for the second one.

On both tests the *SwitchSniffer* application was used; this

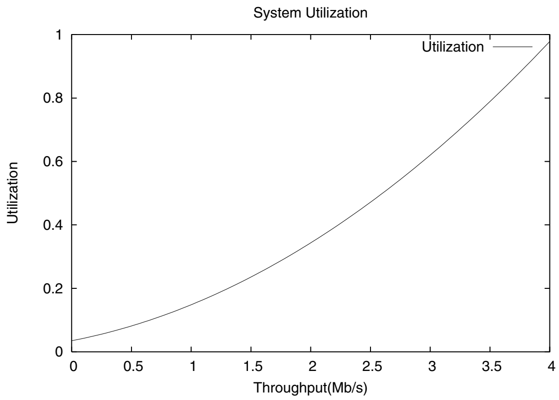

Figure 8. System utilization vs. throughput of incoming data.

application allows performing MitM attacks through user selection of a target connection between two hosts; the attack takes place through fake ARP Reply messages. ARP cache poisoning was confirmed by analysing the ARP tables of the target hosts.

When testing the reactive prototype we checked that attack detection was immediate, and that if the switch configuration allows for message generation by the IDS, poisoned route caches are also repaired immediately and with a reduced number of packets (one per poisoned cache entry).

Concerning the proactive prototype, it generates corrective messages at a rate: $R = N_{Act.Hosts} \times rr$, where $N_{Act.Hosts}$ is the number of active hosts in the network and rr is the refresh rate (in seconds) set by the IDS administrator. As expected, we found that no MitM attack takes effect for more than rr seconds.

In terms of limitations, we should point out that our prototype is limited in terms of input bandwidth and memory.

Since the micro-controller operates at a relatively low frequency, it may not process packets fast enough if the input rate is high. We performed measurements at different load levels in order to relate the throughput of incoming data to the resource utilization in the embedded system (results shown in figure 8).

As can be seen, our prototype reaches the saturation point for an input load of 4 Mbit/s. Such limitation is only relevant for the reactive version of the prototype, since the proactive one does not have to handle large amounts of data.

In terms of memory, our current prototype is limited to the scarce memory offered by the micro-controller used (4 Kbytes), which is able to offer service for up to 20 hosts.

Both limitations presented by this prototype can be overcome through a faster micro-controller, a faster interface and external memory connected through, e.g., an I2C Philips bus [6].

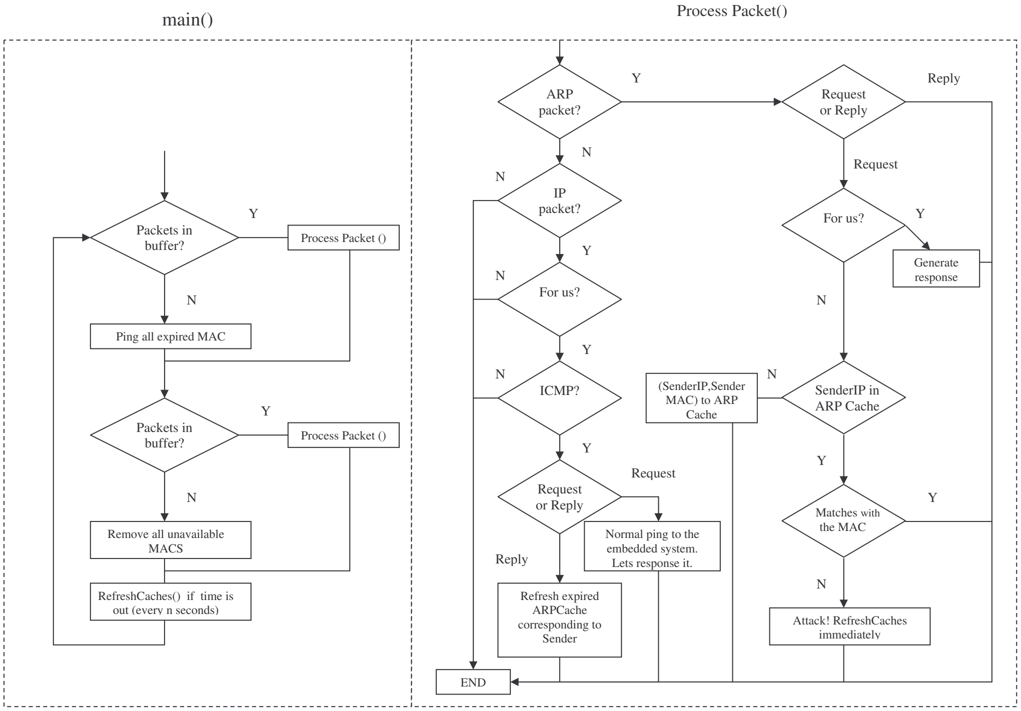

Figure 7. Flowchart for the proactive version of the firmware.

6 Conclusions

In this paper we emphasize the weaknesses of current IDSs in detecting ARP-based MitM attacks in switched LAN environments, evidencing the significant security risk incurred if no preventive action is taken.

To solve this problem we present a low-cost embedded IDS that is effective in detecting MitM attacks, as well as in repairing the poisoned ARP caches of hosts. We developed two prototype variants (reactive and proactive) that offer different functionality and have different requirements and performance.

An experimental testbed demonstrated the effectiveness of the proposed prototype, as well as its limitations. These were found to be strictly related to the available hardware resources.

We consider that massive deployment and use of low-cost devices, such as the one proposed in this paper, on switched LANs, will help at improving the security of companies and institutions with low budgets.

References

[1] Snort. http://www.snort.org/.

[2] David C. Plummer. An Ethernet Address Resolution Protocol. IETF RFC 826, November 1982.

[3] Gordon Ahn. SwitchSniffer. http://www.nextsecurity.net/.

[4] IEEE 802.11 WG. IEEE P802.11i/D10.0. Medium Access Control (MAC) Security Enhancements, Amendment 6 to IEEE Standard for Information Technology - Telecom. and Information exchange between systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2004.

[5] Massimiliano Montoro. Cain & Abel. http://www.oxid.it/cain.html.

[6] Philips Semiconductors. The I2C-Bus Specification, January 2000.

[7] Shaomeng Li, Jim Torresen, and Oddvar Soraasen. Exploiting Reconfigurable Hardware for Network Security. In *Proceedings of the 11th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, pages 202–293, Napa, California, USA, April 2003.

[8] Steve Buer. Arpoison. http://www.arpoison.net/.

[9] Tomoaki Sato and Masa-aki Fukase. Reconfigurable Hardware Implementation of Host-Based IDS. In *The 9th Asia-Pacific Conference on Communications*, pages 849–853, Penang, Malaysia, September 2003.