Man-in-middle-attack (MITM)

Für die Computer-Netze Übungen im WS 2002/2003 an der Uni Bern

Markus Anwander
Ehsan Maghsoudi
Daniel Schweri
Stefan Leuenberger

Contents

Glossary

Sniffer

Usually software programs who intercept streams of data on a network are referred to as sniffers. To achieve this, they switch the network interface card to a so called “promiscuous mode” enabling the software to listen in and display every packet on the segment and not only to the ones targeted at the computer running the sniffer software. It is very important that the topology of the network is suitable for this kind of wiretapping, an example of a very vulnerable topology would be a repeated ethernet network which is very common today.

Spoofing

Spoofing is the common name for the manipulation of data-packets in order to fake the originator address of a packet. An intercepted and manipulated packet has to be forwarded with the senders original address to disguise the interference and to pass unnoticed.
There are several known methods, eg. ARP spoofing which involves forging packet source hardware address (MAC address) to the address of the host you pretend to be. Many applications and tools in UNIX and/or Windooze systems rely on source IP address authentication and are thus likely to get compromised by such forms of spoofing.

MAC-address

MAC is an abbreviation for media access control. The term 'MAC-address' is being used as synonym for 'LAN-address'. The MAC-address has a length of 6 byte (this is true for the most common LAN type ethernet) and is commonly written in hexadecimal notation (e.g. 1A-23-F9-CD-06-9B). Every LAN-adapter has a different, unique MAC-address. To find out the MAC-address of another computer ARP is being used.

ARP / ARP table

ARP stands for address resolution protocol. With the entries in the ARP-table IP-addresses (or more common: addresses of the networking layer) are mapped to LAN-addresses (or more common: addresses of the link layer) and vice versa.
An ARP-table contains only information for the nodes of the LAN it is part of. The information is being stored for a limited period of time only (time-to-live (abbr: ttl)). If an ARP-module receives a request for an address that is not stored in its table, the requesting node has to send an ARP-packet to the ARP-module. This ARP-packet contains information on the senders address and the IP-address of the receiver. The ARP-module sends the ARP-packet to the other nodes in the LAN (broadcast). The receiving nodes check whether they are the node that is being searched. If so an
ARP-answer-packet with the mapping information (IP-address and LAN-address) is returned. The ARP-module inserts that information into its table.
To be able to find nodes outside the senders LAN, a packet with the target IP-address and the LAN-address of the router the sender is connected to is sent (the LAN-address of the router may be transmitted to the sender with ARP). The router sends directly or indirectly to the router to which the receiver is connected to. This router uses ARP to determine the
receivers LAN-address and delivers the packet accordingly.

POTS

The „plain old telephone system", which still is the most widespread form of distant communication across the world. It's really just an abbreviation for the telephone and it's connection you probably got at home.

Basics of a MITM attack

Procedure

The attacker must somehow get "between" the two hosts he wants to listen to. Mislead by hundreds of cheap secret agent movies and such, most of you will think about a pair of pliers and some wire to accomplish this, but this is not what we intend to show you how to do, we will show you how to do it in the James Bond way. In other words, the victim must never know we where there. So, how do we do it in a stealthy manner?
Unfortunately, there is no easy answer to this. The way to do this differs with every network technology and topology involved. We will concentrate on ethernet networks in our explanations because it is by far the most popular and widespread networking technology available today.
Abstracted, you would take the following steps to conduct a successful MITM attack:
1) Find out as much as you can about the LAN the computer is on. (Eg. Is it protected by a firewall, what is the topology used, what kind of technology is used, is there a wireless access point..)
2) Gain access to the LAN the computer is on either by hacking one of the hosts already on the network or by gaining physical access in order to you plug in your own computer.
3) Now either change the arp tables of the hosts you want to listen to or just put your network card in promiscuous mode – depends on the technology used.
4) Use tcpdump or a similar tool to record all the data you need
5) Wait for the correct data to be sent
6) Manipulate, record, delete, forward … the data
7) Get the hell out of there

Damage potential

The attacker is able to listen to, modify and delete data-packets at the attackers discretion. With the right tools, it is even possible to listen in to secured and crypted forms of data transfers. So, even if you see the little closed lock symbol in your browser, beware!
In conclusion the authors would like to point out how unsafe and dangerous it still is out on the 'net. With the techniques described here, every imaginable form of data manipulation is possible. Everyone

should use the credit card for online payments with great caution and only if there is really no other way to pay.

Prerequisits of a MITM attack

The attacker must obtain the following things in order to launch a successful MITM attack:
- Software to perform the attack
- Hardware to perform the attack
- Access to a computer on the targeted computers' LAN OR
- Access to the targeted computers' LAN
- Time
- Knowledge of the protocol he wished to manipulate

The attacker must be able to intercept the datastream, this can usually only be achieved if the attackers computer is in the same LAN as the victim. As an example, we look at two computers which reside in the same ethernet segment. In such a setup, it is astonishingly simple to listen in to the traffic, since every packet from and to the victim is also sent to the attacker. All the attacker has to do is to switch his network interface card to promiscuous mode and start up tcpdump. Since this technique solely relies on unswitched ethernet networks, we are not sure if this is indeed a MITM attack – you just aren't "in the middle" of the two computers you listen to.
In switched ethernet networks it is a little bit trickier to play MITM. One will have to actively dissect the network by manipulating the arp table entries on both nodes the villain wishes to wiretap (eg. a workstation and the router). Please have a look at the Flash animation should you have problems understanding what happens to the arp tables on the computers.
Another possibility would be to gain access to the router who forwards the packets. We will not go into this subject any further as this has nothing to do with MITM directly and is – for the most part – only possible by taking advantages of vulnerabilities in the routers' operating system.

Spoofing is another technique broadly used in conjunction with MITM attacks. Most of the time, one will spoof IP or DNS adresses.

Different types of MITM attacks

The Objective of the following paragraph is to understand the execution of a "Man-In-the-Middle" attack on different networks. We start off with MITM on ethernet, followed by an attack on GSM.

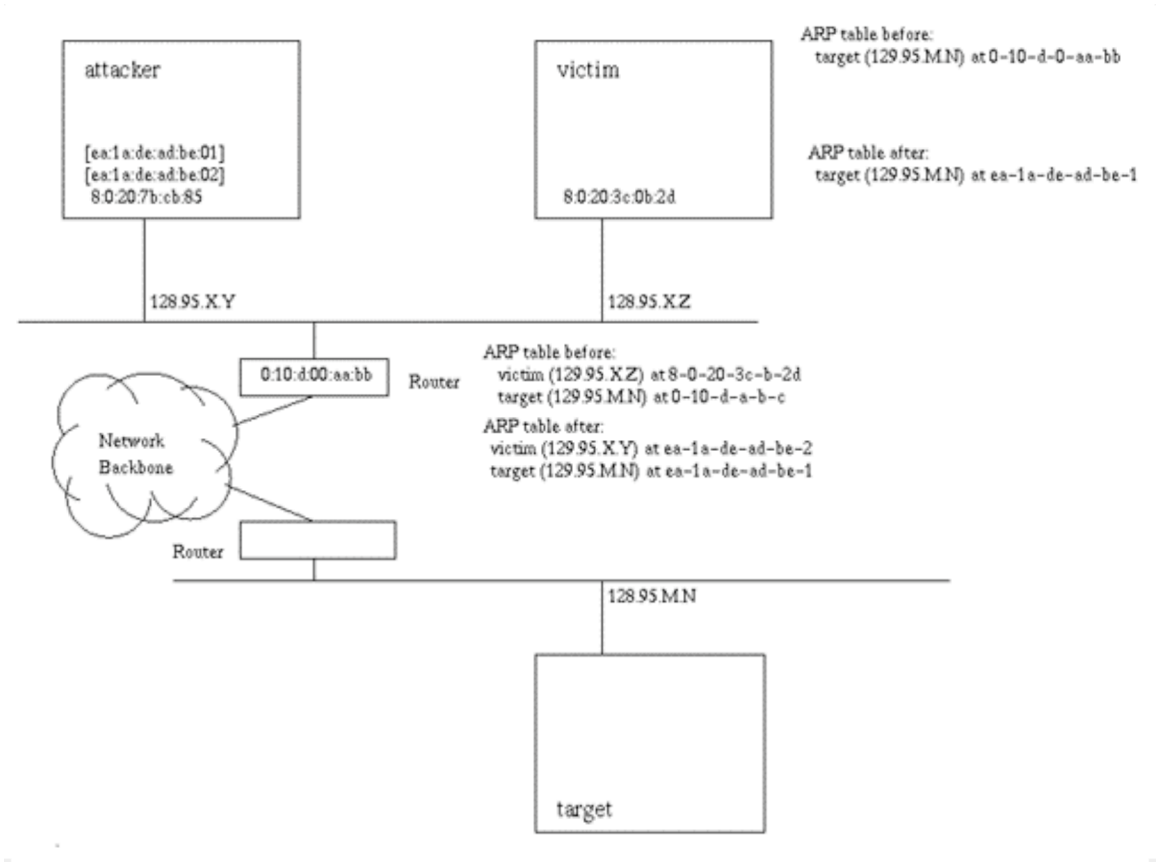
Ethernet / TCP

Overview:
The "Man In The Middle" or "TCP Hijacking" attack is a well known attack where an attacker sniffs packets from a network, modifies them and inserts them back into the network. There are few programs/source codes available for doing a TCP hijack. Juggernaut, T-Sight and Hunt are some these programs.
In this paper we shall explore Hunt for understanding how TCP Hijacking is deployed on an Ethernet segment.

Relevance:
TCP Hijacking is an exploit that targets the victims TCP based applications like Telnet, rlogin, ftp, mail application, web browser etc. An attacker can grab unenrcypted confidential information from a victim's network based TCP application. He can further tamper the Authenticity and Integrity of the data.

Simple Active Attack against TCP connections - An attack in which the attacker does not merely eavesdrop but takes action to change, delete, reroute, add, forge or divert data. Perhaps the best-known active attack is Man-In-the-Middle.

The Attack:



Attack Scenario involves three hosts: Attacker, Victim, and Target.

Attacker:
The system used by the attacker for the hijack.

Victim:
The system used by the victim for Telnet client connections to the target system.

Target:
The target system that the intruder wants to compromise. It is where the telnetd daemon is running.

A simple diagram of the network shows the Attacker and Victim hosts are on the same network (which can be Ethernet switched and the attack will still work), while the target system can be anywhere. Actually, either victim or target can be on the same network as attacker: it doesn't matter.
For the attack to succeed, the victim must use Telnet, rlogin, ftp, or any other non-encrypted TCP/IP utility. Use of SecurID card, or other

token-based two-factor authentication is useless as protection against hijacking, as the attacker can simply wait until after the user authenticates, then hijack the session.
The attack scenario can be as simple as:

Attacker: Spends some time determining the IP addresses of target and victim systems. Determining trust relationships can be easily done with utilities like SATAN, finger, systat, rwho or running who, ps, or last from previously stolen (or wide open "guest" style) accounts.

Attacker: Runs hunt as root on attacking host. Waits for hunt to indicate a session has been detected.

Attacker: Starts ARP relay daemon, prepares RST daemon entry for use later, sets option to enable host name resolution (for convenience).

Victim: Logs in to target using Telnet. Runs pine to read/compose email.

Attacker: Sees new connection; lists active connections to see if this one is potentially "interesting." If it is, attacker can either watch the session (packet sniffing) or hijack the session. In our example, the attacker decides to hijack.

Victim: Sees strange new prompt. Tries pressing RETURN and doesn't know what to think. Tries web browser and notices that it still works fine (not a network problem). Not sure what to think. Attacker: Finds this is a user session and decides to give it back (resynchronizes TCP/IP stream).

Victim: Sees prompt for keystroke, follows request, gets session back. Puzzled, decides to log in to root account to take a closer look.

Attacker: Turns on RST daemon to prevent new connections, waits to hijack root session.

Victim: Runs ssu to get SecurID protected root shell.

Attacker: Completes hijack after seeing root login.

Victim: Sees strange prompt. Tries pressing RETURN again. Same result as before. Tries web browser again. Same thing. Tries getting a new Telnet session. Fails. Tries ftp. Fails.

Attacker: Sets up backdoor, disables command history, resets session, turns off RST daemon.

Victim: Finally gets a new session. Original session is now gone. Assumes network outage or Windows TCP/IP stack corruption. Reboots system and everything is back to "normal".

Attacker: Waits for admin's sessions to all disappear (gone home for the night), then logs in using new backdoor. Installs rootkit (more backdoors, sniffer), cleans log files.

Brief Overview of the Daemons / threads that are used by the exploit:

Reset daemon:
Reset daemon is used to perform automatic resets of ongoing connections that hunt can see.
You can describe which connections should be terminated by giving src/dst host/mask and src/dst ports.

ARP daemon:
ARP daemon is used to do ARP spoofing of hosts.
You enter src and dst addresses and desired src MAC. The dst is then forced to think that src has srcMAC. You can use some fake MAC or better MAC of host that is currently down.

Sniff daemon:
Sniff daemon can log specified packets.
The sniff daemon can also search for a simple pattern (string) in the data stream. You can specify which connection you are interested in, where to search (src, dst, both), what do you want to search, how many bytes you want to log, from what direction (src, dst, both) and to what file should the daemon write.

MAC discovery daemon:
MAC discovery daemon is used to collect MAC addresses corresponding to the specified IP range.

ISMI-Catcher in GSM Networks

The following gives a brief introduction to the security functions available in GSM. The following functions exist:

- Access control by means of a personal smart card (called subscriber identity module, SIM) and PIN (personal identification number).
- Authentication of the users towards the network carrier and generation ofa session key in order to prevent abuse.
- Encryption of communication on the radio interface, i.e. between mobilestation and base station.
- Concealing the users` identity on the radio interface, i.e. a temporary valididentity code (TMSI) is used for the identification of a mobile user insteadof the IMSI.

The cryptographic algorithms which are used to generate and change the TMSI for encryption and authentication are kept secret (and, in part, remain secret). They are based on symmetric cryptography, i.e. both the network carrier and the mobile user (more precisely, the users` smart card) share a secret key Ki (unique for each user). All security parameters (encryption key, authentication data) are derived from the Ki. We now describe some known security problems and attacks on GSM.

The IMSI Catcher described discloses the identities of all users within a radio cell, while the attackers described in later sections attempt to make phone calls at the expense of other users. The last section describes which data is collected for billing.

IMSI Catcher:
The name of the IMSI Catcher refers to the abbreviation for the network-internal call number IMSI (international mobile subscriber identity).
By directly utilizing weaknesses in the authentication protocols of GSM, it is possible to determine the IMSIs of all users of a radio cell, i.e. the target of an attacker is to determine the identities of mobile users. The IMSI Catcher is even capable of signaling to the mobile phone that it should discontinue using encryption on the radio link, i.e. between the mobilephone and the base stations.
The type of attack which the IMSI Catcher launches is very simple and is referred to as a man-in-the-middle-attack. It behaves like a base station towards the mobile phone and in relation to the genuine base station of the network carrier it behaves like a mobile phone. Such an attack could have been easily prevented if in place of a one-sided authentication (the mobile phone only authenticates to the network carrier) mutual authentication (both users to network and network to users) had been used. Unfortunately, the GSM standard only defines a one-sided authentication.

The technical expenditure for the use of mutual authentication would have been only slightly higher.
The IMSI catcher sends its "location area identity" with a higher power than the genuine network.

SIM cloning:
In addition to the monitoring of users described in the preceding section, the question arises of whether it is possible to make unauthorized phone calls at someone else`s expense.
The challenge-response authentication of GSM Catching of IMSIs and suppression of encryption protection in mobile communications is intended to prevent this abuse.
Many network carriers connect their base stations by (non-public) radio links. The goal of this procedure is to communicate as much as possible via their own communication links in order to decrease costs. These radio links are usually not encrypted. If a user registers (location updating) at a base station, he is requested to authenticate himself. The network then creates authentication information (called authentication triple RAND, SRES, Kc) from the individual Ki.
The creation of this data is processed in the authentication center, located at the home network operator.
Since the challenge-response authentication protocol is processed in the mobile switching center which is being visited, the authentication triple has to be transmitted from the home network operator to the visited network operator.
If the authentication triple is transmitted over a (usually) unencrypted radio link, an attacker can eavesdrop and record the information. In the actual authentication request the attacker only needs to answer (respond) to the RAND with the SRES he has just intercepted and he will be authenticated.

Billing and privacy:
As long as the processing of data is in the responsibility of one operator alone, he will be able to make his own decisions to a large extent. Otherwise, legal regulations (e.g. privacy laws) apply. A billing record in GSM contains at least the following data:

- Number of the subscriber who made the call.
- Number of the subscriber called.
- Location identifier consisting of the switching center location area and cell-identifier.
- Trunk group (the physical line, through which the switching centerconnects the call).
- Serial number and type of the mobile terminal used.

- beginning, end, duration of the call.

We also made flash animations to further clarify the things you have just learned with 2 additional methods on how to attack a wireless LAN as well as a telephone installation, also known as POTS.

Tools to perform a MITM attack with

Ettercap

Ettercap is described by it's coders as follows: „Ettercap is a multipurpose sniffer/interceptor/logger for switched LAN.
It supports active and passive dissection of many protocols (even ciphered ones) and includes many feature for network and host analysis."

Due to it's easy to use GUI, ettercap is a deadly weapon in the hand of even the most apprentice man in the middle. Some of it's features include:

- Characters injection in an established connection : *you can inject character to server (emulating commands) or to client (emulating replies) maintaining the connection alive !!*
- SSH1 support : *you can sniff User and Pass, and even the data of an SSH1 connection. ettercap is the first software capable to sniff an SSH connection in FULL-DUPLEX*
- HTTPS support : *you can sniff http SSL secured data... and even if the connection is made through a PROXY*
- Remote traffic through GRE tunnel: *you can sniff remote traffic through a GRE tunnel from a remote cisco router and make mitm attack on it*
- Plug-ins support : *You can create your own plugin using the ettercap's API.*
- Password collector for : *TELNET, FTP, POP, RLOGIN, SSH1, ICQ, SMB, MySQL, HTTP, NNTP, X11, NAPSTER, IRC, RIP, BGP, SOCKS 5, IMAP 4, VNC, LDAP, NFS, SNMP, HALF LIFE, QUAKE 3, MSN, YMSG (other protocols coming soon...)*
- Paket filtering/dropping: *You can set up a filter that search for a particular string (even hex) in the TCP or UDP payload and replace it with yours or drop the entire packet.*
- OS fingerprint: *you can fingerprint the OS of the victim host and even its network adapter*
- Kill a connection: *from the connections list you can kill all the connections you want*
- Passive scanning of the LAN: *you can retrive infos about: hosts in the lan, open ports, services version, type of the host (gateway, router or simple host) and extimated distance in hop.*
- Check for other poisoners: *ettercap has the ability to actively or passively find other poisoners on the LAN*

- Bind sniffed data to a local port: *you can connect to that port with a client and decode unknown portocols or inject data to it (only in arp based mode)*

Get ettercap here: http://ettercap.sourceforge.net/

Ethereal

Ethereal is a free network protocol analyzer for Unix and Windows. It allows you to examine data from a live network or from a capture file on disk. You can interactively browse the capture data, viewing summary and detail information for each packet. Ethereal has several powerful features, including a rich display filter language and the ability to view the reconstructed stream of a TCP session.
Ethereal is mostly used to analyze captured network traffic. It's powerful interface allows one to easily collect all the data one needs to commit a MITM attack.

It's primary use in a MITM attack would be in the early stages, where you try to grab as much information about the victim network as you can get. Ethereal will assist you greatly, should you have planned such a malice task.

Get ethereal here: http://www.ethereal.com/

tcpdump

Tcpdump is widely known an used on *nix plattforms to get an impression on what network traffic looks like. It is described by it's man page as follows:

Tcpdump prints out the headers of packets on a network interface that match the boolean *expression*.

It is a tool not unlike a swiss army knife, can be used for a wide range of network related tasks, eg. Output captured by tcpdump can be analyzed by ethereal.

It is also very usefull in the early stages of a MITM attack to get an impression on what the „opposition" looks like.

Tcpdump is supplied by most *nix as a standart tool.

Countermeasures against and detection of MITM attacks

Encryption

1) General remarks

MITM-attacks are usually performed with a specific goal in mind such as altering the information exchanged or catching a password. Therefore the attacker must partly 'understand' the exchanged information.
Though encryption does not prevent a MITM-attack, it turns a MITM-attack useless by prohibiting that the attacker can use
the information gathered because he won't understand it.

2) Types of encryption

2a) Symmetric encryption

With symmetric encryption the keys for encrypting and decrypting are identical. When used in networking, this encryption is based on exchanging the
keys while establishing the connection.
An MITM-attack by C (Carol) against A (Alice) and B (Bob) is possible if
"(...)an opponent Carol intercepts Alice's public value and sends her own public value to Bob.
When Bob transmits his public value, Carol substitutes it with her own and sends it to Alice.
Carol and Alice thus agree on one shared key and Carol and Bob agree on another shared key.
After this exchange, Carol simply decrypts any messages sent out by Alice or Bob, and then reads and possibly modifies them before re-encrypting with the appropriate key and transmitting them to the other party.
This vulnerability is present because Diffie-Hellman key exchange does not authenticate the participants. Possible solutions include the use of digital signatures and other protocol variants."
[quotation from http://www.rsasecurity.com/rsalabs/faq/3-6-1.html, checked 2002-30-11]

2b) Asymmetric encryption

Sender wants to send a message to receiver. For encrypting the receivers public key is being used. Decrypting that message cannot be done with the same key but must be done
with the receivers (secret) private key. If receiver wants to send an answer he uses the senders public key. Only the sender can decrypt that message with his private key.
Because no key needs to be exchanged (except the public keys that are public anyway) this encryption is secure against MITM-attacks.

Fixed ARP table

Check every entry in the ARP table of your computer and then flush it and reenter everything from hand. Make settings to not allow to automatically update the table.
If a new host enters your network, you'll have to add it by hand.

Ettercap

By using tools like ettercap you can easily detected manipulation of ARP entries. Ettercap has a built in detection mechanism which allows you to easily spot ARP changes and react accordingly.