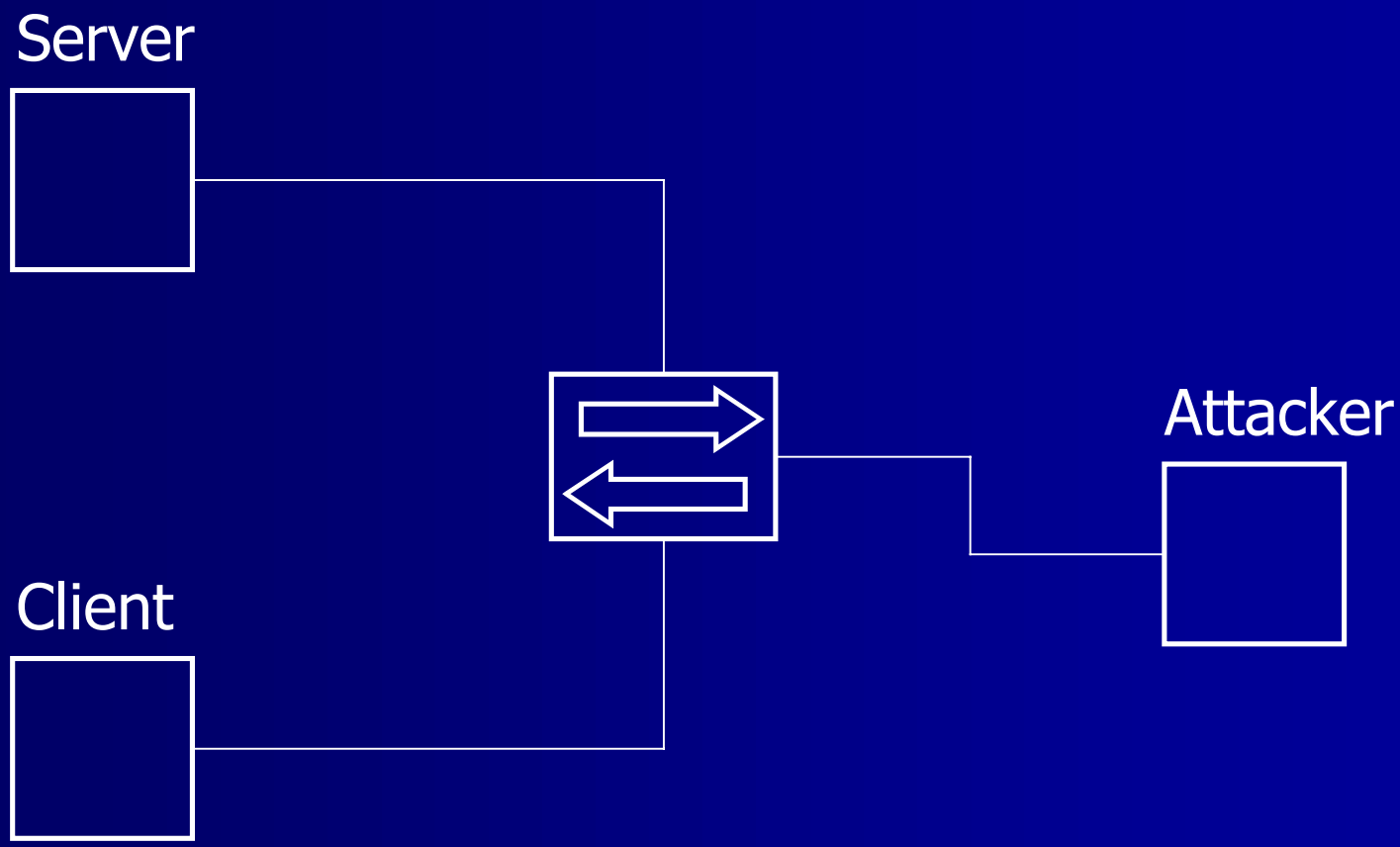
Alberto Ornaghi <alor@antifork.org>
Marco Valleri <naga@antifork.org>

Man in the middle attacks Demos

The scenario



Once in the middle…

- Injection
- Key Manipulation
- Downgrade attack
- Filtering

Injecting

- Possibility to add packets to an already established connection (only possible in full-duplex mitm)
- The attacker can modify the sequence numbers and keep the connection synchronized while injecting packets.
- If the mitm attack is a “proxy attack” it is even easier to inject (there are two distinct connections)

Injecting
Command injection

- Useful in scenarios where a one time authentication is used (e.g. RSA token). In such scenarios sniffing the password is useless, but hijacking an already authenticated session is critical
- Injection of commands to the server
- Emulation of fake replies to the client

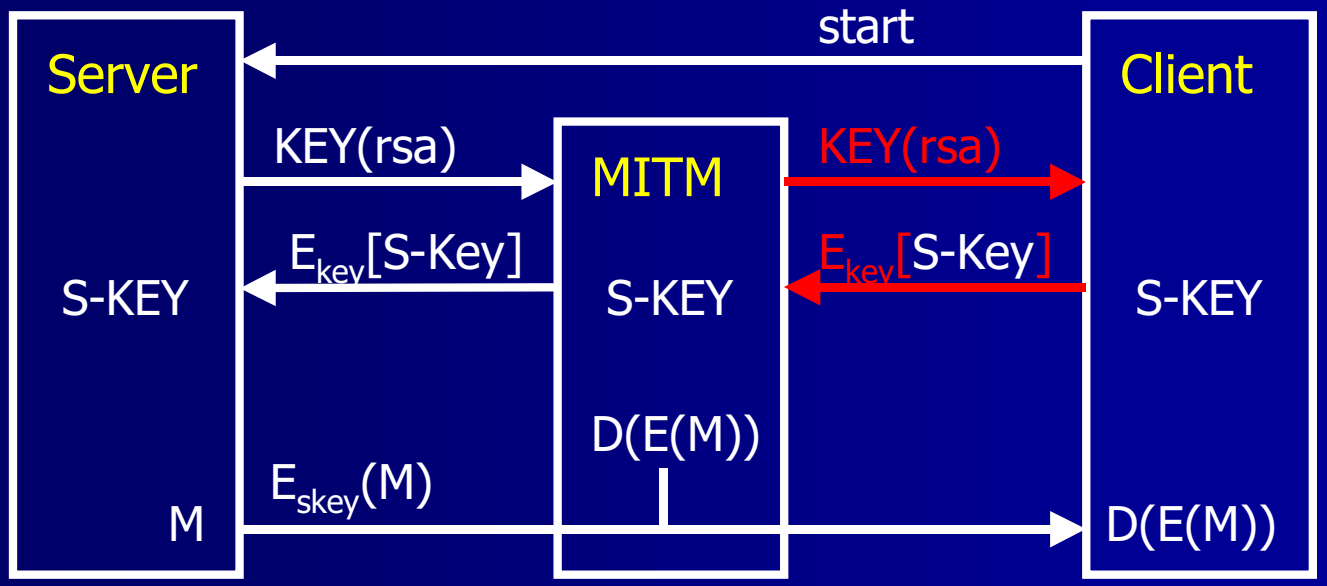
Command Injection DEMO

Key Manipulation

- SSH v1
- IPSEC
- HTTPS

Key Manipulation
SSH v1

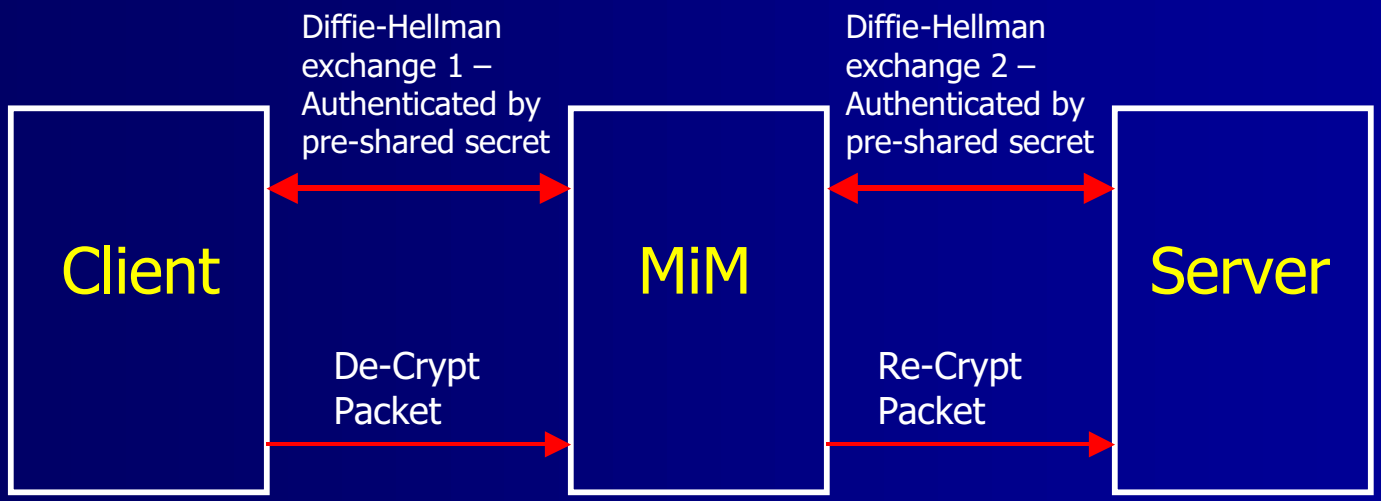
- Modification of the public key exchanged by server and client.



SSH v1 Attack DEMO

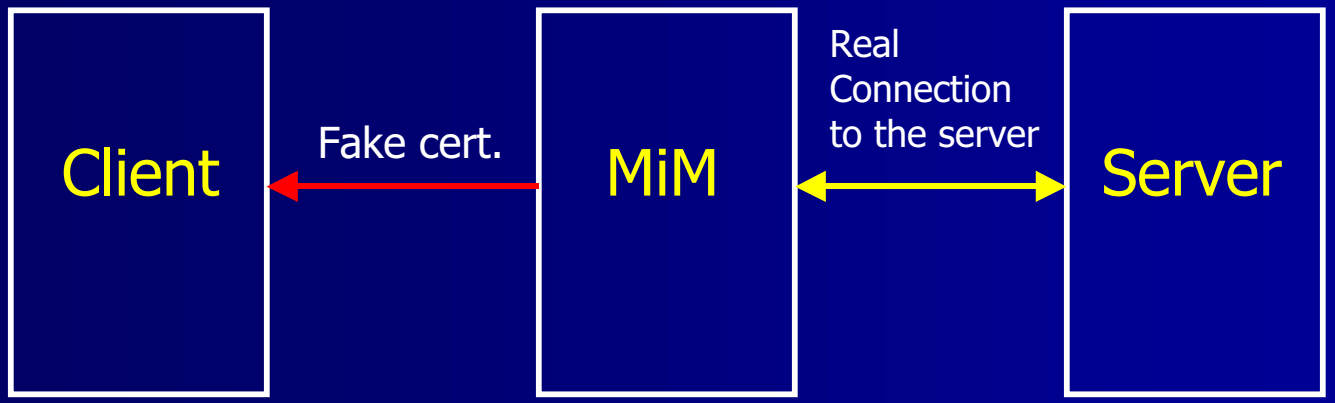
Key Manipulation
IPSEC

If two or more clients share the same "secret", each of them can impersonate the server with another client.



Key Manipulation
HTTPS

We can create a fake certificate (eg: issued by VerySign) relying on browser misconfiguration or user dumbness.



HTTPS Attack DEMO

Filtering

- The attacker can modify the payload of the packets by recalculating the checksum
- He/she can create filters on the fly
- The length of the payload can also be changed but only in full-duplex (in this case the seq has to be adjusted)

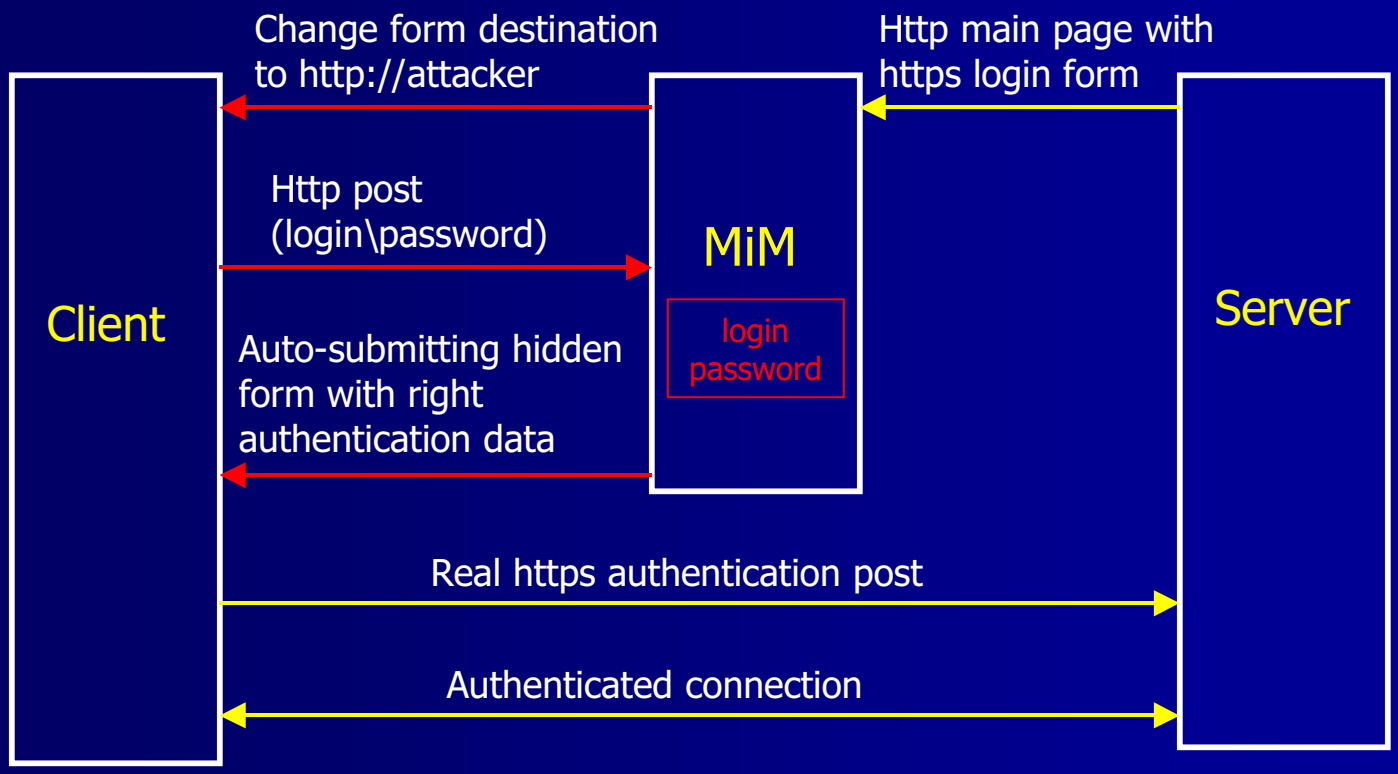
Filtering
Code Filtering / Injection

- Insertion of malicious code into web pages or mail (javascript, trojans, virus, ecc)
- Modification on the fly of binary files during the download phase (virus, backdoor, ecc)

Binary Modification DEMO

Filtering
HTTPS redirection

Let's see an example



HTTPS Redirection Attack DEMO

Downgrade Attacks

- SSH v2
- IPSEC
- PPTP

Downgrade Attacks
SSH v2 → v1

- Parameters exchanged by server and client can be substituted in the beginning of a connection. (algorithms to be used later)
- The attacker can force the client to initialize a SSH1 connection instead of SSH2.
  - The server replies in this way:
    - SSH-1.99 -- the server supports ssh1 and ssh2
    - SSH-1.51 -- the server supports ONLY ssh1
  - The attacker makes a filter to replace "1.99" with "1.51"
- Possibility to circumvent known_hosts

SSH v2 Downgrade DEMO

Downgrade Attacks
IPSEC Failure

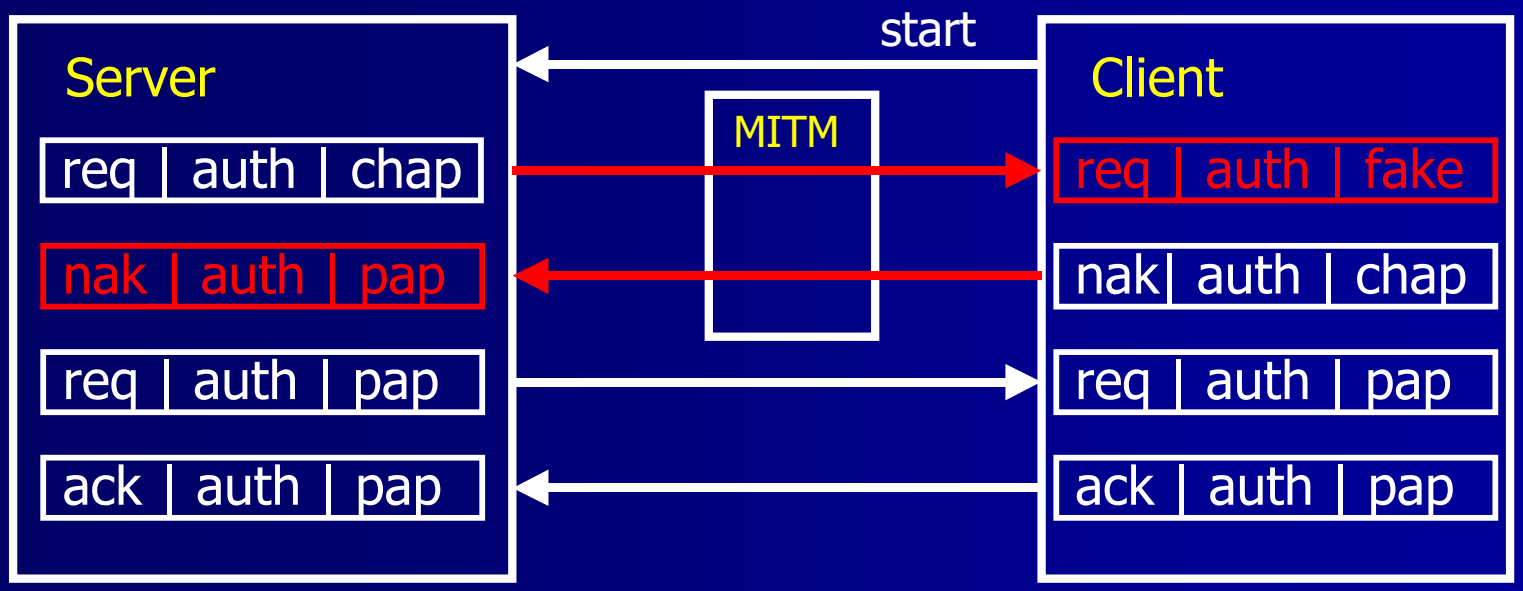
- Block the keymaterial exchanged on the port 500 UDP
- End points think that the other cannot start an IPSEC connection
- If the client is configured in rollback mode, there is a good chance that the user will not notice that the connection is in clear text

Downgrade Attacks
PPTP attack (1)

- During negotiation phase
  - Force PAP authentication (almost fails)
  - Force MS-CHAPv1 from MS-CHAPv2 (easier to crack)
  - Force no encryption
- Force re-negotiation (clear text terminate-ack)
  - Retrieve passwords from existing tunnels
  - Perform previous attacks
- Force "password change" to obtain password hashes
  - Hashes can be used directly by a modified SMB or PPTP client
  - MS-CHAPv2 hashes are not usefull (you can force v1)

Downgrade Attacks
PPTP attack (2)

Force PAP from CHAP



We don't have to mess with GRE sequences...

Downgrade Attacks
L2TP rollback

- L2TP can use IPSec ESP as transport layer (stronger than PPTP)
- By default L2TP is tried before PPTP
- Blocking ISAKMP packets results in an IPSec failure
- Client starts a request for a PPTP tunnel (rollback)
- Now you can perform PPTP previous attacks

PPTP Attack
DEMO

MITM attacks

Different attacks in different scenarios:

LOCAL AREA NETWORK:

- ARP poisoning
- DNS spoofing
- STP mangling
- Port stealing

FROM LOCAL TO REMOTE (through a gateway):

- ARP poisoning
- DNS spoofing
- DHCP spoofing
- ICMP redirection
- IRDP spoofing
- route mangling

REMOTE:

- DNS poisoning
- traffic tunneling
- route mangling

WIRELESS:

- Access Point Reassociation

MITM attacks
ARP poisoning

- ARP is stateless (we all knows how it works and what the problems are)
- Some operating systems do not update an entry if it is not already in the cache, others accept only the first received reply (e.g solaris)
- The attacker can forge a spoofed ICMP packets to force the host to make an ARP request. Immediately after the ICMP it sends the fake ARP replay
- Usefull on switched lan (the switch will not notice the attack)

MITM attacks

ARP poisoning - countermeasures

- YES - passive monitoring (arpwatch)
- YES - active monitoring (ettercap)
- YES - IDS (detect but not avoid)

- YES - Static ARP entries (avoid it)
- YES - Secure-ARP (public key auth)

- NO - Port security on the switch
- NO - anticap, antidote, middleware approach

ARP Poisoning DEMO

(all we have done until now…)

ARP Poisoning
Antidote Kernel Patch

- http://www.securityfocus.com/archive/1/299929
- "Kernel will send ARP request to test if there is a host at old MAC address. If such response is received it lets us know than one IP pretends to have several MAC addresses at one moment, that probably caused by ARP spoof attack."
- We can fake this protection if the ARP entry is not in the cache and the real mac address will be banned

Antidote Attack DEMO

MITM attack
Port stealing

- The attacker sends many layer 2 packets with:
  - Source address equal to victim hosts' address
  - Destination address equal to its own mac address
- The attacker now has "stolen" victim hosts' ports
- When the attacker receives a packet for one of the victims it generates a broadcast ARP request for the victim's IP address.
- When the attacker receives the ARP reply from the victim, the victim's port has been restored to the original binding state
- The attacker can now forward the packet and restart the stealing process
- Possibility to circumvent static-mapped arp entries

MITM attack
Port stealing - countermeasures

- YES - port security on the switch
- NO - static ARP

Port Stealing DEMO

Q & A

Alberto Ornaghi <alor@antifork.org>

Marco Valleri <naga@antifork.org>