

Man-in-the-Middle: Vulnerabilities in SSH/Public-Key

SYSC4907 PROPOSAL

Carleton University

September 16, 2004

Revision 1.0

Submitted By:

Group 84

Hubert Sugeng, 100*****

Jesse Pool, 100*****

Submitted To:

Professor Trevor Pearce

1.0 Objectives

To create a "Man-in-the-Middle" box to be used as a transparent step in a public-key encryption session. This will demonstrate how a public-key based session, Secure Shell (SSH), can be spoofed and monitored on the Internet by a malicious attacker.

1.1 Primary Objective

Demonstrate the Man-in-the-Middle (MiM) vulnerability of public-key encryption over SSH in a completely controlled environment. This environment will have the MiM inline between the client (target) and the server. The following will be studied:

- 1) Look at some of the weaknesses of public-key encryption.
- 2) Find ways of detecting that an attack has taken place.
- 3) Study the attack and look for workarounds.

1.2 Secondary Objective

Expand the MiM software to succeed when the attacker is not inline with the client and server. This will demonstrate a real-world scenario as played out over the Internet.

2.0 Background

Secure Shell (SSH) is a protocol that allows a user to securely login and administer a Unix like server from a remote computer. It provides command line access, giving a user an encrypted connection across a network, like the Internet or private LAN. SSH uses the widely accepted public-key encryption method to prevent a malicious attacker from viewing transmitted data in plain text.

Public-key encryption is a very common technique used in data transmissions. It is an asymmetric system where a sender uses a known public-key to encrypt a message for a recipient. The recipient can decrypt the message with its private-key. The public-key is advertised by the recipient, who is the sole possessor of the private-key. The public and private keys are a pair. Public-keys can only encrypt a message, while private-keys are only used to decrypt a message.

Man-in-the-Middle (MiM) is a method used when actively sniffing a network connection. It is often used in malicious attacks where a machine can act as a proxy between two valid network nodes. By employing this method, a machine can intercept data communications between a client and server. This interception is often undetected and can appear transparent to the parties involved.

3.0 Motivation

Public-key is increasingly becoming the encryption method of choice for moving data across the Internet. It is currently being used in Secure HTTP, Secure Socket Layer and many other common data delivery protocols. Most websites supporting encrypted login use the public-key method, which has proven easy to implement with common servers such as Apache.

Secure Shell has become the default replacement for the aged Telnet protocol. System administrators use SSH for much more than an encrypted remote command line. The protocol also supports secure tunneling and secure file transfers. These features are implemented over the widely accepted public-key encryption and make SSH an extremely invaluable tool that must be understood and used properly.

By studying public-key and the SSH protocol, advantages and disadvantages of current architectures can be assessed. As the Internet moves toward secure data communications, it is important to understand weaknesses in the system before malicious actions are taken against them.

4.0 Solution Description

The proposed solution will involve two (2) parts that will be combined together after they are completed. This will result in a working product that will show how Man-in-the-Middle vulnerabilities in public-key encryption can be exploited in the real-world.

The first part will involve a controlled environment where the MiM will be placed directly inline between an SSH client and server. This part will focus on merely exploiting the vulnerability. The second part will involve expanding the MiM so that it can operate in a real-world network. Here, it cannot be placed inline with an SSH client and server.

4.1 Controlled Environment: Inline Man-in-the-Middle

The primary objective of the project will assume that an SSH TCP session has already been compromised. This means that the target is already unknowingly sending its packets to the MiM (thinking it is the server) and the SSH server is sending its packets to the MiM (thinking it is the client).

During the initial connection handshake of the SSH protocol, a client and server exchange public keys that will be used to encrypt their data. The weakness in the handshake is that public-keys can be intercepted and changed in transit. Neither server nor client would be aware of the exchange.

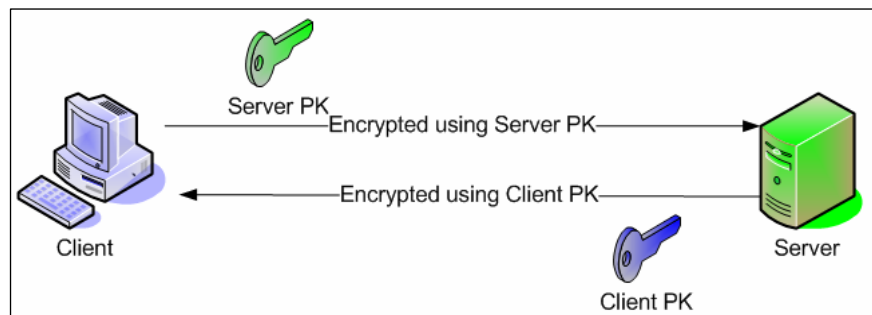


Figure 1: Public-key Transfer

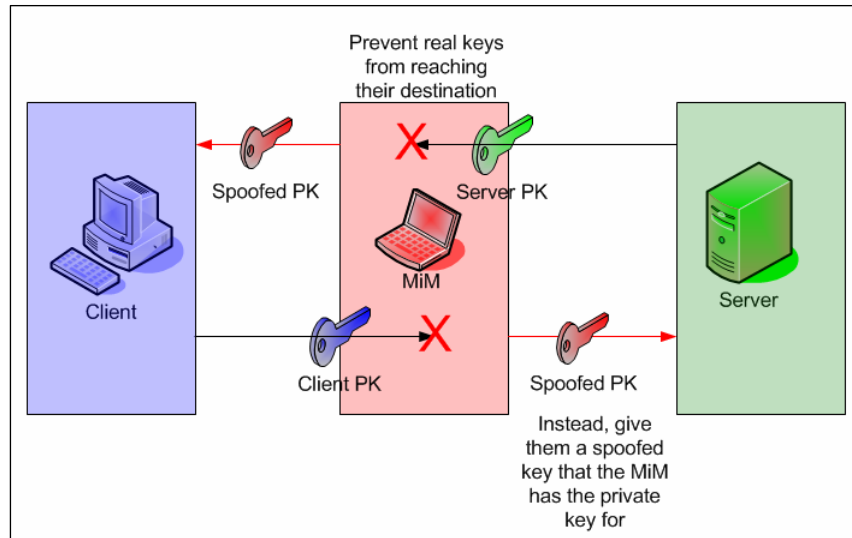


Figure 2: Public-key Interception

Once the MiM has established itself as a transparent step, it uses the intercepted public-keys to re-encrypt data that is sent between the server and client. In this way an SSH session can be read in plain text by the MiM. The following diagram demonstrates how this would look.

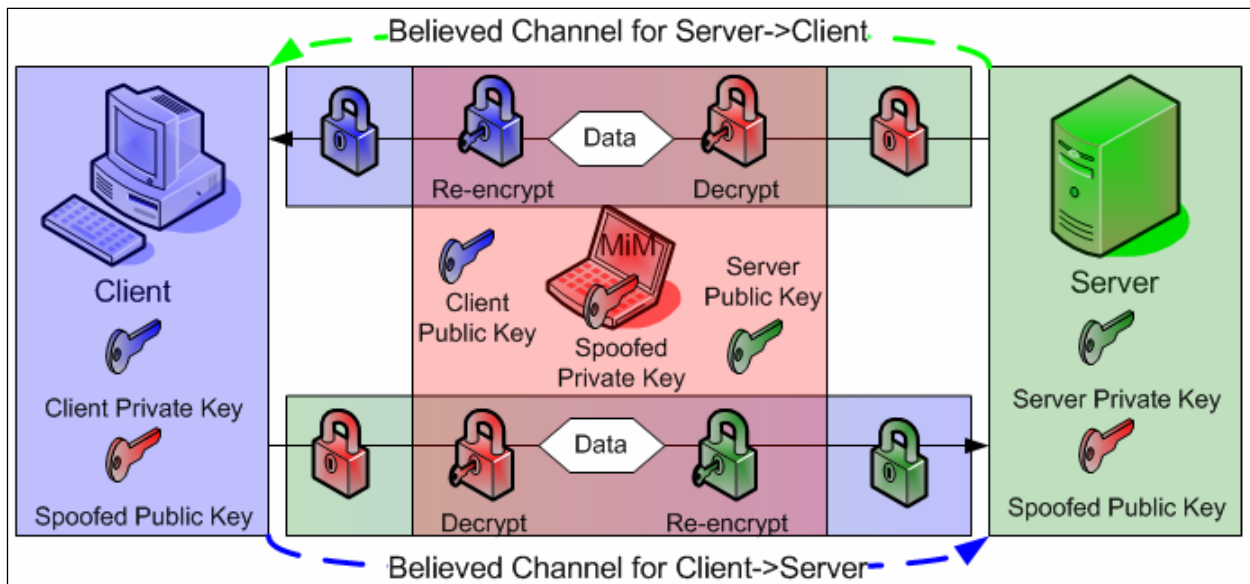


Figure 3: MiM Attack on Public-key

This is the goal of the first part of the application. The Man-in-the-Middle software will be able to listen in on the entire SSH session.

4.2 Real-World: Parallel Man-in-the-Middle

The secondary objective of the project will focus on demonstrating that Man-in-the-Middle vulnerabilities in public-key can be exploited in real-world environments. The study of this aspect deals with the combination of the Internet and Ethernet protocols.

The MiM software will exploit a weakness in IP networks, routed over Ethernet, called ARP (Address Resolution Protocol) spoofing. Ethernet switches, routers and operating systems use ARP when constructing and directing packets on the network. A Man-in-the-Middle application can use this aspect to force a LAN into thinking packets should be routed in its direction.

Ethernet devices do not initially know the layout of the machines around them. They use ARP packets to discover network topology by broadcasting to the entire system. The MiM software will corrupt a targets internal ARP table by replying to these broadcasts so that packets are routed at it instead of their true destination. In this way, packets that were supposed to go to a server will be read, modified, and re-injected back into the network by the MiM machine.

5.0 Milestones

Several milestones have been defined in order to organize the solution. Each milestone also contains a deliverable, which can be used in demonstration.

Objective	Analyze the SSH protocol and its relation with public key encryption
Deliverable	Summary and Analysis of the SSH protocol and drafting of the technical details of an SSH module for the MiM software.
Milestone Date	<i>September 28, 2004</i>

Objective	Create an IP forwarder to be used in our controlled “inline” environment (Like a proxy)
Deliverable	Working application that transparently forwards packets to another IP address and back. Or screen shots.
Milestone Date	<i>October 12, 2004</i>

Objective	Detect SSH session and spoof public keys (inline)
Deliverable	Logs showing real keys getting swapped with our spoofed keys
Milestone Date	<i>October 26, 2004</i>

Objective	Successful MiM attack on SSH session (inline)
Deliverable	Working program, or screenshots on paper, logs of the SSH session
Milestone Date	<i>November 16, 2004</i>

Objective	Sniff ARP packets over a network
Deliverable	Software shows an ARP table
Milestone Date	<i>December 6, 2004</i>

Objective	Evaluate the progress to date
Deliverable	A progress report outlining successes, failures and any needed changes to the milestones.
Milestone Date	<i>December 6, 2004</i>

Objective	Spoof Reply ARP (compromises the switch) and continue acting transparently with the now compromised switch
Deliverable	Log showing packets from target now going to MiM box
Milestone Date	<i>December 30, 2004</i>

Objective	Successful MiM attack on a telnet session
Deliverable	Working software program demonstrating the attack in progress (or screenshots on paper)
Milestone Date	<i>January 18, 2005</i>

~ Oral Presentations in January ~

Objective	Tie together the MiM with the SSH inline attack for a real-world application
Deliverable	Working application that can hijack an SSH session, and/or screenshots
Milestone Date	<i>February 15, 2005</i>

Objective	Prepare a presentation for the department
Deliverable	A presentation for the department about vulnerabilities of SSH (and public keys)
Milestone Date	<i>March 1, 2005</i>

~ Poster Fair in March ~

Objective	First Draft of Final Report
Deliverable	First draft of final report
Milestone Date	<i>March 25, 2005</i>

Objective	Final Report
Deliverable	All final deliverables
Milestone Date	<i>April 8, 2005</i>

6.0 Equipment Requirement

This project is primarily software based. However, several hardware requirements are listed below.

1. Ethernet Switch (Switching HUB)
2. Three (3) Computers (Client, Server, Attacker)

This equipment does not need to be provided by the Department.

7.0 Deliverables

The deliverables listed below will indicate completion of the project.

1. Man-in-the-Middle Solution for SSH
2. Final Project Report
3. Presentation for Department
4. Source Code: Compact Disc and CVS
5. Software User Manual