CSE 543 - Fall 2018 - Project 2: SSH Man-in-the-Middle

1 Dates

- **Out:** *October 9, 2018*
- **Due:** *November 2, 2018*

2 Introduction

In this project, you will develop two different types of man-in-the-middle (MITM) attacks against the SSH protocol that you implemented in Project 1 and implement a defense to prevent the second attack. In the first attack, you will create a MITM that pretends to be the target server, but instead opens a secure connection to the client that it can use to read/modify client communications forwarded to the target server. In the second attack, you will create a MITM that forwards encrypted messages to the target server without decrypting them, enabling it to replay client-server communications without detection.

Again, the system you produce must run on the Westgate Linux lab machines. These machines are named `cse-p204instXX.cse.psu.edu`, where `XX` is a number between 01 and about 40. All these machines should be identical and already have the OpenSSL library installed.

3 Overview

The original SSH protocol that you implemented is prone to two kinds of MITM attacks.

Server Spoofing: In the first attack, a malicious entity may intercept client communications intended for a target server to create a MITM connection, where the malicious entity pretends it is the server to the client and as a client to the target server. Since the SSH protocol does not authenticate that the public key provided by the server is really associated with the target server to which the client is intending to connect, a malicious entity can take advantage of this. The SSH systems take some measures to prevent this attack outside the crypto protocol. You will extend the server of Project 1 to produce a `MITM server` that performs this server spoofing MITM attack against the (unmodified) client and (mostly unmodified) server implemented in Project 1.

MITM Replay: In the second attack, a malicious entity may record and replay messages in the construction of an SSH session to collect and replay messages (at least a subset of them). You will implement this attack by extending your `MITM server` to also implement methods to record and replay communications between the (unmodified) client and (mostly unmodified) server from Project 1. In addition, you will implement an extension of the SSH project to prevent this attack by modifying the SSH protocol for the client and server from Project 1.

4 Project Tasks

For this project, please complete the following tasks:

1. **Download project tarball**:

   From `http://www.cse.psu.edu/~trj1/cse543-f18/p2-assign.tgz`. The project includes source code and a Makefile for building the tarball for the MITM code. Only a slight modification is made to the file `cse543-p1.c` to create `cse543-p2.c`.

2. **Use your Project 1 code**: For the client and target server, use the client and server you implemented in Project 1. For the attacks, you will run these components unmodified except for a small modification to one function (`receive_file`) used by the server. To prevent the attacks, you will need to modify the client and server SSH protocol implementations.

3. **Implement server spoofing attack**: In this MITM attack, you will modify the code from the Project 2 tarball (see above) to perform the **server spoofing** attack.

4. **Implement MITM replay attack**: In this MITM attack, you will extend the code from the Project 2 tarball further to record encrypted messages between a client and target server (from Project 1 code) and replay them later to construct a new SSH session and submit session messages. Use the ability to replay to craft an attack of your choice. **The most creative attack will earn a 10 point bonus.**

5. **Prevent record and replay MITM attack**: In this task, modify the SSH protocol implemented in the Project 1 client and server code to prevent the MITM replay attack. Not necessary to prevent the server spoofing attack.

6. **Evaluate Other's SSH Protocol**: You will be given another student's description of the SSH protocol. Identify any flaws in the description or justify the correctness of the description.

4.1 Download Tarball

The tarball for Project 2 consists of one new file `cse543-p2.c`, which replaces the old `cse543-p1.c` file, and the assigned code for Project 1.

The `cse543-p2.c` code only differs from the Project 1 code only that the interface to start the server is slightly different (see MIM_USAGE in the file). The first two arguments are the MITMs public and private keys as before. These keys will be different than the target server's public and private keys, of course. The third argument is still an IP address, but it is the IP address of the real (target) server. The fourth argument is the "MITM option" signifying which MITM attack is being performed: (1) server spoofing attack or (2) MITM replay attack.

The code for checking these options starts at line 137 of `cse543-p2.c`. For option "1", invoke `server_secure_transfer` from Project 1, but modify that function to perform the MITM proxy attack, as described below. For option "2" (not "1"), invoke the `mim_record_and_replay` function to perform the MITM record and replay attack, as described below.

The code in `cse543-proto.c` is missing the functions you implemented in Project 1 for the MITM server. Please copy the code from those functions where you see `/*** YOUR CODE from Project 1 ***/`.

4.2 Use Your Project 1 Code

The client and target server will be largely unchanged from Project 1. Thus, you can use the code you submitted.

If you would like to use my Project 1 client and server, you may obtain them for cost of 15 percent of your Project 1 grade.

Please modify the `open` call in the function `receive_file` in `cse543-proto.c` to include the flag `O_TRUNC` among the flags. Specifically, edit that line to the following

```
if ( (fh=open( fname, O_WRONLY|O_CREAT|O_TRUNC, 0700)) > 0 );
```

4.3 Server Spoofing Attack

In this attack, you will modify the code in `server_secure_transfer` to perform the SSH protocol as the server to a Project 1 client, but also initiate a new SSH connection to a Project 1 server as the client. This attack emulates the case where a MITM may obtain an IP address for another server or sit between the client and server on the network.

The attack must: (1) construct a SSH connection with the client sufficient for the client to transfer the file (from Project 1); (2) replay the client messages to create a second SSH connection to another Project 1 server; and (3) forward the client's messages to transfer the same file to the Project 1 server.

The easiest way to do that is to start with the server protocol and add the steps necessary to replay client messages to the Project 1 server sufficient to transfer the file. This should not take a lot of code given what you have for Project 1.

To test for this, run the client and MITM server on the same machine, and have the MITM server communicate with another server on a different machine. Both servers will receive connections listening to the same port (9165).

4.4 MITM Record and Replay Attack

In this attack, the aim is for the MITM to record messages sent by the Project 1 client to a target (Project 1) server and then replay these messages to create a new SSH session with the same target server to replay the previous messages, modifying the file on the target server in some manner.

To implement this attack, implement the function `mim_record_and_replay` to: (1) receive client messages; (2) record these client messages; (3) forward these messages to complete the client's correct SSH session with the target server (i.e., also forward the server messages back to the client); (4) replay the client's session messages to the target server to create a new SSH session; and (5) replay a subset of the client messages to attack the client's file sent to the server. Note that in this case, the MITM does not decrypt the client or server messages, but simply records the client messages and replays them to the server. In the replayed session, the MITM ignores the content of the target server messages.

Perform an attack that modifies the file from what the client sent originally. **There are some basic attacks, but the most creative attack will be awarded with a 10 point bonus.**

To complete all these steps will take a lot more code than the first attack, so you may create subroutines as necessary to implement the recording and replaying.

4.5 Prevent MITM Replay Attack

Change the Project 1 client and server code to prevent the MITM Record and Replay attack. Specifically, change the SSH protocol to enable the target server to detect that the fake client (the MITM) replaying messages does not really know the session key.

5 Testing

I will test your submission on machines in the Linux lab in W204 Westgate. The machines are named cse-p204instXX.cse.psu.edu, where XX is a number from 01 to at least 40.

As mentioned above, I will run the client and MITM on the same host, and run the target server on a different host.

You should SSH into those machines to verify that your code works. I developed and tested the project code on those machines, so should work fine, but it is up to you to make sure.

You will need to speak to the CSE IT folks if you do not have access to those machines.

6 Deliverables

Please submit the following:

1. A tarball of the MITM server code derived from `p2.tgz` using `make tar`.
2. A tarball of the extended Project 1 code implementing the defense against replay attacks using `make tar` over the project 1 code.
3. PDF describing your attack (for record and replay) and how defense prevents - briefly, but sufficiently for me to know what I should be looking for.

7 Grading

The assignment is worth 100 points total broken down as follows.

1. I can build and run what you have submitted without incident (10 points).
2. Server Spoofing Attack (20 points).
3. MITM Replay Attack (30 points).
4. SSH protocol to prevent MITM Replay Attack and protocol description (30 points)
5. 10 points for your prior efforts on writing SSH protocols. (10 points)