Information and Organisational Security

Guides for Practical Classes

João Paulo Barraca and Vitor Cunha

Department of Electronics, Telecommunications and Informatics
University of Aveiro

2018–2019

Contents

7

Secure communication using SSH

7.1 Introduction

In this guide you are going to configure and explore the use SSH (Secure Shell) secure sessions. You will need two Linux computers (virtual machines are ok). One computer, M1, will have the role of server and must have two network interfaces (NICs). The other computer, M2, will have the role of client and only needs one network interface.

7.2 Configuration

In this section you are going to configure the server (M1), the client (M2) and the local network where both connect.

7.2.1 Server Computer(M1)

Server computer must have two network interfaces, one to connect to an external network, to connect to the Internet, and the other to connect to an internal network, to communicate with M2.

External network interface configuration

Connect on of the server (M1) network interfaces (e.g., interface eth0) to an external network that provides connection to the Internet. If your server is a virtual machine, in the virtualizer (e.g., VirtualBox) you must attach the virtual interface to `Bridged Adapter` or `NAT`. We will designate this interface as the external interface.

At the operating system level, using the Network Manager, you must configure the interface to use `DHCP` to get its network configuration. Confirm that the internet correctly receives the network configuration and that the computer is connected to the Internet.

Services' configuration

Check if the `Telnet` and `SSH` services are installed and enabled in the server computer (M1). For that, use the following command to check if the TCP ports used by those services are available to accept connections.

```
netstat -atn
```

With the following commands you may install and configure the SSH and Telnet services in a Linux computer. Use those that are adequate to install the services missing in M1.

Start by updating the repositories with:

```
sudo apt update
```

Install the Telnet service (`telnetd`), using:

```
sudo apt install telnetd
```

If you need to restart the Telnet service, you may use the command:

```
sudo /etc/init.d/openbsd-inetd restart
```

Install the SSH service with the following command:

```
sudo apt install openssh-server
```

If you need to restart the SSH service, you may use the command:

```
sudo service ssh restart
```

Check that both services are up and running.

You also need to install `Wireshark` because you will need to capture and analyse packets exchanged between M1 and M2 computers.

Internal network interface configuration

The second network interface of M1 (e.g., eth1) will connect to an internal network to which M2 computer will also be connected. If your server is a virtual machine, in the virtualizer, you must attach the virtual interface to `Internal Network` and select the `intnet` internal network. We will designate this interface as the internal interface.

At the operating system level, the internal interface must be configured to use a static IP address and mask, for example the `10.0.0.1/24` that we will assume from now on. Use the `Network Manager` to configure a static IP address for the internal interface.

The internal network connected to internal interface will communicate to the Internet through the M1 computer. For that, M1 must route to the external network the traffic incomming from the incomming network and not target to the M1 computer. For this, you must execute the following two commands, the first to enable IP Forwarding and the second to enable `NAT`, in which `ethX` is the name of the external interface (change as the interface name as required).

```
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -A POSTROUTING -t nat -s 10.0.0.0/24 -o ethX -j MASQUERADE
```

7.2.2 Client Computer (M2)

Network interface configuration

The network interface com computer M2 will connect to the internal network to which M1 computer is already connected. If your server is a virtual machine, in the virtualizer, you must attach the virtual interface to `Internal Network` and select the `intnet` internal network.

The network interface must be configured with a static IP adress, for example the `10.0.0.2/24`, that we will assume for now on. You must also provide the IP addresses for the `Gateway` and for a `DNS Server`, that must be the IP addres of the M1 internal interface and the IP address of the `DNS Server` in

use by M1, respectively. Use the `Netwrok Manager` to configure the interface and check if it cammunicates with the Internet through M1.

Instalation of applications

By default, SSH and Telnet client applications come installed in most Linux distributions. In case you need to install any of them, you may the following commands (use the ones that apply):

```
sudo apt-get install telnet
```

and/ or

```
sudo apt-get install ssh
```

Note: A client application for Windows is `PuTTY`, available at `http://www.chiark.greenend.org.uk/~sgtatham/putty`. It is both a Telnet and SSH client.

7.3 Inspecting network traffic

In the client computer, using `Wireshark` start a capture of the traffic exchanged between the server and the client. Apply a filter to only observe `Telnet` and `SSH` packets, by applying the following rule in the filter window:

```
telnet || ssh
```

With this filter applyed, `Wiresshark` will show all the SSH and Telnet packets exchanged between server and client computers.

7.3.1 Inspecting `Telnet` traffic

In the client computer (M2), with an active traffic capture, start a `Telnet` session to the server computer (M1). `Login` and and list the contents of the root folder, using the command:

```
ls /
```

Close the `Telnet` session, stop the `Wireshark` capture and analyse the captured packets. Can you observe, in the captured packets, the contents of the root folder you listed above?

Still analysing the captured Telnet traffic, can you find the password you used to login when starting the Telnet session in the server computer (M1)?

What do you conclude regarding the security of the Telnet protocol?

7.3.2 Inspecting SSH traffic

In the client computer start a new capture and apply a filter to only show the captured SSH packets.

In the client computer (M2) start an SSH session with the server computer (M1). In case it is the first connection to that server, the server will authenticate presenting its public key and you are asked if you trust the key and want to save it. Say yes to proceed. Login into the server and list the contents of the root folder using the command:

```
ls /
```

Close the `Telnet` session, stop the `Wireshark` capture and analyse the captured packets. Can you observe any data in the packets. What do you conclude about the general security of SSH, comparing to Telnet?

7.4 Authentication in SSH

In this section you are going to analyse how the server and clients and clients authenticate in the SSH protocol. It is assumed that SSH configuration files are in their original state, i.e., no change has been made to the SSH configuration.

7.5 Server autentication

In the first interaction of an SSH client with an SSH server, the client must validate the server identity. For that purpose, the SSH server send its public key to the client and the respective fingerprint in presented to the user, so she can verify it and indicate if the key is correct or not, i.e., if the user trusts the server to which a session is being started, or not. For this, the user must be in the possesion of the public key fingerprint of the SSH server to which she wants to connect (she should be given the public key fingerprint

when her account was created). To get the fingerprint of the SSH server public key, issue the following command in the server computer, and write the fingerprint value:

```
sudo ssh-keygen -l -f /etc/ssh/ssh_host_ecdsa_key.pub
```

In the client computer (M2) eliminate the public keys that you possibly stored from previous SSH connections. For that purpose, use the following command:

```
rm ~/.ssh/known_hosts
```

From the client computer, start an SSH session with the server, using the following command, where *username* is your account name in the server and *host* is the name or IP address of the computer with the SSH server (M1):

```
ssh username@host
```

The user in the client computer will be presented with a message similar to the one shown bellow, asking if the user trusts the identity of the SSH server she is trying to connect to. Compare the presented public key fingerprint with the fingerprint you get in the SSH server. If they are equal, then you are trying to connect to the SSH server you intent to connect and you must answer `yes`.

```
The authenticity of host '10.0.0.1 (10.0.0.1)' can't be established.
RSA key fingerprint is 6a:de:e0:af:56:f8:0c:04:11:5b:ef:4d:49:ad:09:23.
Are you sure you want to continue connecting (yes/no)? no
```

After the confirmationm the client stores the public key in the `~/.ssh/known_hosts` file and the user is never again requested to confirm that SSH server public key, as long as the server keeps presenting the same public key.

Proceed to start the SSH session, by introducing your password to authenticate yourself to the SSH server. Login and password is the default mechanism for user authentication with SSH servers.

7.6 User authentication with an assymetric key pair

In the client computer you must generate an assymetric key pair to use for client authentication when starting a ssesion in an SSH server. For that, use the following command:

```
ssh-keygen -t rsa
```

Press `Enter` to accept the default name for the file where the key pair will be stored, and define a password for key protection.

Now, you need to install your public key in the SSH server, in order the server can use it to authenticate you when starting SSH sessions. For that purpose, use the following command, where `~/.ssh/id_rsa.pub` is the default filename used to store client keys (replace it with the name of the file where you stored your key pair, in case you haven't used the default file name), and `username` and `server` are the name of the account and the name of the computer where you want to install your public key:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub <username>@<server>
```

This command stores your public key in the `~/.ssh/authorizedkeys` file in the home folder of the user account in the server computer. This public key will be used to verify the user possesion of the corresponding private key, whenever the user starts an SSH session with the server.

In the client computer, start a new SSH session to the SSH server in M1. Notice that your public key is used for your authentication and you are not requested to introduce your password.

Compare public key authetication with password authentication when starting an SSH session and analyse the adavantages and disadvantages from one in relation to the other.

Public key authentication on SSH is substancially different from the authentication in SSL protocol. Explain the differences.

Do you think it is adequate to use the SSH public key authentication to authenticate Web servers in the Internet, for example to authenticate Google or Facebook servers? Explain.

7.7 User authentication with one-time passwords

Situations may exist where assymetric key pairs might not be the most adequate mechanism to authenticate users when establishing SSH sessions. For example, one such situation may occur when the user uses a shared computer and fears that its key pair can be compromised. In such situations, one-time passwords may be the most adequate mechanism for user authentication when establishing an SSH session with a remote server, since one-time

passwords can only be used one single time (they are not reusable), and therefore no security mechanisms are needed to secure its use.

7.7.1 Server configuration

To use one-time passwords in SSH, you need to install `OTPW` in the server. Use the following command:

```
sudo apt-get install libpam-otpw otpw-bin
```

Then you need to configure PAM (*Plugable authentication Module*) module of SSH. For that, you must edit the `/etc/pam.d/sshd` configuration file (using vim or gedit, for example). You must disable password authentication, by commenting the following line (by placing a # character in the beginning of the line):

```
#@include common-auth
```

Then, you must add the following two lines to enable user authentication using one-time passwords:

```
auth required  pam_otpw.so
session optional pam_otpw.so
```

Then you must configure the SSH server to accept one-time passwords. For that, you must edit the `/etc/ssh/sshd_config` file to enable the following three parameters (check that none of these parameters is duplicated, to avoid server faillures when starting):

```
UsePrivilegeSeparation yes
ChallengeResponseAuthentication yes
UsePAM yes
```

You also need to disable password authentication. However, we will allow public key authentication to prevent the possibility of all one-time passwords were used. For that, edit again the `/etc/ssh/sshd_config` file and check the following two parameters have the indicated values:

```
PubkeyAuthentication yes
PasswordAuthentication no
```

SSh server configuration is completed, and must be restarted.

7.7.2 One-time passwords generation

One-time passwords for user authentication must be generated and given to the user, in order she can use to authenticate when establishin an SSH session. To generate the user one-time passwords, the user must be logged in the SSH server (not has root) and must execute the following command, where `fname.txt` is a name for the file to store the generated one-time passwords. When executing the command, the user will be requested to introduce a prefix for the one-time passwords, that the user must memorize because it will be requested, together with the one-time password, when authenticating.

```
otpw-gen > fname.txt
```

Look at the contents of the file with the generated one-time passwords. Notive that the one-time passwords are numbered and that each occupies two columns to make them easy to read. When the user is requested to introduce a one-time password for authentication, she must concatenate the text in the two columns, i.e., the blank space between the two columns must not be introduced.

The user must print the file to bring the one-time passwords with her, in order she can use them whenever she establishes a new SSH session with the server.

7.7.3 Using one-time passwords

The establishment of a new SSH session is done with the same command as indicated elsewhere above in this guide, but now, the server will request one of the one-time passwords the user generated. The requested one-time password is identifyed by its number in a format similar to:

```
Password 123:
```

The full one-time password to be introduced is composed by two parts: the prefix (defined by the user when she generated the one-time passwords) and the identifyed one-time password that the user must read from the file containing the one-passwords she generated). The user must concatenate the prefix with the requested one-time password. If the correct password is introduced, SSH session is established.

7.8 Bibliography

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