Laboratory Assignment 3
Secure Shell (SSH)

April 23, 2008

1 Purpose

In this laboratory assignment you will learn how to use SSH Secure Shell for Workstations Windows client. You will also use two tools, i.e., Wireshark and GPG4win, which are introduced by our previous labs. The SSH Secure Shell for Workstations Windows client is an essential tool for today's network environment which has power for against the numerous security hazards that threaten network communications. It allows secure network services over an insecure network. It replaces other, insecure terminal applications, such as Telnet and FTP and allows you to securely login to remote host computers, to execute commands safely on a remote computer, and to provide secure encrypted and authenticated communications between two hosts in an untrusted network.

2 Preparations

Recommended reading is specified as below.

1. SSH
   (http://en.wikipedia.org/wiki/Secure_Shell)
2. SSH Secure Shell online help
   (http://www.ssh.com/support/documentation/online/ssh/winhelp/40/)

3 Reporting

To pass this assignment you have to send a report(1-2 pages) to your teacher at yan.wang@hh.se. The report with at least the following items:

1. **Your names and IDs**;
2. **Introduction** giving an overview of the assignment and content of the report;

3. **Answers** for the questions that you can find in the **Exercise Section**;
4. **Conclusions** containing your reflections about the secure terminal applications and the assignment.

4 Introduction of SSH

SSH Secure Shell for Workstations Windows client uses SSH protocol(version 2, i.e., SSH2).

- SSH (Secure Shell) is a protocol for creating a secure connection between two computers.
- The secure SSH connection provides authentication and encryption.
- SSH also provides compression.
- SSH uses multiple ciphers for encryption, including e.g. 3DES, Blowfish and AES.
- SSH protocol uses password, public key, certificate, smart card, PAM and SecurID authentication methods.
- SSH was designed as a replacement for the legacy 'telnet' application.
- The computer being connected to must be running an ssh daemon, or server, process.
- The computer which is attempting to connect must be running an SSH client. Automatic and secure authentication of both ends of connection. Both the server and the client are authenticated to prevent identity spoofing, Trojan horses, etc.

5 Exercises

The following exercises are to be done in the laboratory. And please answer the questions in your report.

You can remotely login the server **remote.stud.hh.se** by using your own account and password (as the same as the regular ones to access workstation in the lab).

5.1 Configuration

Before establishing a connection to a remote host computer, you should first check your connection settings. The connection setting can be changed by using the Setting option, which can be found on the toolbar and the **Edit** menu. The **Settings** dialog as a tree structure, click on a branch to display the settings associated with it. You can change the settings by changing the selections displayed on the right hand side of the settings window.

5.2 Profile Setting

Go to the **Settings** tree structure, click on **Connection** branch. Feel free to select your desired items, and answer the following questions in your report.

- **Encryption algorithm**
  Question: What do u choose from the dropdown menu as your desired encryption algorithm? Why u choose it?
  Question: If you go to **Cipher List** under the **Connection**, you will see DES is not considered as a default choice. Why?
  Question: What will happen, if u choose **none** as the encryption algorithm?

- **MAC algorithm**
  Question: What does MAC stand for?
  Question: What will happen, if u choose **none**?

- **Terminal Answerback**
  Question: Explain briefly 2 of the possible choices:ansi, vt100, vt102, vt220, vt320 and xterm.

5.3 Global Setting

Go to the **Settings** tree structure, click on **Global Setting** branch. Feel free to select your desired items, e.g., appearance, user authentication, etc.

In most situations, the most convenient user authentication methods are public-key authentication. Complete the flowing tasks and answer the questions in your report:

- Generate a key pair for public-key authentication.
  After you finish your key generation, you will be asked to upload your new key. **Question:** What kind of key is supposed to be uploaded: public key, private key or a key pair? **Question:** Where you can find your uploaded key? (In which folder of the remote host computer?)

- Modify your *authorization* file manually to allow connection with the new key. (If the *authorization* file is already there, please just check it.)
  For more information, see SSH Secure Shell Windows client help Section Manually Editing the Authorization File.

- Click the button **Configure** to write the identification file that is used by the command line tool ssh2.exe to specify which keys can be used for authentication.

The most convenient server authentication methods are public-key authentication. Public host keys used in server authentication (remote host authentication) process can be managed using the **Host Keys** page of the **Settings** dialog. Answer the questions in your report:

- **Question:** How many host keys can you find in **Host Keys** pane?
- **Question:** How do they come? And where have they been stored?

5.4 File Transfer

- Go to My Computer — H:, create an empty folder named **NSlab3**.
- Generate a very simple file called **Test1.txt** which only contain one simple sentence, e.g., *How are you!* under the folder **NSlab3**.
- Copy **Test1.txt** into another file named **Test2.txt**

Next, we will compare the difference between the regular approach and SSH in downloading and uploading files. Please answer the following questions in your report.

- Start Wireshark.
- Copy and paste **Text1.txt** from H: to your local disk.

Question: Can you detect the content of the file **Text1.txt** by packet sniffer? If you can, what is it? How it comes? If you can not, why?

- Start Wireshark.
- Download **Text2.txt** from H: to your local disk by using **SSH Secure Shell** —**Secure File Tranfer Client**.

Question: Can you release the content of the file **Text2.txt** by packet sniffer? If you can, what is it? How it comes? If you can not, why?

(Hint: Actually, theoretically you can do it by using GPG4win and Wiresharp, because you own your private key which works for the download files. Back to the **Global Setting** step, instead create a new key pair, you can import a key pair produced by GPG4win. So you can collect and append all the encrypt messages with the help from Wireshark, and then use your private key to decrypt them in GPG4win. Of course, you can also export your key pair from SSH, and import it into GPG4win.)

References