Software Construction Laboratory

Week 7 Part 2

Lab 3

Mushi Zhou

Winter 2017

UCLA

Prof. Paul Eggert

Outline

Tools to Minimize Security Threats

- OpenSSH
- Ssh-agent
- GNU Privacy Guard
- Public & private keys

OpenSSH

- A connectivity tool for remote login with the SSH protocol
- Remote login -> Like you use Putty to login SEASNet Servers
- All traffics are encrypted to eliminate eavesdropping, connection hijacking, and other attacks
- Provides additional secure tunneling capabilities, authentication methods, and protective configuration options

SSH

- A program for logging into a remote machine
- Executing commands on a remote machine
- Provides secure encrypted communications between two untrusted hosts over an insecure network
- Connects and logs into the specified hostname (with optional user name)
- The user must prove his/her identity to the remote machine using one of several methods
- Manual Page: http://man.openbsd.org/OpenBSD-current/man1/ssh.1

SCP

- Copies files between hosts on a network
- Uses the same authentication and provides the same security as ssh
- Will ask for passwords or passphrases if they are needed for authentication
- File names may contain a user and host specification to indicate that the file is to be copied to/from that host
- Copies between two remote hosts are also permitted

The GNU Privacy Guard

- Free implementation of the OpenPGP standard
- Pretty Good Privacy (PGP) is an encryption program that provides cryptographic privacy and authentication for data communication
- PGP is often used for signing, encrypting, and decrypting texts, e-mails, files, directories
- In addition to PGP, GNUPG allows features a versatile key management system as well as access modules for all kinds of public key directories
- A command line tool with features for easy integration with other applications
- A wealth of frontend applications and libraries are available

Encryption Methods

Symmetric:

- Encryption and decryption share a secret key
- Relative easy to do
- Problem: difficult to share the key over insecure networks

Asymmetric:

- Encryption has different key than Decryption
- Problem: how to use different keys to produce the same results

Public & Private Key Pairs

- Asymmetric
- Keys are created in pairs
- Private key is private
- Public key can be public to anyone without compromising security
- Very long keys based on mathematical computations

Two goals of Public & Private Key Pairs

Authentication:

- Public key can be used to verify that it is the owner of the private key who is requesting access or sending files
- So anyone with the public key can verify that, but that is useless for anyone except for the host/server that the private key owner is trying to access

For example:

- If I want to “sign” a message, encrypt it with my private key
- Only I know private key, so no one else could create that message
- Everyone knows my public key, so everyone can check my claim directly

Two goals of Public & Private Key Pairs

Encryption:

- Only the private key can decrypt messages encrypted with public key
- Any person can encrypt a message using the public key
- But such a message can be decrypted only with the receiver's private key
- That means the public key encryption is irreversible (One way hash)

- This are a given features of the key pairs
- They do not work the other way around

Some Help for Assignment 7

To solve SSH hostname:

- If team is using the same network, use the local network address
- If team is using different internet, try use the real IP address (Could have potential problem, if the internet has certain security policies that is masking individual address parts, etc. (For example, if there is a NET, it will not know which machine you want to access if only given an IP address. And you can not SSH to SEASNet servers without using a UCLA WIFI)
- If is your task as a team to learn and figure out how to establish SSH connections for this assignment

Some Help for Assignment 7

To establish SSH connections you must prove your identity:

- Consider server client pairs, i.e. your host is the server for your teammates, and you are the client for your teammates' host servers
- Add accounts on each server for all clients
- Try to use created passwords to establish SSH connections
- Your task here is to figure out how to add accounts on ubuntu

Some Help for Assignment 7

Once you can establish SSH connection using passwords:

- Each team member should create his/her own public & private key pairs
- Use SCP to copy public keys to each other and configure SSH to use those public keys
- Establish connection without passwords
- Your task here is to figure out how to generate key pairs using The GNU Privacy Guard
- Figure out how to use SCP to copy files over
- Figure out how to allow SSH to use key pairs instead of passwords
- Finally, figure out how to allow all X-window sessions to display on your screen instead of the remote server

Some Help for Assignment 7

For the homework part:

- You are exploring the signature use of public and private keys
- Also some mathematic concepts behind generating keys

Things to Think About

- Isn’t it amazing that you just typed your hostname and you are connected. Are you still secure?
- If you don’t use a password, how was the authentication ensured?
- We copied the public key to our remote server, was that safe?