Operating System fundamentals

Configure and secure SSH

Karel de Grote Hogeschool

Contents

1. How does SSH work?
2. Encryption
3. Key based authentication
4. Key managers
5. SSH configuration

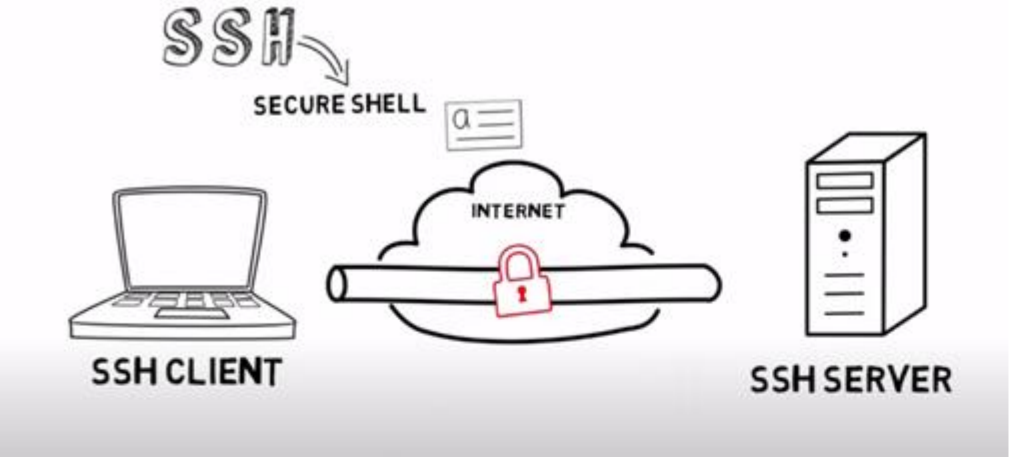
Course text

- Chapter 15. Configure and Secure SSH
  - (RedHat chapter 10 + extra material)
  - Access the Remote Command Line with SSH
  - Configure SSH Key-based Authentication
  - Customize OpenSSH Service Configuration

How does SSH work?

How does SSH work?

- server has a “fingerprint”
- ssh checks if the fingerprint corresponds to known fingerprints
- if ok: creates an encrypted connection with the server



Checking the fingerprint

```
The authenticity of host '192.168.56.110 (192.168.56.110)' can't be
established.

ED25519 key fingerprint is
SHA256:egKYWuLRs+un6P62++dUHet8I7ifbsl7PnxagouFxzE.

This key is not known by any other names

Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Examples

```
[student@workstation ~]$ ssh hosta
student@hosta password: student
...output omitted...
[student@hosta ~]$
```

user account not specified, current user

```
[student@host ~]$ ssh developer@hosta
developer@hosta's password: shadowman
...output omitted...
[developer@hosta ~]$
```

different user account specified

```
[student@host ~]$ ssh developer@hosta hostname
developer@hosta's password: shadowman
hosta.lab.example.com
[developer@hosta ~]$
```

do not login, simply run a command

Examples

```
[student@host ~]$ ssh developer@10.20.30.5
developer@10.20.30.5 password: shadowman
...output omitted...
[developer@hosta ~]$
```

Sometimes ip addresses are used when the hostname cannot be resolved using dns

```
[student@host ~]$ ssh developer@myserver.kdg.be
developer@myserver.kdg.be password: shadowman
...output omitted...
[developer@myserver ~]$
```

or a full domain name can be used

Identifying remote users

```
[developer@host ~]$ ssh developer@hosta
developer@hosta's password: redhat
[developer@hosta ~]$ w -f
 16:13:38 up 36 min,  1 user,  load average: 0.00, 0.00, 0.00
USER             TTY      FROM             LOGIN@   IDLE   JCPU   PCPU
WHAT
developer2   pts/0    172.25.250.10    16:13    7:30   0.01s  0.01s -bash
developer1   pts/1    172.25.250.10    16:24    3.00s  0.01s  0.00s w
[developer@hosta ~]$
```

Exercise

- Create an SSH connection from your graphical VM to the headless VM and look at the output of the “w” command

Encryption

2

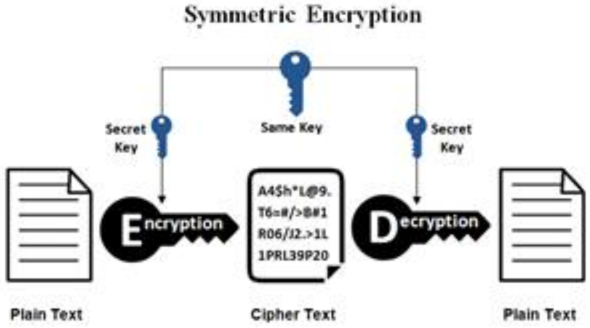
Symmetric and asymmetric encryption

- SSH uses 2 types of encryption
  - symmetric encryption: 1 key is used for encryption and decryption
  - asymmetric encryption: 2 keys are used (1 for encryption and the other for decryption)

Symmetric encryption

- One key is negotiated with which messages are encrypted
- The same key is used to decrypt the message

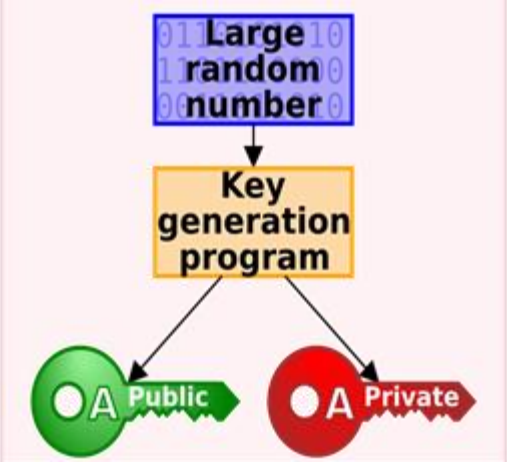
- advantage
  - encryption and decryption is fast
- disadvantage
  - you have to communicate the key with the other party -> possible security problem is somebody can intercept this



Asymmetric encryption

Two key are generated

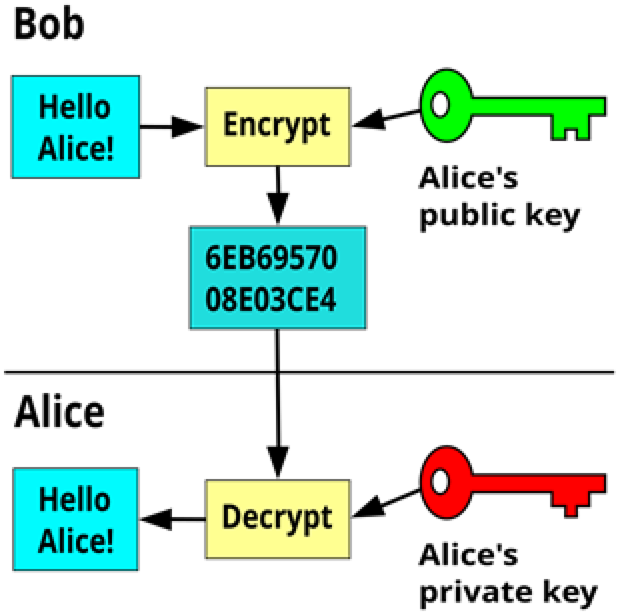
- the public key is visible for everybody
- the private key is not communicated with anybody



source: wikipedia

Asymmetric encryption

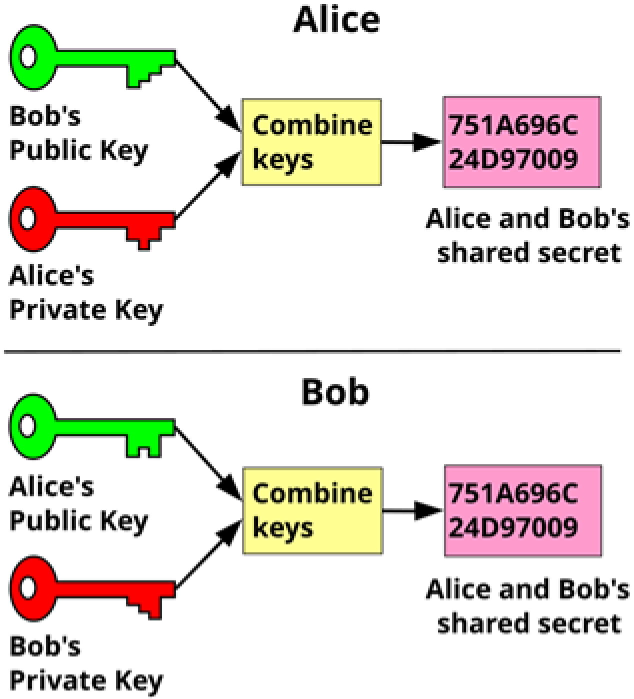
You can use one key to encrypt and the other to decrypt



source: wikipedia

Asymmetric encryption

Keys of both parties can be used in coniunction



source: wikipedia

SSH and encryption

- In order to create a connection with a remote server asymmetric encryption is used
- The fingerprint is the public key of the remote machine
- After this a new (symmetric) key is negotiated
- The new key is then used to encrypt all network traffic

Configuration and fingerprints

Take a look at these files on your virtual machine

- The configuration of SSH can be found in:
  - /etc/ssh/ssh_config (system wide)
  - ~/.ssh/config (per user)
- SSH saves fingerprints (public keys) of known servers in:
  - /etc/ssh/ssh_known_hosts (system wide)
  - ~/.ssh/known_hosts (per user)
- if the fingerprint does not correspond to the known one an error is displayed and access is denied
  - can be because of a man-in-the-middle attack
  - can also be a reset of the server

Key based authentication

SSH authentication

- You can login using SSH in two ways
  - with password
  - using an (asymmetrical) key
- Using a key:
  - generate a pair of keys using “**ssh-keygen**”
  - copy the public key to the server using“**ssh-copy-id**”
  - connect now using ssh

SSH key generation

```
[user@host ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa): Enter
Created directory '/home/user/.ssh'.
Enter passphrase (empty for no passphrase): Enter
Enter same passphrase again: Enter
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:vxutUNPio3QDCyvkYm1 user@host.lab.example.com
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|   .     .       |
|  o o     o      |
| . = o   o .     |
|  o + = S E .    |
| ..O o + * +     |
|.+% O . + B .    |
|=*oO . . + *     |
|++.      . +.    |
+----[SHA256]-----+
```

SSH key generation

- Remarks
  - you can protect the private key with a password
    - the password will be asked for, every time you use the private key
    - advantage: the password is not sent over the network. It stays on your local machine
  - using the -f option another name for the key pair can be specified (default is “id_rsa”)
  - if the name of an existing key pair is given, the keys are <u>overwritten</u>!
    - backup regularly!

Copy the public key to the server

```
[user@host ~]$ ssh-copy-id -i .ssh/id_rsa.pub user@remotehost
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
"/home/user/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new
key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if
you are prompted now it is to install the new keys
user@remotehost's password: redhat
Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'user@remotehost'"
and check to make sure that only the key(s) you wanted were added.
```

Exercise

- remove the ~/.ssh directory in your graphical VM and also in the headless server
- Create a key pair on your graphical VM (use a password to protect it)
- Can you find the location of the generated keys?
- Transfer the public key to the headless server
- Look what happens in the .ssh directory
- Look what happens to the .ssh directory on the headless server (authorized_keys)
- Create an SSH connection from the graphical VM towards the headless server. What happens when you have to provide the password for the private key?
- Log out and log back in with SSH. Do you need to provide the password again?

4

Key managers

What is a key manager?

- When you create a private key with a password, it needs to be provided every time you use it
- A key manager stores passwords locally so you don’t need to type them in every time

Manual key manager

- In the graphical version the key manager is automatically activated
- In the headless version you can enable the key manager using this command:
  eval $(ssh-agent)
- now you have to manually add the password to the key manager using **ssh-add**

Exercise

- Create a key pair on the headless server (using a password)
- Put the public key on the graphical VM
- Create twice an ssh connection to the graphical VM
  - do you need to enter the password every time?
- Start the key manager
- Add the password to the key manager
- Create an ssh connection with the graphical VM
  - normally you should not have to enter a password now

SSH configuration

5

SSH configuration

- You can improve the security of SSH
  - make sure that root cannot use SSH to login
  - disable the possibility to login using a password (enforce the key pair method)

Disabling root access

- Open the file **/etc/ssh/sshd_config**
- Search for “PermitRootLogin”
- Set it to “no”
- Restart the SSH service with **systemctl reload sshd**

Disable password acces with SSH

- Open the file **/etc/ssh/sshd_config**
- Search for “PasswordAuthentication”
- Set this to “no”
- Restart the SSH service with **systemctl reload sshd**

Exercises

6

Exercises

- KdG
  - …
- RedHat
  - ch10s02
  - ch10s04
  - ch10s06
  - ch10s07

KdG Karel de Grote
Hogeschool