LPI 110.3 - Securing data with encryption

Curs 2021 - 2022

ASIX M01-ISO 110 Security

Securing data with encryption

Description

Key concepts:

- ❏ Perform basic OpenSSH 2 client configuration and usage.
- ❏ Understand the role of OpenSSH 2 server host keys.
- ❏ Perform basic GnuPG configuration, usage and revocation.
- ❏ Use GPG to encrypt, decrypt, sign and verify files.
- ❏ Understand SSH port tunnels (including X11 tunnels).

Commands and files:

- ❏ ssh
- ❏ ssh-keygen
- ❏ ssh-agent
- ❏ ssh-add
- ❏ ~/.ssh/id_rsa and id_rsa.pub
- ❏ ~/.ssh/id_dsa and id_dsa.pub
- ❏ ~/.ssh/id_ecdsa and id_ecdsa.pub
- ❏ ~/.ssh/id_ed25519 and id_ed25519.pub
- ❏ /etc/ssh/ssh_host_rsa_key and ssh_host_rsa_key.pub
- ❏ /etc/ssh/ssh_host_dsa_key and ssh_host_dsa_key.pub
- ❏ /etc/ssh/ssh_host_ecdsa_key and ssh_host_ecdsa_key.pub
- ❏ /etc/ssh/ssh_host_ed25519_key and ssh_host_ed25519_key.pub
- ❏ ~/.ssh/authorized_keys
- ❏ ssh_known_hosts
- ❏ gpg
- ❏ gpg-agent
- ❏ ~/.gnupg/

SSH Secure Shell

The SSH protocol is used to provide secure remote login and other services. The SSH protocol uses public key cryptography for authenticating the remote host and providing an encrypted channel.

OpenSSH has largely replaced telnet as a remote client because telnet sends all data, including usernames and passwords, in clear (unencrypted) text whereas SSH encryption begins even before username authentication.

- ssh
- sftp
- scp

Client Configuration

SSH client can be configured i three levels of precedence (from + to -):

1. command line options
2. user-specific file
3. system-wide file

Configuration files:

- /etc/ssh/ssh_config
- /etc/ssh/ssh_config.d
- ~/.ssh/config
- ~/.ssh/known_hosts
- ~/.ssh/authorized_keys

Example /etc/ssh/ssh_conf

```
# Host *
#   ForwardAgent no
#   ForwardX11 no
#   PasswordAuthentication yes
#   HostbasedAuthentication no
#   GSSAPIAuthentication no
#   GSSAPIDelegateCredentials no
#   GSSAPIKeyExchange no
#   GSSAPITrustDNS no
#   BatchMode no
#   CheckHostIP yes
#   AddressFamily any
#   ConnectTimeout 0
#   StrictHostKeyChecking ask
#   IdentityFile ~/.ssh/id_rsa
#   IdentityFile ~/.ssh/id_dsa
#   IdentityFile ~/.ssh/id_ecdsa
#   IdentityFile ~/.ssh/id_ed25519
#   Port 22
#   Ciphers aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc
#   MACs hmac-md5,hmac-sha1,umac-64@openssh.com
#   EscapeChar ~
#   Tunnel no
#   TunnelDevice any:any
#   PermitLocalCommand no
#   VisualHostKey no
#   ProxyCommand ssh -q -W %h:%p gateway.example.com
#   RekeyLimit 1G 1h
```

Client configuration directives:

Host

Applies all forwarded declarations and options in the configuration file for those hosts that match one of the patterns given after the Host keyword

ForwardAgent

Specifies which connection authentication agent should be forwarded to the remote machine

ForwardX11Trusted

Specifies if X11 sessions should be automatically redirected to the remote machine

Port

Specifies the port number on which ssh connects to the remote host (default value is 22)

PasswordAuthentication

Set to yes to use password based authentication; no otherwise

RSAAuthentication

Specifies if RSA authentication is to be used

BatchMode

Specifies if username and password check on connection will be disabled. This option is generally used while invoking ssh from scripts to provide a non-interactive mode of operation.

CheckHostIP

Specifies if the IP address of the host should be checked for DNS spoofing

StrictHostKeyChecking

Specifies if new hosts should be automatically added by ssh to the .ssh/known_hosts file

IdentityFile

Specifies an alternate RSA authentication identity file to use

Cipher

Specifies the cipher method to be used for encryption

```
pue@debian:~$ ssh pue@172.16.5.1
The authenticity of host '172.16.5.1 (172.16.5.1)' can't be established.
ECDSA key fingerprint is SHA256:q3wSAaiFc4or9G9zcDdU7rugOY19vYb91LGmcFBFOuo.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.5.1' (ECDSA) to the list of known hosts.
pue@172.16.5.1's password:
Activate the web console with: systemctl enable --now cockpit.socket
Last login: Wed Sep 29 16:19:31 2021 from 172.16.5.254
```

```
pue@debian:~$ ssh pue@172.16.5.1
pue@172.16.5.1's password:
Activate the web console with: systemctl enable --now cockpit.socket
Last login: Thu Nov 18 18:18:57 2021 from 172.16.5.2
```

```
pue@debian:~$ ls -la ~/.ssh/known_hosts
-rw-r--r-- 1 pue pue 222 nov 18 18:18 /home/pue/.ssh/known_hosts

pue@debian:~$ cat  ~/.ssh/known_hosts
|1|HvsPSnyjD7I+Olx8lugkZ2/9VEc=|ONSttBnGTx8jRClfv7Io9F/F/64= ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBJk3c+WEBqzjpf7dSSJPxtKltRww60AgFOue
jpgkkEvxQkgXN7wSujZhLxRcdMGaObtnKcfyUfmEF3Jdn69XhNY=
```

Practice SSH Client

- Host Centos SSH Server

- Host Debian SSH client

Host Centos SSH server: start and verify

```
[pue@localhost ~]$ sudo systemctl start sshd
[sudo] password for pue:

[pue@localhost ~]$ sudo systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset:
enabled)
   Active: active (running) since Thu 2021-11-18 18:09:41 CET; 5min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 1013 (sshd)
    Tasks: 1 (limit: 23548)
   Memory: 2.3M
   CGroup: /system.slice/sshd.service
           └─1013 /usr/sbin/sshd -D
-oCiphers=aes256-gcm@openssh.com,chacha20-poly1305@openssh.com,aes256-ctr,aes256-cbc,aes
128-gcm@openssh.com,a>

nov 18 18:09:41 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
nov 18 18:09:41 localhost.localdomain sshd[1013]: Server listening on 0.0.0.0 port 22.
nov 18 18:09:41 localhost.localdomain sshd[1013]: Server listening on :: port 22.
nov 18 18:09:41 localhost.localdomain systemd[1]: Started OpenSSH server daemon.

[pue@localhost ~]$ ip a s ens3
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
default qlen 1000
    link/ether 52:54:00:b3:03:01 brd ff:ff:ff:ff:ff:ff
    inet 172.16.5.1/24 brd 172.16.5.255 scope global dynamic noprefixroute ens3
       valid_lft 3175sec preferred_lft 3175sec
    inet6 fe80::c8:eb2b:2bf:a6e9/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

[pue@localhost ~]$ nmap 172.16.5.1
Starting Nmap 7.70 ( https://nmap.org ) at 2021-11-18 18:16 CET
Nmap scan report for 172.16.5.1
Host is up (0.00023s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
3389/tcp open  ms-wbt-server
8080/tcp open  http-proxy
```

Host client debian: connect first time / exit

```
pue@debian:~$ ssh pue@172.16.5.1
The authenticity of host '172.16.5.1 (172.16.5.1)' can't be established.
ECDSA key fingerprint is SHA256:q3wSAaiFc4or9G9zcDdU7rugOY19vYb91LGmcFBFOuo.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.5.1' (ECDSA) to the list of known hosts.
pue@172.16.5.1's password:
Activate the web console with: systemctl enable --now cockpit.socket
Last login: Wed Sep 29 16:19:31 2021 from 172.16.5.254

[pue@localhost ~]$ id
uid=1000(pue) gid=1000(pue) grupos=1000(pue),10(wheel)

[pue@localhost ~]$ cat /etc/os-release
NAME="CentOS Linux"
VERSION="8"
...

[pue@localhost ~]$ exit
logout
Connection to 172.16.5.1 closed.
```

Host client debian: connect / exit

```
pue@debian:~$ ssh pue@172.16.5.1
pue@172.16.5.1's password:
Activate the web console with: systemctl enable --now cockpit.socket
Last login: Thu Nov 18 18:18:57 2021 from 172.16.5.2

[pue@localhost ~]$ exit
logout
Connection to 172.16.5.1 closed.
```

Host client debian: show known hosts

```
pue@debian:~$ ls -la ~/.ssh/known_hosts
-rw-r--r-- 1 pue pue 222 nov 18 18:18 /home/pue/.ssh/known_hosts

pue@debian:~$ cat  ~/.ssh/known_hosts
|1|HvsPSnyjD7I+Olx8lugkZ2/9VEc=|ONSttBnGTx8jRClfv7Io9F/F/64= ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBJk3c+WEBqzjpf7dSSJPxtKltRww60AgFOue
jpgkkEvxQkgXN7wSujZhLxRcdMGaObtnKcfyUfmEF3Jdn69XhNY=

pue@debian:~$ ssh pue@172.16.5.254
The authenticity of host '172.16.5.254 (172.16.5.254)' can't be established.
ECDSA key fingerprint is SHA256:Yh6jrVXFT7Kdpbrnlr5iDC+5INwYdz68c2frYukdA/o.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.5.254' (ECDSA) to the list of known hosts.
pue@172.16.5.254: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).

pue@debian:~$ cat  ~/.ssh/known_hosts
|1|HvsPSnyjD7I+Olx8lugkZ2/9VEc=|ONSttBnGTx8jRClfv7Io9F/F/64= ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBJk3c+WEBqzjpf7dSSJPxtKltRww60AgFOue
jpgkkEvxQkgXN7wSujZhLxRcdMGaObtnKcfyUfmEF3Jdn69XhNY=
|1|BepAOvaQB/MHIDQD3MuckR6opcM=|gl1qq2yHElhlBrgFa49Ffq01K40= ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBMUKj1m6UIEydvbtFwrqQsBTXaRrdRRXI42m
7r/vLDlY0Pteg9UrFfaf4w746uxYzB3SOwMM0TP3eu+mljLcIFM=
```

Host client debian: Configure /etc/hosts to practice man-in-the middle-attack

```
pue@debian:~$ sudo vim /etc/hosts

pue@debian:~$ cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      debian
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.16.5.1 mycentos
172.16.5.2 mydebian
```

Host client debian: ssh to mycentos

```
pue@debian:~$ ssh pue@mycentos
The authenticity of host 'mycentos (172.16.5.1)' can't be established.
ECDSA key fingerprint is SHA256:q3wSAaiFc4or9G9zcDdU7rugOY19vYb91LGmcFBFOuo.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'mycentos' (ECDSA) to the list of known hosts.
pue@mycentos's password:
Activate the web console with: systemctl enable --now cockpit.socket
Last login: Thu Nov 18 18:24:17 2021 from ::1

[pue@localhost ~]$ logout
Connection to mycentos closed.
```

Simulate a server host change

```
pue@debian:~$ cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      debian
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
```

```
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.16.5.254 mycentos
172.16.5.2 mydebian
```

Host client debian: connect to centos (changed fingerprint)

```
pue@debian:~$ ssh pue@mycentos
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@       WARNING: POSSIBLE DNS SPOOFING DETECTED!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
The ECDSA host key for mycentos has changed,
and the key for the corresponding IP address 172.16.5.254
is unchanged. This could either mean that
DNS SPOOFING is happening or the IP address for the host
and its host key have changed at the same time.
Offending key for IP in /home/pue/.ssh/known_hosts:2
  remove with:
  ssh-keygen -f "/home/pue/.ssh/known_hosts" -R "172.16.5.254"
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:Yh6jrVXFT7Kdpbrnlr5iDC+5INwYdz68c2frYukdA/o.
Please contact your system administrator.
Add correct host key in /home/pue/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /home/pue/.ssh/known_hosts:3
  remove with:
  ssh-keygen -f "/home/pue/.ssh/known_hosts" -R "mycentos"
ECDSA host key for mycentos has changed and you have requested strict checking.
Host key verification failed.
```

Host client debian: change the /etc/hosts and suppress the wrong known_hosts entry

```
pue@debian:~$ sudo vim /etc/hosts

pue@debian:~$ cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      debian
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.16.5.1 mycentos
172.16.5.2 mydebian

pue@debian:~$   ssh-keygen -f "/home/pue/.ssh/known_hosts" -R "mycentos"
# Host mycentos found: line 3
/home/pue/.ssh/known_hosts updated.
Original contents retained as /home/pue/.ssh/known_hosts.old
```

Host client debian: reconnect to mycentos server

```
pue@debian:~$ ssh pue@mycentos
The authenticity of host 'mycentos (172.16.5.1)' can't be established.
ECDSA key fingerprint is SHA256:q3wSAaiFc4or9G9zcDdU7rugOY19vYb91LGmcFBFOuo.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'mycentos' (ECDSA) to the list of known hosts.
pue@mycentos's password:
Activate the web console with: systemctl enable --now cockpit.socket
Last login: Thu Nov 18 18:34:18 2021 from 172.16.5.2

[pue@localhost ~]$ exit
logout
Connection to mycentos closed.
```

SSH Server

- /etc/ssh/sshd_config
- /etc/ssh/sshd_config.d
- /etc/ssh/ssh_host_<type>_key
- /etc/ssh/ssh_host_<type>_key.pub
- <type> of keys: RSA, DSA ECDSA and ED25519

When installing an ssh server it generates the hosts keys. These keys identify the host. Some type of keys are: RSA, DSA ECDSA and ED25519. There is a couple of keys for each type:

- private key
- public key (.pub)

```
[pue@mycentos ~]$  ls -l /etc/ssh/
-rw-r--r--. 1 root root     577388 abr 27  2020 moduli
-rw-r--r--. 1 root root       1770 abr 27  2020 ssh_config
drwxr-xr-x. 2 root root         28 abr 27  2020 ssh_config.d
-rw-------. 1 root root       4269 abr 27  2020 sshd_config
-rw-r-----. 1 root ssh_keys    492 sep 29  2020 ssh_host_ecdsa_key
-rw-r--r--. 1 root root        162 sep 29  2020 ssh_host_ecdsa_key.pub
-rw-r-----. 1 root ssh_keys    387 sep 29  2020 ssh_host_ed25519_key
-rw-r--r--. 1 root root         82 sep 29  2020 ssh_host_ed25519_key.pub
-rw-r-----. 1 root ssh_keys   2578 sep 29  2020 ssh_host_rsa_key
-rw-r--r--. 1 root root        554 sep 29  2020 ssh_host_rsa_key.pub
```

The file or directory /etc/ssh/sshd_config or /etc/ssh/sshd_config.d contains the server configuration

```
[pue@mycentos ~]$ sudo head -n 50 /etc/ssh/sshd_config
#       $OpenBSD: sshd_config,v 1.103 2018/04/09 20:41:22 tj Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# This system is following system-wide crypto policy. The changes to
# crypto properties (Ciphers, MACs, ...) will not have any effect here.
# They will be overridden by command-line options passed to the server
# on command line.
```

```
# Please, check manual pages for update-crypto-policies(8) and sshd_config(5).

# Logging
#SyslogFacility AUTH
SyslogFacility AUTHPRIV
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes
...
```

Some common configuration directives:

Port
: Specifies the port which sshd listens to for incoming connections; the default port is 22

ListenAddress
: Specifies the IP address on which the sshd server socket will bind

HostKey
: Specifies where the private host key is stored

KeyRegenerationInterval
: Specifies the time interval in seconds for the server to automatically regenerate its key

ServerKeyBits
: Specifies the number of bits to be used by sshd for RSA key generation

LoginGraceTime
: Specifies the time interval in seconds to wait for the user's response before disconnecting the server

PermitRootLogin
: Specifies if root login over SSH is permitted or not

RSAAuthentication
: Specifies if RSA authentication can be used

PermitEmptyPasswords
: Specifies if user logins to the server with empty password is allowed

PasswordAuthentication
: Specifies if password based authentication must be used

X11Forwarding
: Specifies whether X11 forwarding must be turned on or off. If GUI has been installed on the server, then this option can be enabled

AllowUsers / DenyUsers
: Specifies users who will be allowed access / denied access

AllowGroups / DenyGroups
: Specifies groups who will be allowed access / denied access

Public Key authentication

SSH supports several different authentication methods:

- Public key authentication
- Host-based authentication
- Password authentication

The public key authentication method is the most commonly-used SSH authentication method. It is implemented both on the server as well as the client side. To use this, a public-private key pair must be generated using a key-generation utility.

The algorithm generates keys such that the public and private keys are linked. The private key stored on the client's machine is protected by a passphrase (similar to a password, except it is a series of words which can also be empty).

The system administrator can select either RSA or DSA keys while configuring the SSH public key based authentication. DSA (Digital Signature Algorithm) is a US government standard defined for digital signatures while RSA is named after its creators, Ron Rivest, Adi Shamir and Leonard Adleman.

The ssh-keygen command is used to generate and manage keys used by SSH; it uses the RSA algorithm by default. This program will prompt the user for the location to store the key (~/.ssh is the default) and the passphrase.

- ssh-keygen
- ssh-copy-id

Some of the key options of the ssh-keygen command are:

-b num_bits
: Specifies the number of bits for the key, the range for RSA keys is 768 – 2048 bits (default is 2048 bits) while DSA keys are exactly 1024 bits

-F host_name
: Find the occurrence of the specified hostname in the known_hosts file

-R host_name
: Deletes all keys for the specified hostname from the known_hosts file

-f file_name
: Specifies the file name for the key

host server mycentos

```
[pue@mycentos ~]$ sudo useradd unix01

[pue@mycentos ~]$ sudo useradd unix02

[pue@mycentos ~]$ sudo passwd  unix01
Cambiando la contraseña del usuario unix01.
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña tiene menos de 8 caracteres
```

```
Vuelva a escribir la nueva contraseña:
passwd: todos los tokens de autenticación se actualizaron exitosamente.

[pue@mycentos ~]$ sudo passwd  unix02
Cambiando la contraseña del usuario unix02.
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña tiene menos de 8 caracteres
Vuelva a escribir la nueva contraseña:
passwd: todos los tokens de autenticación se actualizaron exitosamente.
```

Host client debian: create ssh keys for user pue

```
pue@debian:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/pue/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/pue/.ssh/id_rsa.
Your public key has been saved in /home/pue/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:eKpcFRfbGCyiSGKhj+zDci83pIb7TK2ofIS65TwVcUg pue@debian
The key's randomart image is:
+---[RSA 2048]----+
| ...E.   .o      |
|o.. o o . .*     |
|oo . + ...+ .    |
|.o. o  . o       |
|..o  .. S        |
|o. oo  +         |
|o==+. o          |
|=O*=oo           |
|**B++.           |
+----[SHA256]-----+

pue@debian:~$ ls -l ~/.ssh/
-rw------- 1 pue pue 1811 nov 18 19:03 id_rsa
-rw-r--r-- 1 pue pue  392 nov 18 19:03 id_rsa.pub
-rw------- 1 pue pue  666 nov 18 18:39 known_hosts
-rw-r--r-- 1 pue pue  666 nov 18 18:34 known_hosts.old
```

The ssh-copy-id command do:

- copy the public key to the server
- add the contents of the client's ~/.ssh/id_rsa.pub file to the ~/.ssh/authorized_keys file of user unix01 on the server

Host client debian. copy pue public key to ssh server centos, user unix01

```
pue@debian:~$ ssh-copy-id unix01@mycentos
/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
unix01@mycentos's password:
Number of key(s) added: 1
Now try logging into the machine, with:   "ssh 'unix01@mycentos'"
and check to make sure that only the key(s) you wanted were added.
```

Host server centos: check for the authorized_keys

```
[pue@mycentos ~]$ sudo ls -l /home/unix01/.ssh
total 4
-rw------- 1 unix01 unix01 392 nov 18 19:08 authorized_keys

[pue@mycentos ~]$ sudo cat /home/unix01/.ssh/authorized_keys
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDORyydkGoKfo/YpPCPIpZxPBygRORrYRtIhFS97KdbWeZ+CaS8kFRaA21m
```

```
mvniwcT6rCfjUKI2oaZyGkbs7GdKcXlU+wyq4NHRppvqWTJ+brxrvo3bT5/RrXZup/qM3aGEoDjZ3OMh8Nmkaxm/
CWZLi0wRm5B9XAAjtVKWYgOIDGARy3Zbq7TUoyud0uFVwm9nrYPu50/MeeCW3pD09+Dg2f+BR83Sr7baMinm9gaO
kVTR0vUCG4E674baHUr6nG7asd+LP7+L5q5qSYY2AkaiMfs6owQKPF3XJ0bmRPF7CYI3rcgDGPNk0ZP6v3pP8uLz
WmBp0Isv9YpRZFT9EhP1 pue@debian
```

Host client debian: now debian user pu cann connect to user01 in the centos ssh server with PublicKey Authorization

```
pue@debian:~$ ssh unix01@mycentos
Activate the web console with: systemctl enable --now cockpit.socket
Last failed login: Thu Nov 18 19:07:12 CET 2021 from 172.16.5.2 on ssh:notty
There was 1 failed login attempt since the last successful login.

[unix01@mycentos ~]$ exit
logout
Connection to mycentos closed.
```

Host based authentication

The host-based authentication model allows a host to authenticate on behalf of all or some users on that host. [deprecated]

The /etc/ssh/ssh_known_hosts file on the server must hold the public keys of all the hosts that need to be authenticated. The entry in this file implies that the host is trusted by the server and knows its public key.

Example /etc/ssh/ssh_known_hosts

```
122.110.17.32 ssh-rsa
ABFFB3NzaC1yc2EABFFDAQABAAABAQC6XtOSGVEY9PUnMXS6vzvJigeQQtGYwdX2v2zAAsqwYRlaNN/ddV76btf4
PL812r91WYGTgcXT0r0bfSGJ9dmJQ8dPenMAKyviR2BLV1SaIqxqUSjdkXFrlHkC7alILoKrwhMvNWb+Jaa3ecuY
ffKThNadFTHftyntdaVkYxwW7Hr1MknksfZKMPsJjW+Mp3aZVV2wVnQkOgkSsVY8y2pT7h7KuTa66IdqkwO2ZTEX
L2D1X1wIEqGqAJ2VFPQayzclqaGbCzFUYyFsCT1WUL+BzRnehI9L9IVlP3katLSokoBzbxHeu0eb92VXngnrQJ1C
0dA+5O4vp2KxFGEMuwdV
```

Configuration directive in the ssh server

```
HostbasedAuthentication yes
```

SSH Client utilities

The openssh and openssh-clients packages must be installed on the client machine to connect to an OpenSSH server.

- ssh user@host “command”
- scp
- sftp

SSH Agent

If the user's private key is protected by a passphrase, then the passphrase needs to be entered by the user while invoking any ssh program. This can be inconvenient in scripts.

The SSH agent is an application, which is used to cache the decrypted private key and provide it to SSH client programs when required. This effectively means the passphrase has to be entered only once by the user.

Generally, the agent runs after the user logs in and maintains the cached information for the duration of the session.

- eval "$(ssh-agent -s)"
- ssh-add ~/.ssh/user-private-key

The ssh-add command is used to add private keys to the agent's repository. The agent will be running on the user's terminal or desktop and authentication data is not shared with any other system over the network.

The identity files should be readable only to the user, if they can be read by other users then it indicates possible incorrect configuration or some unauthorized access.

```
pue@debian:~$ eval $(ssh-agent -s)
Agent pid 3344

pue@debian:~$ ssh-add ~/.ssh/id_rsa
Identity added: /home/pue/.ssh/id_rsa (pue@debian)

pue@debian:~$ ssh-add -l
2048 SHA256:eKpcFRfbGCyiSGKhj+zDci83pIb7TK2ofIS65TwVcUg pue@debian (RSA)

pue@debian:~$ ssh-agent -k
unset SSH_AUTH_SOCK;
unset SSH_AGENT_PID;
echo Agent pid 3344 killed;
```

Some of the most useful options of the ssh-add command are as follows:

-d id_file	Deletes the identity specified by the file from the agent
-D	Deletes all identities stored by the agent
-x	Locks the ssh-agent with a password This will restrict addition, deletion and listing of identity entries
-X	Unlocks the ssh-agent

```
Practice: create an ssh-key and use it to connect to GIT
```

SSH Tunneling

By default, TCP/IP is not a secure connection stream and is open to network attacks. SSH encapsulates the TCP/IP connections in a secure layer and thus creates a tunnel for communication. The data passing through the tunnel is encrypted as well as verified for integrity.

This feature is called SSH Tunneling or SSH Port Forwarding.

/etc/ssh/sshd_config

```
AllowTcpForwarding yes
```

Port forwarding examples:

```
$ ssh -L 9102:testdbhost:1521 testdbhost

$ ssh -L 8586:localhost:8586 test_user@weblogicserver1
```

reverse tunnel example:

```
$ ssh devuser@dev.netdevgroup1.com -R 8000:192.168.1.12:8000
```

X11 Forwarding

SSH is also capable of forwarding graphical applications over the network. To enable X11 forwarding, the /etc/ssh/sshd_config file must contain the option:

```
$ ssh -X pluto.netdevgroup1.com

$ echo $DISPLAY
localhost:10.0

$ graphical-program
```

Example Exercises

1. Realitza els exercicis indicats a: 110.1 Perform security administration tasks
2. Realitza els exercicis del Question-Topics 110.1