<epam>

SSH Overview

Linux Essentials

TRAINING

CENTER

<epam>

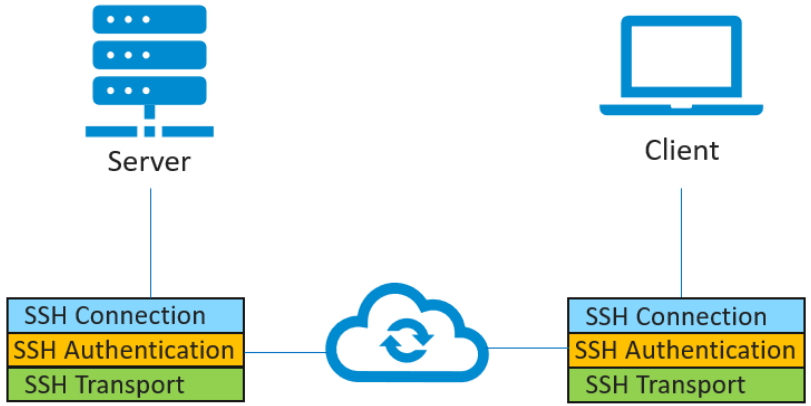
What is SSH

- SSH (Secure Shell) is a cryptographic network protocol for operating network services securely over an unsecured network.
- SSH provides a secure channel for data transmission
- SSH supports secure remote logins, secure remote command execution and secure file transfers
- SSH has a client-server architecture

SSH Architecture

SSH adopts a three-layer architecture:

- SSH Transport Layer Protocol
  - ✓ Initial connection
  - ✓ Sever authentication
  - ✓ Sets up secure channel between client and server
- SSH Authentication Protocol
  - ✓ Client authentication over secure transport layer channel
- SSH Connection Protocol
  - ✓ Supports multiple connections over a single transport layer protocol secure channel
  - ✓ Efficiency (session re-use)



Components of SSH

Component	Description
SSHD Server	A program that allows incoming SSH connections to a machine, handling authentication, authorization
SSH Clients	A program (utility) that connects to SSH servers and makes requests for service
Session	An ongoing connection between a client and server

How SSH Authenticates Users

Authentication method	Description
Password	Traditional way of authenticating with using password. Passwords are encrypted. Less secure.
SSH Keys	SSH keys are a set of cryptographic keys that contains private and public key.
Host-based	Allows users to be logged in from one host (trusted) to another without asking passwords.

SSH Keys Management

- **Generate SSH Keys Using SSH Keygen**

To improve the security of SSH connection, generate a key pair with the keygen utility. The pair comprises a public and private key. The public ley can be shared, while the private key needs to stay secure. When you create SSH key pair, there is no longer a need to enter a password to access a server.
On the host machine's terminal, use this command to create a pair

```
$ ssh-keygen
```

Algorithms for generating key pairs – RSA, DSA, ECDSA. RSA is a default type.
For DSA algorithm, the command is:

```
$ ssh-keygen -t dsa
```

SSH Keys Management

Private Key

```
[user1@centos7 ~]$ cat /home/user1/.ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAu5L3cLpkfO+mvRnULYV91L6OH1GhTtcUBrVyV4eb/dtsMK/n
QQ7P5xOAb1JKRnT7QNk+X23ckH2KZ9BdzMqHzumRMIboRGYhgJP30GABhKjXuZ6M
i5jwIWto0hWV8zO/bErip6Dm7ZAaySAjf9QHtaqMzvAvhvTxkjW2WUqm/Bhc0Is9
y3fepnrivOaVp9xA4zez4lt7/svWwm45QPRrkX5KyZ4WF1qi7SsVXx0pMca/QL1o
Kk0Gi5R2svFNv2Ycf5U3Af55HLy7ydBQH2A1Mzo5Sh40qWyUNkunp0oYF5gy/sT6
7ZwDnEceEvnsquqnJgMJXPhy75sWNH53X0AqSwIDAQABAoIBAQCesUoyzx8xLDfv
CKkgEi5RDR6ElLVaVFCXFG+2af6pmThBQLDvs65lG3Svufko/tOLG8DPT80pWWz9
Q6X4GuknPygc3CYpqBZ0B7vDSVY/e/F3ExJpEG0oeQznoGeKqJ8s6XY0x8ZX+nEO
8+fUv/f9lFrowQAUt6qhPlwV0zjPUgGjjwJWHNprLVofCq2wGrI9GonU5od86i4d
EnKNs6A/uNHblrfD+uELjY7hhWBnn9k6Wzj/16m+CLf2d3+5H/lfCiAaGHxZ1TdK
j/8Wf8GpkCFsTqAJSTCqounB99XNGSuy5AN0vG01+vWsSrUA3qo64/ib6Hj/txro
ZSEnwTExAoGBAOphI8XK7uUOMY0V3S7Anz1OjK2E3y3nVpVf3OW6pZv7Lhyv8d64
9WLR/nv14pO0Z2TICcYHVeZuquVk66N5MuJ5pGaZtI4lSQt7PZ6ERrugrBWPRpk+
f3TIU9XnstmpwP91SY+qGOF+koa+No0l3IHitTHepGou+NzHLykA55wtAoGBAMzg
hS9XlvZc3Ymjpz+IwuiP9toUm4YsbjIbGf6KoO5SVdL5p7HK7BVnuRiD1Nd9wlcX
QEE/D92lgK96COyG3AQ4UymSZzwBKxWK50WCBlTluDAHHddGNJPKZ4hxc+ozFDCJ
H5CRA/gd8do8hYjtirgd2WQ20qGKdMLtOysbtdNXAoGBAMAqNu9DrE0HLx9S7s8X
RTg5v+PAzk9PnlDIkU+VkZTrA2qn4zJOCXslW3Z9KnEZ96/Mxx6jAw30bI+6B+03
QL4pJ96rZTsqXNJaEDJ4l5ncaVsf5q1cbZntabc2yg/mO4P3wQFWYCH56d8p84YU
pc+kdhVWSw14R7iUYxlGUZ9dAoGAH24aQIcJa8YXXnQyxjZu4EqlrhMXQOA0D4/E
zx3pTmZnMDSIm2fsC+Sbqr0eu+hZFg9FFj1ca75mHYV1NQQDYMpbR7dkCvqK54Ul
sa4e1otfVE7jLxFgI4Qbmfz9sWIgqMmAhCcbGsdNQROj0LvQWiqKHSuj6JBbFURb
nWNk44UCgYBfTEaGqLCDoc4n3xRcUX6MQx4zX+i/JMNoOl2oHHo2pNizdJ6OfjSu
vU9vh0Nx6FRlC5t1HTsF0USFfKmg9Mr5Y/NExZN/S1+1G/Iz9HdjtUNos2nnJ7bg
ujM1Z7HfbiDOzcQPq3N1F8uSBmIlDyP2ov/VVmgO9a733xxA9A+ZQ==
-----END RSA PRIVATE KEY-----
[user1@centos7 ~]$
```

SSH-Keygen

```
[user1@centos7 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user1/.ssh/id_rsa):
Created directory '/home/user1/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user1/.ssh/id_rsa.
Your public key has been saved in /home/user1/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:nTohZ8s4SL2iMyEh5ZqbpDWSoel0ep5cbPci2KAt7g4 user1@centos7
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|  .              |
| o               |
|+ .  .   . .     |
|oB  . o S o      |
|Oo++.. B +       |
|E*=o*++.=        |
|==+=++.o..       |
|++=*  . ..       |
+----[SHA256]-----+
```

Public Key

```
[user1@centos7 ~]$ cat /home/user1/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC7kvdwumR876a9GdQthX3Uvo4fUaFO1xQGtXJXh5v922wwr+dBDs/nE4BvUkpGdPtA2T5fbdyQfYpn0F3MyofO6ZEwhuhEZiGAk/fQYAGEqNe5noyLmPAha2jSFZX
zM79sSuKnoObtkBrJICN/1Ae1qozO8C+G9PGSNbZZSqb8GFzQiz3Ld96meuK85pWn3EDjN7PiW3v+y9bCbjlA9GuRfkrJnhYXWqLtKxVfHSkxxr9AvWgqTQaLlHay8U2/Zhx/lTcB/nkcvLvJ0FAfYDUzOjlKHjSpbJ
Q2S6enShgXmDL+xPrtnAOcRx4S+eyq6qcmAwlc+HLvmxY0fndfQCpL user1@centos7
[user1@centos7 ~]$
```

SSH Keys Management

- **Copy Public SSH Key**

  In order to use the key pair for SSH authentication, you'll need to copy key to a server. The key is the file **id_rsa.pub.** To copy your key to a server, run this command

```
$ ssh-copy-id username@remote_host
```

  This will prompt you for the account's password on the remote system. After typing in the password, the content of **id_rsa.pub** key will be appended to the end of user account's home directory /home/username/.ssh/**authorized_key** file

SSH Keys Management

- **Copy Public SSH Key**

  There is a possibility of adding public keys to **authorized_key** file manually without **ssh-copy-id** command.
  It is needed to open **authorized_key** file via text editor (vi/vim/nano) and append required public key.

```
$ vim ~/.ssh/authorized_key
```

  The tilde (**~**) – is a Linux “shortcut” to denote a user’s home directory of current user.

SSH Keys Management

- **Login to a server with SSH key**

```
$ ssh username@remote_host
```

By default SSH is listening on port 22. In case of another SSH port being listened, it is needed to use option **–p** to specify the port

```
$ ssh –p port_number username@remote_host
```

In order to login with identity different from the default (~/.ssh/id_rsa) it is using **–i** option

```
$ ssh –i  ~/keys/my_own_key.rsa username@remote_host
```

SSH Keys Management

- **Copy a file remotely over SSH with SCP**

  You can securely files over the SSH protocol using the **SCP** tool.
  The basic syntax is:

```
$ scp filename user@remote_host:/home/username/destination
```

```
[user1@centos7 ~]$ scp myfile.txt ansible@ecsc00a07ffe.epam.com:/home/ansible/
myfile.txt                                                    100%   16     0.2KB/s   00:00
[user1@centos7 ~]$ ssh ansible@ecsc00a07ffe.epam.com
Last login: Fri Aug  7 13:01:02 2020
[ansible@ecsc00a07ffe ~]$
[ansible@ecsc00a07ffe ~]$ ls -l ~
total 4
-rw-rw-r-- 1 ansible ansible 16 Aug  7 14:44 myfile.txt
[ansible@ecsc00a07ffe ~]$
```

Server-Side SSH Configuration Files

- The system wide configuration files are stored in **/etc/ssh** directory
  - **sshd_config** – configuration file for sshd server daemon
  - **ssh_config** - Client configuration file. It is overridden by configuration files in user's home directory
  - **ssh_host_dsa_key** – The DSA private key used by the sshd daemon
  - **ssh_host_dsa_key.pub** – The DSA public key used by the sshd daemon
  - **ssh_host_rsa_key** – The RSA private key used by the sshd daemon for version 2 of the SSH protocol
  - **ssh_host_rsa_key.pub** – The RSA public key used by the sshd for version 2 of the SSH protocol

Server-Side SSH Configuration Options

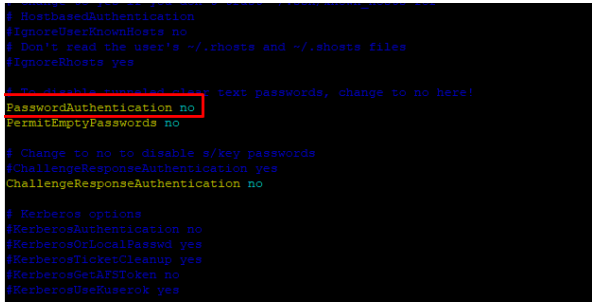
- **Disabling Password Authentication**

  In case SSH keys are configured and working properly, it's recommended to disable password authentication. This will prevent any user from signing in with SSH using a password.

  Open on remote server **/etc/ssh/sshd_config** with root or sudo privileges via **vim** or **nano** text editor.

```
$ sudo vim /etc/ssh/sshd_config
```

Uncomment **PasswordAuthentication** directive, if it is commented out and set it to "**no**".
After performed changes, needed to restart SSH service.




```
$ sudo systemctl restart sshd
```

User Specific Configuration Files

- The user specific configuration files are stored in **/home/username/.ssh** directory
  - **id_rsa** – Contains the RSA private key of user
  - **id_rsa.pub** – The RSA public key of user
  - **id_dsa** – Contains the DSA private key of user
  - **id_dsa.pub** – The DSA public key of user
  - **known_hosts** – This file contains DSA host keys of SSH servers accessed by the user. It is very important for ensuring that the SSH client is connecting to the correct SSH server
  - **authorized_keys** – This file holds a list of authorized public keys for users
  - **config** – User's own configuration file

User Specific Configuration Files

- **SSH Config File (~/.ssh/config) Structure and Patterns**

  The SSH Config file takes the following structure:

```
Host        hostname1
            SSH_OPTION1 value
Host        hostname2
            SSH_OPTION2 value
Host        *
            SSH_OPTION3 value
```

```
Host        dev
            HostName dev.example.com
            User john
            Port
```

For instance, to log in as user named **john** to a host **dev.example.com** on port **2322** you would type:

```
$ ssh john@dev.example.com –p 2322
```

By means **config** file when you type **ssh dev**, the ssh client will read this file and use connection details that are specified for the **dev** host

```
$ ssh dev
```

SSH Clients

- **SSH Client for Linux**
  - **Standard Consol Terminal**
- **SSH Clients for Windows:**
  - **Putty**
  - **Xshell**
  - **WinSCP**