SSH Tunneling!

Πίνακας περιεχομένων

1. Install swarmlab-sec (Home PC)

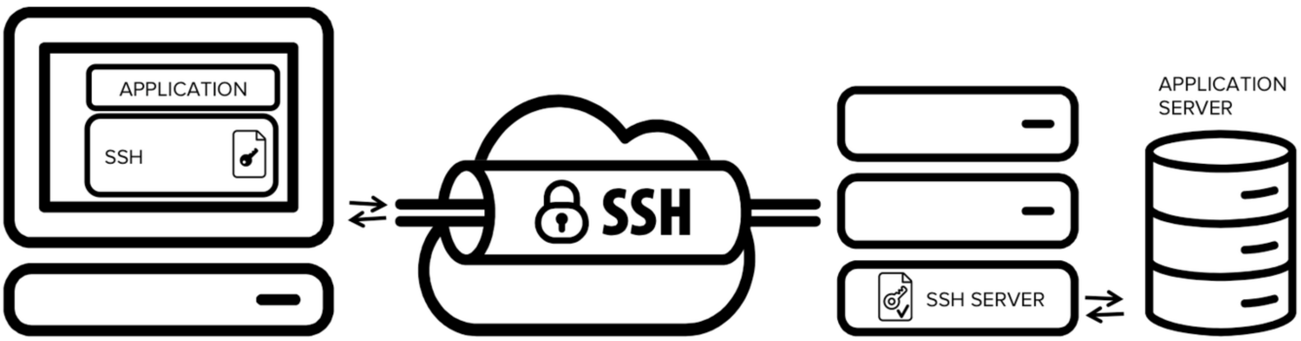
HowTo: See http://docs.swarmlab.io/lab/sec/sec.adoc.html

ℹ *NOTE*

Assuming you’re already logged in

2. SSH Tunneling

SSH Tunneling, is the ability to use ssh to create a bi-directional encrypted network connection between machines over which data can be exchanged, typically TCP/IP.



NOTE

SSH is a standard for secure remote logins and file transfers over untrusted networks. It also provides a way to secure the data traffic of any given application using port forwarding, basically tunneling any TCP/IP port over SSH. This means that the application data traffic is directed to flow inside an encrypted SSH connection so that it cannot be eavesdropped or intercepted while it is in transit. SSH tunneling enables adding network security to legacy applications that do not natively support encryption.

3. Local Port Forwarding

local port forwarding

```
ssh -nNT -L 8000:localhost:3306 user@192.168.89.5
```

The above command sets up an ssh tunnel between your machine and the server, and forwards all traffic from localhost:3306 to localhost:8000 (on your machine).

So now you could connect to MySQL running on your server via localhost on port 8000 on your machine.

4. Remote Port Forwarding

remote port forwarding

```
ssh -nNT -R 4000:localhost:3000 user@192.168.89.5
```

The above command sets up an ssh tunnel between your machine and the server, and forwards all traffic from localhost:3000 (on your machine) to localhost:4000 (in the context of the server).

So now you can connect to the locally running service on port 3000 on the server on port 4000

5. SSH Command

Practically every Linux system includes the ssh command. This command is used to start the SSH client program that enables secure connection to the SSH server on a remote machine. The ssh command is used from logging into the remote machine, transferring files between the two machines, and for executing commands on the remote machine.

5.1. Connect to server

connect

```
ssh  user@192.168.89.5

The authenticity of host '192.168.89.5' cannot be established.
DSA key fingerprint is 04:48:30:31:b0:f3:5a:9b:01:9d:b3:a7:38:e2:b1:0c.
Are you sure you want to continue connecting (yes/no)?
```

Type yes to continue. This will add the server to your list of known hosts (~/.ssh/known_hosts) as seen in the following message:

```
Warning: Permanently added '192.168.89.5' (DSA) to the list of known hosts.
```

Each server has a host key, and the above question related to verifying and saving the host key, so that next time you connect to the server, it can verify that it actually is the same server.

5.2. Executing remote commands on the server

```
ssh user@192.168.89.5 /bin/bash -c "ls -al"
```

6. sshd_config - SSH Server Configuration

The OpenSSH server reads a configuration file when it is started. Usually this file is /etc/ssh/sshd_config, but the location can be changed using the -f command line option when starting sshd.

6.1. Cryptographic policy

- Symmetric algorithms for encrypting the bulk of transferred data are configured using the Ciphers option. A good value is aes128-ctr,aes192-ctr,aes256-ctr.
- Host key algorithms are selected by the HostKeyAlgorithms option. A good value is ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-rsa,ssh-dss.
- Key exchange algorithms are selected by the KexAlgorithms option. recommend ecdh-sha2-

nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group14-sha1,diffie-hellman-group-exchange-sha256.

> not recommend allowing diffie-hellman-group1-sha1, unless needed for compatibility. It uses a 768 bit prime number, which is too small by today's standards and may be breakable by intelligence agencies in real time. Using it could expose connections to man-in-the-middle attacks when faced with such adversaries.

6.2. Verbose logging

It is strongly recommended that LogLevel be set to VERBOSE. This way, the key fingerprint for any SSH key used for login is logged. This information is important for SSH key management, especially in legacy environments.

```
LogLevel VERBOSE
```

6.3. Root login

root access should generally go through a privileged access management system

To disable passwords for root, but still allow key-based access without forced command, use:

```
PermitRootLogin prohibit-password
```

To disable passwords and only allow key-based access with a forced command, use:

```
PermitRootLogin forced-commands-only
```

6.4. Port forwarding

Generally prevent port forwarding on servers, unless expressly needed for tunneling legacy applications. There is substantial risk that users will use SSH tunneling to open backdoors into the organization through the firewall to get access to work machines from home.

6.5. Generate a key pair

```
ssh-keygen
```

Output:

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Created directory '/home/user/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
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:
8c:2a:ed:82:98:6d:12:0a:3a:ba:b2:1c:c0:25:be:5b
```

6.6. Install your public key

```
sh-copy-id -i ~/.ssh/id_rsa.pub UserName@RemoteServer
```

Output:

```
UserName@RemoteServer's password: ********
```

Now try logging into the machine, with "ssh 'username@remoteserver'", and check in:

```
~/.ssh/authorized_keys
```

7. run graphics applications remotely

X11 forwarding needs to be enabled on both the client side and the server side.

- On the client side, the -X (capital X) option to ssh enables X11 forwarding
- On the server side, X11Forwarding yes must specified in /etc/ssh/sshd_config.
- The xauth program must be installed on the server side.

```
ssh -X user@192.168.89.5 gimp
```

8. Copy Files and Directories Between Two Systems

8.1. Copy a file from a local to a remote system

To copy a file from a local to a remote system run the following command:

```
scp file.txt user@192.168.89.5:/remote/directory
```

8.2. Copy a Remote File to a Local System using the scp ommand

To copy a file named file.txt from a remote server with IP 192.168.89.5 run the following command:

```
scp user@192.168.89.5:/remote/file.txt /local/directory
```

ℹ *Reminder*

Caminante, no hay camino,
se hace camino al andar.

Wanderer, there is no path,
the path is made by walking.

Antonio Machado Campos de Castilla