

SSH - Secure Shell

Jakob Krebs und Viktor Reusch

ESE 2020

NERD101 - ESE - ifsr - TU Dresden

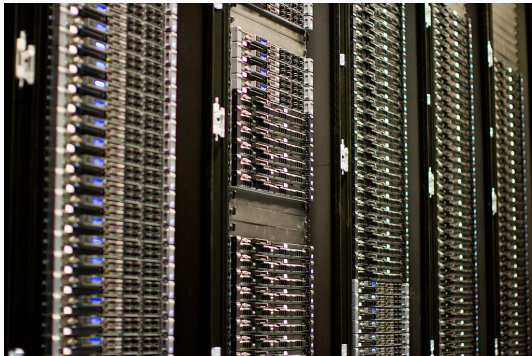
Introduction

Motivation

Servers are versatile.

- Uptime $\Rightarrow$ Build a Website
- Location $\Rightarrow$ Bypass Firewalls
- Compute Power $\Rightarrow$ Run a Simulation
- Connectivity $\Rightarrow$ Host a Game Server

How do you securely control a server remotely?



[?]

SSH solves this problem.

- Encrypted Network Connection
- Replaces insecure Telnet
- Remote Command-Line Interface
- Proxy and Port Forwarding
- File Transfer

Basic Usage

Connecting to a Server

```
me@my_machine:~ $ ssh remote_user@example.com  
remote_user@example.com's password:  
remote_user@example:~ $
```

If the SSH server is not on port 22:

```
me@my_machine:~ $ ssh -p 1234 remote_user@example.com
```

For more options:

```
$ man ssh
```

```
me@my_machine:~ $ ssh remote_user@example.com
The authenticity of host 'example.com' can't be established.
ED25519 key fingerprint is SHA256:R[...]3.
Are you sure you want to continue connecting (yes/no)?
```

Asymmetric cryptography with public and private server keys.

Check fingerprint through a server control panel or with:

```
remote@example:~ $ ssh-keygen -lf /etc/ssh/ssh_host_ed25519_key.pub
256 SHA256:R[...]3 root@example (ED25519)
```

Terminate a Connection

```
remote_user@example:~ $ exit  
logout  
Connection to example closed.  
me@my_machine:~ $
```

Terminate a unresponsive connection.

For more options:

Keyfiles

Keyfiles

Why do you need keyfiles?

- Security
- Laziness

Generate key:

```
$ ssh-keygen -t ed25519 -C me@mypc
```

```
$ ssh-keygen -b 4096 -t rsa
```

Copy public key to server: `~/.ssh/authorized_keys`

```
$ ssh-copy-id my@example.com
```

Now you can log in without password.

Resulting `~/.ssh/id_ed25519.pub`:

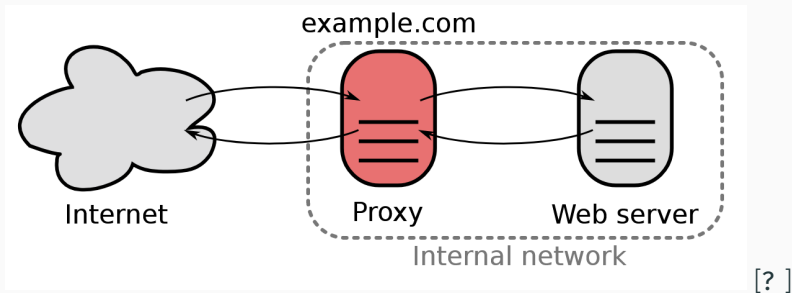
```
ssh-ed25519 AAAAC3NzaC1lZ  
DI1NTE5AAAAIJDzhzcMdEg7Nz  
wwgB0bdOP1pDSKv5fsN4l4q8Y  
ysLLX viktor@nerd101
```

Proxying

ProxyJump

Connect to a not directly accessible server:

```
$ ssh -J proxy.example.org:22 server.example.org
```



SOCKS Proxy

Use a server as a *SOCKS* proxy:

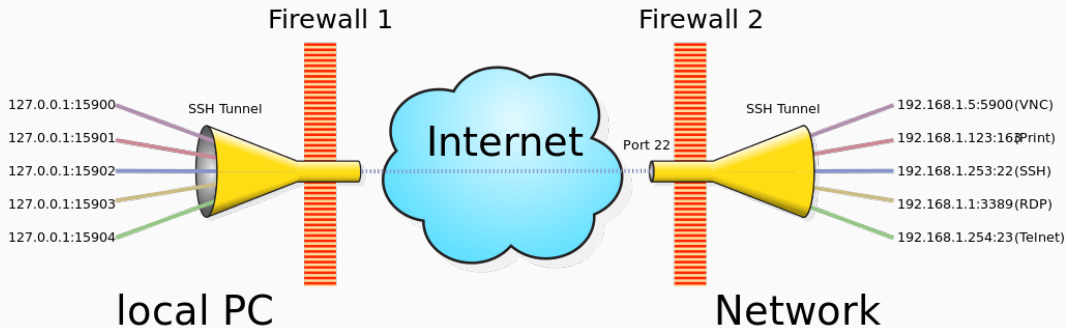
```
$ ssh -D 1080 -N server.example.org
```

`-N` can be used to not start a shell on the server.

Use localhost with port 1080 in the proxy settings of your browser.

Port Forwarding

Visualization SSH Tunneling



[?]

Local Portforwarding

Connect to a remote website (port 80), that is blocked by a firewall:

```
$ ssh -L 8080:localhost:80 server.example.org  
$ ssh -L 8080:server.example.org:80 firewall.example.org
```

Remote Portforwarding

Make your local website accessible by programs on the server.

```
$ ssh -R 80:localhost:8080 server.example.org
```

Config File

Motivation

You are too lazy to type the awful long SSH commands?

```
$ ssh long_user_name@long.nasty.server.name -p 9999
```

Config File

Configure an alias for your connection in the `~/.ssh/config` file:

```
Host myserver
Hostname long.nasty.server.name
User long_user_name
Port 9999
ProxyJump firewall.example.com:22
LocalForward 8201 localhost:8200
```

Usage:

```
$ ssh myserver
```

Applied Options

Show what options are applied on a connection:

```
$ ssh -G switch-bu24.agdsn.network  
user jakob  
hostname switch-bu24.agdsn.network  
port 22  
[...]  
proxyjump login.agdsn.tu-dresden.de
```

```
Host switch-bu24.agdsn.network  
User jakob  
Host *  
IdentitiesOnly yes  
ControlMaster auto  
ControlPersist 60s  
Host login.agdsn.tu-dresden.de  
ProxyCommand none  
Hostname 141.76.119.134  
Host *.agdsn *.agdsn.network  
ProxyJump login.agdsn.tu-dresden.de  
IdentityFile ~/.ssh/id_ed25519
```

Reuse Connections

Multiple connections to a single server is inefficient (e.g. in Robolab).

How can you use a single connection for multiple sessions?

```
Host *  
ControlMaster auto  
ControlPath ~/.ssh/sockets/%r@%h-%p  
ControlPersist 600
```

Copying Files

How to copy a file from or to a server?

```
$ scp /file/on/my/machine remote_user@server:/destination
```

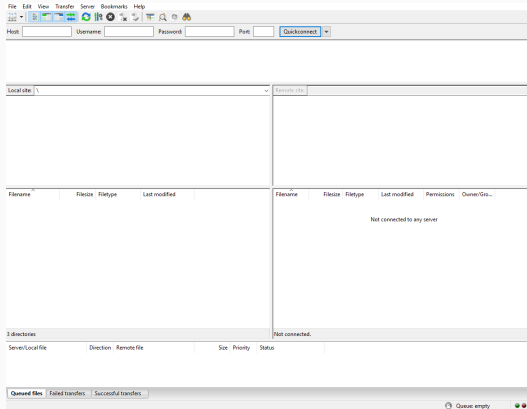
And a directory?

```
$ scp -r /my/directory/ remote_user@server:/destination/
```

This also uses your SSH config.

SFTP

- More elaborate operations (e.g. rename, deletion)
- Opens a session for multiple operations
- Has many graphical user interfaces

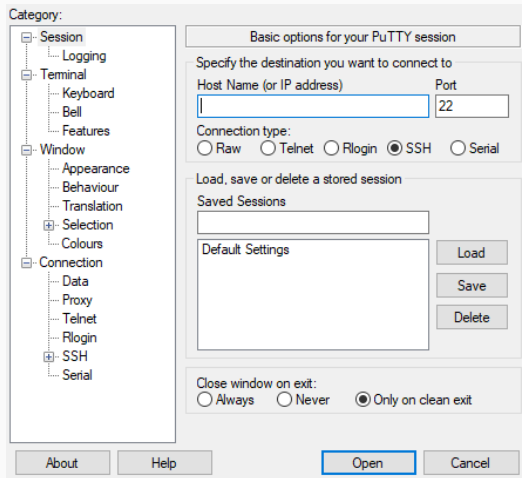


[?]

Windows

Help! I am using Windows.

- Install Linux
- Port of SSH in *Git for Windows*
- Windows Subsystem for Linux
- Cross-platform SSH clients (e.g. *PuTTY*)



[?]

