SSH

Packages: openssh

Client config: /etc/ssh/ssh_config

Server config: /etc/ssh/sshd_config

What is SSH?

Notable config options:

Port # default port(22) is sometimes blocked on networks X11Forwarding # Lets you connect to the X server(forward GUI apps) Banner # Display a message before logging in(warning messages are required in some countries), file /etc/issue.net is usually used for that. Alternatively you can show a message after login, simply edit /etc/motd for that. PasswordAuthentication no # Force use of SSH keys ChallengeResponseAuthentication no # Force use of SSH keys(default set to no?)

sudo systemctl enable --now sshd - Enable sshd service and start it, this is required if you want to host a SSH server so it starts at boot.

By default SSH server accepts user logins(root is disabled by default), but you might want to generate and use SSH keys instead.

Default crypto used is 2048 bit RSA. This is a sane default, you could possibly use 4096 bit RSA(or higher), which has diminishing returns. It takes about 8x more resources to decrypt 4096 RSA than 2048 RSA.

Consider using the newer Ed25519 cipher. Ed25519 is supposedly the best current option. There is no need to set the key size, as all Ed25519 keys are 256 bits. The only problem should be compatibility with old openssh versions.

ssh-keygen *-t ed25519* - Generate a keypair - you'll be prompted for a filepath and a password] to secure the key. The passphrase uses AES-128 for encryption. You probably don't want to use a passphrase though, so just leave it empty.

ssh-copy-id -i ~/SSHkey.pub -p 1234 hostname.org - Copy the public key to the server via SSH. In the example there is specified file path, port and hostname/IP.

By default the public keys allowed to connect to your machine are saved per line in ~/.ssh/authorized_keys

Cool thing that SSH can do is port forwarding:

Let's say I'm running a webserver on 192.168.122.254 - this command would forward the port 80 to port 20123, only for 127.0.0.1, so you could look at the website via http://localhost:20123 from the host machine you executed the ssh command on! You can of course replace the host(127.0.0.1) with whatever and forward your traffic through just for that website.

```
ssh -L :20123:127.0.0.1:80 username@192.168.122.254
```

This is remote mapping instead - executing this would forward the host's port 22 to the remote server's port 20123 - useful if ISP is blocking ports and you want to forward something through another server!

```
ssh -R :20123:127.0.0.1:22 username@rys.pw
```

Revision #2
Created 28 June 2021 18:14:08 by C0rn3j
Updated 29 June 2021 11:27:05 by C0rn3j