Learning SSH 2 Cheat Sheet

by Anthony.Dominguez via cheatography.com/202519/cs/43019/

Installing OpenSSH

Command	Options	Arguments	Arguments
sudo apt		update	Check for updated package definitions
sudo apt install		opens--server	Install OpenSSH server
sudo systemctl		status sshd	Information about SSH configuration
sudo ufw		status	Check firewall status and rules
sudo ufw		allow ssh	Allows SSH traffic through the firewall
sudo ufw		enable	Enables firewall
sudo nano		/etc/ssh/ssh-d_config	Allows editing the SSH server configuration
sudo nano		/etc/ssh/ssh-_config	Allows editing the SSH client configuration

By default, SSH runs in port 22 using TCP protocol
SSH server configuration is stored in /etc/ssh/sshd_config.
SSH client configuration is stored in /etc/ssh/ssh_config.

Managing Users and Access

Command	Argument	Description
AllowUser	leigh vishal stefan	Only allow these users to log in
DenyUser	bob mary paulina	Allow all users except these
AllowGroup	admins developers	Only allow users in these groups to log in
DenyGroup	sales marketing	Allow all users except those in this groups

Managing Users and Access (cont)

Match Address	10.0.1.0/24	To be able to connect remotely only from 10.0.1.0/24 addresses
Match User	alice bob	Alice and bob are able to connect remotely according to subsequent parameters

All the information above is available in the file /etc/ssh/sshd_config
Precedence matters when defining access

Connecting to a server using a password

Command	Arguments	Description
ssh	*username@ip address*	Connect to a remote SSH server

The first time connection is established to a remote SSH server, a host fingerprint is indicated in the screen, and if the fingerprint is accepted, the local device saves the fingerprint together with information about the connection into a folder into the file /home/ssh/known_hosts

Creating a key pair with ssh-keygen

Command	**Description
ssh-keygen	Generate public/private rsa key pair.
ssh-keyscan *server ip address*	Displays keys to share depending on the encryption algorithm to be used.

It is recommended to generate a key pair for only one purpose (one user/one server). It is also recommended to save the keys in separate folders in the /home/*user*/.ssh/ directory. A passphrase can also be added as an extra layer of security for the key pair.

Managing and using key pairs

Command	Description
ssh-copy-id -i ~/.ssh/*mykey*.pub *user@server*	Add key to ~/.ssh/authorized_keys if access to ssh server already exists
Add key to ~/.ssh/autho-rized_keys out of band	Add key to ~/.ssh/authorized_keys if access to ssh server does not exists

When changing the configuration of ~/etc/ssh/sshd_config with nano, remember to restart the service for the new settings to apply, with **sudo systemctl restart** sshd.
When having a lot of keys, we can speed up the connection process by specifying which key we want to use to connect to the server, like: **ssh** *user@server* -i ~/.ssh/*key directory*

Client Configuration Options

Host *name*

Hostname *ip address*

Port *port number*

User *username*

IdentityFile ~/.ssh/*key name*

For information about precedence's and priorities, consult **man** ssh_config. SSH obtains configuration data from the following sources in the following order:
1. command-line options
2. user's configuration file (~/.ssh/config)
3. system-wide configuration file (/etc/ssh/ssh_config)

It is also good practice to change ~/.ssh/config to be only read and write by the user with **chmod** 600 ~/.ssh/config

Transferring Files with SFTP

Command	Option(s)	Argument(s)	Description
sftp		*user@ip address*	Initiate SFTP connection with remote server.
bye			Terminates SFTP connection to remote server.
help			Shows a list of available commands while in SFTP mode, including commands to change working directories.
put		*file name*	Sends a file from the local working directory to the remote local directory.

Transferring Files with SCP

Command	Option(s)	Argument(s)	Description
scp		*local file name user@ip address*:	Copies a file from the local working directory to the remote working directory.
scp		*user@ip address:remote file name local file name*	Copies a file from the remote working directory to the local working directory.

The colon represents the remote user home directory, and both relative and absolute paths can be used to refer to a different directory than the home directory.

Learning SSH 2 Cheat Sheet

by Anthony.Dominguez via cheatography.com/202519/cs/43019/

Multi-Step SSH Connections

Command	Option(s)	Argument(s)	Description
ssh	-J	*user@server1,user@server2 user@server3*	Enable multi-step SSH connection by providing the credentials to all intermediate and the final server to be accessed, without manually establishing all connections separately.

```
Host myserver
        Hostname ip address
        Port port number
        User username
        IdentityFile ~/.ssh/key name

Host server2
        Hostname ip address
        ProxyJump user@ip address of myserver
```

Port Forwarding with SSH

Command	Option(s)	Argument(s)	Description
ssh	-L	*[bind_addr:]port:host:port user@ip address*	Local port forwarding.
ssh	-R	*[bind_addr:]port:host:port*	Remote port forwarding.
ssh	-D	*[bind_addr:]port*	Dynamic port forwarding.
	-f		Fork the SSH process into the background
	-n		Don't read from STDIN.
	-N		Don't run remote commands.
	-T		Don't allocate a TTY
ps x | grep		ssh	Find processes owned by the user, including those without a controlling terminal
kill		*process port*	Ends the process that belongs to a process port.

Port forwarding can also be configured in the client file ~/.ssh/config

```
...
Host server1
\tHostname 10.0.1.110
\t# Access remote port 3306 through local port 3333
\tLocalForward 3333:localhost:3306
\t# Access local port 22 through remote port 5432
\tRemoteForward 22:localhost:5432
\t# Starts a SOCKS proxy on local port 3000
\tDynamicForward 3000
```

Cheatography

Learning SSH 2 Cheat Sheet

by Anthony.Dominguez via cheatography.com/202519/cs/43019/

Troubleshooting SSH

Command	Option(s)	Argument(s)	Description
systemctl	status	sshd	Check the status of the SSH service.
systemctl	restart	sshd	Restarts the SSH service.
journalctl	-u	ssh	See the log for SSH services, to look at problems.
sudo ufw		status	Looks at the rules set for the firewall.
sudo cat		/etc/shadow/	Looks at the shadow file.
| grep		*username*	Pipes the search with grep to look for the username, if there is an exclamation mark at the beginning of the password field, that means the user is locked.
sudo usermod	-U	*username*	Unlocks the locked user account.

Securing a SSH Server

1) Don't allow the root user to log in	**PermitRootLogin** no (or **prohibit-password**)
2) Prevent password logins, and allow keys	**PasswordAuthentication** no **PubKeyAuthentication** yes
3) Change the service port	**Port** *port number*
4) Change the encryption ciphers the server allows	**Ciphers** ... (see **man** sshd_config)

Securing a SSH Server (cont)

5) Enact user control	**AllowUser** **DenyUser** **AllowGroup** **DenyGroup**
6) Consider using software like Fail2ban to help prevent repeated malicious login attempts	
7) Consider designing your system to use a bastion host	
8) Consider putting your SSH server or bastion host behind a VPN	

Tools That Use SSH (Mosh, Mobile Shell)

Command	Option(s)	Argument(s)	Description
sudo apt install		mosh	Install Mosh (needed in both the client and the server)
sudo ufw	allow	60001/udp	Opens ports for Mosh (in the 60,000 range, only needed in the server).
mosh		*user@ip address*	Starts a Mosh session, just like a SSH connection.

Mosh provides a fault-tolerant shell experience. Mosh has to be configured in both the client and the server.

By **Anthony.Dominguez**
cheatography.com/anthony-dominguez/

Not published yet.
Last updated 22nd April, 2024.