U02

Ramin Dadgar

Praktiska delen

Ubuntu 20.04 server setup instructions

user information:

Keyboard-layout: Swedish Username: user Server-name: Server Server prompt: $

Guides followed:

- SSH-keys: https://www.digitalocean.com/community/tutorials/how-to-set-up-ssh-keys-on-ubuntu-1804
- Dropbear: https://linuxconfig.org/how-to-install-and-configure-dropbear-on-linux
- Firewall: https://www.digitalocean.com/community/tutorials/ufw-essentials-common-firewall-rules-and-commands
- Remove SSH-server: https://www.linuxnix.com/remove-ssh-server-linux/

Start the setup with update and upgrade: Login to the Ubuntu 20.04 LTS server on VirtualBox. Make yourself root

User@server:~$ sudo su [sudo] password for user: root@altai:/home/user#

- Check availabe updates with 'apt update'

```
root@server:/home/user# apt update
 ...
Fetched 6,459 kB in 2s (3,842 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
47 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

root@server:/home/user#

- Upgrade to the available updates with 'apt upgrade' root@server:/home/user# apt upgrade ...
- After this operation, 378 MB of additional disk space will be used. Do you want to continue? [Y/n] y ... done root@server:/home/user#

Setup SSH-key:

- Start by generating a key pair on the client machine П ssh-keygen
- The following output appeared, I pressed enter and saved the key at the defualt path Generating public/private rsa key pair. Enter file in which to save the key (/c/Users/Ramin_kd/.ssh/id_rsa):
- As I already got an existing key on the client machine, this appeared next /c/Users/Hp/.ssh/id_rsa already exists. Overwrite (y/n)? y

```
    I entered 'y' to overwrite the existing key.
```

Next you will be asked to enter a passphrase, I left it empty to not use a passphrase Enter passphrase (empty for no passphrase): Enter same passphrase again:

- By now the key should be generated and saved with this output

```
Your identification has been saved in /c/Users/Ramin_kd/.ssh/id_rsa
Your public key has been saved in /c/Users/Ramin_kd/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:C6L9SBPLveYcOxIwmoy8AbJx5XQ0XpAeskJnUBGuwJU Ramin_kd@Ramin
The key's randomart image is:
+---[RSA 3072]----+
|  .+=++o.        |
|. oE*.+o         |
|.o *.=..         |
|+.=.o .          |
|**.+o . S        |
|=+ +.= . .       |
|  + *.o .        |
| . ..=o+         |
|    .+*.         |
+----[SHA256]-----+
```

- Next, by using 'ssh-copy-id username@remote_host' I copied the public key from the client machine in to the server

```
    $ ssh-copy-id user@192.168.10.243
```

- With the following output, I entered the password for my user at the server to add the new key

```
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out
any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now
it is to install the new keys
user@192.168.10.243's password:
```

- I got this output to confirm that the key was added to the server

```
Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'user@192.168.10.243'"
and check to make sure that only the key(s) you wanted were added.
```

- Final step was to check if I could connect to the server from the local client without typing in a password

```
Π ssh ninja@192.168.10.152
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-90-generic x86_64)
...
```

```
system load: 0.17
Usage of /: 61.09% of 8.79GB
Memory usage: 16%
Swap usage: 0%
Processes: 123
Users logged in:1
IPv4 address for enpOs3:192.168.10.243
user@server:~$
```

-- Install and configurate dropbear

Installing dropbear:

- First step is to use apt to install dropbear

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
dropbear is already the newest version (2019.78-2build1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@server:/home/user#
```

Configurating dropbear:

- I used nano as an editor to configurate dropbear

```
root@server:/home/user# nano /etc/default/dropbear
```

- Inside the file I made the following changes (modified NO_START and DROPBEAR_PORT):

```
# disabled because OpenSSH is installed
# change to NO_START=0 to enable Dropbear
NO_START=0 #was 1
# the TCP port that Dropbear listens on
DROPBEAR_PORT=30399 #was 22
```

- To activate the changes I restarted dropbear

```
root@server:/home/user# systemctl restart dropbear
```

- From a new local shell I made sure that dropbear is working and that I can connect to the server with my new custom port

```
Ramin_kd@Ramin MINGW64 ~
$ ssh -p 30399 user@192.168.10.243
The authenticity of host '[192.168.10.243]:30399 ([192.168.10.243]:30399)'
can't be established.
ECDSA key fingerprint is SHA256:hUDywwWFj2uxuLtLIT0J/CWPr5FXCUQwTps5Efn/ynk.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.10.243]:30399' (ECDSA) to the list of
known hosts.
user@server:~$
```

- Install ufw(firewall)

  As root I installed ufw on the server

```
root@server:/home/user# apt install ufw
Reading package lists... Done
Building dependency tree
Reading state information... Done
ufw is already the newest version (0.36-6ubuntu1).
ufw set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@server:/home/user#
```

- As ufw was already installed on the server, I checked its status

```
root@server:/home/user# ufw status verbose
Status: inactive
root@server:/home/user#
```

- I allow port 30399, it's the port I configured on dropbear

```
root@altai:/home/user# ufw allow 30399/tcp
Rules updated
Rules updated (v6)
root@server:/home/user#
```

- Next I enabled ufw

```
root@server:/home/user# ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
root@server:/home/user#
```

- And checked the ufw status to make sure it is active

```
root@server:/home/user# ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip
To                         Action      From
```

30399/tcp ALLOW IN Anywhere 30399/tcp (v6) ALLOW IN Anywhere (v6)
root@server:/home/user#

```
- Stop and remove built-in SSH service
 As sudo I stopped the built-in SSH service
```sh
root@server:/home/user# systemctl stop ssh
root@server:/home/user#
```

- With apt I removed the openssh-server
```

```
root@server:/home/user# apt remove openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
libwrap0 openssh-sftp-server ssh-import-id
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
openssh-server
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 1,527 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 114198 files and directories currently installed.)
Removing openssh-server (1:8.2p1-4ubuntu0.3) ...
Processing triggers for man-db (2.9.1-1) ...
root@server:/home/user#
```

- Reboot the server to make sure that everything works after a restart

```
root@server:/home/user# reboot
root@server:/home/user#
```

- After the reboot, login to the server to make sure everythings works

```
Ramin_kd@Ramin MINGW64 ~
$ ssh -p 30399 user@192.168.10.152
ssh: connect to host 192.168.10.152 port 30399: Connection timed out
```