Server System Management - Linux

Lab 05 part 1: SSH and network services

# 

# Lab 05 (part 1): SSH and network services

1. Log in on your vm as your regular user, and sudo adduser week5 to create a new user week5. Do not put the user in the sudo group this time.

Pick a password that you can easily remember.

sudo adduser week5

2. Now login as week5. Prepare it to for ssh access by setting up a special folder and creating a file in it.

*mkdir .ssh*

*touch .ssh/authorized\_keys*

*chmod 700 .ssh*

*chmod 600 .ssh/authorized\_keys*

You need these files AND permissions or it won’t work later on.

3. ON YOUR WINDOWS MACHINE, start a command shell (cmd). Type ‘cd’ to go to your windows’ home directory. use the ssh-keygen tool to generate your own private/public keypair. Accept the default location.

**Note**

A passphrase will be asked to encrypt your private key. It is good practice to enter a strong passphrase here. If your user password is strong, it is acceptable to use the same password.

For this exercise, DO NOT ENTER A PASSPHRASE. You must now be extra careful that your private key remains private, and is never disclosed (eg by accidentally setting your permissions too permissive), because anyone who finds it can use it and claim your identity on other machines where you installed this key (like we will do now).

A screenshot of a computer

Description automatically generated

4. ssh-keygen exists on linux Debian as well ! Use a command from the package management lab to find out which package this command belongs to. Which package was it ? (QUIZ)

dpkg-query -S ssh-keygen

A screen shot of a computer

Description automatically generated

5. If it did not exist yet, a folder .ssh has just been created in your WINDOWS home directory: C:\users\your user/.ssh. Let's go there and investigate its contents using dir

A screenshot of a computer

Description automatically generated

You should find the keypair that was just created:

• id\_rsa.pub is your public key; display it using cat id\_rsa.pub

• id\_rsa is your private key; display it using cat id\_rsa

A screen shot of a computer screen

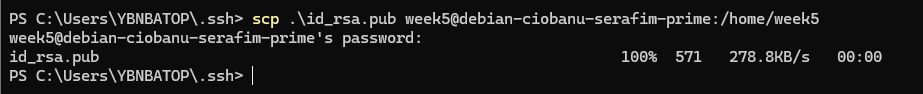
Description automatically generated

Cat does not work in cmd, so I used the PowerShell (because its better)

6. Now copy your public key from your local machine to your vm’s week5 home directory using the windows built-in scp command. In order to connect to your vm, you’ll need to discover your vm’s IP address using ip address show, which we will discuss later.

Which command did you use (in your WINDOWS command prompt) ?

scp .\id\_rsa.pub week5@debian-ciobanu-serafim-prime:/home/week5



This will be the last time your need to type your password to connect to your machine . It's only necessary once, as after the next stip, it will automatically authenticate you without need for a password.

7. Almost there. Go back to your Debian console (where you are still logged in as week5). Now append the content of id\_rsa.pub to your authorized\_keys file in de .ssh directory. Use cat for this. How did you do that ?

cat id\_rsa.pub >> .ssh/authorized\_keys

8. Now try to log in as week5 using ssh from the windows command line using ssh week5@your\_debian\_vm\_ip

This principle is called key-based authentication, which is more secure because the key is much stronger than your password could be. Of course, you need to protect this private key with your life !

A computer screen with white text

Description automatically generated

9. Read /etc/ssh/sshd\_config using nano or less, as well as man sshd\_config, and use grep and man commands to show the answers to the following questions: (QUIZ)

• On which port does the ssh server listen on BY DEFAULT ?

grep -i "Port" /etc/ssh/sshd\_config (22)

A black screen with white text

Description automatically generated

• Which is the directive that specifies the log facility where ssh activity on your server will be logged?

grep -i "^SyslogFacility" /etc/ssh/sshd\_config (AUTH)

• Which option determines whether a message should be shown when connecting via SSH BEFORE succesfullogin?

grep -i "^Banner" /etc/ssh/sshd\_config (none)

• Which option determines whether the root user can log in using ssh?

grep -i "PermitRootLogin" /etc/ssh/sshd\_config (PermitRootLogin)

grep -i "^PermitRootLogin" /etc/ssh/sshd\_config (PermitRootLogin)

If you set this option to without-password, root will be able to login, but only if you have set up a public key in its authorized\_keys file

• Which option should you set to no to only permit logins using public key and no longer accept password login?

grep -i "^PasswordAuthentication" /etc/ssh/sshd\_config (PasswordAuthentication)

grep -i "PasswordAuthentication" /etc/ssh/sshd\_config (PasswordAuthentication)

10. Figure out where your Debian host’s keys are stored.

/etc/hosts

11. Display the contents of your host’s pub rsa key using cat. \*upload screenshot\*

cat /etc/ssh/ssh\_host\_rsa\_key.pub

A screen shot of a computer screen

Description automatically generated

When you connect to a server using ssh, you identify yourself to the server using your private key. The server, being polite, also identifies itself to YOU by offering it’s public key.

12. Check the log files. In question 9, you have probably discovered that the default log facility is AUTH.

Since Debian 12 the default logging service changed from syslog to journald. This service centralizes all logging and makes it available through the journalctl command. To view everything try executing journalctl without any parameters.

13. You can use filters to extract only the information needed. Try to limit the output to only the SSH-service

journalctl \_COMM=sshd

A screenshot of a computer program

Description automatically generated

14. Let’s explore journalctl some more as it is our go to for troubleshooting.

Use man journalctl to accomplish the following:

• ask for an overview of recent logs

journalctl -f

A screen shot of a computer

Description automatically generated

• get the most recent log messages and keep following them live

**Note**

• format the output as JSON for convenient parsing from scripts (for example node, which is javascript)

journalctl -o json

A computer screen with green text

Description automatically generated

• show the kernel messages (also try dmesg which directly asks this info from the kernel)

journalctl -k

• show all logs since yesterday

journalctl –since yesterday

A screenshot of a computer screen

Description automatically generated

• show just all logs generated by the ssh unit

journalctl -u ssh

15. You can even ssh from your own vm to your own vm as another user : try this by setting it up so that user week5 can connect to your Debian as your regular user (familynamefirstname) without needing a password.

HINT : start by using keygen on your Debian as user week5. Then enjoy the power of ssh-copy-id (which DOESN’T exist in windows) to make such a setup ridiculously easy.

ssh-keygen

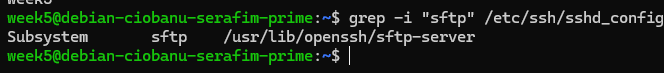
ssh-copy-id ciobanuserafim@debian-ciobanu-serafim-prime

16. Confirm sftp is enabled in your /etc/ssh/sshd\_config file.

grep -i "sftp" /etc/ssh/sshd\_config

What line did you find showing this ?

Subsystem sftp /usr/lib/openssh/sftp-server



17. Be sure you are logged into your debian vm as user week5 and your current working directory is week5’s home directory (cd to it if needed). Now sftp connect using your regular user( sftp regularuser@localhost.)

sftp ciobanuserafim@localhost

A screen shot of a computer

Description automatically generated

sftp works similar to the command-line ftp command you learned about last year

18. Use the put command to upload /usr/share/dict/words to the sftp-connected regular user’s home directory, to a file named wordlist1.txt.

put /usr/share/dict/words wordlist1.txt

A black screen with white text

Description automatically generated

19. Now (while staying in the same sftp session) use the get command to download that wordlist1.txt as wordlist2.txt. Do not provide a specific download location using the get command Where did both wordlists end up finally ? (QUIZ)

get wordlist1.txt wordlist2.txt

It should end up in the home directory where we are located by default

20. Let's also set up an old-fashioned vsftp server. Make sure you do this as root: regular users cannot install system software.

I am still working with user week5 and hence I had to append the user to the sudo group.

apt update (ALWAYS APT UPDATE BEFORE YOU INSTALL SMTH)

apt install vsftpd

apt install ftp

25. Now go back to week5 user. Test ftp from your Debian VM as week5 to your Debian VM as your regularuser

ftp ciobanuserafim@localhost

A computer screen with white text

Description automatically generated

cd /tmp

ftp localhost

Log in using your user regularuser and its password.

We first went to /tmp so we can temporarily download files from our own homedirectory there, to avoid confusion.

**Note**

You cannot use the ssh key to avoid typing the password here! That's because ftp has nothing to do with ssh - whereas sftp is a subsystem for ssh and can also take advantage of ssh's key management.

Confirm you can download wordlist1.txt which was put there in step 18.

get wordlist1.txt

Log out again and check if the file was downloaded correctly:

ls -l /tmp/wordlist1.txt

A screenshot of a computer

Description automatically generated

26. Now make life easier for yourself by setting it up so your public key on your windows machine also allows you to connect without password as your regular user on your Debian vm for future labs. Don’t you miss your ssh-copy-id ? 😊

scp .\id\_rsa.pub ciobanuserafim@debian-ciobanu-serafim-prime:/home/ciobanuserafim

cat id\_rsa.pub >> .ssh/authorized\_keys

A screen shot of a computer screen

Description automatically generated

I do not remember the command of ssh-copy-id but whatever, it should be fine to find a way to do iit like that too.

27. Another traditional simple and reliable file sharing service used on UNIX servers since the early days is NFS.

It has been superseded a bit by iSCSI, but it is still gaining popularity again for virtualization due to its simplicity. Read this article:

http://www.infoworld.com/article/2612735/data-center/old-nfs-is-the-new-darling-in-virtualization.html

28. Let's set up NFS.

Use apt to install the package nfs-kernel-server.

Use dpkg -L nfs-kernel-server | grep /etc to get an idea of the config files that we may need to modify.

A screen shot of a computer

Description automatically generated

29. Use your editor to edit /etc/exports. This file also has a man page which you can consult.

We will use the most often used NFSv3. Use the examples and/or man page to export /usr/share/doc to the entire 192.168.0.0/16 range read only.

A screenshot of a computer program

Description automatically generated

Apply the config change using this command:

exportfs -r

A screen shot of a computer program

Description automatically generated

30. You can now mount your exposed share as if it was a remote network location on your /mnt.

Which mount command did you use ?

sudo mount -t nfs <ip\_of\_vm>:/usr/share/doc /mnt

sudo mount -t nfs 192.168.206.133:/usr/share/doc /mnt

specifying the localhost as a IP address would fail! and give errors!

31. Use find /mnt and redirect it to a file /tmp/list1. See ? You can now use that mount as if it were a local resource.

find /mnt >> /tmp/list1

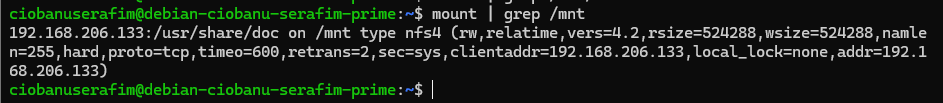
32. Use mount without parameters to confirm this (network-mounted) filesystem is indeed in the list of mounted filesystems for your server.

mount

A screenshot of a computer screen

Description automatically generated

mount | grep /mnt



Which is the filesystem type listed by the /usr/share/doc mount on /mnt? (QUIZ)

type nfs4

33. Unmount /mnt again:

sudo umount /mnt

34. Create a new folder /public which should get all permissions for **everyone**, and a sticky bit. Then export that folder with nfs.

sudo mkdir /public

sudo chmod 1777 /public

sudo nano /etc/exports

/public 192.168.0.0/16(rw,no\_root\_squash)

sudo exportfs -r

A screen shot of a computer

Description automatically generated

Use the extra no\_root\_squash option in /etc/exports to allow the root user to write, which is normally disabled because root is normally converted to user nobody over NFS for security.

Test if you can mount it on your own system under /mnt. Try to create a new file, both as user root and as user week5. For example:

date > /mnt/file1 (as root)

echo hello > /mnt/file2 (other user)

ls -R /usr > /mnt/file3 (other user)

How many files does every user own in /mnt now ? (QUIZ)

A screen shot of a computer program

Description automatically generated

Everybody owns 1 file each, but also can be considered 3, because they can reach each others files.

35. Windows crossover

You can even access the exported /public folder of your Linux VM on your Windows Machine (or VM from Windows Labs). (You MAY need to install/activate the Services for NFS windows feature). Try it and show those three files from before showing up in a windows file explorer (SCREENSHOT)

A screenshot of a computer

Description automatically generated

to access it, you just go to the path \\<ip>\<share>. In this case it would be \\192.168.206.133\public

A screenshot of a computer program

Description automatically generated