FINAL REPORT - CYBERSECURITY AND NETWORKS

Group Project

BPROG

Group 13: Stardrop

Sigrun Andersen Høgstedt

Martyna Maria Juga

Inga Kvam

Janne Wenger

NTNU GJØVIK, **17/11/2021**

Table of Contents

1. Introduction

This project is all about exploring and gaining experience around cybersecurity and networks, by making a server and securing it. The goal of this project is to make a small secure server that fits our needs, that we can later make more complex as we gain a better understanding of this topic.

The project consists of the server we ended up making, as well as this report.
We will evaluate our project by testing its security and backup, as well as by reflecting over how well we met our goal, and what we could have done better.
After this project is over, we hope to have gained a better understanding of servers, as well as having done good preparation for our exam.

2. Background

These are the tools and commands we used in our project:

<u>Openstack</u>
A cloud computing software used to manage virtual machines.

<u>Secure Shell (SSH)</u>
A network protocol used for operating securely on an unsecure network.

<u>Secure File Transfer Protocol (SFTP)</u>
A more secure alternative to FTP. Used for transferring files securely over a network.

<u>Ubuntu Linux</u>
We used Ubuntu Linux as our main operating system. since this is well suited for beginners, and easy to use[1].

[1] T. Haddon «An introduction to Ubuntu»

Kali Linux

We used Kali Linux as the operating system on our penetration testing VM, since this operating system is geared towards information security tasks[2].

These are some of the tools that we used[3]:

- WinSCP

○ WinSCP is an open source SFTP client. It was a tool that was super useful, especially when we were making files for our main server. It gives a visualization of the server files which makes file management much easier.

- PuTTY

○ We used this tool solely to be able to connect to the VM's using WinSCP. WinSCP required a file generated by PuTTY in .ppk format instead of a pem file in order to establish a ssh connection with WinSCP[4].

3. Project Design

3.1 Objectives

In this project we had 4 virtual machines and 8GB RAM at our disposal. We decided we wanted to make a simple file sharing server that we could use to save our own important files.

We set up our server so that one of our virtual machines acted as a server, and another one as a backup to ensure that the data is not lost in the case of a malfunction on the main server. The remaining two VMs were to act as a client and as a penetration testing server.

[2] G0g0tmi1k «What is Kali Linux»
[3] D. Wayne "What is WinSCP?"
[4] University of Sussex «PuTTY»

Our server uses Ubuntu Server as the main operating system. The main server uses SFTP protocol based on SSH. This means that it uses one connection and encrypts both authentication information and the data files being transferred. We chose this because of the security and accessibility it provides. The Ubuntu server is used as file storage and transfer.

We also created an attack server in order to check the security of our main server. The goal was to have the server be secure enough to not let the penetration test through. For this task, we chose Kali Linux as the operating system. The reason for this is that Kali Linux has pre-built tools that help with penetration testing and other security needs.

3.2 Requirement

We want to make our server so it can support simple file sharing and storage. For that we need it to be stable and reliable, so it doesn't crash. It needs to have good security so no one else has access to it. And it needs to have a good backup, that we can manually save. Other than that we also need to set up usernames and passwords to log in to the server, this makes it even more secure. Even if people were to use our own computers, they will not have access without these.

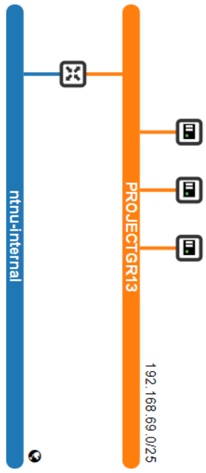
Something we consciously do not dedicate much effort to, is capacity. Both in how many can be on it at a time, and also in the sizes of the files. The reason for this being, that we only are 4 people, the chances that multiple people are online at a time is small. And the storage does not need to be big, since we delete files when they are no longer relevant.

3.3 Network Topology and Setting

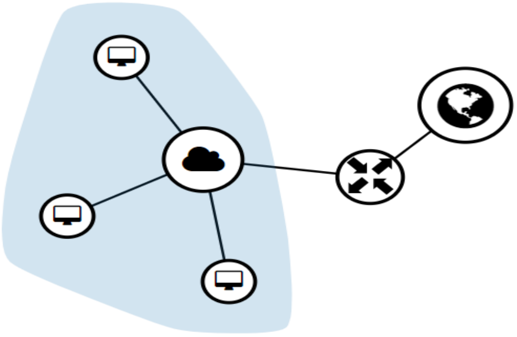
We have three servers connected to our network. There is one main server, one backup server, and one penetration test server. The main and backup servers have an identical setup. They use Ubuntu (20.04 LTS Focal Fossa) as their operating system, because it is a reliable Linux system, but at the same time has a small enough size, so that our quota is not exceeded[5].

The flavor we chose for them was t1.small. The t1.tiny flavor has too few megabytes of RAM, and choosing t1.medium would have resulted in us not having enough space left for our other virtual machines.

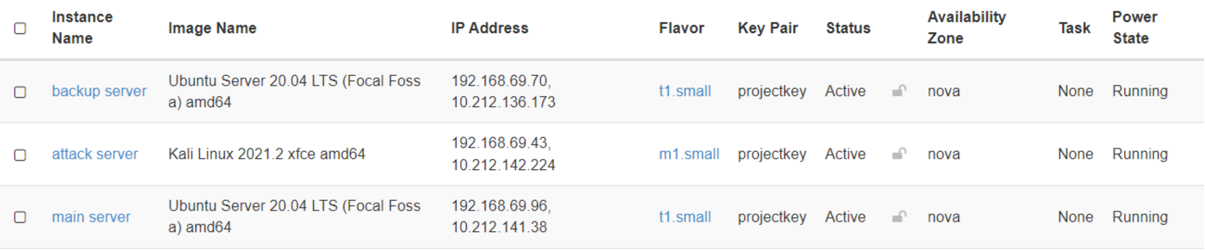
For the penetration test server we chose Kali Linux (2021.2) as its operating system. The reason for this is that Kali contains several good penetration testing tools, which we were interested in trying out and utilizing. For the flavor, we chose m1.small. The server with this operating system requires at least 1024 megabytes of RAM. We also wanted to use most of our VM quota, which is why we chose this flavor.



All three VMs are connected to one local network, which contains one subnet. We chose to have only one subnet, since we don't require a complicated network to support 4 users. Our subnet has this IP address: 192.168.69.0/25. Because of this, our local IP addresses will be in the range 192.168.69.1 - 192.168.69.126, and the subnet mask is 255.255.255.128. As a result of this, our network is of class c. This local network is connected to the NTNU internal network via a router.



Instance overview:



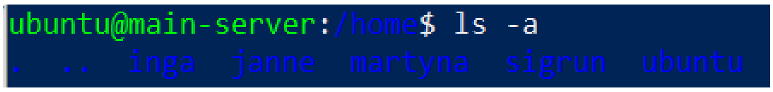
□	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status		Availability Zone	Task	Power State
□	backup server	Ubuntu Server 20.04 LTS (Focal Fossa) amd64	192.168.69.70, 10.212.136.173	t1.small	projectkey	Active	🔓	nova	None	Running
□	attack server	Kali Linux 2021.2 xfce amd64	192.168.69.43, 10.212.142.224	m1.small	projectkey	Active	🔓	nova	None	Running
□	main server	Ubuntu Server 20.04 LTS (Focal Fossa) amd64	192.168.69.96, 10.212.141.38	t1.small	projectkey	Active	🔓	nova	None	Running

[5] B. Moore «Ubuntu 20.04 (Focal Fossa)»

3.4 Security Mechanisms.

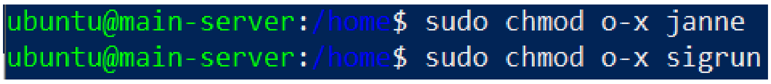
Authentication Approach & Users

We created four different users, one for each of us, and with each user owning their own personal folder.



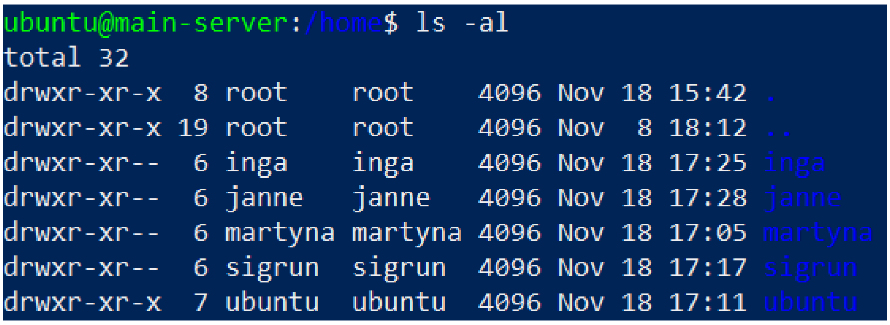
```
ubuntu@main-server:/home$ ls -a
.  ..  inga  janne  martyna  sigrun  ubuntu
```

After the users were made, we had to change permissions so that only the directory owner could open the user folders.



```
ubuntu@main-server:/home$ sudo chmod o-x janne
ubuntu@main-server:/home$ sudo chmod o-x sigrun
```

Here you can see that the execute permission for “others” is removed for all of the users folders:



```
ubuntu@main-server:/home$ ls -al
total 32
drwxr-xr-x  8 root    root    4096 Nov 18 15:42 .
drwxr-xr-x 19 root    root    4096 Nov  8 18:12 ..
drwxr-xr--  6 inga    inga    4096 Nov 18 17:25 inga
drwxr-xr--  6 janne   janne   4096 Nov 18 17:28 janne
drwxr-xr--  6 martyna martyna 4096 Nov 18 17:05 martyna
drwxr-xr--  6 sigrun  sigrun  4096 Nov 18 17:17 sigrun
drwxr-xr-x  7 ubuntu  ubuntu  4096 Nov 18 17:11 ubuntu
```

So now you can only access the folders as the user that owns them.

Ports used

Port 22 is the only port that is open on all of our servers. This is because port 22 is the port that SSH uses. This is another security measure, since having just this port open ensures that the machines are only able to connect with secure shell. all other ports were closed, since its good practice to always close all unnecessary unused ports.

SFTP

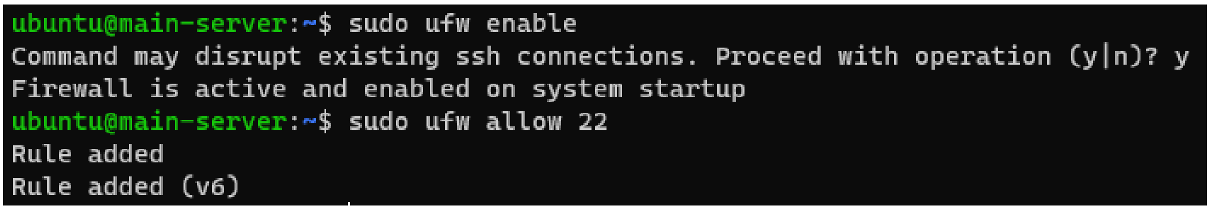
We decided on using SFTP instead of FTP to transfer the files between our server. We decided on this because SFTP uses encryption which FTP does not. This is another layer of security on our server, and makes transfering files more secure, since you need an encryption key to make sence of the data that we transfer.

The NTNU internal network

Another safety measure is that our network is connected to the NTNU internal network, which means that in order to have a chance at accessing the network, you have to physically be at the campus of NTNU and use the campus wifi. The wifi requires a password which is the same as the one students use as their course management system. This means that the user either needs a NTNU account, or needs to know the password of someone who studies at this university. Another option, which is the one we have used the most, is to use NTNU's own VPN, which requires a NTNU student's username and password.

Firewall

We used the built-in (Linux/Ubuntu) firewall on the main and back-up servers using the command sudo ufw enable.



```
ubuntu@main-server:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
ubuntu@main-server:~$ sudo ufw allow 22
Rule added
Rule added (v6)
```

Using the command "sudo ufw allow 22" we opened up port 22 that is used for ssh connections so that we can connect to it. Port 22 is also open for SFTP connections.

Key Pairs

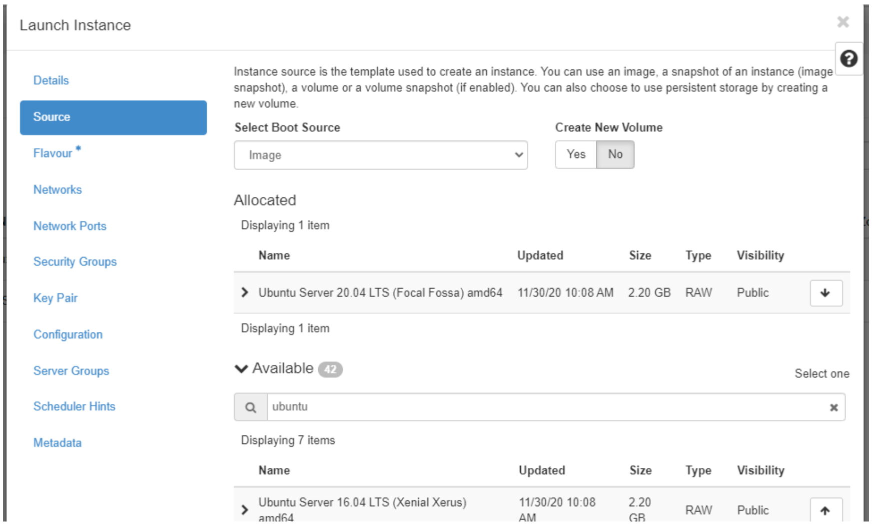
In order to access the virtual machines, the user will have to use the private key which only the four of us in this group has a copy of. This key is a reliable security measure which keeps unauthorized persons from being able to access the files uploaded to the VMs.

4. Implementation

The various tools we used in our project are described throughout this project report. Below are a few additional screenshots we have taken, with the description of what was attempted. In addition to this, here are a list of our most frequently used commands in SSH:

- pwd
- Shows where you currently are in the directory
- cd
  - Used to navigate the directory
  - Can specify pathname, go directly to the root or go to the parent directory of the one you are currently in
- ls
  - Lists all of the contents in the current directory
- mv
  - Used to move or rename a file
- rm
  - Used to delete a file
- cp
  - Used to copy items to another directory
- sudo
  - Allows a regular user to execute commands as another user
- sudo su
  - Allows a user to start a session as another user

Creating the VMs



Instance source is the template used to create an instance. You can use an image, a snapshot of an instance (image snapshot), a volume or a volume snapshot (if enabled). You can also choose to use persistent storage by creating a new volume.

Select Boot Source: Image

Create New Volume: Yes [✓ No]

Allocated

Displaying 1 item

Name	Updated	Size	Type	Visibility
Ubuntu Server 20.04 LTS (Focal Fossa) amd64	11/30/20 10:08 AM	2.20 GB	RAW	Public

Displaying 1 item

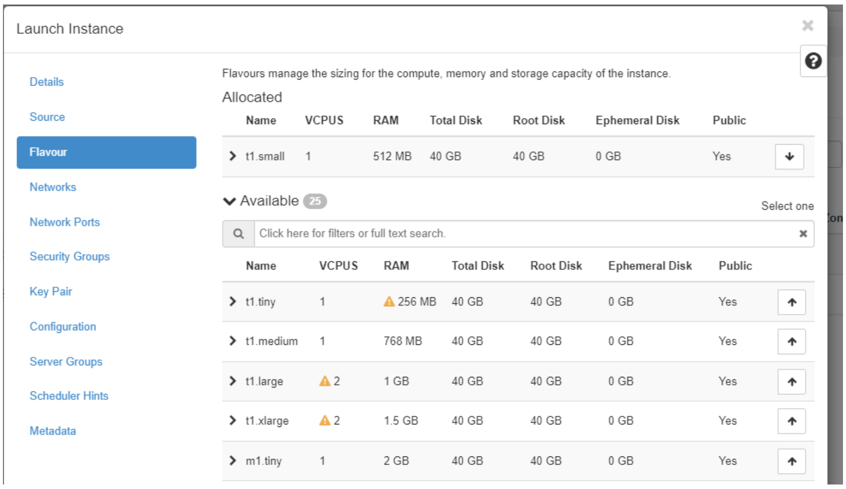
Available 42 — Select one

ubuntu

Displaying 7 items

Name	Updated	Size	Type	Visibility
Ubuntu Server 16.04 LTS (Xenial Xerus) amd64	11/30/20 10:08 AM	2.20 GB	RAW	Public

We chose Focal Fossa as the server type



Flavours manage the sizing for the compute, memory and storage capacity of the instance.

Allocated

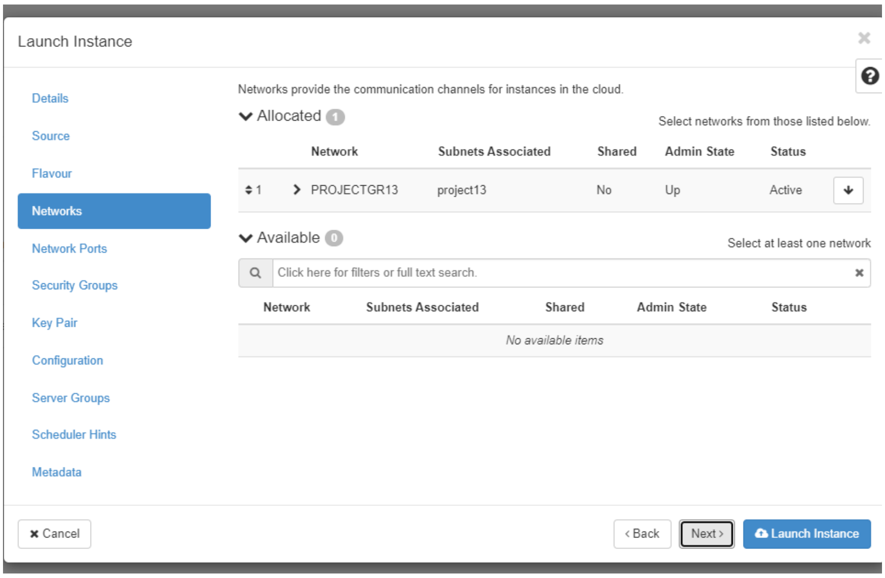
Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
t1.small	1	512 MB	40 GB	40 GB	0 GB	Yes

Available 25 — Select one

Click here for filters or full text search.

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
t1.tiny	1	⚠ 256 MB	40 GB	40 GB	0 GB	Yes
t1.medium	1	768 MB	40 GB	40 GB	0 GB	Yes
t1.large	⚠ 2	1 GB	40 GB	40 GB	0 GB	Yes
t1.xlarge	⚠ 2	1.5 GB	40 GB	40 GB	0 GB	Yes
m1.tiny	1	2 GB	40 GB	40 GB	0 GB	Yes

t1.small as flavor



Networks provide the communication channels for instances in the cloud.

Allocated 1 — Select networks from those listed below.

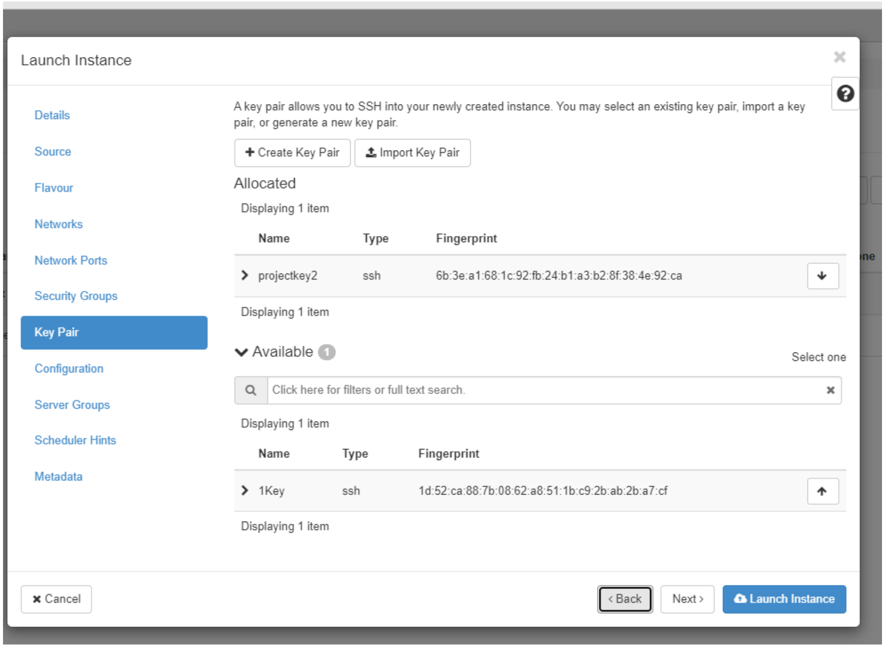
	Network	Subnets Associated	Shared	Admin State	Status
1	PROJECTGR13	project13	No	Up	Active

Available 0 — Select at least one network

Click here for filters or full text search.

Network	Subnets Associated	Shared	Admin State	Status
No available items				

Connecting the instance to our local network



A key pair allows you to SSH into your newly created instance. You may select an existing key pair, import a key pair, or generate a new key pair.

Allocated

Displaying 1 item

Name	Type	Fingerprint
projectkey2	ssh	6b:3e:a1:68:1c:92:fb:24:b1:a3:b2:8f:38:4e:92:ca

Displaying 1 item

Available 1 Select one

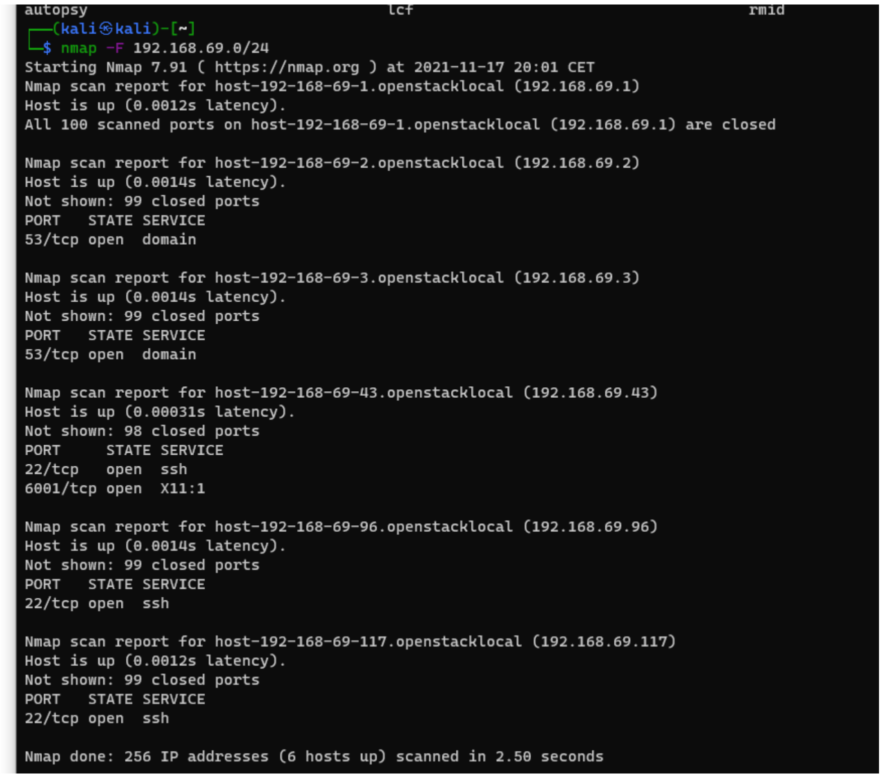
Click here for filters or full text search.

Displaying 1 item

Name	Type	Fingerprint
1Key	ssh	1d:52:ca:88:7b:08:62:a8:51:1b:c9:2b:ab:2b:a7:cf

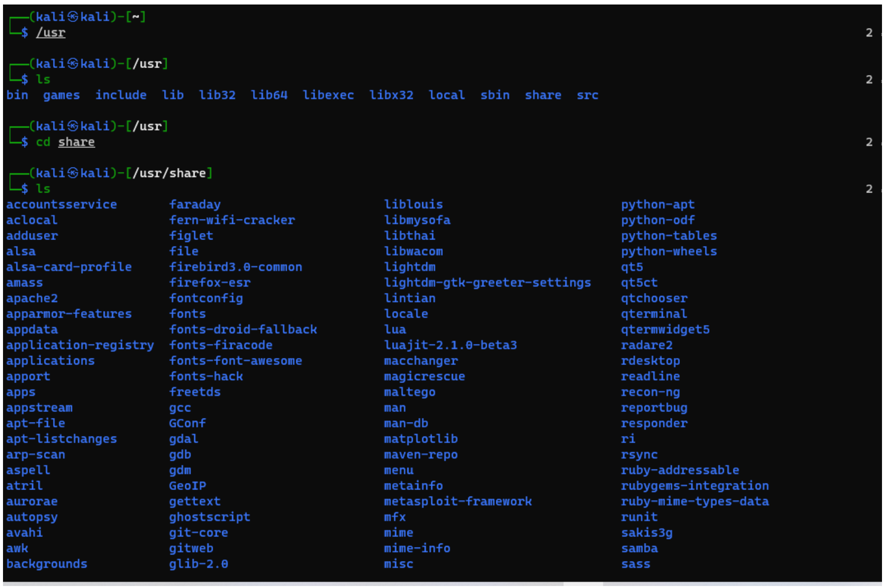
Displaying 1 item

Assigning our key pair



```
autopsy                                lcf                                    rmid
┌──(kali㉿kali)-[~]
└─$ nmap -F 192.168.69.0/24
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-17 20:01 CET
Nmap scan report for host-192-168-69-1.openstacklocal (192.168.69.1)
Host is up (0.0012s latency).
All 100 scanned ports on host-192-168-69-1.openstacklocal (192.168.69.1) are closed

Nmap scan report for host-192-168-69-2.openstacklocal (192.168.69.2)
Host is up (0.0014s latency).
Not shown: 99 closed ports
PORT   STATE SERVICE
53/tcp open  domain

Nmap scan report for host-192-168-69-3.openstacklocal (192.168.69.3)
Host is up (0.0014s latency).
Not shown: 99 closed ports
PORT   STATE SERVICE
53/tcp open  domain

Nmap scan report for host-192-168-69-43.openstacklocal (192.168.69.43)
Host is up (0.00031s latency).
Not shown: 98 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
6001/tcp open  X11:1

Nmap scan report for host-192-168-69-96.openstacklocal (192.168.69.96)
Host is up (0.0014s latency).
Not shown: 99 closed ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap scan report for host-192-168-69-117.openstacklocal (192.168.69.117)
Host is up (0.0012s latency).
Not shown: 99 closed ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 256 IP addresses (6 hosts up) scanned in 2.50 seconds
```

Attempt at penetration testing



```
┌──(kali㉿kali)-[~]
└─$ /usr

┌──(kali㉿kali)-[/usr]
└─$ ls
bin  games  include  lib  lib32  lib64  libexec  libx32  local  sbin  share  src

┌──(kali㉿kali)-[/usr]
└─$ cd share

┌──(kali㉿kali)-[/usr/share]
└─$ ls
accountsservice       faraday                      liblouis                        python-apt
aclocal               fern-wifi-cracker            libmysofa                       python-odf
adduser               figlet                       libthai                         python-tables
alsa                  file                         libwacom                        python-wheels
alsa-card-profile     firebird3.0-common           lightdm                         qt5
amass                 firefox-esr                  lightdm-gtk-greeter-settings    qt5ct
apache2               fontconfig                   lintian                         qtchooser
apparmor-features     fonts                        locale                          qterminal
appdata               fonts-droid-fallback         lua                             qtermwidget5
application-registry  fonts-firacode               luajit-2.1.0-beta3              radare2
applications          fonts-font-awesome           macchanger                      rdesktop
apport                fonts-hack                   magicrescue                     readline
apps                  freetds                      maltego                         recon-ng
appstream             gcc                          man                             reportbug
apt-file              GConf                        man-db                          responder
apt-listchanges       gdal                         matplotlib                      ri
arp-scan              gdb                          maven-repo                      rsync
aspell                gdm                          menu                            ruby-addressable
atril                 GeoIP                        metainfo                        rubygems-integration
aurorae               gettext                      metasploit-framework            ruby-mime-types-data
autopsy               ghostscript                  mfx                             runit
avahi                 git-core                     mime                            sakis3g
awk                   gitweb                       mime-info                       samba
backgrounds           glib-2.0                     misc                            sass
```

Another attempt at penetration testing

..

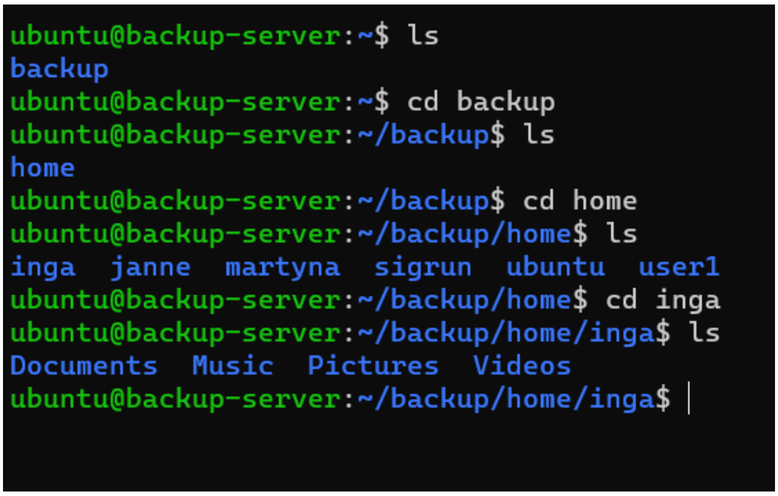
5. Evaluation

….

Functionality

Backup server

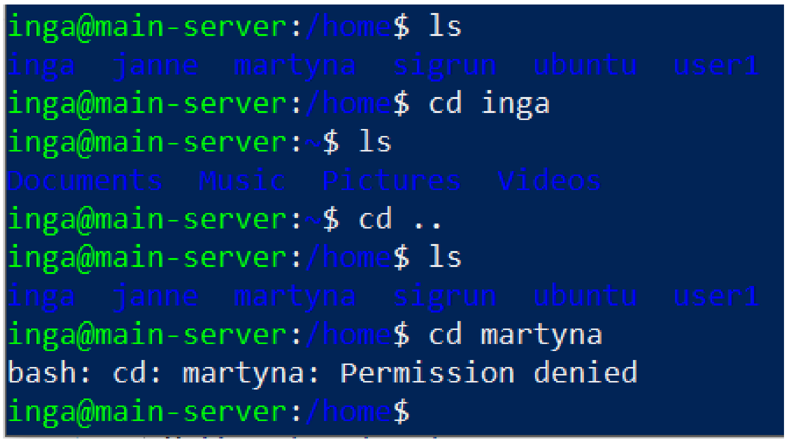
All of the user’s files are stored in the backup server. Here you can see one of the user’s folders which contain their files.



```
ubuntu@backup-server:~$ ls
backup
ubuntu@backup-server:~$ cd backup
ubuntu@backup-server:~/backup$ ls
home
ubuntu@backup-server:~/backup$ cd home
ubuntu@backup-server:~/backup/home$ ls
inga  janne  martyna  sigrun  ubuntu  user1
ubuntu@backup-server:~/backup/home$ cd inga
ubuntu@backup-server:~/backup/home/inga$ ls
Documents  Music  Pictures  Videos
ubuntu@backup-server:~/backup/home/inga$ |
```

Main server

When signed into one user’s account, you have access to that user’s files. But you cannot access any of the other user’s files.



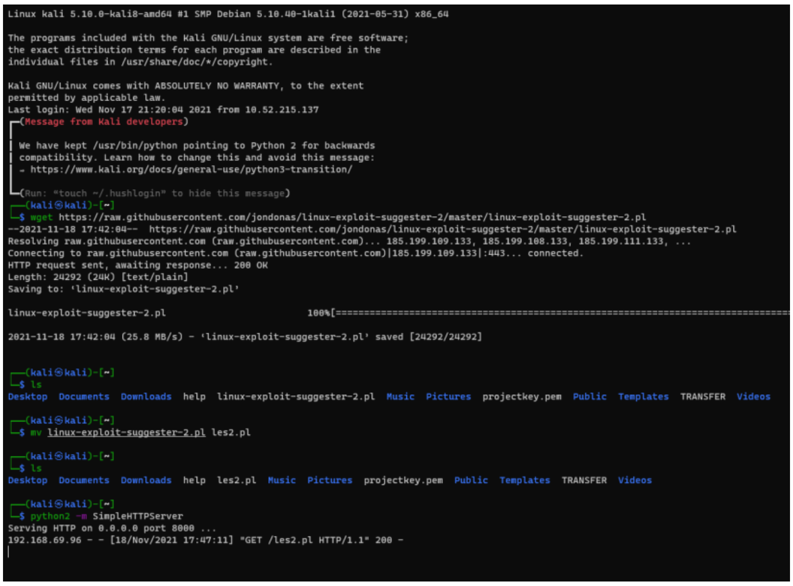
```
inga@main-server:/home$ ls
inga  janne  martyna  sigrun  ubuntu  user1
inga@main-server:/home$ cd inga
inga@main-server:~$ ls
Documents  Music  Pictures  Videos
inga@main-server:~$ cd ..
inga@main-server:/home$ ls
inga  janne  martyna  sigrun  ubuntu  user1
inga@main-server:/home$ cd martyna
bash: cd: martyna: Permission denied
inga@main-server:/home$
```

Penetration testing

In order to test that our server is properly protected, we first attempted to do penetration testing by using Metasploit, which is a program designed for discovering weaknesses in machines. But after being almost finished with the process we discovered that Metasploit had very limited options for testing Linux systems. Because of this we decided to try a different and more accurate method.

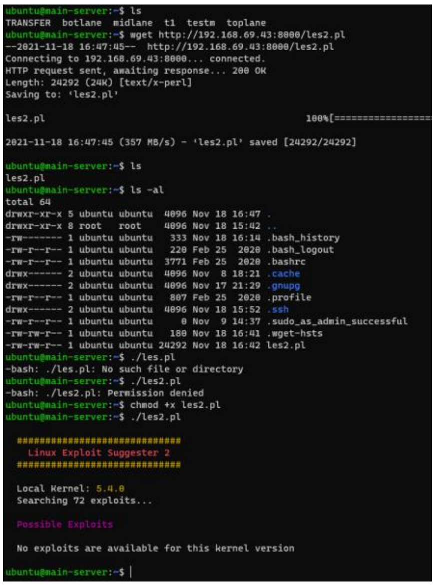
In our second attempt at penetration testing, we tested it using Linux Exploit Suggester 2, which is a tool designed to detect security weaknesses within a server or machine[6]. One of our servers acted as an attack server, which is the one we used for the penetration testing. Using the command “wget” we downloaded Linux Exploit Suggester 2. We then set up a simple http server, and downloaded the exploit suggester onto our main server through this connection. In order to run Linux Exploit Suggester 2, we needed execute permission, which we obtained by using the “chmod +x” command. Then, after running the exploit suggester, we were able to see that there were no weaknesses found on our server.

Attack server



```
Linux kali 5.10.0-kali8-amd64 #1 SMP Debian 5.10.40-1kali1 (2021-05-31) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Nov 17 21:20:04 2021 from 10.52.215.137
┌─(Message from Kali developers)
│
│ We have kept /usr/bin/python pointing to Python 2 for backwards
│ compatibility. Learn how to change this and avoid this message:
│ ⇒ https://www.kali.org/docs/general-use/python3-transition/
│
└─(Run: “touch ~/.hushlogin” to hide this message)
┌──(kali㉿kali)-[~]
└─$ wget https://raw.githubusercontent.com/jondonas/linux-exploit-suggester-2/master/linux-exploit-suggester-2.pl
--2021-11-18 17:42:04--  https://raw.githubusercontent.com/jondonas/linux-exploit-suggester-2/master/linux-exploit-suggester-2.pl
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.109.133, 185.199.108.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24292 (24K) [text/plain]
Saving to: ‘linux-exploit-suggester-2.pl’

linux-exploit-suggester-2.pl                       100%[==========================================================================

2021-11-18 17:42:04 (25.8 MB/s) - ‘linux-exploit-suggester-2.pl’ saved [24292/24292]


┌──(kali㉿kali)-[~]
└─$ ls
Desktop  Documents  Downloads  help  linux-exploit-suggester-2.pl  Music  Pictures  projectkey.pem  Public  Templates  TRANSFER  Videos

┌──(kali㉿kali)-[~]
└─$ mv linux-exploit-suggester-2.pl les2.pl

┌──(kali㉿kali)-[~]
└─$ ls
Desktop  Documents  Downloads  help  les2.pl  Music  Pictures  projectkey.pem  Public  Templates  TRANSFER  Videos

┌──(kali㉿kali)-[~]
└─$ python2 -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
192.168.69.96 - - [18/Nov/2021 17:47:11] "GET /les2.pl HTTP/1.1" 200 -
|
```

Main server



```
ubuntu@main-server:~$ ls
TRANSFER  botlane  midlane  t1  testm  toplane
ubuntu@main-server:~$ wget http://192.168.69.43:8000/les2.pl
--2021-11-18 16:47:45--  http://192.168.69.43:8000/les2.pl
Connecting to 192.168.69.43:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24292 (24K) [text/x-perl]
Saving to: ‘les2.pl’

les2.pl                                           100%[=================

2021-11-18 16:47:45 (357 MB/s) - ‘les2.pl’ saved [24292/24292]

ubuntu@main-server:~$ ls
les2.pl
ubuntu@main-server:~$ ls -al
total 64
drwxr-xr-x 5 ubuntu ubuntu  4096 Nov 18 16:47 .
drwxr-xr-x 8 root   root    4096 Nov 18 15:42 ..
-rw------- 1 ubuntu ubuntu   333 Nov 18 16:14 .bash_history
-rw-r--r-- 1 ubuntu ubuntu   220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu  3771 Feb 25  2020 .bashrc
drwx------ 2 ubuntu ubuntu  4096 Nov  8 18:21 .cache
drwx------ 2 ubuntu ubuntu  4096 Nov 17 21:29 .gnupg
-rw-r--r-- 1 ubuntu ubuntu   807 Feb 25  2020 .profile
drwx------ 2 ubuntu ubuntu  4096 Nov 18 15:52 .ssh
-rw-r--r-- 1 ubuntu ubuntu     0 Nov  9 14:37 .sudo_as_admin_successful
-rw-rw-r-- 1 ubuntu ubuntu   180 Nov 18 16:41 .wget-hsts
-rw-rw-r-- 1 ubuntu ubuntu 24292 Nov 18 16:42 les2.pl
ubuntu@main-server:~$ ./les.pl
-bash: ./les.pl: No such file or directory
ubuntu@main-server:~$ ./les2.pl
-bash: ./les2.pl: Permission denied
ubuntu@main-server:~$ chmod +x les2.pl
ubuntu@main-server:~$ ./les2.pl

  #############################
    Linux Exploit Suggester 2
  #############################

  Local Kernel: 5.4.0
  Searching 72 exploits...

  Possible Exploits

  No exploits are available for this kernel version

ubuntu@main-server:~$ |
```

6. Reflection

6.1 Individual

This section is handed in individually by all the members.

6.2 Teamwork and cooperation

[6] J. Donas «linux-exploit-suggester-2»

The teamwork went well. We all are quite close, and talk to each other often, so bringing up ideas was easy. We trusted each other to do all the work that we could, and to try our best. We also live close to each other in Gjøvik, so meating up to work on the project was easy. Other than that we also got a lot of help from other groups that knew this topic better than us, and when we had a problem they were more than helpful in helping us find a solution.

The working contract we made has more or less been followed. Since we are friends beforehand, and trust each other, we didn't really refer to it that much. But it's nice to have as a backup in case anything happens to any of us, so that we knew the steps to take to fix problems in the group. I think a group with people that dont know each other, have most use of the working contract

6.3 Project

This project was a big roller coaster with highs and lows. Many times we got stuck on dead ends, where we had to start over or find new solutions. We started the project with a very generic proposal with little knowledge on the topic, and gained new insight as we went. as we learned we found better ways to do the project, and had to change our plan. The process of our project can pretty much be divided into two chapters where we had different plans and aspirations.

1. We started with a simple plan of making a big server with three of our VMs acting as a server, and one as a client. This plan was too broad and unspecific, and as a result we had a hard time getting started. In the end we decided this plan was too hard, and we had no use for that big of a server. This plan also left us with no VM to use as a penetration tester, which was not ideal.

2. After this we decided on a simple solution of making a small text message like file sharing. We would then use one VM as the main server, one for backup, one for client and one for penetration testing.

When we then had a solid plan, the project moved along much more smoothly. We started by simply setting up our VMs through skyhigh, which was in itself quite simple, since we have done it before. Then, we worked out how to set up the users, which took quite some time. And after that we learned how to set up the backup. The backup was probably one of the more difficult aspects of our project. We had a lot of trouble making it work. We tried several different methods.

In our first attempt we used a program called Duplicity. This program is used to copy files over to a backup server, and it has a feature where you can make it automatically update your backup files at set intervals. However, after working with this Duplicity for a while, we found out that the program did not contain the commands needed to achieve our goal. Therefore, we tried another solution.

We spent a long time learning how to encrypt and decrypt the folders in order to safely transfer them from the main server to the backup server. But even after trying tirelessly we could not get it to work. After collectively trying for the whole day, we decided to try yet another solution.

Our last plan was to transfer the data manually by using sftp. This did work, but we wish we were able to succeed with one of our previous attempts. Even if it is not the best solution, we now do have a backup server with all of our files in it.

This project can later help us when furthering our general computer understanding and skills. We have learned much about how computers and data systems operate, and also used several security and operating programs that may become useful later in our career. We may have use for it later if we go into computer programming.

Since we now have gone through this entire process, we now know more of what works and what doesn't work, and we will spend less time if we were to do something similar later. If we were to further develop this program, we could set up a website connected to this server, and we can also use that to program websites in the GraffikProgramming course.

7. Conclusions

All four of our group members had no experience around cybersecurity or servers. This made it hard for us as a group to get started, since we had no idea what was expected of us, what type of server was manageable, or what different types of servers meant for us as a group. In the end we had to spend a lot of hours researching and implementing ideas, only to find out that it wouldn't work for us, and we'd have to start over.

In the end all of our trial and error ment we learned a lot when it comes to overall understanding of the topic. We also gained a lot of weird knowledge that we probably don't need yet. Like how we spent a lot of time researching IDS, only to then find out that mostly you need a database to make it work[7].

Our group also feels like a big help to next year's group,could be to give them more specifics for the project (what is expected, or examples of what this year did), and maybe tips on how to get started, since that is what we heard a lot of groupsg had problems with.

[7] Teaching Assistant we don't know the name of

8. References

ieeetr reference style

- T. Haddon «An introduction to Ubunto» https://www.connectingup.org/learn/articles/introduction-ubuntu 2007 Acessed: 2021/10/18
- g0tmi1k «What is Kali Linux» https://www.kali.org/docs/introduction/what-is-kali-linux/ Nov 2021 Acessed: 2021/11/11
- D. Wayne “What is WinSCP?” https://smallbusiness.chron.com/winscp-77185.html No Date Acessed: 2021/11/11
- University of Sussex «PuTTY» https://www.sussex.ac.uk/its/services/software/owncomputer/putty No date Acessed: 2021/10/27
- B. Moore «Ubuntu 20.04 (Focal Fossa)» https://in.pcmag.com/mobile-operating-system/136421/ubuntu-2004-focal-fossa Apr 2021 Acessed: 2021/10/27
- J. Donas «linux-exploit-suggester-2» https://github.com/jondonas/linux-exploit-suggester-2/blob/master/linux-exploit-suggester-2.pl Apr 2021 Acessed: 2021/11/16
- B. King «Ubuntu: A Beginners Guide» https://www.makeuseof.com/tag/ubuntu-an-absolute-beginners-guide/ Oct 2017 Acessed: 2021/10/18
- J. Ellingwood «How To Use SFTP to Securely Transfer Files with a Remote Server»https://www.digitalocean.com/community/tutorials/how-to-use-sftp-to-securely-transfer-files-with-a-remote-server Nov 2020 Acessed: 2021/11/06
- Teaching Assistant

9. Appendix

Work Contract

Arbeidskontrakt for Gruppe 13

Gruppas medlemmer:

- Sigrun Andersen Høgstedt, sigruah@stud.ntnu.no
- Martyna Maria Juga, martynmj@stud.ntnu.no
- Inga Kvam, ingakv@stud.ntnu.no
- Janne Wenger, jannew@stud.ntnu.no

Denne arbeidskontrakten bygger på et sett med typiske mål, oppgavefordelinger, prosedyrer og retningslinjer for interaksjoner for studentarbeider. Arbeidskontrakten er utfylt med egne fortolkninger av hva man mener med disse og hvordan man skal oppnå dette.

Forventninger:

- Vi skal møtes for å jobbe på prosjektet under labene / øvingstimene.
  - Hvis nødvendig, møtes utenfor timene.
  - Møtene skal enten være digitalt på Discord, eller fysisk på skolen eller annet avtalt sted
  - Planlegge møtene før de skal holdes
  - Hvis tidene som avtales ikke passer, kan vi jobbe med prosjektet til en annen tid
- Møtes presis
- Arbeidsmengden skal spres jevnt over alle i gruppa
- Hvis mulig skal alle møte opp til tidene vi avtaler
  - Si ifra på forhånd hvis du ikke kan møtes
- Alle medlemmer skal være forberedt før møtene
- Gjøre ferdig innleveringer en liten stund før fristen (for eksempel en dag før)
- Dokumenter og innleveringer lages på Google Docs, slik at lagring skjer automatisk.

Resultatmål

- Alle oppgaver skal leveres til rett tid.
- Alle skal gjøre arbeidet som gruppa blir enige opp, til sine beste evne innen avtalt tid.
- Alt arbeid skal bli sett over og alle skal bli enige før arbeidet leveres inn.

- Gruppa har et kollektivt ansvar for at alle skjønner og forstår arbeidet som leveres inn.

Rolle- og ansvarsfordeling

- Gruppeleder: Janne Wenger
- Ansvar for å levere inn arbeidet, og ansvar for opprettelse av nye dokumenter/prosjekter: Inga Kvam
- Ansvar for å arrangere ekstra møter hvis nødvendig: Martyna Maria Juga
- Ansvar for at alle vurderingskriterier er møtt: Sigrun Andersen Høgstedt

Hvordan løse uforventede problemer

- Hvis det oppstår noen problemer, skal gruppelederen kontaktes først. Gruppelederen skal da beslutte om et gruppemøte er nødvendig eller om et en-til-en møte skal til
- Gruppemøte kan kalles inn av gruppeleder, om det er behov for dette. alle har da ansvar for å prøve sitt beste for å finne en tidligst mulig ledig tid hvor alle kan møtes.
- Hvis det oppstår uenighet skal det bli holdt en avstemning.
  - Hvis resultatene av avstemningen er uavgjort, skal gruppa spørre en studentassistent eller lærer / foreleser om deres mening.
- Hvis noen ikke kan møte opp over lengre tid skal gruppemøte tilkalles, og ansvaret til den personen fordeles over de andre medlemmene, og lærer varsles.

Signaturer:

- Janne Wenger 22/09/2021
- Inga Kvam 22/09/2021
- Martyna Maria Juga 22/09/2021
- Sigrun Andersen Høgstedt 22/09/2021

Project Proposal Version 1.

Project proposal Group Stardrop

Each group will get 4 VMS

1. How many of them will act as a server. What type of server are you planning to set up? how many of them will act like a client?

We are planning to use 3 of them as a server, since we then have a couple ones to work with, with out it becoming to much work. And we will use the last one to act as the client, so we can use it to connect to the server.

We are planning on just using the SSH VM's we already have, through skyhigh, since that is what we are most familiar with.

1. How do you plan to protect/secure your server?

We are using ssh authentication and key pairs, we are going to block animated guessing, and we are also going to keep track of the hashvalue of any files created, to make sure they don't change..

2. How will you prove that your server is secure?

If we figure out how, we will test the server with our one VM acting as the client, or maybe another way, we'll se when we get there.

We keep the right to change our scenario as we go, since we are probably going to learn more about these things in the coming weeks. And most likely, we are going to meet dead ends, and well have to change paths.

Project Proposal Version 2.

Project proposal Group Stardrop (updated)

Each group will get 4 VMS

1. How many of them will act as a server. What type of server are you planning to set u? how many of them will act like a client?

We are planning to use one of them as a server, and another one as a backup. The server will work as a chat room, where small amounts of data gets uploaded at a time and is stored in the virtual machines. One of our remaining VMs will act as a client, where the chat texts can be uploaded and viewed from. Our last virtual machine will be used to test our server's firewalls and security.

We are planning on using SSH VM through skyhigh, and ssh servers since that is what we are most familiar with.

1. How do you plan to protect/secure your server?

We are using ssh authentication and key pairs, and we are also going to keep track of the hashvalue of any files created, to make sure they don't change. We will also set up a firewall.

2. How will you prove that your server is secure?

Our backup server will prevent data loss. Additionally, we will test the security via our attack server.

We keep the right to change our scenario as we go, since we are probably going to learn more about these things in the coming weeks. And most likely, we are going to meet dead ends, and we'll have to change paths.

Individual reflection note regarding the IIKG project

After working on this project I know a lot more about cybersecurity than when I first started. I had no previous knowledge and didn’t know where to even begin. But after the initial breakthrough when we were finally able to get our virtual machines to work, it got better. There has been a lot of trial and error to get this project to work. We have tried getting the servers to communicate in the way we want them to, we have tried setting up certain defence mechanisms, we have tried following a lot of articles and instructive videos, but most of the time we run into errors without finding a working workaround. Because of this, we have switched up some details of our project and our methods of getting those to work. As I wrote earlier, every member of our group has no previous knowledge about this subject, and having to learn everything from other students and the internet has been hard. I would have liked it if we could have learned some of the basic knowledge needed in the lectures. However, I appreciate that we were able to choose our own group, as being able to have my friend group work together on a common project is much better than working with separate groups.