Lab 2-3

Install and Use ClamAV

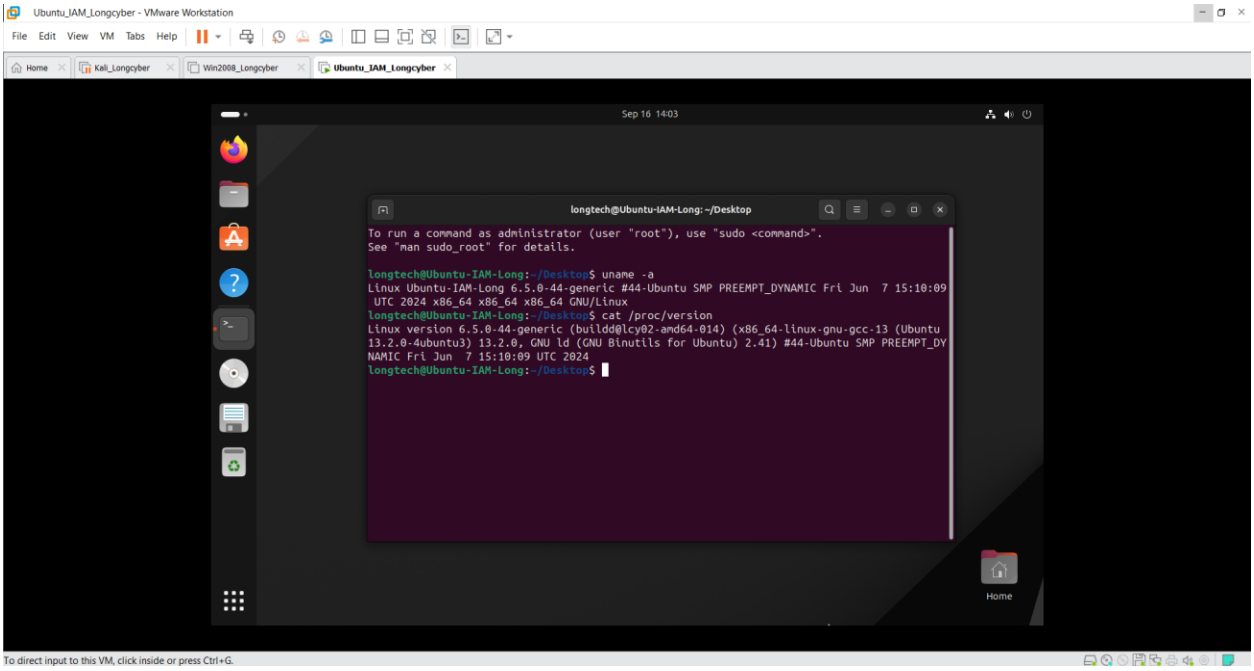
Course Name: IAM302

Student Name: Phạm Thành Long

Instructor Name: Mai Hoàng Đỉnh

Lab Due Date: 16/09/2024

OS Install:

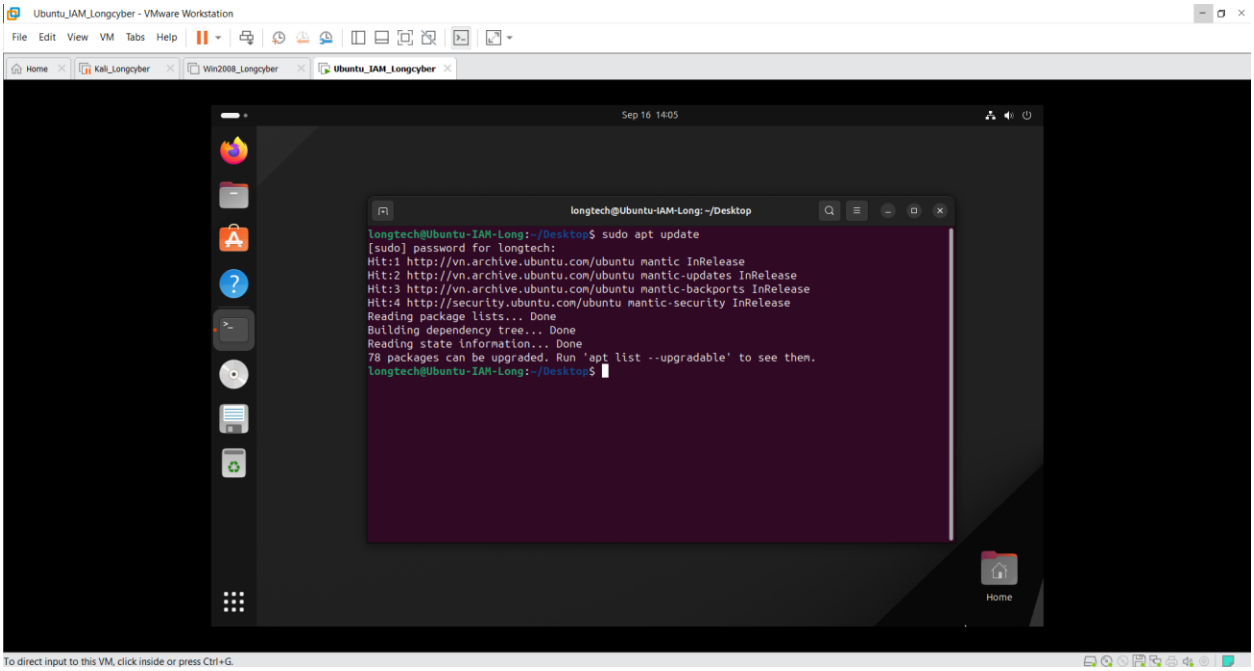


ClamAV is a well-reputed free and open-source antivirus software tool. It provides a command line interface that quickly scans the Linux system against viruses and malware attacks. The “ClamAV” helps scan the important part of Linux, i.e., mail gateways and emails directly affecting the network.

Install ClamAV on Ubuntu

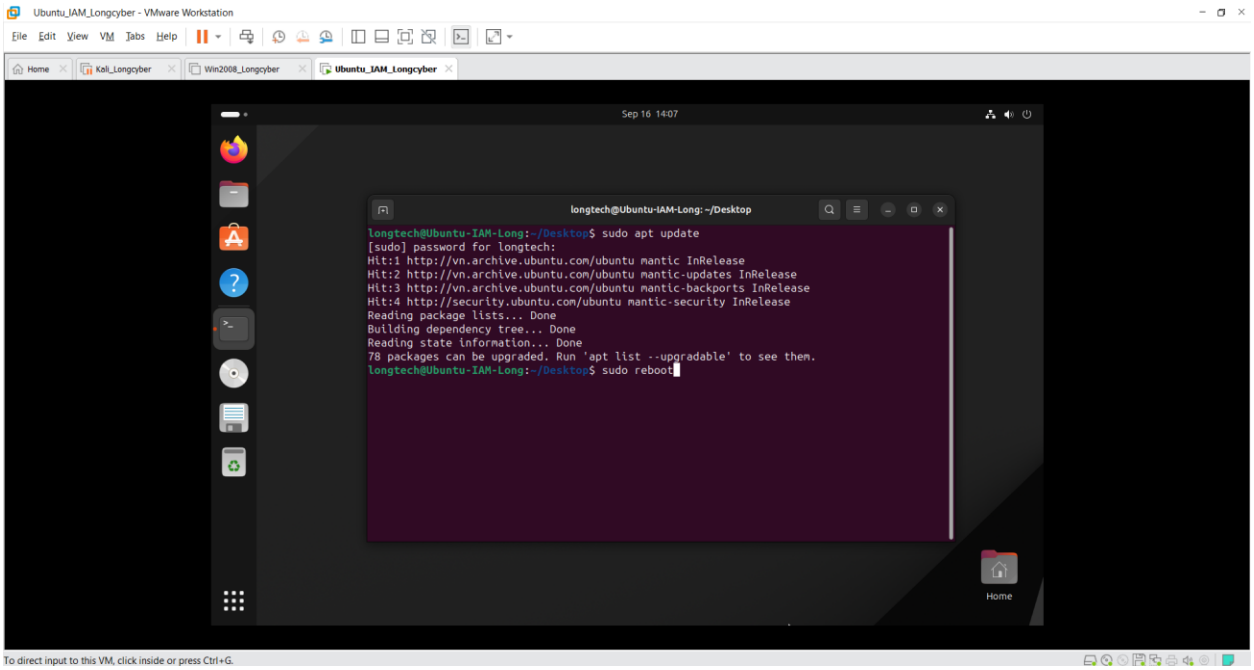
Step 1: Update the Repository

```
sudo apt-get update
```



We need to reboot after the update is complete

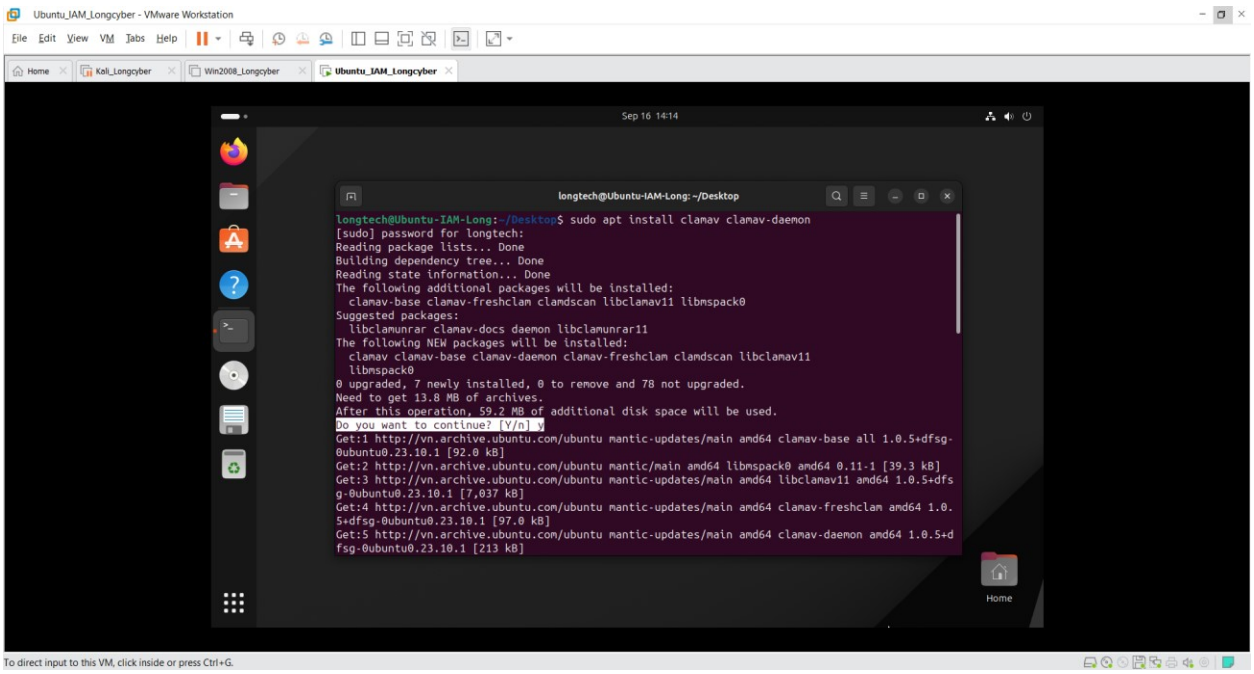
```
sudo reboot
```

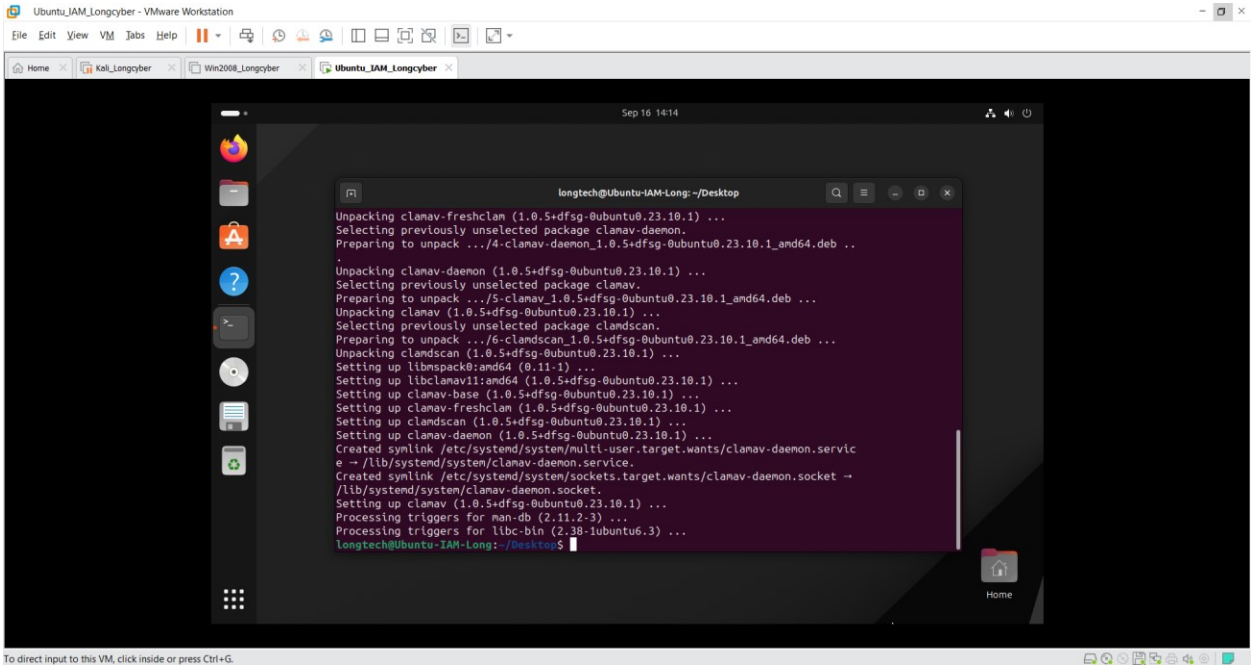


Step 2: Install ClamAV

Install the “ClamAV” application alongside the “clamav-daemon” from the standard repository of Ubuntu using the default “apt” package manager:

```
sudo apt install clamav clamav-daemon
```

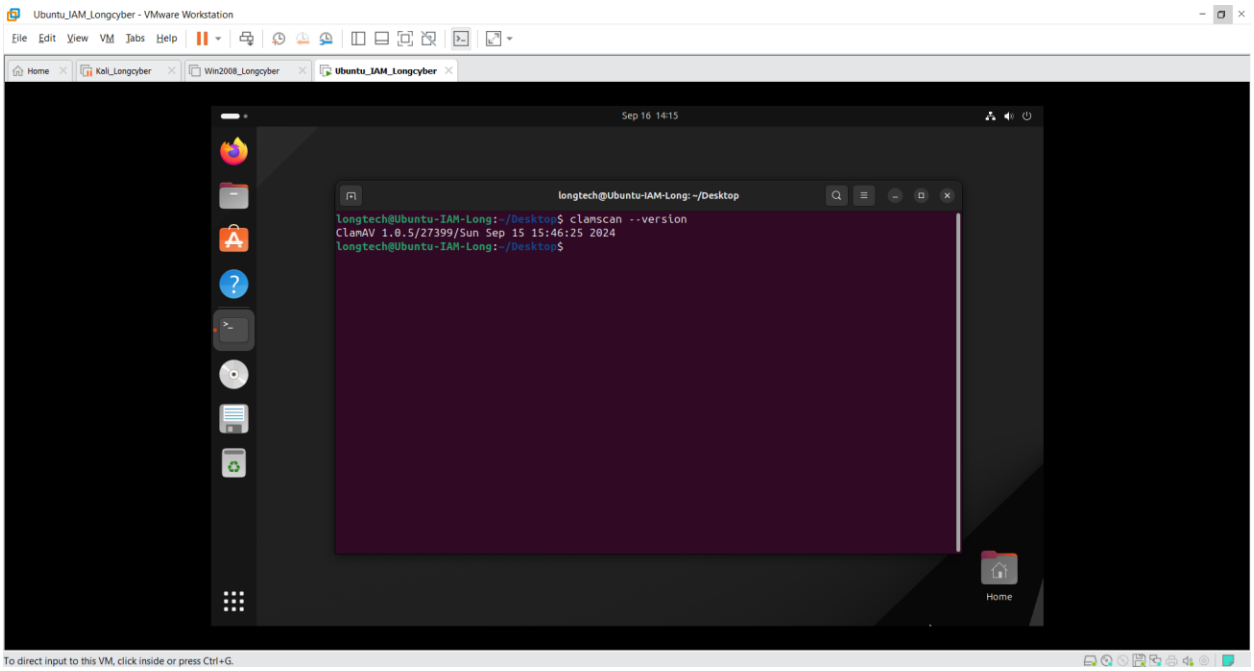




Step 3: Verify ClamAV

Check the installed version of the “clamav” scanner for verification purposes:

```
clamscan --version
```



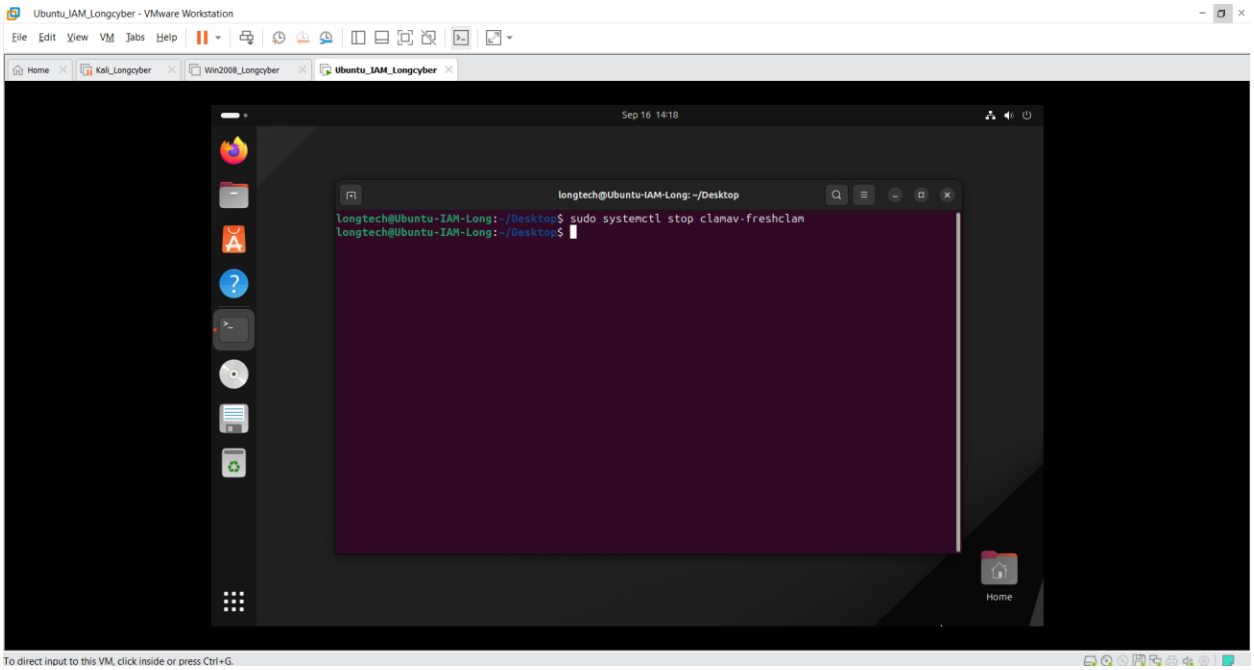
The “ClamAV” works on a signature database that identifies the malware. It requires updation regularly that ensures the application is up to date for protection against the latest threats.

Keeping this in view, Let’s update the installed “ClamAV” signature database:

Disable the “freshclam” Service

The pre-installed “freshclam” service automatically downloads the database updates. For the manual updation, disable/stop the “freshclam” service using the “systemctl” command:

```
sudo systemctl stop clamav-freshclam
```

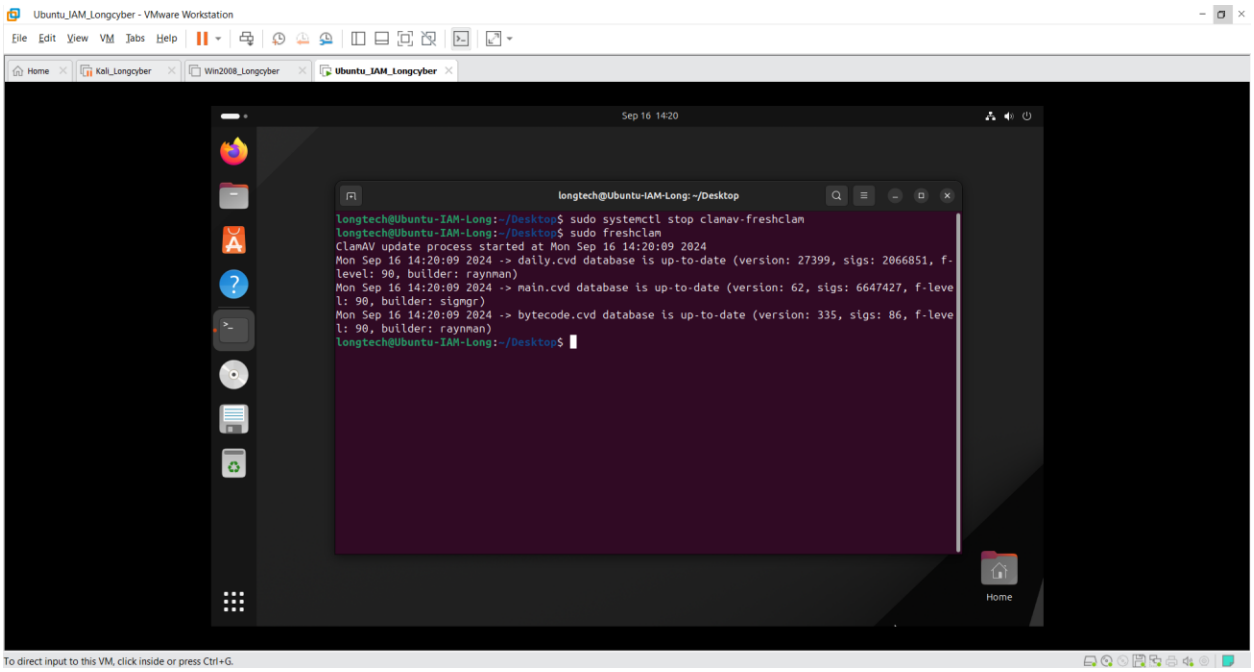


The “freshclam” service has been stopped.

Download Updates Using freshclam (First Method)

The first convenient way is to download the latest signature database update using “freshclam” via the superuser privileges, i.e., “sudo” command:

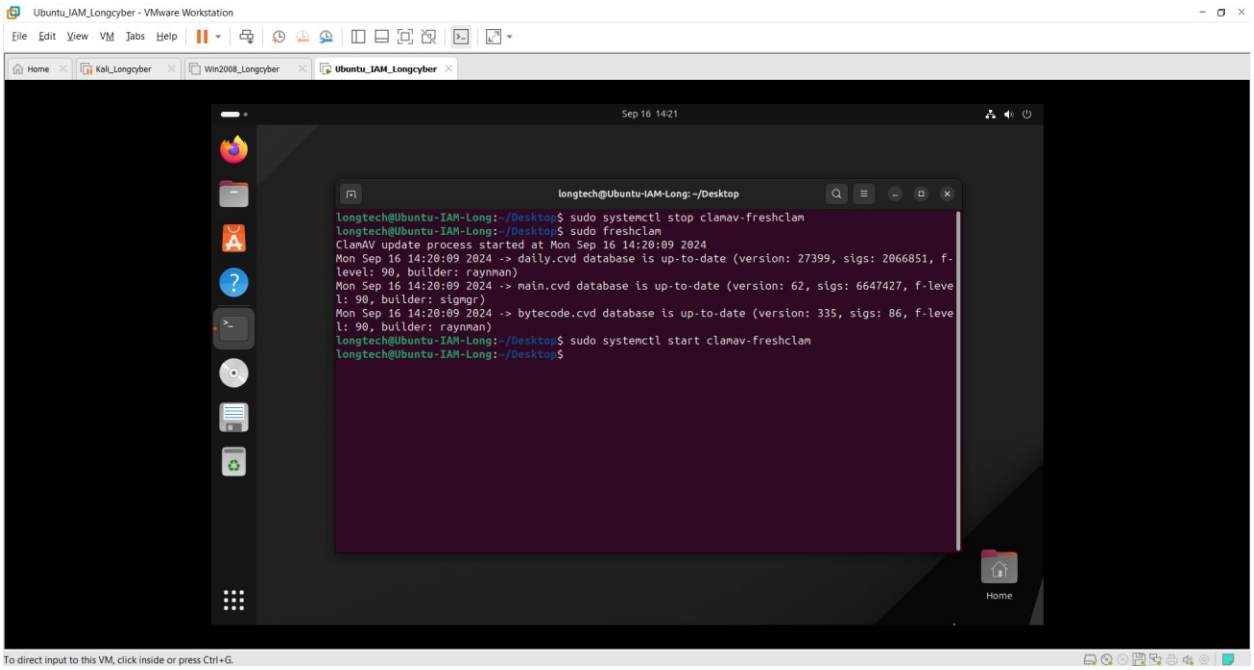
```
sudo freshclam
```



The output shows that the installed “ClamAV” database is up to date.

When all the updates are downloaded, start/enable the “freshclam” service again with the help of the “systemctl” command:

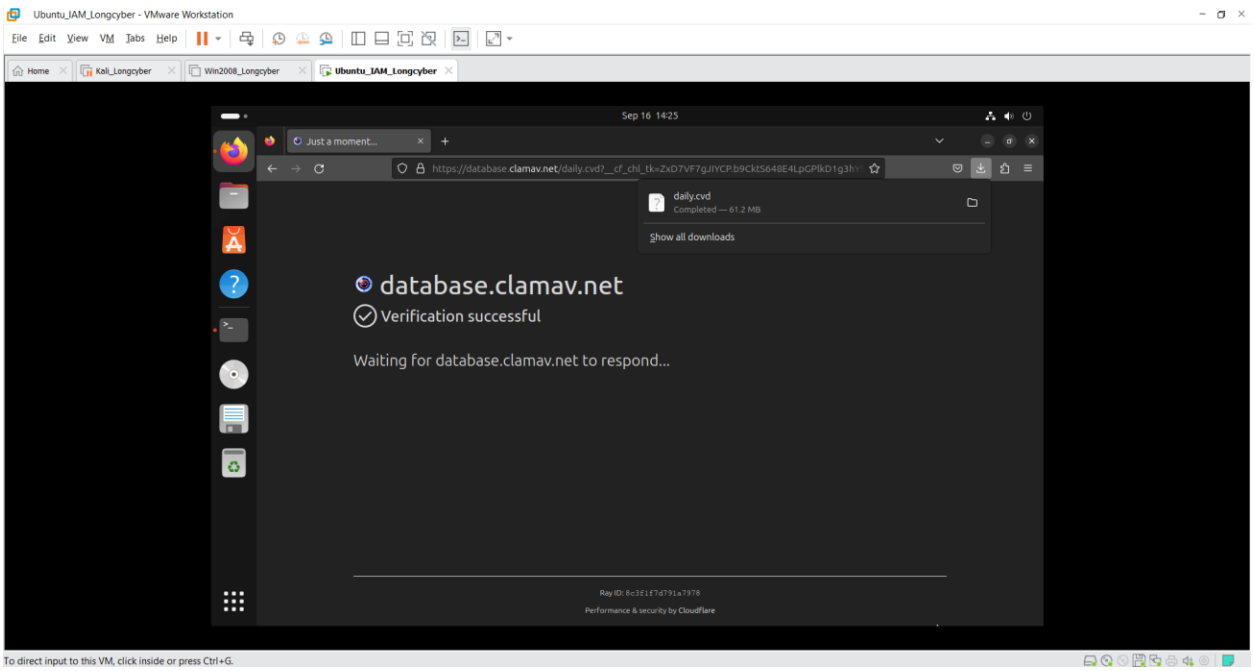
```
sudo systemctl start clamav-freshclam
```



Download Updates Using Official Website (Second Method)

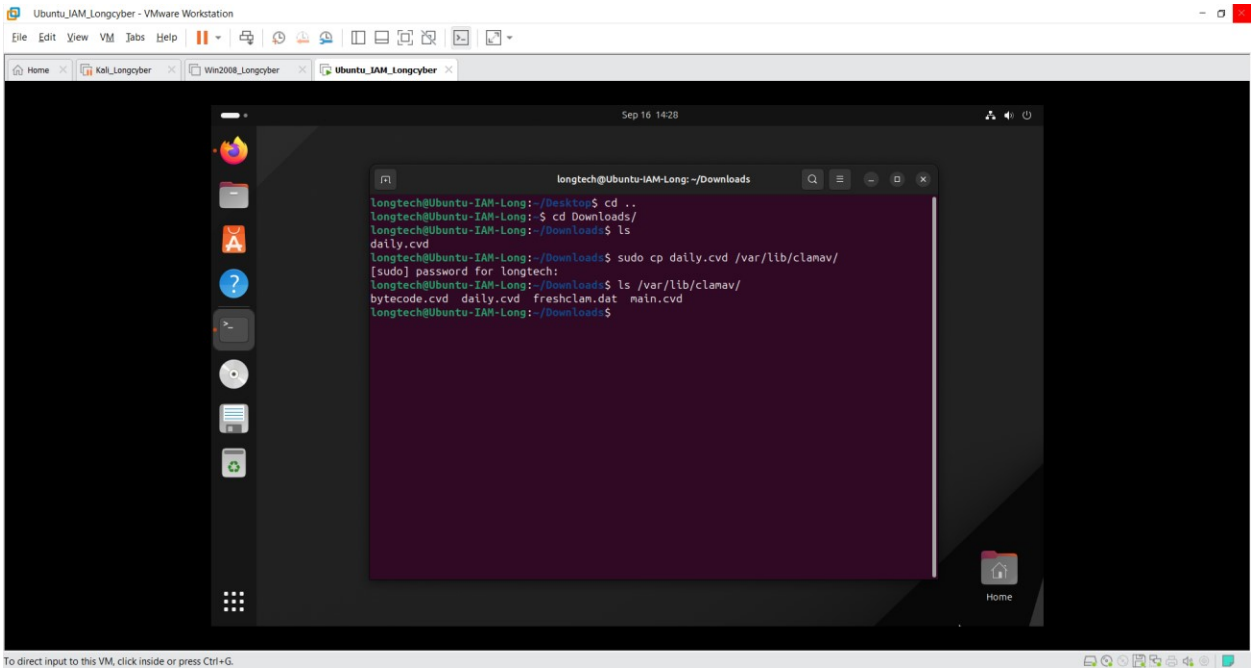
Another way is to download the “ClamAV” database from its official website https://database.clamav.net/daily.cvd

Click on the provided link, and it downloads the “daily.cvd” file.



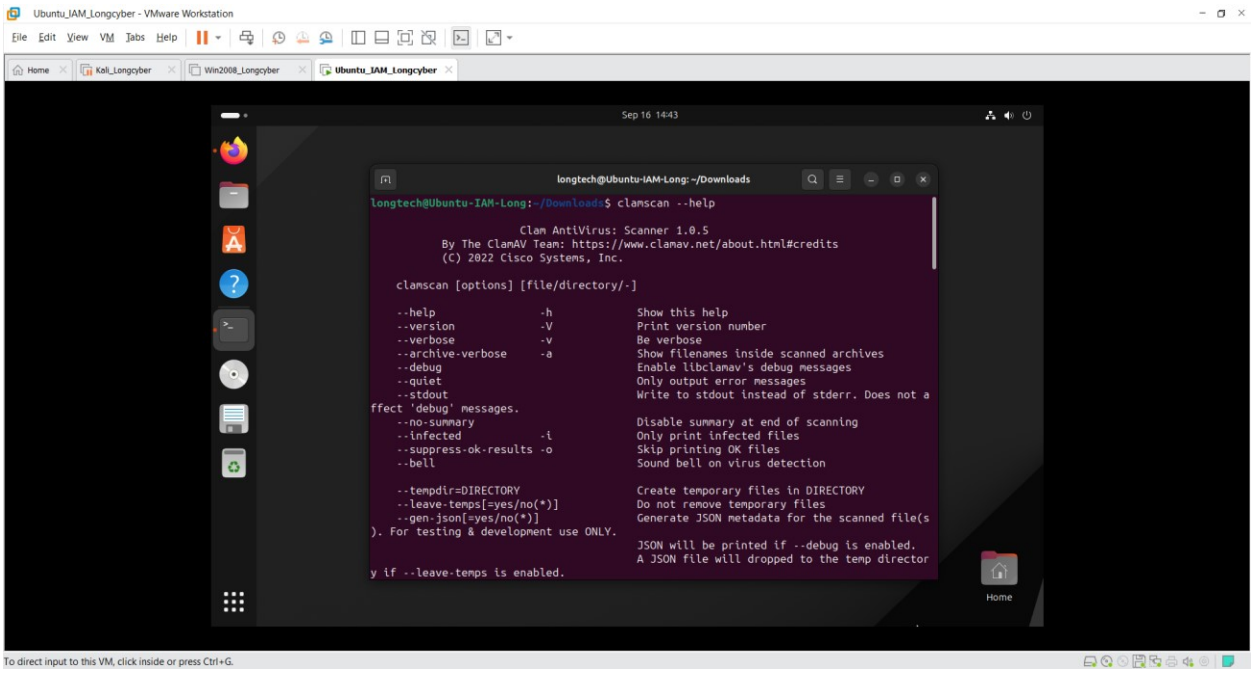
Copy the “daily.cvd” file into the “var/lib/clamav” file through the copy command “cp”:

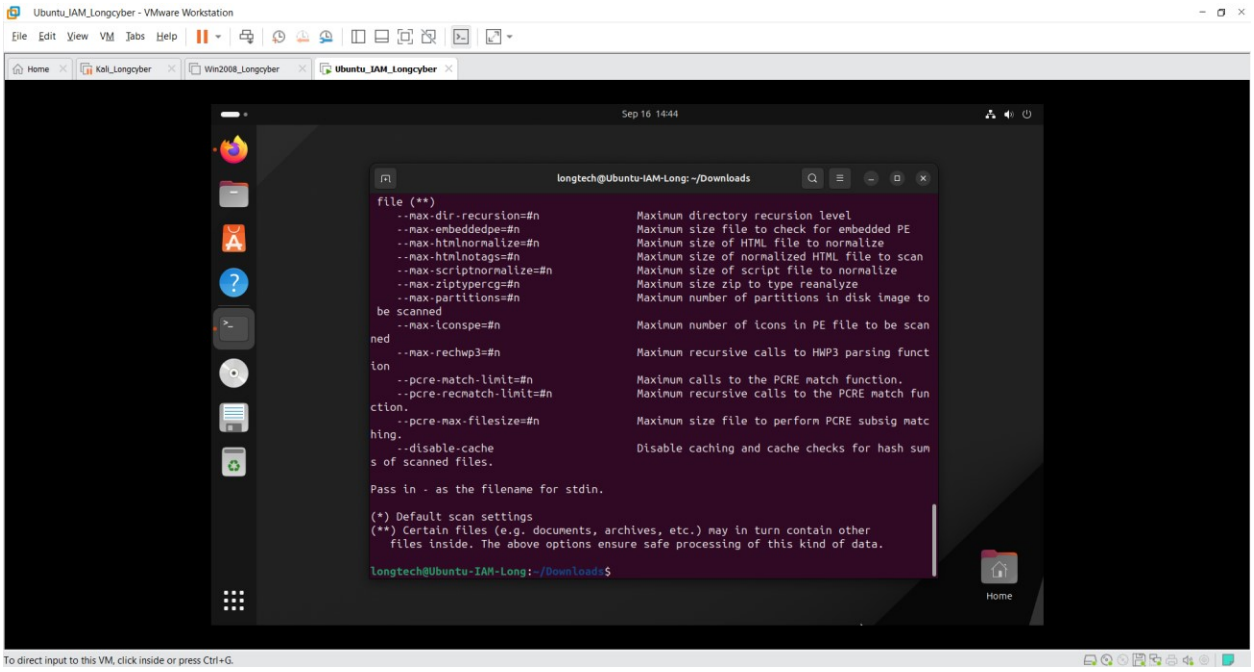
```
sudo cp daily.cvd /var/lib/clamav/
```



The “clamscan” provides a wide range of options that can be seen through its “help” command:

```
clamscan –help
```

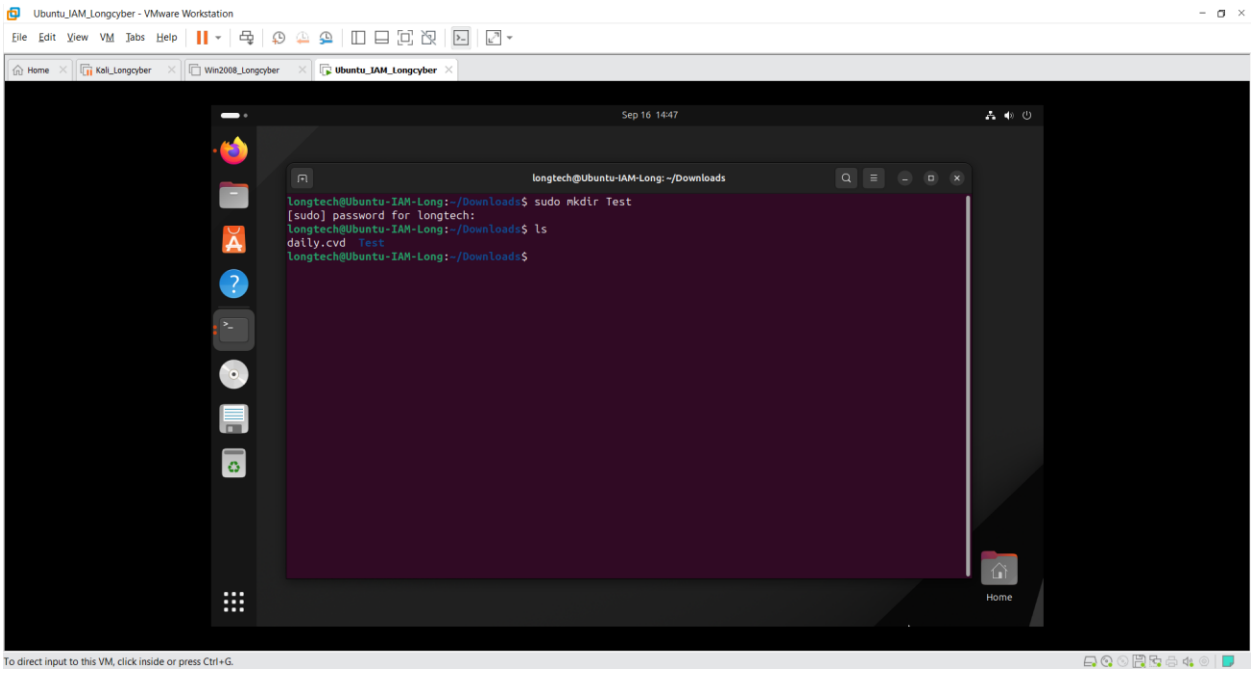




Execute the “clamscan” command with the “sudo” combination to scan the “Documents” directory “–recursive (including subdirectories)” in this format:

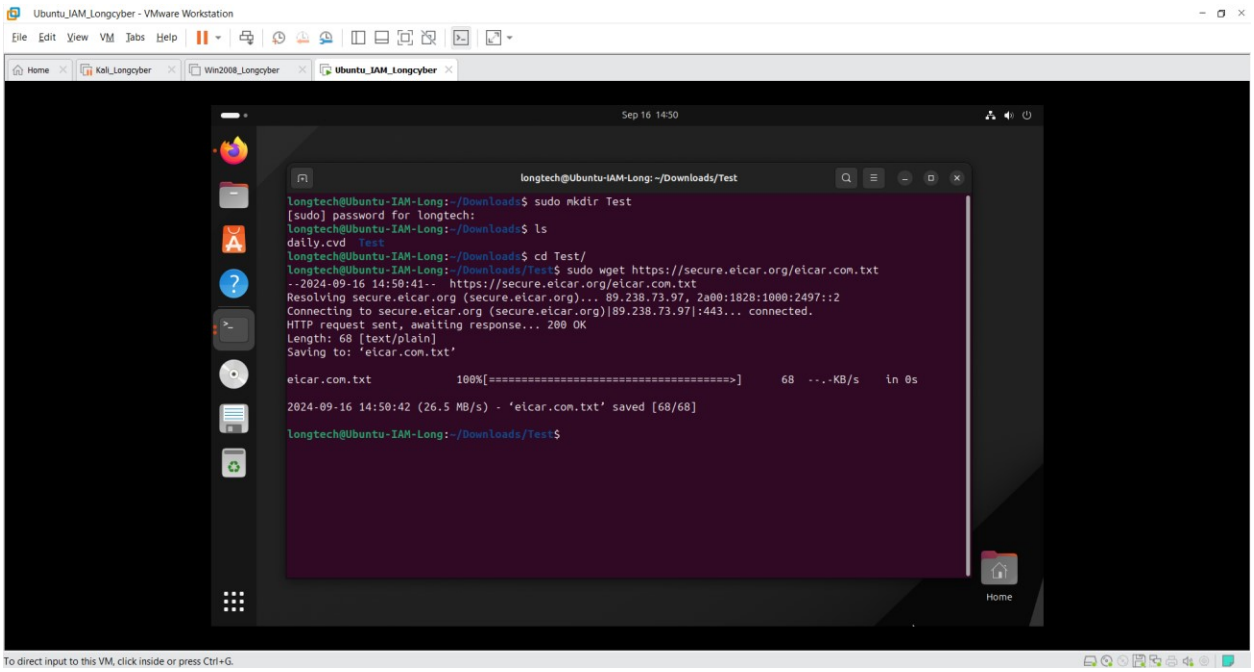
Create a Test folder for testing:

```
sudo mkdir Test
```



Download a malicious code on https://secure.eicar.org/ to test

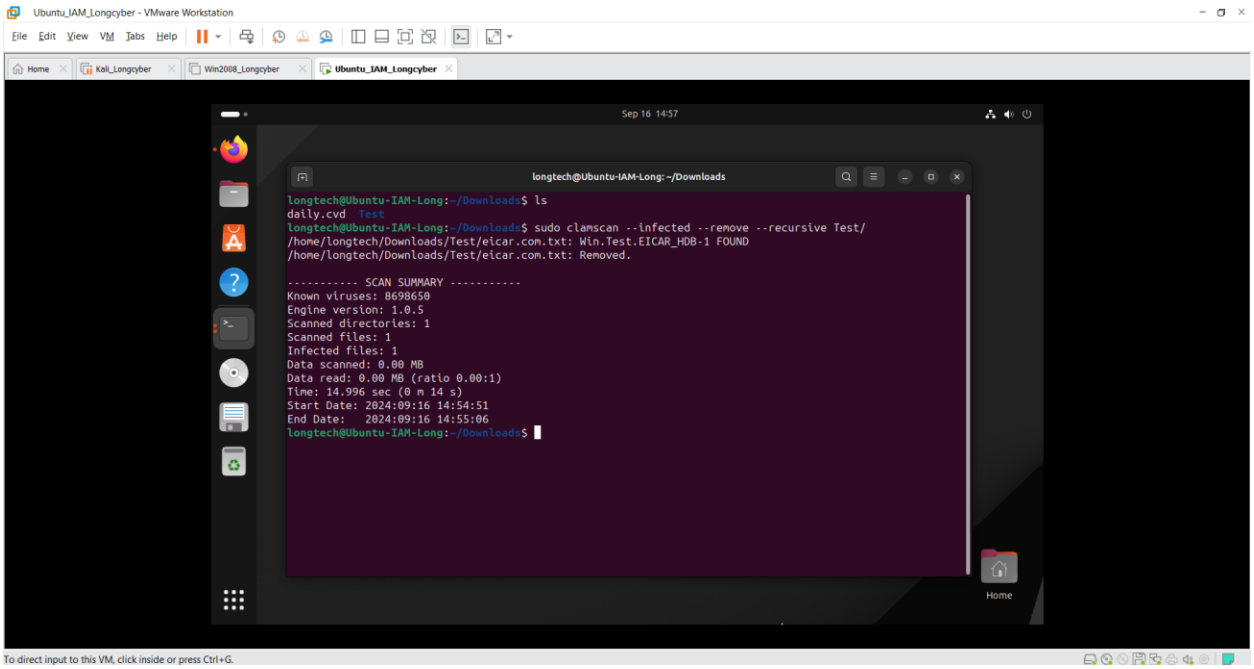
`sudo wget https://secure.eicar.org/eicar.com.txt`



Scan a Directory

Execute the “clamscan” command with the “sudo” combination to scan the “Documents” directory “–recursive (including subdirectories)” in this format:

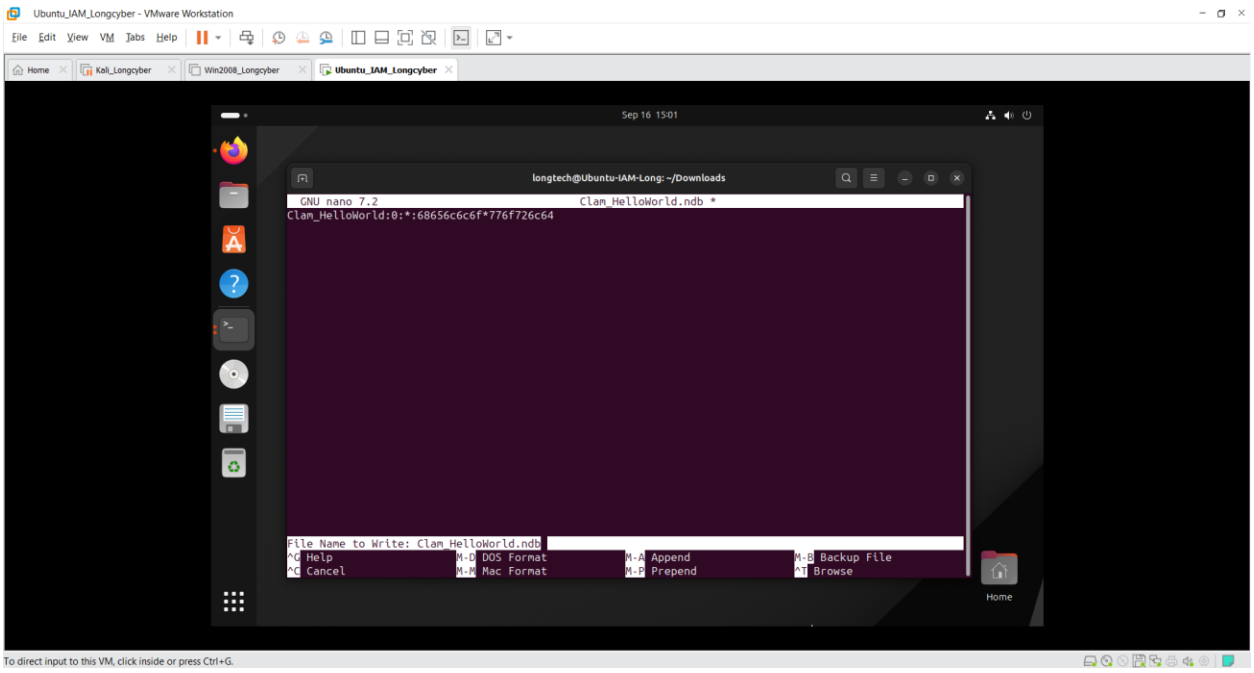
```
sudo clamscan --infected --remove --recursive Test/
```

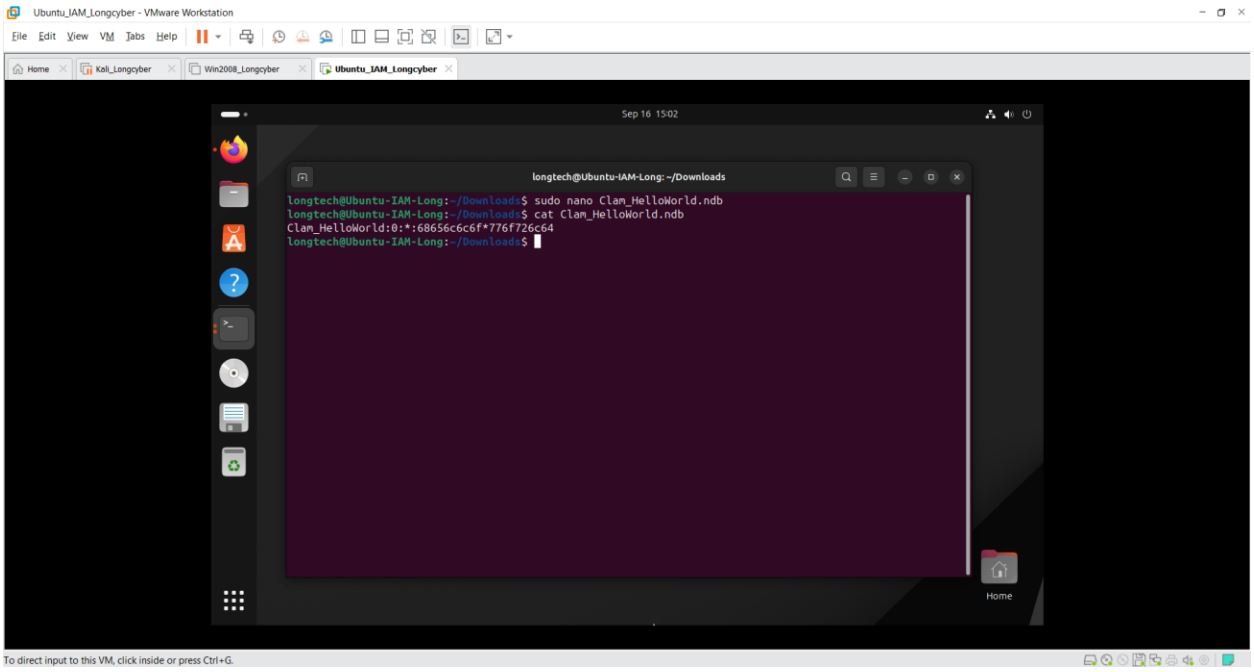


Use this command to create a signature for clamav:

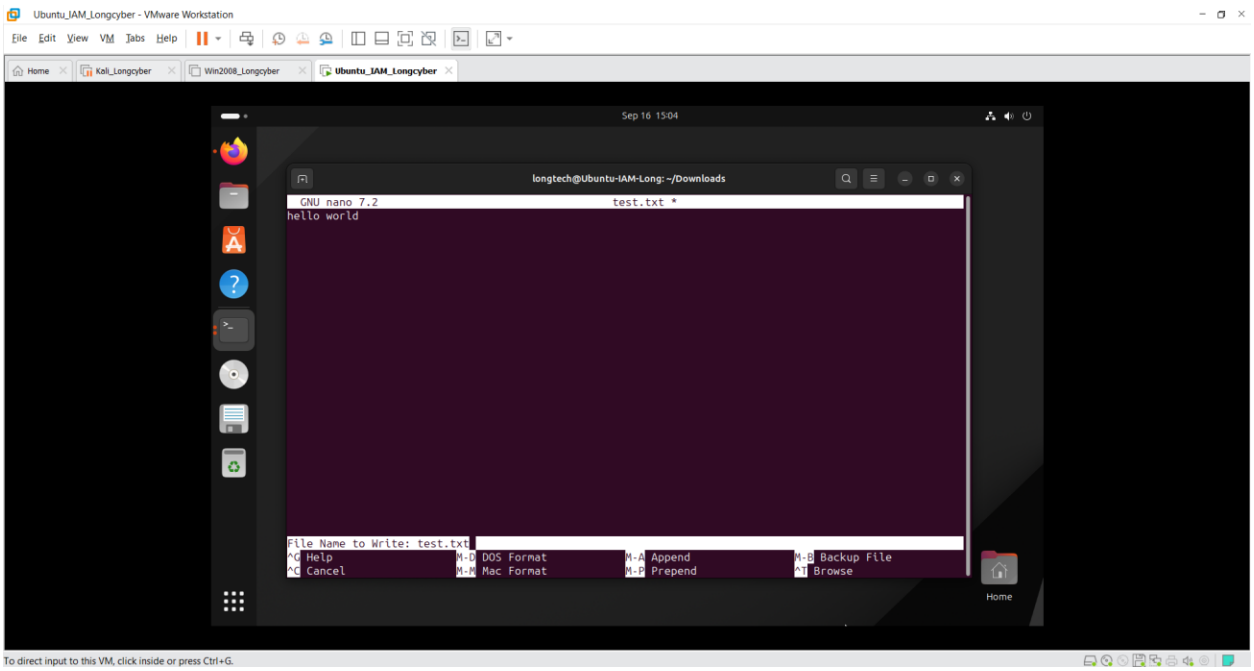
```
sudo nano Clam_HelloWorld.ndb
```

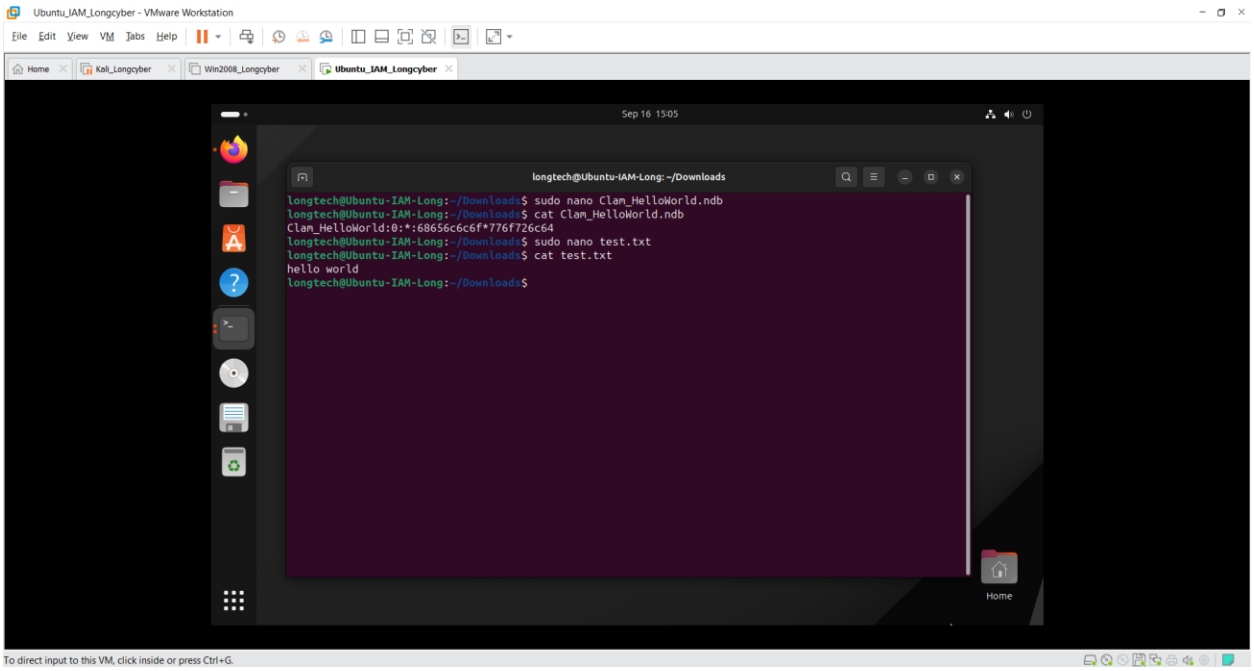
Then we enter this **Clam_HelloWorld:0:*:68656c6c6f*776f726c64** into the file Clam_HelloWorld.ndb. (This file will be signed so that ClamAV can scan the file, specifically if any txt file contains the words "hello" and the word "world" it will be considered to be injected with malicious code).





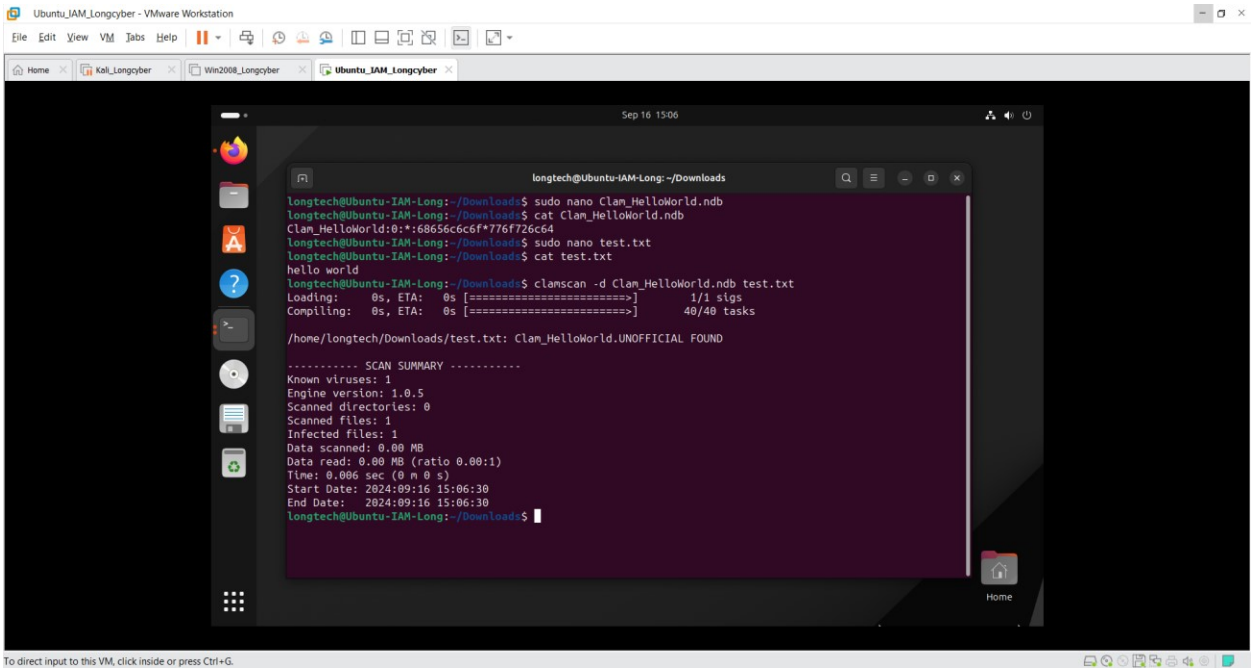
Next, we create the file test.txt with the command `sudo nano test.txt`, this file will contain 2 words "hello world".





Then we use this command to scan:

```
clamscan -d Clam_HelloWorld.ndb test.txt
```


Ubuntu_IAM_Longcyber - VMware Workstation
File Edit View VM Tabs Help
Home
Kali_Longcyber
Win2008_Longcyber
Ubuntu_IAM_Longcyber
Sep 16 15:06
longtech@Ubuntu-IAM-Long: ~/Downloads
longtech@Ubuntu-IAM-Long:~/Downloads$ sudo nano Clam_HelloWorld.ndb
longtech@Ubuntu-IAM-Long:~/Downloads$ cat Clam_HelloWorld.ndb
Clam_HelloWorld:0:*:68656c6c6f*776f726c64
longtech@Ubuntu-IAM-Long:~/Downloads$ sudo nano test.txt
longtech@Ubuntu-IAM-Long:~/Downloads$ cat test.txt
hello world
longtech@Ubuntu-IAM-Long:~/Downloads$ clamscan -d Clam_HelloWorld.ndb test.txt
Loading: 0s, ETA: 0s [=========>] 1/1 sigs
Compiling: 0s, ETA: 0s [=========>] 40/40 tasks
/home/longtech/Downloads/test.txt: Clam_HelloWorld.UNOFFICIAL FOUND
---------- SCAN SUMMARY ----------
Known viruses: 1
Engine version: 1.0.5
Scanned directories: 0
Scanned files: 1
Infected files: 1
Data scanned: 0.00 MB
Data read: 0.00 MB (ratio 0.00:1)
Time: 0.006 sec (0 m 0 s)
Start Date: 2024:09:16 15:06:30
End Date: 2024:09:16 15:06:30
longtech@Ubuntu-IAM-Long:~/Downloads$
Home
To direct input to this VM, click inside or press Ctrl+G.

LAB 4

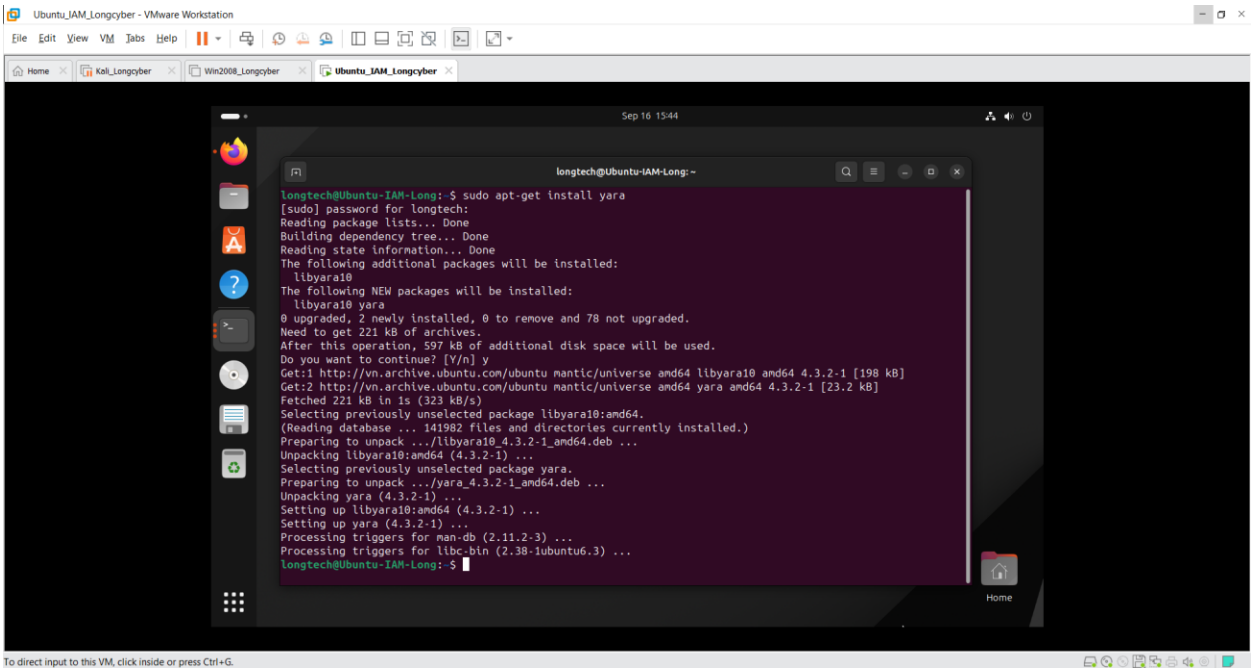
Detect Malware Capabilities with YARA

- YARA is a popular tool that provides a robust language.

- It is used to examine the suspected files/directories and match strings as is defined in the YARA rules with the file.

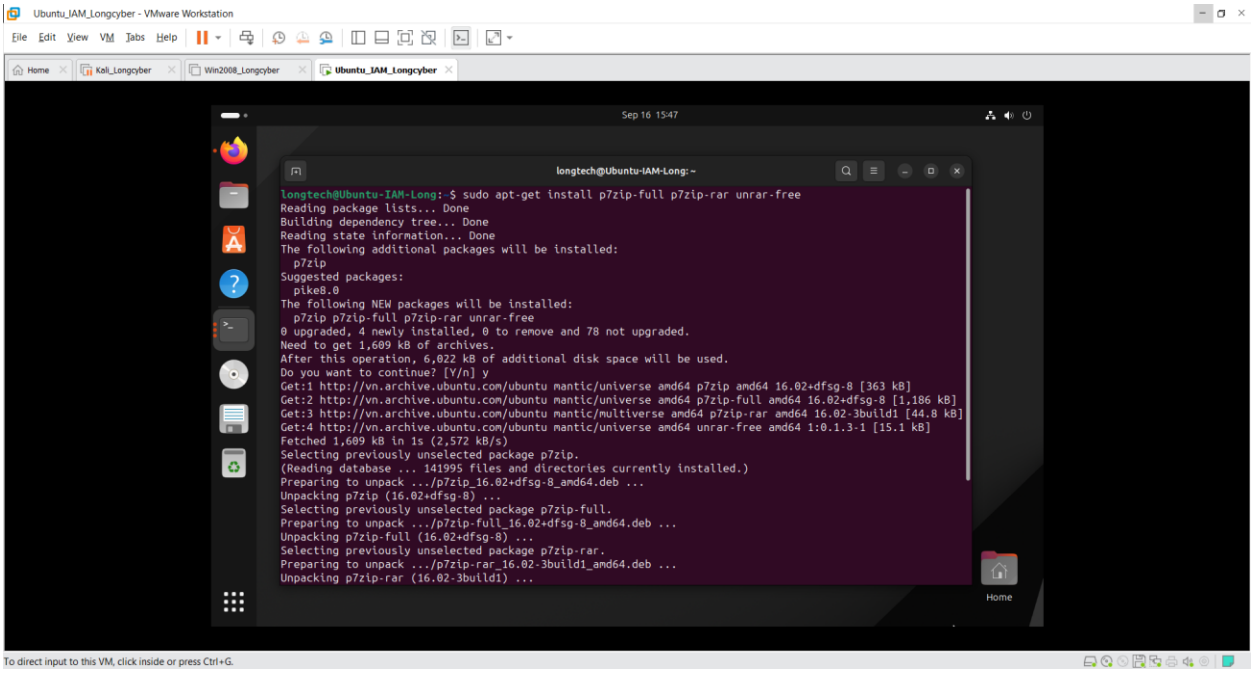
Install YARA

```
Sudo apt-get install yara
```



Install p7zip-full:

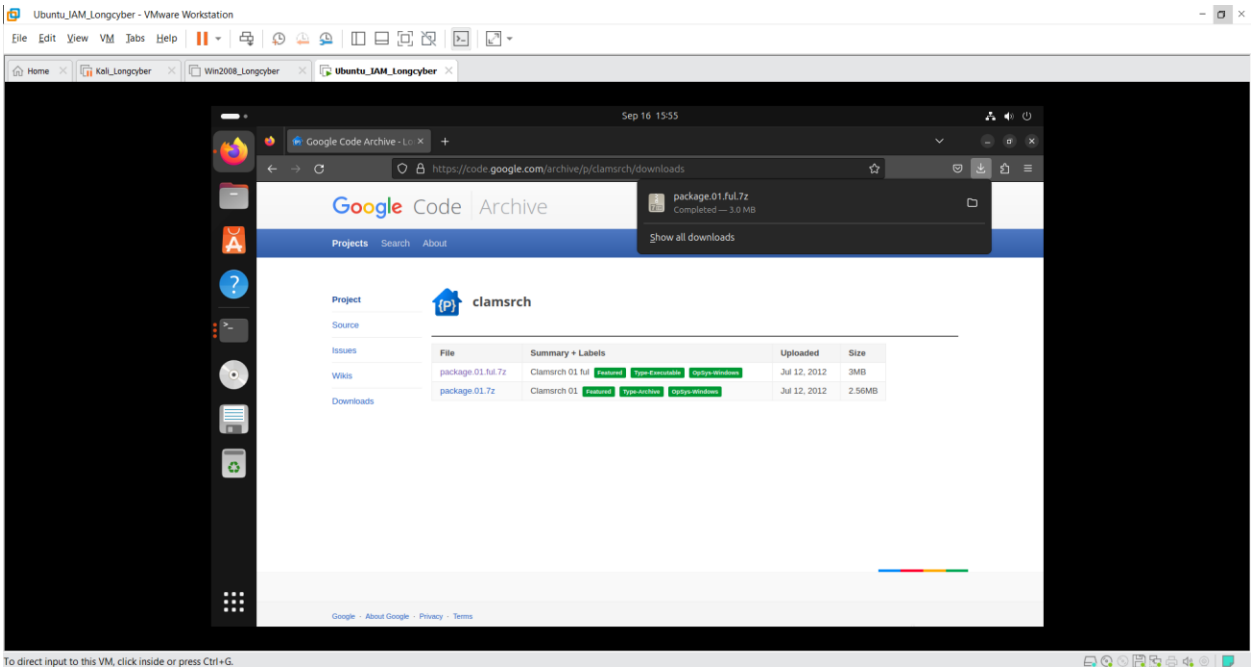
```
sudo apt-get install p7zip-full p7zip-rar unrar-free
```

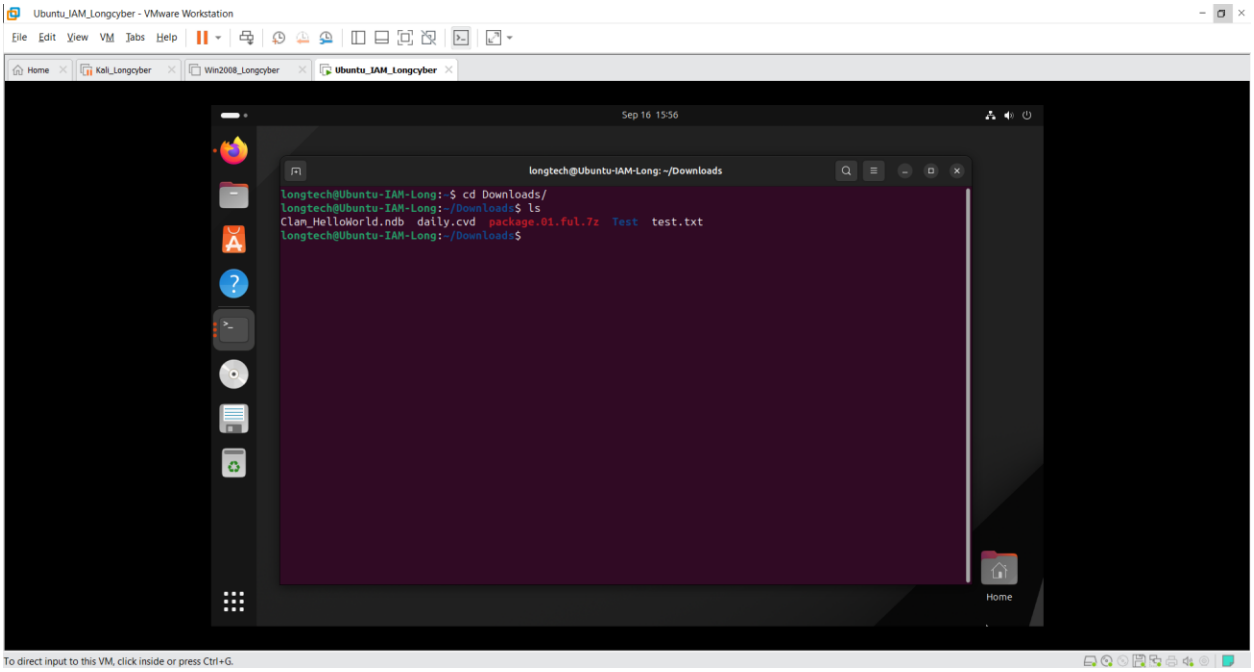


Download package.01.ful.7z:

https://code.google.com/archive/p/clamsrch/downloads

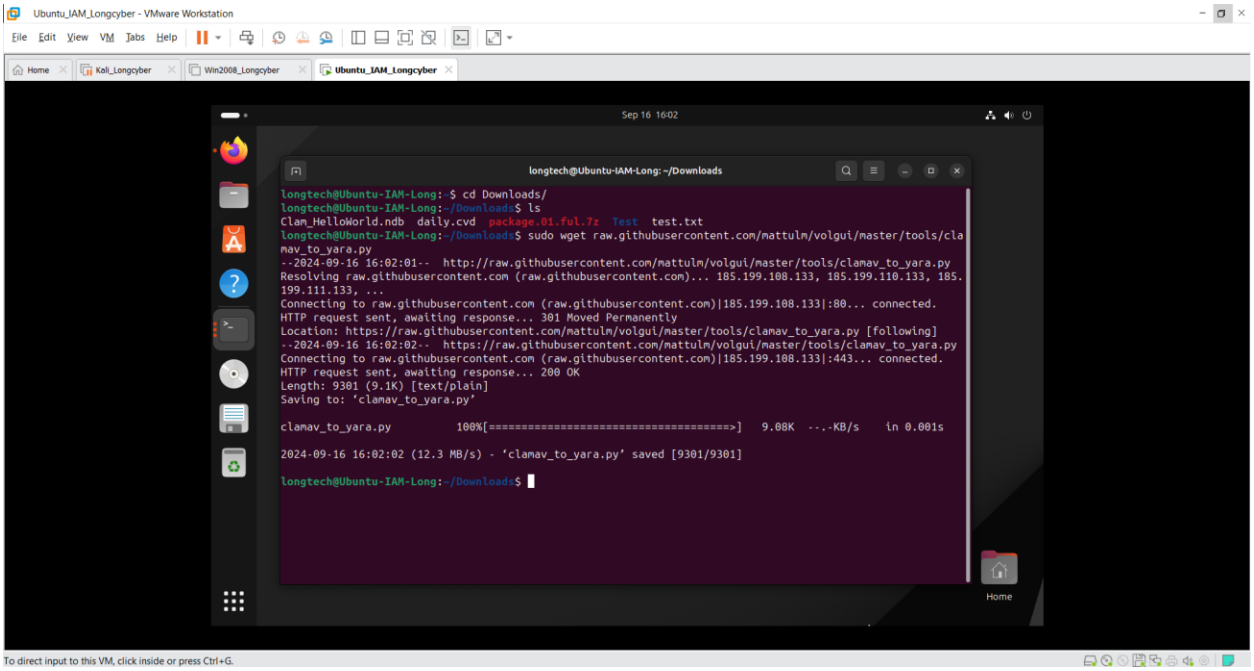
(https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/clamsrch/package.01.ful.7z)





Download file clam_to_yara.py:

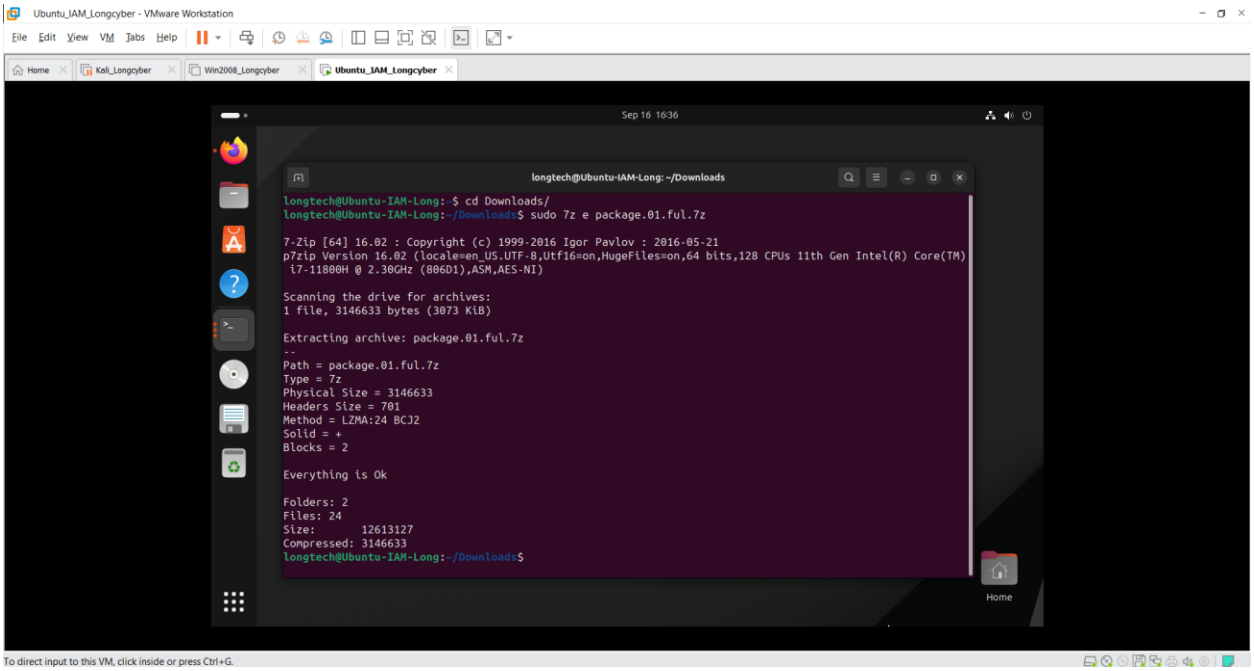
`sudo wget`
`raw.githubusercontent.com/mattulm/volgui/master/tools/clamav_to_yara.py`

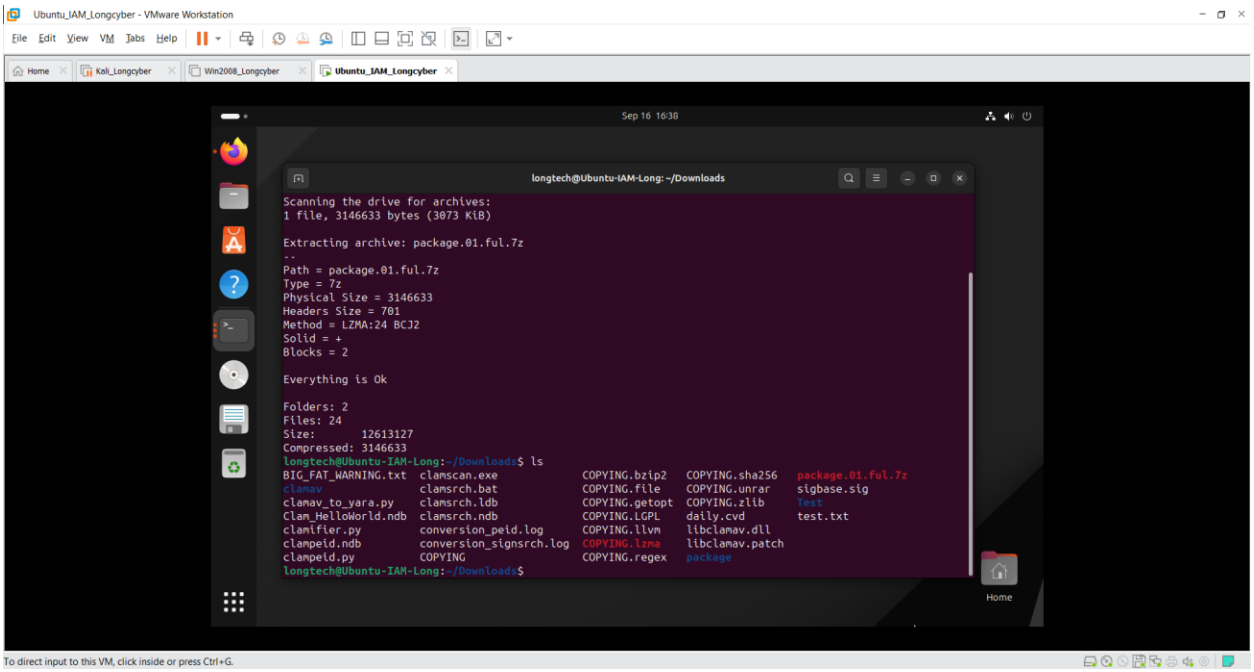


Convert file clamav to yara:

Unzip the package.01.ful.7z file with the command:

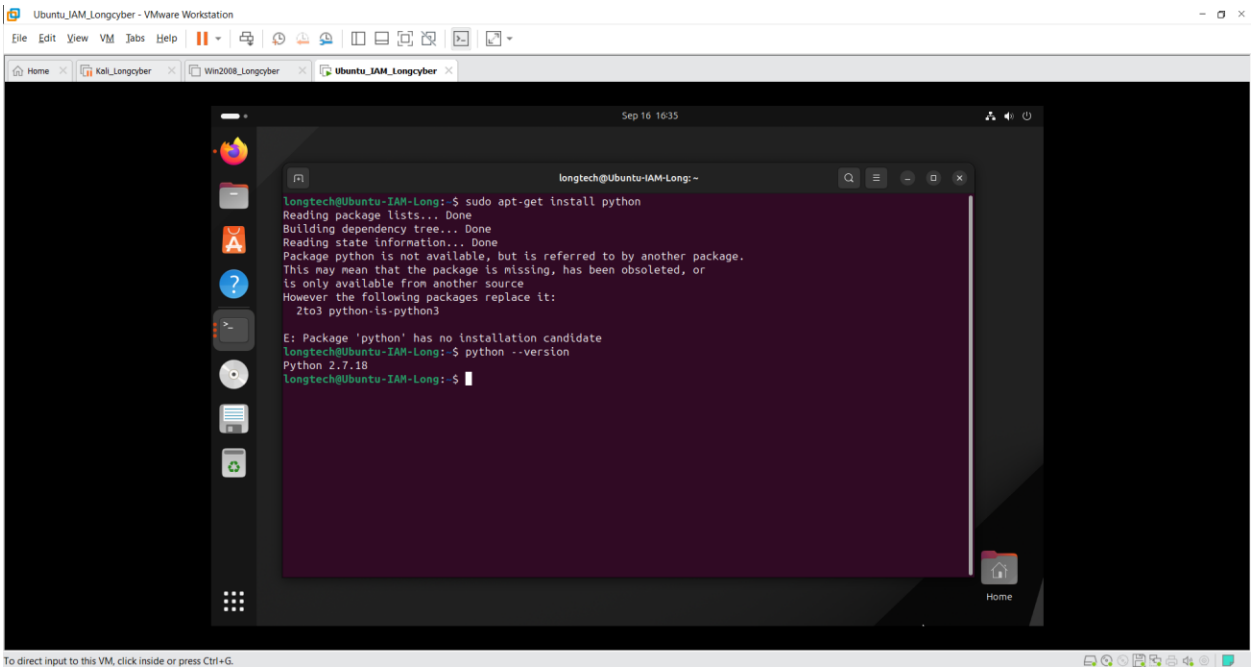
```
sudo 7z e package.01.ful.7z
```



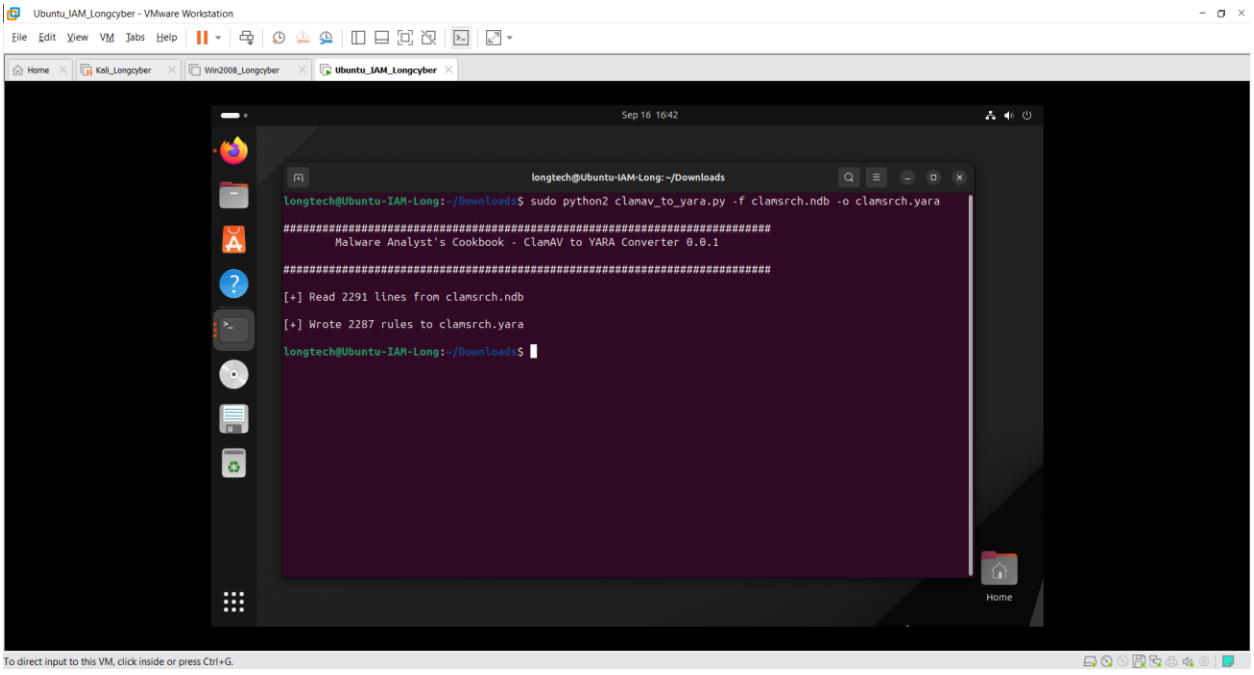


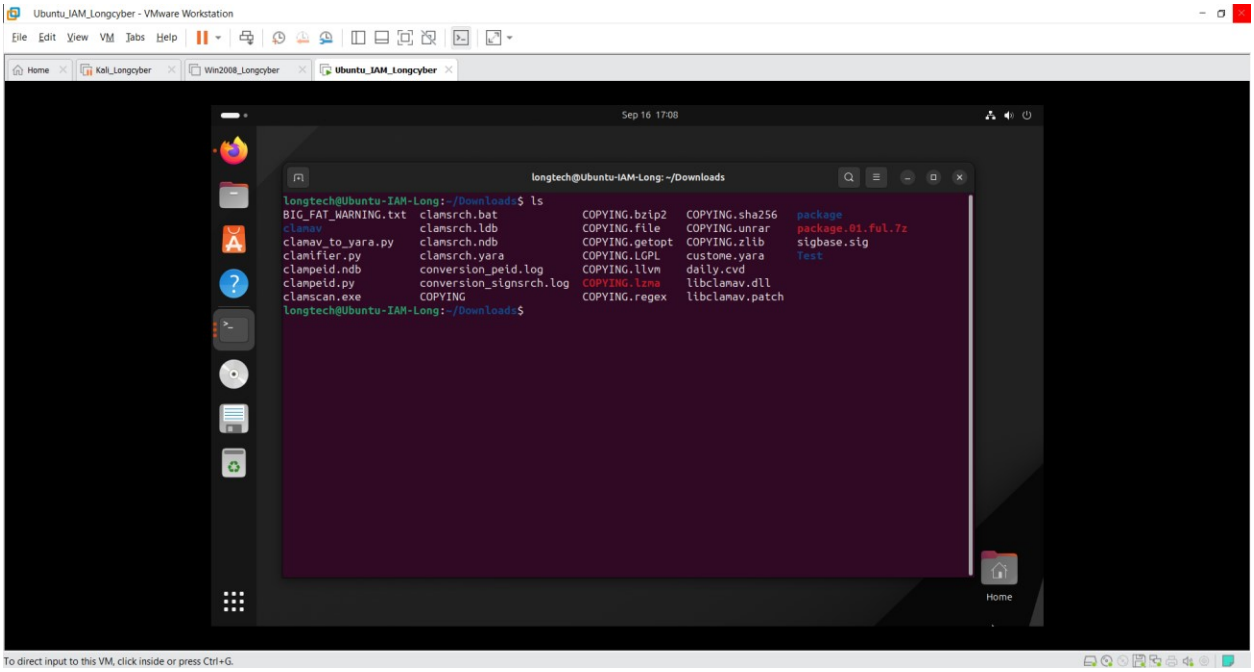
You must first install python2 because the clamav_to_yara.py file is written in python2.

```
sudo apt-get install python2
```



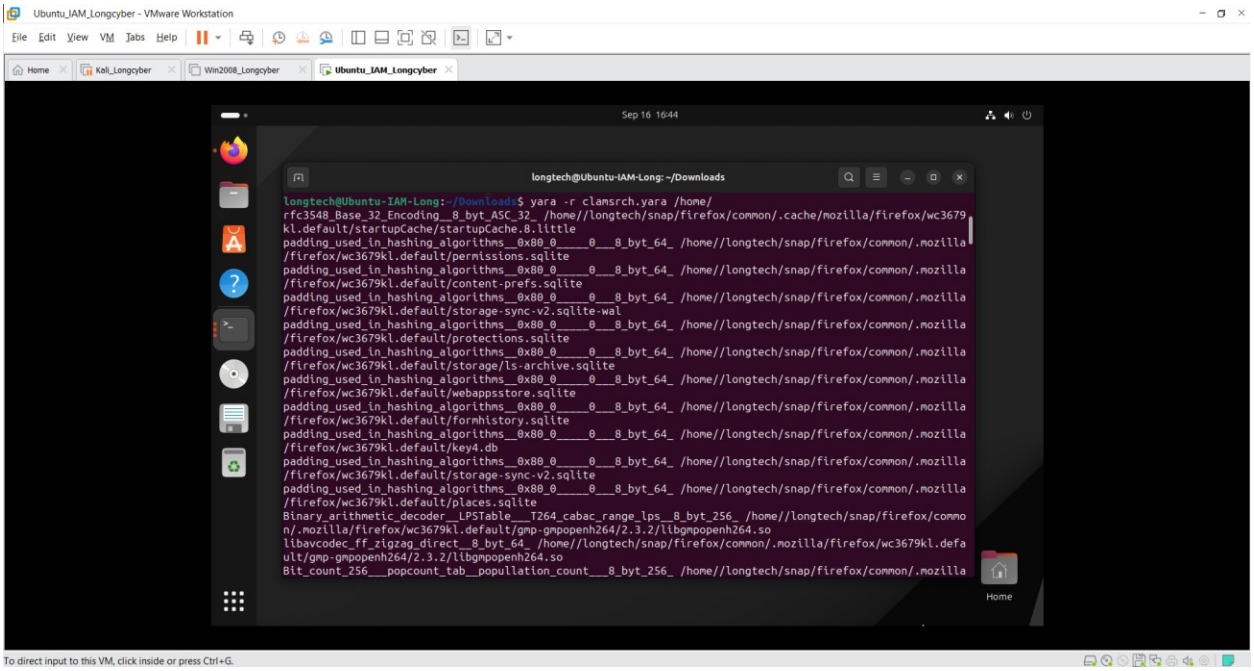

```
sudo python2 clamav_to_yara.py -f clamsrch.ndb -o clamsrch.yara
```

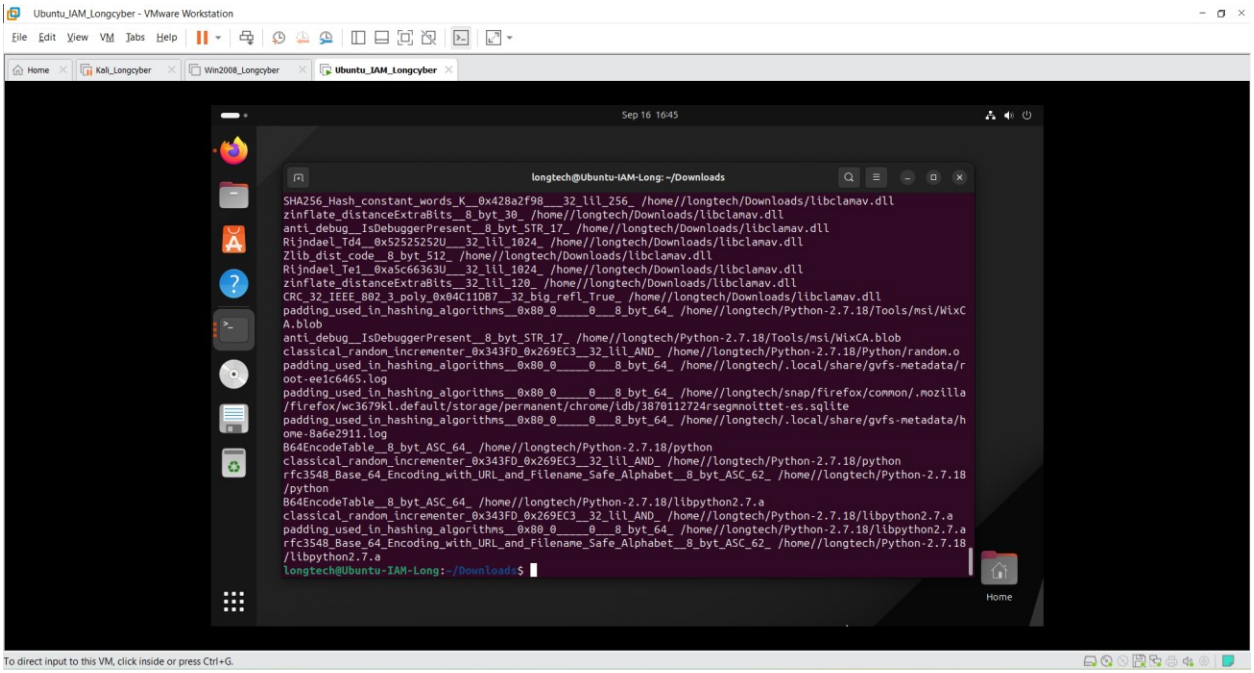




Start scanning with the yara

```
yara -r clamsrch.yara /home/
```



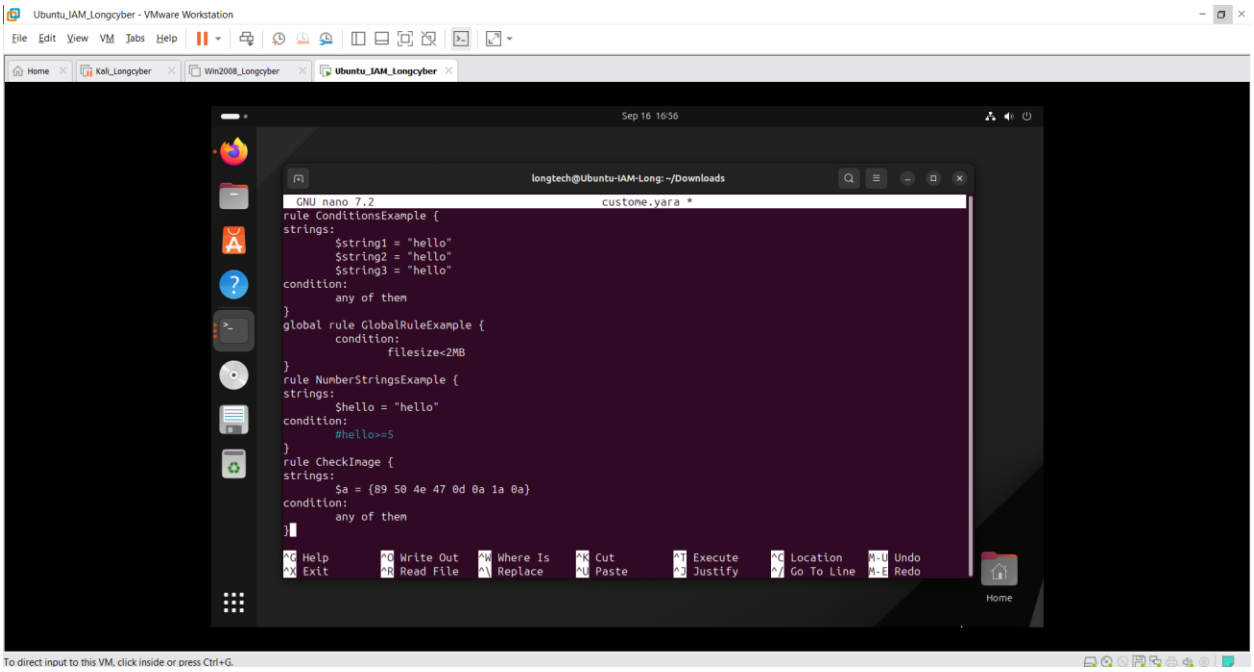


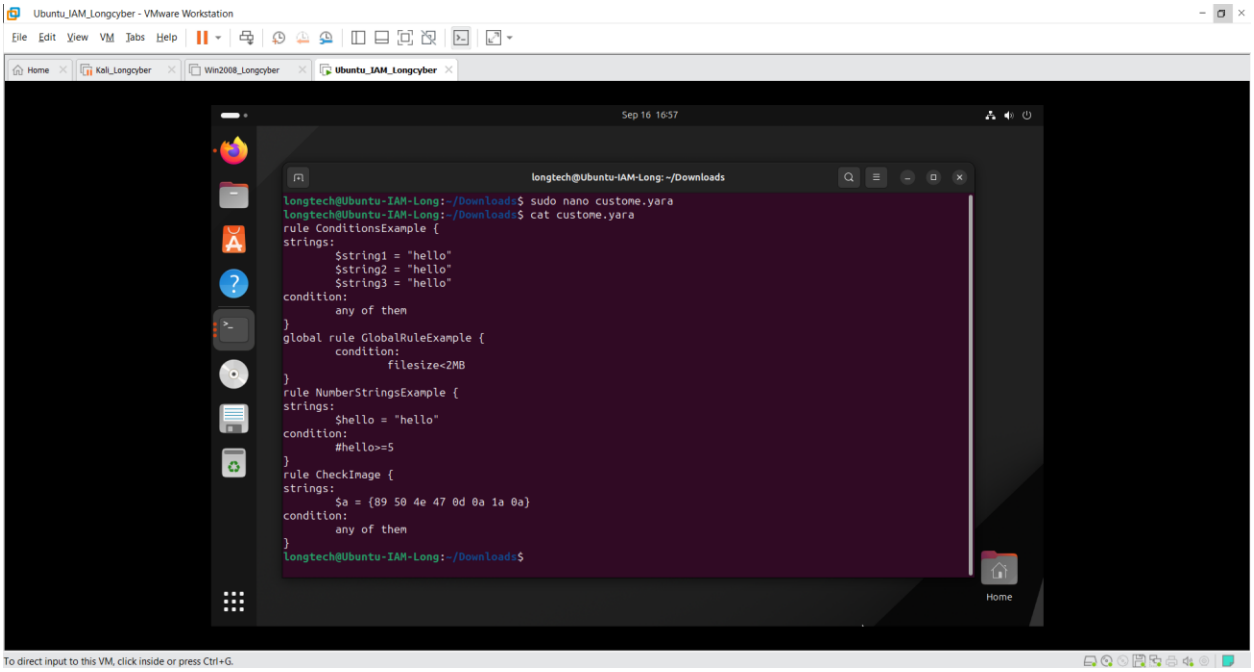
Create a new rule file called custome.yara

```
sudo nano custome.yara
```

```
rule ConditionsExample {
strings:
        $string1 = "hello"
        $string2 = "hello"
        $string3 = "hello"
condition:
        any of them
}
global rule GlobalRuleExample {
        condition:
                filesize < 2MB
}
rule NumberStringsExample {
strings:
```

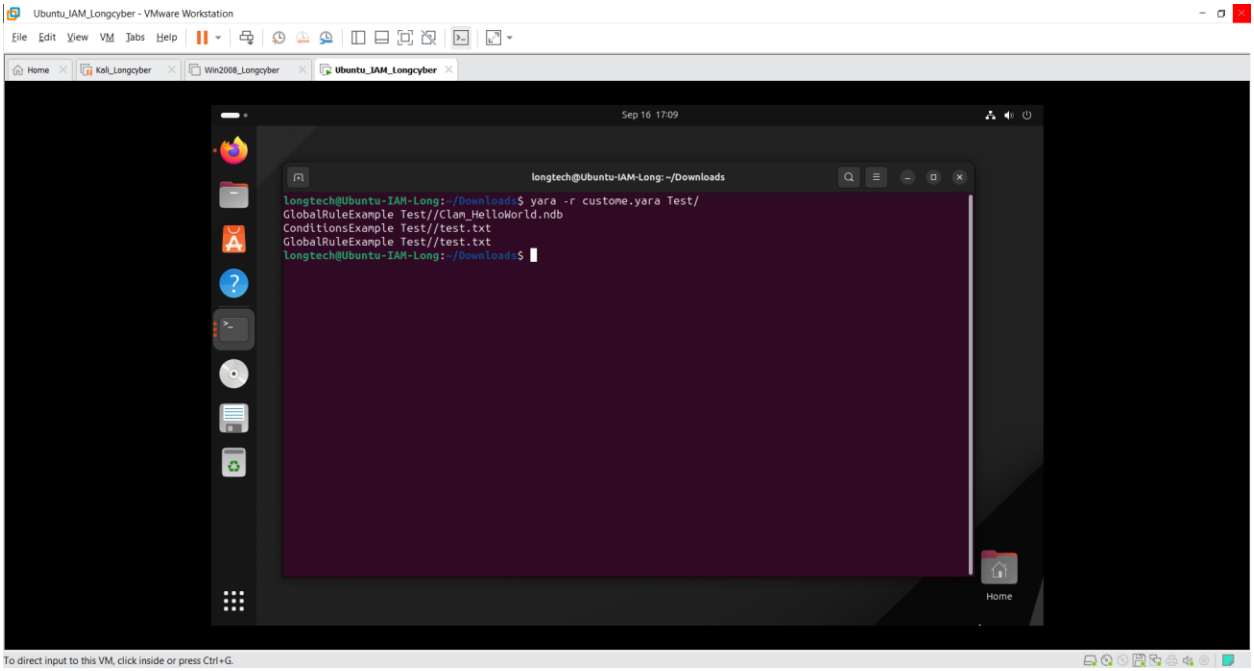
```
        $hello = "hello"
condition:
        #hello >=5
}
rule CheckImage {
strings:
        $a = {89 50 4e 47 0d 0a 1a 0a}
condition:
        any of them
}
```





Test yara rules:

```
yara -r custome.yara Test/
```



Here we see yara's report for the rule we created as follows: ConditionExample means that yara has detected that the test.txt file matches the rule we provided and contains the string “hello” in there. In addition, other files will match yara's GlobalRuleExample