

Lab 2-3

Install and Use ClamAV

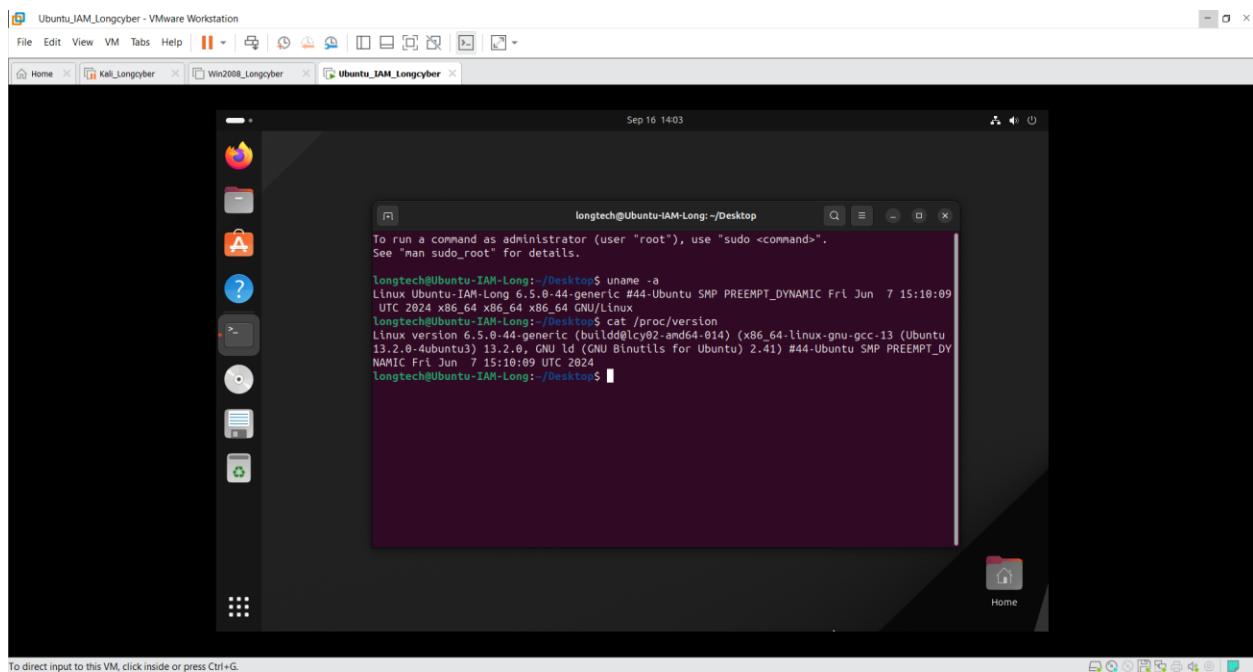
Course Name: IAM302

Student Name: Phạm Thành Long

Instructor Name: Mai Hoàng Đinh

Lab Due Date: 16/09/2024

OS Install:

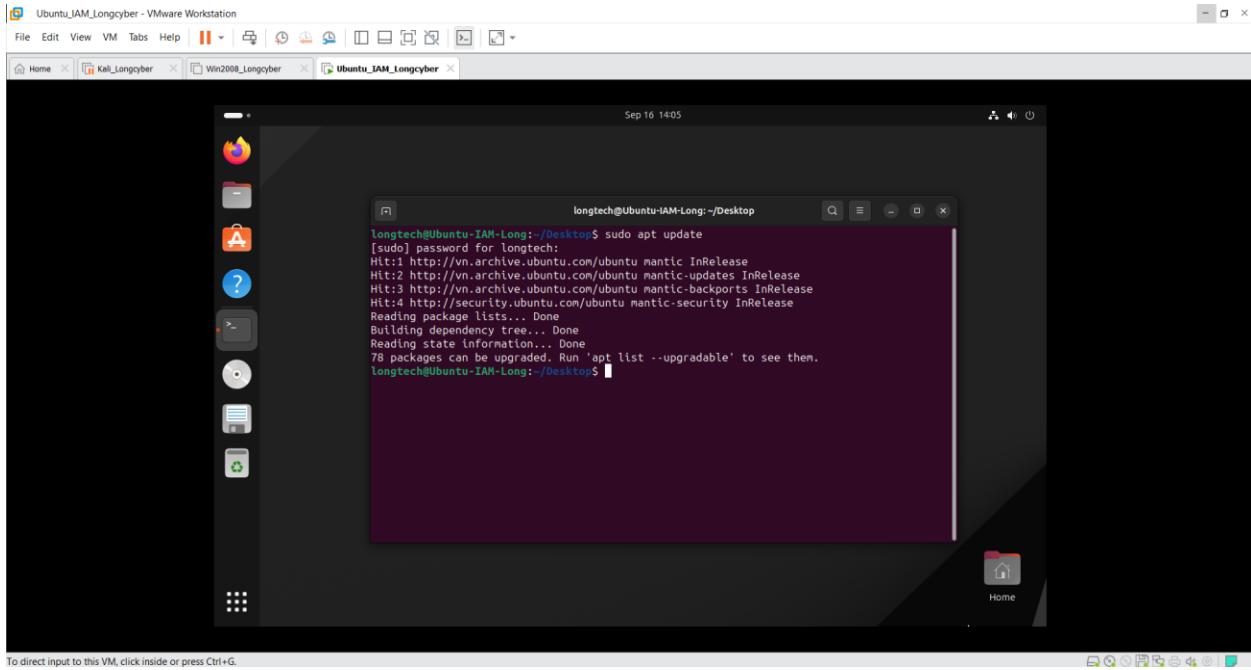


ClamAV is a well-reputed free and open-source antivirus software tool. It provides a command line interface that quickly scans the Linux system against viruses and malware attacks. The “ClamAV” helps scan the important part of Linux, i.e., mail gateways and emails directly affecting the network.

Install ClamAV on Ubuntu

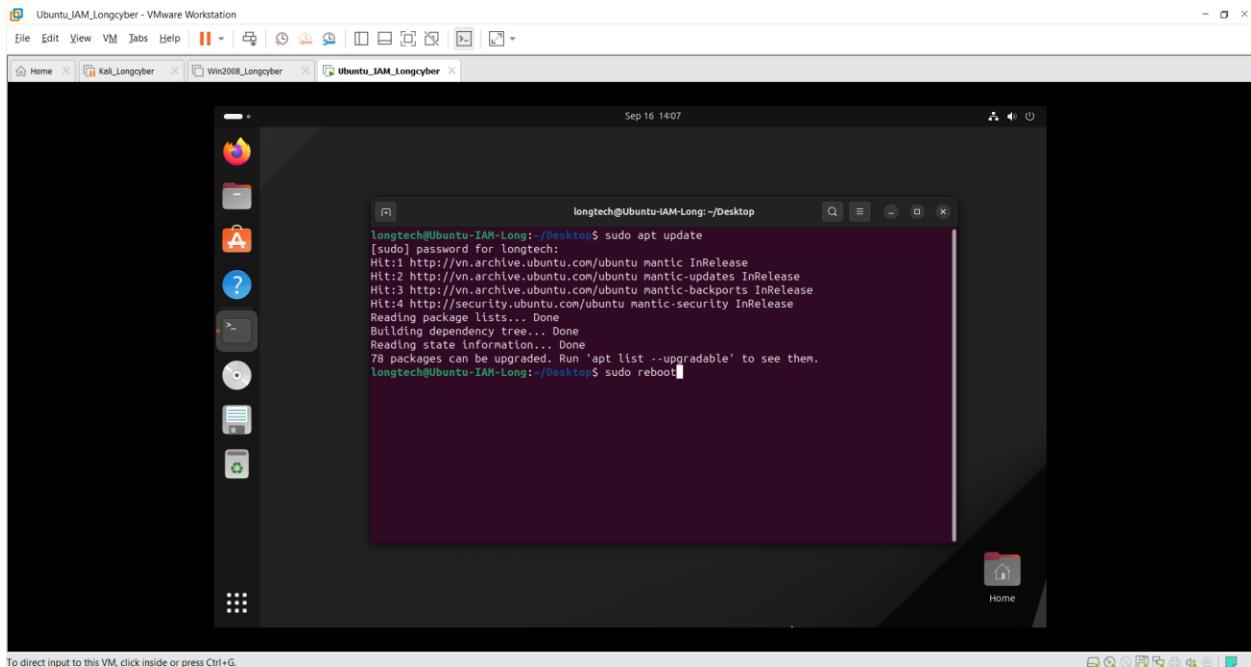
Step 1: Update the Repository

```
sudo apt-get update
```



We need to reboot after the update is complete

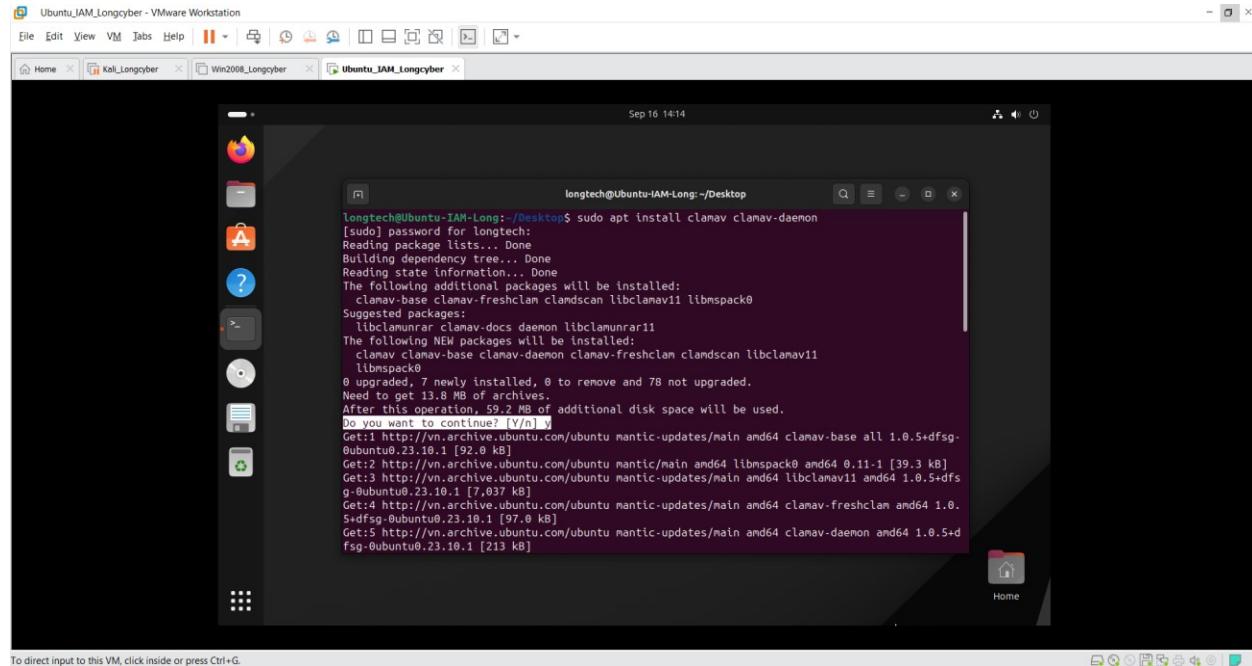
`sudo reboot`



Step 2: Install ClamAV

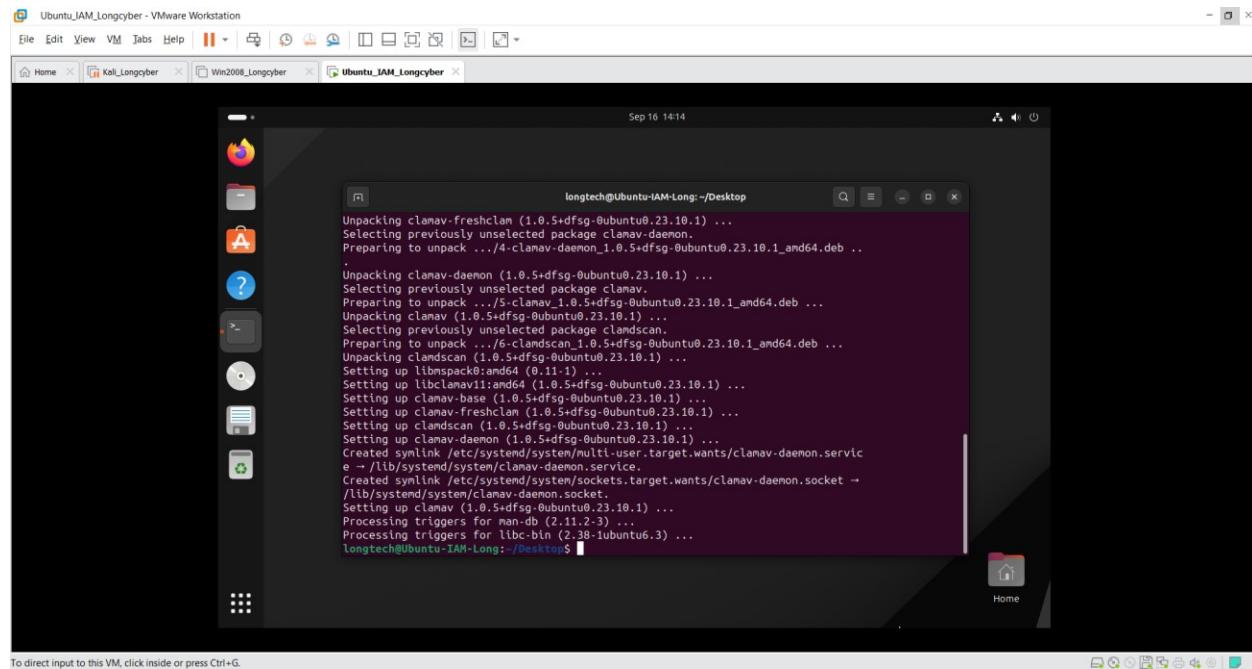
Install the “ClamAV” application alongside the “clamav-daemon” from the standard repository of Ubuntu using the default “apt” package manager:

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



The screenshot shows a terminal window titled "Ubuntu.IAM_Longcyber - VMware Workstation". The command "sudo apt install clamav clamav-daemon" is being run. The output shows the package manager reading lists, building dependency trees, and installing additional packages like clamav-base, clamav-freshclam, and libclamav11. It also lists suggested packages such as libclamunrar and clamav-docs. The process installs 7 new packages, upgrades 0, removes 0, and does not upgrade 78. A total of 13.8 MB of disk space will be used. The user is prompted with "Do you want to continue? [Y/n] y". The terminal window has a dark background with light-colored text. The desktop environment includes icons for a browser, file manager, and system tools.

To direct input to this VM, click inside or press Ctrl+G.



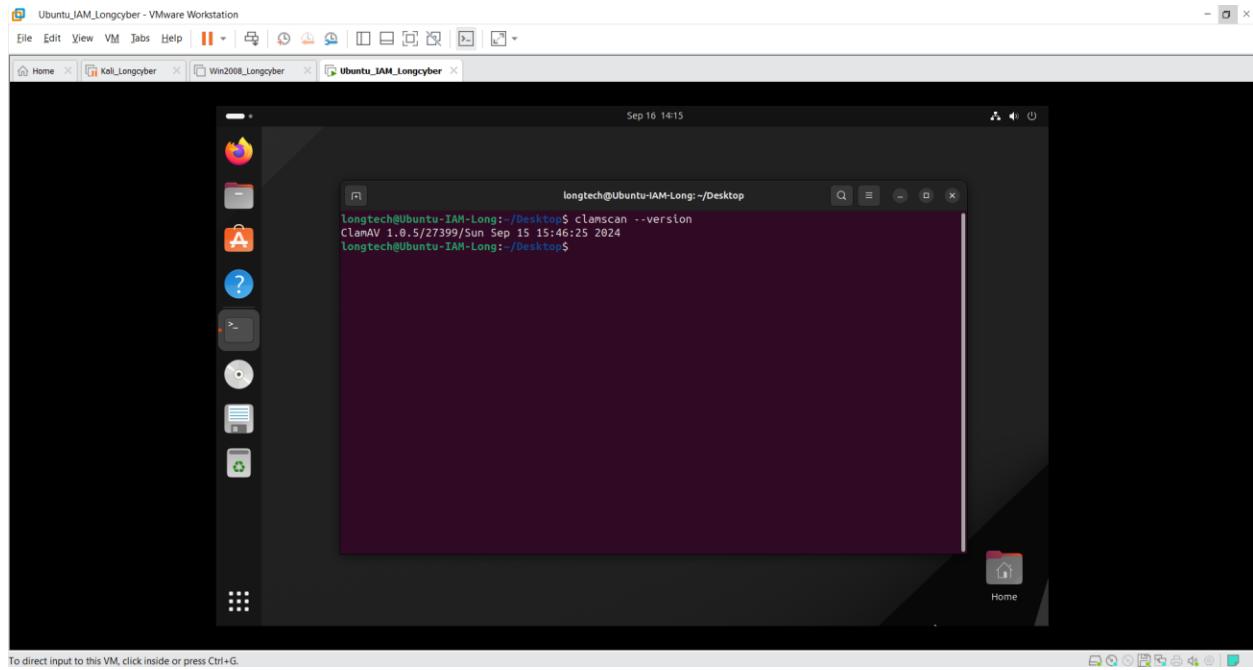
The screenshot shows a terminal window titled "Ubuntu.IAM_Longcyber - VMware Workstation". The command "dpkg -i" is being run to install the previously downloaded ClamAV packages. The output shows the unpacking of clamav-freshclam, clamav-daemon, clamav, clamdscan, and libclamav11. It also creates symbolic links for the clamav-daemon service and socket. The terminal window has a dark background with light-colored text. The desktop environment includes icons for a browser, file manager, and system tools.

To direct input to this VM, click inside or press Ctrl+G.

Step 3: Verify ClamAV

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

```
clamscan --version
```



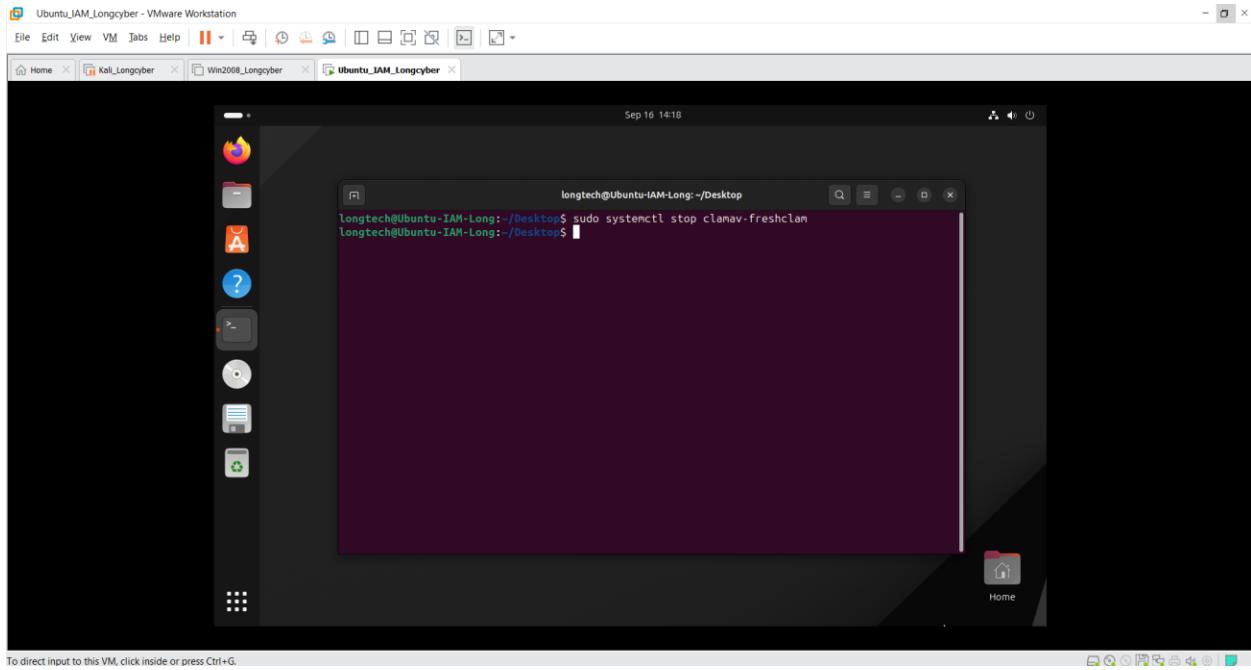
The “ClamAV” works on a signature database that identifies the malware. It requires updation regularly that ensures the application is up to date for protection against the latest threats.

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
```



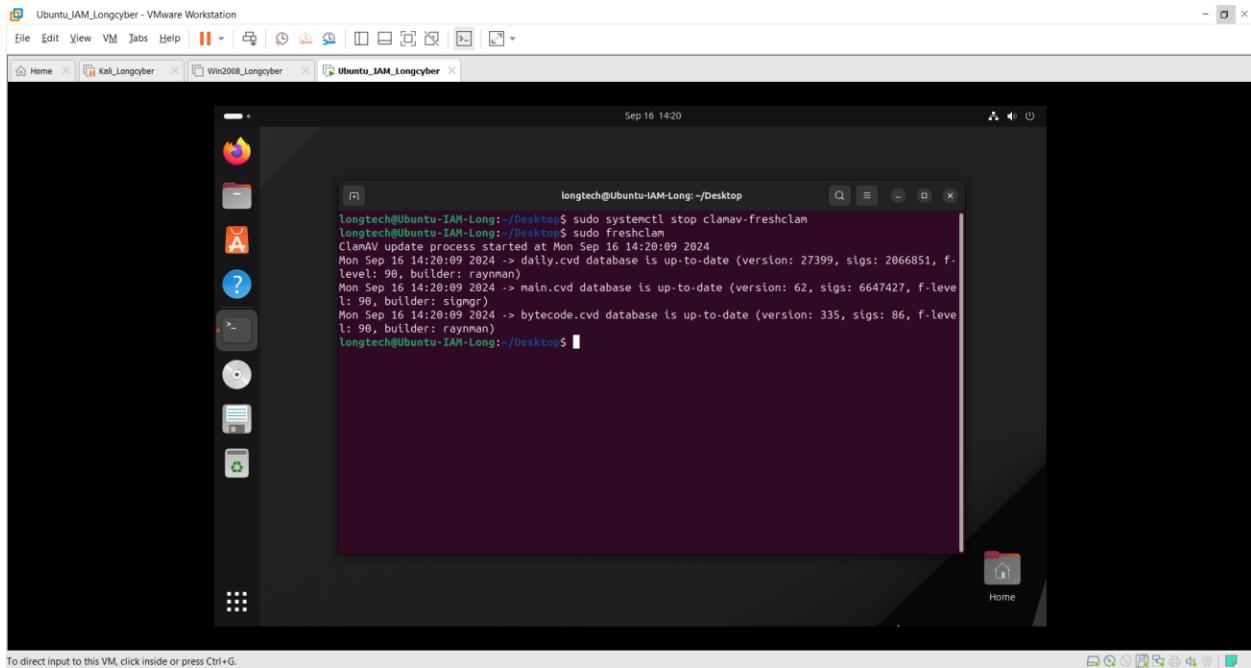
```
longtech@Ubuntu-IAM-Long:~/Desktop$ sudo systemctl stop clamav-freshclam
longtech@Ubuntu-IAM-Long:~/Desktop$
```

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
```

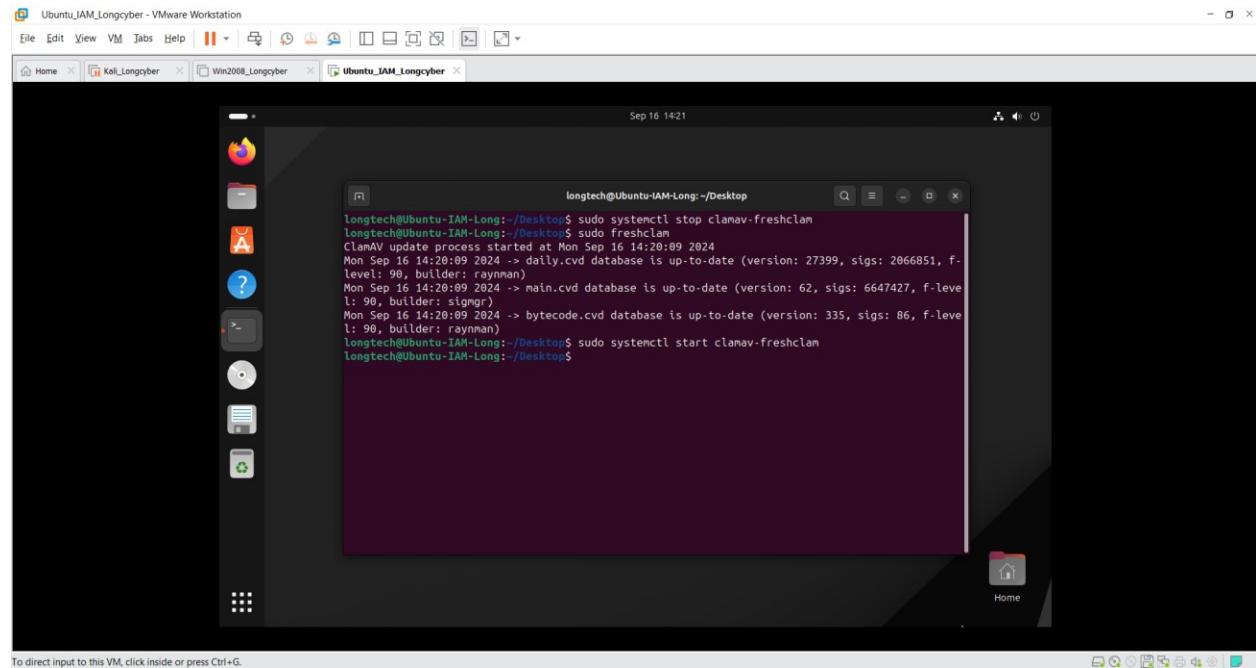


```
longtech@Ubuntu-IAM-Long:~/Desktop$ sudo systemctl stop clamav-freshclam
longtech@Ubuntu-IAM-Long:~/Desktop$ sudo freshclam
ClamAV update process started at Mon Sep 16 14:20:09 2024
Mon Sep 16 14:20:09 2024 -> daily.cvd database is up-to-date (version: 27399, sigs: 2066851, f-level: 90, builder: raynman)
Mon Sep 16 14:20:09 2024 -> main.cvd database is up-to-date (version: 62, sigs: 6647427, f-level: 90, builder: sigmgr)
Mon Sep 16 14:20:09 2024 -> bytecode.cvd database is up-to-date (version: 335, sigs: 86, f-level: 90, builder: raynman)
longtech@Ubuntu-IAM-Long:~/Desktop$
```

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
```

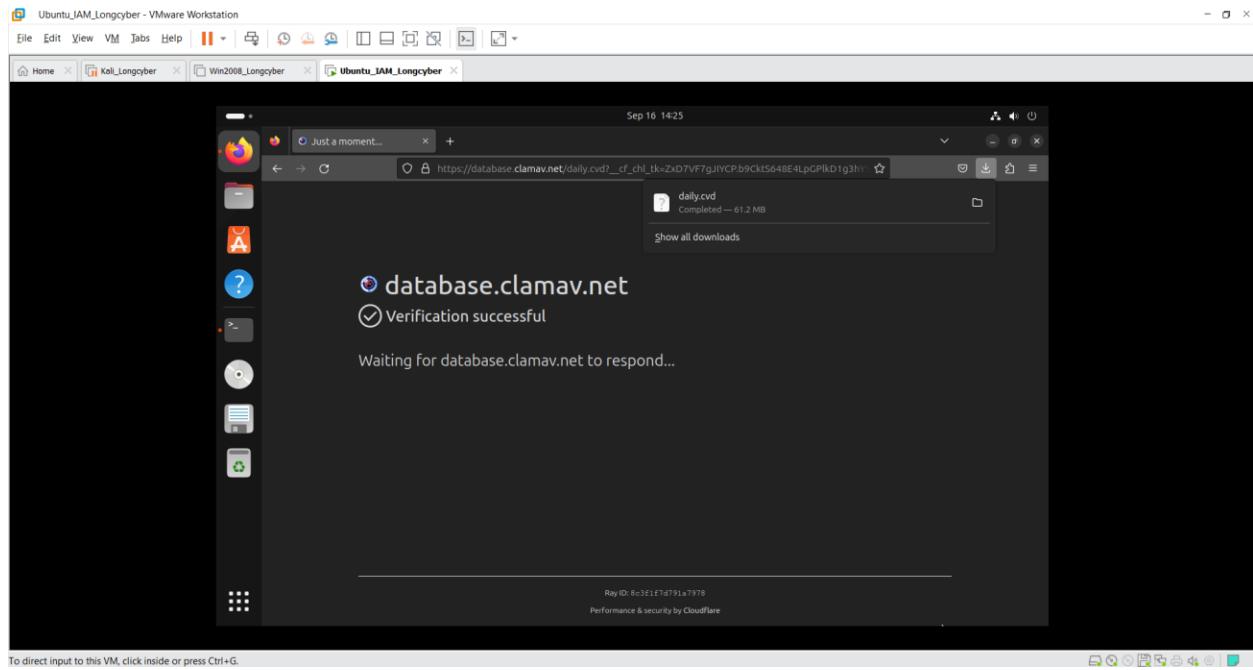


```
longtech@Ubuntu-IAM-Long:~/Desktop$ sudo systemctl stop clamav-freshclam
longtech@Ubuntu-IAM-Long:~/Desktop$ sudo freshclam
ClamAV update process started at Mon Sep 16 14:20:09 2024
Mon Sep 16 14:20:09 2024 -> daily.cvd database is up-to-date (version: 27399, sigs: 2066851, f-level: 90, builder: rayman)
Mon Sep 16 14:20:09 2024 -> main.cvd database is up-to-date (version: 62, sigs: 6647427, f-level: 90, builder: sigmgr)
Mon Sep 16 14:20:09 2024 -> bytecode.cvd database is up-to-date (version: 335, sigs: 86, f-level: 90, builder: rayman)
longtech@Ubuntu-IAM-Long:~/Desktop$ sudo systemctl start clamav-freshclam
longtech@Ubuntu-IAM-Long:~/Desktop$
```

Download Updates Using Official Website (Second Method)

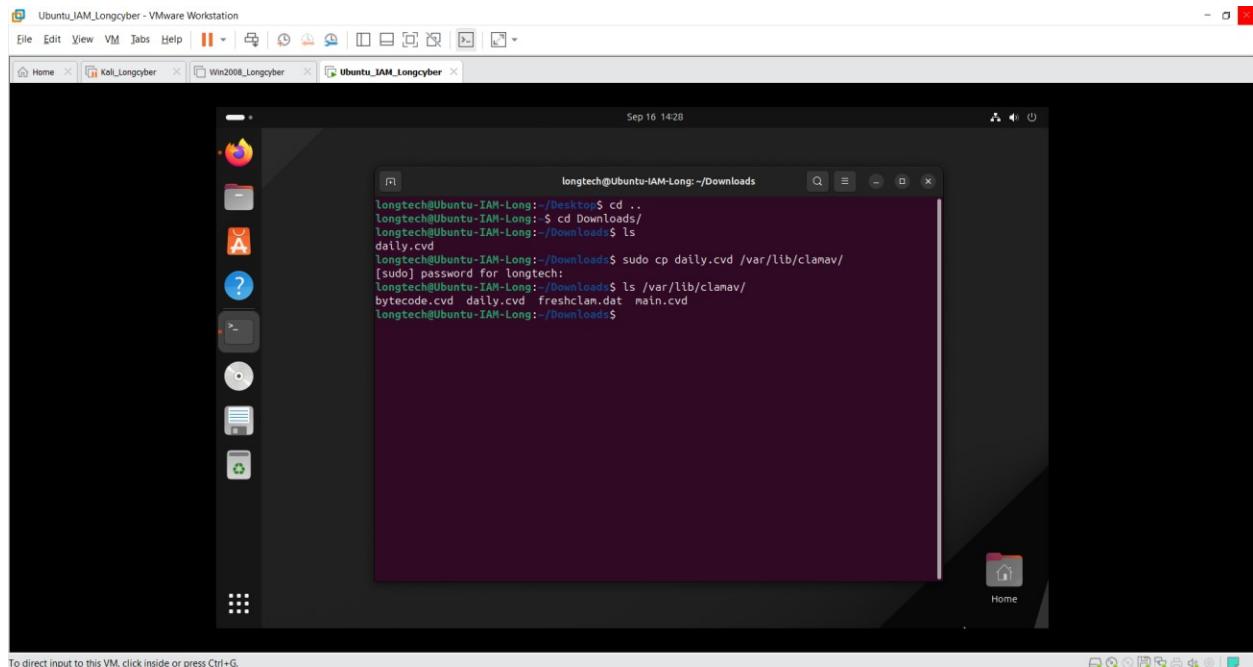
Another way is to download the “ClamAV” database from its official website <https://database.clamav.net/daily.cvd>

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
```

To direct input to this VM, click inside or press Ctrl+G.

```
longtech@Ubuntu-IAM-Long:~/Downloads$ clamscan --help
Clam AntiVirus: Scanner 1.0.5
By The ClamAV Team: https://www.clamav.net/about.html#credits
(C) 2022 Cisco Systems, Inc.

clamscan [options] [file/directory/-]

--help           -h      Show this help
--version        -V      Print version number
--verbose        -v      Be verbose
--archive-verbose -a      Show filenames inside scanned archives
--debug          -d      Enable libclamav's debug messages
--quiet          -q      Only output error messages
--stdout         -s      Write to stdout instead of stderr. Does not a
ffect 'debug' messages.
--no-summary     -n      Disable summary at end of scanning
--infected       -i      Only print infected files
--suppress-ok-results -o  Skip printing OK files
--bell           -b      Sound bell on virus detection

--tempdir=DIRECTORY Create temporary files in DIRECTORY
--leave-temp[=yes/no(*)]
--gen-json[=yes/no(*)]
). For testing & development use ONLY.
JSON will be printed if --debug is enabled.
A JSON file will be dropped to the temp director
y if --leave-temp is enabled.
```

To direct input to this VM, click inside or press Ctrl+G.

```
longtech@Ubuntu-IAM-Long:~/Downloads$ clamscan --help
file (**)
  --max-dir-recursion=#n      Maximum directory recursion level
  --max-embeddedpe=#n        Maximum size file to check for embedded PE
  --max-htmlnormalize=#n    Maximum size of HTML file to normalize
  --max-htmlnotags=#n       Maximum size of normalized HTML file to scan
  --max-scriptnormalize=#n  Maximum size of script file to normalize
  --max-ziptypercgs=#n     Maximum size zip to type reanalyze
  --max-partitions=#n       Maximum number of partitions in disk image to
be scanned
  --max-iconsize=#n          Maximum number of icons in PE file to be scan
ned
  --max-rechwp3=#n          Maximum recursive calls to MP3 parsing funct
ion
  --pcre-match-limit=#n     Maximum calls to the PCRE match function.
  --pcre-recmatch-limit=#n  Maximum recursive calls to the PCRE match fun
ction.
  --pcre-max-filename=#n   Maximum size file to perform PCRE subsig matc
hing.
  --disable-cache           Disable caching and cache checks for hash sum
s of scanned files.

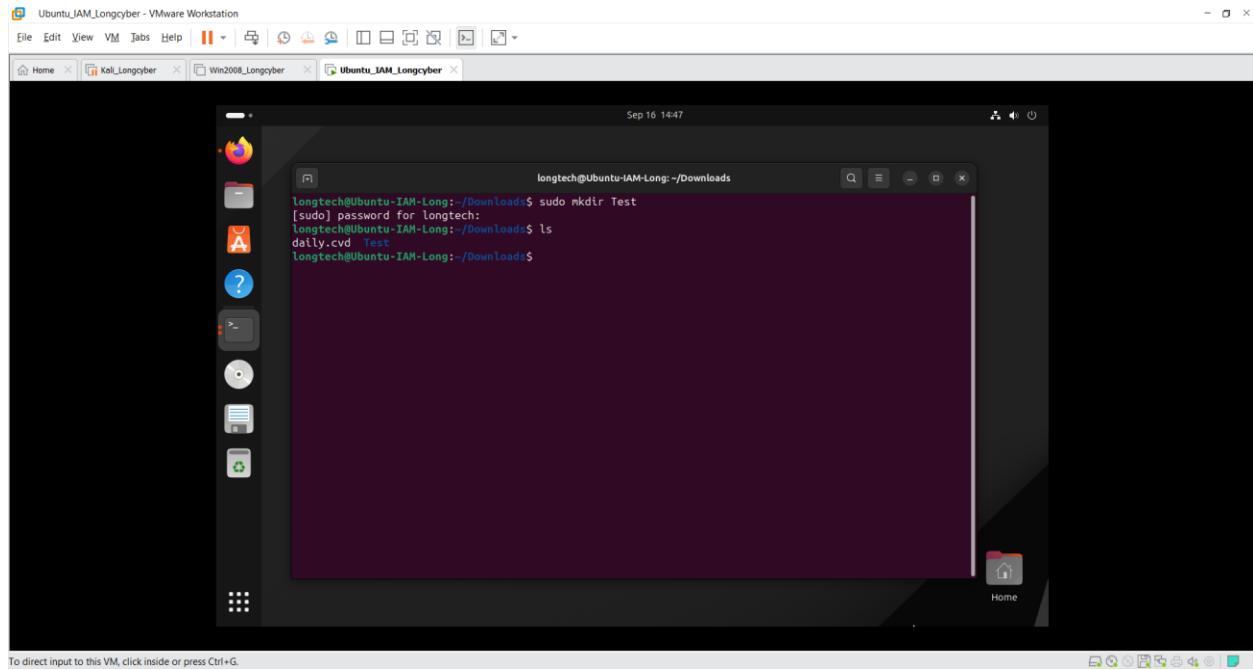
Pass in - as the filename for stdin.

(*) Default scan settings
(**) Certain files (e.g. documents, archives, etc.) may in turn contain other
files inside. The above options ensure safe processing of this kind of data.
```

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

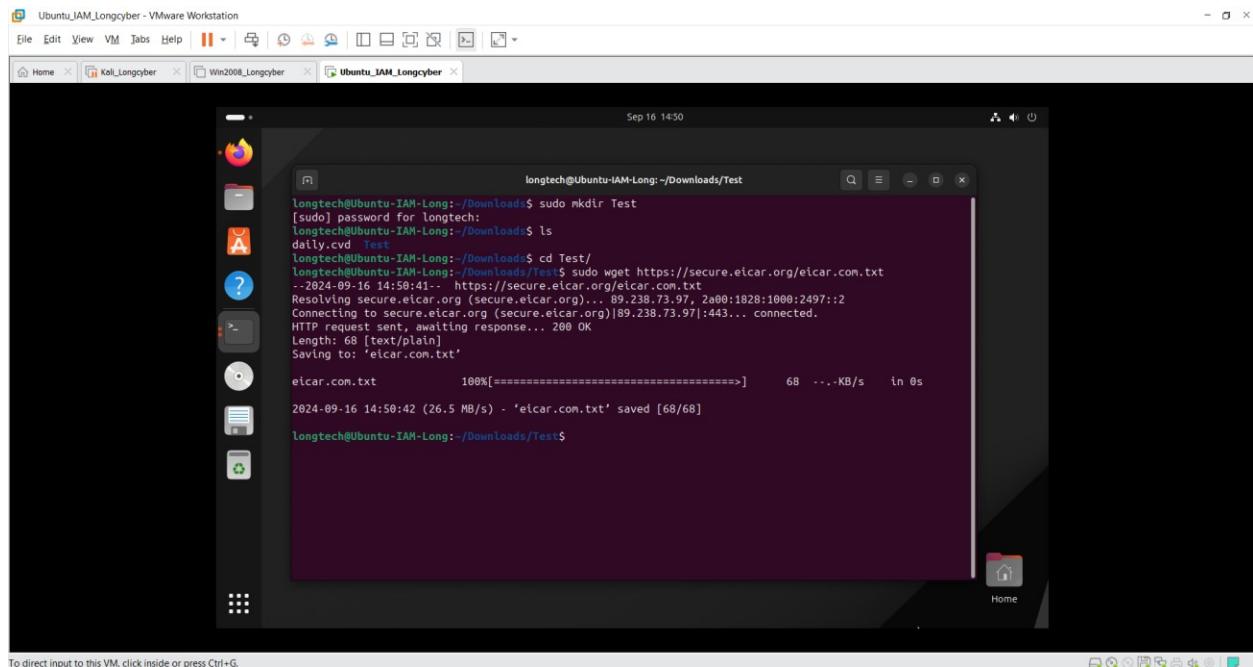
Create a Test folder for testing:

```
sudo mkdir Test
```



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

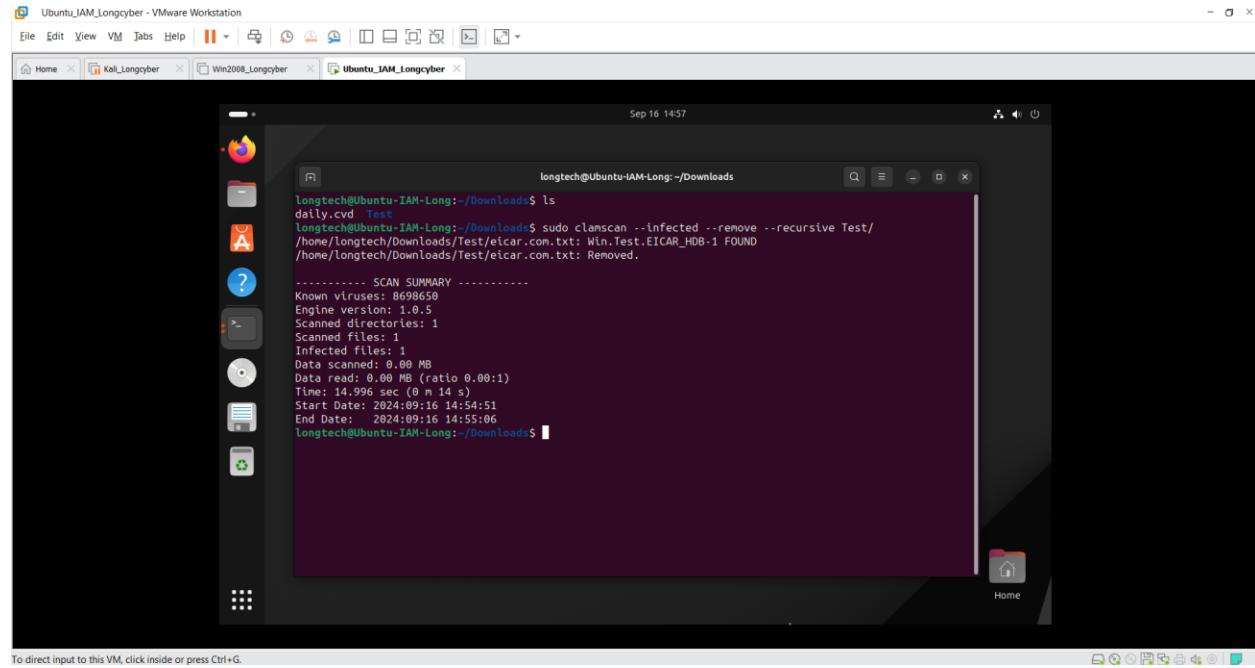
```
sudo wget https://secure.eicar.org/eicar.com.txt
```



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/
```



The screenshot shows a terminal window titled "Ubuntu_IAM_Longcyber - VMware Workstation". The terminal is running on an Ubuntu system. The command entered is "sudo clamscan --infected --remove --recursive Test/". The output of the command is displayed in the terminal window, showing that it found and removed a file named "Win.test.EICAR_H0B-1" located in the "Test/" directory.

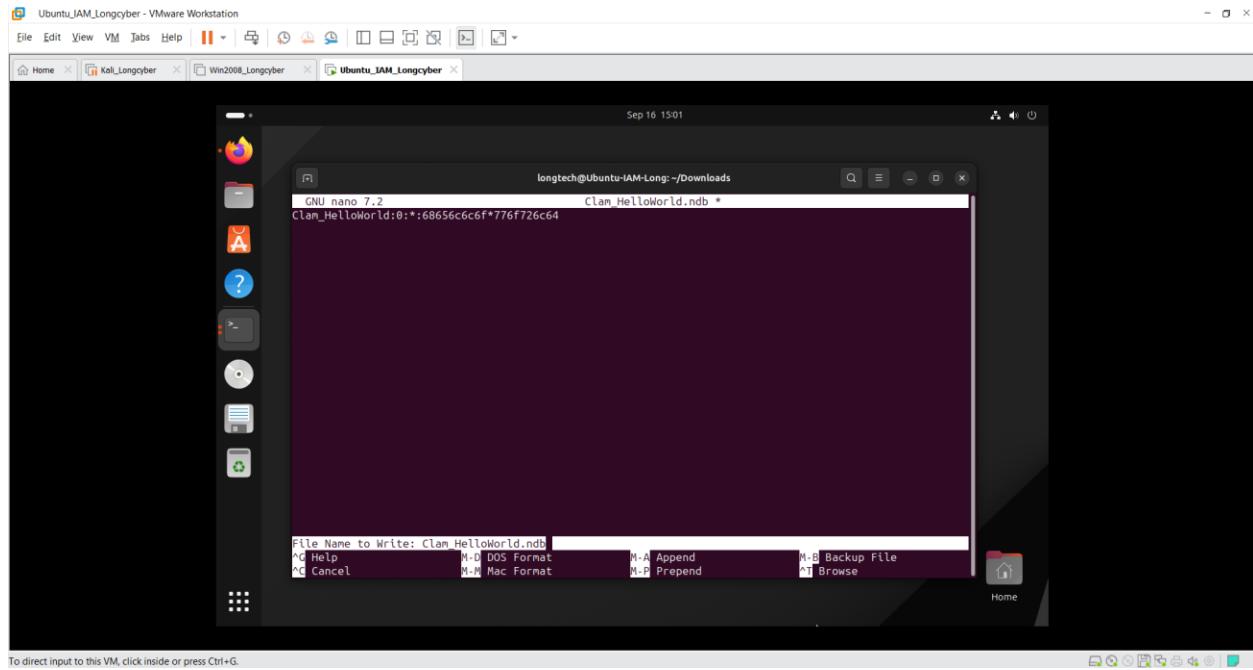
```
longtech@Ubuntu-IAM-Long:~/Downloads$ sudo clamscan --infected --remove --recursive Test/
/home/longtech/Downloads/Test/elcar.com.txt: Win.test.EICAR_H0B-1 FOUND
/home/longtech/Downloads/Test/elcar.com.txt: Removed.

----- SCAN SUMMARY -----
Known viruses: 8698650
Engine version: 1.0.5
Scanned directories: 1
Scanned files: 1
Infected files: 1
Data scanned: 0.00 MB
Data read: 0.00 MB (ratio 0.00:1)
Time: 14.996 sec (0 m 14 s)
Start Date: 2024:09:16 14:54:51
End Date: 2024:09:16 14:55:06
longtech@Ubuntu-IAM-Long:~/Downloads$
```

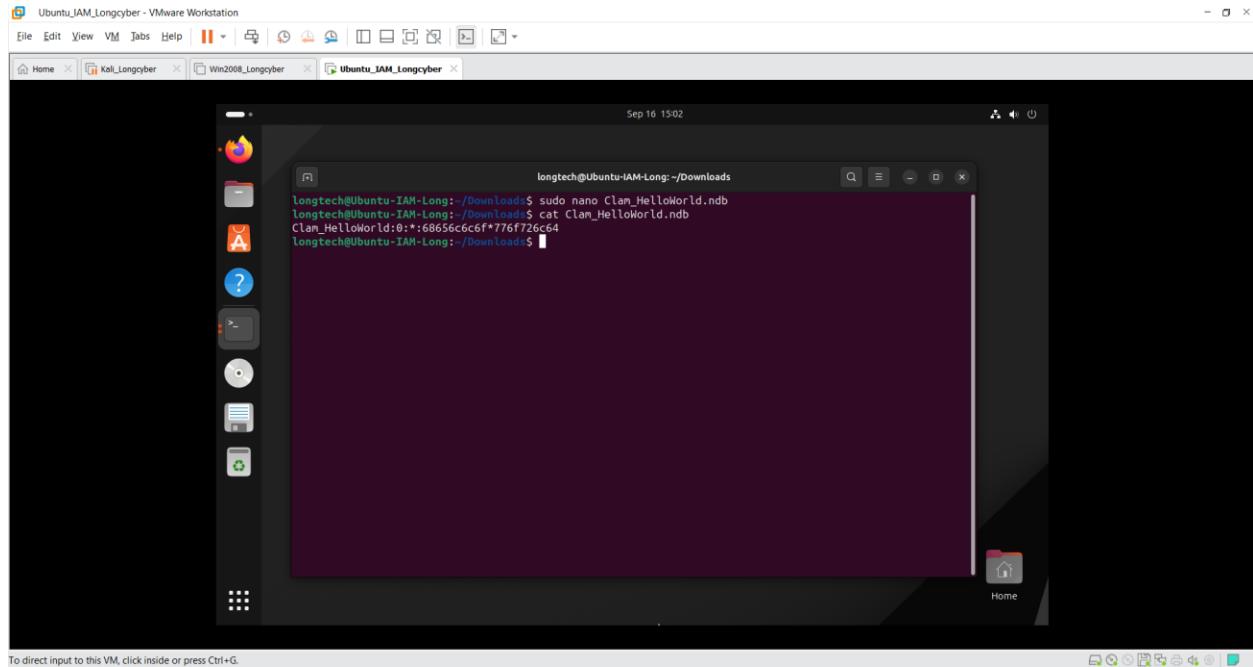
Use this command to create a signature for clamav:

```
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).

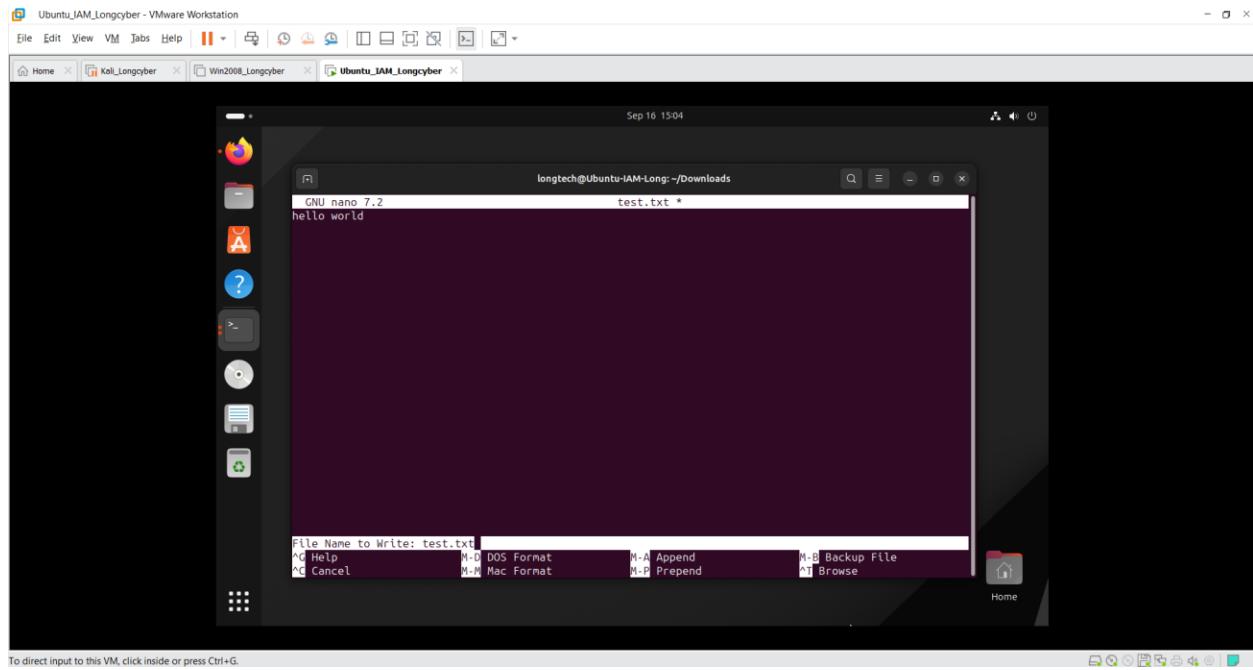


To direct input to this VM, click inside or press Ctrl+G.

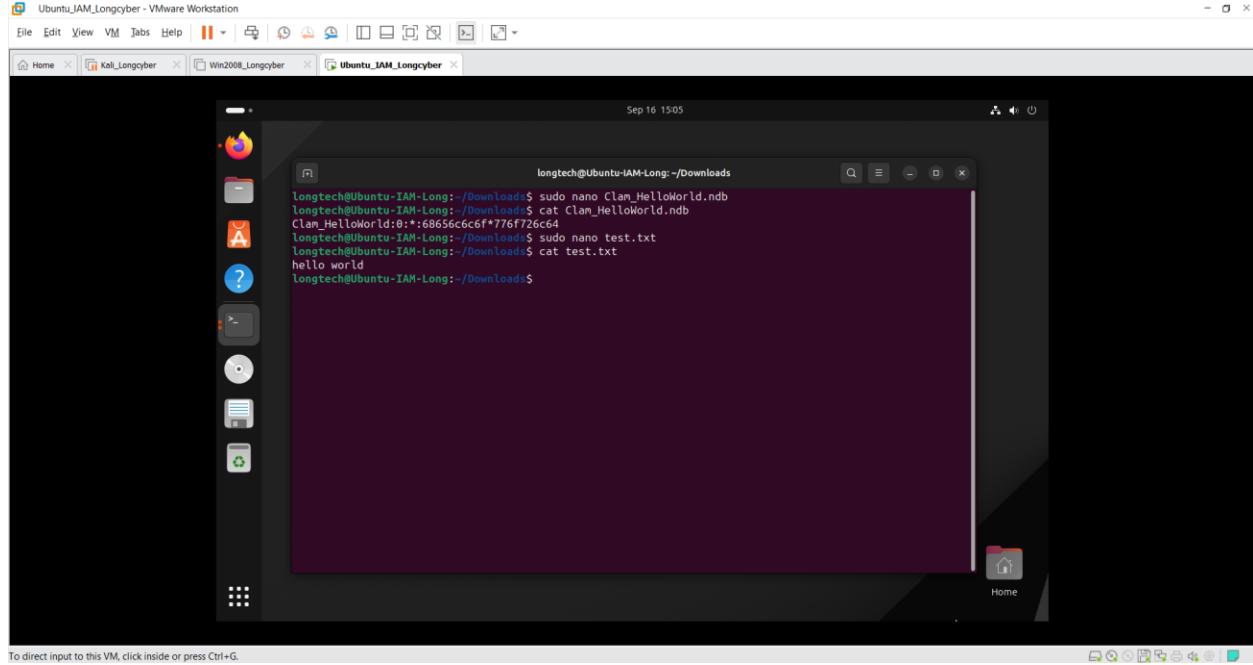


To direct input to this VM, click inside or press Ctrl+G.

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



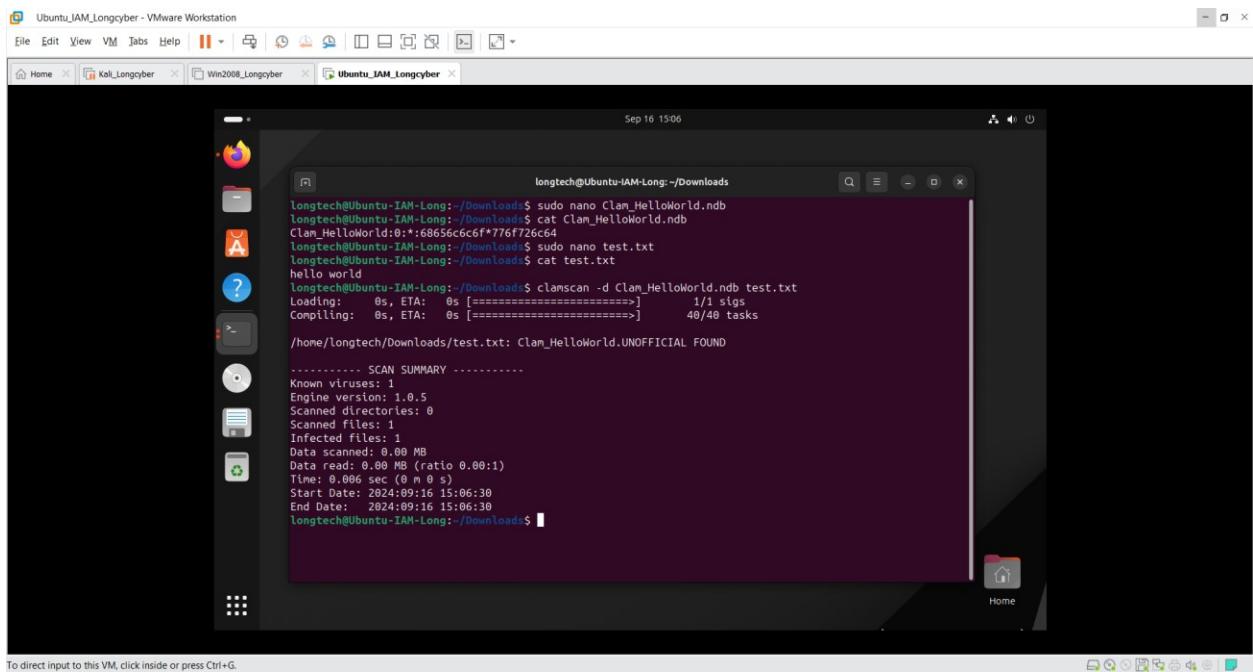
To direct input to this VM, click inside or press Ctrl+G.



To direct input to this VM, click inside or press Ctrl+G.

Then we use this command to scan:

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



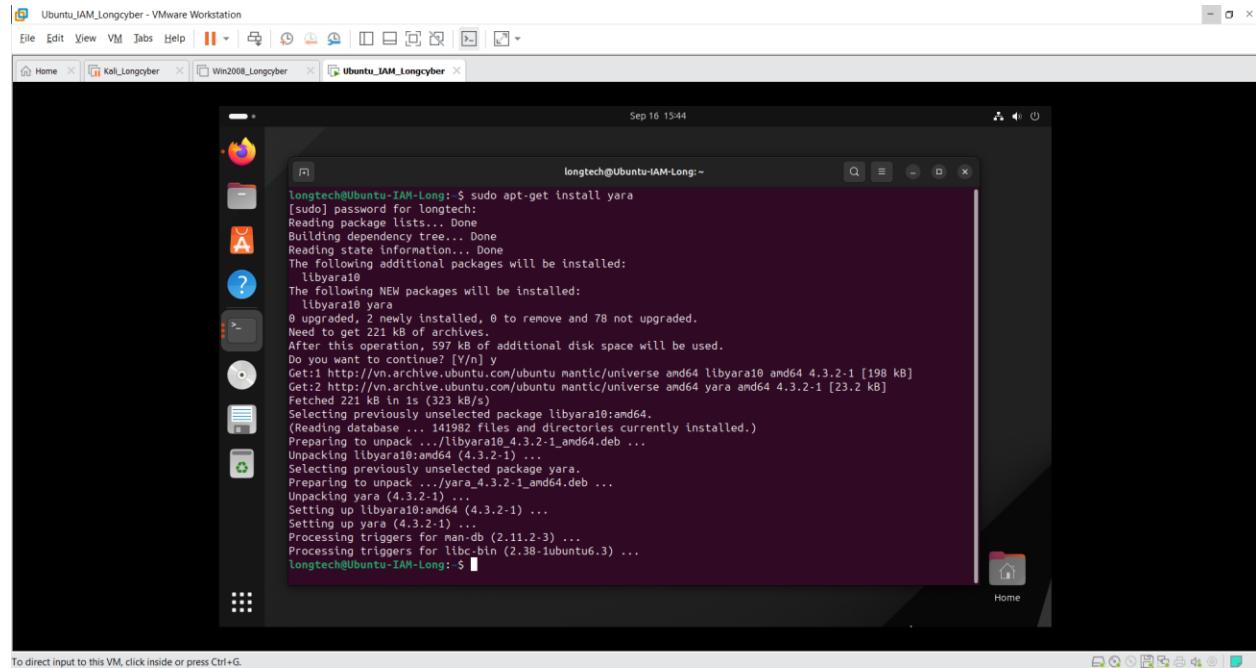
LAB 4

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.

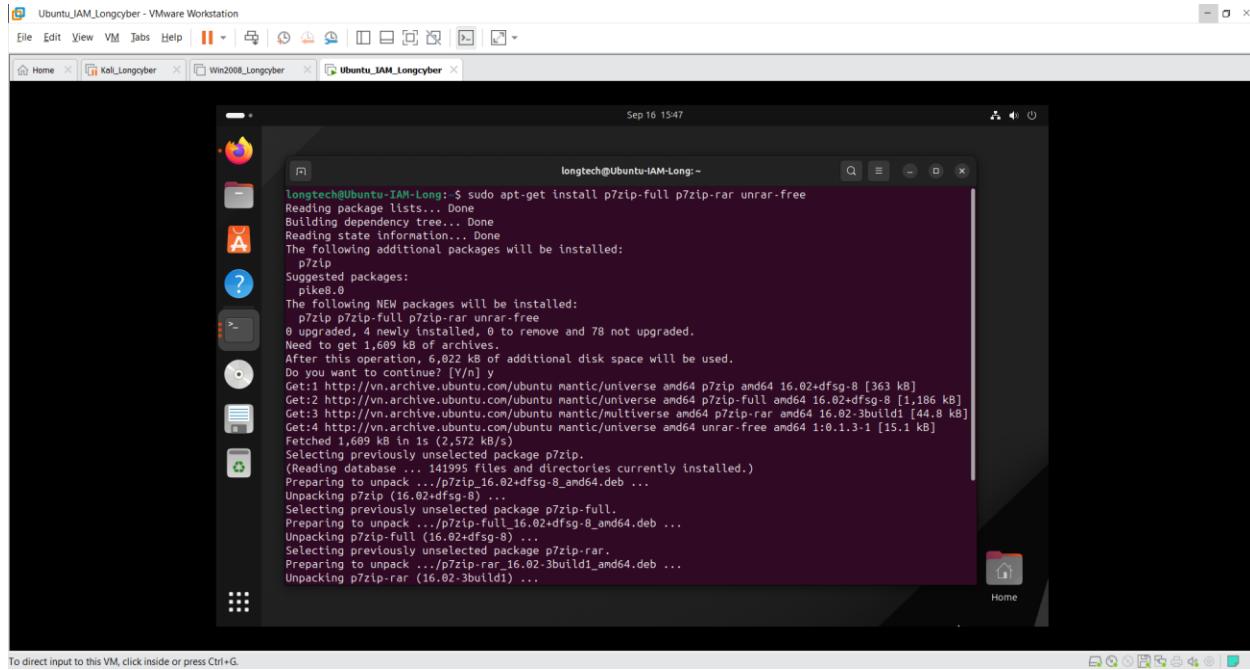
Install YARA

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



Install p7zip-full:

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

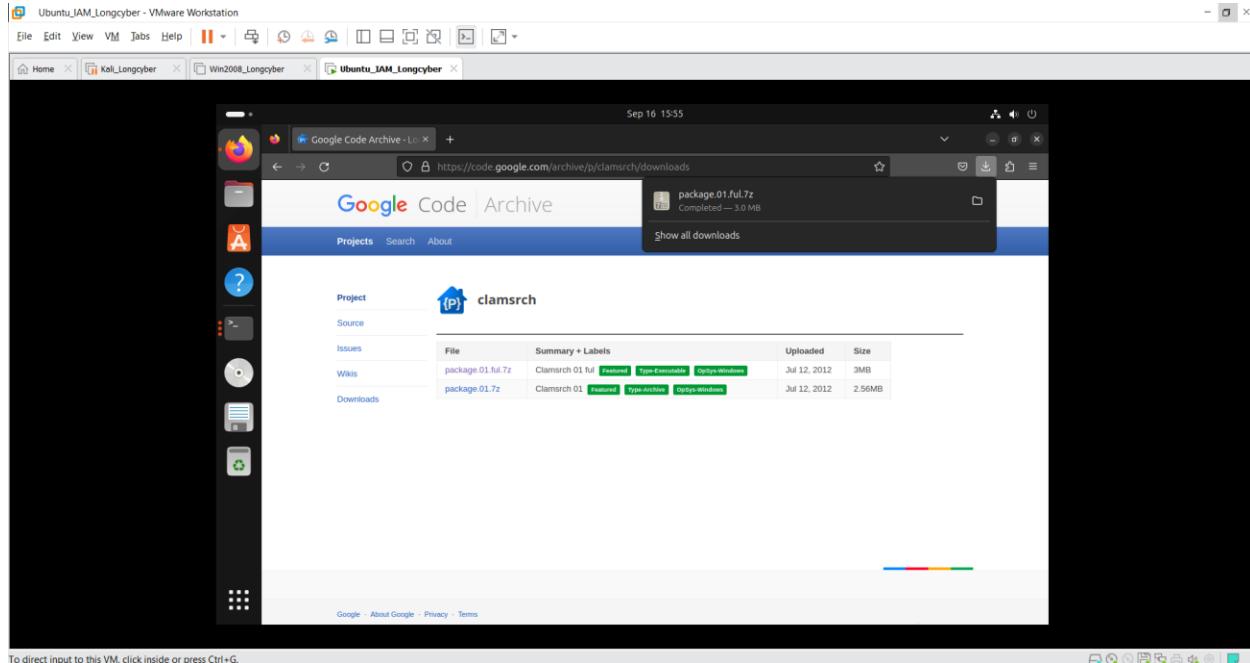


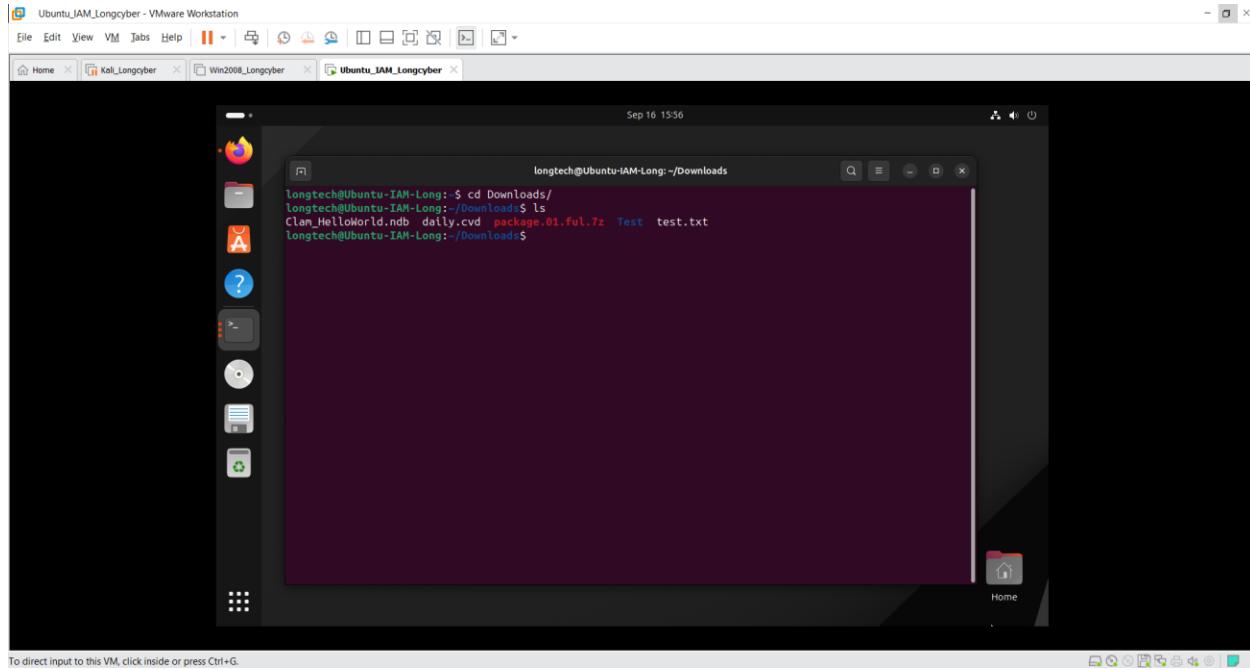
```
longtech@Ubuntu-TAM-Long: ~$ sudo apt-get install p7zip-full p7zip-rar unrar-free
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  p7zip
Suggested packages:
  p1ke8.0
The following NEW packages will be installed:
  p7zip p7zip-full p7zip-rar unrar-free
0 upgraded, 4 newly installed, 0 to remove and 78 not upgraded.
Need to get 1,699 kB of archives.
After this operation, 6,022 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://vm.archive.ubuntu.com/ubuntu natty/universe amd64 p7zip amd64 16.02+dfsg-8 [363 kB]
Get:2 http://vm.archive.ubuntu.com/ubuntu natty/universe amd64 p7zip-full amd64 16.02+dfsg-8 [1,186 kB]
Get:3 http://vm.archive.ubuntu.com/ubuntu natty/multiverse amd64 p7zip-rar amd64 16.02~3build1 [44.8 kB]
Get:4 http://vm.archive.ubuntu.com/ubuntu natty/universe amd64 unrar-free amd64 1:0.1.3-1 [15.1 kB]
Fetched 1,699 kB in 1s (2,572 kB/s)
Selecting previously unselected package p7zip.
(Reading database ... 141995 files and directories currently installed.)
Preparing to unpack .../p7zip_16.02+dfsg-8_amd64.deb ...
Unpacking p7zip (16.02+dfsg-8) ...
Selecting previously unselected package p7zip-full.
Preparing to unpack .../p7zip-full_16.02+dfsg-8_amd64.deb ...
Unpacking p7zip-full (16.02+dfsg-8) ...
Selecting previously unselected package p7zip-rar.
Preparing to unpack .../p7zip-rar_16.02~3build1_amd64.deb ...
Unpacking p7zip-rar (16.02~3build1) ...
```

Download package.01.ful.7z:

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

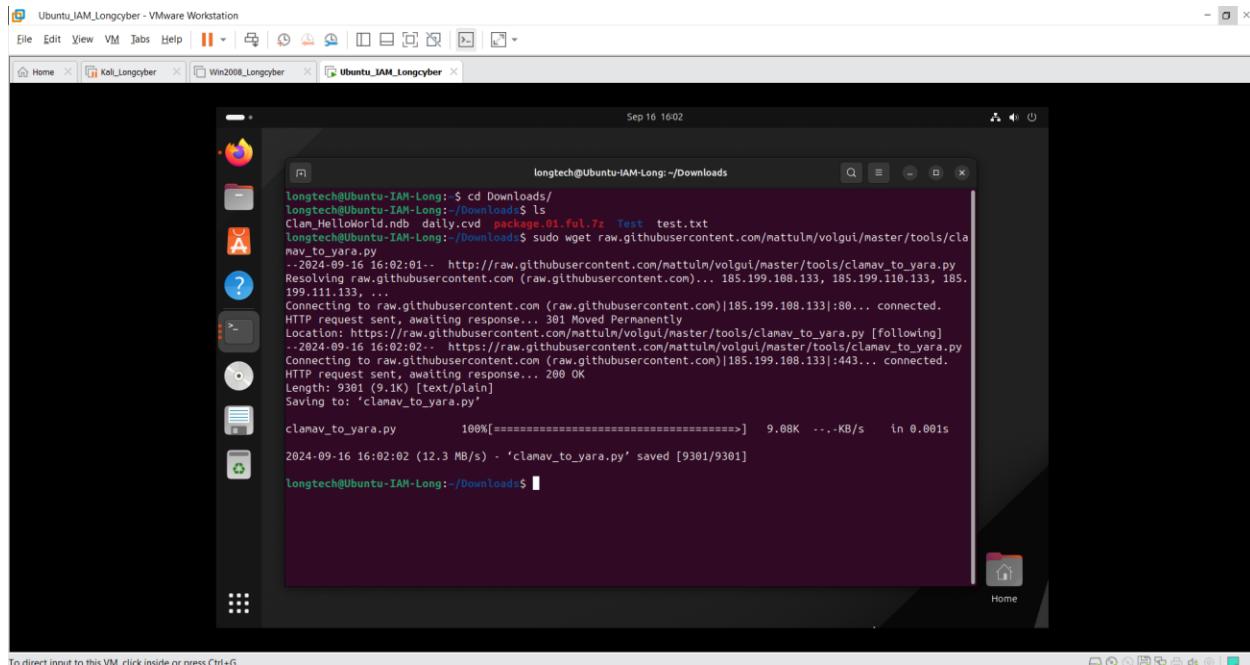
(<https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/clamsrch/package.01.ful.7z>)





Download file clam_to_yara.py:

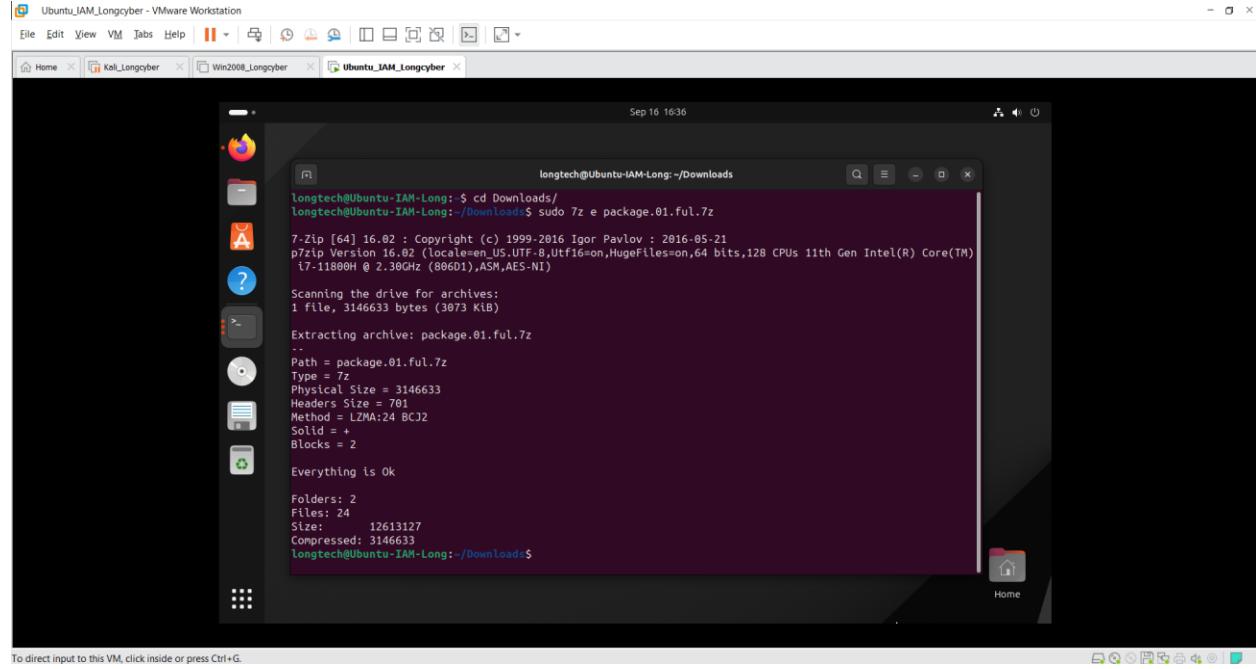
```
sudo wget  
raw.githubusercontent.com/mattulm/volgui/master/tools/clamav\_to\_yara.py
```



Convert file clamav to yara:

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

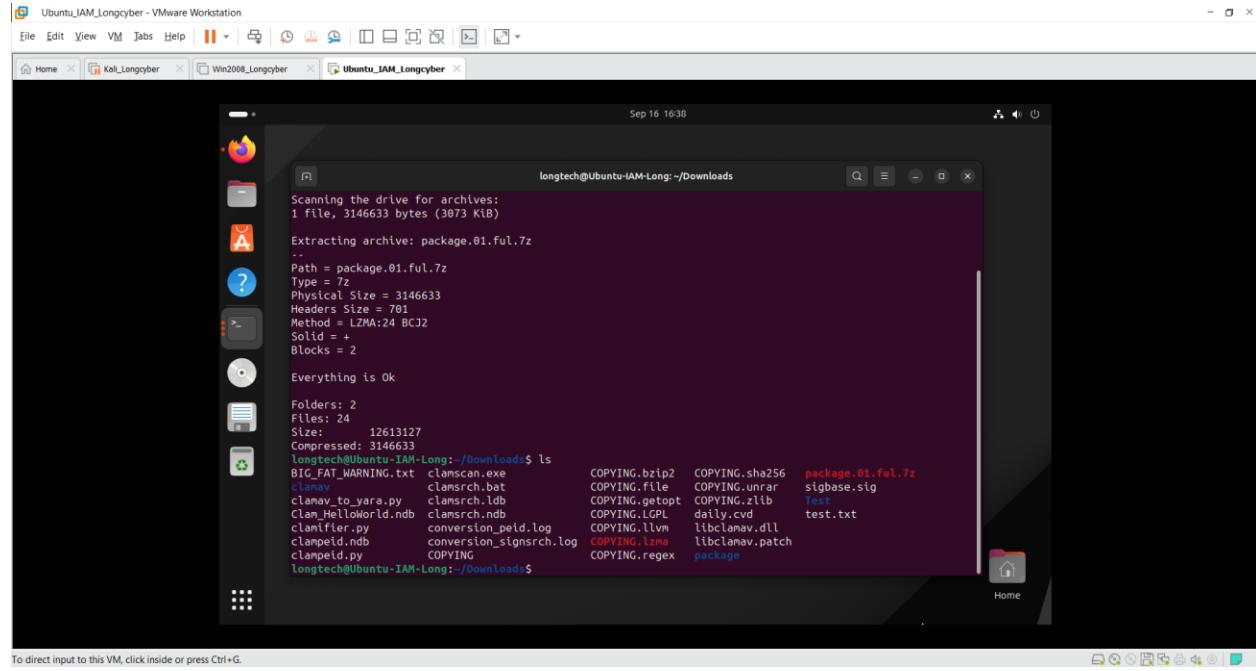
```
sudo 7z e package.01.ful.7z
```



The screenshot shows a terminal window titled "longtech@Ubuntu-IAM-Long: ~/Downloads". The command "sudo 7z e package.01.ful.7z" is run, and the output shows the extraction process. The terminal window has a dark background with white text. The status bar at the bottom indicates "Sep 16 16:36".

```
longtech@Ubuntu-IAM-Long: ~$ cd Downloads/
longtech@Ubuntu-IAM-Long: ~/Downloads$ sudo 7z e package.01.ful.7z
7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,128 CPUs 11th Gen Intel(R) Core(TM)
i7-11800H @ 2.30GHz (806D1),AES-NI)
Scanning the drive for archives:
1 file, 3146633 bytes (3073 KB)
Extracting archive: package.01.ful.7z
-
Path = package.01.ful.7z
Type = 7z
Physical Size = 3146633
Headers Size = 701
Method = LZMA:24 BCJ2
Solid = +
Blocks = 2
Everything is Ok
Folders: 2
Files: 24
Size: 12613127
Compressed: 3146633
longtech@Ubuntu-IAM-Long: ~/Downloads$
```

To direct input to this VM, click inside or press Ctrl+G.



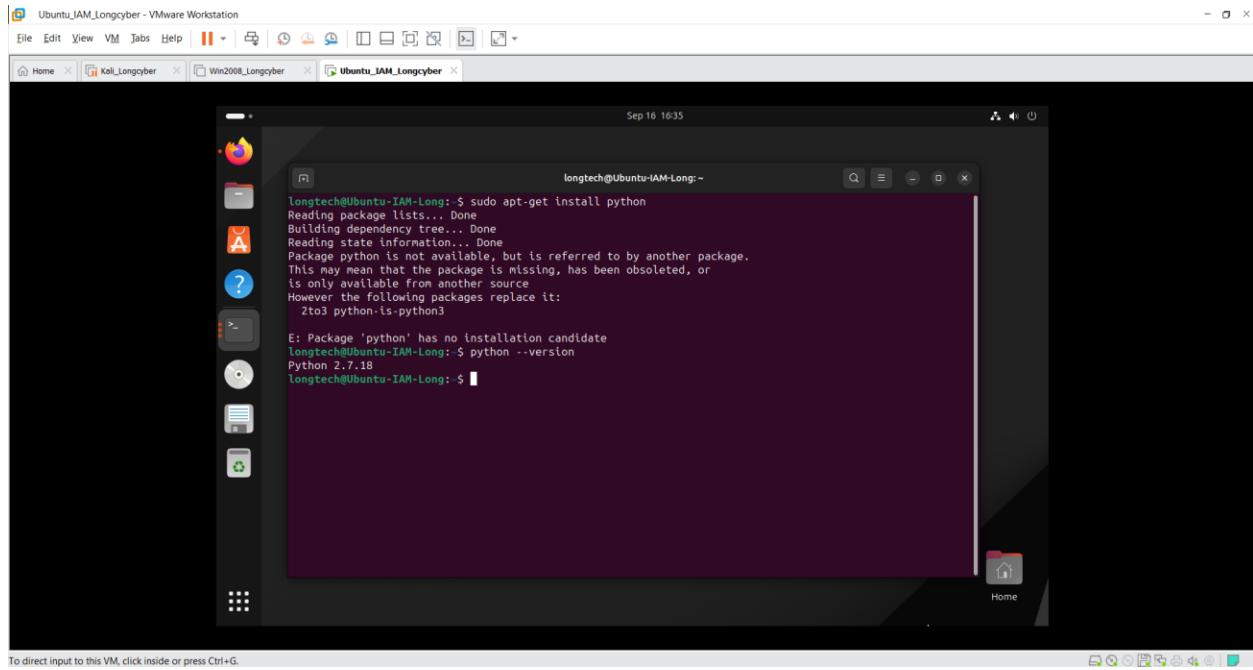
The screenshot shows a terminal window titled "longtech@Ubuntu-IAM-Long: ~/Downloads". After extracting the archive, the user runs "ls" to list the contents of the directory. The terminal window has a dark background with white text. The status bar at the bottom indicates "Sep 16 16:38".

```
longtech@Ubuntu-IAM-Long: ~$ cd Downloads/
longtech@Ubuntu-IAM-Long: ~/Downloads$ 7z e package.01.ful.7z
7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,128 CPUs 11th Gen Intel(R) Core(TM)
i7-11800H @ 2.30GHz (806D1),AES-NI)
Scanning the drive for archives:
1 file, 3146633 bytes (3073 KB)
Extracting archive: package.01.ful.7z
-
Path = package.01.ful.7z
Type = 7z
Physical Size = 3146633
Headers Size = 701
Method = LZMA:24 BCJ2
Solid = +
Blocks = 2
Everything is Ok
Folders: 2
Files: 24
Size: 12613127
Compressed: 3146633
longtech@Ubuntu-IAM-Long: ~/Downloads$ ls
BIG_FAT_WARNING.txt clamscan.exe COPYING.bzip2 COPYING.sha256 package.01.ful.7z
clamav clamsrch.bat COPYING_file COPYING.unrar sigbase.sig
clamav_to_yara.py clamsrch.lbd COPYING_getopt COPYING_zlib Test
Clam_HelloWorld.ndb clamsrch.nbd COPYING_LGPL daily.cvd test.txt
clamfier.py conversion_peild.log COPYING_llvm libclamav.dll
clampeid.nbd conversion_signsrch.log COPYING_lzma libclamav.patch
clampeid.py COPYING_REGEX package
longtech@Ubuntu-IAM-Long: ~/Downloads$
```

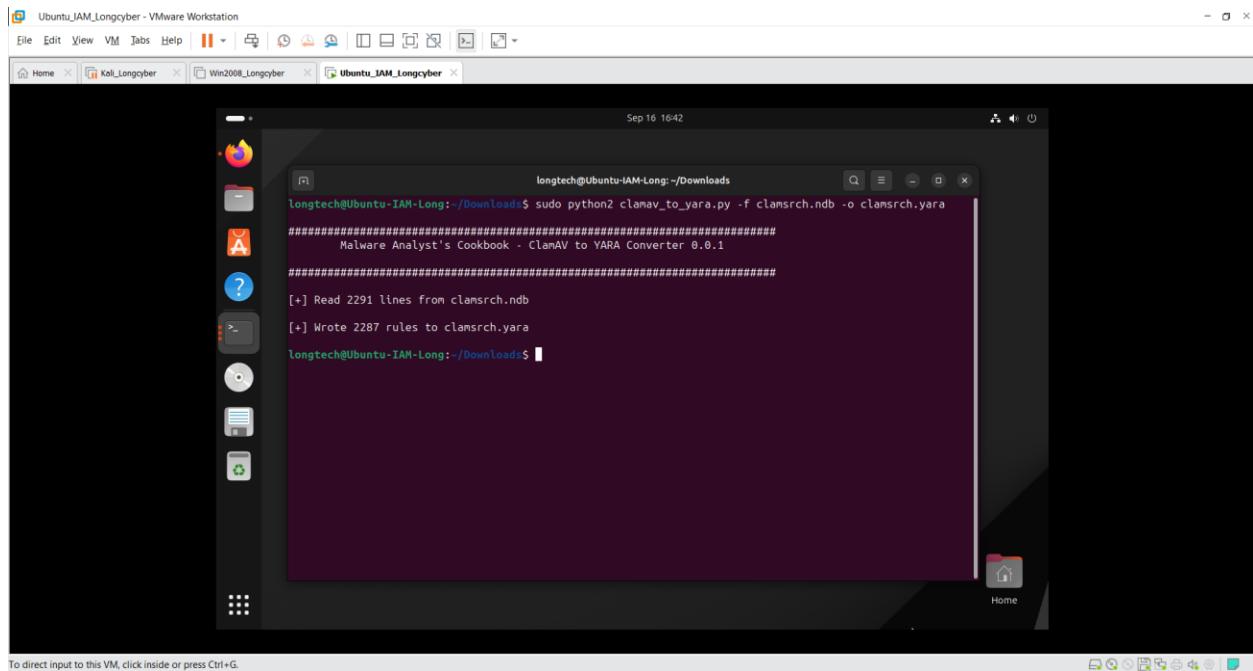
To direct input to this VM, click inside or press Ctrl+G.

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
```

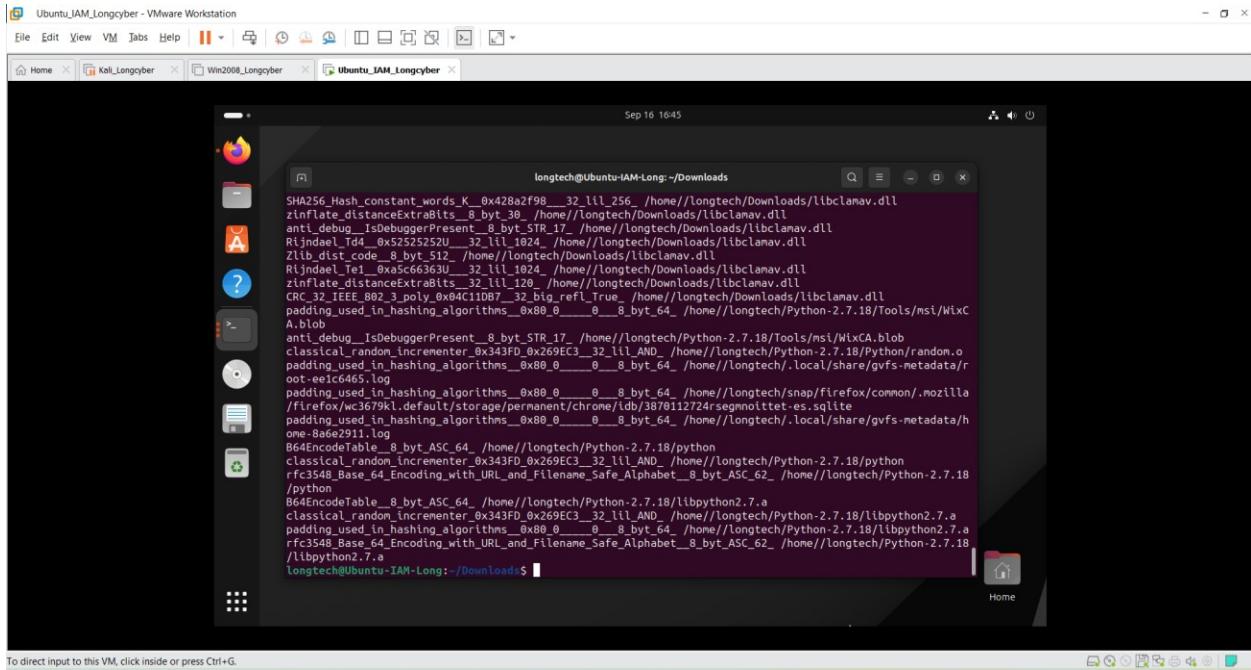


```
longtech@Ubuntu-IAM-Long:~/Downloads$ ls
BIG_FAT_WARNING.txt clamsrch.bt COPYING.bzip2 COPYING.sha256 package
clanav clamsrch.lbd COPYING_file COPYING.unrar package.01.ful.7z
clanav_to_yara.py clamsrch.ndb COPYING_getopt COPYING_zlib sigbase.sig
clamflier.py clamsrch.yara COPYING_LGPL custome.yara Test
clampeid.ndb conversion_peid.log COPYING_llvm daily.cvd
clampeid.py conversion_signsrch.log COPYING_lzma libclamav.dll
clamscan.exe COPYING COPYING_regex libclamav.patch
```

Start scanning with the yara

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

```
longtech@Ubuntu-IAM-Long:~/Downloads$ yara -r clamsrch.yara /home/
rfc3548_Base_32_Encoding_8_byt_ASC_32_ /home/longtech/snap/firefox/common/.cache.mozilla/firefox/wc3679
kl.default/startupCache/startupCache.8.little
padding_used_in_hashing_algorithms_0x88_0_0_8_byt_64_ /home/longtech/snap/firefox/common/.mozilla
/firefox/wc3679kl.default/permissions.sqlite
padding_used_in_hashing_algorithms_0x88_0_0_8_byt_64_ /home/longtech/snap/firefox/common/.mozilla
/firefox/wc3679kl.default/content-prefs.sqlite
padding_used_in_hashing_algorithms_0x88_0_0_8_byt_64_ /home/longtech/snap/firefox/common/.mozilla
/firefox/wc3679kl.default/storage-sync-v2.sqlite-wal
padding_used_in_hashing_algorithms_0x88_0_0_8_byt_64_ /home/longtech/snap/firefox/common/.mozilla
/firefox/wc3679kl.default/protections.sqlite
padding_used_in_hashing_algorithms_0x88_0_0_8_byt_64_ /home/longtech/snap/firefox/common/.mozilla
/firefox/wc3679kl.default/storage/ls-archive.sqlite
padding_used_in_hashing_algorithms_0x88_0_0_8_byt_64_ /home/longtech/snap/firefox/common/.mozilla
/firefox/wc3679kl.default/webappstore.sqlite
padding_used_in_hashing_algorithms_0x88_0_0_8_byt_64_ /home/longtech/snap/firefox/common/.mozilla
/firefox/wc3679kl.default/default/directory.sqlite
padding_used_in_hashing_algorithms_0x88_0_0_8_byt_64_ /home/longtech/snap/firefox/common/.mozilla
/firefox/wc3679kl.default/key4.db
padding_used_in_hashing_algorithms_0x88_0_0_8_byt_64_ /home/longtech/snap/firefox/common/.mozilla
/firefox/wc3679kl.default/storage-sync-v2.sqlite
padding_used_in_hashing_algorithms_0x88_0_0_8_byt_64_ /home/longtech/snap/firefox/common/.mozilla
/firefox/wc3679kl.default/places.sqlite
Binary_arithmetic_decoder_LPSTable_T264_cabac_range_lps_8_byt_256_ /home/longtech/snap/firefox/commo
n/.mozilla/firefox/wc3679kl.default/gmp-gmpopenh264/2.3.2/libgmpopenh264.so
libavcodec_ff_zigzag_direct_8_byt_64_ /home/longtech/snap/firefox/common/.mozilla/firefox/wc3679kl.defa
ult/gmp-gmpopenh264/2.3.2/libgmpopenh264.so
Bit_count_256__popcount_tab_population_count__8_byt_256_ /home/longtech/snap/firefox/common/.mozilla
```



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 {
```

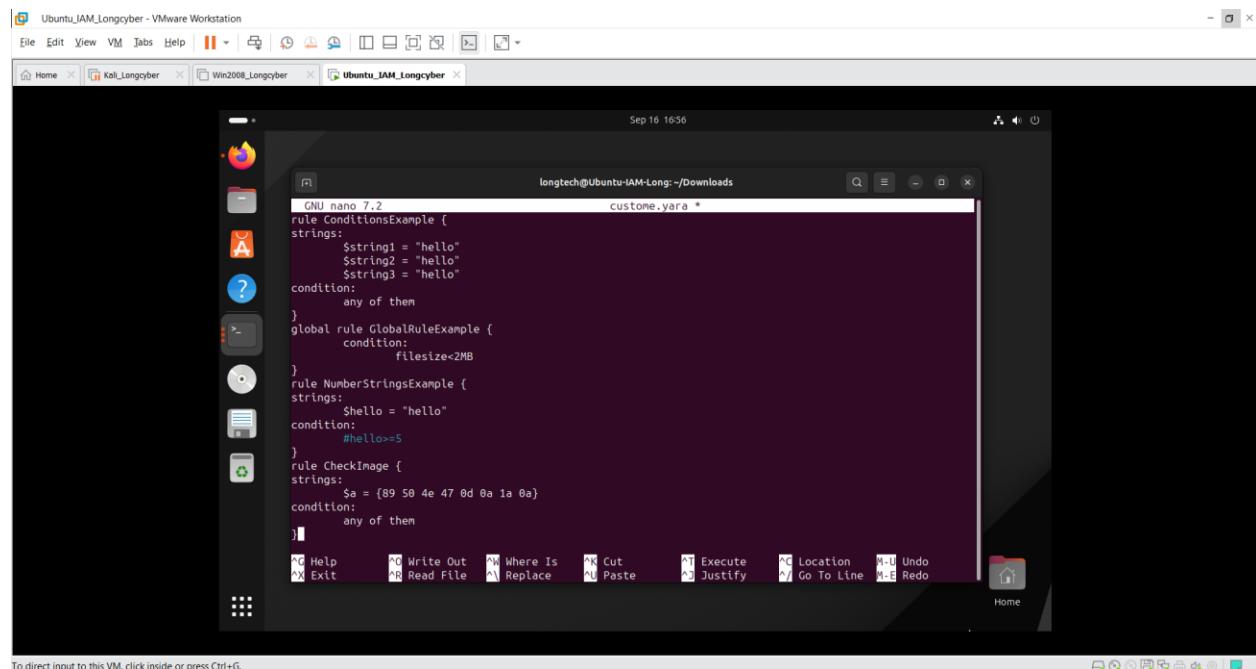
strings:

```
$a = {89 50 4e 47 0d 0a 1a 0a}
```

condition:

any of them

```
}
```



```
GNU nano 7.2          longtech@Ubuntu-IAM-Long: ~/Downloads      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
}
```

```
Ubuntu_IAM_Longcyber - VMware Workstation
File Edit View VM Tabs Help | + | X | 
Home Kali_Longcyber Win2008_Longcyber Ubuntu_IAM_Longcyber | 
Sep 16 16:57
longtech@Ubuntu-IAM-Long:~/Downloads$ sudo nano custome.yara
longtech@Ubuntu-IAM-Long:~/Downloads$ cat custome.yara
rule ConditionsExample {
strings:
    $string1 = "hello"
    $string2 = "hello"
    $string3 = "hello"
condition:
    any of them
}
rule GlobalRuleExample {
    condition:
        filesize<2MB
}
rule NumberStringsExample {
strings:
    $Hello = "hello"
condition:
    #Hello>=5
}
rule CheckImage {
strings:
    $a = {89 50 4e 47 0d 0a 1a 0a}
condition:
    any of them
}
longtech@Ubuntu-IAM-Long:~/Downloads$
```

Test yara rules:

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

```
Ubuntu_IAM_Longcyber - VMware Workstation
File Edit View VM Tabs Help | + | X | 
Home Kali_Longcyber Win2008_Longcyber Ubuntu_IAM_Longcyber | 
Sep 16 17:09
longtech@Ubuntu-IAM-Long:~/Downloads$ yara -r custome.yara Test/
GlobalRuleExample Test//clam_HelloWorld.ndb
ConditionsExample Test//test.txt
GlobalRuleExample Test//test.txt
GlobalRuleExample Test//test.txt
longtech@Ubuntu-IAM-Long:~/Downloads$
```

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