Course: CYB301 Security Defense and Response (Canadian Context)

Lab 11: NIDS (Snort with Three Modes)

Coordinator and Instructor:

Muhammad Siddiqui

Student: Saiprasad Raman (23074624)

Installing Ubuntu (VMWare)

Installing Ubuntu is similar to how you installed Kali Linux previously.

Step 1: Download an Ubuntu ISO and install the OS through VMware Workstation Player. a)

Go to: https://ubuntu.com/

b) Click the Download link in the menu at the top of the page.

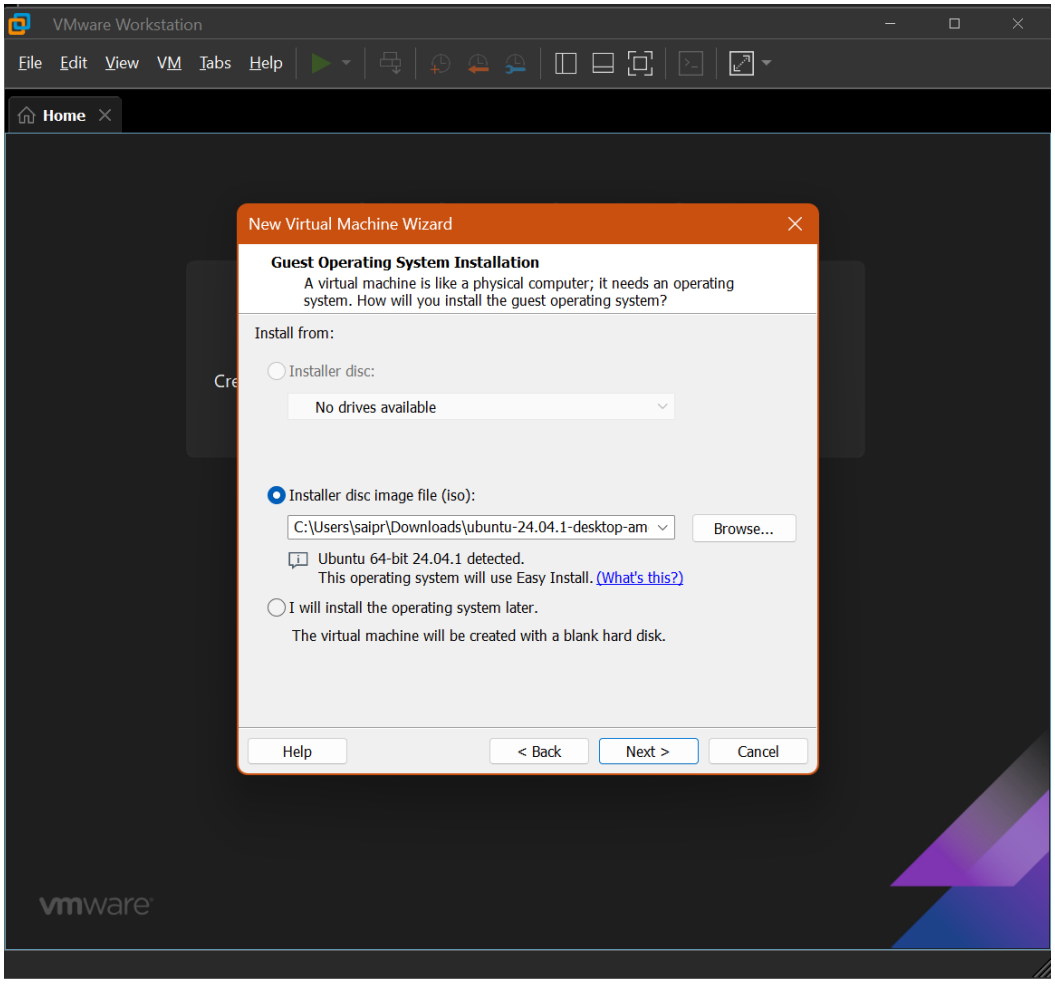
c) Under Ubuntu Desktop, click the current version. At the time of publication, it is 20.04 LTS.

d) The download should start automatically, but if not, click the **Download Now** hyperlink at the top.

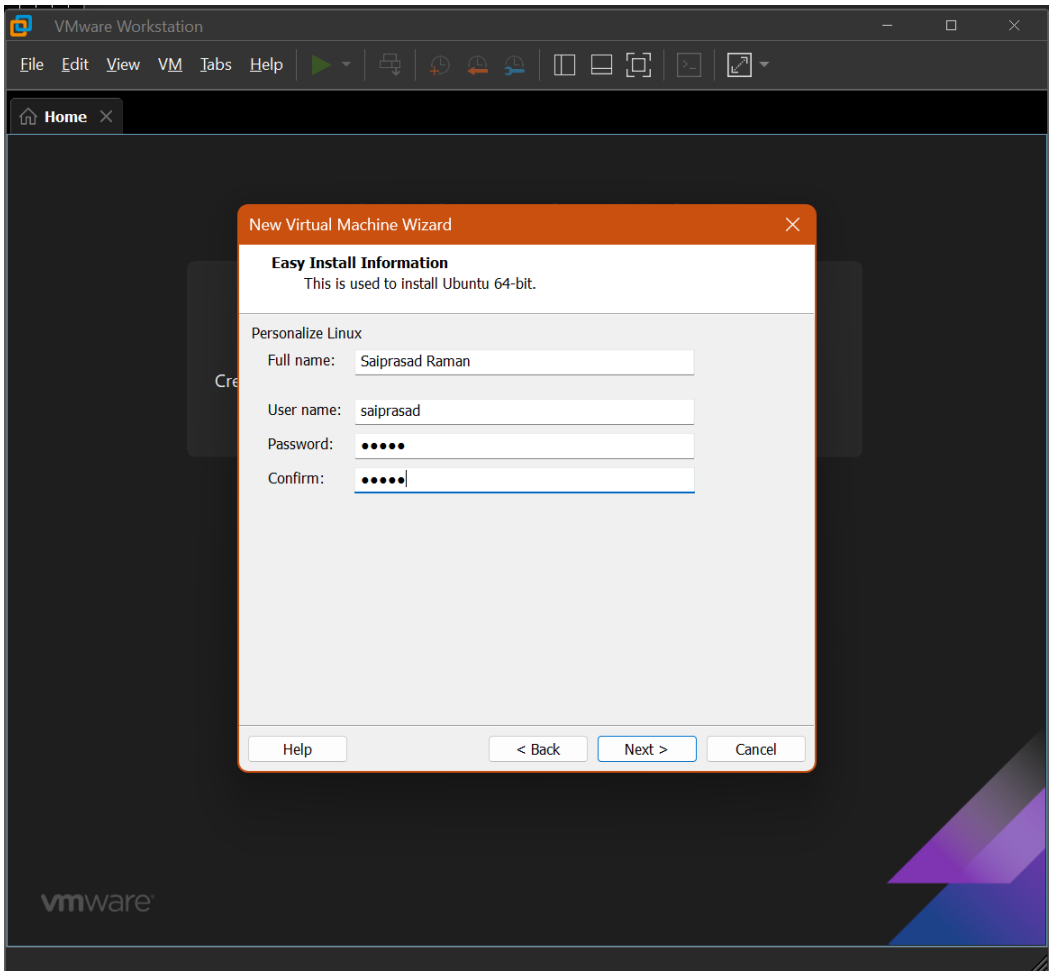
e) Run **VMware Workstation Player**.

f) Click **Create a New Virtual Machine**.

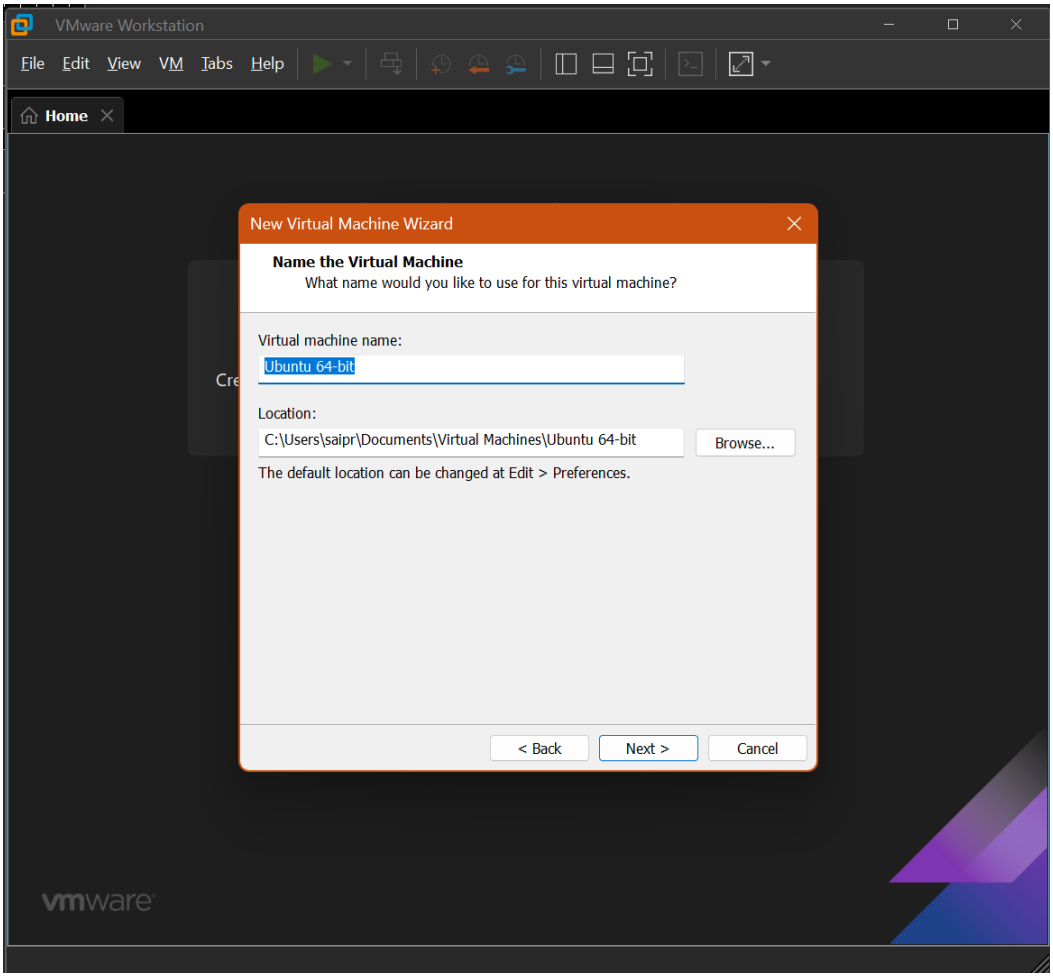
g) With the **Installer Disc Image File (ISO):** radio button selected, browse to the Ubuntu ISO, which will be in your Downloads folder. Click **Next**.



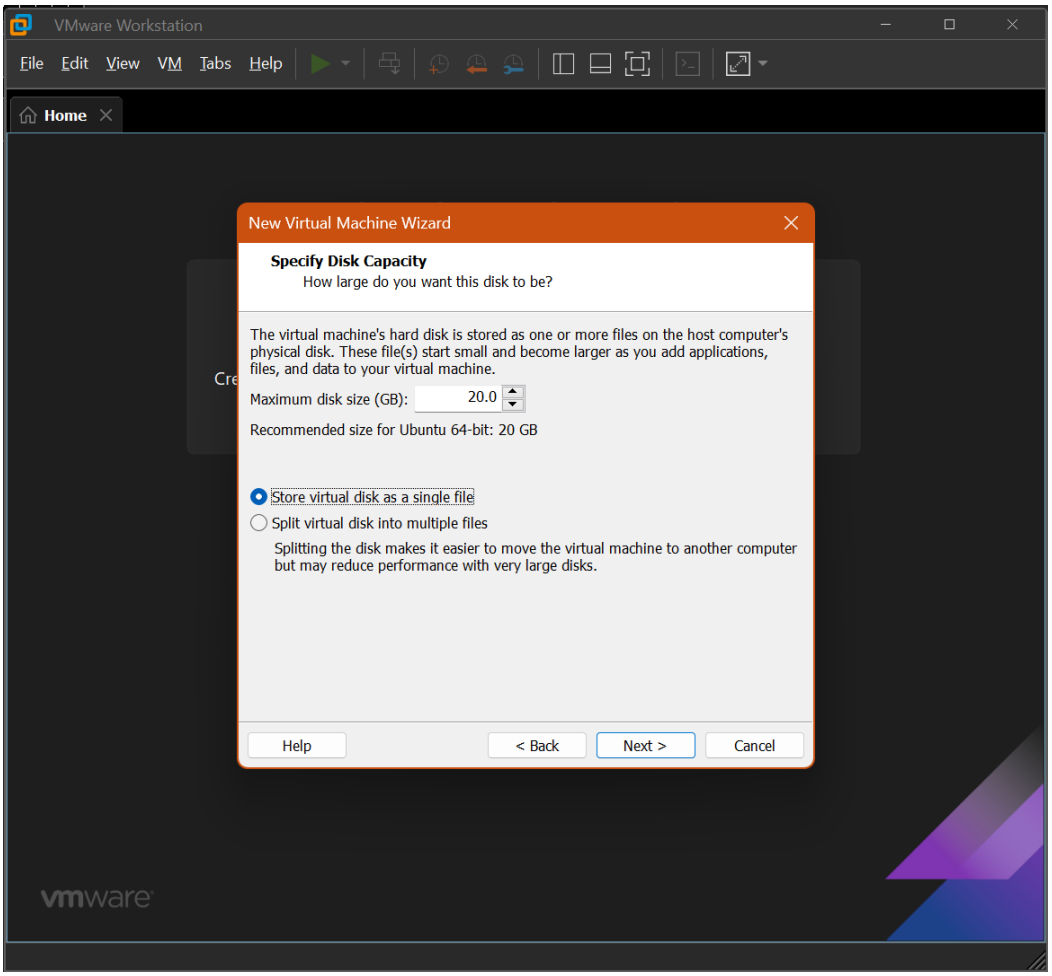
h) Provide the easy install information, including full name, username, password, and password confirmation. Click **Next**.



i) In the textbox, change or keep the name of Ubuntu 64-bit and then click **Next**.

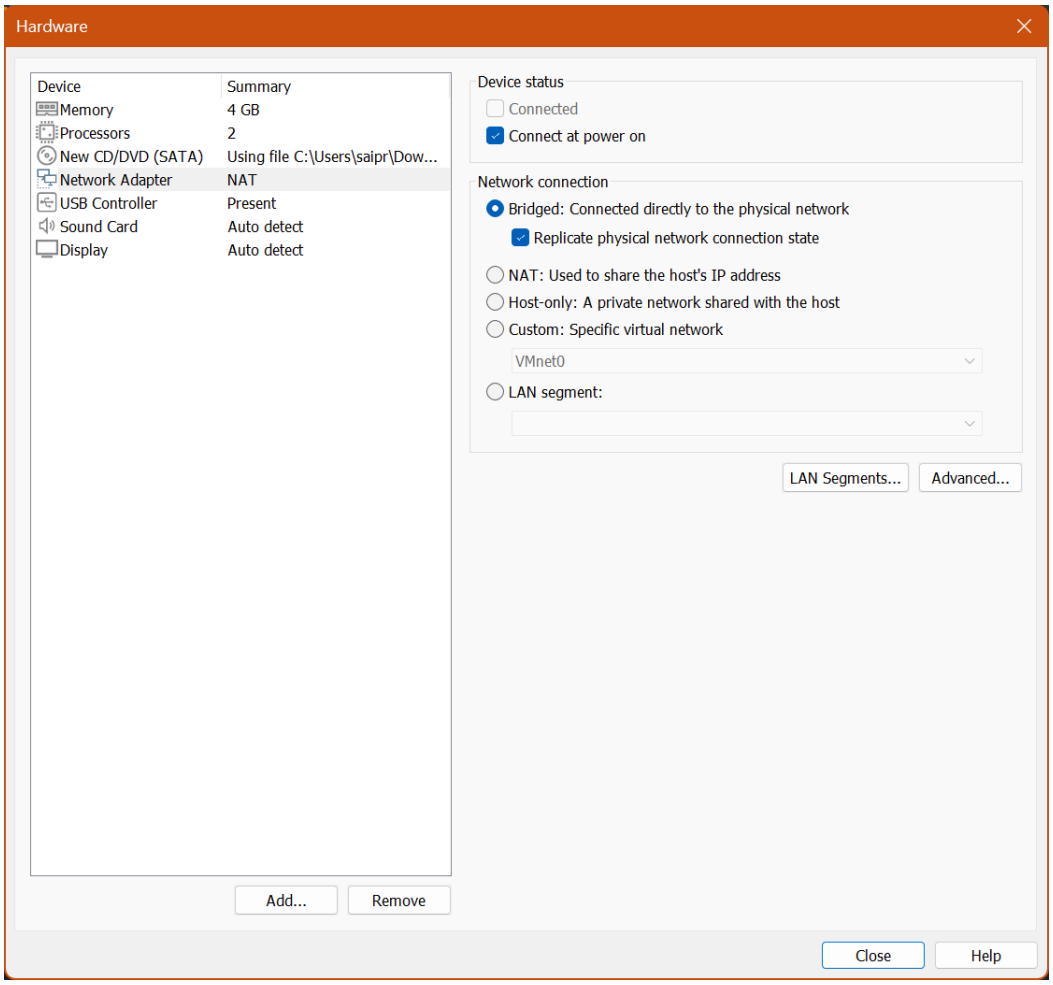


j) Keep the default maximum disk size but select the Store Virtual Disk As a Single File radio button instead of using the default selection. Click **Next**.



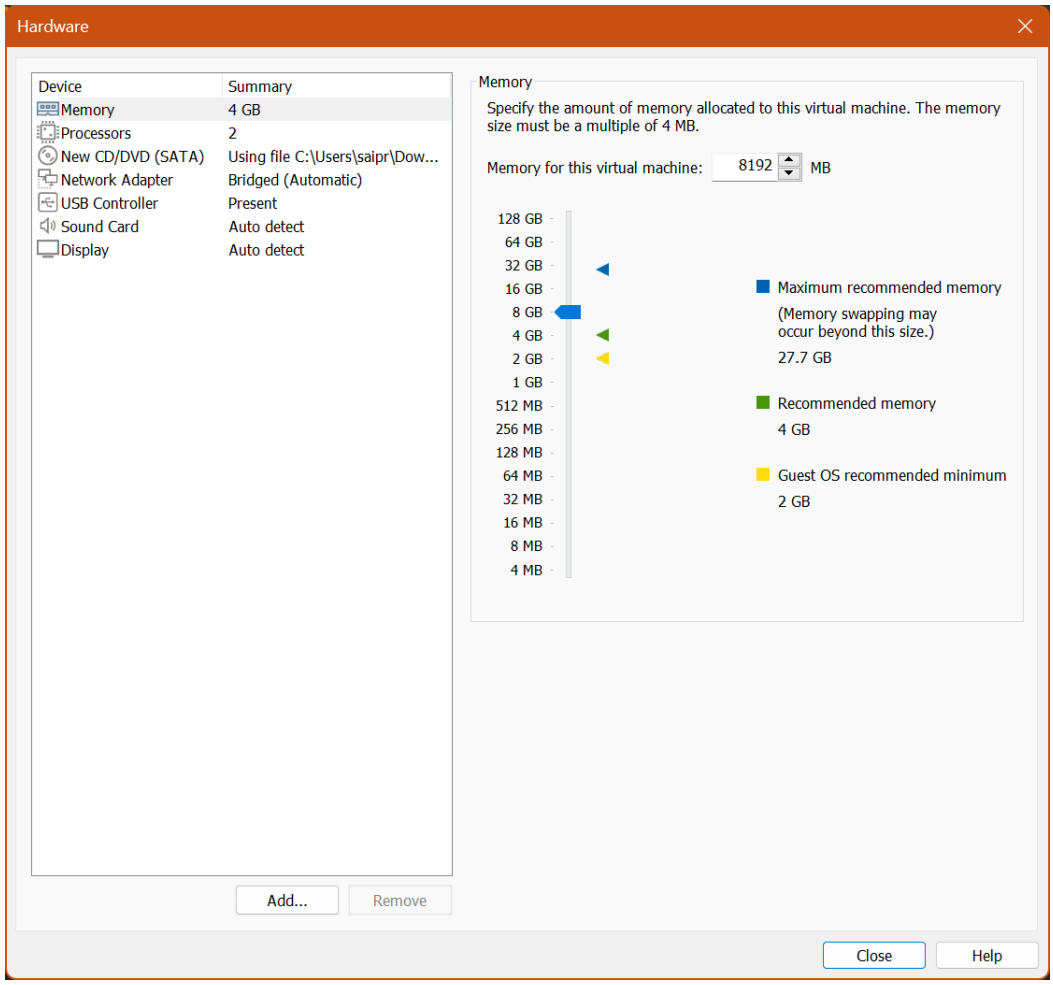
k) Click **Customize Hardware….**

l) On the left, select **Network Adapter** and select the radio button **Bridged: Connected Directly To The Physical Network** and the checkbox **Replicate Physical Network Connection State**.

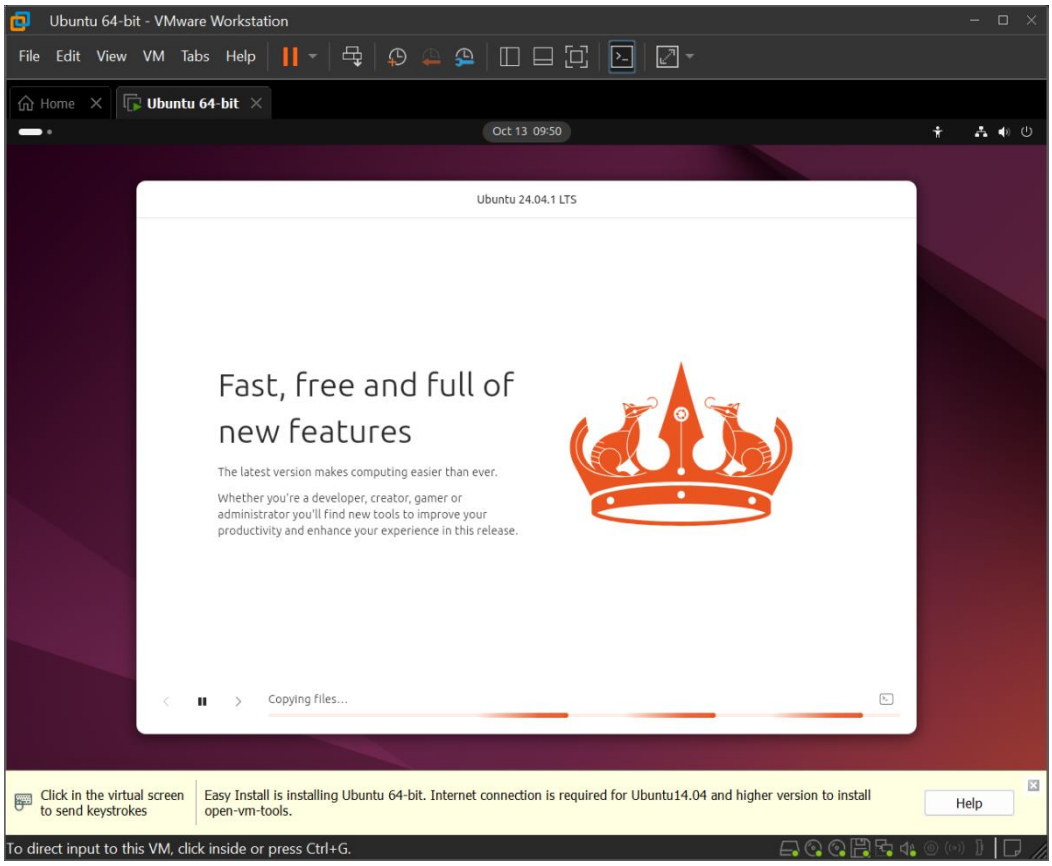


m) Feel free to increase the VM's RAM, if desired, by clicking Memory and increasing the allocated memory.

n) Click **Close** and then click **Finish**.

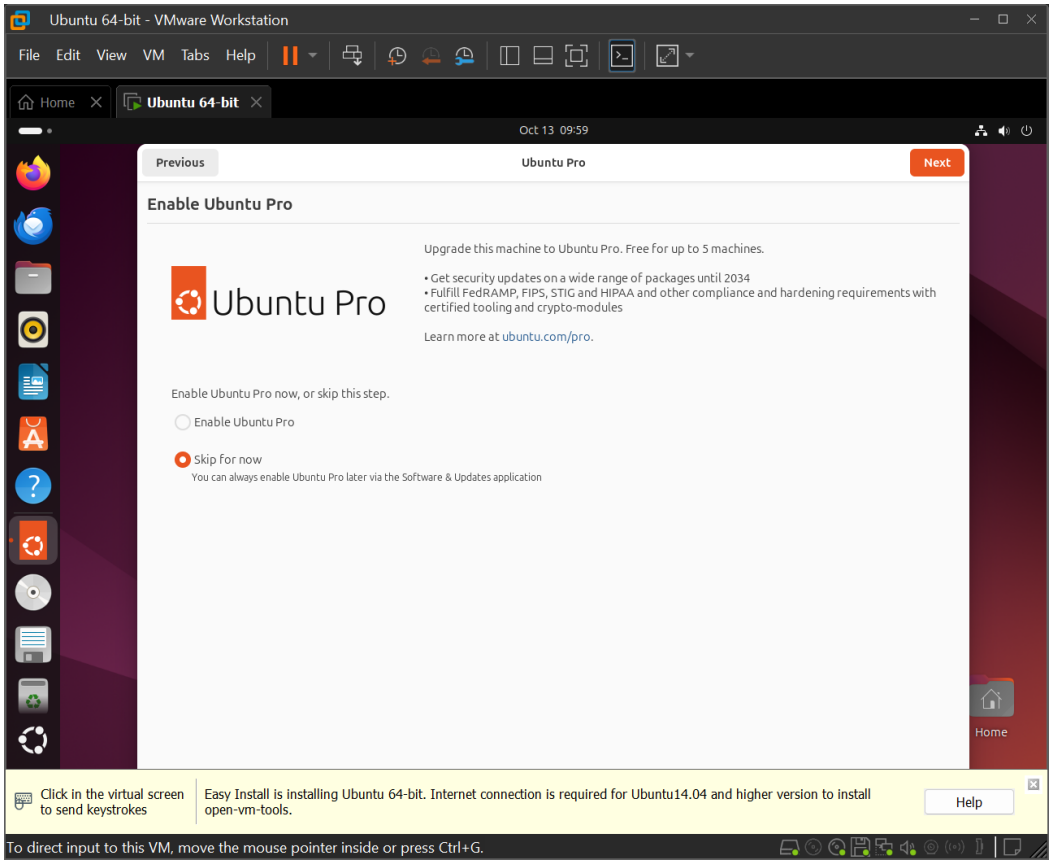


o) After some verification checks and the copying of files, the installation will start.



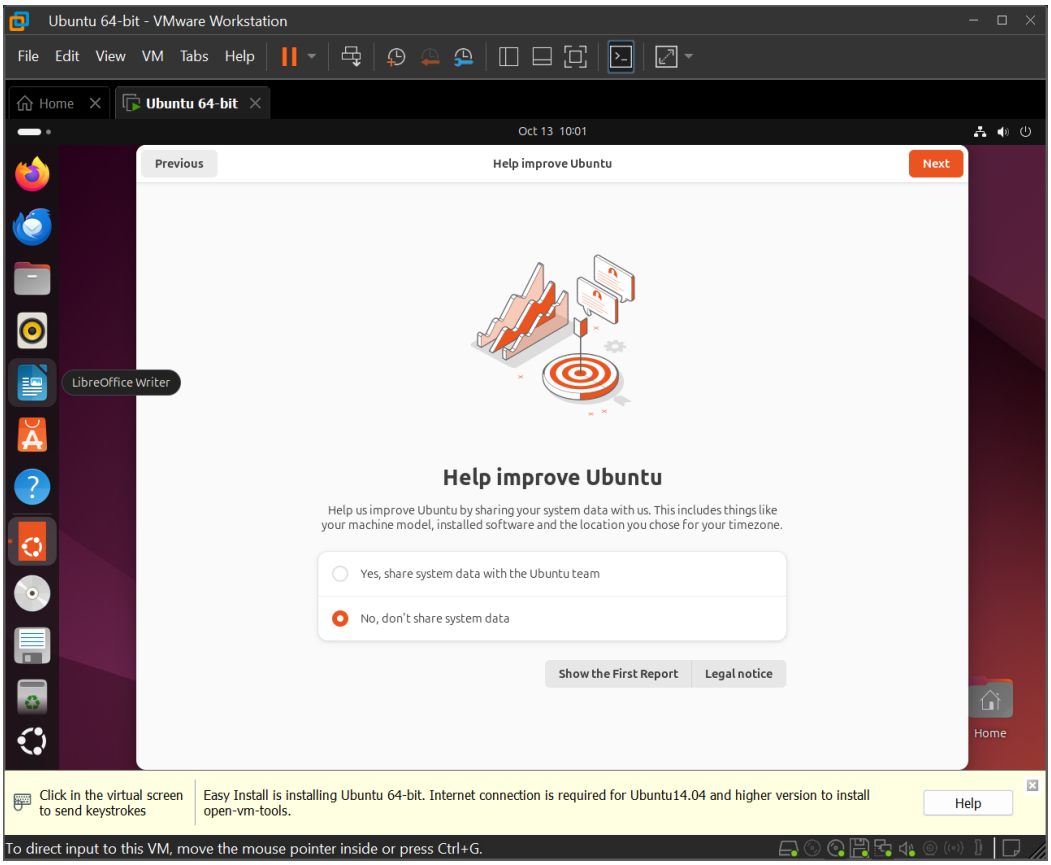
p) When the installation completes, click your username, provide your password, and press **ENTER** to log in.

q) Click **Skip** at the top right of the Online Accounts screen.



r) Click **Next** at the top right on the Livepatch screen.

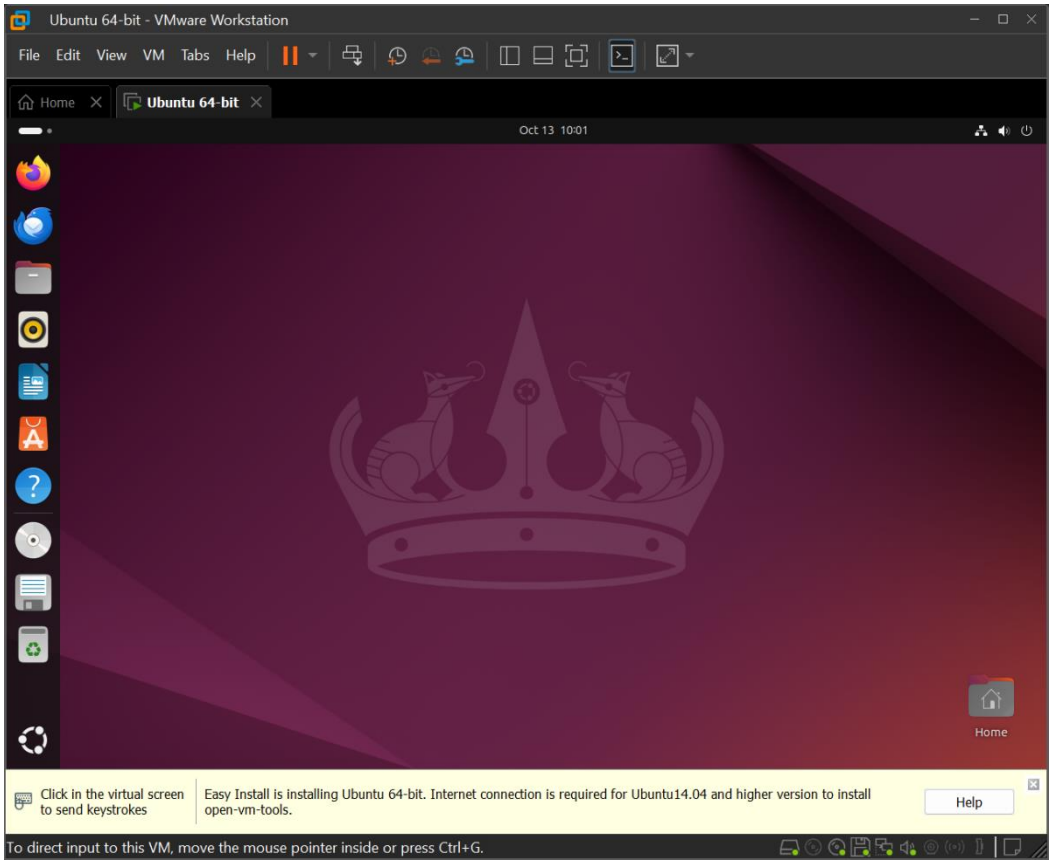
s) Click the radio button next to **No, Don't Send System Info** and then click **Next** at the top right of the screen.



t) Click **Next** at the top right of the Privacy screen.

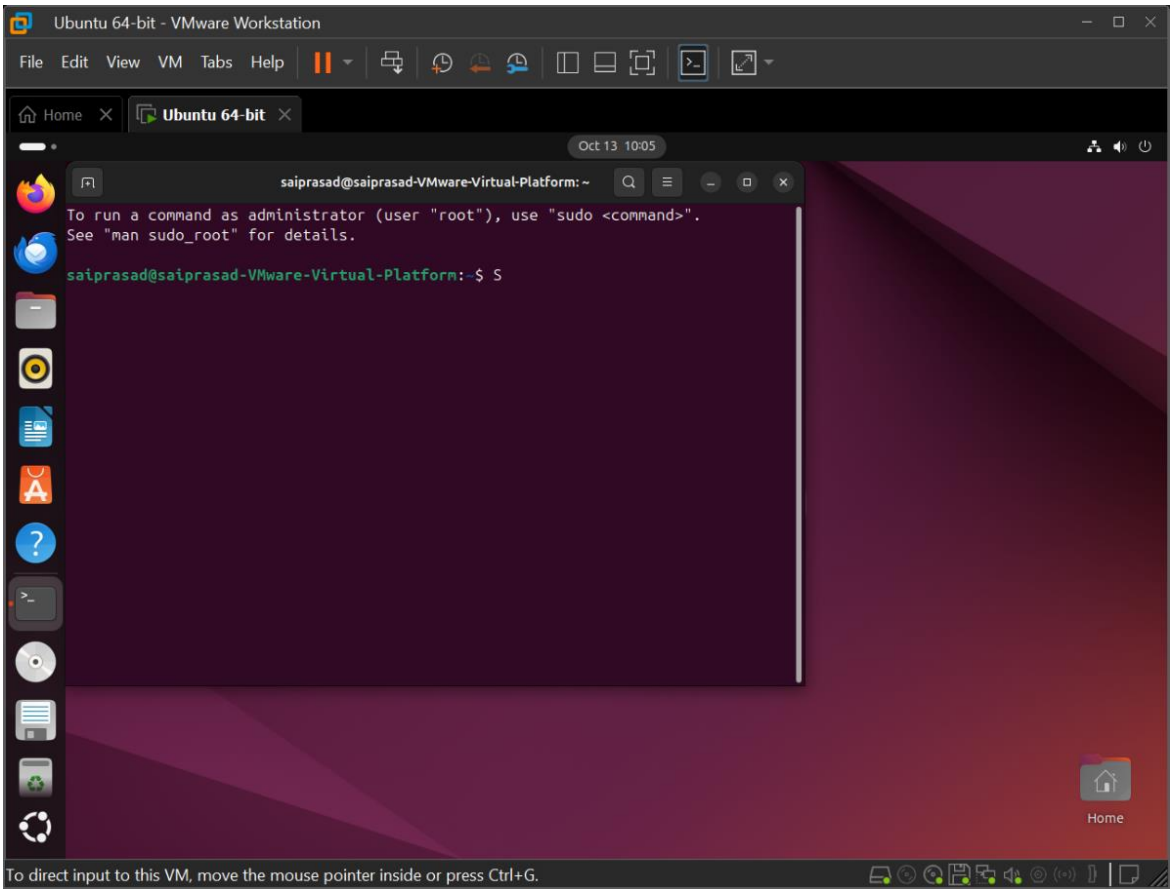
u) Click **Done** in the top right of the Ready to Go screen.

v) If the Software Updater pops up, click the Install Now button. When prompted, provide your password. Then, when you see The Computer Needs to Restart to Finish Installing Updates, click Restart Now. After the VM reboots, log in once again.



Installing Wireshark and Snort
Step 2: Download and install Wireshark and Snort. a) Press **CTRL-G** or click in the VM to work in the VM. Press **CTRL-ALT** to return focus to the host machine.

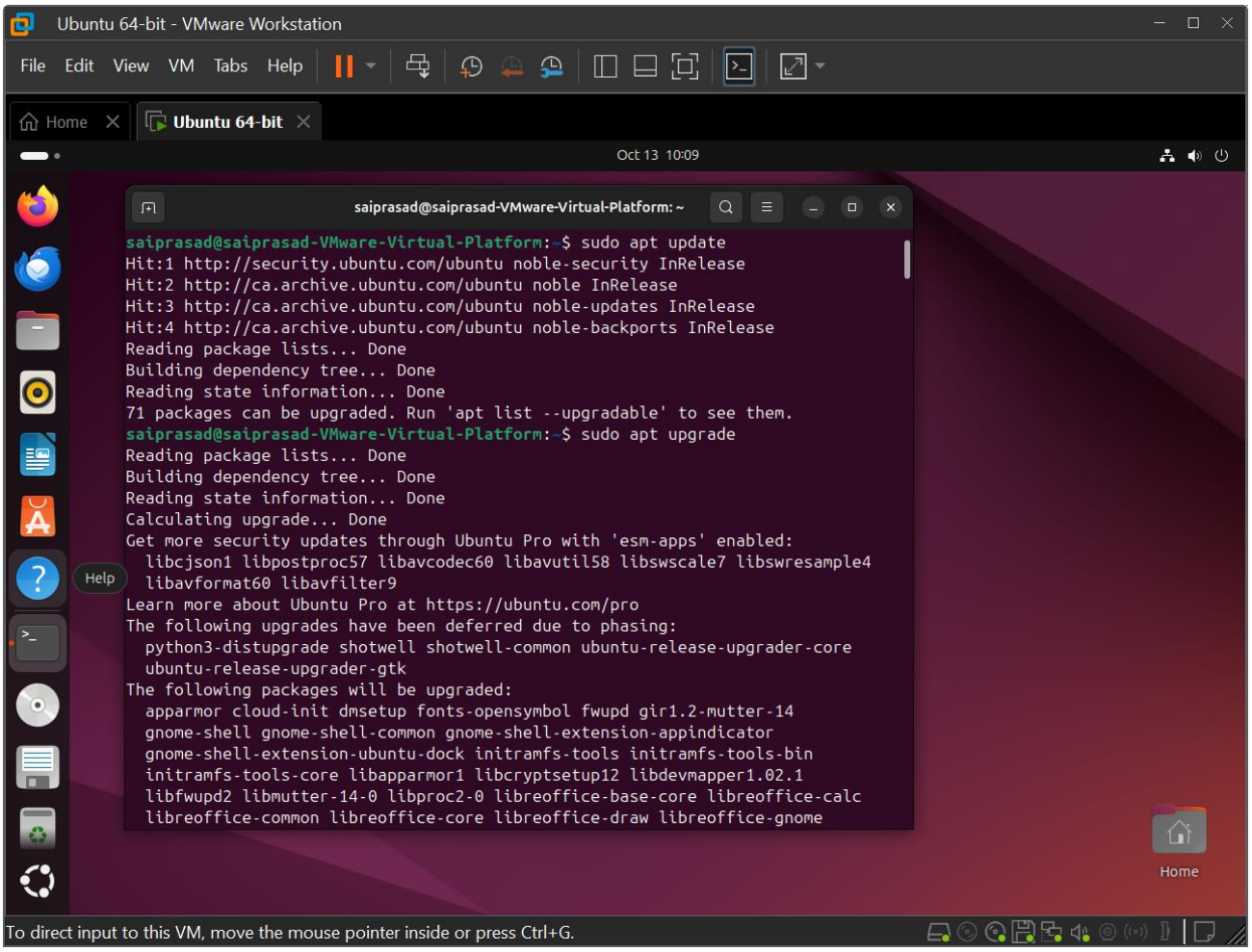
b) In the Ubuntu VM, click the **Show Applications** button at the bottom of the Ubuntu Launcher Bar on the left of the screen. Click in the search bar, type **Terminal**, and click the **Terminal** icon. Alternatively, pressing CTRL-ALT-T will open up a terminal.



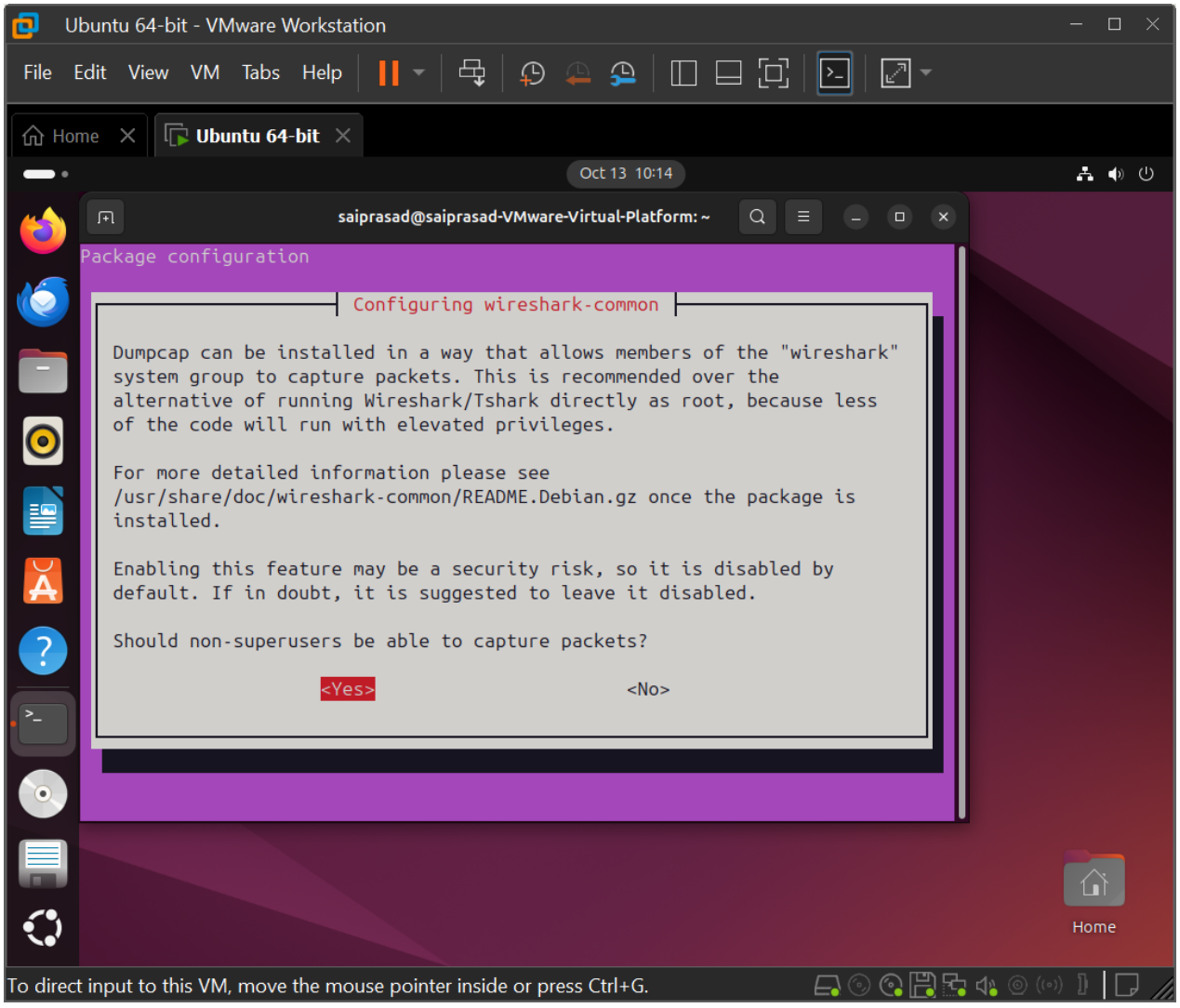
c) Execute the following two commands. The first downloads package information from every configured source. The second upgrades all installed packages to their latest versions.

sudo apt update

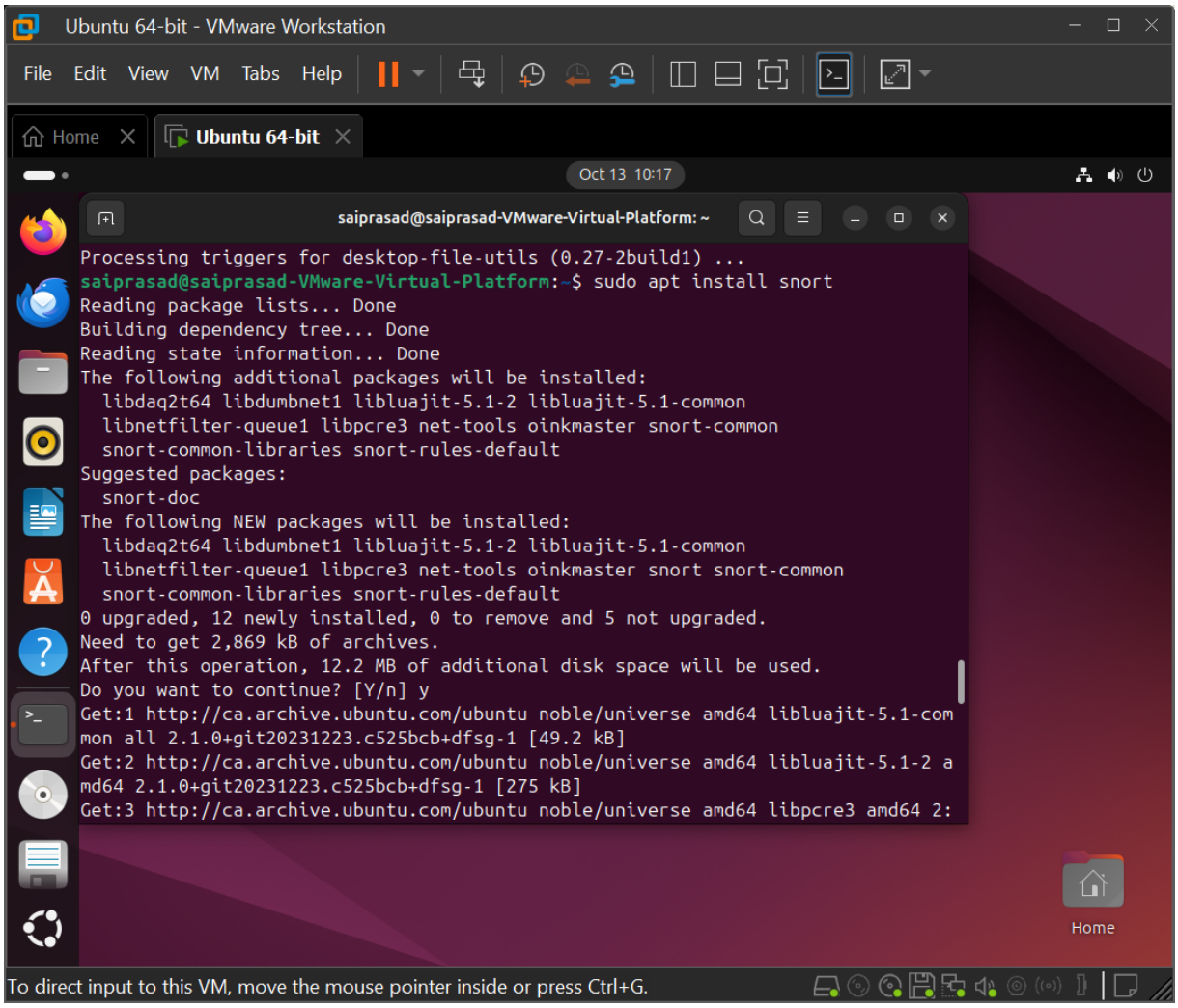
sudo apt upgrade



d) Unlike Kali Linux, Ubuntu does not come with Wireshark, the world-renowned packet sniffer. Execute the following to command to download and install it: **sudo apt install wireshark**
Put in your password and press **ENTER** when prompted now and throughout this lab. Type **Y** and press **ENTER** when prompted to continue. At the "Should Non-superusers Be Able to Capture Packets?" question, press the left arrow to select **Yes** and then press **ENTER**.



e) Enter the following command to download and install Snort: **sudo apt install snort**
Type **Y** and press **ENTER** when prompted to continue. At the Configuring Snort screen, press **ENTER** to select **OK**. In the Interface(s) Which Snort Should Listen On: textbox, using BACKSPACE, change the eth0 entry to ens33, which is the interface name used by Ubuntu. Press **ENTER** to select **OK**. If you see an Invalid Interface message, press ENTER to select OK. You will be brought back to the first screen again. Once again, press **ENTER** to select **OK**, and then press **ENTER** again to select **OK** with ens33 still in the textbox from before. You will get that Invalid Interface message again. Press **ENTER** to select **OK**. This time, the installation completes. This is a known bug for this version of Snort at the time of writing, and it may be resolved by the time you are doing this lab exercise. Even with that pushback from the Snort installer, Snort has been successfully installed.



Activity 1: Snort Sniffer Mode

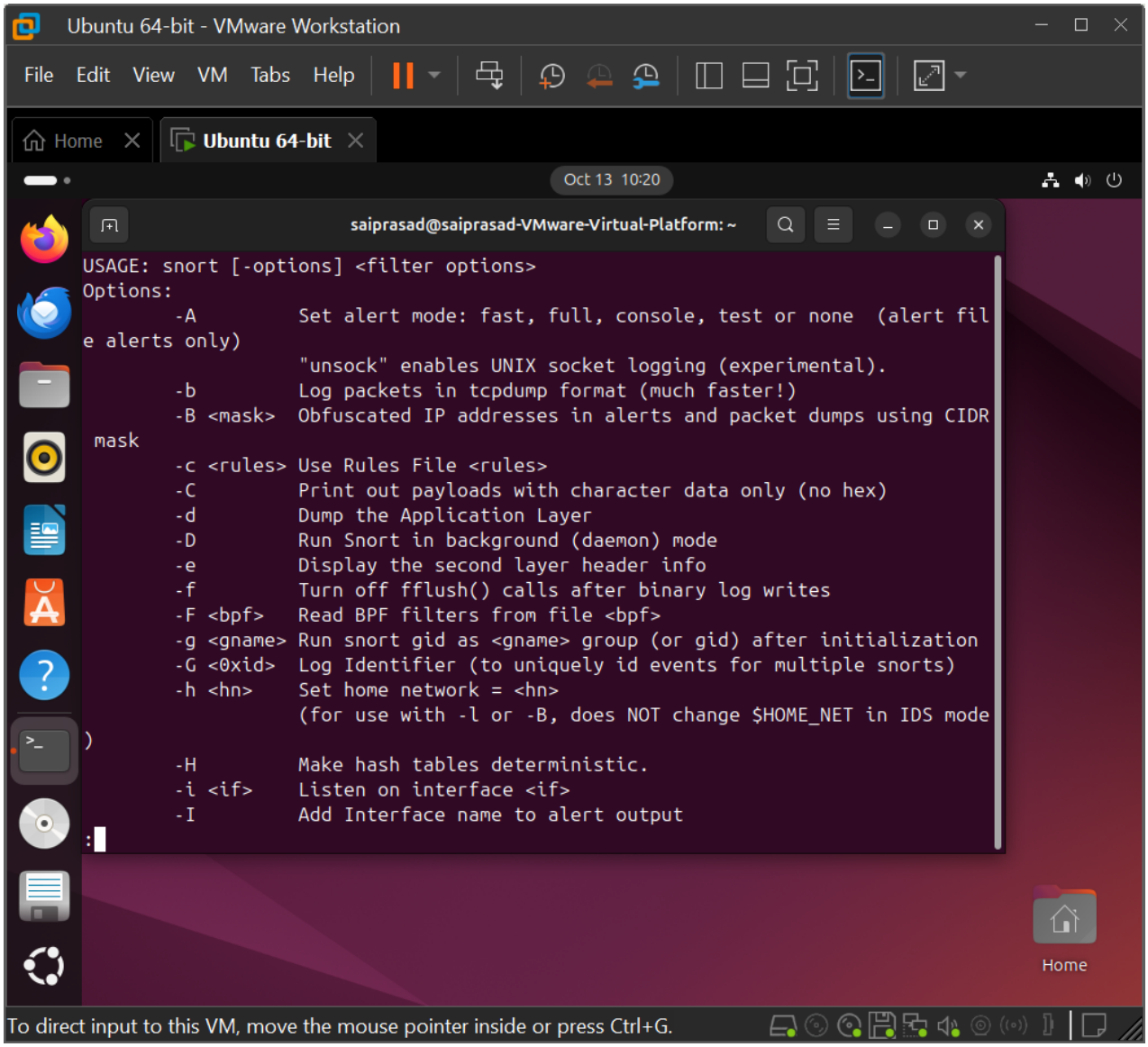
In this lab exercise, you will use Snort as a packet sniffer. Just like Wireshark, Snort can sniff packets.
The fields and values that Snort sniffs are displayed to the console in Sniffer mode.

- In the Ubuntu VM, press **CTRL-ALT-T** to open up a new console. Press **ENTER** after every command.

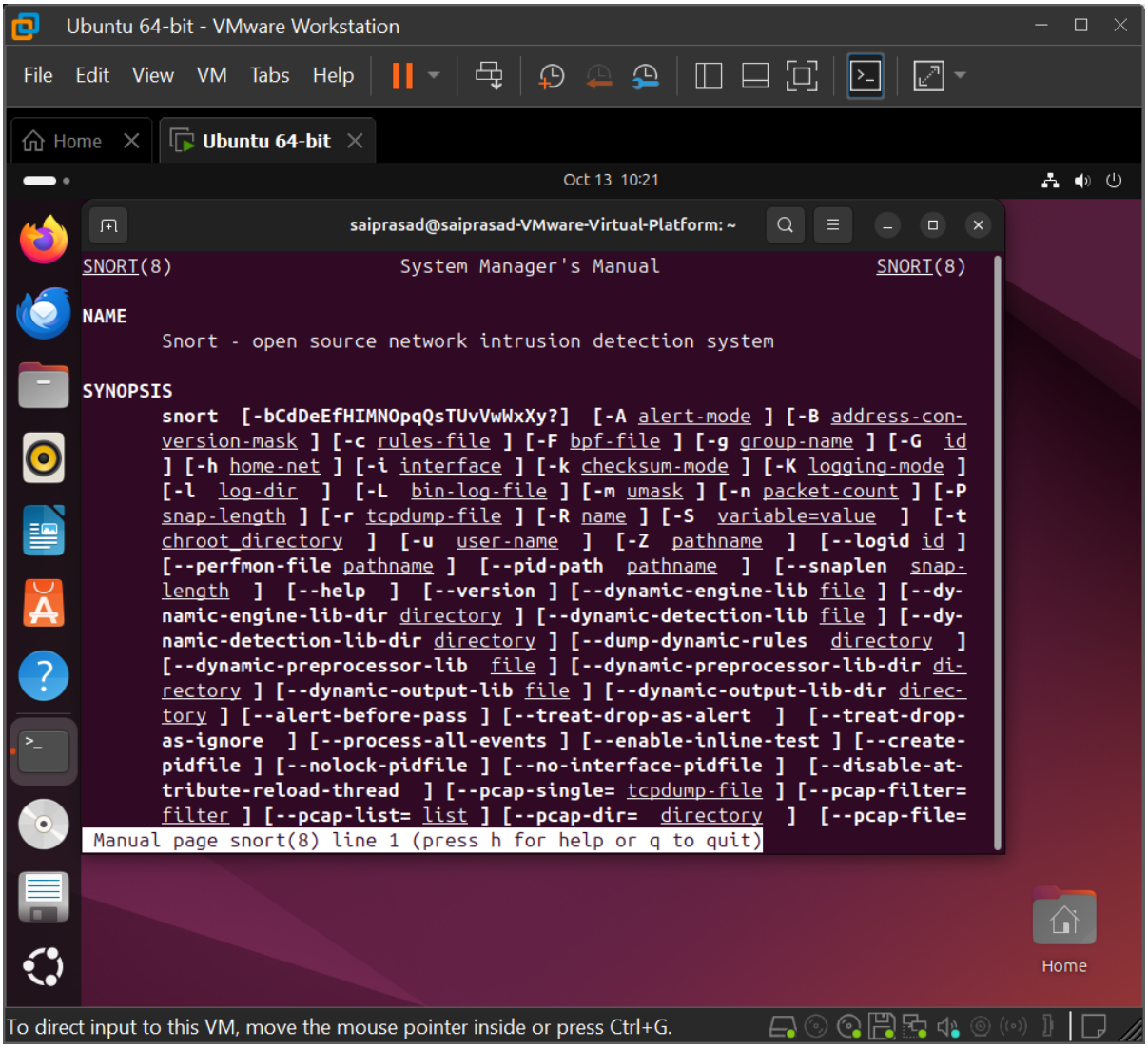
Step 1: Examine the Snort help output and man page entry, as well as the version of Snort.

a) Look at the Snort help output: **snort -h | less**

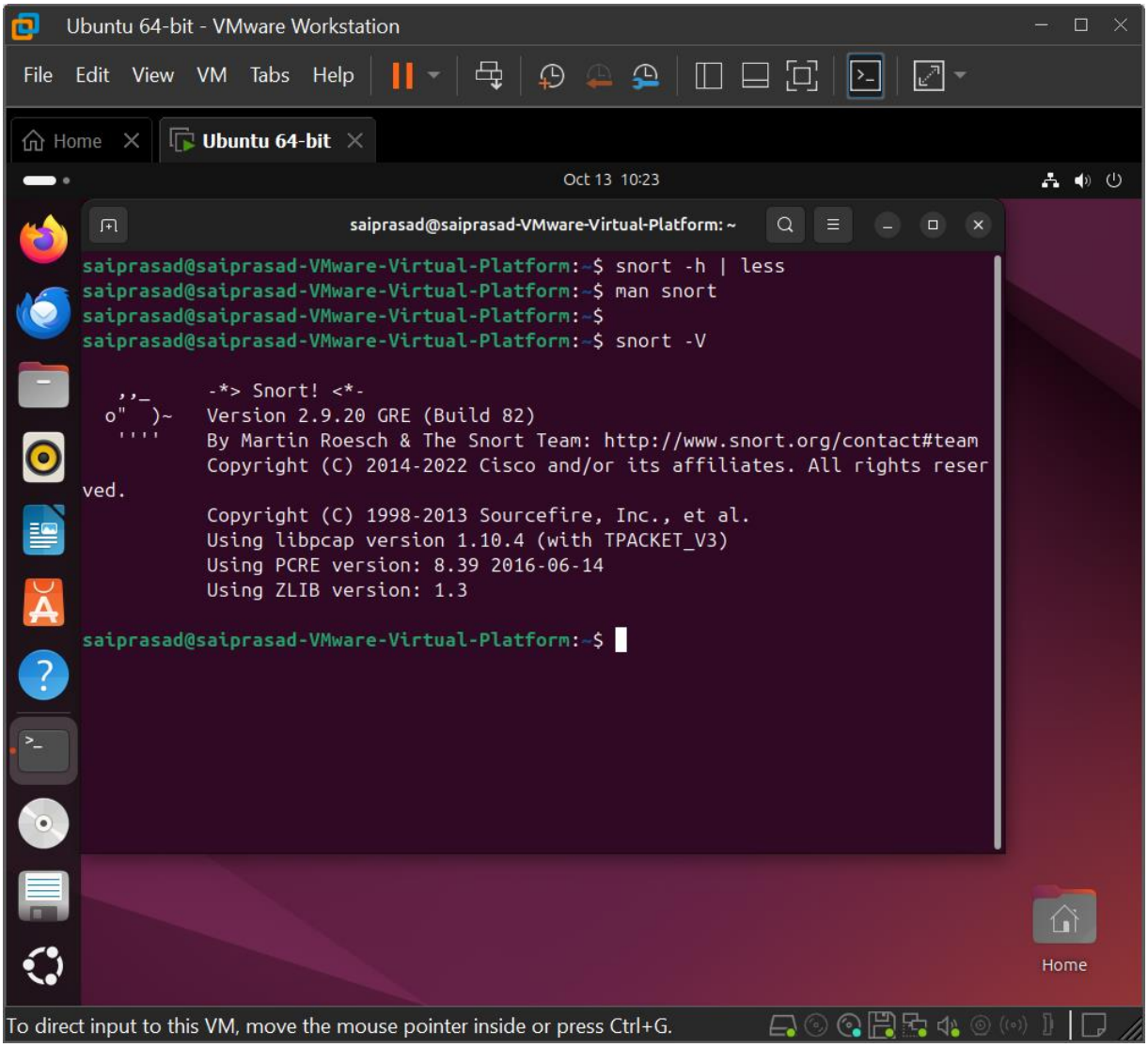
Advance line by line by pressing **ENTER**. Advance page by page with the spacebar. You can use the up and down arrows to move back and forth. Type **q** to quit.



b) Check out the man page entry for Snort: **man snort**

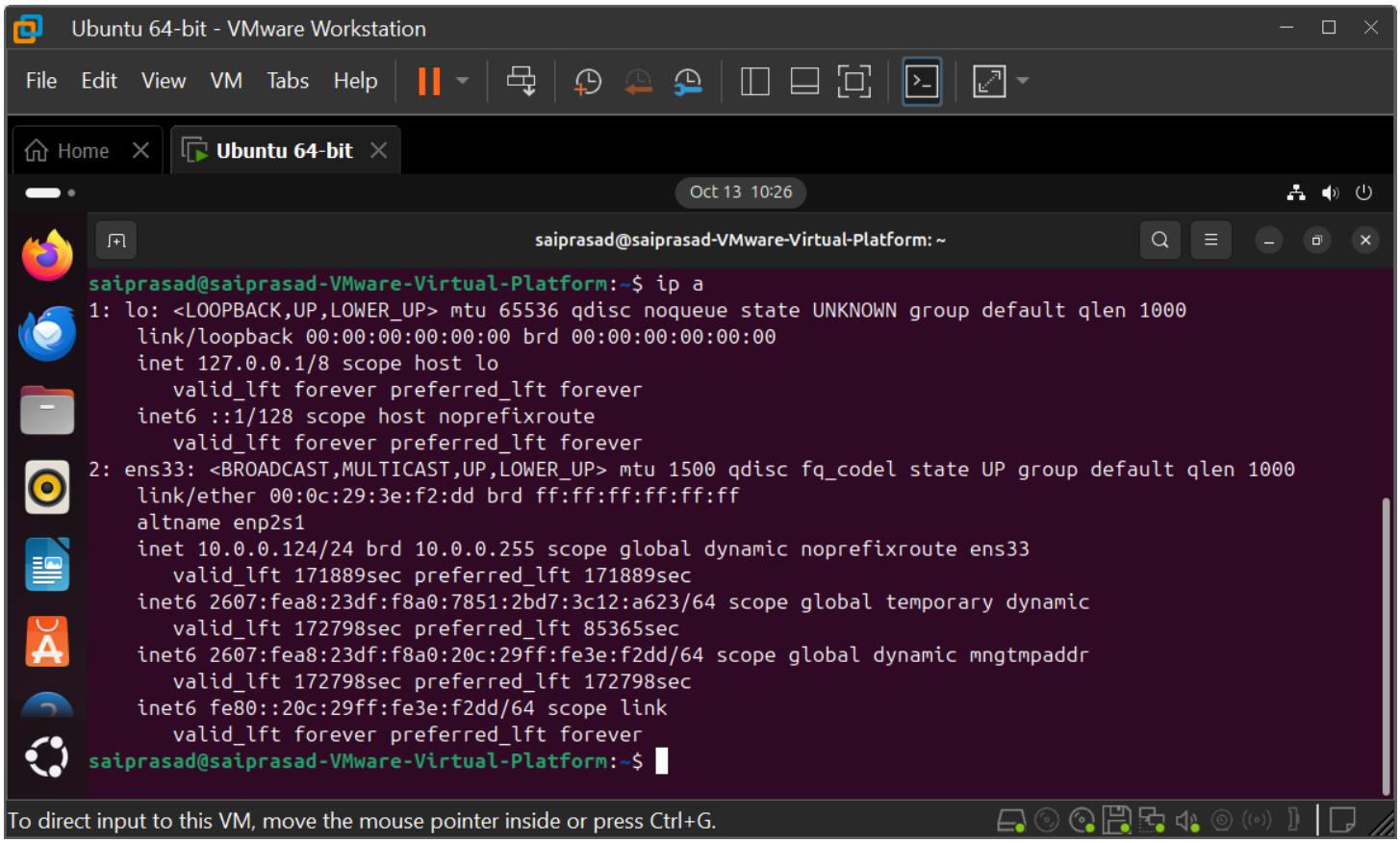


c) See the version number of Snort: **snort -V**



Step 2: Get your VM's IP address and start using Snort, generating output of certain Layer 3 and (if applicable) Layer 4 header information.

a) Find the Ubuntu VM's IP address by entering the following command: **ip a**
It is listed after inet in the ens33 interface section.

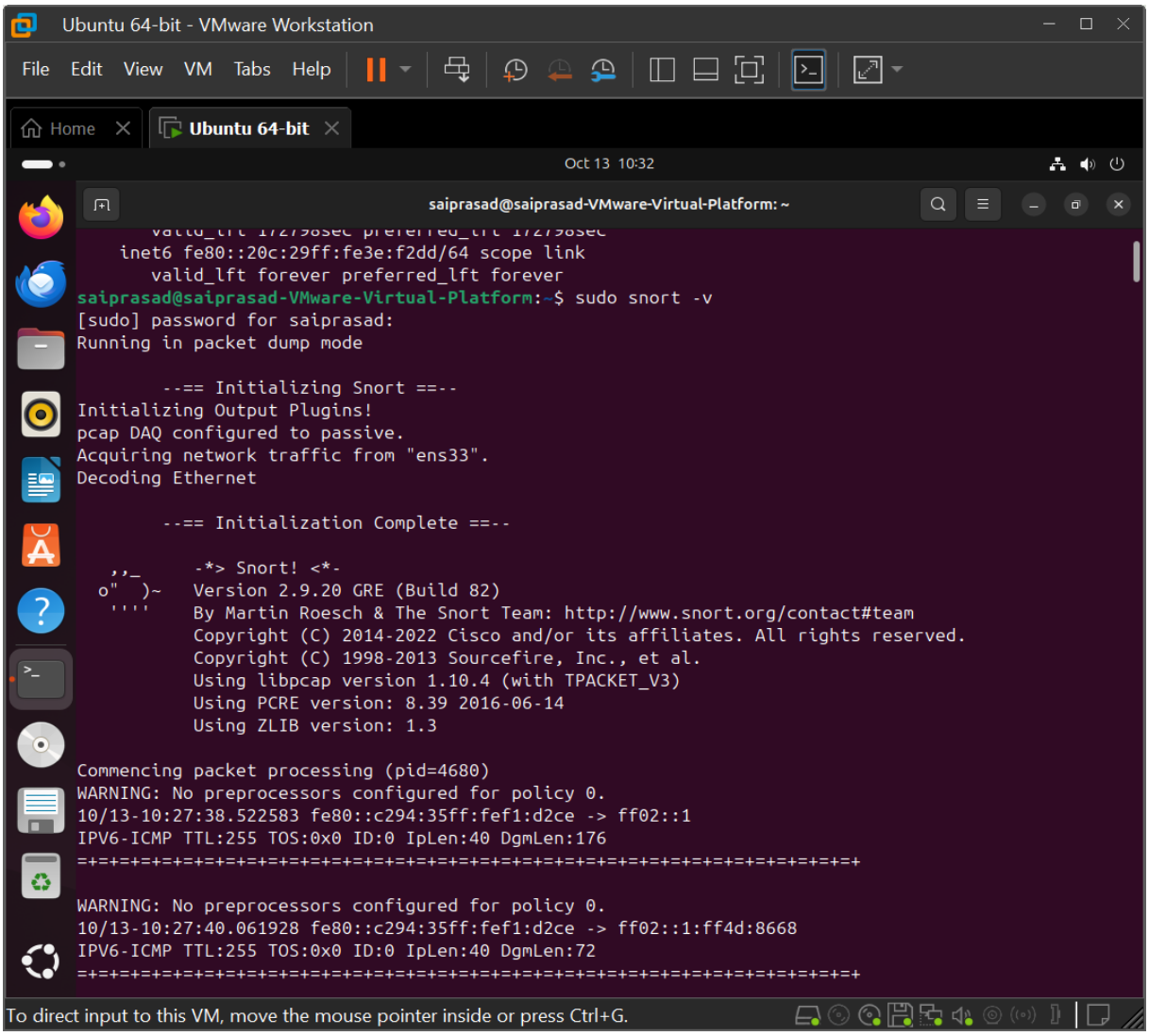


b) Enter the following command: **sudo snort -v**

The following is from the Snort man page:

-v Be verbose. Prints packets out to the console. There is one big problem with verbose mode: it is slow. If you are doing IDS work with Snort, do not use the '-v' switch, you WILL drop packets Snort runs and only displays information from the Layer 3 header, although not all fields and values are included. Snort will also display port numbers from Layer 4 headers when TCP or UDP is used. Ignore the "WARNING: No preprocessors configured for policy 0." message that is repeatedly shown. We will fix that shortly. Any packet going in and out of the Ubuntu VM will be displayed in the console.

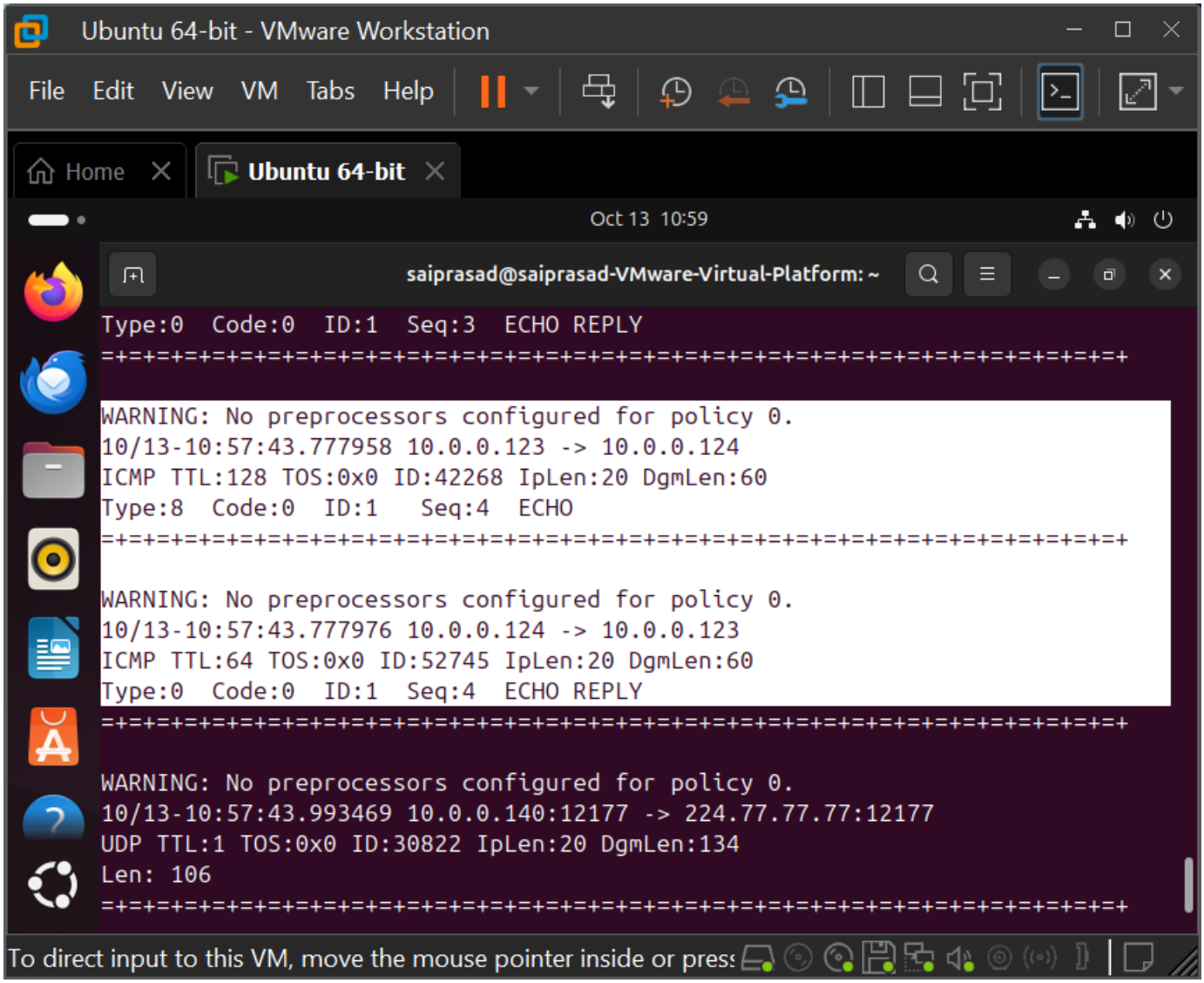
You can stop Snort with **CTRL-C**. It might take a few seconds for Snort to stop and for information such as the following to be displayed: how long Snort ran for, how many packets it processed, the number of packets per minute and packets per second, memory usage summary, packet I/O totals, and a breakdown by protocol.



c) Enter the following command: **sudo snort -v -i ens33**

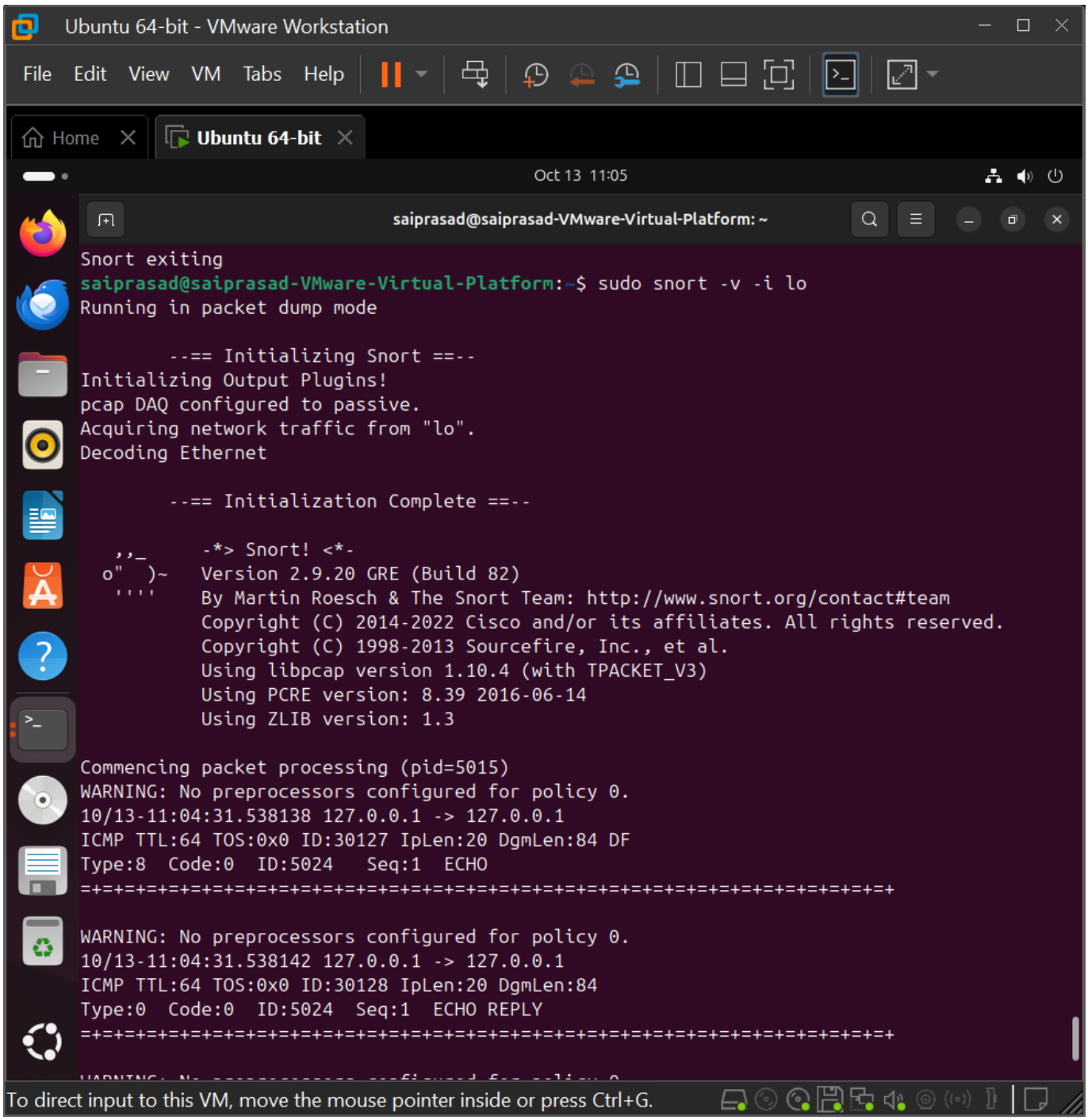
If you had multiple interfaces, the -i option would be the way to specify a certain interface that Snort should sniff on. Try it now, although with the current setup featuring one nonloopback interface, the same thing will happen with or without the -i option.

From the Windows 10 VM or host machine, ping the IP address of the Ubuntu VM. You should be able to see the pings in the output of Snort. **Take the screenshot.** Break out with **CTRL-C**.



d) Enter the following command: **sudo snort -v -i lo**

e) To see the difference, open up a new terminal, ping the loopback address (127.0.0.1), and observe the output in the terminal in which Snort is sniffing. **Take the screenshot.** You can break out with **CTRL-C**.



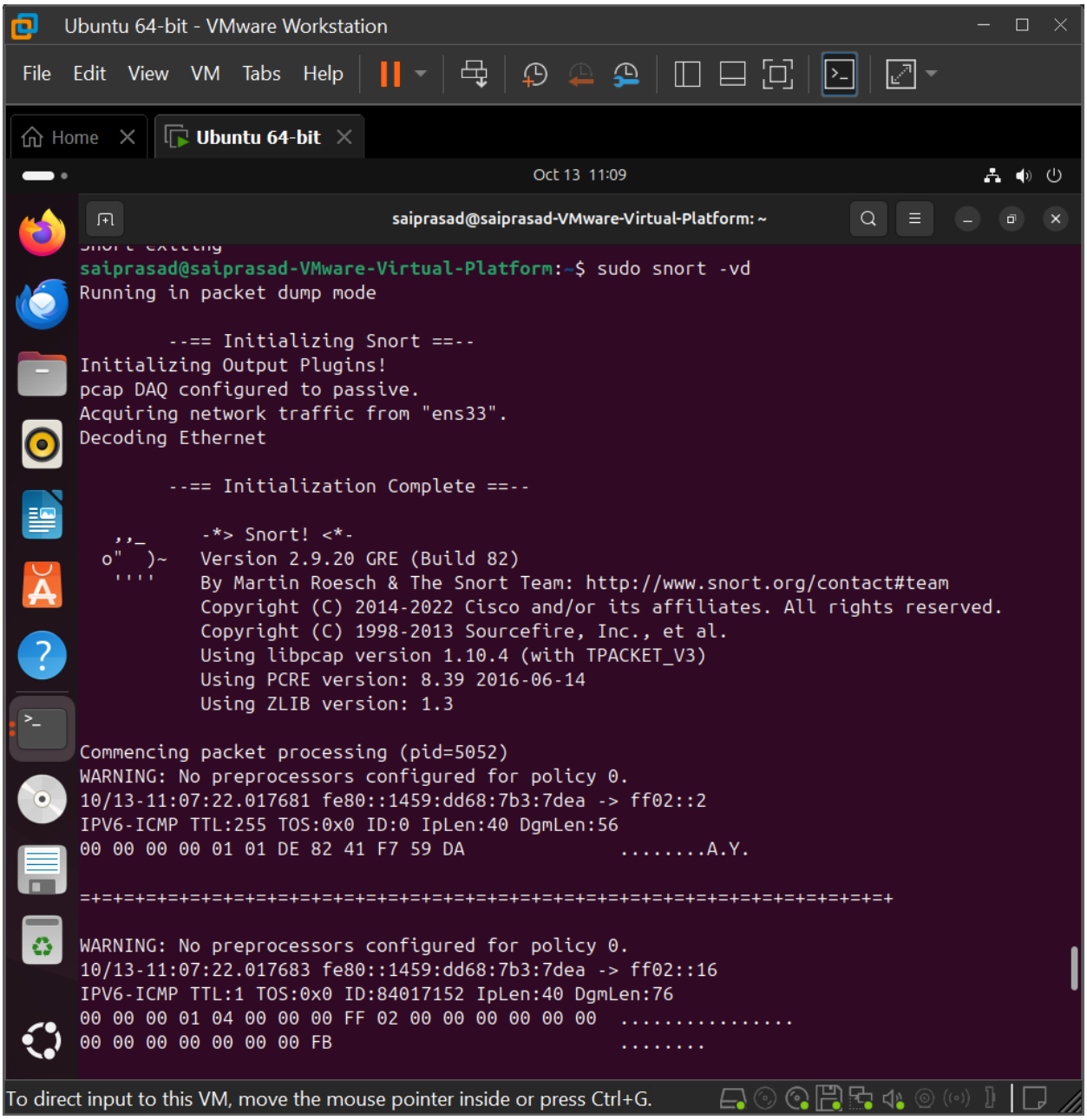
Step 3: Run Snort, generating upper layer data, in addition to Layer 3 and Layer 4 header information.

a) Enter the following command: **sudo snort -vd**

The following is from the Snort man page:

-d Dump the application layer data when displaying packets in verbose or packet logging mode. After running Snort with the -v and -d options together (-vd), stop Snort with CTRL-C.

Snort runs and shows the upper layer data, in addition to information from Layer 3 and Layer 4 (when TCP or UDP is used). **Take the screenshot.**

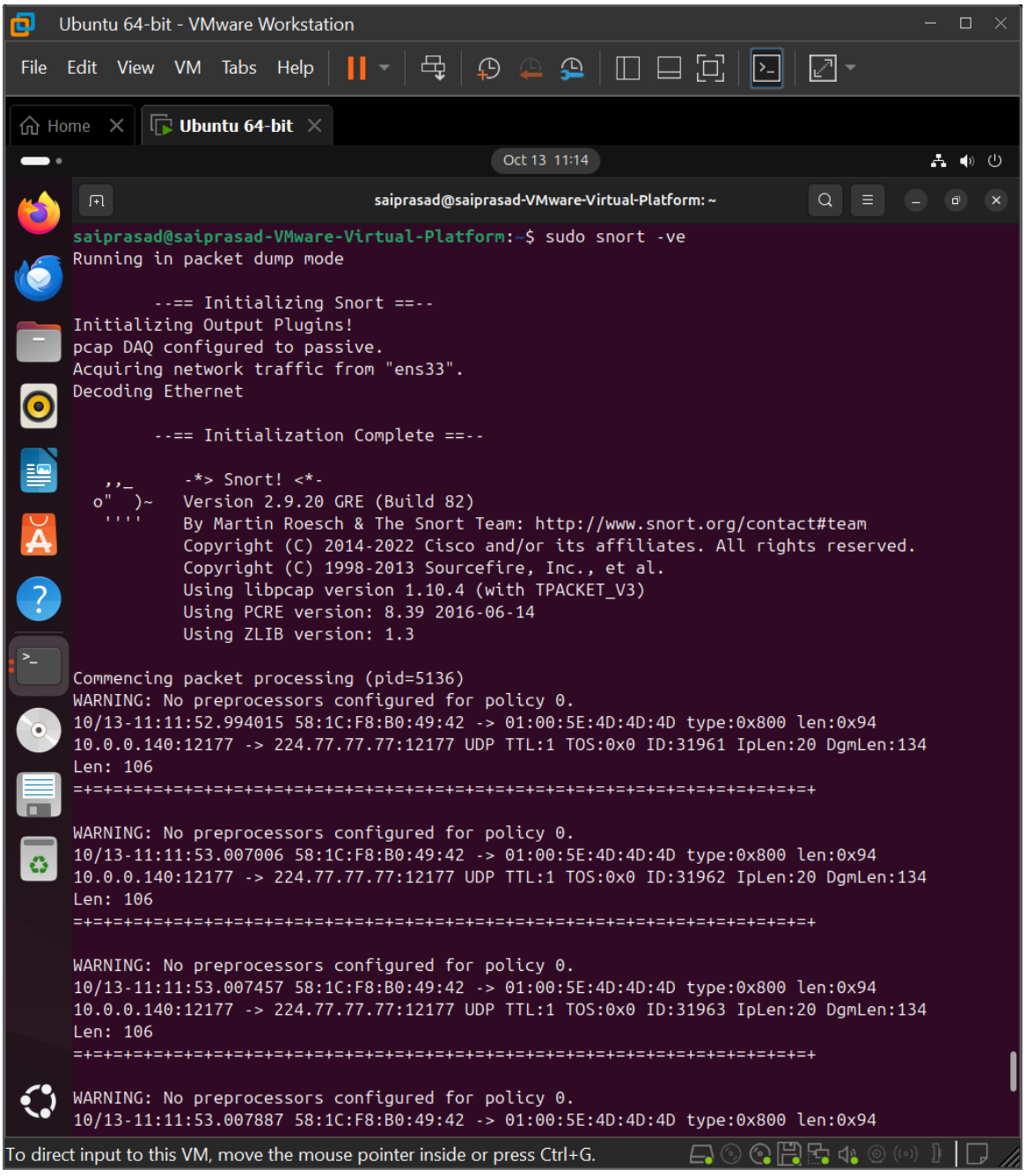


b) Enter the following command:

sudo snort -ve

The following is from the Snort man page:

-e Display/log the link layer packet headers. After running Snort with the -v and -e options together (-ve), stop Snort with CTRL-C. Snort runs and shows information from Layer 2, in addition to information from Layer 3. **Take the screenshot.**



c) Enter the following command:

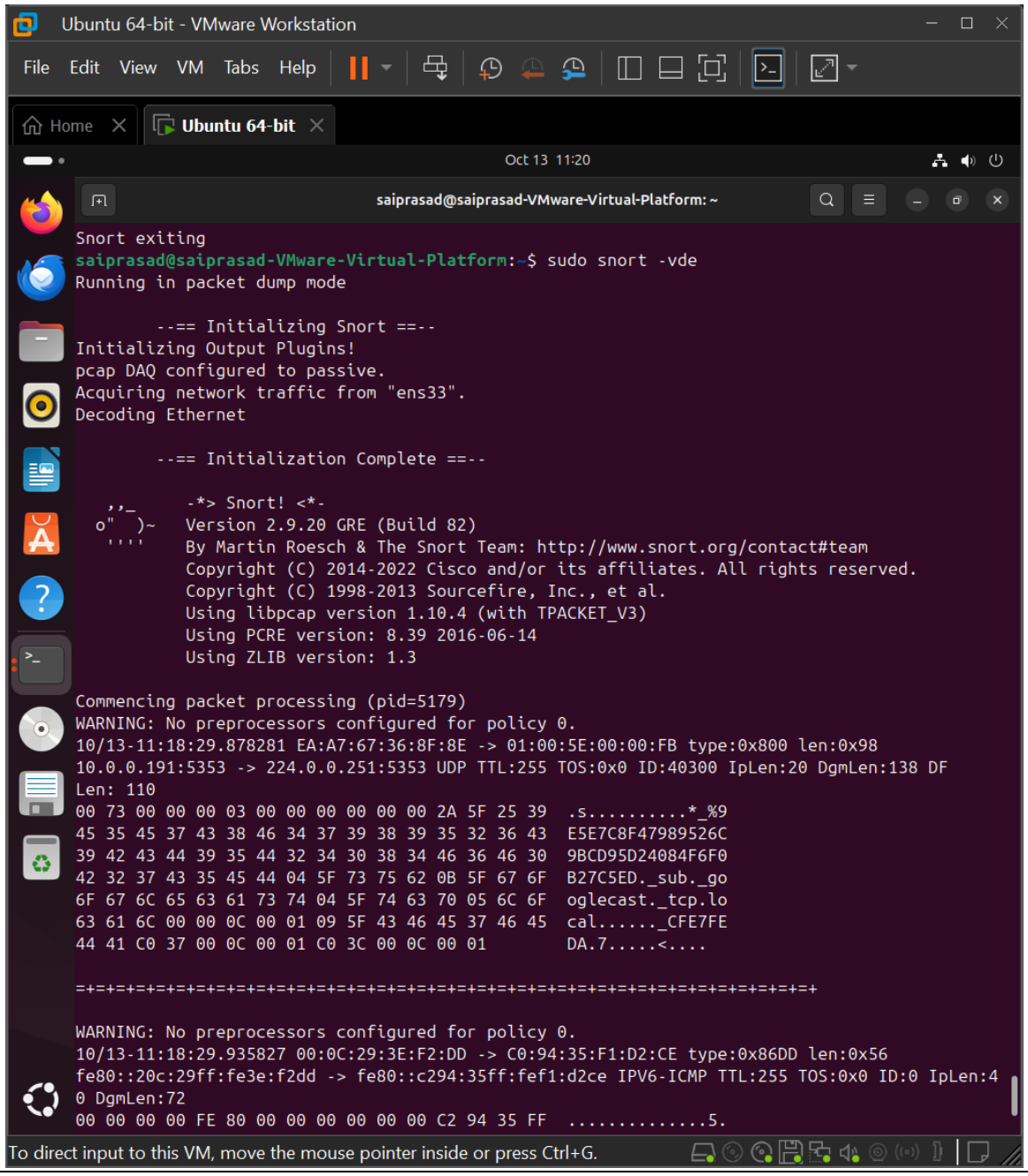
sudo snort -vde

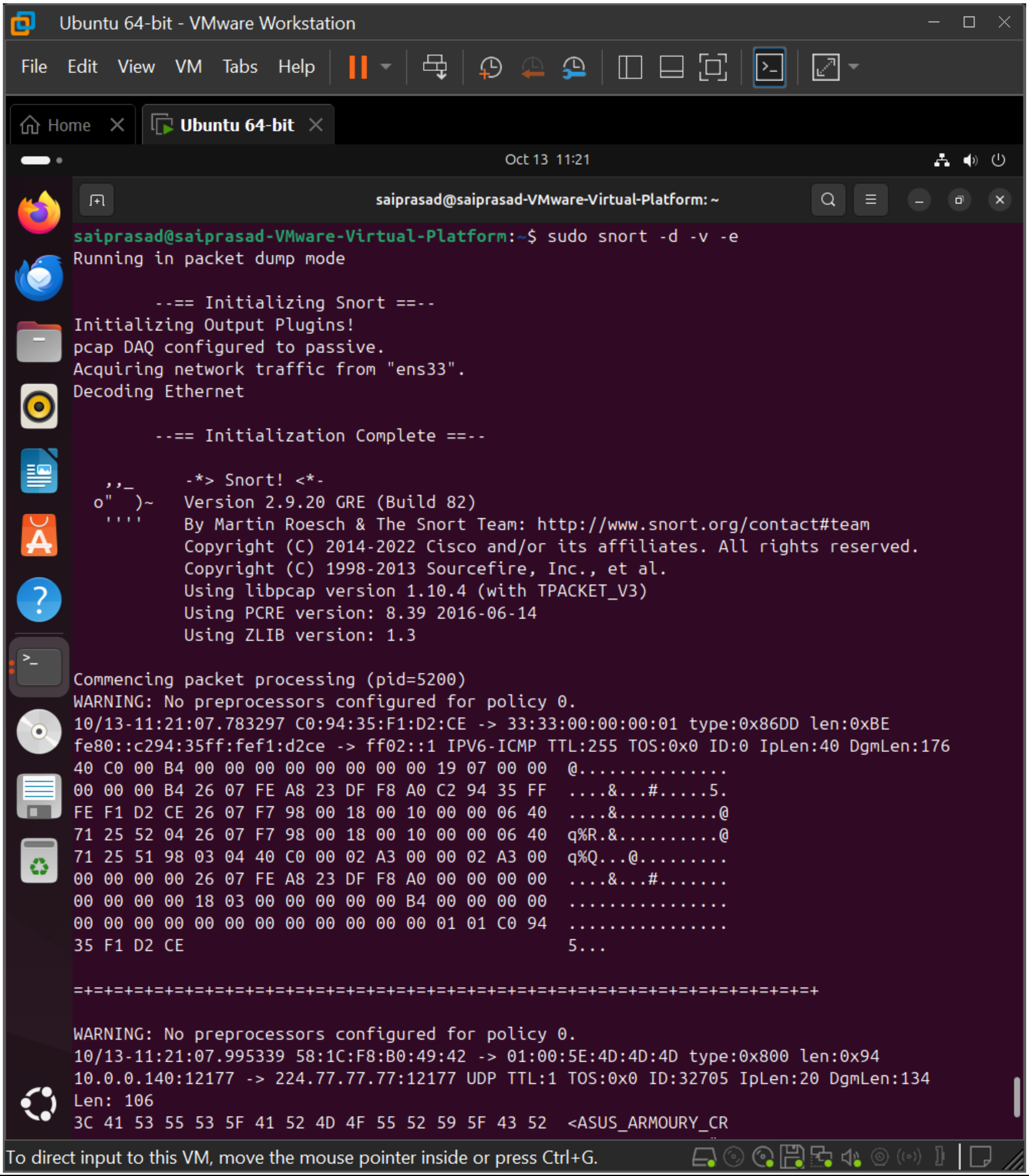
After running Snort with the -v, -d, and -e options together (-vde), stop Snort with CTRL-C.

Snort runs and shows header information from Layer 2, Layer 3, and upper-layer data.

The switches could have been typed separately, and in a different order, for the same results, like this: **sudo snort -d -v -e**

Furthermore, the -v option becomes redundant with either the -d or -e option. In other words, if you were to leave off the -v option, sudo snort -de would have done the same thing.




Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home
Ubuntu 64-bit
Oct 13 11:21
saiprasad@saiprasad-VMware-Virtual-Platform: ~
saiprasad@saiprasad-VMware-Virtual-Platform:~$ sudo snort -d -v -e
Running in packet dump mode
--== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "ens33".
Decoding Ethernet
--== Initialization Complete ==--
-*> Snort! <*-
Version 2.9.20 GRE (Build 82)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.10.4 (with TPACKET_V3)
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.3
Commencing packet processing (pid=5200)
WARNING: No preprocessors configured for policy 0.
10/13-11:21:07.783297 C0:94:35:F1:D2:CE -> 33:33:00:00:00:01 type:0x86DD len:0xBE
fe80::c294:35ff:fef1:d2ce -> ff02::1 IPV6-ICMP TTL:255 TOS:0x0 ID:0 IpLen:40 DgmLen:176
40 C0 00 B4 00 00 00 00 00 00 00 00 19 07 00 00 @...............
00 00 00 B4 26 07 FE A8 23 DF F8 A0 C2 94 35 FF ....&...#.....5.
FE F1 D2 CE 26 07 F7 98 00 18 00 10 00 00 06 40 ....&..........@
71 25 52 04 26 07 F7 98 00 18 00 10 00 00 06 40 q%R.&..........@
71 25 51 98 03 04 40 C0 00 02 A3 00 00 02 A3 00 q%Q...@.........
00 00 00 00 26 07 FE A8 23 DF F8 A0 00 00 00 00 ....&...#.......
00 00 00 00 18 03 00 00 00 00 00 B4 00 00 00 00 ....
00 00 00 00 00 00 00 00 00 00 00 00 01 01 C0 94 ....
35 F1 D2 CE 5...
=+
WARNING: No preprocessors configured for policy 0.
10/13-11:21:07.995339 58:1C:F8:B0:49:42 -> 01:00:5E:4D:4D:4D type:0x800 len:0x94
10.0.0.140:12177 -> 224.77.77.77:12177 UDP TTL:1 TOS:0x0 ID:32705 IpLen:20 DgmLen:134
Len: 106
3C 41 53 55 53 5F 41 52 4D 4F 55 52 59 5F 43 52 <ASUS_ARMOURY_CR
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Activity 2: Snort Packet Logger Mode

In this lab exercise, you will sniff with Snort, but instead of viewing the output in the console, you will log the packets to a file.

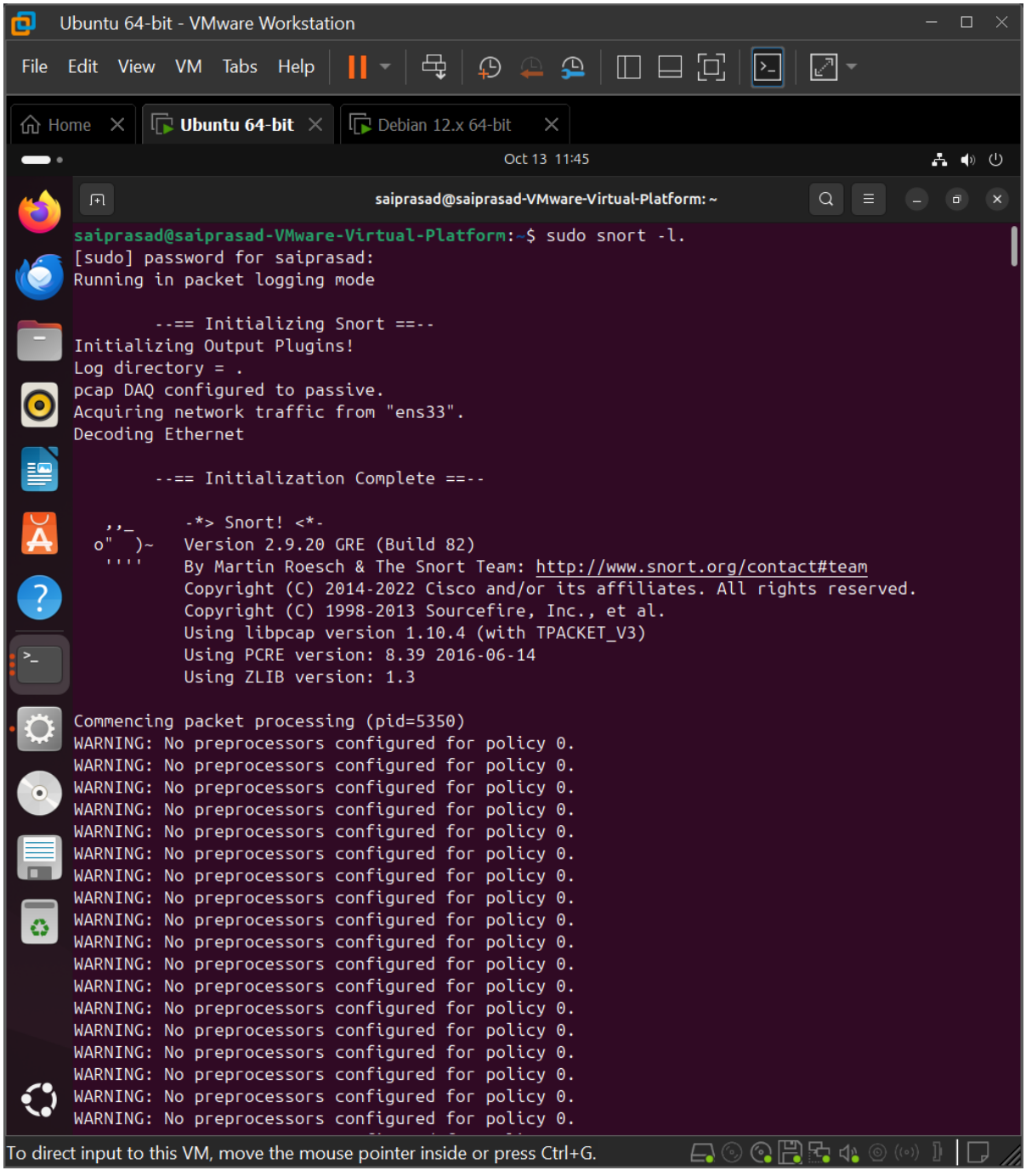
You can instruct Snort to record packets to a file by using the -l option and specifying a directory that the log should be sent to. Then the log file can be opened up in a packet sniffer, like Wireshark.

- In the Ubuntu VM, press **CTRL-ALT-T** to open up a new terminal. Press **ENTER** after every command.

Step 1: Log packets that Snort sniffs to a file.

a) Enter the following command: **sudo snort -l .**
The dot at the end of the command represents the current directory. The following is from the Snort man page:
-l log-dir
Set the output logging directory to log-dir. All plain text alerts and packet logs go into this directory. If this option is not specified, the default logging directory is set to /var/log/snort.
You do not need -v, -d, or -e, contrary to documentation out there, including Snort's own documentation.

b) Send a ping from the Windows 10 VM or host machine to the Ubuntu VM running Snort. Stop Snort with **CTRL-C**. **Take the screenshot.**

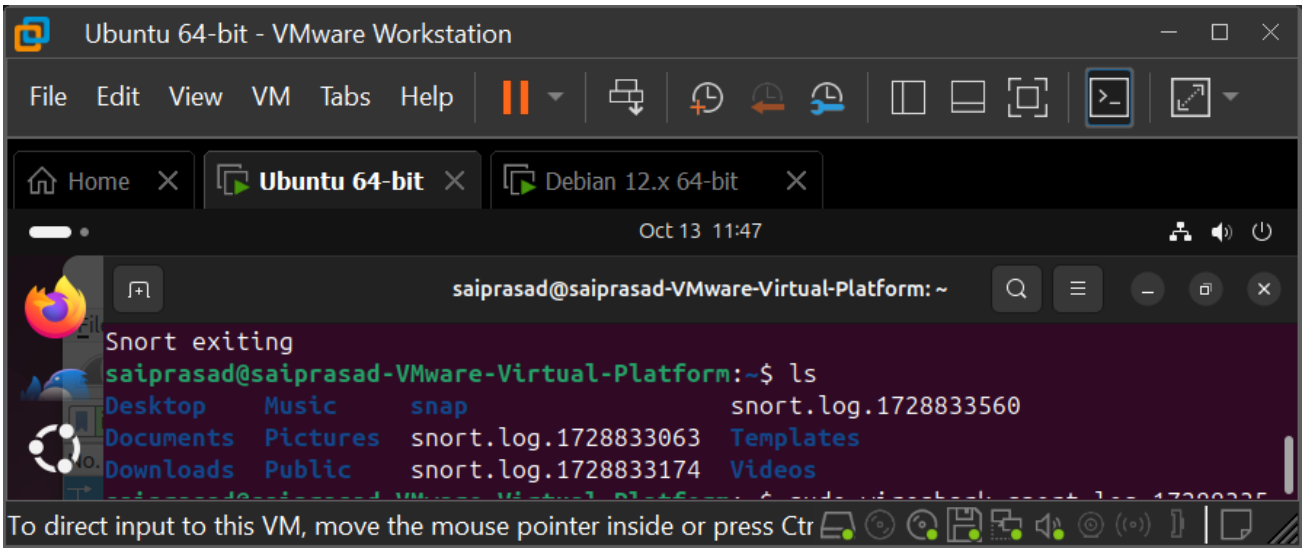

Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home
Ubuntu 64-bit
Debian 12.x 64-bit
Oct 13 11:45
saiprasad@saiprasad-VMware-Virtual-Platform: ~
saiprasad@saiprasad-VMware-Virtual-Platform:~$ sudo snort -l.
[sudo] password for saiprasad:
Running in packet logging mode
--== Initializing Snort ==--
Initializing Output Plugins!
Log directory = .
pcap DAQ configured to passive.
Acquiring network traffic from "ens33".
Decoding Ethernet
--== Initialization Complete ==--
-*> Snort! <*-
Version 2.9.20 GRE (Build 82)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.10.4 (with TPACKET_V3)
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.3
Commencing packet processing (pid=5350)
WARNING: No preprocessors configured for policy 0.
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Step 2: Examine the logged packets.

a) Enter the following command:

ls

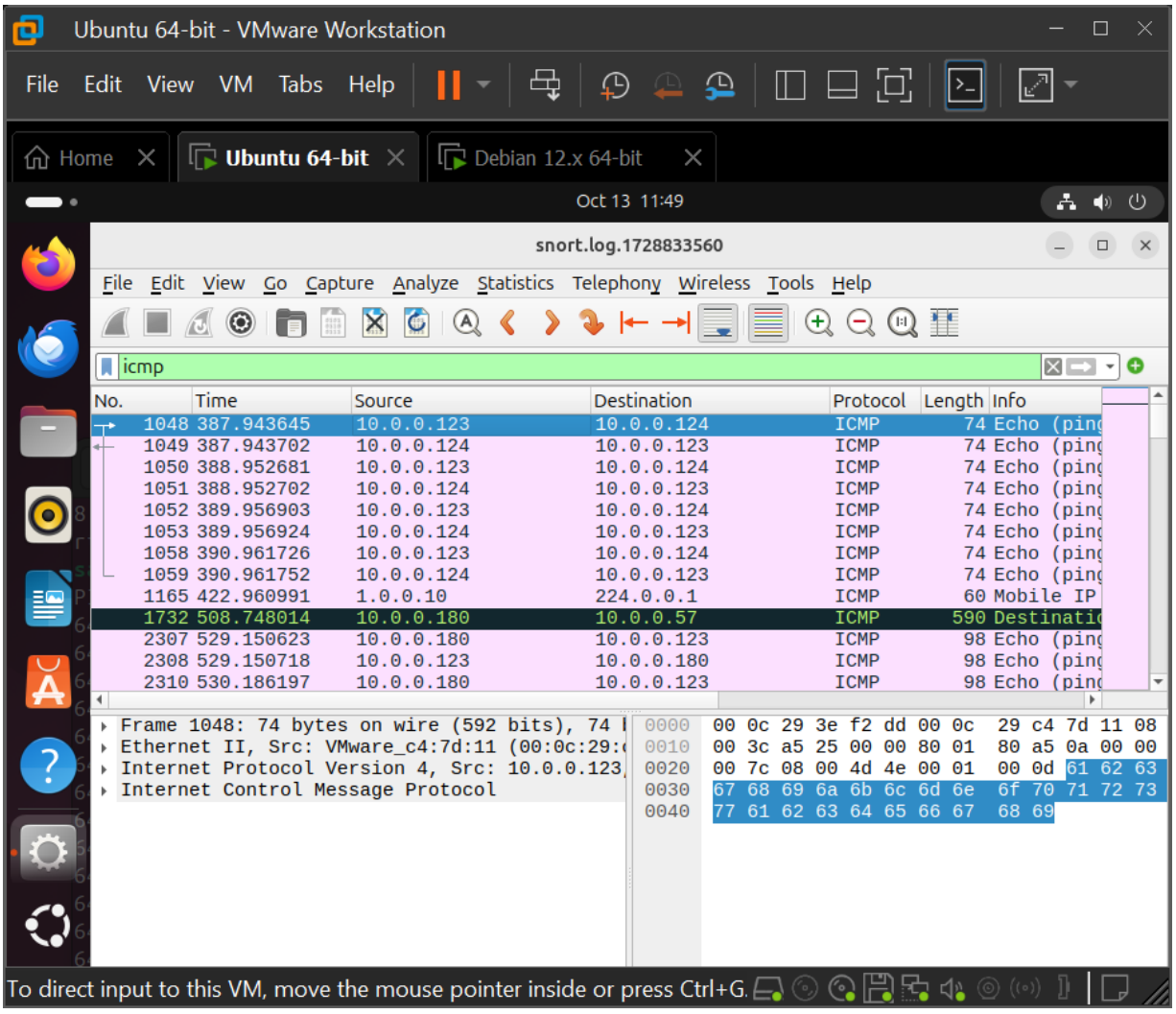
See the name of the log file generated.



b) Enter the following command: **sudo wireshark**

snort.log.159604382

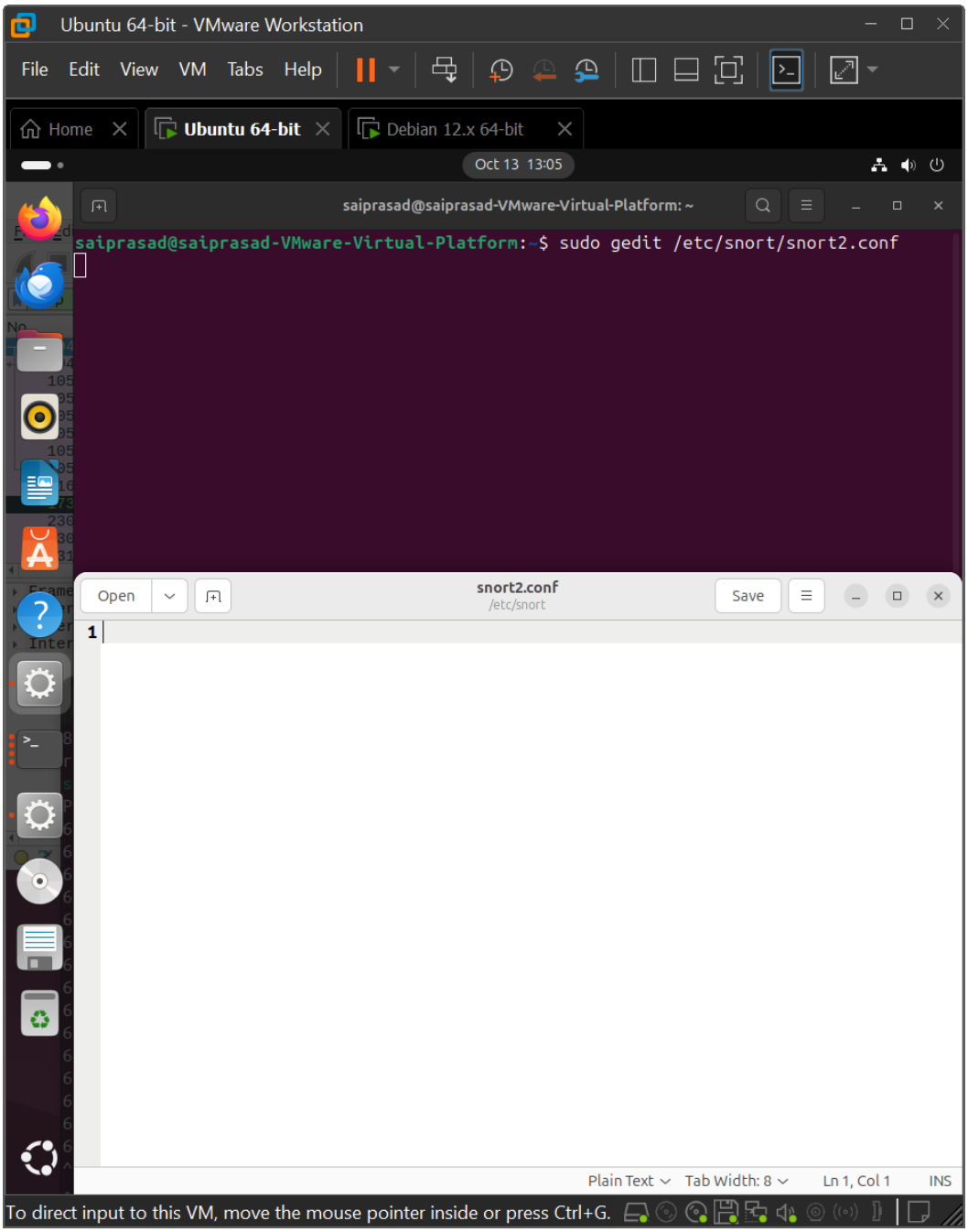
Wireshark and packet sniffing in general were introduced previously and will be explored further in this lab. For now, simply open up the logged packets in Wireshark. In the display filter bar (where it says “Apply a display filter…”), type icmp and press ENTER to see just the pings from Step 1b. **Take the screenshot.**



Activity 3: Snort NIDS Mode (Network Intrusion Detection System)

Step 1 Use custom and default Snort configuration files.

a) Before we use Snort's default configuration file and rules, let's write a simple one, to get started: **sudo gedit /etc**



Use the gedit text editor to create and edit the snort2.conf file, stored in /etc/snort, which is where the default snort.conf file is. For the variable HOME_NET (of type ipvar), use 192.168.2.0/24 if that is your network ID. Otherwise, modify it for your IP addressing scheme. The route command will show you your network ID. Add the following lines (using your network ID on line 3) to your configuration file:

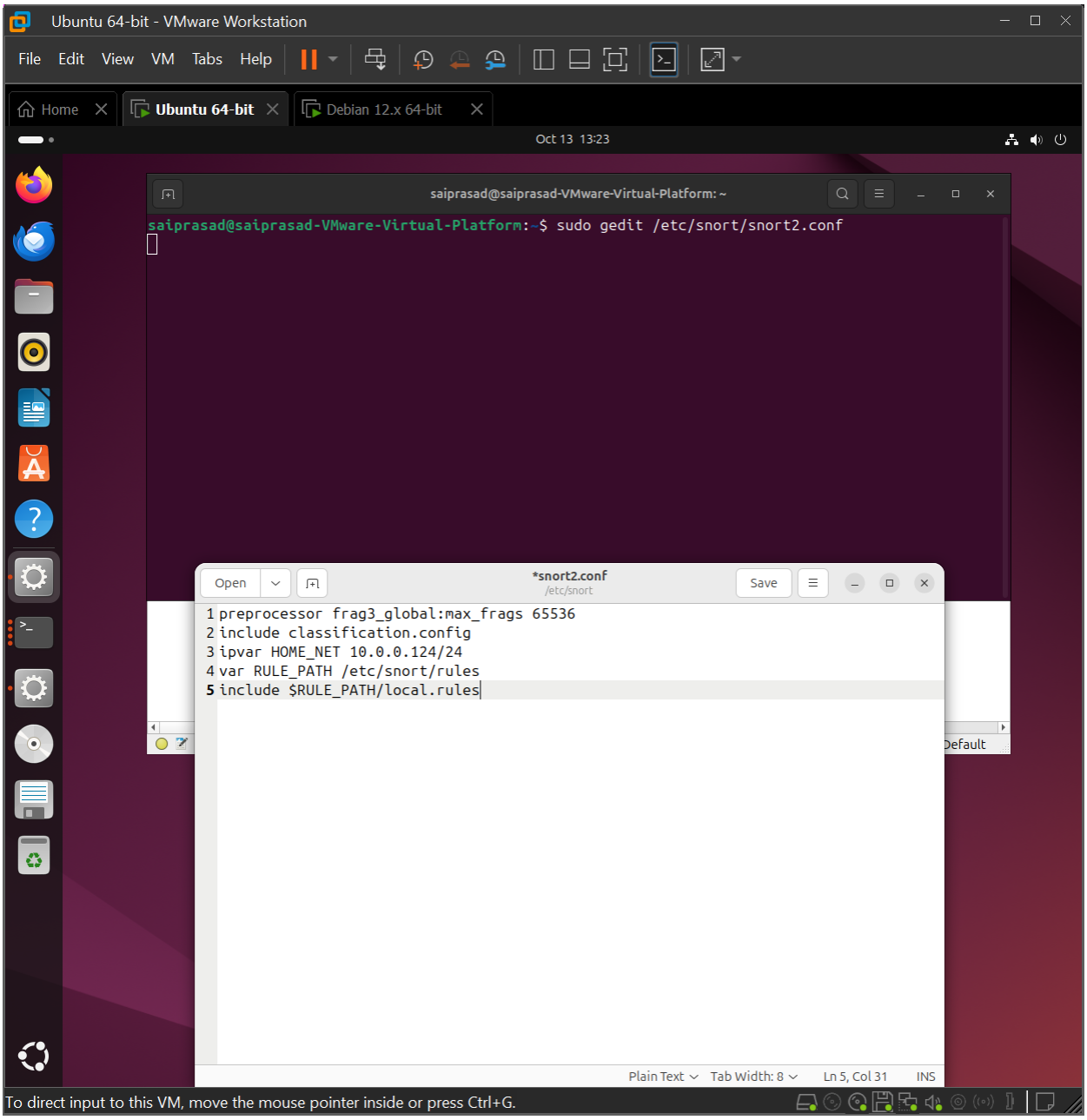
preprocessor frag3_global: max_frags 65536
include classification.config
ipvar HOME_NET 192.168.1.0/24
var RULE_PATH /etc/snort/rules
include $RULE_PATH/local.rules

Click the red X at the top right of the screen and then click the **Save** button. This adds a minimal set of lines to the co
use when started.

A default preprocessor is included to get rid of the message "WARNING: No preprocessors configured for policy 0." You can read more about preprocessors, including this specific one, at: http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node17.html The file, classification.config, is included.

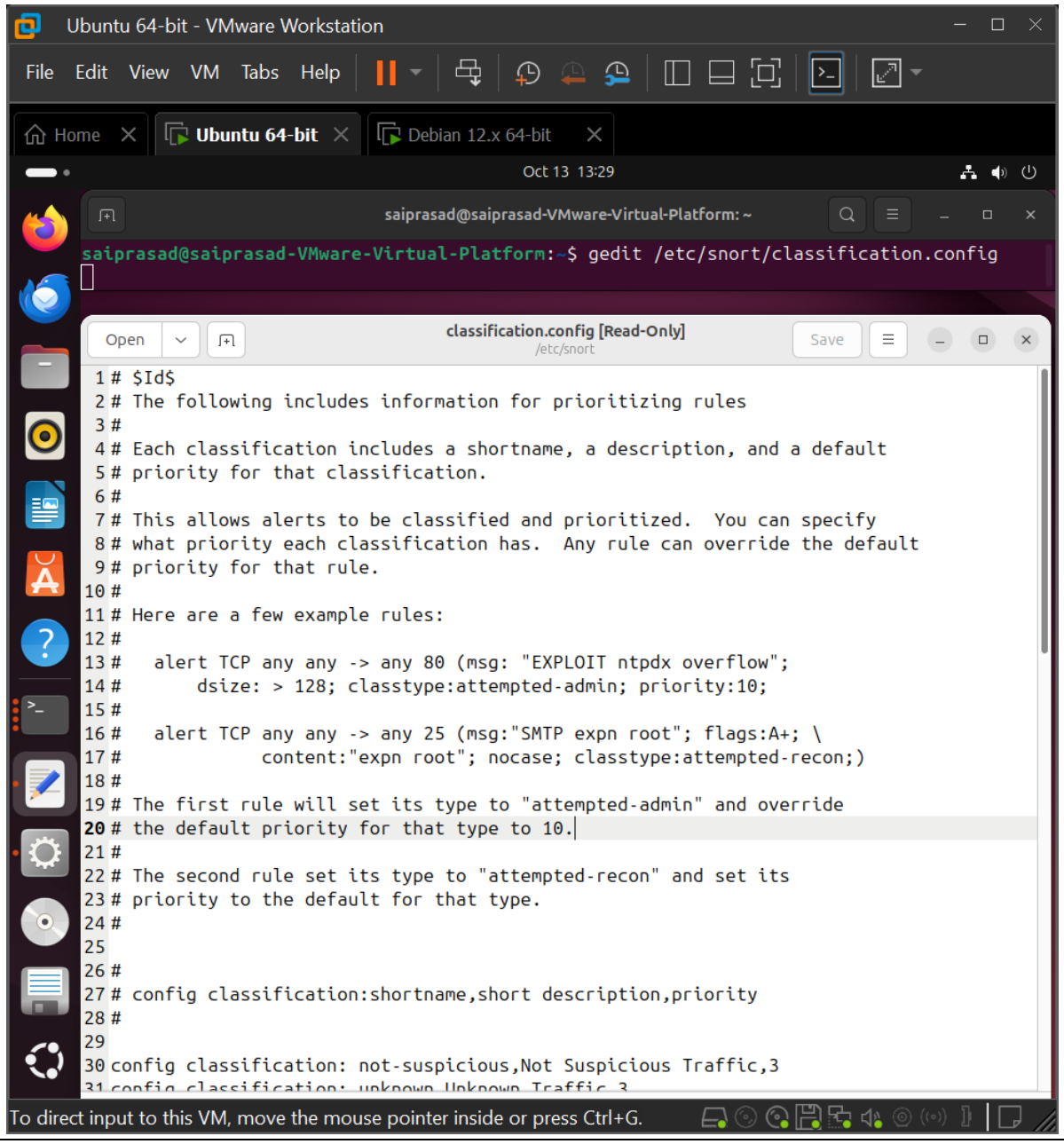
The following is from http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node16.html :

The include keyword allows other snort config files to be included within the snort.conf indicated on the Snort comm
#include from the C programming language, reading the contents of the named file, and adding the contents in the
statement appears in the file.


Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home
Ubuntu 64-bit
Debian 12.x 64-bit
Oct 13 13:23
saiprasad@saiprasad-VMware-Virtual-Platform: ~
saiprasad@saiprasad-VMware-Virtual-Platform:~$ sudo gedit /etc/snort/snort2.conf
Open
*snort2.conf
/etc/snort
Save
1 preprocessor frag3_global:max_frags 65536
2 include classification.config
3 ipvar HOME_NET 10.0.0.124/24
4 var RULE_PATH /etc/snort/rules
5 include $RULE_PATH/local.rules
Plain Text
Tab Width: 8
Ln 5, Col 31
INS
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

To see the classification.config file on your system, type:
gedit /etc/snort/classification.config
and then press **ENTER**. **Take the screenshot.**

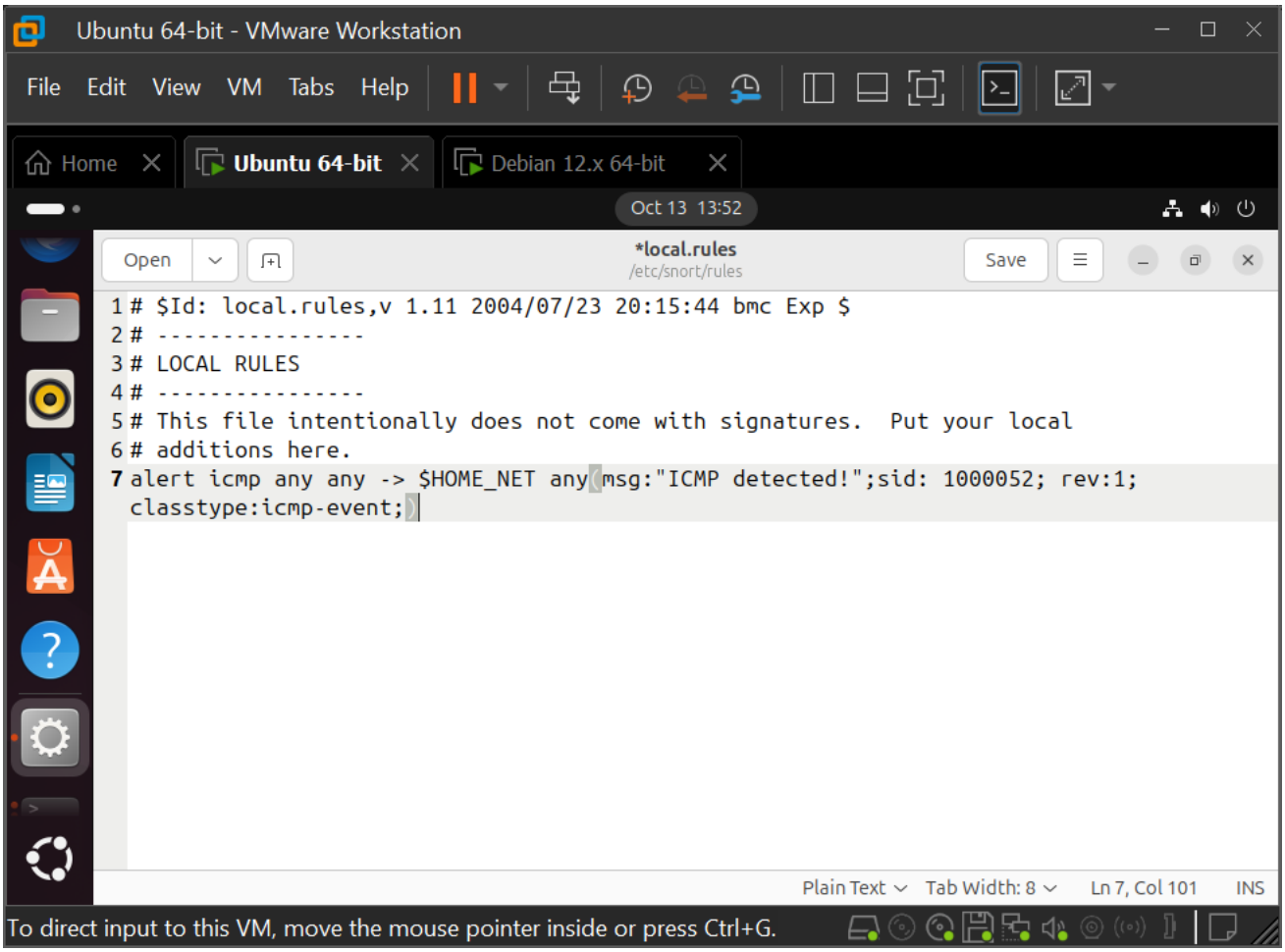
For the variable RULE_PATH (of type var), assign the value /etc/snort/rules. The include statement will expand that path (/etc/snort/rules) before /local.rules to produce an absolute reference for the rules file. Using a variable for the rule path allows you to use the variable for multiple references and also allows you to change just the RULE_PATH variable, if necessary, as opposed to each include statement. Close **gedit** after looking through **classification.config**.



b) Enter the following command: **sudo gedit /etc/snort/rules/local.rules**

The following is from the comments section in the file:

This file intentionally does not come with signatures. Put your local additions here. Add the following rule on one single line. Do not press ENTER to break up the line. The text will wrap, if necessary, to the next line. **alert icmp any any -> $HOME_NET any (msg:"ICMP detected!";sid: 1000052; rev:1; classtype:icmp-event;)**

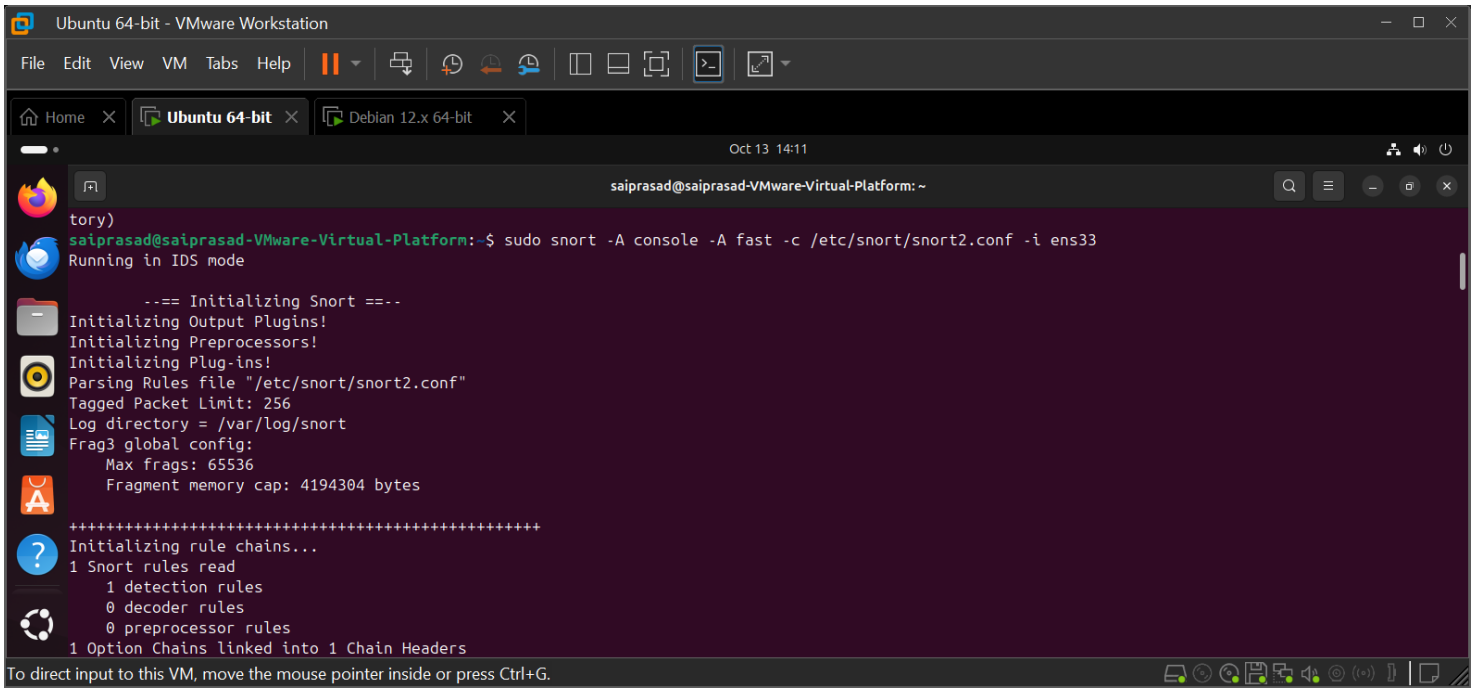


c) Now start Snort in IDS mode and instruct it to display alerts to the console: **sudo snort -A console -A fast -c /etc/snort/snort2.conf -i ens33** Here, **-c** specifies the configuration file and **-i** specifies the interface. The following is from the Snort man page:

-A alert-mode

Alert using the specified alert-mode. Valid alert modes include fast, full, none, and unsock. Fast writes alerts to the default “alert” file in a single-line, syslog style alert message. Full writes the alert to the “alert” file with the full decoded header as well as the alert message. None turns off alerting. Unsock is an experimental mode that sends the alert information out over a UNIX socket to another process that attaches to that socket. -c config-file Use the rules located in file config-file.

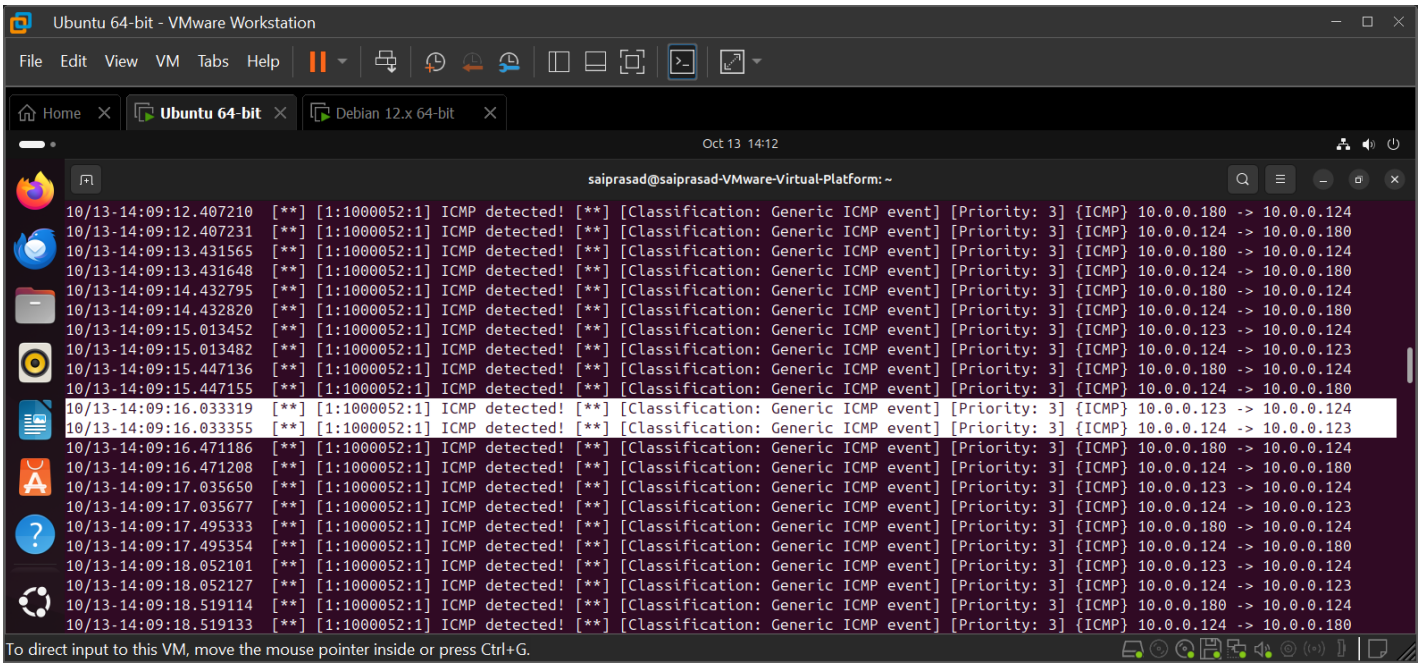
More on alert modes can be found here at http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node6.html

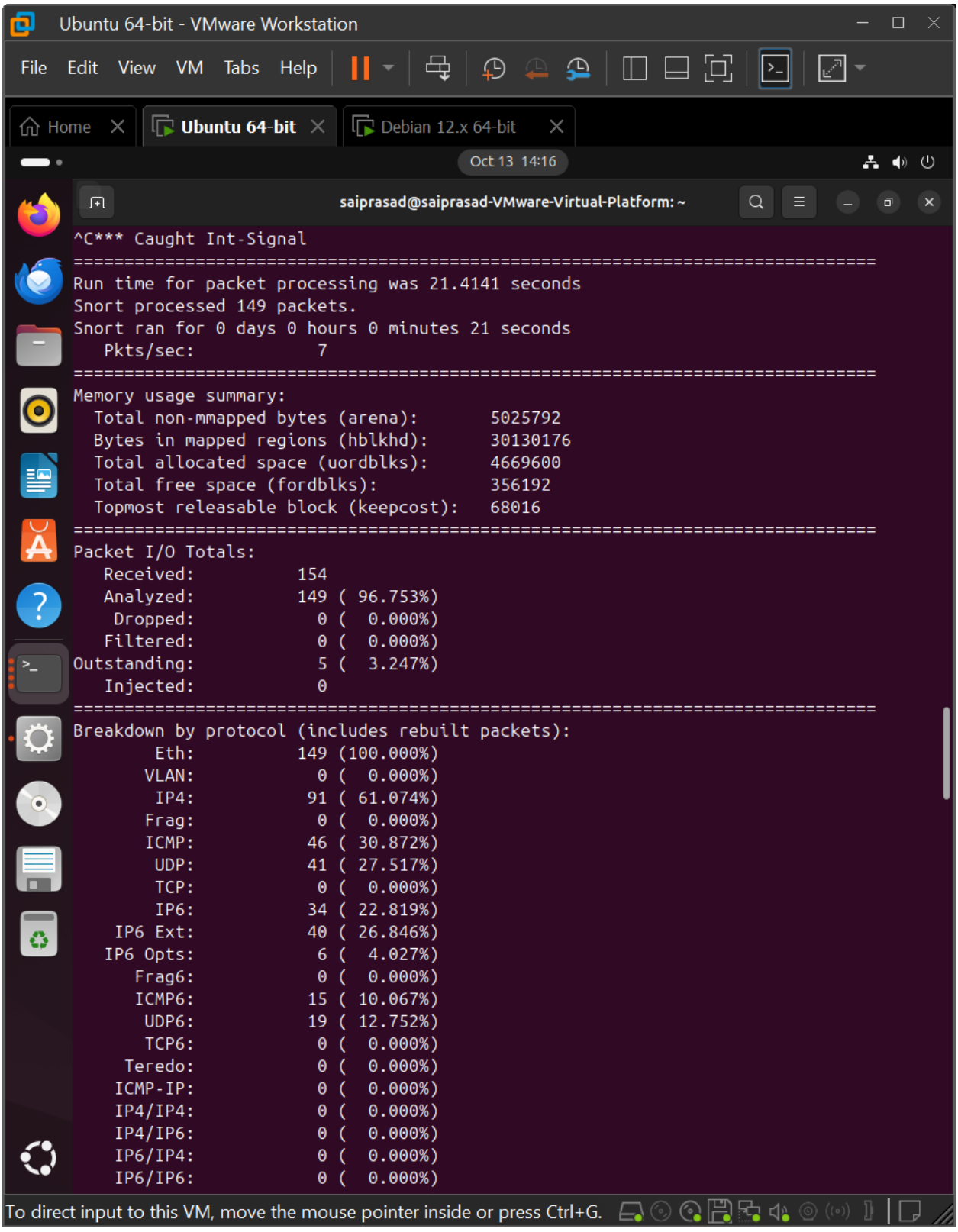


d) From the Windows 10 VM or host machine, ping the Ubuntu VM. You will notice eight ICMP alerts (four Echo Requests and four Echo Replies). **Take the screenshot.**

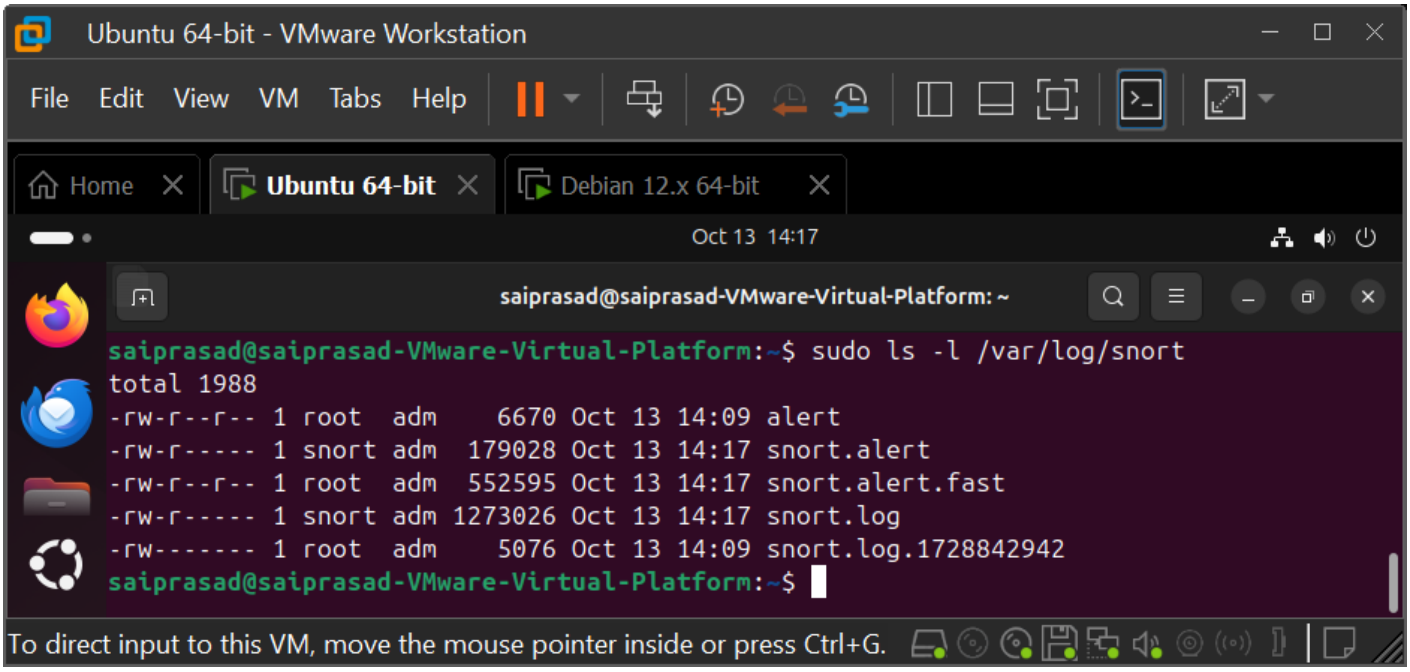
Press **CTRL-C** to break out. Notice all the information at the end of the output, including the following sections:

- Run time for packet processing
- Number of packets processed
- Memory usage summary
- Packet I/O Total
- Breakdown by protocol (includes rebuilt packets)
- Action stats
- Frag3 statistics
- Stream statistics
- SMTP Preprocessor Statistics
- Dcerpc2 Preprocessor Statistics
- SIP Preprocessor Statistics




Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home
Ubuntu 64-bit
Debian 12.x 64-bit
Oct 13 14:16
saiprasad@saiprasad-VMware-Virtual-Platform: ~
^C*** Caught Int-Signal
Run time for packet processing was 21.4141 seconds
Snort processed 149 packets.
Snort ran for 0 days 0 hours 0 minutes 21 seconds
Pkts/sec: 7
Memory usage summary:
Total non-mmapped bytes (arena): 5025792
Bytes in mapped regions (hblkhd): 30130176
Total allocated space (uordblks): 4669600
Total free space (fordblks): 356192
Topmost releasable block (keepcost): 68016
Packet I/O Totals:
Received: 154
Analyzed: 149 (96.753%)
Dropped: 0 (0.000%)
Filtered: 0 (0.000%)
Outstanding: 5 (3.247%)
Injected: 0
Breakdown by protocol (includes rebuilt packets):
Eth: 149 (100.000%)
VLAN: 0 (0.000%)
IP4: 91 (61.074%)
Frag: 0 (0.000%)
ICMP: 46 (30.872%)
UDP: 41 (27.517%)
TCP: 0 (0.000%)
IP6: 34 (22.819%)
IP6 Ext: 40 (26.846%)
IP6 Opts: 6 (4.027%)
Frag6: 0 (0.000%)
ICMP6: 15 (10.067%)
UDP6: 19 (12.752%)
TCP6: 0 (0.000%)
Teredo: 0 (0.000%)
ICMP-IP: 0 (0.000%)
IP4/IP4: 0 (0.000%)
IP4/IP6: 0 (0.000%)
IP6/IP4: 0 (0.000%)
IP6/IP6: 0 (0.000%)
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

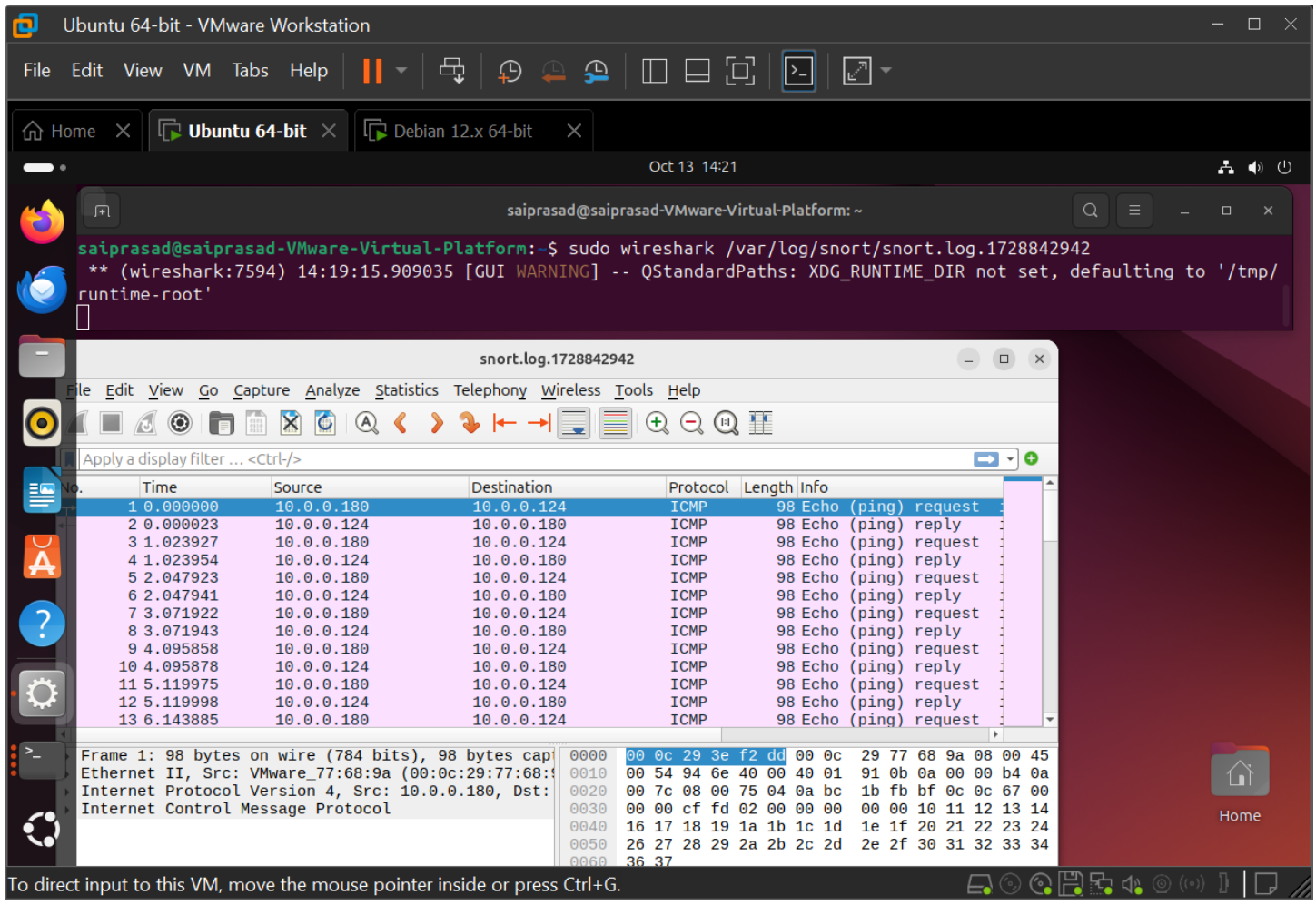
e) Enter the following command to see the name of the log file: **sudo ls -l /var/log/snort** **(Take the screenshot.)**



f) Open the log file in Wireshark:

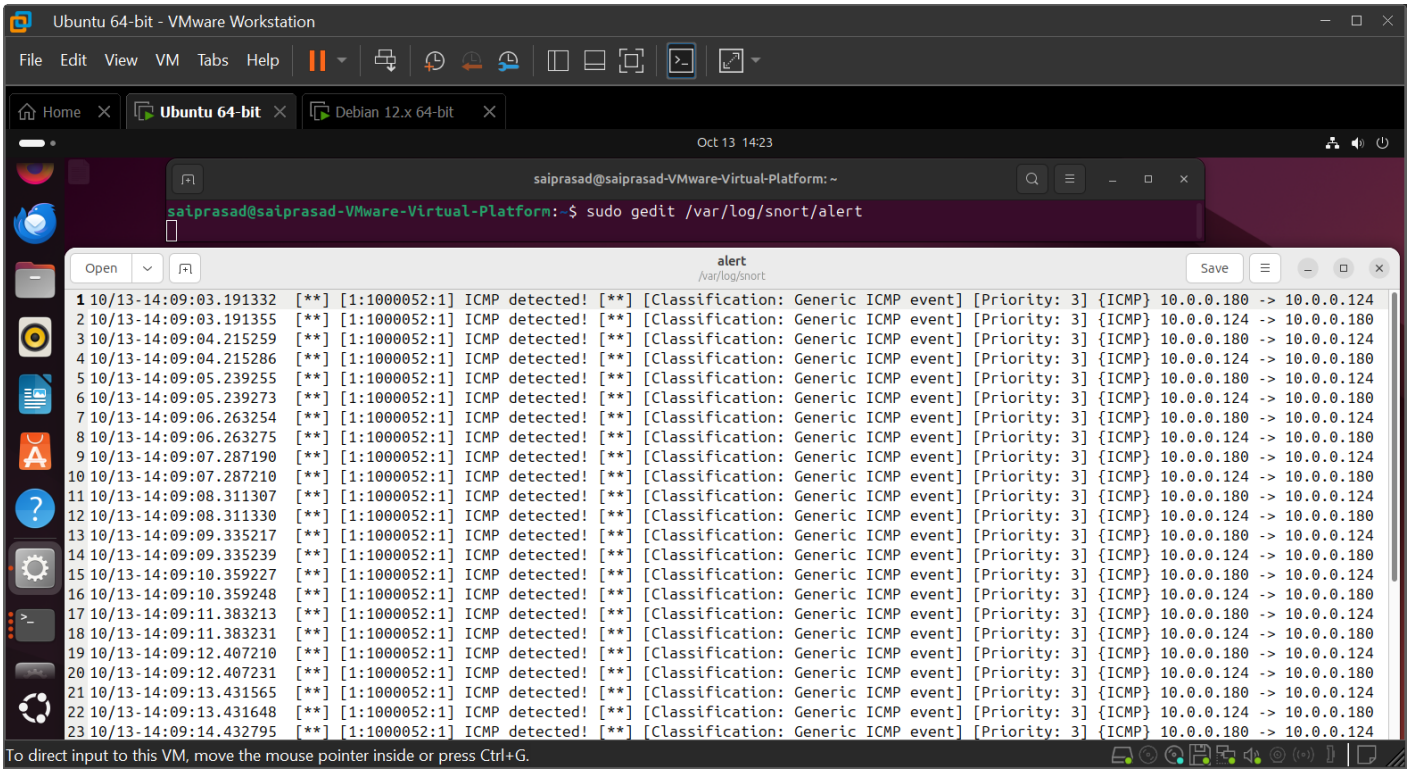
sudo wireshark /var/log/snort/snort.log.[numbers seen here]

When you get to the end of the filename in the command after snort.log., press **TAB**, and the rest of the filename, consisting of numbers, will autocomplete.



g) Open the alert file: **sudo gedit /var/log/snort/alert**

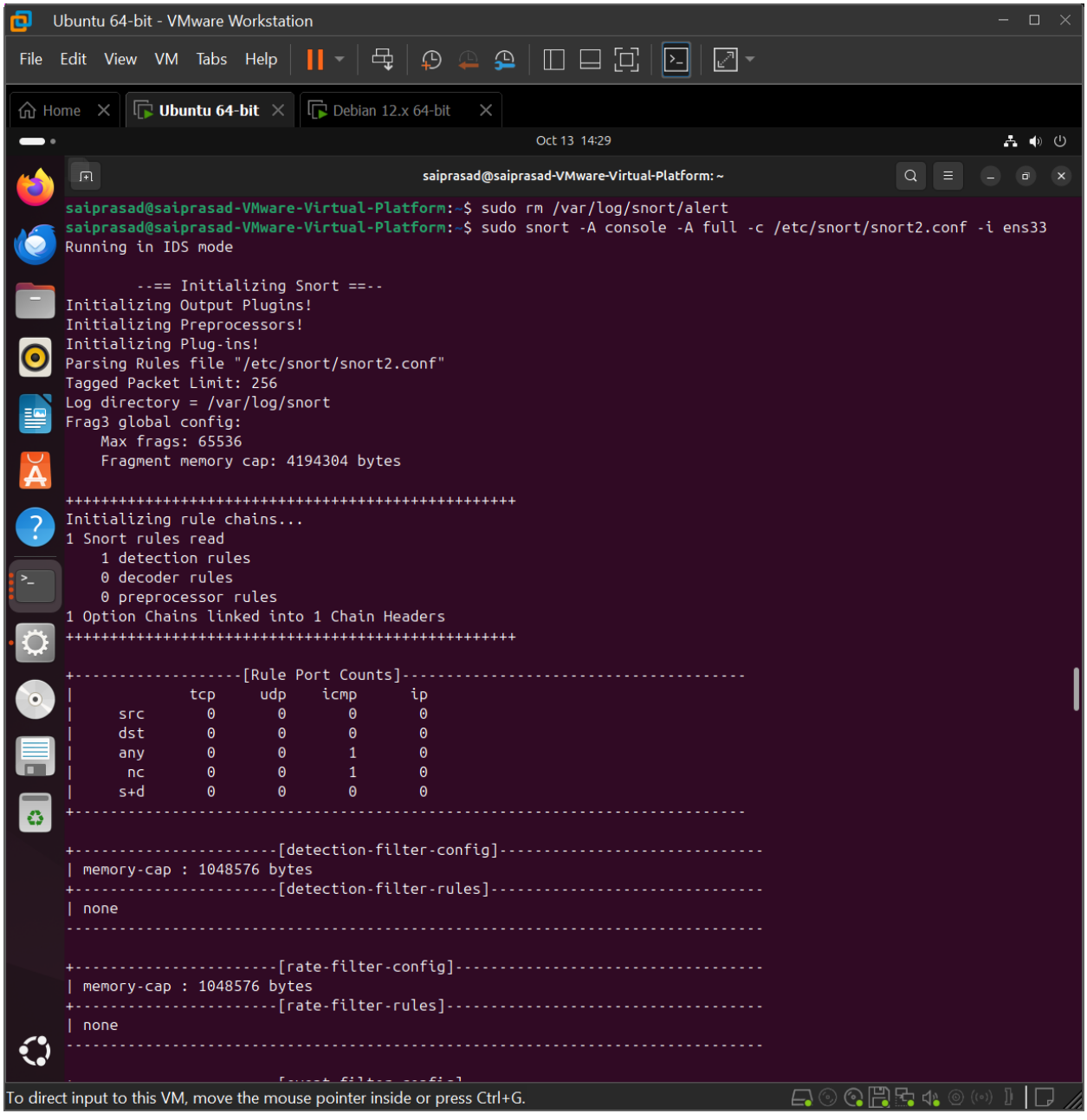
Take the screenshot. The file contains contents similar to what was outputted in the Terminal earlier.

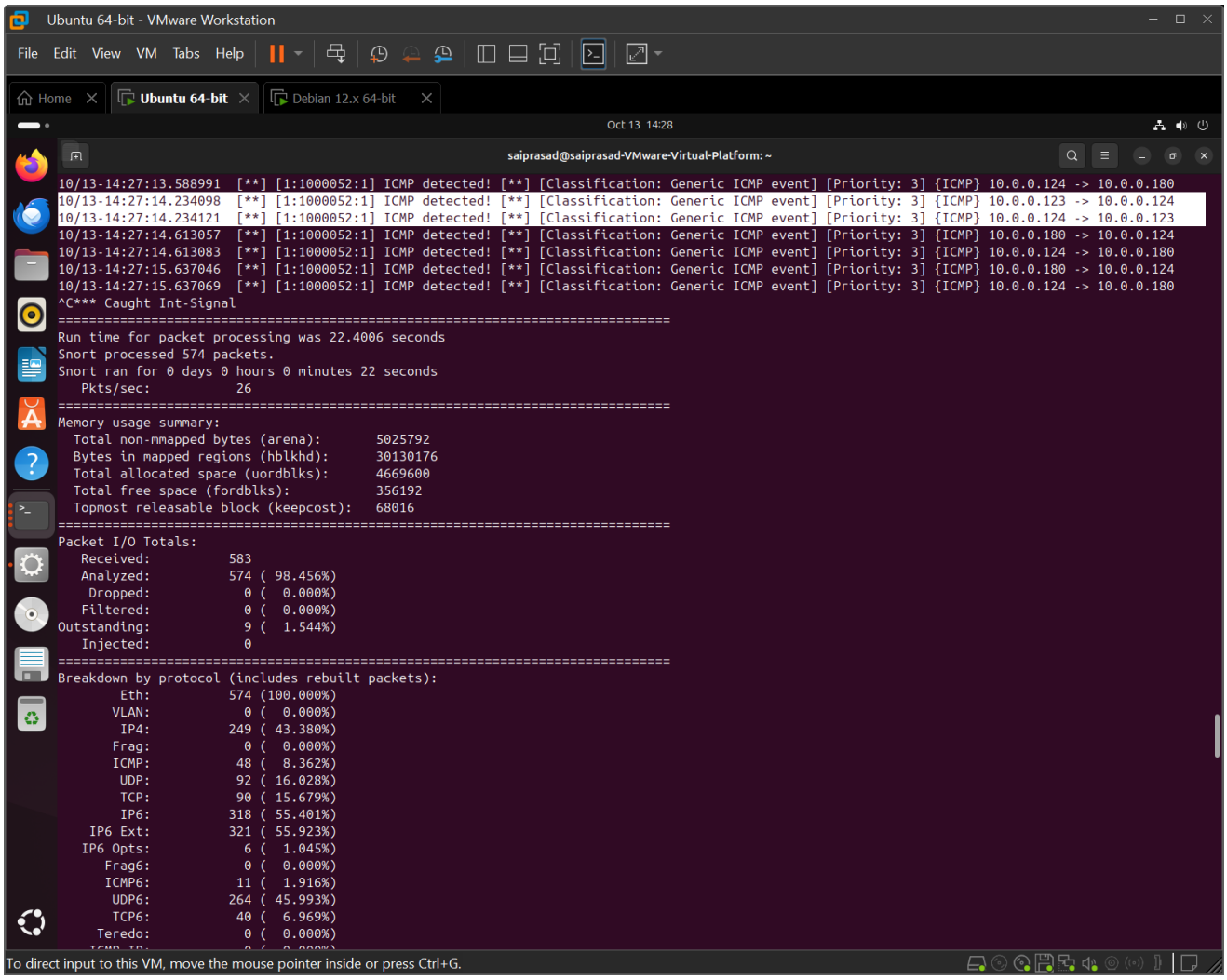


h) Remove the alert file and restart Snort, changing the **-A** value from **fast** to **full**: **sudo rm /var/log/snort/alert**

sudo snort -A console -A full -c /etc/snort/snort2.conf -i ens33 From the Windows 10 VM or host machine, ping the Ubuntu VM.

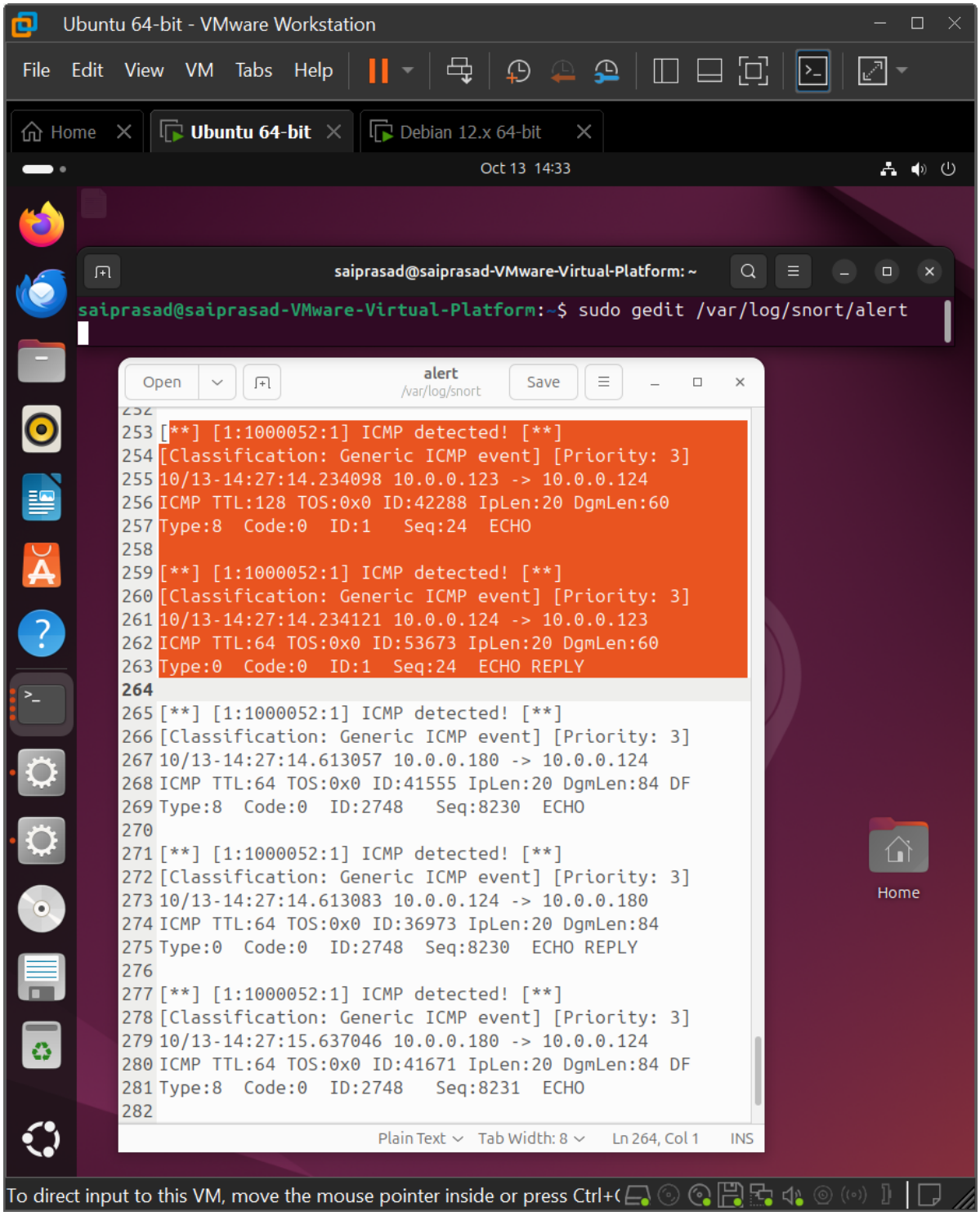
You will notice eight ICMP alerts (four Echo Requests and four Echo Replies). Press **CTRL-C** to break out. The fast option has been replaced by full.




Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home
Ubuntu 64-bit
Debian 12.x 64-bit
Oct 13 14:28
saiprasad@saiprasad-VMware-Virtual-Platform: ~
10/13-14:27:13.588991 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.124 -> 10.0.0.180
10/13-14:27:14.234098 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.123 -> 10.0.0.124
10/13-14:27:14.234121 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.124 -> 10.0.0.123
10/13-14:27:14.613057 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.180 -> 10.0.0.124
10/13-14:27:14.613083 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.124 -> 10.0.0.180
10/13-14:27:15.637046 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.180 -> 10.0.0.124
10/13-14:27:15.637069 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.124 -> 10.0.0.180
^C*** Caught Int-Signal
Run time for packet processing was 22.4006 seconds
Snort processed 574 packets.
Snort ran for 0 days 0 hours 0 minutes 22 seconds
Pkts/sec: 26
Memory usage summary:
Total non-mmapped bytes (arena): 5025792
Bytes in mapped regions (hblkhd): 30130176
Total allocated space (uordblks): 4669600
Total free space (fordblks): 356192
Topmost releasable block (keepcost): 68016
Packet I/O Totals:
Received: 583
Analyzed: 574 (98.456%)
Dropped: 0 (0.000%)
Filtered: 0 (0.000%)
Outstanding: 9 (1.544%)
Injected: 0
Breakdown by protocol (includes rebuilt packets):
Eth: 574 (100.000%)
VLAN: 0 (0.000%)
IP4: 249 (43.380%)
Frag: 0 (0.000%)
ICMP: 48 (8.362%)
UDP: 92 (16.028%)
TCP: 90 (15.679%)
IP6: 318 (55.401%)
IP6 Ext: 321 (55.923%)
IP6 Opts: 6 (1.045%)
Frag6: 0 (0.000%)
ICMP6: 11 (1.916%)
UDP6: 264 (45.993%)
TCP6: 40 (6.969%)
Teredo: 0 (0.000%)
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

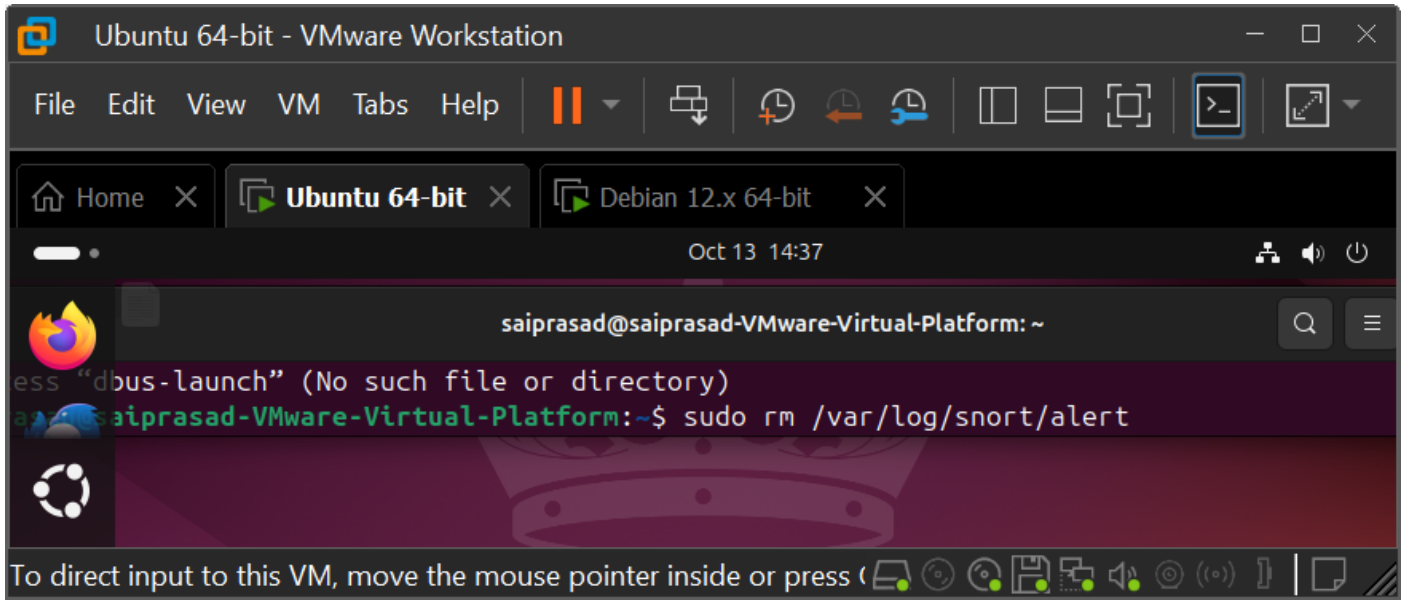
i) Enter the following command: **sudo gedit /var/log/snort/alert**

Take the screenshot. Notice the full IP headers (although, excluding some fields) for each alert in the alert file.



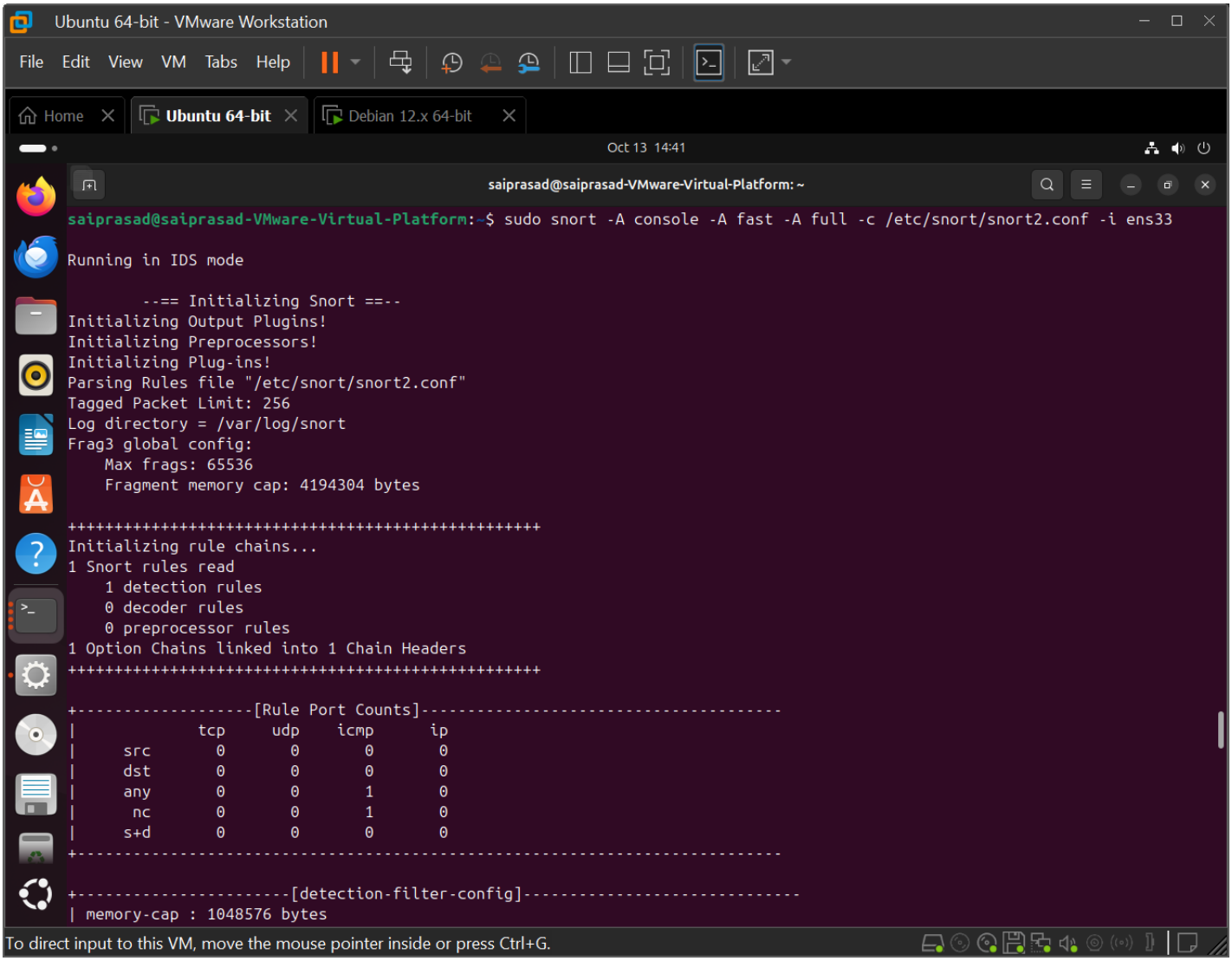
j) Remove the alert file and restart Snort, using both fast and full with two -A options:

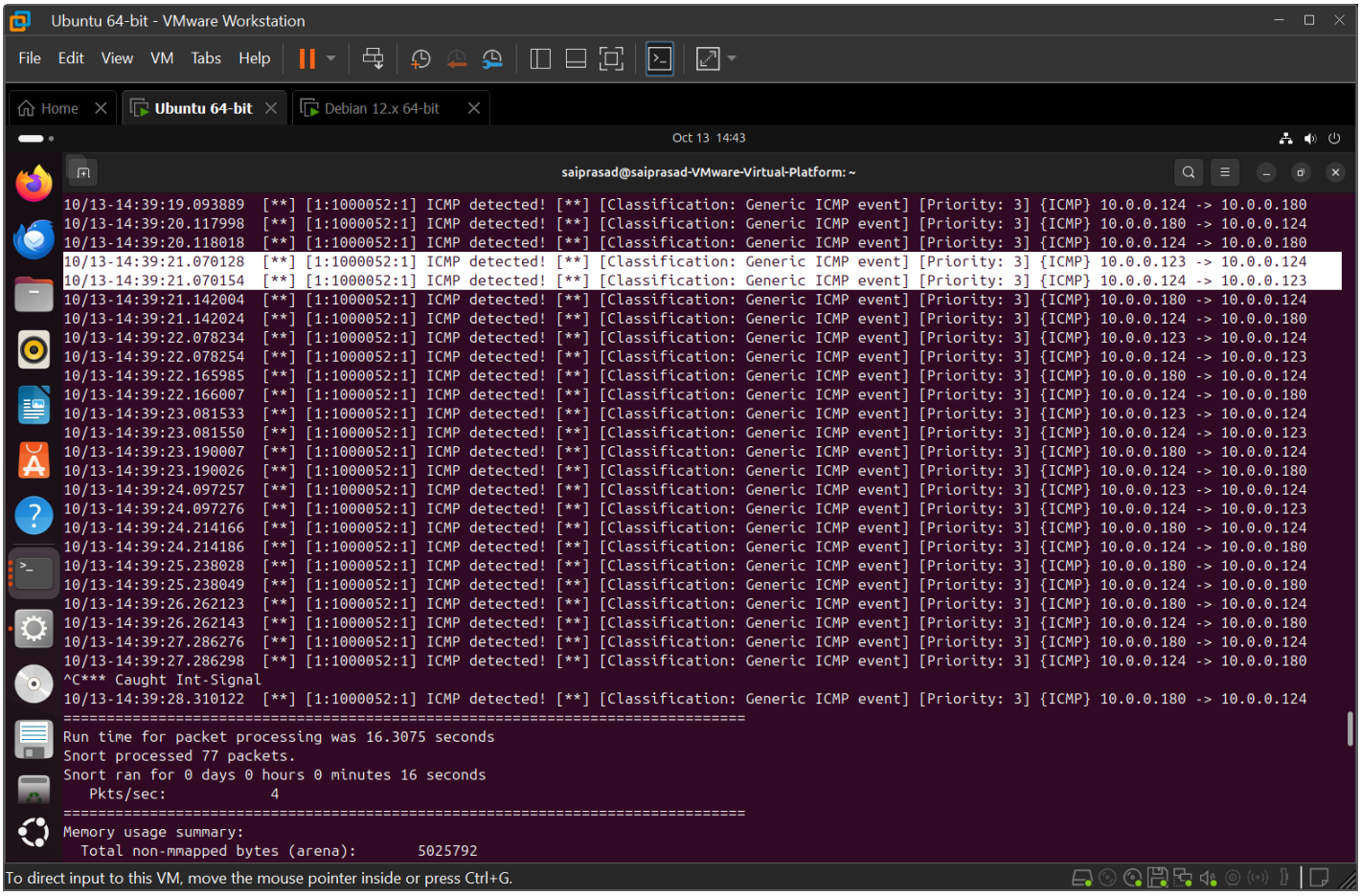
sudo rm /var/log/snort/alert



sudo snort -A console -A fast -A full -c /etc/snort/snort2.conf -i ens33

From the Windows 10 VM or host machine, ping the Ubuntu VM. You will notice eight ICMP alerts (four Echo Requests and four Echo Replies). Press **CTRL-C** to break out. Both the **full** and **fast** options are being used now.




Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home
Ubuntu 64-bit
Debian 12.x 64-bit
Oct 13 14:43
saiprasad@saiprasad-VMware-Virtual-Platform: ~
10/13-14:39:19.093889 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.124 -> 10.0.0.180
10/13-14:39:20.117998 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.180 -> 10.0.0.124
10/13-14:39:20.118018 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.124 -> 10.0.0.180
10/13-14:39:21.070128 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.123 -> 10.0.0.124
10/13-14:39:21.070154 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.124 -> 10.0.0.123
10/13-14:39:21.142004 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.180 -> 10.0.0.124
10/13-14:39:21.142024 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.124 -> 10.0.0.180
10/13-14:39:22.078234 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.123 -> 10.0.0.124
10/13-14:39:22.078254 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.124 -> 10.0.0.123
10/13-14:39:22.165985 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.180 -> 10.0.0.124
10/13-14:39:22.166007 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.124 -> 10.0.0.180
10/13-14:39:23.081533 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.123 -> 10.0.0.124
10/13-14:39:23.081550 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.124 -> 10.0.0.123
10/13-14:39:23.190007 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.180 -> 10.0.0.124
10/13-14:39:23.190026 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.124 -> 10.0.0.180
10/13-14:39:24.097257 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.123 -> 10.0.0.124
10/13-14:39:24.097276 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.124 -> 10.0.0.123
10/13-14:39:24.214166 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.180 -> 10.0.0.124
10/13-14:39:24.214186 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.124 -> 10.0.0.180
10/13-14:39:25.238028 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.180 -> 10.0.0.124
10/13-14:39:25.238049 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.124 -> 10.0.0.180
10/13-14:39:26.262123 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.180 -> 10.0.0.124
10/13-14:39:26.262143 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.124 -> 10.0.0.180
10/13-14:39:27.286276 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.180 -> 10.0.0.124
10/13-14:39:27.286298 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.124 -> 10.0.0.180
^C*** Caught Int-Signal
10/13-14:39:28.310122 [**] [1:1000052:1] ICMP detected! [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 10.0.0.180 -> 10.0.0.124
Run time for packet processing was 16.3075 seconds
Snort processed 77 packets.
Snort ran for 0 days 0 hours 0 minutes 16 seconds
Pkts/sec: 4
Memory usage summary:
Total non-mmapped bytes (arena): 5025792
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

k) Enter the following command: **sudo gedit /var/log/snort/alert**
With both the fast and full options specified at the same time, you will now see output from both (one after the other) in each individual alert. **Take the screenshot.**