Project: Network Intrusion Detection System (NIDS) Using Snort

CYBER_SECURITY_TWO_MONTHS_BATCH-6

Month-1

Group No: G-1

INTERN NAME: VISHAL SANJAY JADHAV.

{ WEEK 1 REPORT }

TASKS :

• Introduction to NIDS and Snort

• Install Linux (Ubuntu/Kali)

• Install and verify Snort

• Basic Linux command-line navigation

❖ Introduction to NIDS and Snort

1. **NIDS :** A **Network Intrusion Detection System (NIDS)** is a **security tool** that monitors network traffic for suspicious or harmful activity.

   Work of NIDS :

   - NIDS watches all the data moving across a network.
   - If it sees anything unusual (like a hacker attack), it alerts the admin.
   - It helps detect attacks like viruses, port scans, or unauthorized access.

2. **Snort : Snort** is a **free** and **open-source Network Intrusion Detection System (NIDS)** created by Cisco. It helps you detect and prevent suspicious activities on your network.

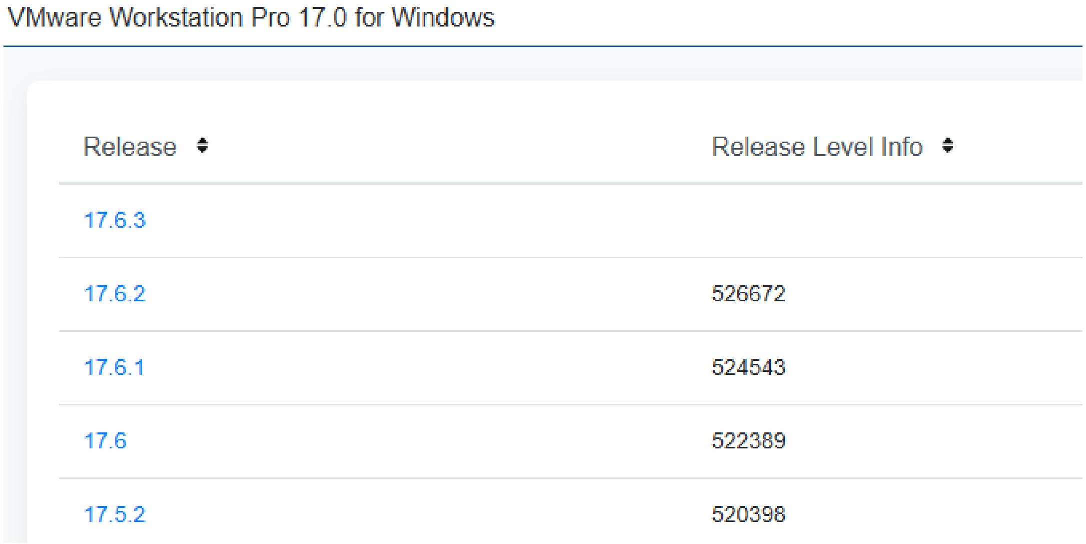
   Work of Snort :

   - **Monitors** network traffic in real time.
   - **Detects attacks** (like malware, port scans, buffer overflows).
   - **Alerts you** or **blocks** the traffic if it sees something dangerous.

- **Install Linux (Kali) in windows**

1. Install VMWARE :

- If not already registered then First register your account here - https://profile.broadcom.com/web/registration
- After registration/login head to- https://support.broadcom.com/group/ecx/productdownloads?subfamily=VMware%20Workstation%20Pro&freeDownloads=true
- Click on the latest one and download it.

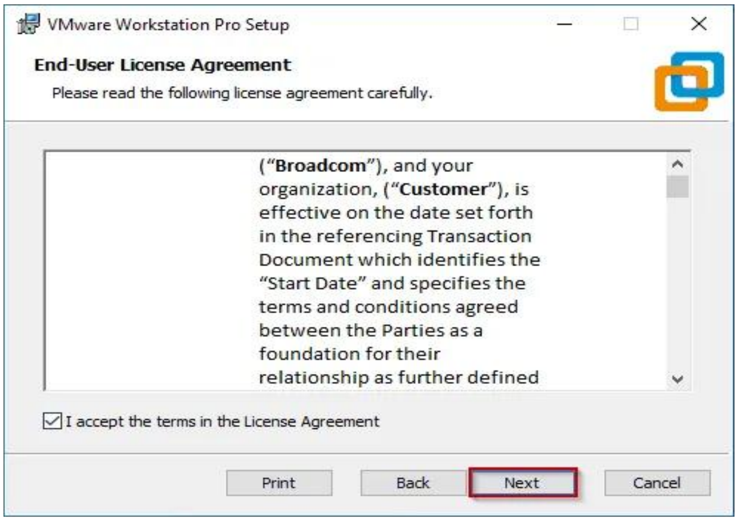


VMware Workstation Pro 17.0 for Windows

Release ⬍	Release Level Info ⬍
17.6.3	
17.6.2	526672
17.6.1	524543
17.6	522389
17.5.2	520398

- Also download latest kali linux for VMware on it’s official website: https://www.kali.org/get-kali/#kali-virtual-machines
- After both things downloaded install VMware-

1



VMware Workstation Pro Setup

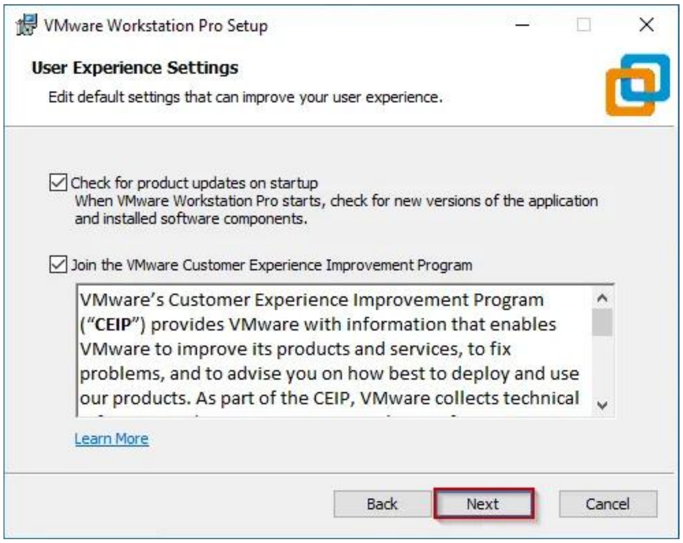
End-User License Agreement

Please read the following license agreement carefully.

(“**Broadcom**”), and your organization, (“**Customer**”), is effective on the date set forth in the referencing Transaction Document which identifies the “Start Date” and specifies the terms and conditions agreed between the Parties as a foundation for their relationship as further defined

☑ I accept the terms in the License Agreement

2



VMware Workstation Pro Setup

User Experience Settings

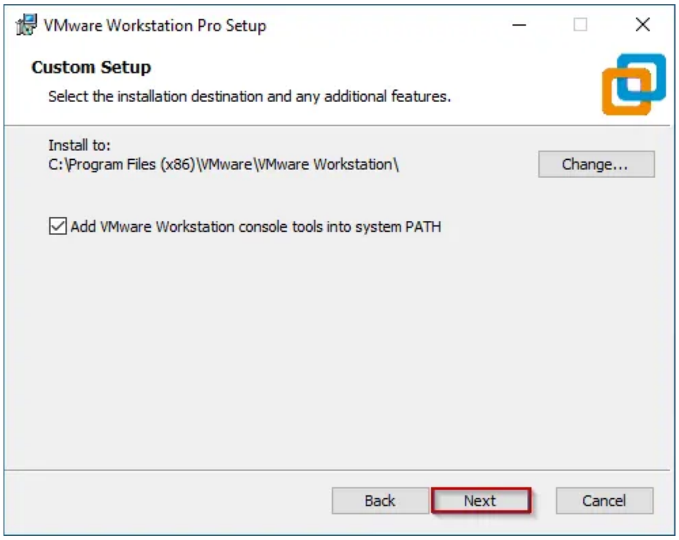
Edit default settings that can improve your user experience.

☑ Check for product updates on startup
When VMware Workstation Pro starts, check for new versions of the application and installed software components.

☑ Join the VMware Customer Experience Improvement Program

VMware’s Customer Experience Improvement Program (“**CEIP**”) provides VMware with information that enables VMware to improve its products and services, to fix problems, and to advise you on how best to deploy and use our products. As part of the CEIP, VMware collects technical

3



VMware Workstation Pro Setup

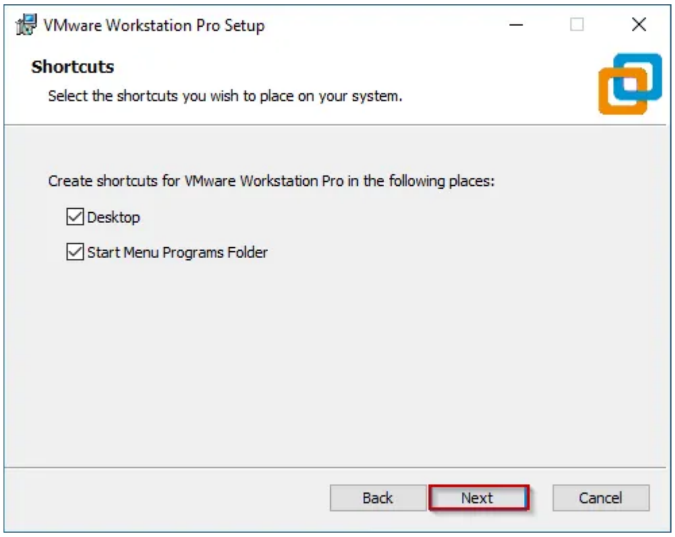
Custom Setup

Select the installation destination and any additional features.

Install to:
C:\Program Files (x86)\VMware\VMware Workstation\

☑ Add VMware Workstation console tools into system PATH

4



VMware Workstation Pro Setup

Shortcuts

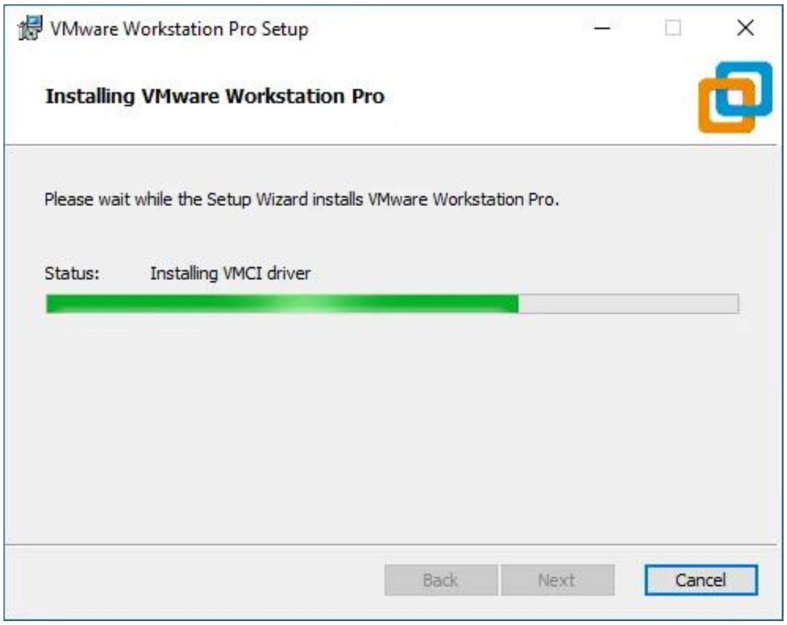
Select the shortcuts you wish to place on your system.

Create shortcuts for VMware Workstation Pro in the following places:

☑ Desktop

☑ Start Menu Programs Folder

5



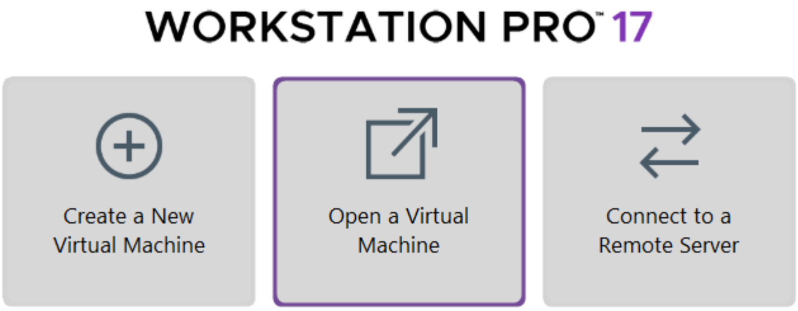
VMware Workstation Pro Setup

Installing VMware Workstation Pro

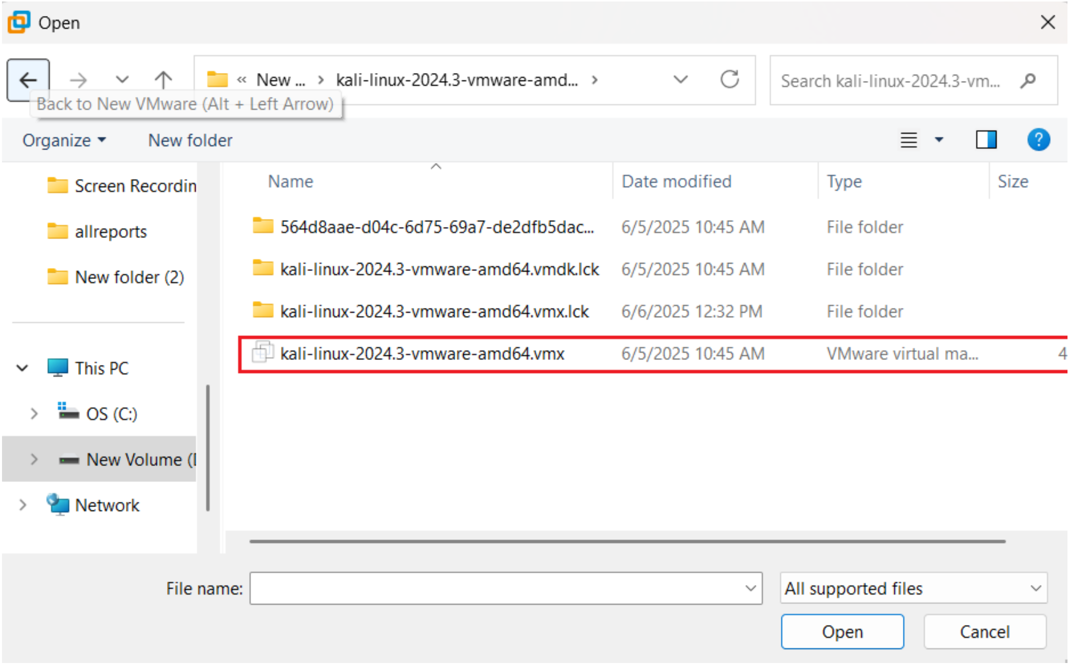
Please wait while the Setup Wizard installs VMware Workstation Pro.

Status: Installing VMCI driver

- ❖ **Now Open VMware and “Click on Open a Virtual Machine”-**



- ❖ **Select the vmx configuration file of kali linux**


Open
« New ... › kali-linux-2024.3-vmware-amd... ›
Search kali-linux-2024.3-vm...
Back to New VMware (Alt + Left Arrow)
Organize ▾
New folder
Screen Recordin
allreports
New folder (2)
This PC
OS (C:)
New Volume (I
Network
Open
Cancel

Name	Date modified	Type	Size
564d8aae-d04c-6d75-69a7-de2dfb5dac...	6/5/2025 10:45 AM	File folder	
kali-linux-2024.3-vmware-amd64.vmdk.lck	6/5/2025 10:45 AM	File folder	
kali-linux-2024.3-vmware-amd64.vmx.lck	6/6/2025 12:32 PM	File folder	
kali-linux-2024.3-vmware-amd64.vmx	6/5/2025 10:45 AM	VMware virtual ma...	4

File name:

All supported files

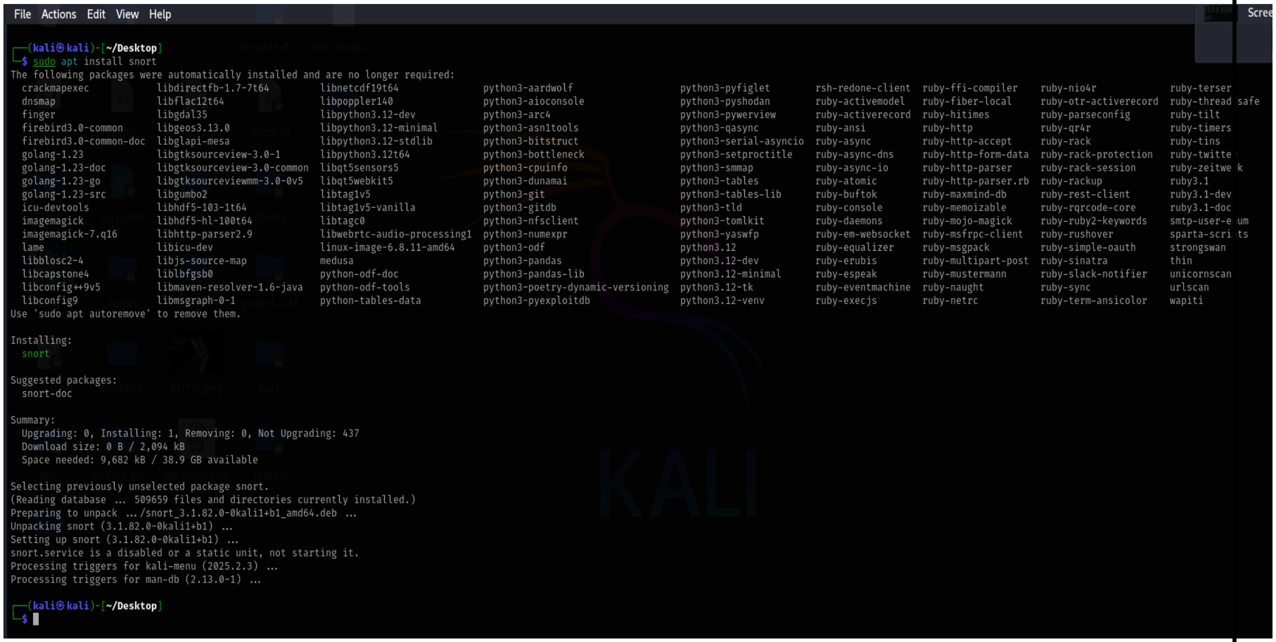
After opening the kali linux kindly update and upgrade the kali linux.

1. **Command For Update : apt-get update**
2. **Command For Upgrade : apt-get upgrade**

❖ Install and verify snort

1. Install Snort

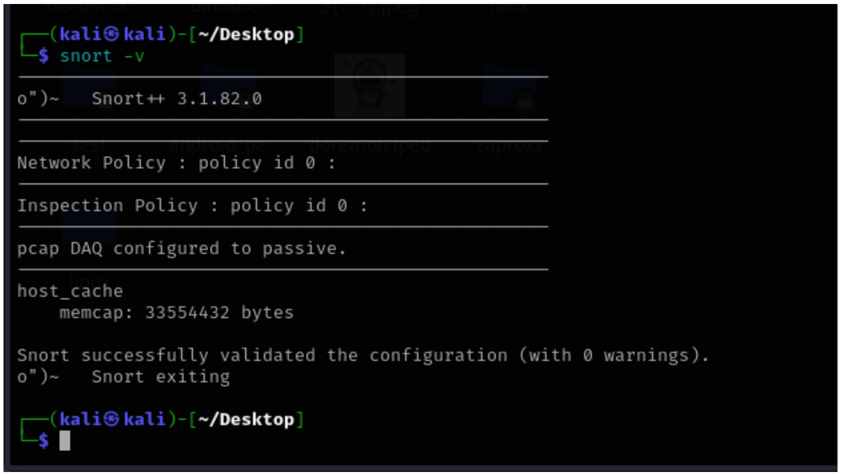
Use this command : <u>sudo apt install snort</u>



```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo apt install snort
The following packages were automatically installed and are no longer required:
  crackmapexec           libdirectfb-1.7-7t64            libnetcdf19t64                  python3-aardwolf                    python3-pyfiglet          rsh-redone-client  ruby-ffi-compiler     ruby-nio4r                ruby-terser
  dnsmap                 libflac12t64                    libpoppler140                   python3-aioconsole                  python3-pyshodan          ruby-activemodel   ruby-fiber-local      ruby-otr-activerecord     ruby-thread-safe
  finger                 libgdal35                       libpython3.12-dev               python3-arc4                        python3-pywerview         ruby-activerecord  ruby-hitimes          ruby-parseconfig          ruby-tilt
  firebird3.0-common     libgeos3.13.0                   libpython3.12-minimal           python3-asn1tools                   python3-qasync            ruby-ansi          ruby-http             ruby-qr4r                 ruby-timers
  firebird3.0-common-doc libglapi-mesa                   libpython3.12-stdlib            python3-bitstruct                   python3-serial-asyncio    ruby-async         ruby-http-accept      ruby-rack                 ruby-tins
  golang-1.23            libgtksourceview-3.0-1          libpython3.12t64                python3-bottleneck                  python3-setproctitle      ruby-async-dns     ruby-http-form-data   ruby-rack-protection      ruby-twitter
  golang-1.23-doc        libgtksourceview-3.0-common     libqt5sensors5                  python3-cpuinfo                     python3-smmap             ruby-async-io      ruby-http-parser      ruby-rack-session         ruby-zeitwerk
  golang-1.23-go         libgtksourceviewmm-3.0-0v5      libqt5webkit5                   python3-dunamai                     python3-tables            ruby-atomic        ruby-http-parser.rb   ruby-rackup               ruby3.1
  golang-1.23-src        libgumbo2                       libtag1v5                       python3-git                         python3-tables-lib        ruby-buftok        ruby-maxmind-db       ruby-rest-client          ruby3.1-dev
  icu-devtools           libhdf5-103-1t64                libtag1v5-vanilla               python3-gitdb                       python3-tld               ruby-console       ruby-memoizable       ruby-rqrcode-core         ruby3.1-doc
  imagemagick            libhdf5-hl-100t64               libtagc0                        python3-nfsclient                   python3-tomlkit           ruby-daemons       ruby-mojo-magick      ruby-ruby2-keywords       smtp-user-enum
  imagemagick-7.q16      libhttp-parser2.9               libwebrtc-audio-processing1     python3-numexpr                     python3-yaswfp            ruby-em-websocket  ruby-msfrpc-client    ruby-rushover             sparta-scripts
  lame                   libicu-dev                      linux-image-6.8.11-amd64        python3-odf                         python3.12                ruby-equalizer     ruby-msgpack          ruby-simple-oauth         strongswan
  libblosc2-4            libjs-source-map                medusa                          python3-pandas                      python3.12-dev            ruby-erubis        ruby-multipart-post   ruby-sinatra              thin
  libcapstone4           liblbfgsb0                      python-odf-doc                  python3-pandas-lib                  python3.12-minimal        ruby-espeak        ruby-mustermann       ruby-slack-notifier       unicornscan
  libconfig++9v5         libmaven-resolver-1.6-java      python-odf-tools                python3-poetry-dynamic-versioning   python3.12-tk             ruby-eventmachine  ruby-naught           ruby-sync                 urlscan
  libconfig9             libmsgraph-0-1                  python-tables-data              python3-pyexploitdb                 python3.12-venv           ruby-execjs        ruby-netrc            ruby-term-ansicolor       wapiti
Use 'sudo apt autoremove' to remove them.

Installing:
  snort

Suggested packages:
  snort-doc

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 437
  Download size: 0 B / 2,094 kB
  Space needed: 9,682 kB / 38.9 GB available

Selecting previously unselected package snort.
(Reading database ... 509659 files and directories currently installed.)
Preparing to unpack .../snort_3.1.82.0-0kali1+b1_amd64.deb ...
Unpacking snort (3.1.82.0-0kali1+b1) ...
Setting up snort (3.1.82.0-0kali1+b1) ...
snort.service is a disabled or a static unit, not starting it.
Processing triggers for kali-menu (2025.2.3) ...
Processing triggers for man-db (2.13.0-1) ...

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

2. Verify snort

Use this command : snort -v



```
┌──(kali㉿kali)-[~/Desktop]
└─$ snort -v
--------------------------------------------------
o")~   Snort++ 3.1.82.0
--------------------------------------------------
--------------------------------------------------
Network Policy : policy id 0 :
--------------------------------------------------
Inspection Policy : policy id 0 :
--------------------------------------------------
pcap DAQ configured to passive.
--------------------------------------------------
host_cache
    memcap: 33554432 bytes

Snort successfully validated the configuration (with 0 warnings).
o")~   Snort exiting

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

Basic Linux command-line navigation

1. ctrl+alt+t	Open new tab
2. sudo su	normal user to root user
3. mkdir	make new directory
4. cd	change directory
5. pwd	show current working directory
6. ls	list show
7. ls -a	used to view hidden files or directories
8. mv	move file or rename the file
9. cp	copy file
10. rm	remove file
11. rm -rf	delete folder or directory
12. rmdir	remove directory
13. cat	read the text file
14. touch	create empty text file
15. nano	create a text file with some data
16. whoami	show current user
17. cal	show calendar
18. history	show history
19. apt-get update	used to update the kali
20. apt-get upgrade	used to upgrade the kali
21. clear	clear the screen
22. ifconfig	shows all active interface
23. poweroff	used to shutdown kali linux
24. echo	used to print something
25. exit	exit the terminal session

{WEEK 2 REPORT}

Tasks:

• Identify active network interface

• Configure Snort with monitored IP range

• Run Snort in detection mode

• Monitor live traffic and alerts

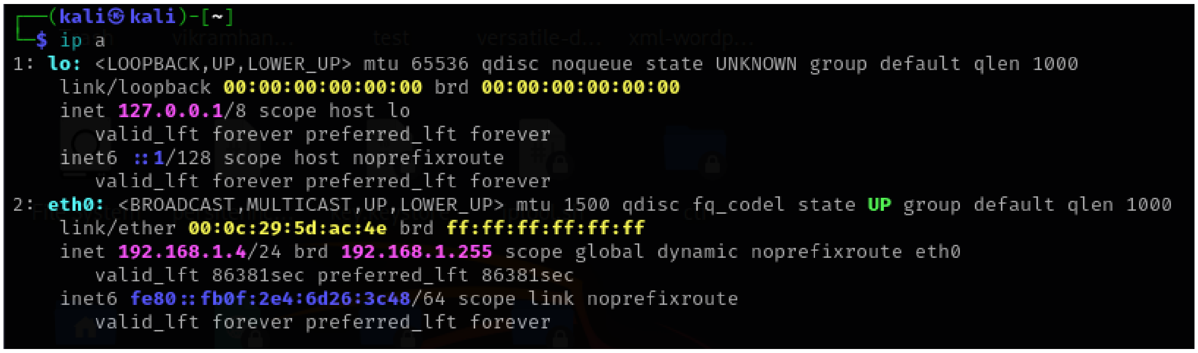
❖ Identify active network interface

For identify active network interface we can use the “ip a” command in kali terminal

Use of ip a command:

1. all network interfaces. (e.g., eth0,lo, wlan0)
2. ip addresses assigned to each interface (both IPV4 and IPV6)
3. Interface status whether it's UP or DOWN

Command: - ip a



```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:5d:ac:4e brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.4/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0
       valid_lft 86381sec preferred_lft 86381sec
    inet6 fe80::fb0f:2e4:6d26:3c48/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

❖ Configure Snort with monitored IP range

To configure Snort with the monitored IP range, we first identify our internal network using the ip a command. In this case, the internal network range is **192.168.1.0/24.**

Next, we edit the Snort 3 configuration file by running:

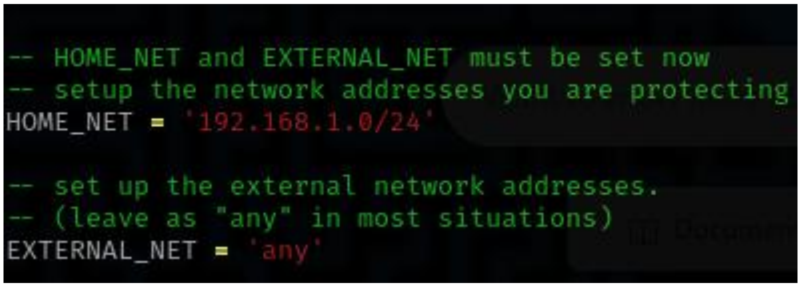
- **sudo nano /usr/local/snort/etc/snort.lua**

Inside the configuration, we define the monitored network (HOME_NET) and external network (EXTERNAL_NET) as follows:

HOME_NET = '192.168.1.0/24'

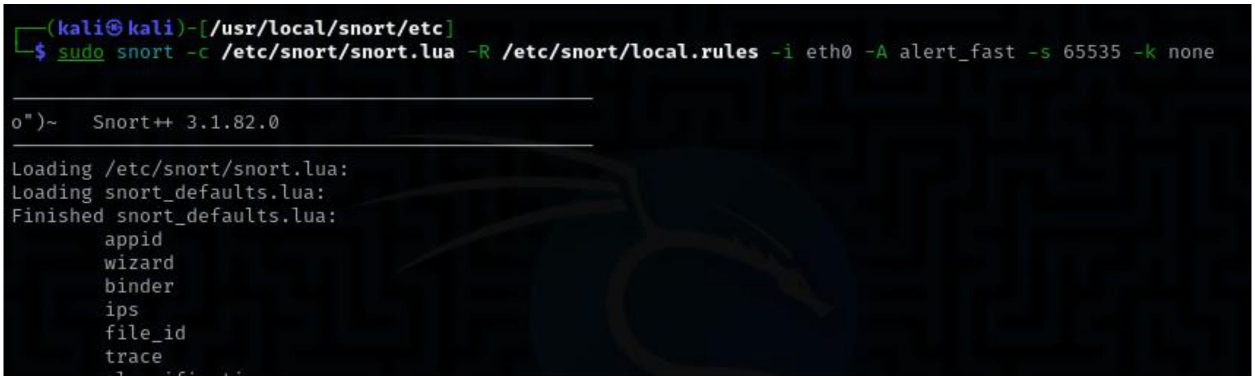
EXTERNAL_NET = 'any'

HOME_NET represents the internal network you wish to monitor, while EXTERNAL_NET refers to all other traffic sources (typically set to 'any').



```
-- HOME_NET and EXTERNAL_NET must be set now
-- setup the network addresses you are protecting
HOME_NET = '192.168.1.0/24'

-- set up the external network addresses.
-- (leave as "any" in most situations)
EXTERNAL_NET = 'any'
```

- **Run Snort in detection mode**
  Command:
  sudo snort -c /etc/snort/snort.lua -R /etc/snort/local.rules -I eth0 -A alert_fast -s 65535 -k none



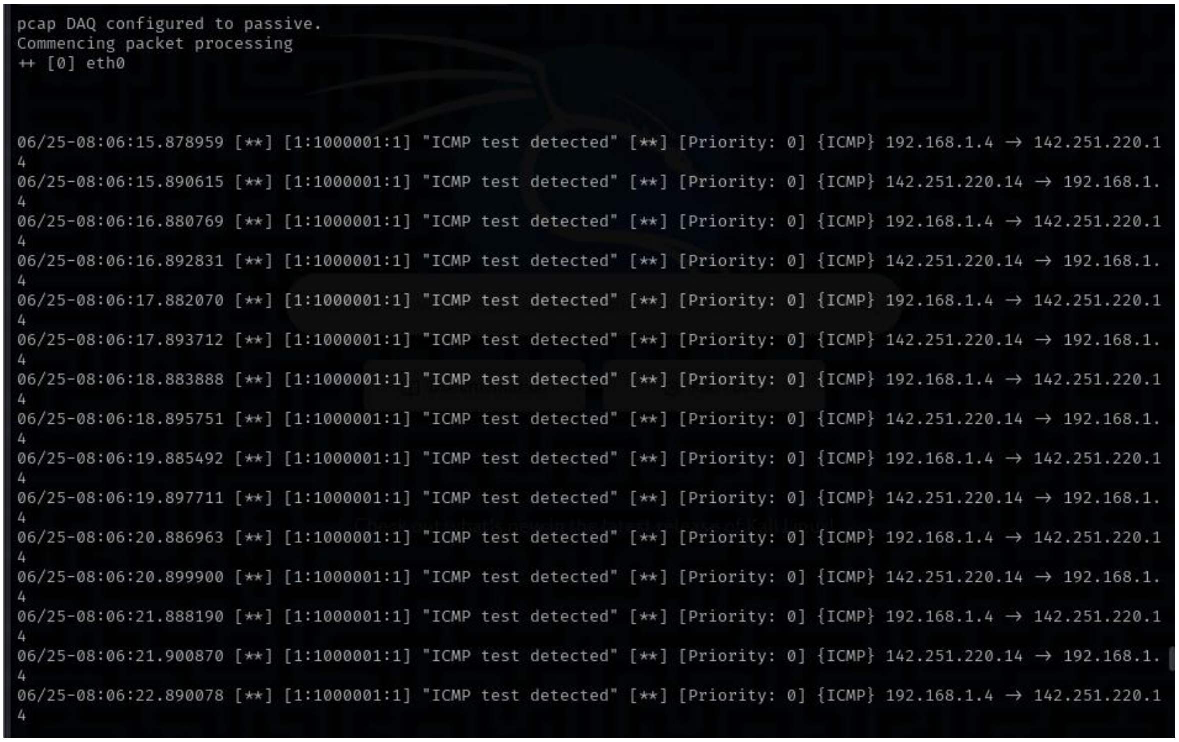
```
┌──(kali㉿kali)-[/usr/local/snort/etc]
└─$ sudo snort -c /etc/snort/snort.lua -R /etc/snort/local.rules -i eth0 -A alert_fast -s 65535 -k none

--------------------------------------------------
o")~   Snort++ 3.1.82.0
--------------------------------------------------
Loading /etc/snort/snort.lua:
Loading snort_defaults.lua:
Finished snort_defaults.lua:
        appid
        wizard
        binder
        ips
        file_id
        trace
```

Command Breakdown:

sudo :	Runs Snort as superuser
snort :	Launches the snort executable
-c /etc/snort/snort.lua :	Specifies the main snort configuration file
-R /etc/snort/local.rules :	Loads custom rule file directly
-i eth0 :	Tells snort to monitor the eth0 network interface
-A alert_fast :	Sets alert output mode to “fast”
-s 65535 :	Sets snap length capture full packets up to 65535
-k none :	Disables checksum verification

- **Monitor live traffic and alerts**

  After running Snort in detection mode, alerts are generated in real time, allowing you to monitor live network traffic and detect potential intrusions as they occur.



```
pcap DAQ configured to passive.
Commencing packet processing
++ [0] eth0



06/25-08:06:15.878959 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 192.168.1.4 → 142.251.220.1
4
06/25-08:06:15.890615 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 142.251.220.14 → 192.168.1.
4
06/25-08:06:16.880769 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 192.168.1.4 → 142.251.220.1
4
06/25-08:06:16.892831 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 142.251.220.14 → 192.168.1.
4
06/25-08:06:17.882070 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 192.168.1.4 → 142.251.220.1
4
06/25-08:06:17.893712 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 142.251.220.14 → 192.168.1.
4
06/25-08:06:18.883888 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 192.168.1.4 → 142.251.220.1
4
06/25-08:06:18.895751 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 142.251.220.14 → 192.168.1.
4
06/25-08:06:19.885492 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 192.168.1.4 → 142.251.220.1
4
06/25-08:06:19.897711 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 142.251.220.14 → 192.168.1.
4
06/25-08:06:20.886963 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 192.168.1.4 → 142.251.220.1
4
06/25-08:06:20.899900 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 142.251.220.14 → 192.168.1.
4
06/25-08:06:21.888190 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 192.168.1.4 → 142.251.220.1
4
06/25-08:06:21.900870 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 142.251.220.14 → 192.168.1.
4
06/25-08:06:22.890078 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 192.168.1.4 → 142.251.220.1
4
```

{ WEEK 3 REPORT }

Task:
• Simulate attacks (e.g., ping flood)
• Observe Snort alerts
• Understand Snort alert formats
• Review alert logs

❖ Simulate attacks

A **ping flood** is a basic Denial of Service (DoS) test using ICMP packets. This sends rapid ping requests
to overload the system.

On Kali or any Linux terminal, run:

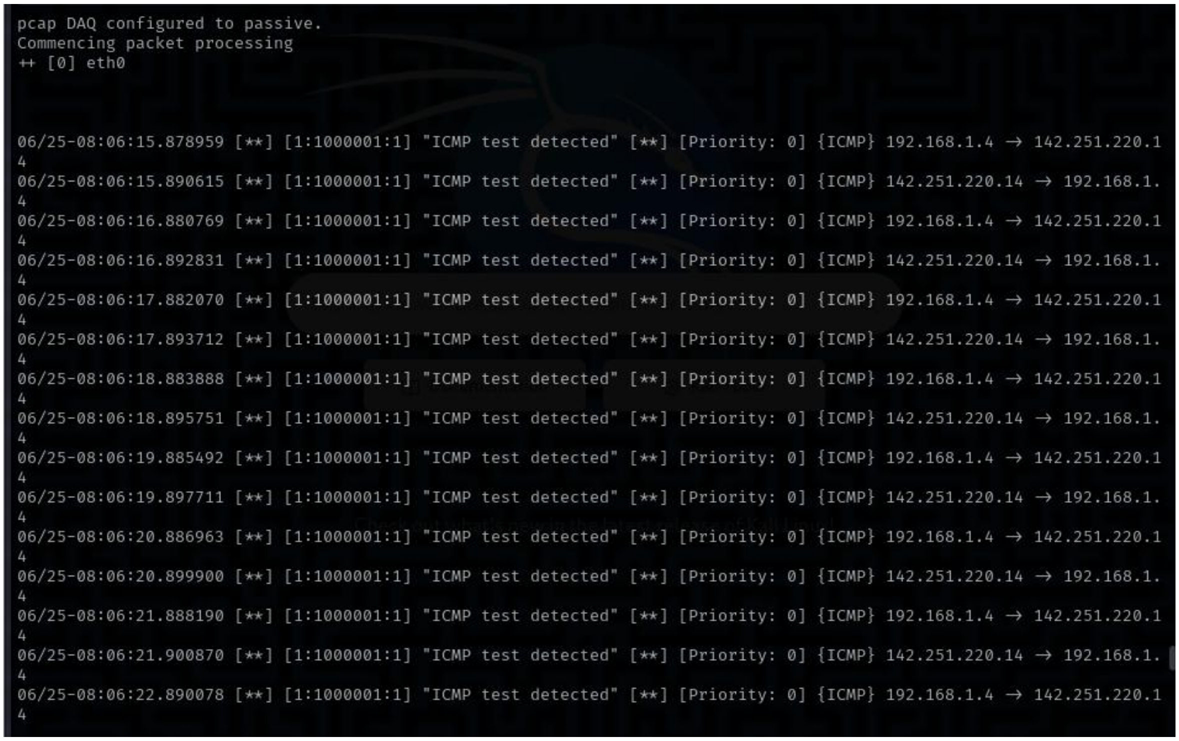
```
Ping -f <target-ip>
```

- Replace <target-ip> with your system's IP (e.g., 192.168.1.5)
- If it says "operation not permitted ," then try with sudo:

```
Sudo ping -f <target-ip>
```

- ❖ **Observe Snort alerts**

Run this command :
sudo snort -c /etc/snort/snort.lua -R /etc/snort/local.rules -I eth0 -A alert_fast -s 65535 -k none



```
pcap DAQ configured to passive.
Commencing packet processing
++ [0] eth0


06/25-08:06:15.878959 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 192.168.1.4 → 142.251.220.1
4
06/25-08:06:15.890615 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 142.251.220.14 → 192.168.1.
4
06/25-08:06:16.880769 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 192.168.1.4 → 142.251.220.1
4
06/25-08:06:16.892831 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 142.251.220.14 → 192.168.1.
4
06/25-08:06:17.882070 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 192.168.1.4 → 142.251.220.1
4
06/25-08:06:17.893712 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 142.251.220.14 → 192.168.1.
4
06/25-08:06:18.883888 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 192.168.1.4 → 142.251.220.1
4
06/25-08:06:18.895751 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 142.251.220.14 → 192.168.1.
4
06/25-08:06:19.885492 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 192.168.1.4 → 142.251.220.1
4
06/25-08:06:19.897711 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 142.251.220.14 → 192.168.1.
4
06/25-08:06:20.886963 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 192.168.1.4 → 142.251.220.1
4
06/25-08:06:20.899900 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 142.251.220.14 → 192.168.1.
4
06/25-08:06:21.888190 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 192.168.1.4 → 142.251.220.1
4
06/25-08:06:21.900870 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 142.251.220.14 → 192.168.1.
4
06/25-08:06:22.890078 [**] [1:1000001:1] "ICMP test detected" [**] [Priority: 0] {ICMP} 192.168.1.4 → 142.251.220.1
4
```

06/25-08:06:21.900870 [**] [1:1000001:1] “ICMP test detected” [**] [Priority: 0] {ICMP} 192.168.1.4 -> 142.251.220.14

This means:

Rule SID: 1000001

Message: “ICMP test detected”

Protocol: ICMP

Source IP: 192.168.1.4

Destination IP: 142.251.220.14

- **Understand Snort alert formats**

06/25-08:06:21.900870 :-	**Timestamp**
[**] :-	**Alert marker**
[1:1000001:1] :-	**[Generator ID : Snort Rule SID : [Revision]**
“ICMP test detected” :-	**Alert message**
[Priority: 0] :-	**Alert priority (0 = low, 1=high)**
{ICMP} :-	**Protocol type**
192.168.1.4 -> 142.251.220.14 :-	**Source → Destination**

{ WEEK 4 REPORT }

Task:
• Explore default Snort rules and structure
• Learn rule components (actions, protocols, etc.)
• Prepare a basic report with screenshots on configuration
and alerts

❖ Explore default Snort rules and structure

Snort uses rule files (with .rules extension) to define how to detect suspicious network activity. These files are usually located in the rules/ directory. In this project, we used the local.rules file to write and test a custom rule.

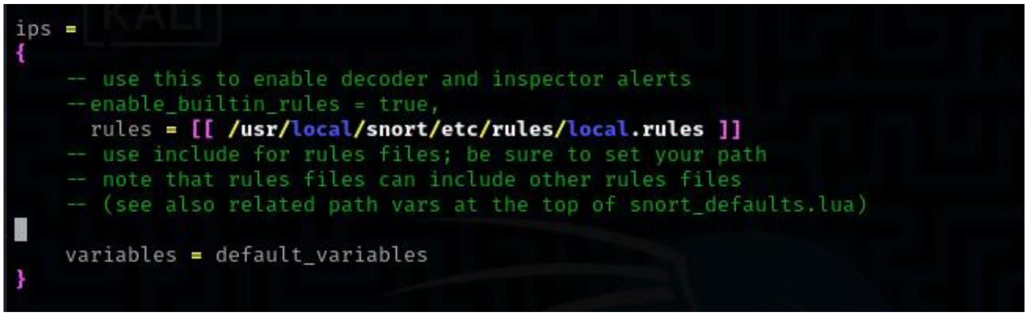
❖ Learn rule components

Each Snort rule consists of several key elements:

- **Action:** Determines what Snort should do (e.g., alert, log, drop)
- **Protocol:** Type of traffic to inspect (tcp, udp, icmp, etc.)
- **Source/Destination IP and Ports:** Defines direction and scope of traffic
- **Options:** Includes metadata like message, rule ID (SID), and revision

❖ Prepare a basic report with screenshots on configuration
and alerts

- Snort was configured using snort.lua
- Custom rules were added in local.rules
- Config path: /usr/local/snort/etc/
- Rule path linked in snort.lua:



```
ips =
{
    -- use this to enable decoder and inspector alerts
    --enable_builtin_rules = true,
      rules = [[ /usr/local/snort/etc/rules/local.rules ]]
    -- use include for rules files; be sure to set your path
    -- note that rules files can include other rules files
    -- (see also related path vars at the top of snort_defaults.lua)

    variables = default_variables
}
```