



# **REPORT OF FIREWALL IDS / IPS**

**MODULE12**

Aniket Sunil Pagare

## **Table of Contents**

---

### **1. Firewall**

- 1.1 What is a Firewall
  - 1.2 Types of Firewalls
  - 1.3 Why Firewall is Important
  - 1.4 Working of a Firewall
- 

### **2. Intrusion Detection System (IDS)**

- 2.1 What is IDS
  - 2.2 Uses/Objectives of IDS
  - 2.3 Types of IDS
  - 2.4 How IDS Detects an Intrusion
  - 2.5 Types of IDS Alerts
- 

### **3. Intrusion Prevention System (IPS)**

- 3.1 What is IPS
  - 3.2 How IPS Works
  - 3.3 Types of IPS
- 

### **4. Honeypot**

- 4.1 What is a Honeypot
  - 4.2 Main Goal of a Honeypot
  - 4.3 How a Honeypot Works
  - 4.4 Types of Honeypots
-

## **5. Snort**

- 5.1 Definition of Snort
  - 5.2 Uses/Objectives of Snort
  - 5.3 Snort Configuration
- 

## **6. Windows Firewall Configuration**

- 6.1 Definition of Windows Firewall
  - 6.2 Purpose of Windows Firewall
  - 6.3 Key Features
  - 6.4 Inbound Rules
  - 6.5 Outbound Rules
- 

## **Extra Activity Section**

---

## **7. ZoneAlarm Firewall**

- 7.1 Definition of ZoneAlarm Firewall
  - 7.2 Key Features
  - 7.3 How to Use ZoneAlarm Firewall
- 

## **8. Attack Detection Tools**

### **8.1 HoneyBot (Attack Detection)**

- 8.1.1 Definition
- 8.1.2 Common Use Cases
- 8.1.3 How to Use HoneyBot

### **8.2 KFSensor (Attack Detection)**

- 8.2.1 Definition
- 8.2.2 Key Features
- 8.2.3 How to Use KFSensor

### **8.3 Wireshark (Attack Detection)**

8.3.1 Definition

8.3.2 Wireshark Features

8.3.3 How to Use Wireshark

---

## **9. Defending Against Evasion Techniques**

### **9.1 Defending Against Firewall Evasion**

9.1.1 Common Firewall Evasion Techniques Attackers Use

9.1.2 How to Defend Against Firewall Evasion

### **9.2 Defending Against IDS Evasion**

9.2.1 Common IDS Evasion Techniques Attackers Use

9.2.2 How to Defend Against IDS Evasion

### **9.3 Defending Against IPS Evasion**

9.3.1 Common IPS Evasion Techniques Attackers Use

9.3.2 How to Defend Against IPS Evasion

---

# **FIREWALL IDS / IPS**

## **What Is Firewall:-**

A **firewall** is a **network security device or software** that monitors and controls **incoming and outgoing network traffic** based on **predefined security rules**. Its main purpose is to **block unauthorized access** to or from a private network while allowing legitimate communication to pass through.

### **🔥 Types of Firewalls:**

#### **1. Packet-Filtering Firewall**

- Checks packets against rules (e.g., IP, port, protocol).
- Basic and fast, but limited inspection.

#### **2. Stateful Inspection Firewall**

- Tracks active connections and makes decisions based on the state of the traffic.
- More secure than packet-filtering.

#### **3. Application-Level Gateway (Proxy Firewall)**

- Filters traffic at the application layer (e.g., HTTP, FTP).
- Can inspect the contents of traffic, providing deep security.

#### **4. Next-Generation Firewall (NGFW)**

- Combines traditional firewall with additional features like:

- Deep packet inspection
- Intrusion prevention
- Application awareness
- Malware protection

## 5. Software Firewall

- Installed on individual devices (like Windows Defender Firewall).
- Protects that single device.

## 6. Hardware Firewall

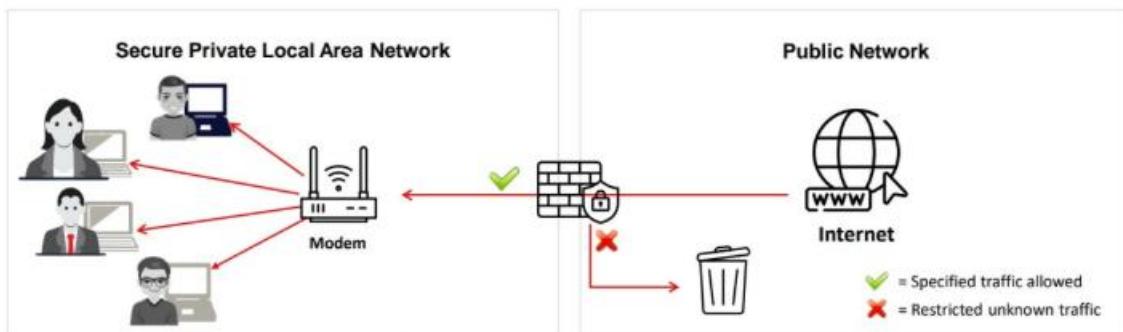
- A physical device placed between your network and the internet.
- Common in business environments.



### Why Firewalls Are Important:

- Prevent **unauthorized access**
- Protect against **malware and cyberattacks**
- Enforce **security policies**
- Log and audit network activity

## **Working Of Firewall :-**



## **What is IDS (Intrusion Detection System)?**

An **Intrusion Detection System (IDS)** is a security tool that monitors and analyzes network or system activity to detect **unauthorized access, malicious behavior, or policy violations**.

### **Uses/Objectives of IDS :**

1. Monitors network or system traffic for suspicious activities.
2. Detects potential intrusions and security breaches.
3. Alerts administrators about detected threats.
4. Identifies malware, exploits, and attack patterns.
5. Helps in early detection of cyber attacks.
6. Assists in forensic analysis after an incident.
7. Enforces organizational security policies.
8. Complements firewalls and other security tools.
9. Tracks user activity to detect insider threats.
10. Logs security events for auditing and compliance.

## **Types Of IDS--:**

### **NIDS (Network-Based Intrusion Detection System)**

#### **Definition:**

Monitors network traffic across a segment to detect suspicious activity.

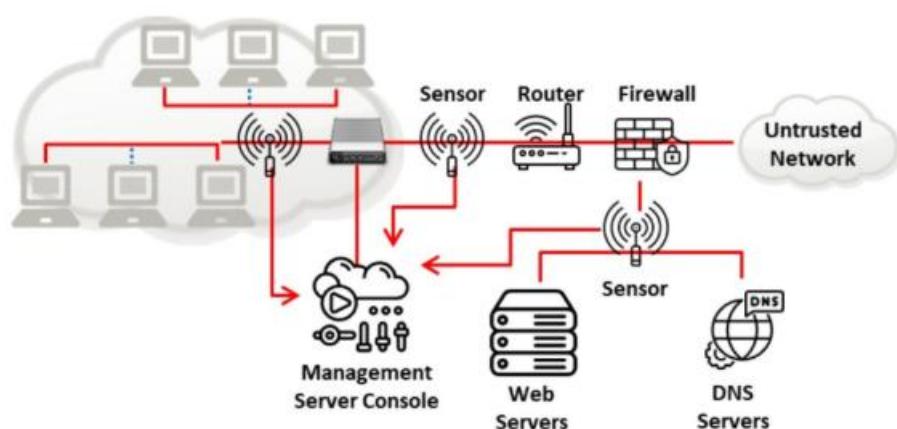
#### **Key Points:**

- Placed at strategic points in the network (e.g., near the gateway or DMZ).
- Analyzes all incoming and outgoing traffic.
- Best for detecting large-scale or external attacks.

**Examples:** Snort, Suricata

#### **Use Cases:**

- Detecting port scans, DoS attacks, and malware traffic.
- Monitoring real-time data flow in networks.



**Fig-:Network-Based IDS**



### **HIDS (Host-Based Intrusion Detection System)**

### **Definition:**

Monitors the internal behavior of a specific host or endpoint.

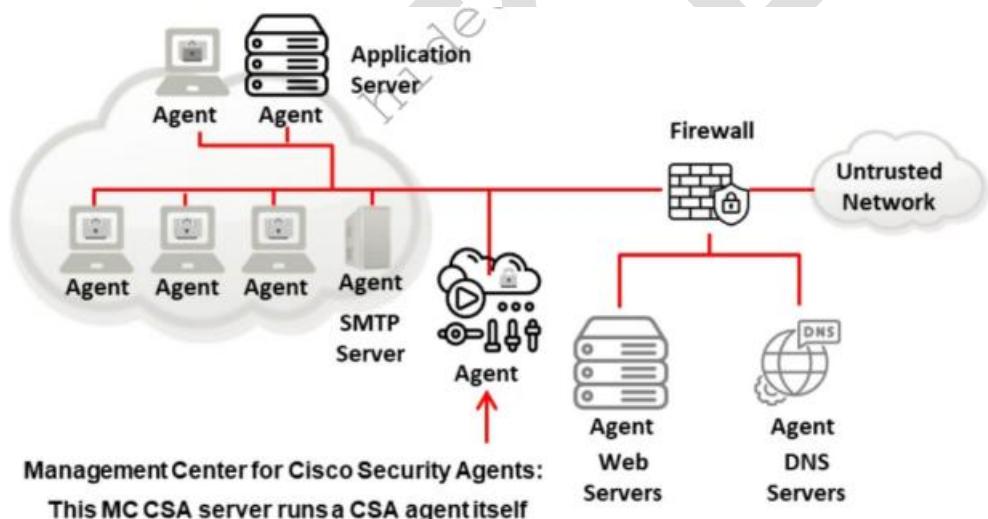
### **Key Points:**

- Installed directly on individual machines (servers, workstations).
- Analyzes system logs, file integrity, and user behavior.
- Best for detecting insider threats or local compromises.

**Examples:** OSSEC, Tripwire

### **Use Cases:**

- Detecting unauthorized file changes or logins.
- Protecting critical servers from local exploits.



**Fig- Host-Based IDS**

### **How an IDS Detects an Intrusion**

An **Intrusion Detection System (IDS)** detects intrusions using one or more of the following techniques:

---

#### **1. Signature-Based Detection**

---

- Compares traffic or system activity against a database of known attack patterns (signatures).
- **Example:** Detects a known malware hash or a specific exploit packet.

 **Strength:** Accurate for known threats

 **Weakness:** Can't detect new or unknown attacks

---

## 2. Anomaly-Based Detection

- Learns normal behavior (baseline) of a system or network, then flags deviations.
- **Example:** Sudden spike in outbound traffic from a server at midnight.

 **Strength:** Detects unknown or zero-day attacks

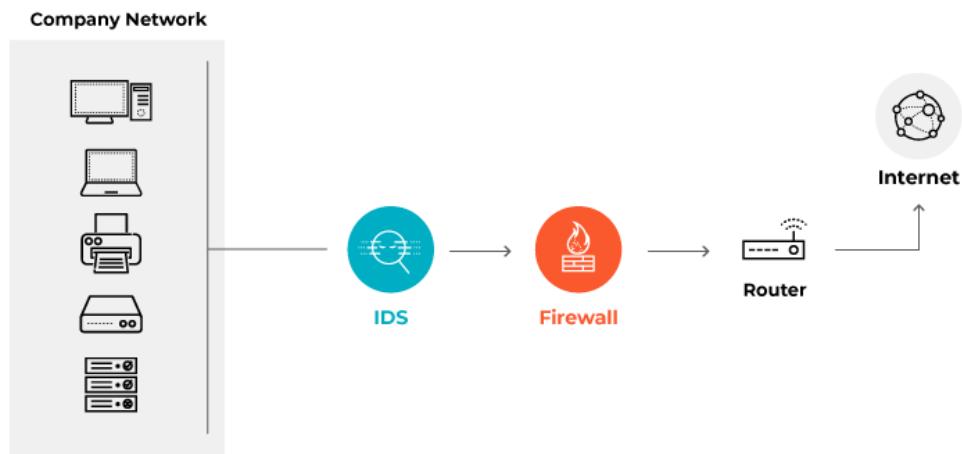
 **Weakness:** May produce false positives

---

## 3. Protocol Anomaly Detection (PAD) in IDS

- **Protocol Anomaly Detection** is a type of **anomaly-based intrusion detection** that identifies intrusions by **detecting deviations from normal behavior defined by protocol standards** (like TCP, HTTP, DNS, etc.).

## Intrusion Detection System



### Types of IDS Alerts

These are categorized based on how accurately the IDS identifies malicious activity:

#### 1. True Positive (TP)

**Attack is present, and the IDS detects it.**

- Correct detection.
- Example: An actual SQL injection attack occurs, and the IDS raises an alert.

#### 2. True Negative (TN)

**No attack is present, and the IDS does not raise an alert.**

- Correct behavior.
- Example: Normal user traffic occurs, and the IDS remains silent.

---

### **3. False Positive (FP)**

 **No attack is present, but the IDS raises an alert.**

-  Incorrect detection.
  - Example: A regular user uploads a file, and the IDS flags it as malware mistakenly.
- 

### **4. False Negative (FN)**

 **Attack is present, but the IDS fails to detect it.**

-  Dangerous because the attack goes unnoticed.
  - Example: A new type of ransomware hits the network, but the IDS doesn't recognize it.
- 

## **What is IPS (Intrusion Prevention System ):-**

 **Definition:**

An **IPS** sits **in-line** (directly in the traffic path) between network devices and **monitors traffic continuously**. When it detects **suspicious activity**, it can **automatically take action** to stop the threat, such as:

- Dropping malicious packets
  - Blocking traffic from an IP address
  - Resetting connections
  - Alerting administrators
-

## **How It Works:**

1. **Traffic Inspection:** It analyzes network traffic using deep packet inspection (DPI).
2. **Signature Matching:** Compares traffic patterns to known attack signatures (like viruses, exploits).
3. **Behavior Analysis:** Detects anomalies (e.g., too many requests from one IP).
4. **Prevention:** Automatically takes action before the attack can succeed

## **Types Of Intrusion Prevention System :-**

### **1. Network-based IPS (NIPS)**

- **Location:** Deployed at key points in the network (e.g., behind a firewall).
- **Purpose:** Monitors all traffic between hosts on the network.
- **Detection Scope:** Protects the entire network.
- **Examples:** Cisco Firepower, Suricata, Snort (IPS mode).

 **Best for:** Detecting and blocking threats that move across the network.

---

### **◆ 2. Host-based IPS (HIPS)**

- **Location:** Installed on individual hosts (servers, desktops, laptops).
- **Purpose:** Monitors system calls, logs, application behavior, and file access.
- **Detection Scope:** Protects only the local system it is installed on.

- **Examples:** OSSEC (with HIPS modules), Symantec Endpoint Protection.

 **Best for:** Detecting exploits like privilege escalation or file modification on a specific machine.

---

#### ◆ 3. Wireless IPS (WIPS)

- **Location:** Monitors wireless network traffic and infrastructure.
- **Purpose:** Detects and blocks rogue devices, unauthorized access points, and wireless attacks (e.g., deauthentication attacks).
- **Examples:** Aruba WIPS, Cisco Meraki AirMarshal.

 **Best for:** Securing wireless environments like corporate Wi-Fi or public access areas.

---

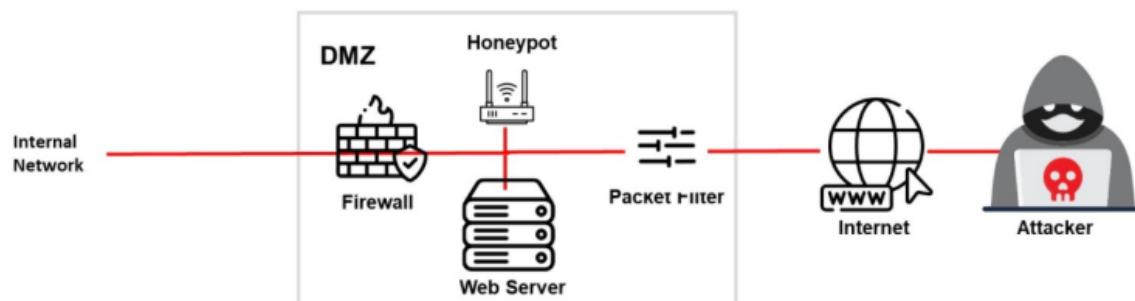
#### ◆ 4. Network Behavior Analysis (NBA) IPS

- **Location:** Typically part of advanced IPS or security appliances.
- **Purpose:** Monitors traffic to detect unusual patterns (e.g., large data transfers, DDoS).
- **Detection Method:** Uses behavior modeling and anomaly detection.
- **Examples:** Darktrace, Stealthwatch (Cisco).

 **Best for:** Identifying zero-day threats, insider threats, and large-scale attacks.

# What is Honeypot?

A **honeypot** is a **cybersecurity mechanism** that is deliberately designed to **attract attackers**. It simulates a vulnerable system, application, or network service, so that attackers interact with it — allowing defenders to **monitor**, **detect**, and **analyze** malicious activity without risking real systems.



## ⌚ Main Goals of a Honeypot

1. **Detection** – Detect unauthorized access and malicious activity.
2. **Diversion** – Distract attackers from real targets.
3. **Analysis** – Study attacker behavior, tools, and techniques.
4. **Prevention** – Improve defenses based on insights gathered.

## ✳️ How a Honeypot Works

1. It mimics a legitimate target like a web server, database, or IoT device.
2. It's placed in a **controlled environment** (isolated from real systems).
3. When an attacker interacts with it (scanning, exploitation, etc.), it **logs** and **records** all actions.
4. The security team uses this data for investigation and threat intelligence.

---

## **Types of HoneyPots:-**

- ◆ **Based on Level of Interaction**

1. **Low-interaction honeypots** – Simulate limited services with minimal risk.
  2. **Medium-interaction honeypots** – Provide more interaction without full OS.
  3. **High-interaction honeypots** – Emulate real systems for in-depth attacker engagement.
- 

- ◆ **Based on Purpose**

4. **Research honeypots** – Used to study attacker behavior and collect threat intelligence.
  5. **Production honeypots** – Deployed in live networks to detect and deflect attacks.
- 

- ◆ **Based on Location**

6. **External honeypots** – Placed outside the firewall to catch external attackers.
  7. **Internal honeypots** – Placed inside the network to detect insider threats or lateral movement.
- 

- ◆ **Based on Technology**

8. **Malware honeypots** – Designed to capture and analyze malware.
9. **Spam honeypots** – Mimic open relays to attract spam emails.

10. **Database honeypots** – Simulate vulnerable databases to catch SQL injection and data exfiltration attempts.
  11. **Web application honeypots** – Mimic websites or CMS systems to detect web-based attacks.
  12. **Industrial Control System (ICS) honeypots** – Simulate SCADA/ICS environments to detect threats targeting critical infrastructure.
- 

◆ **Honeynet**

13. **Honeynet** – A network of multiple interconnected honeypots for broader attack simulation and analysis.

# Snort

**Snort** is an open-source **network intrusion detection system (NIDS)** and intrusion prevention system (IPS) developed by **Cisco Systems**. It's one of the most widely used tools for real-time traffic analysis and packet logging on IP networks.

## Uses/Objectives of Snort:-

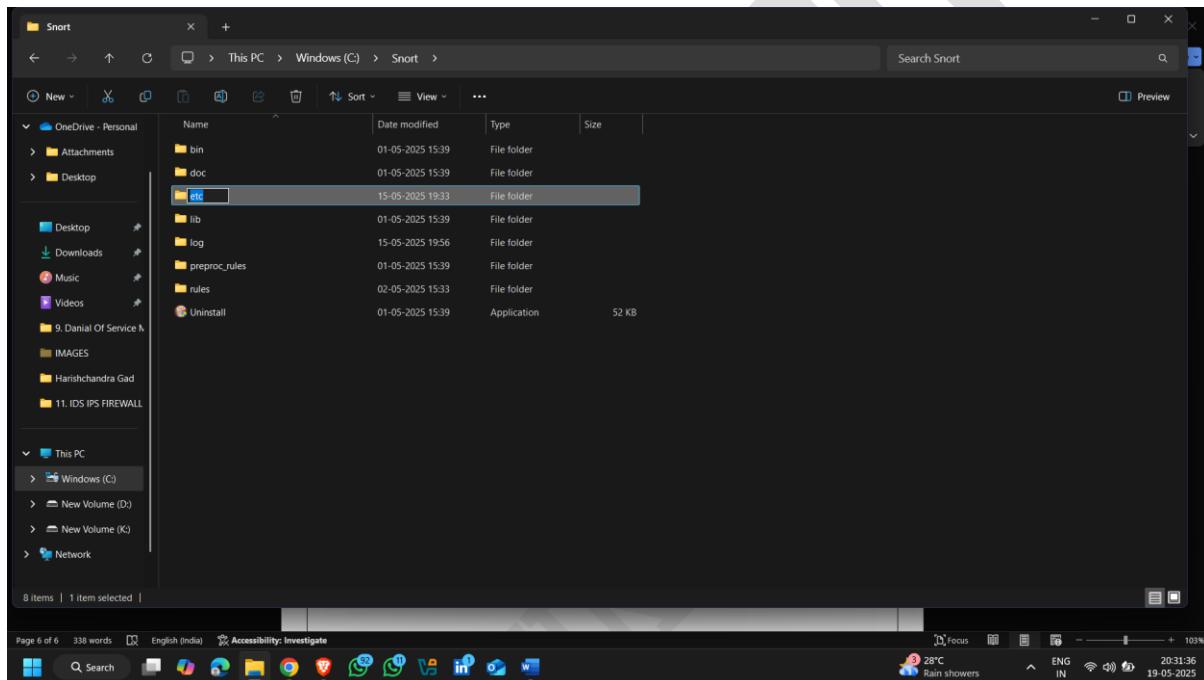
- Detects network intrusions in real-time.
- Performs deep packet inspection (DPI).
- Monitors traffic for suspicious behavior.
- Logs packets for later analysis.
- Prevents attacks when used in IPS mode.
- Detects port scans and probes.
- Identifies malware and exploit attempts.
- Enforces security policies on networks.
- Supports custom rule creation for threat detection.
- Assists in forensic and incident response analysis.

# Snort Configuration

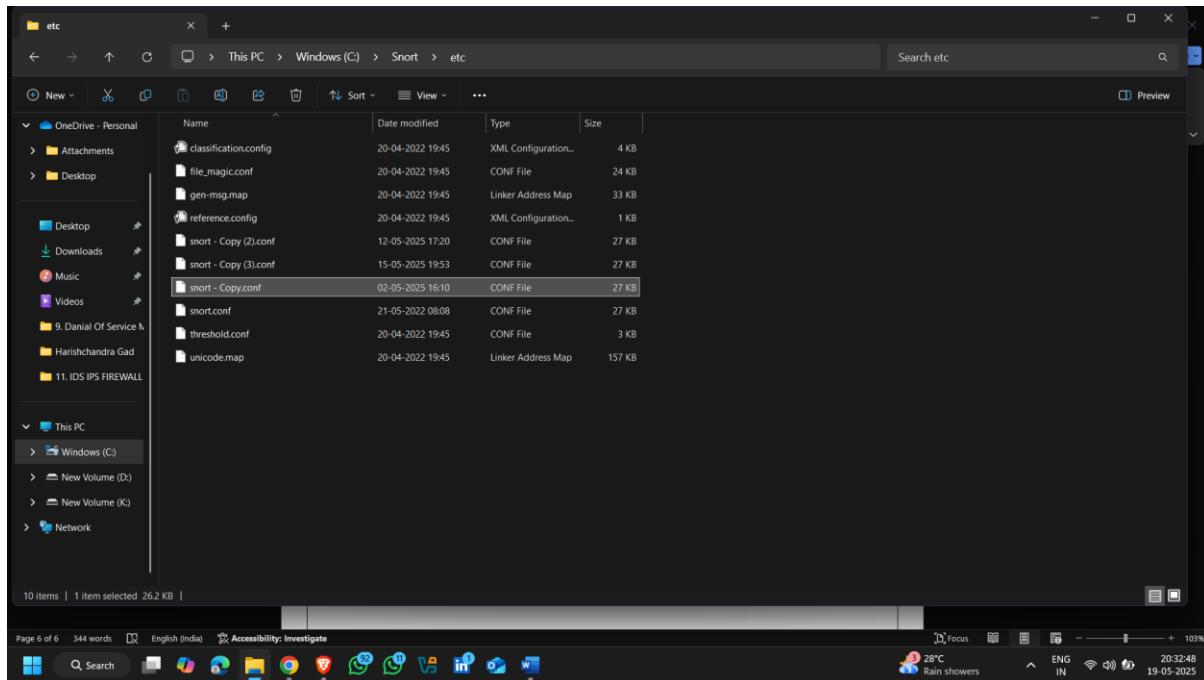
**Download Link :-**<https://www.snort.org/downloads>

**How to use it :-**

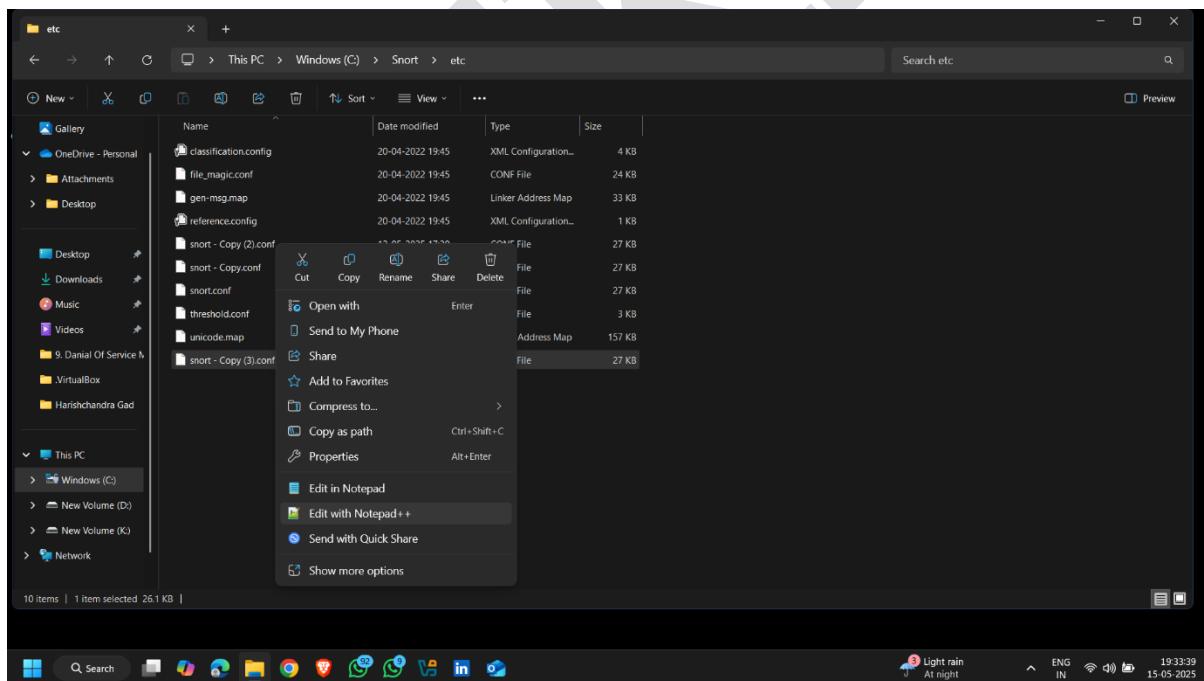
- After installation snort , go to the snort location and open etc folder



- Copy snort.exe file and paste it



- Now , open copied file in **Notepad++**



- Go to the **line number 45**

C:\Snort\etc\snort - Copy (3).conf - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

```
1 # VERSI0NS : 2.9.20
2 #
3 # Short build options:
4 # OPTIONS : --enable-gre --enable-mpls --enable-targetbased --enable-pgm --enable-perfprofiling --enable-zlib --enable-active-response --enable-normalizer --enable-reload --enable-
5 #
6 # Additional information:
7 # This configuration file enables active response, to run snort in
8 # test mode -T you are required to supply an interface -l <interface>
9 # or test mode will fail to fully validate the configuration and
10# exit with a FATAL error
11#
12#####
13# This file contains a sample snort configuration.
14# You should take the following steps to create your own custom configuration:
15#
16# 1) Set the network variables.
17# 2) Configure the decoder
18# 3) Configure the base detection engine
19# 4) Configure dynamic loaded libraries
20# 5) Configure preprocessors
21# 6) Configure output plugins
22# 7) Customize your rule set
23# 8) Customize preprocessor and decoder rule set
24# 9) Customize shared object rule set
25#####
26#
27# Step #1: Set the network variables. For more information, see README.variables
28#
29#
30# Setup the network addresses you are protecting
31ipvar HOME_NET any
32#
33# Set up the external network addresses. Leave as "any" in most situations
34ipvar EXTERNAL_NET any
35#
36# List of DNS servers on your network
37ipvar DNS_SERVERS $HOME_NET
38#
39# List of SMTP servers on your network
40ipvar SMTP_SERVERS $HOME_NET
41#
42#####
43#
44#
45#
46#
47#
48#
49#
50#
51#
52#
53#
54#
```

- And replace **Any** word to **the ip range**

\*C:\Snort\etc\snort - Copy (3).conf - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

```
12 # VERSIONS : 2.9.20
13 #
14 #
15 # Short build options:
16 # OPTIONS : --enable-gre --enable-mpls --enable-targetbased --enable-ppm --enable-perfprofiling --enable-zlib --enable-active-response --enable-normalizer --enable-reload --enable-
17 #
18 # Additional information:
19 # This configuration file enables active response, to run snort in
20 # test mode, if you are required to supply an interface -i <interface>
21 # or test mode will fail to fully validate the configuration and
22 # exit with a FATAL error
23 #
24 -----
25 #####
26 # This file contains a sample snort configuration.
27 # You should take the following steps to create your own custom configuration:
28 #
29 # 1) Set the network variables.
30 # 2) Configure the decoder
31 # 3) Configure the base detection engine
32 # 4) Configure dynamic loaded libraries
33 # 5) Configure preprocessors
34 # 6) Configure output plugins
35 # 7) Customize your rule set
36 # 8) Customize preprocessor and decoder rule set
37 # 9) Customize shared object rule set
38 #####
39 #
40 #####
41 # Step #1: Set the network variables. For more information, see README.variables
42 #####
43 #
44 # Setup the network addresses you are protecting
45 ipvar HOME_NET 192.168.1.0/24
46 #
47 # Set up the external network addresses. Leave as "any" in most situations
48 ipvar EXTERNAL_NET any
49 #
50 # List of DNS servers on your network
51 ipvar DNS_SERVERS $HOME_NET
52 #
53 # List of SMTP servers on your network
54 ipvar SMTP_SERVERS $HOME_NET
55 
```

Properties file

length: 26.809 lines: 690 Ln: 45 Col: 30 Pos: 1.861 Unix (LF) UTF-8 INS

AirSatisfactory Tomorrow ENG IN 15-05-2025

- Go to the **line number 48**

```

16 # OPTIONS : --enable-gre --enable-mpls --enable-targetbased --enable-ppm --enable-perfprofiling --enable-zlib --enable-active-response --enable-normalizer --enable-reload --enal
17 #
18 # Additional information:
19 # This configuration file enables active response, to run snort in
20 # test mode -T you are required to supply an interface -i <interface>
21 # or test mode will fail to fully validate the configuration and
22 # exit with a FATAL error
23 -----
24 #####
25 # This file contains a sample snort configuration.
26 # You should take the following steps to create your own custom configuration:
27 #
28 # 1) Set the network variables.
29 # 2) Configure the decoder
30 # 3) Configure the base detection engine
31 # 4) Configure dynamic loaded libraries
32 # 5) Configure preprocessors
33 # 6) Configure output plugins
34 # 7) Customize your rule set
35 # 8) Customize preprocessor and decoder rule set
36 # 9) Customize shared object rule set
37 #####
38 #####
39 #####
40 #####
41 # Step #1: Set the network variables. For more information, see README.variables
42 #####
43 #
44 # Setup the network addresses you are protecting
45 ipvar HOME_NET 192.168.1.0/24
46 #
47 # Set up the external network addresses. Leave as "any" in most situations
48 ipvar EXTERNAL_NET any
49 #
50 # List of DNS servers on your network
51 ipvar DNS_SERVERS $HOME_NET
52 #
53 # List of SMTP servers on your network
54 ipvar SMTP_SERVERS $HOME_NET
55 #
56 # List of web servers on your network
57 ipvar HTTP_SERVERS $HOME_NET
58 =

```

Properties file

length: 26.809 lines: 690 Ln: 48 Col: 1 Sel: 22 | 1 Unix (LF) UTF-8 INS

Air: Satisfactory Tomorrow ENG IN 19:35:54 15-05-2025

- Replace Any to the !\$HOME\_NET

```

16 # OPTIONS : --enable-gre --enable-mpls --enable-targetbased --enable-ppm --enable-perfprofiling --enable-zlib --enable-active-response --enable-normalizer --enable-reload --enal
17 #
18 # Additional information:
19 # This configuration file enables active response, to run snort in
20 # test mode -T you are required to supply an interface -i <interface>
21 # or test mode will fail to fully validate the configuration and
22 # exit with a FATAL error
23 -----
24 #####
25 # This file contains a sample snort configuration.
26 # You should take the following steps to create your own custom configuration:
27 #
28 # 1) Set the network variables.
29 # 2) Configure the decoder
30 # 3) Configure the base detection engine
31 # 4) Configure dynamic loaded libraries
32 # 5) Configure preprocessors
33 # 6) Configure output plugins
34 # 7) Customize your rule set
35 # 8) Customize preprocessor and decoder rule set
36 # 9) Customize shared object rule set
37 #####
38 #####
39 #####
40 #####
41 # Step #1: Set the network variables. For more information, see README.variables
42 #####
43 #
44 # Setup the network addresses you are protecting
45 ipvar HOME_NET 192.168.1.0/24
46 #
47 # Set up the external network addresses. Leave as "any" in most situations
48 ipvar EXTERNAL_NET !$HOME_NET
49 #
50 # List of DNS servers on your network
51 ipvar DNS_SERVERS $HOME_NET
52 #
53 # List of SMTP servers on your network
54 ipvar SMTP_SERVERS $HOME_NET
55 #
56 # List of web servers on your network
57 ipvar HTTP_SERVERS $HOME_NET
58 =

```

Properties file

length: 26.816 lines: 690 Ln: 48 Col: 30 Pos: 1967 Unix (LF) UTF-8 INS

Air: Satisfactory Tomorrow ENG IN 19:36:12 15-05-2025

- Go to the line number 104
- Set rules folder location

```
74 # List of ports you run web servers on
75 portvar HTTP_PORTS [80,81,311,383,591,593,901,1220,1414,1741,1830,2301,2381,2809,3037,3128,3702,4343,4848,5250,6988,7000,7001,7144,7145,7510,7777,7779,8000,8008,8014,8028,8080,8085,8086]
76
77 # List of ports you want to look for SHELLCODE on.
78 portvar SHELLCODE_PORTS !80
79
80 # List of ports you might see oracle attacks on
81 portvar ORACLE_PORTS 1024-
82
83 # List of ports you want to look for SSH connections on:
84 portvar SSH_PORTS 22
85
86 # List of ports you run ftp servers on
87 portvar FTP_PORTS [21,2100,3535]
88
89 # List of ports you run SIP servers on
90 portvar SIP_PORTS [5060,5061,5060]
91
92 # List of file data ports for file inspection
93 portvar FILE_DATA_PORTS [SHTTP_PORTS,110,143]
94
95 # List of GTP ports for GTP preprocessor
96 portvar GTP_PORTS [2123,2152,3386]
97
98 # other variables, these should not be modified
99 ipvar AIM_SERVERS [64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,205.188.5.0/24,205.188.7.0/24,205.188.9.0/24,205.188.153.0/24,205.188.179.0]
100
101 # Path to your rules files (this can be a relative path)
102 # Note for Windows users: You are advised to make this an absolute path,
103 # such as: c:\snort\rules
104 var RULE_PATH ..\rules
105 var SO_RULE_PATH ..\so_rules
106 var PREPROC_RULE_PATH ..\preproc_rules
107
108 # If you are using reputation preprocessor set these
109 # Currently there is a bug with relative paths, they are relative to where snort is
110 # not relative to snort.conf like the above variables
111 # This is completely inconsistent with how other vars work, BUG 89986
112 # See the absolute path appropriately
113 var WHITE_LIST_PATH ..\rules
114 var BLACK_LIST_PATH ..\rules
115
116 #####
117 # Step #2: Configure the decoder. For more information, see README.decode
118 #####
119
120 # Stop generic_decode_alerts:
121 config disable_decode_alerts
122
123 # Stop Alerts on experimental TCP options
```

- C:\\Snort\\rules

```
74 # List of ports you run web servers on
75 portvar HTTP_PORTS {80,81,311,383,593,901,1220,1414,1741,1830,2301,2381,2809,3037,3128,3702,4343,4848,5250,6988,7000,7001,7144,7145,7510,7777,7779,8000,8008,8014,8028,8080,8085,8086}
76
77 # List of ports you want to look for SHELLCODE on.
78 portvar SHELLCODE_PORTS !80
79
80 # List of ports you might see oracle attacks on
81 portvar ORACLE_PORTS 1024;
82
83 # List of ports you want to look for SSH connections on:
84 portvar SSH_PORTS 22
85
86 # List of ports you run ftp servers on
87 portvar FTP_PORTS {21,2100,3535]
88
89 # List of ports you run SIP servers on
90 portvar SIP_PORTS {5060,5061,5600}
91
92 # List of file data ports for file inspection
93 portvar FILE_DATA_PORTS {8080,8081,110,143]
94
95 # List of GTP ports for GTP preprocessor
96 portvar GTP_PORTS {2123,2152,3386]
97
98 # other variables, these should not be modified
99 ipvar AIM_SERVERS {64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,205.188.5.0/24,205.188.7.0/24,205.188.9.0/24,205.188.153.0/24,205.188.179.0/24,205.188.179.1/24}
100
101 # Path to your rules files (this can be a relative path)
102 # Note for Windows users: You are advised to make this an absolute path,
103 # such as: c:\snort\rules
104 var RULE_PATH c:\snort\rules
105 var SO_RULE_PATH ..\so_rules
106 var PREPROC_RULE_PATH ..\preproc_rules
107
108 # If you are using reputation preprocessor set these
109 # Currently there is a bug with relative paths, they are relative to where snort is
110 # not relative to snort.conf like the above variables
111 # This is completely inconsistent with how other vars work, BUG 89986
112 # Set the absolute path appropriately
113 var WHITELIST_PATH ..\rules
114 var BLACKLIST_PATH ..\rules
115
116 ######
117 # Step #2: Configure the decoder. For more information, see README.decode
118 #####
119
120 # Stop generic decode events:
121 config disable_decode_alerts
122
123 # Stop Alerts on experimental TCP options
```

- Go to the **line number 105**

```

73 # List of ports you run web servers on
74 portvar HTTP_PORTS [80,81,311,383,591,593,901,1220,1414,1741,1830,2301,2381,2809,3037,3128,3702,4343,4848,5250,6988,7000,7001,7144,7145,7510,7777,7779,8000,8008,8014,8028,8080,8085,80
75
76 # List of ports you want to look for SHELLCODE on.
77 portvar SHELLCODE_PORTS !80
78
79
80 # List of ports you might see oracle attacks on
81 portvar ORACLE_PORTS 1024:
82
83 # List of ports you want to look for SSH connections on:
84 portvar SSH_PORTS 22
85
86 # List of ports you run ftp servers on
87 portvar FTP_PORTS [21,2100,3535]
88
89 # List of ports you run SIP servers on
90 portvar SIP_PORTS [5060,5061,5600]
91
92 # List of file data ports for file inspection
93 portvar FILE_DATA_PORTS [SHHTTP_PORTS,110,143]
94
95 # List of GTP ports for GTP preprocessor
96 portvar GTP_PORTS [2123,2152,3386]
97
98 # other variables, these should not be modified
99 ipvar AIM_SERVERS [64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,205.188.5.0/24,205.188.7.0/24,205.188.9.0/24,205.188.153.0/24,205.188.179.0,
100
101 # Path to your rules files (this can be a relative path)
102 # Note for Windows users: You are advised to make this an absolute path,
103 # such as: c:\snort\rules
104 var RULE_PATH c:\snort\rules
105 var SO_RULE_PATH ../so_rules
106 var PREPROC_RULE_PATH ../preproc_rules
107
108 # If you are using reputation preprocessor set these
109 # Currently there is a bug with relative paths, they are relative to where snort is
110 # not relative to snort.conf like the above variables
111 # This is completely inconsistent with how other vars work, BUG 89986
112 # Set the absolute path appropriately
113 var WHITE_LIST_PATH ../rules
114 var BLACK_LIST_PATH ../rules
115
116 ######
117 # Step #2: Configure the decoder. For more information, see README.decode
118 #####
119
120 # Stop generic decode events:
121 config disable_decode_alerts
122
123 # Stop Alerts on experimental TCP options

```

Rain coming 7:58 pm ENG IN 19:38:49 15-05-2025

## • Add # on the front of the line number 105

```

73 # List of ports you run web servers on
74 portvar HTTP_PORTS [80,81,311,383,591,593,901,1220,1414,1741,1830,2301,2381,2809,3037,3128,3702,4343,4848,5250,6988,7000,7001,7144,7145,7510,7777,7779,8000,8008,8014,8028,8080,8085,80
75
76 # List of ports you want to look for SHELLCODE on.
77 portvar SHELLCODE_PORTS !80
78
79
80 # List of ports you might see oracle attacks on
81 portvar ORACLE_PORTS 1024:
82
83 # List of ports you want to look for SSH connections on:
84 portvar SSH_PORTS 22
85
86 # List of ports you run ftp servers on
87 portvar FTP_PORTS [21,2100,3535]
88
89 # List of ports you run SIP servers on
90 portvar SIP_PORTS [5060,5061,5600]
91
92 # List of file data ports for file inspection
93 portvar FILE_DATA_PORTS [SHHTTP_PORTS,110,143]
94
95 # List of GTP ports for GTP preprocessor
96 portvar GTP_PORTS [2123,2152,3386]
97
98 # other variables, these should not be modified
99 ipvar AIM_SERVERS [64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,205.188.5.0/24,205.188.7.0/24,205.188.9.0/24,205.188.153.0/24,205.188.179.0,
100
101 # Path to your rules files (this can be a relative path)
102 # Note for Windows users: You are advised to make this an absolute path,
103 # such as: c:\snort\rules
104 var RULE_PATH c:\snort\rules
105 #var SO_RULE_PATH ../so_rules
106 var PREPROC_RULE_PATH ../preproc_rules
107
108 # If you are using reputation preprocessor set these
109 # Currently there is a bug with relative paths, they are relative to where snort is
110 # not relative to snort.conf like the above variables
111 # This is completely inconsistent with how other vars work, BUG 89986
112 # Set the absolute path appropriately
113 var WHITE_LIST_PATH ../rules
114 var BLACK_LIST_PATH ../rules
115
116 ######
117 # Step #2: Configure the decoder. For more information, see README.decode
118 #####
119
120 # Stop generic decode events:
121 config disable_decode_alerts
122
123 # Stop Alerts on experimental TCP options

```

Rain coming 7:58 pm ENG IN 19:39:09 15-05-2025

## • Go to the line number 106 and set preproc\_rules

```

73 # List of ports you run web servers on
74 portvar HTTP_PORTS [80,81,311,383,591,593,901,1220,1414,1741,1830,2301,2381,2809,3037,3128,3702,4343,4848,5250,6988,7000,7001,7144,7145,7510,7777,7779,8000,8008,8014,8028,8080,8085,80
75
76 # List of ports you want to look for SHELLCODE on.
77 portvar SHELLCODE_PORTS !80
78
79 # List of ports you might see oracle attacks on
80 portvar ORACLE_PORTS 1024:
81
82 # List of ports you want to look for SSH connections on:
83 portvar SSH_PORTS 22
84
85 # List of ports you run ftp servers on
86 portvar FTP_PORTS [21,2100,3535]
87
88 # List of ports you run SIP servers on
89 portvar SIP_PORTS [5060,5061,5600]
90
91 # List of file data ports for file inspection
92 portvar FILE_DATA_PORTS [SHHTTP_PORTS,110,143]
93
94 # List of GTP ports for GTP preprocessor
95 portvar GTP_PORTS [2123,2152,3386]
96
97 # other variables, these should not be modified
98 ipvar AIM_SERVERS [64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,205.188.5.0/24,205.188.7.0/24,205.188.9.0/24,205.188.153.0/24,205.188.179.0,
100
101 # Path to your rules files (this can be a relative path)
102 # Note for Windows users: You are advised to make this an absolute path,
103 # such as: c:\snort\rules
104 var RULE_PATH c:\snort\rules
105 #var SURE_RULE_PATH ../so_rules
106 #var PREPROC_RULE_PATH ..\preproc_rules
107
108 # If you are using reputation preprocessor set these
109 # Currently there is a bug with relative paths, they are relative to where snort is
110 # not relative to snort.conf like the above variables
111 # This is completely inconsistent with how other vars work, BUG 89986
112 # Set the absolute path appropriately
113 var WHITE_LIST_PATH ..\rules
114 var BLACK_LIST_PATH ..\rules
115
116 ##### Step #1: Configure the encoder. For more information, see README.encoder
117 ##### Step #2: Configure the decoder. For more information, see README.decoder
118 #####
119
120 # Stop generic decode events:
121 config disable_decode_alerts
122
123 # Stop Alerts on experimental TCP options

```

## • C:\Snort\preproc\_rules

```

73 # List of ports you run web servers on
74 portvar HTTP_PORTS [80,81,311,383,591,593,901,1220,1414,1741,1830,2301,2381,2809,3037,3128,3702,4343,4848,5250,6988,7000,7001,7144,7145,7510,7777,7779,8000,8008,8014,8028,8080,8085,80
75
76 # List of ports you want to look for SHELLCODE on.
77 portvar SHELLCODE_PORTS !80
78
79 # List of ports you might see oracle attacks on
80 portvar ORACLE_PORTS 1024:
81
82 # List of ports you want to look for SSH connections on:
83 portvar SSH_PORTS 22
84
85 # List of ports you run ftp servers on
86 portvar FTP_PORTS [21,2100,3535]
87
88 # List of ports you run SIP servers on
89 portvar SIP_PORTS [5060,5061,5600]
90
91 # List of file data ports for file inspection
92 portvar FILE_DATA_PORTS [SHHTTP_PORTS,110,143]
93
94 # List of GTP ports for GTP preprocessor
95 portvar GTP_PORTS [2123,2152,3386]
96
97 # other variables, these should not be modified
98 ipvar AIM_SERVERS [64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,205.188.5.0/24,205.188.7.0/24,205.188.9.0/24,205.188.153.0/24,205.188.179.0,
100
101 # Path to your rules files (this can be a relative path)
102 # Note for Windows users: You are advised to make this an absolute path,
103 # such as: c:\snort\rules
104 var RULE_PATH c:\snort\rules
105 #var SURE_RULE_PATH ../so_rules
106 #var PREPROC_RULE_PATH ..\preproc_rules
107
108 # If you are using reputation preprocessor set these
109 # Currently there is a bug with relative paths, they are relative to where snort is
110 # not relative to snort.conf like the above variables
111 # This is completely inconsistent with how other vars work, BUG 89986
112 # Set the absolute path appropriately
113 var WHITE_LIST_PATH ..\rules
114 var BLACK_LIST_PATH ..\rules
115
116 ##### Step #1: Configure the encoder. For more information, see README.encoder
117 ##### Step #2: Configure the decoder. For more information, see README.decoder
118 #####
119
120 # Stop generic decode events:
121 config disable_decode_alerts
122
123 # Stop Alerts on experimental TCP options

```

- go to the line number 113 and 114
- and add rules file path/location

```
80 # List of ports you might see oracle attacks on
81 portvar ORACLE_PORTS 1024;
82
83 # List of ports you want to look for SSH connections on:
84 portvar SSH_PORTS 22
85
86 # List of ports you run ftp servers on
87 portvar FTP_PORTS [21,2100,3535]
88
89 # List of ports you run SIP servers on
90 portvar SIP_PORTS [5060,5061,5060]
91
92 # List of file data ports for file inspection
93 portvar FILE_DATA_PORTS [SHMTP_PORTS,110,143]
94
95 # List of GTP ports for GTP preprocessor
96 portvar GTP_PORTS [2123,2152,3386]
97
98 # other variables, these should not be modified
99 ipvar AIM_SERVERS {64.12.24.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,205.188.5.0/24,205.188.7.0/24,205.188.9.0/24,205.188.153.0/24,205.188.179.0/24}
100
101 # Path to your rules files (this can be a relative path)
102 # Note: If you are using Windows users! You are advised to make this an absolute path,
103 # such as c:\snort\rules
104 var RULE_PATH c:\snort\rules
105 #var SO_RULE_PATH ./so_rules
106 var PREPROC_RULE_PATH c:\Snort\preproc_rules
107
108 # If you are using reputation preprocessor set these
109 # Currently there is a bug with relative paths, they are relative to where snort is
110 # not relative to snort.conf like the above variables
111 # This is completely inconsistent with how other vars work, BUG 89986
112 # Set the absolute path if appropriate
113 var REPUTATION_LIST_PATH c:\Snort\trivuls
114 var BLACK_LIST_PATH c:\snort\tables
115
116 ######
117 # Step #2: Configure the decoder. For more information, see README.decode
118 ######
119
120 # Stop generic decode events:
121 config disable_decode_alerts
122
123 # Stop Alerts on experimental TCP options
124 config disable_tcpopt_experimental_alerts
125
126 # Stop Alerts on obsolete TCP options
127 config disable_tcpopt_obsolete_alerts
128
129 # Stop Alerts on T/TCP alerts
```

- Step 2

```

 1 # such as: ./snortrules
 2
 3 var RULE_PATH c:\snort\rules
 4 #var SO_RULE_PATH ..\so_rules
 5 var PREPROC_RULE_PATH c:\Snort\preproc_rules
 6
 7
 8 # If you are using reputation preprocessor set these
 9 # Currently there is a bug with relative paths, they are relative to where snort is
10 # not relative to snort.conf like the above variables
11 # This is completely inconsistent with how other vars work, BUG 89986
12 # Set the absolute path appropriately
13 var WHITE_LIST_PATH c:\Snort\rules
14 var BLACK_LIST_PATH c:\Snort\rules
15
16
17 ##### Step #2: Configure the decoder. For more information, see README.decode #####
18 #####
19
20 # Stop generic decode events:
21 config disable_decode_alerts
22
23 # Stop Alerts on experimental TCP options
24 config disable_tcpopt_experimental_alerts
25
26 # Stop Alerts on obsolete TCP options
27 config disable_tcpopt_obsolete_alerts
28
29 # Stop Alerts on T/TCP alerts
30 config disable_tcpopt_ttcp_alerts
31
32 # Stop Alerts on all other TCPOption type events:
33 config disable_tcpopt_alerts
34
35 # Stop Alerts on invalid ip options
36 config disable_ipopt_alerts
37
38 # Alert if value in length field (IP, TCP, UDP) is greater than length of the packet
39 # config enable_decode_oversized_alerts
40
41 # Same as above, but drop packet if in Inline mode (requires enable_decode_oversized_alerts)
42 # config enable_decode_oversized_drops
43
44 # Configure IP / TCP checksum mode
45 config checksum_mode: all
46
47 # Configure maximum number of flowbit references. For more information, see README.flowbits
48 # config flowbits_size: 64
49
50 # Configure ports to ignore
51 # config ignore_ports: tcp 21 6667:6671 1356
52 # config ignore_ports: udp 1:17 53
53

```

- go to the line number 186

```

143 # CUMULUS_CHECHSUM_HOOG: d11
146 # Configure maximum number of flowbit references. For more information, see README.flowbits
148 # config flowbits_size: 64
149
150 # Configure ports to ignore
151 # config ignore_ports: tcp 21 6667:6671 1356
152 # config ignore_ports: udp 1:17 53
153
154 # Configure active response for non inline operation. For more information, see README.active
155 # config response: eth0 attempts 2
156
157 # Configure DAQ related options for inline operation. For more information, see README.daq
158 #
159 # config daq: <type>
160 # config daq_dir: <dir>
161 # config daq_mode: <mode>
162 # config daq_var: <var>
163 #
164 # <type> ::= pcap | aipacket | dump | nfq | ipq | ipfw
165 # <mode> ::= read-file | passive | inline
166 # <var> ::= arbitrary <name>=<value> passed to DAQ
167 # <dir> ::= path as to where to look for DAQ module so's
168
169 # Configure specific UID and GID to run snort as after dropping privs. For more information see snort -h command line options
170 #
171 # config set_gid:
172 # config set_uid:
173
174 # Configure default snaplen. Snort defaults to MTU of in use interface. For more information see README
175 # config snaplen:
176
177 #
178 # Configure default bpf_file to use for filtering what traffic reaches snort. For more information see snort -h command line options (-F)
179 # config bpf_file:
180
181 #
182 # Configure default log directory for snort to log to. For more information see snort -h command line options (-l)
183 # config logdir:
184
185 ##### Step #3: Configure the base detection engine. For more information, see README.decode
186 #####
187
188 # Configure PCRE match limitations
189 config pcre_match_limit: 3500
190 config pcre_recursion: 1500

```

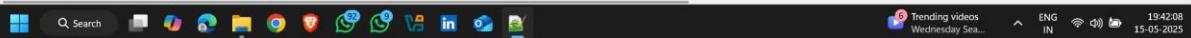


- remove hash

```

143 # CUMULUS_CHECHSUM_HOOG: d11
146 # Configure maximum number of flowbit references. For more information, see README.flowbits
148 # config flowbits_size: 64
149
150 # Configure ports to ignore
151 # config ignore_ports: tcp 21 6667:6671 1356
152 # config ignore_ports: udp 1:17 53
153
154 # Configure active response for non inline operation. For more information, see README.active
155 # config response: eth0 attempts 2
156
157 # Configure DAQ related options for inline operation. For more information, see README.daq
158 #
159 # config daq: <type>
160 # config daq_dir: <dir>
161 # config daq_mode: <mode>
162 # config daq_var: <var>
163 #
164 # <type> ::= pcap | aipacket | dump | nfq | ipq | ipfw
165 # <mode> ::= read-file | passive | inline
166 # <var> ::= arbitrary <name>=<value> passed to DAQ
167 # <dir> ::= path as to where to look for DAQ module so's
168
169 # Configure specific UID and GID to run snort as after dropping privs. For more information see snort -h command line options
170 #
171 # config set_gid:
172 # config set_uid:
173
174 # Configure default snaplen. Snort defaults to MTU of in use interface. For more information see README
175 # config snaplen:
176
177 #
178 # Configure default bpf_file to use for filtering what traffic reaches snort. For more information see snort -h command line options (-F)
179 # config bpf_file:
180
181 #
182 # Configure default log directory for snort to log to. For more information see snort -h command line options (-l)
183 # config logdir:c:\Snort\log
184
185 ##### Step #3: Configure the base detection engine. For more information, see README.decode
186 #####
187
188 # Configure PCRE match limitations
189 config pcre_match_limit: 3500
190 config pcre_recursion: 1500

```



- Step 3
- Don't do anything , no need to change

```

157 # Configure DAQ related options for inline operation. For more information, see README.usq
158 #
159 # config daq: <type>
160 # config daq_dir: <dir>
161 # config daq_mode: <mode>
162 # config daq_var: <var>
163 #
164 # <type> ::= pcap | aifpacket | dump | nfq | ipq | ipfw
165 # <mode> ::= read-file | passive | inline
166 # <var> ::= arbitrary <name>=<value> passed to DAQ
167 # <dir> ::= path as to where to look for DAQ module so's
168 #
169 # Configure specific UID and GID to run snort as after dropping privs. For more information see snort -h command line options
170 #
171 # config set_gid:
172 # config set_uid:
173 #
174 # Configure default snaplen. Snort defaults to MTU of in use interface. For more information see README
175 #
176 # config snaplen:
177 #
178 #
179 # Configure default bpf_file to use for filtering what traffic reaches snort. For more information see snort -h command line options (-F)
180 #
181 # config bpf_file:
182 #
183 #
184 # Configure default log directory for snort to log to. For more information see snort -h command line options (-l)
185 #
186 config logdir:c:\snort\log
187 #
188 #####
189 # Step #3: Configure the base detection engine. For more information, see README.decode
190 #####
191 #
192 #
193 # Configure PCRE match limitations
194 config pcre_match_limit: 3500
195 config pcre_match_limit_recursion: 1500
196 #
197 # Configure the detection engine. See the Snort Manual, Configuring Snort - Includes - Config
198 config detection: search-method ac-split search-optimize max-pattern-len 20
199 #
200 # Configure the event queue. For more information, see README.event_queue
201 config event_queue: max_queue 8 log 5 order_events content_length
202 #
203 #####
204 ## Configure GTP if it is to be used.
205 ## For more information, see README.GTP
206 #####

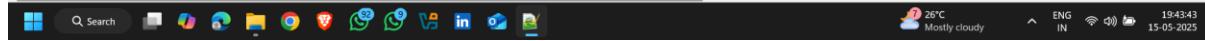
```



- Step 4

```

227 #####
228 # Configure Perf Profiling for debugging
229 # For more information see README.Perfprofiling
230 #####
231 #
232 #config profile_rules: print all, sort avg_ticks
233 #config profile_preprocs: print all, sort avg_ticks
234 #
235 #####
236 # Configure protocol aware flushing
237 # For more information see README.stream5
238 #####
239 config paf_max: 16000
240 #
241 #####
242 # Step #4: Configure dynamic loaded libraries.
243 # For more information, see Snort Manual, Configuring Snort - Dynamic Modules
244 #####
245 #
246 # path to dynamic preprocessor libraries
247 dynamicpreprocessor directory /usr/local/lib/snort_dynamicpreprocessor/
248 #
249 # path to base preprocessor engine
250 dynamicengine /usr/local/lib/snort_dynamicengine/libsf_engine.so
251 #
252 # path to dynamic rules libraries
253 dynamicrules directory /usr/local/lib/snort_dynamicrules
254 #
255 #####
256 # Step #5: Configure preprocessors
257 # For more information, see the Snort Manual, Configuring Snort - Preprocessors
258 #####
259 #
260 # GTP Control Channel Preprocessor. For more information, see README.GTP
261 # preprocessor gtp: ports { 2123 3386 2152 }
262 #
263 # Inline packet normalization. For more information, see README.normalize
264 # Does nothing in IDS mode
265 preprocessor normalize_ip4
266 preprocessor normalize_tcp: ips ecn stream
267 preprocessor normalize_ip6
268 preprocessor normalize_ip6
269 preprocessor normalize_icmp6
270 #
271 # Target-based IP defragmentation. For more information, see README.frag3
272 preprocessor frag3_global: max frags 65536
273 preprocessor frag3_engine: policy windows detect_anomalies overlap_limit 10 min_fragment_length 100 timeout 180
274 #
275 # Target-Based stateful inspection/stream reassembly. For more information, see README.stream5
276 streams4_reassembly streams 4/164! track tm via: \
```



- Go to the line number 247

```

227 ##### Configure Perf Profiling for debugging
228 # For more information see README.Perfprofiling
229 ##########
230
231 #config profile_rules: print all, sort avg_ticks
232 #config profile_pprocs: print all, sort avg_ticks
233
234 #####
235 # Configure protocol aware flushing
236 # For more information see README.stream5
237 ##########
238 config paf_max: 1600
239
240 #####
241 # Step #4: Configure dynamic loaded libraries.
242 # For more information, see Snort Manual, Configuring Snort - Dynamic Modules
243 ##########
244
245
246 # path to dynamic preprocessor libraries
247 dynamicppreprocessor directory /usr/local/lib/snort_dynamicppreprocessor/
248
249 # path to base preprocessor engine
250 dynamicengine /usr/local/lib/snort_dynamicengine/libsf_engine.so
251
252 # path to dynamic rules libraries
253 dynamicdetection directory /usr/local/lib/snort_dynamicrules
254
255 #####
256 # Step #5: Configure preprocessors
257 # For more information, see the Snort Manual, Configuring Snort - Preprocessors
258 ##########
259
260 # GTP Control Channel Preprocessor. For more information, see README.GTP
261 # preprocessor gtp: ports { 2123 3386 2152 }
262
263 # Inline packet normalization. For more information, see README.normalize
264 # Does nothing in IDS mode
265 preprocessor normalize_ip4
266 preprocessor normalize_tcp: ipo ecn stream
267 preprocessor normalize_icmp4
268 preprocessor normalize_ip6
269 preprocessor normalize_icmp6
270
271 # Target-based IP defragmentation. For more information, see README.frag3
272 preprocessor frag3_global: max frags 65536
273 preprocessor frag3_engine: policy windows detect_anomalies overlap_limit 10 min_fragment_length 100 timeout 180
274
275 # Target-Based stateful inspection/stream reassembly. For more information, see README.stream5
276 preprocessor stream5_global: track tmr ver. \

```

- Set snort\_dynamicppreprocessor file path/location :-  
**c:\Snort\lib\snort\_preprocessor**

```

227 ##### Configure Perf Profiling for debugging
228 # For more information see README.Perfprofiling
229 ##########
230
231 #config profile_rules: print all, sort avg_ticks
232 #config profile_pprocs: print all, sort avg_ticks
233
234 #####
235 # Configure protocol aware flushing
236 # For more information see README.stream5
237 ##########
238 config paf_max: 1600
239
240 #####
241 # Step #4: Configure dynamic loaded libraries.
242 # For more information, see Snort Manual, Configuring Snort - Dynamic Modules
243 ##########
244
245
246 # path to dynamic preprocessor libraries
247 dynamicppreprocessor directory c:\Snort\lib\snort_dynamicppreprocessor/
248
249 # path to base preprocessor engine
250 dynamicengine /usr/local/lib/snort_dynamicengine/libsf_engine.so
251
252 # path to dynamic rules libraries
253 dynamicdetection directory /usr/local/lib/snort_dynamicrules
254
255 #####
256 # Step #5: Configure preprocessors
257 # For more information, see the Snort Manual, Configuring Snort - Preprocessors
258 ##########
259
260 # GTP Control Channel Preprocessor. For more information, see README.GTP
261 # preprocessor gtp: ports { 2123 3386 2152 }
262
263 # Inline packet normalization. For more information, see README.normalize
264 # Does nothing in IDS mode
265 preprocessor normalize_ip4
266 preprocessor normalize_tcp: ipo ecn stream
267 preprocessor normalize_icmp4
268 preprocessor normalize_ip6
269 preprocessor normalize_icmp6
270
271 # Target-based IP defragmentation. For more information, see README.frag3
272 preprocessor frag3_global: max frags 65536
273 preprocessor frag3_engine: policy windows detect_anomalies overlap_limit 10 min_fragment_length 100 timeout 180
274
275 # Target-Based stateful inspection/stream reassembly. For more information, see README.stream5
276 preprocessor stream5_global: track tmr ver. \

```

- Go to the line number number 250
- Set sf\_engine.dll file location :-  
**c:\Snort\lib\snort\_dynamicengine/sf\_engine.dll**

```
625
227 ##### Configure Perf Profiling for debugging
228 # For more information see README.Perfprofiling
229 ##########
230
231 #config profile_rules: print all, sort avg_ticks
232 #config profile_preprefs: print all, sort avg_ticks
233
234 ##########
235 # Configure protocol aware flushing
236 # For more information see README.stream5
237 ##########
238 config paf_max: 16000
239
240
241 ##########
242 # Step #4: Configure dynamic loaded libraries.
243 # For more information see Snort Manual, Configuring Snort - Dynamic Modules
244 ##########
245
246 # path to dynamic preprocessor libraries
247 | dynamicpreprocessor directory c:\Snort\lib\snort_dynamicpreprocessor/
248
249 # path to base preprocessor engine
250 | dynamicengine c:\Snort\lib\snort_dynamicengine\sf_engine.dll
251
252 # path to dynamic rules libraries
253 dynamicdetection directory /usr/local/lib/snort_dynamicrules
254
255 ##########
256 # Step #5: Configure preprocessors
257 # For more information, see the Snort Manual, Configuring Snort - Preprocessors
258 ##########
259
260 # GTP Control Channel Preprocessor. For more information, see README.GTP
261 # preprocessor gtp: ports { 2123 3386 2152 }
262
263 # Inline packet normalization. For more information, see README.normalize
264 # Does nothing in IDS mode
265 preprocessor normalize_ip4
266 preprocessor normalize_tcp: ips ecn stream
267 preprocessor normalize_icmp4
268 preprocessor normalize_ip6
269 preprocessor normalize_icmp6
270
271 # Target-based IP defragmentation. For more information, see README.frag3
272 preprocessor frag3_global: max frags 65536
273 preprocessor frag3_engine: policy windows detect_anomalies overlap_limit 10 min_fragment_length 100 timeout 180
274
275 # Target-Based stateful inspection/stream reassembly. For more information, see README.stream5
276 preprocessor stream5_global: track_tcp yes. \
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
```

- Go to the line number 253

```
625
233 ##### Configure protocol aware flushing
234 # For more information see README.stream5
235 ##########
236 config paf_max: 16000
237
238 ##########
239 # path to dynamic preprocessor libraries
240 | dynamicpreprocessor directory c:\Snort\lib\snort_dynamicpreprocessor/
241
242 # path to base preprocessor engine
243 | dynamicengine c:\Snort\lib\snort_dynamicengine\sf_engine.dll
244
245 # path to dynamic rules libraries
246 | dynamicdetection directory /usr/local/lib/snort_dynamicrules
247
248 ##########
249 # Step #5: Configure preprocessors
250 # For more information, see the Snort Manual, Configuring Snort - Preprocessors
251 ##########
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
```

- Remove # on front of the line number 253

```
236 #config profile_rules: print_dif, sort_avg_ticks
237 #config profile_procs: print_all, sort_avg_ticks
238 ######
239 # Configure protocol aware flushing
240 # For more information see README.stream5
241 ######
242 config paf_max: 16000
243 ######
244 # Step #4: Configure dynamic loaded libraries.
245 # For more information, see Snort Manual, Configuring Snort - Dynamic Modules
246 ######
247 # path to dynamic preprocessor libraries
248 | dynamicpreprocessor directory c:\Snort\lib\snort_dynamicpreprocessor/
249 |
250 # path to base preprocessor engine
251 | dynamicengine c:\Snort\lib\snort_dynamicengine\sf_engine.dll
252 |
253 # path to dynamic rules libraries
254 | dynamicdetection directory /usr/local/lib/snort_dynamicrules
255 |
256 ######
257 # Step #5: Configure preprocessors
258 # For more information, see the Snort Manual, Configuring Snort - Preprocessors
259 ######
260 # GTP Control Channel Preprocessor. For more information, see README.GTP
261 # preprocessor gtp: ports { 2123 3386 2152 }
262 |
263 # Inline packet normalization. For more information, see README.normalize
264 # Does nothing in IDS mode
265 preprocessor normalize_ip4
266 preprocessor normalize_tcp: ips ecn stream
267 preprocessor normalize_icmp4
268 preprocessor normalize_ip6
269 preprocessor normalize_icmp6
270 |
271 # Target-based IP defragmentation. For more information, see README.frag3
272 preprocessor frag3_global: max frags 65536
273 preprocessor frag3_engine: policy windows detect_anomalies overlap_limit 10 min_fragment_length 100 timeout 180
274 |
275 # Target-Based stateful inspection/stream reassembly. For more information, see README.stream5
276 preprocessor stream5_global: track_tcp yes, \
277 | track_udp yes, \
278 | track_icmp no, \
279 | max_tcp 262144, \
280 | max_udp 131072, \
281 | max_active_responses 2, \
282 | min_response_seconds 5
```

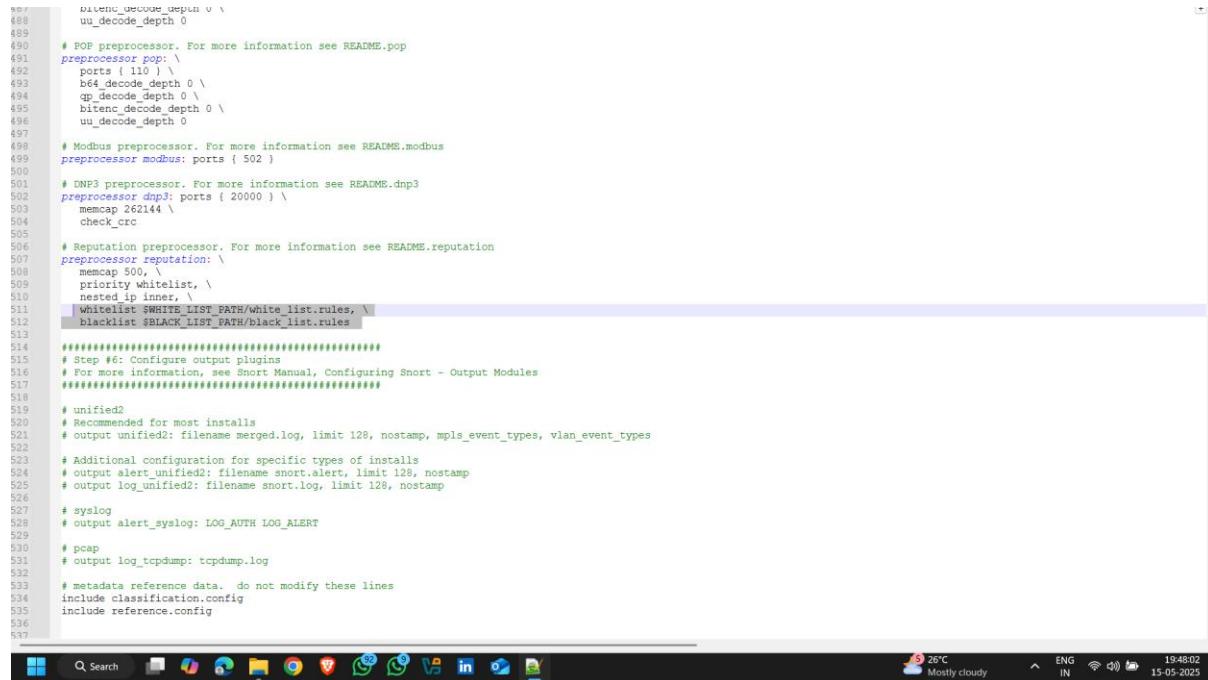
- Step 5

```
236 #config profile_rules: print_dif, sort_avg_ticks
237 #config profile_procs: print_all, sort_avg_ticks
238 ######
239 # Configure protocol aware flushing
240 # For more information see README.stream5
241 ######
242 config paf_max: 16000
243 ######
244 # Step #4: Configure dynamic loaded libraries.
245 # For more information, see Snort Manual, Configuring Snort - Dynamic Modules
246 ######
247 # path to dynamic preprocessor libraries
248 | dynamicpreprocessor directory c:\Snort\lib\snort_dynamicpreprocessor/
249 |
250 # path to base preprocessor engine
251 | dynamicengine c:\Snort\lib\snort_dynamicengine\sf_engine.dll
252 |
253 # path to dynamic rules libraries
254 | dynamicdetection directory /usr/local/lib/snort_dynamicrules
255 |
256 ######
257 # Step #5: Configure preprocessors
258 # For more information, see the Snort Manual, Configuring Snort - Preprocessors
259 ######
260 # GTP Control Channel Preprocessor. For more information, see README.GTP
261 # preprocessor gtp: ports { 2123 3386 2152 }
262 |
263 # Inline packet normalization. For more information, see README.normalize
264 # Does nothing in IDS mode
265 preprocessor normalize_ip4
266 preprocessor normalize_tcp: ips ecn stream
267 preprocessor normalize_icmp4
268 preprocessor normalize_ip6
269 preprocessor normalize_icmp6
270 |
271 # Target-based IP defragmentation. For more information, see README.frag3
272 preprocessor frag3_global: max frags 65536
273 preprocessor frag3_engine: policy windows detect_anomalies overlap_limit 10 min_fragment_length 100 timeout 180
274 |
275 # Target-Based stateful inspection/stream reassembly. For more information, see README.stream5
276 preprocessor stream5_global: track_tcp yes, \
277 | track_udp yes, \
278 | track_icmp no, \
279 | max_tcp 262144, \
280 | max_udp 131072, \
281 | max_active_responses 2, \
282 | min_response_seconds 5
```

- Go to the line number 511 and 512

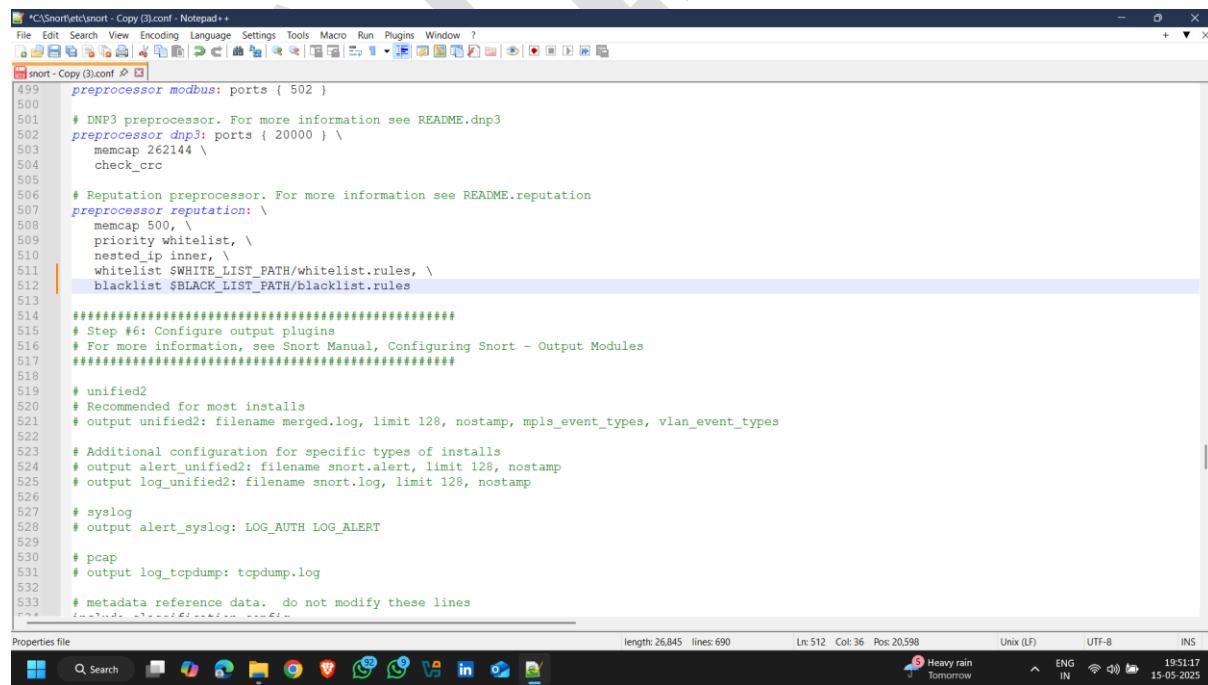
**Note:-** before changing on the line number 511 and 512 , firstly go to the **snort >>rules folder and there is file in folder i.e**

## **blacklist.rules copy file and paste in same folder and rename copy file to the whitelist.rules**



```
507         bitenc_decode_depth 0 \
508         uu_decode_depth 0
509
510     # POP preprocessor. For more information see README.pop
511     preprocessor pop: \
512         ports { 110 } \
513         b64_decode_depth 0 \
514         qp_decode_depth 0 \
515         bitenc_decode_depth 0 \
516         uu_decode_depth 0
517
518     # Modbus preprocessor. For more information see README.modbus
519     preprocessor modbus: ports { 502 }
520
521     # DNP3 preprocessor. For more information see README.dnp3
522     preprocessor dnp3: ports { 20000 } \
523         memcap 262144 \
524         check_crc
525
526     # Reputation preprocessor. For more information see README.reputation
527     preprocessor reputation: \
528         memcap 500, \
529         priority whitelist, \
530         nested_ip_inner, \
531         whitelist $WHITE_LIST_PATH/white_list.rules, \
532         blacklist $BLACK_LIST_PATH/black_list.rules
533
534     ##### Step #6: Configure output plugins #####
535     # For more information, see Snort Manual, Configuring Snort - Output Modules
536
537     # unified2
538     # Recommended for most installs
539     # output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_types
540
541     # Additional configuration for specific types of installs
542     # output alert_unified2: filename snort.alert, limit 128, nostamp
543     # output log_unified2: filename snort.log, limit 128, nostamp
544
545     # syslog
546     # output alert_syslog: LOG_AUTH LOG_ALERT
547
548     # pcap
549     # output log_tcpdump: tcpdump.log
550
551     # metadata reference data. do not modify these lines
552     include classification.config
553     include reference.config
554
555
```

- Replace white\_list.rules to whitelist.rules on 511
- Replace black\_list.rules to blacklist.rules on 512



```
499         preprocessor modbus: ports { 502 }
500
501     # DNP3 preprocessor. For more information see README.dnp3
502     preprocessor dnp3: ports { 20000 } \
503         memcap 262144 \
504         check_crc
505
506     # Reputation preprocessor. For more information see README.reputation
507     preprocessor reputation: \
508         memcap 500, \
509         priority whitelist, \
510         nested_ip_inner, \
511         whitelist $WHITE_LIST_PATH/white_list.rules, \
512         blacklist $BLACK_LIST_PATH/blacklist.rules
513
514     ##### Step #6: Configure output plugins #####
515     # For more information, see Snort Manual, Configuring Snort - Output Modules
516
517     # unified2
518     # Recommended for most installs
519     # output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_types
520
521     # Additional configuration for specific types of installs
522     # output alert_unified2: filename snort.alert, limit 128, nostamp
523     # output log_unified2: filename snort.log, limit 128, nostamp
524
525     # syslog
526     # output alert_syslog: LOG_AUTH LOG_ALERT
527
528     # pcap
529     # output log_tcpdump: tcpdump.log
530
531     # metadata reference data. do not modify these lines
532
533
```

- Step 6
- Don't do anything , no need to change

```

511     whitelist $WHITE_LIST_PATH/whitelist.rules, \
512     blacklist $BLACK_LIST_PATH/blacklist.rules
513
514 ##### Step #6: Configure output plugins
515 # For more information, see Snort Manual, Configuring Snort - Output Modules
516 #####
517
518 # unified2
519 # Recommended for most installs
520 # output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_types
521
522 # Additional configuration for specific types of installs
523 # output alert_unified2: filename snort.alert, limit 128, nostamp
524 # output log_unified2: filename snort.log, limit 128, nostamp
525
526
527 # syslog
528 # output alert_syslog: LOG_AUTH LOG_ALERT
529
530 # pcap
531 # output log_tcpdump: tcpdump.log
532
533 # metadata reference data. do not modify these lines
534 include classification.config
535 include reference.config
536
537 #####
538 # Step #7: Customize your rule set
539 # For more information, see Snort Manual, Writing Snort Rules
540 #
541 # NOTE: All categories are enabled in this conf file
542 #####
543
544 # site specific rules
545 include $RULE_PATH/local.rules
546
547
548 include $RULE_PATH/app-detect.rules
549 include $RULE_PATH/attack-responses.rules
550 include $RULE_PATH/backdoor.rules
551 include $RULE_PATH/bad-traffic.rules
552 include $RULE_PATH/blacklist.rules
553 include $RULE_PATH/botnet-cnc.rules
554 include $RULE_PATH/browser-chrome.rules
555 include $RULE_PATH/browser-firefox.rules
556 include $RULE_PATH/browser-ie.rules
557 include $RULE_PATH/browser-other.rules
558 include $RULE_PATH/browser-plugins.rules
559 include $RULE_PATH/browser-webkit.rules
560 include $RULE_PATH/chat.rules
561 include $RULE_PATH/content-replace.rules
562 include $RULE_PATH/ddos.rules
563 include $RULE_PATH/dns.rules
564 include $RULE_PATH/dos.rules
565 include $RULE_PATH/experimental.rules
566 include $RULE_PATH/exploit-kit.rules
567 include $RULE_PATH/exploit.rules
568 include $RULE_PATH/file-executable.rules
569 include $RULE_PATH/file-flash.rules
570 include $RULE_PATH/file-fuzzing.rules
571

```

Properties file

length: 26,845 lines: 690 Ln: 515 Col: 1 Set: 35 | 1 Unix (LF) UTF-8 INS

Heavy rain Tomorrow ENG IN 19:51:39 15-05-2025

- step 7

```

535 include reference.config
536
537 #####
538 # Step #7: Customize your rule set
539 # For more information, see Snort Manual, Writing Snort Rules
540 #
541 # NOTE: All categories are enabled in this conf file
542 #####
543
544 # site specific rules
545 include $RULE_PATH/local.rules
546
547
548 include $RULE_PATH/app-detect.rules
549 include $RULE_PATH/attack-responses.rules
550 include $RULE_PATH/backdoor.rules
551 include $RULE_PATH/bad-traffic.rules
552 include $RULE_PATH/blacklist.rules
553 include $RULE_PATH/botnet-cnc.rules
554 include $RULE_PATH/browser-chrome.rules
555 include $RULE_PATH/browser-firefox.rules
556 include $RULE_PATH/browser-ie.rules
557 include $RULE_PATH/browser-other.rules
558 include $RULE_PATH/browser-plugins.rules
559 include $RULE_PATH/browser-webkit.rules
560 include $RULE_PATH/chat.rules
561 include $RULE_PATH/content-replace.rules
562 include $RULE_PATH/ddos.rules
563 include $RULE_PATH/dns.rules
564 include $RULE_PATH/dos.rules
565 include $RULE_PATH/experimental.rules
566 include $RULE_PATH/exploit-kit.rules
567 include $RULE_PATH/exploit.rules
568 include $RULE_PATH/file-executable.rules
569 include $RULE_PATH/file-flash.rules
570 include $RULE_PATH/file-fuzzing.rules
571

```

Properties file

length: 26,845 lines: 690 Ln: 539 Col: 3 Set: 32 | 1 Unix (LF) UTF-8 INS

Heavy rain Tomorrow ENG IN 19:51:52 15-05-2025

- go to the line number 546

- replace "/" to "\" 546 upto 651

```

535 include reference.config
536
537 #####
538 # Step #7: Customize your rule set
539 # For more information, see Snort Manual, Writing Snort Rules
540 #
541 # NOTE: All categories are enabled in this conf file
542 #####
543
544 # site specific rules
545 include $RULE_PATH/local.rules
546
547 include $RULE_PATH/app-detect.rules
548 include $RULE_PATH/attack-responses.rules
549 include $RULE_PATH/backdoor.rules
550 include $RULE_PATH/bad-traffic.rules
551 include $RULE_PATH/blacklist.rules
552 include $RULE_PATH/botnet-cnc.rules
553 include $RULE_PATH/browser-chrome.rules
554 include $RULE_PATH/browser-firefox.rules
555 include $RULE_PATH/browser-ie.rules
556 include $RULE_PATH/browser-other.rules
557 include $RULE_PATH/browser-plugins.rules
558 include $RULE_PATH/browser-webkit.rules
559 include $RULE_PATH/chat.rules
560 include $RULE_PATH/content-replace.rules
561 include $RULE_PATH/ddos.rules
562 include $RULE_PATH/dns.rules
563 include $RULE_PATH/dos.rules
564 include $RULE_PATH/experimental.rules
565 include $RULE_PATH/exploit-kit.rules
566 include $RULE_PATH/exploit.rules
567 include $RULE_PATH/file-executable.rules
568 include $RULE_PATH/file-flash.rules
569

```

Properties file length: 26,845 lines: 690 Ln: 546 Col: 20 Sel: 1 | 1 Unix (LF) UTF-8 INS

Heavy rain tomorrow ENG IN 19:51:57 15-05-2025

- press **ctrl + f** , click on replace

Find

Find what: '

Replace with:

Count:

Backward direction

Match whole word only

Match case

Wrap around

Search Mode

Normal

Extended (\n, \r, \t, \b, \v,...)

Regular expression

matches newline

Transparency

On losing focus

Always

Find Next

Find in Current Document

Find All in All Opened Documents

Close

```

535 include reference.config
536
537 #####
538 # Step #7: Customize your rule set
539 # For more information, see Snort Manual, Writing Snort Rules
540 #
541 # NOTE: All categories are enabled in this conf file
542 #####
543
544 # site specific rules
545 include $RULE_PATH/local.rules
546
547 include $RULE_PATH/app-detect.rules
548 include $RULE_PATH/attack-responses.rules
549 include $RULE_PATH/backdoor.rules
550 include $RULE_PATH/bad-traffic.rules
551 include $RULE_PATH/blacklist.rules
552 include $RULE_PATH/botnet-cnc.rules
553 include $RULE_PATH/browser-chrome.rules
554 include $RULE_PATH/browser-firefox.rules
555 include $RULE_PATH/browser-ie.rules
556 include $RULE_PATH/browser-other.rules
557 include $RULE_PATH/browser-plugins.rules
558 include $RULE_PATH/browser-webkit.rules
559 include $RULE_PATH/chat.rules
560 include $RULE_PATH/content-replace.rules
561 include $RULE_PATH/ddos.rules
562 include $RULE_PATH/dns.rules
563 include $RULE_PATH/dos.rules
564 include $RULE_PATH/experimental.rules
565 include $RULE_PATH/exploit-kit.rules
566 include $RULE_PATH/exploit.rules
567 include $RULE_PATH/file-executable.rules
568 include $RULE_PATH/file-flash.rules
569

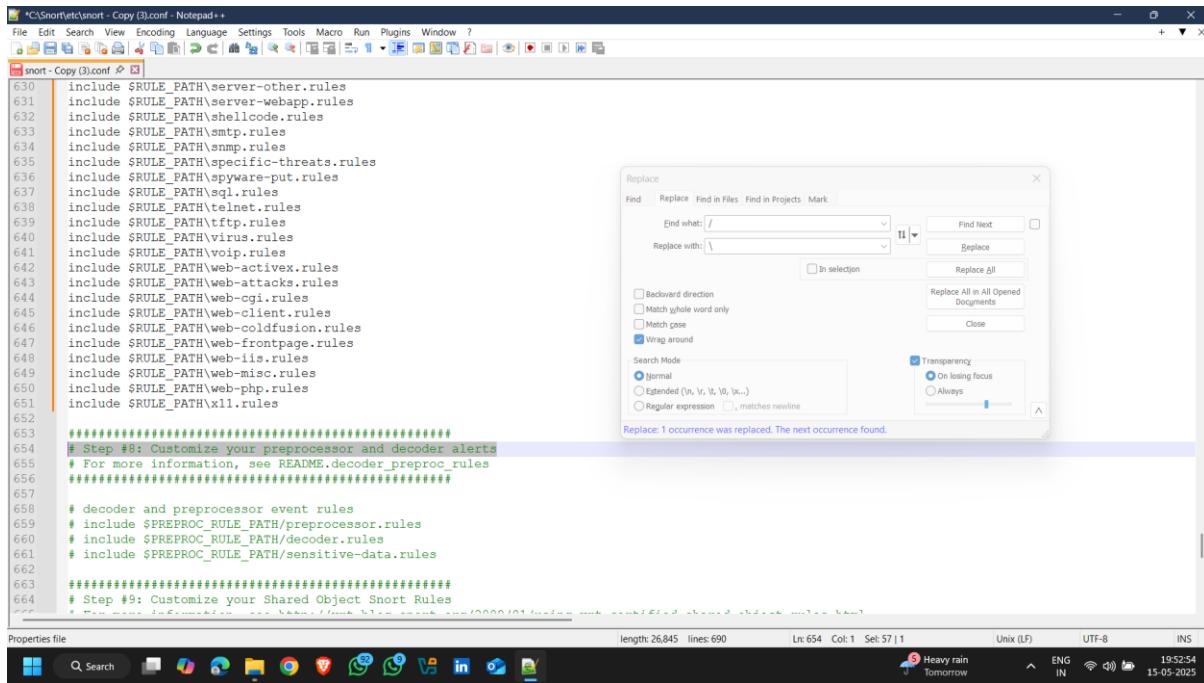
```

Properties file length: 26,845 lines: 690 Ln: 546 Col: 20 Sel: 1 | 1 Unix (LF) UTF-8 INS

Heavy rain tomorrow ENG IN 19:52:12 15-05-2025

- add "/" in first box and add "\", in second column

- click on replace  and replace one by one



```

630 include $RULE_PATH\server-other.rules
631 include $RULE_PATH\server-webapp.rules
632 include $RULE_PATH\shellcode.rules
633 include $RULE_PATH\smtp.rules
634 include $RULE_PATH\sntp.rules
635 include $RULE_PATH\specific-threats.rules
636 include $RULE_PATH\spyware-put.rules
637 include $RULE_PATH\sql.rules
638 include $RULE_PATH\telnet.rules
639 include $RULE_PATH\tftp.rules
640 include $RULE_PATH\virus.rules
641 include $RULE_PATH\voip.rules
642 include $RULE_PATH\web-activex.rules
643 include $RULE_PATH\web-attacks.rules
644 include $RULE_PATH\web-cgi.rules
645 include $RULE_PATH\web-client.rules
646 include $RULE_PATH\web-coldfusion.rules
647 include $RULE_PATH\web-frontpage.rules
648 include $RULE_PATH\web-iis.rules
649 include $RULE_PATH\web-misc.rules
650 include $RULE_PATH\web-php.rules
651 include $RULE_PATH\xml.rules
652 #####
653 # Step #8: Customize your preprocessor and decoder alerts
654 # For more information, see README.decoder_preproc_rules
655 #####
656
657 # decoder and preprocessor event rules
658 # include $PREPROC_RULE_PATH\preprocessor.rules
659 # include $PREPROC_RULE_PATH\decoder.rules
660 # include $PREPROC_RULE_PATH\sensitive-data.rules
661
662 #####
663 # Step #9: Customize your Shared Object Snort Rules
664 # For more information, see http://vrt-blog.snort.org/2009/01/using-vrt-certified-shared-object-rules.html
665
666 #####
667 # dynamic library rules
668 # include $SO_RULE_PATH\bad-traffic.rules
669 # include $SO_RULE_PATH\chat.rules
670 # include $SO_RULE_PATH\dns.rules
671

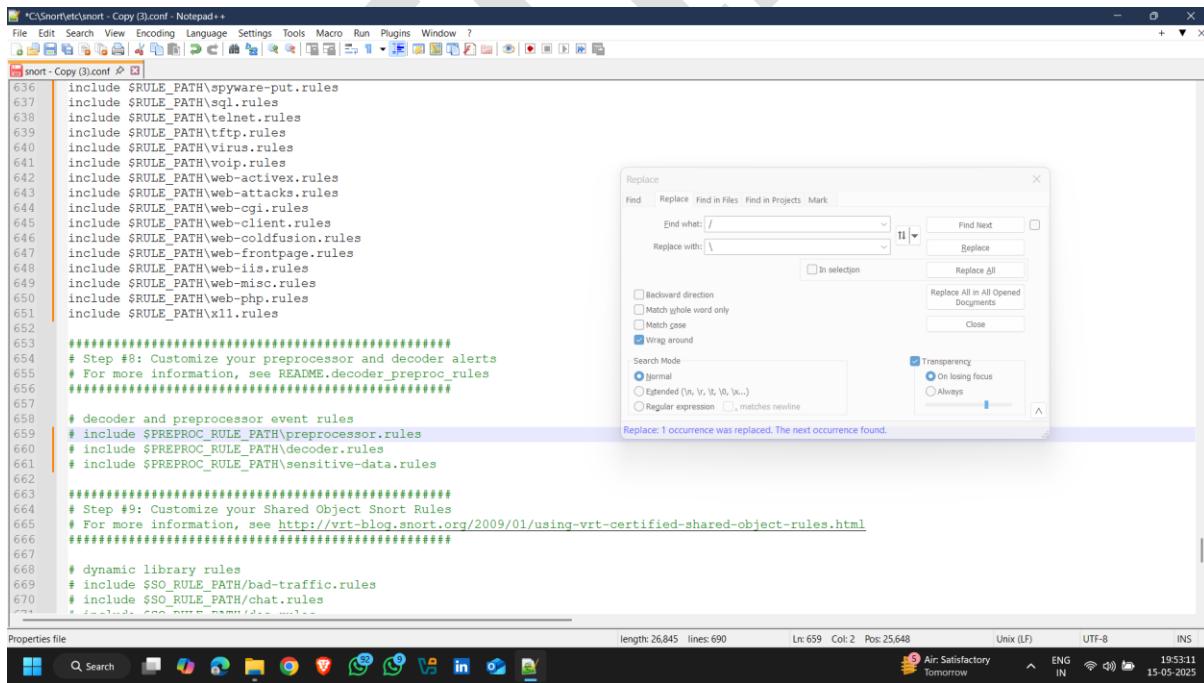
```

Properties file

length: 26,845 lines: 690 Ln: 654 Col: 1 Sel: 57 | 1 Unix (LF) UTF-8 INS

Heavy rain Tomorrow ENG IN 19:52:54 15-05-2025

- step 8



```

630 include $RULE_PATH\spyware-put.rules
631 include $RULE_PATH\sql.rules
632 include $RULE_PATH\telnet.rules
633 include $RULE_PATH\tftp.rules
634 include $RULE_PATH\virus.rules
635 include $RULE_PATH\voip.rules
636 include $RULE_PATH\web-activex.rules
637 include $RULE_PATH\web-attacks.rules
638 include $RULE_PATH\web-cgi.rules
639 include $RULE_PATH\web-client.rules
640 include $RULE_PATH\web-coldfusion.rules
641 include $RULE_PATH\web-frontpage.rules
642 include $RULE_PATH\web-iis.rules
643 include $RULE_PATH\web-misc.rules
644 include $RULE_PATH\web-php.rules
645 include $RULE_PATH\xml.rules
646 #####
647 # Step #8: Customize your preprocessor and decoder alerts
648 # For more information, see README.decoder_preproc_rules
649 #####
650
651 # decoder and preprocessor event rules
652 # include $PREPROC_RULE_PATH\preprocessor.rules
653 # include $PREPROC_RULE_PATH\decoder.rules
654 # include $PREPROC_RULE_PATH\sensitive-data.rules
655
656 #####
657 # Step #9: Customize your Shared Object Snort Rules
658 # For more information, see http://vrt-blog.snort.org/2009/01/using-vrt-certified-shared-object-rules.html
659
660 #####
661 # dynamic library rules
662 # include $SO_RULE_PATH\bad-traffic.rules
663 # include $SO_RULE_PATH\chat.rules
664 # include $SO_RULE_PATH\dns.rules
665

```

Properties file

length: 26,845 lines: 690 Ln: 659 Col: 2 Pos: 25,648 Unix (LF) UTF-8 INS

Air Satisfactory Tomorrow ENG IN 19:53:11 15-05-2025

- follow step 7 process as it is on line number 659 to line number 661

```

636 include $RULE_PATH\spyware-put.rules
637 include $RULE_PATH\sql.rules
638 include $RULE_PATH\telnet.rules
639 include $RULE_PATH\tftp.rules
640 include $RULE_PATH\virus.rules
641 include $RULE_PATH\voip.rules
642 include $RULE_PATH\web-activex.rules
643 include $RULE_PATH\web-attacks.rules
644 include $RULE_PATH\web-cgi.rules
645 include $RULE_PATH\web-client.rules
646 include $RULE_PATH\web-coldfusion.rules
647 include $RULE_PATH\web-frontpage.rules
648 include $RULE_PATH\web-iis.rules
649 include $RULE_PATH\web-misc.rules
650 include $RULE_PATH\web-php.rules
651 include $RULE_PATH\xml.rules
652
653 #####
654 # Step #8: Customize your preprocessor and decoder alerts
655 # For more information, see README.decoder_preproc_rules
656 #####
657
658 # decoder and preprocessor event rules
659 include $PREPROC_RULE_PATH\preprocessor.rules
660 include $PREPROC_RULE_PATH\decoder.rules
661 | include $PREPROC_RULE_PATH\sensitive-data.rules
662
663 #####
664 # Step #9: Customize your Shared Object Snort Rules
665 # For more information, see http://vrt-blog.snort.org/2009/01/using-vrt-certified-shared-object-rules.html
666 #####
667
668 # dynamic library rules
669 # include $SO_RULE_PATH\bad-traffic.rules
670 # include $SO_RULE_PATH\chat.rules
671

```

- Configuration done

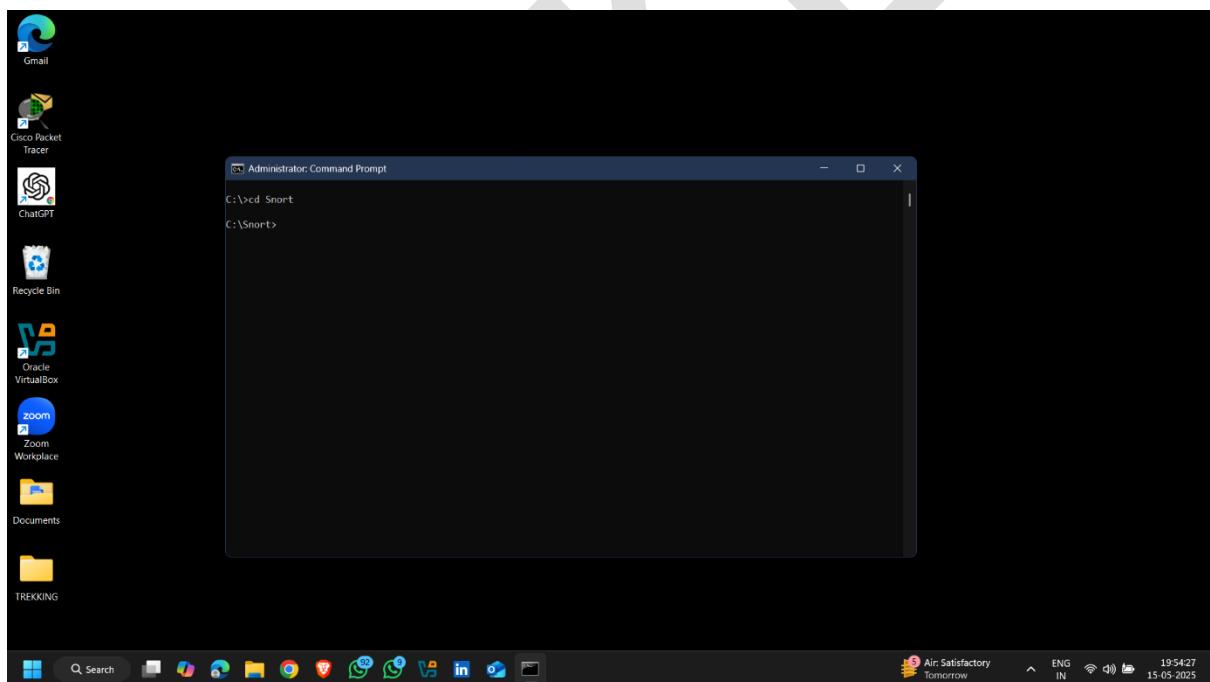
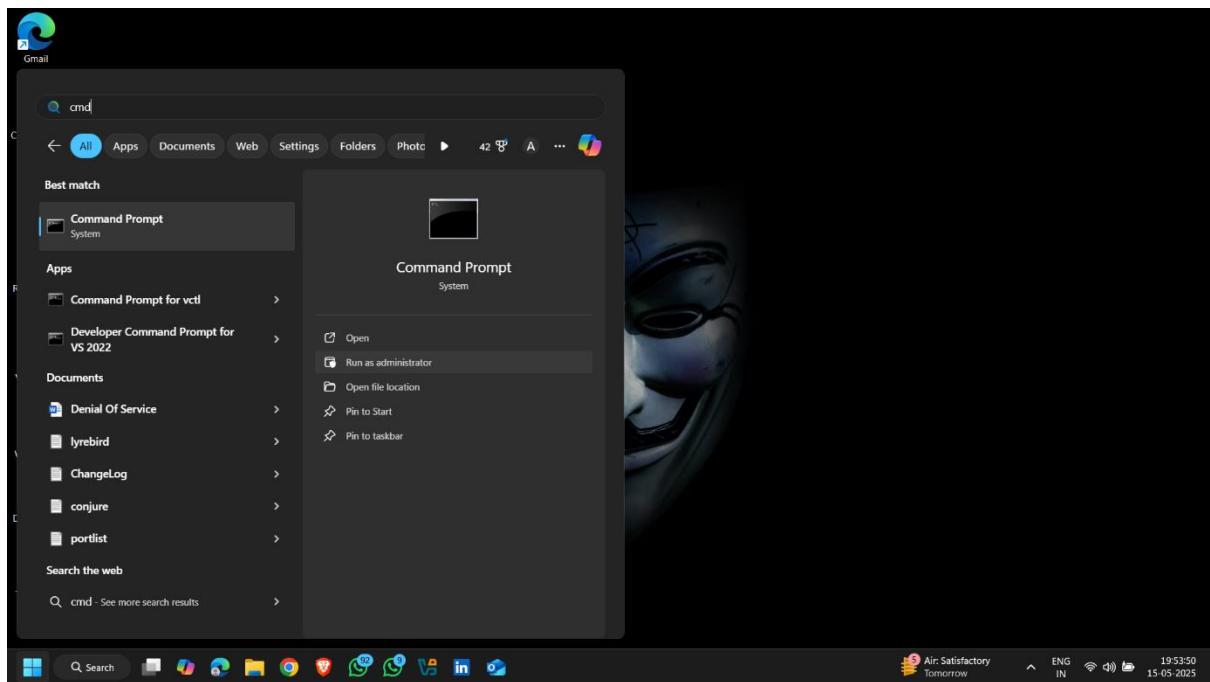
## Note :- Step 9 for Linux Operating System, no need to change

```

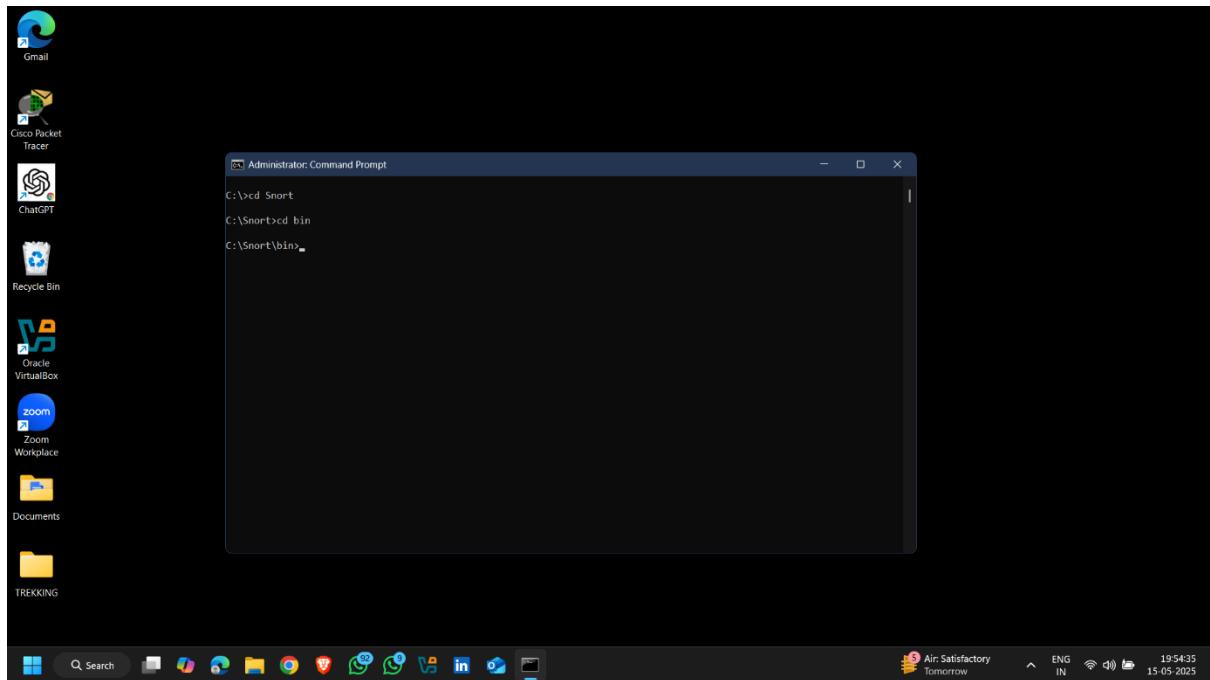
642 include $RULE_PATH\web-activex.rules
643 include $RULE_PATH\web-attacks.rules
644 include $RULE_PATH\web-cgi.rules
645 include $RULE_PATH\web-client.rules
646 include $RULE_PATH\web-coldfusion.rules
647 include $RULE_PATH\web-frontpage.rules
648 include $RULE_PATH\web-iis.rules
649 include $RULE_PATH\web-misc.rules
650 include $RULE_PATH\web-php.rules
651 include $RULE_PATH\xml.rules
652
653 #####
654 # Step #8: Customize your preprocessor and decoder alerts
655 # For more information, see README.decoder_preproc_rules
656 #####
657
658 # decoder and preprocessor event rules
659 include $PREPROC_RULE_PATH\preprocessor.rules
660 include $PREPROC_RULE_PATH\decoder.rules
661 include $PREPROC_RULE_PATH\sensitive-data.rules
662
663 #####
664 # Step #9: Customize your Shared Object Snort Rules
665 # For more information, see http://vrt-blog.snort.org/2009/01/using-vrt-certified-shared-object-rules.html
666 #####
667
668 # dynamic library rules
669 # include $SO_RULE_PATH\bad-traffic.rules
670 # include $SO_RULE_PATH\chat.rules
671 # include $SO_RULE_PATH\dos.rules
672 # include $SO_RULE_PATH\exploit.rules
673 # include $SO_RULE_PATH\icmp.rules
674 # include $SO_RULE_PATH\imap.rules
675 # include $SO_RULE_PATH\misc.rules
676 # include $SO_RULE_PATH\multimedia.rules
677

```

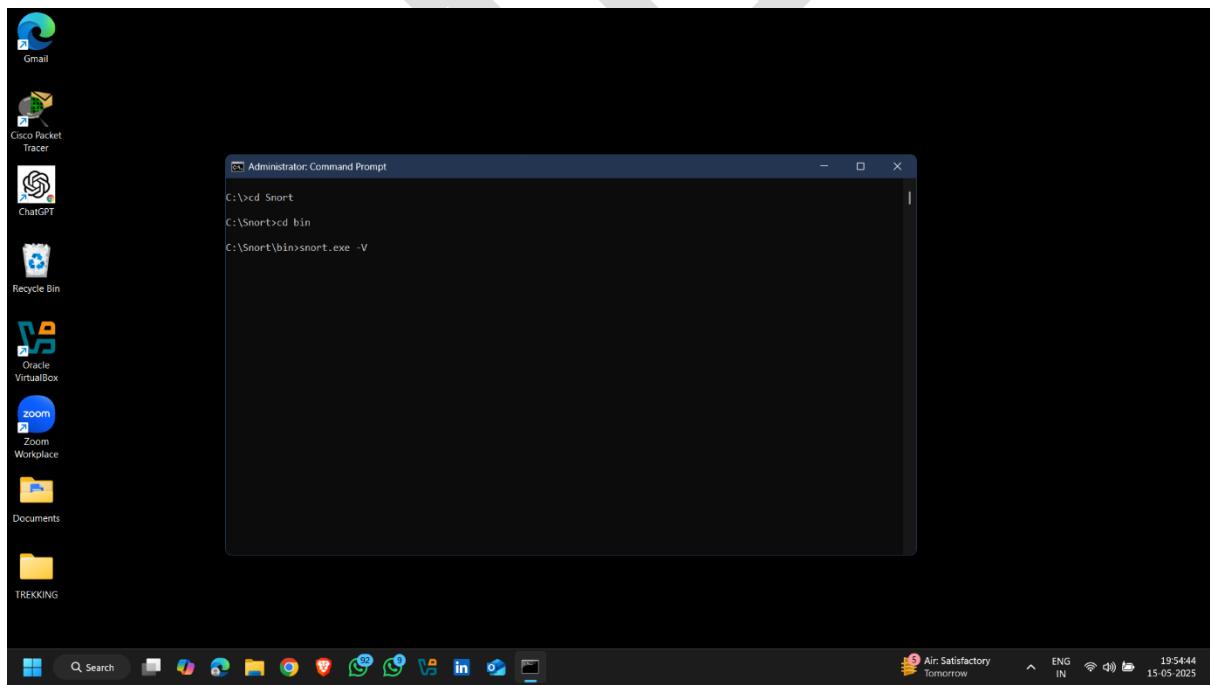
**Now open command line interface as a administrator**

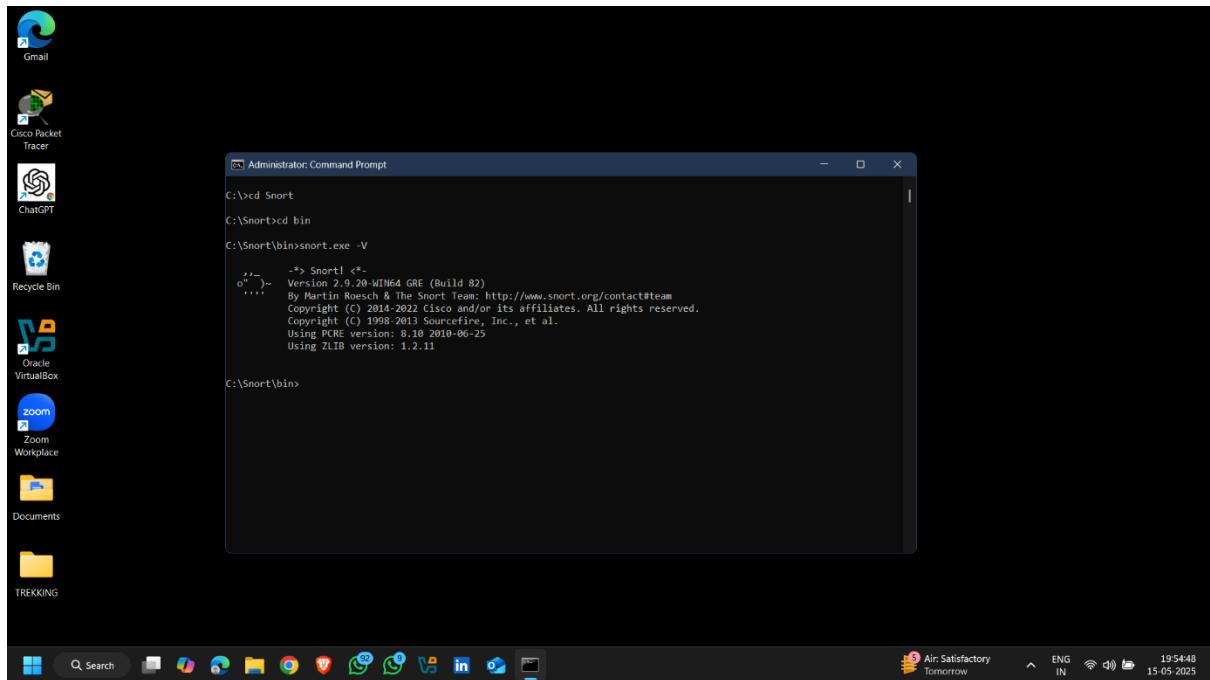


- Go to the Snort\bin Folder

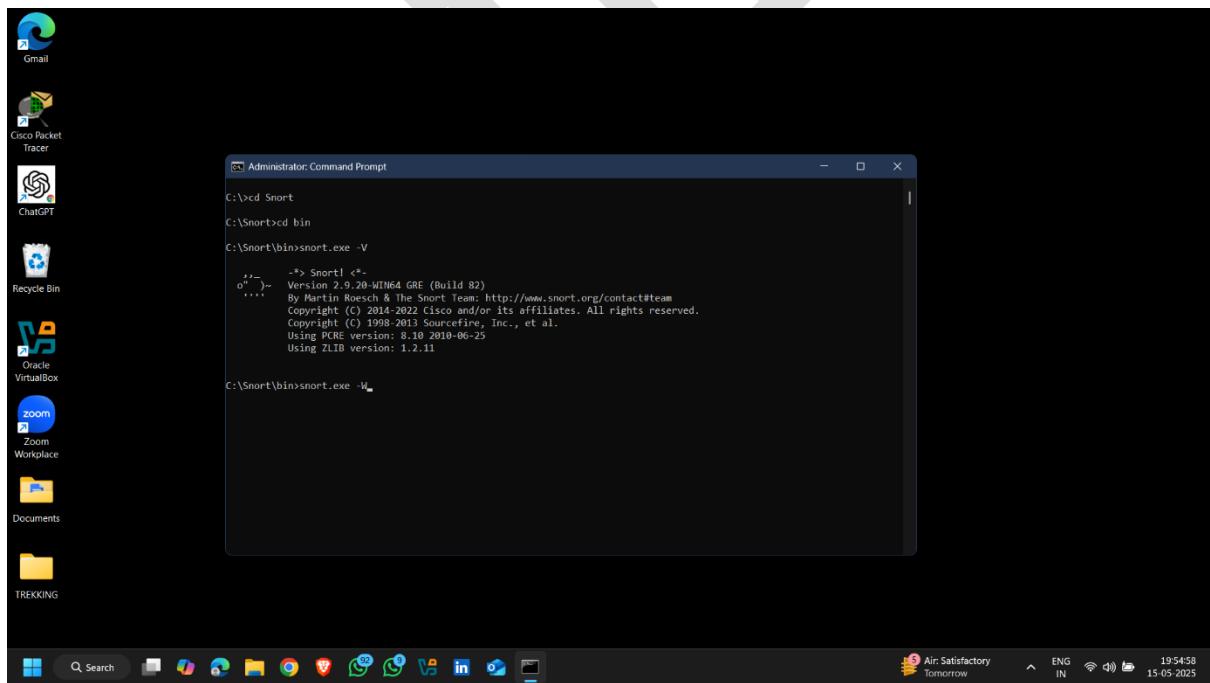


- There is file in snort bin folder >> snort.exe
- Run file snort.exe -v ( v for version)





- use next command for finding network interface
- snort.exe -W



- network interface 4

```

Select Administrator: Command Prompt
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 PCRE version: 8.32 (2010-06-25)
Using ZLIB version: 1.2.11

Index Physical Address IP Address Device Name Description
---- -----
1 00:00:00:00:00:00 disabled \Device\NPF_{8E900EE0-1030-4B1F-BCC3-0008E7C790F} WAN Miniport (Netw
ork Monitor)
2 00:00:00:00:00:00 disabled \Device\NPF_{043518B1-3002-4351-BA2C-78C7C00A5205} WAN Miniport (IP
v6)
3 00:00:00:00:00:00 disabled \Device\NPF_{BC9CEDAD-8867-4368-BD76-87EAAA54C0E1} WAN Miniport (IP
)
4 2C:38:70:9C:E4:A7 192.168.251.254 \Device\NPF_{27C3CS44-5E52-4658-9046-45345C808AED} Realtek RTL8822C
E 882.11ac PCIe Adapter
5 00:50:56:C0:00:08 192.168.217.1 \Device\NPF_{14BF4BED-489E-447E-90F8-04040DC6AE6B} VMware Virtual E
thernet Adapter for VMnet8
6 00:50:56:C0:00:01 192.168.170.1 \Device\NPF_{15A77215-1A55-4EDC-91ED-381687B19CC} VMware Virtual E
thernet Adapter for VMnet1
7 AE:31:30:9C:E4:A7 169.254.11.206 \Device\NPF_{1C5DD980-E020-4E12-99AF-5B5AF430AD08} Microsoft Wi-Fi
Direct Virtual Adapter #2
8 2E:38:70:9C:E4:A7 169.254.228.90 \Device\NPF_{A6203F57-C183-4B28-88A0-7B5B63A02024} Microsoft Wi-Fi
Direct Virtual Adapter
9 0A:00:27:00:00:04 192.168.56.1 \Device\NPF_{080CCE99-719E-4B95-85E4-3727F00EAD55} VirtualBox Host-
ffic capture
10 00:00:00:00:00:00 0000:0000:0000:0000:0000:0000 \Device\NPF_Loopback Adapter for loopback tra
C:\Snort\bin>

```

AirSatisfactory Tomorrow ENG IN 19:55:07 15-05-2025

- use command for configuration testing

**command :- snort.exe -i 4 -c "c:\Snort\etc\file name" -T**

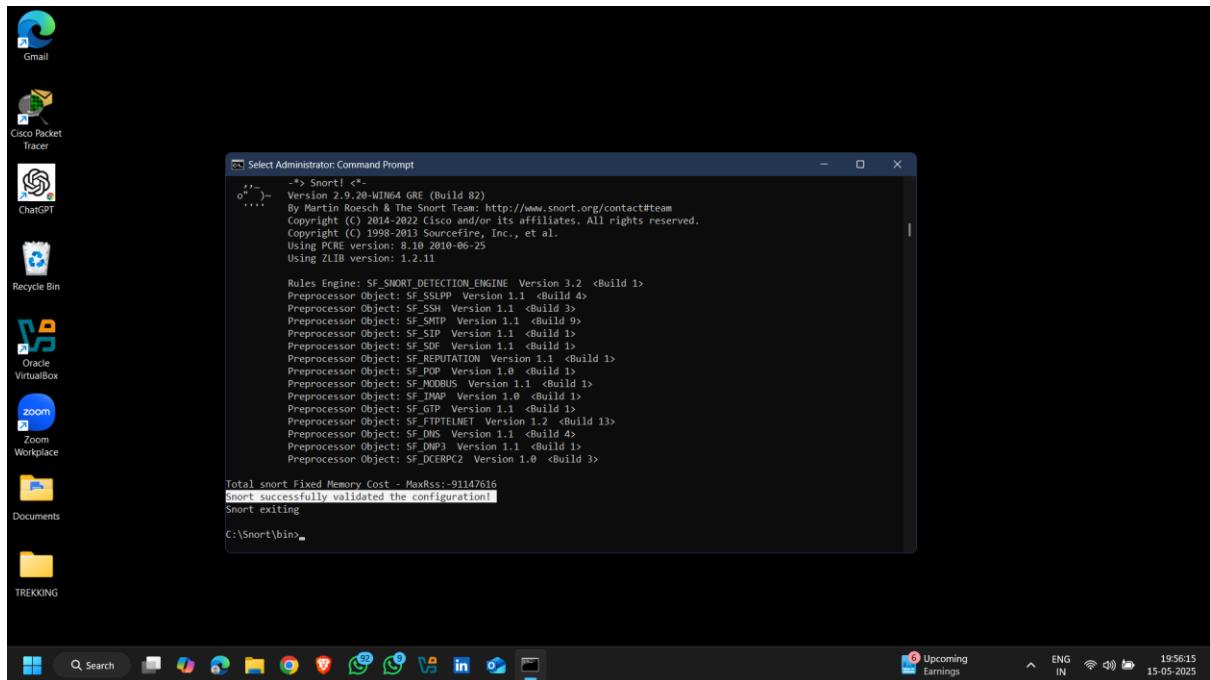
```

Administrator: Command Prompt
C:\Snort\bin>snort.exe -i 4 -c "c:\Snort\etc\snort - Copy (3).conf" -T

```

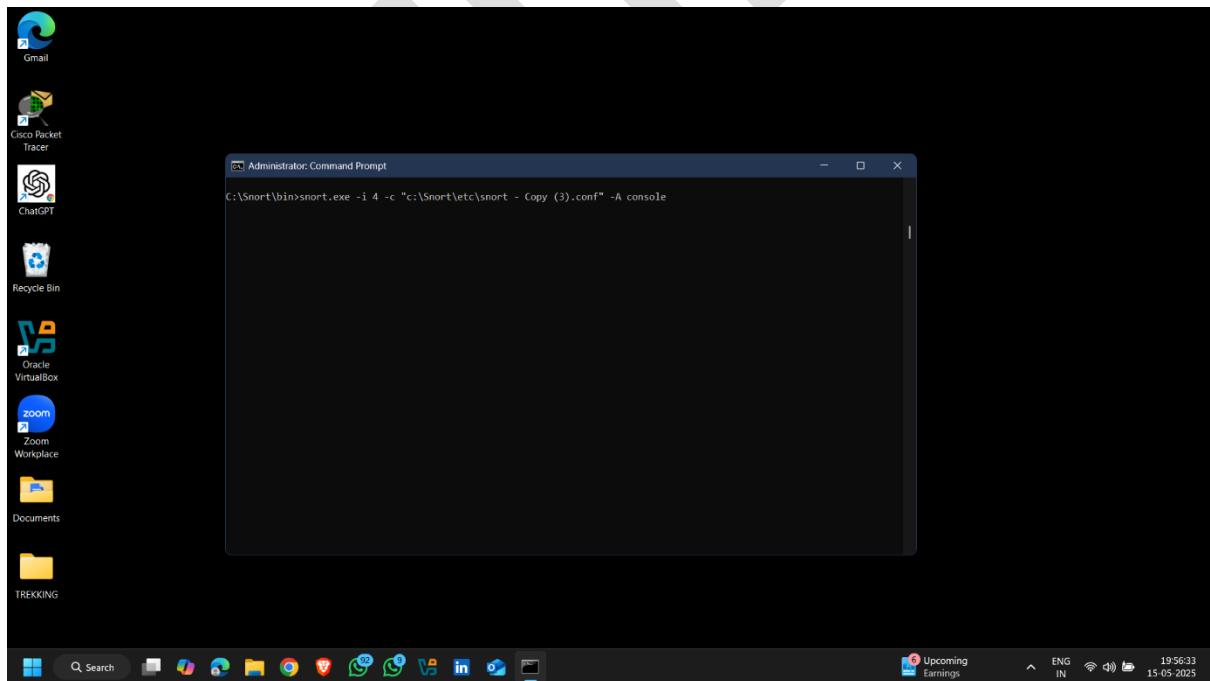
Finance headline India Car Sales R... ENG IN 19:56:03 15-05-2025

- Configuration successful validate

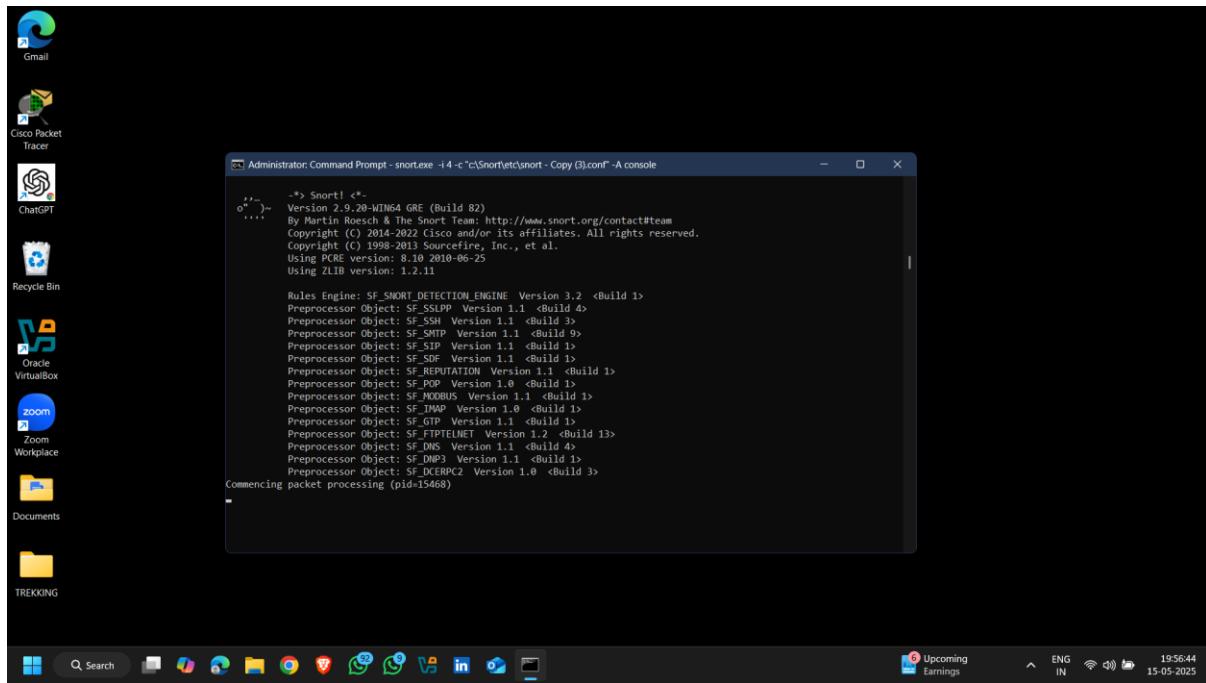


- Type command to start snort

**Command :- snort.exe -i 4 -c "c:\Snort\etc\file name" -A console**



- Here snort started 👍



- Started capturing network traffic

Administrator: Command Prompt - snort.exe -i 4 <"c:\Snort\etc\snort - Copy (3).conf">-A console

```

PCap D40 configured to passive.
The D40 version does not support reload,
Acquiring network traffic from "[DeviceNPF_{27C3C5A4-5E52-465B-90A6-45345C88BAED}]".
Decoding Ethernet
==== Initialization Complete ====
--> Snort! <-
Version 2.9.20-WIN64 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 PCRE version: 8.10 2010-06-25
Using ZLIB version: 1.2.11

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 3.2 <Build 1>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_POP Version 1.0 <Build 1>
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_DMP3 Version 1.1 <Build 1>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>

Commencing packet processing (pid=15468)
05/15-19:57:18.980537 [*] [129:12:1] Consecutive TCP small segments exceeding threshold [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 20.190.146.35:443 -> 192.168.251.254:54672
05/15-19:57:31.791631 [*] [129:15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:30590 -> 172.64.150.5:443
05/15-19:57:32.725373 [*] [129:15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:43708 -> 104.18.37.251:443
05/15-19:58:06.889625 [*] [129:15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:52874 -> 172.64.150.5:443
05/15-19:58:08.699373 [*] [129:15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:33988 -> 104.18.37.251:443
05/15-19:58:08.657150 [*] [129:15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:52891 -> 172.64.150.5:443
05/15-19:58:09.192400 [*] [129:15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:48212 -> 172.64.150.5:443
05/15-19:58:19.426275 [*] [129:15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:48226 -> 104.18.37.251:443
05/15-19:59:20.820005 [*] [129:15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.191:48238 -> 104.18.37.251:443
05/15-19:59:21.103283 [*] [129:15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:48308 -> 104.18.37.251:443
05/15-19:58:22.153845 [*] [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP server response before client request [*] [Classification: Unknown Traffic] [Priority: 3] [TCP] 163.70.144.61:80 -> 192.168.251.254:546
82
05/15-19:58:22.155516 [*] [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP server response before client request [*] [Classification: Unknown Traffic] [Priority: 3] [TCP] 163.70.144.61:80 -> 192.168.251.254:546
82
05/15-19:58:22.160656 [*] [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP server response before client request [*] [Classification: Unknown Traffic] [Priority: 3] [TCP] 163.70.144.61:80 -> 192.168.251.254:546
83
05/15-19:58:22.164293 [*] [120:18:3] (http_inspect) PROTOCOL-OTHER HTTP server response before client request [*] [Classification: Unknown Traffic] [Priority: 3] [TCP] 163.70.144.61:80 -> 192.168.251.254:546
83

```

# Windows Firewall Configuration

**Windows Firewall** (now known as **Windows Defender Firewall**) is a **built-in security feature** in Microsoft Windows operating systems that helps **protect your computer** by filtering **incoming and outgoing** network traffic based on security rules.

---

## Purpose of Windows Firewall:

To prevent **unauthorized access** to or from a private network — acting as a **barrier between your PC and external threats**.

---

## Key Features:

### 1. Inbound & Outbound Filtering:

- Blocks or allows traffic **based on rules**.
- Inbound = traffic coming into your computer.
- Outbound = traffic leaving your computer.

### 2. Predefined Security Rules:

- Automatically configures rules for common applications and system services.

### 3. Application Control:

- Prompts when an **unknown app** tries to access the network.

### 4. Network Profiles:

- Customize rules for:
  - **Private** networks (home/trusted).
  - **Public** networks (unsafe, like public Wi-Fi).

- **Domain** networks (used in corporate environments).

## 5. Integration with Windows Security Center:

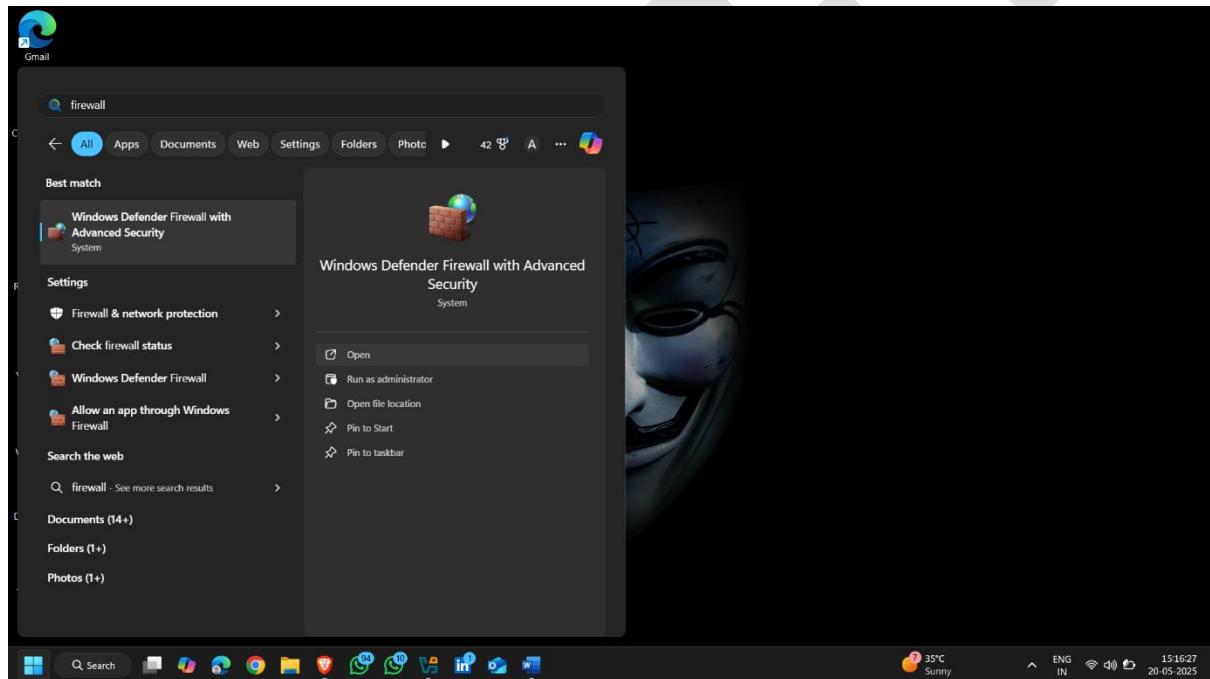
- Easily managed via **Control Panel** or **Windows Security Settings**.

## 6. Logging and Monitoring:

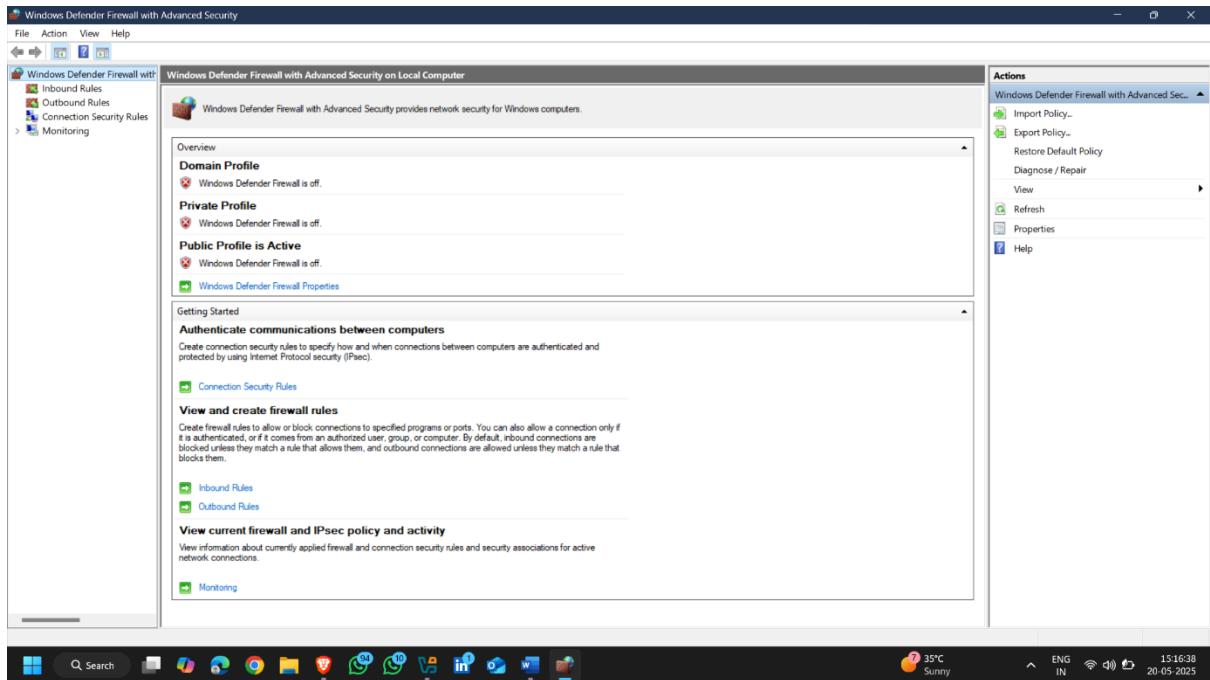
- Tracks dropped packets and connection attempts.

### How to do it :- (INBOUND RULES)

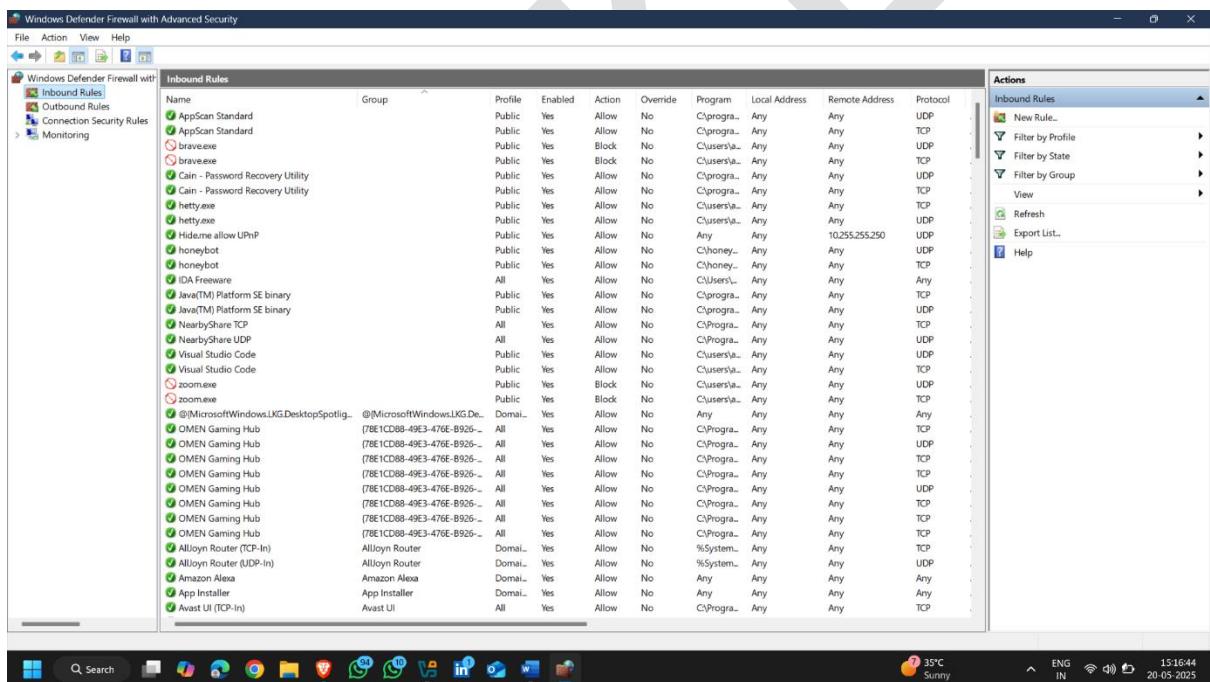
- Click on Windows Button and search Firewall and open it .



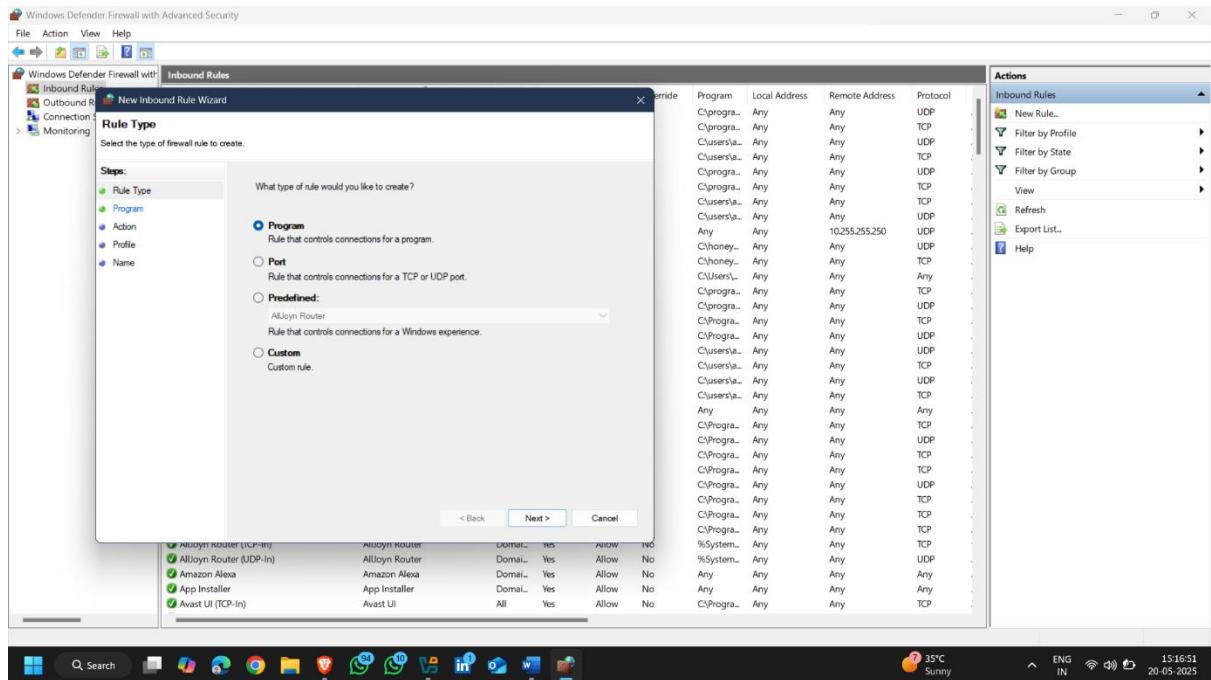
- New Window Open , there is option **inbound rules** – it means set rules for incoming network



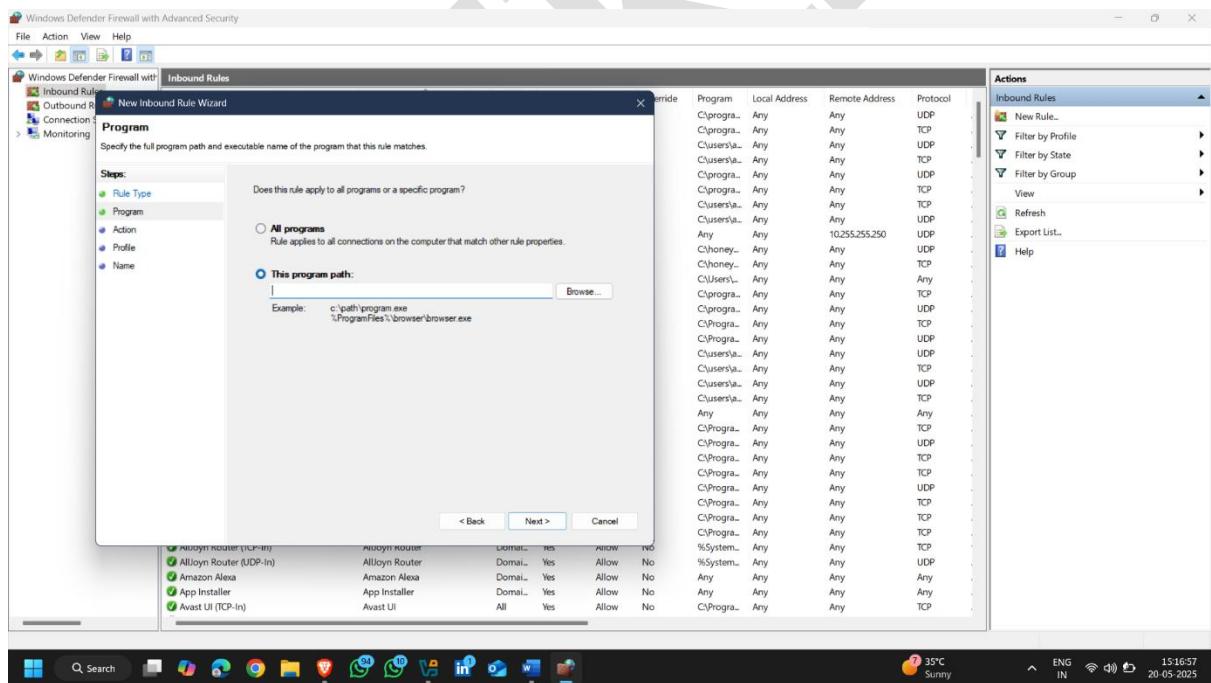
**Click on Inbound rules then click on new rule**



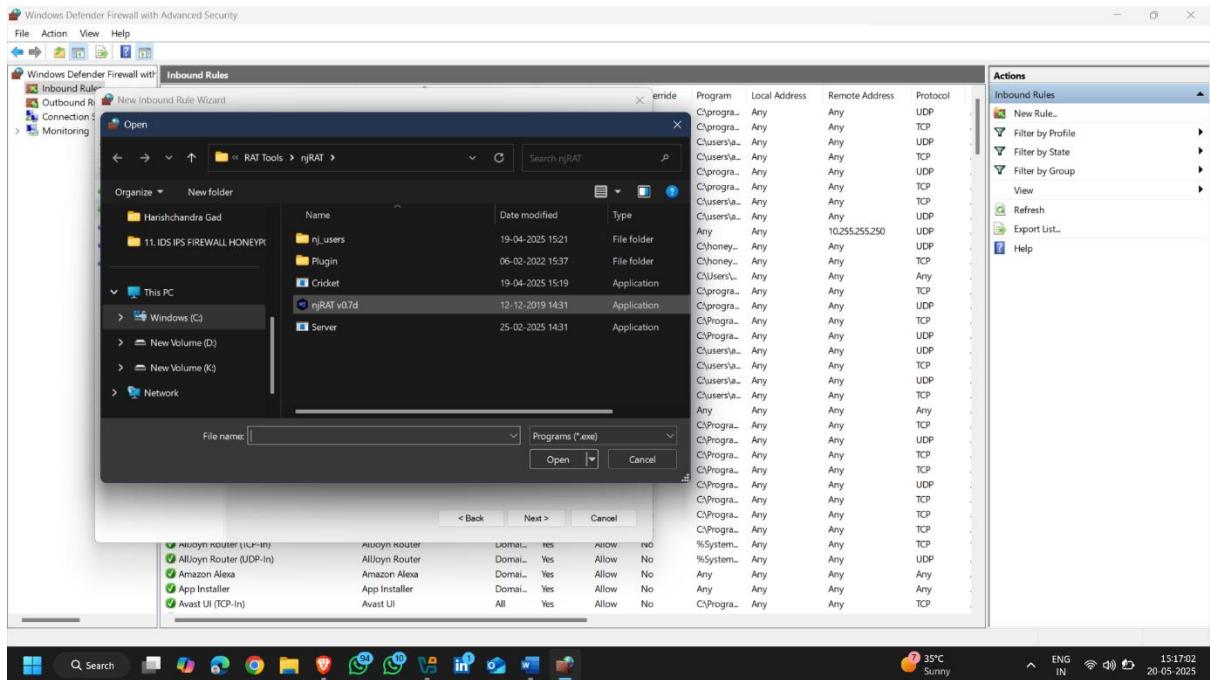
- Next pop up appears , there is many options ,then select the option that you want to set configuration like particular port or program ,this kind of stuff and then click on next



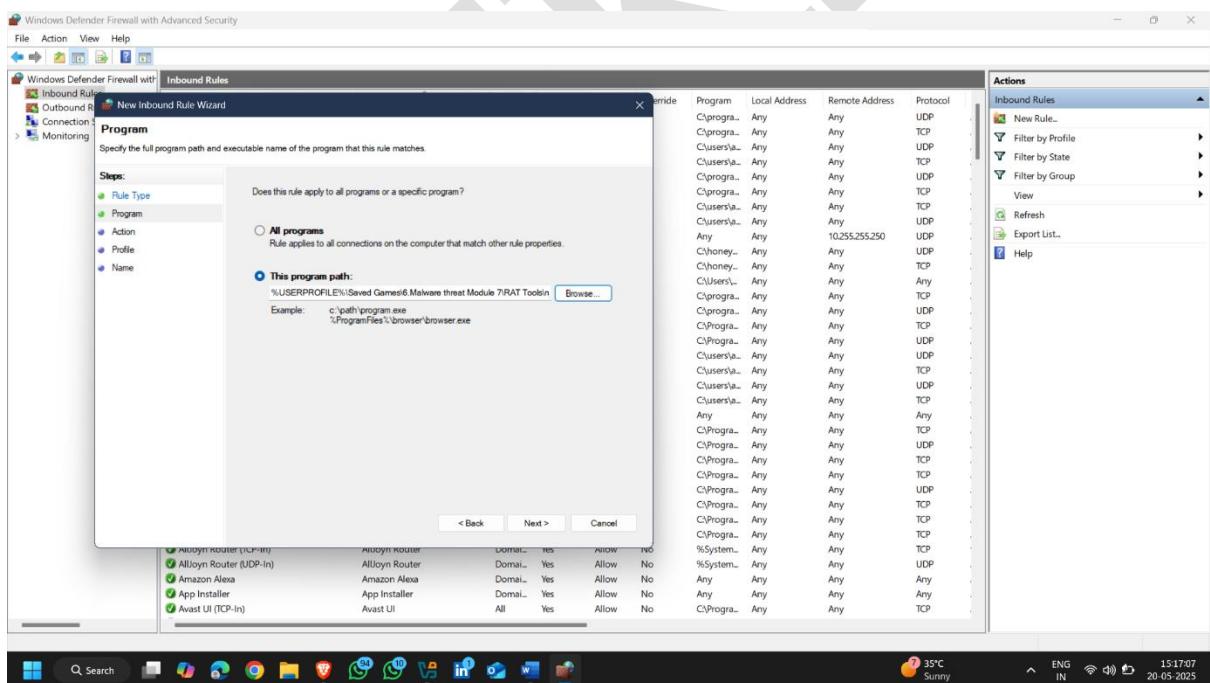
- Then set a path of program



- Select a file and click on open



- Click on next



**Now there is a three option :-**

- 1. allow the connection.
- 2. allow the connection if it is secure .

- 3. Block the connection .

## 1. Allow the connection

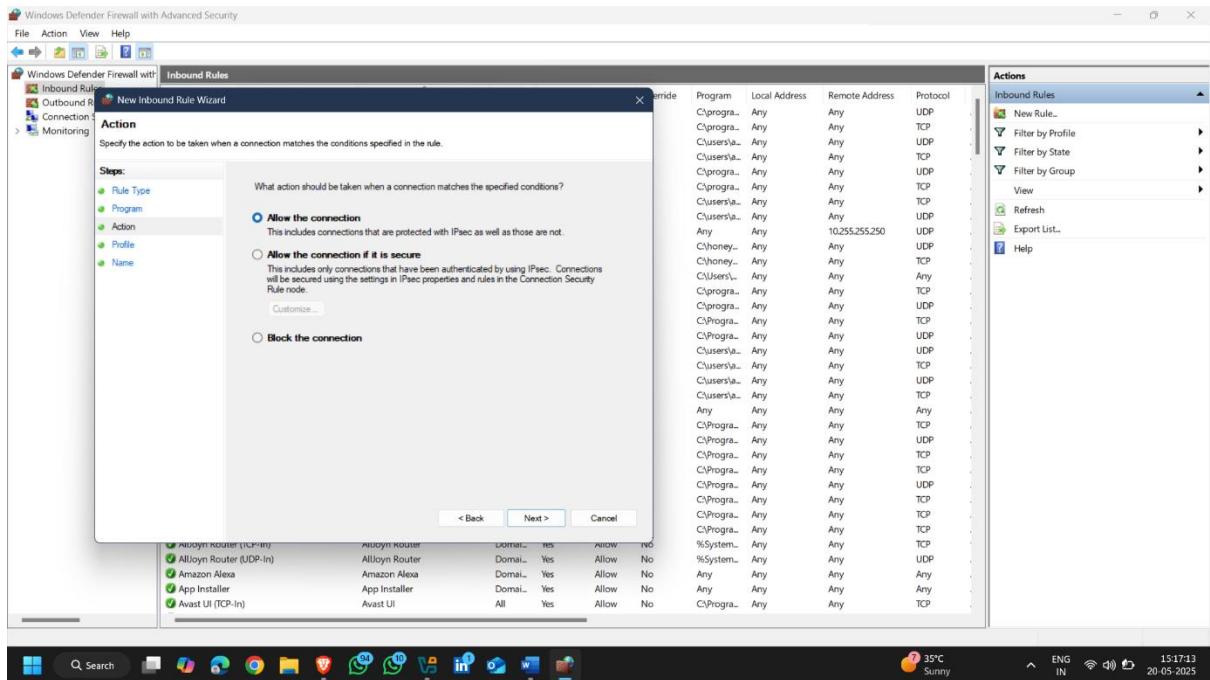
- **Description:** This allows all matching incoming traffic, whether or not it is secured.
  - **Security:** Not restricted to authenticated or encrypted connections.
  - **Use When:** You trust the traffic or program and want to permit communication without extra security restrictions.
- 

## 2. Allow the connection if it is secure

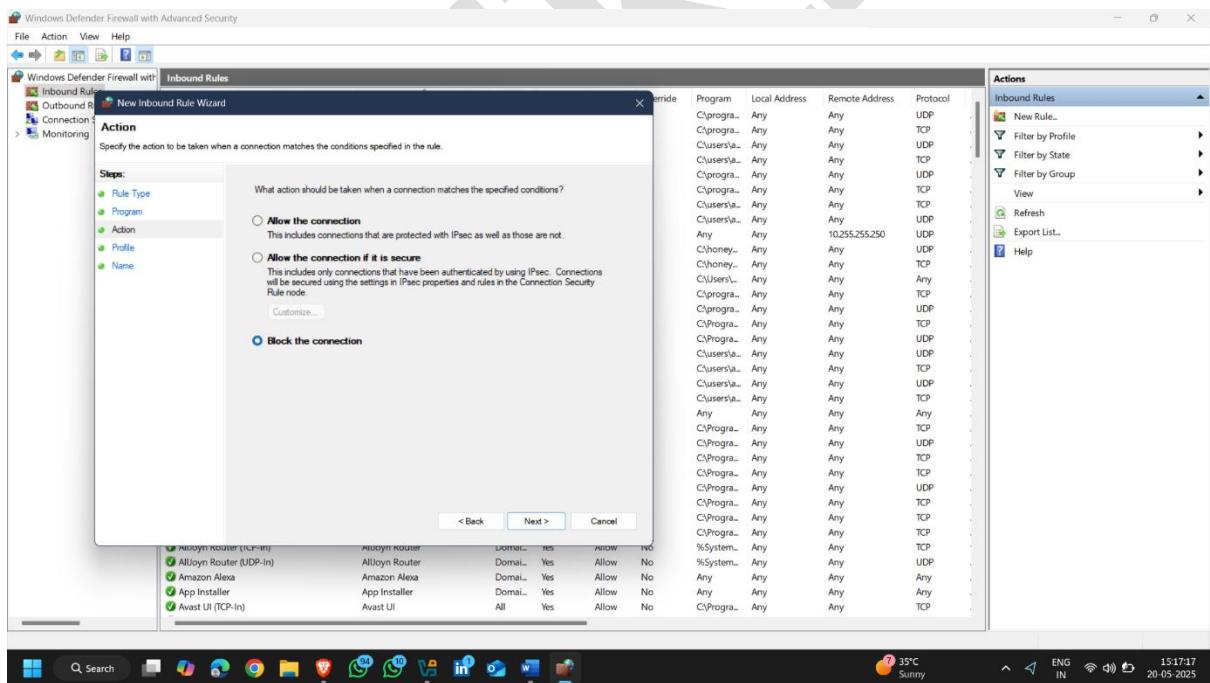
- **Description:** Only allows connections authenticated using IPsec (Internet Protocol Security).
  - **Requires:** Proper configuration of IPsec policies.
  - **Use When:** You want to only permit secure (encrypted and authenticated) connections for sensitive systems or data transfer.
  - **Customize button** is usually greyed out unless IPsec is configured.
- 

## 3. Block the connection

- **Description:** Denies the connection completely, regardless of whether it's secure or not.
  - **Security:** Strictest setting; completely blocks the matched traffic.
  - **Use When:** You want to prevent any communication through the specified ports, programs, or IP addresses.
-



- After select option then click on next



**Again Three options appear**

**Domain**

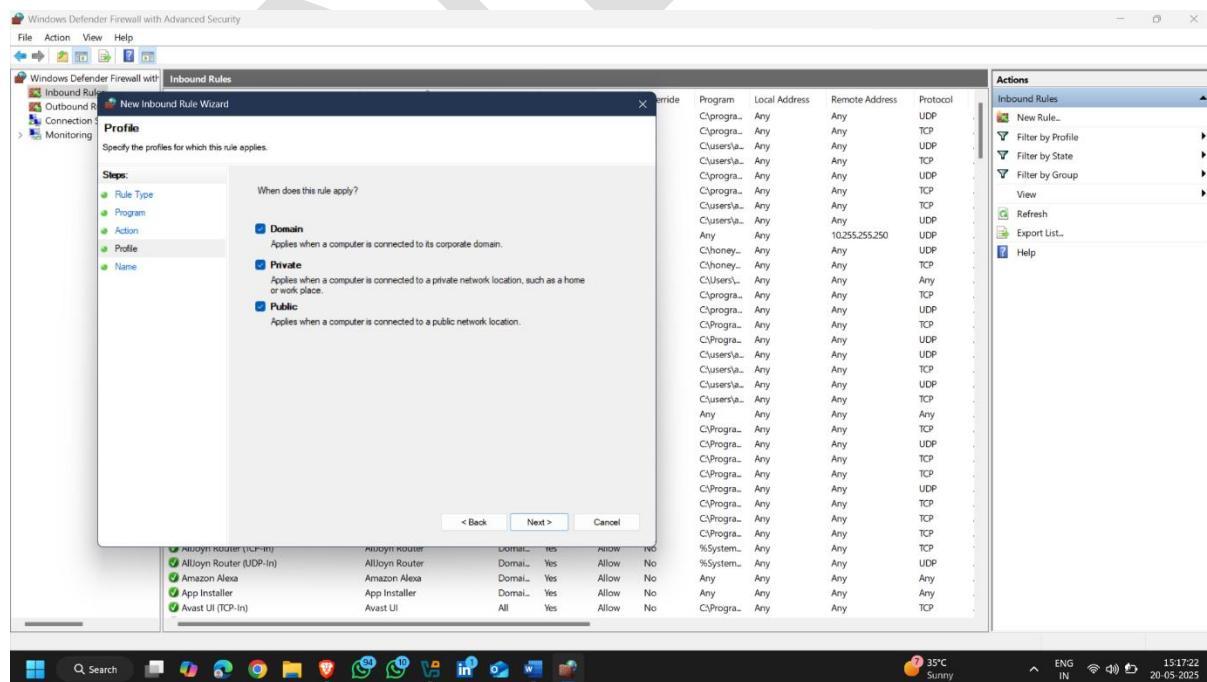
- **Applies When:** The system is connected to a corporate network joined to a domain (e.g., office or enterprise environment with Active Directory).
- **Use Case:** Apply this rule when you want it to work only in secure, managed networks.

## Private

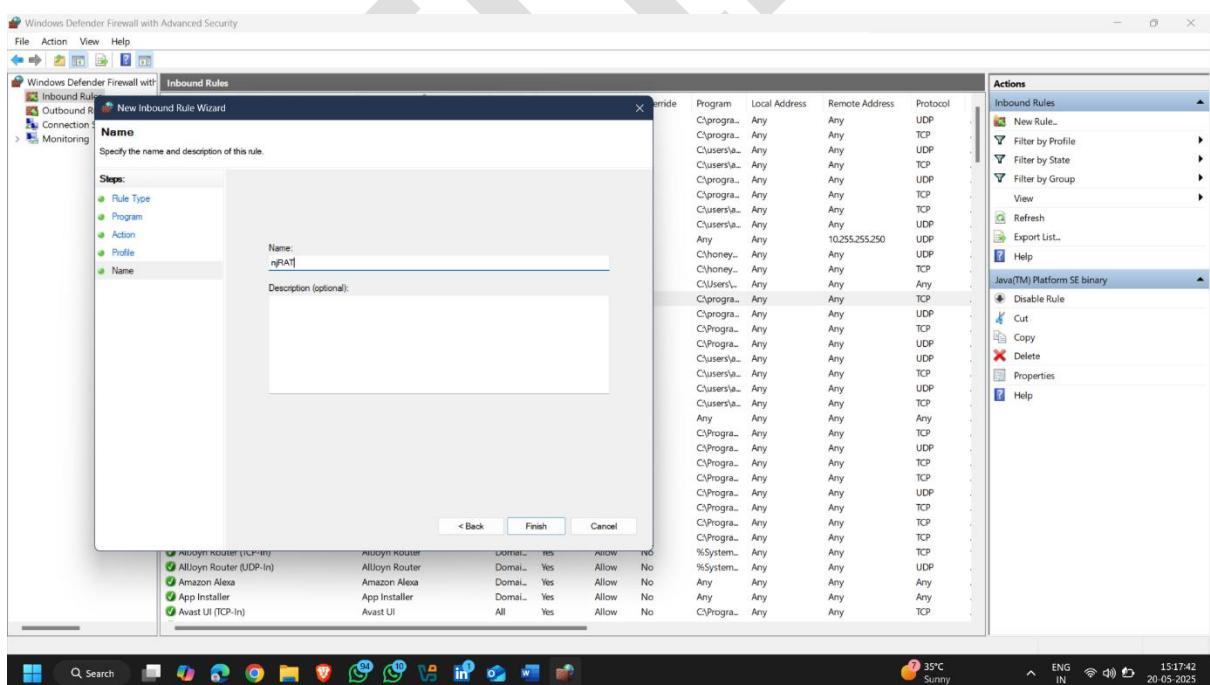
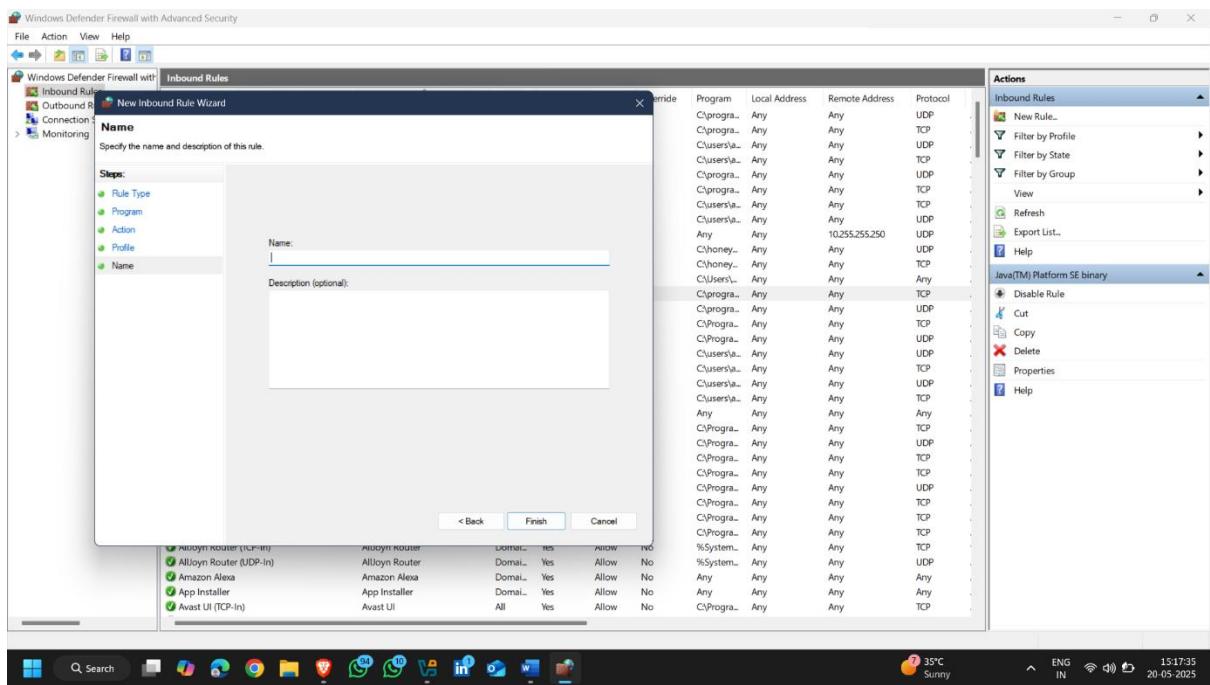
- **Applies When:** The computer is connected to a trusted private network, like your home Wi-Fi or a small office network.
- **Use Case:** Enable the rule when you're at home or in a network you trust.

## Public

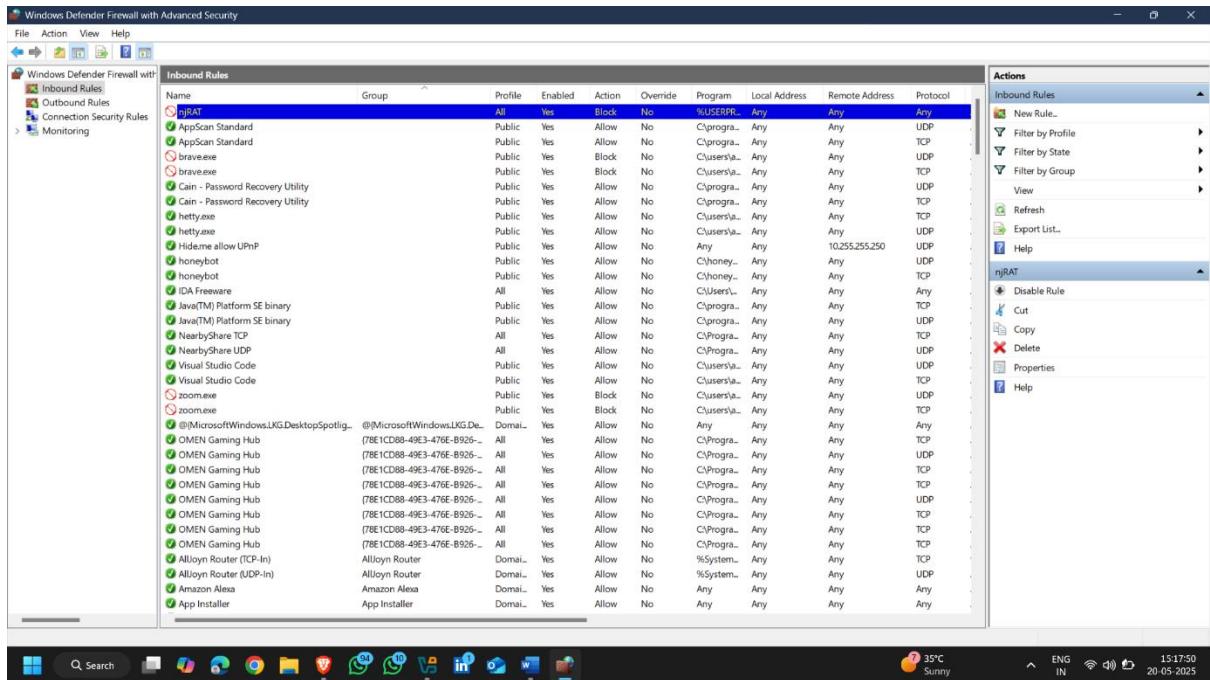
- **Applies When:** The system is connected to a public network, like at a café, airport, or hotel.
- **Use Case:** Apply the rule in public places. Be cautious here—public networks are less secure.



- Set a any name and click on finish

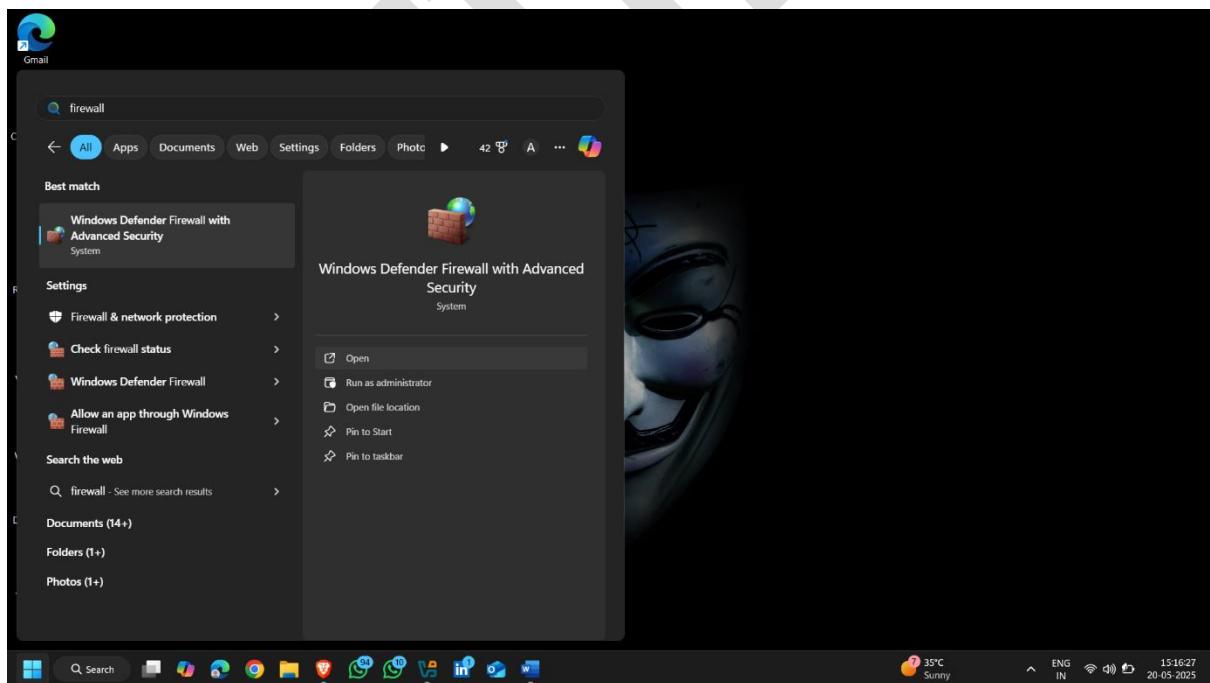


- Configuration set for inbound rules

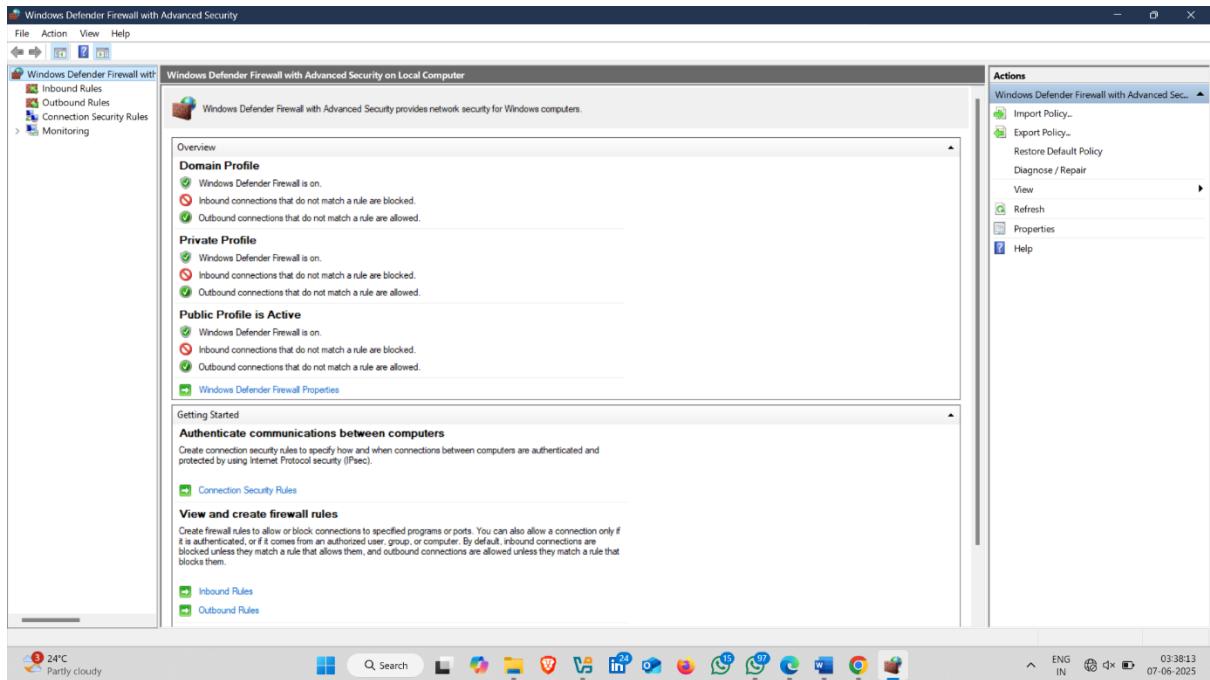


## How to do it :- (OUTBOUND RULES) :-

- Click on Windows Button and search Firewall and open it



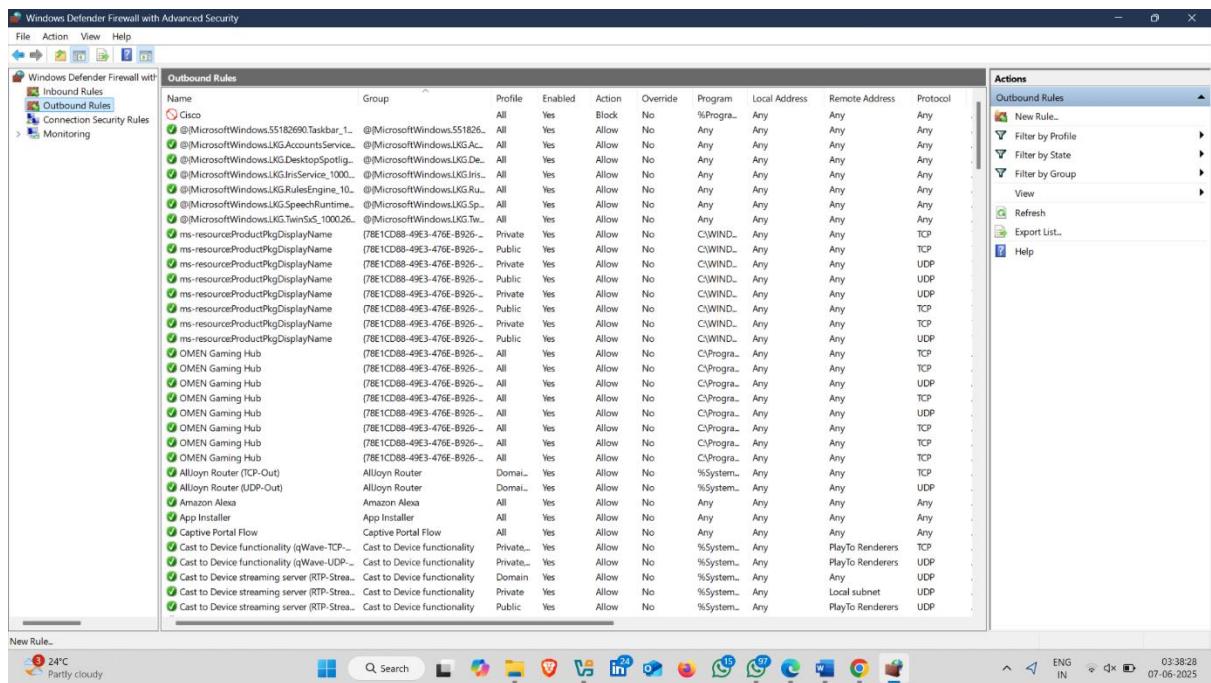
- New Window Open , there is option **outbound rules** – it means set rules for outgoing network ,



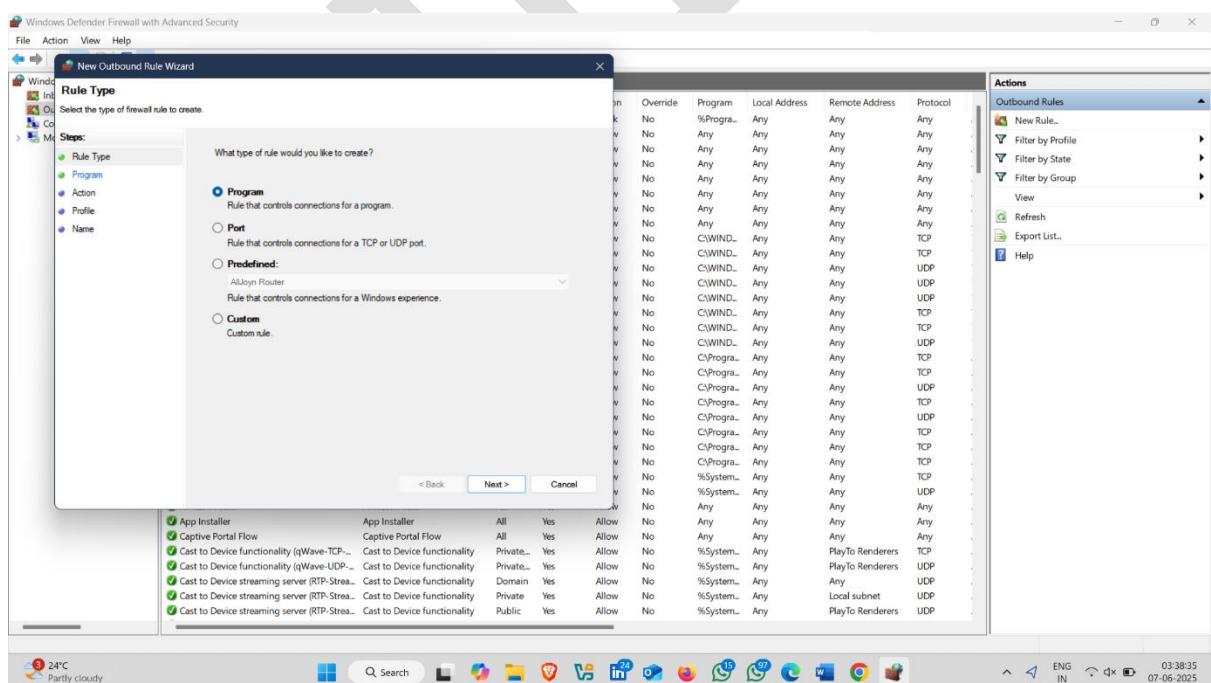
- Click on Outbound rules

Name	Group	Profile	Enabled	Action	Override	Program	Local Address	Remote Address	Protocol
☐ Cisco	@MicrosoftWindows.55182690.Taskbar_1...	All	Yes	Block	No	%Program%	Any	Any	Any
☐ @MicrosoftWindows.LKG.AccountsService...	@MicrosoftWindows.LKG.Ac...	All	Yes	Allow	No	Any	Any	Any	Any
☐ @MicrosoftWindows.LKG.DesktopSprintig...	@MicrosoftWindows.LKG.De...	All	Yes	Allow	No	Any	Any	Any	Any
☐ @MicrosoftWindows.LKG.InsService_1000...	@MicrosoftWindows.LKG.Ins...	All	Yes	Allow	No	Any	Any	Any	Any
☐ @MicrosoftWindows.LKG.Ruleskringne_10...	@MicrosoftWindows.LKG.Rul...	All	Yes	Allow	No	Any	Any	Any	Any
☐ @MicrosoftWindows.LKG.TwinSS_100026...	@MicrosoftWindows.LKG.Twi...	All	Yes	Allow	No	Any	Any	Any	Any
☐ ms-resourceProductPkgDisplayName	(78E1CDB8-49E3-476E-B926-...	Private	Yes	Allow	No	CWIND...	Any	Any	TCP
☐ ms-resourceProductPkgDisplayName	(78E1CDB8-49E3-476E-B926-...	Public	Yes	Allow	No	CWIND...	Any	Any	TCP
☐ ms-resourceProductPkgDisplayName	(78E1CDB8-49E3-476E-B926-...	Private	Yes	Allow	No	CWIND...	Any	Any	UDP
☐ ms-resourceProductPkgDisplayName	(78E1CDB8-49E3-476E-B926-...	Public	Yes	Allow	No	CWIND...	Any	Any	UDP
☐ ms-resourceProductPkgDisplayName	(78E1CDB8-49E3-476E-B926-...	Private	Yes	Allow	No	CWIND...	Any	Any	UDP
☐ ms-resourceProductPkgDisplayName	(78E1CDB8-49E3-476E-B926-...	Public	Yes	Allow	No	CWIND...	Any	Any	TCP
☐ ms-resourceProductPkgDisplayName	(78E1CDB8-49E3-476E-B926-...	Private	Yes	Allow	No	CWIND...	Any	Any	TCP
☐ OMEN Gaming Hub	(78E1CDB8-49E3-476E-B926-...	All	Yes	Allow	No	CProg...	Any	Any	TCP
☐ OMEN Gaming Hub	(78E1CDB8-49E3-476E-B926-...	All	Yes	Allow	No	CProg...	Any	Any	TCP
☐ OMEN Gaming Hub	(78E1CDB8-49E3-476E-B926-...	All	Yes	Allow	No	CProg...	Any	Any	UDP
☐ OMEN Gaming Hub	(78E1CDB8-49E3-476E-B926-...	All	Yes	Allow	No	CProg...	Any	Any	TCP
☐ OMEN Gaming Hub	(78E1CDB8-49E3-476E-B926-...	All	Yes	Allow	No	CProg...	Any	Any	UDP
☐ OMEN Gaming Hub	(78E1CDB8-49E3-476E-B926-...	All	Yes	Allow	No	CProg...	Any	Any	TCP
☐ OMEN Gaming Hub	(78E1CDB8-49E3-476E-B926-...	All	Yes	Allow	No	CProg...	Any	Any	UDP
☐ OMEN Gaming Hub	(78E1CDB8-49E3-476E-B926-...	All	Yes	Allow	No	CProg...	Any	Any	TCP
☐ OMEN Gaming Hub	(78E1CDB8-49E3-476E-B926-...	All	Yes	Allow	No	CProg...	Any	Any	TCP
☐ AllJoyn Router (TCP-Out)	AllJoyn Router	Domai...	Yes	Allow	No	%System...	Any	Any	TCP
☐ AllJoyn Router (UDP-Out)	AllJoyn Router	Domai...	Yes	Allow	No	%System...	Any	Any	UDP
☐ Amazon Alexa	Amazon Alexa	All	Yes	Allow	No	Any	Any	Any	Any
☐ App Installer	App Installer	All	Yes	Allow	No	Any	Any	Any	Any
☐ Captive Portal Flow	Captive Portal Flow	All	Yes	Allow	No	Any	Any	Any	Any
☐ Cast to Device functionality (qWave-TCP-...	Cast to Device functionality	Private...	Yes	Allow	No	%System...	Any	PlayTo Renderers	TCP
☐ Cast to Device functionality (qWave-UDP-...	Cast to Device functionality	Private...	Yes	Allow	No	%System...	Any	PlayTo Renderers	UDP
☐ Cast to Device streaming server (RTP-Strea...	Cast to Device functionality	Domain	Yes	Allow	No	%System...	Any	Any	UDP
☐ Cast to Device streaming server (RTP-Strea...	Cast to Device functionality	Private	Yes	Allow	No	%System...	Any	Local subnet	UDP
☐ Cast to Device streaming server (RTP-Strea...	Cast to Device functionality	Public	Yes	Allow	No	%System...	Any	PlayTo Renderers	UDP

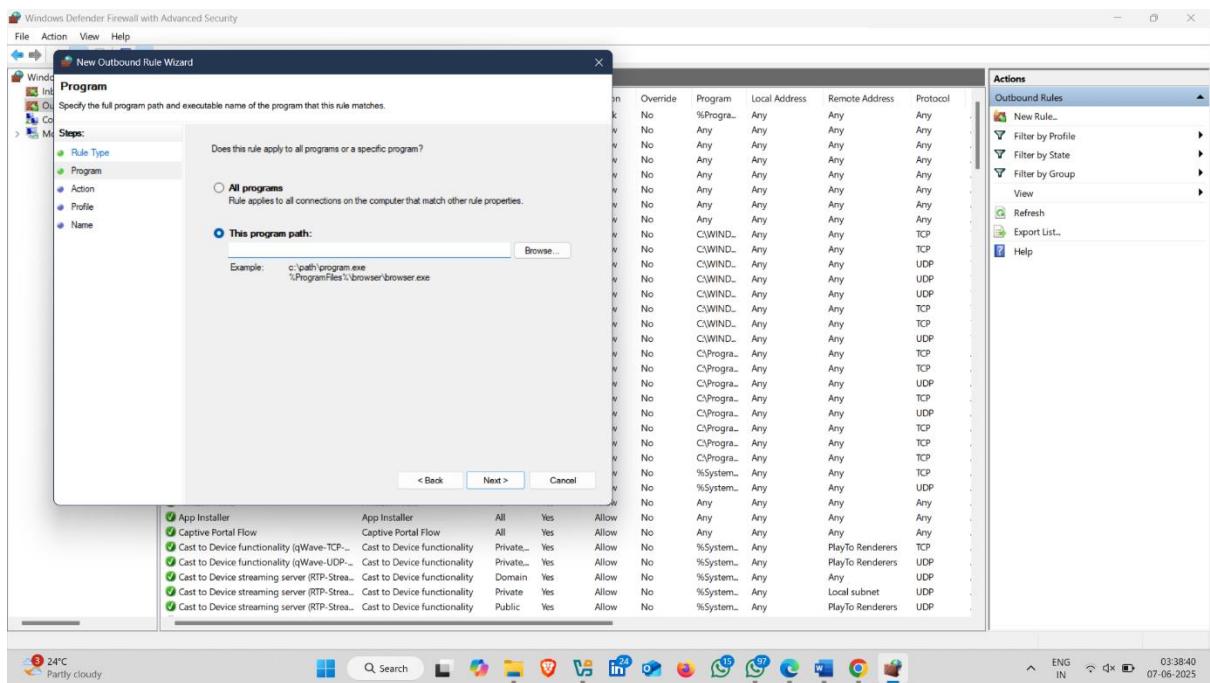
- Then click on New Rule



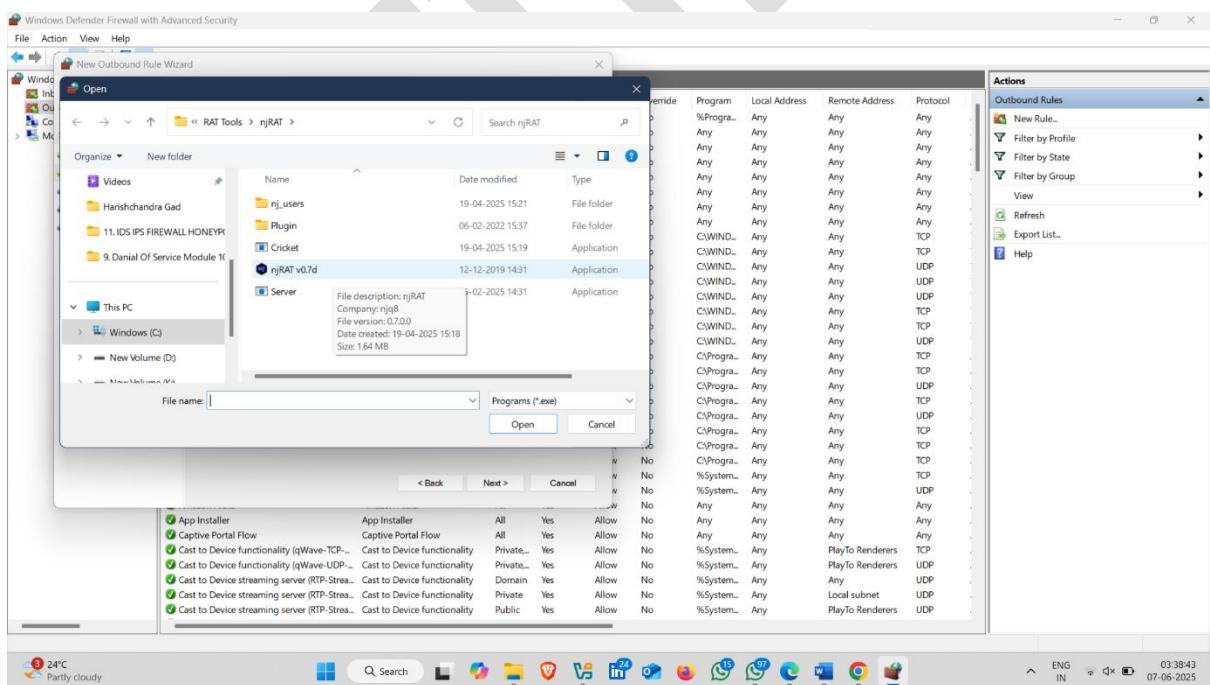
- New Pop Up appear with Three Options ,click on that you want to set a rule



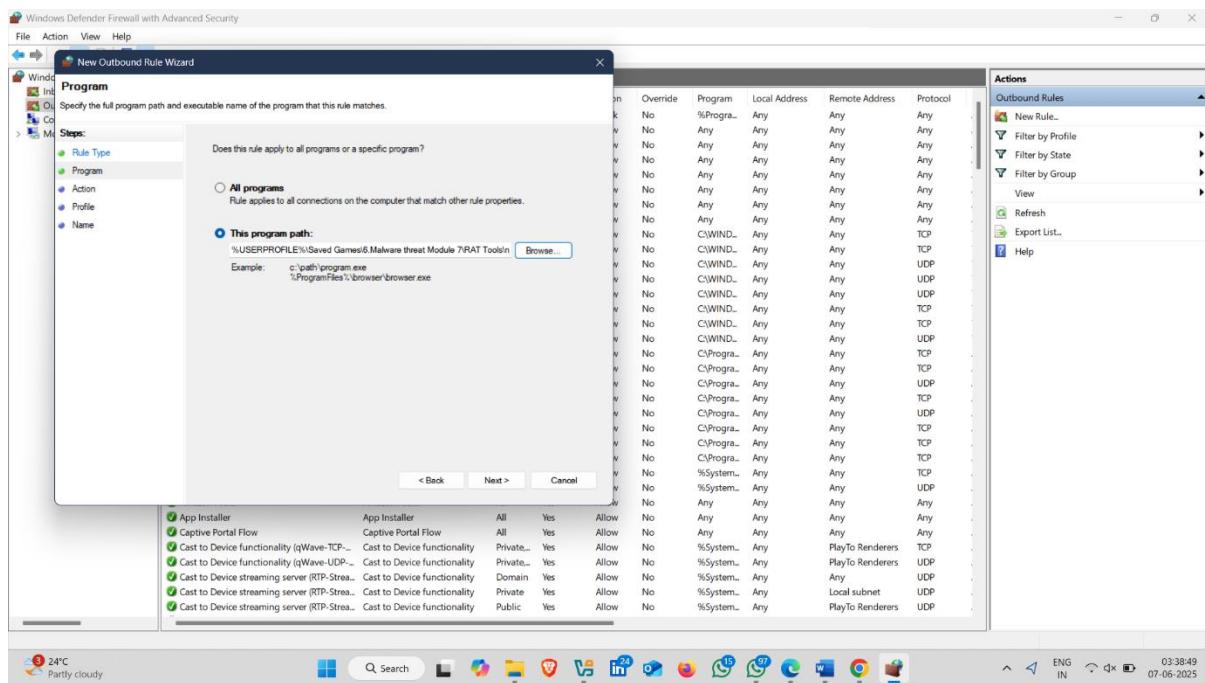
- Now , set the path by clicking on **Browse**



- Select the file and then click on open



- Click on Next



**Now there is a three option :-**

- 1. allow the connection.
- 2. allow the connection if it is secure .
- 3. Block the connection .

## 1. Allow the connection

- **✓ Description:** This allows all matching incoming traffic, whether or not it is secured.
- **🔒 Security:** Not restricted to authenticated or encrypted connections.
- **✓ Use When:** You trust the traffic or program and want to permit communication without extra security restrictions.

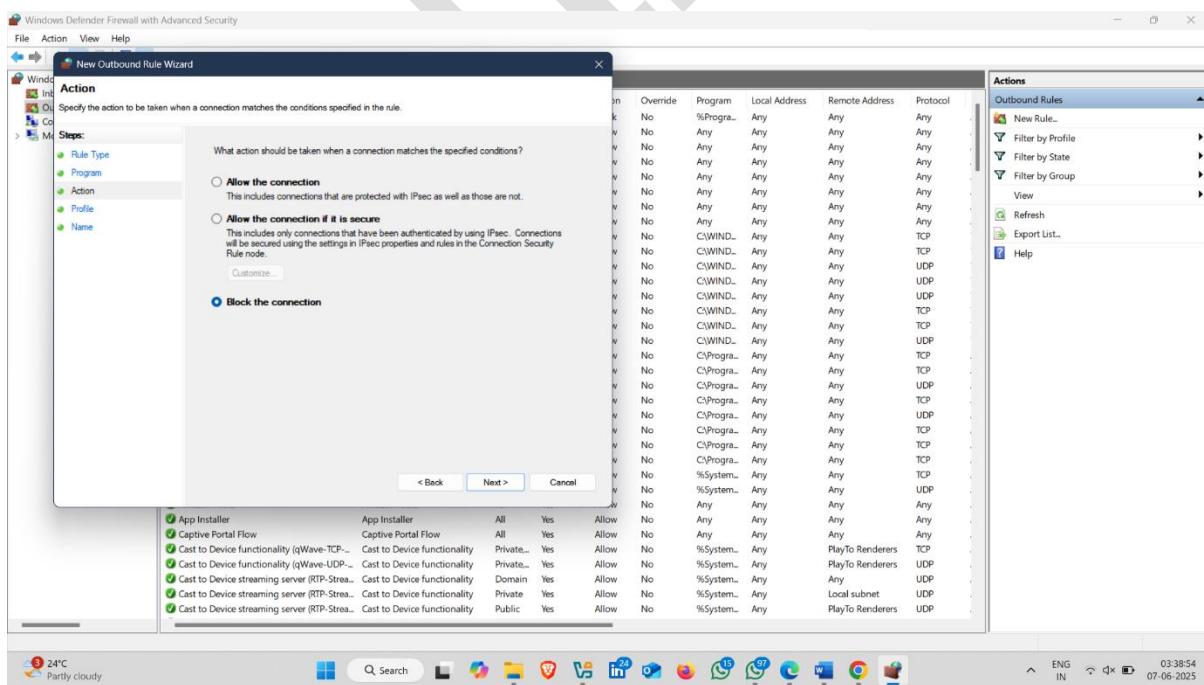
## 2. Allow the connection if it is secure

- **🔒 Description:** Only allows connections authenticated using IPsec (Internet Protocol Security).

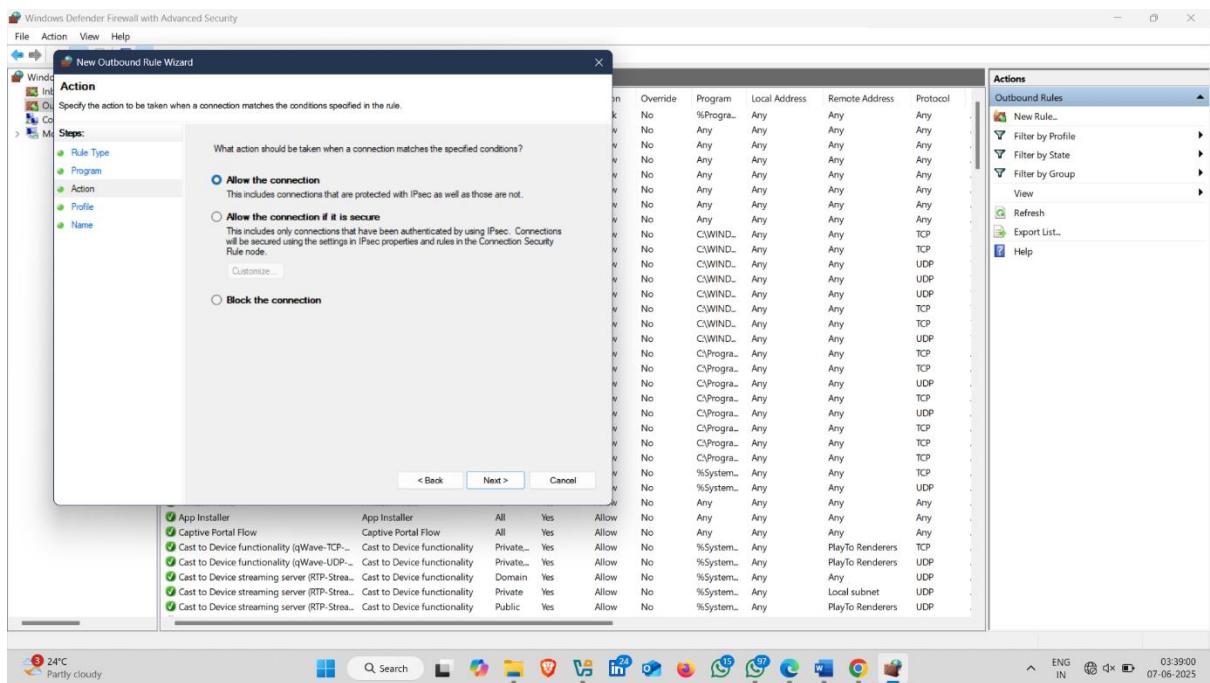
- **Requires:** Proper configuration of IPsec policies.
  - **Use When:** You want to only permit secure (encrypted and authenticated) connections for sensitive systems or data transfer.
  - **Customize button** is usually greyed out unless IPsec is configured.
- 

### 3. Block the connection

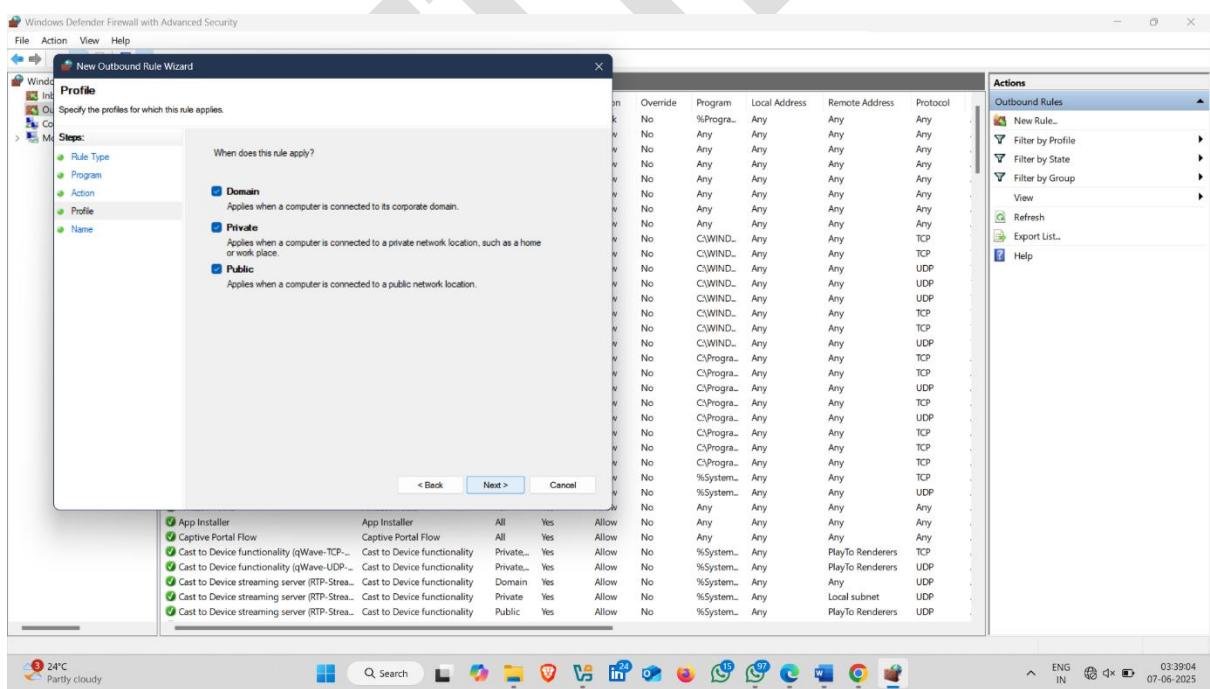
- **Description:** Denies the connection completely, regardless of whether it's secure or not.
  - **Security:** Strictest setting; completely blocks the matched traffic.
  - **Use When:** You want to prevent any communication through the specified ports, programs, or IP addresses.
- 



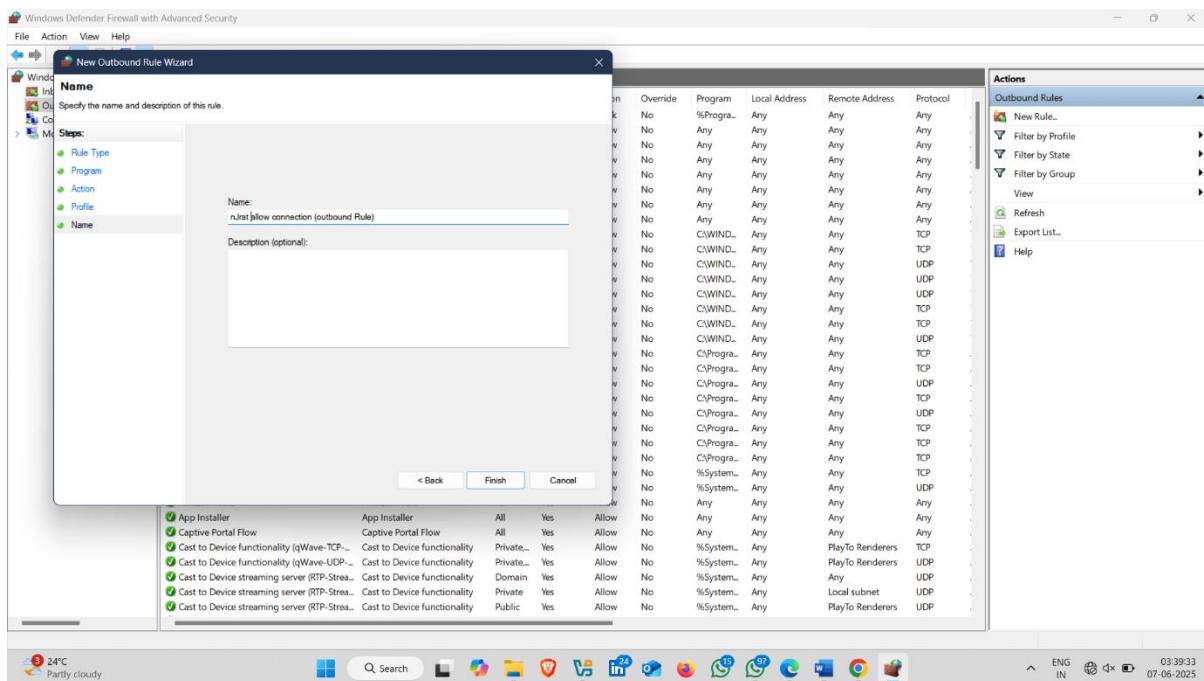
- Select option and click on next



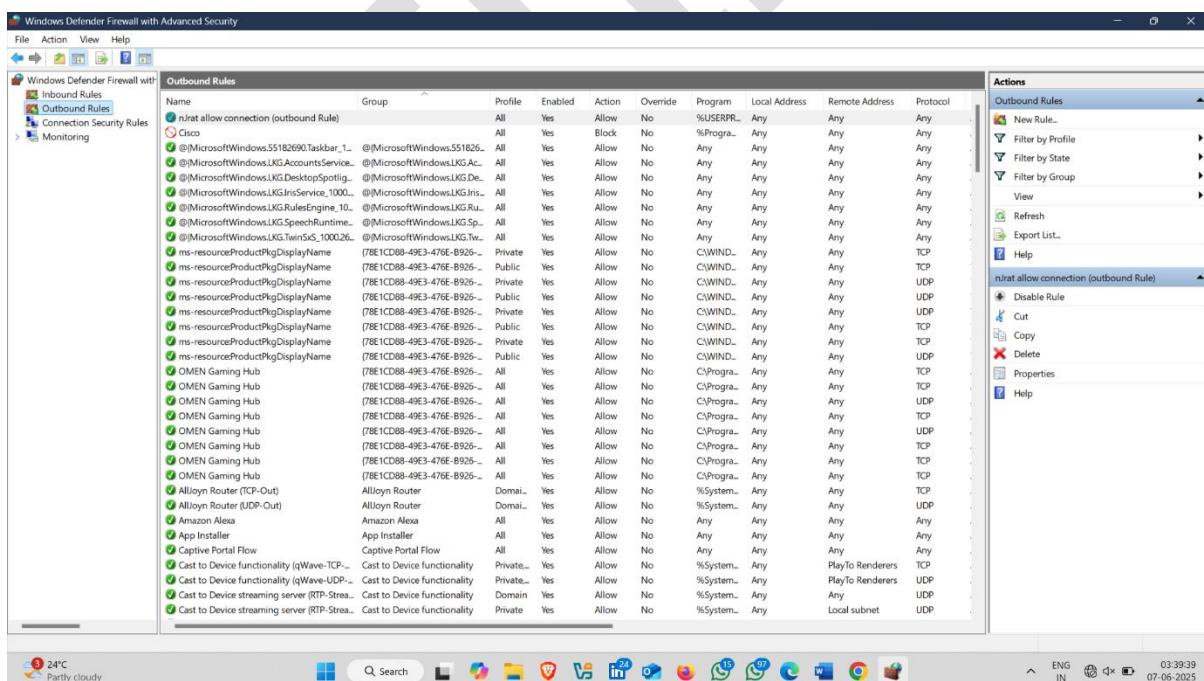
- Click on next



- Set the name of the Rule and description and click on next



- Rule set successfully ✅ 👍



# **EXTRA ACTIVITY**

## **1. Zone Alarm Firewall**

**ZoneAlarm Firewall** is a third-party **software firewall** for Windows developed by **Check Point Software Technologies**. It provides an extra layer of protection beyond the default **Windows Defender Firewall**, especially useful for users who want more detailed control over network activities.

---

### **Key Features of ZoneAlarm Firewall:**

#### **1. Two-Way Firewall:**

- **Monitors incoming and outgoing traffic.**
- Blocks unauthorized inbound and outbound connections.

#### **2. Stealth Mode:**

- Makes your PC **invisible** to hackers on the internet by hiding open ports.

#### **3. Application Control:**

- Alerts you when programs try to access the internet.
- Allows or blocks access on a per-application basis.

#### **4. Real-Time Monitoring:**

- Tracks all network activity.
- Shows what programs are using your internet.

#### **5. DefenseNet™:**

- A cloud-based feature that uses community feedback to **auto-configure** trusted apps.

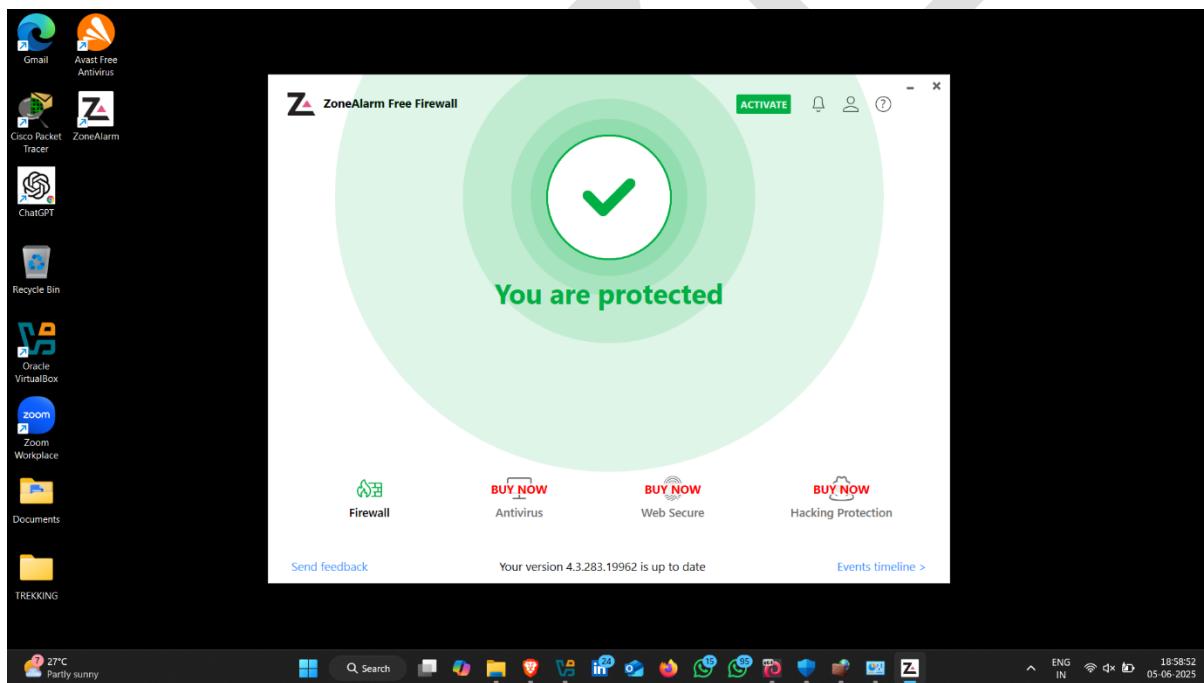
## 6. Identity Protection (Optional in Pro Version):

- Alerts you if personal data is at risk.
- Includes credit monitoring features (U.S.-based users).

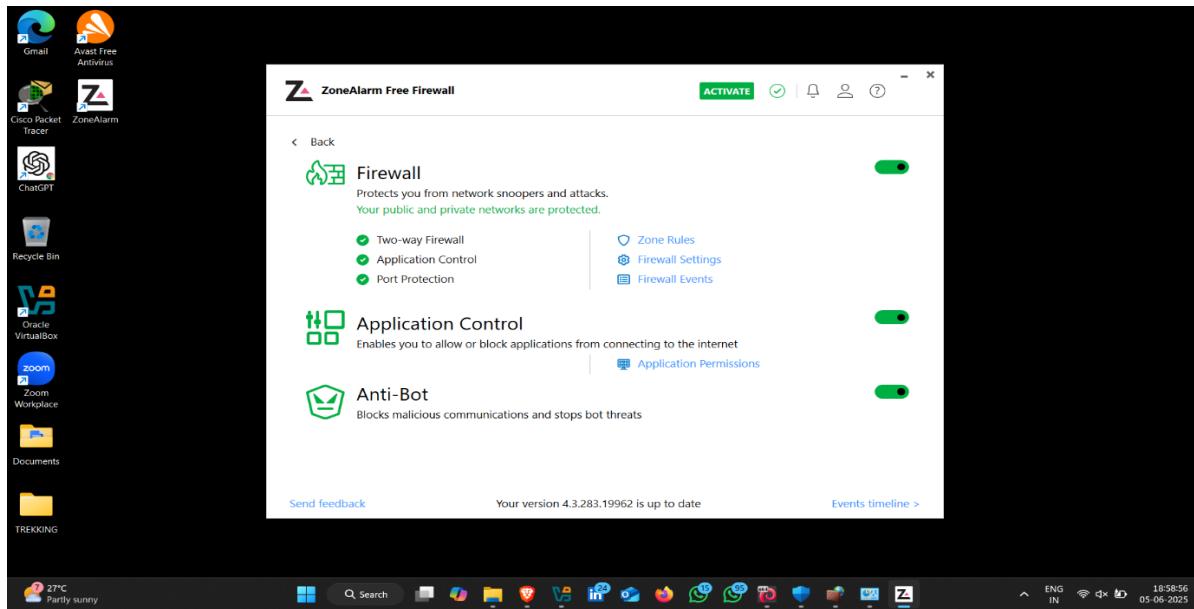
### How to use it :-

After Installation , Open the application

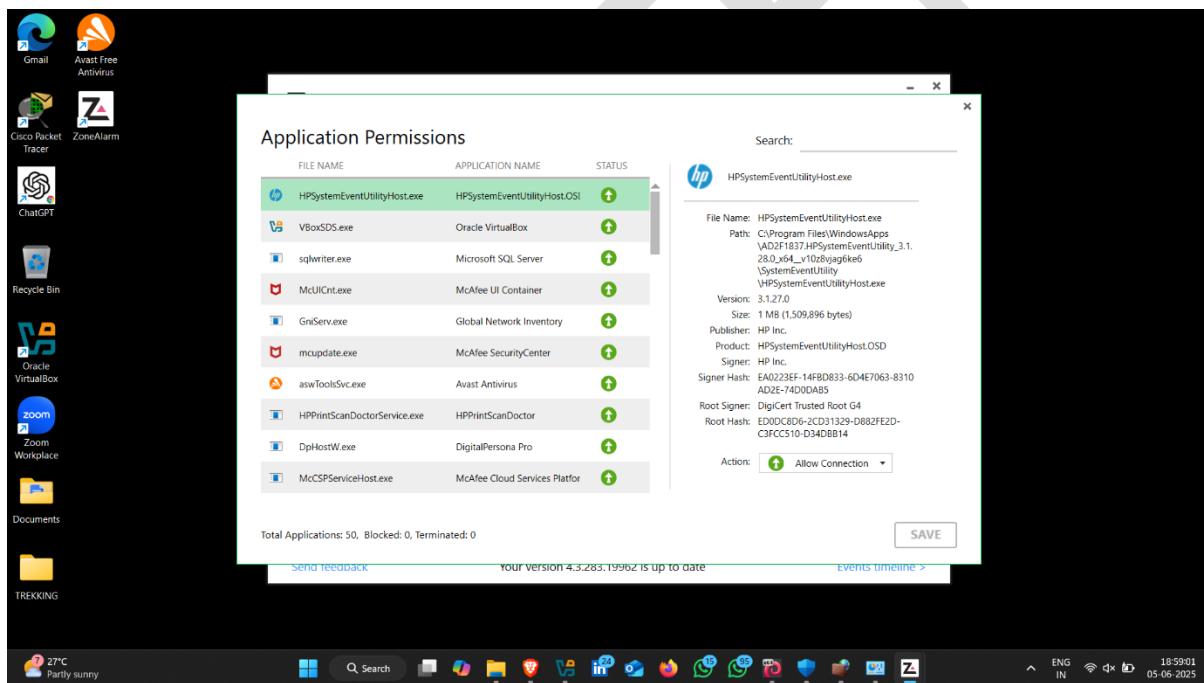
- Now click on firewall



- Click on Application Permissions



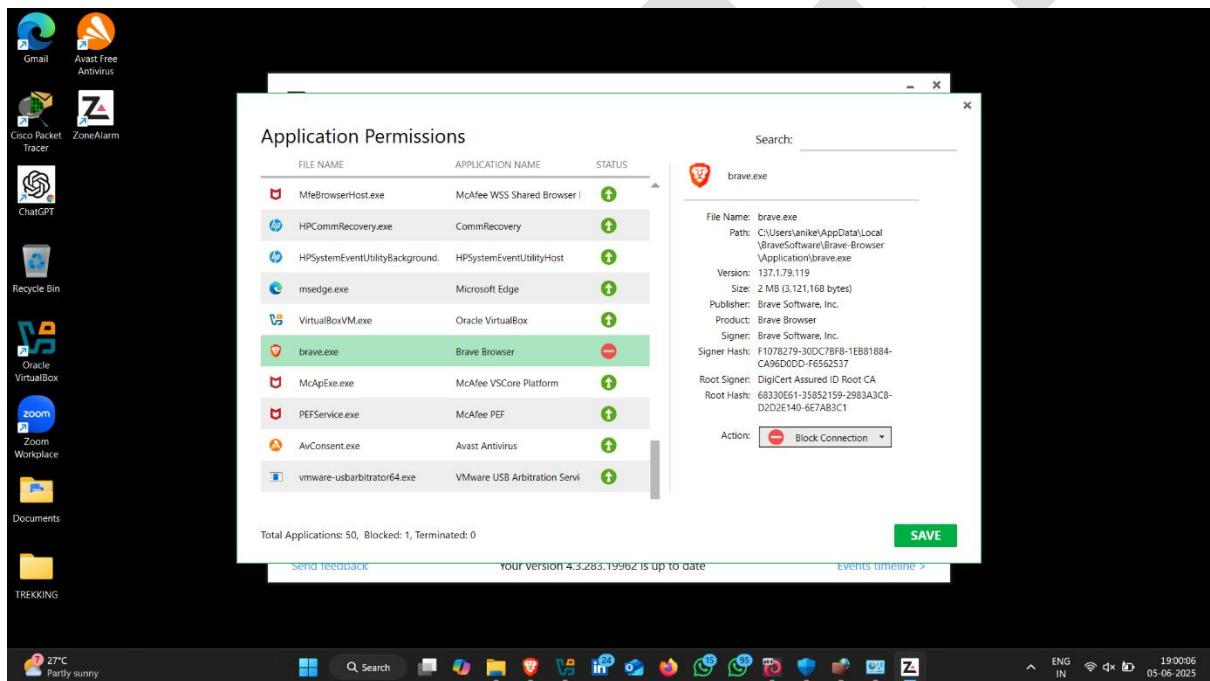
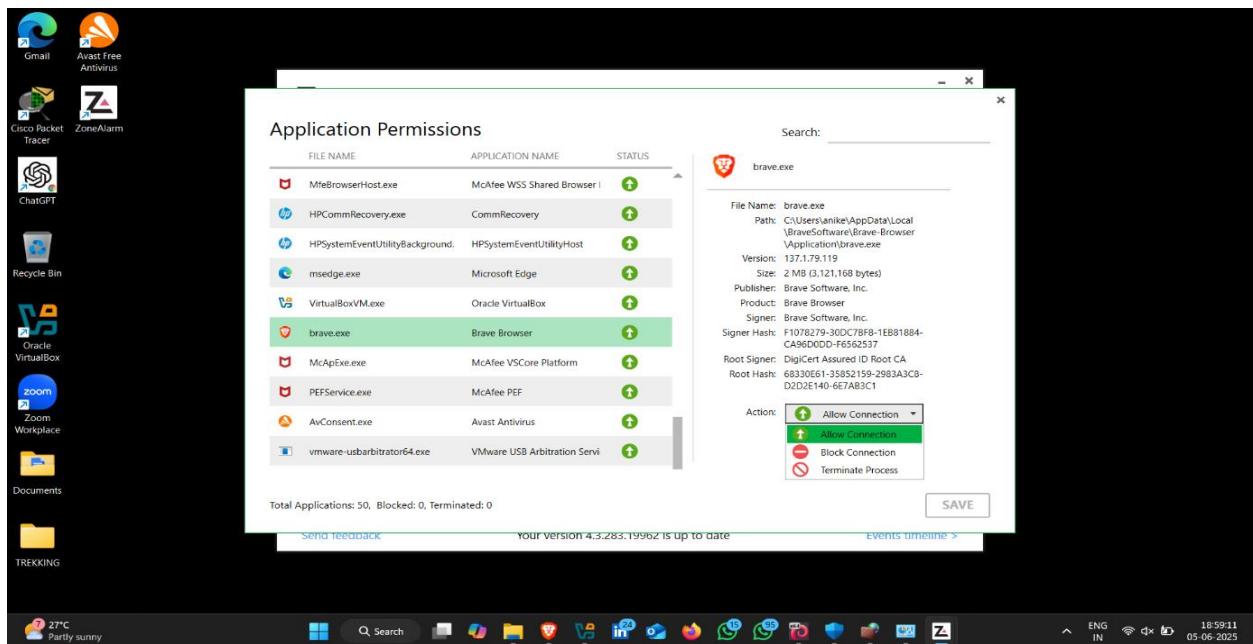
- Select the program that you want set a rule



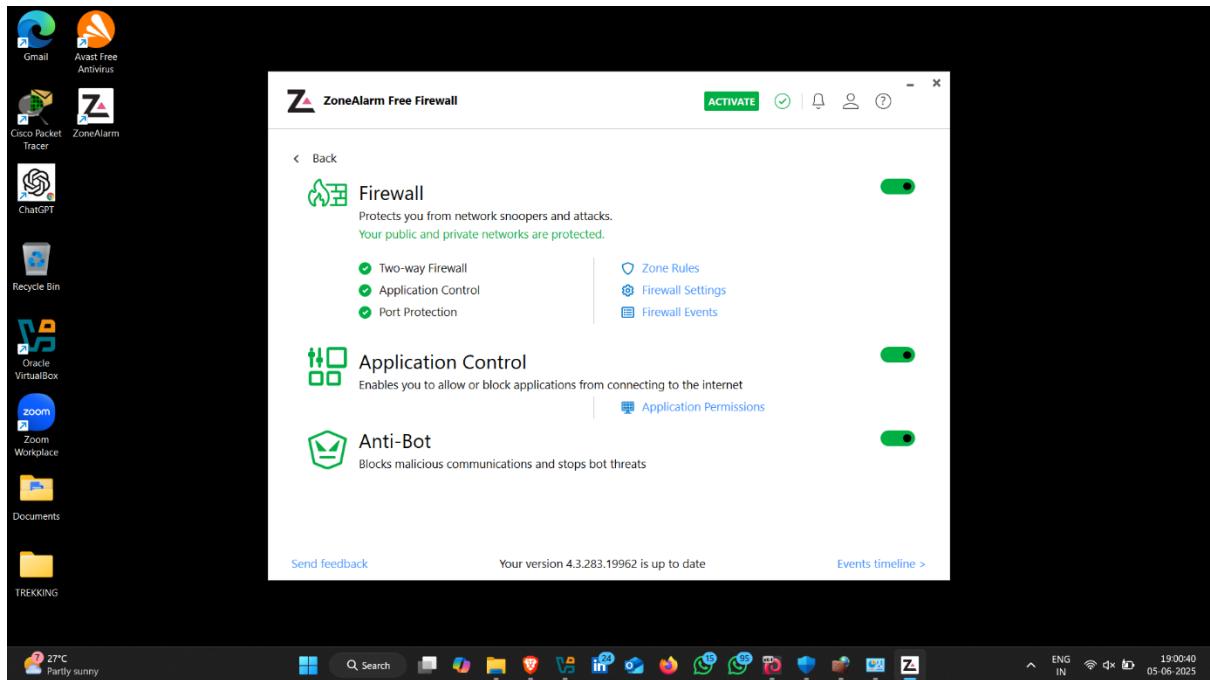
- Now , you can allow connection , Block Connection and Terminate Process

#### Action Dropdown:

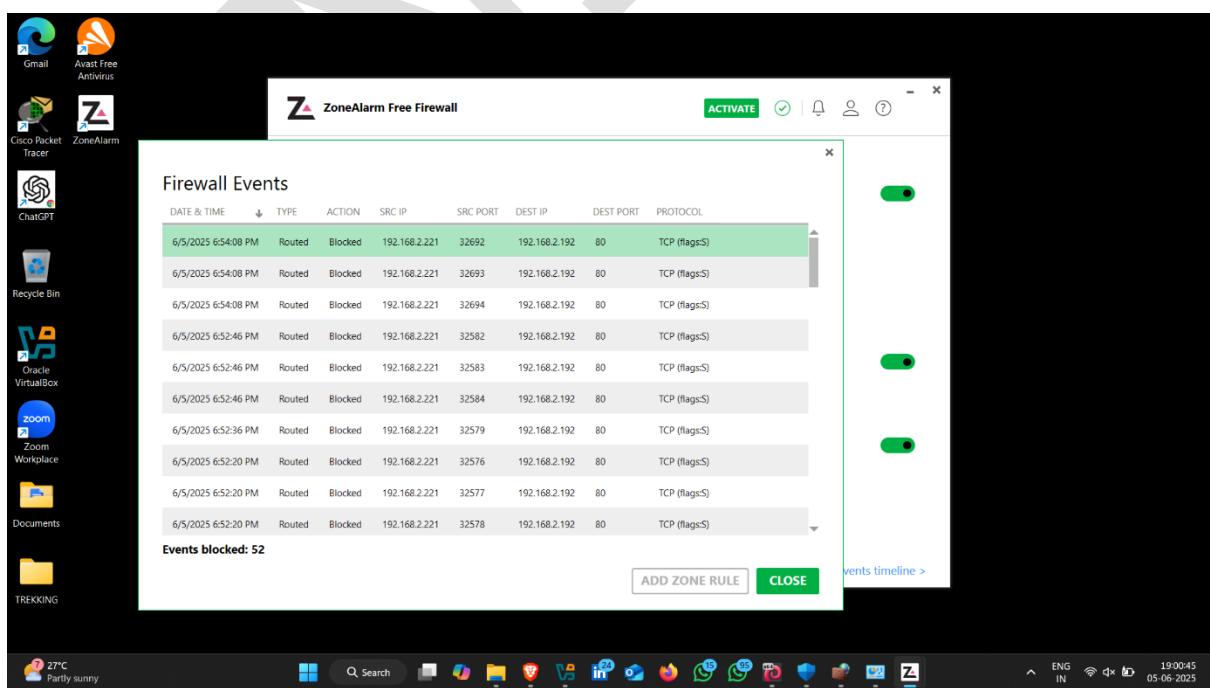
- **Allow Connection (Selected)** – The firewall allows to access the internet.
- **Block Connection** – Prevents from accessing the internet.
- **Terminate Process** – Stops the process if it is running.



- Now click on Firewall Events that show the event logs that they capture during monitoring and capturing



- It show all the event



# Attack Detection Tools

## 1. Perform Attack Detection Using HoneyBoT

**HoneyBOT** is a **Windows-based honeypot** software used for **cybersecurity monitoring** and **intrusion detection**. It simulates a vulnerable system or services to attract malicious attackers, allowing security professionals to **monitor, detect, and analyze attack behavior** in a controlled environment.

---

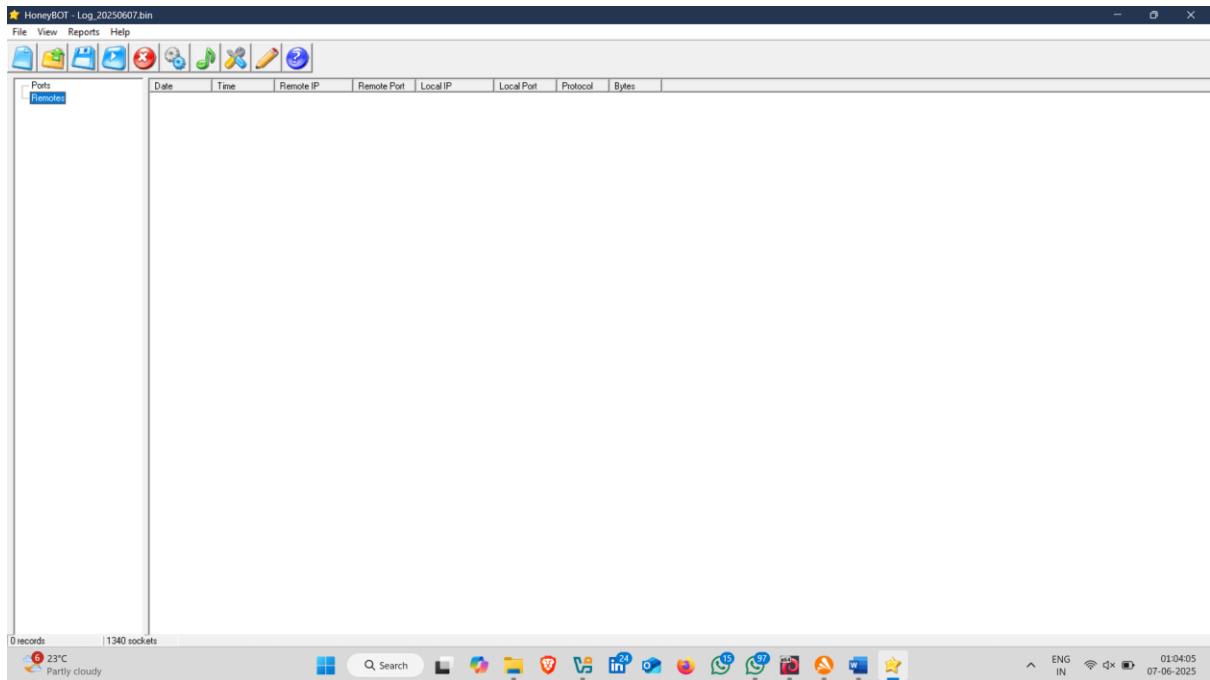
### Common Use Cases:

- Detecting early signs of network intrusions.
  - Learning about attackers' tactics and tools.
  - Gathering intelligence on malware and exploits.
  - Testing firewall and IDS/IPS effectiveness.
- 

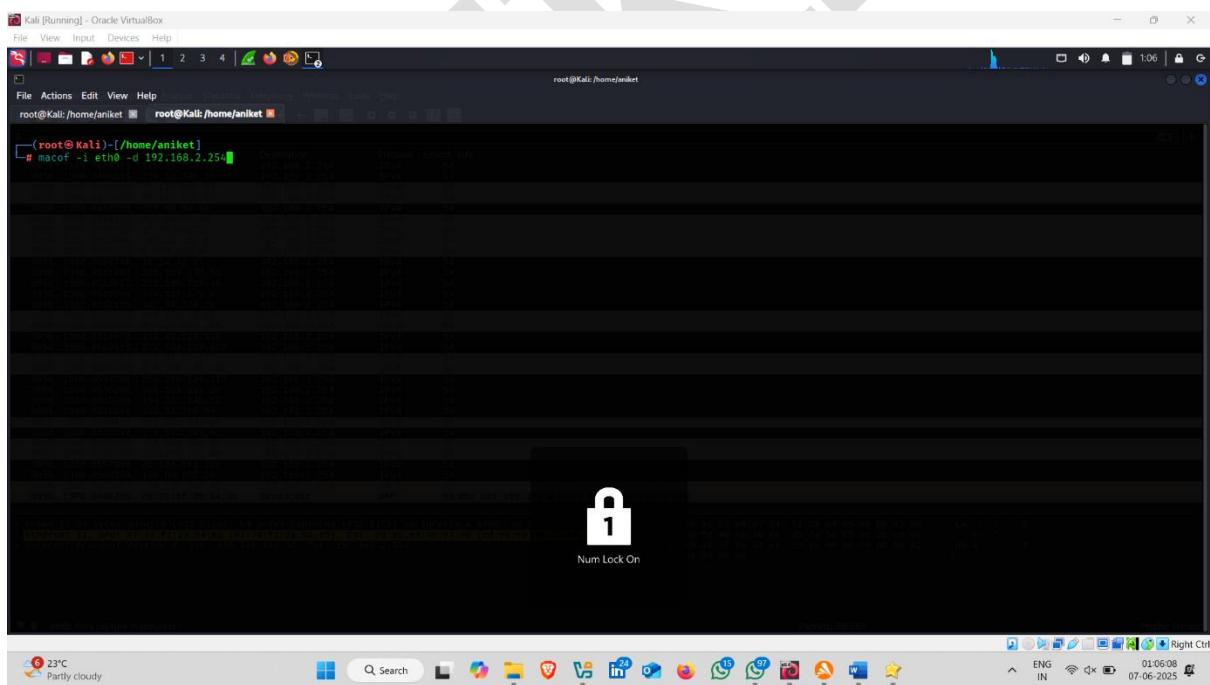
Download Link:- <https://honeybot.software.informer.com/download/>

**How to use it :-**

After installation HoneyBot Open it



- Now open kali linux and start attack on your machine



- Attack started 

- Detection started

The screenshot shows the HoneyBOT application window. The title bar reads "HoneyBOT - Log\_20250605.bin". The menu bar includes "File", "View", "Reports", and "Help". Below the menu is a toolbar with icons for file operations like Open, Save, Print, and Help. A tree view on the left lists "Pots" (with entry "138") and "Replies" (with entries "192.168.217.1", "192.168.254", "192.168.170.1", "192.168.56.1", and "192.168.2.118"). The main area displays a table of network log entries:

	Date	Time	Remote IP	Remote Port	Local IP	Local Port	Protocol	Bytes
138	05-06-2025	19:51:24	192.168.217.1	138	0.0.0.0	138	UDP	201
192.168.217.1	05-06-2025	19:51:24	192.168.217.1	138	0.0.0.0	138	UDP	201
192.168.254	05-06-2025	19:51:24	192.168.170.1	138	0.0.0.0	138	UDP	201
192.168.170.1	05-06-2025	19:51:24	192.168.56.1	138	0.0.0.0	138	UDP	201
192.168.56.1	05-06-2025	19:51:30	192.168.2.118	138	0.0.0.0	138	UDP	201

At the bottom, status bars show "5 records" and "1340 sockets". The taskbar at the bottom of the screen includes icons for File Explorer, Edge browser, FileZilla, InfraGard, OneDrive, Firefox, GitHub, and others, along with system status icons for battery, signal, and date/time.



## 2. Perform Attack Detection Using KFSensor

**KFSensor** is a **commercial honeypot-based Intrusion Detection System (IDS)** developed for Microsoft Windows environments. It simulates vulnerable network services to attract and detect malicious activity, such as port scanning, malware infections, and exploitation attempts.

- Platform: Windows
  - Type: Honeypot + IDS
- 

### \* Key Features of KFSensor

- Multiple Protocol Support
- Stealth Operation
- Custom Services
- Payload Capture
- GeoIP Location
- Windows Integration
- Alerting
- Log Analysis
- Plugins
- Real-time Monitoring
- Web-based Attack Detection
- Email Notification
- Syslog Integration
- Service Banner Customization
- Port Scanning Detection
- Fake File Server Emulation
- Attack Signature Matching
- Attack Source Identification
- Easy GUI Interface
- Exportable Reports

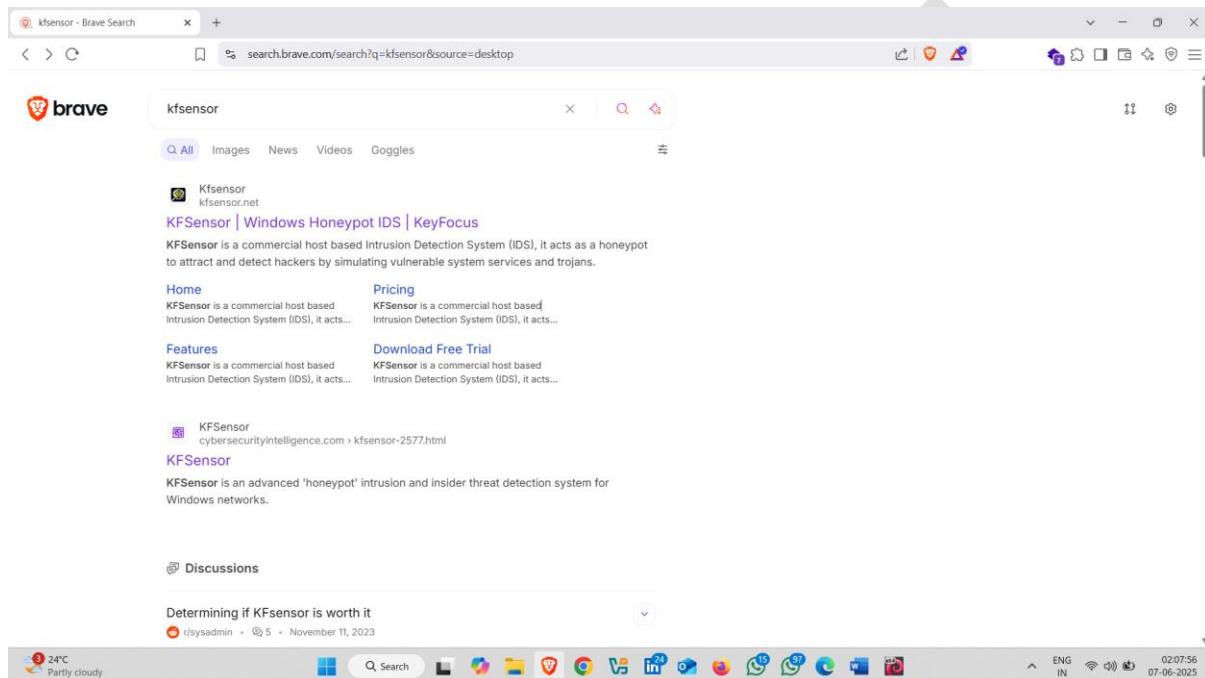
**Download Link :- <https://www.kfsensor.net/>**

**How to download it :-**

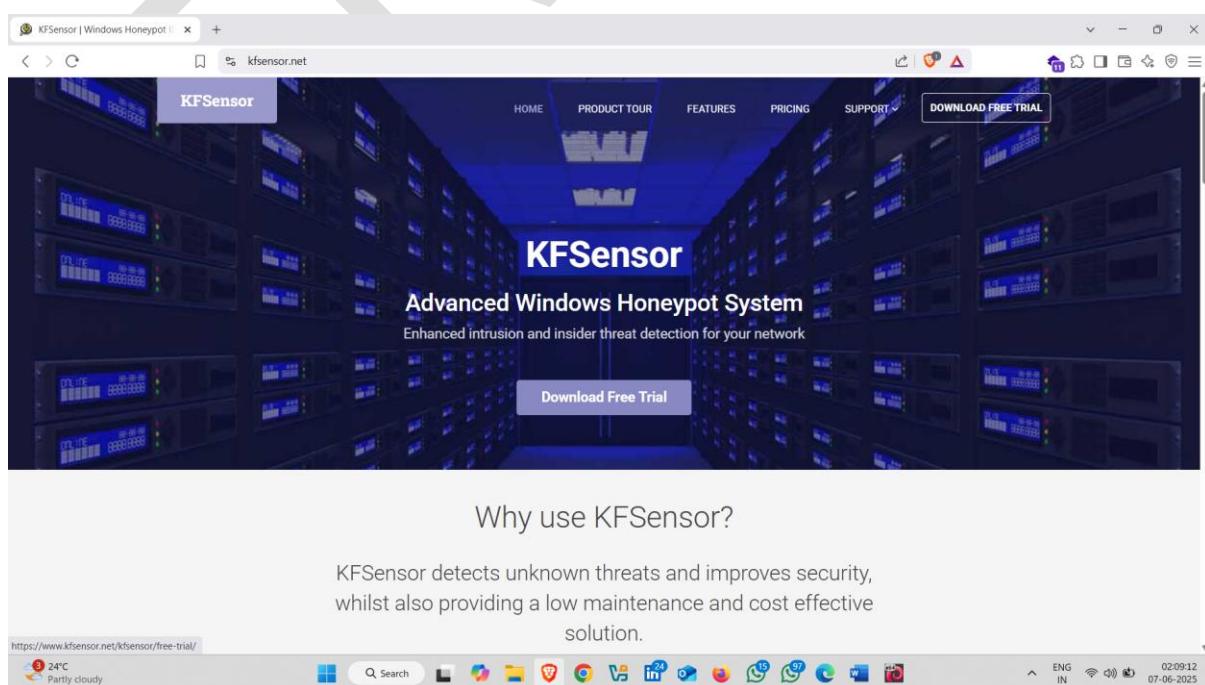
- Simply open the browser and search kfsensor download

**Note:- kfsensor is a paid tool , Download its free trial Version**

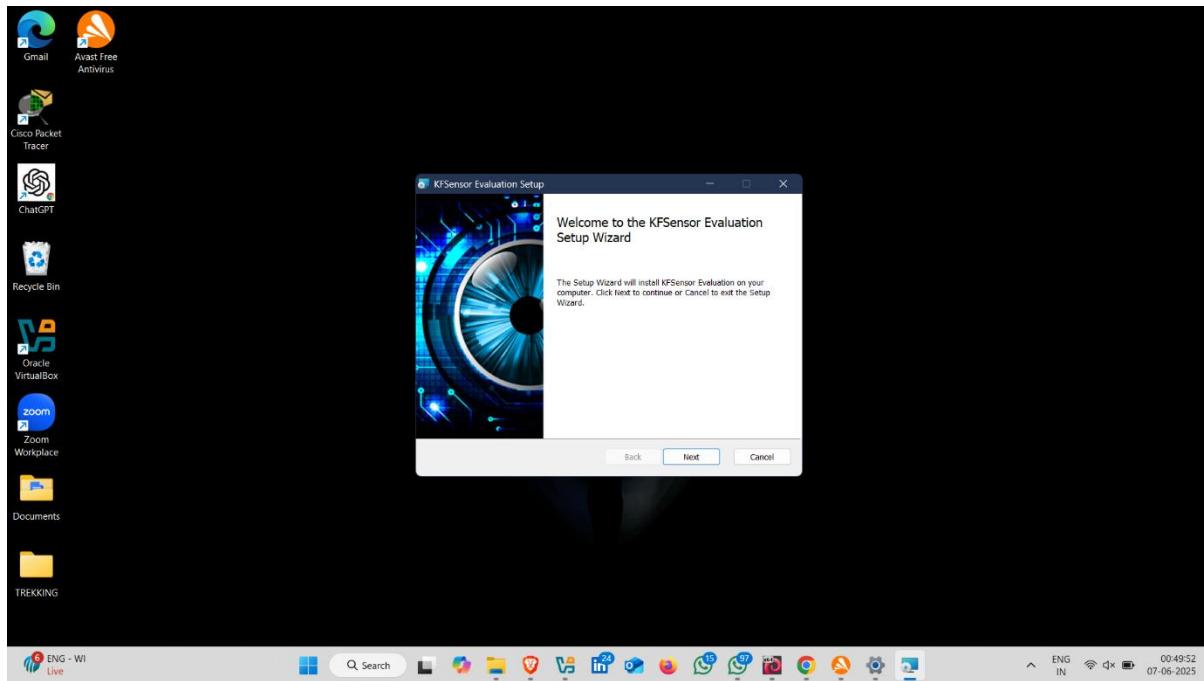
- Open First official Website 



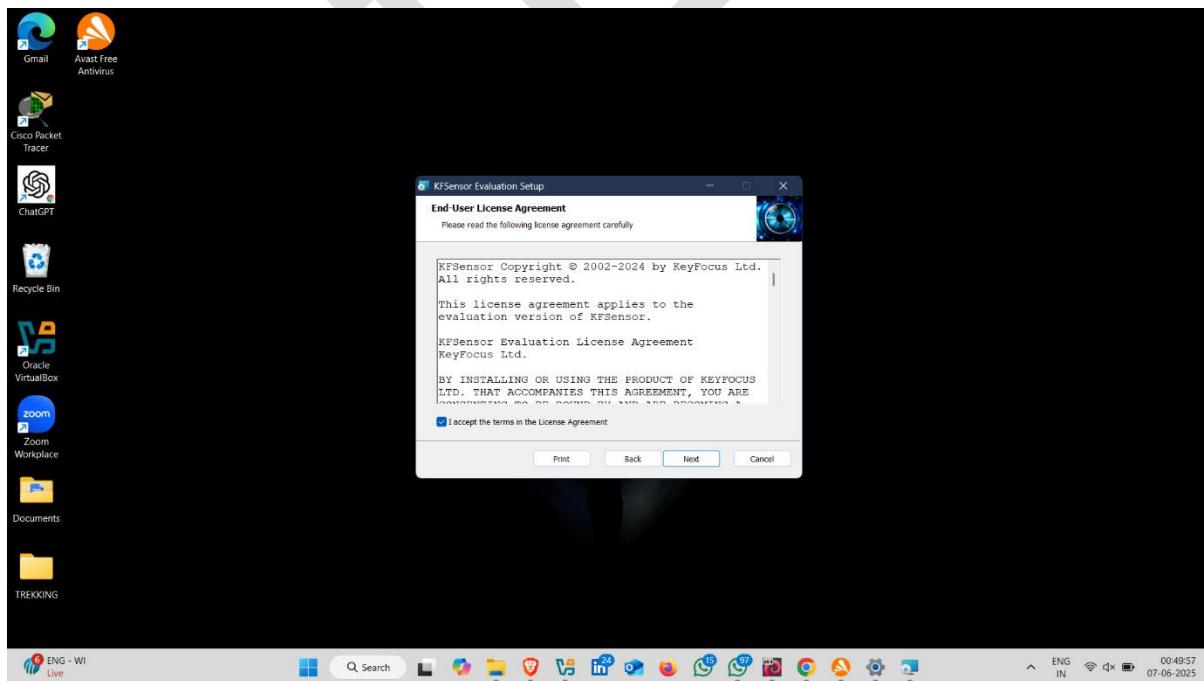
- Click on **Download Free Trial** and Download It



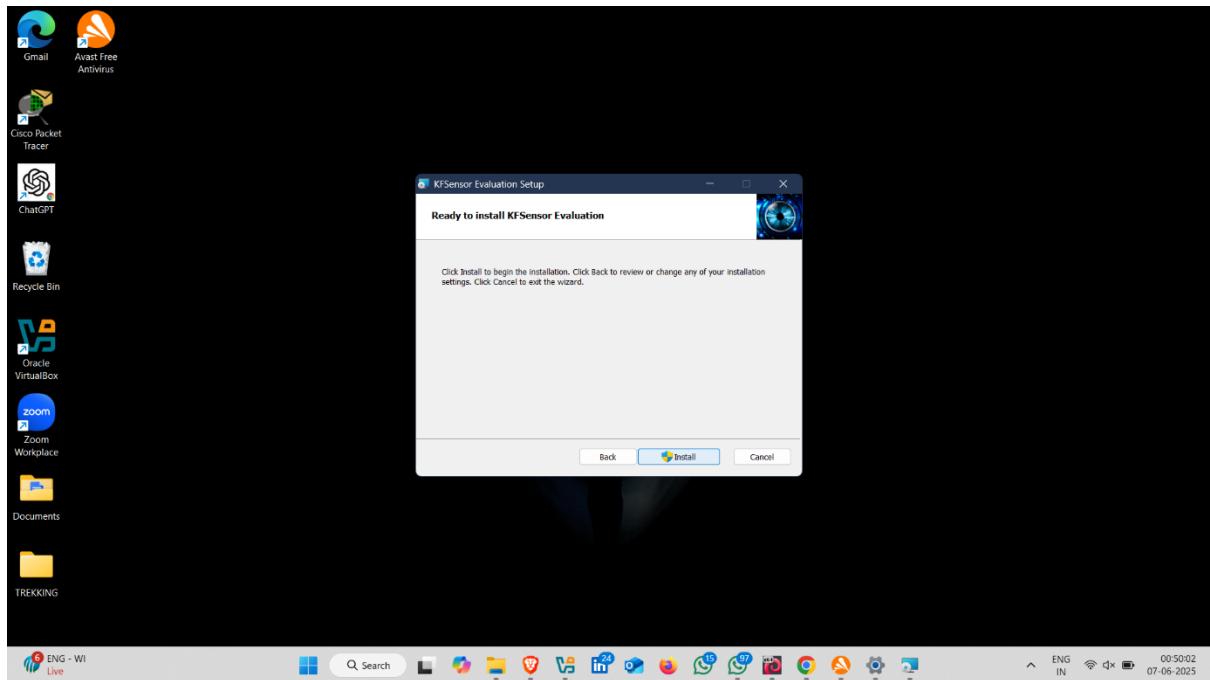
- After installation open the kfsensor.
- Click on Next



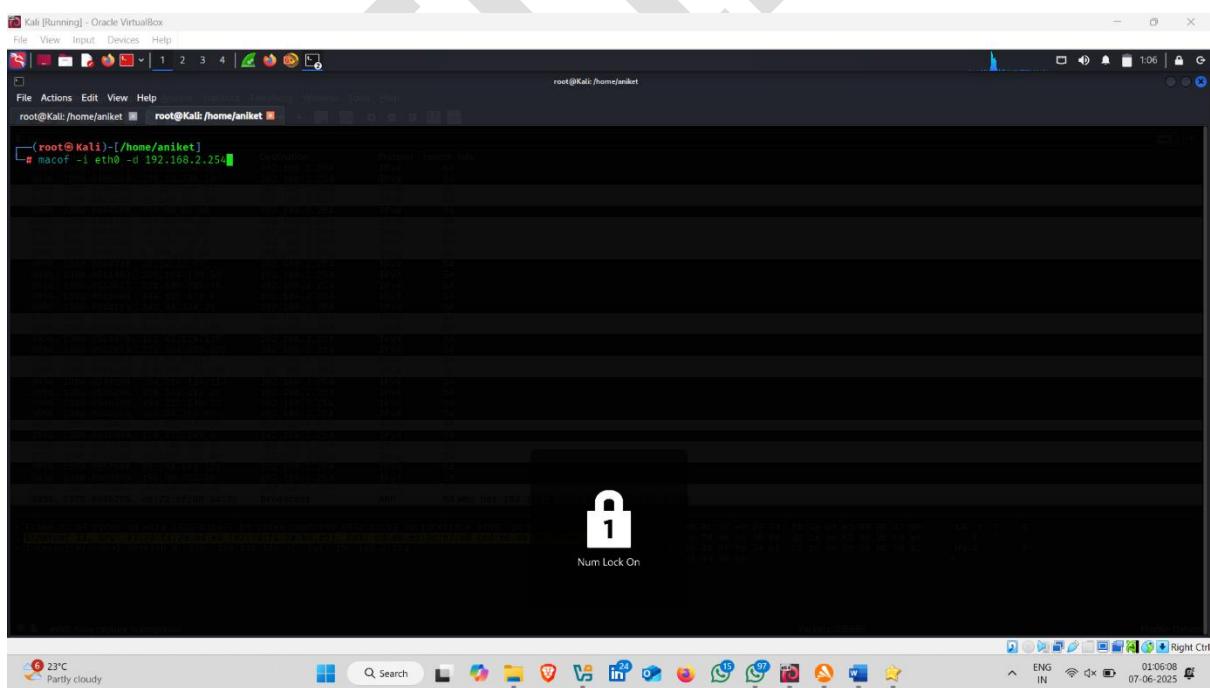
- Now accept the license and then click on next



- Click on Finish



- Now open kali linux virtual machine and start attack 



## • Attack Started

- Go back to the kfsensor , here detection started

KFSensor Professional - Evaluation Trial

File View Scenario Signatures Settings Help

TCP

- 0 Closed TCP Ports
- 21 FTP - Recent Activity
- 22 SSH - Recent Activity
- 23 Telnet - Recent Activity
- 25 SMTP - Recent Activity
- 53 DNS - Recent Activity
- 68 DHCP
- 80 IIS - Recent Activity
- 81 IIS 81 - Recent Activity
- 82 IIS 82 - Recent Activity
- 83 IIS 83 - Recent Activity
- 110 POP3 - Recent Activity
- 119 NTP
- 135 MS-RPC - Native service
- 139 NBT Session Service
- 389 LDAP
- 443 IIS HTTPS - Recent Activity
- 445 NBT-SMB - Native service
- 465 SMTP SSL
- 587 SMTP TLS
- 598 CIS - Recent Activity
- 993 POP3
- 1028 MSCS
- 1080 SOCKS - Recent Activity
- 1433 SQL Server
- 1723 Microsoft PPTP VPN - Recent...
- 1800 Message message broker - Rec...
- 2323 Telnet IoT - Recent Activity
- 2525 SMTP2
- 2869 MS UPNP Host - Recent Acti...
- 3128 IIS Proxy - Recent Activity
- 3389 Remote Desktop
- 4433 IIS HTTPS - Recent Activity
- 5000 MS Uni Plug and Play - Rece...
- 5357 Web Services for Devices
- 5358 Web Services on Devices API -
- 5555 Sensors Transport - Recent Act...
- 5800 VNC HTTP - Recent Activity
- 5985 Microsoft Windows Remote Ma...

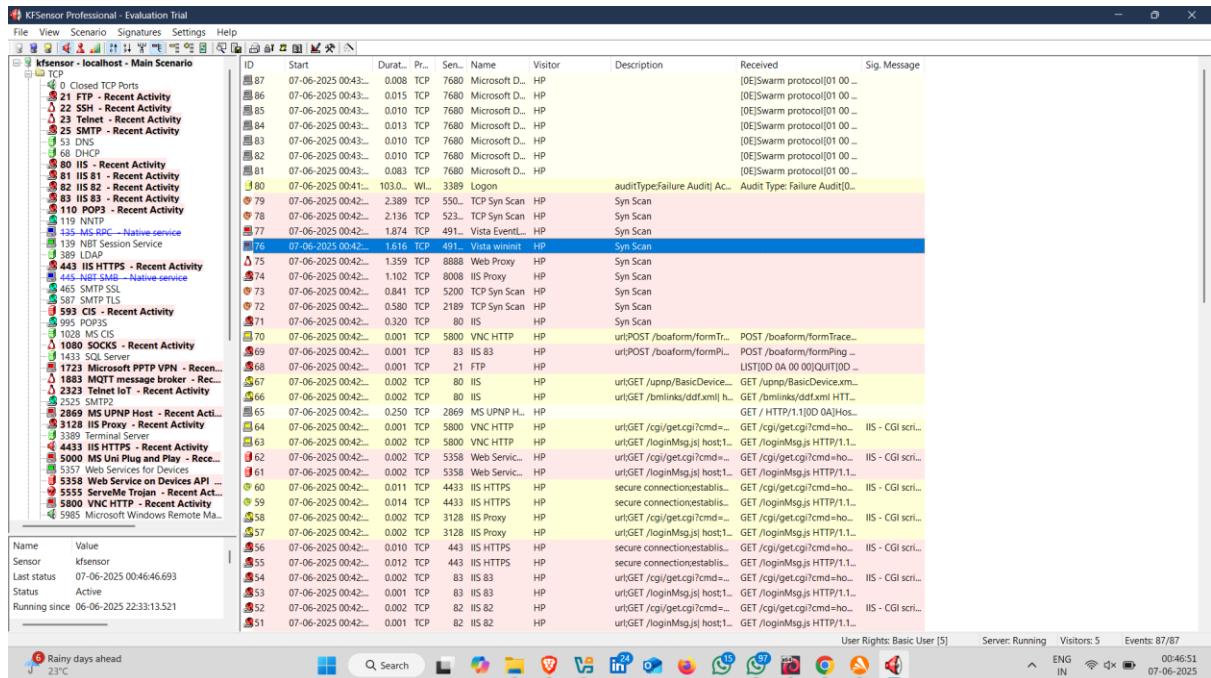
ID	Start	Durat...	Pr...	Sen...	Name	Visitor	Description	Received	Sig. Message
27	07-06-2025 00:41...	2.501	TCP	22	SSH	HP		SSH-2.0-libssh2_1.11.0(OD ...	

Name Value  
Sensor kfsensor  
Last status 07-06-2025 00:47:06.712  
Status Active  
Running since 06-06-2025 22:33:13.521

Rainy days ahead  
23°C

User Rights: Basic User [5] Server: Running Visitors: 5 Events: 1/87

ENG IN. 07-06-2025



## 3. Perform Attack Detection Using Wireshark

**Wireshark** is a free and open-source network protocol analyzer used to capture and inspect packets in real time across networks. It's used for network troubleshooting, security analysis, and protocol development.

### Wireshark Features :

- 1. Packet Capture**
- 2. Protocol Analysis**
- 3. Display Filters**
- 4. Capture Filters**
- 5. Color Coding**
- 6. Promiscuous Mode**

- 7. Follow TCP/UDP Stream**
- 8. Live Traffic Monitoring**
- 9. Packet Reassembly**
- 10. Expert Information**
- 11. Name Resolution**
- 12. Multiple Interface Capture**
- 13. Command-Line Support (TShark)**
- 14. Packet Export**
- 15. Custom Columns**
- 16. VoIP Analysis**
- 17. Decryption Support**
- 18. Statistics Tools**
- 19. Cross-Platform Support**
- 20. Free & Open Source**

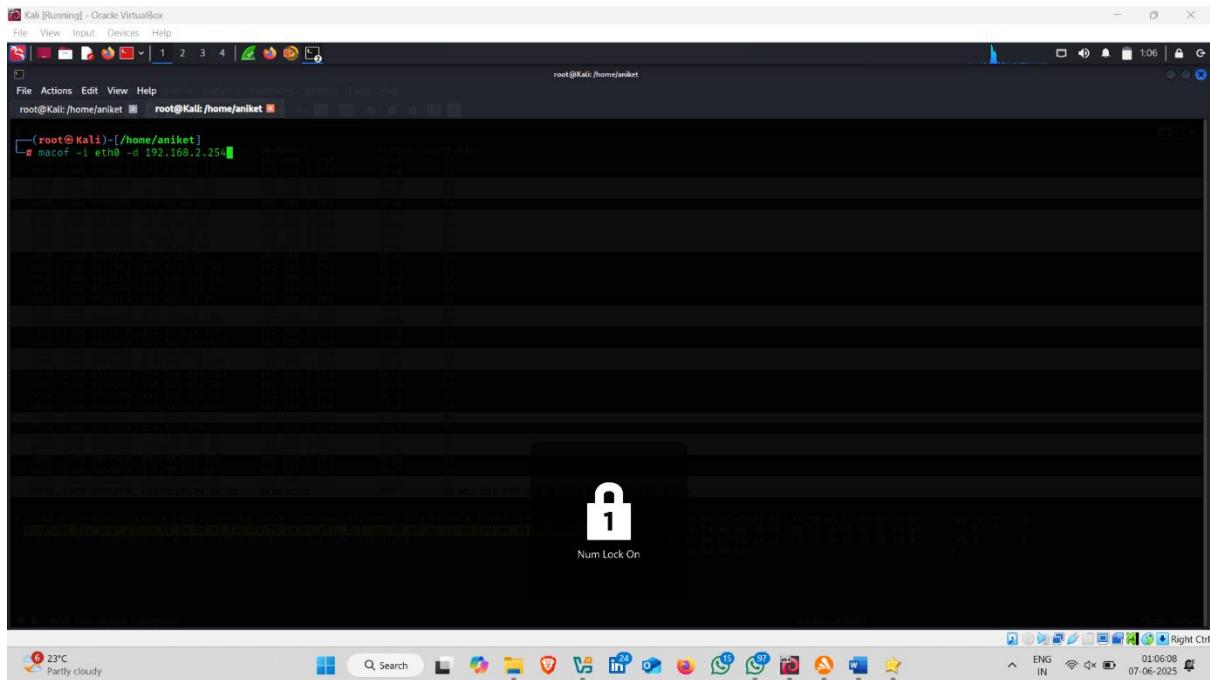
---

**How to use wireshark as a detection tool:-**

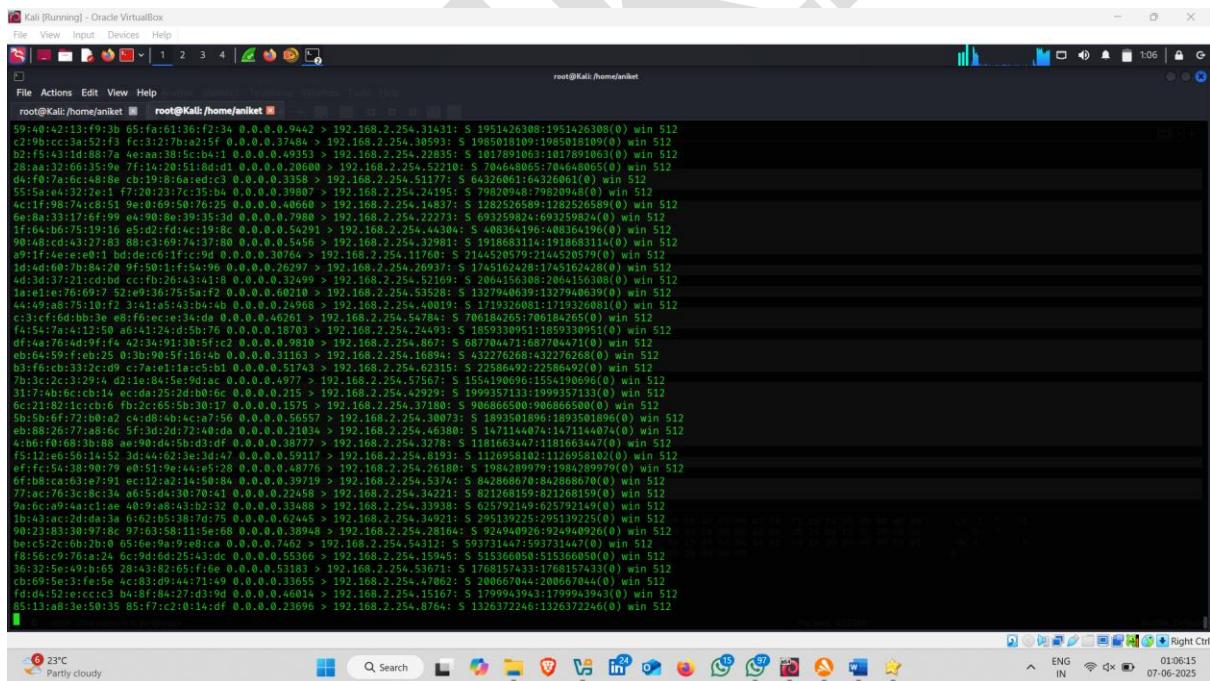
**Attacker machine – Kali Linux**

**Victim Machine – Windows 11**

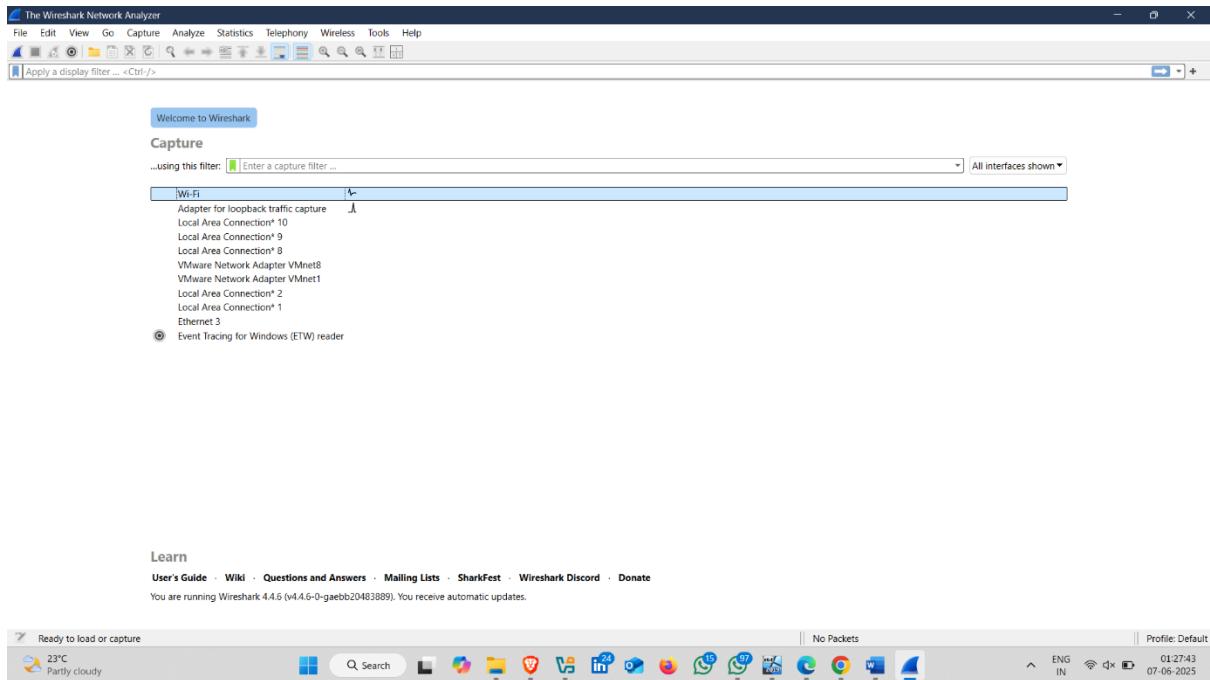
- Open Kali linux for Attack on Target
- Attack on target 



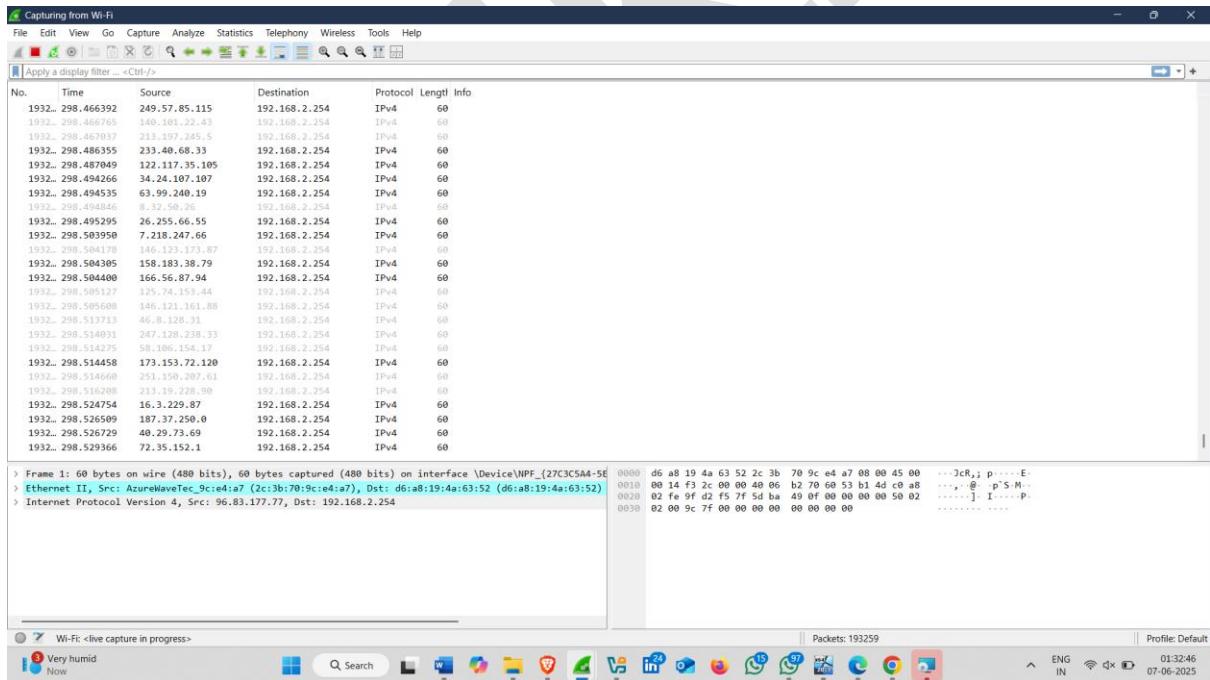
- Attack Started ✅ ⏱



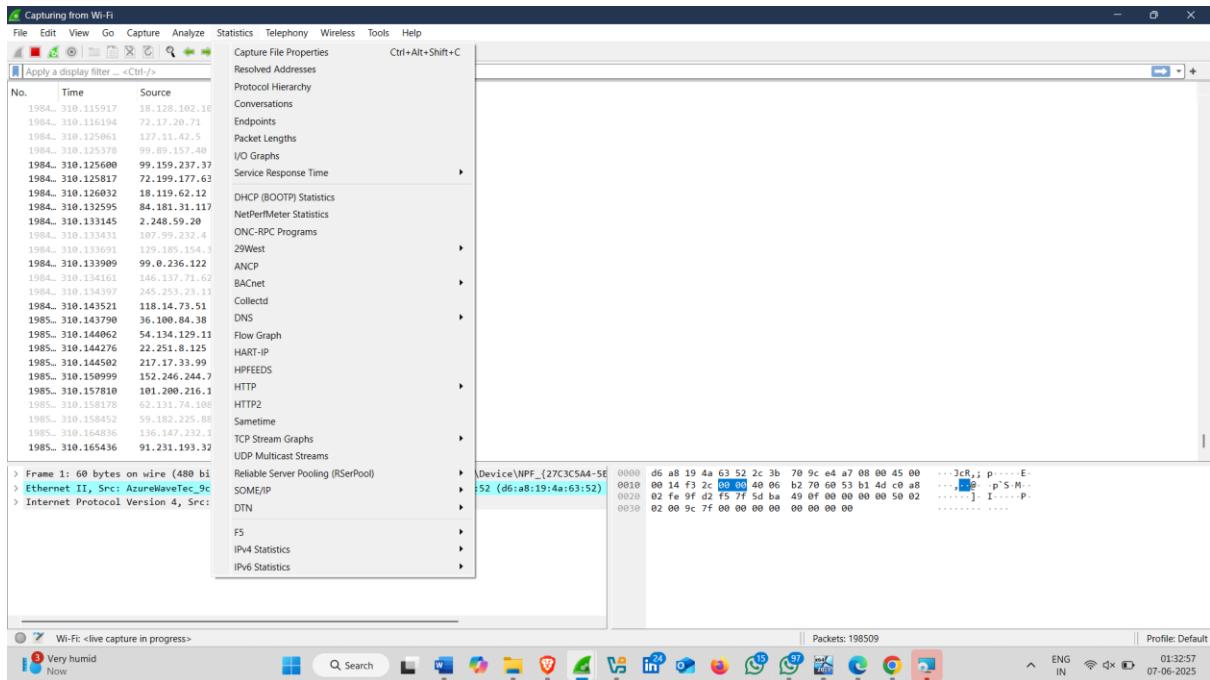
**Now Back to the Victim machine and open Wireshark**



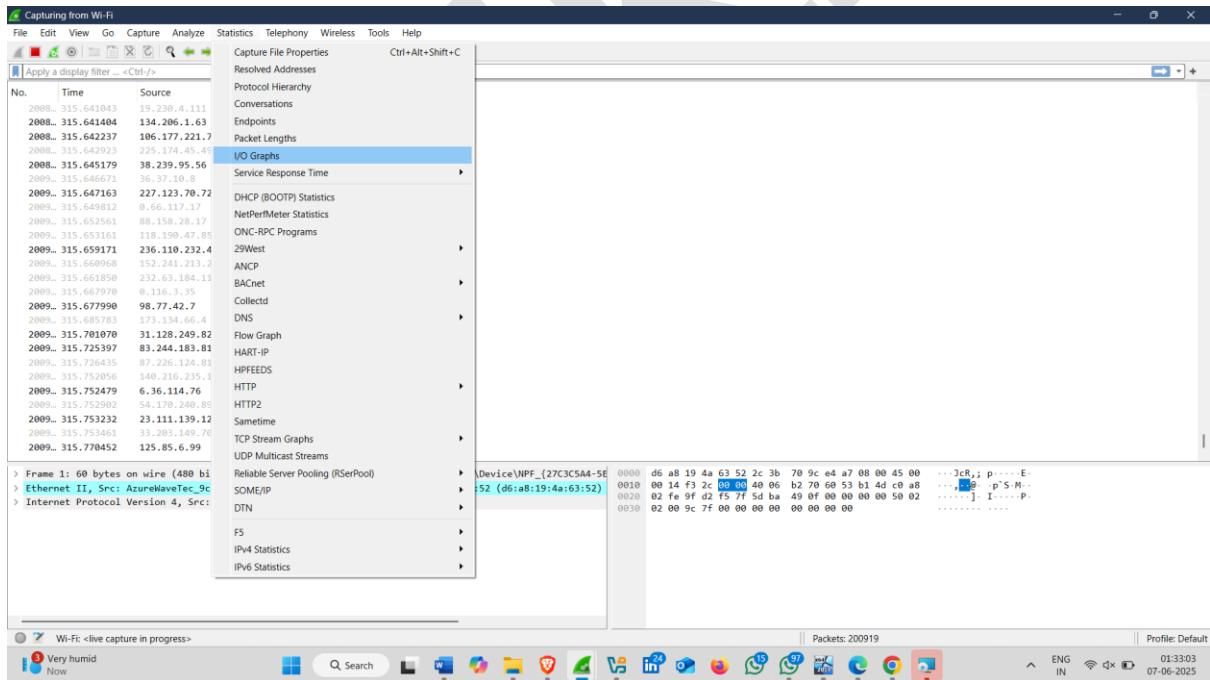
- Here , unknown flood of packets are received and captured



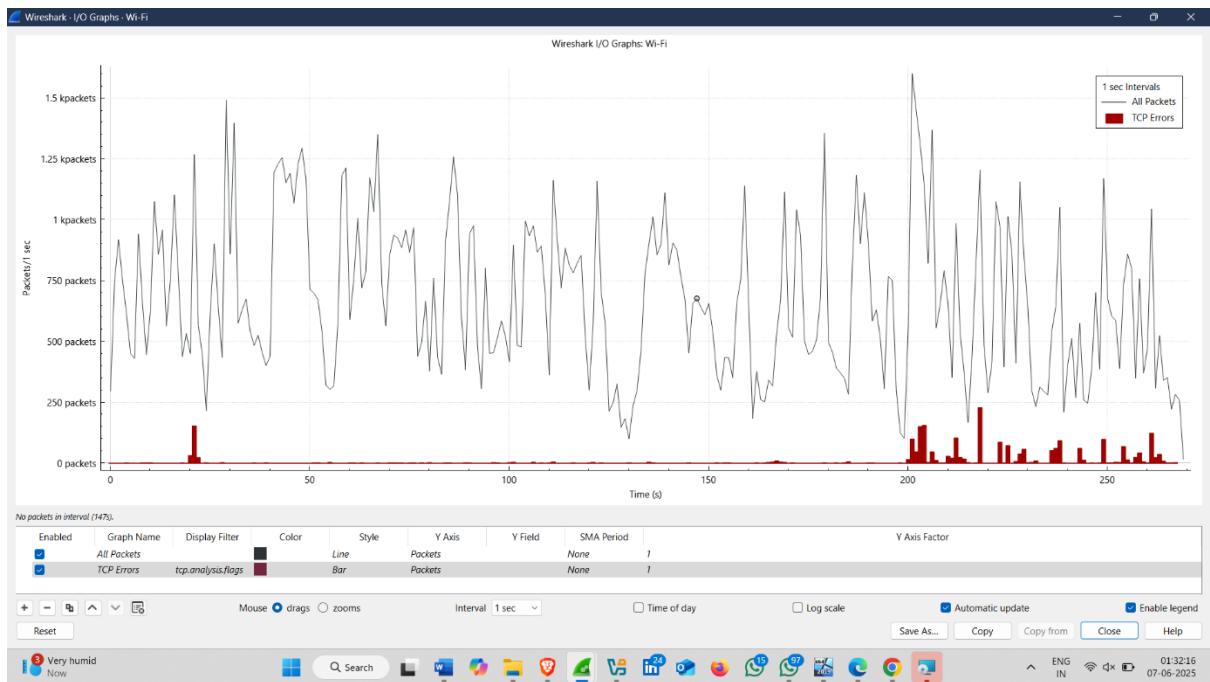
- You can also view on I/O Graphs
- Click on Statistics tab



- Click on I/O Graphs



- Here, You can See the Graph



# **1. How To Defend Against Firewall Evasion**

## **Common Firewall Evasion Techniques Attackers Use**

### **1. IP Spoofing**

- Faking the source IP address to appear as a trusted host.

### **2. Port Spoofing and Changing**

- Using non-standard or allowed ports to sneak in traffic instead of blocked ones.

### **3. Fragmentation Attacks**

- Splitting malicious packets into fragments to bypass inspection.

### **4. Protocol Anomalies and Violations**

- Sending malformed or unexpected packets to confuse firewall inspection.

### **5. Tunneling and Encapsulation**

- Wrapping malicious traffic inside allowed protocols (e.g., HTTP tunneling, DNS tunneling, VPN).

### **6. Use of Encrypted Traffic**

- Using SSL/TLS encryption so firewall cannot inspect packet contents.

### **7. Source Routing**

- Manipulating IP header fields to control packet route, potentially bypassing firewall filters.

### **8. Session Hijacking and TCP Sequence Prediction**

- Manipulating TCP session details to bypass firewall's stateful inspection.

## **9. Fragment Overlaps and Reassembly Confusion**

- Exploiting firewall's inability to properly reassemble and inspect fragmented packets.

## **10. Malicious Payload in Allowed Traffic**

- Hiding malware inside allowed web traffic (like HTTPS or DNS).
- 

## **How to Defend Against Firewall Evasion: Best Practices**

### **1. Implement Stateful Inspection Firewalls**

- Use firewalls that track the full state of active connections.
- Stateful firewalls can reject out-of-state or unexpected packets.

### **2. Enable Strict Packet Filtering and Validation**

- Validate protocol compliance for all traffic.
- Drop malformed, unexpected, or suspicious packets.

### **3. Use Deep Packet Inspection (DPI)**

- Inspect beyond headers, look into packet payloads.
- Detect tunneling and malicious payload hidden in allowed protocols.

### **4. Block IP Spoofing**

- Enable **Reverse Path Forwarding (RPF)** or Unicast RPF on routers/firewalls.
- Reject packets with source IP addresses that don't match legitimate routes.

### **5. Disallow Source Routing**

- Block IP packets with source routing options set.
- Many firewalls provide this option to reject such packets.

## **6. Reassemble Fragments for Inspection**

- Ensure firewall can fully reassemble fragmented IP packets before filtering.
- Prevent attackers from hiding data in packet fragments.

## **7. Inspect and Control Encrypted Traffic**

- Use SSL/TLS inspection or proxy servers to decrypt and analyze HTTPS traffic.
- Detect malware or command-and-control traffic hidden inside encryption.

## **8. Implement Strict Port and Protocol Controls**

- Only allow necessary ports and protocols explicitly.
- Block all unused or unnecessary services.

## **9. Regularly Update Firewall Firmware and Rules**

- Keep firewall software and firmware updated to patch evasion vulnerabilities.
- Regularly review and tune firewall rules.

## **10. Use Intrusion Prevention System (IPS) Alongside Firewall**

- IPS can detect and block evasive attack signatures the firewall misses.
- Adds an extra layer of inspection.

## **11. Monitor Logs and Traffic Patterns**

- Continuously monitor firewall logs for unusual or suspicious traffic.
- Detect patterns that may indicate evasion attempts.

## **12. Employ Network Segmentation and Zero Trust**

- Limit access between network segments.

- Use zero-trust principles so no traffic is trusted by default.

### **13. Implement Rate Limiting and Traffic Shaping**

- Limit traffic bursts and scanning attempts.
- Helps block slow or stealthy evasion attempts.

### **14. Use Anti-Spoofing Filters at Network Edge**

- Configure routers and switches to drop spoofed packets.
- Helps reduce malicious traffic before reaching firewall.

---

## **2. How To Defend Against IDS Evasion**

---

### **Common IDS Evasion Techniques Attackers Use**

#### **1. Fragmentation**

- Breaking malicious payloads into small IP fragments so IDS cannot reassemble or detect full attack.

#### **2. Packet Overlapping**

- Sending overlapping TCP/IP packets with conflicting data to confuse IDS packet reassembly.

#### **3. Protocol Anomalies**

- Using unusual or malformed packets that IDS may not parse correctly.

#### **4. Polymorphic Shellcode**

- Changing malware code structure or encryption to avoid signature detection.

#### **5. Payload Encoding**

- Encoding payload in formats like Base64, URL encoding, or Unicode to hide attack patterns.

## 6. Traffic Timing and Rate

- Sending packets slowly or in bursts to avoid threshold-based detection.

## 7. Using Encrypted Traffic

- Using SSL/TLS or VPN tunnels so IDS cannot inspect payload.

## 8. Evasion of Signature Matching

- Using variants or zero-day exploits that IDS signatures don't cover.

## 9. Avoiding Known Ports

- Using uncommon or allowed ports to hide malicious traffic.

## 10. Session Splicing

- Splitting attack across multiple sessions or connections to avoid detection.

---

## How to Defend Against IDS Evasion: Key Strategies

### 1. Use a Combination of Detection Methods

- Combine **signature-based** IDS with **anomaly-based** IDS.
- Anomaly-based IDS can detect unusual traffic behavior even if signature evasion is attempted.

### 2. Enable Full Packet Reassembly

- Configure IDS to fully reassemble fragmented IP packets and TCP streams before analysis.
- Prevent evasion via fragmentation or overlapping packets.

### **3. Regularly Update IDS Signatures**

- Frequently update IDS rules and signatures to cover latest exploits and evasion techniques.
- Use threat intelligence feeds and vendor updates.

### **4. Use Deep Packet Inspection (DPI)**

- Inspect packet payloads in detail, not just headers.
- Detect obfuscated or encoded payloads by decoding before analysis.

### **5. Implement SSL/TLS Inspection**

- Use SSL decryption capabilities or proxies to inspect encrypted traffic.
- Helps detect malicious content inside encrypted sessions.

### **6. Behavioral & Contextual Analysis**

- Correlate IDS alerts with logs from other systems (firewalls, endpoints).
- Use context like user behavior, time of day, and historical data to identify evasion.

### **7. Deploy Multi-Layered Security**

- Combine IDS with Intrusion Prevention Systems (IPS), firewalls, endpoint protection, and network segmentation.
- Defense in depth reduces chances of evasion success.

### **8. Tune IDS Rules to Reduce False Negatives**

- Customize rules to your environment to catch evasions without overwhelming false positives.
- Remove or modify ineffective rules.

### **9. Monitor Traffic Patterns**

- Look for unusual traffic timing, burstiness, or inconsistent flows.
- Use Network Behavior Anomaly Detection (NBAD) tools.

## **10. Log and Audit Everything**

- Keep detailed logs of IDS alerts, network flows, and system events.
- Helps analyze suspicious activity and detect stealthy attacks.

## **11. Use Honeypots and Deception Techniques**

- Deploy honeypots to lure attackers and detect evasive behaviors.
- Can help identify new evasion methods.

## **12. Train Security Personnel**

- Keep SOC analysts and network admins aware of IDS evasion tactics.
- Train to recognize suspicious activity that may bypass IDS.

---

## **3. How To Defend Against IPS Evasion**

### **Common IPS Evasion Techniques Attackers Use**

#### **1. Packet Fragmentation**

- Splitting attack payload into tiny fragments to avoid detection during packet reassembly.

#### **2. TCP Stream Manipulation**

- Manipulating TCP packets (sequence numbers, overlaps, out-of-order segments) to confuse IPS stream reassembly.

### **3. Protocol Violations and Anomalies**

- Using malformed packets or unusual protocol behavior to evade signature or protocol checks.

### **4. Encoding and Obfuscation**

- Encoding payloads (e.g., Base64, Unicode) or using polymorphic malware to bypass signature detection.

### **5. Traffic Timing Evasion**

- Sending attacks slowly over time (low-and-slow) to avoid triggering threshold-based IPS rules.

### **6. Use of Encrypted Traffic**

- Encrypting attack payloads inside SSL/TLS so IPS can't inspect contents without decryption.

### **7. Tunneling and Encapsulation**

- Wrapping attacks inside allowed protocols (HTTP, DNS, SSH tunnels).

### **8. Payload Padding and NOP Sleds**

- Adding no-op instructions or junk bytes to disrupt signature matching.

### **9. Attack Variants and Zero-day Exploits**

- Using unknown or slightly modified exploits unknown to IPS signatures.

### **10. Session Splicing**

- Splitting attacks across multiple TCP sessions or connections.

## **How to Defend Against IPS Evasion: Key Strategies**

### **1. Use Stateful and Full Stream Reassembly**

- IPS should perform full TCP stream reassembly.
- Proper handling of out-of-order, overlapping, or fragmented packets to reconstruct complete payload.

### **2. Employ Deep Packet Inspection (DPI)**

- Analyze packet payloads deeply, including decoding common encodings.
- Detect obfuscated or encoded attacks.

### **3. Decrypt Encrypted Traffic**

- Use SSL/TLS inspection capabilities or proxies to inspect encrypted sessions.
- Identify attacks hidden inside encrypted tunnels.

### **4. Combine Signature-based and Anomaly-based Detection**

- Signature-based detects known patterns.
- Anomaly-based detects unusual or suspicious behavior.

### **5. Keep IPS Signatures and Software Up-to-Date**

- Regularly update to detect new exploits and evasion methods.
- Subscribe to threat intelligence feeds.

### **6. Use Behavioral Analysis and Correlation**

- Monitor traffic behavior, session patterns, and correlate with endpoint logs.
- Detect stealthy, low-rate, or polymorphic attacks.

### **7. Implement Multi-layered Security**

- Combine IPS with firewalls, endpoint protection, network segmentation.

- Defense in depth reduces evasion success.

## **8. Tune IPS Rules Carefully**

- Adjust sensitivity to balance false positives and negatives.
- Customize rules for your environment.

## **9. Monitor and Analyze Logs Continuously**

- Use Security Information and Event Management (SIEM) tools.
- Detect signs of evasion attempts.

## **10. Perform Regular Penetration Testing**

- Test IPS resilience to evasion.
- Identify weaknesses and improve configuration.

**THANK YOU**

**ANTIQUE**