



INTRUSION



DETECTION



SYSTEM



FIREWALL

-ATHARVA KATKAR



INTRUSION



PREVENTION



SYSTEM

TABLE OF CONTENTS

TABLE OF CONTENTS

1. Firewall

- 1.1 What is a Firewall
- 1.2 Types of Firewalls
 - 1.2.1 Packet-Filtering Firewall
 - 1.2.2 Stateful Inspection Firewall
 - 1.2.3 Application-Level Gateway (Proxy Firewall)
 - 1.2.4 Next-Generation Firewall (NGFW)
 - 1.2.5 Software Firewall
 - 1.2.6 Hardware Firewall
- 1.3 Why Firewalls Are Important

2. Intrusion Detection System (IDS)

- 2.1 What is IDS
- 2.2 Uses / Objectives of IDS
- 2.3 How IDS Works (Step-by-Step)
- 2.4 Types of IDS
 - 2.4.1 Network-Based IDS (NIDS)
 - 2.4.2 Host-Based IDS (HIDS)
- 2.5 IDS Detection Techniques
 - 2.5.1 Signature-Based Detection
 - 2.5.2 Anomaly-Based Detection
 - 2.5.3 Protocol Anomaly Detection
- 2.6 Types of IDS Alerts
 - 2.6.1 True Positive
 - 2.6.2 True Negative
 - 2.6.3 False Positive
 - 2.6.4 False Negative

3. Intrusion Prevention System (IPS)

- 3.1 What is IPS
- 3.2 IPS Prevention Actions
- 3.3 How IPS Works
- 3.4 Types of IPS
 - 3.4.1 Network-Based IPS (NIPS)
 - 3.4.2 Host-Based IPS (HIPS)

3.4.3 Wireless IPS (WIPS)

3.4.4 Network Behaviour Analysis IPS (NBA IPS)

4. Honeypot

4.1 What is a Honeypot

4.2 Main Goals of Honeypots

4.3 How a Honeypot Works

4.4 Types of Honeypots

 4.4.1 Level of Interaction

 4.4.2 Based on Purpose

 4.4.3 Based on Location

 4.4.4 Based on Technology

4.5 Honeynet

5. Snort

5.1 What is Snort

5.2 Features of Snort

5.3 Main Functions of Snort

5.4 How Snort Works

5.5 Snort Installation & Configuration

 5.5.1 Configuration File Editing

 5.5.2 Running Snort Commands

6. Windows Firewall Configuration

6.1 Purpose of Windows Firewall

6.2 Key Features of Windows Firewall

6.3 Inbound Rules Configuration

6.4 Outbound Rules Configuration

7. Extra Activity – ZoneAlarm Firewall

7.1 What is ZoneAlarm Firewall

7.2 Key Features of ZoneAlarm

7.3 Application Permissions

7.4 Connection Options

8. Firewall Evasion Defence

- 8.1 How to Defend Against Firewall Evasion
- 8.2 Common Firewall Evasion Techniques

9. IDS Evasion Defence

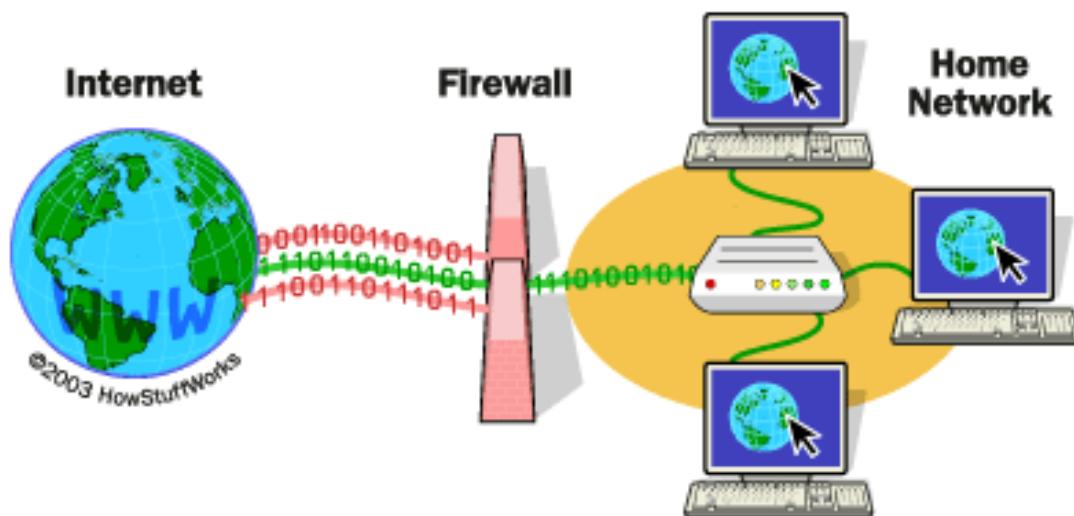
- 9.1 How to Defend Against IDS Evasion
- 9.2 IDS Evasion Techniques

ATHARVA

FIREWALL IDS / IPS

What Is a 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 predefined rules such as IP address, port number, and protocol.
- It is basic and fast but provides limited inspection of traffic.

2. Stateful Inspection Firewall

- Monitors and tracks active connections.
- Makes filtering decisions based on the state of the traffic.
- More secure than packet-filtering firewalls.

3. Application-Level Gateway (Proxy Firewall)

- Filters traffic at the application layer (for example, HTTP and FTP).
- Inspects the content of traffic, providing deep and detailed security.

4. Next-Generation Firewall (NGFW)

- Combines traditional firewall capabilities with advanced security features such as:
 - Deep packet inspection
 - Intrusion prevention
 - Application awareness
 - Malware protection

5. Software Firewall

- Installed on individual devices (for example, Windows Defender Firewall).
- Protects only the device on which it is installed.

6. Hardware Firewall

- A physical device placed between the internal network and the internet.
- Commonly used in organizations and business environments.

Why Firewalls Are Important

- Prevent unauthorized access to networks
- Protect systems from malware and cyberattacks
- Enforce organizational security policies
- Log and audit network activity for monitoring and analysis

INTRUSION DETECTION SYSTEM (IDS)

What Is an Intrusion Detection System (IDS)

An Intrusion Detection System (IDS) is a security tool that monitors and analyses network or system activities to detect unauthorized access, malicious behaviour, or violations of security policies.

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 the early detection of cyberattacks.
6. Assists in forensic analysis after a security incident.
7. Enforces organizational security policies.
8. Complements firewalls and other security tools.
9. Tracks user activity to detect insider threats.

How IDS Works (Step-by-Step)

1. Traffic / Activity occurs
2. IDS captures data
3. IDS analyses behaviour
4. Match with rules or patterns
5. Alert is generated

➤ **Types of IDS:**

1. Network-Based Intrusion Detection System (NIDS)

Definition

A Network-Based Intrusion Detection System (NIDS) monitors network traffic across a specific network segment to detect suspicious or malicious activities.

Key Points

- Placed at strategic points in the network, such as near gateways or the DMZ.
- Analyses all incoming and outgoing network traffic.
- Most effective for detecting large-scale or external attacks.

Examples

- Snort
- Suricata

Use Cases

- Detecting port scans, Denial-of-Service (DoS) attacks, and malware traffic.
- Monitoring real-time data flow within a network environment

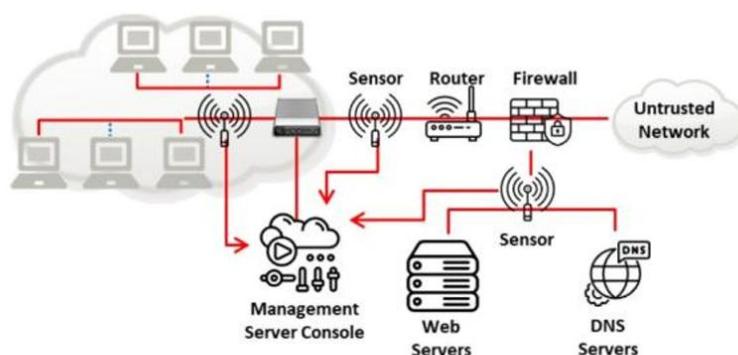


Fig:-Network-Based IDS

2. Host-Based Intrusion Detection System (HIDS)

Definition

A Host-Based Intrusion Detection System (HIDS) is a security solution that monitors the internal behaviour and activities of a specific host or endpoint to detect suspicious or malicious actions.

Key Points

- Installed directly on individual systems such as servers and workstations.
- Analyses system logs, file integrity, configuration changes, and user behaviour.
- Most effective for detecting insider threats and local system compromises.

Examples

- OSSEC, Tripwire

Use Cases

- Detecting unauthorized file modifications and suspicious login attempts.
- Protecting critical servers and endpoints from local exploits and attacks.

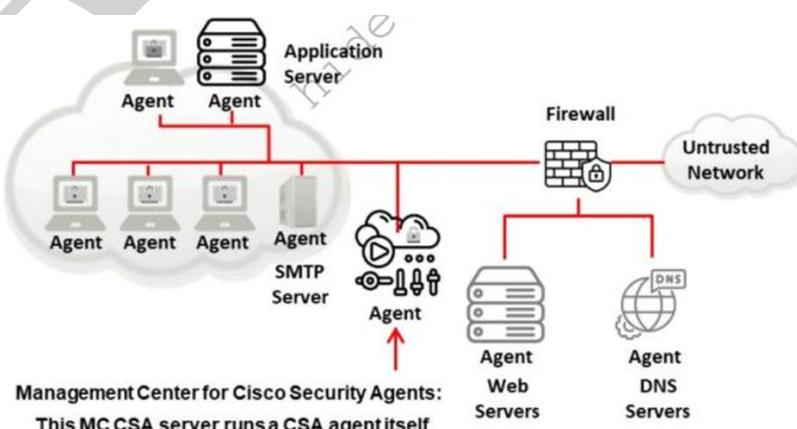


Fig- Host-Based IDS

➤ How an IDS Detects an Intrusion?

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

1. Signature-Based Detection

- Compares network traffic or system activity against a database of known attack patterns called signatures.
- Detects attacks by matching activity with predefined signatures.

Example:

- Identifies a known malware hash or a specific exploit packet.

Strength:

- Highly accurate for detecting known threats.

Weakness:

- Cannot detect new, unknown, or zero-day attacks.

2. Anomaly-Based Detection

- Establishes a baseline of normal behaviour for a system or network.
- Flags any activity that deviates significantly from this normal behaviour.

Example:

- A sudden spike in outbound network traffic from a server at midnight.

Strength:

- Capable of detecting unknown or zero-day attacks.

Weakness:

May generate false positives due to unusual but legitimate activity.

3. Protocol Anomaly Detection (PAD)

- A specialized form of anomaly-based intrusion detection.
- Identifies intrusions by detecting deviations from normal protocol behaviour as defined by protocol standards such as TCP, HTTP, and DNS.

Example:

- Detecting malformed packets or abnormal protocol usage that violates standard protocol rules.

➤ Types of IDS Alerts

IDS alerts are categorized based on how accurately the Intrusion Detection System identifies malicious activity.

1. True Positive (TP)

- An attack is present, and the IDS successfully detects it.
- This represents a correct and accurate 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 any alert.
- This indicates correct system behaviour.

Example:

- Normal user traffic occurs, and the IDS remains silent.

3. False Positive (FP)

- No attack is present, but the IDS raises an alert.
- This is an incorrect detection and may lead to unnecessary investigation.

Example:

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

4. False Negative (FN)

- An attack is present, but the IDS fails to detect it.
- This is dangerous because the attack goes unnoticed.

Example:

- A new type of ransomware enters the network, but the IDS does not generate an alert.

INTRUSION PREVENTION SYSTEM (IPS)

An Intrusion Prevention System (IPS) is a network security solution that sits in-line (directly in the traffic path) between network devices and continuously monitors network traffic. When suspicious or malicious activity is detected, the IPS can automatically take preventive action to stop the threat.

Actions performed by an IPS include:

- Dropping malicious packets
- Blocking traffic from a specific IP address
- Resetting network connections
- Alerting system or network administrators

➤ How an IPS Works

1. Traffic Inspection
 - Analyses network traffic using Deep Packet Inspection (DPI).
2. Signature Matching
 - Compares traffic patterns with known attack signatures such as viruses and exploits.
3. Behaviour Analysis
 - Detects abnormal behaviour, such as excessive requests from a single IP address.
4. Prevention
 - Automatically blocks or stops attacks before they can succeed.

Types of Intrusion Prevention Systems:

1. Network-Based IPS (NIPS)

- Location: Deployed at key points in the network, usually behind a firewall.
- Purpose: Monitors all traffic flowing between network hosts.
- Detection Scope: Protects the entire network.

Examples:

- Cisco Firepower
- Suricata
- Snort (IPS mode)

Best For:

- Detecting and blocking threats that spread across the network.

2. Host-Based IPS (HIPS)

- Location: Installed on individual systems such as servers, desktops, and laptops.
- Purpose: Monitors system calls, logs, application behavior, and file access.
- Detection Scope: Protects only the local system on which it is installed.

Examples:

- OSSEC (with HIPS modules)
- Symantec Endpoint Protection

Best For:

- Detecting local exploits such as privilege escalation and unauthorized file modifications.

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 such as corporate Wi-Fi networks and public access areas.

4. Network Behavior Analysis (NBA) IPS

- Location: Typically integrated into advanced IPS or security appliances.
- Purpose: Monitors network traffic to identify unusual behavior such as large data transfers or Distributed Denial-of-Service (DDoS) attacks.
- Detection Method: Uses behavior modeling and anomaly detection techniques.

Examples:

- Darktrace
- Cisco Stealthwatch

Best For:

- Identifying zero-day attacks, insider threats, and large-scale network attacks.

HONEYBOT

What is a 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. This allows defenders to monitor, detect, and analyse malicious activity without putting real systems at risk.

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 security defenses based on collected insights.

How a Honeypot Works

1. It mimics a legitimate target such as a web server, database, or IoT device.
2. It is placed in a controlled and isolated environment, separate from real systems.
3. When an attacker interacts with it (scanning, exploitation, etc.), all actions are logged and recorded.
4. The security team analyzes this data for investigation and threat intelligence.

Types of Honeypots

1. Based on Level of Interaction

- Low-Interaction Honeypots – Simulate limited services with minimal risk.
- Medium-Interaction Honeypots – Provide more interaction without a full operating system.
- High-Interaction Honeypots – Fully emulate real systems for in-depth attacker engagement.

2. Based on Purpose

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

3. Based on Location

- External Honeypots – Placed outside the firewall to capture external attackers.
- Internal Honeypots – Placed inside the network to detect insider threats or lateral movement.

4. Based on Technology

- Malware Honeypots – Designed to capture and analyze malware.
- Spam Honeypots – Mimic open email relays to attract spam messages.
- Database Honeypots – Simulate vulnerable databases to detect SQL injection and data exfiltration attempts.
- Web Application Honeypots – Mimic websites or CMS platforms to detect web-based attacks.
- Industrial Control System (ICS) Honeypots – Simulate SCADA/ICS environments to detect threats targeting critical infrastructure.

Honeynet

Honeynet – A network of multiple interconnected honeypots designed for broader attack simulation and detailed analysis.

SNORT

What is SNORT?

SNORT is an open-source Network Intrusion Detection System (NIDS) and Intrusion Prevention System (IPS) developed by Martin Roesch. It is used to monitor network traffic in real time, detect malicious activity, and generate alerts when suspicious behavior is found. SNORT analyzes network packets and compares them against a database of predefined rules and signatures to identify attacks such as malware, denial-of-service (DoS), port scans, buffer overflows, and other threats.

Features of SNORT

- Real-time traffic analysis
- Packet logging and inspection
- Signature-based detection
- Protocol analysis
- Custom rule creation
- Can work as IDS or IPS
- Lightweight and fast
- Free and open-source

Main Functions of SNORT

1. **Packet Sniffing** – Captures and displays network packets.
2. **Packet Logging** – Logs packets to disk for later analysis.
3. **Intrusion Detection** – Detects attacks using rules and signatures.
4. **Intrusion Prevention** – Blocks malicious traffic when used in inline mode.

How SNORT Works

1. SNORT captures network traffic using a network interface card (NIC).
2. Captured packets are decoded into readable format.
3. Preprocessors normalize and prepare traffic for analysis.
4. The detection engine compares traffic with predefined rules.
5. If a rule matches, an alert is generated or action is taken.
6. Logs and alerts are stored for security analysis.

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

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

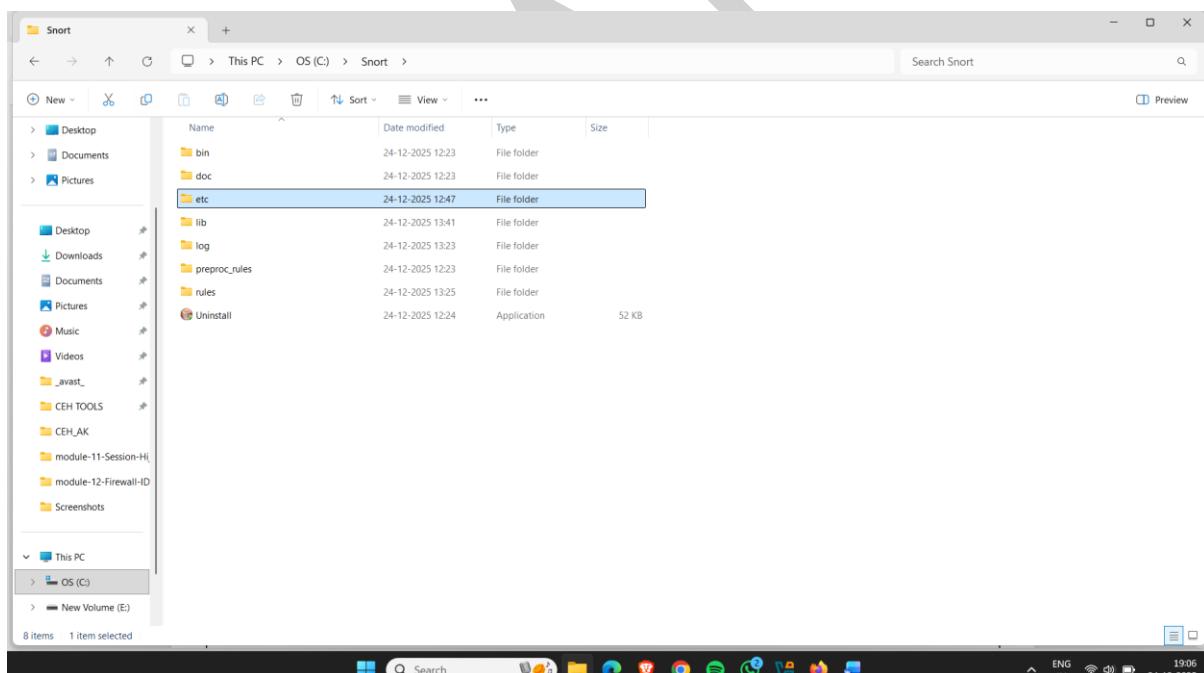


Figure 1

- Copy snort.exe file and paste it again on same location.

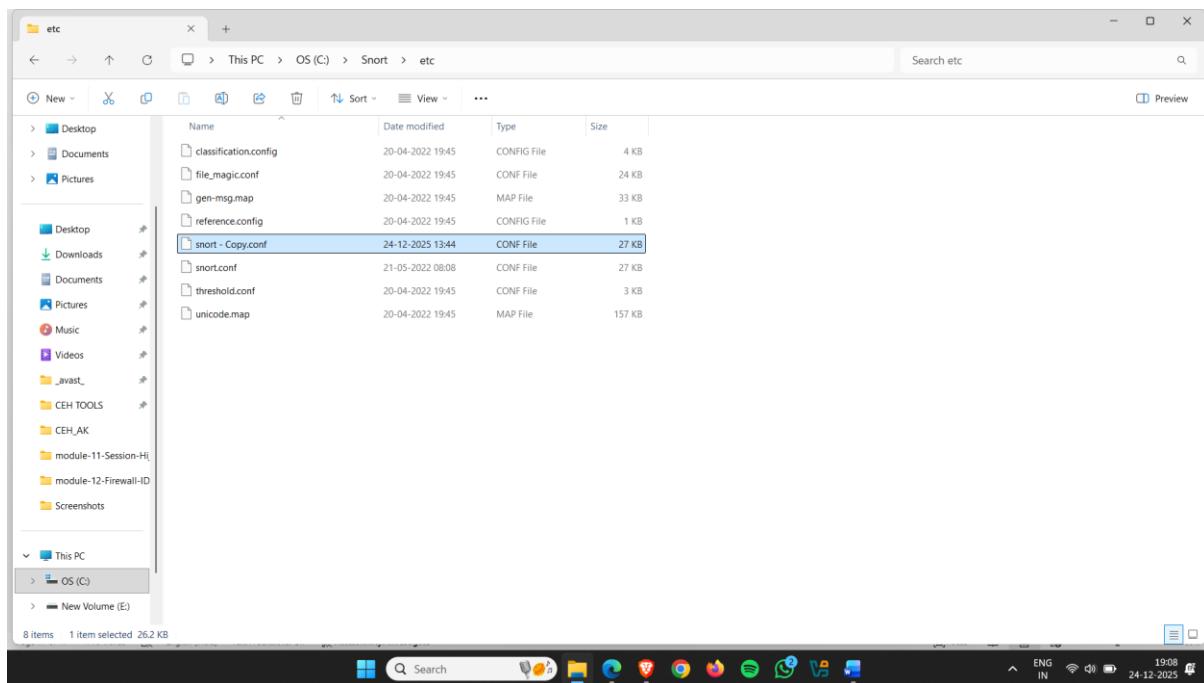


Figure 2

- Now , open copied file in Notepad++

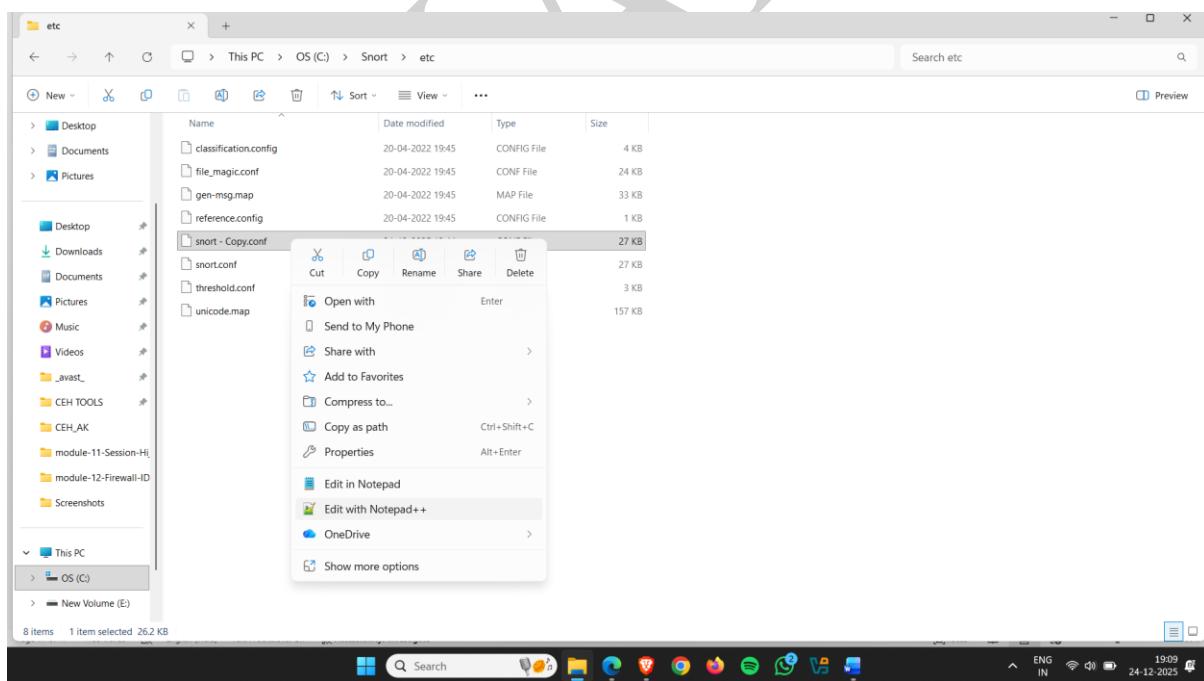
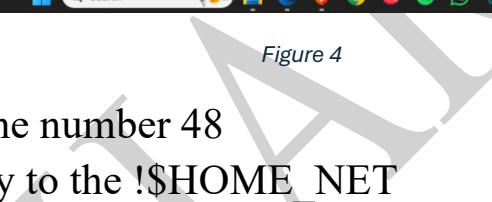


Figure 3

- Go to the line number 45
- Replace any word to ip address



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

```

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
snort - Copy.conf & [x]
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 ##### Step #1: Set the network variables. For more information, see README.variables
41 #####
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 #####
59 # List of sql servers on your network
60 ipvar SQL_SERVERS $HOME_NET
61 #####
62 # List of telnet servers on your network
63 ipvar TELNET_SERVERS $HOME_NET

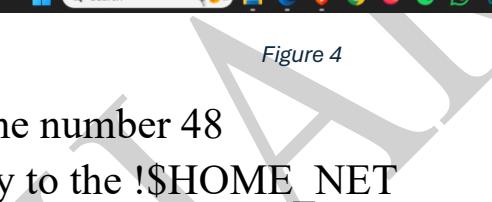
```

Properties file length: 26.838 lines: 690 Ln: 45 Col: 1 Sel: 29 | 1 Unix (LF) UTF-8 INS

ENG IN 24-12-2025 19:11

Figure 4

- Go to the line number 48
- Replace Any to the !\$HOME_NET



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

```

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
snort - Copy.conf & [x]
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 ##### Step #1: Set the network variables. For more information, see README.variables
41 #####
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 #####
59 # List of sql servers on your network
60 ipvar SQL_SERVERS $HOME_NET
61 #####
62 # List of telnet servers on your network
63 ipvar TELNET_SERVERS $HOME_NET

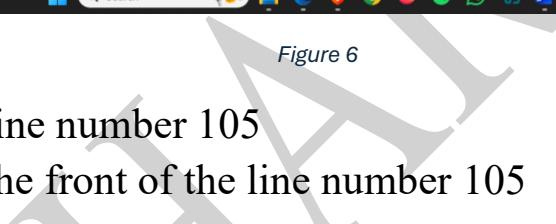
```

Properties file length: 26.838 lines: 690 Ln: 48 Col: 1 Sel: 29 | 1 Unix (LF) UTF-8 INS

ENG IN 24-12-2025 19:12

Figure 5

- Go to the line number 104
- Set rules folder location (C:\\Snort\\rules)



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

```

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 [$HTTP_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 [64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,205.188.5.0/24,205.188.7.0/24,2
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 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 appropriately
113 var WHITE_LIST_PATH C:\Snort\rules
114 var BLACK_LIST_PATH C:\Snort\rules
115
116 #####
117 # Step #2: Configure the decoder. For more information, see README.decode
118 #####
119
120 # Stop generic decode events:

```

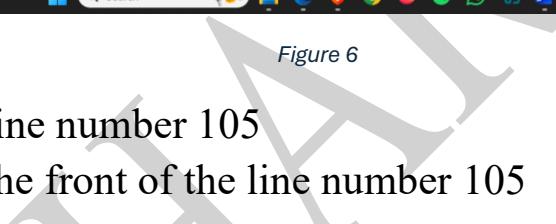
Properties file

length: 26,838 lines: 690 Ln: 104 Col: 1 Set: 29 | 1 Unix (LF) UTF-8 INS

ENG IN 24-12-2025 1914

Figure 6

- Go to the line number 105
- Add # on the front of the line number 105



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

```

91
92 # List of file data ports for file inspection
93 portvar FILE_DATA_PORTS [$HTTP_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 [64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,205.188.5.0/24,205.188.7.0/24,2
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 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 appropriately
113 var WHITE_LIST_PATH C:\Snort\rules
114 var BLACK_LIST_PATH C:\Snort\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

```

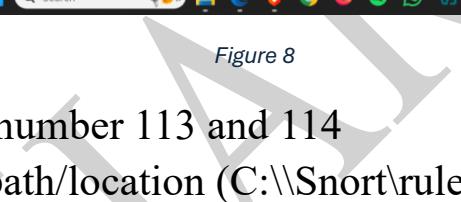
Properties file

length: 26,838 lines: 690 Ln: 105 Col: 1 Set: 29 | 1 Unix (LF) UTF-8 INS

ENG IN 24-12-2025 1915

Figure 7

- Go to the line number 106
- Set preproc_rules (C:\Snort\preproc_rules)



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

```

91 # List of file data ports for file inspection
92 portvar FILE_DATA_PORTS [$HTTP_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 [64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,205.188.5.0/24,205.188.7.0/24,2
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 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 appropriately
113 var WHITE_LIST_PATH C:\Snort\rules
114 var BLACK_LIST_PATH C:\Snort\rules
115
116 ##### Step #2: Configure the decoder. For more information, see README.decode
117 # Step #2: Configure the decoder. For more information, see README.decode
118 #####
119
120 # Stop generic decode events:
config disable_decode_alerts
122
123 # Stop Alerts on experimental TCP options
config disable_tcpopt_experimental_alerts
125
126 # Stop Alerts on obsolete TCP options
config disable_tcpopt_obsolete_alerts
128
129 # Stop Alerts on T/TCP alerts

```

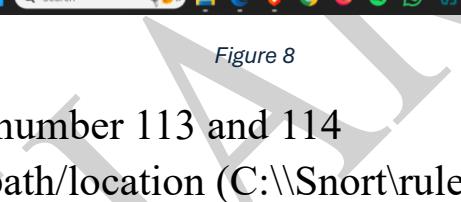
Properties file

length: 26,838 lines: 690 Ln: 106 Col: 1 Sel: 44 | 1 Unix (LF) UTF-8 INS

ENG IN 1917 24-12-2025

Figure 8

- Go to the line number 113 and 114
- add rules file path/location (C:\Snort\rules)



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

```

97 # other variables, these should not be modified
98 ipvar AIM_SERVERS [64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,205.188.5.0/24,205.188.7.0/24,2
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 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 appropriately
113 var WHITE_LIST_PATH C:\Snort\rules
114 var BLACK_LIST_PATH C:\Snort\rules
115
116 ##### Step #2: Configure the decoder. For more information, see README.decode
117 # Step #2: Configure the decoder. For more information, see README.decode
118 #####
119
120 # Stop generic decode events:
config disable_decode_alerts
122
123 # Stop Alerts on experimental TCP options
config disable_tcpopt_experimental_alerts
125
126 # Stop Alerts on obsolete TCP options
config disable_tcpopt_obsolete_alerts
128
129 # Stop Alerts on T/TCP alerts

```

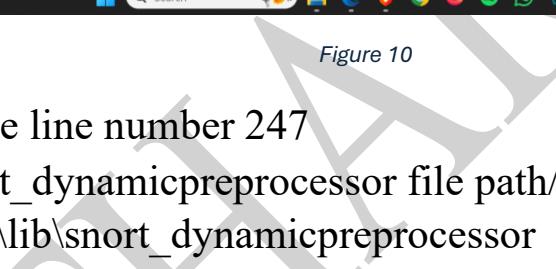
Properties file

length: 26,838 lines: 690 Ln: 113 Col: 1 Sel: 69 | 2 Unix (LF) UTF-8 INS

ENG IN 1919 24-12-2025

Figure 9

- Go to the line number 186
- Remove hash



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

```

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
snort - Copy.conf & [ ] 
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.

```

Properties file length: 26,838 lines: 690 Ln: 186 Col: 1 Set: 26 | 1 Unix (LF) UTF-8 INS

1921 ENG IN 24-12-2025

Figure 10

- Go to the line number 247
- Set snort_dynamicpreprocessor file path/location :-
c:\Snort\lib\snort_dynamicpreprocessor



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

```

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
snort - Copy.conf & [ ] 
232 #config profile_rules: print all, sort avg_ticks
233 #config profile_procs: 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 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 dynamicrules /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 TDS mode

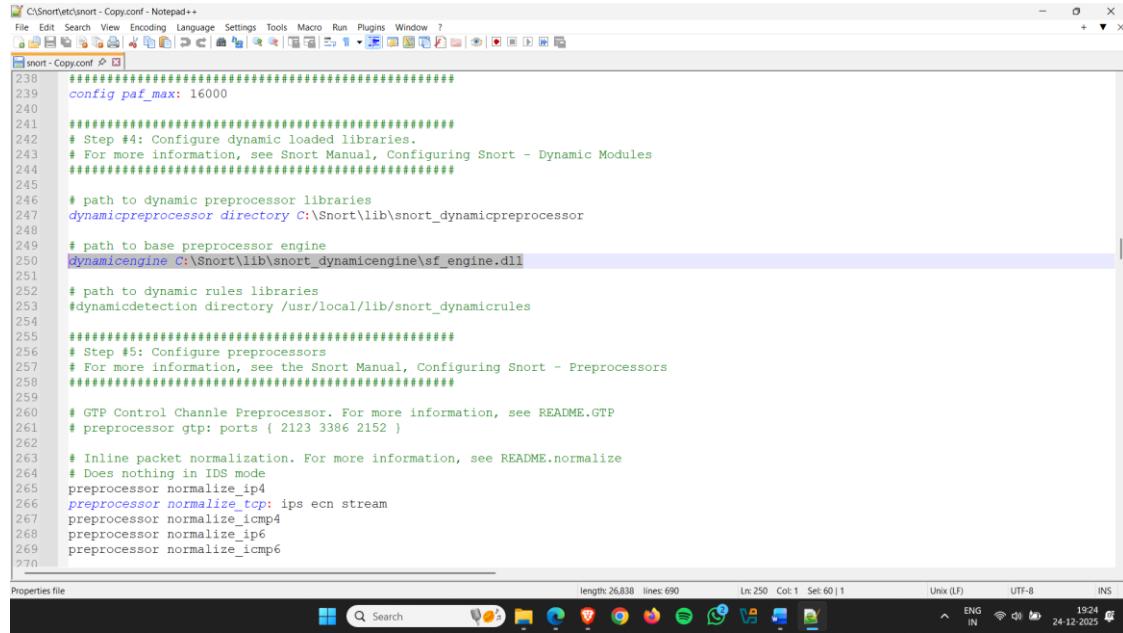
```

Properties file length: 26,838 lines: 690 Ln: 247 Col: 1 Set: 68 | 1 Unix (LF) UTF-8 INS

1922 ENG IN 24-12-2025

Figure 11

- Go to the line number number 250
- Set sf_engine.dll file location :-
c:\Snort\lib\snort_dynamicengine\sf_engine.dll



```

C:\Snort\etc\snort - Copy.conf - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
snort - Copy.conf [x]
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 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

```

Figure 12

- 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 named, blacklist.rules copy file and paste in same folder and rename copy file to the whitelist.rules.

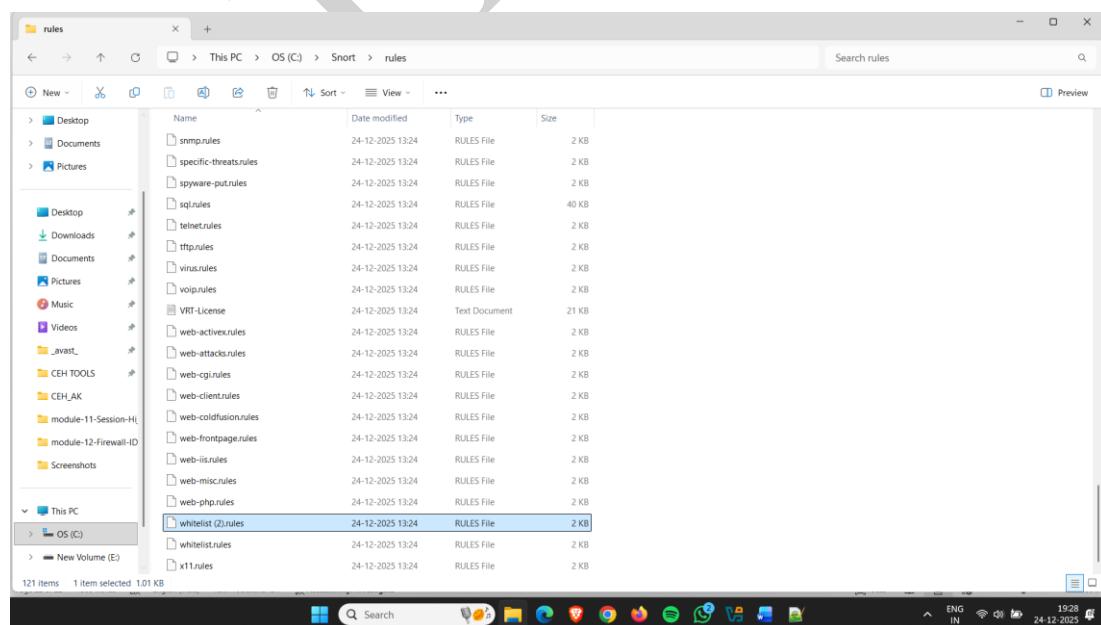


Figure 13

- Go to the line number 511 and 512
 - Replace white_list.rules to whitelist.rules
 - Replace black_list.rules to blacklist.rules

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

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

snort - Copy.conf

```
496     uu_decode_depth 0
497
498 # Modbus preprocessor. For more information see README.modbus
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/whitelist.rules, \
512     blacklist $BLACK_LIST_PATH/blacklist.rules
513
514 ######
515 # Step #6: Configure output plugins
516 # For more information, see Snort Manual, Configuring Snort - Output Modules
517 #####
518
519 # unified2
520 # Recommended for most installs
521 # output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_types
522
523 # Additional configuration for specific types of installs
524 # output alert_unified2: filename snort.alert, limit 128, nostamp
525 # output log_unified2: filename snort.log, limit 128, nostamp
526
527 # syslog
528 # output alert syslog: LOG_AUTH LOG ALERT
```

Figure 14

- Go to the line number 546
 - Replace “/” to “\” 546 upto 651
 - Press ctrl+f

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

snort - Copy (3).conf > x

```
535 include reference.config
536
537
538 #####
539 # Step #7: Customize your rule set
540 # For more information, see Snort Manual, Writing Snort Rules
541 #
542 # NOTE: All categories are enabled in this conf file
543 #####
544
545 # site specific rules
546 include $RULE_PATH/local.rules
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
```

Find

Find Replace Find in Files Find in Projects Mark

Find what:

Backward direction

In selection

Count

Find Next

Match whole word only

Match case

Find All in Current Document

Wrap around

Find Mode

Transparency

Normal

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

Regular expression matches newline

On losing focus

Always

Properties file

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

Figure 15

- Add “/” in first box
- Add “\” in second column & click on replace

```

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
snort - Copy.conf <-->
530 include $RULE_PATH\server-other.rules
531 include $RULE_PATH\server-webapp.rules
532 include $RULE_PATH\shellcode.rules
533 include $RULE_PATH\smtp.rules
534 include $RULE_PATH\snmp.rules
535 include $RULE_PATH\specific-threats.rules
536 include $RULE_PATH\spyware-put.rules
537 include $RULE_PATH\sql.rules
538 include $RULE_PATH\telnet.rules
539 include $RULE_PATH\tftp.rules
540 include $RULE_PATH\virus.rules
541 include $RULE_PATH\voip.rules
542 include $RULE_PATH\web-activex.rules
543 include $RULE_PATH\web-attacks.rules
544 include $RULE_PATH\web-cgi.rules
545 include $RULE_PATH\web-client.rules
546 include $RULE_PATH\web-coldfusion.rules
547 include $RULE_PATH\web-frontpage.rules
548 include $RULE_PATH\web-iis.rules
549 include $RULE_PATH\web-misc.rules
550 include $RULE_PATH\web-php.rules
551 include $RULE_PATH\xml.rules
552
#####
# Step #8: Customize your preprocessor and decoder alerts
# For more information, see README.decoder_preproc_rules
#####

# decoder and preprocessor event rules
# include $PREPROC_RULE_PATH/preprocessor.rules
# include $PREPROC_RULE_PATH/decoder.rules
# include $PREPROC_RULE_PATH/sensitive-data.rules
#####
# Step #9: Customize your Shared Object Snort Rules
#####

Properties file
length: 26.045 lines: 690 Ln: 654 Col: 1 Sel: 57 | 1 Unix (LF) UTF-8 INS
Heavy rain ENG 19:52:54

```

Figure 16

- All the slash are being replaced.

```

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
snort - Copy.conf <-->
544
545 # site specific rules
546 include $RULE_PATH\local.rules
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\dos.rules
563 include $RULE_PATH\ddos.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 include $RULE_PATH\file-identify.rules
570 include $RULE_PATH\file-image.rules
571 include $RULE_PATH\file-multimedia.rules
572 include $RULE_PATH\file-office.rules
573 include $RULE_PATH\file-other.rules
574 include $RULE_PATH\file-pdf.rules
575
576 include $RULE_PATH\finger.rules

Properties file
length: 26.038 lines: 690 Ln: 575 Col: 34 Sel: 1,055 | 30 Unix (LF) UTF-8 INS
ENG 19:33 24-12-2025

```

Figure 17

- Open CMD & do run as administrator

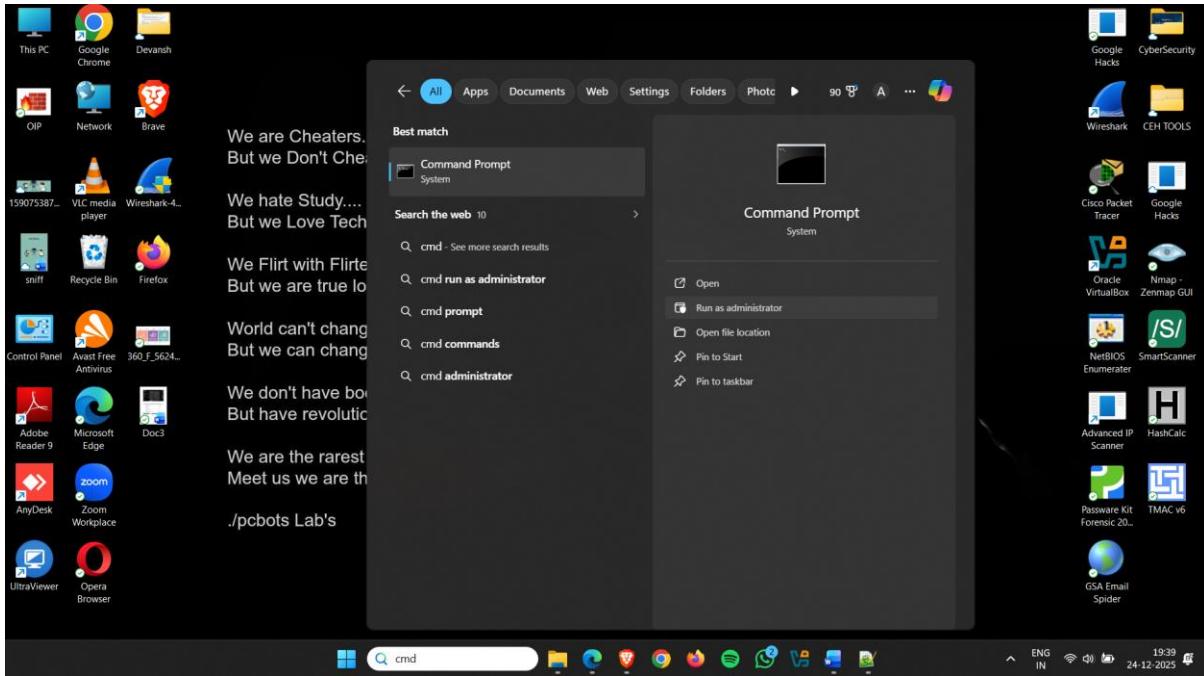


Figure 18

- Go to the Snort folder

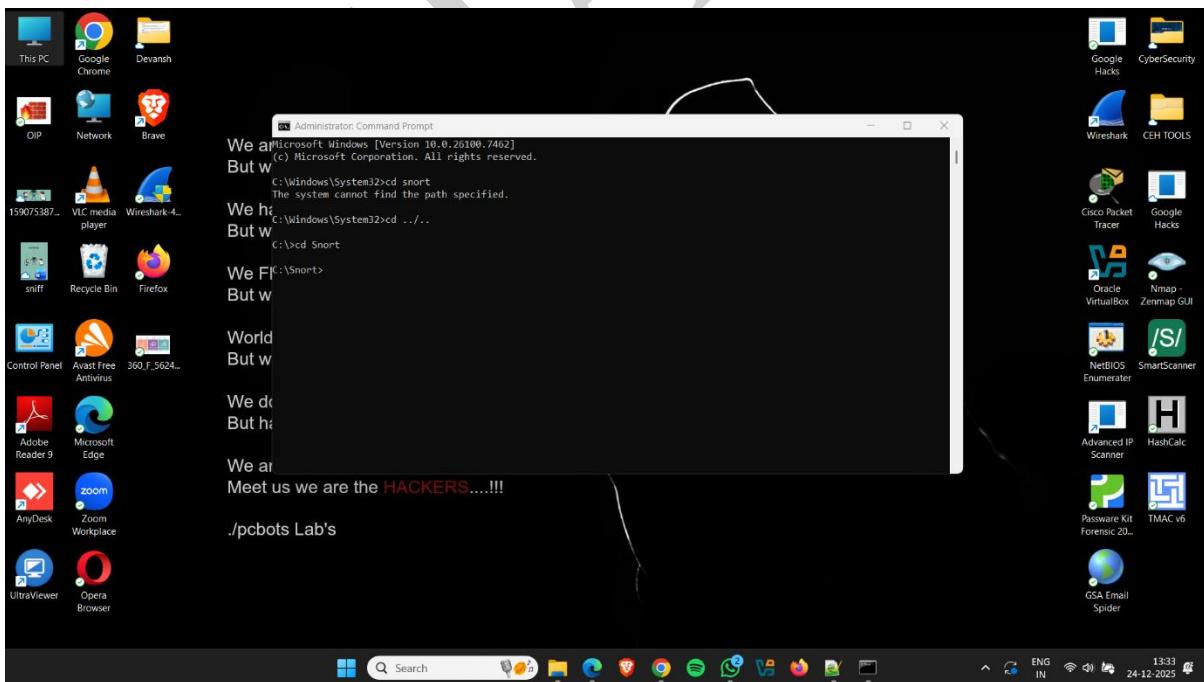


Figure 19

- Go to the Snort\bin Folder

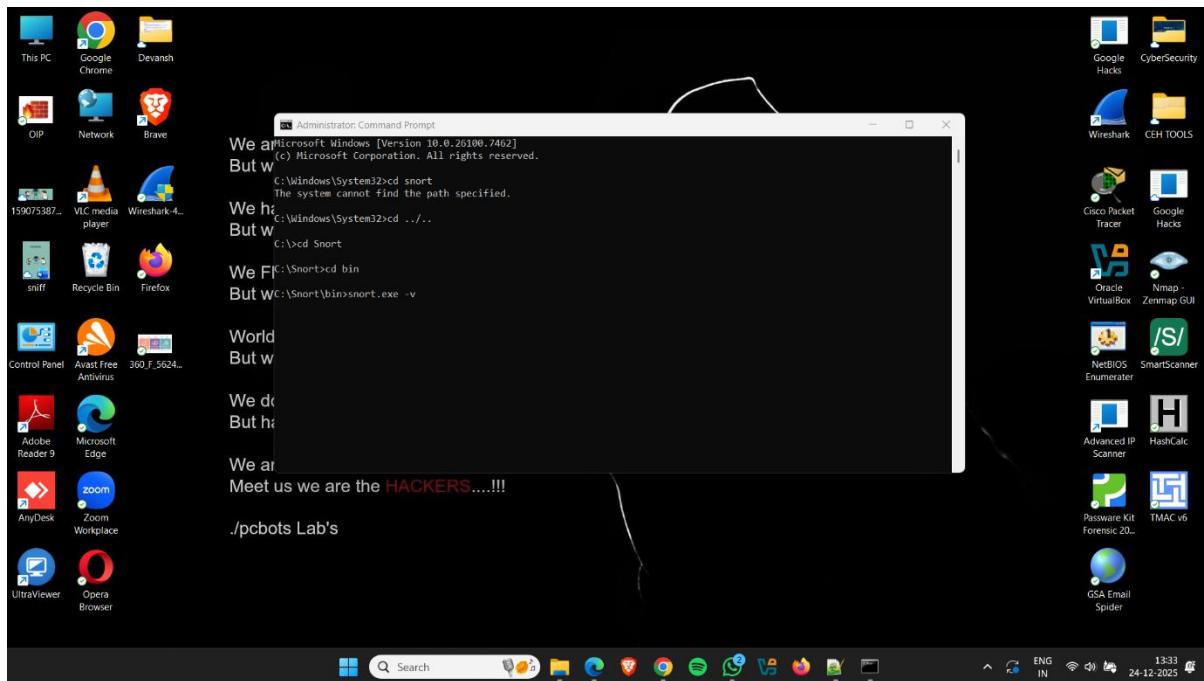


Figure 20

- Run file snort.exe -v (v for version)

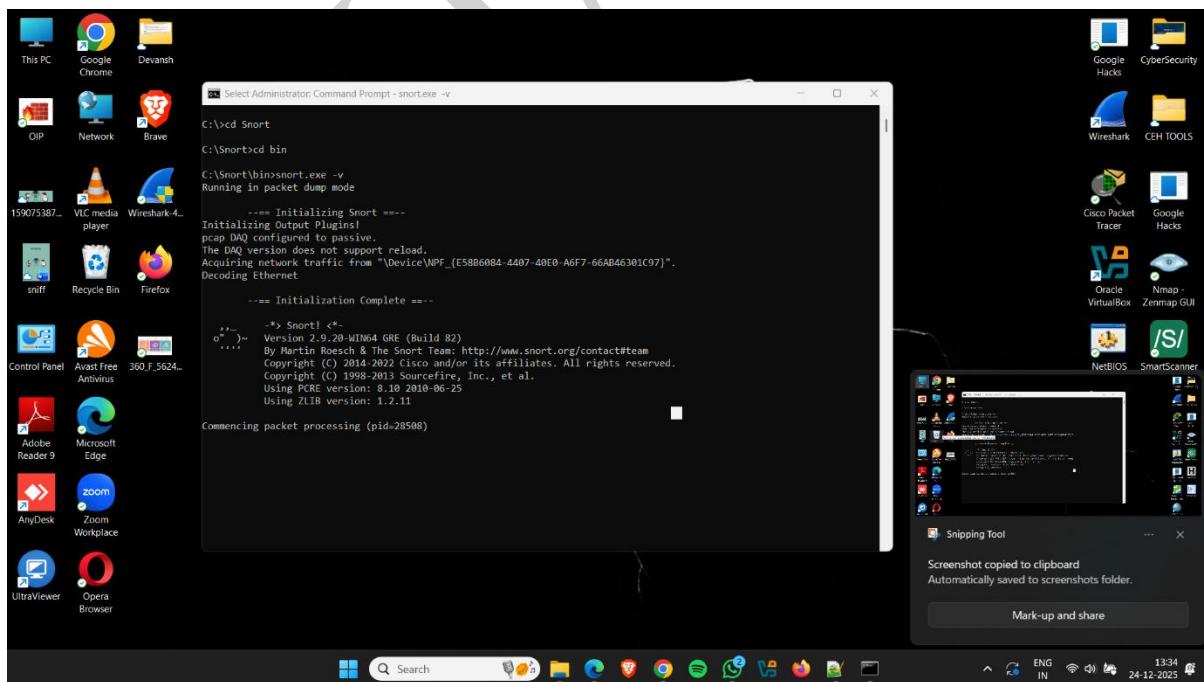


Figure 21

- Use next command for finding network interface
(snort.exe -W)

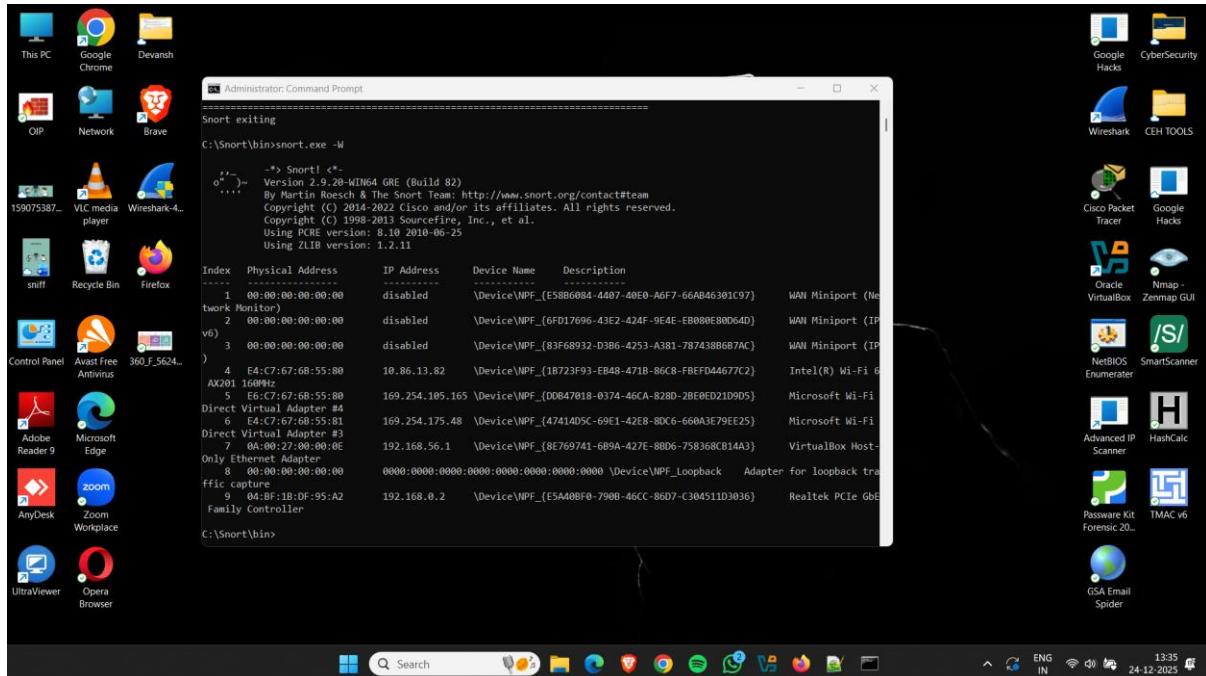


Figure 22

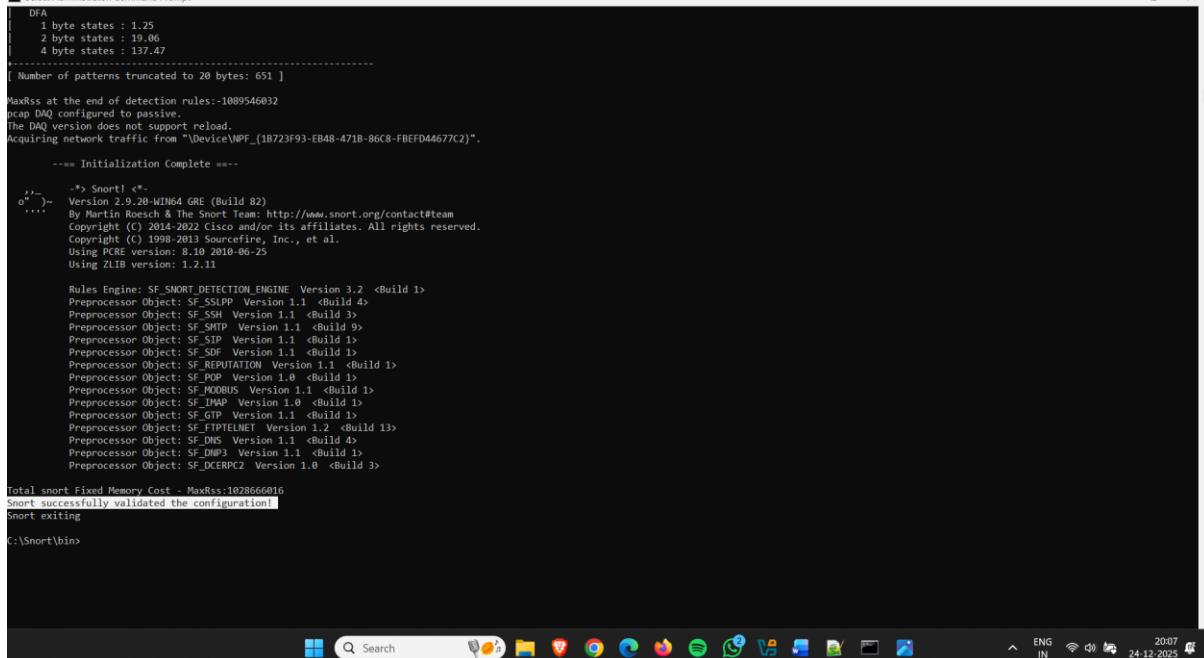
- Command for configuration testing :-
snort.exe -i 4 -c "c:\Snort\etc\file name" -T

```
Administrator: Command Prompt
C:\Snort\bin>snort.exe -i 4 -c "c:\Snort\etc\snort - Copy.conf" -T
Running in Test mode

    === Initializing Snort ===
Initializing Output Plugins!
initializing Preprocessors!
initializing Plug-ins!
Parsin Rules file "c:\Snort\etc\snort - Copy.conf"
PortVar 'HTTP_PORTS' defined : [ 80:81 110 143 311 383 591 593 901 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 8088 8090 8118 8123 8130:8131 8243 8280 8300 8808 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555 ]
PortVar 'TELNET_PORTS' defined : [ 1024:1025 1029:1030 1048:1049 1053:1054 1067:1068 1073:1074 1081:1082 1093:1094 1103:1104 1113:1114 1123:1124 1133:1134 1143:1144 1153:1154 1163:1164 1173:1174 1183:1184 1193:1194 1203:1204 1213:1214 1223:1224 1233:1234 1243:1244 1253:1254 1263:1264 1273:1274 1283:1284 1293:1294 1303:1304 1313:1314 1323:1324 1333:1334 1343:1344 1353:1354 1363:1364 1373:1374 1383:1384 1393:1394 1403:1404 1413:1414 1423:1424 1433:1434 1443:1444 1453:1454 1463:1464 1473:1474 1483:1484 1493:1494 1503:1504 1513:1514 1523:1524 1533:1534 1543:1544 1553:1554 1563:1564 1573:1574 1583:1584 1593:1594 1603:1604 1613:1614 1623:1624 1633:1634 1643:1644 1653:1654 1663:1664 1673:1674 1683:1684 1693:1694 1703:1704 1713:1714 1723:1724 1733:1734 1743:1744 1753:1754 1763:1764 1773:1774 1783:1784 1793:1794 1803:1804 1813:1814 1823:1824 1833:1834 1843:1844 1853:1854 1863:1864 1873:1874 1883:1884 1893:1894 1903:1904 1913:1914 1923:1924 1933:1934 1943:1944 1953:1954 1963:1964 1973:1974 1983:1984 1993:1994 2003:2004 2013:2014 2023:2024 2033:2034 2043:2044 2053:2054 2063:2064 2073:2074 2083:2084 2093:2094 2103:2104 2113:2114 2123:2124 2133:2134 2143:2144 2153:2154 2163:2164 2173:2174 2183:2184 2193:2194 2203:2204 2213:2214 2223:2224 2233:2234 2243:2244 2253:2254 2263:2264 2273:2274 2283:2284 2293:2294 2303:2304 2313:2314 2323:2324 2333:2334 2343:2344 2353:2354 2363:2364 2373:2374 2383:2384 2393:2394 2403:2404 2413:2414 2423:2424 2433:2434 2443:2444 2453:2454 2463:2464 2473:2474 2483:2484 2493:2494 2503:2504 2513:2514 2523:2524 2533:2534 2543:2544 2553:2554 2563:2564 2573:2574 2583:2584 2593:2594 2603:2604 2613:2614 2623:2624 2633:2634 2643:2644 2653:2654 2663:2664 2673:2674 2683:2684 2693:2694 2703:2704 2713:2714 2723:2724 2733:2734 2743:2744 2753:2754 2763:2764 2773:2774 2783:2784 2793:2794 2803:2804 2813:2814 2823:2824 2833:2834 2843:2844 2853:2854 2863:2864 2873:2874 2883:2884 2893:2894 2903:2904 2913:2914 2923:2924 2933:2934 2943:2944 2953:2954 2963:2964 2973:2974 2983:2984 2993:2994 2006:2007 2016:2017 2026:2027 2036:2037 2046:2047 2056:2057 2066:2067 2076:2077 2086:2087 2096:2097 2106:2107 2116:2117 2126:2127 2136:2137 2146:2147 2156:2157 2166:2167 2176:2177 2186:2187 2196:2197 2206:2207 2216:2217 2226:2227 2236:2237 2246:2247 2256:2257 2266:2267 2276:2277 2286:2287 2296:2297 2306:2307 2316:2317 2326:2327 2336:2337 2346:2347 2356:2357 2366:2367 2376:2377 2386:2387 2396:2397 2406:2407 2416:2417 2426:2427 2436:2437 2446:2447 2456:2457 2466:2467 2476:2477 2486:2487 2496:2497 2506:2507 2516:2517 2526:2527 2536:2537 2546:2547 2556:2557 2566:2567 2576:2577 2586:2587 2596:2597 2606:2607 2616:2617 2626:2627 2636:2637 2646:2647 2656:2657 2666:2667 2676:2677 2686:2687 2696:2697 2706:2707 2716:2717 2726:2727 2736:2737 2746:2747 2756:2757 2766:2767 2776:2777 2786:2787 2796:2797 2806:2807 2816:2817 2826:2827 2836:2837 2846:2847 2856:2857 2866:2867 2876:2877 2886:2887 2896:2897 2906:2907 2916:2917 2926:2927 2936:2937 2946:2947 2956:2957 2966:2967 2976:2977 2986:2987 2996:2997 ] PortVar 'ORACLE_PORTS' defined : [ 1024:1025 1029:1030 1048:1049 1053:1054 1067:1068 1073:1074 1081:1082 1093:1094 1103:1104 1113:1114 1123:1124 1133:1134 1143:1144 1153:1154 1163:1164 1173:1174 1183:1184 1193:1194 1203:1204 1213:1214 1223:1224 1233:1234 1243:1244 1253:1254 1263:1264 1273:1274 1283:1284 1293:1294 1303:1304 1313:1314 1323:1324 1333:1334 1343:1344 1353:1354 1363:1364 1373:1374 1383:1384 1393:1394 1403:1404 1413:1414 1423:1424 1433:1434 1443:1444 1453:1454 1463:1464 1473:1474 1483:1484 1493:1494 1503:1504 1513:1514 1523:1524 1533:1534 1543:1544 1553:1554 1563:1564 1573:1574 1583:1584 1593:1594 1603:1604 1613:1614 1623:1624 1633:1634 1643:1644 1653:1654 1663:1664 1673:1674 1683:1684 1693:1694 1703:1704 1713:1714 1723:1724 1733:1734 1743:1744 1753:1754 1763:1764 1773:1774 1783:1784 1793:1794 1803:1804 1813:1814 1823:1824 1833:1834 1843:1844 1853:1854 1863:1864 1873:1874 1883:1884 1893:1894 1903:1904 1913:1914 1923:1924 1933:1934 1943:1944 1953:1954 1963:1964 1973:1974 1983:1984 1993:1994 2003:2004 2013:2014 2023:2024 2033:2034 2043:2044 2053:2054 2063:2064 2073:2074 2083:2084 2093:2094 2103:2104 2113:2114 2123:2124 2133:2134 2143:2144 2153:2154 2163:2164 2173:2174 2183:2184 2193:2194 2203:2204 2213:2214 2223:2224 2233:2234 2243:2244 2253:2254 2263:2264 2273:2274 2283:2284 2293:2294 2303:2304 2313:2314 2323:2324 2333:2334 2343:2344 2353:2354 2363:2364 2373:2374 2383:2384 2393:2394 2403:2404 2413:2414 2423:2424 2433:2434 2443:2444 2453:2454 2463:2464 2473:2474 2483:2484 2493:2494 2503:2504 2513:2514 2523:2524 2533:2534 2543:2544 2553:2554 2563:2564 2573:2574 2583:2584 2593:2594 2603:2604 2613:2614 2623:2624 2633:2634 2643:2644 2653:2654 2663:2664 2673:2674 2683:2684 2693:2694 2703:2704 2713:2714 2723:2724 2733:2734 2743:2744 2753:2754 2763:2764 2773:2774 2783:2784 2793:2794 2803:2804 2813:2814 2823:2824 2833:2834 2843:2844 2853:2854 2863:2864 2873:2874 2883:2884 2893:2894 2903:2904 2913:2914 2923:2924 2933:2934 2943:2944 2953:2954 2963:2964 2973:2974 2983:2984 2993:2994 2006:2007 2016:2017 2026:2027 2036:2037 2046:2047 2056:2057 2066:2067 2076:2077 2086:2087 2096:2097 2106:2107 2116:2117 2126:2127 2136:2137 2146:2147 2156:2157 2166:2167 2176:2177 2186:2187 2196:2197 2206:2207 2216:2217 2226:2227 2236:2237 2246:2247 2256:2257 2266:2267 2276:2277 2286:2287 2296:2297 2306:2307 2316:2317 2326:2327 2336:2337 2346:2347 2356:2357 2366:2367 2376:2377 2386:2387 2396:2397 2406:2407 2416:2417 2426:2427 2436:2437 2446:2447 2456:2457 2466:2467 2476:2477 2486:2487 2496:2497 2506:2507 2516:2517 2526:2527 2536:2537 2546:2547 2556:2557 2566:2567 2576:2577 2586:2587 2596:2597 2606:2607 2616:2617 2626:2627 2636:2637 2646:2647 2656:2657 2666:2667 2676:2677 2686:2687 2696:2697 2706:2707 2716:2717 2726:2727 2736:2737 2746:2747 2756:2757 2766:2767 2776:2777 2786:2787 2796:2797 2806:2807 2816:2817 2826:2827 2836:2837 2846:2847 2856:2857 2866:2867 2876:2877 2886:2887 2896:2897 2906:2907 2916:2917 2926:2927 2936:2937 2946:2947 2956:2957 2966:2967 2976:2977 2986:2987 2996:2997 ] PortVar 'SSH_PORTS' defined : [ 22 ] PortVar 'FTP_PORTS' defined : [ 21 2100 3535 ] PortVar 'SIP_PORTS' defined : [ 5060:5061 5600 ] PortVar 'FILE_DATA_PORTS' defined : [ 80:81 110 143 311 383 591 593 901 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 8088 8090 8118 8123 8130:8131 8243 8280 8300 8808 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555 ] PortVar 'GTP_PORTS' defined : [ 2123 2125 3386 ] Detection: Search Method = AC-Hall-Q Smart Any/Any group = enabled Search Method Optimizations = enabled Maximum pattern length = 20 Tagged Packet Limit: 256 Loading dynamic engine C:\Snort\lib\snort_dynamicengine\sf_engine.dll... done Loading all dynamic preprocessor libs from C:\Snort\lib\snort_dynamicpreprocessor... Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_dce2.dll... done Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_dmp.dll... done Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_dns.dll... done Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_ftptelnet.dll... done Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_icmp.dll... done Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imap.dll... done Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imd.dll... done Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imd.pop.dll... done Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imd.reputation.dll... done Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imd.sdf.dll... done Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imd.sip.dll... done Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imd.smtp.dll... done Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imd.ssl.dll... done Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imd.tcp.dll... done Finished loading all dynamic preprocessor libs from C:\Snort\lib\snort_dynamicpreprocessor Log directory = C:\Snort\log WARNING: ip4 normalizations disabled because not inline. WARNING: tcp normalization disabled because not inline. WARNING: icmp4 normalizations disabled because not inline. WARNING: ip6 normalizations disabled because not inline. WARNING: icmp6 normalizations disabled because not inline. Frag3 global config:
```

Figure 23

- Snort successfully validated the configuration!



```
DFA
1 byte states : 1.25
2 byte states : 19.06
4 byte states : 137.47
-----
[ Number of patterns truncated to 20 bytes: 651 ]

MaxRss at the end of detection rules:-1089546032
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "DeviceWPF_{1B723F93-E848-471B-86C8-BEFD44677C2}".

==== Initialization Complete ====
.*> Snort! <*
o"--> 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

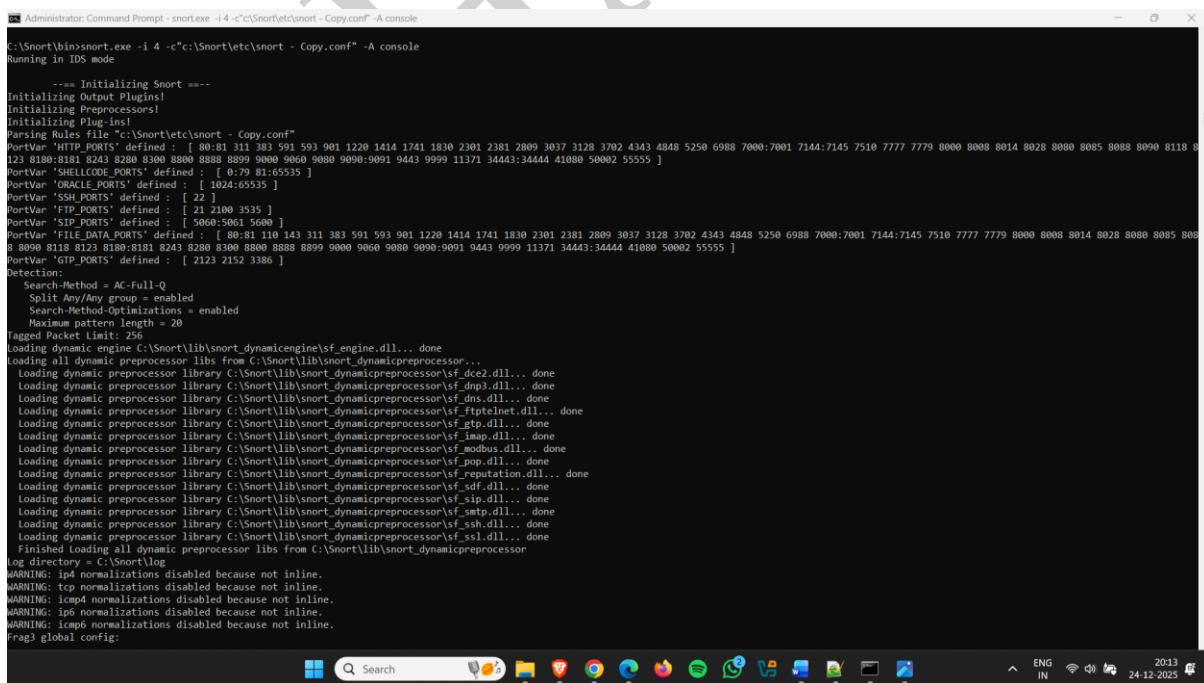
Rules Engine: SF_SNORT_DETECTION_ENGINE Version 3.2 <Build 1>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_TFTP Version 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_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_DCEP_C2 Version 1.0 <Build 3>

Total snort Fixed Memory Cost = MaxRss:1028666016
Snort successfully validated the configuration!
Snort exiting

C:\Snort\bin>
```

Figure 24

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

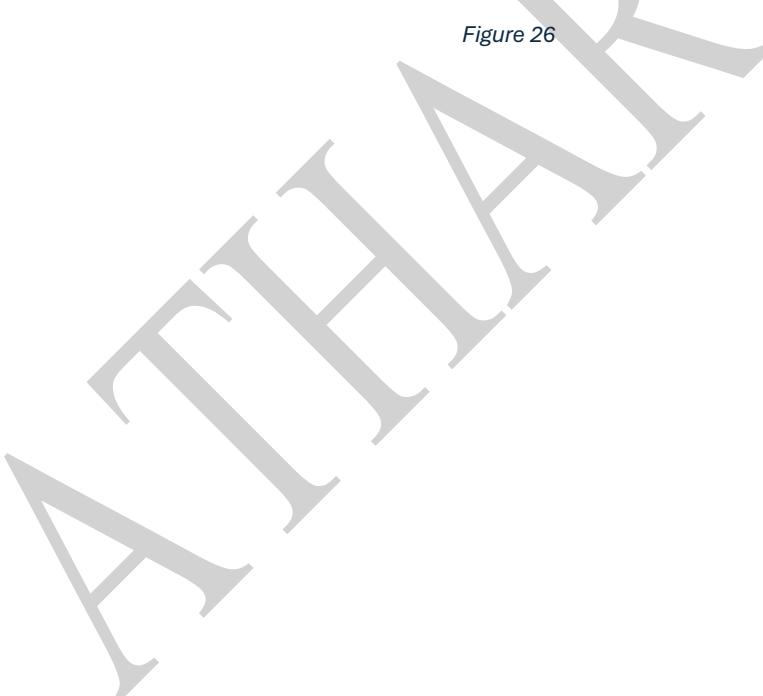


```
Administrator: Command Prompt - snort.exe -i 4 -c"c:\Snort\etc\snort - Copy.conf" -A console
Running in IDS mode

==== Initializing Snort ====
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plugins!
Parsing Rules file "c:\Snort\etc\snort - Copy.conf"
PortVar 'HTTP_PORTS' defined : [ 80:81 113 311 383 591 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5258 6988 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8090 8118 8123 8180:8181 8243 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555 ]
PortVar 'SHLLCODE_PORTS' defined : [ 1024:65535 ]
PortVar 'ORACLE_PORTS' defined : [ 22 ]
PortVar 'SSH_PORTS' defined : [ 22 ]
PortVar 'IMAP_PORTS' defined : [ 22 ]
PortVar 'SIP_PORTS' defined : [ 5060:5061 5080 ]
PortVar 'GTP_PORTS' defined : [ 2123 2125 3386 ]
Detection:
Search-Method = AC-Full-Q
Split Any/Any group = enabled
Search-Method-Optimizations = enabled
Maximum pattern length = 20
Tagged4 Packet Limit = 250
Loading dynamic engine C:\Snort\lib\snort\dynamicengine\sf_engine.dll... done
Loading all dynamic preprocessor libs from C:\Snort\lib\snort_dynamicpreprocessor...
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_dce2.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_dnp3.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_dnp4.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_ftptelnet.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_gtp.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imap.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg2.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg3.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg4.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg5.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg6.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg7.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg8.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg9.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg10.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg11.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg12.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg13.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg14.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg15.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg16.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg17.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg18.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg19.dll... done
Loading dynamic preprocessor library C:\Snort\lib\snort_dynamicpreprocessor\sf_imsg20.dll... done
Finished loading all dynamic preprocessor libs from C:\Snort\lib\snort_dynamicpreprocessor
Log directory = C:\Snort\log
WARNING: ip4 normalization disabled because not inline.
WARNING: ip6 normalization disabled because not inline.
WARNING: icmp4 normalizations disabled because not inline.
WARNING: icmp6 normalizations disabled because not inline.
WARNING: icmp6 normalizations disabled because not inline.
Frag3 global config:
```

Figure 25

- Started capturing network traffic



```

Administrator: Command Prompt - snorterw -i 4 <"c:\Snort\etc\snort - Copy 3.conf"> -A console
snmp DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "DeviceNPF_{27C3CSA4-5E52-465B-90A6-45345CB8BAED}".
Decoding Ethernet
==== Initialization Complete ====
-> Snort! <-
Version 2.9.20-WIN64 GRE (Build R2)
o/...)- 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_SSLLPP Version 1.1 <Build 4>
Preprocessor Object: SF_SMB Version 1.1 <Build 1>
Preprocessor Object: SF_SMB2 Version 1.1 <Build 1>
Preprocessor Object: SF_STP 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_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_DCERPC Version 1.0 <Build 3>
Commencing socket processing (pid=15468)
05/15-19:57:18.9880537 [*] [129:12:2] Consecutive TCP small segments exceeding threshold [**] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 20.190.146.35:443 -> 192.168.251.254:54672
05/15-19:57:11.791631 [*] [129:15:2] Reset outside window [**] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:50590 -> 172.64.158.5:443
05/15-19:57:32.725372 [*] [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] 192.168.251.192:52878 -> 172.64.158.5:443
05/15-19:58:08.699373 [*] [129:15:2] Reset outside window [**] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:33980 -> 104.18.37.251:443
05/15-19:58:08.657150 [*] [129:15:2] Reset outside window [**] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:52880 -> 172.64.158.5:443
05/15-19:58:18.819480 [*] [129:15:2] Reset outside window [**] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:35912 -> 172.64.158.5:443
05/15-19:58:19.429279 [*] [129:15:2] Reset outside window [**] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:48296 -> 104.18.37.251:443
05/15-19:58:20.164293 [*] [129:15:2] Reset outside window [**] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:48308 -> 104.18.37.251:443
05/15-19:58:21.107283 [*] [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
02
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
02
05/15-19:58:22.160659 [*] [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
03
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
03

```

Figure 26

Windows Firewall Configuration

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

Purpose of Windows Firewall

The primary purpose of Windows Firewall is to prevent unauthorized access to or from a private network by acting as a protective barrier between the computer and external threats such as hackers, malware, and unauthorized applications.

Key Features of Windows Firewall

1. Inbound and Outbound Filtering

- Blocks or allows network traffic based on security rules.
- **Inbound traffic:** Data coming into the computer.
- **Outbound traffic:** Data leaving the computer.

2. Predefined Security Rules

- Automatically configures security rules for commonly used applications and system services.
- Reduces the need for manual configuration.

3. Application Control

- Prompts the user when an unknown or unauthorized application attempts to access the network.
- Allows users to permit or block applications as needed.

4. Network Profiles

Windows Firewall allows different security settings based on the network type:

- **Private Network:** Used for home or trusted networks.
- **Public Network:** Used for unsecured networks such as public Wi-Fi.
- **Domain Network:** Used in corporate or organizational environments.

5. Integration with Windows Security Center

- Can be easily managed through:
 - Control Panel
 - Windows Security Settings
- Provides centralized security management.

6. Logging and Monitoring

- Maintains logs of blocked packets and connection attempts.
- Helps administrators monitor suspicious activity and troubleshoot network issues.

(INBOUND RULES)

- Click on Windows Button and search Firewall and open.

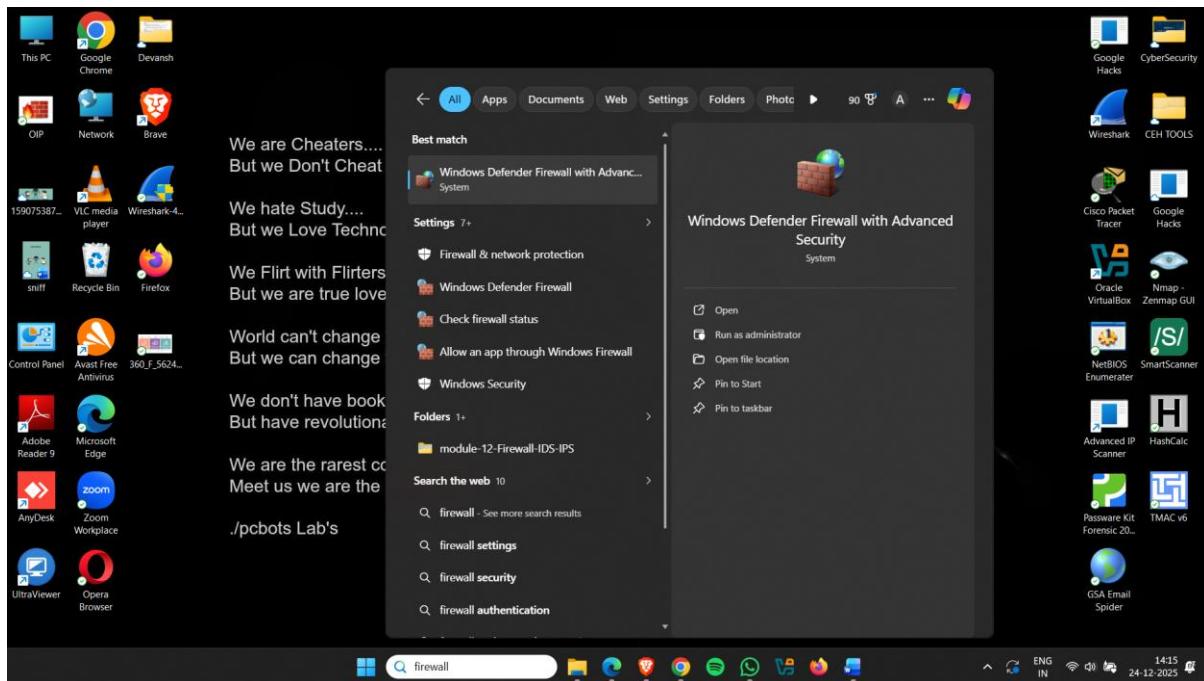


Figure 27

- There is option inbound rules – it means setting rules for incoming network.

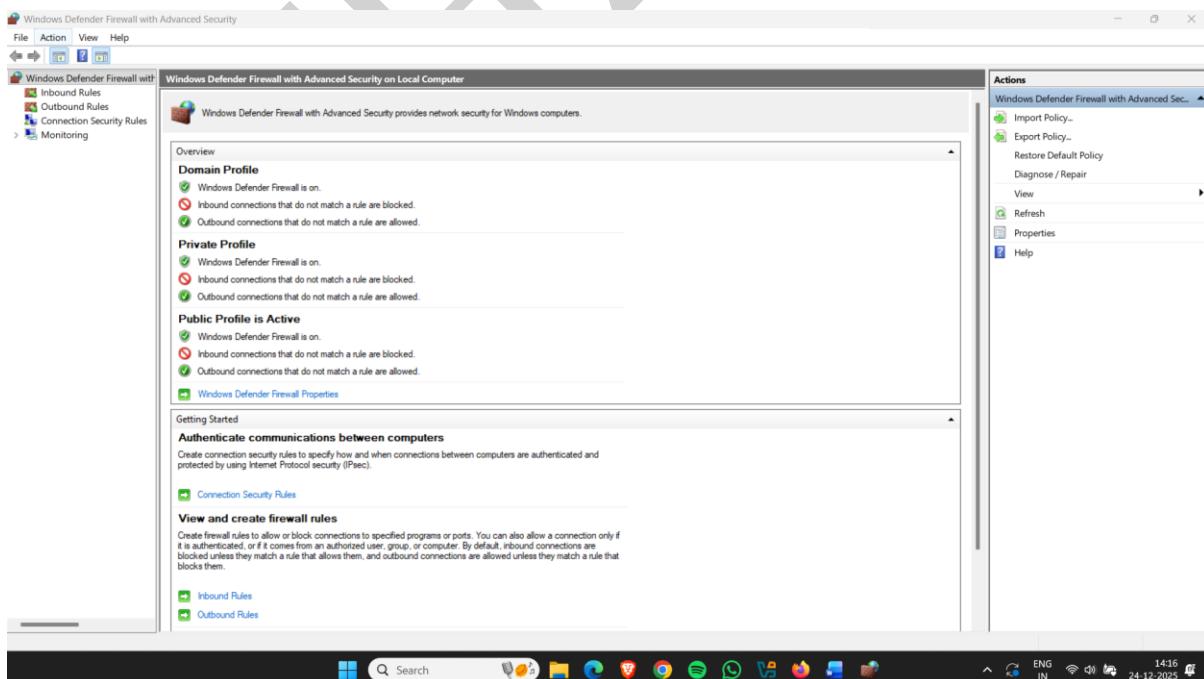


Figure 28

- Click on Inbound rules then click on new rule

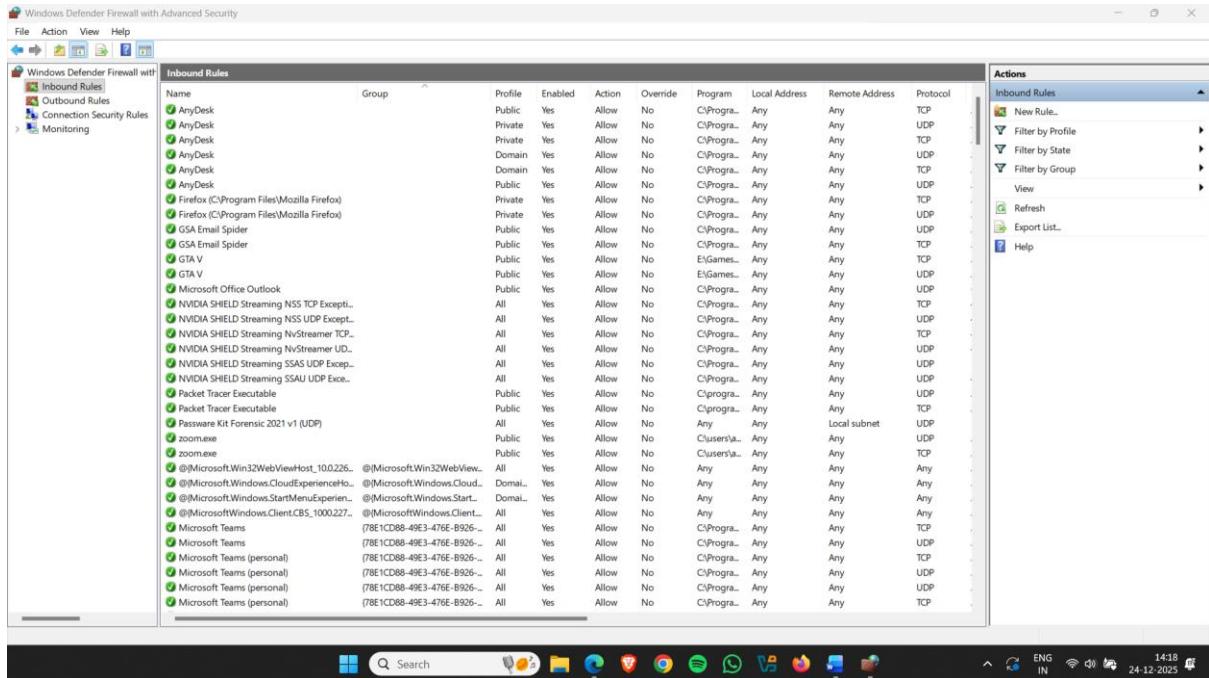


Figure 29

- Select the option that you want to set configuration like particular port or program , and then click on next.

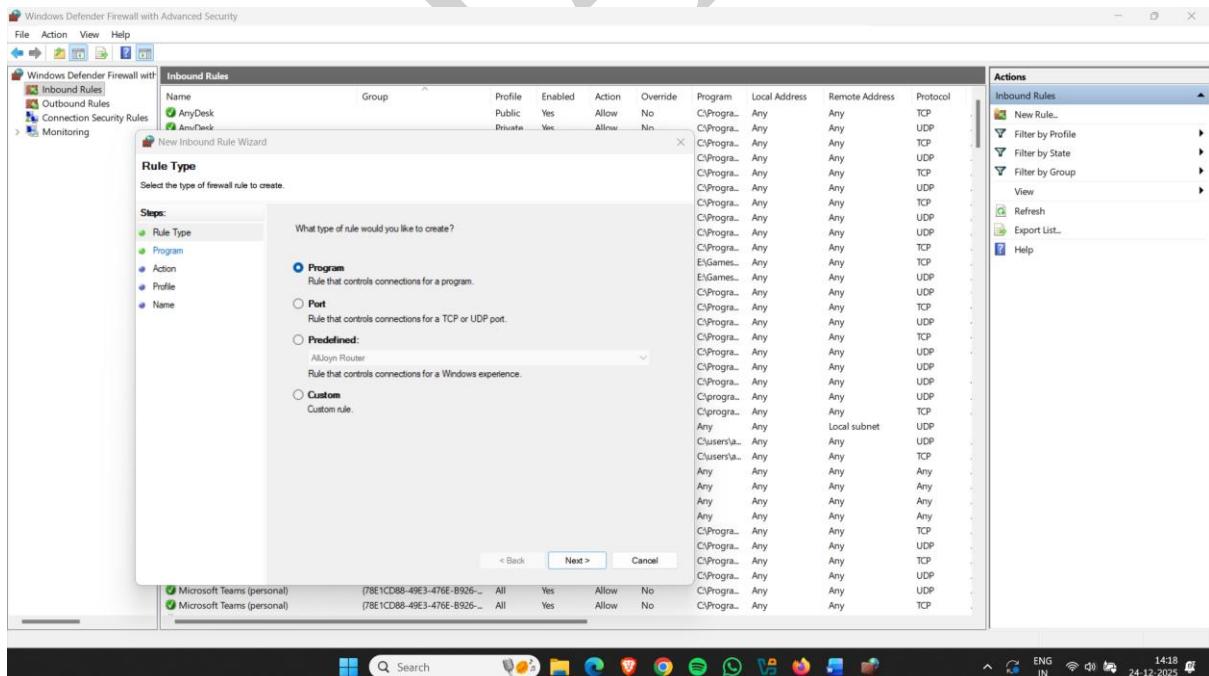


Figure 30

- Then set a path of program
Select a file and click on open

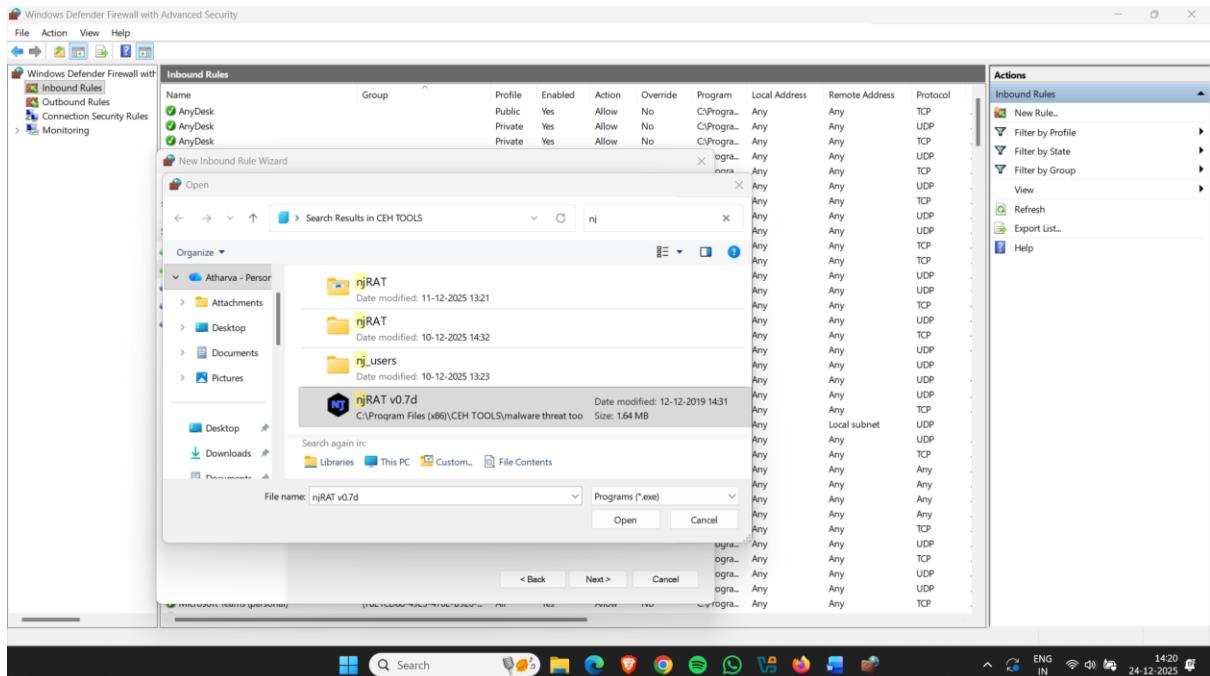


Figure 31

- Click on next

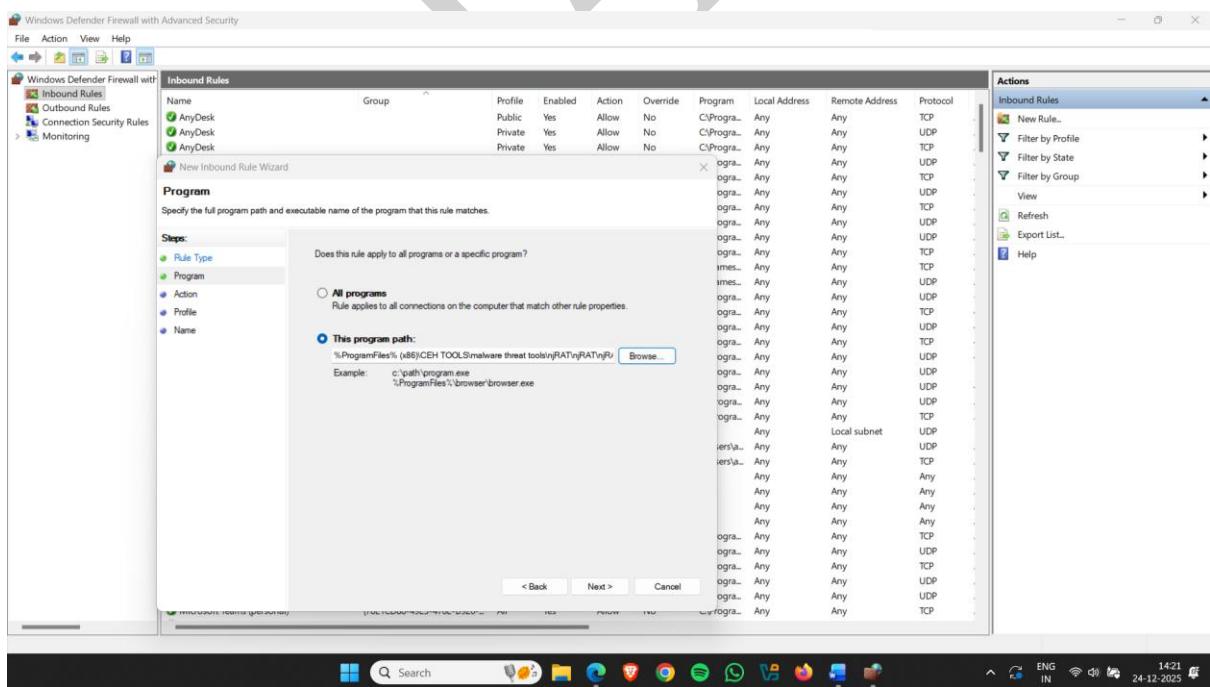


Figure 32

- **Connection Action Options**

There are three available options to control how incoming connections are handled:

1. Allow the Connection

Description:

Allows all matching incoming traffic, whether the connection is secured or not.

Security:

- No restriction on authentication or encryption
- Unsecured traffic is also permitted

Use When:

- You trust the application, port, or IP address
-

2. Allow the Connection if It Is Secure

Description:

Allows only those connections that are authenticated using **IPsec (Internet Protocol Security)**.

Requires:

- Proper configuration of IPsec policies
- IPsec must be enabled and correctly set up

Security:

- Ensures encrypted and authenticated communication
- Blocks unsecured connections

Use When:

- Protecting sensitive systems or confidential data

Note:

- The **Customize** button remains greyed out unless IPsec is configured

3. Block the Connection

Description:

Completely denies the connection, regardless of whether it is secure or unsecured.

Security:

- Strictest security setting
- No traffic is allowed

Use When:

- You want to prevent any communication through specific ports, programs, or IP addresses
- Select an option & then click on next

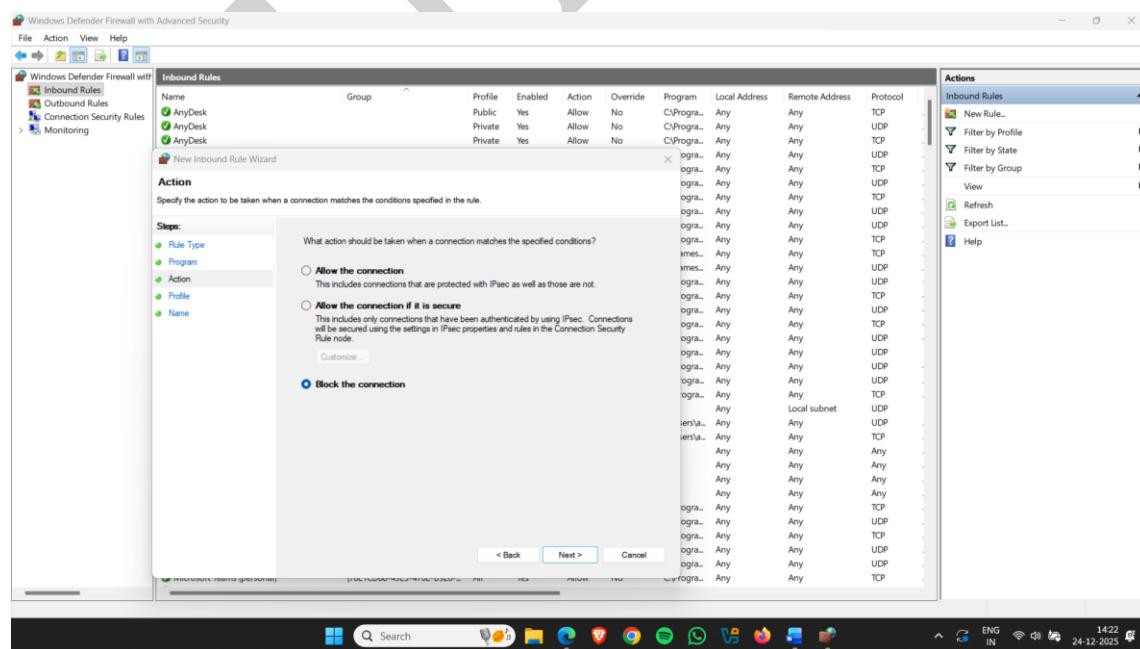


Figure 33

- **Network Profile Options**

Again, three network profile options appear. These determine **when the firewall rule will be applied** based on the type of network the system is connected to.

1. Domain

Applies When:

- The system is connected to a corporate or organizational network
- The computer is joined to a **domain** (e.g., office or enterprise environment using **Active Directory**)

Use Case:

- Apply this rule when you want it to work only in secure, managed networks
 - Suitable for enterprise-level security policies
-

2. Private

Applies When:

- The computer is connected to a trusted private network
- Examples include home Wi-Fi or a small office network

Use Case:

- Enable this rule when you are at home or on a network you trust
 - Allows flexibility while maintaining security
-

3. Public

Applies When:

- The system is connected to a public network
- Examples include cafés, airports, hotels, or other open networks

Use Case:

- Apply the rule when using public networks
- Extra caution is required because public networks are less secure

- Select an option & then click on next

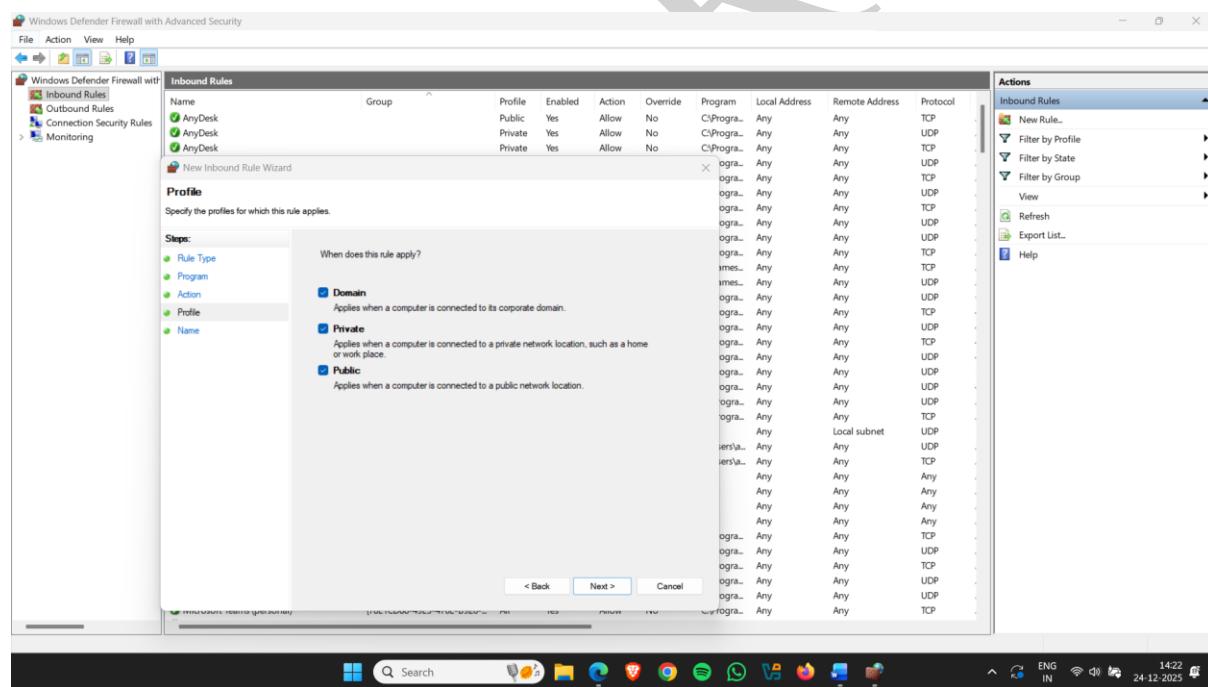


Figure 34

- Set any name and click on finish

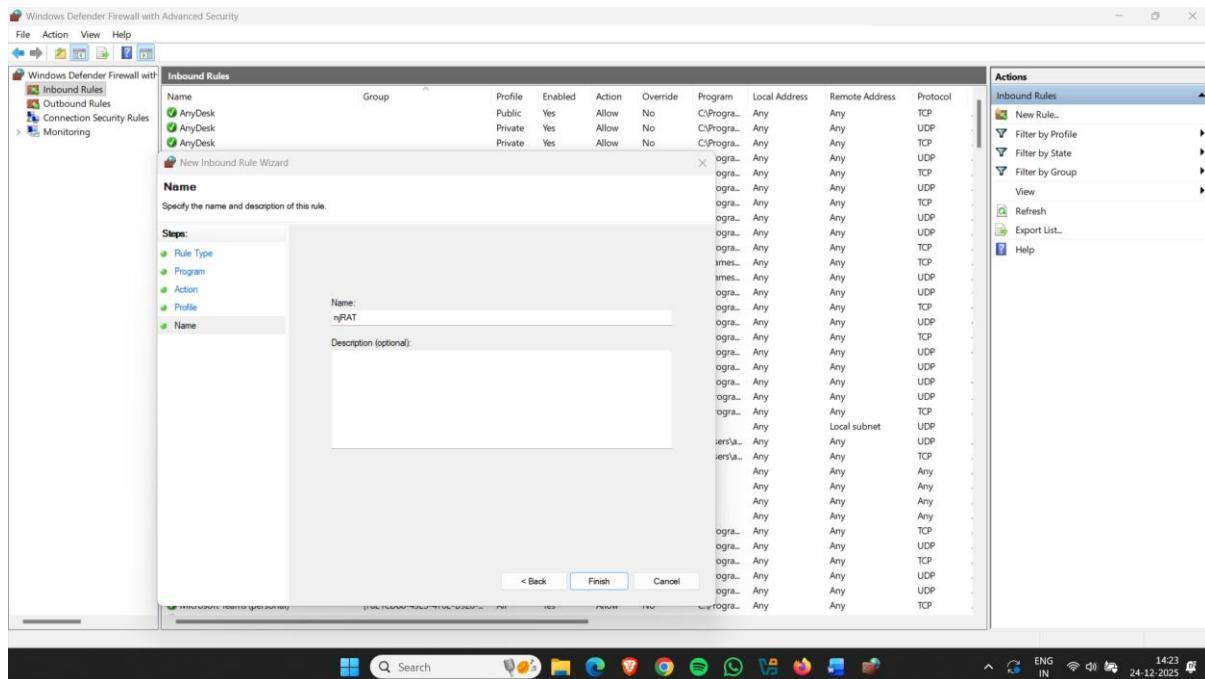


Figure 35

- Configuration has been set for inbound rules

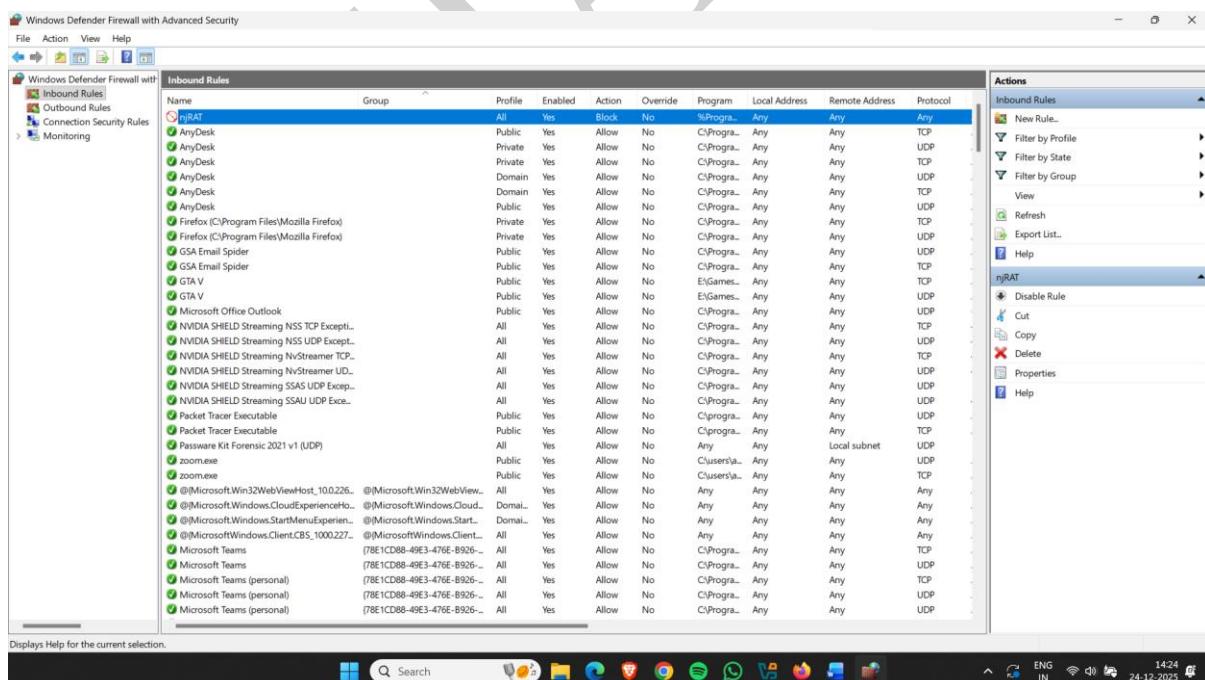


Figure 36

(OUTBOUND RULES)

- Click on Windows Button and search Firewall and open it

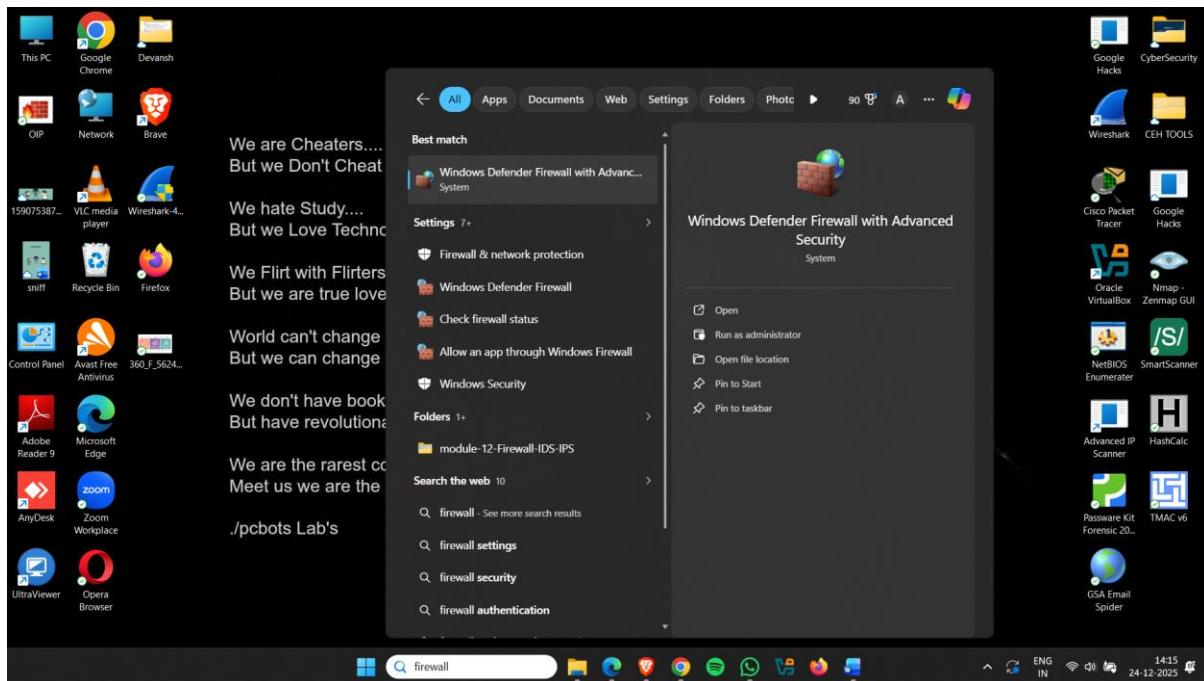


Figure 37

- There is option outbound rules – it means set rules for outgoing network

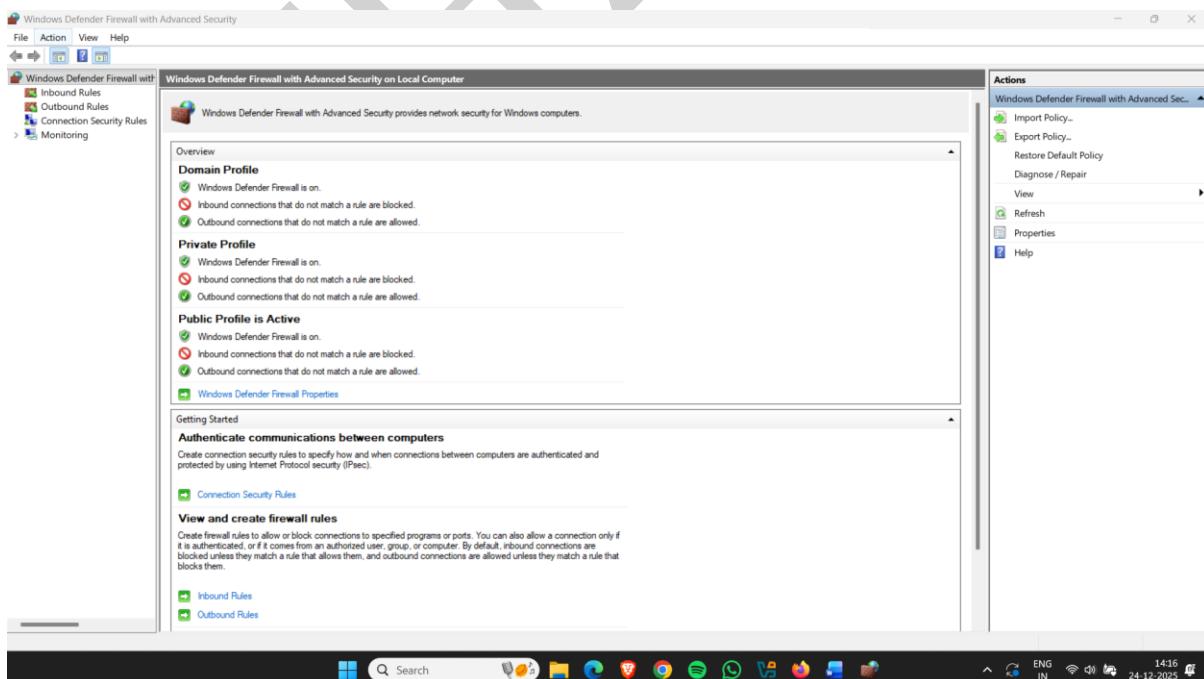


Figure 38

- Click on Outbound rules

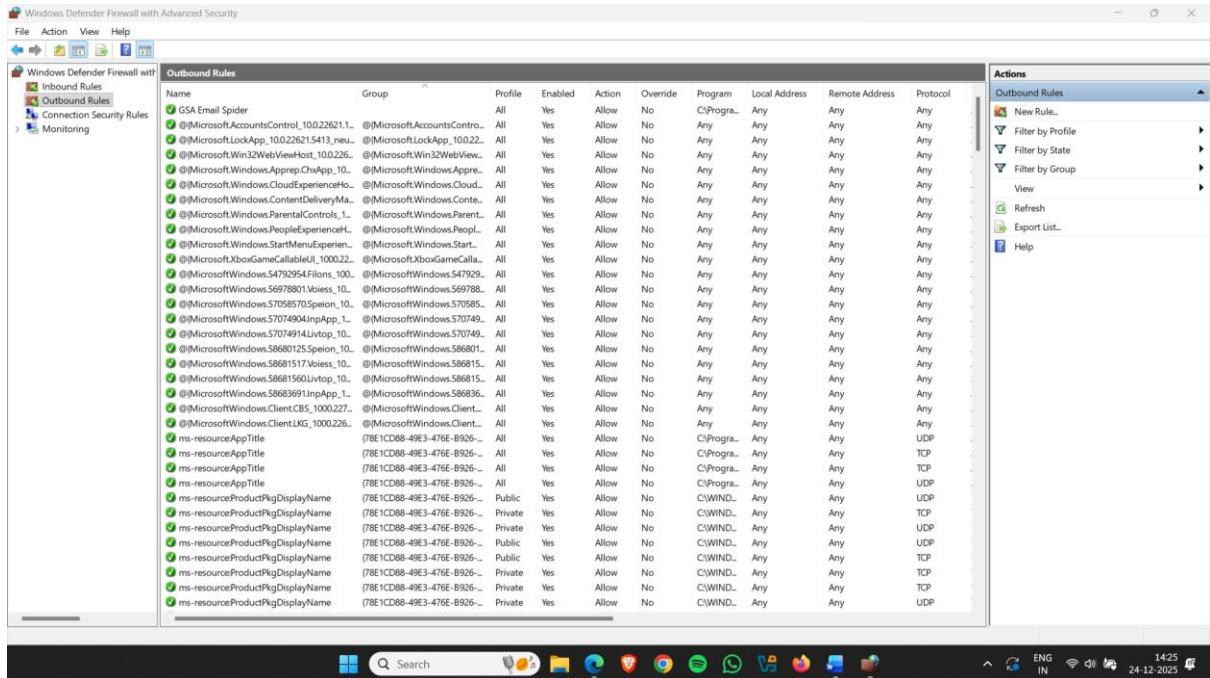


Figure 39

- Then click on New Rule
- New Pop Up appear with Three Options ,click
- on that you want to set a rule, and click on next.

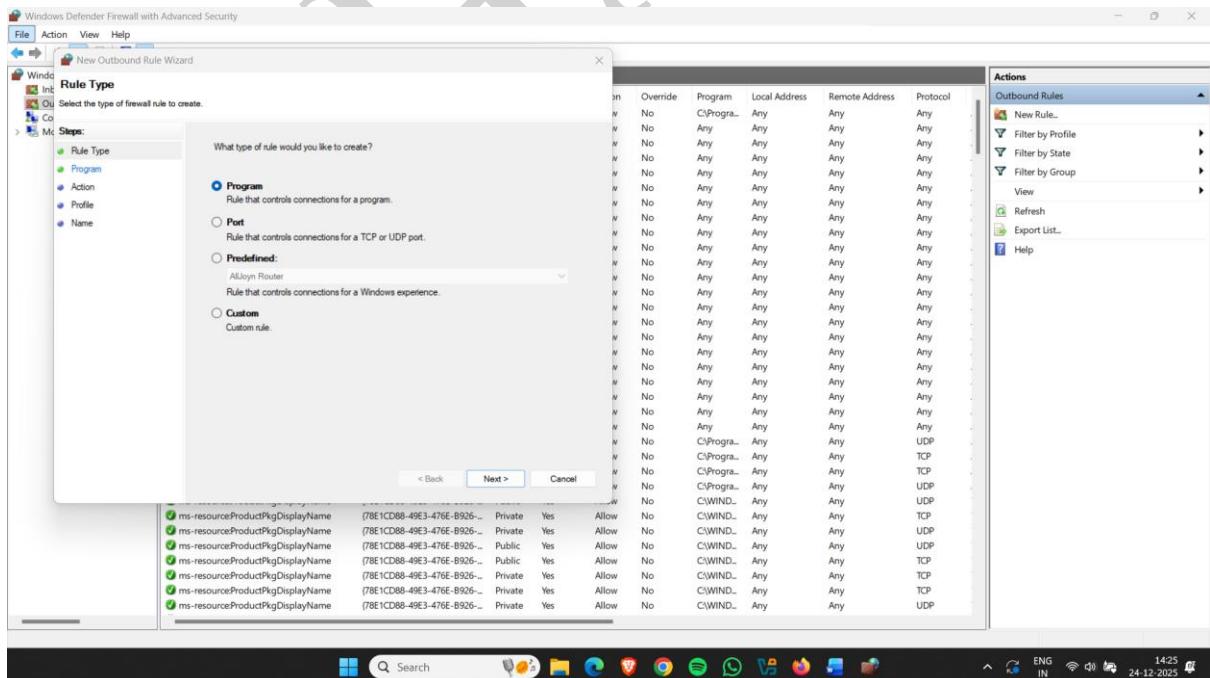


Figure 40

- Then set a path of program
Select a file and click on open

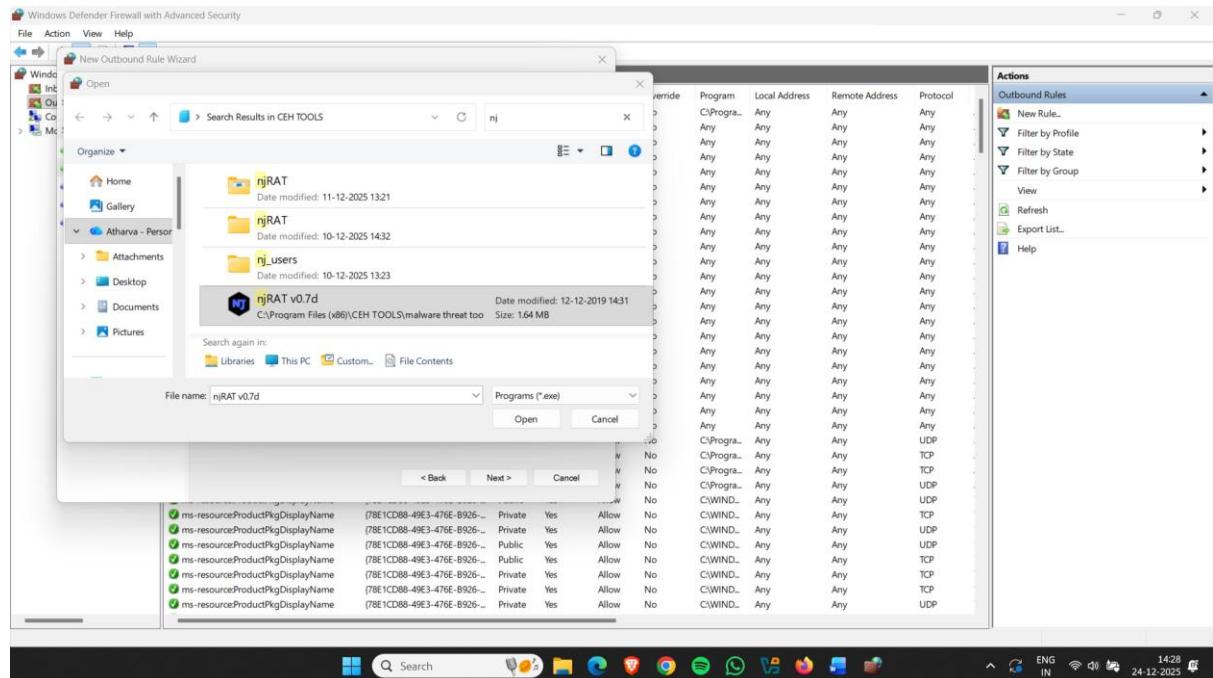


Figure 41

- Now ,set the path by clicking on Browse

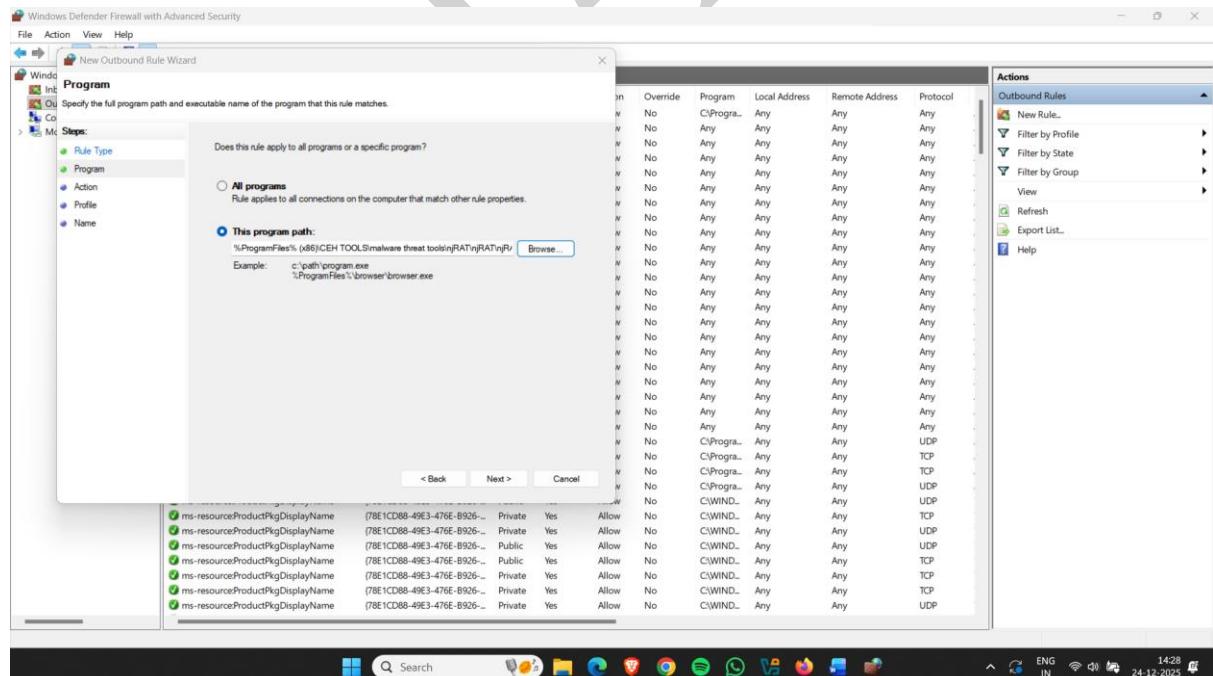


Figure 42

When configuring a firewall rule, three connection action options are available:

- **Allow the Connection**

Description:

This option allows all matching incoming traffic, regardless of whether the connection is secured or not.

Security Level:

Not restricted to authenticated or encrypted connections.

Allows both secure and non-secure traffic.

Use When:

You trust the application, service, or traffic source.

You want to permit communication without enforcing additional security measures.

- **Allow the Connection if it is Secure**

Description:

This option allows only those connections that are authenticated and secured using **IPsec (Internet Protocol Security)**.

Requirements:

Proper configuration of IPsec policies is required.

Security Level:

Provides encrypted and authenticated communication.

More secure than allowing all connections.

Use When:

You want to protect sensitive systems or data transfers.

Only trusted and verified devices should be allowed to connect.

- **Block the Connection**

Description:

This option completely denies the connection, whether it is secure or not.

Security Level:

Strictest security setting.

Prevents all matched traffic.

Use When:

You want to stop all communication on specific ports, programs, or IP addresses.

The traffic is untrusted or potentially harmful.

- Select an option & then click on next

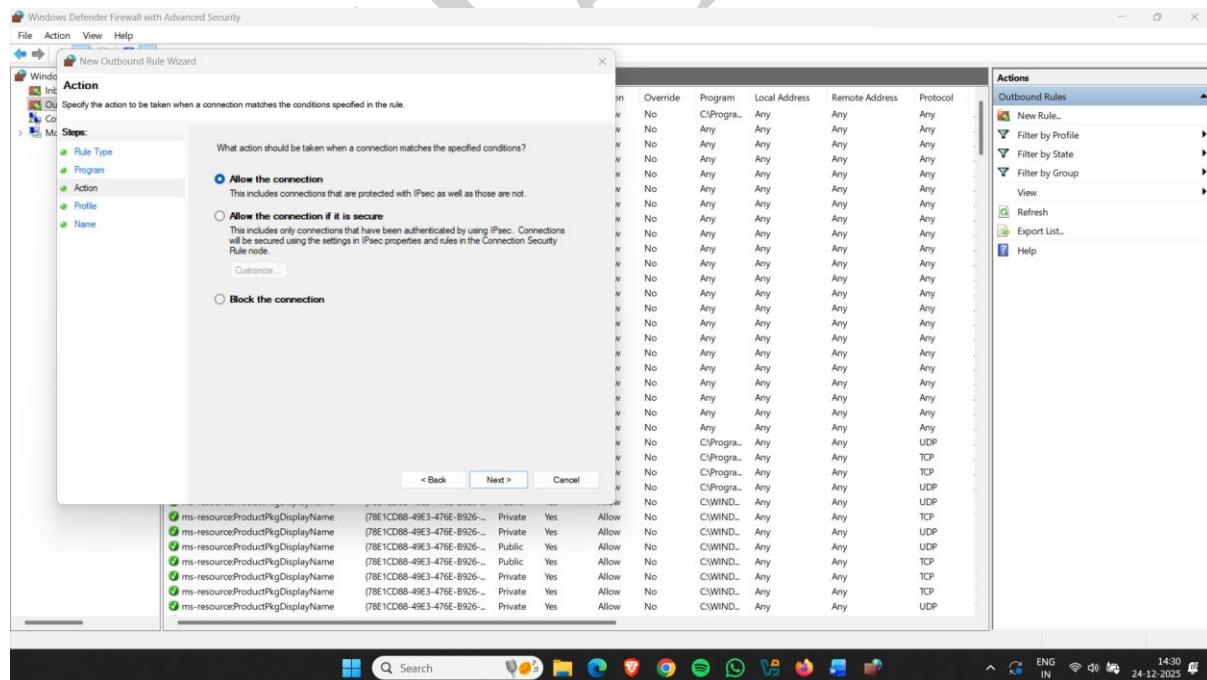


Figure 43

- Click on NEXT

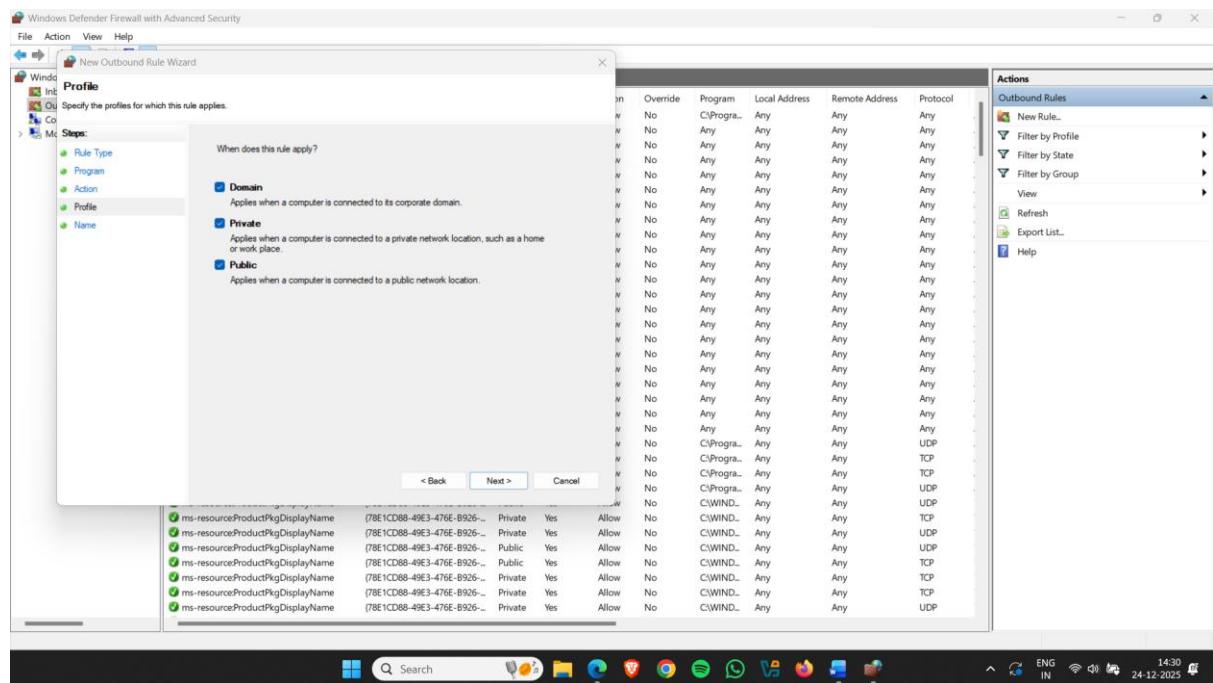


Figure 44

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

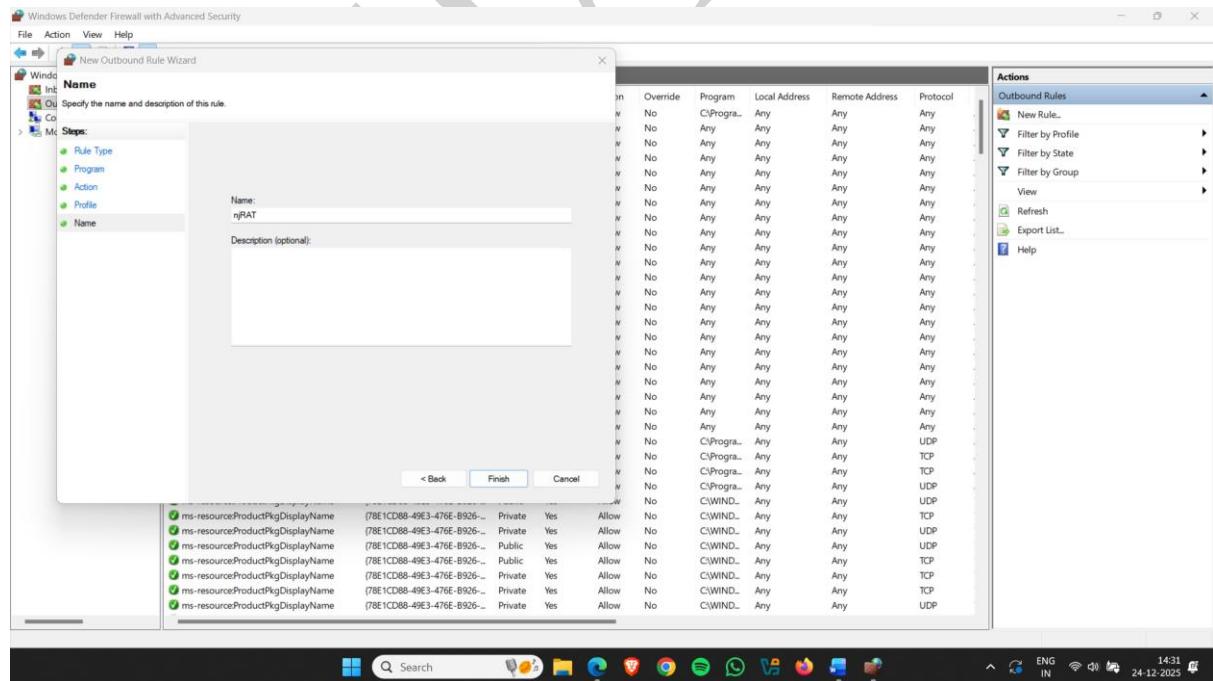


Figure 45

- Rule set successfully

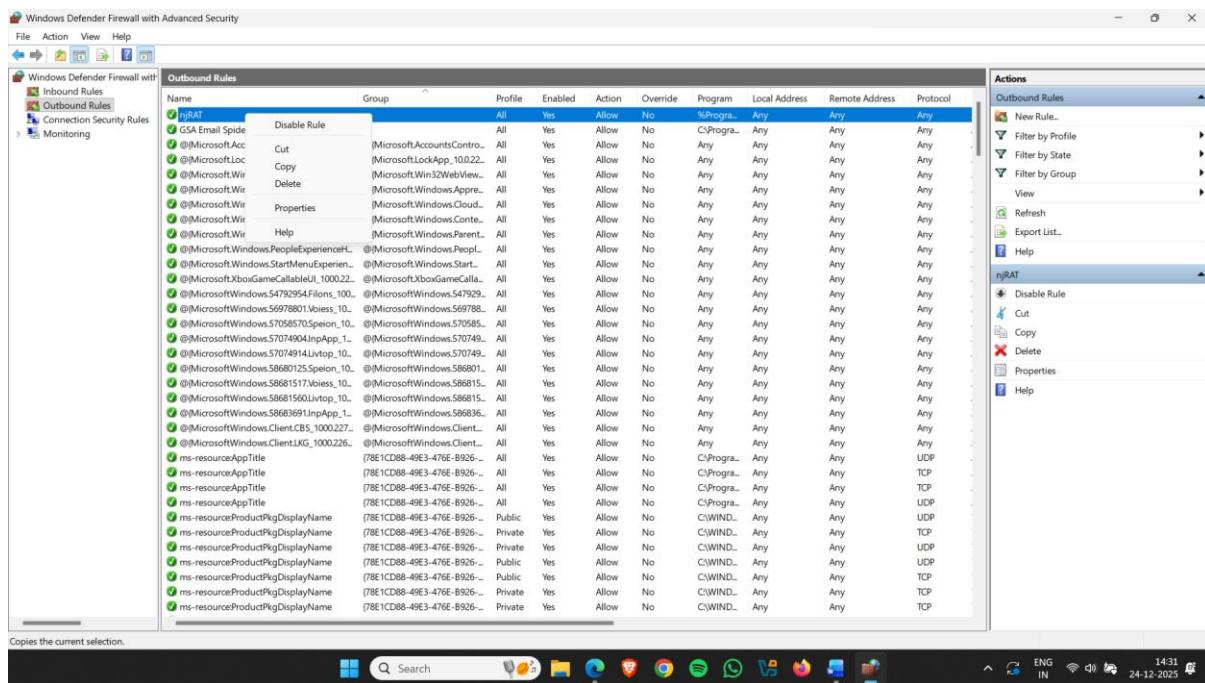


Figure 46

EXTRA ACTIVITY

Zone Alarm Firewall

ZoneAlarm Firewall is a third-party software firewall for Windows developed by Check Point Software Technologies. It provides an additional layer of security beyond the default Windows Defender Firewall, making it especially useful for users who want greater control over network activities.

Key Features of ZoneAlarm Firewall

1. Two-Way Firewall

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

2. Stealth Mode

- Makes the PC invisible to hackers on the internet.
- Hides open ports from external scans and attacks.

3. Application Control

- Alerts users when applications attempt to access the internet.
- Allows or blocks internet access on a per-application basis.

4. Real-Time Monitoring

- Continuously tracks all network activity.
- Displays which programs are currently using the internet.

5. DefenseNet™

- A cloud-based security feature.
- Uses community feedback and intelligence to automatically configure trusted applications.

6. Identity Protection (*Optional – Pro Version*)

- Alerts users if personal or sensitive data is at risk.
- Includes credit monitoring features (available for U.S.-based users).

How to use it:

- After Installation , Open the application
- Now click on firewall

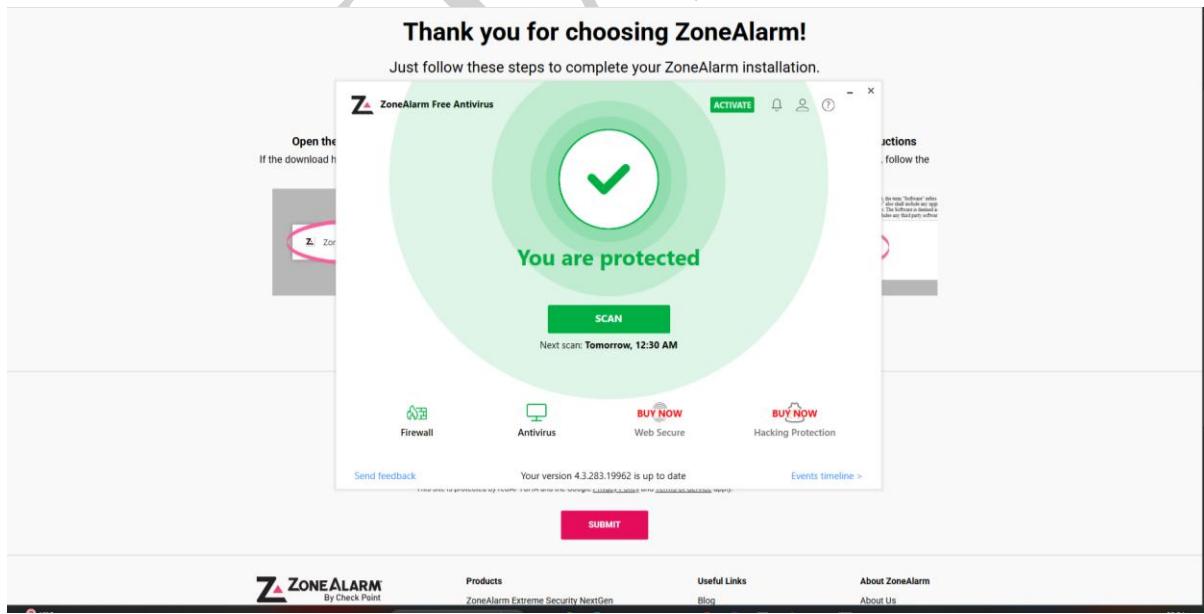


Figure 47

- Click on Application Permissions

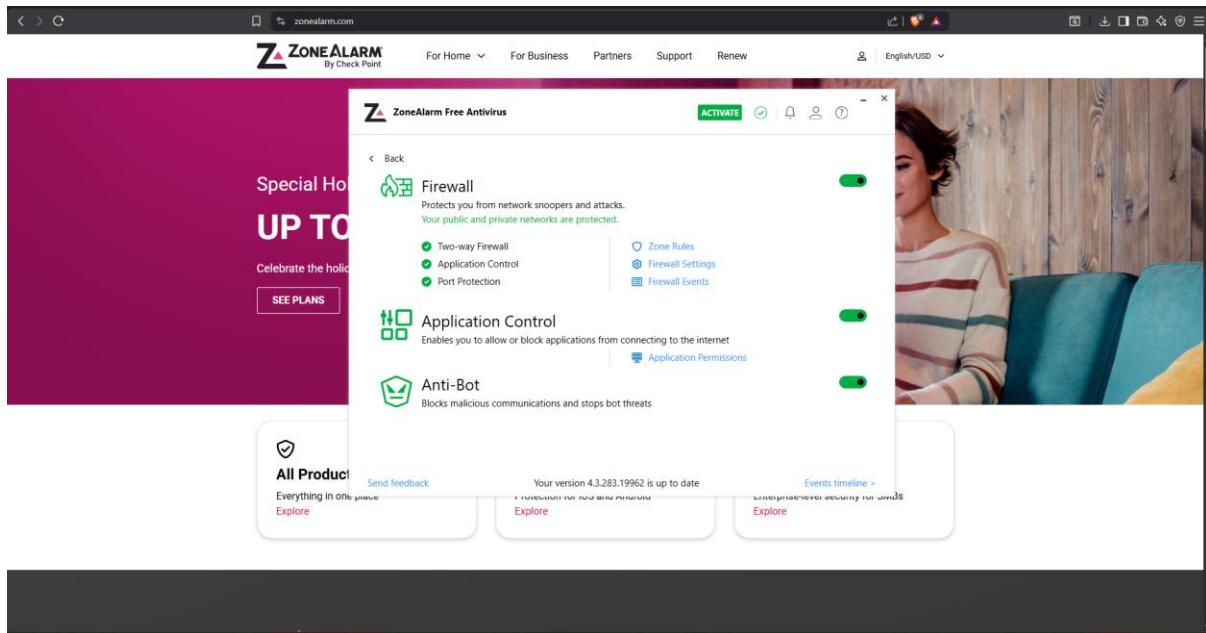


Figure 48

- Select the program that you want set a rule

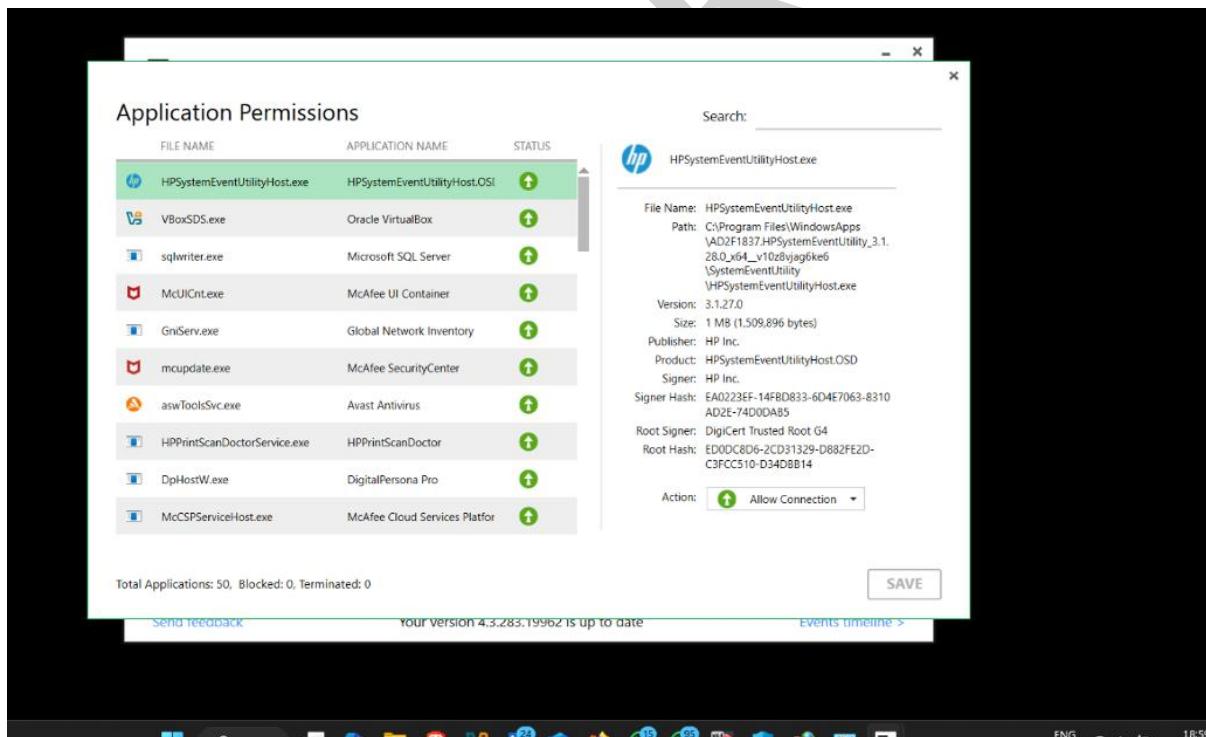


Figure 49

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

1. Allow Connection

Description:

- Permits the selected application or service to access the internet.
- The firewall allows all related network communication.

Use When:

- The application is trusted and requires internet access.
-

2. Block Connection

Description:

- Prevents the selected application or service from accessing the internet.
- All network communication for the application is denied.

Use When:

- The application is untrusted or should not communicate over the network.
-

3. Terminate Process

Description:

- Immediately stops the selected process if it is currently running.
- Prevents further network or system activity by the process.

Use When:

- The process is suspicious, unresponsive, or consuming excessive resources.

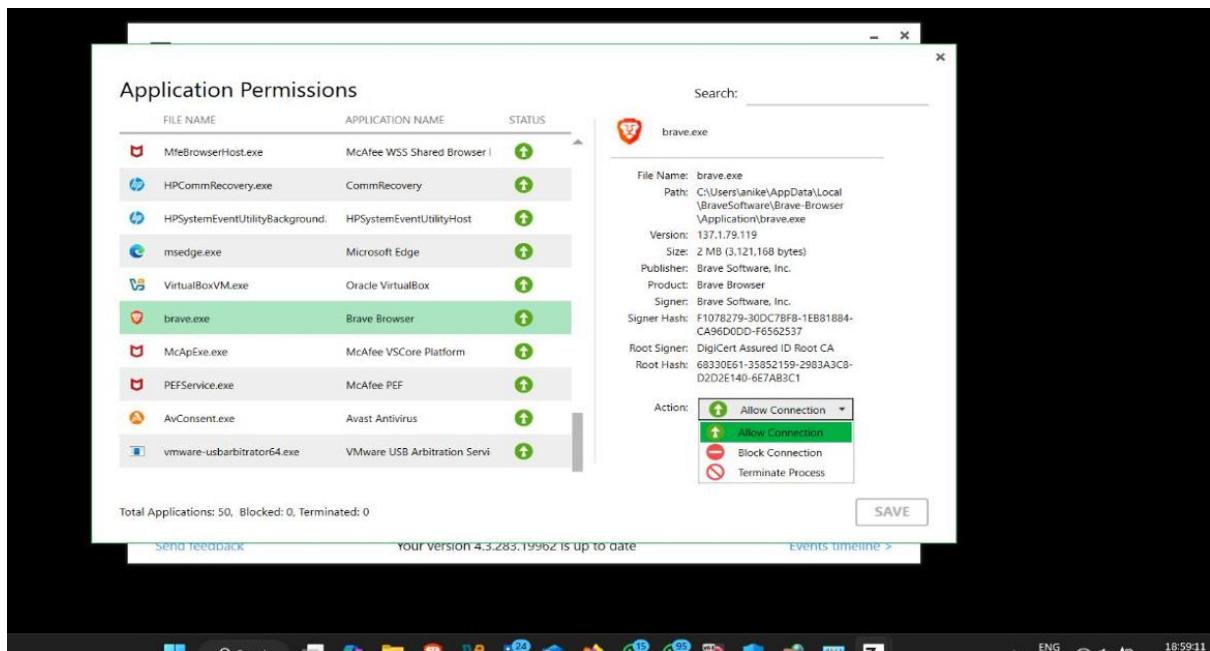


Figure 50

- Select an option & then click on save

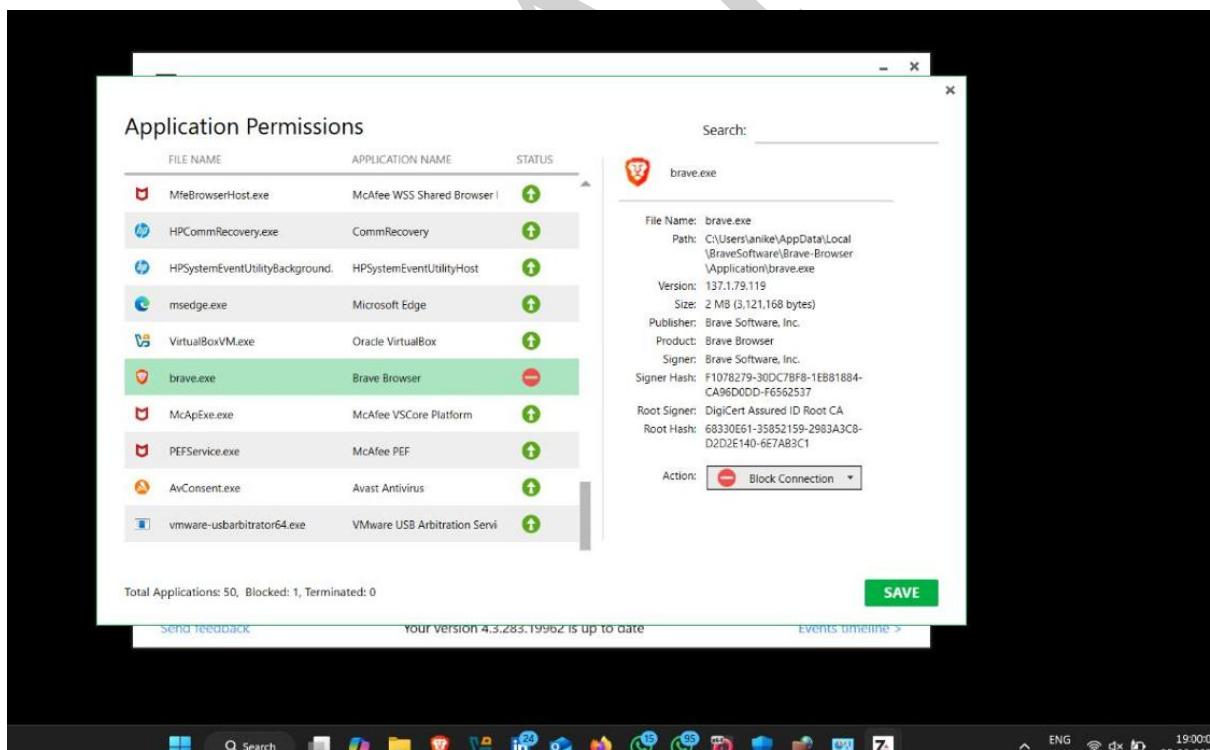


Figure 51

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

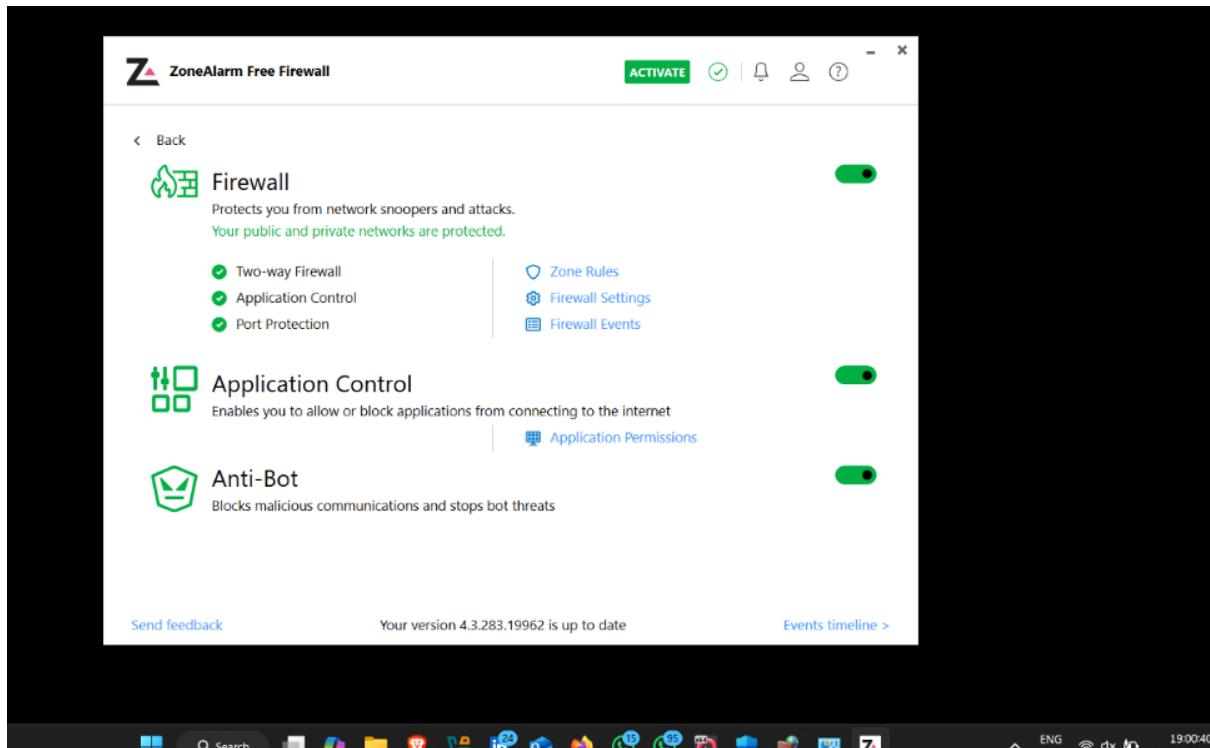


Figure 52

- It show all the event

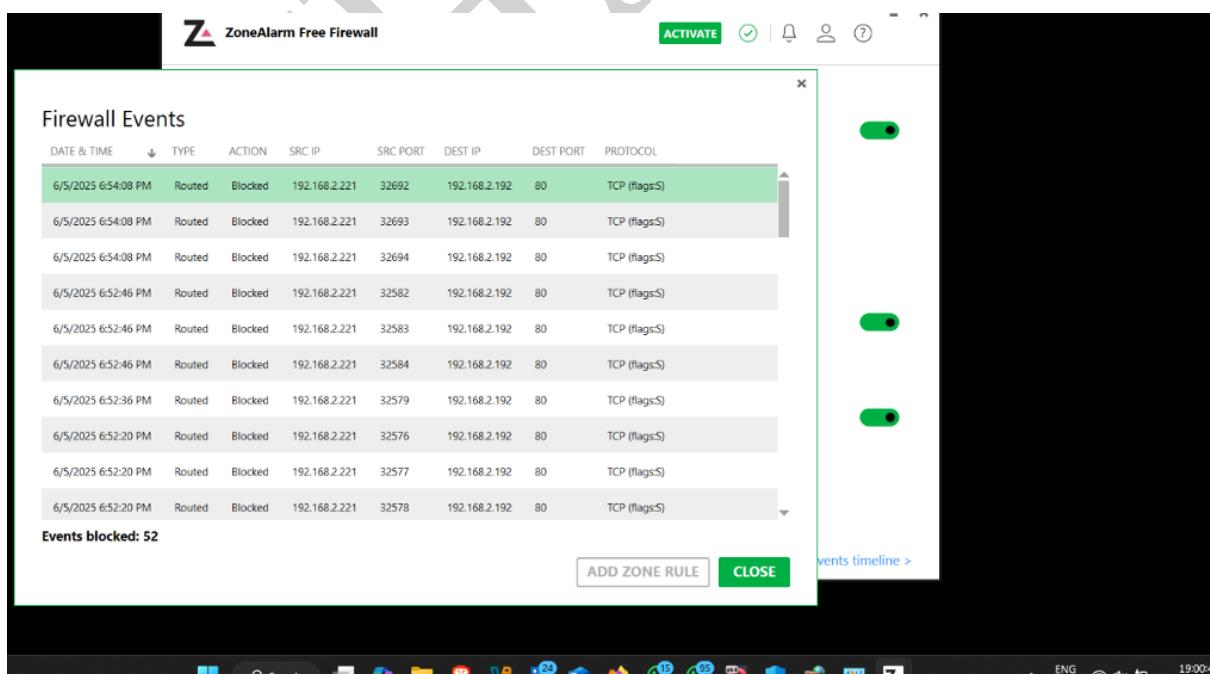


Figure 53

How to Defend Against Firewall Evasion

Defending against firewall evasion requires a combination of advanced firewall capabilities, strict traffic validation, continuous monitoring, and layered security controls. The following best practices help prevent attackers from bypassing firewall protections.

1. Implement Stateful Inspection Firewalls

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

2. Enable Strict Packet Filtering and Validation

- Validate protocol compliance for all incoming and outgoing traffic.
- Drop malformed, unexpected, or suspicious packets immediately.

3. Use Deep Packet Inspection (DPI)

- Inspect packet payloads in addition to packet headers.
- Detect tunneling techniques and malicious payloads hidden inside allowed protocols.

4. Block IP Spoofing

- Enable **Reverse Path Forwarding (RPF)** or **Unicast RPF** on routers and firewalls.
- Reject packets with source IP addresses that do not match legitimate routing paths.

5. Disallow Source Routing

- Block IP packets that have source routing options enabled.
- Most modern firewalls provide configuration options to reject such packets.

6. Reassemble Fragments for Inspection

- Ensure the firewall can fully reassemble fragmented IP packets before inspection.
- Prevent attackers from hiding malicious data within packet fragments.

7. Inspect and Control Encrypted Traffic

- Use SSL/TLS inspection or secure proxy servers to decrypt and analyze HTTPS traffic.
- Detect malware or command-and-control traffic concealed inside encrypted channels.

8. Implement Strict Port and Protocol Controls

- Explicitly allow only required ports and protocols.
- Block unused, unnecessary, or risky services.

9. Regularly Update Firewall Firmware and Rules

- Keep firewall software and firmware updated to fix known evasion vulnerabilities.
- Regularly review, audit, and fine-tune firewall rules.

10. Use an Intrusion Prevention System (IPS) Alongside the Firewall

- Deploy an IPS to detect and block sophisticated or evasive attack signatures.
- Adds an additional layer of real-time traffic inspection.

11. Monitor Logs and Traffic Patterns

- Continuously monitor firewall logs for abnormal or suspicious activity.
- Identify patterns that may indicate firewall evasion attempts.

12. Employ Network Segmentation and Zero Trust

- Divide the network into segments to limit lateral movement.
- Apply **Zero Trust principles**, where no traffic is trusted by default.

13. Implement Rate Limiting and Traffic Shaping

- Restrict excessive traffic bursts and scanning behavior.
- Helps block slow, stealthy, or low-rate evasion techniques.

14. Use Anti-Spoofing Filters at the Network Edge

- Configure edge routers and switches to drop spoofed packets.
- Reduces malicious traffic before it reaches the firewall.

Common Firewall Evasion Techniques Used by Attackers

1. IP Spoofing

- Attackers fake the source IP address of packets to make them appear as if they are coming from a trusted host.

2. Port Spoofing and Port Changing

- Malicious traffic is sent through non-standard or commonly allowed ports to bypass blocked or monitored ports.

3. Fragmentation Attacks

- Malicious packets are split into smaller fragments to evade firewall inspection and reassembly mechanisms.

4. Protocol Anomalies and Violations

- Attackers send malformed or unexpected packets to confuse firewall protocol analysis and inspection engines.

5. Tunneling and Encapsulation

- Malicious traffic is hidden inside permitted protocols such as:
 - HTTP tunneling
 - DNS tunneling
 - VPN encapsulation

6. Use of Encrypted Traffic

- Attackers use SSL/TLS encryption so the firewall cannot inspect packet contents effectively.

7. Source Routing

- Manipulation of IP header fields to control packet routing and potentially bypass firewall filtering rules.

8. Session Hijacking and TCP Sequence Prediction

- Attackers manipulate TCP session parameters to bypass stateful firewall inspection mechanisms.

9. Fragment Overlaps and Reassembly Confusion

- Exploiting weaknesses in how firewalls reassemble fragmented packets to avoid detection.

10. Malicious Payload in Allowed Traffic

- Malware is concealed within legitimate-looking traffic such as HTTPS or DNS communications.

How to Defend Against IDS Evasion

Intrusion Detection System (IDS) evasion refers to techniques used by attackers to bypass detection mechanisms by manipulating network traffic patterns, packet structure, or payload content. Understanding these evasion techniques helps security professionals strengthen IDS effectiveness.

1. Use a Combination of Detection Methods

- Combine **signature-based IDS** with **anomaly-based IDS**.
- Anomaly-based IDS can detect unusual traffic behavior even when attackers attempt signature evasion.

2. Enable Full Packet Reassembly

- Configure the IDS to fully reassemble fragmented IP packets and TCP streams before analysis.
- Prevent evasion techniques such as fragmentation and overlapping packets.

3. Regularly Update IDS Signatures

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

4. Use Deep Packet Inspection (DPI)

- Inspect packet payloads in detail rather than only headers.
- Decode encoded or obfuscated payloads before analysis to reveal hidden attacks.

5. Implement SSL/TLS Inspection

- Use SSL/TLS decryption capabilities or proxy solutions to inspect encrypted traffic.

- Helps identify malicious content hidden inside encrypted sessions.

6. Behavioral and Contextual Analysis

- Correlate IDS alerts with logs from firewalls, endpoints, and other security systems.
- Use contextual information such as user behavior, time of day, and historical traffic patterns to detect evasion.

7. Deploy Multi-Layered Security

- Integrate IDS with:
 - Intrusion Prevention Systems (IPS)
 - Firewalls
 - Endpoint protection solutions
 - Network segmentation
- A **defense-in-depth** strategy significantly reduces the success of evasion attempts.

8. Tune IDS Rules to Reduce False Negatives

- Customize IDS rules based on the specific network environment.
- Modify or remove ineffective rules while balancing false positives.

9. Monitor Traffic Patterns

- Watch for unusual traffic timing, burst patterns, or inconsistent data flows.
- Use **Network Behavior Anomaly Detection (NBAD)** tools.

10. Log and Audit Everything

- Maintain detailed logs of IDS alerts, network flows, and system events.
- Enables detection and analysis of stealthy or long-term attacks.

11. Use Honeypots and Deception Techniques

- Deploy honeypots to attract attackers and observe evasive behavior.
- Helps in identifying new or unknown evasion techniques.

12. Train Security Personnel

- Keep SOC analysts and network administrators updated on IDS evasion tactics.
- Train teams to recognize suspicious activity that may bypass IDS controls.

Common IDS Evasion Techniques Used by Attackers

1. Fragmentation

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

2. Packet Overlapping

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

3. Protocol Anomalies

- Using unusual, malformed, or non-standard packets that the IDS may fail to parse correctly.

4. Polymorphic Shellcode

- Continuously changing malware code structure or encryption methods to evade signature-based detection.

5. Payload Encoding

- Encoding attack payloads using formats such as:
 - Base64 encoding
 - URL encoding
 - Unicode encoding
- This helps hide known attack patterns from IDS signatures.

6. Traffic Timing and Rate Manipulation

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

7. Use of Encrypted Traffic

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

8. Evasion of Signature Matching

- Exploiting zero-day vulnerabilities or using modified attack variants not covered by existing IDS signatures.

9. Avoiding Known Ports

- Transmitting malicious traffic through uncommon or commonly allowed ports to bypass IDS monitoring rules.

10. Session Splicing

- Splitting an attack across multiple sessions or connections to prevent the IDS from detecting a complete attack sequence.

How to Defend Against IPS Evasion

Intrusion Prevention System (IPS) evasion refers to techniques used by attackers to bypass inline security controls by manipulating packet structure, traffic behavior, or payload content. Understanding these techniques helps in strengthening IPS detection and prevention mechanisms.

1. Implement Stateful Inspection Firewalls

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

2. Enable Strict Packet Filtering and Validation

- Validate protocol compliance for all incoming and outgoing traffic.
- Drop malformed, unexpected, or suspicious packets immediately.

3. Use Deep Packet Inspection (DPI)

- Inspect packet payloads in addition to packet headers.
- Detect tunneling techniques and malicious payloads hidden inside allowed protocols.

4. Block IP Spoofing

- Enable **Reverse Path Forwarding (RPF)** or **Unicast RPF** on routers and firewalls.
- Reject packets with source IP addresses that do not match legitimate routing paths.

5. Disallow Source Routing

- Block IP packets that have source routing options enabled.
- Most modern firewalls provide configuration options to reject such packets.

6. Reassemble Fragments for Inspection

- Ensure the firewall can fully reassemble fragmented IP packets before inspection.
- Prevent attackers from hiding malicious data within packet fragments.

7. Inspect and Control Encrypted Traffic

- Use SSL/TLS inspection or secure proxy servers to decrypt and analyze HTTPS traffic.
- Detect malware or command-and-control traffic concealed inside encrypted channels.

8. Implement Strict Port and Protocol Controls

- Explicitly allow only required ports and protocols.
- Block unused, unnecessary, or risky services.

9. Regularly Update Firewall Firmware and Rules

- Keep firewall software and firmware updated to fix known evasion vulnerabilities.
- Regularly review, audit, and fine-tune firewall rules.

10. Use an Intrusion Prevention System (IPS) Alongside the Firewall

- Deploy an IPS to detect and block sophisticated or evasive attack signatures.
- Adds an additional layer of real-time traffic inspection.

11. Monitor Logs and Traffic Patterns

- Continuously monitor firewall logs for abnormal or suspicious activity.
- Identify patterns that may indicate firewall evasion attempts.

12. Employ Network Segmentation and Zero Trust

- Divide the network into segments to limit lateral movement.
- Apply **Zero Trust principles**, where no traffic is trusted by default.

13. Implement Rate Limiting and Traffic Shaping

- Restrict excessive traffic bursts and scanning behavior.
- Helps block slow, stealthy, or low-rate evasion techniques.

14. Use Anti-Spoofing Filters at the Network Edge

- Configure edge routers and switches to drop spoofed packets.
- Reduces malicious traffic before it reaches the firewall

Common IPS Evasion Techniques Used by Attackers

1. Packet Fragmentation

- Splitting the attack payload into very small fragments to avoid detection during packet inspection and reassembly.

2. TCP Stream Manipulation

- Manipulating TCP packets by altering:
 - Sequence numbers
 - Overlapping segments
 - Out-of-order packets
- This confuses the IPS during stream reassembly.

3. Protocol Violations and Anomalies

- Sending malformed packets or using unusual protocol behaviour's to evade signature-based or protocol-aware inspection.

4. Encoding and Obfuscation

- Encoding payloads using techniques such as Base64 or Unicode.
- Using polymorphic malware to bypass signature-based detection.

5. Traffic Timing Evasion

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

6. Use of Encrypted Traffic

- Encrypting attack payloads inside SSL/TLS sessions so the IPS cannot inspect contents without decryption capabilities.

7. Tunneling and Encapsulation

- Wrapping malicious traffic inside allowed protocols such as:
 - HTTP/DNS/SSH tunnels

8. Payload Padding and NOP Sleds

- Adding no-operation (NOP) instructions or junk bytes to disrupt signature matching.

9. Attack Variants and Zero-Day Exploits

- Using unknown, new, or slightly modified exploits that are not yet included in IPS signature databases.

10. Session Splicing

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