REPORT OF EVADING IDS, FIREWALLS, AND HONEYPOTS

BY SACHCHITANAND YADAV

EVADING IDS, FIREWALLS, AND HONEYPOTS

MODULE - 12

Learning Objectives -

- Explain Evading IDS, Firewalls, and Honeypots Concepts
- Perform Intrusion Detection using Various Tools
- Explain Evading IDS, Firewalls, and Honeypots Countermeasures

Table of Contents

2.6 Snort Installation and Basic Setup
2.7 Advantages of Snort
2.8 Limitations of Snort

---

3. Snort Configuration and Implementation

3.1 Download and Installation
3.2 Configuration File Setup
3.3 Rule Configuration
3.4 Command-Line Execution
3.5 Validation and Testing of Configuration
3.6 Starting Snort in IDS Mode

---

4. Evading IDS, Firewalls, and Honeypots – Countermeasures

4.1 Countermeasures Against IDS Evasion
4.2 Countermeasures Against Firewall Evasion
4.3 Countermeasures Against Honeypot Evasion
4.4 Conclusion

---

5. Module Summary

5.1 Evading IDS
5.2 Evading Firewalls
5.3 Evading Honeypots

Evading IDS, Firewalls, and Honeypots Concepts: -

IDS, IPS, and Firewall Concepts

Ethical hackers should understand the function, role, placement, and design of firewalls, IDS, and IPS, as well as how attackers evade these security measures. This section provides an overview of these concepts.

Firewall: The Network Control Barrier

A firewall is a security mechanism—hardware or software—that regulates network traffic using predefined security policies. Its primary role is to create a controlled boundary between trusted internal networks and untrusted external networks, such as the internet.

Rather than trusting everything, a firewall assumes traffic must prove itself first.

Classification of Firewalls

1. Packet Filtering Firewall
Operates at the network layer by examining packet headers such as source/destination IP, port numbers, and protocols.
It is fast and resource-efficient but lacks deep inspection capabilities.

2. Stateful Inspection Firewall
Tracks the state of active connections and evaluates traffic based on session context.
This method offers better protection by understanding whether packets belong to legitimate sessions.

3. Application-Level Gateway (Proxy Firewall)
Functions at the application layer and acts as an intermediary between users and services.
It inspects application data directly, offering enhanced security against application-layer attacks.

4. Next-Generation Firewall (NGFW)
An advanced firewall that integrates traditional filtering with modern security features such as:

- Deep packet inspection
- Intrusion prevention
- Application identification
- Malware and threat intelligence integration

5. Software Firewall
Installed on individual endpoints to protect a single system from unauthorized access.

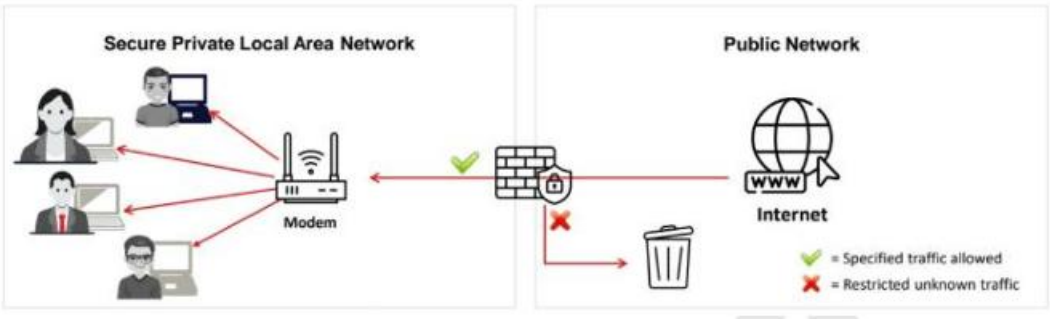
6. Hardware Firewall
A dedicated physical device that secures an entire network, commonly used in enterprise environments.

Importance of Firewalls

Firewalls play a crucial role in network security by:

- Preventing unauthorized access
- Reducing exposure to cyber threats
- Enforcing organizational security rules
- Monitoring and logging network activity

Despite newer technologies, firewalls remain a foundational security control.



Firewall Operational Overview

When traffic attempts to enter or leave a network, the firewall evaluates it against predefined rules. If the traffic complies, it is allowed; otherwise, it is blocked and logged for analysis.

Simple logic. Zero tolerance.

Intrusion Detection System (IDS)

An Intrusion Detection System (IDS) is a monitoring solution that analyzes network or system activity to identify suspicious behavior, security violations, or attempted attacks. Unlike firewalls, IDS does not block traffic—it detects and alerts.

Think of it as the network's surveillance system.

Objectives of IDS

IDS is designed to:

- Monitor traffic and system behavior
- Detect intrusion attempts
- Generate alerts for security teams
- Identify malware and exploit patterns
- Enable early incident detection
- Support forensic investigations
- Detect insider threats
- Maintain logs for compliance and audits

IDS enhances visibility where firewalls stop short.

Types of Intrusion Detection Systems

Network-Based IDS (NIDS)

Monitors traffic across network segments to detect malicious patterns.
It is deployed at strategic locations such as gateways or DMZs.

Advantages:

- Broad network visibility
- Effective against external attacks

Examples: Snort, Suricata

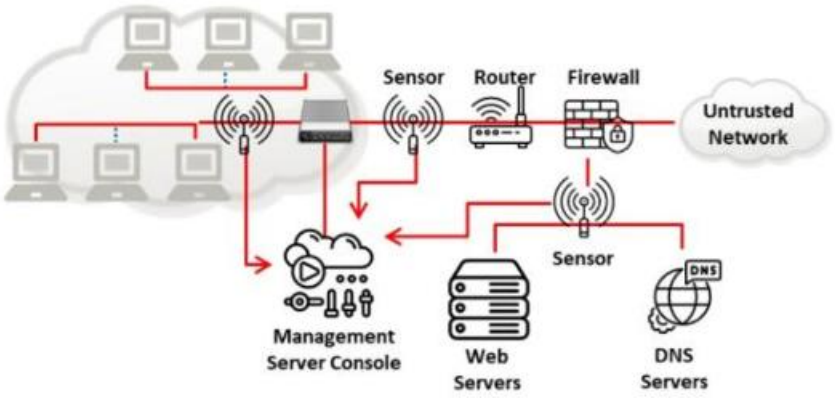


Fig-:Network-Based IDS

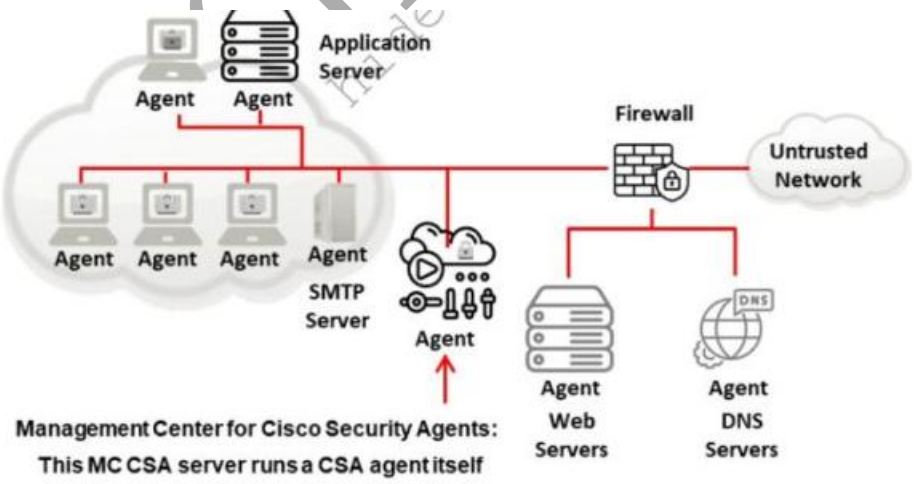
Host-Based IDS (HIDS)

Installed on individual systems to monitor internal activity such as file changes, logs, and user actions.

Advantages:

- Detects insider threats
- Protects critical servers

Examples: OSSEC, Tripwire

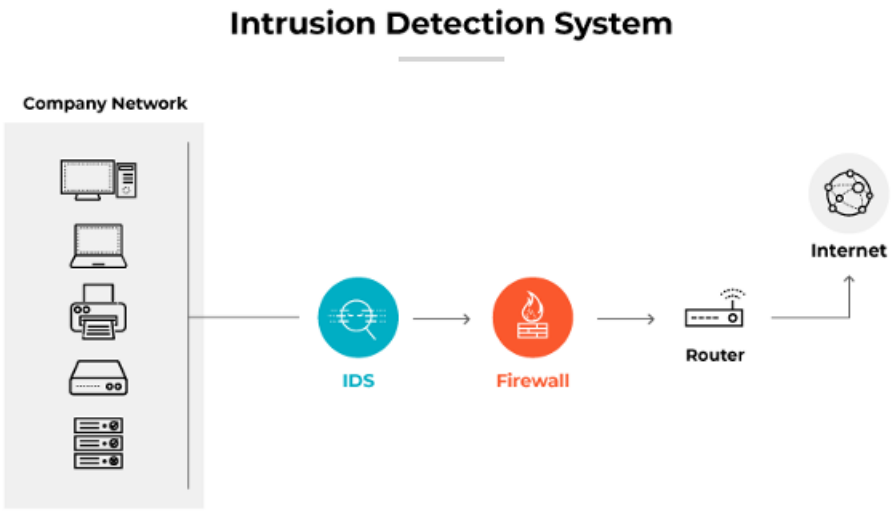


Intrusion Detection Techniques

Signature-Based Detection
Matches activity against known attack signatures.
Highly accurate for known threats but ineffective against new attacks.

Anomaly-Based Detection
Establishes a baseline of normal behavior and flags deviations.
Capable of detecting zero-day attacks but may generate false alerts.

Protocol Anomaly Detection
Identifies violations of protocol standards, indicating possible exploitation attempts.



IDS Alert Categories

- **True Positive:** Correct detection of an attack
- **True Negative:** Correct identification of normal behavior
- **False Positive:** Benign activity flagged as malicious
- **False Negative:** Attack not detected (most dangerous scenario)

Security success depends on minimizing false negatives without drowning in false positives.

Intrusion Prevention System (IPS)

An Intrusion Prevention System (IPS) is an inline security device that actively monitors and blocks malicious traffic in real time. Unlike IDS, IPS can take immediate corrective action.

It doesn't warn—it intervenes.

IPS Response Actions

- Dropping malicious packets
- Blocking IP addresses
- Resetting connections
- Generating alerts

IPS Functional Workflow

1. Continuous traffic inspection
2. Signature and behavior analysis
3. Threat identification
4. Automatic prevention before damage occurs

Speed is everything here.

Types of IPS

Network-Based IPS (NIPS):
Protects entire networks and monitors traffic between hosts.

Host-Based IPS (HIPS):
Secures individual systems by monitoring system calls and application behavior.

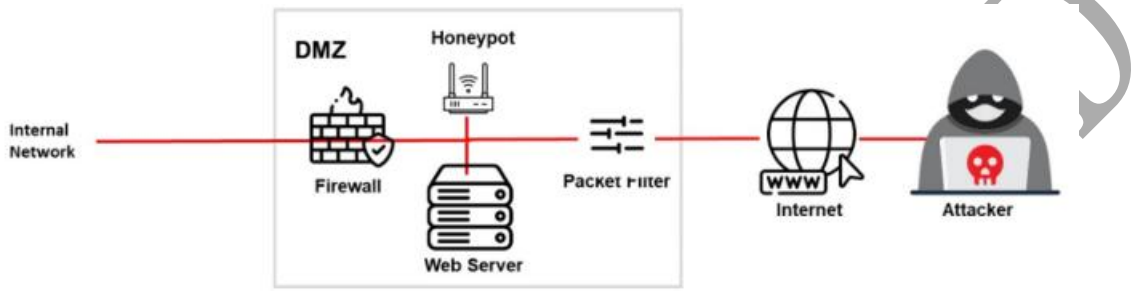
Wireless IPS (WIPS):
Protects wireless networks from rogue devices and wireless attacks.

Network Behavior Analysis IPS (NBA):
Uses traffic behavior modeling to detect anomalies, insider threats, and large-scale attacks.

Honeypot: Deception-Based Security

A honeypot is a deliberately vulnerable system designed to attract attackers. It serves no legitimate purpose other than to observe, record, and analyze malicious behavior.

The attacker thinks they're winning. They're actually being studied.



Objectives of Honeypots

- Detect unauthorized activity
- Divert attackers from real assets
- Collect threat intelligence
- Improve defensive strategies

Honeypot Operation

Honeypots simulate real systems in isolated environments. Any interaction with them is treated as suspicious and logged for analysis.

Zero noise. Pure signal.

Types of Honeypots

- Low, medium, and high interaction honeypots
- Research and production honeypots
- Internal and external honeypots
- Specialized honeypots (malware, web, database, ICS)

Honeynet:
A group of interconnected honeypots that simulate an entire network for large-scale attack analysis.

Conclusion

Firewalls establish control.
IDS provides visibility.
IPS delivers immediate defense.
Honeypots offer intelligence through deception.

Perform Intrusion Detection using Various Tools

Snort

Snort is an open-source **Network Intrusion Detection System (NIDS)** and **Intrusion Prevention System (IPS)** originally developed by Martin Roesch and currently maintained by **Cisco Systems**. It is widely used for real-time traffic analysis, packet inspection, and logging on IP-based networks.

Snort operates by analyzing network packets and comparing them against a set of predefined rules to detect malicious activity. When configured in IPS mode, it can actively block threats, making it both a detection and prevention tool.

Objectives and Uses of Snort

Snort is used in network security environments to:

- Detect network intrusions in real time
- Perform deep packet inspection (DPI)
- Monitor network traffic for suspicious patterns
- Capture and log packets for later analysis
- Prevent attacks when deployed in IPS mode
- Detect port scanning and reconnaissance activities
- Identify malware, exploits, and suspicious payloads
- Enforce organizational security policies
- Support custom rule creation for advanced threat detection
- Assist in forensic analysis and incident response

Key Features of Snort

- **Open-source and customizable**
- **Signature-based detection engine**
- **Flexible deployment (IDS or IPS)**
- **Real-time alerting and logging**
- **Cross-platform support (Linux, Windows)**
- **Large community and rule database**

Snort Architecture

Snort follows a modular architecture consisting of:

1. **Packet Decoder**
   Captures packets from the network interface.
2. **Preprocessors**
   Normalize traffic, reassemble fragments, and detect protocol anomalies.
3. **Detection Engine**
   Compares traffic against Snort rules to identify attacks.
4. **Logging and Alerting System**
   Generates alerts and logs suspicious activity.
5. **Output Modules**
   Stores logs in files, databases, or sends alerts to consoles or SIEM tools.

Modes of Operation of Snort

Snort can operate in three main modes:

1. Sniffer Mode

Displays packets on the console in real time.
Used mainly for learning and traffic observation.

2. Packet Logger Mode

Logs packets to disk for offline analysis.
Helpful during investigations.

3. Network Intrusion Detection / Prevention Mode

Analyzes traffic against rules and generates alerts or blocks traffic.
This is where Snort earns its reputation.

How to Use Snort (Basic Setup)

1. Download and install Snort from the official website.
2. Navigate to the Snort installation directory.
3. Open the **`etc`** folder, which contains:
   - `snort.conf` (main configuration file)
   - Rule files

    - Preprocessor settings
4. Configure the following in `snort.conf`:
    - Network variables (HOME_NET, EXTERNAL_NET)
    - Rule paths
    - Preprocessor settings
5. Update or add custom rules in the **rules** directory.
6. Run Snort using command-line options based on the desired mode (IDS or IPS).
7. Monitor alerts and logs generated by Snort for suspicious activity.

Advantages of Snort

- Free and open-source
- Highly customizable rule-based detection
- Strong community and documentation
- Suitable for small networks and enterprises
- Integrates well with SIEM tools

Limitations of Snort

- Requires manual tuning to reduce false positives
- Performance depends on hardware and rule complexity
- Signature-based detection may miss zero-day attacks
- Command-line interface has a learning curve

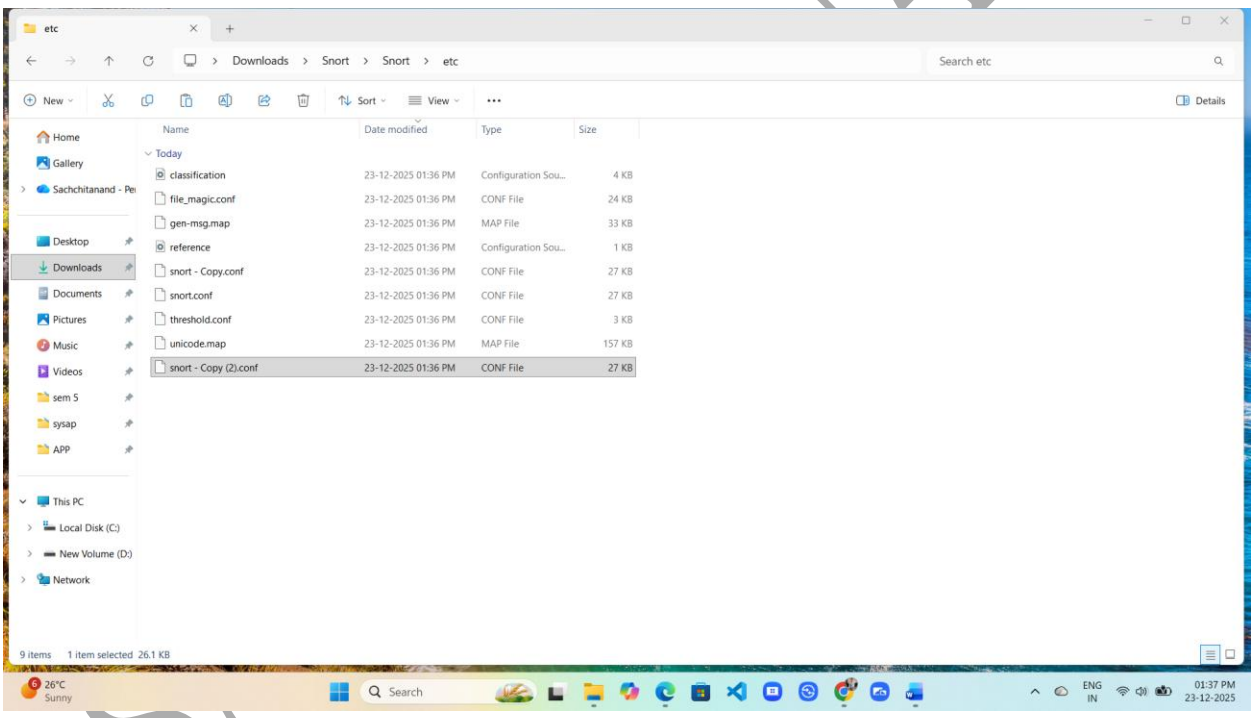
Snort Configuration

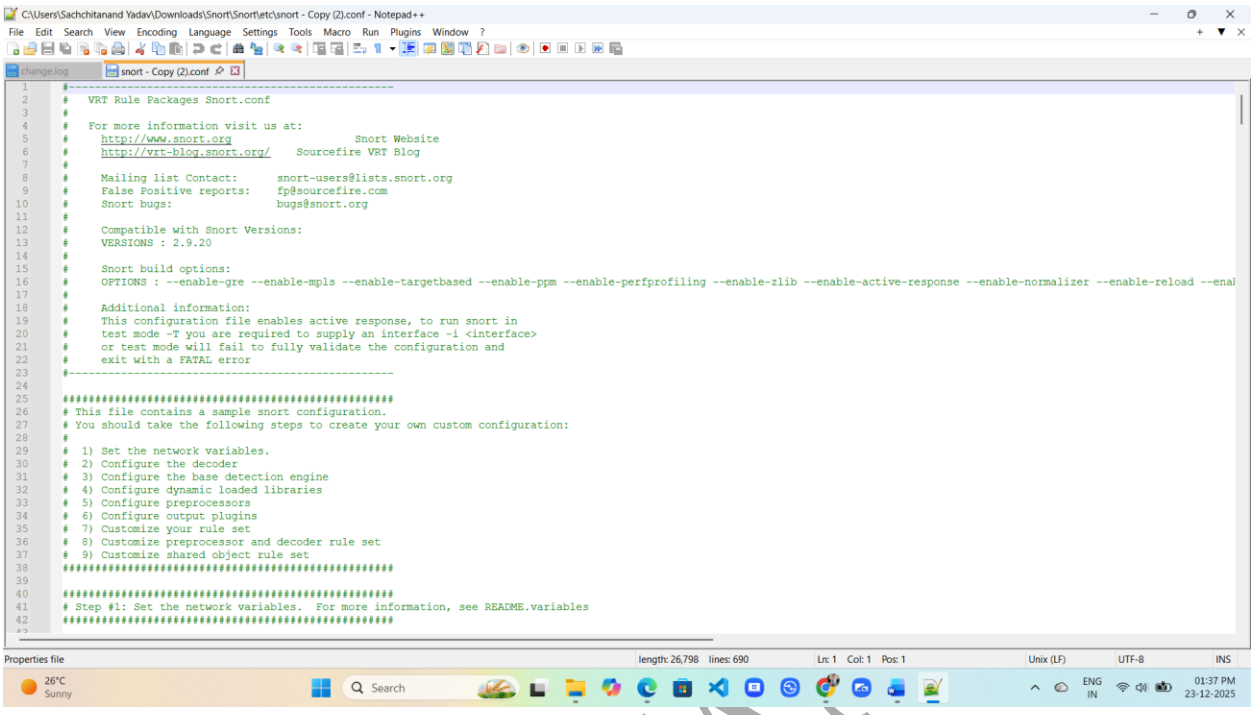
Download Link

https://www.snort.org/downloads

How to use it -:

- After installation snort, go to the snort location and open etc folder
- Copy snort file and paste it
- Now , open copied file in Notepad++



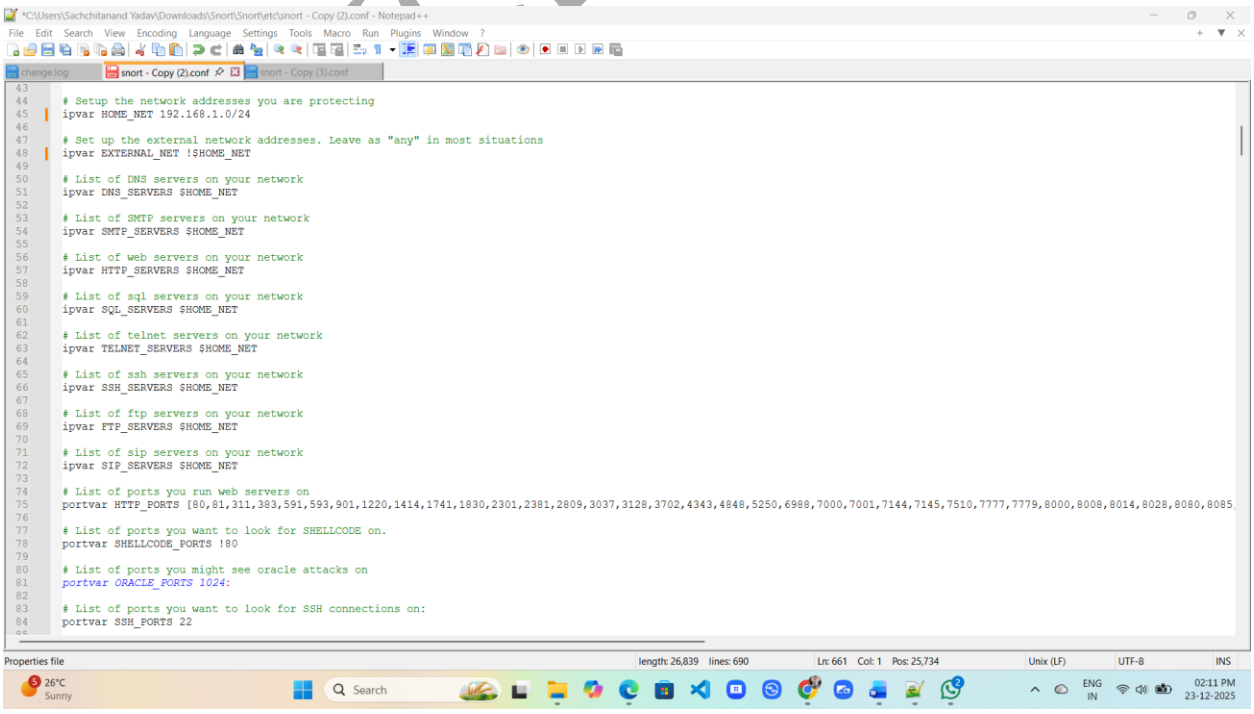


- **Go to the line number 45**

  And replace Any word to the ip range

- **Go to the line number 48**

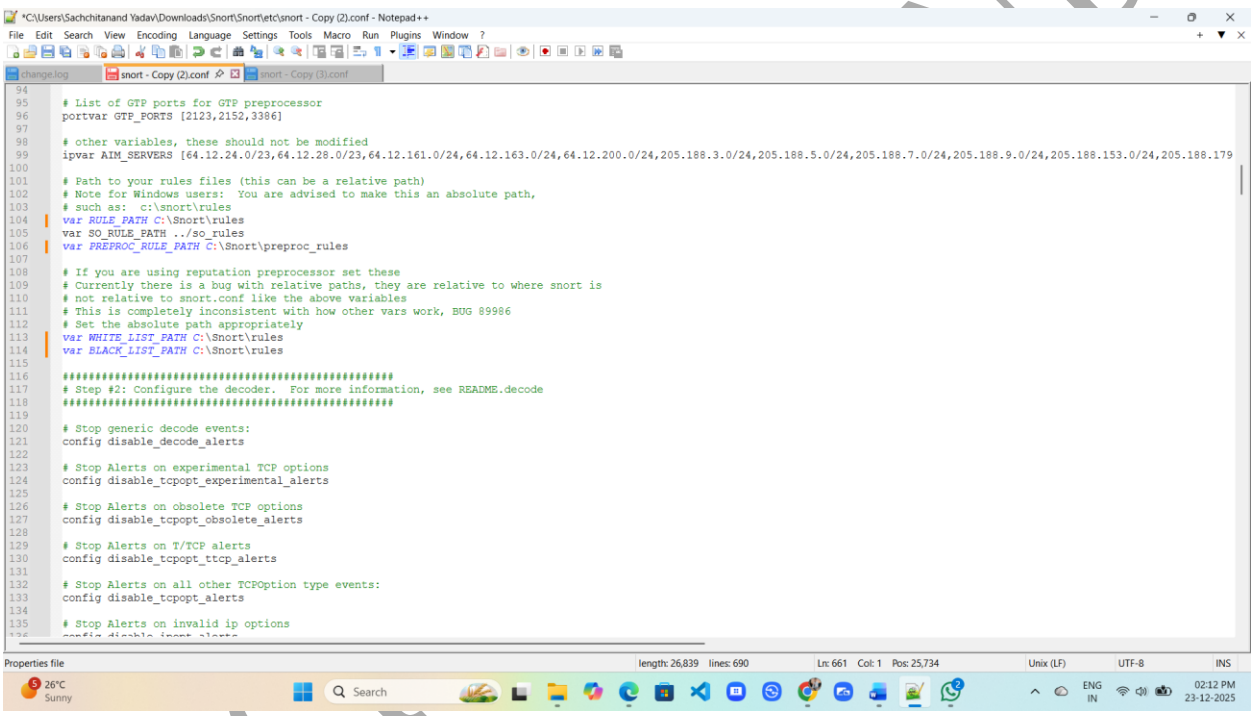
  Replace Any to the !$HOME_NET



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

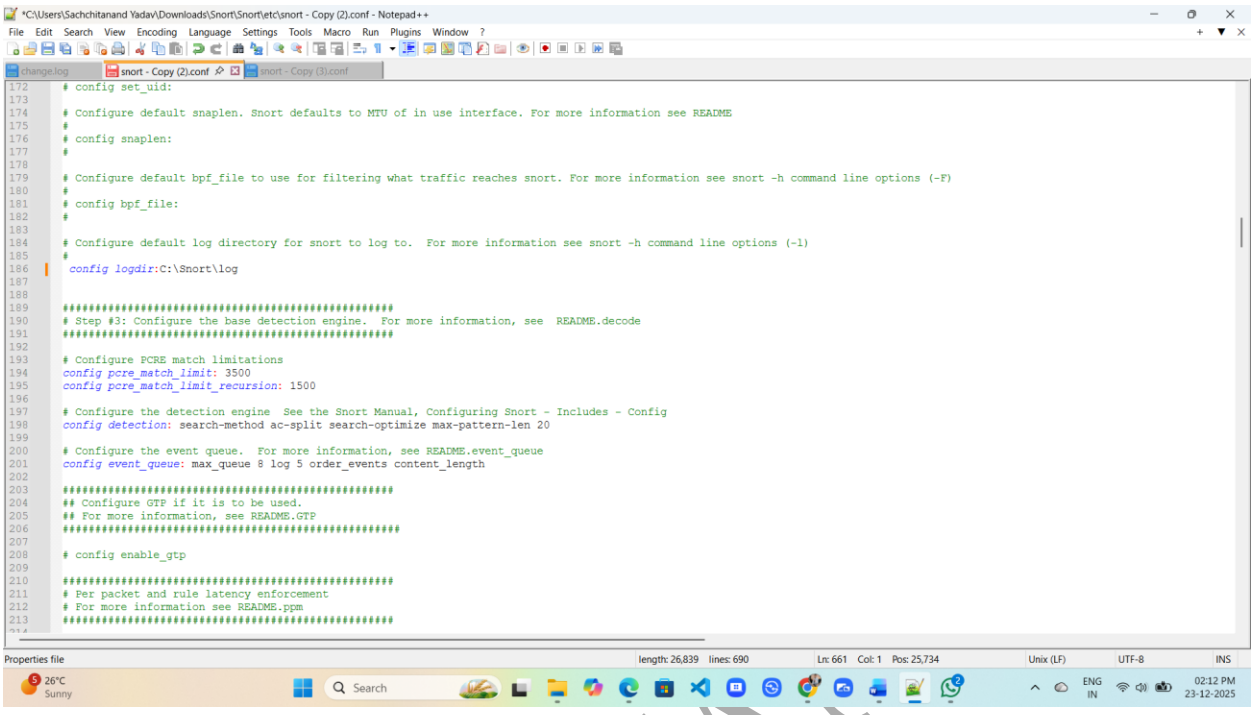
  C:\\Snort\rules
- Go to the line number 106 and set preproc_rules

  C:\Snort\preproc_rules
- go to the line number 113 and 114
- and add rules file path/location



- **Step 2**
  - go to the line number 186

    remove hash
- **Step 3**
  - Don’t do anything , no need to change



- **Step 4**
  - Go to the line number 247

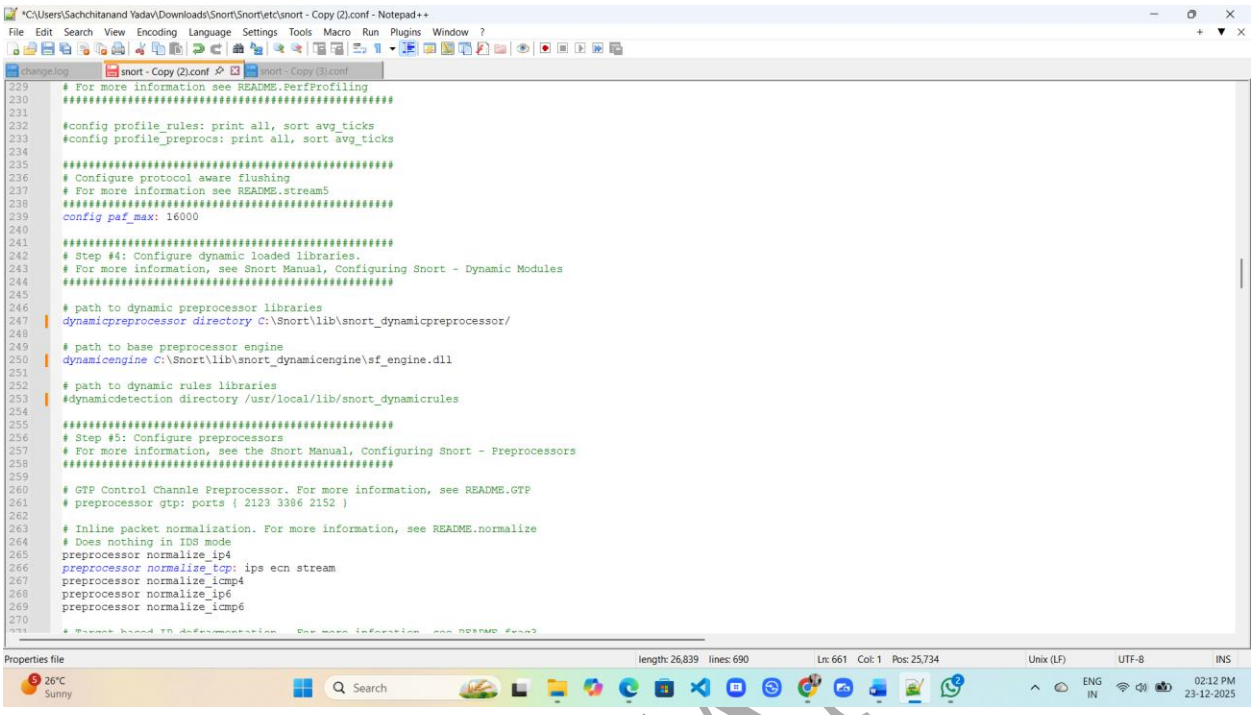
    Set snort_dynamicpreprocessor file path/location -: c:\Snort\lib\snort_preprocessor

  - Go to the line number number 250

    Set sf_engine.dll file location -: c:\Snort\lib\snort_dynamicengine/sf_engine.dll

  - Go to the line number 253

    Add # on front of the line number 253



• **Step 5**

  • Go to the line number 511 and 512

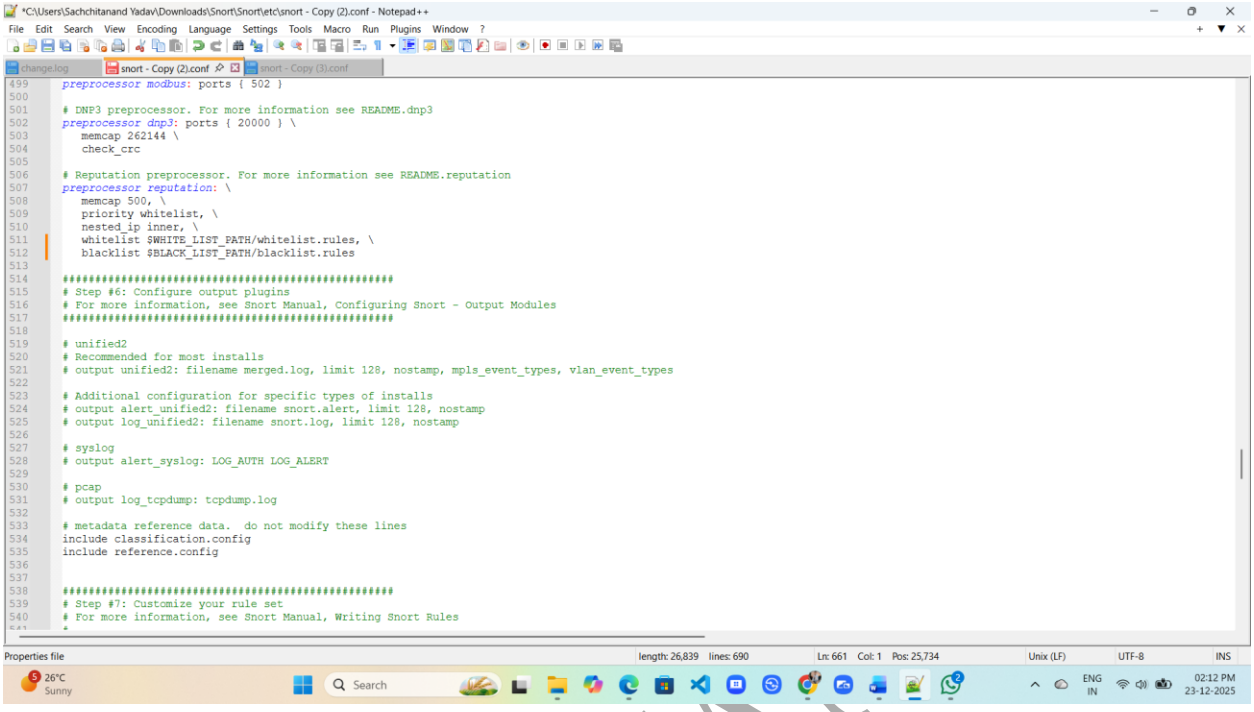
Note-: before changing on the line number 511 and 512 , firstly go to the snort >>rules folder and there is file in folder i.e blacklist.rules copy file and paste in same folder and rename copy file to the whitelist.rules

Replace white_list.rules to whitelist.rules on 511

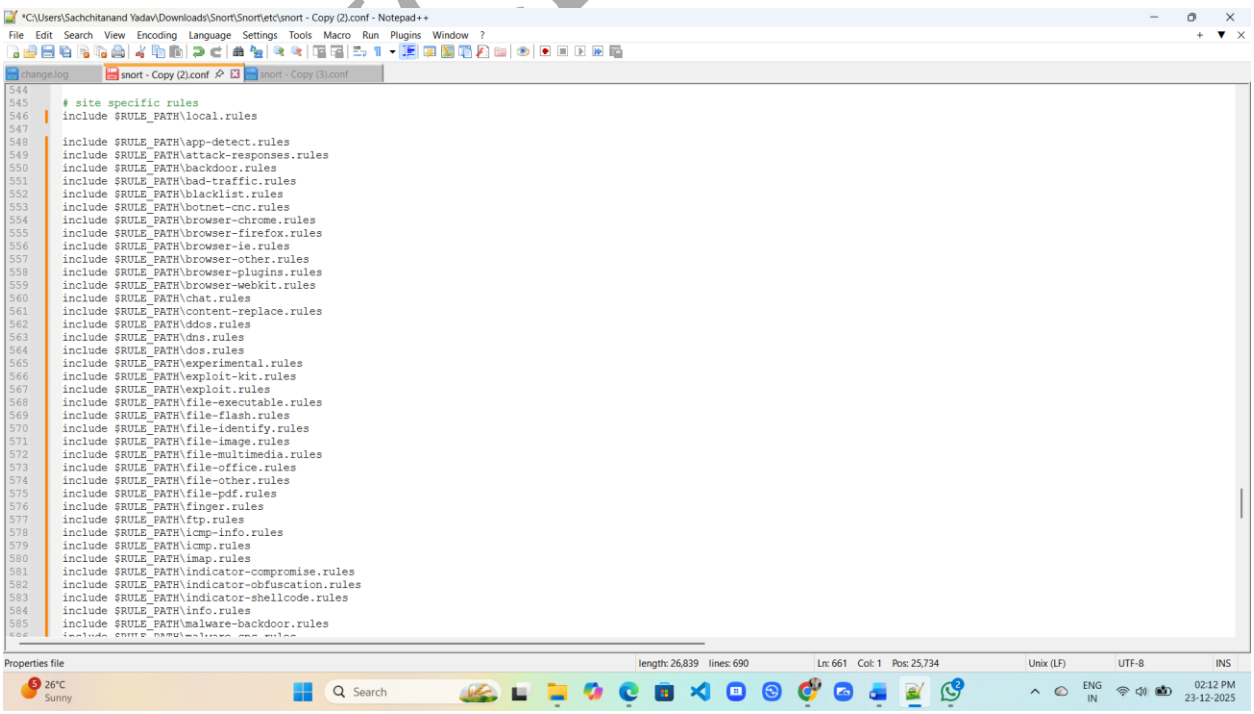
Replace black_list.rules to blacklist.rules on 512

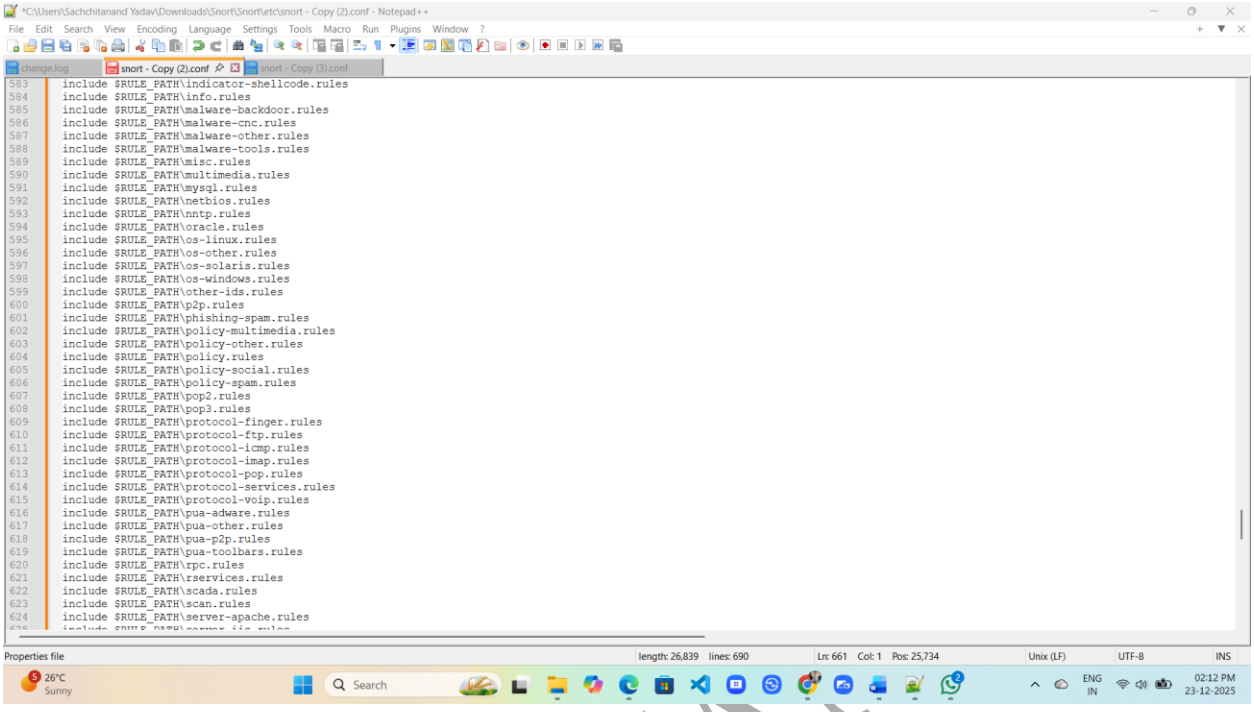
• **Step 6**

  • Don’t do anything , no need to change



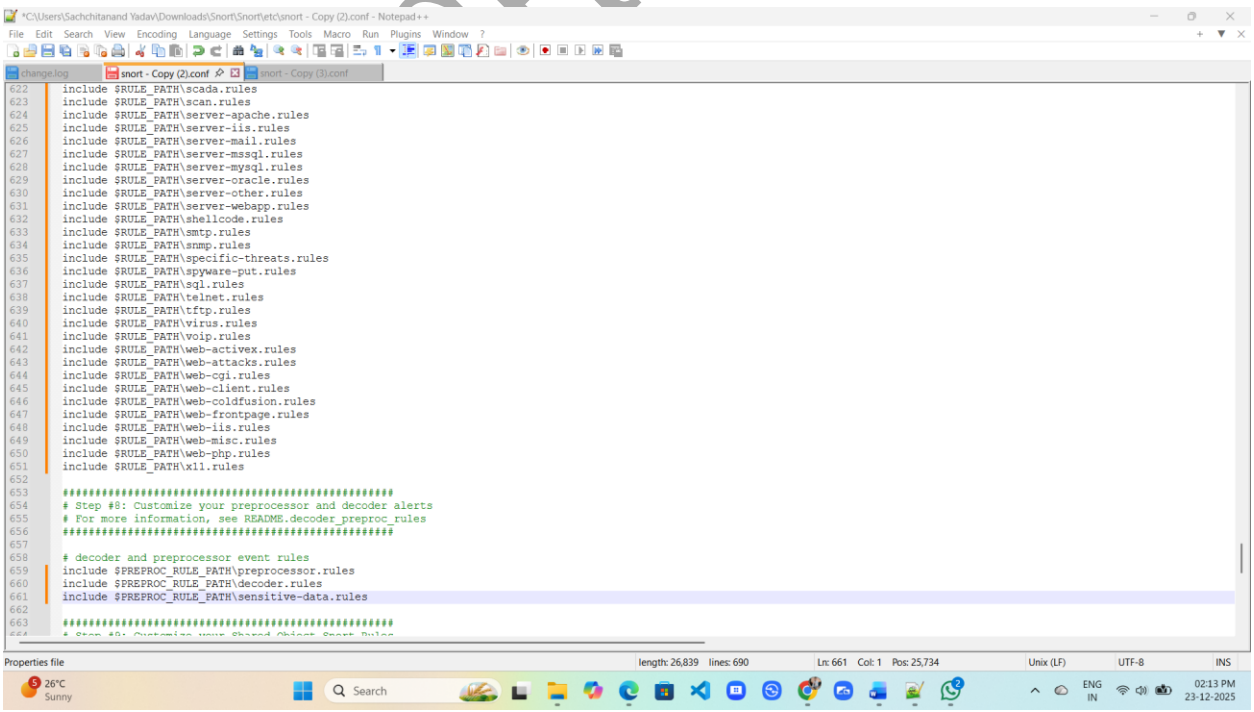
- **step 7**
  - go to the line number 546
  - replace “/” to “\” 546 upto 651





- **step 8**
  - go to the line number 659

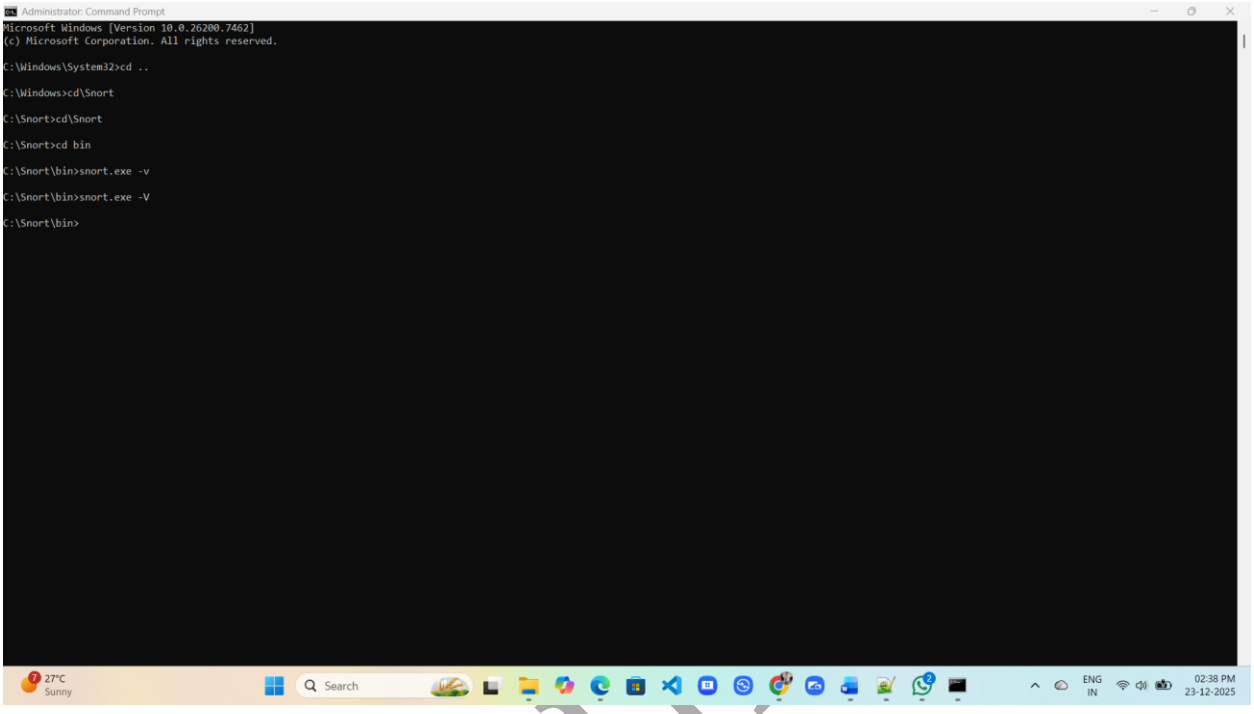
    remove hash line number 659 to 661
  - replace "/" to "\" 659 upto 661



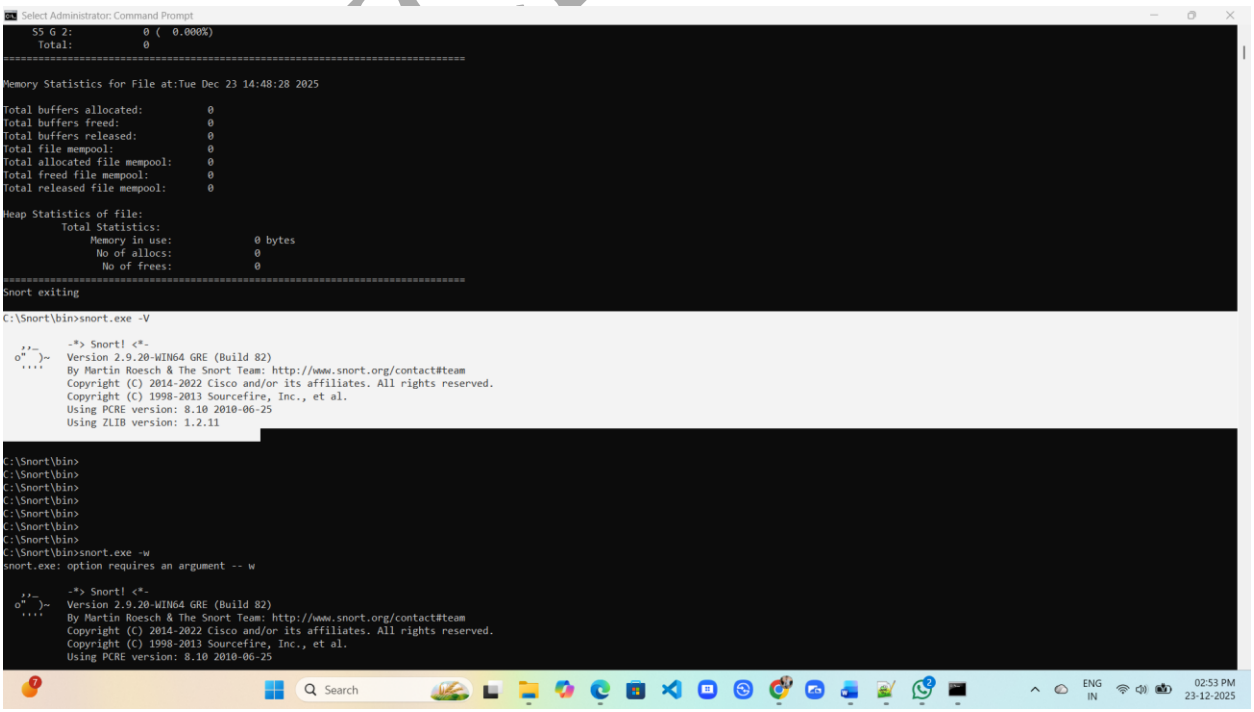
- **Configuration done**

Now open command line interface as a administrator

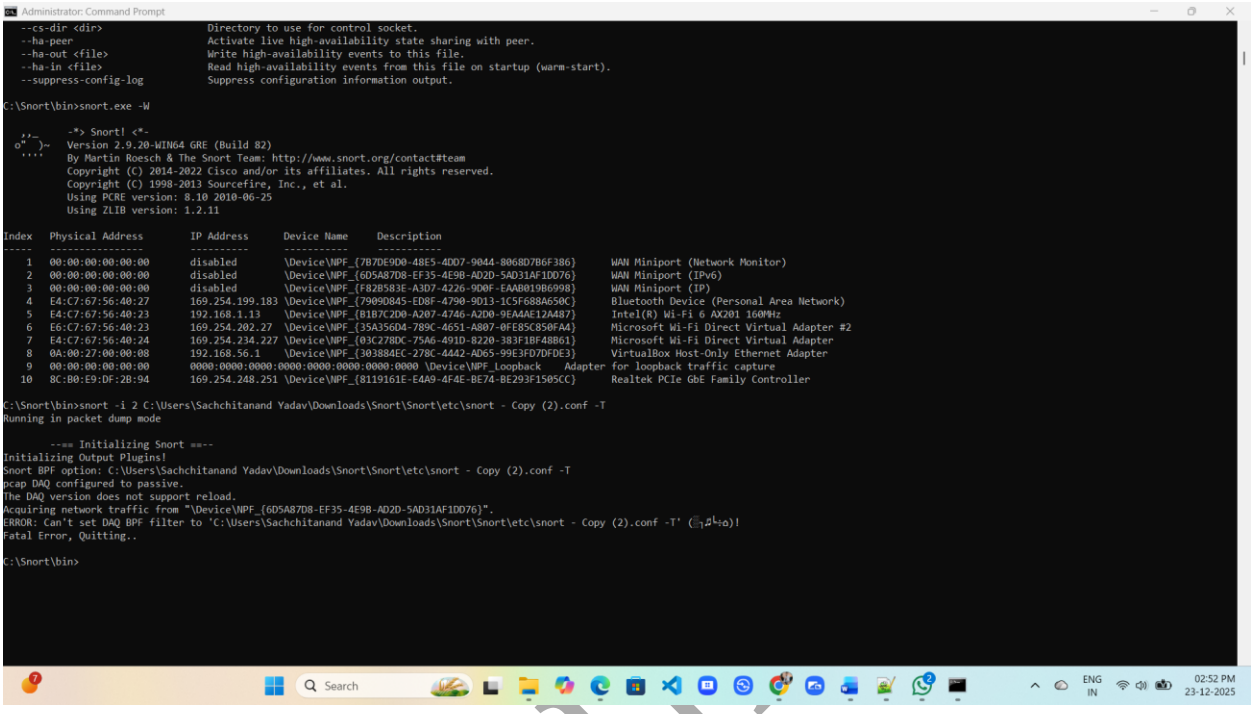
- Go to the Snort\bin Folder



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

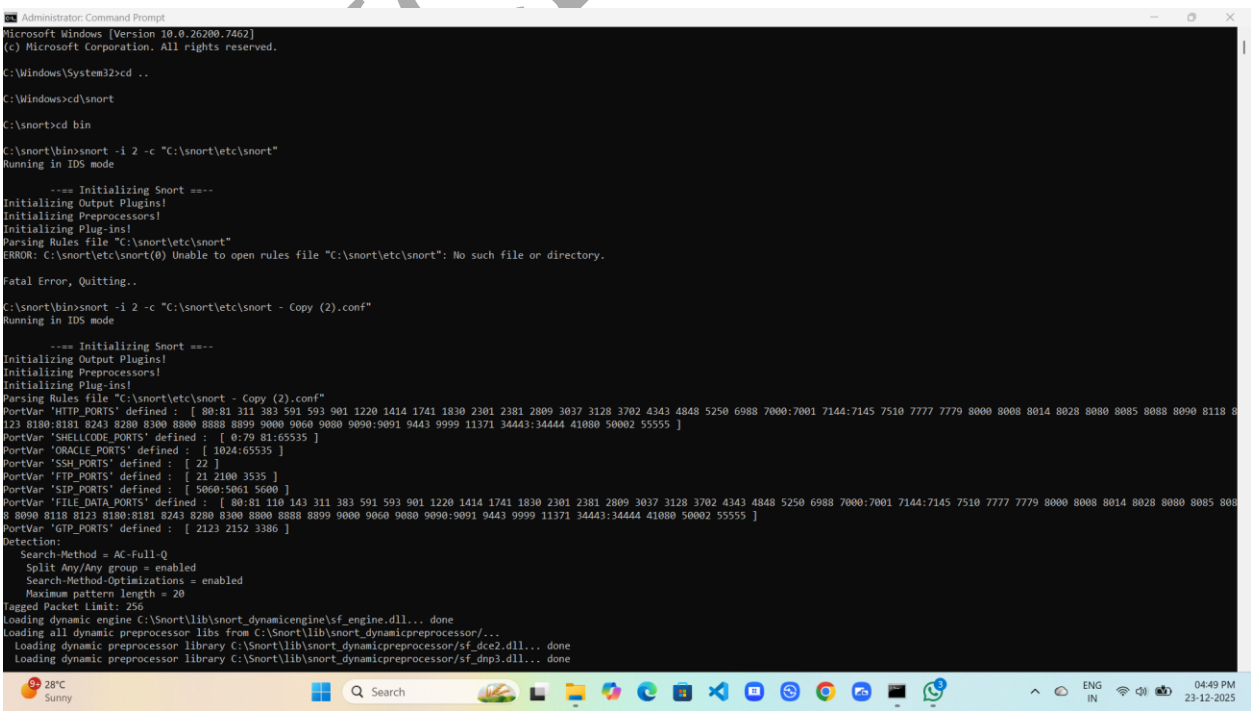


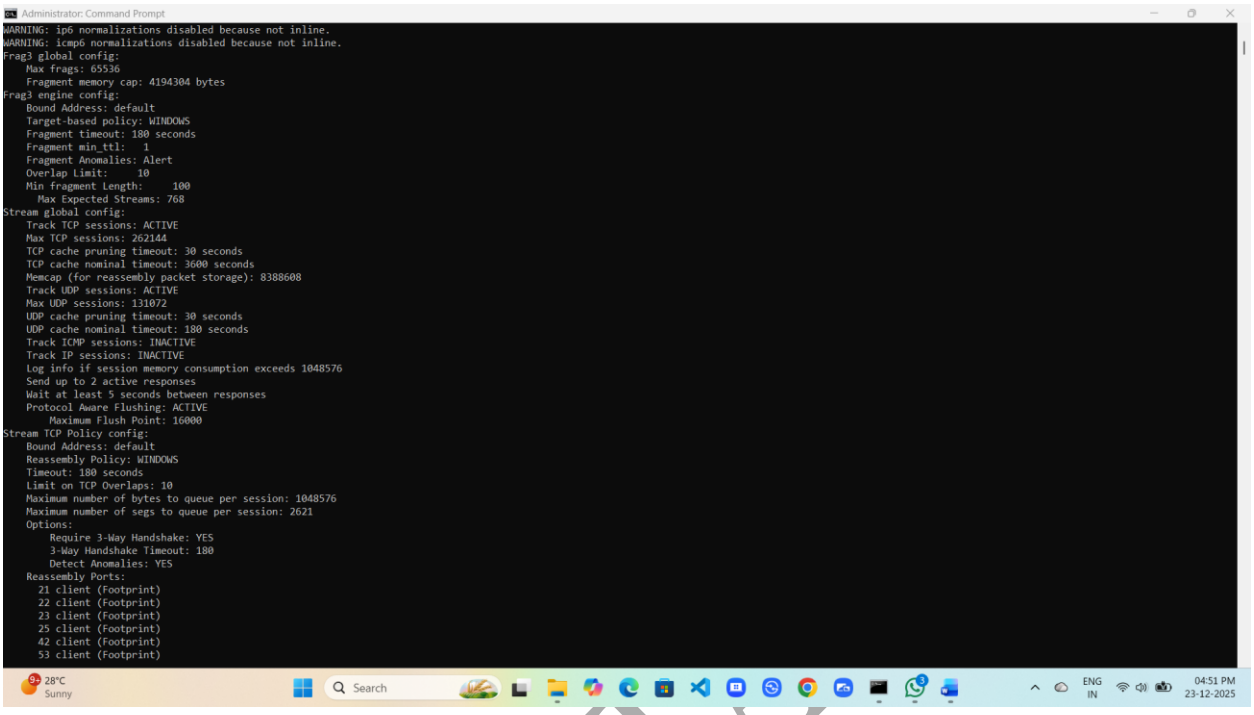
- use next command for finding network interface
- snort.exe –W

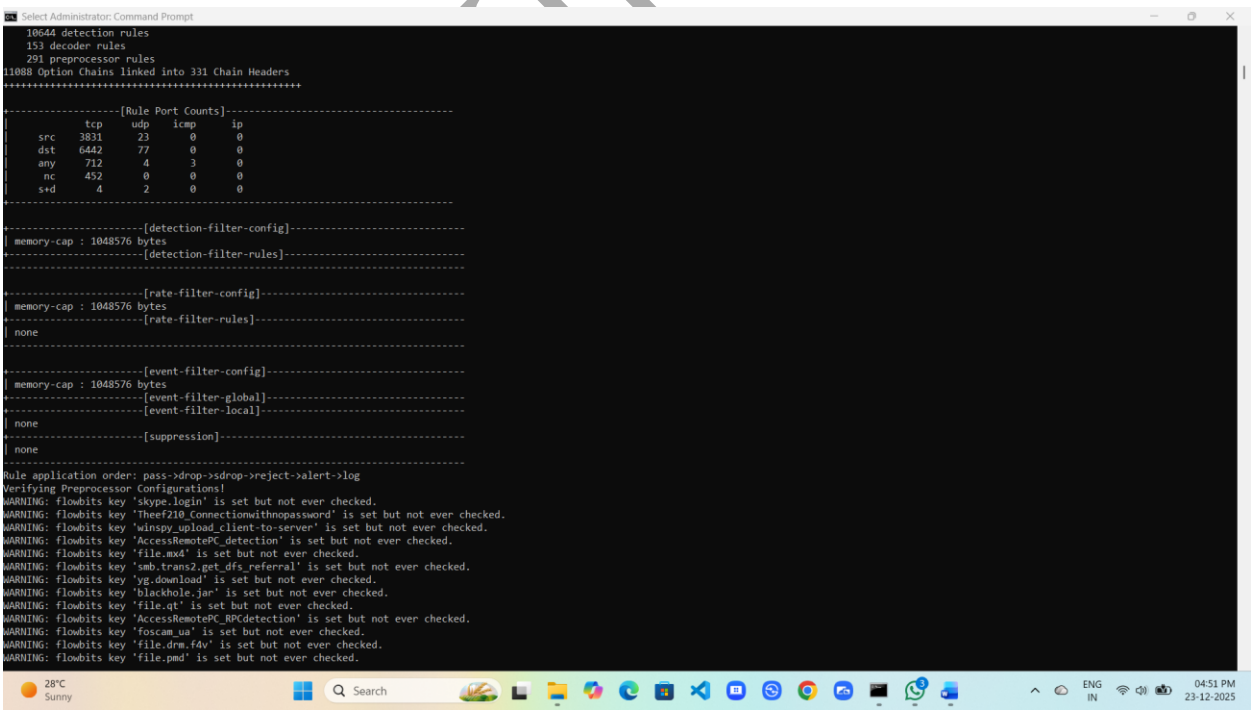


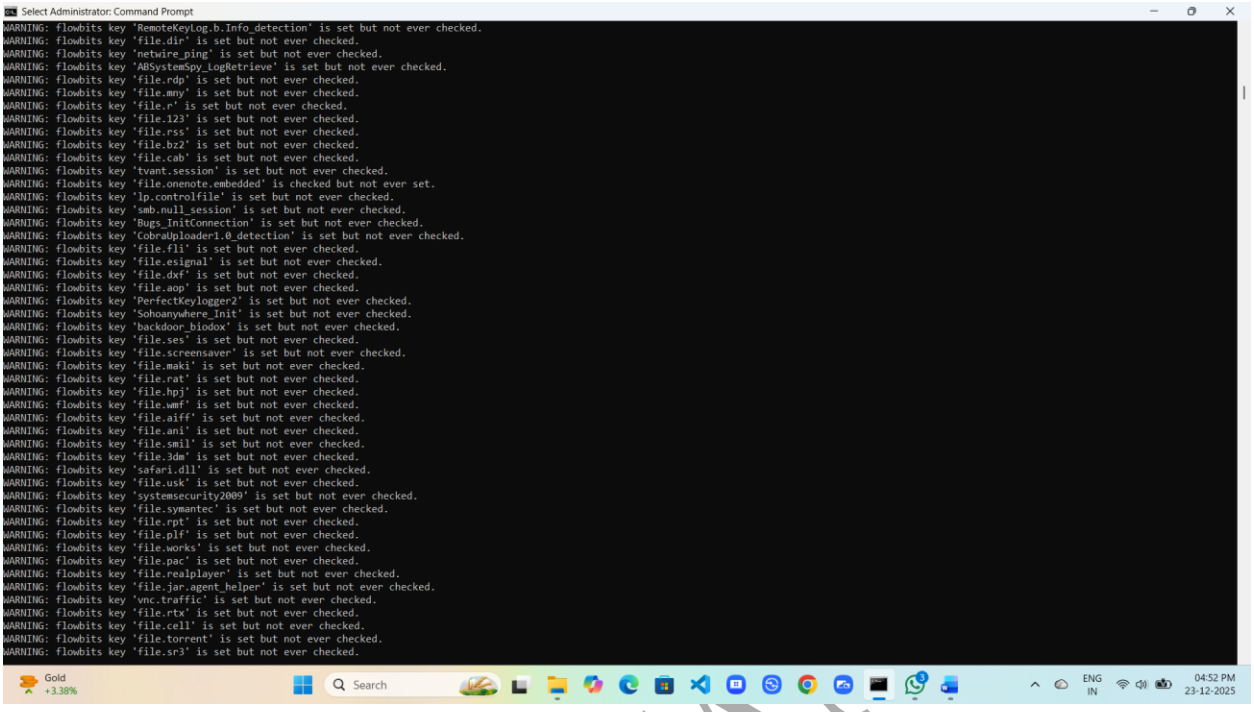
- use command for configuration testing

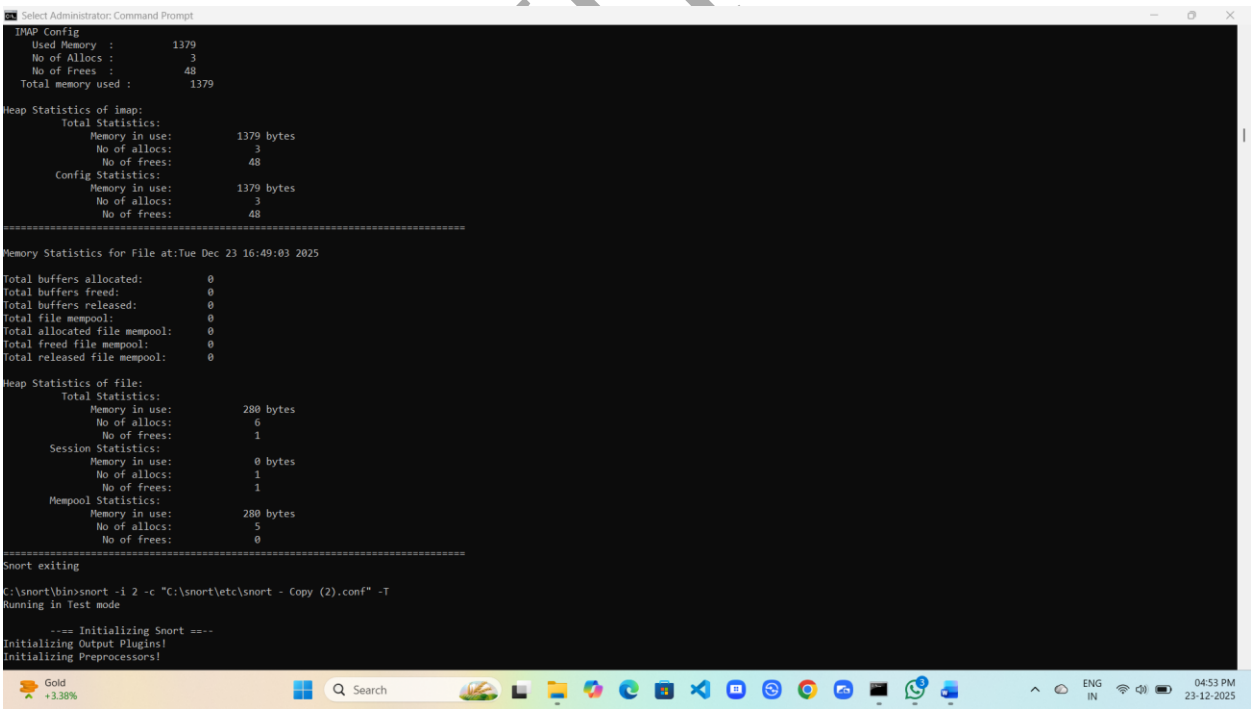
command -: snort.exe -i 2 -c "c:\snort\etc\file name "




Administrator: Command Prompt
WARNING: ip6 normalizations disabled because not inline.
WARNING: icmp6 normalizations disabled because not inline.
Frag3 global config:
Max frags: 65536
Fragment memory cap: 4194304 bytes
Frag3 engine config:
Bound Address: default
Target-based policy: WINDOWS
Fragment timeout: 180 seconds
Fragment min_ttl: 1
Fragment Anomalies: Alert
Overlap Limit: 10
Min fragment Length: 100
Max Expected Streams: 768
Stream global config:
Track TCP sessions: ACTIVE
Max TCP sessions: 262144
TCP cache pruning timeout: 30 seconds
TCP cache nominal timeout: 3600 seconds
Memcap (for reassembly packet storage): 8388608
Track UDP sessions: ACTIVE
Max UDP sessions: 131072
UDP cache pruning timeout: 30 seconds
UDP cache nominal timeout: 180 seconds
Track ICMP sessions: INACTIVE
Track IP sessions: INACTIVE
Log info if session memory consumption exceeds 1048576
Send up to 2 active responses
Wait at least 5 seconds between responses
Protocol Aware Flushing: ACTIVE
Maximum Flush Point: 16000
Stream TCP Policy config:
Bound Address: default
Reassembly Policy: WINDOWS
Timeout: 180 seconds
Limit on TCP Overlaps: 10
Maximum number of bytes to queue per session: 1048576
Maximum number of segs to queue per session: 2621
Options:
Require 3-Way Handshake: YES
3-Way Handshake Timeout: 180
Detect Anomalies: YES
Reassembly Ports:
21 client (Footprint)
22 client (Footprint)
23 client (Footprint)
25 client (Footprint)
42 client (Footprint)
53 client (Footprint)
28°C
Sunny
Search
ENG
IN
04:51 PM
23-12-2025

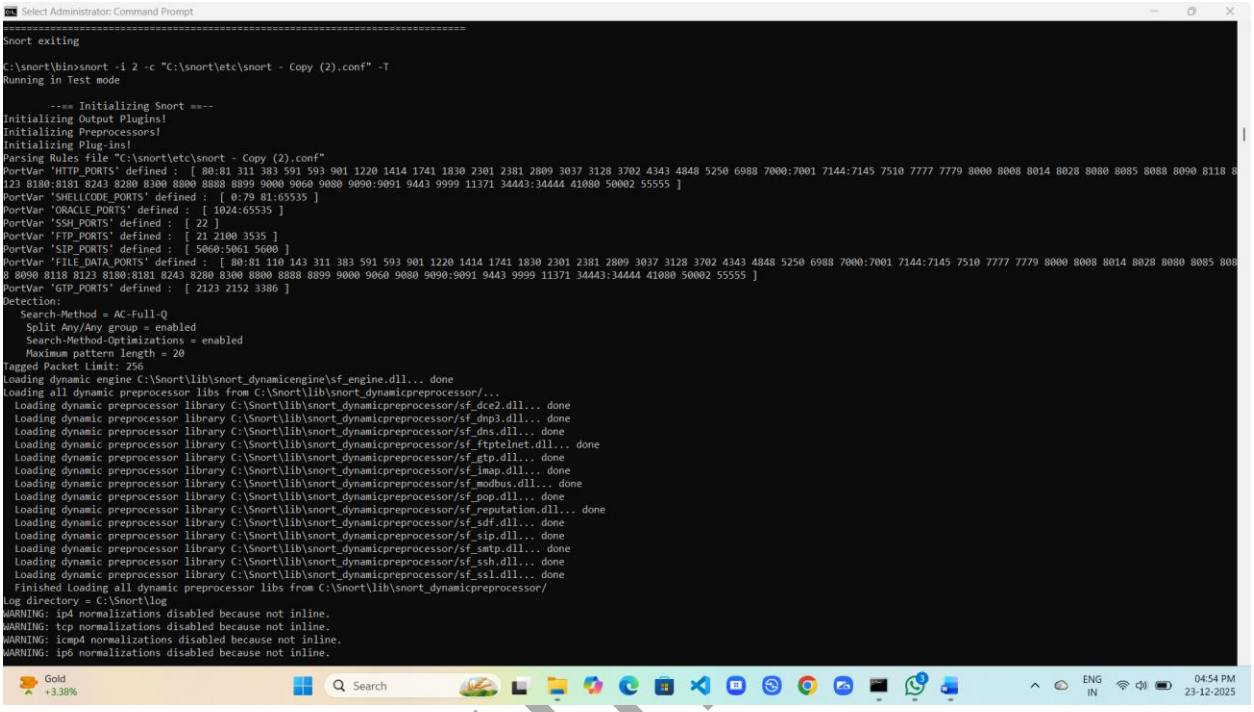

Select Administrator: Command Prompt
10644 detection rules
153 decoder rules
291 preprocessor rules
11088 Option Chains linked into 331 Chain Headers
[Rule Port Counts]
tcp udp icmp ip
src 3831 23 0 0
dst 6442 77 0 0
any 712 4 3 0
nc 452 0 0 0
s+d 4 2 0 0
[detection-filter-config]
memory-cap : 1048576 bytes
[detection-filter-rules]
none
[rate-filter-config]
memory-cap : 1048576 bytes
[rate-filter-rules]
none
[event-filter-config]
memory-cap : 1048576 bytes
[event-filter-global]
[event-filter-local]
none
[suppression]
none
Rule application order: pass->drop->sdrop->reject->alert->log
Verifying Preprocessor Configurations!
WARNING: flowbits key 'skype.login' is set but not ever checked.
WARNING: flowbits key 'Theef210_Connectionwithnopassword' is set but not ever checked.
WARNING: flowbits key 'winspy_upload_client-to-server' is set but not ever checked.
WARNING: flowbits key 'AccessRemotePC_detection' is set but not ever checked.
WARNING: flowbits key 'file.mx4' is set but not ever checked.
WARNING: flowbits key 'smb.trans2.get_dfs_referral' is set but not ever checked.
WARNING: flowbits key 'yg.download' is set but not ever checked.
WARNING: flowbits key 'blackhole.jar' is set but not ever checked.
WARNING: flowbits key 'file.qt' is set but not ever checked.
WARNING: flowbits key 'AccessRemotePC_RPCdetection' is set but not ever checked.
WARNING: flowbits key 'foscam_ua' is set but not ever checked.
WARNING: flowbits key 'file.drm.f4v' is set but not ever checked.
WARNING: flowbits key 'file.pmd' is set but not ever checked.
28°C
Sunny
Search
ENG
IN
04:51 PM
23-12-2025

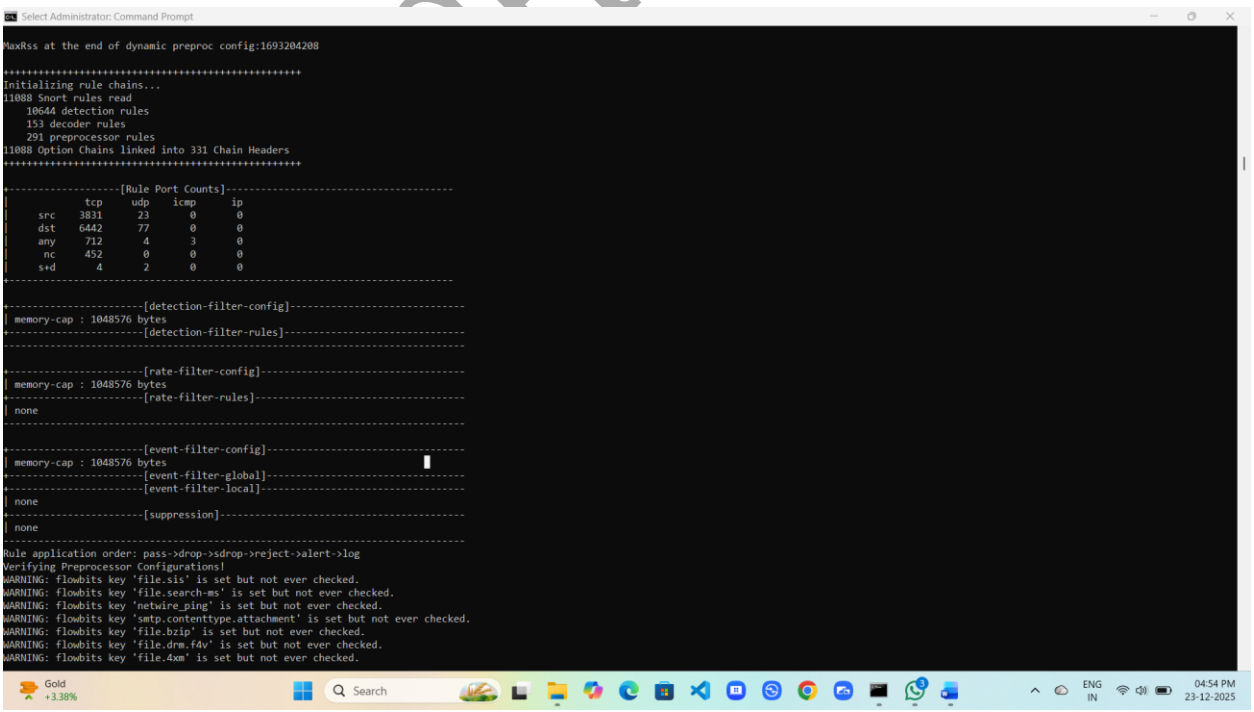

Select Administrator: Command Prompt
WARNING: flowbits key 'RemoteKeyLog.b.Info_detection' is set but not ever checked.
WARNING: flowbits key 'file.dir' is set but not ever checked.
WARNING: flowbits key 'netwire_ping' is set but not ever checked.
WARNING: flowbits key 'ABSystemSpy_LogRetrieve' is set but not ever checked.
WARNING: flowbits key 'file.rdp' is set but not ever checked.
WARNING: flowbits key 'file.mny' is set but not ever checked.
WARNING: flowbits key 'file.r' is set but not ever checked.
WARNING: flowbits key 'file.123' is set but not ever checked.
WARNING: flowbits key 'file.rss' is set but not ever checked.
WARNING: flowbits key 'file.bz2' is set but not ever checked.
WARNING: flowbits key 'file.cab' is set but not ever checked.
WARNING: flowbits key 'tvant.session' is set but not ever checked.
WARNING: flowbits key 'file.onenote.embedded' is checked but not ever set.
WARNING: flowbits key 'lp.controlfile' is set but not ever checked.
WARNING: flowbits key 'smb.null_session' is set but not ever checked.
WARNING: flowbits key 'Bugs_InitConnection' is set but not ever checked.
WARNING: flowbits key 'CobraUploader1.0_detection' is set but not ever checked.
WARNING: flowbits key 'file.fli' is set but not ever checked.
WARNING: flowbits key 'file.esignal' is set but not ever checked.
WARNING: flowbits key 'file.dxf' is set but not ever checked.
WARNING: flowbits key 'file.aop' is set but not ever checked.
WARNING: flowbits key 'PerfectKeylogger2' is set but not ever checked.
WARNING: flowbits key 'Sohoanywhere_Init' is set but not ever checked.
WARNING: flowbits key 'backdoor_biodox' is set but not ever checked.
WARNING: flowbits key 'file.ses' is set but not ever checked.
WARNING: flowbits key 'file.screensaver' is set but not ever checked.
WARNING: flowbits key 'file.maki' is set but not ever checked.
WARNING: flowbits key 'file.rat' is set but not ever checked.
WARNING: flowbits key 'file.hpj' is set but not ever checked.
WARNING: flowbits key 'file.wmf' is set but not ever checked.
WARNING: flowbits key 'file.aiff' is set but not ever checked.
WARNING: flowbits key 'file.ani' is set but not ever checked.
WARNING: flowbits key 'file.smil' is set but not ever checked.
WARNING: flowbits key 'file.3dm' is set but not ever checked.
WARNING: flowbits key 'safari.dll' is set but not ever checked.
WARNING: flowbits key 'file.usk' is set but not ever checked.
WARNING: flowbits key 'systemsecurity2009' is set but not ever checked.
WARNING: flowbits key 'file.symantec' is set but not ever checked.
WARNING: flowbits key 'file.rpt' is set but not ever checked.
WARNING: flowbits key 'file.plf' is set but not ever checked.
WARNING: flowbits key 'file.works' is set but not ever checked.
WARNING: flowbits key 'file.pac' is set but not ever checked.
WARNING: flowbits key 'file.realplayer' is set but not ever checked.
WARNING: flowbits key 'file.jar.agent_helper' is set but not ever checked.
WARNING: flowbits key 'vnc.traffic' is set but not ever checked.
WARNING: flowbits key 'file.rtx' is set but not ever checked.
WARNING: flowbits key 'file.cell' is set but not ever checked.
WARNING: flowbits key 'file.torrent' is set but not ever checked.
WARNING: flowbits key 'file.sr3' is set but not ever checked.
Gold
+3.38%
Search
ENG
IN
04:52 PM
23-12-2025

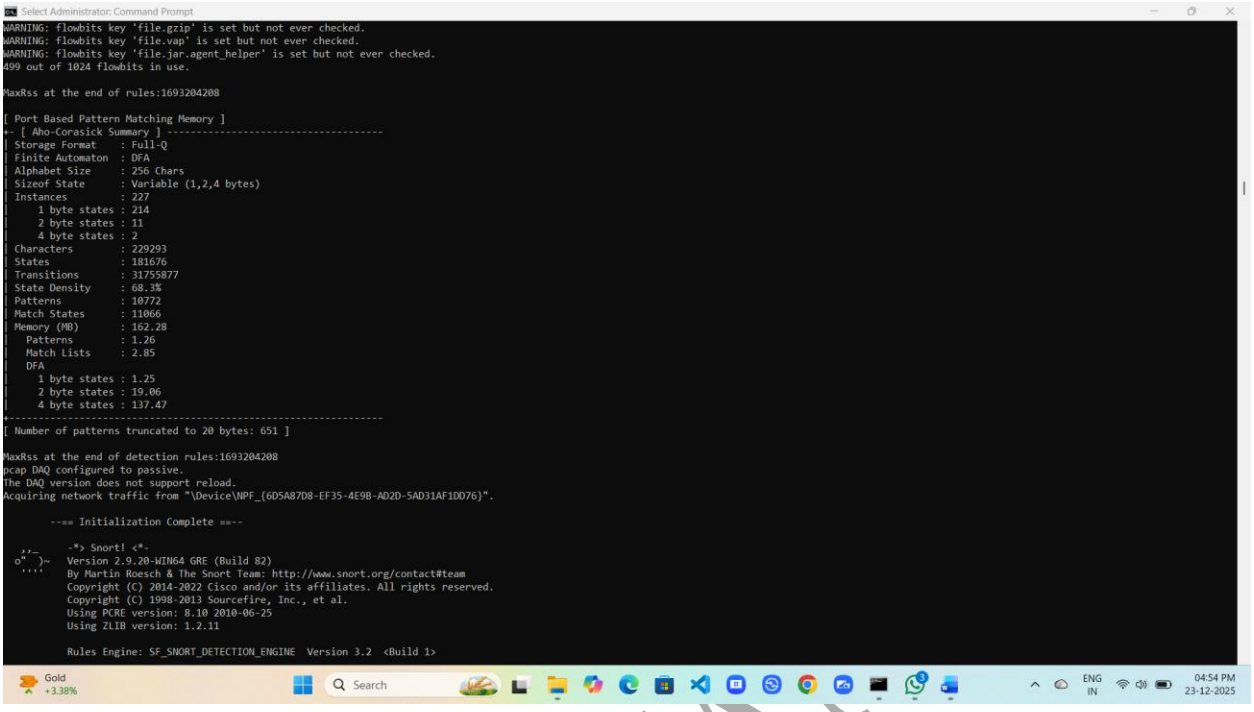

Select Administrator: Command Prompt
IMAP Config
Used Memory : 1379
No of Allocs : 3
No of Frees : 48
Total memory used : 1379
Heap Statistics of imap:
Total Statistics:
Memory in use: 1379 bytes
No of allocs: 3
No of frees: 48
Config Statistics:
Memory in use: 1379 bytes
No of allocs: 3
No of frees: 48
Memory Statistics for File at:Tue Dec 23 16:49:03 2025
Total buffers allocated: 0
Total buffers freed: 0
Total buffers released: 0
Total file mempool: 0
Total allocated file mempool: 0
Total freed file mempool: 0
Total released file mempool: 0
Heap Statistics of file:
Total Statistics:
Memory in use: 280 bytes
No of allocs: 6
No of frees: 1
Session Statistics:
Memory in use: 0 bytes
No of allocs: 1
No of frees: 1
Mempool Statistics:
Memory in use: 280 bytes
No of allocs: 5
No of frees: 0
Snort exiting
C:\snort\bin>snort -i 2 -c "C:\snort\etc\snort - Copy (2).conf" -T
Running in Test mode
--== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Gold
+3.38%
Search
ENG
IN
04:53 PM
23-12-2025

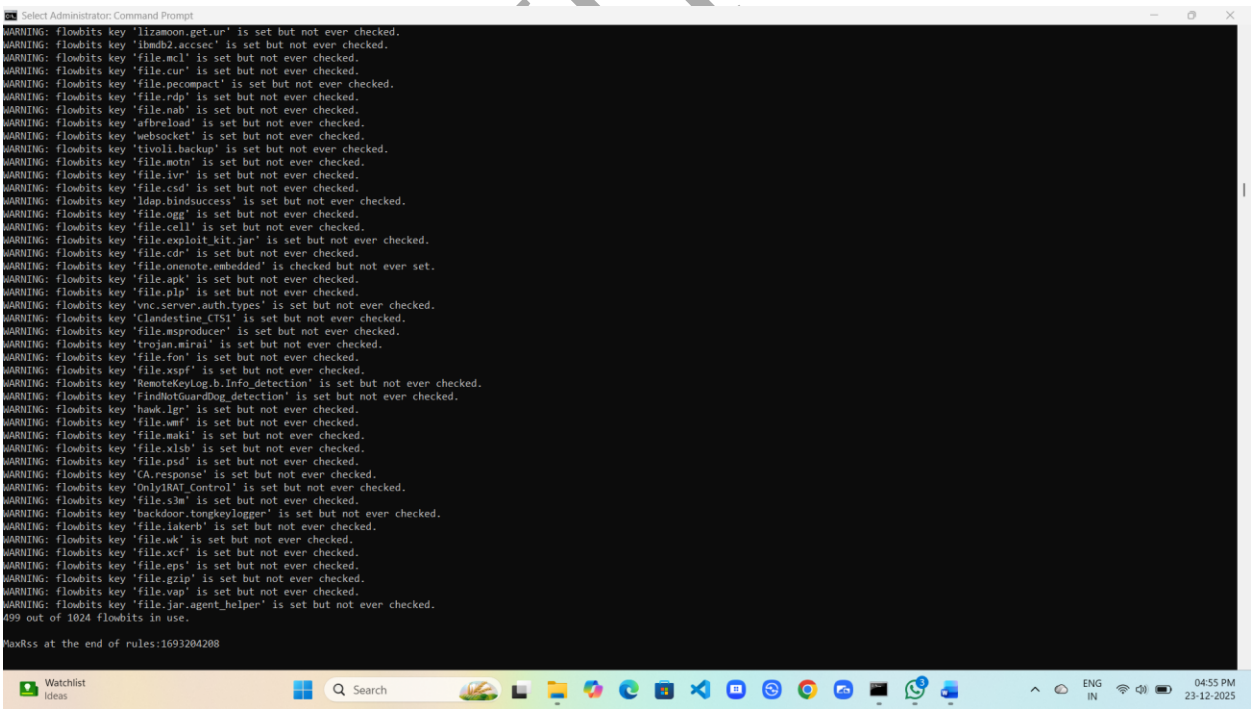
• use command for configuration testing

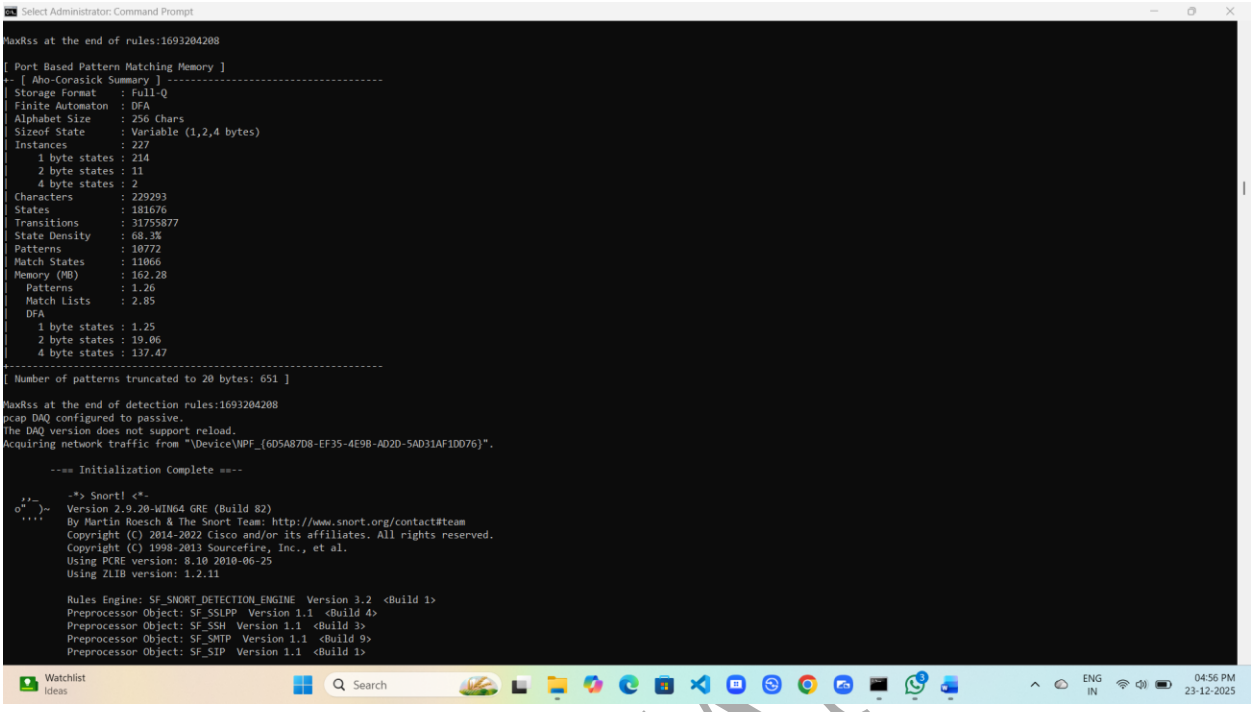
command -: snort.exe -i 2 -c “c:\Snort\etc\file name “ -T



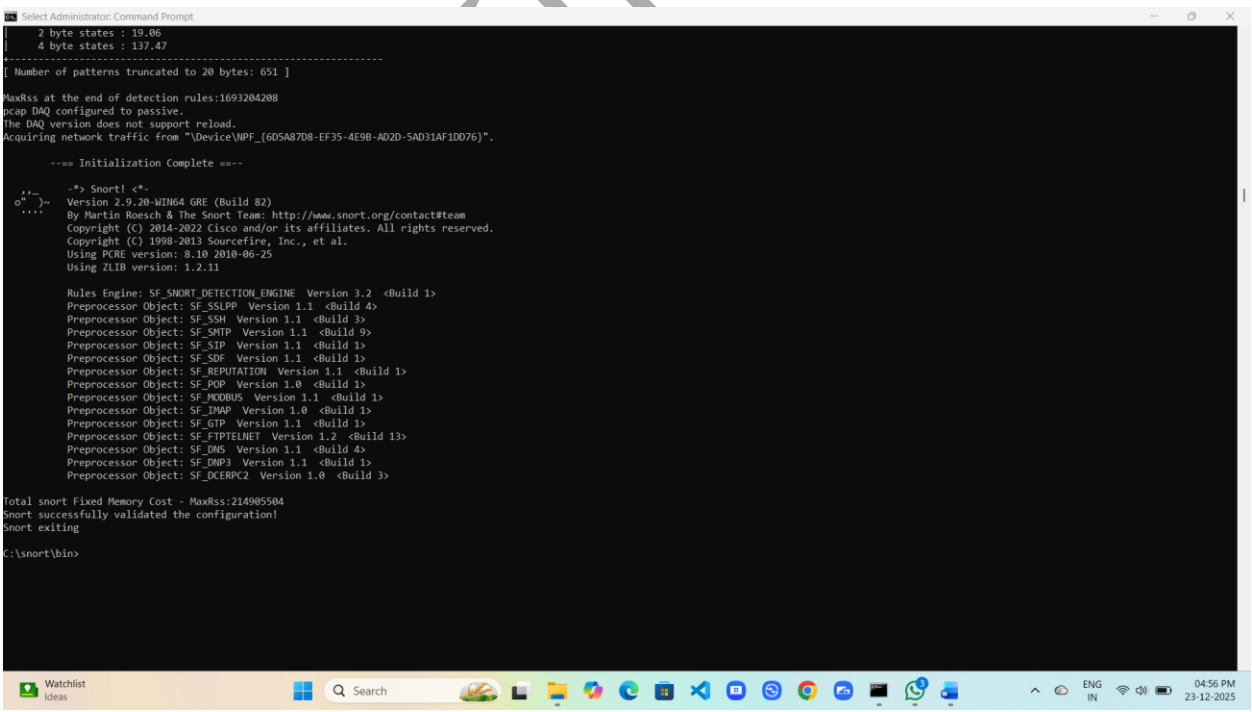



Select Administrator: Command Prompt
WARNING: flowbits key 'file.gzip' is set but not ever checked.
WARNING: flowbits key 'file.vap' is set but not ever checked.
WARNING: flowbits key 'file.jar.agent_helper' is set but not ever checked.
499 out of 1024 flowbits in use.
MaxRss at the end of rules:1693204208
[Port Based Pattern Matching Memory]
[Aho-Corasick Summary]
Storage Format : Full-Q
Finite Automaton : DFA
Alphabet Size : 256 Chars
Sizeof State : Variable (1,2,4 bytes)
Instances : 227
1 byte states : 214
2 byte states : 11
4 byte states : 2
Characters : 229293
States : 181676
Transitions : 31755877
State Density : 68.3%
Patterns : 10772
Match States : 11066
Memory (MB) : 162.28
Patterns : 1.26
Match Lists : 2.85
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:1693204208
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{6D5A87D8-EF35-4E9B-AD2D-5AD31AF1DD76}".
--== 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>
Gold +3.38%
Search
ENG IN
04:54 PM 23-12-2025


Select Administrator: Command Prompt
WARNING: flowbits key 'lizamoon.get.ur' is set but not ever checked.
WARNING: flowbits key 'ibmdb2.accsec' is set but not ever checked.
WARNING: flowbits key 'file.mcl' is set but not ever checked.
WARNING: flowbits key 'file.cur' is set but not ever checked.
WARNING: flowbits key 'file.pecompact' is set but not ever checked.
WARNING: flowbits key 'file.rdp' is set but not ever checked.
WARNING: flowbits key 'file.nab' is set but not ever checked.
WARNING: flowbits key 'afbreload' is set but not ever checked.
WARNING: flowbits key 'websocket' is set but not ever checked.
WARNING: flowbits key 'tivoli.backup' is set but not ever checked.
WARNING: flowbits key 'file.motn' is set but not ever checked.
WARNING: flowbits key 'file.ivr' is set but not ever checked.
WARNING: flowbits key 'file.csd' is set but not ever checked.
WARNING: flowbits key 'ldap.bindsuccess' is set but not ever checked.
WARNING: flowbits key 'file.ogg' is set but not ever checked.
WARNING: flowbits key 'file.cell' is set but not ever checked.
WARNING: flowbits key 'file.exploit_kit.jar' is set but not ever checked.
WARNING: flowbits key 'file.cdr' is set but not ever checked.
WARNING: flowbits key 'file.onenote.embedded' is checked but not ever set.
WARNING: flowbits key 'file.apk' is set but not ever checked.
WARNING: flowbits key 'file.plp' is set but not ever checked.
WARNING: flowbits key 'vnc.server.auth.types' is set but not ever checked.
WARNING: flowbits key 'Clandestine_CTS1' is set but not ever checked.
WARNING: flowbits key 'file.msproducer' is set but not ever checked.
WARNING: flowbits key 'trojan.mirai' is set but not ever checked.
WARNING: flowbits key 'file.fon' is set but not ever checked.
WARNING: flowbits key 'file.xspf' is set but not ever checked.
WARNING: flowbits key 'RemoteKeyLog.b.Info_detection' is set but not ever checked.
WARNING: flowbits key 'FindNotGuardDog_detection' is set but not ever checked.
WARNING: flowbits key 'hawk.lgr' is set but not ever checked.
WARNING: flowbits key 'file.wmf' is set but not ever checked.
WARNING: flowbits key 'file.maki' is set but not ever checked.
WARNING: flowbits key 'file.xlsb' is set but not ever checked.
WARNING: flowbits key 'file.psd' is set but not ever checked.
WARNING: flowbits key 'CA.response' is set but not ever checked.
WARNING: flowbits key 'Only1RAT_Control' is set but not ever checked.
WARNING: flowbits key 'file.s3m' is set but not ever checked.
WARNING: flowbits key 'backdoor.tongkeylogger' is set but not ever checked.
WARNING: flowbits key 'file.iakerb' is set but not ever checked.
WARNING: flowbits key 'file.wk' is set but not ever checked.
WARNING: flowbits key 'file.xcf' is set but not ever checked.
WARNING: flowbits key 'file.eps' is set but not ever checked.
WARNING: flowbits key 'file.gzip' is set but not ever checked.
WARNING: flowbits key 'file.vap' is set but not ever checked.
WARNING: flowbits key 'file.jar.agent_helper' is set but not ever checked.
499 out of 1024 flowbits in use.
MaxRss at the end of rules:1693204208
Watchlist Ideas
Search
ENG IN
04:55 PM 23-12-2025



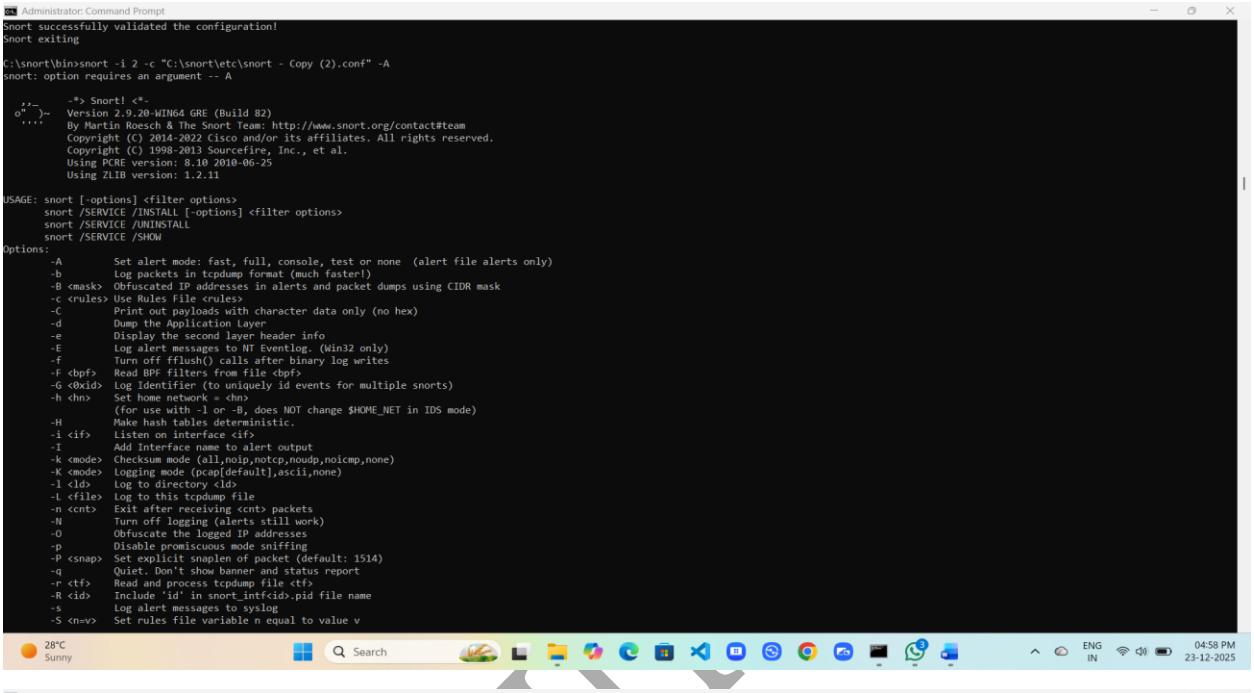
- **Configuration successful validate**

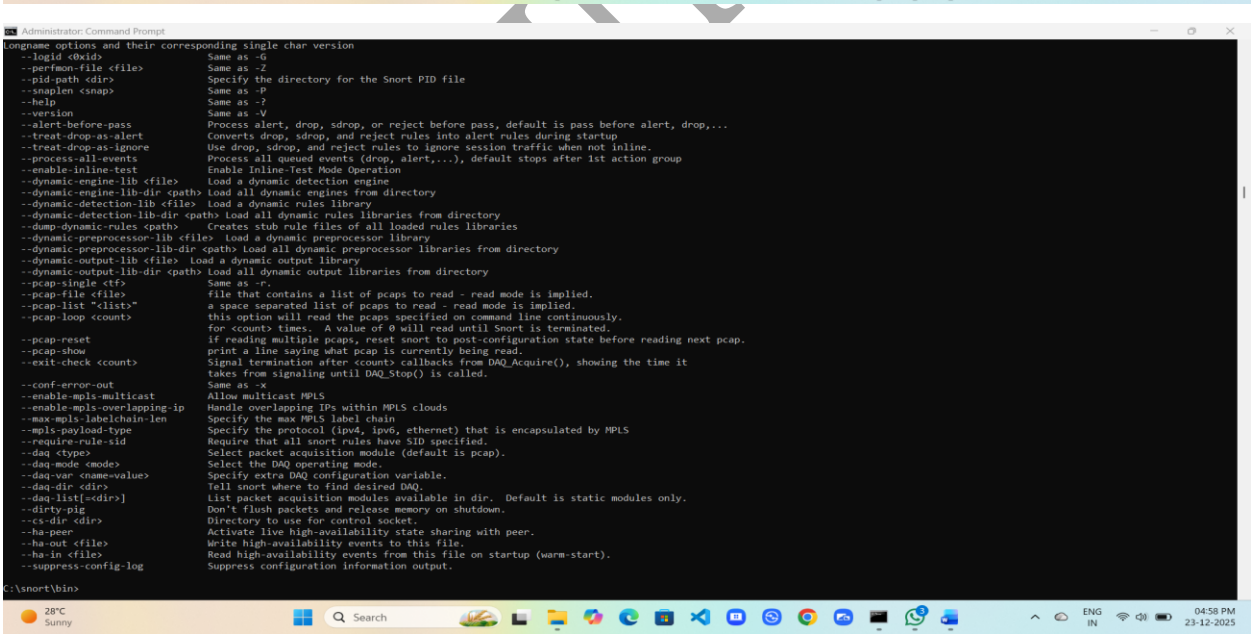


• Type command to start snort

Command -: snort.exe -i 2 -c"c:\snort\etc\file name " -A console

• Here snort started





Conclusion

Snort remains one of the most trusted and widely deployed intrusion detection and prevention tools in cybersecurity. Its flexibility, open-source nature, and real-time detection capabilities make it an essential component of network defense strategies.

Evading IDS, Firewalls, and Honeypots – Countermeasures

Evading techniques are used by attackers to bypass security mechanisms like IDS, firewalls, and honeypots. To stop this sneaky business, defenders deploy **countermeasures**—basically smarter locks, sharper eyes, and better bait. Old-school principles, modern execution.

---

1. Countermeasures Against IDS Evasion

IDS watches the road. Attackers try to drive slow, sideways, or in pieces.

Intrusion Detection Systems can be evaded using packet fragmentation, obfuscation, or encryption. To counter these evasion techniques:

- **Traffic normalization** is used to reassemble fragmented packets before analysis, eliminating ambiguity.
- **Signature-based detection updates** ensure the IDS recognizes new attack patterns.
- **Anomaly-based detection** identifies unusual traffic behavior rather than relying only on known signatures.
- **Encrypted traffic inspection** helps analyze malicious activity hidden inside SSL/TLS traffic.
- **Log correlation and centralized monitoring** combine multiple alerts to detect stealthy attacks.

Reality check: IDS that isn't updated is just a very expensive diary.

---

2. Countermeasures Against Firewall Evasion

Firewalls are gatekeepers. Attackers wear disguises.

Attackers often tunnel traffic through allowed ports or spoof protocols to bypass firewalls. Effective countermeasures include:

- **Stateful inspection firewalls** track session states, not just individual packets.
- **Deep Packet Inspection (DPI)** examines packet payloads, not only headers.
- **Application-layer firewalls** identify traffic based on application behavior rather than port numbers.
- **Strict rule-set management** reduces misconfigurations that attackers exploit.
- **Egress filtering** monitors outgoing traffic to prevent malware from communicating externally.

Classic rule still wins: *deny by default, allow by necessity.*

---

3. Countermeasures Against Honeypot Evasion

Honeypots are traps. Attackers try to smell the trap.

Skilled attackers attempt to detect honeypots by analyzing system behavior. Defenders counter this by making honeypots more realistic:

- **High-interaction honeypots** simulate real operating systems and services.
- **Randomized system responses** prevent fingerprinting.
- **Delayed and human-like responses** mimic real network latency.
- **Blending honeypots into production environments** avoids standing out.
- **Regular updates and patching** make honeypots appear legitimate.

A bad honeypot screams "trap." A good one whispers "home."

Conclusion

Security is not about one tool—it's about layers working together. IDS detects, firewalls block, honeypots deceive, and administrators decide. Attackers evolve, so defenses must evolve faster.

Module Summary: Evading IDS, Firewalls, and Honeypots

This module explores how attackers attempt to bypass security mechanisms and how defenders counter these evasion techniques using layered, intelligent defenses. The focus is not on breaking systems, but on **strengthening them against stealthy attacks**.

Intrusion Detection Systems can be evaded through packet fragmentation, traffic obfuscation, encryption, and low-and-slow attack techniques. To counter this, organizations deploy traffic normalization, session reassembly, anomaly-based detection, frequent signature updates, and centralized log correlation. Modern IDS solutions increasingly rely on behavioral analysis and machine learning to detect unknown threats.

Firewalls are commonly bypassed using port manipulation, protocol tunneling, spoofing, and crafted packets. Countermeasures include stateful inspection, deep packet inspection, application-aware filtering, strict rule management, anti-spoofing techniques, and the use of next-generation firewalls. The Zero Trust model further strengthens firewall defenses by verifying every access request regardless of network location.

Honeypots are targeted by attackers who attempt to identify and avoid them through fingerprinting and behavioral analysis. To prevent honeypot evasion, defenders use high-interaction and hybrid honeypots, realistic system behavior, dynamic configurations, delayed responses, and seamless integration with production networks. Honeynets extend this concept by simulating entire network environments.

Overall, the module emphasizes **defense in depth**, where IDS, firewalls, honeypots, SIEM systems, and human analysis work together. Security is shown as a continuous process that requires constant monitoring, updating, and adaptation to evolving threats.

THANK YOU