SNORT :

Snort is the foremost Open Source Intrusion Prevention System (IPS)

INFORMATION SECURITY

INTERNAL ASSESSMENT 1

16010122012 RISHIKA BANERJEE
16010122016 SANIKA BHALBAR
16010122025 PARTH CHAMPANERKAR

WHAT IS SNORT?

Snort is an open-source NON PENETRATIVE Intrusion Prevention System (IPS) that monitors network traffic for malicious activity and takes action based on predefined rules. It was created by Martin Roesch in 1998 and is now maintained by Cisco.

What Snort Does:

- Detects suspicious network activity by analyzing packets.
- Logs network traffic for debugging and analysis.
- Prevents intrusions by blocking harmful packets in real time.

How Snort Works:

1. Packet Capture – Snort examines network packets as they travel.
2. Rule Matching – It compares packets against a database of attack signatures.
3. Alert Generation – If a match is found, Snort notifies administrators or takes action.

Modes of Operation:

- Sniffer Mode – Captures and displays network traffic.
- Packet Logger Mode – Stores packets for later analysis.
- Intrusion Prevention Mode – Blocks malicious packets in real time.

Why Snort is Important:

Snort helps organizations detect, analyze, and prevent cyber threats efficiently. Its customizable rules, real-time analysis, and free availability make it a popular choice for network security.

FEATURES OF SNORT

Open-source and lightweight – Snort is free to use and does not require heavy system resources, making it accessible for individuals and organizations.

• Signature-based detection – Snort uses predefined attack signatures to identify threats, allowing for quick and accurate intrusion detection.

• Real-time traffic analysis – Snort inspects packets as they pass through the network, enabling instant threat detection and alerting.

- Open-source and lightweight
- Signature-based detection
- Real-time traffic analysis

Sniffer Mode

Intrusion Prevention Mode

Packet Logger Mode

MODES OF OPERATION

Sniffer Mode – Snort functions as a packet sniffer, similar to tcpdump, allowing users to view network traffic in real time.

- Packet Logger Mode – Stores network traffic for later analysis, which is useful for security audits and forensic investigations.
- Intrusion Prevention Mode – Acts as an IPS (Intrusion Prevention System), detecting and blocking malicious packets to prevent cyberattacks.

HOW SNORT WORKS

Packet capture – Snort analyzes network packets in real time by capturing incoming and outgoing data.

• Rule matching – It compares captured packets against a database of predefined rules to detect suspicious activity.

• Alert generation – If a packet matches a rule, Snort generates an alert or takes action, such as logging the event or blocking traffic.

Packet capture

Rule matching

Alert generation

SNORT RULES OVERVIEW

Structure of Snort rules – Snort rules follow a structured syntax consisting of an action, protocol, source, destination, and rule options. Example:

alert tcp any any -> 192.168.1.100 80 (msg:"Possible Attack Detected"; sid:1000001;)

This rule generates an alert if a TCP packet is sent to IP 192.168.1.100 on port 80.

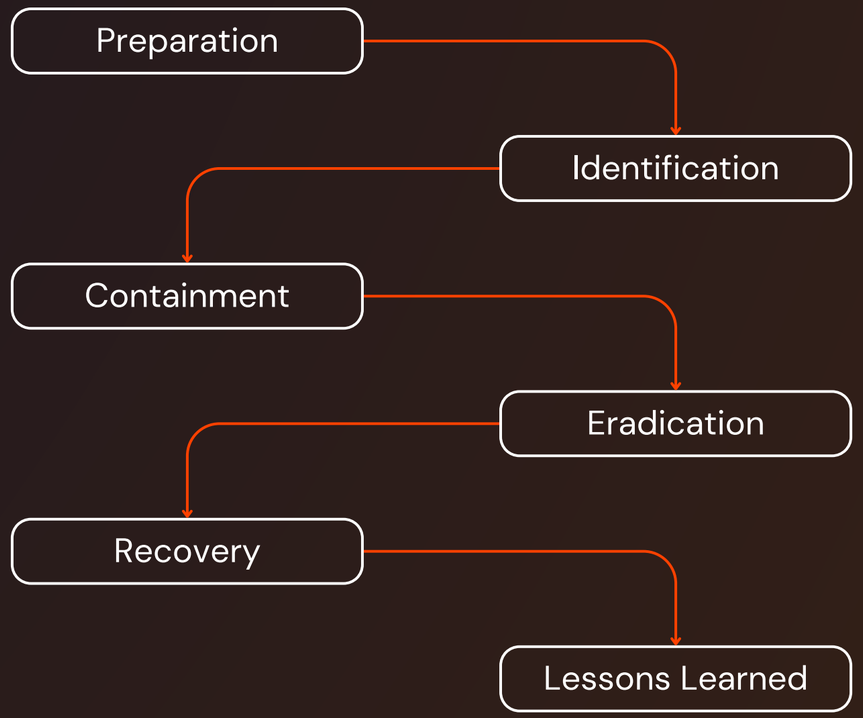
• Example of a basic rule – A simple rule might detect ICMP (ping) traffic, helping to monitor network activity.

SNORT INSTALLATION & SETUP

Steps to install Snort on Linux/Windows:

1. Download Snort from the official website or package manager.
2. Install necessary dependencies like pcap and DAQ for packet capturing.
3. Configure Snort with rule sets and network configurations.
4. Test the installation by running Snort in sniffer mode.

• Required dependencies – Libraries like libpcap and DAQ are needed for Snort to function properly.

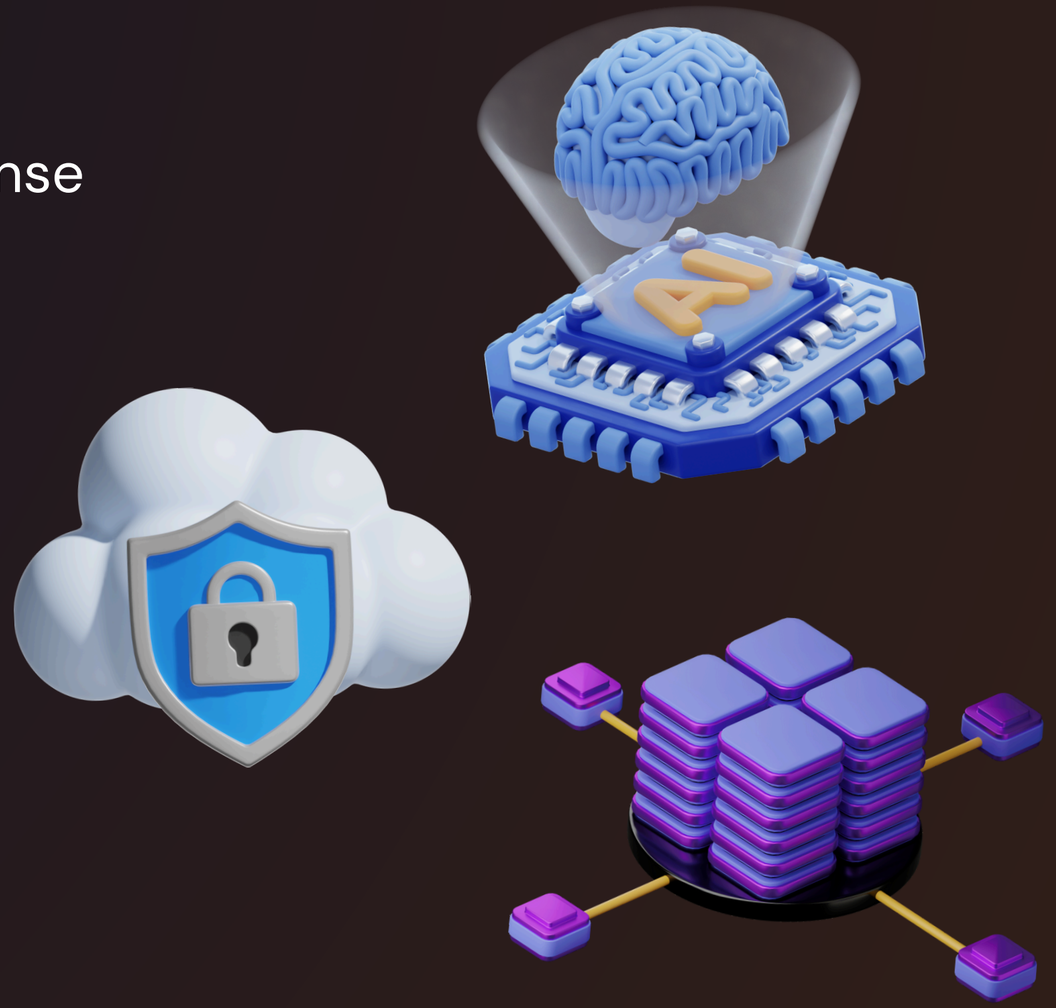


CONFIGURING SNORT

Advanced Technologies in Security & Defense

Setting up Snort rules – Users can create or modify rule files (*.rules) to define attack patterns and security policies.

• Running Snort in different modes – Snort can be executed in sniffer mode (snort -v), logging mode (snort -l ./log), or IDS/IPS mode (snort -c /etc/snort/snort.conf).

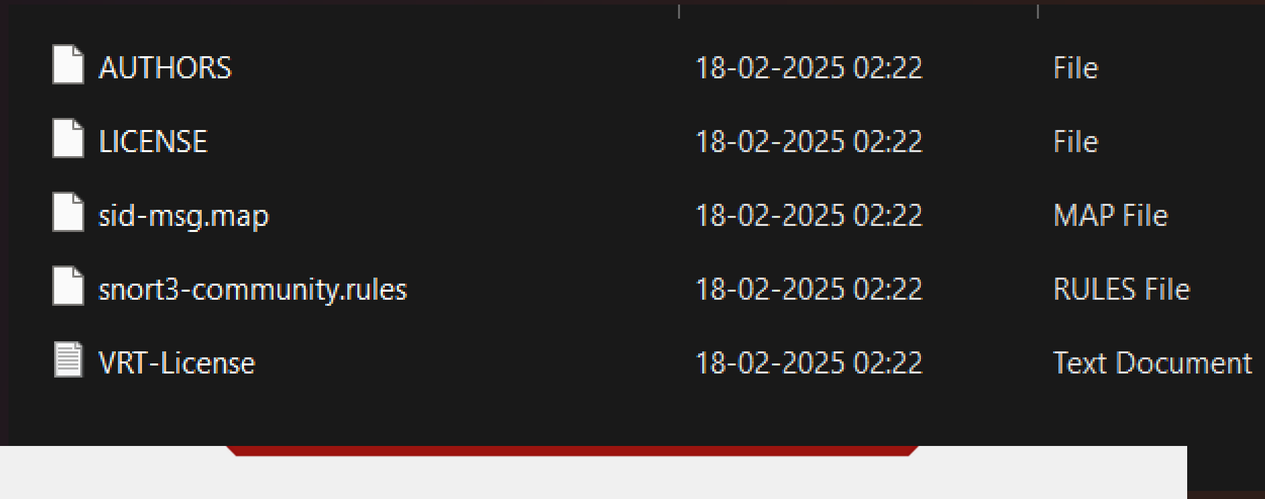


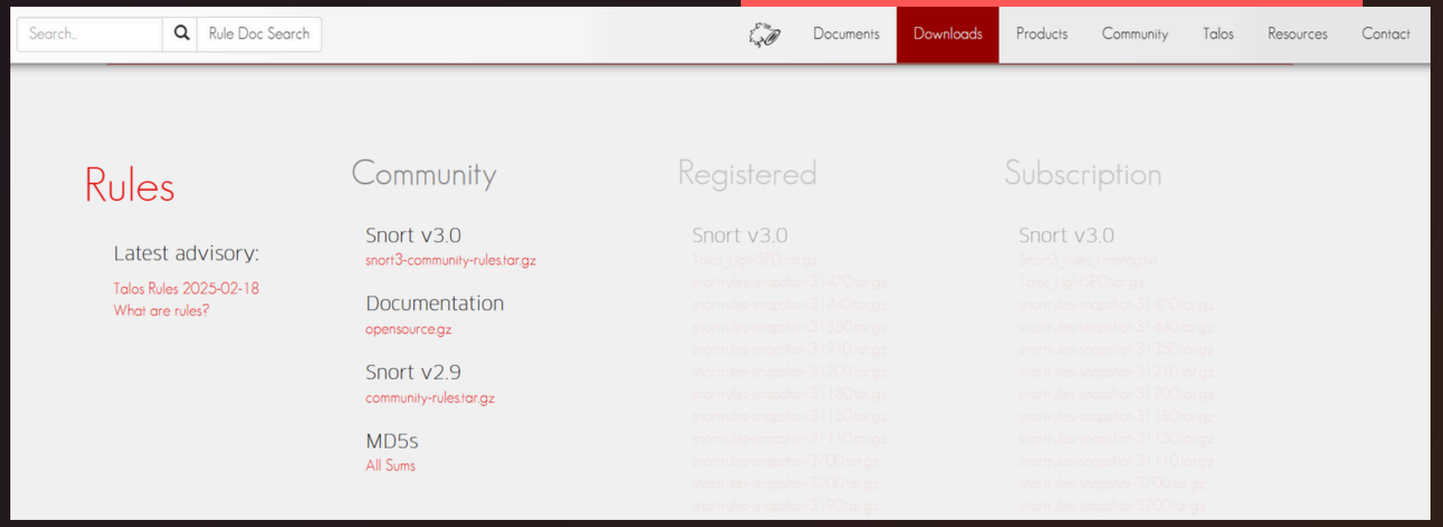
DEMONSTRATION OF SNORT EXECUTION (SCREENSHOTS)

Setting Up Snort: Required Downloads

1. **Snort Download**
The Snort installation package was obtained from the official Snort download page, which provides various versions and setup documentation.

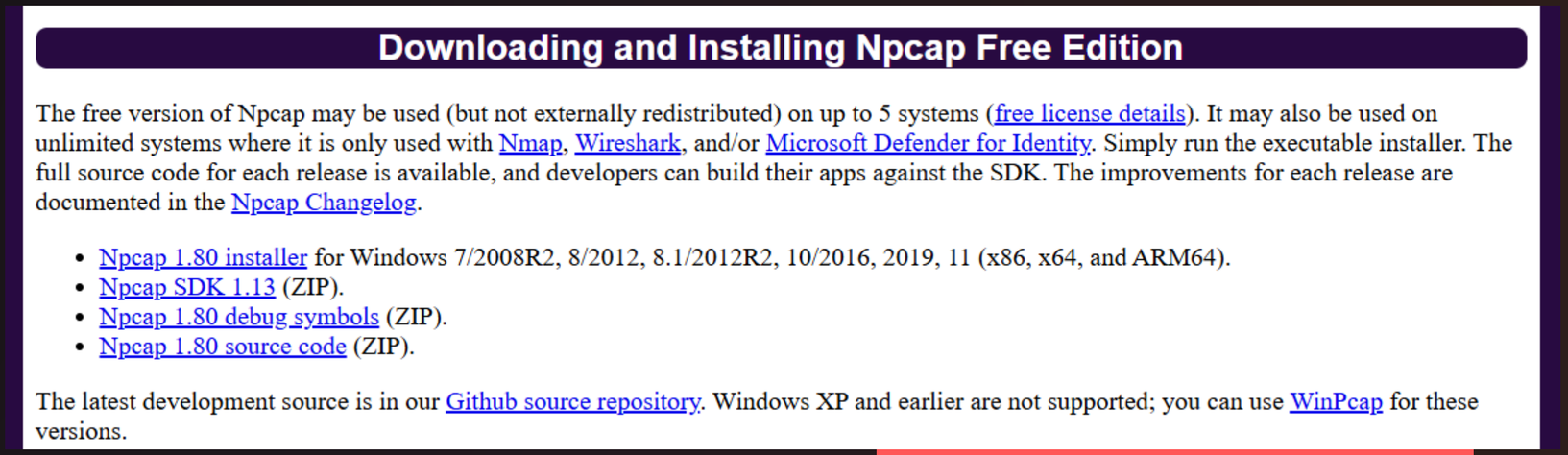
2.**Including Snort Community Rules**
The snort3-community-rules.tar.gz file was downloaded and added to snort.conf to enhance Snort's threat detection capabilities.





DEMONSTRATION OF SNORT EXECUTION (SCREENSHOTS)

Setting Up Snort: Required Downloads

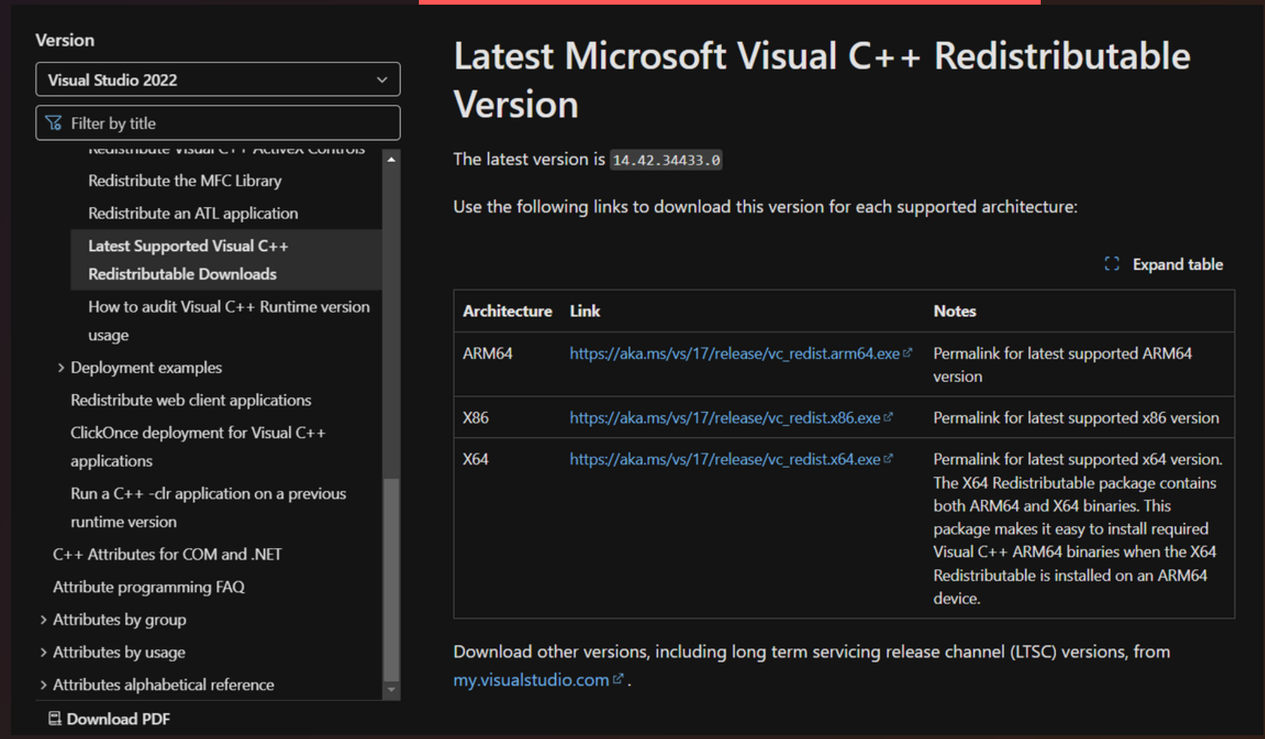


3 . Npcap Download

Npcap, required for Snort to capture network traffic on Windows, was downloaded from its official site.

4 . C++ Redistributable Download

The necessary Microsoft Visual C++ Redistributable was downloaded to ensure Snort runs properly on Windows.



DEMONSTRATION OF SNORT EXECUTION (SCREENSHOTS)

Things in snort.conf:

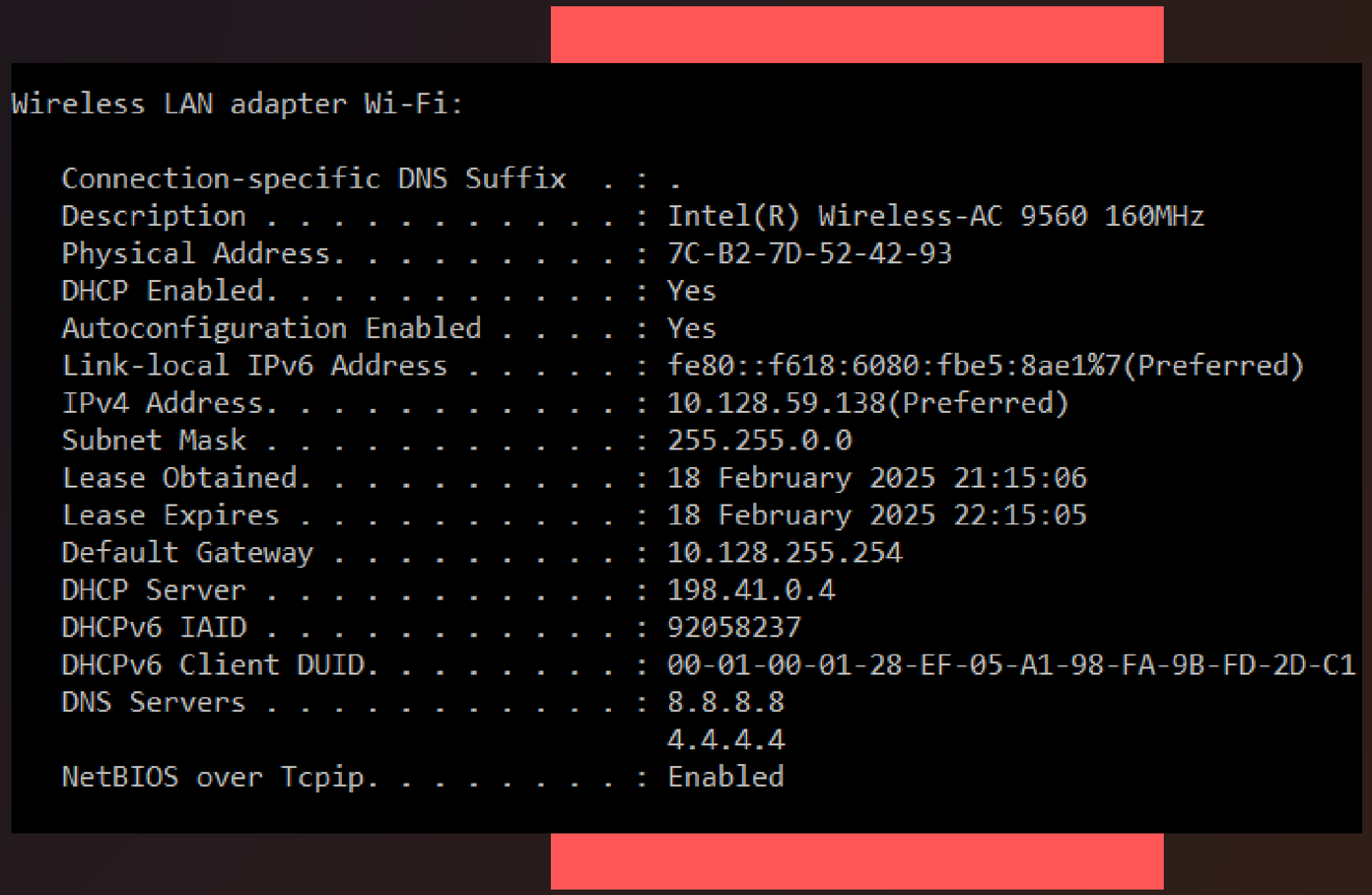
This configuration file contains essential settings for Snort, including network variable definitions, rule paths, and preprocessor settings. The ipvar variable was modified to configure the internal and external network addresses, ensuring accurate traffic monitoring.

```
# Setup the network addresses you are protecting
ipvar HOME_NET 10.128.0.0/16

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any

# List of DNS servers on your network
ipvar DNS_SERVERS $HOME_NET

# List of SMTP servers on your network
ipvar SMTP_SERVERS $HOME_NET

# List of web servers on your network
ipvar HTTP_SERVERS $HOME_NET

# List of sql servers on your network
ipvar SQL_SERVERS $HOME_NET
```

DEMONSTRATION OF SNORT EXECUTION (SCREENSHOTS)

Determining the ipvar Value:

To set the correct ipvar value, the network configuration was checked. The subnet mask 255.255.0.0 corresponds to /16 in CIDR notation. This information was used to accurately configure Snort’s network settings.



DEMONSTRATION OF SNORT EXECUTION (SCREENSHOTS)

Modifying ipvar in snort.conf:

```
# Path to your rules files (this can be a relative path)
# Note for Windows users:  You are advised to make this an absolute path,
# such as:  c:\snort\rules
var RULE_PATH ../rules
var SO_RULE_PATH ../so_rules
var PREPROC_RULE_PATH ../preproc_rules
include $RULE_PATH/community.rules
```

The ipvar variable was updated in snort.conf to define the network range. This configuration allows Snort to correctly analyze and filter traffic based on specified IP addresses. Adjusting ipvar is a crucial step in setting up Snort for a specific network environment.

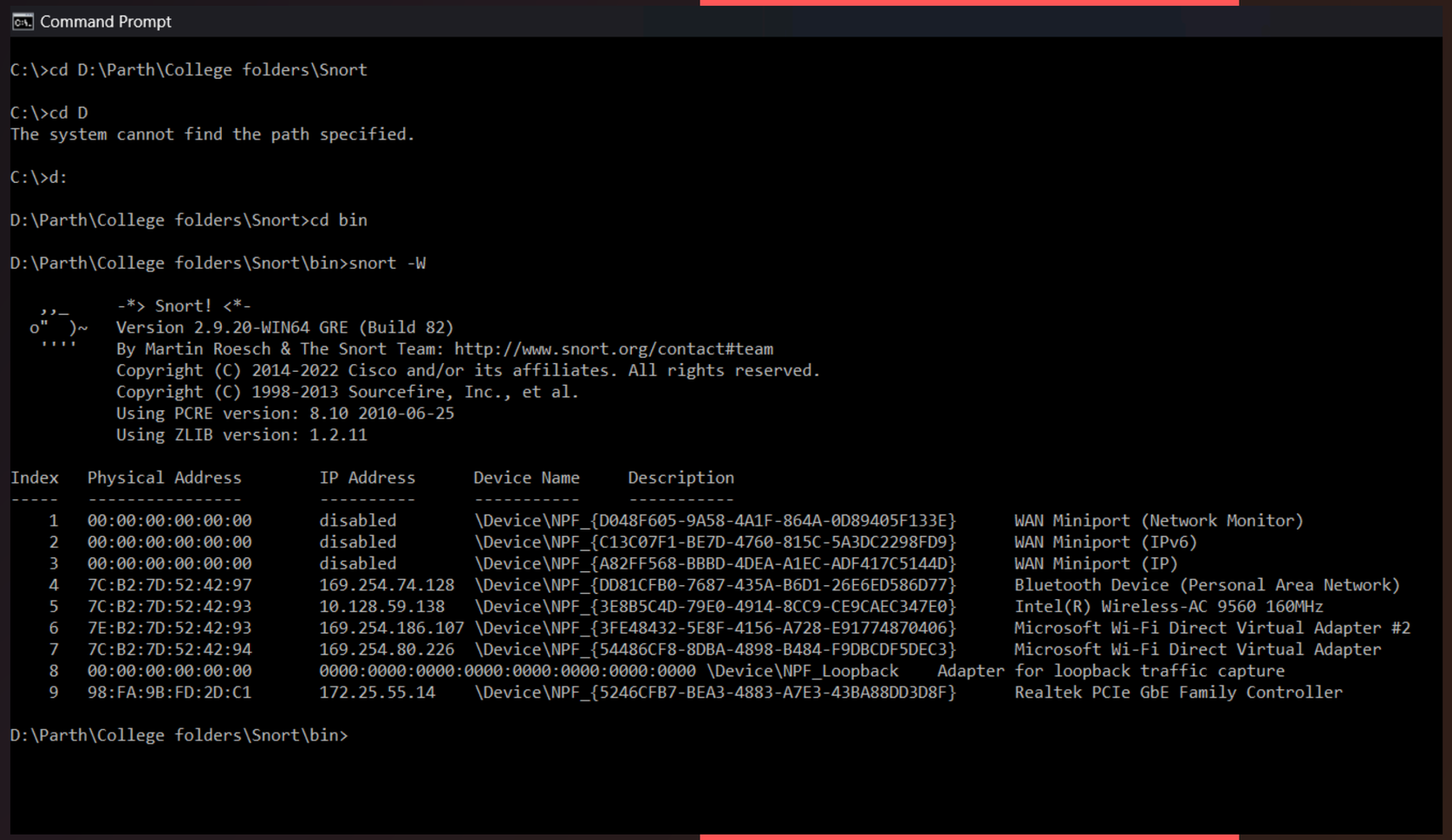
DEMONSTRATION OF SNORT EXECUTION (SCREENSHOTS)

Running Snort in Sniffer Mode:

snort –v

This command captures and displays network packets.

Example of detecting a simple attack: Running Snort with a rule to detect an ICMP request (ping) can generate alerts when someone pings a system.



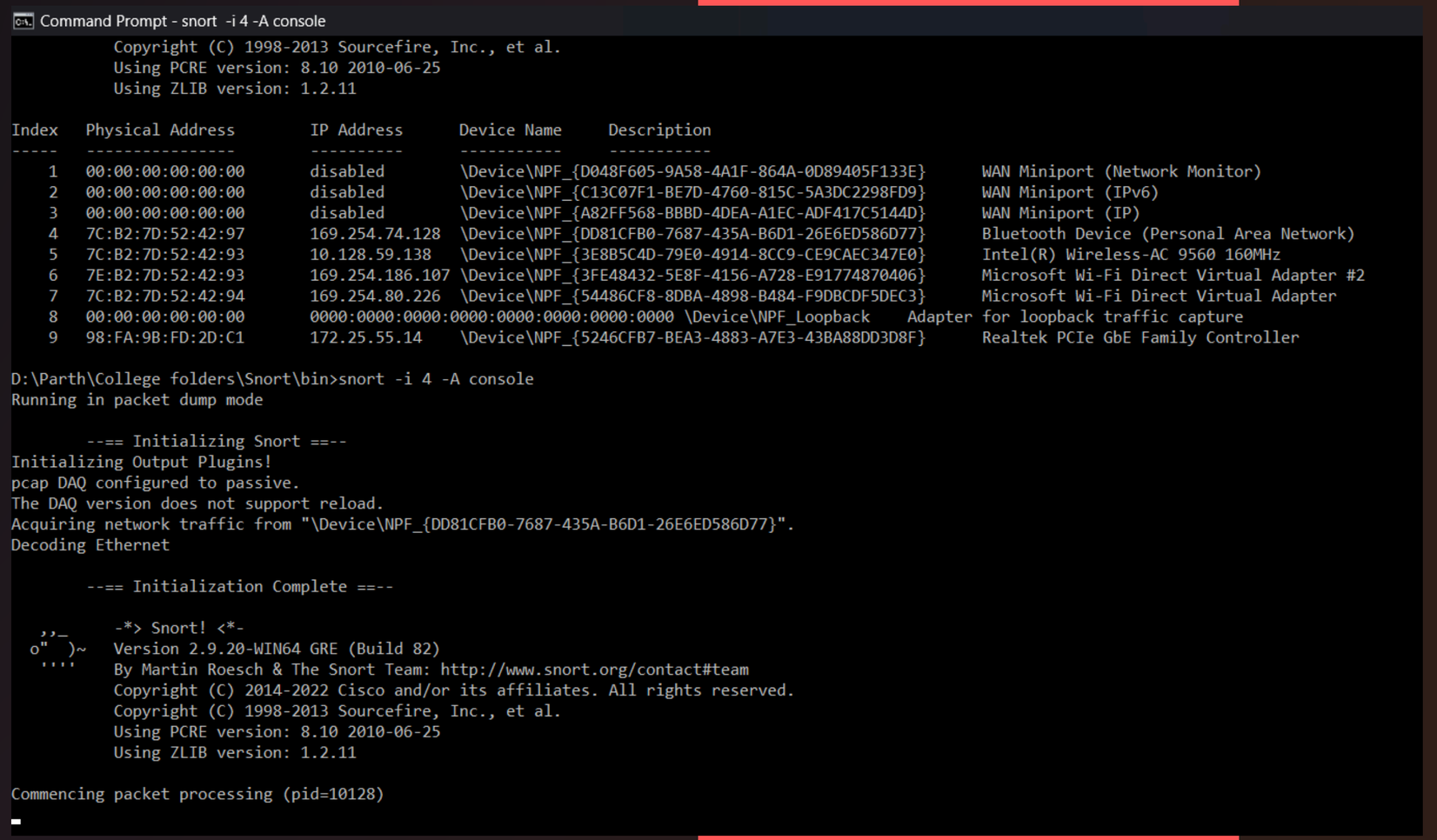
DEMONSTRATION OF SNORT EXECUTION (SCREENSHOTS)

Running Snort in Sniffer Mode:

snort –v

This command captures and displays network packets.

Example of detecting a simple attack: Running Snort with a rule to detect an ICMP request (ping) can generate alerts when someone pings a system.



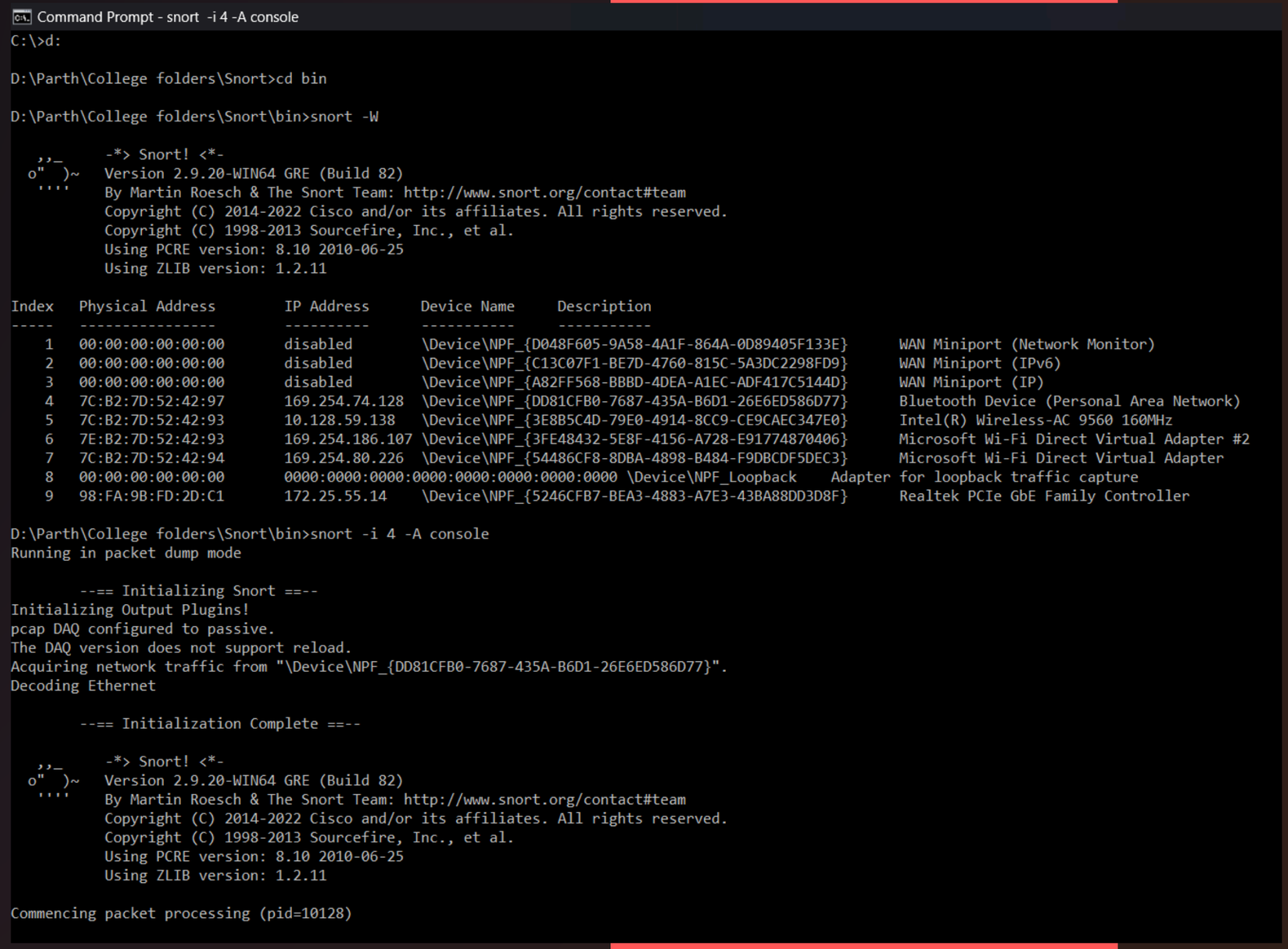
DEMONSTRATION OF SNORT EXECUTION (SCREENSHOTS)

Running Snort in Sniffer Mode:

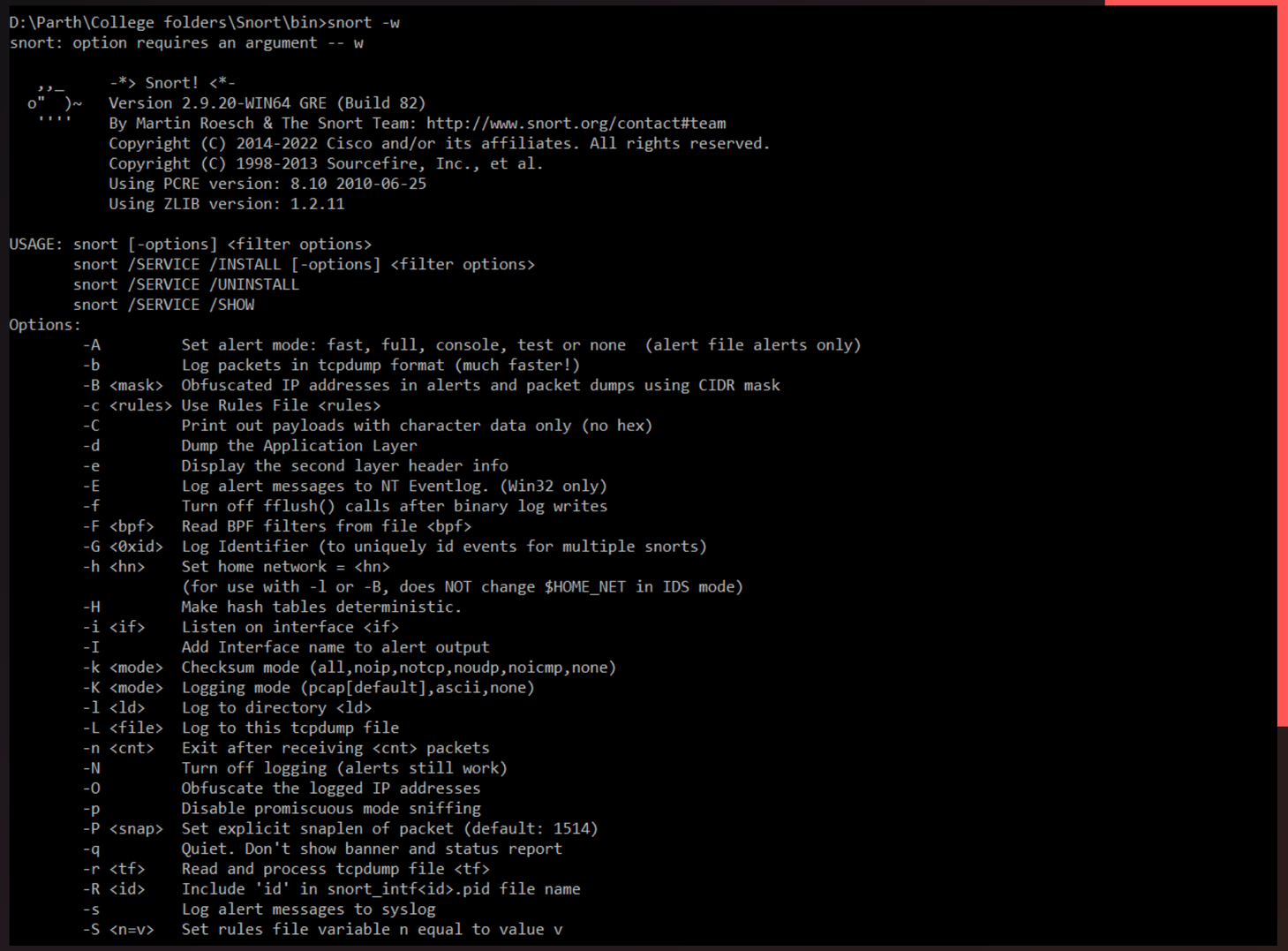
snort –v

This command captures and displays network packets.

Example of detecting a simple attack: Running Snort with a rule to detect an ICMP request (ping) can generate alerts when someone pings a system.



DEMONSTRATION OF SNORT EXECUTION (SCREENSHOTS)



```
D:\Parth\College folders\Snort\bin>snort -w
snort: option requires an argument -- w

   ,,_     -*> 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

USAGE: snort [-options] <filter options>
       snort /SERVICE /INSTALL [-options] <filter options>
       snort /SERVICE /UNINSTALL
       snort /SERVICE /SHOW
Options:
        -A         Set alert mode: fast, full, console, test or none  (alert file alerts only)
        -b         Log packets in tcpdump format (much faster!)
        -B <mask>  Obfuscated IP addresses in alerts and packet dumps using CIDR mask
        -c <rules> Use Rules File <rules>
        -C         Print out payloads with character data only (no hex)
        -d         Dump the Application Layer
        -e         Display the second layer header info
        -E         Log alert messages to NT Eventlog. (Win32 only)
        -f         Turn off fflush() calls after binary log writes
        -F <bpf>   Read BPF filters from file <bpf>
        -G <0xid>  Log Identifier (to uniquely id events for multiple snorts)
        -h <hn>    Set home network = <hn>
                   (for use with -l or -B, does NOT change $HOME_NET in IDS mode)
        -H         Make hash tables deterministic.
        -i <if>    Listen on interface <if>
        -I         Add Interface name to alert output
        -k <mode>  Checksum mode (all,noip,notcp,noudp,noicmp,none)
        -K <mode>  Logging mode (pcap[default],ascii,none)
        -l <ld>    Log to directory <ld>
        -L <file>  Log to this tcpdump file
        -n <cnt>   Exit after receiving <cnt> packets
        -N         Turn off logging (alerts still work)
        -O         Obfuscate the logged IP addresses
        -p         Disable promiscuous mode sniffing
        -P <snap>  Set explicit snaplen of packet (default: 1514)
        -q         Quiet. Don't show banner and status report
        -r <tf>    Read and process tcpdump file <tf>
        -R <id>    Include 'id' in snort_intf<id>.pid file name
        -s         Log alert messages to syslog
        -S <n=v>   Set rules file variable n equal to value v
```

DEMONSTRATION OF SNORT EXECUTION (SCREENSHOTS)

SNORT COMMAND IS RUNNING IN IDS MODE

```
D:\Parth\College folders\Snort\bin>snort -i 5 -c "D:\Parth\College folders\Snort\etc\snort.conf" -l "D:\Parth\College folders\Snort\log" -A console
Running in IDS mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "D:\Parth\College folders\Snort\etc\snort.conf"
PortVar 'HTTP_PORTS' defined :  [ 80:81 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8
088 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 'SHELLCODE_PORTS' defined :  [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined :  [ 1024:65535 ]
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 802
8 8080 8085 8088 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 'GTP_PORTS' defined :  [ 2123 2152 3386 ]
Detection:
   Search-Method = AC-Full-Q
    Split Any/Any group = enabled
    Search-Method-Optimizations = enabled
    Maximum pattern length = 20
```

DEMONSTRATION OF SNORT EXECUTION (SCREENSHOTS)

SNORT IN PACKET LOGGING MODE WITH DETAILED PACKET ANALYSIS.

```
D:\Parth\College folders\Snort\bin>snort -i 5 -l "D:\Parth\College folders\Snort\log" -dev
Running in packet logging mode

       --== Initializing Snort ==--
Initializing Output Plugins!
Log directory = D:\Parth\College folders\Snort\log
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{3E8B5C4D-79E0-4914-8CC9-CE9CAEC347E0}".
Decoding Ethernet

       --== 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
```

```
Run time for packet processing was 5.16000 seconds
Snort processed 1732 packets.
Snort ran for 0 days 0 hours 0 minutes 5 seconds
   Pkts/sec:          346
===============================================================
Packet I/O Totals:
   Received:         1754
   Analyzed:         1732 ( 98.746%)
    Dropped:            0 (  0.000%)
   Filtered:            0 (  0.000%)
Outstanding:           22 (  1.254%)
   Injected:            0
===============================================================
Breakdown by protocol (includes rebuilt packets):
        Eth:         1732 (100.000%)
       VLAN:            0 (  0.000%)
        IP4:         1728 ( 99.769%)
       Frag:            0 (  0.000%)
       ICMP:            0 (  0.000%)
        UDP:         1713 ( 98.903%)
        TCP:           15 (  0.866%)
        IP6:            2 (  0.115%)
    IP6 Ext:            2 (  0.115%)
   IP6 Opts:            0 (  0.000%)
      Frag6:            0 (  0.000%)
      ICMP6:            0 (  0.000%)
       UDP6:            2 (  0.115%)
       TCP6:            0 (  0.000%)
     Teredo:            0 (  0.000%)
    ICMP-IP:            0 (  0.000%)
      EAPOL:            0 (  0.000%)
    IP4/IP4:            0 (  0.000%)
    IP4/IP6:            0 (  0.000%)
    IP6/IP4:            0 (  0.000%)
    IP6/IP6:            0 (  0.000%)
        GRE:            0 (  0.000%)
    GRE Eth:            0 (  0.000%)
   GRE VLAN:            0 (  0.000%)
    GRE IP4:            0 (  0.000%)
    GRE IP6:            0 (  0.000%)
GRE IP6 Ext:            0 (  0.000%)
   GRE PPTP:            0 (  0.000%)
    GRE ARP:            0 (  0.000%)
    GRE IPX:            0 (  0.000%)
   GRE Loop:            0 (  0.000%)
```

```
   GRE VLAN:            0 (  0.000%)
    GRE IP4:            0 (  0.000%)
    GRE IP6:            0 (  0.000%)
GRE IP6 Ext:            0 (  0.000%)
   GRE PPTP:            0 (  0.000%)
    GRE ARP:            0 (  0.000%)
    GRE IPX:            0 (  0.000%)
   GRE Loop:            0 (  0.000%)
       MPLS:            0 (  0.000%)
        ARP:            0 (  0.000%)
        IPX:            0 (  0.000%)
   Eth Loop:            0 (  0.000%)
   Eth Disc:            0 (  0.000%)
   IP4 Disc:            0 (  0.000%)
   IP6 Disc:            0 (  0.000%)
   TCP Disc:            0 (  0.000%)
   UDP Disc:            0 (  0.000%)
  ICMP Disc:            0 (  0.000%)
All Discard:            0 (  0.000%)
      Other:            2 (  0.115%)
Bad Chk Sum:         1478 ( 85.335%)
    Bad TTL:            0 (  0.000%)
     S5 G 1:            0 (  0.000%)
     S5 G 2:            0 (  0.000%)
      Total:         1732
===============================================================

Memory Statistics for File at:Wed Feb 19 11:55:22 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:              0 bytes
              No of allocs:              0
               No of frees:              0
===============================================================
Snort exiting
```

CREATING CUSTOM SNORT RULES

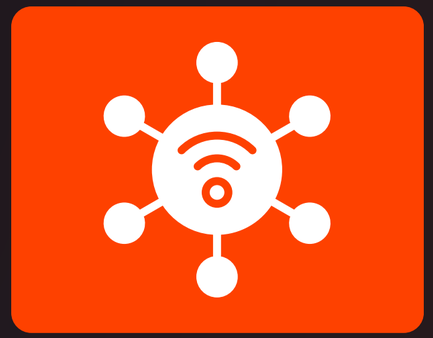
Writing a basic rule. Example:

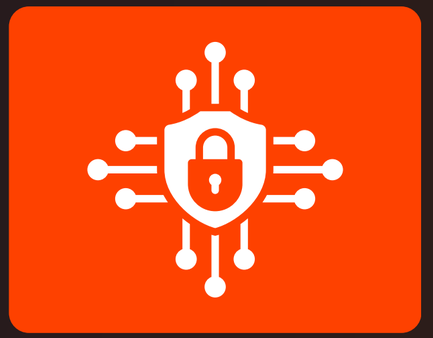
alert icmp any any -> any any (msg:"ICMP Ping Detected"; sid:1000002;)

This rule generates an alert when an ICMP ping is detected.

Testing and applying rules: After writing a rule, Snort must be restarted to apply changes (snort -c /etc/snort/snort.conf).

LOGGING AND ALERTS IN SNORT



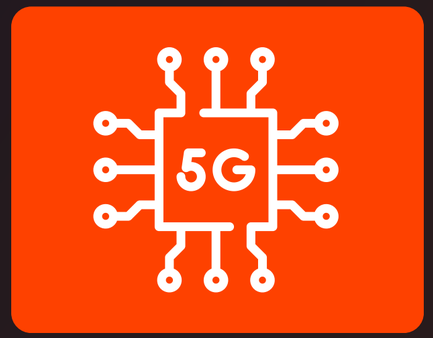


Evolving Threats and Emerging Technologies

How Snort logs detected threats: Logs are stored in directories like /var/log/snort/ and include details about flagged packets.
Examples of alert messages: An example log entry might look like:

[**] [1:1000001:0] Possible Attack Detected [**] Â [Priority: 1] Â

This indicates a high-priority alert.



NON-PENETRATIVE TRAITS OF SNORT

Defensive, not offensive: Snort does not launch attacks; it only monitors and blocks threats.

Rule-based detection: Unlike penetration testing tools that actively probe for weaknesses, Snort relies on predefined rules to identify threats.

Passive mode availability: Snort can operate in a passive mode where it only detects threats without blocking traffic.

No direct exploitation: Unlike penetration testing tools (e.g., Metasploit), Snort does not attempt to exploit security vulnerabilities.

ADVANTAGES & LIMITATIONS

Advantages:
Open-source and customizable: Users can create custom rules to fit specific security needs.
Real-time intrusion prevention: Detects and blocks malicious traffic immediately.
Works with other security tools: Can be integrated with firewalls and SIEM systems for enhanced protection.

Limitations:
Can generate false positives: May mistakenly flag legitimate traffic as malicious.
Needs frequent rule updates: Must be updated regularly to recognize new threats.
Performance impact on high-traffic networks: Can slow down traffic if not optimized properly.

IMPORTANCE OF SNORT IN CYBERSECURITY

Critical for threat detection: Helps identify and mitigate cyber threats in real time.

Enhances network security: Provides an extra layer of protection against malicious traffic.

Supports compliance: Helps organizations meet security standards such as PCI DSS, HIPAA, and GDPR.

Used in SOCs (Security Operations Centers): Security analysts use Snort to monitor networks and respond to attacks.

FUTURE OF SNORT AND INTRUSION PREVENTION

AI and machine learning integration: Future versions of Snort may use AI to automatically detect and respond to unknown threats.

Cloud-based deployment: Snort is increasingly being deployed in cloud environments for scalable security solutions.

Advanced automation: Automating rule updates and threat response will improve efficiency and accuracy.

Growing importance in securing IoT and 5G networks: As IoT and 5G expand, Snort will play a crucial role in protecting these infrastructures.