Extra Credit Project Report

Intrusion Detection System using Snort

By,

Prachi Rishikesh Manwar

Team Mate Name:

Swathi Priya Reddy Kaduru

Intrusion Detection System:

An intrusion detection system(IDS) is a device or a system that monitors network traffic for suspicious activity, malicious activity, policy violations, or issues alert. Any malicious venture or violation is typically reported or collected centrally using a security information and event management system.

While monitoring networks for potentially harmful behavior, intrusion detection systems are also prone to raising false alarms. Consequently, enterprises must adjust their IDS products after initial installation. It entails correctly configuring intrusion detection systems to distinguish between legitimate network traffic and malicious activities. Network packets entering the system are also monitored by intrusion prevention systems to look for any malicious activity and immediately send out alerts.

Need of IDS:

A high level of security is necessary for today's networked corporate environments to provide reliable and secure information sharing between multiple entities. After conventional technologies fail, an intrusion detection system serves as a flexible safety net for system security. The sophistication of cyberattacks will only increase, hence defense technology change must counter them.

IDS are classified into 5 types:

1. Network Intrusion Detection System (NIDS)

2. Host Intrusion Detection System (HIDS)

3. Protocol-based Intrusion Detection System (PIDS)

4. Application Protocol-based Intrusion Detection System (APIDS)

5. Hybrid Intrusion Detection System

From these 5 types we worked on the first type, ie., Network Intrusion Detection System.

Snort:

Snort is a well-known IDS/IPS system that performs traffic/protocol analysis, and content matching, and may be used to identify and stop different attacks based on predefined rules. It is free and open-source.

Numerous users and contributors to Snort actively participate in its development and create rules to keep it up to speed with the most recent attacks.

Snort has 3 main operational modes:

1. Packet Sniffing - Collects and displays network traffic as Wireshark does
2. Packet Logging - Collects and logs network traffic into a file
3. Network intrusion Detection - Analyzes packets and matches traffic against signatures

Snort uses pattern matching to find malicious communications or assaults. When activated, Snort collects packets, breaks them down, examines them, and then decides what should be done with the packet by established rules. Similar to standard firewall rules, Snort rules compare network activity to predefined patterns or signatures and then decide whether to issue an alert or discard the traffic as a result (in the case of IPS). Starting, Snort has several rule sets developed by the community that is quite helpful.

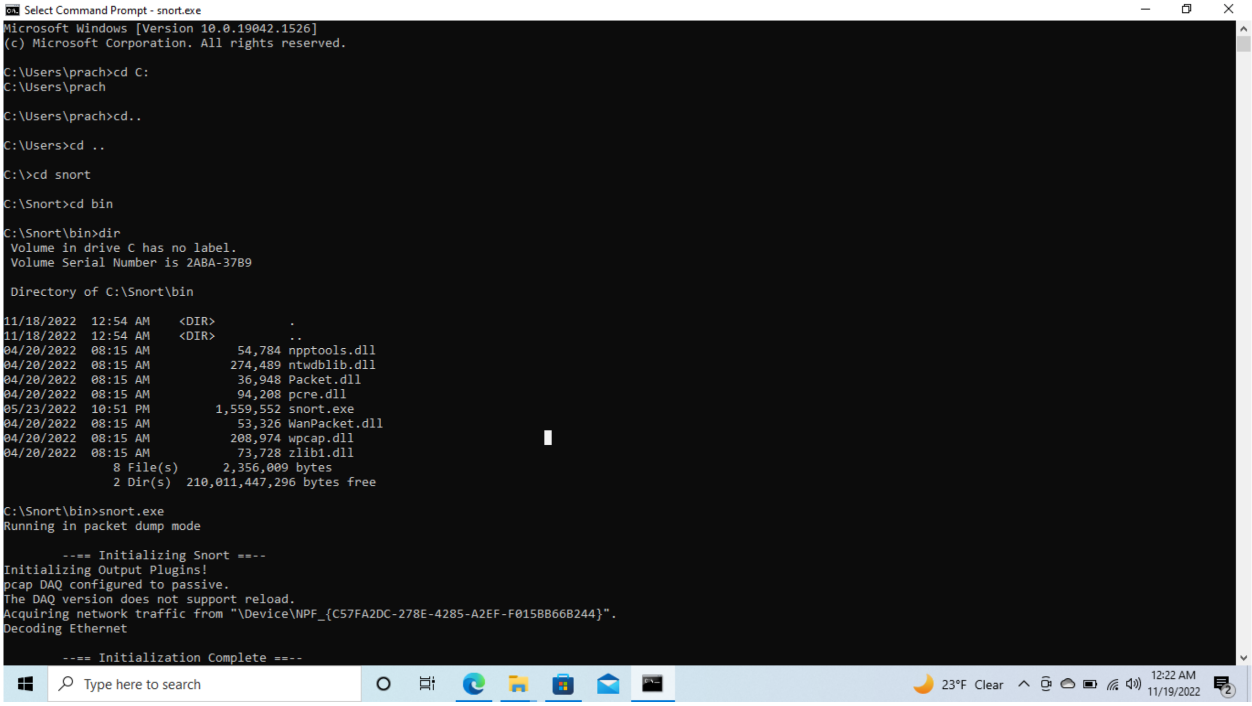
Snort Rules:

1. Community rules - Free rule sets created by the Snort community. Registered rules - Free rule sets created by Talos. To use them, you must register for an account.
2. Subscription-only rules - These rule sets require an active paid subscription to be accessed and used.

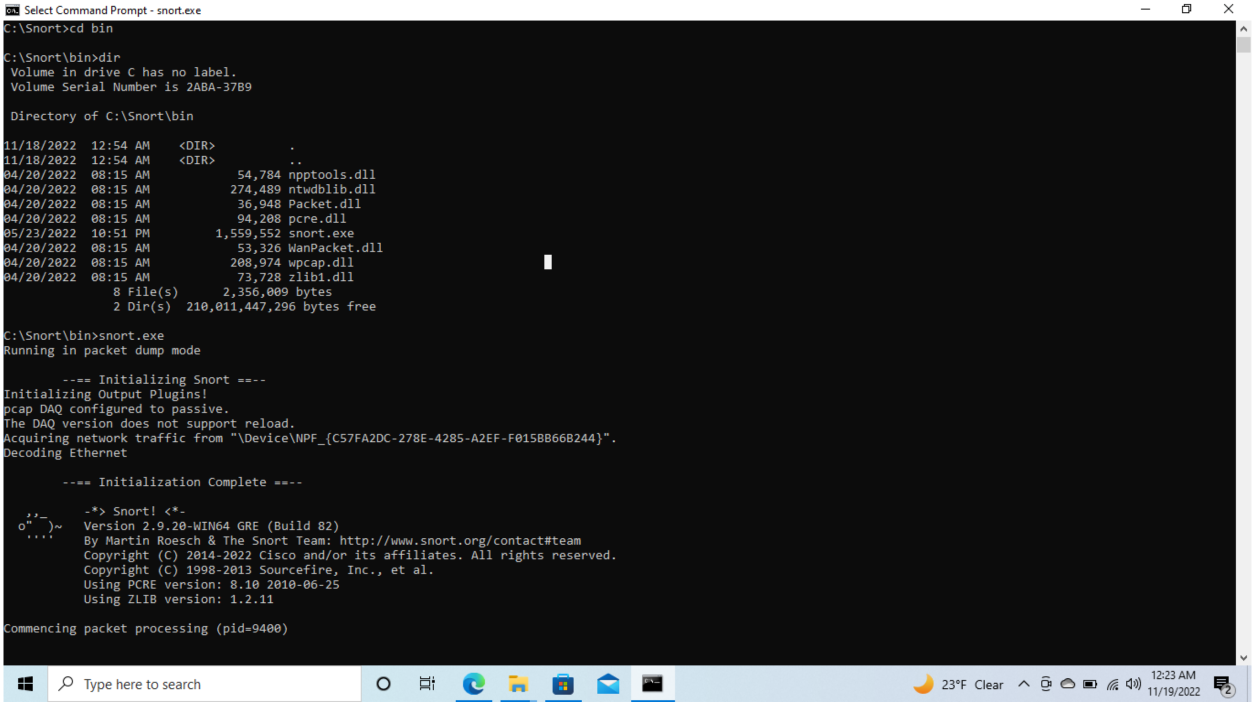
An intrusion detection system with a snort:

Snort offers a Windows setup and signatures that can be used with any operating system. Snort should be a dedicated computer in your network. This computer's logs should be reviewed often to see malicious activities on your network.
Firstly we need to install snort on our system. We used the Windows system to execute this project. Later we need to install WinPcap, it is important to have WinPcap installed.
Then we need to use the command prompt to start snort.

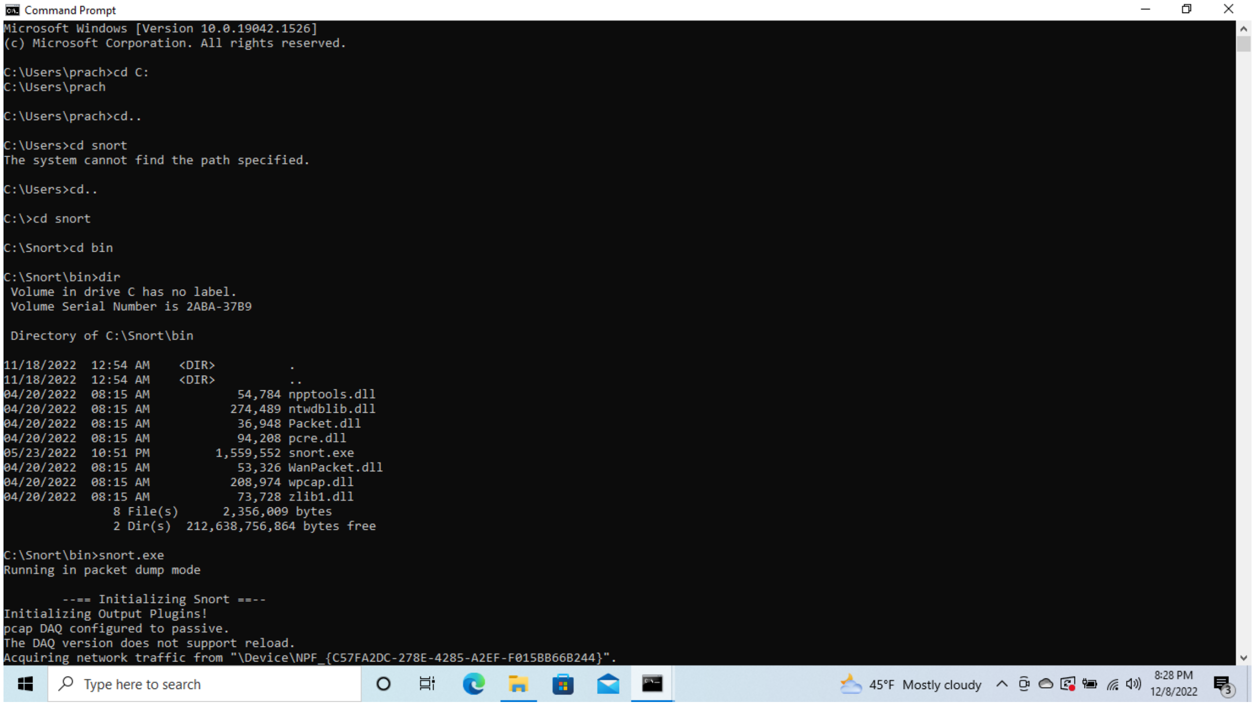
The implemented intrusion Detection system is as follows:



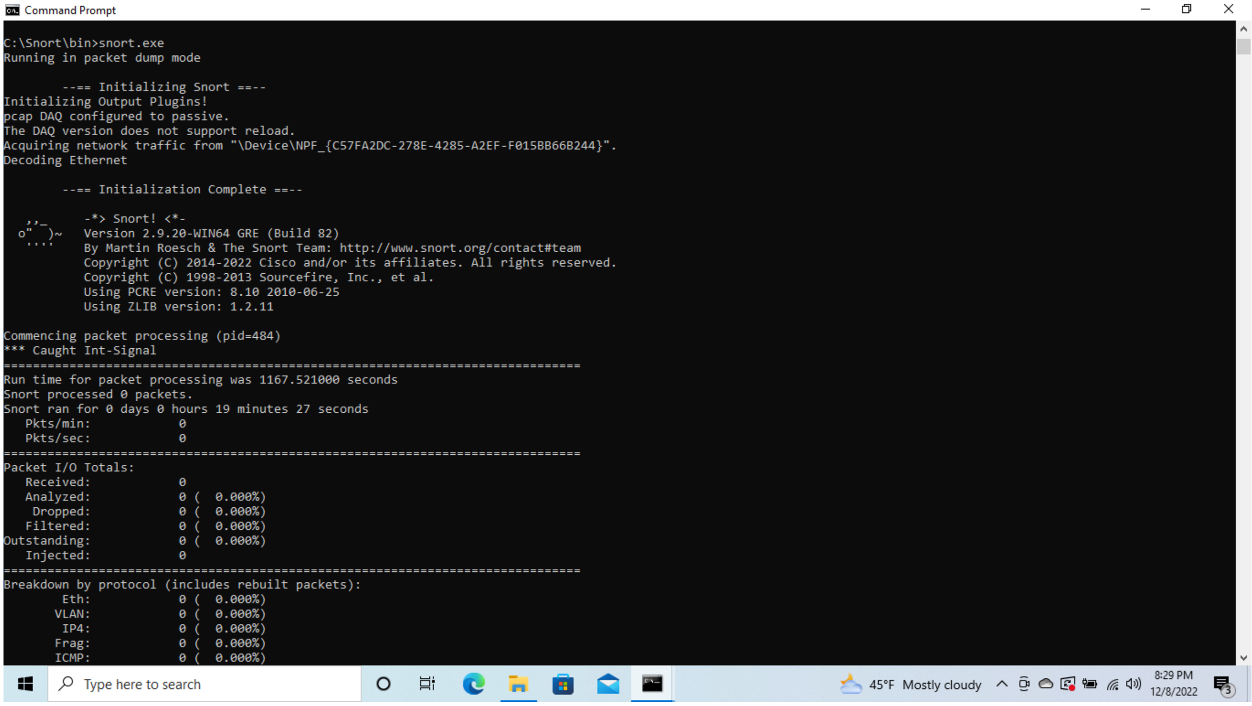
```
Microsoft Windows [Version 10.0.19042.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Users\prach>cd C:
C:\Users\prach

C:\Users\prach>cd..

C:\Users>cd ..

C:\>cd snort

C:\Snort>cd bin

C:\Snort\bin>dir
 Volume in drive C has no label.
 Volume Serial Number is 2ABA-37B9

 Directory of C:\Snort\bin

11/18/2022  12:54 AM    <DIR>          .
11/18/2022  12:54 AM    <DIR>          ..
04/20/2022  08:15 AM            54,784 npptools.dll
04/20/2022  08:15 AM           274,489 ntwdblib.dll
04/20/2022  08:15 AM            36,948 Packet.dll
04/20/2022  08:15 AM            94,208 pcre.dll
05/23/2022  10:51 PM         1,559,552 snort.exe
04/20/2022  08:15 AM            53,326 WanPacket.dll
04/20/2022  08:15 AM           208,974 wpcap.dll
04/20/2022  08:15 AM            73,728 zlib1.dll
               8 File(s)      2,356,009 bytes
               2 Dir(s)  210,011,447,296 bytes free

C:\Snort\bin>snort.exe
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{C57FA2DC-278E-4285-A2EF-F015BB66B244}".
Decoding Ethernet

        --== Initialization Complete ==--
```



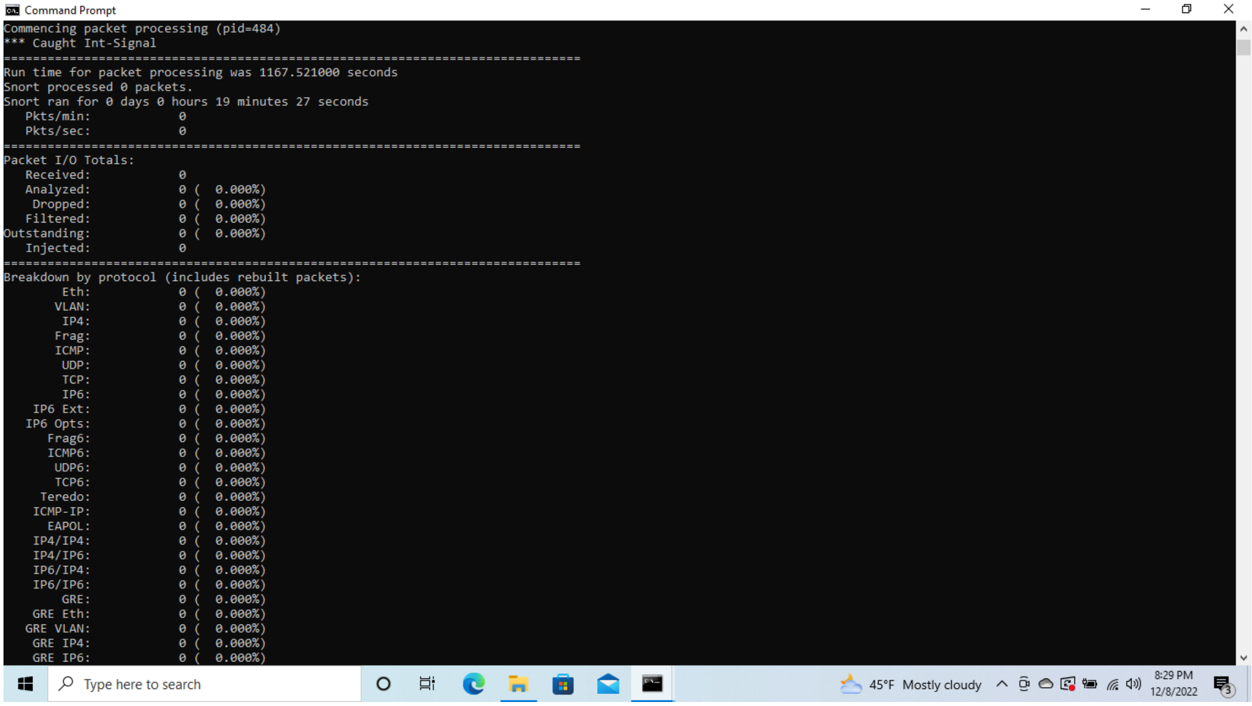
```
C:\Snort>cd bin

C:\Snort\bin>dir
 Volume in drive C has no label.
 Volume Serial Number is 2ABA-37B9

 Directory of C:\Snort\bin

11/18/2022  12:54 AM    <DIR>          .
11/18/2022  12:54 AM    <DIR>          ..
04/20/2022  08:15 AM            54,784 npptools.dll
04/20/2022  08:15 AM           274,489 ntwdblib.dll
04/20/2022  08:15 AM            36,948 Packet.dll
04/20/2022  08:15 AM            94,208 pcre.dll
05/23/2022  10:51 PM         1,559,552 snort.exe
04/20/2022  08:15 AM            53,326 WanPacket.dll
04/20/2022  08:15 AM           208,974 wpcap.dll
04/20/2022  08:15 AM            73,728 zlib1.dll
               8 File(s)      2,356,009 bytes
               2 Dir(s)  210,011,447,296 bytes free

C:\Snort\bin>snort.exe
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{C57FA2DC-278E-4285-A2EF-F015BB66B244}".
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

Commencing packet processing (pid=9400)
```



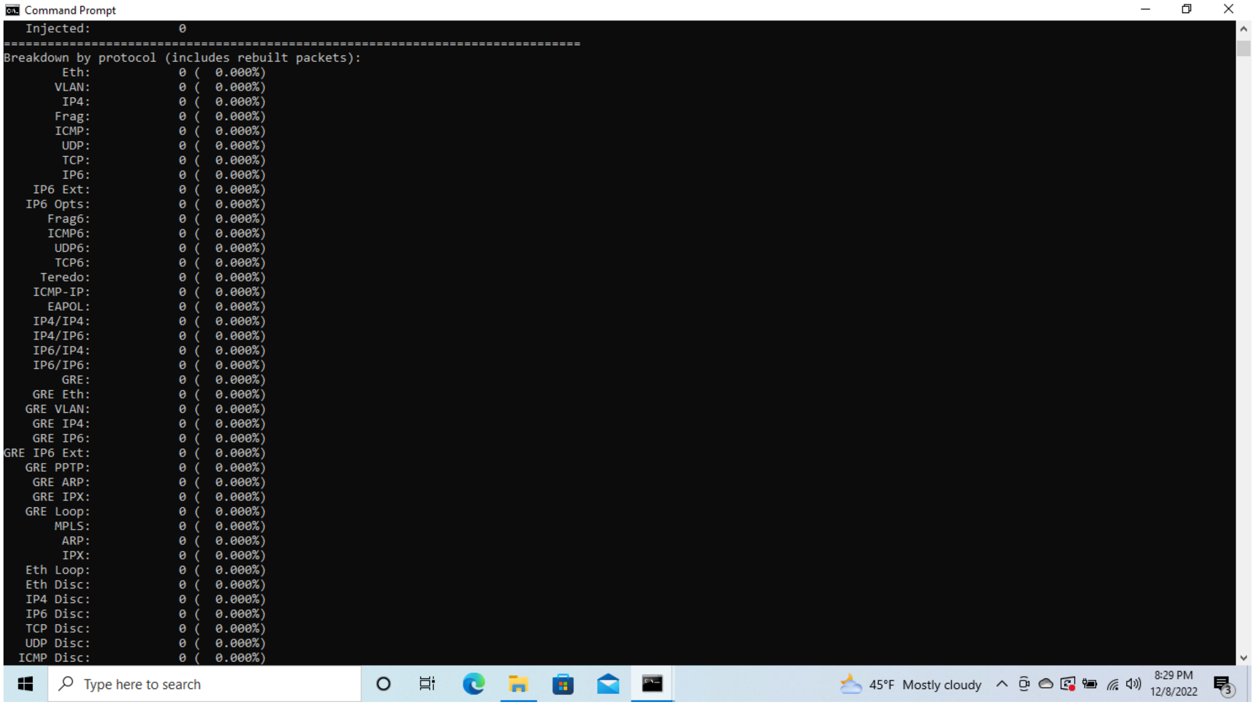
```
Microsoft Windows [Version 10.0.19042.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Users\prach>cd C:
C:\Users\prach

C:\Users\prach>cd..

C:\Users>cd snort
The system cannot find the path specified.

C:\Users>cd..

C:\>cd snort

C:\Snort>cd bin

C:\Snort\bin>dir
 Volume in drive C has no label.
 Volume Serial Number is 2ABA-37B9

 Directory of C:\Snort\bin

11/18/2022  12:54 AM    <DIR>          .
11/18/2022  12:54 AM    <DIR>          ..
04/20/2022  08:15 AM            54,784 npptools.dll
04/20/2022  08:15 AM           274,489 ntwdblib.dll
04/20/2022  08:15 AM            36,948 Packet.dll
04/20/2022  08:15 AM            94,208 pcre.dll
05/23/2022  10:51 PM         1,559,552 snort.exe
04/20/2022  08:15 AM            53,326 WanPacket.dll
04/20/2022  08:15 AM           208,974 wpcap.dll
04/20/2022  08:15 AM            73,728 zlib1.dll
               8 File(s)      2,356,009 bytes
               2 Dir(s)  212,638,756,864 bytes free

C:\Snort\bin>snort.exe
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{C57FA2DC-278E-4285-A2EF-F015BB66B244}".
```



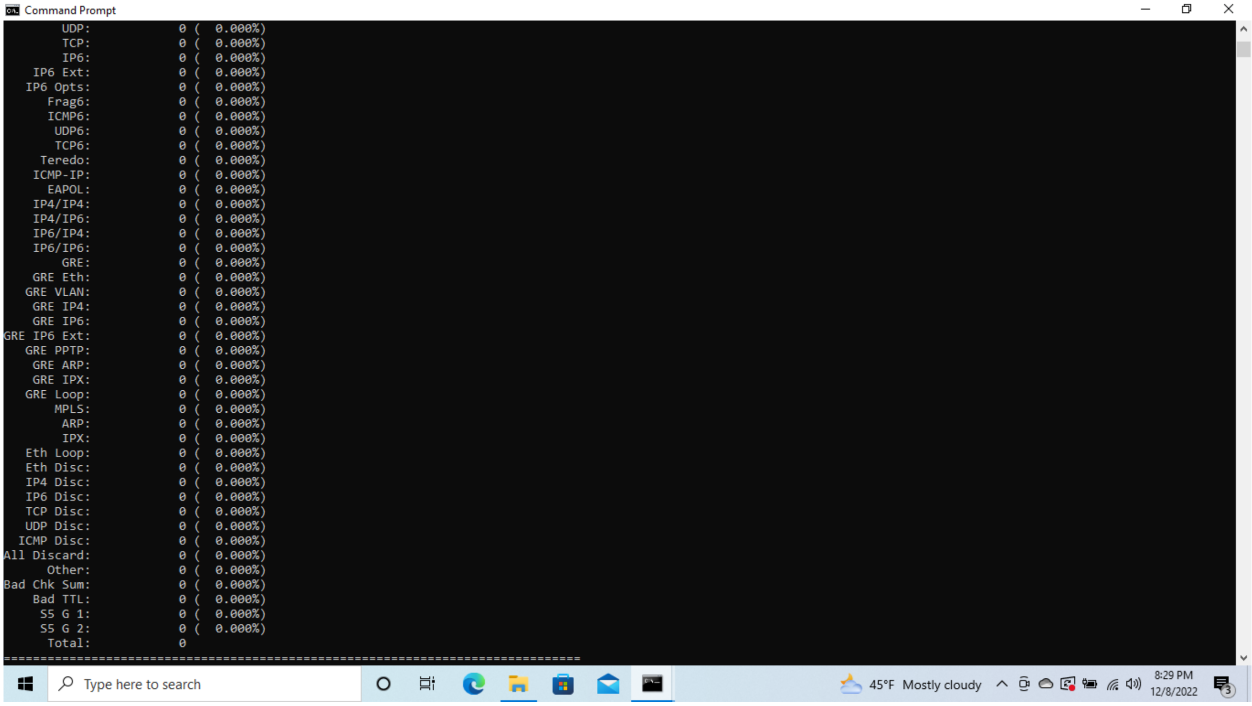
```
C:\Snort\bin>snort.exe
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{C57FA2DC-278E-4285-A2EF-F015BB66B244}".
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

Commencing packet processing (pid=484)
*** Caught Int-Signal
===============================================================================
Run time for packet processing was 1167.521000 seconds
Snort processed 0 packets.
Snort ran for 0 days 0 hours 19 minutes 27 seconds
   Pkts/min:            0
   Pkts/sec:            0
===============================================================================
Packet I/O Totals:
   Received:            0
   Analyzed:            0 (  0.000%)
    Dropped:            0 (  0.000%)
   Filtered:            0 (  0.000%)
Outstanding:            0 (  0.000%)
   Injected:            0
===============================================================================
Breakdown by protocol (includes rebuilt packets):
        Eth:            0 (  0.000%)
       VLAN:            0 (  0.000%)
        IP4:            0 (  0.000%)
       Frag:            0 (  0.000%)
       ICMP:            0 (  0.000%)
```



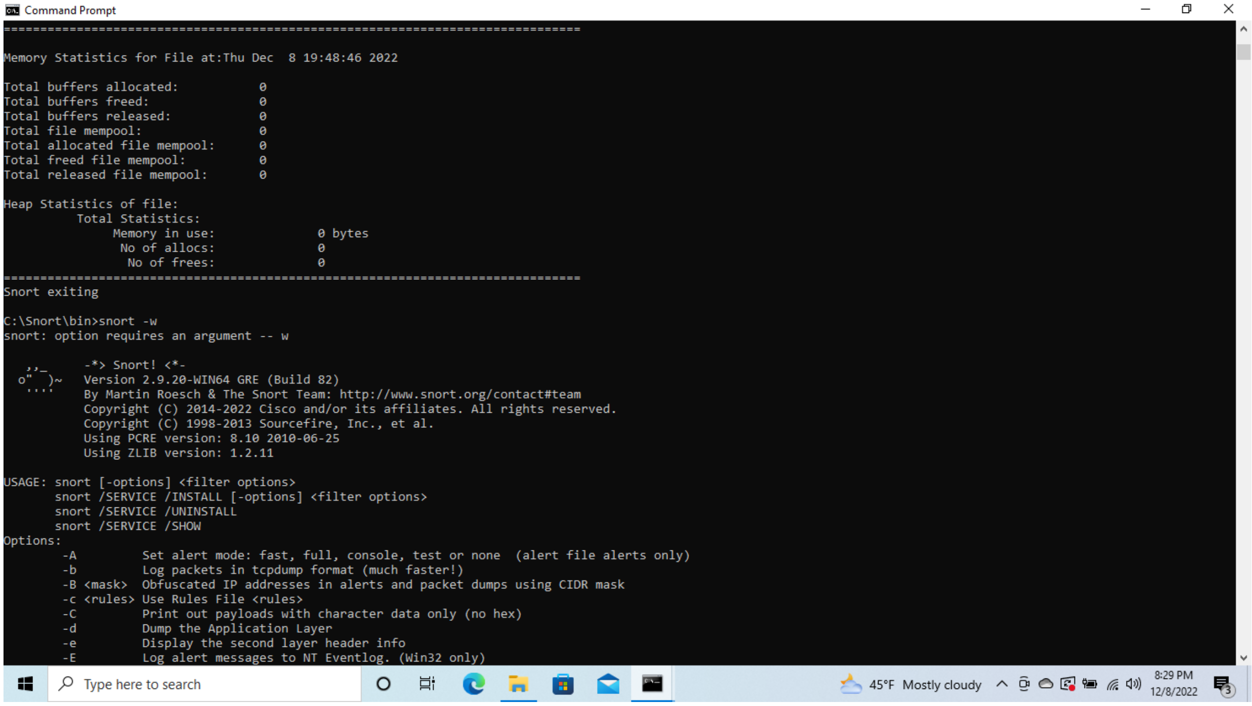
```
Commencing packet processing (pid=484)
*** Caught Int-Signal
===============================================================================
Run time for packet processing was 1167.521000 seconds
Snort processed 0 packets.
Snort ran for 0 days 0 hours 19 minutes 27 seconds
   Pkts/min:            0
   Pkts/sec:            0
===============================================================================
Packet I/O Totals:
   Received:            0
   Analyzed:            0 (  0.000%)
    Dropped:            0 (  0.000%)
   Filtered:            0 (  0.000%)
Outstanding:            0 (  0.000%)
   Injected:            0
===============================================================================
Breakdown by protocol (includes rebuilt packets):
        Eth:            0 (  0.000%)
       VLAN:            0 (  0.000%)
        IP4:            0 (  0.000%)
       Frag:            0 (  0.000%)
       ICMP:            0 (  0.000%)
        UDP:            0 (  0.000%)
        TCP:            0 (  0.000%)
        IP6:            0 (  0.000%)
    IP6 Ext:            0 (  0.000%)
   IP6 Opts:            0 (  0.000%)
      Frag6:            0 (  0.000%)
      ICMP6:            0 (  0.000%)
       UDP6:            0 (  0.000%)
       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%)
```



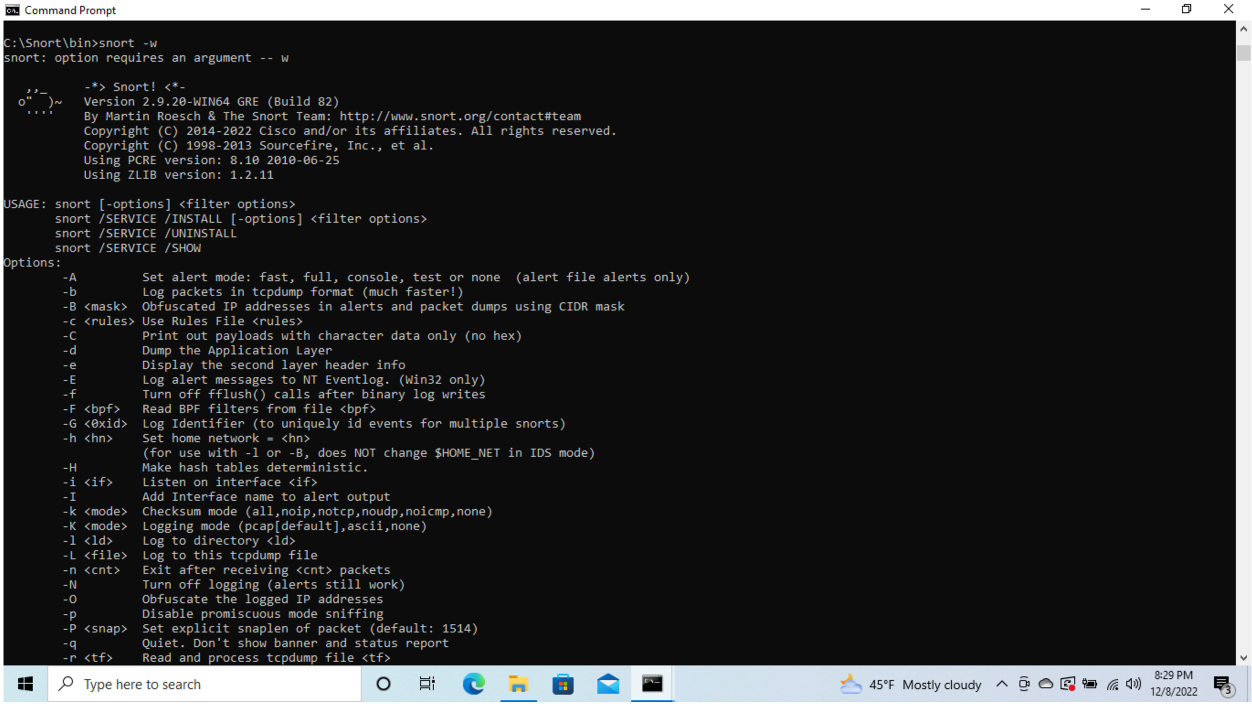
```
   Injected:            0
===============================================================================
Breakdown by protocol (includes rebuilt packets):
        Eth:            0 (  0.000%)
       VLAN:            0 (  0.000%)
        IP4:            0 (  0.000%)
       Frag:            0 (  0.000%)
       ICMP:            0 (  0.000%)
        UDP:            0 (  0.000%)
        TCP:            0 (  0.000%)
        IP6:            0 (  0.000%)
    IP6 Ext:            0 (  0.000%)
   IP6 Opts:            0 (  0.000%)
      Frag6:            0 (  0.000%)
      ICMP6:            0 (  0.000%)
       UDP6:            0 (  0.000%)
       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%)
       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%)
```



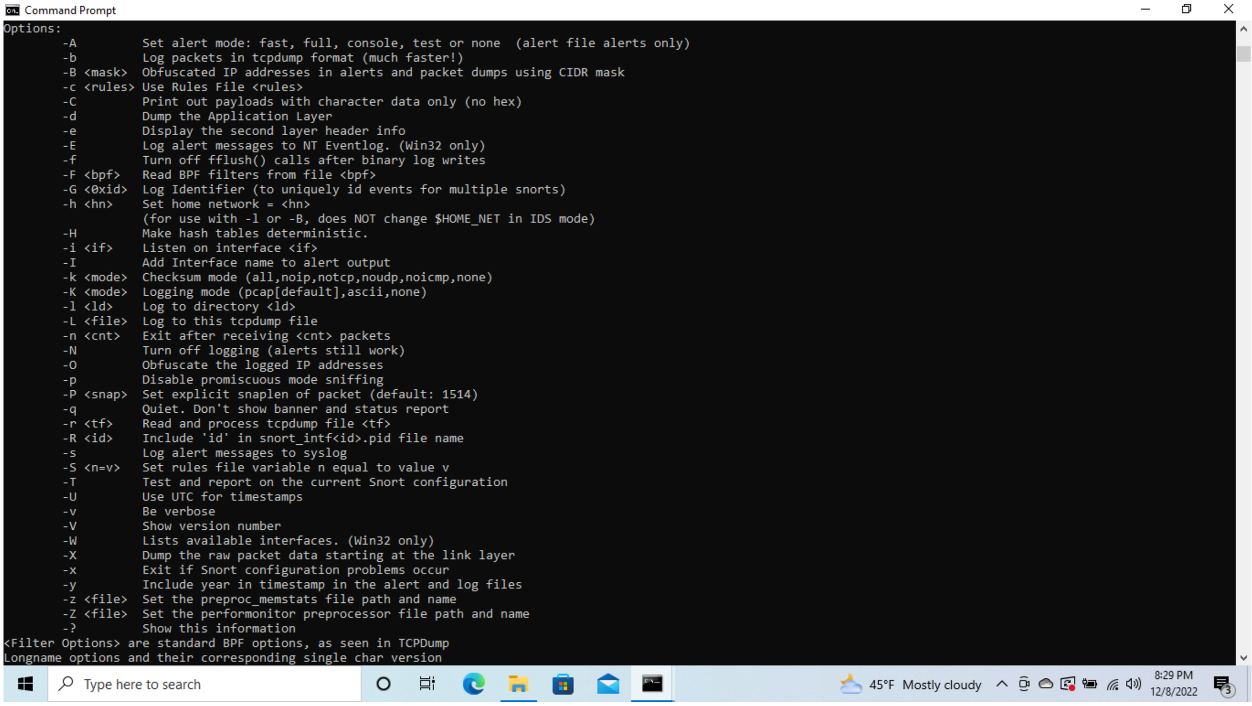
```
         UDP:            0 (  0.000%)
         TCP:            0 (  0.000%)
         IP6:            0 (  0.000%)
     IP6 Ext:            0 (  0.000%)
    IP6 Opts:            0 (  0.000%)
       Frag6:            0 (  0.000%)
       ICMP6:            0 (  0.000%)
        UDP6:            0 (  0.000%)
        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%)
        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:            0 (  0.000%)
 Bad Chk Sum:            0 (  0.000%)
     Bad TTL:            0 (  0.000%)
      S5 G 1:            0 (  0.000%)
      S5 G 2:            0 (  0.000%)
       Total:            0
===============================================================================
```



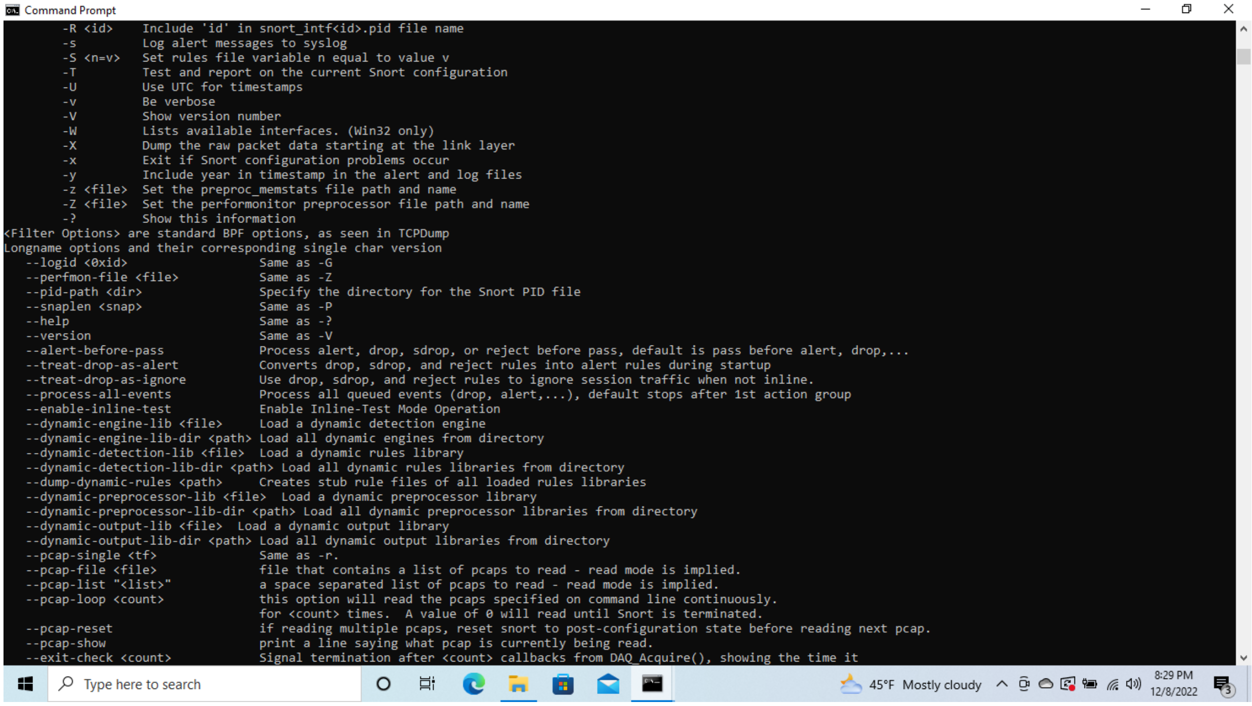
```
===============================================================================

Memory Statistics for File at:Thu Dec  8 19:48:46 2022

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

C:\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)
```



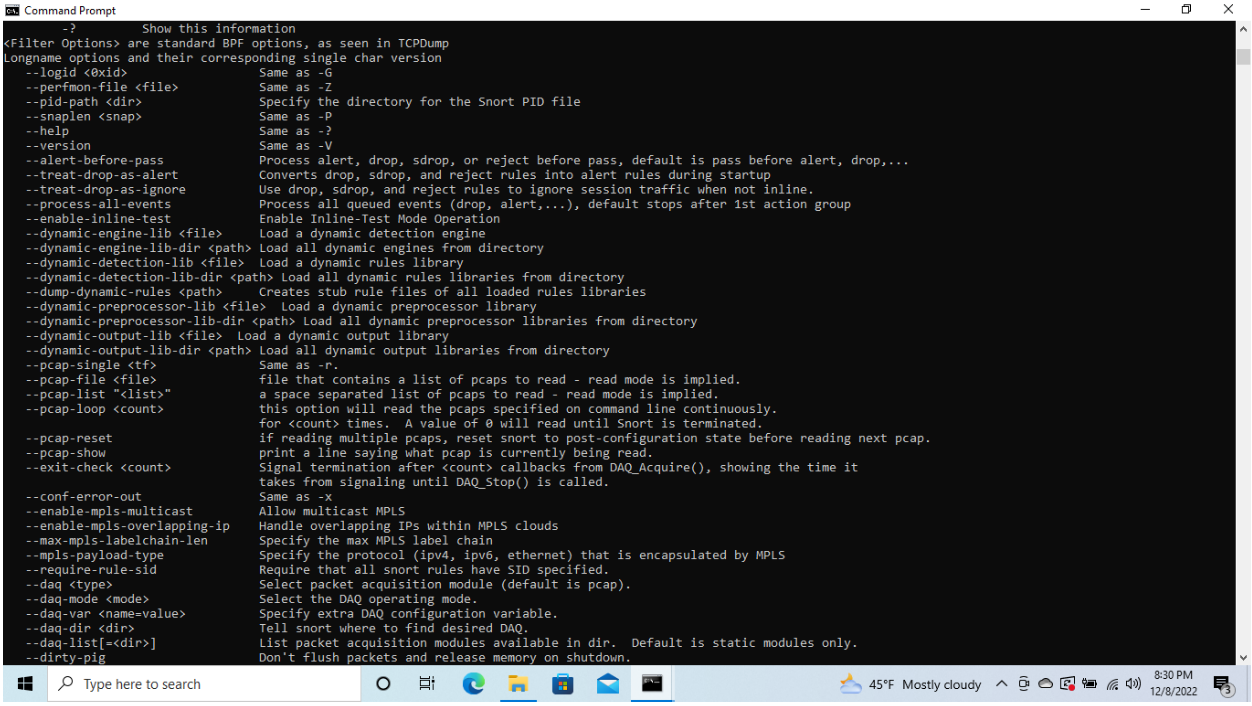
```
C:\Snort\bin>snort -w
snort: option requires an argument -- w

   ,,_     -*> 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>
```



```
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
        -T         Test and report on the current Snort configuration
        -U         Use UTC for timestamps
        -v         Be verbose
        -V         Show version number
        -W         Lists available interfaces. (Win32 only)
        -X         Dump the raw packet data starting at the link layer
        -x         Exit if Snort configuration problems occur
        -y         Include year in timestamp in the alert and log files
        -z <file>  Set the preproc_memstats file path and name
        -Z <file>  Set the perfomonitor preprocessor file path and name
        -?         Show this information
<Filter Options> are standard BPF options, as seen in TCPDump
Longname options and their corresponding single char version
```

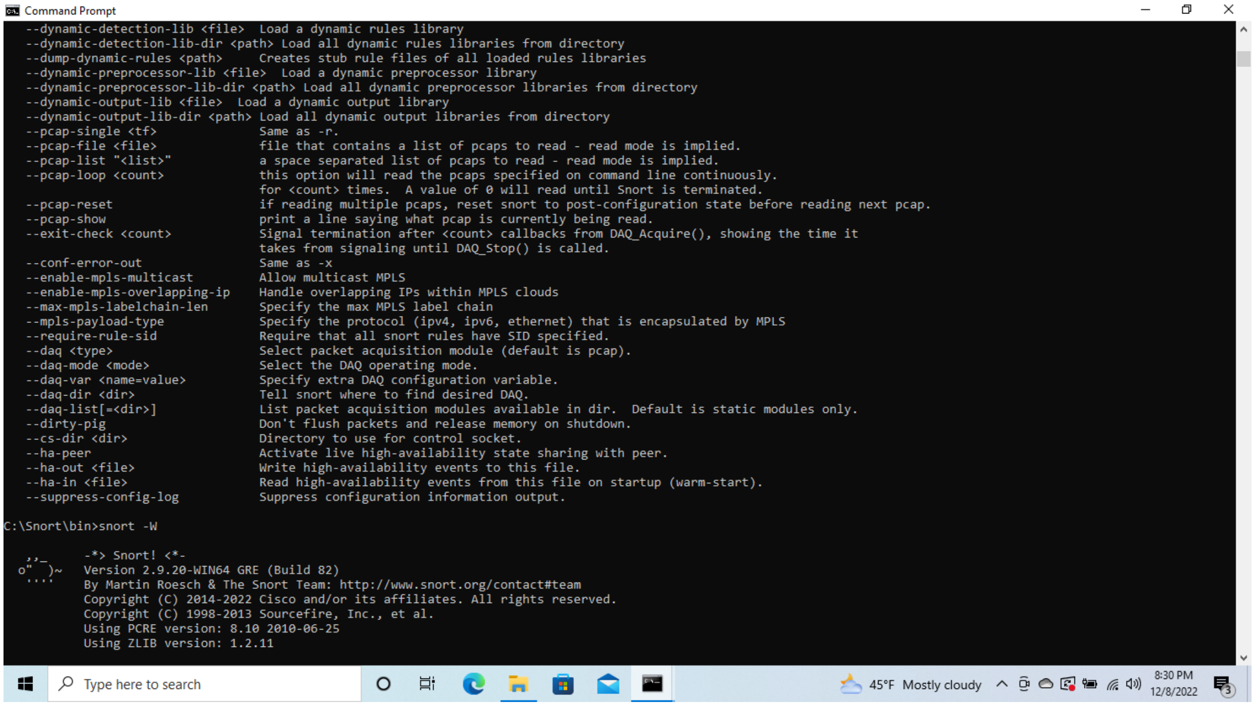


```
        -R <id>    Include 'id' in snort_intf<id>.pid file name
        -s         Log alert messages to syslog
        -S <n=v>   Set rules file variable n equal to value v
        -T         Test and report on the current Snort configuration
        -U         Use UTC for timestamps
        -v         Be verbose
        -V         Show version number
        -W         Lists available interfaces. (Win32 only)
        -X         Dump the raw packet data starting at the link layer
        -x         Exit if Snort configuration problems occur
        -y         Include year in timestamp in the alert and log files
        -z <file>  Set the preproc_memstats file path and name
        -Z <file>  Set the performonitor preprocessor file path and name
        -?         Show this information
<Filter Options> are standard BPF options, as seen in TCPDump
Longname options and their corresponding single char version
   --logid <0xid>                  Same as -G
   --perfmon-file <file>           Same as -Z
   --pid-path <dir>                Specify the directory for the Snort PID file
   --snaplen <snap>                Same as -P
   --help                          Same as -?
   --version                       Same as -V
   --alert-before-pass             Process alert, drop, sdrop, or reject before pass, default is pass before alert, drop,...
   --treat-drop-as-alert           Converts drop, sdrop, and reject rules into alert rules during startup
   --treat-drop-as-ignore          Use drop, sdrop, and reject rules to ignore session traffic when not inline.
   --process-all-events            Process all queued events (drop, alert,...), default stops after 1st action group
   --enable-inline-test            Enable Inline-Test Mode Operation
   --dynamic-engine-lib <file>     Load a dynamic detection engine
   --dynamic-engine-lib-dir <path> Load all dynamic engines from directory
   --dynamic-detection-lib <file>  Load a dynamic rules library
   --dynamic-detection-lib-dir <path> Load all dynamic rules libraries from directory
   --dump-dynamic-rules <path>     Creates stub rule files of all loaded rules libraries
   --dynamic-preprocessor-lib <file>  Load a dynamic preprocessor library
   --dynamic-preprocessor-lib-dir <path> Load all dynamic preprocessor libraries from directory
   --dynamic-output-lib <file>  Load a dynamic output library
   --dynamic-output-lib-dir <path> Load all dynamic output libraries from directory
   --pcap-single <tf>              Same as -r.
   --pcap-file <file>              file that contains a list of pcaps to read - read mode is implied.
   --pcap-list "<list>"            a space separated list of pcaps to read - read mode is implied.
   --pcap-loop <count>             this option will read the pcaps specified on command line continuously.
                                   for <count> times.  A value of 0 will read until Snort is terminated.

   --pcap-reset                    if reading multiple pcaps, reset snort to post-configuration state before reading next pcap.
   --pcap-show                     print a line saying what pcap is currently being read.
   --exit-check <count>            Signal termination after <count> callbacks from DAQ_Acquire(), showing the time it
```

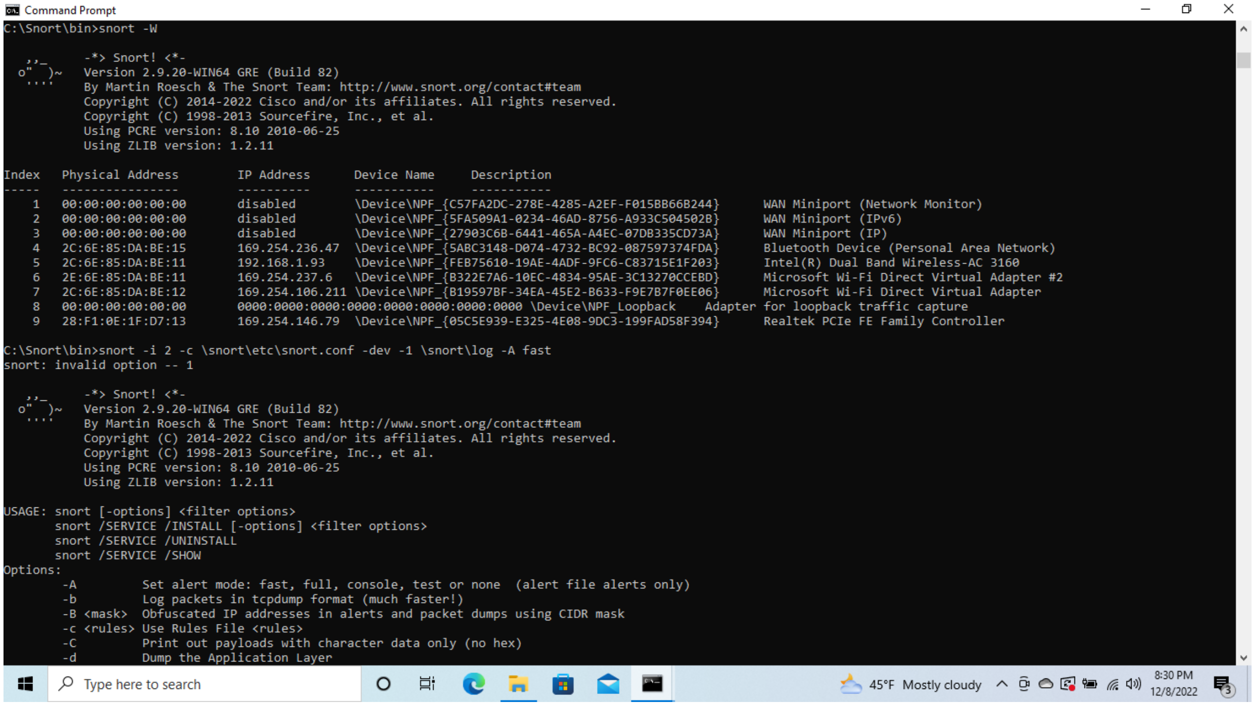


```
        -?         Show this information
<Filter Options> are standard BPF options, as seen in TCPDump
Longname options and their corresponding single char version
   --logid <0xid>                  Same as -G
   --perfmon-file <file>           Same as -Z
   --pid-path <dir>                Specify the directory for the Snort PID file
   --snaplen <snap>                Same as -P
   --help                          Same as -?
   --version                       Same as -V
   --alert-before-pass             Process alert, drop, sdrop, or reject before pass, default is pass before alert, drop,...
   --treat-drop-as-alert           Converts drop, sdrop, and reject rules into alert rules during startup
   --treat-drop-as-ignore          Use drop, sdrop, and reject rules to ignore session traffic when not inline.
   --process-all-events            Process all queued events (drop, alert,...), default stops after 1st action group
   --enable-inline-test            Enable Inline-Test Mode Operation
   --dynamic-engine-lib <file>     Load a dynamic detection engine
   --dynamic-engine-lib-dir <path> Load all dynamic engines from directory
   --dynamic-detection-lib <file>  Load a dynamic rules library
   --dynamic-detection-lib-dir <path> Load all dynamic rules libraries from directory
   --dump-dynamic-rules <path>     Creates stub rule files of all loaded rules libraries
   --dynamic-preprocessor-lib <file>  Load a dynamic preprocessor library
   --dynamic-preprocessor-lib-dir <path> Load all dynamic preprocessor libraries from directory
   --dynamic-output-lib <file>  Load a dynamic output library
   --dynamic-output-lib-dir <path> Load all dynamic output libraries from directory
   --pcap-single <tf>              Same as -r.
   --pcap-file <file>              file that contains a list of pcaps to read - read mode is implied.
   --pcap-list "<list>"            a space separated list of pcaps to read - read mode is implied.
   --pcap-loop <count>             this option will read the pcaps specified on command line continuously.
                                   for <count> times.  A value of 0 will read until Snort is terminated.

   --pcap-reset                    if reading multiple pcaps, reset snort to post-configuration state before reading next pcap.
   --pcap-show                     print a line saying what pcap is currently being read.
   --exit-check <count>            Signal termination after <count> callbacks from DAQ_Acquire(), showing the time it
```

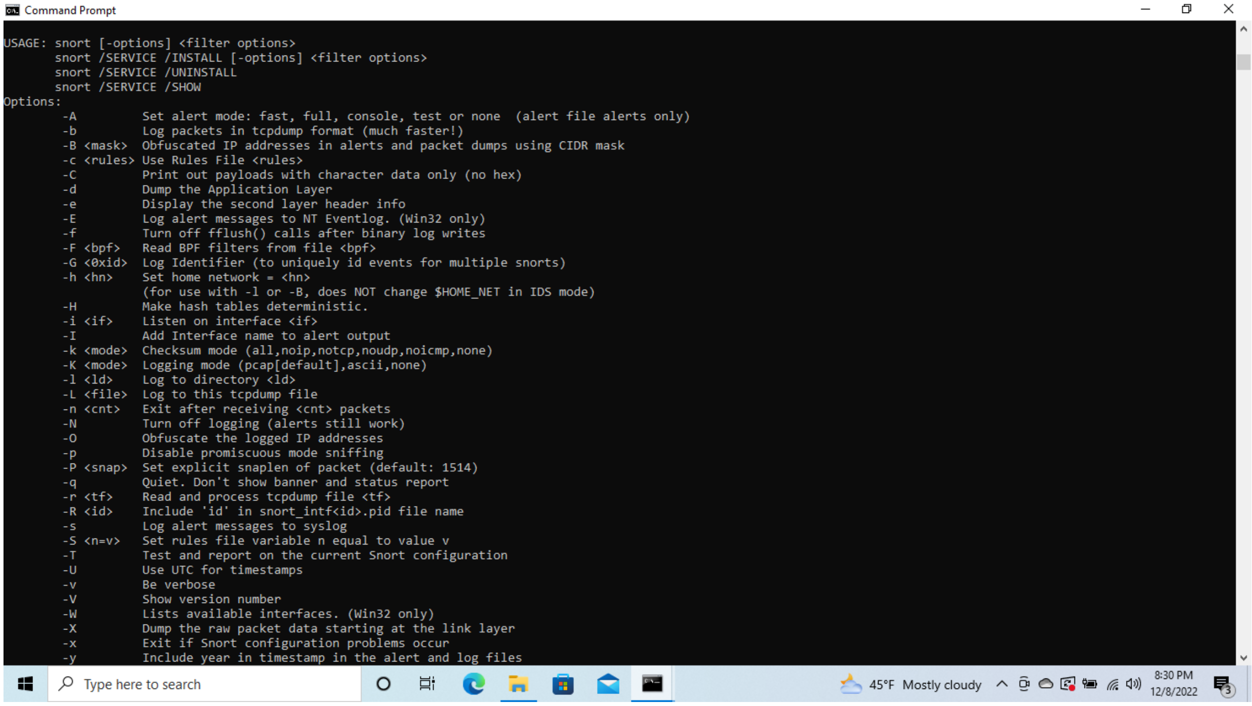
```
                                   takes from signaling until DAQ_Stop() is called.
   --conf-error-out                Same as -x
   --enable-mpls-multicast         Allow multicast MPLS
   --enable-mpls-overlapping-ip    Handle overlapping IPs within MPLS clouds
   --max-mpls-labelchain-len       Specify the max MPLS label chain
   --mpls-payload-type             Specify the protocol (ipv4, ipv6, ethernet) that is encapsulated by MPLS
   --require-rule-sid              Require that all snort rules have SID specified.
   --daq <type>                    Select packet acquisition module (default is pcap).
   --daq-mode <mode>               Select the DAQ operating mode.
   --daq-var <name=value>          Specify extra DAQ configuration variable.
   --daq-dir <dir>                 Tell snort where to find desired DAQ.
   --daq-list[=<dir>]              List packet acquisition modules available in dir.  Default is static modules only.
   --dirty-pig                     Don't flush packets and release memory on shutdown.
```



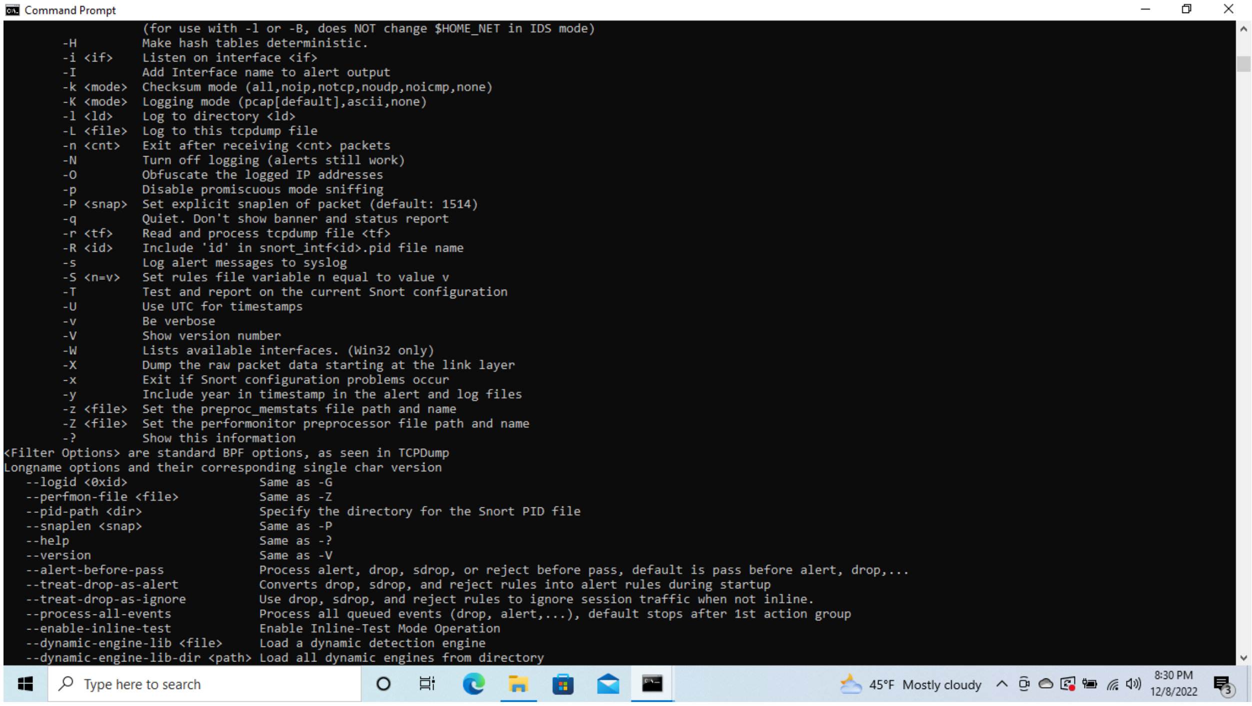
```
  --dynamic-detection-lib <file>  Load a dynamic rules library
  --dynamic-detection-lib-dir <path> Load all dynamic rules libraries from directory
  --dump-dynamic-rules <path>     Creates stub rule files of all loaded rules libraries
  --dynamic-preprocessor-lib <file>  Load a dynamic preprocessor library
  --dynamic-preprocessor-lib-dir <path> Load all dynamic preprocessor libraries from directory
  --dynamic-output-lib <file>  Load a dynamic output library
  --dynamic-output-lib-dir <path> Load all dynamic output libraries from directory
  --pcap-single <tf>              Same as -r.
  --pcap-file <file>              file that contains a list of pcaps to read - read mode is implied.
  --pcap-list "<list>"            a space separated list of pcaps to read - read mode is implied.
  --pcap-loop <count>             this option will read the pcaps specified on command line continuously.
                                  for <count> times.  A value of 0 will read until Snort is terminated.
  --pcap-reset                    if reading multiple pcaps, reset snort to post-configuration state before reading next pcap.
  --pcap-show                     print a line saying what pcap is currently being read.
  --exit-check <count>            Signal termination after <count> callbacks from DAQ_Acquire(), showing the time it
                                  takes from signaling until DAQ_Stop() is called.
  --conf-error-out                Same as -x
  --enable-mpls-multicast         Allow multicast MPLS
  --enable-mpls-overlapping-ip    Handle overlapping IPs within MPLS clouds
  --max-mpls-labelchain-len       Specify the max MPLS label chain
  --mpls-payload-type             Specify the protocol (ipv4, ipv6, ethernet) that is encapsulated by MPLS
  --require-rule-sid              Require that all snort rules have SID specified.
  --daq <type>                    Select packet acquisition module (default is pcap).
  --daq-mode <mode>               Select the DAQ operating mode.
  --daq-var <name=value>          Specify extra DAQ configuration variable.
  --daq-dir <dir>                 Tell snort where to find desired DAQ.
  --daq-list[=<dir>]              List packet acquisition modules available in dir.  Default is static modules only.
  --dirty-pig                     Don't flush packets and release memory on shutdown.
  --cs-dir <dir>                  Directory to use for control socket.
  --ha-peer                       Activate live high-availability state sharing with peer.
  --ha-out <file>                 Write high-availability events to this file.
  --ha-in <file>                  Read high-availability events from this file on startup (warm-start).
  --suppress-config-log           Suppress configuration information output.

C:\Snort\bin>snort -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
```



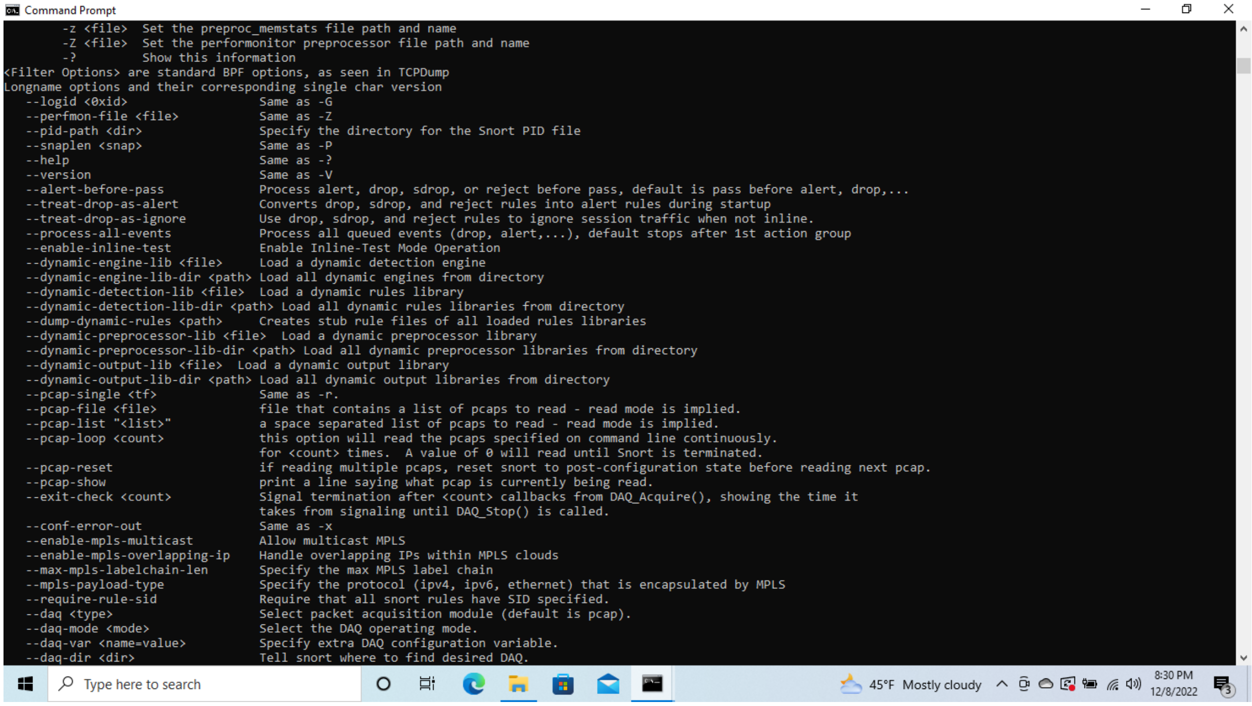
```
C:\Snort\bin>snort -W

   ,,_     -*> 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

Index   Physical Address        IP Address      Device Name     Description
-----   -----------------       ----------      -----------     -----------
    1   00:00:00:00:00:00       disabled        \Device\NPF_{C57FA2DC-278E-4285-A2EF-F015BB66B244}      WAN Miniport (Network Monitor)
    2   00:00:00:00:00:00       disabled        \Device\NPF_{5FA509A1-0234-46AD-8756-A933C504502B}      WAN Miniport (IPv6)
    3   00:00:00:00:00:00       disabled        \Device\NPF_{27903C6B-6441-465A-A4EC-07DB335CD73A}      WAN Miniport (IP)
    4   2C:6E:85:DA:BE:15       169.254.236.47  \Device\NPF_{5ABC3148-D074-4732-BC92-087597374FDA}      Bluetooth Device (Personal Area Network)
    5   2C:6E:85:DA:BE:11       192.168.1.93    \Device\NPF_{FEB75610-19AE-4ADF-9FC6-C83715E1F203}      Intel(R) Dual Band Wireless-AC 3160
    6   2E:6E:85:DA:BE:11       169.254.237.6   \Device\NPF_{B322E7A6-10EC-4834-95AE-3C13270CCEBD}      Microsoft Wi-Fi Direct Virtual Adapter #2
    7   2C:6E:85:DA:BE:12       169.254.106.211 \Device\NPF_{B19597BF-34EA-45E2-B633-F9E7B7F0EE06}      Microsoft Wi-Fi Direct Virtual Adapter
    8   00:00:00:00:00:00       0000:0000:0000:0000:0000:0000:0000:0000 \Device\NPF_Loopback    Adapter for loopback traffic capture
    9   28:F1:0E:1F:D7:13       169.254.146.79  \Device\NPF_{05C5E939-E325-4E08-9DC3-199FAD58F394}      Realtek PCIe FE Family Controller

C:\Snort\bin>snort -i 2 -c \snort\etc\snort.conf -dev -1 \snort\log -A fast
snort: invalid option -- 1

   ,,_     -*> 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
```



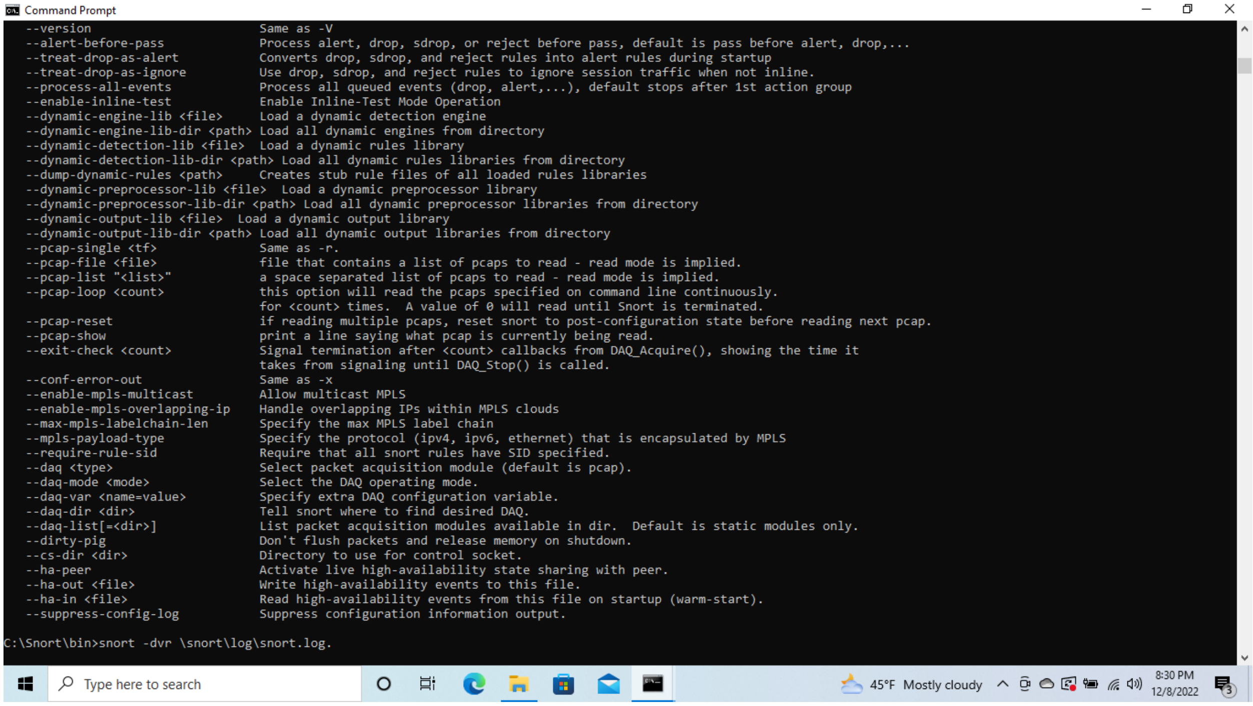
```
USAGE: snort [-options] <filter options>
       snort /SERVICE /INSTALL [-options] <filter options>
       snort /SERVICE /UNINSTALL
       snort /SERVICE /SHOW
Options:
        -A         Set alert mode: fast, full, console, test or none  (alert file alerts only)
        -b         Log packets in 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
        -T         Test and report on the current Snort configuration
        -U         Use UTC for timestamps
        -v         Be verbose
        -V         Show version number
        -W         Lists available interfaces. (Win32 only)
        -X         Dump the raw packet data starting at the link layer
        -x         Exit if Snort configuration problems occur
        -y         Include year in timestamp in the alert and log files
```



```
                   (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
        -T         Test and report on the current Snort configuration
        -U         Use UTC for timestamps
        -v         Be verbose
        -V         Show version number
        -W         Lists available interfaces. (Win32 only)
        -X         Dump the raw packet data starting at the link layer
        -x         Exit if Snort configuration problems occur
        -y         Include year in timestamp in the alert and log files
        -z <file>  Set the preproc_memstats file path and name
        -Z <file>  Set the perfomonitor preprocessor file path and name
        -?         Show this information
<Filter Options> are standard BPF options, as seen in TCPDump
Longname options and their corresponding single char version
   --logid <0xid>                  Same as -G
   --perfmon-file <file>           Same as -Z
   --pid-path <dir>                Specify the directory for the Snort PID file
   --snaplen <snap>                Same as -P
   --help                          Same as -?
   --version                       Same as -V
   --alert-before-pass             Process alert, drop, sdrop, or reject before pass, default is pass before alert, drop,...
   --treat-drop-as-alert           Converts drop, sdrop, and reject rules into alert rules during startup
   --treat-drop-as-ignore          Use drop, sdrop, and reject rules to ignore session traffic when not inline.
   --process-all-events            Process all queued events (drop, alert,...), default stops after 1st action group
   --enable-inline-test            Enable Inline-Test Mode Operation
   --dynamic-engine-lib <file>     Load a dynamic detection engine
   --dynamic-engine-lib-dir <path> Load all dynamic engines from directory
```



```
        -z <file>  Set the preproc_memstats file path and name
        -Z <file>  Set the perfomonitor preprocessor file path and name
        -?         Show this information
<Filter Options> are standard BPF options, as seen in TCPDump
Longname options and their corresponding single char version
   --logid <0xid>                  Same as -G
   --perfmon-file <file>           Same as -Z
   --pid-path <dir>                Specify the directory for the Snort PID file
   --snaplen <snap>                Same as -P
   --help                          Same as -?
   --version                       Same as -V
   --alert-before-pass             Process alert, drop, sdrop, or reject before pass, default is pass before alert, drop,...
   --treat-drop-as-alert           Converts drop, sdrop, and reject rules into alert rules during startup
   --treat-drop-as-ignore          Use drop, sdrop, and reject rules to ignore session traffic when not inline.
   --process-all-events            Process all queued events (drop, alert,...), default stops after 1st action group
   --enable-inline-test            Enable Inline-Test Mode Operation
   --dynamic-engine-lib <file>     Load a dynamic detection engine
   --dynamic-engine-lib-dir <path> Load all dynamic engines from directory
   --dynamic-detection-lib <file>  Load a dynamic rules library
   --dynamic-detection-lib-dir <path> Load all dynamic rules libraries from directory
   --dump-dynamic-rules <path>     Creates stub rule files of all loaded rules libraries
   --dynamic-preprocessor-lib <file>  Load a dynamic preprocessor library
   --dynamic-preprocessor-lib-dir <path> Load all dynamic preprocessor libraries from directory
   --dynamic-output-lib <file>  Load a dynamic output library
   --dynamic-output-lib-dir <path> Load all dynamic output libraries from directory
   --pcap-single <tf>              Same as -r.
   --pcap-file <file>              file that contains a list of pcaps to read - read mode is implied.
   --pcap-list "<list>"            a space separated list of pcaps to read - read mode is implied.
   --pcap-loop <count>             this option will read the pcaps specified on command line continuously.
                                   for <count> times.  A value of 0 will read until Snort is terminated.
   --pcap-reset                    if reading multiple pcaps, reset snort to post-configuration state before reading next pcap.
   --pcap-show                     print a line saying what pcap is currently being read.
   --exit-check <count>            Signal termination after <count> callbacks from DAQ_Acquire(), showing the time it
                                   takes from signaling until DAQ_Stop() is called.
   --conf-error-out                Same as -x
   --enable-mpls-multicast         Allow multicast MPLS
   --enable-mpls-overlapping-ip    Handle overlapping IPs within MPLS clouds
   --max-mpls-labelchain-len       Specify the max MPLS label chain
   --mpls-payload-type             Specify the protocol (ipv4, ipv6, ethernet) that is encapsulated by MPLS
   --require-rule-sid              Require that all snort rules have SID specified.
   --daq <type>                    Select packet acquisition module (default is pcap).
   --daq-mode <mode>               Select the DAQ operating mode.
   --daq-var <name=value>          Specify extra DAQ configuration variable.
   --daq-dir <dir>                 Tell snort where to find desired DAQ.
```



```
   --version                       Same as -V
   --alert-before-pass             Process alert, drop, sdrop, or reject before pass, default is pass before alert, drop,...
   --treat-drop-as-alert           Converts drop, sdrop, and reject rules into alert rules during startup
   --treat-drop-as-ignore          Use drop, sdrop, and reject rules to ignore session traffic when not inline.
   --process-all-events            Process all queued events (drop, alert,...), default stops after 1st action group
   --enable-inline-test            Enable Inline-Test Mode Operation
   --dynamic-engine-lib <file>     Load a dynamic detection engine
   --dynamic-engine-lib-dir <path> Load all dynamic engines from directory
   --dynamic-detection-lib <file>  Load a dynamic rules library
   --dynamic-detection-lib-dir <path> Load all dynamic rules libraries from directory
   --dump-dynamic-rules <path>     Creates stub rule files of all loaded rules libraries
   --dynamic-preprocessor-lib <file>  Load a dynamic preprocessor library
   --dynamic-preprocessor-lib-dir <path> Load all dynamic preprocessor libraries from directory
   --dynamic-output-lib <file>  Load a dynamic output library
   --dynamic-output-lib-dir <path> Load all dynamic output libraries from directory
   --pcap-single <tf>              Same as -r.
   --pcap-file <file>              file that contains a list of pcaps to read - read mode is implied.
   --pcap-list "<list>"            a space separated list of pcaps to read - read mode is implied.
   --pcap-loop <count>             this option will read the pcaps specified on command line continuously.
                                   for <count> times.  A value of 0 will read until Snort is terminated.
   --pcap-reset                    if reading multiple pcaps, reset snort to post-configuration state before reading next pcap.
   --pcap-show                     print a line saying what pcap is currently being read.
   --exit-check <count>            Signal termination after <count> callbacks from DAQ_Acquire(), showing the time it
                                   takes from signaling until DAQ_Stop() is called.
   --conf-error-out                Same as -x
   --enable-mpls-multicast         Allow multicast MPLS
   --enable-mpls-overlapping-ip    Handle overlapping IPs within MPLS clouds
   --max-mpls-labelchain-len       Specify the max MPLS label chain
   --mpls-payload-type             Specify the protocol (ipv4, ipv6, ethernet) that is encapsulated by MPLS
   --require-rule-sid              Require that all snort rules have SID specified.
   --daq <type>                    Select packet acquisition module (default is pcap).
   --daq-mode <mode>               Select the DAQ operating mode.
   --daq-var <name=value>          Specify extra DAQ configuration variable.
   --daq-dir <dir>                 Tell snort where to find desired DAQ.
   --daq-list[=<dir>]              List packet acquisition modules available in dir.  Default is static modules only.
   --dirty-pig                     Don't flush packets and release memory on shutdown.
   --cs-dir <dir>                  Directory to use for control socket.
   --ha-peer                       Activate live high-availability state sharing with peer.
   --ha-out <file>                 Write high-availability events to this file.
   --ha-in <file>                  Read high-availability events from this file on startup (warm-start).
   --suppress-config-log           Suppress configuration information output.

C:\Snort\bin>snort -dvr \snort\log\snort.log.
```

GitHub Link:

https://github.com/prachi24s/Intrusion_Detection_System