


**Secure IDS
Configurations and
Management
Module 08**





Configuring Snort IDS


Snort is an open-source network IDS/IPS.

ICON KEY

 Valuable information

 Test your knowledge

 Web exercise

 Workbook review

Lab Scenario

An IDS/IPS is an important network security measure which is used behind a firewall and works from inside the network. An ID inspects the network traffic and looks for heuristics and pattern matches for the intrusions. However, improper IDS configuration and management can make IDPS unworthy. So, IDS configuration and deployment should be performed with careful planning, preparation, prototyping, testing, and specialized training. As a network administrator, you should be able to configure IDS/IPS in your organization's network

Lab Objectives

The objective of this lab is to demonstrate how to configure snort IDS in a network.

In this lab, you will need to:

- Install Snort and verify Snort alerts
- Configure and validate the snort.conf file
- Test Snort is working by carrying out a test attack
- Perform Intrusion detection

Lab Environment

To complete this lab, you will need:

- A virtual machine running Windows Server 2012
- A virtual machine running Windows 10
- Snort located at **Z:\CND-Tools\CND Module 08 Secure IDS Configuration and Management\IDS and IPS Solutions\Snort**
- You can download the latest version of Snort from <https://www.snort.org/downloads>. If you decide to download the latest version, screenshots may differ
- WinPcap drivers installed in Windows Server 2012 virtual machine

- Notepad++ installed in Windows Server 2012 virtual machine
- Administrative privileges to configure settings and run tools

Lab Duration


Time: 25 Minutes

Overview of IPSs and IDSs

An intrusion prevention system is a network security appliance that monitors a network and systems for malicious activity. The IPS's main functions are to identify malicious activity, log information about any activity, attempt to block/stop it, and report it.

An intrusion detection system is a device or software application that monitors a network and/or systems for malicious activity or policy violations and produces reports to a management station. The IDS performs intrusion detection and attempts to stop detected incidents.

Lab Tasks

 **T A S K 1**

Install Snort

Note: Before starting this lab Turn off Windows Firewall in the Administrator's machine i.e., **Windows Server 2012**, for the lab demonstration purpose. Later you can turn on the Firewall.

1. Launch **Windows server 2012** Virtual machine.
2. To install Snort, navigate to **Z:\CND-Tools\CND Module 08 Secure IDS Configuration and Management\IDS and IPS Solutions\Snort**
3. Double-click the **Snort_2_9_8_3_Installer.exe** file. The Snort installation wizard appears
4. If **Open File - Security Warning** pop-up appears, click **Run**

5. Accept the **License Agreement**, and install Snort by selecting the default options that appear **step by step** in the wizard

Snort is an open source network intrusion prevention and detection system (IDS/IPS).

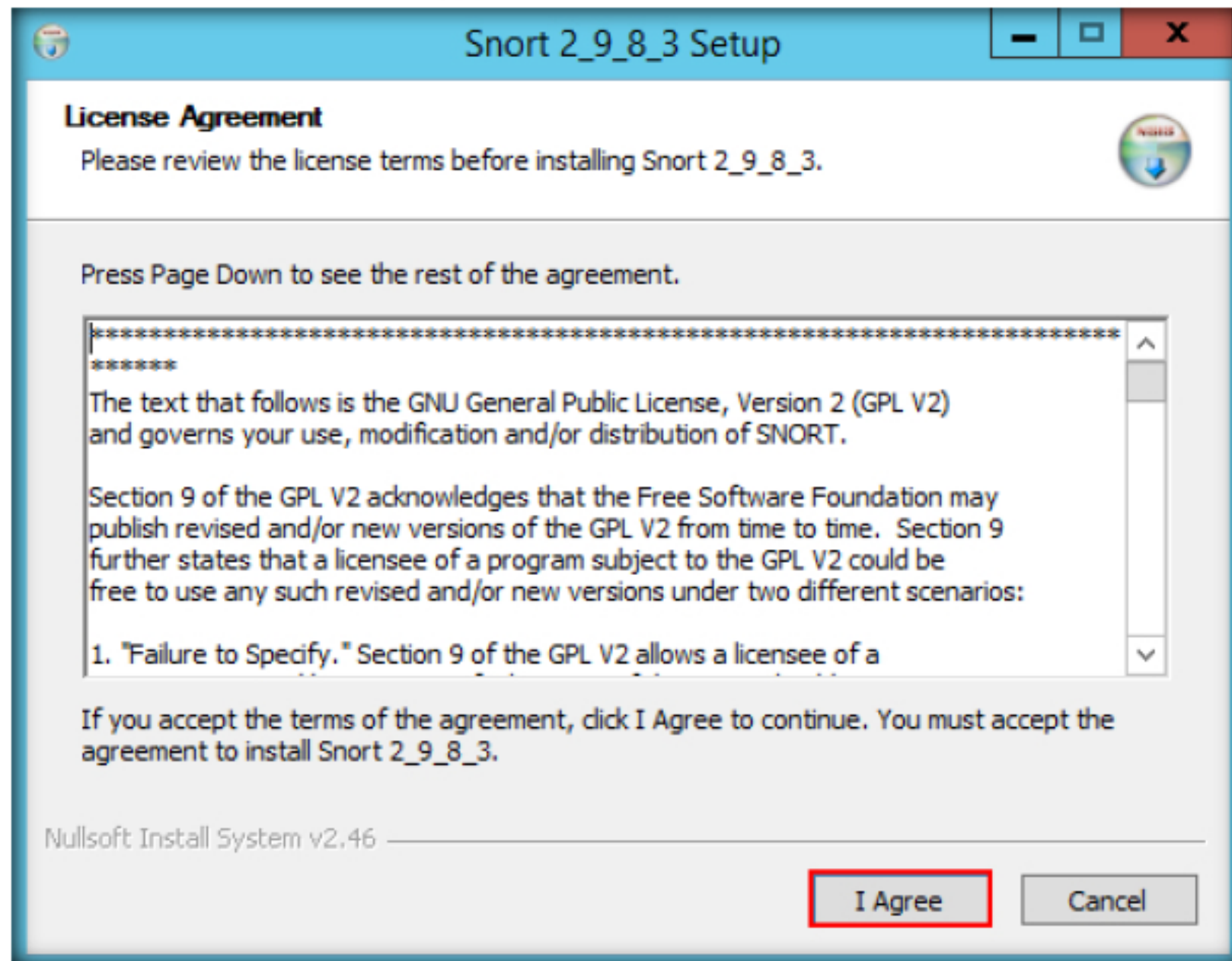


FIGURE 1.1: License Agreement

6. A window appears after the successful installation of Snort. Click **Close**

You can also download Snort from <http://www.snort.org>.

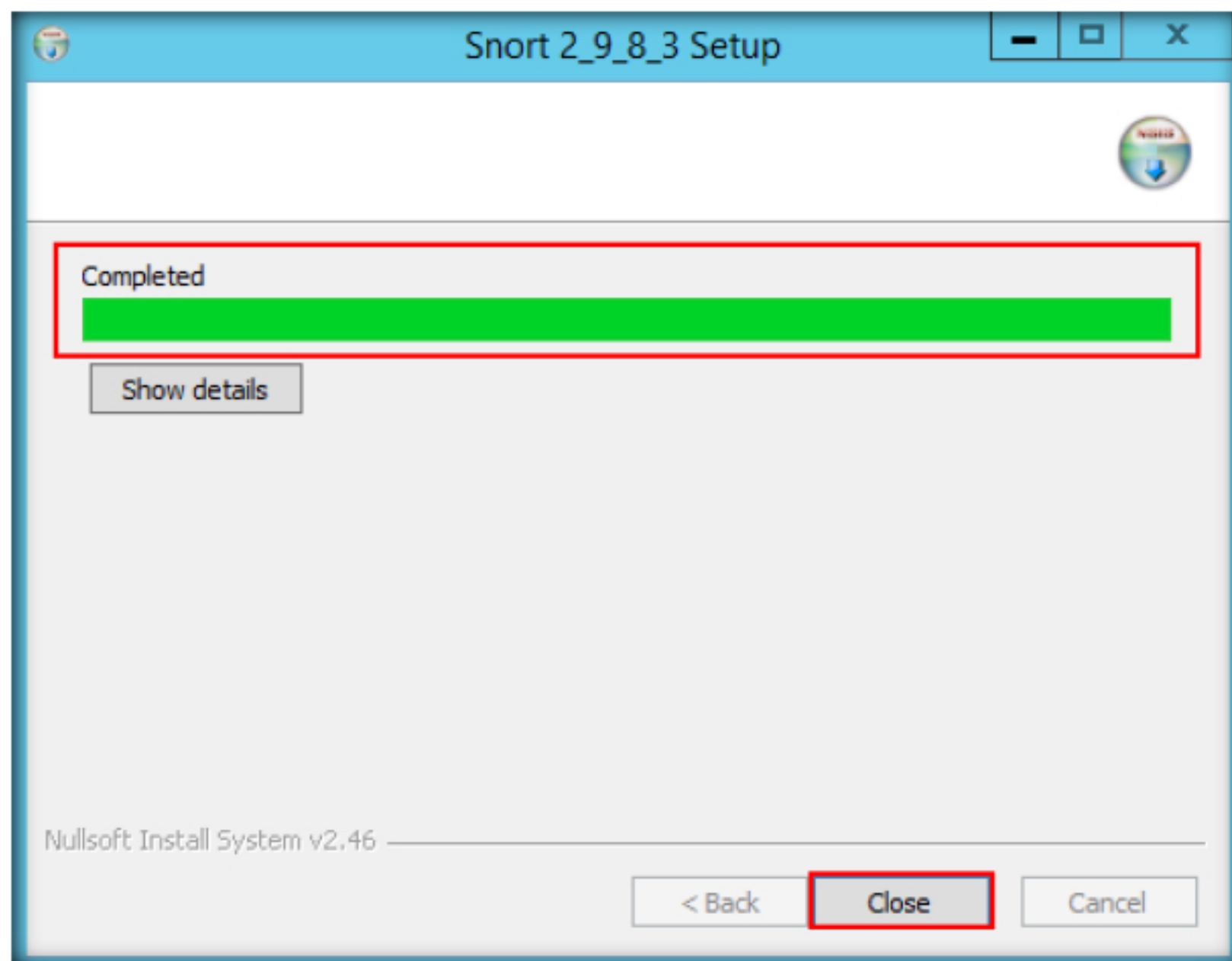



FIGURE 1.2: Snort Setup completed

7. Snort requires **WinPcap** to be installed on your machine.
8. If you have already installed the application click **OK** to exit the **Snort Installation** window and skip to the next step

 WinPcap is a tool for link-layer network access that allows applications to capture and transmit network packets to bypass the protocol stack.

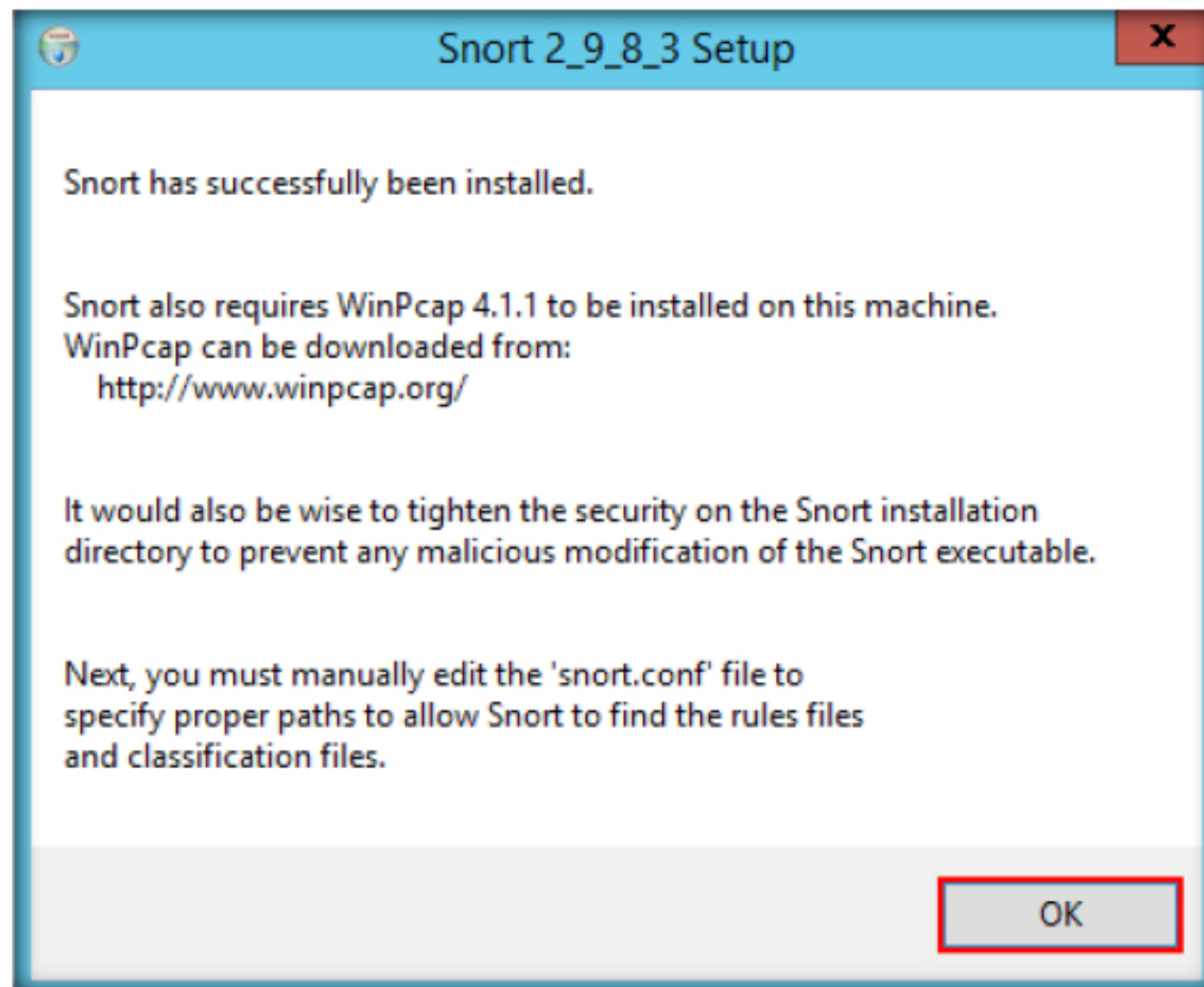



FIGURE 1.3: Snort Successful Installation Window

9. By default, Snort installs itself in **C:\Snort** (depending on the disk drive in which the OS is installed)
10. Navigate to the **etc** folder in the specified location, **Z:\CND-Tools\CND Module 08 Secure IDS Configuration and Management\IDS and IPS Solutions\Snort\snortrules\etc** copy **snort.conf**, and paste it in **C:\Snort\etc**
11. If **Snort.conf** is already present in **C:\Snort\etc**; replace it with the snortrule's **snort.conf** file
12. Copy the **so_rules** folder from **Z:\CND-Tools\CND Module 08 Secure IDS Configuration and Management\IDS and IPS Solutions\Snort\snortrules**, and paste it in **C:\Snort**
13. Copy the **preproc_rules** folder from **Z:\CND-Tools\CND Module 08 Secure IDS Configuration and Management\IDS and IPS Solutions\Snort\snortrules**, and paste it in **C:\Snort**. The **preproc_rules** folder is already present in **C:\Snort**; replace this folder with the **preproc_rules** folder taken from snortrules
14. In the same way, copy the **rules** folder from **Z:\CND-Tools\CND Module 08 Secure IDS Configuration and Management\IDS and IPS Solutions\Snort\snortrules**, and paste it in **C:\Snort**. The **rules** folder is already present in **C:\Snort**; replace the folder

 To print out the TCP/IP packet headers to the screen (i.e., sniffer mode), type: `snort -v`.

TASK 2

Verify Snort

- Now navigate to **C:\Snort**, and press **Shift + right-click** on **bin**; click **Open command window here** from the context menu to open it in a command prompt

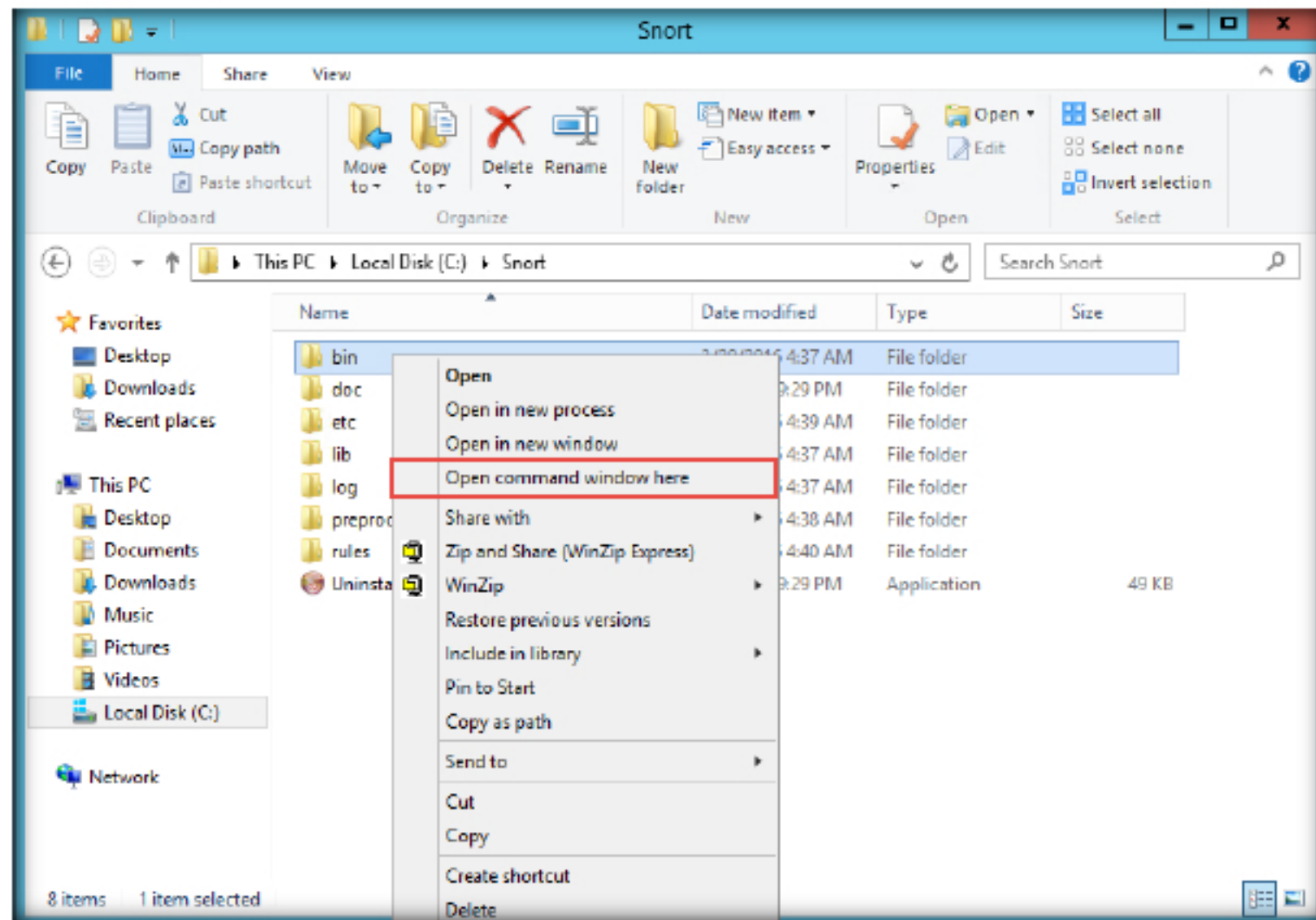


FIGURE 1.4: Starting Command Prompt from Snort\bin

- The **Command Prompt** window appears. Type **snort** and press **Enter**
- A rapid scrolling text will appear in the command terminal, scroll up. This command will show you that snort is configured in your machine successfully. The Process ID will differ in your lab environment.

To specify a log into logging directory, type `snort -dev -l /logdirectorylocationand`, Snort automatically knows to go into packet logger mode.

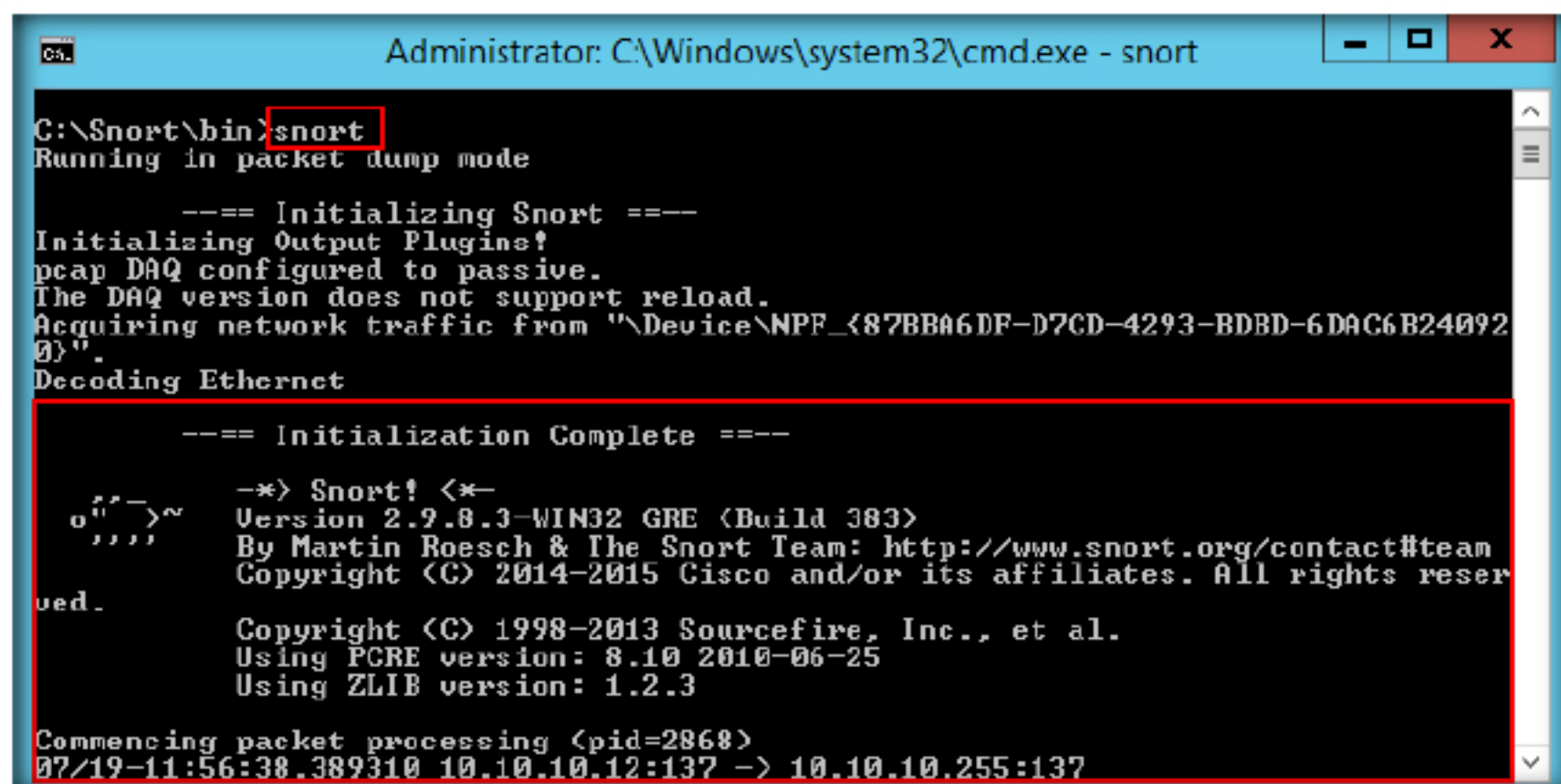

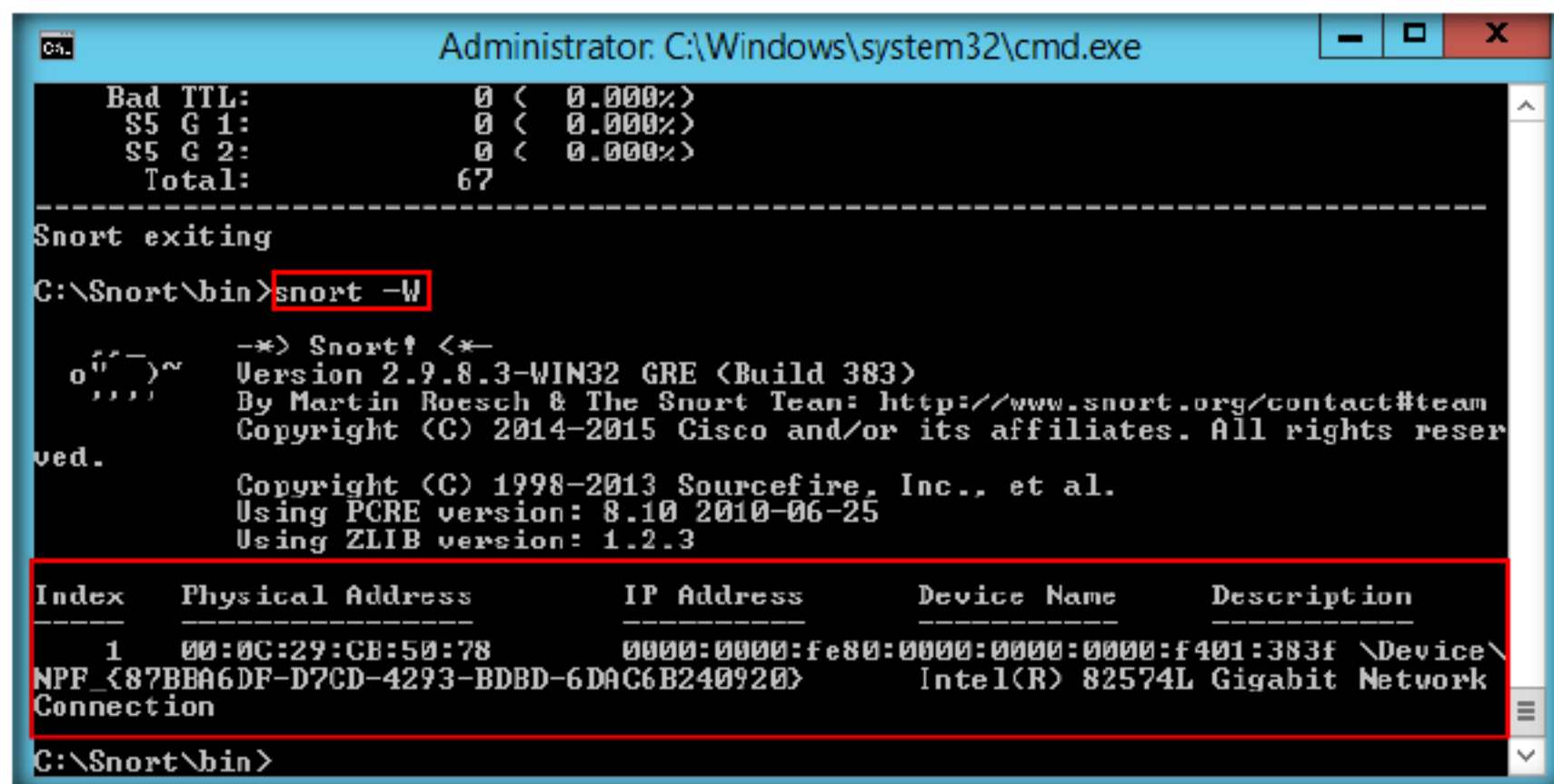


FIGURE 1.5: Basic Snort Command

- The **Initialization Complete** message displays. Press **Ctrl+C**. Snort exits and comes back to **C:\Snort\bin**

- Now type **snort -W**. This command lists your machine's Physical Address, IP Address, and Ethernet Drivers, but all are disabled by default

 Ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS] [-r count] [-s count] [[-j host-list] | [-k host-list]] [-w timeout] destination-list.



```

Bad TTL:          0 ( 0.000%)
S5 G 1:          0 ( 0.000%)
S5 G 2:          0 ( 0.000%)
Total:           67
-----
Snort exiting
C:\Snort\bin>snort -W

-*> Snort! <*-
Version 2.9.8.3-WIN32 GRE (Build 383)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2015 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: 1.2.3

Index  Physical Address      IP Address      Device Name      Description
-----
1      00:0C:29:CB:50:78         0000:0000:fe80:0000:0000:f401:383f \Device\NPF_{87BBA6DF-D7CD-4293-BDBD-6DAC6B240920} Intel(R) 82574L Gigabit Network Connection
C:\Snort\bin>
    
```

FIGURE 1.6: Snort -W Command


- Observe your Ethernet Driver **index number** and write it down (in this lab, it is **1**)

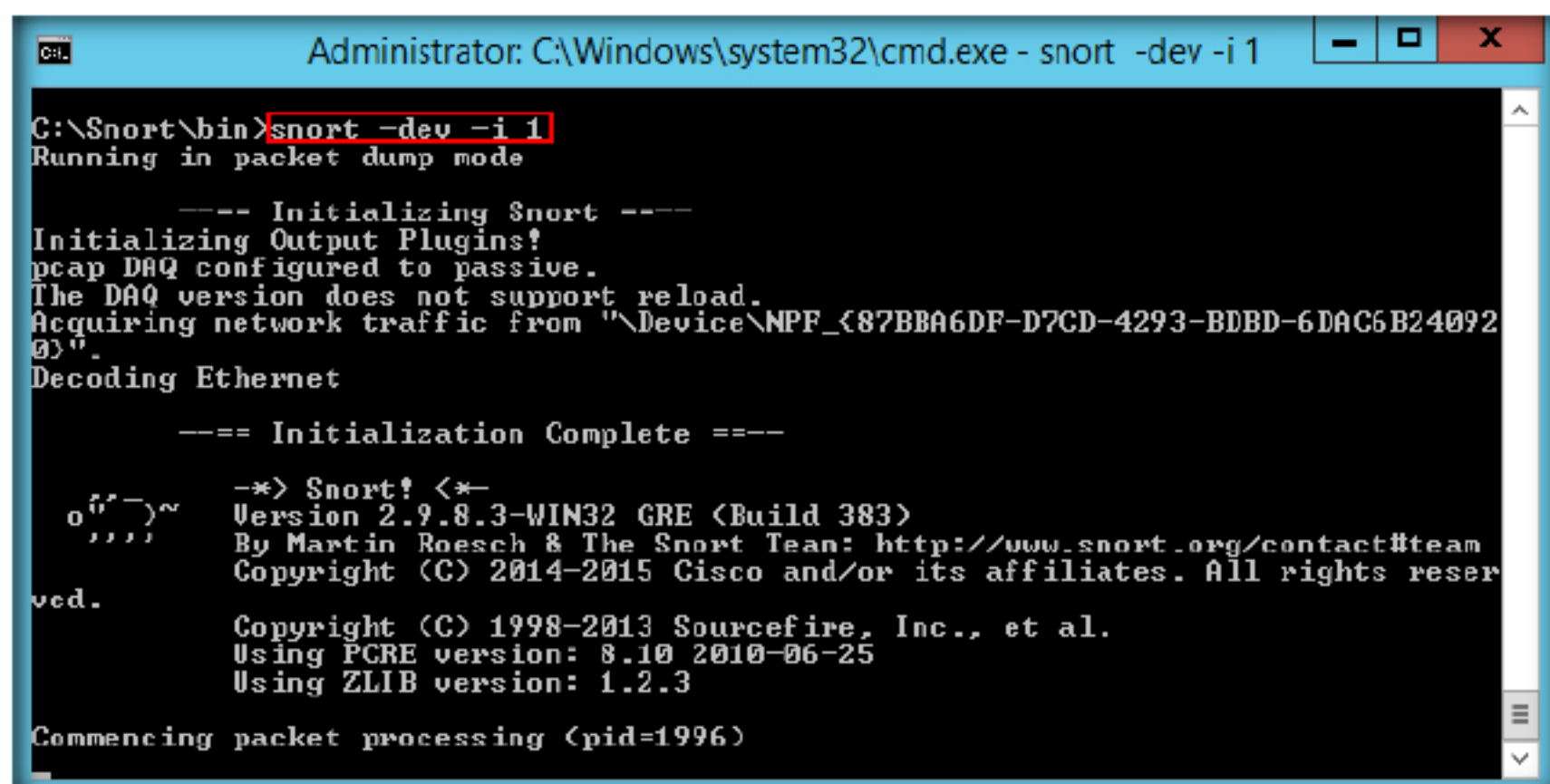
Note: Ethernet Index numbers may vary in your lab environment, if two or more drivers are installed.

- To enable the Ethernet Driver, in the command prompt, type **snort -dev -i 1** and press **Enter**

Note: In the command 1 is nothing but the Index number of the Ethernet adapter installed in your machine.

- You see a rapid scroll text in the command prompt, which means that the Ethernet Driver is enabled and working properly

 To enable Network Intrusion Detection System (NIDS) mode so that you don't record every single packet sent down the wire, type: `snort -dev -l ./log -h 192.168.1.0/24 -c snort.conf`.



```

Administrator: C:\Windows\system32\cmd.exe - snort -dev -i 1
C:\Snort\bin>snort -dev -i 1
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_{87BBA6DF-D7CD-4293-BDBD-6DAC6B240920}"
Decoding Ethernet

=== Initialization Complete ===

-*> Snort! <*-
Version 2.9.8.3-WIN32 GRE (Build 383)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2015 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: 1.2.3

Commencing packet processing (pid=1996)
    
```

FIGURE 1.7: Snort -dev -i 1 Command

 The frag3 preprocessor is a target-based IP defragmentation module for Snort.

23. Leave the Snort command prompt window open, and launch another command prompt. Right-click on the **Start** icon and click the **Command Prompt** from the context menu.

24. In a new command prompt type **ping 10.10.10.10** and press **Enter**

Note: 10.10.10.10 is the IP address of Windows 10 machine.

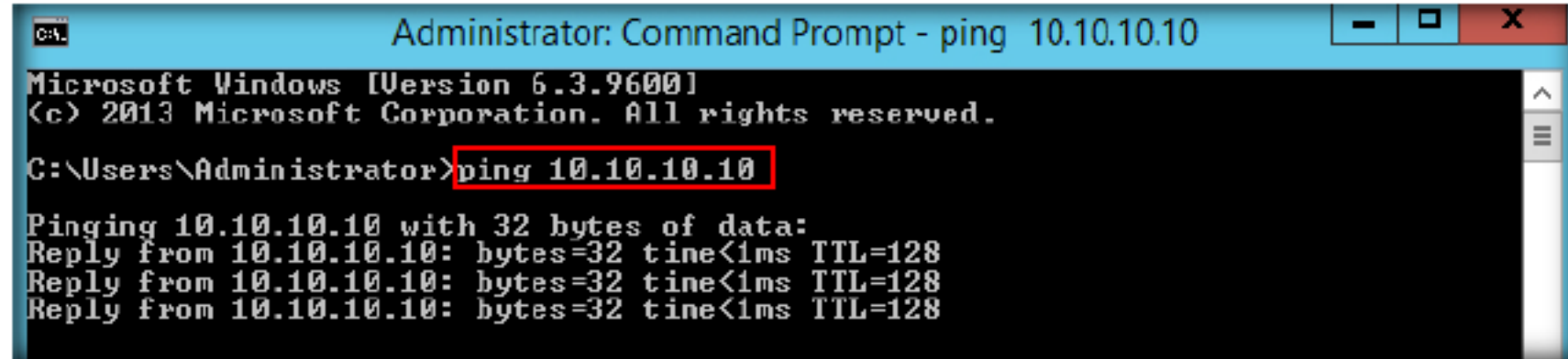



FIGURE 1.8: Ping Command in New Command Prompt

25. This ping command triggers a Snort alert in the Snort command prompt with rapid scrolling text.

 Notepad++ is a free source code editor and Notepad replacement that supports several languages. It runs in the MS Windows environment.

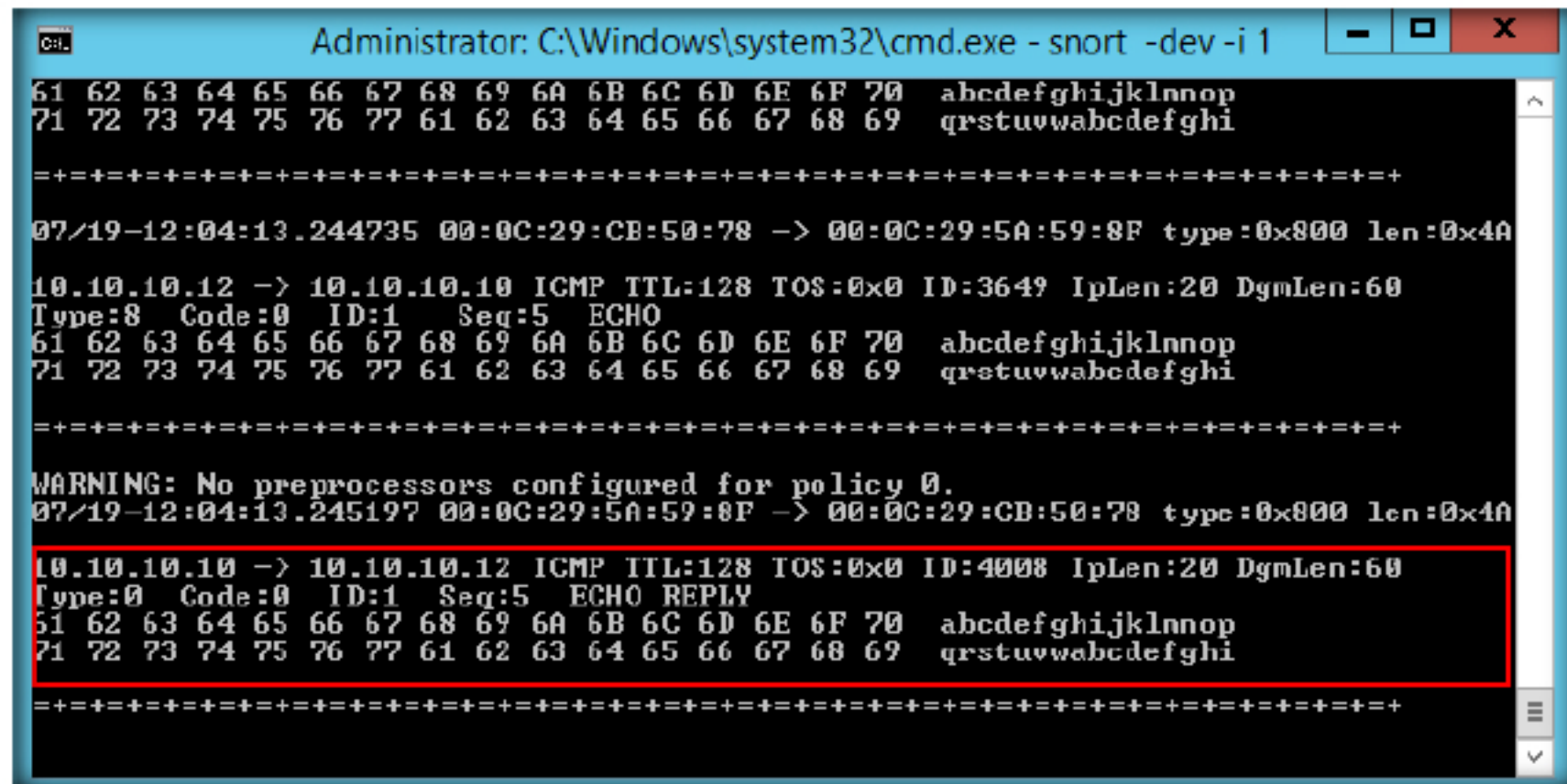


FIGURE 1.9: Snort Showing Captured Ping Request

26. Close both command prompt windows. The verification of the Snort installation and triggering alerts is complete, and Snort is working correctly in verbose mode.

TASK 3

Configure snort.conf File

27. Configure the **snort.conf** file, located at **C:\Snort\etc**
28. Open the **snort.conf** file with Notepad++ using the right-click menu
29. The **snort.conf** file opens in Notepad++, as shown in the screenshot

Note: If any Notepad ++ update pop-up appears close them

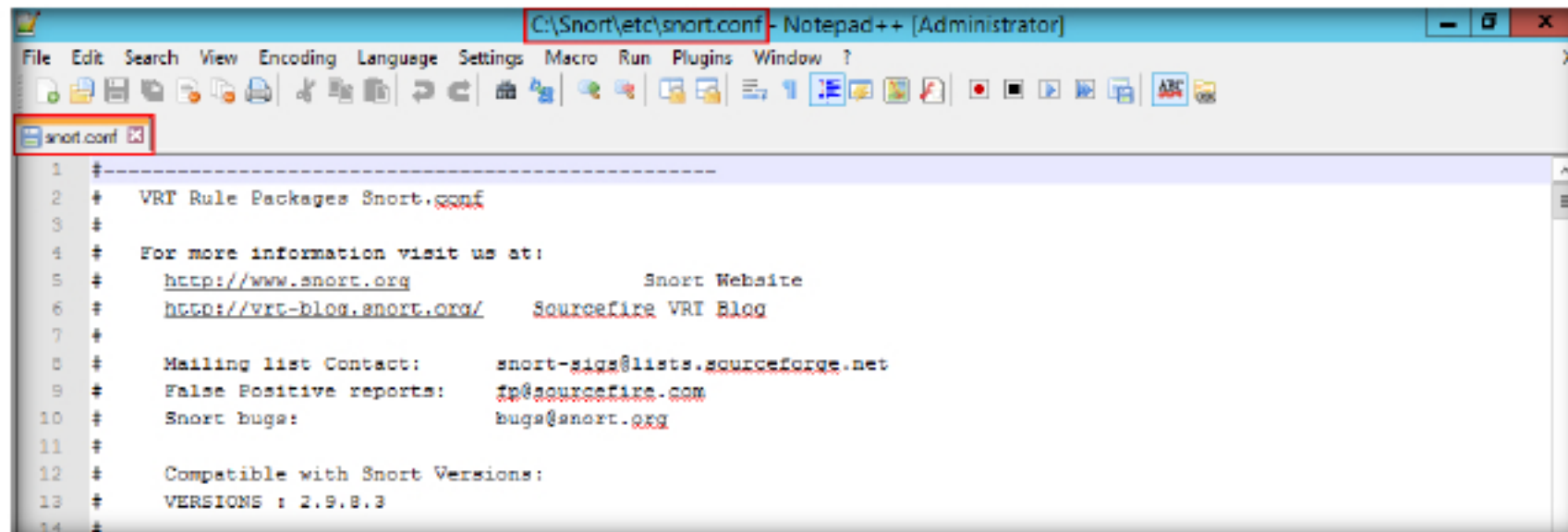



FIGURE 1.10: Snort.conf File in Notepad++

30. Scroll down to **Step #1: Set the network variables** section (Line 41) of snort.conf file. In the **HOME_NET** line (Line 45), replace **any** with the IP address of the machine (Network Administrator Machine) on which Snort is running. Here, the Network Administrator Machine is Windows Server 2012, and the IP address is 10.10.10.12

Note: This IP address may vary in your lab environment

 The element 'any' can be used to match all IPs, although 'any' is not allowed. Also, negated IP ranges that are more general than non-negated IP ranges are not allowed.

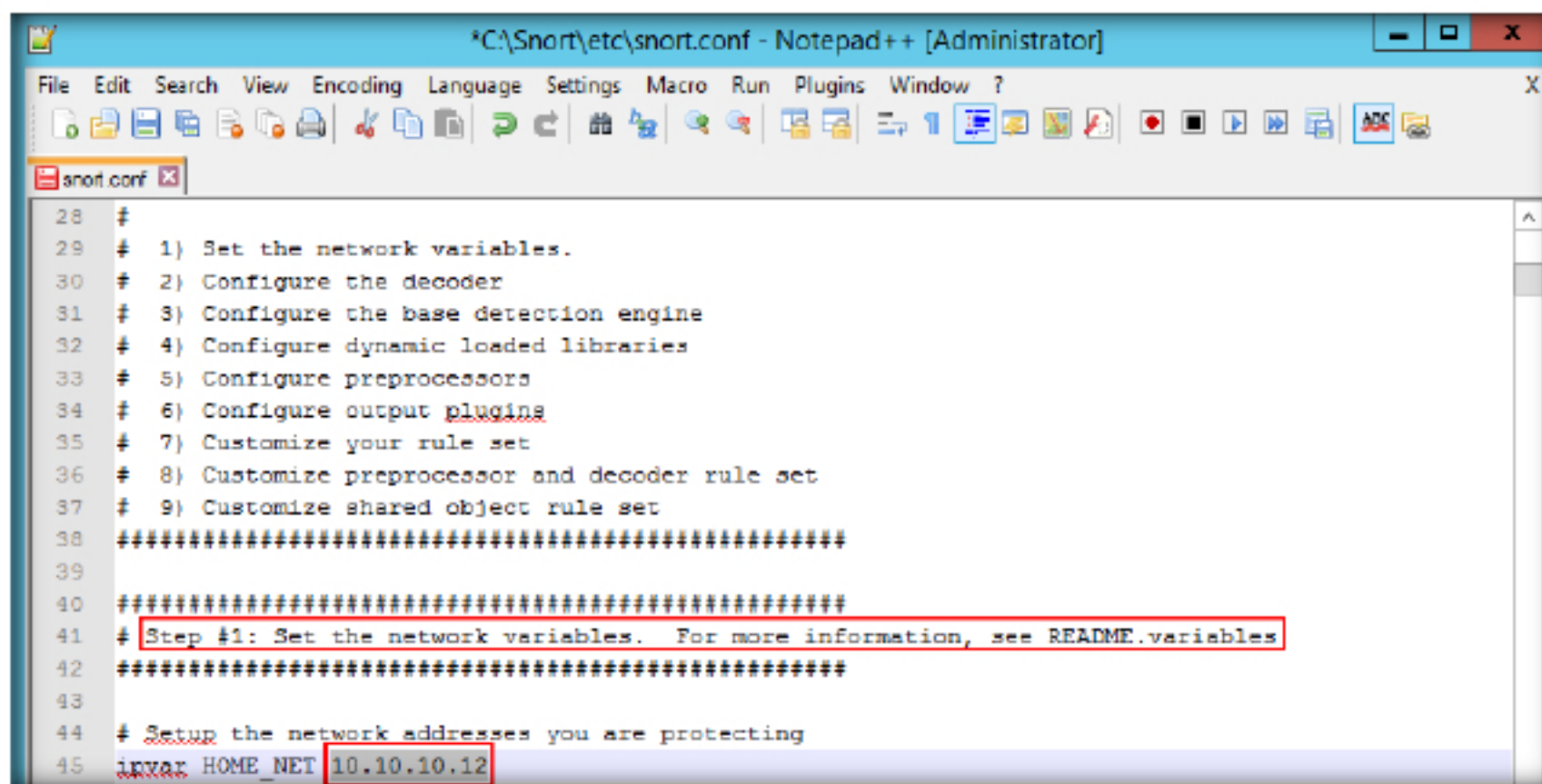


FIGURE 1.11: Configuring Snort.conf File in Notepad++

31. Leave the **EXTERNAL_NET any** line as it is

32. If you have a **DNS Server**, then make changes in the **DNS_SERVERS** line by replacing **\$HOME_NET** with your DNS Server IP address; otherwise, leave this line as it is

Log packets in tcpdump format and to produce minimal alerts, type: snort -b -A fast -c snort.conf.

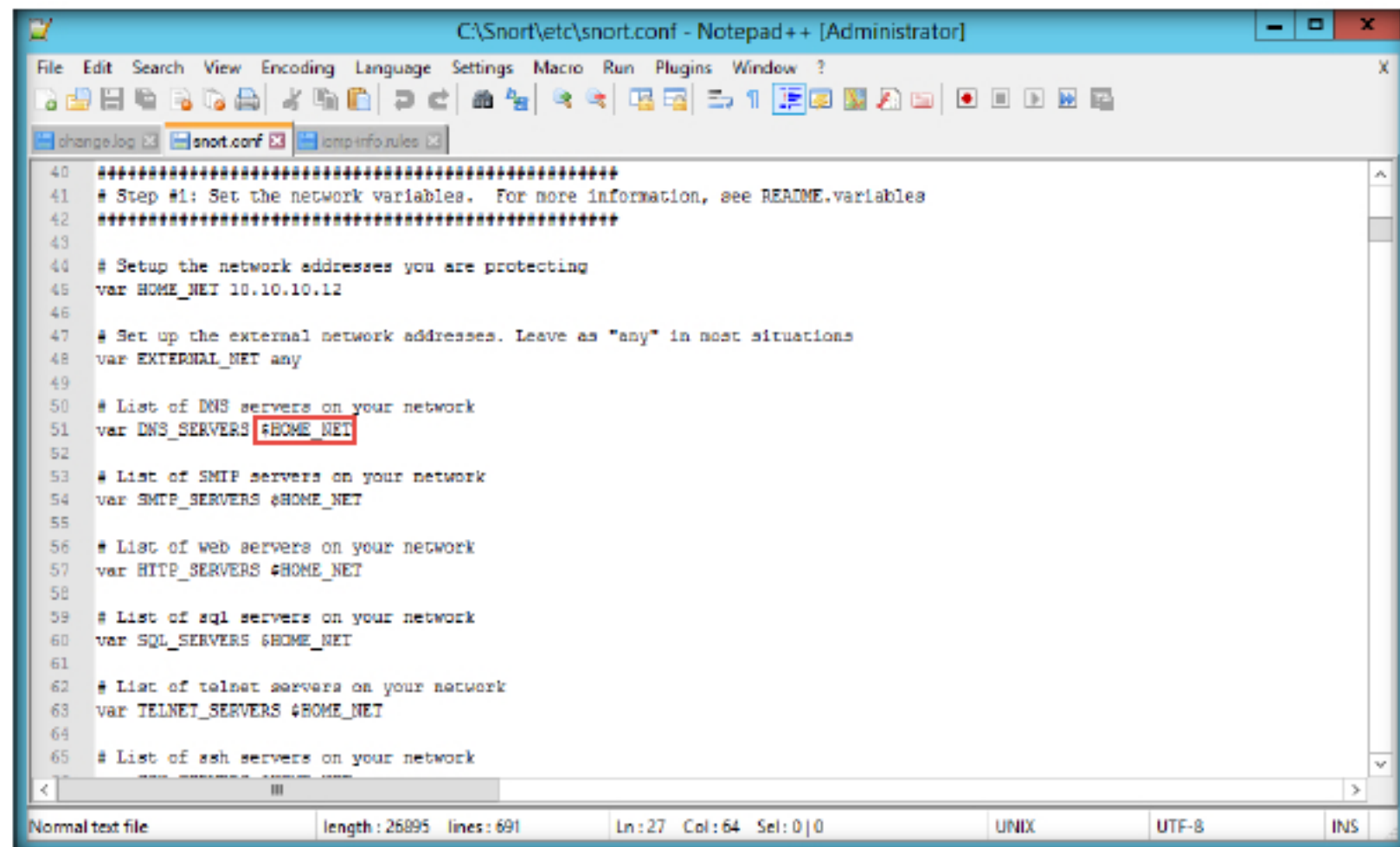


FIGURE 1.12: Configuring Snort.conf File in Notepad++

33. The same applies to SMTP_SERVERS, HTTP_SERVERS, SQL_SERVERS, TELNET_SERVERS, and SSH_SERVERS
34. Remember that if you don't have any servers running on your machine, leave the line as it is. **DO NOT** make any changes in that line
35. Scroll down to **RULE_PATH** (Line 104). In Line 104 replace **../rules** with **C:\Snort\rules**, in Line 105 **../so_rules** replace with **C:\Snort\so_rules**, and in Line 106 replace **../preproc_rules** with **C:\Snort\preproc_rules**

Rule variable names can be modified in several ways. You can define meta-variables using the \$ operator. These can be used with the variable modifier operators ? and -.

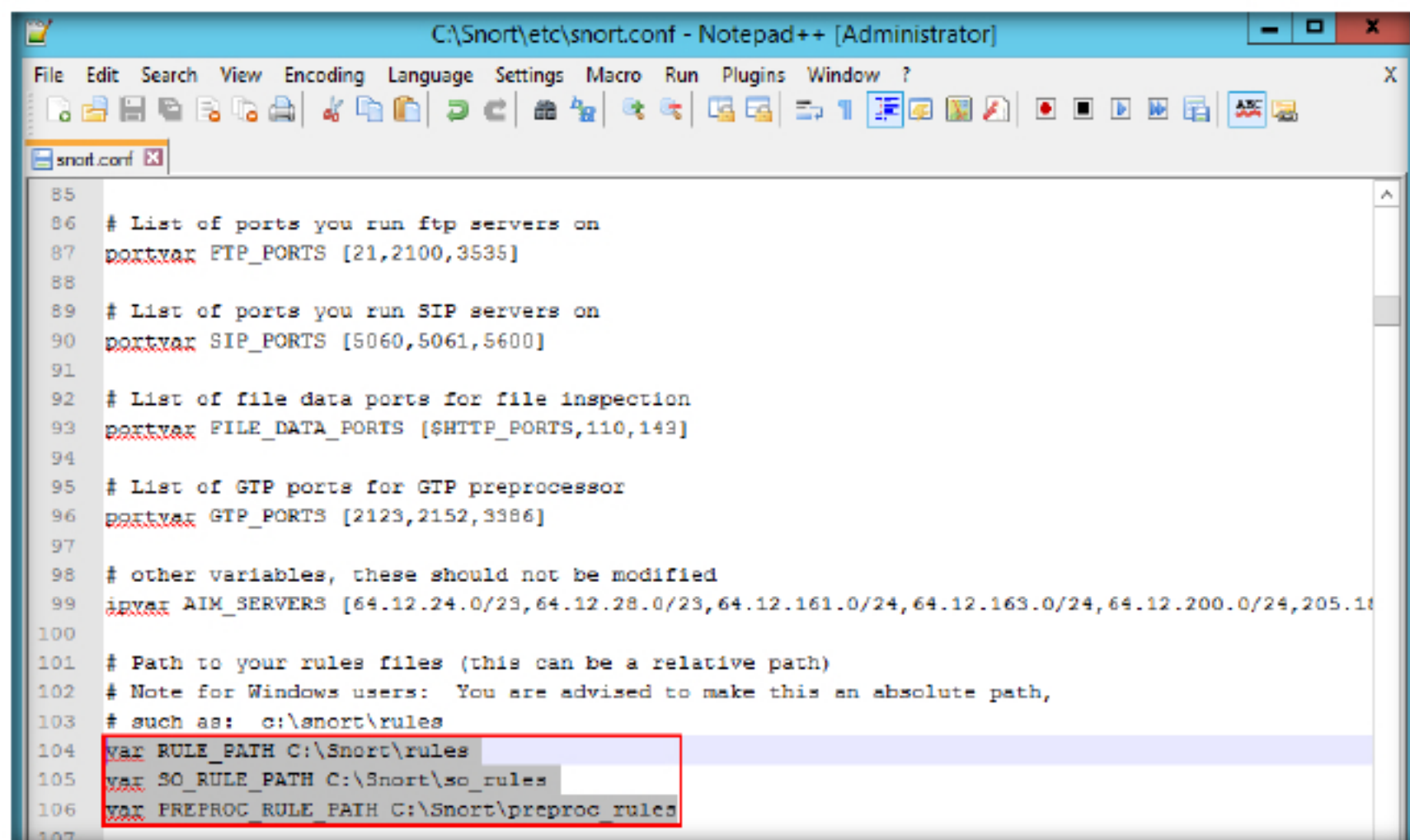



FIGURE 1.13: Configuring Snort.conf File in Notepad++

36. In Lines 109 and 110, replace **../rules** with **C:\Snort\rules**

 The include keyword allows other rule files to be included within the rule file indicated on the Snort command line. It works much like an #include from the C programming language, reading the contents of the named file and adding the contents in the place where the include statement appears in the file.

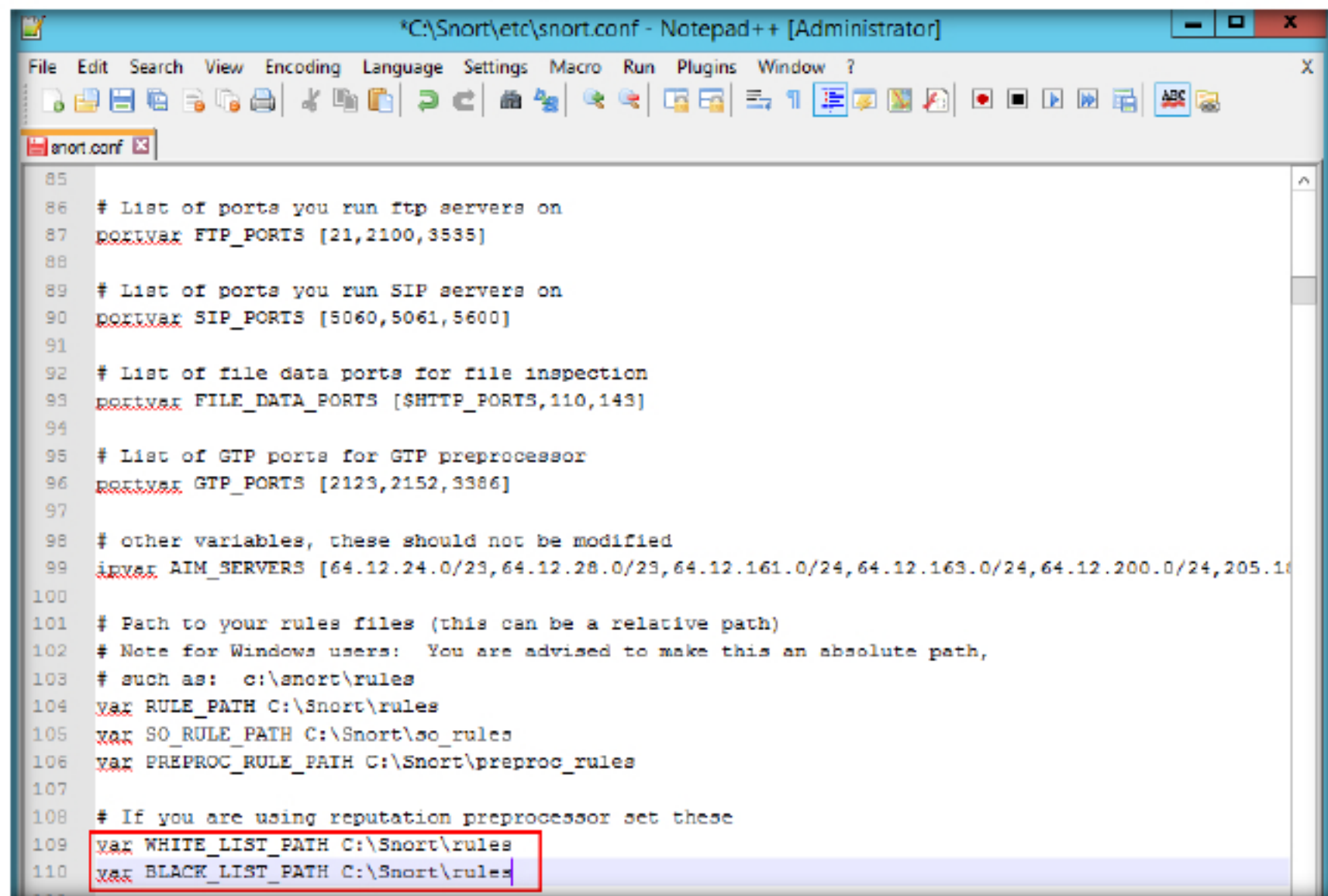



FIGURE 1.14: Configuring Snort.conf File in Notepad++

37. Navigate to **C:\Snort\rules**, and create two text files; name them **white_list** and **black_list** and change their file extensions from **.txt** to **.rules**
38. While changing the extension, if any pop-up appears, click **Yes**
39. Switch back to Notepad ++, scroll down to **Step #4: Configure dynamic loaded libraries** section (Line 238). Configure **dynamic loaded libraries** in this section
40. At the path to dynamic preprocessor libraries (Line 243), replace **/usr/local/lib/snort_dynamicpreprocessor/** with your dynamic preprocessor libraries folder location
41. In this lab, dynamic preprocessor libraries are located at **C:\Snort\lib\snort_dynamicpreprocessor**
42. At the path to base preprocessor (or dynamic) engine (Line 246); replace **/usr/local/lib/snort_dynamicengine/libsf_engine.so** with your base preprocessor engine **C:\Snort\lib\snort_dynamicengine\sf_engine.dll**

 Preprocessors allow the functionality of Snort to be extended by allowing users and programmers to drop modular plug-ins into Snort fairly easily.

43. **Comment (#)** the dynamic rules libraries line as you already configured the libraries in dynamic preprocessor libraries (Line 249)

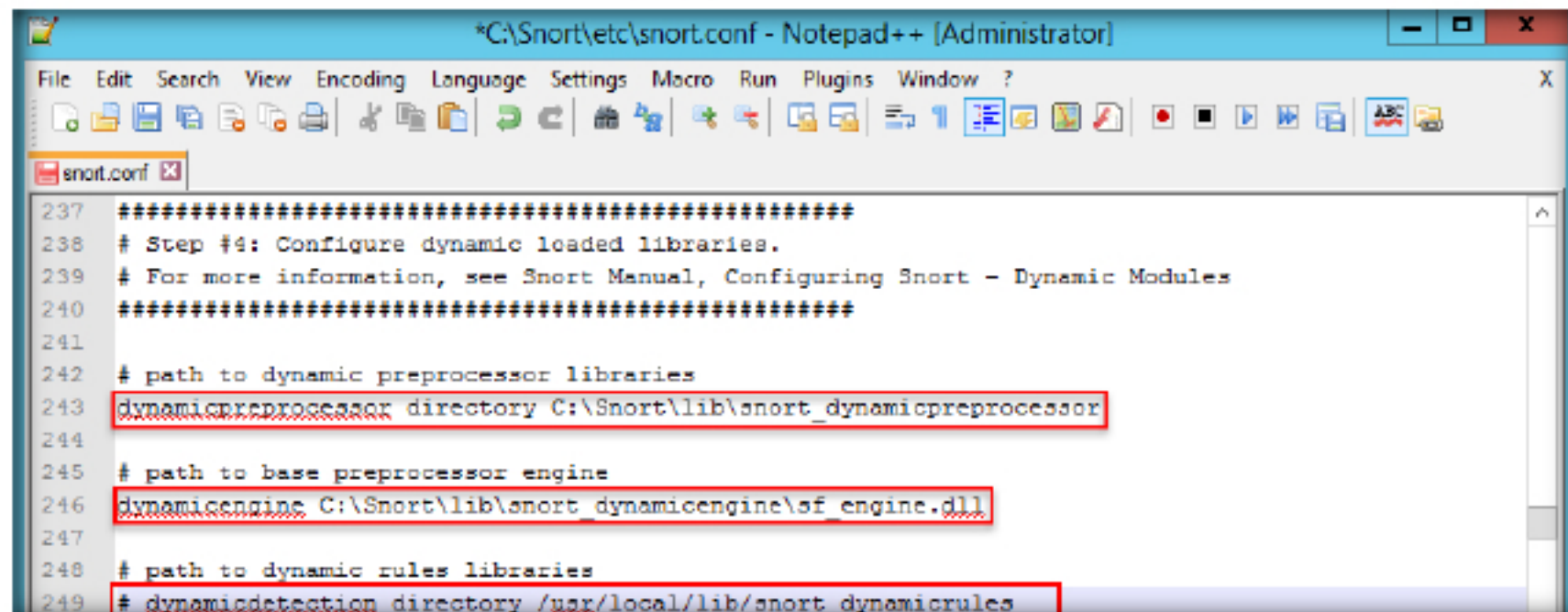



FIGURE 1.15: Configuring Snort.conf File in Notepad++

 Note: Preprocessor code is run before the detection engine is called, but after the packet has been decoded. The packet can be modified or analyzed in an out-of-band manner using this mechanism.

44. Scroll down to **Step #5: Configure Preprocessors** section (Line 252), the listed preprocessor. Do nothing in IDS mode, but generate errors at runtime
45. Comment all the preprocessors listed in this section by adding # before each preprocessor rule (261-265)

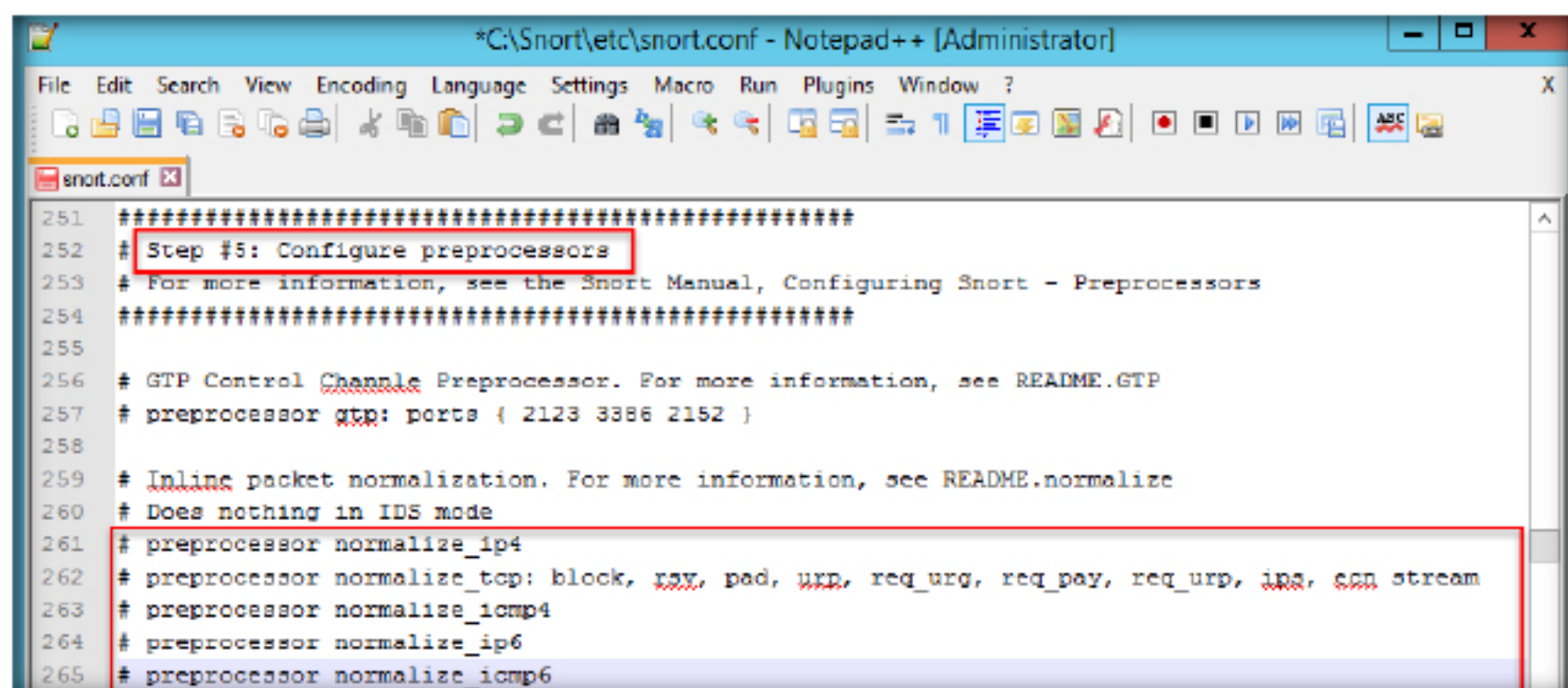


FIGURE 1.16: Configuring Snort.conf File in Notepad++

46. Go to lines **502-506** and remove the backslash at the end of each line

Make sure to grab the rules for the version of Snort you are installing.

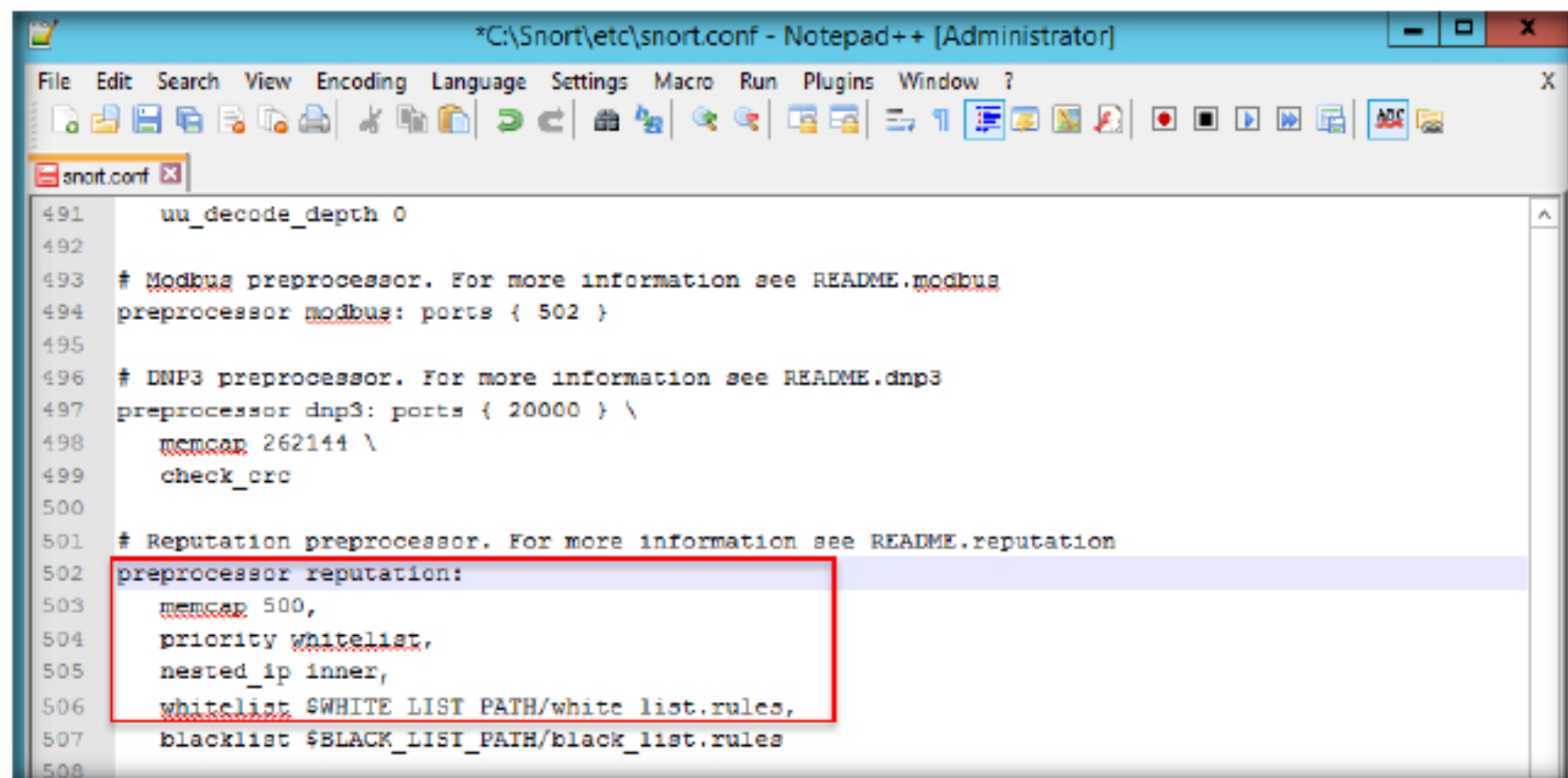


FIGURE 1.17: Configuring Snort.conf File in Notepad++

47. Comment (add #) the lines **502-507**, as shown in the screenshot

Frag3 is intended as a replacement for the frag2 defragmentation module and was designed with the following goals:

1. Faster execution than frag2 with less complex data management.
2. Target-based host modeling anti-evasion techniques.

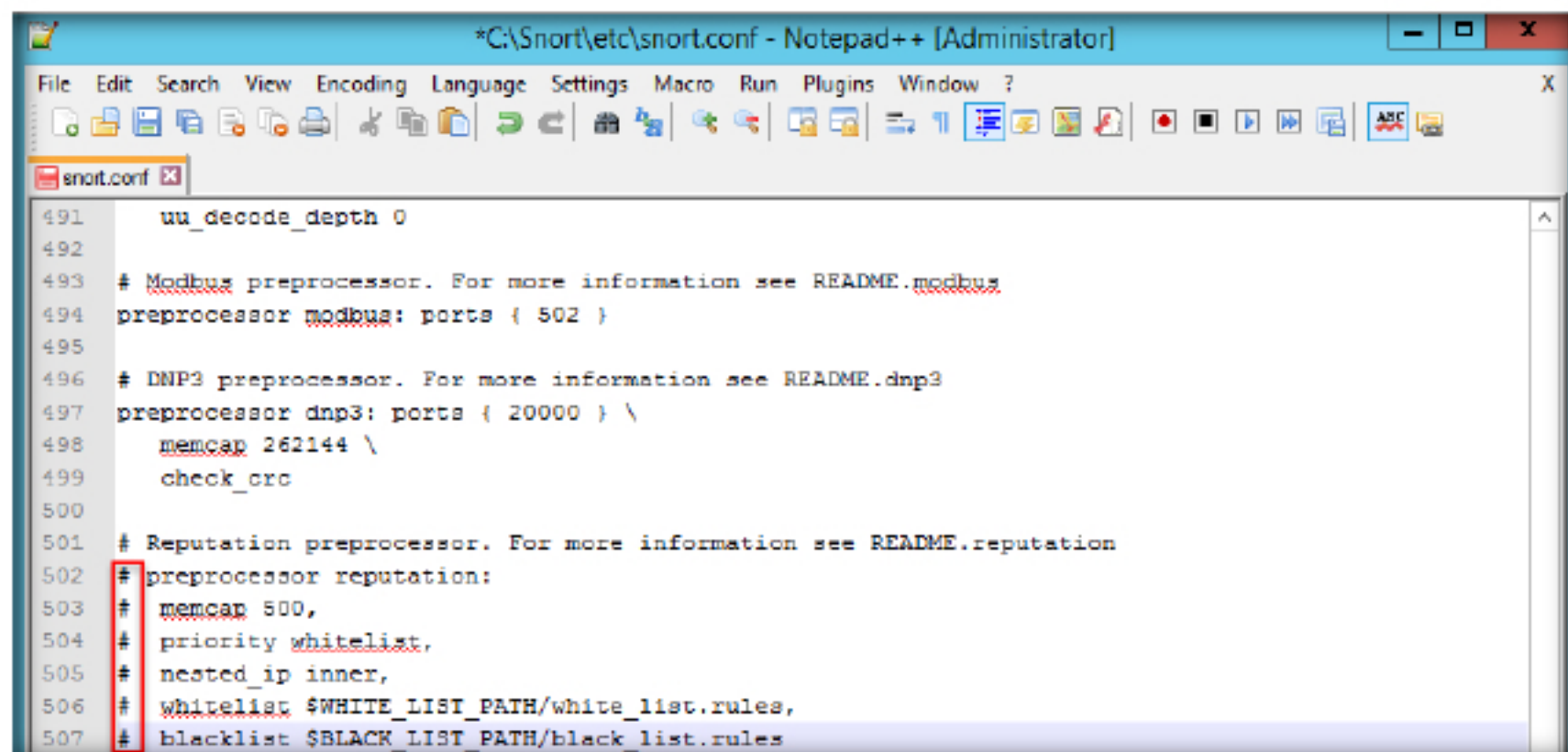


FIGURE 1.18: Configuring Snort.conf File in Notepad++

48. Scroll down to **Step #6: Configure output plugins** (Line 510). In this step, provide the location of the **classification.config** and **reference.config** files

49. These two files are in **C:\Snort\etc**. Provide this location of files in configure output plugins (in Lines 529 and 530) i.e., **C:\Snort\etc\classification.config** and **C:\Snort\etc\reference.config**

Many configuration and command line options of Snort can be specified in the configuration file. Format: config <directive> [<value>].

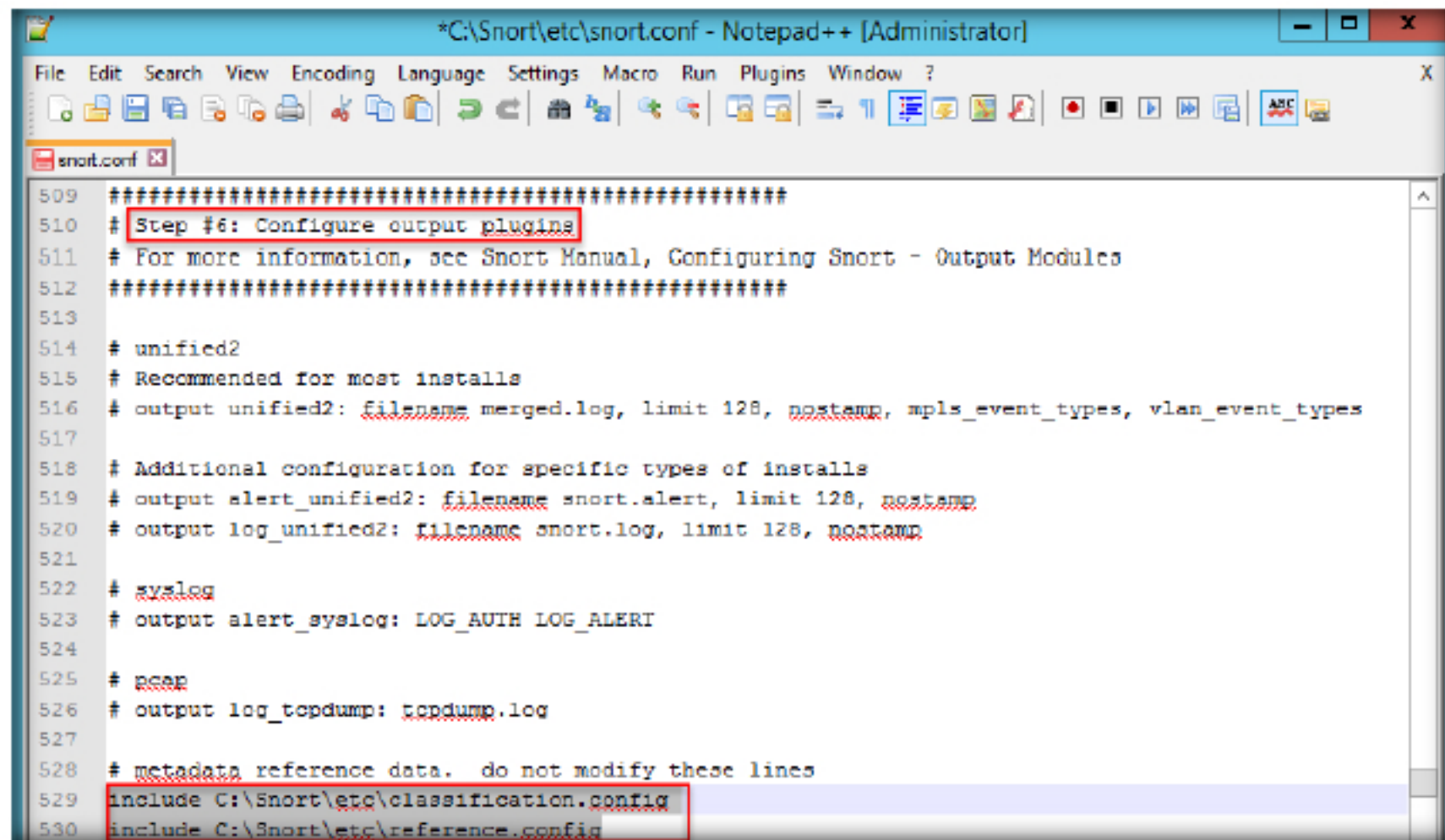


FIGURE 1.19: Configuring Snort.conf File in Notepad++

50. In **Step #6**, add the line (531) **output alert_fast: alerts.ids**, for Snort to dump all logs in the **alerts.ids** file

Note: 'ipvar's are enabled only with IPv6 support. Without IPv6 support, use a regular 'var.'

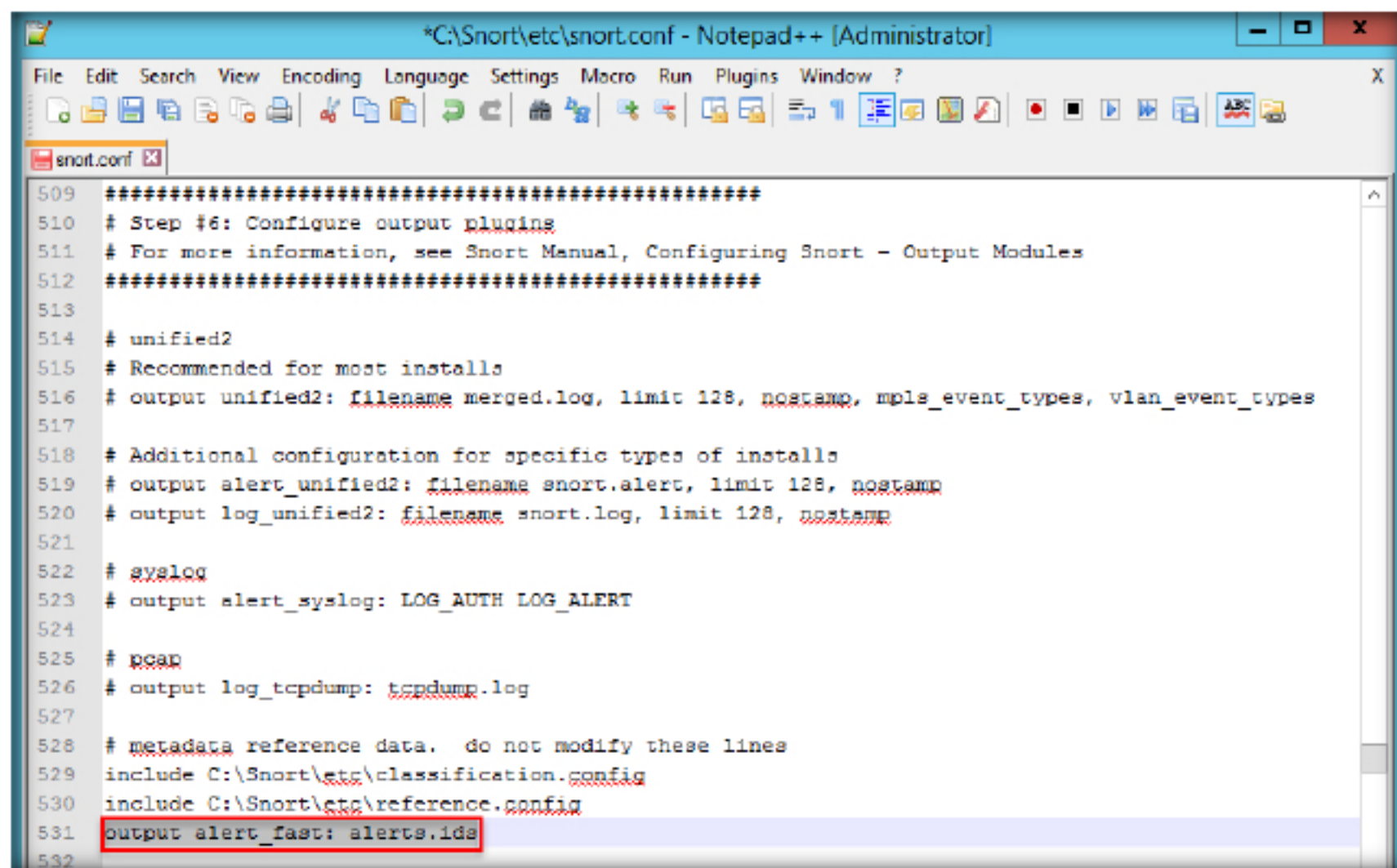



FIGURE 1.20: Configuring Snort.conf File in Notepad++

51. Save the **snort.conf** file
52. Before running Snort, you need to enable detection rules in the Snort rules file. For this lab, we have enabled the ICMP rule so that Snort can detect any host discovery ping probes to the system running Snort

53. Navigate to **C:\Snort\rules** and open the **icmp-info.rules** file with Notepad ++
54. Type **alert icmp \$EXTERNAL_NET any -> \$HOME_NET 10.10.10.12 (msg:"ICMP-INFO PING"; icode:0; itype:8; reference:arachnids,135; reference:cve,1999-0265; classtype:bad-unknown; sid:472; rev:7;)** in line 21, and save it

 To run Snort as a daemon, add -D switch to any combination. Notice that if you want to be able to restart Snort by sending a SIGHUP signal to the daemon, specify the full path to the Snort binary when you start it, for example:

```
/usr/local/bin/snort -d -h 192.x.x.x/24 \-l /var/log/snortlogs -c /usr/local/etc/snort.conf -s -D
```

Note: The IP address (10.10.10.12) mentioned in \$HOME_NET may vary in your lab environment

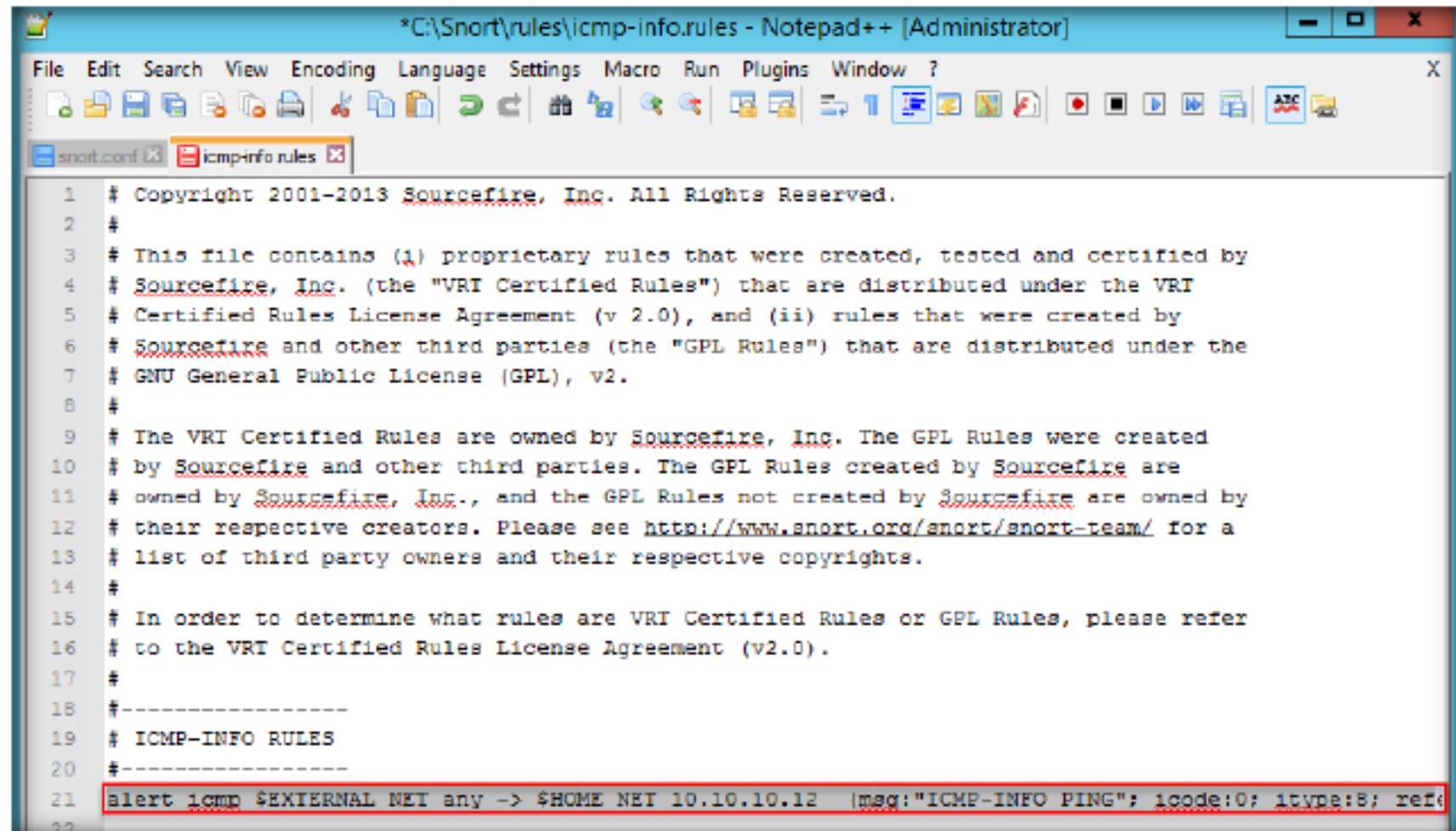


FIGURE 1.21: Configuring icmp-info.rules File in Notepad++

TASK 4

Validate Configurations

55. Minimize all the windows that were open and navigate to **C:\Snort** and select **bin** folder, and press **Shift + right-click**, and then click **Open command window here** from the context menu to open it in the command prompt
56. Type **snort -iX -A console -c C:\Snort\etc\snort.conf -l C:\Snort\log -K ascii** and press **Enter** to start Snort (replace **X** with your device index number; in this lab: **X** is 1)

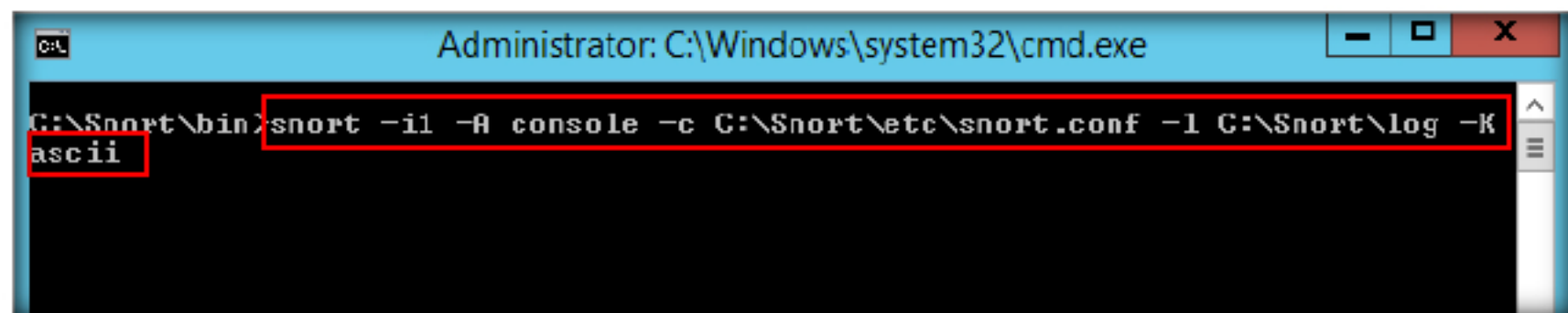



FIGURE 1.22: Command to activate Snort and save the stored log files

57. If you receive a **fatal error**, you should first **verify** that you have typed all modifications correctly into the **snort.conf** file, and then search through the file for **entries** matching your fatal error message
58. If you receive an error stating “**Could not create the registry key,**” then run the command prompt as an **Administrator**

59. Snort starts running in IDS mode. It first initializes output plug-ins, preprocessors, plug-ins, load dynamic preprocessors libraries, rule chains of Snort, and then logs all signatures
60. If you enter all the command information correctly, you receive a comment stating **Commencing packet processing <pid=xxxx>** (the value of xxxx may be any number; in this lab, it is 2616), as shown in the screenshot

 Preprocessors are loaded and configured using the 'preprocessor' keyword. The format of the preprocessor directive in the Snort rules file is: `preprocessor <name>: <options>`.

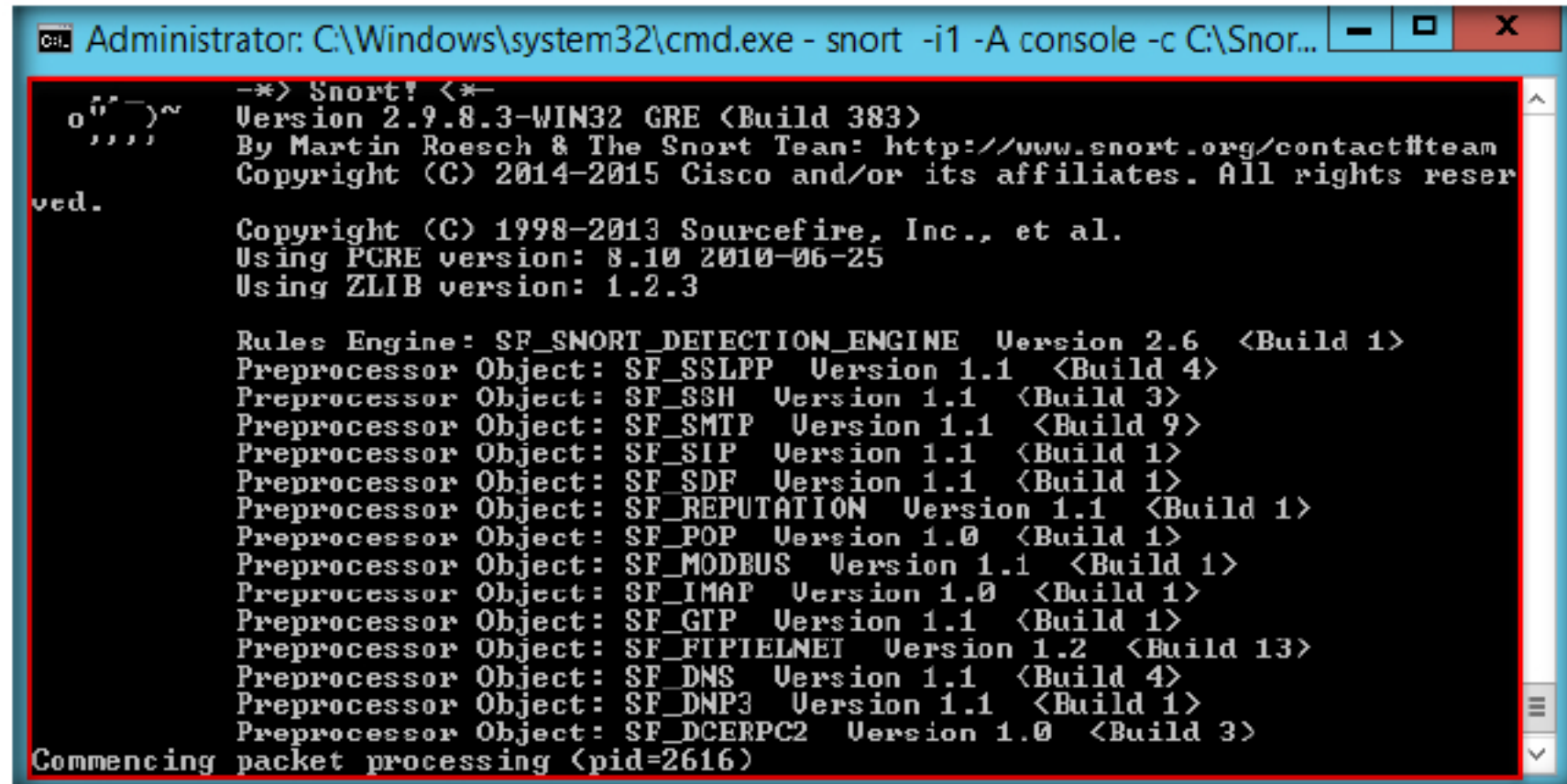



FIGURE 1.23: Initializing Snort Rule Chains Window

61. After initializing the interface and logged signatures, Snort starts and waits for an attack and triggers an alert when attacks occur on the machine
62. Leave the Snort command prompt running
63. Attack your own machine, and check whether Snort detects it or not
64. Launch the **Windows 10** virtual machine and login
65. Open the command prompt and issue the command `ping 10.10.10.12 -t` from the **Windows 10** machine

Note: **10.10.10.12** is the IP address of the **Windows Server 2012** machine. This IP address may differ in your lab environment

 **TASK 5**

Ping the host Machine

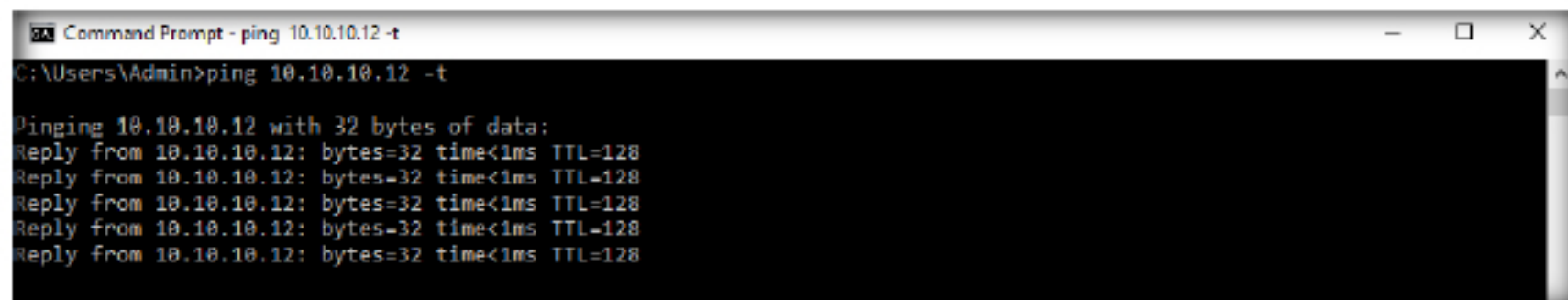


FIGURE 1.24: Pinging the target machine from host machine

66. Switch back to the Windows Server 2012 machine. Observe that Snort triggers an alarm, as shown in the screenshot

Run Snort as a Daemon syntax:
 /usr/local/bin/snort -d -h 192.x.x.x/24 -l /var/log/snortlogs -c /usr/local/etc/snort.conf -s -D.

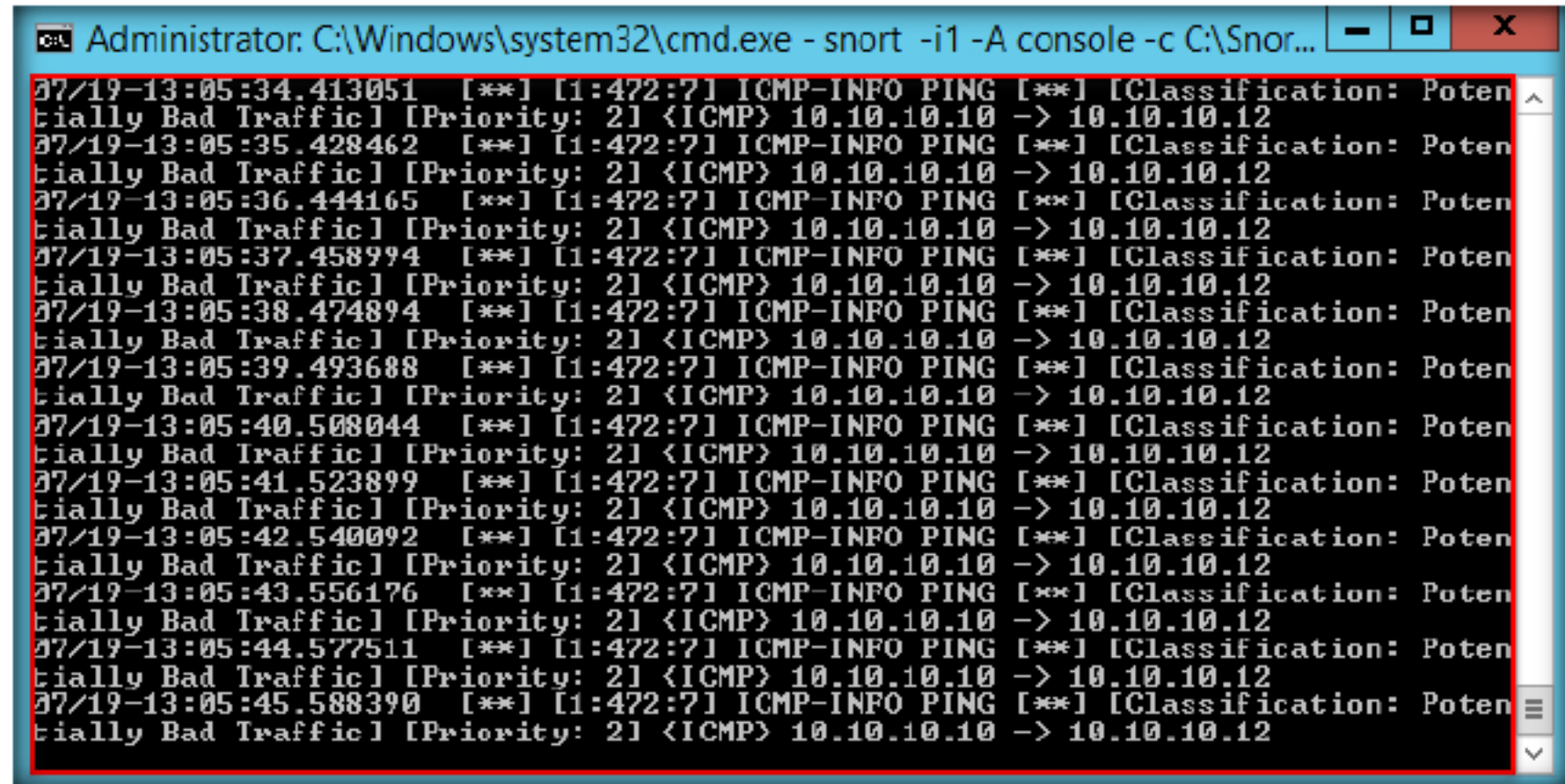


FIGURE 1.25: Snort Alerts.ids Window Listing Snort Alert

67. Press **Ctrl+C** to stop Snort. Snort exits.

When Snort is run as a Daemon, the daemon creates a PID file in the log directory.

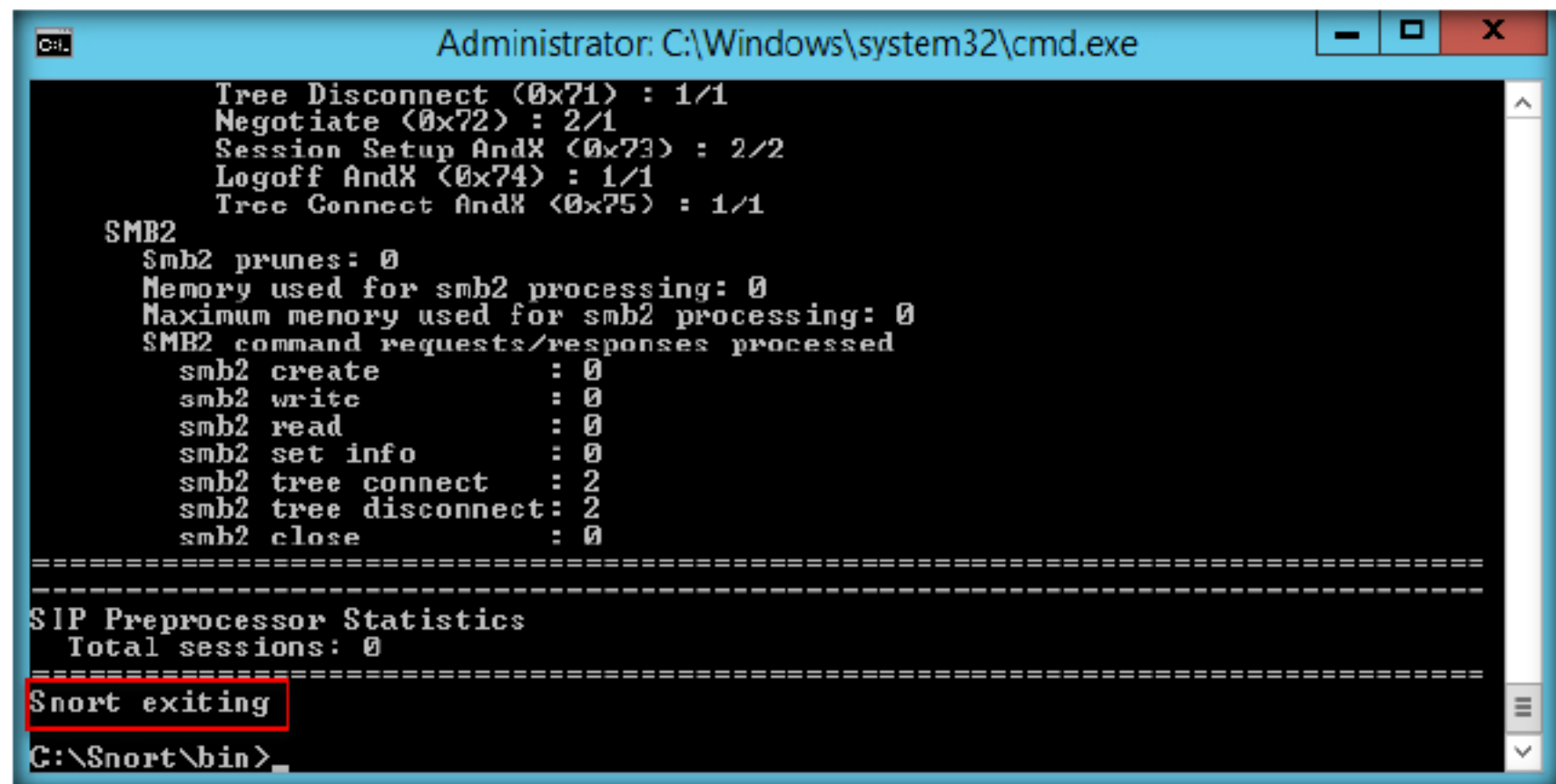


FIGURE 1.26: Exiting snort by pressing Ctrl+C

68. Go to the **C:\Snort\log\10.10.10.10** folder, and open the **ICMP_ECHO.ids** file with Notepad++. You see that all the log entries are saved in the **ICMP_ECHO.ids** file

Note: The folder name 10.10.10.10 might vary in your lab environment, depending on the IP address of **Windows 10** machine

Three types of variables may be defined in Snort:

- Var
- Portvar
- ipvar

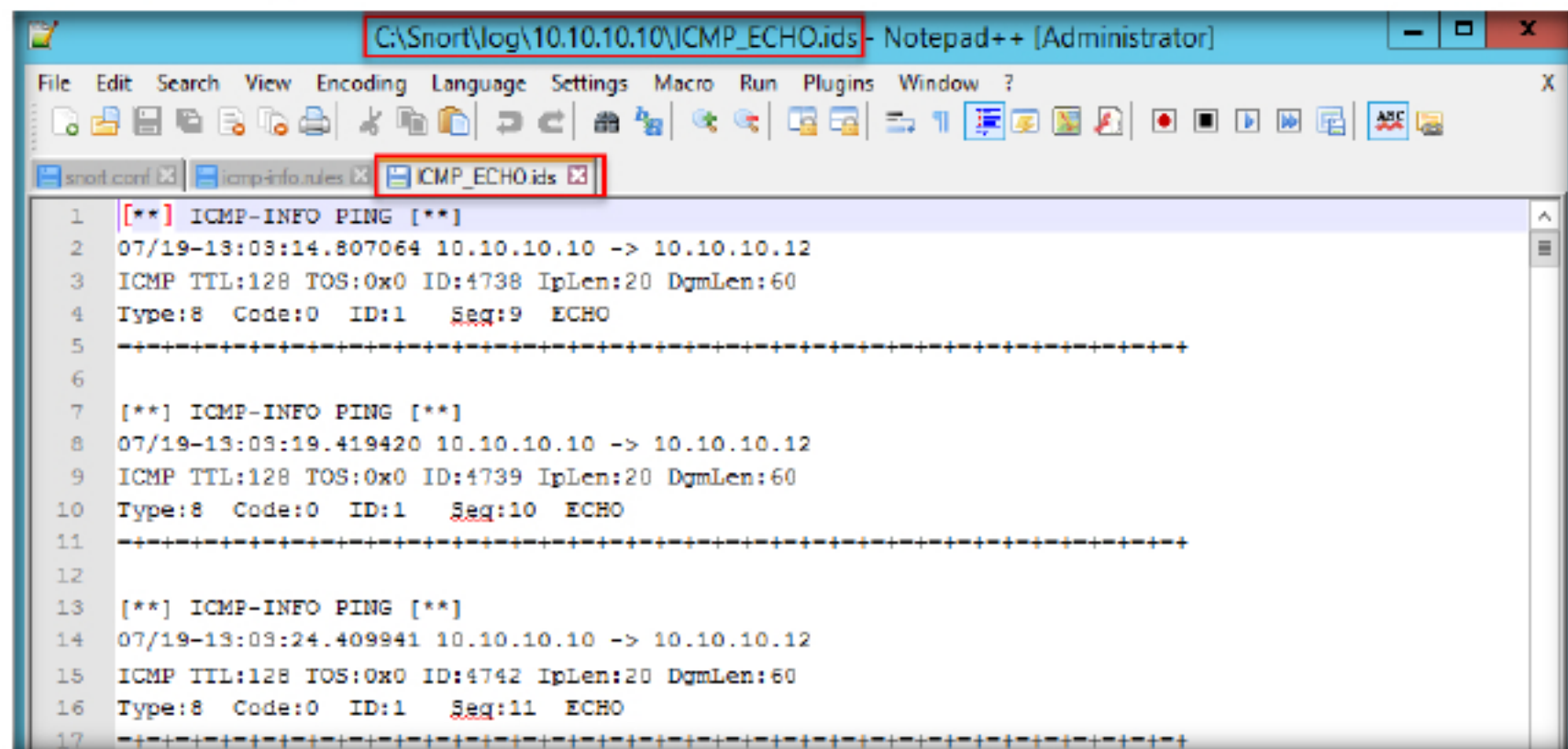


FIGURE 1.27: Saved Snort log file

69. This means, whenever an intruder attempts to connect or communicate with the machine, Snort immediately triggers an alarm

70. So, you can become alert and take certain security measures to break the communication with the organization network

Lab Analysis

Analyze and document the results of the lab exercise. Give your opinion on your target’s security posture and exposure through free public information.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS ABOUT THIS LAB.


Internet Connection Required	
<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> No
Platform Supported	
<input checked="" type="checkbox"/> Classroom	<input checked="" type="checkbox"/> iLabs





Detecting Intruders and Worms using KFSensor Honeypot IDS


KFSensor is a Windows-based honeypot IDS.

ICON KEY

 Valuable information

 Test your knowledge

 Web exercise

 Workbook review

Lab Scenario

Intrusion detection plays a key role in ensuring the integrity of a system's security. Network Intrusion Detection Systems (NIDSs) have long been the best method for identifying assaults. KFSensor is an NIDS that is easy to install and configure. No special hardware is required, and its efficient design enables it to run even on low-specification Windows machines.

As a network administrator, you must possess sound knowledge of network IPSs and IDSs, identify network malicious activity and log information, and stop or block malicious network activity.

Lab Objectives

The objective of this lab is to demonstrate the use and configuration of KFSensor Honeypot IDS.


In this lab, you will:

- Detect hackers and worms in a network
- Provide network security

Lab Environment

To complete this lab, you will need:

- KF Sensor is located at **Z:\CND-Tools\CND Module 08 Secure IDS Configuration and Management\Honey Pot and Padded Cell System Tools\KFSensor**
- KF Sensor installed in **Windows Server 2012**
- MegaPing located at **Z:\CND-Tools\CND Module 08 Secure IDS Configuration and Management\Honey Pot and Padded Cell System Tools\MegaPing**

 You can also download KFSensor from <http://www.keyfocus.net>

- MegaPing is installed in **Windows 10**
- If you have decided to download the latest of version of these tools, then the screen shots may differ
- Administrative privileges to configure settings and run tools

Lab Duration

Time: 25 Minutes

Overview of the Lab

KFSensor contains a powerful internet daemon service that is built to handle multiple ports and IP addresses. It is written to resist denial of service and buffer overflow attacks.

Building on this flexibility KFSensor can respond to connections in a variety of ways, from simple port listening and basic services (such as echo), to complex simulations of standard system services. For the HTTP protocol KFSensor accurately simulates the way Microsoft's web server (IIS) responds to both valid and invalid requests. As well as being able to host a website it also handles complexities such as range requests and client side cache negotiations. This makes it extremely difficult for an attacker to fingerprint, or identify KFSensor as a honeypot.

Lab Tasks



TASK 1

Configure KFSensor


Note: Ensure that WinPcap is installed before running this lab. Before starting this lab make sure that Windows Firewall is turned off in the Windows Server 2012 machine for demonstration purposes. After the completion of this exercise you can turn on the Windows Firewall.

To turn off Windows Firewall navigate to Control Panel, in the Control Panel window click on the Windows Firewall, and in the Windows Firewall window click the Turn Windows Firewall on/off link from the left hand-side. Customize Settings window appears, select **Turn off Windows Firewall (not recommended)** radio button for both the profiles and click **OK**.

1. Log into **Windows Server 2012** virtual machine.
2. Navigate to **Z:\CND-Tools\CND Module 08 Secure IDS Configuration and Management\Honey Pot and Padded Cell System Tools\KFSensor**, double-click **kfsens40.msi** and follow the wizard driven installation steps to install KFSensor.

Note: After installation if it prompts to reboot the system. **Reboot** the virtual machine.

- Once the installation is finished make sure that the Launch KFSensor option is checked and clicks **Finish**. To launch KFSensor automatically.

 The Set up Wizard is used to perform the initial configuration of KFSensor.

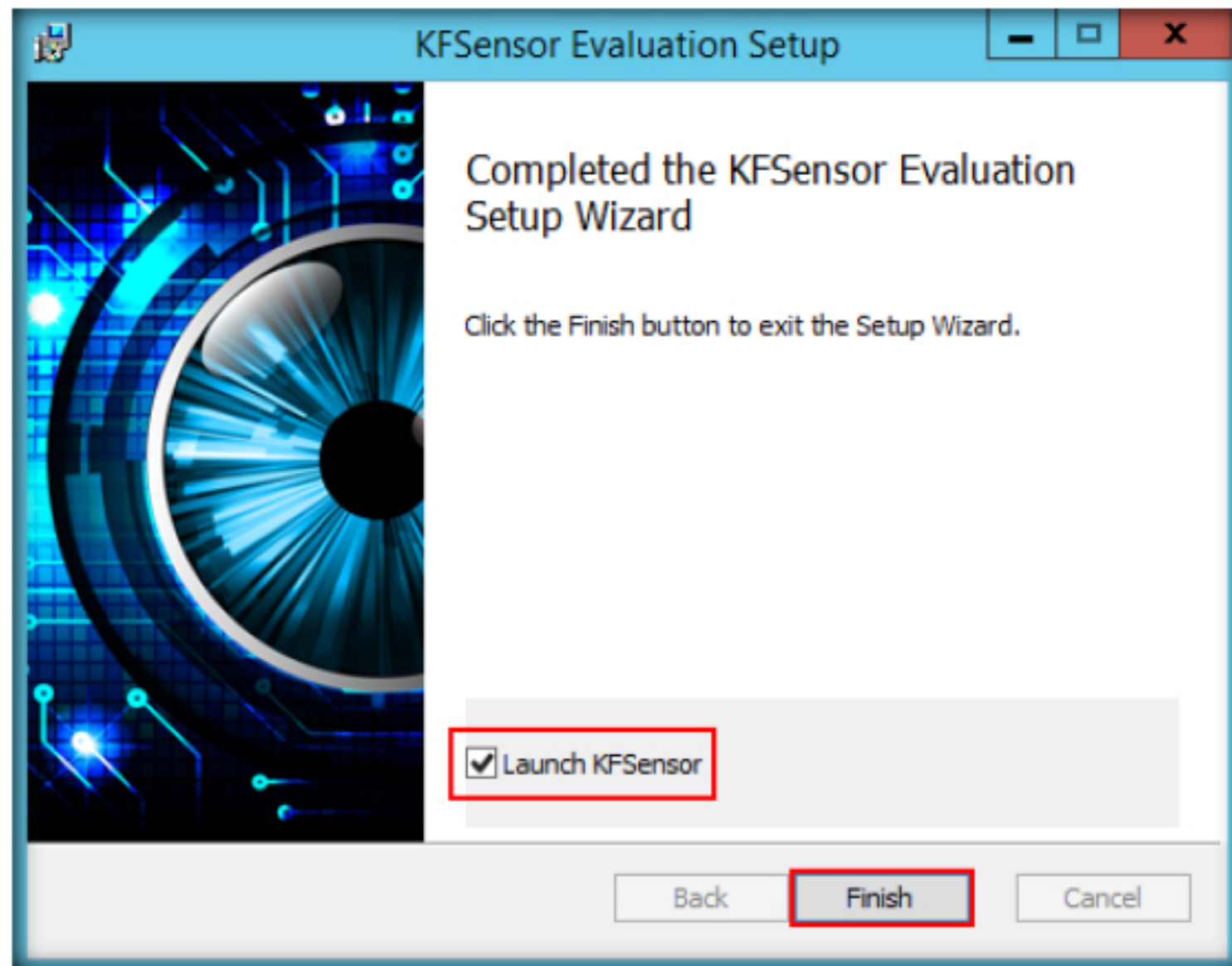



FIGURE 2.1: Launching KFSensor

- On first launch of **KFSensor**, the **setup wizard** appears; click **Next**

 The KFSensor Monitor is a module that provides the user interface to the KFSensor system. With it you can configure the KFSensor Server and examine the events that it generates.

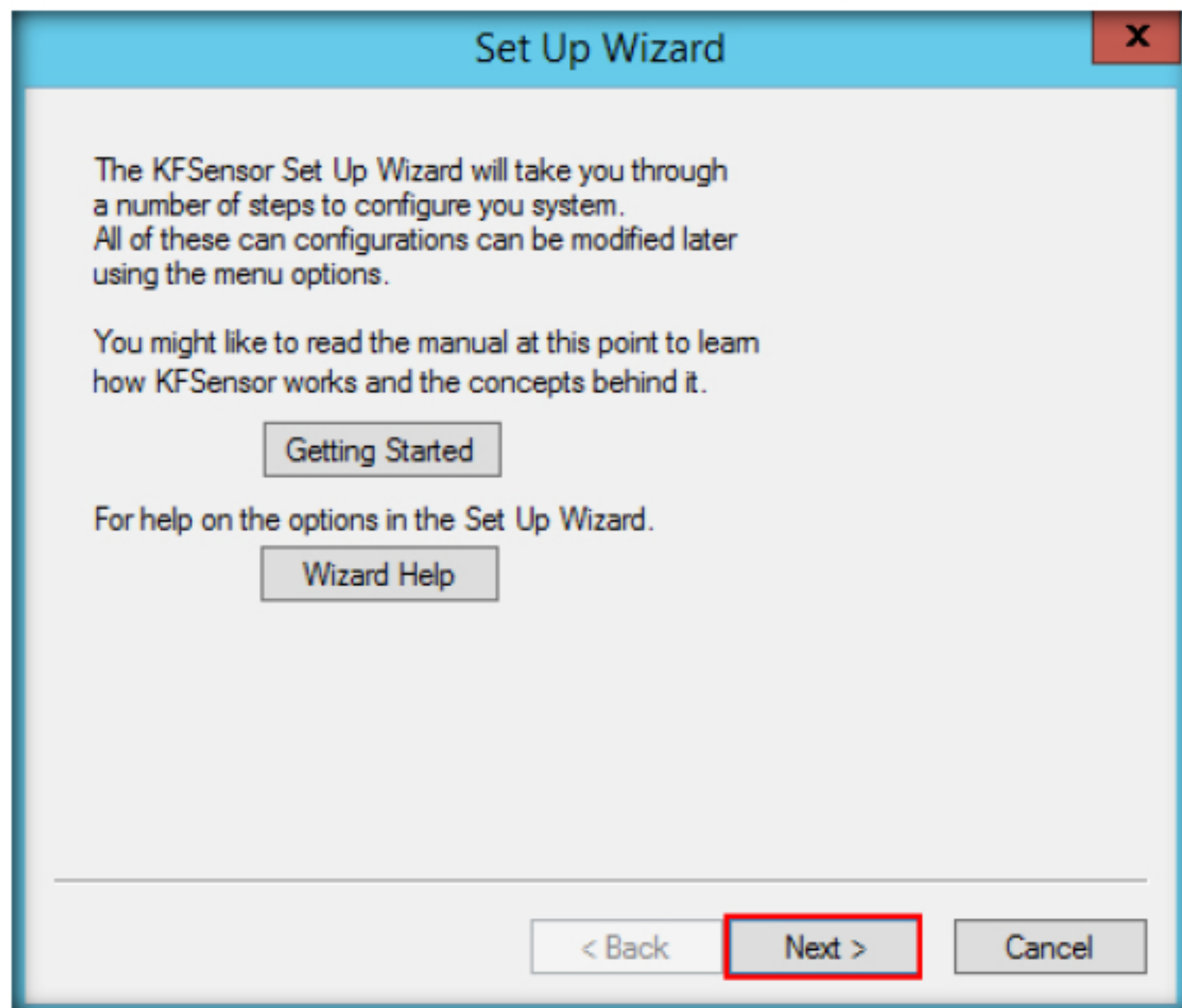



FIGURE 2.2: KFSensor Set Up wizard

- Uncheck all the **ports with all active native services** to include, and click **Next**

 A systems service is a special type of application that Windows runs in the background and is similar in concept to a UNIX daemon.

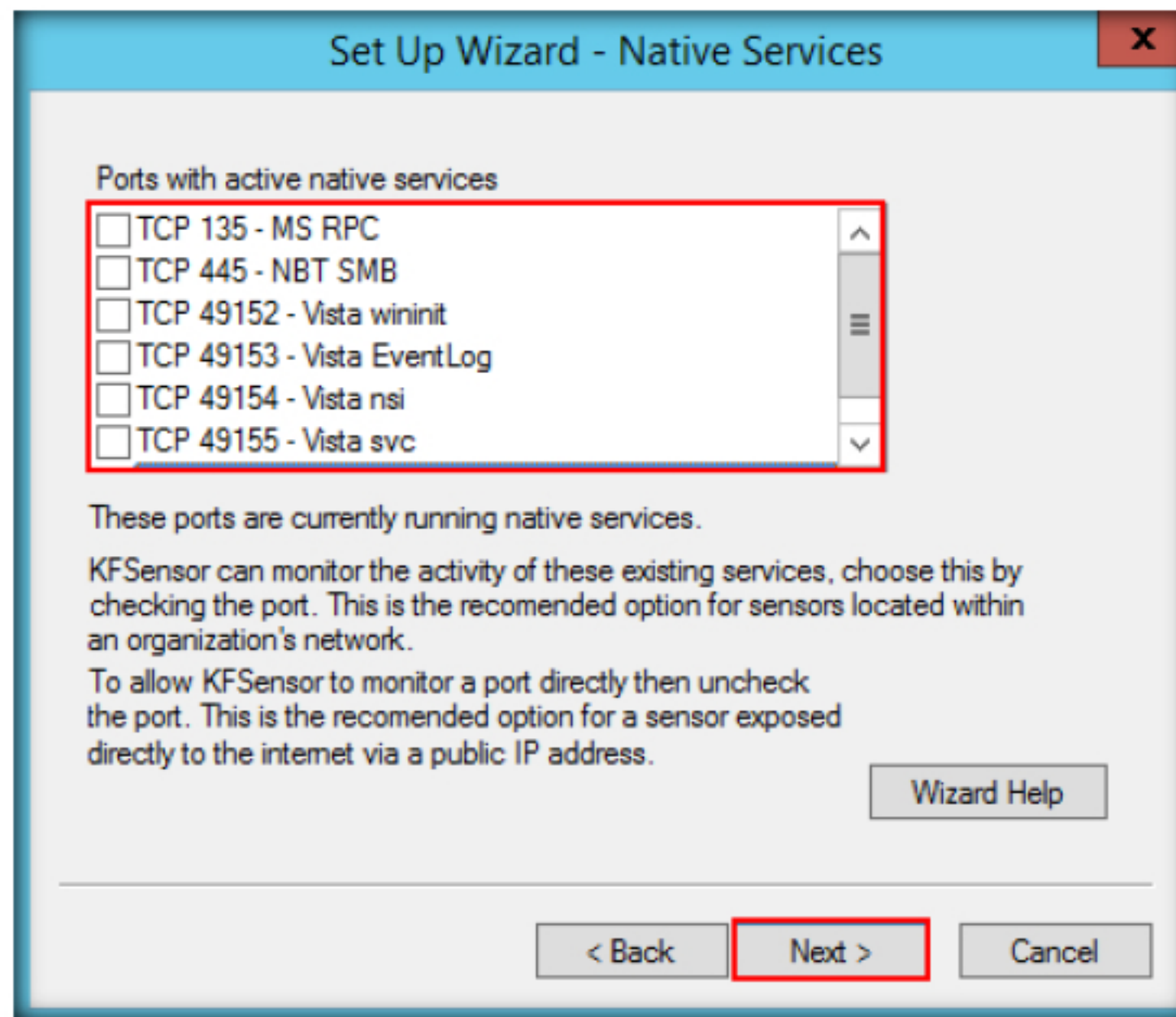



FIGURE 2.3: KFSensor Native Services

- If you want to send **KFSensor alerts** by email, specify email address details, or leave the fields empty and click **Next**

 To set up common ports KFSensor has a set of pre-defined listen definitions. They are:

- Windows Workstation
- Windows Server
- Windows Internet Services
- Windows Applications
- Linux (services not usually in Windows)
- Trojans and worms

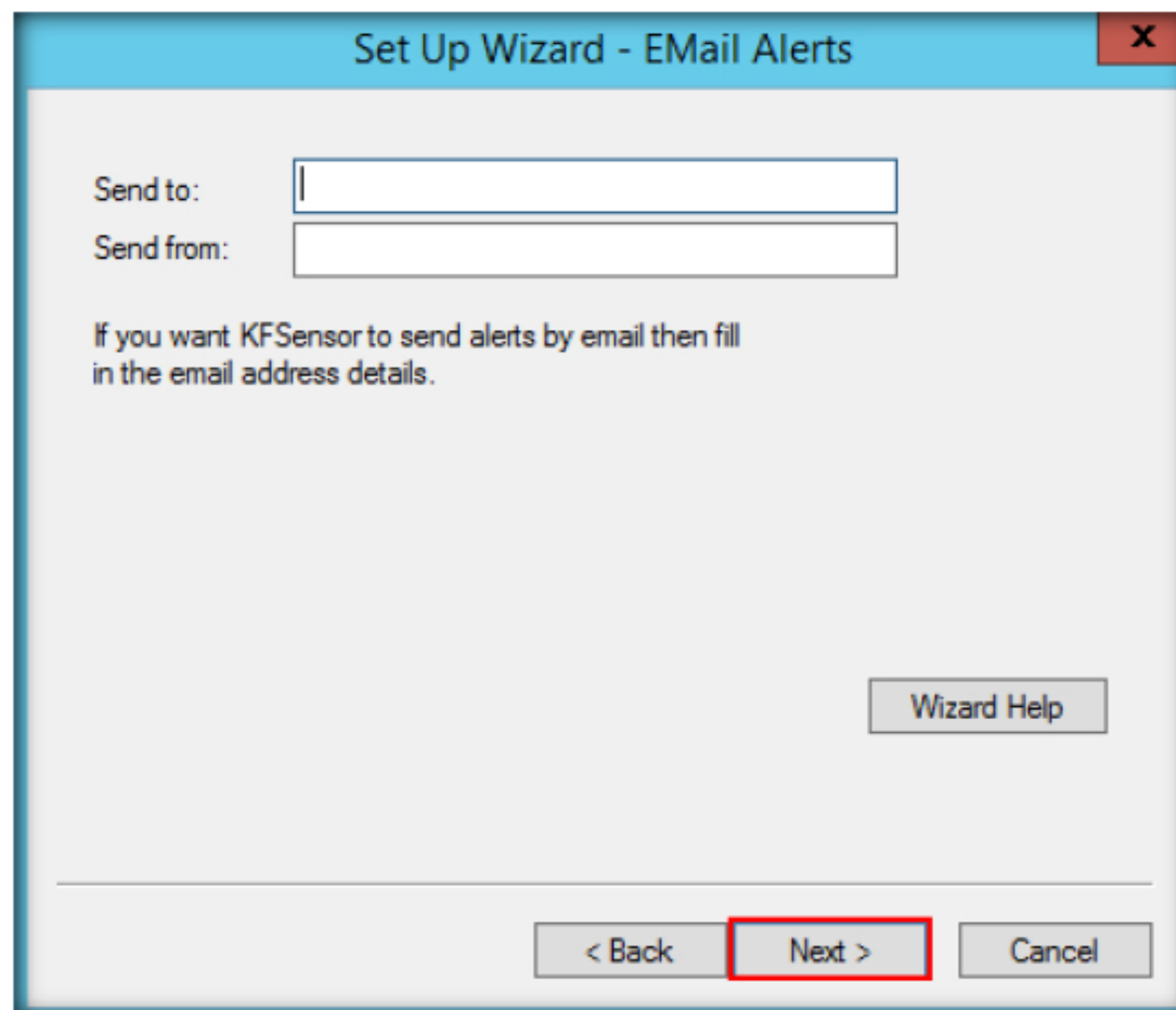


FIGURE 2.4: KFSensor Email Alerts

- Click **Finish** to complete the setup

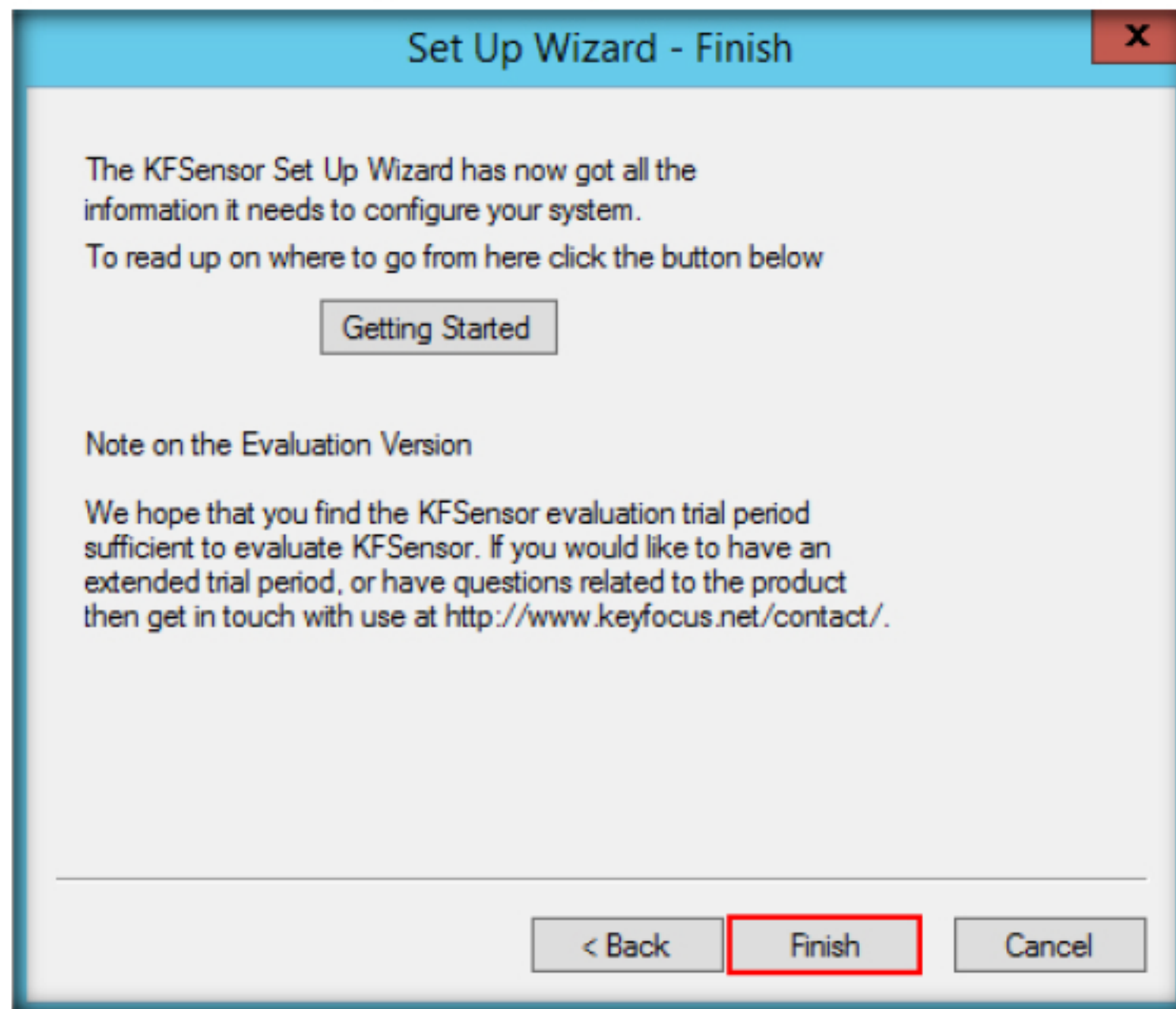



FIGURE 2.5: KFSensor Set Up Finished

 The Ports View is displayed on the left panel of the main window. It comprises of a tree structure that displays the name and status of the KFSensor Server and the ports on which it is listening.

- The **KFSensor** main window appears. It displays a list of **ID protocols**, **Visitor**, and **Received** automatically when it starts. In the window (shown below), all the nodes in the Left block crossed with **blue lines** are the **ports** currently in use

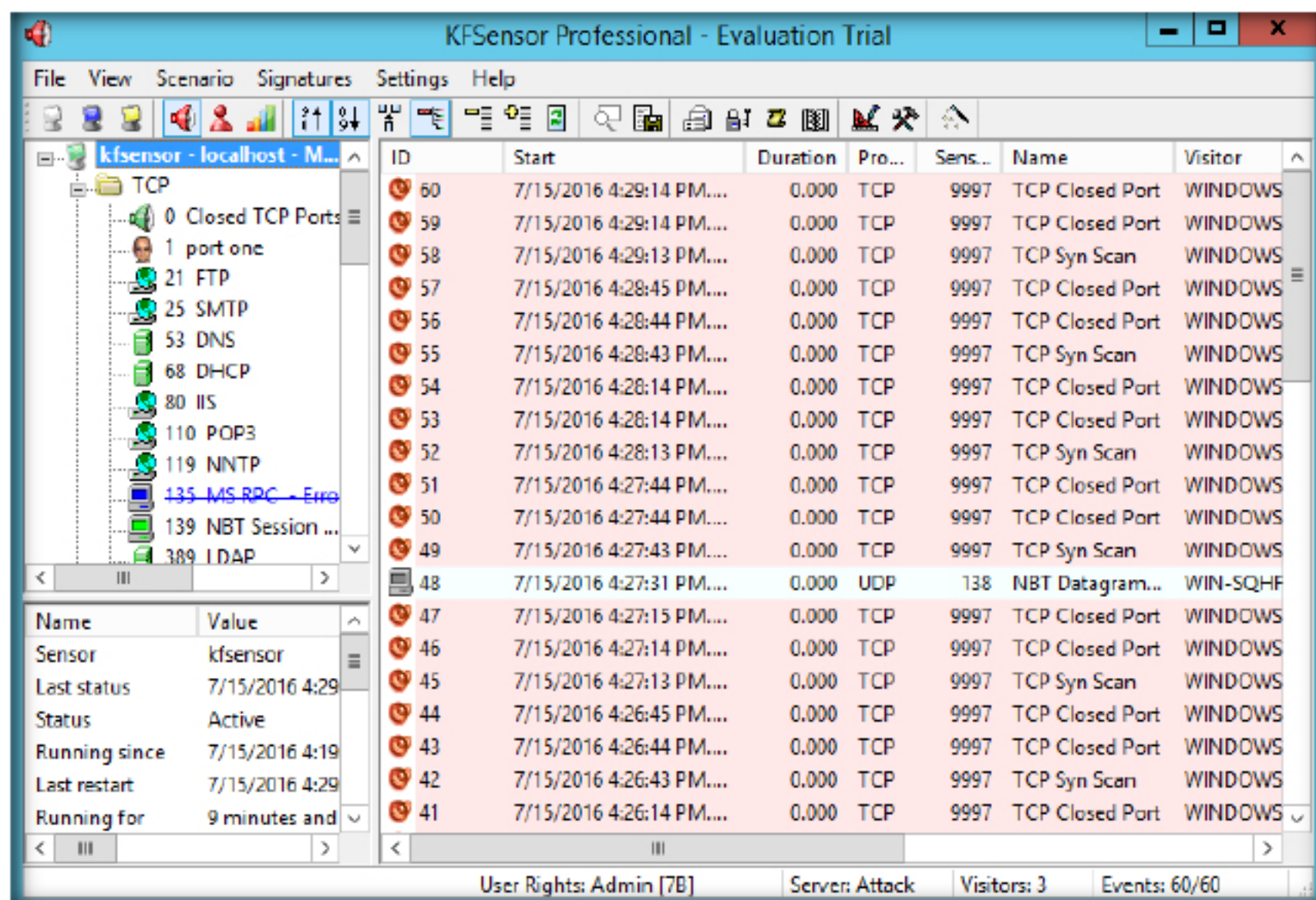



FIGURE 2.6: KFSensor Main Window

9. Launch the **Command Prompt** from the **Apps** screen
10. In the command prompt, type **netstat -an** and press **Enter**
11. This will display a list of **listening ports**

 The top level item is the server. The IP address of the KFSensor Server and the name of the currently active Scenario are displayed. The server icon indicates the state of the server.

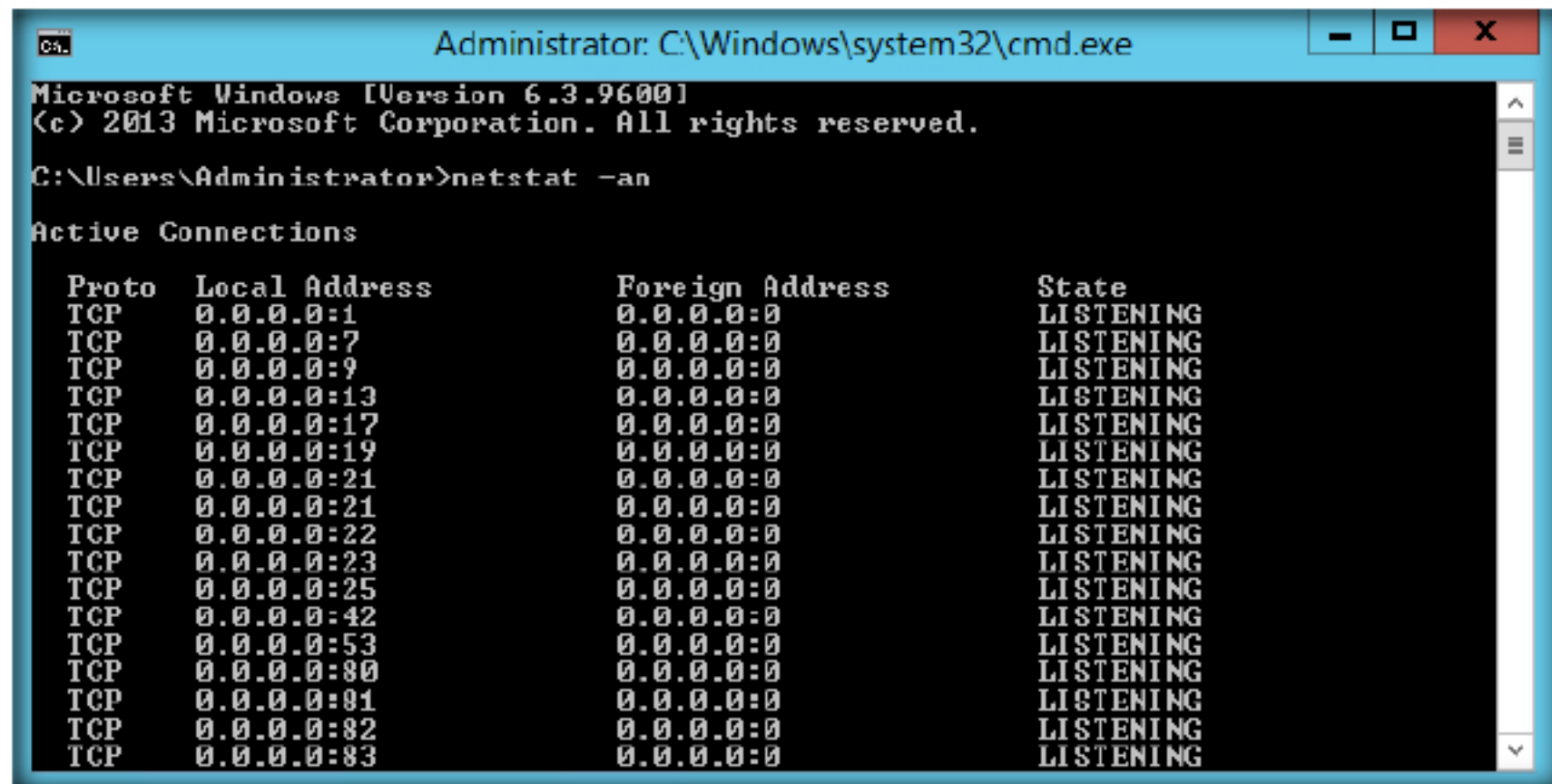



FIGURE 2.7: Command Prompt with netstat -an

12. Log into **Windows 10** virtual machine as a local Administrator.
13. Navigate to **Z:\CND-Tools\CND Module 08 Secure IDS Configuration and Management\Honey Pot and Padded Cell System Tools\MegaPing** and double-click **megaping_setup.exe** and follow the wizard driven installation steps to install **MegaPing**.
14. Once the installation is completed make sure the **Launch the program** option is checked and then click **Finish**, so that MegaPing will launch automatically.

 Each visitor detected by the KFSensor Server is listed. The visitor's IP address and domain name are displayed.

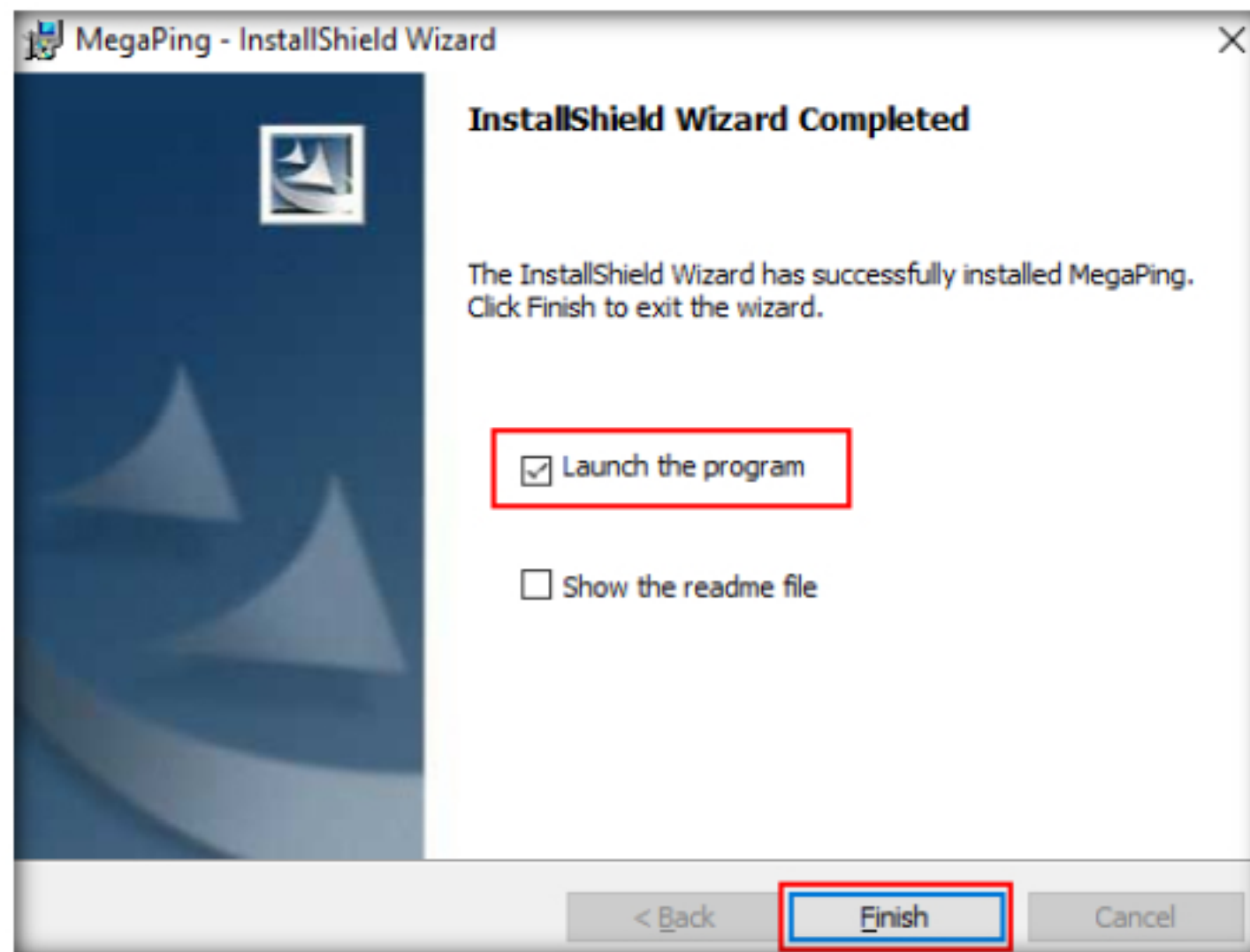


FIGURE 2.8: Launching MegaPing in Windows 10

15. The **About MegaPing** pop-up appears; click **I Agree** to continue

TASK 2
Configure MegaPing

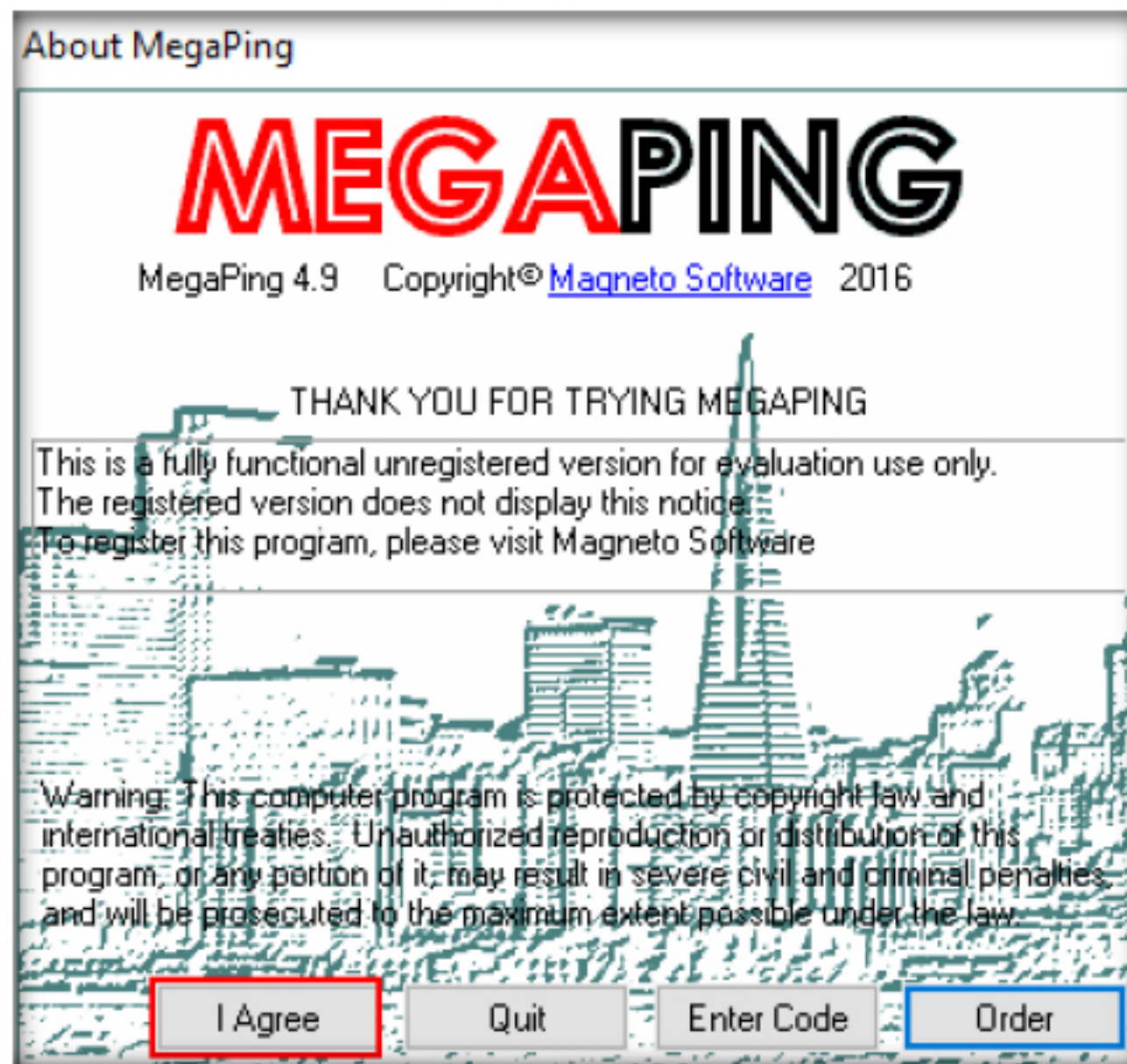


FIGURE 2.9: MegaPing License Agreement

16. **MegaPing** main window appears as shown in the screenshot.

The protocol level of KFSensor is used to group the ports based on their protocol; either TCP or UDP.

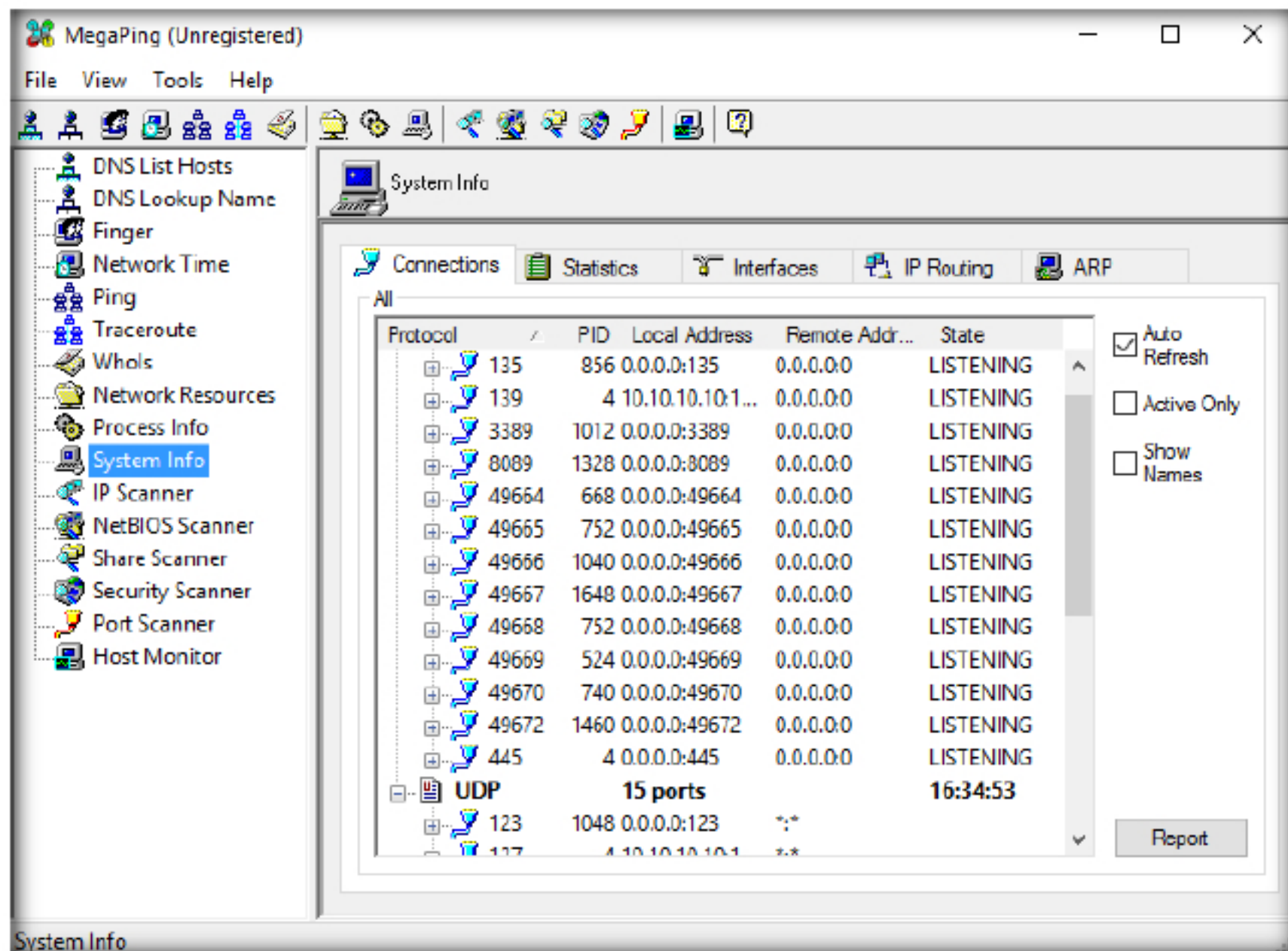


FIGURE 2.10: MegaPing Main Window

TASK 3
Perform Port Scanning

Visitor is obtained by a reverse DNS lookup on the visitor's IP address. An icon is displayed indicating the last time the visitor connected to the server:

17. Select **Port Scanner** in the left pane
18. Enter the IP address of the Windows Server 2012 (10.10.10.12) in the Destination Address List field and click **Add**

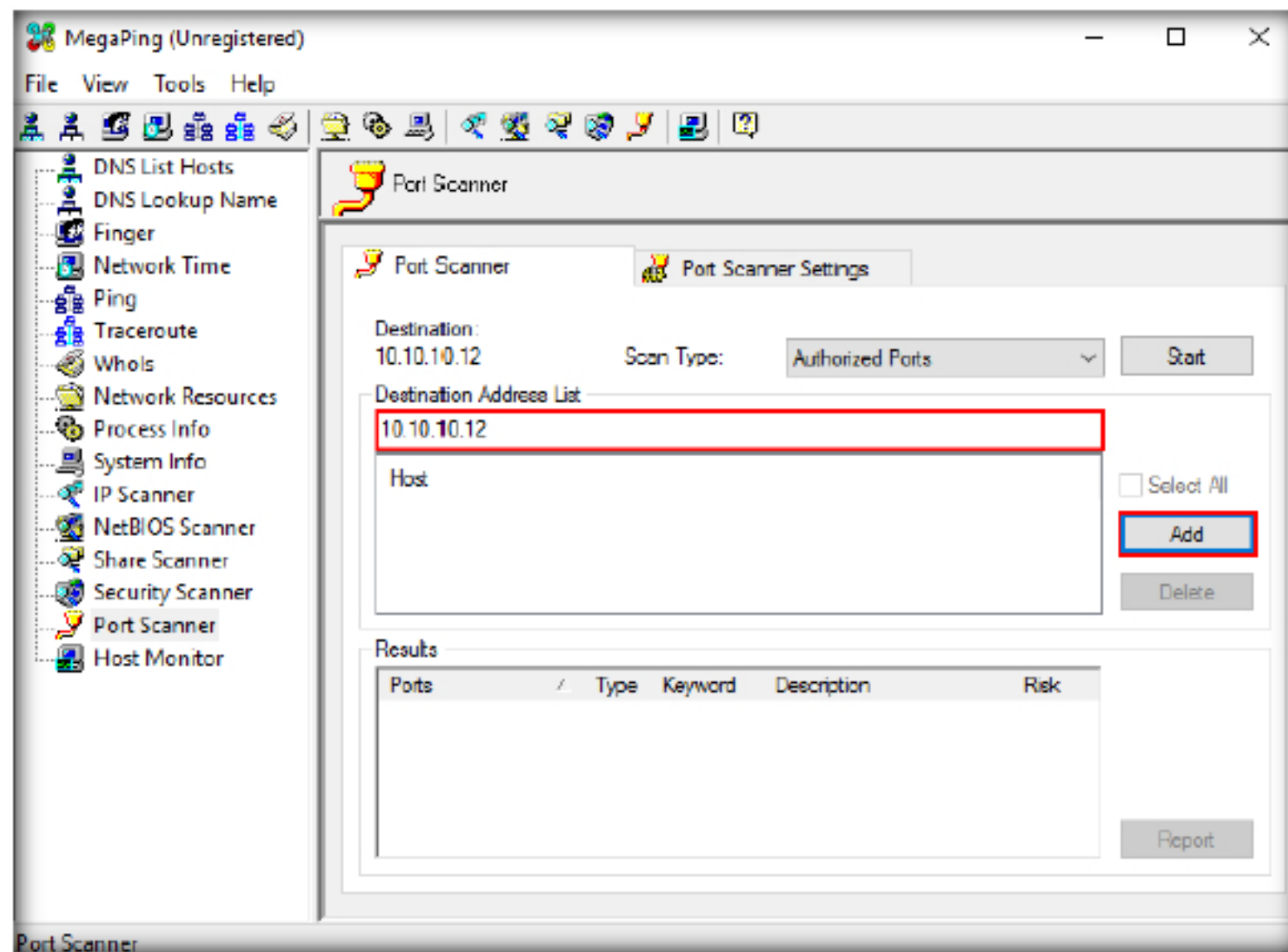


FIGURE 2.11: MegaPing Port Scanner

19. Check the IP address in the Host section, and click the **Start** button to start listening to the traffic on **10.10.10.12**.

Note: This IP address may vary in your lab environment

The Ports View can be displayed by selecting the Ports option from the View menu.

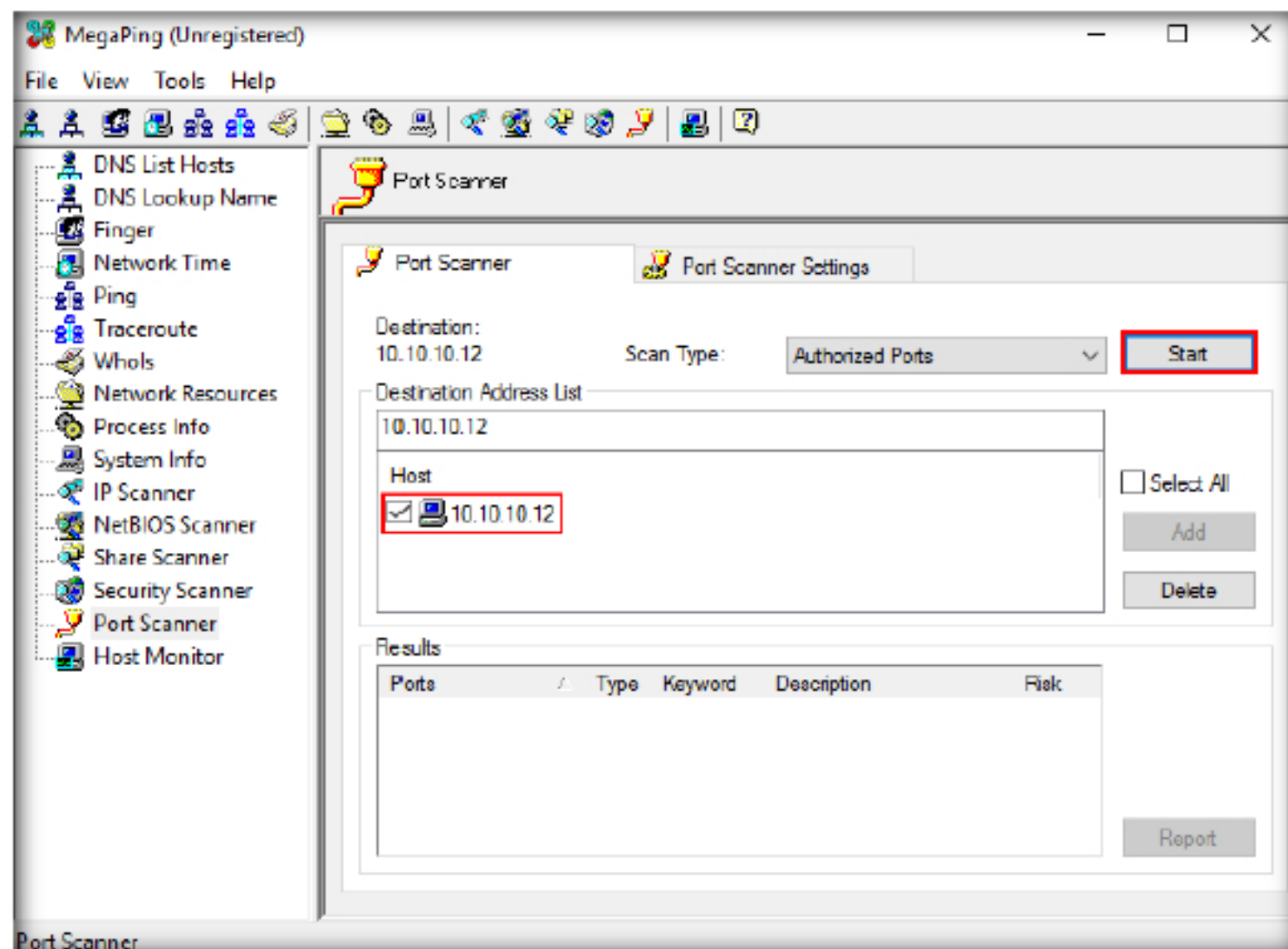


FIGURE 2.12: Beginning the Scan on 10.10.10.12

20. The image below shows the identification of **Telnet** on **port 23**
21. MegaPing begins to scan for open ports and displays a list of ports
22. You can observe **Telnet** on **port 23**, which allows hackers to connect to a remote machine through Telnet

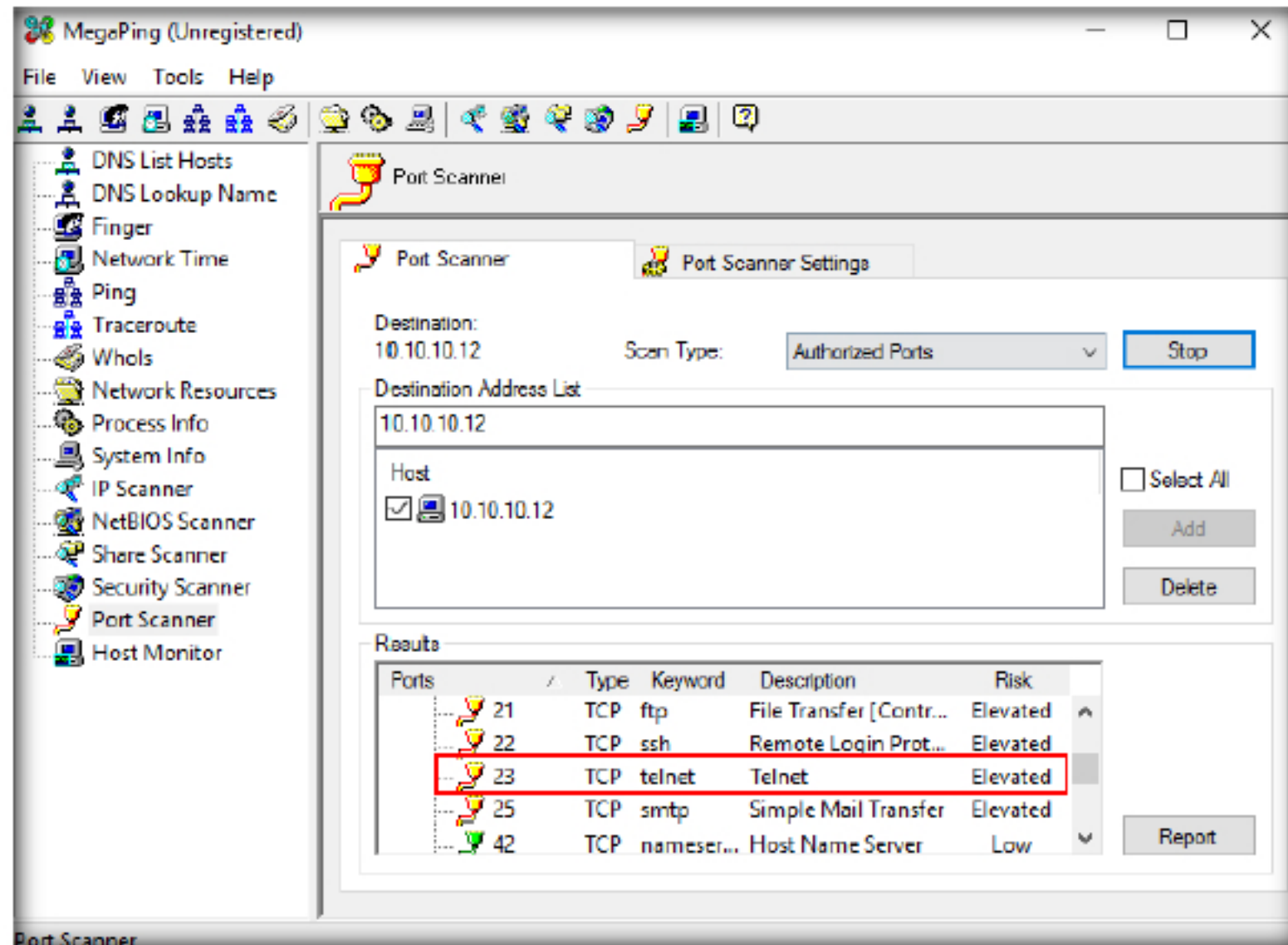


FIGURE 2.13: MegaPing Telnet Port

TASK 4

Analyze the Result

23. Now, switch back to the **Windows server 2012** virtual machine. Observe that KFSensor has detected **port 23** is open.
24. Seeing this port open, you can take proper security measures to close the port, thereby preventing intruders from connecting to this machine from the outside.

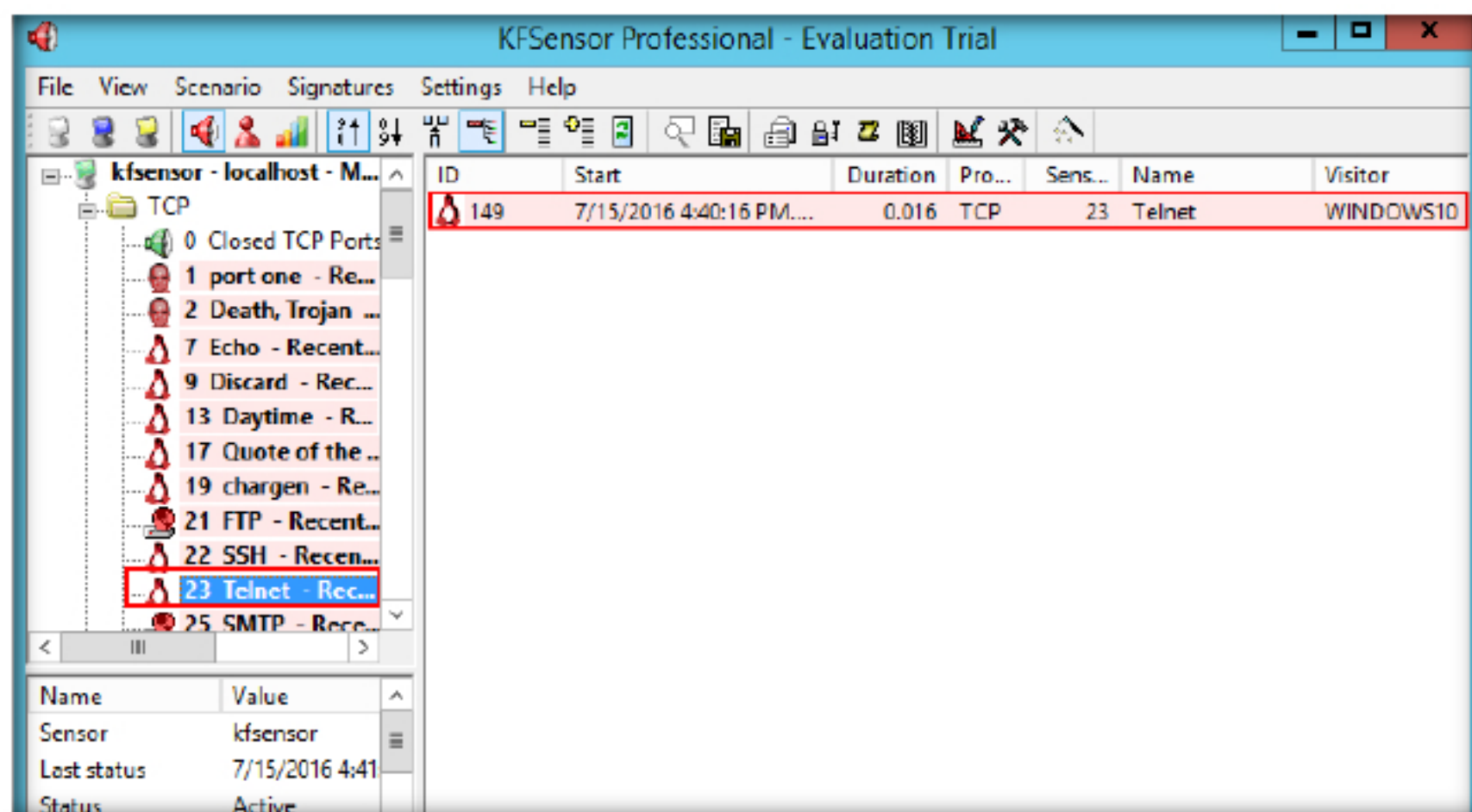


FIGURE 2.14: Telnet port open Alert in KFSensor

25. The image below displays the data of a **Death Trojan** on **port 2**. Seeing this port open, a network administrator can add a firewall rule to block **port 2**, thereby securing the system from being affected by **Death Trojan**

The Visitors View is linked to the Events View and acts as a filter to it. If you select a visitor then only those events related to that visitor will be displayed in the Events View.

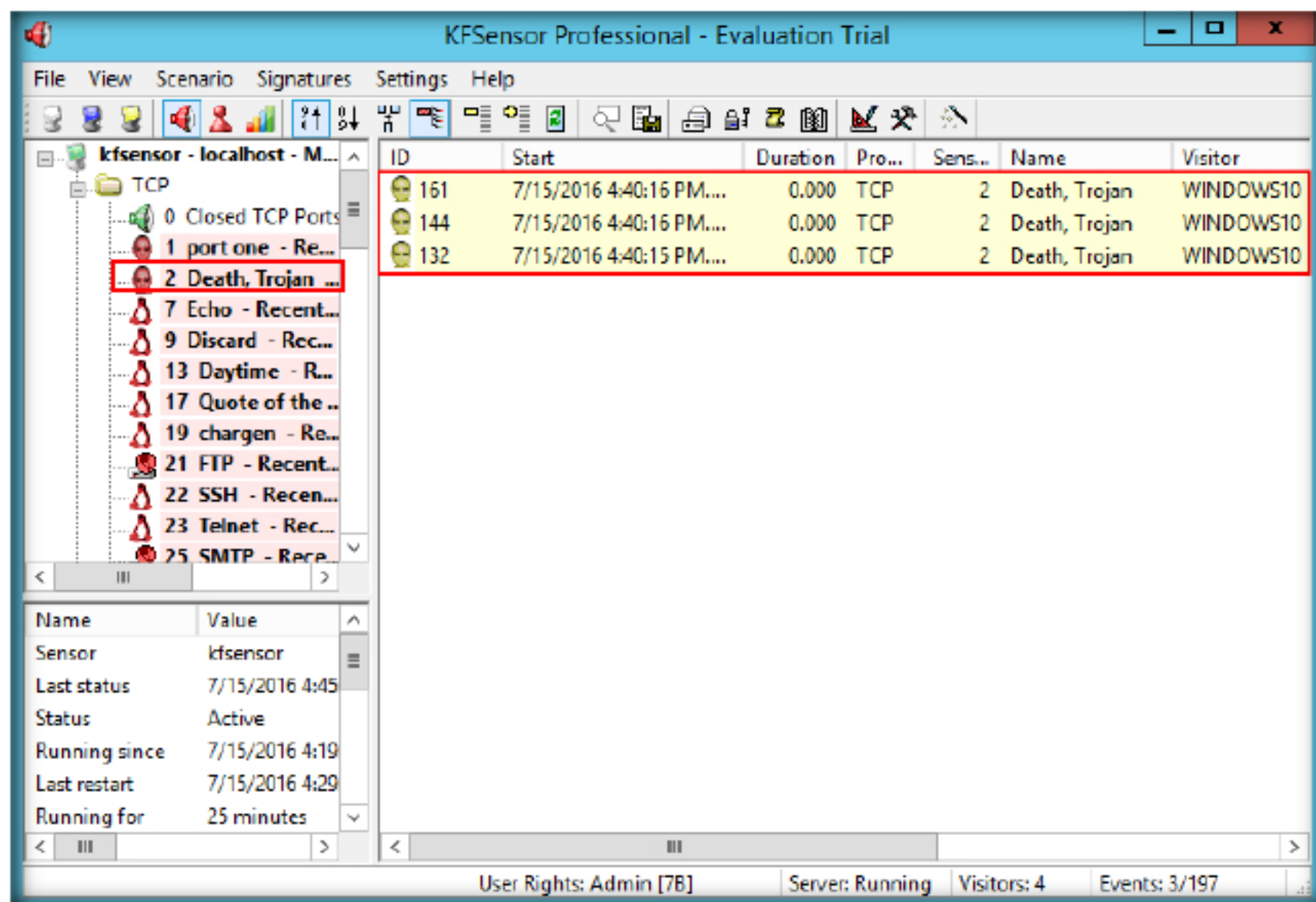


FIGURE 2.15: Death Trojan data on KFSensor

Lab Analysis

Analyze and document the results related to this lab exercise. Provide your opinion of your target’s security posture and exposure.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> No
Platform Supported	
<input checked="" type="checkbox"/> Classroom	<input checked="" type="checkbox"/> iLabs