**Free Open-Source Network Monitoring Tool Performance Comparison Analysis**

By Joshua Larkin

**Abstract**

Network monitoring software is a valuable tool for IT professionals to track the health and safety of the networks that they are responsible for. They do this by sniffing packets from the network. There are many tools for this on the market, some of which are free and open-source tools that allow monitoring at no additional cost to IT professionals. This paper seeks to find which of the three tools Wireshark, Suricata, and Snort perform best in the metrics of ease of download, installation difficulty, ease of use, and quality of data from packet sniffing. Of the programs tested for this purpose, Wireshark performs incredibly well in all categories, especially the quality of data from packet sniffing. Evident when compared to Suricata and Snort which are more complicated to setup and utilize and don't provide the same quality of packet data.

**Introduction**

Network monitoring tools allow an IT professional to capture packets that traverse the networks that they oversee. The process of capturing these packets is called packet sniffing [1]. There is a myriad of tools available to IT professionals to perform this task. Some of these tools provide more information on the packets that are sniffed than others. Free and open-source tools are of particular interest to IT professionals who may be constrained by a budget. For a program to be considered open-source its source code must be available and be able to be modified, copies must be able to be made and disseminated, and the user must have the ability to make changes to it [2]. Some of the more notable tools currently available to perform this task include Wireshark, Snort, and Suricata. Computer scientists have been testing these tools to find which ones are best suited for the task. Tools such as Wireshark are usable across multiple operating systems and are useful for packet sniffing and network intrusion detection and were found to be comparable to other tools such as Snort and Suricata [3].

Wireshark claims to be the best among the tools to sniff packets on the network. This is a claim that previous studies have shown can be the case depending on the metrics that are being measured. It is my contention that this is also the case, and I will put it to the test seeking to find the tool that provides the most comprehensive information on packets sniffed for the task of network security and administration. The information provided by these tools can assist a network administrator in identifying and countering traffic that is attacking their network from a malicious source [4]. In the example of a DDoS attack if the network monitoring software shows that a specific port is being targeted the administrator could restrict access to that port to put a stop to the attack, if malicious emails are being sent from the same address consistently that address can be blocked from sending traffic to the network. This means that finding the correct tool for the job of monitoring networks and attacks is paramount for network administration and security especially as networks continue to grow in size and complexity.

**Materials and Methods**

**Refer to github for full methods numbered in order for execution.**

**<https://github.com/joshualarkin/Joshua-Larkin-Senior-Research>**

The free and open-source network management tools Wireshark, Suricata and Snort were downloaded from their respective websites: Wireshark.org, suricata.io/download/, and Snort.org. These were downloaded to a desktop PC running windows 10 pro 64 bit with an AMD Ryzen 7 3700x 8-core processor running at 3.6 GHz and having 64 GB of RAM and a 500Mbps download and upload speed. The NIC is the onboard connection for the motherboard, an ASUS X570-PRO.

Wireshark Installation:

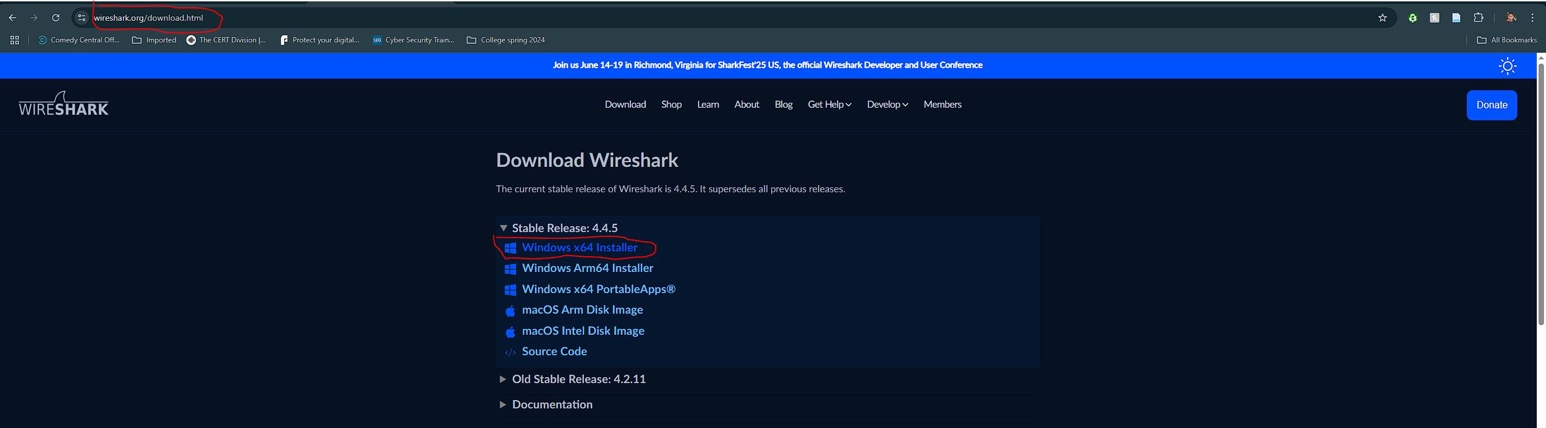
First the program was downloaded from wireshark.org/download.html 

Figure 1 Wireshark Download

After the download was completed the install file was run and the prompts were followed.

When prompted click finish.

Suricata Installation:

Download Suricata from suricata.io/download/

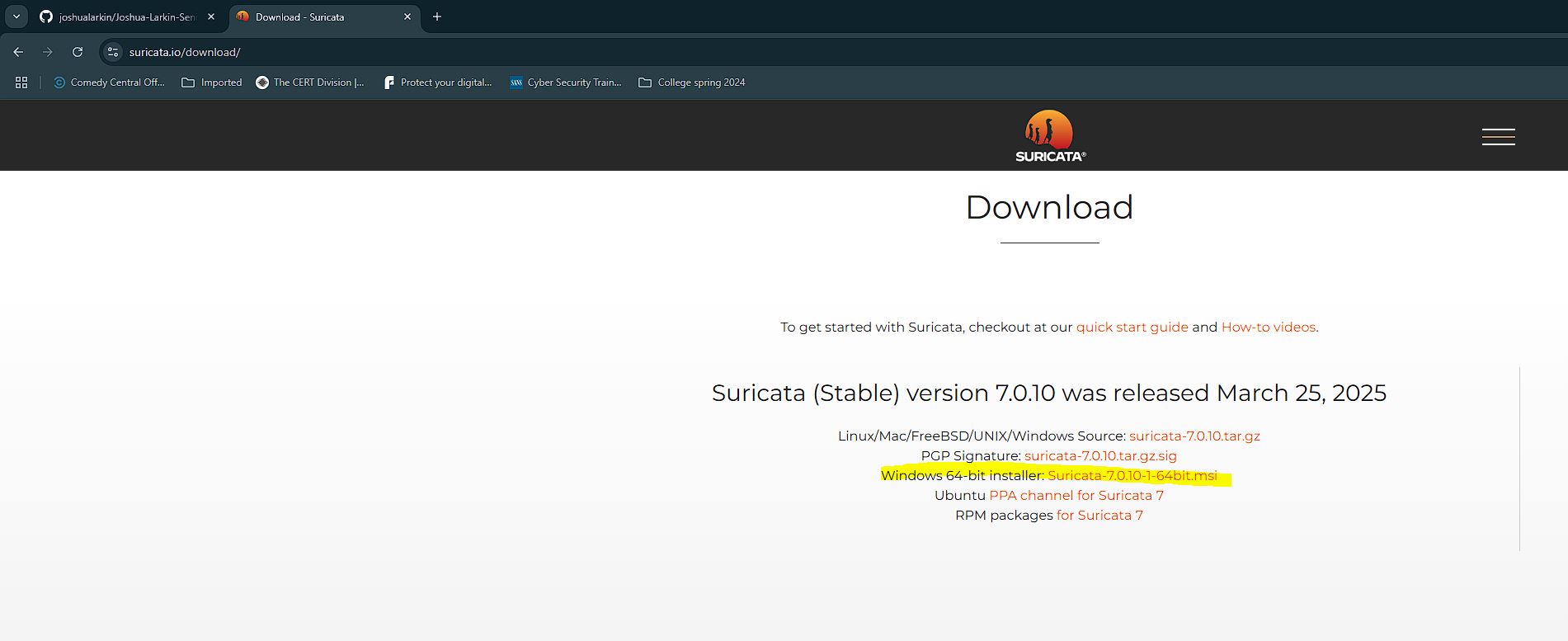


Figure 2 Suricata Download

After download has completed run the installation file.

As necessitated by the Suricata install, npcap must be installed. This can be obtained from npcap.com/dist/.

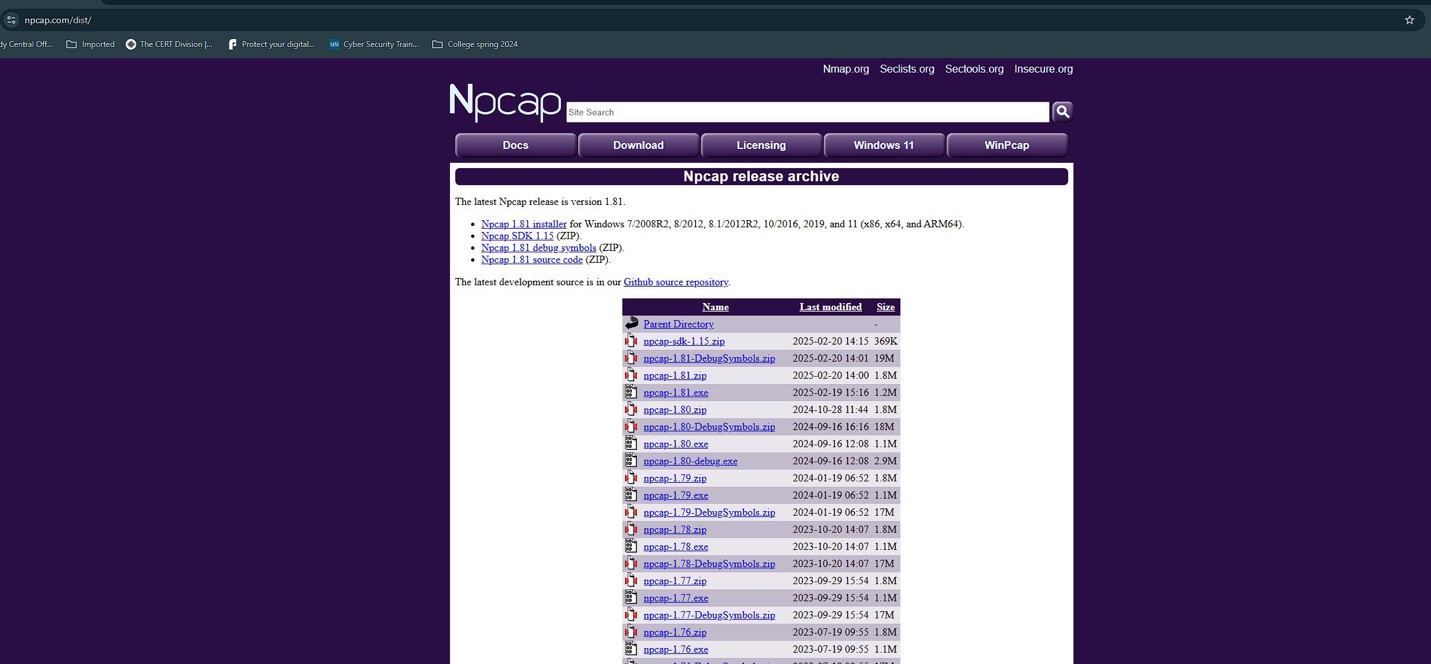


Figure 3 Npcap Download

Npcap 1.81 was selected and installed. Upon completion of the download run the installer.

Select to finish once prompted.

Snort Installation:

First the installation was downloaded from the site snort.org.

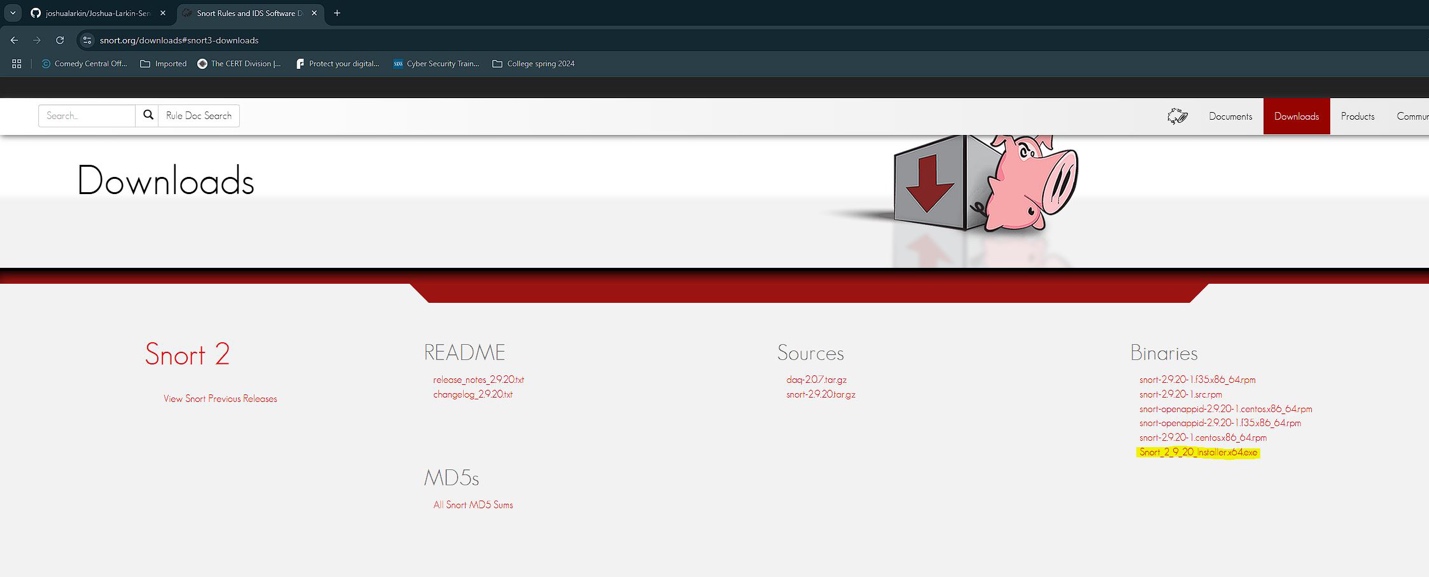


Figure 4 Snort 2 Download

Snort version 2 is what will be tested.

Once the binary was downloaded the installer was run.

The version of Npcap already installed will also work with Snort 2.

After installation the configuration file for Snort was set up for the network.

This completed all the tool installation. For complete step by step visual install instructions refer to the github page. The metrics by which these tools will be measured are ease of download, installation difficulty, ease of use and quality of data from packet sniffing.

Next perform the packet sniffing. The sniffing took place during an email sent from a HTTPS source, the source being a yahoo.com address. The email was simply titled test, there was no attached data and the body said test. Then communication with the HTTP website http://example.com was performed and sniffed.

**RESULTS**

Wireshark:

Ease of Download:

Wireshark is easy to download, navigate to wireshark.com/download (see figure 1).

Installation Difficulty:

The install of Wireshark is simple, it is only needed to click through the prompts until completion. Full installation available on the project github page.

Ease of Use:

Once Wireshark is installed it is ready to use, simple packet sniffing is easy to execute, you just select the interface that you are looking to sniff and then press start (see figure 5). Wireshark even shows you which interface currently has active traffic with an activity line. Wireshark does the rest from that point and once you feel you’ve collected enough data you press stop and can analyze the data you’ve collected.

A black and orange line

AI-generated content may be incorrect.

Figure 5 Wireshark interface selection

Quality of Data from Packet Sniffing:

HTTPS (yahoo.com email send):

Wireshark can collect and show packets that are encrypted, it doesn’t appear to have the functionality to show the encrypted data. Although it does show who is sending and receiving (see figures 6 and 7). There is lots of data about each packet and many packets that were captured in mere moments.

A close-up of a white paper

AI-generated content may be incorrect.

Figure 6 encrypted Wireshark packet sniff

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 7 encrypted Wireshark packet sniff information

HTTP (http://example.com):

For the unencrypted packet sniffing for Wireshark the program collects all packets available on the network and proceeds to give very detailed information about each packet (see figure 8 and figure 9).

A screen shot of a computer

AI-generated content may be incorrect.

Figure 8 unencrypted Wireshark packet sniff

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 9 unencrypted Wireshark packet information

Suricata:

Ease of Download:

Suricata is easy to download, just navigate to suricata.io/download/ (see figure 2).

Installation Difficulty:

The installation only requires that you click through the prompts until finished. Full step by step installation available on project github.

Ease of Use:

The Suricata setup requires extra work to get it to function. Npcap must be installed (see figure 3). Also, a rules set must be downloaded that Suricata can use in order to know what traffic it is looking for. The instructions for this are found in a how to for windows file in the Suricata directory (see figure 10). Suricata also requires you to configure the Suricata.yaml file to set the correct path for the rules and indicate which rules you do and do not want the program to use by marking them off with a # symbol next to them (see figure 11). After all of this is setup Suricata is ready to sniff packets so long as you use the correct syntax from the command line ensuring that you are running the command prompt with administrative access (see figure 12). You are also required to know precisely the IP address of the interface that you want to monitor.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 10 location of Suricata How To

A screenshot of a computer

AI-generated content may be incorrect.

Figure 11 Suricata.yaml

A screenshot of a computer

AI-generated content may be incorrect.

Figure 12 Suricata command to run

Quality of Data from Packet Sniffing:

HTTPS (yahoo.com email send):

While Suricata shows the types of packets that were collected it doesn’t show any distinct information about the packets, this makes it difficult to identify any data linked specifically to the sending of the email or even encrypted data in general (see figure 13).

A screenshot of a computer

AI-generated content may be incorrect.

Figure 13 Suricata encrypted packet sniff

HTTP (http://example.com):

For the unencrypted traffic there is no discernable difference from the encrypted data. Suricata doesn’t provide in depth information in its logs about the packets, just general information and how many of which types of packets were collected (see figure 14).

A screenshot of a computer

AI-generated content may be incorrect.

Figure 14 Suricata Unencrypted packet sniff

Snort:

Ease of Download:

Snort 2 is an easy download, locating the webpage to download the installation file at snort.com/downloads#snort3-downloads. Snort 2 contains an install package while at the time of this writing Snort 3 did not and as such was not chosen for this project. See Figure 4.

Installation Difficulty:

Snort 2 is easy to install, as with most programs today it is a matter of clicking through the prompts until finished. The complete installation process has been recorded and is available on the projects github.

Ease of Use:

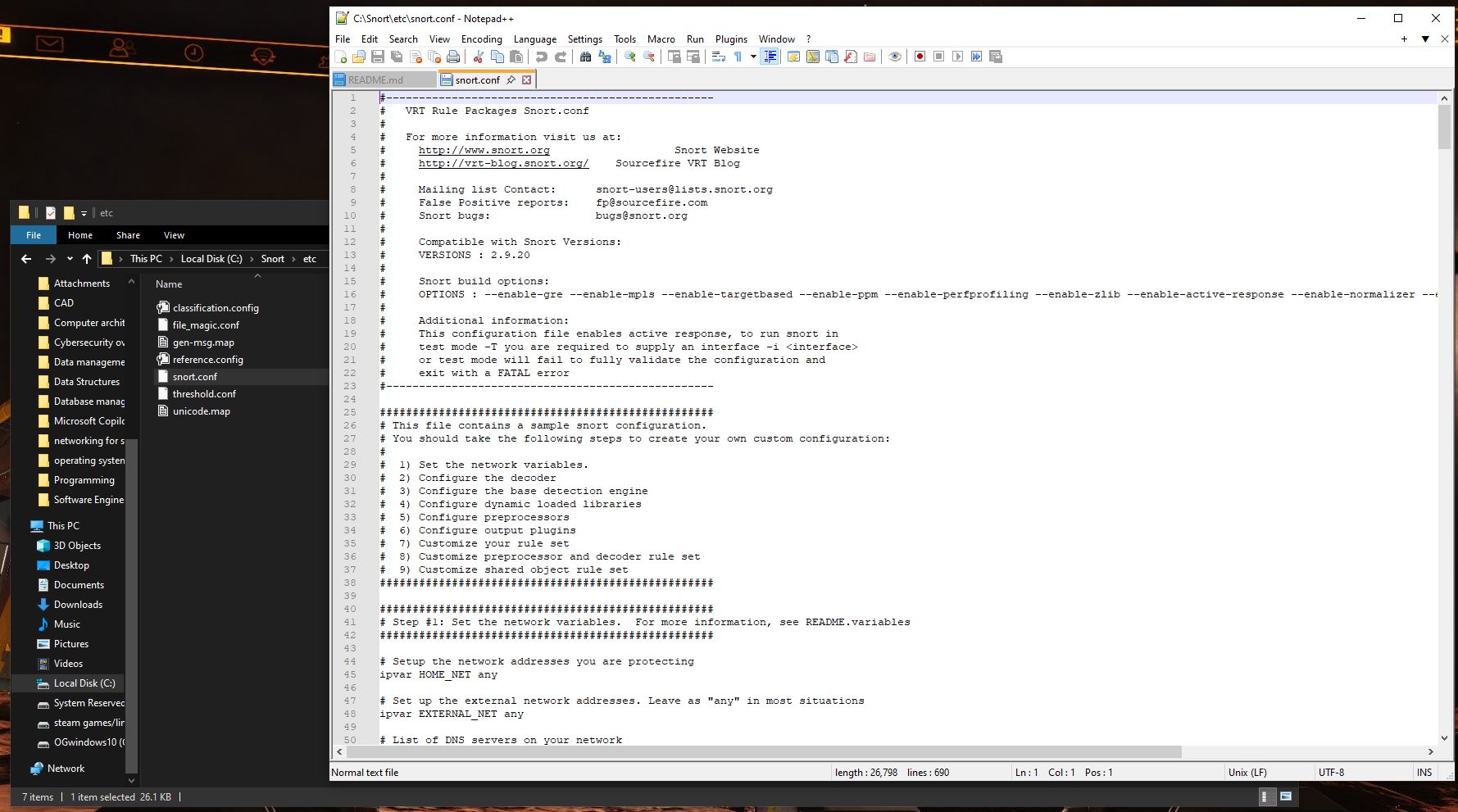
Snort 2 requires that the snort.conf file be edited with specific network information so that Snort has access to all the information about the network that it will be monitoring (see figures 15, 16 and 17). Snort also requires that Npcap be installed (see figure 3). 

Figure 15 snort config file setup

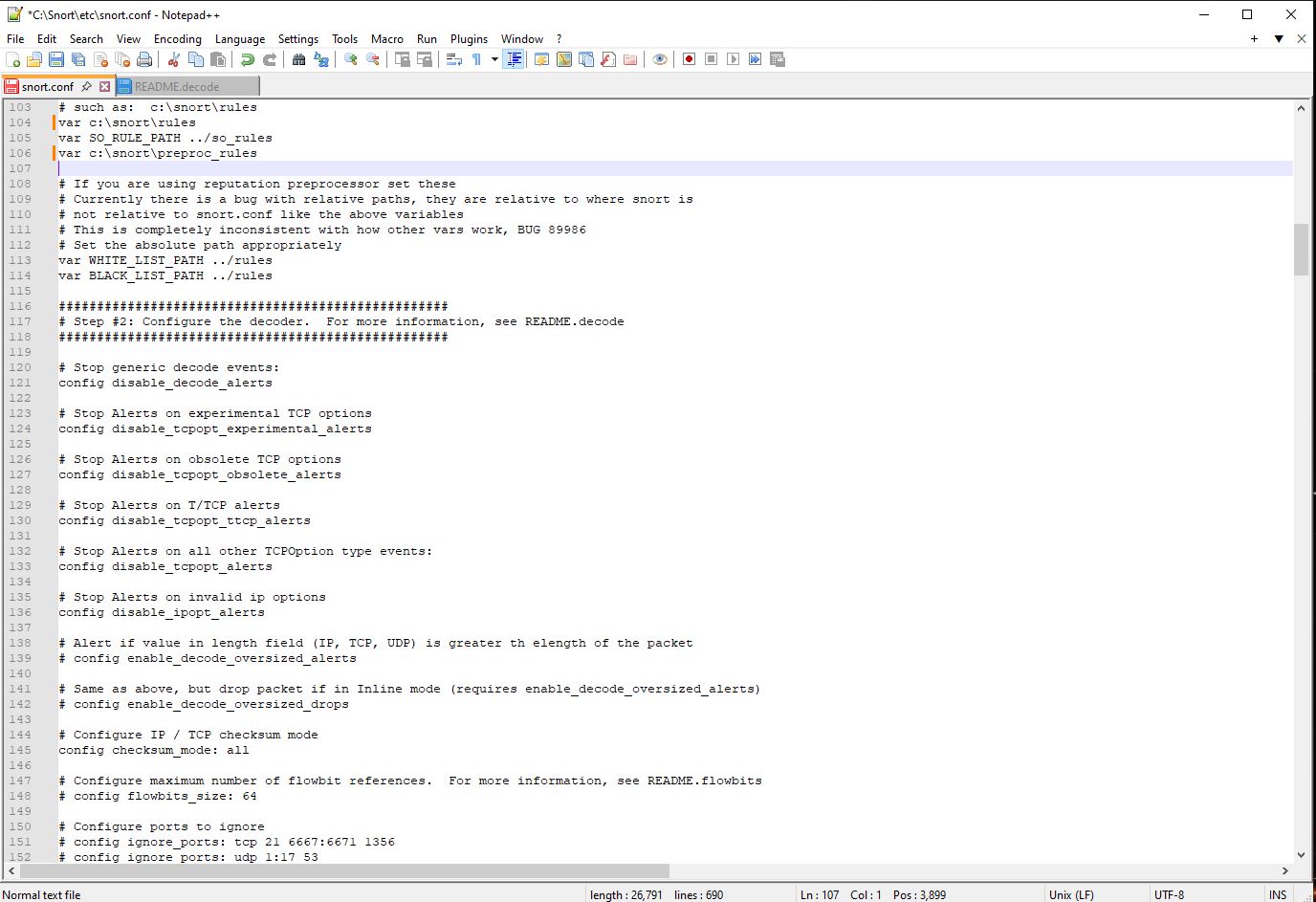


Figure 16 Snort config file contintued

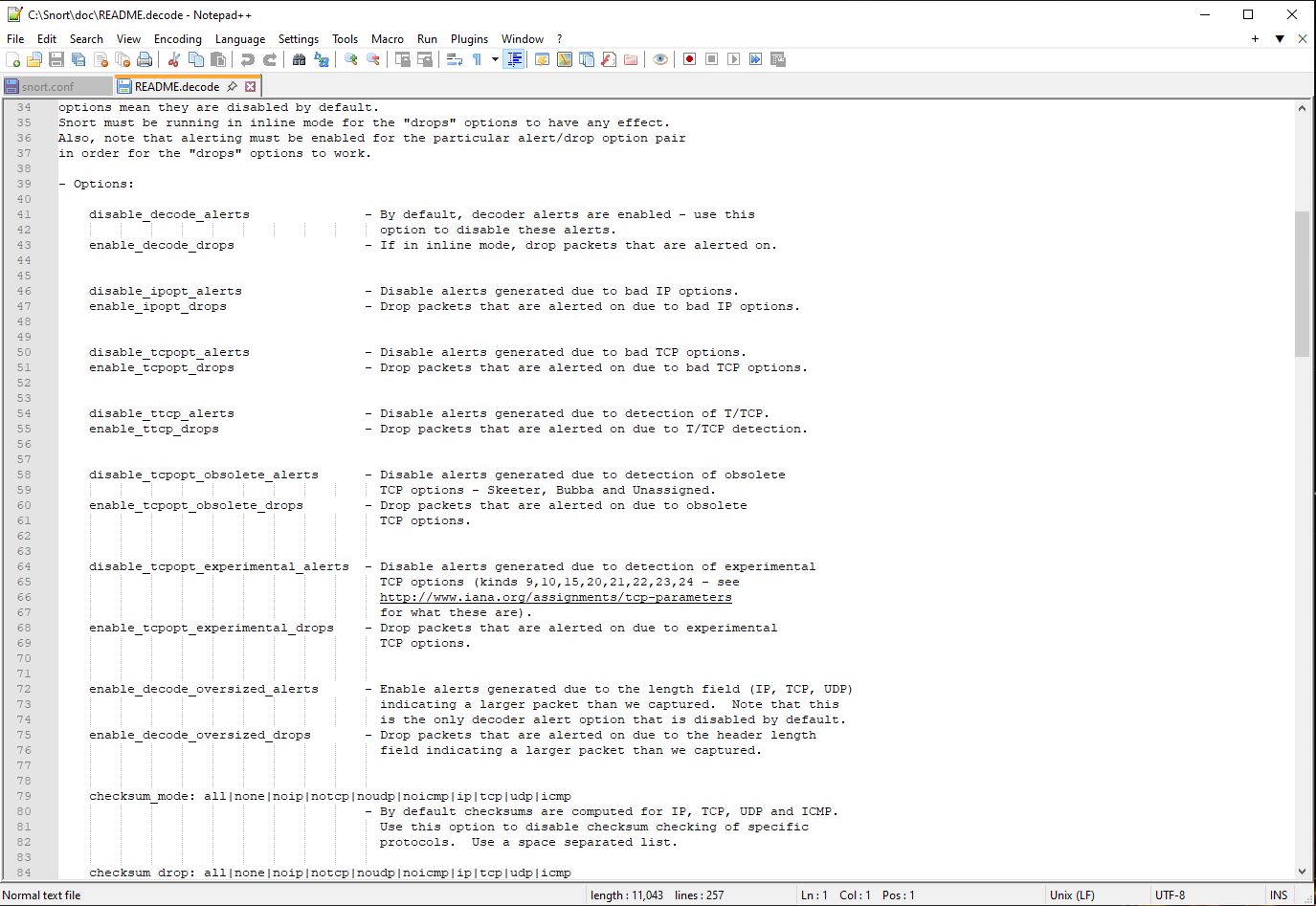


Figure 17 more Snort configuration file

Quality of data from Packet Sniffing:

HTTPS ( None):

Unable to collect results as sniffing was not able to be carried out.

HTTP (None):

Unable to collected packet sniffing results as sniffing could not be carried out.

**Discussion**

Of the metrics that were tested, ease of download, installation difficulty, ease of use and quality of data from packet sniffing there is only one of the programs that performed well across all metrics.. From the results that were captured it is suggestive that Wireshark outperforms its competition in this testing. Suricata and Snort while both easy to install they are more difficult to setup. Both require the added installation of Npcap to sniff packets and additional setup for the programs to sniff the network. Suricata requires editing the Suricata.yaml file and downloading a rules set to make it function. While Snort requires setting up the config file, it may require a rules set though I did not see any documentation in its setup to suggest this. The program simply wouldn’t sniff packets and may only be useful with far more advanced knowledge of the system and the program. Wireshark comes out ahead in all metrics, after install it is ready to start sniffing packets with no need to setup configuration files or install extra software. Additionally Wireshark provides a wealth of information about the packets it collects even though it can’t display the contents of encrypted packets, while Suricata only displays the type of information collected and no details. This study is limited to looking at data from contacting an unsecure website and monitoring the sending of an encrypted email from a browser based email service. There are many different types of traffic and with different types of testing Suricata or Snort may prove more useful than Wireshark. According to the information available Snort is primarily for monitoring a connection for specific types of data and sending an alert so that IT personnel can intervene and in this way it may outperform other programs, further testing would be required to show this. Targeted monitoring for specific types of traffic would afford an IT professional to intervene and block harmful traffic if it became necessary. Computer scientists have been testing these tools to find which ones are best suited to the task of network monitoring. Studies have shown that Wireshark outperformed Snort and Suricata when it came to packet capture and analysis [3]. There are other tools available that are free and open source that could be tested to see how they perform with regards to these tools that have been tested. The work done to test these tools can be used to further improve not only these free and open-source tools but also other available tools. This information can also be applied to future tools, informing what those tools need capture for information and how to make them more user friendly and usable by network administrators to monitor and secure their networks.

**References**

[1]

D. A. Varanasi and P. Swathi, “Comparative Study of Packet Sniffing tools for HTTP Network Monitoring and Analyzing,” *IJCSET*, vol. 6, no. 12, p. 4, 2016.

[2]

B. Perens, M. Sroka, and M. Stu, “The Open Source Definition”.

[3]

E. John, C. Kalu, and P. Asuquo, “Comparative Performance Analysis Of Cybersecurity Tools On A Wireless Network With WPA2 Encryption,” *Journal of Multidisciplinary Engineering Science and Technology (JMEST)*, vol. 9, no. 12, p. 17, 2022.

[]

“What is Network Monitoring? | IBM.” Accessed: Feb. 27, 2025. [Online]. Available: <https://www.ibm.com/think/topics/network-monitoring>