WIRESHARK REPORT

Sergiusz Pieszak
[redacted]

Contents

1. Introduction 2
2. Tool overview 2
3. Simulating Traffic 3
   3.1 Traffic Generators 3
   3.2 Custom Scripts or Applications 3
   3.3 Virtual Machines or Containers 3
   3.4 Network Traffic Replay 4
4. Capturing Traffic 4
   4.1 Understanding the PCAP File 4
   4.2 Simulated Traffic Generation 4
5. Conclusion 6
6. Reference 7

1. Introduction

In today's digitally interconnected world, the need for safeguarding against cyber threats is increasing for organizations striving to maintain the integrity and security of their digital assets. [redacted] cyber security department is at the forefront of this endeavour, with a steadfast commitment to safeguard the organization's defences against potential vulnerabilities and breaches. The department's mandate is to showcase its proficiency in monitoring network traffic and discerning its intricate patterns with precision and efficiency.

This report explores the meticulous process employed by [redacted] cyber security department, as it strives to demonstrate its capabilities in the field of network traffic analysis. The department is responsible for ensuring robust cybersecurity measures, ensuring that the processes involved in this critical attempt are outlined. From the inception of simulating traffic scenarios to the intricate task of capturing real-time data streams, this report serves as a comprehensive guide to the methodologies employed by the department. Furthermore, it focuses on techniques employed to decipher and show insights from the captured data, utilizing the sophisticated capabilities of Wireshark as a cornerstone tool in this analysis process.

2. Tool overview

According to CompTIA (2020) Wireshark is a powerful network protocol analyser used for network troubleshooting, analysis, software and communications protocol development, and education. It allows users to capture and interact with the traffic running on a computer network in real time. Wireshark provides a wide range of protocols and provides extensive inspection capabilities, allowing users to examine the contents of packets and understand network behaviour. It is an essential tool for network administrators, security professionals, and anyone else who needs to understand and troubleshoot network traffic. Some of its key features include:

1. Packet Capture: Wireshark has the ability to capture live network traffic from various interfaces or to read packet capture files from other network analysers.

2. Protocol Decoding: It can decode numerous protocols across different network layers, allowing users to inspect the details of each packet exchanged between network devices.

3. Deep Inspection: Wireshark provides detailed information about each packet, including source and destination addresses, protocol-specific details, packet timing, and payload data.

4. Filtering and Search: Users can apply numerous filters to focus on specific types of traffic or search for packets based on various criteria, such as IP addresses, protocol types, and packet contents.

5. Packet Analysis: Wireshark offers extensive tools enabling users to analyse packet streams, including statistics, flow diagrams, and sequence analysis, to identify patterns, anomalies, and potential security threats.

6. Protocol Support: It supports a wide range of network protocols, which include TCP/IP, HTTP, DNS, FTP, SSH, SSL/TLS, and many others, making it suitable for analysing diverse network environments.

7. Customization and Extensibility: Wireshark gives users the ability to customize the analysis environment, create custom dissectors for proprietary protocols, and extend its functionality through plugins and scripting.

8. Visualization: It offers various visualization options, such as packet graphs and IO graphs, to help users visualize network traffic patterns and performance metrics.

Wireshark is a comprehensive tool that facilitates in-depth analysis and troubleshooting of network issues, security incidents, and performance optimizations. Its robust feature set and user-friendly interface make it an indispensable tool for network administrators, security analysts, developers, and researchers alike.

3. Simulating Traffic

As stated by F. Melakessou and T. Engel (2009), simulating traffic involves generating network activity that mimics real-world scenarios. To achieve this, we employ various methods and techniques such as packet generators or network emulators. We will use simulated traffic to demonstrate the monitoring capabilities of our network infrastructure during the demonstration DNStuff (2020).

In Wireshark, you can't directly simulate network traffic, but you can capture and analyse real network traffic. If you want to simulate different types of traffic for testing purposes, you would typically use other tools or techniques to generate that traffic and then capture it with Wireshark for analysis. Below are a few approaches:

3.1 Traffic Generators

In the field of cybersecurity, a variety of tools are available for generating various types of network traffic. These include DDoS simulators, network emulators, and traffic generators such as Ostinato, Ostinato (no date), or D-ITG. These tools are vital for creating specific traffic patterns, which can then be captured and analysed using Wireshark. For instance, DDoS simulators can replicate large-scale attacks, while network emulators allow for the creation of diverse network environments. Traffic generators offer flexibility in crafting customized traffic patterns. Such tools are excellent educational tools for SOC engineers to learn how certain threats behave and look like. In this example, we'll explore the creation of a specific traffic pattern to illustrate this process.

3.2 Custom Scripts or Applications

You can write custom scripts or applications to generate specific types of network traffic. For example, you could write a script to simulate HTTP requests, FTP transfers, or VoIP traffic. Once you run your script or application, you can capture the generated traffic with Wireshark. A good example of such a script would be Open Traffic Generator, GitHub(2021).

3.3 Virtual Machines or Containers

You can construct virtual machines or containers to simulate different network environments and generate traffic within those environments. Tools like VirtualBox, VMware, Docker, or Kubernetes can be used to create isolated network environments where you can generate and capture traffic using Wireshark, Microsoft Azure (no date).

3.4 Network Traffic Replay

If you have captured network traffic from a real location, you can replay that traffic using tools like tcpreplay, Tcpreplay (No date), or Ostinato. This allows you to replay the captured traffic onto a test network and analyse it with Wireshark as if it were real-time traffic.

Remember to utilize these methods responsibly and ensure that any simulated traffic does not cause harm to real networks or violate any security regulations.

4. Capturing Traffic

Capturing network traffic is essential for analysing communication patterns, identifying potential threats, and understanding network behaviour. We employ packet capture (pcap) technology to capture data packets traversing the network, SolarWinds (no date). This allows us to inspect the contents of packets, including source and destination addresses, protocols used, and payload data.

4.1 Understanding the PCAP File

Upon capturing network traffic, we analyse the pcap file using Wireshark, a powerful network protocol analyser. Wireshark provides detailed information on network activity, allowing us to detect anomalies, suspicious behaviour, and potential security breaches. Understanding the pcap file involves examining various parameters such as packet headers, timestamps, packet size, and payload contents.

4.2 Simulated Traffic Generation

The decision was made to use a python script to simulate HTTP traffic, Zaczyński, B. (2023). Using the commands written in the figure below I hosted a HTTP server on http://localhost:8080. This allowed me to run Wireshark traffic capture through the local host address of 127.0.0.1.

```
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sergi>python -m http.server 8080
Serving HTTP on :: port 8080 (http://[::]:8080/) ...
```

Figure 1 – Starting http server in command line using python

Figure 2 shows how the localhost website was accessed. As you can see each request is documented with ip number, date and time, and the type of request made by the user.

```
ConnectionAbortedError: [WinError 10053] An established connection was aborted
----------------------------------------
::ffff:127.0.0.1 - - [25/Apr/2024 14:11:47] "GET /Pictures/ HTTP/1.1" 200 -
::ffff:127.0.0.1 - - [25/Apr/2024 14:12:37] "GET / HTTP/1.1" 200 -
::ffff:127.0.0.1 - - [25/Apr/2024 14:12:41] "GET /Pictures/ HTTP/1.1" 200 -
::ffff:127.0.0.1 - - [25/Apr/2024 14:13:33] "GET /Documents/ HTTP/1.1" 200 -

Keyboard interrupt received, exiting.
```

Figure 2 – Details of the active http site are displayed in the cmd

Furthermore, further analysis can be conducted using Wireshark to examine the transmitted traffic in detail. Figure 3 displays the entire website frame sent to the user, while Figure 4 breaks down the HTTP Header into sections. This breakdown provides insights into the data structure, aiding in identifying anomalies or security threats. Wireshark, along with these visual aids, facilitates a comprehensive analysis of network traffic, enhancing cybersecurity measures effectively.

```
0010  7f 00 00 01 7f 00 00 01  1f 90 eb 49 20 78 c0 23   ····· ··· ···I x·#
0020  8c d1 89 41 50 18 20 f8  6b 9d 00 00 3c 21 44 4f   ···AP· · k···<!DO
0030  43 54 59 50 45 20 48 54  4d 4c 3e 0a 3c 68 74 6d   CTYPE HT ML>·<htm
0040  6c 20 6c 61 6e 67 3d 22  65 6e 22 3e 0a 3c 68 65   l lang=" en">·<he
0050  61 64 3e 0a 3c 6d 65 74  61 20 63 68 61 72 73 65   ad>·<met a charse
0060  74 3d 22 75 74 66 2d 38  22 3e 0a 3c 74 69 74 6c   t="utf-8 ">·<titl
0070  65 3e 44 69 72 65 63 74  6f 72 79 20 6c 69 73 74   e>Direct ory list
0080  69 6e 67 20 66 6f 72 20  2f 44 6f 63 75 6d 65 6e   ing for  /Documen
0090  74 73 2f 3c 2f 74 69 74  6c 65 3e 0a 3c 2f 68 65   ts/</tit le>·</he
00a0  61 64 3e 0a 3c 62 6f 64  79 3e 0a 3c 68 31 3e 44   ad>·<bod y>·<h1>D
00b0  69 72 65 63 74 6f 72 79  20 6c 69 73 74 69 6e 67   irectory  listing
00c0  20 66 6f 72 20 2f 44 6f  63 75 6d 65 6e 74 73 2f    for /Do cuments/
00d0  3c 2f 68 31 3e 0a 3c 68  72 3e 0a 3c 75 6c 3e 0a   </h1>·<h r>·<ul>·
00e0  3c 6c 69 3e 3c 61 20 68  72 65 66 3d 22 42 41 43   <li><a h ref="BAC
00f0  4b 55 50 25 32 30 4d 2e  52 2e 25 32 30 49 6e 73   KUP%20M. R.%20Ins
0100  6f 6c 76 65 6e 63 79 2f  22 3e 42 41 43 4b 55 50   olvency/ ">BACKUP
0110  20 4d 2e 52 2e 20 49 6e  73 6f 6c 76 65 6e 63 79    M.R. In solvency
0120  2f 3c 2f 61 3e 3c 2f 6c  69 3e 0a 3c 6c 69 3e 3c   /</a></l i>·<li><
0130  61 20 68 72 65 66 3d 22  42 61 6e 64 69 63 61 6d   a href=" Bandicam
0140  2f 22 3e 42 61 6e 64 69  63 61 6d 2f 3c 2f 61 3e   /">Bandi cam/</a>
0150  3c 2f 6c 69 3e 0a 3c 6c  69 3e 3c 61 20 68 72 65   </li>·<l i><a hre
0160  66 3d 22 43 75 73 74 6f  6d 25 32 30 4f 66 66 69   f="Custo m%20Offi
0170  63 65 25 32 30 54 65 6d  70 6c 61 74 65 73 2f 22   ce%20Tem plates/"
0180  3e 43 75 73 74 6f 6d 20  4f 66 66 69 63 65 20 54   >Custom  Office T
0190  65 6d 70 6c 61 74 65 73  2f 3c 2f 61 3e 3c 2f 6c   emplates /</a></l
01a0  69 3e 0a 3c 6c 69 3e 3c  61 20 68 72 65 66 3d 22   i>·<li><a href="
01b0  43 79 72 6f 78 25 32 30  57 6f 72 6b 2f 22 3e 43   Cyrox%20 Work/">C
01c0  79 72 6f 78 20 57 6f 72  6b 2f 3c 2f 61 3e 3c 2f   yrox Wor k/</a></
01d0  6c 69 3e 0a 3c 6c 69 3e  3c 61 20 68 72 65 66 3d   li>·<li>·<a href=
01e0  22 44 65 66 61 75 6c 74  2e 72 64 70 22 3e 44 65   "Default .rdp">De
01f0  66 61 75 6c 74 2e 72 64  70 3c 2f 61 3e 3c 2f 6c   fault.rd p</a></l
0200  69 3e 0a 3c 6c 69 3e 3c  61 20 68 72 65 66 3d 22   i>·<li><a href="
0210  64 65 73 6b 74 6f 70 2e  69 6e 69 22 3e 64 65 73   desktop. ini">des
0220  6b 74 6f 70 2e 69 6e 69  3c 2f 61 3e 3c 2f 6c 69   ktop.ini </a></li
0230  3e 0a 3c 6c 69 3e 3c 61  20 68 72 65 66 3d 22 45   >·<li><a  href="E
0240  4e 54 52 45 50 52 45 4e  45 55 52 53 48 49 50 2f   NTREPREN EURSHIP/

Frame (1387 bytes)   Reassembled TCP (1499 bytes)
```

Figure 3 – Screenshot from Wireshark captured traffic

```
▸ Frame 32: 510 bytes on wire (4080 bits), 510 bytes captured (4080 bits) on interface \Device\NPF_Loopbac
▸ Null/Loopback
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ Transmission Control Protocol, Src Port: 60227, Dst Port: 8080, Seq: 1, Ack: 1, Len: 466
▾ Hypertext Transfer Protocol
  ▾ GET / HTTP/1.1\r\n
    ▸ [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /
      Request Version: HTTP/1.1
    Host: localhost:8080\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-GB,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate, br\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    Sec-Fetch-Dest: document\r\n
    Sec-Fetch-Mode: navigate\r\n
    Sec-Fetch-Site: none\r\n
    Sec-Fetch-User: ?1\r\n
    DNT: 1\r\n
    Sec-GPC: 1\r\n
    \r\n
    [Full request URI: http://localhost:8080/]
    [HTTP request 1/1]
    [Response in frame: 50]
```

Figure 4 - Screenshot from Wireshark showing GET request

As depicted in Figure 4 above, a GET request is initiated by the browser. The graphical user interface (GUI) enhances the usability and readability of the header information, thereby expediting the tasks of Security Operations Center (SOC) personnel and network engineers, who can efficiently dissect the data.

5. Conclusion

In conclusion, the contemporary digital landscape demands vigilant measures to safeguard against the ever-evolving range of cyber threats. [redacted] cyber security department stands as a beacon of resilience in this ongoing battle, driven by an unwavering commitment to fortify the organization's defences and uphold the integrity of its digital assets. Through the meticulous exploration of network traffic analysis processes outlined in this essay, it becomes evident that proactive vigilance and strategic deployment of advanced tools like Wireshark are paramount in ensuring robust cybersecurity measures.

[redacted] cyber security department utilizes a proactive approach to confront potential vulnerabilities and breaches. This essay is not only a testament to the department's ability to monitor network traffic, but also a guide for organizations seeking to improve their cybersecurity posture. As digital landscapes continue to evolve, the lessons gleaned from [redacted] approach underscore the imperative of ongoing vigilance and adaptation in the face of emerging cyber threats.

6. Reference

CompTIA (2020) *What Is Wireshark and How to Use It*. Available at: https://www.comptia.org/content/articles/what-is-wireshark-and-how-to-use-it. (Accessed on: 20 April 2024)

DNSstuff (2020) *6 Best Network Traffic Generator and Simulator Stress Test Tools*. Available at: https://www.dnsstuff.com/network-traffic-generator-software. (Accessed on: 18 April 2024)

F. Melakessou and T. Engel (2009) *Network Traffic Simulator 2.0: Simulating the internet traffic*. Available at: https://ieeexplore.ieee.org/document/5416788. (Accessed on: 19 April 2024)

Github (2021) *Open Traffic Generator*. Available at: https://github.com/open-traffic-generator. (Accessed on: 20 April 2024)

Microsoft Azure (no date) *What Is a Virtual Machine and How Does It Work* . Available at: https://azure.microsoft.com/en-gb/resources/cloud-computing-dictionary/what-is-a-virtual-machine. (Accessed on: 22 April 2024)

Ostinato (no date) *Ostinato User Guides*. Available at: https://userguide.ostinato.org/. (Accessed on: 20 April 2024)

SolarWinds (no date) *What Is Packet Capture (PCAP)?* Available at: https://www.solarwinds.com/resources/it-glossary/pcap. (Accessed on: 20 April 2024)

Tcpreplay (No date) *Pcap editing and replaying utilities*. Available at: https://tcpreplay.appneta.com/. (Accessed on: 20 April 2024)

Zaczyński, B. (2023) *How to Launch an HTTP Server in One Line of Python Code*. Available at: https://realpython.com/python-http-server/. (Accessed on: 19 April 2024)