Southeast Technological University

High Level Packet Sniffer

Research Document

Thassanai McCabe – C00250439
Supervisor: Paul Barry

Table of Contents

Table of Figures

1 Introduction

This document presents the findings of the 4th-year project in Cybercrime & IT Security, which aimed to develop a high-level application for packet sniffing. The research carried out and the steps taken to create a plan for the application's development are outlined in this document. Specifically, the document provides an overview of what packet sniffing is and how it can be used by both network administrators and hackers. It also describes the PCAP API and various .pcap file formats, along with their use cases. The document reports on the research conducted on both paid, closed-source and open-source packet sniffers, including screenshots of key features of these tools. Additionally, the document presents the results of the research carried out on programming languages and their PCAP libraries, with screenshots of code that implements these libraries included.

2 The Packet Sniffer

Packet sniffers/analysers can be a piece of software or hardware that examines the flow of data between computers on a wired or wireless network. They utilize filters to allow users to discover only relevant data that they are interested in. The efficiency of packet sniffers can vary depending on where in the network they are deployed, and how the network is configured. Limitations could arise from the placement of network switches for example. On wireless networks, a sniffer will scan one channel at a time. Capability can be expanded using multiple wireless interfaces. Data collected by packet sniffers can be saved to a computer's hard drive for later examination. (Kaspersky, 2022)

Examples of software packet sniffers include Wireshark and Tcpdump. A hardware example of a packet sniffer would be the ProfiShark Network TAP. While software packet sniffers utilize a machine's NIC, a hardware packet sniffer such as the ProfiShark uses RJ45 pins and USB to capture packets. Hardware solutions like ProfiShark boast non-intrusive and reliable packet capturing when compared to software solutions, ensuring quality results. (Profitap, 2022)

The act of packet sniffing can be categorized into two types: active and passive. Active sniffing involves inserting ARPs into a network and overloading a switch table. This results in traffic being redirected to other ports, allowing the sniffer to collect data from the switch. A hacker may utilize an active hacking technique to perform a spoofing attack. Passive sniffing only involves listening in on a network and using a host on that network, unlike active sniffing which involves injecting ARPs into a network. Passively sniffing a network that utilizes hubs would be efficient because traffic on the network is visible to all hosts. It can be difficult to detect passive sniffing. (BasuMallick, 2022)

2.1 Components of a Packet Sniffer/Network Analyser

(Orebaugh, Ramirez and Beale, 2006) State five key software and hardware components of a network analyser/packet sniffer:

- Hardware: Some analysers can alert users of hardware faults such as voltage problems and negotiation errors.
- Capture Driver: The part of the tool that captures raw network traffic. It is the "core" of the network analyser.
- Buffer: Where captured data is stored. These can be disk-based, or memory based.

- Real-time Analysis: The ability to analyse data in real-time. It can be used to discover network issues or intruders.
- Decode: Displaying content that is discovered on the network in a human-readable form. Decoding used depends on the protocol and different packet sniffers will utilize different decodes.

2.2 How a Packet Sniffer Works

A packet is a name given to the smallest unit of data that flows across network nodes to reach the various endpoints that may exist on that network. When a packet is sent, it is assigned a destination address. As the packet travels from its source to its destination, intermediate devices ignore the data in the packet and instead just facilitate the flow of the packet across the network. Packet sniffers alter this behaviour and instead can allow a node to log the data of packets that are passed along the network, even if that packet is not addressed to that node. The node running a packet sniffer can analyse the data of packets it receives and transform it into a human-readable form. "Promiscuous mode" is enabled on the node which is running the packet sniffer. As a hardware packet sniffer is plugged directly into a node, the data that it captures physically travel across it. This allows it to capture data reliably without losing packets to filters or intentional/unintentional causes (Edwards, 2020)

2.3 Applications of a Packet Sniffer

The networks we see today can be large and complex and can contain many different systems which use a large range of different protocols. Therefore, troubleshooting and managing networks today would require the use of a capable tool such as a packet sniffer. Packet sniffers are an essential tool for any network administrator to have. While packet sniffers are a strong choice for network diagnostics, packet sniffers are commonly utilized by malicious actors to gain unauthorized access to devices, steal sensitive information or spy on network activity. These tools can also be used to identify devices on a network without permission. Packet sniffers can expose sensitive information on a network, as well as reveal misconfigurations, bottlenecks, and vulnerabilities, making them a significant threat to network security. (Garcia, 2008)

Security on a network can be addressed using a packet sniffer with an example being used by a company that wants to keep track of how its employees utilize the network. Information such as websites and their contents, emails, and files downloaded can be viewed and logged by administrators using packet sniffers. Furthermore, incoming traffic can be analysed by a packet sniffer for potentially malicious code. (Kaspersky, 2022).

Network administrators can monitor bandwidth usage, and DNS resolution, identify rogue DHCP servers, discover performance issues and respond to incidents by using a packet sniffer. They are given the ability to identify the root cause of a network problem which helps minimize network downtime. The use of PCAP and packet sniffers can reveal how the resources of a network are consumed. (Keary, 2022). A company can see benefits for both security and availability when utilizing a packet sniffer on its network.

While a network administrator monitoring and maintaining their network is a legitimate use of a packet sniffer, black hat hackers may use them for illegitimate reasons. A hacker can use a packet sniffer to discover usernames, passwords, devices, vulnerabilities, and other sensitive information on unsecured networks. Little to no hacking skills are required to run many packet sniffers and extract this information from networks (PagerDuty, 2022). Packet sniffers can allow unauthorized malicious users to execute methods such as password sniffing, TCP Session Hijacking, DNS Poisoning, JavaScript Card Sniffing Attacks, ARP Sniffing, and DHCP Attacks. (BasuMallick, 2022).

3 PCAP: Packet Capture

3.1 PCAP Formats

PCAP files, which can be used to view information about network packets can be generated using sniffers and analysers. They can come in various formats. (Keary 2022). I found that the main two I should consider at this time were "pcap" and "pcapng".

3.1.1 Pcap

PCAP is the name given to the programming application interface (API) that captures network packet data. This network packet data can be written to .pcap files. Wireshark is an example of an application that uses a .pcap file to capture and store network data. A pcap file can come in different formats with examples being Libpcap, WinPcap, and PCAPng. Using pcap files it is possible to view TCP/IP as well as UDP network packet data. Data from layers 2-7 of the OSI model can be collected in pcap files. (Keary, 2022).

3.1.2 PcapNG

Users have the option to save data captured with Wireshark using the .pcapng format. As (Wireshark, 2022) states:

> *"The PCAP Next Generation Dump File Format (or pcapng for short) is an attempt to overcome the limitations of the currently widely used (but limited) libpcap format."*

It is still possible to save files using the .pcap extension on Wireshark. Libpcap 1.1.0 has limited support for reading .pcapng files. Wireshark can open both file types. (Wireshark, 2022).

Pcapng captures additional information and statistics compared to pcap files. Pcapng also has features such as user comments and extended timestamp precision. (Keary, 2022)

3.2 Use of PCAP Files

In section 3.1.1 I describe PCAP formats, here I describe how these files and formats can be used. PCAP files allow IT specialists and administrators to ensure security on their network as real-time traffic data can be analysed using these files. Malware, intrusions, and suspicious activity can be detected on a network as a result. Congestion in a network can be identified as data travel time and bandwidth usage can be read using PCAP. Furthermore, the ability to dive into granular detail into each packet on a network allows for efficient troubleshooting and incident response which minimizes downtime. (Gurubaran, 2021)

The use of PCAP files for security has limitations. An example would be a scenario where attackers use encrypted communications, preventing a packet sniffer from identifying the attack. Furthermore, many cyberattacks are not performed on a victim's network and instead could be carried out through a different attack vector. The location of a packet sniffer can also limit the use of

PCAP files for security purposes as a packet sniffer might not have access to all conversations on a network. (Keary, 2022)

4 Existing Packet Sniffers

This is a list of popular packet sniffers that are currently available. Software available for UNIX, Linux and Windows was researched.

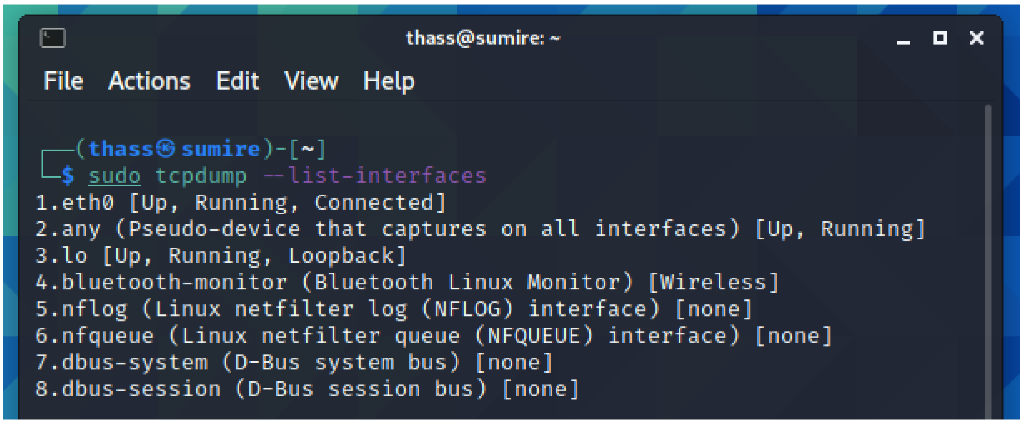
4.1 Tcpdump

Requirements: Unix, Linux, and Microsoft Windows

Tcpdump is a network analysis tool written in C and runs natively on Linux. While there is a version of Tcpdump that runs on Windows, it is commonly found preinstalled on many distributions such as Mac OS X and BSD. (Casey, 2010). Tcpdump allows users to analyse traffic for security reasons as well as carry out maintenance to troubleshoot network issues. It is a command line tool making it suitable for a device to which an administrator may connect remotely. The use of flags allows users to change the behaviour of how the application will run. Tcpdump relies on Libpcap which is a library used for capturing network packets. (Gerardi, 2020)

Below are screenshots demonstrating the use of tcpdump on Kali Linux.

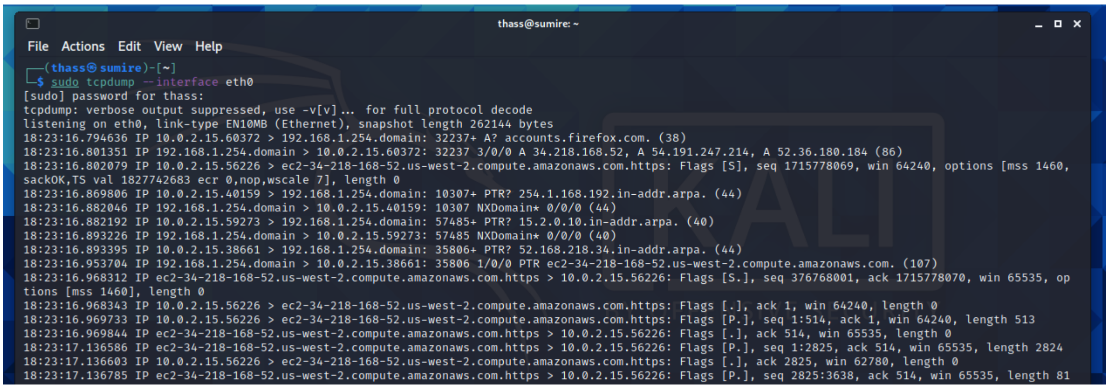
4.1.1 Tcpdump Screenshots



```
┌──(thass㉿sumire)-[~]
└─$ sudo tcpdump --list-interfaces
1.eth0 [Up, Running, Connected]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]
5.nflog (Linux netfilter log (NFLOG) interface) [none]
6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
7.dbus-system (D-Bus system bus) [none]
8.dbus-session (D-Bus session bus) [none]
```

Figure 1 Listing interfaces using tcpdump using the flag: "--list-interfaces".

As (Gerardi, 2020) stated, it was important to note interface “any” which allowed for the capturing of packets on all interfaces.



```
┌──(thass㉿sumire)-[~]
└─$ sudo tcpdump --interface eth0
[sudo] password for thass:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
18:23:16.794636 IP 10.0.2.15.60372 > 192.168.1.254.domain: 32237+ A? accounts.firefox.com. (38)
18:23:16.801351 IP 192.168.1.254.domain > 10.0.2.15.60372: 32237 3/0/0 A 34.218.168.52, A 54.191.247.214, A 52.36.180.184 (86)
18:23:16.802079 IP 10.0.2.15.56226 > ec2-34-218-168-52.us-west-2.compute.amazonaws.com.https: Flags [S], seq 1715778069, win 64240, options [mss 1460,
sackOK,TS val 1827742683 ecr 0,nop,wscale 7], length 0
18:23:16.869806 IP 10.0.2.15.40159 > 192.168.1.254.domain: 10307+ PTR? 254.1.168.192.in-addr.arpa. (44)
18:23:16.882046 IP 192.168.1.254.domain > 10.0.2.15.40159: 10307 NXDomain* 0/0/0 (44)
18:23:16.882192 IP 10.0.2.15.59273 > 192.168.1.254.domain: 57485+ PTR? 15.2.0.10.in-addr.arpa. (40)
18:23:16.893226 IP 192.168.1.254.domain > 10.0.2.15.59273: 57485 NXDomain* 0/0/0 (40)
18:23:16.893395 IP 10.0.2.15.38661 > 192.168.1.254.domain: 35806+ PTR? 52.168.218.34.in-addr.arpa. (44)
18:23:16.953704 IP 192.168.1.254.domain > 10.0.2.15.38661: 35806 1/0/0 PTR ec2-34-218-168-52.us-west-2.compute.amazonaws.com. (107)
18:23:16.968312 IP ec2-34-218-168-52.us-west-2.compute.amazonaws.com.https > 10.0.2.15.56226: Flags [S.], seq 376768001, ack 1715778070, win 65535, op
tions [mss 1460], length 0
18:23:16.968343 IP 10.0.2.15.56226 > ec2-34-218-168-52.us-west-2.compute.amazonaws.com.https: Flags [.], ack 1, win 64240, length 0
18:23:16.969733 IP 10.0.2.15.56226 > ec2-34-218-168-52.us-west-2.compute.amazonaws.com.https: Flags [P.], seq 1:514, ack 1, win 64240, length 513
18:23:16.969844 IP ec2-34-218-168-52.us-west-2.compute.amazonaws.com.https > 10.0.2.15.56226: Flags [.], ack 514, win 65535, length 0
18:23:17.136586 IP ec2-34-218-168-52.us-west-2.compute.amazonaws.com.https > 10.0.2.15.56226: Flags [P.], seq 1:2825, ack 514, win 65535, length 2824
18:23:17.136603 IP 10.0.2.15.56226 > ec2-34-218-168-52.us-west-2.compute.amazonaws.com.https: Flags [.], ack 2825, win 62780, length 0
18:23:17.136785 IP ec2-34-218-168-52.us-west-2.compute.amazonaws.com.https > 10.0.2.15.56226: Flags [P.], seq 2825:3638, ack 514, win 65535, length 81
```

Figure 2 Capturing packets on interface "eth0" using "sudo tcpdump –interface eth0". After starting the capture, Firefox was opened on the machine.

Without the use of the –c or "count" flag, tcpdump will continue to capture packets unless interrupted by a SIGINT/SIGTERM signal.

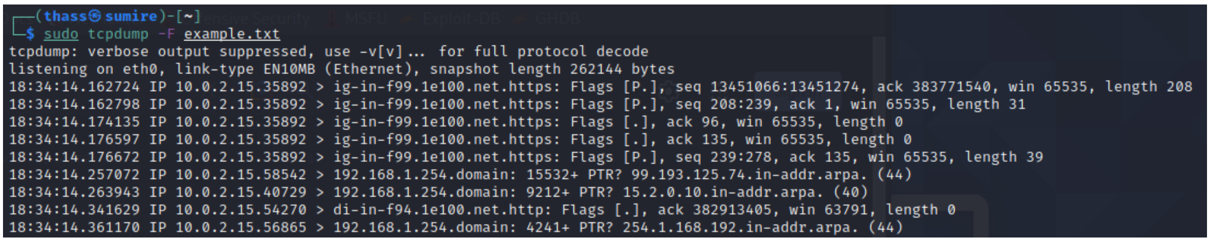
4.1.2 Understanding Tcpdump Output

A portion of the output includes a field that displays Flags followed by a set of square brackets. This is found after the source IP address. The following table provided by (Gerardi, 2020) shows what these characters represent.

Value	***Flag Type***	***Description***
S	SYN	Connection Start
F	FIN	Connection Finish
P	PUSH	Data push
R	RST	Connection reset
.	ACK	Acknowledgement

Users may find that the packets being captured are not relevant to what they are trying to achieve. The use of filters allows users to simplify the output of tcpdump by specifying the parameters of what they are capturing. Parameters can include IP addresses, ports, and protocols. Combining these filters can make tcpdump a very powerful tool. (Gerardi, 2020)

Below is an example of how you would filter packets using tcpdump. Filters can be specified from the command line, or they can be passed in using a file. Below I use a file called "example.txt" which contains "src 10.0.2.15". This is the IP address of the Linux machine running tcpdump. I use the -F flag which expects a filename as a parameter.

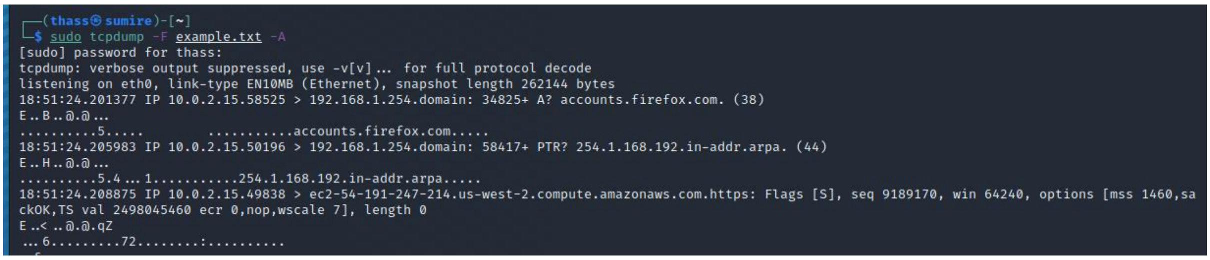


```
┌──(thass㉿sumire)-[~]
└─$ sudo tcpdump -F example.txt
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
18:34:14.162724 IP 10.0.2.15.35892 > ig-in-f99.1e100.net.https: Flags [P.], seq 13451066:13451274, ack 383771540, win 65535, length 208
18:34:14.162798 IP 10.0.2.15.35892 > ig-in-f99.1e100.net.https: Flags [P.], seq 208:239, ack 1, win 65535, length 31
18:34:14.174135 IP 10.0.2.15.35892 > ig-in-f99.1e100.net.https: Flags [.], ack 96, win 65535, length 0
18:34:14.176597 IP 10.0.2.15.35892 > ig-in-f99.1e100.net.https: Flags [.], ack 135, win 65535, length 0
18:34:14.176672 IP 10.0.2.15.35892 > ig-in-f99.1e100.net.https: Flags [P.], seq 239:278, ack 135, win 65535, length 39
18:34:14.257072 IP 10.0.2.15.58542 > 192.168.1.254.domain: 15532+ PTR? 99.193.125.74.in-addr.arpa. (44)
18:34:14.263943 IP 10.0.2.15.40729 > 192.168.1.254.domain: 9212+ PTR? 15.2.0.10.in-addr.arpa. (40)
18:34:14.341629 IP 10.0.2.15.54270 > di-in-f94.1e100.net.http: Flags [.], ack 382913405, win 63791, length 0
18:34:14.361170 IP 10.0.2.15.56865 > 192.168.1.254.domain: 4241+ PTR? 254.1.168.192.in-addr.arpa. (44)
```

Figure 3 The result of using the filter specified in "example.txt" shows only packets where the source 10.0.2.15.

The -A flag can be used to print each captured packet in ASCII.

Inspecting the contents of the packet is a useful utility that can be used for troubleshooting. Users can review sent and received packets to determine if they contain the expected contents. -X can be used to print the contents in hex. (Gerardi, 2020)

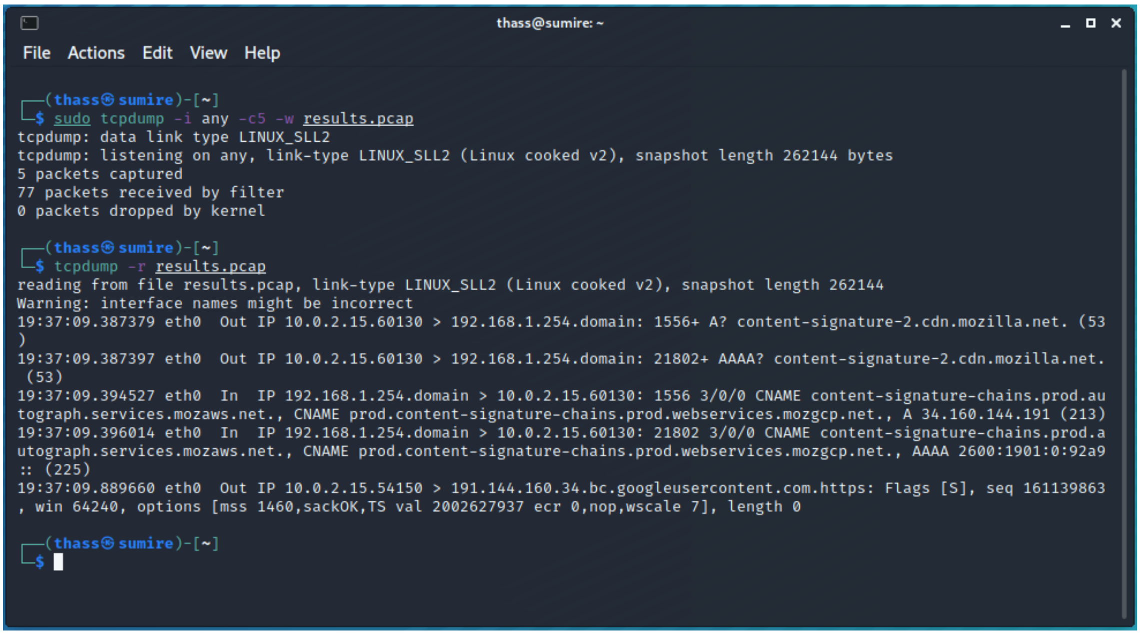


```
┌──(thass㉿sumire)-[~]
└─$ sudo tcpdump -F example.txt -A
[sudo] password for thass:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
18:51:24.201377 IP 10.0.2.15.58525 > 192.168.1.254.domain: 34825+ A? accounts.firefox.com. (38)
E..B..@.@...
..........5.....        ...........accounts.firefox.com.....
18:51:24.205983 IP 10.0.2.15.50196 > 192.168.1.254.domain: 58417+ PTR? 254.1.168.192.in-addr.arpa. (44)
E..H..@.@...
..........5.4...1...........254.1.168.192.in-addr.arpa.....
18:51:24.208875 IP 10.0.2.15.49838 > ec2-54-191-247-214.us-west-2.compute.amazonaws.com.https: Flags [S], seq 9189170, win 64240, options [mss 1460,sa
ckOK,TS val 2498045460 ecr 0,nop,wscale 7], length 0
E..< ..@.@.qZ
...6.........72........:..........
```

Figure 4 After starting tcpdump with the previous filter, Firefox was opened.

4.1.3 .pcap files and Tcpdump

Users can save network data into a pcap file and read existing pcap files using the -r flag which means "read".



```
┌──(thass㉿sumire)-[~]
└─$ sudo tcpdump -i any -c5 -w results.pcap
tcpdump: data link type LINUX_SLL2
tcpdump: listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
5 packets captured
77 packets received by filter
0 packets dropped by kernel

┌──(thass㉿sumire)-[~]
└─$ tcpdump -r results.pcap
reading from file results.pcap, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144
Warning: interface names might be incorrect
19:37:09.387379 eth0  Out IP 10.0.2.15.60130 > 192.168.1.254.domain: 1556+ A? content-signature-2.cdn.mozilla.net. (53
)
19:37:09.387397 eth0  Out IP 10.0.2.15.60130 > 192.168.1.254.domain: 21802+ AAAA? content-signature-2.cdn.mozilla.net.
 (53)
19:37:09.394527 eth0  In  IP 192.168.1.254.domain > 10.0.2.15.60130: 1556 3/0/0 CNAME content-signature-chains.prod.au
tograph.services.mozaws.net., CNAME prod.content-signature-chains.prod.webservices.mozgcp.net., A 34.160.144.191 (213)
19:37:09.396014 eth0  In  IP 192.168.1.254.domain > 10.0.2.15.60130: 21802 3/0/0 CNAME content-signature-chains.prod.a
utograph.services.mozaws.net., CNAME prod.content-signature-chains.prod.webservices.mozgcp.net., AAAA 2600:1901:0:92a9
:: (225)
19:37:09.889660 eth0  Out IP 10.0.2.15.54150 > 191.144.160.34.bc.googleusercontent.com.https: Flags [S], seq 161139863
, win 64240, options [mss 1460,sackOK,TS val 2002627937 ecr 0,nop,wscale 7], length 0

┌──(thass㉿sumire)-[~]
└─$ █
```

Figure 5 Tcpdump is set to capture 5 packets and write the results to a file called "results.pcap". After this command is run, "results.pcap" is read by tcpdump and the output is displayed.

4.2 Omnipeek

Requirements: MacOS, Windows

4.2.1 LiveAction Organisation

Omnipeek is a network analyser that is owned by LiveAction. LiveAction is an organisation that provides tools to support large and complex networks. They assist enterprises with threat response and security operations as well as lowering the cost of the network for customers. LiveAction utilizes a range of telemetry to achieve this, with examples being NetFlow, Packet, SNMP, API, and IPFIX. Enterprises can utilize services offered by LiveAction to support campus networks, data centres, WANs, and branches. LiveAction can help enterprises support their network by offering simple user interfaces, automated network data collection, packet analysis, automated reporting and much more. (LiveAction, 2022). Omnipeek can be used to open pcap files. By default, Omnipeek saves network capture data as a .pkt file.

The (Live Action, 2022) mission statement is as follows:

> *"Support both our enterprise and service provider customers by simplifying their network management experience by using our innovative software platforms".*

4.2.2 Omnipeek Packet Analyser

Omnipeek Network Protocol Analyzer can provide analysis of all network segment types including 1/10/40/100 Gigabit and Voice over IP. It allows users to customise workflows and visualisations of their networks and get information in real-time with the ability to drill down into single packets. This can allow users to carry out forensics as well as find performance issues on their network. Omnipeek can be used with a USB Wi-Fi adapter which allows for wireless packet capture. The 802.11ac adapter supports 802.11ac capture of up to 2 transmit/receive streams and supports 20MHz, 40Mhz, and 80MHz channel operation. Omnipeek can be used to resolve application-level issues at remote locations and branches, WAN connections, and data centres when integrated with LiveCapture. LiveCapture is another packet capture and analysis tool offered by LiveAction. (LiveAction, 2021)

4.2.3 Exploring Omnipeek Features

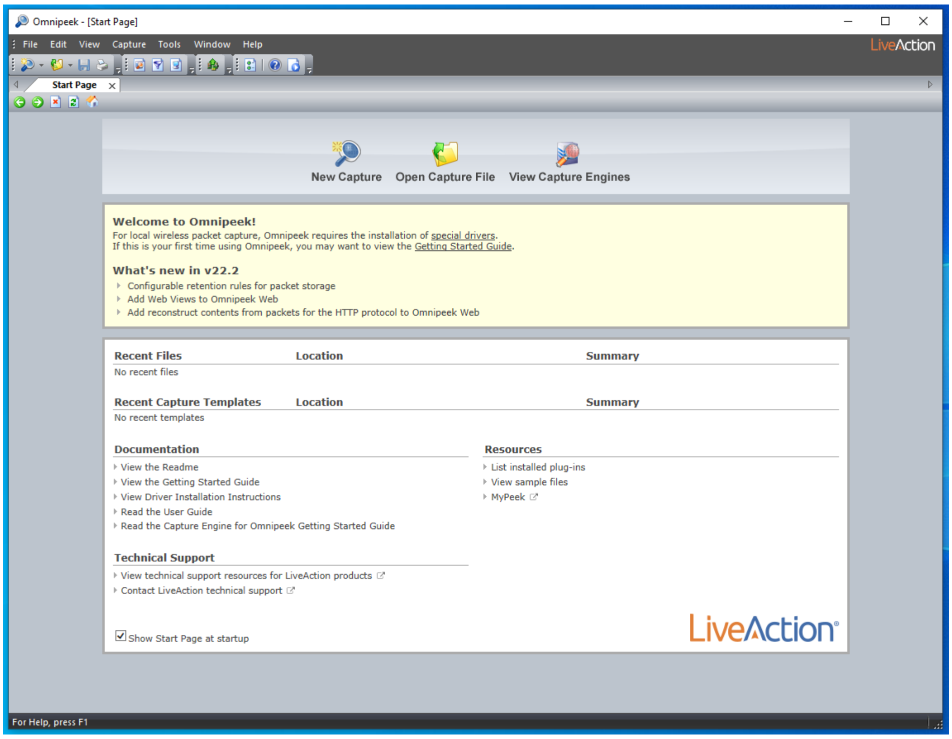
Omnipeek Packet Visualiser allows users to view a multi-segment analysis of their network, showing the speed and state of flow within their network. Omnipeek can also estimate and show the impact of network modifications and adjustments on network performance. This is called the "What-if" view and allows users to know how their network will perform after a change is made before they implement it. (Neox, 2022)

Omnipeek offers useful features such as Network Policy Violation detection, Expert Analysis, and the ability to act on findings during network analysis, without interrupting analysis. The time needed to diagnose and fix issues is decreased by these features. Furthermore, users have access to the Traffic Dashboard, Peer Map and detailed node, protocol, and network maps. (Neox, 2022)

The Peer Map in Omnipeek allows users to highlight nodes of interest and pre-configured or custom alerts can be set to notify important contacts using email, Syslog protocol or SNMP trap if certain conditions are met. Omnipeek can collect data from many networks and features access to Cisco, Aruba, Xirrus, Ruckus, NetFlow, SNMP, and TCP Dump. Capture files can be compared within Omnipeek to efficiently troubleshoot issues on a network. Omnipeek enterprise can be used to gather metrics on voice and video traffic, jitter, packet loss, network delay, signalling, voice, and image quality. (Neox, 2022)

4.2.4 Omnipeek Screenshots

When first opening Omnipeek users are presented with a dashboard showing recent files, capture templates, documentation, technical support, and resources. The top right contains buttons which allow users to start capturing packets or open an existing packet capture file. Capture engines and filters can also be selected here.



Welcome to Omnipeek!
For local wireless packet capture, Omnipeek requires the installation of special drivers.
If this is your first time using Omnipeek, you may want to view the Getting Started Guide.

What's new in v22.2
- Configurable retention rules for packet storage
- Add Web Views to Omnipeek Web
- Add reconstruct contents from packets for the HTTP protocol to Omnipeek Web

Recent Files	Location	Summary
No recent files		

Recent Capture Templates	Location	Summary
No recent templates		

Documentation
- View the Readme
- View the Getting Started Guide
- View Driver Installation Instructions
- Read the User Guide
- Read the Capture Engine for Omnipeek Getting Started Guide

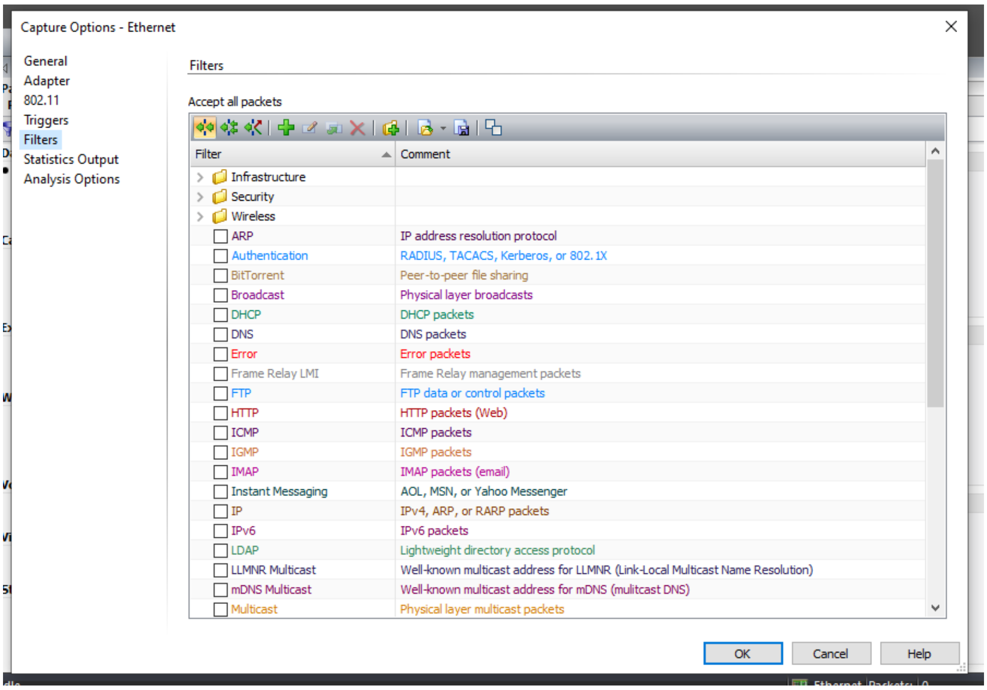
Resources
- List installed plug-ins
- View sample files
- MyPeek

Technical Support
- View technical support resources for LiveAction products
- Contact LiveAction technical support

☑ Show Start Page at startup

Figure 6 Omnipeek trial start page on Windows 10.

Upon starting packet capture, users are given the option to select filters before data is collected.



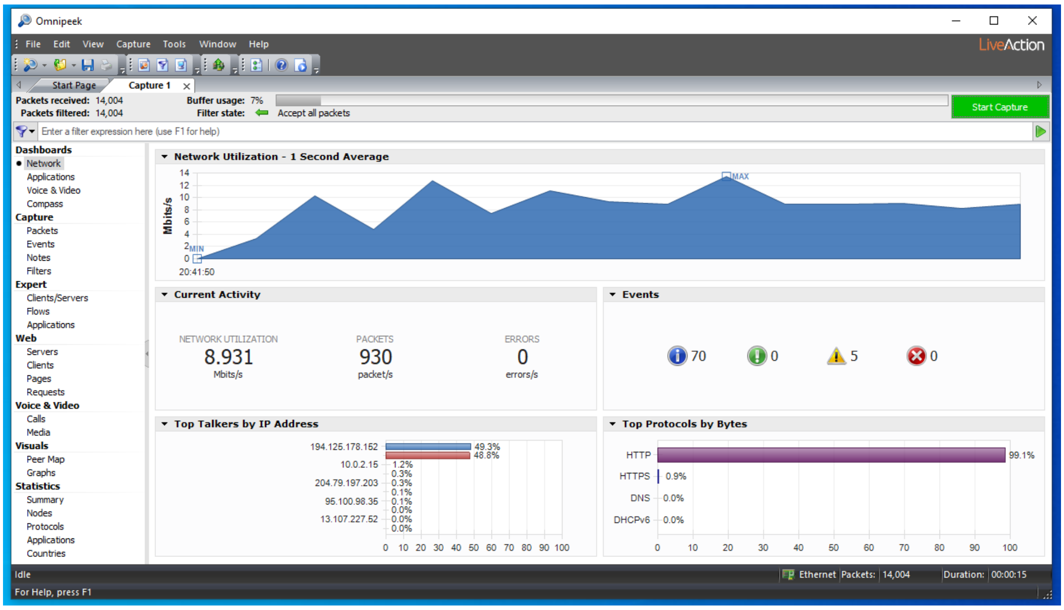
Filters

Accept all packets

Filter	Comment
Infrastructure	
Security	
Wireless	
□ ARP	IP address resolution protocol
□ Authentication	RADIUS, TACACS, Kerberos, or 802.1X
□ BitTorrent	Peer-to-peer file sharing
□ Broadcast	Physical layer broadcasts
□ DHCP	DHCP packets
□ DNS	DNS packets
□ Error	Error packets
□ Frame Relay LMI	Frame Relay management packets
□ FTP	FTP data or control packets
□ HTTP	HTTP packets (Web)
□ ICMP	ICMP packets
□ IGMP	IGMP packets
□ IMAP	IMAP packets (email)
□ Instant Messaging	AOL, MSN, or Yahoo Messenger
□ IP	IPv4, ARP, or RARP packets
□ IPv6	IPv6 packets
□ LDAP	Lightweight directory access protocol
□ LLMNR Multicast	Well-known multicast address for LLMNR (Link-Local Multicast Name Resolution)
□ mDNS Multicast	Well-known multicast address for mDNS (mulitcast DNS)
□ Multicast	Physical layer multicast packets

Figure 7 Some of the filters available on Omnipeek.

After beginning packet capture, telemetry is displayed in real-time to the user. The dashboard available on the right allows users to view statistics such as application response time, packet information, and events.



Packets received: 14,004
Packets filtered: 14,004
Buffer usage: 7%
Filter state: Accept all packets

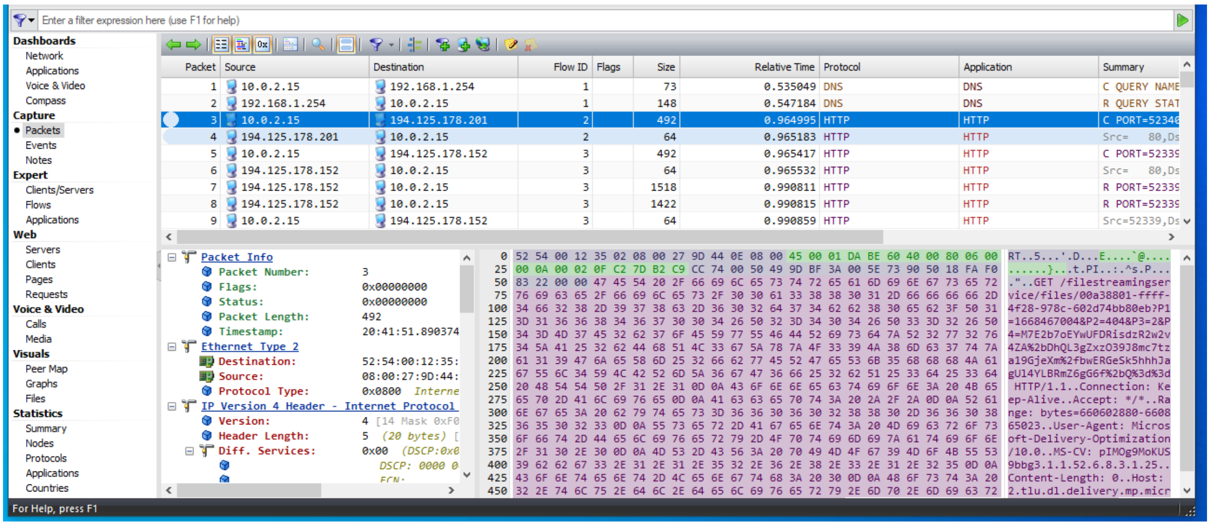
Enter a filter expression here (use F1 for help)

NETWORK UTILIZATION 8.931 Mbits/s
PACKETS 930 packet/s
ERRORS 0 errors/s

Idle
Ethernet Packets: 14,004 Duration: 00:00:15
For Help, press F1

Figure 8 Omnipeek "Network" dashboard.

Changing the dashboard to "Packets" displays the following:



Packet	Source	Destination	Flow ID	Flags	Size	Relative Time	Protocol	Application	Summary
1	10.0.2.15	192.168.1.254	1		73	0.535049	DNS	DNS	C QUERY NAME
2	192.168.1.254	10.0.2.15	1		148	0.547184	DNS	DNS	R QUERY STAT
3	10.0.2.15	194.125.178.201	2		492	0.964995	HTTP	HTTP	C PORT=52340
4	194.125.178.201	10.0.2.15	2		64	0.965183	HTTP	HTTP	Src= 80,Ds
5	10.0.2.15	194.125.178.152	3		492	0.965417	HTTP	HTTP	C PORT=52339
6	194.125.178.152	10.0.2.15	3		64	0.965532	HTTP	HTTP	Src= 80,Ds
7	194.125.178.152	10.0.2.15	3		1518	0.990811	HTTP	HTTP	R PORT=52339
8	194.125.178.152	10.0.2.15	3		1422	0.990815	HTTP	HTTP	R PORT=52339
9	10.0.2.15	194.125.178.152	3		64	0.990859	HTTP	HTTP	Src=52339,Ds

```
⊟ Packet Info
    Packet Number:        3
    Flags:                0x00000000
    Status:               0x00000000
    Packet Length:        492
    Timestamp:            20:41:51.890374
⊟ Ethernet Type 2
    Destination:          52:54:00:12:35:
    Source:               08:00:27:9D:44:
    Protocol Type:        0x0800  Interne
⊟ IP Version 4 Header - Internet Protocol
    Version:              4 [14 Mask 0xF0
    Header Length:        5  (20 bytes) [
  ⊟ Diff. Services:       0x00  (DSCP:0x0
                            DSCP: 0000 0
                            ECN:
```

```
  0 52 54 00 12 35 02 08 00 27 9D 44 0E 08 00 45 00 01 DA BE 60 40 00 80 06 00  RT..5...'.D...E....`@....
 25 00 0A 00 02 0F C2 7D B2 C9 CC 74 00 50 49 9D BF 3A 00 5E 73 90 50 18 FA F0  ......}...t.PI..:.^s.P...
 50 83 22 00 00 47 45 54 20 2F 66 69 6C 65 73 74 72 65 61 6D 69 6E 67 73 65 72  ."..GET /filestreamingser
 75 76 69 63 65 2F 66 69 6C 65 73 2F 30 30 61 33 38 38 30 31 2D 66 66 66 66 2D  vice/files/00a38801-ffff-
100 34 66 32 38 2D 39 37 38 63 2D 36 30 32 64 37 34 62 62 38 30 65 62 3F 50 31  4f28-978c-602d74bb80eb?P1
125 3D 31 36 36 38 34 36 37 30 30 34 26 50 32 3D 34 30 34 26 50 33 3D 32 26 50  =1668467004&P2=404&P3=2&P
150 34 3D 4D 37 45 32 62 37 6F 45 59 77 55 46 44 52 69 73 64 7A 52 32 77 32 76  4=M7E2b7oEYwUFDRisdzR2w2v
175 34 5A 41 25 32 62 44 68 51 4C 33 67 5A 78 7A 4F 33 39 4A 38 6D 63 37 74 7A  4ZA%2bDhQL3gZxzO39J8mc7tz
200 61 31 39 47 6A 65 58 6D 25 32 66 62 77 45 52 47 65 53 6B 35 68 68 68 4A 61  a19GjeXm%2fbwERGeSk5hhhJa
225 67 55 6C 34 59 4C 42 52 6D 5A 36 67 47 36 66 25 32 62 51 25 33 64 25 33 64  gUl4YLBRmZ6gG6f%2bQ%3d%3d
250 20 48 54 54 50 2F 31 2E 31 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65   HTTP/1.1..Connection: Ke
275 65 70 2D 41 6C 69 76 65 0D 0A 41 63 63 65 70 74 3A 20 2A 2F 2A 0D 0A 52 61  ep-Alive..Accept: */*..Ra
300 6E 67 65 3A 20 62 79 74 65 73 3D 36 36 30 36 30 32 38 38 30 2D 36 36 30 38  nge: bytes=660602880-6608
325 36 35 30 32 33 0D 0A 55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 69 63 72 6F 73  65023..User-Agent: Micros
350 6F 66 74 2D 44 65 6C 69 76 65 72 79 2D 4F 70 74 69 6D 69 7A 61 74 69 6F 6E  oft-Delivery-Optimization
375 2F 31 30 2E 30 0D 0A 4D 53 2D 43 56 3A 20 70 49 4D 4F 67 39 4D 6F 4B 55 53  /10.0..MS-CV: pIMOg9MoKUS
400 39 62 62 67 33 2E 31 2E 31 2E 35 32 2E 36 2E 38 2E 33 2E 31 2E 32 35 0D 0A  9bbg3.1.1.52.6.8.3.1.25..
425 43 6F 6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 30 0D 0A 48 6F 73 74 3A 20  Content-Length: 0..Host: 
450 32 2E 74 6C 75 2E 64 6C 2E 64 65 6C 69 76 65 72 79 2E 6D 70 2E 6D 69 63 72  2.tlu.dl.delivery.mp.micr
```

Figure 9 Omnipeek "Packets" dashboard.

4.3 Wireshark

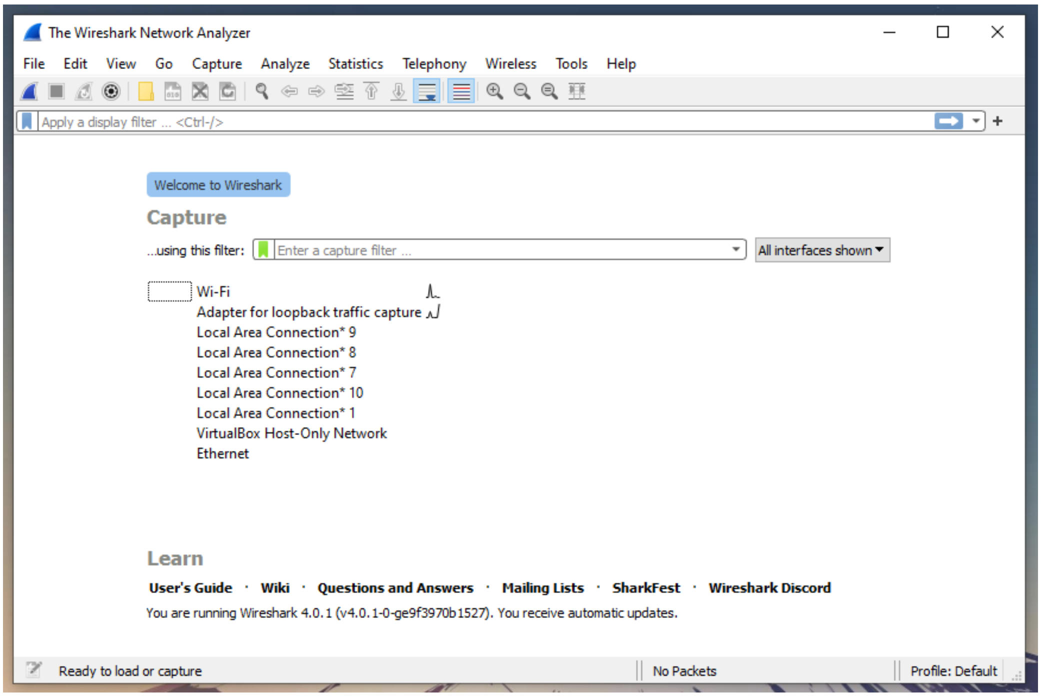
Requirements: Unix, Linux, and Microsoft Windows

Wireshark is a free and open-source project released under GPL. It was written using C. It is very popular and considered to be one of the best packet sniffers/analysers available today. It is available for UNIX and Windows and contains features such as filter criteria, colourized packet display, protocol dissectors, the ability to save and import packet data in many formats and view packets with detailed protocol information. Wireshark can capture data from media types such as Ethernet, Wireless LAN, Bluetooth, and USB. Wireshark also has protocol dissectors. The intended purposes of Wireshark include education, development, quality assurance, troubleshooting, and security. Wireshark is not designed to be used as a network IDS. It cannot manipulate the network, and instead only listens to it. Wireshark can't send packets onto a network except for domain name resolution. (Wireshark, 2022).

When utilizing Wireshark on networks with encrypted traffic, decryption support is offered for protocols such as IPsec, ISAKMP and WPA2. Various capture file formats supported by Wireshark include tcpdump (libpcap), Cisco Secure IDS iplog, Microsoft Network Monitor, Sniffer Pro, and plain text files. Two versions of the application are available. These are TShark, which provides a command line interface and the more popular version, Wireshark which features a GUI. Today, an active community of developers, educators and users support the tool. Furthermore, things such as an enforced code of conduct and support from corporate sponsors have allowed Wireshark to remain relevant and become one of the best packet sniffers available despite being a 24-year-old tool. (Breeden II, 2022)

4.3.1 Wireshark Screenshots

Below are screenshots demonstrating the use of Wireshark on Windows 10.


The Wireshark Network Analyzer
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark
Capture
All interfaces shown
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists · SharkFest · Wireshark Discord

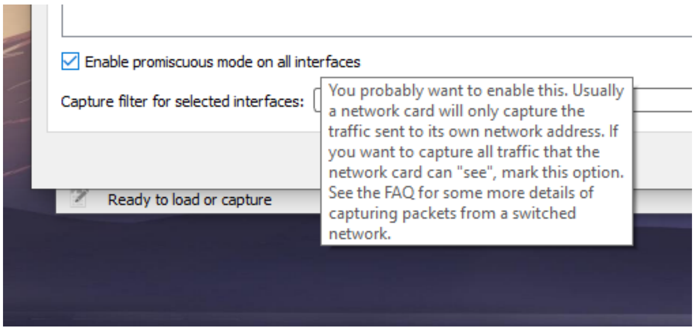
...using this filter: Enter a capture filter ...

Wi-Fi
Adapter for loopback traffic capture
Local Area Connection* 9
Local Area Connection* 8
Local Area Connection* 7
Local Area Connection* 10
Local Area Connection* 1
VirtualBox Host-Only Network
Ethernet

You are running Wireshark 4.0.1 (v4.0.1-0-ge9f3970b1527). You receive automatic updates.

Ready to load or capture | No Packets | Profile: Default

Figure 10 When first opening Wireshark, users are presented with a list of interfaces that they can start collecting packets from.


☑ Enable promiscuous mode on all interfaces

Capture filter for selected interfaces:

You probably want to enable this. Usually a network card will only capture the traffic sent to its own network address. If you want to capture all traffic that the network card can "see", mark this option. See the FAQ for some more details of capturing packets from a switched network.

Ready to load or capture

Figure 11 Enabling promiscuous mode on Wireshark. This is done before packet capturing starts.

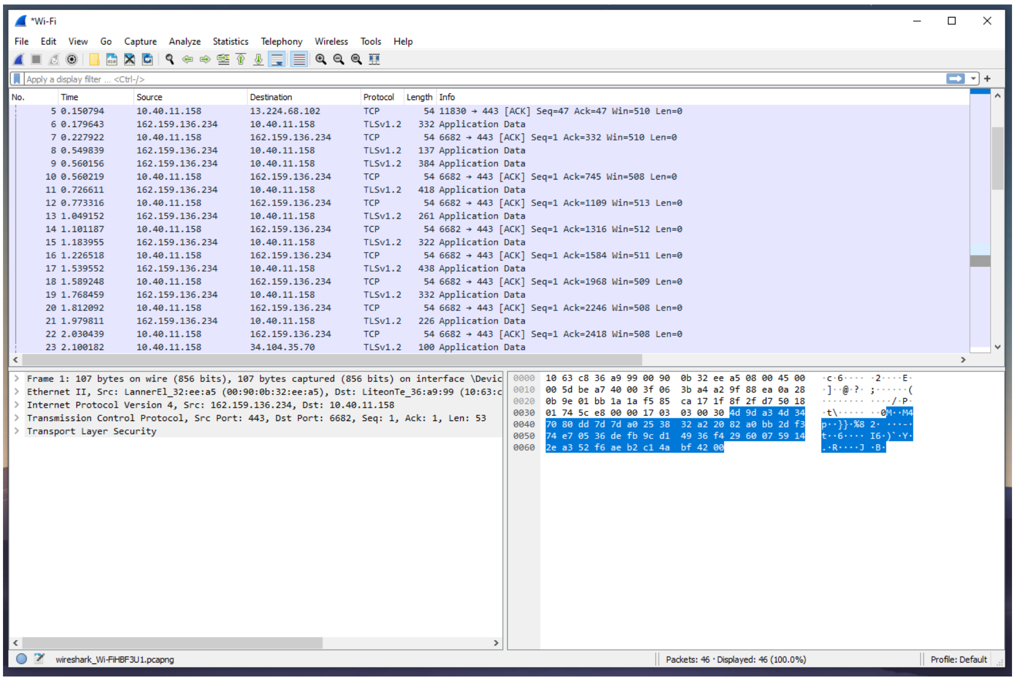
After selecting an interface and starting the packet capture process, Wireshark will display data in real-time to the user. The three panes as shown in the screenshot below allow you to inspect data in various ways. If a packet is part of a conversation, a user can select that packet and follow only the packets part of that conversation.

Top pane: This is the Packet List pane. Clicking a listing from this pane will result in the bottom panes displaying details to represent the selected packet. Information like source, destination and length of the packet can be found on this page. It is updated in real time when packets are being captured.

Bottom left pane: This is the Packet Details pane. This page displays readable information about the packet.

Bottom right pane: This is the Packet Byte pane. This displays the packet exactly as it was captured in hexadecimal.

(Kinzie, 2022)


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time      Source           Destination      Protocol Length Info
    5 0.150794  10.40.11.158     13.224.68.102    TCP        54 11830 → 443 [ACK] Seq=47 Ack=47 Win=510 Len=0
    6 0.179643  162.159.136.234  10.40.11.158     TLSv1.2   332 Application Data
    7 0.227922  10.40.11.158     162.159.136.234  TCP        54 6682 → 443 [ACK] Seq=1 Ack=332 Win=510 Len=0
    8 0.549839  162.159.136.234  10.40.11.158     TLSv1.2   137 Application Data
    9 0.560156  162.159.136.234  10.40.11.158     TLSv1.2   384 Application Data
   10 0.560219  10.40.11.158     162.159.136.234  TCP        54 6682 → 443 [ACK] Seq=1 Ack=745 Win=508 Len=0
   11 0.726611  162.159.136.234  10.40.11.158     TLSv1.2   418 Application Data
   12 0.773316  10.40.11.158     162.159.136.234  TCP        54 6682 → 443 [ACK] Seq=1 Ack=1109 Win=513 Len=0
   13 1.049152  162.159.136.234  10.40.11.158     TLSv1.2   261 Application Data
   14 1.101187  10.40.11.158     162.159.136.234  TCP        54 6682 → 443 [ACK] Seq=1 Ack=1316 Win=512 Len=0
   15 1.183955  162.159.136.234  10.40.11.158     TLSv1.2   322 Application Data
   16 1.226518  10.40.11.158     162.159.136.234  TCP        54 6682 → 443 [ACK] Seq=1 Ack=1584 Win=511 Len=0
   17 1.539552  162.159.136.234  10.40.11.158     TLSv1.2   438 Application Data
   18 1.589248  10.40.11.158     162.159.136.234  TCP        54 6682 → 443 [ACK] Seq=1 Ack=1968 Win=509 Len=0
   19 1.768459  162.159.136.234  10.40.11.158     TLSv1.2   332 Application Data
   20 1.812092  10.40.11.158     162.159.136.234  TCP        54 6682 → 443 [ACK] Seq=1 Ack=2246 Win=508 Len=0
   21 1.979811  162.159.136.234  10.40.11.158     TLSv1.2   226 Application Data
   22 2.030439  10.40.11.158     162.159.136.234  TCP        54 6682 → 443 [ACK] Seq=1 Ack=2418 Win=508 Len=0
   23 2.100182  10.40.11.158     34.104.35.70     TLSv1.2   100 Application Data
```

```
> Frame 1: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface \Devic
> Ethernet II, Src: LannerEl_32:ee:a5 (00:90:0b:32:ee:a5), Dst: LiteonTe_36:a9:99 (10:63:c
> Internet Protocol Version 4, Src: 162.159.136.234, Dst: 10.40.11.158
> Transmission Control Protocol, Src Port: 443, Dst Port: 6682, Seq: 1, Ack: 1, Len: 53
> Transport Layer Security
```

```
0000  10 63 c8 36 a9 99 00 90  0b 32 ee a5 08 00 45 00   ·c·6····  ·2····E·
0010  00 5d be a7 40 00 3f 06  3b a4 a2 9f 88 ea 0a 28   ·]··@·?·  ;······(
0020  0b 9e 01 bb 1a 1a f5 85  ca 17 1f 8f 2f d7 50 18   ········  ····/·P·
0030  01 74 5c e8 00 00 17 03  03 00 30 4d 9d a3 4d 34   ·t\·····  ··0M··M4
0040  70 80 dd 7d 7d a0 25 38  32 a2 20 82 a0 bb 2d f3   p··}}·%8  2· ····-·
0050  74 e7 05 36 de fb 9c d1  49 36 f4 29 60 07 59 14   t··6····  I6·)`·Y·
0060  2e a3 52 f6 ae b2 c1 4a  bf 42 00                  .·R····J  ·B·
```

wireshark_Wi-FiHBF3U1.pcapng | Packets: 46 · Displayed: 46 (100.0%) | Profile: Default

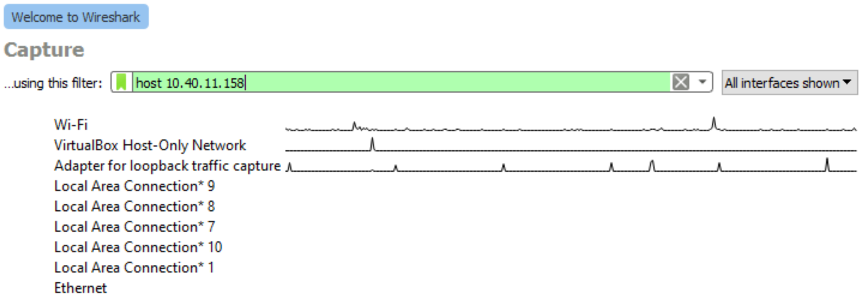
Figure 12 Packet List, Packet Details and Packet Bytes pane on Wireshark.

4.3.2 Wireshark filters

Wireshark won't save packets that do not match specified filters. (Kinzie, 2022) provides the following filters as examples of how they might be formatted:

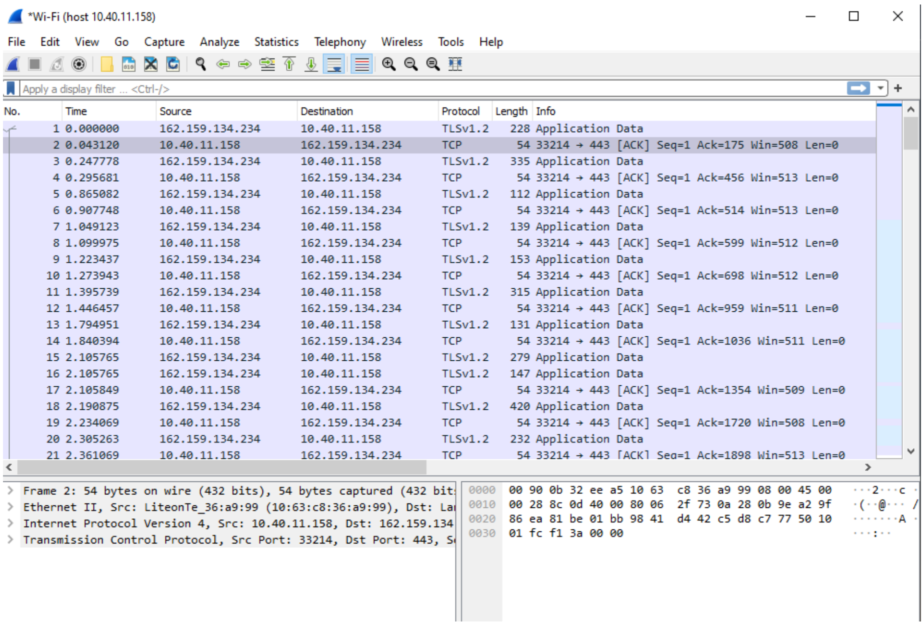
Host IP-address:	Capture traffic to and from this IP address
Net 192.168.0.0/24:	Capture traffic on this subnet
Dst host IP-address:	Capture packets being sent to this host
Port 53:	Capture traffic on port 53 only
Port not 53 and not arp:	Capture all traffic except DNS and ARP traffic

An example of a filter being used is highlighted in green in the screenshot below:



...using this filter: host 10.40.11.158

Figure 13 Using the "host" filter and enter the IP address of the machine running Wireshark.



```
No.  Time        Source           Destination      Protocol  Length Info
   1 0.000000    162.159.134.234  10.40.11.158     TLSv1.2   228 Application Data
   2 0.043120    10.40.11.158     162.159.134.234  TCP        54 33214 → 443 [ACK] Seq=1 Ack=175 Win=508 Len=0
   3 0.247778    162.159.134.234  10.40.11.158     TLSv1.2   335 Application Data
   4 0.295681    10.40.11.158     162.159.134.234  TCP        54 33214 → 443 [ACK] Seq=1 Ack=456 Win=513 Len=0
   5 0.865082    162.159.134.234  10.40.11.158     TLSv1.2   112 Application Data
   6 0.907748    10.40.11.158     162.159.134.234  TCP        54 33214 → 443 [ACK] Seq=1 Ack=514 Win=513 Len=0
   7 1.049123    162.159.134.234  10.40.11.158     TLSv1.2   139 Application Data
   8 1.099975    10.40.11.158     162.159.134.234  TCP        54 33214 → 443 [ACK] Seq=1 Ack=599 Win=512 Len=0
   9 1.223437    162.159.134.234  10.40.11.158     TLSv1.2   153 Application Data
  10 1.273943    10.40.11.158     162.159.134.234  TCP        54 33214 → 443 [ACK] Seq=1 Ack=698 Win=512 Len=0
  11 1.395739    162.159.134.234  10.40.11.158     TLSv1.2   315 Application Data
  12 1.446457    10.40.11.158     162.159.134.234  TCP        54 33214 → 443 [ACK] Seq=1 Ack=959 Win=511 Len=0
  13 1.794951    162.159.134.234  10.40.11.158     TLSv1.2   131 Application Data
  14 1.840394    10.40.11.158     162.159.134.234  TCP        54 33214 → 443 [ACK] Seq=1 Ack=1036 Win=511 Len=0
  15 2.105765    162.159.134.234  10.40.11.158     TLSv1.2   279 Application Data
  16 2.105765    162.159.134.234  10.40.11.158     TLSv1.2   147 Application Data
  17 2.105849    10.40.11.158     162.159.134.234  TCP        54 33214 → 443 [ACK] Seq=1 Ack=1354 Win=509 Len=0
  18 2.190875    162.159.134.234  10.40.11.158     TLSv1.2   420 Application Data
  19 2.234069    10.40.11.158     162.159.134.234  TCP        54 33214 → 443 [ACK] Seq=1 Ack=1720 Win=508 Len=0
  20 2.305263    162.159.134.234  10.40.11.158     TLSv1.2   232 Application Data
  21 2.361069    10.40.11.158     162.159.134.234  TCP        54 33214 → 443 [ACK] Seq=1 Ack=1898 Win=513 Len=0
```

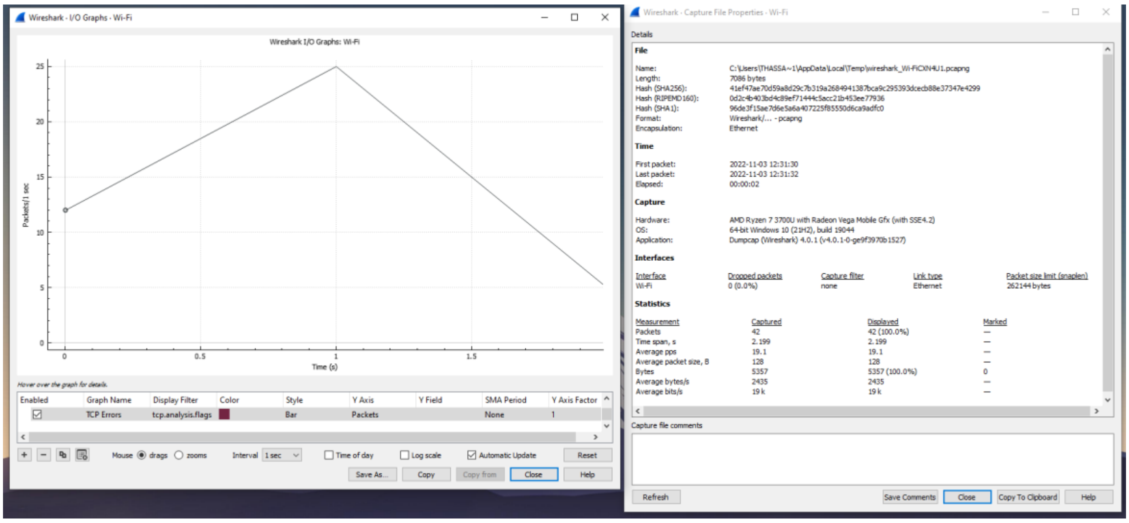
```
> Frame 2: 54 bytes on wire (432 bits), 54 bytes captured (432 bit
> Ethernet II, Src: LiteonTe_36:a9:99 (10:63:c8:36:a9:99), Dst: La
> Internet Protocol Version 4, Src: 10.40.11.158, Dst: 162.159.134
> Transmission Control Protocol, Src Port: 33214, Dst Port: 443, S
```

```
0000  00 90 0b 32 ee a5 10 63  c8 36 a9 99 08 00 45 00   ···2···c ·
0010  00 28 8c 0d 40 00 80 06  2f 73 0a 28 0b 9e a2 9f   ·(··@··· /
0020  86 ea 81 be 01 bb 98 41  d4 42 c5 d8 c7 77 50 10   ·······A ·
0030  01 fc f1 3a 00 00                                  ···:··
```

Figure 14 The result of using the filter is that only data sent to and from the machine running Wireshark are displayed. (10.40.11.158).

After packet capture, display filters can be used to narrow down search results and a user can reveal only what is relevant to them from the data captured.

Statistics about the capture are also available in the statistics menu. Examples of an I/O graph and file properties screen are shown below.


Wireshark · I/O Graphs · Wi-Fi
Wireshark I/O Graphs: Wi-Fi
Packets/1 sec
25
20
15
10
5
0
0
0.5
1
1.5
Time (s)
Hover over the graph for details.

Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Period	Y Axis Factor
☑	TCP Errors	tcp.analysis.flags	■	Bar	Packets		None	1

Mouse ● drags ○ zooms Interval 1 sec □ Time of day □ Log scale ☑ Automatic Update

+
−
Reset
Save As...
Copy
Copy from
Close
Help
Wireshark · Capture File Properties · Wi-Fi
Details

File

Name:	C:\Users\THASSA~1\AppData\Local\Temp\wireshark_Wi-FiCXN4U1.pcapng
Length:	7086 bytes
Hash (SHA256):	41ef47ae70d59a8d29c7b319a2684941387bca9c295393dcecb88e37347e4299
Hash (RIPEMD160):	0d2c4b403bd4c89ef71444c5acc21b453ee77936
Hash (SHA1):	96de3f15ae7d6e5a6a407225f85550d6ca9adfc0
Format:	Wireshark/... - pcapng
Encapsulation:	Ethernet

Time

First packet:	2022-11-03 12:31:30
Last packet:	2022-11-03 12:31:32
Elapsed:	00:00:02

Capture

Hardware:	AMD Ryzen 7 3700U with Radeon Vega Mobile Gfx (with SSE4.2)
OS:	64-bit Windows 10 (21H2), build 19044
Application:	Dumpcap (Wireshark) 4.0.1 (v4.0.1-0-ge9f3970b1527)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit (snaplen)
Wi-Fi	0 (0.0%)	none	Ethernet	262144 bytes

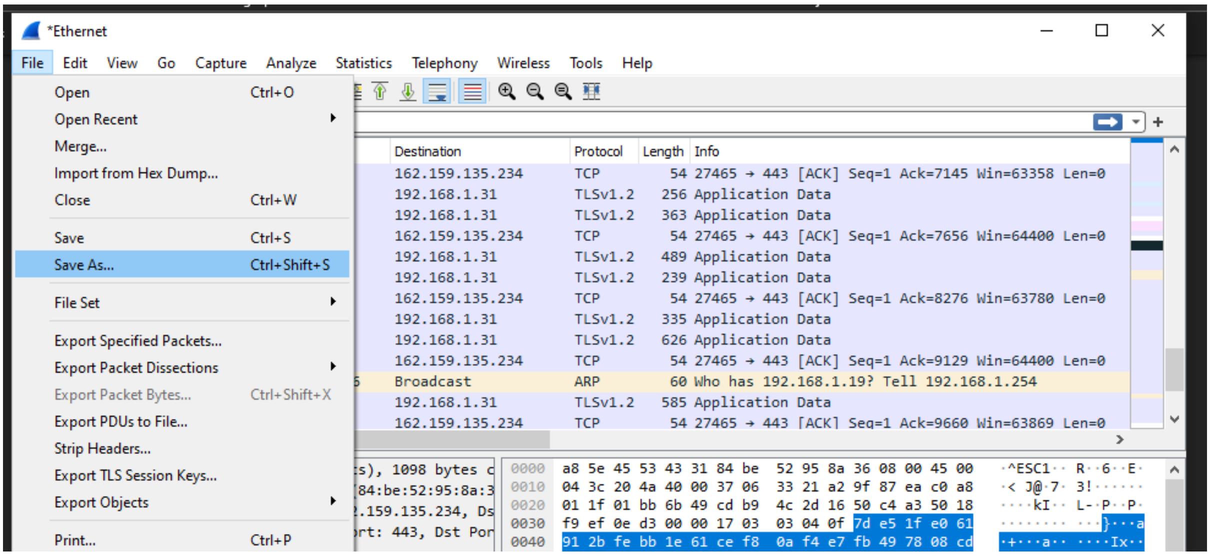
Statistics

Measurement	Captured	Displayed	Marked
Packets	42	42 (100.0%)	—
Time span, s	2.199	2.199	—
Average pps	19.1	19.1	—
Average packet size, B	128	128	—
Bytes	5357	5357 (100.0%)	0
Average bytes/s	2435	2435	—
Average bits/s	19 k	19 k	—

Figure 15 Wireshark capture statistics

4.3.3 Saving Network Data on Wireshark

Formats such as .pcap and .pcapng are available to users when saving data captured by Wireshark.



```
Destination       Protocol  Length Info
162.159.135.234   TCP         54 27465 → 443 [ACK] Seq=1 Ack=7145 Win=63358 Len=0
192.168.1.31      TLSv1.2    256 Application Data
192.168.1.31      TLSv1.2    363 Application Data
162.159.135.234   TCP         54 27465 → 443 [ACK] Seq=1 Ack=7656 Win=64400 Len=0
192.168.1.31      TLSv1.2    489 Application Data
192.168.1.31      TLSv1.2    239 Application Data
162.159.135.234   TCP         54 27465 → 443 [ACK] Seq=1 Ack=8276 Win=63780 Len=0
192.168.1.31      TLSv1.2    335 Application Data
192.168.1.31      TLSv1.2    626 Application Data
162.159.135.234   TCP         54 27465 → 443 [ACK] Seq=1 Ack=9129 Win=64400 Len=0
Broadcast         ARP         60 Who has 192.168.1.19? Tell 192.168.1.254
192.168.1.31      TLSv1.2    585 Application Data
162.159.135.234   TCP         54 27465 → 443 [ACK] Seq=1 Ack=9660 Win=63869 Len=0
```

```
:s), 1098 bytes c
[84:be:52:95:8a:3
.159.135.234, Ds
rt: 443, Dst Por
```

```
0000  a8 5e 45 53 43 31 84 be  52 95 8a 36 08 00 45 00   ·^ESC1·· R··6··E·
0010  04 3c 20 4a 40 00 37 06  33 21 a2 9f 87 ea c0 a8   ·< J@·7· 3!······
0020  01 1f 01 bb 6b 49 cd b9  4c 2d 16 50 c4 a3 50 18   ····kI·· L-·P··P·
0030  f9 ef 0e d3 00 00 17 03  03 04 0f 7d e5 1f e0 61   ········ ···}···a
0040  91 2b fe bb 1e 61 ce f8  0a f4 e7 fb 49 78 08 cd   ·+···a·· ····Ix··
```

Figure 16 After packet capture is paused, users can click save as to choose the file format they want to use.

4.4 SolarWinds Network Performance Monitor

Requirements: Windows Server 2016, Windows Server 2019, Microsoft Windows 10.

4.4.1 The Network Performance Monitor

A packet sniffer is one of many components found within the SolarWinds Network Performance Monitor (NPM). (SolarWinds, 2022) describes the application to provide "multi-vendor network monitoring built to scale and expand with the needs of your network.". There are a huge number of features available to users with some being Intelligent mapping, performance analysis, critical path visualisation, network availability monitoring, and PCAP files. (SolarWinds, 2022)

4.4.2 NPM Packet Sniffer

With the packet sniffer found in SolarWinds NPM, users can identify performance issues in their network and discover what is causing them. Path latency, response time, types of traffic, and bottlenecks in a network can be identified by SolarWinds NPM. Administrators are provided with the relevant metrics that can allow them to diagnose problems. Out of the box, SolarWinds NPM can analyse packets associated with many applications with examples being Dropbox, FTP, HTTP, Skype, and Amazon Web Services. SolarWinds NPM collects application meta values rather than full packet data, preventing unnecessary traffic data from taking up database storage as well as allowing NPM to run without a high load. NPM allows users to drill down into devices and applications. (Solarwinds, 2022)

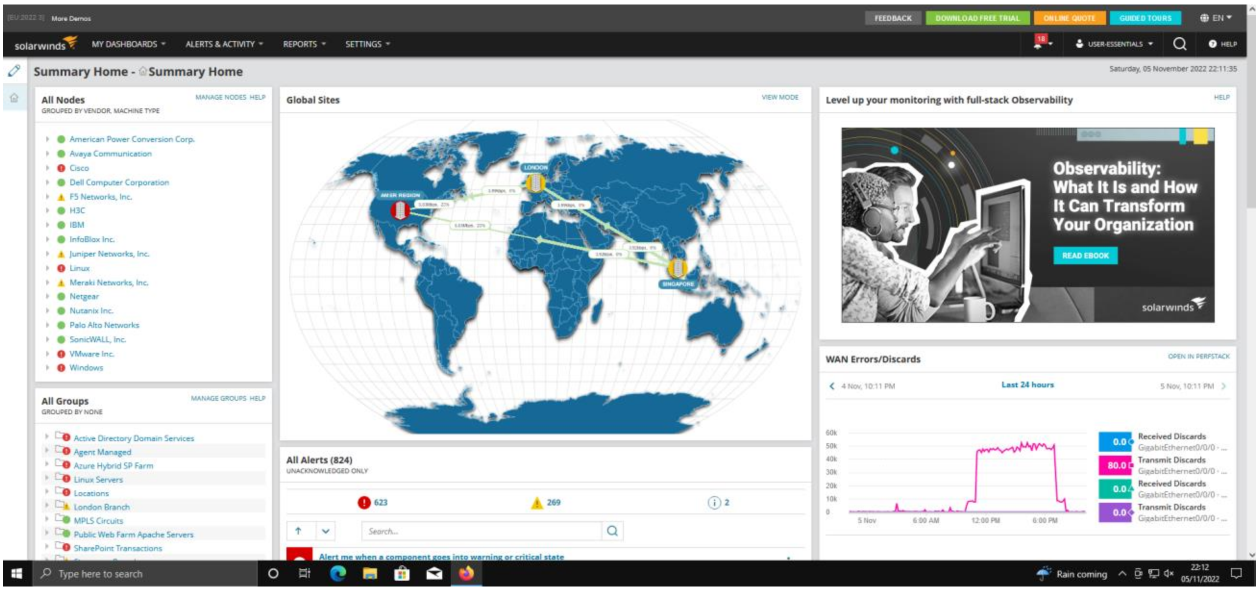
SolarWinds NPM categorizes the network traffic that it captures and estimates a risk level with this traffic. This helps administrators get an in-depth insight into their network. SolarWinds NPM provides a centralized dashboard with a user-friendly interface. From this interface, administrators can investigate devices and applications. Charts and graphs are provided to allow administrators to get a quick visualized overview of their network. Custom packet scan alerts can allow administrators to be notified of issues immediately, giving them more time for incident response and preventing security risks. SolarWinds NPM utilizes email alerts and SMS alerts. (SolarWinds, 2022)

Use of SolarWinds Network Performance Monitor is free for 30 days. Users would be required to request a quote from SolarWinds to continue to use this product after the trial. A subscription starts at €1,335. The high price and in-depth analysis provided by this tool make it unsuitable for non-tech-savvy people or people using a packet sniffer for educational purposes.

SolarWinds NPM utilises Winpcap and is suitable only for large networks that require frequent monitoring and analysis. It is designed for managing a full network, unlike Wireshark or tcpdump which only analyses packet data. (Keary, 2022)

4.4.3 Solar Winds NPM Screenshots

Below is a screenshot of the demonstration provided by SolarWinds.



Summary Home - Summary Home

Saturday, 05 November 2022 22:11:35

All Nodes MANAGE NODES HELP
GROUPED BY VENDOR, MACHINE TYPE

- American Power Conversion Corp.
- Avaya Communication
- Cisco
- Dell Computer Corporation
- F5 Networks, Inc.
- H3C
- IBM
- InfoBlox Inc.
- Juniper Networks, Inc.
- Linux
- Meraki Networks, Inc.
- Netgear
- Nutanix Inc.
- Palo Alto Networks
- SonicWALL, Inc.
- VMware Inc.
- Windows

All Groups MANAGE GROUPS HELP
GROUPED BY NONE

- Active Directory Domain Services
- Agent Managed
- Azure Hybrid SP Farm
- Linux Servers
- Locations
- London Branch
- MPLS Circuits
- Public Web Farm Apache Servers
- SharePoint Transactions

Global Sites VIEW MODE

All Alerts (824)
UNACKNOWLEDGED ONLY

623 269 2

Search...

Alert me when a component goes into warning or critical state

Level up your monitoring with full-stack Observability HELP

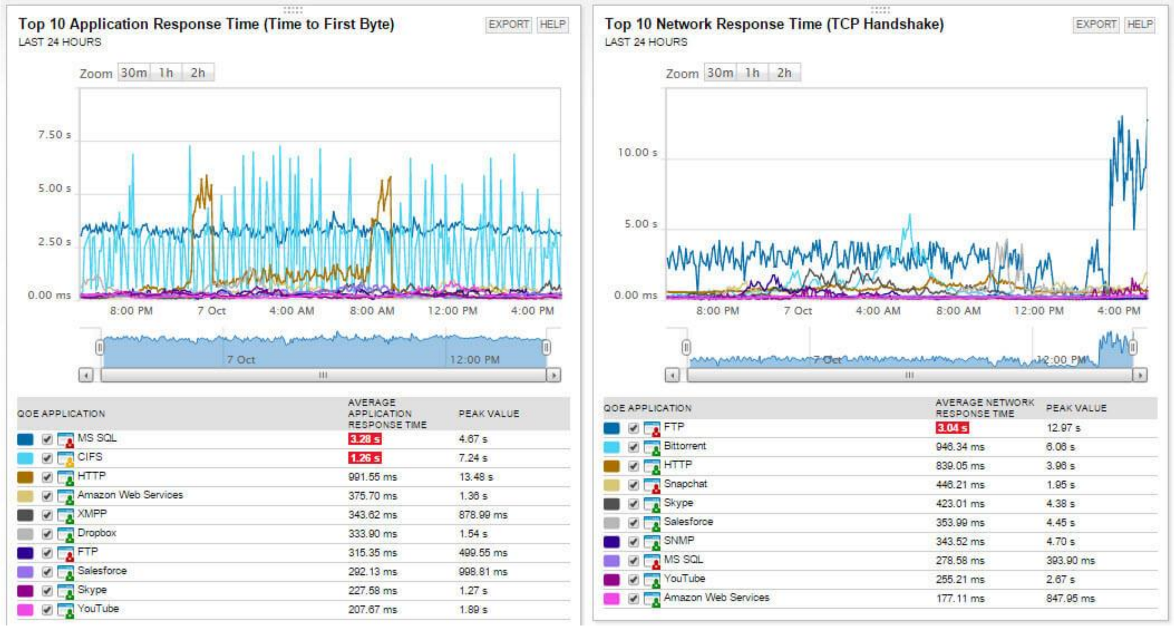
WAN Errors/Discards OPEN IN PERFSTACK

4 Nov, 10:11 PM Last 24 hours 5 Nov, 10:11 PM

0.0	Received Discards GigabitEthernet0/0/0 - ...
80.0	Transmit Discards GigabitEthernet0/0/0 - ...
0.0	Received Discards GigabitEthernet0/0/0 - ...
0.0	Transmit Discards GigabitEthernet0/0/0 - ...

Figure 17 SolarWinds Home Dashboard

As seen below, the SolarWinds Network Performance Monitor provides a high-level user-friendly interface for viewing packets and their associated applications. This is like the "High-Level Packet Sniffer" final year project.



Top 10 Application Response Time (Time to First Byte)
LAST 24 HOURS

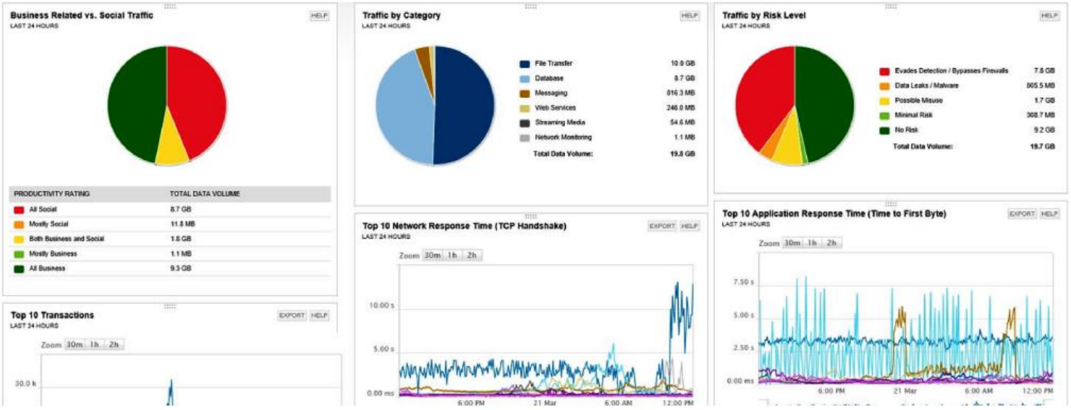
QOE APPLICATION	AVERAGE APPLICATION RESPONSE TIME	PEAK VALUE
■ ☑ MS SQL	3.28 s	4.67 s
■ ☑ CIFS	1.26 s	7.24 s
■ ☑ HTTP	991.55 ms	13.48 s
■ ☑ Amazon Web Services	375.70 ms	1.36 s
■ ☑ XMPP	343.62 ms	878.99 ms
■ ☑ Dropbox	333.90 ms	1.54 s
■ ☑ FTP	315.35 ms	499.55 ms
■ ☑ Salesforce	292.13 ms	998.81 ms
■ ☑ Skype	227.58 ms	1.27 s
■ ☑ YouTube	207.67 ms	1.89 s

Top 10 Network Response Time (TCP Handshake)
LAST 24 HOURS

QOE APPLICATION	AVERAGE NETWORK RESPONSE TIME	PEAK VALUE
■ ☑ FTP	3.04 s	12.97 s
■ ☑ Bittorrent	946.34 ms	6.06 s
■ ☑ HTTP	839.05 ms	3.96 s
■ ☑ Snapchat	446.21 ms	1.95 s
■ ☑ Skype	423.01 ms	4.38 s
■ ☑ Salesforce	353.99 ms	4.45 s
■ ☑ SNMP	343.52 ms	4.70 s
■ ☑ MS SQL	278.58 ms	393.90 ms
■ ☑ YouTube	255.21 ms	2.67 s
■ ☑ Amazon Web Services	177.11 ms	847.95 ms

Figure 18 (SolarWinds, 2022) "Use a network packet sniffer to isolate performance issues and drill down on root causes." [Online] Available at: https://www.solarwinds.com/network-performance-monitor/use-cases/packet-sniffer [Accessed 19/11/2022]

"Deep packet inspection" lets administrators examine packets on a granular level. Smart charts are used to display information on packet metadata. NPM uses metadata from packets instead of simulated estimations to calculate end-user experience.



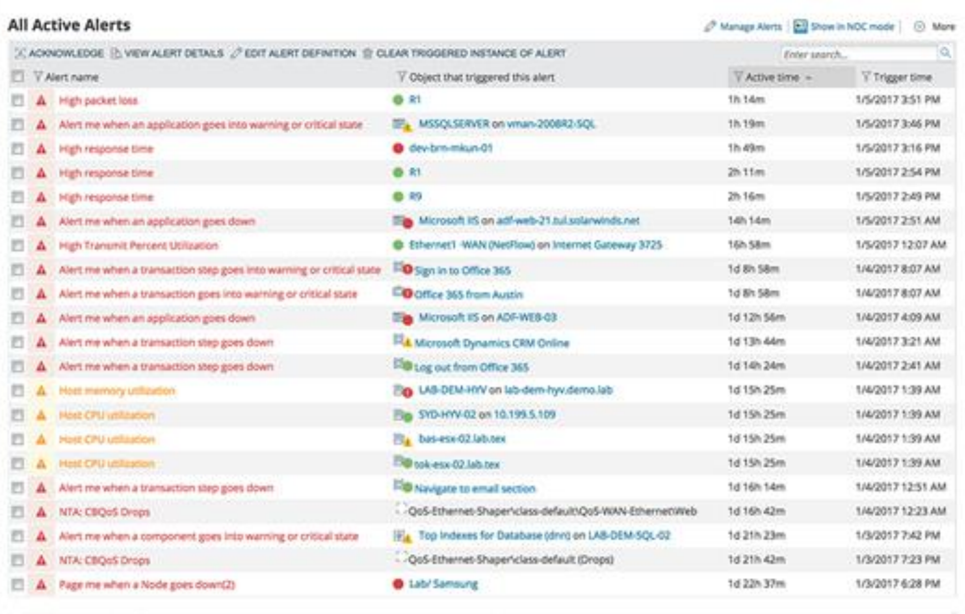
PRODUCTIVITY RATING	TOTAL DATA VOLUME
🟥 All Social	8.7 GB
🟧 Mostly Social	11.8 MB
🟨 Both Business and Social	1.8 GB
🟩 Mostly Business	1.1 MB
🟩 All Business	9.3 GB

🟦 File Transfer	10.0 GB
🟦 Database	8.7 GB
🟫 Messaging	816.3 MB
🟨 Web Services	246.0 MB
⬛ Streaming Media	54.6 MB
⬜ Network Monitoring	1.1 MB
Total Data Volume:	19.8 GB

🟥 Evades Detection / Bypasses Firewalls	7.8 GB
🟧 Data Leaks / Malware	805.5 MB
🟨 Possible Misuse	1.7 GB
🟩 Minimal Risk	308.7 MB
🟩 No Risk	9.2 GB
Total Data Volume:	19.7 GB

Figure 19 (SolarWinds, 2022) "Dig deeper into packet analysis and gain insights with a nuanced network packet sniffer" [Online] Available at: https://www.solarwinds.com/network-performance-monitor/use-cases/packet-sniffer [Accessed 19/11/2022]

This screenshot displays some of the alerts which users can configure using SolarWinds NPM. Users can receive alerts only on incidents that they're concerned about, avoiding the influx of too many irrelevant alerts in their inboxes.



☐	Alert name	Object that triggered this alert	Active time	Trigger time
☐	High packet loss	R1	1h 14m	1/5/2017 3:51 PM
☐	Alert me when an application goes into warning or critical state	MSSQLSERVER on vman-2008R2-SQL	1h 19m	1/5/2017 3:46 PM
☐	High response time	dev-brn-mkun-01	1h 49m	1/5/2017 3:16 PM
☐	High response time	R1	2h 11m	1/5/2017 2:54 PM
☐	High response time	R9	2h 16m	1/5/2017 2:49 PM
☐	Alert me when an application goes down	Microsoft IIS on adf-web-21.tul.solarwinds.net	14h 14m	1/5/2017 2:51 AM
☐	High Transmit Percent Utilization	Ethernet1 · WAN (NetFlow) on Internet Gateway 3725	16h 58m	1/5/2017 12:07 AM
☐	Alert me when a transaction step goes into warning or critical state	Sign in to Office 365	1d 8h 58m	1/4/2017 8:07 AM
☐	Alert me when a transaction goes into warning or critical state	Office 365 from Austin	1d 8h 58m	1/4/2017 8:07 AM
☐	Alert me when an application goes down	Microsoft IIS on ADF-WEB-03	1d 12h 56m	1/4/2017 4:09 AM
☐	Alert me when a transaction step goes down	Microsoft Dynamics CRM Online	1d 13h 44m	1/4/2017 3:21 AM
☐	Alert me when a transaction step goes down	Log out from Office 365	1d 14h 24m	1/4/2017 2:41 AM
☐	Host memory utilization	LAB-DEM-HYV on lab-dem-hyv.demo.lab	1d 15h 25m	1/4/2017 1:39 AM
☐	Host CPU utilization	SYD-HYV-02 on 10.199.5.109	1d 15h 25m	1/4/2017 1:39 AM
☐	Host CPU utilization	bas-esx-02.lab.tex	1d 15h 25m	1/4/2017 1:39 AM
☐	Host CPU utilization	tok-esx-02.lab.tex	1d 15h 25m	1/4/2017 1:39 AM
☐	Alert me when a transaction step goes down	Navigate to email section	1d 16h 14m	1/4/2017 12:51 AM
☐	NTA: CBQoS Drops	QoS-Ethernet-Shaper\class-default\QoS-WAN-Ethernet\Web	1d 16h 42m	1/4/2017 12:23 AM
☐	Alert me when a component goes into warning or critical state	Top Indexes for Database (dnn) on LAB-DEM-SQL-02	1d 21h 23m	1/3/2017 7:42 PM
☐	NTA: CBQoS Drops	QoS-Ethernet-Shaper\class-default (Drops)	1d 21h 42m	1/3/2017 7:23 PM
☐	Page me when a Node goes down(2)	Lab/ Samsung	1d 22h 37m	1/3/2017 6:28 PM

Figure 20 (SolarWinds, 2022) "Set custom, automated packet scanner alerts" [Online] Available at: https://www.solarwinds.com/network-performance-monitor/use-cases/packet-analyzer [Accessed 19/11/2022]

5 Programming Languages and Libraries

5.1 Python

Python was created in 1991 by programmer Guido Van Rossum. Van Rossum wishes for Python to be understandable and clear as plain English. It is an open-source language. Pythons' accessibility has allowed it to become popular and versatile. It is among the top five programming languages in the world and is used by organisations such as Google, CERN, and NASA. (CodeInstitute, n.d).

5.1.1 Use of Python

- AI & Machine Learning: Artificial Intelligence (AI) is commonly created using Python. 57% of data scientists and ML developers use Python.
- Data Analytics: Data and analytics tools are commonly written in Python. The tools available make it an excellent tool for data science.
- Web Development: Python is often chosen as a backend language in Web development. Its simplicity and syntax can save time and energy. Due to Python's popularity, web developers have many frameworks to choose from, an example being Django.
- Search Engine Optimisation: SEO professionals can optimize and automate tasks using Python. Web applications can be tested for errors too.
- Blockchain: Python has proven itself to be a useful language for Blockchain development. This is because Python offers flexibility, functionality, and security.
- Game Development: Python is a popular choice for game development. Its simplicity allows for fast prototyping and idea development in the game industry.
- Automation: Python is commonly utilized to automate tedious tasks such as checking information on databases, data visualizations and financial analysis. It is a useful language to use when working with large datasets.

(CodeInstitute, n.d)

5.1.2 Python Packet Capture Libraries

5.1.2.1 Scapy

Scapy is an "Interactive packet manipulation program". That can read and create packets with different protocols. It handles tasks such as scanning, tracerouting, and probing. Scapy is also capable of 802.11 frame injecting, ARP cache poisoning, and sending invalid frames. Its original author was Philippe Biondi. At the time of writing this report, Scapy was on version 2.4.5 which was released on April 19th, 2021. It can be used on Python version 2.7, 3.4 or newer and is cross-platform between Windows and Linux. (Pypi, 2022)

5.1.2.2 PyPCAPKit

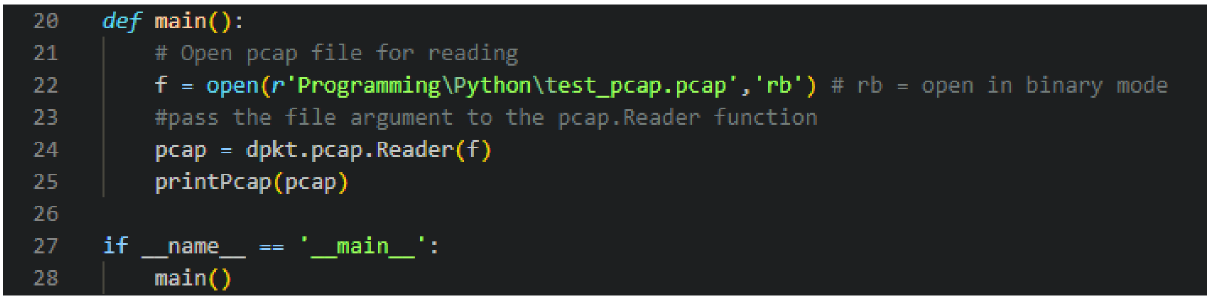
PyPCAPKit is an open-source program with a focus on parsing and analysing PCAP files. PyPCAPKit uses a streaming strategy to read files. They are read frame by frame which enhances efficiency. This library utilizes “DictDumper” and has various report output formats. The project's original author was Jarry Shaw. At the time of writing this report, PyPCAPKit was on version 0.16.3 which was released on October 31st, 2022. It is compatible with Python 3.6 or newer. It can be used on macOS, Windows, and Linux systems. PyPCAPKit can be used with Scapy, Pyshark, and DPKT (Pypi, 2022).

5.1.2.3 DPKT

DPKT is a Python module that allows users to create and parse network packets and features basic TCP/IP protocol definitions. Its original author was Dug Song. It is open source and has had many contributors. At the time of writing this report, DPKT was on version 1.9.8 which was released on August 18th, 2022. DPKT is compatible with Python 3.9. (Pypi, 2022)

5.1.3 Code Example

As described by (StudyTonight, 2022), after importing DPKT in Python, it can be used to parse .pcap files with an example shown below. Here line 24 specified dpkt.pcap.Reader to read parameter ‘f’ which is an existing .pcap file on the system.



```
20  def main():
21      # Open pcap file for reading
22      f = open(r'Programming\Python\test_pcap.pcap','rb') # rb = open in binary mode
23      #pass the file argument to the pcap.Reader function
24      pcap = dpkt.pcap.Reader(f)
25      printPcap(pcap)
26
27  if __name__ == '__main__':
28      main()
```

Figure 21 Using Python to open “test_pcap.pcap”, which is a file generated by Wireshark. Environment: Visual Studio code

5.2 Java

Java was created by Sun Microsystems in 1995. It has since become very popular and is widely used in today's world. Many services and applications are built using Java. Java is free and development kits are available from the Oracle website. Java runs in a virtual machine called the "Java Virtual Machine” and makes up only part of the process of running a Java application (Oracle, 2022).

Features of Java include the ability to run numerous applications, or a portion of an application concurrently, object-oriented programming, portability, and security as Java code is converted to bytecode, making it unreadable to humans. (Code Institute, n.d)

Key features as described by (Coursera, 2022) include the following:

- Robustness: Java helps developers achieve error-free programming. One way it does this is with the use of runtime checking.
- Secure: Java is preloaded with numerous security layers. This allows developers to use virus-free environments. An example of security in Java includes the lack of an explicit pointer and the automatic conversion of Java applications into bytecode.
- Platform independent: Java can be used as an all-in-one solution instead of writing unique Mac, Linux, or Windows-specific code.
- Multithreaded: Many tasks can be carried out at once using multithreading. Applications can run independently and the burden on the CPU is eased.

5.2.1 Use of Java

As mentioned in the previous paragraph, Java is found in a wide range of applications. These include televisions, mobile phones, game consoles, supercomputers, and original manufacturer equipment. Java is useful for projects that require high performance. (Code Institute, n.d).

(Coursera, 2022) describes the following as practical uses for Java:

- Cloud Applications
- Chatbots/Marketing Tools
- Enterprise-level web apps.
- Artificial Intelligence and Internet of Things devices.

5.2.2 Java and Pcap Files

5.2.2.1 Pcap4j

The first author of Pcap4j was Kaito Yamada. Pcap4j allowed users to capture packets on a network and turn them into Java objects. Packet headers can be set using these objects and packets can also be created and sent on a network. Supported protocols include Ethernet, Linux SSL, raw IP, ARP, ICMPv4 and ICMPv6, TCP, and DNS. Protocols can be added without the need to modify the Pcap4J library. Pcap files can be read by Pcap4J. It requires Java 5.0 or newer. When writing this report, Pcap4J last saw an update in 2019. It is compatible with Windows, macOS and Linux/UNIX. (Yamada, 2019)

5.2.2.2 jNetPcap

jNetPcap was produced by Slytechs and allows Java developers to use libpcap library bindings. Native libpcap functions can be used through this Java API. Npcap, legacy WinPcap tools and API extensions are supported by jNetPcap. Utilities provided by jNetPcap include byte array to hex string. (Slytechs, 2022). jNetPcap is implemented at a low level, making it efficient to use and is compatible with JRE 1.5 and relies on the WinPcap system. The use of the Libpcap native model makes learning jNetPcap easy. The most recent update to jNetPcap at the time of this report was in 2019. jNetPcap compiles on Windows and Linux systems. (Sourceforge, n.d).

Core functions of jNetPcap as outlined by (Sourceforge, n.d):

- List network interfaces on a system
- Open PCAP capture files.
- Apply packet filter.
- Dump packets to a pcap file
- Transmit raw packets over the network.
- Gather network interface statistics.

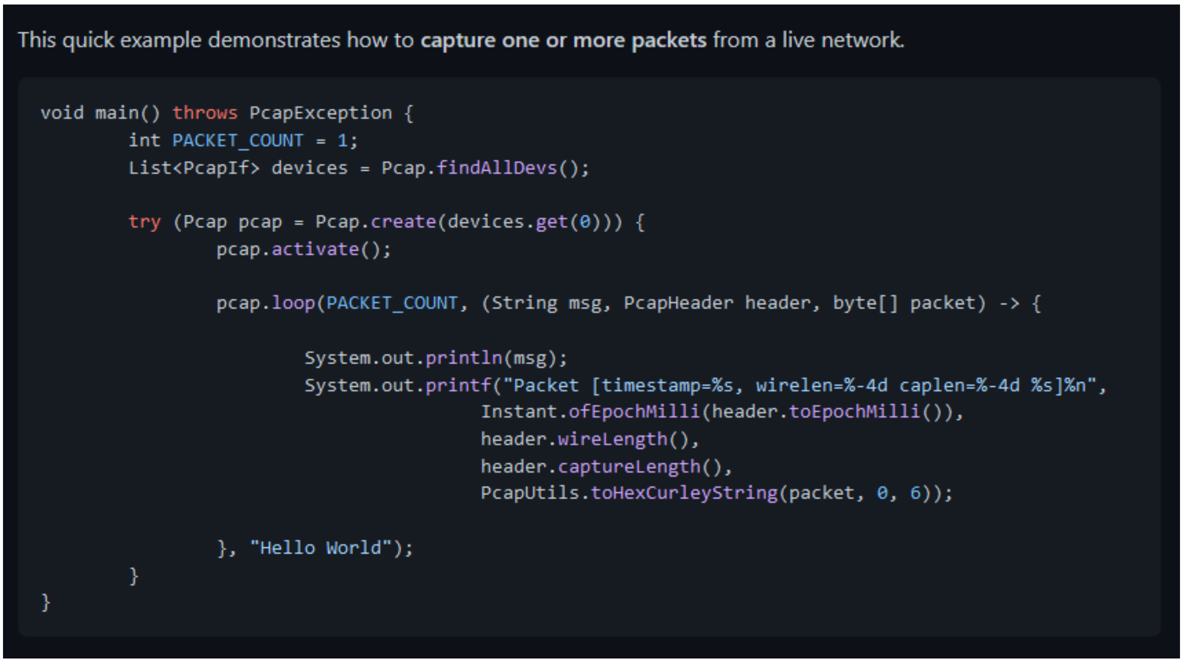
5.2.2.3 Pkts.io

Pkts.io. As stated by (Aboutsip, 2014) “pkts.io is a pure java library for reading and writing pcaps. Its primary purpose is to manipulate/analyse existing pcaps, allowing you to build various tools around pcaps.”. A feature Pkts.io has is the ability to use “streams”. The underlying protocol used determines what a stream represents in the Pkts.io library. All messages that are transmitted and received over the same local and remote port-pair constitute a stream when using the UDP protocol. Pkts.io can be accessed using Maven Central. Pkts.io would be useful for making tools that run on different systems as many current libraries utilize libpcap for example. A pure Java library makes portability easier for developers. (Aboutsip, 2014).

5.2.3 Code Example

5.2.3.1 jNetPcap

This example (Slytechs, 2022) captures a live packet using Java.



This quick example demonstrates how to **capture one or more packets** from a live network.

```java
void main() throws PcapException {
        int PACKET_COUNT = 1;
        List<PcapIf> devices = Pcap.findAllDevs();

        try (Pcap pcap = Pcap.create(devices.get(0))) {
                pcap.activate();

                pcap.loop(PACKET_COUNT, (String msg, PcapHeader header, byte[] packet) -> {

                        System.out.println(msg);
                        System.out.printf("Packet [timestamp=%s, wirelen=%-4d caplen=%-4d %s]%n",
                                        Instant.ofEpochMilli(header.toEpochMilli()),
                                        header.wireLength(),
                                        header.captureLength(),
                                        PcapUtils.toHexCurleyString(packet, 0, 6));

                }, "Hello World");
        }
}
```

Figure 22 Slytechs, 2022., jNetPcap version 2 [Online] Available at: https://github.com/slytechs-repos/jnetpcap [Accessed 21/11/2022]

5.3 C++

C++ was created by Bjorne Stroustrop in the early 1980s as a successor to the C family of languages. It was created as a cross-platform upgrade to C. Programmers using C++ had more control over system resources such as memory. While C++ is object-oriented, it also supports procedural and functional programming. Flexibility and scalability have allowed C++ to be used to create various things such as operating systems, graphical user interfaces, and games. While C++ was a successor to C, Challenging principles that defined C's design are also present in C++. Pointers are a difficult concept to understand. Misuse of pointers can cause system crashes and unexpected memory usage. (Buttice, 2021)

5.3.1 Use of C++

C++ has found use in both big software infrastructures as well as resource-constrained applications where there is little hardware space or energy available to power the application. The access that C++ has to the hardware it runs on allows great flexibility for programmers. C++ is a good choice when building applications that will carry out critical tasks as it is reliable, fast, and efficient. As outlined below C++ has found use in applications from aeroplanes to smartwatches. (Xiao, 2021)

Some real-world applications of C++ are outlined by (Xiao, 2021) as follows:

- Operating systems: C++ plays an important role in operating systems such as macOS and Windows. Operating systems need to be as fast and reliable as possible. Low-level capabilities of the C++ language allow this.
- Game development: Games like StarCraft and Counterstrike were made using C++ as it is a language that allows for the optimization of resources.
- IoT devices: C++ is used in devices like TVs and smartwatches due to their limited computing resources.
- Databases: C++ was utilized to create MySQL and MongoDB. Efficient storage is supported by C++.
- Web browsers: Chrome and Safari utilize C++ in their back-end processes to complete tasks quickly and efficiently.
- Critical tasks such as those found in flight software commonly use C++. Coding guidelines are carefully followed to ensure vital components function predictably throughout flights.

5.3.2 C++ Packet Capture Libraries

5.3.2.1 Libpcap Library

Libpcap was created in 1994 by researchers at the University of California in Berkeley. Today it is maintained by the Tcpdump group. It was designed to be used with C/C++; However, wrappers are available for the API and can allow developers to use the following languages:

- Perl
- Python
- Java
- C#
- Ruby

(Garcia, 2008)

The Libpcap API allows developers to capture link-layer packets on UNIX operating systems. The link layer is known as layer two on the OSI model. Libpcap utilizes a machine's network card to capture, decode and display packets. Developers using the Libpcap API do not need to be familiar with different operating systems' network interface cards or drivers to process packets from them. (Beale J, Foster J, Caswell B, 2003)

Windows versions of Libpcap include WinPcap and Npcap. Winpcap is based on an old version of Libpcap while Npcap is based on a more recent version. As of this writing this report, only Npcap is actively maintained and is available for machines running Windows 7 or later. (Wireshark, 2020)

As (Keary, 2022) states; “Promiscuous mode” is enabled on a Network Interface Card to effectively capture packets. In this mode, the filter that only allows packets addressed to the NIC is disabled.

The result of this is all packets on the network are processed by the NIC. A NIC in non-promiscuous mode will drop all frames unless they are addressed to it.

5.3.2.2 Npcap

The Npcap packet capture library for Microsoft Windows was developed by the team behind Nmap. Npcap provides access to raw network data for Windows applications. This allows such applications to capture, filter, transmit, and gather statistics using network packets. (Npcap, n.d).

Npcap is used in a range of applications such as monitors, analysers, traffic generators, and general security tools. Furthermore, it is designed for modern Windows such as Windows 10 and 8.1. It replaced WinPcap in many applications. Npcap includes the latest version of libpcap (Npcap, n.d).

5.3.2.3 PcapPlusPlus

PcapPlusPlus is a C++ library that allows for the capture, parsing and creation of network packets. A large amount of networking protocols can be forged and decoded and wrappers for engines such as libpcap, WinPcap, and DPDK are provided. PcapPlusPlus is efficient and lightweight with support for Windows, FreeBSD, Android, Linux, and macOS. When writing packets to files developers can choose formats such as pcap and pcapng. At the time of writing this report, stable PcapPlusPlus is on version 22.11. Its original author goes by the name of Seladb, and the project has found a sizable number of contributors on GitHub. (PcapPlusPlus, 2022)

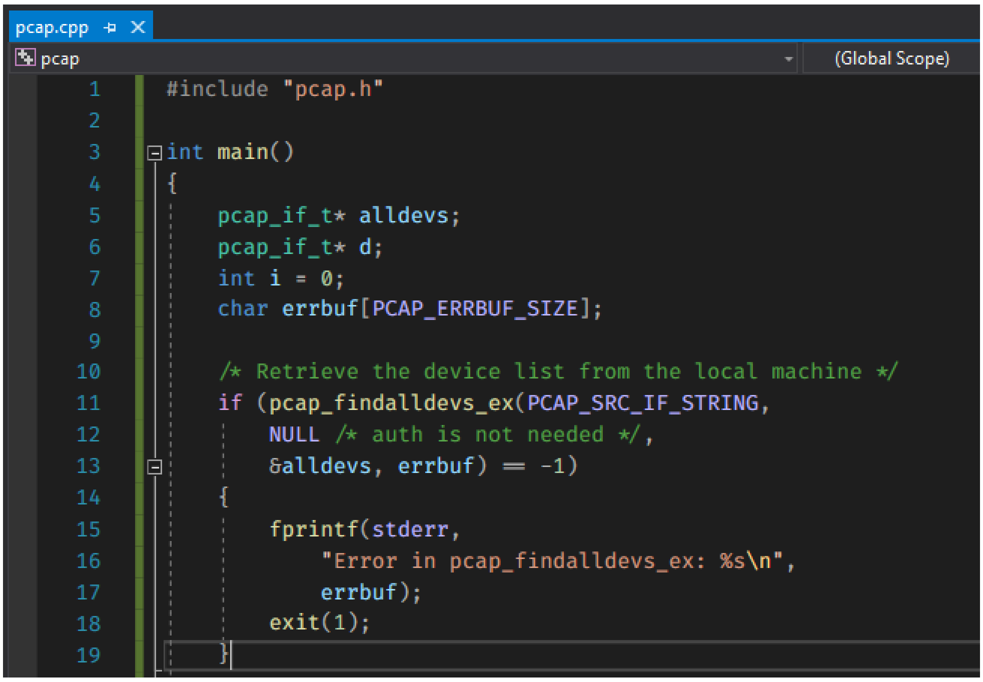
5.3.3 Code Example

Before working with pcap files in C++, Npcap was installed. Npcap was chosen over WinPcap as WinPcap ceased development.

After Npcap was downloaded, a project was made on Visual Studio and an additional directory was added from the project properties option. The “Include” directory from Npcap was selected.

- “WPCAP” and “HAVE_REMOTE” were added as pre-processors.
- In the linker tab, “Lib” from the Npcap download was added as an additional library directory.
- In the input tab, additional dependencies "wpcap.lib" and "packet.lib" was added.

The following code example which obtains a network adapter list, and prints them to the screen, is from the (Npcap, n.d) development tutorial page.



```
#include "pcap.h"

int main()
{
    pcap_if_t* alldevs;
    pcap_if_t* d;
    int i = 0;
    char errbuf[PCAP_ERRBUF_SIZE];

    /* Retrieve the device list from the local machine */
    if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING,
        NULL /* auth is not needed */,
        &alldevs, errbuf) == -1)
    {
        fprintf(stderr,
            "Error in pcap_findalldevs_ex: %s\n",
            errbuf);
        exit(1);
    }
```

Figure 23 Getting a list of network adapters on a device using Npcap and C++ using (Npcap, n.d) documentation. Environment: Visual Studio 2019

6 Feature Matrix

6.1 Programming Language Capabilities

During the research process of libraries for the different programming languages, I took note of the features and capabilities offered by each library. I found that there were libraries available for each language that offered many essential features required for the development of my application.

6.1.1 Language Capability Matrix

Feature	***Python***	***Java***	***C++***
Transmit packets on a network	✓	✓	✓
Read packets on a network	✓	✓	✓
Has GUI framework	✓	✓	✓
Produce PCAP format files	✓	✓	✓
Read PCAP format files	✓	✓	✓
Linux compatible	✓	✓	✓
macOS compatible	✓	✓	✓
Microsoft Windows compatible	✓	✓	✓

6.2 Network Analysis Libraries

The feature matrix in section 6.1 demonstrated that adequate libraries existed for the three programming languages I considered. As a result, I conducted further research to identify the most suitable library for each of the programming languages with which I was familiar.

During my research, I found two main contenders for a core packet capture library to be used in the project. These were:

1. *Libpcap, with C/C++*
2. *Scapy, with Python*

Libpcap and Scapy were both found to meet the core requirements of the project, including compatibility with Microsoft Windows 10, support for network packet capture, and compatibility with packet capture file formats. Both libraries are also actively maintained and well-documented, making them suitable for use in development. Moreover, as popular choices for packet sniffing, these libraries are likely to have solutions readily available for any common issues that may arise during development, such as on Stack Overflow.

6.2.1 Library Matrix

Feature	***Libpcap***	***Scapy***
Packet Capture	Raw 802.11 Packet Capture Support	Raw 802.11 Packet Capture Support
PCAP File Support	✓	✓
Microsoft Windows 10 Support	✓	✓
Packet Creation	✓	✓
Adequate Documentation	Yes, e.g., “Documentation” on www.tcpdump.org	Yes, e.g., “Build your own tools” on scapy.readthedocs.io
Access to library	Free download from developer website.	Free download from developer website.

(Tcpdump, 2023)

(Scapy, 2023)

7 References

Solarwinds, 2022., *Network Performance Monitor.* [Online] Available at: https://www.solarwinds.com/network-performance-monitor/use-cases/packet-analyzer [Accessed 24 OCT 2022]

Solarwinds, 2022., *Documentation for Hybrid Cloud Observability Essentials and Network Performance Monitor* [Online] Available at: https://documentation.solarwinds.com/en/success_center/npm/content/system_requirements/npm_12-5_system_requirements.htm [Accessed 24 OCT 2022]

Casey, E., 2010 *Handbook of Digital Forensics and Investigation* [Online] Available at: https://www.sciencedirect.com/topics/computer-science/tcpdump [Accessed 31 OCT 2022]

Gerardi, R., 2020 *An introduction to using tcpdump at the Linux command line* [Online] Available at: https://opensource.com/article/18/10/introduction-tcpdump [Accessed 31 OCT 2022]

Beale J., Foster J., Caswell B., *Snort Intrusion Detection 2.0* [Online] Available at: https://www.sciencedirect.com/topics/computer-science/libpcap-library [Accessed 31 OCT 2022]

Wireshark, 2022., *Packet capture library (libpcap)* [Online] Available at: https://wiki.wireshark.org/ [Accessed 31 OCT 2022]

Keary, T., 2022 *PCAP: Packet Capture, what it is & what you need to know* [Online] Available at: https://www.comparitech.com/net-admin/pcap-guide/ [Accessed 31 OCT 2022]

Garcia, L., 2008, *Programming with Libpcap – Sniffing the network from our own application* [Online] HAKING. Available at: http://recursos.aldabaknocking.com/libpcapHakin9LuisMartinGarcia.pdf [Accessed 31 OCT 2022]

Kaspersky, 2022., *What is a Packet Sniffer?* [Online] Available at: https://www.kaspersky.com/resource-center/definitions/what-is-a-packet-sniffer [Accessed 01 NOV 2022]

Profitap, 2022., *ProiShark Network TAPs* [Online] Available at: https://www.profitap.com/profishark-network-taps/ [Accessed 01 NOV 2022]

Edwards, J, 2020., *What is Packet Sniffing and How Does it Work?* [Online] Available at: https://www.whatsupgold.com/blog/what-is-packet-sniffing [Accessed 01 NOV 2022]

BasuMallick, C., 2022 *What is Packet Sniffing? Meaning, Methods, Examples, and Prevention Best Practices for 2022* [Online] Available at: https://www.spiceworks.com/it-security/network-security/articles/what-is-packet-sniffing/ [Accessed 02 NOV 2022]

PagerDuty, 2022., *Network Sniffers: What Are They and How Can I Use Them?* [Online] Available at: https://www.pagerduty.com/resources/learn/what-are-network-sniffers/ [Accessed 02 NOV 2022]

Breeden II, J, 2022., *What is Wireshark?* [Online] Available at: https://www.networkworld.com/article/3663021/what-is-wireshark.html [Accessed 02 NOV 2022]

Kinzie, K, 2022., *How to Use Wireshark: Comprehensive Tutorial + Tips* [Online] Available at: https://www.varonis.com/blog/how-to-use-wireshark [Accessed 03 NOV 2022]

CodeInstitute, n.d., *What is Python used for? 7 Practical Uses* [Online] Available at: https://codeinstitute.net/ie/blog/what-is-python-used-for/ [Accessed 05 NOV 2022]

LiveAction, 2022., *About LiveAction* [Online] Available at: https://www.liveaction.com/company/ [Accessed 06 NOV 2022]

Neox, 2022., *LiveAction OmniPeek Identify network problems and create a quick & meaningful diagnosis* [Online] Available at: https://www.neox-networks.com/en/network-analysis-troubleshooting/ [Accessed 07 NOV 2022]

Gurubaran, 2021., *What is PCAP File, why do we need to Use & How it Works?* [Online] Available at: https://cybersecuritynews.com/pcap/ [Accessed 13 NOV 2022]

Pypi, 2022., *Find, install, and publish Python packages with the Python Package Index* [Online] Available at: https://pypi.org/ [Accessed 14 NOV 2022]

Yamada, K.,2019 *Pcap4J* [Online] Available at: https://github.com/kaitoy/pcap4j [Accessed 15 NOV 2022]

Oracle, 2022., *What is Java technology and why do I need it?* [Online] Available at: https://www.java.com/en/download/help/whatis_java.html [Accessed 19 NOV 2022]

Code Institute, n.d., *What is Java and what is it used for?* [Online] Available at: https://codeinstitute.net/ie/blog/what-is-java/ [Accessed 19 NOV 2022]

Coursera, 2022., *What Is Java Used For?* [Online] Available at: https://www.coursera.org/articles/what-is-java-used-for [Accessed 19 NOV 2022]

StudyTonight, 2022., *Analyzing Networking Traffic using dpkt library* [Online] Available at: https://www.studytonight.com/network-programming-in-python/analyzing-network-traffic [Accessed 20 NOV 2022]

Slytechs, 2022., *jNetPcap version 2* [Online] Available at: https://github.com/slytechs-repos/jnetpcap [Accessed 21 NOV 2022]

Sourceforge, n.d., *jNetPcap version 1.0 Production* [Online] Available at: https://jnetpcap.sourceforge.net/docs/jnetpcap-1.0-javadoc/overview-summary.html [Accessed 21 NOV 2022]

Aboutsip, 2014., *pkts.io* [Online] Available at: https://www.aboutsip.com/pktsio/ [Accessed 21 NOV 2022]

Buttice, C, 2021., *C Plus Plus Programming Language (C++)* [Online] Available at: https://www.techopedia.com/definition/26184/c-plus-plus-programming-language [Accessed 21 NOV 2022]

Xiao, L., 2021., *What is C++ Used For?* [Online] Available at: https://www.codecademy.com/resources/blog/what-is-c-plus-plus-used-for/ [Accessed 21 NOV 2022]

PcapPlusPlus, 2022., *Welcome to PcapPlusPlus!* [Online] Available at: https://pcapplusplus.github.io/ [Accessed 23 NOV 2022]

Npcap, n.d., *Npcap Development Tutorial* [Online] Available at: https://npcap.com/guide/npcap-tutorial.html [Accessed 23 NOV 2022]

Orebaugh, A., Ramirez, G. and Beale, J. (2006). *Wireshark & Ethereal Network Protocol Analyzer Toolkit*. Elsevier.

Scapy, 2023, *General documentation* [Online] Available at: https://scapy.readthedocs.io/en/latest/index.html [Accessed 9th APR 2023]

Tcpdump, 2023, *Documentation* [Online] Available at: https://www.tcpdump.org/ [Accessed 10th APR 2023]