

Identifying Security threats

Cyber Security



Noel Rottebiller N0931129

Maks Miketa N0945100

Jawad Naeem N0972124

Phillip Radasevic N0930646

Contents

[What are PCAP Files? 3](#_Toc98188954)

[Use of PCAP Files? 3](#_Toc98188955)

[Importance of Packet Capturing Monitoring? 4](#_Toc98188956)

[How the code works to read the PCAP file 4](#_Toc98188957)

[PCAP File Global Header Analysis 5](#_Toc98188958)

[The length of the global header 5](#_Toc98188959)

[The magic number and the endianness of the PCAP file 5](#_Toc98188960)

[The major and minor version numbers of the file format 5](#_Toc98188961)

[The SnapLength 6](#_Toc98188962)

[The data link type 6](#_Toc98188963)

[Analysis of DHCP frame 6](#_Toc98188964)

[What Is the DHCP frame? 6](#_Toc98188965)

[The timestamp indicating when this packet was captured 6](#_Toc98188966)

[The actual GMT corresponding to this timestamp 6](#_Toc98188967)

[The length of this DHCP frame 6](#_Toc98188968)

[The source and destination MAC addresses of the captured communication 7](#_Toc98188969)

[The source and destination IP addresses of the captured communication 7](#_Toc98188970)

[The name of the host PC 7](#_Toc98188971)

[How the code works to find specifc parts of the DHCP packet. 7](#_Toc98188972)

[Using Regular Expression to find a susceptible website 7](#_Toc98188973)

[Finding search engine of the website 7](#_Toc98188974)

[Generic Python code to identify more potential security threats 7](#_Toc98188975)

[Bibliography 7](#_Toc98188976)

[Appendices 8](#_Toc98188977)

[Appendix A 9](#_Toc98188978)

[Appendix B 10](#_Toc98188979)

[10](#_Toc98188980)

# What are PCAP Files?

PCAP or we call it Packet Capture is something we can say an API (Application Programming Interface) that captures live network data from OSI model and specifically from layers 2 to 7 which includes data link layer, network layer, transport layer, session layer, presentation layer, and application layer (What is OSI Model | 7 Layers Explained | Imperva 2021).

Graphical user interface

Description automatically generated

Figure 1 - OSI Model Layers (What is OSI Model | 7 Layers Explained | Imperva, 2021)

These PCAP files can be used to view TCP/IP and UDP network packets. If you want to record network traffic then you need to create a .pcapfile. You can create a .pcapfile by using a network analyser or packet sniffing tool like Wireshark.

PCAP otherwise known as libcap are captured using a certain application such as Wireshark. The files can hold various information such as the network traffic/packet information captured within a certain period.

## Use of PCAP Files?

There are multiple different uses of a PCAP file which range from home use to big business. Home use can be to see what traffic is coming to your device. If there is any unnecessary traffic that could be prevented. When it comes to big business it is used for network forensics. This can be for different uses security reasons to detect a breach or to see how much traffic their websites are getting. These are both very important when it comes to business as one secures the business and the other shows the amount of traffic a website is getting.

PCAP is a valuable resource for file analysis and to monitor your network traffic. Packet collection tools like Wireshark allow you to collect network traffic and translate it into a format that’s human-readable. There are many reasons why PCAP is used to monitor networks. Some of the most common include monitoring bandwidth usage, identifying rogue DHCP servers, detecting malware, DNS resolution, and incident response.

For network administrators and security researchers, packet file analysis is a good way to detect network intrusions and other suspicious activity. For example, if a source is sending the network lots of malicious traffic, you can identify that on the software agent and then take action to remediate the attack.

IT teams prefer using packet monitor to perform crucial tasks, such as:

• Monitoring WAN utilization

• Monitoring bandwidth and traffic volume

• Tracking network usage

• Isolating compromised systems

• Demonstrating compliance

• Detecting suspicious content

• Identify rogue DHCP

## Importance of Packet Capturing Monitoring?

Packet capture enables teams to deal with complex network issues with ease and efficiency. Management of organizations' networks is daunting. It involves checking client IP addresses, DNS servers, and more following the standard tests to identify the root cause of the issues.

Different Versions of PCAP Files:

There are many versions of a PCAP file with its own set of capabilities and use cases. For example:

• WinPcap: WinPcap file format is similar to a portable packet capture library designed specifically for Windows devices.

• Libpcap: Libpcap is an open-source C++ library Used by Mac OS and Linux devices to capture and filter packets. It’s mainly used by packet sniffing tools.

• Npcap: Npcap is a packet sniffing library that supports other Windows devices. This library is known for its fast and secure functionalities.

• PCAPng: Users can loopback packet capture injection and sniff loopback packets.

However, within this portfolio, the captured PCAP file will be analysed through Python code.

# PCAP File Global Header Analysis

## The length of the global header

A PCAP file begins with the global header which has a fixed size of 24 bytes consisting of the magic number which are the first 4 bytes of the file. The next 2 bytes of the file are the major version number followed by next 2 bytes that are the minor version number of the file. The next 4 bytes represent the correction time in seconds between GMT and the local time zone of the packet header timestamps. The next 4 bytes show the accuracy of the timestamp in the capture. The next 4 bytes show the snaplength and the last 4 bytes show the data link type resulting in 24 bytes within the global header. Within the python code, the globalHeader function prints the global header with all the information stored within the global header. The information can be stored on a text file. Please view appendix A for the global header information and Appendix B to view the code used to read the bytes in the header.

## The magic number and the endianness of the PCAP file

The magic number is 32 bits long and is used to detect the file format itself and the byte ordering. The magic number determines if the file is a big or little-endian file. Endianness is the ordering of bytes where little-endian stores the least significant bytes before the more significant bytes and big-endian stores the most significant bytes before the less significant bytes. If the magic number has the value of 0xa1b2c3d4, the file is big endian. If the hexadecimal value is 0xd4c3b2a1, file is little endian and must be processed in reverse order (Jorquera, 2020). The magic number of the PCAP file is 0xd4c3b2a1 therefore, it is a little-endian file. Line 14 reads the magic number and if it is “d4c3b2a1” then it appends the MagicNumber carriable as little but, if it is not that then it appends as ‘big’.

Text

Description automatically generated

Figure – Python Code

## The major and minor version numbers of the file format

The major and minor versions are unsigned values that give the number of the current major/minor version of the file format. The major version of the file format is 2 and the minor version is shown as 4 so the version of the file is 2.4. The code to read the files is on lines 20 and 21.

Text

Description automatically generated

Figure - Python Code

## The SnapLength

The snaplength is the snapshot length for the capture. A captured packet in the file may not contain all data in the packet as appeared on the network so the captured file might contain at most the first N bytes of each packet for some value of N. The value of N is the snapshot length which in this file is 65535.

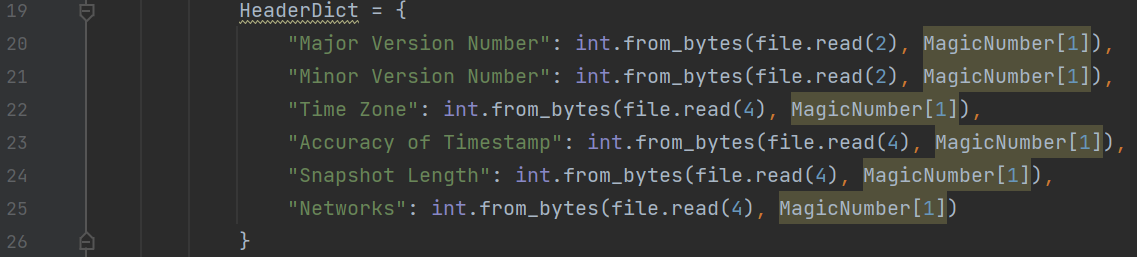


Figure - Python Code

## The data link type

The data link type specifies the link-layer type value which is 1 in the captured file. Number 1 value means that the data link type used is ethernet.

Text

Description automatically generated

Figure - Python Code

# Analysis of DHCP frame

## What Is the DHCP frame?

The Dynamic Host Configuration Protocol (DHCP) is the first protocol captured within a PCAP file. It contains a lot of basic information such as the valid TCP/IP configuration parameters for all clients on a network and valid IP addresses, maintained in a pool for assignment to clients. The DHCP is necessary for servers to connect to each other and is very helpful in identifying certain destinations that your device could have connected to.

## The timestamp indicating when this packet was captured

The timestamp is 1485557588. This is called the Epoch time. This is captured inside the DHCP and the way we have coded it is that we split the packet and read certain parts the packet and split the message up and print it off via a text file.

## The actual GMT corresponding to this timestamp

Once you have the epoch time is just a simple decode to get the actual GMT time. We decode this using Epoch in python and simply printing out the GMT time. The time that the DHCP packet was captured was at 2017-01-27 22:53:08.

## The length of this DHCP frame

The length of DHCP packet is 348 bytes. The way this was found via python was first we created a menu. This was for the user to be able to decide what option they want. 1 is to analyse the pcap file and 2 was to choose an exact packet out of the 6033 packets. Knowing DHCP was number one we would select 1 out of 6033. This is when the code to find the length of the DHCP is used. The way this works is that it takes the packet and splits it. As we know different bits of information are sent at different times in the packet. This means we know the time stamp is before the included length and original length. Then we take the information and print it out decoded to be able to be read which in this case we get 348 bytes which is the length of the packet.

## The source and destination MAC/ IP addresses of the captured communication

## Graphical user interface, text Description automatically generated

Figure - Python Code

## The name of the host PC

PC Host Name: [b'Stewie-PC']

## How the code works to find specific parts of the DHCP packet.

First, it’s important to understand how a packet sends information. The DHCP packet hold a lot of important information such as the MAC address; Ip address; magic cookie and the time it was sent or received. The DHCP packet is split into different parts such as the timestamp, length and even the MAC address or iP address of the source and destination.

The code below shows how the information is gathered from the packet. Knowing how big the packet is we can use Wireshark to understand what parts of the packet are sent through and when. Using this information, we split the packet apart as seen in the square brackets and then print out the information for that specific DHCP packet. However, we also had to convert the text from hexadecimal to something readable for us. This is where we use the .hex which converts it into normal writing.

# Text Description automatically generated

Figure - Python Code

Once we have created a way of decoding the text and getting certain information, we needed it was just about formatting the information. The code below shows that we split the information and format for the DHCP dictionary so that it comes out as something readable for us.

Text

Description automatically generated

Figure - Python Code

Printing it was simple the code is below. It was simply formatting it to look presentable and using the .get to get the after its been formatted to print next to the title.

Text

Description automatically generated

Figure - Python Code

# Using Regular Expression to find a susceptible website

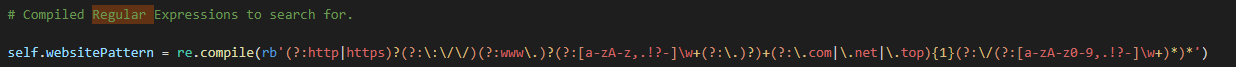
Using regex, a website for creating our own regular expression we came up with the code below. We have to use re.complier to make sure that it knows to use the regular expression we have put in when looking for the website. This regular express then gives us all the websites that have .top.net and .com. 

Figure - Python Code

Then to find the websites with just .top we would use.

rePattern = re.compile(r'(?:http:\/\/)[a-z, A-Z 0-9 .]+(?:.top)')

Below is what we received

Text

Description automatically generated

Figure - Python Code

# Finding search engine of the website

Seeing the code below shows us that. We would search for the first .top entry. Then via the code use it to find the search engine and recommended site. It also tells us the susceptible website and where the user was redirected to. The thought process behind it was finding the first .top entry once that has been done we would assume to that the first

Text

Description automatically generated

Figure - Python Code

Text

Description automatically generated

Figure - Python Code

# Generic Python code to identify more potential security threats

Text

Description automatically generated

Figure - Python Code

Text

Description automatically generated

Figure - Python Code

Text

Description automatically generated

Figure - Python Code

Text

Description automatically generated

Figure - Python Code

Text

Description automatically generated

Figure - Python Code

Text

Description automatically generated

Figure - Python Code

# Bibliography

Cyber Security News. 2021. What is PCAP File, Why do we Need to Use & How it Works?. [online] Available at: <https://cybersecuritynews.com/pcap/> [Accessed 5 March 2022].

Filext. 2022. Opening PCAP Files. [online] Available at: <https://filext.com/file-extension/PCAP> [Accessed 5 March 2022].

Harris, G. and Richardson, M., 2022. PCAP Capture File Format. [online] Tools.ietf.org. Available at: <https://tools.ietf.org/id/draft-gharris-opsawg-pcap-00.html> [Accessed 4 March 2022].

Jorquera, Z., 2020. What Is Little-Endian And Big-Endian Byte Ordering?. [online] Engineering Education (EngEd) Program | Section. Available at: <https://www.section.io/engineering-education/what-is-little-endian-and-big-endian/> [Accessed 4 March 2022].

Learning Center. 2021. What is OSI Model | 7 Layers Explained | Imperva. [online] Available at: <https://www.imperva.com/learn/application-security/osi-model/> [Accessed 5 March 2022].

Wiki.wireshark.org. 2022. LibpcapFileFormat. [online] Available at: <https://wiki.wireshark.org/Development/LibpcapFileFormat> [Accessed 4 March 2022].

# Appendices

## Appendix A

Function to print out the global header information.

Text

Description automatically generated

Information printed from global header which can be saved as a text file.

Text

Description automatically generated

## Appendix B

Dictionary is created to read bytes from the file then, the information is stored in the dictionary. The bytes are read using file.read().

## Text Description automatically generated