

E-ISSN: 2582-2160 • Website: <u>www.ijfmr.com</u> • Email: editor@ijfmr.com

# Exploring Wireshark for Network Traffic Analysis

# Brij Mala<sup>1</sup>, Sanskar Agrawal<sup>2</sup>, Aditya Sharma<sup>3</sup>, Rupinder Kaur<sup>4</sup>

<sup>1,2,3</sup>B. E Scholars, Dept. of Computer Science and Engineering Chandigarh University- Mohali, India <sup>4</sup>Professor, Deptartment of Computer Science and Engineering Chandigarh University-Mohali, India

# Abstract

In today's world security and privacy are the most needed qualities of any network. From calling to online transactions all occur on the network on daily basis. So there is the need to analyse the network in which all these activities occur. Network analysis thus plays a vital role in maintaining and optimizing network performance, security and troubleshooting. This paper researches on the topic network traffic analysis using Wireshark. Wireshark is an open source network packet analyser and it can be used by security administrators to analyse the traffic, analyse data packets and their role in shaping network behaviour. The network analysis using Wireshark will help to know any suspicious or malicious traffic in our network so that timely action can be taken to avoid attacks like DDOS, Man-in-the middle etc.

Keywords: Network traffic analysis, Wireshark, Network Packet Analyzer

#### 1 Introduction

# **1.1** Network Traffic Analysis

It is a process of capturing the traffic or data packets in the network and analyse them [3]. It also includes monitoring and evaluating the data that is transmitted over the network from the computers to the servers and back. It involves the examination of the captured data packets that flow through the network in order to gain insights into the network's behaviour, performance and security. The analyses helps the network administrators to track the utilization of the network, identify and suspicious activity, bottlenecks. In the event of breach, network traffic analysis provides valuable evidence for investigating, understanding the reasons for the attack. Network traffic analysis can be performed using various tools and techniques, like intrusion detection systems, packet sniffers, Wireshark etc.

# 1.2 Wireshark

Wireshark is an network packet analyzer used for network analysis by security administrators, network administrators in order to understand the flow of packets over the network [1]. Wireshark provides a platform to capture, inspect and analyse the packets and the traffic. It was originally known as Ethereal. It came into existence in order to better understand the inner working of network communication protocols like TCP, UDP, HTTP etc. Over time it has evolved as a versatile and indispensable resource to drive insights and reveal hidden dynamics of the network behaviour [2]. Wireshark thus helps in trouble-shooting network issues, analysing packets, analysing various protocols and their role in the network traffic. Wireshark also provide various statistical methods and graphs to clearly visualize and understand the network traffic and thus helping the network security administrators to detect various active and pas-



sive attacks over the network. With the help of Wireshark we can get the IP addresses and the Port numbers of sender and receiver.

#### 1.3 Packet Capturing

Packet capturing refers to the process of recording and intercepting individual data packets that flow over the network. These packets are the fundamental units of data transmission. The packets includes data like the IP address, the port numbers, length of the data, data itself and much more. Capturing allows the network administrators, security professionals, analysts to inspect, analyse and examine the network traffic in real time or offline.

#### 2 Literature Review

Author/	Title	Purpose	Source	Summary
Creator				
Muhammed	A Deeper Look	This paper discusses the optimiza-	Research	The paper discussed
Alfawareh	into Network	tion of traffic analysis performance,	Paper	the use of wireshark
	Traffic Analy-	detection of network forensics and		for network traffic
	sis using	spam, network proofing with pene-		analysis, its role in ad-
	Wireshark	tration testing, policy formation,		dressing network fo-
		and data delivery in integrated sys-		rensics, and the risks
		tems, while also discussing coun-		associated with obtain-
		termeasures to reduce this risk [3].		ing useful information
				for attacks or stealing
				data, while also ad-
				dressing solutions [3].
G Jain,	Application of	Intrusion detection systems analyze	Research	This paper shows how
Anubha	SNORT and	packets to secure data transmission	Paper	to use Wireshark tool
	Wireshark in	in networks. Wireshark is a popular		to generate all the pos-
	Network Traf-	tool for detecting intrusions, but it		sible details of log file.
	fic Analysis.	can intercept and analyze network		The I/O graph pro-
		encrypted traffic. SNORT captures		vides an overview of
		live packets from the internet, gen-		packet flow showing
		erating a log file. These log files are		total traffic, measured
		exported to Wireshark, where the		in bytes or packets per
		captured network packets are ana-		second [2].
		lyzed. The I/O graph displays the		
		packet flow, total traffic, TCP er-		
		rors [2].		
Bindu	Malicious Traf-	This paper demonstrated the appli-	Research	This paper showcases
Dodiya,	fic Analysis	cation of Wireshark in diagnosis of	Paper	packet analysis using
Umesh	using	network protocol and identify mal-		indicators of compro-
Kumar	Wireshark by	ware by collecting compro-		mise, demonstrating its
Singh	collection of	mise indicators [1].		usefulness in network

#### **Table 1. Review Summary**



E-ISSN: 2582-2160 • Website: <u>www.ijfmr.com</u> • Email: editor@ijfmr.com

Indicators of		forensics. W	'ireShark's
Compromise		analysis can	identify
		various secur	ity threats
		and attacks	on net-
		worked	comput-
		er systems [1]	].

# 3 Methodology

Wireshark uses several phases for network traffic analysis. The steps are as follows for capturing the network packet:

- Select the network on which you want to perform traffic analysis from various available options like Ethernet, Wi-Fi , LAN connection etc.
- If you wish to apply filters before starting the capture you can do so by typing the same in the text field provided by the Wireshark. Filters like choosing a specific protocol or port and many more can be applied.
- Traffic would be captured and be available for analysis. We can also save the captured traffic for future work if we wish to do so.
- The packets are then decoded, examined, and analysed by the users.





# 4 Transmission Control Protocol

TCP (Transmission Control Protocol) is a transport layer protocol that is used to transmit packets from sender to receiver over the network. It is a connection oriented protocol, and thus uses three-way hand-shake for connecting devices before sending data over the Internet. This connection remains established till the communication is completed.

# **TCP HEADER:**

- Source Port: It includes the source or sender's unique port number. The field has 16 bits[5].
- **Destination Port:** It includes the receiver's or destination's precise port number. The field has 16 bits[5].
- **Sequence Number:** This identifies the amount of data transferred throughout the TCP Session. It's a 32-field here. Initial sequence number for a new connection is a random 32-bit value. Utilizing this sequence number, the recipient replies with an acknowledgment to the sender. To make it simpler to comprehend, Wireshark utilizes relative sequence numbers beginning with 0[5].
- **DO** (**Data Offset**): The header length is another name for the 4 bit data offset field. It provides the length of the TCP header to identify when the actual data starts [5].
- **RSV:** The reserved field's three bits are set to 0 and not used[5].



E-ISSN: 2582-2160 • Website: <u>www.ijfmr.com</u> • Email: editor@ijfmr.com

- **Flags:** There are nine flag bits, are also known as control bits. This field is used to create connections, transmit data and break connections:
- **Urgent Pointer:** (**URG**) The data should be viewed as having priority over other data when this bit is set [5].
- ACK: abbreviation for acknowledgement [5].
- **PSH:** This stands for PUSH. This instructs a program to send the data right away rather than waiting for it to fill the whole TCP segment [5].
- **RST**: This resets the connection. We must immediately cut off the connection, if we receive it. This is not the typical technique to close a TCP connection; it is only done when there are irrecoverable faults [5].
- **SYN:** This is used to establish the first sequence number and for the initial three-way handshake [5].
- **FIN:** The TCP connection is ended using this finish bit. TCP is full duplex, so in order to terminate a connection; both parties must use the FIN bit. This is how we typically cut off a connection [5].
- Window: The window field is of 16-bit. It is used to represent the maximum length of bytes that a receiver will accept. It does this by informing the number of bytes after the sequence number in the acknowledgment field [5].
- **Checksum:** Checksum is a 16 bit field which is used to tell whether the TCP header is in a good situation or not [5].
- **Urgent Pointer:** URG is a 16 bit field and when this bit is set then it acts as a marker for the end of the urgent data [5].
- **Options:** This is an optional field. It's size lies between 0 to 320 bits [5].



Fig 2. TCP Header

# 5 Wireshark: Network Traffic Capturing

The Wireshark has **easy to use interface**. Users can easily capture the packets and analyse those packets. It is available for both **UNIX and Windows**. Captured packets can also be exported and imported in a number of capture file formats for further analysis. Colour coding is also provided by Wireshark, helping the user to analyse the packets. Below table shows the various colouring rules.



🚄 Wireshark · Coloring Rules Defau	lt X
Name	Filter
Bad TCP	top.analvsis,flags && ltop.analvsis.window update && ltop.analvsis.keep alive && ltop.analvsis.keep alive ack
✓ HSRP State Change	hsrp.state != 8 && hsrp.state != 16
Spanning Tree Topology Chang	e stp.type == 0x80
✓ OSPF State Change	ospf.msg != 1
✓ ICMP errors	icmp.type eq 3    icmp.type eq 4    icmp.type eq 5    icmp.type eq 11    icmpv6.type eq 1    icmpv6.type eq 2    icmpv6.type eq 3    icmpv6.type eq 4
ARP	arp
✓ ICMP	icmp    icmpv6
✓ TCP RST	tcp.flags.reset eq 1
SCTP ABORT	sctp.chunk_type eq ABORT
TTL low or unexpected	(! ip.dst == 224.0.0.0/4 && ip.ttl < 5 && !pim && !ospf)    (ip.dst == 224.0.0.0/24 && ip.dst != 224.0.0.251 && ip.ttl != 1 && !(vrrp    carp))
Checksum Errors	eth.fcs.status=="Bad"    ip.checksum.status=="Bad"    tcp.checksum.status=="Bad"    udp.checksum.status=="Bad"    sctp.checksum.status=="Bad"    mstg
SMB	smb    nbss    nbns    netbios
HTTP HTTP	http    tcp.port == 80    http2
✓ DCERPC	dcerpc
Routing	hsrp    eigrp    ospf    bgp    cdp    vrrp    carp    gvrp    igmp    ismp
TCP SYN/FIN	tcp.flags & 0x02    tcp.flags.fin == 1
✓ TCP	tcp
UDP	udp
✓ Broadcast	eth[0] & 1
System Event	systemd_journal    sysdig
<	,
Pouble click to addit Oraza to mouse Rules are process	and is ender with a match is found
Course circle to early to move, Roles are process	es in order units a metur a nouro.
+ - 4	
	OK Copy from  ✓ Cancel Import Export Help

Fig 3. Coloring Rules

#### Interface to capture traffic:

Wireshark provides easy to use interface to capture the traffic. The below image shows the interface provided by Wireshark to capture the traffic. To start the traffic capture one needs to select the network like Wi-Fi, local area connection, adapter etc. We can also select various interfaces like wired, Bluetooth, wireless from the drop down menu. If we wish to apply filters like specific port number or specific protocol before starting the capture, we can do so by typing the same in **using the filter text box.** If we have captured any traffic before we can also open that traffic from the recent files available on the very first interface of Wireshark. As soon as a network is chosen, Wireshark starts the network traffic capturing and whatever is searched on various platforms, like Google Chrome, traffic pertaining to those platforms is captured for traffic analysis.



Fig 4. Wireshark Interface



E-ISSN: 2582-2160 • Website: <u>www.ijfmr.com</u> • Email: editor@ijfmr.com

	analysis5sites.pcapng												- 0	×	
Fil	e Edit View Go Capt	ure Analyze Statistics Te	elephony Wireless Tools	Help											
	🔳 🖉 🛞 📜 🛅 🗙 🛛	ै। 🤇 🗰 🗯 🚟 🛓	📜 📃 0, 0, 0, 🔢												
	Apply a display filter <ctrl-></ctrl->														
No.	Time	Source	Source MAC	Destination	Destination MAC		Protocol	Length I	nfo					^	
	10.000000	2404:6800:4002:	2404:6800:4002:	2401:4900:5c74:	2401:4900:5c74:d9eb:	14ad:d3	TCP	86 4	43 → 63826	[ACK]	Seg=1 Ack=1 N	√in=269 Len=	0 SLE=0 SR	E:	
	20.684578	2401:4900:5c74:	2401:4900:5c74:	2404:6800:4002:	2404:6800:4002:80b:	2003	TCP	75 6	3828 → 443	[ACK]	Seq=1 Ack=1 N	√in=2078 Len	=1 [TCP se	gr	
	30.888860	2401:4900:5c74:	2401:4900:5c74:	2001:b28:f23f:9	2001:b28:f23f:9::852	:437	TLSv1.2	2 257 4	pplication	Data					
	40.890002	2404:6800:4002:	2404:6800:4002:	2401:4900:5c74:	2401:4900:5c74:d9eb:	14ad:d3	TCP	86 4	43 → 63828	[ACK]	Seq=1 Ack=2 N	√in=265 Len=	0 SLE=1 SR	E	
	51.291670	2401:4900:5c74:	2401:4900:5c74:	2001:b28:f23f:9	2001:b28:f23f:9::852	:437	TLSv1.2	2 191 A	pplication	Data					
	61.293468	2001:b28:f23f:9	2001:b28:f23f:9	2401:4900:5c74:	2401:4900:5c74:d9eb:	14ad:d3	TLSv1.2	2 187 A	pplication	Data					
	7 1.336987	2401:4900:5c74:	2401:4900:5c74:	2001:b28:f23f:9	2001:b28:f23f:9::852	:437	TCP	74 6	3807 → 443	[ACK]	Seq=301 Ack=	14 Win=4186	Len=0		Various Packots
	8 2.019817	2001:b28:f23f:9	2001:b28:f23f:9	2401:4900:5c74:	2401:4900:5c74:d9eb:	14ad:d3	TCP	74 4	43 → 63807	[ACK]	Seq=114 Ack=	801 Win=180	Len=0		various Fackets
	9 2.019817	2001:b28:f23f:9	2001:b28:f23f:9	2401:4900:5c74:	2401:4900:5c74:d9eb:	14ad:d3	TLSv1.2	2 187 A	pplication	Data					captured in the
	10 2.067267	2401:4900:5c74:	2401:4900:5c74:	2001:b28:f23f:9	2001:b28:f23f:9::852	:437	TCP	74 E	3807 → 443	[ACK]	Seq=301 Ack=3	227 Win=4186	Len=0		traffic appear in
	11 2.845569	2401:4900:5c74:	2401:4900:5c74:	2404:6800:4002:	2404:6800:4002:819:	200a	TCP	75 e	3827 → 443	[ACK]	Seq=1 Ack=1 N	/in=257 Len=	1 [TCP seg	JTHE	this section
	12 2.845570	2401:4900:5c74:	2401:4900:5c74:	2404:6800:4002:	2404:6800:4002:81a::	200a	TCP	75 6	3829 → 443	[ACK]	Seq=1 Ack=1 N	√in=258 Len=	1 [TCP seg	7014	1113 3001011
	13 3.697065	2404:6800:4002:	2404:6800:4002:	2401:4900:5c74:	2401:4900:5c74:d9eb:	14ad:d3	TCP	86 4	43 → 63829	[ACK]	Seq=1 Ack=2 N	√in=271 Len=	0 SLE=1 SR	E:	
	14 3.697065	2404:6800:4002:	2404:6800:4002:	2401:4900:5c74:	2401:4900:5c74:d9eb:	14ad:d3	TCP	86 4	43 → 63827	[ACK]	Seq=1 Ack=2 N	√in=265 Len=	0 SLE=1 SR	E	
	15 3.800514	2401:4900:5c74:	2401:4900:5c74:	2404:6800:4002:	2404:6800:4002:80b::	2003	TCP	75 6	3831 → 443	[ACK]	Seq=1 Ack=1 N	√in=258 Len=	1 [TCP seg	<b>m</b> -	
	16 4.303139	2401:4900:5c74:	2401:4900:5c74:	2001:b28:f23f:9	2001:b28:f23f:9::852	:437	TLSv1.2	2 257 A	pplication	Data					
	17 4.304911	2404:6800:4002:	2404:6800:4002:	2401:4900:5c74:	2401:4900:5c74:d9eb:	14ad:d3	TCP	86 4	43 → 63831	[ACK]	Seq=1 Ack=2 N	lin=265 Len=	0 SLE=1 SR	E	
	18 4.309219	2401:4900:5c74:	2401:4900:5c74:	2404:6800:4002:	2404:6800:4002:812::	2004	TCP	75 6	3830 → 443	[ACK]	Seq=1 Ack=1 N	√in=254 Len=	1 [TCP seg	mi	
	19 4.925475	2404:6800:4002:	2404:6800:4002:	2401:4900:5c74:	2401:4900:5c74:d9eb:	14ad:d3	TCP	86 4	43 → 63830	[ACK]	Seq=1 Ack=2 N	√in=280 Len=	0 SLE=1 SR	E:	
	20 4.925475	2001:b28:f23f:9	2001:b28:f23f:9	2401:4900:5c74:	2401:4900:5c74:d9eb:	14ad:d3	TCP	74 4	43 → 63807	[ACK]	Seq=227 Ack=4	484 Win=180	Len=0		
	21 5.851273	2001:b28:f23f:9	2001:b28:f23f:9	2401:4900:5c74:	2401:4900:5c74:d9eb:	14ad:d3	TLSv1.2	2 187 A	pplication	Data					
	22 5.903279	2401:4900:5c74:	2401:4900:5c74:	2001:b28:f23f:9	2001:b28:f23f:9::852	:437	TCP	74 6	3807 → 443	[ACK]	Seq=484 Ack=	340 Win=4185	Len=0		
<	23 6 01///10	107 168 738 76	107 168 738 76	107 168 738 101	107 168 738 101		DNC	76 0	tandand due	anu avi	6/ha AAAA wah	teleanam on	~	<b>`</b>	
	France All OC hosts	in- (coo bi	+-> of hut-s	ward (con hits)	a data fara Apadala	0000	10 -0 -0	05 30 50 k	- 26 00 4	0 40 6	2 96 44 69 94	01/ 2		-	
1	Frame 1: 86 byte	s on wire (688 Di	ts), 86 bytes capt	tured (688 Dits) d	in interface (Devic ~	0000	08 08 48	20 06 70 1	4 04 69 04	9 40 0	2 88 00 88 80		h.@		
2	Ethernet II, Src	: De:31:98:09:08:	62 (De:31:98:d9:d8	3:62), Dst: Azurev	lav_85:30:59 (d8:c0	0010	00 00 00	20 00 78 2	4 64 68 66	0 40 0 9 5c 7	2 08 12 00 00 4 d9 eb 14 ad		T.\+		Packet Bytes
2	Internet Protoco	I Version 6, Src:	2404:6800:4002:8	12::2004, DST: 246	1:4900:5C/4:d9eD:1	0030	d3 88 88	ff 4f c4 6	1 bb f9 53	2 1b 7	3 46 63 9e 51		·R·sFc·O		
Ť	Transmission Con	troi protocoi, sr	c Port: 445, Ust i	ront: 65826, Seq:	1, ACK: 1, Len: 0	0040	f2 a4 80	10 01 0d 1	0 72 00 00	0 01 0	1 05 0a 9e 51	r		~	<ul> <li>Hexbump / Raw</li> </ul>
	Source Port: 4	45		Dotails of an	vicelected	0050	f2 a3 9e	51 f2 a4				· · · Q · ·			Data appear in
	Destination Po	ol 03			yselected										this section
	[Stream index:	6]		<ul> <li>packet appe</li> </ul>	ars in this										
	[TCD Segment	compileceness: inc	ompiece (28)]	section											
	Lice segment L	entoj nul (noletiva													
	sequence numbe	relative	sequence number)		~										
<	7				>				Bashata 28471	Disealary 1	3F454 (105 551)			the Defect	
	<ul> <li>anarysis/sites.pcaping</li> </ul>								Packets: 35494 * 1	unspidyed: :	23434 (1007040)		Pro	me: persuit	

Fig 5. Packet Capturing Interface

Wireshark captured traffic can be seen as in the above figure. It mainly has the following sections:

- **1. Menu Options**: From here various menus can be selected like file menu, edit menu, analyze menu etc. to help in analysis , importing and exporting of captured traffic.
- **2. Filters:** In this section filters can be applied while capturing the packets. Specific protocol like http, tcp ,udp etc can be selected. Applying the filters helps in analyzing specific and relevant packets instead of all packets.
- **3. Packet Traffic List:** This section shows the various packets captured in the traffic with details like time, source address, destination address, protocol, descriptions etc.
- **4. Packet Details:** This section shows details regarding a particular selected packet like the frame details, TCP header details etc. From here various details can be seen regarding any packet like destination port, source port, length of packets, Bytes in the packets etc to help analyze the packet.
- **5.** Packet Bytes Hex Dump: This sections shows the hex dump of the packets or any raw data in human readable form.

#### 6 Results and Discussion

During the traffic analysis, traffic was captured from different sites and for a particular duration. This was done in order to check the scalability of the Wireshark depending on the number of sites searched, amount of activity done on those sites etc.

Following images show the summary of the captured traffic depending on the number of sites and amount of activity performed. It shows information regarding the capture, such as time remaining, packet and byte counts, and similar data [4].



Wireshark - Capture File Propert	ties - networkanalysis3sites.pcapng			- 0	×	Wireshark - Capture File Prope	rties - analysis5sites.pcapng			-	□ ×
Details						Detais					
File Name: Length: Hash (SHA256): Hash (RIPEMD160): Hash (SHA1): Format: Encapsulation:	E:\BI LAB FILES SEM 7\ne 9139 kB 1f8b517bc9afc143103bbcl e19987d3d38140971596e Sc5de04d4a6c5494431e41 Wireshark/ pcapng Ethernet	atworkanalysis3sitas.pcapng b426a376e92c4afd0c777e34 119523b9ae2c0435532 1b7975602088d9703c4	4bba9a13a589d14fa1		^	File Name: Length: Hash (SHA256): Hash (RIPEMD160): Hash (SHA1): Format: Encapsulation:	E:\BI LAB FILES SEM 7\a 32 MB 871118e24ba00934aa6b/ 039d7e839f0b7f1a9d9a8 1306a13840c0d6899d28! Wireshark/ pcapng Ethernet	nalysis5sites.pcapng id802704df2b1477b2dd7( cb6778b8fd1bde424df ff389763857b5978b12c	Ida8eaa9e21f25fef88d8eb		•
Time First packet: Last packet: Elapsed:	2023-09-01 19:28:56 2023-09-01 19:31:11 00:02:15					Time First packet: Last packet: Elapsed:	2023-09-01 19:33:30 2023-09-01 19:35:48 00:02:18				
Capture Hardware: OS: Application:	AMD Ryzen 5 3500U with 64-bit Windows 10 (22H2) Dumpcap (Wireshark) 4.0.	Radeon Vega Mobile Gfx (w ), build 19045 8 (v4.0.8-0-g81696bb7485;	th SSE4.2) ')			Capture Hardware: OS: Application:	AMD Ryzen 5 3500U with 64-bit Windows 10 (22H2 Dumpcap (Wireshark) 4.0	Radeon Vega Mobile Gfx !), build 19045 .8 (v4.0.8-0-g81696bb74	(with SSE4.2) 857)		
Interface WI-FI	Dropped packets 0 (0.0%)	Capture filter none	Link type Ethernet	Packet size limit (snaplen) 262144 bytes		Interface Wi-Fi	Dropped packets 0 (0.0%)	Capture filter none	Link type Ethernet	Packet size limit (snap) 262144 bytes	len)
Statistics Measurement Packets Time span, s Average pps Average packet size, B Bytes Average bits/s C Capture file comments	Captured 10488 135.021 77.7 838 8788256 65 k 520 k	Displays 10488 ( 135.021 77.7 838 878825 65 k 520 k	d 100.0%) 5 (100.0%)	Marked 	* }	Subbols     Measurement     Packets     Time span, s     Average pos     Average pos     Average bytes     Average bytes/s     Average bits/s     Copture file comments	Capturad 35494 138.526 256.2 870 30866668 222 k 1782 k	Displ 3549 138.1 256.5 870 3086 222 1 1782	nyad 4 (100.0%) 126 1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	Marked    0  	*
Refresh			Save Co	mments Close Copy To Clipboard	Help	Refresh			Save C	Comments Close Copy To Clipboard	неір

Fig 6. Capture File Properties – 1

Fig 7. Capture File Properties – 2

Name:         E-(B1 LAB FLES SEM 7 Janabysis118kes.pogng Length:         II MB           Het         Stription:         11 MB           Het (SVE5):         Stription:         11 MB           Frank:         Weishaw,	Wireshark - Capture File Proper	rties - analysis11sites.pcapng					-		$\times$
File         File           Name:         E-UBI LAB FILES SEM Tysekykist158es.pr.apng Langht:         11 MB           Hahr (SK425):         557-87-223exd37c45e_2070391c1510597-454-320353879.pdf6454-2428.2d496           Hahr (SK425):         527-87223exd37c45e_2070391c1510597-454-320353879.2df6454-2428.2d496           Hahr (SK425):         527-87223exd37c45e_2070391c1510597-454-320353879.2df6454-2428.2d496           Hahr (SK415):         2273581f543/3d980553733939315518/641d2982           Format:         Wireshark/ r.opng           Encopulation:         Elbernet           Time         Filt sockat:           Part sockat:         000:02:11           Capture         Hardware:           AMD Ryan 5 35000 with Radeen Vega Mobile GK (with 5554.2)           OS:         64-bit Windows 10 (22142), build 19045           Application:         Dumpcap (Wireshark) 4.0.8 (v4.0.8-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0	Details								
Langth:         11 M8           Hach (SH225):         11 M8           Hach (SH225):         527547543649642071591121559734947303535797846465423236496           Hach (SH215):         22753515543649604274964642436664835979           Hach (SH215):         17 dc:173349555573393978044274966484359835979           Hach (SH215):         17 dc:1733495555733939780442749664843202           Format:         Wirebark/ = r.o.gong           Encopulation:         Element           Part packet:         2023-09-01 19-37-03           Last packet:         00-02-11           Cotton:         Dumpcap (Wreshark) 4.0.8 (vd.0.8 -0g456660-74857)           Hardware:         A-D.D.D.mpcap (Wreshark) 4.0.8 (vd.0.8 -0g456660-74857)           Statico         Displayed         Packets bit for the static	File Name:	E:\BI LAB FILES SEM 7\a	analysis11sites.pcapr	na					^
Time           Trist spacial:         2023-09-01.19-37-03           List spacial:         2023-09-01.19-37-03           Bapadel:         00:02:11           Capture	Length: Hash (SHA256): Hash (RIPEMD160): Hash (SHA1): Format: Encapsulation:	11 MB 657e47229add3fc45e20f 2e783e1f543de9e09d44 f1dc13f33abfb85a5f3b35 Wireshark/ pcapng Ethernet	3291c191b597454f3 2f48e6ebea3d5f8830 a913ab518d641d29	- 8d8353e79a0f6fe54b42b3 179 82	cf496				
First packet:         2023-09-01 19:37:03           Last packet:         2023-09-01 19:37:03           Elspacet:         00:02:11           Copbure	Time								
Capture           Hardware:         AHD Ryan 5 3500U with Roden Vega Mobile GK (with 5554.2) .08:         Get bit Windows 10 (2212); buil 15945           Application:         Dumpare (Wireshan) 4.0.3 (vk.0.0-4-gd 656bb/r487)           Interface         Dropped packets           Distante         Capture filter           UHF         0 (0.0%)           Reset size limit (anglen)           Average park         10251           1035         1035.100.0%)           Average park         101251           Average park         1114438           11144438         1114438 (100.0%)           Average parket size, B         611	First packet: Last packet: Elapsed:	2023-09-01 19:37:03 2023-09-01 19:39:15 00:02:11							
introducers:         AVID Syzan 3 35000 with Racken Vaga Mobile GR (with SSE4.2)           OG:         64-bit Windows 10 (2242), build 19045           Application:         Dumpcap (Wireshank) 4.0.8 (v4.0.8-0-gB1696bb74857)           Interfaces         Inferiance         Dispage (Wireshank) 4.0.8 (v4.0.8-0-gB1696bb74857)           Interfaces         Extension         Dispage (Wireshank) 4.0.8 (v4.0.8-0-gB1696bb74857)           Interfaces         Extension         Dispage (Wireshank) 4.0.8 (v4.0.8-0-gB1696bb74857)           Statictic         Dispage (Wireshank) 4.0.8 (v4.0.8-0-gB1696bb74857)         Packet size limit (snaplein) 2.82144 bytes           Statictic         Dispage (Wireshank) 4.0.8 (v4.0.8-0-gB1696bb74857)         Dispage (Wireshank) 4.0.8 (v4.0.8-0-gB1696bb74857)           Mass.emment         Captures 1         Displayed         Marking           Message basis         Displayed         Displayed (Wireshank) 4.0.8 (v4.0.8-0-gB1696bb74857)           Message basis         Displayed (Mireshank) 4.0.8 (v4.0.8-0-gB1696bb74857)         Displayed (Mireshank) 4.0.8 (v4.0.8-0-gB1696bb7487)           Message basis         Displayed (Mireshank) 4.0.8 (v4.0.8-0-gB1696bb74857)         Displayed (Mireshank) 4.0.8 (v4.0.8-0-gB1696bb7487)           Message basis         Displayed (Mireshank) 4.8 (v4.0.8-0-gB1696bb7487)         Displayed (Mireshank) 4.8 (v4.0.8-0-gB1696bb7487)         Displayed (Mireshank) 4.8 (v4.0.8-0-gB1696bb7487) <t< td=""><td>Capture</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>	Capture								
Interfaces         Disopped packets         Capture filter         Link type         Packet size limit (nonlem)           WiFi         0 (0.0%)         Capture filter         Link type         Packet size limit (nonlem)           Statistics	Hardware: OS: Application:	AMD Ryzen 5 3500U with 64-bit Windows 10 (22H Dumpcap (Wireshark) 4.0	n Radeon Vega Mobi 2), build 19045 0.8 (v4.0.8-0-g8169	le Gfx (with SSE4.2) 5bb74857)					
Interface         Decoped packets         Capture filter         Link type         Packet size limit (ronglem)           Statistics	Interfaces								
Statistics           Statistics           Messacrement         Captured         Marked           Products         18251         10251         100.0%)           Time spin 1         131.516         -           Anerogo point         5         135.5         -           Points         131.516         -         -           Points         131.516         -         -           Points         50.5         -         -           Points         51.5         135.5         -           Points         51.5         135.5         -           Points         51.6         134.6         -           Averago bytexit         64.k         84.k         -           Averago bytexits         676.k         676.k         -           Course the connexits         -         -         -	Interface Wi-Fi	Dropped packets 0 (0.0%)	Capture filter none	Link type Ethernet		Packe 2621	t size limit (snaple 44 bytes	en)	
Measurement         Captured         Dipolayed         Marked           Pockats         10251         10251         10251         10251           Time spon, s         131.616	Statistics								
Time gap p. s         131.815         131.816         -           Arenga p. s         131.815         131.816         -           Arenga p. s         138.5         138.5         -           Arenga politic size, B         611         611         -           Priors         11144438         11144438 (100.0%)         0           Arenga hytic/s         64 k         64 k         -           Arenga hytic/s         676 k         -         -           Option         676 k         -         -           Option         Comments         -         -	Measurement	Captured		Displayed 18251 (100.0%)	Mark	ed			
Aversge pot 138.5 Aversge pot 138.5 Bytes 5.4	Time span, s	131.816		131.816	-				
And Edge Docks Sub, 5         0         11144438         01114438	Average pps	138.5		138.5	-				
Average byte/s B4 k B4 k - Average byte/s 676	Average packet size, b Bytes	11144438		11144438 (100.0%)	0				
Average Dist/s         676 k	Average bytes/s	84 k		84 k	-				
C Copyre Re connexts Copyre Copyoret Help	Average bits/s	676 k		676 k	-				~
Refresh Save Commetts Class Copy To Opboard Help	Capture file comments								,
Refresh Sew Comments Class Copy To Opboard Help									
Refresh Save Comments Close Copy To Clipboard Help									
	Refresh				Save Comments	Close	Copy To Clipboard	Help	

Fig 8. Capture File Properties – 3

Below table shows the comaprison of the above captured traffic :

Sr. No.	No of Sites	Amount of Activity	Duration (in minutes)	Size of the file	Dropped Packets	Average pps	Captured traffic packets
1	3	Low	02:15	8925KB	0	77.7	10488
2	5	Moderate	02:18	30MB	0	256.2	35494
3	11	Low	02:11	11MB	0	138.5	18251

 Table 2. Comparison Of Above Captured Traffic



												-	D X
Edit View Go Cap	ture Analyze Statistics T	felephony Wireless Tools	Help										
🔳 🧟 🔘 📕 🗋 🗙	🖉 🤇 🗰 🖷 著 🛓												
pp)y a display filter <ctri- <="" th=""><th>×</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></ctri->	×												
Time	Source	Source MAC	Destination	Destination MAC		Protocol	Leng	th Info					
10 2.067267	2401:4900:5c74:_	. 2401:4900:5c74:	2001:b28:f23f:9	2001:b28:f23f:9:	:852:437	TCP		74 63807	→ 443 [ACK	] Seq=301 Ac	k=227 Win=4186	Len=0	
11 2.845569	2401:4900:5c74:_	. 2401:4900:5c74:	2404:6800:4002:	2404:6800:4002:8	19::200a	TCP		75 63827	→ 443 [ACK	] Seq=1 Ack=	1 Win=257 Len=:	L [TCP	segm
12 2.845570	2401:4900:5c74:_	. 2401:4900:5c74:	2404:6800:4002:	2404:6800:4002:8	1a::200a	TCP		75 63829	→ 443 [ACK	] Seq=1 Ack=	1 Win=258 Len=:	L [TCP	segm
13 3.697065	2404:6800:4002:_	. 2404:6800:4002:	2401:4900:5c74:	2401:4900:5c74:d	9eb:14ad:d	3 TCP		86 443 +	63829 [ACK	] Seq=1 Ack=	2 Win=271 Len=0	SLE=1	SRE:
14 3.697065	2404:6800:4002:_	. 2404:6800:4002:	2401:4900:5c74:	2401:4900:5c74:d	9eb:14ad:d	3 TCP		86 443 +	63827 [ACK	] Seq=1 Ack=	2 Win=265 Len=0	3 SLE=1	SRE
15 3.800514	2401:4900:5c74:_	. 2401:4900:5c74:	2404:6800:4002:	2404:6800:4002:8	9b::2003	TCP	1000	75 63831	→ 443 [ACK	] Seq=1 Ack=	1 Win=258 Len=1	L [TCP	segm
16 4.303139	2491:4999:5c74:	2401:4900:5c74:	2001:h28:f23f:9	2001:h28:f23f:9:	857:437	TI SV1	1.7 2	57 Annlia	ation Data				
ame 11: 75 byt	tes on wire (600 b	pits), 75 bytes cap	otured (600 bits)	on interface \Dev	ice 8000	be 3f 9	8 d9 d8 6	2 d8 c0	a6 85 30	59 86 dd 60	06 ·?···b··	- 0Y	•U
chernet II, Sro	:: AzureWav_85:30:	:59 (d8:c0:a6:85:30	9:59), Dst: be:3f	98:d9:d8:62 (be:3	f:9 8018	41 62 0	0 15 06 f	e 24 01	49 00 5c	74 d9 eb 14	ad Ab \$ 1	[.\t	53. 
ternet Protoco	ol Version 6, Src:	: 2401:4900:5c74:d	eb:14ad:d388:88f	:4fc4, Dst: 2404:	686 8828	d3 88 8	8 ff 4f c	4 24 04	68 00 40	02 08 19 00	00 0.\$. 1	1.6	
ansmission Con	ntrol Protocol, Sr	rc Port: 63827, Dst	: Port: 443, Seq:	1, Ack: 1, Len: 1	0030	00 00 0	0 00 20 0	a 19 53	01 bb db	0d 47 82 ef	d3 ···· S ·	G	
Source Port:	63827				0040	bd 83 5	0 10 01 0	1 86 57	69 69 69		· · P · · · · W		
Destination P	ort: 443												
[Stream index	: 3]												
[Conversation	completeness: Inc	complete (28)]											
TCP Segment	Len: 1]												
Sequence Numb	er: 1 (relative	e sequence number)											
Sequence Numb	er (raw) · 3675080	578											
[Next Sequence	a Number: 2 (re	alative sequence n	umban)]										
Acknowledgeen	t Number: 1 (n	alative ack number	under /j										
Acknowledgmen		elacive ack number	)										
Acknowledgmen	c number (raw): 4	025051255											
0101 = H	eader Length: 20 (	bytes (5)											
Flags: 0x010	(ACK)												
Window: 257													
[Calculated w	indow size: 257]												
[Window size	scaling factor: -:	1 (unknown)]											
Checksum: 0x8	b57 [unverified]												
[Checksum Sta	tus: Unverified]												
Urgent Pointe	r: 0												
[Timestamps]													
[SEQ/ACK anal	ysis]												
TCP payload (	1 byte)												
TCP segment d	ata (1 byte)												
-					>								

The above image shows the TCP Header details for the **packet number 11** captured.

Clearly from above: Source Port: 63827 Destination Port: 443 Sequence Number: 3675080578 Acknowledgment Number: 4023631235 Data Offset: 0101 Flag: ACK Window: 257

#### Applying filters and analysing the packets:

In this traffic, protocol filter of HTTP was applied, which resulted in showing the packets related to http protocol only as shown in below figure.

analysis5sites.pcap.pd								-	a
le Edit View Go Canture Analyze Statistics To	alaphony Wiralace Toole Hale	0							
		P							
									(2) [==]
Time Eautre	Farmer MIC	Destination	Particulies MAC	Bratacal	Length Tofo				GD 00.0
22050 100 260008 102 168 228 76	102 168 238 76	152 105 39 76	152 105 39 76	HTTP	378 GET /DigiContTL	CDCACWADEEDO	20CA1 ont WITE	/1 1	
23030 100.200500 152.100.230.70	152.100.230.70	102.155.50.70	102.159.30.70	UTTO	378 GET / Digicercit	(applicati	op/pkiv-cont)	/1.1	
231/3 100.304040 132.133.30.70	102.159.30.70	152.100.250.70	152.105.258.70	HITE	271 GET (DisiGastG)	(appricaci	ton/pkix-cerc)		
23451 100.553654 152.100.256.70	152.100.230.70	102.195.50.70	102.195.50.70	UTTO	1227 HTTP/1 1 200 OF	(applicati	ion (nkiv-cont)		
23022 200.003024 232.233.30.70	152.155.56.76	192.100.290.70	192.100.290.70		1557 1117/1.1 200 00	(appricaci	ton/prix-cerc/		
rame 23050: 378 bytes on wire (3	024 bits), 378 bytes	captured (3024	bits) on interface \Dev	ice\NPF_{4885E2	FA-C7CA-4597-97A7-5C1( ^	0000 be 3	3+ 98 d9 d8 62	00 8L	a6 8
thernet II, Src: AzureWav_85:30:	59 (d8:c0:a6:85:30:5	9), Dst: be:3†:9	98:d9:d8:62 (be:3†:98:d9	:d8:62)		0010 01 6	5C 4a 04 40 00	50 00	40 0
nternet Protocol Version 4, Src:	192.168.238.76, Dst	: 152.195.38.76				0020 20 4	1 67 00 00 00	17 45	54 2
ransmission Control Protocol, Sr	c Port: 64127, Dst P	ort: 80, Seq: 1,	Ack: 1, Len: 324			0040 65 7	72 74 54 4c 53	52 53	41 5
Source Port: 64127						0050 30 3	32 30 43 41 31	2e 63	72 7
Destination Port: 80						0060 31 2	e 31 0d 0a 48	6f 73	74 3
[Stream index: 282]						0070 74 7	73 2e 64 69 67	69 63	65 7
[Conversation completeness: Inc	complete, DATA (15)]					0080 0a 4	13 6f 6e 6e 65	53 74	69 E
[TCP Segment Len: 324]						0090 70 2	2d 61 6c 69 76	55 Ød	0a 5
Sequence Number: 1 (relative	e sequence number)					00a0 65 6	5e 74 3a 20 4d	6f 7a	69 <del>6</del>
Sequence Number (raw): 41814955	578					00b0 20 2	28 57 69 6e 64	6f 77	73 2
[Next Sequence Number: 325 (	relative sequence nu	mber)]				00c0 30 3	3b 20 57 69 6e	36 34	3b 2
Acknowledgment Number: 1 (re	lative ack number)					00d0 70 7	70 6c 65 57 65	62 4b	69 7
Acknowledgment number (raw): 27	44341742					00e0 36 2	20 28 4b 48 54	4d 4c	2c 2
0101 = Header Length: 20 b	ytes (5)					00f0 65 6	53 6b 6f 29 20	43 68	72 6
Flags: 0x018 (PSH, ACK)						0100 2e 3	30 Ze 30 Ze 30	20 53	61 6
Window: 257						0110 37 2	e 33 36 0d 0a	41 63	63 6
[Calculated window size: 65792]						0120 67 6	04 09 08 67 3a	20 67	/a e
[Window size scaling factor: 25	6]					0140 67 7	01 /4 05 00 0a	*1 63	03 0
Checksum: 0x67e0 [unverified]	dow size scaling factor: 256] ksum: 0x67e0 [unverified]							20 30	20 6
[Checksum Status: Unverified]	sum: bxb/e0 [unverified] :ksum Status: Unverified]							6e 3h	71 3
Urgent Pointer: 0	T Pointer: 0								ed e
<pre>     [Timestamps] </pre>					2	507	2 50 50 20 50		
						<			
7 Deserved Transfer Destantic Destant					Redutes 25424 - Divisional	4 (0.05/)			II putter

Fig 10. HTTP Protocol Captured Packets



E-ISSN: 2582-2160 • Website: www.ijfmr.com • Email: editor@ijfmr.com

analysisSsites.pcapng							-		×
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help									
📶 🔳 🖉 🐵 🔰 🖄 🖄 🔍 🖛 🍁 🖀 🛊 👱 🜉 🔍 🔍 🎘 🗮									
http								× •	+
No. Time Source Source MAC Destination Destination MAC	Pr	otocol	Length Ir	fo				_	^
1		TTO	270.0	PT (P1-1)				,	~
	0000	40 54		24 25 24	20.22.20	20.00.45.45.0		200.01	
> Frame 231/3: 335 bytes on wire (2680 bits), 335 bytes captured (2680 bits) on interface \Devic	0000	48 54	54 50 ZT	31 Ze 31	20 32 30	50 20 41 40 00	HITP/1.1	200 UK	
> Ethernet II, Src: be:31:98:d9:d8:62 (be:31:98:d9:d8:62), Dst: AZUreWav_85:30:59 (d8:c0:a6:85:3	0010	62 79	74 65 73	70 74 20 0d 0a 41	67 65 3	20 31 32 38 34	hytes:	Ranges:	84
> Internet Protocol Version 4, Src: 152.195.38.76, DSt: 192.168.238.76	0020	35 36 6	Ad 0a 63	61 63 68	65 2d 63	6f 6e 74 72 6f	56 cach	e-contr	~~
> Transmission Control Protocol, Src Port: 80, DSt Port: 6412/, Seq: 1319, Ack: 325, Len: 281	0040	6c 3a	20 6d 61	78 2d 61	67 65 30	31 37 32 38 36	) 1: max-a	ge=1728	30
<pre>&gt; [2 Reassembled ICP Segments (1599 bytes): #231/1(1518), #231/3(281)]</pre>	0050	30 2c	20 70 75	62 6c 69	63 0d 0a	43 6f 6e 74 65	0, publi	c · · Cont	te
Hypertext Transfer Protocol	0060	6e 74 3	2d 54 79	70 65 3a	20 61 70	70 6c 69 63 61	nt-Type:	applic	a
HIP/I.I 200 UKIVIN	0070	74 69 0	6f 6e 2f	70 6b 69	78 2d 63	65 72 74 0d 0a	tion/pki	x-cert	
Accept-Kanges: bytes/r/n	0080	44 61	74 65 3a	20 46 72	69 2c 20	30 31 20 53 65	Date: Fr	i, 01 S	ŝe 🛛
Age: 128456 \r\n	0090	70 20 3	32 30 32	33 20 31	34 3a 30	) 35 3a 31 31 20	p 2023 1	4:05:11	L
cache-control: max-age=1/2800, public\r\n	00a0	47 4d 9	54 0d 0a	45 74 61	67 3a 20	22 35 66 36 64	GMT··Eta	g: "5f6	5d
Content-Type: application/pkix-cert/r\n	0000	34 34 3	31 30 2d	34 65 65	22 Ød Øa	65 78 70 69 72	4410-4ee	"··expi	In
Date: Fr1, 01 Sep 2023 14:05:11 GMI (P\n	0000	22 20 2	3a 20 53	75 68 2C	20 30 33	20 53 65 70 26	es: Sun,	03 Sep	SM
ETag: "55604410-446" (r\n	0000	54 04 0	32 33 20 9a 6c 61	51 54 5a 73 74 5d	50 55 56 64 6 <del>f</del> 64	1 51 51 20 47 40 1 69 66 69 65 67	2025 14: Tulacta	modifie	and be
expires: Sun, 03 Sep 2023 14:05:11 GM/(r/n	0010	3a 20 4	46 72 69	20 20 32	35 20 53	65 70 20 32 30	Eri. 2	5 Sen 2	20
last-modified: Fr1, 25 Sep 2020 01:12:48 GMI/r/n	0100	32 30 2	20 30 31	3a 31 32	3a 34 38	20 47 4d 54 0d	20 01:12	:48 GMT	r ·
Server: ELACC (hd1/b314)(r\n	0110	0a 53 (	65 72 76	65 72 3a	20 45 43	41 63 63 20 28	·Server:	ECAcc	(
X-Cache: HII(\n	0120	6e 64 (	6c 2f 44	33 31 34	29 0d 0a	58 2d 43 61 63	nd1/D314	)··X-Ca	ac
> Content-Length: 1262\P\n	0130	68 65 3	3a 20 48	49 54 Ød	0a 43 6f	6e 74 65 6e 74	he: HIT	Conter	ht
	0140	2d 4c 6	65 6e 67	74 68 3a	20 31 32	36 32 0d 0a 0d	I -Length:	1262 · ·	
[HIIP response 1/2]	0150	0a 30 1	82 04 ea	30 82 03	d2 a0 03	02 01 02 02 10			
[IIme since request: 0.123138000 seconds]	0160	0a 35 (	08 d5 5c	29 2b 01	7d f8 ac	65 c0 0f f7 e4	· · 5 · · \)+·	}···e···	1
[Request in Trame: 23050]	0170	30 0d (	06 09 Za	86 48 86	+7 0d 01	01 05 05 00 30	0		0
[Next request in trame: 23451]	0180	61 31 6	00 30 09	06 03 55	04 06 13	67 60 43 65 7	, a1.0U	DiniCa	1.
INEXT response in trame: 238221	0190	74 20	10 60 62	21 10 20	17 06 03	5 67 69 45 65 74	t Tnc1.4	DIGICE	11°
[Request UK1: http://dacerts.digicert.com/DigicertiLSKSASHA2562020LAI.crt]	alba	77 77	77 20 64	69 67 69	63 65 73	74 20 63 6f 60	una digi	cent co	
File Data: 1262 bytes	0100	31 20	30 1e 06	03 55 04	03 13 17	44 69 67 69 43	1 0 U	···Digi	iC
PRIX CERT FILE Format Prix Cert File Format Prix Cert File Format	01d0	65 72	74 20 47	6c 6f 62	61 6c 20	52 6f 6f 74 20	ert Glob	al Root	2
<ul> <li>Cercificate (id-ac-commonwame=uigicert iLS RSA SMA256 2020 CAI,id-at-organizationwame=uigicert incodentiations</li> </ul>	01e0	43 41	30 1e 17	0d 32 30	30 39 32	34 30 30 30 30	CA0 · · · 20	0924000	90
<pre>&gt; signed.ertities(is) (shall(ith)CAF=sumtise)</pre>	01f0	30 30 5	5a 17 0d	33 30 30	39 32 33	3 32 33 35 39 35	00Z · · 300	9232359	95
Paddian A	0200	39 5a 3	30 4f 31	0b 30 09	06 03 55	04 06 13 02 55	9Z001 · 0 ·	· · U · · · ·	U
radualig. 0	0210	53 31 3	15 30 13	06 03 55	04 0a 13	0c 44 69 67 69	s1.0U	····Dig	gi v
encrypted. //a00//a2/30ae0010/1e03a30030488030/11/00224/TC489Te000184048183600349	<								>
	Frame (33	35 bytes)	Reassembled T	CP (1599 bytes)					
The second				ackate: 25404 -	Dicalmond: 4 (0.00	51		Profile: F	Anderson

Fig 11. HTTP Protocol Packet Details

On selecting the Hypertext Transfer Protocol in Packet Details section on **packet number 23173** it shows the HTTP header details. From above it can be seen that this packet contain details regarding the certificate.

From above following details can be figured out:

- Request URI: http://cacerts.digicert.com/DigiCertTLSRSASHA2562020CA1.crt
- Algorithm used for encryption: SHA 256 along with RSA
- serialNumber: 0x0a3508d55c292b017df8ad65c00ff7e4
- Algorithm Id: 1.2.840.113549.1.1.11 (in this SHA256 along with RSA encryption is used)

Such field can be used by attackers in wrong way which can also result in various attacks. This shows that if any http site is used by any user, and if any attacker had access to their network they can easily trace these packets and manipulate the data within the packets, **resulting in the Man –in – the- middle or redirecting attack.** 

#### **Exploring the Statistics on the Captured Traffic:**

**Protocol Hierarchy:** Protocol Hierarchy display the number of packets and number of bytes in those packets for various protocols that were captured during for network analysis all the protocols are arranged in the same hierarchy as they were found in the traffic. It provides the count of packets in which the protocol is present and the packet in which it is the last protocol in the stack. These last-protocol counts let you know how many packets—along with the corresponding byte count—ended in a certain protocol. They are listed under "End Packets" and "End Bytes" in the table[4].



E-ISSN: 2582-2160 • Website: www.ijfmr.com

• Email: editor@ijfmr.com

otocol	Percent	Packets	Packets	Percent	Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs		
Frame		100.0	35494		100.0	30866668	1782 k	0	0	0	35494		
✓ Ethernet		100.0	35494		1.6	496916	28 k	0	0	0	35494		
<ul> <li>Internet Protocol Version 6</li> </ul>		45.1	16005	i	2.1	540200	36 k	0	0	0	16005		
<ul> <li>User Datagram Protocol</li> </ul>		12.5	4454		0.1	35632	2057	0	0	0	4454		
Session Traversal Utilities for NAT		0.0	2		0.0	54	3	2	64	3	2		
QUIC IETF		12.0	4267		9.3	2855724	164 k	4267	2758370	159 k	4435		
Multicast Domain Name System		0.0	8		0.0	1092	63	8	1092	63	8		
Data		0.5	177		0.4	112628	6504	177	112628	6504	177		
<ul> <li>Transmission Control Protocol</li> </ul>		32.5	11539		36.9	11403835	658 k	7902	6595955	380 k	11539		
XMPP Protocol		0.0	1		0.0	376	21	1	376	21	1		
Transport Laver Security		10.1	3591		38.9	11994249	692 k	3591	8292463	478 k	4169		
Malformed Packet		0.0	1		0.0	)	0	1	0	0	1		
Data		0.1	44		0.0	44	2	44	44	2	44		
Internet Control Message Protocol v6		0.0	12		0.0	384	22	12	384	22	12		
<ul> <li>Internet Protocol Version 4</li> </ul>		54.9	19479		1.3	389580	22 k	0	0	0	19479		
<ul> <li>User Datagram Protocol</li> </ul>		13.2	4675		0.1	37400	2159	0	0	0	4675		
Simple Service Discovery Protoco	_	0.0	10		0.0	1750	101	10	1750	101	10		
Session Traversal Utilities for NAT		0.0	4		0.0	104	6	4	104	6	4		
QUIC IETF		8.4	2968		7.6	2333034	134 k	2968	2274537	131 k	3055		
Multicast Domain Name System	-	0.0	8	_	0.0	1092	63	8	1092	63	8		
Domain Name System	1	4.7	1685		0.5	142904	8252	1685	142904	8252	1685		
<ul> <li>Transmission Control Protocol</li> </ul>		41.7	14800		40.5	12491174	721 k	10150	8153485	470 k	14800		
Transport Layer Security		13.0	4598		41.7	12861883	742 k	4598	9210008	531 k	5008		
<ul> <li>Hypertext Transfer Protocol</li> </ul>		0.0	4		0.0	3523	203	2	641	37	4		
PKIX CERT File Format		0.0	2		0.0	2209	127	2	2209	127	2		
Domain Name System		0.0	2		0.0	516	35	2	616	35	2		
Data		0.1	46		0.0	46	2	46	46	2	46		
<ul> <li>Internet Control Message Protocol</li> </ul>		0.0	4		0.0	2224	128	0	0	0	4		
OUIC IETF		0.0	4		0.0	2080	120	4	2080	120	4		
Address Resolution Protocol		0.0	10		0.0	280	16	10	280	16	10		

Fig 12. Protocol Hierarchy Statistics

Flow Graph: Flow graph shows the connection between the hosts. For each connection that was captured it shows the packet timing, direction, ports, and comments. It provides filters like ICMP (Internet Control Message Protocol) flows, ICMPv6 flows, UIM flows, and TCP flow [4]. The flow graph window provides different controls based on that. With the help of flow graph you can easily figure out various port numbers and IP addresses and thus can easily get to get know if any unusual port number or IP address occurs in the traffic [4]. The below figure shows the flow graph for the captured traffic.



# Fig 13. Flow Diagram

IO GRAPH: Display the number of packets or the amount of bytes per second for all packets that match the chosen filter. By default, only one graph displaying the number of packets per second will be shown [4].



E-ISSN: 2582-2160 • Website: <u>www.ijfmr.com</u> • Email: editor@ijfmr.com



Fig 14. I/O Graphs

**HTTP -> Packet Counter:** The packet counter the data regarding the HTTP packets. From here, we can analyse if there was any redirection or any kind of error. It helps in knowing if any attack like DDOS attack took place, or were any packets redirected to any other unusual address. From the below figure, we analysed that all the packets have 2xx Response, indicating that all packets were transmitted successfully and no packet was dropped.

Wireshark · Packet Counter ·	analysis	osites.pcap	ng					-		×
Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start		
<ul> <li>Total HTTP Packets</li> </ul>	14				0.0001	100%	0.0100	17.488		
Other HTTP Packets	0				0.0000	0.00%	-	-		
<ul> <li>HTTP Response Packets</li> </ul>	5 2				0.0000	14.29%	0.0100	100.384		
???: broken	0				0.0000	0.00%	-	-		
5xx: Server Error	0				0.0000	0.00%	-	-		
4xx: Client Error	0				0.0000	0.00%	-	-		
3xx: Redirection	0				0.0000	0.00%	-	-		
<ul> <li>2xx: Success</li> </ul>	2				0.0000	100.00%	0.0100	100.384		
200 OK	2				0.0000	100.00%	0.0100	100.384		
1xx: Informational	0				0.0000	0.00%	-	-		
<ul> <li>HTTP Request Packets</li> </ul>	12				0.0001	85.71%	0.0100	17.488		
SEARCH	10				0.0001	83.33%	0.0100	17.488		
GET	2				0.0000	16.67%	0.0100	100.261		
isplay filter:									Apply	/
							Сору	Save as	Close	,

Fig 15. Packet Counter

# 7 Conclusion

In this paper, we examined network traffic analysis and its significance. The document also includes information and instructions for capturing traffic with Wireshark. The paper discusses how Wireshark can assist security and network administrators with packet capture and analysis. This paper demonstrates how



graphs such as the flow graph and the IO graph may be plotted and used to investigate captured traffic. We learned about the TCP header and how to trace IP addresses, port numbers, sequence numbers, and other information from captured traffic. Wireshark's efficiency was examined as traffic increased. Wireshark is a very significant tool in network traffic analysis, and if used properly, it may assist administrators notice any suspicious or anomalous activity in the network in real time, allowing them to take appropriate action to prevent any attacks.

#### References

- 1. Dodiya, Bindu, and Umesh Kumar Singh. "Malicious Traffic analysis using Wireshark by collection of Indicators of Compromise." *Int J Comput Appl* 183.53 (2022): 1-6.
- 2. Jain, G. "Application of snort and wireshark in network traffic analysis." *IOP Conference Series: Materials Science and Engineering*. Vol. 1119. No. 1. IOP Publishing, 2021.
- 3. Alfawareh, Muhamed. "A deeper Look into Network Traffic Analysis Using Wireshark." (2015).
- 4. *Wireshark(1) Manual Page*. wireshark(1). (n.d.). https://www.wireshark.org/docs/man-pages/wireshark.html
- 5. Molenaar, R. (2019, October 21). *TCP header*. NetworkLessons.com. https://networklessons.com/cisco/ccie-routing-switching-written/tcp-header