

Wireshark

A Pentester Guide

Table of Contents

Abstract3
Introduction4
Network Packet Forensic4
Examine Layers captured by Wireshark5
Ethernet Header (Data Link)6
IP Header (Network Layer)7
TCP Header (Transport Layer)
Structure of TCP segment
Different Types of TCP flags10
Password Sniffing12
Capture HTTP Password12
Dissect HTTPS Packet Captures14
Capture Telnet Password
Capture FTP Password19
Capture SMTP password20
Analyzing SNMP Community String23
Capture MSSQL Password25
Capture PostgreSQL Password
Creating Firewall Rules with Wireshark
Conclusion
References

<u>Abstract</u>

Many people wonder if Wireshark can capture passwords. The answer is undoubtedly yes! Wireshark can capture not only passwords, but any type of data passing through a network – usernames, email addresses, personal information, pictures, videos, or anything else.

Wireshark can sniff the passwords passing through as long as we can capture network traffic. But the question is, what kind of passwords are they? Or, more precisely, which network protocols' passwords can we obtain? That is the subject of this report.

Disclaimer: This report is provided for educational and informational purpose only (Penetration Testing). Penetration Testing refers to legal intrusion tests that aim to identify vulnerabilities and improve cybersecurity, rather than for malicious purposes.

Introduction

In the first section of this report, we'll delve into 'Network Packet Forensics,' exploring vital aspects such as data transfer between nodes, the 'OSI 7-layer model,' and how Wireshark stores information across layers when capturing network traffic.

Moving on to the second part, we'll examine how Wireshark can capture passwords, a result of certain unencrypted network protocols known as clear text protocols. These protocols expose all data, including passwords, making it visible to anyone with the ability to intercept the communication, such as a man-in-the-middle.

<u>Network Packet Forensic</u>

As we know for transferring the data from one system to other, we need a network connection which can be wired or wireless connection. But in the actual transmission of data does not only depend upon network connection apart from that it involves several phases for transmitting data from one system to another which was explained by the OSI model.

OSI stands for **O**pen **S**ystems Interconnection model which is a conceptual model that defines and standardizes the process of communication between the sender's and receiver's system. The data is transfer through 7 layers of architecture where each layer has a specific function in transmitting data over the next layer.

Now have a look over given below image where we had explained the functionality of each layer in the OSI model. So, when data is transmitted by sender's network then it will go in downward direction and data move from application layer to physical layer whereas when the receiver will receive the transmitted data it will come in an upward direction from physical layer to application layer.

Flow of Data from Sender's network: **Application** > **Presentation** > **Session** > **Transport** > **Network** > **Data Link** > **Physical**

Flow of Data from Receiver's network: **Physical** > **Data Link** > **Network** > **Transport** > **Session** > **Presentation** > **Application**

OSI Model



Examine Layers captured by Wireshark

Basically, when a user opens an application for sending or receiving Data then he directly interacts with the application layer for both operations either sending or receiving of data. For example, we act as a client when use Http protocol for uploading or Downloading a Game; FTP for downloading a File; SSH for accessing the shell of the remote system.

While connecting with any application for sharing data between server and client we make use of Wireshark for capturing the flow of network traffic stream to examine the OSI model theory through captured traffic.

From given below image you can observe that Wireshark has captured the traffic of four layers in direction of the source (sender) to destination (receiver) network.

Here it has successfully captured **Layer 2** > **Layer 3** > **Layer 4** and then **Layer 7** information.

				b		6	۹, ۱	+ +	Ç .	•	⇒		€	ର୍	۹.			
	ip.addr == 192.168.1.101																	
No).		Time	Source	e		Dest	ination			Protoc	Length	Info					
		71	4.6602	192.3	168.1	.104	192	.168.3	1.101		TCP	74	3857	4 → 80	[SYN]	Seq=0	Win=29	2
		72	4.6604	192.3	168.1	.101	192	.168.	1.104	}_	TCP	66	80 →	38574	[SYN,	ACK]	Seq=0 A	
		73	4.6604	192.3	168.1	.104	192	.168.3	1.101		TCP	54	3857	4 → 80	[ACK]	Seq=1	Ack=1	V
∔•		74	4.6606	192.3	168.1	.104	192	.168.3	1.101		HTTP	393	GET .	/ HTTF	9/1.1			
		77	4.7006	192.1	168.1	.101	192	.168.3	1.104		TCP	60	80 →	38574	[ACK]	Seq=1	Ack=34	¢
-		96	7.3729	192.3	168.1	.101	192	.168.3	1.104		HTTP	975	HTTP	/1.1 2	200 OK	(text	/html)	
4																	Þ	
►	Fra	ame 1	74: 393	bytes	on w	ire (3	3144	bits)	, 393	3 by	tes d	apture	d (31	44 bi	ts) on	inter	face 0	
►	Eth	nerne	et II, S	rc:Lav	/er 2 =	_d1:8e	e:0c	(00:0	c:29:	d1:	8e:00	;), Dst	: Gig	a-Byt	6a:9a:	a2 (fo	::aa:14:	6a:
►	Int	erne	et Proto	col Ve	ersio	n 4, s	Src:	192.1	68.1.	104	, Dst	: 192.	168.1	.101	Layer 3			
►	Tra	insm:	ission C	ontro	l Pro	tocol,	Src	Port	: 385	574,	Dst	Port:	80, S	eq: 1	Ack:	1, Ler	n: S <mark>Layer</mark>	4
•	Нур	erte	ext Tran	sfer I	Proto	col		Layer	7									

Ethernet Header (Data Link)

Data link layer holds 6 bytes of **Mac address** of sender's system and receiver's system with 2 bytes of **Ether type** is used to indicate which protocol is encapsulated i.e. IPv4/IPv6 or ARP.

In Wireshark Ethernet II layer represent the information transmitted over the data link layer. From given below image you can observe that highlighted lower part of Wireshark is showing information in Hexadecimal format where the first row holds information of Ethernet headers details.

So here you can get the source and destination Mac address which also available in Ethernet Header.

The row is divided into three columns as described below:

Ethernet header 14 bytes	Destination MAC Address 6 Bytes	Source MAC Address 6 Bytes	Ether Type 2 Bytes		
Bits Color	Gray	Light Green	Pink		
Hexadecimal value	Fc:aa:14:6a:9a:a2	00:0c:29:d1:8e:0c	0800		

As we know the MAC address of the system is always represented in Hexadecimal format but both types are generally categorized in the ways given below:

Ether Type	Hexadecimal Value
ARP: Address Resolution Protocol	0x0806
IPv4: Internet Protocol version 4	0x0800
IPv6: Internet Protocol version 6	0x86dd
IEEE 802.1Q	0x8100

Once again if you notice the given below image then you can observe the highlighted text in pink colour is showing hex value **08 00** which indicates that here **IPv4** is used.

ī	Fran	ne 1	.7:	74	byt	es	on	wir	e (592	bit	s),	74	by	tes	Ca	iptu	ire	d (5	592	bit	s)	or	in	ter	face	e 0)				
	Ethe	erne	t I	laye	er 2	V	'mwa	re_	d1:	8e:0	с (00:	0c:	29:	d1:	8e:	0c)	, I	Dst:	: Gi	.ga	Byt	t_6	a:9	a:a	2 (f	c:	aa:	14	:6a	:9a	:a2
	Inte	erne	t P	rot	oco	1 V	ers	ion	4,	Src	: 1	92.	168	.1.	104	, 0	st:	19	92.1	168.	1.1	L01										
	Transmission Control Protocol, Src Port: 60914, Dst Port: 80, Seq: 0, Len: 0																															
											Laye	er 2 I	Hexa	a de	cima	ıl da	Ita															•
	0000	fc	aa	14	6a	9a	a2	00	0c	29	d1	8e	0c	08	00	45	00			j)		Ε.								
	0010	00	3c	bb	c5	40	00	40	06	fa	d8	сO	a8	01	68	<u>c0</u>	a8		.<.	.@.(<u>@</u> .			n								
	0020	01	65	ed	f2	00	50	8c	44	36	e7	00	00	00	00	a0	02	n	.е.	P	. D	6										
	0030	72	10	84	4c	00	00	02	04	05	b4	04	02	08	0a	94	74		r	L				t								
	0040	fa	1a	00	00	00	00	01	03	03	07																					

IP Header (Network Layer)

IP header in Wireshark has described the network layer information which is also known as the backbone of the OSI model as it holds Internet Protocol version 4's complete details. Network layer divides data frame into packets and defines its routing path through some hardware devices such as routers, bridges, and switches. These packets are identified through their logical address i.e. source or destination network IP address.

In the image of Wireshark, I have highlighted six most important values which contain vital information of a data packet and this information always flows in the same way as they are encapsulated in the same pattern for each IP header.

Now here, **45** represent IP header length where "4" indicates **IP version 4** and "5" is header length of **5 bits**. while **40** is time to live (**TTL**) of packet and **06** is hex value for **TCP** protocol which means these values changes if anything changes i.e. TTL, Ipv4 and Protocol.

Therefore, you can take help of given below table for examining TTL value for the different operating system.

Operating System	Hex Value TTL	Decimal value TTL
Windows	80	128
Linux	40	64
MAC	39	57

Similarly, you can take help of given below table for examining other Protocol value.

Protocol	Hex Value	Decimal Value
ICMP	1	1
ТСР	w ⁶ .hackinga	r ⁶ cles in
EGP	8	8
UDP	11	17

From given below image you can observe Hexadecimal information of the IP header field and using a given table you can study this value to obtain their original value.

IP header	Header	Total Length	TTL	Protocol	Source IP	Destination IP
(20 bytes)	length					
Bits Color	Red	Orange	Yellow	Dark Green	Dark Brown	Black
Hex Value	5	Зc	40	06	C0.a8.01.68	C0.a8.01.65
Decimal value	5	60	64	6	192.168.1.104	192.168.1.105

The IP header length is always given in form of the bit and here it is 5 bytes which are also minimum IP header length and to make it 20 bytes, multiply 4 with 5 i.e., 20 bytes.

Frame 17: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
 Ethernet II, Src: Vmware_d1:8e:0c (00:0c:29:d1:8e:0c), Dst: Giga-Byt_6a:9a:a2 (fc:a)
 Internet Protocol Version 4, Src: 192.168.1.104, Dst: 192.168.1.101 layer 3

Transmission Control Protocol, Src Port: 60914, Dst Port: 80, Seq: 0, Len: 0

																_	
•																laye	r 3 Hexa decimal data
0000	fo	~ ~	14	60	0.0	- 2	00	00	20	d1	0.0	00	00	00	45	00	i) E
0000	TC	aa	14	0a	9d	a۷	00	00	29	uт	oe	00	00	00	45	00	···J····)···· ⊑·
0010	00	3c	bb	c5	40	00	40	06	fa	d8	c0	a8	01	68	сO	a8	.<@.@h
0020	01	65	ed	f2	00	50	8c	44	36	e7	00	00	00	00	a0	02	.e <mark>P.D 6</mark>
0030	72	10	84	4c	00	00	02	04	05	b4	04	02	08	0a	94	74	rLt
0040	fa	1a	00	00	00	00	01	03	03	07							
				000													
				U.													

TCP Header (Transport Layer)

Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) and Internet Control Message Protocol (ICMP) are the major protocols as it gives host-to-host connectivity at the Transport Layer of the OSI model. It is also known as Heart of OSI model as it plays a major role in transmitting errors free data.

By examining Network Layer information through Wireshark, we found that here TCP is used for establishing a connection with destination network.

We knew that a computer communicates with another device like a modem, printer, or network server; it needs to handshake with it to establish a connection.

TCP follows Three-Way-Handshakes as describe below:

- A client sends a TCP packet to the server with the SYN flag
- A server responds to the client request with the SYN and ACK flags set.
- Client completes the connection by sending a packet with the ACK flag set

Structure of TCP segment

Transmission Control Protocol accepts data from a data stream, splits it into chunks, and adds a TCP header creating a TCP segment. A TCP segment only carries the sequence number of the first byte in the segment.

A TCP segment consists of a segment header and a data section. The TCP header contains mandatory fields and an optional extension field.

Source Port	The 16-bit source port number, Identifies the sending port.
Destination Port	The 16-bit destination port number. Identifies the receiving port
Sequence Number	The sequence number of the first data byte in this segment. If the
	SYN control bit is set, the sequence number is the initial sequence
	number (n) and the first data byte is n+1.
Acknowledgment	If the ACK control bit is set, this field contains the value of the next
Number	sequence number that the receiver is expecting to receive.
Data Offset	The number of 32-bit words in the TCP header. It indicates where the
	data begins.
Reserved	Six bits reserved for future use; must be zero.

Flags	CWR, ECE, URG, ACK, PSH, RST, SYN, FIN
Window	Used in ACK segments. It specifies the number of data bytes,
	beginning with the one indicated in the acknowledgment number
	field that the receiver (the sender of this segment) is willing to
	accept.
Checksum	The 16-bit one's complement of the one's complement sum of all 16-
	bit words in a pseudo-header, the TCP header, and the TCP data.
	While computing the checksum, the checksum field itself is
	considered zero.
Urgent Pointer	Points to the first data octet following the urgent data. Only
	significant when the URG control bit is set.
Options	Just as in the case of IP datagram options, options can be either:
	– A single byte containing the option number
	– A variable length option in the following format
Padding	The TCP header padding is used to ensure that the TCP header ends
	and data begins on a 32-bit boundary. The padding is composed of
	zeros.

Different Types of TCP flags

TCP flags are used within TCP header as these are control bits that specify particular connection states or information about how a packet should be set. TCP flag field in a TCP segment will help us to understand the function and purpose of any packet in the connection.

List of flags	Description	Decimal Value	Hex Value
CWR	Congestion Window Reduced (CWR) flag is set by	128	80
	the sending host to shows that it received a TCP segment with the ECE flag set		
ECE	ECN-Echo indicate that the TCP peer is ECN capable during 3-way handshake	64	40
URG	Indicates that the urgent pointer field is significant in this segment.	32	20
АСК	Indicates that the acknowledgment field is significant in this segment.	16	10
PSH	Push function to transfer data	08	08
RST	Resets the connection.	04	04
SYN	Synchronizes the sequence numbers.	02	02
FIN	Last packet from sender which means there is no more data.	01	01
NS	Nonce Sum flag used for concealment protection.	00	00

From given below image you can observe Hexadecimal information of TCP header field and using the given table you can study this value to obtain their original value.

Sequence and acknowledgment numbers are is a major part of TCP, and they act as a way to guarantee that all data is transmitted consistently since all data transferred through a TCP connection must be acknowledged by the receiver in a suitable way. When an acknowledgment is not received, then the sender will again send all data that is unacknowledged.

TCP Header	Bits Color	Hex Value	Decimal value
Source Port	Pink	ed f2	60914
Destination Port	Lemon Yellow	00 50	80
(HTTP)			
Sequence Number	Dark Brown	8c 44 36 e7	2353280743
Acknowledgment	Grey	00 00 00 00	0
Number			
Flag (SYN)	Dark Yellow	02	02
Window size	Green	72 10	29,200
Checksum	Orange	84 4c	33,868
Urgent Pointer	Light Brown	00 00	00
Options	Red	*	*

))	Fran Ethe Inte	ne 1 erne erne	7: t I t P	74 I, rot	byt Src	es : V l V	on mwa ers	wir re_ ion	e (5 d1:8	592 3e:0 Src	bit)c (:: 1	s), 00: 92.	74 0c: 168	by 29:	tes d1: 104	ca 8e: , D	ptu 0c) st: t P	red , D 19	(592 st: (2.168	2 bi Giga B.1	its) a-By .101	on t_6a	inte :9a:	erfa :a2
	ויימו	131111	.331	.011	U			ha	ick	in	ja		die	S	'n	03	L F		. 00,	, 30	зЧ÷	layer		0
4																								
	0000	fc	aa	14	6a	9a	a2	00	0c	29	d1	8e	0c	08	00	45	00		j.).	E		
	0010	00	Зc	bb	c5	40	00	40	06	fa	d8	c0	a8	01	68	c0	a8		.<@	.@.		h.		
	0020	01	65	ed	f2	00	50	8c	44	36	e7	90	00	00	00	a0	02		.e	P.D	6.			
	0030	72	10	84	4c	00	00	02	04	05	b4	04	02	08	0a	94	74	1	гL.				t	
	0040	fa	1a	00	00	00	00	01	03	03	07													
					N	M	W	ha	rck	ine	a	laye	er 4	Hexa	deo	ima	l da	ta						

Using given below table you can read Hex value of other Port Number and their Protocol services. Although these services operate after getting acknowledgment from the destination network and explore at application layer OSI model.

In this way, you can examine every layer of Wireshark for Network Packet Forensic.

Ports Number	Services	Hex Value	Decimal Value
21	FTP	15	21
22	SSH	16	22
23	Telnet	17	23
25	SMTP	19	25
53	DNS	35	53
80	HTTP	50	80

Password Sniffing

Because clear text protocols do not encrypt communication, all data, including passwords, is visible to the naked eye. Anyone who is in a position to see the communication (for example, a man in the middle) can eventually see everything.

In the sections that follow, we'll take a closer look at these protocols and see examples of captured passwords using Wireshark.

Capture HTTP Password

No introduction is certainly needed for the Hypertext Transfer Protocol (HTTP). It usually works on port 80/TCP, and as it is a text protocol, it does not give the communication parties much or no privacy. Anyone who's able to communicate can catch everything, including passwords, via that channel.

While all major browser vendors have made considerable efforts to prevent the use of HTTP as far as possible, during penetration testing, HTTP can be used on internal media.

Here is an example of login credentials captured in a POST request in an HTTP communication:

📕 http password.pcapn	g		- 🗆 X
File Edit View Go	Capture Analyze Stat	stics Telephony Wireless Tools Help	
	🎗 🔂 🍳 👄 🔿 🕾	T 4 📃 🗏 Q Q Q 🎹	
http			+
Source	Destination	Protocol Length Info	
192.168.0.100	224.0.0.251	MDNS 152 Standard guery 0x0041 PTR %9E5E7C8F4798952	26C9BCD95D24084F6F0B27C
192.168.0.104	224.0.0.251	MDNS 437 Standard guery response 0x0000 PTR Y-Series	-896e8e3a4462f9a459a9f
192.168.0.104	224.0.0.251	MDNS 404 Standard guery response 0x0000 PTR Y-Series	-896e8e3a4462f9a459a9f
192.168.0.104	224.0.0.251	MDNS 389 Standard query response 0x0000 PTR Y-Series	-896e8e3a4462f9a459a9f
192.168.0.1	224.0.0.1	IGMPv3 60 Membership Query, general	
192.168.0.107	18.192.172.30	TCP 74 52356 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=	1460 SACK_PERM=1 TSval
18.192.172.30	192.168.0.107	TCP 74 80 → 52356 [SYN, ACK] Seq=0 Ack=1 Win=28960) Len=0 MSS=1440 SACK_F
192.168.0.107	18.192.172.30	TCP 66 52356 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=	0 TSval=2163476701 TSe
192.168.0.107	18.192.172.30	HTTP 605 POST /userinfo.php HTTP/1.1 (application/x	(-www-form-urlencoded)
18.192.172.30	192.168.0.107	TCP 66 80 → 52356 [ACK] Seq=1 ACK=540 Win=30208 Le	n=0 ISVal=//33//626 IS
18.192.172.30	192.168.0.107	HTTP 342 HTTP/1.1 302 Found (text/html)	
192.168.0.107	18.192.172.30	TCP 66 52356 → 80 [ACK] Seq=540 Ack=277 Win=64128	Len=0 TSval=2163476852
192 168 0 107	18 192 172 30	HTTP 460 GET /login nhn HTTP/1 1	
<pre>Wireshark · Packet 4 Accept-Lang Accept-Enco Content-Typ Content-Len Origin: htt Connection: Referer: ht Upgrade-Ins \r\n [Full reque [HTTP reque [HTTP reque File Data:</pre>	<pre>002 http password.pcapng uage: en-US,en;q=0.5\ ding: gzip, deflate\r e: application/x-www- gth: 36\r\n p://testphp.vulnweb.c keep-alive\r\n tp://testphp.vulnweb. ecure-Requests: 1\r\n st URI: http://testph st 1/2] n frame: 404] st in frame: 406] 36 bytes Encoded: application/ "uname" = "vijaymehta me ijaymehta "pass" = "maxelladivi s axelladiviner</pre>	<pre> × 1232 .251 .251</pre>	Len=0 TSval=773377641 for any sources Len=0 TSval=216347700 :64128 Len=0 TSval=2162 Len=0 TSval=773377812 :31232 Len=0 TSval=7733 Len=0 TSval=216347868 .250 for any sources for any sources >
<		>	

Monitoring HTTPS packets over SSL or TLS

Dissect HTTPS Packet Captures

Open the provided HTTPS/TLS.pcapng file. Where you can see

- The 3-way handshake is happening
- Hello from SSL Client and the ACK from server
- Server Hello and then ACK
- Exchanging some key and Cipher information
- Started Exchanging Data

File	Edit View Go Ca	pture Analyze Statis	stics Telephony Wireless	Tools Help		
	🧟 🖲 📙 🗟 🗙	🔓 🔍 🗢 🗢 🗟	Ŧ 🕹 📃 🗏 🔍 Q	€, ₩		
Арр	ly a display filter <ctrl-< td=""><td>/></td><td></td><td></td><td>•</td><td>• +</td></ctrl-<>	/>			•	• +
No.	Time	Source	Destination	Protocol Length Info		^
_	1 0.000000	127.0.0.1	127.0.0.1	TCP 74 38713 → 4	443 [SYN] Seq=0 Win=32767 Len=0 MSS=16396 SAG	5
	2 0.000021	127.0.0.1	127.0.0.1	TCP 74 443 → 387	713 [SYN, ACK] Seq=0 Ack=1 Win=32767 Len=0 MS	5
	3 0.000037	127.0.0.1	127.0.0.1	TCP 66 38713 → 4	443 [ACK] Seg=1 Ack=1 Win=32767 Len=0 TSval=5	5
	4 0.000158	127.0.0.1	127.0.0.1	SSLv2 171 Client He	2110 -	
	5 0.000178	127.0.0.1	127.0.0.1	TCP 66 443 → 387	713 [ACK] Seq=1 Ack=106 Win=32767 Len=0 TSva]	1
	6 0.002160	127.0.0.1	127.0.0.1	SSLv3 995 Server He	ello, Certificate, Server Hello Done 🔫 🗕	
	7 0.002609	127.0.0.1	127.0.0.1	TCP 66 38713 → 4	443 [ACK] Seq=106 Ack=930 Win=32767 Len=0 TSv	/
	8 2.808933	127.0.0.1	127.0.0.1	SSLv3 278 Client Ke	ey Exchange, Change Cipher Spec, Encrypted Ha	а
	9 2.822770	127.0.0.1	127.0.0.1	SSLv3 141 Change Ci	ipher Spec, Encrypted Handshake Message	
	10 2.822809	127.0.0.1	127.0.0.1	TCP 66 38713 → 4	143 [ACK] Seq=318 Ack=1005 Win=32767 Len=0 TS	5
	11 2.833071	127.0.0.1	127.0.0.1	SSLv3 503 Applicati	ion Data 🔫——	
	12 2.873275	127.0.0.1	127.0.0.1	TCP 66 443 → 387	713 [ACK] Seq=1005 Ack=755 Win=32767 Len=0 TS	5
	13 2.938485	127.0.0.1	127.0.0.1	SSLv3 103 Encrypted	d Handshake Message	
	14 2.938750	127.0.0.1	127.0.0.1	SSLv3 183 Encrypted	d Handshake Message	
	15 2.938761	127.0.0.1	127.0.0.1	TCP 66 443 → 387	713 [ACK] Seq=1042 Ack=872 Win=32767 Len=0 TS	5
	16 2.938999	127.0.0.1	127.0.0.1	SSLv3 1073 Encrypted	d Handshake Message, Encrypted Handshake Mess	5.
	17 2.940026	127.0.0.1	127.0.0.1	SSLv3 337 Encrypted	d Handshake Message, Change Cipher Spec, Encr	-
	18 2.943406	127.0.0.1	127.0.0.1	SSLv3 172 Change Ci	upher Spec, Encrypted Handshake Message	
	19 2.944825	127.0.0.1	127.0.0.1	SSLv3 5756 Applicati	ion Data, Application Data	_
<					>	
 > Fra > Eth > Int > Tra 	ame 11: 503 bytes mernet II, Src: 00 ternet Protocol Ve	on wire (4024 bits :00:00_00:00:00 (0 rsion 4, Src: 127.), 503 bytes captured 0:00:00:00:00:00), Dst 0.0.1, Dst: 127.0.0.1 1: 38713 Dst Port: 40	(4024 bits) : 00:00:00_00:00:00 (00	0:00:00:00:00)	
	ansmort Laver Secu	rity	c. 56/15, DSC POLC. 4-	+5, 364. 310, ACK. 1003,	, Len. 457	
-	insport cuyer seeu					
0000	00 00 00 00 00 00	0 00 00 00 00 00 0	00 08 00 45 00	· · · · · · · · · E ·		^
0010	01 e9 49 72 40 0	0 40 06 f1 9a 7f (30 00 01 7f 00Ir@) - @		
0020	00 01 97 39 01 bl	b 78 8c 3a d4 78	c5 28 c5 80 18 ···9·	·x· :·x·(···		
0030	7f ff ff dd 00 00	0 01 01 08 0a 1f 9	53 7c 14 1f 53 ·····	····s ··s		
0040	hc Ac 9a 84 d7 b	1 DV 4a C3 3e 9d . 9 99 9c - 21 19 f e -	///8/01/20/04 ····	···J ·>·WX·,·		
0050	77 fb 72 42 4f a	d 50 4a d0 aa 6f :	a 44 6c 62 94 wir80	Pl ··o·Dlb·		
0070	1b c5 fe e9 1c 5	e de 85 0b 0e 05 0	e4 18 6e d2 d3 ·····	^n		
0.7	a					~
02	rsasnakeoil2.cap				Packets: 58 · Displayed: 58 (100.0%) Profile: De	efault 🔡

Then, if we click on any application data, that data is unreadable to us. However, with Wireshark, we can decrypt that data... all we need is the server's Private Key. *Don't worry we have already provided the key along with the PCAP file.*

Fi	le l	Edit View Go Cap	ture Analyze Statist	tics Telephony Wireless	Tools	Help	
		🧕 🖲 📙 🛅 🗙	📓 🔍 🗢 🔿 🗟	፻ ୬ 📃 🗏 € ୧	Q 🎹		
	App	ly a display filter <ctrl- <="" td=""><td>></td><td></td><td></td><td></td><td></td></ctrl->	>				
No		Time	Source	Destination	Protocol	Length	Info
		1 0.000000	127.0.0.1	127.0.0.1	TCP	74	38713 → 443 [SYN] Seq=0 Win=32767 Len=
		2 0.000021	127.0.0.1	127.0.0.1	тср	74	443 \rightarrow 38713 [SYN, ACK] Seq=0 Ack=1 Win
		3 0.000037	127.0.0.1	127.0.0.1	тср	66	38713 → 443 [ACK] Seq=1 Ack=1 Win=3276
		4 0.000158	127.0.0.1	127.0.0.1	SSLv2	171	Client Hello
		5 0.000178	127.0.0.1	127.0.0.1	TCP	66	443 → 38713 [ACK] Seq=1 Ack=106 Win=32
		6 0.002160	127.0.0.1	127.0.0.1	SSLv3	995	Server Hello, Certificate, Server Hello
		7 0.002609	127.0.0.1	127.0.0.1	TCP	66	38713 → 443 [ACK] Seq=106 Ack=930 Win=
		8 2.808933	127.0.0.1	127.0.0.1	SSLv3	278	Client Key Exchange, Change Cipher Spe
		9 2.822770	127.0.0.1	127.0.0.1	SSLv3	141	Change Cipher Spec, Encrypted Handshak
		10 2.822809	127.0.0.1	127.0.0.1	TCP	66	38713 → 443 [ACK] Seq=318 Ack=1005 Win
		11 2.833071	127.0.0.1	127.0.0.1	SSLv3	503	Application Data
		12 2.873275	127.0.0.1	127.0.0.1	тср	66	443 → 38713 [ACK] Seq=1005 Ack=755 Win
		13 2.938485	127.0.0.1	127.0.0.1	SSLv3	103	Encrypted Handshake Message
		14 2.938750	127.0.0.1	127.0.0.1	SSLv3	183	Encrypted Handshake Message
		15 2.938761	127.0.0.1	127.0.0.1	TCP	66	443 → 38713 [ACK] Seq=1042 Ack=872 Win
		16 2.938999	127.0.0.1	127.0.0.1	SSLv3	1073	Encrypted Handshake Message, Encrypted
		17 2.940026	127.0.0.1	127.0.0.1	SSLv3	337	Encrypted Handshake Message, Change Ci
		18 2.943406	127.0.0.1	127.0.0.1	SSLv3	172	Change Cipher Spec, Encrypted Handshak
		19 2.944825	127.0.0.1	127.0.0.1	SSLv3	5756	Application Data, Application Data
<							
>	Fra	ame 11: 503 bytes o	n wire (4024 bits)	, 503 bytes captured	(4024 b	its)	
>	Eth	hernet II, Src: 00:	00:00_00:00:00 (00	:00:00:00:00:00), Dst	: 00:00	:00_00	:00:00 (00:00:00:00:00)
>	Int	ternet Protocol Ver	sion 4, Src: 127.0	.0.1, Dst: 127.0.0.1			
>	Tra	ansmission Control	Protocol, Src Port	: 38713, Dst Port: 44	3, Seq:	318, /	Ack: 1005, Len: 437
~	Tra	ansport Layer Secur	ity				
	~	SSLv3 Record Layer	: Application Data	Protocol: Applicatio	n Data		
		Content Type: A	pplication Data (2	3)			
		Version: SSL 3.	0 (0x0300)				
		Length: 432					
		Encrypted Appli	cation Data: 4ac33	e9d7778012cb4bc4c9a84	d7b9900a	:2110f0	fa007c16bb77fb72424fad504ad0aa6f…
	I						

To Decrypt the Encrypted Application Data over TLS or SSL Navigate to

Edit > Preference > Protocol > TLS

Thread A Thrift Tibia TIME TIPC TIPC TIVOConnect TLS TNS Token-Ring TPCP TPKT TPM2.0 TPNCP TPKT TPM2.0 TPNCP TSDNS TSP TTE TURNCHANI' TURNCHANI' TURNCHANI' TURNCHANI' TURNCHANI' TUXEDO TZSP UA3G UASIP UAUDP UBDP	Transport Layer Security RSA keys list Edit TLS debug file Browse Reassemble TLS records spanning multiple TCP segments Reassemble TLS Application Data spanning multiple TLS records Message Authentication Code (MAC), ignore "mac failed" Pre-Shared Key (Pre)-Master-Secret log filename Browse
>	

And add these values

IP address: 127.0.0.1

Port: 443

Key File:

4		TLS Decry	pt					×
	IP	address	Port	Protocol	Key File	Passv	vord	
		127.0.0.1	443	http	C:/Users/vijvi/OneDrive/Desktop/HTTPSTLS/HTTPSTLS.ke	y		
							J	
					1			
	+	– P	^	\[<u>C: Users (vijvi)</u>	AppData	a (Roaming Wi	resharkijssi keys
					OK Copy from	m	Cancel	Help

Hurray!!! As you can see, we have Successfully decrypted the Data over the TLS.

F	ile Edit	View	Go Cap	oture	Analyz	e Sta	atistics	Teleph	iony V	Wireles	s Tools	Help									
			010	C	۹ 듣	⇒ ≦	2			Ð, Q	Q 🎹										
	Apply a	display filter	<ctrl- <="" td=""><td>></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>-</td><td>•</td><td>+</td></ctrl->	>															-	•	+
No).	Time		Sourc	e		De	stination			Protoc	ol Leng	th Info								
	26	2.964819	Э	127.	0.0.1		12	7.0.0.1	1		TCP		56 443 → 38	714 [ACK]	Seg=1	Ack=12	21 Win:	=32767 L	.en=0	1	
	27	2.992274	4 1//1	127.	0.0.1		12	7.0.0.1	es.		SSLv3	2	20 Server H	ello, Cha	nge Ci	oher Sp	bec, Fi	inished			
	28	2 992313	2	127	0 0 1		12	7 0 0	1		тср		<u>6 38714</u> →	443 [ACK]	Seg=1	21 Ack	155 W	in=32767	7 Len=		
4	29	2.992855	5	127.	0.0.1		12	7.0.0.1	1		HTTP	5	52 GET /ico	ns/debiar	/openlo	ogo-25.	jpg H	TTP/1.1			
4	30	2.99350	1	127.	0.0.1		12	7.0.0.	1		HTTP	5	96 HTTP/1.1	404 Not	Found	(text/	/html)				
~	31	2.993840	а	127.	0.0.1		12	7.0.0.1	1		HTTP	4	71 GET /ico	ns/apache	pb.png	g HTTP/	1.1				
	32	2.994179	Э	127.	0.0.1		12	7.0.0.	1		HTTP	18	28 HTTP/1.1	200 OK	(PNG)	-					
Т	33	3.004256	5	127.	0.0.1		12	7.0.0.	1		TCP		56 443 → 38	713 [ACK]	Seq=78	845 Ack	=1548	Win=327	767 Le	2	
	34	3.033250	3	127.	0.0.1		12	7.0.0.	1		TCP		56 38714 →	443 [ACK]	Seq=1	022 Ack	=2447	Win=327	767 Le	-	
Т	35	3.501643	3	127.	0.0.1		12	7.0.0.	1		HTTP	5	88 HTTP/1.1	404 Not	Found	(text/	(html)				
ł	36	3.507003	1	127.	0.0.1		12	7.0.0.	1		HTTP	4	39 GET /fav	icon.ico	HTTP/1	.1					
ł	37	3.507543	1	127.	0.0.1		12	7.0.0.	1		HTTP	5	30 HTTP/1.1	404 Not	Found	(text/	/html)				
Т	38	3.507555	5	127.	0.0.1		12	7.0.0.	1		TCP	(6 38714 →	443 [ACK]	Seq=13	395 Ack	=2961	Win=327	767 Le		
	39	3.541174	4	127.	0.0.1		12	7.0.0.	1		TCP		6 38713 →	443 [ACK]	Seq=1	548 Ack	=8367	Win=327	767 Le	2	
Т	40	6.037880	9	127.	0.0.1		12	7.0.0.	1		HTTP	5	l1 GET /tes	t HTTP/1.	1						
Т	41	6.037932	2	127.	0.0.1		12	7.0.0.	1		TCP		56 443 → 38	713 [ACK]	Seq=83	367 Ack	(=1993	Win=327	767 Le	2	
	42	6.041185	5	127.	0.0.1		12	7.0.0.	1		HTTP	6	4 HTTP/1.1	301 Move	d Perma	anently	/ (tex	kt/html))		
	43	6.041367	7	127.	0.0.1		12	7.0.0.	1		TCP		56 38713 →	443 [ACK]	Seq=19	993 Ack	(=8945	Win=327	767 Le		
	44	6.088943	3	127.	0.0.1		12	7.0.0.	1		HTTP	5	l1 GET /tes	t/ HTTP/1	.1						-
<																			>		
~	Trans	port Lave	er Secur	itv																	
	✓ ssi	.v3 Recor	d Layer	·: Apr	olicati	ion Da	ata Pr	otocol	: http	-over	-tls										
		Content	Type: A	pplic	ation	Data	(23)														
		Version:	SSL 3.	0 (0)	(0300)																
		Length:	344																		
		Encrypte	d Appli	catio	on Data	a: 1aa	ab72a9	9faeed	998838	fdc3f	8262708	2e5b3a	a7b4b71158	317031287	eba13cb	dc3adb	bd				
		[Applica	tion Da	ta Pr	rotocol	l: htt	tp-ove	r-tls]													
	TLS	5 segment	data (317 ł	bytes)																
	Y SSI	.v3 Recor	'd Layer	•: App	plicati	ion Da	ata Pr	rotocol	: http	-over	-tls										
		Content	Type: A	pplic	ation	Data	(23)														
		Version:	SSL 3.	0 (0)	(0300)																
		Length:	1408																		
		Encounto	d Annli	+÷	n Data	1. 1.f.	144334	-Efren	000£12	0000-	6704eee	60-A6-	00005£30€4	768-4-000	01-4506	10000	0.2				_
0	000 48	54 54 5	0 2f 31	2e 3	31 20	32 30	30 2	0 4f 4	b 0d	HTTP,	/1.1 2	00 OK									
0	010 02	44 61 7	4 65 3a	20 2	+a 6†	be 20	20 3	2 34 2	0 41	·Date	e: Mon	, 24 A									
F	rame (18	28 bytes)	Decrypt	ed TLS	(317 byt	tes)	Decryp	oted TLS ((1385 byt	tes)	Reassem	bled SSL	(1702 bytes)								

Page | 17

Capture Telnet Password

No introduction is required for Telnet protocol using port tcp/23. It is mainly used for administrative convenience and is known for its insecurity. Since encryption is not available, privacy or unauthorized access protection is not available. Telnet is still used today, however...

Telnet is a protocol used for administration on a wide range of devices. Telnet is the only option for some devices, with no other options (e.g. there is no SSH nor HTTPS web interface available). This makes it extremely difficult for organizations to completely eliminate it. Telnet is commonly seen on:

- Video Conferencing Systems
- Mainframes
- Network equipment
- Storage and Tape systems
- Imaging devices
- Legacy IP based Phones

Since telnet is a plain-text protocol, an opponent can wake up to the communication and capture it all, including passwords. The following screenshot shows an example of a telnet communication with the captured password:

	mergedtest.pcapng						_		
File	Edit View Go	Capture Analyze	Statistics Telepho	ony Wireles	s Tools Help				
	🔳 🔬 🛞 📙 📠	🔀 🖾 🍳 👄 🖻	> 😤 🗿 👲 📃	📃 Q Q	€, ∰				
t	elnet						X	→	•
о.	Time	Source	Destination	Protocol L	ength Info				
	34 2.575598	192.168.0.2	192.168.0.1	TELNET	69 Telnet Data				
	35 2.576676	192.168.0.1	192.168.0.2	TCP	66 23 → 1550 [ACK]	Seq=143	Ack=207	Win=1	1
	36 2.577672	192.168.0.1	192.168.0.2	TELNET	75 Telnet Data				
	37 2.589229	192.168.0.2	192.168.0.1	TCP	66 1550 → 23 [ACK]	Seq=207	Ack=152	Win=3	
	38 3.581505	192.168.0.2	192.168.0.1	TELNET	72 Telnet Data 🔫				
	39 3.582813	192.168.0.1	192.168.0.2	тср —	►66 23 → 1550 [ACK]	Seq=152	Ack=213	Win=1	
	40 3.847152	192.168.0.1	192.168.0.2	TELNET	68 Telnet Data			J	
	41 3.859250	192.168.0.2	192.168.0.1	TCP	66 1550 → 23 [ACK]	Seq=213	Ack=154	Win=3	
	42 3.860413	192.168.0.1	192.168.0.2	TELNET	69 Telnet Data				L
	9600,9600#.bar color OpenBSD/i386 (oof)	"	#&\$. 'DISPLAY.ba	.&&\$ m.zing.org	#			1=3 1=3	
	login: fake Password:use	er							
	<mark></mark> Last login: Warning: no Kerber OpenBSD 2.6-beta	Sat Nov 27 20:1 ros tickets issu (OOF) #4: Tue Oc	1:43 on ttyp2 f wed. t 12 20:42:32 C	rom bam.zi DT 1999	ng.org				
	Welcome to OpenBSI	D: The proactive	ely secure Unix-	like opera	ting system.				
	Please use the ser Before reporting a version of the con enough information	ndbug(1) utility a bug, please tr de. With bug re n to reproduce f	v to report bugs by to reproduce ports, please t the problem is e	in the sy it with th ry to ensu nclosed, a	stem. e latest re that nd if a				
	known fix for it e	exists, include	that as well.						

So, that now you can see an attacker completely overtake the Mainframe System.



File Transfer Protocol (FTP) usually uses the TCP/20 or the TCP/21 ports. Although this protocol is very old, it is still used in their networks by some organizations. FTP is a plain text protocol so a well-positioned attacker can capture FTP login credentials with Wireshark very easily. This screenshot shows a captured FTP password with Wireshark as an example:

_											_
	🧲 ftp.pca	р									
	ile Edit	View Go	o Capture Anal	yze Statistics	Telephony	Wireless To	ools Help				
4		0	ै 🔀 🖸 🍳 🤄	> ⇔ 🕾 ү	₺ 📃 🔳	.					
	tcp.strea	m eq 0									
N	o.	Time	Source	Destinatio	n Pro	tocol Length	Info				
	- 9	0.788801	10.0.1.100	198.145	.20.140 TC	P 78	3 55919 → 21	[SYN]	Seq=0 Win=6	5535 Len=0 MS	5S=14
	10	0.901830	198.145.20	.140 10.0.1.	100 тс	P 74	21 → 55919	[SYN,	ACK] Seq=0	Ack=1 Win=144	480 L
Π	11	0.902022	10.0.1.100	198.145	.20.140 TC	P 66	5 55919 → 21	[ACK]	Seq=1 Ack=1	Win=132480 l	_en=0
	12	1.020035	198.145.20	.140 10.0.1.	100 FT	P 93	Response: 2	20 We	lcome to ker	nel.org	
	13	1.020201	10.0.1.100	198.145	.20.140 TC	P <u>66</u>	55919 → 21	[ACK]	Seq=1 Ack=2	8 Win=132448	Len=
	14	12.651747	10.0.1.100	198.145	.20.140 FT	P 76	6 Request: US	SER ftp	р		
	15	12.784279	198.145.20	.140 10.0.1.	100 TC	P 66	5 21 → 55919	[ACK]	Seq=28 Ack=	11 Win=14592	Len=
	16	12.784290	198.145.20	.140 10.0.1.	100 FT	P 100	Response: 3	331 Ple	ease specify	the password	d. –
	17	12.784481	10.0.1.100	198.145	.20.140 TC	P 66	5 55919 → 21	[ACK]	Seq=11 Ack=	62 Win=132410	5 Len
	18	19.293792	10.0.1.100	198.145	.20.140 FT	P 88	8 Request: PA	∖SS jo∖	wens@yccc.ed	u	
	19	19.400899	198.145.20	.140 10.0.1.	100 FT	P 89	Response: 2	230 Log	gin success†	ul.	-
	20	19.401116	10.0.1.100	198.145	.20.140 TC	P 66	5 55919 → 21	[ACK]	Seq=33 Ack=	85 Win=132392	2 Len
	21	19.401370	10.0.1.100	198.145	.20.140 FT	P 72	Request: S	/ST			
	22	19.523265	198.145.20	.140 10.0.1.	100 FT	P 85	Response: 2	215 UNI	IX Type: L8		
	23	19.523444	10.0.1.100	198.145	.20.140 TC	P 66	5 55919 → 21	[ACK]	Seq=39 Ack=	104 Win=13237	76 Le
	24	19.523671	10.0.1.100	198.145	.20.140 FT	P 72	Request: FE	AT			
	25	19 615248	198 145 20	140 10 0 1	100 FT	P 81	Resnonse: 0	011-Fea	atures		
Ľ											
	Frame	18: 88 by	tes on wire (70	04 bits), 88	bytes captu	red (704 b	oits)				
	Ethern	et II, Sr	c: Apple_23:15:	a4 (f0:b4:79):23:15:a4),	Dst: Cisc	:o-Li_fe:bd:f	F9 (0 0	:23:69:fe:bd	:f9)	
	Intern	et Protoc	ol Version 4, S	irc: 10.0.1.1	100, Dst: 19	8.145.20.1	40				
	Transm	ission Co	ntrol Protocol,	Src Port: 5	5919, Dst P	ort: 21, S	eq: 11, Ack	62,	Len: 22		
Ŀ	/ File T	ransfer P	rotocol (FTP)								
	Y PAS	5 jowens@	/ccc.edu\r\n								
	1	Request co	ommand: PASS								
	1	Request ar	g: jowens@yccc	.edu							
	[Curre	nt workin	g directory:]								

As you can see by sitting in a network, we can easily capture FTP credentials.

Capture SMTP password

For many decades, we have also been accompanied by SMTP (Simple Mail Transfer Protocol). It uses TCP/25 and although the port TCP/464 is secure, today the port TCP/25 is almost opened on each mail server because of reverse compatibility.

Many TCP/25 servers need the command 'STARTTLS' to begin the encryption of SSL/TLS before any attempts are made to authenticate it. However, mail servers still support plain text authentication across the unencrypted channel within certain organizations. Mostly because of heritage systems in your internal networks.

If someone is using plain text authentication during an SMTP transaction, the credentials can be sniffed from a well-positioned attacker. The attacker must only decode the username and password from base64. SMTP uses Base 64 encoding for the transaction to encode the username and password.

A captured SMTP credentials can be seen in the following screenshot with Wireshark and the consequent base64 decoder using the base64 utility.

Mo.	smtp	💌 📑 🔜 🗙	🙆 । ९ 🗢 🔿 🖄	🗿 U 🧮 📃 Q Q (0 777		
No.	smtp				₹.₩		
No.							×
		Time	Source	Destination	Protocol	Length	Info
	1584	310967609.832	74.53.140.153	10.10.1.4	SMTP	235	S: 220-xc90.websitewelcome.com ESMTP Exim
	1584	310967609.837	10.10.1.4	74.53.140.153	SMTP	63	C: EHLO GP
	1584	310967610.178	74.53.140.153	10.10.1.4	SMTP	191	S: 250-xc90.websitewelcome.com Hello GP [1
	1584	310967610.181	10.10.1.4	74.53.140.153	SMTP	66	C: AUTH LOGIN
4	1584	310967610.523	74.53.140.153	10.10.1.4	SMTP	72	S: 334 VXNlcm5hbWU6
	1584	310967610.524	10.10.1.4	74.53.140.153	SMTP	84	C: User: Z3VycGFydGFwQHBhdHJpb3RzLmlu
	1584	310967610.866	74.53.140.153	10.10.1.4	SMTP	72	S: 334 UGFzc3dvcmQ6
	1584	310967610.866	10.10.1.4	74.53.140.153	SMTP	72	C: Pass: cHVuamFiQDEyMw==
	1584	310967611.226	74.53.140.153	10.10.1.4	SMTP	84	S: 235 Authentication succeeded
	1584	310967611.227	10.10.1.4	74.53.140.153	SMTP	90	C: MAIL FROM: <gurpartap@patriots.in></gurpartap@patriots.in>
	1584	310967611.569	74.53.140.153	10.10.1.4	SMTP	62	S: 250 OK
	1584	310967611.570	10.10.1.4	74.53.140.153	SMTP	93	C: RCPT TO: <raj_deol2002in@yahoo.co.in></raj_deol2002in@yahoo.co.in>
	1584	310967611.932	74.53.140.153	10.10.1.4	SMTP	68	S: 250 Accepted
	1584	310967611.933	10.10.1.4	74.53.140.153	SMTP	60	C: DATA
	1584	310967612.274	74.53.140.153	10.10.1.4	SMTP	110	S: 354 Enter message, ending with "." on a
	1584	310967612.305	10.10.1.4	74.53.140.153	SMTP	1514	C: DATA fragment, 1460 bytes
	1584	310967612.305	10.10.1.4	74.53.140.153	SMTP	1514	C: DATA fragment, 1460 bytes
	1584	310967612.305	10.10.1.4	74.53.140.153	SMTP	1514	C: DATA fragment, 1460 bytes
	1584	310967612.305	10.10.1.4	74.53.140.153	SMTP	1514	[TCP Window Full] C: DATA fragment, 1460 b
	1584	310967612.307	192.168.1.1	10.10.1.4	ICMP	590	Destination unreachable (Fragmentation nee
<							
2	Frame	158406: 84 byte	es on wire (672 bit	s), 84 bytes captured	(672 bi	its)	
2	Ethern	et II, Src: Cra	dlepo_3c:17:c2 (00	:e0:1c:3c:17:c2), Dst	: Netgea	ar_d9:8	31:60 (00:1f:33:d9:81:60)
2	Intern	et Protocol Ver	rsion 4, Src: 10.10	0.1.4, Dst: 74.53.140.1	153		
2	Transm	ission Control	Protocol, Src Port	:: 1470, Dst Port: 25,	Seq: 22	2, Ack:	: 337, Len: 30
× 1	Simple	Mail Transfer	Protocol				
	Use	rname: Z3VycGFy	rdGFwQHBhdHJpb3RzLm	lu			

There are many methods available to decode the base64 strings. For this, I'm using an online tool that is designed specifically for decoding such as <u>base64decode.org</u> or <u>base64decode.net</u>. But we should beware – we may not want to disclose private credentials on the Internet to other parties. In the course of penetration tests and offensive tests, sensitivity and privacy are especially crucial. This is particularly important.

Now, just copy the value of strings of user and password and decode it via base64 decoder as shown below image. As of now, I'm decrypting the user string

BASE64	🕞 Decode	*										
Decode and Encode	D Encode											
Do you have to deal with B and to encode or deco	Then this site is perfect for you! Use our super handy online tool to encode or decode your data.											
Decode from Base64 Simply enter your data then push	format the decode button.											
Z3VycGFydGFwQHBhdHJpb	3RzLmlu											
Eor encoded binaries (like im	ages, documents, etc.) use the file upload form a little further down on this page											
UTF-8 V Source	e character set.											
Decode each line separately	(useful for when you have multiple entries).											
D Live mode OFF Deco	les in real-time as you type or paste (supports only the UTF-8 character set).											
< DECODE > Deco	les your data into the area below.											
gurpartap@patriots.in												
		Ŧ										

User: - Z3VycGFydGFwQHBhdHJpb3RzLmlu

As you can see in the above screenshot, we have successfully able to see the user's name in clear text format. Similarly, we can decrypt the password

Password: - cHVuamFiQDEyMw==

BASE64	🔁 Decode										
Decode and Encode	C Encode										
Do you have to deal with Base64 format? Then this site is perfect for you! Use our super handy online tool to encode or decode your data.											
Decode from Base64 Simply enter your data then push t cHVuamFiQDEyMw==	format he decode button.										
For encoded binaries (like ima	ges, documents, etc.) use the file upload form a little further down on this page.										
UTF-8 V Source	character set.										
Decode each line separately (useful for when you have multiple entries).										
D Live mode OFF Decode	es in real-time as you type or paste (supports only the UTF-8 character set).										
< DECODE > Decode	es your data into the area below.										
punjab@123											

Hurray!!! Now we have got enough credentials to take over a system.

Analyzing SNMP Community String

Simple Network Management Protocol (SNMP) typically runs on port UDP/161. The main objective is network devices and their functions to manage and monitor. SNMP have 3 versions and the first 2 (v1 and v2c) versions are plain text. SNMP uses something that is equivalent to authentication, named community string. Therefore, it is almost the same to capture the SNMP community string as to capture credentials.

While SNMPv3 has been with us for nearly two decades, it takes time. In their internal networks, most organizations still use v1 or v2c. Typically this is due to the backwards compatibility in their networks with legacy systems.

An example of the SNMP community string captured using Wireshark is:

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help Image: Color Streem eq 1 No. Time Source Destination Protocol Length Info 15262 620.143771 192.168.0.2 192.168.0.1 SNMP 98 iso.3.6.1.4.1.4.1.2.21 15264 620.23655 192.168.0.2 192.168.0.1 SNMP 92 trap iso.3.6.1.4.1.4.1.2.21 15265 620.23656 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15266 620.336364 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15266 620.433642 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15266 620.433645 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15269 620.423625 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.
No. Time Source Destination Protocol Length Info 15262 620.143771 192.168.0.2 192.168.0.1 SNMP 99 iso.3.6.1.4.1.4.1.2.21[Mal 15263 620.183662 192.168.0.2 192.168.0.1 SNMP 98 iso.3.6.1.4.1.4.1.2.21[Mal 15264 620.223650 192.168.0.2 192.168.0.1 SNMP 92 trap iso.3.6.1.4.1.4.1.2.21 15265 620.23653 192.168.0.2 192.168.0.1 SNMP 92 trap iso.3.6.1.4.1.4.1.2.21 15266 620.23654 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15267 620.33646 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15269 620.433642 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15270 620.433657 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15270 620.633618 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15272 620.623614 </td
udp.stream eq 1 Protocol Length Info 15262 620.143771 192.168.0.2 192.168.0.1 SNMP 99 iso.3.6.1.4.1.4.1.2.21[Ma] 15262 620.143771 192.168.0.2 192.168.0.1 SNMP 98 iso.3.6.1.4.1.4.1.2.21[Ma] 15263 620.138362 192.168.0.2 192.168.0.1 SNMP 92 trap iso.3.6.1.4.1.4.1.2.21 15265 620.23653 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15265 620.33646 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15266 620.33630 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15269 620.43625 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15270 620.43627 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15271 620.53618 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21
No. Time Source Destination Protocol Length Info 15262 620.143771 192.168.0.2 192.168.0.1 SNMP 99 iso.3.6.1.4.1.4.1.2.21[Ma] 15263 620.183662 192.168.0.2 192.168.0.1 SNMP 98 iso.3.6.1.4.1.4.1.2.21[Ma] 15264 620.223650 192.168.0.2 192.168.0.1 SNMP 92 trap iso.3.6.1.4.1.4.1.2.21 15265 620.303646 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15266 620.303646 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15267 620.343630 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15269 620.423627 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15270 620.4363631 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15271 620.538613 192.168.0.2 192.168.0.1 SNM
No. Time Source Destination Protocol Length Info 15262 620.143771 192.168.0.2 192.168.0.1 SNMP 99 iso.3.6.1.4.1.4.1.2.21[Ma] 15263 620.133662 192.168.0.2 192.168.0.1 SNMP 98 iso.3.6.1.4.1.4.1.2.21[Ma] 15264 620.23653 192.168.0.2 192.168.0.1 SNMP 92 trap iso.3.6.1.4.1.4.1.2.21 15265 620.23653 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15265 620.33646 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15265 620.43625 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15270 620.43627 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15271 620.503618 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15272 620.503614 192.168.0.2 192.168.0.1 SNMP
<pre>15262 620.143771 192.168.0.2 192.168.0.1 SMMP 99 iso.3.6.1.4.1.4.1.2.21[Ma] 15263 620.183662 192.168.0.2 192.168.0.1 SNMP 98 iso.3.6.1.4.1.4.1.2.21[Ma] 15264 620.223650 192.168.0.2 192.168.0.1 SNMP 92 trap iso.3.6.1.4.1.4.1.2.21 15265 620.263653 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15266 620.303646 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15267 620.343642 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15268 620.383630 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15269 620.423625 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15269 620.423625 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15270 620.463627 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15271 620.503618 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15272 620.543611 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15273 620.583613 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15273 620.583613 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15273 620.663608 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15276 620.703604 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15276 620.703604 192.168.0.2 192.168.0.1 SNMP 225 trap iso.3.6.1.4.1.4.1.2.21 15276 620.703604 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15276 620.703604 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15276 620.703604 192.168.0.2 192.168.0.1 SNMP 63 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 63 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0</pre>
<pre>15263 620.183662 192.168.0.2 192.168.0.1 SNMP 98 iso.3.6.1.4.1.4.1.2.21[Mal 15264 620.223650 192.168.0.2 192.168.0.1 SNMP 92 trap iso.3.6.1.4.1.4.1.2.21 15265 620.263653 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15265 620.303646 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15267 620.343642 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15269 620.423625 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15269 620.423625 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15270 620.463627 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15270 620.463627 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15271 620.503618 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15272 620.543611 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15273 620.583613 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15274 620.623614 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15275 620.663608 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15276 620.703604 192.168.0.2 192.168.0.1 SNMP 154 trap iso.3.6.1.4.1.4.1.2.21 15276 620.703604 192.168.0.2 192.168.0.1 SNMP 225 trap iso.3.6.1.4.1.4.1.2.21 15276 620.78393 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15278 620.78393 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15278 620.78393 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.78393 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.78393 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.16</pre>
<pre>15264 620.223650 192.168.0.2 192.168.0.1 SNMP 92 trap iso.3.6.1.4.1.4.1.2.21 15265 620.263653 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15266 620.303646 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15267 620.343642 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15268 620.383630 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15269 620.423625 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15270 620.463627 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15271 620.503618 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15272 620.543611 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15273 620.633613 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15273 620.633613 192.168.0.2 192.168.0.1 SNMP 102 trap iso.3.6.1.4.1.4.1.2.21 15275 620.663608 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15275 620.663608 192.168.0.2 192.168.0.1 SNMP 225 trap iso.3.6.1.4.1.4.1.2.21 15277 620.73596 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 19</pre>
<pre>15265 620.263653 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15266 620.303646 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15267 620.343642 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15268 620.423625 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15270 620.463627 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15271 620.503618 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15272 620.543611 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15273 620.583613 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15274 620.623614 192.168.0.2 192.168.0.1 SNMP 102 trap iso.3.6.1.4.1.4.1.2.21 15275 620.663608 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15275 620.663608 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15275 620.663608 192.168.0.2 192.168.0.1 SNMP 154 trap iso.3.6.1.4.1.4.1.2.21 15277 620.743596 192.168.0.2 192.168.0.1 SNMP 225 trap iso.3.6.1.4.1.4.1.2.21 15278 620.7639604 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 62</pre>
<pre>15266 620.303646 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15267 620.343642 192.168.0.2 192.168.0.1 SNMP 100 iso.3.6.1.4.1.4.1.2.21 15268 620.383630 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15269 620.423625 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15270 620.463627 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15271 620.503618 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15272 620.543611 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15273 620.583613 192.168.0.2 192.168.0.1 SNMP 102 trap iso.3.6.1.4.1.4.1.2.21 15274 620.623614 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15275 620.663608 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15276 620.703604 192.168.0.2 192.168.0.1 SNMP 255 trap iso.3.6.1.4.1.4.1.2.21 15278 620.783933 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.8236</pre>
<pre>15267 620.343642 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 1.3 15268 620.383630 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15269 620.423625 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15270 620.463627 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15271 620.503618 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15272 620.543611 192.168.0.2 192.168.0.1 SNMP 102 trap iso.3.6.1.4.1.4.1.2.21 15274 620.623614 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15275 620.663608 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15275 620.663608 192.168.0.2 192.168.0.1 SNMP 154 trap iso.3.6.1.4.1.4.1.2.21 15276 620.703604 192.168.0.2 192.168.0.1 SNMP 225 trap iso.3.6.1.4.1.4.1.2.21 15277 620.743596 192.168.0.2 192.168.0.1 SNMP 259 trap iso.3.6.1.4.1.4.1.2.21 15278 620.783933 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15277 620.743596 192.168.0.2 192.168.0.1 SNMP 259 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.8</pre>
<pre>15268 620.383630 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15269 620.423625 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15270 620.463627 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15271 620.503618 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15272 620.543611 192.168.0.2 192.168.0.1 SNMP 102 trap iso.3.6.1.4.1.4.1.2.21 15273 620.583613 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15274 620.623614 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15275 620.663608 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15276 620.703604 192.168.0.2 192.168.0.1 SNMP 154 trap iso.3.6.1.4.1.4.1.2.21 15277 620.743596 192.168.0.2 192.168.0.1 SNMP 225 trap iso.3.6.1.4.1.4.1.2.21 15278 620.783933 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15277 620.743596 192.168.0.2 192.168.0.1 SNMP 259 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602</pre>
<pre>15269 620.423625 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15270 620.463627 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15271 620.503618 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15272 620.543611 192.168.0.2 192.168.0.1 SNMP 102 trap iso.3.6.1.4.1.4.1.2.21 15273 620.583613 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15274 620.623614 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15275 620.663608 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15276 620.703604 192.168.0.2 192.168.0.1 SNMP 225 trap iso.3.6.1.4.1.4.1.2.21 15277 620.743596 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15278 620.783933 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 536 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602</pre>
<pre>15270 620.463627 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15271 620.503618 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15272 620.543611 192.168.0.2 192.168.0.1 SNMP 102 trap iso.3.6.1.4.1.4.1.2.21 15273 620.583613 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15274 620.623614 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15275 620.663608 192.168.0.2 192.168.0.1 SNMP 154 trap iso.3.6.1.4.1.4.1.2.21 15276 620.703604 192.168.0.2 192.168.0.1 SNMP 225 trap iso.3.6.1.4.1.4.1.2.21 15277 620.743596 192.168.0.2 192.168.0.1 SNMP 259 trap iso.3.6.1.4.1.4.1.2.21 15278 620.783933 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 15278 conrespondence Protocol </pre>
<pre>15271 620.503618 192.168.0.2 192.168.0.1 SNMP 100 trap iso.3.6.1.4.1.4.1.2.21 15272 620.543611 192.168.0.2 192.168.0.1 SNMP 102 trap iso.3.6.1.4.1.4.1.2.21 15273 620.583613 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15274 620.623614 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15275 620.663608 192.168.0.2 192.168.0.1 SNMP 154 trap iso.3.6.1.4.1.4.1.2.21 15276 620.703604 192.168.0.2 192.168.0.1 SNMP 225 trap iso.3.6.1.4.1.4.1.2.21 15277 620.743596 192.168.0.2 192.168.0.1 SNMP 259 trap iso.3.6.1.4.1.4.1.2.21 15278 620.783933 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 </pre>
<pre>15272 620.543611 192.168.0.2 192.168.0.1 SNMP 102 trap iso.3.6.1.4.1.4.1.2.21 15273 620.583613 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15274 620.623614 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15275 620.663608 192.168.0.2 192.168.0.1 SNMP 154 trap iso.3.6.1.4.1.4.1.2.21 15276 620.703604 192.168.0.2 192.168.0.1 SNMP 225 trap iso.3.6.1.4.1.4.1.2.21 15277 620.743596 192.168.0.2 192.168.0.1 SNMP 259 trap iso.3.6.1.4.1.4.1.2.21 15278 620.783933 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 (\$ Simple Network Management Protocol version: version-1 (0) community: public * data: trap (4) * trap enterprise: 1.3.6.1.4.1.4.1.2.21 (iso.3.6.1.4.1.4.1.2.21)</pre>
<pre>15273 620.583613 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15274 620.623614 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15275 620.663608 192.168.0.2 192.168.0.1 SNMP 154 trap iso.3.6.1.4.1.4.1.2.21 15276 620.703604 192.168.0.2 192.168.0.1 SNMP 225 trap iso.3.6.1.4.1.4.1.2.21 15277 620.743596 192.168.0.2 192.168.0.1 SNMP 259 trap iso.3.6.1.4.1.4.1.2.21 15278 620.783933 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 </pre>
<pre>15274 620.623614 192.168.0.2 192.168.0.1 SNMP 86 trap iso.3.6.1.4.1.4.1.2.21 15275 620.663608 192.168.0.2 192.168.0.1 SNMP 154 trap iso.3.6.1.4.1.4.1.2.21 15276 620.703604 192.168.0.2 192.168.0.1 SNMP 225 trap iso.3.6.1.4.1.4.1.2.21 15277 620.743596 192.168.0.2 192.168.0.1 SNMP 259 trap iso.3.6.1.4.1.4.1.2.21 15278 620.783933 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 </pre>
<pre>15275 620.663608 192.168.0.2 192.168.0.1 SNMP 154 trap iso.3.6.1.4.1.4.1.2.21 15276 620.703604 192.168.0.2 192.168.0.1 SNMP 225 trap iso.3.6.1.4.1.4.1.2.21 15277 620.743596 192.168.0.2 192.168.0.1 SNMP 259 trap iso.3.6.1.4.1.4.1.2.21 15278 620.783933 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 </pre>
<pre>15276 620.703604 192.168.0.2 192.168.0.1 SNMP 225 trap iso.3.6.1.4.1.4.1.2.21 15277 620.743596 192.168.0.2 192.168.0.1 SNMP 259 trap iso.3.6.1.4.1.4.1.2.21 15278 620.783933 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 </pre> Simple Network Management Protocol version: version-1 (0) community: public v trap enterprise: 1.3.6.1.4.1.4.1.2.21 (iso.3.6.1.4.1.4.1.2.21)
<pre>15277 620.743596 192.168.0.2 192.168.0.1 SNMP 259 trap iso.3.6.1.4.1.4.1.2.21 15278 620.783933 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 </pre> Simple Network Management Protocol version: version-1 (0) community: public v trap enterprise: 1.3.6.1.4.1.4.1.2.21 (iso.3.6.1.4.1.4.1.2.21)
<pre>15278 620.783933 192.168.0.2 192.168.0.1 SNMP 432 trap iso.3.6.1.4.1.4.1.2.21 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 </pre> Simple Network Management Protocol version: version-1 (0) community: public v trap enterprise: 1.3.6.1.4.1.4.1.2.21 (iso.3.6.1.4.1.4.1.2.21)
<pre> 15279 620.823602 192.168.0.2 192.168.0.1 SNMP 636 trap iso.3.6.1.4.1.4.1.2.21 Simple Network Management Protocol version: version-1 (0) community: public v data: trap (4) v trap enterprise: 1.3.6.1.4.1.4.1.2.21 (iso.3.6.1.4.1.4.1.2.21) </pre>
<pre> < Simple Network Management Protocol version: version-1 (0) community: public v data: trap (4) v trap enterprise: 1.3.6.1.4.1.4.1.2.21 (iso.3.6.1.4.1.4.1.2.21) </pre>
<pre> Simple Network Management Protocol version: version-1 (0) community: public data: trap (4) v trap enterprise: 1.3.6.1.4.1.4.1.2.21 (iso.3.6.1.4.1.4.1.2.21)</pre>
<pre>version: version-1 (0) community: public v data: trap (4) v trap enterprise: 1.3.6.1.4.1.4.1.2.21 (iso.3.6.1.4.1.4.1.2.21)</pre>
<pre>community: public data: trap (4) frap enterprise: 1.3.6.1.4.1.4.1.2.21 (iso.3.6.1.4.1.4.1.2.21)</pre>
<pre> data: trap (4) frap enterprise: 1.3.6.1.4.1.4.1.2.21 (iso.3.6.1.4.1.4.1.2.21) </pre>
<pre>v trap enterprise: 1.3.6.1.4.1.4.1.2.21 (iso.3.6.1.4.1.4.1.2.21)</pre>
enterprise: 1.3.6.1.4.1.4.1.2.21 (iso.3.6.1.4.1.4.1.2.21)
agent-addr: 127.0.0.1
generic-trap: egpNeighborLoss (5)
specific-trap: 0
time-stamp: 15270
✓ variable-bindings: 1 item
> 1 3 6 1 2 1 2 1 0 .
0000 00 e0 29 68 8b fb 00 20 af 1b 07 fa 08 00 45 00 ···)h··· ·····E·
0010 00 56 4T 6C 00 00 40 11 a9 d7 c0 a8 00 02 c0 a8 ·VOI··@······
0020 00 01 04 10 00 d2 00 42 9a 1T 30 36 02 01 00 04
Image: State of the s

An attacker could now use the community string and collect detailed system information. This could enable the attacker to learn about the system insensitive detail and to make further attempts. Note that the community string sometimes also allows you to modify your remote system configuration (read/write access).

Capture MSSQL Password

The Microsoft SQL server usually runs on TCP/1433 port; this is yet another service we can use with Wireshark to capture the password. If the server is not configured using the ForceEncryption option, it is possible to record plain text authentication directly or via a downgrade attack. MSSQL credentials can be easily captured by a man in the middle.

Here's an example of a Wireshark-captured MSSQL

Fil	e Edit	View	Go	Capture	Analyze	Statistics	Teleph	nony	Wirele	ess To	ols Help			
		0	010	🗙 😂	ء 🗢) 😤 👔	& ⊒		⊕ ∈					
	Apply a d	lisplay filt	ter <(Ctrl-/>										
No		Time		Source		Destinatio	'n	Pro	tocol	Length	Info			
_	1	0.0000	000	192.16	8.0.254	192.168	.0.254	TC	0	74	56162 → 3306	[SYN]	Sea=0	Win=3279
	2	0.0000	946	192.16	8.0.254	192.168	.0.254	TC		74	3306 → 56162	[SYN,	ACK]	Seq=0 Ack
T	3	0.0000	977	192.16	8.0.254	192.168	.0.254	тс		66	56162 → 3306	[ACK]	Seq=1	Ack=1 Wi
	4	0.0002	265	192.16	8.0.254	192.168	.0.254	Mys	SQL	122	Server Greet	ing pro	oto=10	version=
	5	0.0002	286	192.16	8.0.254	192.168	.0.254	TC	0	66	56162 → 3306	[ACK]	Seq=1	Ack=57 W
	6	0.0005	59	192.16	8.0.254	192.168	.0.254	My:	SQL	132	Login Reques	t user	-tfoer	ste
	7	0.0005	583	192.16	8.0.254	192.168	.0.254	TC	0	66	3306 → 56162	[ACK]	Seq=5	7 Ack=67
	8	0.0006	595	192.16	8.0.254	192.168	.0.254	My:	SQL	77	Response OK			
	9	0.0008	393	192.16	8.0.254	192.168	.0.254	My:	SQL	103	Request Quer	у		
	10	0.0010)51	192.16	8.0.254	192.168	.0.254	My:	SQL	162	Response			
	11	0.0407	792	192.16	8.0.254	192.168	.0.254	TC	•	66	56162 → 3306	[ACK]	Seq=1	04 Ack=16
	12	5.6988	332	192.16	8.0.254	192.168	.0.254	My:	SQL	88	Request Quer	у		
	13	5.6990)11	192.16	8.0.254	192.168	.0.254	My:	SQL	130	Response			
	14	5.6990	35	192.16	8.0.254	192.168	.0.254	TC	0	66	56162 → 3306	[ACK]	Seq=1	26 Ack=22
	15	5.6992	226	192.16	8.0.254	192.168	.0.254	My:	SQL	75	Request Use	Databa	se	
	16	5.6993	324	192.16	8.0.254	192.168	.0.254	My:	5QL	77	Response OK			
	17	5.6995	573	192.16	8.0.254	192.168	.0.254	My:	5QL	85	Request Quer	у		
	18	5.6999	998	192.16	8.0.254	192.168	.0.254	My:	SQL	174	Response			
	19	5.7001	180	192.16	8.0.254	192.168	.0.254	My:	SQL	82	Request Quer	У		
	20	5.7004	18	192.16	8.0.254	192.168	.0.254	My:	5QL	160	Response			
<														
>	Transm	ission	Contr	rol Prot	ocol, Sr	Port:	56162,	Dst Po	ort:	3306,	Seq: 1, Ack:	57, L	en: 66	
\sim	MySQL I	Protoc	ol								•			
	Pacl	ket Le	ngth:	62										
	Pac	ket Nu	mber:	1										
	✓ Logi	in Req	uest								1			
	> 0	lient	Capab	ilities	0xa685									
	> E	Extende	ed Cli	ent Capa	abilities	: 0x0003	3							
	N	IAX Pa	cket:	16777210	5									
	0	harset	t: utf	8 COLLAT	TE utf8_g	general_o	i (33)							
	ι ι	Jnused	: 0000	00000000	00000000	00000000	000000	000000	0000	00				
	ι I	Jsernar	me: tf	oerste 🚽	←									
	F	asswoi	rd: ee	fd6d5562	2851bc596	6a0b4123	36ae3f2	315efo	c4 🚽	←				

Now, we have a privileged account of the MSSQL server. Therefore, this would have a critical impact allowing the attacker to take complete control over the database server or it could also lead to remote command execution (RCE).

Capture PostgreSQL Password

PostgreSQL is yet another widely used SQL database server. It runs on TCP port 5432 and accepts a variety of authentication methods. It is usually set to disallow clear-text authentication, but it can also be set to allow it. In such cases, a well-positioned attacker could intercept network traffic and obtain the username and password.

It should be noted that PostgreSQL authentication occurs in multiple packets. The username and database name comes first:

File	Edit	View	Go	Cap	oture	Ar	nalyz	e S	Statisti	cs	Tele	pho	ny	Wire	less	То	ols He	lp			
		۲		1	C	٩	¢	⇒	21	<u>}</u>	2			€ (Q, €	2					
Pg	jsql																				
No.		Time		s	ource			[Destina	tion			Pro	otocol	Leng	jth	Info				
	7	0.002	956	1	27.0	0.0.1	L	1	127.0	.0.1	L	_	PG	SQL	1	04	>				
	9	0.003	985	1	27.0	0.0.1	L	1	127.0	.0.1	L		PG	SQL	1	04	>				
	11	0.003	253	1	27.0	0.0.1	L	1	127.0	.0.1	L		PG	SQL		79	<r< td=""><td></td><td></td><td></td><td></td></r<>				
	13	0.003	340	1	27.0	0.0.1	a	cla	127.0	.0.1	-		PG	SQL		79	<r< td=""><td></td><td></td><td></td><td></td></r<>				
	15	0.0034	458	1	27.0	0.0.1		1	127.0	.0.1	a starting		PG	SQL	1	07	>p				
	16	0.003	582	1	27.0	0.0.1	L	1	127.0	.0.1	L		PG	SQL	1	07	>p				
	19	0.077	978	1	27.0	0.0.1	L	1	127.0	.0.1	L		PG	SQL	2	27	<r s="" s<="" td=""><td>/s/s/</td><td>/S/K/</td><td>Z</td><td></td></r>	/s/s/	/S/K/	Z	
1	20	0.0779	986	1	27.0	0.0.1	L	1	127.0	.0.1	L		PG	SQL	2	27	<r s="" s<="" td=""><td>/s/s/</td><td>/S/K/</td><td>Z</td><td></td></r>	/s/s/	/S/K/	Z	
	21	0.078	998	1	27.0	0.0.1	L	1	127.0	.0.1	L		PG	SQL	2	22	>P/B/D	/E/S/	/P/B/	D/E	/s
ł	23	0.078	598	1	27.0	0.0.1	L	1	127.0	.0.1	L		PG	SQL	1	92	>P/B/E	/P/B/	/D/E/	s	
	25	0.080	906	1	27.0	0.0.1	L	1	127.0	.0.1	L		PG	SQL	1	64	<1/2/C	/1/2/	/T/D/	c/z	
	26	0.081	378	1	27.0	0.0.1	L	1	127.0	.0.1	L		PG	SQL	1	18	>P/B/D	/E/S			
-	27	0.082	503	1	27.0	0.0.1	L	1	127.0	.0.1	L		PG	SQL		99	<1/2/n	/c/z			
	28	0.083	214	1	27.0	0.0.1	L	1	127.0	.0.1	L		PG	SQL	2	51	<1/2/T	/c/z			
	29	0.084	898	1	27.0	0.0.1	L	1	127.0	.0.1	L		PG	SQL	1	45	<1/2/T	/c/z			
	32	8.9494	436	1	27.0	0.0.1	L	1	127.0	.0.1	L		PG	SQL	3	27	>P/B/D	/E/S			
	34	9.0993	196	1	27.0).0.1	L	1	127.0	.0.1			PG	SQL	3	48	<1/2/T	/c/z			
	36	15.399	9924	1	27.0	.0.1	าล	CI	127.0	.0.1	rt		PG	SQL	1	14	>B/D/E	/s			
<																					
> Iı > Tı • Pe	ransm ostgr Typ Len Pro Pro Par	et Pro ission eSQL e: Sta gth: 3 tocol tocol ameter ameter	n Cor n Cor n Cor 8 majo majo n nan n val	ol Ven ntrol o mess or ver or ver ne: us lue: o	rsio Pro age sion sion sion ser	n 4, toco n: 3 n: 0	Sr0	c: 1 Src	27.0. Port:	0.1 45	, D	st: , D	127 st P	.0.0 ort:	543	2,	Seq: 1	, Acl	k: 1,	Le	n: 3
	Par	ameter	nar	ne: da	atab	ase															
	Par	ameter	val	lue: n	nail	stor	e														
0000	00	00 00	00	00 00	00	00	00	00	00 00	08	00	45	00								
0010	00	5a bd	98	40 00	40	0 6	7f	03	7f 00	00	01	7f	00	٠Z	··@·(<u>a</u> .					
0020	00	01 b3	6b	15 38	c9	01	b0	bd	c9 49	e2	1d	80	18		·k·8	•••	· · · I · · ·				
0030) 7f	ff a0	48	00 00	01	01	08	0a	13 42	0d	2c	13	42		·H··	•••	···В·,	в			
\bigcirc	1	PostgreS	QL: Pr	rotocol																	Pack

File	Edit	View	Go	0	Capt	ure	An	alyz	e	Stat	istic	S	Tele	pho	ny	Wire	less	То	ols	Help				
		۲		110		C	٩	¢	⇒	2	Ŷ	4				e,	Q,	Ð,						
	gsql																							
No.		Time			Sou	ırce				Dest	tinati	on			Pr	otocol	Ler	igth	Info					
	7	0.002	956		12	7.0	.0.1	100		127	.0.	0.1	-		PO	iSQL		- 104	>					
	9	0.003	085		12	7.0	.0.1	щ		127	.0.	0.1	3		PG	iSQL		104	>					
li	11	0.003	253		12	7.0	.0.1			127	.0.	0.1			PG	iSQL		79	<r< td=""><td></td><td></td><td></td><td></td><td></td></r<>					
	13	0.003	340		12	7.0	.0.1			127	.0.	0.1			PG	iSQL		79	<r< td=""><td></td><td></td><td></td><td></td><td></td></r<>					
	15	0.003	458		12	7.0	.0.1			127	.0.	0.1		E	PO	iSQL		107	>p					
	16	0.003	582		12	7.0	.0.1			127	.0.	0.1			PG	iSQL		107	>p					
	19	0.077	078		12	7.0	.0.1			127	.0.	0.1			PG	iSQL		227	<r s<="" td=""><td>/s/s</td><td>/s/:</td><td>5/K/</td><td>Z</td><td></td></r>	/s/s	/s/:	5/K/	Z	
	20	0.077	986		12	7.0	.0.1			127	.0.	0.1			PG	iSQL		227	<r s<="" td=""><td>/s/s</td><td>/s/:</td><td>5/K/</td><td>Z</td><td></td></r>	/s/s	/s/:	5/K/	Z	
	21	0.078	098		12	7.0	.0.1			127	.0.	0.1			PG	iSQL		222	>P/B	/D/E	/s/I	Р/В/	D/E	/s
	23	0.078	598		12	7.0	.0.1			127	.0.	0.1			PG	iSQL		192	>P/B	/E/P	/B/I	D/E/	S	
	25	0.080	906		12	7.0	.0.1			127	.0.	0.1			PG	iSQL		164	<1/2	/C/1	/2/	T/D/	c/z	
	26	0.081	378		12	7.0	.0.1			127	.0.	0.1			PG	iSQL		118	>P/B	/D/E	/s			
	27	0.082	503		12	7.0	.0.1			127	.0.	0.1			PG	iSQL		99	<1/2	/n/0	/Z			
	28	0.083	214		12	7.0	.0.1			127	.0.	0.1			PG	iSQL		251	<1/2	/т/с	/Z			
	29	0.084	898		12	7.0	.0.1			127	.0.	0.1	1		PO	iSQL		145	<1/2	/т/с	/Z			
	32	8.949	436		12	7.0	.0.1			127	.0.	0.1	IC.		PO	iSQL		327	>P/B	/D/E	/S			
	34	9.099	196		12	7.0	.0.1			127	.0.	0.1			PO	iSQL		348	<1/2	/T/C	/Z			
	36	15.39	9924		12	7.0	.0.1			127	.0.	0.1			PO	iSQL		114	>B/D	/E/S				
<																								
> F > E	rame thern	15: 10 et II,	97 bj , Sr	ytes c: 0	s on 90:0	n wi)0:0	.re 0_0	(850 0:00	5 b 0:0	its) 0 ((),: 00:(107 30:0	by 00:0	tes 00:	cap 00:0	oture 00),	d (Dst	856 : 0	bits 0:00:) 00_0	0:0	0:00	(0	0:00
> 1	ntern	et Pro	otoc	ol V	/ens	ion	4,	Sho	:::	127	.0.0	0.1	, D	st:	127	.0.0	.1							
> T	ransm	issior	ı Co	ntro	01 F	rot	000	1. 9	Snc	Po	rt:	459	930	, D	st P	Port:	54	32.	Seq:	39,	Ac	k: 1	4,	Len
~ P	ostgr	eSQL																						
	Тур	e: Pas	SWO	rd m	less	age																		
	Len	gtn: 4			£-0		4-71	- 44 0	220		C.L.	- 0.	-01	10										
	Pas	swora:	ma	сет	тсө	тас	ae/:	5410	\$290	aeei	F6D:	e90	1914	+2										
004	9 Ød	2c 70	00	00	00	28	6d	64	35	63	65	66	66	63	30	1.1	p · · · q	(m	d5ce	ffcØ				
005	31	64 63	64	65	37	35	34	31	38	32	39	64	65	65	66	1d	cde7	754	1829	deef				
0060	36	62 35	65	39	63	39	31	34	32	00						6b	5e90	91	42.					
0	2	A passwo	ord. (j	ogsql	.pas	swor	d), 3	6 byt	tes															Pac

We can also see the PostgreSQL password in the following network packet:

Creating Firewall Rules with Wireshark

Although Wireshark cannot block network traffic, it can assist us in the development of firewall rules for our firewall. Wireshark will create firewall rules based on the traffic we're looking at. To block a packet, all we have to do is pick it and navigate through the menu:

mergedtest.pcapng								- 🗆	\times					
File Edit View Go	Cap	ture Analyze	Statistics Teleph	ony Wireless	Tools Help									
	\mathbf{x}	ء 🗢 🍳 🔁	ي 🚯 😫 🕈 😫	≣ ଇ ଇ	Q. 🎹									
Apply a display filter	<ctrl- :<="" th=""><th>></th><th></th><th></th><th></th><th></th><th></th><th></th><th>• +</th></ctrl->	>							• +					
No. Time	So	urce	Destination	Protocol Le	ength Info				^					
1 0.000000	19	2.168.0.2	192.168.0.1	ТСР	74 1550 → 23 [SYN] Seq=0 Win=32120 Len=0 M	ISS=1460 S/	ACK_PERM=1 TS						
2 0.002525	19	2.168.0.1	192.168.0.2	TCP	74 23 → 1550 [SYN	I, ACK] Seq=0 Ack=1 Win=17	376 Len=0	MSS=1448 WS=						
3 0.002572	19	02.168.0.2	192.168.0.1	TCP	66 1550 → 23 [ACK	[] Seq=1 Ack=1 Win=32120 L	en=0 TSva	l=10233636 TS						
4 0.004160	19	02.168.0.2	192.168.0.1	TELNET	93 Telnet Data	•								
5 0.150335	19	02.168.0.1	192.168.0.2	TELNET	69 Telnet Data			-1 400000554 T						
0 0.150402	19	2.168.0.2	192.168.0.1	TELNET	60 IS50 → 23 [ACK	[] Seq=28 ACK=4 W1n=32120	Len=0 ISV	a1=10233651 1						
8 0 151946	10	2.168.0.2	192.100.0.1	TCP	66 23 → 1550 [ACK	1 Seg=4 Ack=31 Win=17376	Len=0 TSv	al=2467372 TS						
9.0.153657	1	.2.100.0.1	152.100.0.2	i ci	00 25 × 1550 [Ac.	(] 5cq-4 Ack-51 WIN-17570	Len-o 150							
10 0.153865	1	📕 Wireshark ·	Firewall ACL Rules	mergedtest.pca	ipng	—								
11 0.154984	1							al=2467372 T						
12 0.155577	1	# Netfilter (ipta	bles) rules for merged	test.pcapng, pao	ket 1. Change eth0 to a va	alid interface if needed.								
13 0.155656	1	# IPv4 source a	address.											
14 0.156646	1	iptablesapper	IPv4 source address. Itablesappend INPUTin-interface eth0source 192, 168.0.2/32jump DROP											
15 0.159016	1	# IDv4 dection	ion address											
16 0.159227	1	iptablesapper	nd INPUTin-interface	eth0source 1	92.168.0.1/32jump DRO	P		val=10233652						
17 0.159844	1													
18 0.161018	1	iptablesapper	nd INPUTin-interface	eth0protocol	tcpsource-port 1550ju	Imp DROP		=0 Sva1=102	= ×					
S France 4 + 74 hotes		# Destination p	ort.					-						
> Frame 1: 74 bytes	s on	iptablesapper	nd INPUTin-interface	eth0protocol	tcpsource-port 23jum	p DROP								
> Internet Protoco	1 Ve	# IPv4 source a	address and port											
> Transmission Cont	trol	iptablesapper	nd INPUTin-interface	eth0protocol	tcpsource 192.168.0.2/	32source-port 1550jump DROP								
		# IPv4 destinat	tion address and port. nd INPUTin-interface	eth0protocol	tcpsource 192.168.0.1/	32source-port 23jump DROP								
		# MAC source a	address.											
		iptablesappel		e eu ivmac-sou	rce oolaolociopipitiajun	il uror								
		# MAC destinat iptablesapper	tion address. nd INPUTin-interface	eth0mac-sou	rce 00:00:c0:9f:a0:97ju	mp DROP								
0000 00 00 c0 9f a	a0 9								^					
0010 00 3c 46 3c 4	10 01	Create rules for	Netfilter (intables)	~		Inbound	Denv							
0020 00 01 06 0e 0	00 1 AA A	create rules for	recirce (prables)	-			E Seny		~					
mergedtest.pcap	png				Save	Close Copy	Help	6) Profile: De	fault					

Selected rules can now be copied and pasted directly into our firewall. The following firewalls' syntax is supported by Wireshark:

- Windows Firewall(netsh)
- IP Filter(ipfw)
- NetFilter (iptables)
- Packet Filter(pf)

Conclusion

Wireshark can catch authentication for a wide range of network protocols. There is a possibility as long as we have the ability to eavesdrop on network traffic and the communication is not encrypted. Passwords aren't the only thing that a well-placed attacker can capture; virtually any type of data passing through the network can be captured.

Hence, one can make use of these commands as a cybersecurity professional to assess vulnerabilities on systems and keep these systems away from threat.

References

- <u>https://www.hackingarticles.in/network-packet-forensic-using-wireshark/</u>
- <u>https://www.hackingarticles.in/wireshark-for-pentester-password-sniffing/</u>
- <u>https://www.wireshark.org/docs/</u>