

VoIPShark: Open Source VoIP Analysis Platform

Nishant Sharma

Jeswin Mathai

Ashish Bhangale

PentesterAcademy.com & AttackDefense.com

About Us

Me, Nishant Sharma

- R&D Manager and Lead Trainer, Pentester Academy
- Firmware developer, Enterprise WiFi APs and WIPS Sensors
- Masters degree in Infosec
- Published research at Blackhat US/Asia, DEF CON USA and other venues

Co-authors

- **Ashish Bhangale**, Sr. Security Researcher
- **Jeswin Mathai**, Security Researcher

PentesterAcademy.com

https://www.pentesteracademy.com

PentesterAcademy Courses and Online Labs

Follow @SecurityTube 117K followers
Recommend 291K Share

COURSES ONLINE LABS PRICING WHY SUBSCRIBE TESTIMONIALS RED TEAM LABS BLOG MEMBER ACCESS

40+ COURSES
1500+ HD VIDEOS
700+ ONLINE LABS
UNLIMITED LAB TIME
EXPERT TRAINERS
TOP CERTIFICATIONS

Training Professionals from



vmware®



McAfee



SONY

AttackDefense.com

https://attackdefense.com/latestlabs

ATTACK DEFENSE

- Dashboard
- Ongoing Labs
- Latest Additions**
- Community Labs

EARN CREDENTIALS

- Badges

THE BASICS

- Network Recon
- Real World Webapps
- Traffic Analysis
- Webapp CVEs
- Metasploit
- Offensive Python
- Network Pivoting

◀ Dashboard

Latest Additions: 925

Our team has been working hard to get these to you!

 Challenge III
Level: Easy
badge-tshark-basics, 4 days ago

 Challenge II
Level: Easy
badge-tshark-basics, 4 days ago

 Challenge I
Level: Easy
badge-tshark-basics, 4 days ago

 Metasploit CTF I
Level: Easy
metasploit-ctf, 12 days ago

 x86_64 Assembly Lab: GUI Access
Level: Easy
pa-assembly-x86-64-video-labs, 18 days ago

 x86_64 Assembly Lab: CLI Access
Level: Easy
pa-assembly-x86-64-video-labs, 19 days ago

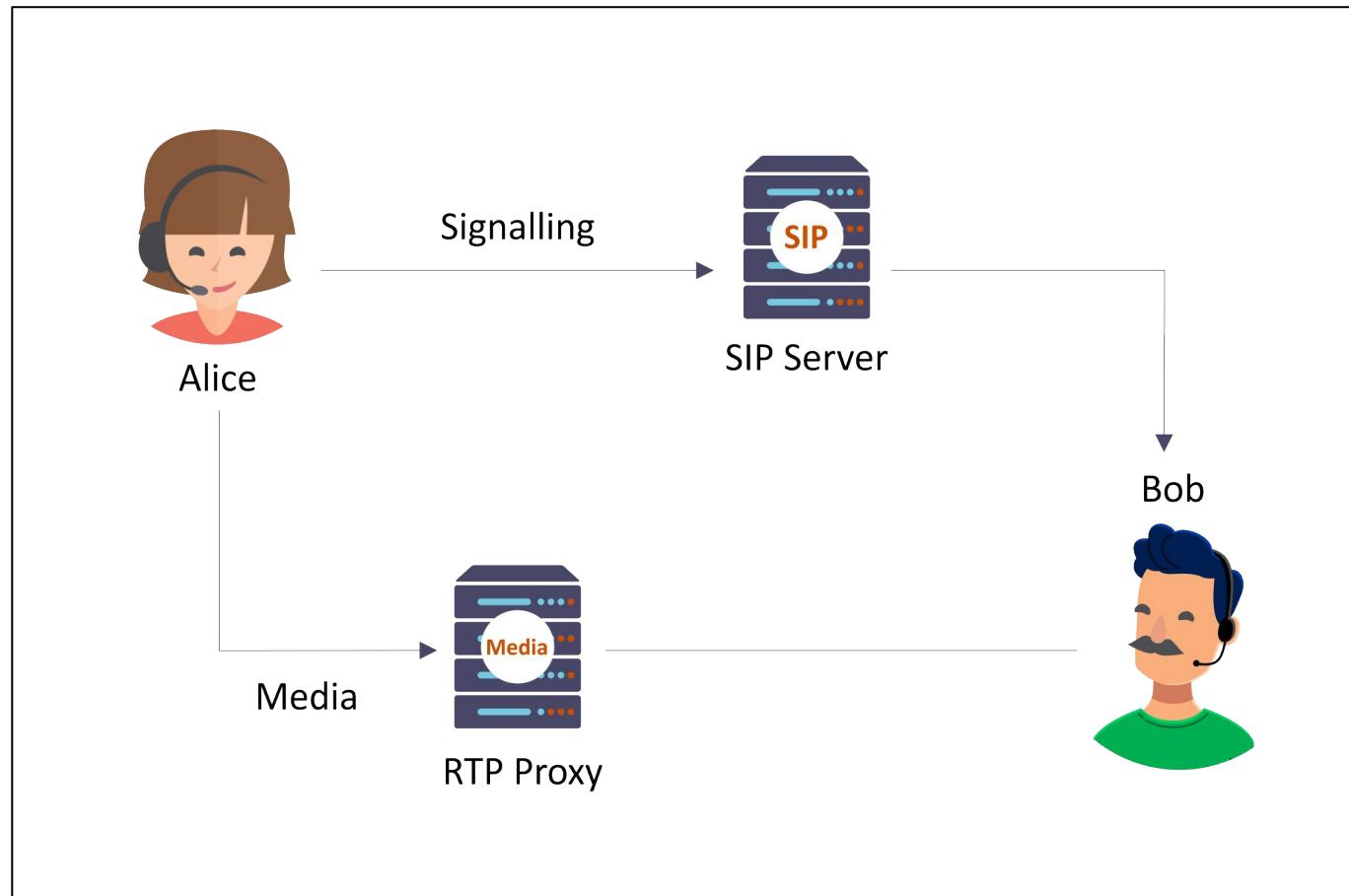
Start Start Start Start

Talk Overview

- VoIP Basics
 - SIP, RTP
 - Secure: TLS, SRTP
- Recovering/Decrypting VoIP Calls
- Current open source tools and issues
- VoIPShark
 - Architecture and Internals
 - Analyzing VoIP Traffic
 - Recovering Calls
 - Detecting Attacks Passively
 - Demo

VoIP Telephony

- Signalling + Media



Signalling Protocols

SIP (Session Initiation Protocol)

- Developed by the IETF
- Replacement for the desk phones and PSTN (Public Switched Telephone Network)

H.323

- Created by the ITU-T
- Focused on videoconferencing but also used for voice calls

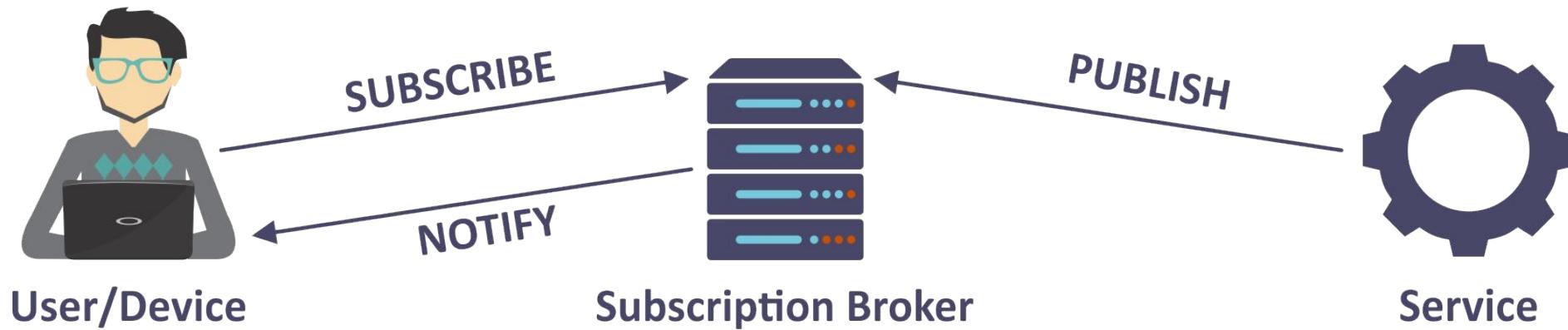
SCCP (Skinny)

- Cisco proprietary protocol used for line-side control of phones

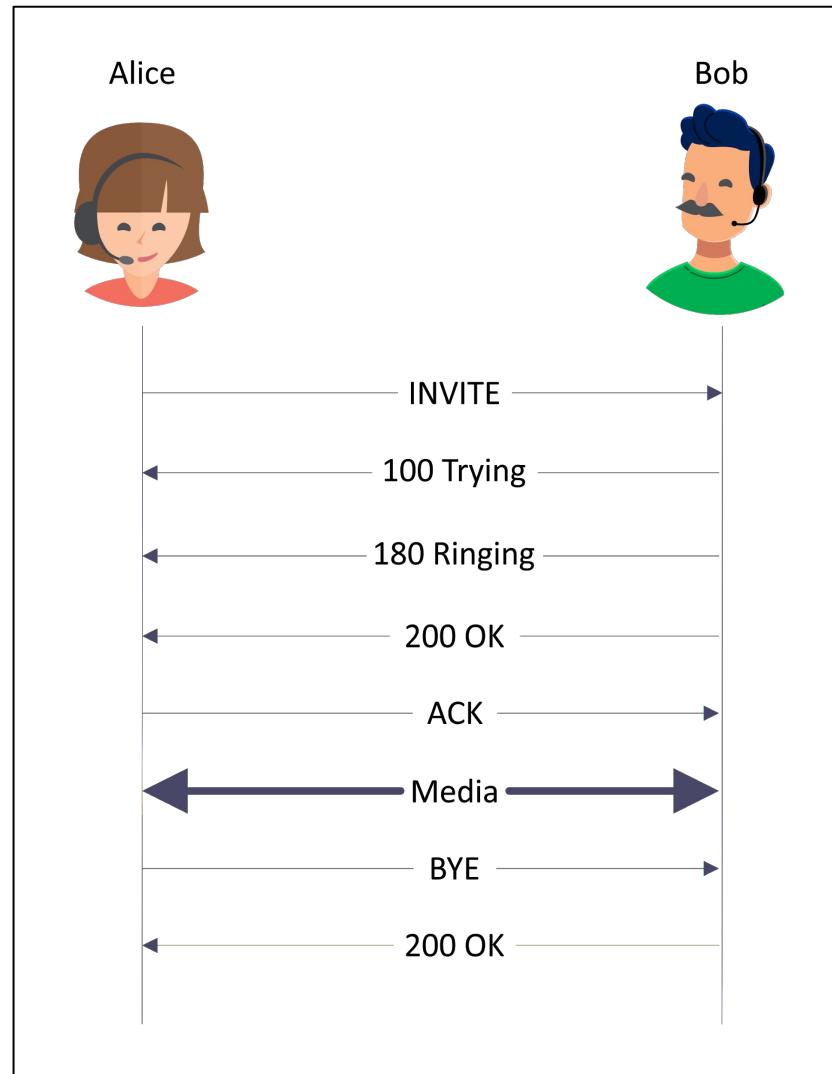
Session Initiation Protocol

- Text-based protocol
- Applications
 - Calls (audio, video) using other media streams like RTP
 - Text messages using SIP “Message” method
- Works with other protocols
- Session Description Protocol (SDP) to define with media negotiation and setup
- Can operate over TCP, UDP or SCTP (Stream Control Transmission Protocol)
- Security is provided by TLS (Transport Layer Security) i.e. SIP over TLS.

SUBSCRIBE, PUBLISH and NOTIFY



Session Initiation Protocol: Sample Call Flow



User Agent Server (UAS) Solutions



Softphone clients

- Program for making telephone calls over IP
- Some options
 - Zoiper
 - X Lite www.counterpath.com/x-lite-download
 - LinPhone
 - MicroSIP



www.microsip.org



www.zoiper.com

Factors in choosing a good softphone client

- Check codec support
- Check encryption capabilities (Especially in free versions)
- Other functionalities (i.e. Text message option, hold, waiting etc.)



www.linphone.org



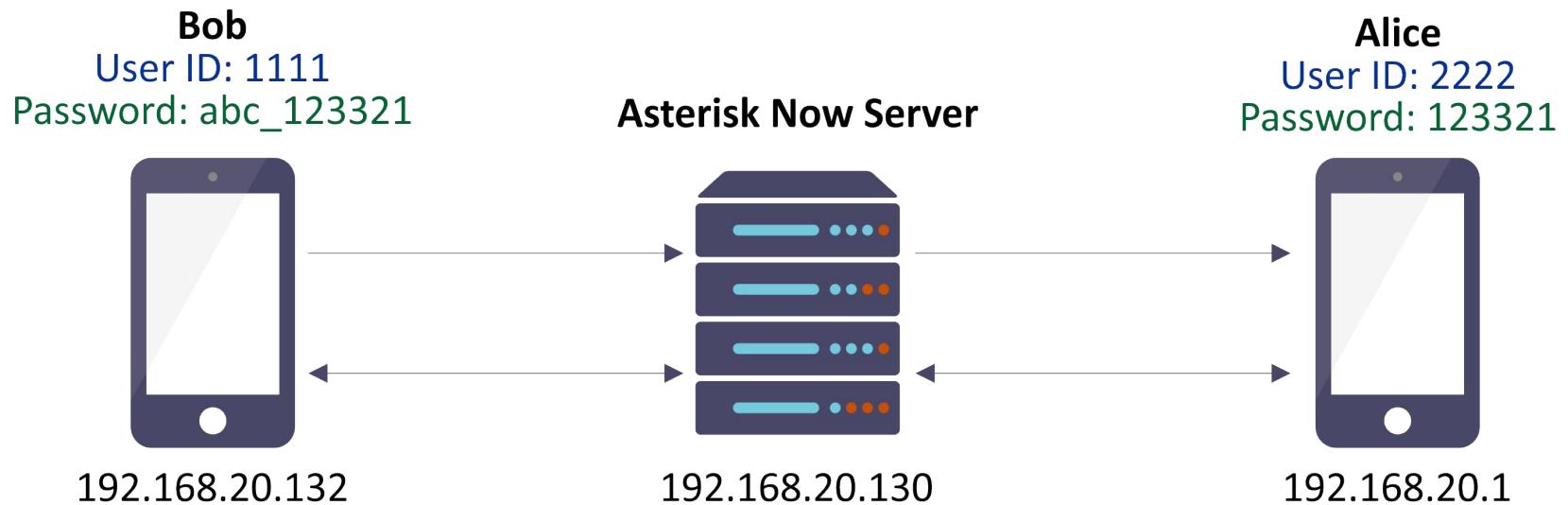
Asterisk Now



=



Scenario



Possible Configurations

- SIP + RTP
- SIP over TLS + RTP
- SIP + SRTP
- SIP over TLS + SRTP

Possible Configurations

- **SIP + RTP**
- SIP over TLS + RTP
- SIP + SRTP
- SIP over TLS + SRTP

SIP/SDP Packets

Complete_normal_call.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

sdp

No.	Time	Source	Destination	Protocol	Length	Ta	Info
34	17.478218	192.168.20.132	192.168.20.130	SIP/SDP	1374		Request: INVITE sip:2222@192.168.20.130
37	17.598013	192.168.20.130	192.168.20.1	SIP/SDP	1089		Request: INVITE sip:2222@192.168.20.1:52987;ob
71	22.145095	192.168.20.1	192.168.20.130	SIP/SDP	1014		Status: 200 OK
74	22.150650	192.168.20.130	192.168.20.132	SIP/SDP	1046		Status: 200 OK
78	22.158359	192.168.20.132	192.168.20.130	SIP/SDP	919		Request: UPDATE sip:192.168.20.130:5060

Frame 71: 1014 bytes on wire (8112 bits), 1014 bytes captured (8112 bits)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_f8:0d:44 (00:0c:29:f8:0d:44)
Internet Protocol Version 4, Src: 192.168.20.1, Dst: 192.168.20.130
User Datagram Protocol, Src Port: 52987, Dst Port: 5060
Session Initiation Protocol (200)
 Status-Line: SIP/2.0 200 OK
 Message Header
 Message Body
 Session Description Protocol
 Session Description Protocol Version (v): 0
 Owner/Creator, Session Id (o): - 3731351734 3731351735 IN IP4 192.168.5.103
 Session Name (s): pjmedia
 Bandwidth Information (b): AS:84
 Time Description, active time (t): 0 0
 Session Attribute (a): X-nat:0
 Media Description, name and address (m): audio 4000 RTP/AVP 0 101
 Connection Information (c): IN IP4 192.168.5.103
 Bandwidth Information (b): TIAS:64000
 Media Attribute (a): rtcp:4001 IN IP4 192.168.5.103
 Media Attribute (a): sendrecv
 Media Attribute (a): rtpmap:0 PCMU/8000
 Media Attribute (a): rtpmap:101 telephone-event/8000

RTCP Packets

Complete_normal_call.pcap

No. Time Source Destination Protocol Length Ta Info

2170	32.479679	192.168.20.1	192.168.20.130	RTCP	122	Sender Report	Source description
3108	37.158822	192.168.20.130	192.168.20.1	RTCP	106	Sender Report	Source description
3109	37.158934	192.168.20.130	192.168.20.132	RTCP	106	Sender Report	Source description
3136	37.287057	192.168.20.132	192.168.20.130	RTCP	122	Sender Report	Source description
3207	37.640101	192.168.20.1	192.168.20.130	RTCP	122	Sender Report	Source description

Frame 3108: 106 bytes on wire (848 bits), 106 bytes captured (848 bits)
Ethernet II, Src: VMware_f8:0d:44 (00:0c:29:f8:0d:44), Dst: VMware_c0:00:08 (00:50:56:c0:00:08)
Internet Protocol Version 4, Src: 192.168.20.130, Dst: 192.168.20.1
User Datagram Protocol, Src Port: 15675, Dst Port: 4001
Real-time Transport Control Protocol (Sender Report)
Real-time Transport Control Protocol (Source description)
[Stream setup by SDP (frame 37)]
10.. = Version: RFC 1889 Version (2)
.0. = Padding: False
.0 0001 = Source count: 1
Packet type: Source description (202)
Length: 2 (12 bytes)
Chunk 1, SSRC/CSRC 0x3C988166
Identifier: 0x3c988166 (1016627558)
SDES items
Type: CNAME (user and domain) (1)
Length: 0
Type: END (0)
[RTCP frame length check: OK - 64 bytes]

RTP Packets

Complete_normal_call.pcap

No. Time Source Destination Protocol Length Info

3103	37.140222	192.168.20.1	192.168.20.130	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x294823, Seq=5909, Time=120000
3104	37.141062	192.168.20.130	192.168.20.132	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0xAFD8AB5, Seq=21275, Time=120000
3105	37.143728	192.168.20.132	192.168.20.130	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x43572C47, Seq=30108, Time=120000
3106	37.144098	192.168.20.130	192.168.20.1	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x3C988166, Seq=26401, Time=120000
3110	37.160340	192.168.20.1	192.168.20.130	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x294823, Seq=5910, Time=120160

Frame 3106: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits)
Ethernet II, Src: Vmware_f8:0d:44 (00:0c:29:f8:0d:44), Dst: Vmware_c0:00:08 (00:50:56:c0:00:08)
Internet Protocol Version 4, Src: 192.168.20.130, Dst: 192.168.20.1
User Datagram Protocol, Src Port: 15674, Dst Port: 4000

Real-Time Transport Protocol

- [Stream setup by SDP (frame 37)]
 - 10.. = Version: RFC 1889 Version (2)
 - ..0. = Padding: False
 - ...0 = Extension: False
 - 0000 = Contributing source identifiers count: 0
 - 0.... = Marker: False
- Payload type: ITU-T G.711 PCMU (0)
- Sequence number: 26401
- [Extended sequence number: 91937]
- Timestamp: 120000
- Synchronization Source identifier: 0x3c988166 (1016627558)
- Payload: 5f5f606265696b6c6e70777b7d7d7e7d7a797efaf8fb7e7d...

Recovered VoIP Calls

Wireshark · VoIP Calls · Complete_normal_call

Start Time	Stop Time	Initial Speaker	From	To	Protocol	Duration	Packets	State	Comments
17.478218	38.752328	192.168.20.132	<sip:1111@192.168.20.130	<sip:2222@192.168.20.130	SIP	00:00:21	10	COMPLETED	INVITE 200
17.598013	38.757037	192.168.20.130	"Bob" <sip:1111@192.168.20.130	<sip:2222@192.168.20.1;ob	SIP	00:00:21	7	COMPLETED	INVITE 200

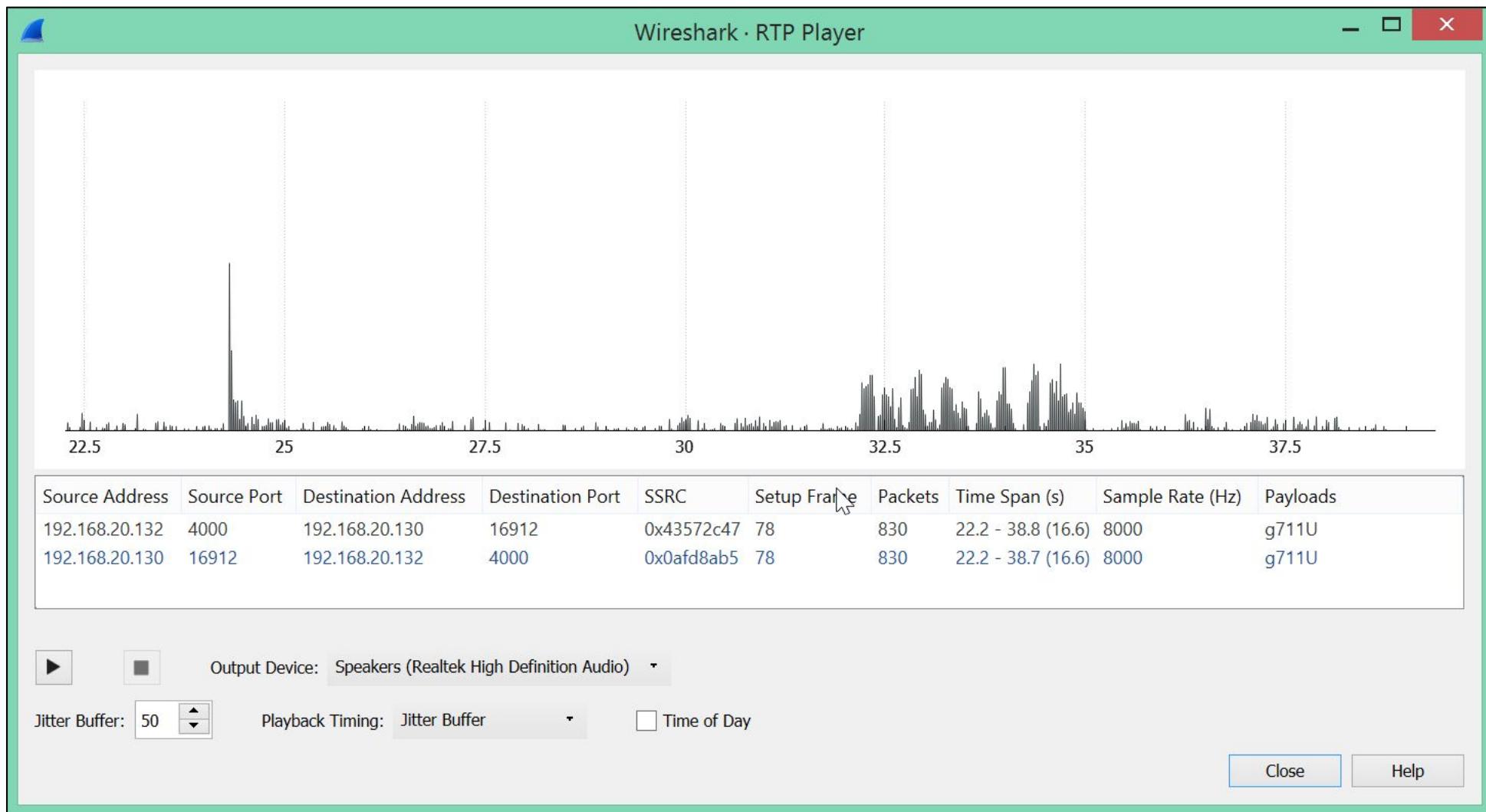
Time of Day

OK Cancel Prepare Filter Flow Sequence ► Play Streams Copy Help

Flow Sequence

Time	192.168.20.132	192.168.20.130	192.168.20.1	Comment
17.478218	58655	INVITE SDP (opus g711A g711U telephone-event)	5060	SIP INVITE From: <sip:1111@192.168.20.130 To:...
17.485438	58655	100 Trying	5060	SIP Status 100 Trying
17.597307	58655	180 Ringing	5060	SIP Status 180 Ringing
17.598013			5060 52987	SIP INVITE From: "Bob" <sip:1111@192.168.20.1...
17.603920			5060 52987	SIP Status 100 Trying
17.604301			5060 52987	SIP Status 180 Ringing
17.605610	58655	180 Ringing	5060	SIP Status 180 Ringing
22.145095			5060 52987	SIP Status 200 OK
22.148286			5060 52987	SIP Request INVITE ACK 200 CSeq:28747
22.150650	58655	200 OK SDP (g711U g711A telephone-event)	5060	SIP Status 200 OK
22.156664	58655	ACK	5060	SIP Request INVITE ACK 200 CSeq:20778
22.158359	58655	UPDATE SDP (g711U telephone-event)	5060	SIP UPDATE From: <sip:1111@192.168.20.130 To:...
22.160190			15674 4000	RTP, 830 packets. Duration: 16.581s SSRC: 0x294...
22.160191	4000	RTP (g711U)	16912	RTP, 830 packets. Duration: 16.581s SSRC: 0xAF...
22.161608	58655	200 OK SDP (g711U telephone-event)	5060	SIP Status 200 OK
22.161703	4000	RTP (g711U)	16912	RTP, 830 packets. Duration: 16.588s SSRC: 0x435...
22.162308			15674 4000	RTP, 830 packets. Duration: 16.589s SSRC: 0x3C9...
38.751436	58655	BYE	5060	SIP Request BYE CSeq:20780
38.752328	58655	200 OK	5060	SIP Status 200 OK

Reconstructed Call



Possible Configurations

- SIP + RTP
- SIP over TLS + RTP
- **SIP + SRTP**
- SIP over TLS + SRTP

SRTP key in SDP packet

Normal_Call_two_parties.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

sdp

No.	Time	Source	Destination	Protocol	Length	Ta	Info
128	27.128753	192.168.20.132	192.168.20.130	SIP/SDP	278		Request: INVITE sip:2222@192.168.20.130
131	27.301506	192.168.20.130	192.168.20.1	SIP/SDP	1174		Request: INVITE sip:2222@192.168.20.1:60168;ob
173	29.293203	192.168.20.1	192.168.20.130	SIP/SDP	1101		Status: 200 OK
178	29.314263	192.168.20.130	192.168.20.132	SIP/SDP	1131		Status: 200 OK

Internet Protocol Version 4, Src: 192.168.20.1, Dst: 192.168.20.130
User Datagram Protocol, Src Port: 60168, Dst Port: 5060
Session Initiation Protocol (200)
Status-Line: SIP/2.0 200 OK
Message Header
Message Body
Session Description Protocol
Session Description Protocol Version (v): 0
Owner/Creator, Session Id (o): - 3730471310 3730471311 IN IP4 192.168.5.114
Session Name (s): pjmedia
Bandwidth Information (b): AS:84
Time Description, active time (t): 0 0
Session Attribute (a): X-nat:0
Media Description, name and address (m): audio 4000 RTP/SAVP 0 101
Connection Information (c): IN IP4 192.168.5.114
Bandwidth Information (b): TIAS:64000
Media Attribute (a): rtcp:4001 IN IP4 192.168.5.114
Media Attribute (a): sendrecv
Media Attribute (a): rtpmap:0 PCMU/8000
Media Attribute (a): rtpmap:101 telephone-event/8000
Media Attribute (a): fmtp:101 0-16
Media Attribute (a): ssrc:965767637 cname:66bf37b000942b74
Media Attribute (a): crypto:1 AES_CM_128_HMAC_SHA1_80 inline:2stvbBcXXf3HtaHCSSB8WACeRBst9f7lwLqlzqE

SRTP Traffic

Normal_Call_two_parties.pcap

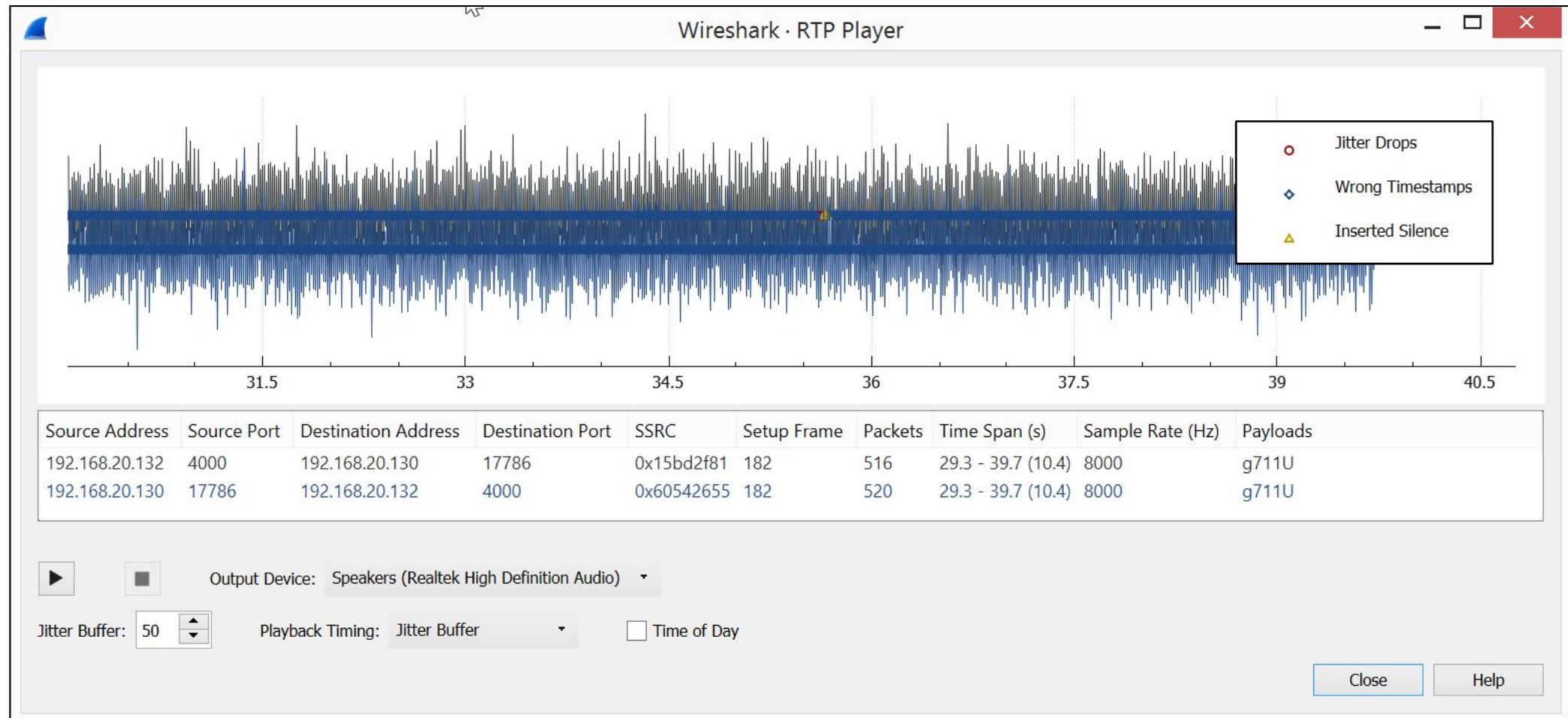
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

rtp

No.	Time	Source	Destination	Protocol	Length	Tag	Info
195	29.354843	192.168.20.132	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x15BD2F81, Seq=15576, Time=320
196	29.355005	192.168.20.130	192.168.20.1	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x4EFA778B, Seq=4650, Time=320
197	29.372665	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25653, Time=640
198	29.372952	192.168.20.130	192.168.20.132	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x60542655, Seq=16570, Time=640
199	29.375160	192.168.20.132	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x15BD2F81, Seq=15577, Time=480
200	29.375356	192.168.20.130	192.168.20.1	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x4EFA778B, Seq=4651, Time=480
204	29.393539	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25654, Time=800
205	29.393821	192.168.20.130	192.168.20.132	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x60542655, Seq=16571, Time=800
206	29.395768	192.168.20.132	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x15BD2F81, Seq=15578, Time=640

Frame 195: 224 bytes on wire (1792 bits), 224 bytes captured (1792 bits)
Ethernet II, Src: Vmware_6f:87:d6 (00:0c:29:6f:87:d6), Dst: Vmware_ff:65:9b (00:0c:ff:65:9b)
Internet Protocol Version 4, Src: 192.168.20.132, Dst: 192.168.20.130
User Datagram Protocol, Src Port: 4000, Dst Port: 17786
Real-Time Transport Protocol

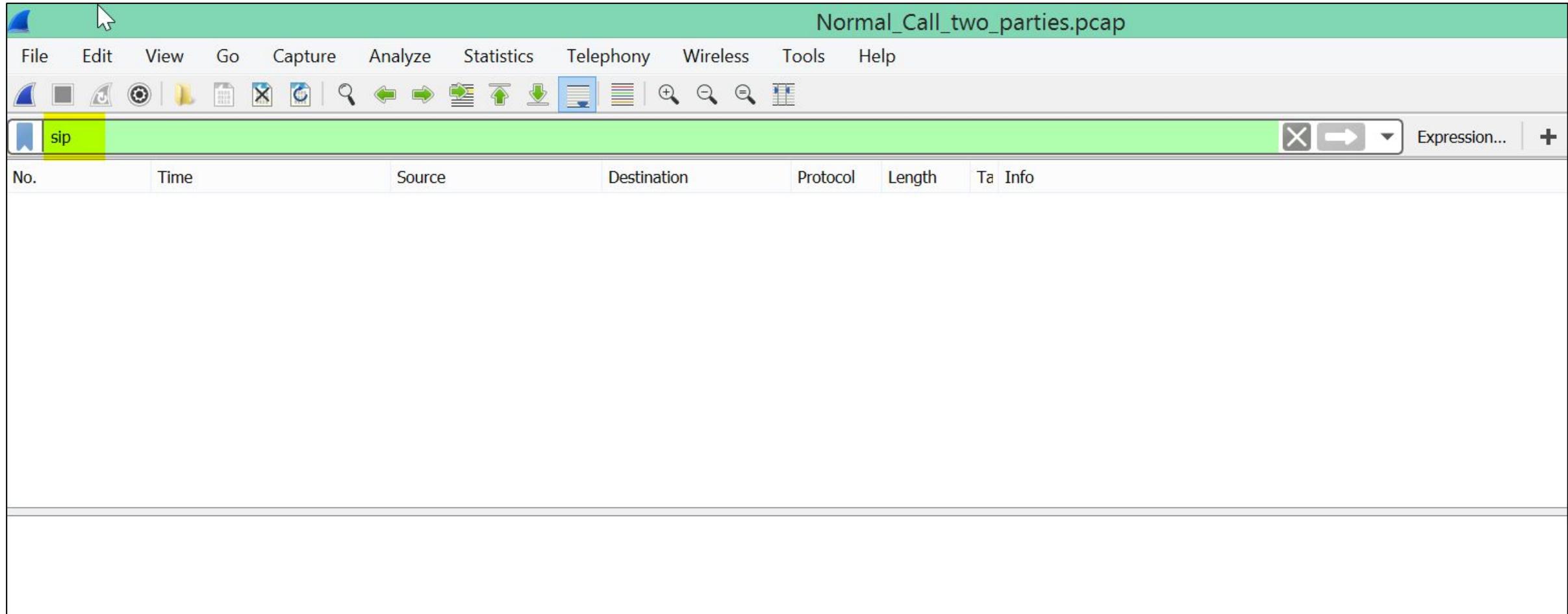
Encrypted Call



Possible Configurations

- SIP + RTP
- **SIP over TLS + RTP**
- SIP + SRTP
- SIP over TLS + SRTP

No SIP Traffic



TLS Traffic (SIP over TLS)

Normal_Call_two_parties.pcap

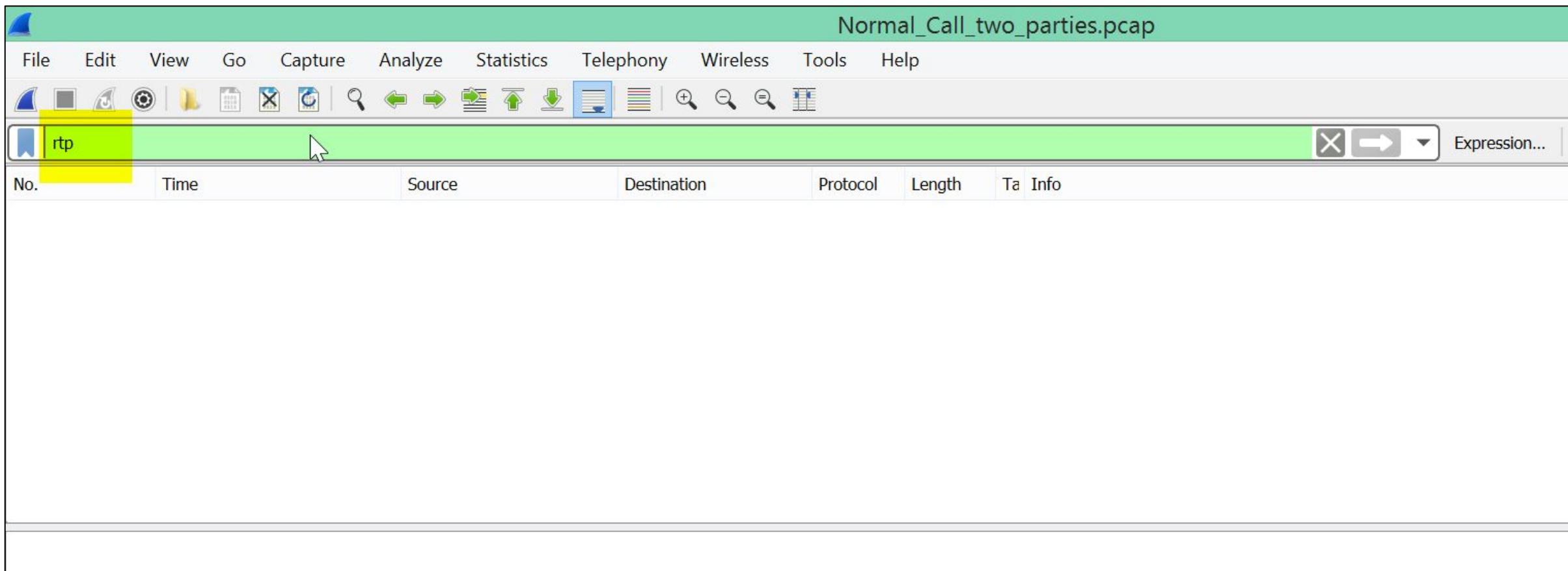
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ssl Expression... torrent cleanup_own_ssid cleanup_probe

No.	Time	Source	Destination	Protocol	Length	Ta	Info
4	0.011835	192.168.20.132	192.168.20.130	TLSv1	253		Client Hello
6	0.016672	192.168.20.130	192.168.20.132	TLSv1	1246		Server Hello, Certificate, Server Key Exchange, Certificate Request, Server Hell...
7	0.020041	192.168.20.132	192.168.20.130	TLSv1	200		Certificate, Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
8	0.020930	192.168.20.130	192.168.20.132	TLSv1	304		New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
9	0.021214	192.168.20.132	192.168.20.130	TLSv1	784		Application Data, Application Data
10	0.021727	192.168.20.130	192.168.20.132	TLSv1	688		Application Data, Application Data
11	0.022063	192.168.20.132	192.168.20.130	TLSv1	1088		Application Data, Application Data
12	0.025192	192.168.20.130	192.168.20.132	TLSv1	656		Application Data, Application Data
14	0.076523	192.168.20.130	192.168.20.132	TLSv1	1370		Application Data, Application Data, Application Data, Application Data
15	0.076842	192.168.20.132	192.168.20.130	TLSv1	928		Application Data, Application Data
17	0.117462	192.168.20.132	192.168.20.130	TLSv1	512		Application Data, Application Data

Frame 4: 253 bytes on wire (2024 bits), 253 bytes captured (2024 bits)
Ethernet II, Src: Vmware_6f:87:d6 (00:0c:29:6f:87:d6), Dst: Vmware_ab:b1:84 (00:0c:29:ab:b1:84)
Internet Protocol Version 4, Src: 192.168.20.132, Dst: 192.168.20.130
Transmission Control Protocol, Src Port: 49484, Dst Port: 5061, Seq: 1, Ack: 1, Len: 199
Secure Sockets Layer

No RTP Traffic



Why No RTP Traffic?

- Wireshark uses SDP packet to figure out the port RTP/SRTP stream will use.
 - SIP and SDP are encrypted, so wireshark can't figure out.

Frame	Source IP	Destination IP	Protocol	Sequence	Action
14	23.132688	192.168.20.130	192.168.20.1	RTCP	86 Receiver Report Source description
15	23.630139	192.168.20.132	192.168.20.130	SIP/SDP	1079 Request: INVITE sip:1111@192.168.20.130
16	23.631114	192.168.20.130	192.168.20.132	SIP	605 Status: 401 Unauthorized
17	23.633029	192.168.20.132	192.168.20.130	SIP	420 Request: ACK sip:1111@192.168.20.130

► Ethernet II, Src: Vmware_6f:87:d6 (00:0c:29:6f:87:d6), Dst: Vmware_ff:65:9b (00:0c:29:ff:65:9b)

► Internet Protocol Version 4, Src: 192.168.20.132, Dst: 192.168.20.130

► User Datagram Protocol, Src Port: 63214, Dst Port: 5060

▲ Session Initiation Protocol (INVITE)

► Request-Line: INVITE sip:1111@192.168.20.130 SIP/2.0

► Message Header

▲ Message Body

▲ Session Description Protocol

 Session Description Protocol Version (v): 0

 ► Owner/Creator, Session Id (o): - 3730467468 3730467468 IN IP4 192.168.20.132

 Session Name (s): pjmedia

 ► Bandwidth Information (b): AS:84

 ► Time Description, active time (t): 0 0

 ► Session Attribute (a): X-nat:0

▲ Media Description, name and address (m): audio 4004 RTP/AVP 123 8 0 101

 Media Type: audio

 Media Port: 4004

 Media Protocol: RTP/AVP

 Media Format: DynamicRTP-Type-123

 Media Format: ITU-T G.711 PCMA

 Media Format: ITU-T G.711 PCMU

 Media Format: DynamicRTP-Type-101

► Connection Information (c): IN IP4 192.168.20.132

► Bandwidth Information (b): TIAS:64000

► Media Attribute (a): rtcp:4005 IN IP4 192.168.20.132

Undecoded RTP Traffic

Normal_Call_two_parties.pcap

The screenshot shows a Wireshark capture of RTP traffic from a file named "Normal_Call_two_parties.pcap". The interface has a green header bar with menu items: File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with various icons. A search bar at the top says "Apply a display filter ... <Ctrl-/>". The main window displays a table of network frames. The columns are: No., Time, Source, Destination, Protocol, Length, and Info. The "Protocol" column is highlighted in yellow. The "Info" column shows RTP packets being exchanged between two hosts. Frame 662 is selected, and its details are shown in the bottom pane:

No.	Time	Source	Destination	Protocol	Length	Ta	Info
661	23.884012	192.168.20.130	192.168.20.132	UDP	214	17430 → 4000	Len=172
662	23.903032	192.168.20.132	192.168.20.130	UDP	214	4000 → 17430	Len=172
663	23.903302	192.168.20.130	192.168.20.1	UDP	214	16374 → 4000	Len=172
664	23.904066	192.168.20.1	192.168.20.130	UDP	214	4000 → 16374	Len=172
665	23.904167	192.168.20.130	192.168.20.132	UDP	214	17430 → 4000	Len=172
666	23.923545	192.168.20.132	192.168.20.130	UDP	214	4000 → 17430	Len=172
667	23.923824	192.168.20.130	192.168.20.1	UDP	214	16374 → 4000	Len=172
668	23.924438	192.168.20.1	192.168.20.130	UDP	214	4000 → 16374	Len=172
669	23.924589	192.168.20.130	192.168.20.132	UDP	214	17430 → 4000	Len=172
670	23.943786	192.168.20.132	192.168.20.130	UDP	214	4000 → 17430	Len=172
671	23.944063	192.168.20.130	192.168.20.1	UDP	214	16374 → 4000	Len=172

Frame 662 details:

- Frame 662: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits)
- Ethernet II, Src: Vmware_6f:87:d6 (00:0c:29:6f:87:d6), Dst: Vmware_ab:b1:84 (00:0c:29:ab:b1:84)
- Internet Protocol Version 4, Src: 192.168.20.132, Dst: 192.168.20.130
- User Datagram Protocol, Src Port: 4000, Dst Port: 17430
- Data (172 bytes)

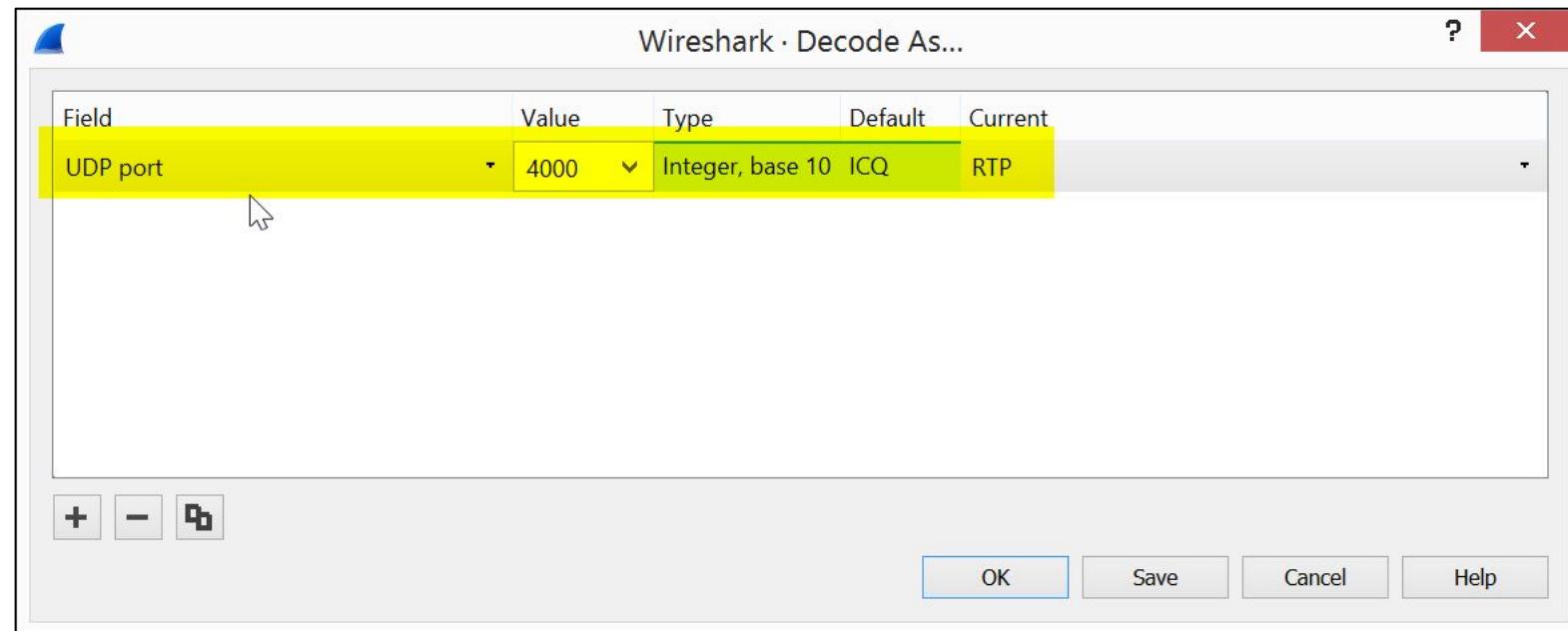
Decode As

The screenshot shows the Wireshark interface with a packet list titled "Normal_Call_two_parties.pcap". A right-click context menu is open over the second row of the list, which contains the following items:

- Mark/Unmark Packet
- Ignore/Unignore Packet
- Set/Unset Time Reference
- Time Shift...
- Packet Comment...
- Edit Resolved Name
- Apply as Filter
- Prepare a Filter
- Conversation Filter
- Colorize Conversation
- SCTP
- Follow
- Copy
- Protocol Preferences
- Decode As...** (highlighted with a yellow box)
- Show Packet in New Window

The packet details and bytes panes at the bottom show the selected UDP frame's structure.

Decode As RTP



RTP Traffic

Normal_Call_two_parties.pcap

No.	Time	Source	Destination	Protocol	Length	Ta	Info
653	23.843404	192.168.20.130	192.168.20.132	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x47A214A7, Seq=26410, Time=21440
654	23.862647	192.168.20.132	192.168.20.130	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x32D417E6, Seq=29493, Time=21600
655	23.863368	192.168.20.130	192.168.20.1	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x5B7C483D, Seq=10392, Time=21600
656	23.863618	192.168.20.1	192.168.20.130	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x294823, Seq=14717, Time=21600
657	23.863759	192.168.20.130	192.168.20.132	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x47A214A7, Seq=26411, Time=21600
658	23.882829	192.168.20.132	192.168.20.130	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x32D417E6, Seq=29494, Time=21760
659	23.883135	192.168.20.130	192.168.20.1	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x5B7C483D, Seq=10393, Time=21760
660	23.883902	192.168.20.1	192.168.20.130	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x294823, Seq=14718, Time=21760
661	23.884012	192.168.20.130	192.168.20.132	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x47A214A7, Seq=26412, Time=21760
662	23.903032	192.168.20.132	192.168.20.130	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x32D417E6, Seq=29495, Time=21920
663	23.903302	192.168.20.130	192.168.20.1	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x5B7C483D, Seq=10394, Time=21920

```
Frame 662: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits)
Ethernet II, Src: Vmware_6f:87:d6 (00:0c:29:6f:87:d6), Dst: Vmware_ab:b1:84 (00:0c:29:ab:b1:84)
Internet Protocol Version 4, Src: 192.168.20.132, Dst: 192.168.20.130
User Datagram Protocol, Src Port: 4000, Dst Port: 17430
Real-Time Transport Protocol
```

Checking RTP Streams

Normal_Call_two_parties.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source
653	23.843404	192.168.20.130
654	23.862647	192.168.20.132
655	23.863368	192.168.20.130
656	23.863618	192.168.20.1
657	23.863759	192.168.20.130
658	23.882829	192.168.20.132
659	23.883135	192.168.20.130
660	23.883902	192.168.20.1
661	23.884012	192.168.20.130
662	23.903032	192.168.20.132
663	23.903302	192.168.20.130

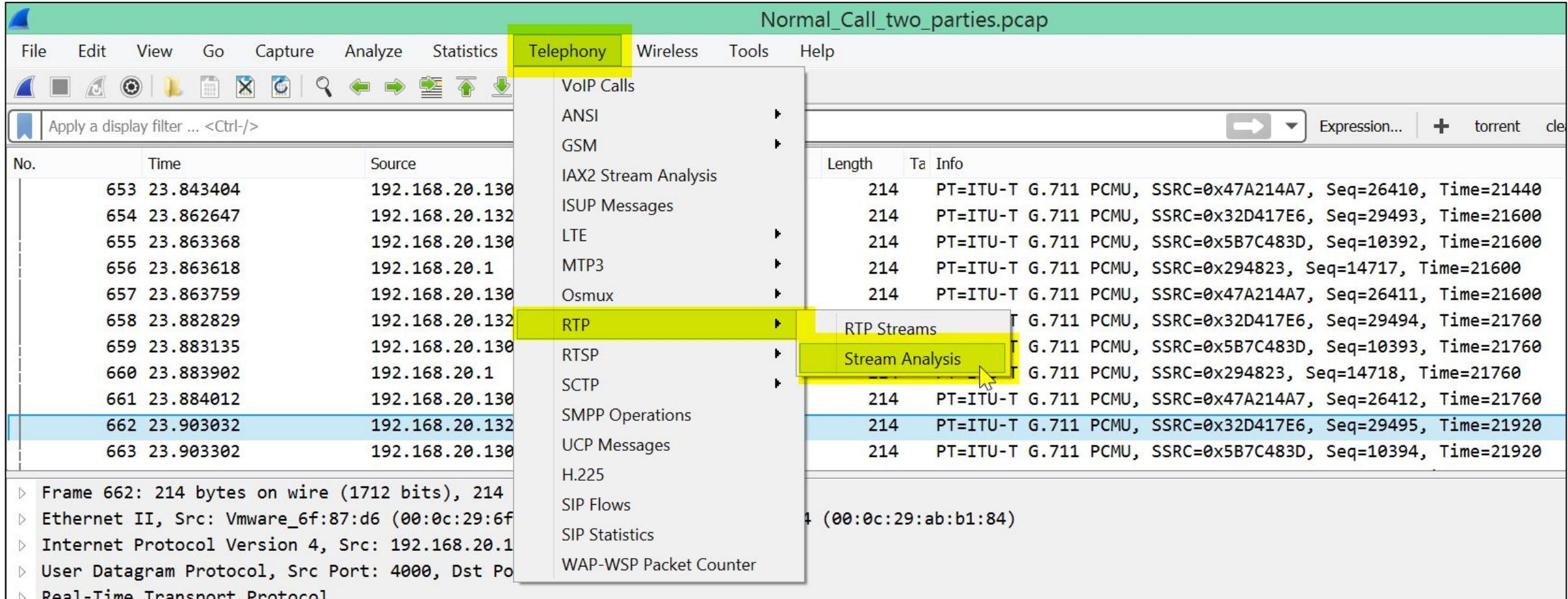
Frame 662: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface br0 at 2011-01-01 10:00:29.950000000 UTC
Ethernet II, Src: Vmware_6f:87:d6 (00:0c:29:6f:
Internet Protocol Version 4, Src: 192.168.20.1
User Datagram Protocol, Src Port: 4000, Dst Po
Real-Time Transport Protocol

VoIP Calls
ANSI
GSM
IAX2 Stream Analysis
ISUP Messages
LTE
MTP3
Osmux
RTP
RTSP
SCTP
SMPP Operations
UCP Messages
H.225
SIP Flows
SIP Statistics
WAP-WSP Packet Counter

Length Ta Info

214 PT=ITU-T G.711 PCMU, SSRC=0x47A214A7, Seq=26410, Time=21440
214 PT=ITU-T G.711 PCMU, SSRC=0x32D417E6, Seq=29493, Time=21600
214 PT=ITU-T G.711 PCMU, SSRC=0x5B7C483D, Seq=10392, Time=21600
214 PT=ITU-T G.711 PCMU, SSRC=0x294823, Seq=14717, Time=21600
214 PT=ITU-T G.711 PCMU, SSRC=0x47A214A7, Seq=26411, Time=21600
214 PT=ITU-T G.711 PCMU, SSRC=0x32D417E6, Seq=29494, Time=21760
214 PT=ITU-T G.711 PCMU, SSRC=0x5B7C483D, Seq=10393, Time=21760
214 PT=ITU-T G.711 PCMU, SSRC=0x294823, Seq=14718, Time=21760
214 PT=ITU-T G.711 PCMU, SSRC=0x47A214A7, Seq=26412, Time=21760
214 PT=ITU-T G.711 PCMU, SSRC=0x32D417E6, Seq=29495, Time=21920
214 PT=ITU-T G.711 PCMU, SSRC=0x5B7C483D, Seq=10394, Time=21920

(00:0c:29:ab:b1:84)



Analysing RTP Streams

Wireshark · RTP Stream Analysis · Normal_Call_two_parties

192.168.20.132:4000 ↔ 192.168.20.130:17430

Forward

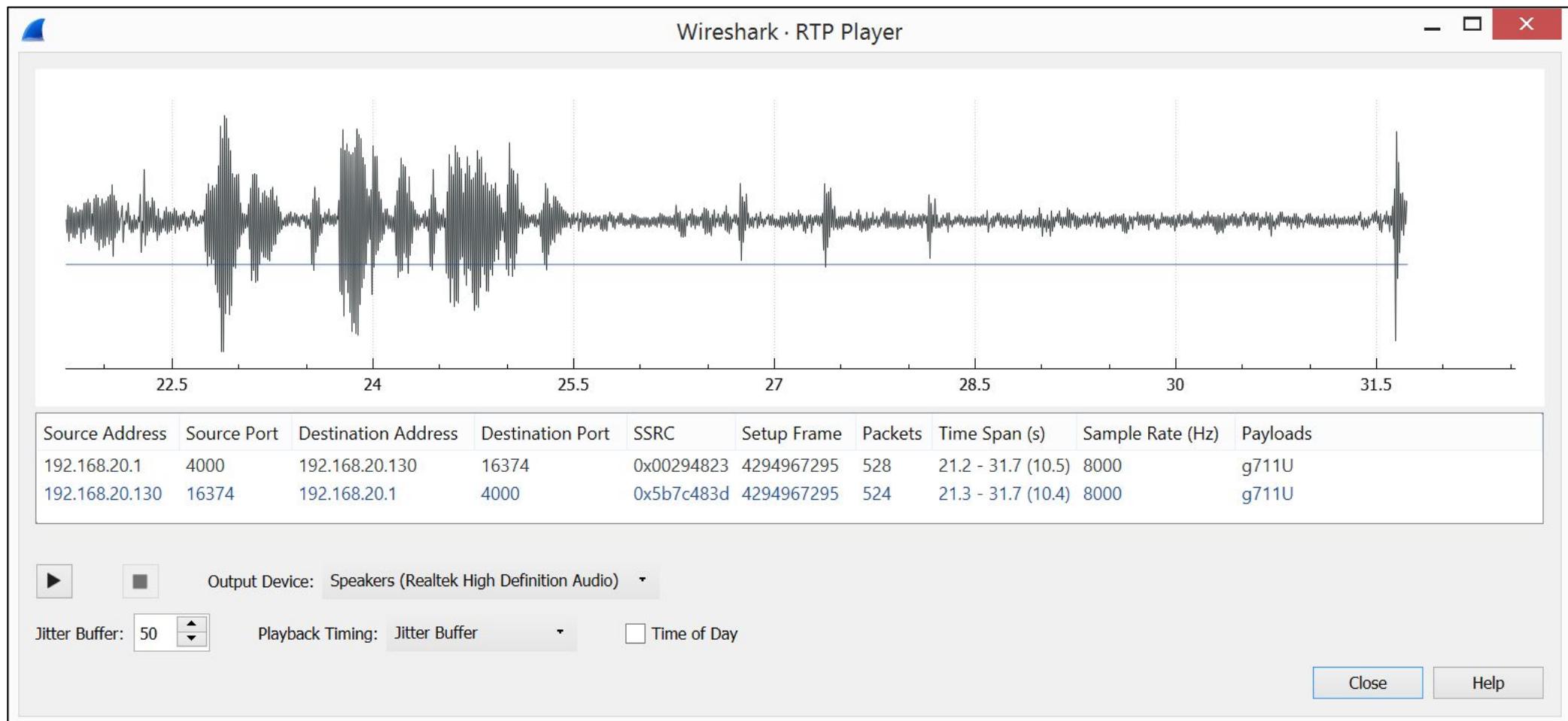
Packet	Sequence	Delta (ms)	Jitter (ms)	Skew	Bandwidth	Marker	Status
2251	29887	19.69	0.80	35.78	81.60	✓	
2243	29886	20.15	0.84	35.47	81.60	✓	
2239	29885	19.34	0.88	35.63	81.60	✓	
2235	29884	20.26	0.90	34.96	81.60	✓	
2231	29883	20.45	0.94	35.22	81.60	✓	
2227	29882	21.64	0.97	35.67	81.60	✓	
2223	29881	20.32	0.93	37.30	81.60	✓	
2220	29880	20.62	0.97	37.62	81.60	✓	
2215	29879	19.74	0.99	38.25	81.60	✓	
2211	29878	20.82	1.04	37.99	81.60	✓	
2207	29877	20.61	1.06	38.81	81.60	✓	
2203	29876	19.69	1.09	39.42	81.60	✓	
2199	29875	21.34	1.14	39.11	81.60	✓	
2195	29874	19.44	1.12	40.44	81.60	✓	
2192	29873	10.54	1.16	39.89	81.60	✓	
SSRC	0x32d417e6						
Max Delta	23.37 ms @ 989						
Max Jitter	1.49 ms						
Mean Jitter	0.87 ms						
Max Skew	40.44 ms						
RTP Packets	529						
Expected	529						
Lost	0 (0.00 %)						
Seq Errs	0						
Start at	21.201381 s @ 108						
Duration	10.52 s						
Clock Drift	-1030 ms						
Freq Drift	7217 Hz (-9.79 %)						
Reverse							
2195	29874	19.44	1.12	40.44	81.60	✓	
2192	29873	10.54	1.16	39.89	81.60	✓	
SSRC	0x47a214a7						
Max Delta	24.31 ms @ 180						
Max Jitter	1.32 ms						
Mean Jitter	0.77 ms						
Max Skew	30.31 ms						
RTP Packets	524						
Expected	524						
Lost	0 (0.00 %)						
Seq Errs	0						
Start at	21.269697 s @ 125						
Duration	10.44 s						
Clock Drift	-1053 ms						
Freq Drift	7193 Hz (-10.09 %)						
Forward to reverse							
start diff	0.068316 s @ 17						
2 streams found.							
Save		Close		▶ Play Streams		Help	

Forward to reverse
start diff 0.068316 s @ 17

2 streams found.

Save Close ▶ Play Streams Help

Playing RTP Streams



Possible Configurations

- SIP + RTP
- SIP over TLS + RTP
- SIP + SRTP
- **SIP over TLS + SRTP**

TLS key exchange methods

- TLS uses symmetric ciphers (i.e. AES, Blowfish) to encrypt the data
- **Two options under realistic approach**
 - DHE (Diffie Hellman Key Exchange)
 - RSA (Asymmetric encryption)

Diffie Hellman Exchange

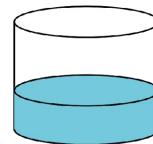
Assumption

- Attacker even after seeing the exchanged colours can't guess the secret colour.



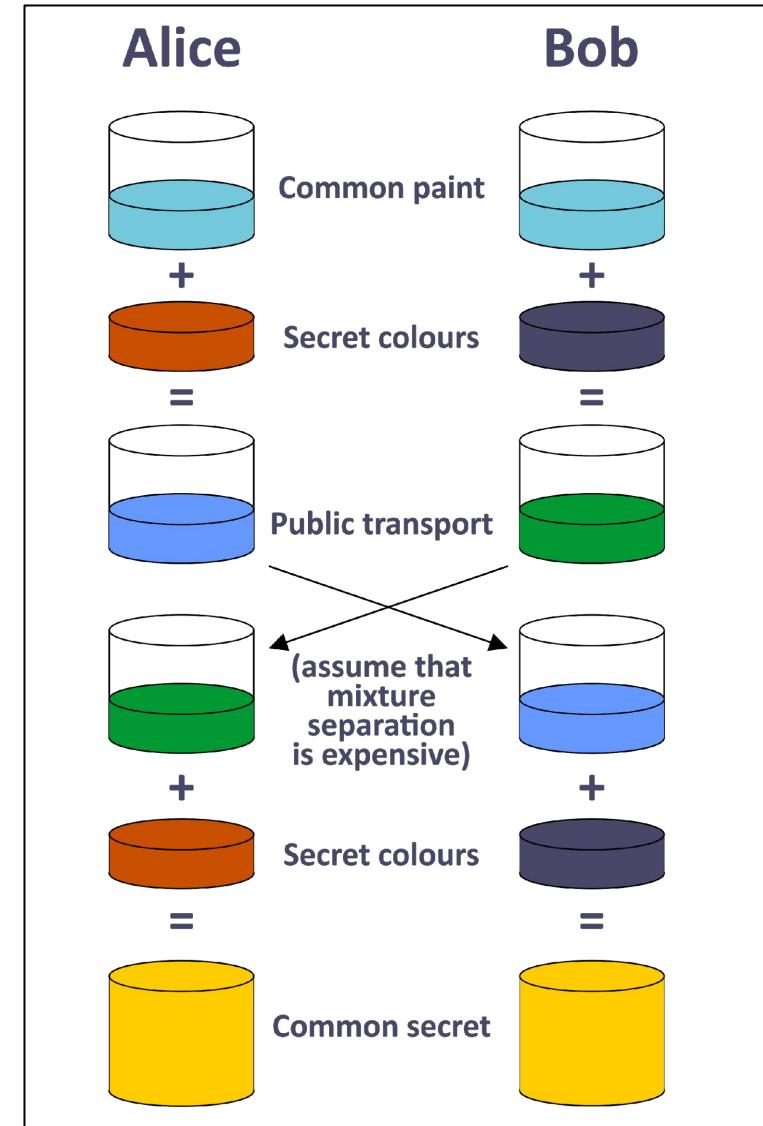
- Attacker knows

and also

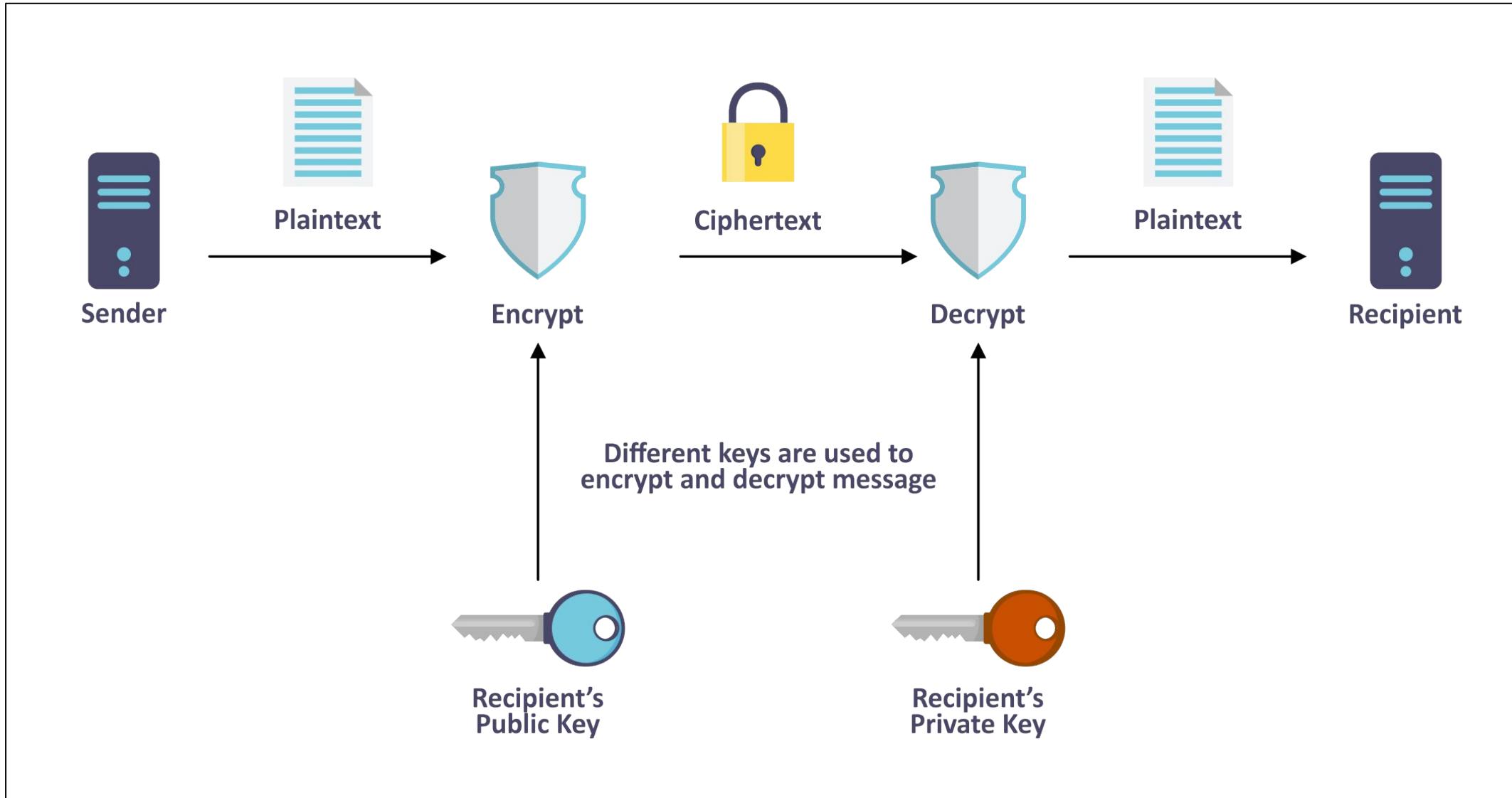


But can't know which colour is added.

More on: en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange



RSA (Asymmetric Encryption)



Observations?

- Can't recover keys derived with **ECDHE/DHE** by listening to traffic
- For **RSA**, if we can get private key of server, we can decrypt traffic

TLS Traffic (SIP over TLS)

Normal_Call_two_parties.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ssl Expression... torrent cleanup_own_s

No.	Time	Source	Destination	Protocol	Length	Ta	Info
15	10.172139	192.168.20.132	192.168.20.130	TLSv1	253		Client Hello
18	10.177721	192.168.20.130	192.168.20.132	TLSv1	1246		Server Hello, Certificate, Server Key Exchange, Certificate Request,
19	10.181390	192.168.20.132	192.168.20.130	TLSv1	200		Certificate, Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
20	10.182741	192.168.20.130	192.168.20.132	TLSv1	304		New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
21	10.183127	192.168.20.132	192.168.20.130	TLSv1	784		Application Data, Application Data
22	10.183904	192.168.20.130	192.168.20.132	TLSv1	688		Application Data, Application Data
23	10.184221	192.168.20.132	192.168.20.130	TLSv1	1088		Application Data, Application Data
24	10.187834	192.168.20.130	192.168.20.132	TLSv1	656		Application Data, Application Data
26	10.237912	192.168.20.130	192.168.20.132	TLSv1	1370		Application Data, Application Data, Application Data, Application Data
27	10.238220	192.168.20.132	192.168.20.130	TLSv1	928		Application Data, Application Data
29	10.277703	192.168.20.132	192.168.20.130	TLSv1	512		Application Data, Application Data

Frame 15: 253 bytes on wire (2024 bits), 253 bytes captured (2024 bits)
Ethernet II, Src: Vmware_6f:87:d6 (00:0c:29:6f:87:d6), Dst: Vmware_ff:65:9b (00:0c:29:ff:65:9b)
Internet Protocol Version 4, Src: 192.168.20.132, Dst: 192.168.20.130
Transmission Control Protocol, Src Port: 49532, Dst Port: 5061, Seq: 1, Ack: 1, Len: 199
Secure Sockets Layer

Diffie Hellman Exchange

Normal_Call_two_parties.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ssl

No.	Time	Source	Destination	Protocol	Length	Traffic Info
15	10.172139	192.168.20.132	192.168.20.130	TLSv1	253	Client Hello
18	10.177721	192.168.20.130	192.168.20.132	TLSv1	1246	Server Hello, Certificate, Server Key Exchange, Certificate Request, Server Hell...
19	10.181390	192.168.20.132	192.168.20.130	TLSv1	200	Certificate, Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
20	10.182741	192.168.20.130	192.168.20.132	TLSv1	304	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
21	10.183127	192.168.20.132	192.168.20.130	TLSv1	784	Application Data, Application Data
22	10.183904	192.168.20.130	192.168.20.132	TLSv1	688	Application Data, Application Data
23	10.184221	192.168.20.132	192.168.20.130	TLSv1	1088	Application Data, Application Data
24	10.187834	192.168.20.130	192.168.20.132	TLSv1	656	Application Data, Application Data
26	10.237912	192.168.20.130	192.168.20.132	TLSv1	1370	Application Data, Application Data, Application Data

Frame 19: 200 bytes on wire (1600 bits), 200 bytes captured (1600 bits)
Ethernet II, Src: Vmware_6f:87:d6 (00:0c:29:6f:87:d6), Dst: Vmware_ff:65:9b (00:0c:29:ff:65:9b)
Internet Protocol Version 4, Src: 192.168.20.132, Dst: 192.168.20.130
Transmission Control Protocol, Src Port: 49532, Dst Port: 5061, Seq: 200, Ack: 1193, Len: 146

Secure Sockets Layer

- ▶ TLSv1 Record Layer: Handshake Protocol: Certificate
 - Content Type: Handshake (22)
 - Version: TLS 1.0 (0x0301)
 - Length: 70
- ◀ Handshake Protocol: Client Key Exchange
 - Handshake Type: Client Key Exchange (16)
 - Length: 66
 - ◀ EC Diffie-Hellman Client Params
 - Pubkey Length: 65
 - Pubkey: 04e1bbe88bbb7a1912ebce555234524173dcb7cd098cd041...
- ▶ TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
- ▶ TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message

Undecoded SRTP Traffic

Normal_Call_two_parties.pcap

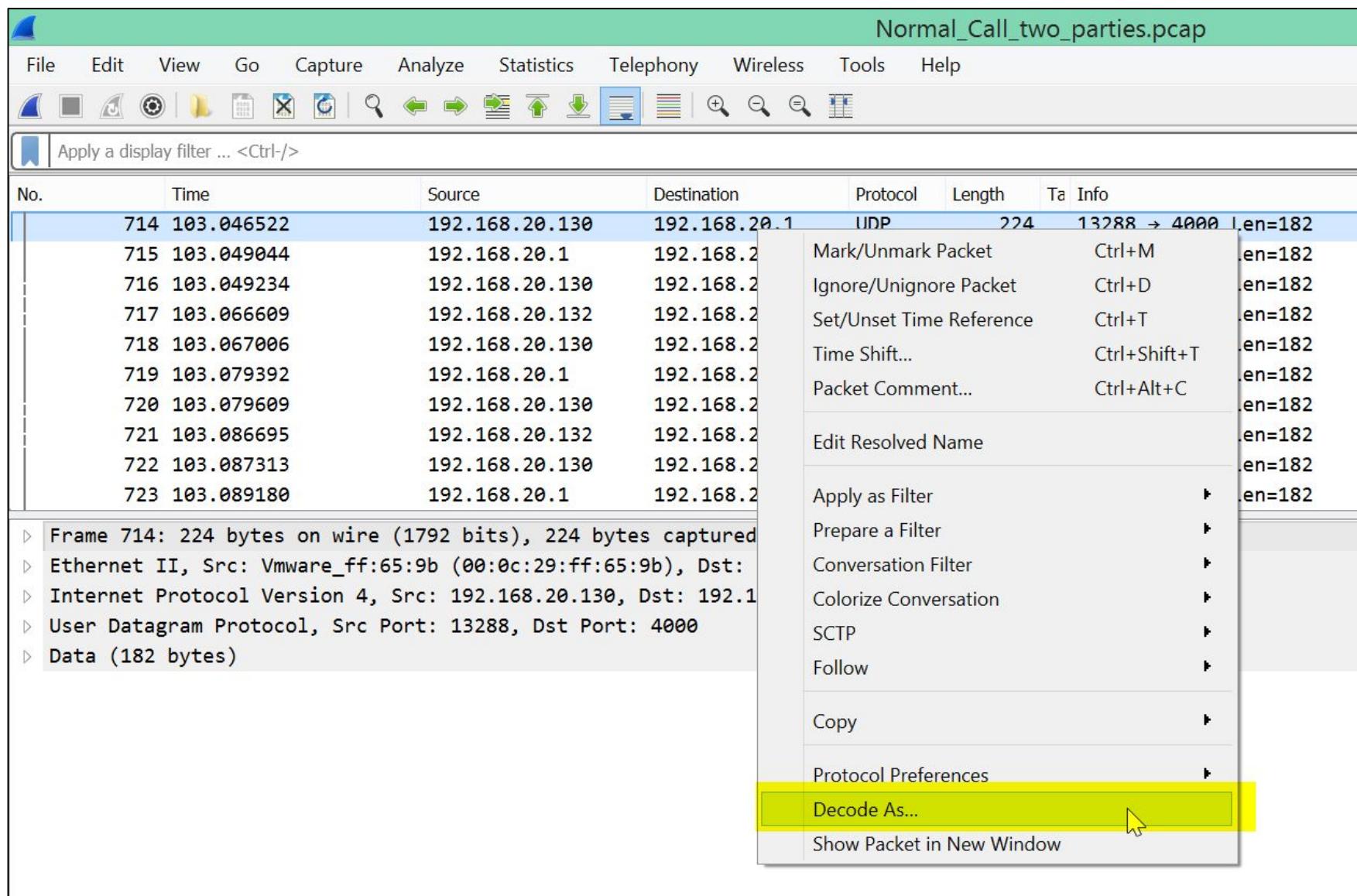
The screenshot shows a Wireshark interface with the file 'Normal_Call_two_parties.pcap' loaded. The packet list table displays 10 UDP frames (No. 714 to 723) between two hosts: 192.168.20.130 and 192.168.20.1. The frames are color-coded by protocol, with UDP frames highlighted in yellow. Frame 719 is selected and expanded in the details pane below.

No.	Time	Source	Destination	Protocol	Length	Ta	Info
714	103.046522	192.168.20.130	192.168.20.1	UDP	224	13288 → 4000	Len=182
715	103.049044	192.168.20.1	192.168.20.130	UDP	224	4000 → 13288	Len=182
716	103.049234	192.168.20.130	192.168.20.132	UDP	224	13408 → 4000	Len=182
717	103.066609	192.168.20.132	192.168.20.130	UDP	224	4000 → 13408	Len=182
718	103.067006	192.168.20.130	192.168.20.1	UDP	224	13288 → 4000	Len=182
719	103.079392	192.168.20.1	192.168.20.130	UDP	224	4000 → 13288	Len=182
720	103.079609	192.168.20.130	192.168.20.132	UDP	224	13408 → 4000	Len=182
721	103.086695	192.168.20.132	192.168.20.130	UDP	224	4000 → 13408	Len=182
722	103.087313	192.168.20.130	192.168.20.1	UDP	224	13288 → 4000	Len=182
723	103.089180	192.168.20.1	192.168.20.130	UDP	224	4000 → 13288	Len=182

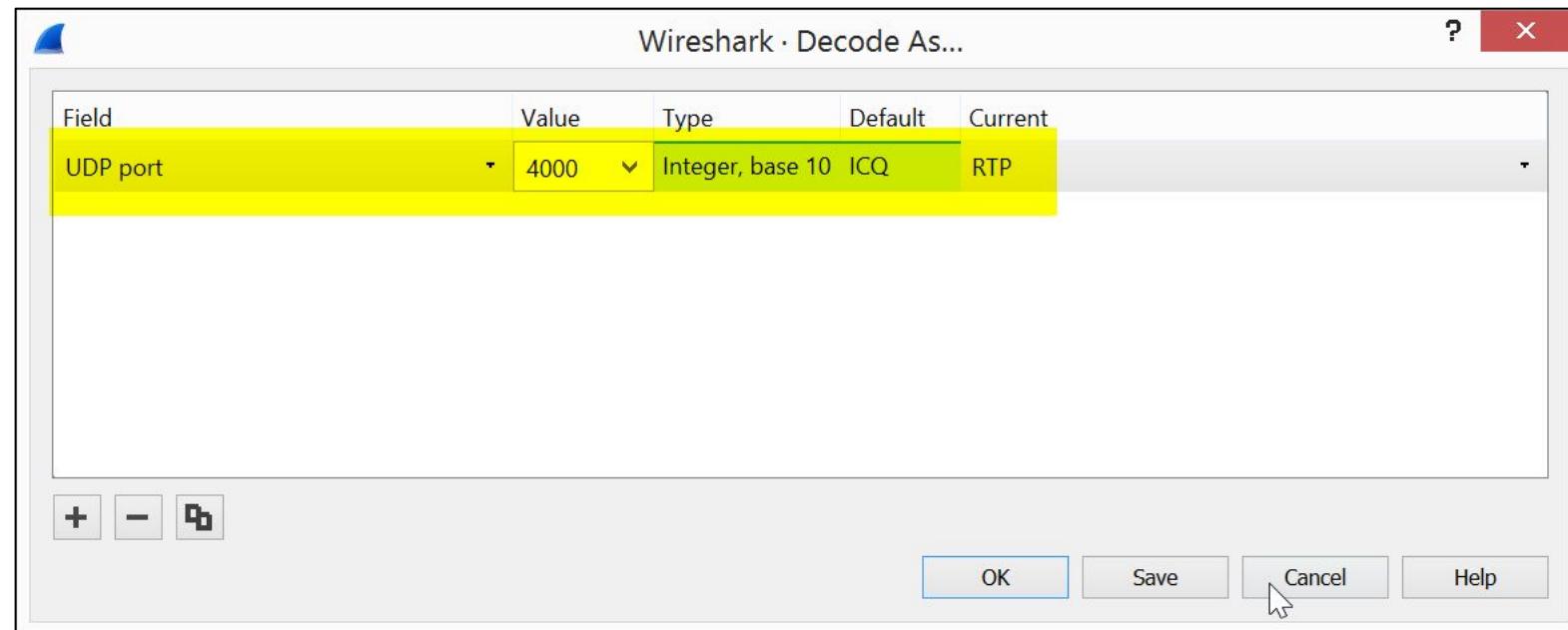
Frame 719 details:

- Frame 719: 224 bytes on wire (1792 bits), 224 bytes captured (1792 bits)
- Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_ff:65:9b (00:0c:29:ff:65:9b)
- Internet Protocol Version 4, Src: 192.168.20.1, Dst: 192.168.20.130
- User Datagram Protocol, Src Port: 4000, Dst Port: 13288
- Data (182 bytes)

Decode As



Decode As RTP



Checking RTP Streams

The screenshot shows the Wireshark interface with the file `Normal_Call_two_parties.pcap` loaded. The **Telephony** tab is selected. In the center pane, a list of RTP packets is displayed, each showing a length of 224 bytes and a PT=ITU-T G.711 PCMU payload. The **RTP** menu is open, and the **Stream Analysis** option is highlighted with a yellow box and a cursor arrow pointing to it.

No. Time Source

706	103.005510	192.168.20.130
707	103.018094	192.168.20.1
708	103.018467	192.168.20.130
709	103.025686	192.168.20.132
710	103.026046	192.168.20.130
711	103.038299	192.168.20.1
712	103.038516	192.168.20.130
713	103.045972	192.168.20.132
714	103.046522	192.168.20.130
715	103.049044	192.168.20.1

Frame 714: 224 bytes on wire (1792 bits), 224 bytes captured (1792 bits) on interface `Ethernet II, Src: Vmware_ff:65:9b (00:0c:29:ff:ff:ff)`

Internet Protocol Version 4, Src: 192.168.20.1

User Datagram Protocol, Src Port: 13288, Dst Port: 50000

Real-Time Transport Protocol

Length Ta Info

224	PT=ITU-T G.711 PCMU, SSRC=0x3EFBC86D, Seq=27905, Time=7040
224	PT=ITU-T G.711 PCMU, SSRC=0x4DCD5225, Seq=16871, Time=7040
224	PT=ITU-T G.711 PCMU, SSRC=0x6A41E0F3, Seq=385, Time=7040
224	PT=ITU-T G.711 PCMU, SSRC=0x294823, Seq=15098, Time=7200
224	PT=ITU-T G.711 PCMU, SSRC=0x3EFBC86D, Seq=27906, Time=7200
224	PT=ITU-T G.711 PCMU, SSRC=0x4DCD5225, Seq=16872, Time=7200
224	PT=ITU-T G.711 PCMU, SSRC=0x6A41E0F3, Seq=386, Time=7200
224	PT=ITU-T G.711 PCMU, SSRC=0x294823, Seq=15099, Time=7360
224	PT=ITU-T G.711 PCMU, SSRC=0x3EFBC86D, Seq=27907, Time=7360
224	PT=ITU-T G.711 PCMU, SSRC=0x4DCD5225, Seq=16873, Time=7360

3 (00:50:56:c0:00:08)

Analysing RTP Streams

Wireshark · RTP Stream Analysis · Normal_Call_two_parties

192.168.20.130:13288 ↔ 192.168.20.1:4000

Forward

SSRC 0x3efbc86d
Max Delta 40.57 ms @ 540
Max Jitter 1.52 ms
Mean Jitter 0.88 ms
Max Skew 53.93 ms
RTP Packets 615
Expected 616
Lost 1 (0.16 %)
Seq Errs 1
Start at 102.171933 s @ 525
Duration 12.25 s
Clock Drift -6967 ms
Freq Drift 3451 Hz (-56.86 %)

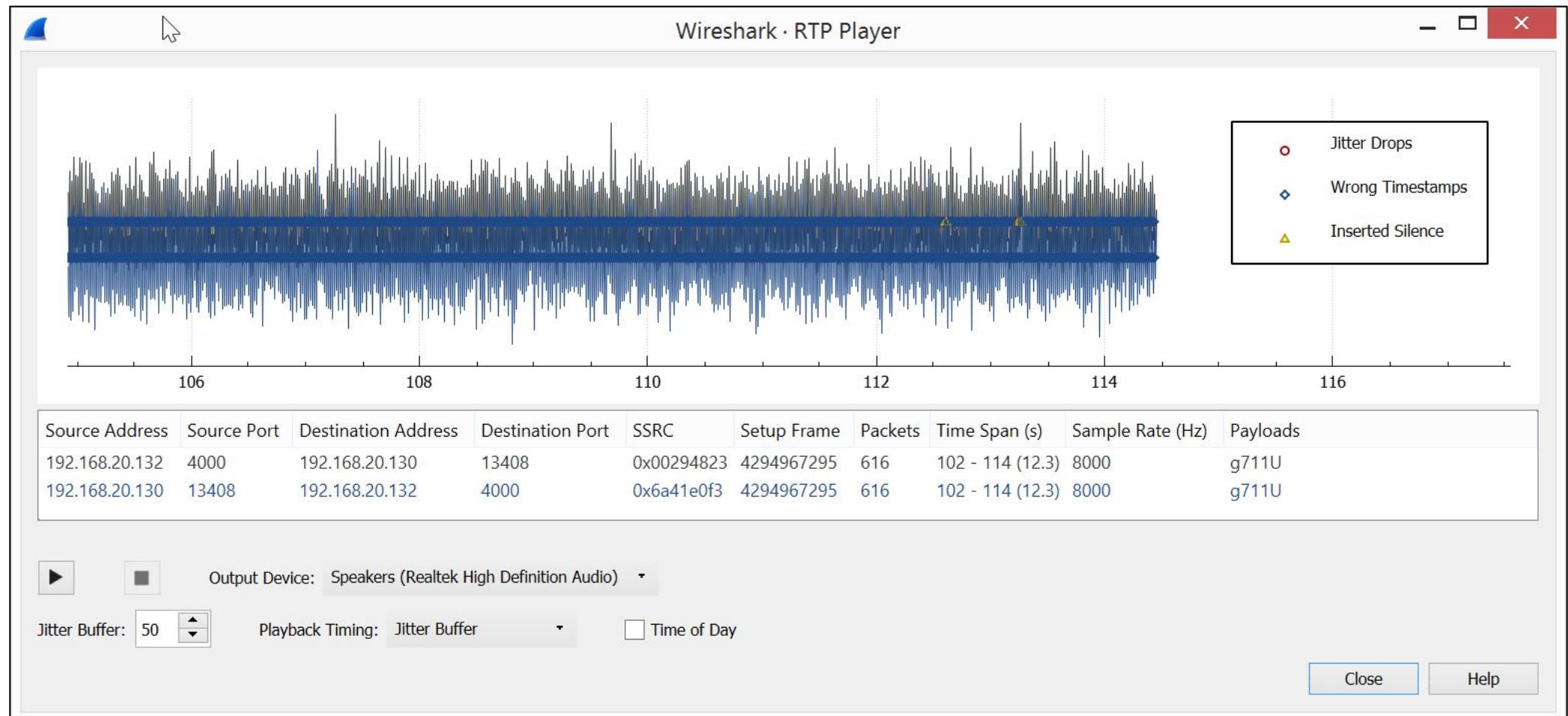
Packet	Sequence	Delta (ms)	Jitter (ms)	Skew	Bandwidth	Marker	Status	
525	27862	0.00	0.00	0.00	1.68	✓		
527	27863	3.53	1.03	16.47	3.36	✓		
540	27865	40.57	1.00	15.90	5.04	✓	Wrong sequence number	
545	27866	20.60	0.98	15.30	6.72	✓		
549	27867	19.99	0.91	15.31	8.40	✓		
553	27868	20.83	0.91	14.49	10.08	✓		
557	27869	19.74	0.87	14.74	11.76	✓		
561	27870	20.25	0.83	14.49	13.44	✓		
565	27871	20.00	0.78	14.49	15.12	✓		
570	27872	10.97	1.29	23.51	16.80	✓		
575	27873	19.61	1.24	23.91	18.48	✓		
579	27874	20.49	1.19	23.42	20.16	✓		
583	27875	19.54	1.14	23.88	21.84	✓		
587	27876	20.37	1.10	23.50	23.52	✓		
591	27877	19.71	1.05	23.79	25.20	✓		
SSRC	0x4dcd5225	595	27878	20.37	1.00	23.42	26.88	✓
Max Delta	30.43 ms @ 1370	599	27879	19.86	0.95	23.56	28.56	✓
Max Jitter	2.39 ms	603	27880	20.49	0.92	23.06	30.24	✓
Mean Jitter	0.90 ms	607	27881	20.59	0.90	22.48	31.92	✓
Max Skew	37.18 ms	611	27882	20.41	0.87	22.07	33.60	✓
RTP Packets	617	618	27883	20.74	0.86	21.32	35.28	✓
Expected	617	622	27884	19.93	0.81	21.39	36.96	✓
Lost	0 (0.00 %)	626	27885	20.33	0.78	21.06	38.64	✓
Seq Errs	0	630	27886	20.18	0.74	20.88	40.32	✓
Start at	102.157587 s @ 522	634	27887	21.32	0.78	19.56	42.00	✓
Duration	12.29 s	638	27888	20.63	0.77	18.93	43.68	✓
Clock Drift	-6961 ms	642	27889	19.52	0.75	19.42	45.36	✓
Freq Drift	3468 Hz (-56.64 %)	646	27890	20.61	0.74	18.81	47.04	✓

Forward to reverse
start diff -0.014346 s @ -3

2 streams found.

Save Close ▶ Play Streams Help

Playing RTP Streams



TLS Traffic (SIP over TLS)

Call_to_VoiceMail.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ssl

No.	Time	Source	Destination	Protocol	Length	Ta	Info
9	3.025978	192.168.20.132	192.168.20.130	TLSv1	253		Client Hello
11	3.031243	192.168.20.130	192.168.20.132	TLSv1	1030		Server Hello, Certificate, Certificate Request, Server Hello Done
12	3.032252	192.168.20.132	192.168.20.130	TLSv1	264		Certificate, Client Key Exchange, Change Cipher Spec, Encrypted Handsh...
13	3.033610	192.168.20.130	192.168.20.132	TLSv1	304		New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
14	3.035114	192.168.20.132	192.168.20.130	TLSv1	784		Application Data, Application Data
15	3.036454	192.168.20.130	192.168.20.132	TLSv1	688		Application Data, Application Data
16	3.036892	192.168.20.132	192.168.20.130	TLSv1	1088		Application Data, Application Data
17	3.039477	192.168.20.130	192.168.20.132	TLSv1	656		Application Data, Application Data
19	3.089799	192.168.20.130	192.168.20.132	TLSv1	1370		Application Data, Application Data, Application Data, Application Data
20	3.090170	192.168.20.132	192.168.20.130	TLSv1	928		Application Data, Application Data
22	3.130640	192.168.20.132	192.168.20.130	TLSv1	512		Application Data, Application Data
28	10.968782	192.168.20.132	192.168.20.130	TLSv1	1584		Application Data, Application Data
30	10.970517	192.168.20.130	192.168.20.132	TLSv1	688		Application Data, Application Data
31	10.970920	192.168.20.132	192.168.20.130	TLSv1	528		Application Data, Application Data
32	10.971375	192.168.20.132	192.168.20.130	TLSv1	1888		Application Data, Application Data
34	10.973943	192.168.20.130	192.168.20.132	TLSv1	496		Application Data, Application Data
36	11.075535	192.168.20.130	192.168.20.132	TLSv1	1184		Application Data, Application Data

Frame 9: 253 bytes on wire (2024 bits), 253 bytes captured (2024 bits)
Ethernet II, Src: Vmware_6f:87:d6 (00:0c:29:6f:87:d6), Dst: Vmware_ab:b1:84 (00:0c:29:ab:b1:84)
Internet Protocol Version 4, Src: 192.168.20.132, Dst: 192.168.20.130
Transmission Control Protocol, Src Port: 49481, Dst Port: 5061, Seq: 1, Ack: 1, Len: 199
Secure Sockets Layer

RSA based key exchange

Call_to_VoiceMail.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ssl

No.	Time	Source	Destination	Protocol	Length	Ta	Info
9	3.025978	192.168.20.132	192.168.20.130	TLSv1	253		Client Hello
11	3.031243	192.168.20.130	192.168.20.132	TLSv1	1030		Server Hello, Certificate, Certificate Request, Server Hello Done
12	3.032252	192.168.20.132	192.168.20.130	TLSv1	264		Certificate, Client Key Exchange, Change Cipher Spec, Encrypted Handsh...
13	3.033610	192.168.20.130	192.168.20.132	TLSv1	304		New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
14	3.035114	192.168.20.132	192.168.20.130	TLSv1	784		Application Data, Application Data
15	3.036454	192.168.20.130	192.168.20.132	TLSv1	688		Application Data, Application Data
16	3.036892	192.168.20.132	192.168.20.130	TLSv1	1088		Application Data, Application Data

Frame 12: 264 bytes on wire (2112 bits), 264 bytes captured (2112 bits)
Ethernet II, Src: Vmware_6f:87:d6 (00:0c:29:6f:87:d6), Dst: Vmware_ab:b1:84 (00:0c:29:ab:b1:84)
Internet Protocol Version 4, Src: 192.168.20.132, Dst: 192.168.20.130
Transmission Control Protocol, Src Port: 49481, Dst Port: 5061, Seq: 200, Ack: 977, Len: 210

Secure Sockets Layer

- ▷ TLSv1 Record Layer: Handshake Protocol: Certificate
- △ TLSv1 Record Layer: Handshake Protocol: Client Key Exchange
 - Content Type: Handshake (22)
 - Version: TLS 1.0 (0x0301)
 - Length: 134
- △ Handshake Protocol: Client Key Exchange
 - Handshake Type: Client Key Exchange (16)
 - Length: 130
- △ RSA Encrypted PreMaster Secret
 - Encrypted PreMaster length: 128
 - Encrypted PreMaster: 080fc2605ded04d2d87eb750a20e2df2ad8e66e211887e8e...

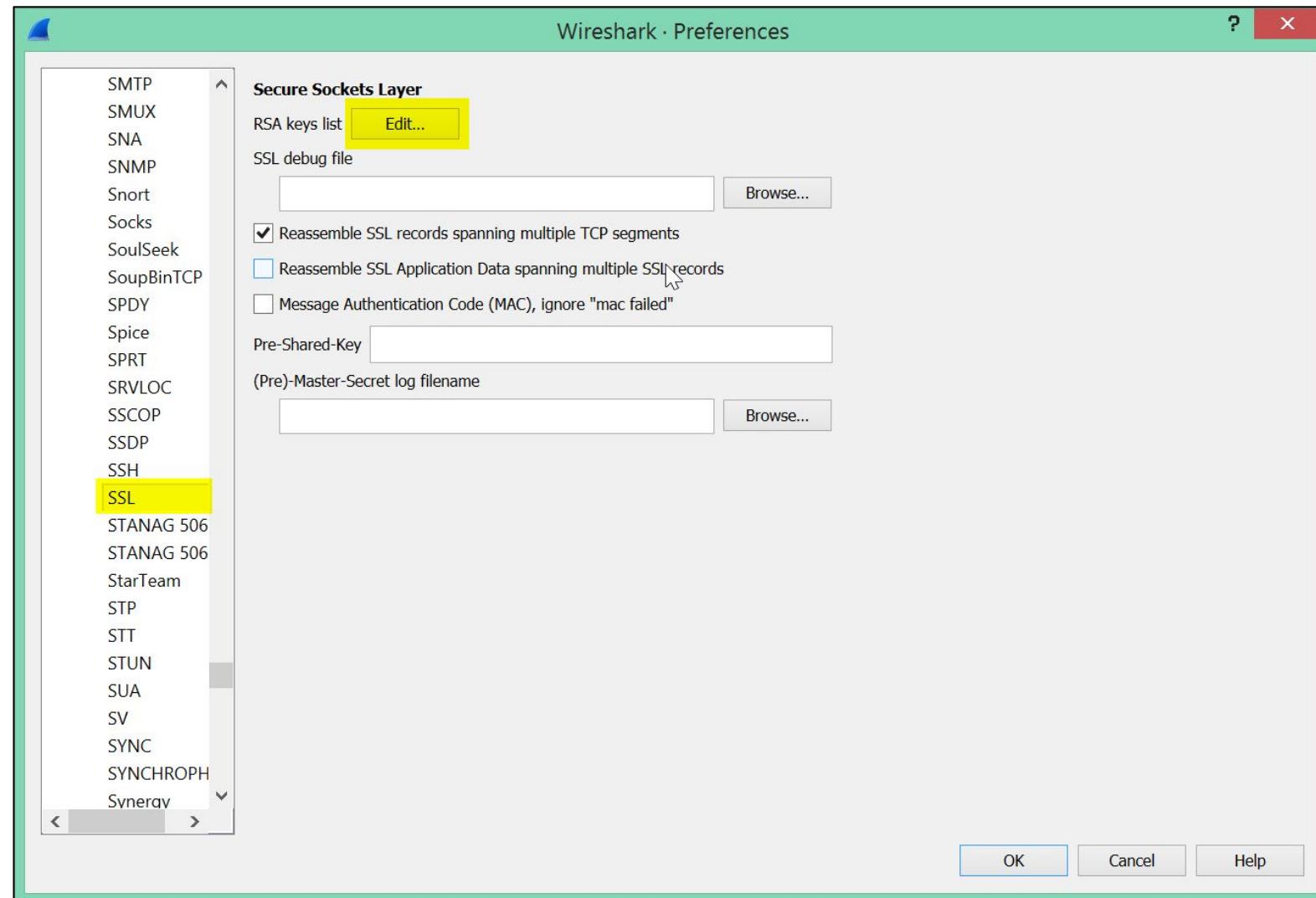
▷ TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
▷ TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message

Decrypting TLS traffic

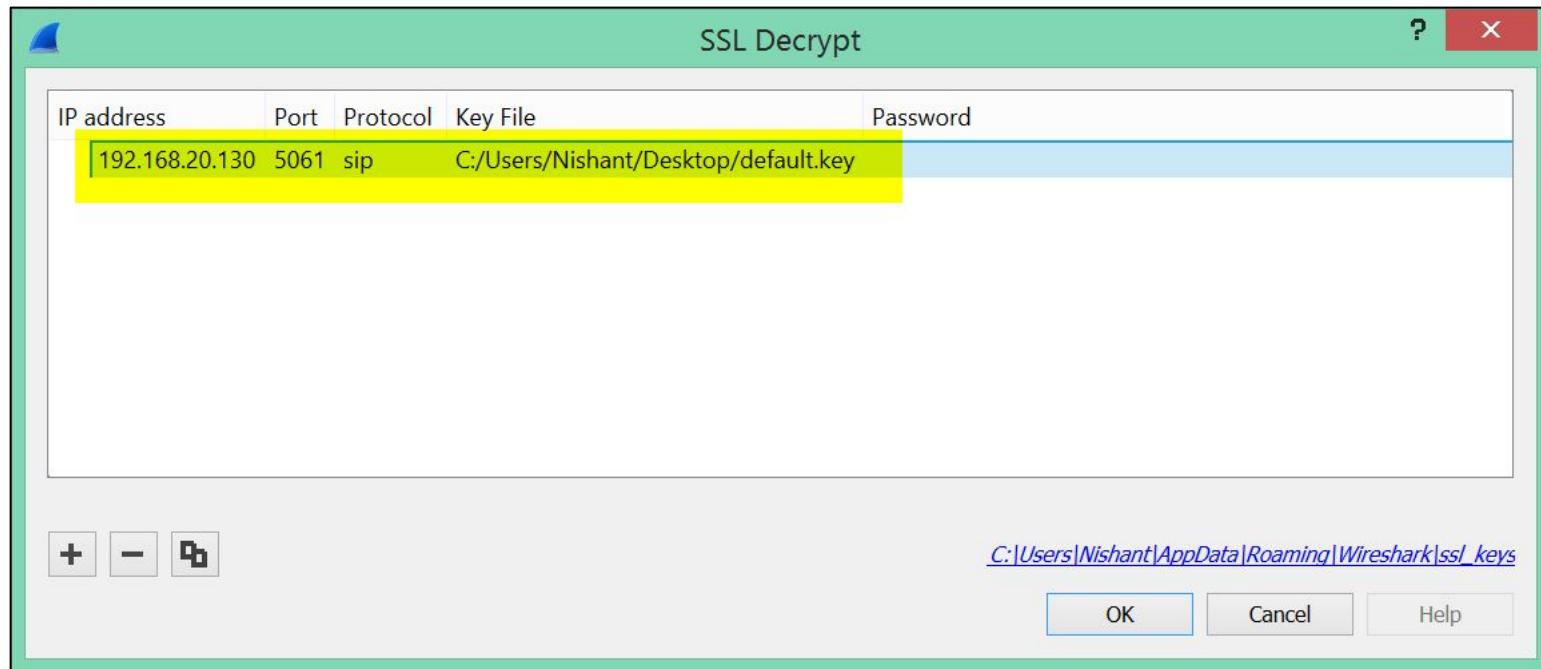
- RSA is used to exchange keys
- We can decrypt with private key installed on Asterisk One
- Keys and certificate location on Asterisk One: /etc/asterisk/keys
- We have to get the **default.key** from the server

```
[root@localhost ~]# cd /etc/asterisk/keys/
[root@localhost keys]# ls -l
total 32
-rw-rw-r--. 1 asterisk asterisk 215 Mar 19 03:59 ca.cfg
-rw-rw-r--. 1 asterisk asterisk 1789 Mar 19 03:59 ca.crt
-rw-rw-r--. 1 asterisk asterisk 3311 Mar 19 03:59 ca.key
-rw-----. 1 asterisk asterisk 1253 Mar 19 03:59 default.crt
-rw-----. 1 asterisk asterisk 595 Mar 19 03:59 default.csr
-rw-----. 1 asterisk asterisk 891 Mar 19 03:59 default.key
-rw-----. 1 asterisk asterisk 2144 Mar 19 03:59 default.pem
drwxrwxr-x. 2 asterisk asterisk 4096 Mar 19 03:59 integration
```

Edit > Preferences > Protocol > SSL



Adding Asterisk default private key



Decrypted SIP traffic

Screenshot of Wireshark showing decrypted SIP traffic for a call to a voicemail.

The capture file is titled "Call_to_VoiceMail.pcap".

Selected filter: sip

Table of captured frames:

No.	Time	Source	Destination	Protocol	Length	Ta	Info
17	3.039477	192.168.20.130	192.168.20.132	SIP	656		Status: 200 OK (1 binding)
19	3.089799	192.168.20.130	192.168.20.132	SIP	1370		Request: OPTIONS sip:1111@192.168.20.132:49481;transport=TLS;ob Requ...
20	3.090170	192.168.20.132	192.168.20.130	SIP	928		Status: 200 OK
22	3.130640	192.168.20.132	192.168.20.130	SIP	512		Status: 200 OK
28	10.968782	192.168.20.132	192.168.20.130	SIP/SDP	1584		Request: INVITE sip:2222@192.168.20.130;transport=tls
30	10.970517	192.168.20.130	192.168.20.132	SIP	688		Status: 401 Unauthorized
31	10.970920	192.168.20.132	192.168.20.130	SIP	528		Request: ACK sip:2222@192.168.20.130;transport=tls
32	10.971375	192.168.20.132	192.168.20.130	SIP/SDP	1888		Request: INVITE sip:2222@192.168.20.130;transport=tls
34	10.973943	192.168.20.130	192.168.20.132	SIP	496		Status: 100 Trying
36	11.075535	192.168.20.130	192.168.20.132	SIP/SDP	1184		Status: 200 OK
39	11.077488	192.168.20.132	192.168.20.130	SIP	512		Request: ACK sip:192.168.20.130:5061;transport=TLS
48	11.117569	192.168.20.132	192.168.20.130	SIP/SDP	1120		Request: UPDATE sip:192.168.20.130:5061;transport=TLS
50	11.118325	192.168.20.130	192.168.20.132	SIP/SDP	1152		Status: 200 OK
2302	33.695049	192.168.20.130	192.168.20.132	SIP	592		Request: BYE sip:1111@192.168.20.132:49481;transport=TLS;ob
2303	33.695785	192.168.20.132	192.168.20.130	SIP	496		Status: 200 OK

Frame details for frame 50:

- Frame 50: 1152 bytes on wire (9216 bits), 1152 bytes captured (9216 bits)
- Ethernet II, Src: Vmware_ab:b1:84 (00:0c:29:ab:b1:84), Dst: Vmware_6f:87:d6 (00:0c:29:6f:87:d6)
- Internet Protocol Version 4, Src: 192.168.20.130, Dst: 192.168.20.132
- Transmission Control Protocol, Src Port: 5061, Dst Port: 49481, Seq: 5985, Ack: 8868, Len: 1098
- Secure Sockets Layer
- Session Initiation Protocol (200)

SRTP key in SIP/SDP decrypted packet

Screenshot of Wireshark showing a SIP/SDP session between two hosts. The session consists of several SIP messages and an SDP offer. The SDP offer includes a media description and a media attribute containing an SRTP key.

The table below shows the captured SIP messages:

No.	Time	Source	Destination	Protocol	Length	Ta	Info
28	10.968782	192.168.20.132	192.168.20.130	SIP/SDP	1584		Request: INVITE sip:2222@192.168.20.130;transport=tls
32	10.971375	192.168.20.132	192.168.20.130	SIP/SDP	1888		Request: INVITE sip:2222@192.168.20.130;transport=tls
36	11.075535	192.168.20.130	192.168.20.132	SIP/SDP	1184		Status: 200 OK
48	11.117569	192.168.20.132	192.168.20.130	SIP/SDP	1120		Request: UPDATE sip:192.168.20.130:5061;transport=TLS
50	11.118325	192.168.20.130	192.168.20.132	SIP/SDP	1152		Status: 200 OK

Message details for frame 50:

- Frame 50: 1152 bytes on wire (9216 bits), 1152 bytes captured (9216 bits)
- Ethernet II, Src: Vmware_ab:b1:84 (00:0c:29:ab:b1:84), Dst: Vmware_6f:87:d6 (00:0c:29:6f:87:d6)
- Internet Protocol Version 4, Src: 192.168.20.130, Dst: 192.168.20.132
- Transmission Control Protocol, Src Port: 5061, Dst Port: 49481, Seq: 5985, Ack: 8868, Len: 1098
- Secure Sockets Layer
- Session Initiation Protocol (200)
 - Status-Line: SIP/2.0 200 OK
 - Message Header
 - Message Body
 - Session Description Protocol
 - Session Description Protocol Version (v): 0
 - Owner/Creator, Session Id (o): - 3730743973 3730743976 IN IP4 192.168.20.130
 - Session Name (s): Asterisk
 - Connection Information (c): IN IP4 192.168.20.130
 - Time Description, active time (t): 0 0
 - Media Description, name and address (m): audio 11382 RTP/SAVP 0 101
 - Media Attribute (a): crypto:1 AES_CM_128_HMAC_SHA1_80 inline:Hp12NjJuNxH5IceK1Mqm3/QaVPQjBFUwV+3SpXC8
 - Media Attribute (a): rtpmap:0 PCMU/8000
 - Media Attribute (a): rtpmap:101 telephone-event/8000

Open Source Tools for Decrypting SRTP

- SRTP Decrypt
- Libsrtp

SRTP Decrypt

- Tool to decipher SRTP packets
- Takes symmetric key to decrypt the SRTP traffic
- Output decrypted packets in form of hexdump
- Wireshark can reconstruct RTP packets from the hexdump

SRTP Decrypt

- GitHub: github.com/gteissier/srtp-decrypt

The screenshot shows the GitHub repository page for 'srtp-decrypt' owned by 'gteissier'. The repository has 5 stars, 12 forks, and 10 commits. It contains files like .gitignore, Makefile, README.md, marseillaise-srtp.pcap, srtp-decrypt.c, and srtp.c. The latest commit is from January 18, 2016.

Deciphers SRTP packets

10 commits | 1 branch | 0 releases | 1 contributor

Branch: master ▾ | New pull request | Find file | Clone or download ▾

File	Description	Time
.gitignore	Initial commit	5 years ago
Makefile	initial import.	5 years ago
README.md	Update README.md	5 years ago
marseillaise-srtp.pcap	initial import.	5 years ago
srtp-decrypt.c	Better default offset and handle correctly streams starting with seq ...	3 years ago
srtp.c	Increment offset using words, not bytes	2 years ago

SRTP Decrypt: Pre-Installation

- Installing libgcrypt

```
pentester@PentesterAcademy:~/work/srtp-decrypt$ sudo apt-get install libgcrypt-dev
sudo: unable to resolve host PentesterAcademy
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'libgcrypt20-dev' instead of 'libgcrypt-dev'
The following additional packages will be installed:
  libgcrypt20 libgpg-error-dev
Suggested packages:
```

- Installing libpcap

```
pentester@PentesterAcademy:~/work/srtp-decrypt$ sudo apt-get install libpcap-dev
sudo: unable to resolve host PentesterAcademy
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libpcap0.8-dev
The following NEW packages will be installed:
  libpcap-dev libpcap0.8-dev
```

SRTP Decrypt: Installation

- Cloning

```
root@PentesterAcademy:/work# git clone https://github.com/gteissier/srtp-decrypt.git
Cloning into 'srtp-decrypt'...
remote: Counting objects: 35, done.
remote: Total 35 (delta 0), reused 0 (delta 0), pack-reused 35
Unpacking objects: 100% (35/35), done.
```

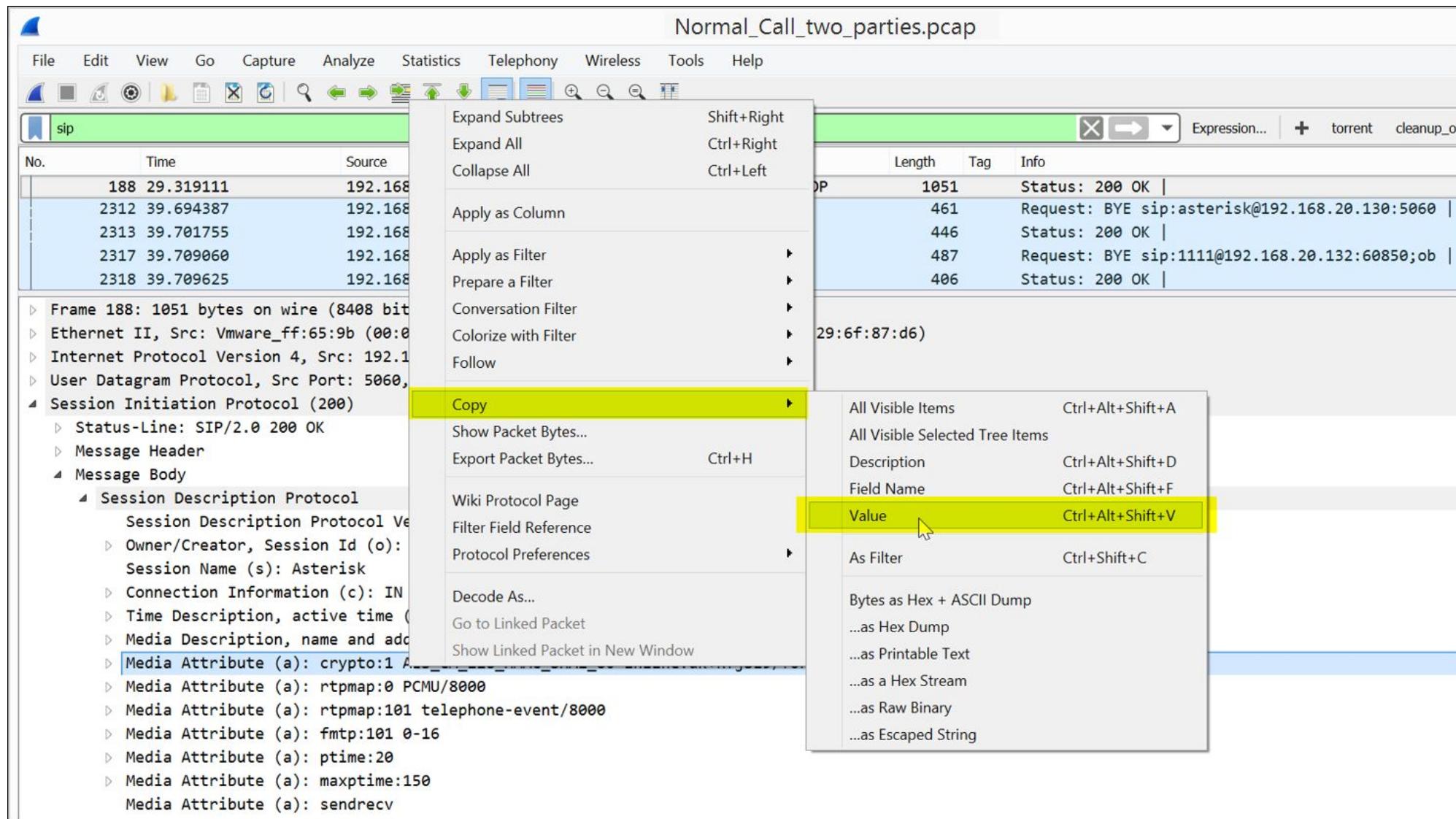
- Compiling

```
root@PentesterAcademy:/work/srtp-decrypt# make
cc -g -Os -Wall -c -o srtp.o srtp.c
cc -g -Os -Wall -c -o srtp-decrypt.o srtp-decrypt.c
cc -o srtp-decrypt srtp-decrypt.o srtp.o -lpcap -lgcrypt
```

SRTP Decrypt: Ready

```
root@PentesterAcademy:/work/srtp-decrypt# ls -l
total 2964
-rw-r--r-- 1 root root      273 Mar 17 05:36 Makefile
-rw-r--r-- 1 root root 2853144 Mar 17 05:36 marseillaise-srtp.pcap
-rw-r--r-- 1 root root     945 Mar 17 05:36 README.md
-rw-r--r-- 1 root root    22057 Mar 17 05:36 srtp.c
-rwxr-xr-x 1 root root   54112 Mar 17 05:40 srtp-decrypt
-rw-r--r-- 1 root root    3917 Mar 17 05:36 srtp-decrypt.c
-rw-r--r-- 1 root root   26464 Mar 17 05:40 srtp-decrypt.o
-rw-r--r-- 1 root root    2720 Mar 17 05:36 srtp.h
-rw-r--r-- 1 root root   52096 Mar 17 05:40 srtp.o
```

SRTP Decrypt: Copying SRTP key



SRTP Decrypt: UDP Ports

Normal_Call_two_parties.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

rtp

No.	Time	Source	Destination	Protocol	Length	Tag	Info
195	29.354843	192.168.20.132	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x15BD2F81, Seq=15576, Time=320
196	29.355005	192.168.20.130	192.168.20.1	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x4EFA778B, Seq=4650, Time=320
197	29.372665	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25653, Time=640
198	29.372952	192.168.20.130	192.168.20.132	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x60542655, Seq=16570, Time=640
199	29.375160	192.168.20.132	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x15BD2F81, Seq=15577, Time=480

Frame 195: 224 bytes on wire (1792 bits), 224 bytes captured (1792 bits)
Ethernet II, Src: Vmware_6f:87:d6 (00:0c:29:6f:87:d6), Dst: Vmware_ff:65:9b (00:0c:29:ff:65:9b)
Internet Protocol Version 4, Src: 192.168.20.132, Dst: 192.168.20.130
User Datagram Protocol, Src Port: 4000, Dst Port: 17786
Real-Time Transport Protocol

SRTP Decrypt: Decrypting SRTP Traffic

Command: ./srtp-decrypt -k uK+RfjSi9/fUFr8zoJu6zdqPw6MGtONhgX4yqwRj </Normal_Call_two_parties.pcap > decoded.raw

- -k : Defined SRTP key (uK+RfjSi9/fUFr8zoJu6zdqPw6MGtONhgX4yqwRj in this case)
- Normal_Call_two_parties.pcap Input file
- decoded.raw Output file

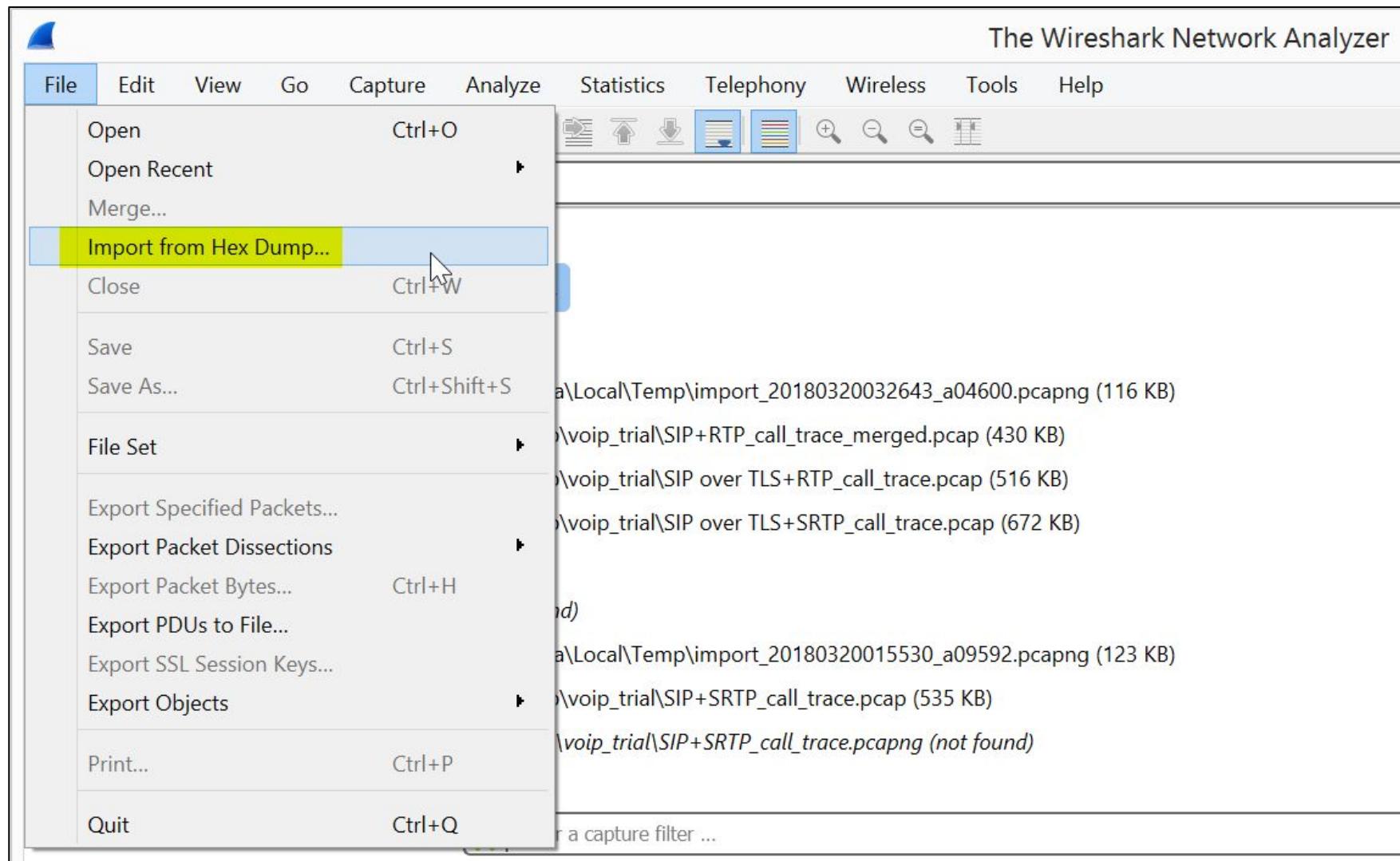
```
root@PentesterAcademy:/work/srtp-decrypt# ./srtp-decrypt -k uK+RfjSi9/fUFr8zoJu6zdqPw6M  
GtONhgX4yqwRj < ..../Normal_Call_two_parties.pcap > decoded.raw  
frame 0 dropped: decoding failed 'Permission denied'  
frame 1 dropped: decoding failed 'Permission denied'  
frame 2 dropped: decoding failed 'Permission denied'  
frame 3 dropped: decoding failed 'Permission denied'  
frame 4 dropped: decoding failed 'Permission denied'  
frame 5 dropped: decoding failed 'Permission denied'  
frame 6 dropped: decoding failed 'Permission denied'  
frame 7 dropped: decoding failed 'Permission denied'  
frame 8 dropped: decoding failed 'Permission denied'
```

SRTP Decrypt: decoded.raw

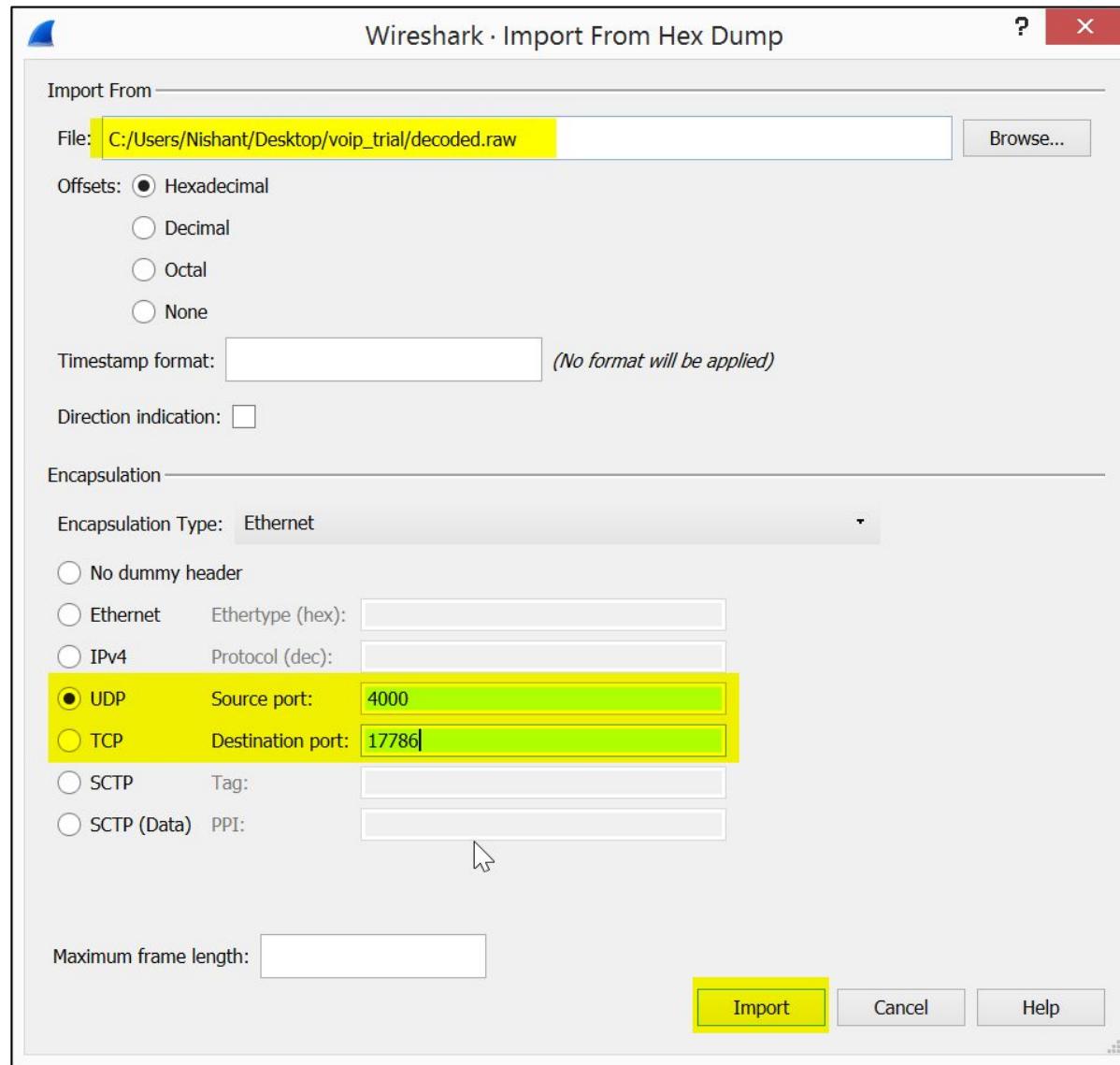
```
1 00:08.731764
2 0000 80 00 64 2e 00 00 00 a0 58 2f 39 0c 7e 7e 7e 7e
3 0010 7e 7e ff ff ff ff ff ff ff fe fe fe fe fe fe
4 0020 fe ff ff ff 7e 7e 7e ff ff ff ff 7e 7e 7e 7e 7e
5 0030 7e ff ff ff
6 0040 ff 7e 7e 7e ff ff fe fe fe fe fe fe fe fe fe fe
7 0050 fe fe fe fe ff ff 7e 7e 7e 7e 7e 7e 7e 7e 7e 7e
8 0060 7e 7e 7e 7e 7e 7e 7e ff ff ff 7e 7e 7e 7e 7e 7e
9 0070 7e 7e 7e ff ff ff fe fe fe fe fe fe fe fe ff
10 0080 7e 7e 7e 7e 7d 7d 7d 7d 7e 7e ff fe fe fe fe fe
11 0090 fe fe fe fe fe ff ff 7e 7e 7e 7d 7d 7d 7d 7d 7e
12 00a0 7e 7e ff ff ff fe fe fe fe fe fe fe fe
13 00:08.752171
14 0000 80 00 64 2f 00 00 01 40 58 2f 39 0c fe ff ff 7e
15 0010 7e 7e 7e 7d 7d 7d 7d 7d 7e 7e ff fe fe fd fd
16 0020 fd fe fe fe fe ff ff 7e 7e 7e 7e 7e 7e 7e 7e ff
17 0030 ff ff fe ff ff fe ff 7e 7e 7e 7e 7e 7e 7e 7e 7e
18 0040 7e ff ff ff fe fe fe ff ff 7e 7e 7e 7e 7d 7d 7d
19 0050 7d 7e 7e ff fe fe fd fd fd fd fe fe fe fe ff ff
20 0060 7e ff ff ff ff ff fe fe
21 0070 fe ff ff 7e ff 7e 7e 7e 7e 7e 7e 7e 7e 7e ff ff

"decoded.raw" 12838 lines --0%
```

SRTP Decrypt: Importing Decrypted Content



SRTP Decrypt: Importing Decrypted Content



SRTP Decrypt: Imported Decrypted UDP Packets

The screenshot shows the Wireshark interface with the title bar "import_20180320032955_a10724.pcapng". The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with various icons. A search bar at the top right contains the text "Expression...". The main window displays a table of network traffic with columns: No., Time, Source, Destination, Protocol, Length, Tag, and Info. Seven UDP frames are listed, all originating from 1.1.1.1 to 2.2.2.2, with a length of 214 bytes and a tag of 4000. The "Info" column shows "4000 → 17786 Len=172". Below the table, a detailed analysis pane shows the following for Frame 1:

- Frame 1: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits)
- Ethernet II, Src: Send_00 (20:53:45:4e:44:00), Dst: Receive_00 (20:52:45:43:56:00)
- Internet Protocol Version 4, Src: 1.1.1.1, Dst: 2.2.2.2
- User Datagram Protocol, Src Port: 4000, Dst Port: 17786
- Data (172 bytes)

SRTP Decrypt: Decode As

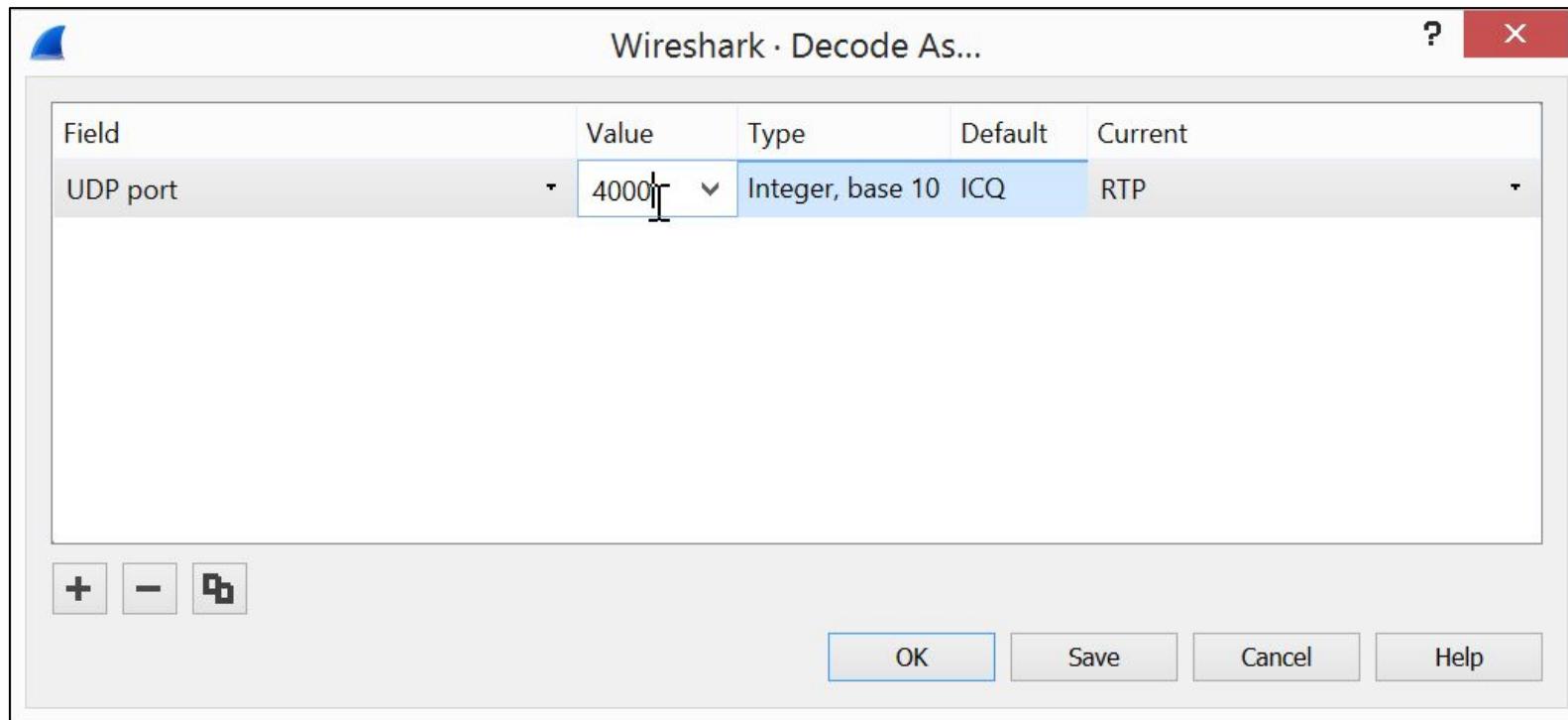
The screenshot shows the Wireshark interface with a packet list window titled "import_20180320032955_a10724.pcapng". The packet list displays several UDP packets from source IP 1.1.1.1 to destination IP 2.2.2.2, all with length 214 and tag 4000. A context menu is open over the first packet, listing options like "Mark/Unmark Packet", "Ignore/Unignore Packet", and "Decode As...". The "Decode As..." option is highlighted with a blue selection bar.

No.	Time	Source	Destination	Protocol	Length	Tag	Info
1	0.000000	1.1.1.1	2.2.2.2	UDP	214	4000	→ 17786 Len=172
2	0.000001	1.1.1.1			214	4000	→ 17786 Len=172
3	0.000002	1.1.1.1			214	4000	→ 17786 Len=172
4	0.000003	1.1.1.1			214	4000	→ 17786 Len=172
5	0.000004	1.1.1.1			214	4000	→ 17786 Len=172
6	0.000005	1.1.1.1			214	4000	→ 17786 Len=172
7	0.000006	1.1.1.1			214	4000	→ 17786 Len=172

Frame 1: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface
Ethernet II, Src: Send_00 (20:53:45:4e:44:00), Dst: (08:00:22:22:22:22) [ethernet]
Internet Protocol Version 4, Src: 1.1.1.1, Dst: 2.2.2.2 [ip]
User Datagram Protocol, Src Port: 4000, Dst Port: 17786 [udp]
Data (172 bytes)

Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift... Ctrl+Shift+T
Packet Comment... Ctrl+Alt+C
Edit Resolved Name
Apply as Filter
Prepare a Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As...
Show Packet in New Window

SRTP Decrypt: Decode As RTP



SRTP Decrypt: Decoded Packets

Screenshot of Wireshark showing captured RTP packets from a pcapng file named "import_20180320032955_a10724.pcapng".

The packet list table displays the following columns:

- No.
- Time
- Source
- Destination
- Protocol
- Length
- Tag
- Info

The table shows 7 captured RTP frames (Frame 1 to Frame 7) between source 1.1.1.1 and destination 2.2.2.2. All frames have a length of 214 bytes and are identified as RTP. The "Info" column provides detailed information about each frame, including PT=ITU-T G.711, PCMU, SSRC=0x60542655, and sequence numbers ranging from 16567 to 16573.

The packet details pane below the table shows the following analysis for Frame 1:

- Frame 1: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits)
- Ethernet II, Src: Send_00 (20:53:45:4e:44:00), Dst: Receive_00 (20:52:45:43:56:00)
- Internet Protocol Version 4, Src: 1.1.1.1, Dst: 2.2.2.2
- User Datagram Protocol, Src Port: 4000, Dst Port: 17786
- Real-Time Transport Protocol

SRTP Decrypt: Checking RTP Streams

Screenshot of Wireshark showing RTP streams analysis.

The screenshot shows the Wireshark interface with the file "import_20180320032955_a10724.pcapng" loaded. The "Telephony" tab is selected. A context menu is open over an RTP frame, specifically over the "RTP" entry in the VoIP Calls submenu. The menu items shown are "RTP Streams" and "Stream Analysis".

The main pane displays a list of RTP frames. The first few frames are:

No.	Time	Source	Protocol	Length	Tag	Info
1	0.000000	1.1.1.1	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x60542655, Seq=16567, Time=160
2	0.000001	1.1.1.1	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x60542655, Seq=16568, Time=320
3	0.000002	1.1.1.1	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x60542655, Seq=16569, Time=480
4	0.000003	1.1.1.1	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x60542655, Seq=16570, Time=640
5	0.000004	1.1.1.1	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x60542655, Seq=16571, Time=800
6	0.000005	1.1.1.1	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x60542655, Seq=16572, Time=960
7	0.000006	1.1.1.1	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x60542655, Seq=16573, Time=1120

The details pane shows the following information for the selected RTP frame:

- Frame 1: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface
- Ethernet II, Src: Send_00 (20:53:45:4e:44:00), Dst: (08:00:22:00:00:00)
- Internet Protocol Version 4, Src: 1.1.1.1, Dst: 2.2.2.2
- User Datagram Protocol, Src Port: 4000, Dst Port: 4000
- Real-Time Transport Protocol

SRTP Decrypt: Analysing RTP Streams

Wireshark · RTP Stream Analysis · import_20180320032955_a10724

1.1.1.1:4000 ↔
2.2.2.2:17786

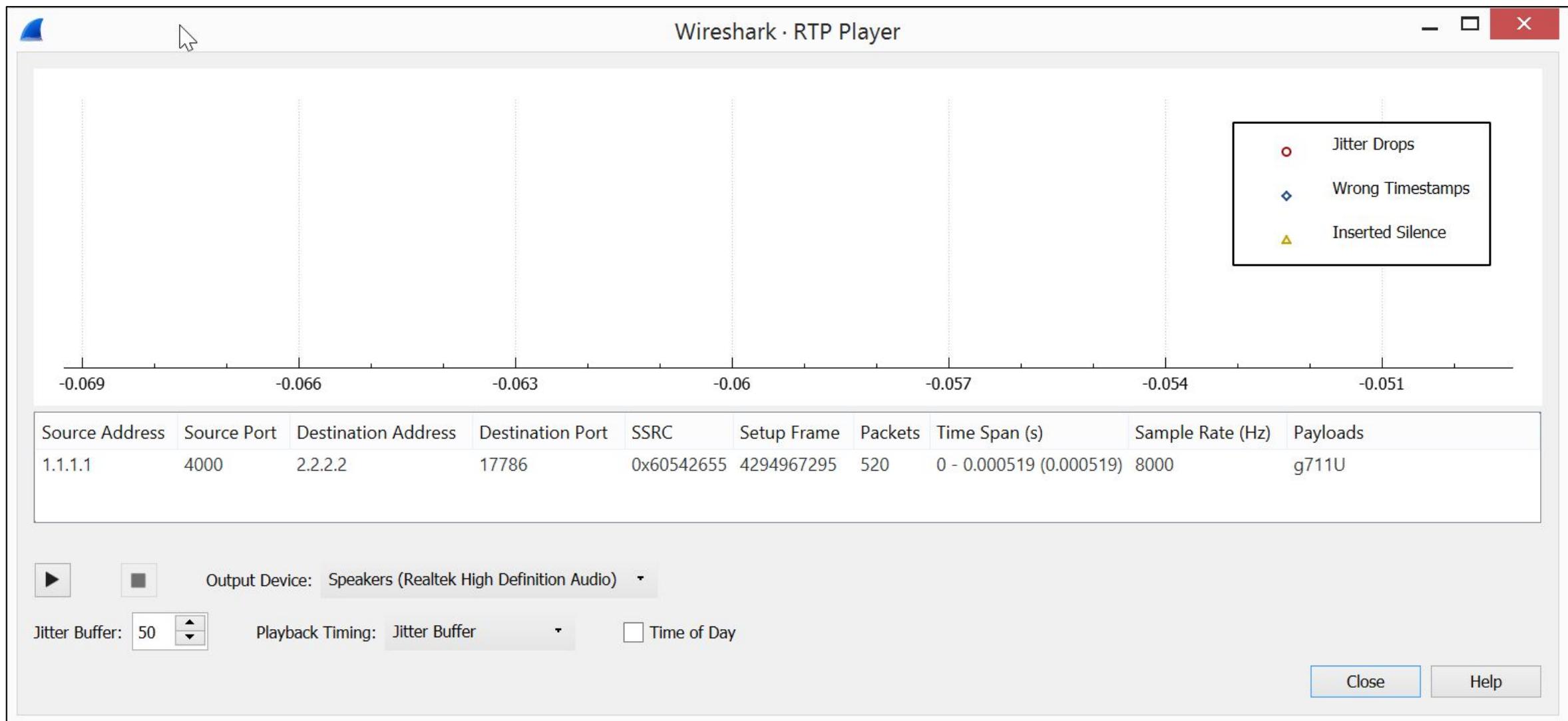
Forward

Packet	Sequence	Delta (ms)	Jitter (ms)	Skew	Bandwidth	Marker	Status
520	17086	0.00	20.00	10379.48	832.00	✓	
519	17085	0.00	20.00	10359.48	830.40	✓	
518	17084	0.00	20.00	10339.48	828.80	✓	
517	17083	0.00	20.00	10319.48	827.20	✓	
516	17082	0.00	20.00	10299.49	825.60	✓	
515	17081	0.00	20.00	10279.49	824.00	✓	
514	17080	0.00	20.00	10259.49	822.40	✓	
513	17079	0.00	20.00	10239.49	820.80	✓	
512	17078	0.00	20.00	10219.49	819.20	✓	
511	17077	0.00	20.00	10199.49	817.60	✓	
510	17076	0.00	20.00	10179.49	816.00	✓	
509	17075	0.00	20.00	10159.49	814.40	✓	
508	17074	0.00	20.00	10139.49	812.80	✓	
507	17073	0.00	20.00	10119.49	811.20	✓	
506	17072	0.00	20.00	10099.50	809.60	✓	
505	17071	0.00	20.00	10079.50	808.00	✓	
504	17070	0.00	20.00	10059.50	806.40	✓	
503	17069	0.00	20.00	10039.50	804.80	✓	
502	17068	0.00	20.00	10019.50	803.20	✓	
501	17067	0.00	20.00	9999.50	801.60	✓	
500	17066	0.00	20.00	9979.50	800.00	✓	
499	17065	0.00	20.00	9959.50	798.40	✓	
498	17064	0.00	20.00	9939.50	796.80	✓	
497	17063	0.00	20.00	9919.50	795.20	✓	
496	17062	0.00	20.00	9899.50	793.60	✓	
495	17061	0.00	20.00	9879.51	792.00	✓	
494	17060	0.00	20.00	9859.51	790.40	✓	
493	17059	0.00	20.00	9839.51	788.80	✓	

1 streams found.

Save Close ▶ Play Streams Help

SRTP Decrypt: Playing Decrypted Call



Libsrtp

- Implementation of the Secure Real-time Transport Protocol (SRTP)
- Can decipher SRTP packets

Libsrtp

- GitHub: github.com/cisco/libsrtp

The screenshot shows the GitHub repository page for `cisco/libsrtp`. The repository has 1,039 commits, 8 branches, 16 releases, and 48 contributors. The latest commit was 1447dfb, 13 days ago. The repository description is "Library for SRTP (Secure Realtime Transport Protocol)".

Key statistics:

- 1,039 commits
- 8 branches
- 16 releases
- 48 contributors

Recent activity:

Author	Commit Message	Date
pabuhler	Merge pull request #404 from pabuhler/add-extern-to-global-variables ...	Latest commit 1447dfb 13 days ago
crypto	Merge pull request #404 from pabuhler/add-extern-to-global-variables	13 days ago
doc	doc/Doxyfile.in: Remove rtp.h	11 months ago
include	Merge pull request #356 from thisisG/format_include_getopt_s_h	6 months ago
srtplib	Conform to clang-format in srtplib_get_session_keys	13 days ago
test	Ensure returned trailer length is sufficient	a month ago
.clang-format	clang-format aes_gcm_ossll.c	7 months ago

Libsrtp: Installation

- Cloning

```
root@PentesterAcademy:/work# git clone https://github.com/cisco/libsrtp.git
Cloning into 'libsrtp'...
remote: Counting objects: 6495, done.
remote: Total 6495 (delta 0), reused 0 (delta 0), pack-reused 6495
Receiving objects: 100% (6495/6495), 5.28 MiB | 126.00 KiB/s, done.
Resolving deltas: 100% (4442/4442), done.
root@PentesterAcademy:/work# cd libsrtp/
```

Libsrtp: Installation

- Configure

```
root@PentesterAcademy:/work/libsrtp# ./configure
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking how to run the C preprocessor... gcc -E
checking for ar... ar
checking the archiver (ar) interface... ar
checking for ranlib... ranlib
checking for a BSD-compatible install... /usr/bin/install -c
checking for a sed that does not truncate output... /bin/sed
checking for grep that handles long lines and -e... /bin/grep
```

Libsrtp: Installation

- Make

```
root@PentesterAcademy:/work/libsrtp# make
gcc -DHAVE_CONFIG_H -Icrypto/include -I./include -I./crypto/include -fPIC -Wall
all-loops -c srtp/srtp.c -o srtp/srtp.o
gcc -DHAVE_CONFIG_H -Icrypto/include -I./include -I./crypto/include -fPIC -Wall
all-loops -c srtp/ekt.c -o srtp/ekt.o
gcc -DHAVE_CONFIG_H -Icrypto/include -I./include -I./crypto/include -fPIC -Wall
all-loops -c crypto/cipher/cipher.c -o crypto/cipher/cipher.o
gcc -DHAVE_CONFIG_H -Icrypto/include -I./include -I./crypto/include -fPIC -Wall
all-loops -c crypto/cipher/null_cipher.c -o crypto/cipher/null_cipher.o
gcc -DHAVE_CONFIG_H -Icrypto/include -I./include -I./crypto/include -fPIC -Wall
all-loops -c crypto/cipher/aes_icm.c -o crypto/cipher/aes_icm.o
gcc -DHAVE_CONFIG_H -Icrypto/include -I./include -I./crypto/include -fPIC -Wall
all-loops -c crypto/cipher/aes.c -o crypto/cipher/aes.o
gcc -DHAVE_CONFIG_H -Icrypto/include -I./include -I./crypto/include -fPIC -Wall
all-loops -c crypto/hash/null_auth.c -o crypto/hash/null_auth.o
gcc -DHAVE_CONFIG_H -Icrypto/include -I./include -I./crypto/include -fPIC -Wall
all-loops -c crypto/hash/auth.c -o crypto/hash/auth.o
gcc -DHAVE_CONFIG_H -Icrypto/include -I./include -I./crypto/include -fPIC -Wall
all-loops -c crypto/hash/hmac.c -o crypto/hash/hmac.o
```

Libsrtp: Ready

```
root@PentesterAcademy:/work/libsrtp/test# ./rtp_decoder -h
Using libsrtp2 2.2.0-pre [0x2020000]
usage: ./rtp_decoder [-d <debug>]* [[-k][-b] <key> [-a][-e]]
or      ./rtp_decoder -l
where   -a use message authentication
        -e <key size> use encryption (use 128 or 256 for key size)
        -g Use AES-GCM mode (must be used with -e)
        -t <tag size> Tag size to use (in GCM mode use 8 or 16)
        -k <key> sets the srtp master key given in hexadecimal
        -b <key> sets the srtp master key given in base64
        -l list debug modules
        -f "<pcap filter>" to filter only the desired SRTP packets
        -d <debug> turn on debugging for module <debug>
        -s "<srtp-crypto-suite>" to set both key and tag size based
          on RFC4568-style crypto suite specification
```

Libsrtp: SRTP key

Normal_Call_two_parties.pcap

No. Time Source Destination Protocol Length Info

128	27.128753	192.168.20.132	192.168.20.130	SIP/SDP	278	Request: INVITE sip:2222@192.168.20.130
131	27.301506	192.168.20.130	192.168.20.1	SIP/SDP	1174	Request: INVITE sip:2222@192.168.20.1:60168;ob
173	29.293203	192.168.20.1	192.168.20.130	SIP/SDP	1101	Status: 200 OK
178	29.314263	192.168.20.130	192.168.20.132	SIP/SDP	1131	Status: 200 OK

Internet Protocol Version 4, Src: 192.168.20.1, Dst: 192.168.20.130
User Datagram Protocol, Src Port: 60168, Dst Port: 5060
Session Initiation Protocol (200)
Status-Line: SIP/2.0 200 OK
Message Header
Message Body
Session Description Protocol
Session Description Protocol Version (v): 0
Owner/Creator, Session Id (o): - 3730471310 3730471311 IN IP4 192.168.5.114
Session Name (s): pjmedia
Bandwidth Information (b): AS:84
Time Description, active time (t): 0 0
Session Attribute (a): X-nat:0
Media Description, name and address (m): audio 4000 RTP/SAVP 0 101
Connection Information (c): IN IP4 192.168.5.114
Bandwidth Information (b): TIAS:64000
Media Attribute (a): rtcp:4001 IN IP4 192.168.5.114
Media Attribute (a): sendrecv
Media Attribute (a): rtpmap:0 PCMU/8000
Media Attribute (a): rtpmap:101 telephone-event/8000
Media Attribute (a): fmtp:101 0-16
Media Attribute (a): ssrc:965767637 cname:66bf37b000942b74
Media Attribute (a): crypto:1 AES_CM_128_HMAC_SHA1_80 inline:2stvabBcXXf3HtaHCSSsB8wACeRBst9f7lwLqlzqE

Libsrtp: Copying SRTP key

Normal_Call_two_parties.pcap

No. Time Source Destination Protocol Length Info

128	27.128753	192.168.20.132	192.168.20.130	SIP/SDP	278	Request: INVITE sip:2222@192.168.20.130
131	27.301506	192.168.20.130	192.168.20.1	SIP/SDP	1174	Request: INVITE sip:2222@192.168.20.1:60168;ob
173	29.293203	192.168.20.1	192.168.20.130	SIP/SDP	1101	Status: 200 OK
178	29.314263	192.168.20.130	192.168.20.132	SIP/SDP	1131	Status: 200 OK

Internet Protocol Version 4, Src: 192.168.20.1, Dst: 192.168.20.130
User Datagram Protocol, Src Port: 60168, Dst Port: 5060
Session Initiation Protocol (200)
Status-Line: SIP/2.0 200 OK
Message Header
Message Body
Session Description Protocol
Session Description Protocol Version (v): 0
Owner/Creator, Session Id (o): - 3730471310 3730471311 IN IP4 192.168.5.114
Session Name (s): pjmedia
Bandwidth Information (b): AS:84
Time Description, active time (t): 0 0
Session Attribute (a): X-nat:0
Media Description, name and address (m): audio 4000 RTP/SAVP
Connection Information (c): IN IP4 192.168.5.114
Bandwidth Information (b): TIAS:64000
Media Attribute (a): rtcp:4001 IN IP4 192.168.5.114
Media Attribute (a): sendrecv
Media Attribute (a): rtpmap:0 PCMU/8000
Media Attribute (a): rtpmap:101 telephone-event/8000
Media Attribute (a): fmtp:101 0-16
Media Attribute (a): ssrc:965767637 cname:66bf37b000942b74
Media Attribute (a): crypto:1 AES_CM_128_HMAC_SHA1_80 inline:

Expand Subtrees Shift+Right
Expand All Ctrl+Right
Collapse All Ctrl+Left
Apply as Column
Apply as Filter
Prepare a Filter
Conversation Filter
Colorize with Filter
Follow

Copy Ctrl+Alt+Shift+V
Show Packet Bytes...
Export Packet Bytes... Ctrl+H
Wiki Protocol Page
Filter Field Reference
Protocol Preferences
Decode As...
Go to Linked Packet
Show Linked Packet in New Window

Libsrtp: Filtering for one sender

The screenshot shows the Wireshark interface with a list of network packets. A context menu is open over the fourth packet in the list, which has the source IP 192.168.20.130 and destination IP 192.168.20.132. The menu path is "Selected".

Context menu options visible:

- Expand Subtrees (Shift+Right)
- Expand All (Ctrl+Right)
- Collapse All (Ctrl+Left)
- Apply as Column
- Apply as Filter (highlighted)
- Prepare a Filter
- Conversation Filter
- Colorize with Filter
- Follow
- Copy
- Show Packet Bytes...
- Export Packet Bytes... (Ctrl+H)
- Wiki Protocol Page
- Filter Field Reference
- Protocol Preferences
- Decode As...
- Go to Linked Packet
- Show Linked Packet in New Window

Selected packet details:

- No. 178 Time 29.314263
- Source 192.168.20.130
- Destination 192.168.20.132
- Protocol SIP/SDP

Packet bytes and information pane:

- Internet Protocol Version 4, Src: 192.168.20.1, Dst: 192.168.20.130
 - 0100 = Version: 4
 - 0101 = Header Length: 20 bytes (5)
 - Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 - Total Length: 1087
 - Identification: 0x14d4 (5332)
 - Flags: 0x00
 - Fragment offset: 0
 - Time to live: 128
 - Protocol: UDP (17)
 - Header checksum: 0x7806 [validation disabled]
 - [Header checksum status: Unverified]
- Source: 192.168.20.1
- Destination: 192.168.20.130
- [Source GeoIP: Unknown]
- [Destination GeoIP: Unknown]
- User Datagram Protocol, Src Port: 60168, Dst Port: 5060
- Session Initiation Protocol (200)
 - Status-Line: SIP/2.0 200 OK

Libsrtp: Filtering single RTP stream

Normal_Call_two_parties.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.src == 192.168.20.1 && rtp

No.	Time	Source	Destination	Protocol	Length	Ta	Info
177	29.311833	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25650, Time=160, Mark
189	29.332471	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25651, Time=320
193	29.352961	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25652, Time=480
197	29.372665	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25653, Time=640
204	29.393539	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25654, Time=800
208	29.413260	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25655, Time=960
212	29.434077	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25656, Time=1120
216	29.453993	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25657, Time=1280
220	29.474710	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25658, Time=1440
225	29.494627	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25659, Time=1600
230	29.515344	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25660, Time=1760
234	29.535085	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25661, Time=1920
238	29.555804	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25662, Time=2080
242	29.575801	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25663, Time=2240
247	29.596513	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25664, Time=2400
251	29.616324	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25665, Time=2560
255	29.636923	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25666, Time=2720
260	29.657564	192.168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25667, Time=2880

Frame 177: 224 bytes on wire (1792 bits), 224 bytes captured (1792 bits)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_ff:65:9b (00:0c:29:ff:65:9b)
Internet Protocol Version 4, Src: 192.168.20.1, Dst: 192.168.20.130
User Datagram Protocol, Src Port: 4000, Dst Port: 16450
Real-Time Transport Protocol

Libsrtp: Exporting filtered traffic

Normal_Call_two_parties.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Open Ctrl+O

Open Recent

Merge...

Import from Hex Dump...

Close Ctrl+W

Save Ctrl+S

Save As... Ctrl+Shift+S

File Set

Export Specified Packets... **→**

Export Packet Dissections

Export Packet Bytes... Ctrl+H

Export PDUs to File...

Export SSL Session Keys...

Export Objects

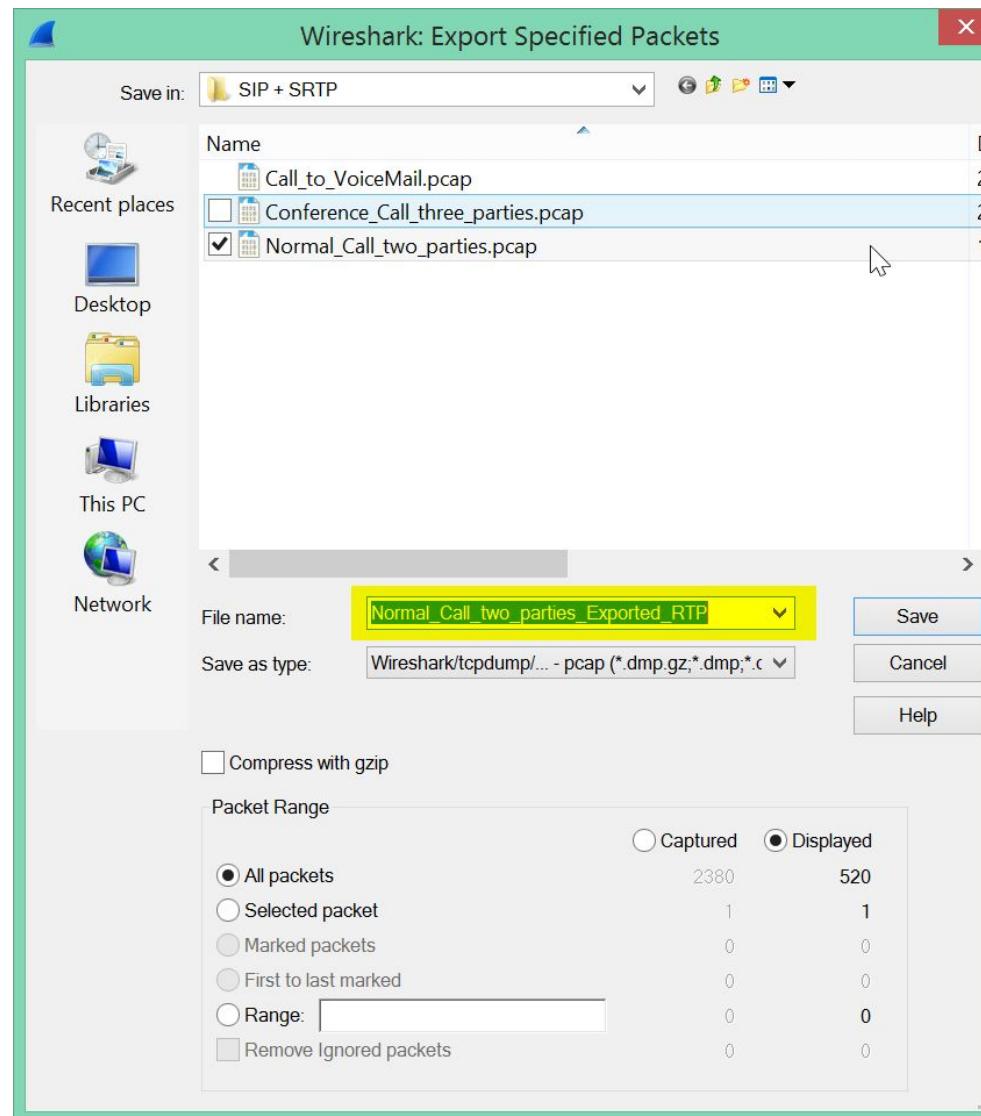
Print... Ctrl+P

Quit Ctrl+Q

	Destination	Protocol	Length	Ta	Info
168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25650, Time=160, Mark
168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25651, Time=320
168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25652, Time=480
168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25653, Time=640
168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25654, Time=800
168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25655, Time=960
168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25656, Time=1120
168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25657, Time=1280
168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25658, Time=1440
168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25659, Time=1600
168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25660, Time=1760
168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25661, Time=1920
168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25662, Time=2080
168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25663, Time=2240
168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25664, Time=2400
168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25665, Time=2560
168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25666, Time=2720
168.20.1	192.168.20.130	SRTP	224		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25667, Time=2880

Frame 177: 224 bytes on wire (1792 bits), 224 bytes captured (1792 bits)
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_ff:65:9b (00:0c:29:ff:65:9b)
Internet Protocol Version 4, Src: 192.168.20.1, Dst: 192.168.20.130
User Datagram Protocol, Src Port: 4000, Dst Port: 16450
Real-Time Transport Protocol

Libsrtp: Saving exported traffic



Libsrtp: Command

- `./rtp_decoder -a -t 10 -e 128 -b 2stvabBcXXf3HtaHCSsB8WACeRBst9f7lwLqlzqE * < ./Normal_Call_two_parties_Exported_RTP.pcap`
- `-a` Use message authentication
- `-t` Authentication tag size (80 bits so 10 bytes)
- `-e` Length of encryption key. In our case, `AES_CM_128_HMAC_SHA1_80` is cipher.
Hence, 128 bit key is used.
- `-b` SRTP key in ASCII format

Libsrtp: Command output

```
root@PentesterAcademy:/work/libsrtp/test# ./rtp_decoder -a -t 10 -e 128 -b 2stvabBcXXf3HtaHCSSsB8WACeRBst9f7lwLqlzqE * < ../../Normal_Call_two_parties_Exported_RTP.pcap
Using libsrtp2 2.2.0-pre [0x2020000]
security services: confidentiality message authentication
setting tag len 10
set master key/salt to dacb6f69b05c5d77f71ed687092b01f1/600279106cb7d7fb9702ea973a84
Starting decoder
00:00.000000
0000 80 80 64 32 00 00 00 a0 39 90 71 d5 ff ff ff ff
0010 7e ff 7e ff 7e 7e ff 7e fe 7d fe 7d fd 7d fe 7e
0020 7d fb 78 f6 73 f2 71 f2 76 fd f4 66 d5 29 1b 1e
0030 1e 22 24 27 29 2c 33 3a 3f 46 5a ef d5 c9 bb af
0040 ab a9 a8 a6 a3 9f 9e 9e 9d 9d 9c 9c 9b 9b 9a 99
0050 98 97 97 97 97 97 97 97 98 99 9a 9c 9d 9e 9e 9f
0060 a2 a6 a8 aa ac af b9 c2 ce df 60 4a 40 3b 37 35
0070 33 31 2e 2c 29 27 27 27 27 29 29 28 27 27 2a
0080 2c 2d 2d 2d 2e 2f 2e 2c 2b 2b 2c 2d 2e 2e 2e 2f
0090 2f 2f 2f 32 36 3c 47 4e 55 56 56 67 ed e2 e4
00a0 d5 c3 bb b7 b6 b7 b6 b3 ae ad ac aa
00:00.020638
0000 80 00 64 33 00 00 01 40 39 90 71 d5 a9 aa aa aa
0010 a8 a7 a7 a6 a7 a8 a9 a8 a6 a5 a5 a4 a3 a2 a3 a3
0020 a3 a2 a2 a2 a4 a4 a5 a6 a7 a8 a8 a9 aa ab ac ad
0030 af b0 b4 b8 ba bd c0 c9 d7 ec 69 57 49 40 3d 3d
0040 3c 39 34 30 2e 2d 2d 2d 2c 2c 2c 2d 2c 2c 2b 2a
0050 2a 29 29 29 2a 29 28 27 27 27 27 27 27 2a 2c 2e
0060 2e 2f 32 36 3a 3c 3d 3d 40 42 42 3e 3a 35 2f 2b
0070 2a 29 29 2b 2f 3e fe c0 b2 a9 a1 9d 9b 9a 9b 9c
0080 9e a5 ae b9 d0 4c 37 2e 2b 2c 2d 2d 2a 28 25 25
0090 27 2d 3b 5d c4 ae a1 9a 96 94 93 93 95 99 9f aa
00a0 b9 d8 45 31 28 22 1e 1e 1d 1d 1e 1f
00:00.041128
```

Libsrtplib: text2pcap help

```
root@PentesterAcademy:~# text2pcap
Must specify input and output filename
[  
Usage: text2pcap [options] <infile> <outfile>
where <infile> specifies input filename (use - for standard input)
      <outfile> specifies output filename (use - for standard output)

Input:
  -o hex|oct|dec          parse offsets as (h)ex, (o)ctal or (d)eclimal;
                          default is hex.
  -t <timefmt>           treat the text before the packet as a date/time code;
                          the specified argument is a format string of the sort
                          supported by strftime.
                          Example: The time "10:15:14.5476" has the format code
                          "%H:%M:%S."
                          NOTE: The subsecond component delimiter, '.', must be
                          given, but no pattern is required; the remaining
                          number is assumed to be fractions of a second.
                          NOTE: Date/time fields from the current date/time are
                          used as the default for unspecified fields.
  -D                      the text before the packet starts with an I or an O,
                          indicating that the packet is inbound or outbound.
                          This is only stored if the output format is PCAP-NG.
  -a                      enable ASCII text dump identification.
                          The start of the ASCII text dump can be identified
                          and excluded from the packet data, even if it looks
                          like a HEX dump.
                          NOTE: Do not enable it if the input file does not
                          contain the ASCII text dump.
```

Libsrtp: text2pcap

- `text2pcap -t "%M:%S." -u 10000,10000 -- > ./Normal_Call_two_parties_Decrypted.pcap`
- `-t` Treat the text before the packet as a date/time code
- `%M:%S` Time format
- `-u` Prepend dummy UDP header with specified source and destination ports

Libsrtp: Decrypting RTP traffic

```
root@PentesterAcademy:/work/libsrtp/test# ./rtp_decoder -a -t 10 -e 128 -b 2stvabBcXXf3HtaHCSSsB8WACeRBst9f7lwLqlzqE * < ./Normal_Call_two_parties_Exported_RTP.pcap | text2pcap -t "%M:%S." -u 10000,10000 - - > ./Normal_Call_two_parties_Decrypted.pcap
Input from: Standard input
Output to: Standard output
Output format: PCAP
Generate dummy Ethernet header: Protocol: 0x800
Generate dummy IP header: Protocol: 17
Generate dummy UDP header: Source port: 10000. Dest port: 10000
Using libsrtp2 2.2.0-pre [0x2020000]
security services: confidentiality message authentication
setting tag len 10
set master key/salt to dacb6f69b05c5d77f71ed687092b01f1/600279106cb7d7fb9702ea973a84
Starting decoder
Wrote packet of 214 bytes.
```

Libsrtp: Decrypted traffic

Normal_Call_two_parties_Decrypted.pcap

The screenshot shows a Wireshark capture window titled "Normal_Call_two_parties_Decrypted.pcap". The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with various icons. A search bar at the top says "Apply a display filter ... <Ctrl-/>". The main pane displays a table of network traffic. The columns are: No., Time, Source, Destination, Protocol, Length, Ta, and Info. The "Protocol" column is highlighted with a yellow background. The "Info" column shows entries like "10000 → 10000 Len=172". The table contains 18 rows of data. At the bottom of the table, there is a detailed description of the first frame.

No.	Time	Source	Destination	Protocol	Length	Ta	Info
1	0.000000	10.1.1.1	10.2.2.2	UDP	214		10000 → 10000 Len=172
2	0.020638	10.1.1.1	10.2.2.2	UDP	214		10000 → 10000 Len=172
3	0.041128	10.1.1.1	10.2.2.2	UDP	214		10000 → 10000 Len=172
4	0.060832	10.1.1.1	10.2.2.2	UDP	214		10000 → 10000 Len=172
5	0.081706	10.1.1.1	10.2.2.2	UDP	214		10000 → 10000 Len=172
6	0.101427	10.1.1.1	10.2.2.2	UDP	214		10000 → 10000 Len=172
7	0.122244	10.1.1.1	10.2.2.2	UDP	214		10000 → 10000 Len=172
8	0.142160	10.1.1.1	10.2.2.2	UDP	214		10000 → 10000 Len=172
9	0.162877	10.1.1.1	10.2.2.2	UDP	214		10000 → 10000 Len=172
10	0.182794	10.1.1.1	10.2.2.2	UDP	214		10000 → 10000 Len=172
11	0.203511	10.1.1.1	10.2.2.2	UDP	214		10000 → 10000 Len=172
12	0.223252	10.1.1.1	10.2.2.2	UDP	214		10000 → 10000 Len=172
13	0.243971	10.1.1.1	10.2.2.2	UDP	214		10000 → 10000 Len=172
14	0.263968	10.1.1.1	10.2.2.2	UDP	214		10000 → 10000 Len=172
15	0.284680	10.1.1.1	10.2.2.2	UDP	214		10000 → 10000 Len=172
16	0.304491	10.1.1.1	10.2.2.2	UDP	214		10000 → 10000 Len=172
17	0.325090	10.1.1.1	10.2.2.2	UDP	214		10000 → 10000 Len=172
18	0.345731	10.1.1.1	10.2.2.2	UDP	214		10000 → 10000 Len=172

Frame 1: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits)
Ethernet II, Src: 0a:01:01:01:01:01 (0a:01:01:01:01:01), Dst: 0a:02:02:02:02:02 (0a:02:02:02:02:02)
Internet Protocol Version 4, Src: 10.1.1.1, Dst: 10.2.2.2
User Datagram Protocol, Src Port: 10000, Dst Port: 10000
Data (172 bytes)

Libsrtp: Decode as

Normal_Call_two_parties_Decrypted.pcap

No. Time Source Destination Protocol Length Ta Info

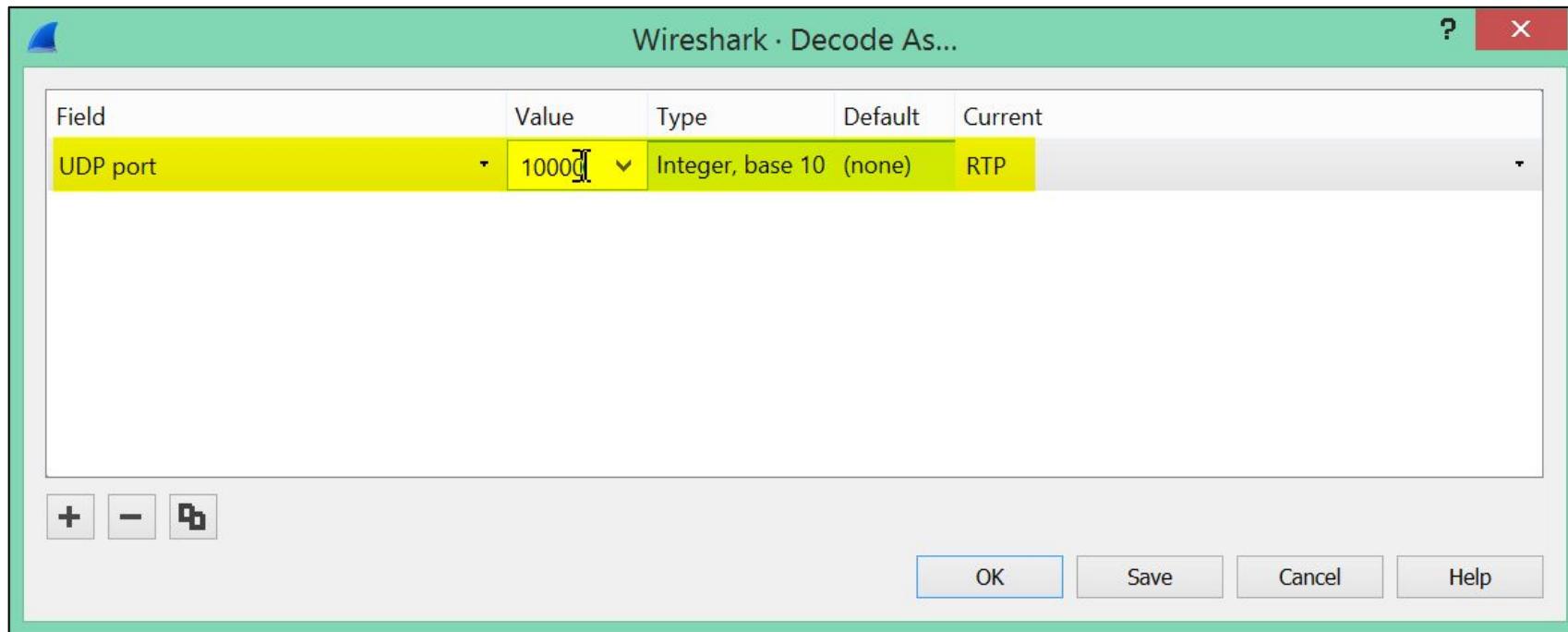
1	0.000000	10.1.1.1	10.2.2.2	UDP	214	10000 → 10000	Len=172	
2	0.020638	10.1.1.1	10.2.2.2	UDP	214	10000 → 10000	Len=172	
3	0.041128	10.1.1.1	10.2.2.2	UDP	214	10000 → 10000	Len=172	
4	0.060832	10.1.1.1	10.2.2.2			Mark/Unmark Packet	Ctrl+M	10000 Len=172
5	0.081706	10.1.1.1	10.2.2.2			Ignore/Unignore Packet	Ctrl+D	10000 Len=172
6	0.101427	10.1.1.1	10.2.2.2			Set/Unset Time Reference	Ctrl+T	10000 Len=172
7	0.122244	10.1.1.1	10.2.2.2			Time Shift...	Ctrl+Shift+T	10000 Len=172
8	0.142160	10.1.1.1	10.2.2.2			Packet Comment...	Ctrl+Alt+C	10000 Len=172
9	0.162877	10.1.1.1	10.2.2.2			Edit Resolved Name		10000 Len=172
10	0.182794	10.1.1.1	10.2.2.2					10000 Len=172
11	0.203511	10.1.1.1	10.2.2.2					10000 Len=172
12	0.223252	10.1.1.1	10.2.2.2			Apply as Filter		10000 Len=172
13	0.243971	10.1.1.1	10.2.2.2			Prepare a Filter		10000 Len=172
14	0.263968	10.1.1.1	10.2.2.2			Conversation Filter		10000 Len=172
15	0.284680	10.1.1.1	10.2.2.2			Colorize Conversation		10000 Len=172
16	0.304491	10.1.1.1	10.2.2.2			SCTP		10000 Len=172
17	0.325090	10.1.1.1	10.2.2.2			Follow		10000 Len=172
18	0.345731	10.1.1.1	10.2.2.2					10000 Len=172

Frame 3: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface br0
Ethernet II, Src: 0a:01:01:01:01:01 (0a:01:01:01:01:01), Dst: 10.2.2.2 (0a:01:01:01:01:02)
Internet Protocol Version 4, Src: 10.1.1.1, Dst: 10.2.2.2
User Datagram Protocol, Src Port: 10000, Dst Port: 10000
Data (172 bytes)

Decoding options for frame 3:

- Mark/Unmark Packet (Ctrl+M)
- Ignore/Unignore Packet (Ctrl+D)
- Set/Unset Time Reference (Ctrl+T)
- Time Shift... (Ctrl+Shift+T)
- Packet Comment... (Ctrl+Alt+C)
- Edit Resolved Name
- Apply as Filter
- Prepare a Filter
- Conversation Filter
- Colorize Conversation
- SCTP
- Follow
- Copy
- Protocol Preferences
- Decode As... (highlighted)
- Show Packet in New Window

Libsrtp: Decode as RTP



Libsrtp: Decrypted RTP traffic

Normal_Call_two_parties_Decrypted.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression... torrent cleanup_own_s

No.	Time	Source	Destination	Protocol	Length	Ta	Info
1	0.000000	10.1.1.1	10.2.2.2	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25650, Time=160, Mark
2	0.020638	10.1.1.1	10.2.2.2	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25651, Time=320
3	0.041128	10.1.1.1	10.2.2.2	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25652, Time=480
4	0.060832	10.1.1.1	10.2.2.2	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25653, Time=640
5	0.081706	10.1.1.1	10.2.2.2	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25654, Time=800
6	0.101427	10.1.1.1	10.2.2.2	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25655, Time=960
7	0.122244	10.1.1.1	10.2.2.2	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25656, Time=1120
8	0.142160	10.1.1.1	10.2.2.2	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25657, Time=1280
9	0.162877	10.1.1.1	10.2.2.2	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25658, Time=1440
10	0.182794	10.1.1.1	10.2.2.2	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25659, Time=1600
11	0.203511	10.1.1.1	10.2.2.2	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25660, Time=1760
12	0.223252	10.1.1.1	10.2.2.2	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25661, Time=1920
13	0.243971	10.1.1.1	10.2.2.2	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25662, Time=2080
14	0.263968	10.1.1.1	10.2.2.2	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25663, Time=2240
15	0.284680	10.1.1.1	10.2.2.2	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25664, Time=2400
16	0.304491	10.1.1.1	10.2.2.2	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25665, Time=2560
17	0.325090	10.1.1.1	10.2.2.2	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25666, Time=2720
18	0.345731	10.1.1.1	10.2.2.2	RTP	214		PT=ITU-T G.711 PCMU, SSRC=0x399071D5, Seq=25667, Time=2880

Frame 3: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits)
Ethernet II, Src: 0a:01:01:01:01:01 (0a:01:01:01:01:01), Dst: 0a:02:02:02:02:02 (0a:02:02:02:02:02)
Internet Protocol Version 4, Src: 10.1.1.1, Dst: 10.2.2.2
User Datagram Protocol, Src Port: 10000, Dst Port: 10000
Real-Time Transport Protocol

Libsrtplib: Analysing RTP Streams

Screenshot of Wireshark showing RTP analysis for a call between two parties.

The screenshot shows the Wireshark interface with the file "Normal_Call_two_parties_Decrypted.pcap" loaded. The "Telephony" tab is selected in the menu bar. A context menu is open over an RTP frame, specifically over the "RTP" entry in the VoIP Calls submenu. The context menu options are "RTP Streams" and "Stream Analysis".

The main pane displays a list of RTP frames. Each frame is a 214-byte payload containing G.711 PCM audio data. The frames are timestamped from 0.000000 to 0.345731. The source IP address for most frames is 10.1.1.1, and the destination IP address is 10.2.2.2. The port number is consistently 10000.

The bottom pane shows the detailed description and bytes for the selected frame (Frame 3), which is an Ethernet II frame with an Internet Protocol Version 4 header and a User Datagram Protocol header.

No.	Time	Source
1	0.000000	10.1.1.1
2	0.020638	10.1.1.1
3	0.041128	10.1.1.1
4	0.060832	10.1.1.1
5	0.081706	10.1.1.1
6	0.101427	10.1.1.1
7	0.122244	10.1.1.1
8	0.142160	10.1.1.1
9	0.162877	10.1.1.1
10	0.182794	10.1.1.1
11	0.203511	10.1.1.1
12	0.223252	10.1.1.1
13	0.243971	10.1.1.1
14	0.263968	10.1.1.1
15	0.284680	10.1.1.1
16	0.304491	10.1.1.1
17	0.325090	10.1.1.1
18	0.345731	10.1.1.1

Frame 3: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits)
Ethernet II, Src: 0a:01:01:01:01:01 (0a:01:01:01:01:01), Dst: 0a:02:02:02:02:02 (0a:02:02:02:02:02)
Internet Protocol Version 4, Src: 10.1.1.1, Dst: 10.2.2.2
User Datagram Protocol, Src Port: 10000, Dst Port: 10000
Real-Time Transport Protocol

©PentesterAcademy.com

Libsrtp: Analysing RTP Streams

Wireshark · RTP Stream Analysis · Normal_Call_two_parties_Decrypted

10.1.1.1:10000 ↔ 10.2.2.2:10000

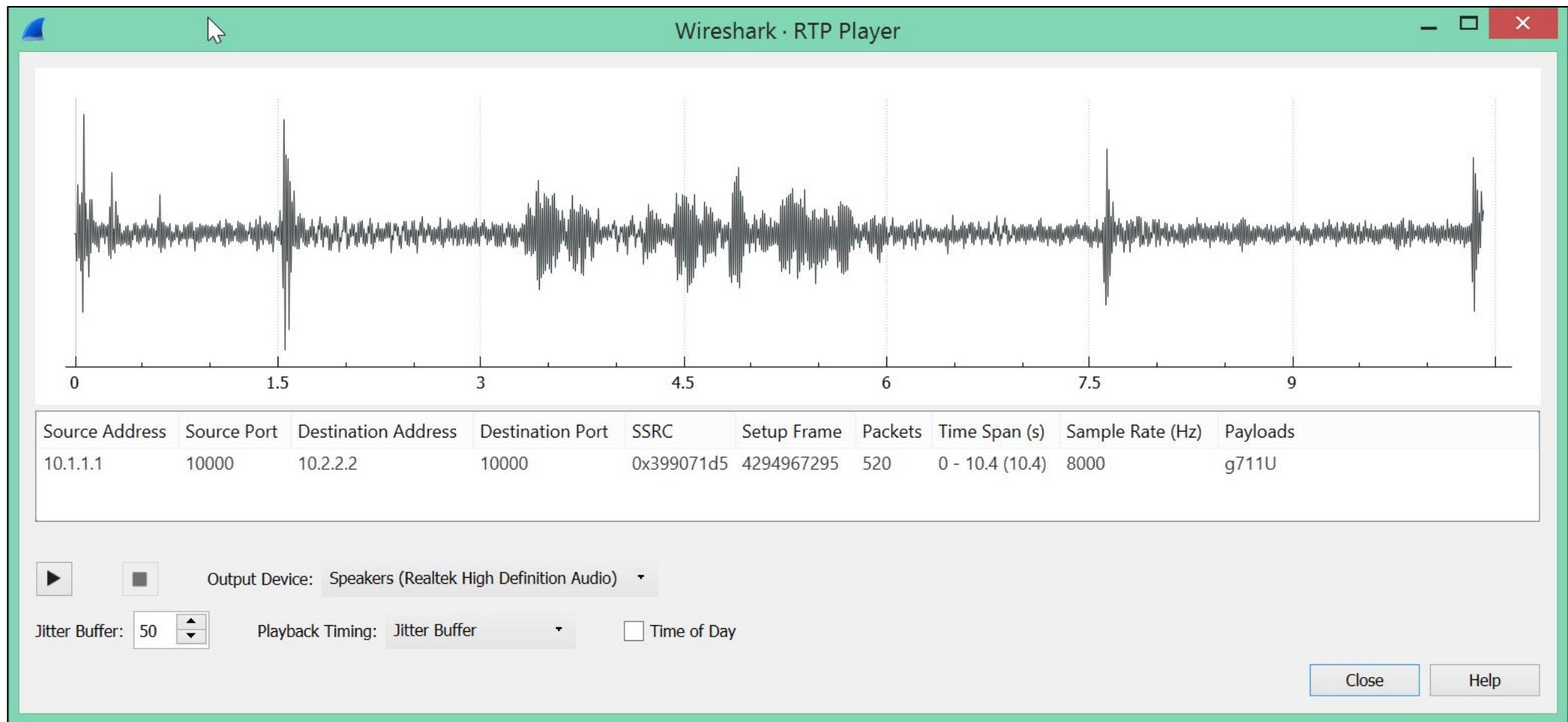
Forward

Packet	Sequence	Delta (ms)	Jitter (ms)	Skew	Bandwidth	Marker	Status
520	26169	19.59	0.82	-1.83	81.60	✓	
519	26168	20.50	0.84	-2.24	81.60	✓	
518	26167	20.60	0.87	-1.74	81.60	✓	
517	26166	20.50	0.89	-1.14	81.60	✓	
516	26165	19.67	0.91	-0.63	81.60	✓	
515	26164	20.45	0.95	-0.96	81.60	✓	
514	26163	20.71	0.98	-0.50	81.60	✓	
513	26162	20.51	1.00	0.21	81.60	✓	
512	26161	19.25	1.03	0.71	81.60	✓	
511	26160	20.34	1.05	-0.04	81.60	✓	
510	26159	20.64	1.10	0.31	81.60	✓	
509	26158	10.07	1.13	0.95	81.60	✓	
508	26157	20.54	0.54	-8.99	80.00	✓	
Reverse							
507	26156	20.45	0.54	-8.45	80.00	✓	
506	26155	20.31	0.55	-8.00	80.00	✓	
SSRC	0x00000000						
Max Delta	0.00 ms @ 0						
Max Jitter	0.00 ms						
Mean Jitter	0.00 ms						
Max Skew	0.00 ms						
RTP Packets	0						
Expected	1						
Lost	1 (100.00 %)						
Seq Errs	0						
Start at	0.000000 s @ 0						
Duration	0.00 s						
Clock Drift	0 ms						
Freq Drift	1 Hz (0.00 %)						

1 streams found.

Save Close ► Play Streams Help

Libsrtp: Playing decrypted call



Other Important Parts?

- DTMF
- Messages (SMS)
- Exporting Call

RTP DTMF

NetworkMiner analysis of DTMF Lab 1 SIP+RTP traffic.

The screenshot shows the NetworkMiner interface with the following details:

- File Menu:** File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help.
- Toolbar:** Includes icons for File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help, and various search and selection tools.
- Search Bar:** Contains the filter "rtp".
- Table Headers:** No., Time, Source, Destination, Protocol, Length, Ta, Info.
- Data Rows:** The table lists 10 network frames. Frames 2594, 2595, 2596, 2598, 2599, 2600, 2601, 2602, and 2603 have their "Protocol" field set to "RTP EVENT". Frame 2597 has its "Protocol" field set to "RTP".
 - Frame 2594: 192.168.20.130 to 192.168.20.1, RTP, PT=ITU-T G.711 PCMU, SSRC=0x4BDB6E8A, Seq=21265, Time=97280.
 - Frame 2595: 192.168.20.1 to 192.168.20.130, RTP, PT=ITU-T G.711 PCMU, SSRC=0x294823, Seq=12503, Time=97920.
 - Frame 2596: 192.168.20.130 to 192.168.20.136, RTP, PT=ITU-T G.711 PCMU, SSRC=0x71781F5A, Seq=1568, Time=97920.
 - Frame 2597: 192.168.20.136 to 192.168.20.130, RTP EVENT, Payload type=RTP Event, DTMF One 1.
 - Frame 2598: 192.168.20.130 to 192.168.20.1, RTP EVENT, Payload type=RTP Event, DTMF One 1.
 - Frame 2599: 192.168.20.130 to 192.168.20.1, RTP EVENT, Payload type=RTP Event, DTMF One 1.
 - Frame 2600: 192.168.20.1 to 192.168.20.130, RTP, PT=ITU-T G.711 PCMU, SSRC=0x294823, Seq=12504, Time=98080.
 - Frame 2601: 192.168.20.130 to 192.168.20.1, RTP EVENT, Payload type=RTP Event, DTMF One 1.
 - Frame 2602: 192.168.20.130 to 192.168.20.136, RTP, PT=ITU-T G.711 PCMU, SSRC=0x71781F5A, Seq=1569, Time=98080.
 - Frame 2603: 192.168.20.136 to 192.168.20.130, RTP EVENT, Payload type=RTP Event, DTMF One 1.
- Details Panel:** Shows expanded information for the selected frame (Frame 2597).
 - Frame 2597: 58 bytes on wire (464 bits), 58 bytes captured (464 bits).
 - Ethernet II, Src: Vmware_23:37:1f (00:50:56:23:37:1f), Dst: Vmware_ab:b1:84 (00:0c:29:ab:b1:84)
 - Internet Protocol Version 4, Src: 192.168.20.136, Dst: 192.168.20.130
 - User Datagram Protocol, Src Port: 4000, Dst Port: 16290
 - Real-Time Transport Protocol
 - RFC 2833 RTP Event
 - Event ID: DTMF One 1 (1)
 - 0.... = End of Event: False
 - .0.. = Reserved: False
 - ..00 1010 = Volume: 10
 - Event Duration: 160

SIP Message

SIP_Message_SIP+RTP.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No. Time Source Destination Protocol Length Ta Info

60	33.429572	192.168.20.130	192.168.20.136	SIP	543	Status: 202 Accepted
61	33.429573	192.168.20.130	192.168.20.1	SIP	513	Request: MESSAGE sip:2222@192.168.20.1:63825;ob (text/plain)
62	33.430944	192.168.20.1	192.168.20.130	SIP	348	Status: 200 OK

Frame 61: 513 bytes on wire (4104 bits), 513 bytes captured (4104 bits)
Ethernet II, Src: Vmware_ab:b1:84 (00:0c:29:ab:b1:84), Dst: Vmware_c0:00:08 (00:50:56:c0:00:08)
Internet Protocol Version 4, Src: 192.168.20.130, Dst: 192.168.20.1
User Datagram Protocol, Src Port: 5160, Dst Port: 63825
Session Initiation Protocol (MESSAGE)
 Request-Line: MESSAGE sip:2222@192.168.20.1:63825;ob SIP/2.0
 Message Header
 Via: SIP/2.0/UDP 192.168.20.130:5160;branch=z9hG4bK5a87574e
 Max-Forwards: 70
 From: "Unknown" <sip:1111@192.168.20.130:5160>;tag=as008f816f
 To: <sip:2222@192.168.20.1:63825;ob>
 Contact: <sip:1111@192.168.20.130:5160>
 Call-ID: 073e1f452da9a1e17dbf255754c503a9@[::1]:5160
 CSeq: 102 MESSAGE
 User-Agent: FPBX-13.0.194.2(13.12.1)
 Content-Type: text/plain; charset=UTF-8
 Content-Length: 29
 Message Body
 Line-based text data: text/plain
 Hello world to sip messaging!

PCAP2WAV: Online service

The screenshot shows a web browser window with the title bar "PCAP2WAV RTP2WAV" and the URL "pcap2wav.xplico.org". The page content is as follows:

PCAP2WAV converts RTP streams to WAV files

Codecs supported: G711ulaw, G711alaw, G722, G729, G723, G726 and RTAudio (x-msrta: Real Time Audio).

PCAP2WAV is an [Xplico](#) customization and it runs in [Linux](#).

Try it now, drag & drop here the PCAP file.

This session is visible only from your IP (182.48.243.162).

Demo rules:

- Only network files (**CAP, PCAP**) are allowed.
- The maximum file size for uploads is **5 MB**.
- Uploaded files will be deleted automatically at **00:00 GMT**.
- You can **drag & drop** files from your desktop on this webpage with Google Chrome, Mozilla Firefox and Apple Safari.

Buttons at the bottom left: **+ Add files...** (yellow background), **Delete** (red background), and a small gray square.

PCAP2WAV: Uploading PCAP and Downloading Wav

The screenshot shows a web browser window for the URL pcap2wav.xplico.org. The title bar reads "PCAP2WAV RTP2WAV". The page content is as follows:

PCAP2WAV converts RTP streams to WAV files

Codecs supported: G711ulaw, G711alaw, G722, G729, G723, G726 and **RTAudio (x-msrta: Real Time Audio)**.
PCAP2WAV is an [Xplico](#) customization and it runs in [Linux](#).
Try it now, drag & drop here the **PCAP file**.
This session is visible only from your IP ([182.48.243.162](#)).

Demo rules:

- Only network files (**CAP, PCAP**) are allowed.
- The maximum file size for uploads is **5 MB**.
- Uploaded files will be deleted automatically at **00:00 GMT**.
- You can **drag & drop** files from your desktop on this webpage with Google Chrome, Mozilla Firefox and Apple Safari.

WAV Files:

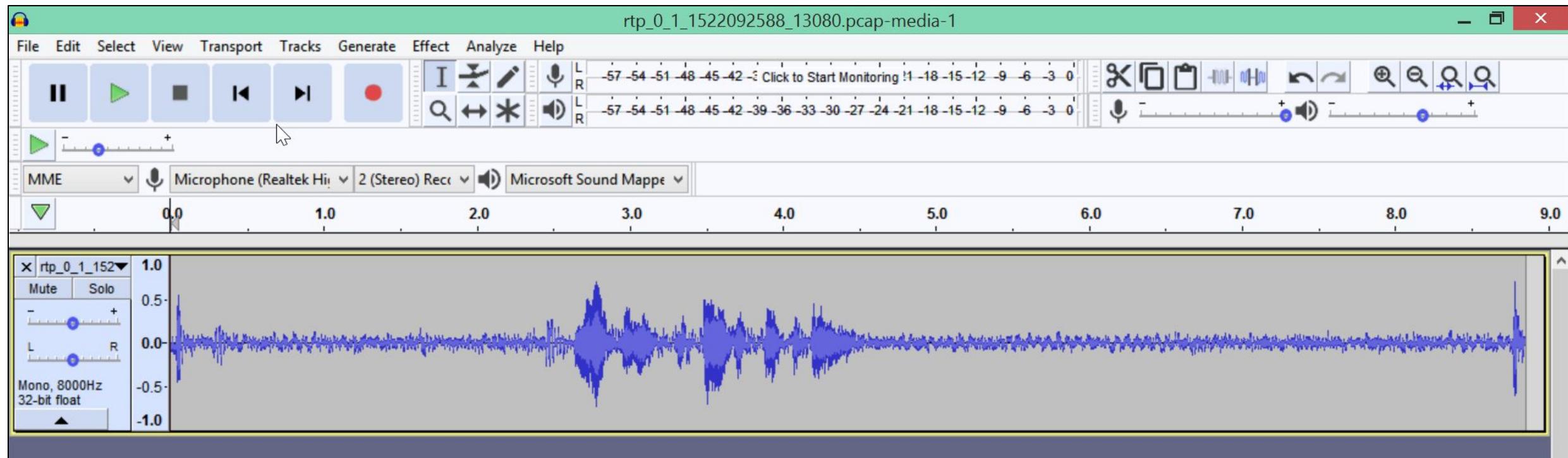
File Name	Size	Action
rtp_0_1_1522092588_13080.pcap-media-1.wav	70920	Delete
rtp_0_2_1522092588_13080.pcap-media-1.wav	70280	Delete

File Upload Area:

+ Add files... [Delete](#)

SIP+RTP_call_trace_from_caller_to_PBX.pcap 226.76 KB [Delete](#)

PCAP2WAV: Wav in audacity



PCAP2WAV: Offline script

- Bash script to extract the audio from VoIP calls
- Outputs .wav file
- Uses tshark and sox
- GitHub: <https://gist.github.com/avimar/d2e9d05e082ce273962d742eb9acac16>

PCAP2WAV: Help

```
root@PentesterAcademy:/work/pcap2wav# ./pcap2wav.sh -h
```

pcap2wav is a simple utility to make it easier to extract the audio from a pcap

Dependencies:

```
apt-get install -y tshark sox  
yum install wireshark sox
```

Usage:

```
pcap2wav [opts] filename.pcap [target filename]
```

Script attempts to create a few files: a .<codec> file and a .wav file for each RTP stream

It requires Tshark to be installed on the system. If a codec other than PCMA or PCMU is used then the script will attempt to use fs_cli to decode and create a wav.

Supported codecs:

PCMU (G711 ulaw)

PCMA (G711 Alaw)

GSM

G722 (requires fs_encode)

G729 (requires fs_encode with mod_com_g729)

Supported options:

- z Perform "clean and zip" - After converting to wav files the program will "clean up" by putting the wav files into a .tgz file and then removing the .wav and .<codec> files from the disk.

PCAP2WAV: Installing tshark and sox

```
root@PentesterAcademy:/work# apt-get install -y tshark sox
Reading package lists... Done
Building dependency tree
Reading state information... Done
tshark is already the newest version (2.4.4-1).
The following additional packages will be installed:
  libsox-fmt-alsa libsox-fmt-base libsox3
Suggested packages:
  libsox-fmt-all
The following NEW packages will be installed:
  libsox-fmt-alsa libsox-fmt-base libsox3 sox
0 upgraded, 4 newly installed, 0 to remove and 1826 not upgraded.
Need to get 530 kB of archives.
After this operation, 1,292 kB of additional disk space will be used.
Get:1 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 libsox3 amd64 14.4.2-3 [264 kB]
Get:2 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 libsox-fmt-alsa amd64 14.4.2-3 [51.3 kB]
Get:3 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 libsox-fmt-base amd64 14.4.2-3 [72.8 kB]
Get:4 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 sox amd64 14.4.2-3 [142 kB]
Fetched 530 kB in 6s (84.7 kB/s)
Selecting previously unselected package libsox3:amd64.
(Reading database ... 336924 files and directories currently installed.)
Preparing to unpack .../libsox3_14.4.2-3_amd64.deb ...
Unpacking libsox3:amd64 (14.4.2-3) ...
```

PCAP2WAV: Running the tool

```
root@PentesterAcademy:/work/pcap2wav# ./pcap2wav.sh SIP+RTP_call_trace_from_caller_to_PBX.pcap ./output_call.wav
Found SIP+RTP_call_trace_from_caller_to_PBX.pcap, working...
Using ./output_call.wav
Checking SIP+RTP_call_trace_from_caller_to_PBX.pcap for RTP streams...
Running as user "root" and group "root". This could be dangerous.
tshark: Lua: Error during loading:
[string "/usr/share/wireshark/init.lua"]:44: dofile has been disabled due to running Wireshark as superuser. See https://wiki.wireshark.org/CaptureSetup/CapturePrivileges for help in running Wireshark as an unprivileged user.
Running as user "root" and group "root". This could be dangerous.
tshark: Lua: Error during loading:
[string "/usr/share/wireshark/init.lua"]:44: dofile has been disabled due to running Wireshark as superuser. See https://wiki.wireshark.org/CaptureSetup/CapturePrivileges for help in running Wireshark as an unprivileged user.
Target files to create:
./output_call.wav_1.PCMU and ./output_call.wav_1.wav
./output_call.wav_2.PCMU and ./output_call.wav_2.wav

Stream 1 ssrc / port: 0x0fbb0c8d / 13080
Stream 2 ssrc / port: 0x4fce51a / 4004

Extracting payloads 1 from 0x0fbb0c8d...
Extracting payloads 2 from 0x4fce51a...
Combining 2 streams into a single wav file for convenience
No clean option specified - leaving <codec> and .wav files on system.
```

```
root@PentesterAcademy:/work/pcap2wav# ls -l
total 22
-rwxr-xr-x 1 root root 5927 Mar 27 01:18 pcap2wav.sh
-rw----- 1 root root 226760 Mar 19 17:29 SIP+RTP_call_trace_from_caller_to_PBX.pcap
```

PCAP2WAV: Directory contents

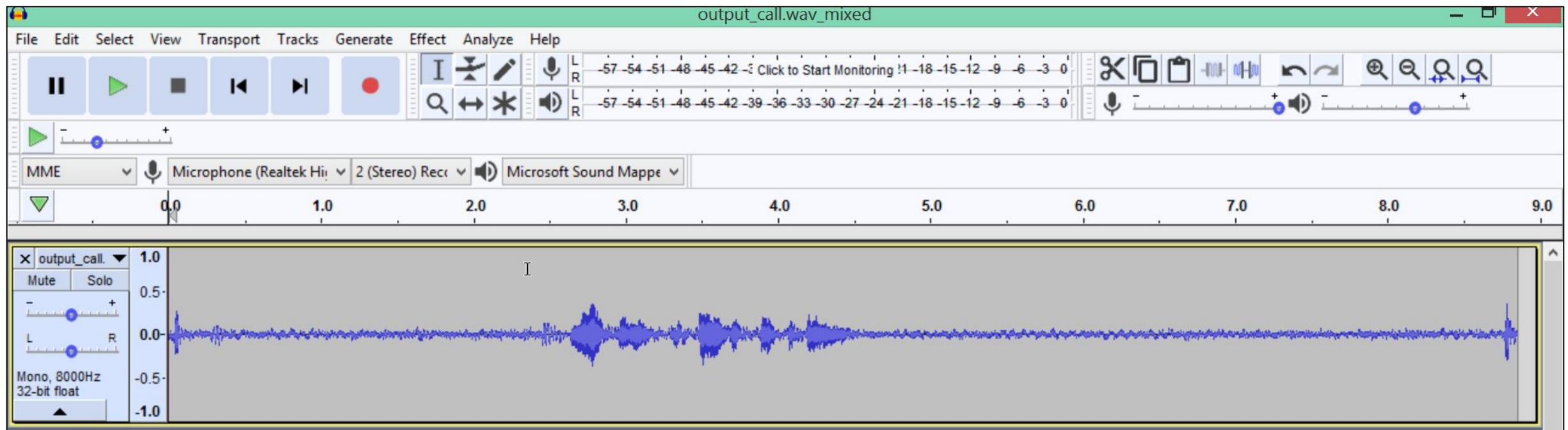
- Directory content before running the script

```
root@PentesterAcademy:/work/pcap2wav# ls -l
total 232
-rwxr-xr-x 1 root root 5927 Mar 27 01:18 pcap2wav.sh
-rw----- 1 root root 226760 Mar 19 17:29 SIP+RTP_call_trace_from_caller_to_PBX.pcap
```

- Directory content after running the script

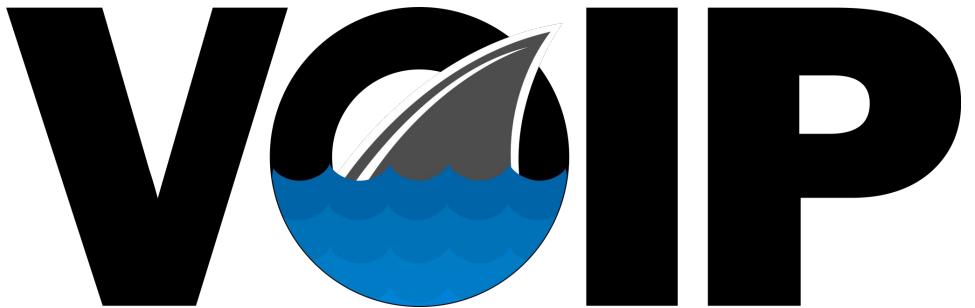
```
root@PentesterAcademy:/work/pcap2wav# ls -l
total 592
-rw-r--r-- 1 root root 70240 Mar 27 03:57 output_call.wav_1.PCMU
-rw-r--r-- 1 root root 70298 Mar 27 03:57 output_call.wav_1.wav
-rw-r--r-- 1 root root 70880 Mar 27 03:57 output_call.wav_2.PCMU
-rw-r--r-- 1 root root 70938 Mar 27 03:57 output_call.wav_2.wav
-rw-r--r-- 1 root root 70938 Mar 27 03:57 output_call.wav_mixed.wav
-rwxr-xr-x 1 root root 5927 Mar 27 01:18 pcap2wav.sh
-rw----- 1 root root 226760 Mar 19 17:29 SIP+RTP_call_trace_from_caller_to_PBX.pcap
```

PCAP2WAV: Wav in audacity



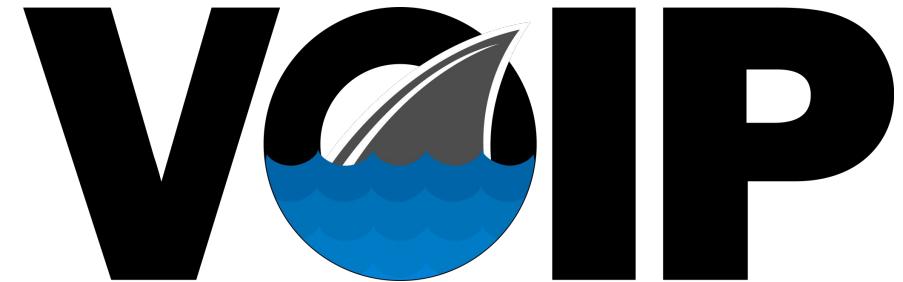
VoIPShark

- Collection of Wireshark plugins to
 - Decrypt VoIP calls
 - Export call audio
 - Overview of traffic (Extensions, SMS, DTMF)
 - Common VoIP attacks
- GPL just like Wireshark
- Github: github.com/pentesteracademy/voipshark



VoIPShark: Need?

- Cumbersome and complex process
- Multiple tools
 - Need compilation, hence time consuming to set-up
 - Not easy to use
 - User dependent, prone to mistakes
- Inability to retain timestamp, IP addresses etc. during decryption
- Live traffic not supported

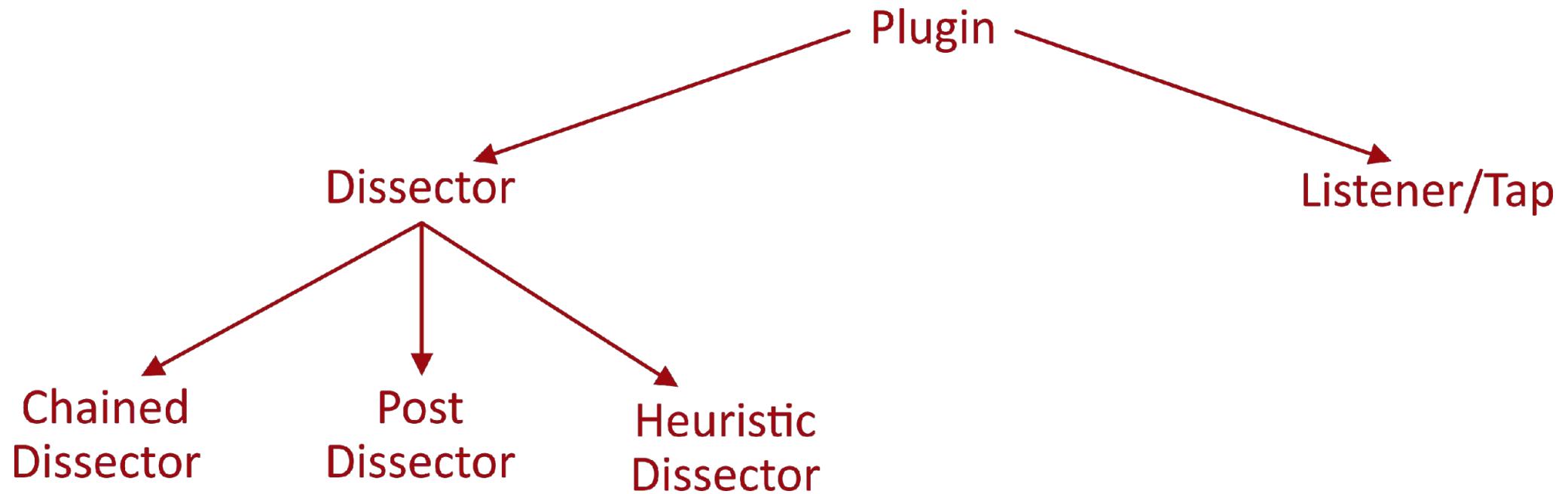


Why Wireshark Plugins?

- Plug and play
- Plugins can be
 - Lua scripts
 - Compiled C/C++ code
- Harnessing power of Wireshark
- OS independent
- Large user base



Wireshark Plugins Types



Dissector

- To interpret the payload data
- Decodes its part of the protocol and passes the payload to next

Example Dissection Flow



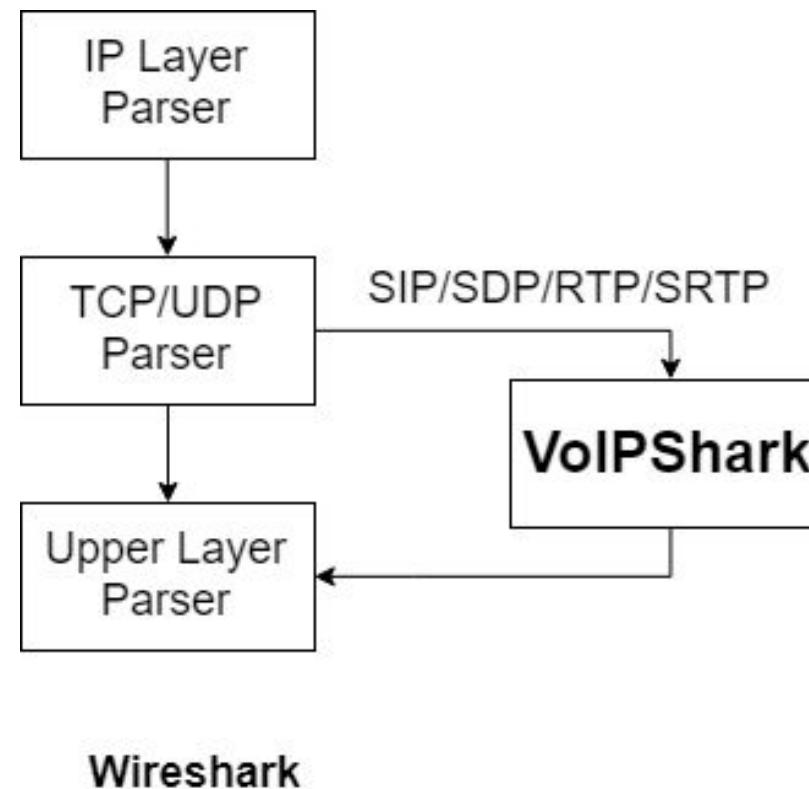
Chained Dissector

- Takes data from previous dissector, processes its part and pass the payload to next dissector

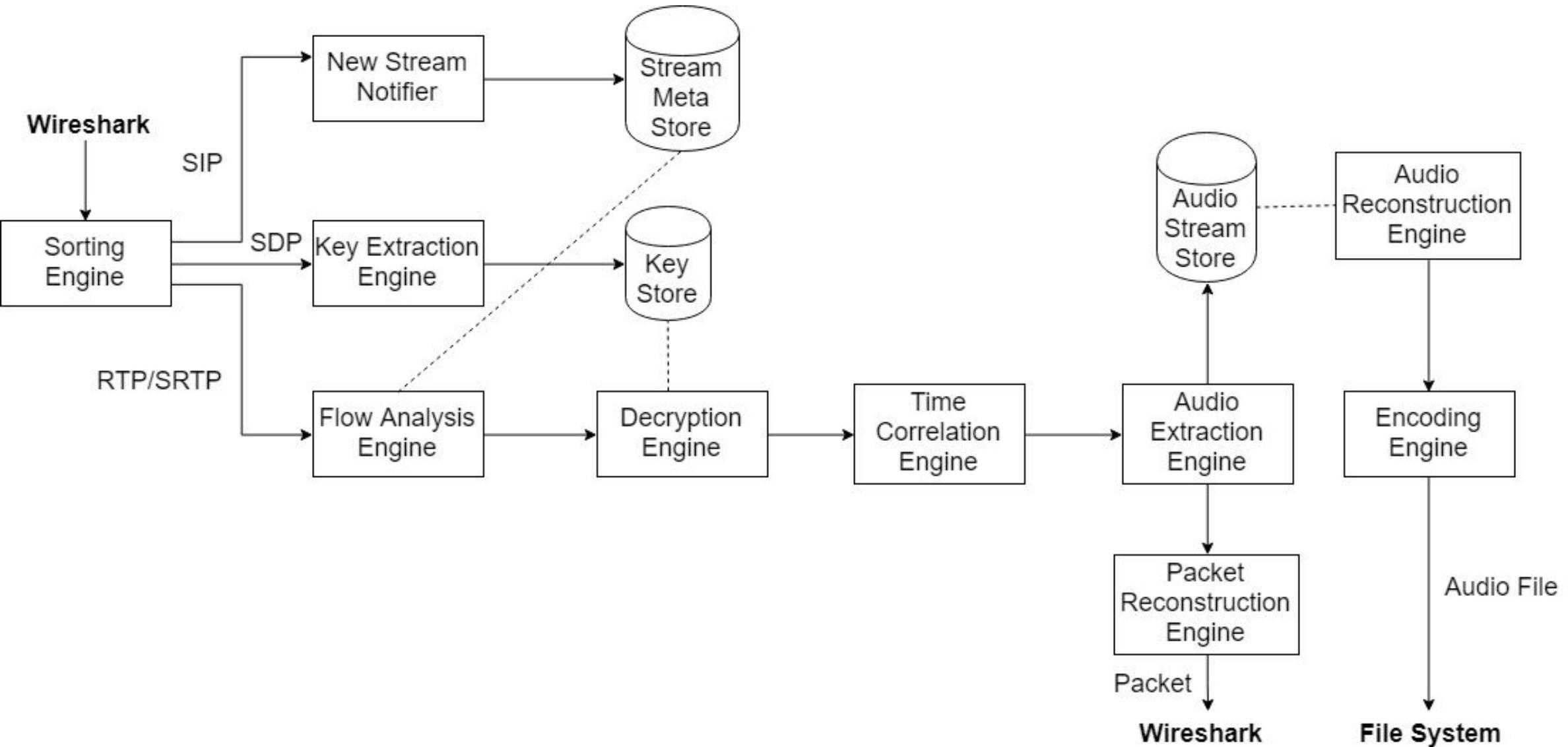
Example Dissection Flow



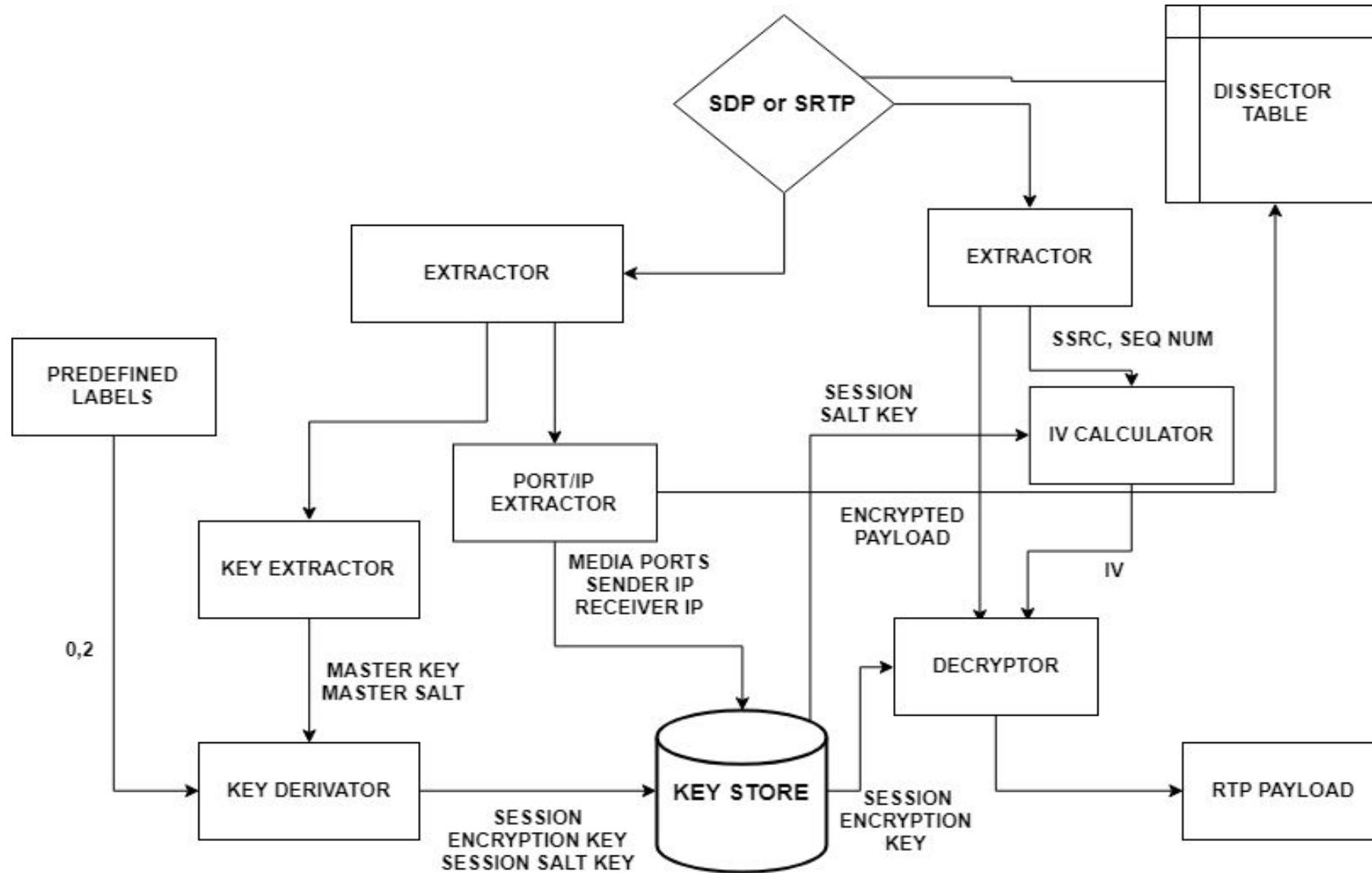
VoIPShark: Hook in Dissector Chain



VoIPShark: Overall Architecture



VoIPShark: Decryption Routines



Plugins locations

- Check Help > About Wireshark > Folders

Windows

About Wireshark		
Wireshark	Authors	Folders
Name	Location	Typical Files
"File" dialogs	C:\Users\Nishant\Desktop\Testing Wireshark Plugin	capture files
Temp	C:\Users\Nishant\AppData\Local\Temp	untitled capture files
Personal configuration	C:\Users\Nishant\AppData\Roaming\Wireshark	dfilters, preferences, ethers, ...
Global configuration	C:\Program Files\Wireshark	dfilters, preferences, manuf, ...
System	C:\Program Files\Wireshark	ethers, ipxnets
Program	C:\Program Files\Wireshark	program files
Personal Plugins	C:\Users\Nishant\AppData\Roaming\Wireshark\plugins	dissector plugins
Global Plugins	C:\Program Files\Wireshark\plugins\2.4.5	dissector plugins
Extcap path	C:\Program Files\Wireshark\extcap	Extcap Plugins search path

Ubuntu

About Wireshark		
Wireshark	Authors	Folders
Name	Folder	Typical Files
"File" dialogs	/root/	capture files
Temp	/tmp	untitled capture files
Personal configuration	/root/.wireshark/	"dfilters", "preferences", "ethers", "ipxnets"
Global configuration	/usr/share/wireshark	"dfilters", "preferences", "manuf", "ethers", "ipxnets"
System	/etc	program files
Program	/usr/bin	dissector plugins
Personal Plugins	/root/.wireshark/plugins	dissector plugins
Global Plugins	/usr/lib/x86_64-linux-gnu/wireshark/plugins/1.12.1	dissector plugins

Decrypting SRTP: SRTP Packets

Normal_Call_two_parties.pcap

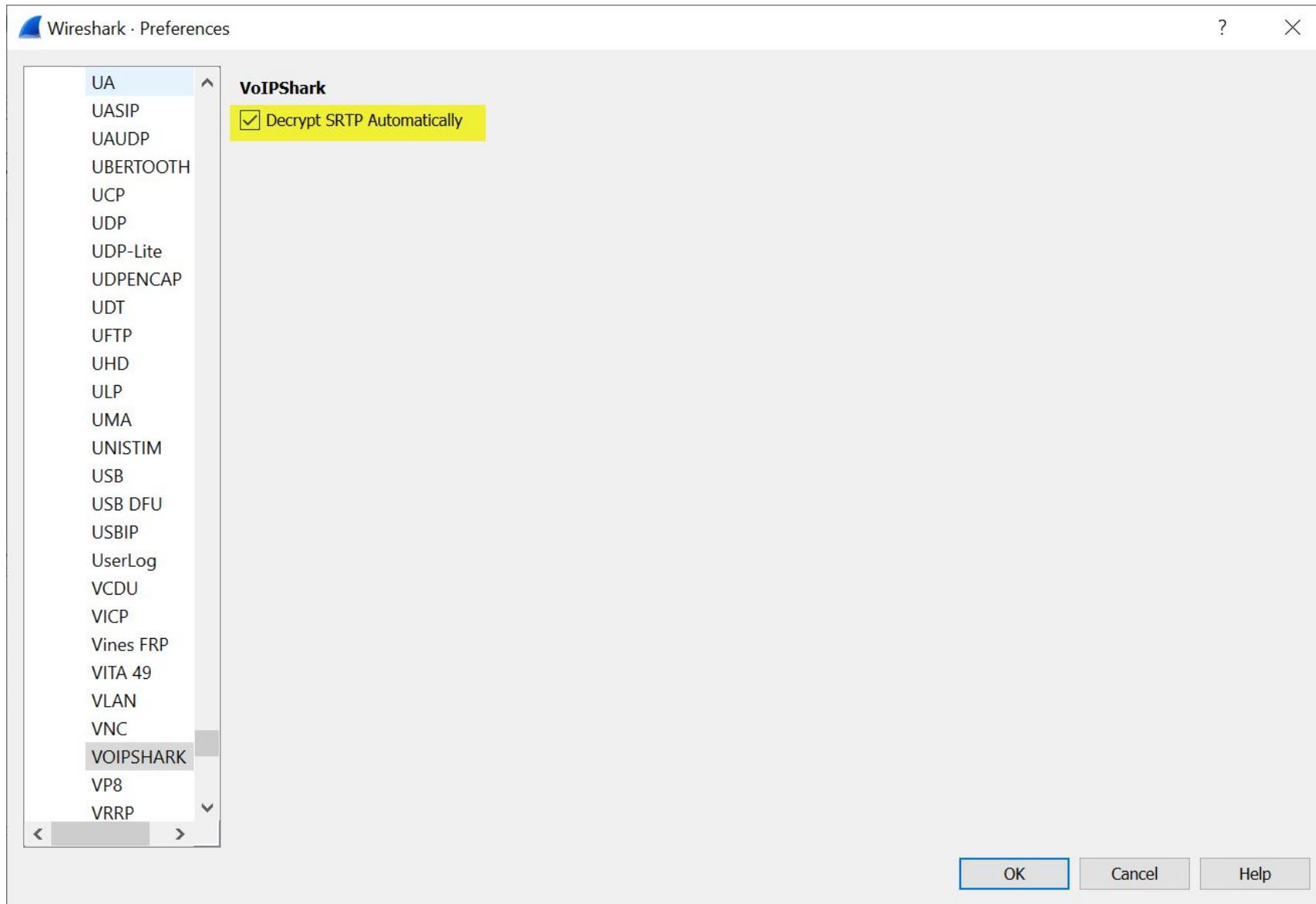
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

rtp

No.	Time	Source	Destination	Protocol	Length	SSID	Sequence number	Info
177	29.311833	192.168.20.1	192.168.20.130	SRTP	224			PT=ITU-T G.711 PCMU, SSRC=0x3
183	29.316949	192.168.20.130	192.168.20.132	SRTP	224			PT=ITU-T G.711 PCMU, SSRC=0x6
189	29.332471	192.168.20.1	192.168.20.130	SRTP	224			PT=ITU-T G.711 PCMU, SSRC=0x3
190	29.333063	192.168.20.130	192.168.20.132	SRTP	224			PT=ITU-T G.711 PCMU, SSRC=0x6
191	29.334585	192.168.20.132	192.168.20.130	SRTP	224			PT=ITU-T G.711 PCMU, SSRC=0x1
192	29.334904	192.168.20.130	192.168.20.1	SRTP	224			PT=ITU-T G.711 PCMU, SSRC=0x4
193	29.352961	192.168.20.1	192.168.20.130	SRTP	224			PT=ITU-T G.711 PCMU, SSRC=0x3
194	29.353301	192.168.20.130	192.168.20.132	SRTP	224			PT=ITU-T G.711 PCMU, SSRC=0x6
195	29.354843	192.168.20.132	192.168.20.130	SRTP	224			PT=ITU-T G.711 PCMU, SSRC=0x1
196	29.355005	192.168.20.130	192.168.20.1	SRTP	224			PT=ITU-T G.711 PCMU, SSRC=0x4
197	29.372665	192.168.20.1	192.168.20.130	SRTP	224			PT=ITU-T G.711 PCMU, SSRC=0x3
198	29.372952	192.168.20.130	192.168.20.132	SRTP	224			PT=ITU-T G.711 PCMU, SSRC=0x6
199	29.375160	192.168.20.132	192.168.20.130	SRTP	224			PT=ITU-T G.711 PCMU, SSRC=0x1

> Frame 177: 224 bytes on wire (1792 bits), 224 bytes captured (1792 bits)
> Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_ff:65:9b (00:0c:29:ff:65:9b)
> Internet Protocol Version 4, Src: 192.168.20.1, Dst: 192.168.20.130
> User Datagram Protocol, Src Port: 4000, Dst Port: 16450
> Real-Time Transport Protocol

Decrypting SRTP: Enabling Auto Decryption



Decrypting SRTP: Decrypted SRTP (RTP)

No.	Time	Source	Destination	Protocol	Length	SSID	Sequence number	Info
177	29.311833	192.168.20.1	192.168.20.130	RTP	224			PT=ITU-T G.711 PCMU, SSRC=0x3
183	29.316949	192.168.20.130	192.168.20.132	RTP	224			PT=ITU-T G.711 PCMU, SSRC=0x6
189	29.332471	192.168.20.1	192.168.20.130	RTP	224			PT=ITU-T G.711 PCMU, SSRC=0x3
190	29.333063	192.168.20.130	192.168.20.132	RTP	224			PT=ITU-T G.711 PCMU, SSRC=0x6
191	29.334585	192.168.20.132	192.168.20.130	RTP	224			PT=ITU-T G.711 PCMU, SSRC=0x1
192	29.334904	192.168.20.130	192.168.20.1	RTP	224			PT=ITU-T G.711 PCMU, SSRC=0x4
193	29.352961	192.168.20.1	192.168.20.130	RTP	224			PT=ITU-T G.711 PCMU, SSRC=0x3
194	29.353301	192.168.20.130	192.168.20.132	RTP	224			PT=ITU-T G.711 PCMU, SSRC=0x6
195	29.354843	192.168.20.132	192.168.20.130	RTP	224			PT=ITU-T G.711 PCMU, SSRC=0x1
196	29.355005	192.168.20.130	192.168.20.1	RTP	224			PT=ITU-T G.711 PCMU, SSRC=0x4
197	29.372665	192.168.20.1	192.168.20.130	RTP	224			PT=ITU-T G.711 PCMU, SSRC=0x3
198	29.372952	192.168.20.130	192.168.20.132	RTP	224			PT=ITU-T G.711 PCMU, SSRC=0x6
199	29.375160	192.168.20.132	192.168.20.130	RTP	224			PT=ITU-T G.711 PCMU, SSRC=0x1

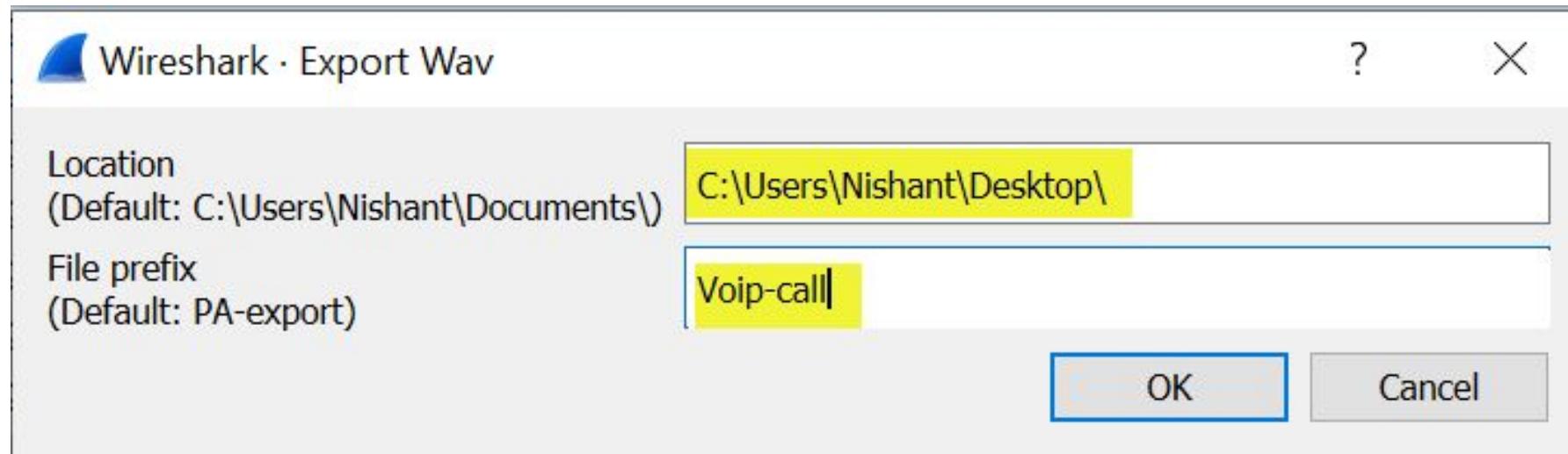
> Frame 177: 224 bytes on wire (1792 bits), 224 bytes captured (1792 bits)
> Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_ff:65:9b (00:0c:29:ff:65:9b)
> Internet Protocol Version 4, Src: 192.168.20.1, Dst: 192.168.20.130
> User Datagram Protocol, Src Port: 4000, Dst Port: 16450
> Real-Time Transport Protocol

VoIPShark: Exporting Call Audio

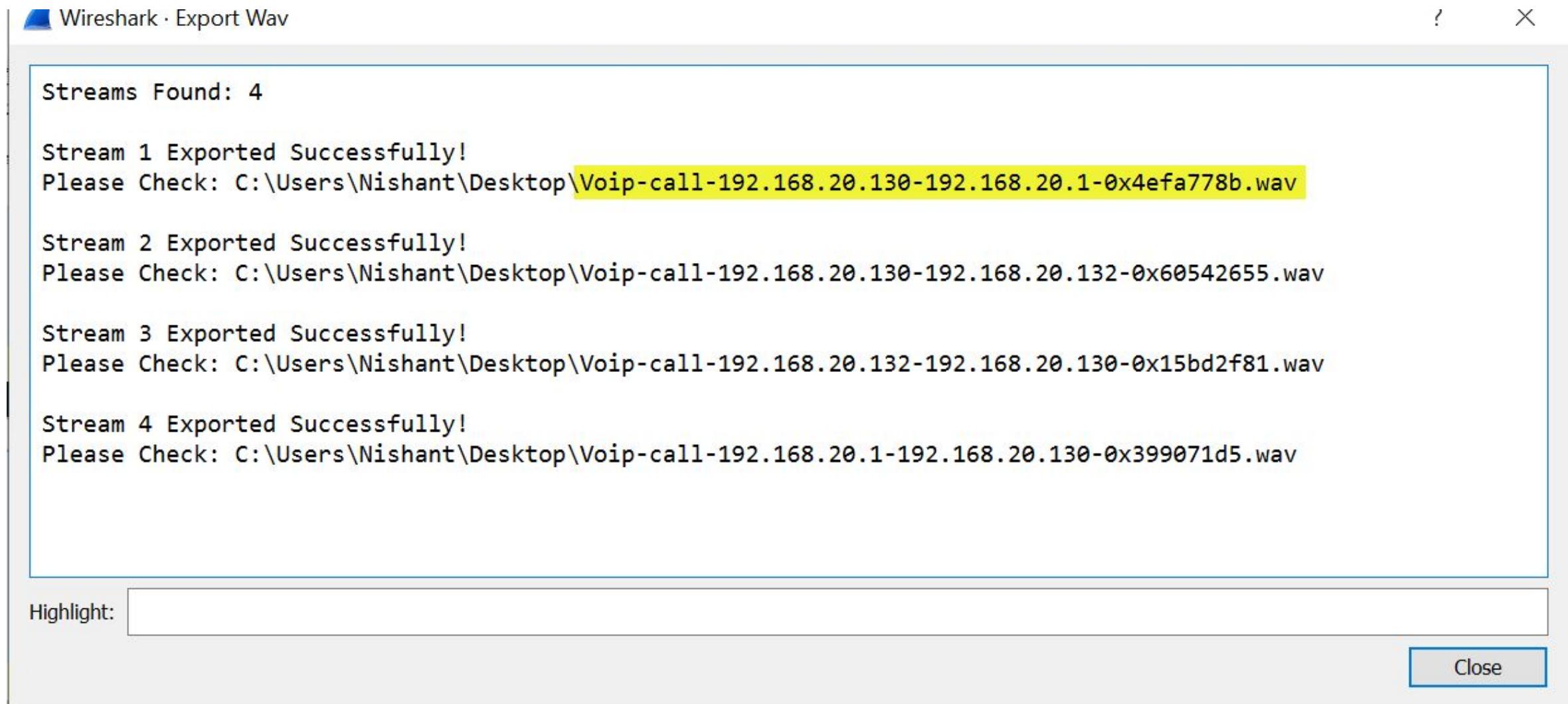
The screenshot shows the VoIPShark application interface. The main window displays a list of network packets captured over the 'rtp' interface. The columns in the table are: No., Time, Source, Destination, Protocol, and Sequence number. The sequence numbers for the listed packets range from 177 to 195. The 'Protocol' column shows most entries as RTP, except for the last two which are 224. A context menu is open at the bottom right of the screen, specifically over the 'VOIP' option under the 'Tools' menu. The menu items shown are: Firewall ACL Rules, Lua, VOIP (which is the selected item), Export Wav (highlighted in yellow), SIP Information Gathering, and VOIP Attack Detection.

No.	Time	Source	Destination	Protocol	Sequence number
177	29.311833	192.168.20.1	192.168.20.130	RTP	
183	29.316949	192.168.20.130	192.168.20.132	RTP	
189	29.332471	192.168.20.1	192.168.20.130	RTP	224
190	29.333063	192.168.20.130	192.168.20.132	RTP	224
191	29.334585	192.168.20.132	192.168.20.130	RTP	224
192	29.334904	192.168.20.130	192.168.20.1	RTP	224
193	29.352961	192.168.20.1	192.168.20.130	RTP	224
194	29.353301	192.168.20.130	192.168.20.132	RTP	224
195	29.354843	192.168.20.132	192.168.20.130	RTP	224

Exporting Call Audio: Specifying Location and File name



Exporting Call Audio: Exported Streams



VoIPShark: SIP Information Gathering

The screenshot shows the VoIPShark application interface. The main window displays a list of network packets captured over time, with columns for Time, Source, Destination, Protocol, and Sequence number. The Protocol column shows most entries as RTP, while the last entry is labeled RTP. The Sequence number column shows values such as 224, 224, 224, 224, 224, 224, 224, 224, 224, 224, 224, 224, 224, 224, and 224. The last row of the table is partially visible with the sequence number 224.

The top navigation bar includes links for Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The Tools menu is currently open, showing options like Firewall ACL Rules, Lua, VOIP, Export Wav, SIP Information Gathering, and VOIP Attack Detection. The SIP Information Gathering option is highlighted with a yellow box.

A secondary context menu is also open, listing DTMF Sequences, Extensions, RTP Packet Transfers, SIP Auth Export, Servers and Proxy, and Unique Messages, all of which are also highlighted with yellow boxes.

Time	Source	Destination	Protocol	Sequence number	Info
29.311833	192.168.20.1	192.168.20.130	RTP	224	U-T G.711
29.316949	192.168.20.130	192.168.20.132	RTP	224	U-T G.711
29.332471	192.168.20.1	192.168.20.130	RTP	224	U-T G.711
29.333063	192.168.20.130	192.168.20.132	RTP	224	U-T G.711
29.334585	192.168.20.132	192.168.20.130	RTP	224	U-T G.711
29.334904	192.168.20.130	192.168.20.1	RTP	224	U-T G.711
29.352961	192.168.20.1	192.168.20.130	RTP	224	U-T G.711
29.353301	192.168.20.130	192.168.20.132	RTP	224	PT=ITU-T G.711
29.354843	192.168.20.132	192.168.20.130	RTP	224	PT=ITU-T G.711
29.355005	192.168.20.130	192.168.20.1	RTP	224	PT=ITU-T G.711
29.372665	192.168.20.1	192.168.20.130	RTP	224	PT=ITU-T G.711
29.372952	192.168.20.130	192.168.20.132	RTP	224	PT=ITU-T G.711
29.375160	192.168.20.132	192.168.20.130	RTP	224	PT=ITU-T G.711

SIP Information Gathering : DTMF

Wireshark · DTMF Sequence

S.no	Call Source	Call Destination	Media Port	DTMF Sequence
1	192.168.20.132	192.168.20.130	4000 -> 15766	4492 9226 1492 6989 1222 679
2	192.168.20.130	192.168.20.1	12684 -> 4000	4492 9226 1492 6989 1222 679

Highlight:

Reset Search Close

SIP Information Gathering: Extensions

Wireshark · Extensions

S.no	Extension	Username	Host	User Agent
1	1111	Bob	192.168.20.132	MicroSIP/3.18.2
2	2222	Alice	192.168.20.1	MicroSIP/3.18.2

Highlight:

SIP Information Gathering: RTP Packet Transfers

Wireshark · RTP Packet Transfers

S.no	Call ID	Caller	Callee	Media Port	Packets Sent	Packets Received
1	bb7b800c-bbae-4cd8-b30f-ea9d6c6dfd6	192.168.20.1	192.168.20.130	4000<->16450	520	515
2	df715f19130d447a8d790f6c57c6a049	192.168.20.130	192.168.20.132	17786<->4000	520	516

Highlight:

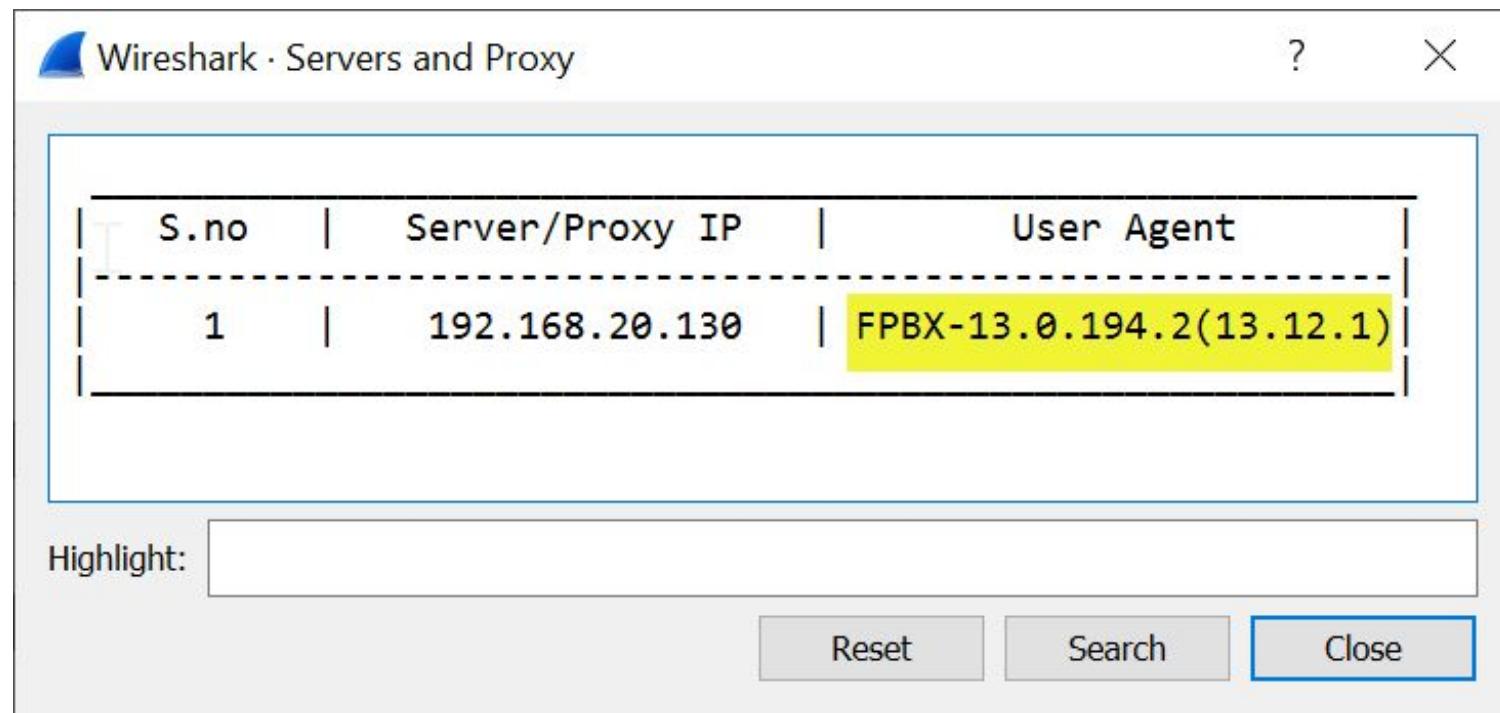
SIP Information Gathering : SIP Auth Export

Wireshark · SIP Auth Export

S.no	username	Client IP	Server IP	Message Digest
1	1111	192.168.20.132	192.168.20.130	6a09af4b796d1b5ff376726fa9ae1ad9 \$sip\$***1111*asterisk*REGISTER*sip*192.168.20.130**1522268723/ f872129e9c735809884cb64de141967e*1c109c4b8a064ef5ae277c4d7d07c4d1*00000001*auth*MD5*6a09af4b796d1b5ff376726f a9ae1ad9
2	2222	192.168.20.1	192.168.20.130	f28aa9d6f10944e06f8693337fd3ba19 \$sip\$***2222*asterisk*REGISTER*sip*192.168.20.130**1522268729/ b27f0c3e27b25533a8ae9a41de712696*81aca7938c994d1d93d4abc8007095b5*00000001*auth*MD5*f28aa9d6f10944e06f869333 7fd3ba19

Highlight:

SIP Information Gathering : Servers and Proxy



The screenshot shows a 'Wireshark - Servers and Proxy' search results dialog. The table has three columns: S.no, Server/Proxy IP, and User Agent. One row is displayed, with the User Agent column highlighted in yellow. The highlighted row contains the values 1, 192.168.20.130, and FPBX-13.0.194.2(13.12.1). Below the table is a 'Highlight:' input field and three buttons: 'Reset', 'Search', and 'Close'.

S.no	Server/Proxy IP	User Agent
1	192.168.20.130	FPBX-13.0.194.2(13.12.1)

Highlight:

Reset Search Close

SIP Information Gathering: Unique Messages

Wireshark · Unique Messages

S.no	Sender Username	Message Sender IP	Reciever Username	Message Reciever IP	Message
1	1111	192.168.20.136	2222	192.168.20.130	Hello world to sip messaging!
2	1111	192.168.20.130	2222	192.168.20.1	Hello world to sip messaging!

Highlight:

Reset Search Close

VoIPShark: VoIP Attack Detection

The screenshot shows the VoIPShark application interface. The main window displays a list of network traffic captures. The top navigation bar includes tabs for 'Analyze', 'Statistics', 'Telephony', 'Wireless', 'Tools' (which is currently selected), and 'Help'. Below the navigation bar is a toolbar with various icons for search, zoom, and file operations. A context menu is open over one of the traffic entries, specifically under the 'Protocol' column. This menu is titled 'VOIP' and contains several options: 'Export Wav', 'SIP Information Gathering', and 'VOIP Attack Detection'. The 'VOIP Attack Detection' option is highlighted with a green background. A secondary submenu for 'VOIP Attack Detection' is displayed, listing five attack types: 'Brute Force', 'Invite Flooding', 'MITM Attempts', 'Message Flooding', and 'Unauthenticated Users'. The main table area shows columns for 'Source', 'Destination', 'Protocol', 'Sequence number', and 'Info'. Most entries show RTP traffic between various IP addresses, with sequence numbers mostly at 224. The 'Info' column includes some specific PT=ITU-T G.711 entries.

Source	Destination	Protocol	Sequence number	Info
192.168.20.1	192.168.20.130	RTP	224	PT=ITU-T G.711
192.168.20.130	192.168.20.132	RTP	224	-T G.711
192.168.20.1	192.168.20.130	RTP	224	-T G.711
192.168.20.130	192.168.20.132	RTP	224	-T G.711
192.168.20.132	192.168.20.130	RTP	224	-T G.711
192.168.20.130	192.168.20.1	RTP	224	-T G.711
192.168.20.1	192.168.20.130	RTP	224	-T G.711
192.168.20.130	192.168.20.132	RTP	224	PT=ITU-T G.711
192.168.20.132	192.168.20.130	RTP	224	PT=ITU-T G.711
192.168.20.130	192.168.20.1	RTP	224	PT=ITU-T G.711
192.168.20.1	192.168.20.130	RTP	224	PT=ITU-T G.711
192.168.20.130	192.168.20.132	RTP	224	PT=ITU-T G.711
192.168.20.132	192.168.20.130	RTP	224	PT=ITU-T G.711

VoIP Attack Detection: Bruteforce

Wireshark · Brute Force

S.no	Attacker Machine	Target Extension	Target Machine	Requests Sent	Failed Attempts	Requests Per second
1	192.168.20.134	1111	192.168.20.130	7	6	167.54
2	192.168.20.134	2222	192.168.20.130	9	8	151.65

Highlight:

Reset Search Close

VoIP Attack Detection: Invite Flooding

Wireshark · Invite Flooding

S.no	Attacker Machine	Attacker Extension	Target Extension	Target Machine	Invites Sent	Invites Per second
1	192.168.20.134	PentesterAcademy	1111	192.168.20.132	14	0.22

Highlight:

Reset Search Close

VoIP Attack Detection: Message Flooding

Wireshark · Message Flooding

S.no	Attacker Machine	Target Machine	Messages Sent	Messages Per second
1	192.168.20.134	192.168.20.130	1024	176.20

Highlight:

Reset Search Close

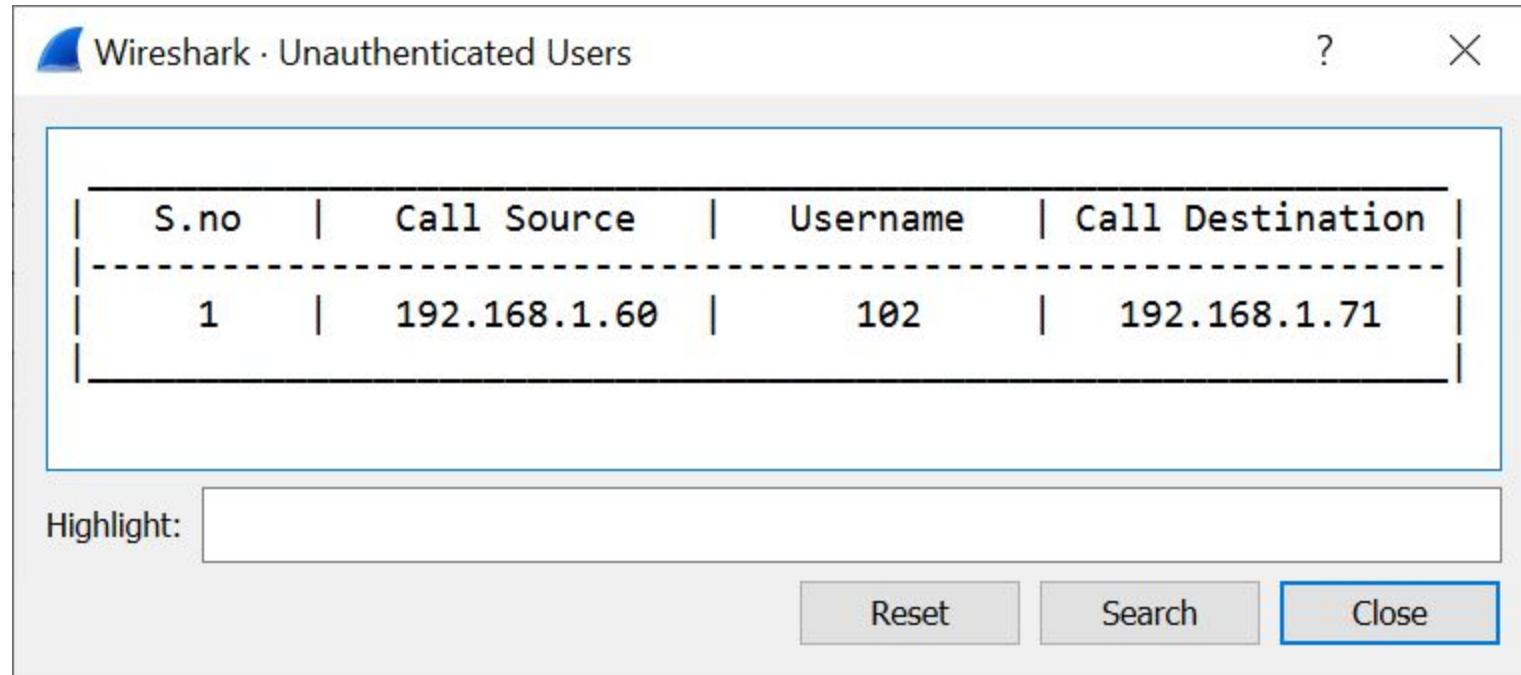
VoIP Attack Detection: MiTM Attempts

Wireshark · MITM Attempts

S.no	Call Source	Call Destination	Source Mac	Destination Mac	Attacker Mac
1	192.168.1.60	192.168.1.71	00:0c:29:9c:2f:3f ,48:0f:cf:4b:06:c9	48:0f:cf:4b:06:c9 ,f8:a9:63:4b:c4:4d	48:0f:cf:4b:06:c9

Highlight:

VoIP Attack Detection: Unauthenticated Users



The screenshot shows a Wireshark dialog box titled "Wireshark · Unauthenticated Users". The dialog contains a table with four columns: "S.no", "Call Source", "Username", and "Call Destination". A single row is listed with values: 1, 192.168.1.60, 102, and 192.168.1.71. Below the table is a "Highlight:" input field and three buttons: "Reset", "Search", and "Close".

S.no	Call Source	Username	Call Destination
<hr/>			
1	192.168.1.60	102	192.168.1.71

Highlight:

Reset Search Close

Demo

Q & A

Github: github.com/pentesteracademy/voipshark
nishant@attackdefense.com