VOIPSHARK:开源VOIP分析平台是与非

Nishant Sharma
Jeswin Mathai
Ashish Bhangale
PentesterAcademy.com & AttackDefense.com

©PentesterAcademy.com

关于我们

我, **Nishant Sharma**

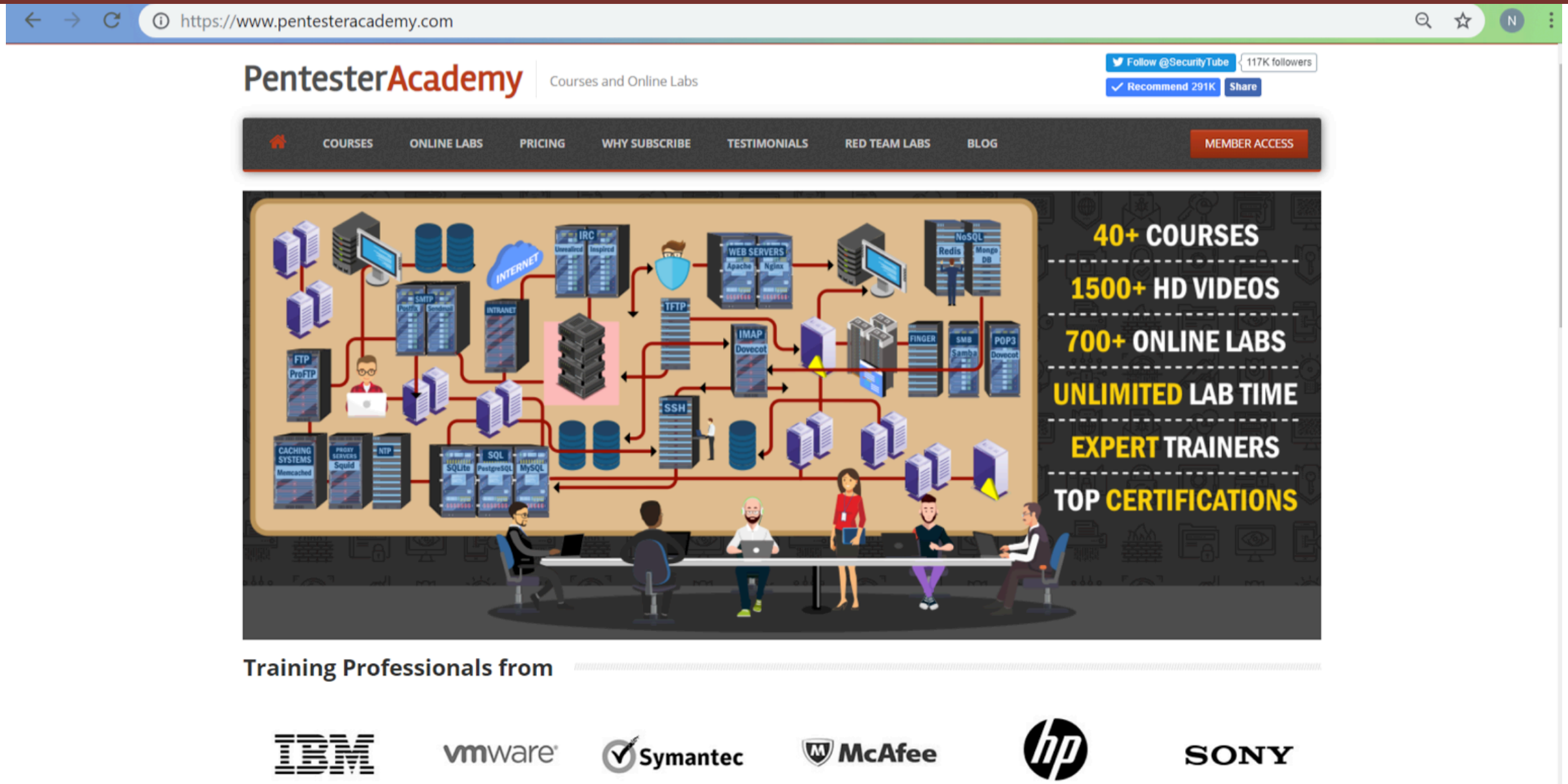
- R&D 经理及资深培训师, Pentester Academy
- 固件开发者, 企业级WiFi APs和WIPS传感器
- 信息安全硕士
- 曾在US/Asia Blackhat、DEFCON USA及其它大会上发表演讲

合作者

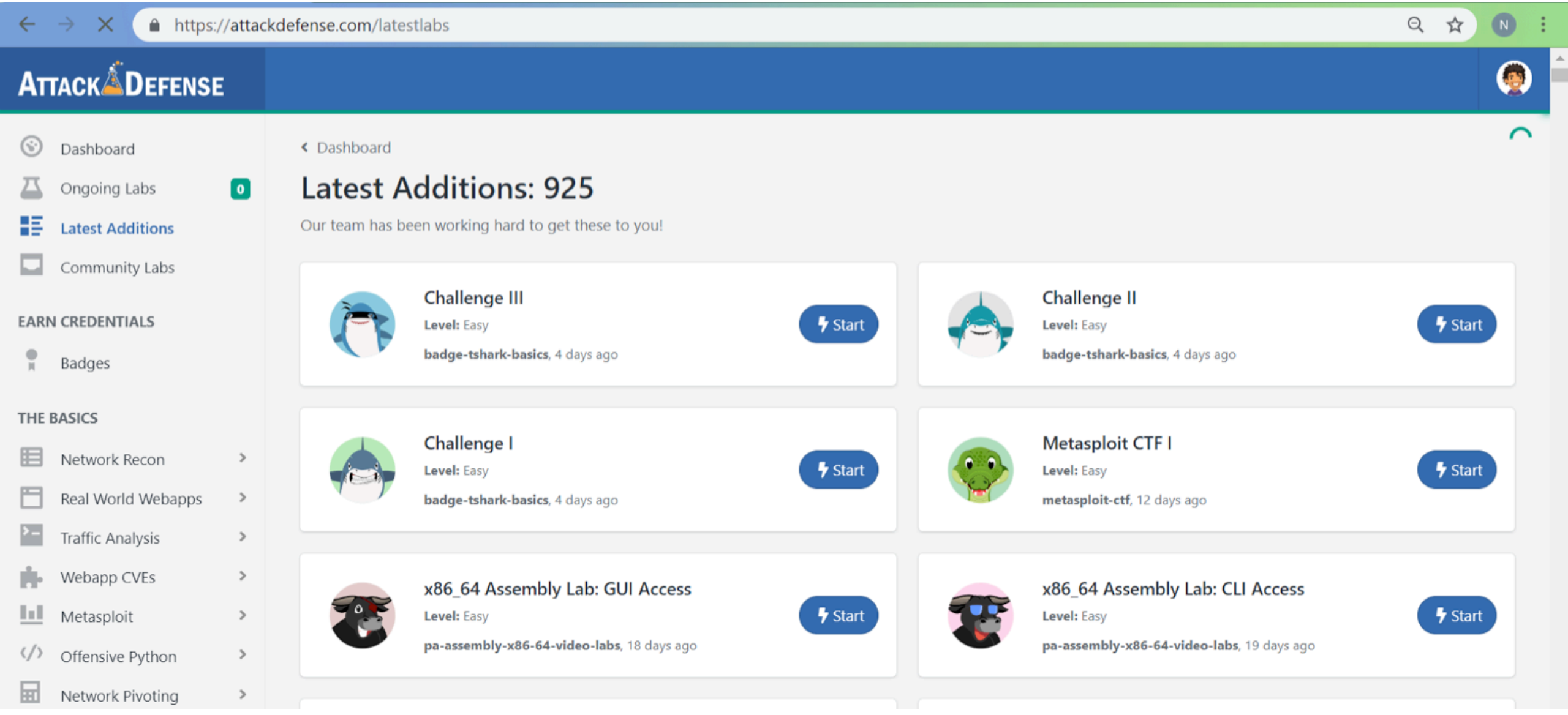
- **Ashish Bhangale**, 高级安全研究员
- **Jeswin Mathai**, 安全研究员

©PentesterAcademy.com

PentesterAcademy.com



AttackDefense.com



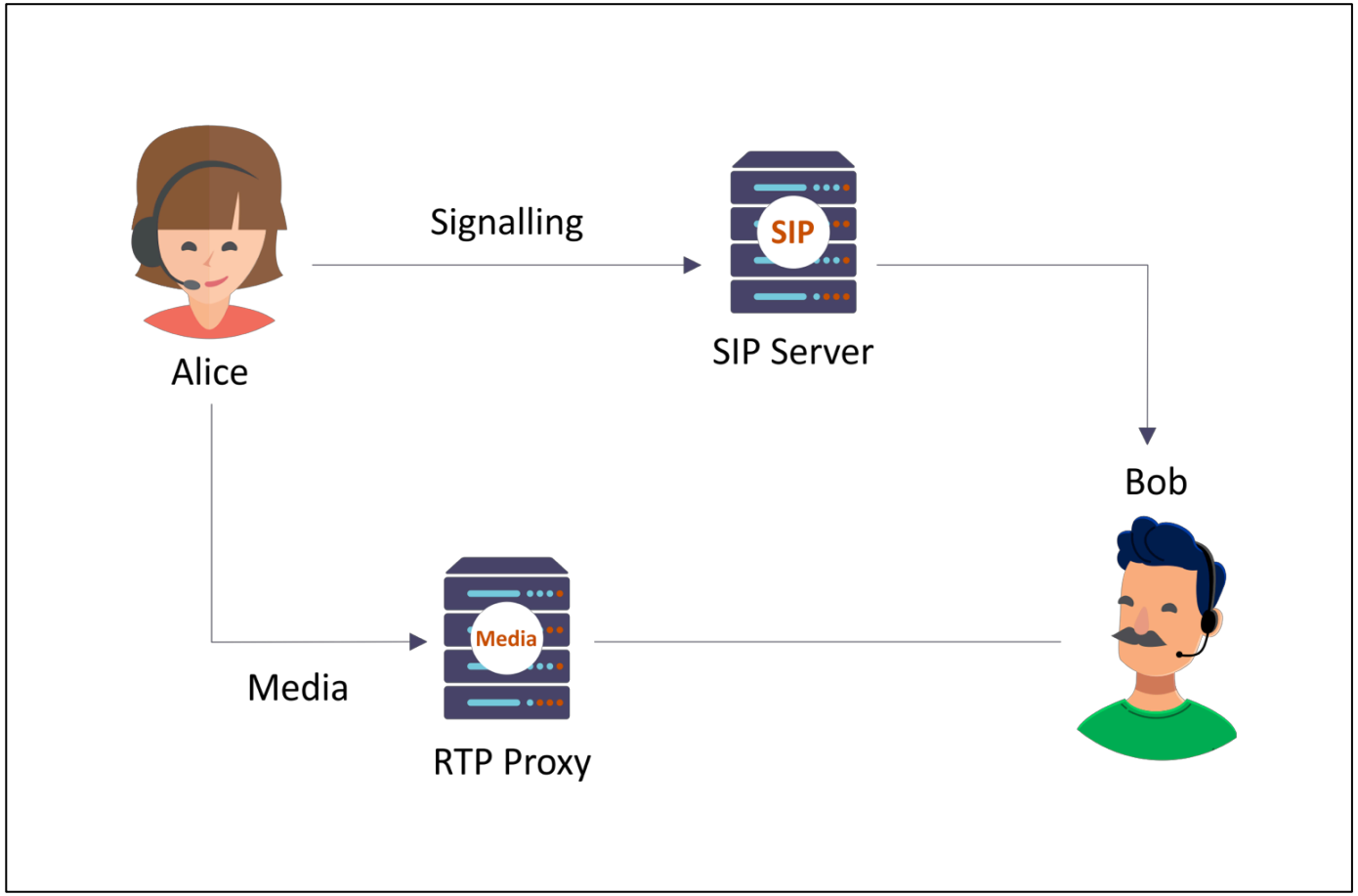
演讲概览

- VoIP基本知识
  - SIP, RTP
  - 安全相关: TLS, SRTP
- 复原/解密VoIP通话
- 目前已有的开源工具及其问题
- VoIPShark
  - 架构及内部原理
  - 分析VoIP流量
  - 复原通话
  - 被动攻击检测
  - 演示

©PentesterAcademy.com

VoIP电话通讯

- 信令 + 媒体



©PentesterAcademy.com

信令协议

SIP (会话初始协议)

- IETF制定
- 替代固话及PSTN(公共电话交换网络)

H.323

- ITU-T制定
- 主要为视频会议制定，也用于语音通话

SCCP (瘦客户端)

- 用于电话线路侧控制的思科专有协议

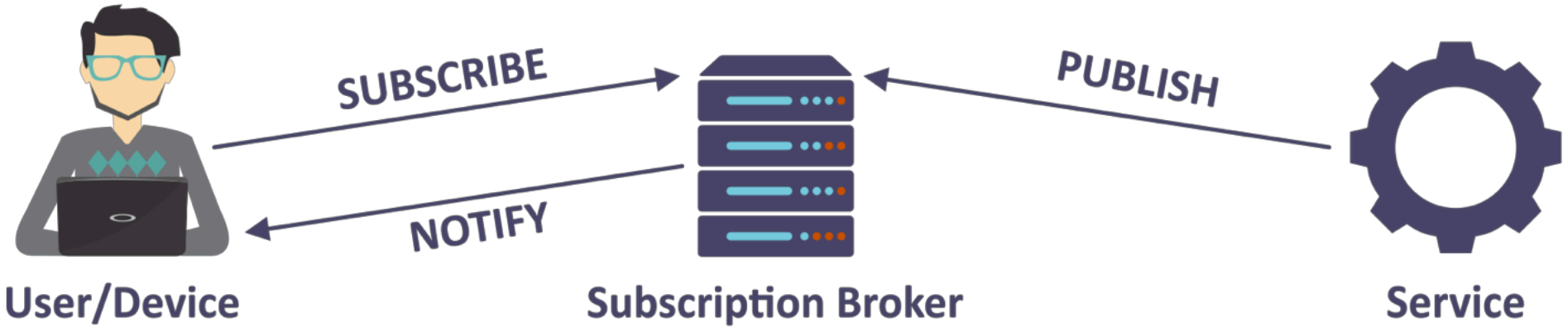
©PentesterAcademy.com

会话初始协议

- 基于文本的协议
- 应用
  - 使用其它媒体流的通话(语音、视频)，如RTP
  - 使用SIP协议的“Message”方法发送文本消息
- 与其它协议协同工作
- 会话描述协议 (SDP) 定义媒体协调和设置过程
- 可在TCP, UDP 或 SCTP (流控制传输协议)上工作
- 安全性由TLS (安全传输层协议) 提供，如SIP over TLS

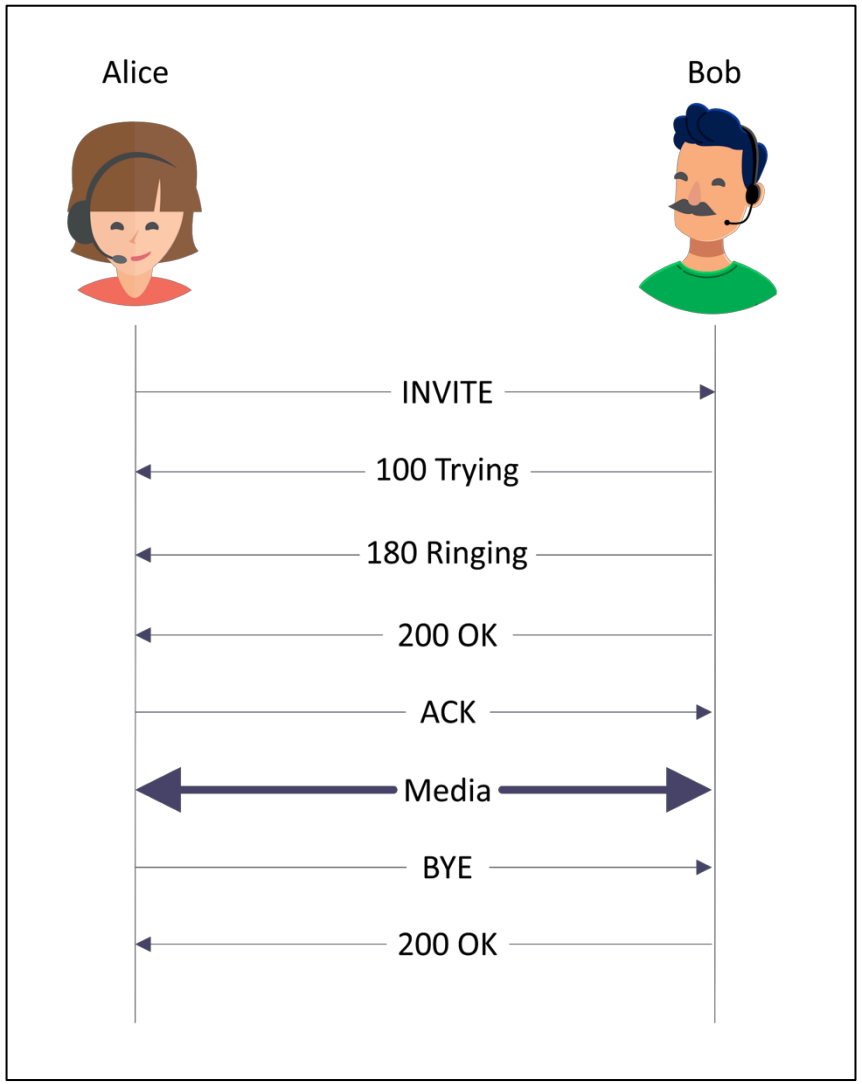
©PentesterAcademy.com

订阅, 发布和通告



©PentesterAcademy.com

会话初始协议: 通话过程示例



©PentesterAcademy.com

用户代理服务 (UAS) 解决方案



www.asterisk.org



www.sipfoundry.org



www.3cx.com



www.elastix.org



freeswitch.org

©PentesterAcademy.com

软电话客户端

- 基于IP的电话服务
- 可选软件
  - Zoiper
  - X Lite
  - LinPhone
  - MicroSIP



www.counterpath.com/x-lite-download



www.microsip.org



www.zoiper.com

选择软电话客户端需考虑的因素

- 是否有编译码支持
- 是否可以加密 (尤其是免费版)
- 其它功能 (如：文本消息、挂起、等待)



www.linphone.org



www.3cx.com

©PentesterAcademy.com

Asterisk Now



=

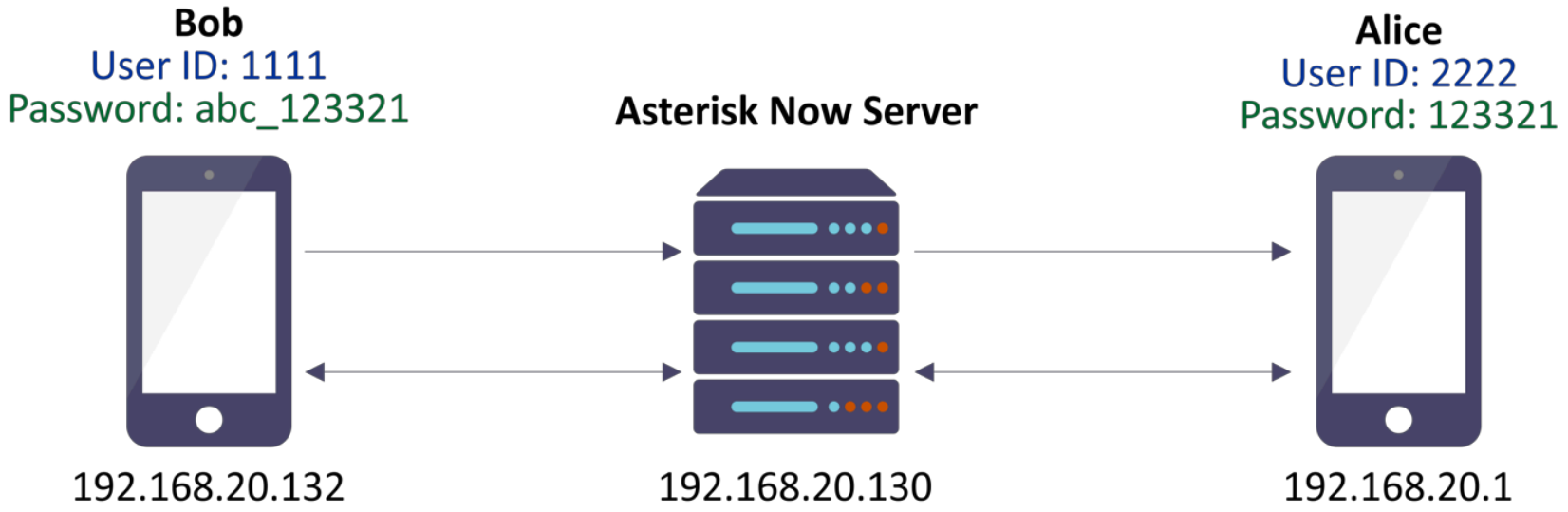


+



©PentesterAcademy.com

Scenario



©PentesterAcademy.com

配置方案

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

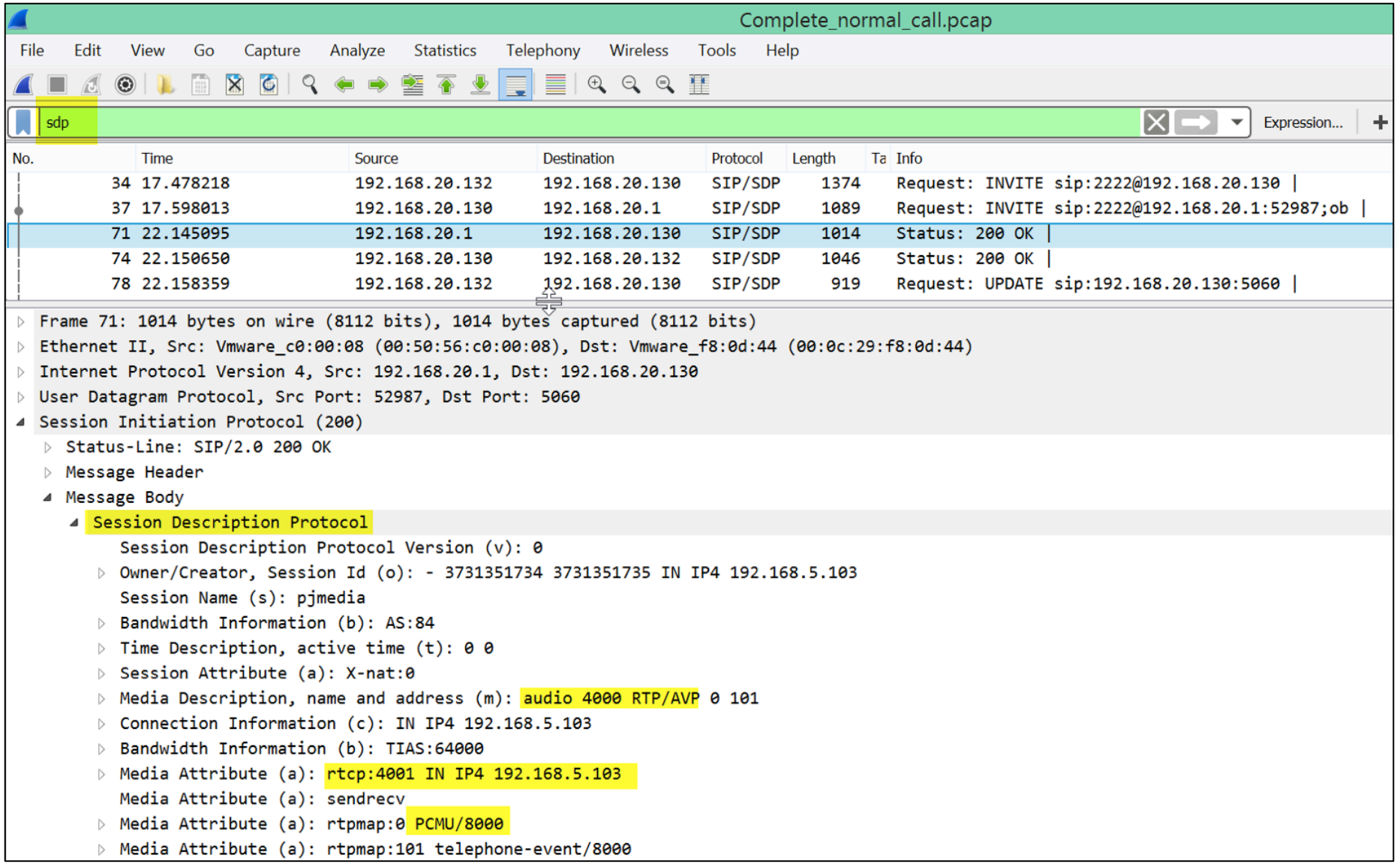
©PentesterAcademy.com

配置方案

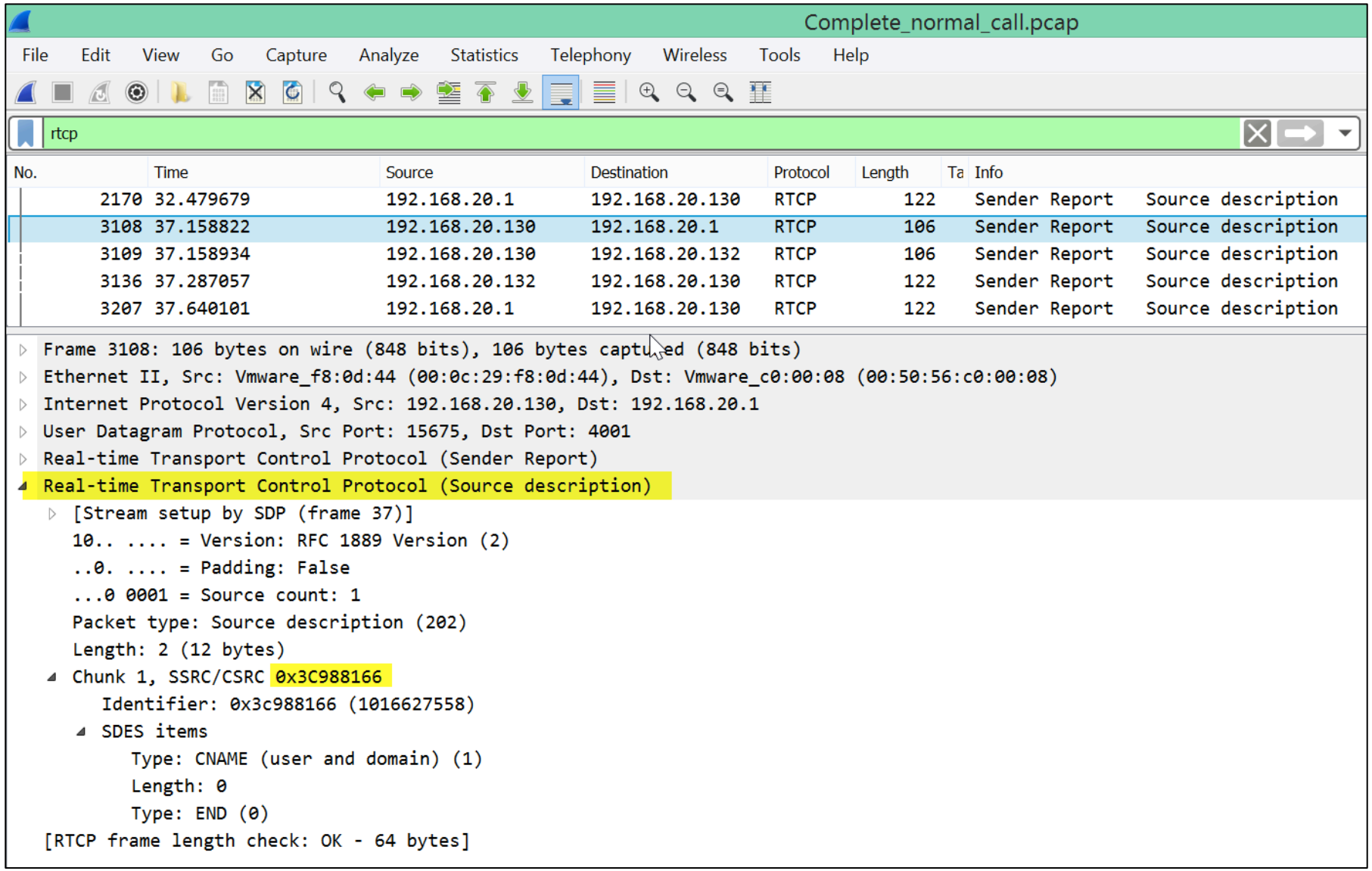
- **SIP + RTP**
- SIP over TLS + RTP
- SIP + SRTP
- SIP over TLS + SRTP

©PentesterAcademy.com

SIP/SDP数据包

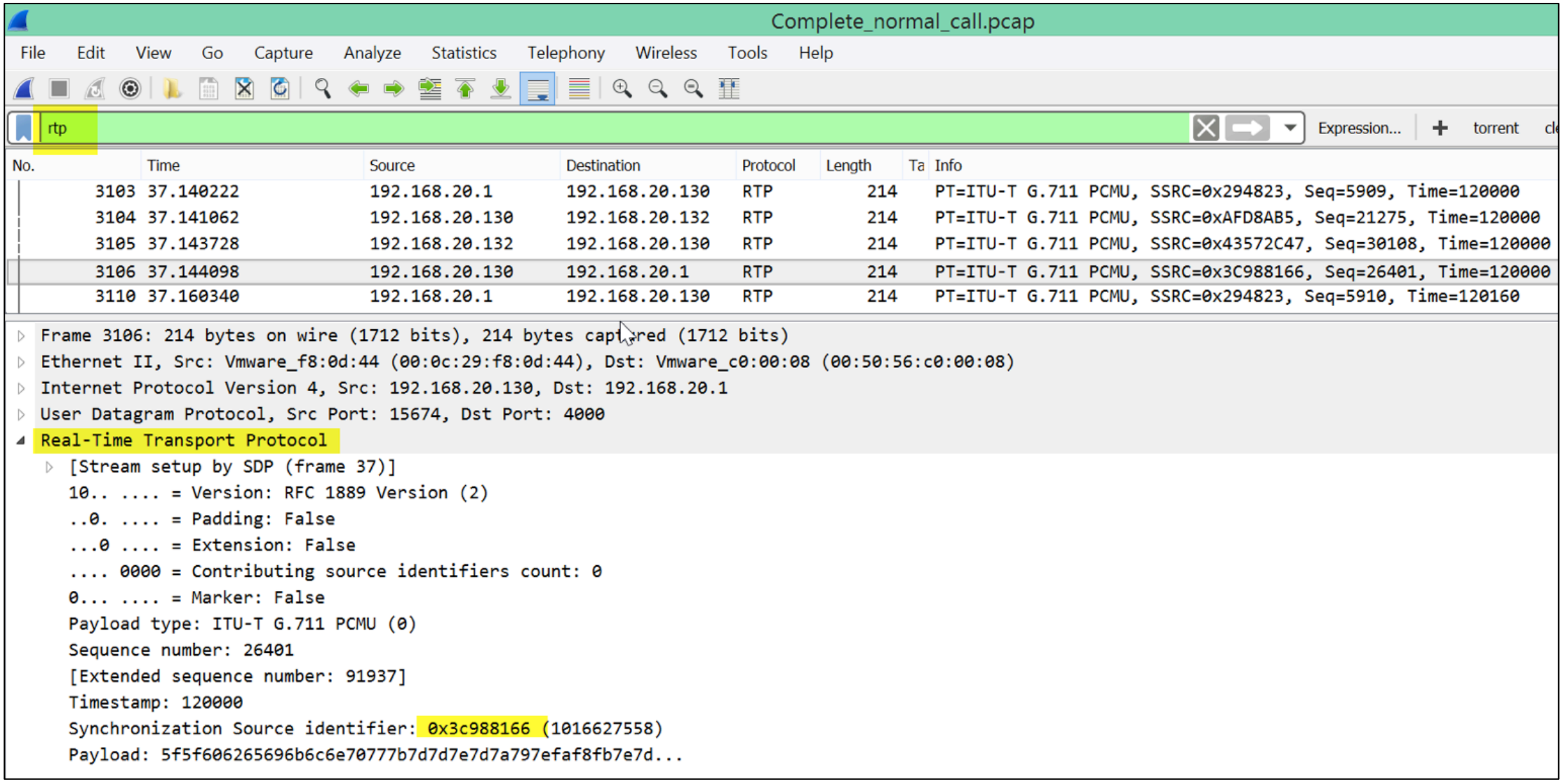

Complete_normal_call.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
sdp
Expression...
No. Time Source Destination Protocol Length Ta Info
34 17.478218 192.168.20.132 192.168.20.130 SIP/SDP 1374 Request: INVITE sip:2222@192.168.20.130 |
37 17.598013 192.168.20.130 192.168.20.1 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数据包



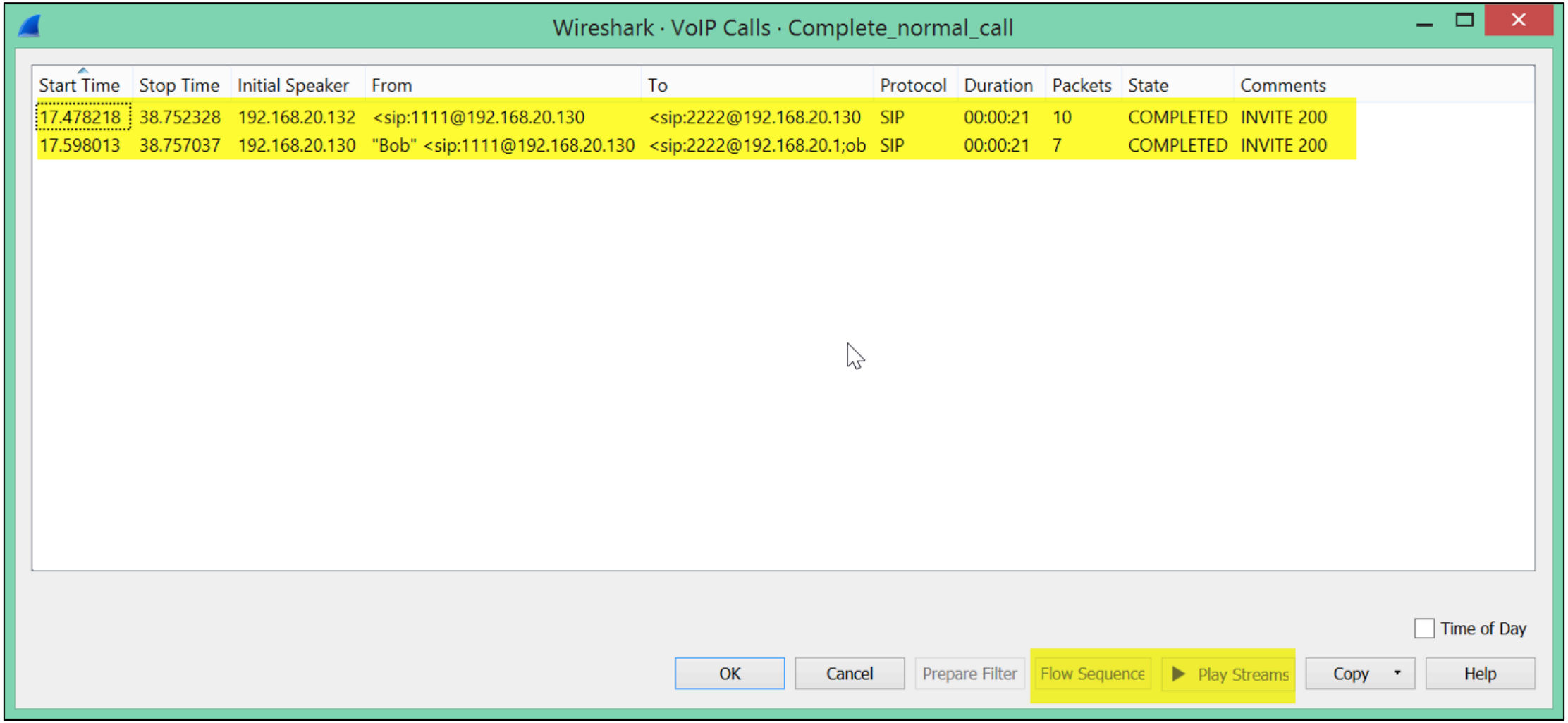
©PentesterAcademy.com

RTP数据包



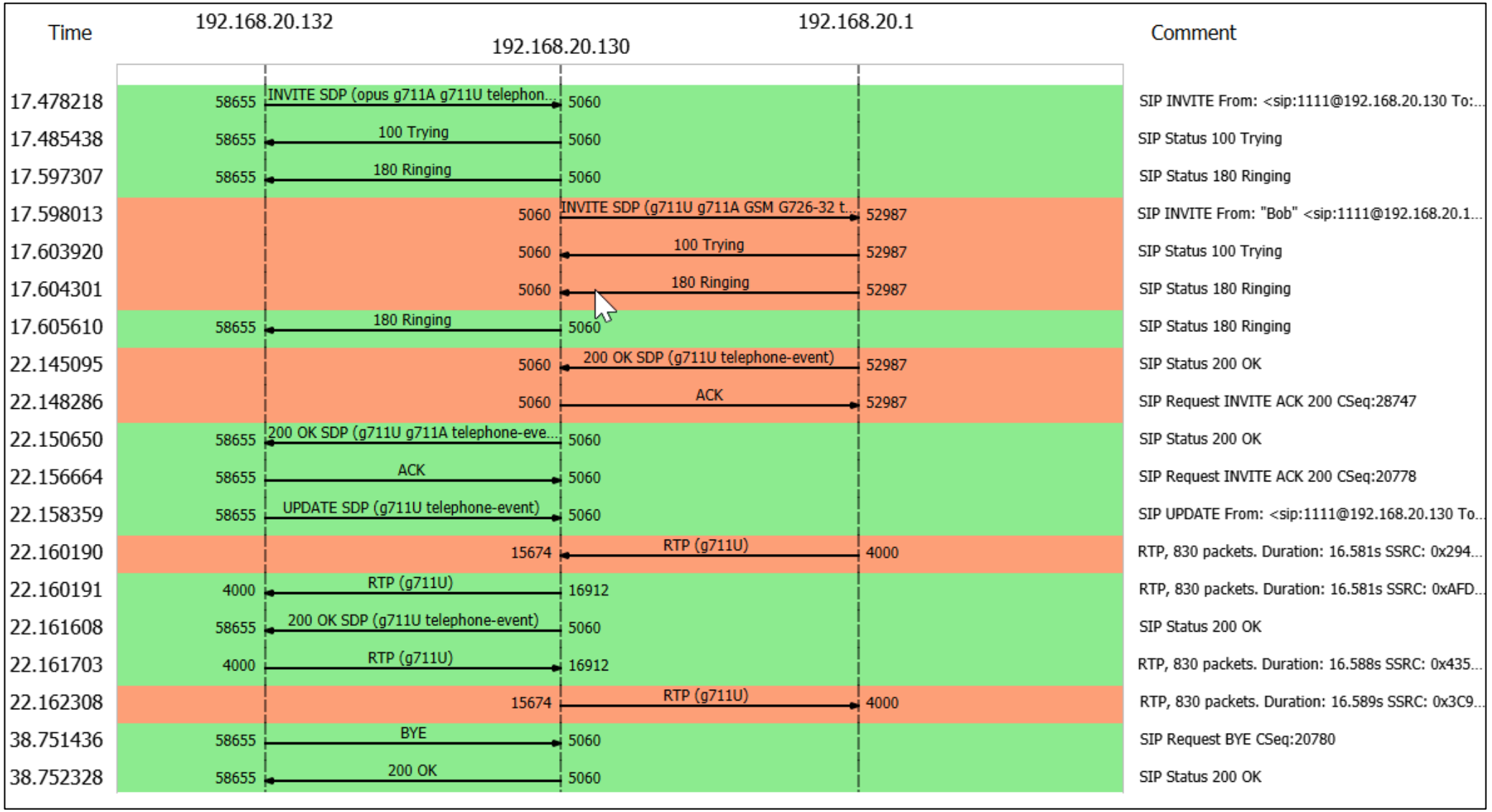
©PentesterAcademy.com

复原VoIP通话



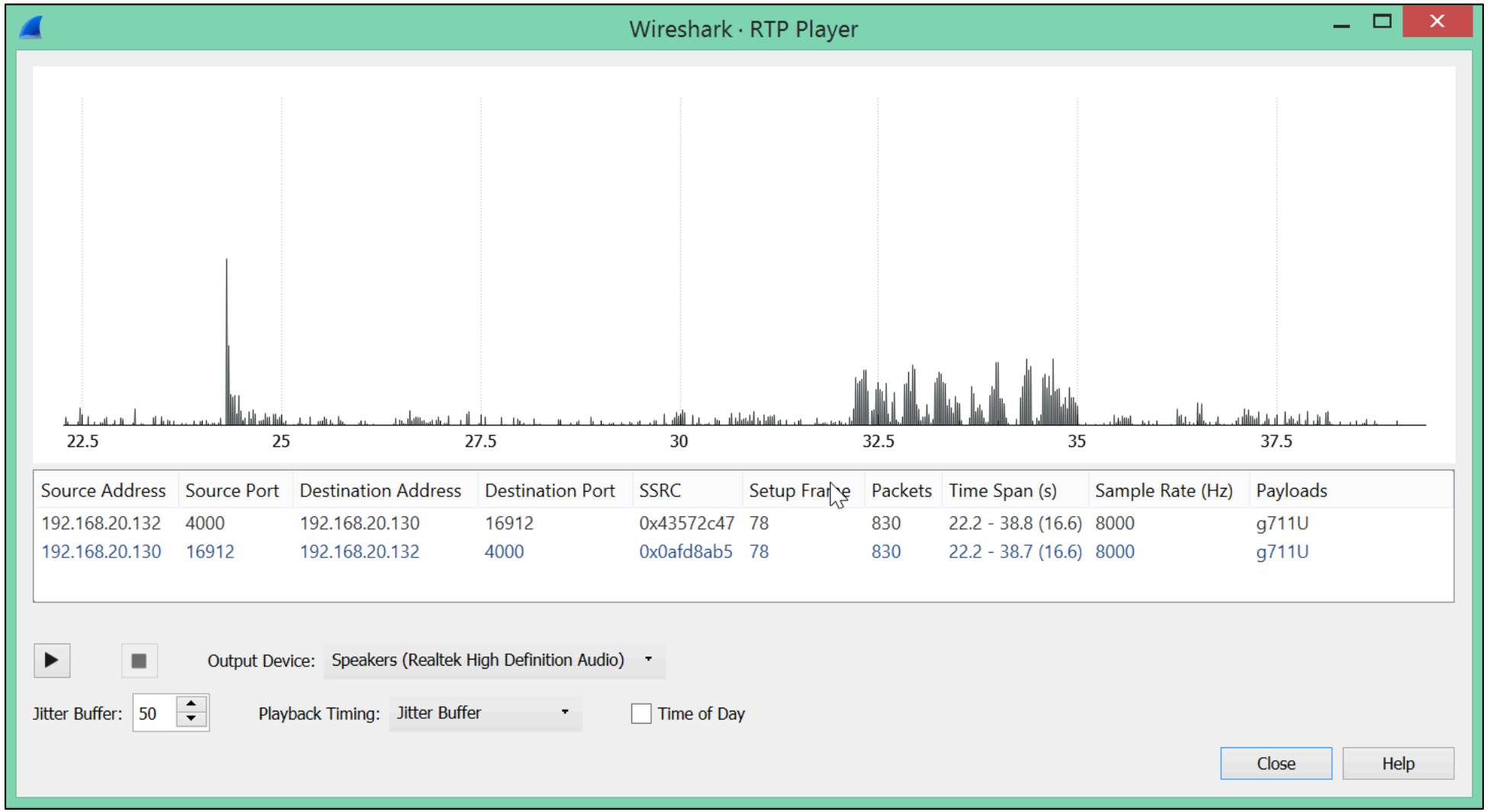
©PentesterAcademy.com

通话流



©PentesterAcademy.com

通话重建



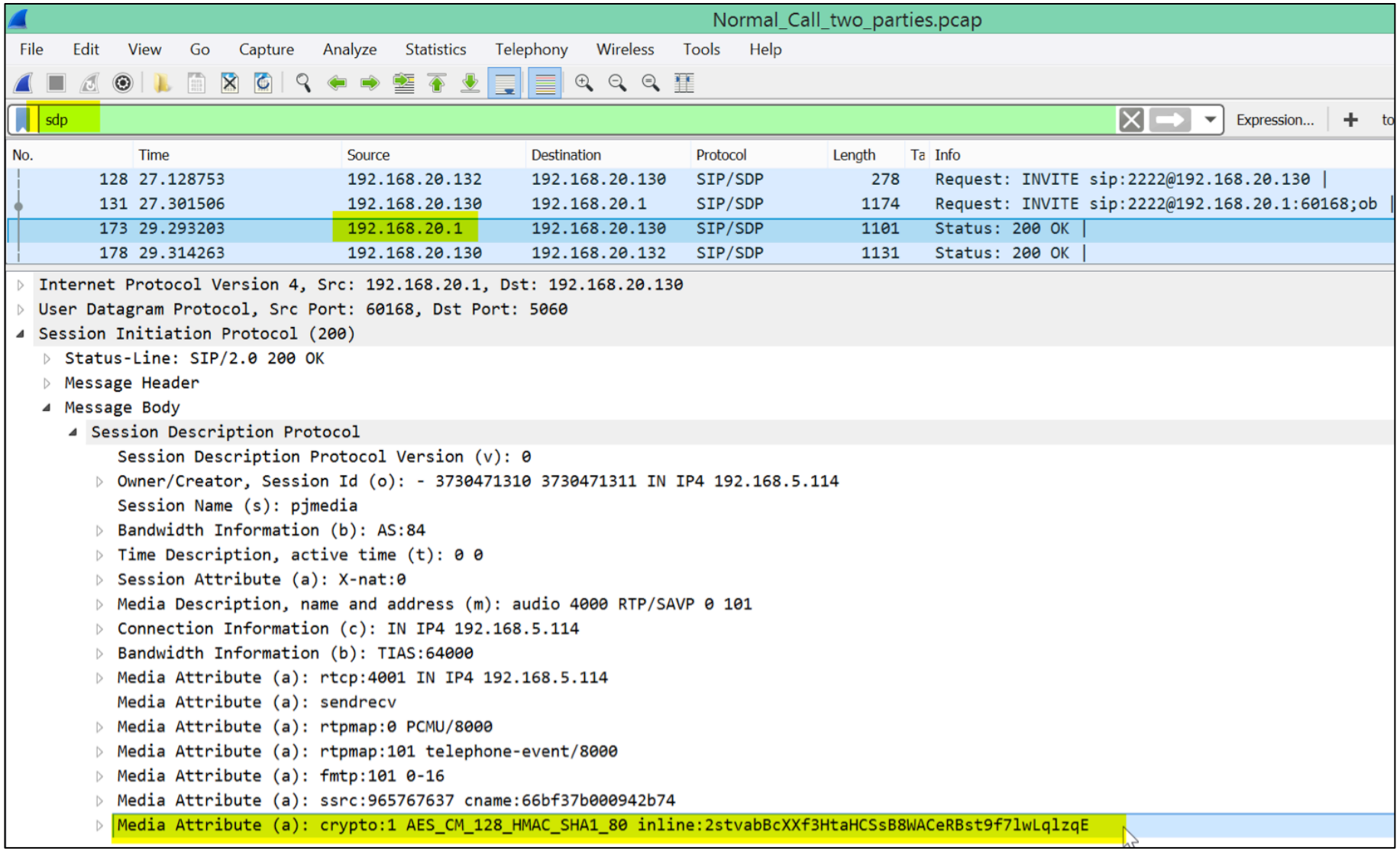
©PentesterAcademy.com

配置方案

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

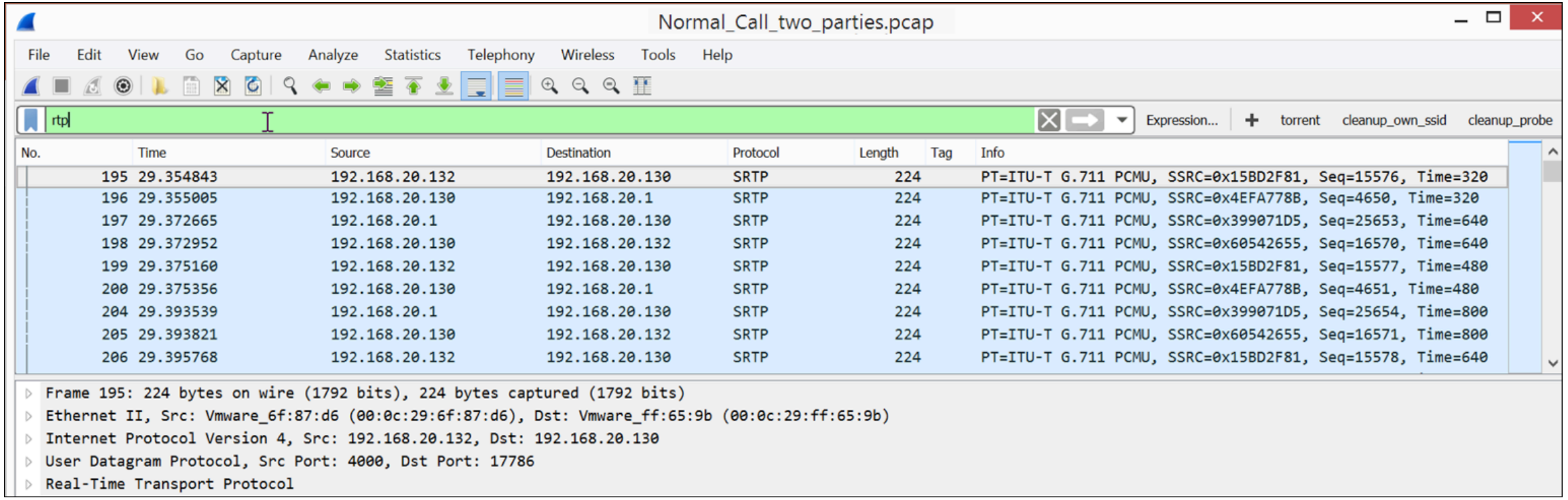
©PentesterAcademy.com

在SDP数据包中传输SRTP密钥



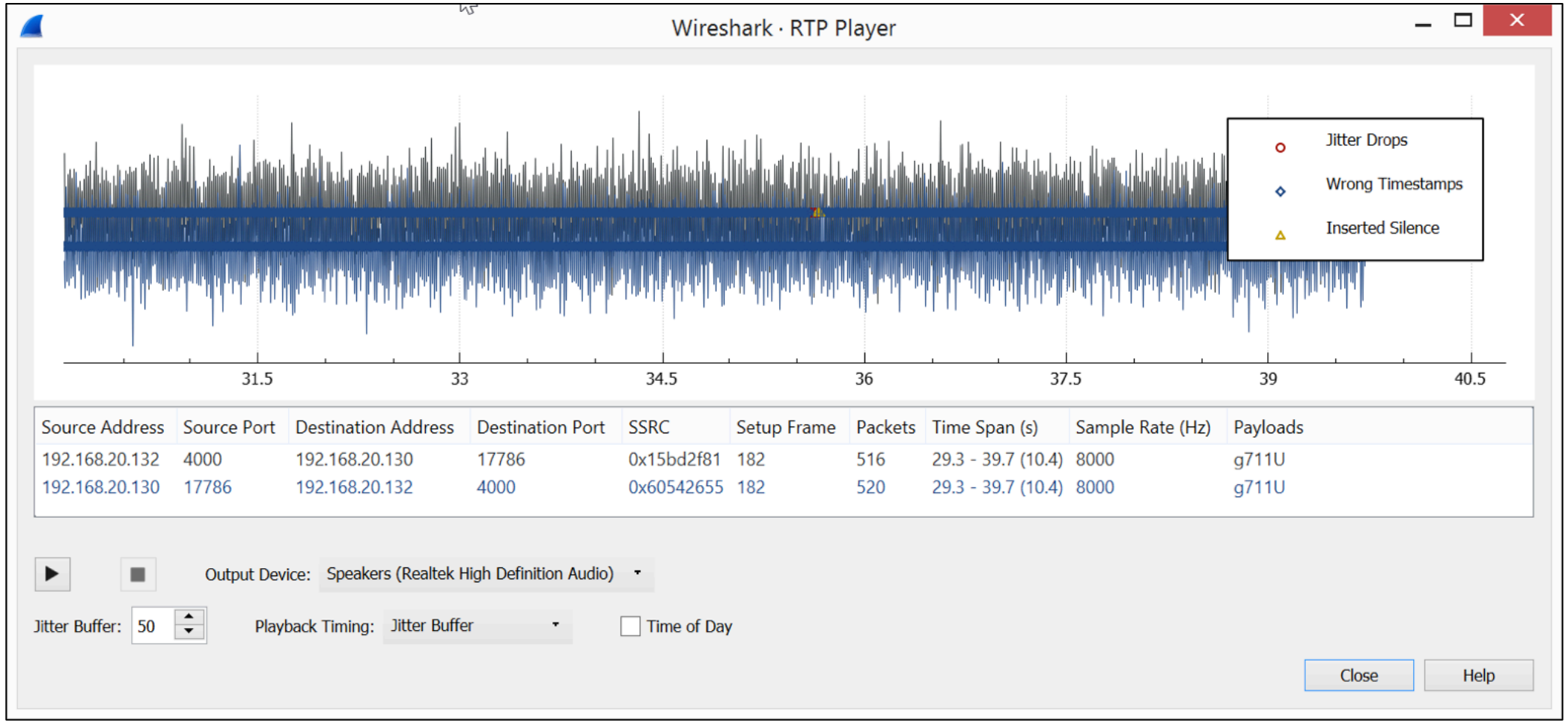
©PentesterAcademy.com

SRTP流量



©PentesterAcademy.com

加密后的通话



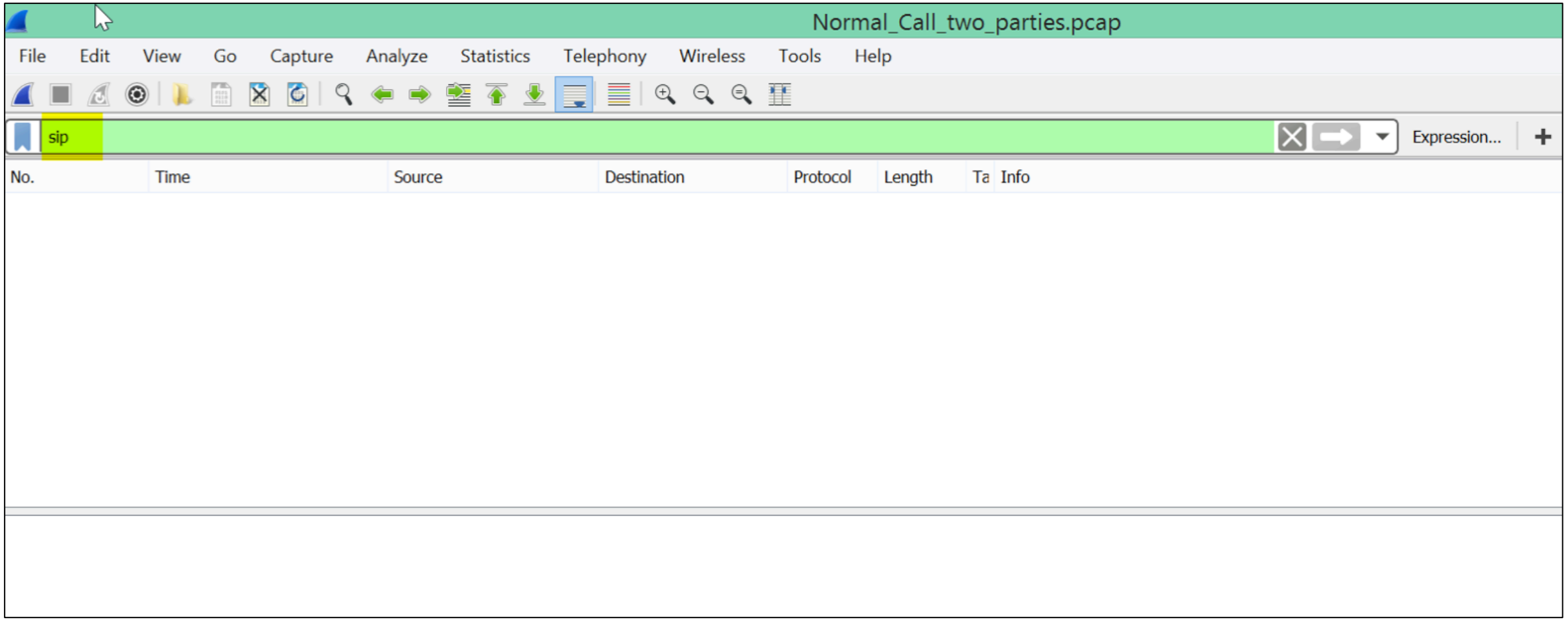
©PentesterAcademy.com

配置方案

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

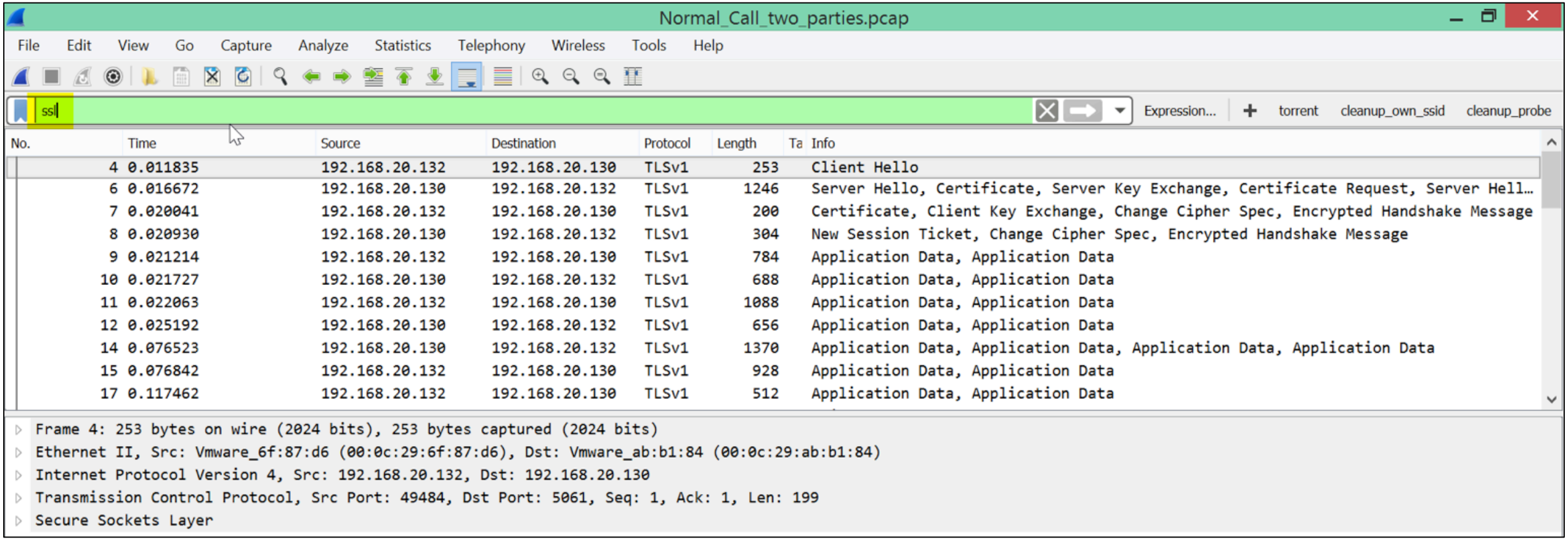
©PentesterAcademy.com

无SIP流量



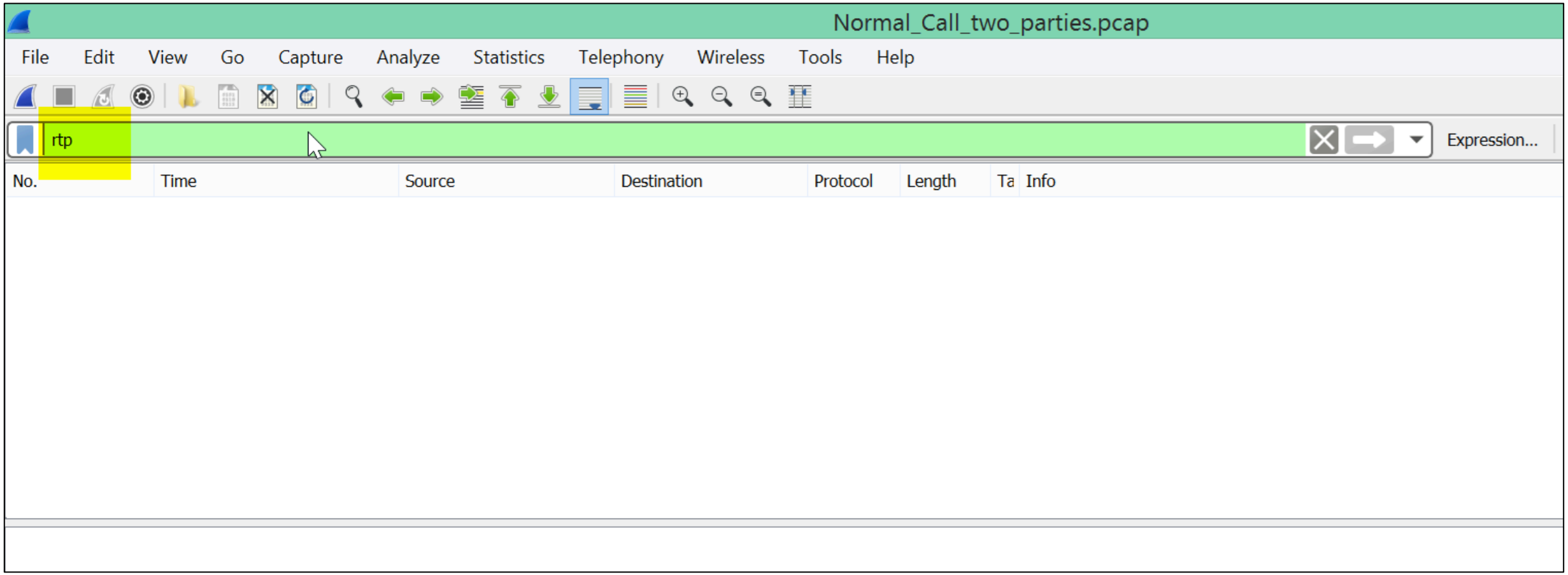
©PentesterAcademy.com

TLS流量(SIP over TLS)



©PentesterAcademy.com

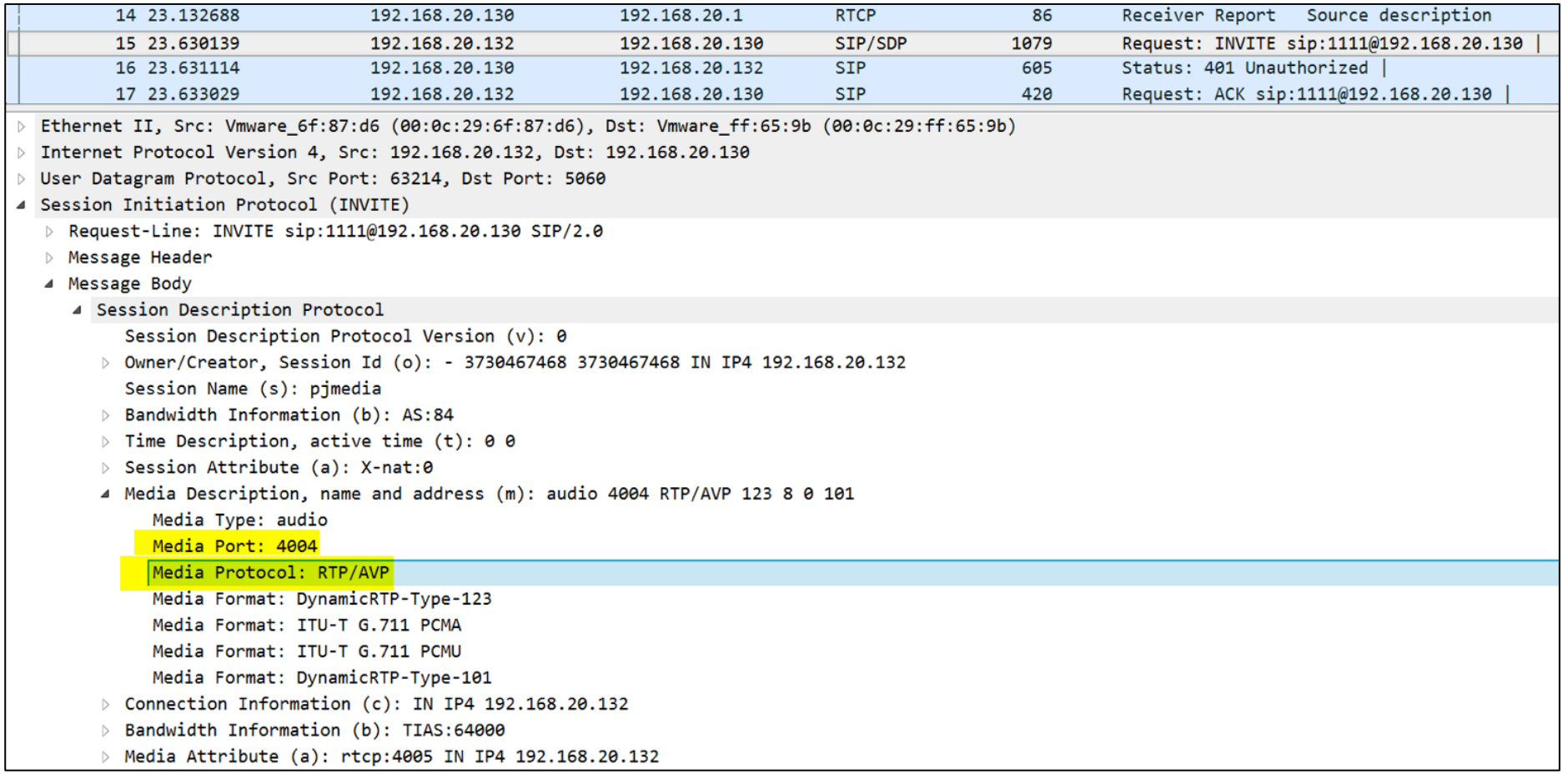
无RTP流量



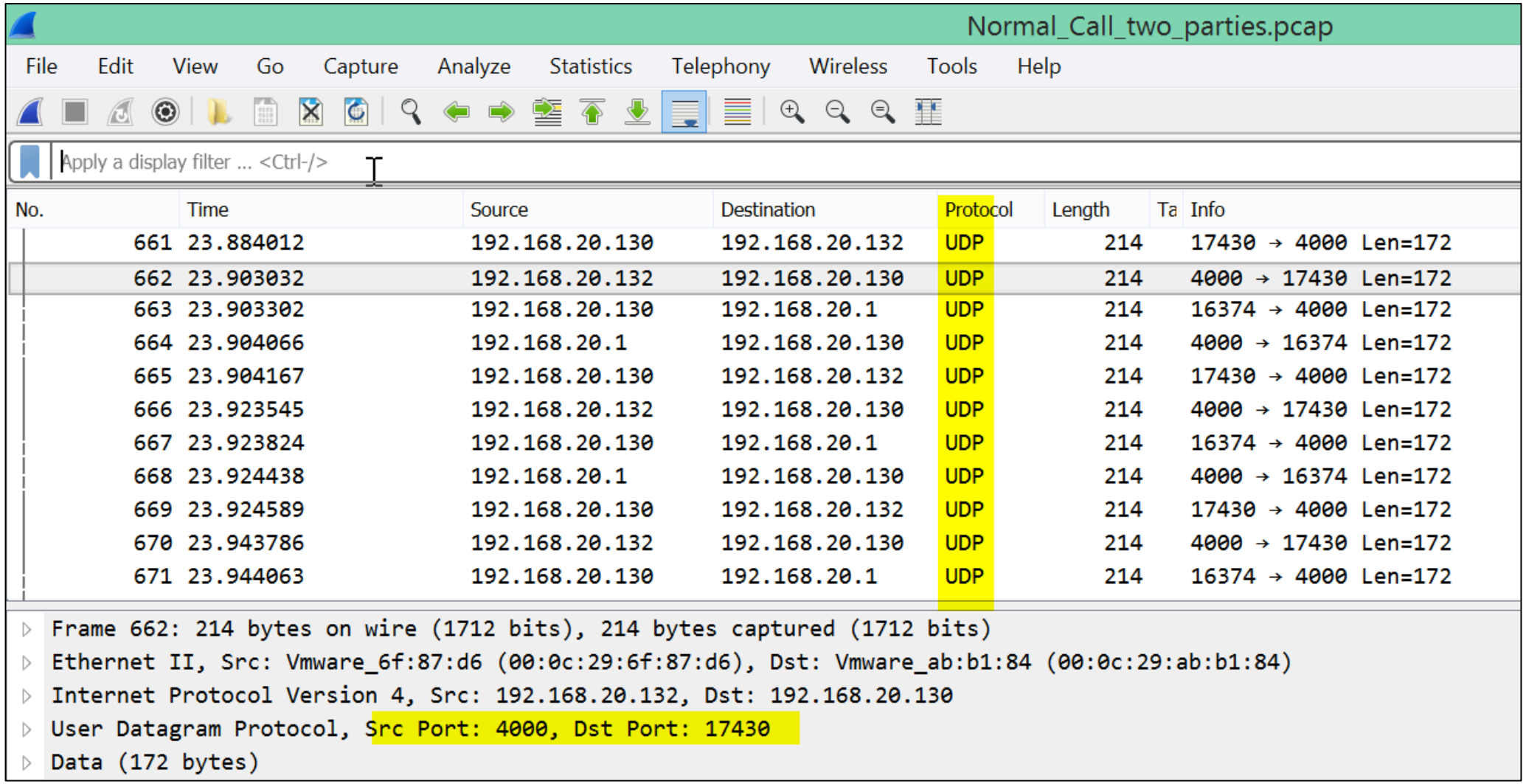
©PentesterAcademy.com

为什么没有RTP流量?

- Wireshark通过SDP数据包得到RTP/SRTP流使用的端口号
- SIP和SDP被加密, 故wireshark无法得知端口号

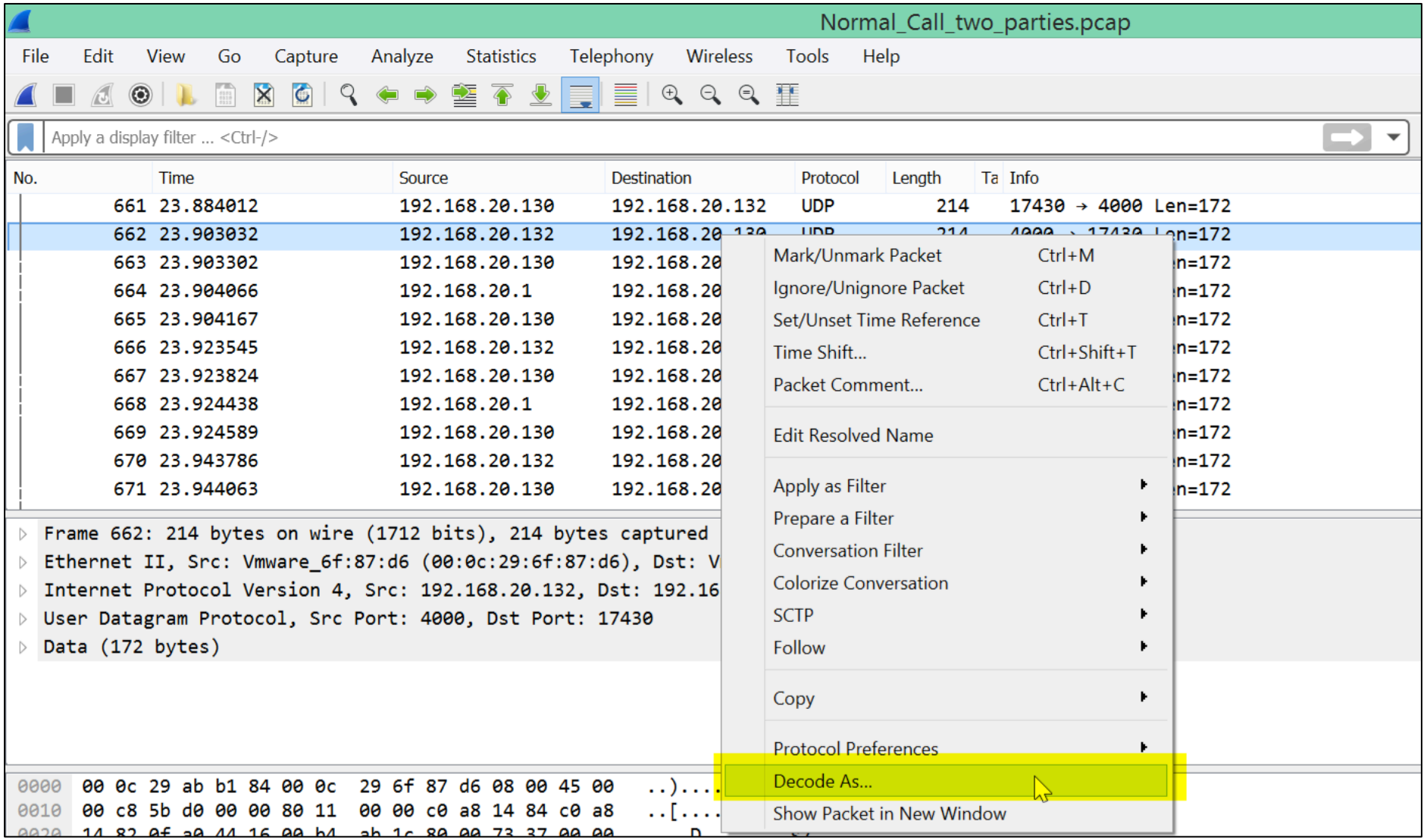


未解码的RTP流量



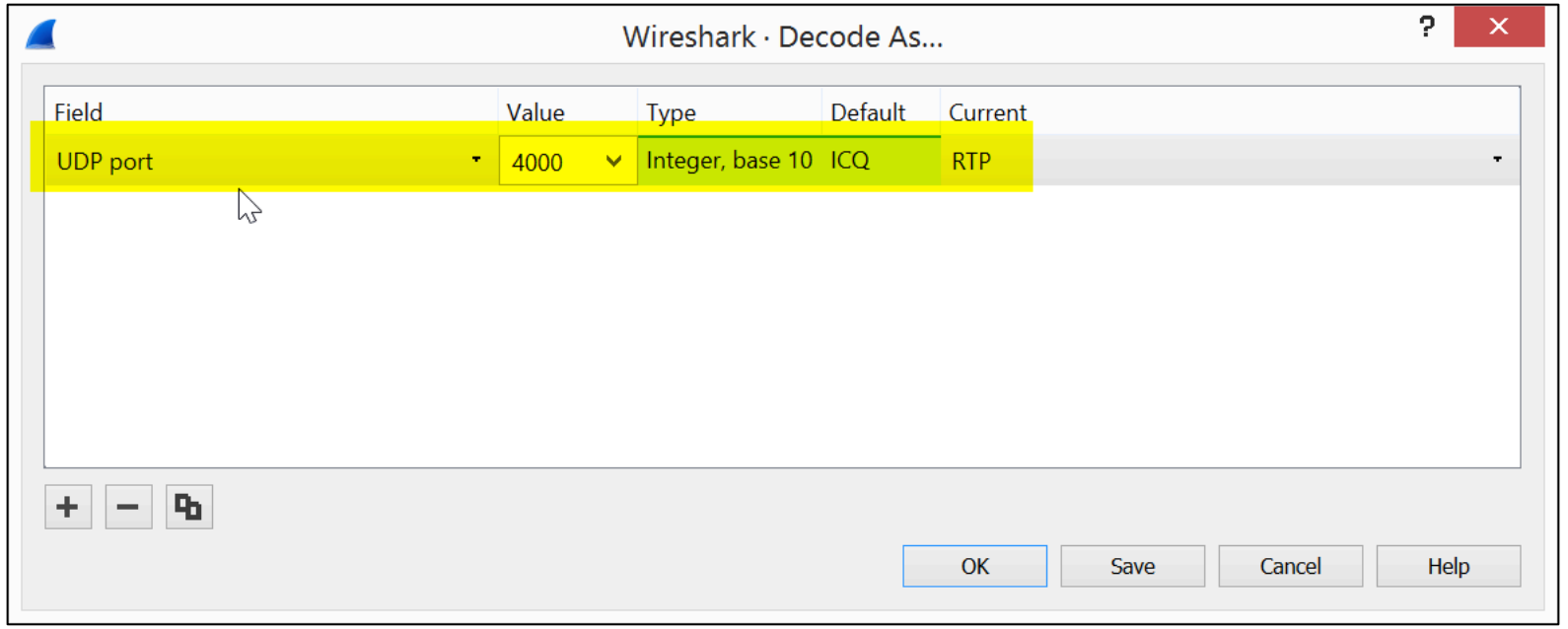
©PentesterAcademy.com

解码



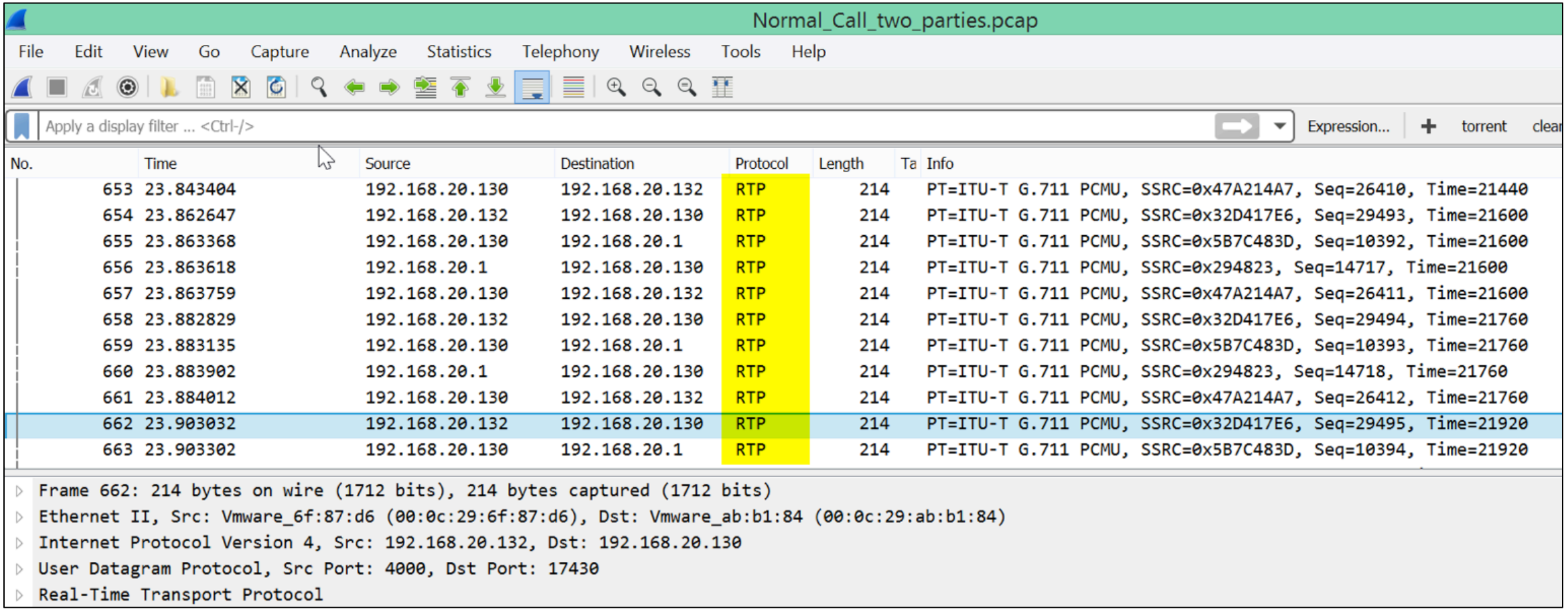
©PentesterAcademy.com

解码为RTP



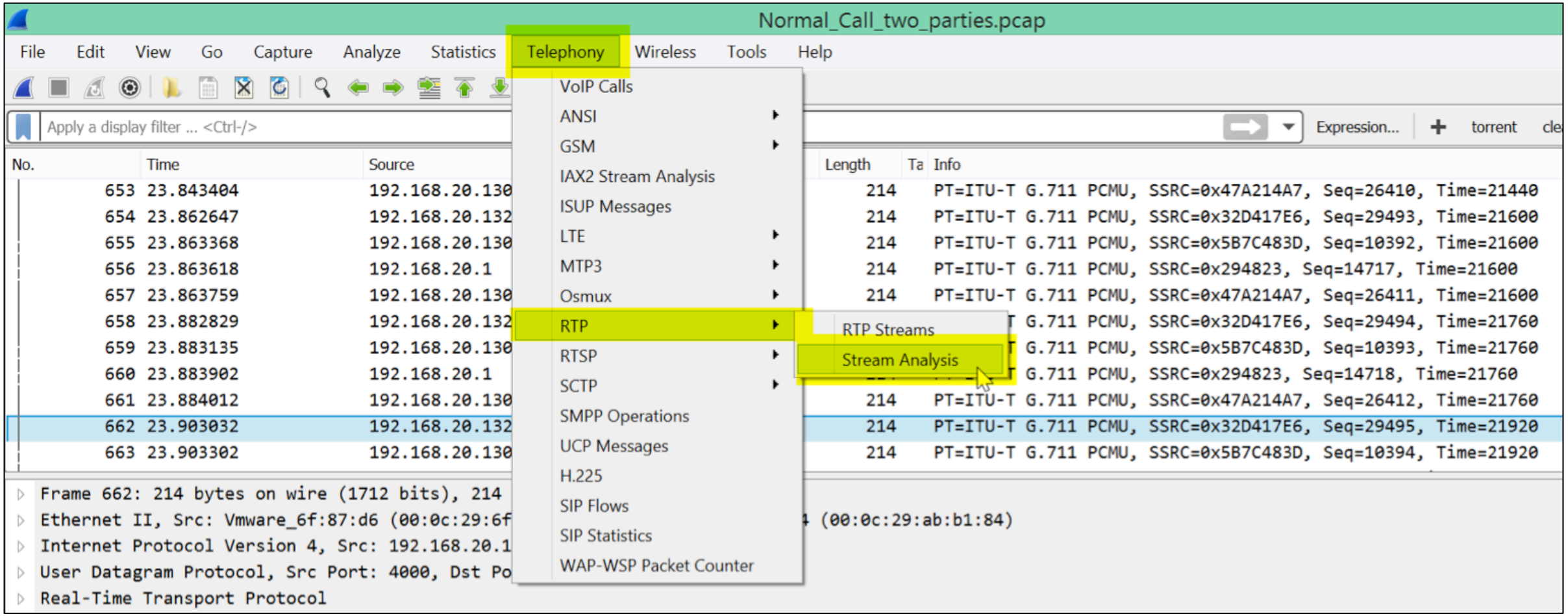
©PentesterAcademy.com

RTP流量



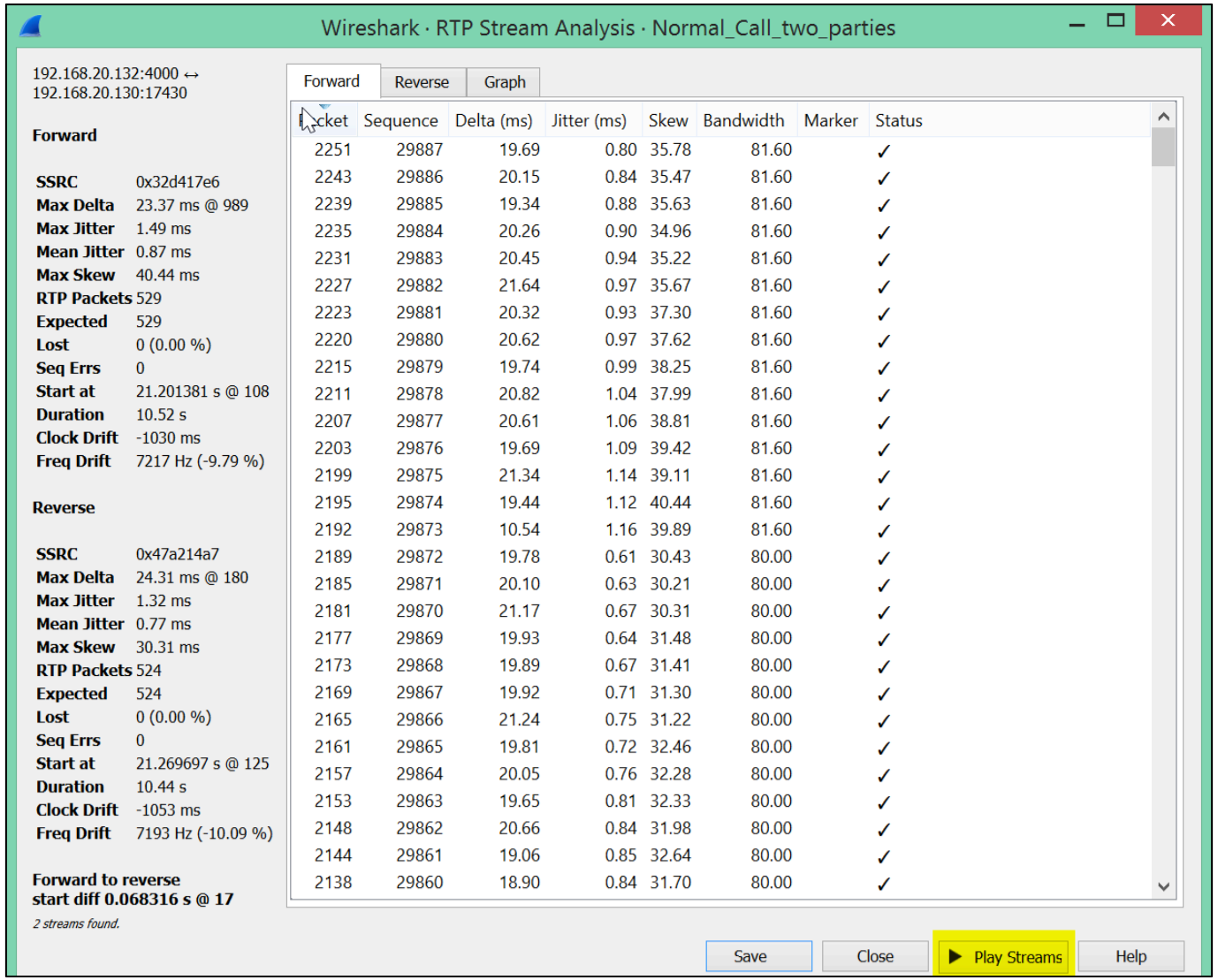
©PentesterAcademy.com

检查RTP流



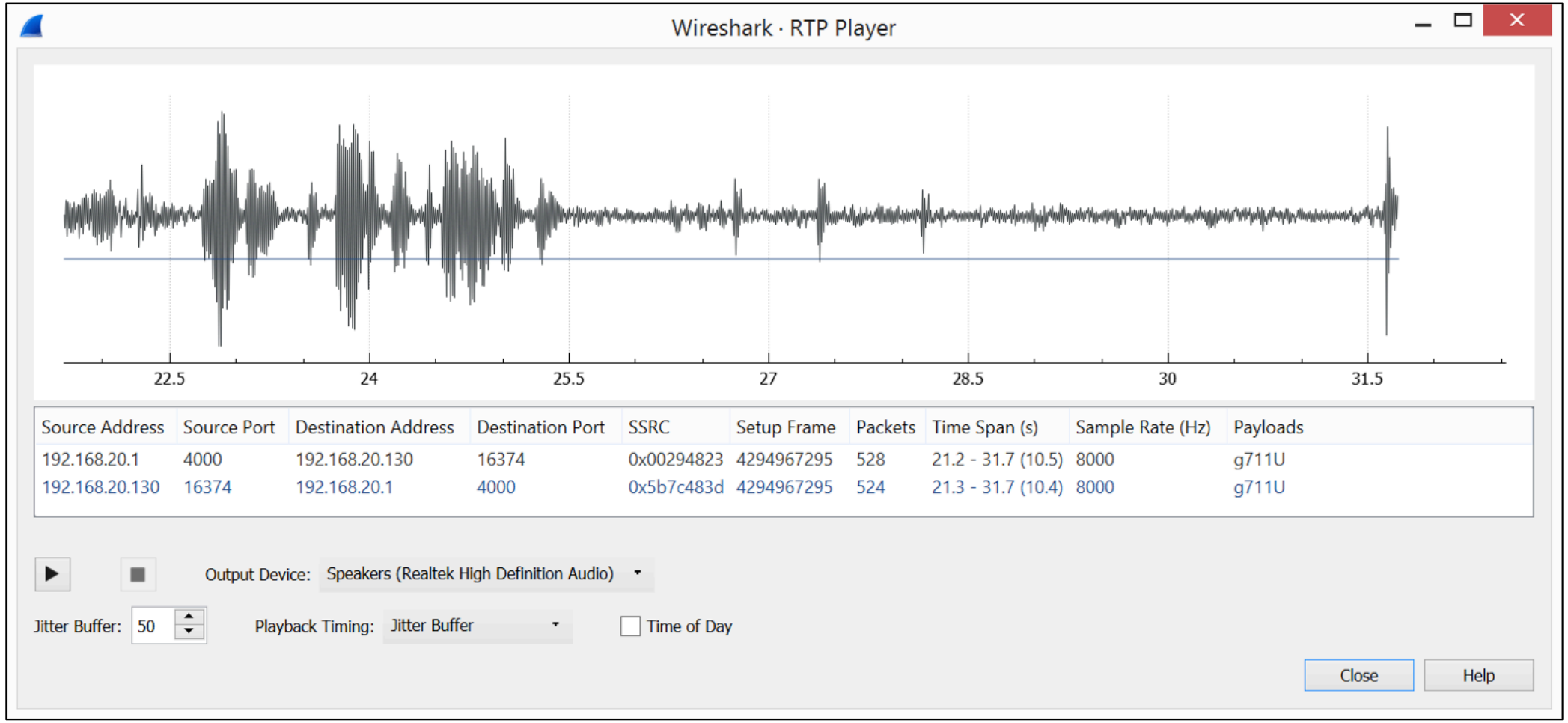
©PentesterAcademy.com

分析RTP流



©PentesterAcademy.com

播放RTP流



©PentesterAcademy.com

配置方案

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

©PentesterAcademy.com

TLS密钥交换方法

- TLS 使用对称加密算法 (如AES, Blowfish) 加密数据

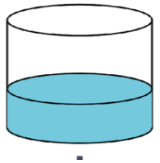
- **两种可行方法**
  - DHE (Diffie Hellman密钥交换)
  - RSA (非对称加密)

©PentesterAcademy.com

Diffie Hellman密钥交换

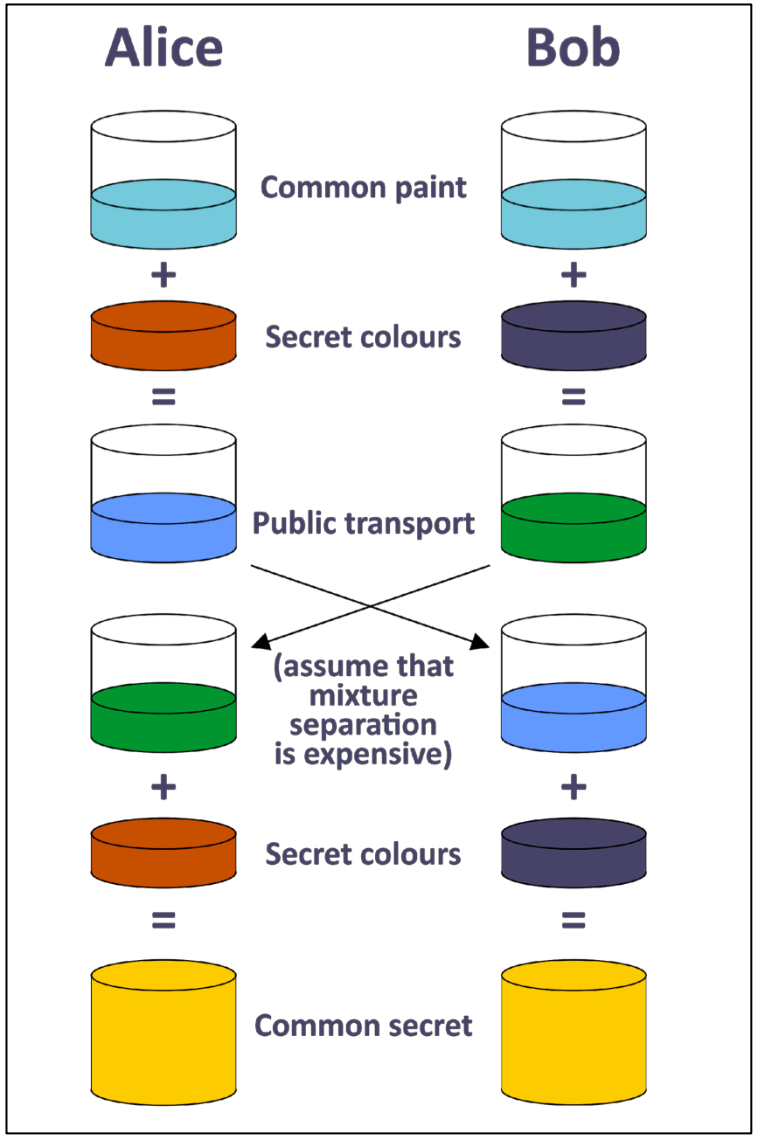
假设

- 攻击者即使看到交换过程也无法猜测原始颜色
- 攻击者可以看到 

  以及 

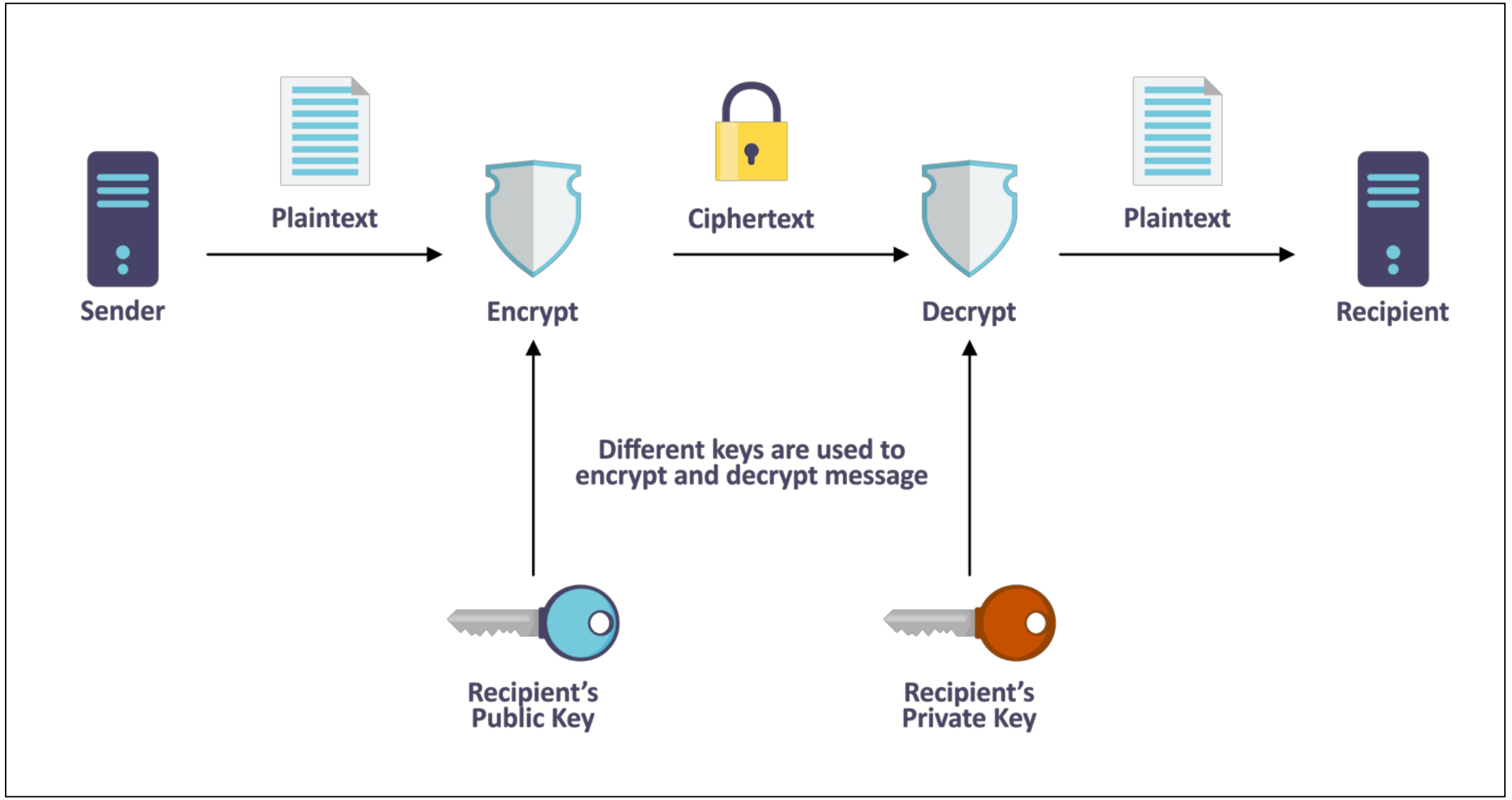
  但不知道添加的是什么颜色

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



©PentesterAcademy.com

RSA (非对称加密)



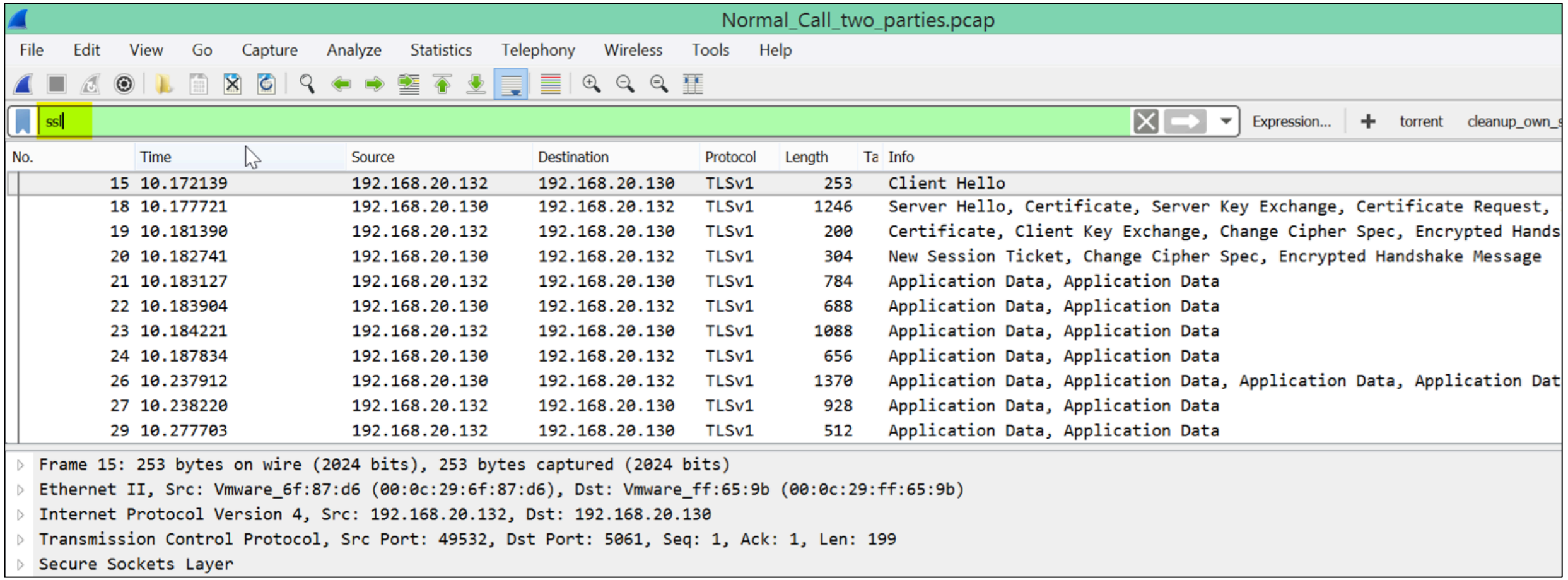
©PentesterAcademy.com

现状

- 无法通过监听流量恢复**ECDHE/DHE**派生的密钥
- 对于**RSA**，获取到服务器私钥即可解密流量

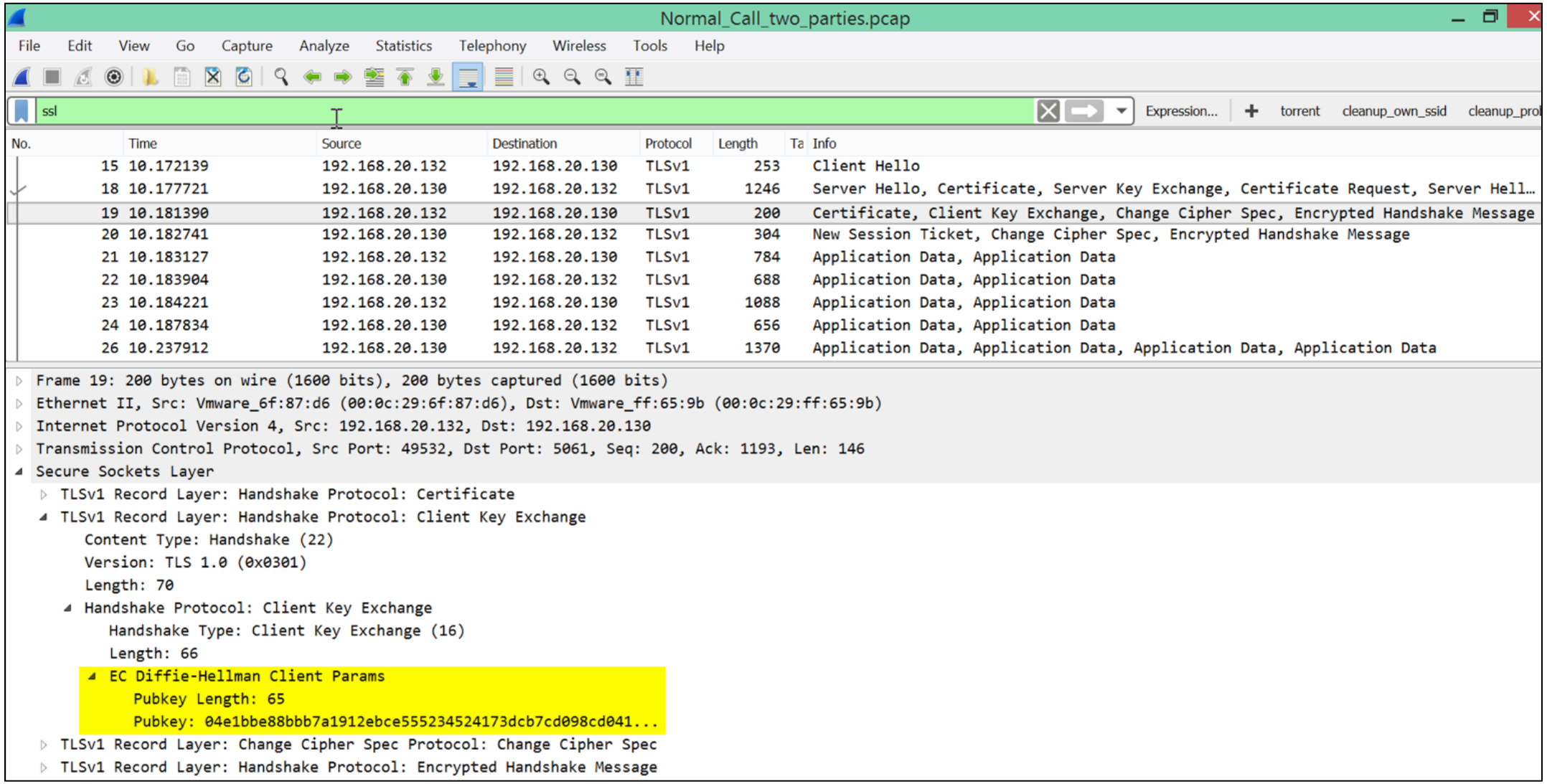
©PentesterAcademy.com

TLS流量(SIP over TLS)



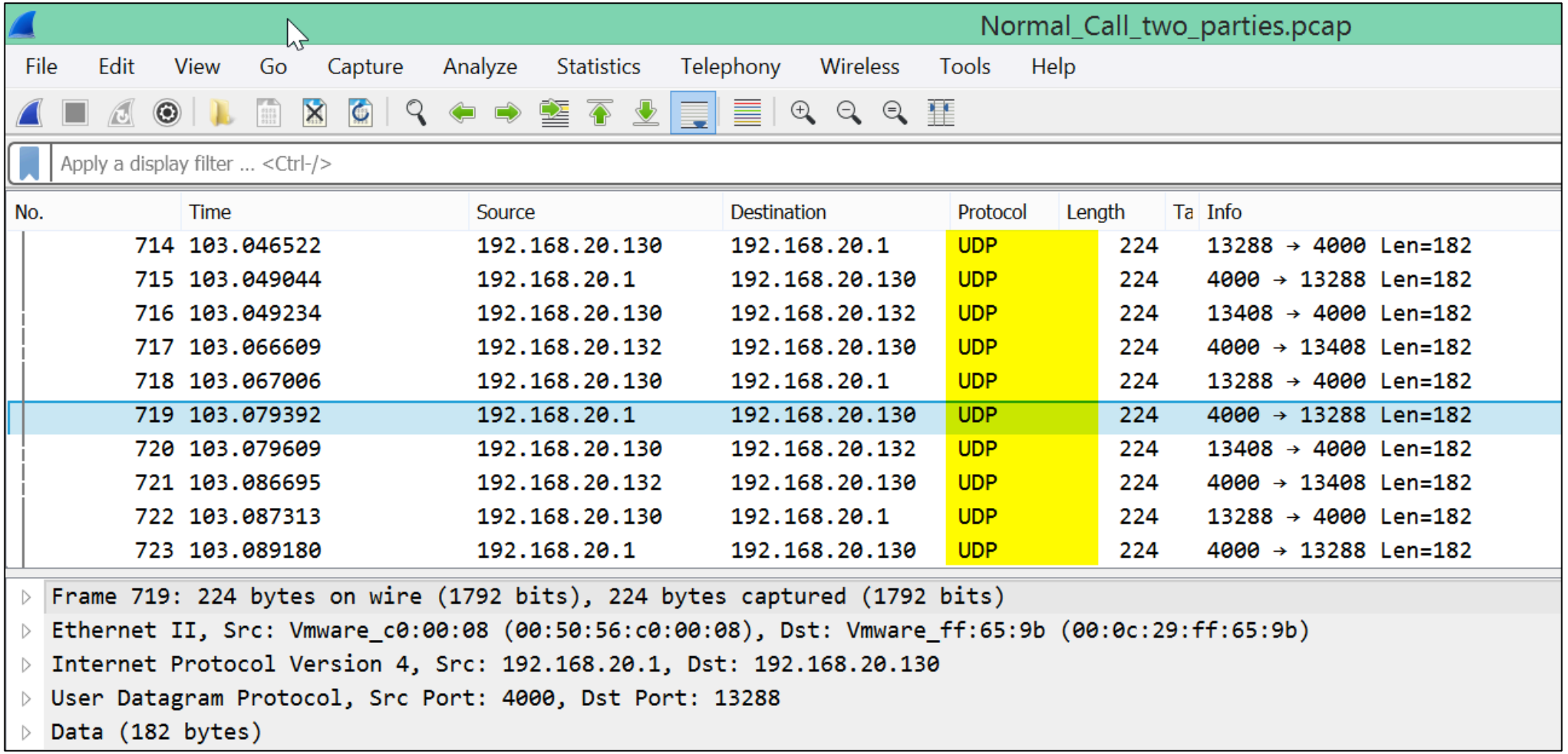
©PentesterAcademy.com

Diffie Hellman密钥交换



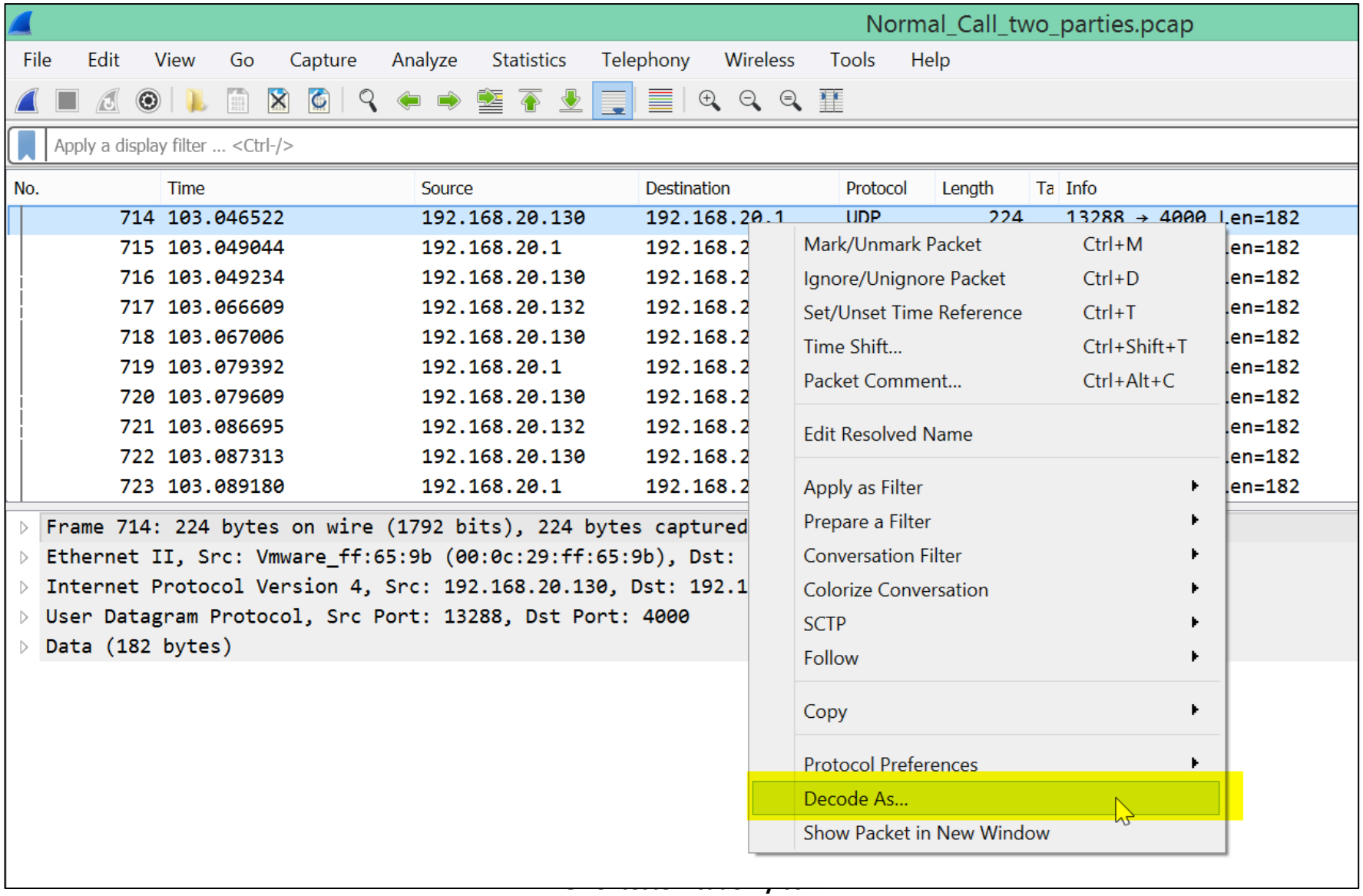
©PentesterAcademy.com

未解码的SRTP流量

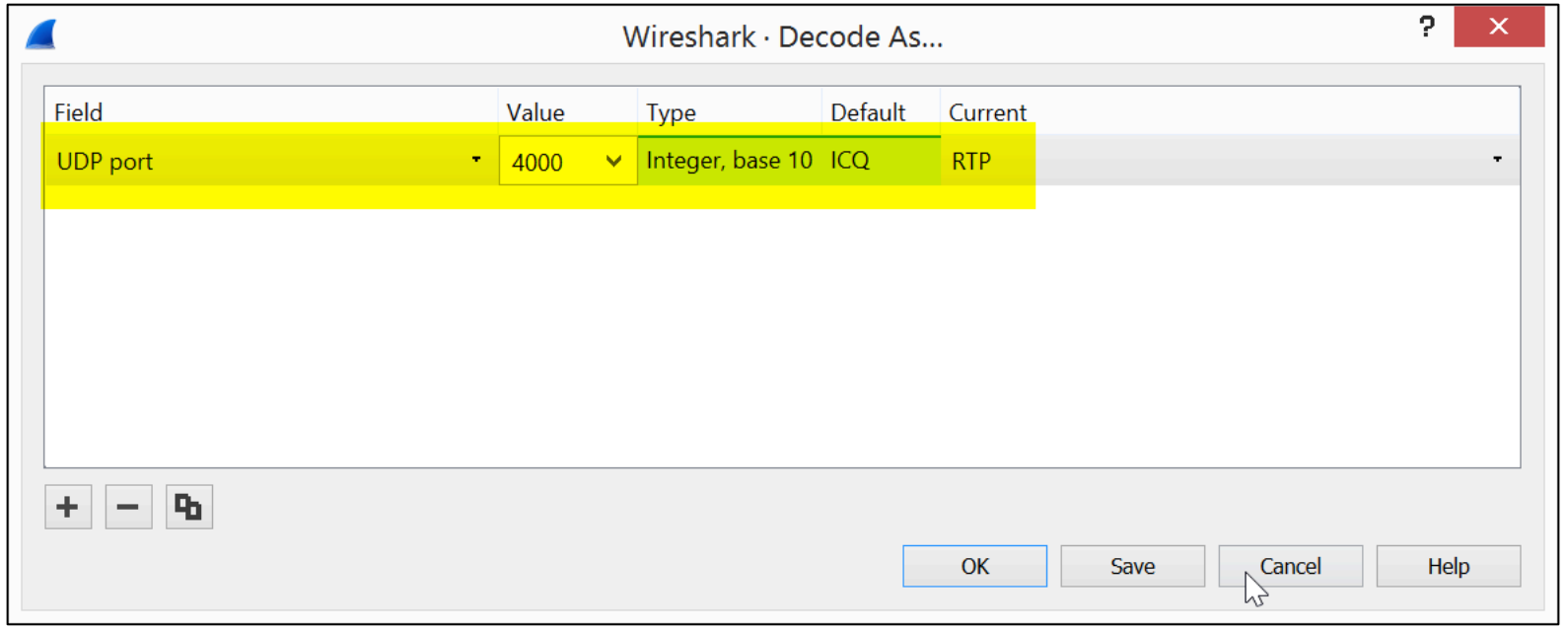


©PentesterAcademy.com

解码

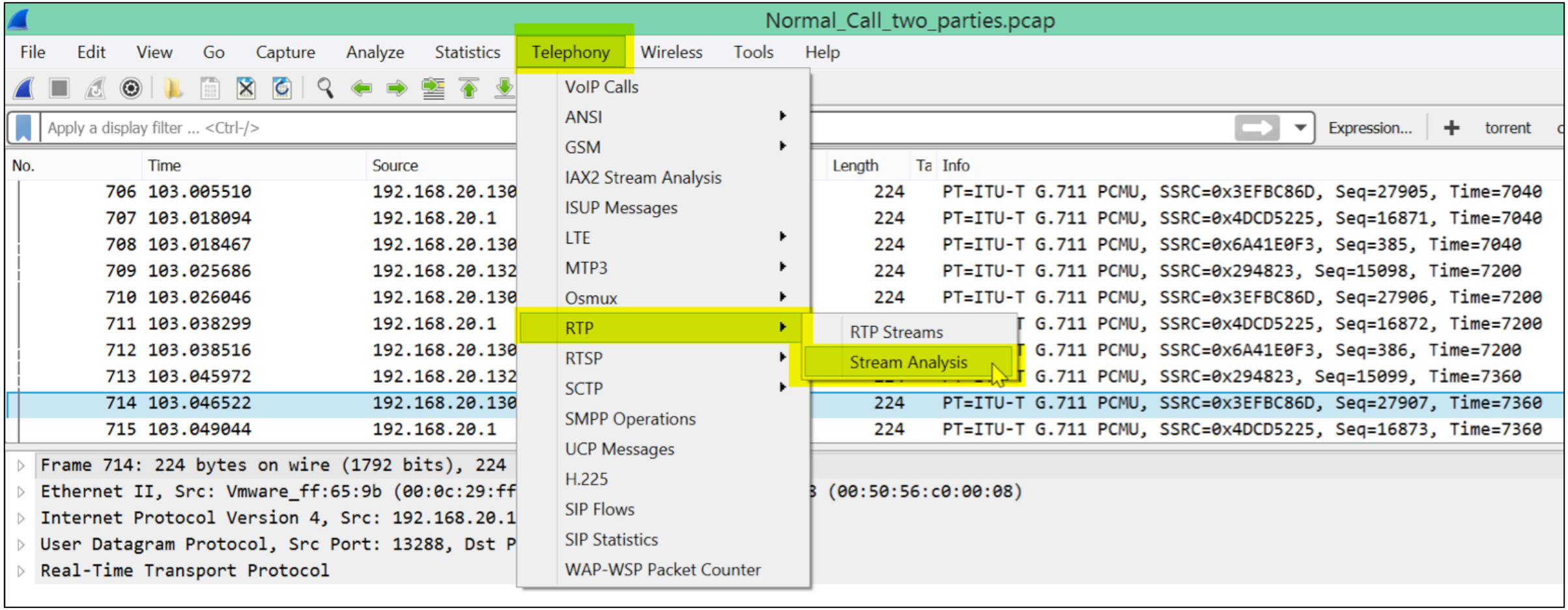

Normal_Call_two_parties.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Ta Info
714 103.046522 192.168.20.130 192.168.20.1 UDP 224 13288 → 4000 Len=182
715 103.049044 192.168.20.1 192.168.2 Len=182
716 103.049234 192.168.20.130 192.168.2 Len=182
717 103.066609 192.168.20.132 192.168.2 Len=182
718 103.067006 192.168.20.130 192.168.2 Len=182
719 103.079392 192.168.20.1 192.168.2 Len=182
720 103.079609 192.168.20.130 192.168.2 Len=182
721 103.086695 192.168.20.132 192.168.2 Len=182
722 103.087313 192.168.20.130 192.168.2 Len=182
723 103.089180 192.168.20.1 192.168.2 Len=182
Frame 714: 224 bytes on wire (1792 bits), 224 bytes captured
Ethernet II, Src: Vmware_ff:65:9b (00:0c:29:ff:65:9b), Dst:
Internet Protocol Version 4, Src: 192.168.20.130, Dst: 192.1
User Datagram Protocol, Src Port: 13288, Dst Port: 4000
Data (182 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

解码为RTP



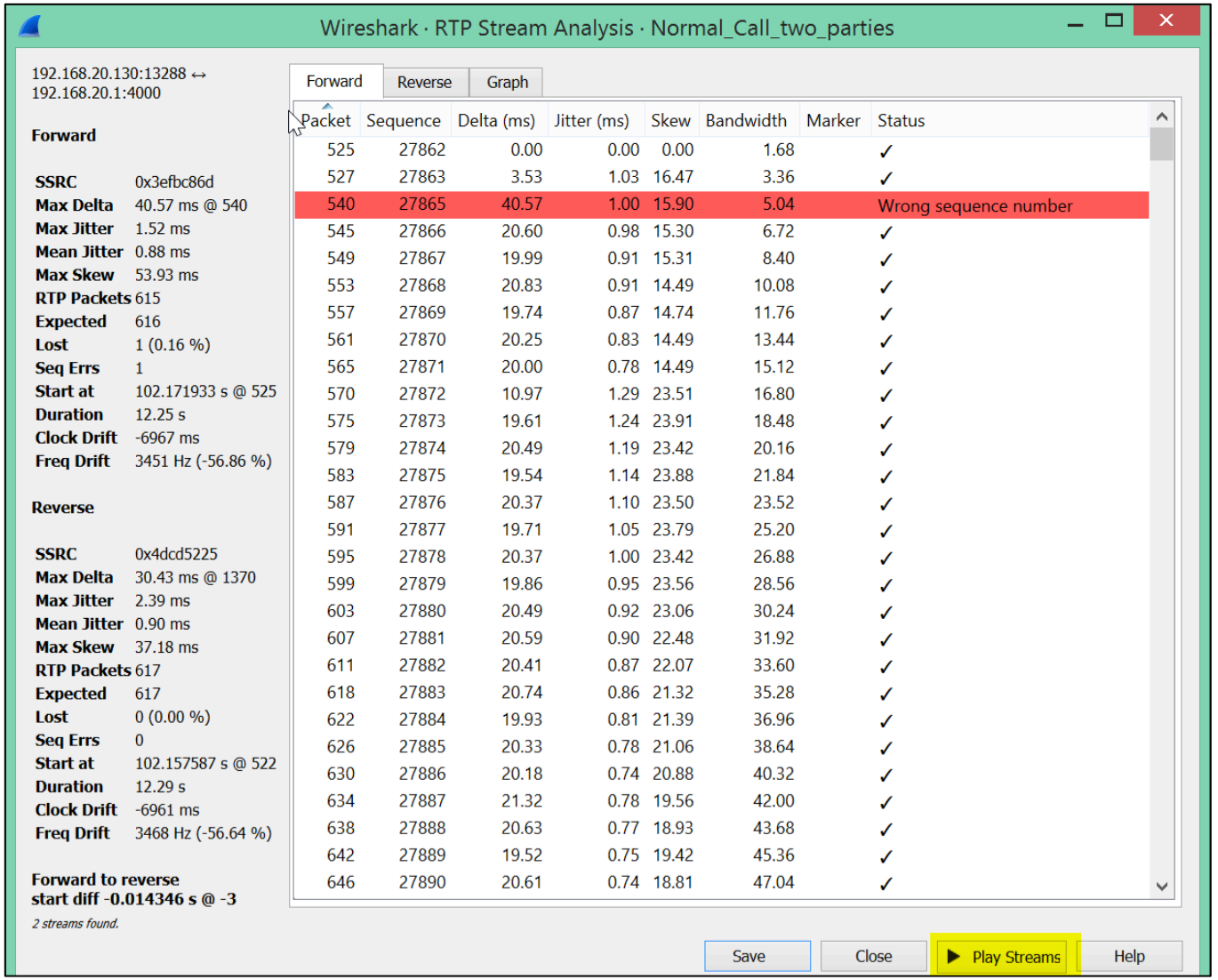
©PentesterAcademy.com

检查RTP流



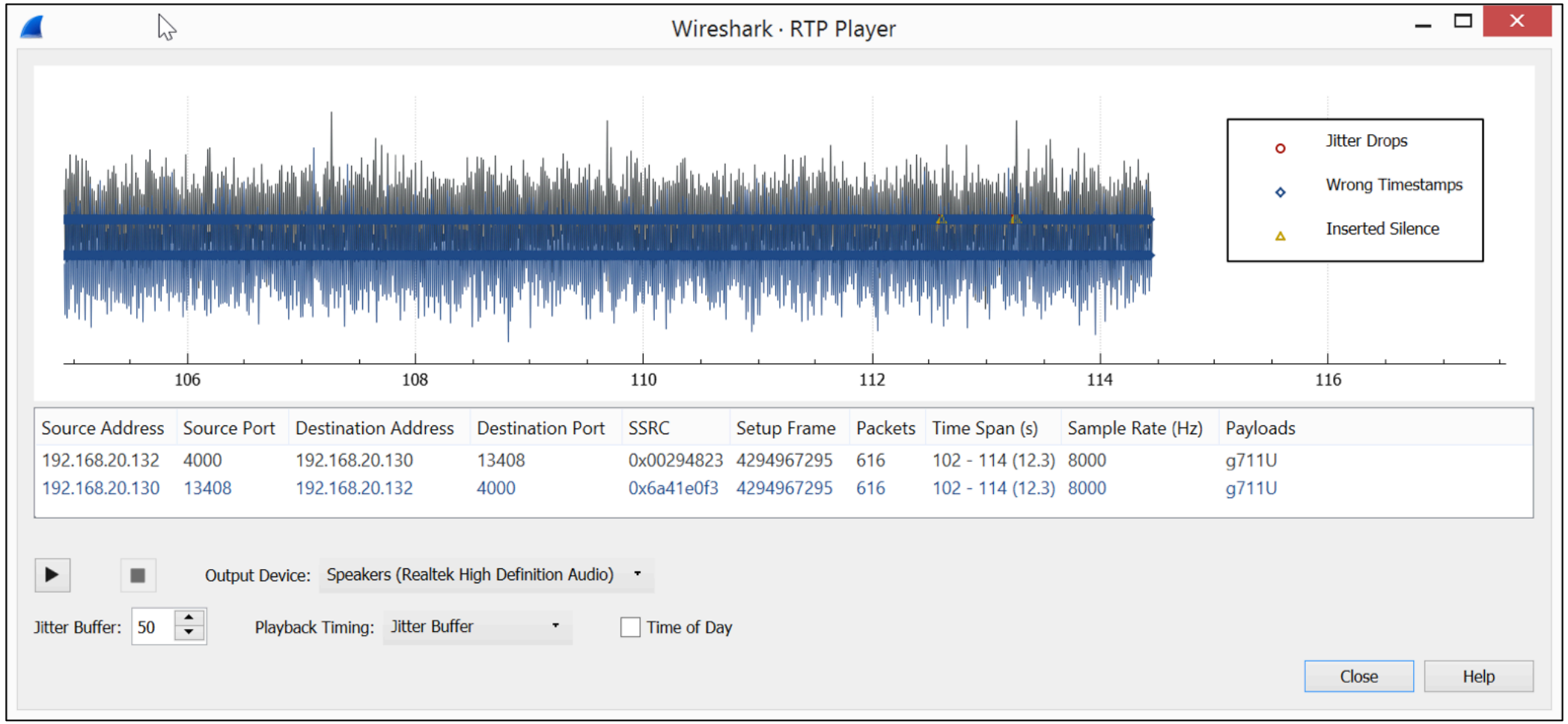
©PentesterAcademy.com

分析RTP流



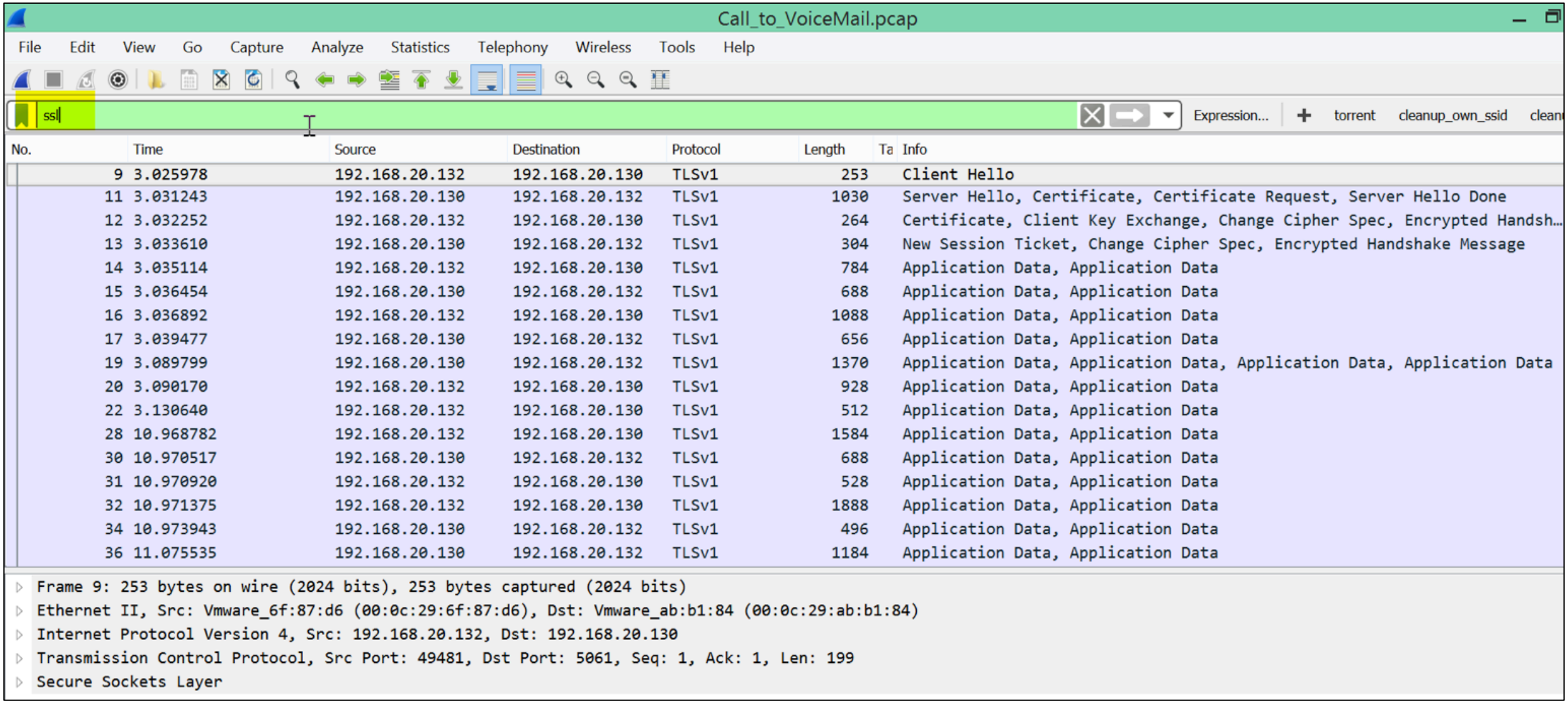
©PentesterAcademy.com

播放RTP流



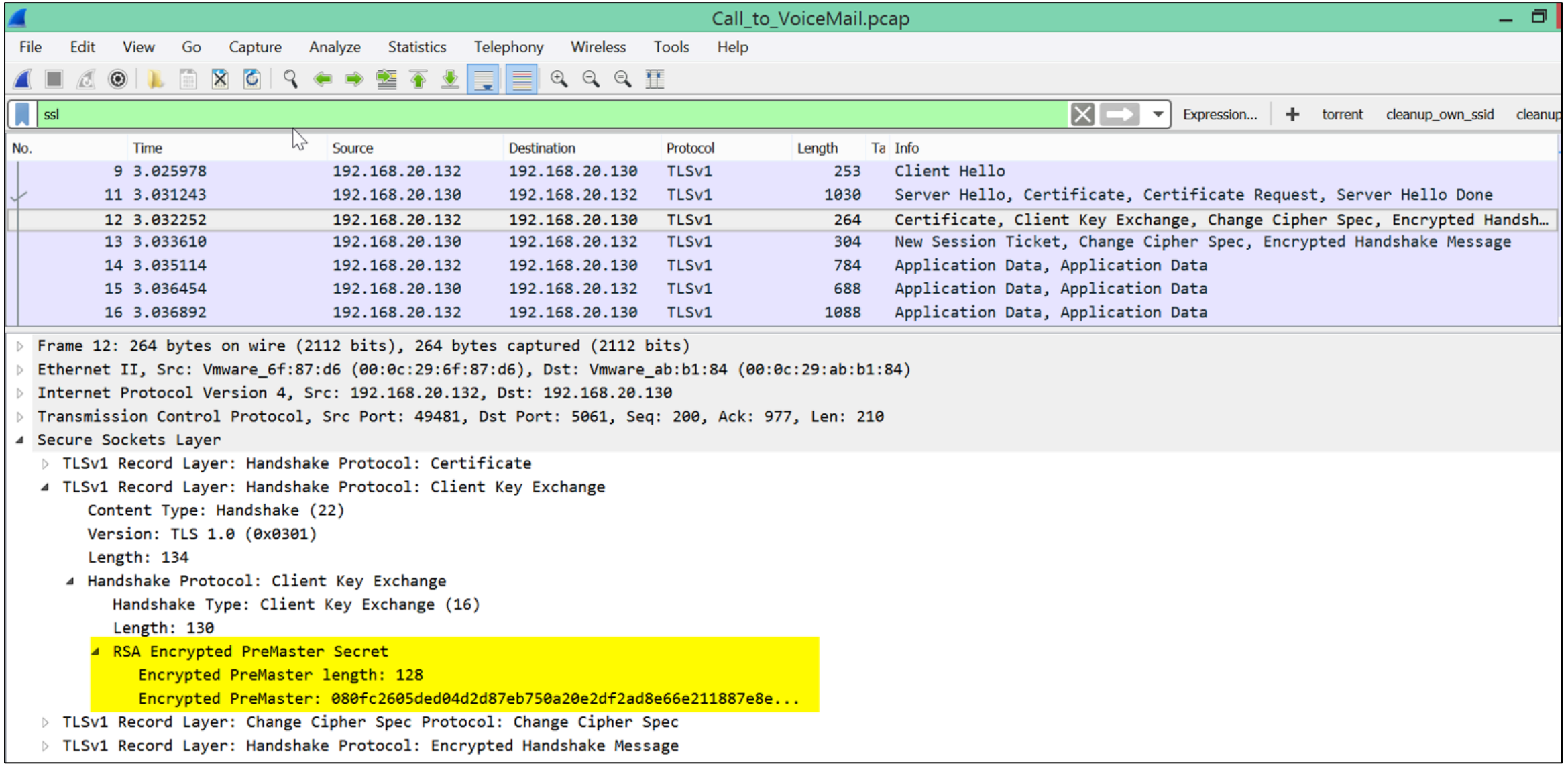
©PentesterAcademy.com

TLS 流量(SIP over TLS)



©PentesterAcademy.com

基于RSA的密钥交换



©PentesterAcademy.com

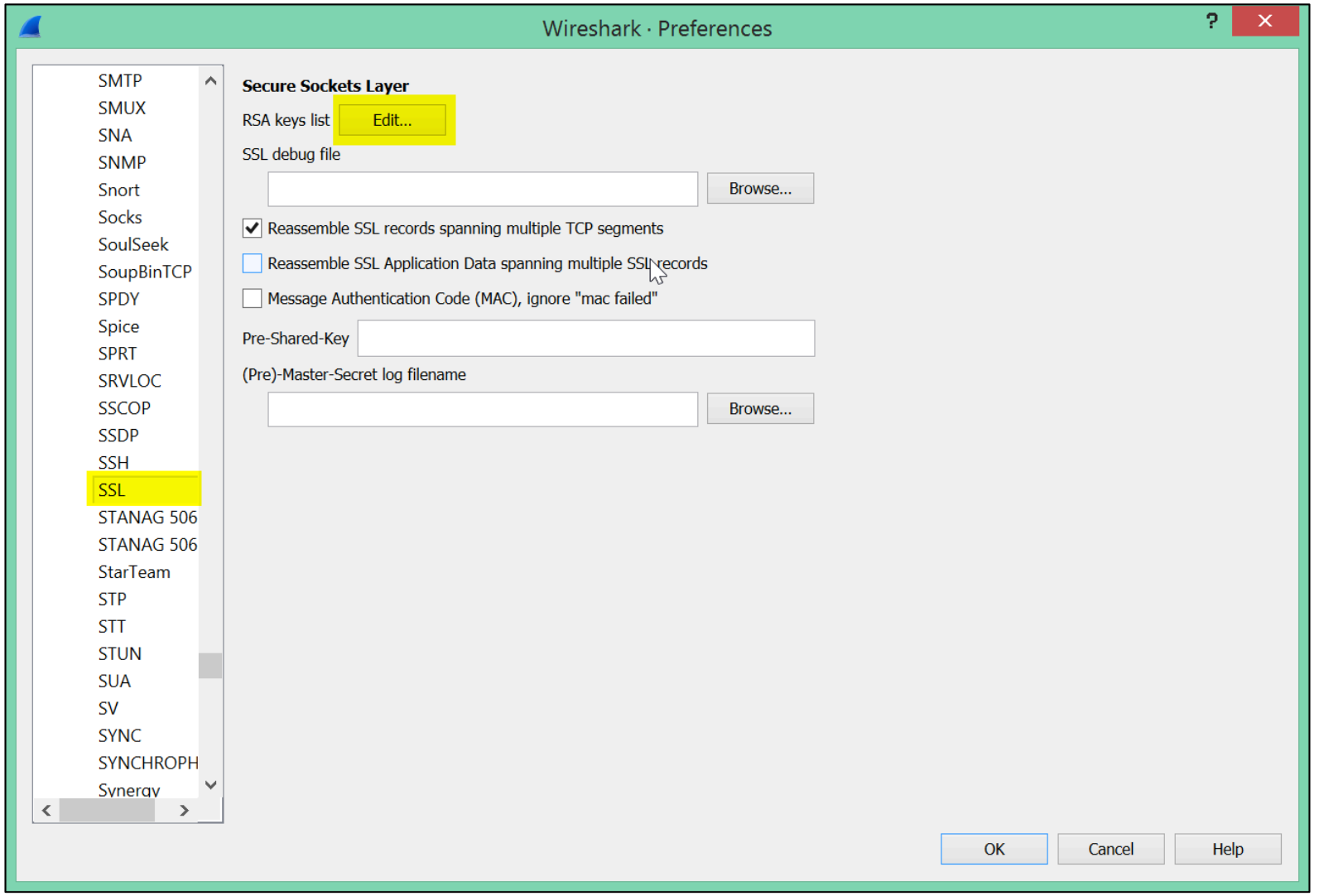
解密TLS流量

- 使用RSA交换密钥
- 可解密安装在Asterisk One上的密钥
- **Asterisk One密钥和证书的位置:** /etc/asterisk/keys
- 必须从服务器得到**default.key**

```
[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
```

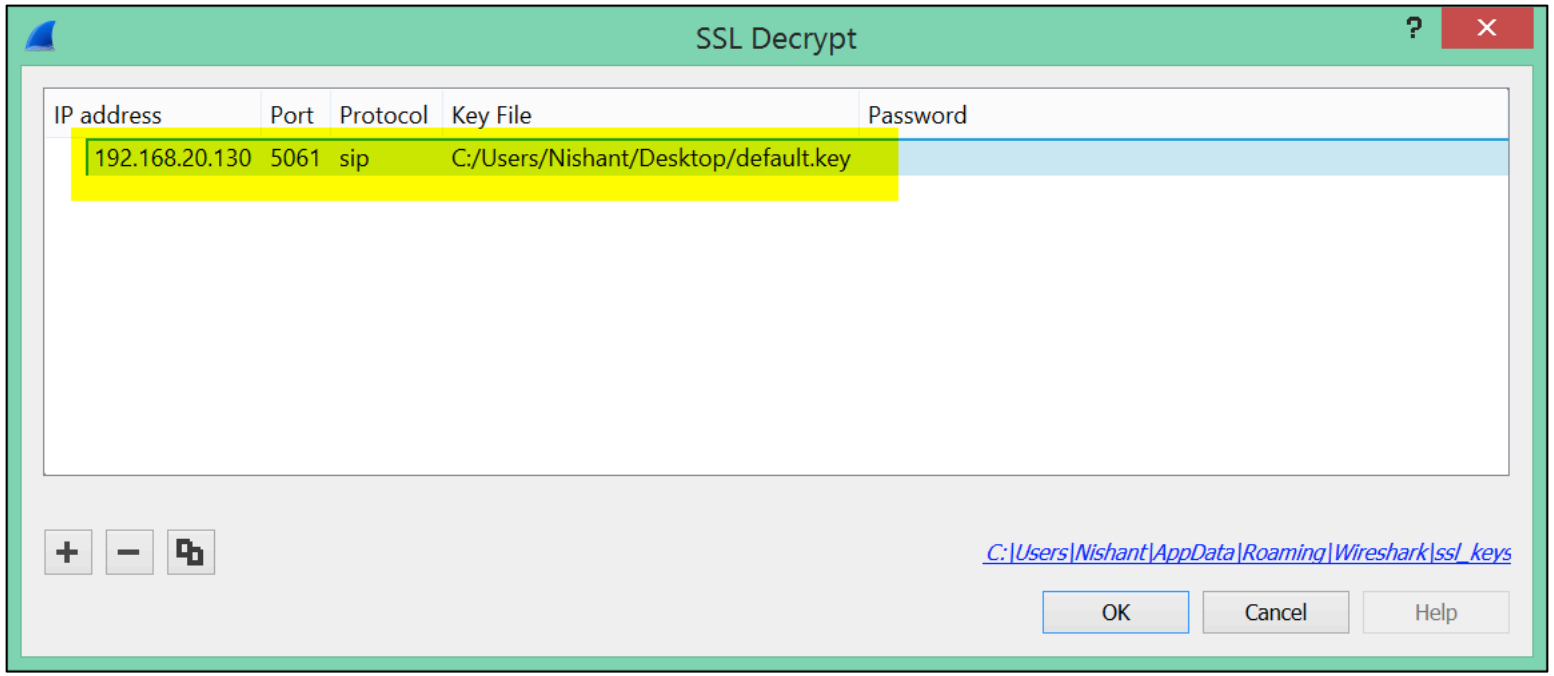
©PentesterAcademy.com

编辑 > 偏好 > 协议 > SSL



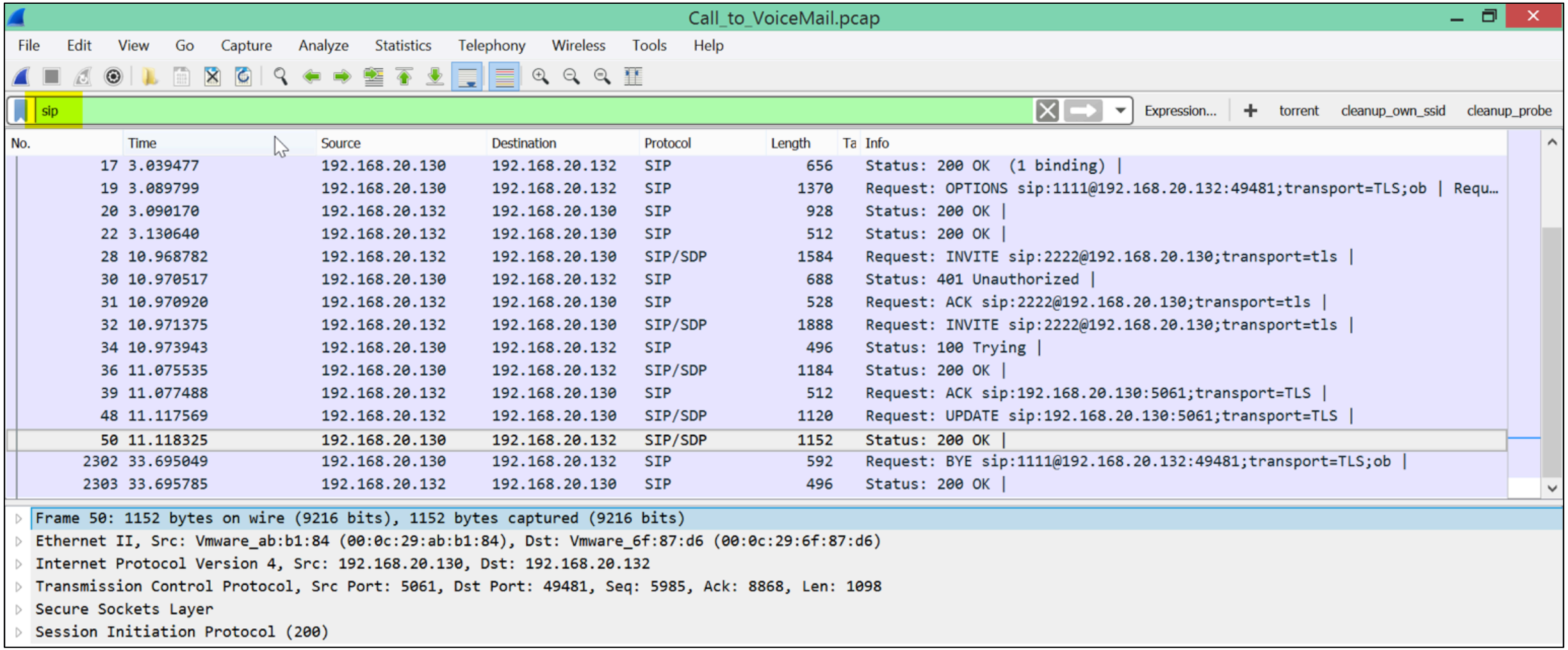
©PentesterAcademy.com

添加Asterisk默认私钥



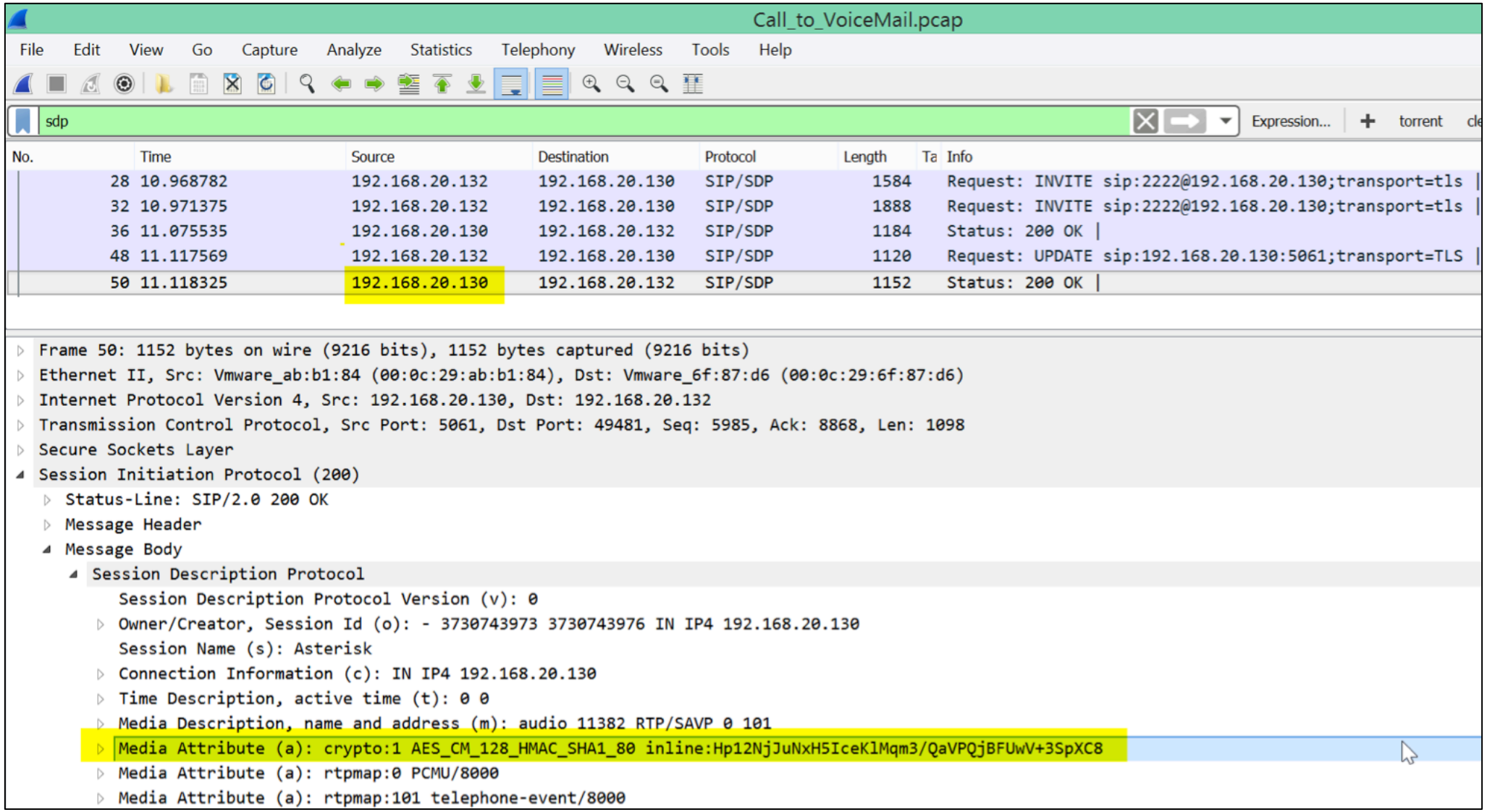
©PentesterAcademy.com

解密SIP流量



©PentesterAcademy.com

SIP/SDP解密包中的SRTP密钥



©PentesterAcademy.com

解密SRTP的开源工具

- SRTP Decrypt
- Libsrtp

©PentesterAcademy.com

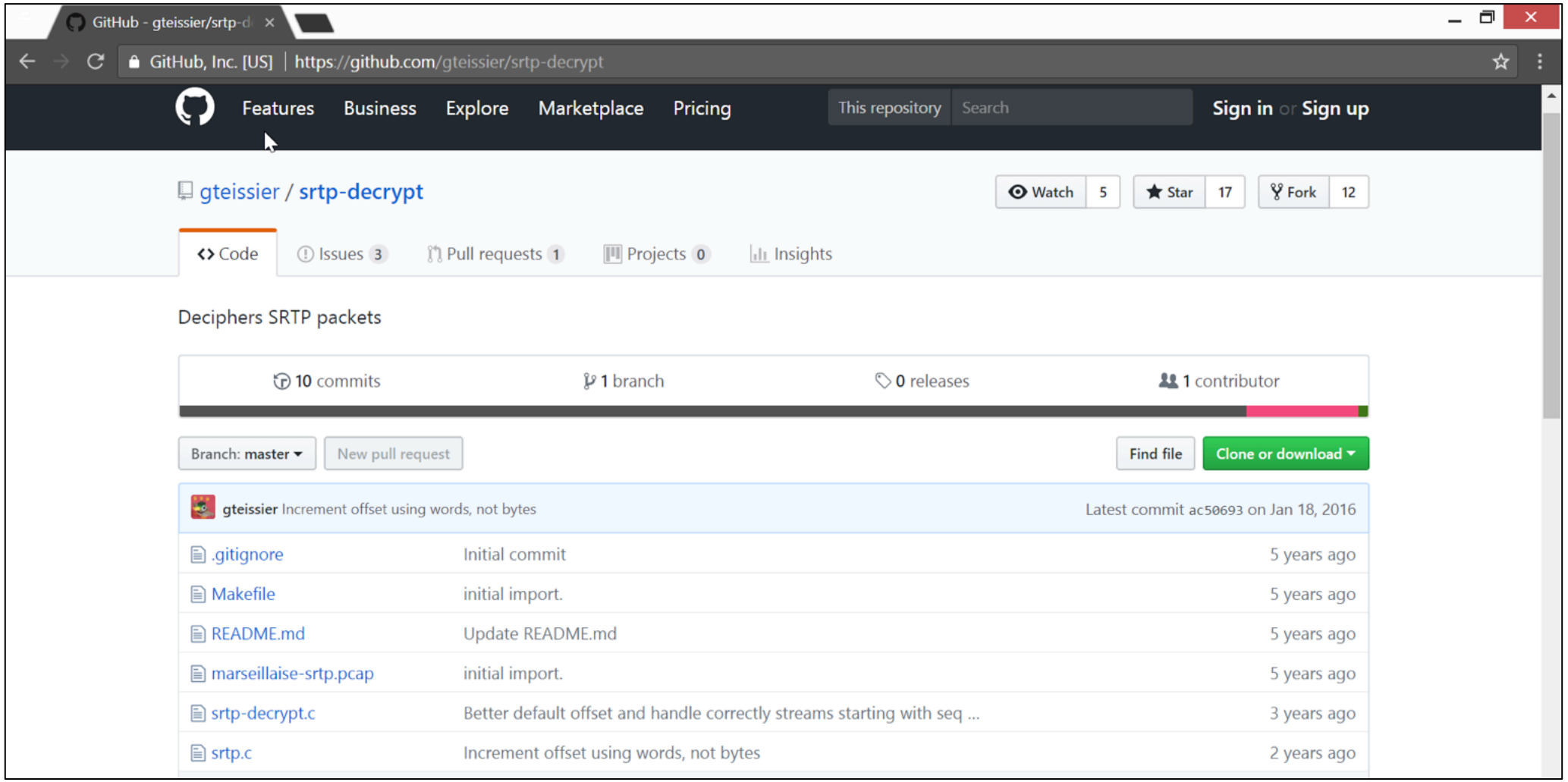
SRTP解密

- 解密SRTP数据包的工具
- 利用对称密钥解密SRTP流量
- 以十六进制方式(hexdump)输出解密包
- Wireshark可从十六进制文件中复原RTP数据包

©PentesterAcademy.com

SRTP解密

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



SRTP解密: 安装前的准备

- 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
```

©PentesterAcademy.com

SRTP解密: 安装

- 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
```

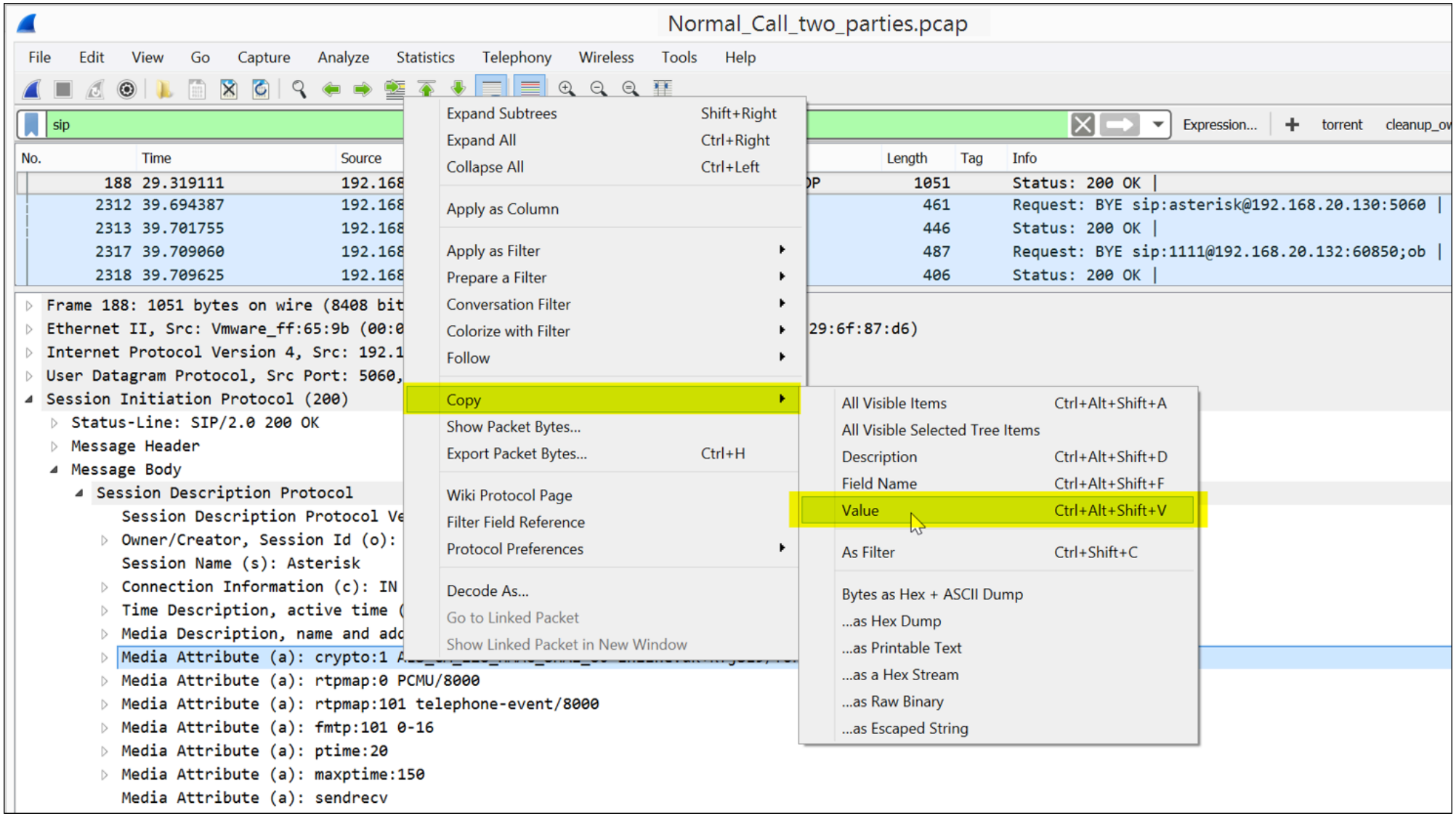
©PentesterAcademy.com

SRTP解密: 准备完成

```
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
```

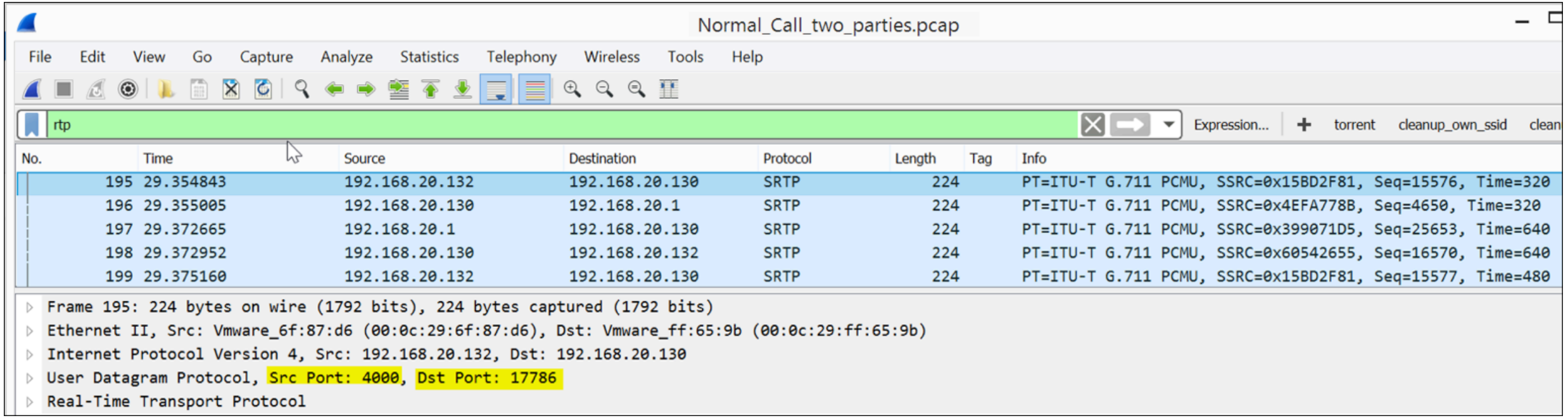
©PentesterAcademy.com

SRTP解密: 复制SRTP密钥



©PentesterAcademy.com

SRTP解密: UDP端口



©PentesterAcademy.com

SRTP解密: 解密SRTP流量

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

- -k : Defined SRTP key (该例为uK+RfjSi9/fUFr8zoJu6zdqPw6MGtONhgX4yqwRj)
- Normal_Call_two_parties.pcap 输入文件
- decoded.raw 输出文件

```
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'
```

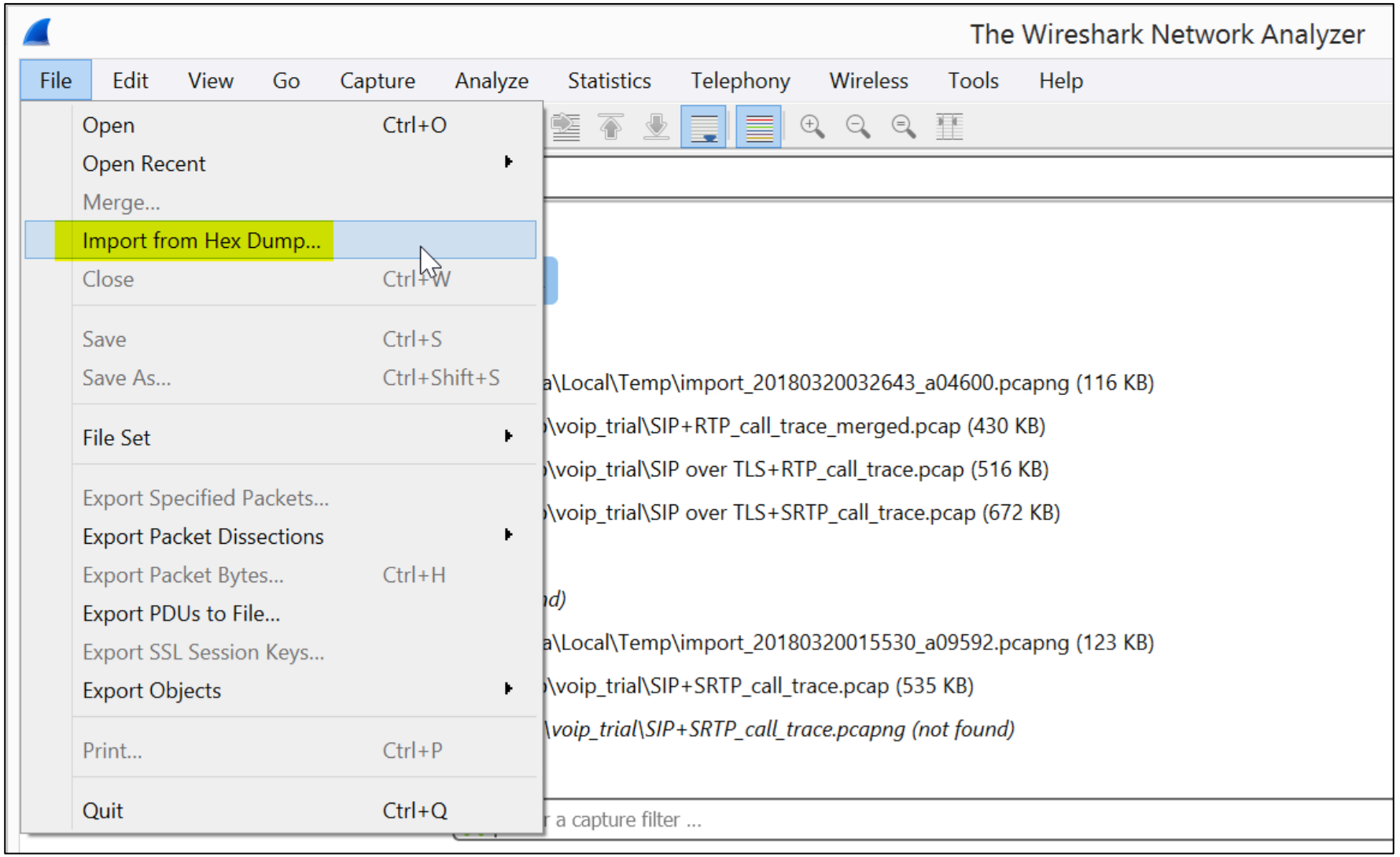
©PentesterAcademy.com

SRTP解密: 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 ff ff 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 7e 7e 7e 7e 7e 7e 7e 7e 7e 7e 7e 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 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
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 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 7e 7e 7e 7e 7e 7e 7e ff 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%--
```

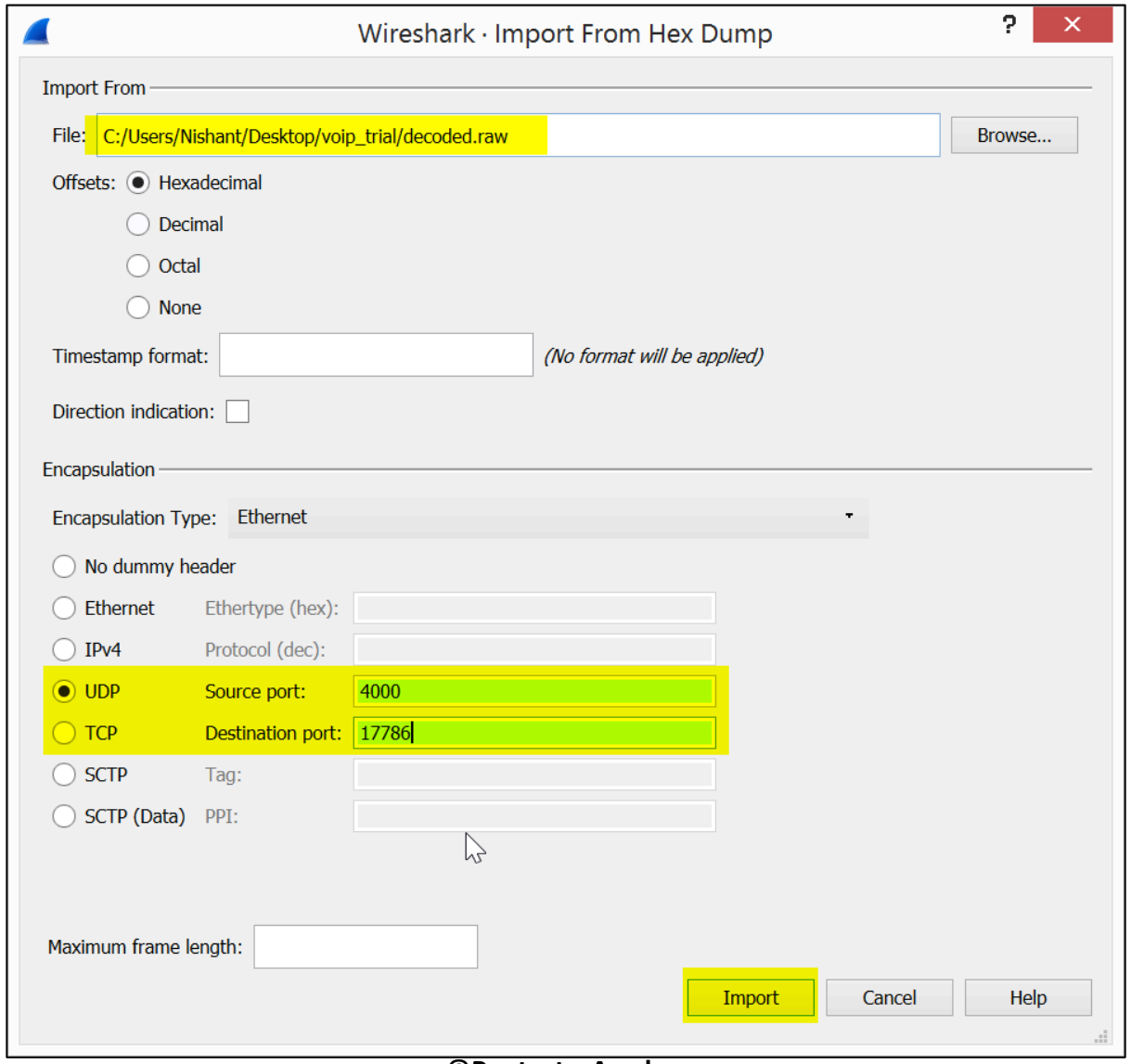
©PentesterAcademy.com

SRTP解密: 导入解密内容



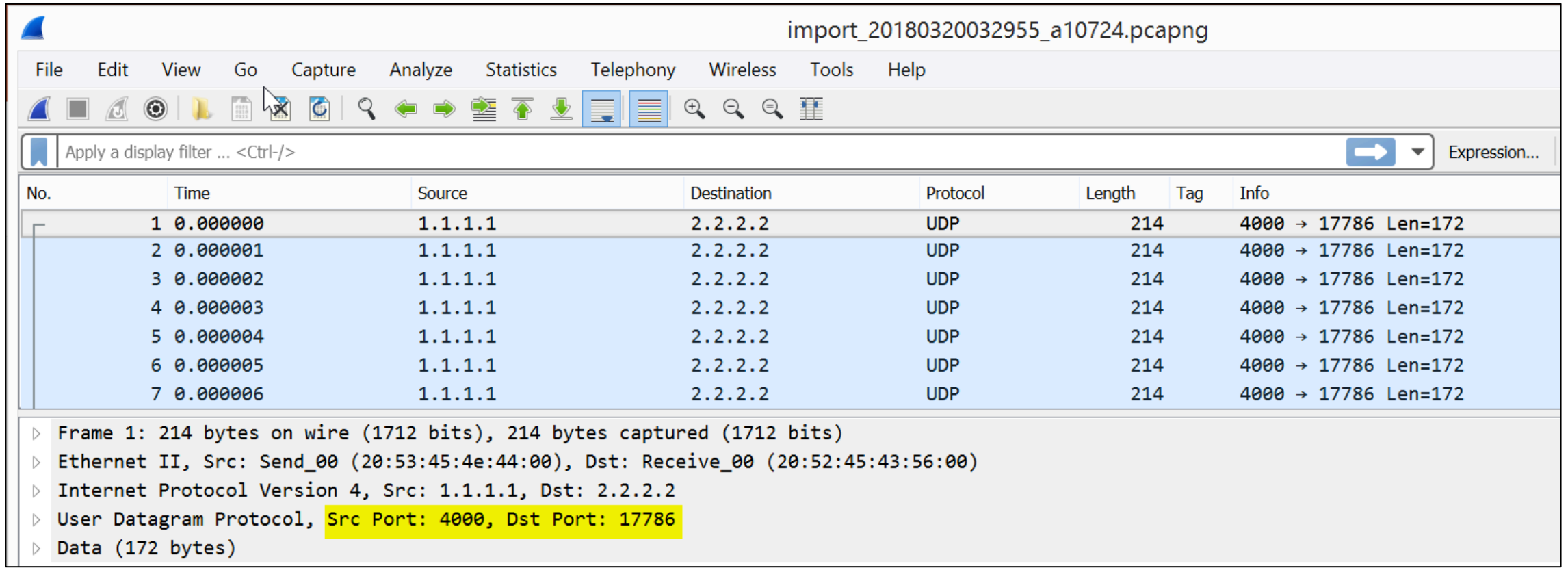
©PentesterAcademy.com

SRTP解密: 导入解密内容



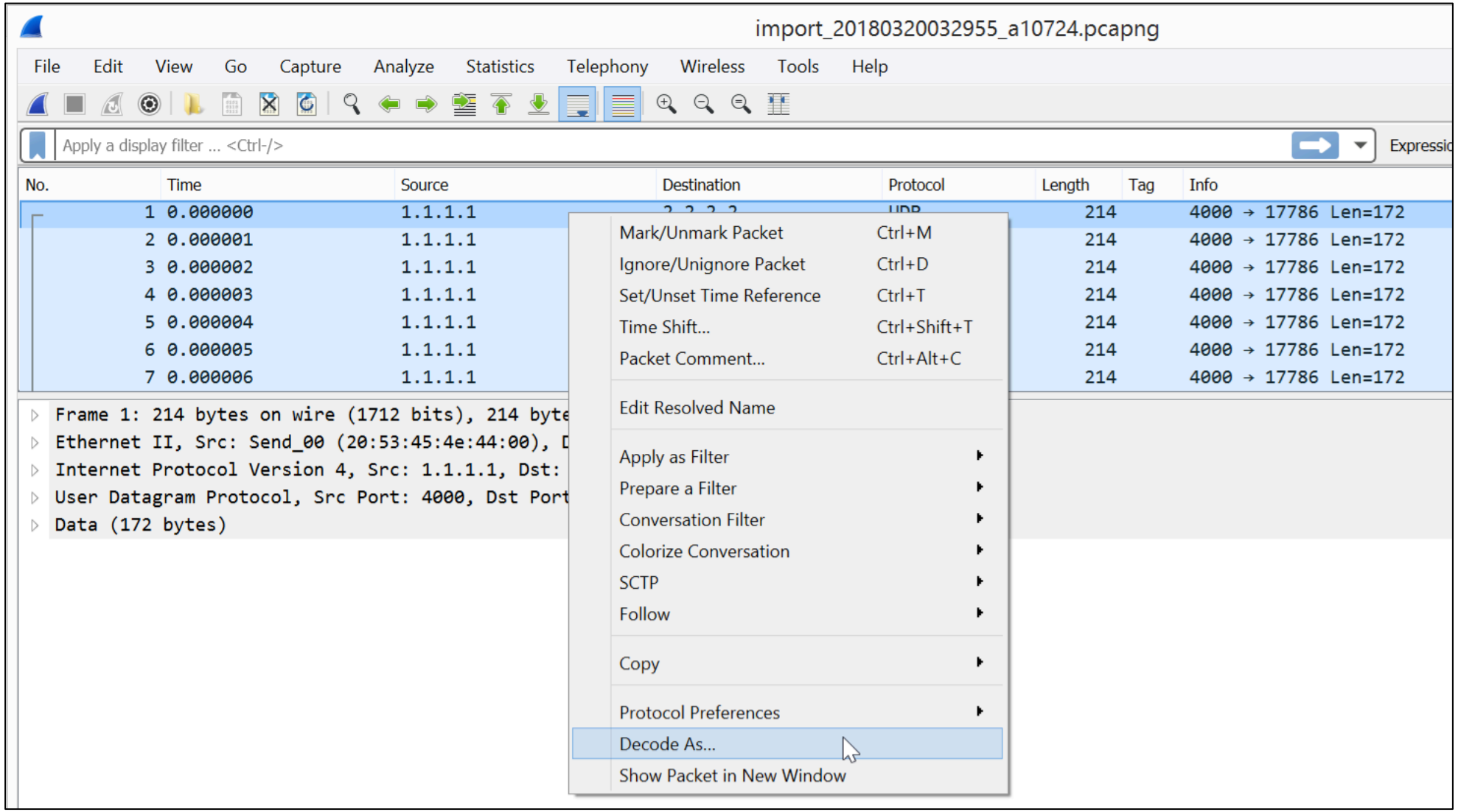
©PentesterAcademy.com

SRTP解密: 导入解密的UDP数据包



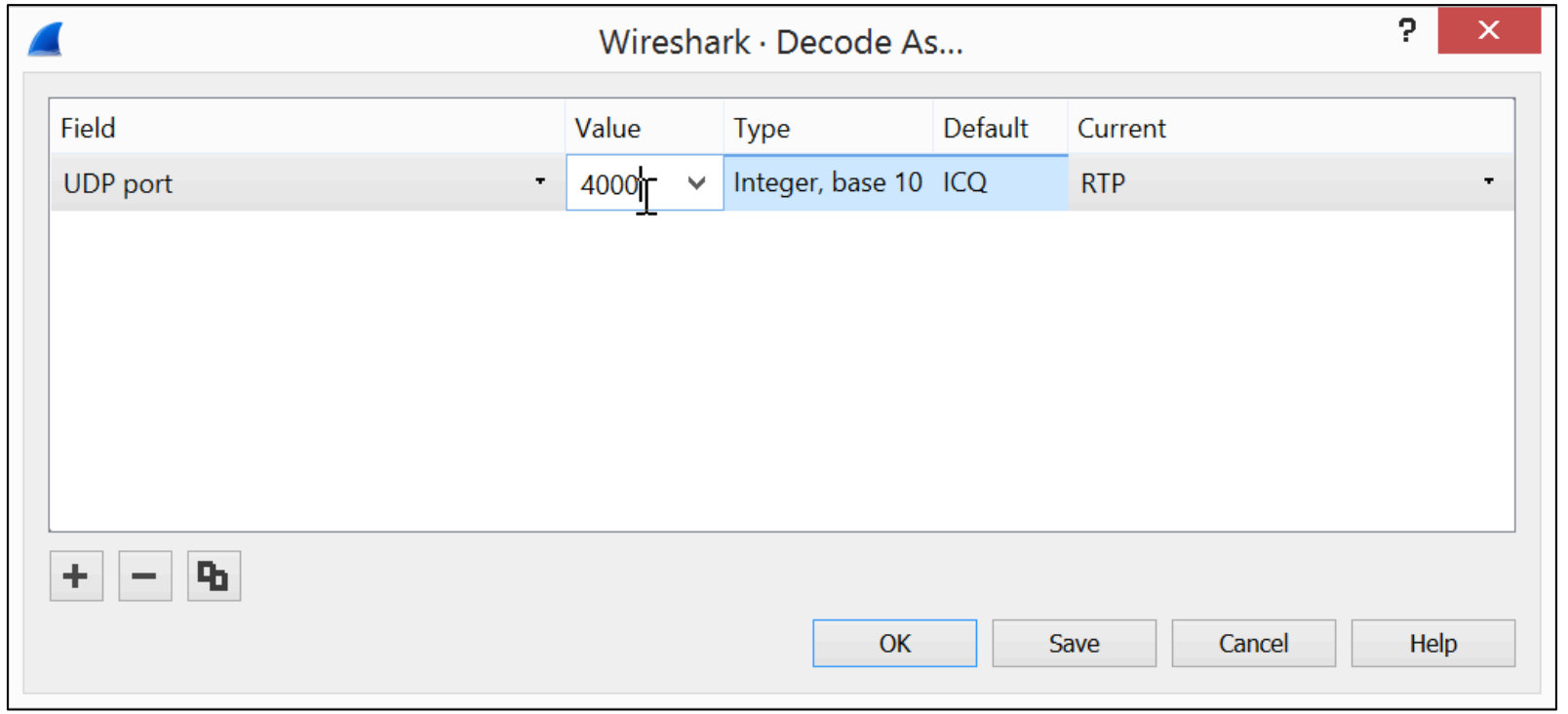
©PentesterAcademy.com

SRTP解密: 解码



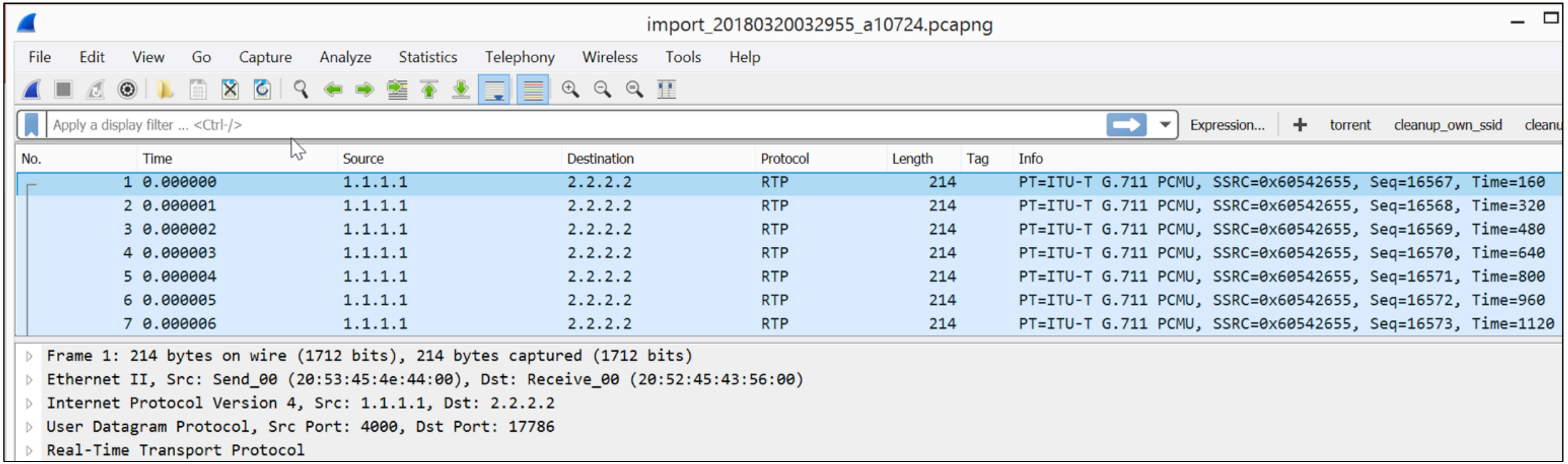
©PentesterAcademy.com

SRTP解密: 解码为RTP



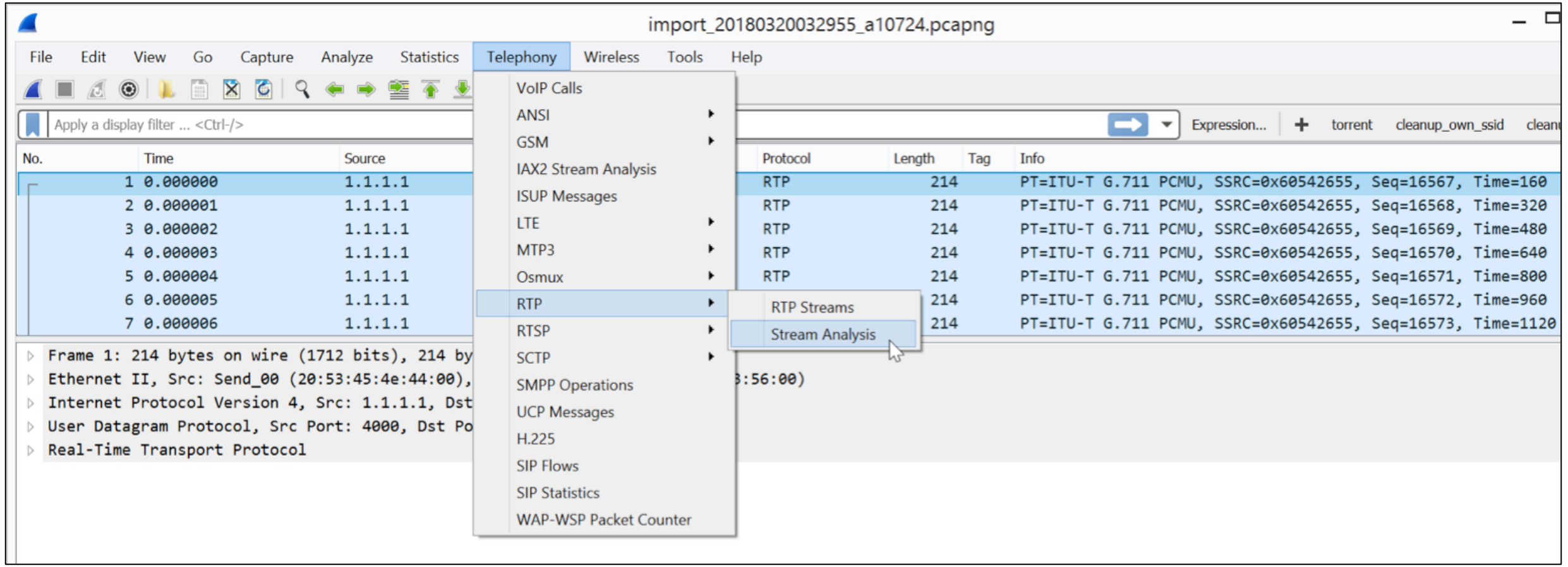
©PentesterAcademy.com

SRTP解密: 解码后的数据包



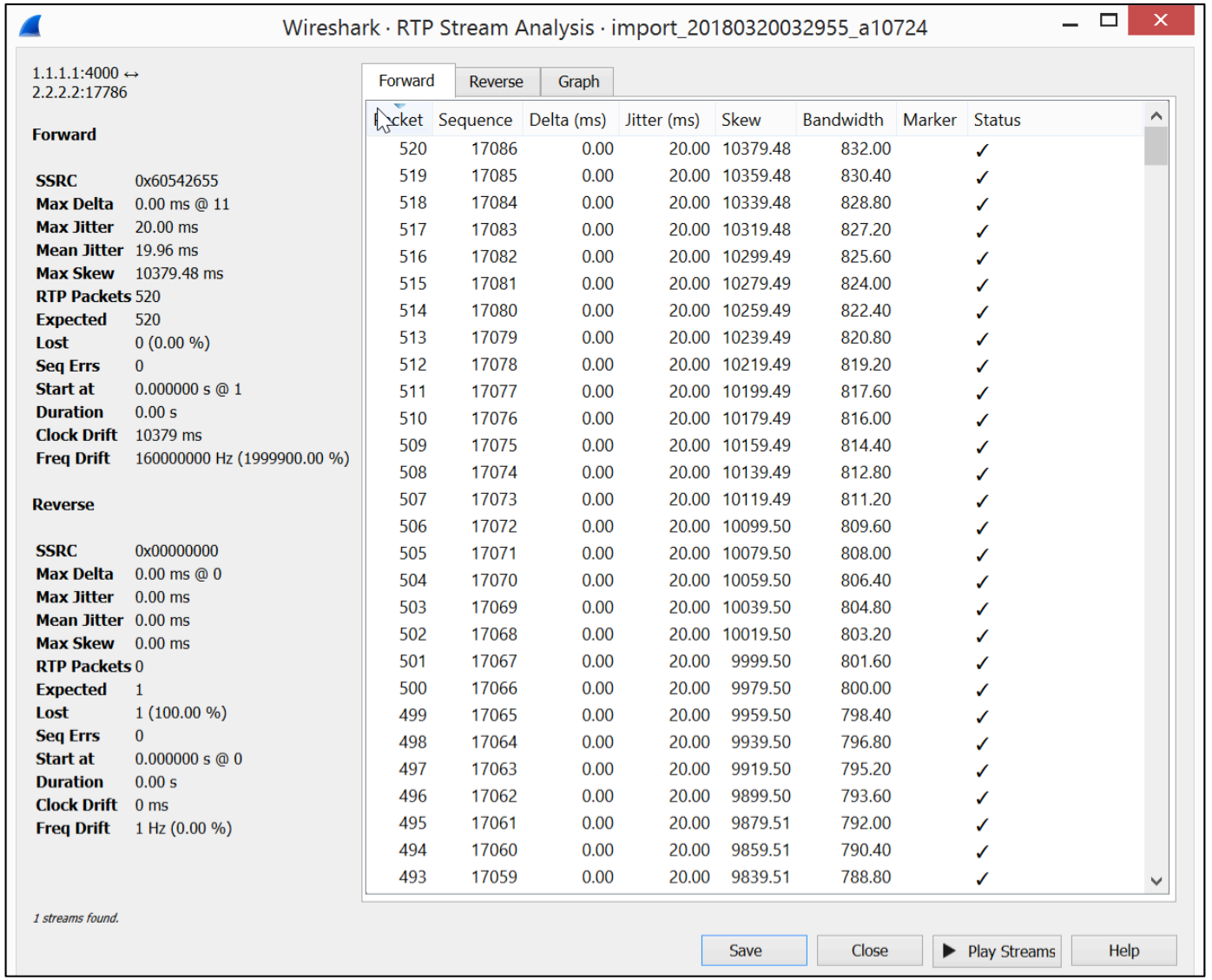
©PentesterAcademy.com

SRTP解密: 检查RTP流



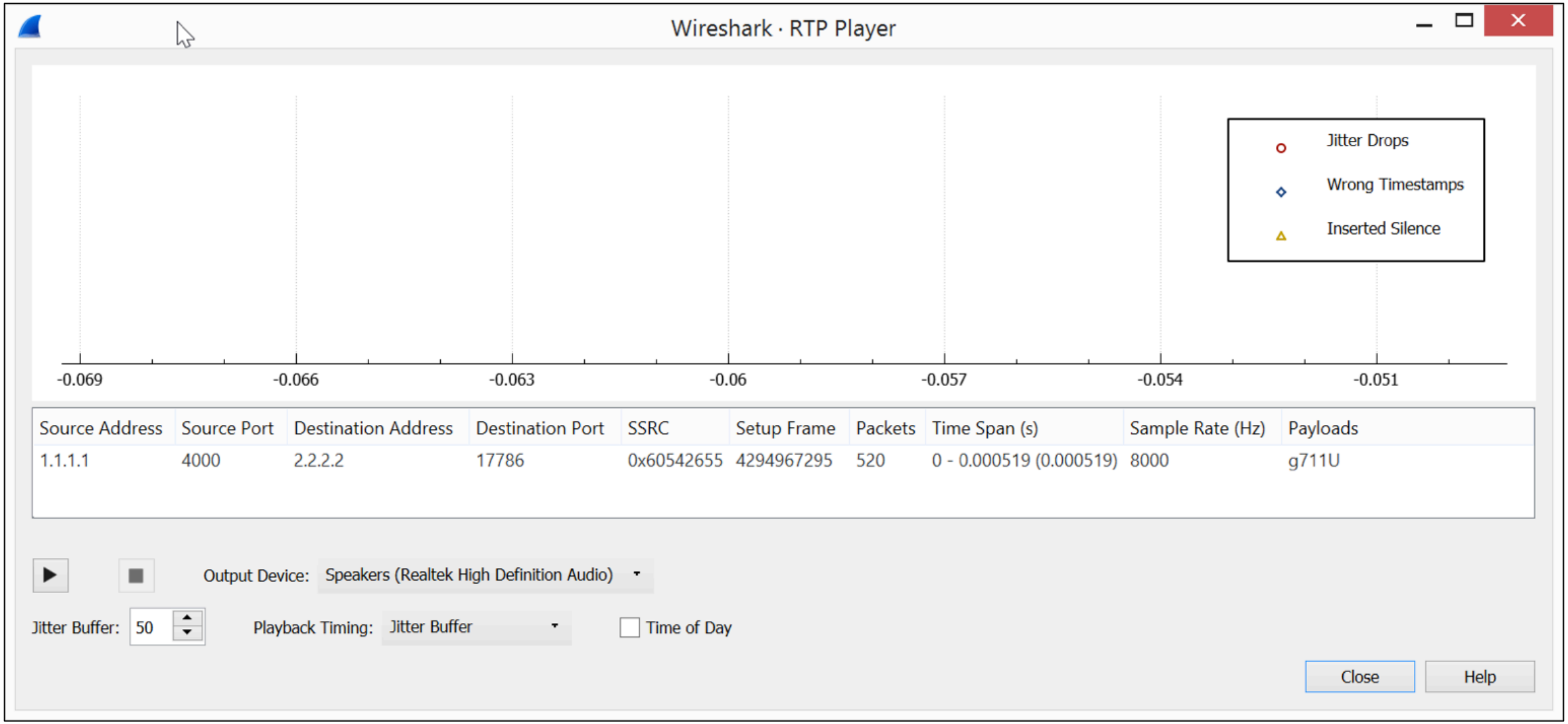
©PentesterAcademy.com

SRTP解密: 分析RTP流



©PentesterAcademy.com

SRTP解密: 播放解密后的通话



©PentesterAcademy.com

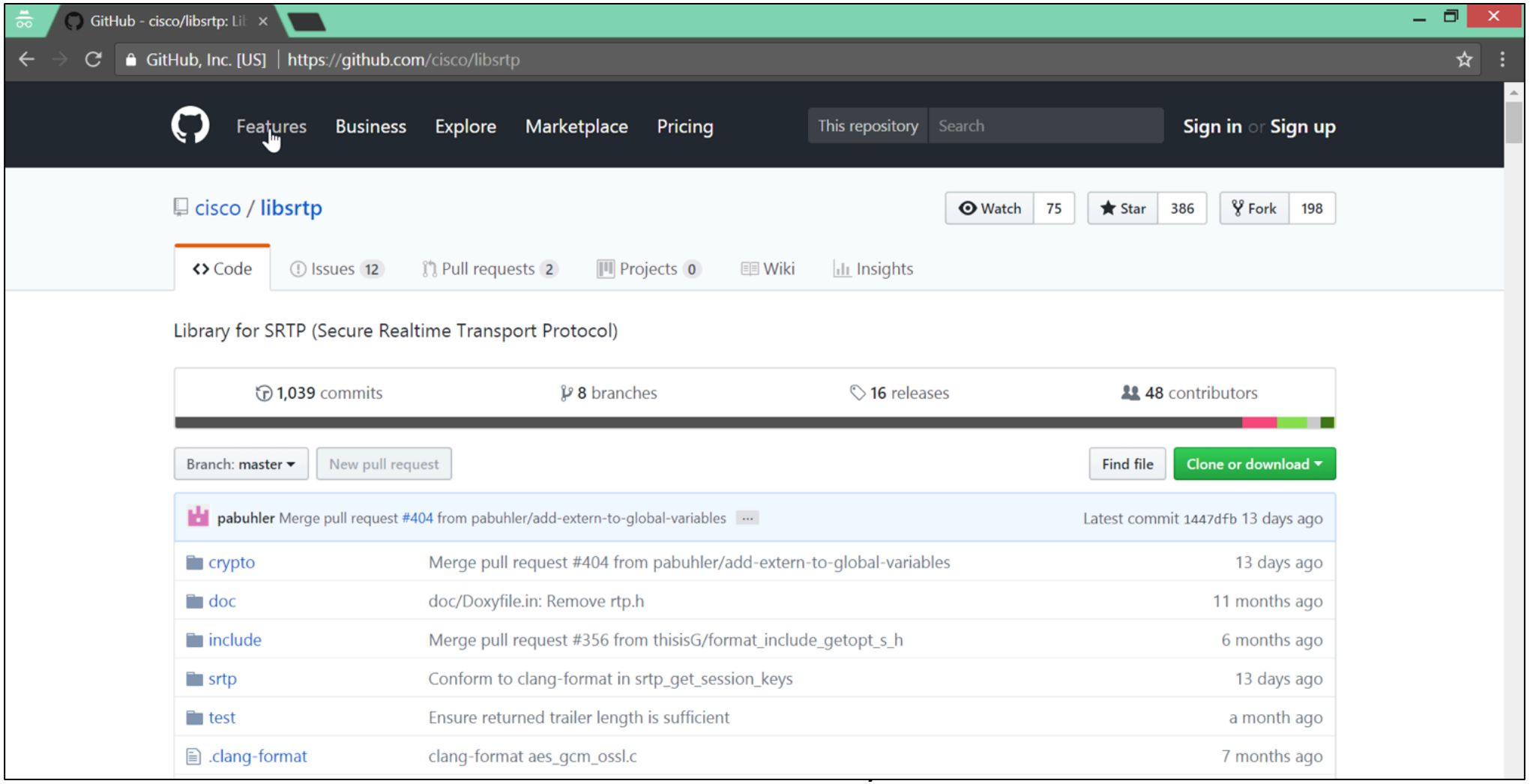
Libsrtp

- 安全实时传输协议(SRTP)的具体实现
- 可解密SRTP数据包

©PentesterAcademy.com

Libsrtp

- GitHub: github.com/cisco/libsrtp



Libsrtp: 安装

- 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/
```

©PentesterAcademy.com

Libsrtp: 安装

- 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
```

©PentesterAcademy.com

Libsrtp: 安装

- Make

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

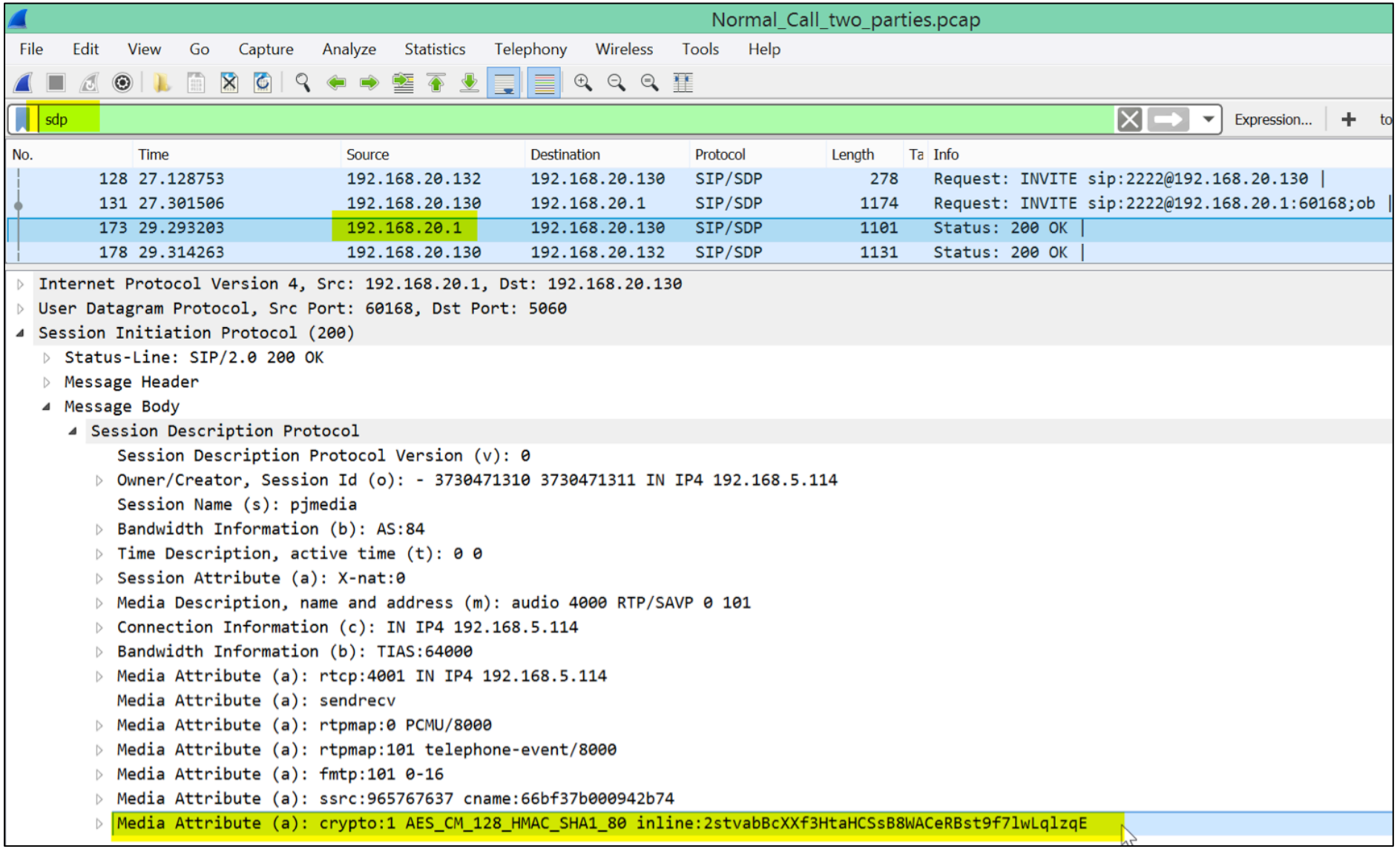
©PentesterAcademy.com

Libsrtp: 准备完成

```
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
```

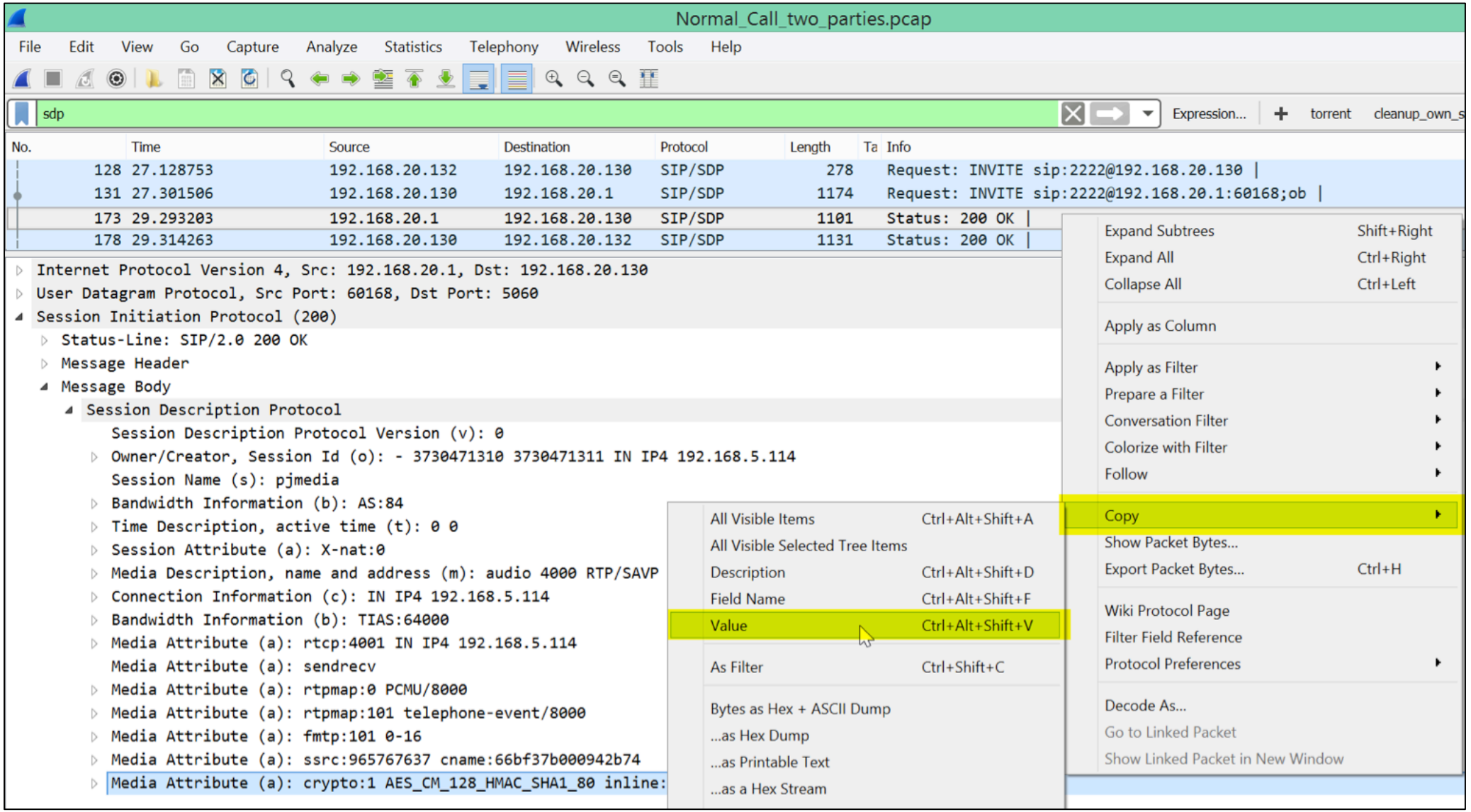
©PentesterAcademy.com

Libsrtp: SRTP密钥



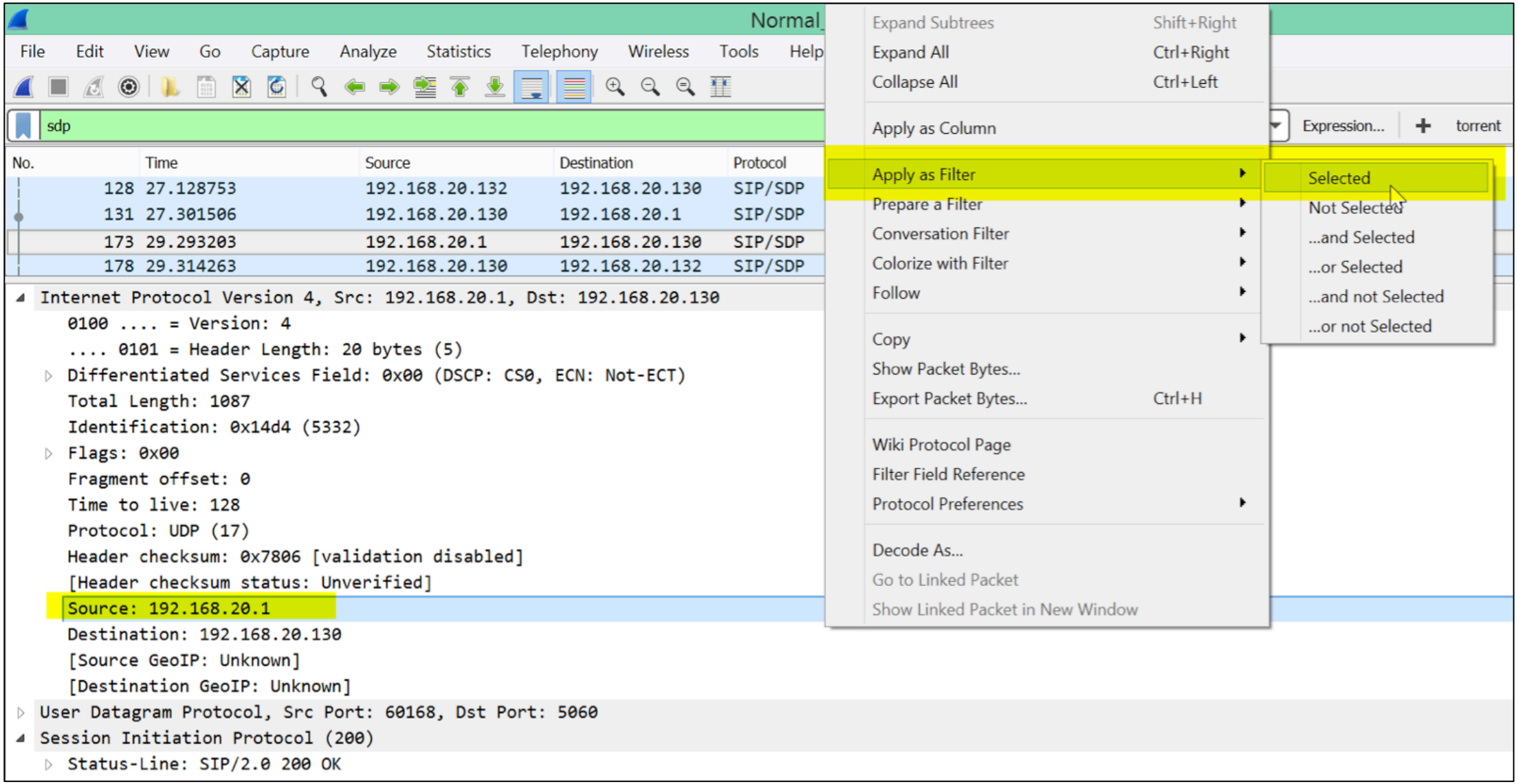
©PentesterAcademy.com

Libsrtp: 复制SRTP密钥



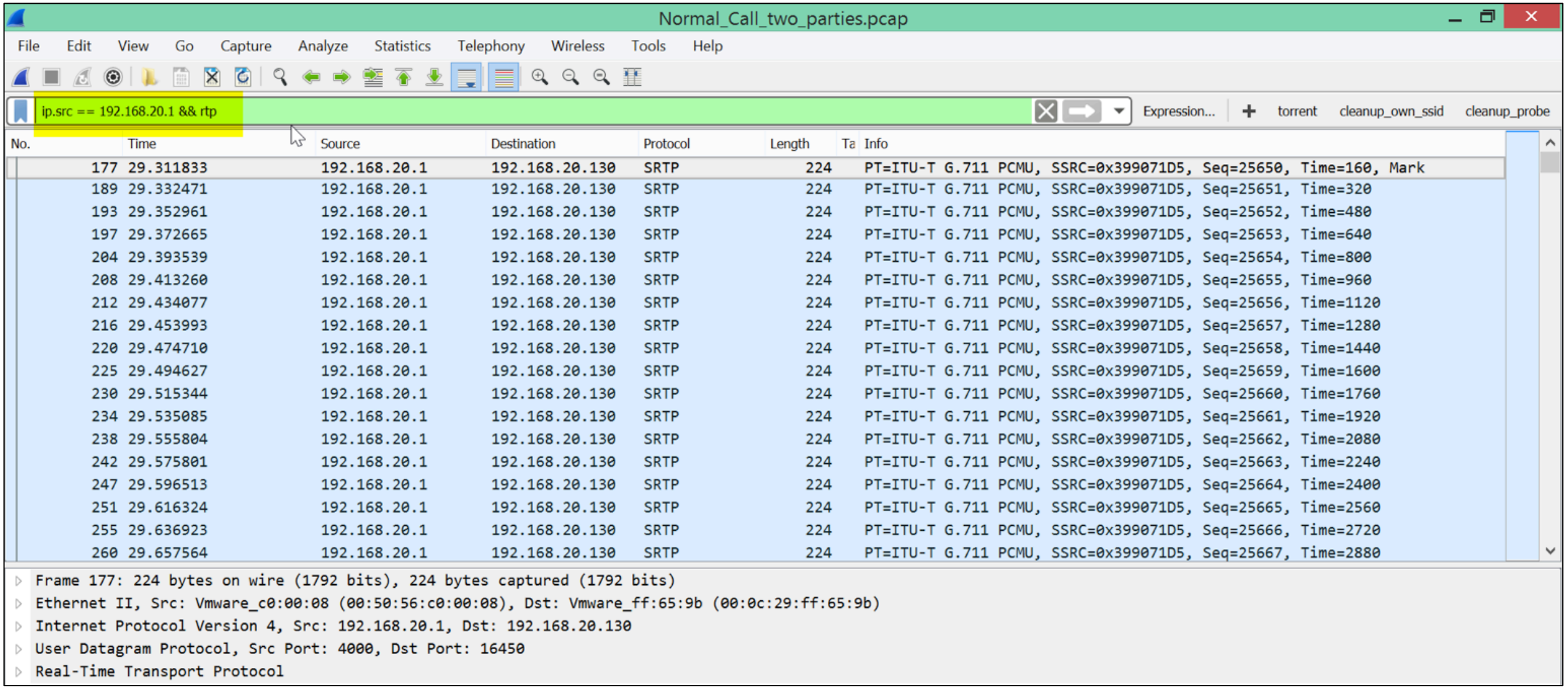
©PentesterAcademy.com

Libsrtp: 过滤单一发送者



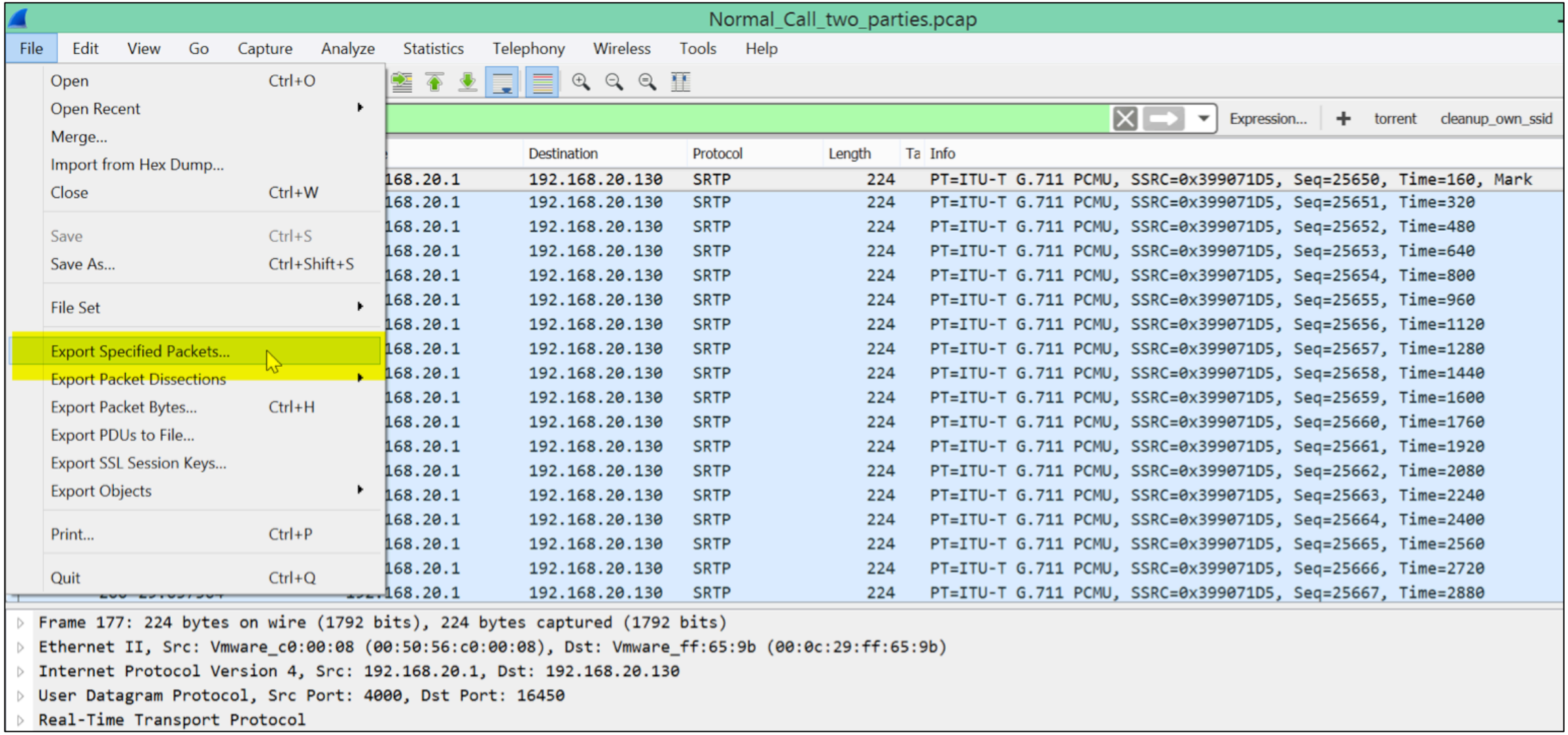
©PentesterAcademy.com

Libsrtp: 过滤单一RTP流



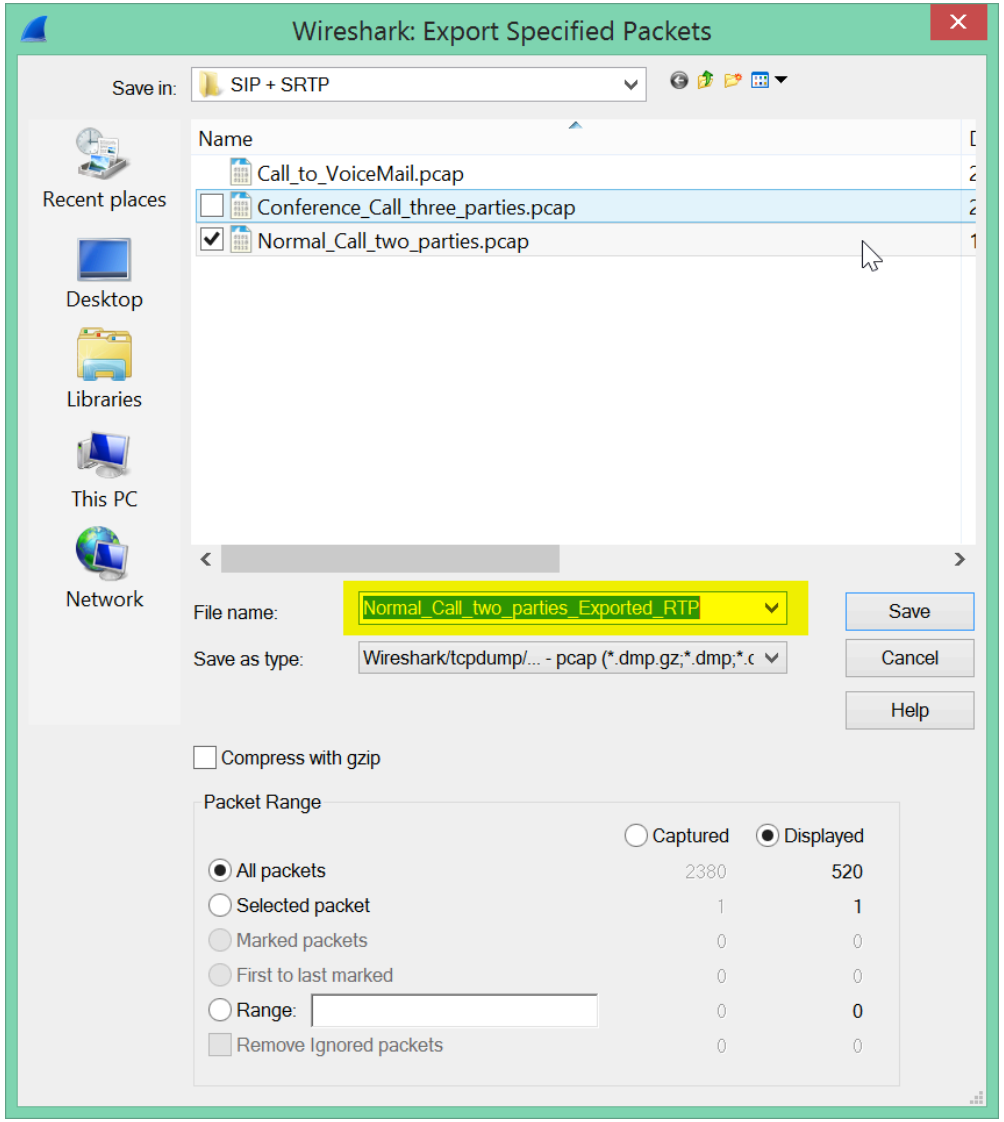
©PentesterAcademy.com

Libsrtp: 导出过滤的流量



©PentesterAcademy.com

Libsrtp: 保存导出流量



©PentesterAcademy.com

Libsrtp: 指令

- ./rtp_decoder -a -t 10 -e 128 -b 2stvabBcXXf3HtaHCSsB8WACeRBst9f7lwLqlzqE * < ./Normal_Call_two_parties_Exported_RTP.pcap

- -a 使用消息验证
- -t 认证标签大小(80位，即10字节)
- -e 加密密钥的长度，该例使用AES_CM_128_HMAC_SHA1_80
  密钥长度是128位
- -b ASCII格式的SRTP密钥

©PentesterAcademy.com

Libsrtp: 指令输出

```
root@PentesterAcademy:/work/libsrtp/test# ./rtp_decoder -a -t 10 -e 128 -b 2stvabBcXXf3HtaHCSsB8WACeRBst9f7lwLqlzqE * < ../../Normal_Call_two_part
ies_Exported_RTP.pcap
Using libsrtp2 2.2.0-pre [0x2020000]
security services: confidentiality message authentication
setting tag len 10
set master key/salt to dacb6f69b05c5d77f71ed6870 92b01f1/600279106cb7d7fb9702ea973a84
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 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 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
```

©PentesterAcademy.com

Libsrtp: text2pcap的帮助信息

```
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)ecimal;
                          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 strptime.
                          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 将数据包前的文本视为日期和时间
- %M:%S 时间格式
- -u 使用既定的源、目的端口预先设置UDP数据包头部

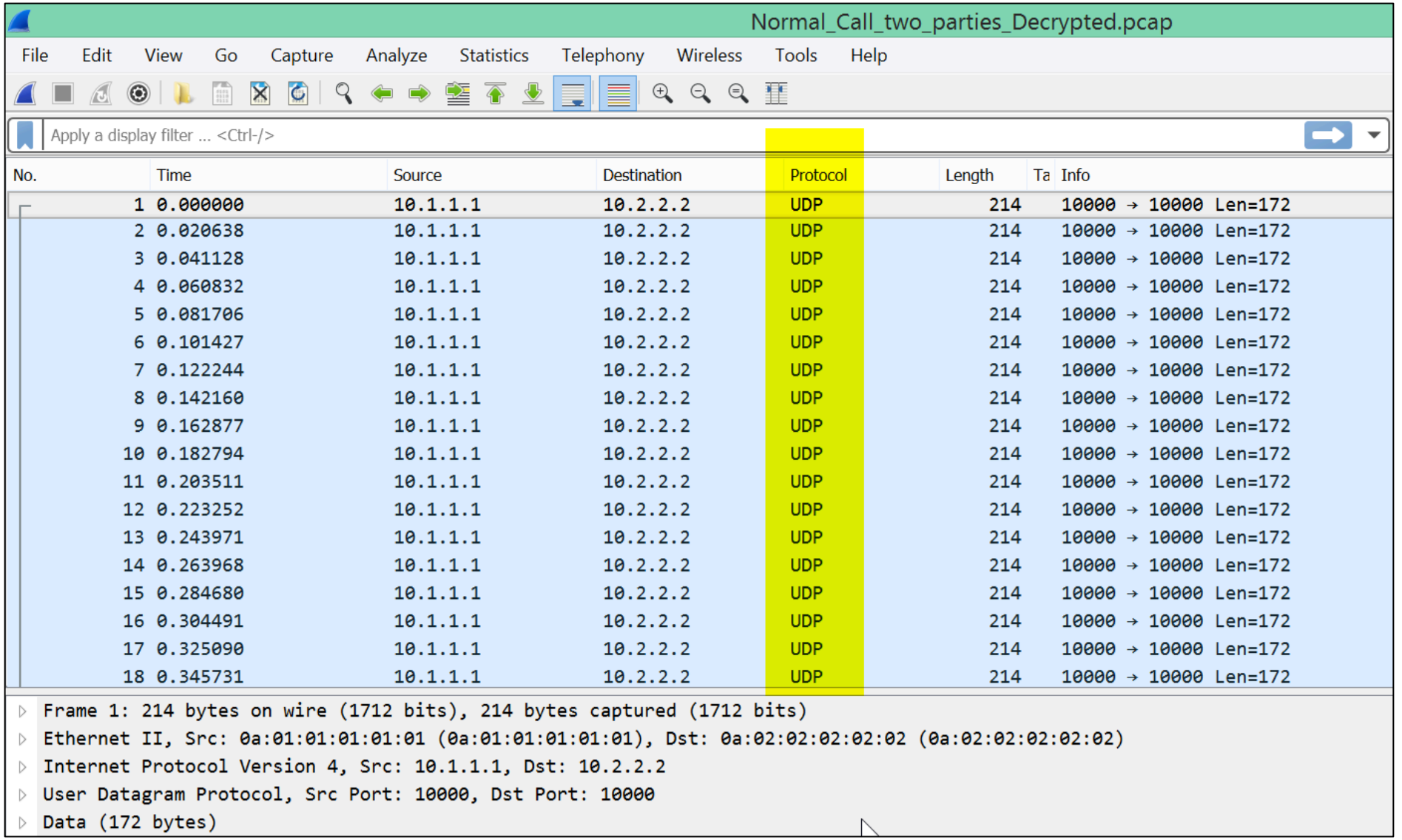
©PentesterAcademy.com

Libsrtp: 解密RTP流量

```
root@PentesterAcademy:/work/libsrtp/test# ./rtp_decoder -a -t 10 -e 128 -b 2stvabBcXXf3Ht
aHCSsB8WACeRBst9f7lwLqlzqE  *  < ./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.
Wrote packet of 214 bytes.
Wrote packet of 214 bytes.
Wrote packet of 214 bytes.
Wrote packet of 214 bytes.
Wrote packet of 214 bytes.
Wrote packet of 214 bytes.
Wrote packet of 214 bytes.
Wrote packet of 214 bytes.
```

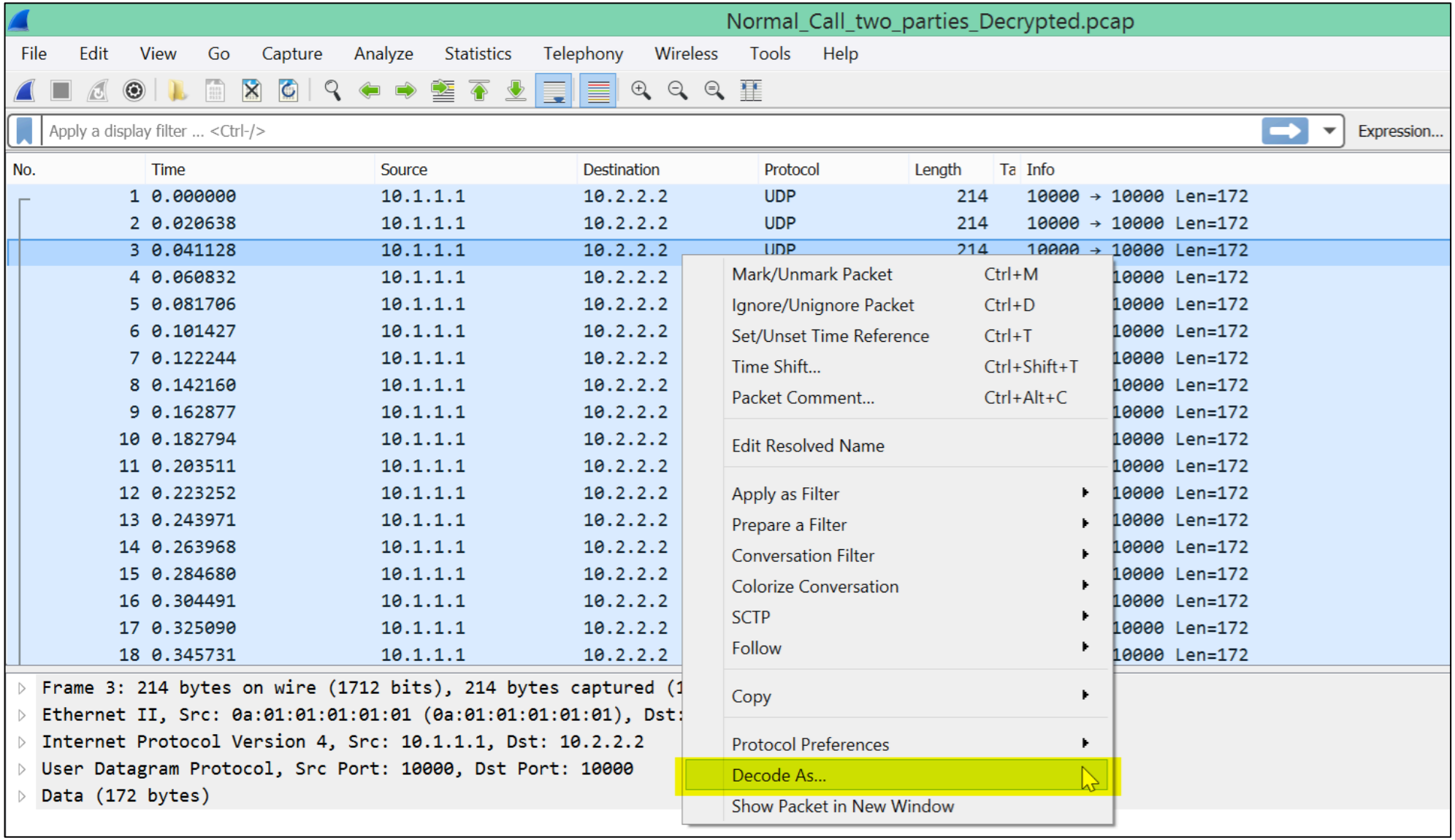
©PentesterAcademy.com

Libsrtp: 解密后的流量



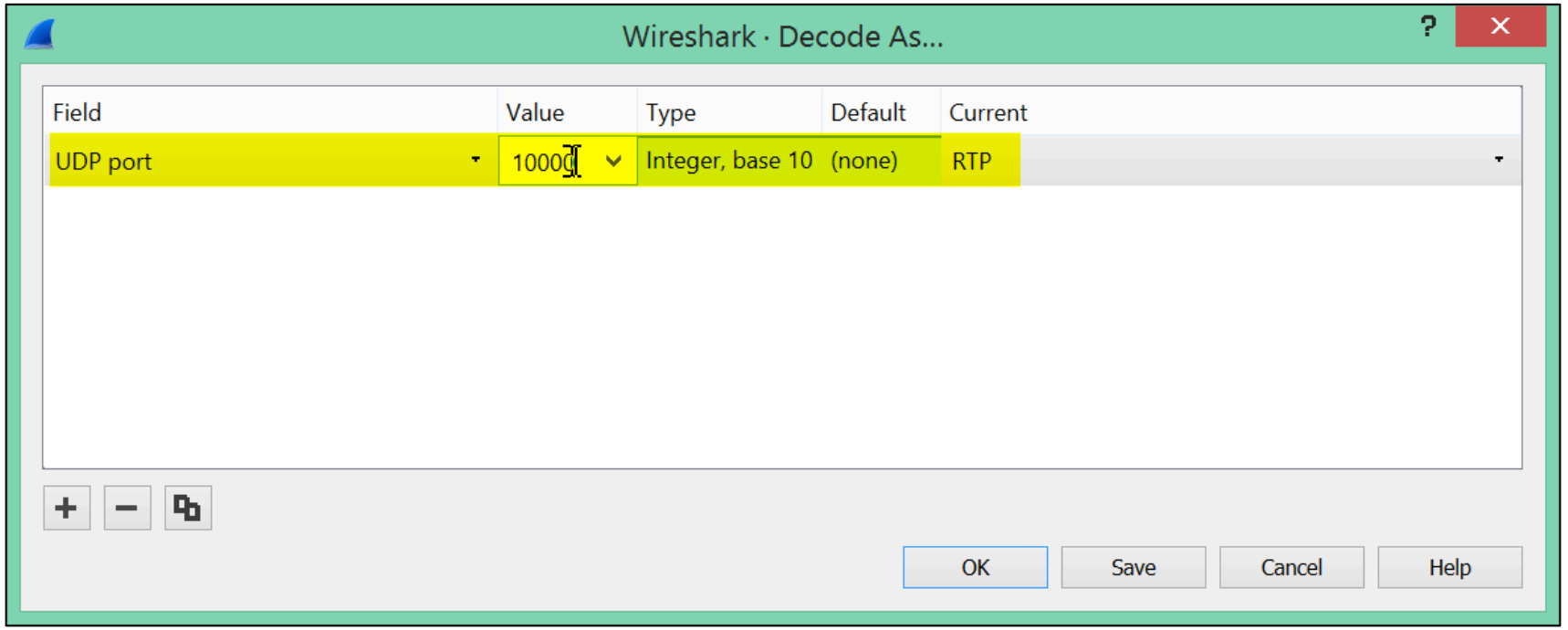
©PentesterAcademy.com

Libsrtp: 解码



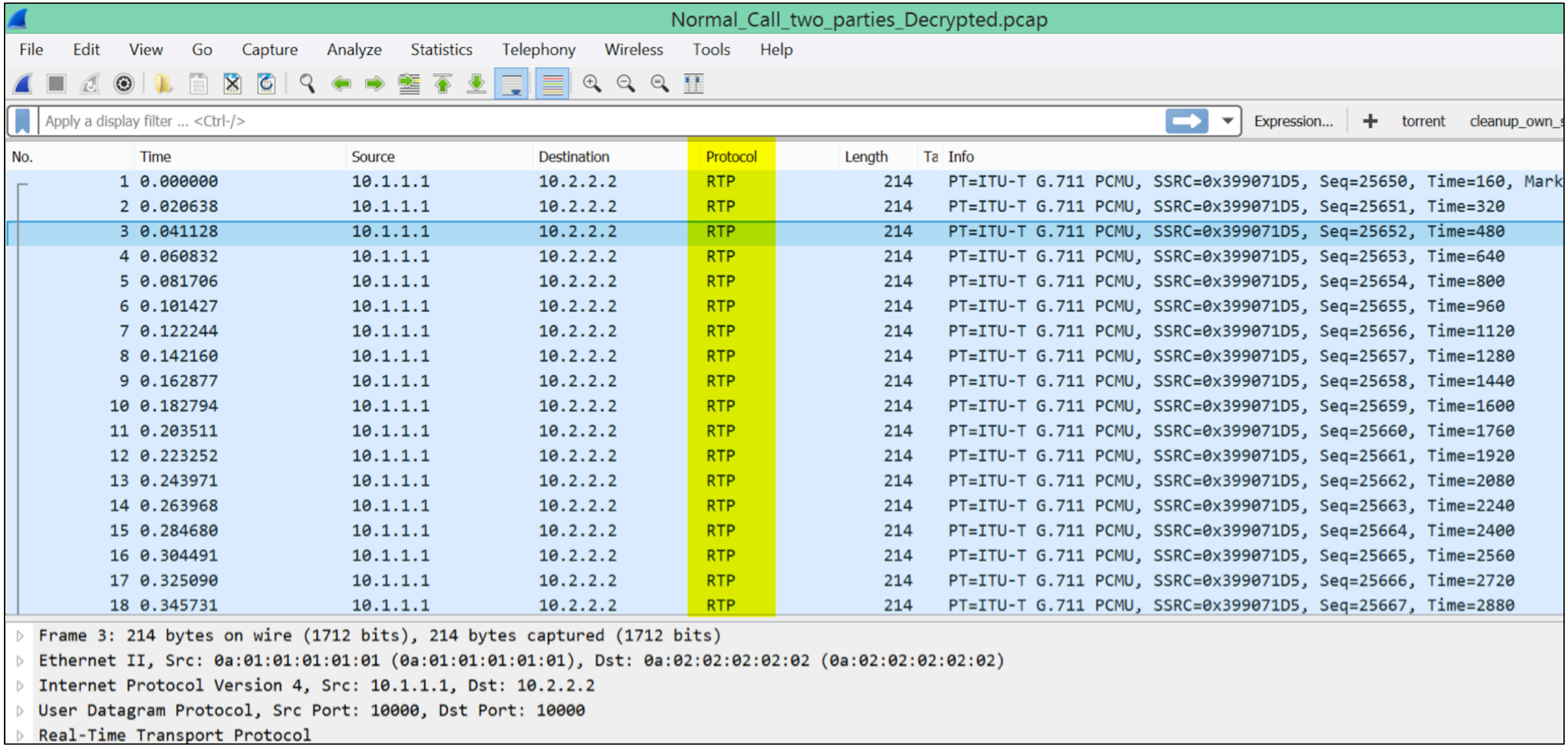
©PentesterAcademy.com

Libsrtp: 解码为RTP



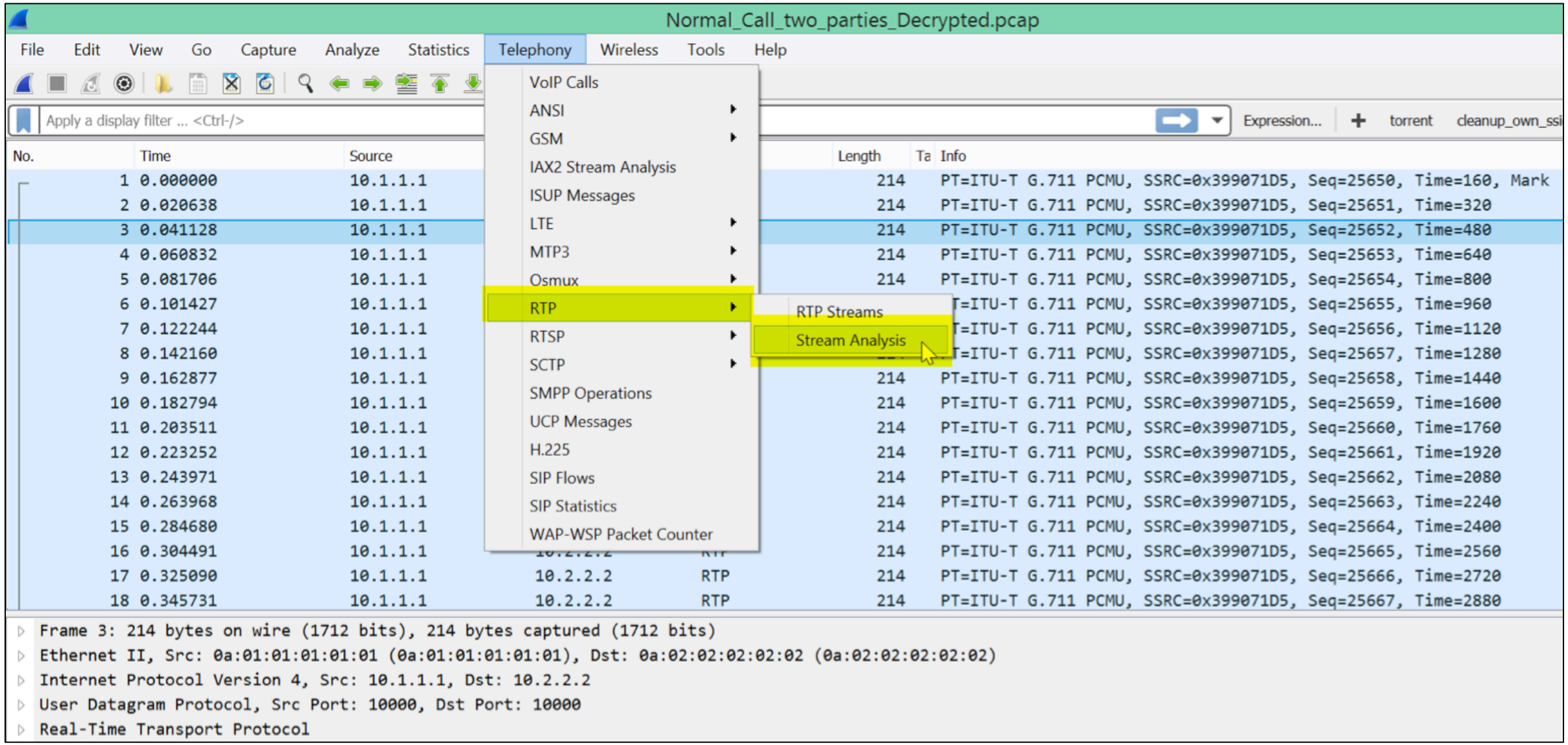
©PentesterAcademy.com

Libsrtp: 解码后的RTP流量



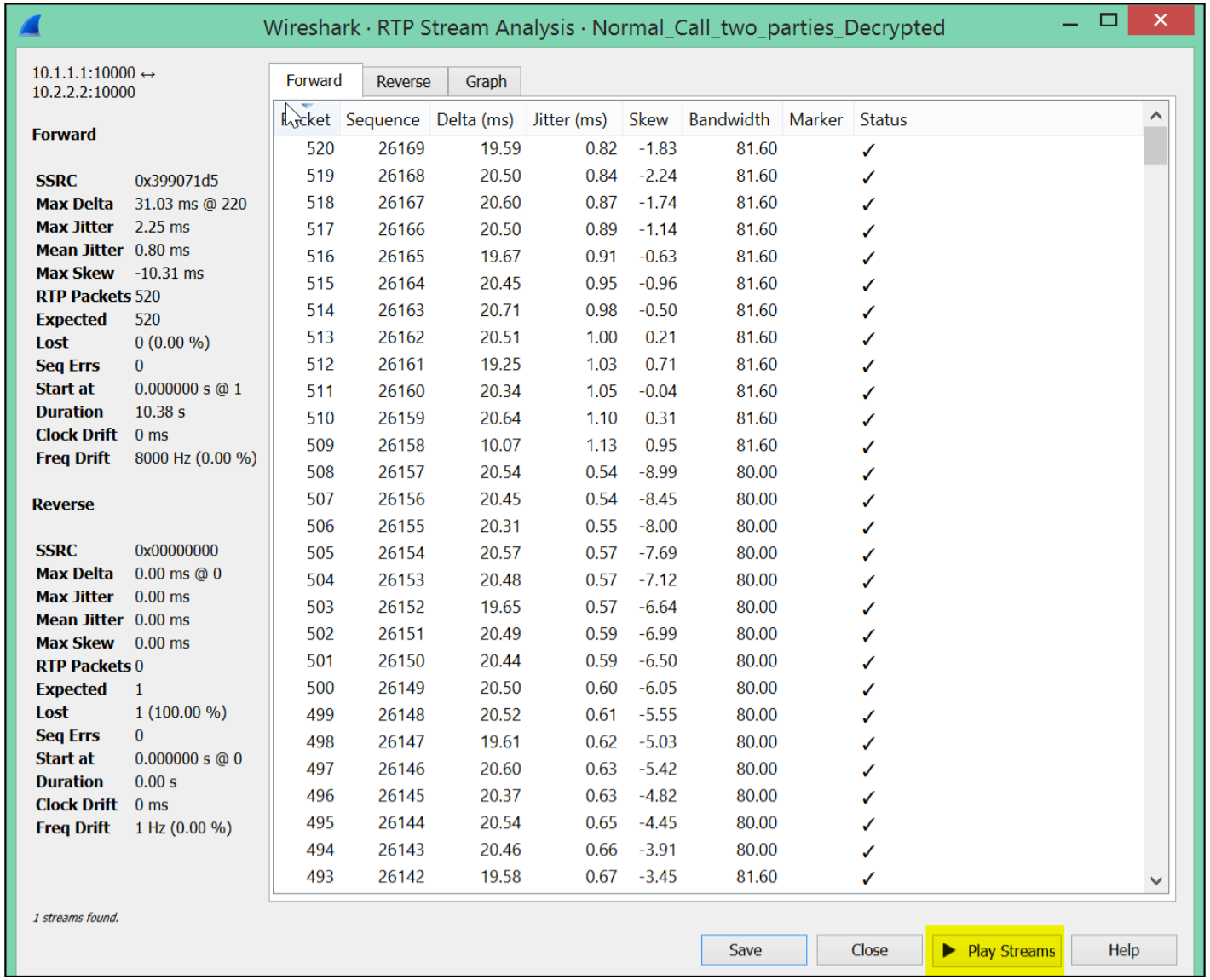
©PentesterAcademy.com

Libsrtp: 分析RTP流



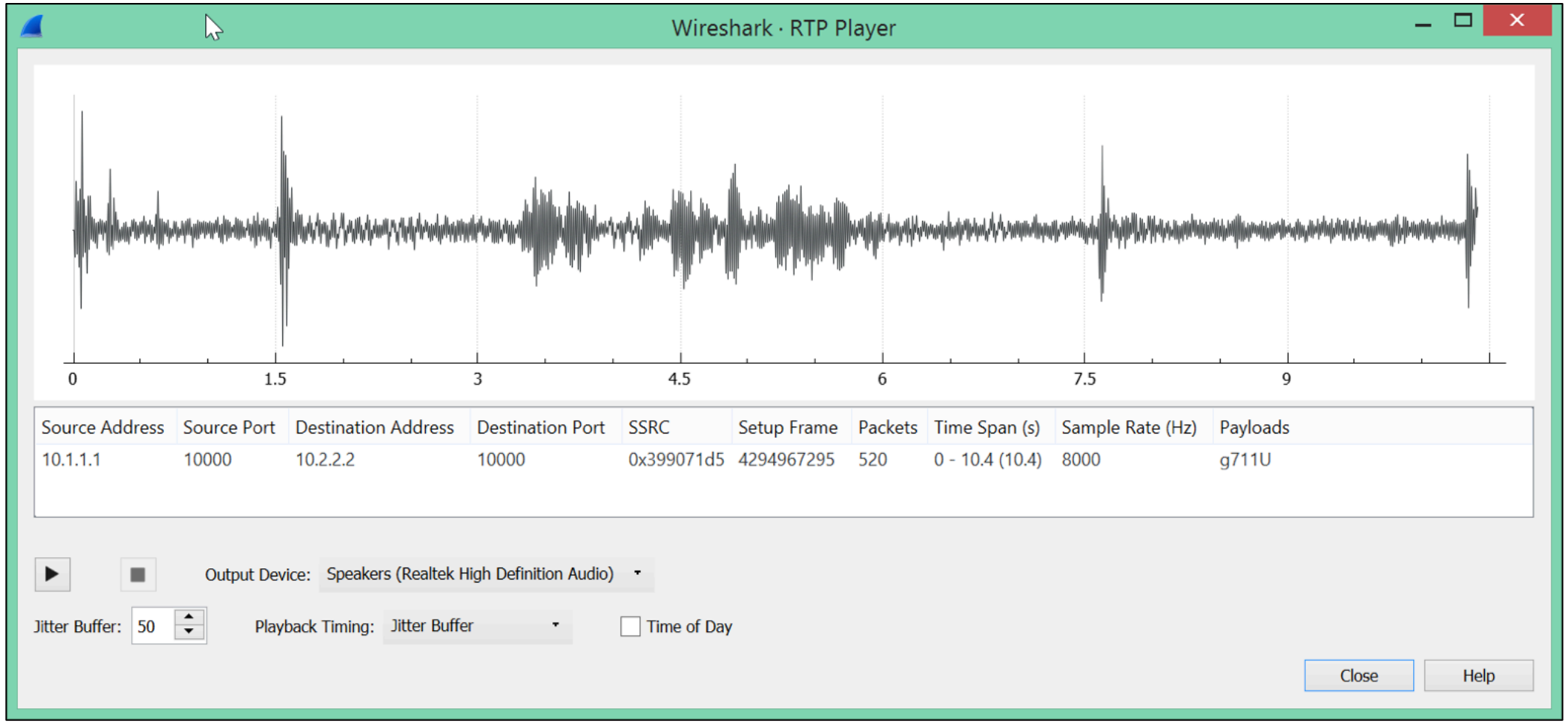
©PentesterAcademy.com

Libsrtp: 分析RTP流



©PentesterAcademy.com

Libsrtp: 播放解密后的通话



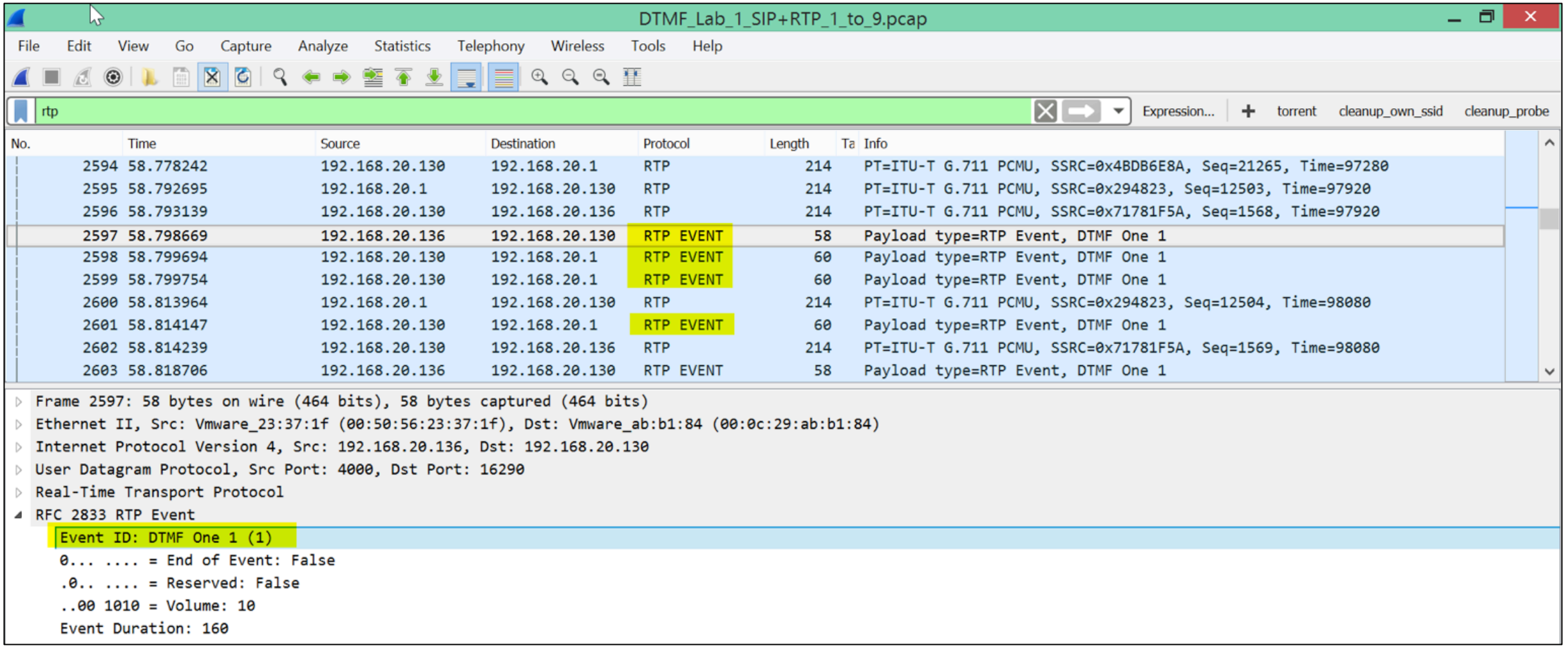
©PentesterAcademy.com

其它重要的部分?

- DTMF
- 短消息(SMS)
- 导出通话

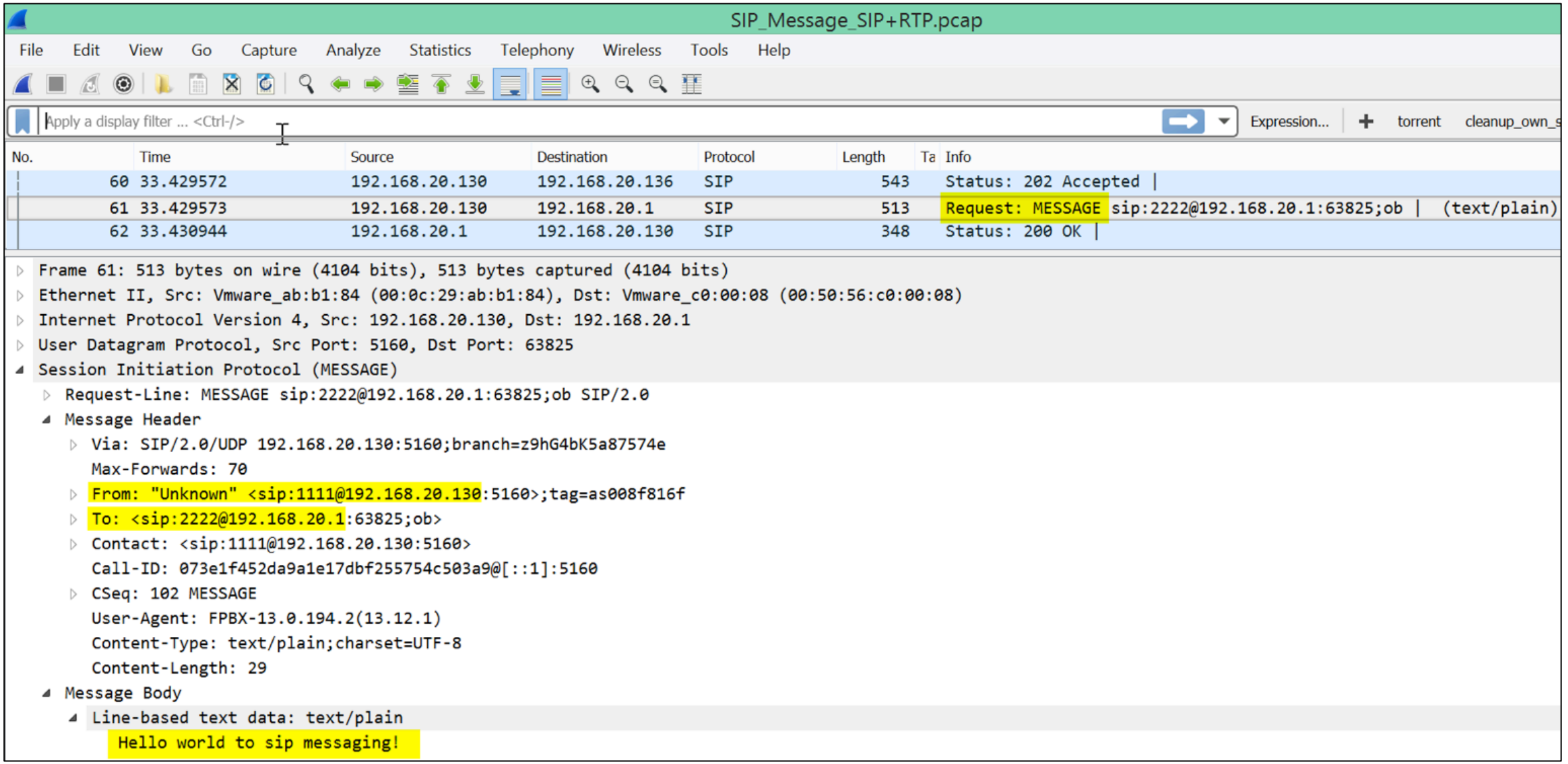
©PentesterAcademy.com

RTP DTMF



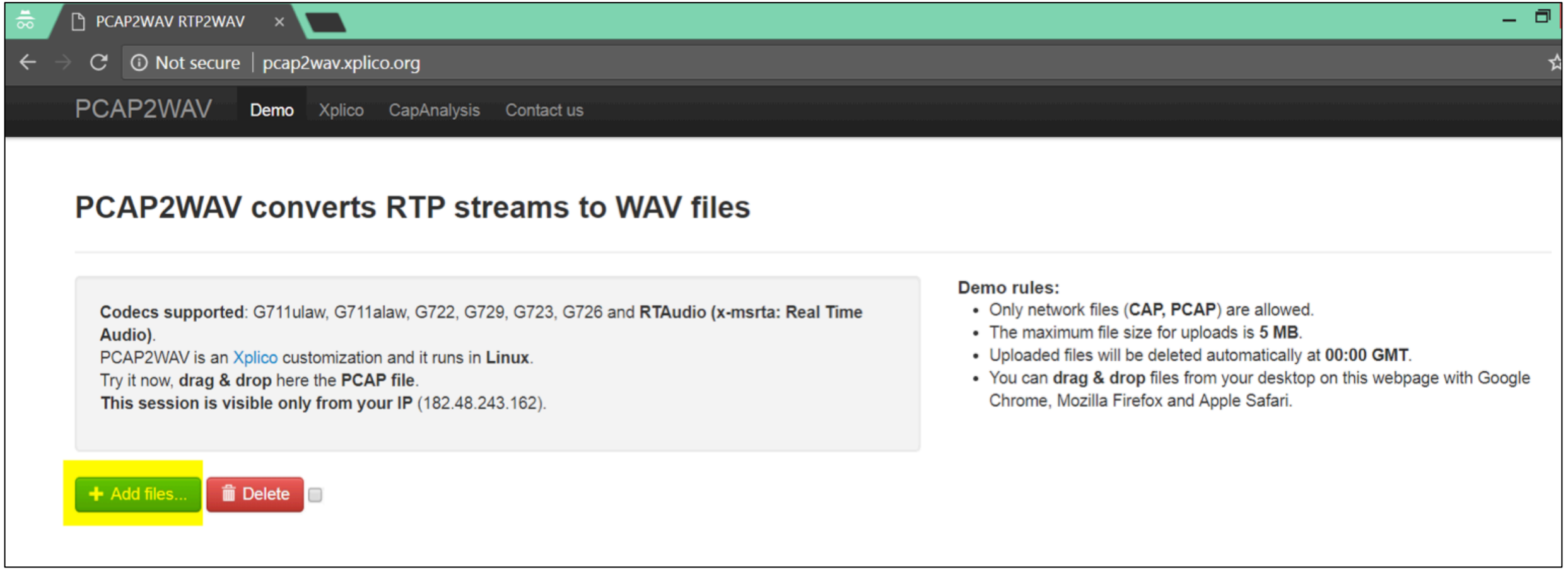
©PentesterAcademy.com

SIP消息



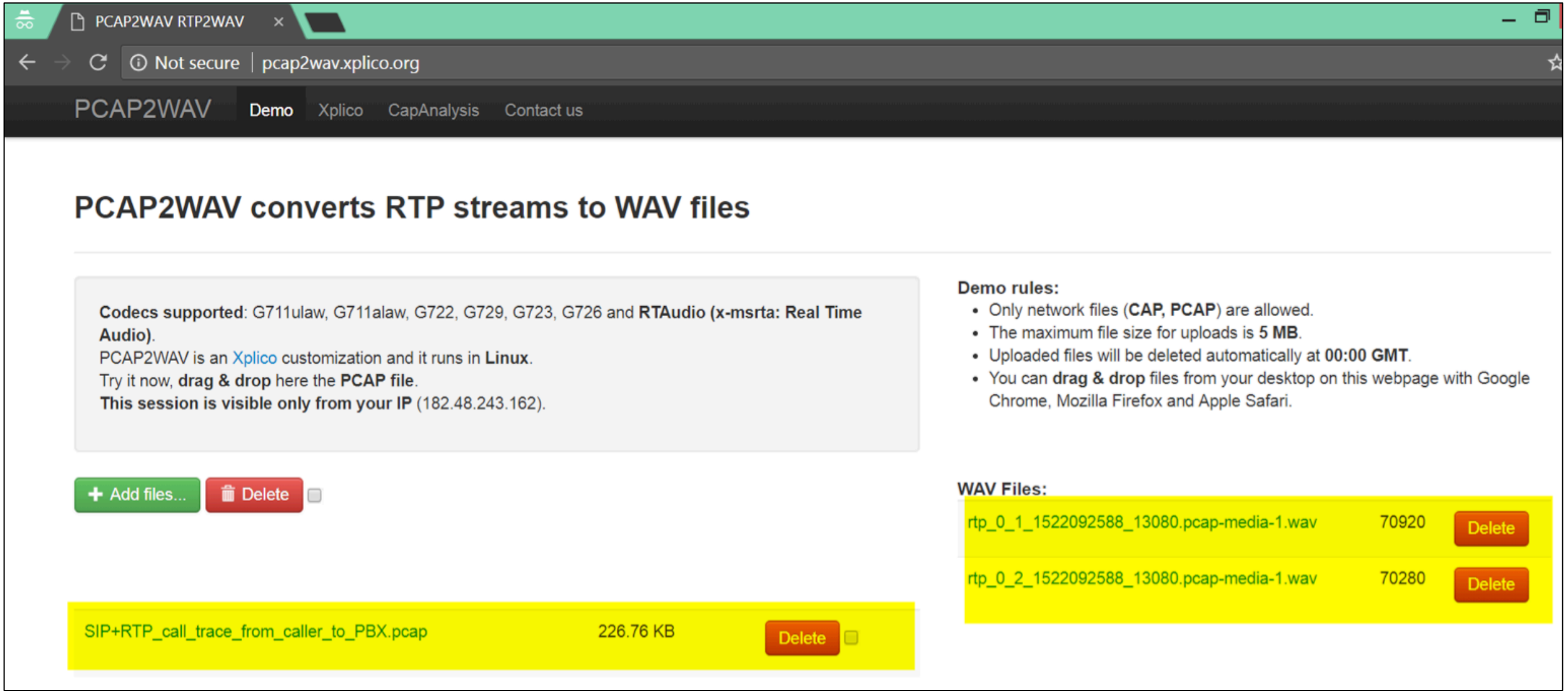
©PentesterAcademy.com

PCAP2WAV: 在线工具



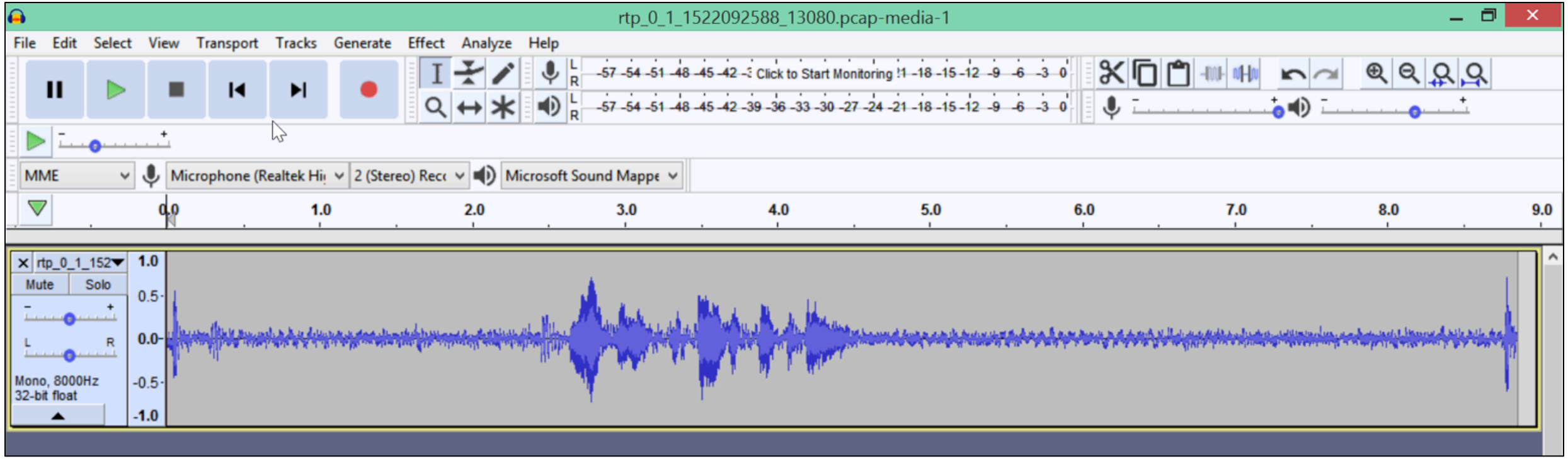
©PentesterAcademy.com

PCAP2WAV: 上传PCAP并下载Wav



©PentesterAcademy.com

PCAP2WAV: audacity中显示的Wav文件



©PentesterAcademy.com

PCAP2WAV: 离线脚本

- Bash脚本用于从VoIP通话中提取音频
- Outputs .wav为输出文件
- 使用tshark和sox
- GitHub: https://gist.github.com/avimar/d2e9d05e082ce273962d742eb9acac16

©PentesterAcademy.com

PCAP2WAV: 帮助信息

```
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 (requres 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.
```

©PentesterAcademy.com

PCAP2WAV: 安装tshark和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) ...
```

©PentesterAcademy.com

PCAP2WAV: 运行工具

```
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/Captu
reSetup/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/Captu
reSetup/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: 0x4fcef51a / 4004

Extracting payloads 1 from 0x0fbb0c8d...
Extracting payloads 2 from 0x4fcef51a...
Combining 2 streams into a single wav file for convenience
No clean option specified - leaving .<codec> and .wav files on system.
```



©PentesterAcademy.com

PCAP2WAV: 目录内容

- 运行脚本前的目录内容

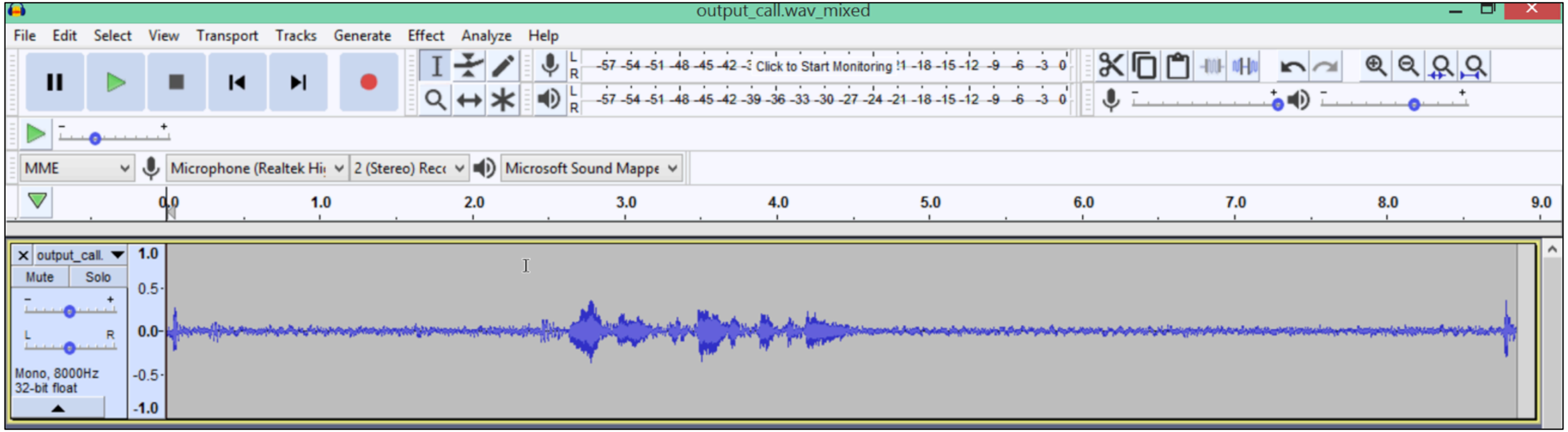
```
root@PentesterAcademy:/work/pcap2wav# ls -l
total 232
-rwxr-xr-x 1 root root   5927 Mar 27 01:18 pcap2wav.sh
-rw------- 1 root root 226760 Mar 19 17:29 SIP+RTP_call_trace_from_caller_to_PBX.pcap
```

- 运行脚本后的目录内容

```
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
```

©PentesterAcademy.com

PCAP2WAV: audacity中显示的Wav文件



©PentesterAcademy.com

VoIPShark

- 一系列Wireshark插件的集合
  - 解密VoIP通话
  - 导出通话音频
  - 流量总览 (扩展, SMS, DTMF)
  - 基本VoIP攻击



- 使用与Wireshark相同的GPL

- Github: github.com/pentesteracademy/voipshark

©PentesterAcademy.com

VoIPShark: 开发背景

- 传统分析过程繁琐且复杂
- 大量的工具
  - 需要编译，设置过程比较耗时
  - 使用起来相对复杂
  - 依赖用户，容易出错
- 解密过程中无法保留时间戳、IP地址等
- 不支持实时流量显示



©PentesterAcademy.com

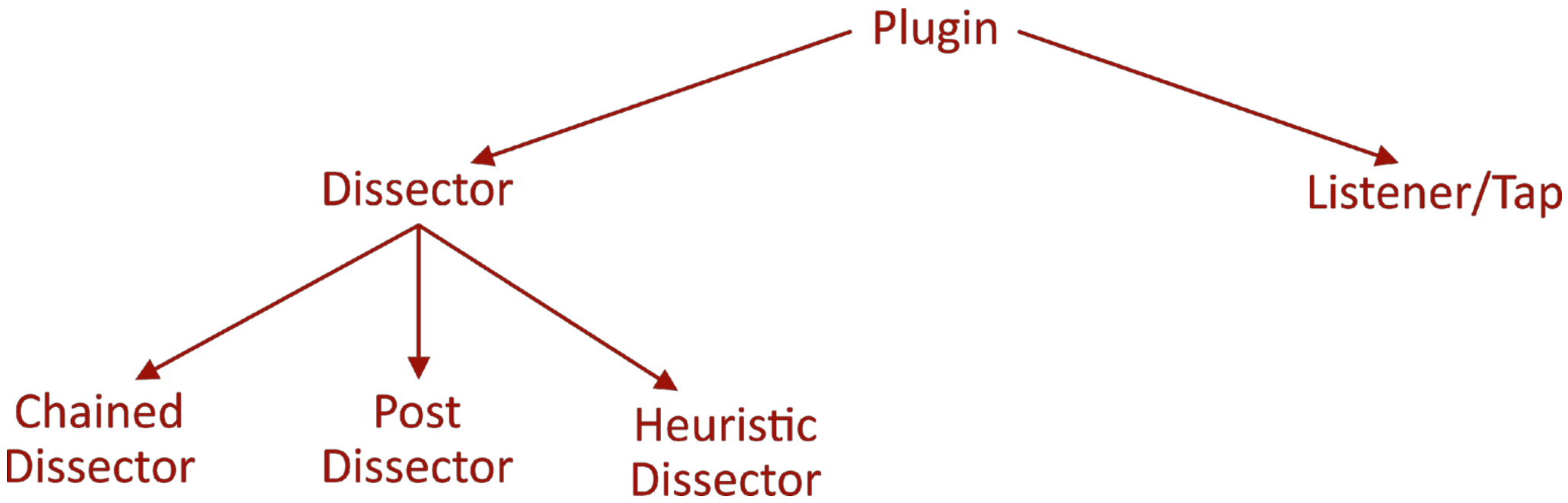
为什么使用wireshark的插件?

- 即插即用
- 插件来源
  - Lua脚本
  - 编译的C/C++代码
- 利用Wireshark的强大功能
- 独立于操作系统
- 牢固的用户基础



©PentesterAcademy.com

Wireshark插件类型



©PentesterAcademy.com

解析器

- 解析载荷数据
- 解析对应部分的协议并把载荷传递给下一个

解析流程图示例

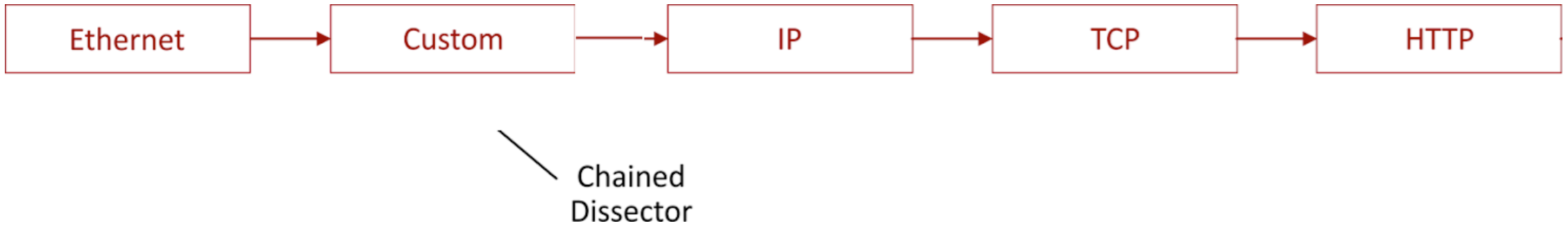


©PentesterAcademy.com

链式解析器

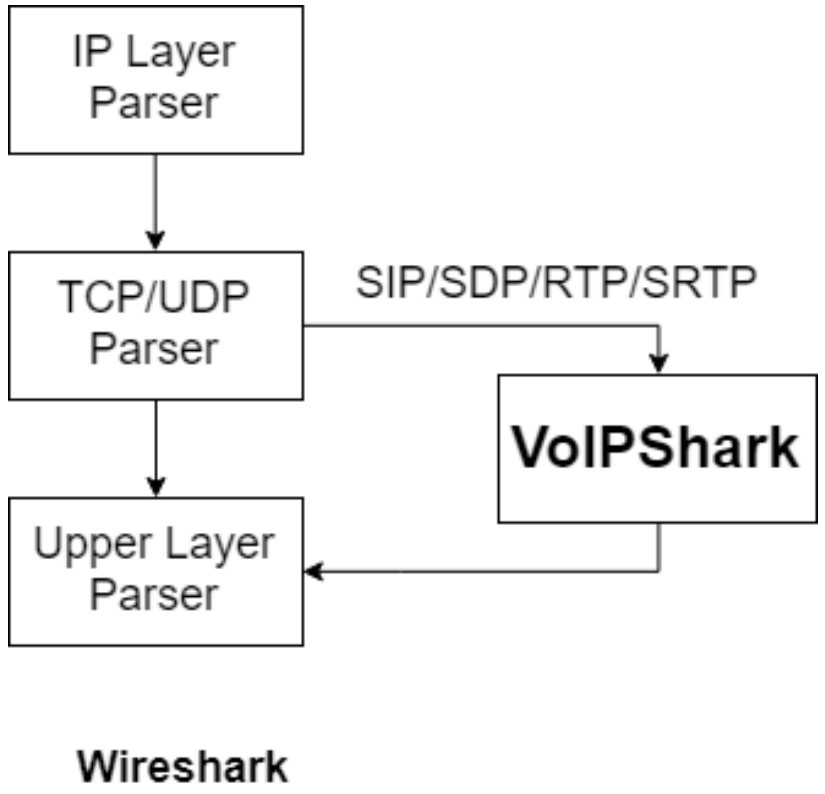
- 从前一解析器获取数据，处理对应部分后，传递给下一解析器

解析流程图示例



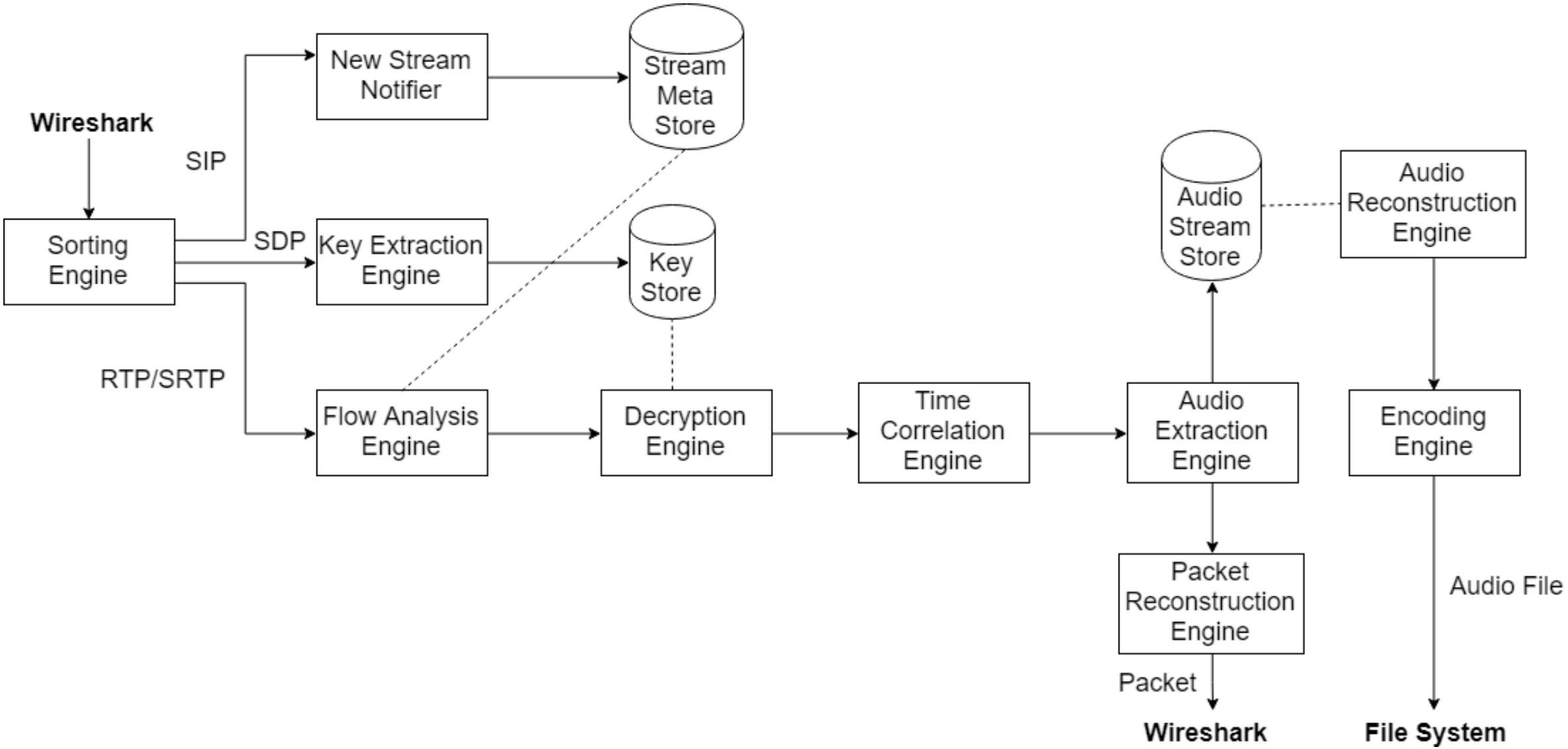
©PentesterAcademy.com

VoIPShark: 解析链中的钩子



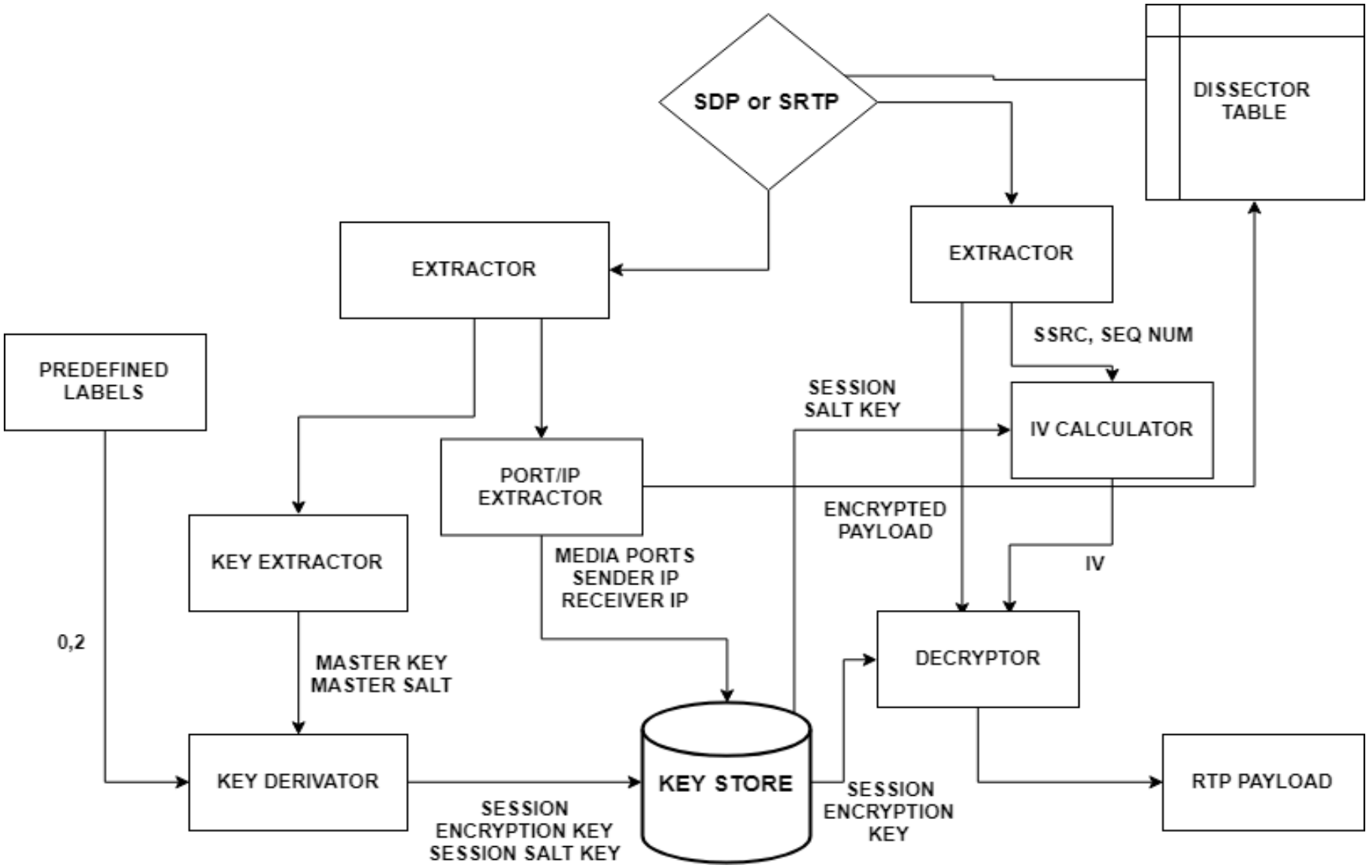
©PentesterAcademy.com

VoIPShark: 总体架构



©PentesterAcademy.com

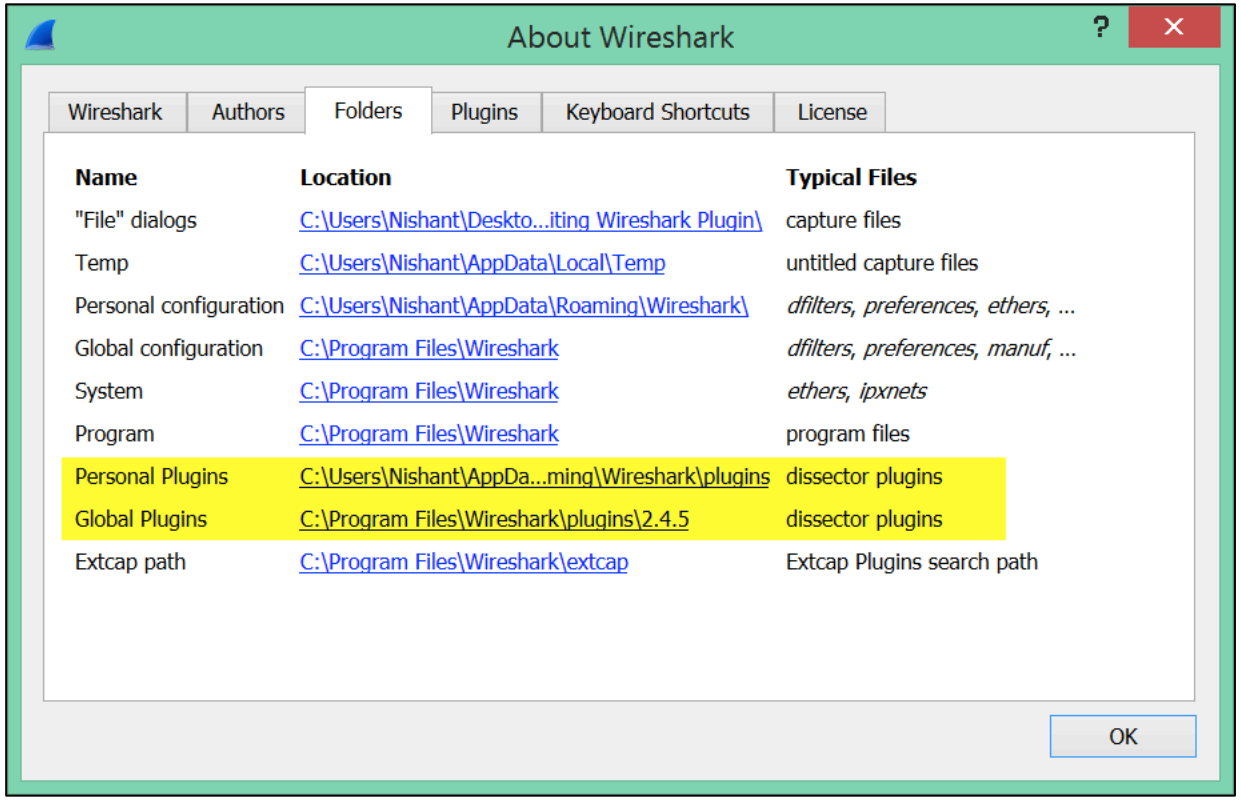
VoIPShark: 解密流程



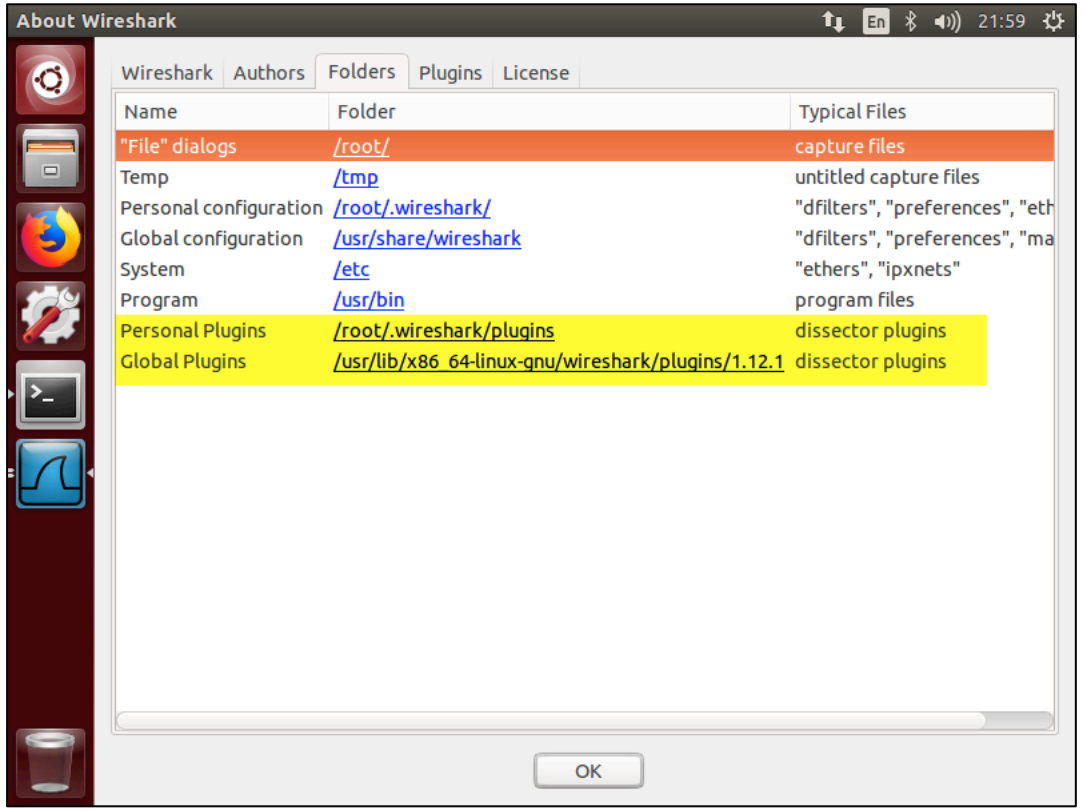
插件地址

- 位置: Help > About Wireshark > Folders

Windows

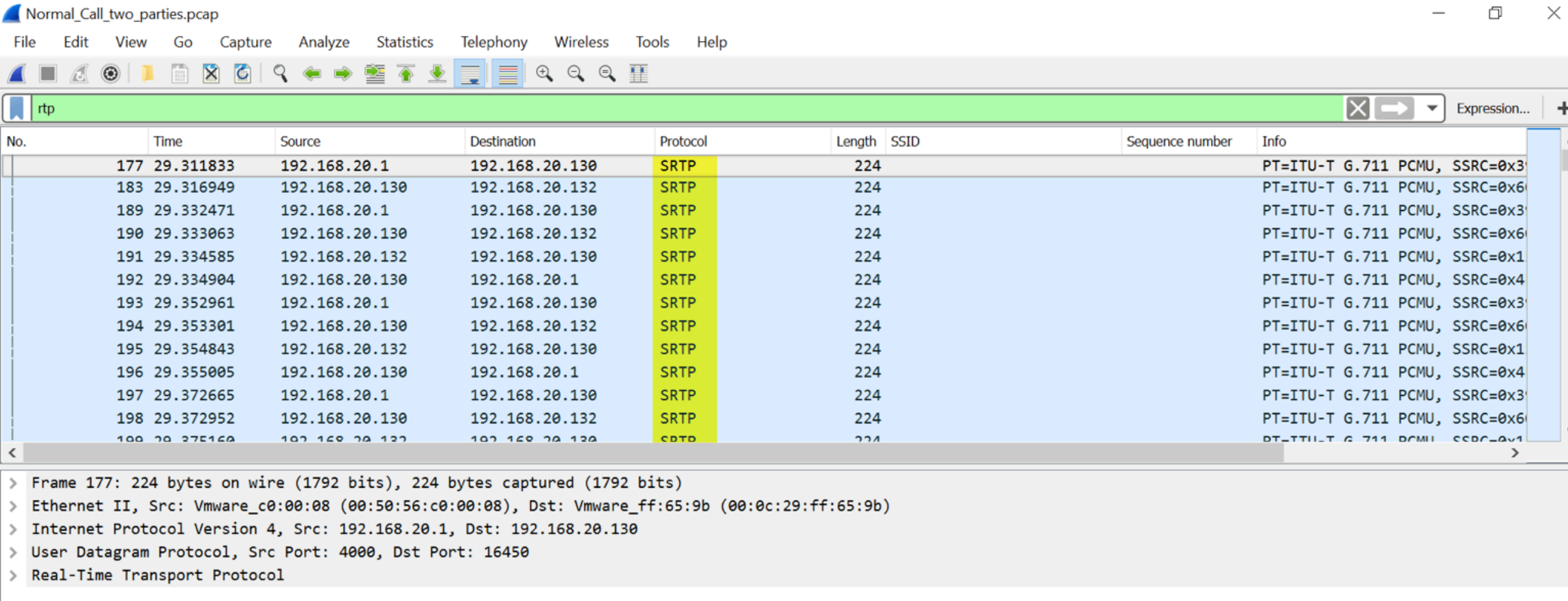


Ubuntu



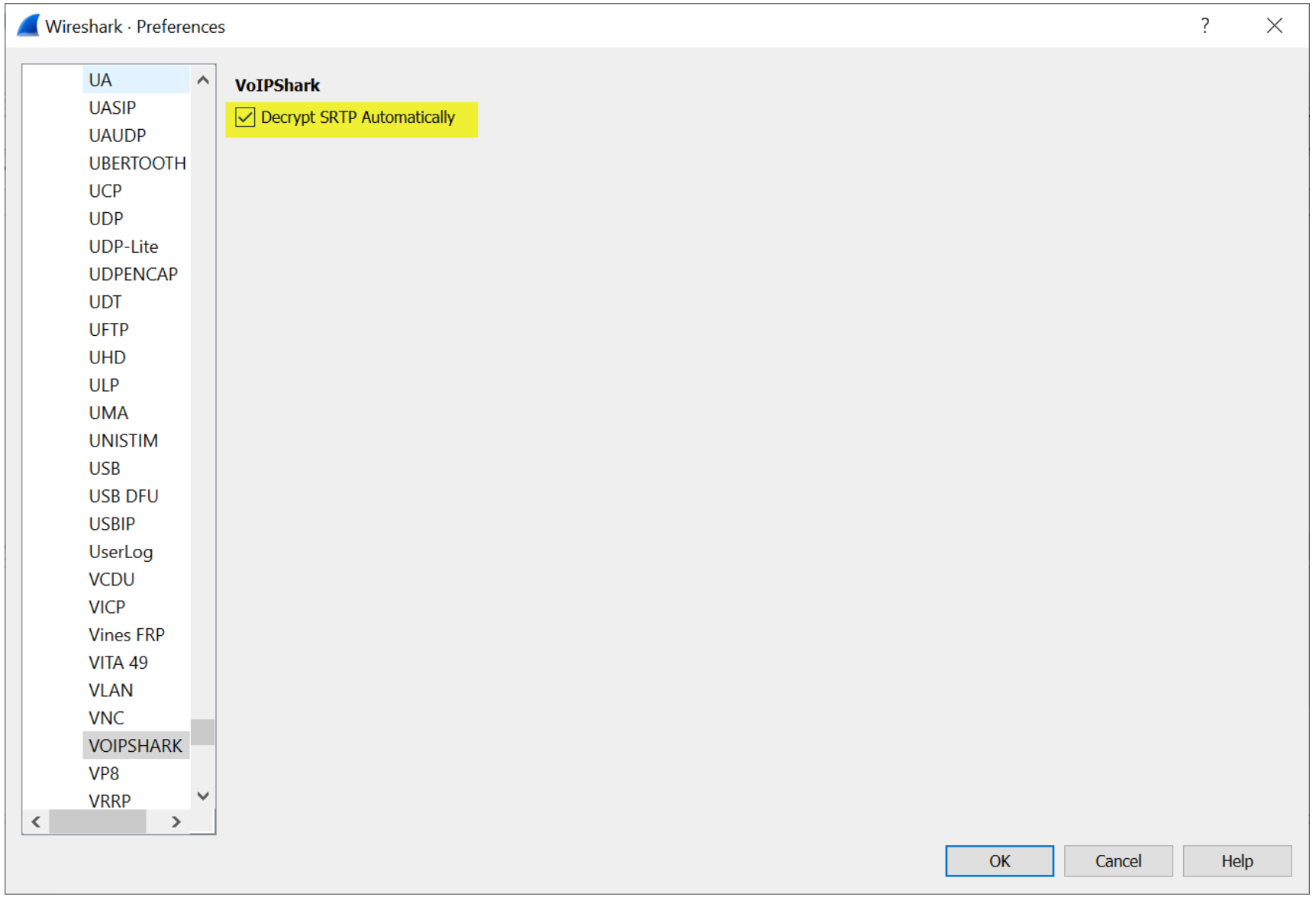
©PentesterAcademy.com

解密SRTP: SRTP数据包

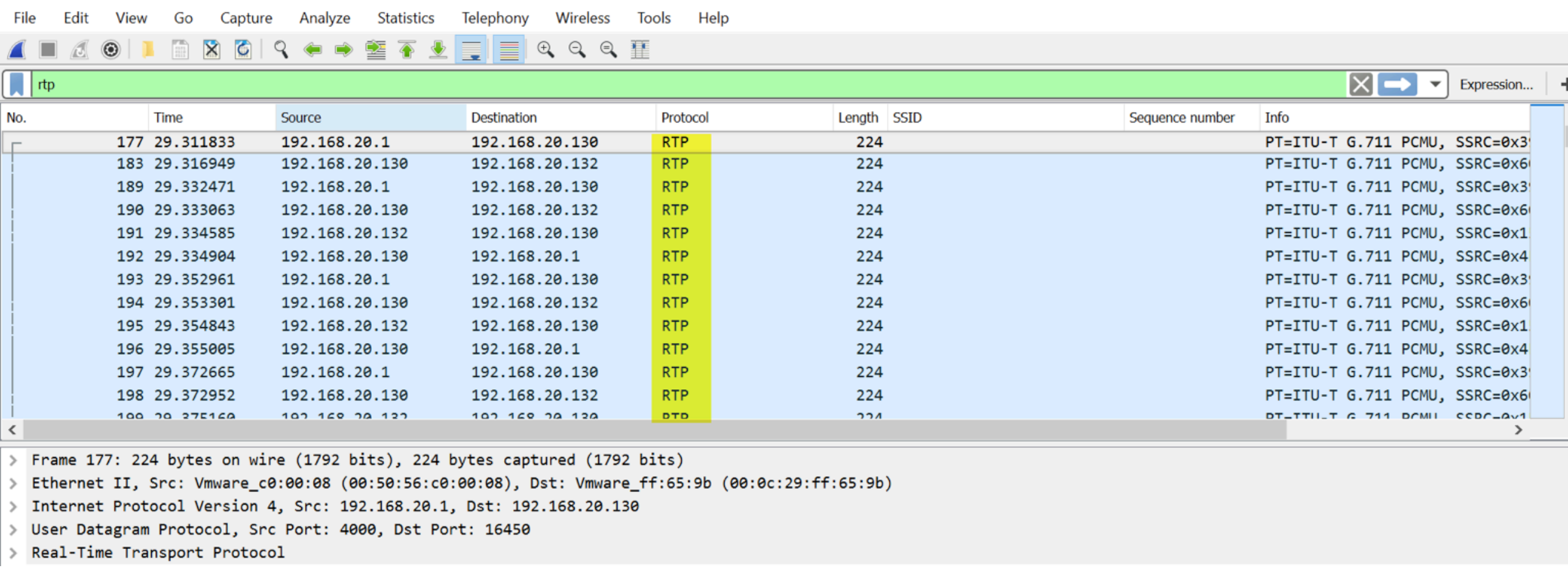


©PentesterAcademy.com

解密SRTP: 开启自动解密

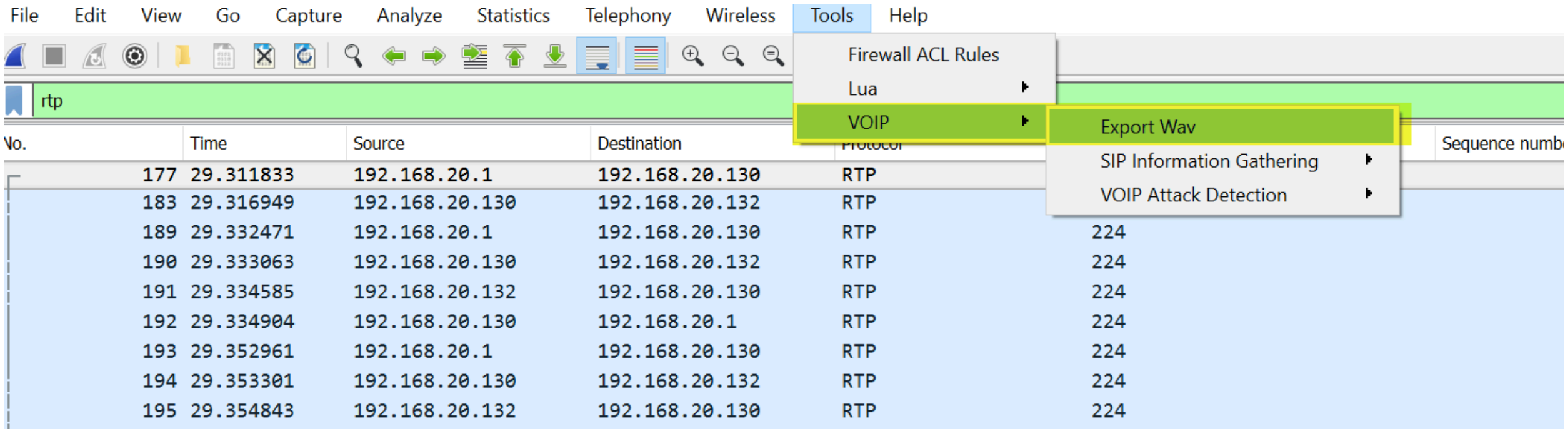


解密SRTP: 解密后的SRTP (RTP)



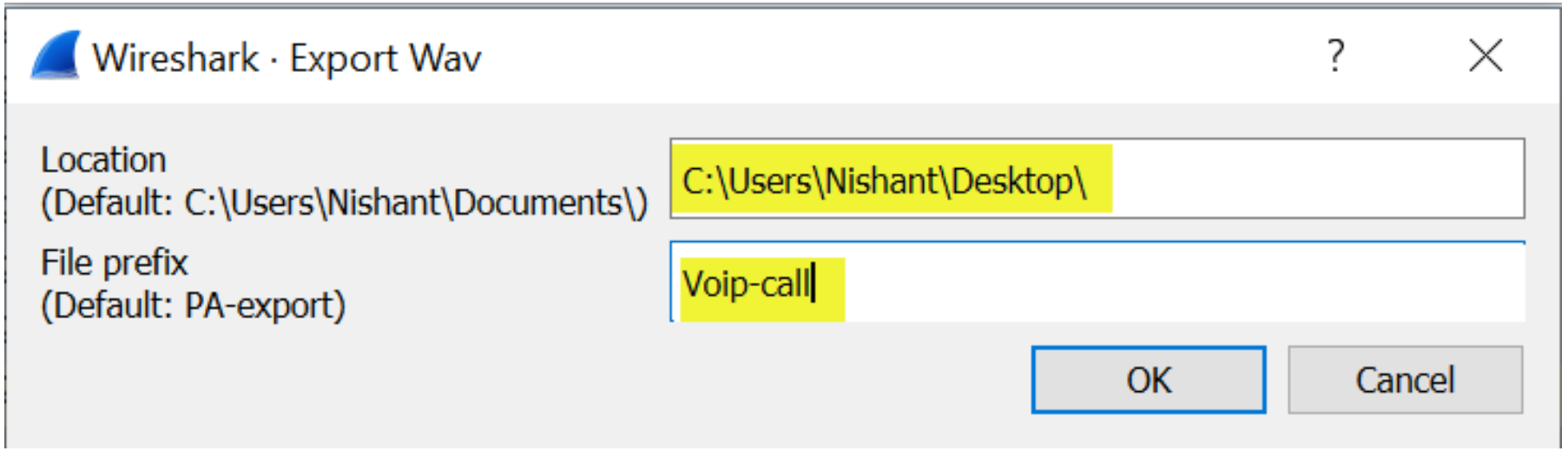
©PentesterAcademy.com

VoIPShark: 导出通话音频



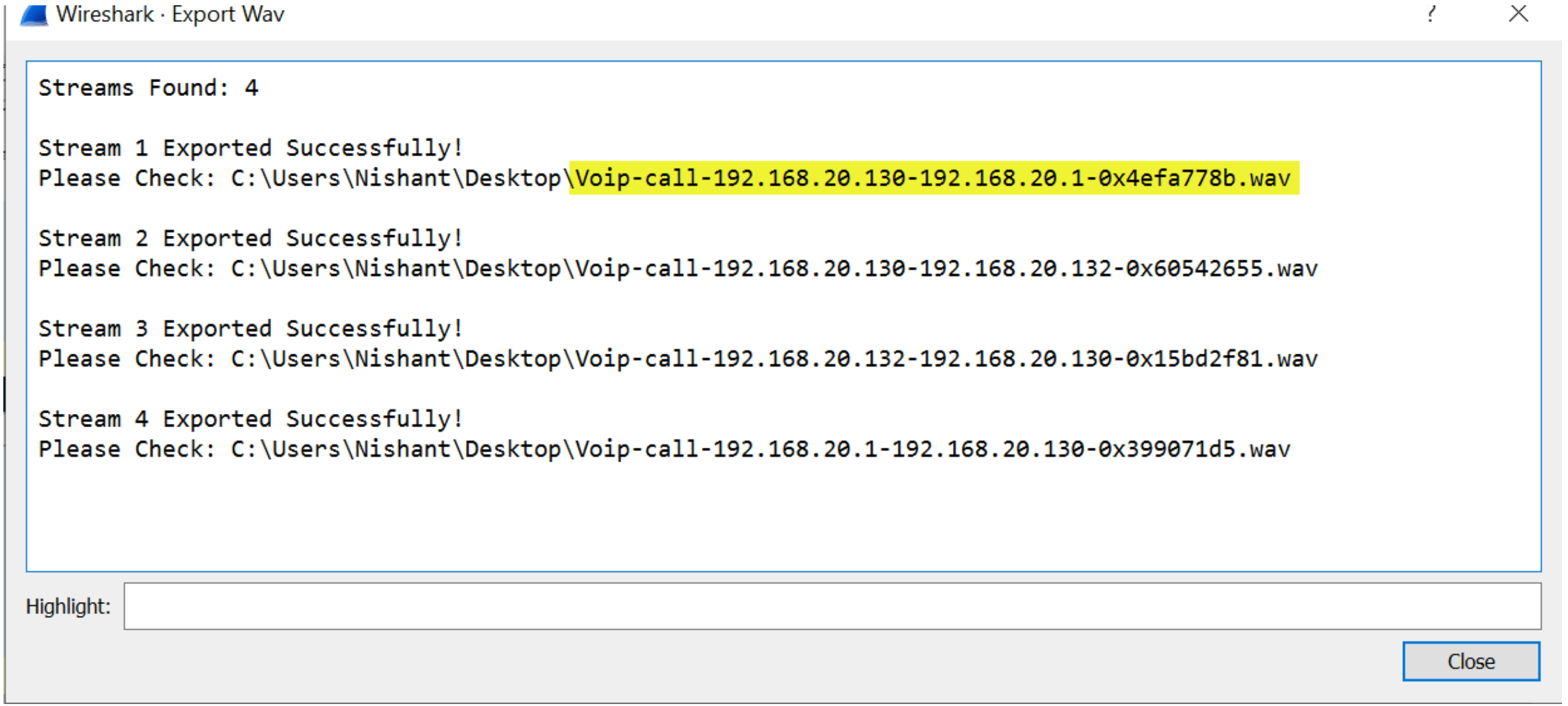
©PentesterAcademy.com

导出通话音频: 指定位置和文件名



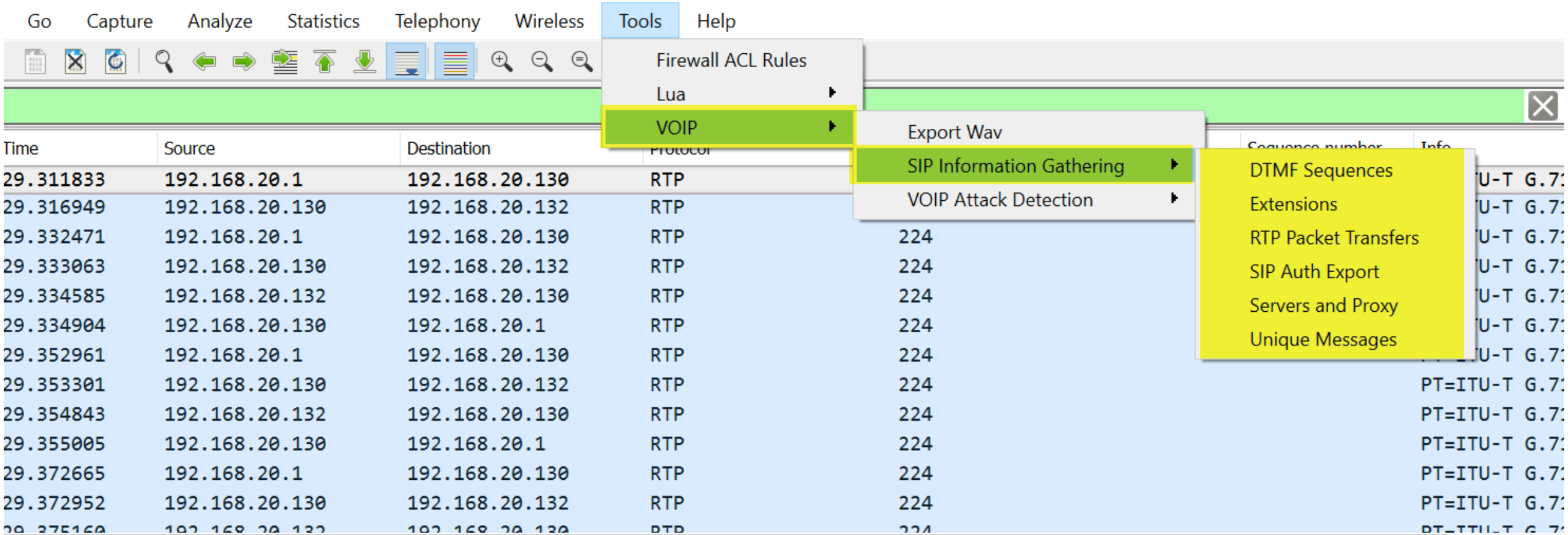
©PentesterAcademy.com

导出通话音频: 导出流



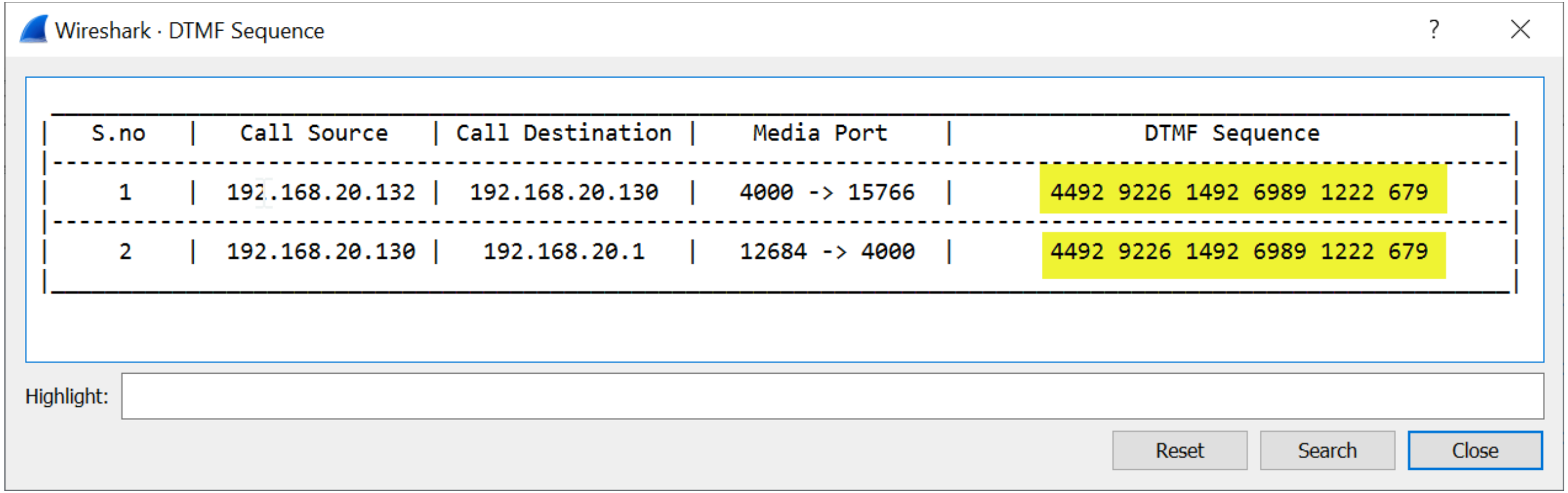
©PentesterAcademy.com

VoIPShark: SIP信息收集



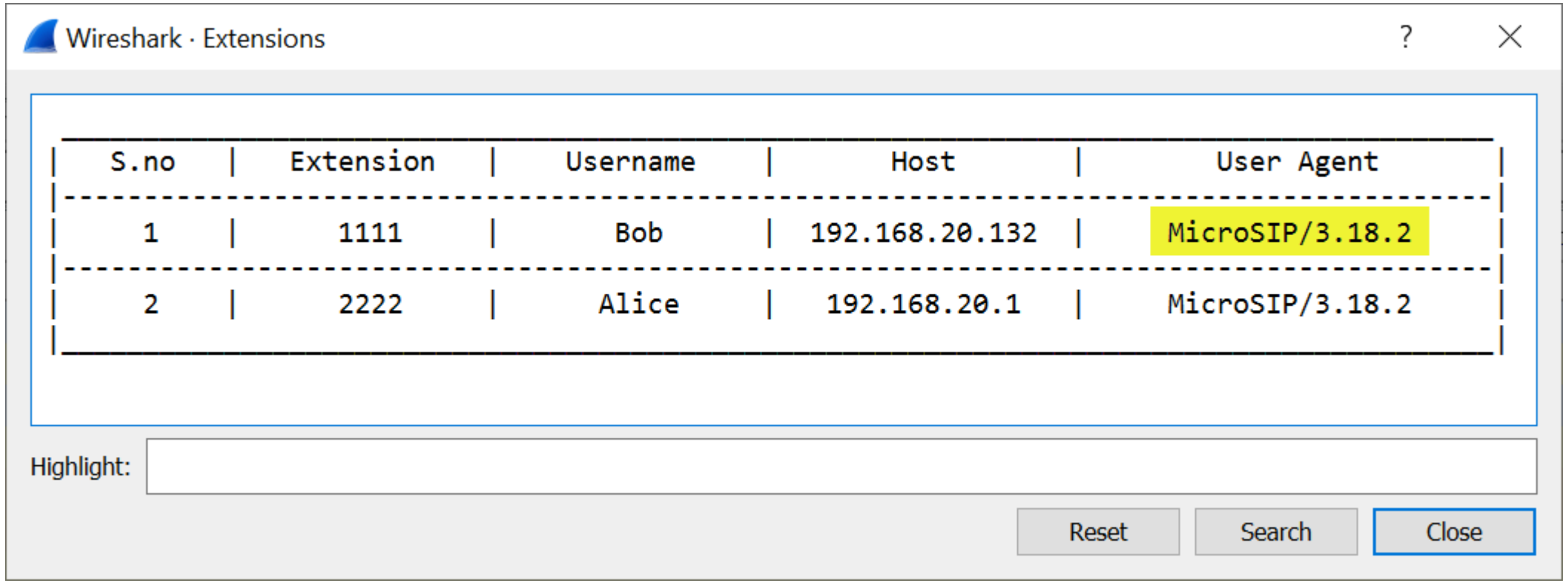
©PentesterAcademy.com

SIP信息收集: DTMF



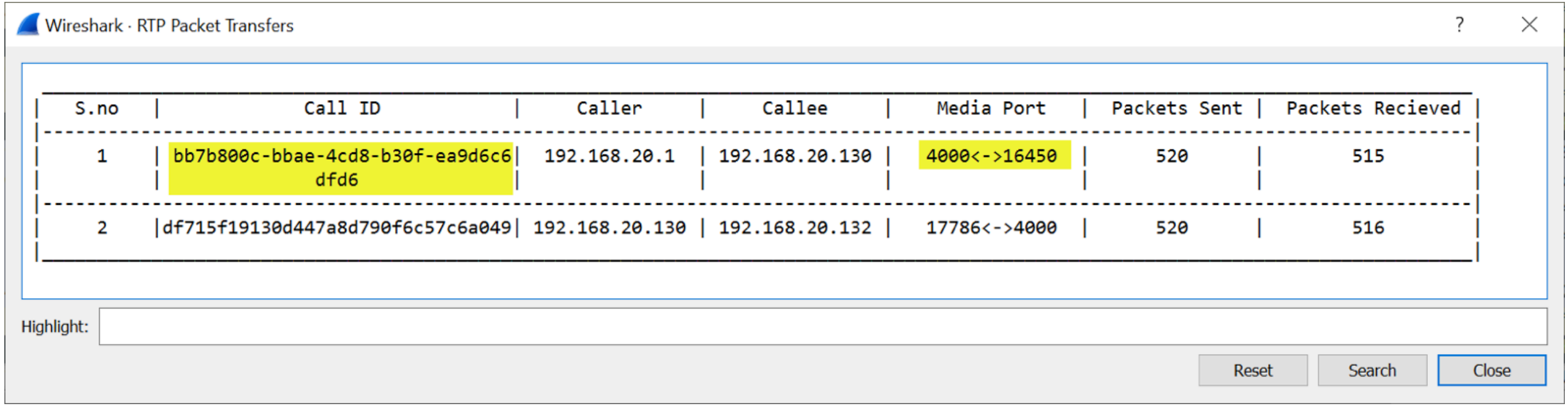
©PentesterAcademy.com

SIP信息收集: 扩展



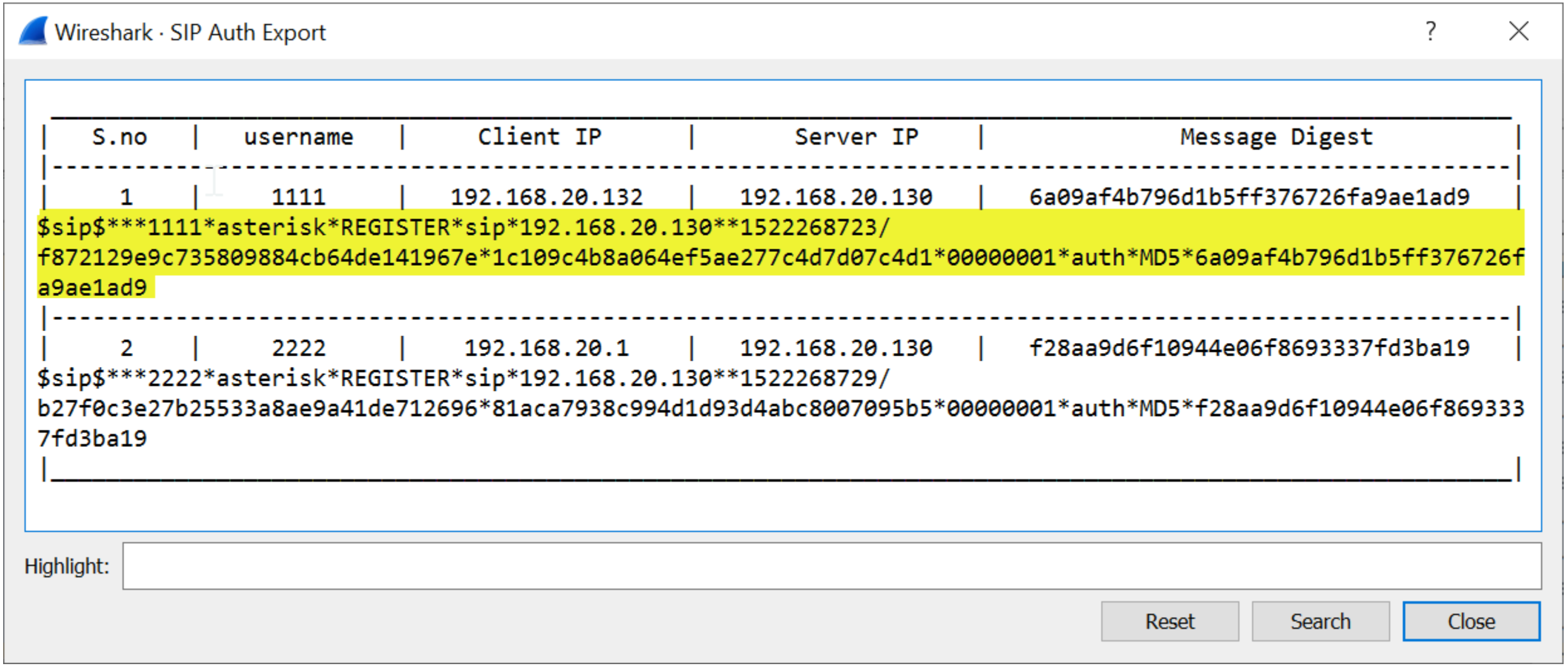
©PentesterAcademy.com

SIP信息收集: RTP数据包传送



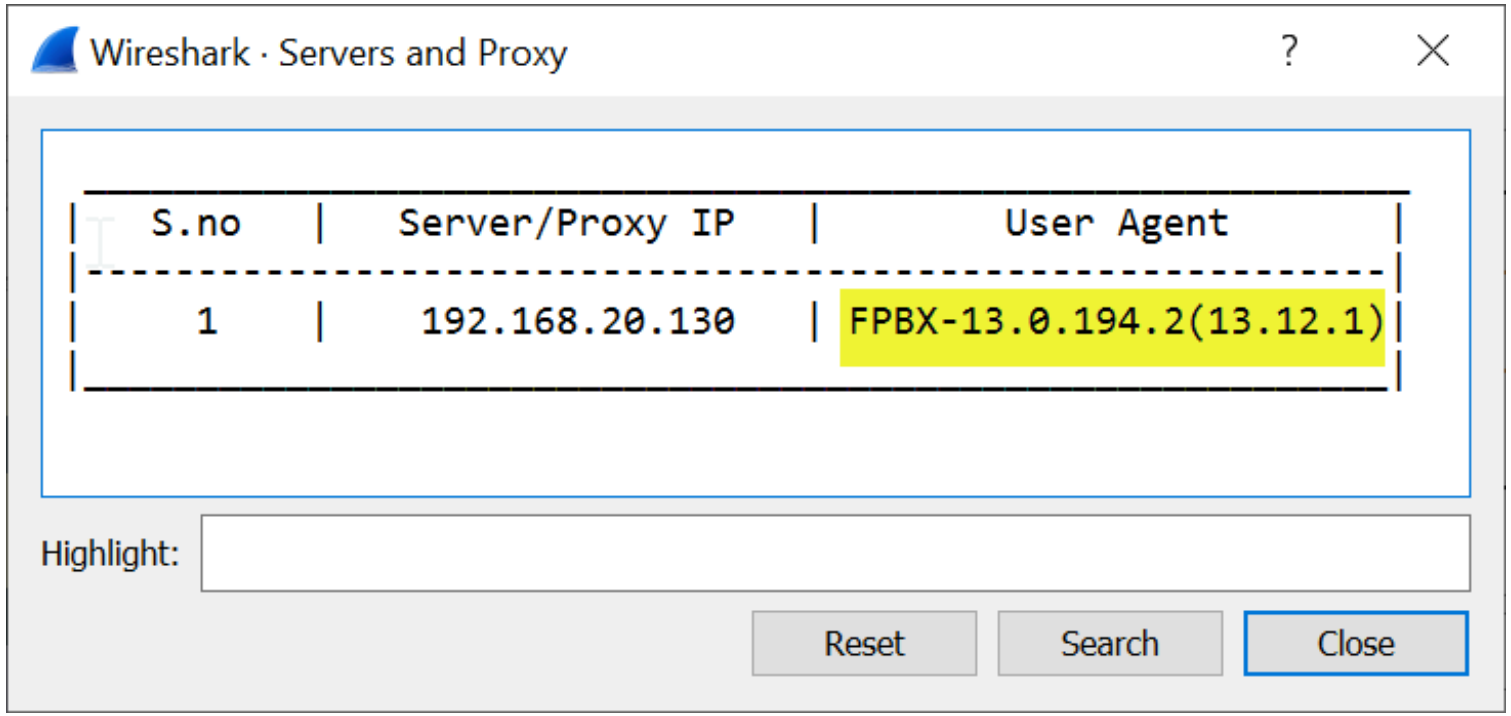
©PentesterAcademy.com

SIP信息收集: SIP认证导出



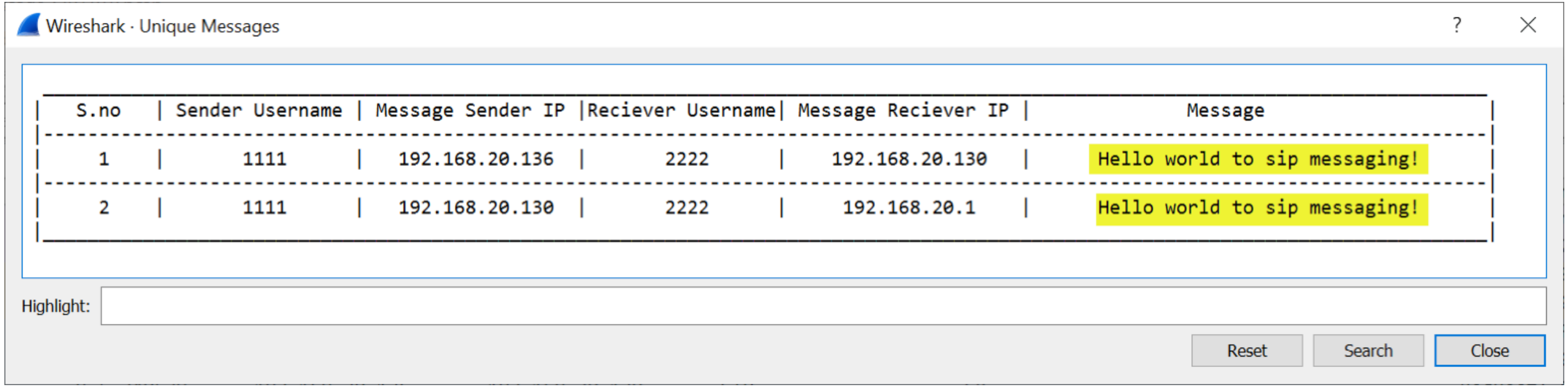
©PentesterAcademy.com

SIP信息收集: 服务器和代理



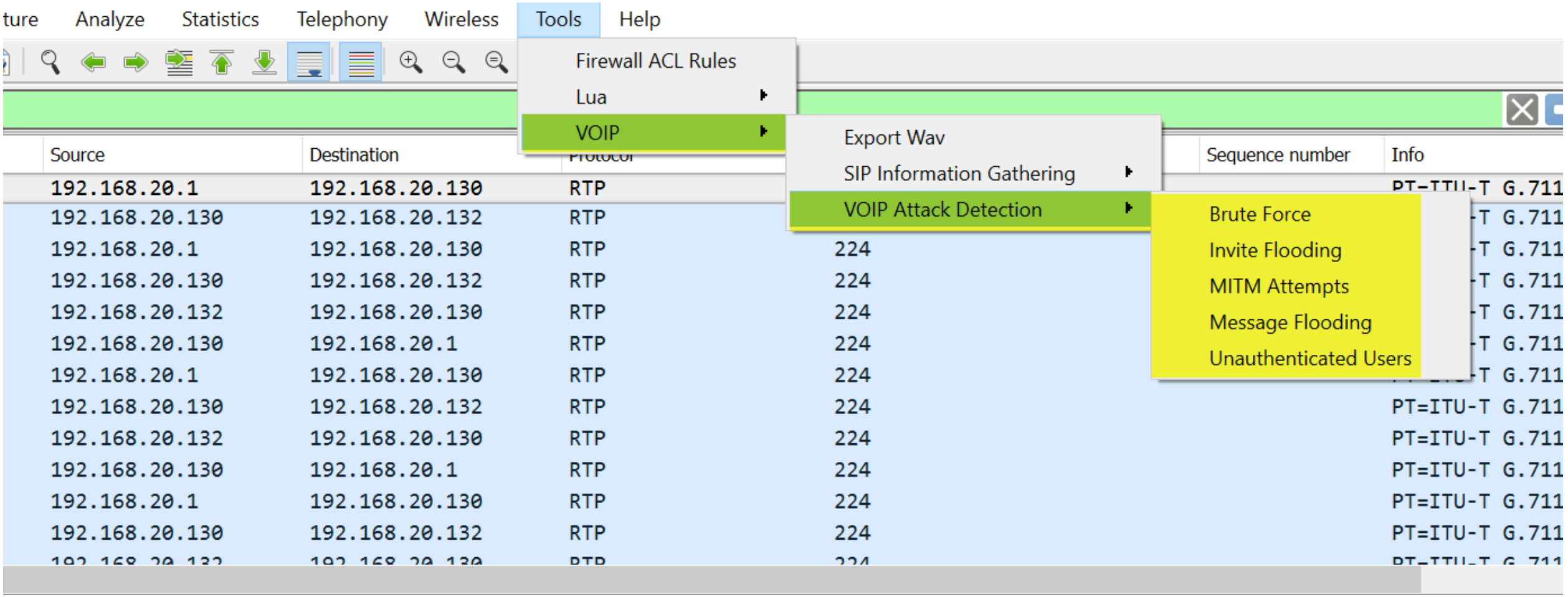
©PentesterAcademy.com

SIP信息收集: 消息



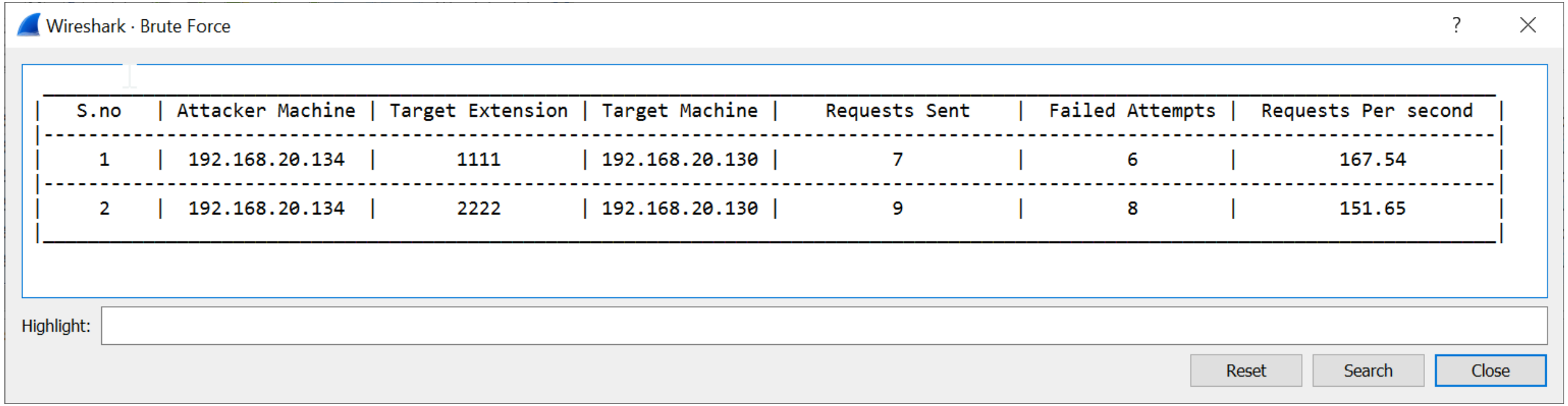
©PentesterAcademy.com

VoIPShark: VoIP攻击检测



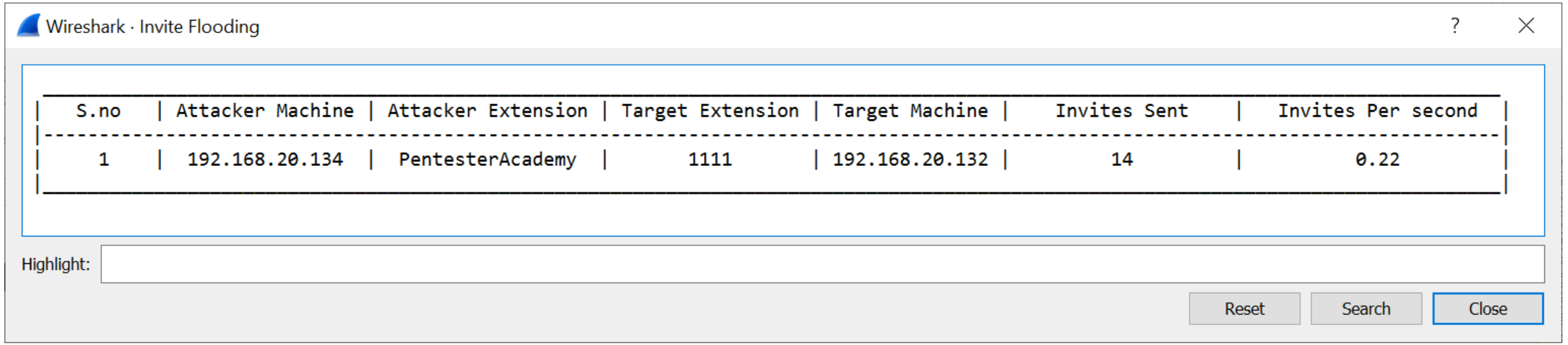
©PentesterAcademy.com

VoIP攻击检测: 暴力破解



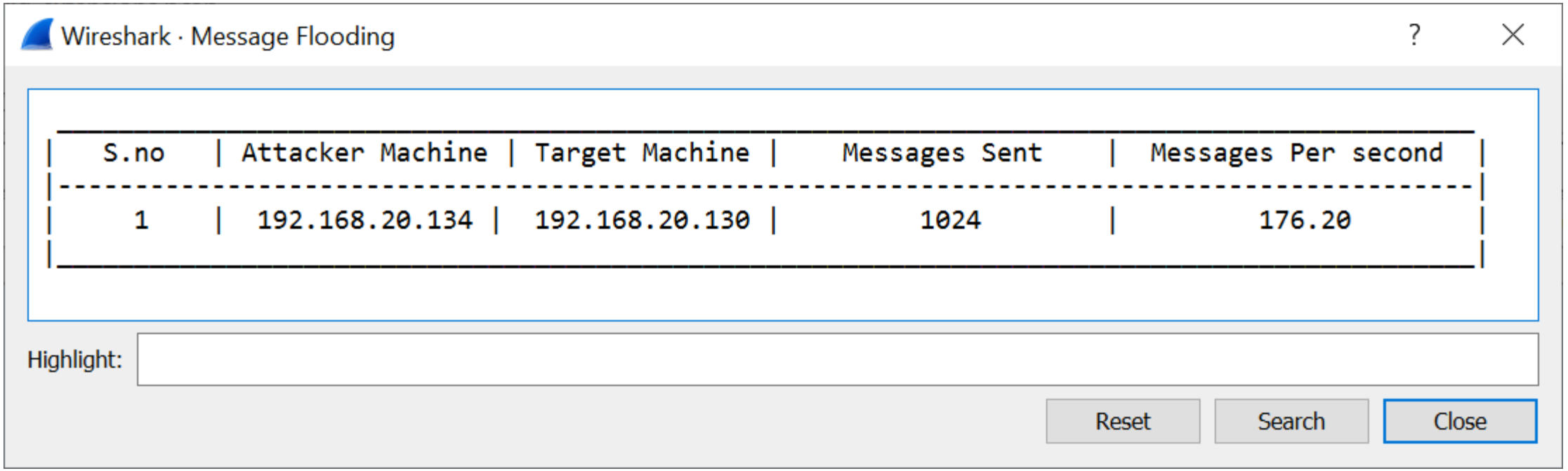
©PentesterAcademy.com

VoIP攻击检测: Invite泛洪攻击



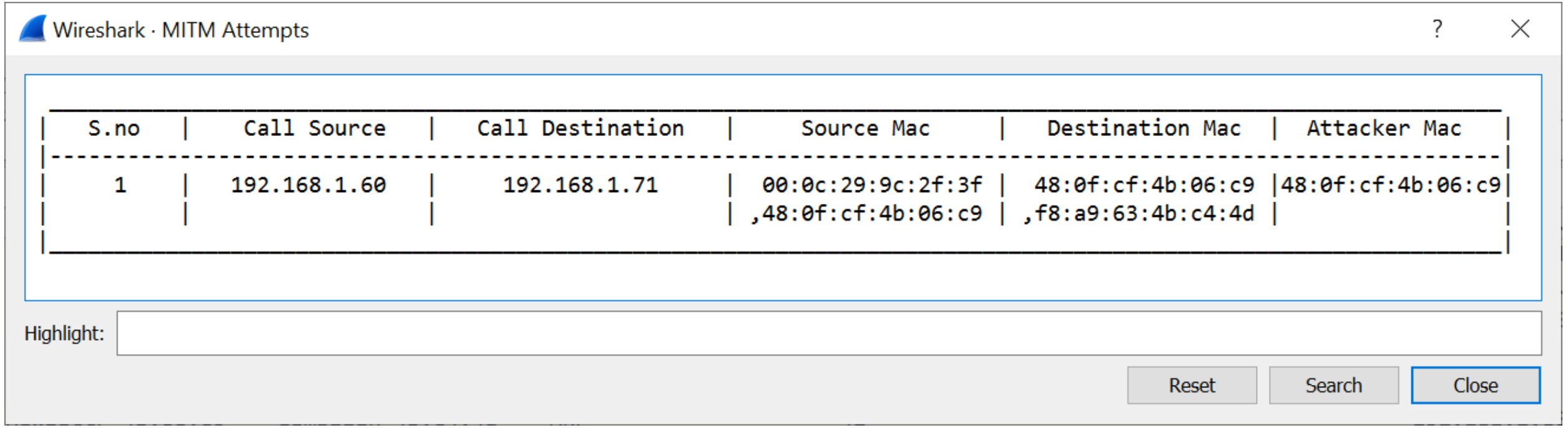
©PentesterAcademy.com

VoIP攻击检测: 报文泛洪



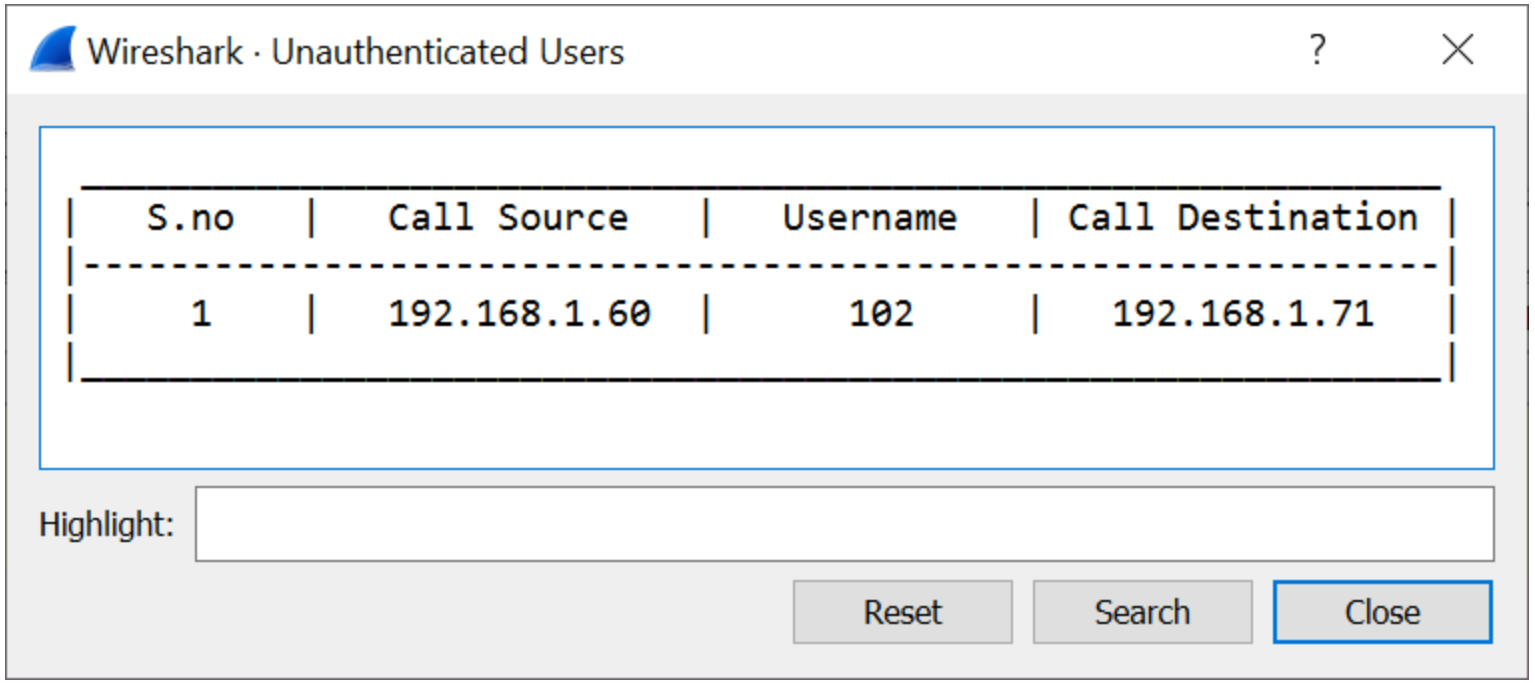
©PentesterAcademy.com

VoIP攻击检测: 中间人攻击



©PentesterAcademy.com

VoIP攻击检测: 未认证用户



©PentesterAcademy.com

演示

©PentesterAcademy.com

提问环节

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

©PentesterAcademy.com