WeLCoMe Back To ComSec !!!

Wireshark

66

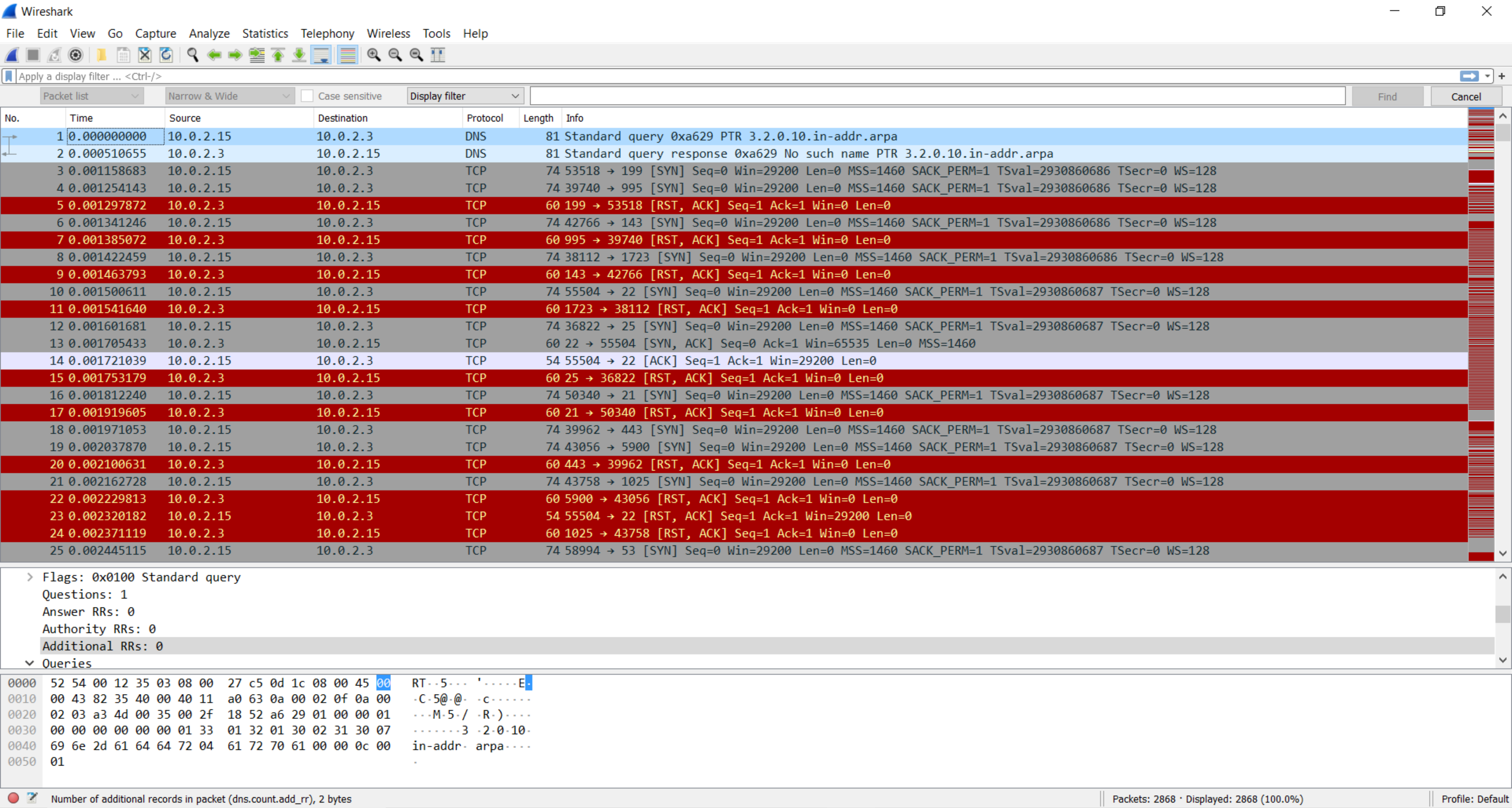
What is Wireshark?

- Is **NOT** sharks and wires
- It **IS** a Packet Sniffer
- *“Wireshark is the world’s foremost and widely-used network protocol analyzer. It lets you see what’s happening on your network at a microscopic level and is the de facto (and often de jure) standard across many commercial and non-profit enterprises, government agencies, and educational institutions. Wireshark development thrives thanks to the volunteer contributions of networking experts around the globe and is the continuation of a project started by Gerald Combs in 1998.”*

What can it be used for?

- **Deep inspection** of hundreds of protocols, with more being added all the time
- **Live capture** and **offline analysis**
- Standard three-pane **packet browser**
- Multi-platform: Runs on Windows, Linux, macOS, Solaris, FreeBSD, NetBSD, and many others
- Captured network data can be browsed via a GUI, or via the TTY-mode TShark utility
- The most powerful **display filters** in the industry
- Rich **VoIP analysis**
- Read/write many different capture file formats: tcpdump (libpcap), Pcap NG, Catapult DCT2000, Cisco Secure IDS iplog, Microsoft Network Monitor, Network General Sniffer® (compressed and uncompressed), Sniffer® Pro, and NetXray®, Network Instruments Observer, NetScreen snoop, Novell LANalyzer, RADCOM WAN/LAN Analyzer, Shomiti/Finisar Surveyor, Tektronix K12xx, Visual Networks Visual UpTime, WildPackets EtherPeek/TokenPeek/AiroPeek, and many others
- Capture files **compressed** with gzip can be decompressed on the fly
- **Live data** can be read from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI, and others (depending on your platform)
- **Decryption support** for many protocols, including IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP, and WPA/WPA2
- **Coloring rules** can be applied to the packet list for quick, intuitive analysis
- **Output can be exported** to XML, PostScript®, CSV, or plain text

Live mode and analytical mode? (IS IT the right spelling because I don’t know lmao) idk



```
No.  Time         Source      Destination  Protocol  Length  Info
   1 0.000000000  10.0.2.15   10.0.2.3     DNS       81 Standard query 0xa629 PTR 3.2.0.10.in-addr.arpa
   2 0.000510655  10.0.2.3    10.0.2.15    DNS       81 Standard query response 0xa629 No such name PTR 3.2.0.10.in-addr.arpa
   3 0.001158683  10.0.2.15   10.0.2.3     TCP       74 53518 → 199 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2930860686 TSecr=0 WS=128
   4 0.001254143  10.0.2.15   10.0.2.3     TCP       74 39740 → 995 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2930860686 TSecr=0 WS=128
   5 0.001297872  10.0.2.3    10.0.2.15    TCP       60 199 → 53518 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   6 0.001341246  10.0.2.15   10.0.2.3     TCP       74 42766 → 143 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2930860686 TSecr=0 WS=128
   7 0.001385072  10.0.2.3    10.0.2.15    TCP       60 995 → 39740 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   8 0.001422459  10.0.2.15   10.0.2.3     TCP       74 38112 → 1723 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2930860686 TSecr=0 WS=128
   9 0.001463793  10.0.2.3    10.0.2.15    TCP       60 143 → 42766 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  10 0.001500611  10.0.2.15   10.0.2.3     TCP       74 55504 → 22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2930860687 TSecr=0 WS=128
  11 0.001541640  10.0.2.3    10.0.2.15    TCP       60 1723 → 38112 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  12 0.001601681  10.0.2.15   10.0.2.3     TCP       74 36822 → 25 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2930860687 TSecr=0 WS=128
  13 0.001705433  10.0.2.3    10.0.2.15    TCP       60 22 → 55504 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
  14 0.001721039  10.0.2.15   10.0.2.3     TCP       54 55504 → 22 [ACK] Seq=1 Ack=1 Win=29200 Len=0
  15 0.001753179  10.0.2.3    10.0.2.15    TCP       60 25 → 36822 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  16 0.001812240  10.0.2.15   10.0.2.3     TCP       74 50340 → 21 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2930860687 TSecr=0 WS=128
  17 0.001919605  10.0.2.3    10.0.2.15    TCP       60 21 → 50340 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  18 0.001971053  10.0.2.15   10.0.2.3     TCP       74 39962 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2930860687 TSecr=0 WS=128
  19 0.002037870  10.0.2.15   10.0.2.3     TCP       74 43056 → 5900 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2930860687 TSecr=0 WS=128
  20 0.002100631  10.0.2.3    10.0.2.15    TCP       60 443 → 39962 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  21 0.002162728  10.0.2.15   10.0.2.3     TCP       74 43758 → 1025 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2930860687 TSecr=0 WS=128
  22 0.002229813  10.0.2.3    10.0.2.15    TCP       60 5900 → 43056 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  23 0.002320182  10.0.2.15   10.0.2.3     TCP       54 55504 → 22 [RST, ACK] Seq=1 Ack=1 Win=29200 Len=0
  24 0.002371119  10.0.2.3    10.0.2.15    TCP       60 1025 → 43758 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
  25 0.002445115  10.0.2.15   10.0.2.3     TCP       74 58994 → 53 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2930860687 TSecr=0 WS=128
```

```
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
```

```
0000  52 54 00 12 35 03 08 00  27 c5 0d 1c 08 00 45 00   RT··5··· '·····E·
0010  00 43 82 35 40 00 40 11  a0 63 0a 00 02 0f 0a 00   ·C·5@·@· ·c······
0020  02 03 a3 4d 00 35 00 2f  18 52 a6 29 01 00 00 01   ···M·5·/ ·R·)····
0030  00 00 00 00 00 00 01 33  01 32 01 30 02 31 30 07   ·······3 ·2·0·10·
0040  69 6e 2d 61 64 64 72 04  61 72 70 61 00 00 0c 00   in-addr· arpa····
0050  01                                                 ·
```

Number of additional records in packet (dns.count.add_rr), 2 bytes

Packets: 2868 · Displayed: 2868 (100.0%)

Profile: Default

Packets in Wireshark

- Number of packet
- Time offset from start of capture
- Source IP and destination IP
- Protocol used (e.g. DNS, TCP, HTTP)
- Size of packet in **bytes**
- Information about the packet

- Packet information
- Hex Dump of packet

Display Filters

- NOT CAPTURE FILTERS!

- https://wiki.wireshark.org/DisplayFilters
- https://networksecuritytools.com/list-wireshark-display-filters/
- https://www.wireshark.org/docs/dfref/

Searching

- Display filter
- Hex value
- String
- Regex

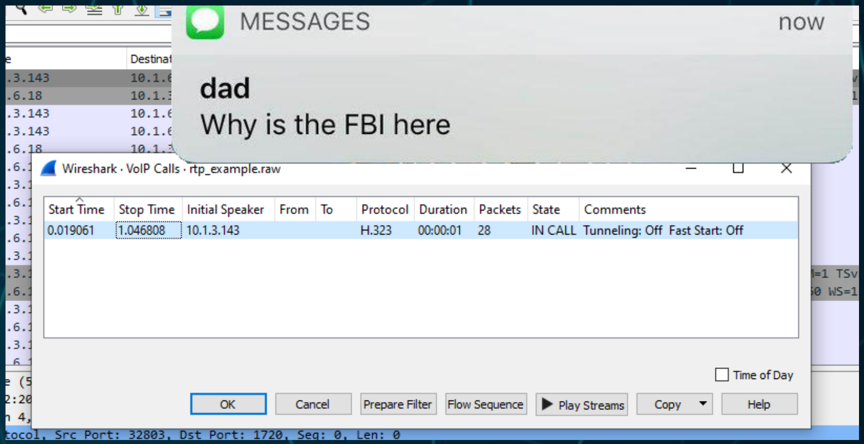
- Search in different parts of the packet and in different encodings

Exporting stuff

- Single packet
- Parts of packets
- PDUs
- TLS session keys
- Objects
  - DICOM
  - HTTP
  - IMF
  - SMB
  - TFTP

Telephony

- You can export a lot of different voice protocols
- You can listen to the audio within Wireshark



MESSAGES now

dad
Why is the FBI here

Wireshark · VoIP Calls · rtp_example.raw

Start Time	Stop Time	Initial Speaker	From	To	Protocol	Duration	Packets	State	Comments
0.019061	1.046808	10.1.3.143			H.323	00:00:01	28	IN CALL	Tunneling: Off Fast Start: Off

□ Time of Day

tocol, Src Port: 32803, Dst Port: 1720, Seq: 0, Len: 0

Wireless

- Wireshark supports variety of wireless protocols
- Bluetooth / Wi-Fi / WLAN

More

- Live capture

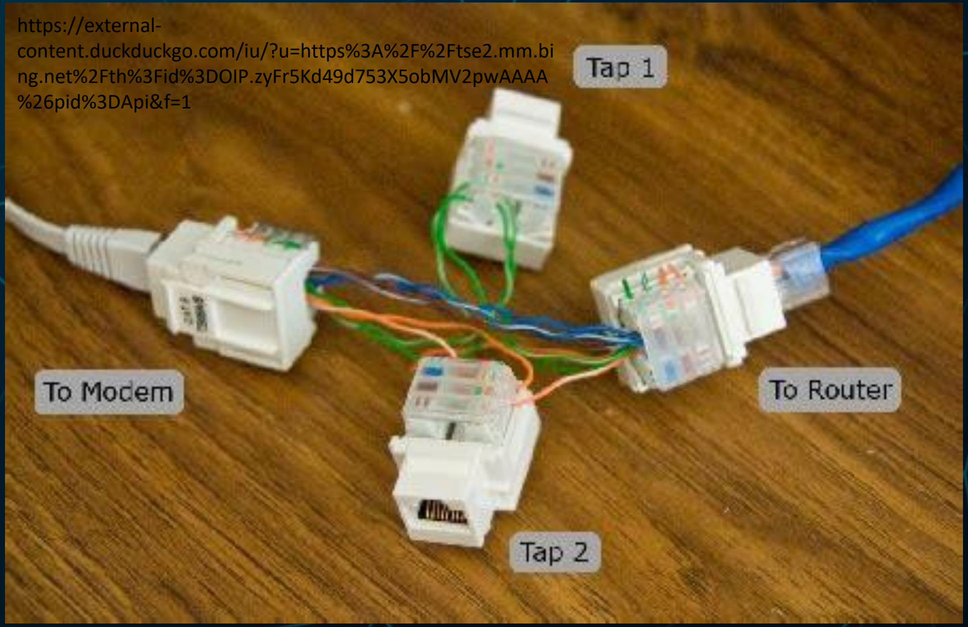
- Statistics tools
- Scripting (Within and as a library)
- Stream following
- Automatic filters (RMB)
- etc

Protocol configs – import/export certs

A fun physical idea (will tidy up tomorrow)

- An ethernet cable, strip wires, connect tx to RX on the third connector, connect rx to RX on the fourth connector.
- Plug cable in as usual.
- Connect both third and fourth to a computer, open wireshark
- You now have a wiretap, for Wireshark :P

- You can also set managed switches to
Mirror traffic to another port and sniff that.



https://external-content.duckduckgo.com/iu/?u=https%3A%2F%2Ftse2.mm.bing.net%2Fth%3Fid%3DOIP.zyFr5Kd49d753X5obMV2pwAAAA%26pid%3DApi&f=1

Excersise!

- Good luck