Wireshark 101

Its all about packets

Robert Hess – GoTo Dresden

robert.hess@goto.com

Material download & instructions:
https://github.com/rohess/ws-training

Why do we talk about Wireshark

- Understand network protocols
- Analyse network problems
- Find out what happens in your network

Wireshark can help you with this – and its fun to use

What is Wireshark

- Tool to capture and analyze network packets (all kinds of)
- OpenSource – you can build it yourself
- Highly customizable
- On MacOS, Linux, Windows (based on QT5)
- ARM version for Windows and Mac
- 1.5 Mio downloads per month
- 3000 protocols, 250k fields
- 2300 authors
- Two yearly conferences

History

- First iteration started by Gerald Combs in 1997 as Ethereal
- Since 2006 called Wireshark
- 2008 V1.0 & first Sharkfest
- 2023 V4.0 & Wireshark Foundation

Today

- Install Wireshark
- Look at the UI
- Filter packets
- Go through a sample capture
- demo the most common features
- Have a short look at how to capture

- → Install instructions & Sample file download: https://github.com/rohess/ws-training

Installation

- Recent version of Wireshark, at least 4.0 – current is 4.4.2
- On MacOS and Windows – just download the installer and install
  → https://www.wireshark.org/download.html
- Linux – check installation steps on Github link
  Repo versions tend to be outdated

The capture file

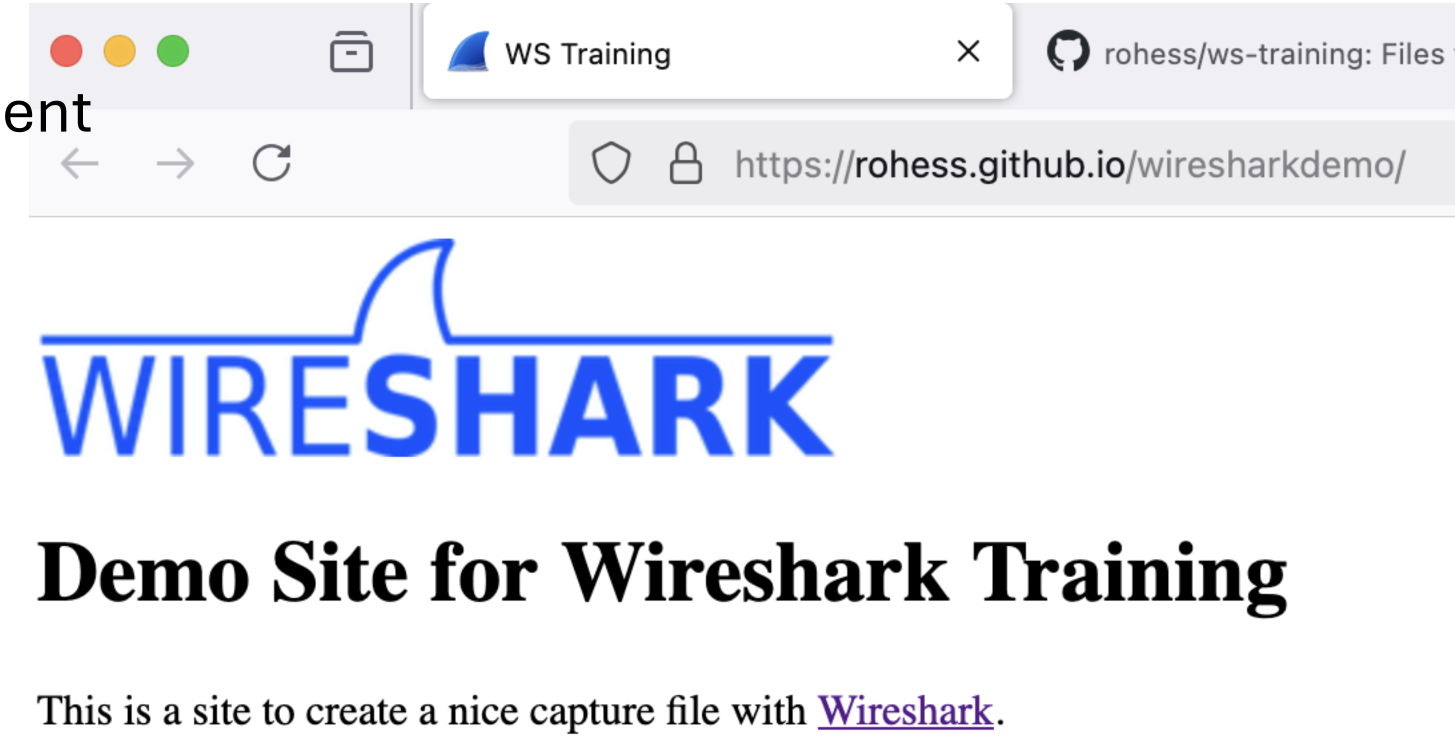
- .pcap format – industry standard for all kind of packet capture software (tcpdump, dumbcap etc.)
- .pcapng format – adds meta data to pcap (file notes, packet notes, custom DNS names)
- contains interesting traffic & noise → remove the noise

- → open the sample capture file from https://github.com/rohess/ws-training

Whats in it:

Download of a single webpage via Chrome

- DNS requests
- TCP Connection establishment
- TLS negotiation
- HTTP traffic downloading
  - HTML Code
  - Picture
  - favicon
- Connection tear down

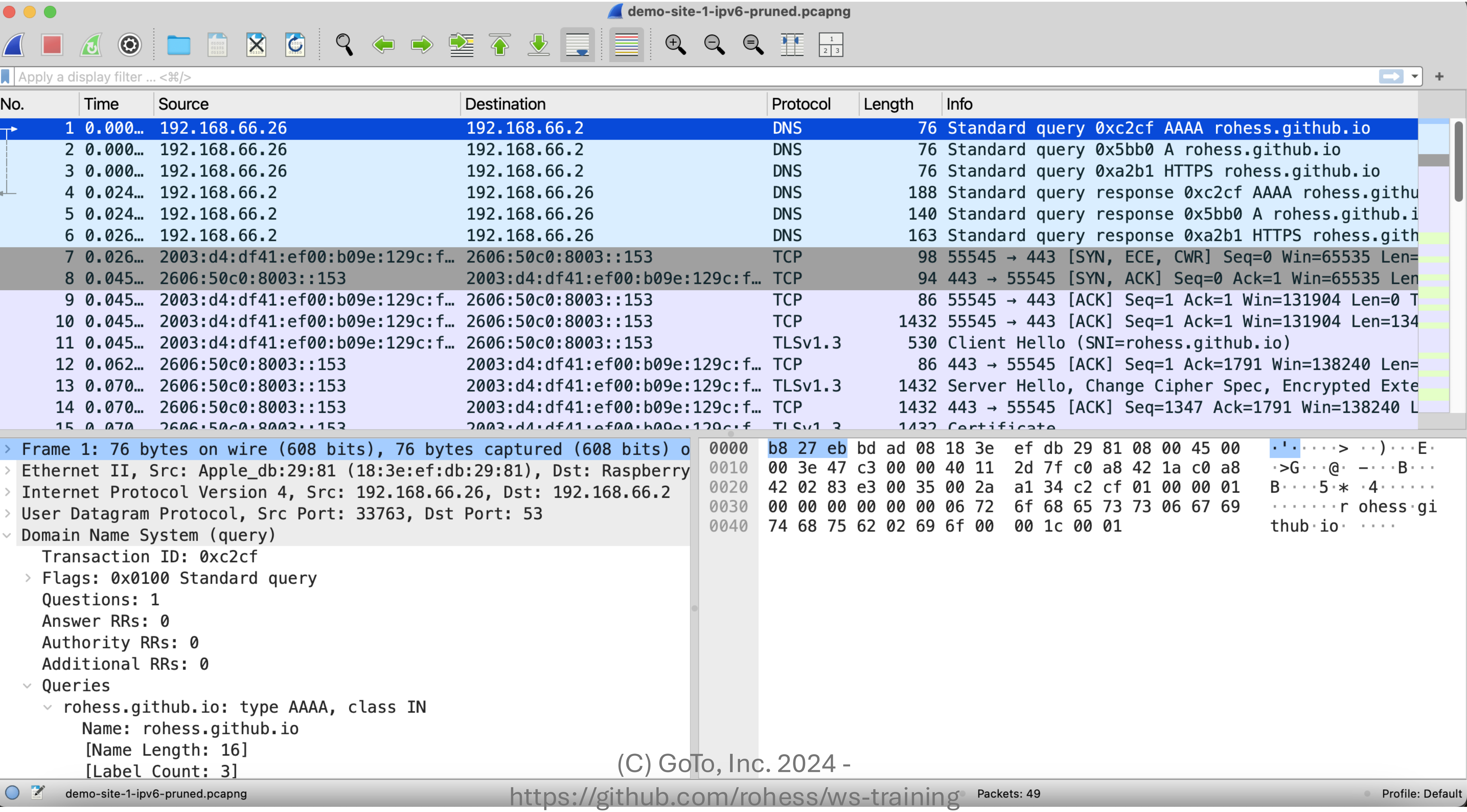

WS Training
rohess/ws-training: Files
https://rohess.github.io/wiresharkdemo/
WIRESHARK

Demo Site for Wireshark Training

This is a site to create a nice capture file with Wireshark.

The Wireshark UI

- Tool bar
  - Capture section
- Initial layout – 3 panes
  - Packet list
  - Packet Details
  - Packet Bytes → much cooler Packet Diagram
- Filter bar
  - Filter string
  - Filter buttons
- Packet list
  - Columns – configurable



```
No.  Time      Source                           Destination                        Protocol  Length  Info
  1  0.000…   192.168.66.26                    192.168.66.2                       DNS           76  Standard query 0xc2cf AAAA rohess.github.io
  2  0.000…   192.168.66.26                    192.168.66.2                       DNS           76  Standard query 0x5bb0 A rohess.github.io
  3  0.000…   192.168.66.26                    192.168.66.2                       DNS           76  Standard query 0xa2b1 HTTPS rohess.github.io
  4  0.024…   192.168.66.2                     192.168.66.26                      DNS          188  Standard query response 0xc2cf AAAA rohess.githu
  5  0.024…   192.168.66.2                     192.168.66.26                      DNS          140  Standard query response 0x5bb0 A rohess.github.i
  6  0.026…   192.168.66.2                     192.168.66.26                      DNS          163  Standard query response 0xa2b1 HTTPS rohess.gith
  7  0.026…   2003:d4:df41:ef00:b09e:129c:f…   2606:50c0:8003::153                TCP           98  55545 → 443 [SYN, ECE, CWR] Seq=0 Win=65535 Len=
  8  0.045…   2606:50c0:8003::153              2003:d4:df41:ef00:b09e:129c:f…     TCP           94  443 → 55545 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len
  9  0.045…   2003:d4:df41:ef00:b09e:129c:f…   2606:50c0:8003::153                TCP           86  55545 → 443 [ACK] Seq=1 Ack=1 Win=131904 Len=0 T
 10  0.045…   2003:d4:df41:ef00:b09e:129c:f…   2606:50c0:8003::153                TCP         1432  55545 → 443 [ACK] Seq=1 Ack=1 Win=131904 Len=134
 11  0.045…   2003:d4:df41:ef00:b09e:129c:f…   2606:50c0:8003::153                TLSv1.3      530  Client Hello (SNI=rohess.github.io)
 12  0.062…   2606:50c0:8003::153              2003:d4:df41:ef00:b09e:129c:f…     TCP           86  443 → 55545 [ACK] Seq=1 Ack=1791 Win=138240 Len=
 13  0.070…   2606:50c0:8003::153              2003:d4:df41:ef00:b09e:129c:f…     TLSv1.3     1432  Server Hello, Change Cipher Spec, Encrypted Exte
 14  0.070…   2606:50c0:8003::153              2003:d4:df41:ef00:b09e:129c:f…     TCP         1432  443 → 55545 [ACK] Seq=1347 Ack=1791 Win=138240 L
 15  0.070…   2606:50c0:8003::153              2003:d4:df41:ef00:b09e:129c:f…     TLSv1.3     1432  Certificate
```

```
> Frame 1: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) o
> Ethernet II, Src: Apple_db:29:81 (18:3e:ef:db:29:81), Dst: Raspberry
> Internet Protocol Version 4, Src: 192.168.66.26, Dst: 192.168.66.2
> User Datagram Protocol, Src Port: 33763, Dst Port: 53
v Domain Name System (query)
    Transaction ID: 0xc2cf
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    v rohess.github.io: type AAAA, class IN
        Name: rohess.github.io
        [Name Length: 16]
        [Label Count: 3]
```

```
0000  b8 27 eb bd ad 08 18 3e  ef db 29 81 08 00 45 00   ·'·····> ··)···E·
0010  00 3e 47 c3 00 00 40 11  2d 7f c0 a8 42 1a c0 a8   ·>G···@· -···B···
0020  42 02 83 e3 00 35 00 2a  a1 34 c2 cf 01 00 00 01   B····5·* ·4······
0030  00 00 00 00 00 00 06 72  6f 68 65 73 73 06 67 69   ·······r ohess·gi
0040  74 68 75 62 02 69 6f 00  00 1c 00 01               thub·io· ····
```

Packet list

- Time – highly configurable via *View/Time Display Format*
- Source – IP or DNS
- Destination – IP or DNS
- Protocol – TCP/UDP/DNS …
- Length – Packet length on wire
- Info – lots of useful stuff that Wireshark found out via its dissectors

Tools – column width, stop scroll, colorize on/off

How does it work

- Wireshark reads pcap/pcapng file and builds the packet tree.
- Each packet is run through applicable list of dissectors and classified accordingly. Results are shown in the various windows
- If keys are available encrypted payloads are decrypted
- When filters are applied the packet tree is rescanned
  - This is single threaded and takes time
  - Keep capture files small (10-100Mbytes) – up to 1 GB works
  - Don’t click while Wireshark rescans in I/O Graph– there are race conditions

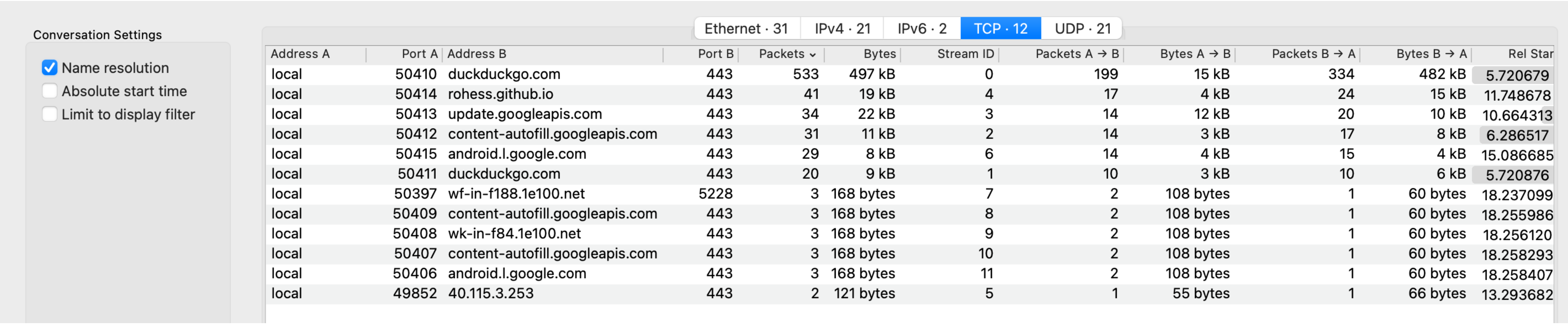
A better packet list

- Switch time display format to time of the day in UTC Time of the Day
- Add columns for Source and Destination Port
- Use Payload length instead of length
- Show Stream index

Investigation – prep steps

1. Reduce size of capture file by removing unnecessary data
2. Pick out the relevant packets based on characteristics (target systems, ports, number of packets)
3. Create the right views to inspect these packets
4. Form a hypothesis in your head how the packet streams should look like
5. Look at the packets and find the differences and google for explanations

Statistics/Conversations



Conversation Settings

☑ Name resolution
□ Absolute start time
□ Limit to display filter

Ethernet · 31 | IPv4 · 21 | IPv6 · 2 | TCP · 12 | UDP · 21

Address A	Port A	Address B	Port B	Packets ⌄	Bytes	Stream ID	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Star
local	50410	duckduckgo.com	443	533	497 kB	0	199	15 kB	334	482 kB	5.720679
local	50414	rohess.github.io	443	41	19 kB	4	17	4 kB	24	15 kB	11.748678
local	50413	update.googleapis.com	443	34	22 kB	3	14	12 kB	20	10 kB	10.664313
local	50412	content-autofill.googleapis.com	443	31	11 kB	2	14	3 kB	17	8 kB	6.286517
local	50415	android.l.google.com	443	29	8 kB	6	14	4 kB	15	4 kB	15.086685
local	50411	duckduckgo.com	443	20	9 kB	1	10	3 kB	10	6 kB	5.720876
local	50397	wf-in-f188.1e100.net	5228	3	168 bytes	7	2	108 bytes	1	60 bytes	18.237099
local	50409	content-autofill.googleapis.com	443	3	168 bytes	8	2	108 bytes	1	60 bytes	18.255986
local	50408	wk-in-f84.1e100.net	443	3	168 bytes	9	2	108 bytes	1	60 bytes	18.256120
local	50407	content-autofill.googleapis.com	443	3	168 bytes	10	2	108 bytes	1	60 bytes	18.258293
local	50406	android.l.google.com	443	3	168 bytes	11	2	108 bytes	1	60 bytes	18.258407
local	49852	40.115.3.253	443	2	121 bytes	5	1	55 bytes	1	66 bytes	13.293682

Investigation

- Open Statistics/Conversations
- Check TCP & UDP
- Select TCP – look for the stream with the most packets
- Follow TCP Stream – show encrypted stream
- Follow TLS Stream – work only if you have the encryption keys
- As it’s a http2 stream, you will have sub streams within the connection
- You can check the decrypted content, and you can also download the objects of the web page

Filtering

- Manual or automatically created from your packets
- There are capture filters and display filters
- Filter for protocols:

```
tcp, udp, dns, http
```

- Filter for numerical values:

```
tcp.stream in {5..8}
ip.addr eq 192.168.1.1
udp.dstport == 53
```

- Filter for string:

```
ip.host matches "github.io”
tls.handshake.extensions_server_name == "rohess.github.io”
```

- **&&, ||, normal parenthesis rules**

More to see

- Check timing - Set Time Reference
- Check GeoIP – only for IPv4
- I/O Graph
- DNS – round robin, HTTPS entries
- Filter buttons

Capturing

- On your system
- Start browser via *Tools/TLS Keylog launcher* (make sure its not already running)
- Capture on the right interface (or just all)
- Disable promiscuous mode
- Clear browser cache, and if DNS is relevant also DNS Cache of the browser (chrome://net-internals/#dns)
- Afterwards Edit/Inject TLS Keys

Advanced stuff

- Build columns with rules in it
- Add GeoIP resolution
- I/O Graphs view
- Get capture files via Command Line or via Mirror port capture
- Get Captures from Mobile device
  - On iOS use a Mac connected via USB
  - On Android use an app like Pcapdroid - supports decryption

Legal ramifications

- Its not always legal in Germany:

  Vorbereiten des Ausspähens und Abfangens von Daten" (§202c des deutschen StGB) aus dem Jahr 2007

  Wikipedia: https://de.wikipedia.org/wiki/Vorbereiten_des_Aussp%C3%A4hens_und_Abfangens_von_Daten
- Do it in your own network to learn and analyse
- If you capture outside: written customer consent / data is PII

Have fun & thanks