New(ish) features in Wireshark

... and a foundation update!

Sake Blok
Relational therapist for computer systems
sake.blok@SYN-bit.nl

$ whoami

EURONET INTERNET

ABN·AMRO

ion ip

SYN-bit
deep traffic analysis

SYN-bit

deep traffic analysis

Application and network troubleshooting

Protocol and packet analysis

Training (Wireshark, TCP, SSL)

www.SYN-bit.nl

$ a bit about you

https://menti.com
6353 7541

Wireshark turns 25!

Wireshark Foundation

- Company owned trademarks
  - The "bus" problem (too much dependency on Gerald)
  - Company could steer Wireshark in a direction that the community does not want
  - Difficult/impossible to open up to donations
  - No more name changes!!!
- Long journey
  - Explored umbrella organisations (2015 onwards)
  - We were heading towards joining one (2019-2021)
  - Back to setting up a non-profit (2021)
  - Temporarily to sysdig as Gerald started working (again) with Loris Degioanni (2022)
  - **Wireshark Foundation non-profit organisation (2023)**

It's official!

Wireshark Foundation @WiresharkNews · Mar 2

It's official! We're now the Wireshark Foundation, a 501(c)(3) nonprofit that will host SharkFest, host Wireshark, and promote analysis and troubleshooting education.

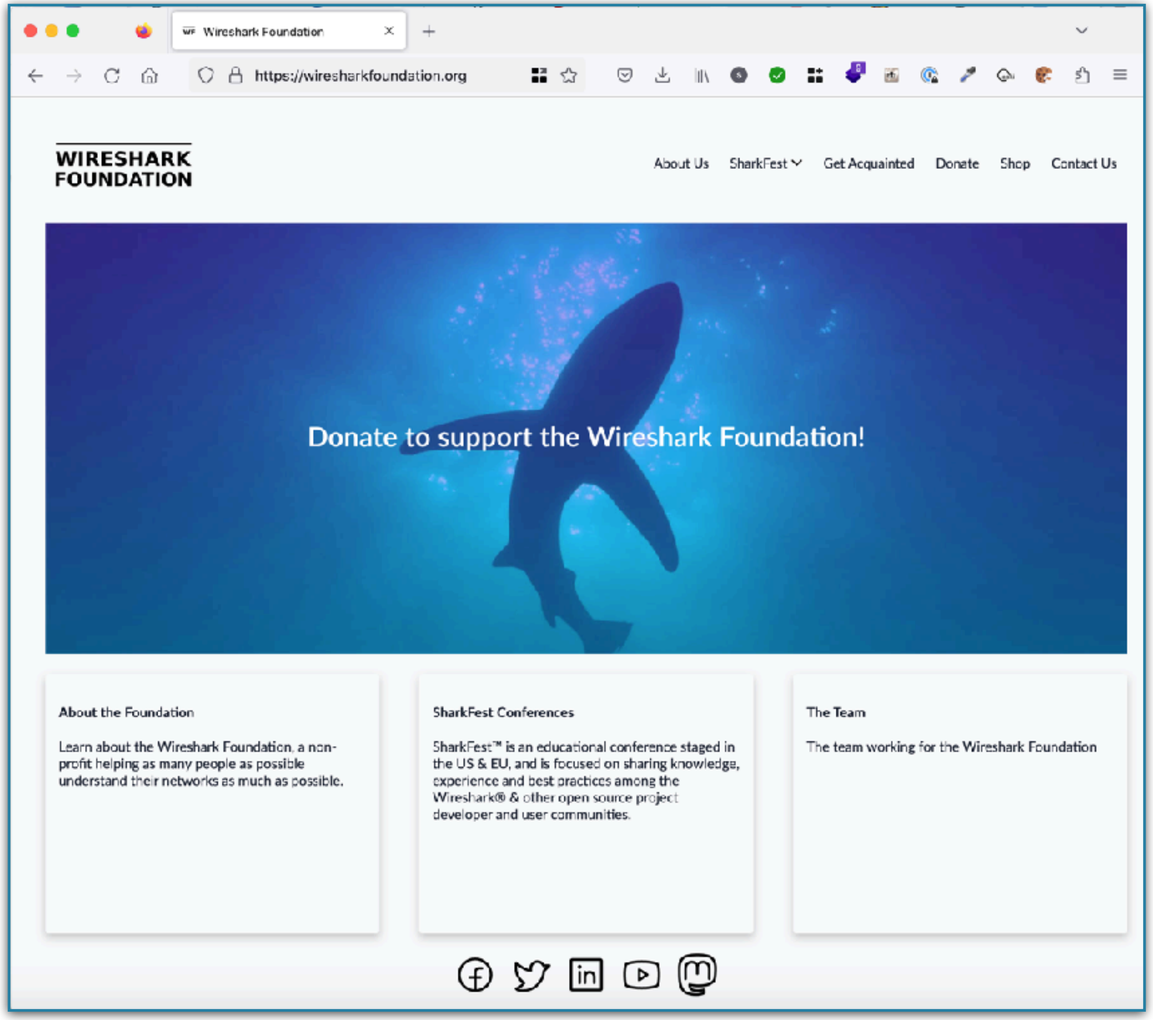
You can find out more at wiresharkfoundation.org.

3 25 97 5,224

Wireshark Foundation
1,870 followers
2d •

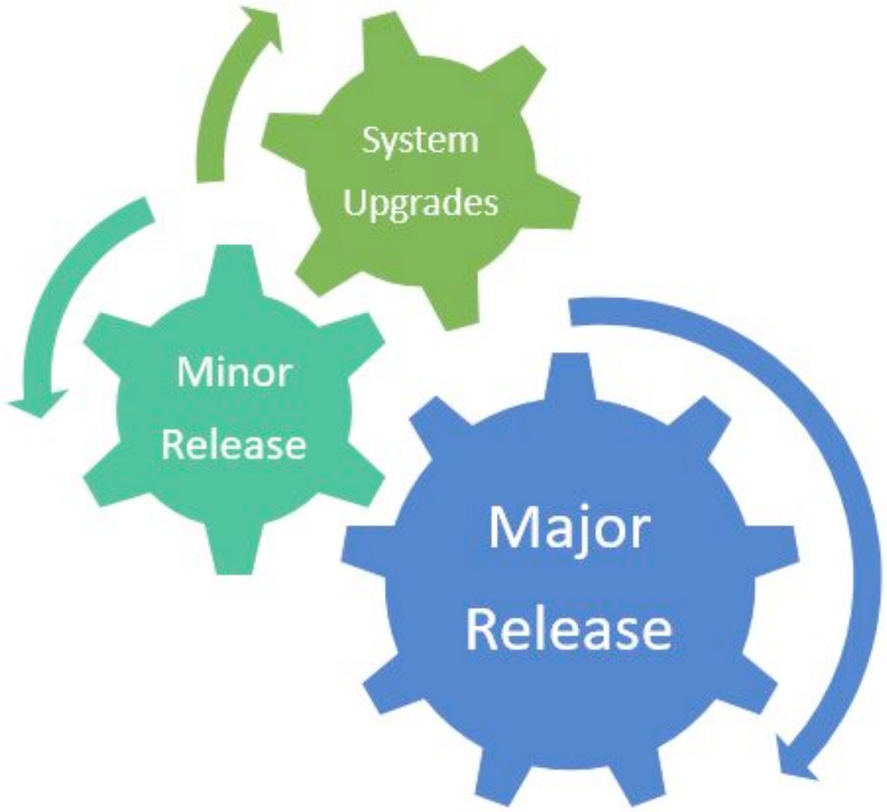
We're beyond thrilled to officially announce something that we've been working on for a very long time. The Wireshark community now has a permanent home: the Wireshark Foundation. The foundation is a 501(c)(3) nonprofit and will host SharkFest, our developer and user conference, help to facilitate Wireshark's development, and promote analysis and troubleshooting education. You can find out more at https://lnkd.in/gg-kPfZr.

Wireshark Foundation
wiresharkfoundation.org



Wireshark major releases

- v1.0.0 (March 31, 2008)
  - After numerous (10!) years of development
  - Released during first Sharkfest
- v2.0.0 (November 18, 2015)
  - Move to Qt graphical library
- v3.0.0 (February 28, 2019)
  - GTK+ GUI toolkit was dropped
  - Npcap as Windows capture library was introduced
- v4.0.0 (October 4, 2022)
  - support to Qt6 as GUI toolkit
  - significant rework of the display filter engine
  - overhaul of endpoint/conversation dialogs



Official Wireshark releases

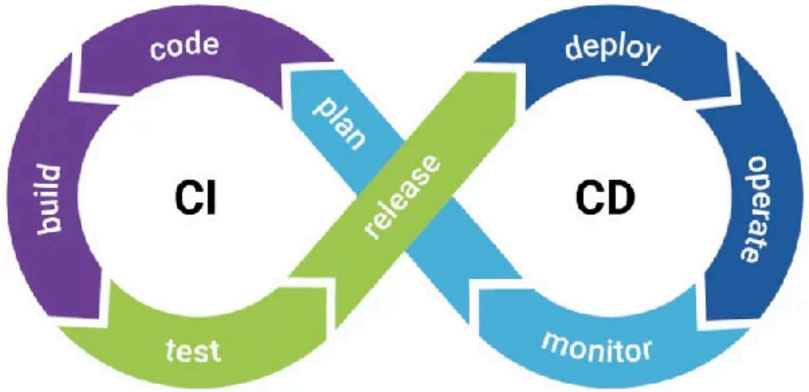
- Uses the minor release number (2.6, 3.0, 3.2, 3.4, 3.6, 4.0, etc)
- Support for 18+ months
  - some releases have long-term-support
- Even numbers for stable
- Odd numbers for development

Maintenance releases

- Use the maintenance release number
  - 4.0.1, 4.0.2, 4.0.3, 4.0.4, etc
- No new features
- only bug fixes and security fixes

Release candidates and automated builds

- Release candidates
  - Released before official releases to test new features
  - We need people to test (see issues with 4.0.0 release)



- Automated builds
  - Created after each submit (CI/CD)
  - Mostly stable, but can introduce issues
  - *Wireshark-win64-3.3.0rc0-1595-g584569932b06.exe* means the following:
    - This is a development release (3 is odd)
    - There have been 1595 commits since the last minor release (3.2.0) was branched off
    - This build was triggered by commit 584569932b06

"Supported" Wireshark versions

Version	Stable Release Date	End of Support	Notes
2.6	*April 18, 2018*	*October 18, 2020*	*Long term support (LTS).* *Last release to support GTK+ and Qt4.* *Last release to support Red Hat EL 6.* *Last release to support Mac OS X 10.6 - 10.11*
3.0	*February 28, 2019*	*October 29, 2020*	
3.2	*December 18, 2019*	*November 22, 2021*	*Last release to support Windows 7 and Windows Server 2008 R2*
3.4	*October 29, 2020*	*September 28, 2022*	
3.6	**November 22, 2021**	**May 22, 2024**	**Long term support (LTS).** **Last release to support 32-bit Windows.** **Last release to support Red Hat EL 7.** **Last release to support SUSE Linux ES 12.**
4.0	**September 28, 2022**	**18+ months**	**Support ends when 4.4.0 is released (Q4 2024 ?)**
4.2	*Q4, 2023*	*18+ months*	*Support ends when 4.6.0 is released (Q4 2025 ?)*

New features in v3.0 (28/2/2019)

- Npcap instead of winpcap
- IP map is back
- UDP stream timestamps
- Copy items between profiles
- Protocol name changes
  - bootp -> dhcp
  - ssl -> tls
- Embedding TLS keys in pcapng format
- string() function added to display filter engine

New features in v3.2 (18/12/2019)

- select multiple packets
- follow HTTP/2 and QUIC streams
- mark/unmark with middle mouse button
  - Somehow got lost in 2009
- drag & drop filtering
- drag & drop columns
- import/export profiles from ZIP or directories

New features in v3.4 (29/10/2020)

- iLBC and opus codec support
- packet diagram
- group filter buttons with "//"

New features in v3.6 (22/11/2021)

- A != B now means *all* A != B instead of *any* A != B
  - Text in release notes is not exactly right
- A ~= B means *any* A != B
- Raw strings format introduced (r'xxx')
- List filters should now use a comma
- "a not in {b, c}" introduced
- Apple Silicon version
- New TCP field: tcp.completeness
- Windows ETL files can now be read
- Redesign of RTP player
- Follow SIP Call (follow stream based on Call-ID)
- tshark: --export-tls-session-keys

New features in v4.0 (5/10/2022)

- Many libraries updates
  - (Biggest one: Qt 6.0 migration)
- Complete overhaul endpoints overview
- Complete overhaul conversations overview
- Improvements to the display filter engine
- New default window layout
- PCRE2 instead of GNU regex
- ciscodump now supports IOS, IOS-XE and ASA remote capturing

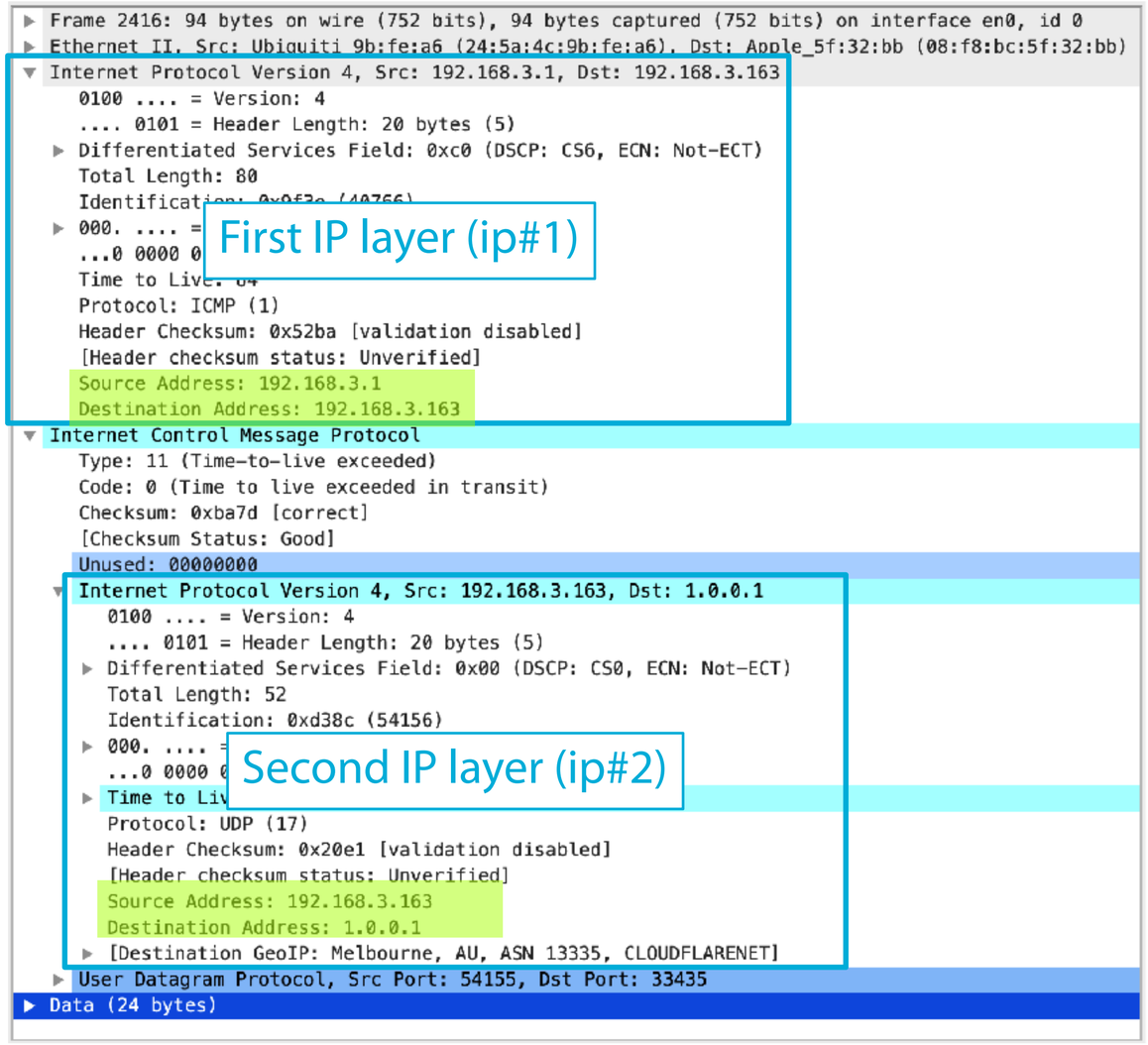
Changes in the display filter engine #1: Syntax changes

- *tcp.port in {80 443 8080..8089}* => tcp.port in {80, 443, 8080..8089}
- *ip.addr ~= 10.0.0.1* => ip.addr !== 10.0.0.1
- Logical AND now has higher precedence than logical OR
- Regular expressions now use PCRE2 instead of GNU Regex
- Every value with a leading dot is a protocol or protocol field
- ~~Every value in between angle brackets is a literal value~~
  - Revoked: https://gitlab.com/wireshark/wireshark/-/merge_requests/8444

Changes in the display filter engine #2: handling of fields with multiple occurences

- ip.addr != 10.0.0.10 now means:
  - all ip.addr fields should not have the value 10.0.0.10
  - Can also be written as: ip.addr all_ne 10.0.0.10
- ip.addr !== 10.0.0.10 (works the same as the old "!=") means
  - any ip.addr fields should not have the value 10.0.0.10
  - Can also be written as ip.addr any_ne 10.0.0.10
  - "~=", introduced in v3.6 is now deprecated
- ip.addr any_eq 10.0.0.10 is the same as ip.addr == 10.0.0.10
- ip.addr all_eq 10.0.0.10 is the same as ip.addr === 10.0.0.10 (new)
- New filter elements: "all" and "any" (e.g. "all tcp.port >= 1024")

Changes in the display filter engine #3: protocol layers



Field	192.168.3.1	192.168.3.163	1.0.0.1
ip.addr	✔	✔	✔
ip.addr#1	✔	✔	
ip.addr#2		✔	✔
ip.src	✔	✔	
ip.src#1	✔		
ip.src#2		✔	
ip.dst		✔	✔
ip.dst#1		✔	
ip.dst#2			✔

Changes in the display filter engine #4: field references: ${<field>}

- Used to be implemented as a macro
- Could not handle multiple occurences of a field
  - tcp.port == ${tcp.port} was not a valid filter
- Could not handle missing fields
  - tcp.stream == ${tcp.stream} was an invalid filter on a non-tcp packet
- In v4.0 part of display filter syntax
  - "udp.port==${udp.port} || tcp.port==${tcp.port}" now works!
- Especially useful in filter buttons!

Changes in the display filter engine #5: changes with slices

- Now possible to test the existence of a slice
  - frame[100:10] will match if the frame has at least a size of 110 octets
- Negative indices can be used to slice from the end of a field
  - tcp[-4:] == 0d:0a:0d:0a
  - udp[-10--6] == "dummy"

Changes in the display filter engine #6: arithmetics

- The "bitwise and" operator is now a first-class bit operator, not a boolean operator
  - tcp.flags & 18 == 2 is now possible
- Math with + - * / % operators possible
  - Arithmetic expressions must be grouped using curly brackets (not parenthesis)
  - {tcp.port + 1} % 1000 == 0 (matches tcp ports 999, 1999, 2999, etc)
- Functions max(), min(), abs() are added
  - tcp.port == min(${tcp.srcport}, ${tcp.dstport})

Changes in the display filter engine #7: value notations (integers, floats, booleans)

- Integers
  - Prefixes 0b and 0B can be used for a binary value
  - ip.ttl == 0b10000000 (same as: ip.ttl == 128, ip.ttl == 0x80, ip.ttl == 0200)
  - All integer sizes are now compatible
    - Unless overflow occurs any integer field can be compared with any other
- Floats
  - must be written with a leading and ending digit
  - .7 now needs to be written as 0.7 and 7. now needs to be written as 7.0
- Booleans
  - can be written as True/TRUE or False/FALSE
  - tcp.flags.syn == True && tcp.flags.ack == False

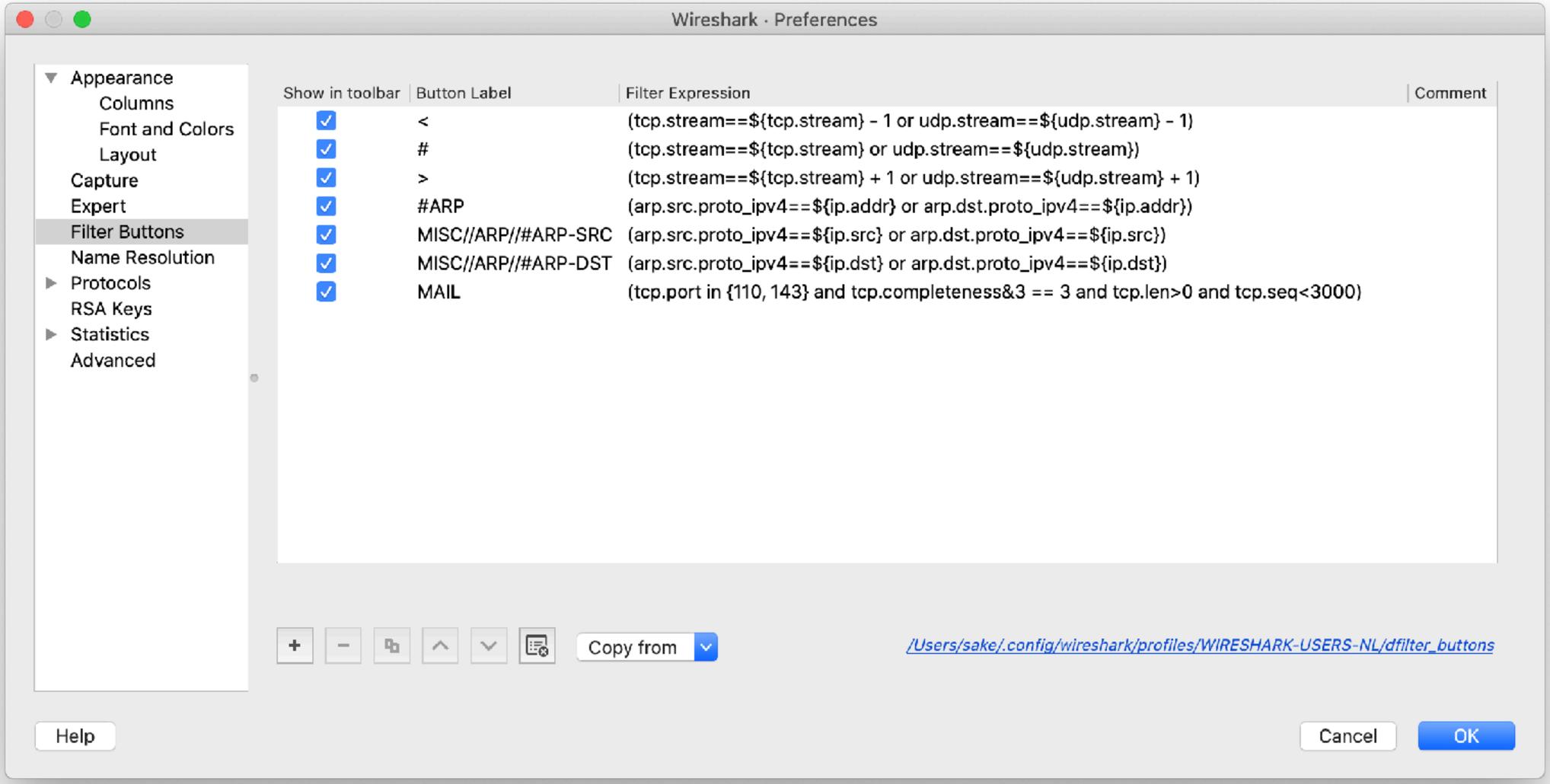
Changes in the display filter engine #8: value notations (Strings, Times)

- String literals
  - \<nnn> (octal), \x<nn> (hex)
  - \a, \b, \f, \n, \r, \t, \v, \0, \\, \', \" (escaped characters)
  - \uNNNN, \UNNNNNNNN (unicode characters)

- Dates and times
  - can be given in UTC using ISO 8601 (with 'Z' timezone) or by appending the suffix "UTC" to the legacy formats. Otherwise local time is used.
  - frame.time < "2023-02-22T16:26:19Z"
  - frame.time >= "Feb 22, 2023 16:26:18.060734000 UTC"


LIVE DEMO

Filter buttons used in the demo




Q&A
You have
Questions
We have
Answers

FIN/ACK/FIN/ACK

Still questions?

sake.blok@SYN-bit.nl