

Useful Wireshark Filters

☐☐ IP & Netzwerk-Grundlagen (IP & Network Basics)

- `ip.addr == 10.0.0.1`
Zeigt den gesamten Datenverkehr mit 10.0.0.1 als Quelle oder Ziel an.
- `ip.addr == 10.0.0.0/24`
Zeigt den gesamten Datenverkehr von und zu einer beliebigen Adresse im Subnetz 10.0.0.0/24 an.
- `ip.src == 10.0.0.1 && ip.dst == 10.0.0.2`
Zeigt den gesamten Datenverkehr von der Quell-IP 10.0.0.1 zur Ziel-IP 10.0.0.2 an.
- `!(ip.addr == 10.0.0.1)`
Schließt den gesamten Datenverkehr von oder zu der Adresse 10.0.0.1 aus.
- `ip.ttl < 10`
Zeigt Pakete mit einer Time-to-Live (TTL) unter 10 (nützlich zum Aufspüren von Routing-Schleifen).

☐☐ Transportprotokolle & Ports (TCP / UDP)

- `tcp or udp`
Zeigt ausschließlich TCP- oder UDP-Datenverkehr an.
- `tcp.port == 80`
Zeigt TCP-Datenverkehr auf Port 80 (HTTP) an.
- `tcp.srcport < 1000`
Zeigt TCP-Datenverkehr, dessen Quellport kleiner als 1000 ist.
- `(tcp or udp)`
Filtert nach Nicht-TCP/UDP-Datenverkehr (hilft beim Finden ungewöhnlicher Protokolle).

⚡ TCP-Analyse & Flags (TCP Analysis & Flags)

- `tcp.flags.syn == 1`
Zeigt TCP-Pakete, bei denen das SYN-Flag gesetzt ist (Verbindungsaufbau).

- `tcp.flags == 0x012`
Zeigt TCP-Pakete, bei denen sowohl das SYN- als auch das ACK-Flag gesetzt sind.
- `tcp.analysis.retransmission`
Zeigt alle erneut übertragenen TCP-Pakete (Retransmissions).
- `tcp.analysis.lost_segment`
Zeigt TCP-Segmente, die als verloren markiert wurden.
- `tcp.analysis.duplicate_ack`
Zeigt doppelte TCP-ACKs an, was auf Paketverlust hindeuten kann.

☐☐ Web-Protokolle (HTTP & DNS)

- `http or dns`
Zeigt den gesamten HTTP- oder DNS-Datenverkehr an.
- `http.request.method == "GET"`
Zeigt HTTP-Pakete, die mit einer HTTP-GET-Anfrage verknüpft sind.
- `http.response.code == 404`
Zeigt Pakete, die mit einem HTTP 404 (Not Found) Statuscode antworten.
- `http.host == "www.test.com"`
Zeigt HTTP-Datenverkehr, der mit dem spezifischen Host-Header übereinstimmt.
- `dns.qry.name contains "cnn.com"`
Zeigt DNS-Anfragepakete (Queries), die "cnn.com" enthalten.
- `dns.resp.name contains "cnn.com"`
Zeigt DNS-Antwortpakete (Responses), die "cnn.com" enthalten.
- `dns.qry.name matches "\.xyz$|\|.club$"`
Zeigt DNS-Anfragen für Domains, die auf die TLDs `.xyz`` oder `.club`` enden.

☐☐ Verschlüsselung (TLS)

- `tls.handshake`
Zeigt ausschließlich TLS-Handshake-Pakete an.
- `tls.handshake.type == 1`
Zeigt das "Client Hello"-Paket während des TLS-Handshakes.

☐☐ Infrastruktur & Layer 2 (DHCP, Ethernet, VLAN)

- `icmp.type == 3`
Zeigt ICMP-Pakete vom Typ "Destination Unreachable" (Ziel unerreichbar).
- `dhcp and ip.addr == 10.0.0.0/24`
Zeigt DHCP-Datenverkehr für das Subnetz 10.0.0.0/24.

- `dhcp.hw.mac_addr == 00:11:22:33:44:55`
Zeigt DHCP-Pakete für eine bestimmte Client-MAC-Adresse.
- `eth.addr == 00:11:22:33:44:55`
Zeigt den gesamten Datenverkehr von oder zu der angegebenen MAC-Adresse.
- `eth[0x47:2] == 01:80`
Filtert nach Ethernet-Frames, bei denen 2 Bytes am Offset 0x47 dem Wert 01:80 entsprechen.
- `!(arp or icmp or stp)`
Filtert Hintergrund-Rauschen von ARP, ICMP und STP (Spanning Tree Protocol) heraus.
- `vlan.id == 100`
Zeigt Pakete, die mit der VLAN-ID 100 getaggt sind.

☐☐ Rahmen- & Textsuche (Frame & Payload)

- `frame contains "keyword"`
Zeigt alle Pakete an, die das spezifische Wort "keyword" im Payload/Rahmen enthalten.
- `frame.len > 1000`
Zeigt alle Pakete an, deren Gesamtlänge größer als 1000 Bytes ist.

Revision #3

Created 2026-06-02 11:19:59 UTC by Oliver Schoenemann

Updated 2026-06-03 07:12:36 UTC by Oliver Schoenemann