



26

Wireshark FILTERS YOU SHOULD KNOW

1. Show Bidirectional traffic between two hosts.

```
(ip.src == 192.168.0.11 && ip.dst == 192.168.0.102) || (ip.src == 192.168.0.102 && ip.dst == 192.168.0.11)
```

Capture and analyze all communication between two specific hosts for troubleshooting, pattern analysis, or security investigation.

2. Show TCP SYN packets.

```
tcp.flags.syn == 1 && tcp.flags.ack == 0
```

Detect new connection attempts, port scans or initial handshakes that may indicate reconnaissance or brute force attacks.

3. Show TCP FIN packets.

```
tcp.flags.fin == 1 && tcp.flags.ack == 1
```

Monitor TCP connection terminations and detect anomalies like improper closures or hijacking attempts.

SWIPE





4. Show all TCP retransmissions.

`tcp.analysis.retransmission`

Identify network congestion or packet loss by highlighting TCP retransmissions.

5. Show HTTP POST requests.

`http.request.method == "POST"`

Monitor POST requests to track data submissions, login attempts, API calls, or potential data exfiltration.

6. Display HTTP traffic where the word "password" is visible.

`http contains "password"`

Detect potential credential, sensitive data leakage in unencrypted HTTP traffic.

7. FTP Login Attempts with Passwords Sent in Plaintext.

`ftp.request.command == "PASS"`

Monitor FTP authentication and detect plaintext password exposure or brute-force attacks.

SWIPE





8. Show all Telnet sessions, a protocol known for lack of encryption.

`telnet`

Monitor and identify unencrypted Telnet sessions to detect security risks or policy violations.

9. TLS Client Hello packets

`tls.handshake.type == 1`

Identify and analyze TLS handshake behavior, client application traits, and encryption usage for fingerprinting or network troubleshooting.

10. TLS 1.0 Traffic (Deprecated Protocol Version)

`ssl.record.version == 0x0301`

Identify systems still using deprecated TLS 1.0 to mitigate security risks and ensure compliance with modern encryption standards.

11. PowerShell Detected in Packet Contents.

`frame contains "powershell"`

Detect and monitor unauthorized or malicious PowerShell activity in network traffic. Useful for threat hunting.

SWIPE





12. Show traffic over common reverse shell ports.

```
tcp.port == 4444 || tcp.port == 1337
```

Detect potential reverse shell activity by monitoring network traffic on commonly used ports by attackers.

13. Show all traffic to or from a specific IP address.

```
ip.addr == 192.168.1.10
```

Monitor all communications involving a specific host for troubleshooting, security analysis, or sniffing for authentication in plain text.

14. Show SNMP traffic.

```
udp.port == 161
```

Detect SNMP-based reconnaissance or misconfigurations that could expose network information.

15. Capture HTTP responses.

```
http.set_cookie
```

Capture and analyze cookies from HTTP responses to assess session management and detect potential session hijacking.

SWIPE





16. Show requests where the User-Agent header includes "curl".

`http.user_agent contains "curl"`

Detect automated scripts, API testing tools, or potential malicious automation tools often used in attacks or reconnaissance activities.

17. Show TCP analysis flags.

`tcp.analysis.flags`

Comprehensive TCP troubleshooting by displaying all anomalies including retransmissions, duplicate ACKs, window updates, and other connection issues.

18. Show frames larger than 1000 bytes.

`frame.len > 1000`

Identify large data transfers that could indicate file transfers, data exfiltration, backup operations, or potential performance bottlenecks caused by oversized packets, buffer overflow and DOS attacks.

19. Traffic over Windows Remote Management (WinRM).

`tcp.port == 5985 || tcp.port == 5986`

Monitor remote Windows administration activities, detect lateral movement attempts in compromised networks, or investigate unauthorized remote access.

SWIPE





20. Packets with Inter-Arrival Time Over 1 Second.

`frame.time_delta > 1`

Detect beaconing behavior often associated with malware command-and-control (C2) communication, or identify abnormal latency and irregular traffic timing.

21. Look for machine accounts in Kerberos.

`kerberos.CNameString contains "$"`

Detect potential Golden Ticket attacks or other Kerberos-based attacks targeting machine accounts, which are often indicators of advanced persistent threats.

22. Large SMB packets

`tcp.port == 445 && frame.len > 1000`

Identify potential file transfers over SMB, detect data exfiltration attempts, or monitor large file operations that could impact network performance.

23. Show NTLM authentication attempts.

`tcp contains "NTLMSSP"`

Monitor Windows authentication activities, detect Pass-the-Hash attacks, investigate authentication failures, or analyze domain security events.



24. Detect PowerShell post-exploitation tool usage.

`tcp` contains "Invoke-Mimikatz"

Detect specific post-exploitation tools like Mimikatz being used to harvest credentials, indicating a potential security breach requiring immediate response.

25. Detect obfuscated command and control via shorteners.

`dns.qry.name` contains "bit.ly" || `dns.qry.name` contains "tinyurl"

Identify potential command and control communications using URL shortening services to obfuscate malicious domains, a common evasion technique or internal phishing attacks.

26. Detect TLS traffic to file-sharing services.

`tls.handshake.extensions_server_name` contains "dropbox.com"

Monitor data exfiltration attempts through cloud storage services, enforce data loss prevention policies, or investigate unauthorized file uploads.

BONUS - WIRESHARK FEATURES

The Telephony Feature - VoIP Analysis Menu

Wireshark includes a dedicated Telephony menu that provides in-depth analysis of all VoIP traffic, including SIP, RTP, and call flow diagrams. This feature is invaluable for troubleshooting voice communications, reconstructing calls, and analyzing voice protocols often used in corporate environments.

Purpose: Ideal for identifying insecure VoIP setups, performing call flow analysis, and detecting VoIP-based attacks or misconfigurations.

Wireless Feature – WLAN and Access Point Analysis

Wireshark has a dedicated Wireless menu that displays WLAN traffic, nearby access points, signal strength, and client association/disassociation events. It supports detailed analysis of 802.11 frames, including beacons, probes, authentication, and handshakes.

Purpose: Perfect for wireless reconnaissance, identifying rogue access points, troubleshooting connectivity issues, and monitoring WPA/WPA2 handshakes during red team operations.

