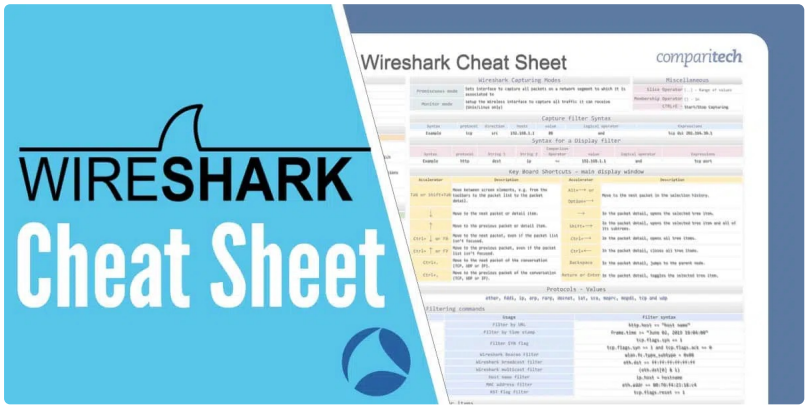
Wireshark Cheat Sheet – Commands, Captures, Filters & Shortcuts

Wireshark is an essential tool for network administrators, but very few of them get to unleash its full potential. Having all the commands and useful features in one place is bound to boost productivity. So we put together a power-packed Wireshark Cheat Sheet. You can download it for free as a PDF or JPG.

TIM KEARY - NETWORK SECURITY AND ADMINISTRATION EXPERT
Updated: December 29, 2023



All the information that has been provided in the cheat sheet is also visible further down this page in a format that is easy to copy and paste.

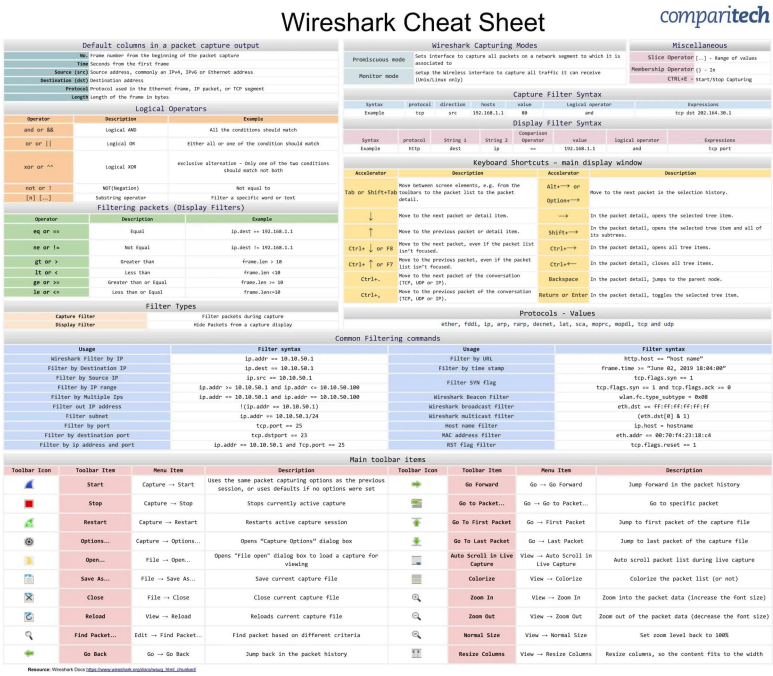
WHAT'S IN THIS ARTICLE?

The cheat sheet covers:

- Wireshark Capturing Modes
- Filter Types
- Capture Filter Syntax
- Display Filter Syntax
- Protocols – Values
- Filtering packets (Display Filters)
- Logical Operators
- Default columns in a packet capture output
- Miscellaneous Items
- Keyboard Shortcuts
- Common Filtering Commands
- Main Toolbar Items

View or Download the Cheat Sheet JPG image

Right-click on the image below to save the JPG file (2500 width x 2096 height in pixels), or click here to open it in a new browser tab. Once the image opens in a new window, you may need to click on the image to zoom in and view the full-sized jpeg.



View or Download the cheat sheet JPG image

Click on the link to download the Cheat Sheet PDF. If it opens in a new browser tab, simply right click on the PDF and navigate to the download selection.

What's included in the Wireshark cheat sheet?

The following categories and items have been included in the cheat sheet:

Wireshark Capturing Modes

Wireshark Capturing Modes	
Promiscuous mode	Sets interface to capture all packets on a network segment to which it is associated to
Monitor mode	setup the Wireless interface to capture all traffic it can receive (Unix/Linux only)

Filter Types

Filter Types	
Capture filter	Filter packets during capture
Display Filter	Hide Packets from a capture display

Capture Filter Syntax

Capture filter Syntax						
Syntax	**protocol**	**direction**	**hosts**	**value**	**Logical operator**	**Expressions**
Example	tcp	src	192.168.1.1	80	and	tcp dst 202.164.30.1

Display Filter Syntax

Display Filter Syntax							
Syntax	**protocol**	**String 1**	**String 2**	**Comparison Operator**	**value**	**logical operator**	**Expressions**
Example	http	dest	ip	==	192.168.1.1	and	tcp port

Protocols – Values

Protocols - Values
ether, fddi, ip, arp, rarp, decnet, lat, sca, moprc, mopdl, tcp and udp

Filtering packets (Display Filters)

Filtering packets (Display Filters)		
Operator	**Description**	**Example**
eq or ==	Equal	ip.dest == 192.168.1.1
ne or !=	Not Equal	ip.dest != 192.168.1.1
gt or >	Greater than	frame.len > 10
lt or <	Less than	frame.len <10
ge or >=	Greater than or Equal	frame.len >= 10
le or <=	Less than or Equal	frame.len<=10

Miscellaneous Items

Miscellaneous	
Slice Operator	[…] - Range of values
Membership Operator	{} - In
CTRL+E -	Start/Stop Capturing

Logical Operators

Logical Operators		
Operator	**Description**	**Example**
and or &&	Logical AND	All the conditions should match
or or ||	Logical OR	Either all or one of the condition should match
xor or ^^	Logical XOR	exclusive alternation – Only one of the two conditions should match not both
not or !	NOT(Negation)	Not equal to

Logical Operators		
[n] […]	Substring operator	Filter a specific word or text

Default columns in a packet capture output

Default columns in a packet capture output	
No.	Frame number from the beginning of the packet capture
Time	Seconds from the first frame
Source (src)	Source address, commonly an IPv4, IPv6 or Ethernet address
Destination (dst)	Destination address
Protocol	Protocol used in the Ethernet frame, IP packet, or TCP segment
Length	Length of the frame in bytes

Keyboard Shortcuts

Keyboard Shortcuts – main display window			
Accelerator	**Description**	**Accelerator**	**Description**
Tab or Shift+Tab	Move between screen elements, e.g. from the toolbars to the packet list to the packet detail.	**Alt+→ or Option+→**	Move to the next packet in the selection history.
↓	Move to the next packet or detail item.	→	In the packet detail, opens the selected tree item.
↑	Move to the previous packet or detail item.	**Shift+→**	In the packet detail, opens the selected tree item and all of its subtrees.
Ctrl+ ↓ or F8	Move to the next packet, even if the packet list isn't focused.	**Ctrl+→**	In the packet detail, opens all tree items.
Ctrl+ ↑ or F7	Move to the previous packet, even if the packet list isn't focused.	**Ctrl+←**	In the packet detail, closes all tree items.
Ctrl+.	Move to the next packet of the conversation (TCP, UDP or IP).	**Backspace**	In the packet detail, jumps to the parent node.
Ctrl+,	Move to the previous packet of the conversation (TCP, UDP or IP).	**Return or Enter**	In the packet detail, toggles the selected tree item.

Common Filtering Commands

Usage	Filter syntax
Wireshark Filter by IP	ip.addr == 10.10.50.1
Filter by Destination IP	ip.dest == 10.10.50.1

Usage	Filter syntax
Filter by Source IP	ip.src == 10.10.50.1
Filter by IP range	ip.addr >= 10.10.50.1 and ip.addr <= 10.10.50.100
Filter by Multiple Ips	ip.addr == 10.10.50.1 and ip.addr == 10.10.50.100
Filter out/ Exclude IP address	!(ip.addr == 10.10.50.1)
Filter IP subnet	ip.addr == 10.10.50.1/24
Filter by multiple specified IP subnets	ip.addr == 10.10.50.1/24 and ip.addr == 10.10.51.1/24
Filter by Protocol	• dns • http • ftp • ssh • arp • telnet • icmp
Filter by port (TCP)	tcp.port == 25
Filter by destination port (TCP)	tcp.dstport == 23
Filter by ip address and port	ip.addr == 10.10.50.1 and Tcp.port == 25
Filter by URL	http.host == “host name”
Filter by time stamp	frame.time >= “June 02, 2019 18:04:00”
Filter SYN flag	tcp.flags.syn == 1 tcp.flags.syn == 1 and tcp.flags.ack == 0
Wireshark Beacon Filter	wlan.fc.type_subtype = 0x08
Wireshark broadcast filter	eth.dst == ff:ff:ff:ff:ff:ff
WiresharkMulticast filter	(eth.dst[0] & 1)
Host name filter	ip.host = hostname
MAC address filter	eth.addr == 00:70:f4:23:18:c4
RST flag filter	tcp.flags.reset == 1

Main Toolbar Items

Main toolbar items			
Toolbar Icon	Toolbar Item	Menu Item	Description
	Start	Capture → Start	Uses the same packet capturing options as the previous session, or uses defaults if no options were set
	Stop	Capture → Stop	Stops currently active capture

Main toolbar items			
	Restart	Capture → Restart	Restarts active capture session
	Options…	Capture → Options…	Opens “Capture Options” dialog box
	Open…	File → Open…	Opens "File open" dialog box to load a capture for viewing
	Save As…	File → Save As…	Save current capture file
	Close	File → Close	Close current capture file
	Reload	View → Reload	Reloads current capture file
	Find Packet…	Edit → Find Packet…	Find packet based on different criteria
	Go Back	Go → Go Back	Jump back in the packet history
	Go Forward	Go → Go Forward	Jump forward in the packet history
	Go to Packet…	Go → Go to Packet…	Go to specific packet
	Go To First Packet	Go → First Packet	Jump to first packet of the capture file
	Go To Last Packet	Go → Last Packet	Jump to last packet of the capture file
	Auto Scroll in Live Capture	View → Auto Scroll in Live Capture	Auto scroll packet list during live capture
	Colorize	View → Colorize	Colorize the packet list (or not)
	Zoom In	View → Zoom In	Zoom into the packet data (increase the font size)
	Zoom Out	View → Zoom Out	Zoom out of the packet data (decrease the font size)
	Normal Size	View → Normal Size	Set zoom level back to 100%
	Resize Columns	View → Resize Columns	Resize columns, so the content fits to the width

Lesser known yet handy Wireshark features

1. **Statistics and analysis tools**: Wireshark comes with a range of built-in tools to analyze

network data. Examples include the ability to visualize statistics, like protocol hierarchy, endpoints, packet lengths, and more. I added this command as an example: `Statistics > Protocol Hierarchy`.

2. **Stream Follow**: The ability to follow a TCP, UDP, or SSL/TLS stream, which can be very useful for understanding what's happening in a particular connection. I added this example: Right-click on a packet and select `Follow > TCP Stream`.
3. **Exporting and Saving Data**: Commands for exporting specific packets, saving packet data, or creating PCAP files can also be useful. I added this example: `File > Export Specified Packets...`.
4. **Capture Interfaces**: Selecting and managing capture interfaces can be helpful. I added this example: `Capture > Options > Manage Interfaces`.
5. **Time Display Format and Precision**: Commands for changing the time display format and precision might also be helpful for some users. I added this example:, `View > Time Display Format > Seconds Since Beginning of Capture`.

More Wireshark tutorials:

- Wireshark cheat sheet
- How to decrypt SSL with Wireshark
- Using Wireshark to get the IP address of an Unknown Host
- Running a remote capture with Wireshark and tcpdump
- Wireshark 'no interfaces found' error explained
- Identify hardware with OUI lookup in Wireshark
- Best Wireshark alternatives

Wireshark FAQs

What are the filters in Wireshark?

Wireshark filters reduce the number of packets displayed in the Wireshark data viewer. This function lets you see the packets that are relevant to your research. There are two types of filters:

- capture filters
- display filters

Applying a filter to the packet capture process reduces the volume of traffic that Wireshark reads

How do I capture a filter in Wireshark?

You can reduce the amount of packets Wireshark copies with a capture filter.

1. Locate the **Capture** section on the Home screen. The first line in this section is labeled **using this filter:**
2. The file that follows this prompt allows you to enter a filter statement.
3. Select an interface to capture from and then click on the shark fin icon on the menu bar to start a capture.

If you don't see the Home page:

1. Click on **Capture** on the menu bar and then select **Options** from that drop-down menu. You will see a list of available interfaces and the capture filter field towards the bottom of the screen.
2. Select an interface by clicking on it, enter the filter text, then click on the **Start** button.

How does Wireshark capture packets?

Wireshark accesses a separate program to collect packets from the wire of the network through the network card of the computer that hosts it. This program is based on the pcap protocol, which is implemented in libpcap for Unix, Linux, and macOS, and by WinPCap on Windows. The installer for Wireshark will also install the necessary pcap program.

2 Comments Leave a comment

Pablo
July 9, 2020 at 12:18 am

thanks for the effort, good thing to have

Reply ▸

Francisco Duque
January 27, 2020 at 7:52 am

Thanks. A great job.

Reply ▸

Leave a Reply

Comment

Name *

Leave Comment

This site uses Akismet to reduce spam. Learn how your comment data is processed.