

WIRESHARK

1)How to Download Wireshark:

Downloading and installing Wireshark is easy. Step one is to check the official <https://www.wireshark.org/download.html> for the operating system you need. The basic version of Wireshark is free.

- Wireshark for windows:

Wireshark comes in two flavors for Windows, 32 bit and 64 bit. Pick the correct version for your OS. The current release is 3.4.1 as of this writing. The installation is simple and shouldn't cause any issues.

- Wireshark for Mac:

Wireshark is available on Mac as a [Homebrew](#) install. To install Homebrew, you need to run this command at your Terminal prompt:
`/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"`. Once you have the Homebrew system in place, you can access several open-source projects for your Mac. To install Wireshark run this command from the Terminal

```
brew install wireshark
```

Homebrew will download and install Wireshark and any dependencies so it will run correctly.

- Wireshark for Linux:

Installing Wireshark on Linux can be a little different depending on the Linux distribution. If you aren't running one of the following distros, please double-check the commands.

From a terminal prompt, run these commands:

1. `sudo apt-get install wireshark`
2. `sudo dpkg-reconfigure wireshark-common`
3. `sudo adduser $USER wireshark`

Those commands download the package, update the package, and add user privileges to run Wireshark.

Red Hat Fedora:

From a terminal prompt, run these commands:

1. `sudo dnf install wireshark-qt`
2. `sudo usermod -a -G wireshark username`

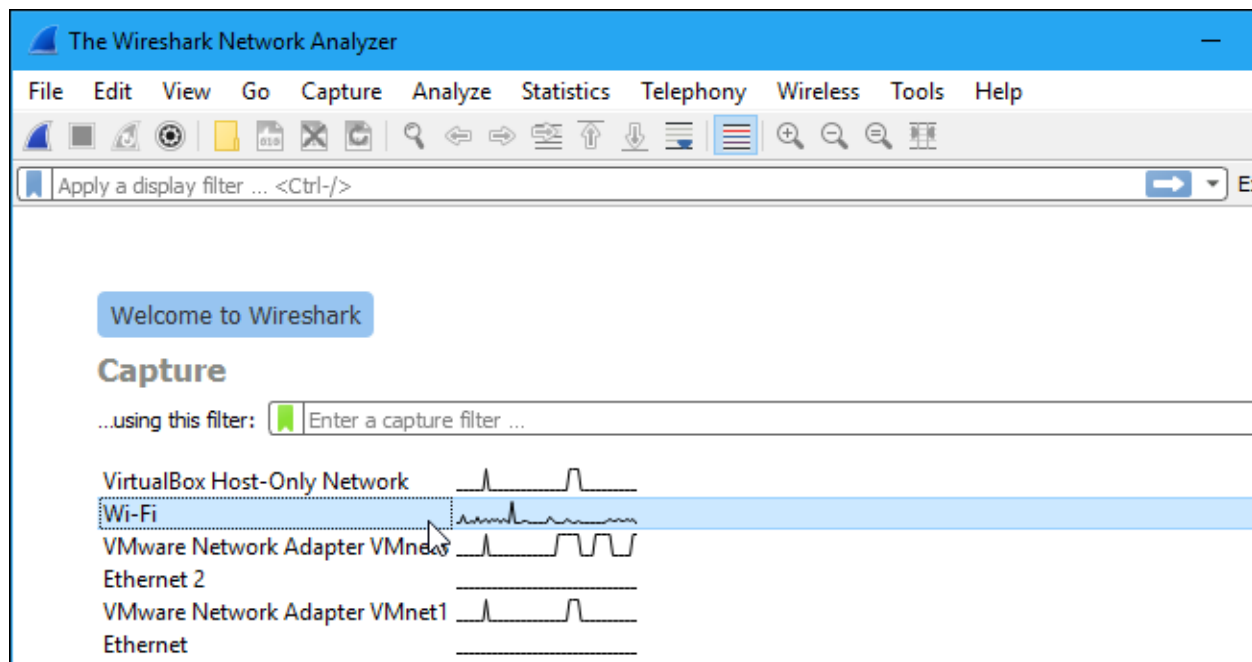
The first command installs the GUI and CLI version of Wireshark, and the second adds permissions to use Wireshark.

Kali Linux:

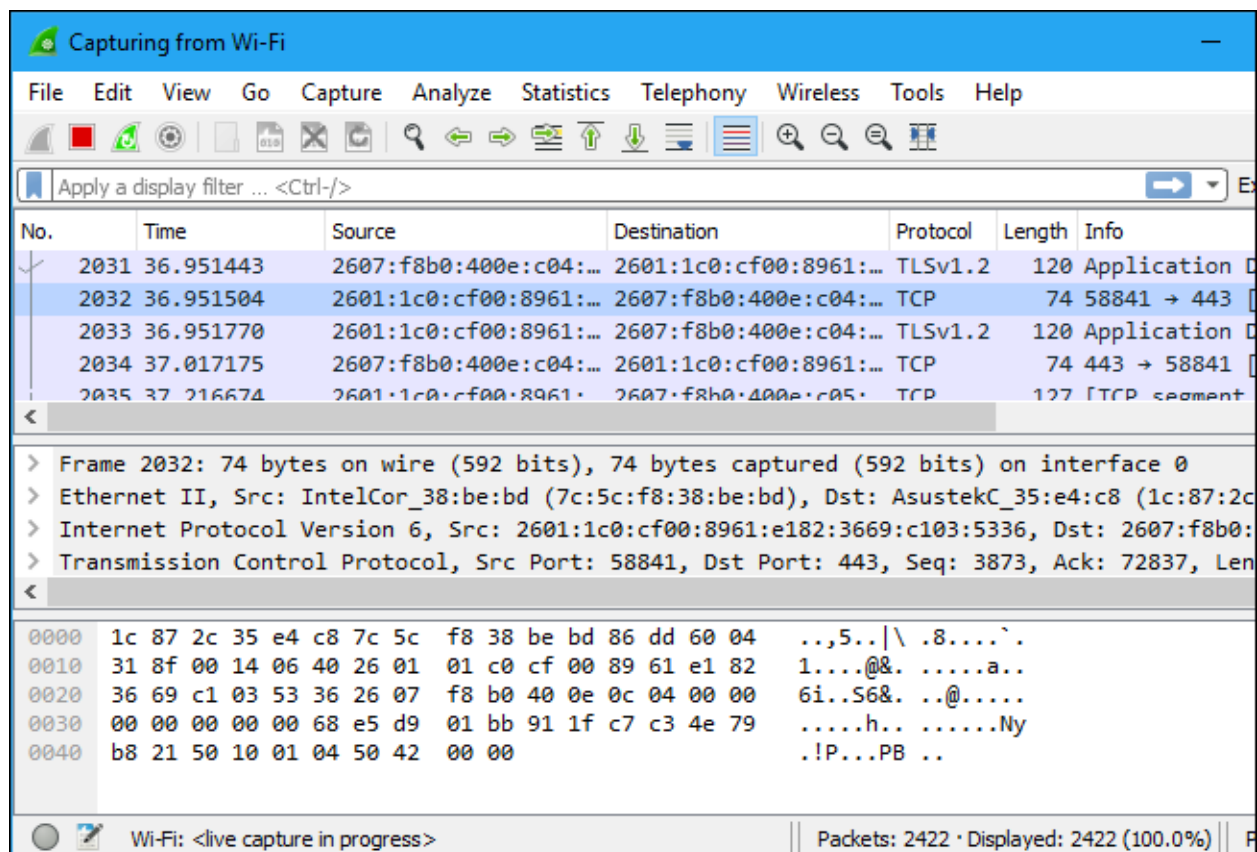
Wireshark is probably already installed! It's part of the basic package. Check your menu to verify. It's under the menu option "Sniffing & Spoofing."

1)Capturing Packets:

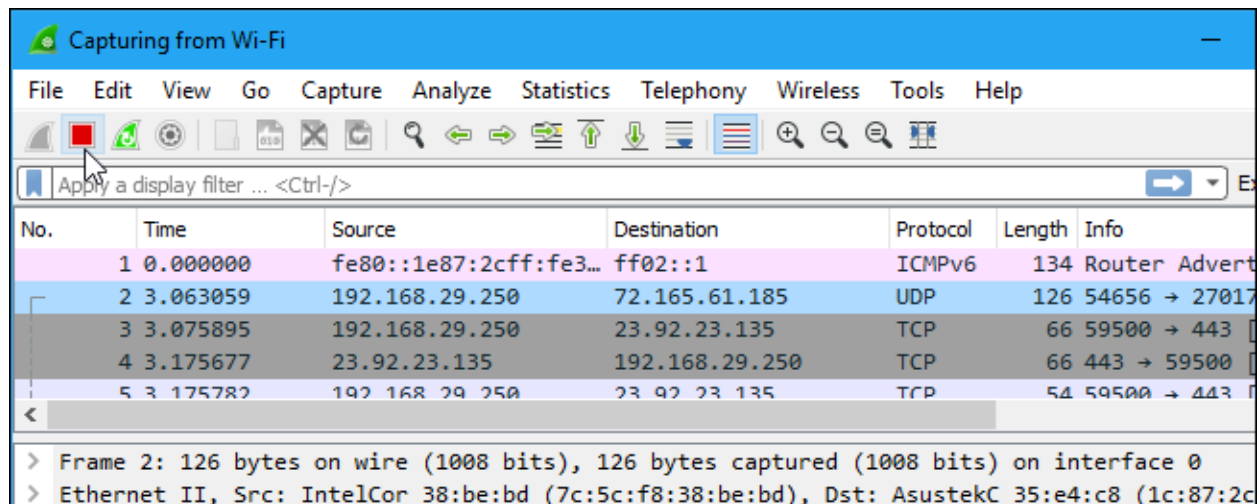
After downloading and installing Wireshark, you can launch it and double-click the name of a network interface under Capture to start capturing packets on that interface. For example, if you want to capture traffic on your wireless network, click your wireless interface.



As soon as you click the interface's name, you'll see the packets start to appear in real time. Wireshark captures each packet sent to or from your system.



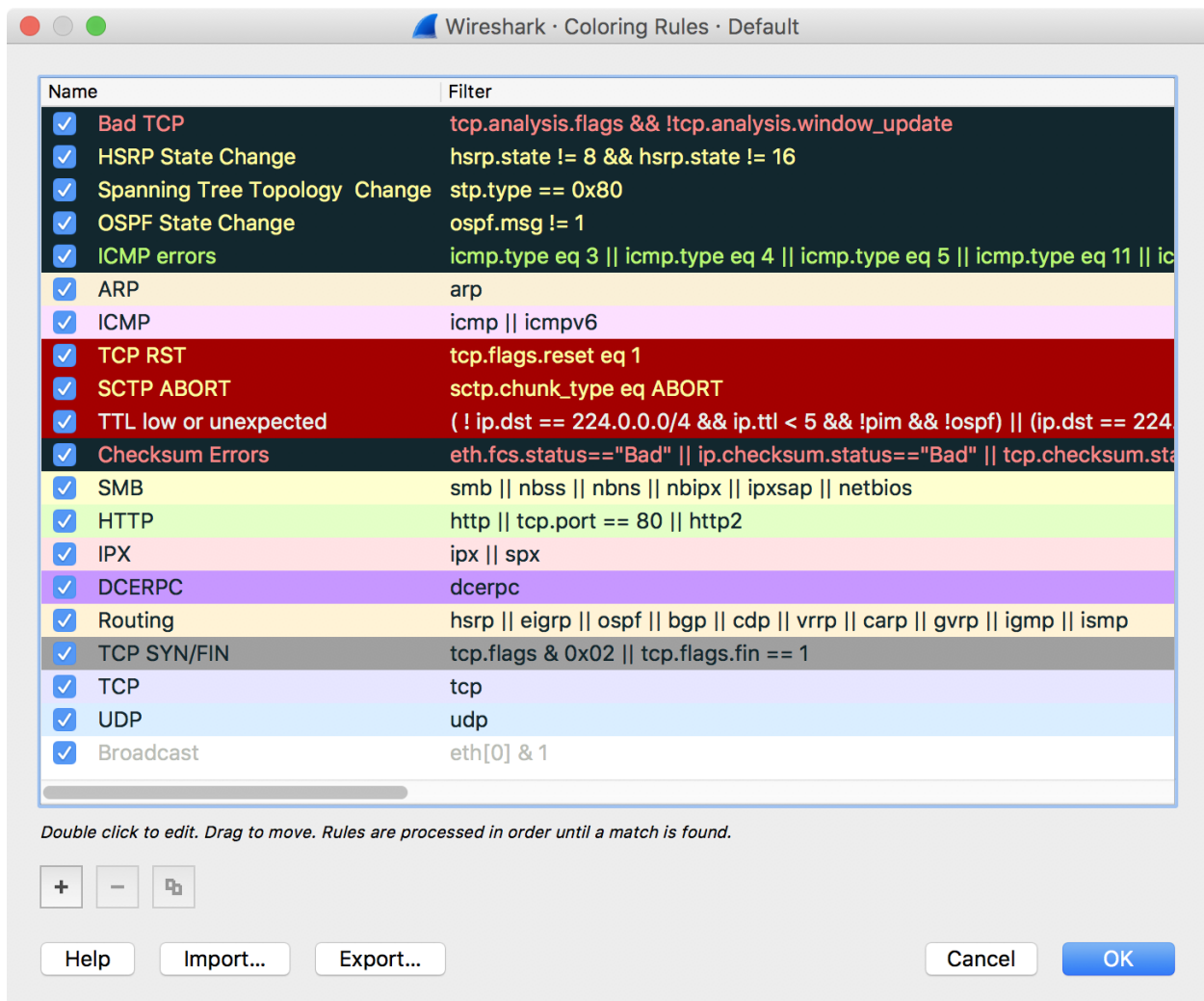
Click the red “Stop” button near the top left corner of the window when you want to stop capturing traffic.



2)Color Coding:

You'll probably see packets highlighted in a variety of different colors. Wireshark uses colors to help you identify the types of traffic at a glance. By default, light purple is TCP traffic, light blue is UDP traffic, and black identifies packets with errors—for example, they could have been delivered out of order.

To view exactly what the color codes mean, click View > Coloring Rules. You can also customize and modify the coloring rules from here, if you like.



In most cases, I apply coloring rules to individual conversations. **That's even easier. Pick a packet in a capture file, right-click it, and hover over Colorize Conversation. The slide-out menu will reveal options that allow you to define the type of conversation (IPv4, IPv6, TCP, etc).**

No.	Time	Source	Destination	Protocol	Length	Info
55	0.550795	199.181.133.61	172.16.16.154	TCP	1514	80 → 64861 [ACK] Seq=21758 Ack=382 Win=4761 L
56	0.550796	199.181.133.61	Mark/Unmark Packet	⊞M	1514	80 → 64861 [ACK] Seq=23206 Ack=382 Win=4761 L
57	0.550797	199.181.133.61	Ignore/Unignore Packet	⊞D	140	80 → 64861 [PSH, ACK] Seq=24654 Ack=382 Win=4
58	0.550828	172.16.16.154	Set/Unset Time Reference	⊞T	66	64861 → 80 [ACK] Seq=382 Ack=21758 Win=65535
59	0.550879	172.16.16.154	Time Shift...	⊞T	66	64861 → 80 [ACK] Seq=382 Ack=24654 Win=65535
60	0.550879	172.16.16.154	Packet Comment...	⊞C	66	64861 → 80 [ACK] Seq=382 Ack=24728 Win=65535
61	0.551286	199.181.133.61			1514	80 → 64861 [ACK] Seq=24728 Ack=382 Win=4761 L
62	0.551916	199.181.133.61	Edit Resolved Name		1514	80 → 64861 [ACK] Seq=26176 Ack=382 Win=4761 L
63	0.551917	199.181.133.61			1514	80 → 64861 [ACK] Seq=27624 Ack=382 Win=4761 L
64	0.551947	172.16.16.154	Apply as Filter	▶	66	64861 → 80 [ACK] Seq=382 Ack=27624 Win=65535
65	0.552659	199.181.133.61	Prepare a Filter	▶	1514	80 → 64861 [PSH, ACK] Seq=29072 Ack=382 Win=4
66	0.552691	172.16.16.154	Conversation Filter	▶	66	64861 → 80 [ACK] Seq=382 Ack=30520 Win=65535
67	0.553063	72.21.91.8	Colorize Conversation	▶		Seq=0 Ack=1 Win=65535 L
68	0.553100	172.16.16.154	SCTP	▶		Seq=1 Ack=1 Win=131744 Len=0
69	0.553292	172.16.16.154	Follow	▶		7b6 A assets.espn.go.com
70	0.553964	172.16.16.154				Seq=0 Win=65535 Len=0 MSS=14
71	0.554110	172.16.16.154	Copy	▶		s HTTP/1.1
72	0.565551	72.246.56.35				7K1 Seq=0 Ack=1 Win=14400 L
73	0.565633	172.16.16.154	Protocol Preferences	▶		
74	0.565877	172.16.16.154	Decode As...	▶		
75	0.578362	4.2.2.1	Show Packet in New Window	▶		
76	0.579477	172.16.16.154		TCP		
77	0.579590	72.21.91.8		TCP		
78	0.580420	72.21.91.8		TCP	1514	80 → 64867 [ACK] Seq=382 Ack=21758 Win=4761 L
79	0.580785	172.16.16.154		TCP	1514	80 → 64867 [ACK] Seq=382 Ack=21758 Win=4761 L
80	0.580785	172.16.16.154		TCP	66	64867 → 80 [ACK] Seq=382 Ack=21758 Win=65535
81	0.581541	72.21.91.8		TCP	1514	80 → 64867 [ACK] Seq=382 Ack=21758 Win=4761 L
82	0.581665	172.16.16.154		TCP	66	64867 → 80 [ACK] Seq=382 Ack=21758 Win=65535
83	0.581997	72.21.91.8		TCP	1514	80 → 64867 [ACK] Seq=382 Ack=21758 Win=4761 L
84	0.582063	172.16.16.154		TCP	1514	80 → 64867 [ACK] Seq=382 Ack=21758 Win=4761 L
85	0.582063	172.16.16.154		TCP	66	64867 → 80 [ACK] Seq=333 Ack=7241 Win=128160

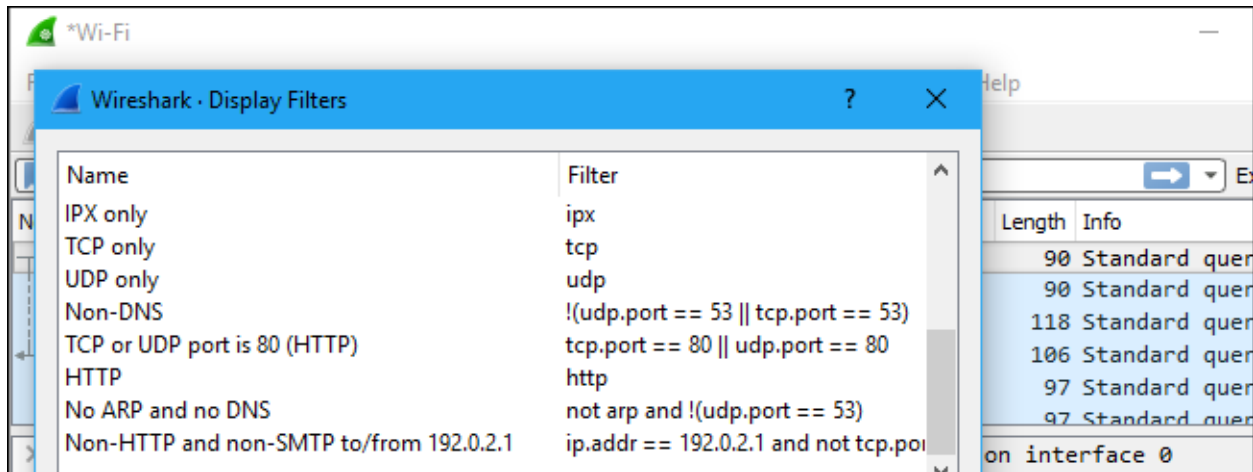
3)Filtering Packets:

If you're trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close down all other applications using the network so you can narrow down the traffic. Still, you'll likely have a large amount of packets to sift through. That's where Wireshark's filters come in.

The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type "dns" and you'll see only DNS packets. When you start typing, Wireshark will help you autocomplete your filter.

*Wi-Fi						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
[Icons]						
dns						
No.	Time	Source	Destination	Protocol	Length	Info
305	5.248733	2601:1c0:cf00:8961::...	2601:1c0:cf00:8961::...	DNS	90	Standard quer
306	5.249092	2601:1c0:cf00:8961::...	2601:1c0:cf00:8961::...	DNS	90	Standard quer
307	5.269967	2601:1c0:cf00:8961::...	2601:1c0:cf00:8961::...	DNS	118	Standard quer
308	5.270325	2601:1c0:cf00:8961::...	2601:1c0:cf00:8961::...	DNS	106	Standard quer

You can also click Analyze > Display Filters to choose a filter from among the default filters included in Wireshark. From here, you can add your own custom filters and save them to easily access them in the future.



Display Filter:

- Show only [SMTP](#) (port 25) and [ICMP](#) traffic:
`tcp.port eq 25 or icmp`
- Show only traffic in the LAN (192.168.x.x), between workstations and servers -- no Internet:
`ip.src==192.168.0.0/16 and ip.dst==192.168.0.0/16`
- [TCP](#) buffer full -- *Source is instructing Destination to stop sending data*
`tcp.window_size == 0 && tcp.flags.reset != 1`
- Match HTTP requests where the last characters in the uri are the characters "gl=se":
`http.request.uri matches "gl=se$"`

Note: The \$ character is a PCRE punctuation character that matches the end of a string, in this case the end of http.request.uri field.

- Filter by a protocol (e.g. SIP) and filter out unwanted IPs:
`ip.src != xxx.xxx.xxx.xxx && ip.dst != xxx.xxx.xxx.xxx && sip`
- Capture only traffic to or from IP address 172.18.5.4:
`host 172.18.5.4`

- Capture traffic from a range of IP addresses:

`src net 192.168.0.0/24 (OR) src net 192.168.0.0 mask 255.255.255.0`

`(OR) dst net 192.168.0.0/24 (OR) dst net 192.168.0.0 mask 255.255.255.0`

- Capture only DNS (port 53) traffic:

`port 53`

- Capture non-HTTP and non-SMTP traffic on your server (both are equivalent):

`host www.example.com and not (port 80 or port 25) (OR)`

`host www.example.com and not port 80 and not port 25`

- Capture traffic within a range of ports

`tcp portrange 1501-1549`

Display Filter comparison operators:

English	C-like	Description	Example
eq	==	Equal	<code>ip.src==10.0.0.5</code>
ne	!=	Not equal	<code>ip.src!=10.0.0.5</code>
gt	>	Greater than	<code>frame.len > 10</code>
lt	<	Less than	<code>frame.len < 128</code>
ge	>=	Greater than or equal to	<code>frame.len ge 0x100</code>
le	<=	Less than or equal to	<code>frame.len <= 0x20</code>
contains		Protocol, field or slice contains a value	<code>sip.To contains "a1762"</code>
matches	~	Protocol or text field matches a Perl-compatible regular expression	<code>http.host matches "acme\.(org com net)"</code>
bitwise_and	&	Bitwise AND is non-zero	<code>tcp.flags & 0x02</code>

Display Filter Logical Operations:

English	C-like	Description	Example
and	&&	Logical AND	<code>ip.src==10.0.0.5 and tcp.flags.fin</code>
or		Logical OR	<code>ip.scr==10.0.0.5 or ip.src==192.1.1.1</code>
xor	^^	Logical XOR	<code>tr.dst[0:3] == 0.6.29 xor tr.src[0:3] == 0.6.29</code>
not	!	Logical NOT	<code>not llc</code>
[...]		Subsequence	See “Slice Operator” below.
in		Set Membership	<code>http.request.method in {"HEAD" "GET"}</code> . See “Membership Operator” below.

NOTE: WIRESHARK CHEAT SHEET PDF FILE

4)Inspecting Packets: