

3

INTRODUCTION TO WIRESHARK



As mentioned in Chapter 1, several packet-sniffing applications are available for performing network analysis, but we'll focus mostly on Wireshark in this book. This chapter introduces Wireshark.

A Brief History of Wireshark

Wireshark has a very rich history. Gerald Combs, a computer science graduate of the University of Missouri at Kansas City, originally developed it out of necessity. The first version of Combs's application, called Ethereal, was released in 1998 under the GNU Public License (GPL).

Eight years after releasing Ethereal, Combs left his job to pursue other career opportunities. Unfortunately, his employer at that time had full rights to the Ethereal trademarks, and Combs was unable to reach an agreement that would allow him to control the Ethereal brand. Instead, Combs and the rest of the development team rebranded the project as *Wireshark* in mid-2006.

Wireshark has grown dramatically in popularity, and its collaborative development team now boasts more than 500 contributors. The program that exists under the Ethereal name is no longer being developed.

The Benefits of Wireshark

Wireshark offers several benefits that make it appealing for everyday use. Aimed at both the up-and-coming and the expert packet analyst, it offers a variety of features to entice each. Let's examine Wireshark according to the criteria defined in Chapter 1 for selecting a packet-sniffing tool.

Supported protocols Wireshark excels in the number of protocols that it supports—more than 1,000 as of this writing. These range from common ones like IP and DHCP to more advanced proprietary protocols like DNP3 and BitTorrent. And because Wireshark is developed under an open source model, new protocol support is added with each update.

NOTE

In the unlikely event that Wireshark doesn't support a protocol you need, you can code that support yourself. Then you can submit your code to the Wireshark developers for consideration for inclusion in the application. You can learn about what is required to contribute code to the Wireshark project at <https://www.wireshark.org/develop.html>.

User-friendliness The Wireshark interface is one of the easiest to understand of any packet-sniffing application. It is GUI based, with clearly written context menus and a straightforward layout. It also provides several features designed to enhance usability, such as protocol-based color coding and detailed graphical representations of raw data. Unlike some of the more complicated command line-driven alternatives, like tcpdump, the Wireshark GUI is accessible to those just entering the world of packet analysis.

Cost Since it's open source and released under the GNU Public License (GPL), Wireshark's pricing can't be beat: it's absolutely free. You can download and use Wireshark for any purpose, whether personal or commercial.

NOTE

Although Wireshark may be free, some people have made the mistake of paying for it by accident. If you search for packet sniffers on eBay, you may be surprised by how many people would love to sell you a "professional enterprise license" for Wireshark for the low, low price of \$39.95. If you decide you really want to buy it, give me a call, and we can talk about some oceanfront property in Kentucky I have for sale!

Program support A software package's level of support can make or break it. Freely distributed software such as Wireshark may not come with any formal support, so the open source community often relies on its user base to provide assistance. Luckily for us, the Wireshark community is one of the most active of any open source project. The Wireshark website links directly to several forms of support, including online documentation; a wiki; FAQs; and a place to sign up for the Wireshark mailing list, which is monitored by most of the program's top developers. Paid support for Wireshark is also available from Riverbed Technology.

Source code access Wireshark is open source software, so you can access the code at any time. This can be useful for troubleshooting application issues, understanding how protocol dissectors work, or making your own contributions.

Operating system support Wireshark supports all major modern operating systems, including Windows, Linux-based, and OS X platforms. You can view a complete list of supported operating systems on the Wireshark home page.

Installing Wireshark

The Wireshark installation process is surprisingly simple. However, before you install Wireshark, make sure that your system meets the following requirements:

- Any modern 32-bit x86 or 64-bit CPU
- 400MB available RAM, but more for larger capture files
- At least 300MB of available storage space, plus space for capture files
- NIC that supports promiscuous mode
- WinPcap/libpcap capture driver

The WinPcap capture driver is the Windows implementation of the pcap packet-capturing application programming interface (API). Simply put, this driver interacts with your operating system to capture raw packet data, apply filters, and switch the NIC in and out of promiscuous mode.

Although you can download WinPcap separately (from <http://www.winpcap.org/>), it is typically better to install WinPcap from the Wireshark installation package, because the included version of WinPcap has been tested to work with Wireshark.

Installing on Windows Systems

The current version of Wireshark is tested to support versions of Windows that are still within their extended support lifetime. As of the writing of this book, that encompasses Windows Vista; Windows 7; Windows 8;

Windows 10; and Windows Servers 2003, 2008, and 2012. While Wireshark will often work on other versions of Windows (like Windows XP), those versions are not officially supported.

The first step when installing Wireshark on Windows is to obtain the latest installation build from the official Wireshark web page, <http://www.wireshark.org/>. Navigate to the Download Wireshark section on the website and choose a release mirror. Once you've downloaded the package, follow these steps:

1. Double-click the *.exe* file to begin installation and then click **Next** in the introductory window.
2. Read the licensing agreement and click **I Agree** if you agree.
3. Select the components of Wireshark you wish to install, as shown in Figure 3-1. For our purposes, you can accept the defaults by clicking **Next**.

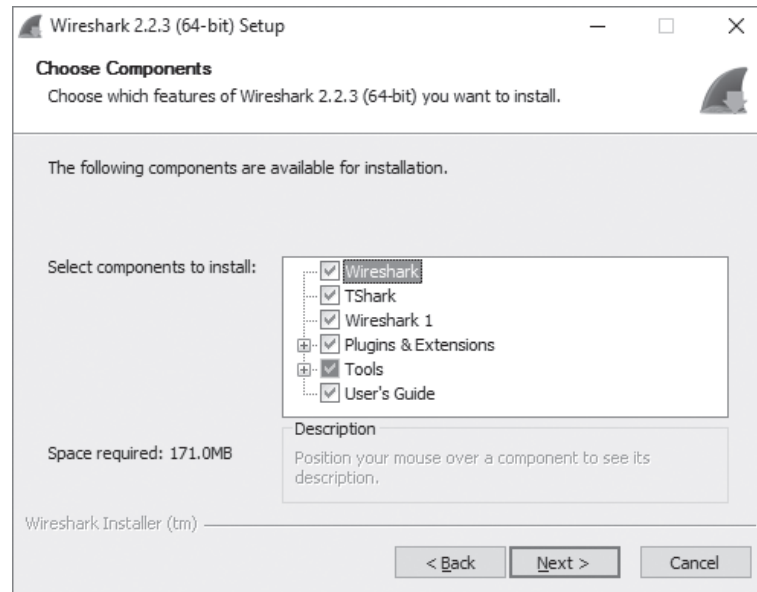


Figure 3-1: Choosing the Wireshark components you wish to install

4. Click **Next** in the Additional Tasks window.
5. Select the location where you wish to install Wireshark and click **Next**.
6. When the dialog asks whether you want to install WinPcap, first make sure the **Install WinPcap** box is checked, as shown in Figure 3-2. Then click **Install**. The installation process should begin.
7. About halfway through the Wireshark installation, the WinPcap installation should start. When it does, click **Next** in the introductory window, read the licensing agreement, and click **I Agree**.
8. You'll be given the option to install USBPcap, a utility for collecting data from USB devices. Select the appropriate check box if you wish to do so and click **Next**.

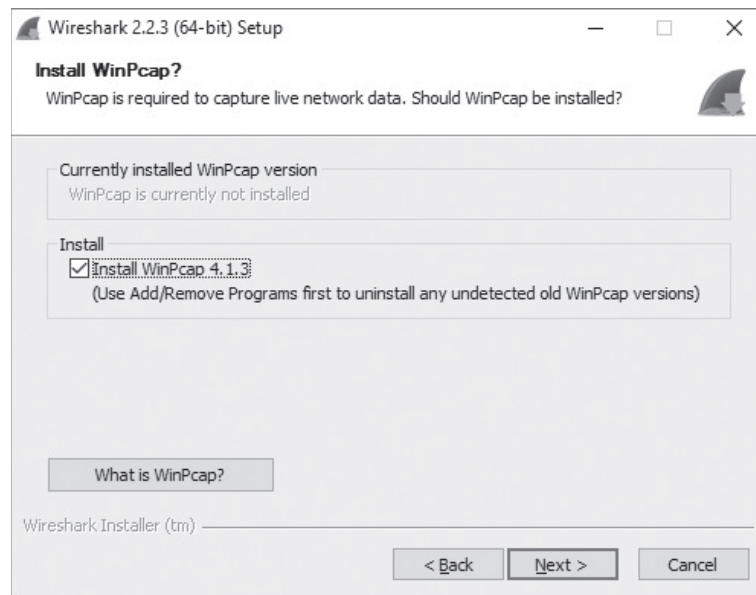


Figure 3-2: Selecting the option to install the WinPcap driver

9. WinPcap and, if you selected it, USBPcap should install on your computer. After this installation is complete, click **Finish**.
10. Wireshark should complete its installation. When it's finished, click **Next**.
11. In the installation confirmation window, click **Finish**.

Installing on Linux Systems

Wireshark works on most modern Unix-based platforms. It can be installed either by using the distributions package manager of choice or by downloading and installing the package appropriate for your distribution. It isn't realistic to cover installation procedures for everyone, so we'll just look at a few.

Typically, for system-wide software, root access is a requirement. However, local software installations compiled from source can usually be installed without root access.

RPM-Based Systems

If you're using Red Hat Linux or a distribution based on it, like CentOS, then it's likely the OS has the Yum package management tool installed by default. If that's the case, you may be able to install Wireshark the quick way by pulling it from the distribution's software repository. To do this, open a console window and enter the following command:

```
$ sudo yum install wireshark
```

If any dependencies are needed, you'll be prompted to install them. If everything completes successfully, then you should be able to run Wireshark from the command line and access it via the GUI.

DEB-Based Systems

Most DEB-based distributions, such as Debian or Ubuntu, include the APT package management tool, which allows you to install Wireshark from the OS software repository. To install Wireshark with this tool, open a console window and enter the following:

```
$ sudo apt-get install wireshark wireshark-qt
```

Once again, you'll be prompted to install any required dependencies to complete the installation.

Compiling from Source

Due to changes in operation system architecture and Wireshark features, the instructions for compiling Wireshark from source might change over time. That's one reason it's recommended to use your operating system package manager to perform the installation. However, if your Linux distribution doesn't use an automated package management software or you require a specialized installation, Wireshark can be installed manually by compiling it from source. To compile do so, complete the following steps:

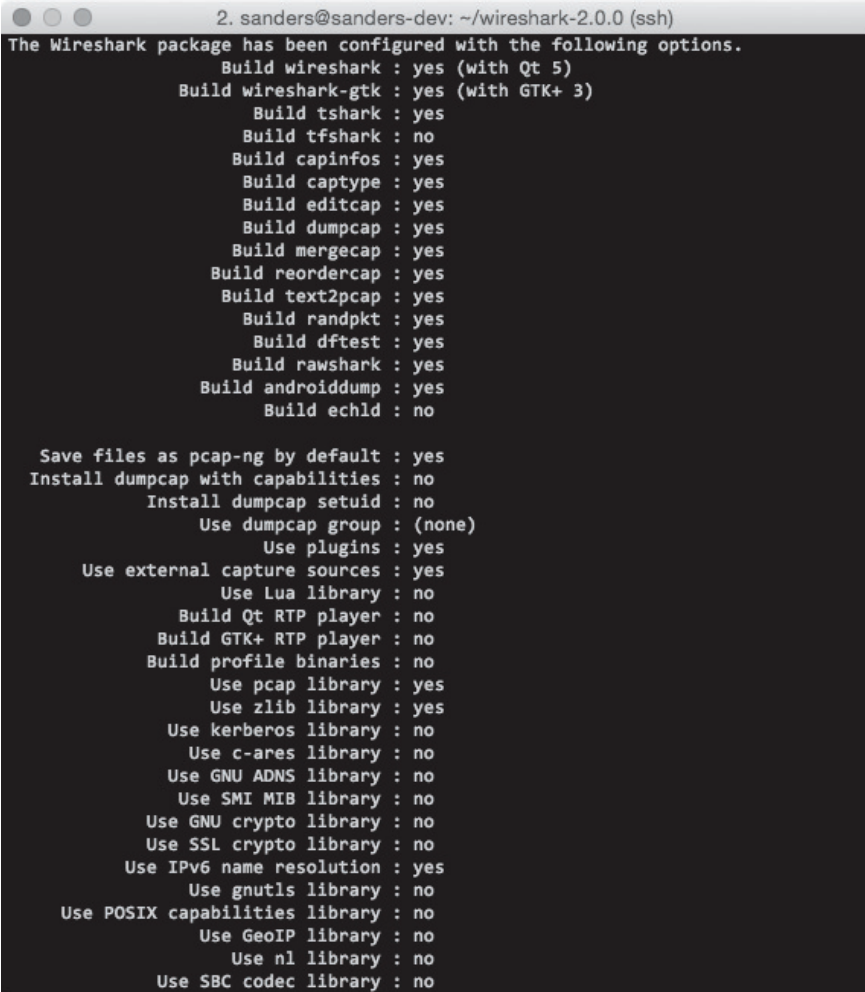
1. Download the source package from the Wireshark web page.
2. Extract the archive by entering the following (substituting the filename of your downloaded package as appropriate):

```
$ tar -jxvf <file_name_here>.tar.bz2
```

3. Before configuring and installing Wireshark, a few dependencies may be required depending on your chosen Linux flavor. For example, Ubuntu 14.04 requires the installation of a few other packages for Wireshark to work. These can be installed by issuing the following command (you'll need to do this as a root-level user or by invoking `sudo` before the command):

```
$ sudo apt-get install pkg-config bison flex qt5-default libgtk-3-dev  
libpcap-dev qttools5-dev-tools
```

4. After installing prerequisites, navigate to the directory where the Wireshark files were extracted.
5. Configure the source so that it will build correctly for your distribution of Linux by using the command `./configure`. If you wish to deviate from the default installation options, you can specify those options at this point in the installation. If any dependencies are missing, you'll most likely receive an error. You must install and configure those dependencies before proceeding. If configuration is successful, you should see a message noting success, as shown in Figure 3-3.



```

2. sanders@sanders-dev: ~/wireshark-2.0.0 (ssh)
The Wireshark package has been configured with the following options.
  Build wireshark : yes (with Qt 5)
  Build wireshark-gtk : yes (with GTK+ 3)
    Build tshark : yes
    Build tfshark : no
  Build capinfos : yes
  Build captype : yes
  Build editcap : yes
  Build dumpcap : yes
  Build mergecap : yes
  Build reordercap : yes
  Build text2pcap : yes
  Build randpkt : yes
  Build dftest : yes
  Build rawshark : yes
  Build androiddump : yes
  Build echld : no

  Save files as pcap-ng by default : yes
  Install dumpcap with capabilities : no
    Install dumpcap setuid : no
      Use dumpcap group : (none)
      Use plugins : yes
  Use external capture sources : yes
    Use Lua library : no
    Build Qt RTP player : no
    Build GTK+ RTP player : no
    Build profile binaries : no
    Use pcap library : yes
    Use zlib library : yes
    Use kerberos library : no
    Use c-ares library : no
    Use GNU ADNS library : no
    Use SMI MIB library : no
    Use GNU crypto library : no
    Use SSL crypto library : no
  Use IPv6 name resolution : yes
    Use gnutls library : no
  Use POSIX capabilities library : no
    Use GeoIP library : no
    Use nl library : no
    Use SBC codec library : no

```

Figure 3-3: When the `./configure` command is successful, a message is displayed with the selected configurations.

6. Enter the `make` command to build the source into a binary.
7. Initiate the final installation with `sudo make install`.
8. Run `sudo /sbin/ldconfig` to complete the installation.

NOTE

If you run into an error following these steps, you may have to install an additional package.

Installing on OS X Systems

To install Wireshark on OS X, complete these steps:

1. Download the OS X package from the Wireshark web page.
2. Run the installation utility and proceed through its steps. Once you've accepted the required end user license agreement, you'll have the option to select the installation location.
3. Complete the installation wizard.

Wireshark Fundamentals

Once you've successfully installed Wireshark on your system, you can begin to familiarize yourself with it. Now you finally get to open your fully functioning packet sniffer and see . . . absolutely nothing!

Okay, so Wireshark isn't very interesting when you first open it. For things to really get exciting, you need to get some data.

Your First Packet Capture

To get packet data into Wireshark, you'll perform your first packet capture. You may be thinking, "How am I going to capture packets when nothing is wrong on the network?"

First, there is *always* something wrong on the network. If you don't believe me, then go ahead and send an email to all of your network users and let them know that everything is working perfectly.

Secondly, there doesn't need to be something wrong in order for you to perform packet analysis. In fact, most packet analysts spend more time analyzing problem-free traffic than traffic that they are troubleshooting. After all, you need a baseline for comparison to effectively troubleshoot network traffic. For example, if you ever hope to solve a problem with DHCP by analyzing its traffic, you must understand what the flow of working DHCP traffic looks like.

More broadly, to find anomalies in daily network activity, you must know what normal daily network activity looks like. When your network is running smoothly, your observations become a baseline representing what traffic looks like in a normal state.

So, let's capture some packets!

1. Open Wireshark.
2. From the main drop-down menu, select **Capture** and then **Options**. You should see a dialog listing the various interfaces that can be used to capture packets, along with some basic information about each one (Figure 3-4). Take note of the Traffic heading, which shows a line graph indicating the amount of traffic currently passing through that interface. Peaks on a line tell you that you are actually capturing packets. If you aren't, the line will be flat. You can also expand each interface by clicking the arrow to the left of it to see the addressing information, such as the MAC address or IP address, tied to it.
3. Click the interface you wish to use and click **Start**. Data should begin filling the window.
4. Wait about a minute or so, and when you are ready to stop the capture and view your data, click the **Stop** button from the Capture drop-down menu.

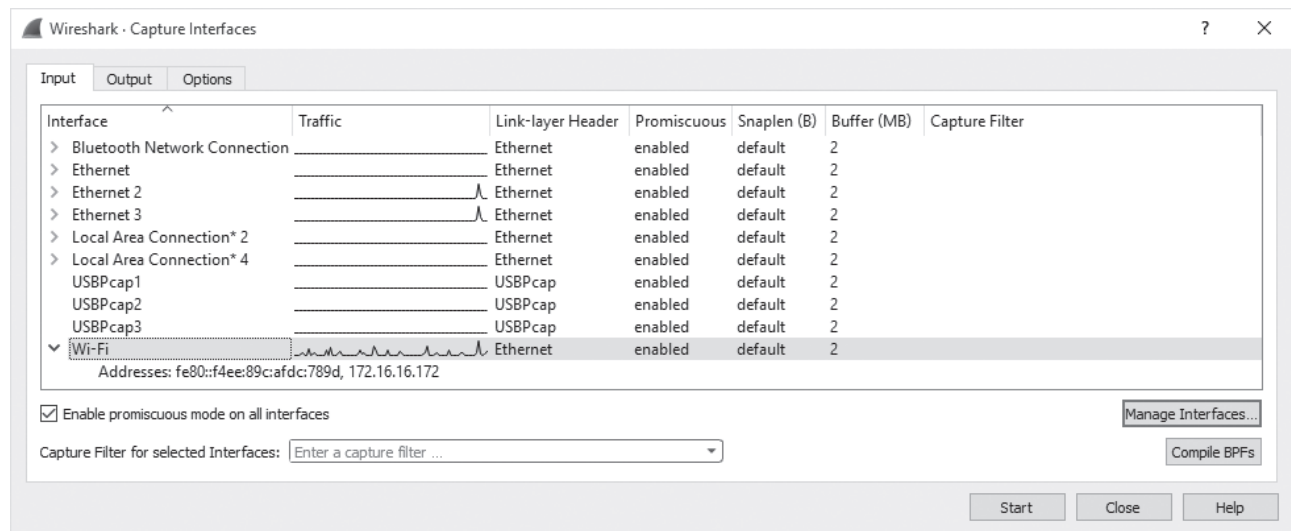


Figure 3-4: Selecting an interface on which to perform your packet capture

Once you have completed these steps and finished the capture process, the Wireshark main window should be alive with data. As a matter of fact, you might be overwhelmed by the amount of data that appears, but it will all start to make sense quickly as we break down the main window of Wireshark one piece at a time.

Wireshark's Main Window

You'll spend most of your time in the Wireshark main window. This is where all of the packets you capture are displayed and broken down into a more understandable format. Using the packet capture you just made, let's take a look at Wireshark's main window, shown in Figure 3-5.

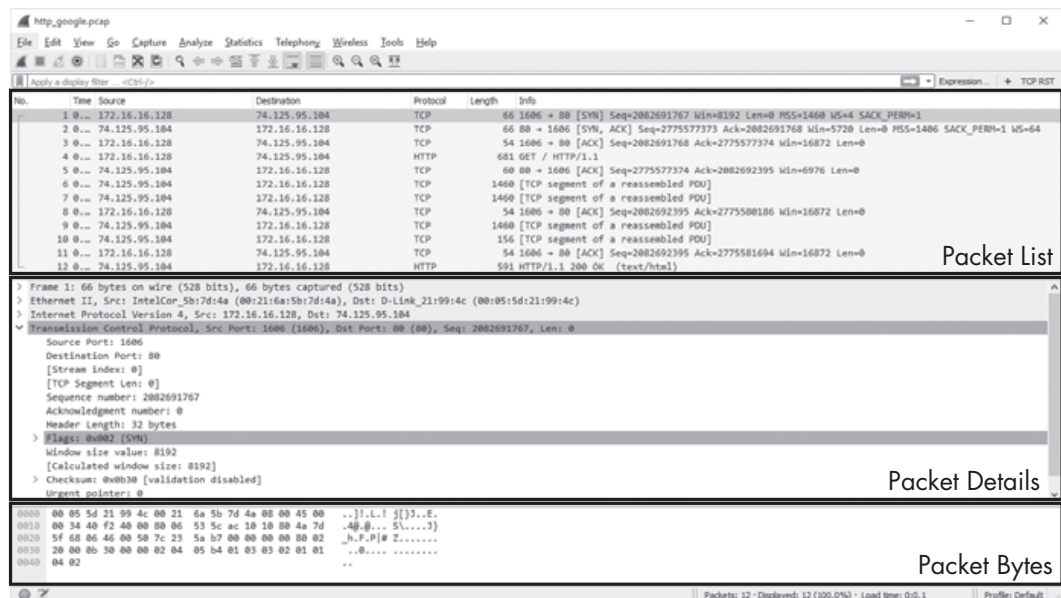


Figure 3-5: The Wireshark main window uses a three-pane design.

The three panes in the main window—Packet List, Packet Details, and Packet Bytes from top to bottom—depend on one another. To view the details of an individual packet in the Packet Details pane, you must first select it in the Packet List pane. When you select a portion of the packet in the Packet Details pane, the Packet Bytes pane displays the bytes that correspond with that portion.

NOTE

Notice that Figure 3-5 lists a few different protocols in the Packet List pane. There is no visual separation of protocols on different layers (other than via color coding); all packets are shown as they are received on the wire.

Here's what each pane contains:

Packet List The top pane displays a table containing all packets in the current capture file. It has columns containing the packet number, the relative time the packet was captured, the source and destination of the packet, the packet's protocol, and some general information found in the packet.

NOTE

When I refer to traffic, I'm referring to all packets displayed in the Packet List pane. When I refer to DNS traffic specifically, I mean the DNS protocol packets in the Packet List pane.

Packet Details The middle pane contains a hierarchical display of information about a single packet and can be collapsed or expanded to show all of the information collected about the individual packet.

Packet Bytes The lower pane—perhaps the most confusing—displays a packet in its raw, unprocessed form; that is, it shows what the packet looks like as it travels across the wire. This is raw information with nothing warm or fuzzy to make it easier to follow. We'll discuss methods for interpreting this type of data in Appendix B.

Wireshark Preferences

Wireshark has several preferences that can be customized to meet your needs. To access Wireshark's preferences, select **Edit** from the main drop-down menu and click **Preferences**. You'll see the Preferences dialog, which contains several customizable options, as shown in Figure 3-6.

Wireshark's preferences are divided into six major sections plus an Advanced section:

Appearance These preferences determine how Wireshark presents data. You can change most options here according to your personal preferences, including whether to save window positions, the layout of the three main panes, the placement of the scroll bar, the placement of the Packet List pane columns, the fonts used to display the captured data, and the background and foreground colors.

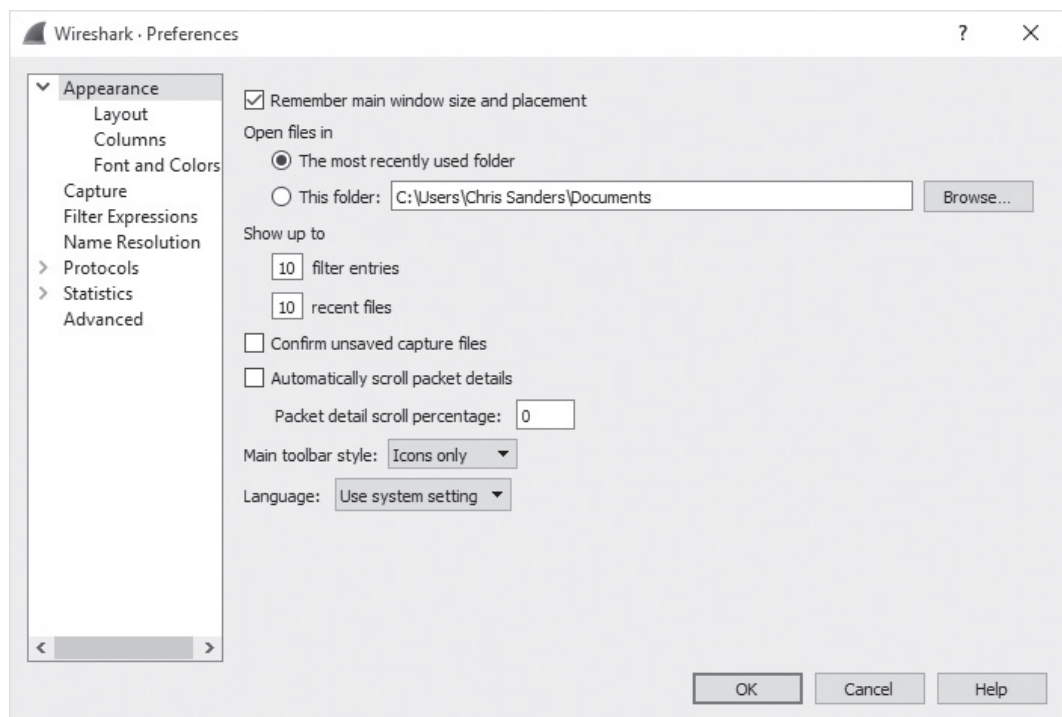


Figure 3-6: You can customize Wireshark using the Preferences dialog options.

Capture These preferences allow you to specify options related to the way packets are captured, including your default capture interface, whether to use promiscuous mode by default, and whether to update the Packet List pane in real time.

Filter Expressions Later we will discuss how Wireshark allows you to filter traffic based on specific criteria. This section of the Preferences dialog allows you to create and manage those filters.

Name Resolution Through these preferences, you can activate features of Wireshark that allow it to resolve addresses into more recognizable names (including MAC, network, and transport name resolution) and specify the maximum number of concurrent name resolution requests.

Protocols This section allows you to manipulate options related to the capture and display of the various packets Wireshark is capable of decoding. Not every protocol has configurable preferences, but some have several options that can be changed. These options are best left at their defaults unless you have a specific reason to change them.

Statistics This section provides a few configurable options for Wireshark's statistical features, which will be covered in more depth in Chapter 5.

Advanced Settings that don't fit neatly into any of the previous categories can be found here. Editing these settings is something typically only done by Wireshark power users.

Packet Color Coding

If you are anything like me, you enjoy shiny objects and pretty colors. If so, you probably got excited when you saw all those different colors in the Packet List pane, as in the example in Figure 3-7 (well, the figure is in black and white if you're reading this book in print, but you get the idea). It may seem as if these colors are randomly assigned to each packet, but this isn't the case.

27	1.807280	172.16.16.128	172.16.16.255	NBNS	92 Name query NB ISATAP<00>
28	2.557340	172.16.16.128	172.16.16.255	NBNS	92 Name query NB ISATAP<00>
29	3.009402	172.16.16.128	4.2.2.1	DNS	86 standard query 0xb86a PTR 128.16.16.172.in-addr.arpa
30	3.050866	4.2.2.1	172.16.16.128	DNS	163 standard query response 0xb86a No such name
31	3.180870	172.16.16.128	157.166.226.25	TCP	66 2918-80 [SYN] Seq=0 win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
32	3.241650	157.166.226.25	172.16.16.128	TCP	66 80-2918 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0 MSS=1406 SACK_PE
33	3.241744	172.16.16.128	157.166.226.25	TCP	54 2918-80 [ACK] Seq=1 Ack=1 win=16872 Len=0
34	3.241956	172.16.16.128	209.85.225.148	TCP	54 2867-80 [RST, ACK] Seq=1 Ack=1 win=0 Len=0
35	3.242063	172.16.16.128	209.85.225.118	TCP	54 2866-80 [RST, ACK] Seq=1 Ack=1 win=0 Len=0
36	3.242129	172.16.16.128	209.85.225.118	TCP	54 2865-80 [RST, ACK] Seq=1 Ack=1 win=0 Len=0
37	3.242223	172.16.16.128	209.85.225.133	TCP	54 2864-80 [RST, ACK] Seq=1 Ack=1 win=0 Len=0
38	3.242292	172.16.16.128	209.85.225.133	TCP	54 2863-80 [RST, ACK] Seq=1 Ack=1 win=0 Len=0
39	3.242311	172.16.16.128	157.166.226.25	HTTP	804 GET / HTTP/1.1

Figure 3-7: Wireshark's color coding allows for quick protocol identification.

Each packet is displayed in a certain color for a reason. The color can reflect the packet's protocol and specific field values. For example, all UDP traffic is blue and all HTTP traffic is green by default. The color coding allows you to quickly differentiate between various protocols so that you don't need to read the protocol field in the Packet List pane for every packet. You'll find that this greatly speeds up the time it takes to browse through large capture files.

Wireshark makes it easy to see which colors are assigned to each protocol through the Coloring Rules window, shown in Figure 3-8. To open this window, select **View** from the main drop-down menu and click **Coloring Rules**.

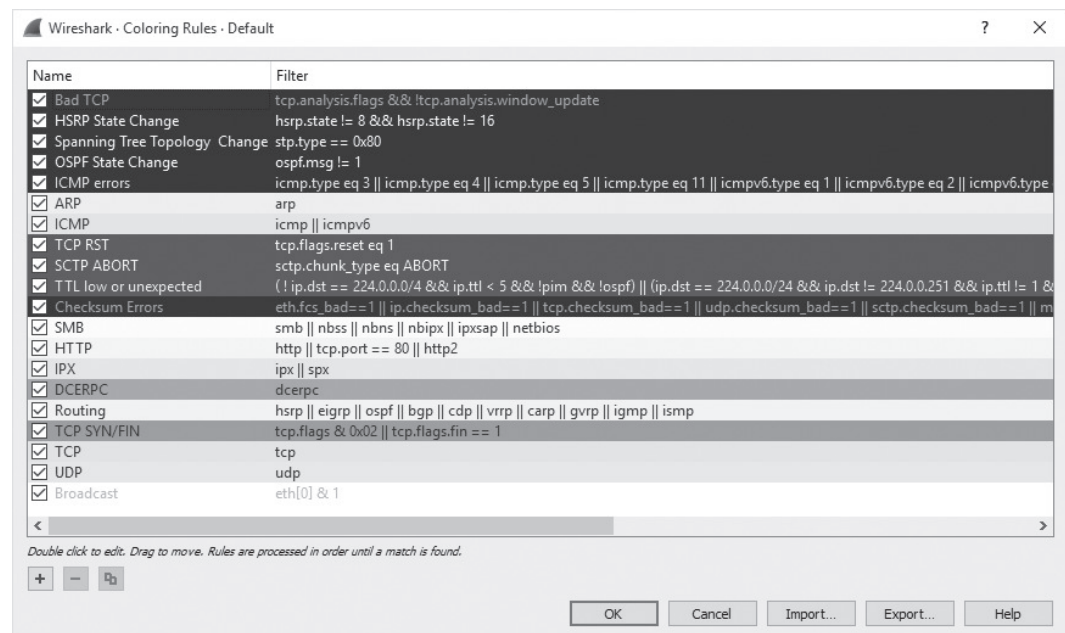


Figure 3-8: The Coloring Rules window lets you view and modify the coloring of packets.

Coloring rules are based on Wireshark filters, which we will look at in Chapter 4. Using these filters, you can define your own coloring rules and modify existing ones. For example, to change the background color used for HTTP traffic from the default green to lavender, follow these steps:

1. Open Wireshark and access the Coloring Rules window (**View ▶ Coloring Rules**).
2. Find the HTTP coloring rule in the coloring rules list and select it by clicking it once.
3. You'll see the foreground and background colors listed at the bottom of the screen, as shown in Figure 3-9.

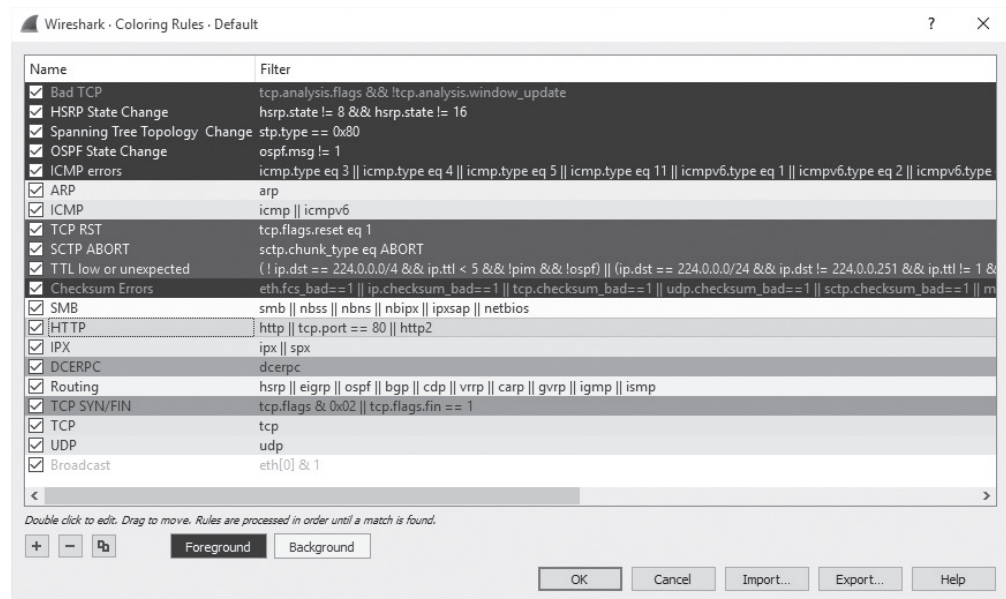


Figure 3-9: When editing a color filter, you can modify both the foreground and background colors.

4. Click the **Background Color** button.
5. Select the color you wish to use on the color wheel and click **OK**.
6. Click **OK** once more to accept the changes and return to the main window. The user interface should then reload itself to reflect the updated color scheme.

As you work with Wireshark on your network, you'll begin to notice that you deal with certain protocols more than others. Here's where color-coded packets can make your life a lot easier. For example, if you think that there is a rogue DHCP server on your network handing out IP leases, you could modify the coloring rule for the DHCP protocol so that it shows up in bright yellow (or some other easily identifiable color). This would allow you to pick out all DHCP traffic much more quickly, making your packet analysis more efficient.

NOTE

Not too long ago, I was discussing Wireshark coloring rules during a presentation to a local group of students. One student was relieved to find out he could edit the coloring rules because he was color-blind and had trouble distinguishing certain protocols based on the default coloring. The ability to modify the default coloring rules thus provides some degree of accessibility.

Configuration Files

It's helpful to understand where Wireshark stores various configuration settings should you ever need to modify those files directly. You can find the location of the Wireshark configuration files by selecting **Help** from the main drop-down menu, choosing **About Wireshark**, and clicking the **Folders** tab. This window is shown in Figure 3-10.

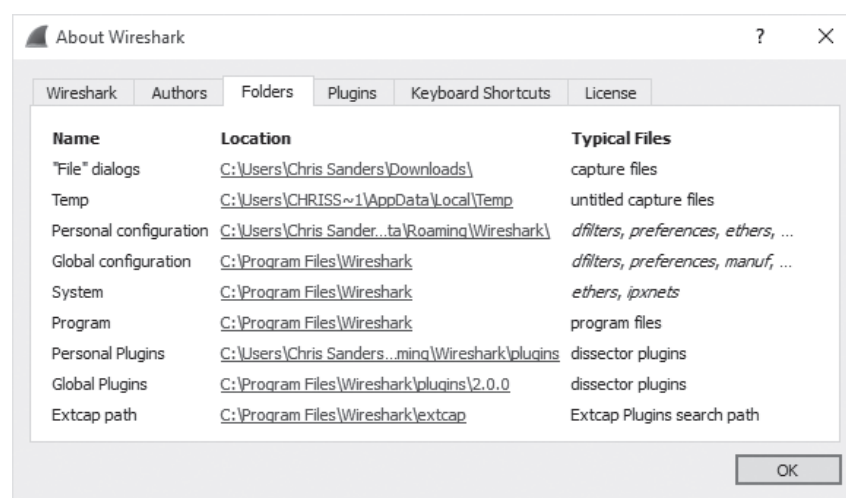


Figure 3-10: Locating Wireshark configuration files

The two most important locations in terms of Wireshark customization are the personal and global configuration directories. The global configuration directory contains all of the default settings for Wireshark and is where the default profile stores its settings. The personal configuration folder contains customized settings and profiles unique to your account. Any new profiles you create will be stored in a subdirectory of the personal configuration folder using whatever name you provide.

The difference between global and personal configuration directories is an important one, because any changes made to the global configuration files will affect every Wireshark user on a system.

Configuration Profiles

After learning about Wireshark's preferences, you may find that sometimes you want to use one set of preferences but then quickly switch to another set to address a different scenario. Instead of making you manually reconfigure your preferences every time this occurs, Wireshark introduced configuration profiles, which allow users to create saved sets of preferences.

A configuration profile stores the following:

- Preferences
- Capture filters
- Display filters
- Coloring rules
- Disabled protocols
- Forced decodes
- Recent settings, such as pane sizes, view menu settings, and column widths
- Protocol-specific tables, such as SNMP users and custom HTTP headers

To view the list of profiles, click **Edit** in the main drop-down menu and choose the **Configuration Profiles** option. Alternatively, you can right-click the profiles section at the bottom right side of the screen and select the **Manage Profiles** option. When you arrive at the Configuration Profiles window, you'll see that Wireshark comes with a few standard profiles, including the Default, Bluetooth, and Classic profiles shown in Figure 3-11. The Latency Investigation profile is a custom profile I've added and is in plaintext, while the global and default profiles are in italics.

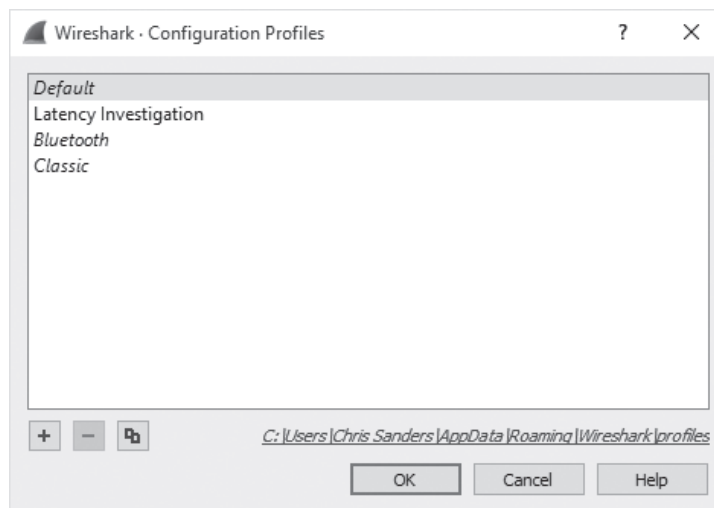


Figure 3-11: Viewing configuration profiles

The Configuration Profiles window allows you to create, copy, delete, and apply configuration profiles. The process of creating a new profile is very simple.

1. Configure Wireshark with the settings you'd like to save to a profile.
2. Proceed to the Configuration Profiles window by clicking **Edit** in the main drop-down menu. Select the **Configuration Profiles** option.
3. Click the **Plus (+)** button and give the profile a descriptive name.
4. Click **OK**.

When you'd like to switch profiles, you can go to the Configuration Profile window, click the profile name, and click **OK**. You can do this more quickly by clicking the Profile heading at the bottom right of the Wireshark window and selecting the profile you'd like to use, as shown in Figure 3-12.

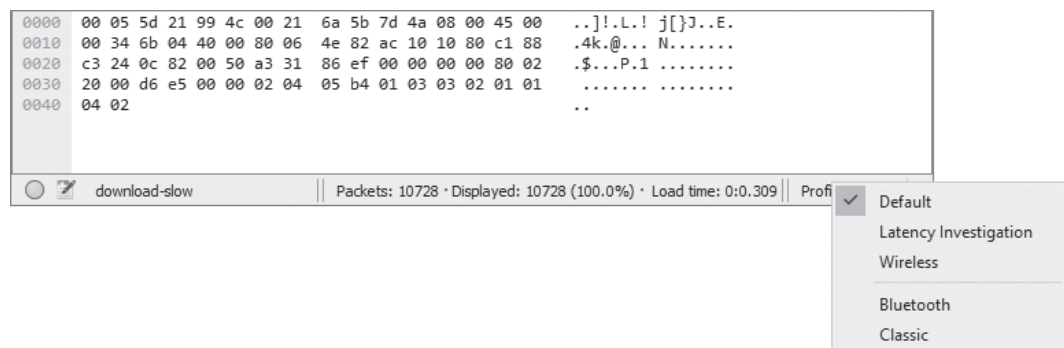


Figure 3-12: Quickly switch between profiles through the Profile heading.

One of the most useful aspects of configuration profiles is that each profile is stored in its own directory with a series of configuration files. This means that you can back up your profiles and share them with others. The folders tab shown in Figure 3-10 provides paths to personal and global configuration file directories. To share a profile with a user on another computer, just copy the folder matching the name of the profile you want to share and paste it into the same directory for the appropriate user on another computer.

While reading along in this book, you may find the need to create a few high-level profiles for general troubleshooting, finding the source of network latency, and investigating security issues. Don't be afraid to use profiles liberally. They are real time-savers when you want to quickly switch a few preference options on or off. I've known people who have used dozens of profiles to address different scenarios with great success.

Now that you have Wireshark up and running, you're ready to do some packet analysis. The next chapter describes how you can work with the packets you've captured.