



Tactical Wireshark

A Deep Dive into Intrusion Analysis,
Malware Incidents, and Extraction of
Forensic Evidence

Kevin Cardwell

Apress®

Tactical Wireshark

A Deep Dive into Intrusion Analysis,
Malware Incidents, and Extraction
of Forensic Evidence

Kevin Cardwell

Apress®

Tactical Wireshark: A Deep Dive into Intrusion Analysis, Malware Incidents, and Extraction of Forensic Evidence

Kevin Cardwell
California, CA, USA

ISBN-13 (pbk): 978-1-4842-9290-7
<https://doi.org/10.1007/978-1-4842-9291-4>

ISBN-13 (electronic): 978-1-4842-9291-4

Copyright © 2023 by Kevin Cardwell

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Aditee Mirashi
Development Editor: James Markham
Coordinating Editor: Mark Powers

Cover designed by eStudioCalamar

Cover image by Luemen Rutkowski on Unsplash (www.unsplash.com)

Distributed to the book trade worldwide by Apress Media, LLC, 1 New York Plaza, New York, NY 10004, U.S.A. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Printed on acid-free paper

This book is dedicated to all of the students I have trained for more than 35 years. The joy of these classes where you learn something every class has made for an incredible cybersecurity adventure, and I thank them for this.

Table of Contents

About the Author	xi
About the Technical Reviewer	xiii
Introduction	xv
Chapter 1: Customization of the Wireshark Interface	1
Configuring Wireshark	2
Column Customization.....	5
Malware	17
Summary.....	25
Chapter 2: Capturing Network Traffic	27
Capturing Network Traffic	27
Prerequisites for Capturing Live Network Data.....	28
Normal Mode	30
Promiscuous Mode.....	31
Wireless.....	33
Working with Network Interfaces	35
Exploring the Network Capture Options.....	36
Filtering While Capturing.....	44
Summary.....	51
Chapter 3: Interpreting Network Protocols	53
Investigating IP, the Workhorse of the Network	53
Analyzing ICMP and UDP.....	63
ICMP	63
UDP.....	70

TABLE OF CONTENTS

- Dissection of TCP Traffic 72
 - Transport Layer Security (TLS) 80
- Reassembly of Packets 86
- Interpreting Name Resolution 89
 - DNS 89
 - Windows Name Resolution 91
- Summary 94
- Chapter 4: Analysis of Network Attacks 95**
 - Introducing a Hacking Methodology 95
 - Planning 96
 - Non-intrusive Target Search 96
 - Intrusive Target Search 100
 - Examination of Reconnaissance Network Traffic Artifacts 112
 - Leveraging the Statistical Properties of the Capture File 114
 - Identifying SMB-Based Attacks 118
 - Uncovering HTTP/HTTPS-Based Attack Traffic 127
 - XSS 127
 - SQL Injection 130
 - HTTPS 136
 - Set the Environment Variable 138
 - Configure Wireshark 139
 - Summary 141
- Chapter 5: Effective Network Traffic Filtering 143**
 - Identifying Filter Components 143
 - Investigating the Conversations 148
 - Extracting the Packet Data 155
 - Building Filter Expressions 159
 - Decrypting HTTPS Traffic 168
 - Kerberos Authentication 176
 - Summary 182

Chapter 6: Advanced Features of Wireshark	183
Working with Cryptographic Information in a Packet	183
Exploring the Protocol Dissectors of Wireshark	188
Viewing Logged Anomalies in Wireshark	192
Capturing Traffic from Remote Computers.....	197
Command-Line Tool TShark	203
Creating Firewall ACL Rules	208
Summary.....	219
Chapter 7: Scripting and Interacting with Wireshark.....	221
Lua Scripting.....	221
Interacting with Pandas	232
Leveraging PyShark	243
Summary.....	254
Chapter 8: Basic Malware Traffic Analysis	255
Customization of the Interface for Malware Analysis.....	255
Extracting the Files	264
Recognizing URL/Domains of an Infected Site.....	275
Determining the Connections As Part of the Infected Machine.....	281
Scavenging the Infected Machine Meta Data	285
Exporting the Data Objects	289
Summary.....	290
Chapter 9: Analyzing Encoding, Obfuscated, and ICS Malware Traffic.....	291
Encoding	291
Investigation of NJRat.....	298
Analysis of WannaCry.....	302
Exploring CryptoLocker and CryptoWall.....	312
Dissecting TRITON.....	313
Examining Trickbot.....	314
Understanding Exploit Kits.....	317

TABLE OF CONTENTS

Establish Contact.....	317
Redirect.....	318
Exploit.....	318
Infect.....	318
Summary.....	322
Chapter 10: Dynamic Malware Network Activities.....	323
Dynamic Analysis and the File System.....	323
Setting Up Network and Service Simulation.....	332
Monitoring Malware Communications and Connections at Runtime and Beyond.....	337
Detecting Network Evasion Attempts.....	350
Investigating Cobalt Strike Beacons.....	355
Exploring C2 Backdoor Methods.....	360
Identifying Domain Generation Algorithms.....	363
Summary.....	367
Chapter 11: Extractions of Forensics Data with Wireshark.....	369
Interception of Telephony Data.....	373
Discovering DOS/DDoS.....	381
Analysis of HTTP/HTTPS Tunneling over DNS.....	392
Carving Files from Network Data.....	397
Summary.....	400
Chapter 12: Network Traffic Forensics.....	401
Chain of Custody.....	401
Isolation of Conversations.....	404
Detection of Spoofing, Port Scanning, and SSH Attacks.....	408
Spoofing.....	409
Port Scanning.....	414
SSH.....	417
Reconstruction of Timeline Network Attack Data.....	422
Extracting Compromise Data.....	425
Summary.....	431

Chapter 13: Conclusion.....	433
Intrusion Analysis.....	433
Malware Analysis.....	437
Forensics.....	440
Summary.....	444
Index.....	445

About the Author



Kevin Cardwell is an instructor, curriculum developer, and technical editor and author of computer forensics and hacking courses. He is the author of the EC Council Certified Penetration Testing Professional, Ethical Hacking Core Skills, Advanced Penetration Testing, and ICS/SCADA Security courses. He has presented at the Black Hat USA, Hacker Halted, ISSA, and TakeDownCon conferences as well as many others. He has chaired the Cybercrime and Cyberdefense Summit in Oman and was Executive Chairman of the Oil and Gas Cyberdefense Summit. He is the author of *Defense and Deception: Confuse and Frustrate the Hackers*, *Building Virtual Pentesting Labs for Advanced*

Penetration Testing, 1st and 2nd editions, and *Backtrack: Testing Wireless Network Security*. He holds a BS in Computer Science from National University in California and an MS in Software Engineering from Southern Methodist University (SMU) in Texas.

About the Technical Reviewer



Shyam Sundar Ramaswami is a Senior Staff Cyber Security Architect at GE Healthcare, and his areas of work include security research, healthcare forensics, offensive security, and defensive security for health-care products. Shyam is a two-time TEDx speaker, co-author of the book titled *It's Your Digital Life*, and a teacher of cybersecurity. Shyam has delivered talks in top-notch international cybersecurity conferences like Black Hat, Qubit, Nullcon, Deepsec, and Hack fest. Shyam has delivered 100+ bootcamps on malware and memory forensics across the globe. Shyam runs a mentoring program called “Being Robin” where he mentors students all over the globe on cybersecurity. Interviews with him have been published on leading websites like ZDNet and CISO MAG.

Introduction

I wrote this book so that people who want to leverage the fantastic capabilities of Wireshark have a reference where you get the “hands-on” tactical concepts that are not covered in most publications about Wireshark. I wrote this from an analysis perspective based on more than 30 years of being an analyst, training analysts and leading analysis teams across the globe. Within this book, you will find the tips and techniques that I have mastered and refined over those years of extensive analysis. For the most part, the process has not changed, but the methods and sophistication of the attackers and criminals have, and this is why we have to continue to enhance and hone our skills.

As the title suggests, this book is broken down into three main parts:

- Intrusion Analysis
- Malware Analysis
- Forensics Analysis

The book does not go deep into topics or concepts that are not part of what we use from a tactical standpoint of Wireshark. There are plenty of references that are available for this. Wherever possible, we do explain some areas outside of Wireshark, and this is most evident when we talk about memory and how malware uses system calls for connections. We start off with a review of what an actual intrusion looks like, and then we introduce a methodology. This is a common theme of the book; we present methodologies that are proven when it comes to performing a systematic analysis process. Each of the areas can be taken on its own, so if you just want to focus on malware, then you can read that section.

CHAPTER 1

Customization of the Wireshark Interface

While it might not seem like a big deal, the fact is the customization of the interface is very important in the creation of an effective analysis plan. The Wireshark interface by default will display the following columns of information:

- **Nos.** – For the number identification of the packet within the display window.
- **Time** – The time the packet was captured; this is one of the columns we will want to perform some changes to.
- **Source** – The source of the generated packet; this can be in the form of a layer two MAC address or a layer three IP address.
- **Destination** – The destination of the generated packet; this too can be in the form of a layer two MAC address or a layer three IP address.
- **Protocol** – The protocol that the Wireshark tool has determined is in the packet.
- **Length** – The length of the data that is contained within the packet.
- **Info** – Where additional information can be displayed about the packet that has been captured.

In this chapter, we will review different methods of how to customize the columns of Wireshark to assist our analysis with special tasks. We will review a customization that can be used to assist with malware analysis.

Configuring Wireshark

An example of the default Wireshark display configuration is shown in Figure 1-1.

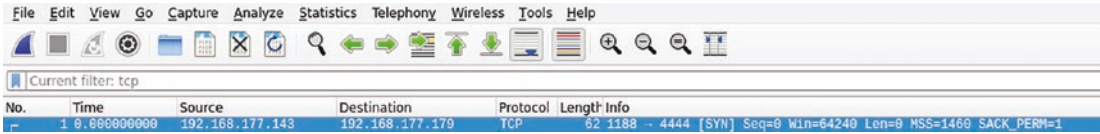


Figure 1-1. *The Wireshark default display configuration*

The figure reflects the default columns and the information that is reflected. As a reference, the Protocol is Modbus.

If you are not familiar with the Modbus protocol, it was originally created by the company Modicon in 1979. They published the protocol as a method of communication with their Programmable Logic Controllers or PLC. Modbus has become a popular communication protocol and is now a commonly available means of connecting industrial electronic devices. Modbus is popular in industrial environments because it is openly published and royalty-free. The company Modicon is known as Schneider Electric today. As you continue to review the packet capture, you can see in the “Info” section additional information about the captured packet. As the information indicates, the packet capture is that of a Transaction Query, the number of the Query is 209, the Unit is 1, and the Query is of type 3, which means it is a reading of the Holding Registers.

We will not cover any more details here of this packet that has been captured; however, as the book progresses, you will get much more data on this and many other types of protocols.

As we stated at the beginning of this chapter, we want the Wireshark interface to be configured so we can get the best results when we process our data capture files, and while the default settings are okay, they are not providing us the best opportunity to get the most from the Wireshark tool.

The first thing we want to do is to clean up the current columns on the Wireshark tool. When we start thinking about the process and concept for analysis, we need to have the port information of our communications, and with the current settings, we do not have this. We can look for it, but it is much more efficient to have the port information easily at our disposal. When you think of a port, a good analogy is that of a door, so when we have a port open on a machine, it is equivalent to an open door, and since it is open, then there can be connections to it. This is what we want to focus

on when we are reviewing a capture file, because everything starts with a connection. Once the connection is made, then the data will flow, especially when we discuss the communication protocol Transmission Control Protocol (TCP) later in the book.

So now that we have a little bit of an idea on the ports and the concept of connections, let's see how to make the customizations and changes.

The main Wireshark settings when it comes to the display options are accessed via the main top bar menu; we access the Preferences settings by clicking on **Edit ► Preferences**. An example of this is shown in Figure 1-2.

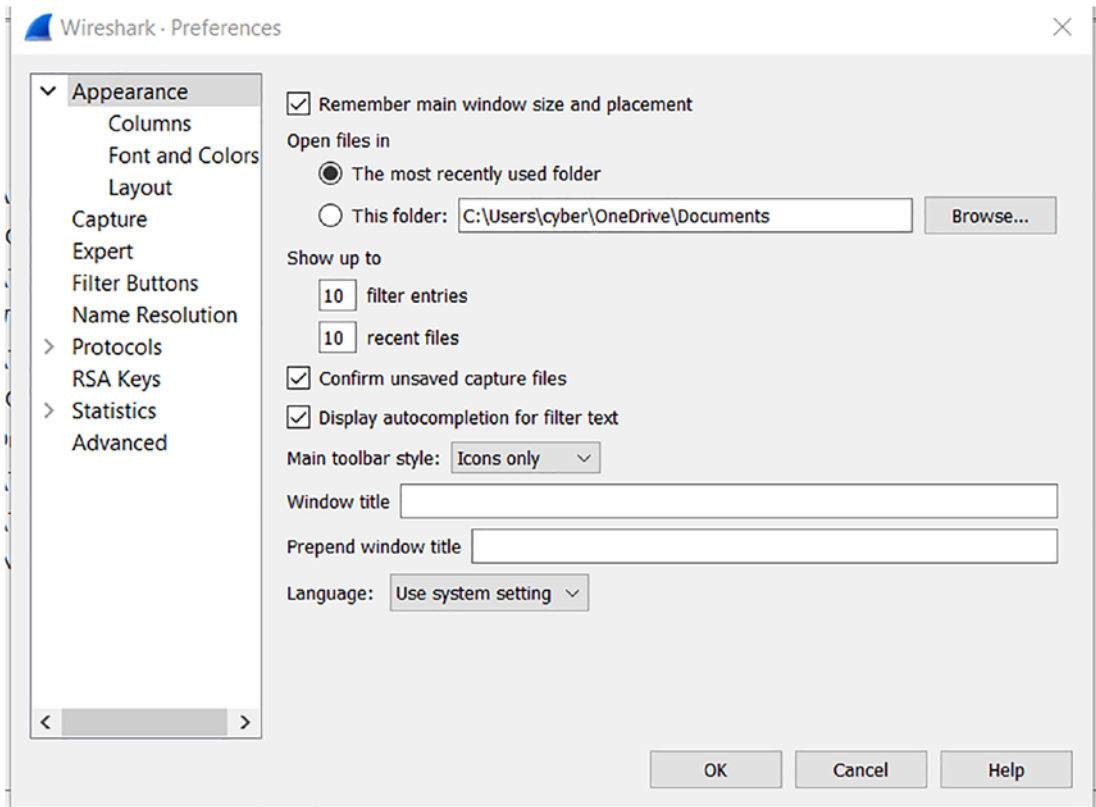


Figure 1-2. *The Wireshark Preferences settings*

As shown in the image, we do have a variety of settings that we can select to change the way our captured data is displayed. Having said that, for our purposes here, we will just focus on the UTC settings, which is our representation of the GMT zone. Since we have more than one setting available, we will use the **UTC Time of Day**. Additionally, we will change the setting from Automatic to **Seconds**. An example of the format changes is shown in Figure 1-3.

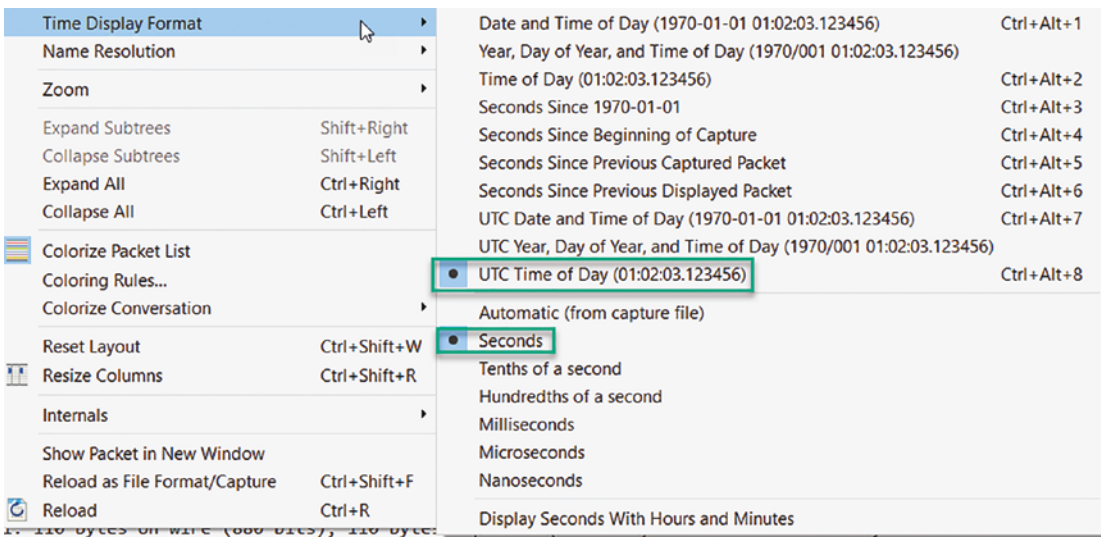
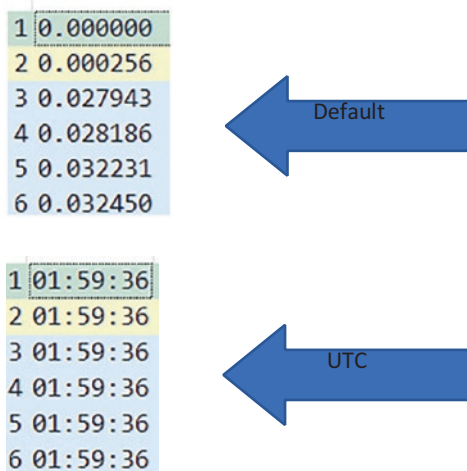


Figure 1-3. Time format changes

Now that we have made the settings changes, we can refer to what the capture file looks like. An example of the time field before the settings and one with the settings is shown next.



For most people, including your author, it is preferred to have the normal time format and not the default selection of number of seconds ticked off when captured.

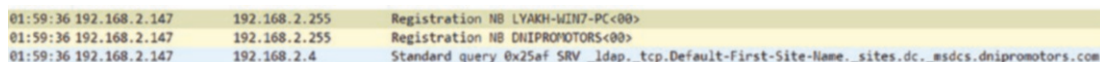
Column Customization

We next want to review and make some changes to our columns; this will assist us when we are performing different types of capture file analysis tasks. We return to our Columns settings located in the Preferences menu and review the columns that are displayed by default. It is true that the columns that are displayed are a matter of personal preference; however, there are some that are displayed that are in many cases rarely referenced. Since our User Interface does have some limitations, we want to get the most from our displayed data. The columns that we can delete for our first analysis profile are the following:

1. No
2. Length

These columns are not commonly used, so it is a good idea to remove them. Another column that you might want to remove when doing malware analysis is the Protocol, while it is good to see the protocol, we can determine this by more than one method, so it is a matter of personal preference if we leave this displayed.

Once we have removed these columns, our Wireshark User Interface will reflect that shown in Figure 1-4.



01:59:36	192.168.2.147	192.168.2.255	Registration NB LYAKH-WIN7-PC<00>
01:59:36	192.168.2.147	192.168.2.255	Registration NB DNIIPROMOTORS<00>
01:59:36	192.168.2.147	192.168.2.4	Standard query 0x25af SRV _ldap._tcp.Default-First-Site-Name._sites.dc._msdcs.dnipromotors.com

Figure 1-4. Custom columns

As reflected in Figure 1-4, we now have a more streamlined display for our interface. We now want to add some additional columns to discover information we commonly use in our analysis.

We add columns via the same menu selections from before and access the settings within the **Edit ► Preferences ► Columns** path. Once we are there, we need to click on the “+” sign to add a new column. An example of this is shown in Figure 1-5.

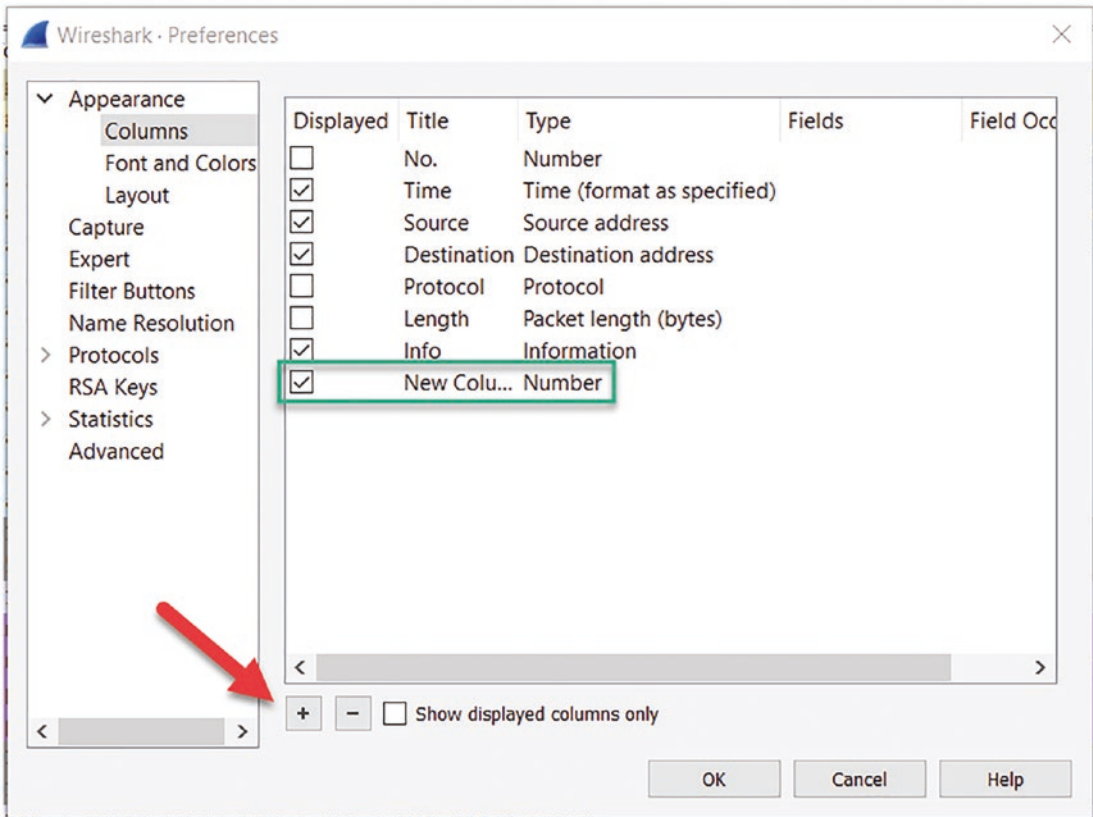


Figure 1-5. Adding columns

Once we have added the new column, we want to customize it, we do this by double-clicking the name, and this will highlight the name in blue so it can be edited directly. For the first custom column, we will use the **Source Port** as the name, so enter this in the Name field. An example of this is shown in Figure 1-6.

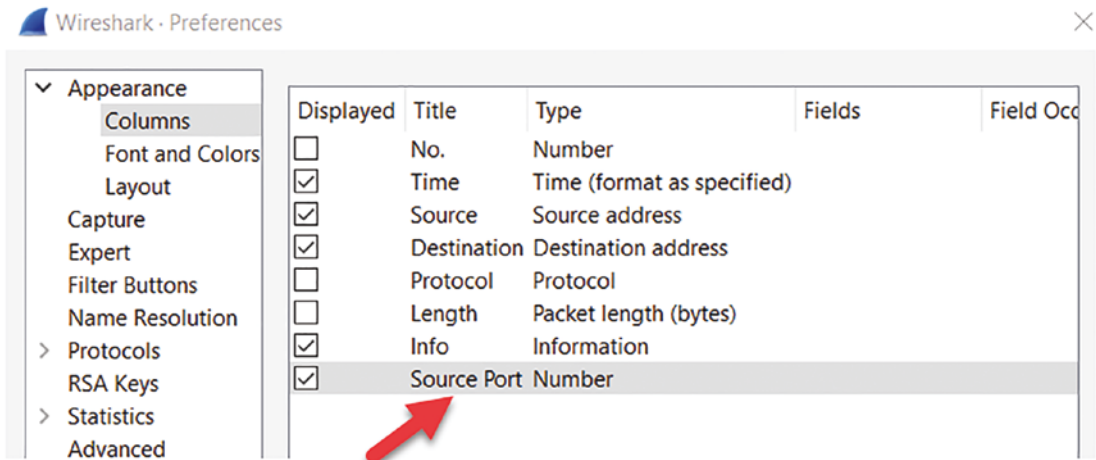


Figure 1-6. *The source port column*

Any time we create a custom setting, it is always good to put as much amplifying information as possible. We do this in the **Type** field. When you double-click on the **Type** field, a listing of the different type options will be displayed; an example of this is shown in Figure 1-7.

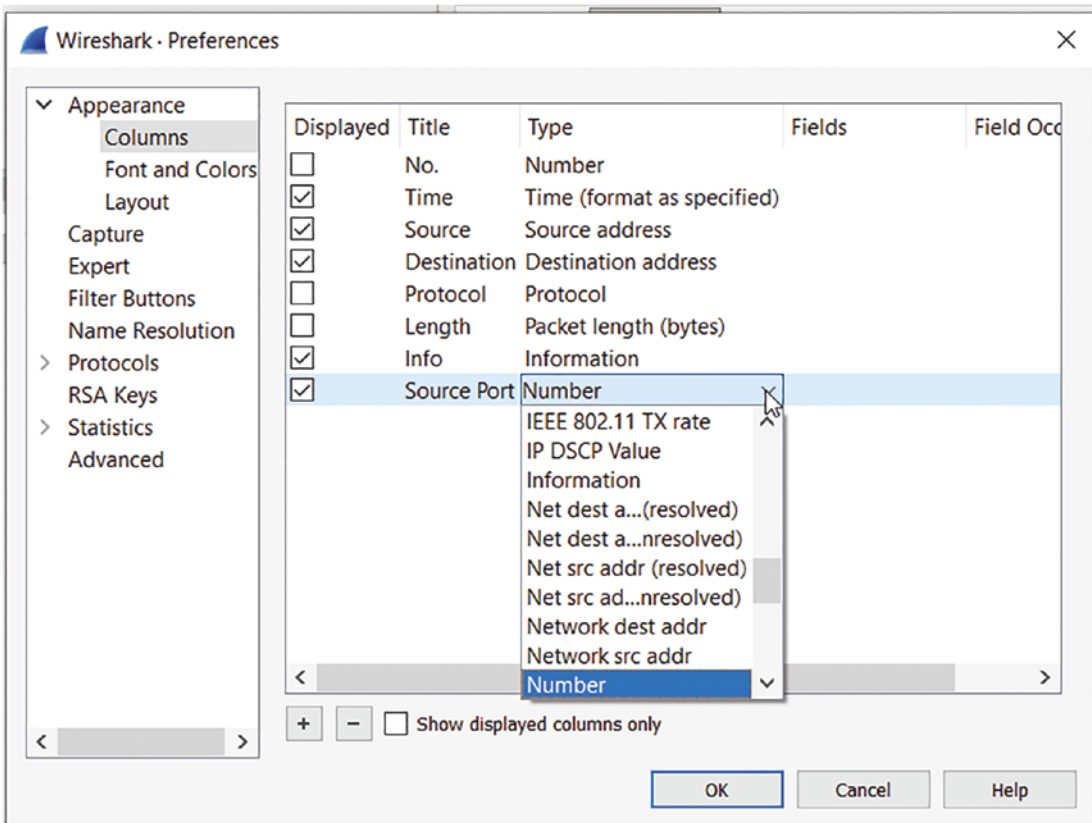


Figure 1-7. Column type options

For our Source Port column, we want to select the Src port (unresolved). An example of this is shown in Figure 1-8.

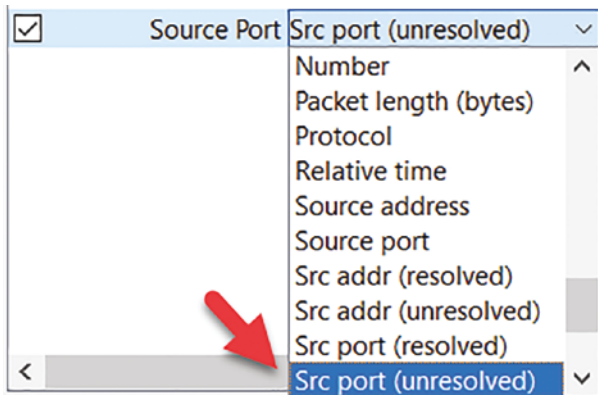


Figure 1-8. Src port unresolved setting

The source port is one of those important items that we want to be able to see in a relatively quick manner. We need this when we are reviewing network communication sequences between machines. As a refresher, network communication is usually from a client to a server; this connection from the client is usually at a port >1023, so by displaying the source port, it allows a quick review of the method of communication that is reflected in the capture file. When we see a port that is <1023 to another port that is <1023, this could be suspicious. We say “could” because unfortunately, over time the normal communications procedures of the network protocols are not as structured as when we started. While it is normally a fact that the client connection comes from a port >1023, it is not always guaranteed. These ports >1023 are referred to as ephemeral ports. This means the ports are considered transitory in nature, because a client should make the connection, receive the required data, and then disconnect, and this is a temporary sequence, hence the name.

The next column we want to add to the display is that of the destination port; the process is the same as before; we click on the “+” and then double-click on the name and enter the name of **Dest Port**. Then as before, we click in the drop-down of the **Type** field and select **Destination Port (unresolved)**. You should now have two custom ports that you have added. Great job! A port is resolved if the tool recognizes the service running on the port. An example of our two ports is shown in Figure 1-9.

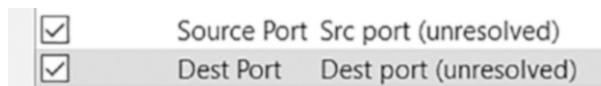


Figure 1-9. *Src and Dest port columns*

We now want to get the display order set with our two new columns. We can achieve this very easily by dragging the columns into the order that we prefer. A good location for the Source Port is right after the Source Address, so we can drag this to that location. Now, we want to do the same for the Destination Port and place it right after the Destination Address. An example of these changes is shown in Figure 1-10.

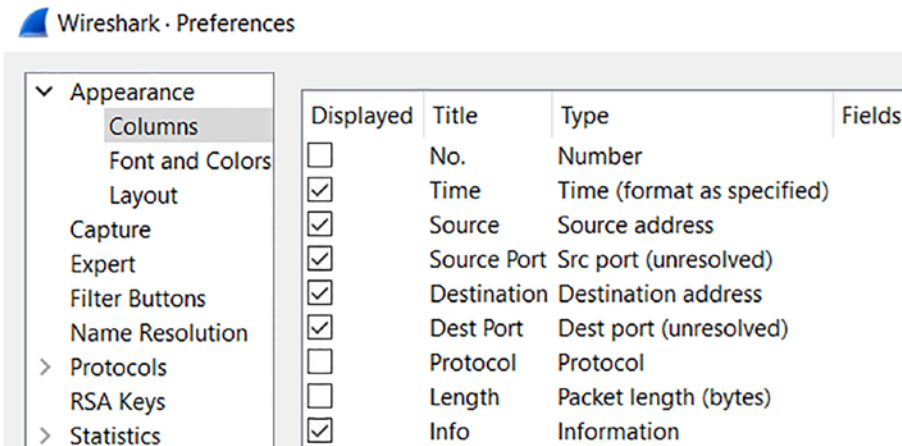


Figure 1-10. Setting the order of the display columns

You might find it a little tricky to get the column to move, so look for the red circle that is displayed to change and you should be able to drop the column there.

After adding the source and destination port columns, click the “**OK**” button to apply the changes. These new columns are automatically aligned to the right, so right-click on each column header to align them to the left so they match the other columns. An example of this is shown in Figure 1-11.

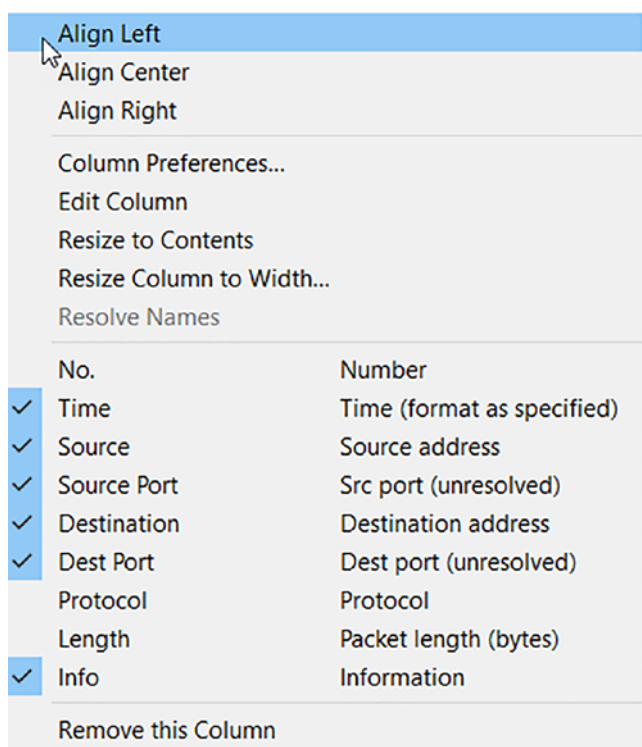


Figure 1-11. The list of selected columns

Once you have finished this, then the display should reflect that as shown in Figure 1-12.

TIME	SOURCE	SOURCE PORT	DESTINATION	INFO
01:59:41	192.168.2.4	123	192.168.2.147	123 RTP Version 3, server
01:59:36	192.168.2.147	49155	192.168.2.4	135 49155 → 135 [SYN] Seq=0 Win=0 Len=0 MSS=5440 MTU=256 SACK_PERM=1
01:59:36	192.168.2.147	49155	192.168.2.4	135 49155 → 135 [ACK] Seq=1 Ack=1 Win=65536 Len=0
01:59:36	192.168.2.147	49155	192.168.2.4	135 Rind: call_15 2, Fragment: Single, 3 context items: IPv4 V3.0 (32bit NDR), IPv4 V3.0 (64bit NDR), IPv4 V3.0 (64bit NDR)
01:59:36	192.168.2.147	49155	192.168.2.4	135 Rtp request, RFC 3551, 30bit NDR

Figure 1-12. Wireshark custom column display

We can now quickly determine the source and destination port. This allows us to identify a potential service that could be targeted. We will look at an example of this now. A common method of attack is to look for a service and then attempt to gain access once a service is discovered that could provide us access, so with our new display that we have just customized, we can see how easy it is to identify when a service is getting either attacked or a lot of attention. The first service we will look at here is the File Transfer Protocol, otherwise known as FTP. Now, many of you reading this might be saying, “FTP. It is old!” While this is true and an argument could be made for this, it is just being used as an example here and in many environments is still used today,

especially in Industrial Control Systems (ICS) enterprise networks. As a refresher, the FTP uses two ports: one for communication and one for data. With our now custom display, we should be able to identify this, which will also allow us to demonstrate the analysis and determination as to the mode of FTP. But before we do this, we need to have a good understanding of FTP. So what exactly is it? A good source and probably one of the best ones is that of the Request for Comments (RFC) that have been released as a recommended standard for FTP. We refer to this as “recommended” because there is no requirement that you have to follow the RFC, and unfortunately, many vendors do not, but that is a topic outside of this book. Now we could refer to the Internet Engineering Task Force at <https://ietf.org>, which is shown in Figure 1-13.

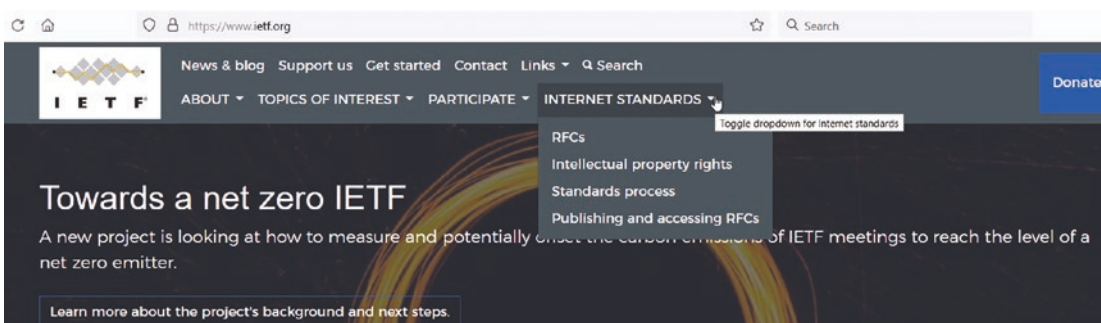


Figure 1-13. Internet Engineering Task Force

As the image shows, we have the Internet Standards menu option, and within this, we have the RFCs. An example of when the menu item is selected is shown in Figure 1-14.

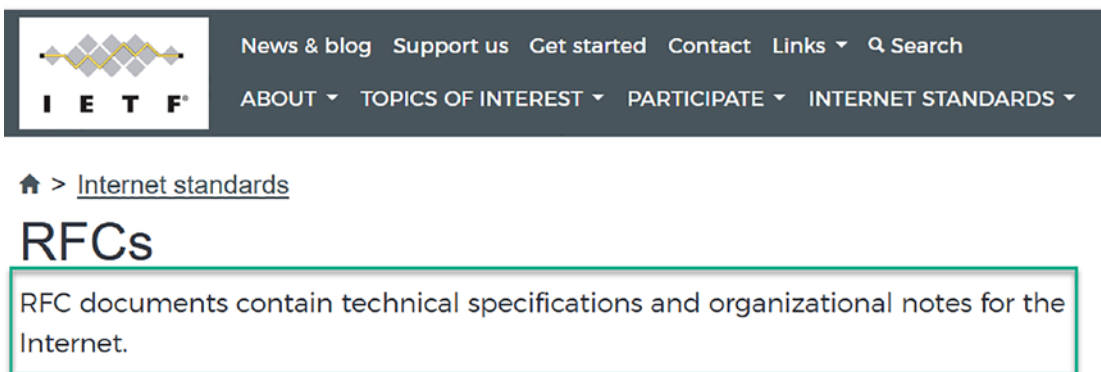


Figure 1-14. Request for Comments

The green box in Figure 1-14 is the main thing about the RFC; these are the notes and specification for the Internet! So we must be familiar with them if we are going to work in IT. These are documents that are in a text format and not the best structure to read, so it does take some time to get used to them. An example of an RFC is shown in Figure 1-15.

→ ↻ 🏠 🔒 https://www.rfc-editor.org/rfc/rfc1918

[RFC Home] [TEXT] [PDF] [HTML] [Tracker] [IPR] [Errata] [Info page]

Updated by: [6761](#)
 Network Working Group
 Request for Comments: 1918
 Obsoletes: [1627](#), [1597](#)
 BCP: 5
 Category: Best Current Practice

BEST CURRENT PRACTICE
Errata Exist
 Y. Rekhter
 Cisco Systems
 B. Moskowitz
 Chrysler Corp.
 D. Karrenberg
 RIPE NCC
 G. J. de Groot
 RIPE NCC
 E. Lear
 Silicon Graphics, Inc.
 February 1996

Address Allocation for Private Internets

Status of this Memo

This document specifies an Internet Best Current Practices for the Internet Community, and requests discussion and suggestions for improvements. Distribution of this memo is unlimited.

1. Introduction

For the purposes of this document, an enterprise is an entity autonomously operating a network using TCP/IP and in particular determining the addressing plan and address assignments within that network.

Figure 1-15. Example of an RFC

Figure 1-15 reflects the RFC 1918, which is the standards document that identifies the private addressing for IP addresses that should not be routed. These are the following addresses:

1. 10.0.0.0 (10/8)
2. 172.16-172.31 (172.16/12)
3. 192.168 (192.168/16)

We will refer to the first block as “24-bit block”, the second as “20-bit block”, and to the third as “16-bit” block. Note that (in pre-CIDR notation) the first block is nothing but a single class A network number, while the second block is a set of 16 contiguous class B network numbers, and third block is a set of 256 contiguous class C network numbers.

—RFC 1918

The power of the RFC is anytime someone wants to research or understand a communication protocol, the first reference is that of the RFC. Having said that, for some, they can be a challenge to read, so there are Internet sites that can assist with that. Even the IETF has a set of tools that can assist us with the interpretation of an RFC; the site can be found at <https://tools.ietf.org>. An example of this is shown in Figure 1-16.

IETF Tools

*IETF-related tools, standalone or hosted on tools.ietf.org.
(Tools hosted by the secretariat are listed at <http://www.ietf.org/tools>).*

Which license? See [Preferred License](#)



Prepare documents

[RFC dependency checker](#)

Joe Touch

A [script](#) to check the references in Internet Drafts for dependencies and updates.

[Bibtex Citation Converter](#)

Yaron Sheffer

This tool converts bibtex-formatted citations into the bibxml format used in xml2rfc. Many (if not most) academic papers have bibtex citations available online, and the tool makes it easier to reference them in Internet Drafts.

[Templates for xml2rfc work](#)

Elwyn Davies

Elwyn Davies has produced a template as a starting point for writing drafts using xml2rfc. You can find a copy of the schema v3 version of the [XML template at tools.ietf.org](#).

[Draft Submission API](#)

Henrik Levkowitz

A simplified draft submission interface, intended for automation, is available at <https://datatracker.ietf.org/api/submit>. The interface accepts only xml uploads which can be processed on the server, and requires the user to have a datatracker account. A successful submit still requires the same email confirmation roundtrip as submissions done through the regular submission tool.

[BibXML to Markdown Converter](#)

Yaron Sheffer

This simple script, bibxml2md, converts bibxml references extracted from xml2rfc files into markdown, for use in kramdown-rfc2629 Internet Drafts.



Search, show and print documents

[Download the latest documents](#)

Rsync access to various document archives:

- Unpurged IETF drafts repository:

To list the content, do:

```
rsync rsync.tools.ietf.org::tools.id
```

To sync the content, do:

```
rsync -avz rsync.tools.ietf.org::tools.id ./id
```

- Currently available htmlized drafts and RFCs:

To list the content, do:

```
rsync rsync.tools.ietf.org::tools.html
```

To sync the content, do:

```
rsync -avz rsync.tools.ietf.org::tools.html ./html
```

- For a full list of the various rsync sources at tools.ietf.org, do:

```
rsync rsync.tools.ietf.org::
```

[Access IETF-related files from the command line](#)

Paul Hoffman

The "ietf" program lets you access IETF-related files from the command line. It creates a local copy of these files on your computer using rsync, and gives a friendly way to access them. You can give commands from your normal shell, or you can run an interactive shell that is part of the program.

[Chrome: Rewrite IETF ID URLs to the Tools or Datatracker versions](#)

Warren Kumar

This will rewrite the "official" IETF Internet Draft URLs (<https://www.ietf.org/id/foo-42.txt>) to the Tools (<https://tools.ietf.org/html/foo-42>) or Datatracker (<https://datatracker.ietf.org/docs/foo>) versions instead.

Figure 1-16. *The IETF tools*

We will take a brief moment to explain some of the components of an RFC. There should be a header related to the RFC; an example of this is shown in Figure 1-17.

```
Internet Engineering Task Force (IETF)                                R. Fielding, Ed.
Request for Comments: 7230                                           Adobe
Obsoletes: 2145, 2616                                               J. Reschke, Ed.
Updates: 2817, 2818                                                greenbytes
Category: Standards Track                                           June 2014
ISSN: 2070-1721
```

Figure 1-17. *RFC header*

At the top left, this header states “Internet Engineering Task Force (IETF)”. That indicates that this is a product of the IETF; although it’s not widely known, there are other ways to publish an RFC that don’t require IETF consensus; for example, the Independent Submission Stream allows RFC publication for some documents that are

outside the official IETF/IAB/IRTF process but are relevant to the Internet community and achieve reasonable levels of technical and editorial quality.

Now that we have an understanding of protocols that we can research. We have a better way that we can research this information as we are conducting our analysis.

We will now revisit our FTP; furthermore, as has been stated in this chapter, the port number is an important component for doing our analysis. The FTP has two main ports that are used; the first is that of the Control and Communication, and this port is assigned to port 21. The FTP is defined in RFC 959; an example of the RFC is shown in Figure 1-18.

Network Working Group
Request for Comments: 959

Obsoletes RFC: 765 (IEN 149)

J. Postel
J. Reynolds
ISI
October 1985

FILE TRANSFER PROTOCOL (FTP)

This RFC (converted to hypertext in 1994 by Tim BL) consists of the following sections:

- [Status of this memo](#)
- [Introduction](#)
- [Overview](#)
- [Data Transfer Functions](#) (about modes)
- [File Transfer Functions](#) (actual commands)
- [Declarative Specifications](#)
- [State Diagrams](#)
- [A Typical FTP Scenario](#)
- [Connection Establishment](#)
- [Appendix 1: Page Structure](#)
- [Appendix 2: Directory Commands](#)
- [Appendix 3: RFCs on FTP](#)
- [References](#)

Figure 1-18. FTP RFC

As the figure shows, the FTP RFC has a date of 1985, so this does verify that it is an older protocol. The section we want to review here is the Data Transfer Functions, because it states that it defines the modes. Once you select this, you will see the

additional information on how the FTP works. This is beyond the scope here, but you do have the information if you want to pursue the topic further.

In addition to port 21, we also have a data port used with FTP. That port is traditionally 20 for active FTP and >1023 selectable for passive FTP. Again, these are things that as analysts you need to be aware of when you are reviewing a capture file. In fact, an understanding of the challenges with respect to filtering of passive vs. active FTP is an important concept as well. A synopsis of this is as follows:

- **Active Mode** – The client issues a PORT command to the server signaling that the client will “actively” provide an IP and port number to open the Data Connection back to the client.
- **Passive Mode** – The client issues a PASV command to indicate that the client will wait “passively” for the server to supply an IP and port number, after which the client will create a Data Connection to the server.

As you can see, this or any other protocol for that matter takes time to understand, and it is worth investing that time so you can better perform your analysis.

Malware

When we investigate malware, the Wireshark columns that are displayed by default are not the best to use when it comes to our task of malware analysis, so thus far, we have customized some of the columns so they can provide us with a more efficient analysis capability. Now that we have done this, we need to add additional columns to assist us with our analysis tasks. It is important to understand that we can and often will customize our user interface in different ways to assist us with our analysis of capture files. We will now look specifically at an example of this for when we configure our user interface to maximize our efficiency for malware analysis.

When we customize our interface, we want to plan for this and focus on what exactly are the characteristics that we are wanting to review. With our example of malware, one of the main things we want to track for our analysis is the web traffic and communication sequences. This is because malware often involves web traffic. This is due to the desire to “blend” into the network communication traffic and appear to be normal traffic on the network. We can also see the communication channel for command and control (C2) that is many times disguised in web traffic. Wireshark’s default column configuration is

not ideal when investigating such malware-based infection traffic. However, Wireshark can be customized to provide a better view of the activity.

Earlier we customized the time reference, and we customized our interface in such a way that it is more streamlined and can assist us with being more efficient with our analysis and that is the goal.

Currently, we have the following columns we have customized for our interface:

1. Time (UTC)
2. Source IP address
3. Source port
4. Destination IP address
5. Destination port
6. Info

This is a good start, and you can use it as a foundation for the different types of analysis tasks you will perform. For our malware analysis, we want to add additional information by adding more columns; an example of the additional columns is shown here:

1. HTTP host
2. HTTPS server

Wireshark allows us to add custom columns based on almost any value found in the frame details window. This is how we add domain names used in HTTP and HTTPS traffic to our Wireshark column display. We can quickly identify the domains in a capture file by entering a filter. For our example here, we want to set the filter on `http.request`. An example of this is shown in Figure 1-19.

The screenshot shows the Wireshark packet list pane with the filter `http.request` applied. The table below represents the data shown in the interface:

Time	Source	Source	Destination	Dest Port	Host	Info
01:59:42	192.168.2.147	491...	23.211.124....	80	True	GET /ncsi.txt HTTP/1.1
02:01:37	192.168.2.147	575...	239.255.255...	1900	True	M-SEARCH * HTTP/1.1
02:01:37	192.168.2.147	575...	239.255.255...	1900	True	M-SEARCH * HTTP/1.1
02:01:40	192.168.2.147	575...	239.255.255...	1900	True	M-SEARCH * HTTP/1.1
02:01:40	192.168.2.147	575...	239.255.255...	1900	True	M-SEARCH * HTTP/1.1
02:01:43	192.168.2.147	575...	239.255.255...	1900	True	M-SEARCH * HTTP/1.1
02:01:43	192.168.2.147	575...	239.255.255...	1900	True	M-SEARCH * HTTP/1.1
02:02:13	192.168.2.147	492...	198.54.126....	80	True	GET /hojuks/vez.exe HTTP/1.1

Figure 1-19. The `http.request` filter