Community Experience Distilled

Wireshark Network Security

A succinct guide to securely administer your network using Wireshark

Piyush Verma

[PACKT] open source* PUBLISHING community experience distilled

Table of Contents

Wireshark Network Security

Wireshark Network Security

First published: July 2015

Production reference: 1240715

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham B3 2PB, UK.

ISBN 978-1-78439-333-5

www.packtpub.com

Credits

Author

Piyush Verma

Reviewers

David Guillen Fandos

Mikael Kanstrup

Jaap Keuter

Tigran Mkrtchyan

Commissioning Editor

Amarabha Banerjee

Acquisition Editor

Larissa Pinto

Content Development Editor

Siddhesh Salvi

Technical Editor

Madhunikita Sunil Chindarkar

Copy Editor

Dipti Mankame

Project Coordinator

Nidhi Joshi

Proofreader

Safis Editing

Indexer

Priya Sane

Production Coordinator

Shantanu N. Zagade

Cover Work

Shantanu N. Zagade

About the Author

Piyush Verma currently serves as a senior security analyst at NII Consulting, India, and enjoys hacking his way into organizations (legally) and fixing the vulnerabilities encountered. He strongly values hands-on experience over certifications; however, here are a few certifications he has earned so far: OSCP, CEH, CHFI, CCNA Security, and CompTIA Security+. He is a highly sought-after professional speaker and has delivered security training to folks working in public, private, and "secret" sectors. He can be contacted at https://in.linkedin.com/in/infosecpiyushverma.

Acknowledgment

G.B. Stern quoted: "Silent gratitude isn't much use to anyone."

First and foremost, my deepest gratitude goes to my family, for being the perfect mix of love and chaos. My father, for his guidance and faith in my decisions; my mother, for her unconditional love and the awesome delicacies I much relish; and my sisters, for their love and support.

Thanks to these influential personalities in my journey so far: Mr. Dheeraj Katarya, my mentor, for all that you've taught me, which goes beyond the technical lessons; Mr. Sanjay Sharma, who is always a big motivator; Mr. Rahul Kokcha, for making the most difficult concepts easy to comprehend; Mr. Santosh Kumar, for his expert insights on Wireshark; Mr. K.K. Mookhey, for whom nothing is unachievable and he strives even bigger; Mr. Jaideep Patil, who is lavish in his praise and hearty in his approbation.

It has indeed been a pleasure to work with some of the great minds of the industry. Thanks t Mr. Wasim Halani, who has an answer for everything relevant and is rightly called the "Google" of our organization; Mr. Vikash Tiwary, for whom nothing matches his enthusiasm and the depth of knowledge he possesses. Special thanks to Saman, Parag, and Avinash for their feedback.

I'd also like to thank my friends, who made the most difficult times fun and fun times the mos memorable.

Also, this book would have been difficult to achieve without the fantastic editorial team at Packt Publishing and the prodigious reviewers who helped bring out the best in me.

Ultimately, as the genius Albert Einstein quoted:

"I am thankful to all those who said *no*. It's because of them I did it myself."

About the Reviewers

David Guillen Fandos is a young Spanish engineer who enjoys being surrounded by computers and anything related to them. He pursued both his degrees, an MSc in computer science and an MSc in telecommunications, in Barcelona and has worked in the microelectronics industry since then.

He enjoys playing around in almost any field, including network security, software and hardware reverse engineering, and anything that could be considered security. Despite his age, David enjoys not-so-new technologies and finds himself working with compilers and assemblers. In addition to networking, he enjoys creating hacking tools to exploit various types of attacks.

David is now working at ARM after spending almost 2 years at Intel, where he does some hardware-related work in the field of microprocessors.

I'd like to thank those people in my life who continuously challenge me to do new things, do things better than we do, or just change the way we look at life—especially those who believe in what they do and who never surrender no matter how hard it gets.

Mikael Kanstrup is a software engineer with a passion for adventure and the thrills in life. In his spare time, he likes kitesurfing, riding motocross, or just being outdoors with his family an two kids. Mikael has a BSc degree in computer science and years of experience in embedde software development and computer networking. For the past decade, he has been working as a professional software developer in the mobile phone industry.

Jaap Keuter has been working as a development engineer in the telecommunications indust for telephony to Carrier Ethernet equipment manufacturers for the past 2 decades. He has been a Wireshark user since 2002 and a core developer since 2005. He has worked on various internal and telephony-related features of Wireshark as well as custom-made protoc dissectors, fixing bugs and writing documentation.

Tigran Mkrtchyan studied physics at Yerevan State University, Armenia, and started his IT career as an X25 network administrator in 1995. Since 1998, he has worked at Deutsches Elektronen-Synchrotron (DESY)—an international scientific laboratory, located in Hamburg, Germany. In November 2000, he joined the dCache project, where he leads the developmen of the open source distributed storage system, which is used around the world to store and process hundreds of petabytes of data produced by the Large Hadron Collider at CERN. Since 2006, Tigran has been involved in IETF, where he takes an active part in NFSv4.1 protocol definition, implementation, and testing. He has contributed to many open source projects, such as the Linux kernel, GlassFish application server, Wireshark network packet analyzer, ownCloud, and others.

DESY is a national research center in Germany that operates particle accelerators used to investigate the structure of matter. DESY is a member of the Helmholtz Association and operates at sites in Hamburg and Zeuthen.

DESY is involved in the International Linear Collider (ILC) project. This project consists of a 30-km-long linear accelerator. An international consortium decided to build it with the technology developed at DESY. There has been no final decision on where to build the accelerator, but Japan is the most likely candidate.

Support files, eBooks, discount offers, and more

For support files and downloads related to your book, please visit www.PacktPub.com.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at <service@packtpub.com> for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.

https://www2.packtpub.com/books/subscription/packtlib

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can search, access, and read Packt's entire library of books.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

Free access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktL today and view 9 entirely free books. Simply use your login credentials for immediate access

Preface

Wireshark is the tool of choice for network administration and troubleshooting, but its scalability goes beyond that. It is an excellent aid in performing an in-depth analysis of issues pertaining to the overall security of the network. Several tools and devices are available in the market to detect network-related attacks and take appropriate actions based on a predefined set of rules. However, at a very granular level, it all boils down to frames, or sometimes interchangeably called as packets, and the data they carry.

This book is written from the standpoint of using Wireshark to detect security-concerning flaws in commonly used network protocols and analyze the attacks from popular tools such as Nmap, Nessus, Ettercap, Metasploit, THC Hydra, and Sqlmap. In the later part of the book, we will dive into inspecting malware traffic from an exploit kit and IRC botnet and solve real-world Capture-The-Flag (CTF) challenges using Wireshark, basic Python code, and tools that complement Wireshark.

What this book covers

Chapter 1, *Getting Started with Wireshark – What, Why, and How?*, provides an introduction to sniffing and packet analysis and its purpose. Later, we will look at where Wireshark fits into the picture and how it can be used for packet analysis by performing our first packet capture.

Chapter 2, *Tweaking Wireshark*, discusses the robust features of Wireshark and how they can be useful in terms of network security. We will briefly discuss the different command-line utilities that ship with Wireshark.

Chapter 3, *Analyzing Threats to LAN Security*, dives into performing sniffing and capturing user credentials, analyzing network scanning attempts, and identifying password-cracking activities. In this chapter, we will also learn to use important display filters based on protocols and common attack-tool signatures and also explore regular expression-based filters. Then we will look at tools that complement Wireshark to perform further analysis and finally nail an interesting CTF challenge via the techniques learned in the chapter.

Chapter 4, *Probing E-mail Communications*, focuses on analyzing attacks on protocols used in e-mail communication and solving a couple of real-world e-mail communication challenges using Wireshark.

Chapter 5, *Inspecting Malware Traffic*, starts with creating a new profile under Wireshark for malware analysis and then picks up a capture file from an exploit kit in action and diagnoses with the help of Wireshark. Later, we also give a brief on inspecting IRC-based botnets.

Chapter 6, *Network Performance Analysis*, begins by creating a troubleshooting profile under Wireshark and then discusses and analyzes TCP-based issues and takes up case studies of slow Internet, sluggish downloads, and delves further into picking up on Denial-of-Service attacks using Wireshark.

What you need for this book

To work with this book, you will need to download and install Wireshark on the operating system of your choice, and basic TCP/IP knowledge will be a plus.

Who this book is for

If you are a network administrator or a security analyst with an interest in using Wireshark fo security analysis, this is the book for you. Basic familiarity with common network and application service terms and technologies is assumed; however, expertise in advanced networking topics or protocols is not required.

Conventions

In this book, you will find a number of text styles that distinguish between different kinds of information. Here are some examples of these styles and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "An indicato in that case will be the visibility of popular IRC commands as `USER`, `NICK`, `JOIN`, `MODE`, and `USERHOST`."

Any command-line input or output is written as follows:

```
frame contains "\x50\x4B\x03\x04"
```

New terms and **important words** are shown in bold. Words that you see on the screen, for example, in menus or dialog boxes, appear in the text like this: "To enable or disable the title navigate to **Edit** | **Preferences** | **User Interface** and modify the option **Welcome screen and title bar shows version** to suit your requirement."

> ## Note
>
> Warnings or important notes appear in a box like this.

> ## Tip
>
> Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book–what you liked or disliked. Reader feedback is important for us as it helps us develop titles that you will really get the most out of.

To send us general feedback, simply e-mail <feedback@packtpub.com>, and mention the book's title in the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide at www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you t
get the most from your purchase.

Downloading the color images of this book

We also provide you with a PDF file that has color images of the screenshots/diagrams used
in this book. The color images will help you better understand the changes in the output. You
can download this file from
https://www.packtpub.com/sites/default/files/downloads/3335OS_ColoredImages.pdf.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do
happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—
we would be grateful if you could report this to us. By doing so, you can save other readers
from frustration and help us improve subsequent versions of this book. If you find any errata
please report them by visiting http://www.packtpub.com/submit-errata, selecting your book,
clicking on the **Errata Submission Form** link, and entering the details of your errata. Once
your errata are verified, your submission will be accepted and the errata will be uploaded to
our website or added to any list of existing errata under the Errata section of that title.

To view the previously submitted errata, go to
https://www.packtpub.com/books/content/support and enter the name of the book in the
search field. The required information will appear under the **Errata** section.

Piracy

Piracy of copyrighted material on the Internet is an ongoing problem across all media. At
Packt, we take the protection of our copyright and licenses very seriously. If you come acros
any illegal copies of our works in any form on the Internet, please provide us with the locatio
address or website name immediately so that we can pursue a remedy.

Please contact us at <copyright@packtpub.com> with a link to the suspected pirated material

We appreciate your help in protecting our authors and our ability to bring you valuable
content.

Questions

If you have a problem with any aspect of this book, you can contact us at
<questions@packtpub.com>, and we will do our best to address the problem.

Chapter 1. Getting Started with Wireshark – What, Why, and How?

Sniffing and interpreting traffic on the network has been and always will be an integral part of a network analyst's job profile. It is not only restricted to the network analyst's profession, but it also plays a significant role in the fields of software development, network security, and digital forensics. Wireshark is the tool of choice at most workplaces and does not seem to slow down in terms of popularity and features, hence making it a "must-know" tool. This chapter gives a briefing on:

- Sniffing and its purpose
- Tools of the trade
- Getting up and running with Wireshark

Sniffing

Sniffing, by definition, is using our sense of smell to savor something, like a sniff of perfume. In this case, our nose acts as a sniffer. We can perform sniffing on the network using various tools categorized as packet sniffers to capture or collect the packets flowing in our networks. They are simply a way for us to see the network traffic and bandwidth information over the entire IT infrastructure. The technique of using a packet sniffer to sniff the data flowing over the wire or through thin air (wireless) is called packet sniffing.

The purpose of sniffing

Packet sniffing is performed in order to better understand what flows through our networks. Just as a poison flowing through the veins of the human body has the potential to kill an individual, similarly malicious traffic traversing our networks can have a severe and sometime irreparable effect on the network devices, performance, and business continuity.

Sniffing helps a network analyst verify whether the implementation and functionality of the network and network security devices, such as the router, switch, firewall, IDS, or IPS, are as expected and also confirms that data is traversing through secure channels of communication.

Security analysts use sniffing to gather evidence in the case of a security breach with regard to the source of the attack, time and duration of the attack, protocols and port numbers involved, and data transmitted for the purpose of the attack. It can also help to prove the use of any insecure protocol(s) used to transmit sensitive information.

As Christopher Hitchens, a British-born American author, was once quoted saying:

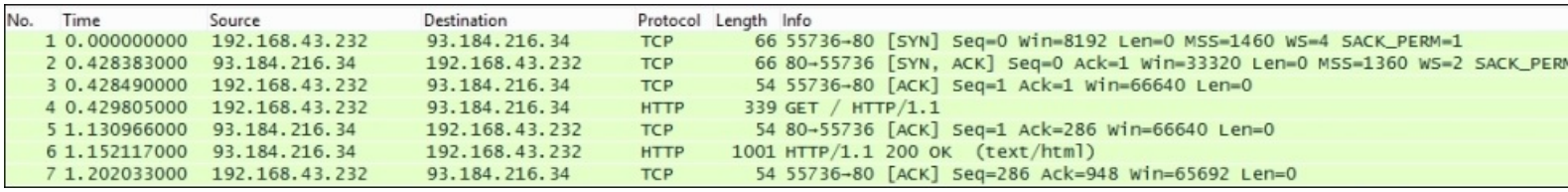
"That which can be asserted without evidence, can be dismissed without evidence."

Using a packet sniffer helps us get that piece of evidence.

Packet analysis

Now, to figure out whether the smell of the perfume is pleasant, ambrosial, or reeking is the analysis part. Hence, the art of interpreting and analyzing packets flowing through the networ is known as packet analysis or network analysis. Mastering this art is a well-honed skill and can be achieved if a network administrator has a solid understanding of the TCP/IP protocol suite, is familiar with packet flows, and has an excellent grasp of any sniffer of choice.

Learning technology at the packet level helps to cement the most difficult concepts. For an easy example, let's say that a user wants to browse a website named `example.com`. As soon as the user enters the URL in the address bar and hits **GO**, the packets start to flow on the network with respect to that request. To understand this packet flow, we need to start sniffing to look at the packets in transit. The following screenshot shows the packets that traversed the network when the user opened `example.com`.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.43.232	93.184.216.34	TCP	66	55736→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
2	0.428383000	93.184.216.34	192.168.43.232	TCP	66	80→55736 [SYN, ACK] Seq=0 Ack=1 Win=33320 Len=0 MSS=1360 WS=2 SACK_PERM
3	0.428490000	192.168.43.232	93.184.216.34	TCP	54	55736→80 [ACK] Seq=1 Ack=1 Win=66640 Len=0
4	0.429805000	192.168.43.232	93.184.216.34	HTTP	339	GET / HTTP/1.1
5	1.130966000	93.184.216.34	192.168.43.232	TCP	54	80→55736 [ACK] Seq=1 Ack=286 Win=66640 Len=0
6	1.152117000	93.184.216.34	192.168.43.232	HTTP	1001	HTTP/1.1 200 OK (text/html)
7	1.202033000	192.168.43.232	93.184.216.34	TCP	54	55736→80 [ACK] Seq=286 Ack=948 Win=65692 Len=0

We can analyze the packets after capturing them using a sniffer of choice, and in our case, we notice the columns that tell us about the source and destination IP addresses, the protoc being used, the length of the individual packets, and other relevant information. We will be digging into more detailed analysis as we progress though this book.

When we talk about enterprise networks, at any given point, there is humongous amount of traffic on the wire and analyzing such traffic is not a walk in the park. This traffic may be generated by numerous network devices communicating among each other, servers responding to user requests, or making their own requests over the Internet when required, and end users trying to accomplish their day-to-day tasks at work. There is no better way to understand this flow of information than to perform a packet-level analysis and, as the famou quote about network analysis goes, *packets never lie*. In addition, Gerald Combs, the man behind Wireshark, once tweeted the following:

""The packets never lie" but as traffic volumes increase you end up with a trillion truths. The trick is finding the important ones."

Learning such tricks comes only with experience, as with anything else in the field of IT. As a example, if you want to improve your programming skills, you have to practice code writing