Packet Analysis with Wireshark

Wireshark provides a very useful way to decode an RFC and examine it. The packet captures displayed in Wireshark give you an insight into the security and flaws of different protocols to help you perform security research and protocol debugging.

This book starts by introducing you to various packet analyzers to find out which one best suits your needs. You will learn how to use the command line and the Wireshark GUI to capture packets by employing filters. Moving on, you will learn about TCP/IP communication and gain an understanding of the SSL/TLS flow with Wireshark. Next, you will perform analysis on application-related protocols. We follow this with some best practices to analyze wireless traffic and identify different attack vectors such as DoS, ARP, Heartbleed, and many more. By the end of the book, you will have developed the skills needed to identify packets for malicious attacks.

Who this book is written for

If you are a network administrator who wants to effectively capture packets, a security consultant who wants to audit packet flows, or a white hat hacker who wants to view and remediate sensitive information, this book is for you. This book requires decoding skills and a basic understanding of networking.

What you will learn from this book

- Utilize Wireshark’s advanced features to analyze packet captures
- Capture network packets with tcpdump and snoop with examples
- Enhance your TCP/IP troubleshooting skill by solving practical use cases on the TCP/IP state diagram and handshake process
- Deep dive into SSL/TLS handshake
- Get to know more about protocols such as DHCPv6, DHCP, DNS, and HTTP with Wireshark
- Set up 802.11 WLAN captures and discover more about the WAN protocol
- Learn security aspects and their mitigation techniques for attack such as DoS / ARP Poisoning and more

Packet Analysis with Wireshark

Leverage the power of Wireshark to troubleshoot your networking issues by using effective packet analysis techniques and performing an improved protocol analysis.
In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 'Packet Analyzers'
- A synopsis of the book’s content
- More information on Packet Analysis with Wireshark
About the Author

Anish Nath is a software engineer who has more than 10 years of experience. He works at CISCO, and at CISCO, he started using Wireshark for the first time. He is thankful to CISCO. He doesn't speak much, but likes to explore new things that he has not tried or not thought of. He also tries his best to be successful at this. Though he fails a lot of time, this gives him more experience, and when success comes, he thanks all of his efforts that had failed him initially.

You can reach him at https://in.linkedin.com/in/anishnath, and his Twitter handle is @anish2good.
The purpose of this book is to identify, learn about, and solve issues related to protocol, network, and security, and see how Wireshark helps to analyze these patterns by allowing its features to troubleshoot effectively. This book has lab exercises and contains packet capture files for offline viewing and analyses. Most of the examples contain production-like scenarios and their solutions and steps to reproduce these solutions.

This book also contains effective capturing methods that can be used directly in production without installing Wireshark.

Wireshark is an awesome tool for troubleshooting and learning, and within the scope of this book, we have taken the best use cases for different types of audiences, such as network administrators, security auditors, protocol learners, and troubleshooters.

What this book covers

Chapter 1, Packet Analyzers, covers the definition of packet analyzers and their use cases, network interfaces naming conventions, pcap/pcapng file extensions, and types of network analyzer tools.

Chapter 2, Capturing Packets, covers how to capture packets using Wireshark, tcpdump, and snoop; how to use Wireshark display filters; and how to use Wireshark's cool features such as Decode-As and protocol preferences. Also, we will cover the TCP stream, exporting images, generating a firewall ACL rule, autocapture setup, and the name resolution feature.

Chapter 3, Analyzing the TCP Network, covers the TCP state machine, TCP connection establishment and closing sequence, practical troubleshooting labs such as (CLOSE_WAIT, TIME_WAIT), how to identify and fix latency issues, and Wireshark TCP sequence analysis flag (zero window, dup-ok, TCP retransmission, and window update) features.
Chapter 4, *Analyzing SSL/TLS*, covers the TLS/SSL two-way mutual authentication process with Wireshark, SSL/TLS decryption with Wireshark, and the identification of handshake failure with Wireshark.

Chapter 5, *Analyzing Application Layer Protocols*, covers how to analyze a protocol using the Wireshark display filter, how these protocols work, how to simulate these packets, capture, and display them using tcpdump/Wireshark.

Chapter 6, *WLAN Capturing*, covers WLAN capture setup and monitor mode, capturing with tcpdump, 802.11 display filters, Layer-2 datagram frames types, Wireshark display filters, and other Wi-Fi Sniffing products available.

Chapter 7, *Security Analysis*, covers the security aspect with Wireshark and discusses uses cases such as the Heartbleed bug, SYN flood/mitigation, ICMP flood/mitigation, MITM, BitTorrent, and host scanning.
1 Packet Analyzers

A packet analyzer is also known as a packet sniffer or a network protocol analyzer. Packet analyzer has the ability to grab the raw packet from the wire, wireless, Bluetooth, VLAN, PPP, and other network types, without getting processed by the application. By doing so it brings the whole science and innovation to this field. In this chapter we will see a few use cases of the packet analyzer by covering the following topics:

- Uses for packet analyzers
- Introducing Wireshark
- Other packet analyzer tools
- Mobile packet capturing

**Uses for packet analyzers**

More practically, packet analyzers are employed in network security and to analyze raw traffic so as to detect scans and attacks, and for sniffing, network troubleshooting, and many more uses, as shown in the following image:
Packet Analyzers

Packet analyzers can be used as follows:

- Network administrators can diagnose problems on a network
- Security architects can perform a security audit on a packet
- Protocol developers can diagnose/learn protocol-related issues
- White-hat hackers can find vulnerabilities in the application and fix them before black-hat hackers find them

The use is not limited to these bullet points. There are lots of new tools and innovations happening in this area. Find a use case and build your own packet analyzer; the best example is Wireshark.

Introducing Wireshark

Wireshark is perhaps one of the best open source packet analyzers available today. Wireshark is a powerful packet analyzer tool, with an easy-to-use, rich GUI and a command-line utility with very active community support: http://ask.wireshark.org.

Wireshark uses pcap (libpcap) to capture packets, which means it can capture packets in offline mode—to view the captured packets—and online mode (live traffic) to capture and display the traffic in the Wireshark GUI. Once open, the Wireshark GUI looks like this:
Wireshark features

We will see some of the important features that are available in Wireshark in the following figure:

Wireshark has the following cool built-in features, few of them are listed as follows:

- Available in both UNIX and Windows
- Ability to capture live packets from various types of interface
- Filters packets with many criteria
- Ability to decode larger sets of protocols
- Can save and merge captured packets
- Can create various statistics
- User-friendly GUI and command-line interface
- Active community support (http://ask.wireshark.org)
Wireshark's dumpcap and tshark

The Wireshark installation provides some command-line tools such as dumpcap and tshark. Wireshark and tshark rely on dumpcap to capture traffic; more advanced functionality is performed by tshark. Also note that dumpcap can be run as its own standalone utility. tshark is a command-line version of Wireshark and can be used in the remote terminal.

The Wireshark packet capture process

The user must be aware of where Wireshark is installed and it should be obliged with your organization policy before start capturing on the TAP (Test Access Point) or Switch Port Analyzer (SPAN) port.

Usually developers install Wireshark on their personal laptop/desktop and capture packets, which goes in-out from the box.

Certain guidelines should be followed to perform this:

1. Make sure you’re allowed to do what you’re going to do; check your corporate policies before capturing a packet.
2. The operating system must support packet capturing:
   - Linux packet socket support is enabled in the kernel by default
   - Windows requires WinPCap to be installed
3. Choose the interface and enable the promiscuous mode on it. Promiscuous mode accepts all packets whether they are addressed to the interface or not.
4. If using a Wi-Fi interface, enable the monitor mode for WLAN capturing.
5. Start capturing and use Wireshark's different features like (filters/statistics/IO/save) for further analysis
Other packet analyzer tools

Wireshark is a packet analysis tool to use features such as packet editing/replaying, performing MITM, ARPspoof, IDS, and HTTP proxy, and there are other packet analyzer tools available and can be used as well.

The following is a list (not limited) of notable packet analyzer tools on the market; many others are commercially available. The table lists tools and their features:

<table>
<thead>
<tr>
<th>Tools</th>
<th>Packet editing</th>
<th>Packet replay</th>
<th>ARPspoof/MITM</th>
<th>Password sniffing</th>
<th>Intrusion detection</th>
<th>HTTP debugger</th>
</tr>
</thead>
<tbody>
<tr>
<td>WireEdit (<a href="https://wireedit.com/">https://wireedit.com/</a>)</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Scapy (<a href="http://www.secdev.org/">http://www.secdev.org/</a>)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Ettercap (<a href="https://ettercap.github.io/ettercap/">https://ettercap.github.io/ettercap/</a>)</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Tcpreplay (<a href="http://tcpreplay.synfin.net/">http://tcpreplay.synfin.net/</a>)</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Bit-Twist (<a href="http://bittwist.sourceforge.net/">http://bittwist.sourceforge.net/</a>)</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Cain (<a href="http://www.oxid.it/cain.html">http://www.oxid.it/cain.html</a>)</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Snort (<a href="https://www.snort.org/">https://www.snort.org/</a>)</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>
Mobile packet capture

Wireshark is not available on mobile platforms such as Android, iOS, or Windows. In order to capture mobile traffic the following tools are suggested based on the platform:

<table>
<thead>
<tr>
<th>Platform</th>
<th>Packet capture tool used</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>iOS</td>
<td>Paros</td>
<td><a href="http://sourceforge.net/projects/paros/">http://sourceforge.net/projects/paros/</a></td>
</tr>
<tr>
<td>Android</td>
<td>Shark for Root</td>
<td><a href="http://www.appbrain.com/app/shark-for-root/lv.n3o.shark">http://www.appbrain.com/app/shark-for-root/lv.n3o.shark</a></td>
</tr>
<tr>
<td></td>
<td>Kismet Android PCAP</td>
<td><a href="http://www.kismetwireless.net/android">http://www.kismetwireless.net/android</a> pcap/</td>
</tr>
</tbody>
</table>

Various other techniques are used to capture mobile traffic using Wireshark. One such technique is creating a Wi-Fi hotspot on the laptop, allowing the mobile phone to use this Wi-Fi, and sniffing traffic on your Wi-Fi interface using Wireshark.

Summary

In this chapter we learned what packet analyzers are and what their use cases are. After a quick introduction to Wireshark, we covered what goes on behind-the-scenes when Wireshark captures packets; Wireshark benefits and important features; the necessary prerequisites before capturing packets; and other packet analyzer tools for packet editing/sniffing/replaying and so on. We also provided a brief overview of mobile packet capturing.

The next chapter will be more specific to Wireshark and its tips and tricks. After that we will explore TCP troubleshooting, then plunge into SSL, and other application protocols such as DHCPv6, DHCP, DNS, and HTTP. We will also analyze Wi-Fi capturing and carry out some security analyses with the help of Wireshark and tcpdump.
Where to buy this book
You can buy Packet Analysis with Wireshark from the Packt Publishing website.
Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.
Click here for ordering and shipping details.