Capture an HTTP and an HTTPS Wireshark PCAP

Reading

35m

Status Incomplete

Introduction

After you report the problems of the servers, the manager wants a test to verify if the information is passing with the HTTP traffic. For the test, you can use Wireshark to capture the traffic and analyze the pieces of information. But before you start with the test, you will recap Wireshark and how to use it.

Reading

Recap of Wireshark

Wireshark is an open-source network protocol analysis software program created by Gerald Combs in 1998. A worldwide organization of networking experts and software developers maintains Wireshark and continues to release updates for new networking technologies and encryption methods.

Government agencies, companies, non-profit organizations, and educational institutions use Wireshark for problem-solving and educational purposes. There are doubts about the legality of Wireshark, as it is a powerful packet analyzer. The bright side of the Force says you should only use Wireshark on networks where you are allowed to inspect data packets. Using Wireshark to examine packages without permission is a path to the dark side.

How Does Wireshark Work?

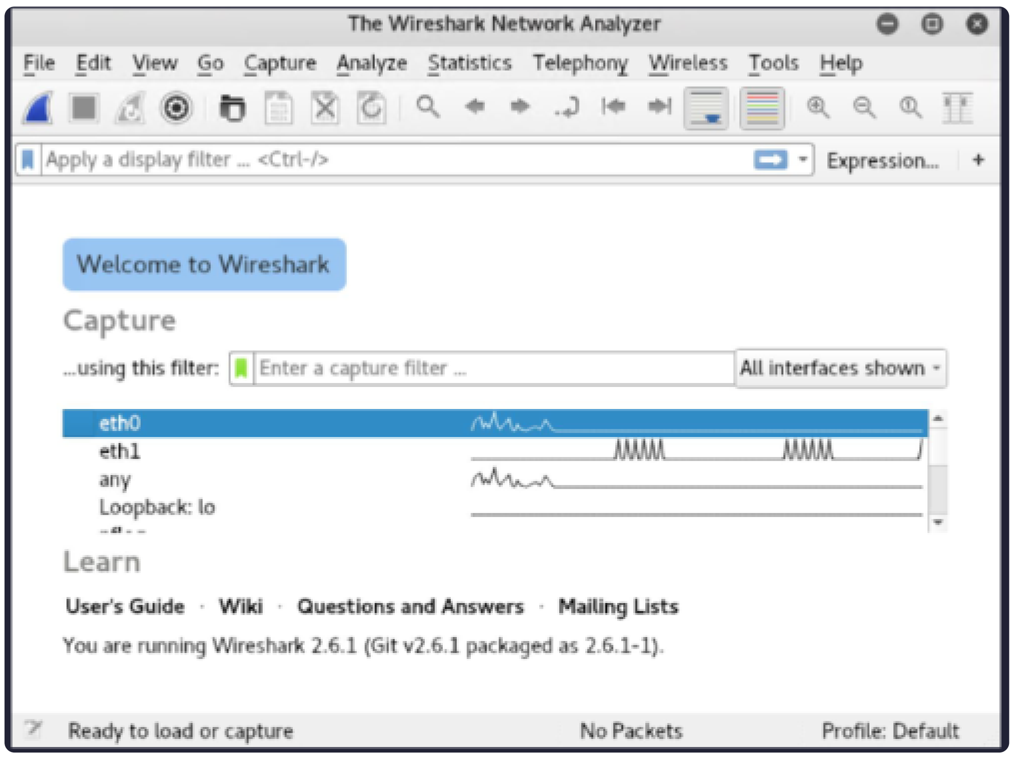
Wireshark is a packet analysis and network sniffer tool. It captures network traffic on the local network and stores this data for offline analysis. Wireshark captures network traffic from Ethernet, Bluetooth, wireless (IEEE.802.11), token rings, frame relay connections, and more. Remember that a "packet" is a single message of any network protocol (i.e., TCP, DNS, etc.) and the LAN traffic is in broadcast mode. This means that a single computer with Wireshark installed can monitor traffic between two other computers. You must capture the packets on your local computer to analyze traffic from an external website.

Wireshark allows you to filter the log before capture starts so you can focus on what you're looking for in the network trace. You can define a filter to monitor TCP/HTTP network traffic between two IP addresses. You can configure the filter only to show packets sent from one computer. The filters in Wireshark are one of the main reasons that made it the standard tool for packet analysis.

How to Capture Data Packets in Wireshark

When you open Wireshark, a list of all network connections you can monitor appears on the screen. There's also a trap filter field, so you capture only the network traffic you want to watch.

As shown in the image below, you can select the network interface to capture the traffic:


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
+
Welcome to Wireshark
Capture
...using this filter:
Enter a capture filter ...
All interfaces shown
eth0
eth1
any
Loopback: lo
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists

You are running Wireshark 2.6.1 (Git v2.6.1 packaged as 2.6.1-1).

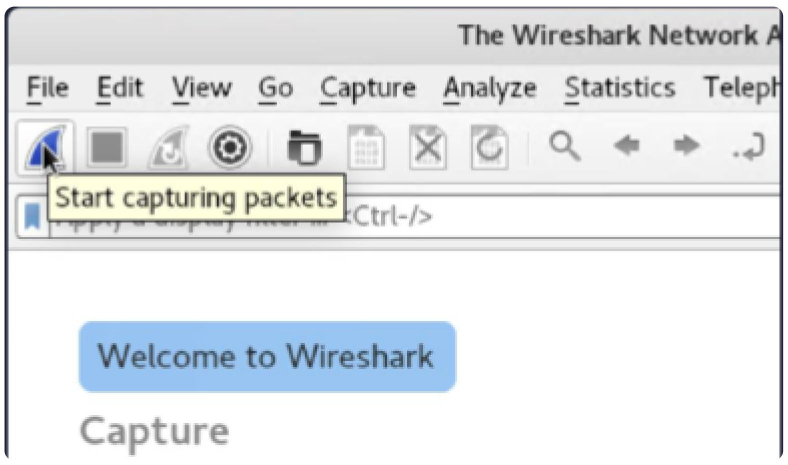
Ready to load or capture

No Packets

Profile: Default

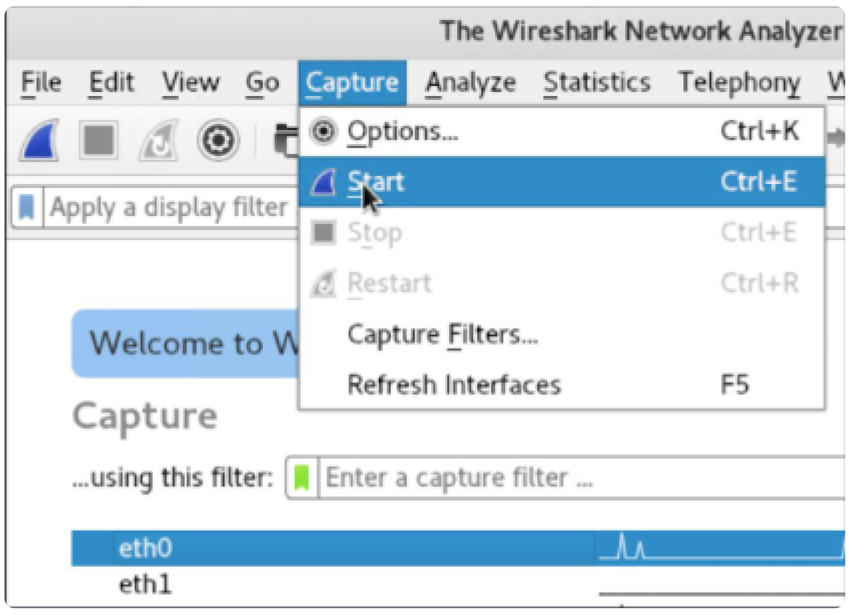
Selecting one or more network interfaces with the "shift" key and the left mouse button is possible. After selecting the network interface, you can start capturing. There are several ways to do this.

Click on “Start Capturing Packets”, the first button on the toolbar, as seen in the image below:

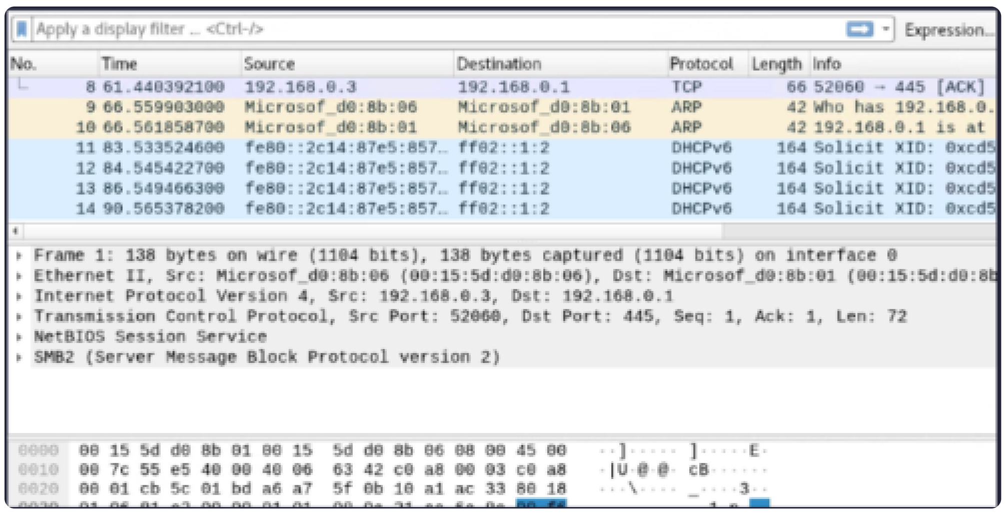


Another example to start the capture is using the menu, clicking Capture -> Start.

The image below shows you can use the "Ctrl" + "E" keys as a shortcut to start the capture:



When the capture starts, and during capture, Wireshark will show the packets it is capturing in real time. When all the packages you need have been captured, use the same keys or click on the same menu options to stop capturing.



```
No.    Time           Source                   Destination          Protocol  Length Info
    8 61.440392100  192.168.0.3              192.168.0.1          TCP          66 52060 → 445 [ACK]
    9 66.559903000  Microsof_d0:8b:06        Microsof_d0:8b:01    ARP          42 Who has 192.168.0.
   10 66.561858700  Microsof_d0:8b:01        Microsof_d0:8b:06    ARP          42 192.168.0.1 is at
   11 83.533524600  fe80::2c14:87e5:857…     ff02::1:2            DHCPv6      164 Solicit XID: 0xcd5
   12 84.545422700  fe80::2c14:87e5:857…     ff02::1:2            DHCPv6      164 Solicit XID: 0xcd5
   13 86.549466300  fe80::2c14:87e5:857…     ff02::1:2            DHCPv6      164 Solicit XID: 0xcd5
   14 90.565378200  fe80::2c14:87e5:857…     ff02::1:2            DHCPv6      164 Solicit XID: 0xcd5

› Frame 1: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits) on interface 0
› Ethernet II, Src: Microsof_d0:8b:06 (00:15:5d:d0:8b:06), Dst: Microsof_d0:8b:01 (00:15:5d:d0:8b
› Internet Protocol Version 4, Src: 192.168.0.3, Dst: 192.168.0.1
› Transmission Control Protocol, Src Port: 52060, Dst Port: 445, Seq: 1, Ack: 1, Len: 72
› NetBIOS Session Service
› SMB2 (Server Message Block Protocol version 2)

0000  00 15 5d d0 8b 01 00 15  5d d0 8b 06 08 00 45 00   ··]····· ]·····E·
0010  00 7c 55 e5 40 00 40 06  63 42 c0 a8 00 03 c0 a8   ·|U·@·@· cB······
0020  00 01 cb 5c 01 bd a6 a7  5f 0b 10 a1 ac 33 80 18   ···\···· _····3··
```

How to Analyze Data Packets in Wireshark

Wireshark displays three different panes for inspecting packet data. The top panel, titled "Packet List", lists all packets in the capture. When clicking on a package, the other two panels change to show details about the selected package. You can also see if the packet is part of a conversation, that is, traffic between two specific endpoints. Some details about each column in the top panel (in the image above) are explained here:

- No.: represents the numerical position of the captured packet. The square bracket indicates that this packet is part of a conversation.
- Time: shows how much time it took from the beginning of the capture to the capture of the packet. Changing the display format of these numbers is possible if you want something different.
- Source: the address of the system that sent the packet.
- Destination: the destination address of the packet.
- Protocol: indicates the type of packet, for example, TCP, DNS, HTTP, or ARP.
- Length: displays the packet size in bytes.
- Info: displays more information about the package content. This column varies by package type.

"Packet Details", the middle pane, displays as much readable information about the package as possible, depending on the package type. You can create filters based on the highlighted text in this field by right-clicking.

The bottom panel, "Packet Bytes", shows the packet precisely as it was captured in hexadecimal.

When looking at a packaged part of a conversation, you can right-click on the package and select "Follow" to see only the packages that are part of that conversation.

Wireshark Filters

One of the best features of Wireshark is capture filters and display filters. Filters allow you to view the capture as needed to troubleshoot issues. See below.

Wireshark Capture filters

Capture filters restrict captured packets to the filter. If packets do not match the filter, they will not be saved by Wireshark. Check out some examples of capture filters:

- host IP-address: limits the capture to traffic originating from and destined to the IP address.
- net 192.168.0.0/24: captures all traffic on the subnet.
- dst host IP-address: captures packets sent to the specified host.
- port 53: captures traffic only on port 53.
- port not 53 and not arp: captures all traffic except DNS and ARP traffic.

Wireshark Display Filters

Wireshark's view filters change the view of the capture during analysis. After you stop capturing packets, use display filters to filter packets in the Packet List panel, so you can solve the problem.

The most useful view filter (based on experience) is:

```
ip.src==IP-address and ip.dst==IP-address
```

This filter shows packets from one computer (ip.src) to another (ip.dst). It is also possible to use ip.addr to show packets originating from and destined to an IP address. Check out some more filters below:

`tcp.port eq 25`: will show all traffic on port 25, which is usually SMTP traffic.

`icmp`: will display only ICMP traffic in the capture (it probably pings).

`ip.addr != IP_address`: displays all traffic except traffic to or from the specified computer.

Analysts even create filters to detect specific attacks, such as the following filter to detect Sasser:

```
ls_ads.opnum==0x09
```

About Network Sniffing

Sniffing is the process of intercepting data packets sent over a network. A specialized software program or hardware equipment can do this.

Sniffing can be used to:

- Capture sensitive data such as login credentials.
- Listen to chat messages.
- Capture files that were transmitted over a network.
- Follow protocols that are vulnerable to sniffing:
- Telnet
- rlogin
- HTTP
- SMTP
- NNTP
- POP
- FTP
- IMAP

Reference

Wireshark.org_

✓ Mark Completed

← Previous
Configure Apache Web Server

Next
Report to IT Manager →

How well did this activity help you to understand the content?

Let us know how we're doing

☆ ☆ ☆ ☆ ☆

W07D3

Wed Aug 7

Lectures (1)

Work (7)

6 hrs

- Configure Apache Web Server
- Capture an HTTP and an HTTPS Wireshark PCAP
- Report to IT Manager
- Hash Functions, Data Integrity and Digital Signatures
- Key Establishment Protocols and Management Techniques
- Data Integrity Email
- Project Reading

W07D3 Schedule »