Netzwerkanalyse mit Wireshark: Was passiert im Netzwerk?

Luis Herzog

April 2023

Abbildung 1: Wireshark Logo

Inhaltsverzeichnis

Es gibt viele verschiedenen Arten von Daten. Somit ebenfalls verschiedene Definitionen.

Bevor man über die Wireshark Software sprechen kann, müssen die Basics geklärt werden:

- Was sind Protokolle?
- Welche Protokolle gibt es?
- Wie funktionieren die Protokolle?
- welcche Protokolle (welche können ausgelesen werden?

1 Daten

1.1 Metadaten

2 Das Programm: Wireshark

Als "Schweizer Taschenmesser"[3] wird es von heise.de bezeichnet. Es wird so gut wie überall in der IT-Branche benutzt. Sei es beim Finden von Netzwerkproblemen oder beim Lernen, wie ein Netzwerk funktioniert. Es bietet eine Menge Möglichkeiten

2.1 Programm

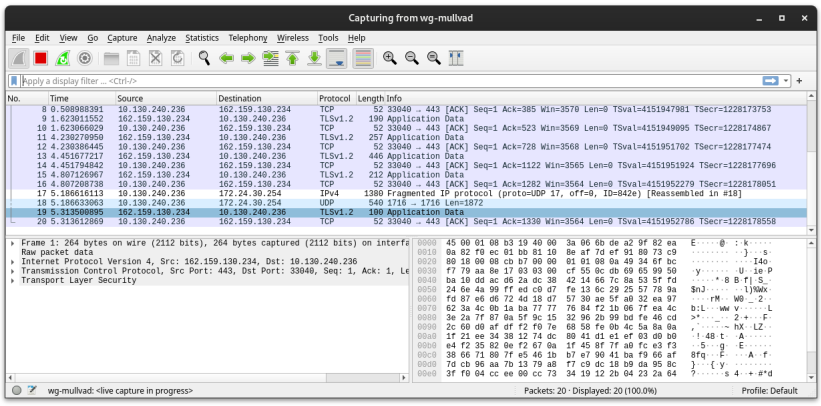

Abbildung 2: Beispielscreenshot aus der Wireshark Software

2.2 Funktionsumfang

Wireshark ist eine Software mit sehr vielen Funktionen. Sie bietet wirklich alles rund um das Thema Netzwerkanalyse. Die wichtigsten Funktionen sind:

- Die Verfügbarkeit aus sehr vielen Platformen
- Die Möglichkeit dumps von anderen Personen importieren zu können
- Das Filtern von Paketen nach sehr vielen Kriterien

Besonders die letztere Funktion ist sehr wichtig, da man ohne diese schnell die Orientierung in der Software verlieren kann. [1]

2.3 Anwendungsbereiche

Wie im oberen Teil schon dargestellt, hat Wireshark sehr viele Funktionen. Allein deswegen wird es auch als Schweizer Taschenmesser der Netzwerktechnik bezeichnet. Somit kann Wireshark in sehr vielen Situationen Anwendung finden. Es kann beispielsweise von Netzwerk Administratoren benutzt werden, um Netzwerk Probleme zu Analysieren und zu lösen. Es kann außerdem von den Sicherheitsanalysten eines Netzwerks benuzt werden, um Sicherheitsprobleme in Netzwerken zu finden. Wiederum kann es auch von Anwendungsentwicklern benutzt werden, um Netzwerkprotokoll Implementationen zu debuggen. Zuletzt kann es auch benutzt werden um mehr über Netzwerktraffic zu lernen und diesen zu Analysieren. Es gibt natürlich auch viele weitere Möglichkeiten Wireshark zu benutzen. [2]

3 Netzwerk

3.1 Aufbau

3.2 Protokolle

3.3 Open System Interconnection Modell

Das Open System Interconnect[1] Modell, auch OSI-Modell genannt, beschreibt die Voraussetzungen, die für eine Kommunikation innerhalb eines Netzwerks nötig sind. Dieses wurde 1983 von durch die 'Internationale Organisation für Normung', kurz 'ISO' standardisiert. Dies ist notwendig, damit sich alle Komponenten im Netwerk, auch wenn diese von verschiedenen Herstellern produziert wurden, reibungslos miteinander funktionieren. Wenn ein Paket beispielsweise von einem Computer im Netwerk losgeschickt wird, muss es mehrere Stationen durchlaufen. Das Paket verlässt den Rechner über die Netzwerkkarte und wird durch ein Übertragungsmedium über weitere Netzwerkkomponenten, wie Hubs oder Router bis zur Netzwerkkarte des Zielrechners geleitet. Dort wird dieses dann Interpretiert, um korrekt dargestellt zu werden. All diese Schritte werden durch ein Protokoll festgehalten und durch das OSI-Modell spezifiziert, damit jede Station auf diesem Weg weiß, wohin das Paket möchte, woher es kommt und welche Eigenschaften es hat. So wird ein Standart geschaffen, mit dem alle Computersysteme miteinander kommunizieren können.

Abbildung 3: OSI-Modell

Da diese Datenkommunikation relativ komplex ist, wurde das Modell in sieben Schichten eingeteilt. Die oberen vier Schichten gehören zu den Anwendungsorientierten Schichten[2]. Die unteren drei Schichten werden Transport Schichten[3] genannt. Jede Schicht behandelt eine Anforderung, die für eine funktionierende Kommunikation erfüllt werden muss. Ein zu übertragenes Paket durchläuft vor der Versendung die Schichten 7 - 2, wobei dem Paket bei jeder Schicht Protokoll-Informationen hinzugefügt werden, die dann im Protokoll des Datenpaketes auffind-

[1] dt: Offenes System für Kommunikationsverbindungen
[2] engl: Host layer
[3] dt: Media layer

bar sind. Die erste und letzte Schicht wandelt das Paket in technisch übertragbare Daten um und schickt dieses über das Übertragsmedium weg. Das Übertragsmedium kann hierbei ein Kabel sein, oder aus einer Antenne bestehen. Auf der Empfängerseite wird dieser Prozess rückwärts durchgeführt. Hierbei wird die jeweilige Protokoll-Information nach der Interpretierung durch die jeweilige Schicht entfernt, bis zum Inhalt des Paketes.

Im folgenden werden die einzelnen Schichten einzeln beleuchtet, um einen besseren Einblick zu gewähren.

3.3.1 Anwendungsschicht

Die Anwendungsschicht[4] stellt die Daten dar, mit welchen der Nutzer interagiert. Softwareanwendungen, wie Web-Browser und E-Mail clients stützen sich auf die siebte Schicht, um dem Nutzer aussagekräftige Daten zu präsentieren.

Hierzu gehören Protokolle, wie HTTP[5], welches benutzt wird, um Websites welche in HTML[6] geschrieben sind zu präsentieren, oder SMTP[7], welches benutzt wird um E-Mails zu präsentieren.

3.3.2 Präsentationsschicht

Die Präsentationsschicht[8] ist in erster Linie dafür verantwortlich, die Daten so aufzubereiten, dass diese in der Anwendungsschicht verwendet werden können. Ein wichtiger Teil dabei ist die Verschlüsselung. Die Präsentationsschicht muss, wenn die Geräte durch eine verschlüsselte Verbindung kommunizieren, auf Senderseite eine Verschlüsselung hinzufügen und diese auf der Empfängerseite korrekt dekodieren.

Ebenso ist die Präsentationsschicht für die Komprimierung der Daten verantwortlich. Dadurch kann die Geschwindigkeit und Effizienz der Kommunikation erhöht und die benötigte Bandbreite minimiert werden.

[4] engl: application layer
[5] Hyper Text Transfer Protocol
[6] Hyper Text Markup Language
[7] Simple Mail Transfer Protocol
[8] engl: presentation layer

3.3.3 Sitzungsschicht

Anschließend folgt die Sitzungsschicht[9]. Diese ist für das Öffnen und Schließen der Kommunikation der beiden Geräte zuständig. Hier wird die Kommunikation in Sitzungen eingeteilt. Eine Sitzung reicht von der Öffnung bis zur Schließung der Verbindung. Somit wird sicher gestellt, dass die Sitzung lange genug geöffnet bleibt, um alle Daten zu übertragen. Wenn alle Daten erfolgreich übertragen wurden, leitet die Sitzungsschicht die umgehende Schließung der Sitzung ein, um Ressourcen zu sparen.

Eine weitere sehr wichtige Aufgabe der Sitzungsschicht ist die Sicherung der Datenverbindung durch synchronisierte Checkpoints. Wenn beispielsweise bei der Übertragung einer 450 Megabyte großen Datei bei 234 Megabyte die Verbindung unterbrochen wird, kann nach einer Neuverbindung die Übertragung bei 230 Megabyte wieder aufgenommen werden, da es einen Checkpoint der Datei bei 230 Megabyte gibt.

3.3.4 Transportschicht

In der Transportschicht[10] werden die Datenpakete vor dem Versenden in Segmente zerlegt. Im Empfangsgerät werden die diese Segmente durch die Transportschicht wieder korrekt zusammengesetzt, sodass diese von der Sitzungsschicht benutzt werden können.

Die Transportebene ist ebenfalls für die Fluss- und Fehlersteuerung zuständig. Hierbei wird die Übertragungsgeschwindigkeit so festgelegt, dass ein ggf. langsamer Empfänger nicht durch die ggf. schnelle Geschwindigkeit des Senders überfordert wird. Beim Empfänger wird durch die Fehlersteuerung ein vollständiger Empfang aller Daten sichergestellt. Wenn die empfangenen Daten nicht vollständig sind, werden diese durch dieses System erneut angefordert, um die Vollständigkeit der Daten zu garantieren.

Hierzu gehören die Protokolle TCP[11] und UDP[12]

Exkurs: UDP/TCP

[9] engl: session layer
[10] engl: transport layer
[11] Transmission Control Protocol
[12] User Datagram Protocol

3.3.5 Netzwerkschicht

Darauf folgt die Netzwerkschicht[13]. Diese gewährt das Kommunizieren zwischen Geräten in verschiedenen, miteinander verbundenen Netzwerken. Beim Versenden, werden die Segmente der Transportschicht erneut in kleinere Datenpakete aufgeteilt und mit weiteren Informationen versehen. Diese Informationen sind für die, auf dem Weg gelegenen, Knoten gedacht, um diesen das Ziel des Pakets aufzuweisen. Es wird der beste physikalisch mögliche Weg ausgesucht, um die Daten sicher an ihr Ziel zu bringen. Dieser Prozess wird Routing[14] genannt. Wenn sich beide Geräte im selben Netzwerk befinden, wird diese Ebene übersprungen.

Zu den Protokollen für diese Schicht gehören das IP[15], das ICMP[16], das IGMP[17] und die IPsec[18] Suite. Ein Beispielgerät wäre ein Router.

3.3.6 Sicherungsschicht

Die Sicherungsschicht[19] stellt die vorletzte Schicht dar. Diese ist der Netzwerkschicht sehr ähnlich. Der wesentliche Unterschied ist, dass die Sicherungsschicht für die Kommunikation von zwei Geräten innerhalb eines Netzwerks zuständig ist. Die Sicherungsschicht ist ebenfalls für die Fluss- und Fehlerkontrolle in der netzinternen Kommunikation zuständig.

Beispielgeräte für diese Ebene wären Bridges und Switches.

3.3.7 Bitübertragungsschicht

Die unterste Schicht wird durch die Bitübertragungsschicht[20] dargestellt. Hier sind die Daten als Bitstrom vorhanden, eine Zeichenkette bestehend aus einen und nullen. Hier muss sich auf mehrere Konventionen geeinigt werden. Zu Beginn müssen die Gegebenheiten des Übertragungsmediums festgelegt werden. Dies betrifft die Wahl des Materials und die Funktion der einzelnen Leitungen. Ein Kabel kann beispielsweise aus Kupfer oder Glasfaser bestehen und zwei innere Leitungen haben: eine Datenleitung und eine Steuerleitung. Bei der Übertragung über Funk

[13] engl: network layer
[14] Quelle!
[15] Internet Protocol
[16] Internet Control Message Protocol
[17] Internet Group Message Protocol
[18] ??
[19] engl: Data Link Layer
[20] engl: physical layer

wird beispielsweise durch Luft übertragen. Es muss ebenfalls die Übertragungsrichtung und Geschwindigkeit festgelegt werden. Ein Kabel kann in eine Richtung[21], abwechselnd in beide Richtungen[22] oder in beide Richtungen gleichzeitig[23] übertragen.

4 Analyse

In der Analyse soll viel benutzte Software getestet werden. Ich habe mich für Web-Browser entschieden, da diese am meisten benutzt werden[24]. Zu beginn soll die Software getestet werden. Hierfür habe ich einen simplen Test erstellt.

4.1 Beispiel

In diesem Beispiel wird ein Webserver erstellt und anschließend kontaktiert. Dieser Transfer wird dann durch Wireshark aufgezeichnet.

4.1.1 Der Webserver

Code:

```
from flask import Flask
import os
from datetime import datetime

hostName = "localhost"
serverPort = 8080
app = Flask(__name__)
@app.route('/')
def index():
return '<p>willkommen in der http webserver demo</p>'
@app.route('/time')
def time():
now = datetime.now()
current_time = now.strftime("%H:%M:%S")
return f'Jetzige Zeit: {current_time}'
```

[21] simplex
[22] halb-duplex
[23] duplex
[24] QUELLE!

```
if __name__ == '__main__':
app.run(host=hostName, port=serverPort)
```

4.1.2 Das ‘loopback device‘

4.1.3 Ergebnis

4.2 Browser

4.2.1 Mozilla Firefox

4.2.2 Microsoft Edge

4.2.3 Mullvad Browser

4.2.4 Opera Browser

5 Fazit

Abbildungsverzeichnis

Literatur

[1] The Wireshark Contributors. Wireshark - features. Zuletzt besucht: 15. Juni 2023, Quelle: https://www.wireshark.org/docs.

[2] The Wireshark Contributors. Wireshark - intended purposes. Zuletzt besucht: 16. Mai 2023, Quelle: https://www.wireshark.org/docs.

[3] Johannes Weber. Mit wireshark die last von ntp-servern messen. Zuletzt besucht: 31. Oktober 2023, Quelle: https://www.heise.de/hintergrund/Mit-Wireshark-die-Last-von-NTP-Servern-messen-4416283.html.

6 Anlagen