Netzwerkanalyse mit Wireshark: Was passiert im Netzwerk?

Luis Herzog

April 2023

Abbildung 1: Wireshark Logo [7]

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“[15] wird es von heise.de bezeichnet. Wireshark wird so gut wie überall in der IT-Branche benutzt. Sei es beim Finden von Netzwerkproblemen oder beim Lernen, wie ein Netzwerk funktioniert. Somit bietet das Programm eine Menge Möglichkeiten.

2.1 Geschichte

Wireshark wurde 1997 von Gerald Combs als ‘Ethereal‘ ins Leben gerufen. Das Programm wurde im Juli 1998 als Version 0.2.0 unter einer GPL Lizenz[9] veröffentlich.[3] Somit konnten viele Menschen am Code des Programms mithelfen, um es zu verbessern. Im Jahr 2006 wurde das Projekt in ‘Wireshark‘ umbenannt. Anschließend wurde im Jahr 2008 die Version 1.0 als erste komplette Version veröffentlicht. Sieben Jahre später wurde 2015 die Version 2.0 mit einem komplett neuen Benutzeroberfläche vorgestellt. Letztendlich wurde im Jahr 2023 die Wireshark Foundation[1] gegründet, welche jetzt das Programm stützt. [1]

[1] https://wiresharkfoundation.org/

2.2 Programm

Wireshark ist ein “network packet analyzer“[4]. Es erlaubt dem Benutzer den Netzwerkverkehr aufzunehmen und zu analysieren. Mit Wireshark kann man sozusagen in ein Netzwerkkabel ‘reinschauen‘.

2.3 Funktionsumfang

Wireshark ist eine Software mit sehr vielen Funktionen und bietet fast alles rund um das Thema Netzwerkanalyse. Mit wireshark kann man den Netzwerkverkehr aufnehmen, mit hohem Detail darstellen und einfach exportieren oder speichern. Außerdem kann man exportierte Netzwerkdumps von Wireshark anderen Programmen importieren. Da Wireshark sowohl auf UNIX Platformen, als auch auf Windows verfügbar ist, geht dies sogar Plattformübergreifend. Ebenso kann man die Datenpakete anhand verschiedener Kriterien Filtern und durchsuchen. Die gerade eben genannten Funktionen kratzen aber nur an der Oberfläche. Wireshark bietet noch viel mehr Funktionen.[5]

2.4 Die Benutzeroberfläche

Das voreingestellte Layout ist im unteren Screenshot dargestellt. In diesem Fall wird gerade die Netzwerkschnittstelle ‘wg-mullvad‘ aufgezeichnet. Im oberen Teil ist die Symbolleiste zu finden. Dort findet man Knöpfe zum starten und stoppen der Aufnahme. Recht daneben befinden sich die Knöpfe zum Laden und Speichern von Netzwerkdumps. Weiter rechts folgen Knöpfe zum Sortieren der Paketliste. Darunter befindet sich die Liste mit den aufgenommenen Paketen. Dort kann direkt die Zeit, Quelle und Ziel des Pakets abgelesen werden. Rechts daneben kann das benutzte Netzwerkprotokoll und die Länge abgelesen werden. Ganz rechts steht weitere Information zum Paket. Unter dieser Liste befinden sich zwei Fenster. Rechts steht der rohe Inhalt des Pakets. Links hingegen steht der von Wireshark automatisch ausgelesene und geordnete Inhalt des Pakets. Hier kann jegliche Information zum Datenpaket gefunden werden. Unter diesem Fenster steht der Name der Quelle, deren Netzwerkverkehr aufgezeichnet wird. Unten ist ein Beispielscreenshot zu sehen.

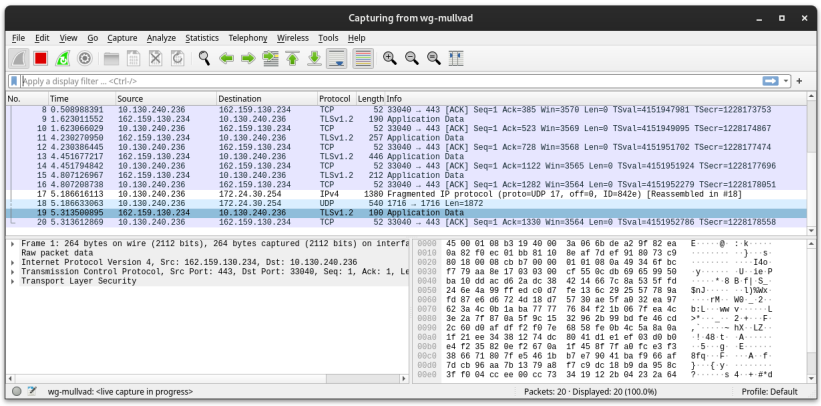

Abbildung 2: Beispielscreenshot aus der Wireshark Software [11]

2.5 Anwendungsbereiche

Wie schon im oberen Teil dargestellt, bietet Wireshark sehr viele Funktionen. Somit kann Wireshark in sehr vielen Situationen Anwendung finden. Es kann beispielsweise von Netzwerk Administratoren zum Finden, Analysieren und Lösen von Netzwerkproblemen benutzt werden. Das Programm kann außerdem von Sicherheitsanalysten eines Netzwerks benutzt werden, um Sicherheitsprobleme in Netzwerken zu finden. Wiederum kann es auch von Anwendungsentwicklern benutzt werden, um die Umsetzung von Netzwerkprotokollen auszutesten. Zuletzt kann es auch benutzt werden um mehr über das Netzwerk und den Netzwerkverkehr zu lernen und diesen zu analysieren. Es gibt natürlich auch viele weitere Möglichkeiten Wireshark zu benutzen. [6]

3 Netzwerk

3.1 Aufbau

3.2 Protokolle

3.3 Open System Interconnection Modell

Das Open System Interconnect[2] 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.

OSI-Modell		
Host Layers	7	Application
	6	Presentation
	5	Session
	4	Transport
Media Layers	3	Network
	2	Data Link
	1	Physical

Tabelle 1: OSI-Modell [10]

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.

Da diese Datenkommunikation relativ komplex ist, wurde das Modell in sieben Schichten eingeteilt. Die oberen vier Schichten gehören zu den Anwendungsori-

[2]dt: Offenes System für Kommunikationsverbindungen

entierten Schichten[3]. Die unteren drei Schichten werden Transport Schichten[4] 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 auffindbar 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[5] 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[6], welches benutzt wird, um Websites welche in HTML[7] geschrieben sind zu präsentieren, oder SMTP[8], welches benutzt wird um E-Mails zu präsentieren.

3.3.2 Präsentationsschicht

Die Präsentationsschicht[9] 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.

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

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.

3.3.3 Sitzungsschicht

Anschließend folgt die Sitzungsschicht[10]. 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[11] 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[12] und UDP[13]

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

Exkurs: UDP/TCP

3.3.5 Netzwerkschicht

Darauf folgt die Netzwerkschicht[14]. 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[15] genannt. Wenn sich beide Geräte im selben Netzwerk befinden, wird diese Ebene übersprungen.

Zu den Protokollen für diese Schicht gehören das IP[16], das ICMP[17], das IGMP[18] und die IPsec[19] Suite. Ein Beispielgerät wäre ein Router.

3.3.6 Sicherungsschicht

Die Sicherungsschicht[20] 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[21] 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

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

kann beispielsweise aus Kupfer oder Glasfaser bestehen und zwei innere Leitungen haben: eine Datenleitung und eine Steuerleitung. Bei der Übertragung über Funk wird beispielsweise durch Luft übertragen. Es muss ebenfalls die Übertragungsrichtung und Geschwindigkeit festgelegt werden. Ein Kabel kann in eine Richtung[22], abwechselnd in beide Richtungen[23] oder in beide Richtungen gleichzeitig[24] 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[25]. Zu beginn soll die Software getestet werden. Hierfür habe ich einen simplen Test erstellt.

4.1 Beispiel

In diesem Kapitel soll eine Beispielkommunikation vereinfacht dargestellt werden. Es wird ein lokaler Webserver erstellt und anschließend kontaktiert. Dieser Transfer wird dann durch Wireshark aufgezeichnet.

4.1.1 Der Webserver

Code:

```
from flask import Flask #importieren von den benoetigten Bibliotheken
import os
from datetime import datetime

hostName = "localhost" #setzen des Hostnamen
serverPort = 8080 #setzen des Ports (Erklaerung unter dem Code)
app = Flask(__name__) #der Flask Server wird instanziert
@app.route('/') #wenn zum index geroutet wird...
def index():
return '<p>willkommen in der http webserver demo</p>' #...soll diese
    response zurueckgeschickt werden
@app.route('/time') #wenn zu /time geroutet wird...
```

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

```
def time():
now = datetime.now()
current_time = now.strftime("%H:%M:%S") #erstellen der response
return f'Jetzige Zeit: {current_time}' #...soll diese response
    zurueckgeschickt werden

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

Mit der ‘flask‘Bibliothek können Webserver in Python leicht umgesetzt werden. Sie erlaubt es Webserver zu erstellen, die auf requests hören und mit dem angefragten Inhalt antworten[8]

Weitere Erklärung zum Code: “Ports sind ein Merkmal der Protokolle TCP und UDP“[12]. Diese werden benutzt, um den Datenverkehr zu ordnen. Da auf einem Server oft mehrere Programme aktiv sind, die gleichzeitig über das Netzwerk kommunizieren müssen diese eingeteilt werden, sodass die Pakete nicht durcheinander kommen. Jedes Programm legt seinen Port fest, bzw. bekommt vom Betriebssystem einen Port zugewiesen. Dieser kann zwischen 0 und 65535 liegen.[14]

Anschließend wird Wireshark geöffnet und das richtige Interface, welches aufgenommen werden soll, ausgewählt. In ‘diesem Fall ist es das ‘Loopback:io‘ interface. **(siehe: Anlagen, Das Wireshark interface, auswählen des Loopback Geräts)**

4.1.2 Das ‘loopback device‘

Das Loopback interace ist eine “Pseudo-Netzwerkschnittstelle zum gefahrenlosen Testen“[13]. Der Nertzwerktraffic wird in diesem Fall, da der Server auf ‘localhost‘live ist. Das loopback interface agiert als virtuelle Netzwerkkarte und ist nicht wirklich als Hardware vorhanden. Da ich diesen Test auf einem Linux Betriebssystem durchführe ist diese Funktion durch Wireshark unterstützt.[2]

4.1.3 Ergebnis

Nun muss der Server kontaktiert werden. Dies kann man mit einem Web-Browser machen. Mit Firefox habe ich dann ‘http://localhost:8080‘ kontaktiert. In Wire-

shark sind direkt mehrere neue Einträge zu erkennen. **(siehe: Anlagen, der Datentransfer wurde aufgezeichnet)**

Im nächsten Schritt wird das erste request, mit dem 'HTTP' Protocoll angesehen. **(siehe: Anlagen, Analyse des 'GET' requests)**. Unten rechts ist der Inhalt des Pakets zu erkennen. Darin sind einige Daten zum Browser enthalten und zum Inhalt, welcher angefragt wird. Diese Daten werden später in der Analyse weiter durchleuchtet.

Der Server antwortet mit diesem angeforderten Inhalt **(siehe: Anlagen, Analyse der response)**. Der Inhalt ist, der oben im Code festgelegte Satz, "Willkommen in der http Webserver demo"

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

Tabellenverzeichnis

Literatur

[1] The Wireshark Contributors. A brief history of wireshark. zuletzt besucht: 2. November 2023, Quelle: https://www.wireshark.org/docs/wsug_html_chunked/ChIntroHistory.html.

[2] The Wireshark Contributors. Loopback. Zuletzt besucht: 1. November 2023, Quelle: https://wiki.wireshark.org/CaptureSetup/Loopback#supported-platforms.

[3] The Wireshark Contributors. Open source software. Zuletzt besucht: 2. November 2023, Quelle: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#_open_source_software.

[4] The Wireshark Contributors. What is wireshark? Zuletzt besucht: 2. November 2023, Quelle: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#ChIntroWhatIs.

[5] The Wireshark Contributors. Wireshark - features. Zuletzt besucht: 15. Juni 2023, Quelle: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#ChIntroFeatures.

[6] The Wireshark Contributors. Wireshark - intended purposes. Zuletzt besucht: 16. Mai 2023, Quelle: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#ChIntroPurposes.

[7] The Wireshark Contributors. Wireshark logo. Quelle: https://de.wikipedia.org/wiki/Wireshark#/media/Datei:Wireshark_icon.svg.

[8] The flask Team. Flask 3.0.0. Zuletzt besucht: 2. November 2023, Quelle: https://pypi.org/project/Flask/.

[9] The gnu Foundation. Gnu general public license. Zuletzt besucht: 2. November 2023, Quelle: https://www.gnu.org/licenses/old-licenses/gpl-2.0.html.

[10] Luis Herzog. Osi-modell. Nachempfunden von https://en.wikipedia.org/wiki/OSI_protocols.

[11] Luis Herzog. Screenshots aus der wireshark software. Selbst erstellt https://luisherzog.de.

[12] Reinhard Schiedermeier. Fußnote [19]: Ports. Zuletzt besucht: 2. November 2023, Quelle: https://sol.cs.hm.edu/4129/html/356-ports.xhtml.

[13] Reinhard Schiedermeier. Loopback-device. Zuletzt besucht: 1. November 2023, Quelle: https://sol.cs.hm.edu/4129/html/353-loopbackdevice.xhtml.

[14] Reinhard Schiedermeier. Ports. Zuletzt besucht: 2. November 2023, Quelle: https://sol.cs.hm.edu/4129/html/356-ports.xhtml.

[15] 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

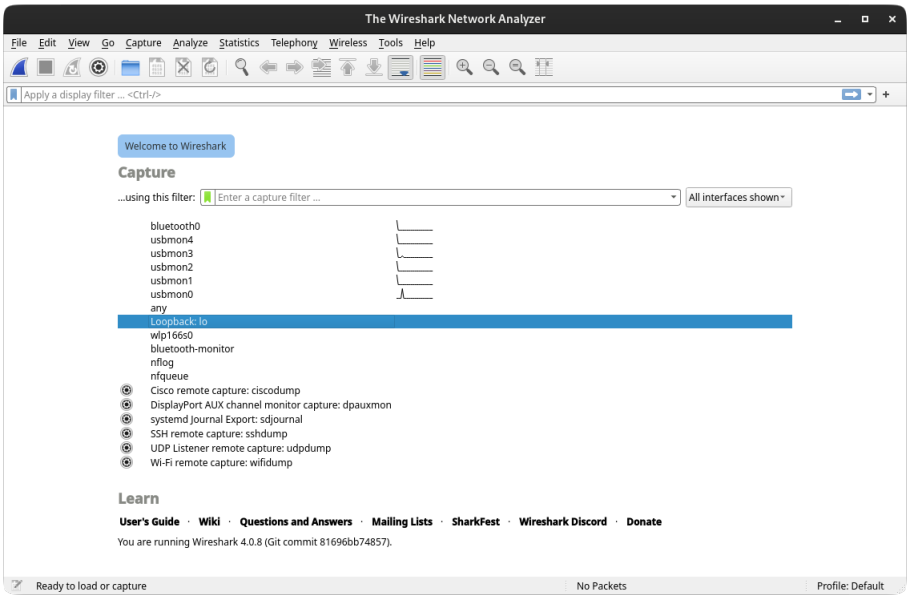

Abbildung 3: Das Wireshark interface, auswählen des Loopback Geräts [11]

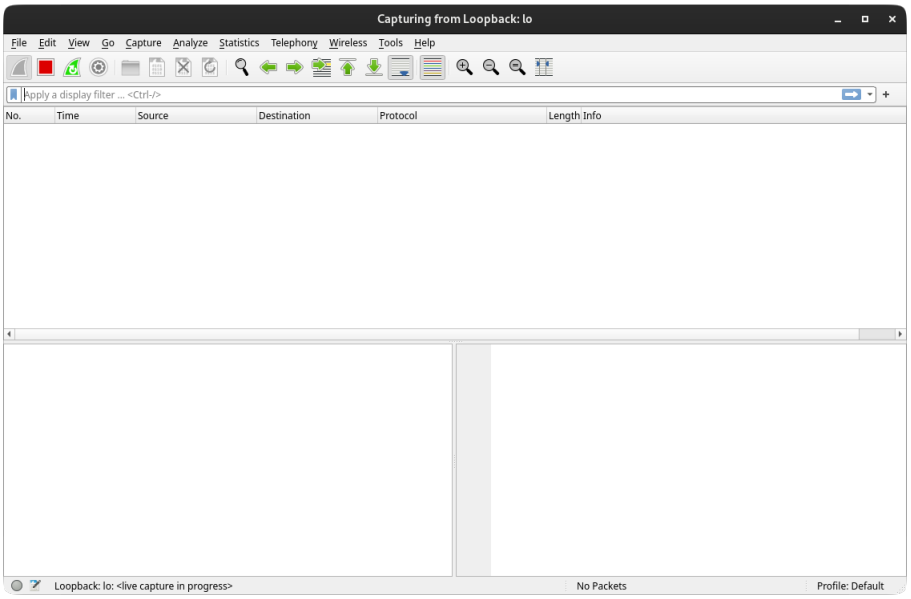

Abbildung 4: Das Wireshark interface, warten auf Datentransfer [11]

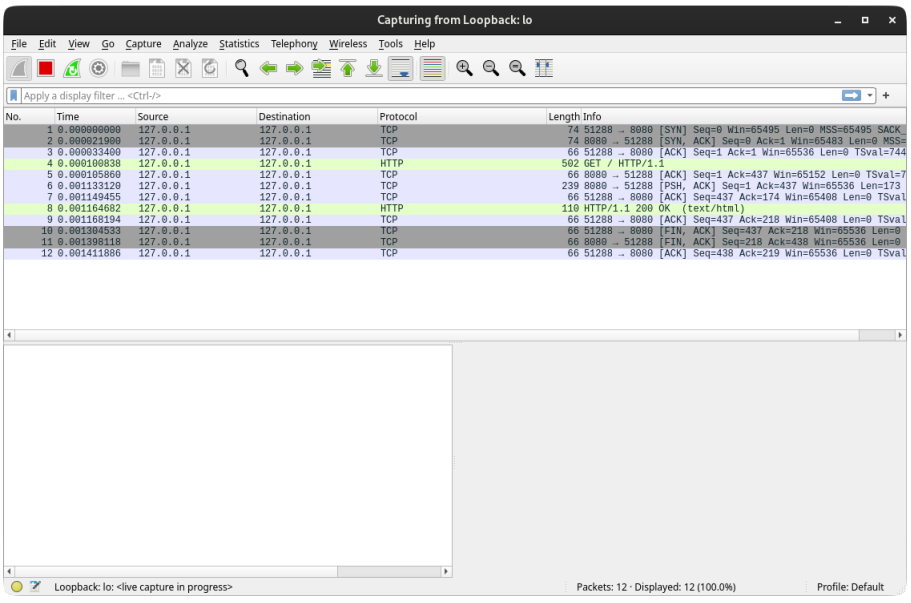

Abbildung 5: Das Wireshark interface, der Datentransfer wurde aufgezeichnet [11]

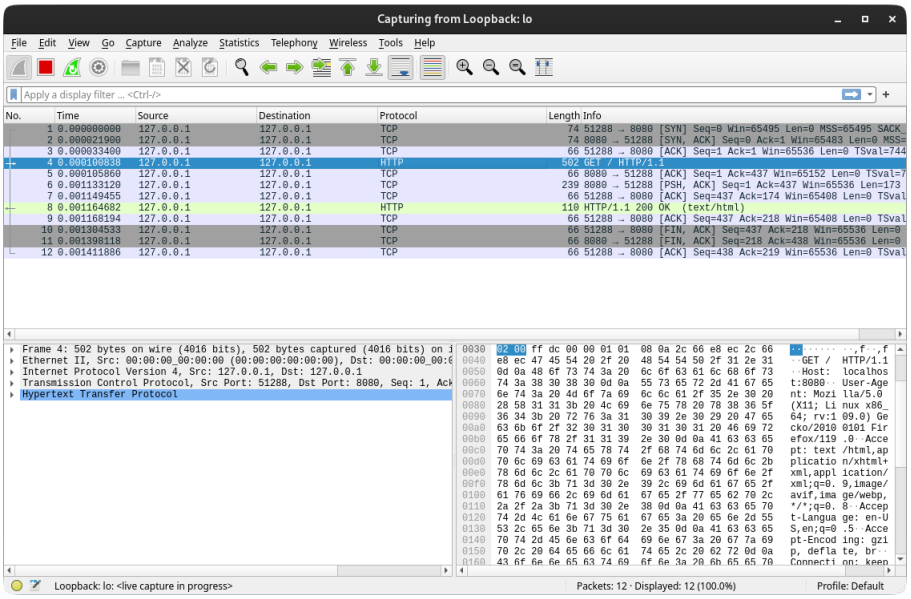

Abbildung 6: Das Wireshark interface, Analyse des ‘GET‘ requests [11]

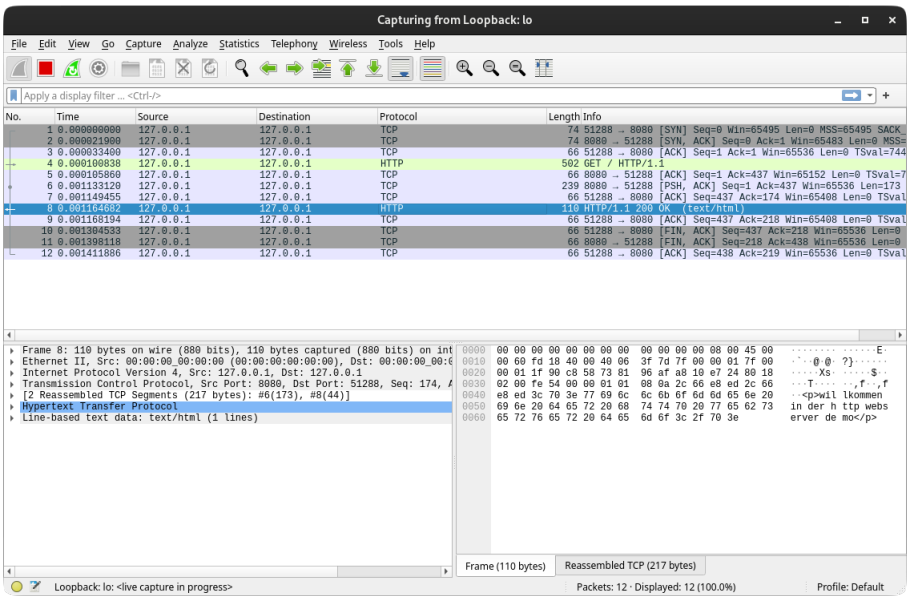

Abbildung 7: Das Wireshark interface, Analyse der response [11]