

Netzwerkanalyse mit Wireshark: Was passiert im Netzwerk?

Luis Herzog

5. November 2023



Abbildung 1: Wireshark Logo [27]

Inhaltsverzeichnis

1	Vorwort	3
2	Daten	3
2.1	Metadaten	3
3	Das Programm: Wireshark	4
3.1	Geschichte	4
3.2	Funktionsumfang	4
3.3	Benutzeroberfläche	5
3.4	Anwendungsbereiche	5
4	Netzwerk	6
4.1	Aufbau	6
4.2	Protokolle	7
4.3	Open System Interconnection Model	8
4.3.1	Anwendungsschicht	9
4.3.2	Präsentationsschicht	9
4.3.3	Sitzungsschicht	10
4.3.4	Transportschicht	10
4.3.5	Exkurs: UDP/TCP	11
4.3.6	Netzwerkschicht	13
4.3.7	Sicherungsschicht	13
4.3.8	Bitübertragungsschicht	13
5	Analyse	14
5.1	Beispiel	14
5.1.1	Webserver	14
5.1.2	Loopback Interface	15
5.1.3	Ergebnis	15
5.2	Browser	16
5.2.1	Firefox	16
6	Fazit	18
7	Anlagen	26

1 Vorwort

Die Digitalisierung ist in aller Munde. Durch die Digitalisierung wächst die Anzahl von Netzwerken, nicht nur in Deutschland, sondern weltweit. Netzwerke werden überall benutzt. Die meisten Menschen besitzen Heimnetzwerke. Aber es gibt noch viel mehr Netzwerke, zum Beispiel in Flughäfen, Cafés, Bibliotheken und noch so viel mehr. Diese Vielfalt von Netzwerken lässt einen Gedanken aufleuchten: „Wie funktionieren Netzwerke?“. Durch die Digitalisierung haben sich Netzwerke stark verändert und vor allem vergrößert. Dadurch sind Netzwerke oft sehr komplex. Die Wireshark Software erlaubt es, einen tieferen Blick in das Netzwerk zu werfen und Details herauszufinden. In dieser Arbeit wird die Arbeitsweise eines Netzwerks dargelegt und die Wireshark Software erklärt.

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

2 Daten

Statista, eine Website für Statistiken, beschreibt Daten als „Messwerte, die im Rahmen von Befragungen, Beobachtungen oder Experimenten erhoben werden“. [39] Der Duden gibt mehrere Definitionen. Die dritte Definition gibt Daten als „elektronisch gespeicherte Zeichen, Angaben, Informationen“ [14] an. Da es viele verschiedene Arten von Daten gibt, gibt es viele verschiedene Definitionen. Im Netzwerk beschreiben Daten den Inhalt und Aufbau von Paketen. Zum Inhalt gehören Informationen zum Paket und die zu vermittelnde Nachricht, welche alle durch elektronisch gespeicherte Zeichen dargestellt werden.

2.1 Metadaten

Metadaten spielen hierbei ebenfalls eine wichtige Rolle. Diese Daten sind die schon oben genannten Informationen über weitere Daten. Mit diesen Metadaten lässt sich ein zugehöriger Inhalt besser Einordnen, Katalogisieren und Analysieren. Metadaten sind strukturierte Daten und können bei physischen Gegenständen sowie bei digitalen Dateien Einsatz finden. Ein Beispiel für Metadaten ist der Autor bei einem Buch oder der Komponist bei einem Musikstück. [30] In der Netzwerktechnik sind Metadaten besonders wichtig. Diese sind im Adresskopf der Datenpakete gespeichert und enthalten beispielsweise Informationen über die

Quelladresse und Zieladresse des Pakets, sowie über die Lebensdauer des Pakets.[35]
Mehr hierzu wird später im Kapitel 4 unter ‚Exkurs: UDP/TCP‘ erklärt.

3 Das Programm: Wireshark

Heise.de bezeichnet Wireshark als „Schweizer Taschenmesser“[40] der Netzwerktechnik. Das Programm wird fast 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. Wireshark ist ein „network packet analyzer“[11]. Es erlaubt dem Benutzer den Netzwerkverkehr aufzunehmen und zu analysieren. Mit Wireshark kann man sozusagen in ein Netzkabel ‚hineinschauen‘.

3.1 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[20] herausgegeben [10]. 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 einer komplett neuen Benutzeroberfläche vorgestellt. Letztendlich wurde im Jahr 2023 die Wireshark Foundation¹ gegründet, welche jetzt das Programm unterstützt. [8]

3.2 Funktionsumfang

Wireshark 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 diese exportierten Netzwerk-dumps von Wireshark und anderen Programmen wieder importieren. Da Wireshark sowohl auf UNIX Plattformen als auch auf Windows verfügbar ist, geht dies Plattformübergreifend. Ebenso kann man die Datenpakete anhand verschiedener Kriterien filtern und durchsuchen. Die genannten Funktionen kratzen aber nur an der Oberfläche. Wireshark bietet noch viel mehr Funktionen.[12]

¹<https://wiresharkfoundation.org/>

3.3 Benutzeroberfläche

Das voreingestellte Layout ist im folgenden Screenshot (siehe Abbildung 2) dargestellt. In diesem Fall wird gerade die Netzwerkschnittstelle ‘wg-mullvad’ aufgezeichnet. Im oberen Teil ist die Symbolleiste zu sehen. Dort findet man Knöpfe zum Starten und Stoppen der Aufnahme. Rechts 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, die Quelle und das Ziel des Pakets abgelesen werden. Rechts daneben ist das genutzte Netzwerkprotokoll und die Länge des Pakets sichtbar. Ganz rechts stehen weitere Informationen zum Paket. Unter dieser Liste befinden sich zwei Fenster. Rechts steht die Rohdaten des Pakets. Links hingegen steht der von Wireshark automatisch ausgelesene und geordnete Inhalt. Hier kann jegliche Information zum Datenpaket gefunden werden. Unter diesem Fenster steht der Name der Netzwerkschnittstelle, deren Netzwerkverkehr aufgezeichnet wird.

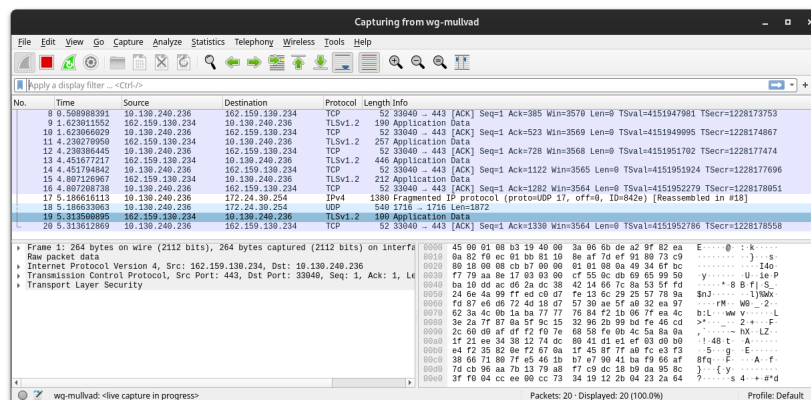


Abbildung 2: Beispielscreenshot aus der Wireshark Software [25]

3.4 Anwendungsbereiche

Wie schon im oberen Teil dargestellt, bietet Wireshark sehr viele Funktionen. Es kann in sehr vielen Situationen Anwendung finden. Es wird beispielsweise von Netzwerk Administratoren zum Finden, Analysieren und Lösen von Netzwerkproblemen benutzt. Das Programm kann auch von Sicherheitsanalysten benutzt werden, um Sicherheitsprobleme in Netzwerken zu finden. Von Anwendungsent-

wickeln wird Wireshark benutzt, um die Umsetzung von Netzwerkprotokollen auszutesten. Eine weitere Anwendung ergibt sich beim Lernen, über das Netzwerk durch Wireshark. Es gibt natürlich auch viele weitere Möglichkeiten das Programm zu benutzen.[13]

4 Netzwerk

In der Informationstechnologie ist ein Netzwerk ein Zusammenschluss mehrerer Computer [...], die untereinander kommunizieren, also Daten untereinander austauschen können.[28]

4.1 Aufbau

Es gibt viele verschiedene Arten von Netzwerken, die unterschiedlich aufgebaut sein können. Netzwerke werden in Topologien unterteilt. Durch diese Topologien können selbst sehr komplexe Netzwerke gut veranschaulicht werden. Es gibt fünf Haupttopologien.

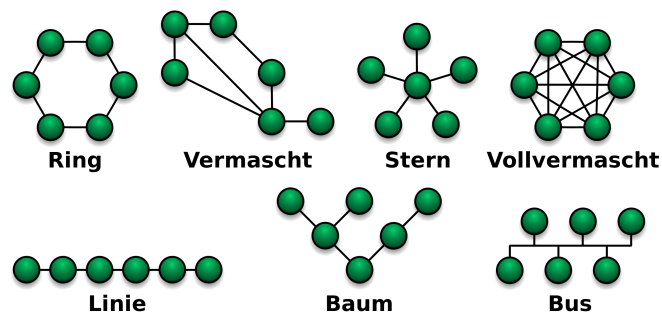


Abbildung 3: Die Netzwerktopologien [18]

Bei der Ringtopologie sind alle Knoten in einem Ring miteinander verbunden. Die Linientopologie ist ähnlich, mit dem Unterschied, dass die Knoten an den Enden nicht mehr direkt miteinander verbunden sind. Bei der Sterntopologie sind alle Knoten mit einem Hauptknoten verbunden. Die Baumtopologie ist eine erweiterte Form der Sterntopologie, bei der mehrere Knoten hierarchisch miteinander verbunden sind. Bei der Bustopologie hängen alle Knoten an einem Übertragungsmedium. Netzwerke können ebenfalls vermascht sein, was bedeutet, dass die Knoten mehr Verbindungen aufweisen, als theoretisch nötig wären. Bei einem vollvermaschten Netzwerk ist jeder Knoten mit allen weiteren Knoten verbunden.[29] Der Beispielaufbau in Abbildung 4 zeigt ein Netzwerk basierend auf der Sterntopologie.

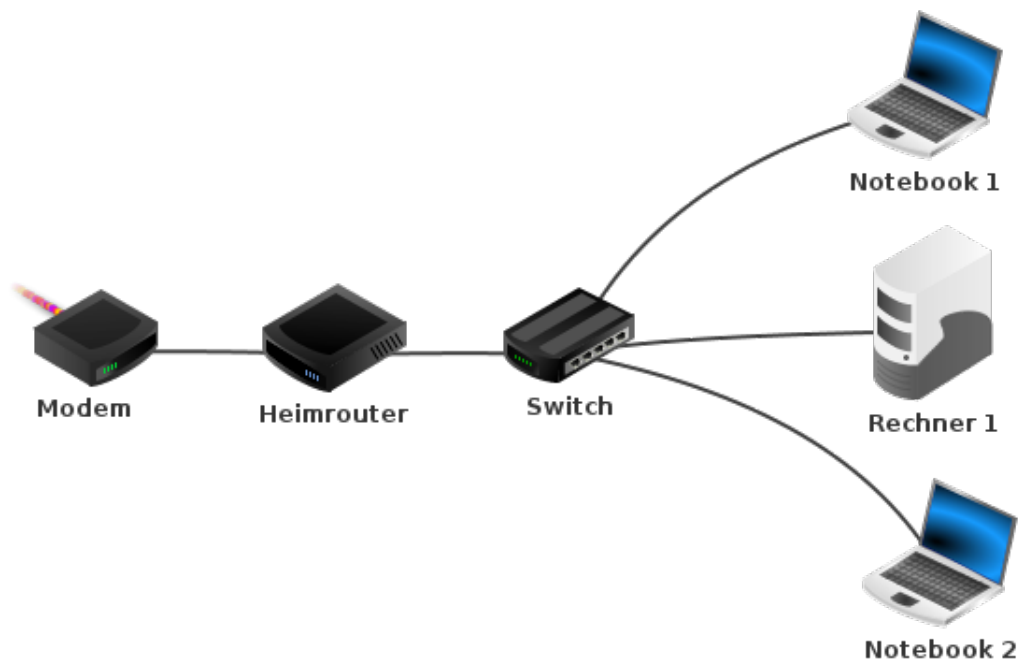


Abbildung 4: Aufbau eines Beispielnetzwerks [21]

4.2 Protokolle

Damit alle Komponenten im Netzwerk richtig miteinander kommunizieren können, wurden standardisierte Protokolle entwickelt. Somit sprechen alle Komponenten ‚die gleiche Sprache‘. Es gibt verschiedene Protokolle für verschiedene Zwecke. Beispielsweise gibt es HTTP für den Transfer von Websites oder FTP zum Senden von Dateien. Da es sehr viele Protokolle gibt und diese recht komplex aufgebaut sind, gibt es das OSI-Modell, um den Netzwerkverkehr mit Protokollen vereinfacht darzustellen. Im nächsten Abschnitt wird dieses Modell genauer beleuchtet.

4.3 Open System Interconnection Model

Das Open System Interconnect² Modell, auch OSI-Modell genannt, beschreibt die Voraussetzungen, die für eine Kommunikation innerhalb eines Netzwerks nötig sind. Dieses wurde 1983 von der ‚Internationalen Organisation für Normung‘ standardisiert. Diese Standardisierung ist notwendig, damit sich alle Komponenten im Netzwerk, auch wenn diese von verschiedenen Herstellern produziert wurden, reibungslos miteinander funk-

tionieren. Wenn ein Paket beispielsweise von einem Computer im Netzwerk 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 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 Standard geschaffen, mit dem alle Computersysteme arbeiten können.

Da diese Datenkommunikation relativ komplex ist, wurde das Modell in sieben Schichten eingeteilt. Die oberen vier Schichten gehören zu den anwendungsorientierten Schichten³. Die unteren drei Schichten werden Transportschichten⁴ genannt. Jede Schicht behandelt eine Anforderung, die für eine funktionierende Kommunikation erfüllt werden muss. Ein zu übertragendes Paket durchläuft vor der Versendung die Schichten 7 - 2, wobei dem Paket bei jeder Schicht Protokollinformationen hinzugefügt werden, die dann im Protokoll des Datenpaketes

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 [24]

²dt: Offenes System für Kommunikationsverbindungen

³engl: host layer

⁴engl: media layer

auffindbar sind. Die erste und letzte Schicht wandelt das Paket in technisch übertragbare Daten um und schickt dieses über das Übertragungsmedium weg. Das Medium kann hierbei ein Kabel sein oder aus einer Antenne bestehen. Auf der Empfängerseite wird dieser Prozess rückwärts durchgeführt. Hierbei wird die jeweilige Protokollinformation nach der Interpretierung durch die zuständige Schicht entfernt, bis der Inhalt des Pakets zum Vorschein kommt.[33]

Im Folgenden werden die einzelnen Schichten beleuchtet, um einen besseren Einblick in die Datenübertragung zu gewähren.

4.3.1 Anwendungsschicht

Die Anwendungsschicht⁵ stellt die Daten dar, mit welchen der Nutzer interagiert. Softwareanwendungen, wie Webbrowser und E-Mail-Clients stützen sich auf die siebte Schicht, um dem Nutzer aussagekräftige Daten zu präsentieren.

Hierzu gehören Protokolle, wie HTTP⁶, welches benutzt wird, um Websites welche in HTML⁷ geschrieben sind zu präsentieren, oder SMTP⁸, welches benutzt wird um E-Mails darzustellen.[5][34]

4.3.2 Präsentationsschicht

Die Präsentationsschicht⁹ 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.[5][34]

⁵engl: application layer

⁶Hyper Text Transfer Protocol

⁷Hyper Text Markup Language

⁸Simple Mail Transfer Protocol

⁹engl: presentation layer

4.3.3 Sitzungsschicht

Anschließend folgt die Sitzungsschicht¹⁰. 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 z.B. 230 Megabyte wieder aufgenommen werden, da es einen Checkpoint der Datei bei 230 Megabyte gibt.[5][34]

4.3.4 Transportschicht

In der Transportschicht¹¹ werden die Datenpakete vor dem Versenden in Segmente zerlegt. Im Empfangsgerät werden 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 TCP¹² und UDP¹³[5][34]

¹⁰engl: session layer

¹¹engl: transport layer

¹²Transmission Control Protocol

¹³User Datagram Protocol

4.3.5 Exkurs: UDP/TCP

Die zwei Protokolle UDP und TCP sind die meist verwendeten Protokolle, weswegen die Protokolle in den oberen Schichten darauf aufbauen.

TCP ist ein verbindungsorientiertes Protokoll. Der Client und der Server müssen erst eine Verbindung herstellen, damit Daten in beide Richtungen übertragen werden können. TCP verfügt außerdem über ein erhebliches System für die Fehlerkontrolle durch beispielsweise Prüfsummen.[35] Dadurch ist TCP ein sehr zuverlässiges, aber eher langsames Protokoll. Es wird benutzt, um Dateien, Websites oder Bilder zu übertragen.[2] In Tabelle 2 sind die Eigenschaften übersichtlich aufgefächert.

Eigenschaften	TCP
Verbindungsstatus	Erfordert eine bestehende Verbindung zur Datenübertragung
Datensequenzierung	Erfordert Sequenzierung
Garantierte Übertragung	Übertragung wird garantiert
Neuübertragung von verlorenen Paketen	Neuübertragung von verlorenen Paketen ist möglich
Fehlerüberprüfung	Umfangreiche Fehlerüberprüfung und Bestätigen der Daten
Geschwindigkeit	Langsamer als UDP
bestmögliche Verwendung	Streaming bei Videokonferenzen, VoIP oder Livestreams

Tabelle 2: TCP [1]

Der Verbindungsaufbau läuft wie folgt ab: Der Client sendet eine Anfrage zum Verbindungsaufbau¹⁴ zum Server. Wenn dies möglich ist, sendet der Server eine Bestätigung¹⁵ zurück und baut den Kanal auf. Jetzt können der Client und Server miteinander kommunizieren und die Daten können übertragen werden. Wenn ein Datenpaket fehlerhaft übertragen wird, wird es nochmal geschickt.[16] In Abbildung 5 ist dieser Datenverkehr in einem Datenflussdiagramm dargestellt.

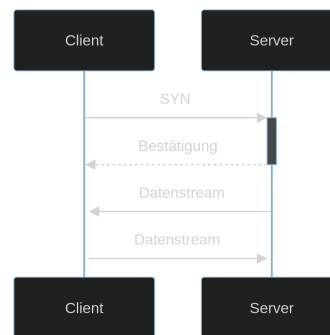


Abbildung 5: Datenflussdiagramm für das Protokoll TCP[22]

¹⁴SYN-Paket

¹⁵ACK-Paket

UDP ist ein simpleres und verbindungsloses Protokoll. Das System für Fehlerkontrolle ist minimal, weswegen die korrekte Datenübertragung nicht garantiert ist. Die Daten werden kontinuierlich zum Empfänger geschickt, auch wenn dieser diese nicht empfängt. Dies macht UDP nicht ideal für das Verschieken von Dateien, aber um so besser zum streamen. Da das Protokoll wesentlich schneller ist, kann es mit einer höheren Bandbreite arbeiten, wenn diese nötig ist. Eine Fehlerkontrolle ist bei Streams aufgrund der Datenmenge nicht möglich. UDP ist somit Ideal für das Umsetzen von Videocalls, VoIP¹⁶ oder das Livestreamen auf Videoportalen oder im Fernsehen.[3] Die genauen Eigenschaften sind in Tabelle 3 dargestellt.

Eigenschaften	UDP
Verbindungsstatus	Verbindungslos, keine weiteren Anforderungen
Datensequenzierung	Keine Sequenzierung
Garantierte Übertragung	Keine garantierte Übertragung
Neuübertragung von verlorenen Paketen	Keine Neuübertragung von verlorenen Paketen
Fehlerüberprüfung	Minimale Fehlerberprüfung durch Checksums
Geschwindigkeit	Schneller als TCP
bestmögliche Verwendung	Schicken von Dateien, Email, Bilder, Websites, etc.

Tabelle 3: UDP [1]

Der Verbindungsaufbau läuft wie folgt ab: Der Client schickt dem Server ein Anfrage, in welcher die Öffnen des Streams erbitet wird. Daraufhin antwortet der Server direkt mit dem Datenstream. Dieser Stream bleibt geöffnet, auch wenn der Client den Datenstream nicht mehr aufnimmt. Der Server muss den Transfer eigenständig stoppen, wenn Bandbreite gespart werden muss. In Abbildung 6 ist der Datenverkehr in einem Datenflussdiagramm dargestellt.[15]

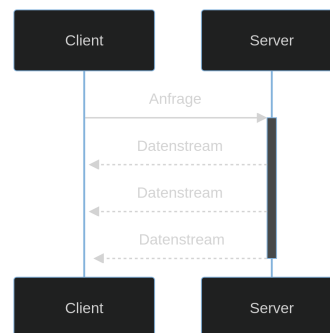


Abbildung 6: Datenflussdiagramm für das Protokoll UDP[23]

¹⁶Voice over IP

4.3.6 Netzwerkschicht

Auf die Transportschicht folgt die Netzwerkschicht¹⁷. 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 genannt.[4] Wenn sich beide Geräte im selben Netzwerk befinden, wird diese Ebene übersprungen.

Zu den Protokollen für diese Schicht gehören das IP¹⁸, das ICMP¹⁹, das IGMP²⁰ und die IPsec²¹. Ein Beispielgerät, welches in dieser Schicht agiert, wäre ein Router.[5][34]

4.3.7 Sicherungsschicht

Die Sicherungsschicht²² 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 sind Bridges und Switches.[5][34]

4.3.8 Bitübertragungsschicht

Die unterste Schicht wird durch die Bitübertragungsschicht²³ dargestellt. Hier sind die Daten als Bitstrom vorhanden, eine Zeichenkette bestehend aus Einsen und Nullen. Hier muss sich auf mehrere Eigenschaften 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

¹⁷engl: network layer

¹⁸Internet Protocol

¹⁹Internet Control Message Protocol

²⁰Internet Group Message Protocol

²¹Internet Protocol Security

²²engl: Data Link Layer

²³engl: physical layer

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 Richtung²⁴, abwechselnd in beide Richtungen²⁵ oder in beide Richtungen gleichzeitig²⁶ übertragen.[5][34]

5 Analyse

In der Analyse soll häufig benutzte Software getestet werden. Ich habe mich für einen Webbrowser entschieden, da diese am meisten benutzt werden.[31] Zu Beginn soll die Wireshark-Installation getestet werden. Hierfür habe ich einen simplen Test erstellt.

5.1 Beispiel

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

5.1.1 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 instanziiert
@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...
def time():
```

²⁴simplex

²⁵halb-duplex

²⁶duplex

```
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 der Programmiersprache Python leicht umgesetzt werden. Sie erlaubt es Webserver zu erstellen, die auf Anfragen hören und mit dem angefragten Inhalt antworten können.[17]

Weitere Erklärung zum Code: „Ports sind ein Merkmal der Protokolle TCP und UDP“[36]. 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.[38]

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

5.1.2 Loopback Interface

Das ‘Loopback Interface‘ ist eine „Pseudo-Netzwerkschnittstelle zum gefahrenlosen Testen“[37]. Der Netzwerkverkehr wird in diesem Fall im genannten Netzwerkinterface zu sehen sein, 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, wird diese Funktion durch Wireshark unterstützt.[9]

5.1.3 Ergebnis

Nun muss der Server kontaktiert werden. Dies kann man mit einem Webbrowser 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 die erste Anfrage, mit dem ‚HTTP‘ Protokoll 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“

5.2 Browser

In dieser Analyse wird hauptsächlich auf das DNS²⁷ Protokoll geschaut. Dieses Protokoll erlaubt dem Client mit dem ‚Domain Name System‘ zu kommunizieren. Dieses System besteht aus Servern, die Hostnamen in IP-Adressen umwandeln können. Hostnames sind die von Menschen erkennbaren Adressen von Websites, wie ‚google.com‘ oder ‚luisherzog.de‘. IP-Adressen sind die von Rechnern verstehbaren Adressen von Servern, wie z.B. ‚192.169.11.4‘ oder ‚115.234.34.2‘. Das DNS-System enthält eine Liste dieser Hostnames mit den dazugehörigen IP-Adressen. Wenn wir beispielsweise mit dem Browser eine Website besuchen wollen, wird durch das DNS-Protokoll eine Anfrage an das DNS-System mit der URL²⁸ geschickt. In der URL ist der Hostname des Servers enthalten. Das DNS-System sucht in der Liste und leitet das Datenpaket auf den Weg zum Zielserver.[6] Im Netzwerk wird diese Aufgabe durch den Router übernommen.

Durch das Analysieren des DNS-Verkehrs, kann man herausfinden, welche Server kontaktiert werden. Der Dateninhalt des Pakets ist oft verschlüsselt und somit unlesbar, was eine tiefere Analyse der verschickten Daten unmöglich macht.

Ich habe mich beispielsweise mit dem Firefox Browser auseinandergesetzt. Die meisten Browser agieren im Netzwerkverkehr ähnlich.

5.2.1 Firefox

Beim Öffnen von Firefox werden mehrere Anfragen an das DNS-System getätigt.

²⁷Domain Name System

²⁸Uniform Resource Locator

1. Firefox macht die IP-Adresse des Routers im Heimnetzwerk ausfindig.
2. Firefox schaut, ob ein Captive Portal vorhanden ist. „Ein Captive Portal ist eine Webseite, die der Benutzer eines öffentlich zugänglichen Netzwerks ansehen und mit ihr interagieren muss, bevor der Zugang gewährt wird.“[7] Diese sind oft in öffentlichen Plätzen, wie Flughäfen, Cafés oder in Schulnetzwerken verwendet. Dies erzielt Firefox mit dem Kontaktieren der Adresse ‚detectportal.firefox.com‘. Hierbei testet Firefox ebenfalls, ob das Netzwerk andere Technologien, wie IPv6 unterstützt.[32]
3. Die Adresse ‚api.accounts.firefox.com‘ wird von Firefox kontaktiert. Dies liegt höchstwahrscheinlich daran, dass ich einen Firefox Account benutze und dadurch automatisch angemeldet werde.
4. Diese Anfrage ist kontroverser zu betrachten. Die Anfrage geht an ‚incoming.telemetry.mozilla.org‘. Was hier genau verschickt wird, ist unbekannt. Der Adresse nach zu urteilen, handelt es sich hier um Messungsdaten, die Firefox aufnimmt und an die Entwickler schickt, um Firefox zu verbessern. Welche Daten das genau sind und ob diese Daten ggf. die Privatsphäre der Nutzer beeinträchtigen, kann aus Wireshark nicht ausgelesen werden, da der Inhalt verschlüsselt ist.
5. Firefox kontaktiert die Adresse ‚push.services.mozilla.com‘. Hier wird der Adresse nach zu urteilen, nach Push-Benachrichtigungen gesucht, die noch zugestellt werden müssen.
6. ‚token.services.mozilla.com‘ ist die nächste Adresse, die kontaktiert wird. Hierbei geht es um die Sync Funktionalität eines Firefox Accounts. Firefox schaut hier, ob sich der synchronisierte Inhalt innerhalb eines Firefox Accounts verändert hat. Dies betrifft beispielsweise Tabs, Lesezeichen und zuletzt besuchte Websites. Diese Anfrage ist gut dokumentiert und erlaubt es somit, diese Funktion auch selbst zu hosten.[19] Bei mir wurde es unter ‚prod.tokenserver.prod.cloudops.mozgcp.net‘ gehostet, was die nächste Adresse war, die kontaktiert wurde.
7. Zuletzt wird den Extensions erlaubt, Anfragen zu stellen. Davor wird noch die Adresse ‚services.addons.mozilla.org‘ kontaktiert, wahrscheinlich um nach updates für die jeweiligen Extensions zu schauen.

Hierbei muss beachtet werden, dass Firefox noch viel mehr Anfragen stellt. Diese gehen zumeist an Zertifikatstationen, um die Sicherheit der Verbindung zu gewähren. Ebenso werden Adressen auch mehrfach kontaktiert, was in der oberen Liste nicht mit inbegriffen ist. Die Abbildung 7 zeigt den Umfang der Anfragen.

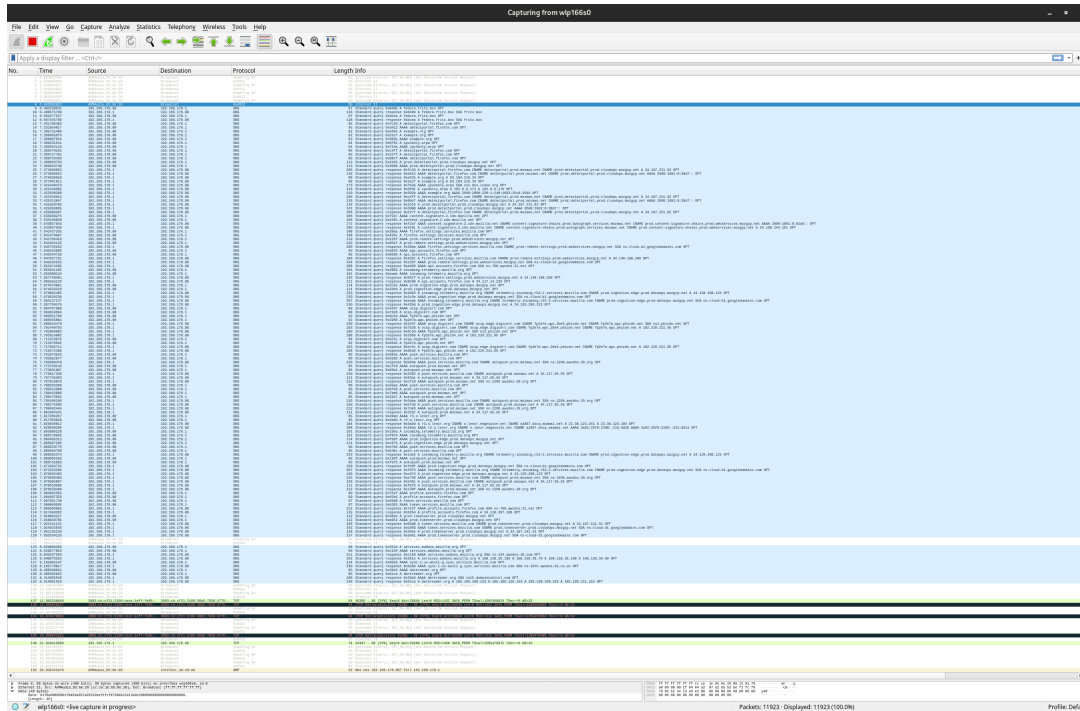
The image is a screenshot of the Wireshark network traffic analysis tool. The main pane displays a list of captured packets, with the 'Length' column highlighted in blue. The packets are primarily HTTP requests, with many of them being GET requests to various domains. The status bar at the bottom indicates that 11923 packets have been displayed out of a total of 11923 captured. The interface includes a menu bar at the top, a toolbar, and a packet list pane on the left.

Abbildung 7: Firefox tätigt sehr viele Anfragen[25]

6 Fazit

Dieses Thema hat mir sehr viel gezeigt. Mir war vorher nicht klar, wie viel wirklich im Netzwerk vor sich geht. Sobald man die Wireshark Software öffnet, wird man direkt mit mehreren Anfragen überschwemmt. Ich finde es trotzdem bemerkenswert, wie der Datenverkehr durch die ganzen oben genannten Schritte optimiert und effizienter gestaltet wird. Jedes einzelne Konzept ist sehr durchdacht und für einen bestimmten Zweck optimiert. Ich finde es trotz dessen schwierig auf Frage „Was passiert im Netzwerk?“ zu antworten. Wie in dieser Arbeit festgestellt gibt es viele verschiedene Arten von Netzwerken mit verschiedenen Verwendungszwecken. Jedes Netzwerk bietet somit unterschiedliche Voraussetzungen, die zu verschiedenen Ergebnissen führen. Die Frage, was wirklich übertragen wird, spielt in diesem Sinne ebenfalls eine große Rolle. Bevor ich zu diesem Thema

recherchiert habe, war ich den Metadaten in Protokollheadern negativ gegenüber gestanden. Ich habe diese immer als Tracker, zum Sammeln von personenbezogenen Daten angesehen. Jetzt weiß ich mehr. Mir ist klar geworden, dass der Datenverkehr ohne diese Metadaten nicht funktioniert. Spongebob aus der Kinderserie „Spongebob Schwammkopf“ wollte das Wasser vorher auch nicht bekommen. Zum Schluss fällt allerdings auf, dass er ohne Wasser ebenso wenig funktioniert, wie der Datenverkehr ohne Metadaten.



Abbildung 8: Spongebob funktioniert nicht ohne Wasser[26]

Abbildungsverzeichnis

1	Wireshark Logo [27]	1
2	Beispielscreenshot aus der Wireshark Software [25]	5
3	Die Netzwerktopologien [18]	6
4	Aufbau eines Beispielnetzwerks [21]	7
5	Datenflussdiagramm für das Protokoll TCP[22]	11
6	Datenflussdiagramm für das Protokoll UDP[23]	12
7	Firefox tätigt sehr viele Anfragen[25]	18
8	Spongebob funktioniert nicht ohne Wasser[26]	19
9	Das Wireshark interface, auswählen des Loopback Geräts [25] . .	27
10	Das Wireshark interface, warten auf Datentransfer [25]	28
11	Das Wireshark interface, der Datentransfer wurde aufgezeichnet[25]	28
12	Das Wireshark interface, Analyse des ‘GET‘ requests [25]	29
13	Das Wireshark interface, Analyse der response [25]	29

Tabellenverzeichnis

1	OSI-Modell [24]	8
2	TCP [1]	11
3	UDP [1]	12

Literatur

- [1] Pauline Ashenden. TPC vs. UDP: What's the Difference? Zuletzt besucht: 4. November 2023, Tabelle inspiriert, Quelle: <https://www.livesize.com/blog/tcp-vs-udp/>.
- [2] Pauline Ashenden. What ist TCP? Zuletzt besucht: 3. November 2023 Quelle: <https://www.livesize.com/blog/tcp-vs-udp/>.
- [3] Pauline Ashenden. What ist UDP? Zuletzt besucht: 3. November 2023 Quelle: <https://www.livesize.com/blog/tcp-vs-udp/>.
- [4] Cloudflare.de. Was ist Routing? — IP-Routing. Zuletzt besucht: 3. November 2023, Quelle: <https://www.cloudflare.com/de-de/learning/network-layer/what-is-routing/>.
- [5] Cloudflare.de. What are the 7 layers of the OSI Model? Zuletzt besucht: 5. November 2023, Quelle: <https://www.cloudflare.com/de-de/learning/ddos/glossary/open-systems-interconnection-model-osi/>.
- [6] Cloudflare.de. Wie funktioniert DNS? Zuletzt besucht: 4. November 2023, Quelle: <https://www.cloudflare.com/de-de/learning/dns/what-is-dns/>.
- [7] Computerweekly.de. Captive Portal. Zuletzt besucht: 4. November 2023, Quelle: <https://www.computerweekly.com/de/definition/Captive-Portal>.
- [8] The Wireshark Contributors. A Brief History of Wireshark. Zuletzt besucht: 2. November 2023, Quelle: https://www.wireshark.org/docs/wsug_html_chunked/ChIntroHistory.html.
- [9] The Wireshark Contributors. Loopback. Zuletzt besucht: 1. November 2023, Quelle: <https://wiki.wireshark.org/CaptureSetup/Loopback#supported-platforms>.
- [10] 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.
- [11] The Wireshark Contributors. What is Wireshark? Zuletzt besucht: 2. November 2023, Quelle: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#ChIntroWhatIs.

- [12] The Wireshark Contributors. Wireshark - Features. Zuletzt besucht: 15. Juni 2023, Quelle: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#ChIntroFeatures.
- [13] The Wireshark Contributors. Wireshark - Intended purposes. Zuletzt besucht: 16. Mai 2023, Quelle: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#ChIntroPurposes.
- [14] Duden.de. Daten. Zuletzt besucht: 3. November 2023, Quelle: <https://www.duden.de/rechtschreibung/Daten>.
- [15] Elektronik-kompendium.de. Funktionsweise von UDP. Zuletzt besucht: 4. November 2023, Quelle: <https://www.elektronik-kompendium.de/sites/net/0812281.htm>.
- [16] Elektronik-kompendium.de. Verbindungsaufbau (Connection Establishment). Zuletzt besucht: 4. November 2023, Quelle: <https://www.elektronik-kompendium.de/sites/net/2009211.htm>.
- [17] The flask Team. Flask 3.0.0. Zuletzt besucht: 2. November 2023, Quelle: <https://pypi.org/project/Flask/>.
- [18] Predatorix Foobaz, Parzi. Netzwerktopologien. Zuletzt besucht: 3. November 2023, Quelle: <https://commons.wikimedia.org/wiki/File:NetzwerkTopologien.svg>.
- [19] Mozilla Foundation. SyncStorage API v1.5. Zuletzt besucht: 4. November 2023, Quelle: <https://mozilla-services.readthedocs.io/en/latest/storage/apis-1.5.html>.
- [20] The gnu Foundation. Gnu General Public License. Zuletzt besucht: 2. November 2023, Quelle: <https://www.gnu.org/licenses/old-licenses/gpl-2.0.html>.
- [21] Luis Herzog. Abbildung eines Beispielnetzwerks. Erstellt mit Filius.
- [22] Luis Herzog. Datenflussdiagramm TCP. Zuletzt besucht: 3. November 2023, Nachempfunden von: Abbildung 1 von <https://www.geeksforgeeks.org/differences-between-tcp-and-udp/> Erstellt mit revealjs.

- [23] Luis Herzog. Datenflussdiagramm UDP. Zuletzt besucht: 3. November 2023, Nachempfunden von: Abbildung 2 von <https://www.geeksforgeeks.org/differences-between-tcp-and-udp/> Erstellt mit revealjs.
- [24] Luis Herzog. Osi-Modell. Zuletzt besucht: 3. November 2023, Tabelle inspiriert, Quelle: https://en.wikipedia.org/wiki/OSI_protocols.
- [25] Luis Herzog. Screenshots aus der Wireshark Software. Selbst erstellt.
- [26] Stephen Hillenburg. ‘I don’t need it’. Zuletzt besucht: 4. November 2023, Coverbild von <https://tuna.voicemod.net/sound/d745ae33-f574-44fa-90fb-6627489f90c4>.
- [27] Ktdryer. Wireshark Logo. Zuletzt besucht: 3. November 2023, Quelle: https://de.wikipedia.org/wiki/Wireshark#/media/Datei:Wireshark_icon.svg.
- [28] kurthelectronic.de. Was ist ein Netzwerk? Zuletzt besucht: 3. November 2023, Quelle: <https://www.kurthelectronic.de/fachwissen/wie-funktioniert-ein-it-netzwerk/>.
- [29] kurthelectronic.de. Welche Topologien können zum Verbinden genutzt werden? Zuletzt besucht: 3. November 2023, Quelle: <https://www.kurthelectronic.de/fachwissen/wie-funktioniert-ein-it-netzwerk/>.
- [30] Dipl. Ing. (FH) Stefan Luber. Was sind Metadaten? Zuletzt besucht: 3. November 2023, Quelle: <https://www.security-insider.de/was-sind-metadaten-a-358dc75ad6978ece7b58d89b045f5ec6/>.
- [31] Joerg Geiger Michael Humpa. Ziemlich sicher auf Ihrem System: Die beliebtesten Programme der Welt. Zuletzt besucht: 3. November 2023, Quelle: https://www.chip.de/news/Ziemlich-sicher-auch-auf-Ihrem-System-Die-beliebtesten-Programme-der-Welt_111539116.html.
- [32] Joni Mark Heijl Valentin k-alex meleansgar03 Dave Rose Michele Rodaro, Mozinet. Captive portal detection. Zuletzt besucht: 4. November 2023, Quelle: <https://support.mozilla.org/en-US/kb/captive-portal>.
- [33] Netzwerke.com. OSI-Schichten-Modell. Zuletzt besucht: 4. November 2023, Quelle: <https://www.netzwerke.com/OSI-Schichten-Modell.htm>.

- [34] Netzwerke.com. Schichten - Details. Zuletzt besucht: 5. November 2023, Quelle: <https://www.netzwerke.com/OSI-Schichten-Modell.htm>.
- [35] Netzwerke.com. TCP/IP – Transmission Control Protocol / Internet Protocol. Zuletzt besucht: 3. November 2023, Quelle: <https://www.netzwerke.com/TCP-IP.htm>.
- [36] Reinhard Schiedermeier. Fußnote [19]: Ports. Zuletzt besucht: 2. November 2023, Quelle: <https://sol.cs.hm.edu/4129/html/356-ports.xhtml>.
- [37] Reinhard Schiedermeier. Loopback-Device. Zuletzt besucht: 1. November 2023, Quelle: <https://sol.cs.hm.edu/4129/html/353-loopbackdevice.xhtml>.
- [38] Reinhard Schiedermeier. Ports. Zuletzt besucht: 2. November 2023, Quelle: <https://sol.cs.hm.edu/4129/html/356-ports.xhtml>.
- [39] Statista.de. Definition Daten. Zuletzt besucht: 3. November 2023, Quelle: <https://de.statista.com/statistik/lexikon/definition/42/daten/>.
- [40] 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>.

Ich erkläre hiermit, dass ich die Seminararbeit ohne fremde Hilfe
angefertigt und nur die im Literaturverzeichnis angeführten
Quellen und Hilfsmittel benützt habe.

Ort, Datum, Unterschrift des Verfassers

7 Anlagen

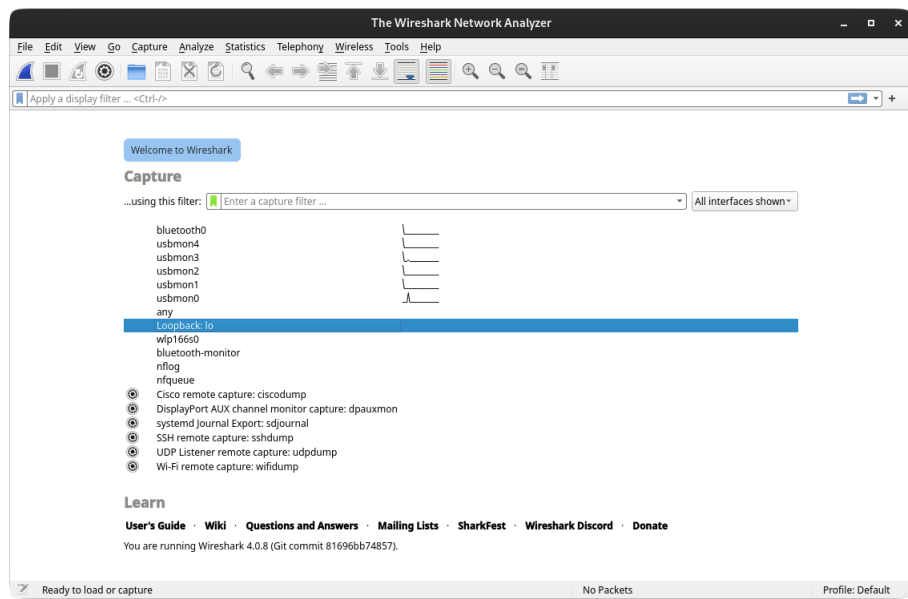


Abbildung 9: Das Wireshark interface, auswählen des Loopback Geräts [25]

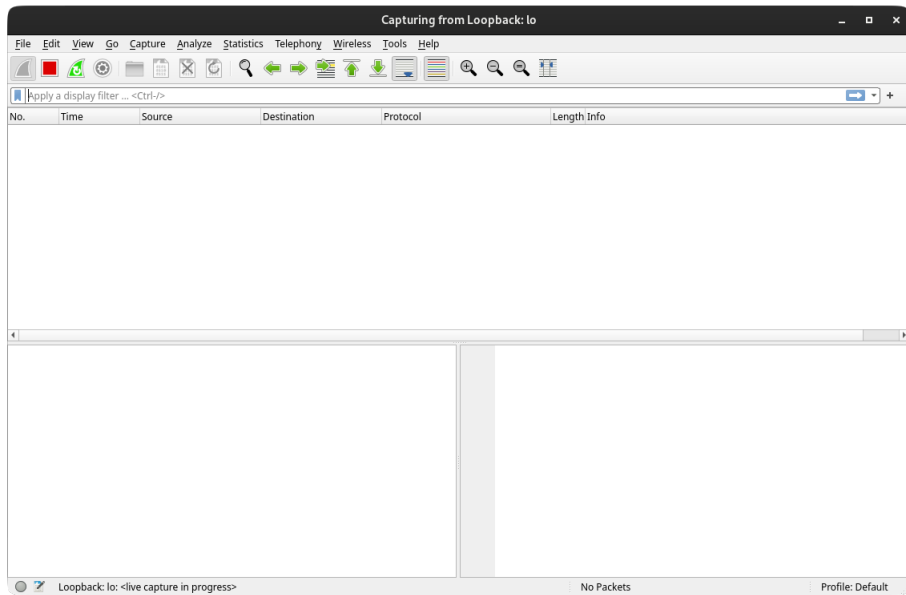


Abbildung 10: Das Wireshark interface, warten auf Datentransfer [25]

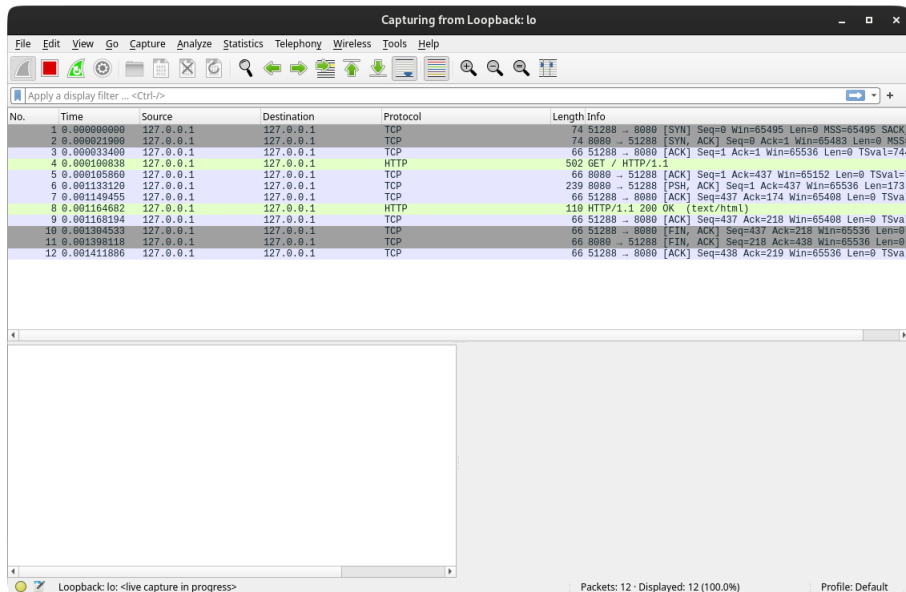


Abbildung 11: Das Wireshark interface, der Datentransfer wurde aufgezeichnet[25]

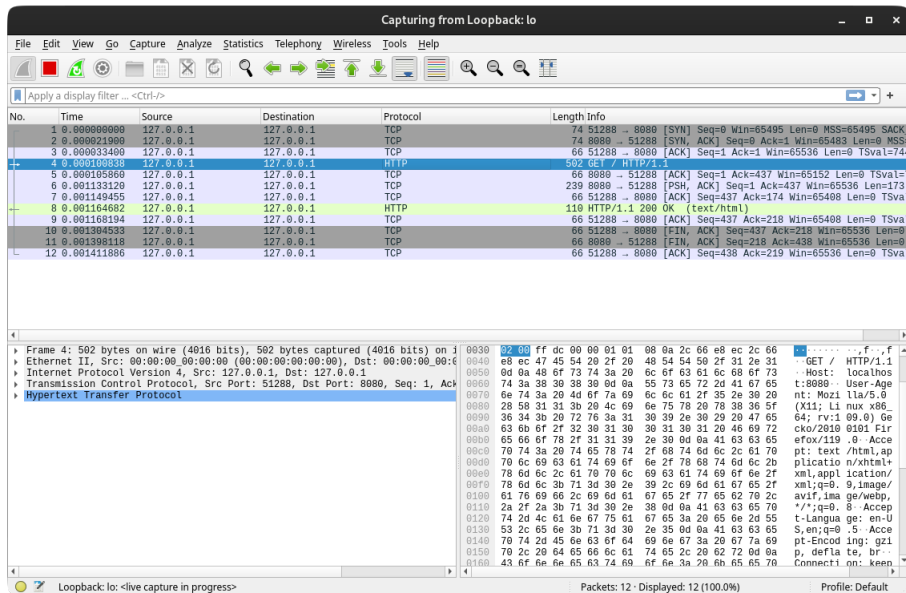


Abbildung 12: Das Wireshark interface, Analyse des ‘GET’ requests [25]

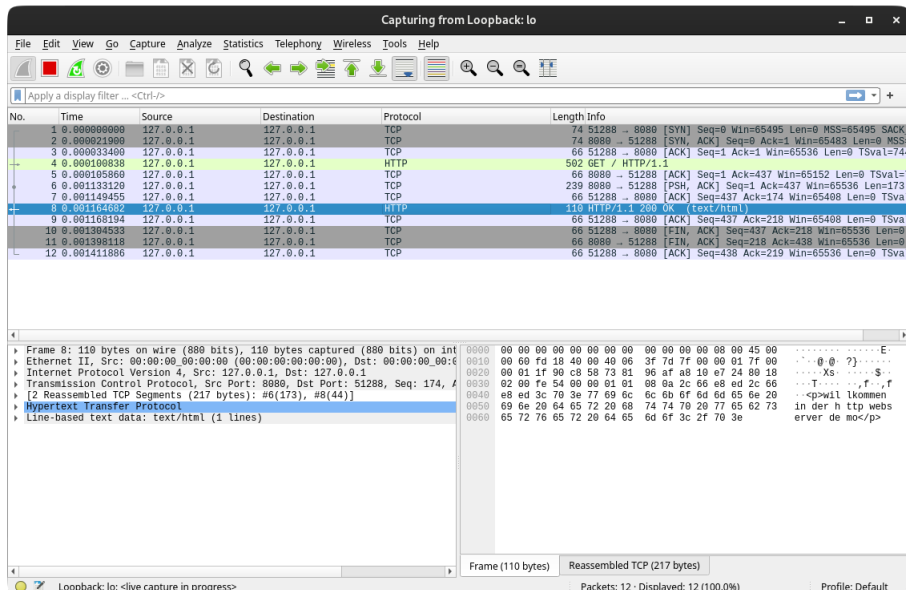


Abbildung 13: Das Wireshark interface, Analyse der response [25]