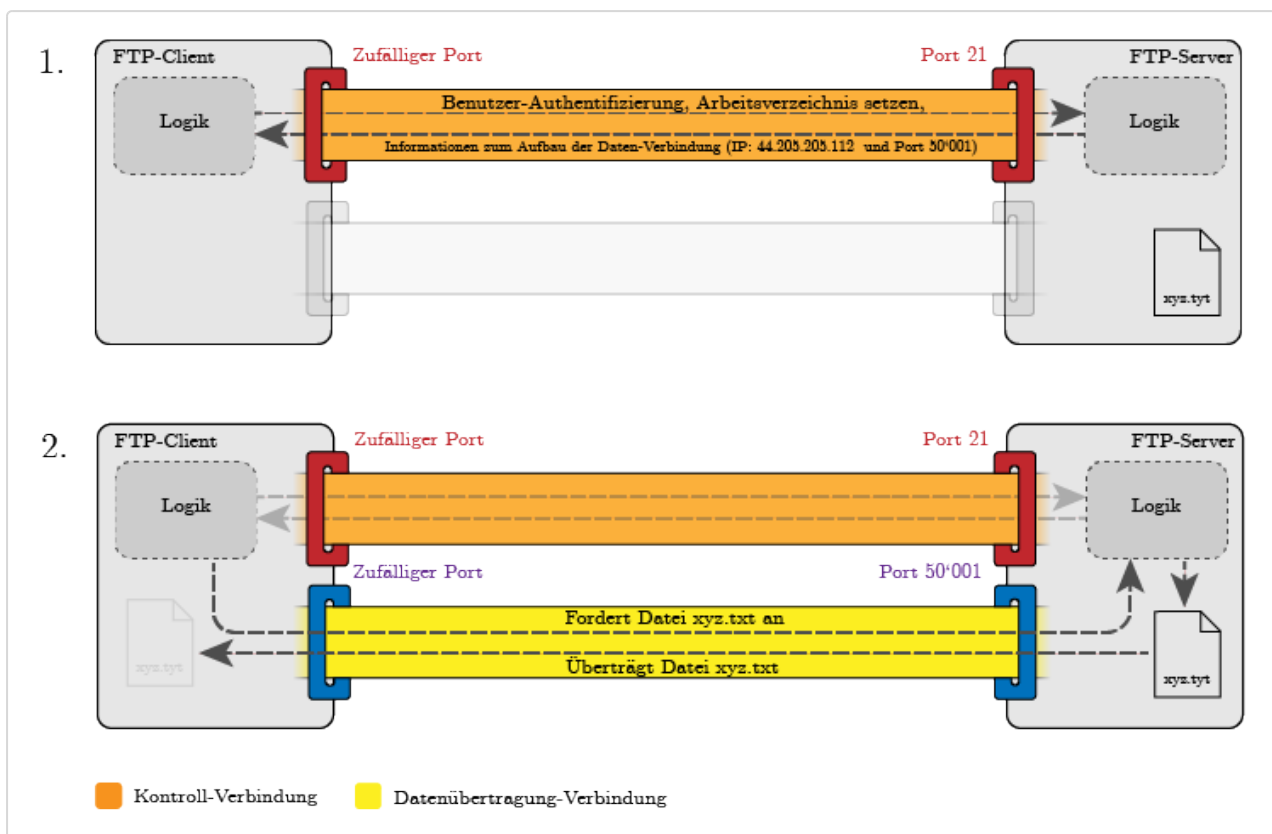


FTP

Sie haben nun bisher das Protokoll TFTP mit einem TFTP-Server und TFTP-Client kennen gelernt. Als nächstes schauen wir uns nun das FTP-Protokoll (File Transfer Protocol) an. Dieses wurde im Jahr 1971 entwickelt und ist ein wesentlicher Bestandteil des Internets und der Computervernetzung. Sie werden feststellen, dass FTP einen wesentlich grösseren Funktionsumfang bietet als TFTP, dazugehörten beispielsweise:

- Benutzer-Authentifizierung
- Verschlüsselung des Datenübertragung
- Berechtigungen
- Auflisten des Ordnerinhaltes

Im Gegensatz zu TFTP verwendet FTP zwei Verbindungen, einerseits eine Verbindung für die Übermittlung der Kontroll-Befehle (Control-Plane) und andererseits eine Verbindung für die Datenübertragung (Data-Plane). TFTP hingegen verwendet nur eine Verbindung, die sowohl für die Kontrolle als auch für die Datenübertragung verwendet wird.



Erklärung: Der Verbindungs-Aufbau erfolgt über die Kontroll-Verbindung, welche in der Grafik orange dargestellt ist. Darüber läuft die Authentifizierung, die Auswahl des korrekten Arbeits-Verzeichnisses sowie der Austausch der Verbindungsinformationen für die Dateiübertragung. Auf dieser Verbindung kommunizieren der Client und der Server ausschliesslich über Text-Kommandos.

Die eigentliche Datenübertragung erfolgt auf einem separaten Kanal, der sogenannten Daten-Verbindung. Dies bringt folgende Vorteile mit sich:

- Es können mehrere Datenübertragungen parallel stattfinden, für jede Datenübertragung wird eine separate Daten-Verbindung geöffnet. Jede Datenverbindung hat einen eigenen Port.
- Die Kommunikation mit dem Server wird nicht unterbrochen durch eine Datenübertragung. Dauert z. B. der Download einer Datei sehr lange, kann die Kommunikation über die Kontroll-Verbindung weiterhin vorgenommen werden. Würde nur eine Verbindung für die Kommunikation und die Datenübertragung verwendet werden, müsste gewartet werden, bis die Datei fertig übertragen wurde.

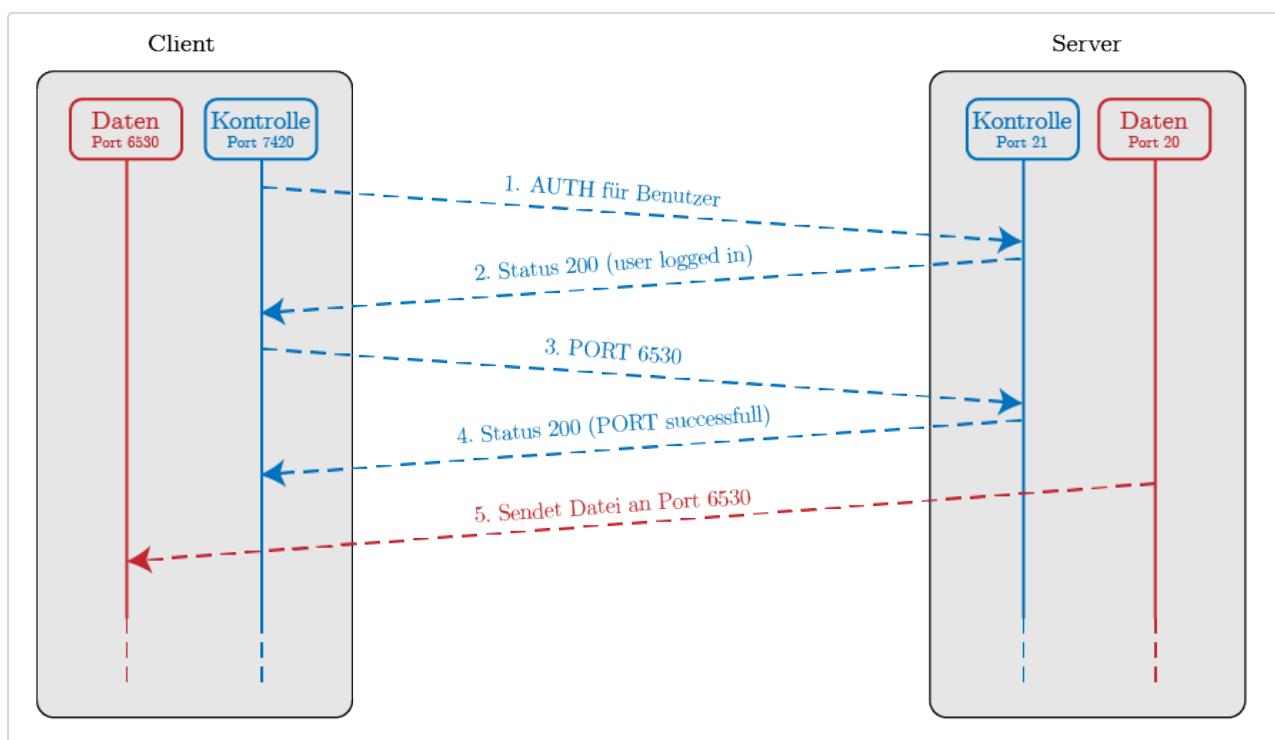
Aktive und passive FTP-Verbindung

Es gibt zwei verschiedenen Varianten wie eine FTP-Verbindung aufgebaut werden kann. Schauen wir uns zu erste den aktiven Modus an, welcher ursprünglich als Standard-Variante vorgesehen war und vor dem passiven Modus existiert hat.

Aktives FTP

Aktives FTP wird so genannt, weil der FTP-Server hierbei die Datenverbindung aktiv zum Client aufbaut. Der Client wählt seine Ports für die Daten- und Kontrollverbindung zufällig, während der Server immer die bekannten Standard-Ports 21 für die Kontroll-Verbindung und 20 für die Datenverbindung zugewiesen hat.

Schauen Sie sich nun den untenstehenden Ablauf an:



1. Der **FTP-CLIENT** baut eine Verbindung zum Server auf. Die IP-Adresse des Servers sowie dessen Port auf der Kontroll-Verbindung sind bekannt. (Port ist in der Regel der definierte Standard-Port 21). Es werden die mit dem AUTH Befehl die Informationen zur Authentifizierung übertragen.
2. Der **FTP-SERVER** bestätigt die erfolgreiche Anmeldung mit der Meldung **user logged in**.

3. Der **FTP-CLIENT** teilt dem Server nun seinen Port für die Daten-Verbindung mit. Dazu sendet der FTP-CLIENT eine sogenannte PORT Nachricht in der die IP-Adresse und der Port in folgenden Format übermittelt wird.

(192,168,1,72,25,130)

Daraus kann der Server nun den Port berechnen. Hierzu mehr unter dem Abschnitt **Port-Berechnung**.

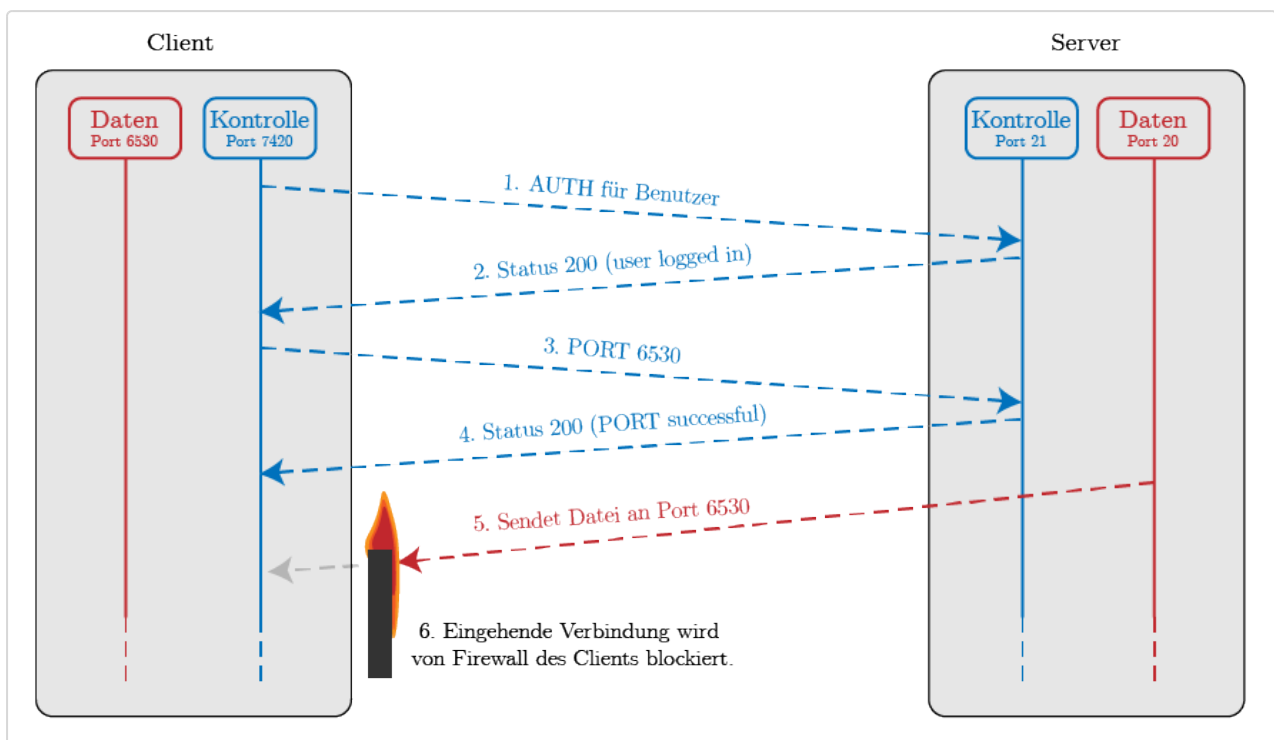
Ab diesem Zeitpunkt weiss der **FTP-Server** nun, auf welchem Port der **FTP-CLIENT** die Datei empfangen kann.

4. Der **FTP-SERVER** bestätigt dem FTP-CLIENT, dass er den Port erhalten hat.
5. Bei einer angefragten Datenübertragung wird der **FTP-Server** nun die Datei an den **FTP-Client** senden. Dazu baut er von sich auf eine neue Verbindung auf den erhaltenen Port des Clients auf.

Das obige Abschlauf-Schema ist nicht vollständig. In Wahrheit werden noch weitere Datenpakete ausgetauscht. z. B. für die Authentifizierung. Dieser Ablauf wurde auf die relevanten Datenpakete reduziert um die Funktionsweise einer aktiven oder passiven FTP-Verbindung zu darzulegen.

Die aktive FTP-Verbindung ist einfach und verständlich aufgebaut. Der FTP-Client teilt dem FTP-Server mit, auf welchem Port er eine Datei entgegen nimmt und der FTP-Server baut dann auf diesen Port eine Daten-Verbindung auf. Jedoch hat sich mit der Zeit in grosses Problem geäussert. Immer mehr Clients waren mit einer Firewall oder einem NAT ausgestattet.

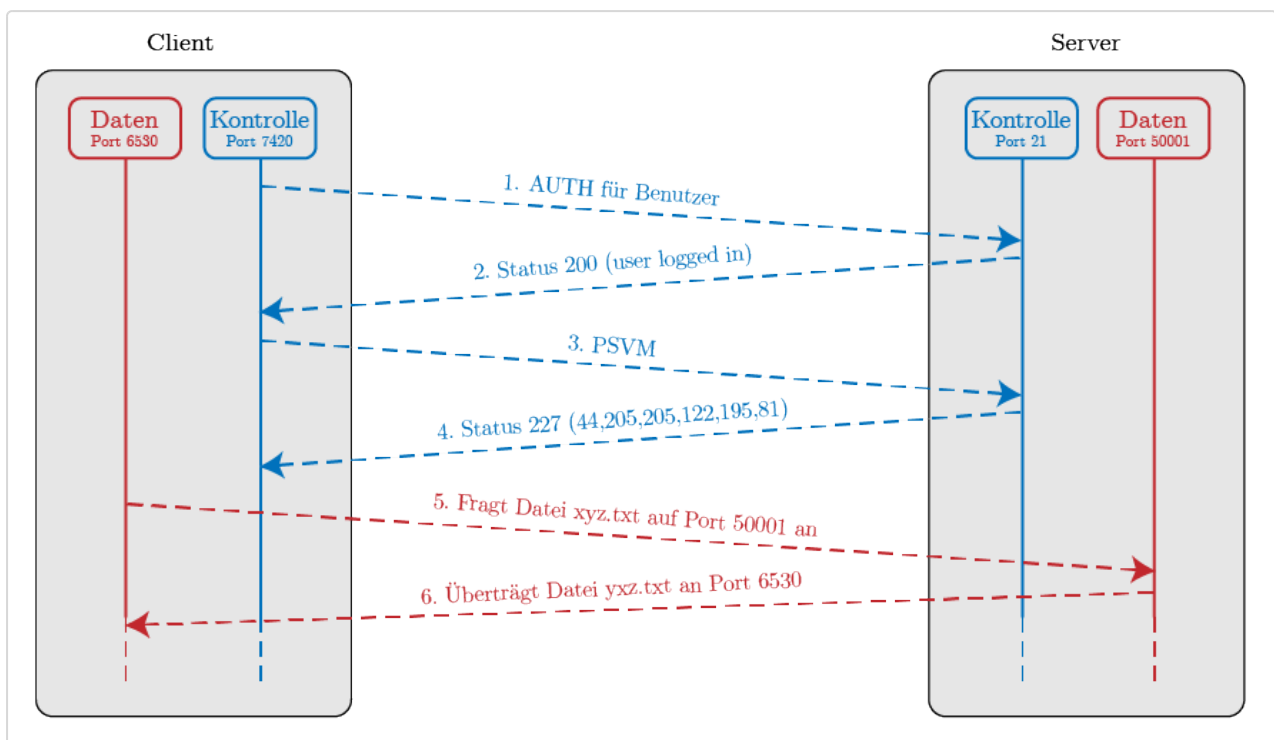
Wie in der nachfolgenden Grafik ersichtlich, blockieren Firewalls eingehende Verbindungen, welche nicht ursprünglich vom Client initiiert wurden. Die Datenpakete der Kontroll-Verbindung werden daher durch die Firewall des Clients gelassen, da diese vom Client selber gestartet wurden. Da aber der FTP-Server die Daten-Verbindung von sich aus startet, wird diese von der Firewall des FTP-Clients blockiert.



Es braucht somit eine andere Lösung, welche dem Client erlaubt die Datenverbindung zu starten. Hierzu wurde passives FTP eingeführt.

Passives FTP

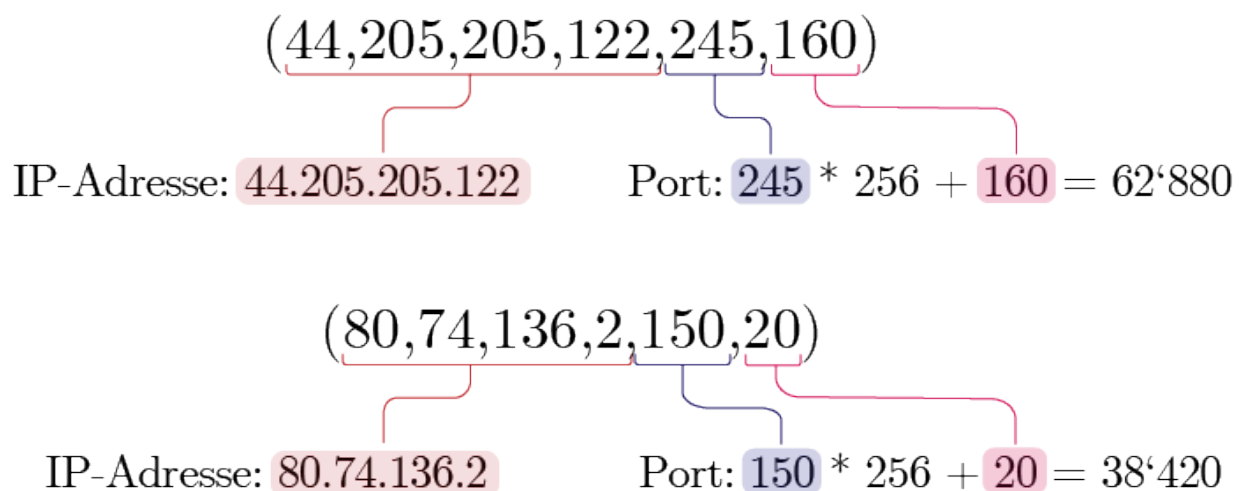
Im Gegensatz zum aktiven FTP, bei dem der Server die Daten-Verbindung aktiv aufbaut, wird beim passiven FTP der Client die Verbindung aufbauen. Der FTP-Server muss die Daten dann nur passiv auf dem kommunizierten Port bereithalten. Passives FTP ist üblich, da viele Firewalls die FTP-Verbindung verunmöglichen, da von aussen her keine Daten-Verbindung aufgebaut werden kann. Weiter ist es sicherer, weil nicht der Server die Daten-Verbindung aufbaut und somit der Client weniger offen für Angriffe ist.



1. Wie beim aktiven FTP wird der **FTP-CLIENT** als erstes eine Authentifizierung vornehmen.
2. Der **FTP-SERVER** bestätigt diese.
3. Nun wird der **FTP-CLIENT** mit einer **PSVM**-Nachricht den passiven-Modus vom **FTP-SERVER** anfordern.
4. Der **FTP-SERVER** bestätigt den passiven Modus mit dem Status 227 und den Port-Informationen. Der **FTP-SERVER** hat dem **FTP-CLIENT** den Port 50001 und die IP 44.205.205.11 übermittelt. Der **FTP-SERVER** öffnet einen neuen Port 50001 auf dem er auf eingehende Verbindungen wartet.
5. Der **FTP-CLIENT** möchte nun die Datei xyz.txt abrufen. Dazu stellt der Client eine Verbindung zum **FTP-SERVER** über den Port 50001 mit. Der Port wurde vom **FTP-SERVER** in der letzten Nachricht mitgeteilt. Mit diesem Datenpaket erhält der FTP-Server automatisch auch den Port der Daten-Verbindung des **FTP-CLIENTS**.
6. Der **FTP-SERVER** sendet nun die angeforderte Datei xyz.txt an den **FTP-CLIENT** über den Port 6530. Weil die Verbindung vom Client aus gestartet wurde (Die Firewall merkt sich beim Datenpaket die IP-Adresse und den Port des Servers) wird das eingehende Datenpaket akzeptiert.

Port Berechnung

Das aktive und passive FTP haben eine Gemeinsamkeit und zwar werden jeweils an einer Stelle Informationen ausgetauscht, auf welchem Port der FTP-Client oder der FTP-Server eine Verbindung geöffnet haben. Diese Information ist nach einem bestimmten Schema codiert.



Mit diesem Schema und den Informationen aus dem Kapitel über aktives und passives FTP sollten Sie nun die unten stehenden Aufgaben lösen können.

Auftrag 1: Nachfolgende Datenpakete einer Kontroll-Verbindung wurde aufgefangen. Leiten Sie daraus ab, ob dieses Paket an einen FTP-Client oder einen FTP-Server übermittelt wurde, die codierte IP-Adresse, den codierten Port und ob es eine aktive oder passive FTP-Verbindung ist.

1.) PORT 192,168,1,72,209,255

Datenpaket wurde an einen _____ versendet.

Die IP-Adresse lautet:

Die Port-Nummer lautet:

Es handelt sich um eine _____ FTP-Verbindung.

2.) PORT 111,222,111,222,111,222

Datenpaket wurde an einen _____ versendet.

Die IP-Adresse lautet:

Die Port-Nummer lautet:

Es handelt sich um eine _____ FTP-Verbindung.

3.) 227 Entering Passive Mode (210,128,268,122,199,199)

Datenpaket wurde an einen _____ versendet.

Die IP-Adresse lautet:

Die Port-Nummer lautet:

Es handelt sich um eine _____ FTP-Verbindung.

4.) PORT 111,222,111,222,111,222

Datenpaket wurde an einen _____ versendet.

Die IP-Adresse lautet:

Die Port-Nummer lautet:

Es handelt sich um eine _____ FTP-Verbindung.

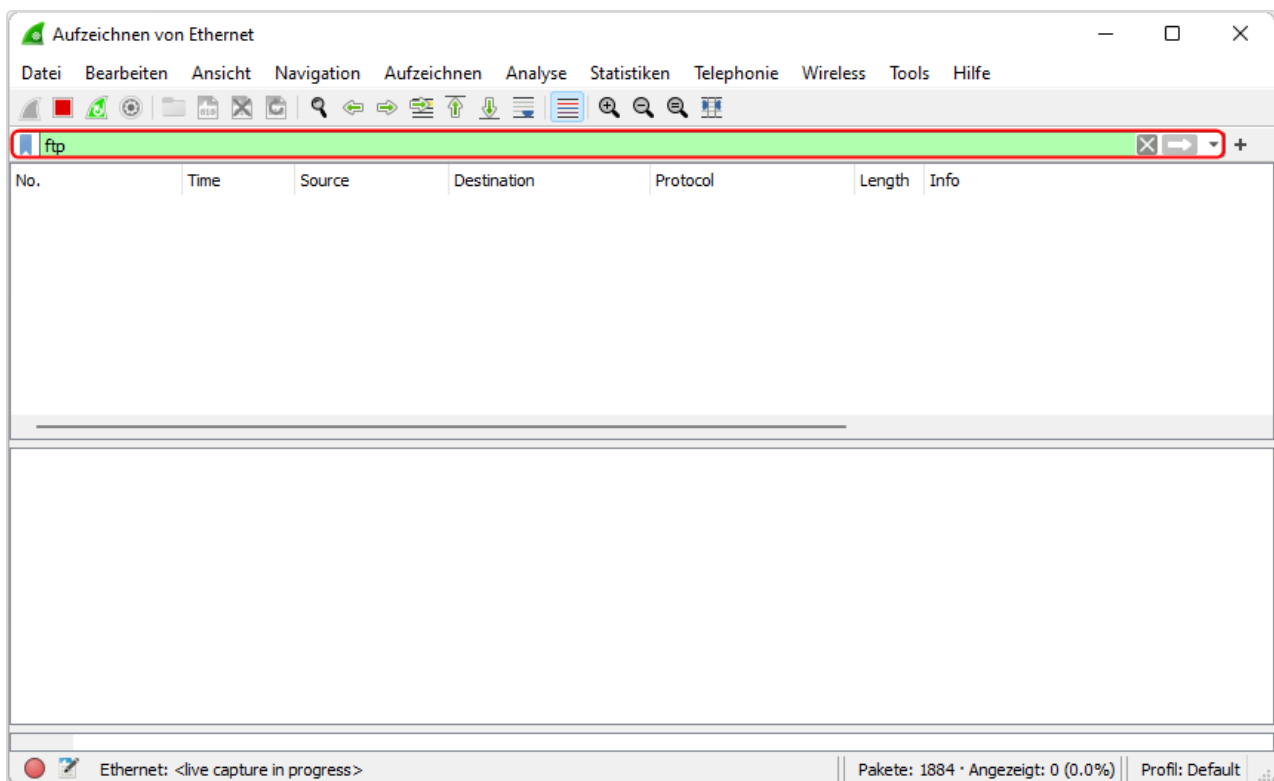
Praktische Übung

Vorbereitung

- Laden die die Software FileZilla Client herunter und installieren Sie diese auf Ihrem Gerät:
<https://filezilla-project.org>
- Aktivieren Sie den Telnet-Client unter Windows: <https://www.heise.de/tipps-tricks/Windows-10-Telnet-Client-aktivieren-4569277.html>
- Wireshark sollte aus der vorhergehenden Aufgabe noch installiert sein.

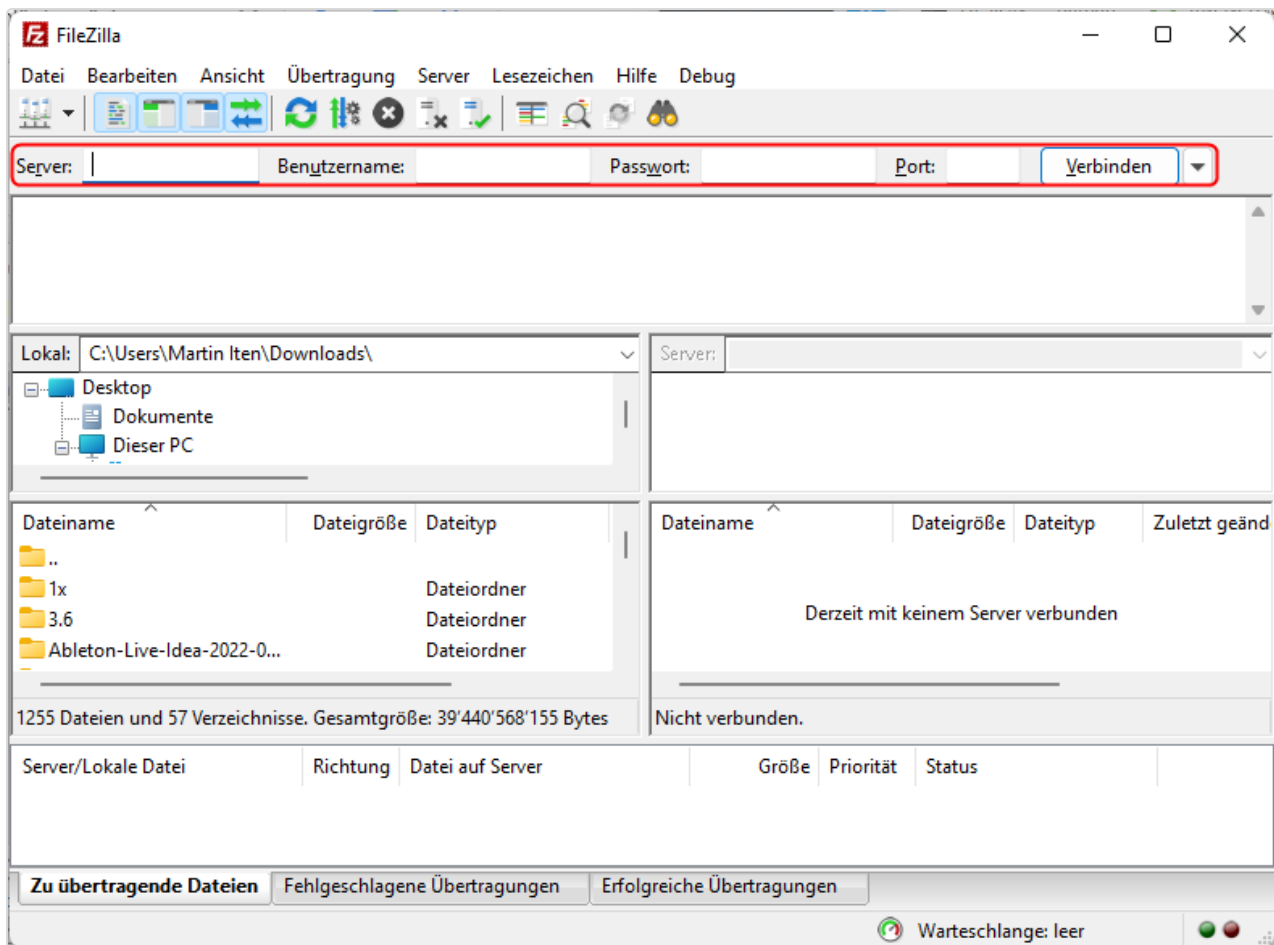
Anmeldung über FileZilla

Starten Sie als erstes die Software Wireshark und setzen Sie einen Filter auf **ftp**:

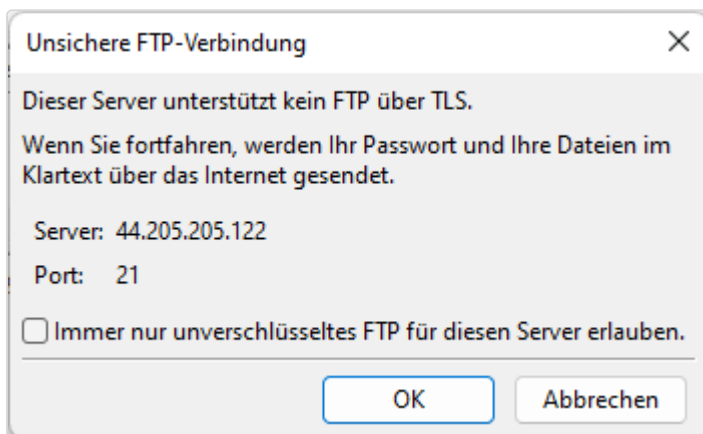


Starten Sie nun die Software FileZilla Client und stellen Sie eine Verbindung mit folgenden Parametern her:

- **Server:** 44.205.205.122
- **Benutzername:** user-read
- **Password:** geheimespasswort
- **Port:** 21



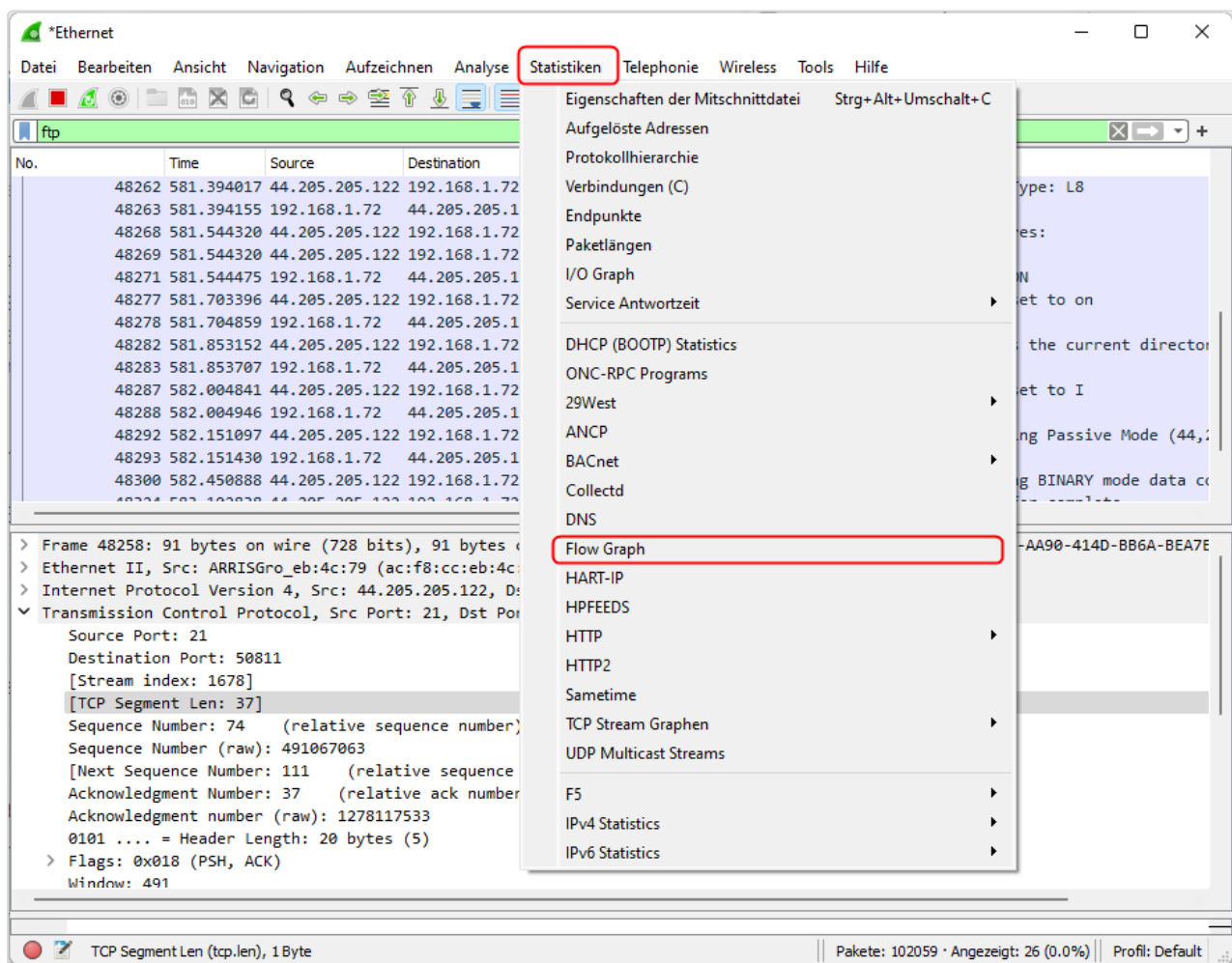
Sie werden die Meldung erhalten, dass dieser FTP-Server kein FTP über TLS bietet. Bestätigen Sie diese Meldung:



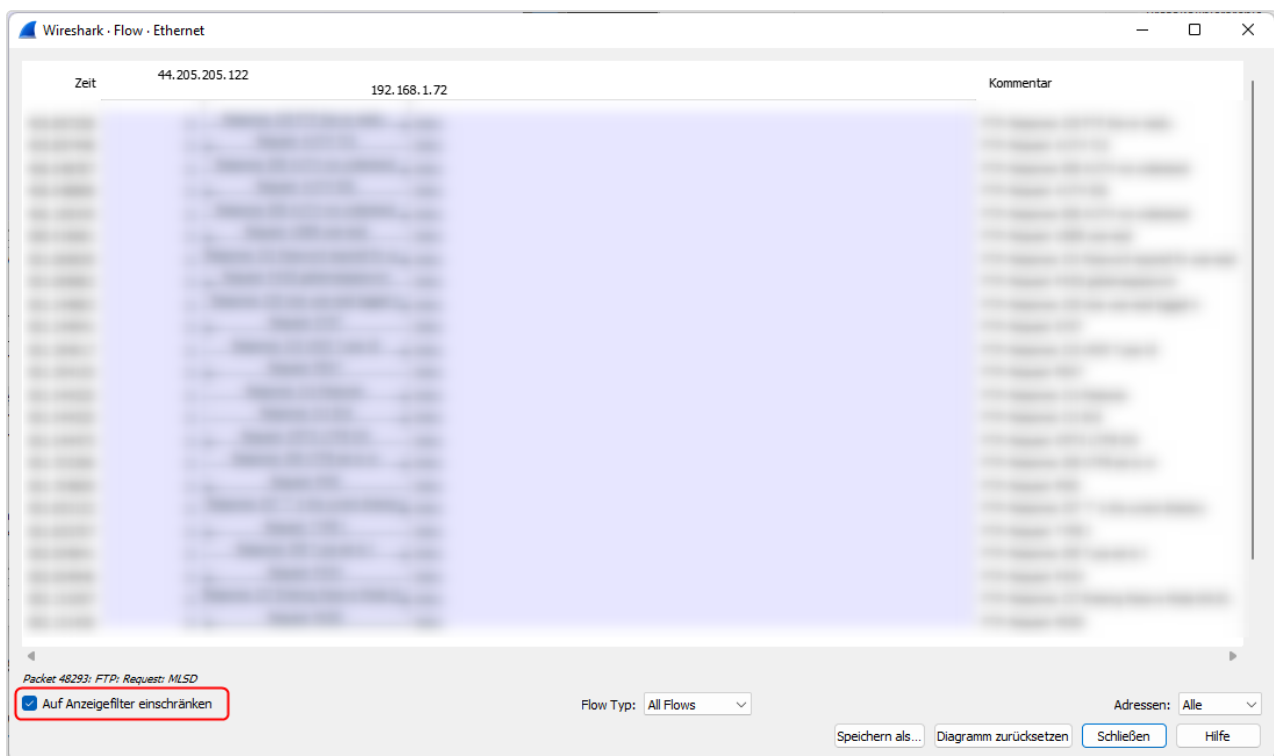
Nun sollten die eine erfolgreiche Verbindung hergestellt haben. **Führen Sie zu diesem Zeitpunkt noch keinen Download aus.** Dies erleichtert Ihnen die Beantwortung der nachfolgenden Fragen.

Warnung: In dieser Übung verwenden wir bewusst eine unverschlüsselte Verbindung, damit wir die übertragenen Daten nachvollziehen können. In der echten Welt sollte aber nie eine unverschlüsselte FTP-Verbindung aufgebaut werden, da jemand der den Datenverkehr mitliest, alle Informationen inkl. Passwörter im Klartext mitlesen kann.

In Wireshark sollten Sie nun die FTP-Datenpakete sehen. Wireshark bietet ein praktisches Tool um die Datenverbindungen grafisch zu visualisieren. Öffnen Sie hierzu über den Punkt **Statistiken** den **Flow Graph**.



Stellen Sie sicher, dass die Option **Auf Anzeigefilter einschränken** aktiviert ist.



Das allererste Datenpaket sollte nun ein Response sein. Der FTP-Client hat zwar die Verbindung zum Server aufgebaut, aber die initiale Verbindung ist eine generische TCP-Verbindung. Erst mit der ersten Bestätigung durch den FTP-Server werden FTP-Pakete ausgetauscht. Daher wird der initiale Verbindungsaufbau zum FTP-Server vom Filter **ftp** nicht abgedeckt.

Somit ist unser erstes FTP-Datenpaket der **Response: 200 FTP Server ready**, was so viel heisst wie, Sie haben einen FTP-Server kontaktiert und ich habe Ihnen geantwortet.

Ignorieren Sie nun die nachfolgenden 4 Datenpakete. Diese enthalten AUTH-Anforderungen, welcher aber aufgrund des fehlenden SSL/TLS Zertifikates nicht erfolgreich sind und dementsprechend mit einer Response: 500 AUTH not understood beantwortet werden. Der FTP-Client versucht zwei mal eine SSL-Verschlüsselung aufzubauen, bevor dieses Vorhaben aufgegeben wird.

Auftrag: Identifizieren Sie nun die Datenpakete, welche in der Authentifizierung des Benutzers beteiligt sind bis zu dem Punkt, wo der Benutzer erfolgreich authentifiziert ist. Notieren Sie den jeweiligen Request und den Response.

Die nachfolgenden Requests sind dazu da, dass der FTP-Client und FTP-Server Informationen über Funktionen und Systemdetails austauschen können. Dies sind namentlich die Requests **SYST**, **FEAT** und **OPTS**. Diese können für dieses Modul mal ingoriert werden.

Nun kommt aber ein spannender Request. Was für eine Information möchte der FTP-Client mit dem **PWD** Request anfragen? Tipp: Im Linux-Kapitel finden die diesen Befehl ebenfalls mit einer Erklärung. Was sagt uns der Response

Prüfen Sie nun im nächsten ob mit einer aktiven oder passive FTP-Verbindung weitergefahren wird. Welche Datenpakete sind hierfür relevant:

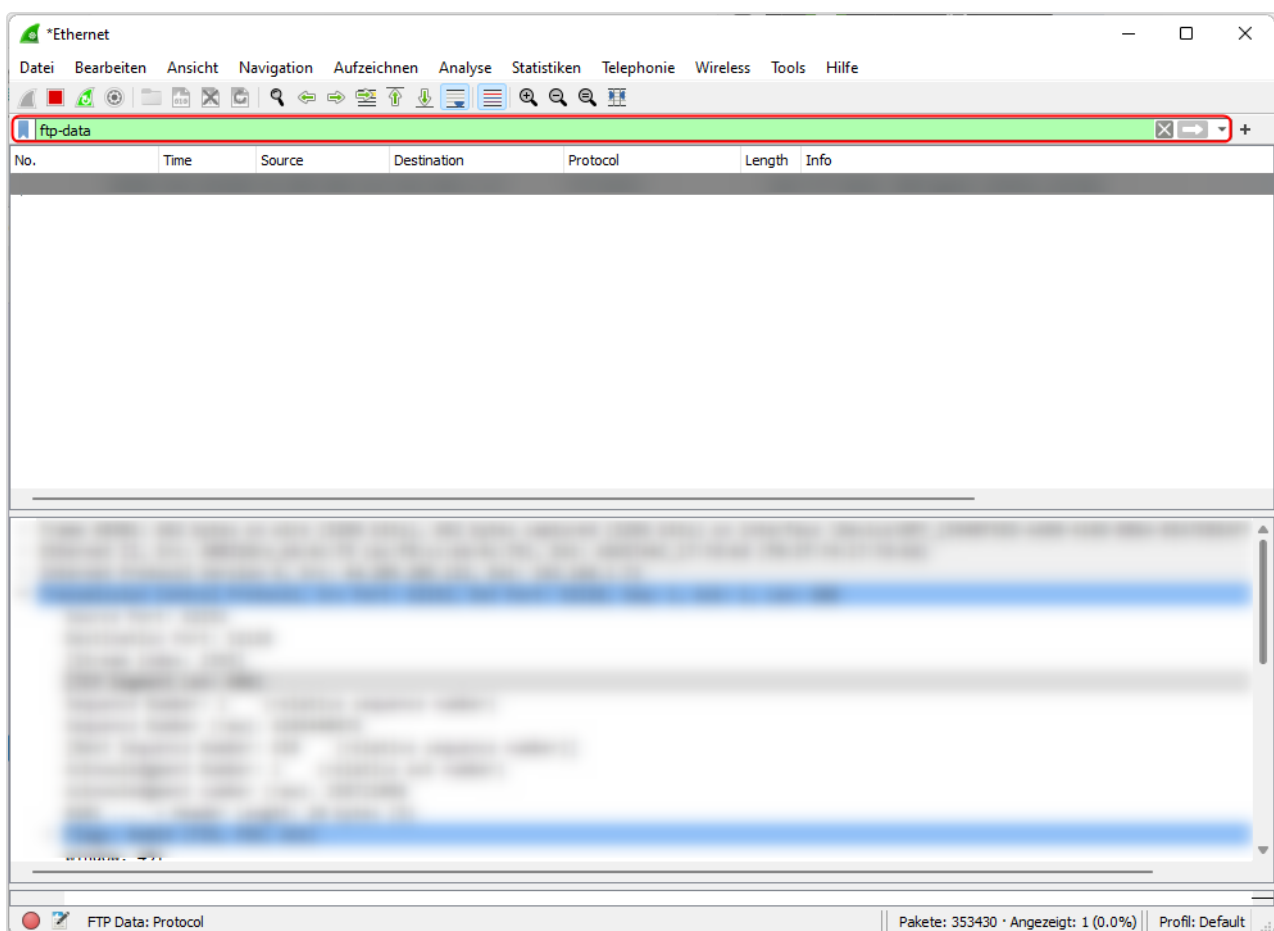
Was für eine FTP-Verbindung wird gesetzt?

Wer definiert die Art der FTP-Verbindung (Client oder Server) und mit welchem Datenpaket geschieht dies genau?

Berechnen Sie aus der Port-Angabe den Port und die IP für die Datenverbindung. Notieren Sie den Lösungsweg und notieren Sie den erhaltenen Port für später.

Nun wird es spannend. Mit dem Request **MLSD** fragt der FTP-Client eine Datenübertragung an. Der FTP-Server antwortet, dass eine Datenverbindung (Binär) für MLSD geöffnet wurde. Somit existiert nun eine weitere Verbindung, die sogenannten Daten-Verbindung. Alle bisherigen Befehle, welche ausgetauscht wurden, liegen auf der Kontroll-Verbindung.

Der Wireshark Filter **ftp** deckt die FTP-Datenverbindung nicht ab, dafür gibt es aber einen separaten Filter, nämlich den **ftp-data**. Nutzen Sie diesen, es sollte nun genau ein Datenpaket ersichtlich sein:



Wählen Sie das betreffende Datenpaket aus und klappen Sie im Protokollstack den Teil **Line-based text data** auf. Erkennen Sie was für Informationen mit MLSD übertragen wurden? Vergleichen Sie die

Informationen mit dem Inhalt im FTP-Client FileZilla. In welchem Format wird die Information übertragen?

Wofür verwendet der FTP-Client diese Information FileZilla die Information aus der MLSD Datenübertragung?

Klappen Sie nun den Teil **Transmission Control Protocol** auf und kontrollieren Sie, ob der Source Port mit dem von Ihnen berechneten Port übereinstimmt.

Auftrag: Wechseln Sie nun in den FTP-Client und laden Sie die **kleine-datei.txt** mit einem Doppelklick herunter. Es sollten nun neue Datenpakete aufgetaucht sein.

Der Befehle MLSD wird benötigt um den Inhalt des Verzeichnisses aufzulisten. Welcher Befehl wird Verwendet um eine Datei abzurufen?

Laden Sie nun die Datei grosse-datei.txt mit einem Doppelklick herunter. Was fällt Ihnen auf im Vergleich zur kleinen Datei? Warum ist das so? (Tipp. Bei TFTP haben Sie eine ähnliche Situation)

Woran können Sie erkennen, dass für die MLSD-Requests, den RETR für die Datei kleine-datei.txt und den RETR für die Datei grosse-datei.txt jeweils eigene Datenverbindungen verwendet wurden? Tipp schauen Sie im Transmission Control Protocol Teil nach.

Wechseln Sie nun den WireShark Filter zu **ftp** und kontrollieren Sie, ob für die Kontroll-Verbindung ebenfalls separate Verbindungen aufgebaut wurden oder ob immer dieselbe Verbindung verwendet wurde insofern die Verbindung nicht manuell Unterbrochen wird oder zulange offen gelassen wurde. (Achtung, ein manueller Unterbruch oder Timeout erzeugt immer eine neue Verbindung)

Revision #4

Created 21 December 2022 08:16:25 by Admin

Updated 12 January 2023 00:40:09 by Admin