

| SMB-Server

- [Einleitung](#)
- [TFTP](#)
- [FTP](#)
- [SMB / SAMBA](#)
- [Auftrag 0003](#)
 - [Der Auftrag](#)
 - [SAMBA Installieren unter Linux](#)
 - [SAMBA Konfigurieren unter Linux](#)
 - [Datei- / Verzeichnisberechtigungen](#)
 - [SMB Freigaben unter Windows einrichten](#)

■ Einleitung

Während auf HTTP basierte Dateiübertragung wie Nextcloud, Goole Drive, OneDrive, iCloud im privaten und zunehmend auch im Geschäftsumfeld eine grössere Rolle einnehmen, spielen die Netzwerkübertragungsprotokolle SMB (CIFS), NFS, FTP, iSCSI weiterhin eine wichtige Rolle. Für IT-Fachpersonen spielt zudem das TFTP Protokoll aufgrund dessen Einfachheit eine wichtige Rolle. Nicht vergessen werden darf SSH, dass nebst dem Remote Terminal auch eine verschlüsselte Dateiübertragung (SFTP) bietet.

Ports

TCP- und UDP-Ports sind eine Software-Abstraktion, um parallele Kommunikationsverbindungen einer oder mehreren Anwendungen voneinander unterscheiden zu können. Ähnlich wie IP-Adressen zur Adressierung von Rechnern in Netzwerken dienen, adressieren Ports spezifische Anwendungen und ihre Verbindungen, die auf einem Rechner laufen.

Zuordnung zwischen Datenpaket und Anwendung: Der TCP-Port

Ein Netzwerk-Port ist eine Adresse, mit deren Hilfe sich UDP- oder TCP-Verbindungen eindeutig bestimmten Anwendungen zuordnen lassen. Zu einer Verbindung gehört auf beiden Seiten jeweils ein Port. Ports können eine fortlaufende Nummer im Bereich von 0 bis 65535 annehmen.

Zweck eines Netzwerk-Ports ist die Zuordnung von UDP- oder TCP-Paketen einer IP-Verbindung zu einer bestimmten Anwendung. Zusammen mit der IP-Adresse bildet der Netzwerk-Port die vollständige Adresse einer Anwendung auf einem Ziel- oder Absendersystem.

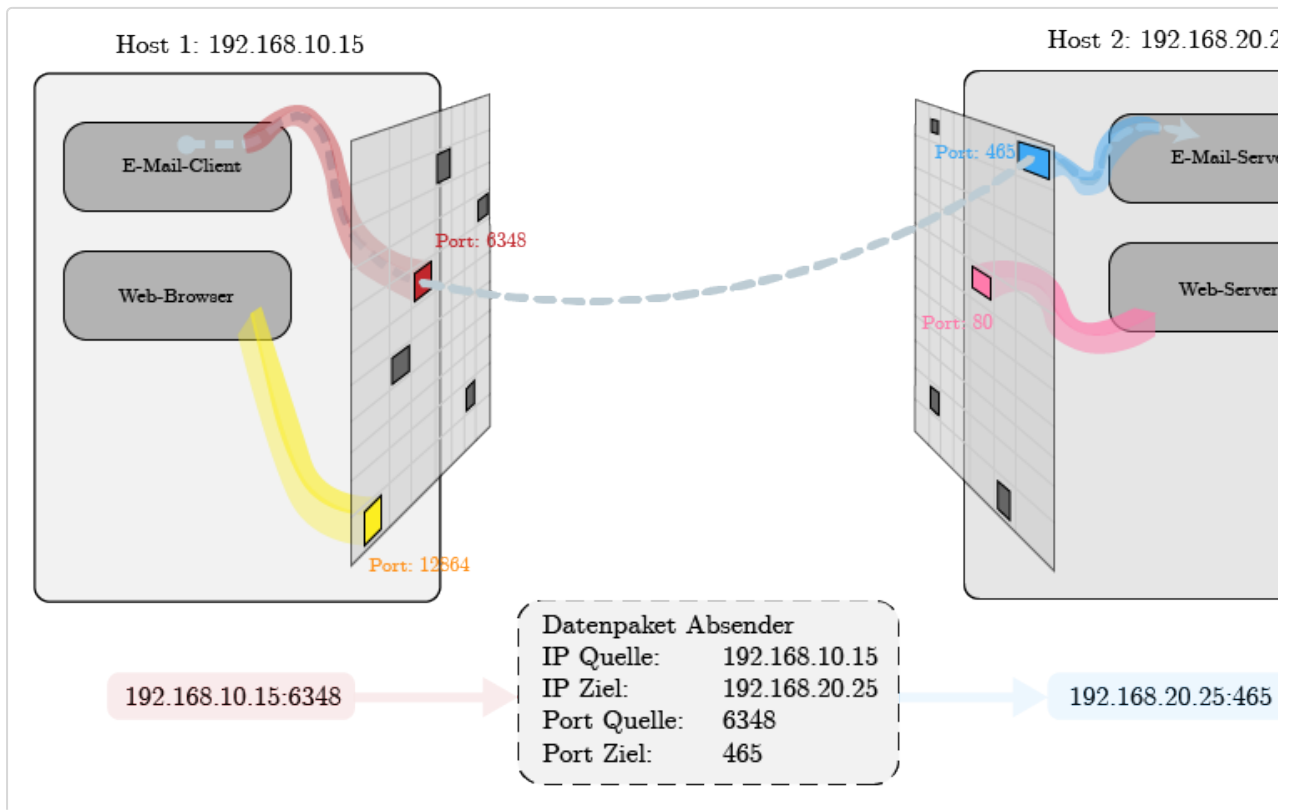
Ein Port ist somit jeweils mit einer spezifischen Applikation verbunden. Im untenstehenden Beispiel sind dies folgende:

Host 1:

- Port 6348 ist mit dem E-Mail Client
- Port 12564 ist mit dem Web-Browser verbunden

Host 2:

- Port 465 ist mit dem E-Mail-Server verbunden
- Port 80 ist mit dem Web-Server verbunden



Es kann nun daher mit der IP-Adresse und der Port-Nummer eine eindeutige Verbindung zwischen zwei Applikationen hergestellt werden. Im obigen Beispiel stellt der E-Mail Client eine Verbindung mit dem E-Mail Server her um Daten auszutauschen. Diese Verbindung funktioniert in **beide Richtungen**. Eine Verbindung hat somit in jedem Fall zwei Ports, eine auf der Client-Seite und eine auf der Server-Seite.

Dafür muss nun aber der Teilnehmer, welche die Verbindung aufbauen möchte die IP und die Port-Nummer des Kommunikations-Partners kennen. Dazu sind Portnummern fest einzelnen Anwendungen zugeordnet. Andere Portnummern können dynamisch vergeben werden. So ein fest zugewiesener Port ist der Port 465. Dahinter ist im Normalfall ein E-Mail Server zu finden. Somit kann der Host 1, welcher eine E-Mail an den Host 2 einliefern möchte, den Port 465 als Zielpport verwenden.

Funktionsweise der Ports

Anwendungsserver (Wie z. B FTP-Server, E-Mail Server, DNS-Server) fordern bei ihrem Start beim Betriebssystem einen Port an, auf dem sie Verbindungen entgegennehmen können. Für Standard-Anwendungen handelt es sich immer um die gleichen festen Portnummern. Anschließend ist der Netzwerk-Port fest mit der Anwendung verknüpft. Clients erhalten vom Betriebssystem in der Regel einen zufälligen Port dynamisch zugewiesen. Diesen können sie als Absender-Port nutzen, um eine Verbindung zu einem Zielsystem aufzubauen. Der Client muss die Ziel-IP-Adresse und den Ziel-Port des Servers kennen. Beim Verbindungsaufbau lernt der Server den Port des Clients kennen und adressiert Antwortpakete an diesen.

Standardisierung und Einteilung der Ports

Die Portnummern sind von der IANA und der ICANN (Internet Corporation for Assigned Names and Numbers) in verschiedene Bereiche eingeteilt. Den Bereich der Standard-Ports oder Well Known Ports bilden die Portnummern 0 bis 1023. Sie sind fest einem bestimmten Anwendungsprotokoll zugeteilt. Anschließend folgt mit den Portnummern 1024 bis 49151 der Bereich der so genannten Registered Ports. Diese Netzwerk-Ports dürfen für die Registrierung verschiedener Anwendungen verwendet werden. Die Reservierung erfolgt wiederum über IANA und ICANN. Die Portnummern von 49152 bis 65535 nennen sich Dynamically Allocated Ports und stehen für die dynamische Zuordnung zur Verfügung. Clients nutzen diese Ports für ihre ausgehenden Verbindungen als Absender-Ports. Beispiele für Standard-Ports sind:

- TCP 21: File Transfer Protocol (FTP)
- TCP 23: Telnet
- TCP 80: Hypertext Transfer Protocol (HTTP)
- UDP 53: Domain Name System (DNS)
- UDP 69: Trivial File Transfer Protocol (TFTP)
- UDP 161: Simple Network Management Protocol (SNMP)

Die verschiedenen Portzustände

Je nach Status einer IP-Verbindung und der jeweiligen Anwendung können Netzwerk-Ports unterschiedliche Zustände annehmen. Über den Zustand ist definiert, ob die Kommunikation mit der verknüpften Anwendung möglich ist. Es existieren die drei grundsätzlichen Zustände offen (open), geschlossen (closed) und gefiltert (filtered).

Ist ein Netzwerk-Port offen, ist die Anwendung hinter diesem Port dazu bereit, Verbindungen entgegen zu nehmen.

Ein geschlossener Netzwerk-Port bedeutet, dass es keine Anwendung gibt, die über den Netzwerk-Port erreichbar ist. Verbindungsaufbauwünsche zu diesem Port werden aktiv abgelehnt. Auch durch eine Firewall geschützte Ports können den Zustand geschlossen aufweisen. In diesem Fall ist die dahinterliegende Anwendung ebenfalls unerreichbar.

Der Zustand gefiltert (auch geblockt genannt) weist darauf hin, dass eine Firewall den Netzwerk-Port schützt. Er ist weder offen noch geschlossen, ermöglicht aber ebenfalls keinen Verbindungsaufbau.

YouTube Videos:

<https://www.youtube.com/watch?v=g2fT-g9PX9o>

<https://www.youtube.com/watch?v=aQoFR6m1yOM>

<https://www.youtube.com/watch?v=RDotMcs0Erg>

TFTP

TFTP (Trivial File Transfer Protocol) ist ein einfaches Protokoll, das es Netzwerkgeräten ermöglicht, Dateien über das Netzwerk zu übertragen. Es ist eine schlanke Version des FTP (File Transfer Protocol) und wird häufig verwendet, um kleine Dateien wie Router-Konfigurationsdateien zu übertragen.

Es ist eine Client-Server-Anwendung, bei der Clients Anfragen an einen Server senden, um Dateien zu erhalten. Der Server überträgt dann die Datei über das Netzwerk an den Client. Es ist ein einfaches, leichtes Protokoll, das eine einfache und zuverlässige Dateiübertragung ermöglicht. TFTP wird häufig zum Herunterladen und Hochladen kleiner Dateien wie Router-Konfigurationsdateien, Firmware-Updates und Log-Dateien verwendet. Es ist auch eine beliebte Wahl für die Übertragung von Boot-Images und anderen Systemdateien auf Netzwerkgeräte.

Das Protokoll stößt aber an seine Grenzen, wenn es um große Dateien geht. Es kann zwar für große Dateien verwendet werden, aber da es langsam und ineffizient sein kann ist es nicht die beste Wahl hierfür. Für Netzwerkadministratoren hingegen ist TFTP ein leistungsstarkes Tool, um Systeme zu konfigurieren und zu administrieren und ist eine gute Wahl für die Übertragung kleiner Dateien. Es ist leicht zu implementieren und hat eine sehr geringe Netzwerkauslastung.

In der Einfachheit von TFTP liegen auch einige Nachteile, so weist das Protokoll beispielsweise keine Benutzerauthentifizierung auf. Somit können Dateien, welche von einem TFTP-Server zur Verfügung gestellt werden, von jedem TFTP-Client uneingeschränkt geladen werden. Auch kann mit TFTP nicht durch Verzeichnisse navigiert werden und es ist nicht möglich, die Dateien und Verzeichnisse mit speziellen Lese- und Schreibberechtigungen auszustatten.

Vorbereitung.

1. Laden Sie die Software Wireshark herunter: <https://www.wireshark.org/download.html>
(Windows Installer)
2. Installieren Sie Wireshark
3. Laden Sie die Software pumpKIN TFTP Client herunter: <https://kin.klever.net/pumpkin/binaries/>
(pumpkin-2.7.3.exe)

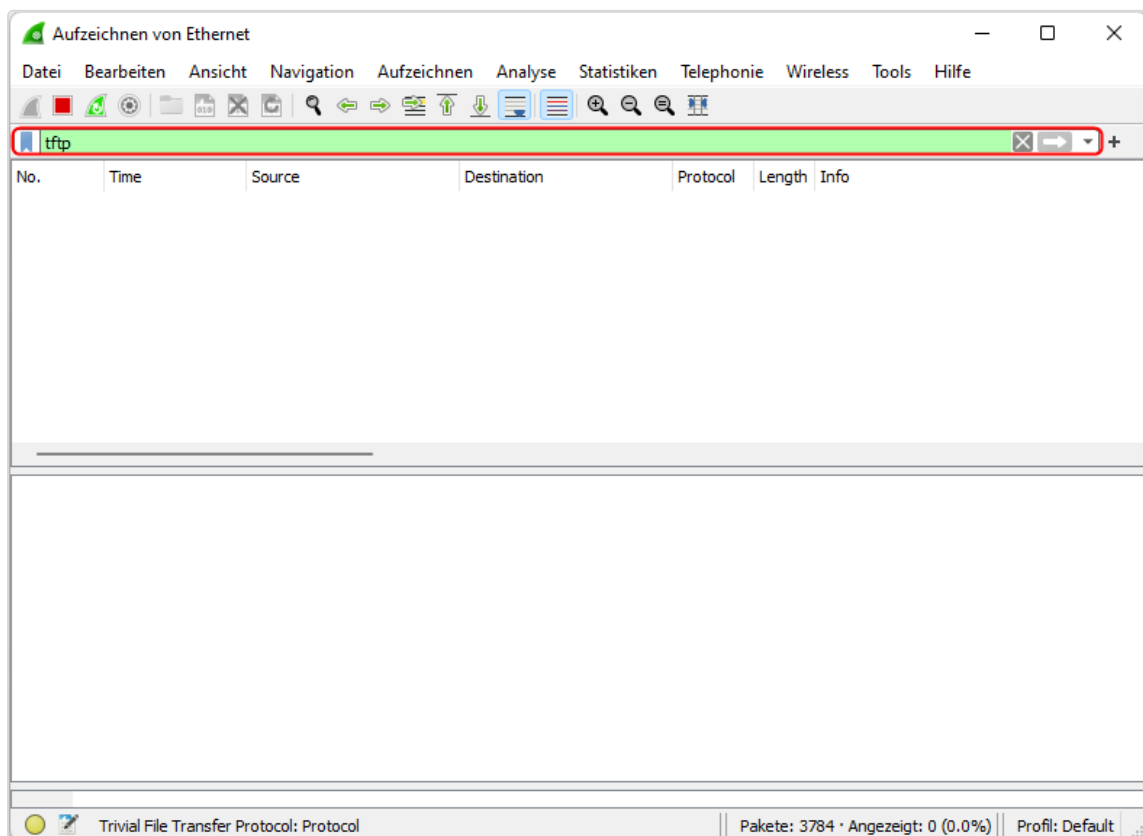
4. Installieren Sie die Software pumpKIN TFTP Client.

Download einer Datei

Machen Sie sich als erstes mit der Software Wireshark vertraut. Hierzu stehen folgende Ressourcen bereit:

- <https://www.youtube.com/watch?v=yn3yzFDub1E> (ca 10 Minuten)
- <https://de.wikipedia.org/wiki/Wireshark> (Grobe Übersicht)
- <https://www.computerbild.de/artikel/cb-Tipps-Software-Wireshark-Anleitung-um-Netzwerkverkehr-zu-analysieren-erste-Schritte-32362313.html>

1. Starten Sie nun die Software Wireshark. Wählen Sie als Schnittstelle Ihr WLAN-Adapter aus.
2. Sie sehen nun den ganzen Traffic aller Datenpakete. Uns interessieren aber nur TFTP-Datenpakete. Aus diesem Grund setzen Sie als erstes einen Filter. Für TFTP-Pakete gibt es bereits einen vorgefertigten Filter **tftp**. Tragen Sie diesen in der Filtereingabe ein:

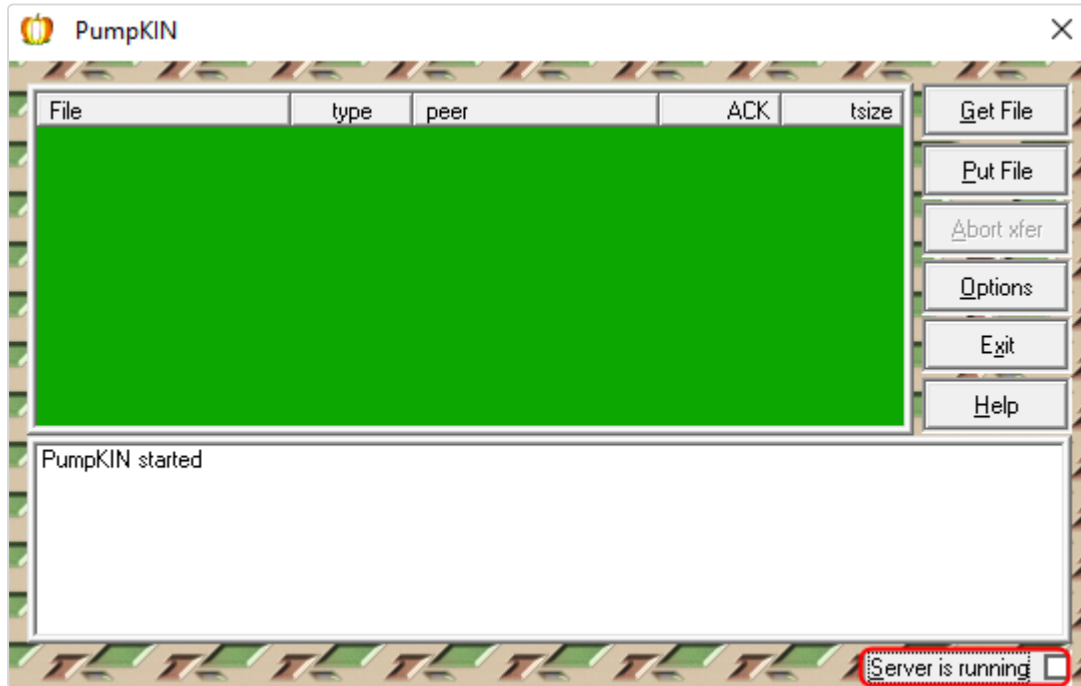


3. Nun sollten Sie keinen anderen Traffic mehr sehen, ausser TFTP-Pakete. Da noch keine versendet wurden, ist die Anzeige dementsprechend leer.

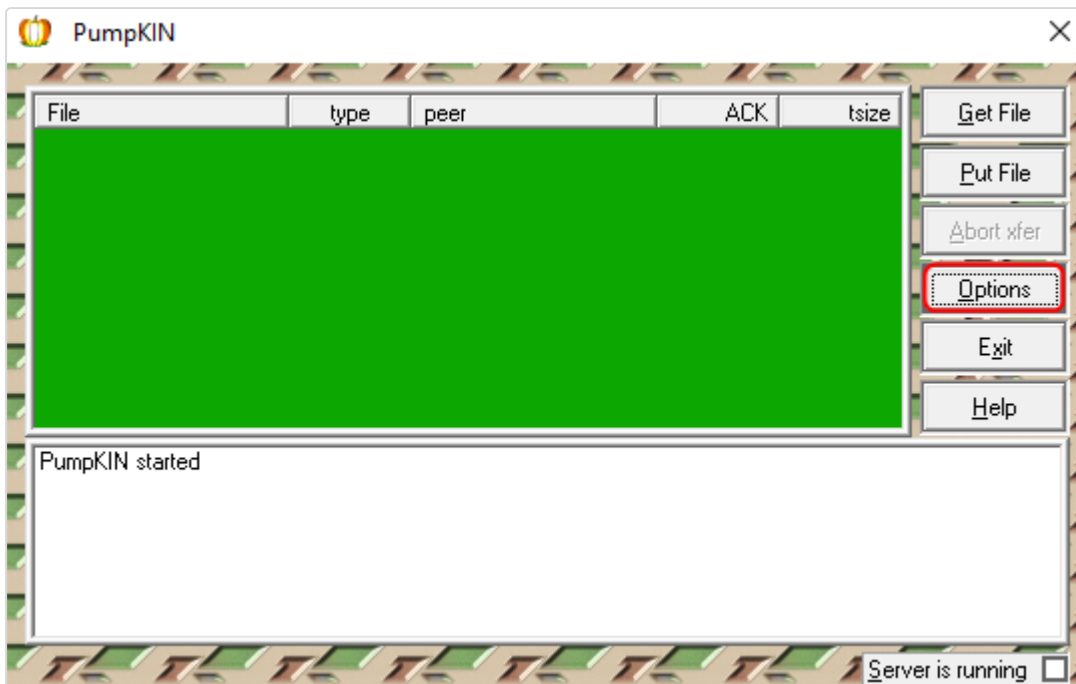
4. Starten Sie nun die Software pumpKIN. Hierbei handelt es sich um eine Software, welche gleichzeitig als TFTP-Server und TFTP-Client fungieren kann. Als erstes werden Sie die reine TFTP-Client Funktion untersuchen.

! Bei der ersten Verwendung müssen Sie die Windows Defender Meldung zulassen !

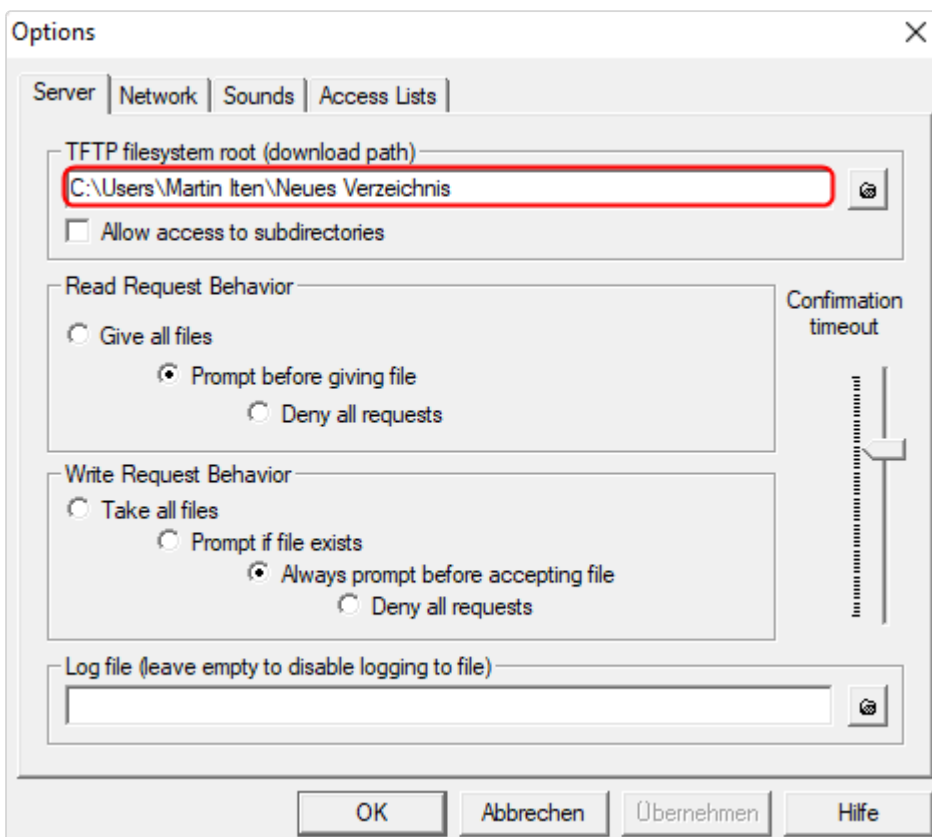
Stellen Sie hierzu sicher, dass die Option **Server is running** deaktiviert ist (Haken entfernen)



5. Nun setzen Sie das Arbeitsverzeichnis des Clients über **Options**:

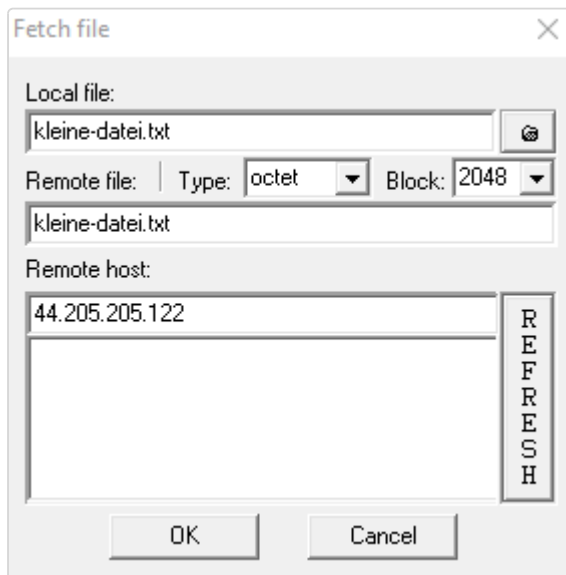


Und dann **TFTP filesystem root (download path)**. Erstellen Sie hier am besten ein neues Verzeichnis und wählen Sie dieses dann in der Einstellung aus:



Bestätigen Sie anschliessend mit **OK**.

6. Nun geht es darum, eine Datei von einem TFTP-Server herunterzuladen. Der TFTP-Client pumpKIN bietet hierzu die Option **GET File**. Verwenden Sie als TFTP-Server den Remote host **44.205.205.122** und als gewünschte Datei zum Herunterladen unter **Remote file** die **kleine-datei.txt**



Bestätigen Sie mit **OK**.

7. Der Download sollte nun erfolgen. Prüfen Sie, ob Sie die Datei sehen.

Aufgabe 1: Prüfen Sie nun Wireshark. Es sollen jetzt Datenpakete ersichtlich sein. Jede Zeile entspricht hierbei einem Datenpaket. Erinnern Sie sich an die DHCP-Theorie. Dort gab es auch verschiedenen Datenpakete, welche zwischen Server und Client ausgetauscht wurden. Bei TFTP ist dies nicht anders.

Frage: Wie viele Datenpakete sind nun im Wireshark vorhanden?

Aufgabe 2: Identifizieren Sie den Typ dieser TFTP-Datenpakete. Schauen Sie hierzu in welche Richtung das Paket versendet wurde (Source, Destination) und was dieses in etwa enthält (Info).

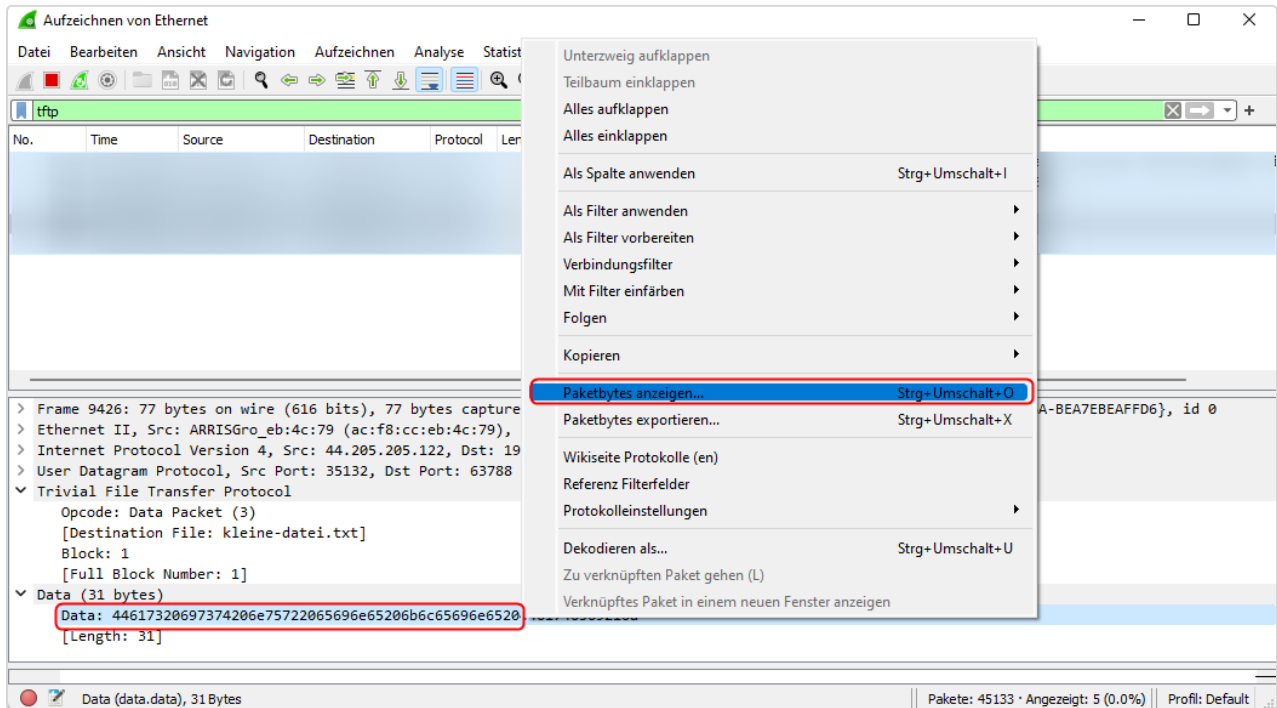
Datenpaket 1: Client -> Server, Read Request für die Datei kleine-datei.txt

Datenpaket 2:

...

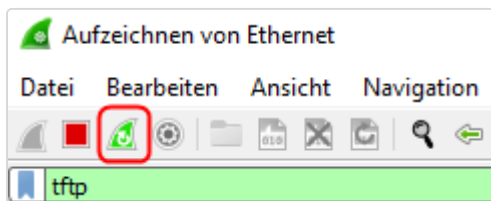
Frage: In welchem Datenpaket werden vermutlich die Daten übertragen?

Klicken Sie nun auf das Datenpaket wo die Daten übertragen wurden. Klappen Sie im Baum das Feld Data aus und machen Sie einen Rechtsklick auf den Data-Wert (44617320.....). Wählen Sie hier die Option Paketbytes anzeigen:



Frage: Was zeigt Ihnen Wireshark nun an?

Auftrag: Leeren Sie nun die aufgezeichneten Datenpakete, indem Sie die Aufzeichnung neu starten:



Speichern Sie die bisher aufgezeichneten Daten, so dass Sie diese später wieder finden.

Führen Sie nun vom selben TFTP-Server einen Download der Datei **grosse-datei.txt** durch. Prüfen Sie nun, ob sich etwas verändert hat im Vergleich zur kleineren Datei.

Tipp: Sie können nun die aufgezeichnete Wireshard-Datei separat öffnen um beide Aufzeichnungen miteinander zu vergleichen. Setzen Sie wieder den Filter **tftp** um nur die relevanten Datenpakete zu sehen.

Frage: Was hat sich verändert?

Frage: Was ist nun Ihre Schlussfolgerung daraus?

Frage: Können Sie anhand der aufgezeichneten ableiten wie gross ein Block bei der Übertragung sein darf bevor er aufgeteilt wird?

Tipp: Vergleichen Sie die vier Blöcke. Die Grösse der Datei ist so gewählt, dass die Grösse des letzten Blocks nicht ausgeschöpft wird und somit kleiner als möglich übertragen wird.

Frage: Untersuchen Sie den TFTP-Client dahingehend ob es eine Möglichkeit gibt, zu prüfen was für Dateien auf dem TFTP-Server sonst noch abgelegt sind.

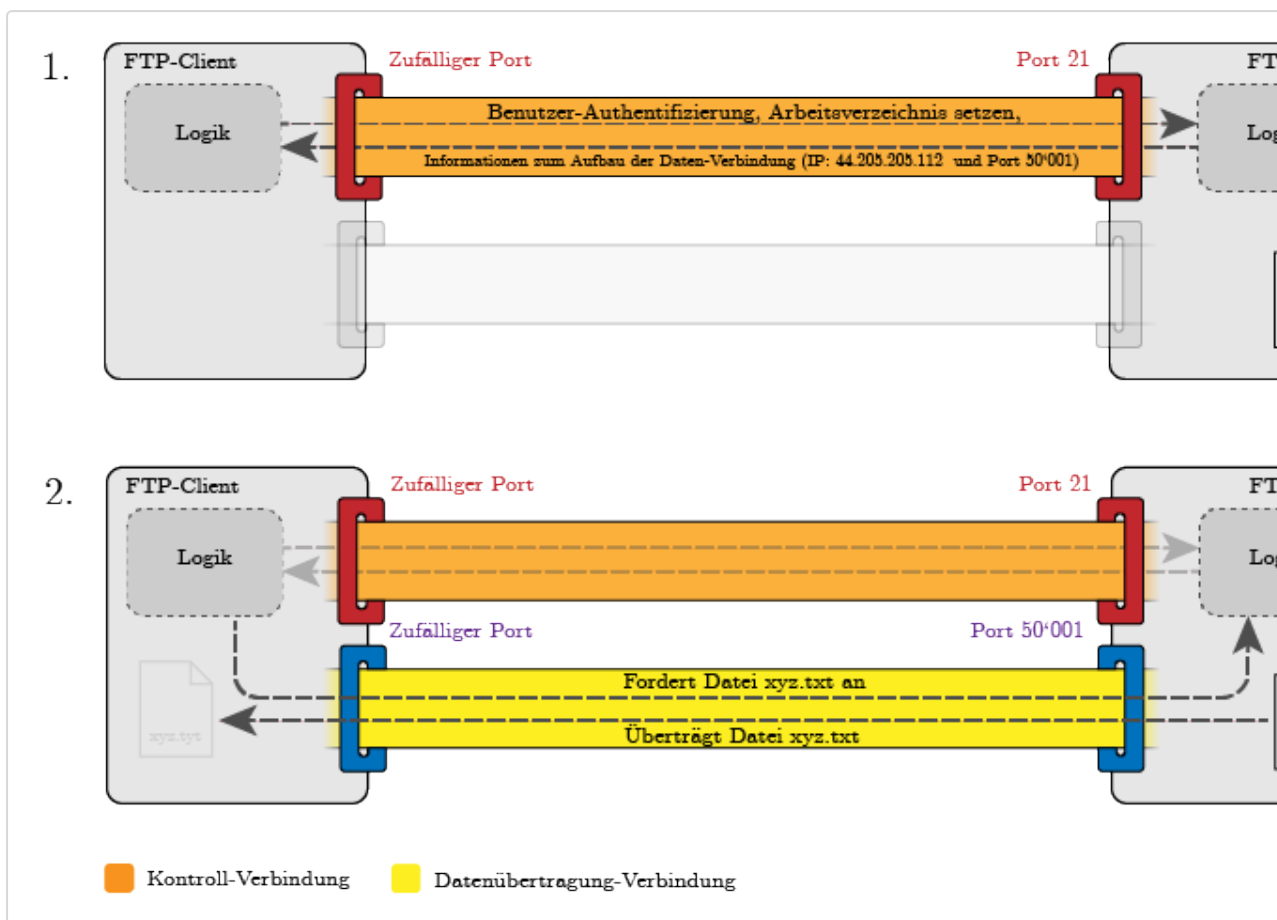
Frage: Was bedeutet dies für Sie als TFTP-Anwender?

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 der 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.

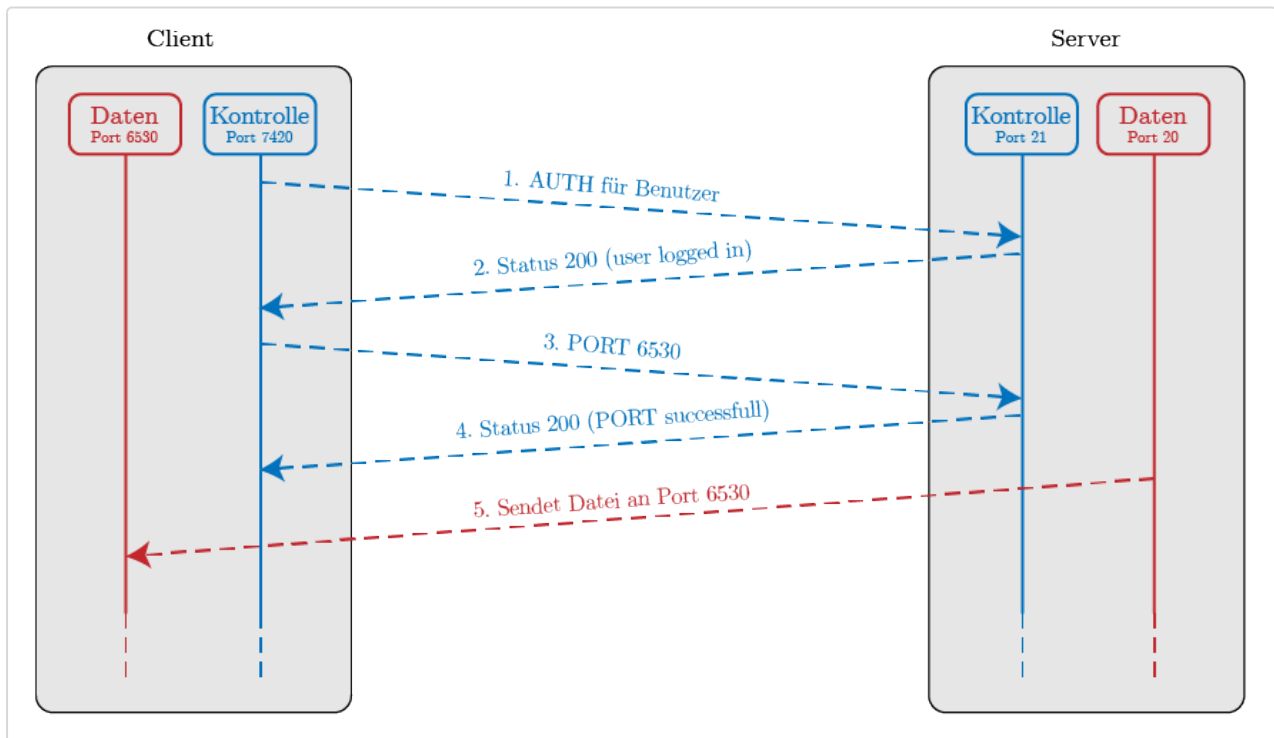
Aktive und passive FTP-Verbindung

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 aus 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 folgendem Format übermittelt wird.

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

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

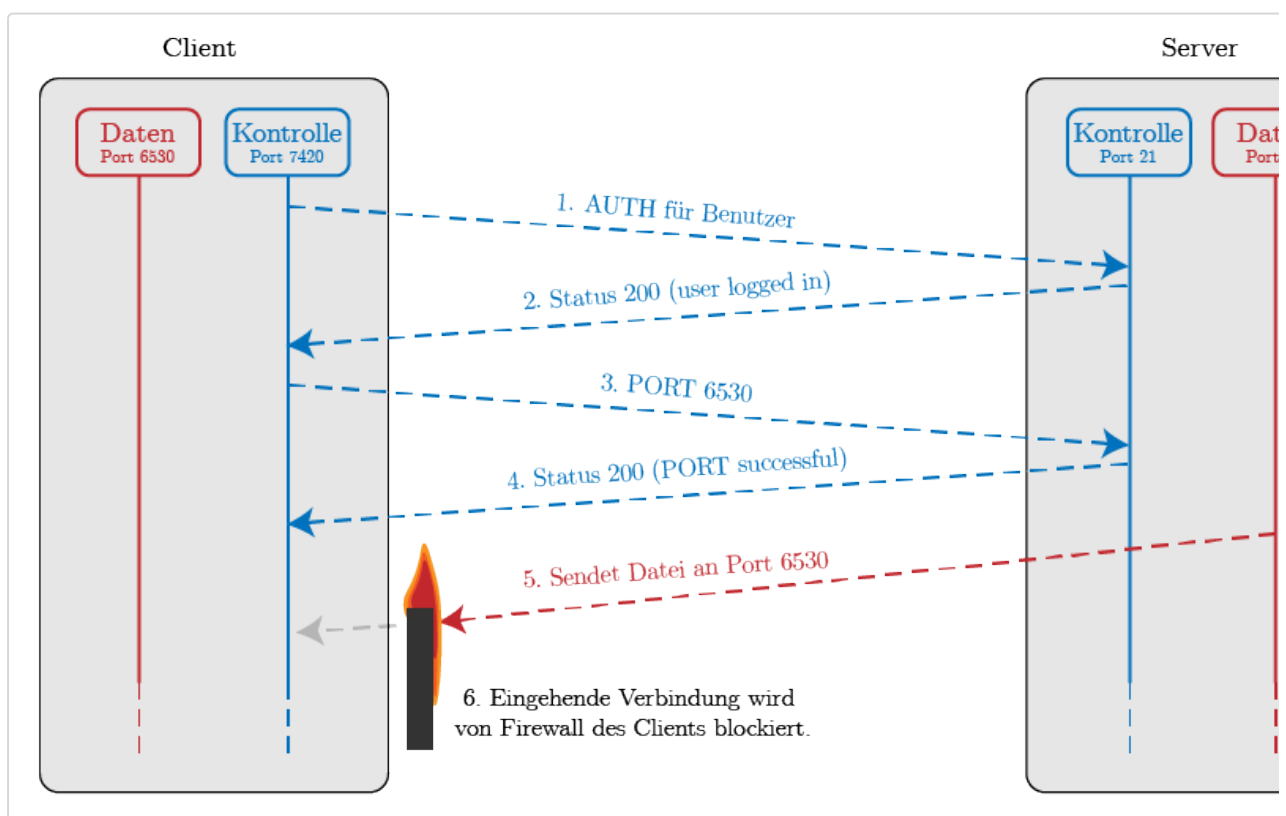
Ab diesem Zeitpunkt weiss der **FTP-Server** nun, auf welchem Port der **FTP-CLIENT** die Datei empfangen kann.

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.

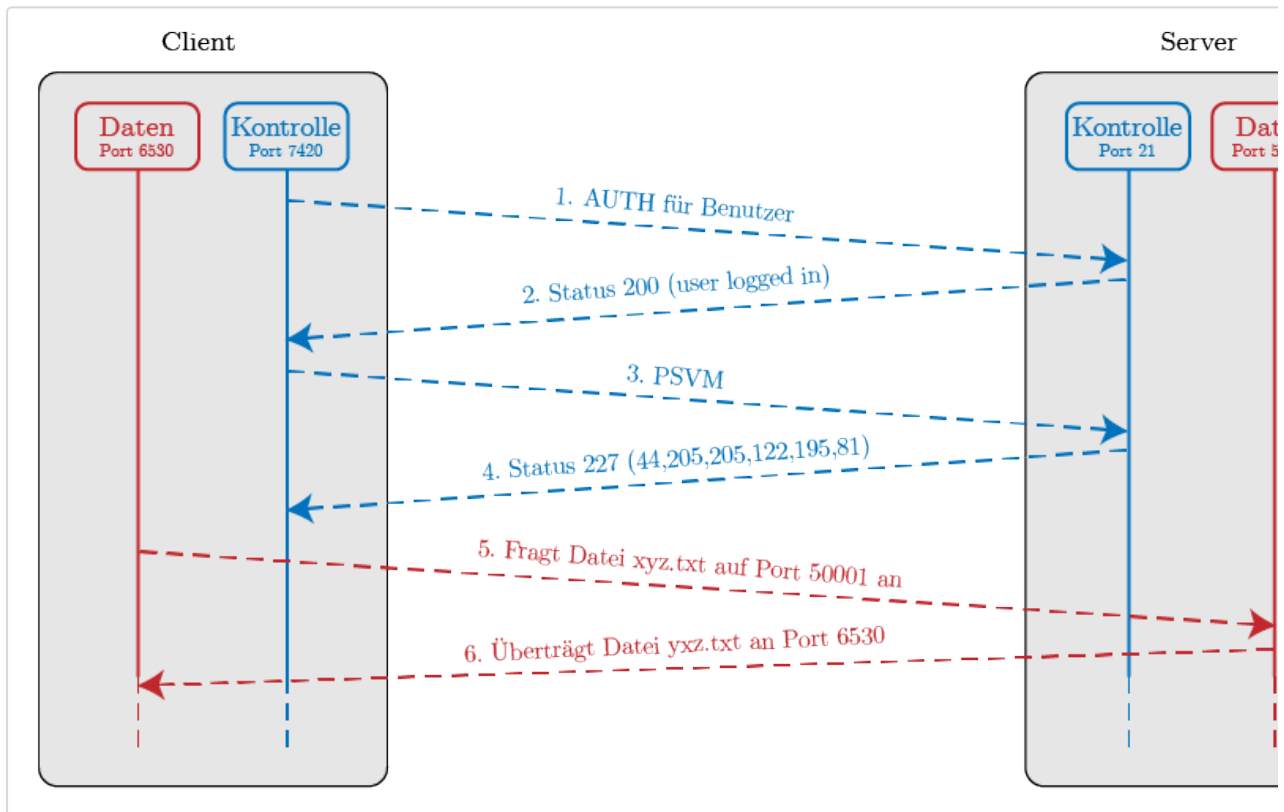
Wie in der nachfolgenden Grafik ersichtlich, blockieren Firewalls eingehende Verbindungen, welche nicht ursprünglich vom Client initiiert wurden. Die Datenpakete der Kontroll-Verbindung werden daher durch die Firewall des Clients gelassen, da diese vom Client selber gestartet wurden. Da aber der FTP-Server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



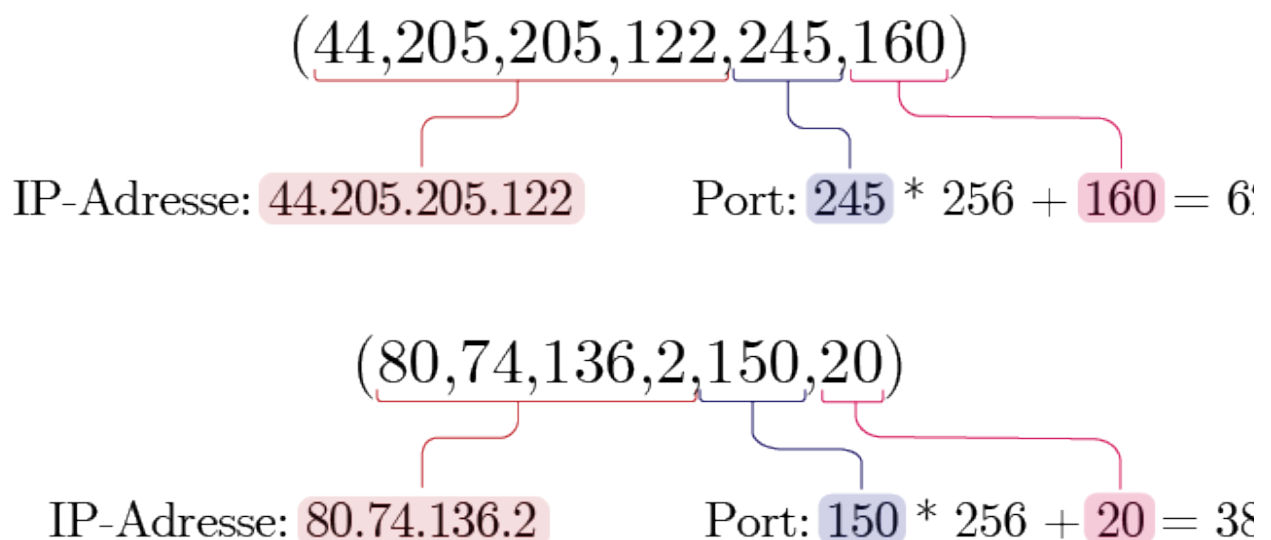
1. Wie beim aktiven FTP wird der **FTP-CLIENT** als erstes eine Authentifizierung vornehmen.
2. Der **FTP-SERVER** bestätigt diese.
3. Nun wird der **FTP-CLIENT** mit einer **PSVM**-Nachricht den passiven-Modus vom **FTP-SERVER** anfordern.
4. Der **FTP-SERVER** bestätigt den passiven Modus mit dem Status 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.

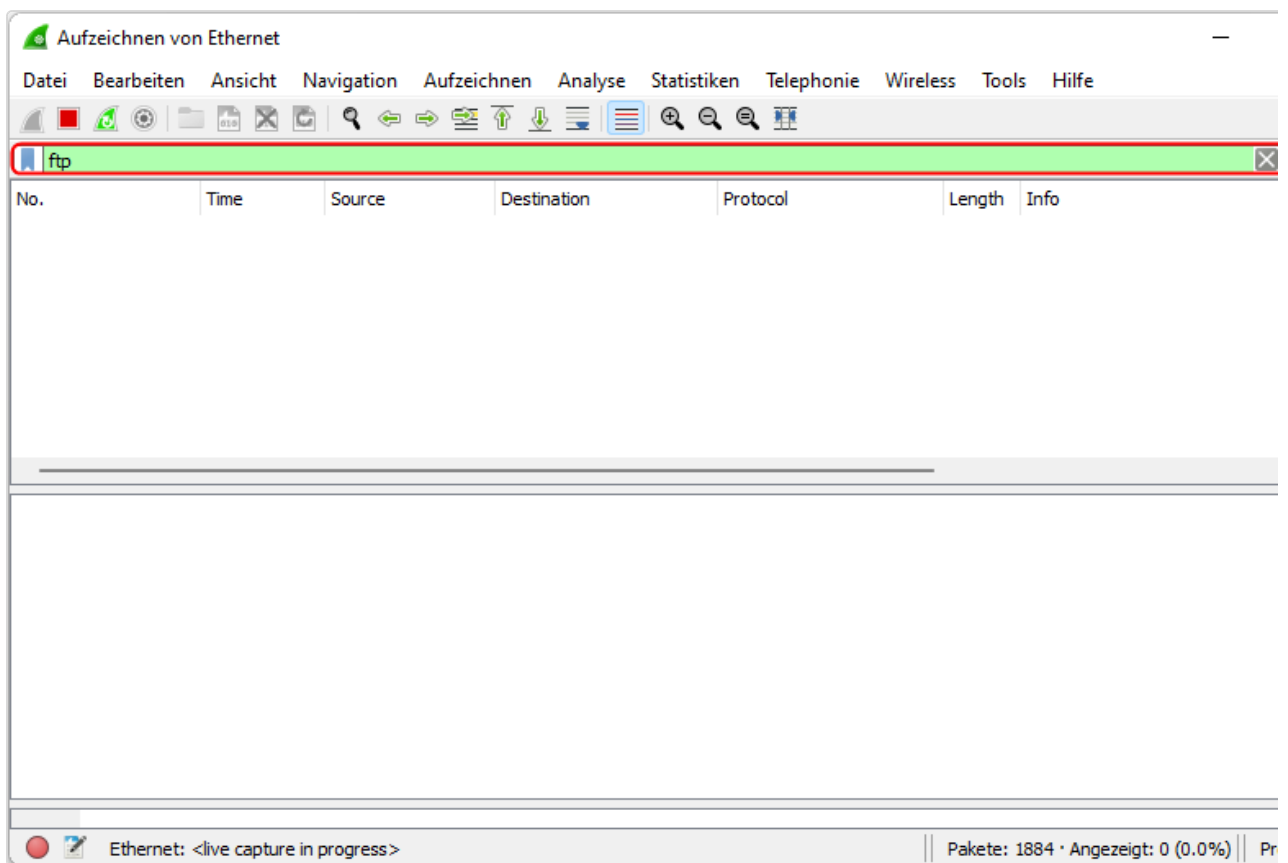
Praktische Übung

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.

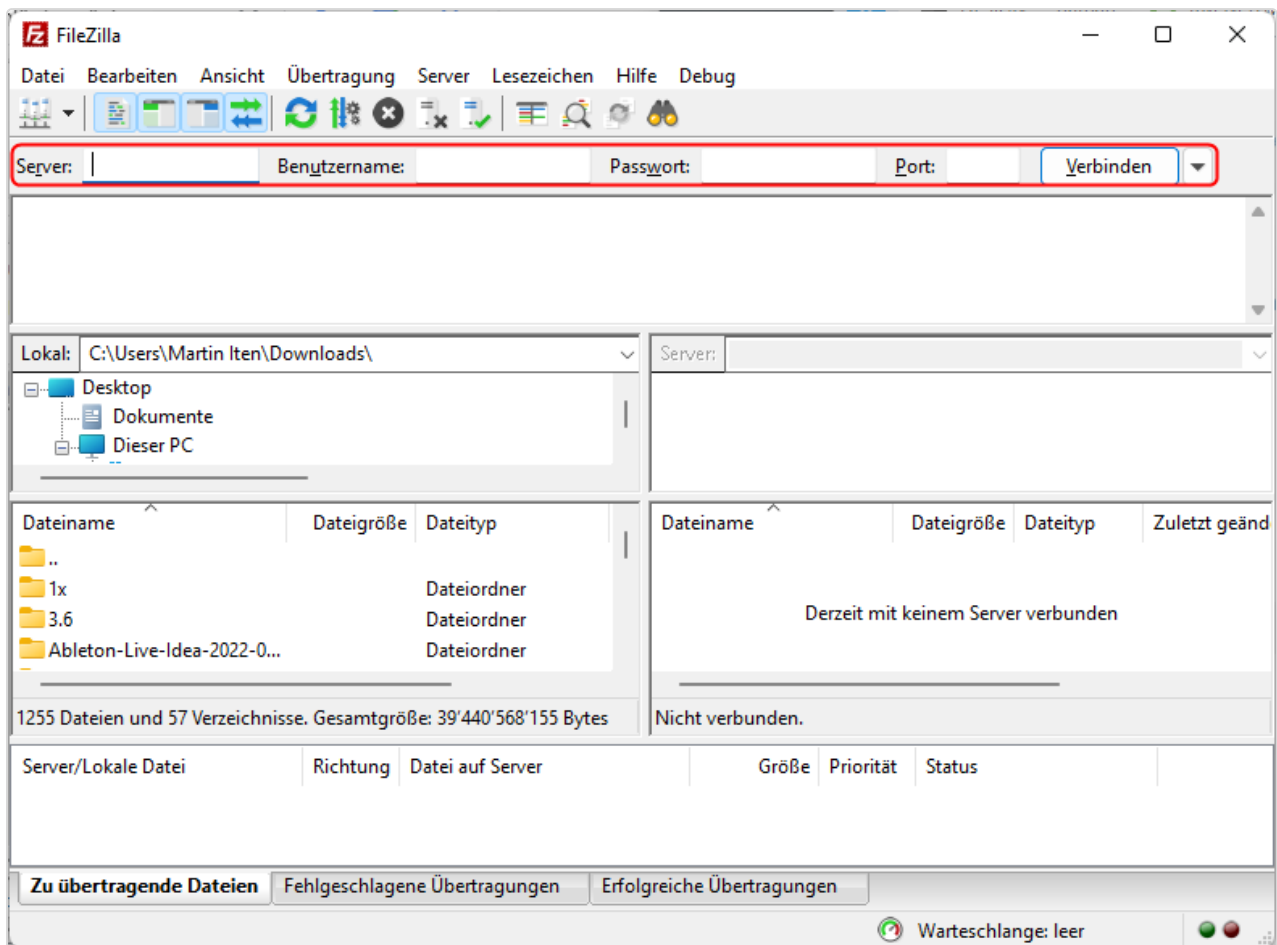
Anmeldung über FileZilla

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

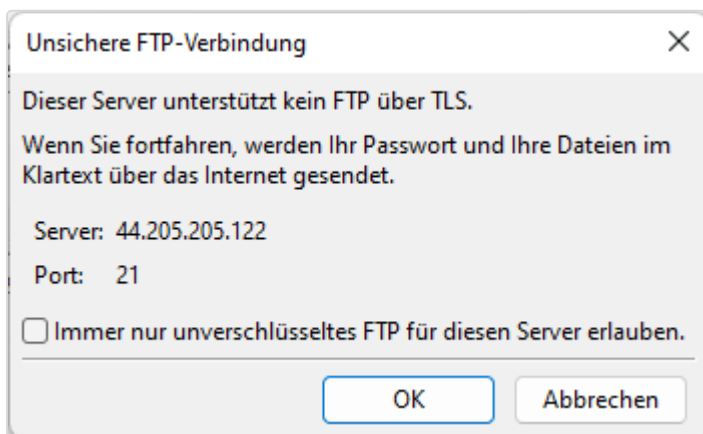


Starten Sie nun die Software FileZilla Client und stellen Sie eine Verbindung mit folgenden Parametern her:

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



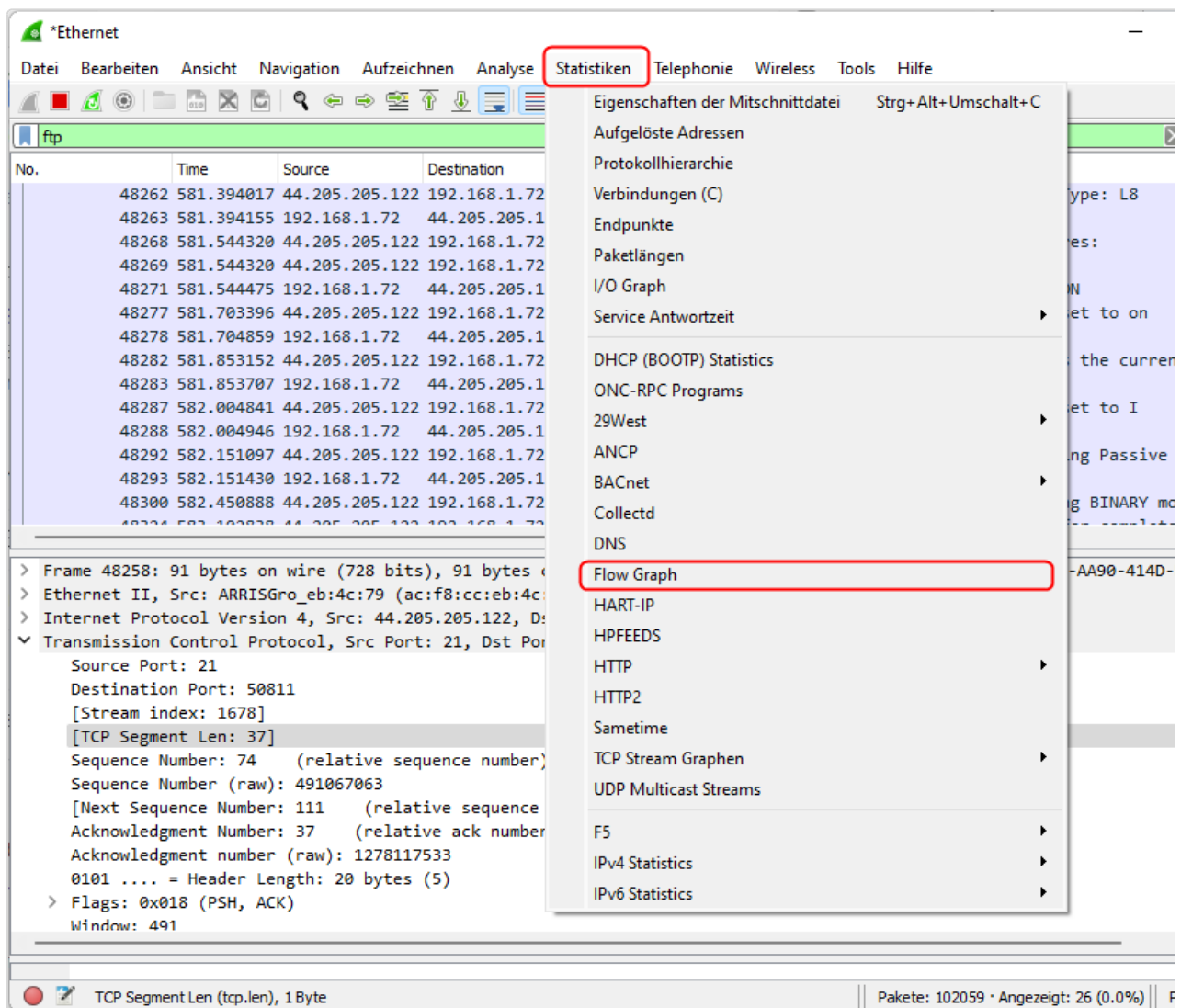
Sie werden die Meldung erhalten, dass dieser FTP-Server kein FTP über TLS bietet. Bestätigen Sie diese Meldung:



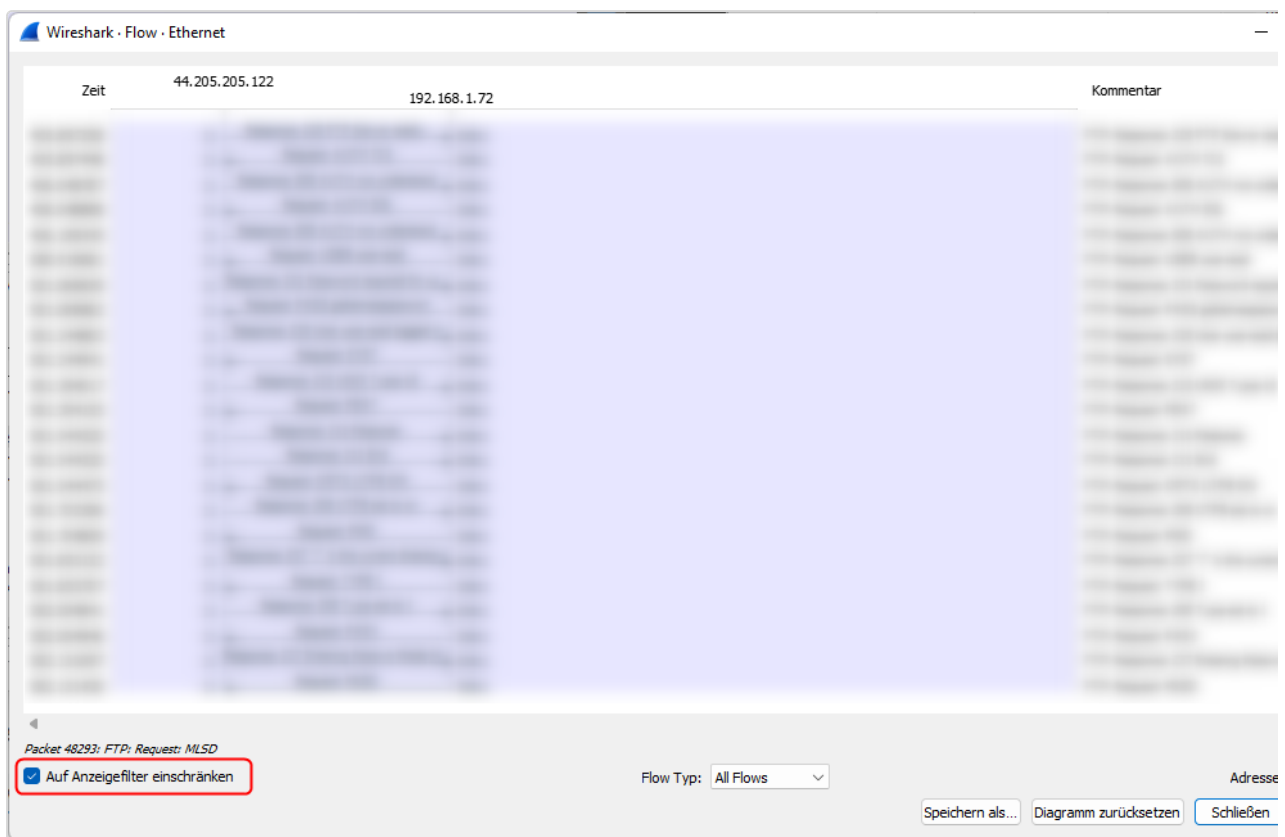
Nun sollten die eine erfolgreiche Verbindung hergestellt haben. **Führen Sie zu diesem Zeitpunkt noch keinen Download aus.** Dies erleichtert Ihnen die Beantwortung der nachfolgenden Fragen.

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.

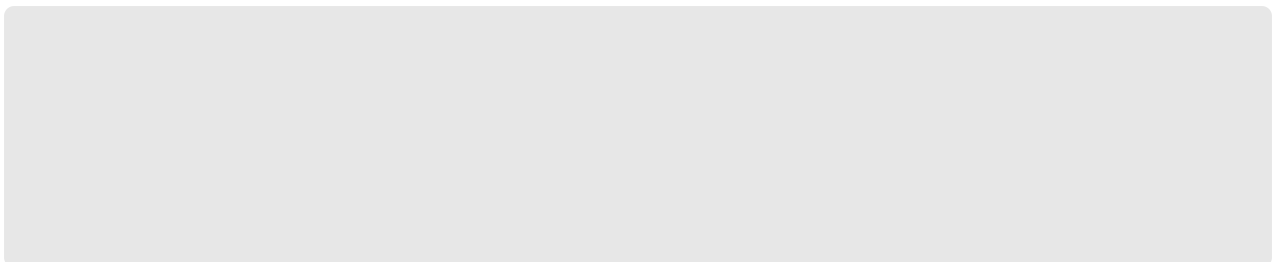


Das allererste Datenpaket sollte nun ein Response sein. Der FTP-Client hat zwar die Verbindung zum Server aufgebaut, aber die initiale Verbindung ist eine generische TCP-Verbindung. Erst mit der ersten Bestätigung durch den FTP-Server werden FTP-Pakete ausgetauscht. Daher wird der initiale Verbindungsaufbau 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 nächstes ob mit einer aktiven oder passive FTP-Verbindung weitergefahren wird. Welche Datenpakete sind hierfür relevant:

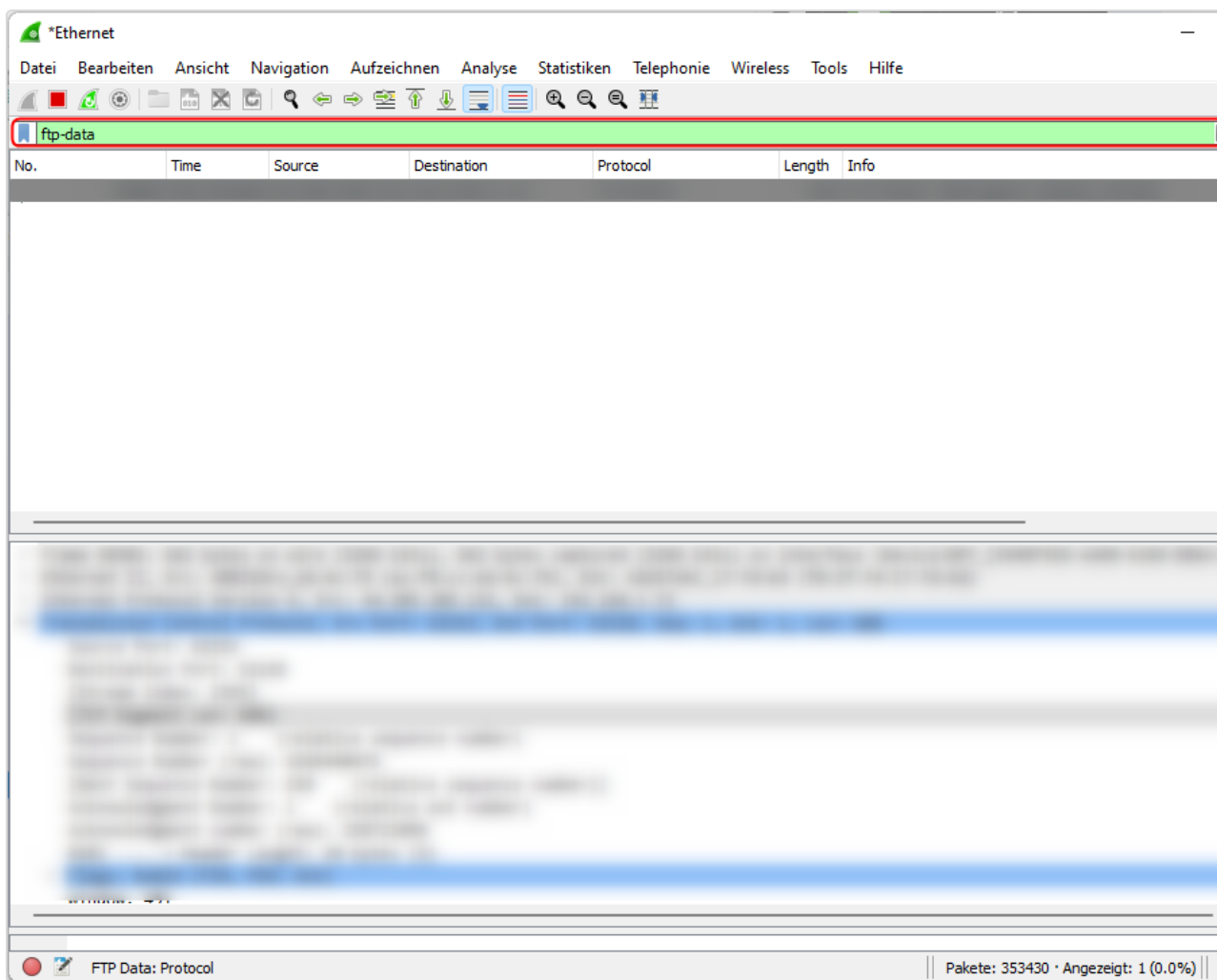
Was für eine FTP-Verbindung wird gesetzt?

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 weiter 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 Befehl 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)

SMB / SAMBA

SMB ist ein Netzwerkprotokoll, welches es Computern ermöglicht, Dateien und andere Ressourcen effizient über Netzwerke zu teilen. SAMBA verwendet das SMB-Netzwerkprotokoll, das auch als Server Message Block oder Common Internet File System bezeichnet wird.

Dieses Protokoll ist um einiges komplexer als die bisher bekannten Protokolle wie TFTP und FTP. Es kann auf Linux, BSD und Microsoft Windows-Betriebssystemen verwendet werden und ist der Standard für Dateifreigabe in Windows-Netzwerken. Dabei ist SAMBA ein freies und quelloffenes Protokoll, das von einigen großen Unternehmen unterstützt wird. Es wird häufig in großen Unternehmensnetzwerken eingesetzt, um ein einheitliches Netzwerk-Dateisystem bereitzustellen. SAMBA kann auch als Netzwerk-Printserver verwendet werden, um Druckaufträge an Drucker im Netzwerk zu senden. Es bringt dabei eine Authentifizierung von Benutzern mit, um genau zu steuern wer auf die auf Netzwerkressourcen zugreifen darf. Dafür wird mit dem bereits bekannten Konzept von Benutzern und Gruppen gearbeitet.

Die Funktionsweise von SMB wird in der praktischen Aufgabe "Auftrag 0003" genauer erläutert.

Auftrag 0003

Auftrag 0003

Der Auftrag

Das Unternehmen **Star Entwicklung AG** möchte die derzeitige Datei-Verwaltung durch eine zentralisierte Server-Lösung ablösen.

Die Lösung soll folgende Möglichkeiten bieten.

- Die Lösung soll direkt Betriebssystem integriert werden.
- Die Mitarbeiter verwenden Windows und Mac Geräte. Diese sollen unterstützt werden.

Diese Funktionen können mit einem SMB-File-Server abgedeckt werden.

- Jeder Benutzer soll ein persönliches Laufwerk für seine Dateien erhalten.
- Es existieren weitere Verzeichnisse, auf jene die Benutzer anhand bestimmter Berechtigungen Zugriff erhalten sollen. Die Geschäftsleitung stellt hierzu weiter unten eine Übersicht bereit.

Der Auftrag erfolgt in drei Schritten:

1. **Konzeptionelles** Erstellen Sie ein Namenskonzept für die Benutzer inkl. eigens definierten Passwörtern und erstellen Sie eine Berechtigungsmatrix für den Dateizugriff.
2. **Installation und Bereitstellung:** Sie installieren die Lösung auf einer AWS-EC2-Instanz, damit diese verwendet werden kann.
3. Erstellung **technische Dokumentation**.

Mitarbeiter und Abteilungen

Nachfolgend finden Sie alle Mitarbeiter im Unternehmen und die Abteilungen, in denen diese arbeiten:

- Dieter Schwartz, Geschäftsleitung
- Stefanie Barth, Geschäftsleitung
- Matthias Köhler, Administration
- Tim Baer, Entwicklung
- Christina Naumann, Entwicklung
- Daniela Hahn, Grafik
- Daniele Hauser, Kundenbetreuung

Es obliegt Ihnen als Unternehmen ein gutes Namens-Konzept auszuarbeiten.

Berechtigungen

Die Geschäftsleitung hat Ihnen wie folgt beschrieben, wie die Berechtigung der einzelnen Benutzer auf die Dateien genau aussehen soll:

- Jeder Benutzer soll die Lese- und Schreibberechtigung auf sein persönliches Verzeichnis haben. Darin sollen die Benutzer die persönlichen Dateien ablegen können. Die Geschäftsleitung hat keinen Zugriff auf die persönlichen Daten der Mitarbeitenden.
- Jede Abteilung hat ein eigenes Verzeichnis in dem die relevanten Daten der Abteilungen liegen. Jede Abteilung soll vollen Zugriff in das eigenen Verzeichnis erhalten. Die Geschäftsleitung hat Lese- und Schreibberechtigung auf alle Abteilungs-Verzeichnisse.
- Die Abteilungen Entwicklung und Grafik haben ein zusätzliches Verzeichnis "projekt_webseite", indem gemeinsam benötigte Dateien abgelegt werden können. Administration und Kundenbetreuung haben auf dieses Verzeichnis keinen Zugriff. Die Geschäftsleitung hat Lese und Schreibzugriff auf diese Freigabe.
- Es existiert das Verzeichnis "info", in dem die Geschäftsleitung regelmässig Informationen für die Mitarbeitenden ablegt. Die Mitarbeitenden sollen diese Informationen nur lesen können. Die Geschäftsleitung benötigt natürlich Schreibberechtigung.
- Im Verzeichnis "share" haben alle Benutzer Zugriff vollen Zugriff.

Erwartete Abgaben

- Ein fertig installierter und getesteter SMB-Server.
- Technische Dokumentation gem. Vorlage als Word oder PDF:
 - [Vorlage_technische_Dokumentation_m123.docx](#)
 - [Anleitung zum Testen von Systemen und Objekten für Berufsschulen.docx](#)
 - In der Dokumentation muss das Namenskonzept als Tabelle / Liste der Benutzer, deren Gruppen inkl. Passwörter enthalten sein.
 - In der Dokumentation muss die Berechtigungsmatrix enthalten sein.

Organisatorisches

Bevor Sie mit der Konfiguration des SMB-Servers beginnen, zeigen Sie der Lehrperson Ihr Namenskonzept und die Berechtigungs-Matrix.

Lohn: 8000

Bonus

Falls Sie alle Aufträge umgesetzt haben können Sie folgende Bonus-Aufgabe in Angriff nehmen.

- Das Verzeichnis Share soll automatisch alle 10 Minuten geleert werden. (Tipp cronjob / rm)

SAMBA Installieren unter Linux

Vorbereitung

Sie benötigen eine neue AWS EC2 Instanz.

1. Erstellen Sie vorgängig eine Sicherheitsgruppe für den SMB-Dienst.

Sicherheitsgruppe erstellen

Folgende Einstellungen müssen gesetzt werden:

Typ	Portbereich	Quelle
SSH		0.0.0.0/0
TCP	139	0.0.0.0/0
SMB		0.0.0.0/0

Regeln für eingehenden Datenverkehr [Info](#)

ID der Sicherheitsgruppenregel	Typ Info	Protokoll Info	Portbereich Info	Quelle Info	Beschreibung – optional Info	
sgr-014953bfb8e89d811	SSH	TCP	22	Benutz... Q 0.0.0.0/0 X		Löschen
sgr-0efdf9e624f93b9fb	Benutzerdefiniertes TCP	TCP	139	Benutz... Q 0.0.0.0/0 X		Löschen
sgr-04488505cfbdc6274	SMB	TCP	445	Benutz... Q 0.0.0.0/0 X		Löschen

2. Nun können Sie eine neue EC2-Instanz mit dieser Sicherheitsgruppe erstellen:

EC2 Instanz anlegen

3. Weisen Sie anschliessend eine statisch IP-Adresse dem neuen Server zu:

Eine statische IP-Adresse zuweisen

Installation von Samba

1. Erstellen Sie eine AWS-EC2 Instanz und verbinden Sie sich mit Ihrem SSH-Client (Alternativ können Sie auch den Webzugriff verwenden)
2. Wechseln Sie mit dem Befehl **sudo su** in den Administrator-Modus.
3. Installieren Sie samba mit folgendem Befehl:

```
yum install samba
```

Verwaltung des Dienstes

Wie beim DNS-Dienst named erfolgt die Verwaltung des Dienstes über systemctl

Wenn Sie wollen, dass der smb-Dienst automatisch startet, wenn der Server aufgestartet wird, dann müssen Sie diesen einmalig aktivieren:

```
systemctl enable smb
```

Dienst starten:

```
systemctl start smb
```

Dienst beenden:

```
systemctl stop smb
```

Dienst neu starten:

```
systemctl stop smb
```

Status des Dienstes ausgeben:

```
systemctl status smb
```


SAMBA Konfigurieren unter Linux

Mit der nachfolgenden Anleitung beschreibt die Grundkonfiguration von SAMBA, einem SMB Server der unter Linux betrieben werden kann.

Sicherheitshinweis: SMB ist ein Protokoll, das es nun schon sehr lange gibt und in den letzten Jahren starke Veränderungen erfahren hat. Ab der SMB Version 3 wurde Verschlüsselung eingeführt. Zuvor war es nicht möglich, den SMB Datenverkehr ausreichend sicher zu verschlüsseln. Eine Nutzung im LAN ist hierbei noch zu vertreten, aber in diesem Setup richten Sie eine SMB-Verbindung über das Internet ein. Dieses Setup darf nicht mit einer SMB Version kleiner als SMBv3.1.1 vorgenommen werden. **SMB1 und SMB2 sind nicht sicher und für den Betrieb über das Internet nicht ausgelegt.**

Die Version SMB3 ist ausreichend sicher und wird auch von Microsoft in der Azure Cloud in dieser Form bereitgestellt. **In diesem Setup wird daher die SMB1 und SMB2 Unterstützung des SAMBA Server deaktiviert. Wird dies vergessen, gehen Sie ein grosses Risiko ein, was entsprechend in der Bewertung der Arbeit sich niederschlagen wird!**

Grundkonfiguration

Sie finden die Konfigurations-Datei unter folgendem Pfad: `/etc/samba/smb.conf`

Die Direktive der SAMBA-Konfiguration sind wie beim BIND Dienst in einzelne Sektionen unterteilt. Diese sind jedoch nicht mit geschweiften Klammern umschlossen. Stattdessen beginne jede Sektion mit einem Titel in eckigen Klammern z. B. die **global**-Sektion:

```
[global]
    workgroup = SAMBA
    security = user

    passdb backend = tdbsam

    printing = cups
    printcap name = cups
    load printers = yes
    cups options = raw
```

Sektion [global]

In der globalen Sektion werden die allgemeinen Einstellungen und Sicherheitsoptionen festgelegt. Sie beginnt, wie der Namen vermuten lässt, mit der Bezeichnung **global**. Diese Bezeichnung ist reserviert und darf nur für die globale Konfiguration verwendet werden. (Es kann keine Freigabe mit diesem Namen erstellt werden). Üblicherweise müssen nur wenige Werte in dieser Sektion angepasst werden.

In der Standard-Konfiguration der **global**-Sektion fehlen ein paar wichtige Punkte, welche zwingend gesetzt werden sollten:

protocol = SMB3	Setzt die Protokoll-Version auf SMB Version 3. Damit können sich zwar ältere Clients nicht mehr verbinden, aber die Verbindung kann dafür sauber verschlüsselt werden. SMB Version 3 wird ab Windows 8 unterstützt. SMT Version 3 wird ab dem Linux Kernel 3.12 unterstützt.
lanman auth = no	Deaktiviert eine unsichere Authentifizierungsmethode
smb encrypt = required	Die Verschlüsselung des SMB Traffics wird erzwungen.
encrypt passwords = yes	Stellt sicher, dass die Passwörter verschlüsselt übertragen werden.
access based share enum = yes	Dadurch werden Freigaben ausgeblendet, wenn keine Berechtigung auf diesen Freigaben existieren. Diese Option ist standardmässig deaktiviert. Somit werden standardmässig dem Benutzer alle Freigaben auf dem SMB-Server angezeigt, auf wenn der Benutzer auf gewisse keinen Zugriff hat. Der Benutzer kann dann diese Freigaben öffnen, erhält aber einen "Zugriff verweigert" Fehler.

Dann gibt es ein paar Punkte, welche wir in der global-Sektion existieren und nicht angepasst werden müssen. Es ist aber wertvoll die Bedeutung dieser Punkte zu verstehen.

workgroup = SAMBA	Wenn der SMB-Server in einer Arbeitsgruppe verwendet wird, müsste hier die korrekte Arbeitsgruppe definiert werden. Dies ist in unserem Fall nicht nötig.
security = user	Definiert, dass die Authentifizierung des SMB-Server gegenüber (vom Client) mittels Benutzername und Passwort erfolgt. Es gibt hier weitere mögliche Optionen: • share: Es wird kein Benutzername benötigt, aber ein Passwort pro Share. • domain: Die Authentifizierung wird über einen Domänencontroller abgewickelt. Der Benutzer muss dann keine Anmeldung vornehmen. • ADS: Die Authentifizierung wird über einen Active Directory Server vorgenommen.
passwd backend = tdbsam	Diese Direktive steuert wo Samba die Passwörter und Benutzerdaten speichert. Diese speichert Samba standardmässig in einer " TDB " (Trivial Database). Die Verwendung dieses Backends erfordert keine zusätzliche Konfiguration. Dieses Backend wird für neue Installationen empfohlen, die kein LDAP benötigen. Wird die Option ldapsam verwendet, dann werden die Benutzerdaten in einem ldap Verzeichnis gespeichert.

Sektion [homes]

Über die homes-Sektion können die Heim-Verzeichnisse der Benutzer aktiviert werden. Jeder Benutzer erhält dadurch automatisch Zugriff auf das Verzeichnis mit seinem Namen.

Die Homes-Verzeichnisse liegen in Linux unter /home und werden automatisch angelegt, wenn ein neuer Benutzer mit **adduser** hinzugefügt wird. Dieser Vorgang ist im Abschnitt "Benutzer hinzufügen" genauer beschrieben.

Standardmässig sieht diese Sektion wie folgt aus:

```
[homes]
comment = Home Directories
valid users = %S
browseable = No
read only = No
inherit acls = Yes
```

Nachfolgend die Erklärung zu den einzelnen Direktiven:

comment = Home Directories	Dies ist ein Kommentar der dieser Freigabe mitgegeben wird.
valid users = %S	Definiert, welche Benutzer auf Ihre Heim-Verzeichnisse zugreifen können. Der Parameter %S wird bei einer Anmeldung automatisch mit dem Benutzername ersetzt. Alternativ kann auch die folgende Option gesetzt werden: %S, %D%w%S D: Name der Domäne oder Arbeitsgruppe des aktuellen Benutzers. w: Winbind Separator. S: Benutzername Dies wird benötigt, wenn die Anmeldung mit dem vollen Domänen-Namen erfolgen soll. In unserem Fall ist dies aber nicht relevant.
browseable = No	Diese Direktive verhindert nicht, wie angenommen werden könnte, dass der Benutzer in seiner Freigabe browsen kann, sondern, dass die Freigabe doppelt sichtbar ist. Einmal unter /homes und einmal unter /benutzername
read only = No	Der SMB-Benutzer erhält Schreibrechte auf diese Freigabe Achtung: alternativ kann auch die Variante writeable = yes verwendet werden.
inherit acls = Yes	Sie kann verwendet werden, um sicherzustellen, dass, wenn Standard-Zugriffsrechte auf übergeordneten Verzeichnissen existieren, diese immer beachtet werden, wenn eine neue Datei oder ein Unterverzeichnis in diesen übergeordneten Verzeichnissen angelegt wird.

Verzeichnis für die Freigabe erstellen

Definieren Sie ein Verzeichnis, in dem Sie die Unterverzeichnisse für die Freigaben einrichten. Erstellen Sie das Verzeichnis /shared im Verzeichnis /var

```
mkdir -p /var/shared
```

Neue Freigabe einrichten

Die oben besprochene Home-Freigabe (Benutzer / Heimatverzeichnis) ermöglichen es dem Benutzer, seine Daten zentral auf dem Server zu organisieren, jedoch bietet diese keine Möglichkeit zum Datenaustausch unter mehreren Benutzern. Die Verzeichnisse sind für andere Benutzer unsichtbar und unerreichbar. Eine Lösung hierfür sind

Eine Freigabe wird eingerichtet, indem ein weiterer Block der Konfiguration hinzugefügt wird:

```
[FREIGABE_NAME]
    path= /var/shared/VERZEICHNIS_NAME
    writeable = yes
```

Standardmässig erhalten alle Benutzer des File-Servers Zugriff, da der Zugriff nicht auf bestimmte Benutzer eingeschränkt ist. Dies kann beispielsweise verwendet werden, um eine unternehmensweite Freigabe zu erstellen.

Gruppenfreigabe

Unter valid users können eine oder mehrere berechtigte Gruppen angegeben werden.

```
[FREIGABE_NAME]
    path= /var/shared/VERZEICHNIS_NAME
    writeable = yes
    valid users = @gruppenname
```

Will man mehreren Gruppen den Zugriff erlauben, dann müssen die Gruppen kommasepariert hinterlegt werden:

```
[FREIGABE_NAME]
    path= /var/shared/VERZEICHNIS_NAME
    writeable = yes
    valid users = @gruppenname1, @gruppenname2
```

Wenn mehrere Gruppen auf eine Freigabe Zugriff haben, muss sichergestellt werden, dass wenn eine neue Datei erstellt wird, die Berechtigung von der Hauptgruppe übernommen wird. (Die Gruppe der auch das Verzeichnis auf dem File-Server gehört), dies wird mit der Direktiven **force group** erreicht. Mit create mask und Directory mask wird zudem sichergestellt, dass die Schreib und Leseberechtigung korrekt gesetzt wird beim Anlegen einer neuen Datei oder eines neuen Verzeichnisses:

```
[FREIGABE_NAME]
    path= /var/shared/VERZEICHNIS_NAME
    writeable = yes
    valid users = @gruppenname1, @gruppenname2
    force group = +gruppenname1
    create mask = 0660
    directory mask = 0770
```

Sollen nur bestimmte Gruppen oder Benutzer Schreibberechtigung auf ein Verzeichnis erhalten, dann muss das Verzeichnis auf readonly gesetzt werden. Jene Benutzer der nun Schreibberechtigung erhalten

soll, muss separat in der Direktive **write list** freigeschaltet werden.

```
[FREIGABE_NAME]
    path= /var/shared/VERZEICHNIS_NAME
    writeable = no
    write list = @gruppenname1
    valid users = @gruppenname1, @gruppenname2
    force group = +gruppenname1
    create mask = 0660
    directory mask = 0770
```

Benutzerverwaltung

Samba verwendet die Berechtigungen der eingerichteten Benutzer auf dem Betriebssystem. Nachfolgend wird beschrieben, wie die Benutzer unter Linux verwaltet werden.

Linux Benutzer hinzufügen

Als erstes muss auf dem Betriebssystem ein Benutzer angelegt werden. Dies wird mit dem Befehl **adduser** erledigt:

```
adduser BENUTZERNAME
```

Nun muss dem neuen Benutzer noch ein Samba-Passwort zugewiesen werden

```
smbpasswd -a BENUTZERNAME
```

Linux Gruppe hinzufügen

Als erstes muss auf dem Betriebssystem eine Gruppe angelegt werden. Dies wird mit dem Befehl **groupadd** erledigt:

```
groupadd BENUTZERNAME
```

Nun kann auf dem Betriebssystem ein Verzeichnis angelegt werden, in dem die Daten dieser Gruppe abgelegt werden:

```
mkdir -p /var/shared/GRUPPE_VERZEICHNIS
```

Anschliessend muss die Gruppe diesem Verzeichnis zugewiesen werden:

```
chgrp GRUPPE /var/shared/GRUPPE_VERZEICHNIS
```

Damit existiert auf dem Betriebssystem nun eine Gruppe mit einem eigenen Verzeichnis. Damit dieses Verzeichnis auch als Freigabe existiert, muss dieses nun noch als Freigabe eingerichtet werden. Dazu wird ein weitere Block eingefügt:

```
[ GRUPPE]  
    path= /var/shared/GRUPPEN_VERZEICHNIS  
    writeable = yes  
    valid users = @GRUPPE
```

Nun kann die Datei gespeichert und der Dienst neu gestartet werden.

Linux Benutzer einer Gruppe hinzufügen:

Um einem Gruppe einen Benutzer hinzuzufügen, wird der Befehl **usermod** verwendet:

```
usermod -a -G gruppenname benutzername
```

Linux Benutzer aus einer Gruppe löschen:

Um einen Benutzer aus einer Gruppe zu löschen, wird der Befehl **gpasswd** verwendet:

```
gpasswd --delete gruppenname benutzername
```

Gruppen eines Benutzers anzeigen:

Wenn Sie wissen möchten, in welchen Gruppen ein Benutzer ist, können Sie den Befehl **groups** verwenden:

```
groups benutzername
```

Konfiguration testen

Mit dem Befehl **testparm** kann die Samba-Konfiguration getestet werden. Ungültige Direktiven und Probleme werden im Bericht ausgegeben:

```
testparm
```

Datei- / Verzeichnisberechtigungen

Damit der Zugriff über die Freigaben auch ordnungsgemäss funktioniert, müssen die freigegebenen auch die korrekten Berechtigungen aufweisen. Dazu müssen Sie wissen, wie Berechtigungen unter Linux aufgebaut sind.

Eine Datei oder ein Verzeichnis gehört grundsätzlich einem **Benutzer** und einer **Gruppe**. Beim Benutzer spricht man auch vom Besitzer oder englisch Owner. Nachfolgend sehen Sie ein die Ausgabe, wenn Sie den Befehl **ls -la** ausführen.

```
[root@ip-172-31-68-46 shares]# ls -la
drwxrw-r--+ 4 miten gruppenname 46 Jan 17 23:20 verzeichnis_name
[] [] ^ ^ ^
| | |
| [] | o- Name des Verzeichnisses oder der Datei
| o----- Name der Gruppe, welche dem Verzeichnis
zugeordnet ist
o----- Name des Benutzers, welchem das Verzeichnis
zugeordnet ist
```

Jedes Verzeichnis und jede Datei hat immer zwingend ein Benutzer und eine Gruppe. Wurde diese vom Root-Benutzer mit der Root-Gruppe erstellt, so sieht die Ausgabe beispielsweise so aus:

```
[root@ip-172-31-68-46 var]# ls -la
drwxr-xr-x 2 root root 19 Dec 15 21:54 account
drwxr-xr-x 2 root root 6 Apr 9 2019 adm
drwxr-xr-x 6 root root 63 Dec 15 21:54 cache
```

Die Berechtigung lässt sich nun über drei Dimensionen steuern:

- Besitzer
- Gruppe
- Andere

Betrachten wir die drei Dimensionen etwas genauer:

Besitzer

Der Besitzer einer Datei oder einem Verzeichnis ist in der Regel der Ersteller der Datei. Die Datei ist primär diesem Benutzer zugeordnet, sprich gehört diesem Benutzer. **Der Besitzer kann immer die Berechtigung steuern / verändern.**

Es kann nun gesteuert werden, was der Benutzer mit dieser Datei oder Verzeichnis machen kann:

Datei:

- **Lesen:** Der Besitzer kann den Inhalt der Datei lesen / einsehen.
- **Schreiben:** Der Besitzer kann den Inhalt der Datei verändern.
- **Ausführen:** Der Besitzer kann eine Datei, welche ein Programm ist, ausführen.

Verzeichnis:

- **Lesen:** Der Besitzer kann den Inhalt in einem Verzeichnis sehen.
- **Schreiben:** Der Besitzer kann im Verzeichnis neue Unterverzeichnisse oder Dateien erstellen
- **Ausführen:** Der Besitzer kann in das Verzeichnis wechseln

Sie können Sich als Besitzer selber die Leserechte auf ein Objekt nehmen. Dann könne Sie die betreffende Datei nicht mehr öffnen. Als Inhaber können Sie die Berechtigung aber jederzeit ändern, somit können Sie sich bei eigenen Dateien nicht aussperren.

Gruppe

Eine Datei kann auch einer speziellen Gruppe zugeordnet werden. Alle Benutzer, welche sich in dieser Gruppe befinden, haben analog den gesetzten Berechtigungen Zugriff auf die Dateien oder Verzeichnisse. **Ein Gruppenmitglied kann aber die Berechtigung einer Datei nicht anpassen, insofern das Gruppenmitglied nicht auch Besitzer ist.**

Jeder Benutzer hat auch immer eine Gruppe, welche auf seinen Namen lautet. Alle seine Dateien sind standardmässig in dieser Gruppe, wenn keine andere Gruppe zugewiesen ist.

Die möglichen Berechtigungen eines Gruppenmitglieds sind identisch mit den Besitzer-Berechtigungen:

- **Lesen**
- **Schreiben**
- **Ausführen**

Andere

Zu guter Letzt, kann noch definiert werden, wie sich eine Datei bei einem Benutzer verhält, welcher nicht Besitzer der Datei oder ein Mitglied der Gruppe ist. Dies kann über die Andere-Berechtigung gesteuert werden. Es gelten wieder die selben Berechtigungen wie beim Inhaber einer Datei oder eines Verzeichnisses:

- **Lesen**
- **Schreiben**
- **Ausführen**

Berechtigungen setzen

Diese Berechtigungen können **beliebig** zusammengesetzt werden. Es gibt hier keine Einschränkung.

Ich kann als Besitzer einer Datei z. B. folgende Berechtigung einer Datei setzen:

- Besitzer : Kann die Datei weder lesen, noch bearbeiten oder ausführen.
- Gruppe: Ein Gruppenmitglied kann die Datei nicht lesen, aber den Inhalt verändern.
- Andere: Jeder andere, welcher nicht Besitzer oder Mitglied der Gruppe ist, kann die Datei lesen.

Wie äussert sich dies nun unter realen Bedingungen?

Der Besitzer kann die Datei weder öffnen noch mit Inhalt beschreiben. Er wird immer ein **Permission denied** (Zugriff verweigert) erhalten.

Ein Mitglied der Gruppe kann die Datei beschreiben, aber nicht lesen. Öffnen das Gruppenmitglied z. B. eine Datei mit nano, dann wird die Datei leer angezeigt, (obwohl eigentlich Inhalt darin ist), das Gruppenmitglied kann nun Inhalt in die Datei schreiben. Beim Speichern wird dann der bereits enthaltene Inhalt mit dem neuen überschrieben.

Ein Benutzer, welcher nicht Besitzer oder Mitglied der Gruppe ist, kann die Datei lesen. Öffnet er die Datei z. B. mit dem Editor nano, dann sieht der Benutzer den Inhalt, kann aber keinen neuen Inhalt hinzufügen.

Wenn Sie nun für diese drei Dimensionen die möglichen Berechtigungen einsetzen, erhalten Sie drei Gruppen, welche jeweils die Berechtigung **L**esen, **S**chreiben und / oder **A**usführen haben können.

Besitzer	Gruppe	Andere
L S A	L S A	L S A

Wenden wir nun die Englischen begriffe für **R**ead, **W**rite und **E**xecute an, dann entsteht folgendes Bild:

Besitzer	Gruppe	Andere
R W X	R W X	R W X

Nun möchten Sie dem Besitzer, der Gruppe und den Anderen die Schreibeberechtigung wegnehmen. Wir streichen also überall das **W**rite.

Besitzer	Gruppe	Andere
R W X	R W X	R W X

In der reinen Darstellung als Text wird eine gestrichene Berechtigung, bez eine nicht gesetzte Berechtigung mit einem Bindestrich dargestellt.

Besitzer	Gruppe	Andere
R - X	R - X	R - X

Geben wir nun dem Besitzer die Schreibeberechtigung wieder zurück, nehmen dafür der Gruppe die Ausführberechtigung und den Anderen alle Berechtigungen, so würde dies folgendes Bild ergeben:

Besitzer	Gruppe	Andere
R W X	R - -	- - -

Nun wurde weiter oben beschrieben, dass sich die Berechtigungen auf Verzeichnisse etwas anders auswirken als auf Dateien. Daher markieren Sie ein Verzeichnis mit einem **D** für Directory und eine Datei mit einem Bindestrich.

	Besitzer	Gruppe	Andere
D	R W X	R - -	- - -
	Besitzer	Gruppe	Andere
-	R W X	R - -	- - -

Nehmen Sie nun nur den Text und Sie haben eine einfache Möglichkeit, die Berechtigung einer Datei oder eines Verzeichnisses abzulesen.

-RWXR--
-RWXR--

Schauen Sie sich nun den **ls -la** vom Anfang nochmals genau an. Sie können nun ablesen, was für eine Berechtigung das Verzeichnis `verzeichnis_name` hat:

```
[root@ip-172-31-68-46 shares]# ls -la
drwxrw-r--  4 miten gruppenname 46 Jan 17 23:20 verzeichnis_name
```

- **Typ** = d = Verzeichnis
- **Besitzer** = rwx = Lesen, Schreiben und Ausführen
- **Gruppe** = rw- = Lesen, Schreiben
- **Andere** = r-- = Lesen

Berechtigungen setzen

Es gibt verschiedene Möglichkeiten eine Berechtigung zu setzen. In jedem Fall wird aber der Befehl **chmod** (**Ch**ange **Mod**us) verwendet. Wir schauen uns in diesem Modul nur die Möglichkeit über rwx die Berechtigung zu setzen an.

Angenommen Sie haben eine Datei mit dem Namen **datei**, welche noch gar keine Berechtigungen hat.

```
[miten@ip-172-31-68-46 Unterverzeichnis]$ ls -la
-----  1 miten miten      0 Jan 18 22:52 datei
```

Nun möchten Sie dem Besitzer die Berechtigung zum Lesen, Schreiben und Ausführen geben, also **rwX**. Verwenden Sie nun den Befehl **chmod** und geben Sie als erstem Parameter die gewünschte Berechtigung an.

u = user = Besitzer

rwX = Berechtigung Lesen, Schreiben und Ausführen

Als zweiten Parameter wird die gewünschte Datei angegeben. Somit sieht der vollständige Befehl wie folgt aus:

```
chmod u=rwX datei
```

Dies resultiert in folgender Berechtigung:

```
[miten@ip-172-31-68-46 Unterverzeichnis]$ ls -la
-rwx-----  1 miten miten      0 Jan 18 22:52 datei
```

Wie sie sehen, hat der Besitzer nun die rwx (Lesen, Schreiben, Ausführen) Berechtigungen erhalten. Nun soll die Gruppe die Berechtigung zum Lesen und Schreiben erhalten.

Dafür wird wieder der Befehl **chmod** verwendet:

g = group = Gruppe

rw = Berechtigung Lesen und Schreiben

```
chmod g=rw datei
```

Dies resultiert in folgender Berechtigung:

```
[miten@ip-172-31-68-46 Unterverzeichnis]$ ls -la
-rwxrw--- 1 miten miten      0 Jan 18 22:52 datei
```

Zu guter Letzt, sollen alle anderen auf die Datei die Berechtigung Lesen und Ausführen erhalten:

o = other= Andere

rx = Berechtigung Lesen und Ausführen

```
chmod o=rx datei
```

Dies resultiert in folgender Berechtigung:

```
[miten@ip-172-31-68-46 Unterverzeichnis]$ ls -la
-rwxrw-r-x 1 miten miten      0 Jan 18 22:52 datei
```

Um etwas schneller vorwärts zu kommen, können Sie die Berechtigungen auch kombinieren:

```
chmod u=rwx,g=rw,o=rx datei
```

Um alle Berechtigungen komplett zu entfernen, setzen Sie einfach keine Berechtigung, also z. B.

```
$ chmod u=rwx,g=,o= datei
$ ls -la
-rwx----- 1 miten miten      0 Jan 18 22:52 datei
```

Typische Berechtigungen

Innerhalb von Samba werden vor allem folgende Berechtigungen benötigt:

Freigabe	Beschreibung	chmod-Befehl
Share (Alle haben Zugriff)	Alle Berechtigungen (rwx) werden für den Besitzer, Gruppe und die Anderen gesetzt.	chmod u=rwx,g=rwx,o=rwx verzeichnis
Gruppen-Freigabe	Der Besitzer und die Gruppen erhalten die volle Berechtigung (rwx). Die anderen erhalten keine Berechtigung (---).	chmod u=rwx,g=rwx,o= verzeichnis
Persönliche Freigabe	Nur der Besitzer erhält die volle Berechtigung (rwx). Die Gruppe und die Anderen erhalten keine Berechtigung (---)	chmod u=rwx,g=,o= verzeichnis

Besitzer ändern / zuweisen

Mit dem Befehl **chown** (**Ch**ange **O**wner) können Sie eine Datei oder ein Verzeichnis einem neuen Benutzer zuordnen. Achtung, eine Datei kann immer nur einem Benutzer gehören. Nur der Root-Benutzer kann die Inhaberschaft einer Datei oder eines Verzeichnisses ändern.

```
# ls -la
drwxrwxrwx  2 miten miten      19 Jan 18 22:55 Unterverzeichnis
```

Angenommen Sie möchten den Inhaber vom Unterverzeichnis auf smeier ändern, dann setzen Sie folgenden Befehl ab:

```
chown smueller Unterverzeichnis
```

Nun hat der Besitzer geändert. Achtung die Gruppe bleibt so unverändert.

```
# ls -la
drwxrwxrwx 2 smueller miten 19 Jan 18 22:55 Unterverzeichnis
```

Gruppe ändern / zuweisen

Mit dem Befehl **chgrp** (**Ch**ange **G**roup) können Sie eine Datei oder ein Verzeichnis einem neuen Benutzer zuordnen.

Nehmen Sie als Ausgangslage das Unterverzeichnis.

```
# ls -la
drwxrwxrwx 2 smueller miten 19 Jan 18 22:55 Unterverzeichnis
```

Nun möchten Sie neu die Gruppe Testgruppe hinterlegen.

```
chgrp testgruppe Unterverzeichnis
```

```
# ls -la
drwxrwxrwx 2 smueller testgruppe 19 Jan 18 22:55 Unterverzeichnis
```

Spezialfall: Mehrere Gruppen pro Verzeichnis

Standardmässig hat eine Datei oder ein Verzeichnis nur eine Gruppe. Diese kann mit `chgrp` geändert werden.

Da aber z. B. mit Samba mehrere Freigaben mit mehreren Gruppen erstellt werden können, benötigen wir eine Möglichkeit mehrere Gruppen einem Verzeichnis zuzuweisen. Dafür können wir die Funktion ACL (Access Control List) verwenden. Damit ist es möglich, einem Verzeichnis mehre Gruppen zuzuweisen:

Hinweis: Definieren sie immer eine Hauptgruppe, welche Sie dem Verzeichnis zuweisen. Die restlichen Gruppen werden anschliessend über die ACL hinzugefügt.

Beispiel: Das Unterverzeichnis gehört der Gruppe **testgruppe**. Diese wurde mit `chgrp` hinzugefügt.

```
# ls -la
drwxrwxrwx 2 smueller testgruppe 19 Jan 18 22:55 Unterverzeichnis
```

Es benötigt nun aber noch eine zweite Gruppe mit dem Namen **video** Zugriff auf das Unterverzeichnis. Diese wird nun mit dem Befehl `setfacl` hinzugefügt:

```
setfacl -m g:video:rwx Unterverzeichnis
```

Erklärung des vollen Befehls:

setfacl	
-m	modify = verändern , damit möchten wir die ACL eines Verzeichnisses verändern.
g:video:rwx	g = group (Wir möchten eine Gruppe hinzufügen) video = Name der Gruppe die hinzugefügt werden soll rwx = Berechtigung die gesetzt werden soll (Lesen, Schreiben, Ausführen)
Unterverzeichnis	Name des Verzeichnisses, welches wir verändern wollen.

Wenn wir nun einen ls -la machen, sieht die Ausgabe immer noch fast identisch aus:

```
# ls -la
drwxrwxrwx+ 2 smueller testgruppe 19 Jan 18 22:55 Unterverzeichnis
```

Die Gruppe hat sich nicht verändert. Es hat jedoch ein + gegeben bei den Berechtigungen. Dies bedeutet, dass dieses Verzeichnis über eine ACL verfügt. Wir können diese mit dem Befehl **getfacl** einsehen:

```
getfacl Unterverzeichnis
# file: Unterverzeichnis
# owner: smueller
# group: testgruppe
user::rwx
group::rwx
group:video:rwx
mask::rwx
other::rwx
```

Man sieht nun, dass neben der Hauptgruppe testgruppe, auch eine Gruppe video mit der Berechtigung rwx existiert.

Löschen einer ACL:

```
setfacl -x g:video Unterverzeichnis
```

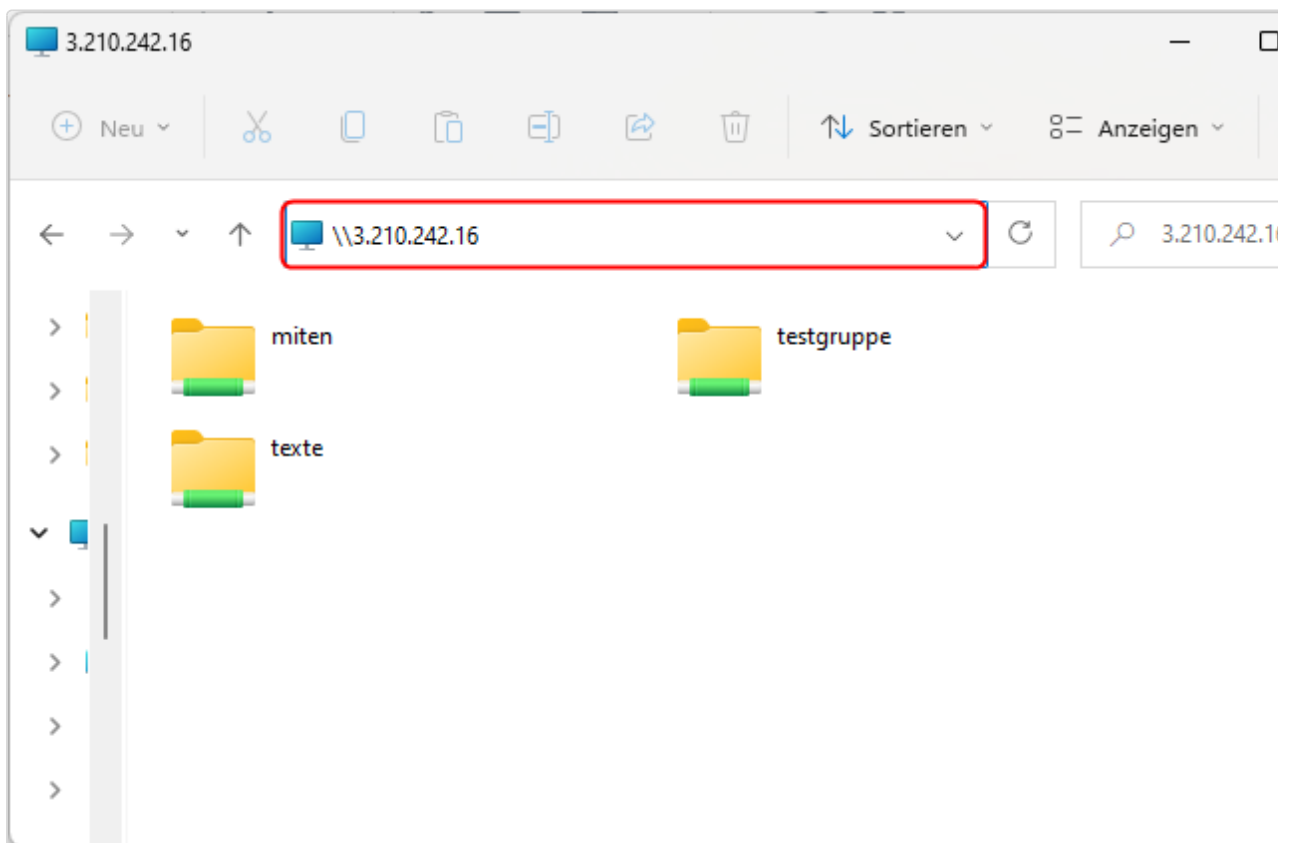
Auftrag 0003

SMB Freigaben unter Windows einrichten

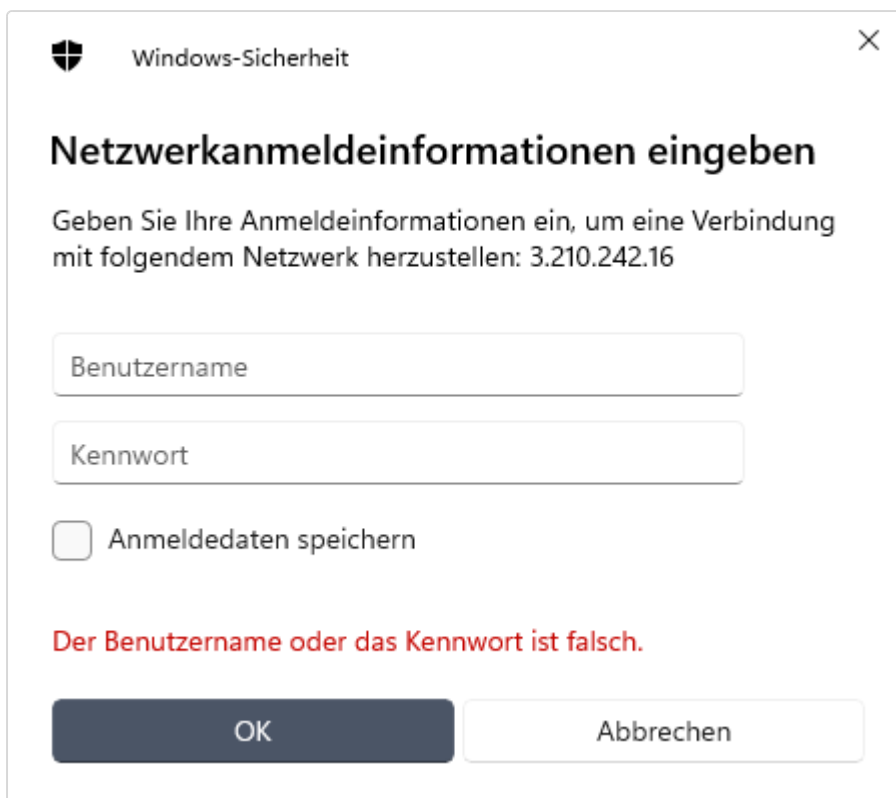
Falls Sie nur kurz eine Freigabe testen möchten, können Sie diese direkt im Windows Explorer öffnen:

Tragen Sie hierzu einfach die IP-Adresse des Servers nach folgendem Schema in der Adresszeile des Explorers ein:

```
\\ip-des-servers
```



Beim ersten Aufruf erscheint eine Aufforderung, dass Passwort und den Benutzernamen zu hinterlegen:



Windows-Sicherheit

Netzwerkanmeldeinformationen eingeben

Geben Sie Ihre Anmeldeinformationen ein, um eine Verbindung mit folgendem Netzwerk herzustellen: 3.210.242.16

Benutzername

Kennwort

☐ Anmeldedaten speichern

Der Benutzername oder das Kennwort ist falsch.

OK Abbrechen

Freigabe schliessen:

Wenn Sie das Explorer-Fenster schliessen und wieder öffnen werden Sie feststellen, dass Sie weiterhin mit dem bereits angemeldeten Benutzer angemeldet sind. Damit Windows die Verbindung sauber trennt, ist ein CMD-Befehl nötig:

```
net use * /delete
```

Danach muss das Explorer-Fenster geschlossen und einige Sekunden abgewartet werden. Danach können Sie die Freigabe wieder aufrufen und es erscheint ein neues Login-Fenster.

Um alle aktiven SMB-Verbindungen aufzulisten, können Sie den nachfolgenden Befehl verwenden:

```
net use
```

Freigabe als Laufwerk einbinden:

Beim Kunden wird die Freigabe schlussendlich als eigenes Laufwerk eingebunden:

