

Le protocole FTP

Le protocole FTP

Objectif

L'objectif de ce cours est de présenter le protocole qui permet le transfert de fichiers entre deux machines.

Différents points seront abordés :

- Rôle et fonctionnement du protocole
- Commandes et réponses
- Mode actif vs Mode passif

Le protocole FTP

Sommaire

- 1. Rôle du protocole FTP**
- 2.Principe de fonctionnement
- 3.Format des commandes et des réponses
- 4.Modes de transfert.
- 5.Conclusion

Introduction

Introduction

- FTP : File Transfer Protocol
 - ↳ protocole de transfert de fichiers
 - ↳ date de 1971 avec la RFC 141
 - ↳ nombreuses RFC ont ensuite apporté des améliorations
 - ↳ actuellement défini par la RFC 3659
- FTP utilise plusieurs connexions TCP
 - ↳ une connexion pour les échanges des commandes : port 21 (coté serveur)
 - ↳ des connexions pour les échanges de données : port 20 (coté serveur)

Introduction

Rôle du protocole FTP

- FTP définit une méthode de transfert de données dans un réseau TCP/IP
- Ces données peuvent être :
 - ↪ binaires : contenu d'un fichier quelconque par exemple (fichier word, image, fichier text)
 - ↪ utiles pour l'utilisateur : contenu d'un répertoire distant
- Avantages :
 - ↪ Permettre un échange de fichiers entre machines distantes
 - ↪ Permettre une indépendance par rapport aux systèmes de fichiers des machines clientes et serveur
 - ↪ Permettre de transférer des données de manière efficace et rapide

5

Le protocole FTP

Sommaire

1. Rôle du protocole FTP
2. Principe de fonctionnement
3. Format des commandes et des réponses
4. Modes de transfert
5. Conclusion

6

Principe de fonctionnement

Modèle Client - Serveur

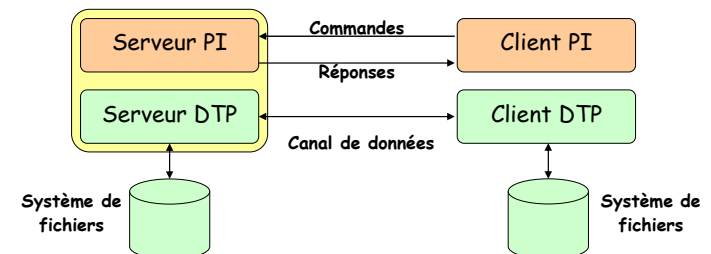
- Une connexion FTP ouvre deux canaux de transmission :
 - ↪ un canal pour les commandes (canal de contrôle)
 - ↪ un canal pour les données
- FTP utilise un modèle client - serveur pour le canal de commandes :
 - ↪ le client envoie des ordres
 - ↪ le serveur attend des requêtes pour effectuer des actions
- Le canal de données est utilisé pour le transfert d'un objet (un contenu de répertoire ou un fichier binaire)
 - ↪ un canal de données par objet transféré

7

Principe de fonctionnement

Communications d'un client vers un serveur

- Le modèle de communication entre un client FTP et un serveur FTP utilise deux concepts :
 - ↪ PI : l'interpréteur de commandes du protocole (Protocol Interpreter)
 - ↪ DTP : le processus de transfert (Data Transfer Process)
- Ils communiquent selon le schéma suivant :



8

Principe de fonctionnement

Communications client -> serveur

- La liaison de commandes véhicule :
 - les commandes définies par le protocole
 - les commandes propres aux extensions proposées par l'implémentation courante du serveur
 - les réponses aux commandes
- La liaison de données transporte les données proprement dites (contenu des objets à échanger)
- Deux modes d'échanges sont possibles :
 - ↪ mode BINAIRE : les données sont échangées sans chercher à les modifier. Le fichier destination est exactement le même que le fichier source.
 - ↪ mode ASCII : les données sont échangées en cherchant à les modifier. Il y a adaptation des "retours à la ligne" entre le mode Windows (deux caractères pour un retour ligne) et le mode Unix (un seul caractère).

9

Le protocole FTP

Sommaire

1. Rôle du protocole FTP
2. Principe de fonctionnement
3. Format des commandes et des réponses
4. Modes de transfert
5. Conclusion

10

Commandes et Réponses

Commandes FTP

- Les commandes standards sont regroupées en différentes catégories :

↪ Commandes de contrôle d'accès :

| | |
|------|---|
| USER | Identification de l'utilisateur. |
| PASS | Mot de passe de l'utilisateur. |
| CWD | Permet à l'utilisateur de se positionner dans la hiérarchie du serveur. |
| CDUP | Commande générique permettant de remonter dans l'arborescence. |
| QUIT | Termine la session. |

↪ Commandes de paramétrage des transferts :

| | |
|------|--|
| PORT | Déclare le point de connexion client de la liaison de données. |
| PASV | Permet de renverser le comportement du serveur sur la liaison de données (serveur passif). |
| TYPE | Permet de contrôler le format de représentation utilisé pour les échanges. |

11

Commandes et Réponses

Commandes FTP

↪ Commandes de service :

| | |
|------|--|
| RETR | Active le transfert du contenu d'un fichier à travers la liaison de données du serveur vers le client. |
| STOR | Active le transfert du contenu d'un fichier à travers la liaison de données du client vers le serveur. |
| REST | Permet de reprendre un transfert sur un point de reprise. |
| RNFR | Permet de renommer un fichier en précisant l'ancien nom. |
| RNTO | Permet de renommer un fichier en précisant le nouveau nom. |
| DELE | Permet la suppression d'un fichier sur le serveur. |
| RMD | Permet la suppression d'un répertoire sur le serveur. |
| MKD | Permet la création d'un répertoire sur le serveur. |

12

Commandes et Réponses

Commandes FTP

| | |
|------|--|
| PWD | Interroge le serveur pour connaître son répertoire courant. |
| LIST | Permet de consulter la liste des fichiers d'un répertoire du serveur. Le format des données renvoyées n'est pas normalisé, car cette commande est libre d'implémentation. |
| NLST | Permet de consulter la liste des fichiers d'un répertoire du serveur. Le format des données est normalisé, ce qui permet à un programme client de réaliser des traitements automatiques de la réponse. |
| SYST | Permet d'obtenir des informations sur le système hôte. |
| STAT | Permet d'obtenir des informations sur un transfert en cours (s'il y en a un), sur l'état du serveur ou sur un fichier particulier. |
| HELP | Permet d'obtenir de l'aide sur les commandes implantées sur le serveur. |
| NOOP | Cette commande ne fait que demander au serveur de répondre positivement. |

13

Commandes et Réponses

Réponses FTP

- Chaque commande FTP reçoit en retour une réponse.
- Une réponse est constituée de :
 - ↳ un code de trois chiffres (code de retour : « xyz »).
 - ↳ caractère "espace".
 - ↳ une ligne de texte à caractère informationnel (sauf pour informer le client FTP d'une adresse IP et d'un numéro de port).

14

Commandes et Réponses

Réponses FTP

- Le code de retour a une sémantique particulière, le premier chiffre « x » peut prendre cinq valeurs différentes :
 - ↳ **1yz** : Réponse positive et préliminaire. L'opération demandée est initiée, mais il faut attendre une réponse définitive pour procéder à une autre commande.
 - ↳ **2yz** : Réponse positive et définitive. L'opération en question s'est normalement terminée.
 - ↳ **3yz** : Réponse positive et intermédiaire. L'opération a été acceptée, mais ne peut-être réalisée par manque d'information. L'utilisateur doit alors y répondre en envoyant une autre commande.
 - ↳ **4yz** : Réponse négative et temporaire. L'opération a échoué mais les causes sont temporaires et l'utilisateur est encouragé à recommencer.
 - ↳ **5yz** : Réponse négative et définitive. L'opération a échoué et les causes de cet échec sont permanentes.

15

Commandes et Réponses

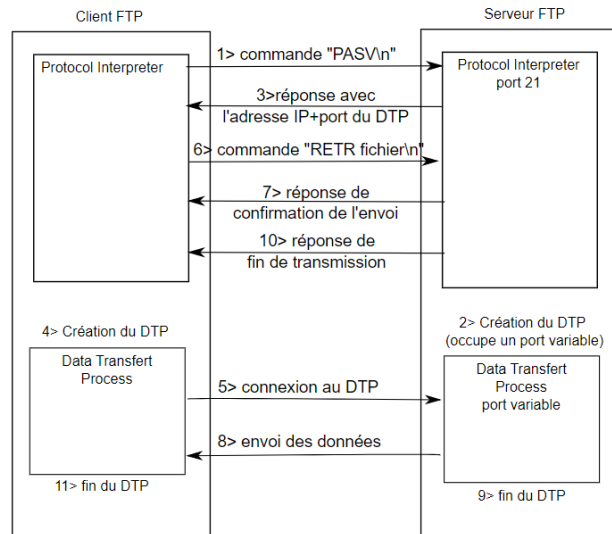
Réponses FTP

- Le second chiffre « y » a la signification suivante :
 - ↳ **x0z** : Syntaxe. Réponses relatives aux erreurs de syntaxe, ou à des commandes syntaxiquement correctes mais n'étant pas représentées dans les autres catégories fonctionnelles
 - ↳ **x1z** : Information. Réponses relatives à des requêtes informationnelles.
 - ↳ **x2z** : Connexion. Réponses relatives à des commandes de contrôle des liaisons.
 - ↳ **x3z** : Contrôle d'accès. Réponses relatives aux processus d'identification.
 - ↳ **x4z** : Non spécifié.
 - ↳ **x5z** : Système de fichier. Réponses relatives aux opérations réalisées sur le système de fichier.
- Le troisième chiffre z est un simple raffinement sémantique du second.

16

Commandes et Réponses

Exemple (<http://www-inf.telecom-sudparis.eu/>)



17

Le protocole FTP

Sommaire

1. Rôle du protocole FTP
2. Principe de fonctionnement
3. Format des commandes et des réponses
4. Modes de transfert
5. Conclusion

18

Modes de Transfert

Différents modes de transfert

- FTP utilise deux connexions TCP en parallèle :
 - ↪ une pour le canal de commandes : port 21.
 - ↪ une pour le canal de données : port 20.
- Fonctionnement du canal de commandes : le client se connecte au serveur.
- Le canal de transfert de données peut fonctionner selon deux modes différents :
 - ↪ Actif
 - ↪ Passif

19

Modes de Transfert

Mode passif

| | | | | | |
|----|----------|-------------------|-------------------|--------|---|
| 1 | 0.000000 | 00:0c:6d:ba:c5:ec | 00:c0:9f:1c:1a:9b | ARP | who has 192.168.0.1 is at 00:0c:6d:ba:c5:ec |
| 2 | 0.000226 | 00:0c:6d:ba:c5:ec | 00:c0:9f:1c:1a:9b | ARP | 192.168.0.1 is at 00:0c:6d:ba:c5:ec |
| 3 | 0.000252 | 192.168.0.5 | 192.168.0.1 | TCP | 1080 > 21 [SYN, Seq=0 Len=0 MSS=1460 |
| 4 | 0.000672 | 192.168.0.1 | 192.168.0.5 | TCP | 21 > 1080 [SYN, ACK] Seq=0 Ack=1 win=6 |
| 5 | 0.000704 | 192.168.0.5 | 192.168.0.1 | TCP | 1080 > 21 [ACK] Seq=1 Ack=1 win=17520 |
| 6 | 0.266036 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 220-GuIdFtP FTP Server (C) |
| 7 | 0.267007 | 192.168.0.1 | 192.168.0.5 | NBNS | Name query NBSTAT *<00><00><00><00> |
| 8 | 0.451878 | 192.168.0.5 | 192.168.0.1 | TCP | 1080 > 21 [ACK] Seq=1 Ack=63 win=17458 |
| 9 | 0.452217 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 220 Please enter your name: |
| 10 | 0.454222 | 192.168.0.5 | 192.168.0.1 | FTP | Request: USER anonymous |
| 11 | 0.625105 | 192.168.0.1 | 192.168.0.5 | TCP | 21 > 1080 [ACK] Seq=92 Ack=17 win=6552 |
| 12 | 1.766722 | 192.168.0.1 | 192.168.0.5 | NBNS | Name query NBSTAT *<00><00><00><00> |
| 13 | 3.266726 | 192.168.0.1 | 192.168.0.5 | NBNS | Name query NBSTAT *<00><00><00><00> |
| 14 | 4.792863 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 331 User name okay, Need pas: |
| 15 | 4.794375 | 192.168.0.5 | 192.168.0.1 | FTP | Request: PASS anonymous |
| 16 | 4.950335 | 192.168.0.1 | 192.168.0.5 | TCP | 21 > 1080 [ACK] Seq=128 Ack=33 win=6552 |
| 17 | 5.067071 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 230 User logged in. |
| 18 | 5.069695 | 192.168.0.5 | 192.168.0.1 | FTP | Request: PWD |
| 19 | 5.097774 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 257 "/" is current directory |
| 20 | 5.099631 | 192.168.0.5 | 192.168.0.1 | FTP | Request: SYST |
| 21 | 5.125482 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 215 UNIX Type: L8 Server |
| 22 | 5.128479 | 192.168.0.5 | 192.168.0.1 | FTP | Request: PASV |
| 23 | 5.213913 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 227 Entering Passive Mode (1 |
| 24 | 5.216659 | 192.168.0.5 | 192.168.0.1 | TCP | 1081 > 20100 [SYN] Seq=0 Len=0 MSS=1460 |
| 25 | 5.217058 | 192.168.0.1 | 192.168.0.5 | TCP | 20100 > 1081 [ACK] Seq=0 Ack=1 win=6 |
| 26 | 5.217085 | 192.168.0.5 | 192.168.0.1 | TCP | 1081 > 20100 [ACK] Seq=1 Ack=1 win=175 |
| 27 | 5.218372 | 192.168.0.5 | 192.168.0.1 | FTP | Request: LIST |
| 28 | 5.269523 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 150 opening ASCII mode data |
| 29 | 5.270785 | 192.168.0.1 | 192.168.0.5 | FTP-DA | FTP Data: 192 Bytes |
| 30 | 5.270848 | 192.168.0.1 | 192.168.0.5 | TCP | 20100 > 1081 [FIN, ACK] Seq=193 Ack=1 |
| 31 | 5.270877 | 192.168.0.5 | 192.168.0.1 | TCP | 1081 > 20100 [ACK] Seq=1 Ack=194 win=1 |
| 32 | 5.298252 | 192.168.0.5 | 192.168.0.1 | TCP | 1081 > 20100 [FIN, ACK] Seq=1 Ack=194 |
| 33 | 5.298551 | 192.168.0.1 | 192.168.0.5 | TCP | 20100 > 1081 [ACK] Seq=194 Ack=2 win=6 |
| 34 | 5.459100 | 192.168.0.5 | 192.168.0.1 | TCP | 1080 > 21 [ACK] Seq=56 Ack=319 win=172 |
| 35 | 5.459436 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 226 Transfer successful. |
| 36 | 5.659371 | 192.168.0.5 | 192.168.0.1 | TCP | 1080 > 21 [ACK] Seq=56 Ack=345 win=171 |
| 37 | 8.530432 | 192.168.0.5 | 192.168.0.1 | FTP | Request: QUIT |
| 38 | 8.602821 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 221 Goodbye. Control connec |
| 39 | 8.603117 | 192.168.0.5 | 192.168.0.1 | TCP | 21 > 1080 [FIN, ACK] Seq=387 Ack=62 win= |
| 40 | 8.603148 | 192.168.0.5 | 192.168.0.1 | TCP | 1080 > 21 [ACK] Seq=62 Ack=388 win=171 |
| 41 | 8.603278 | 192.168.0.5 | 192.168.0.1 | TCP | 1080 > 21 [RST, ACK] Seq=62 Ack=388 win= |

ARP

Ouv TCP

Auth

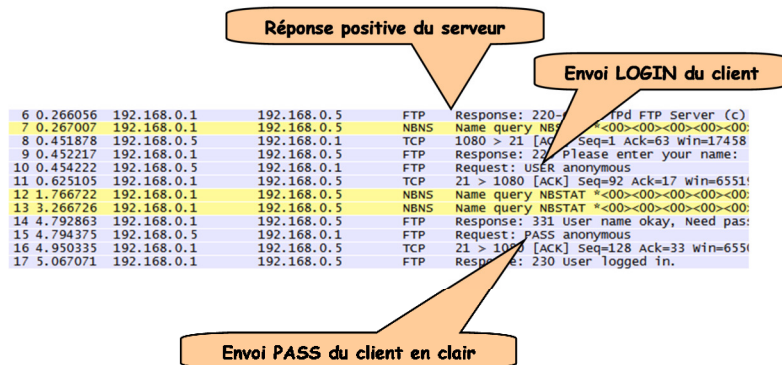
Trans

Ferm

20

Modes de Transfert

Mode passif - Authentification



21

Modes de Transfert

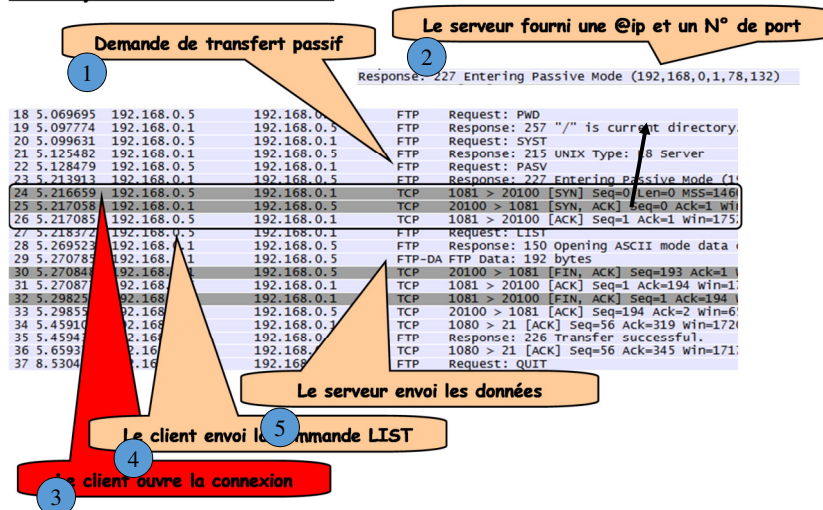
Mode passif - Authentification

- Phase d'authentification : le serveur vérifie l'identité du client qui s'y connecte.
- Très peu de commandes sont acceptées par le serveur :
 - ↳ USER et PASS.
- Par défaut, l'authentification se fait en clair (pas sécurisée).
- Il existe des améliorations pour une authentification en mode chiffré.

22

Modes de Transfert

Mode passif - Transaction



23

Modes de Transfert

Mode passif - Transaction

- En mode passif, le serveur est passif pour les ouvertures de connexions. Cela veut dire que c'est le client qui ouvre les différentes connexions.
- Quand le client désire récupérer des données (par exemple, la liste des fichiers d'un répertoire), il effectue les étapes suivantes :
 - ✓ C → S : Demande de transfert en mode passif.
 - ✓ S → C : Le serveur répond en donnant une adresse IP et un N° de port.
 - ✓ C → S : Le client ouvre la connexion.
 - ✓ C → S : Le client envoie la commande (LIST dans l'exemple).
 - ✓ S → C : Le serveur envoie les données.
 - ✓ S → C : Le serveur ferme la connexion.

24

Modes de Transfert

Mode passif - Transaction

➤ Les conséquences de ce mode de fonctionnement sont multiples :

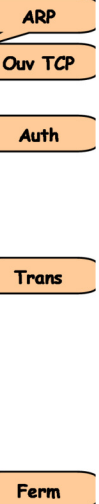
- Le client ouvre les connexions, donc cela permet au client de fonctionner à travers une translation d'adresse dynamique. Le client peut être sur un réseau privé avec un mécanisme de NAT.
- Les données du protocole fournies par le serveur (couche 7) transportent des informations relatives à la couche 3 (une adresse IP) et à la couche 4 (un numéro de port). Donc, il est beaucoup plus difficile de mettre le serveur avec un adressage privé (même dans le cas d'une translation d'adresse statique).
- Le serveur peut envoyer une adresse IP qui n'est pas la sienne. Les données peuvent venir d'une autre machine.

25

Modes de Transfert

Mode actif

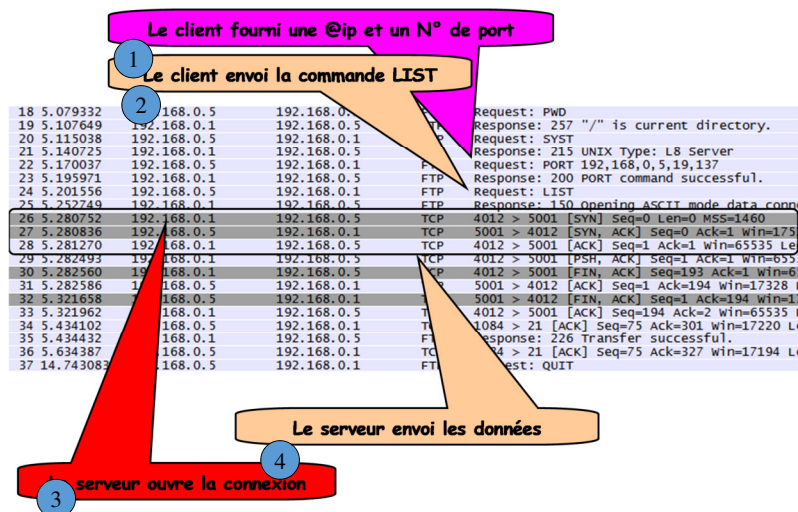
| | | | | | |
|----|-----------|-------------------|-------------------|------|--|
| 1 | 0.000000 | 00:c0:9f:1c:1a:9b | ff:ff:ff:ff:ff:ff | ARP | who has 192.168.0.1? Tell 192.168.0.5 |
| 2 | 0.000224 | 00:c0:9f:1c:1a:9b | 00:c0:9f:1c:1a:9b | ARP | 192.168.0.1 is at 00:c0:9f:1c:1a:9b |
| 3 | 0.000224 | 192.168.0.5 | 192.168.0.1 | TCP | 1084 > 21 [SYN] Seq=0 Len=0 MSS=1460 |
| 4 | 0.000655 | 192.168.0.1 | 192.168.0.5 | TCP | 21 > 1084 [SYN, ACK] Seq=0 Ack=1 win=65535 |
| 5 | 0.000688 | 192.168.0.5 | 192.168.0.1 | TCP | 1084 > 21 [ACK] Seq=1 Ack=1 win=17520 Len=0 |
| 6 | 0.266749 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 220-GUIDFTPd FTP Server (C) 199 |
| 7 | 0.267735 | 192.168.0.1 | 192.168.0.5 | NBNS | Name query NBSTAT <00><00><00><00><00> |
| 8 | 0.426894 | 192.168.0.5 | 192.168.0.1 | TCP | 1084 > 21 [ACK] Seq=1 Ack=63 win=17458 Len=0 |
| 9 | 0.427227 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 220 Please enter your name: |
| 10 | 0.429375 | 192.168.0.5 | 192.168.0.1 | FTP | Request: USER anonymous |
| 11 | 0.548405 | 192.168.0.1 | 192.168.0.5 | TCP | 21 > 1084 [ACK] Seq=92 Ack=17 win=65535 Len=0 |
| 12 | 1.767177 | 192.168.0.1 | 192.168.0.5 | NBNS | Name query NBSTAT <00><00><00><00><00> |
| 13 | 3.267185 | 192.168.0.1 | 192.168.0.5 | NBNS | Name query NBSTAT <00><00><00><00><00> |
| 14 | 4.794253 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 331 User name okay, Need password |
| 15 | 4.795783 | 192.168.0.5 | 192.168.0.1 | FTP | Request: PASS anonymous |
| 16 | 4.974212 | 192.168.0.1 | 192.168.0.5 | TCP | 21 > 1084 [ACK] Seq=128 Ack=33 win=65503 Len=0 |
| 17 | 5.076947 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 230 User logged in. |
| 18 | 5.079332 | 192.168.0.5 | 192.168.0.1 | FTP | Request: PWD |
| 19 | 5.107649 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 257 "/" is current directory. |
| 20 | 5.115038 | 192.168.0.5 | 192.168.0.1 | FTP | Request: SYST |
| 21 | 5.140725 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 215 UNIX Type: L8 Server |
| 22 | 5.170037 | 192.168.0.5 | 192.168.0.1 | FTP | Request: PORT 192,168,0,5,19,137 |
| 23 | 5.195971 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 200 PORT command successful. |
| 24 | 5.201556 | 192.168.0.5 | 192.168.0.1 | FTP | Request: LIST |
| 25 | 5.252749 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 150 Opening ASCII mode data conn |
| 26 | 5.280752 | 192.168.0.1 | 192.168.0.5 | TCP | 4012 > 5001 [SYN] Seq=0 Len=0 MSS=1460 |
| 27 | 5.280836 | 192.168.0.5 | 192.168.0.1 | TCP | 5001 > 4012 [SYN, ACK] Seq=0 Ack=1 win=175 |
| 28 | 5.281270 | 192.168.0.1 | 192.168.0.5 | TCP | 4012 > 5001 [ACK] Seq=1 Ack=1 win=65535 Le |
| 29 | 5.282493 | 192.168.0.1 | 192.168.0.5 | TCP | 4012 > 5001 [PSH, ACK] Seq=1 Ack=1 win=655 |
| 30 | 5.282560 | 192.168.0.1 | 192.168.0.5 | TCP | 4012 > 5001 [FIN, ACK] Seq=192 Ack=1 win=6 |
| 31 | 5.282586 | 192.168.0.5 | 192.168.0.1 | TCP | 5001 > 4012 [ACK] Seq=1 Ack=194 win=17328 |
| 32 | 5.321658 | 192.168.0.5 | 192.168.0.1 | TCP | 5001 > 4012 [FIN, ACK] Seq=1 Ack=194 win=1 |
| 33 | 5.321962 | 192.168.0.1 | 192.168.0.5 | TCP | 4012 > 5001 [ACK] Seq=194 Ack=2 win=65535 |
| 34 | 5.434102 | 192.168.0.5 | 192.168.0.1 | TCP | 1084 > 21 [ACK] Seq=75 Ack=301 win=17220 L |
| 35 | 5.434432 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 226 Transfer successful. |
| 36 | 5.634387 | 192.168.0.5 | 192.168.0.1 | TCP | 1084 > 21 [ACK] Seq=75 Ack=327 win=17194 |
| 37 | 14.743083 | 192.168.0.5 | 192.168.0.1 | FTP | Request: QUIT |
| 38 | 14.803146 | 192.168.0.1 | 192.168.0.5 | FTP | Response: 221 Goodbye. Control connection |
| 39 | 14.803432 | 192.168.0.1 | 192.168.0.5 | TCP | 21 > 1084 [FIN, ACK] Seq=369 Ack=81 win=65 |
| 40 | 14.803468 | 192.168.0.5 | 192.168.0.1 | TCP | 1084 > 21 [ACK] Seq=81 Ack=370 win=17152 |
| 41 | 14.804230 | 192.168.0.5 | 192.168.0.1 | TCP | 1084 > 21 [RST, ACK] Seq=81 Ack=370 win=0 |



26

Modes de Transfert

Mode actif - Transaction



27

Modes de Transfert

Mode actif - Transaction

- Le serveur est actif pour les ouvertures de connexions.
- Le serveur ouvre les différentes connexions pour le transfert des données.
- Quand le client désire récupérer des données (par exemple, la liste des fichiers d'un répertoire), il effectue les étapes suivantes :
 - C → S : Demande de transfert en mode actif. Le client fournit une adresse IP et un numéro de port.
 - C → S : Le client envoie la commande (LIST dans l'exemple).
 - S → C : Le serveur ouvre la connexion.
 - S → C : Le serveur envoie les données.
 - S → C : Le serveur ferme la connexion.

28

Modes de Transfert

Mode actif - Transaction

➤ Les conséquences de ce mode de fonctionnement sont multiples :

- ↪ Le serveur ouvre les connexions, donc le client ne peut pas se situer derrière un mécanisme de translation d'adresses.
- ↪ Les données du protocole fournies par le client (couche 7) transporte des informations relatives à la couche 3 (une adresse IP) et à la couche 4 (un numéro de port). Donc, il est beaucoup plus difficile de mettre le serveur avec un adressage privé (même dans le cas d'une translation d'adresse statique).
- ↪ Le client peut envoyer une adresse IP qui n'est pas la sienne. Les données peuvent aller vers une autre machine.

29

Modes de Transfert

Actif vs Passif

Le paramétrage d'un serveur et d'un client FTP est important :

- ↪ Quand le serveur et le client se trouvent dans un réseau IP publique, le paramétrage est plus simple. Les adresses IP du serveur et du client sont les adresses publiques. Les ouvertures de connexions peuvent se faire indifféremment par le client ou le serveur.
- ↪ Quand le client est dans un réseau privé naté, il est nécessaire que l'échange se fasse de façon passive (c'est le client qui ouvre les connexions).
- ↪ Quand le serveur est dans un réseau privé naté, il doit y avoir une translation sur le port 21 de manière à ce que le client puisse se connecter au serveur. Il est beaucoup plus simple que le transfert se fasse de façon active.
- ↪ Quand le serveur et le client se trouvent dans un réseau privée naté, c'est beaucoup plus complexe à configurer.

30

Le protocole FTP

Sommaire

1. Rôle du protocole FTP
2. Principe de fonctionnement
3. Format des commandes et des réponses
4. Modes de transfert
5. Conclusion

31

Conclusion

Canal de commandes

- ↪ C'est le client qui ouvre la connexion. Celle-ci se ferme lors de la fin des transactions (quand le client se déconnecte ou à la fin d'un time out).
- ↪ Le port côté serveur est le port 21.
- ↪ Le port côté client est un port temporaire.
- ↪ Le client envoie des commandes et le serveur y répond.
- ↪ Par défaut, l'authentification est faite en clair.

32

Conclusion

Canal de données

- ↳ En mode passif, c'est le client qui ouvre les connexions.
- ↳ En mode actif, c'est le serveur qui ouvre les connexions.
- ↳ Il y a une ouverture de connexion par échange de données.
- ↳ Un échange de données peut être :
 - ✓ Un transfert du contenu d'un répertoire.
 - ✓ Un fichier texte.
 - ✓ Un fichier binaire.
- ↳ Le port FTPDATA coté serveur est paramétrable. Cela peut être un port non standard (différent de 20, supérieur à 1024 et temporaire). Cela complique le fonctionnement en cas de translation d'adresse.