

# TP 4 Couche Application - FTP

*TP Réseaux Avancés MISIC-IA*

Ilyas Bambrik

# Table des matières

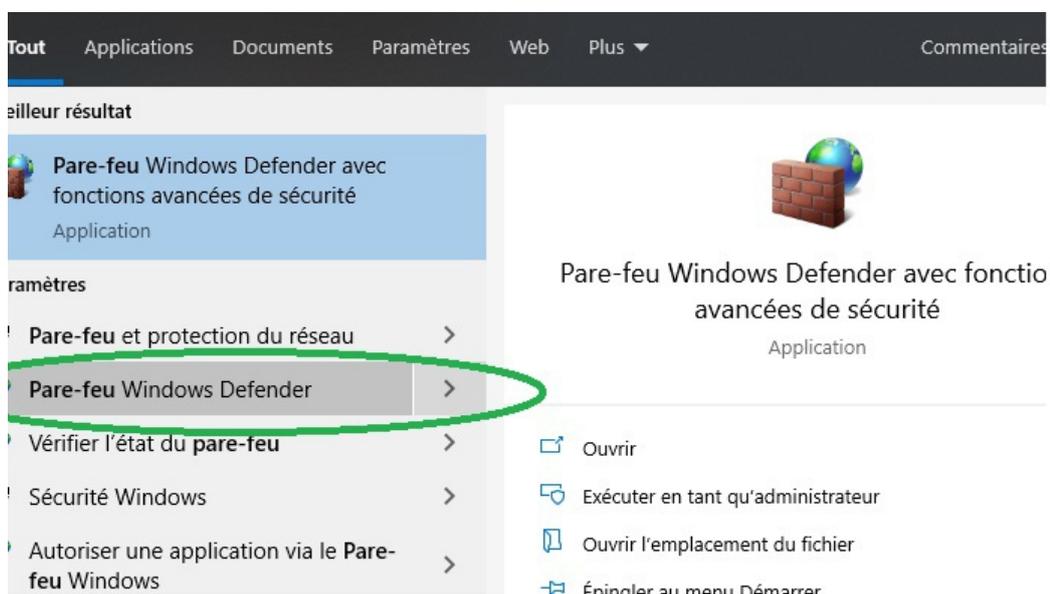


<b>I - Désactivation Pare-feu Windows</b>	<b>3</b>
<b>II - Exercice : Commandes FTP</b>	<b>5</b>
<b>III - Exercice : Implémentation client FTP et ftplib</b>	<b>6</b>
<b>Solutions des exercices</b>	<b>9</b>

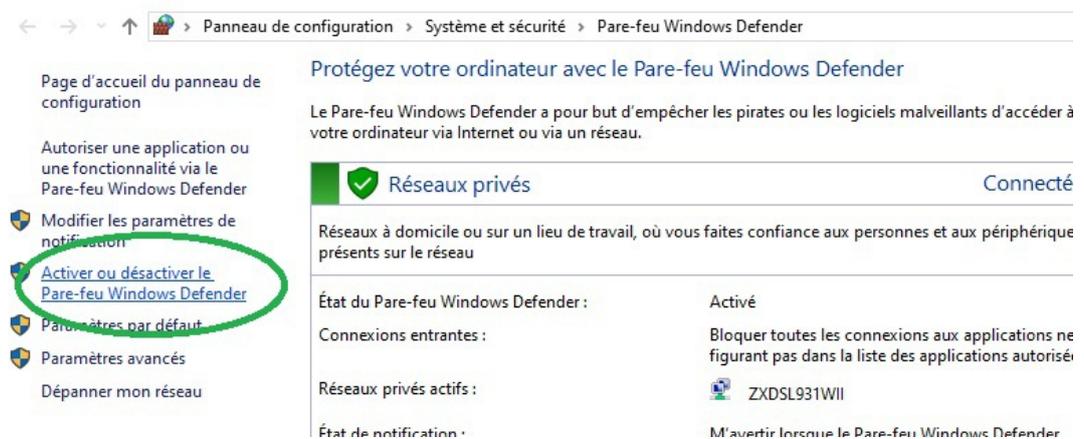
# Désactivation Pare-feu Windows

Avant de commencer à utiliser l'application client ftp de windows depuis le CMD, il faut désactiver le parefeu windows car l'implémentation windows fonctionne exclusivement avec le mode FTP actif.

Ouvrez "Pare-feu Windows Defender" depuis le menu demarage.



Cliquez sur activer ou désactiver le "Parefeu Windows Defender"



Désactiver le parefeu windows (après la fin des tests TP, cliquez sur "Activer le Parefeu Windows Defender")

### Personnaliser les paramètres pour chaque type de réseau

Vous pouvez modifier les paramètres de pare-feu pour chaque type de réseau que vous utilisez.

#### Paramètres des réseaux privés

- Activer le Pare-feu Windows Defender
  - Bloquer toutes les connexions entrantes, y compris celles de la liste des applications autorisées
  - M'avertir lorsque le Pare-feu Windows Defender bloque une nouvelle application
- Désactiver le Pare-feu Windows Defender (non recommandé)

#### Paramètres des réseaux publics

- Activer le Pare-feu Windows Defender
  - Bloquer toutes les connexions entrantes, y compris celles de la liste des applications autorisées
  - M'avertir lorsque le Pare-feu Windows Defender bloque une nouvelle application
- Désactiver le Pare-feu Windows Defender (non recommandé)

OK

Annuler

# Exercice : Commandes FTP



Ouvrez votre CMD et Wireshark. Capturez l'interface Wifi.

Connectez à un serveur FTP depuis votre invité de commande : 192.168.1.2 (serveur local dans la salle TP) ou bien *ftp.vim.org* sur internet. Tapez :

ftp 192.168.1.2

ou

ftp *ftp.vim.org*.

## Question 1

[solution n°1 p.9]

- Authentifiez vous avec un login *anonymous* (accès anonyme) et puis tapez une adresse mail avec un format valide (i@gmail.com par exemple) comme mot de passe.
- Filtrez les communications sur Wireshark pour avoir seulement les échanges avec le serveur FTP (*filtrez avec adresse IP de préférence par exemple : ip.addr==145.220.21.40 afin de filtrer la communication avec le serveur FTP ftp.vim.org*) ;
- Quelles sont les commandes qui précèdent le nom d'utilisateur et le mot passe respectivement?

## Question 2

[solution n°2 p.9]

- Tapez *LS* (comme dans *Linux pour lister le contenu du répertoire courant*)
  - Quelle est la commande FTP transmise par le client suite à l'exécution de *LS* ? (utilisez Wireshark pour la retrouver).
  - Quel est le port TCP utilisé par le serveur pour transmettre le résultat (le listing du répertoire) ? Quels sont les codes transmis par le serveur suite à l'exécution de *LS* ?
- Utilisez le filtre *tcp.port==21* pour repérer la connexion de contrôle et *tcp.port==20* pour la connexion de donnée (en mode actif).

## Question 3

[solution n°3 p.10]

- Naviguez dans le répertoire : pour les gens qui testent ces commandes sur le serveur *ftp.vim.org*, CD pub et puis CD ImageMagick. Téléchargez ImageMagick-6.9.11-29.7z.
- Repérez les commandes transmises depuis le client.
- Exécutez la commande *binary* sur le terminale. Quelle est la commande transmise qui correspond à celle ci.

# Exercice :

# Implémentation client

# FTP et ftplib

III

Lisez et testez le programme suivant (celui ci montre comment lire et écrire dans un fichier en python):

Pour tester le programme, donnez une chemin d'un à fichier source et un chemin de sortie valides. Par exemple *C:\Python27\README.txt*

```

1 # Le programme suivant permet de copier un fichier en mode binaire
2 # (exe, png, etc)
3
4 # Entrez le chemin de votre fichier a copier
5 Chemin_fichier_oreginal=input("Entrez le chemin de votre fichier :")
6 #Entrez le chemin de votre fichier :
7 Chemin_copie=input("Entrez l'emplacement de la copie :")
8 # Ouverture du fichier en mode lecture (r) en octets (b)
9 Fichier_en_lecture=open(Chemin_fichier_oreginal,"rb")
10 # utilisez un chemin de fichier de votre choix
11 Suite_doctets=[]
12 print( "#"*20)
13 for bloque_octets in Fichier_en_lecture:
14     Suite_doctets.append(bloque_octets)
15
16 # Ouverture du fichier en mode ecriture (w) en octets (b)
17 Fichier_en_ecriture=open(Chemin_copie,"wb")
18 for octets in Suite_doctets:
19     # Ecriture de la suite d'octets dans le fichier Image
20     Fichier_en_ecriture.write(octets)
21 # Fermeture du fichier
22 Fichier_en_ecriture.close()
23

```

*Listing 1* - Programme pour copier un fichier binaire

Le programme précédant peut être abrégé comme suite.

```

1 Oreginal=open(input("Entrez le chemin de votre fichier :"),'rb')
2 Copie=open(input("Entrez l'emplacement de la copie :") , "wb")
3 Copie.write(Oreginal.read())
4 Copie.close()

```

Le programme python suivant permet de se connecter à un serveur FTP (port 21) et d'exécuter des commandes d'accès seulement (USER,PASS, CWD, QUIT).

- Que doit être ajouté à ce programme pour permettre l'échange de données ?

```

1 import socket

```

```

2 import re
3 ConnexionAUnServeur = socket.socket()
4
5 # Entrez l'adresse IP du serveur FTP depuis le clavier
6 print "Entrez l'adresse IP du serveur FTP depuis le clavier : "
7 host = raw_input()
8 port = 21 # numero de port
9 ConnexionAUnServeur.connect((host, port)) # etablisement de connexion avec le
10
11 while True:
12     recu=""
13     while True:
14         recu+=ConnexionAUnServeur.recv(2028)
15         m=re.findall("\d\d\d ",recu)
16         if len(m)!=0:
17             break
18     print recu
19     commande=raw_input()
20     ConnexionAUnServeur.send(commande+"\r\n")
21
22 ConnexionAUnServeur.close()

```

Listing 2 - Connexion à un fichier binaire

Le programme suivant utilise la librairie standard python ftplib qui permet d'exécuter n'importe quelle commande FTP. Exécutez le programme et lisez le résultat.

```

1 import ftplib
2 # ouverture d'une connexion au serveur FTP
3 connexionftp=ftplib.FTP("ftp.vim.org")
4 print '\n##### Fin instruction [connexionftp=ftplib.FTP("ftp.vim.org")]\n'
5 # commande login anonyme
6 connexionftp.login()
7
8 print '\n##### Fin instruction [connexionftp.login()]\n'
9 # commande NLST
10 print '\n'.join(connexionftp.nlst())
11
12 print '\n##### Fin instruction [connexionftp.nlst()]\n'
13
14 print connexionftp.cwd("pub")
15 print connexionftp.cwd("ImageMagick")
16 print '\n##### Fin instruction CD pub\ImageMagick'
17 # telechargement du fichier welcome.msg
18 connexionftp.retrbinary('RETR ImageMagick-6.9.11-29.7z', open('ImageMagick-
19     6.9.11-29.7z', 'wb').write)
20 print '\n##### Fin instruction [ftp.retrbinary("RETR welcome.msg", open
21     ("welcome.msg", "wb").write)]\n'

```

Listing 3 - Ftplib

Le dernier programme montre comment exécuter des commandes système à partir d'un programme.

```

1 import os # bibliotheque systeme
2 import time
3 # os.mkdir cree un repertoire courant
4 # cette boucle cree les repertoires "0" jusqu'a "5"
5 for i in range(5):
6

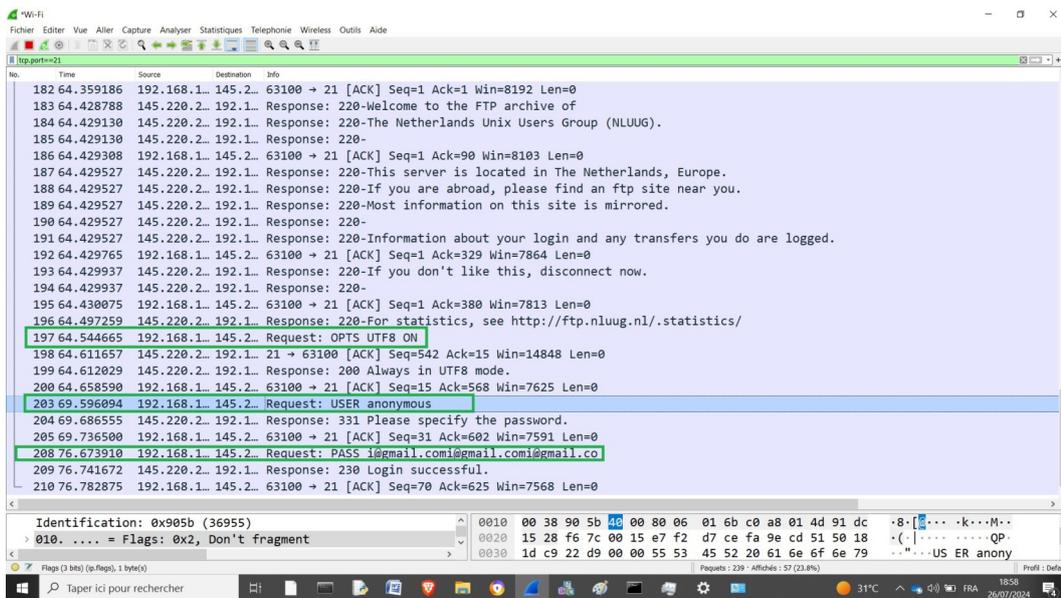
```

```
7         os.mkdir(str(i))
8 # met le programme en pause pour 10sec
9 time.sleep(10)
10 # rmdir supprime un repertoire
11 # cette boucle supprime les repertoires "0" jusqu'a "5"
12 for i in range(5):
13     os.rmdir(str(i))
14
15 # os.listdir imprime les noms des repertoires/fichiers dans
16 # le repertoire courant
17 print os.listdir(".")
18 # os.getcwd() est equivalente a pwd (imprime le chemin actuel)
19 print os.getcwd()
20 # CD .. (retourne au repertoire parant)
21 os.chdir("../")
22 print "#####",os.getcwd(),os.listdir(".")
```

# Solutions des exercices

> Solution n°1

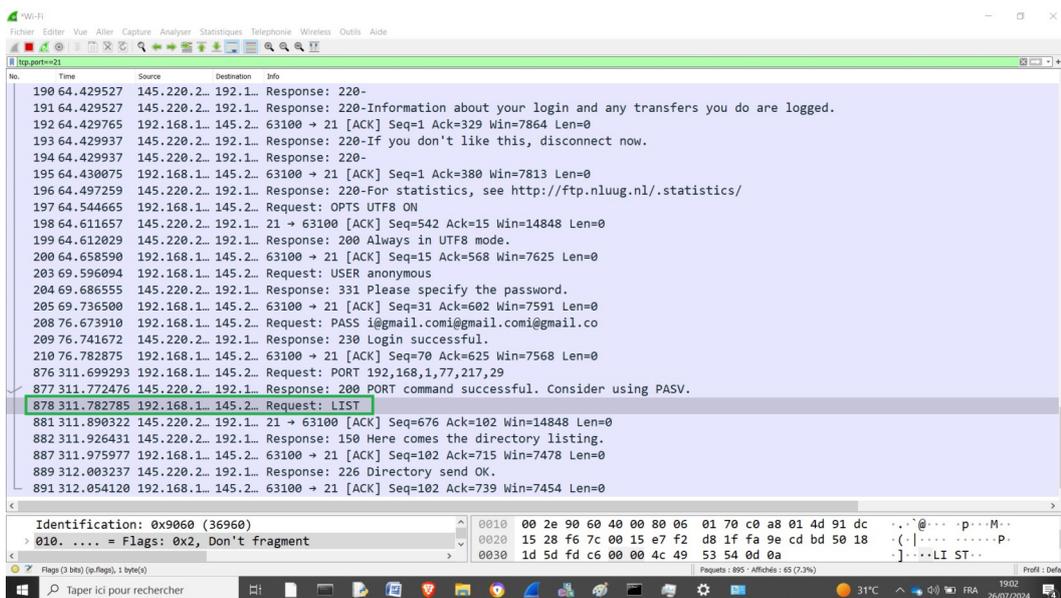
Exercice p. 5



USER et PASS sont transmis avec le login et mot de passe

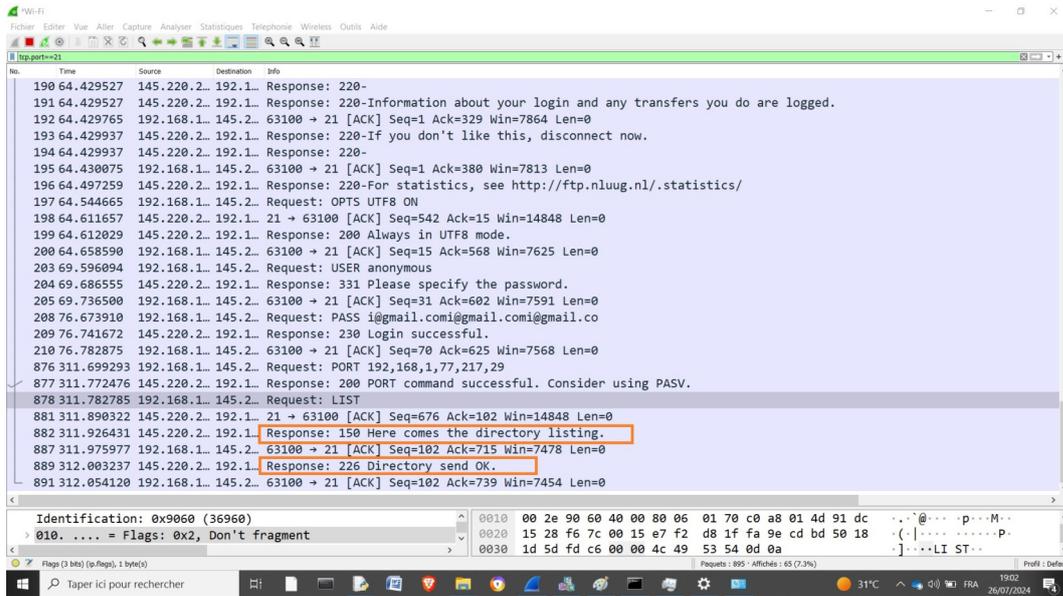
> Solution n°2

Exercice p. 5

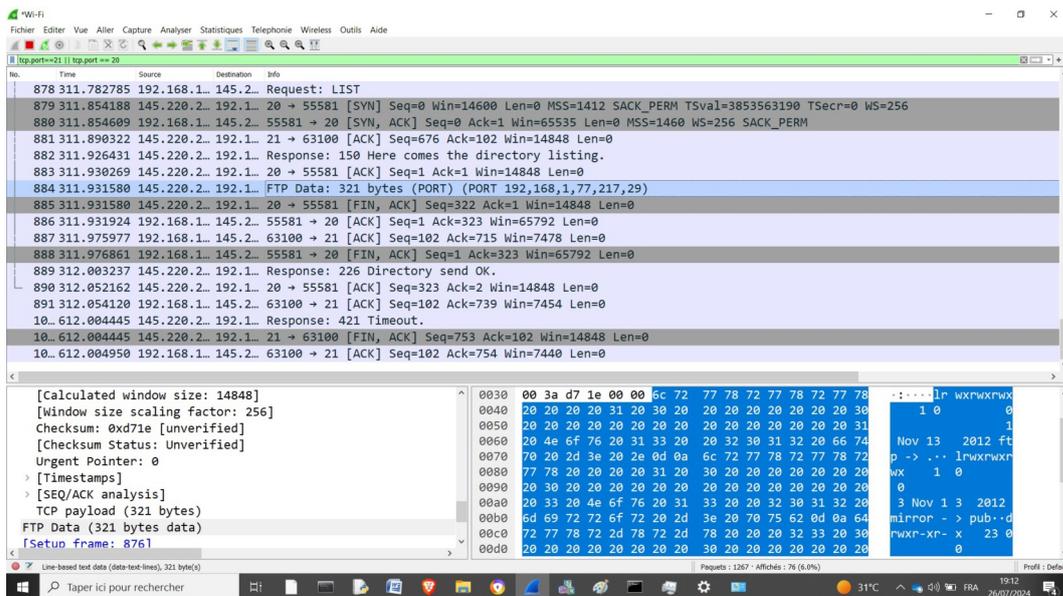


LIST == LS.

Le port numéro 20 est utilisé pour transmettre le listing.



Suite à l'envoi de la commande LIST, les codes renvoyés par le serveur sont 150 et 226.



> Solution n°3

The screenshot shows a Wireshark capture of a network session. The packet list pane shows several packets, with packet 70 highlighted. The packet details pane shows the structure of the selected packet, including Ethernet II, Internet Protocol Version 4, and Hypertext Transfer Protocol. The packet bytes pane shows the raw data in hexadecimal and ASCII. The session appears to be a directory listing followed by a file transfer.

The screenshot shows a Wireshark capture of a network session, focusing on a file transfer. The packet list pane shows several packets, with packet 8958 highlighted. The packet details pane shows the structure of the selected packet, including Ethernet II, Internet Protocol Version 4, and Hypertext Transfer Protocol. The packet bytes pane shows the raw data in hexadecimal and ASCII. The session appears to be a file transfer.