

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



الحمد لله الذي خلق السموات والأرض وجعل الظلمات والنور  
ثم الدين كفروا بربهم يعدلون. العالم بما كان وما هو كائن وما  
سيكون ،الذي إنما أمره إذا أراد شيئاً أن يقول له: كن فيكون،  
الذى يخلق ما يشاء ويختار ما كان لهم الخيرة، سبحانه وتعالى عما  
يشركون، وهو الله لا إله إلا هو، له الحمد في الأولى والآخرة وله  
الحكم وإليه ترجعون،

-----  
- أما بعد فقد وفقنا الله تعالى انا وجميع أعضاء  
منتدى الأميرة للكمبيوتر في إنشاء هذا الكتيب  
الصغير الذي يحوي على :

**بروتوكول الاف تي بي**

ونرجوا أن يعجبكم الكتيب أخوكم المبرمج :  
**أمين دبابسية**

[aminedebabsia@gmail.com](mailto:aminedebabsia@gmail.com)

الطرفين يتحدثون اللغة العربية فسيسهل على كل طرف فهم طلب الآخر. كذلك هو الحال في أنظمة الكمبيوتر.

---

### • مقدمة عن بروتوكول نقل الملفات (ftp).

بروتوكول (ftp) وهو اختصاراً ل (file transfer protocol) أي بروتوكول نقل الملفات. وهذا البروتوكول عبارة عن تطبيق يعمل في الطبقة الرابعة من طبقات (tcp/ip) وهذه الطبقة تدعى بطبقة التطبيقات (application layer). ويقوم هذا البروتوكول عادة بنقل نسخة من الملفات إلى الجهاز الهدف مع ترك نسخة منها في الجهاز المصدر. وقد صمم هذا البروتوكول لكي يعمل بين أجهزة مختلفة في النوع. كذلك ليعمل بين جهازين يعملان بنظامين تشغيليين مختلفين مثلاً يمكن لجهاز يستخدم نظام الويندوز (windows)، وجهاز آخر يعمل بنظام (Linux) أن يتم الربط بينهما دون التأثير على عمل البروتوكول. وبروتوكول نقل الملفات يستطيع أن ينقل فقط أنواع محدودة من الملفات (binary, ASCII). ولكي تستخدم هذا البروتوكول لابد أن يكون لك حساب على الخادم (server) وهذا الحساب يكون باسم مستخدم خاص بك وكذلك كلمة مرور . ويوجد أيضاً هناك بعض الخوادم التي تقدم هذه الخدمة بوضعية التخفي (anonymous) وسيتم مناقشتها لاحقاً في الأمثلة. ويتميز بروتوكول نقل الملفات عن باقي البروتوكولات الأخرى في أنه وعند قيامه بعملية نقل الملفات ينشئ إتصالين وهما:

#### ١. إتصال التحكم (control connection):

وهذا الإتصال خاص بنقل الأوامر بين جهازي الخادم (server) والعميل (client). وهذا الإتصال يأخذ المنفذ (21) كإفتراضي له حيث أن الخادم يفتح هذا البورت وينتظر أن يجري جهاز العميل الإتصال من خلاله أي أن البورت (21) يكون في وضع (listen) أي إنصات لأي إتصال قادم. وهذا النوع من الإتصال يكون دائماً أي لا ينقطع إلا إذا قام أحد الطرفين بقطع الإتصال عن الطرف الآخر بخلاف النوع الثاني من الإتصالات.

#### ٢. إتصال البيانات (data connection):

وهذا الإتصال ينشأ عندما يتم نقل الملفات من العميل إلى الخادم أو العكس. أي أن هذا الإتصال يعتبر مؤقتاً فقط إلى أن يتم إنهاء المهمة الذي يقوم بها وهي عملية النقل للملفات. وسوف نتحدث عن هذين النوعين من الإتصالات في الفقرات القادمة من هذا الملف.

---

### • أوامر بروتوكول نقل الملفات (ftp):

في حقيقة الأمر أن الأوامر التي يقوم المستخدم بكتابتها عند القيام بعمليات نقل الملفات بين الخادم والعميل تختلف عن الأوامر التي يتم إرسالها من جهاز العميل إلى جهاز السيرفر مثلاً الأمر (cd) يستخدم للتنقل بين الأدلة على السيرفر فعندما تكتبه في موجه ال (ftp) فإنه يتحول إلى الأمر (CWD) أي الأمر الذي سوف يرسل من جهازك (العميل) إلى الجهاز الآخر (الخادم) هو الأمر (CWD). وسوف نوضح هنا بعض أهم الأوامر التي تستخدم بكثرة عند استخدام هذا البروتوكول.

#### • الأمر (lcd):

وهذا الأمر يستخدم للتنقل بين الأدلة على جهاز العميل (client) أي جهازك الذي تعمل عليه. لاحظ المثال التالي.

```
ftp> lcd dir1
Local directory now C:\dir1.
```

لاحظ في المثال السابق قد تم الانتقال إلى المجلد (dir1) على جهازك ولاحظ هنا أنه لن يتم إرسال الأمر للسيرفر لأن ذلك متعلق بجهازك أنت . وليس بالسيرفر . ويستخدم الأمر (lcd ..) للانتقال إلى الدليل الذي يعلو الدليل الذي تعمل عليه أنت حالياً.

**ملاحظة: في جميع الأوامر التالية سوف يتم مناقشتها بعد قليل سوف ترسل للسيرفر وذلك لتعلقها به لذلك سوف يكون الأمر الذي تكتبه أنت هو الذي يلي الموجه (ftp>) أما الذي يلي العلامة (>---) فهو الأمر الفعلي الذي يرسل من جهازك للسيرفر.**

#### • الأمر (cd):

وهذا الأمر يستخدم للتنقل بين أدلة المستخدم على الجهاز الخادم (server) وهذا الأمر هو الأمر هو الذي نكتبه عند استخدامنا لهذا الأمر ولكن الأمر الذي يرسل من جهاز العميل (client) هو الأمر (CWD). لاحظ المثال التالي أنه عندما استخدمنا الأمر (cd) فإن الأمر الذي أرسل للجهاز الخادم هو (CWD).

```
ftp> cd downloads
---> CWD downloads
```

لاحظ أن السطر الأول هو الأمر الذي نكتبه نحن، أما السطر الثاني فهو الأمر الحقيقي الذي أرسل للجهاز الخادم (server) ولاحظ هنا أنه (CWD) كما ذكرنا . وسنتعرف في فقرة الأمثلة كيف نتعرف على الأوامر المرسل من أجهزتنا إلى أجهزة السيرفر.

#### • الأمر (cd ..):

وهو يستخدم للانتقال إلى الدليل الذي يعلو الدليل الحالي. واعتقد أن هذا الأمر معروف لدى مستخدمي الدوس أو اللينوكس. لاحظ المثال التالي:

```
ftp> cd ..
---> CWD ..
```

#### • الأمر (pwd):

ويستخدم هذا الأمر للعرض المسار المتواجد عليه حالياً مثل:

```
ftp> pwd
---> XPWD
```

#### • الأمر (ASCII):

وهذا الأمر لتحديد نوع الملف المرسل ويستخدم عادة النوع (ASCII) إذا كان الملف الذي تريد نقله هو ملف نصي. فلابد أن يكون نوع النقل هو (ASCII). لاحظ المثال التالي:

```
ftp> ascii
---> TYPE A
```

لاحظ أن الأمر المرسل من أجهزتنا إلى السيرفر هو الأمر (TYPE A).

#### • الأمر (binary):

وهذا الأمر أيضاً يستخدم لتحديد نوع الملف المرسل ويستخدم عادةً النوع (binary) مع الملفات المضغوطة والبرامج التطبيقية والصور. وعند استخدام هذا الأمر يرسل للسيرفر الأمر (TYPE I) لاحظ المثال التالي:

```
ftp> binary
---> TYPE I
```

#### • الأمر (dir):

ويستخدم هذا الأمر عادةً لسرد محتويات دليل ما على السيرفر وفي حقيقة الأمر أنه عندما نستخدم هذا الأمر فإنه يرسل إلى السيرفر أمرين وليس أمر واحد كما كتبنا نحن لاحظ المثال التالي:

```
ftp> dir
---> PORT 127,0,0,1,4,26
200 PORT command okay
---> LIST
150 File Listing Follows in ASCII mode.
total 31
drwxrwxrwx 1 noone  nogroup    0 Jul 31 16:26 bin
-rwxrwxrwx 1 noone  nogroup  5243 Mar 24 2001 dmlogo.gif
drwxrwxrwx 1 noone  nogroup    0 Jul 31 16:25 html
drwxrwxrwx 1 noone  nogroup    0 Jul 31 16:25 include
drwxrwxrwx 1 noone  nogroup    0 Jul 31 16:26 lib
-rwxrwxrwx 1 noone  nogroup  2483 Feb  6 02:01 license.txt
-rwxrwxrwx 1 noone  nogroup  2279 Mar 17 16:46 readme.html
-rwxrwxrwx 1 noone  nogroup  1544 Jan 29 2002 readme.txt
drwxrwxrwx 1 noone  nogroup    0 Jul 31 16:26 stl
-rwxrwxrwx 1 noone  nogroup   40 Aug 30 2001 v829
```

لاحظ أنه عندما استخدمنا الأمر (dir) تم إرسال الأمر (port) للسيرفر وذلك حتى يتم فتح إتصال لنقل البيانات حيث أن السرد يعرض دائماً على إتصال نقل البيانات. ثم بعد تحديد المنفذ الخاص بإتصال نقل البيانات يتم إرسال الأمر (list) إلى السيرفر وذلك لكي يعرض محتويات الدليل. ويوجد أيضاً هناك أمر يقوم

بنفس مهمة الأمر (dir) وهو الأمر (ls) ولك حرية الاختيار في إستعمال أي منهما . وأعتقد أنك إذا كنت من مستخدمي نظام التشغيل دوس فإنك سوف تستخدم الأمر (dir) أما إذا كنت من مستخدمي أنظمة اليونكس فإنه من المؤكد أنك سوف تختار الأمر (ls).

#### • الأمر (mkdir):

ويستخدم هذا الأمر لإنشاء مجلد جديد على الجهاز الخادم (server). وعند إستخدام هذا الأمر فإن جهاز العميل سوف يرسل الأمر (XMKD). لاحظ المثال التالي:

```
ftp> mkdir dir1
---> XMKD dir1
```

حيث هنا أن إسم المجلد هو (dir1).

#### • الأمر (rmdir):

ويستخدم هذا الأمر لحذف ملف من على السيرفر . وعند إستخدام هذا الأمر فإن جهاز العميل سيرسل الأمر (XRMD). لاحظ المثال التالي:

```
ftp> rmdir dir1
---> XRMD dir1
```

حيث أن إسم المجلد المراد حذفه هو (dir1).

#### • الأمر (rename):

ويستخدم هذا الأمر لإعادة تسمية الأدلة والملفات على السيرفر . وعند إستخدام هذا الأمر فإنه يرسل أمرين من السيرفر لاحظ المثال التالي:

```
ftp> rename dir dir1
---> RNFR dir
350 Enter the name to rename it to...
---> RNTD dir1
```

لاحظ أن (dir) هو اسم الدليل الحالي أما (dir1) فهو إسم الدليل الذي ستغير اسم الدليل إليه. وكما قلنا سابقاً أن هذا الأمر يرسل أمرين وهما (RNFR) وهي إختصار (rename from) أي أعد تسمية هذا الدليل من هذا الاسم إلى الذي سيرسله الأمر (RNTD) .

#### • الأمر (get):

وهذا الأمر يستخدم لأخذ ملف معين من على السيرفر . وعند إستخدام هذا الأمر سوف يرسل أمرين الأول هو الأمر (PORT) وذلك لتحديد البورت الذي سوف يفتح من خلاله إتصال نقل البيانات. أما الأمر الآخر فهم الأمر (RETR) وهو الأمر الذي يخبر السيرفر أن يجلب الملف المعين. لاحظ المثال التالي:

```
ftp> GET TEXT11.TXT
---> PORT 127,0,0,1,4,15
200 PORT command okay
```

```
---> RETR TEXT11.TXT
```

ويوجد هناك أمر آخر يقوم بنفس المهمة وهو الأمر (recv). أما إذا كنت تريد أن تنقل أكثر من ملف في وقت واحد فإنه يمكنك استخدام الأمر (mget) حيث أنه عند نقل كل ملف سوف يتم فتح إتصال بيانات جديد، وسوف نلاحظ أمثلة لهذا الأمر في فقرة الأمثلة على البروتوكول.

#### • الأمر (put):

ويستخدم هذا الأمر لوضع ملف معين على السيرفر . أي أنه يتم أخذ نسخة من الملف الموجود على جهازك ووضعها على السيرفر. وعند استخدام هذا الأمر سوف يرسل إلى السيرفر أمرين وهما (PORT) وذلك لفتح إتصال بيانات، أما الأمر الآخر فهو الأمر (STOR). لاحظ المثال التالي:

```
ftp> put text11.txt
---> PORT 127,0,0,1,4,18
200 PORT command okay
---> STOR text11.txt
```

وهناك أمر آخر يقوم بنفس المهمة وهو الأمر (send). أما إذا كنت تريد أن تضع أكثر من ملف على السيرفر في وقت واحد فإنه يمكنك من استخدام الأمر (mput) حيث أنه عند تحمي كل ملف سوف يفتح إتصال بيانات جديد، وسوف نرى أمثلة هذا الأمر في فقرة الأمثلة.

#### • الأمر (open):

وهو يستخدم لإنشاء إتصال جديد ويستخدم على الصورة التالية:

```
ftp> open server_name
```

حيث أن (server\_name) هو اسم السيرفر الذي تريد الإتصال به.

#### • الأمر (close):

وهو يستخدم لإنهاء إتصالك مع السيرفر ويستخدم على الصورة:

```
ftp> close
```

ويوجد أيضاً الأمر (disconnect) الذي يقوم بنفس المهمة.

#### • الأمر (bye):

ويستخدم هذا الأمر لإنهاء إتصالك مع السيرفر وإرجاعك لموجة الدوس وليس إلى موجة (ftp). وهذا الفرق بينه وبين الأمر (close). ويستخدم على الصورة التالية:

```
ftp> bye
---> QUIT
```

ويوجد هناك أيضاً بعض الأوامر التي لم يتم مناقشتها هنا ويمكنك معرفة هذه الأوامر من خلال استخدام الأمر (help) عند موجة ال(ftp). على الصورة التالية:

```
ftp> help
Commands may be abbreviated. Commands are:
```

!	delete	literal	prompt	send
?	debug	ls	put	status
append	dir	mdelete	pwd	trace
ascii	disconnect	mdir	quit	type
bell	get	mget	quote	user
binary	glob	mkdir	recv	verbose
bye	hash	mls	remotehelp	
cd	help	mput	rename	
close	lcd	open	rmdir	

ولرؤية وظيفة كل أمر إستخدم الأمر :

ftp>? Command

حيث أن (command) هو الأمر الذي تريد الإستعلام عنه.

### • فهم الأكواد الرقمية التي يصدرها بروتوكول نقل الملفات أثناء عملية إرسال وإستقبال الأوامر.

عندما يرسل المستخدم أمراً ما إلى السيرفر فإن السيرفر لابد أن يرد على المستخدم بما قد فعل السيرفر ويتكون هذا الرد عادةً من كود رقمي مكون من ثلاثة خانات حيث أن الخانة الأولى يكون أصغر رقم محتمل في هذه الخانة هو (1) أما أكبر عدد فهو (5). أما الخانة الثانية والثالثة فأقل عدد هو (0) وأكبر عدد هو (5). وكل خانة في هذا الكود الرقمي لها معنى والجدول التالي يبين معاني هذه الشفرات:

الكود	شرحها	مثال
1--	عندما يأتي هذا الرقم في البداية فإنه يرمز إلى أن السيرفر يقوم بعمل ما ولكن لا يستطيع أنت أن تدخل أمر آخر حتى ينتهي من هذا الأمر .	125 data connection already open; transfer string.
2--	عندما يأتي هذا الرقم في البداية فإنه يرمز إلى أن السيرفر قام بعمل ما وتستطيع أن ترسل أمراً آخر ليتم تنفيذه.	200 command ok.
3--	عندما يأتي هذا الرقم في البداية فإنه يرمز أن السيرفر قد قبل الأمر الذي أدخلته ولكنه يحتاج إلى أمر آخر حتى يتم إكماله وغالباً ما يرى مثل هذا الكود عندما يدخل المستخدم اسم المستخدم الخاص به فإن السيرفر لا يستطيع أن يكمل حتى يتم إرسال كلمة السر.	331 Give me your password, please
4--	عندما يأتي هذا الرقم في البداية فإنه يرمز	425 can`t open

data connection.	إلى أن السيرفر يقبل هذا الأمر ولكنك لا تستطيع تنفيذه حالياً. ولكن يمكنك تنفيذه في وقت لاحق.	
500 Unidentified command hello	عندما يأتي هذا الرقم في البداية فإنه يرمز إلى أن السيرفر لا يقبل بهذا الأمر وترى عادة هذا النوع من الرسائل عندما يكون الأمر المرسل لا يعمل على السيرفر أو أن هذا الأمر لا يعمل على السيرفر. كما ستلاحظ في المثال التالي عندما أرسلنا أمراً غير صحيح وهو (hello).	5--
500 Unidentified command hello	عندما يأتي هذا الرقم في المنتصف فإنه يرمز إلى أنه هناك خطأ في الأمر المرسل.	0-
214-The following commands are recognized (* =>'s unimplemented).	عندما يأتي هذا الرقم في المنتصف فإنه يرمز إلى أن هذه الرسالة هي عبارة عن رسالة للمساعدة.	1-
125 data connection already open; transfer string.	عندما يأتي هذا الرقم في المنتصف فإنه يرمز إلى أن هذا الرد متعلق بالاتصالات مثلاً إتصال البيانات أو إتصال التحكم.	2-
331 Give me your password, please	عندما يأتي هذا الرقم في المنتصف فإنه يرمز إلى تصاريح الدخول للسيرفر أو ملف ما. أي أن هذا الرقم مسؤول عن التصاريح.	3-
لا يوجد مثال وذلك لقلة وجود هذا الاحتمال.	عندما يأتي هذا الرقم في المنتصف فإنه يرمز إلى شيء غير محدد والصراحة لم تمر علي أحد هذه الحالات.	4-
150 "C:/text11.txt" file ready to receive in ASCII mode	عندما يأتي هذا الرقم في المنتصف فإنه يرمز إلى حالة ملفات النظام. وترى عادة هذه النوع من الرسائل في حالة نقل وتخزين الملفات.	5-

أما الخانة الثالثة من الرد فهي تعطي أكثر توضيحاً لمعنى الرد.



## • كيف يتم الإتصال بين جهاز الخادم (server) وجهاز العميل (client).

عندما يهيئ جهاز ما على أنه يعمل كخادم بروتوكول نقل الملفات (ftp server) فإنه يتم فتح منفذ في هذا الجهاز (server) حتى يتسنى للأجهزة الأخرى الإتصال به من خلال هذا المنفذ ويستمتع عادة سيرفر بروتوكول نقل الملفات على المنفذ (21). ويتم الإتصال بينه وبين جهاز العميل بطريقة أنه عندما يتم تشغيل سيرفر بروتوكول نقل الملفات فإنه يفتح البورت (21) وينتظر أي إتصال قادم إليه وعندما يقوم جهاز العميل بطلب الإتصال بالسيرفر فإنه يقوم بإتمام هذا الإتصال مبدئياً حيث أنه سيطلب منه اسم المستخدم وكلمة المرور الذي تصرح له بالدخول إلى السيرفر فإذا كانت صحيحة أتم الإتصال وتصبح حالة الإتصال هي (established) أي تم الإتصال وهكذا يتم عمل إتصال التحكم والذي كما قلنا أنه يستخدم لكي يكون حلقة وصل بين السيرفر والمستخدم . ويتم إنشاء أيضاً إتصال آخر عندما يتم نقل الملفات من و إلى السيرفر ويعرف هذا الإتصال بما يسمى بإتصال البيانات أي أنه خاص بنقل البيانات فقط وهذا الإتصال يكون مؤقت فقط لفترة الإرسال وينتهي مهمته بعد ذلك بعكس إتصال التحكم الذي يكون مستمراً أثناء إتصالك بالسيرفر. ويتم إنشاء هذا الإتصال بطريقة أنه عندما مثلاً يريد المستخدم أن يأخذ ملفاً ما من جهاز الخادم فإنه يرسل إلى السيرفر الأمر (PORT) والصيغة العامة لهذا الأمر هو (PORT n1,n2,n3,n4,n5,n6) حيث أن (n1,n2,n3,n4) هو عنوان الإنترنت لك (الايبي الخاص بك) أما (n5,n6) هو الطريقة التي سوف يتم من خلالها معرفة رقم البورت الذي سيفتحه هذا الإتصال لاحظ المثال التالي:

---> PORT 127,0,0,1,4,15

لاحظ أن الايبي الخاص بي هو (127.0.0.1) وذلك لأنني أضع سيرفر (ftp) على جهازي الخاص. أما (4,15) فهي خاصة برقم البورت الذي سوف يفتح في جهازك ويتم حساب ذلك على الصورة:

$4 * 256 + 15$

لاحظ أن ناتج العملية السابقة هو (1039) إذاً سوف يتم فتح البورت (1039) في جهازك وذلك إستعداداً لنقل الملفات. ولاحظ أنه سوف يتم فتح إتصال بيانات جديد كلما أردت أن تنقل ملف آخر غير الإتصال السابق وذلك لأنه عندما يتم الإنتهاء من عملية نقل الملفات فإن السيرفر ينهي هذا الإتصال من خلال البورت رقم (20) وعادةً ينهي السيرفر إتصال البيانات من خلال البورت (20).

## • بعض الأمثلة على بروتوكول نقل الملفات (ftp).

مثال ١: إستعراض محتويات دليل ما.

لاحظ المثال الأول سوف نستعرض فقط محتويات الدليل الرئيسي لمستخدم ويمكنك تطبيق هذا الأمر على جميع أدلة السيرفر. لاحظ في المثال التالي أننا قمنا

باستخدام هذا البروتوكول من خلال بيئة الويندوز (دوس) ويمكن تطبيق نفس المهام على بيئة أنظمة ال (\*nix) أيضا . لاحظ أننا بعد أن قمنا بتشغيل غلاف الدوس كتبنا الأمر (C:\>ftp -d 62.114.95.161) ولاحظ أننا أضفنا الخيار (d) لهذا الأمر حتى يتسنى لنا رؤية الأوامر الحقيقية التي ترسل لسيرفر وهذه الأوامر تتمثل في كل مايلي العلامة (>---) . بعد ذلك إنتظرنا أن يتم الإتصال بين السيرفر والعميل (جهازك).

```
C:\>ftp -d 62.114.95.161
Connected to 62.114.95.161.
220 Serv-U FTP Server v4.0 for WinSock ready...
User (62.114.95.161:(none)): tt
---> USER tt
331 User name okay, need password.
Password:
---> PASS tt
230 User logged in, proceed.
ftp> dir
---> PORT 212,93,164,155,6,65
200 PORT Command successful.
---> LIST
150 Opening ASCII mode data connection for /bin/ls.
drw-rw-rw-  1 user  group      0 Jan  1 06:31 .
drw-rw-rw-  1 user  group      0 Jan  1 06:31 ..
-rw-rw-rw-  1 user  group    28827 Oct  1 1994 BOOKMARK.HTM
drw-rw-rw-  1 user  group      0 Jan 17 20:17 Fav
drw-rw-rw-  1 user  group      0 Jan  1 06:31 Hacking
-r--r--r--  1 user  group   38912 Oct 17 1994 Scrap.shs
226 Transfer complete.
ftp: 376 bytes received in 0.12Seconds 3.11Kbytes/sec.
ftp> bye
---> QUIT
221 Goodbye!
```

لاحظ أنه وكما قلنا أن السيرفر لابد أن يسألك عن اسم المستخدم وكلمة المرور وذلك حتى يرى إن كنت ممن يصرح لهم إستخدام هذا السيرفر ولاحظ أن اسم المستخدم هنا هو (tt) وكلمة المرور هي (tt) كما هو واضح في الجملة (>--- PASS tt) أي أنه يرسل كلمة المرور بدون أي تشفير. ثم بعد ذلك إستخدمنا الأمر (dir) لعرض محتويات الدليل ولاحظ أنه وبعد أن استخدمنا الأمر (dir) تم إرسال الأمر (>--- PORT 212,93,164,155,6,65) وذلك حتى يتم فتح إتصال بيانات جديد ولاحظ أن الايبي الخاص بي هو (212.93.164.155) أما البورت الذي سوف يتم فتحه على جهازي هو (65+256\*6) أي سيتم فتح البورت رقم (1601) لإتصال البيانات. وقد إستخدمت الأمر (a- netstat) حتى أرى الإتصالات التي تم مع السيرفر فوجدت الناتج التالي:

TCP	unix:ftp-data	212.93.164.155:1601	TIME_WAIT
TCP	unix:ftp	212.93.164.155:1597	ESTABLISHED

كان هذا الناتج من الجهاز الذي يعمل عليه السيرفر لاحظ أنه كما قلنا أن السيرفر دائماً ينهي إتصال البيانات من خلال البورت رقم (20) أو كما يدعى ب (ftp-data). وتعني كلمة (TIME\_WAIT) أي أنه قد تم الإنتهاء من هذا المنفذ وسيتم إغلاقه. ولو دققنا جيداً لوجدت أنه قد تم إنهاء إتصال البيانات ولكن لم يتم إنهاء إتصال التحكم الذي هو في وضع (established) أي تام . وهكذا تحققنا من صحة أن إتصال التحكم تام مالم يقطع من السيرفر أما إتصال البيانات فينتهي كلما انتهى من نقل ملف ما.

مثال ٢: أخذ ملف نصي من على السيرفر:

في هذا المثال سوف نتعلم كيف نأخذ ملف واحد فقط من على السيرفر. لاحظ ما يلي:

```
C:\>ftp -d 62.114.95.161
Connected to 62.114.95.161.
220 Serv-U FTP Server v4.0 for WinSock ready...
User (62.114.95.161:(none)): uu
---> USER uu
331 User name okay, need password.
Password:
---> PASS uu
230 User logged in, proceed.
ftp> dir
---> PORT 212,93,164,155,6,140
200 PORT Command successful.
---> LIST
150 Opening ASCII mode data connection for /bin/ls.
drw-rw-rw-  1 user  group      0 Feb 16 06:54 .
drw-rw-rw-  1 user  group      0 Feb 16 06:54 ..
-rw-rw-rw-  1 user  group    45856 Jul 25  2001 DOS.TXT
-rw-rw-rw-  1 user  group    8540 Jul 23  2001 Ports.txt
-r--r--r--  1 user  group    4357 Mar 14  2001 happy99.txt
-r--r--r--  1 user  group    1630 Mar 14  2001 mcafee.txt
226 Transfer complete.
ftp: 382 bytes received in 0.16Seconds 2.39Kbytes/sec.
ftp> ascii
---> TYPE A
200 Type set to A.
ftp> get mcafee.txt
---> PORT 212,93,164,155,6,141
200 PORT Command successful.
---> RETR mcafee.txt
150 Opening ASCII mode data connection for mcafee.txt (1630 bytes).
226 Transfer complete.
ftp: 1630 bytes received in 4.31Seconds 0.38Kbytes/sec.
```

```
ftp> bye
---> QUIT
221 Goodbye!
```

في هذا المثال سوف يفتح البورت (141+256\*6) لإتصال البيانات. ولاحظ أنه قبل أن نستخدم أمر أخذ الملفات (get) الأمر (ascii) وذلك لأننا سوف نأخذ ملف نصي من على السيرفر وعادة عندما يتم تحميل ملفات نصية من على السيرفر أو وضع ملفات نصية على السيرفر فإننا نستخدم نظام النقل (ascii) أما إذا كان الهدف المراد أخذه من السيرفر أو وضعه عليه غير نصي مثلاً صور أو ملفات تشغيلية فإنه يستخدم وسيلة النقل (binary) كما سنرى في المثال القادم. ويجب هنا أن ننبه أنه إذا كنت تريد أن تأخذ أكثر من ملف من على السيرفر في وقت واحد فإنه يتم استخدام الأمر (ftp>mget filename1 filename2 filename3) ثم بعد ذلك سيتم فتح إتصال ببيانات جديد لكل ملف فمثلاً سيفتح إتصال ببيانات للملف (filename1) وكذلك إتصال ببيانات آخر للملف (filename2) وهكذا.

مثال ٣: وضع ملف تشغيلي على السيرفر.  
في هذا المثال سوف نرى كيف يتم نقل الملفات التشغيلية باستخدام نظام النقل (binary). لاحظ ما يلي.

```
C:\>ftp -d 163.121.153.40
Connected to 163.121.153.40.
220 Serv-U FTP Server v4.0 for WinSock ready...
User (163.121.153.40:(none)): uu
---> USER uu
331 User name okay, need password.
Password:
---> PASS uu
230 User logged in, proceed.
ftp> binary
---> TYPE I
200 Type set to I.
ftp> put crypt.exe
---> PORT 212,93,169,233,7,178
200 PORT Command successful.
---> STOR crypt.exe
150 Opening BINARY mode data connection for crypt.exe.
226 Transfer complete.
ftp: 40960 bytes sent in 0.01Seconds 4096.00Kbytes/sec.
ftp> bye
---> QUIT
221 Goodbye!
```

لاحظ في المثال السابق أنه قد استخدم الأمر (binary) وذلك لأننا نريد أن ننقل ملفي تطبيقي وبعد ذلك قمنا باستخدام الأمر (PUT) والذي يستخدم لتخزين ملف

على السيرفر . ولاحظ أننا لو أردنا أن نحمل إلى السيرفر عدة ملفات فإننا نستخدم الأمر (ftp>mput filename1 filename2 filename3) وعند كل ملف يتم نقله إلى السيرفر فإنه يتم فتح إتصال بيانات جديد .

مثال ٤: تسجيل دخول للجهاز الخادم بوضع المجهول (anonymous):  
كما ذكرنا سابقاً أنه عندما تريد استخدام بروتوكول نقل الملفات مع أحد الأجهزة الخادمة فإنه لابد أن يكون لك اسم مستخدم وكلمة مرور أيضاً ولكن يمكنك أيضاً أن تقوم بتسجيل الدخول بوضع التخفي وذلك إذا كان الجهاز الخادم (server) يدعم هذا الوضع لاحظ الجلسة التالية التي أجريتها على أحد أنواع السيرفرات الموجودة على جهازي ولذلك سوف تلاحظ أن قد تم استخدام رقم الايبي (127.0.0.1) ونظراً لأن الطريقة واحدة لتسجيل الدخول فإنه سينطبق على أي سيرفر آخر .

```
C:\>ftp -d 127.0.0.1
Connected to 127.0.0.1.
220-This FTP site is running a copy of WFTPD that is NOT REGISTERED
220-
220-Shareware can only improve if supported by its users.
220-The easiest way to support shareware is to register it.
220-WFTPD costs from $20 to register.
220-
220-To register this program, or receive new details on it, send email
220-to alun@texas.com (Alun Jones), or snail-mail to Texas Imperial Software,
220-1602 Harvest Moon Place, Cedar Park TX 78613-1419 USA
220-
220-As added incentive for the site owner to register, you will be restricted
220-to five (5) transfers - to get more transfers, please re-login.
220-
220-Please note - Alun Jones is only responsible for the software
220-that this site runs, and is not responsible in any way for either
220-the content of this site, nor its location on the Internet.
220 WFTPD 2.4 service (by Texas Imperial Software) ready for new user
User (127.0.0.1:(none)): anonymous
---> USER anonymous
331-Anonymous user access allowed - please enter your email
331-address as the password:
331 Give me your password, please
Password:
---> PASS anonymouse@anonymoys
230 Logged in successfully
ftp> dir
---> PORT 127,0,0,1,5,193
200 PORT command okay
---> LIST
150 File Listing Follows in ASCII mode.
```

وهنا قد تم بمشية الله وعونه إنهاء هذا  
الكتيب الصغير واعتذر لتقصيري  
وتقبلوا خالص تحياتي أخوكم  
المبرمج:

أمين دبابسية

ونشكر كل من ساهم في إنشاء هذا  
الكتيب وخاصة العضو مروان  
دبابسية-- حمام النبائل/قائمة

معدة من طرف المبرمج  
أمين دبابسية

*from programmer  
amine debabsia*