

BACKGROUND

Il protocollo FTP (File Transfer Protocol) prevede due modalità di funzionamento:

1.Active Mode

Il client FTP contatta da una propria porta tcp superiore alla 1023 (ephemeral ports) il server FTP sulla porta tcp 21 di quest'ultimo. Attraverso questa socket, passano i messaggi di controllo del protocollo (es. richiesta credenziali autenticazione, comandi di GET e di PUT, messaggi di stato, etc).

Il trasferimento dei file implica invece la porta tcp 20 del server ed una porta tcp superiore alla 1023 del client scelta arbitrariamente dal server stesso. Ovviamente, perché questo funzioni, non è sufficiente che il server ed il client siano impostati in modalità "active mode". Infatti, negli scenari odierni, caratterizzati da un uso sempre più diffuso dei Personal Firewall installati sui client e dai router NAT usati per l'accesso alla banda larga in ambiente SOHO, è fondamentale che il Firewall / NAT sia impostato per lasciare passare il traffico verso la porta del client che il server utilizzerà per lo scambio dati. Come si può ben comprendere, ciò richiederebbe l'apertura di varie porte sul Firewall / NAT che protegge il client ed è normalmente sconsigliato per evidenti motivi di sicurezza oltre che per la difficoltà della configurazione che ricadrebbe sull'utente del servizio.

2.Passive Mode

L'alternativa alla problematica generata dalle caratteristiche della "active mode", è quella di ribaltare la posizione dei soggetti coinvolti nel transito dati e prevedere che sia il Firewall / NAT del server ad accettare preventivamente un range di porte tra le quali il server stesso sceglierà una particolare porta che poi comunicherà al client per essere contattato. Questa modalità prende il nome di "passive mode".

Il traffico legato alla porta tcp 21 rimane uguale a quello del caso descritto sopra (modalità "active mode").

Per il trasferimento dei file, invece, il client invia al server un messaggio di proposta per il "passive mode" (il comando è "PASV") ed il server FTP risponde al client FTP con un messaggio di controllo del tipo:

```
227 Enter passive mode (Indirizzo IP , IDPorta)
```

```
227 Enter passive mode (195,168,0,254,195,80)
```

Nel nostro esempio, il messaggio indica al client di aprire una socket per il trasferimento dei file, verso l'indirizzo IP 195.168.0.254 (il server ftp o l'interfaccia esterna del Firewall / NAT) e verso la porta 50.000 ($195 \cdot 256 + 80 = 49.920 + 80 = 50.000$).

In questo caso, il client o il suo Firewall / NAT non devono essere impostati in alcun modo per l'apertura di una qualche porta se non per il normale traffico in uscita (solitamente concesso dalle regole di default di qualunque firewall). Al contrario, dovrà essere il Firewall / NAT del server a prevedere regole per l'inoltro delle porte scelte dal server per venire contattato in "passive mode". Solitamente, piuttosto che lasciare aperte tutte le porte superiori alla 1023, si preferisce restringere la superficie d'attacco, scegliendo un range di porte (es. 50.000 -50.100) che naturalmente occorrerà impostare anche sul server FTP in modo che esso scelga solo da questo sottoinsieme la porta da comunicare al client col messaggio 227 di cui sopra.

N.B.

Ai fini della comprensione dell'articolo, si richiede la conoscenza del funzionamento del NAT (Network Address Translation) / PAT (Port Address Translation), dei Firewall, del TCP/IP e del SSL/TLS.

IL PROBLEMA

Come noto, il protocollo FTP prevede il passaggio in chiaro sia dei dati che dei messaggi di controllo tanto in “active mode” che in “passive mode”.

Rilevando il traffico tra client e server FTP con un analizzatore di protocollo, otteniamo:

Src MAC Addr	Dst MAC Addr	P...	Description	Src Other Addr	Dst Other Addr
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, '220-FileZilla Server version 0.9.24 beta'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, '220-written by Tim Rosse (Tim.Rosse@gmx.de)'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, '220 Please visit http://sourceforge.net/projects/f'	195.168.0.254	WIN-XP-01
LOCAL	U.S. RE9821F	FTP	Req. from Port 2115, 'USER mac'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, '331 Password required for mac'	195.168.0.254	WIN-XP-01
LOCAL	U.S. RE9821F	FTP	Req. from Port 2115, 'PASS qwerty0987654321'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, '230 Logged on'	195.168.0.254	WIN-XP-01
LOCAL	U.S. RE9821F	FTP	Req. from Port 2115, 'SYST'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, '215 UNIX emulated by FileZilla'	195.168.0.254	WIN-XP-01
LOCAL	U.S. RE9821F	FTP	Req. from Port 2115, 'FEAT'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, '211-Features:'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, 'MDTM'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, 'REST STREAM'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, 'SIZE'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, 'MLST type*;size*;modify*;'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, 'MLSD'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, 'AUTH SSL'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, 'AUTH TLS'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, 'UTF8'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, 'CLNT'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, 'MFMT'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, '211 End'	195.168.0.254	WIN-XP-01
LOCAL	U.S. RE9821F	FTP	Req. from Port 2115, 'PWD'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, '257 "/" is current directory.'	195.168.0.254	WIN-XP-01
LOCAL	U.S. RE9821F	FTP	Req. from Port 2115, 'TYPE A'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, '200 Type set to A'	195.168.0.254	WIN-XP-01
LOCAL	U.S. RE9821F	FTP	Req. from Port 2115, 'PASV'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, '227 Entering Passive Mode (195,168,0,254,234,102)'	195.168.0.254	WIN-XP-01
LOCAL	U.S. RE9821F	FTP	Req. from Port 2115, 'LIST'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, '150 Connection accepted'	195.168.0.254	WIN-XP-01
U.S. RE9821F	LOCAL	FTP	Resp. to Port 2115, '226 Transfer OK'	195.168.0.254	WIN-XP-01

Si vedono passare perfettamente in chiaro le credenziali di accesso

user: mac

pass:qwerty0987654321

(si consiglia di aumentare il fattore di zoom sulla pagina)

Ovviamente, se continuiamo a monitorare il traffico ed effettuiamo il download di un file da parte del client, otterremo:

No.	Time	Source	Destination	Protocol	Info
1	0.000000	195.168.0.100	195.168.0.254	FTP	Request: RETR prova.txt
2	0.008298	195.168.0.100	195.168.0.254	TCP	2159 > 60029 [SYN] Seq=0 Len=0 MSS=1460
3	0.009072	195.168.0.254	195.168.0.100	TCP	60029 > 2159 [SYN, ACK] Seq=0 Ack=1 win=16384 Len=0 MSS=1460
4	0.009209	195.168.0.100	195.168.0.254	TCP	2159 > 60029 [ACK] Seq=1 Ack=1 win=65535 Len=0
5	0.013307	195.168.0.254	195.168.0.100	FTP	Response: 150 Connection accepted
6	0.014125	195.168.0.254	195.168.0.100	TCP	60029 > 2159 [PSH, ACK] Seq=1 Ack=1 win=17520 Len=18
7	0.014269	195.168.0.254	195.168.0.100	TCP	60029 > 2159 [FIN, ACK] Seq=19 Ack=1 win=17520 Len=0
8	0.014381	195.168.0.100	195.168.0.254	TCP	2159 > 60029 [ACK] Seq=1 Ack=20 win=65517 Len=0
9	0.019755	195.168.0.254	195.168.0.100	FTP	Response: 226 Transfer OK
10	0.019898	195.168.0.100	195.168.0.254	TCP	2158 > ftp [ACK] Seq=16 Ack=42 win=65036 Len=0
11	0.059981	195.168.0.100	195.168.0.254	TCP	2159 > 60029 [FIN, ACK] Seq=1 Ack=20 win=65517 Len=0
12	0.060614	195.168.0.254	195.168.0.100	TCP	60029 > 2159 [ACK] Seq=20 Ack=2 win=17520 Len=0

```

0000  00 10 a7 21 f2 e4 00 c0 49 e9 82 1f 08 00 45 00  ...!... I.....E.
0010  00 3a eb ac 40 00 7f 06 87 5e c3 a8 00 fe c3 a8  ...@... A.....
0020  00 64 ea 7d 08 0f 3e 80 c5 33 f8 cc 02 4d 50 18  .d.}.o>. .3...MP.
0030  44 70 ec 92 00 00 71 75 65 73 74 61 20 e8 20 75  Dp...qu esta . u
0040  6e 61 20 70 72 6f 76 61                          na prova
  
```

Anche qui, è visibile il nome del file (“prova.txt”) e perfino il suo contenuto (“questa è una prova”). Appare a questo punto evidente come la sicurezza non sia la caratteristica principale del protocollo FTP, ma fortunatamente esistono vari approcci a questo problema.

Brevemente, si possono elencare tre possibili scenari:

1. SFTP (FTP Over SSH) – SSH è un potente strumento per comunicare in modo sicuro ben conosciuto in ambiente Unix/Linux come una valida e sicura alternativa al protocollo Telnet (il cui traffico passa in chiaro) al fine di ottenere una shell sicura (Secure Shell = SSH). Oltre a questo suo scopo tradizionale è possibile utilizzare SSH come tunnel per traffico di altra natura, ovviamente anche traffico FTP. La sua implementazione è tipica nel mondo Unix ma trova un certo favore anche in ambiente Windows grazie a software come *OpenSSH for Windows* gratuitamente scaricabile dal sito <http://sshtwindows.sourceforge.net/download/>. La configurazione è piuttosto semplice ma è possibile trovare un semplice tutorial in inglese all’indirizzo <http://www.digitalmediaminute.com/article/1487/setting-up-a-sftp-server-on-windows>
2. FTP Over VPN – Ovviamente, è possibile creare un tunnel VPN (PPTP, L2TP/IPSEC, IPSEC) e inoltrare tutto il traffico sensibile (es. FTP) all’interno del tunnel. Questo approccio se da un lato semplifica l’aspetto della configurazione dei Firewall perimetrali consentendo di aprire solo le porte legate al tunnel e non quelle legate ai protocolli per questo passanti, dall’altro complica la questione in quanto richiede apparati che funzionino da VPN Server e naturalmente configurazioni che possono superare la complessità di gestione tipica del mondo SOHO.
3. FTPS (FTP Over SSL) – Sostanzialmente, si tratta di creare una connessione SSL basata su un certificato digitale, per cifrare tutto il traffico FTP passante (sia quello di controllo che quello relativo ai dati). Per realizzare questa soluzione, occorrerà disporre di un server FTP che gestisca questo tipo di connessione e di un client compatibile. Purtroppo, dal punto di vista della configurazione, nel caso in cui il server si trovi dietro un Firewall / NAT (situazione molto probabile), sorgono alcuni problemi che affronteremo più avanti.

Nel seguito di questo articolo, si è scelto di analizzare a fondo soltanto il caso del FTP Over SSL, per le seguenti ragioni:

1. L’implementazione si basa su software anche disponibili Opensource e nativi per la piattaforma Windows senza richiedere porting di codice Unix o configurazioni a riga di comando (come richiesto da FTP Over SSH)
2. Non è richiesto alcun tipo di hardware di rete particolare o costoso, come invece nel caso delle soluzioni VPN tradizionali.
3. La presenza di un Firewall / NAT presenta difficoltà che meritano una trattazione specifica.

Per il nostro esperimento, adotteremo Filezilla sia come client che come server FTP in quanto si è dimostrata una soluzione particolarmente valida ed è compatibile con l’FTP Over SSL. Inoltre è gratuitamente disponibile sia il client che il server dal sito <http://filezilla-project.org/>

Naturalmente è bene ricordare che nel caso in cui si volesse utilizzare Filezilla (o per la verità, qualunque altro software) per realizzare non un test ma un ambiente di produzione, sarà necessario assicurarsi di possedere la versione più aggiornata del software e controllare periodicamente che non siano state scoperte nuove vulnerabilità a carico della distribuzione utilizzata.

N.B.

Si ricorda che quanto segue presuppone che si conosca il funzionamento di base del software Filezilla (sia il client che il server) ed ovviamente si sappia come installarlo.

LA SOLUZIONE

Come spiegato al principio del presente articolo, esistono due diverse modalità per l'FTP:

1. "Active Mode" – al posto della tcp 21 e tcp 20 del FTP tradizionale, entrano in gioco la tcp 990 (controllo) e la tcp 989 (dati). Permangono le problematiche sulla porta scelta arbitrariamente dal server per comunicare al client dati che originano dalla porta tcp 989 del server stesso.
2. "Passive Mode" – la tcp 990 prende il posto della tcp 21 ed il server indicherà al client quale porta contattare per il trasferimento dati.

Vedremo che a complicare il tutto, esistono due ulteriori possibilità di FTP Over SSL:

1. *Implicit Encryption* – utilizza la tcp 990 e porte superiori alla 1023
2. *Explicit Encryption* – utilizza la tcp 21 e porte superiori alla 1023. In questo caso la porta tcp 21 sarà usata per stabilire una connessione SSL/TLS

Per le motivazioni già descritte noi sceglieremo di adottare la "passive mode" riducendo in questo modo la necessità di configurazione sul client ma aumentandola lato server. Infatti, ipotizzando di avere il server FTP dietro un Firewall / NAT, analizziamo cosa accade durante la creazione di una connessione FTP in "passive mode".

IP CLIENT: 195.168.0.100

IP NAT INTERFACCIA ESTERNA (WAN): 195.168.0.254

IP NAT INTERFACCIA INTERNA (LAN): 192.168.10.1

IP SERVER: 192.168.10.2

CLIENT → NAT: PASV

NAT → SERVER: PASV

SERVER → NAT: 227 Enter passive mode (192,168,10,2,195,80)

NAT → CLIENT: 227 Enter passive mode (195,168,0,254,195,80)

In una normale connessione FTP, il NAT non avrebbe particolari problemi a svolgere le proprie funzioni di "traduzione".

Di seguito sono riportati i log del client FTP:

```
Status: Connecting to 195.168.0.254 ...
Status: Connected with 195.168.0.254. Waiting for welcome message...
Response: 220-FileZilla Server version 0.9.24 beta
Response: 220-written by Tim Kosse (Tim.Kosse@gmx.de)
Response: 220 Please visit http://sourceforge.net/projects/filezilla/
Command: USER mac
Response: 331 Password required for mac
Command: PASS *****
Response: 230 Logged on
Command: SYST
Response: 215 UNIX emulated by FileZilla
Command: FEAT
Response: 211-Features:
Response: MDTM
```

```

Response: REST STREAM
Response: SIZE
Response: MLST type*;size*;modify*;
Response: MLSD
Response: AUTH SSL
Response: AUTH TLS
Response: UTF8
Response: CLNT
Response: MFMT
Response: 211 End
Status: Connected
Status: Retrieving directory listing...
Command: PWD
Response: 257 "/" is current directory.
Command: TYPE A
Response: 200 Type set to A
Command: PASV
Response: 227 Entering Passive Mode (195,168,0,254,195,146).
Command: LIST
Response: 150 Connection accepted
Response: 226 Transfer OK

```

Notiamo come l'indirizzo contenuto nel messaggio 227 che il NAT inoltra al client, riporti correttamente l'indirizzo dell'interfaccia esterna del NAT.

Nel caso in cui si stia utilizzando FTP Over SSL, il messaggio 227 inviato dal server FTP al NAT è cifrato e quindi risulta non comprensibile da quest'ultimo.

Per questo motivo, in tali casi, la prima parte della connessione FTP Over SSL (negoiazione SSL e scambio credenziali), va a buon fine ma la successiva fase di LIST della directory non si conclude positivamente.

Vediamo i log del client in questo caso:

```

Status: Connecting to 195.168.0.254:990 ...
Status: Connected with 195.168.0.254:990, negotiating SSL connection...
Status: SSL connection established. Waiting for welcome message...
Response: 220-FileZilla Server version 0.9.24 beta
Response: 220-written by Tim Kosse (Tim.Kosse@gmx.de)
Response: 220 Please visit http://sourceforge.net/projects/filezilla/
Command: USER mac
Response: 331 Password required for mac
Command: PASS *****
Response: 230 Logged on
Command: SYST
Response: 215 UNIX emulated by FileZilla
Command: FEAT
Response: 211-Features:
Response: MDTM
Response: REST STREAM
Response: SIZE
Response: MLST type*;size*;modify*;
Response: MLSD
Response: AUTH SSL
Response: AUTH TLS
Response: UTF8
Response: CLNT
Response: MFMT
Response: 211 End
Command: PBSZ 0
Response: 200 PBSZ=0
Command: PROT P
Response: 200 Protection level set to P
Status: Connected
Status: Retrieving directory listing...
Command: PWD
Response: 257 "/" is current directory.
Command: TYPE A
Response: 200 Type set to A
Command: PASV
Response: 227 Entering Passive Mode (192,168,10,2,195,147)
Command: LIST
Response: 425 Can't open data connection.

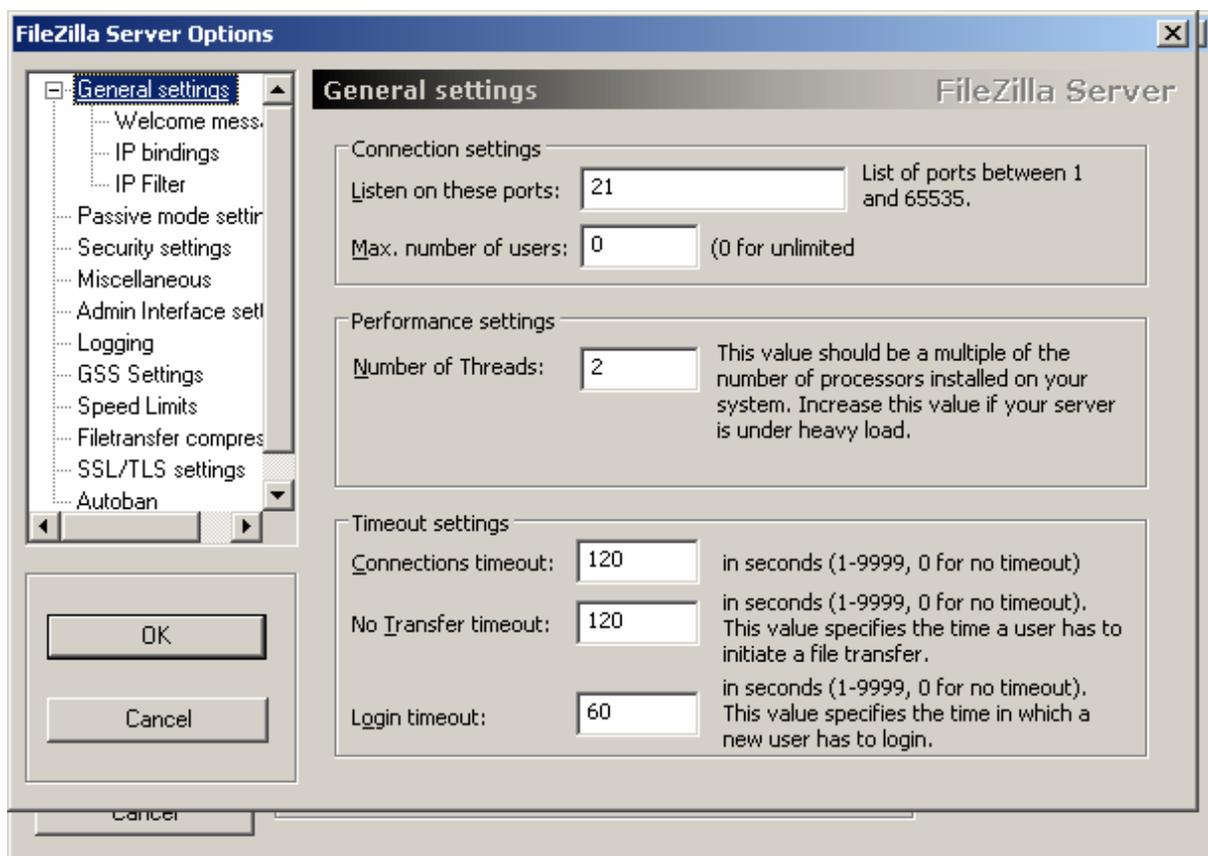
```

Notiamo come l'indirizzo contenuto nel messaggio 227 che il NAT inoltra al client, riporti l'indirizzo del server FTP e non quello dell'interfaccia esterna del NAT. Questo a dimostrare che il NAT non è stato in grado di tradurre correttamente gli indirizzi poiché il traffico è cifrato. Infatti, a differenza del primo esempio, il comando LIST riceve una risposta negativa (425) dal server invece di quella positiva (150).

Per risolvere il problema, occorrerà agire sia sulle impostazioni del server FTP sia sulle regole di inoltro delle porte sul Firewall / NAT.

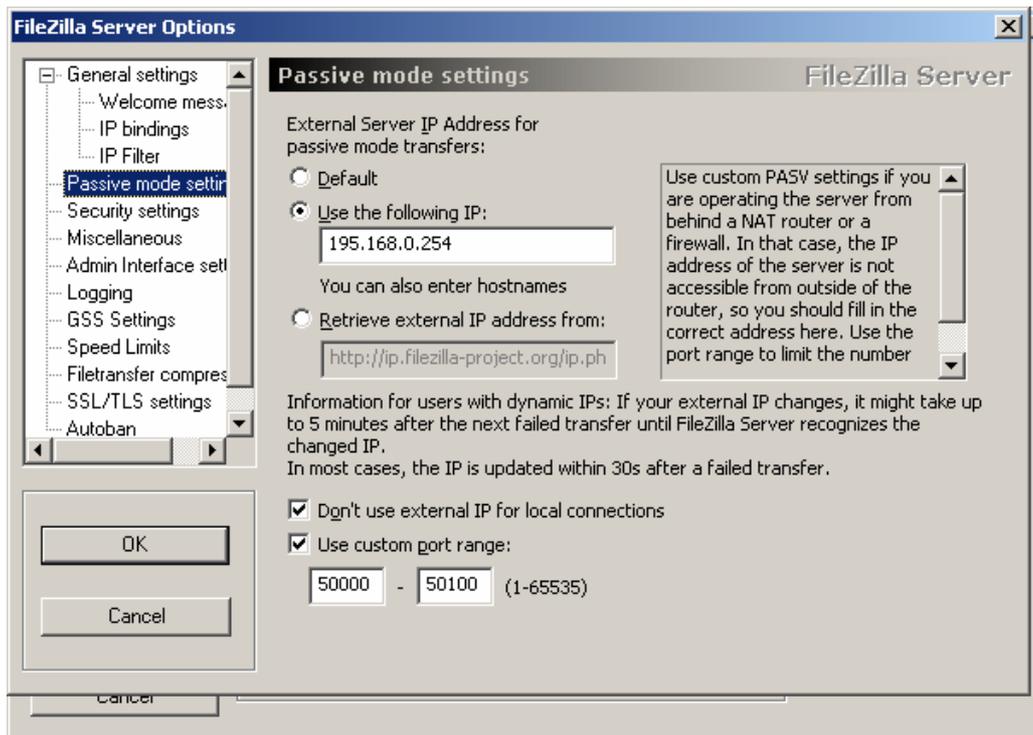
- Impostazioni server FTP

Se accediamo al menu *Settings* della console di amministrazione di Filezilla server e ci spostiamo sotto *General Settings*, troviamo la seguente schermata che ci permette di cambiare la porta su cui il server rimane in attesa delle connessioni FTP tradizionali (per default è la 21).



Passando alla voce *Passive Mode Settings*, troviamo la soluzione ad un primo problema. Infatti qui, nel campo *Use the following IP* possiamo inserire l'indirizzo IP dell'interfaccia esterna del Firewall / NAT (195.168.0.254), evitando in questo modo la mancata traduzione del NAT di cui sopra. Se stiamo usando per l'interfaccia esterna del NAT un IP Pubblico, allora possiamo selezionare la voce *Don't use external IP for local connections*, facendo in modo che questo IP non venga utilizzato per le eventuali connessioni ad altre interfacce del server sul lato intranet ed appartenenti ad indirizzi IP privati.

Attenzione a non selezionare tale voce nel caso in cui si stia per qualche motivo utilizzando un IP privato sull'interfaccia esterna del NAT perché in caso contrario continueremo ad avere l'inconveniente sopra riportato e a non potere effettuare il LIST della directory.

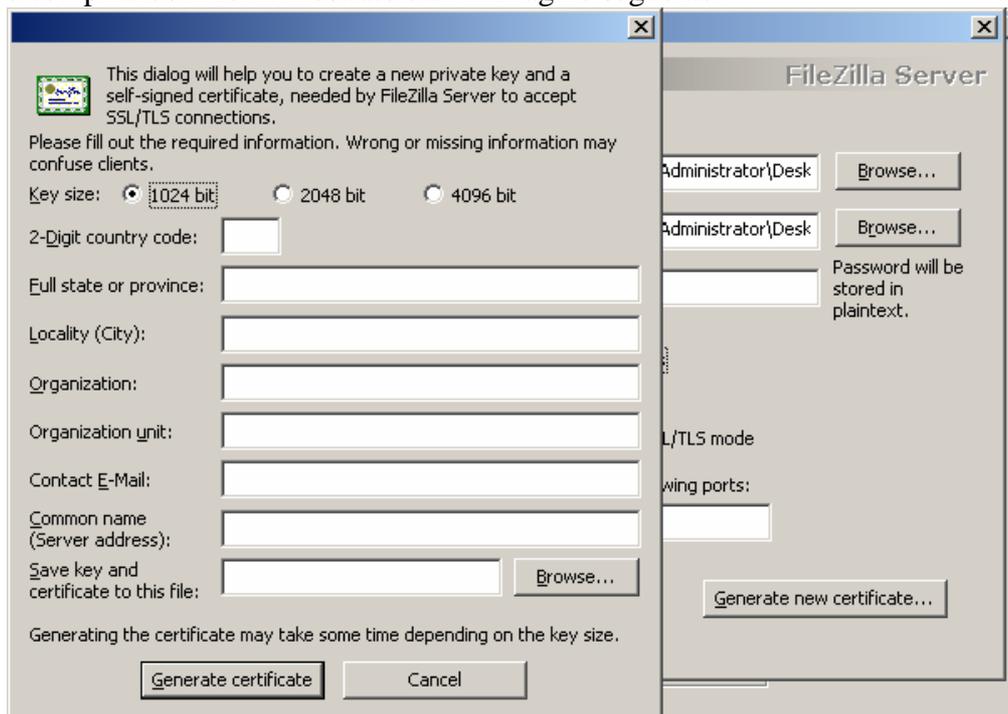


Il campo *Use custom port range* servirà per restringere il numero delle porte che vogliamo che il server utilizzi per la modalità “passive mode” e che comunicherà al client per essere da questo contattato.

Ovviamente, come vedremo, questo range di porte dovrà essere aperto ed inoltrato al server sull'apparato che svolge funzioni di Firewall / NAT.

Spostandoci su *SSL/TLS Settings*, come prima cosa dobbiamo stabilire quale sia il certificato digitale che il server adopererà per la creazione della connessione SSL/TLS.

Possiamo importare un certificato e la relativa chiave privata provenienti da una Certification Authority pubblica oppure generare nostri specifici certificati, tramite il pulsante *Generate new certificate* e compilando il form mostrato dall'immagine seguente.



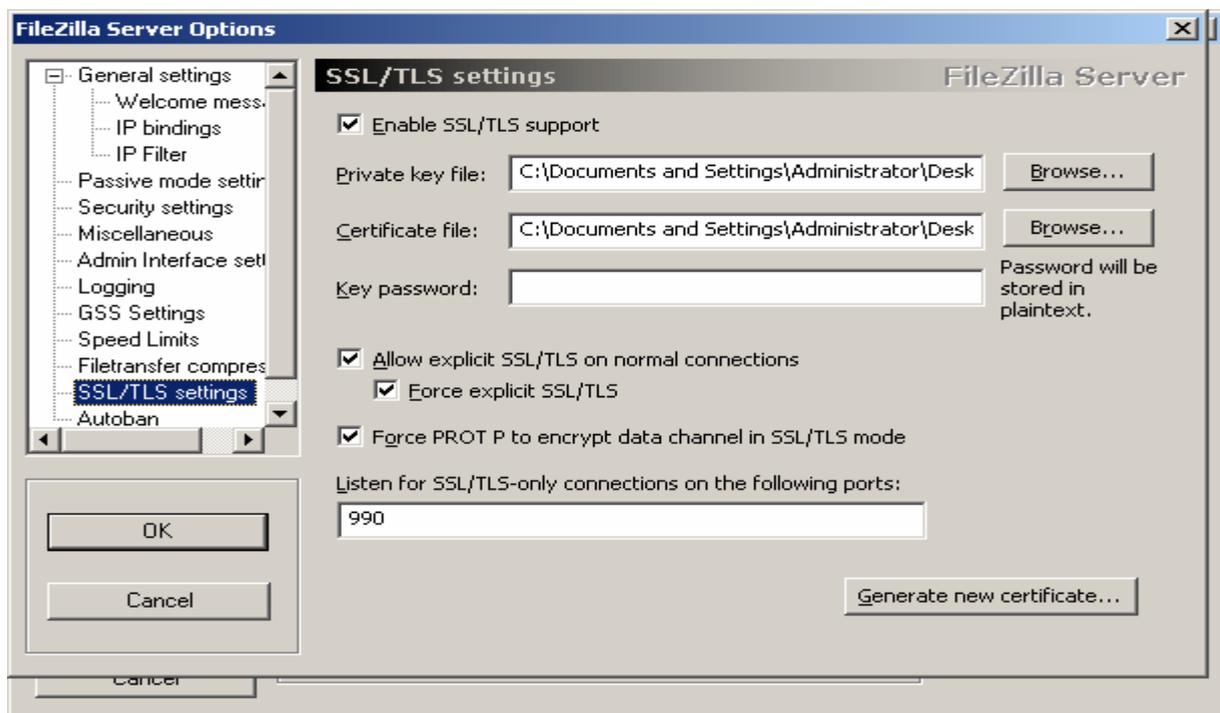
Completata la generazione del certificato, possiamo abilitare *Enable SSL/TLS support* e passare a scegliere quale sarà la porta su cui il server FTP attenderà le richieste di connessione in SSL tramite il campo *Listen for SSL/TLS-only connections on the following ports* (per default è la porta 990) .

Come accennato prima, esistono due ulteriori opzioni possibili per una connessione FTP Over SSL:

1. Implicit Encryption – richiede l’apertura sul Firewall / NAT della sola porta 990 oltre che del suddetto range di porte superiori alla 1023 per il corretto funzionamento della “passive mode”
2. Explicit Encryption – utilizza la tcp 21 e il range di porte superiori alla 1023. In questo caso la porta tcp 21 sarà usata per stabilire una connessione SSL/TLS e non per il normale traffico FTP in chiaro.

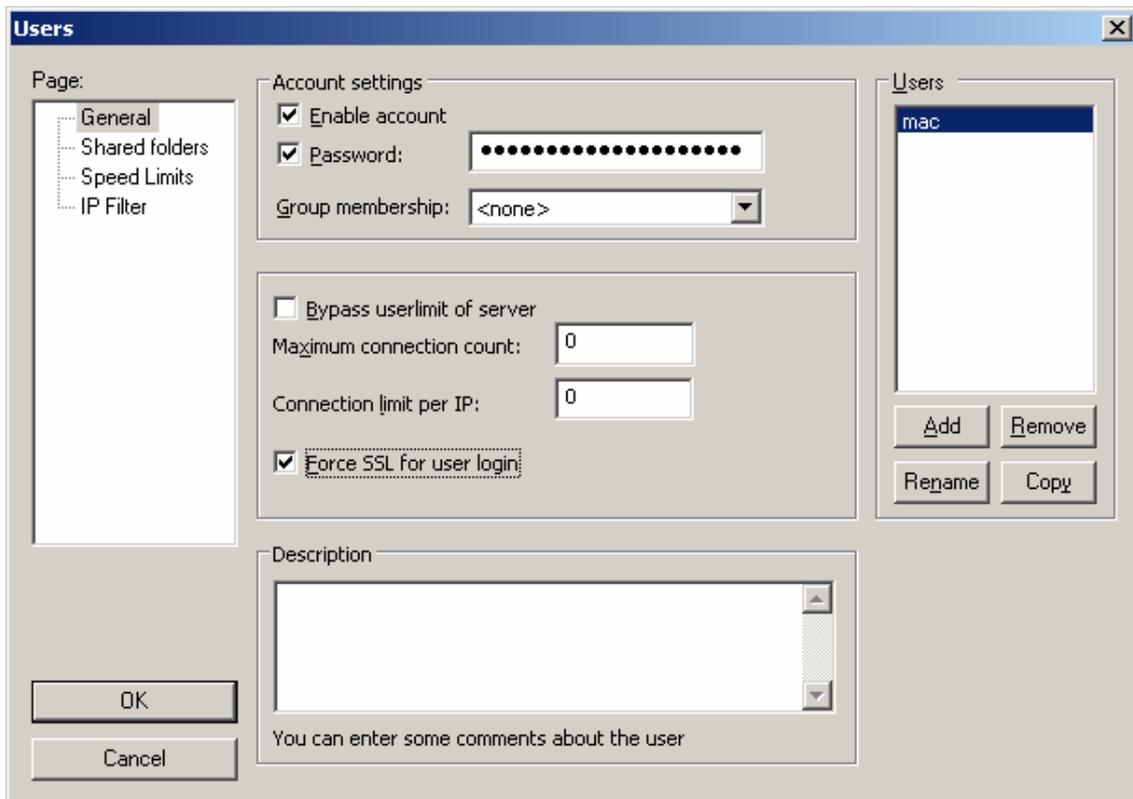
La scelta tra le due possibilità è dipendente dalle politiche di firewall, oltre che dalle impostazioni del client FTP, rimanendo ovviamente uguale in entrambi i casi il livello di sicurezza fornito dalla cifratura del traffico.

Nell’immagine seguente è visibile la finestra di configurazione sul Filezilla server, riguardante l’*explicit* o l’*implicit* mode. Per default viene abilitata la sola *implicit mode* in listening sulla porta 990. Rimane possibile consentire la modalità *explicit*, selezionando l’opzione *Allow explicit SSL/TLS on normal connections* (in aggiunta alla modalità *implicit*), oppure obbligare tutte le connessioni alla porta tcp 21 ad entrare in *explicit* mode (ciò non disabilita la modalità *implicit*).



Quest’ultima scelta è da preferirsi quando si vuole che tutti i client che si connettono alla porta tcp 21 siano rediretti ad una connessione protetta, mentre nel caso in cui si desideri un alto livello di sicurezza soltanto per alcuni client si dovrebbe agire sulle impostazioni di Filezilla server riguardanti gli utenti abilitati e scegliere di forzare SSL per la loro specifica connessione.

Nella figura seguente, si vede come sia stata abilitata per un dato utente la voce *Force SSL for user login* (menu Settings/Users), lasciando agli altri utenti la possibilità di effettuare connessioni FTP tradizionali.



- Impostazioni Firewall / NAT

Quanto detto fin qui, è necessario per impostare adeguatamente il server FTP ma non ha ancora trattato la configurazione del Firewall / NAT che si trova tra il server stesso ed i client FTP. Ricordando che le porte tcp 20 e 989 sono necessarie solo nel caso si scelga l' *active mode* (vivamente sconsigliato per i motivi ricordati in precedenza), le porte con cui avremo a che fare lato firewall saranno le tcp 21, 990 ed un range (il nostro esempio prevede tcp 50000-50100) scelto tra le porte superiori alla 1023 (*passive mode*).

Ovviamente nel caso in cui il Firewall sia anche un NAT, bisognerà creare una mappatura tra porta pubblica e porta privata del NAT ridirigendo il traffico entrante sul NAT al server FTP interno (si rimanda ai manuali specifici dei vari apparati per la corretta configurazione).

Vediamo brevemente i vari scenari:

1. FTP over SSL (Implicit Encryption)
 - a. TCP 990
 - b. TCP 50000-50100
2. FTP over SSL/TLS (Explicit Encryption)
 - a. TCP 21
3. TCP 50000-5010

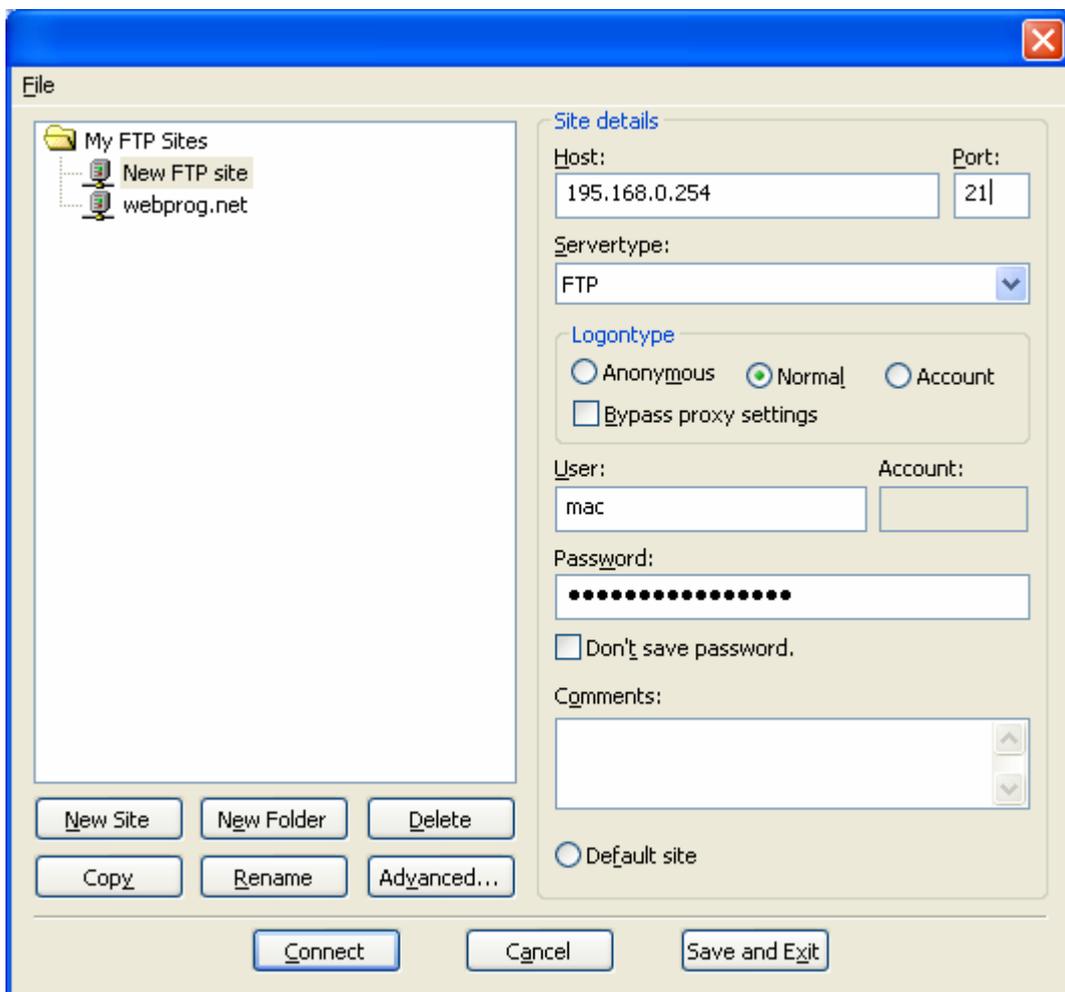
- Impostazioni Client FTP

Sul lato client, Filezilla consente un pieno supporto alle modalità di connessione sicura al server FTP (SSL, TLS, Implicit, Explicit). Vediamone insieme gli esempi di configurazione e i relativi log utili per meglio comprendere le diverse situazioni riscontrabili nella realtà.

Dal menu *File* del client Filezilla scegliamo *Site Manager* e poi creiamo un nuovo profilo di connessione tramite il pulsante *New Site*.

Negli esempi seguenti si ipotizza che l'indirizzo dell'interfaccia esterna del Firewall / NAT sia il 195.168.0.254 (ip pubblico e statico per una connessione WAN).

Esempio n.1a – Tentativo di connessione alla porta tcp 21 in modalità non protetta verso un server FTP su cui l'utente è stato impostato come obbligato al SSL.



Come vediamo dai log, il server FTP risponde con il messaggio “530 SSL Required” e non ci consente la connessione.

```
Status: Connecting to 195.168.0.254 ...
Status: Connected with 195.168.0.254. Waiting for welcome message...
Response: 220-FileZilla Server version 0.9.24 beta
Response: 220-written by Tim Kosse (Tim.Kosse@gmx.de)
Response: 220 Please visit http://sourceforge.net/projects/filezilla/
Command: USER mac
Response: 530 SSL required
Error: Unable to connect!
```

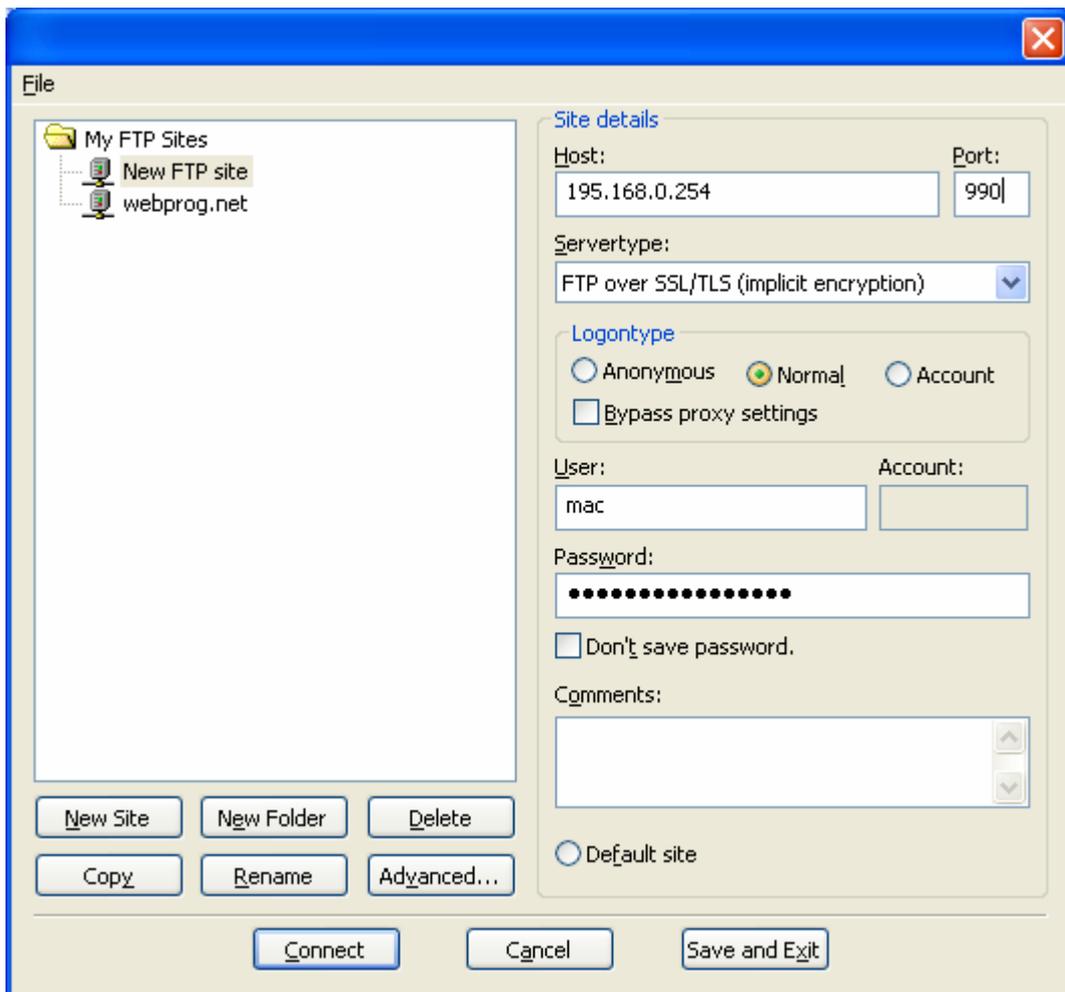
Esempio n.1b – Tentativo di connessione alla porta tcp 21 in modalità non protetta verso un server FTP su cui l’impostazione generale richiede esclusivamente connessioni SSL/TLS per la porta tcp 21 (*explicit mode*).

Anche qui, il log mostra come il server rifiuti il tentativo di connessione in chiaro tramite il messaggio “530 Have to use explicit SSL/TLS bifore logging on”.

Server - SSL/TLS settings: Force explicit SSL/TLS

```
Status: Connecting to 195.168.0.254 ...
Status: Connected with 195.168.0.254. Waiting for welcome message...
Response: 220-FileZilla Server version 0.9.24 beta
Response: 220-written by Tim Kosse (Tim.Kosse@gmx.de)
Response: 220 Please visit http://sourceforge.net/projects/filezilla/
Command: USER mac
Response: 530 Have to use explicit SSL/TLS before logging on.
Error: Unable to connect!
```

Esempio n.2 – Connessione alla porta tcp 990 su un server impostato per l’*implicit encryption*

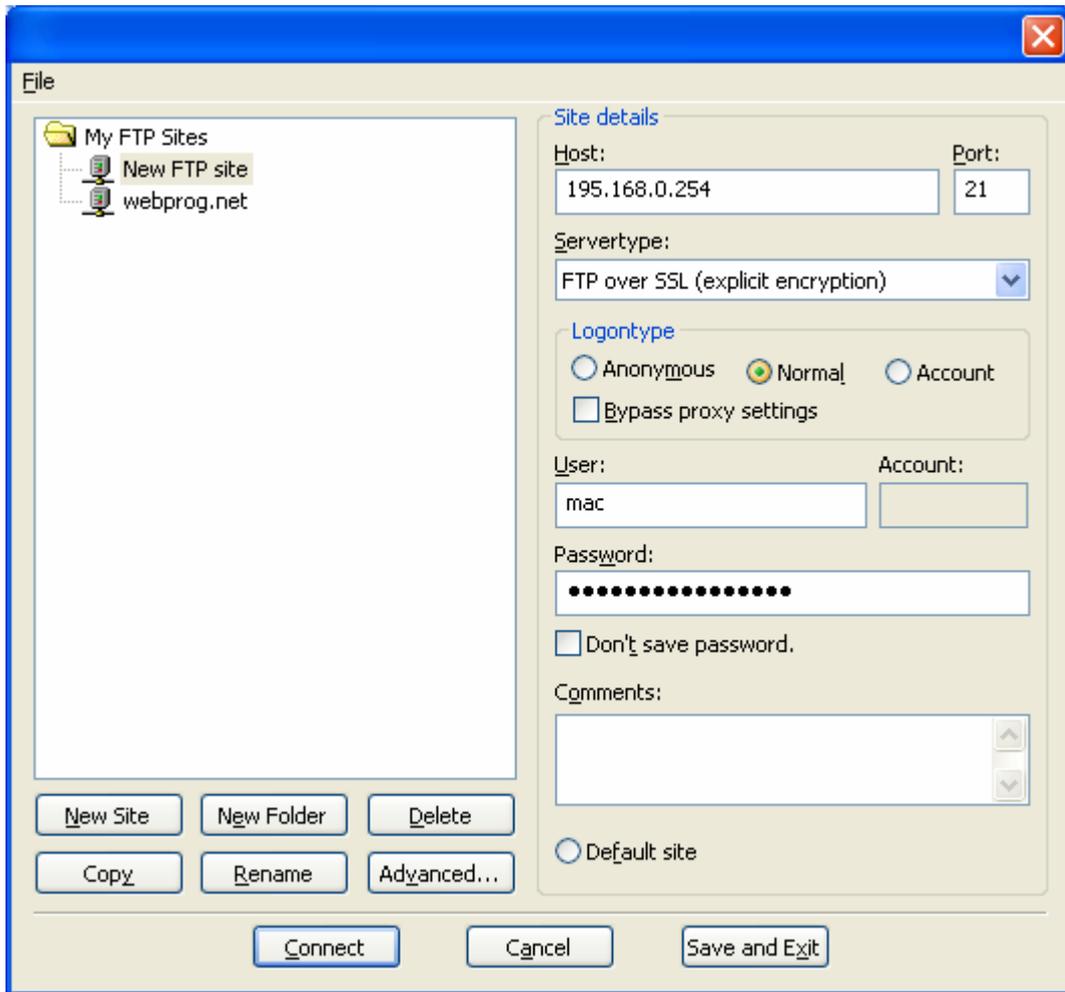


Dai log notiamo I messaggi di stato riguardanti la negoziazione ed il perfezionamento della connessione SSL. Infine, grazie alle impostazioni lato server (sopra riportate), possiamo ricevere risposta affermativa al comando di LIST che precedentemente il Firewall / NAT impediva.

```

Status:      Connecting to 195.168.0.254:990 ...
Status:      Connected with 195.168.0.254:990, negotiating SSL connection...
Status:      SSL connection established. Waiting for welcome message...
Response:    220-FileZilla Server version 0.9.24 beta
Response:    220-written by Tim Kosse (Tim.Kosse@gmx.de)
Response:    220 Please visit http://sourceforge.net/projects/filezilla/
Command:     USER mac
Response:    331 Password required for mac
Command:     PASS *****
Response:    230 Logged on
Command:     SYST
Response:    215 UNIX emulated by FileZilla
Command:     FEAT
Response:    211-Features:
Response:    MDTM
Response:    REST STREAM
Response:    SIZE
Response:    MLST type*;size*;modify*;
Response:    MLSD
Response:    AUTH SSL
Response:    AUTH TLS
Response:    UTF8
Response:    CLNT
Response:    MFMT
Response:    211 End
Command:     PBSZ 0
Response:    200 PBSZ=0
Command:     PROT P
Response:    200 Protection level set to P
Status:      Connected
Status:      Retrieving directory listing...
Command:     PWD
Response:    257 "/" is current directory.
Command:     TYPE A
Response:    200 Type set to A
Command:     PASV
Response:    227 Entering Passive Mode (195,168,0,254,195,148)
Command:     LIST
Response:    150 Connection accepted
Status:      SSL connection established
Response:    226 Transfer OK
Status:      Directory listing successful
  
```

Esempio n.3a – Connessione alla porta tcp 21 verso un server impostato in modalità *explicit encryption* su SSL.



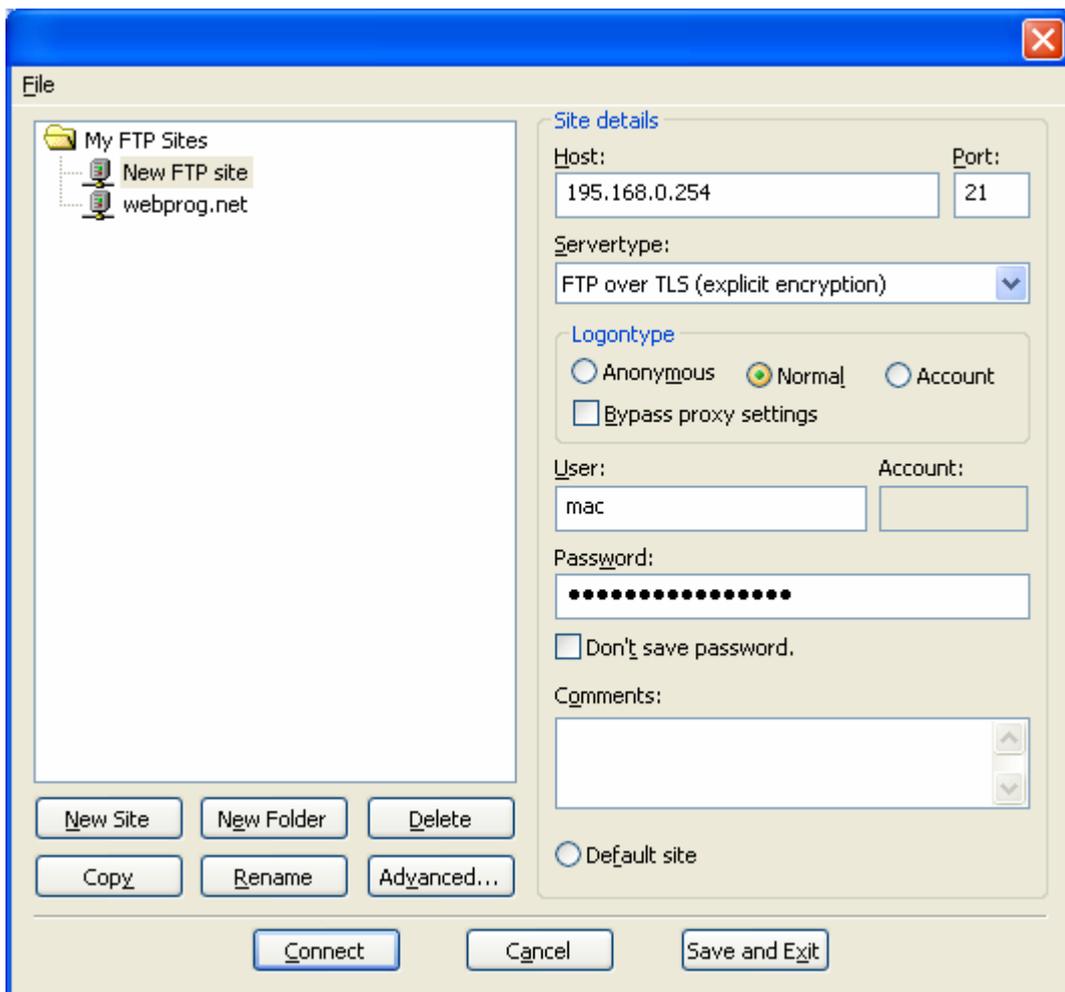
Risulta evidente la negoziazione della connessione che utilizza SSL per l'autenticazione (AUTH SSL) e la riuscita finale del comando di LIST.

```
Status: Connecting to 195.168.0.254 ...
Status: Connected with 195.168.0.254, negotiating SSL connection...
Response: 220-FileZilla Server version 0.9.24 beta
Response: 220-written by Tim Kosse (Tim.Kosse@gmx.de)
Response: 220 Please visit http://sourceforge.net/projects/filezilla/
Command: AUTH SSL
Response: 234 Using authentication type SSL
Status: SSL connection established. Waiting for welcome message...
Command: USER mac
Response: 331 Password required for mac
Command: PASS *****
Response: 230 Logged on
Command: SYST
Response: 215 UNIX emulated by FileZilla
Command: FEAT
Response: 211-Features:
Response: MDTM
Response: REST STREAM
Response: SIZE
Response: MLST type*;size*;modify*;
Response: MLSD
Response: AUTH SSL
Response: AUTH TLS
Response: UTF8
Response: CLNT
Response: MFMT
```

```

Response: 211 End
Command: PBSZ 0
Response: 200 PBSZ=0
Command: PROT P
Response: 200 Protection level set to P
Status: Connected
Status: Retrieving directory listing...
Command: PWD
Response: 257 "/" is current directory.
Command: TYPE A
Response: 200 Type set to A
Command: PASV
Response: 227 Entering Passive Mode (195,168,0,254,195,149)
Command: LIST
Response: 150 Connection accepted
Status: SSL connection established
Response: 226 Transfer OK
Status: Directory listing successful
    
```

Esempio n.3b – Connessione alla porta tcp 21 verso un server impostato in modalità *explicit encryption* su TLS.



```

Status:      Connecting to 195.168.0.254 ...
Status:      Connected with 195.168.0.254, negotiating SSL connection...
Response:    220-FileZilla Server version 0.9.24 beta
Response:    220-written by Tim Kosse (Tim.Kosse@gmx.de)
Response:    220 Please visit http://sourceforge.net/projects/filezilla/
Command:     AUTH TLS
Response:    234 Using authentication type TLS
Status:      SSL connection established. Waiting for welcome message...
Command:     USER mac
Response:    331 Password required for mac
Command:     PASS *****
Response:    230 Logged on
Command:     SYST
Response:    215 UNIX emulated by FileZilla
Command:     FEAT
Response:    211-Features:
Response:    MDTM
Response:    REST STREAM
Response:    SIZE
Response:    MLST type*;size*;modify*;
Response:    MLSD
Response:    AUTH SSL
Response:    AUTH TLS
Response:    UTF8
Response:    CLNT
Response:    MFMT
Response:    211 End
Command:     PBSZ 0
Response:    200 PBSZ=0
Command:     PROT P
Response:    200 Protection level set to P
Status:      Connected
Status:      Retrieving directory listing...
Command:     PWD
Response:    257 "/" is current directory.
Command:     TYPE A
Response:    200 Type set to A
Command:     PASV
Response:    227 Entering Passive Mode (195,168,0,254,195,150)
Command:     LIST
Response:    150 Connection accepted
Status:      SSL connection established
Response:    226 Transfer OK
Status:      Directory listing successful

```

In tutti gli esempi riportati, si nota come il comando USER e PASS siano successivi allo stabilirsi di una connessione protetta (SSL o TLS), a riprova del fatto che le credenziali di accesso al server (ed i dati successivamente trasferiti) passino sulla rete in sicurezza.