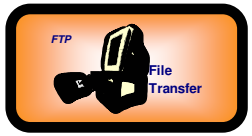
Communications Server z/OS V1R5 and V1R6 Technical Update

FTP Secured With SSL on z/OS®

© Copyright International Business Machines Corporation 2004. All rights reserved.

Topics

- z/OS V1R5
  - SSL/TLS enabled FTP overview
    - Logging in using a client certificate without a password



z/OS FTP - an ongoing story

z/OS V1R2 FTP items

- ✓Native ASCII file tagging - part of Enhanced Ascii support
- ✓ISPF statistics maintenance - for all PDS/PDSE updates
- ✓Socksified FTP client
- ✓SSL/TLS enabled and Kerberized FTP
- ✓Stream-mode restart of HFS file transfers
- ✓FTP server user level options - set user defaults at FTP server side
- ✓Protect against bounce attacks
- ✓Restrict output from DIR commands - using the Catalog Search Interface (CSI)
- ✓Enhanced tracing functions - more granular trace functions
- ✓Surrogate RACF support for anonymous mode
- ✓RFC currency

z/OS V1R4 FTP items

- ✓Detailed FTP server activity logging to SyslogD
- ✓Provide base set of common data to all FTP server security exits
- ✓Enhance FTP server security exit interface to support IPv6 addresses
- ✓Provide a scratch-pad to FTP server security exits for inter-exit communication
- ✓Configure a single-byte character set substitution character for non-mappable code points
- ✓Support new code standard GB18030 for People's Republic of China
- ✓IPv6 enable both FTP server and client

z/OS V1R5 FTP items

- ✓Support allocation of target PDS(E) like existing source PDS(E)
- ✓Enable user to specify if a PDS or a PDSE should be created
- ✓Allow configuration control of passive data port range on FTP server
- ✓Support use of extended passive (EPSV) and active mode (EPRT) for NAT firewall relief
- ✓Provide consistent error codes from FTP client
- ✓Enable MVS syslog message when batch FTP client operation fails (for automation purposes)
- ✓Allow SSL/TLS login to FTP server without password when client authentication is used
- ✓Deliver FTP server load module as RMODE=ANY

z/OS V1R6 FTP items

- ✓A new callable FTP client programming interface that allows a user program full control over every step in a client FTP execution
  - ▸z/OS V1R6 provides a callable interface to be used from COBOL, PL/1, Assembler programs
  - ▸All z/OS FTP client functions are supported through the callable interface
- ✓New MBCS and DBCS support in z/OS FTP built on z/OS operating system functions and conversion tables - as currently implemented by the iconv() function
  - ▸Old DBCS support continues to be supported, but users are encouraged to move to the new support

FTP secured with SSL/TLS

FTP protocol extensions for secure FTP - used for both SSL/TLS and Kerberos security for FTP

- Secure FTP refers to using the standard FTP protocol with one or more security extensions to improve security of data transfers
- "*FTP Security Extensions*", RFC2228 - defines a set of new FTP protocol commands and replies for negotiating secure FTP sessions.
- This RFC defines a framework for securing FTP. SSL/TLS and Kerberos are just two of many potential security mechanisms that could be used with FTP. Others will likely be defined in the future.
- The commands and replies are generic and are used to implement both Kerberos-based and SSL/TLS-based secure FTP sessions.
- Please note that secure FTP (sometimes referred to as ftps) as discussed in this presentation has nothing to do with what is known as ***sftp***
  - sftp is a file transfer protocol under the umbrella of SSH (Secure Shell)
    - SSH is available in an officially supported version for z/OS V1R4+ since May 2004
    - The sftp protocol in SSH on z/OS supports HFS file transfers, not MVS data sets
    - You can use sftp for transfer of HFS files to/from z/OS in an environment that has chosen to standardize on use of SSH
    - You need an SSH sftp client to exchange files with an SSH sftp server
  - sftp has absolutely nothing to do with the normal FTP standards as defined in RFC 959
  - The sftp protocol is its own protocol and sftp under SSH does in no way interoperate with normal FTP

FTP protocol extensions for secure FTP

- **AUTH**
  - Sent by client to server with information about which security mechanism the client requests to use. Supported values for z/OS are GSSAPI (used with Kerberos V5 support) and TLS, TLS-C, TLS-P, and SSL (used with SSL/TLS support)
- **ADAT**
  - Contains optional security data as required by the security mechanism established with the AUTH command. Examples of security data to be sent via an ADAT command is a Kerberos ticket. Binary data is encoded in Base64 encoding. Server reply may include security data from server to client. The SSL/TLS support does not use the ADAT command.
- **PBSZ**
  - A numeric value to establish the size of the data buffer to be exchanged bwteen the client and the server. PBSZ is required, but SSL/TLS doesn't need it so for SSL/TLS you'll see a PBSZ 0 command,
- **PROT**
  - Requesting level of data connection protection:
    - **C** - Clear
    - **S** - Safe (authenticated, but not encrypted) - not supported by SSL/TLS, but is supported for Kerberos
    - **P** - Private (both encrypted and authenticated)

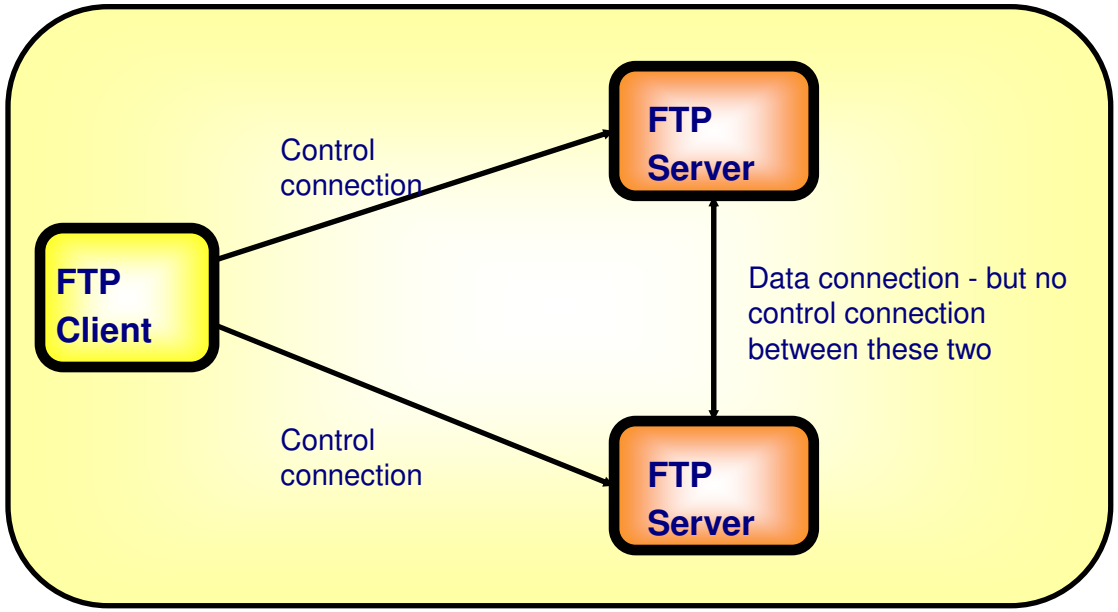
SSL/TLS support in FTP

- The implementation of SSL/TLS in the FTP protocol is described in a draft RFC titled "Securing FTP with TLS".
- To support SSL/TLS, the AUTH command must be exchanged over the control connection
- If SSL/TLS is successfully negotiated, then the control connection will always be encrypted (unless a NULL cipher method is chosen) and authenticated.
- Protection of the data connection(s) is determined based on the PBSZ and PROT command. If a PROT C (clear) command is received on the control connection, then data connections will not be secured.
- Server configuration options determine the policies of the FTP server instance in terms of what it requires as minimum levels of security:
  - **SECURE_FTP** - is an AUTH command required or optional
  - **SECURE_LOGIN** - is a client certificate required to log in to the FTP server
  - **SECURE_PASSWORD** - if client authentication is used, is a password required or optional (z/OS V1R5)
  - **SECURE_DATACONN** - does the server require the data connection to be secure or not

SSL/TLS support in FTP

- The server cannot initiate negotiation of these options, only the client can.
  - The server configuration options are used by the server to accept or reject the terms the client tries to negotiate.
- The standards specifically exclude three-way proxy transfers from the SSL/TLS support - so a data connection established by a client between two servers cannot be SSL/TLS secured.
- Until z/OS V1R5 a remote user must always send USER and PASS commands - even when client certificate authentication is done.
- Current IETF standard is that SSL/TLS must be negotiated using the FTP AUTH command
  - There used to be a separate FTP control port (990) defined for which implicit SSL/TLS was defined - as soon as a client connected to that port, an SSL/TLS handshake would start (without flowing an AUTH command). This type of 'negotiation' has been deprecated and is no longer recommended by the IETF.

Proxy or third-party or three-way file transfers



Once upon a time this capability might have seemed to be a useful and innocent function. Fact is that this function has caused considerable security-related headaches throughout the years and the function is today generally not recommended although it is supported by many FTP clients and servers.

When current security issues have been addressed, this may one day again become a useful function.

RFC2228 "FTP Security Extensions" upon which both Kerberos and SSL/TLS FTP security is based, states about third party file transfers (proxy transfers):

☞"*Third party file transfers cannot be secured using these extensions, since a security context cannot be established between two servers using these facilities (no control connection exists between servers over which to pass ADAT tokens). Further work in this area is deferred*"

Use three-way proxy FTP with care or better, disable use of it on your servers!

!

Enabling SSL/TLS support on the FTP server

EXTENSIONS AUTH_TLS

- Specifies that TLS authentication is supported. This means that the server supports receiving the AUTH command with a value of TLS, TLS-C, TLS-P or SSL.

SECURE_FTP

- **REQUIRED** - Specifies that authentication is required. The server requires receiving the AUTH command before the user can logon.
- **ALLOWED** (default) - Specifies that authentication is supported but not required.

SECURE_LOGIN

- **VERIFY-USER** - Specifies that the TLS handshake process authenticates the client certificate and also provides optional access control to the FTP server port number through the installation's SAF compliant security product - access checked for SERVAUTH resource named EZB.FTP.<system-name>.<ftp-daemon-name>.PORTxxxx - where xxxx is the control port number (example: PORT0021). The client certificate must map to a user ID that matches the user ID received on the USER command.
- **REQUIRED** - Specifies that the FTP server must receive the client certificate. If the certificate is not received during the TLS handshake or it isn't valid, then the login is rejected.
- **NO_CLIENT_AUTH** (default) - Specifies that the FTP server does not request the client certificate.

SECURE_PASSWORD (this is a z/OS V1R5 extension)

- **REQUIRED** - passwords are always required even if client authentication is used
- **OPTIONAL** - if a valid client certificate is presented, it matches a user definition in the security products, and that the security product user ID matches the user ID that is received on the FTP USER command - then a password is not needed.

Enabling SSL/TLS support on the FTP server *(continued)*

SECURE_DATACONN

- **NEVER** - The data channel is required to NOT be integrity protected NOR encrypted. The server will ONLY accept the PROT C command.
- **CLEAR** (default) - The data channel is not required to be integrity protected or encrypted. The server will accept the PROT C and PROT P command.
- **PRIVATE** - The data channel is required to be integrity protected and is required to be encrypted. The client must issue a valid AUTH command before attempting to logon to the FTP server. The server accepts the PROT P command.

Log in to FTP server without password in z/OS V1R5 if client certificate is used

- New server FTP.DATA statement to indicate that password processing is REQUIRED or is OPTIONAL
  - SECURE_PASSWORD REQUIRED | OPTIONAL
    - REQUIRED
      - Specifies that a password is required to login a client whose session is protected by the TLS security mechanism.
    - OPTIONAL
      - Specifies that the password is not required if the client provides a certificate that can be used to authenticate the user.
- To allow login without a password, the certificate that is received from the client must be registered in the security product and must be associated with the user ID that is passed on the USER command to the FTP server.
  - You can use RACDCERT ADD command to register and associate the certificate.
- If you code SECURE_PASSWORD OPTIONAL, you must code SECURE_LOGIN VERIFY_USER or SECURE_LOGIN REQUIRED to require the client certificate.
- SECURE_PASSWORD support has been PTFed back to z/OS V1R4: APAR PQ84185

When is a password optional and when is it required?

When the certificate is registered in the security product and is associated with the user ID that is passed in on the USER command, the SECURE_PASSWORD statement value determines the action taken during the login procedure:

SECURE_PASSWORD	SECURE_LOGIN	Action
REQUIRED	VERIFY_USER REQUIRED	prompt for a password
OPTIONAL	VERIFY_USER REQUIRED	authenticate with the certificate

When either the certificate is not registered in the security product or is not associated with the user ID that is passed in on the USER command, the SECURE_LOGIN statement value determines the action during the login procedure:

SECURE_LOGIN	SECURE_PASSWORD	Action
VERIFY_USER	REQUIRED OPTIONAL	fail the login
REQUIRED	REQUIRED OPTIONAL	prompt for password

Migration Concerns for optional passwords

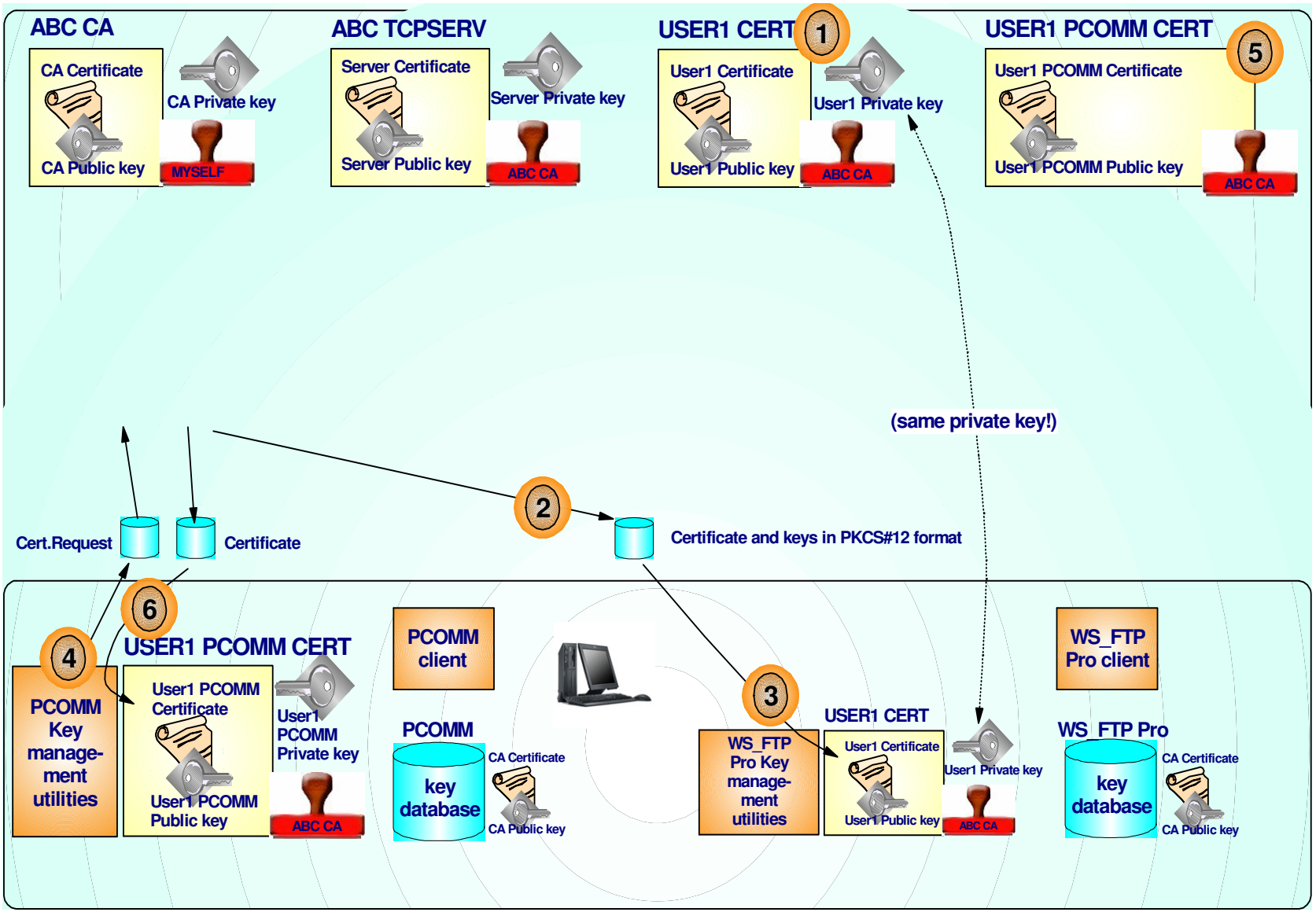
- There is no concern if the function is not requested.
  - If the new statement is not coded, then the behavior defaults to always require a password - this is the current behavior of the server.

- If SECURE_PASSWORD OPTIONAL is coded and the user is authenticated with a certificate, there is no password prompt. A batch job coded as follows will have a problem:

```
9.37.112.22  21
user33
my_password
cd /u/user33
. . .
```

- Since there is no password prompt, the input data my_password is processed as a subcommand and will cause an error. A solution for this is to code the userid and password on the same input line.

Client certificate management



Create personal certificate and keys for USER1

```
//ALFREDCI JOB 1,ALFRED,CLASS=A,MSGCLASS=X,NOTIFY=USER1
//*
//IEFPROC  EXEC PGM=IKJEFT01,REGION=4M,DYNAMNBR=10
//SYSTSPRT DD SYSOUT=*                  BATCH TSO SESSION LOG
//SYSTSIN  DD *
RACDCERT ID(USER1) GENCERT +
         SUBJECTSDN(CN('USER1 CERT') +
         OU('CS Z/OS') +
         O('IBM') +
         C('US')) +
         NOTBEFORE(DATE(2004-01-01)) +
         NOTAFTER(DATE(2010-12-31)) +
         WITHLABEL('USER1 CERT') +
         SIGNWITH(CERTAUTH LABEL('ABC CA'))
RACDCERT ID(USER1) ADDRING(USER1RING)
RACDCERT ID(USER1) CONNECT(CERTAUTH LABEL('ABC CA') +
         RING(USER1RING)
RACDCERT ID(USER1) CONNECT(LABEL('USER1 CERT') +
         RING(USER1RING) DEFAULT)
RACDCERT ID(USER1) LISTRING(USER1RING)
/*
```

Create personal certificate and keys for user USER1

Create keyring for USER1

Connect both our CA certificate and USER1's personal certificate to the keyring and set USER1's personal certificate as the default.

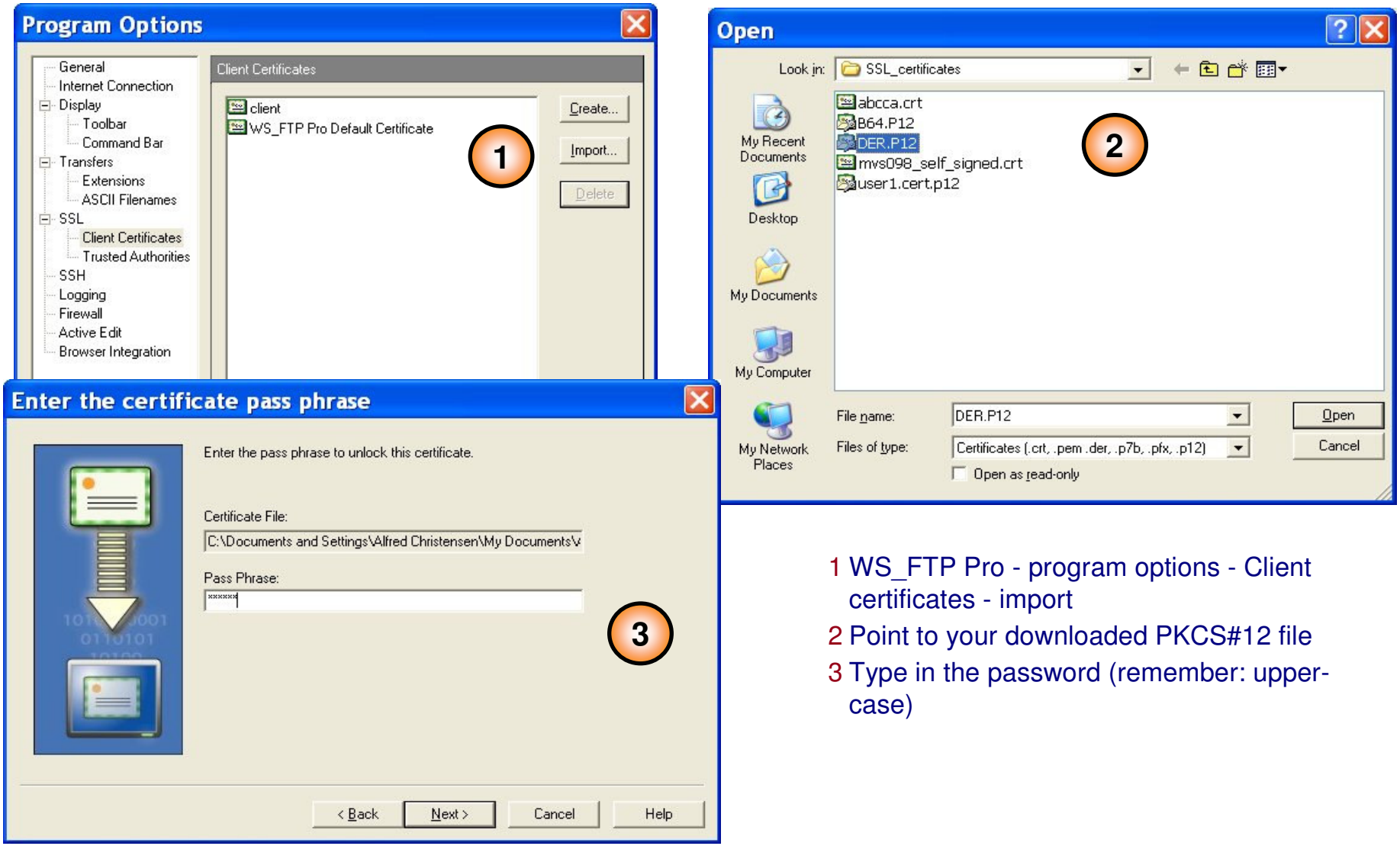
```
Digital ring information for user USER1:

  Ring:
       >USER1RING<
  Certificate Label Name              Cert Owner     USAGE      DEFAULT
  ------------------------------      -----------    --------   -------
  ABC CA                              CERTAUTH       CERTAUTH     NO
  USER1 CERT                          ID(USER1)      PERSONAL     YES
```

Export personal certificate and keys to password-protected data set (PKCS#12 format)

```
//ALFREDCI JOB 1,ALFRED,CLASS=A,MSGCLASS=X,NOTIFY=USER1
//*
//IEFPROC  EXEC PGM=IKJEFT01,REGION=4M,DYNAMNBR=10
//SYSTSPRT DD SYSOUT=*                  BATCH TSO SESSION LOG
//SYSTSIN  DD *
RACDCERT EXPORT (LABEL('USER1 CERT')) +
         DSN('USER1.CERT.DER.P12') +
         FORMAT(PKCS12DER) +
         PASSWORD('xxxxxx')
/*
```

- In this example, we create USER1's personal certificate and key pair on z/OS and then export both the certificate and the key pair to be installed on the workstation.
  - An alternative approach is to create the key pair on the workstation and then request a certificate to be signed by the CA certificate (refer to the PCOMM flow two charts back)
- A PKCS#12 file does not just hold the certificate - it also holds the matching private key. That's why the file itself needs password protection.
- A DER encoded file is a binary file and must be downloaded to the workstation in binary form.
- The certificate and keys could also be exported into a Base64 encoded format (that must be downloaded as a text file), but I had problems getting WS_FTP Pro to import such a file. The DER encoded file worked fine with both PCOM and WS_FTP Pro.
- The password is case sensitive - make sure you enter it in upper-case when you are prompted for the password at the workstation.

Import personal certificate and key into WS_FTP Pro

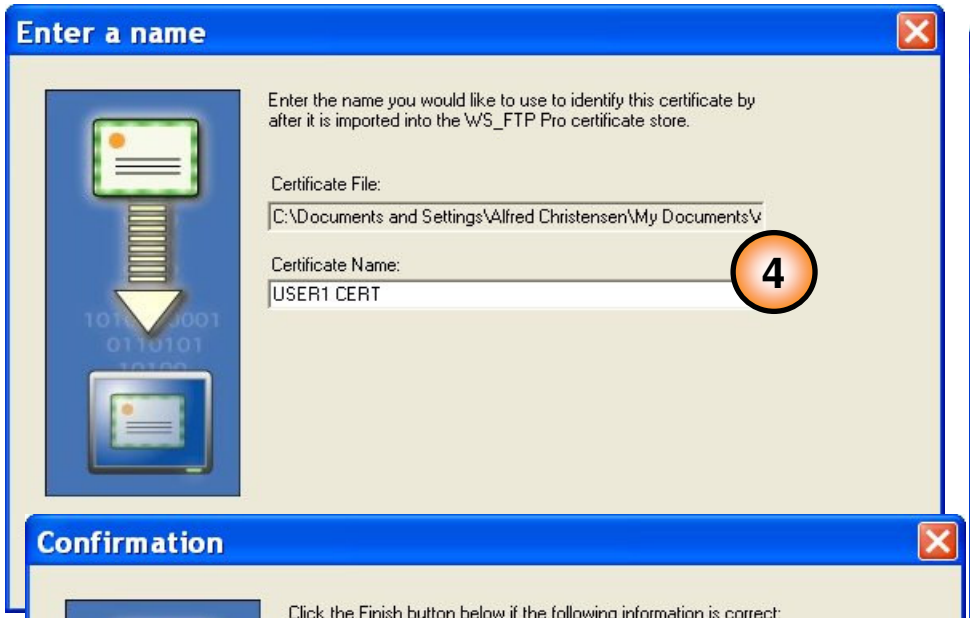


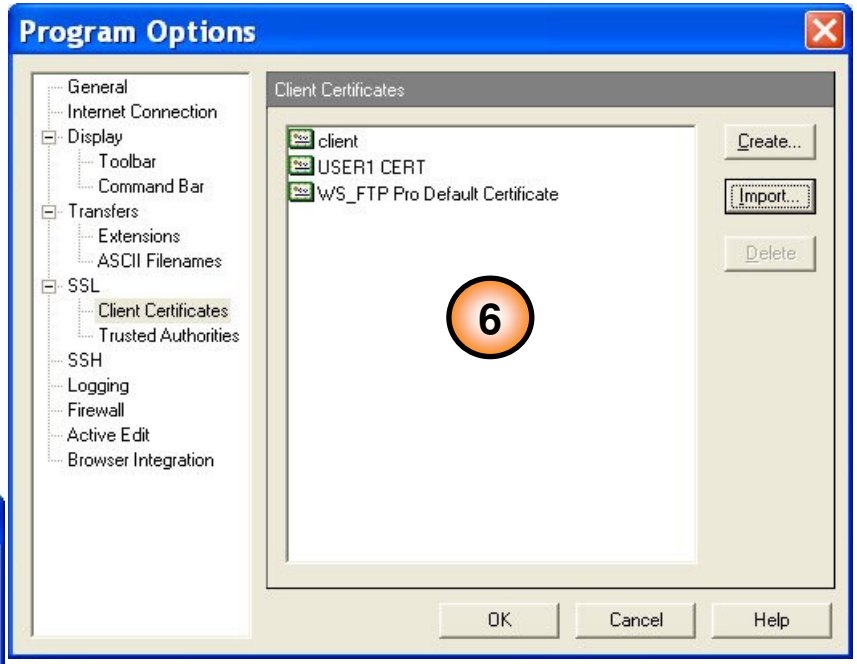
1 WS_FTP Pro - program options - Client certificates - import

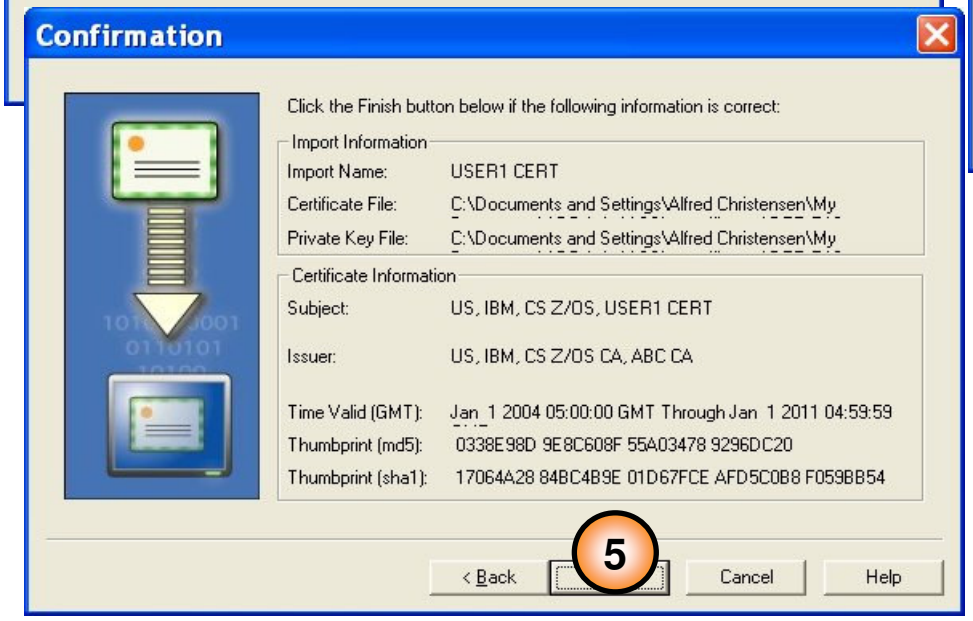
2 Point to your downloaded PKCS#12 file

3 Type in the password (remember: upper-case)

Import personal certificate and key into WS_FTP Pro





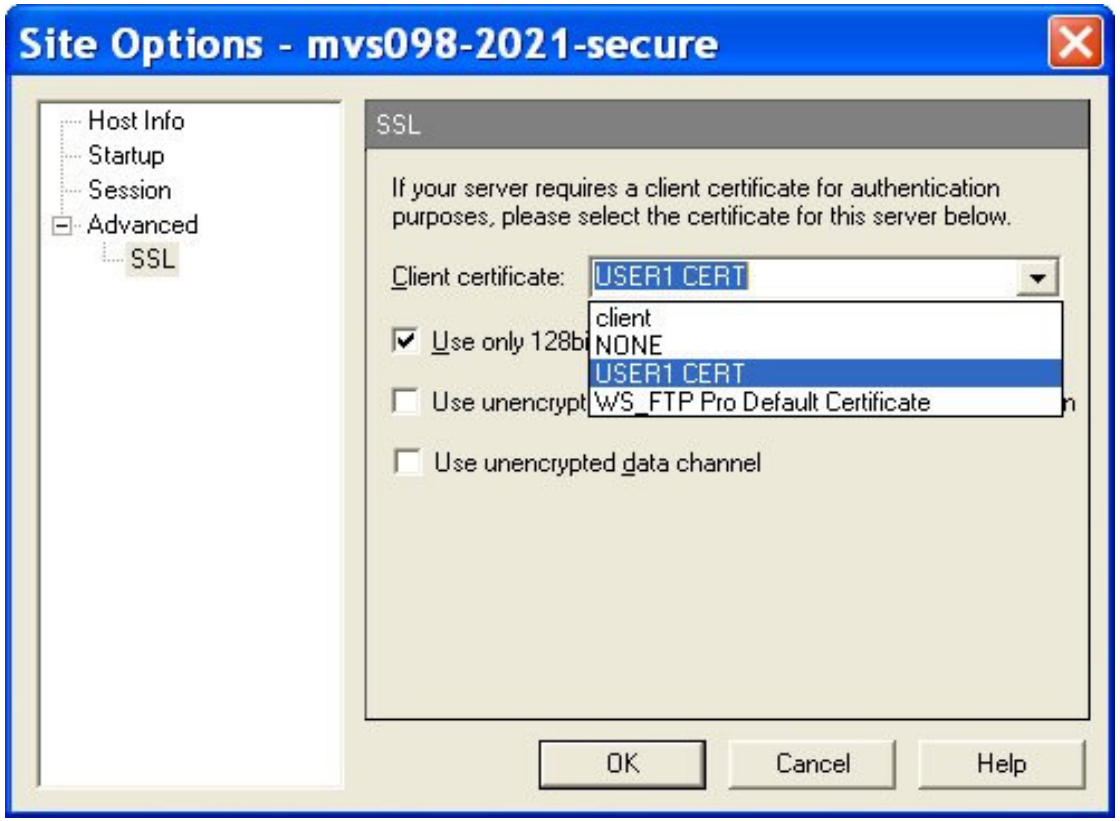


4 Assign a label for it in WS_FTP Pro's key database

5 Verify it is the correct thing

6 And you're done

Configure WS FTP Pro for client authentication



1 For the FTP server connection in question, select the SSL tab and select the client certificate you want to use (the one we installed earlier)

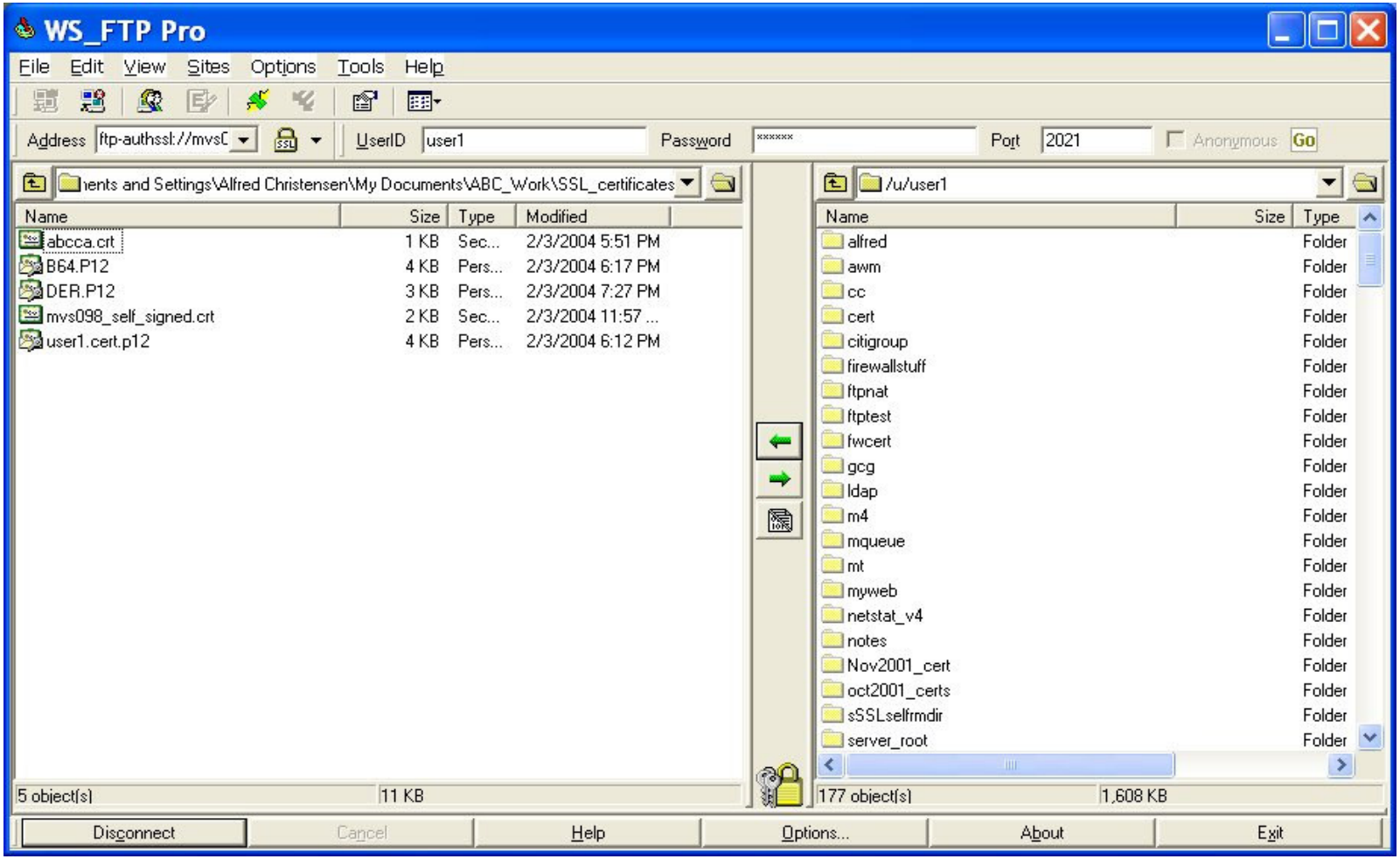
FTP server configuration for client authentication

```
EXTENSIONS        AUTH_TLS          ; Enable TLS authentication
                                    ; Default is disabled.
SECURE_FTP        REQUIRED          ; Authentication indicator
                                    ; ALLOWED        (D)
                                    ; REQUIRED
SECURE_LOGIN      VERIFY_USER       ; Authorization level indicator
                                    ; NO_CLIENT_AUTH (D)
                                    ; REQUIRED
                                    ; VERIFY_USER
SECURE_CTRLCONN   PRIVATE           ; Minimum level of security for
                                    ; the control connection
                                    ; CLEAR          (D)
                                    ; SAFE
                                    ; PRIVATE
SECURE_DATACONN   PRIVATE           ; Minimum level of security for
                                    ; the data connection
                                    ; NEVER
                                    ; CLEAR          (D)
                                    ; SAFE
                                    ; PRIVATE
;SECURE_PBSZ      16384             ; Kerberos maximum size of the
                                    ; encoded data blocks
                                    ; Default value is 16384
                                    ; Valid range is 512 through 32768
```

```
; Name of a ciphersuite that can be passed to the partner during
; the TLS handshake. None, some, or all of the following may be
; specified. The number to the far right is the cipherspec id
; that corresponds to the ciphersuite's name.
;CIPHERSUITE      SSL_NULL_MD5      ; 01
;CIPHERSUITE      SSL_NULL_SHA      ; 02
 CIPHERSUITE      SSL_RC4_MD5_EX    ; 03
 CIPHERSUITE      SSL_RC4_MD5       ; 04
 CIPHERSUITE      SSL_RC4_SHA       ; 05
 CIPHERSUITE      SSL_RC2_MD5_EX    ; 06
 CIPHERSUITE      SSL_DES_SHA       ; 09
 CIPHERSUITE      SSL_3DES_SHA      ; 0A
 KEYRING TCPRING
                                    ; It can be the name of an hfs
                                    ; file (name starts with /) or
                                    ; a resource name in the security
                                    ; product (e.g., RACF)
 TLSTIMEOUT       100               ; Maximum time limit between full
                                    ; TLS handshakes to protect data
                                    ; connections
```

1 The only thing we need to change in the FTP server's FTP.DATA is the SECURE_LOGIN option to REQUIRED or VERIFY_USER

Testing WS FTP Pro with client authentication



WS FTP Pro doesn't have any displays that show which certificates are in use. The only way I've found to verify that client certiificate was requested by the server is by enabling FTP server debug mode and searching for the following debug lines in SyslogD:

```
FR1799 ftpAuth: entered
FR1842 ftpAuth: environment_open()
FR1932 ftpAuth: connect as a server requesting client certificate
FR1975 ftpAuth: environment_init()
FR1984 ftpAuth: environment initialization complete
```

Configure z/OS FTP client FTP.DATA for client authentication

```
SECURE_MECHANISM  TLS              ; Name of the security mechanism
                                   ; that the client uses when it
                                   ; sends an AUTH command to the
                                   ; server.
                                   ; GSSAPI = Kerberos support
                                   ; TLS    = TLS
SECURE_FTP        REQUIRED         ; Authentication indicator
                                   ; ALLOWED        (D)
                                   ; REQUIRED
SECURE_CTRLCONN   PRIVATE          ; Minimum level of security for
                                   ; the control connection
                                   ; CLEAR          (D)
                                   ; SAFE
                                   ; PRIVATE
SECURE_DATACONN   PRIVATE          ; Minimum level of security for
                                   ; the data connection
                                   ; NEVER
                                   ; CLEAR          (D)
                                   ; SAFE
                                   ; PRIVATE
;SECURE_PBSZ      16384            ; Kerberos maximum size of the
                                   ; encoded data blocks
                                   ; Default value is 16384
                                   ; Valid range is 512 through 32768
```

```
;CIPHERSUITE       SSL_NULL_MD5      ; 01
;CIPHERSUITE       SSL_NULL_SHA      ; 02
CIPHERSUITE       SSL_RC4_MD5_EX    ; 03
CIPHERSUITE       SSL_RC4_MD5       ; 04
CIPHERSUITE       SSL_RC4_SHA       ; 05
CIPHERSUITE       SSL_RC2_MD5_EX    ; 06
CIPHERSUITE       SSL_DES_SHA       ; 09
CIPHERSUITE       SSL_3DES_SHA      ; 0A
KEYRING USER1RING                   ; Name of the keyring for TLS
                                    ; It can be the name of an hfs
                                    ; file (name starts with /) or
                                    ; a resource name in the security
                                    ; product (e.g., RACF)
TLSTIMEOUT        100               ; Maximum time limit between full
                                    ; TLS handshakes to protect data
                                    ; connections
                                    ; Default value is 100 seconds.
                                    ; Valid range is 0 through 86400
```

- The client FTP user's FTP.DATA must be configured to point to that user's keyring - the keyring in which that user's personal certificate is defined as the default certificate.
- Also in that keyring is the certificate issuer's root certificate:

```
Ring:
     >USER1RING<
Certificate Label Name              Cert Owner     USAGE       DEFAULT
-------------------------------     ------------   --------    -------
ABC CA                              CERTAUTH       CERTAUTH      NO
USER1 CERT                          ID(USER1)      PERSONAL      YES
```

Login to z/OS V1R5 FTP server without a password

```
SECURE_LOGIN      VERIFY_USER       ; Authorization level indicator
                                    ; NO_CLIENT_AUTH (D)
                                    ; REQUIRED
                                    ; VERIFY_USER
SECURE_PASSWORD   OPTIONAL          ; W. clientuath is PW required?
                                    ; OPTIONAL
                                    ; REQUIRED (D)
```

Verify User is required.

New SECURE_PASSWORD option instructs if a password is required or not.

If user verification based on certificate doesn't succeed, user will be prompted for a password.

```
//ALFREDA JOB 1,ALFRED,CLASS=A,MSGCLASS=X,NOTIFY=USER1
//*
//* test of client authentication
//*
//FTP   EXEC PGM=FTP,PARM='-a TLS'
//SYSTCPD   DD DSN=USER1.TCPCS.TCPPARMS(TCPDATA),DISP=SHR
//SYSFTPD   DD DSN=USER1.TCPCS.TCPPARMS(FTPUSER1),DISP=SHR
//SYSPRINT  DD SYSOUT=*
//INPUT     DD *
;
; Test of client authentication
;
mvs098.tcp.raleigh.ibm.com 2021 (exit
user1
cd 'user1.alfred.cntl'
dir
quit
//OUTPUT    DD SYSOUT=*
```

No password in batch FTP input stream - only the user ID.

No prompt for a password from the server.

```
220-FTPSEC1 IBM FTP CS V1R5 at MVS098.tcp.raleigh.ibm.com, 11:35:39 on 2004-02-04.
220-*
220-* Welcome to the FTP server on MVS098.tcp.raleigh.ibm.com
220-* This system is used by Alfred for testing purposes.
220-* Any issues should be reported to alfredch@us.ibm.com
220-* Your host name is mvs098cs6.tcp.raleigh.ibm.com
220-*
220 Connection will not timeout.
EZA1701I >>> AUTH TLS
234 Security environment established - ready for negotiation
EZA2895I Authentication negotiation succeeded
EZA1701I >>> PBSZ 0
200 Protection buffer size accepted
EZA1701I >>> PROT P
200 Data connection protection set to private
EZA2906I Data connection protection is private
EZA1459I NAME (mvs098.tcp.raleigh.ibm.com:USER1):
EZA1701I >>> USER user1
230-*
230-* USER1 - welcome to the FTP server on MVS098.tcp.raleigh.ibm.com
230-* Login time and date is Wed Feb  4 11:35:41 2004
230-* The current working directory is /u/user1
230-*
230-User USER1 is an authorized user
230 USER1 is logged on.  Working directory is "/u/user1".
EZA1460I Command:
EZA1736I cd 'user1.alfred.cntl'
```

FTP server activity log example

```
EZYFS50I ID=FTPSEC100001 CONN   starts Client IPaddr=::ffff:9.49.159.77 hostname=sig-9-49-159-
77.mts.ibm.com
EZYFS54I ID=FTPSEC100001 SECURE OK     Mechanism=TLS-P
EZYFS56I ID=FTPSEC100001 ACCESS OK     USERID=USER1
EZYFS67I ID=FTPSEC100001 ALLOC  OK     Use HFS filename=/u/user1/testftp/filed2.txt
EZYFS77I ID=FTPSEC100001 DEALL  OK     Release HFS filename=/u/user1/testftp/filed2.txt
EZYFS82I ID=FTPSEC100001 TRANS  HFS filename=/u/user1/testftp/filed2.txt
EZYFS84I ID=FTPSEC100001 TRANS  Stru=F Mode=S Type=A Output=15 bytes
EZYFS80I ID=FTPSEC100001 TRANS  Reply=250 Transfer completed successfully.
EZYFS52I ID=FTPSEC100001 CONN   ends   Input=0 bytes Output=1911 bytes
```

The activity log will include a SECURE entry that indicates the session is secured using TLS.

Trademarks, Copyrights, and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	CICS	IMS	MQSeries	Tivoli
IBM(logo)	Cloudscape	Informix	OS/390	WebSphere
e(logo)business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.