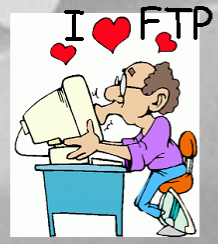
# Safe and Secure Transfers with z/OS FTP




Alfred B Christensen - alfredch@us.ibm.com
IBM Software Group, Enterprise Networking Solutions, Raleigh

IBM Systems

© 2008 IBM Corporation

# Trademarks and notices

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

- Advanced Peer-to-Peer Networking®
- AIX®
- alphaWorks®
- AnyNet®
- AS/400®
- BladeCenter®
- Candle®
- CICS®
- DB2 Connect
- DB2®
- DRDA®
- e-business on demand®
- e-business (logo)
- e business(logo)®
- ESCON®
- FICON®
- GDDM®
- HiperSockets
- HPR Channel Connectivity
- HyperSwap
- i5/OS (logo)
- i5/OS®
- IBM (logo)®
- IBM®
- IMS
- IP PrintWay
- IPDS
- iSeries
- LANDP®
- Language Environment®
- MQSeries®
- MVS
- NetView®
- OMEGAMON®
- Open Power
- OpenPower
- Operating System/2®
- Operating System/400®
- OS/2®
- OS/390®
- OS/400®
- Parallel Sysplex®
- PR/SM
- pSeries®
- RACF®
- Rational Suite®
- Rational®
- Redbooks
- Redbooks (logo)
- Sysplex Timer®
- System i5
- System p5
- System x
- System z
- System z9
- Tivoli (logo)®
- Tivoli®
- VTAM®
- WebSphere®
- xSeries®
- z9
- zSeries®
- z/Architecture
- z/OS®
- z/VM®
- z/VSE

- 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 trademarks of Microsoft Corporation in the United States, other countries, or both.
- Intel, Intel Inside (logos), MMX and Pentium 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 trademark of Linus Torvalds in the United States, other countries, or both.
- Red Hat is a trademark of Red Hat, Inc.
- SUSE® LINUX Professional 9.2 from Novell®
- Other company, product, or service names may be trademarks or service marks of others.
- This information is for planning purposes only. The information herein is subject to change before the products described become generally available.
- All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

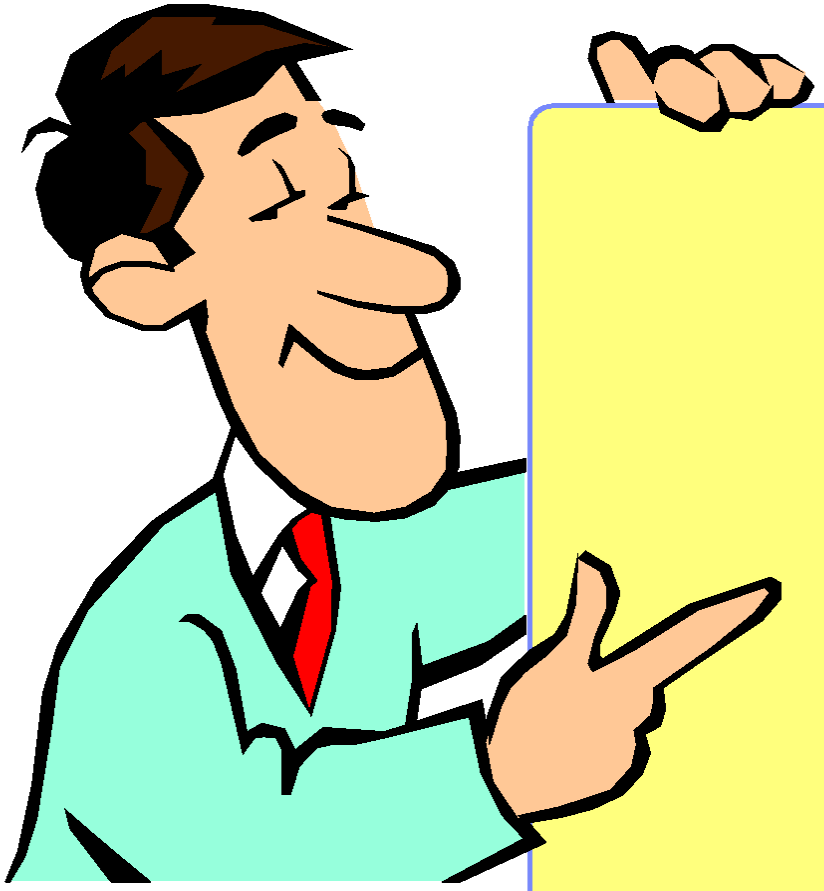
All performance data contained in this publication was obtained in the specific operating environment and under the conditions described and is presented as an illustration. Performance obtained in other operating environments may vary and customers should conduct their own testing.

Refer to www.ibm.com/legal/us for further legal information.



© 2008 IBM Corporation

# Agenda



1. Base FTP protocol review
2. Secure FTP and network traversal challenges and solutions
3. Secure FTP session setup
4. A few words about certificates
5. z/OS FTP Tidbits
6. Hot off the press: FTP in z/OS V1R9 and V1R10


© 2008 IBM Corporation

# Base FTP protocol review


© 2008 IBM Corporation

# Let's clear a little confusion from start

- **FTP or RFC959 FTP or "normal" FTP:**
  - The FTP protocol we all know and have used for years
  - What the z/OS CS FTP client and server support
    - An RFC959 FTP client talks to an RFC959 FTP server, and not to an sftp server
  - Sometimes referred to as normal FTP
    - Mostly because its been around for ever and much longer than any of its alternatives

- **sftp:**
  - Secure Shell file transfer protocol
    - A sub-protocol of SSH (Secure Shell)
    - Supported on z/OS by "IBM Ported tools for z/OS"
    - Has nothing to do with RFC959 FTP - incompatible protocols
    - An sftp client talks to an sftp server and not an RFC959 FTP server

- **ftps or ftp auth-tls or ftp auth-ssl:**
  - Secure RFC959 FTP using a standard security mechanism, such as Kerberos or SSL/TLS
  - The normal FTP protocol but with full network security (authentication and confidentiality)

- **Is normal FTP today a secure technology?**
  - The RFC959 FTP protocol has been extended numerous times since the original RFC 959 was issued in 1985
    - Specific support for both Kerberos-based and SSL/TLS-based security has been added
      - RFC4217 "Securing FTP with TLS"
  - The FTP protocol today includes numerous security features that, when enabled and implemented correctly, do make FTP a secure file transfer technology

**FTP is as secure as you set it up to be**


© 2008 IBM Corporation

# A little FTP history lesson

- **The FTP protocol is a convenient protocol to use for moving data between a variety of hardware and software platforms**
  - FTP has been around since early 1970 - earliest attempts at an FTP RFC are from 1971
  - There are FTP clients and servers for literally every single operating system environment available to us
    - There are thousands of products that implement elements of or the full FTP protocol standard
      - FTP client functions are imbedded into WEB browsers and numerous GUI tools
  - FTP is simple to use
    - Especially with one of a multitude of GUI-based FTP client tools on Windows, Linux, UNIX, etc.
- **FTP was originally designed back when networks were local and somewhat isolated**
  - FTP was designed with maximum flexibility in mind
    - different operating systems,
    - different file systems,
    - ability to control transfers between remote computers from a single control point (3-way proxy),
    - etc.
  - There was no need for authentication beyond user ID and password
  - There was no need for encryption of the data
  - The concept of "firewalls" did not exist
  - No one could imagine that IPv4 addresses one day would run out and spawn the need for something like Network Address Translation (NAT)
- **Firewalls initially addressed the peculiar behavior of the FTP protocol by implementing technologies that depended on the ability to peek into and even change the FTP protocol exchanges**
  - Because of this approach, it has been very difficult to implement security technologies that encrypt the FTP protocol exchanges end-to-end

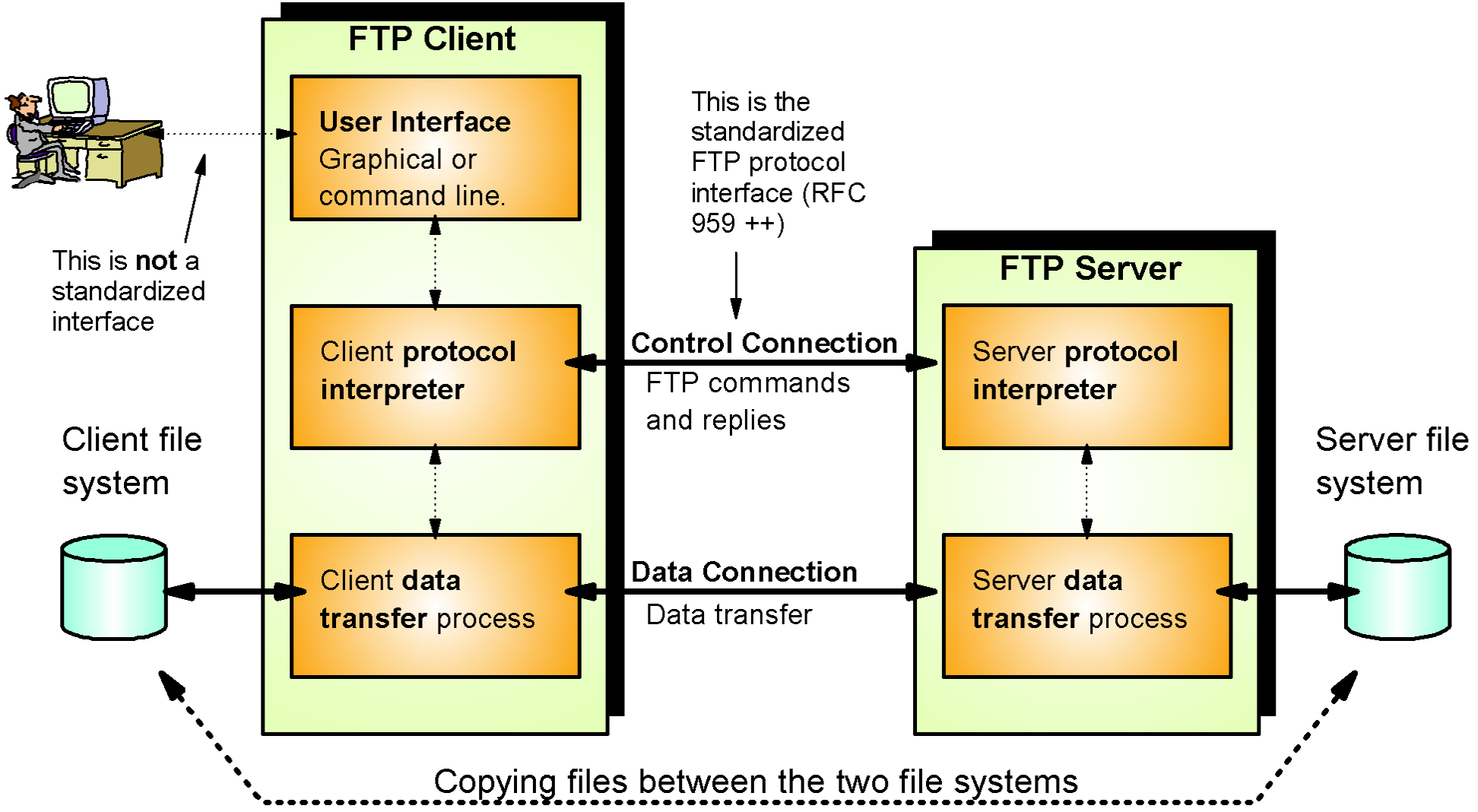

© 2008 IBM Corporation

# File Transfer Protocol (FTP) - revisited

There is both an FTP client and an FTP server on z/OS. The client can be used from TSO, the UNIX shell, batch jobs, or any z/OS program using the FTP client API (C, REXX, Java, Assembler, Cobol, PL/I, etc.)

FTP uses two separate TCP connections to transfer a file:

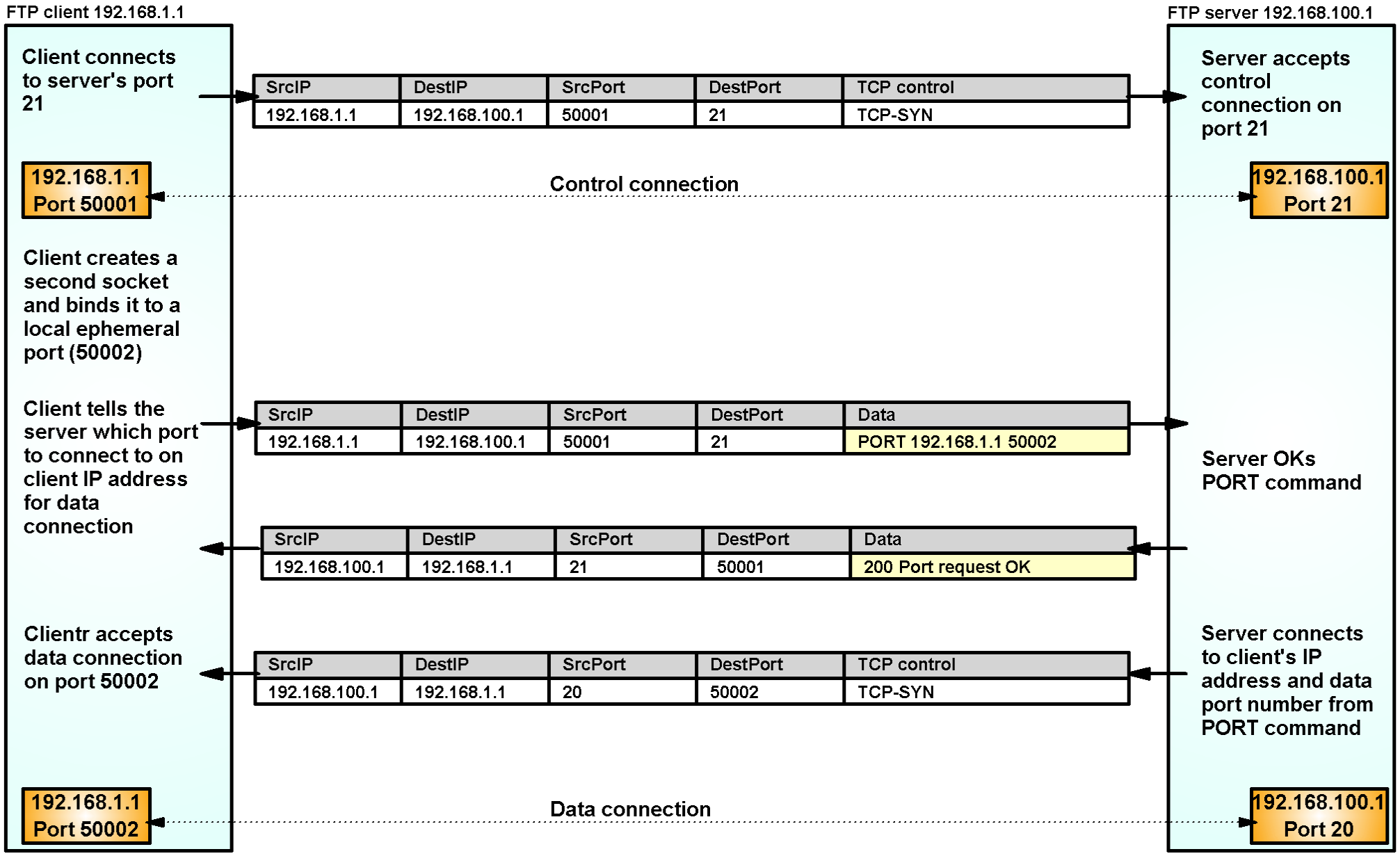
1. One is the ***control connection*** that is used for exchange of FTP commands and replies. Standard FTP server port is 21. This connection stays up during the whole FTP session.
2. The other is the ***data connection*** that is used for the actual data transfer. Stays up for the duration of a single file transfer.





© 2008 IBM Corporation

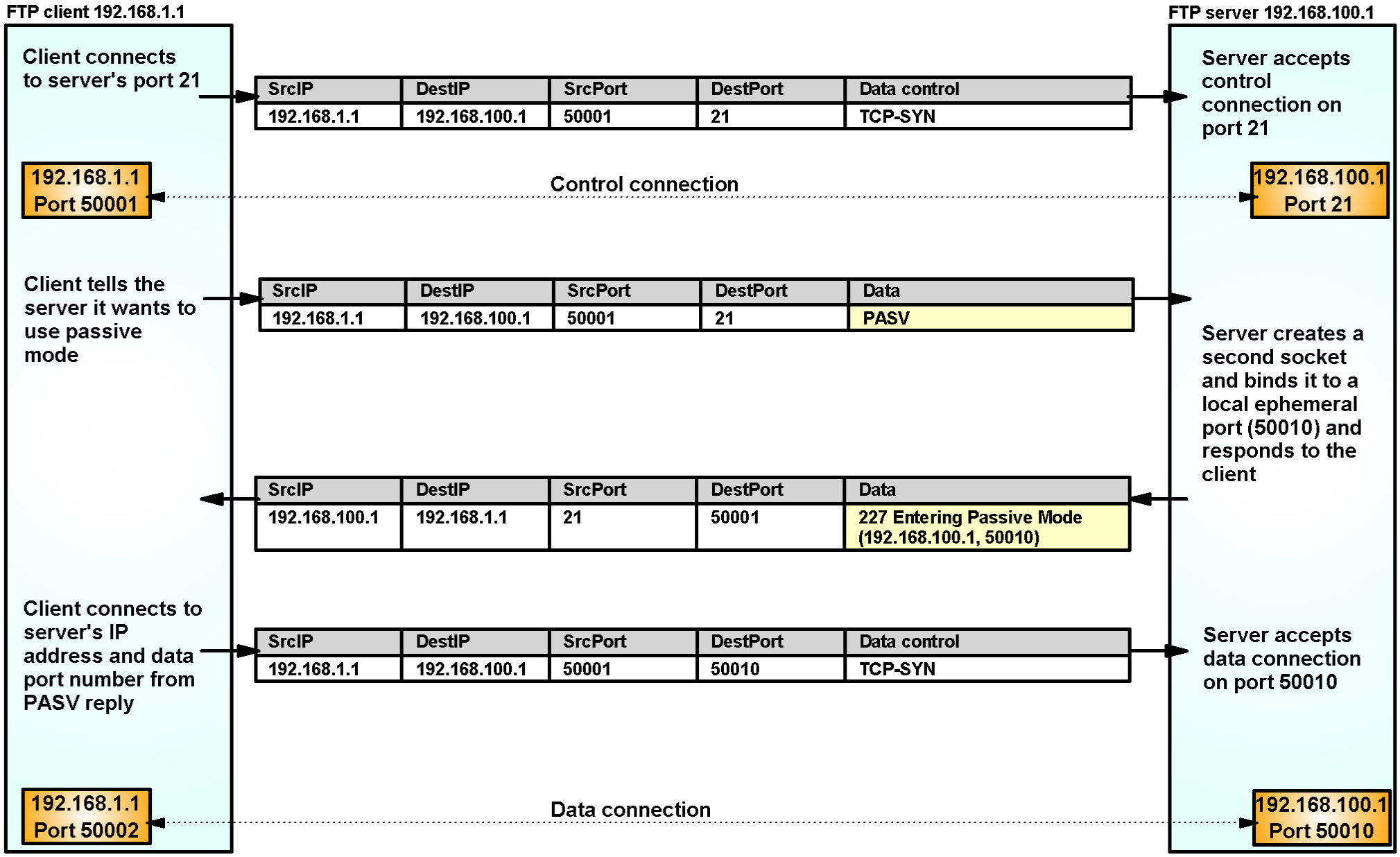
# Active mode data connection setup - revisited





© 2008 IBM Corporation

# Passive mode data connection setup - revisited





© 2008 IBM Corporation

# Secure FTP and network traversal challenges and solutions


© 2008 IBM Corporation

# So what's the problem?

*I am a firewall who wants to inspect the FTP control connection data !*

**No encryption:**

| SrcIP | DestIP | SrcPort | DestPort | Data |
|---|---|---|---|---|
| 192.168.100.1 | 192.168.1.1 | 21 | 50001 | 227 Entering Passive Mode (192.168.100.1, 50010) |

**SSL/TLS encryption:**

| SrcIP | DestIP | SrcPort | DestPort | Data |
|---|---|---|---|---|
| 192.168.100.1 | 192.168.1.1 | 21 | 50001 | @%$#*&&^^!:"J)*GVM>< |

**IPSec encryption:**

| SrcIP | DestIP | SrcPort | DestPort | Data |
|---|---|---|---|---|
| 192.168.100.1 | 192.168.1.1 | >::" | *&hU$$$$ | @%$#dd*&&^s^!:"J)*bGVM>(* hhgvvv< |

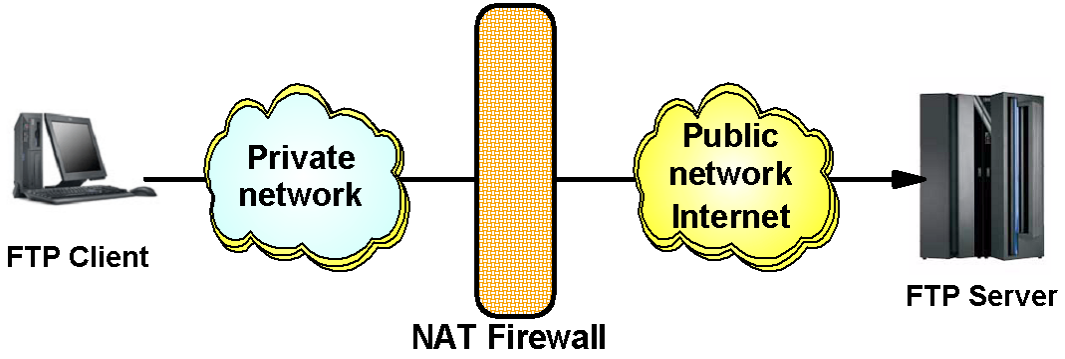
**IP header encryption varies based on transport/tunnel mode, and AH/ESP protocol**

- **In an internal/secure network without firewalls and network address translation devices - there are no problems!**
  - Both secure and non-secure FTP transfers will work just fine
- **If the FTP control connection is insecure (not encrypted or authenticated), firewalls and NAT devices manage by "peeking" into the control connection data flows - and for NAT also by modifying the control connection data flows**
  - Firewalls enable dynamic IP filters based on the IP addresses in the PORT command or the PASV reply
  - NAT devices modify the IP addresses in the PORT command and the PASV reply - in addition to their normal NAT processing of the IP headers
- **If the FTP control connection is secure (encrypted and/or authenticated), the FTP data connection setup will generally fail if the control connection passes through firewalls and/or NAT devices**
  - Firewalls are not able to determine what the IP addresses in the PORT command and PASV reply are and cannot dynamically enable filter rules
  - NAT devices can't find the IP addresses they need to change and even if they did find them and changed them, the message authentication checking will fail the data when it arrives at its final destination


© 2008 IBM Corporation

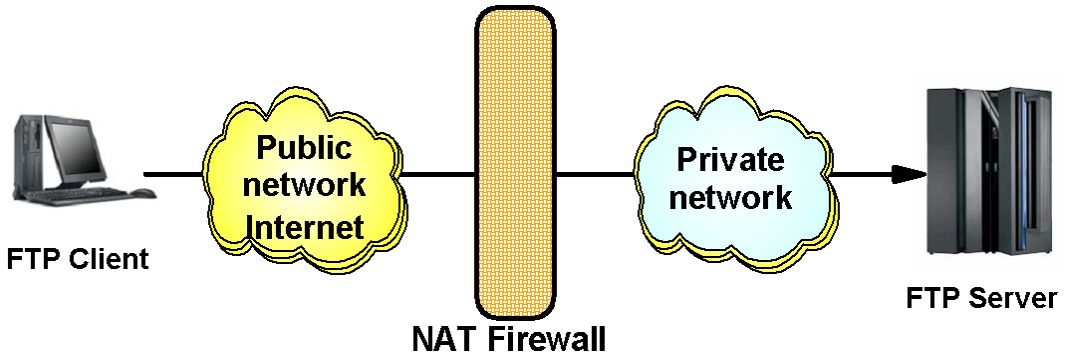
# Why does secure FTP sometimes work through a NAT firewall?

- Passive mode (PASV) may work
- Active mode (PORT) will *not* work



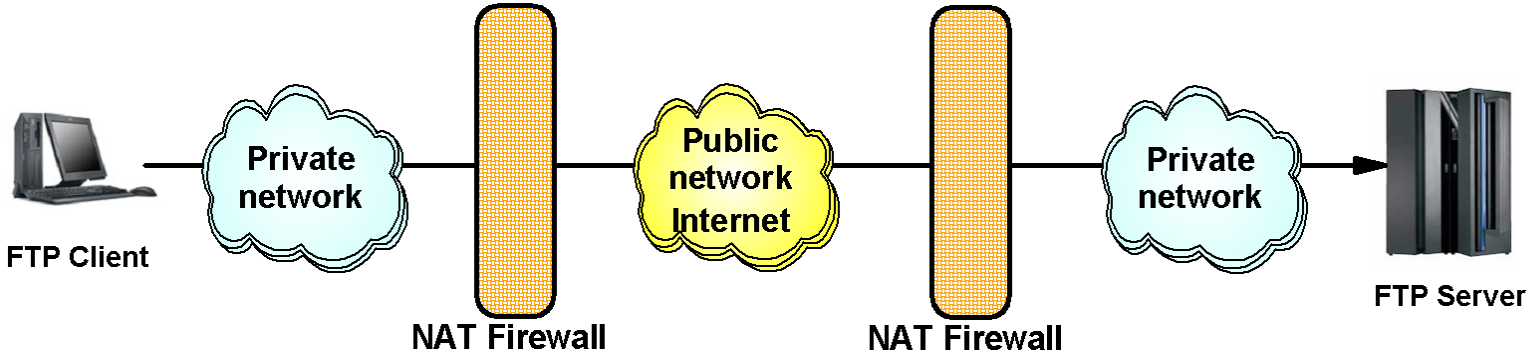

***PASV reply will return a public IP address!***
***(This is the configuration where passive mode got the name Firewall-Friendly from)***

- Passive mode (PASV) will *not* work
- Active mode (PORT) may work




***PORT command will include a public IP address!***

- Neither passive mode (PASV) nor active mode (PORT) will work!

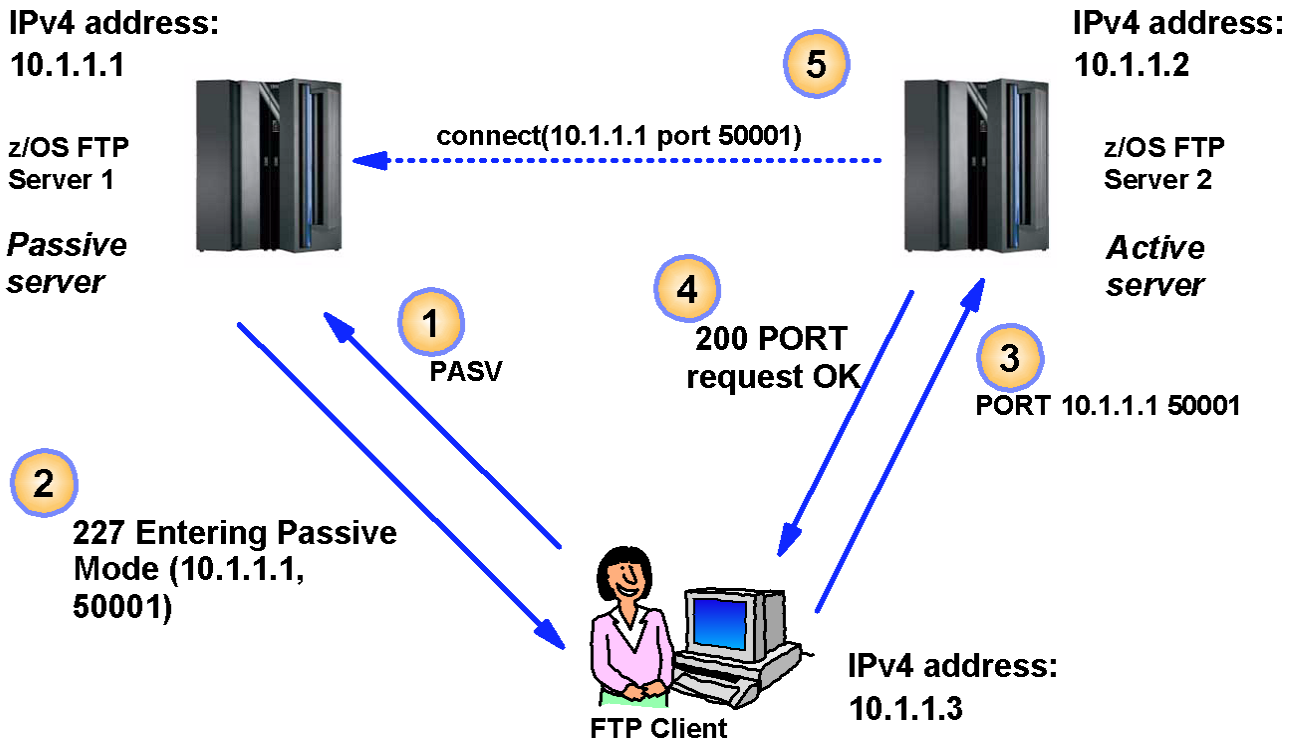




© 2008 IBM Corporation

# What are those IP addresses there for in the first place?

- **Why do we need an IP address in the PASV reply or on the PORT command?**
  - In 99.999% of FTP operations, we don't need it!
  - We only need it if we're engaging into so-called three-way FTP proxy operations
    - Once considered a very important capability of the founding FTP fathers
- **Most z/OS installations do not like three-way proxy operation of their FTP servers**
  - It is generally considered an unnecessary security risk
  - The z/OS FTP server provides configuration options to disable it from being used in a three-way proxy operation
    - PASSIVEDATACONN
    - PORTCOMMANDIPADDR




**PASSIVEDATACONN** says what this server is to do when it receives a data connection setup request from a source IP address that isn't the same as the FTP client IP address

**PORTCOMMANDIPADDR** says what this server is to do when it receives a PORT command with an IP address that isn't the same as the FTP client IP address.

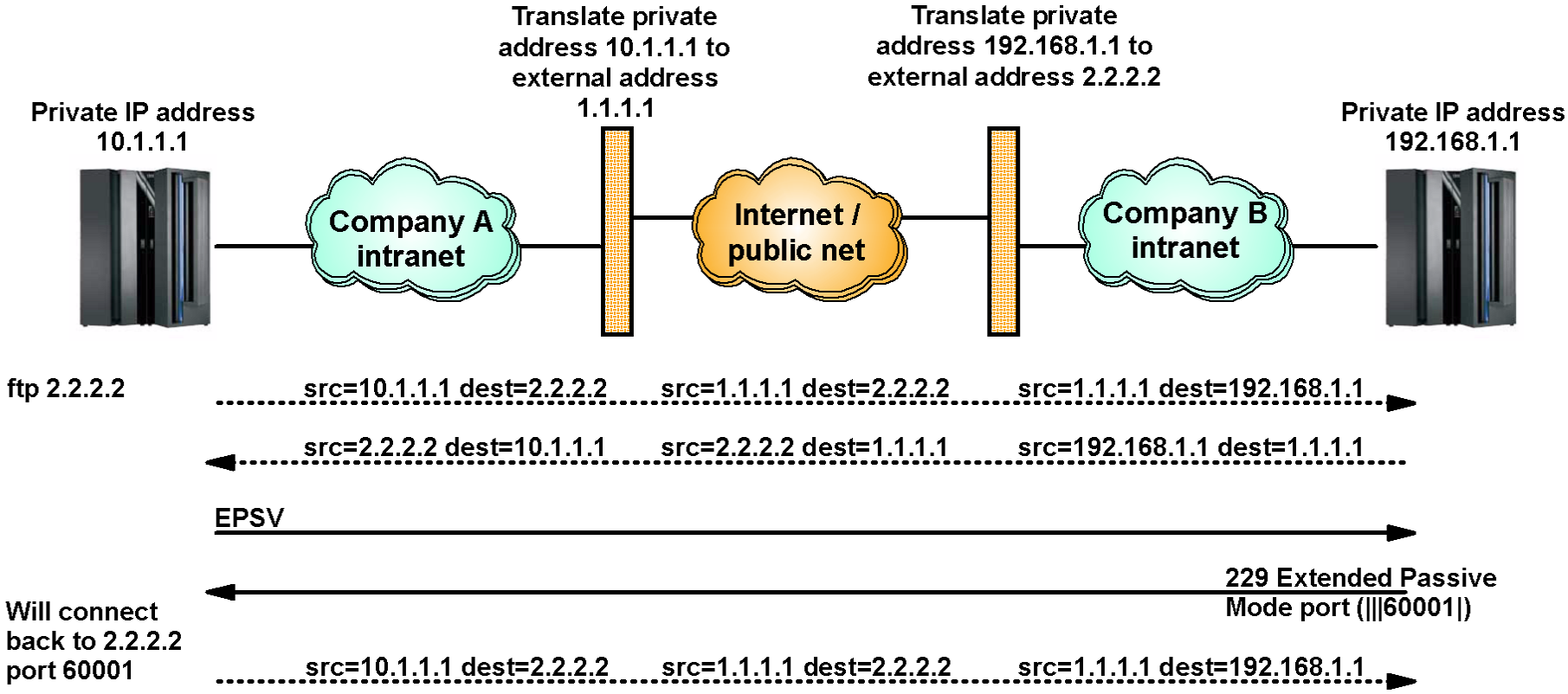

© 2008 IBM Corporation

# Extended passive mode gets rid of the IP address

- **In almost all FTP operations, the control connection and the data connection are between the same two IP addresses**

RFC2428 FTP Extensions for IPv6 and NATs

- **The extended passive mode FTP operation takes advantage of this**
  - The EPSV reply does not include an IP address, but only a port number
    - The FTP client will connect to the same IP address it used for the control connection
  - The EPSV and the accompanying extended port commend (EPRT) are also used to enable IPv6 support in FTP
    - EPSV and EPRT can be used with both IPv4 and IPv6
    - Used with IPv4, the EPSV command provides NAT firewall relief





© 2008 IBM Corporation

# Does EPSV solve all NAT firewall problems for FTP?

- **Not all FTP clients and servers support extended passive mode**
  - Many vendors mis-interpreted RFC 2428 and thought it was only needed if they wanted to implement IPv6 support for FTP
- **EPSV doesn't do anything for the port number**
  - If firewalls also implement static port-based filters, how would they know which port numbers to permit for FTP data connections?
    - Firewall administrators do generally not like adding a permit rule that allows connections to an IP address and any port number !

Connections to any port number except 21


  - If firewalls implement dynamic port filters, how would they know which port number is about to be used for a data connection?
    - The firewall cannot learn what the port number is - it is enrypted

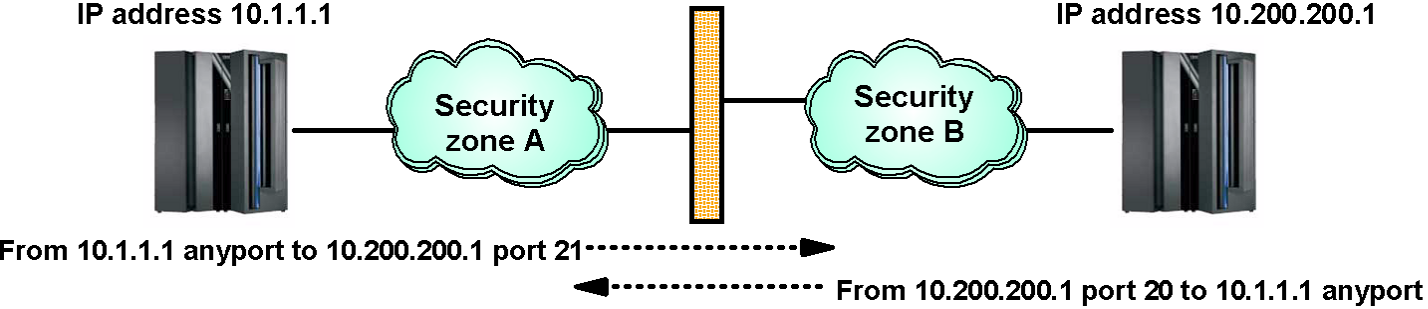
Connections to any port number except 21 or a port seen in a recent PASV reply



© 2008 IBM Corporation

# How to deal with static filters in a firewall

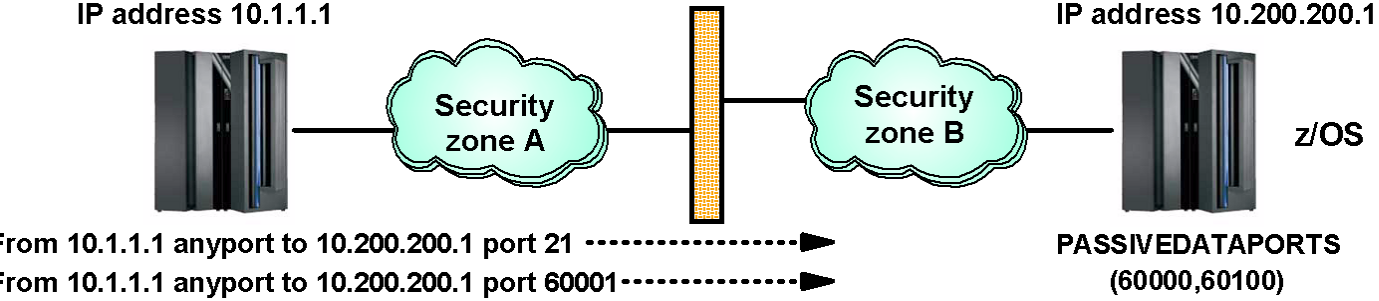
- **If you are able to use active mode FTP, the firewall filters can sometimes be managed:**
  - The control connection is permitted inbound to port 21
  - The data connection is permitted outbound from port 20
  - Will work for both standard active mode (PORT) and extended active mode (EPRT)




**Static firewall filters**

- Connection setup from 10.1.1.1 any port to port 21 on 10.200.200.1 - permit
- Connection setup from 10.200.200.1 port 20 to 10.1.1.1 any port - permit

- **If you use passive mode FTP, and your server is a z/OS FTP server, you can predefine a range of port numbers to be used for passive mode data connections**
  - The control connection is permitted inbound to port 21
  - The data connection is permitted inbound to a port in a pre-defined range
  - Will work for both standard passive mode (PASV) and extended passive mode (EPSV)




**Static firewall filters**

- Connection setup from 10.1.1.1 any port to port 21 on 10.200.200.1 - permit
- Connection setup from 10.1.1.1 any port to a port in the range from 60000 to 60100 on 10.200.200.1 - permit


© 2008 IBM Corporation

# How to deal with dynamic filters in a firewall

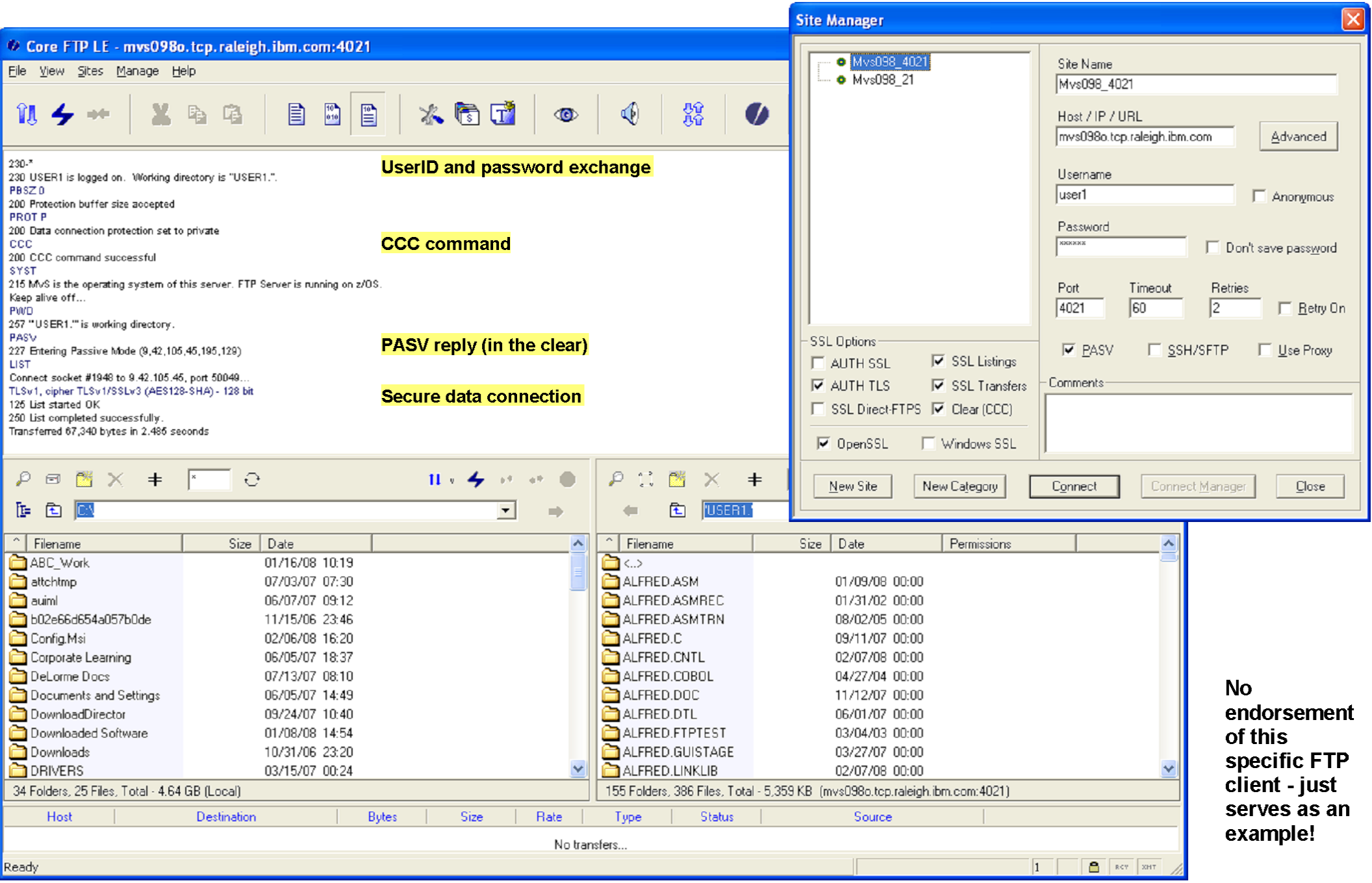
- **When using dynamic filters, the firewall enables (permits) ports based on IP address and/or port number information in the PORT/EPRT command or the PASV/EPSV reply**
  - The original FTP SSL/TLS draft RFC stated that the FTP control connection always had to be encrypted !
  - The final RFC (RFC 4217 "Securing FTP with TLS") relaxes on this requirement and implements a new Clear Command Channel (CCC) FTP command


- **Both the FTP client and server need to support the CCC command according to RFC 4217**
  - Not all FTP clients and servers that support FTP SSL/TLS support the CCC command
    - z/OS added full support for the CCC command in z/OS V1R9 (both z/OS FTP client and server)
      - APAR PK26746 supplied this function for the z/OS FTP client in fall 2006 (back to z/OS V1R4)
  - For those products that claim support, some interoperability issues have been observed !
    - z/OS FTP client works with z/OS FTP server (big surprise !!)
    - CoreFTP client for Windows works fine with z/OS FTP server
    - I have personally had problems with WS_FTP Pro's implementation of CCC
  - So test with your non-z/OS FTP clients carefully before proceeding

- **In general, the CCC command is a solution that solves SSL/TLS-enabled FTP issues with both NAT firewalls and filtering firewalls**


© 2008 IBM Corporation

# CoreFTP client to z/OS FTP server using CCC

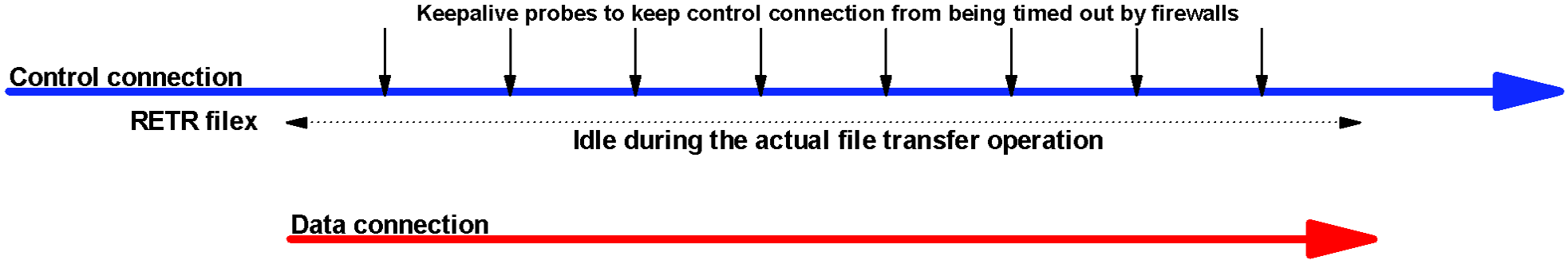



**No endorsement of this specific FTP client - just serves as an example!**


© 2008 IBM Corporation

# A couple of other pesky firewall behaviors - 1/2

- **You start an FTP session with a remote FTP server over a slow network**
  - The control connection comes up
- **You then start transferring a very large file**
  - The data connection starts chugging along
  - Nothing happens on the control connection until the data transfer ends
- **Some firewalls do not like idle TCP connections**
  - They close them down after a period of inactivity
  - If that happens to the control connection, you have no idea if your data was transferred or not
- **Enable the FTPKEEPALIVE option in both your FTP server and FTP client FTP.DATA**
  - `FTPKEEPALIVE 120    ; Keepalive packets every 2 minutes`




**Note**: z/OS V1R10 CS adds a DATAKEEPALIVE option to control keepalive probes on the data connection


© 2008 IBM Corporation

# A couple of other pesky firewall behaviors - 2/2

- **Some firewalls check to see if each message on the control connection ends in an ASCII NL**
  - Normally they do - all FTP commands and replies will end in an ASCII NL
  - If the control connection is encrypted - each messages does not end in an ASCII NL
  - And the firewall shuts the connection down
- **There isn't anything you can do on the FTP client/server end points to resolve this**
  - This has to be resolved on the firewalls by disabling that (useless) check
- **If you have a Checkpoint firewall, make sure you have turned off the FTP enforce new line setting in the firewall's configuration.**
  - Here are instructions on how to do this:

1. Edit the $FWDIR/lib/base.def
2. Change (comment out) the following line:

```
#define FTP_ENFORCE_NL
    to:
//#define FTP_ENFORCE_NL
```

3. Re-install the security policy


© 2008 IBM Corporation

# Some APARs to check out if you're not on the latest z/OS release

- **PK26746**
  - Description:
    - The FTP client should allow the CCC command to flow to clear the control connection after the initial andshake. The data connection can remain private.
  - PTFs available for z/OS V1R4, V1R5, V1R6, V1R7, and V1R8 - rolled into V1R9

- **PQ98005**
  - Description:
    - The FTP client implements a new FTP.DATA statement SECURE_HOSTNAME (REQUIRED or OPTIONAL) that will allow the client to verify the hostname in the server's digital certificate as per RFC 2818. The hostname that the client is connecting to will be verified against the server's certificate. Either the common name or the subject alternate name contained in the Server's X.509 certificate will be used to validate the hostname. If the verification fails, the connection is terminated.
  - PTFs available for z/OS V1R4, V1R5, V1R6, and V1R7 - rolled into V1R8

- **PQ89672 and PK15174**
  - Description:
    - The FTP client sends AUTH TLS, but some older servers require an AUTH SSL instead. The FTP client has been enhanced to support a new option on the SECURE_MECHANISM FTP.DATA statment for the client. The client will now support SECURE_MECHANISM SSL as well as SECURE_MECHANISM TLS and SECURE_MECHANISM GSSAPI.
  - PTFs available for z/OS V1R5 and V1R6 (PQ89672), and V1R7 (PK15174) - rolled into V1R8

- **APAR PQ87711**
  - Description:
    - The FTP client cannot connect with some servers which use the implicit connection by initiating a session on the TLSPORT (normally port 990).
  - PTFs available for z/OS V1R4, V1R5, and V1R6 - rolled into V1R7


© 2008 IBM Corporation

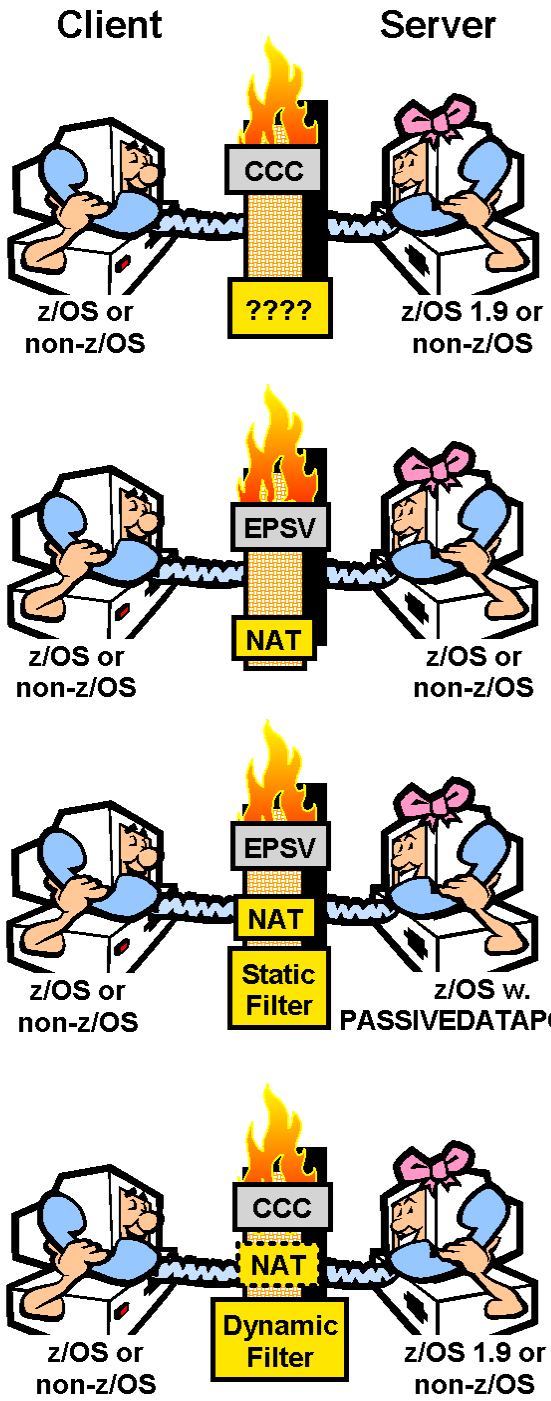
# A few simple steps to make secure FTP work through firewalls

1. **If you don't know what your firewalls do and your z/OS system is at a V1R9 level and your partner secure FTP product supports CCC (and is compatible with RFC 4217)**
   - Use the CCC command method - with z/OS as the client or as the server

2. **If your firewalls only do NAT (no filtering) and your partner secure FTP product supports extended passive mode (EPSV)**
   - Use extended passive mode - with z/OS as the client or as the server

3. **If your firewalls do NAT and static filtering, you use z/OS as the FTP server, and your partner secure FTP client supports extended passive mode (EPSV)**
   - Use the PASSIVEDATAPORTS option on your z/OS FTP server
   - Have your firewall administrator add static filters that allow connections to the range of port numbers in PASSIVEDATAPORTS
   - Use extended passive mode from your FTP client - may be z/OS or other FTP products

4. **If your firewalls do dynamic filtering (with or without NAT)**
   - You must get to z/OS V1R9 and use the CCC command method - it is the only method that will work





© 2008 IBM Corporation

# Secure FTP session setup


© 2008 IBM Corporation

# Two ways to initiate an SSL/TLS FTP session

**There are two ways to indicate if an FTP session is to use SSL/TLS or not:**

- ***Explicit mode*** (also known as AUTH SSL or AUTH TLS mode)
  - FTP client connects to usual FTP server port 21 and sends an FTP command (AUTH) to request use of SSL/TLS
  - This mode is defined in RFC 4217
  - This is the recommended mode according to the RFC standards
  - The FTP server may have both secure and non-secure connections on the same port
- ***Implicit mode*** (also known as SSL direct or FTPS mode)
  - FTP client connects to an alternate FTP server port (for example, port 990) and the client and server implicitly enter SSL/TLS mode as a result of the connection being established to that alternate port number
  - There are no RFC standards that govern how implicit mode FTP sessions are to be set up
  - Use of implicit mode FTP is generally based on how the original Netscape implementation worked
  - z/OS FTP originally chose an alternative method

**You configure the z/OS FTP server to use implicit mode by specifying the TLSPORT option:**

- TLSPORT 0 Disable the protected port - no port is implicitly secured with TLS.
- TLSPORT port Specify the protected port - port 990 is the default value.

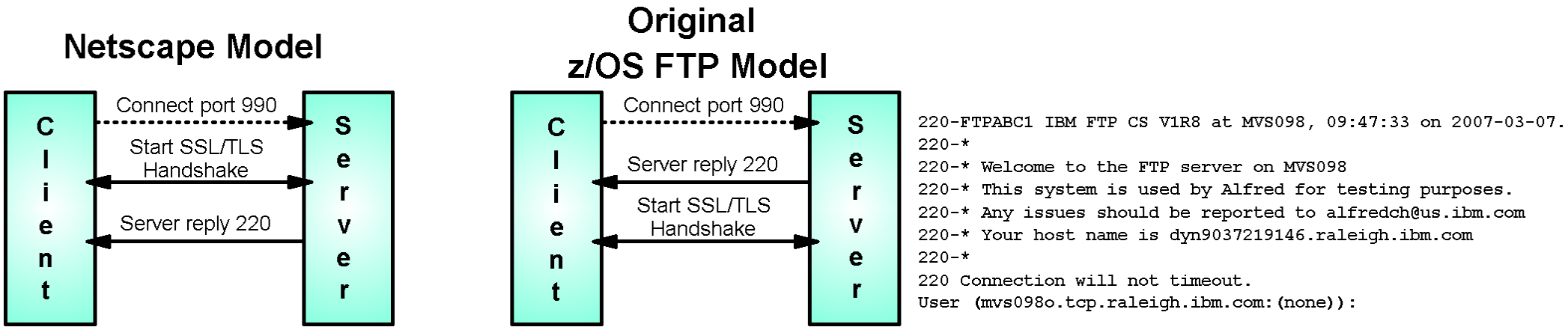
**You configure the z/OS FTP server to use explicit mode by specifying the SECURE_FTP option (along with other security-related options):**

- SECURE_FTP ALLOWED Clients are allowed to send an AUTH command
- SECURE_FTP REQUIRED Clients must send an AUTH command


© 2008 IBM Corporation

# Implicit mode connection setup

- **Since implicit mode FTP wasn't ever defined precisely in the RFC standards, two different incompatible models were used:**
  - The Netscape model
  - The original z/OS FTP model




- **The z/OS FTP client was changed to optionally work according to the Netscape model via APAR PQ87711, which was PTFed back to z/OS V1R4 and rolled into z/OS V1R7:**
  - SECUREIMPLICITZOS=TRUE — The original z/OS FTP model (until V1R10 required for z/OS FTP server)
  - SECUREIMPLICITZOS=FALSE — The Netscape model (to non-z/OS FTP servers)

- **z/OS V1R10 CS adds similar support to the z/OS FTP server**
  - Reusing the SECUREIMPLICITZOS configuration option that is already used by the z/OS FTP client


© 2008 IBM Corporation

# z/OS FTP server security options - page 1 of 4

```
;EXTENSIONS        AUTH_GSSAPI       ; Enable Kerberos authentication
                                     ; Default is disabled.

 EXTENSIONS        AUTH_TLS          ; Enable TLS authentication
                                     ; Default is disabled.

 TLSMECHANISM      ATTLS             ; Server-specific or ATTLS
                                     ; ATTLS - use ATTLS
                                     ; FTP - server-specific (D)

 SECURE_FTP        ALLOWED           ; Authentication indicator
                                     ; ALLOWED         (D)
                                     ; REQUIRED

 SECURE_LOGIN      REQUIRED          ; Authorization level indicator
                                     ; for TLS
                                     ; NO_CLIENT_AUTH (D)
                                     ; REQUIRED
                                     ; VERIFY_USER

 SECURE_PASSWORD   REQUIRED          ; REQUIRED (D) - User must enter
                                     ;     password
                                     ; OPTIONAL - User does not have to
                                     ;     enter a password
                                     ; This setting has meaning only
                                     ; for TLS when implementing client
                                     ; certificate authentication
```

**Switch between FTP's built-in SSL/TLS support and ATTLS support**

**Must all connections be secure or just those who wish to be?**

**Is client authentication required and if so, at what level?**

**If client authentication is used at level 3 and a user ID can be matched, is a password still required or not?**



© 2008 IBM Corporation

# z/OS FTP server security options - page 2 of 4

```
;SECURE_PASSWORD_KERBEROS  REQUIRED   ; REQUIRED (D) - User must enter
                                      ;     password
                                      ; OPTIONAL - User does not have to
                                      ;     enter a password
                                      ; This setting has meaning only
                                      ; for Kerberos

 SECURE_CTRLCONN    CLEAR             ; Minimum level of security for
                                      ; the control connection
                                      ; CLEAR          (D)
                                      ; SAFE
                                      ; PRIVATE

 SECURE_DATACONN    CLEAR             ; 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
```

**Server's requirement to security of the control connection. Must be set to CLEAR for the server to accept the CCC command**

**Server's requirement to security of the data connection**



© 2008 IBM Corporation

# z/OS FTP server security options - page 3 of 4

```
; 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.
;
; When using ATTLS, these are controlled via the ATTLS
; Policy
;
;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_3DES_SHA       ; 0A
 CIPHERSUITE       SSL_AES_128_SHA    ; 2F
;CIPHERSUITE       SSL_AES_256_SHA    ; 35
 CIPHERSUITE       SSL_DES_SHA        ; 09
```

**Server's required ciphersuites**



© 2008 IBM Corporation

# z/OS FTP server security options - page 4 of 4

```
; When using ATTLS, the keyring is controlled via the
; ATTLS policy
;
 KEYRING            TLSRING             ; 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)

;
; When using ATTLS, the TLS timeout value is controlled via the
; ATTLS policy
;
 TLSTIMEOUT         100                 ; Maximum time limit between full
                                        ; TLS handshakes to protect data
                                        ; connections
                                        ; Default value is 100 seconds.
                                        ; Valid range is 0 through 86400

 TLSRFCLEVEL        RFC4217             ; Specify what level of RFC 4217,
                                        ; On Securing FTP with TLS, is
                                        ; supported.
                                        ; DRAFT     (D) Internet Draft level
                                        ; RFC4217       RFC level
```

**Server's keyring - prefixed with FTPD's started task userID: userID.TLSRING**

**Is z/OS FTP server to operate at the old draft RFC level for SSL/TLS or the now existing RFC?**

**NB: default is to use draft - you may want to change that!!!!**

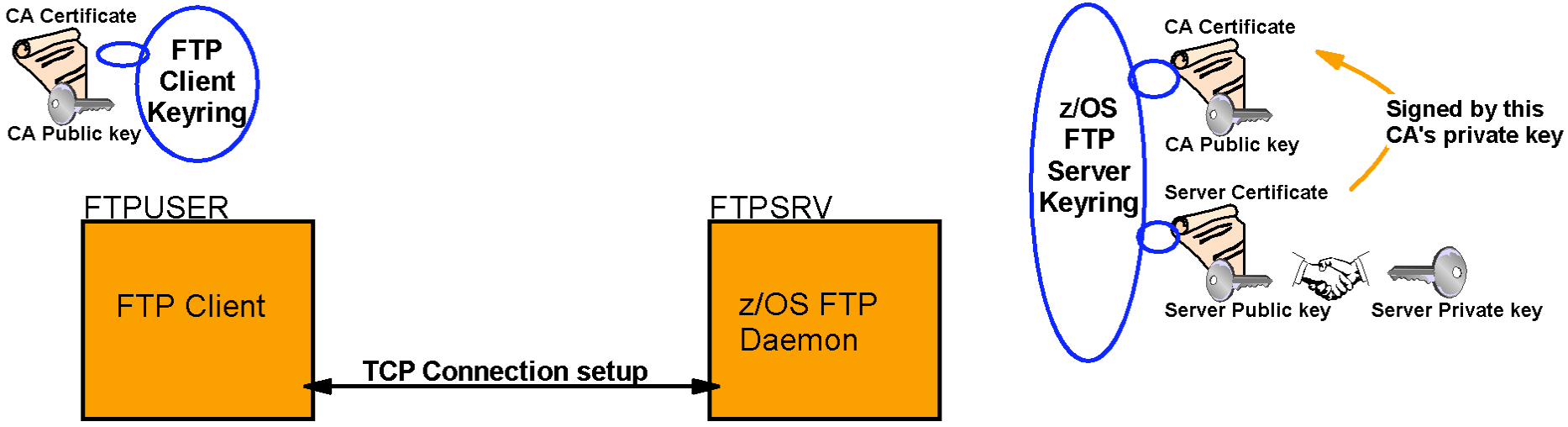


© 2008 IBM Corporation

# A few words about certificates


© 2008 IBM Corporation

# Certificates needed for z/OS FTP server authentication only




**Hello - I want to use SSL/TLS**

**Hello - OK, me too**

**And here is my server certificate**

Server Certificate

Server Public key

1. **Verify server certificate has not expired**
2. **Verify server certificate is valid using CA's public key**
3. **Do optional checks on the server certificate**
4. **Store server's public key for later use**
5. **Generate symmetric key and encrypt under server's public key**

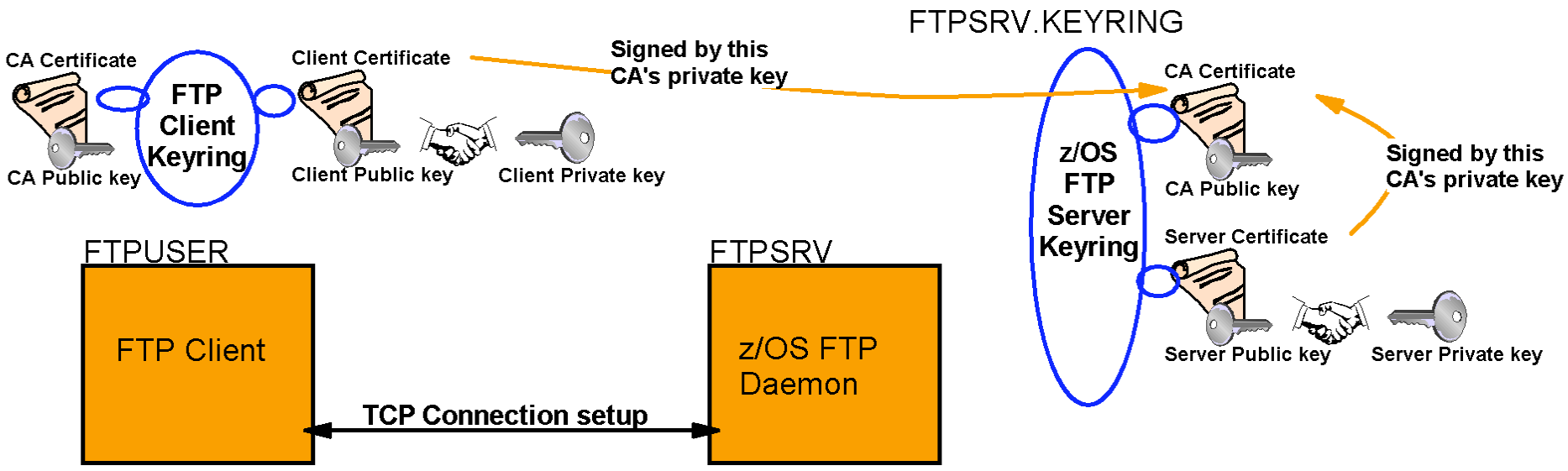
**Here is our secret symmetric key**

**encrypted under your public key**

- **CA may be an external CA, such as Verisign, or it may be an in-house CA**
  - In both cases, the CA root certificate needs to be present at both the client and the server side
- **The server certificate is signed by the CA and is stored on the server side**
  - On z/OS, this will typically be the default certificate in the FTP server's started task userID's keyring in RACF
- **During SSL handshake, the server certificate (not the server private key) is sent to the client**
  - The client verifies the certificates signature using the CA public key in its copy of the CA certificate


© 2008 IBM Corporation

# Certificates needed for z/OS FTP server and FTP client authentication




**Hello - I want to use SSL/TLS**

**Hello - OK, me too**

**Here is my server certificate, and I want to see your client certificate!**

Server Certificate

Server Public key

1. **Verify server certificate has not expired**
2. **Verify server certificate is valid using CA's public key**
3. **Do optional checks on the server certificate**
4. **Store server's public key for later use**
5. **Generate symmetric key and encrypt under server's public key**

**Here is our secret symmetric key encrypted under your public key**

**- and here is my client certificate**

Client Certificate

Client Public key

1. **Verify client certificate has not expired**
2. **Verify client certificate is valid using CA's public key**
3. **Do optional checks on the client certificate**
   1. **Does it map to a RACF user ID (authentication level 2)**
   2. **Is the user permitted to use this service (authenitcation level 3)**



© 2008 IBM Corporation

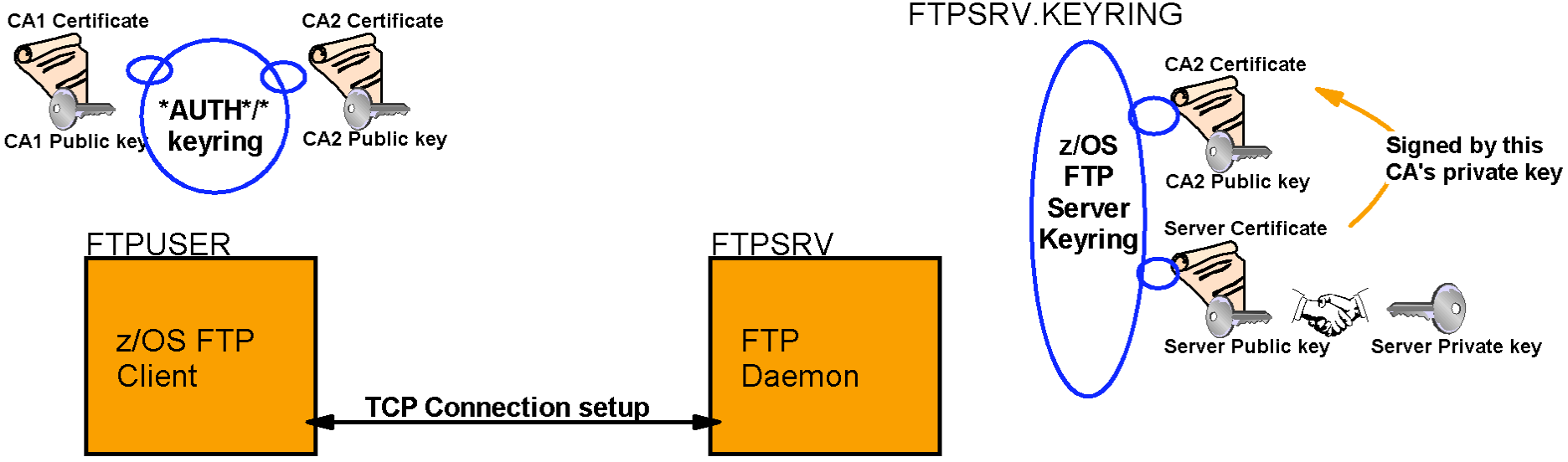
# FTP server client authentication levels

| Authentication level | FTP server SECURE_LOGIN option | Description |
|---|---|---|
| Level 1 | REQUIRED | The authenticity and validity of the client certificate is verified against the trusted roots in the FTP server's keyring. |
| Level 2 | VERIFY_USER | Same as level 1 PLUS a verification that the client certificate is registered by RACF and mapped to a known RACF user ID. |
| Level 3 | VERIFY_USER | Same as level 2 PLUS a verification that the user ID has permission to a SERVAUTH profile that represents this specific FTP server: EZB.FTP.sysname.ftpdaemonname.PORTnnnnn |


© 2008 IBM Corporation

# Virtual keyrings are very useful when z/OS is the FTP client

- **If z/OS is the FTP client, does every FTP user on z/OS have to have a key ring with a copy of the CA certificate?**
  - A few releases back, the answer was yes
    - What we call an "administratively heavy process"
  - z/OS V1R8 added support for something known as a virtual keyring
- **To have System SSL check all CERTAUTH certificates in RACF when verifying a server certificate that was received during the SSL handshake, specify a key ring in the client FTP.DATA (or matching AT-TLS definitions) as:**
  - `KEYRING *AUTH*/*`
- **If client authentication is required, the z/OS FTP user still needs his/her own keyring**





© 2008 IBM Corporation

# z/OS FTP Tidbits


© 2008 IBM Corporation

# REPLYSECURITYLEVEL

- **It is in some scenarios considered a security issue to include IP addresses, port numbers, host names, etc. in responses from servers.**
  - Too much details make life easier for someone who wants to hack your system
- **FTP normally includes such information in various FTP server replies**
- **The z/OS FTP server has an FTP.DATA option to control if the server is allowed to include IP addresses and port numbers in its replies:**
  - FTP.DATA: REPLYSECURITYLEVEL [0 / 1]
    - 0: Default. No restrictions on information included in server replies
    - 1: No IP addresses, hostnames, port numbers, or operating system level information included in replies from the server

**LEVEL 0**

```
C:\>ftp mvs098.tcp.raleigh.ibm.com
Connected to mvs098.tcp.raleigh.ibm.com.
220-FTPABC1 IBM FTP CS V1R4 at MVS098, 16:42:51 on 2002-10-24.
```

**LEVEL 1**

```
C:\>ftp mvs098.tcp.raleigh.ibm.com
Connected to mvs098.tcp.raleigh.ibm.com.
220-IBM FTP, 16:45:57 on 2002-10-24.
```


© 2008 IBM Corporation

# UNIX file system access in general can be controlled

- **Some customers (a few) have expressed a need to be able to disable users from accessing any UNIX file system files.**
- **The FTP server can be configured to perform a check against a SERVAUTH resource to control such access.**
- **SERVAUTH resource name is:**
  - **EZB.FTP.sysname.ftpdaemonname.ACCESS.HFS**
  - The profile may contain wildcards to cover more LPARs and FTP daemons:
    - EZB.FTP.*.*.ACCESS.HFS - will cover all LPARs and all FTP daemons
- **If the above resource is defined, then users must have READ access to it in order for the FTP server to allow users to access any files in the UNIX file system including directory information.**


© 2008 IBM Corporation

# A few useful, but often overlooked FTP.DATA options

- **Allow or disallow the FTP server to send detailed login failure replies to an FTP client**
  - z/OS FTP server FTP.DATA
    - `ACCESSERRORMSGS [TRUE/FALSE]`
- **Use the EPSV4 statement to direct the FTP client to use EPSV and EPRT commands on IPv4 sessions (EPSV and EPRT will always be used on IPv6 connections, but are optional on IPv4 connections)**
  - z/OS FTP client FTP.DATA
    - `EPSV4 [TRUE/FALSE]`
- **Use the FTPLOGGING statement to indicate whether the FTP server should log FTP server activity to SyslogD**
  - z/OS FTP server FTP.DATA
    - `FTPLOGGING [TRUE/TRUENODNS/FALSE]`
- **Use the PROGRESS (FTP client) statement to control the interval between progress report messages generated by the z/OS FTP client during an inbound or outbound file transfer.**
  - z/OS FTP client FTP.DATA
    - `PROGRESS nnn  ; in seconds - default (and minimum) is 10 seconds`



© 2008 IBM Corporation

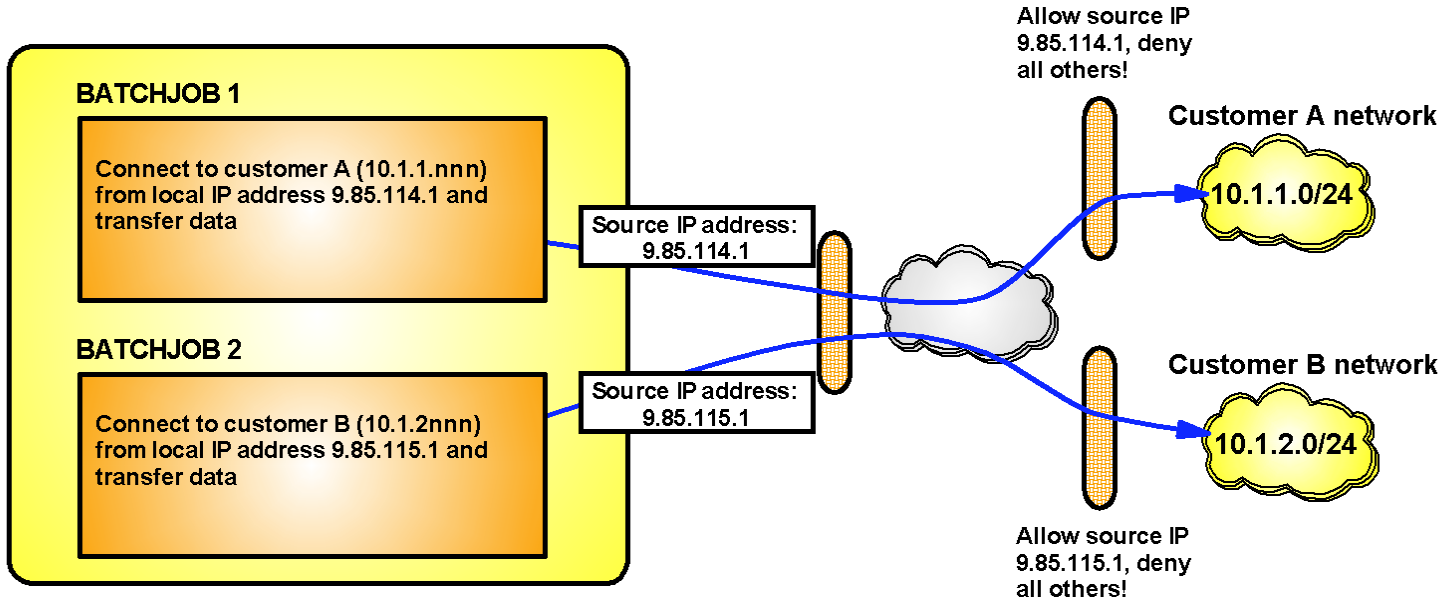
# Hot off the press:
# FTP in z/OS V1R9 and V1R10


© 2008 IBM Corporation

# FTP client allows user to specify source IP address

z/OS V1R9

- **Despite significant flexibility in how to direct TCP/IP to choose source IP address for outbound connections, there remains a need to be be able to specify which source IP address a given FTP client invocation should use when connecting to an FTP server.**
  - If batch FTP client job preparation is done by a group of people who do not have access to update (or maybe even view) the source IP address rules (SRCIP) in the TCP/IP profile, a need for them to specify a specific source IP address when preparing the batch jobs still exists.
- **The z/OS FTP client in z/OS V1R9 implements a new command line option where a user can specify which local IP address the connection to the FTP server should come from.**
  - It is the user's responsibility to verify that the chosen address is a valid local IPv4 address that is reachable from the FTP server node
- **When a source IP address is specified in the command line invocation of the FTP client, that address will override all other source IP address selection rules.**





© 2008 IBM Corporation

# FTP security - provide a SAF-based mechanism to restrict users from using the z/OS FTP server at all

z/OS V1R10

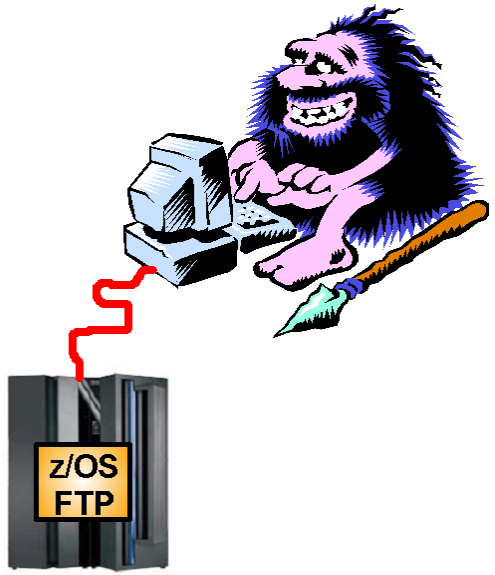
- **Basically, any user who has a valid z/OS user ID can log into and use the z/OS FTP server**

- **There are some ways to restrict access to the z/OS FTP server:**
  - Restrictions can be implemented in general by coding an FTP server password check exit routine (FTCHKPWD)
    - Many installations no longer have the skills to implement exit routines
  - If users log on using SSL/TLS and the SECURE_LOGIN option is set to VERIFY_USER, the FTP server will check if the user has READ access to EZB.FTP.<systemname>.<ftpdaemonname>.PORTxxxx SERVAUTH resource
    - If users do not use SSL/TLS or the VERIFY_USER option isn't set as above, no checking of the SERVAUTH resource is currently done

- **z/OS V1R10 CS optionally extends the check of the already existing SERVAUTH resource to all users who log into the FTP server**
  - Giving installations an easy way to limit use of the FTP server functions in general:
    - Define the SERVAUTH resource with universal access set to NONE
    - Permit those users who are allowed to use the FTP server with READ access to the SERVAUTH resource

- **A new FTP.DATA server option will be used to indicate this new behavior:**
  - VERIFYUSER=[TRUE/FALSE]

**Increased protection of the z/OS FTP server in general**





© 2008 IBM Corporation

# Application data for FTP sockets

z/OS V1R10

- **z/OS CS V1R9 implemented support for TCP applications to associate up to 40 characters of application-specific data with a TCP socket:**
  - Can be set or updated by a TCP application using an IOCTL sockets call
  - Is included in NETSTAT ALL, ALLCONN, and CONN reports and used as a filter
  - Is included in the NMI network monitor interface
  - Suggested format for the field is to use an eight-byte application identifier in the first 8 characters of the 40-character APPLDATA field

```
MVS TCP/IP NETSTAT CS V1R10        TCPIP Name: TCPC
User Id  Conn     State
-------  ----     -----
FTP40211 00000049 Listen
  Local Socket:   ::..4021
  Foreign Socket: ::..0
  Application Data:  EZAFTP0D
FTP40211 00000064 Establsh
  Local Socket:   ::ffff:9.42.105.45..4021
  Foreign Socket: ::ffff:9.37.219.155..3806
  Application Data:  EZAFTP0S C USER1      PTS32F
```

- **Selected CS components began using the application data field in z/OS V1R9**
  - Used by CICS Sockets in V1R9 to associate CICS-specific information with CICS sockets endpoints
    - EZACICSO SRV1 0000123 USER1234 CICA
  - Used by the TN3270 server in V1R9 to associate TN3270-specific information with TN3270 sockets endpoints
    - EZBTNSRV TCPABC80 TSO10001 ET B

- **FTP will start using the application data field in z/OS CS V1R10:**
  - Both the FTP client and the FTP server will associate FTP-specific information with the FTP sockets endpoints:
    - FTP component (Client, Server, Daemon)
    - Type of connection (Control or Data)
    - User ID
    - Security characteristics (SSL/TLS, GSSAPI, Ciphers, etc.)
    - Info about file being transferred (direction, type, location)

Improved NETSTAT and NMI visibility into FTP workload-specific characteristics.



© 2008 IBM Corporation

# Improvements in how FTP handles MVS data set contention

z/OS V1R10

- **z/OS FTP client and server obviously need to obey the usual MVS data set sharing rules**
  - MVS data sets in general at allocation time
  - Members of PDS and PDSE data sets based on ISPF enqueue/dequeue conventions
- **Prior to z/OS V1R10 such a transfer attempt would fail without much detail about the condition:**

```
ftp> cd 'user1.alfred.pdse'
250 The working directory "USER1.ALFRED.PDSE" is a partitioned data set
ftp> get cs4
200 Port request OK.
550 USER1.ALFRED.PDSE(CS4) used exclusively by someone else.
ftp>
```

- **If the FTP client end-user is a human being, the end-user can choose to wait a little and then retry the failed FTP operation**
- **If the FTP client end-user is an automated process running on z/OS or somewhere else, such a condition normally results in a failed automated FTP process**
- **z/OS V1R10 CS adds FTP functions that instead of reporting an error in this case, will wait and retry the operation automatically:**
  - Configurable maximum wait time (in minutes)
  - New messages and replies tell the FTP client who is using the dataset and how many retries are remaining before failure
    - If you choose to specify the new option with a zero value (no wait), the end user will still be informed of the fact that the transfer failed because of a data set contention
  - This gives the user a chance to resolve the conflict before the wait time expires, allowing the FTP attempt to succeed without having to manually retry



© 2008 IBM Corporation

# Sample transfer when data set contention occurs

z/OS V1R10

- **New server FTP.DATA option called DSWAITTIME**
  - Default is zero (as before z/OS V1R10)
    - Will still give details of data set contention if the data set cannot be accessed
  - Maximum waittime in minutes (0 to 14400)
    - FTP server will retry in one-minute intervals

```
DSWAITTIME          2                    ; Wait 2 minutes for dataset
                                         ; contention
```

**APAR PK67061 will allow you to control the amount of detail in the message**

```
get apicln
 >>> PORT 127,0,0,1,4,109
 200 Port request OK.
 >>> RETR apicln
 125-FTP Server unable to obtain SHARED use of USER1.ALFRED.PDSE(APICLN) which
 is held by: 003D USER1    EXCL on SPFEDIT
 125-Data set access will be retried in 1 minute intervals - 2 attempts remaining
 125-FTP Server unable to obtain SHARED use of USER1.ALFRED.PDSE(APICLN) which
 is held by: 003D USER1    EXCL on SPFEDIT
 125-Data set access will be retried in 1 minute intervals - 1 attempts remaining
 125 Sending data set USER1.ALFRED.PDSE(APICLN) FIXrecfm 80
 250 Transfer completed successfully.
 11345 bytes transferred in 0.010 seconds.  Transfer rate 1134.50 Kbytes/sec.
 Command:
```



© 2008 IBM Corporation

# For more information....

| URL | Content |
|---|---|
| http://www.ibm.com/systems/z/ | IBM Mainframe |
| http://www.ibm.com/systems/z/hardware/networking/index.html | IBM Mainframe Networking |
| http://www.ibm.com/software/network/commserver/ | Communications Server product overview |
| http://www.ibm.com/software/network/commserver/zos/ | z/OS Communications Server overview |
| http://www.ibm.com/software/network/commserver/z_lin/ | Communications Server for Linux on system z |
| http://www.ibm.com/software/network/ccl/ | Communication Controller for Linux on system z |
| http://www.ibm.com/software/network/commserver/library/ | Communications Server products - white papers, product documentation, etc. |
| http://www.ibm.com/systems/z/os/zos/bkserv/ | z/OS Internet library - PDF versions of z/OS manuals (including z/OS CS) |
| http://www.redbooks.ibm.com | ITSO Redbooks |
| http://www.ibm.com/software/network/commserver/support | Communications Server technical Support |
| http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/TechDocs | Technical support documentation from ATS (techdocs, flashes, presentations, white papers, etc.) |
| http://www.rfc-editor.org/rfcsearch.html | Request For Comments (RFC) |
| http://publib.boulder.ibm.com/infocenter/ieduasst/stgv1r0/index.jsp | IBM education assistant |


© 2008 IBM Corporation