

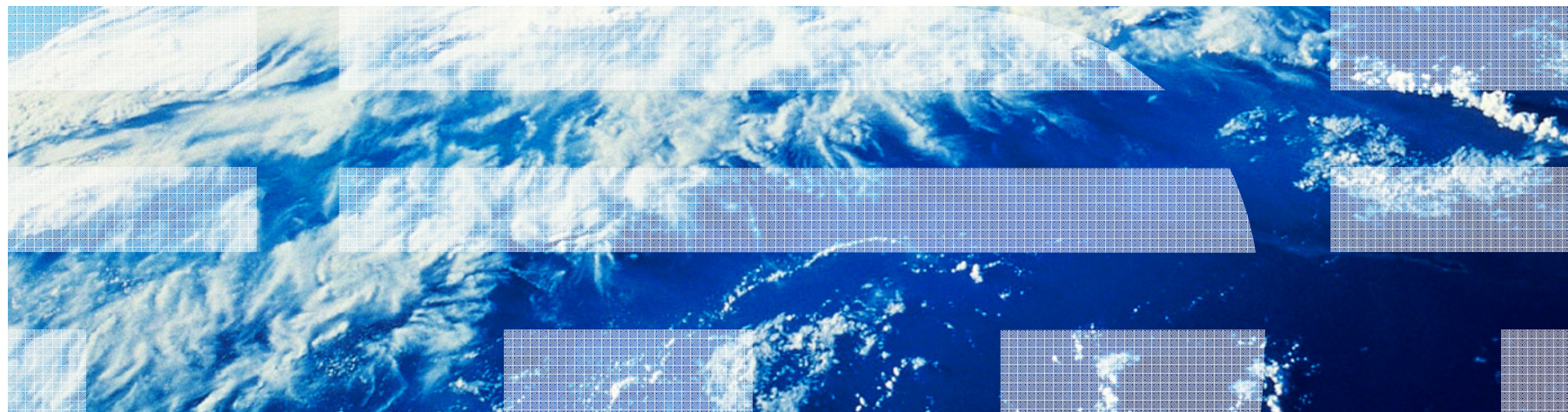


Safe and Secure Transfers with z/OS FTP

SHARE Session 12855

Lin Overby – overbylh@us.ibm.com
Sam Reynolds – samr@us.ibm.com
z/OS Communications Server

IBM Research Triangle Park, NC



February 6, 2013

Trademarks, notices, and disclaimers

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® | • GDDM® | • Language Environment® | • Rational Suite® | • zEnterprise |
| • AIX® | • GDPS® | • MQSeries® | • Rational® | • zSeries® |
| • alphaWorks® | • Geographically Dispersed Parallel Sysplex | • MVS | • Redbooks | • z/Architecture |
| • AnyNet® | • HiperSockets | • NetView® | • Redbooks (logo) | • z/OS® |
| • AS/400® | • HPR Channel Connectivity | • OMEGAMON® | • Sysplex Timer® | • z/VM® |
| • BladeCenter® | • HyperSwap | • Open Power | • System i5 | • z/VSE |
| • Candle® | • i5/OS (logo) | • OpenPower | • System p5 | |
| • CICS® | • i5/OS® | • Operating System/2® | • System x® | |
| • DataPower® | • IBM eServer | • Operating System/400® | • System z® | |
| • DB2 Connect | • IBM (logo)® | • OS/2® | • System z9® | |
| • DB2® | • IBM® | • OS/390® | • System z10 | |
| • DRDA® | • IBM zEnterprise™ System | • OS/400® | • Tivoli (logo)® | |
| • e-business on demand® | • IMS | • Parallel Sysplex® | • Tivoli® | |
| • e-business (logo) | • InfiniBand® | • POWER® | • VTAM® | |
| • e business (logo)® | • IP PrintWay | • POWER7® | • WebSphere® | |
| • ESCON® | • IPDS | • PowerVM | • xSeries® | |
| • FICON® | • iSeries | • PR/SM | • z9® | |
| | • LANDP® | • pSeries® | • z10 BC | |
| | | • RACF® | • z10 EC | |
- * All other products may be trademarks or registered trademarks of their respective companies.

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

- Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
- Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license there from.
- 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.
- InfiniBand is a trademark and service mark of the InfiniBand Trade Association.
- Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
- IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

Notes:

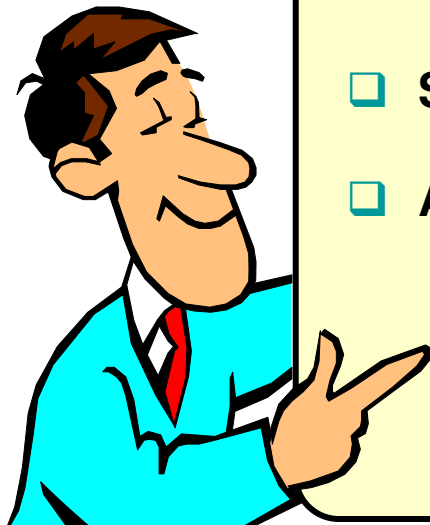
- Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job 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 improvements equivalent to the performance ratios stated here.
- IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
- All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
- This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
- All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
- Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
- Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

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

Agenda



- ❑ **Overview: FTP and Security**
- ❑ **Securing the z/OS FTP client and server**
- ❑ **Securing the FTP connections**
- ❑ **Addressing network traversal challenges**



Let's clear up some common confusion from the start...



▪ **FTP (File Transfer Protocol):**

RFC959
FTP

- Also referred to as RFC959 FTP or “normal” FTP
- The FTP protocol we all know and have used for years.
- Has been extended numerous times since RFC 959 was issued in 1985
- An RFC959 FTP client talks to an RFC959 FTP server - not an sftp server
- What the z/OS CS FTP client and server have supported through many years

▪ **sftp (Secure Shell File Transfer Protocol):**

Secure
Shell
FTP

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

▪ **FTPS (File Transfer Protocol Secure):**

RFC4217
FTP

- Also referred to as FTP – SSL, RFC4217 FTP, FTP AUTH-TLS, FTP AUTH-SSL
- RFC959 FTP but extended with full network security (authentication, data integrity, and data privacy) using a standard security mechanism, such as Kerberos or SSL/TLS
 - SSL/TLS protection specified by RFC4217 "Securing FTP with TLS"
- Both control connection and data connection can be secured
 - No user IDs or password flowing in the clear

Comparison of selected z/OS file transfer technologies from a security perspective

| | FTP With no security RFC959 | FTPS FTP w. SSL/TLS RFC959 + RFC4217 | FTP FTP w. IPsec Any RFC level | SFTP As implemented by IBM Ported Tools |
|---|--|--|---|--|
| User ID and password protection | No | Yes | Yes | Yes |
| Data protection (the file being transferred) | No | Yes | Yes | Yes |
| z/OS UNIX file support | Yes | Yes | Yes | Yes |
| z/OS MVS data set support | Yes | Yes | Yes | No (but add-on products do exist*) |
| Use of System z hardware encryption technologies | n/a | Yes | Yes | Yes (for random number generation) |
| Partner authentication via locally stored copies of public keys | n/a | No | Yes (pre-shared key) | Yes |
| Partner authentication via X509 certificates | n/a | Yes | Yes | No |
| Use of SAF key rings and/or ICSF | n/a | Yes | Yes | Yes |
| FIPS 140-2 mode | n/a | Yes (z/OS V1R11) | Yes (z/OS V1R12) | No |
| Mutual authentication supported | n/a | Yes | Yes (at an IP address level) | Yes |

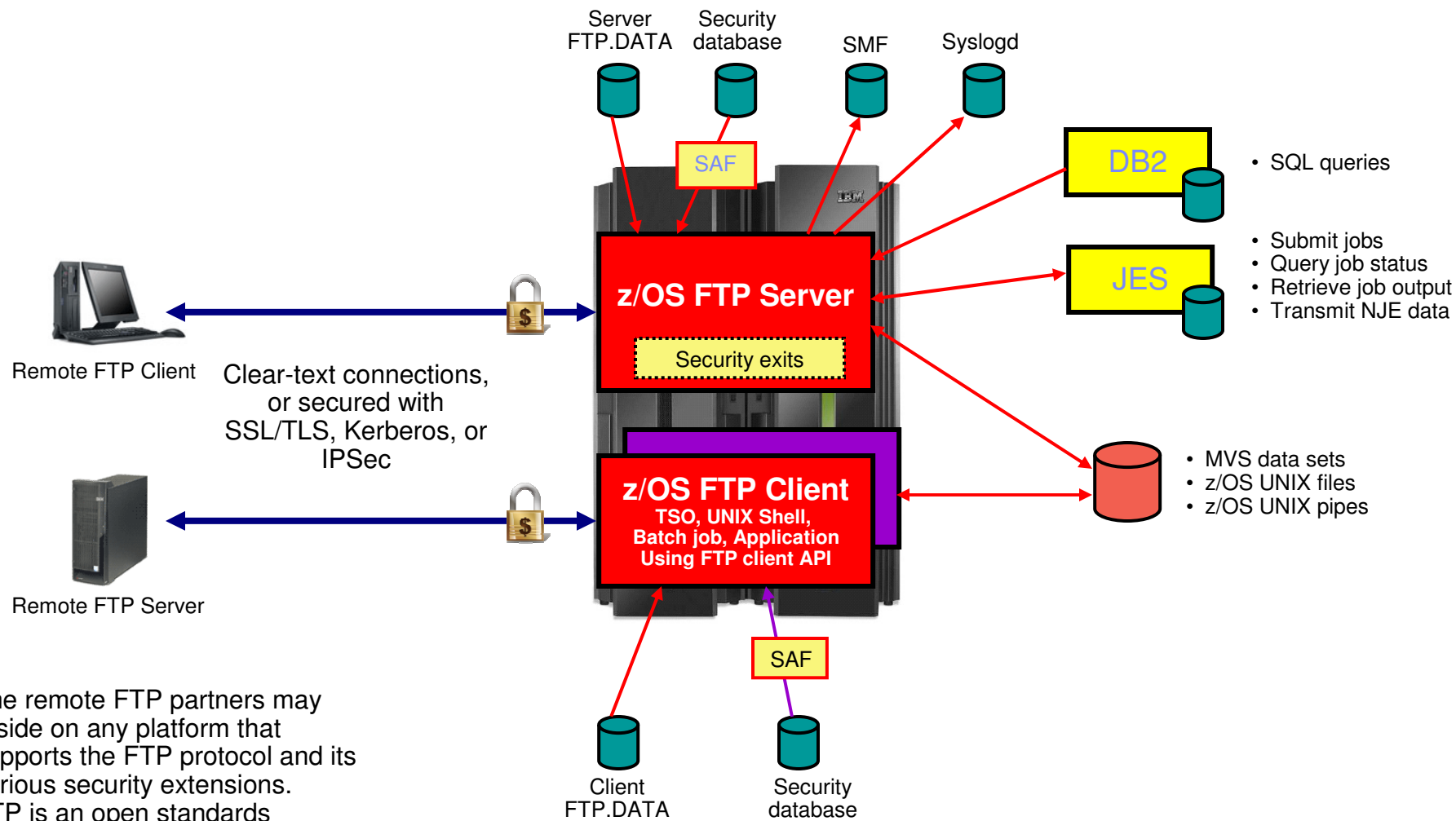
* MVS data set support example: Dovetailed Technologies' Co:Z SFTP

Safe and Secure Transfers with z/OS FTP

Securing the z/OS FTP client and server



z/OS FTP – the big picture



The remote FTP partners may reside on any platform that supports the FTP protocol and its various security extensions. FTP is an open standards protocol.

Securing the local z/OS FTP server

- 1. Basic platform security setup is a prerequisite**
 - Users defined with proper MVS data set access protection
 - z/OS UNIX files defined with proper owning user and group along with user/group/world access permissions
 - ...and so forth
- 2. FTP server-specific SAF resource definitions**
 - Via SERVAUTH resource profiles
- 3. Security-related options in the server's FTP.DATA configuration file**
 - Controlling various aspects of how the FTP server reacts to selected requests, such as a request for anonymous access
- 4. Optional security exits in the FTP server**
 - Can be implemented to provide vary granular levels of controls in the FTP server

Selected SAF resource definitions in the SERVAUTH class

- **EZB.PORTACCESS.sysname.tcpname.port_safname**
 - Controls ability for a started task user ID to establish itself as a server on the matching port number in the TCP/IP Profile port reservation section

- **EZB.FTP.sysname.ftpdname.PORTxxxx**
 - Controls ability to log into an FTP server (control port number) based on the SAF user ID that is being used to log in
 - Initially used for SSL/TLS connections if SECURE_LOGIN VERIFY_USER was coded in the FTP server's FTP.DATA
 - Can be enforced for all types of connections by coding VERIFYUSER TRUE in the server's FTP.DATA - (This support was added in z/OS V1R10)

- **EZB.FTP.sysname.ftpdname.SITE.DUMP** and **EZB.FTP.sysname.ftpdname.SITE.DEBUG**
 - Provides ability to restrict usage of SITE DUMP and DEBUG commands (commands may generate large amount of output)

- **EZB.FTP.sysname.ftpdname.ACCESS.HFS**
 - Provides ability to generally restrict FTP user access to the z/OS UNIX file system

Selected security options in the FTP server's FTP.DATA (1 of 3)

▪ **ANONYMOUS**

- Controls the ability to log into your FTP server as an anonymous user
- If the ANONYMOUS option is not included in the server's FTP.DATA, anonymous access is disabled
- Disabled by default – keep it that way, unless you have specific need for it.
 - If you do enable ANONYMOUS, make sure to change the default value of 1 on the ANONYMOUSLEVEL option to 3
 - Also, verify the settings of all the options that start with “ANONYMOUS” – there are a total of 12 including the ANONYMOUS option itself
 - Use the supplied shell script to build a specific z/OS UNIX file system directory structure for anonymous access
 - EMAILADDRCHECK is a syntax check only of the entered email address

▪ **DEBUGONSITE and DUMPSITE**

- Controls the ability to enable dump and debug SITE command options
- If you set these to TRUE, make sure you define the corresponding SERVAUTH profiles so only authorized users can issue these two SITE command options

▪ **PORTCOMMAND, PORTCOMMANDPORT, PORTCOMMANDIPADDR, and PASSIVEDATACONN**

- Control the ability of your FTP server to participate in three-way proxy mode.

Selected security options in the FTP server's FTP.DATA (2 of 3)

▪ **REPLYSECURITYLEVEL**

- Controls how much identification information is sent on the initial 220 greeting message from the FTP server, and also how much detail is returned when MVS data set contention occurs.
- Default is no restrictions (level 0).
- If your auditors request you to send as little information as possible, use a setting of 1 on this option
 - Level 0: 220-FTPABC1 IBM FTP CS V1R11 at MVS098, 16:42:51 on 2009-05-24.
 - Level 1: 220-IBM FTP, 16:45:57 on 2009-05-24.

▪ **ACCESSERRMSG**

- To prevent details of failed log in attempts to be returned to the FTP client user, set this option to FALSE (which is the default).
- You may change it to TRUE in an internal-only shop if you want your users to receive details about their failed log in attempt.

▪ **SECURE_...**

- There are a number of options that start with SECURE_ - they are all used to control the ability of the FTP server to accept secure connections (SSL/TLS or Kerberos)

Selected security options in the FTP server's FTP.DATA (3 of 3)

▪ VERIFYUSER

- Discussed earlier – extends SAF check of all users' ability to connect to the server's control port number
 - EZB.FTP.sysname.ftpdaemonname.PORTxxxxx

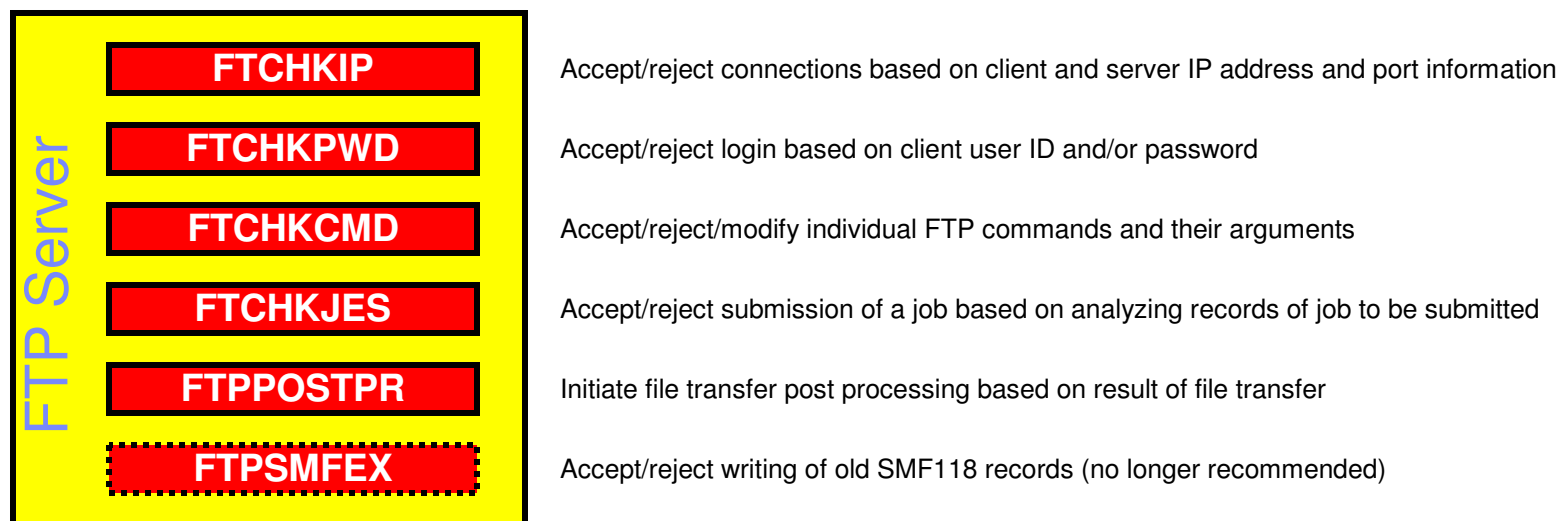
▪ PASSIVEDATAPORTS

- Controls which range of port numbers the server may use for passive mode data connections
- Can be very useful if there are filtering firewalls in-between the FTP client and the FTP server

If you created your server's FTP.DATA data set years ago, we recommend recreating it based on the FTPSDATA member in hlq.SEZAINST. Many new options have been added over the last releases and all are included in this sample member for documentation purposes.



FTP server security exit points – extending FTP server security



- If these exits routines are present they will be loaded and called at the defined exit points
- The FTCHKIP exit is called by the FTP daemon, while the others are called by the FTP server (after the new address space has been created)
- The command check routine is the most widely used. It has information about the current command from the client, what the current working directory is, what file-type we are using, etc. It may reject the command or it may modify the command options, such as the file or data set name on a STOR or RETR command. If it does reject the command, it can also return the text that will be returned to the client in the 500 reply
- The FTCHKCMD exit executes under the logged in user's user ID. Installation-defined SAF resource definitions can be checked in that routine if needed
- The exits are normally coded in assembler, but we have seen examples where they were coded in C.

FTP server security exit details

| Exit point | Called by | Called when | Main input | Possible actions |
|-----------------|----------------------|---|---|---|
| FTCHKIP | Daemon address space | When control connection is being accepted by the FTP daemon | Client and server IP addresses and ports | Accept or reject connection setup |
| FTCHKPWD | Server address space | When the client user sends the PASS command | IP addresses and ports, client user ID and password | Accept or reject login request |
| FTCHKCMD | Server address space | For every command received over the control connection | IP addresses and ports, client user ID, directory type, file type, current directory, and the FTP command and arguments | Accept, reject, or modify the FTP command |
| FTCHKJES | Server address space | For every record in a job that is being submitted to JES | IP addresses and ports, the full JES input record | Accept or reject the job submission |
| FTPOSTPR | Server address space | For every completed file transfer operation | IP addresses and ports, plus details about the completed file transfer | Initiate post processing |

Samples for all in hlq.SEZAINST

Securing the local z/OS FTP client

- Basic platform security setup is a pre-requisite
 - Users defined with proper MVS data set access protection
 - z/OS UNIX files defined with proper user/group/world access permissions
 - Etc.

- FTP client-specific SAF resource definitions
 - None for the FTP client

- Security-related options in the client's FTP.DATA
 - None

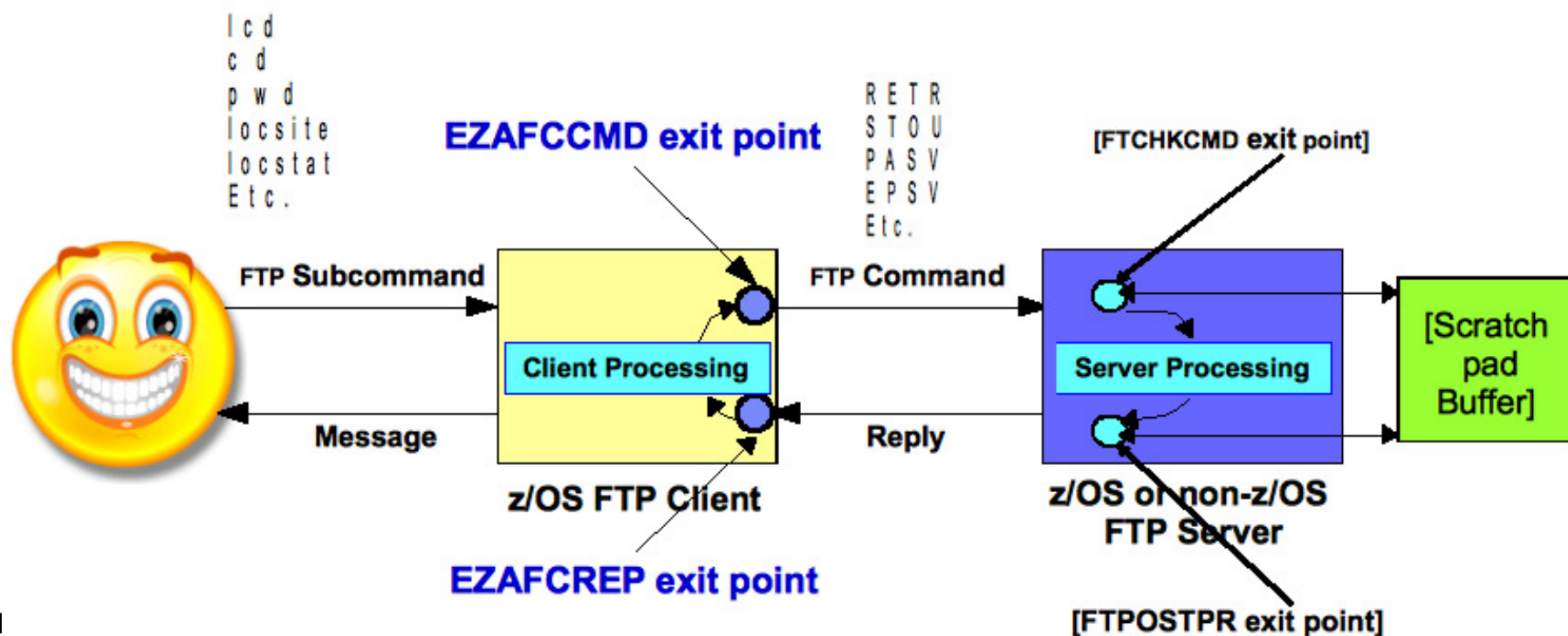
- Optional security exits
 - No exit points in the z/OS FTP client prior to V2R1

FTP client security user exits in V2R1

- You can use FTP server user exits to limit access to an FTP server, but...
- Currently, a system administrator has no way of controlling FTP client commands or other aspects of the processing done in the z/OS FTP client. Examples of tasks that a system administrator might desire:
 - Preventing a dataset (that the user should have at least limited access to) from being moved from the z/OS host
 - The ability to inspect or modify the dataset names specified by FTP client users for inbound and outbound file transfers
 - The ability to cancel an FTP client address space if that client is in the process of sending an “unauthorized” FTP command
- To address this, V2R1 will provide two FTP client user exit points:
 - FTP command user exit – EZAFCCMD
 - Receives control just before an FTP command is being sent
 - Enables the exit to inspect the command, optionally modify the command arguments, reject the command, or request the FTP client session be terminated
 - FTP reply user exit – EZAFCREP
 - Receives control when an FTP server reply arrives
 - Allows the exit to analyze the results of commands sent to the server, and request the FTP client session be terminated

FTP client security user exits in V2R1 ...

- The EZAFCMD user exit will be called whenever the FTP client is about to send an FTP command to the server.
 - The call of this user exit is done before the command is converted to ASCII
- The EZAFCREP user exit will be called whenever the FTP client receives a reply line over the control connection
 - The call of this exit is done after the reply has been converted to EBCDIC

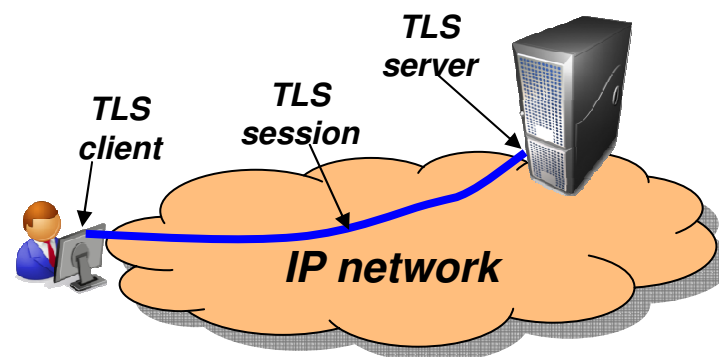


Securing FTP with TLS on z/OS



Transport Layer Security (TLS/SSL) overview

- Transport Layer Security (TLS) is defined by the IETF **
 - Based on Secure Sockets Layer (SSL)
 - TLS defines SSL as a version of TLS for compatibility
- Provides secure connectivity between two TLS security session endpoints
 - TLS session
- Full application payload encryption and data authentication / integrity
- TLS security session endpoint plays either a client or server role
- Session endpoint authentication typically via X.509 certificates
 - Server authentication required
 - Client authentication optional (mutual authentication)



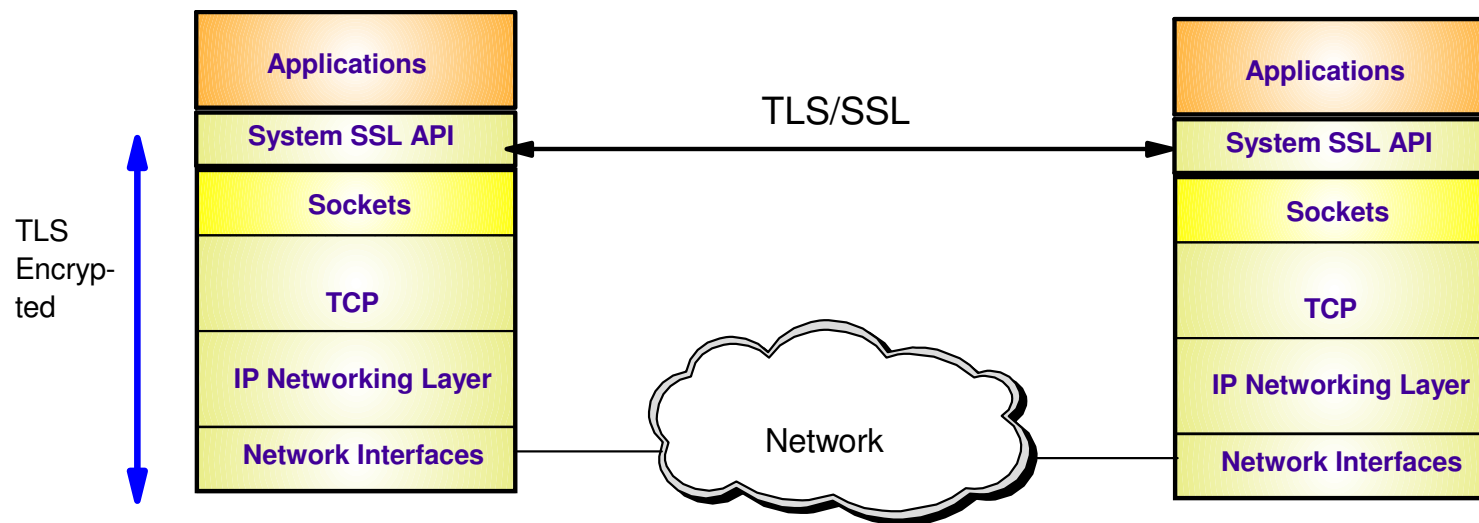
Full application payload encryption

TLS/SSL encryption:

| SrcIP | DestIP | SrcPort | DestPort | Data |
|---------------|-------------|---------|----------|----------------------|
| 192.168.100.1 | 192.168.1.1 | 50002 | 443 | @%\$#*&^^!:'J)*GVM>< |

**** For our purposes, SSL and TLS are equivalent and one term implies the other**

Transport Layer Security enablement



- TLS traditionally provides security services as a socket layer service
 - TLS requires reliable transport layer,
 - Typically TCP (but architecturally doesn't have to be TCP)
 - UDP applications cannot be enabled with traditional TLS
 - There is now a TLS variant called Datagram Transport Layer Security (DTLS) which is defined by the IETF for unreliable transports
- On z/OS, System SSL (a component of z/OS Cryptographic Services) provides an API library for TLS-enabling your C and C++ applications
- Java Secure Sockets Extension (JSSE) provides libraries to enable TLS support for Java applications
 - However, there is an easier way...

z/OS Application Transparent TLS overview

▪ Stack-based TLS

- TLS process performed in TCP layer (via System SSL) without requiring any application change (transparent)
- AT-TLS policy specifies which TCP traffic is to be TLS protected based on a variety of criteria
 - Local address, port
 - Remote address, port
 - Connection direction
 - z/OS userid, jobname
 - Time, day, week, month

▪ Application transparency

- Can be fully transparent to application
- An optional API allows applications to inspect or control certain aspects of AT-TLS processing – “application-aware” and “application-controlled” AT-TLS, respectively

▪ Available to TCP applications

- Includes CICS Sockets
- Supports all programming languages except PASCAL

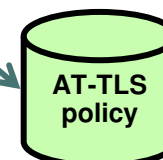
▪ Supports standard configurations

- z/OS as a client or as a server
- Server authentication (server identifies self to client)
- Client authentication (both ends identify selves to other)

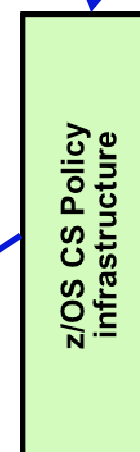
▪ Uses System SSL for TLS protocol processing

- Remote endpoint sees an RFC-compliant implementation
- interoperates with other compliant implementations

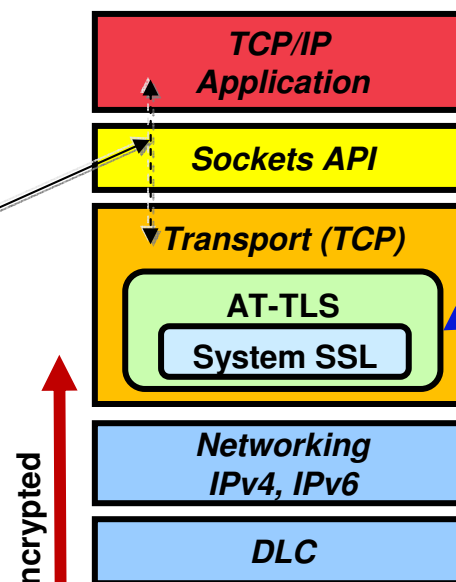
AT-TLS policy administrator using Configuration Assistant



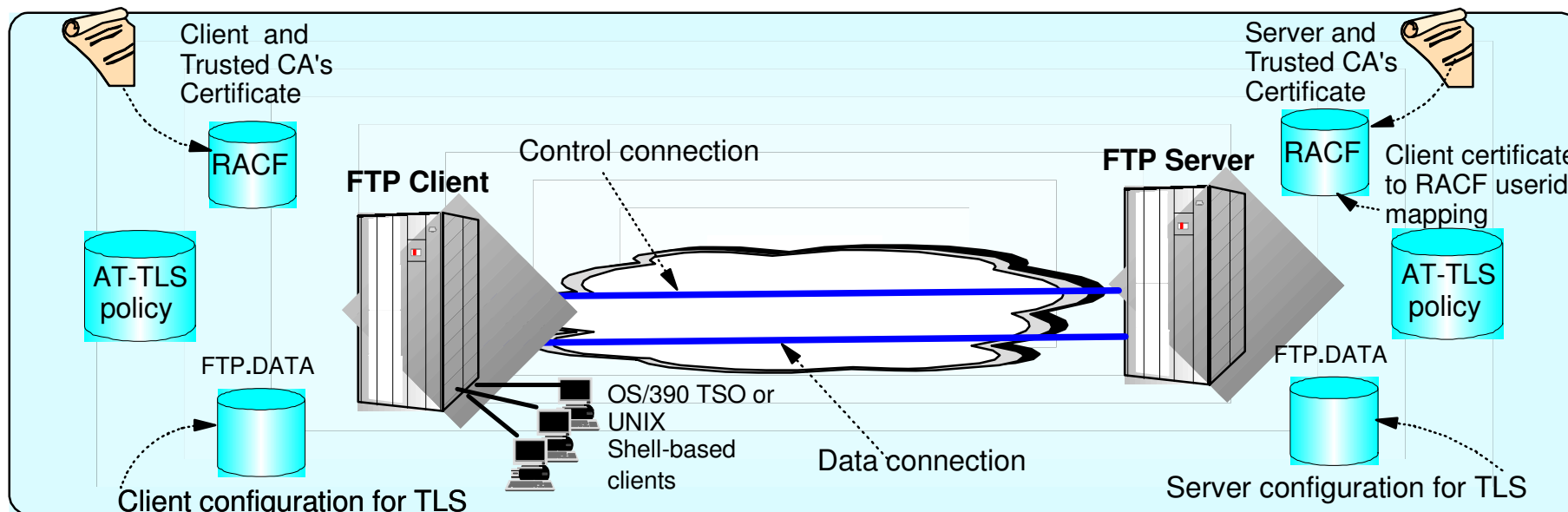
AT-TLS policy



z/OS CS Policy infrastructure



AT-TLS enabled FTP overview



- **TLS protects control and data connection OR control connection alone**
 - Data connection cannot be protected without control connection
- **Authentication of end users :**
 - Basic - userid/password over encrypted control connection
 - Several options based on TLS session authenticated client certificate
- **Three main areas of FTP TLS configuration**
 - RACF keyrings for certificates and private keys
 - AT-TLS policy defines which FTP traffic to protect and how to protect it (TLS security attributes)
 - FTP.DATA configuration (server and client) controls FTP protocol-specific security policy for TLS session

How TLS is requested for FTP

- TLS Modes:
 - ▶ Unconditional TLS
 - Uses separate protected ports for TLS (port 989 and 990)
 - ✓ TLS for client and server assumed
 - ✓ Not included in RFC 4217
 - ▶ Negotiable TLS
 - Both TLS and non-TLS traffic share standard ports (20 and 21)
 - ✓ Negotiation based on subset of the FTP security negotiation functions documented in RFC 2228 and further clarified in RFC 4217

- Negotiable mode TLS requested with client FTP command
 - ▶ AUTH TLS

- Configuration for negotiable mode at the server (FTP.DATA)
 - ▶ Specify that the AUTH comand with TLS is supported
 - EXTENSIONS statement with Auth_TLS
 - ▶ Specify that TLS required or optional on the standard FTP ports
 - SECURE_FTP statement with REQUIRED | ALLOWED

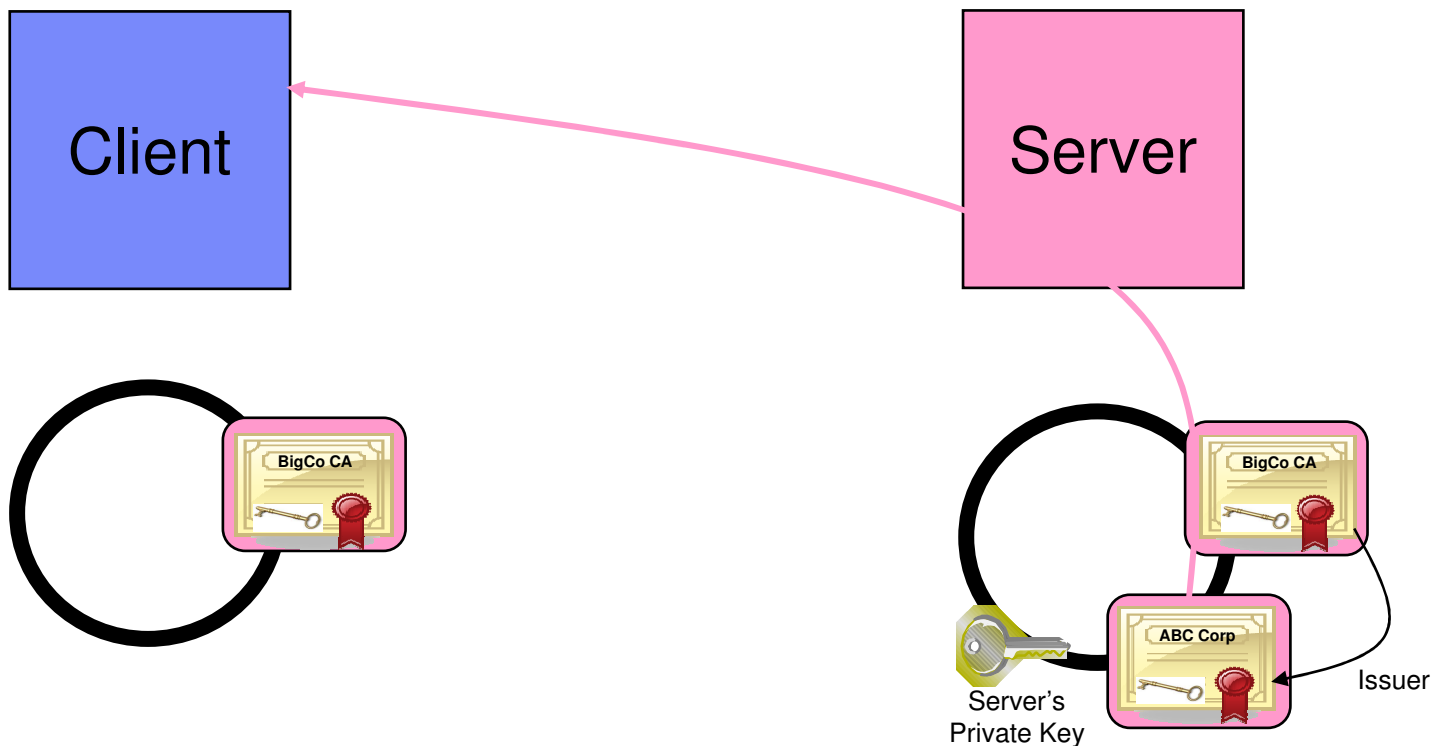
Securing FTP by Connection Type

- FTP has both a control and data connection. Possible combinations of TLS protection are:
 - ▶ Control connection security only
 - ▶ Both control connection and data connection security
 - **Data connection security only not supported**

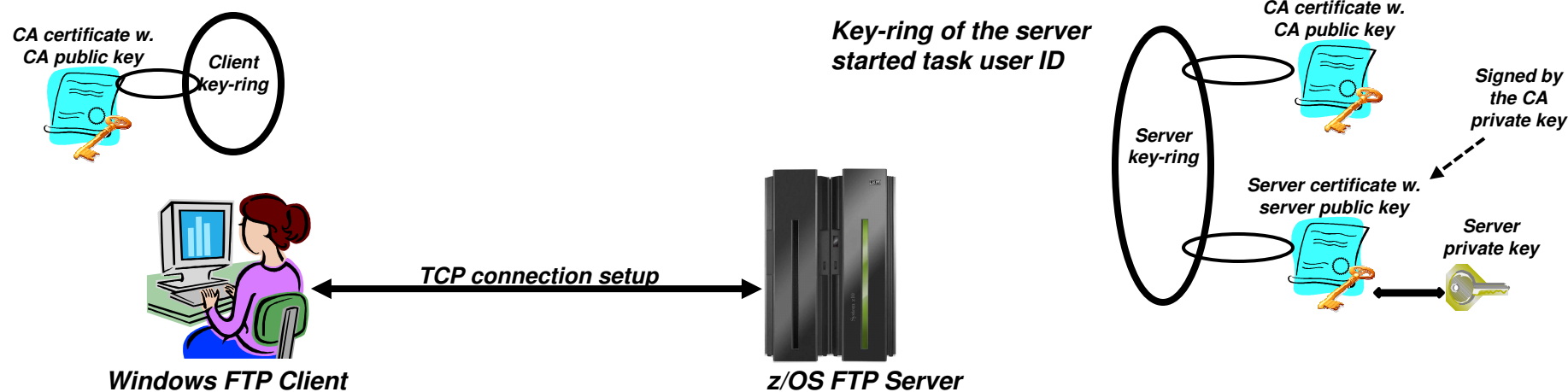
- Data connection protection levels requested by client FTP command
 - ▶ PROTECT private
 - TLS always uses data authentication and integrity / encryption is optional
 - Protection is based on ciphersuite negotiations
 - TLS session is negotiated for each data connection
 - ▶ PROTECT clear
 - No TLS for the data connection

- FTP server can be configured to specify security requirements for the data connection. Options are:
 - ▶ SECURE_DATACONN
 - NEVER - Not allowed
 - CLEAR - Allowed, Let client decide
 - PRIVATE - Required

Certificates in action: SSL server authentication



What is needed for z/OS Server authentication only (which is sufficient for encrypted data exchange)



Windows FTP Client → **z/OS FTP Server**
TCP connection setup

z/OS FTP Server → **Windows FTP Client**
Hello – I want to use SSL/TLS

Windows FTP Client → **z/OS FTP Server**
Hello – OK, me too !!

z/OS FTP Server → **Windows FTP Client**
And here is my server certificate



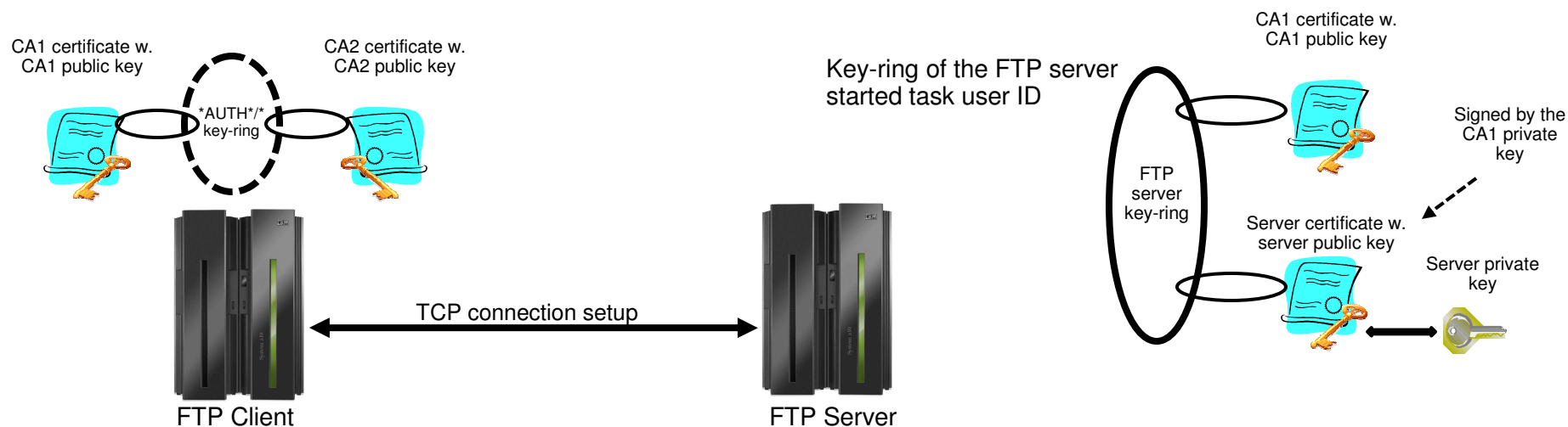
1. Verify server certificate has not expired
2. Verify server certificate is valid using CA's public key
3. Do optional checks on the server certificate
4. Store server's public key for later use
5. Generate symmetric key and encrypt under server's public key

Windows FTP Client → **z/OS FTP Server**
*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 server's started task user ID's key-ring 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

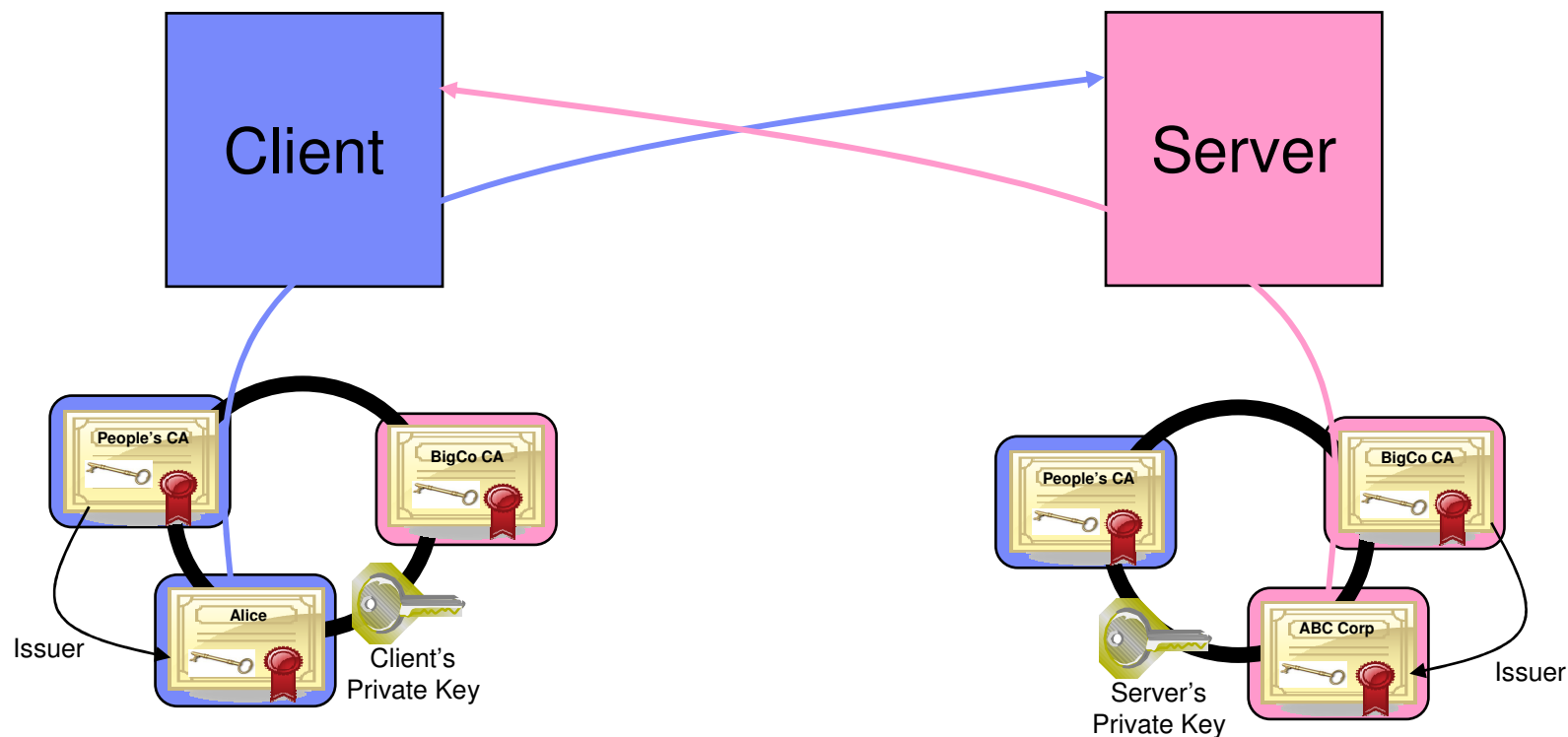
Virtual key-rings are useful when z/OS is the FTP client

- 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?
 - Originally, the answer was yes
 - What we call an "administratively heavy process"
 - z/OS V1R8 added support for something known as a virtual key-ring
- To have System SSL check all CERTAUTH certificates in RACF when verifying a certificate that was received during the SSL handshake, specify a key-ring in AT-TLS policy as:
 - KEYRING *AUTH*/*
- If client authentication is required, the z/OS FTP user still needs his/her own key-ring

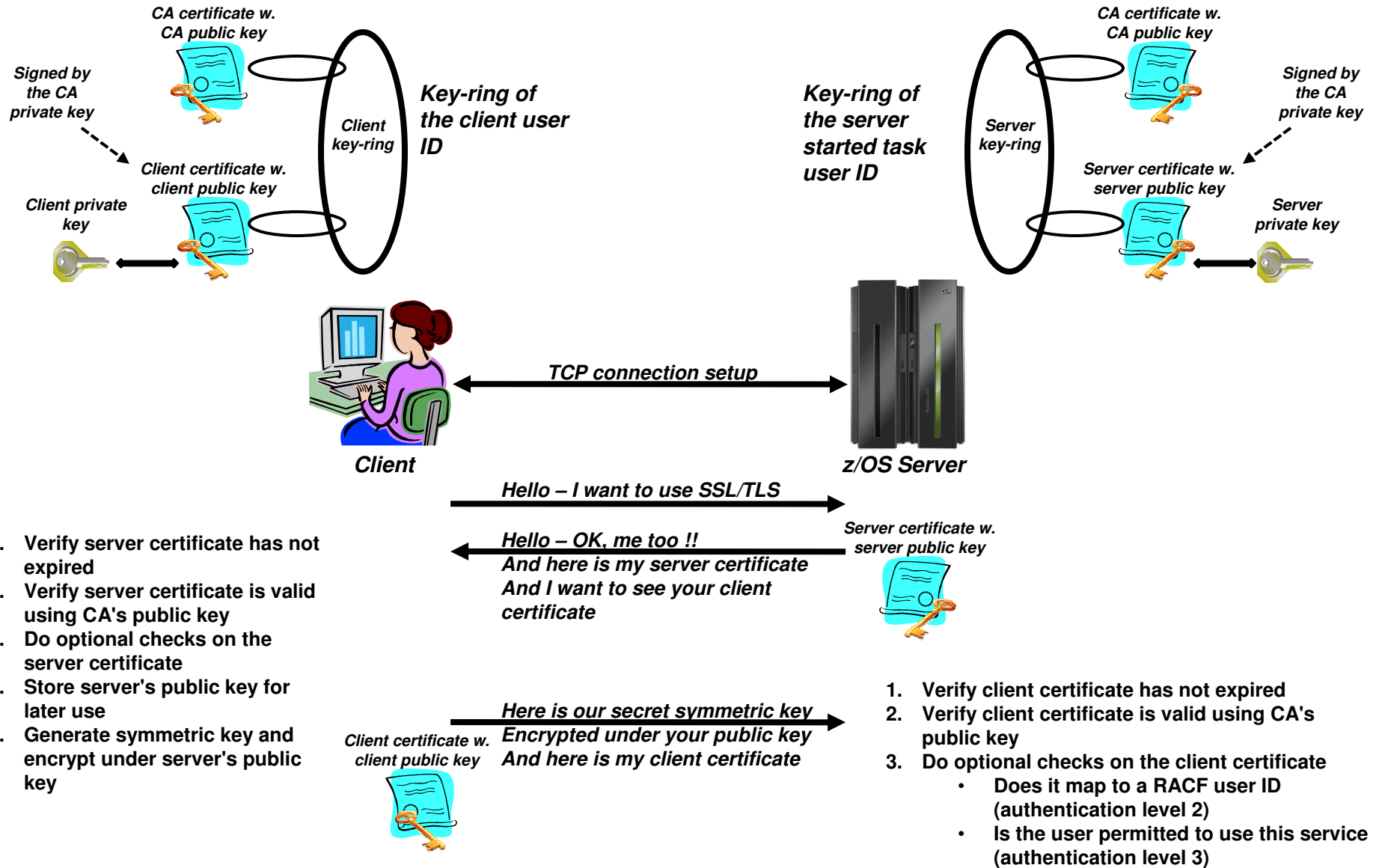


Certificates in action: SSL client authentication

(implies server authentication as well)



What is needed for z/OS Server and client authentication?



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

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
 - Does it map to a RACF user ID (authentication level 2)
 - Is the user permitted to use this service (authentication level 3)

z/OS FTP server options for authenticating an FTP client using client certificates and AT-TLS

| Authentication level | FTP server <code>SECURE_LOGIN</code> option | Description |
|----------------------|---|--|
| Level 1 | REQUIRED | The authenticity and validity of the client certificate is verified against the trusted roots in the FTP server's key-ring. |
| 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 <u>SERVAUTH</u> profile that represents this specific FTP server: EZB.FTP.sysname.ftpdemonname.PORTnnnnn |

Configuring AT-TLS policy: Server traffic descriptor and keyring using IBM Configuration Assistant for z/OS Communications Server

Modify Rule

AT-TLS rule name
 Rule name: * ABC_FTP_4021 Enable rule

Specify settings
 Traffic | Role | Key Ring | Data Endpoints | Security Level | Advanced

Use this panel to specify the traffic settings.
 Application name: * ABC_FTP4021_New

Local port
 All ports
 All ephemeral ports
 Ports: * 4021
Separate multiple ports with a co...

Remote port
 All ports
 All ephemeral ports
 Ports: *
Separate multiple po...

Indicate the TCP connect direction
 Either Inbound only Outbound only

Specify jobname and user ID
 Jobname:

Modify Rule

AT-TLS rule name
 Rule name: * ABC_FTP_4021 Enable rule

Specify settings
 Traffic | Role | Key Ring | Data Endpoints | Security Level | Advanced

Use this panel to specify the key ring database and certificate label to use for this rule.

Key ring database
 Use the key ring database defined for the z/OS image
 Use a Simple name (as in an SAF product or in PKCS #11 Token format):
 Key ring: * TLSRING

Use this z/OS UNIX file system key database:
 Key database: *
 Key database stash file: * or
 Key database password: *

Certificate label:

Configuring AT-TLS policy: Data endpoints

using IBM Configuration Assistant for z/OS Communications Server

Modify Rule

AT-TLS rule name

Rule name: * ABC_FTP_4021 Enable rule

Specify settings

Traffic Role Key Ring **Data Endpoints** Security Level Advanced

Select the address groups of the host endpoints of the traffic you want to protect.

Local data endpoint

Address group

All_IP_Addresses

New... Copy... Modify... View Details

IPv4 or IPv6 address, subnet or range

*

Examples: x.x.x.x, x.x.x.x/yy, x.x.x.x.y.y.y.y
x:x, x:x/yyy, x:x-yy

Remote data endpoint

Address group

All_IP_Addresses

New... Copy... Modify... View Details

IPv4 or IPv6 address, subnet or range

*

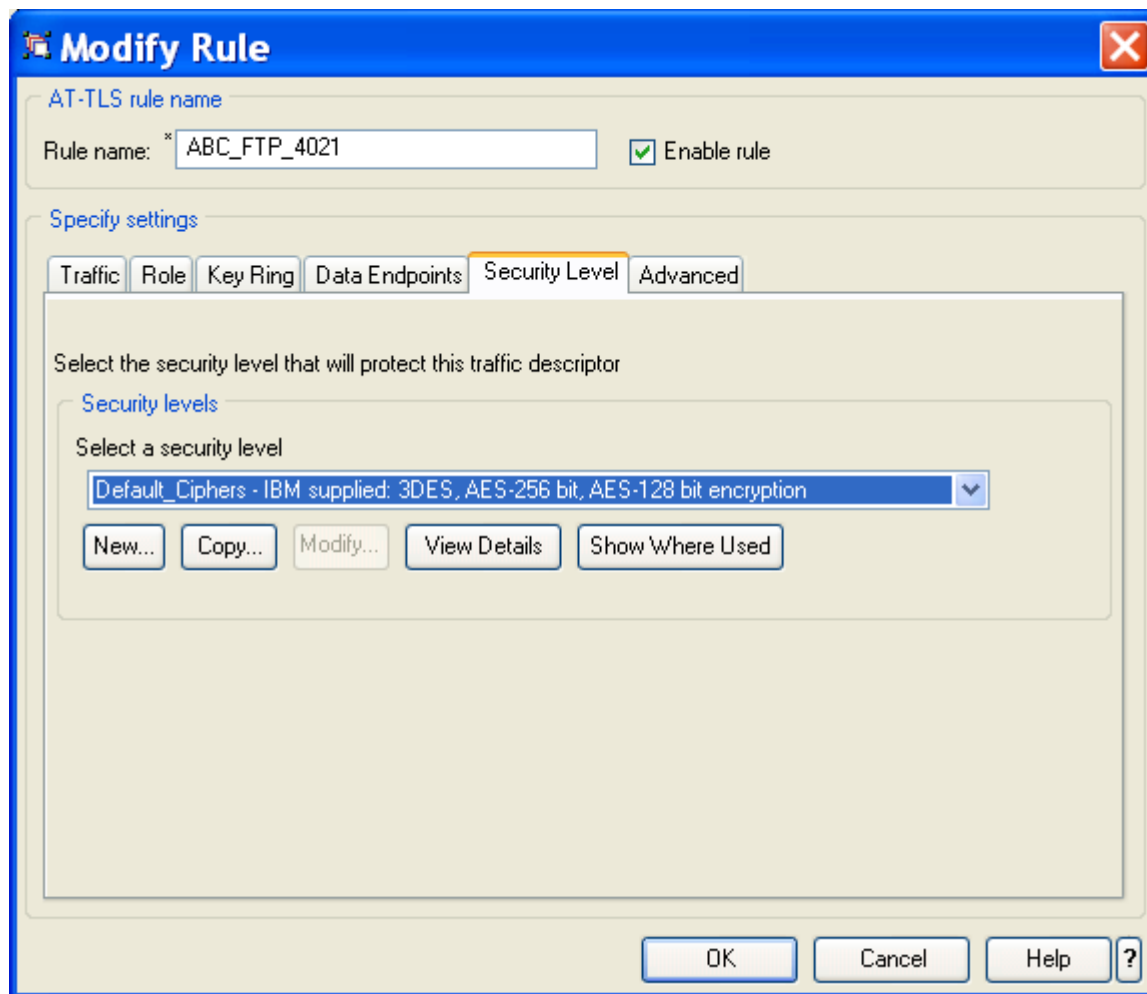
Examples: x.x.x.x, x.x.x.x/yy, x.x.x.x.y.y.y.y
x:x, x:x/yyy, x:x-yy

OK Cancel Help ?

Configuring AT-TLS policy: Security level

using IBM Configuration Assistant for z/OS Communications Server

- **Type:**
 - AT-TLS
- **Encryption:**
 - 0x35 - TLS_RSA_WITH_AES_256_CBC_SHA (first choice)
- **Use TLS Version 1.0:**
 - Yes
- **Use TLS Version 1.1:**
 - Yes
- **Use SSL Version 3:**
 - Yes
- **Use SSL Version 2:**
 - No
- **Client authentication:**
 - None
- **FIPS 140 Support:**
 - Off



z/OS FTP server FTP.DATA parameters for AT-TLS

| | | | |
|------------------------|-----------------|--|--|
| EXTENSIONS | AUTH_TLS | ; Enable TLS authentication ; Default is disabled. | |
| TLSMECHANISM | ATTLS | ; Server-specific or ATTLS ; ATTLS - use ATTLS ; FTP - server-specific (D) | Switch between FTP's built-in SSL/TLS support and ATTLS support |
| SECURE_FTP | ALLOWED | ; Authentication indicator ; ALLOWED (D) ; REQUIRED | Must all connections be secure or just those who wish to be? |
| SECURE_LOGIN | REQUIRED | ; Authorization level indicator ; for TLS ; NO_CLIENT_AUTH (D) ; REQUIRED ; VERIFY_USER | Is client authentication required and if so, at what level? |
| SECURE_PASSWORD | REQUIRED | ; REQUIRED (D) - User must enter ; password ; OPTIONAL - User does not have to ; enter a password | If client authentication is used at level 3 and a user ID can be matched, is a password still required or not? |
| SECURE_DATACONN | CLEAR | ; Minimum level of security for ; the data connection ; NEVER ; CLEAR (D) ; PRIVATE | Server's requirement to security of the data connection |
| TLSRFCLEVEL | RFC4217 | ; Specify what level of RFC 4217, ; On Securing FTP with TLS, is ; supported. ; DRAFT (D) Internet Draft level ; RFC4217 RFC level | Is z/OS FTP server to operate at the old draft RFC level for SSL/TLS or the now existing RFC? The default is to use draft - you may want to change that! |

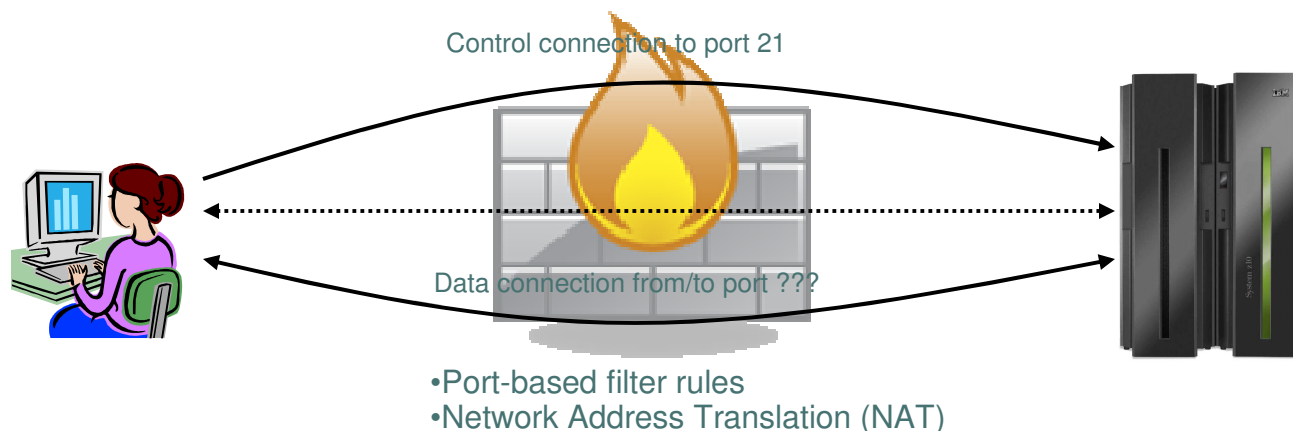
z/OS FTP client FTP.DATA parameters for secure connections

| | | | |
|-------------------------------|----------------------|--|---|
| <code>SECURE_MECHANISM</code> | <code>TLS</code> | <pre> ; Name of the security mechanism ; that the client uses when it ; sends an AUTH command to the ; server. ; GSSAPI = Kerberos support ; TLS = TLS </pre> | Set to use TLS when AUTH command used |
| <code>TLSMECHANISM</code> | <code>ATTLS</code> | <pre> ; SSL/TLS implementer ; FTP - FTP use of system SSL ; ATTLS - the ATTLS component </pre> | Switch between FTP's built-in SSL/TLS support and ATTLS support |
| <code>SECURE_FTP</code> | <code>ALLOWED</code> | <pre> ; Authentication indicator ; ALLOWED (D) ; REQUIRED </pre> | Must all connections be secure or just those who wish to be? |
| <code>SECURE_DATACONN</code> | <code>PRIVATE</code> | <pre> ; Minimum level of security for ; the data connection ; NEVER ; CLEAR (D) ; PRIVATE </pre> | Client's requirement to security of the data connection |

Addressing Network Traversal Challenges



Firewalls and FTP issues



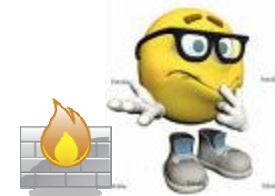
▪ Port-based filter rules – in particular dynamic port rules

- FTP control connection is no problem - pre-defined server port number (default 21)
- Data connection port number (or direction) is not pre-defined, but dynamically negotiated between the FTP client and server
 - The firewall does “deep inspection” (peeks into) the FTP control connection to learn about the negotiated ports and the direction for the data connection

▪ NAT

- FTP control connection is no problem – only IP headers need translation
- PORT command and PASV reply refers to local (intranet) IP addresses
 - Firewall needs to do “deep inspection” of the FTP control connection to locate and modify the IP address information in the PORT command and the PASV reply

Deep inspection and data modification is impossible when the data on the FTP control connection is secured through encryption and message integrity checking at the end points.

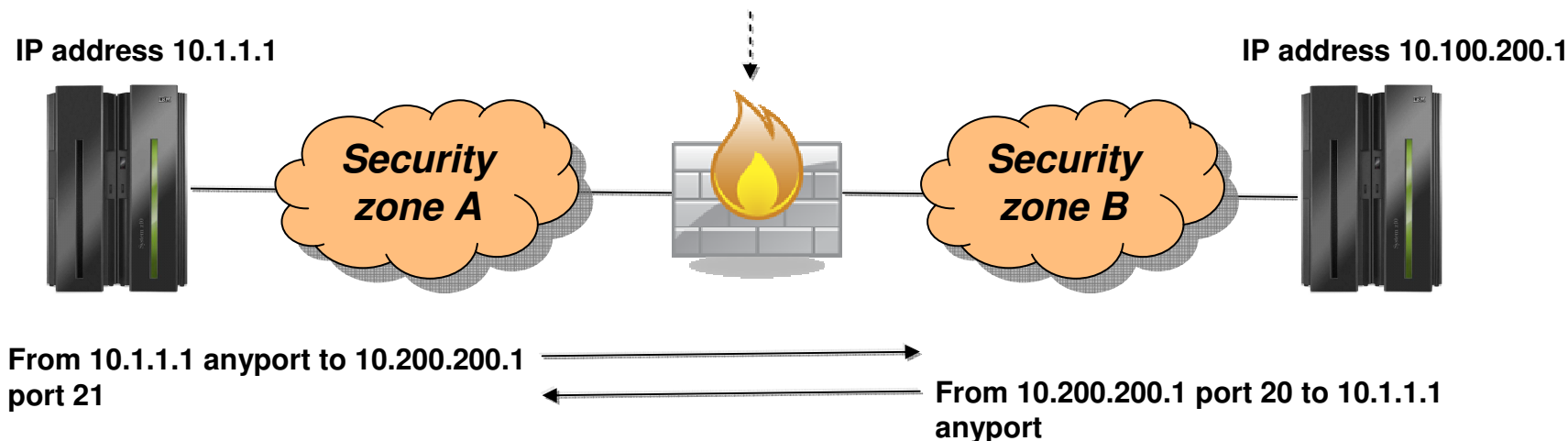


How to deal with static port-based filters in firewalls (active mode)

- 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

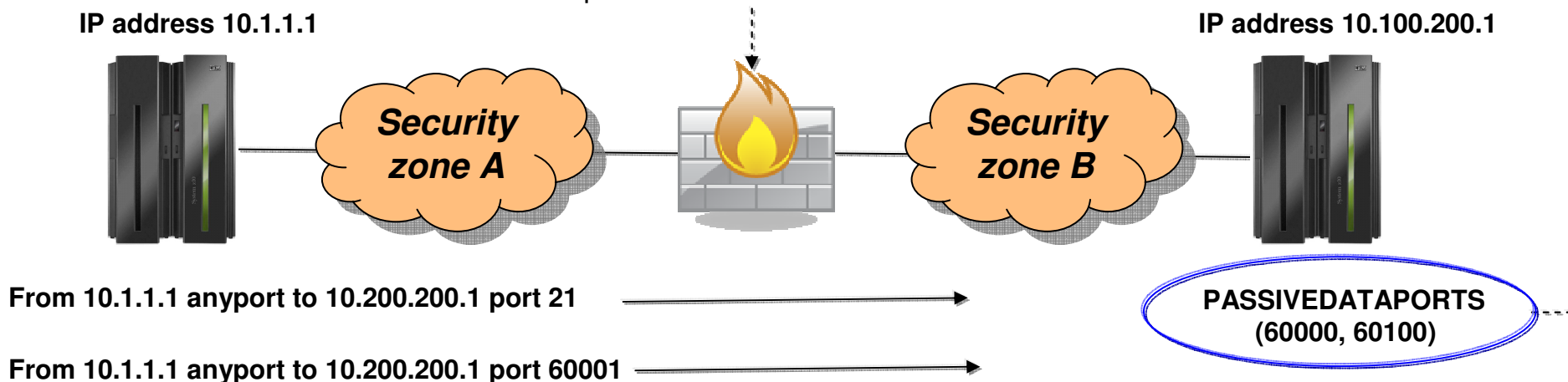


How to deal with static port-based filters in firewalls (passive mode)

- 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)

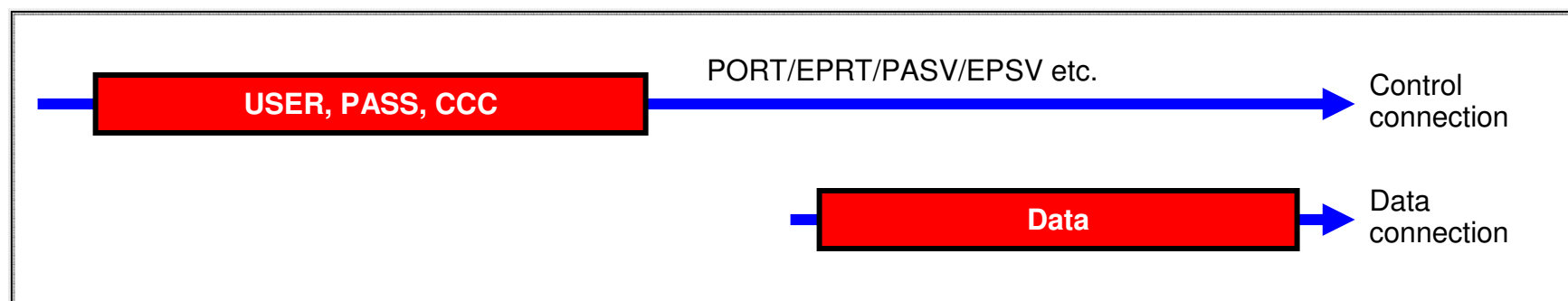
Static firewall filters

- ▶ 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



How to deal with dynamic port-based filters in firewalls

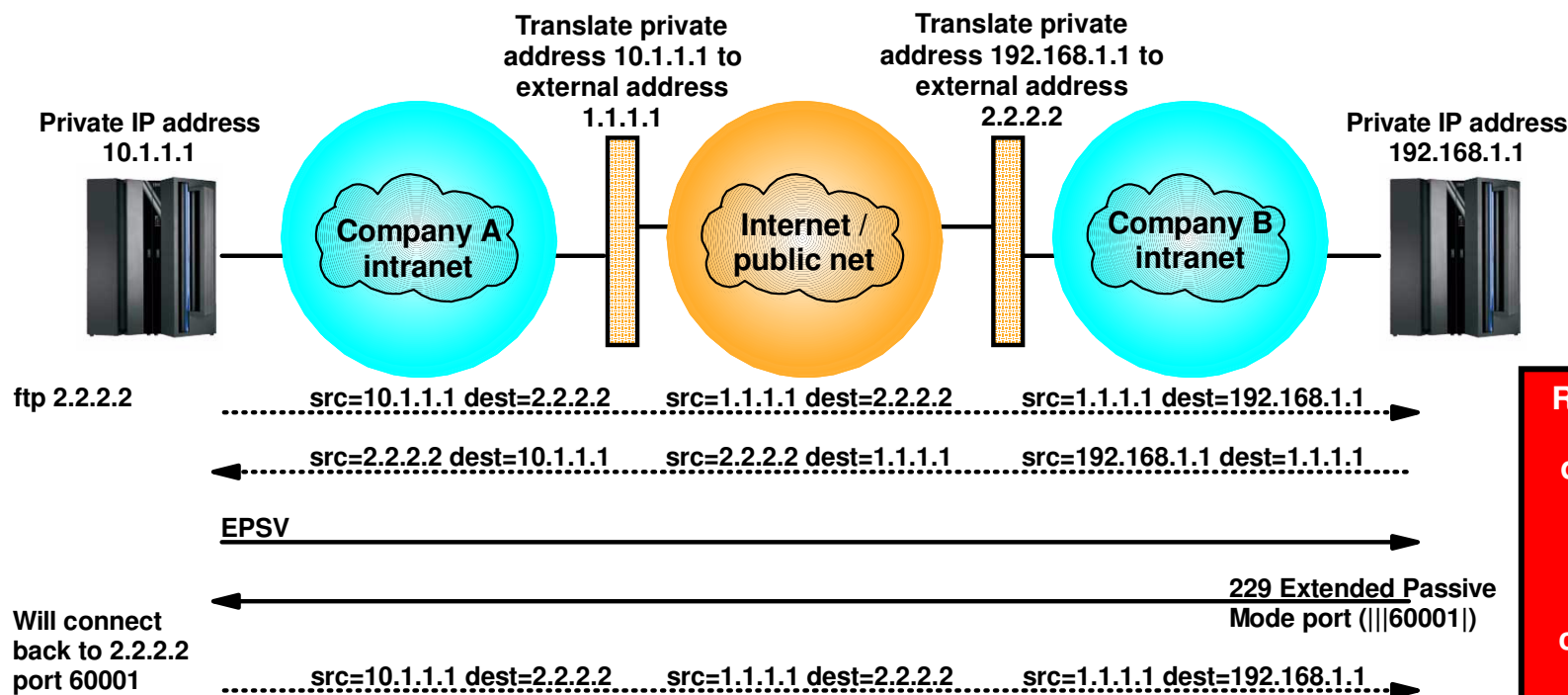
- When using dynamic filters, the firewall enables (permits) ports based on IP address and/or port number information in the PORT/EPRT command or the PASV/EPSV reply
 - The original FTP SSL/TLS draft RFC stated that the FTP control connection always had to be encrypted!
 - The final RFC (RFC 4217 "Securing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



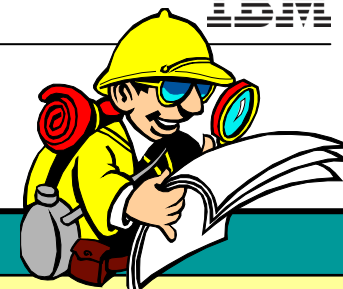
- Both the FTP client and server need to support the CCC command according to RFC 4217
 - Not all FTP clients and servers that support FTP SSL/TLS support the CCC command
 - z/OS added full support for the CCC command in z/OS V1R9 (both z/OS FTP client and server)
 - For those products that claim support, some interoperability issues have been observed !
 - If you have problems getting CCC to work, try to specify `TLSRFCLEVEL CCCNONOTIFY` instead of `TLSRFCLEVEL RFC4217` (applies to both z/OS FTP server and client)

RFC 2428: FTP Extensions for IPv6 and NATs

- Extended passive mode (EPSV) will solve NAT problems for secure FTP sessions
 - If using z/OS FTP client to a server that does not support EPSV, code `PASSIVEIGNOREADDR TRUE` in the FTP client's `FTP.DATA`
- 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 command (EPRT) are also used to enable IPv6 support in FTP
 - Used with IPv4, the EPSV command provides NAT firewall relief



RFC 2428 does not help with dynamic port-based filter rules in Firewalls! CCC FTP command still needed!



For more information

| URL | | Content |
|---|--|---|
| http://www.twitter.com/IBM_Commserver | | IBM Communications Server Twitter Feed |
| http://www.facebook.com/IBMCommserver | | IBM Communications Server Facebook Fan Page |
| http://www.ibm.com/systems/z/ | | IBM System z in general |
| http://www.ibm.com/systems/z/hardware/networking/ | | IBM Mainframe System z networking |
| http://www.ibm.com/software/network/commserver/ | | IBM Software Communications Server products |
| http://www.ibm.com/software/network/commserver/zos/ | | IBM z/OS Communications Server |
| http://www.ibm.com/software/network/commserver/z_lin/ | | IBM Communications Server for Linux on System z |
| http://www.ibm.com/software/network/ccl/ | | IBM Communication Controller for Linux on System z |
| http://www.ibm.com/software/network/commserver/library/ | | IBM Communications Server library |
| http://www.redbooks.ibm.com | | ITSO Redbooks |
| http://www.ibm.com/software/network/commserver/zos/support/ | | IBM z/OS Communications Server technical Support – including TechNotes from service |
| http://www.ibm.com/support/techdocs/atmastr.nsf/Web/TechDocs | | Technical support documentation from Washington Systems Center (techdocs, flashes, presentations, white papers, etc.) |
| http://www.rfc-editor.org/rfcsearch.html | | Request For Comments (RFC) |
| http://www.ibm.com/systems/z/os/zos/bkserv/ | | IBM z/OS Internet library – PDF files of all z/OS manuals including Communications Server |

For pleasant reading

Please fill out your session evaluation

- Safe and Secure Transfers with z/OS FTP
- Session # 12855
- QR Code:

