z/OS® CS - SHARE in Anaheim, CA - February/March 2005

Session 3924 - Networking - TCP/IP

Secure Communications with FTP and TN3270 on z/OS CS: Concepts and Considerations

Thursday 3-Mar-2005 - 9:30 AM

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

Secure Communications with FTP and TN3270 on z/OS CS: Concepts and Considerations

Session Number:	3924
Date:	Thursday, 3-Mar-2005
Time:	9:30 AM
Location:	Hilton Anaheim - Fourth floor - San Clemente
Speaker:	Alfred B Christensen, IBM
Chair:	Alfred B Christensen, IBM
Abstract:	In this session, the speaker will discuss the security-related functions in the z/OS TN3270 server (SSL and Express Logon), and FTP server (SSL/TLS). The session will start with an introduction to the different security technologies, and will discuss considerations for using these functions in an enterprise and between enterprises to implement secure communication for terminal access and file transfer functions

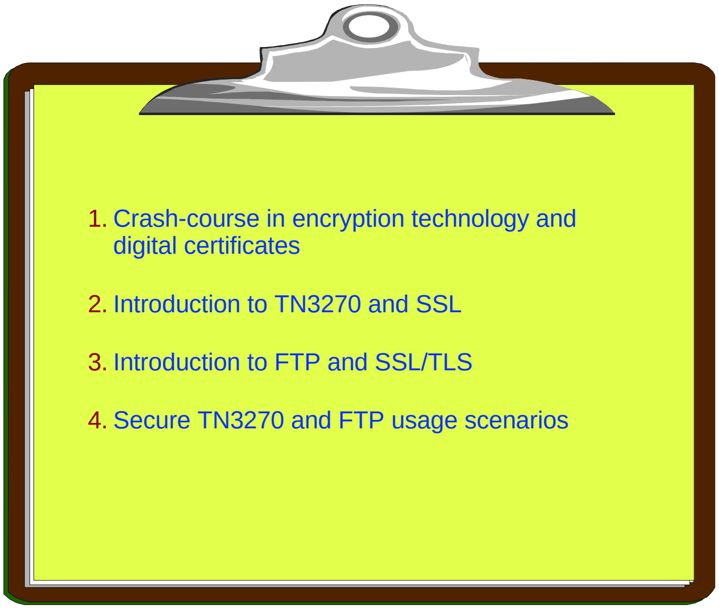
Trademarks and notices

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

- AIX®
- AnyNet®
- AS/400®
- Candle®
- CICS®
- CICSPlex®
- CICS/ESA®
- DB2®
- DB2 Connect™
- DPI®
- DRDA®
- e business(logo)®
- ESCON®
- eServer™
- ECKD™
- FFST™
- GDDM®
- GDPS®
- HiperSockets™
- IBM®
- Infoprint®
- IMS™
- IP PrintWay™
- iSeries™
- Language Environment®
- MQSeries®
- MVS™
- MVS/ESA™
- NetView®
- OS/2®
- OS/390®
- Parallel Sysplex®
- PrintWay™
- PR/SM™
- pSeries®
- RACF®
- Redbooks™
- Redbooks (logo)™
- S/390®
- System/390®
- ThinkPad®
- Tivoli®
- Tivoli (logo)®
- VM/ESA®
- VSE/ESA™
- VTAM®
- WebSphere®
- xSeries®
- z/Architecture™
- z/OS®
- z/VM®
- 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 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.
➢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 represents goals and objectives only.

Topics



1. Crash-course in encryption technology and digital certificates
2. Introduction to TN3270 and SSL
3. Introduction to FTP and SSL/TLS
4. Secure TN3270 and FTP usage scenarios

This session will not discuss Kerberos-based security.

Where do the various IP-based security technologies belong in a protocol stack view?

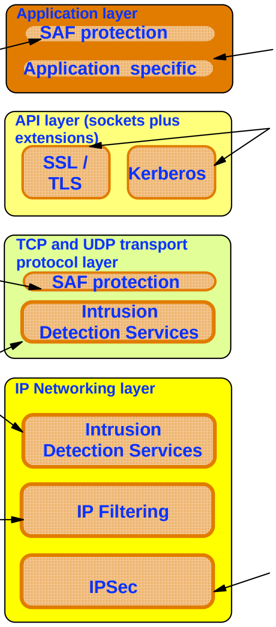
Protect the system

z/OS CS TCP/IP applications use SAF to authenticate users and prevent unauthorized access to datasets, files, and SERVAUTH protected resources..

The SAF SERVAUTH class is used to prevent unauthorized user access to TCP/IP resources (stack, ports, networks)

Intrusion detection services protect against attacks of various types on the system's legitimate (open) services. IDS protection is provided at both the IP and transport layers.

IP filtering blocks out all IP traffic that this systems doesn't specifically permit.


Application layer
SAF protection
Application specific
API layer (sockets plus extensions)
SSL / TLS
Kerberos
TCP and UDP transport protocol layer
SAF protection
Intrusion Detection Services
IP Networking layer
Intrusion Detection Services
IP Filtering
IPSec

Protect data in the network

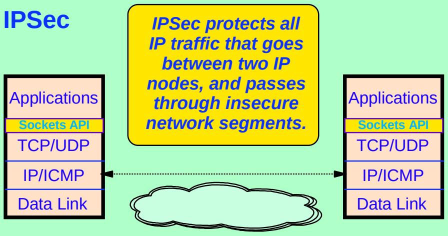
Examples of application protocols with built-in security extensions are SNMPv3, DNS, and OSPF.

Both Kerberos and SSL/TLS are located as extensions to the sockets APIs and applications have to be modified to make use of these security functions. Both SSL/TLS and Kerberos are connection-based and only applicable to TCP (stream sockets) applications, not UDP.

IPSec resides at the networking layer and is transparent to upper-layer protocols, including both transport layer protocol and application protocol.

Network Security
IPSec (VPNs) and SSL/TLS

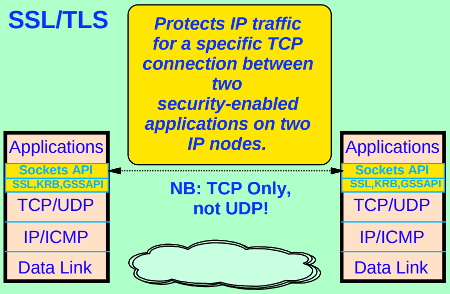
IPSec



IPSec protects all IP traffic that goes between two IP nodes, and passes through insecure network segments.

- Open network layer security protocol endorsed by IETF
- Provides authentication, integrity, and data privacy
- IP node authentication, not user authentication
- Allows secure tunnel between any two IP entities
- Management of crypto keys and security associations can be
  - manual
  - automated via key management protocol (IKE)
- Use of IPSec is transparent to upper layers including application
  - Blanket level protection for upper layer protocols

SSL/TLS

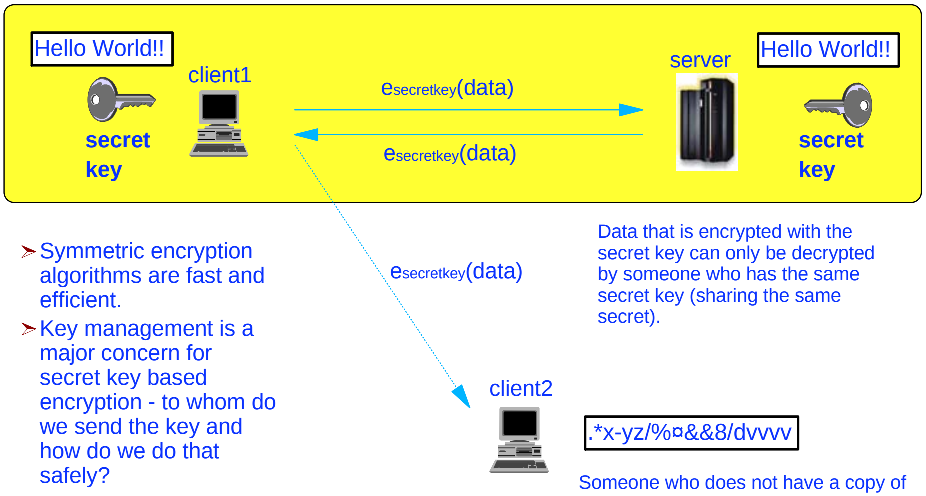


Protects IP traffic for a specific TCP connection between two security-enabled applications on two IP nodes.

NB: TCP Only, not UDP!

- Provides authentication, integrity, and data privacy above TCP layer.
  - Protocols include key exchange using public key cryptography and negotiation of security parameters
- User authentication is supported if client certificates are used
- Applications must be changed to use SSL/TLS API functions
- Some z/OS applications that use SSL/TLS:
  - z/OS HTTP Server
  - z/OS WebSphere Application Server
  - z/OS LDAP server
  - TN3270 Server
  - FTP server and client
  - Policy Agent (PAGENT)
  - Digital Certificate Authentication server (DCAS)
- z/OS provides a System SSL library for z/OS applications
  - Available to LE C/C++ sockets programs only

Symmetric encryption (secret or shared key encryption)

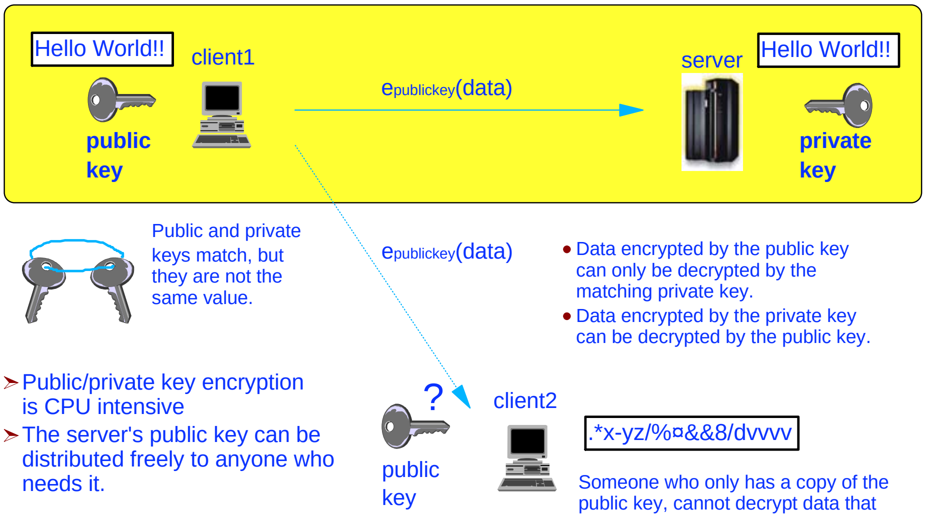


- Symmetric encryption algorithms are fast and efficient.
- Key management is a major concern for secret key based encryption - to whom do we send the key and how do we do that safely?

Data that is encrypted with the secret key can only be decrypted by someone who has the same secret key (sharing the same secret).

Someone who does not have a copy of the secret key cannot decrypt any intercepted data.

RSA public/private key encryption (asymmetric encryption)



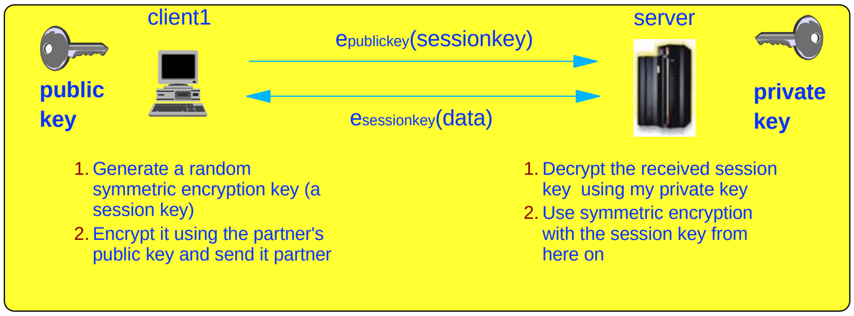
Public and private keys match, but they are not the same value.

- Data encrypted by the public key can only be decrypted by the matching private key.
- Data encrypted by the private key can be decrypted by the public key.

- Public/private key encryption is CPU intensive
- The server's public key can be distributed freely to anyone who needs it.

Someone who only has a copy of the public key, cannot decrypt data that was encrypted with the same public key.

Combining public/private key encryption with symmetric encryption - session keys



1. Generate a random symmetric encryption key (a session key)
2. Encrypt it using the partner's public key and send it partner

1. Decrypt the received session key using my private key
2. Use symmetric encryption with the session key from here on

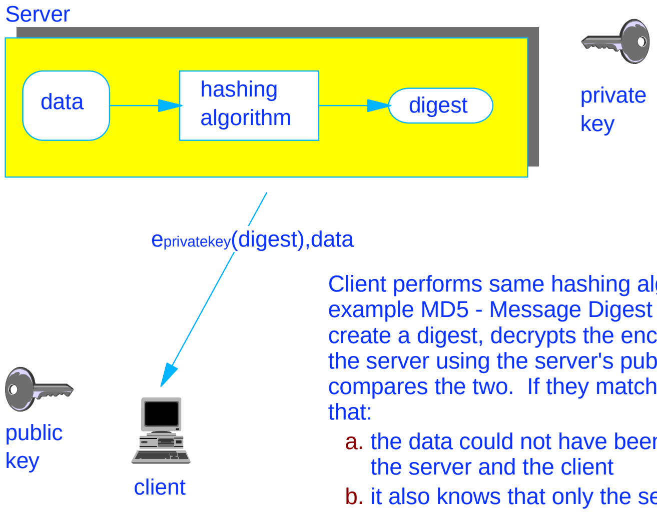
Secure Sockets Layer (SSL) uses RSA to generate symmetric session keys

1. Server has distributed a public key to client earlier.

2. Client generates a random session key, encrypts it under the server's public key, and transmits it to the server. Only the server is able to decrypt this message.

3. Server decrypts the message and the server and the client use the session key for succeeding encrypted data exchanges in this session.

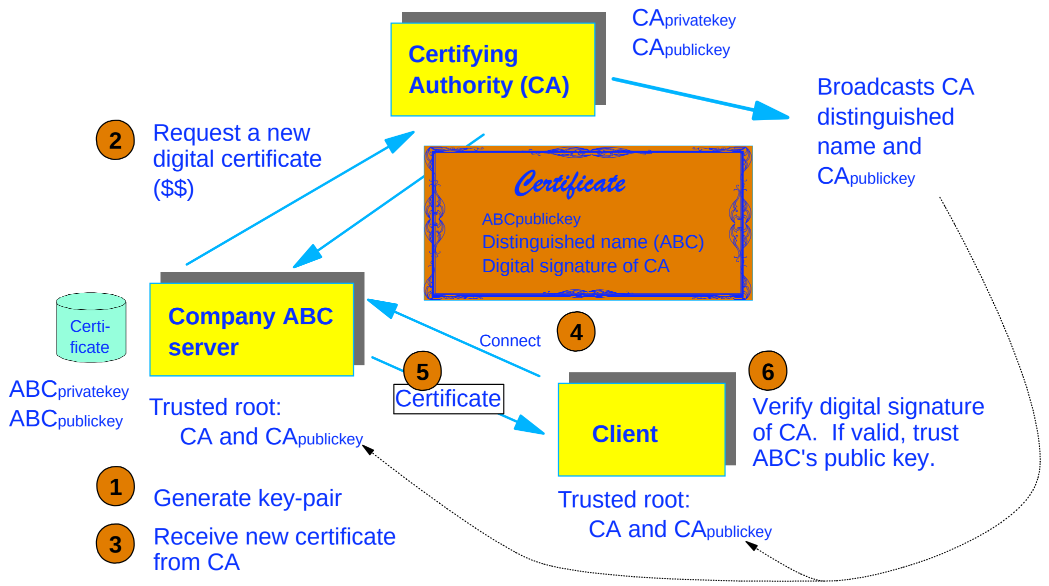
Digital signature / message authentication



Client performs same hashing algorithm (for example MD5 - Message Digest 5) on the data to create a digest, decrypts the encrypted digest from the server using the server's public key, and compares the two. If they match, the client knows that:

a. the data could not have been altered between the server and the client
b. it also knows that only the server with the right private key could have created the message.

X.509 certificates trust relationships



Certifying Authority (CA)

CAprivatekey
CAprivatekey

Broadcasts CA distinguished name and CApublickey

2 Request a new digital certificate ($$)

Certificate
ABCpublickey
Distinguished name (ABC)
Digital signature of CA

Certi-ficate

Company ABC server

ABCprivatekey
ABCpublickey

Trusted root:
CA and CApublickey

Connect 4

5 Certificate

Client

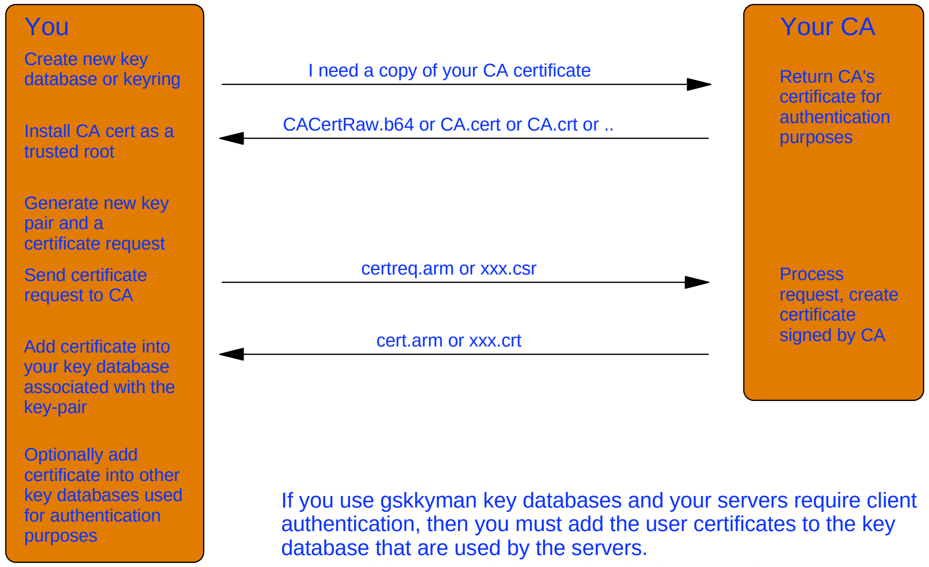
6 Verify digital signature of CA. If valid, trust ABC's public key.

Trusted root:
CA and CApublickey

1 Generate key-pair

3 Receive new certificate from CA

CA flow overview



You

Create new key database or keyring

Install CA cert as a trusted root

Generate new key pair and a certificate request

Send certificate request to CA

Add certificate into your key database associated with the key-pair

Optionally add certificate into other key databases used for authentication purposes

I need a copy of your CA certificate →

← CACertRaw.b64 or CA.cert or CA.crt or ..

certreq.arm or xxx.csr →

← cert.arm or xxx.crt

Your CA

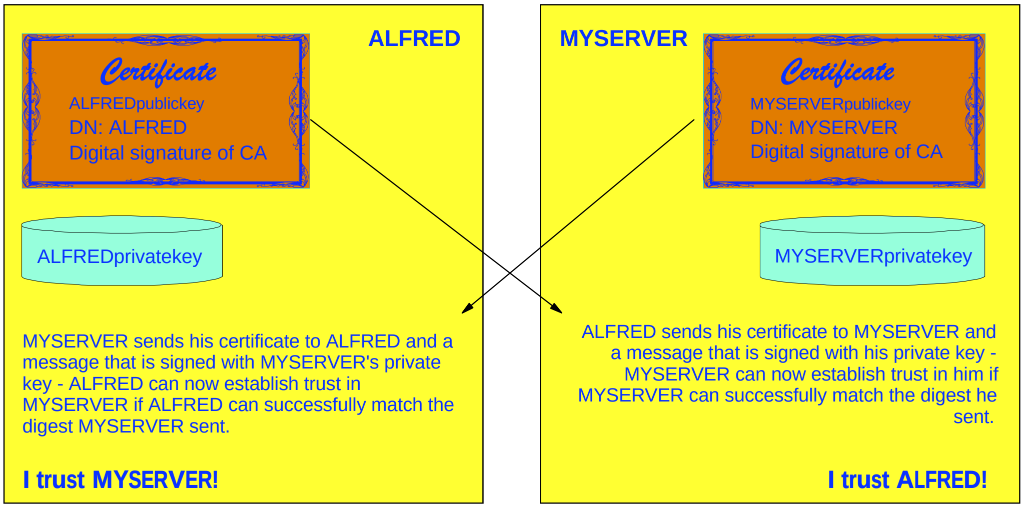
Return CA's certificate for authentication purposes

Process request, create certificate signed by CA

If you use gskkyman key databases and your servers require client authentication, then you must add the user certificates to the key database that are used by the servers.

In general, if you need client authentication, I would recommend using RACF keyrings.

Server and client certificates



MYSERVER sends his certificate to ALFRED and a message that is signed with MYSERVER's private key - ALFRED can now establish trust in MYSERVER if ALFRED can successfully match the digest MYSERVER sent.

I trust MYSERVER!

ALFRED sends his certificate to MYSERVER and a message that is signed with his private key - MYSERVER can now establish trust in him if MYSERVER can successfully match the digest he sent.

I trust ALFRED!

- In order to create session keys and encrypt the data stream, only the server needs to have a certificate.
- If the client has a certificate, then the server can use that to authenticate the client. On z/OS a client certificate can be mapped to a RACF userID. Password authentication may still be required by the server application, but the password can at least now be submitted in encrypted form over the network.

The main objectives

1. Allow the client application/user to be sure that he/she is connected to the server he/she expects to be connected to.
   - *Signed server certificates*
2. Allow the server application to be sure who the client user is before the user sees any data from the server or before the user enters any input.
   - *Signed client certificates*
3. Ensure data confidentiality from end-to-end (from end-user computer to server computer without having the data appearing in the clear on any part of the path between these two end-nodes).
   - *Encryption*
4. Ensure that data is not modified by some node in-between the two end-nodes.
   - *Message authentication*

System SSL on z/OS

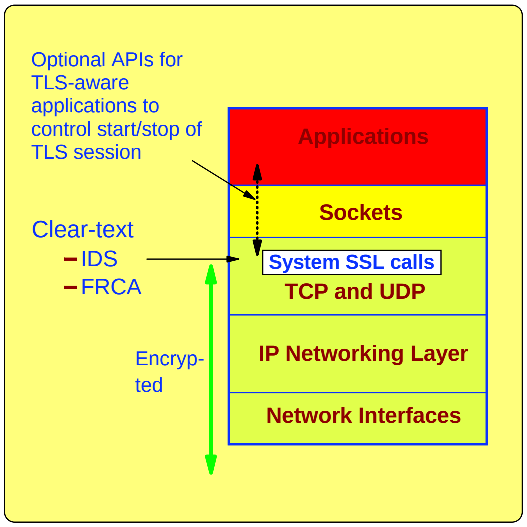
- Both the TN3270(E) server, the FTP server, and the FTP client on z/OS use the system SSL functions to implement secure sockets
- There are no definitions in any of the TCP/IP configuration files to control whether software or hardware encryption is to be used:
  - System SSL checks during SSL application initialization if ICSF is installed and active and if the hardware is enabled and available.
  - If hardware is available, it will be used.
  - If hardware is not available, software encryption will be used.
  - If hardware encryption is to be used, make sure the server started task user IDs and z/OS FTP client user IDs have access to the proper CSFSERV resources (used to control ICSF cryptographic functions) and optionally to CSFKEYS resources (used to protect cryptographic keys in the ICSF key data sets).
- The encryption algorithm for a secure connection is determined based on combining the list of algorithms system SSL supports, the list the server wants to use (can be configured for the z/OS FTP and TN3270(E) servers) and the list the client wants to use (can also be configured for the z/OS FTP client).
  - The order of algorithms is important: the first one listed that everyone agrees to will be used.
- If certificates are stored in RACF (or similar security product), all system SSL users must have READ access to:
  - IRR.DIGTCERT.LIST
  - IRR.DIGTCERT.LISTRING

System SSL notes

NOTES

- Diagnosing system SSL problems can be done by passing an environment variable to the server:
  - GSK_TRACE_FILE=/dir/rawtracefile
- The system SSL trace file can be formatted with
  - gsktrace /dir/rawtracefile > /dir/formattedtracefile
- All RACF certificate-related tasks can be performed using the RACF command interface (RACDCERT) or using the RACF ISPF interface.

CS z/OS V1R7: Application Transparent Transport Layer Security (TLS)



- **Basic TCP/IP stack-based TLS**
  - TLS process performed at TCP layer without requiring any application change (transparent)
  - All connections to specified port are designated as TLS required
  - Transparent TLS policies managed via Policy Agent
- **Transparent TLS can be requested by aplication**
  - Application issue transparent TLS API calls to indicate that connection should start/stop using TLS
- **TCP/IP stack-based TLS with client identification services for application**
  - Application issues TLS API calls to receive user identity information based on X.509 client certificate
- **Available to any TCP application**
  - CICS Sockets is primary focus of this support
  - All programming languages supported

Telnetting into a z/OS system

There are two telnet servers on z/OS:

1. **The TN3270, TN3270E, and SNA line mode telnet server**
2. **The UNIX line mode and raw mode telnet server**

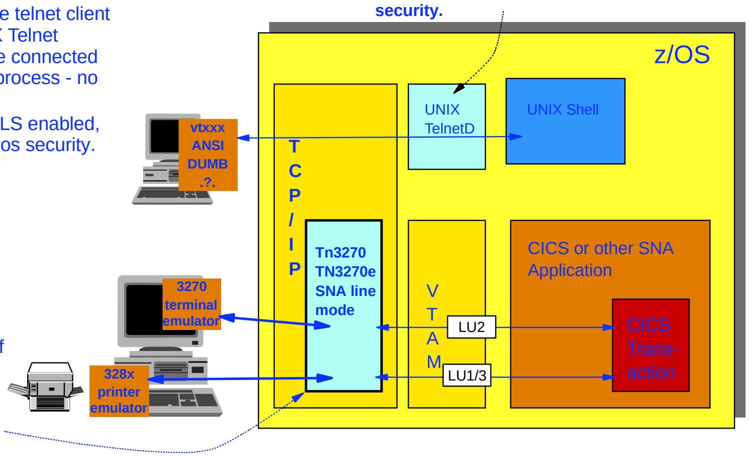
Remote terminal access

A line-mode or raw-mode telnet client can connect to the UNIX Telnet server. The client will be connected directly to a UNIX shell process - no VTAM involvement.

This server is not SSL/TLS enabled, but does support Kerberos security.

A line-mode or TN3270(E) telnet client can connect to the TN3270 server. The connection will be mapped to an SNA LU and an SNA session will be created by the help of VTAM.

This server supports Kerberos-based security.



This is the server that is SSL/TLS enabled.

TN3270(E) server - secure connection support since OS/390 V2R6 (September 1998)

OS/390 V2R6: SSL support with server certificates
OS/390 V2R8: Added support for client certificates
OS/390 V2R10: Added support for negotiable SSL
z/OS V1R4: Added TLS support

Multiple telnet server ports

- Up to 255 ports in one server

Individual telnet server ports defined as:

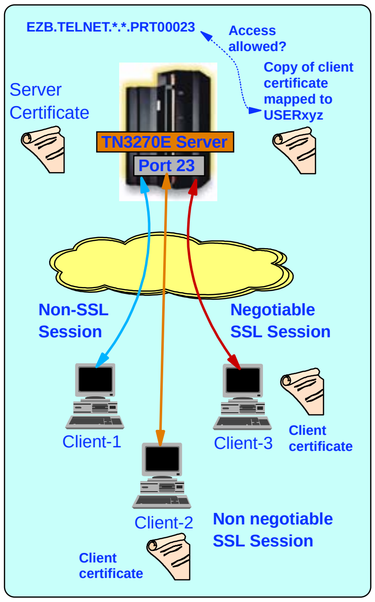
- **Secure** - must use SSL (non-negotiable. or negotiable)
- **NegtSecure** - must use negotiable SSL
- **Any** - all allowed (both SSL and non-SSL)
- **Basic** - no SSL

Access to telnet port optionally checked:

- EZB.TELNET.tcpname.PRTnnnnn - allows for an additional restriction on which authenticated clients can establish TN3270(E) sessions with a specific TN3270(E) server port number

Controlling individual client connections:

- The PARMSGROUP and PARMSMAP statements are, among other things, used to control which clients must use SSL, which clients may use SSL, and which clients are not allowed to use SSL at all.



TN3270(E) secure connection details

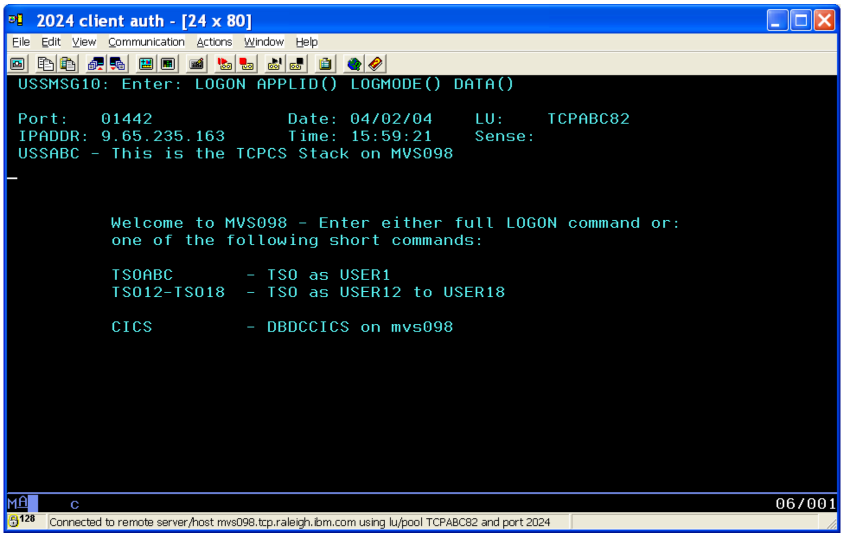
- **CONNTYPE SECURE**
  - tries a standard SSL handshake
  - if the SSL handshake times out, tries 'Negotiate SSL'
  - if the client rejects SSL, the connection is closed
- **CONNTYPE NEGTSECURE**
  - tries 'Negotiated SSL'
  - if the client rejects SSL, the connection is closed
- **CONNTYPE ANY**
  - tries a standard SSL handshake
  - if the SSL handshake times out, tries 'Negotiate SSL'
  - if the client rejects SSL, a basic (non-SSL) connection is used
- **CONNTYPE NONE - no connection allowed**
  - Used with PARMSMAP overrides per client or group of clients to specify different CONNTYPE requirements
- **CONNTYPE BASIC - non-SSL connection used**

- **CLIENTAUTH NONE**
  - Client certificate is not requested
- **CLIENTAUTH SSLCERT**
  - Client certificate is required
  - Client certificate has to be valid: signed by a known CA - known as **level 1** authentication
- **CLIENTAUTH SAFCERT**
  - Client certificate is required and signed by a known CA
  - Client certificate has to be known by RACF: map to a user ID - known as **level 2** authentication
  - Optionally user has to have access to EZB.TELNET.. profile in RACF - known as **level 3** authentication

- **KEYRING**
  - MVS data set name
  - HFS file name
  - SAF key ring name
- **Only one keyring is supported for a telnet server. All secure ports must use the same keyring definition.**

Additional security-related configuration options are **ENCRYPTION** to list the desired cipher-suites, **CRLLDAPSERVER** to point a Certificate Revocation List LDAP server to check for revoked certificates, and **SSLTIMEOUT** to define timeout value for SSL handshake processing.

Secure TN3270E session

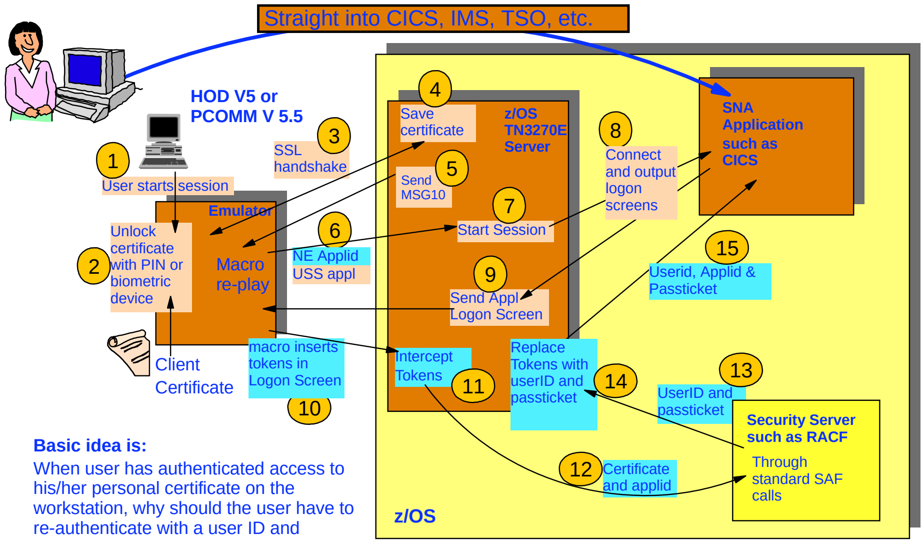

2024 client auth - [24 x 80]
File Edit View Communication Actions Window Help

```
USSMSG10: Enter: LOGON APPLID() LOGMODE() DATA()

Port:    01442              Date: 04/02/04    LU:    TCPABC82
IPADDR: 9.65.235.163        Time: 15:59:21    Sense:
USSABC - This is the TCPCS Stack on MVS098
_

            Welcome to MVS098 - Enter either full LOGON command or:
            one of the following short commands:

            TSOABC        - TSO as USER1
            TSO12-TSO18   - TSO as USER12 to USER18

            CICS          - DBDCCICS on mvs098
```

```
MA    c                                                          06/001
128   Connected to remote server/host mvs098.tcp.raleigh.ibm.com using lu/pool TCPABC82 and port 2024
```

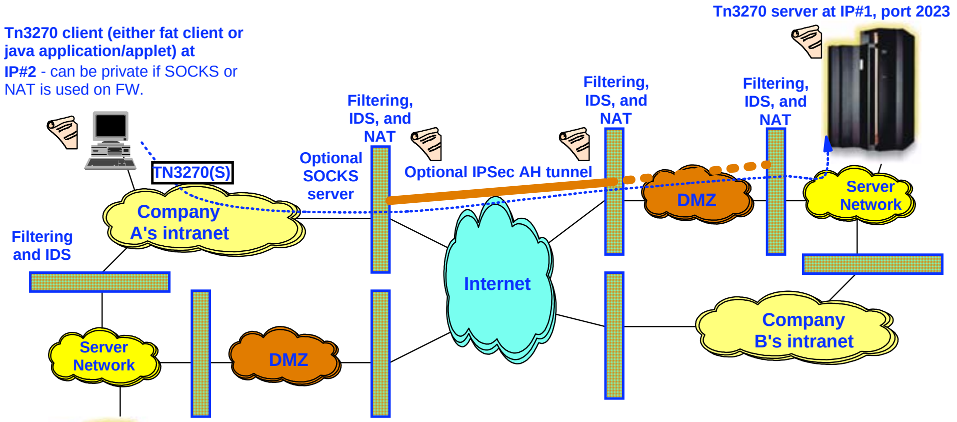
The lock indicates that this is a secure connection.

Express Logon - transparent security and authentication for TN3270e LU2 access


Straight into CICS, IMS, TSO, etc.
HOD V5 or PCOMM V 5.5
1
User starts session
2
Unlock certificate with PIN or biometric device
Emulator
Macro re-play
Client Certificate
3
SSL handshake
4
Save certificate
z/OS TN3270E Server
5
Send MSG10
6
NE Applid USS appl
7
Start Session
8
Connect and output logon screens
SNA Application such as CICS
9
Send Appl Logon Screen
10
macro inserts tokens in Logon Screen
11
Intercept Tokens
12
Certificate and applid
13
UserID and passticket
14
Replace Tokens with userID and passticket
15
Userid, Applid & Passticket
Security Server such as RACF
Through standard SAF calls
z/OS

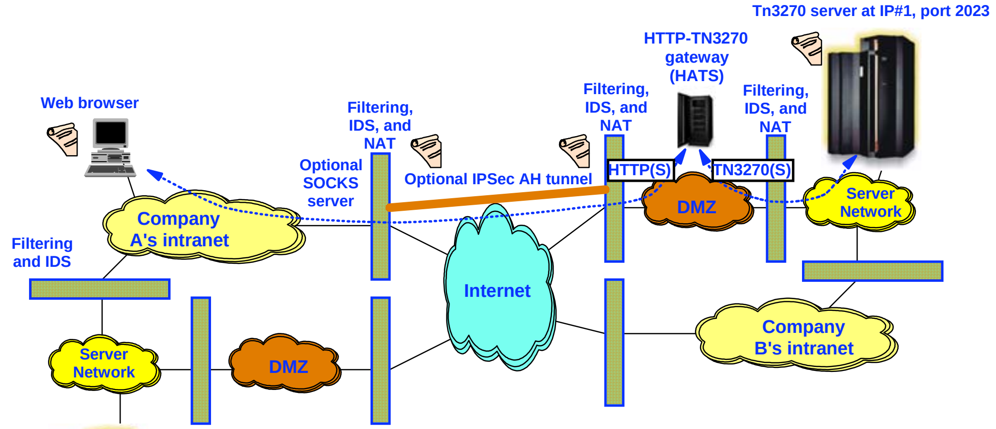
Basic idea is:
When user has authenticated access to his/her personal certificate on the workstation, why should the user have to re-authenticate with a user ID and password to access SNA applications on the host?

Business partners
TN3270 access sample configuration



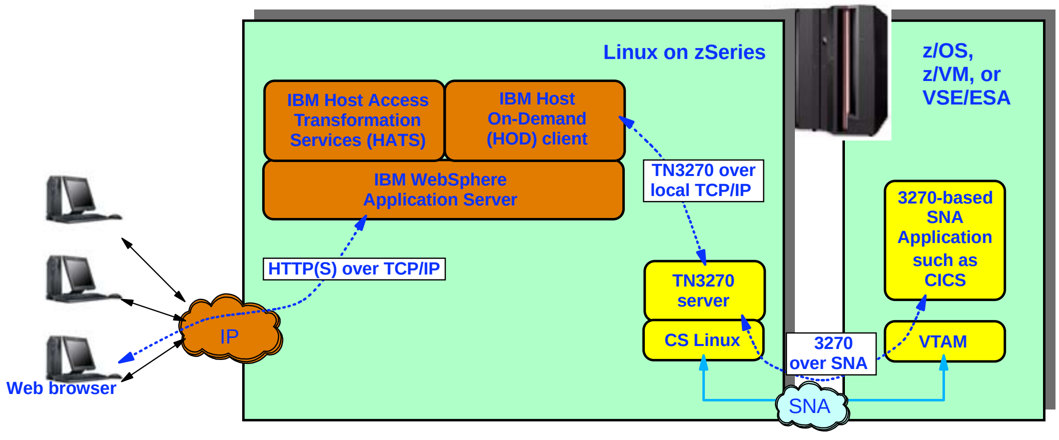
- Secure Tn3270 protocol can be NATed, Socksified, or relayed - no restrictions on use of firewall technologies
- Telnet port (23) needs to be opened in firewalls unless TN3270 over HTTP(S) is used
- End-user authentication can be done on server
- Server resource (port number) can be restricted to selected users (business partners)
- Data privacy and integrity end-to-end
- IPSec AH tunnel between firewalls allows FWs to authenticate with each other and reject traffic to TN3270 server from unauthorized partners (even in the case they fake their source IP addresses)

Business partners
TN3270 over HTTP(S) access sample configuration



- If firewalls cannot be opened for any other protocol than HTTP and HTTPS, there are today solutions that allow a web browser to communicate with a web server that acts as a gateway between HTTP(S) and TN3270(S)
- IBM's Host Access Transformation Services (HATS) product provides one such solution

Host Access Transformation Services


Linux on zSeries
z/OS,
z/VM, or
VSE/ESA
IBM Host Access Transformation Services (HATS)
IBM Host On-Demand (HOD) client
IBM WebSphere Application Server
TN3270 over local TCP/IP
3270-based SNA Application such as CICS
HTTP(S) over TCP/IP
TN3270 server
CS Linux
3270 over SNA
VTAM
IP
SNA
Web browser

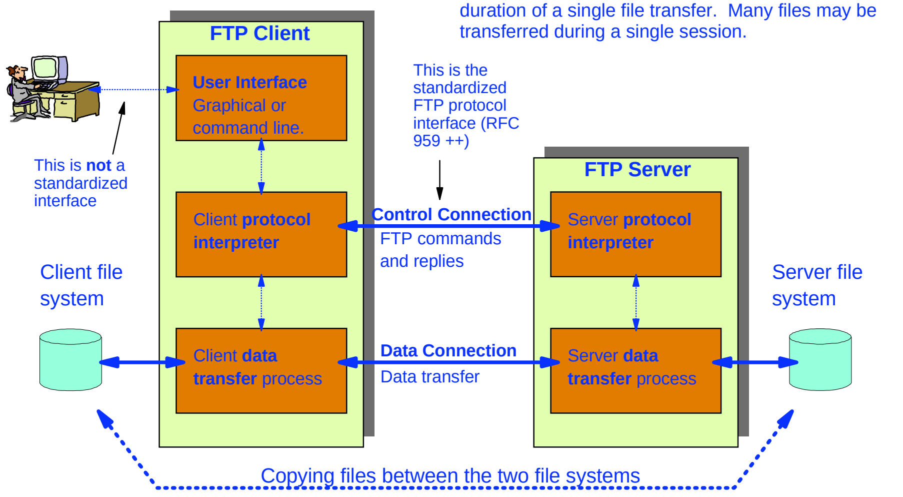
- ➢WebSphere Application Server and HATS can run on z/OS or, as in this example, on Linux for zSeries.
- ➢The TN3270 server does not have to reside on Linux, it could reside on z/OS even when HATS runs in Linux.

File Transfer Protocol (FTP) Overview

There is both an FTP client and an FTP server on z/OS. The client can be used from TSO, the UNIX shell or batch jobs.

An FTP session uses at least two TCP connections:

1. One for the ***control connection***. Server port is 21. This connection stays up during the whole session.
2. One for each data transfer. Active mode FTP uses server port 20 for ***data connections***. Stays up for the duration of a single file transfer. Many files may be transferred during a single session.


FTP Client
User Interface
Graphical or command line.
This is not a standardized interface
Client protocol interpreter
Client data transfer process
Client file system
This is the standardized FTP protocol interface (RFC 959 ++)
Control Connection
FTP commands and replies
Data Connection
Data transfer
FTP Server
Server protocol interpreter
Server data transfer process
Server file system
Copying files between the two file systems

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

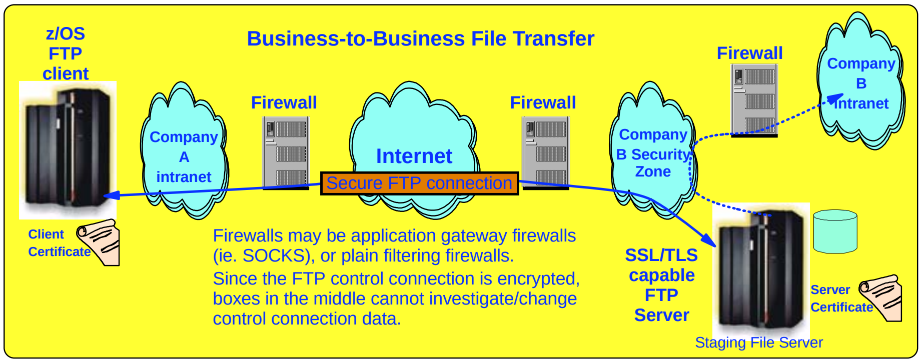
- Secure FTP refers to using the standard FTP protocol with one or more security extensions to improve security of data transfers
- "*FTP Security Extensions*", RFC2228 - defines a set of new FTP protocol commands and replies for negotiating secure FTP sessions.
- This RFC defines a framework for securing 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)
  - 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
  - (Just a real bad choice of name!)

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)

FTP support for secure transfers in z/OS V1R2

- The server can be started in one of two modes:
  - SSL/TLS required - only secure connections will be accepted
  - SSL/TLS allowed - secure connections will be established if the client requests it
- Server options specify whether a client certificate is required or optional
- If a connection is secure, the control connection will always be encrypted. Server options are used to control whether the data connections may or are required to be secure also.
- Most widely used workstation SSL/TLS enabled FTP software is WS-FTP Pro.
- The z/OS FTP server also supports Kerberos-based security.



Firewalls may be application gateway firewalls (ie. SOCKS), or plain filtering firewalls.

Since the FTP control connection is encrypted, boxes in the middle cannot investigate/change control connection data.

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.

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)

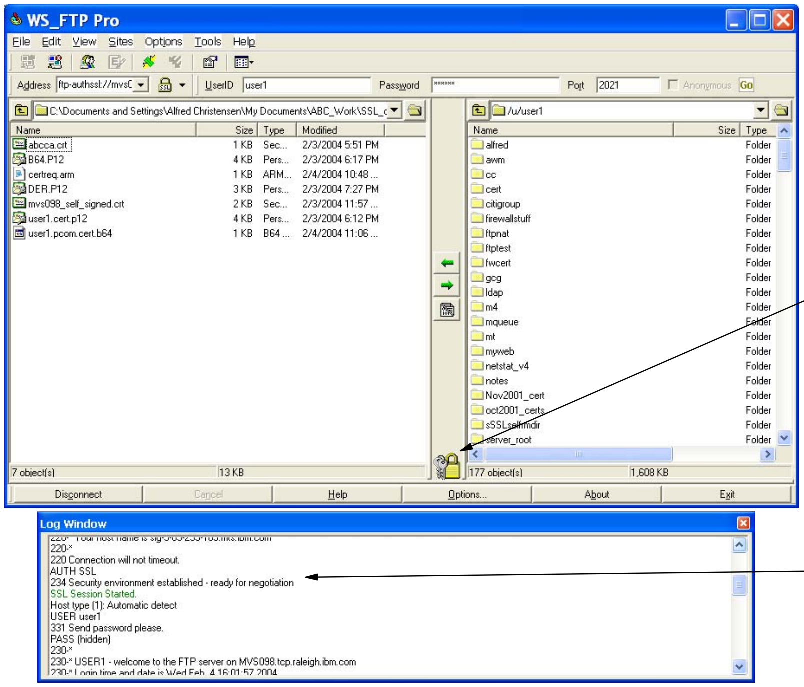
- **REQUIRED** - passwords are always required even if client authentication is used
- **OPTIONAL** - if a valid client certificate is presented, it matches a user definition in the security products, and that the security product user ID matches the user ID that is received on the FTP USER command - 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.

Secure FTP connection



Address: ftp-authssl://mvs0 UserID: user1 Password: ****** Port: 2021 □ Anonymous Go

C:\Documents and Settings\Alfred Christensen\My Documents\ABC_Work\SSL_c

Name	Size	Type	Modified
abcca.crt	1 KB	Sec...	2/3/2004 5:51 PM
B64.P12	4 KB	Pers...	2/3/2004 6:17 PM
certreq.arm	1 KB	ARM...	2/4/2004 10:48 ...
DER.P12	3 KB	Pers...	2/3/2004 7:27 PM
mvs098_self_signed.crt	2 KB	Sec...	2/3/2004 11:57 ...
user1.cert.p12	4 KB	Pers...	2/3/2004 6:12 PM
user1.pcom.cert.b64	1 KB	B64 ...	2/4/2004 11:06 ...

7 object(s) 13 KB

/u/user1

Name	Size	Type
alfred		Folder
awm		Folder
cc		Folder
cert		Folder
citigroup		Folder
firewallstuff		Folder
ftpnat		Folder
ftptest		Folder
fwcert		Folder
gcg		Folder
ldap		Folder
m4		Folder
mqueue		Folder
mt		Folder
myweb		Folder
netstat_v4		Folder
notes		Folder
Nov2001_cert		Folder
oct2001_certs		Folder
sSSLselfmdir		Folder
server_root		Folder

177 object(s) 1,608 KB

```
220-* Your host name is sig-9-65-235-165.mts.ibm.com
220-*
220 Connection will not timeout.
AUTH SSL
234 Security environment established - ready for negotiation
SSL Session Started.
Host type (1): Automatic detect
USER user1
331 Send password please.
PASS (hidden)
230-*
230-* USER1 - welcome to the FTP server on MVS098.tcp.raleigh.ibm.com
230-* Login time and date is Wed Feb  4 16:01:57 2004
```

The lock indicates a secure connection.

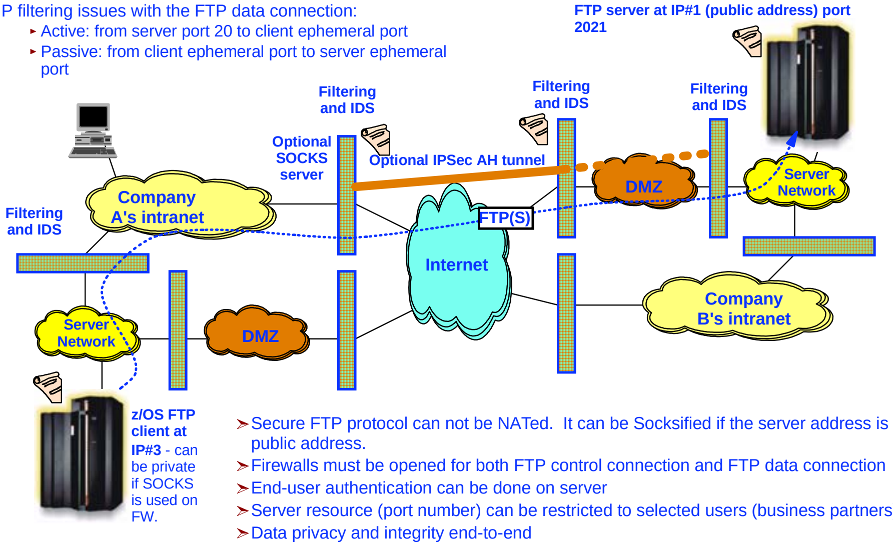
In the lower pane, you can see the AUTH SSL command being sent to the server and the positive 234 reply message.

z/OS secure batch FTP client session log

```
EZA1736I FTP
EZA1450I IBM FTP CS V1R2
EZA1466I FTP: using TCPCS
EZA1456I Connect to ?
EZA1736I mvs098.tcp.raleigh.ibm.com 2021 (exit
EZA1554I Connecting to: mvs098.tcp.raleigh.ibm.com 9.67.113.11 port: 2021.
220-FTPSEC1 IBM FTP CS V1R2 at MVS098.tcp.raleigh.ibm.com, 19:00:58 on 2002-02-0
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:USER2):
EZA1701I >>> USER user2
331 Send password please.
EZA1789I PASSWORD:
EZA1701I >>> PASS
230-Processing FTPS.RC configuration file - USER2.FTPS.RC
230-"USER2." is the working directory name prefix.
230 USER2 is logged on.  Working directory is "USER2.".
EZA1460I Command:
```

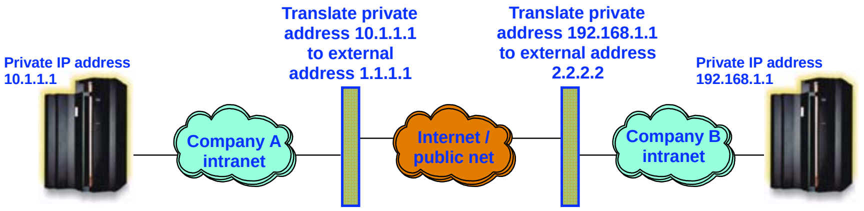
Business partners
FTP access sample configuration

IP filtering issues with the FTP data connection:
- Active: from server port 20 to client ephemeral port
- Passive: from client ephemeral port to server ephemeral port



- Secure FTP protocol can not be NATed. It can be Socksified if the server address is a public address.
- Firewalls must be opened for both FTP control connection and FTP data connection
- End-user authentication can be done on server
- Server resource (port number) can be restricted to selected users (business partners)
- Data privacy and integrity end-to-end
- IPSec AH tunnel between firewalls allows FWs to authenticate with each other and reject traffic to FTP server from unauthorized partners (even in the case they fake their source IP addresses)

Secure FTP and issues with Network Address Translation (NAT) and filtering firewalls today


Private IP address 10.1.1.1
Translate private address 10.1.1.1 to external address 1.1.1.1
Translate private address 192.168.1.1 to external address 2.2.2.2
Private IP address 192.168.1.1
Company A intranet
Internet / public net
Company B intranet

ftp 2.2.2.2

src=10.1.1.1 dest=2.2.2.2 → src=1.1.1.1 dest=2.2.2.2 → src=1.1.1.1 dest=192.168.1.1 →

← src=2.2.2.2 dest=10.1.1.1 ← src=2.2.2.2 dest=1.1.1.1 ← src=192.168.1.1 dest=1.1.1.1

Active:
PORT 10.1.1.1
1026

Will try to connect back to 10.1.1.1 which won't work.

Passive:
PASV

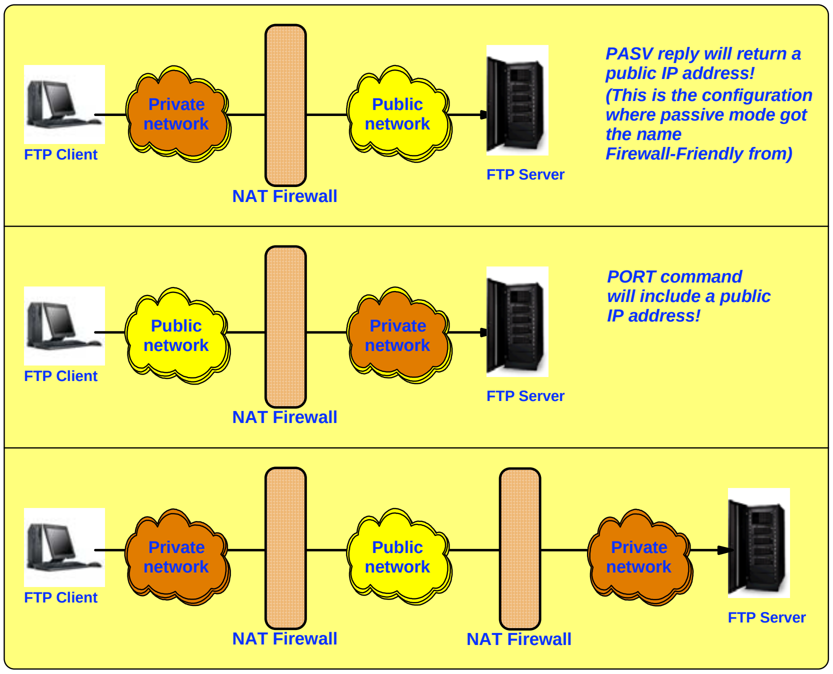
227 Entering Passive Mode
192.168.1.1 1027

Will try to connect back to 192.168.1.1 which won't work.

When firewalls do NAT, they rely on being able to snoop on the content of the FTP control connection and to modify the IP addresses in the PORT command or the reply to the PASV command. When the connection is secure (either because of use of SSL/TLS, Kerberos, or IPSec) that ability doesn't exist!

Another related issue with passive mode is that the server will choose an ephemeral port number for the data connection, which means that the data connection will come from an ephemeral port number to an ephemeral port number (not a nice rule to have in a firewall!).

Which secure FTP NAT configurations may currently work and which will not?



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

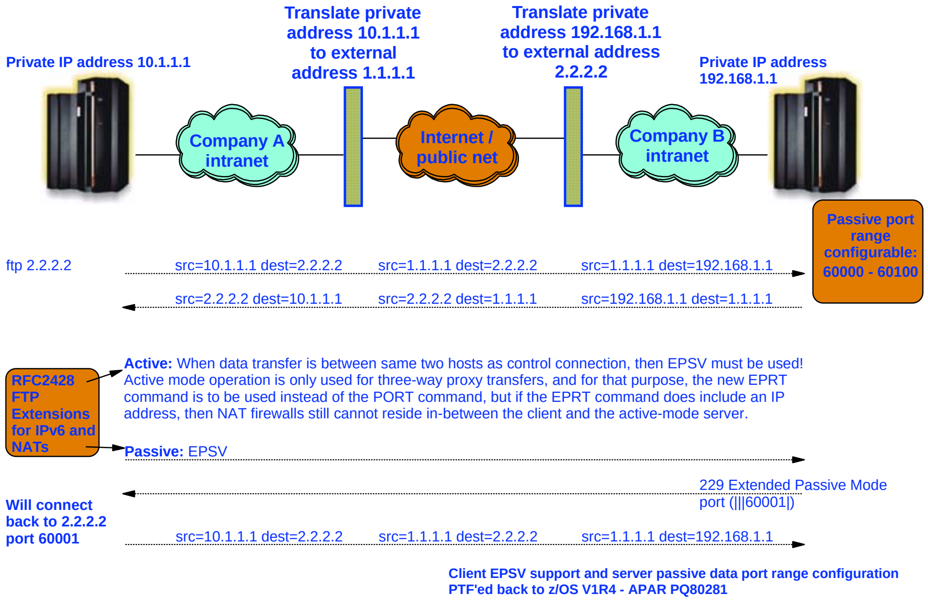
- Passive mode (PASV) will work
- Active mode (PORT) will not work

PORT command will include a public IP address!

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

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

Secure FTP and Network Address Translation (NAT) firewalls in z/OS V1R5


Private IP address 10.1.1.1
Translate private address 10.1.1.1 to external address 1.1.1.1
Translate private address 192.168.1.1 to external address 2.2.2.2
Private IP address 192.168.1.1
Company A intranet
Internet / public net
Company B intranet
Passive port range configurable: 60000 - 60100
ftp 2.2.2.2
src=10.1.1.1 dest=2.2.2.2
src=1.1.1.1 dest=2.2.2.2
src=1.1.1.1 dest=192.168.1.1
src=2.2.2.2 dest=10.1.1.1
src=2.2.2.2 dest=1.1.1.1
src=192.168.1.1 dest=1.1.1.1
RFC2428 FTP Extensions for IPv6 and NATs
Passive: EPSV
229 Extended Passive Mode port (|||60001|)
Will connect back to 2.2.2.2 port 60001
src=10.1.1.1 dest=2.2.2.2
src=1.1.1.1 dest=2.2.2.2
src=1.1.1.1 dest=192.168.1.1

Active: When data transfer is between same two hosts as control connection, then EPSV must be used! Active mode operation is only used for three-way proxy transfers, and for that purpose, the new EPRT command is to be used instead of the PORT command, but if the EPRT command does include an IP address, then NAT firewalls still cannot reside in-between the client and the active-mode server.

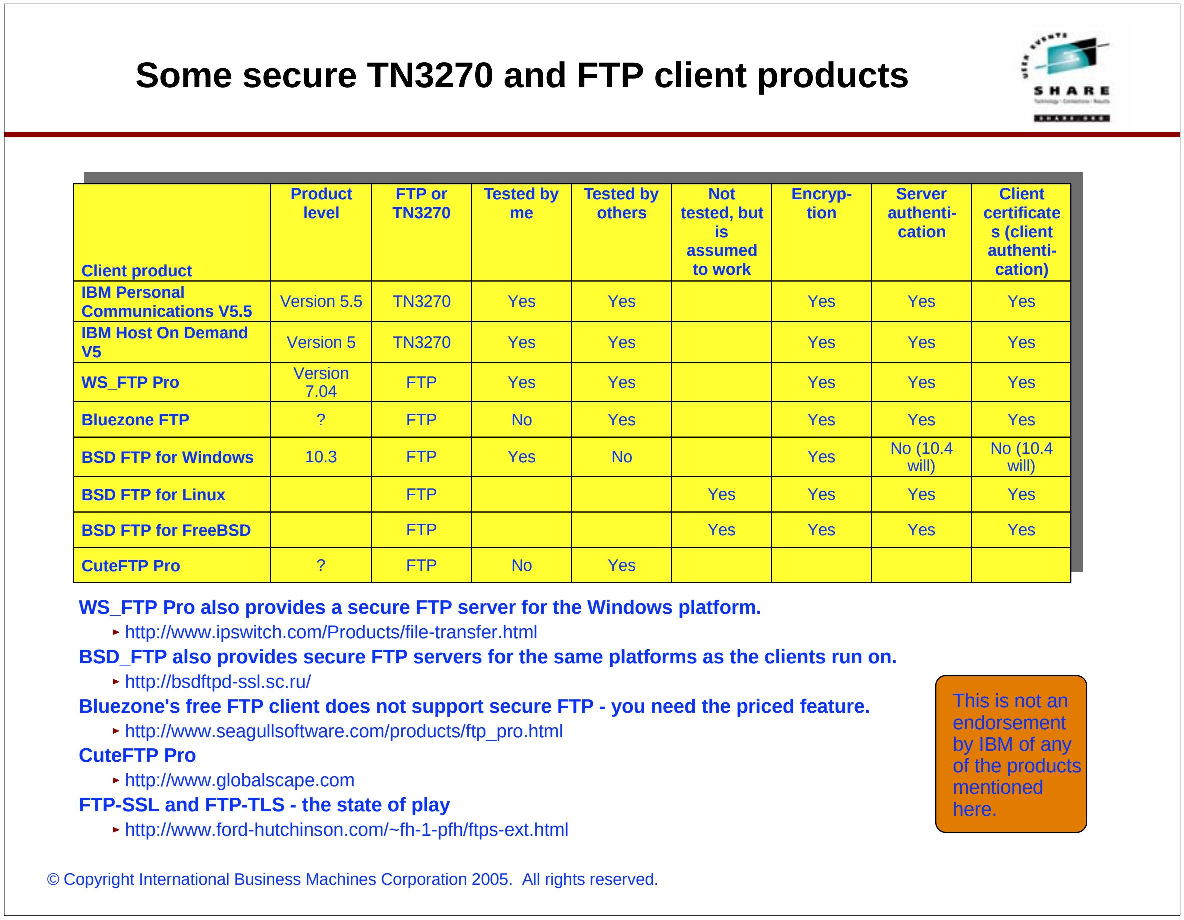
Client EPSV support and server passive data port range configuration PTF'ed back to z/OS V1R4 - APAR PQ80281

z/OS FTP client use of extended passive mode

- z/OS FTP client can be configured to use EPSV for IPv4 always via a client FTP.DATA option:
  - EPSV4 [TRUE | FALSE]
- The z/OS FTP client end-user can switch use of EPSV on/off via LOCSITE commands:
  - LOCSITE [NO]EPSV4
- The z/OS FTP server understands an EPSV command and will act accordingly

```
Command:
locsite EPSV4
Command:
dir t*
>>> EPSV
229 Entering Extended Passive Mode (|||50000|)
>>> LIST t*
125 List started OK
Volume Unit    Referred Ext Used Recfm Lrecl BlkSz Dsorg Dsname
CPDLB3 3390   2002/11/20  1    1  VB      256  6233  PO  TEST.PDS
250 List completed successfully.
Command:
locsite noepsv4
Command:
dir t*
>>> PASV
227 Entering Passive Mode (9,xx,yyy,zz,195,81)
    ... and so on
```

- If an FTP server doesn't understand the EPSV command, it will respond with a "500 Unknown Command" reply.
- The z/OS FTP client will then revert to non extended mode and send either a PASV or a PORT command to that server



Some secure TN3270 and FTP client products

Client product	Product level	FTP or TN3270	Tested by me	Tested by others	Not tested, but is assumed to work	Encryp-tion	Server authenti-cation	Client certificates (client authenti-cation)
IBM Personal Communications V5.5	Version 5.5	TN3270	Yes	Yes		Yes	Yes	Yes
IBM Host On Demand V5	Version 5	TN3270	Yes	Yes		Yes	Yes	Yes
WS_FTP Pro	Version 7.04	FTP	Yes	Yes		Yes	Yes	Yes
Bluezone FTP	?	FTP	No	Yes		Yes	Yes	Yes
BSD FTP for Windows	10.3	FTP	Yes	No		Yes	No (10.4 will)	No (10.4 will)
BSD FTP for Linux		FTP			Yes	Yes	Yes	Yes
BSD FTP for FreeBSD		FTP			Yes	Yes	Yes	Yes
CuteFTP Pro	?	FTP	No	Yes				

WS_FTP Pro also provides a secure FTP server for the Windows platform.

- http://www.ipswitch.com/Products/file-transfer.html

BSD_FTP also provides secure FTP servers for the same platforms as the clients run on.

- http://bsdftpd-ssl.sc.ru/

Bluezone's free FTP client does not support secure FTP - you need the priced feature.

- http://www.seagullsoftware.com/products/ftp_pro.html

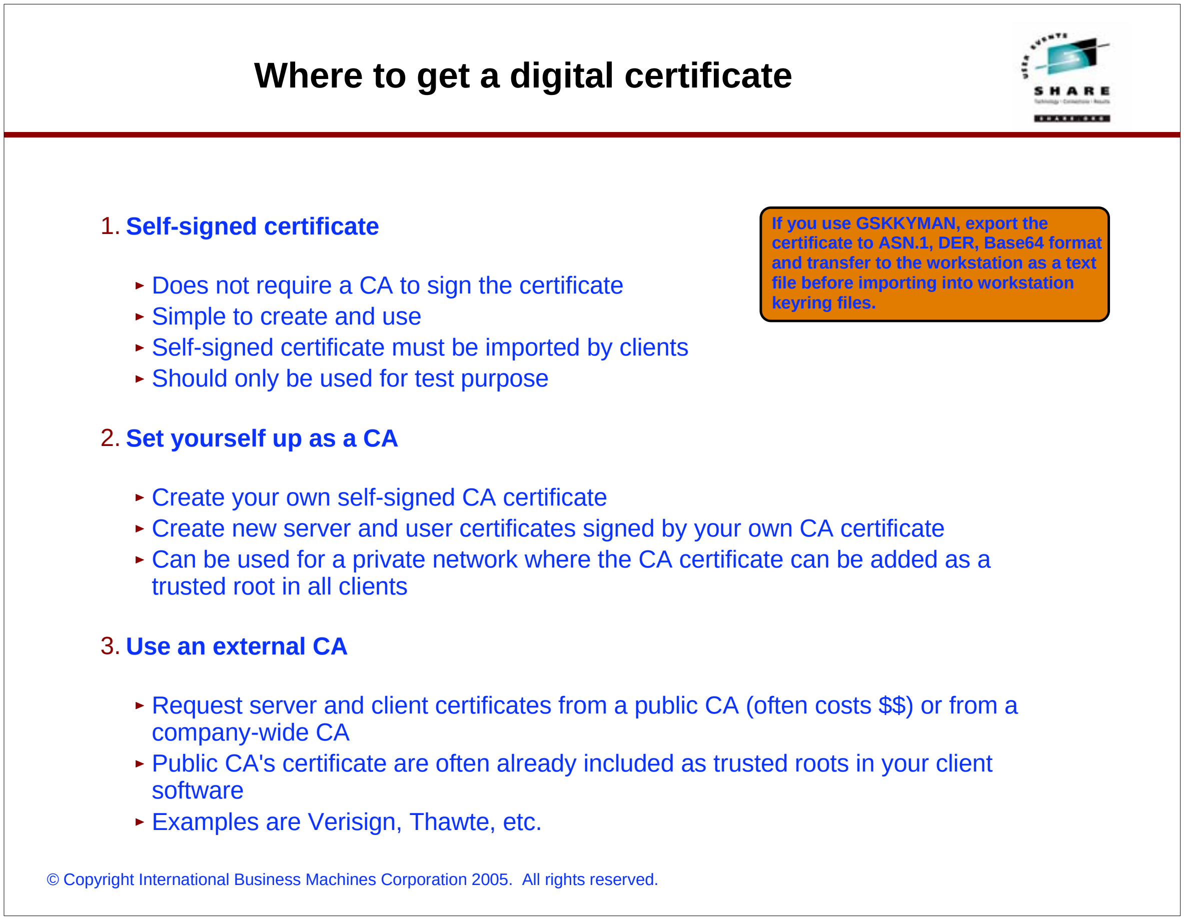
CuteFTP Pro

- http://www.globalscape.com

FTP-SSL and FTP-TLS - the state of play

- http://www.ford-hutchinson.com/~fh-1-pfh/ftps-ext.html

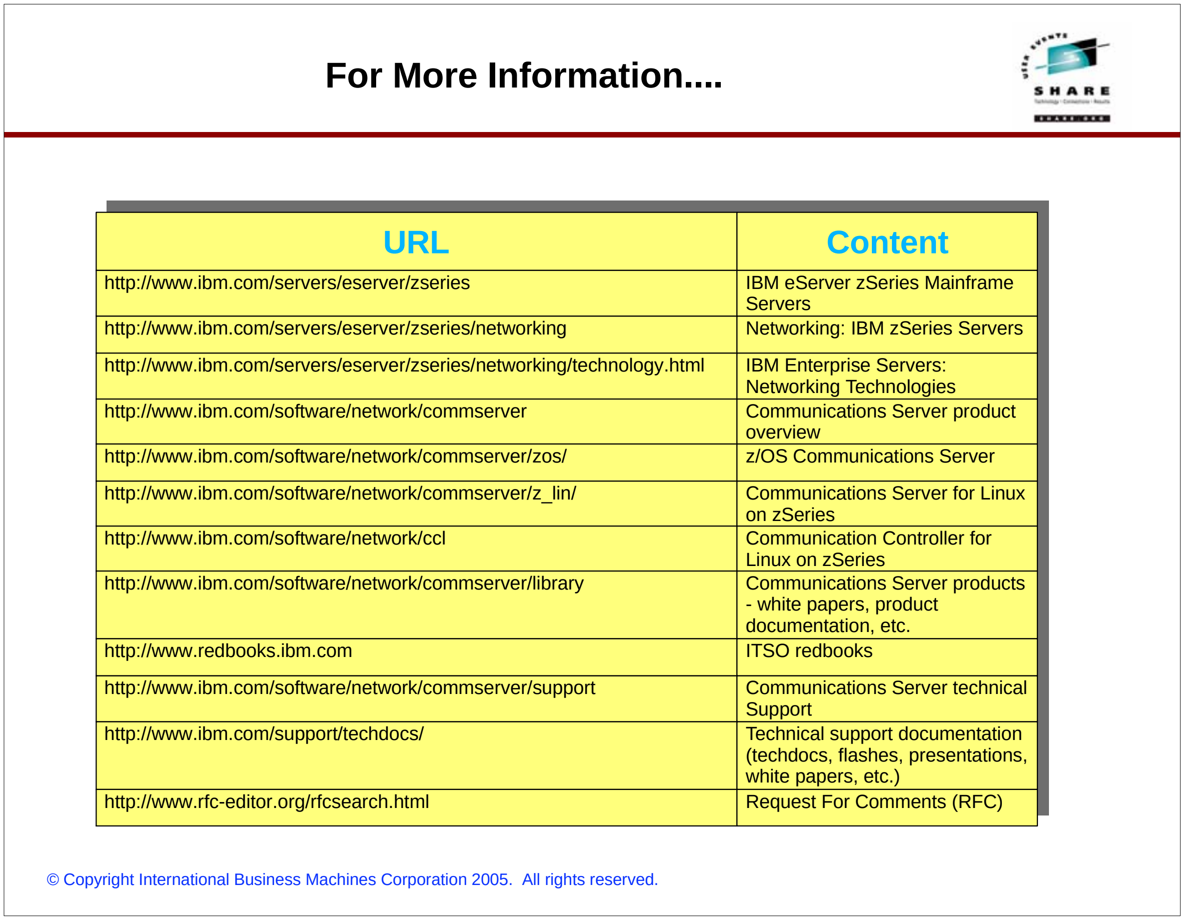
This is not an endorsement by IBM of any of the products mentioned here.



Where to get a digital certificate

If you use GSKKYMAN, export the certificate to ASN.1, DER, Base64 format and transfer to the workstation as a text file before importing into workstation keyring files.

1. **Self-signed certificate**
   - Does not require a CA to sign the certificate
   - Simple to create and use
   - Self-signed certificate must be imported by clients
   - Should only be used for test purpose
2. **Set yourself up as a CA**
   - Create your own self-signed CA certificate
   - Create new server and user certificates signed by your own CA certificate
   - Can be used for a private network where the CA certificate can be added as a trusted root in all clients
3. **Use an external CA**
   - Request server and client certificates from a public CA (often costs $$) or from a company-wide CA
   - Public CA's certificate are often already included as trusted roots in your client software
   - Examples are Verisign, Thawte, etc.



For More Information....

URL	Content
http://www.ibm.com/servers/eserver/zseries	IBM eServer zSeries Mainframe Servers
http://www.ibm.com/servers/eserver/zseries/networking	Networking: IBM zSeries Servers
http://www.ibm.com/servers/eserver/zseries/networking/technology.html	IBM Enterprise Servers: Networking Technologies
http://www.ibm.com/software/network/commserver	Communications Server product overview
http://www.ibm.com/software/network/commserver/zos/	z/OS Communications Server
http://www.ibm.com/software/network/commserver/z_lin/	Communications Server for Linux on zSeries
http://www.ibm.com/software/network/ccl	Communication Controller for Linux on zSeries
http://www.ibm.com/software/network/commserver/library	Communications Server products - white papers, product documentation, etc.
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/	Technical support documentation (techdocs, flashes, presentations, white papers, etc.)
http://www.rfc-editor.org/rfcsearch.html	Request For Comments (RFC)