My Adventures in PCI: FTP, TLS and SMF

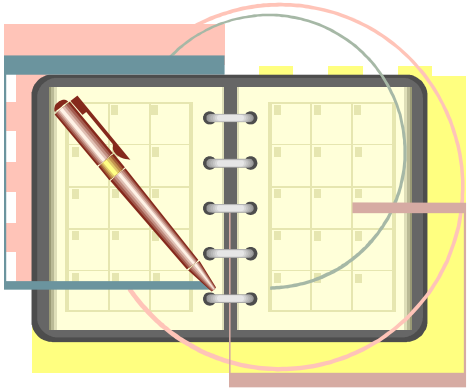
Joel Tilton
Senior Mainframe Security Engineer
www.linkedin.com/in/joeltilton/

Foreword

- My personal thanks to the NYRUG for allowing me to present on this topic.
- This presentation has been modified from its original version, it has been formatted to fit your needs.
  - After my original presentation of the material to the NYRUG on Tuesday March 12th 2013 I further enhanced it from the version originally shown.
    - Questions about MD5 are addressed (yes it's deprecated)
    - as well as steps needed to take to get TLS v1.2 installed for stronger cryptography.
    - Will SHA1 be deprecated?
      How to perform an SSL Trace
    - And more!

Agenda

- The overall goal of this presentation is to give a better understanding of the security related z/OS FTP and TCPIP SMF & encryption settings
- While you might not be an IP expert hopefully this presentation will expose you to new security concepts or perhaps make you think of new things that you can take back to help your environment
- Then when the real world comes knocking on your door for answers some day on this topic you'll have this presentation as reference



Why me?!?!

- That's what I first thought when my phone rang and I was asked to dive deep into an FTP configuration
- "I'm a RACF Engineer" I thought at first "why me?"
- But I have to be more than "just a RACF Engineer"
- I'm a Mainframe Security Engineer!
  - I should be capable of learning and understanding the security implications of any piece of technology that runs on a mainframe.
  - RTFM, PMRs to IBM (hey level 2 misses you open more PMRs☺), peer network, MacGyver your way to a solution!
- Questions are the beginning of wisdom

Why audit the z/OS FTP Server or TCPIP via SMF?

- How do you know who is moving your data around? Or stealing it?
- Are you 100% certain **every** dataset profile is securing your data?
- PCI requires an audit trail:
- Requirement 10: Track and monitor all access to network resources and cardholder data
- A.1.3 Ensure logging and audit trails are enabled and unique to each entity's cardholder data environment and consistent with PCI DSS Requirement 10.
- A.1.3 Verify the shared hosting provider has enabled logging as follows, for each merchant and service provider environment:
  - Logs are enabled for common third-party applications.
  - Logs are active by default.
  - Logs are available for review by the owning entity.
  - Log locations are clearly communicated to the owning entity

How is the z/OS FTP Server configured?

- Take your systems programmer out to lunch
  - (I mean a good lunch not McDonald's!!)
- Find the FTP.DATA file
  - Depends on where your sysprog decided to put it (ask them! See previous bullet)
  - Often stored in a parmlib under TCPIP or SYS1 qualifiers
  - Or search the STARTED class for entries that have FTP in them and go digging through all your parmlibs to find it
    - sr class(started) filter(*ftp*.**)
    - Yes I assuming or *hoping* the proc name will at least have FTP in it
- Now search all of your proc libs for that proc name
- Then look for the SYSFTPD DD card and note the dataset name
- While here make not of the dataset specified by the SYSTCPD DD card as well

SMF type 119 – My Secret Weapon

- SMF type 119 records rock (yes they really do!)
  - Was the control connection encrypted?
  - Was the data connection encrypted?
  - Version of SSL used
    - **PCI requires that SSL v2 or earlier is no longer used**
  - What cryptographic algorithm was used
    - **PCI requires a 128-bit or greater cipher strength**
  - IP addresses recorded in IPv6 format only
  - Appendix E of the z/OS Communications Server IP Programmer's Guide and Reference
    - **http://publibfp.dhe.ibm.com/cgi-bin/bookmgr/BOOKS/F1A1D3B1/E.0?DT=20120118013946**
- SMF type 118 do **not** record any of the information above required for PCI compliance
  - SMF type 118 records have been "stabilized"

Is the z/OS FTP Server recording SMF records?

- Search for all occurrences of “SMF” in the location of the FTP.DATA specified in the SYSFTPD DD card we found earlier.
- There are the three parameters that should be set for SMF recording to occur by the z/OS FTP server:
  - SMF TYPE119
  - SMFJES TYPE119
  - SMFSQL TYPE119
- PCI requires an audit trail (Requirement 9 and A.1.3)
- Wouldn’t you want to log who’s moving your data around using FTP?

Type 119 subtype 70 – FTP Server SMF Record

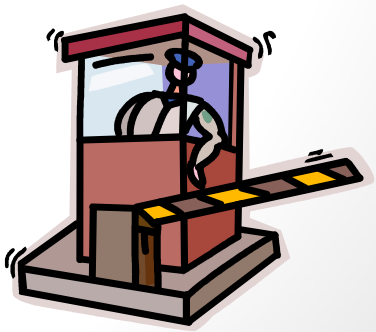
- Be **extremely** careful not to record **both** type 118 and 119 records
  - Doing so **will** create a performance problem!
  - "Records of type 118 and type 119 can both be requested; however, do not do this due to performance implications of writing both record types. "
  - http://publib.boulder.ibm.com/infocenter/zos/v1r13/index.jsp?topic=%2Fcom.ibm.zos.r13.halz001%2Fcsmfsta.htm
- **SMF TYPE119**
  - Cut one FTP type 119 subtype 70 record for every MVS and z/OS UNIX dataset transferred
  - For PDS cut one SMF record *per member* of the PDS
- **SMFSQL TYPE119**
    - SQL commands can be sent directly to your DB2 subsystem via FTP!!
    - If the "DB2 *subsystem_id"* statement **not** specified then the z/OS FTP server assumes a DB2 ssid of simply "DB2"
  - http://publib.boulder.ibm.com/infocenter/zos/v1r13/index.jsp?topic=%2Fcom.ibm.zos.r13.halz001%2Fcsmfsql.htm&path=8_6_20_137

FTP can transfer JES spool datasets too!

- **SMFJES TYPE119**
  - Cut one FTP type 119 subtype 70 record for every JESSPOOL dataset transferred
- **Can you even FTP datasets from the spool?**
  - Check your z/OS FTP FTP.DATA configuration file (SYSFTPD DD card)!
  - Jesinterfacelevel 0 = I can not FTP jobs from the spool ever
  - Jesinterfacelevel 1 = I can only FTP jobs that I own
    - **Jesinterfacelevel 1 This is the IBM shipped default**
  - Jesinterfacelevel 2 = I can transfer any jobs I want unless…
    - Uses SDSF class for granularity of control over job
    - JESSPOOL class is active with profiles defined to it to secure spool datasets
- **http://publib.boulder.ibm.com/infocenter/zos/v1r13/index.jsp?topic=%2Fcom.ibm.zos.r13.halz001%2Fcjesint.htm**
- **http://publib.boulder.ibm.com/infocenter/zos/v1r13/index.jsp?topic=/com.ibm.zos.r13.halu001/jesintdiff.htm**

JESINTERFACELEVEL 2 & SAF

- Some people read the manuals and see “SAF” next to JESINTERFACELEVEL 2
- Then they think “SAF” = external security = we’re safe; yes we want this setting.
- “Sorry wrong answer would you like to go for double Jeopardy where the scores can really change?”
- When you set JESINTERFACELEVEL 2 you need to be **absolutely** sure of the security implications



JESINTERFACLEVEL 2 Security Implications

- In my not so humble opinion you should **not** set JESINTERFACELEVEL 2 **unless**:
- The JESSPOOL class is not only active but configured to secure **all** spool datasets
  - Reminder JESSPOOL is a default RC of 8 class!
- The SDSF class is active with the following SAF resources secured appropriately:
  - ISFCMD.DSP.ACTIVE.*jesx*
  - ISFCMD.DSP.INPUT.*jesx*
  - ISFCMD.DSP.OUTPUT.*jesx*
  - ISFCMD.FILTER.OWNER
  - ISFCMD.FILTER.PREFIX
  - Reminder SDSF is a default RC of 4 class!
- **http://publib.boulder.ibm.com/infocenter/zos/v1r13/index.jsp?topic=%2Fcom.ibm.zos.r13.isfa500%2Fisf4csb082.htm**

How might JESINTERFACELEVEL 2 be abused?

- Ingenious application developers punch jobs straight to JES via FTP! ☹
  - Bypass production control job scheduling process
- Transfer sensitive jobs to your workstation with sensitive data (PCI, HIPAA, payroll, etc.)
- Again my take is if JESSPOOL & SDSF classes are not configured appropriately then JESINTERFACELEVEL should **never** be set to 2

Type 119 subtype 72 – FTP logon failure

- Captures why the FTP logon failed but also…
- Was it encrypted?
- With what version of SSL?
- With what type of algorithm?
- Helpful for tracking end users still attempting to use unencrypted FTP
- Validate someone is not trying to repeatedly login to breach accounts via FTP
- PCI requires an audit trail (Requirement 9 and A.1.3)

Find the TCPIP PROFILE dataset

- The FTP client SMF type 119 subtype 3 records are configured in the TCPIP stack itself
- Find the running TCPIP started task using SDSF, Sysview, etc.
- Browse it and do a find for ‘profile’
- Should see a message similar to:
  - EZZ0300I OPENED PROFILE FILE DD:PROFILE
- Read through the output until you find the PROFILE DD card
- Go browse that dataset or member
- Other options: Just use zSecure’s RE.I menu to validate the entire configuration of TCPIP including which SMF records are enabled for cutting

Validate TCPIP stack SMF logging

- Bring up the TCPIP PROFILE dataset in browse or view mode and do a search for ‘SMFCONFIG’
- If nothing is found then the TCPIP stack is **not** configured to cut **any** SMF records! ☹
  - Hopefully that is not the case
- PCI requires an audit trail (Requirement 9 and A.1.3)

Type 119 subtype 3 – FTP CLIENT SMF record

- In order to cut type 119 subtype 3 SMF records for all FTPs where z/OS is the client (or all OUTBOUND FTPs) the following needs to be added to the TCPIP profile parms:
- **SMFCONFIG TYPE119 FTPCLIENT**
- This is a change that of course requires assistance from systems programming!
  - Did I mention taking your systems programmer to lunch?
- Can be changed dynamically (OBEY file) or with an IPL (cycle of TCPIP)

A parting thought about other TCPIP SMF records

- SMFCONFIG TYPE119
  - TN3270CLIENT
    - Logs outbound telnet connections to other systems
  - PROFILE
    - Logs any changes to the configuration of the TCPIP stack
  - TCPSTACK
    - Logs useful information every time a TCPIP stack is started or stopped
  - **http://publib.boulder.ibm.com/infocenter/zos/v1r13/index.jsp?topic=/com.ibm.zos.r13.halz001/smfcfg.htm**

A parting thought about other TCPIP SMF records (continued)

- SMFCONFIG TYPE119
- TCPTERM
  - Cut a record every time a TCP connection closes
  - Logs version of SSL used (PCI won't allow SSL v2 or earlier)
  - Logs level of security the server required
- http://publib.boulder.ibm.com/infocenter/zos/v1r13/index.jsp?topic=/com.ibm.zos.r13.halz001/smfcfg.htm
- And more…
- Virtual IP Addresses (subtypes 32 – 37)
- New z/OS CS SMTP server (subtypes 48 – 52)
  - New as of z/OS V1R11
- Subtypes 73 – 80 for IPSec
- UDP Socket close (subtype 10)
- Others for statistics

Mine the SMF records

- ICETOOL
- SYS1.TCPIP.AEZASMP1(EZASMF)
  - IBM sample C code to report on type 119 SMF
- Assembler
  - SYS1.MACLIB – EZASMF77
  - http://publib.boulder.ibm.com/infocenter/zos/v1r13/index.jsp?topic=/com.ibm.zos.r13.halz002/accounting.htm
- IBM Security zSecure Audit
  - Via the EV.I menu automatically processes FTP records creating reports to tell you if the FTP was encrypted or not and if so using what version of SSL and which algorithm was used
- Note: This list might not be all inclusive

FTP server transfer completion record – type 119 subtype 70

- Table 257 of the z/OS Communications Server IP Programmer's Guide and Reference shows the FTP server type 119 subtype 70 security section record layout
- Some key fields to report on:

Offset	Name	Description
1	SMF119FT_FSCProtect	Security of the control connection.
2	SMF119FT_FSDProtect	Security of the data connection.
3	SMF119FT_FSLoginMech	Was login via password or certificate?
4	SMF119FT_FSProtoLevel	Version of SSL used.
12	SMF119FT_FSCipherSpec	Encryption algorithm used.

To Encrypt or Not to Encrypt

- To protect the user ID and password of RACF accounts
  - Person in the middle attack.
- It's the 21st century so why are we still sending passwords around any network (even our internal one) in the clear?
  - Do you really trust that your LAN/WAN is bullet proof?
  - Would you take that risk?
- PCI requires encryption of cardholder data:
- 1.1.5 Documentation and business justification for use of all services, protocols, and ports allowed, including documentation of security features implemented for those protocols considered to be insecure.
  - Examples of insecure services, protocols, or ports include but are not limited to FTP, Telnet, POP3, IMAP, and SNMP.
- 4.1 Use strong cryptography and security protocols (for example, SSL/TLS, IPSEC, SSH, etc.) to safeguard sensitive cardholder data during transmission over open, public networks.

z/OS FTP client & server parameters for TLS

- In these examples we use the z/OS FTP servers TLS implementation
- EXTENSIONS AUTH_TLS
  - Enables the use of TLS by the FTP Server or client
  - Default is off
- SECURE_FTP ALLOWED
  - Encrypted FTPs are “ok”
  - Client and server setting
  - PCI requires:
  - SECURE_FTP REQUIRED
    - Changing this setting DENIES any inbound FTP that is unable to establish an encrypted session.
    - The goal after mining the type 119 subtype 3 and 70 SMF records and remediating all unencrypted FTPs is:
    - To be able to changed to REQUIRED with no impact

z/OS FTP client & server parameters for TLS (continued)

- SECURE_CTRLCONN CLEAR
  - CLEAR is the default
  - This means the control port is unencrypted by
  - Unless EXTENSIONS AUTH_TLS is specified
- PCI REQUIRES:
- SECURE_CTRLCONN PRIVATE
  - The goal after unencrypted FTP remediation is to be able to change this setting to **PRIVATE**
    - Then the z/OS FTP server rejects any FTP session that can't establish an encrypted control port connection

z/OS FTP client & server parameters for TLS (continued)

- SECURE_DATACONN CLEAR
  - CLEAR is the default
  - This means that any data sent to the z/OS FTP server is not encrypted!
- PCI REQUIRES:
- SECURE_DATALCONN PRIVATE
  - The goal after unencrypted FTP remediation is to be able to change this setting to **PRIVATE**
    - Then the z/OS FTP server rejects any FTP session that can't establish an encrypted data port connection

z/OS FTP server parameters for TLS

- SECURE_LOGIN NO_CLIENT_AUTH
  - This means we're not requiring a certificate for authentication
  - Some auditors get confused by this setting (I did too at first)
  - NO_CLIENT_AUTH is the default
- SECURE_PASSWORD REQUIRED
  - While this may seem to imply encryption all it means is:
  - We **must** enter a password to login via FTP
  - REQUIRED is the default

PCI requires strong cryptography

- The default z/OS FTP server cipher is a null cipher
  - That means it's a cipher that doesn't actually encrypt anything!!!!
  - Why? Well that's how RFC 4346 is written…
- Search your FTP parms for a statement that starts with:
  - CIPHERSUITE
  - Note by using cryptography parms that are provided by the z/OS FTP server we're using cryptography provided by System SSL TLS v1.1.
  - AT TLS offers stronger cryptography (more on that later)

Encryption too weak for PCI

- CIPHERSUITE SSL_NULL_MD5 ; 01
  - **No encryption** or message authentication and RSA key exchange
- CIPHERSUITE SSL_NULL_SHA ; 02
  - **No encryption** with MD5 message authentication and RSA key exchange
- CIPHERSUITE SSL_RC4_MD5_EX ; 03
  - **40-bit** RC4 encryption with MD5 message authentication and RSA key exchange
- CIPHERSUITE SSL_RC2_MD5_EX ; 06
  - **40-bit** RC2 encryption with MD5 message authentication and RSA key exchange
- CIPHERSUITE SSL_DES_SHA ; 09
  - **56-bit** DES encryption with SHA-1 message authentication and RSA key exchange

MD5 is Deprecated

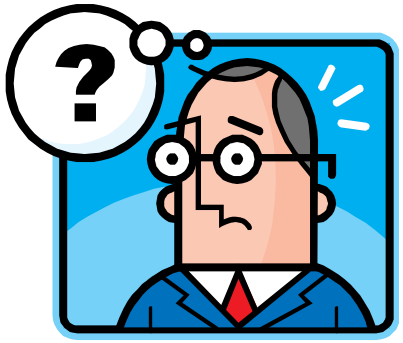
- CIPHERSUITE SSL_RC4_MD5 ; 04
  - **128-bit** RC4 encryption with MD5 message authentication and RSA key exchange
  - MD-5 is now depreciated and won't pass PCI standards
  - From Wikipedia, "…a group of researchers used this technique to fake SSL certificate validity,[7][8] and CMU Software Engineering Institute now says that MD5 "should be considered cryptographically broken and unsuitable for further use",[9] and most U.S. government applications now require the SHA-2 family of hash functions.[10] "
  - http://en.wikipedia.org/wiki/MD5

PCI Acceptable Ciphers (for now) - 128 bits minimum

- CIPHERSUITE SSL_AES_256_SHA ; 35
  - **256-bit** AES encryption with SHA-1 message authentication and RSA key exchange
  - Advanced Encryption Standard is established by the National Institute of Standards and Technology (NIST).
- CIPHERSUITE SSL_AES_128_SHA ; 2F
  - **128-bit** AES encryption with SHA-1 message authentication and RSA key exchange
- CIPHERSUITE SSL_RC4_SHA ; 05
  - **128-bit** RC4 encryption with SHA-1 message authentication and RSA key exchange
- CIPHERSUITE SSL_3DES_SHA ; 0A
  - **168-bit** Triple DES encryption with SHA-1 message authentication and RSA key exchange

SHA1 soon to be deprecated?

- SHA1 is still an approved algorithm for now but …
  - http://csrc.nist.gov/groups/ST/toolkit/secure_hashing.html
- NIST also recommends federal agencies **stop** using SHA-1 for:
  - Generating digital signatures!
  - Generating time stamps!
  - Any application that requires collision resistence!
  - http://csrc.nist.gov/groups/ST/hash/policy.html
- NIST commentson cryptanalytic attacks on SHA-1
  - http://csrc.nist.gov/groups/ST/hash/statement.html



How to specify ciphers in FTP parms

- Order matters!
- The first algorithm is the list is the first to be attempted to be used.

```
CIPHERSUITE     SSL_AES_256_SHA
CIPHERSUITE     SSL_AES_128_SHA
CIPHERSUITE     SSL_3DES_SHA
CIPHERSUITE     SSL_RC4_SHA
```

- If none of the specified algorithms can establish an encrypted session then the FTP connection fails

A sample PCI compliant z/OS FTP Server configuration

```
EXTENSIONS          AUTH_TLS
KEYRING             FTPRing
SECURE_FTP          REQUIRED
SECURE_LOGIN        NO_CLIENT_AUTH
SECURE_PASSWORD     REQUIRED
SECURE_DATACONN     PRIVATE
CIPHERSUITE SSL_AES_256_SHA
CIPHERSUITE SSL_AES_128_SHA
CIPHERSUITE SSL_RC4_SHA
CIPHERSUITE SSL_3DES_SHA
SMF     TYPE119
SMFJES TYPE119
SMFSQL TYPE119
```

A sample PCI compliant z/OS FTP client configuration

```
FWFRIENDLY          TRUE
SECURE_MECHANISM    TLS
KEYRING             *AUTH*/*
SECURE_FTP          REQUIRED
SECURE_DATACONN     PRIVATE
CIPHERSUITE         SSL_AES_256_SHA
CIPHERSUITE         SSL_AES_128_SHA
CIPHERSUITE         SSL_RC4_SHA
CIPHERSUITE         SSL_3DES_SHA
```

Client side FTP settings

- Many of the previous settings also apply to the z/OS FTP client
- z/OS FTP clients use a search order for their settings
  - The order also depends upon whether the TSO or z/OS UNIX FTP client is used
  - If not overridden by the client then it defaults to the TCPIP.DATA dataset specified in the FTP proc's SYSTCPD DD card
  - **http://publib.boulder.ibm.com/infocenter/zos/v1r13/index.jsp?topic=%2Fcom.ibm.zos.r13.halz001%2Fcjesint.htm**
- A z/OS FTP client can choose to override the encryption settings by invoking the FTP with:
  - -a never
  - Which means give me an unencrypted outbound FTP session
- Enhancement 25972
- **http://www.ibm.com/developerworks/rfe/execute?use_case=viewRfe&CR_ID=25972**

Virtual Keyrings – Required for z/OS FTP Clients

- Necessary to allow the FTP client to *automagically* use any CERTAUTH certificate that is in TRUST status without having to explicit connect the certificate to a user's keyring
- Otherwise you will have a fun time with keyring maintenance for **all** of your FTP users
  - You'll be in RACDCERT CONNECT hell constantly connecting in certificate authority certificates depending upon where someone needs to FTP.
- Activate the RDATALIB class and define to it:
  - CERTIFAUTH.IRR_VIRTUAL_KEYRING.LST UACC of READ
  - For shops that have zSecure Access Monitor you can actually tell if a user ID called for access to this resource versus IRR.DIGTCERT.LISTING
    - Which I think is totally cool ☺

System FTP SSL Trace Example

- Useful for tracing the SSL handshake – what certificate gets selected for example
- Very detailed output but you can see certificate common names and entire SSL handshake process
- My thanks to Matt Nuttall of z/OS Comm Server level 2 for this JCL:
- `//FTPSTP1  EXEC PGM=FTP,`
- `//`
  `PARM=('ENVAR("GSK_TRACE=0XFFFF","GSK_TRACE_FILE=/directory/name")',`
- `//  '/-r TLS (TRACE EXIT')`
- `//SYSFTPD  DD  DSN=YOUR.FTP.DATA,DISP=SHR`
- `//SYSPRINT DD  SYSOUT=*`
- `//OUTPUT   DD  SYSOUT=*`
- `//INPUT    DD  *`
- **`1.2.3.4`**
- **`userID password`**
- `locstat`
- `quit`
- `/*`
- Run the output of the trace through this OMVS command to convert to a human readable format.
  - gsktrace **`/directory/name`** > /tmp/ssltrace_output.txt

Stronger Cryptography requires AT-TLS

- New Function APAR OA39422 brings TLS v1.2 support to system SSL for z/OS V1R13 (included in z/OS V2R1)
  - SHA-256, SHA-384 and ciphers which use the AES-GCM symmetric algorithm during the TLS handshake
- Then also apply z/OS Communications Server New Function APAR PM62905 for z/OS V1R13 (included in z/OS V2R1)
  - Gives AT-TLS the ability to exploit TLS v1.2
- So the z/OS FTP & TN3270 servers can only support TLS v1.1
  - TLS v1.1 uses secure hash algorithms that are not so secure anymore
  - MD5 and SHA1 have been around since the internet was created!
- This means to use the strongest cryptography going forward **requires** using AT-TLS!

The Case for Elliptical Curve Cryptography

- Achieve even stronger encryption with smaller key sizes

Symmetric Key Size (bits)	RSA and Diffie-Hellman Key Size (bits)	Equivalent Elliptic Curve Key Size (bits)
80	1024	160
112	2048	224
128	**3072**	**256**
192	7680	384
256	15360	521

- "The US National Institute for Standards and Technology has recommended that [these] 1024-bit systems are sufficient for use until 2010. After that, NIST recommends that they be upgraded to something providing more security."
- http://www.nsa.gov/business/programs/elliptic_curve.shtml

Thank You! – The End

May the “z” be with you

Long Live the Mainframe

References

- Payment Card Industry Data Security Standard v2.0
- https://www.pcisecuritystandards.org/documents/pci_dss_v2.pdf

References

- *z/OS V1R13.0 System SSL Programming*
  - **C.0 Appendix C. Cipher Suite Definitions**
  - **http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/GSKA1A90/CCONTENTS?DT=20110613110223**

- *z/OS V1R13.0 Comm Svr: IP Configuration Reference*
  - **http://publibfp.dhe.ibm.com/cgi-bin/bookmgr/BOOKS/F1A1B4B1/CCONTENTS?DT=20120119011927**

References

- *z/OS V1R13.0 Comm Svr: IP Configuration Guide*
  - **http://publibfp.dhe.ibm.com/cgi-bin/bookmgr/BOOKS/F1A1B3B1/CCONTENTS?DT=20120119110606**

- *z/OS V1R13.0 Comm Svr: IP Programmer's Guide and Reference*
  - **Appendix E. Type 119 SMF records**
  - http://publibfp.dhe.ibm.com/cgi-bin/bookmgr/BOOKS/F1A1D3B1/E.0?DT=20120118013946

References

- JESINTERFACELEVEL configuration
- http://publib.boulder.ibm.com/infocenter/zos/v1r13/index.jsp?topic=%2Fcom.ibm.zos.r13.halz001%2Fcjesint.htm
- JESINTERFACELEVEL comparison chart
- http://publib.boulder.ibm.com/infocenter/zos/v1r13/index.jsp?topic=/com.ibm.zos.r13.halu001/jesintdiff.htm

References

- TCPIP Stack SMFCONFIG statement
- http://publib.boulder.ibm.com/infocenter/zos/v1r13/index.jsp?topic=/com.ibm.zos.r13.halz001/smfcfg.htm
- z/OS FTP RFE (Request for Enhancement) number 25972
- http://www.ibm.com/developerworks/rfe/execute?use_case=viewRfe&CR_ID=25972

References

- FTP configuration statements
- http://publib.boulder.ibm.com/infocenter/zos/v1r13/index.jsp?topic=%2Fcom.ibm.zos.r13.halz001%2Fcjesint.htm
- Protecting SDSF commands
- http://publib.boulder.ibm.com/infocenter/zos/v1r13/index.jsp?topic=/com.ibm.zos.r13.isfa500/isf4cs9168.htm