Securing z/VM TCP/IP

With a focus on TLS/SSL Server and Digital Certificates

Brian W. Hugenbruch, CISSP
Twitter: @Bwhugen
IBM Z Security for Virtualization and Cloud
z/VM Development Lab: Endicott, NY, US

V13.0b – Last updated 13 March 2018 for z/VM V6.4
Trademarks

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

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by © are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

* IBM Systems, IBM System z10®, IBM System Storage®, IBM System Storage DS®, IBM BladCenter®, IBM System z®, IBM System p®, IBM System i®, IBM System x®, IBM IntelliStation®, IBM Power Architecture®, IBM SureOne®, IBM Power Systems™, POWER®, POWER6®, POWER7®, POWER8®, Power®, IBM z/OS®, IBM AIX®, IBM i, IBM z/VSE®, IBM z/VM®, IBM i5/OS®, IBM zEnterprise®, Smarter Planet™, Storwize®, XIV®, PureSystems™, PureFlex™, PureApplication™, IBM Flex System™, Smarter Storage

The following are trademarks or registered trademarks of other companies.

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 therefrom. 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 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.

* All other products may be trademarks or registered trademarks of their respective companies.

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.
Disclaimer

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "AS IS" basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used instead.

Any performance data contained in this document was determined in a controlled environment and, therefore, the results which may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environments.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming or services in your country.
Profiles of Configuration Files used in this Presentation

**PROFILE TCPIP** – controls TCP/IP operations and configuration
– Sits on the TCPMAINT.198 disk, usually accessed at Filemode D
– May have a different filename, will always be filetype TCPIP
– ASSORTEDPARMS, INTERNALCLIENTPARMS, PORT, HOME, OBEY ...

**IBM DTCPARMS** – controls configurations of Service Virtual Machines
– Often renamed as <yoursys>.DTCPARMS
– Also on TCPMAINT.198
– Different definitions for SSL configuration, what TLS protocols are allowed,
  explains where the certificate database is
– For FTP, enables/disables anonymous access, turns on RACF exits
Agenda

Introducing the z/VM TLS/SSL Server
– Managing Digital Certificates in the z/VM environment
– Configuring the SSL Server
  • And Configuring a 3270 Client for Secure Communication
– Recent Changes
– Frequently Asked Questions

Back-Up Topics
– Being your own Certificate Authority
– Debugging the TLS/SSL Server
Introducing the z/VM TLS/SSL Server
This is your LinuxONE System On Lockdown.

Your Virtualization Platform

PR/SM (one LinuxONE Logical Partition)

CPACF

OSA

Crypto Express

WAS

DB2

Mobile First Server

Security Manager

Encrypted Net Traffic

Virtual Memory Management

Architected VM Separation

Role Based Access Controls

TCP/IP with TLS

VSWITCH

SVM

SVM

Encrypted Net Traffic

Mobile First Server

WAS

WAS

DB2

Security Manager

Encrypted Net Traffic
Why Does Securing z/VM TCP/IP Matter to Me?

Managing security controls for the hypervisor is a fundamental part of enterprise security management.

This includes connectivity to the hypervisor layer:
- If your guests are secure, and your hypervisor is not ...
  - ... your guests are not as secure as they should be.

This line of thinking applies both to smaller shops and to larger shops:
- Controlling potential damage
- Auditability of privileged commands
- Restrictions on access to data
- Enforcing scope of responsibility

Additionally, encrypting traffic may be mandated by clients, partners, vendors, industry regulations, or governing bodies.
What can we do to secure z/VM TCP/IP? (1 of 2)

Enable the TLS/SSL Server
– Allows (or requires) encrypted traffic to and from the hypervisor
– For TN3270 connections, it requires a client certificate

Enable z/VM service virtual machines (SVMs) to use TLS/SSL as well
– Telnet, FTP, SMTP
– Port-based controls for other services (web servers, or even SMAPI)

Apply service to TCP/IP
– Especially security-relevant PTFs
– Lack of currency means gaps in security may appear!
What can we do to secure z/VM TCP/IP? (2 of 2)

Adjust other controls as pertinent:

– **TIMEMARK** for timing out Telnet sessions (PROFILE TCPIP)
– Disable Anonymous FTP if appropriate (SRVRFTP.CONFIG)
– Make sure RESTRICTLOWPORTS is enabled (PROFILE TCPIP)
– SMTP FORWARDMAIL (disabled by default in z/VM 6.4)
– Remove unused TCP/IP SVMs (NOLOG in USER DIRECT)
– Enable services for RACFVM control
  • Security labeling for services if appropriate (or SYSNONE for TCPIP)
  • RACF configuration for SSLSERV later in this presentation
z/VM 6.4 TLS/SSL Server

- A CMS-based server associated with the TCP/IP stack
- Secures point-to-point traffic to your hypervisor
- Telnet, FTP and SMTP provide “dynamic” SSL traffic
- Port-based “static” SSL
A digital certificate is a unique identifier

Contains:
- Public key
- X.509 information
- Digital signature

A mechanism for authenticating identity when exchanging a cryptographic secret
The server authenticates itself to the user ...

Server’s public key and certificate

TCPIP

SSL00001

System SSL

GSKADMIN

z/VM 6.4

PR/SM (one IBM Z Logical Partition)

Crypto Express

CPACF

TCPMAINT

By BWHUGEN

BFS

/etc/gskadm/Database.kdb

Server’s private key and .p12
The client authenticates itself to the server
(*TN3270 only*)

- Client’s certificate
- Network
- Crypto Express
- TCPIP
- SSL00001
- System SSL
- GSKADMIN
- PR/SM (one IBM Z Logical Partition)
- TCPMAINT
- BFS
- /etc/gskadm/Database.kdb
- Server’s private key and .p12
- z/VM 6.4
- TCPMAINT By BWHUGEN
- TCPIP
- SSL00001
- SSL00001
- SSL00001
- System SSL
- GSKADMIN
**z/VM TLS/SSL Server – Certificate Management**

- **Userid GSKADMIN**
  - Specifically for certificate management for one z/VM LPAR
  - PROFILE enrolls in and accesses appropriate filepools and directories
  - Gskkyman command-line application runs here
    - Manages databases stored in a Byte-File System (BFS)
    - TLS Servers and LDAP Servers can share databases and certificates
Certificate Management for z/VM TLS

Logging onto GSKADMIN:

Profile... Setting up BFS environment...
Profile... Determining what is currently mounted...
Nothing is mounted

Profile... Mounting root file system...
Profile... Mounting GSKSSLDB file space at: /etc/gskadm/
Profile... Setting working directory to: /etc/gskadm/
Profile... (for direct access to key database files)...
Profile... Checking mounts...
Mount point = '/etc/gskadm'
Type Stat Mounted
BFS R/W '/../'VMBFS:VMSYS:GSKSSLDB/'

Mount point = '/'
Type Stat Mounted
BFS R/W '/../'VMBFS:VMSYS:ROOT/'

Profile... Checking current directory content...
Directory = '/etc/gskadm'
[.........]
Profile... Setup complete; Environment prepared for use of GSKKYMAN
GSKADMIN > Looking around

**openvm listf**

<table>
<thead>
<tr>
<th>Directory</th>
<th>Bytes</th>
<th>Path name component</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/etc/gskadm</code></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Update-Dt</td>
<td>Update-Tm</td>
<td>Type</td>
</tr>
<tr>
<td>02/02/2013 02:41:00</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>01/31/2013 19:45:47</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>01/31/2013 19:46:09</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>01/31/2013 19:46:09</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>01/31/2013 15:44:32</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>02/06/2013 11:12:43</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>02/01/2013 08:23:04</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>02/01/2013 08:22:55</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>01/31/2013 19:20:46</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>01/31/2013 19:39:56</td>
<td>F</td>
<td>1</td>
</tr>
</tbody>
</table>

Ready; T=0.01/0.01 11:37:27
GSKADMIN > Using gskkyman

**gskkyman**

---

**Database Menu**

1 - Create new database  
2 - Open database  
3 - Change database password  
4 - Change database record length  
5 - Delete database  
6 - Create key parameter file  
7 - Display certificate file (Binary or Base64 ASN.1 DER)

0 - Exit program

Enter option number:
Creating a Certificate Database

1. Create new Database

Enter key database name (press ENTER to return to menu):
ForThisPresentation.kdb

Enter database password (press ENTER to return to menu):
Re-enter database password:

Enter password expiration in days (press ENTER for no expiration):
1000

Enter database record length (press ENTER to use 5000):

Enter 1 for FIPS mode database or 0 to continue:
1

Key database /etc/gskadm/ForThisPresentation.kdb created.

Press ENTER to continue.
GSKADMIN > Modifying Permissions

```
openvm listf (own)
```

```
gskadmin security rw- --- --- F 'ForThisPresentation.kdb'
gskadmin security rw- --- --- F 'ForThisPresentation.rdb'
gskadmin security rw- --- --- F 'ForThisPresentation.sth'
```

```
openvm permit Database.kdb rw- r-- --- (replace
```

```
gskadmin security rw- r-- --- F 'ForThisPresentation.kdb'
gskadmin security rw- r-- --- F 'ForThisPresentation.rdb'
gskadmin security rw- r-- --- F 'ForThisPresentation.sth'
```
Certificate Management for z/VM TLS

Default database location: /etc/gskadm

GSKADMIN automatically mounts and accesses the database’s directory

Database should be located at mount point

May require manual configuration if not using the defaults
GSKADMIN > Using gskkyman

Opening a Certificate Database
– 2. Open Database (enter database name, e.g. Database.kdb, and pwd)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Manage keys and certificates</td>
</tr>
<tr>
<td>2</td>
<td>Manage certificates</td>
</tr>
<tr>
<td>3</td>
<td>Manage certificate requests</td>
</tr>
<tr>
<td>4</td>
<td>Create new certificate request</td>
</tr>
<tr>
<td>5</td>
<td>Receive requested certificate or a renewal certificate</td>
</tr>
<tr>
<td>6</td>
<td>Create a self-signed certificate</td>
</tr>
<tr>
<td>7</td>
<td>Import a certificate</td>
</tr>
<tr>
<td>8</td>
<td>Import a certificate and a private key</td>
</tr>
<tr>
<td>9</td>
<td>Show the default key</td>
</tr>
<tr>
<td>10</td>
<td>Store database password</td>
</tr>
<tr>
<td>11</td>
<td>Show database record length</td>
</tr>
<tr>
<td>0</td>
<td>Exit program</td>
</tr>
</tbody>
</table>

Enter option number (press ENTER to return to previous menu):
GSKADMIN > Importing Certificates

**Importing certificates**

Certificates can be imported into the certificate database through gskkyman. But first they need to be placed in the appropriate BFS directory.

If possible, FTP directly into the BFS

```
   – cd /../VMBFS:VMSYS:GSKSSLDB/
```

If not, transfer the certificate to GSKADMIN and then issue the following command ... selecting a bfsline option based on file format.

```
openvm putbfs TESTCERT P12 A /etc/gskadm/testcert.p12 (bfsline none

or

openvm putbfs MYCACERT PEM A /etc/gskadm/mycacert.pem (bfsline nl
```
Standard certificates can be either Base64 or binary format – and `bfsline none` is for binary format only. *If you can open it and read any of it, it’s in Base64!*

```
-----BEGIN CERTIFICATE-----
MII<EOTCCA+0gAwIBAgIODEAAHMA0GCSqGSIb3DQEBBQUAMIGcMQswCQYDVQQGEwJV
UzERMA8GA1UECBMITmV3I1vcmsxETAPBqNVBAcTCEVuZGljb3R0MRgwFgYDVQQK
Ew96Vk0gRGV2ZWxvcGl1bnQxDDAKBgNVBAsTA1NTTDEcMBoGA1UEAxMTQnJpYW4g
Vy4gSHVnZW5icnVjaDEhMB8GCSqGSIb3DQEJARYSYndodWd1bkB1cy5pYm0uY29t
MB4XDTEzMDMyNzE3NTMwOVoiXDE0MDMyNzE3NTMwOVowZjELMAkGA1UEBhMCVVMx
ETAPBgNVBAsTA1NTTDEcMBoGA1UEAxMTQnJpYW4gVy4gSHVnZW5iicnVjaDCCAiIwDQYJ
KoZIhvcNAQEBBQADgggIPADCCAggCgIBApb/rg0Vz+++X71JZ2N7xDcktOesxjvlkA
2n1HRnb3VCO5H1ROKet10xd4QhBoLWL+GJgo2vYy1jBM3fP/KX61FYcC+j+zwUMIu
+eGOB+DrMvfL4cZnVYEkWTgBnEKRLQEIJ+KmgGnJgtJYRjdz54kaX1gB2obupCui
09iYZDVkzdiizu/S1rM0d3jz3p6MRWlMNf9uf6a4bNd+bCI7HnVLsLvfp3wCW
MTtKjAx6n2PAgMBAAGjeb5MAKGA1udEwQCMAAwLAYJ1ZfAYb4qwENBBWHU9w
ZW5TU0tgR2VuXJhdGVkIENlcnRpmzljXRY1MB0GA1UDgQWBBTiatA5nzhUruNDs9/TJPz/F3pTAPbhNHSMEGDAwgBT7hrHge6CiBsJYP2+4DBIqzSB5CeZABkgq
hkiG9w0BAQUFANBAAnwiC/Z/IvzFImTcgvNC3PH99c9u8J0u5KiAT39c6ia+FuZZ
i3tBDKoSCFy2kbBc4x6CQNyazovVSUtJRJquQU=
-----END CERTIFICATE-----
```

"bÑ""b""""fç7""""µb""""b""""""b""""""fç7""""""µb""""""b""""""b""""""fç7""""...
Certificate Management for z/VM TLS

Standard certificates (.pem) tend to be Base64. .p12 files, the PKCS #12 format for a Certificate With Private Key, is binary only.

Once the key is in the BFS directory, access gskkyman. Open the database and select either:

- 8. Import a certificate and a private key
- 1. Manage keys and certificates
  7. Import a certificate
Certificate Management for z/VM TLS

**A few thoughts:**

When making changes to a certificate database in use by a running SSL Server virtual machine, be sure to issue an SSLADMIN REFRESH from a privileged userid.

The server will reload its environment without interrupting existing secure connections.

Important for when certificates need to be renewed, replaced or removed.

SSLADMIN REFRESH will automatically be transmitted to all SSL servers in an SSL Pool.
Configuring the z/VM TLS/SSL Server
z/VM TLS/SSL Server – Certificate Management

- Configuration done on **TCPMAINT**
- Privileged maintenance userid
- Covers operations for all the TCP/IP Service Virtual Machines
- Not necessarily authorized for certificate management, though
# z/VM TLS/SSL Server >>
## DTCPARMS Options

DTCPARMS values associated with your TLS/SSL Server:

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:Admin_ID_list.</td>
<td>Userids authorized to execute privileged commands – e.g., <strong>SSLADMIN</strong> commands</td>
</tr>
<tr>
<td>:Mixedcaseparms.</td>
<td>Parameters are supported in mixed case</td>
</tr>
<tr>
<td>:Mount.</td>
<td>Certificate database location. Default is /etc/gskadm/</td>
</tr>
<tr>
<td>:Parms.</td>
<td><strong>As per the VMSSL command</strong></td>
</tr>
<tr>
<td>:Stack.</td>
<td>Associated TCPIP virtual machine. <em>This tag is required; otherwise, the SSL server / pool cannot be identified during stack initialization!</em></td>
</tr>
<tr>
<td>:Timestamp.</td>
<td>On/Off for timestamps on terminal messages and cmd responses</td>
</tr>
<tr>
<td>:Timezone.</td>
<td>Set timezone of server</td>
</tr>
<tr>
<td>:Vmlink.</td>
<td>Sets a Pool member’s SFS space</td>
</tr>
</tbody>
</table>
Specified either on VMSSL (command-line exec) or DTCPARMS

Persists for the run-time for a server or server pool. Must be consistent for all members of a server pool

Options:

- **KEYFILE** - BFS location of the certificate database
- **CACHELIFE** - for secure connections, in hours, minutes, seconds
- **CACHECLEANUP** - processed every n connections
- **MODE** - sets a cryptographic compliance mode
  - **MODE FIPS-140-2**
  - **MODE NIST-800-131A**
  - FIPS - equivalent to MODE FIPS-140-2
- **PROTOCOL** - enable or disable SSL/TLS levels.
  - TLS 1.2 and TLS 1.1 enabled by default << different from earlier VM releases
  - Available protocols change based on MODE
- **EXEMPT|ENABLE** - disable or enable particular cipher suites
- **GSKTRACE** - enable System SSL tracing
- **TRACE/NOTRACE** - enable SSL Server tracing
  - Can be dynamically manipulated via authorized commands
z/VM TLS/SSL Server >> FIPS 140-2 Compliance

Requires both database support ...

– In *gskkyman*, the *Create New Database* option will prompt for FIPS mode:

```
Enter 1 for FIPS mode database or 0 to continue: 1

Key database /etc/gskadm/ForThisPresentation.kdb created.
```

... and SSL Server Support

– DTCPARMS: **FIPS** (or **MODE FIPS-140-2**) or

– VMSSL: **FIPS** (or **MODE FIPS-140-2**)
z/VM TLS/SSL Server >>
Mode Selection

**MODE FIPS-140-2**
- Replaces ‘FIPS’ keyword
- Minimum Protocol of TLS 1.0
- Export ciphers restricted
- Minimum key exchange value of 1024
- FIPS-compliant database required
  - Integrity checking (HMAC-SHA256): Digitally signs the crypto modules and database against tampering
  - Known Answer Tests – verify integrity after initialization

**MODE NIST-800-131A (z/VM 6.3 on)**
- Minimum Protocol of TLS 1.2
- Minimum key exchange value of 2048
  - DSA certificate usage prohibited in z/VM 6.3
- Minimum hash of SHA2
- No certificate database requirements
  - Integrity checking only (HMAC-SHA256)
- Supersedes FIPS-140-2 where applicable

Requires **APAR PM93363** (z/VM 6.3 only)

*When running in either mode, the cipher suites available adjust according to security settings*
Configuring TCP/IP Services for Secure Connectivity

- **TCPIP Configuration**
  - `SSLLIMITS` (determines volume of concurrent connections per server)
  - `SSLSERVERID` (identifying the server to TCPIP)
    - If detected, TCPIP will autolog SSLSERV automatically
    - Use `*` for a pool of SSL machines – association happens in DTCPARMS

- **Implicit ("static") SSL**
  - Establish a permanently secure port for secure connectivity
  - Standardized in RFC 2228
  - **PROFILE TCPIP: PORT statement**
    
    ```
    PORT 
    21 TCP FTPSERV SECURE tlslabel
    ```

    - `tlslabel` – name of certificate in database (max. of 8 characters)
    - Can use port ranges instead of a single port
Configuring TCP/IP Services for Secure Connectivity

- **Configuration File Updates (for “Dynamic” SSL)**
  - **TN3270**: INTERNALCLIENTPARMS (in PROFILE TCPIP)
    - SECURECONNECTION {Required | Allowed | Never}
    - CLIENTCERTCHECK {FULL | NONE}
    - TLSLABEL <server_certificate_name>
  - **FTP**: SRVRFTP CONFIG (server); FTP DATA (client)
    - PASSIVEPORTRANGE
    - SECURECONTROL, SECUREDATA {Required | Allowed | Never}
    - TLSLABEL <server_certificate_name>
  - **SMTP**: SMTP CONFIG
    - TLS Statement {Required | Allowed | Never}
    - TLSLABEL <server_certificate_name>
    - These can be adjusted dynamically (SMSG, NETSTAT OBEY)
Configuring Secure Connectivity Dynamically

- z/VM Applications support SMSG
  - SMSG FTPSERV QUERY SECURE
  - SMSG FTPSERV SECURE CONTROL REQUIRED
  - SMSG SMTP TLS NEVER

- z/VM Telnet – NETSTAT OBEY / OBEYFILE
  - Adjust INTERNALCLIENTPARMS

- SSL Server
  - Operating parameters (DTCPARMS) cannot be dynamically changed
  - Certificate database changes can be seen by issuing SSLADMIN REFRESH from GSKADMIN (or another authorized userid).
RSCS and TCPNJE

TLS Encryption of RSCS and TCPNJE

– Shipped as an SPE to z/VM 6.3 (APAR PI56474 and associated service)

– Allows RSCS to encrypt traffic to other TCPNJE nodes using the TLS/SSL Server
  • Uses existing key databases or .P12 files
  • CPACF if enabled

– **TLSLABEL** parameter for specifying certificate label

– TLS tag on **SMSG RSCS QUERY LINK** to note which connections are encrypted

– In z/VM 6.4:
  • C and Assembler APIs that made this possible open for system programmer use

– **Best Practices Whitepaper:**
Running the SSL Server

Starting the Server

When properly configured, SSLSERV or an SSL* pool will start when the TCPIP virtual machine is started

– In a pool, the first pool member (e.g., SSL00001) is autologged first

To bring a specific server online:

– SSLADMIN START (SSL SSL00004

or

– NETSTAT SSL START SSL00004
Running the SSL Server

**SSLADMIN command**

- Privileged command ( :Admin_ID_list. )
- Reports information on SSL server status and connections
- Can route commands to specific SSL servers or TCPIP stacks

```
-QUERY STATUS SUMMARY--.
>>--SSLADMIN----.-------------.--'command---------'--operands--------->
   '-diagnostic_op-'

>---------------------------.--------------------------.<
   '-(-- Options |----.-----'`
   ')-'`

Options:

|--.------------------.------------------|--
|   .-ALL----. | 'TCPserver--userid-' 'MONitor--seconds-' |
|   '-SSLserver--userid-'`
```

Running the SSL Server

SSLADMIN command

- **CLEAR**  
  remove userid(s) set by SET
- **CLOSECON / LOG**  
  retrieves console log
- **HELP**  
  displays help information
- **QUERY**
  - Status Summary  
    returns general server data
  - Status Details  
    returns specific server data
  - Settings  
    returns current command defaults
  - Cache  
    returns cache data
  - Sessions  
    returns data on active secure sessions
  - Trace  
    returns trace settings
- **RESTART**  
  quiesces and re-IPL’s SSL server
- **REFRESH**  
  reaccess certificate database
- **SET**  
  sets default targets for SSLADMIN commands
- **START / STOP**  
  starts / stops an SSL server
- **SYSTEM**  
  used to issue CP or CMS commands
- **TRACE / NOTRACE**  
  enables / disables tracing
Running the SSL Server

**Tracing**

– Configured at start-up through DTCPARMS or VMSSL
– Can be turned on/off with SSLADMIN:
Configuring Clients for Secure Connectivity
Configuring External Clients to Connect to z/VM

The compatibility and capabilities of external clients will vary
– Consult the TCPIP service webpage for thoughts

The terminology of external clients may vary (SSL vs TLS)

The certificate management techniques for local clients will also vary (MSCAPI, GSKit, openSSL, x3270 ...)

During the handshake, the external client will need to understand both the server certificate and (if enabled) the client’s certificate
– These may or may not be generated off the same root certificate
– Installation into a local certificate database will be required
Configuring PComm for Client Certificate Validation

Telnet-negotiated: dynamic SSL

**MSCAPI**: certificates are stored in Windows, rather than PComm’s GSKit library.

**TLS**: instead of SSLv3. FIPS mode disabled in this example. TLS 1.1 and TLS 1.2 available in later versions of the client

“Personal Certificate” represents the client’s identifying certificate. This will be sent if z/VM’s Telnet server is configured for `CLIENTCERTCHECK FULL`. 
Configuring PComm for Client Certificate Validation

Example of certificates stored in MSCAPI:
Note that certificates stored in MSCAPI will need to be assigned a particular purpose (in the case of our certificate, enabling for client authentication).
For Linux clients ...

Linux tends to be a little easier – place appropriate certificate files into a local keystore (OpenSSL) and make sure the certificate and/or key files are available when executing OpenSSL or x3270 commands.

X3270 seems not to take P12 files; instead, you’ll be using commands like:

```
x3270 -certfile mycert.cert -keyfile mykey.key -keypasswd string:mypwd -cafile MyRootCA.pem 192.168.0.1
```
Host On Demand
Frequently Asked Questions
Frequently Asked Questions

**Does z/VM SSL use the Crypto Express Cards?**
**Answer:** Yes – as of z/VM 6.4 APAR PI72106. Add CRYPTO APVIRT to your PROFILE TCPSSLU or userid of choice (LDAPSRV, GSKADMIN).

**Why isn’t RACFVM the keystore or certificate store for [insert function here]?**
**Answer:** RACFVM does not support RACDCERT or the DIGTCERT class, so it cannot provide that functionality.

**Can SSL servers for different TCP/IP stacks share the same certificate database?**
**Answer:** Yes, as long as your security policy permits this. Bear in mind that this may require “wildcard” certificates which cover multiple subdomains on your network.
Frequently Asked Questions

**Is FIPS Mode for SSLSERV the same as the Common Criteria certified configuration?**

**Answer:** No. FIPS 140-2 and Common Criteria, while analogous in their cipher requirements, are **not** the same – they have slightly different requirements for key length and cipher suite usage. Additionally, FIPS mode may require changes to your certificate database.

Check your security policy; your environment configuration may require either, or both, or something even more stringent.

**Can RACF and SSL be combined? What implications does this have for configuration?**

**Answer:** Yes! Just be certain that the SSL Server virtual machines have the authorities it needs in order to do its job. For example:
RACF: Reader access for SSL

Authorize all users to send files to the SSL machine's reader.
If there is already a SSL VMRDR profile defined, alter it, by entering:

- RAC RALTER VMRDR SSL00001 UACC(UPDATE)
- RAC RALTER VMRDR SSL00002 UACC(UPDATE)
- RAC RALTER VMRDR SSL00003 UACC(UPDATE)
- RAC RALTER VMRDR SSL00004 UACC(UPDATE)
- RAC RALTER VMRDR SSL00005 UACC(UPDATE)
- RAC RALTER VMRDR SSLDCSSM UACC(UPDATE)
RACF, SSL and VMSEGMT

If RACF is being used to control restricted segments with the VMSEGMT class, give UPDATE authority for SSL to so SSL has shared write access to the DCSS.TCPIP segment.

– RAC RDEFINE VMSEGMT DCSS.TCPIP UACC(NONE)
– RAC PERMIT DCSS.TCPIP CLASS(VMSEGMT) ID(SSL00001) ACCESS(UPDATE)
– RAC PERMIT DCSS.TCPIP CLASS(VMSEGMT) ID(SSL00002) ACCESS(UPDATE)
– RAC PERMIT DCSS.TCPIP CLASS(VMSEGMT) ID(SSL00003) ACCESS(UPDATE)
– RAC PERMIT DCSS.TCPIP CLASS(VMSEGMT) ID(SSL00004) ACCESS(UPDATE)
– RAC PERMIT DCSS.TCPIP CLASS(VMSEGMT) ID(SSL00005) ACCESS(UPDATE)
– RAC PERMIT DCSS.TCPIP CLASS(VMSEGMT) ID(SSLDCSSM) ACCESS(UPDATE)
RACF, SSL, and RACROUTE

To record activity in the RACF system audit trail, they must each be authorized. Enter:

- `RAC SETROPTS CLASSACT(FACILITY)`
- `RAC RDEFINE FACILITY ICHCONN UACC(NONE)`
- `RAC PERMIT ICHCONN CLASS(FACILITY) ID(SSL00001) ACCESS(UPDATE)`
- `RAC PERMIT ICHCONN CLASS(FACILITY) ID(SSL00002) ACCESS(UPDATE)`
- `RAC PERMIT ICHCONN CLASS(FACILITY) ID(SSL00003) ACCESS(UPDATE)`
- `RAC PERMIT ICHCONN CLASS(FACILITY) ID(SSL00004) ACCESS(UPDATE)`
- `RAC PERMIT ICHCONN CLASS(FACILITY) ID(SSL00005) ACCESS(UPDATE)`
If RACF is being used to control minidisk access with VMMDISK class, enable minidisk access for any TCPIP userid that SSL uses.

- RAC PERMIT 6VMTCP40.491 CLASS(VMMDISK) ID(SSLDCSSM) ACCESS(READ)
- RAC PERMIT 6VMTCP40.492 CLASS(VMMDISK) ID(SSLDCSSM) ACCESS(READ)
- RAC PERMIT TCPMAINT.591 CLASS(VMMDISK) ID(SSLDCSSM) ACCESS(READ)
- RAC PERMIT TCPMAINT.198 CLASS(VMMDISK) ID(SSLDCSSM) ACCESS(READ)
- RAC PERMIT 6VMTCP40.491 CLASS(VMMDISK) ID(SSL00001) ACCESS(READ)
- RAC PERMIT 6VMTCP40.492 CLASS(VMMDISK) ID(SSL00001) ACCESS(READ)
- RAC PERMIT TCPMAINT.591 CLASS(VMMDISK) ID(SSL00001) ACCESS(READ)
- RAC PERMIT TCPMAINT.198 CLASS(VMMDISK) ID(SSL00001) ACCESS(READ)

- (repeat for SSL00002, SSL00003 ...) 

RPIDIRCT should cover these already ...
Summary

Protecting connectivity to the hypervisor is a key part of a security policy. z/VM offers you the controls to restrict ports, enable timeouts, and manage access.

The TLS/SSL server provides a scalable SVM for handling encrypted traffic to the hypervisor.

z/VM 6.4 delivers enhanced function:

- Updated strong defaults
- Protocol selection
- Stronger cryptographic modes
- More flexible certificate management
- More resilient hashing
- FIPS 140-2 Validation In Progress

Select the security policy that is right for you and your company.
For More Information ...


z/VM Security resources: http://www.vm.ibm.com/security

Security for Linux on System z (SG24-7728), IBM RedBooks


z/VM TCP/IP Resources: http://www.vm.ibm.com/related/tcpip/

Contact Information:

Brian W. Hugenbruch
IBM Z Security for Virtualization & Cloud
bwhugen at us dot ibm dot com
@Bwhugen
Break glass in case of emergency:
Backup Slides
Back-Up Slides:

HOW TO BE YOUR OWN CERTIFICATE AUTHORITY IN Z/VM TLS/SSL
How To Be Your Own Certificate Authority

**Problem:** Obtaining certificates from a trusted Certificate Authority is good for external-facing zones ... but paying money for the privilege of an officially recognized certificate may be beyond the needs of your environment.

**Solution:** Be your own Certificate Authority

- Can answer certificate requests using **gskkyman**
- Useful for test-oriented or internal-only environments

**References:**

- *z/VM TCP/IP Planning and Customization*, Chapter 18
- *z/VM TCP/IP LDAP Administrator’s Guide*, Chapter 15
# How To Be Your Own Certificate Authority

<table>
<thead>
<tr>
<th>Certificate Authority (System A)</th>
<th>Server or Client (System B)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1 - Create a key database</strong></td>
<td><strong>Step 1 - Create a key database</strong></td>
</tr>
</tbody>
</table>
| Create a key database using the `gskkyman` command:  
  - From the **Database Menu**, select option 1 - Create new database  
  See “Creating, Opening and Deleting a Key Database File” on page 203 for details. | Create a key database using the `gskkyman` command:  
  - From the **Database Menu**, select option 1 - Create new database  
  See “Creating, Opening and Deleting a Key Database File” on page 203 for details. |
| **Step 2 - Create a Root Certificate Authority certificate** | **Step 2 - Create a Root Certificate Authority certificate** |
| Create a Certificate Authority certificate:  
  - From the **Key Management Menu**, select option 6 - Create a self-signed certificate  
  - From the **Certificate Type** menu, select one of the CA values for your certificate type  
  See “Creating a Self-Signed Server or Client Certificate” on page 208 for details. | No action required. |
| **Step 3 - Create a certificate request** | **Step 3 - Create a certificate request** |
| No action required. | Create a certificate request:  
  - From the **Key Management Menu**, select option 4 - Create new certificate request  
  - From the **Certificate Type** menu, select one of the certificate types  
  See “Creating a Certificate Request” on page 211 for details. |
How To Be Your Own Certificate Authority

Step 4 - Send the certificate request to the CA

No action required. Send the certificate request to the CA. See “Sending the Certificate Request” on page 217.

Step 5 - Sign the certificate request

To sign the certificate request, the gskkyman command must be issued using command-line options (see “GSKKYMAN Command Line Mode Syntax” on page 237 for a description of the options). The gskkyman command must be issued with the following parameters:

gskkyman -g -x num-of-valid-days
-cr certificate-request-file-name
-ct signed-certificate-file-name
-k CA-key-database-file-name
-1 label
How To Be Your Own Certificate Authority

| Step 6 - Send the signed CA certificate and the newly signed certificate to the requestor |
| Export the signed CA certificate (created in Step 2) to a Base64 file (DER or PKCS #7) See "Copying a Certificate Without its Private Key" on page 222. Send (for example, without its private key ftp) the Base64 file and the newly signed certificate (created in Step 4) to the requestor. | No action required. |

| Step 7 - Import the CA certificate |
| No action required. | Import the CA certificate. See "Importing a Certificate from a File as a Trusted CA Certificate" on page 231. |

| Step 8 - Receive the signed certificate |
| No action required. | Receive the signed certificate. See "Receiving the Signed Certificate or Renewal Certificate" on page 217. Note: Depending upon the SSL application, you may need to either send the CA certificate to the client, or the server application may actually present the certificate to the client for them during SSL session setup. |
Back-Up Slides:
DEBUGGING THE TLS/SSL SERVER
TLS/SSL Server: Debugging

**Common data** you may need to debug SSL server problems:

TCPIP DATA (connection to the TCP/IP stack)

DTCPARMS (server configuration, SSLDCSS configuration)

- *Most common problems tend to be either a misconfiguration of DTCPARMS or a DTCPARMS / TCPIP mismatch*

PROFILE TCPIP (stack configuration)

SSL, TCP/IP and SSL DCSS Management Agent server console messages

SSLADMIN or NETSTAT command responses

GSKADMIN console information

Trace output from SSL or TCP/IP
TLS/SSL Server: Debugging

**Problem:** The SSL server does not initialize and run SSLSERV MODULE

**Symptoms:**
TCPIP starts, but SSL server and protected services do not

Console messages for the SSL server which resemble:

DTCRUN1028E :Stack.TCPIP11 specified in GDLRCT2 DTCPARMS D1 does not match "TcpipUserid TCPIP10" in the TCPIP DATA file

DTCRUN1099E Server not started - correct problem and retry
TLS/SSL Server: Debugging

**Problem:** The SSL server does not initialize and run SSLSERV MODULE

**Analysis:**

Check the SSL server console for messages

Verify that the TCPIPUSERID statement in TCPIP DATA lists the correct TCPIP virtual machine for your SSL server

Confirm DTCPARMS settings for `:stack.`, tags and `:vmlink.`, tag

For an SSL pool server, confirm that the server has been enrolled in the appropriate SFS file pool, and that an alias to the (common-use) PROFILE EXEC is in place

For an SSL pool server (and, the case of having attempted a restart of the subject server) confirm that DTCPARMS configuration has not been changed, while one or more other pool servers remain in operation
TLS/SSL Server: Debugging

**Problem:** the server cannot use the key database

**Symptoms:**

SSL server does not start

Console messages for the SSL server which resemble:

DMSOVZ2113E Object does not exist: '/..:/VMSYSU:GSKADMIN/etc/gskadl'

DTCRUN1001E "OPENVM MOUNT /..:/VMSYSU:GSKADMIN/etc/gskadl /" failed with return code 28

DTCRUN1099E Server not started - correct problem and retry
TLS/SSL Server: Debugging

**Problem**: the server cannot use the key database

**Analysis:**

Verify that the Byte File System (BFS) parameters for the DTCPARMS `mount` tag

Confirm that the necessary file permissions have been established

- Database.kdb, Database.rdb, Database.sth

Confirm that the file pool server for the BFS user space (**VMSYSU**, by default) is operational

Use the GSKKYMAN utility to confirm that the key database has been properly created, and that the correct database has been identified via the VMSSL command KEYFILE operand
TLS/SSL Server: Debugging

**Problem:** a server cannot use the session cache

**Symptoms:**
TCPIP and SSL pool initialize properly
Connections suddenly cannot be **re**-established
SSLADMIN messages which resemble the following:

```
DTCSSL2421E SSL00001: Communication error: Connection timed out
```
TLS/SSL Server: Debugging

**Problem:** a server cannot use the session cache

**Analysis:**

Verify that the SSL DCSS Management Agent is operational

- QUERY <userid> should indicated that the machine is running disconnected:

  ```
  query ssldcssm
  SSLLDCSSM - DSC
  Ready;
  ```

Verify that SSLDCSSM has been configured properly

- Check DTCPARMS and configuration files
- Issue CP QUERY NSS commands
  
  - **Class E privilege** required for the issuing userid
  - User count should match pool size plus one (SSL* and DCSSM) if servers are running

**Output should look similar to the following:**

```
---> CP QUERY NSS NAME TCPIP MAP
FILE  FILENAME FILETYPE MINSIZE BEGPAG ENDPAG TYPE CL #USERS PARMREGS VMGROUP
9539  TCPIP  DCSS  N/A  10000  100FF SN R 00006  N/A  N/A

---> CP QUERY NSS USERS TCPIP
FILE  FILENAME FILETYPE CLASS
9539  TCPIP  DCSS  R

SSL00005 SSL00004 SSL00002 SSL00003 SSL00001 SSLLDCSSM
```
TLS/SSL Server: Debugging

**Problem:** a server cannot use the session cache

**Analysis:**

Verify that SSLDCSSM has been initialized prior to the SSL server

- DTCRUN1043I Initiating XAUTOLOG of server SSLDCSSM
  This message should appear in the TCPIP stack’s console log prior to any SSL configuration / initialization messages

Confirm that the necessary NAMESAVE statements are present in the CP directories for the SSL server and its DCSS Management Agent
TLS/SSL Server: Debugging

**Problem:** The server cannot connect to the TCP/IP virtual machine

**Analysis:**

Verify the TCPIPUSERID statement in TCPIP DATA file

- should cite the correct TCP/IP server virtual machine

Confirm that the correct TCP/IP server is identified by a DTCPARMS `:stack.` tag defined for the subject SSL server

Verify that the TCP/IP server is started

Check the TCP/IP server console for messages that indicate a problem. *(z/VM: TCP/IP Messages and Codes)*

Use the FLOW or DEBUG traces to gather additional information. Update the DTCPARMS `:parms.` tag for the SSL server to include the TRACE FLOW or TRACE DEBUG operand, then start the server. This will provide debug information during the server start up.
TLS/SSL Server: Debugging

**Problem:** Incorrect parameters are passed to SSL server

**Symptom:** SSL server is running, but not behaving as expected

**Analysis:**

Use SSLADMIN QUERY STATUS to determine which options are in effect

Check that all parameters are correctly specified in the DTCPARMS :parms.

**Tag**

Compare parameters against message DTCRUN1011I in the server console
TLS/SSL Server: Debugging

**Problem:** Protected application server (e.g., FTP) shuts down at start-up

**Symptoms:**
Console files received from application userids on autolog of TCPIP virtual machine
Application server cannot be autologged, will not respond to commands
**Problem:** Protected application server (e.g., FTP) shuts down at start-up

**Analysis:**

Confirm SSL server is running (NETSTAT CONFIG SSL)

Confirm SSL server is listening (NETSTAT CONN or NETSTAT ALLCONN)

Verify the SSLSERVERID statement in PROFILE TCPIP reflects the correct SSL server configuration

Check the application server console for indications of problems. *(z/VM: TCP/IP Messages and Codes)* For example:

12:30:46 DTCFTS8467E Error verifying TLS label NOTHERE: Label is not recognized

*(continued ...)*
Problem: Protected application server (e.g., FTP) shuts down at start-up

Analysis:

Using the GSKKYMAN utility, verify that the TLSLABEL specified is present in the certificate database, and conforms to naming requirements:

- On GSKADMIN or other authorized userid, invoke gskkyman
- Open the appropriate certificate database <filename>.kdb
- Choose option 1, “Manage keys and certificates”
- The certificate with key with the correct TLSLABEL should appear on this list

Verify the TLSLABEL statement and the correct value have been specified in the application server configuration file:

- PROFILE TCPIP (or its equivalent) for TELNET
- SMTP CONFIG (or its equivalent) for SMTP
- SRVRFTP CONFIG (or its equivalent) for FTP

An incorrect or misspelled TLSLABEL value in an application server configuration file can prevent such a server from initializing
TLS/SSL Server: Debugging

**Problem:** Connection to protected application cannot be established

**Symptom, z/VM FTP:**
220 Connection will close if idle for more than 5 minutes.

>>>AUTH TLS
421 Temporarily unable to process security

**Command:**

**Symptom, z/VM Telnet:**
VM TCP/IP Telnet Level 610
SSL Server is not available on local system.

Quitting...bye
TLS/SSL Server: Debugging

**Problem:** Connection to protected application cannot be established

**Analysis:**
Confirm SSL server is running (NETSTAT CONFIG SSL)
Confirm SSL server is listening (NETSTAT CONN or NETSTAT ALLCONN)
Use SSLADMIN QUERY STATUS or NETSTAT CONFIG SSL to determine the current number and maximum number of active sessions
Check SSL server console log for messages
Issue SSLADMIN TRACE CONN
Activate TCPIP tracing (SSL, TCPUP, TCPDOWN) to gather more data
TLS/SSL Server: Debugging

**Problem:** Connection closes due to errors

**Analysis:**

Verify the certificate label is correct:
- `gskkyman` certificate label, in the appropriate database
- TLSLABEL on PORT statement or in application server configuration

Verify that the certificate has not expired
- View certificate information in gskkyman

Verify that the SSL server is accessing the most recent certificate updates (SSLADMIN REFRESH)

Check SSL Server console for messages

Issue SSLADMIN TRACE CONNECTIONS to gather more data
TLS/SSL Server: Debugging

**Problem:** Incorrect input or output inside a secure connection

**Analysis:**
Verify that the subject connection has been established
- SSLADMIN QUERY SESSIONS

Check messages from the SSL server for any problems
- SSLADMIN CLOSECON

Verify that data is flowing correctly through the server
- SSLADMIN TRACE CONNECTIONS DATA
- Try connection again after Trace has been configured
- Consider limiting the trace to a specific IP address / port