IBM TRAINING

Session B37

SECURE SOCKET LAYER

Will J. Roden, Jr.

IBM
SYSTEM z9 AND zSERIES EXPO
October 9 - 13, 2006

Orlando, FL

IBM Systems

Trademarks

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

CICS*
DB2*
DB2 Connect
DB2 Universal Database
DFSMS/MVS*
DFSMS/VM*
e-business logo*
Enterprise Storage Server
ESCON
FICON
GDDM
HiperSockets
IBM*
IBM logo
Language Environment*
MQSeries*
Multiprise*
MVS
NetRexx
OpenEdition*
OpenExtensions
OS/390*
Parallel Sysplex*
PR/SM
QMF
RACF*
RAMAC*
RISC
S/370
S/370
S/390*
S/390 Parallel Enterprise Server
Virtual Image Facility
VisualAge*
VisualGen*
VM/ESA*
VTAM*
VSE/ESA
WebSphere
z/Architecture
zOS
zSeries
z/VM

* Registered trademarks of the IBM Corporation

The following are trademarks or registered trademarks of other companies.

Lotus, Notes, and Domino are trademarks or registered trademarks of Lotus Development Corporation.
Tivoli is a trademark of Tivoli Systems Inc.
Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries
UNIX is a registered trademark of The Open Group in the United States and other countries.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

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.

IBM considers a product "Year 2000 ready" if the product, when used in accordance with its associated documentation, is capable of correctly processing, providing and/or receiving date data within and between the 20th and 21st centuries, provided that all products (for example, hardware, software and firmware) used with the product properly exchange accurate date data with it. Any statements concerning the Year 2000 readiness of any IBM products contained in this presentation are Year 2000 Readiness Disclosures, subject to the Year 2000 Information and Readiness Disclosure Act of 1998.

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.

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 no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment 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 environment.

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.

SSL

- What is SSL?
- How is the VM SSL server set up?
- Steps to make it work
- Linux specifics - start, locations, obtaining control

What is SSL?

- SSL was developed by Netscape to provide secure communications
  - Connection is trusted
    - Certificates authenticate identity
  - Connection is private
    - Cryptographic parameters established during handshake
  - Connection is reliable
    - *Message digest is sent with message*
- Standardized by RFC 2246 (Transport Layer Security - TLS)

Why is this interesting?

- SSL is important in today's marketplace
- Any server running on VM can be secure
  - Update the TCPIP profile
- Can talk with any SSL client that conforms to RFC 2246
  - Telnet clients
  - FTP clients
  - HTTPS clients
- Assistance provided for certificate management

VM TCP/IP support for SSL

- Support for SSL 3.0, TLS 1.0
- Provides security functions for any server
- Certificate database and management
- Provided as a BLACK BOX

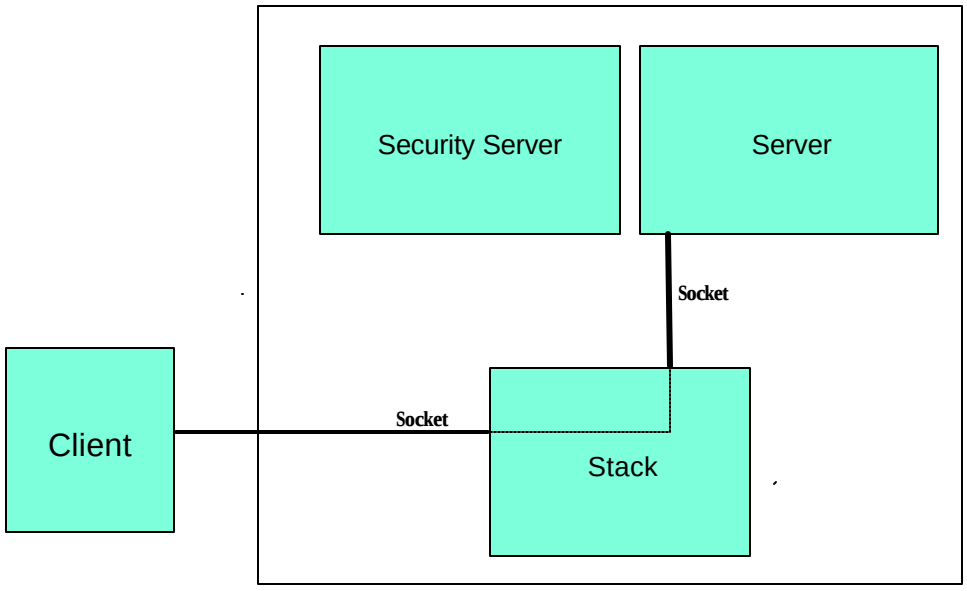
VM TCP/IP support for SSL

- Supports these encryption suites:
  - **FIPS_3DES_168_SHA"**
  - **FIPS_DES_56_SHA"**
  - **RC4_128_SHA"**
  - **RC4_128_MD5"**
  - **3DES_168_SHA"**
  - **RC2_128_MD5"**
  - **DES_EXP1024_56_SHA**
  - **RC4_EXP1024_56_SHA**
  - **RC4_40_MD5"**
  - **RC2_40_MD5"**
  - **DES_56_SHA"**
  - **NULL_SHA"**
  - **NULL_MD5"**
  - **NULL"**

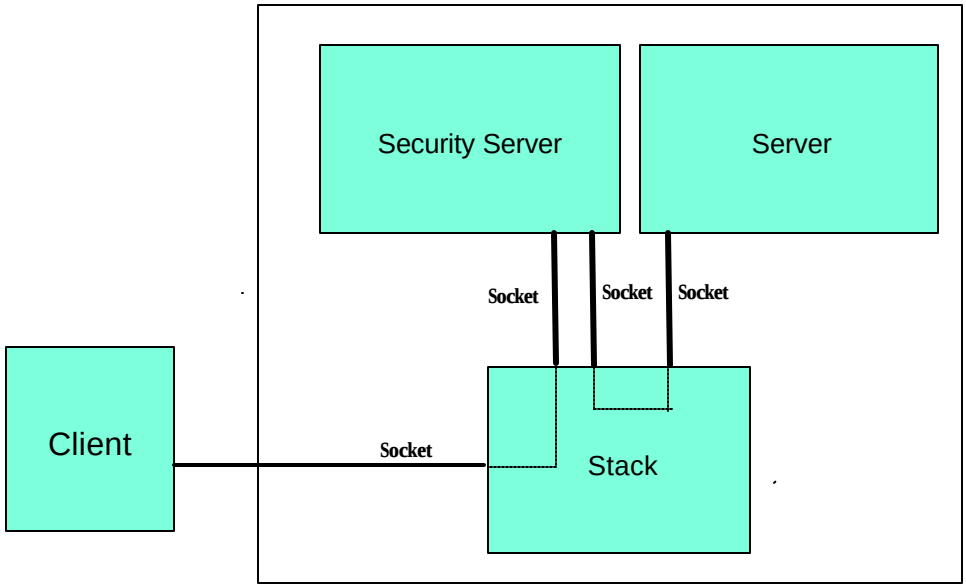
What's not supported

- SSL for VM TCP/IP clients
- Client authentication
- Hardware encryption
- Negotiated security

Client's and Server's View



Stack's View (Reality)



What Happens

- Client Stack SS Stack Server
  (1) Connect (secure port) ----->
  <---------- Accept
  Connect ---------->
  <---------- Accept

  (2) Client Hello ---------->
  <---------- Server Hello
  <---------- Certificate
  <---------- Hello Done
  Key Exchange ---------->
  Chg Cipher ---------->
  Finished ---------->
  <---------- Chg Cipher Spec
  <---------- Finished

What Happens

- Client Stack SS Stack Server

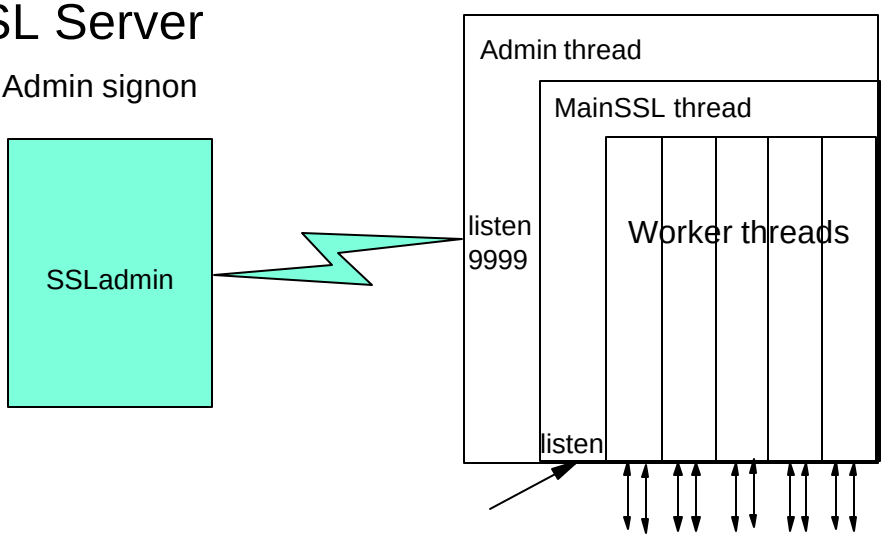
  (3) Send ------ encrypted data ------>
  Send -- decrypted data -->
  <---- unencrypted data --- Send
  <----- encrypted data ---- Send

  (4) Close ---------->
  Close ---------->
  <---------- Close
  <---------- Close

How the Server works

- SSL Server

  Admin signon



How the Server works

- Stack
  - Recognizes that a connection is destined for a secure port
  - Redirects the connection to the Security Server sending:
    - Address of the real server
    - Certificate label to use for authentication
    - Connection number
  - Maintains the illusion

How the Server works

- Security Server
  - Administration thread
    - Listens on a port for commands
      - Server administration
      - Certificate management
    - Commands issued from a CMS virtual machine
    - Only accepts connections from users in the obey list
  - Main SSL thread
    - Obtains a socket
    - Waits for connections
    - Passes connections to a worker thread for handling

How the Server works

- Security Server
  - *Worker thread*
    - Connects to the real server
    - Conducts the handshake with the client:
      - Check for resuming session
      - Sends server certificate for authentication
      - Agrees on protocol version
      - Selects cryptographic algorithms
      - Uses asymmetric encryption to generate shared secrets
      - Generates symmetric keys from shared secrets
    - Decrypts data sent from the client to the real server
    - Encrypts data sent from the real server to the client

How the Server works

- Cache
  - Maintains information from a previous handshake
  - Choices:
    - How many can be saved? (0-4095)
    - How many hours can they kept? (0-24)
- Maxusers
  - Determines how many Worker Threads are set up
- Exempt
  - Cipher Suites that are not to be used
- FIPS - Federal Information Processing Standards
  - Places SSL into FIPS mode

How the Server works

- Trace
  - Set with Initially with DTCPARMS or with SSLadmin.
  - Options:
    - **notrace** - just messages
    - **trace normal** - one message per connection
    - **trace connections nodata** - connection detail
    - **trace connections data** - ... plus every data movement
    - **trace flow** - enter and exit routines with values
    - **trace vmsock** - trace the Linux / VM Stack interface
  - To limit output:
    - ip ip:port :port
    - connection number from *netstat connect*
    - admin and main SSL threads always traced
  - ***And... the Server runs on LINUX***

Why Linux?

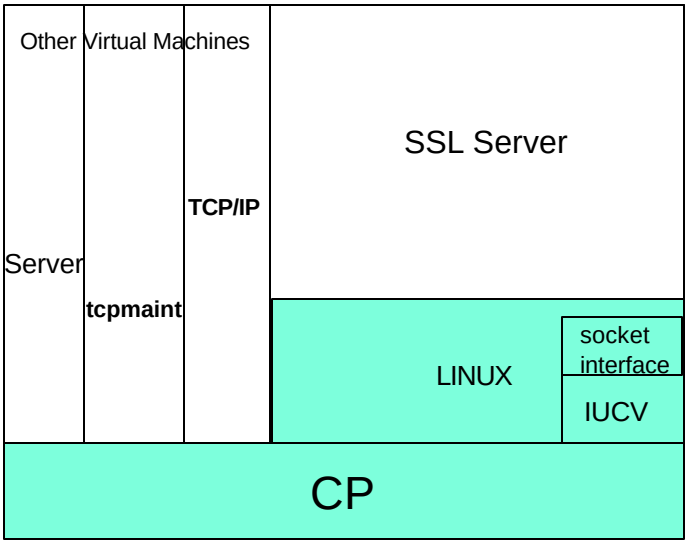
- Made decision to use an IBM-written SSL toolkit:
  - Quicker to marketplace
  - More functionality
  - Better maintainability
  - Easier to enhance
- SSL toolkit written in C++
- Linux provided the capability needed

What Linux?

- **SuSE**
  - SLES 8 31 bit
  - SLES 9 31 bit
  - SLES 9 64 bit
- **Red Hat Enterprise**
  - AS3 31 bit
  - AS3 64 bit
  - AS4 31 bit
  - AS4 64 bit

How it is Set Up on LINUX

- VM



Server on Linux

- Security server looks like any other VM TCP/IP server
  - Does not use the IP stack on Linux
  - Communicates with the VM IP stack using IUCV
  - Installation updates DTCPARMS for the server
  - CMS command interface to communicate with the server
- Enhancements to Linux
  - Specific socket interface layer
  - Specific IUCV device driver
- Proof that Linux can complement VM

How to Set Up the SSL Server

- Install a LINUX
  - IBM GSKit RPM Package (31 bit or 64 bit)
  - SSL RPM Package
- Stack updates
- Start SSL Server
- Certificate Database updates
- Designate Secure Ports (Use OBEYFILE)
- Dynamic Operation

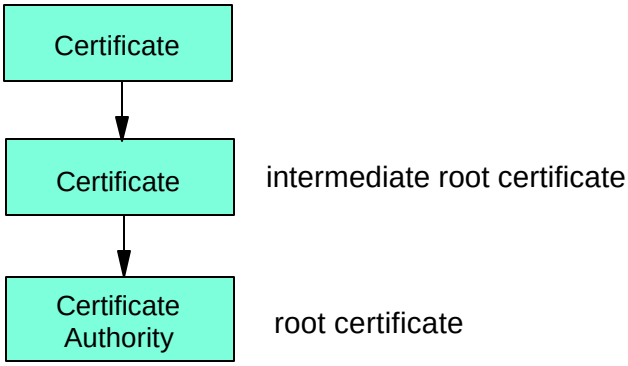
How to Set Up the SSL Server

- Stack updates
  - **PROFILE TCPIP**
    - AUTOLOG SSLSERV 0
    - PORT 9999 TCP SSLSERV
  - **DTCPARMS**
    - MAXUSERS
    - CACHESIZE
    - CACHELIFE
    - EXEMPT ...
    - NOTRACE/TRACE
  - **ETC Services**
    - SSLADMIN 9999/tcp

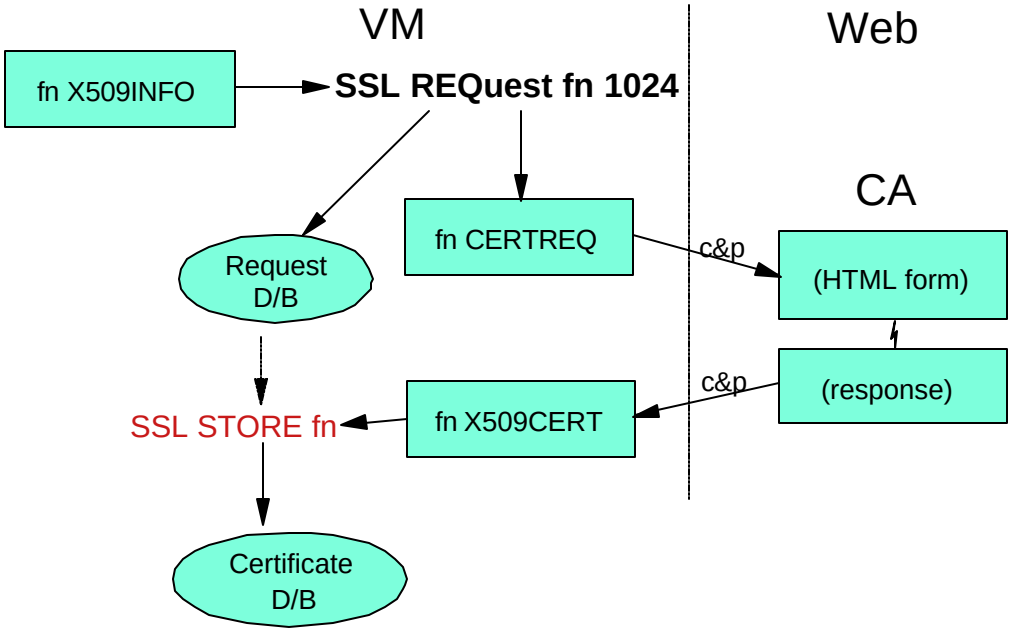
How to Set Up the SSL Server

- Start the Server
  - **Autolog by Stack**
    - Logon SSLSERV
      - DTCPARMS is read
      - VMSSL exec runs
      - admin port is found
      - stack userid is determined
      - funny disk is updated
      - LINUX is IPLed
        - starts VMSSL

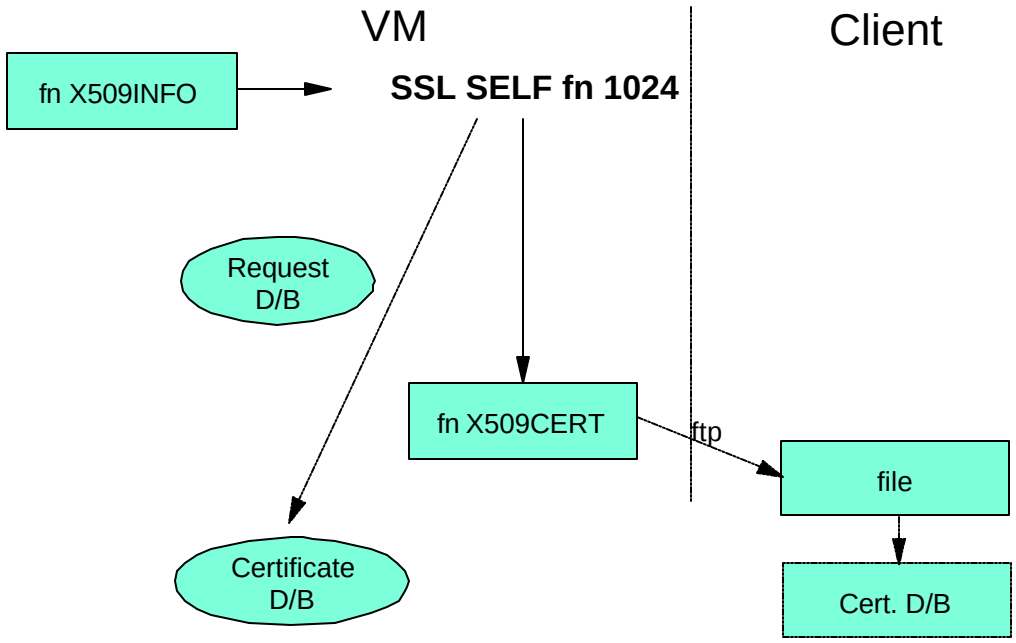
How Certificates work



How to set up Certificates

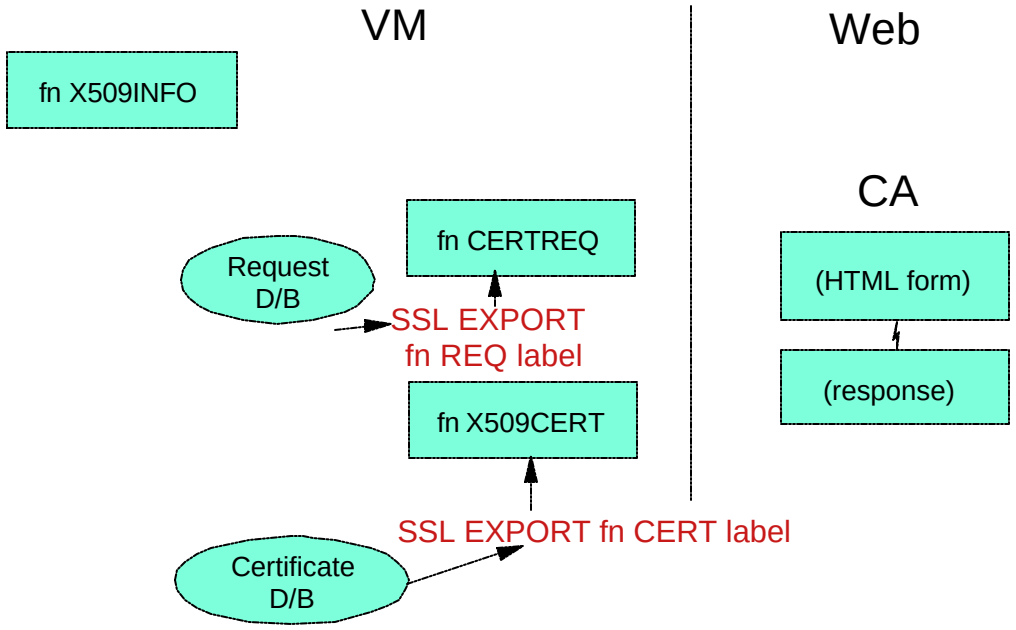


How to set up Self Signed Certificates



IBM Systems

How to Get Certificates Again



IBM Systems

Other SSLadmin Database Commands

- SSLadmin
  - What is there?
    - QUERY REQUEST *
    - QUERY CERTIFICATE *
  - Clean up
    - DELETE REQ label
    - DELETE CERT label
    - REGENERATE - use a new internal password
    - REMOVE - removes certificate database Dangerous!!

How to Set Up the SSL Server

- Re-start the Server
  - **SSLadmin STOP (from tcpmaint)**
    - shutdown LINUX
    - Virtual machine left in CP READ
  - **IPL CMS (from sslserv)**
    - same as: *Start the Server*
  - **Autolog by Stack**

Designate Secure Ports

- TCP/IP profile changes
  - Secure keyword on Port statement
  - Certificate label
    - i.e. PORT 23 SECURE label
- Issue OBEYFILE command or
- restart stack

Dynamic Operation

- SSLadmin - Other Updates
  - QUERY STATUS
  - QUERY CACHE
  - QUERY SESSIONS
  - NOTRACE
  - TRACE NORMAL
  - TRACE CONNECTIONS (data/nodata) (ip:port)
  - TRACE FLOW
  - LOG

Dynamic Operation

- Stack
  - **Netstat changes**
    - For secure connections - Identification of the two connections forming a secure session
  - **To trace stack activity**
    - New process name, SSL

Normal Start

- dtcparms - /opt/vmssl/parms/dtcparms
- startup script - /opt/vmssl/bin/vmsslds s
  - /etc/init.d/vmssld - link to script
- start up dir on SuSE - /etc/rc.d/rc5.d
  - start up - S95vmssl
  - shutdown - K05vmssl
- changing automatic startup.
  - cd /opt/vmssl/bin/
  - modsymlinks -m - to set up vmssl
  - modsymlinks -r - to use Linux stack
  - rm /etc/init.d/vmssl.symlinkdata

Where do I find it in Linux?

- vmssl
  - code - /opt/vmssl/bin
  - No source is provided
- vmsock
  - code - /opt/vmssl/lib
  - source
    - /usr/src/linux/drivers/s390/net/vmsock.c
    - /usr/src/linux/drivers/s390/net/vmsock.h
- log - /var/log/vmssl/vmssl.log
  - alt log: - /var/spool/ssllog

Getting Control

- To Linux:
  - ssladmin stop - stops the vmssl machine.
  - cd /opt/vmssl/bin/
  - rm /etc/init.d/vmssl.symlinkdata
  - ./modsymlinks -r
  - halt
  - #cp i cms (enter twice, sign on as root)
- To VMSSL:
  - cd /opt/vmssl/bin/
  - ./modsymlinks -m
  - halt
  - #cp i cms (enter twice)

Summary

- SSL support for any server on VM
  - No change to the server
  - Easy to configure
  - Connections with SSL-enabled clients
- Certificate database
  - Consolidate certificates and keys in one location
  - Share certificates and keys
  - Obey list authorization
  - CMS command interface for management

Author

- Will Roden, Jr.
  - roden@us.ibm.com
  - http://www.vm.ibm.com/devpages/roden
- SSL web page:
  - http://www.vm.ibm.com/related/tcpip/vmsslinf.html