IBM Software Group

Problem Determination of some common WMQ scenarios on z/OS®

Beverly Kingsley
WebSphere®
MQ Level 2 Support

ON DEMAND BUSINESS

Agenda

- Whether you're new to WebSphere MQ, or already an experienced user, you'll want to know what steps to take to perform basic and enhanced troubleshooting.
- The scenarios covered will include performance problem determination, SSL configuration, and some typical problems with channels.
- This session will also deal with the types of documentation you need to perform problem determination, and what you need to escalate the problem to IBM MQ support.

Topics

1. Scenarios illustrating Performance Problems
2. Typical Channel Problems diagnosed
3. Guidelines to basic MQ SSL channel configuration and setup
4. Documentation required to escalate the problem to IBM MQ support

Performance Problems
Before you involve MQ support…..

- Basic things to check first:
  - Ensure you have followed the recommendations in the performance SupportPacs
  - Make use of monitoring tools such Omegamon and Strobe
  - Search for performance APARs/PTFs
  - Search for performance Technotes
  - Refer to performance Redbooks and White papers
  - Involve your network folks if you think the performance problem is in the network
  - If you cannot find the problem then MQ support may be able to help

Performance Problems
1 - High CPU in Channel Initiator Address Space

- Customer observation:
  - High CPU spotted using Monitoring Tool
  - SYSTEM.CHANNEL.SYNCQ had 2452 messages
  - Strobe report confirmed high CPU in CHINIT address space
  - CHIN joblog showed no relevant errors
  - Customer took dumps of MSTR and CHIN address spaces

Performance Problems
1 - High CPU in Channel Initiator Address Space

- MQ support analysis:
  - IPCS analysis of the dump showed:
    - CHINIT traversing many pages and holding and releasing locks and latches looking for a specific message on SYSTEM.CHANNEL.SYNCQ
    - The Syncq had 308 lock requests
    - Queue manager dump (w/o trace) showed the persistent scavenger holding many locks and latches
    - The SYSTEM.CHANNEL.SYNCQ had 2452 messages and was defined with INDXTYPE=NONE on page set 0
    - Other SYSTEM queues were also defined on Pageset 0
    - Pageset 0 was mapped to Bufferpool 0
    - Bufferpool 0 was defined with 4000 4K buffers

Performance Problems
1 - High CPU in Channel Initiator Address Space

- MQ support recommendations:
  - Alter SYSTEM.CHANNEL.SYNCQ INDXTYPE=NONE to INDXTYPE=MSGID
  - Ensure sample job CSQ4INSX is run
  - Ensure all SYSTEM objects are defined to Pageset other than 0
  - Increase Buffer pool 0 allocation
    - See SupportPac MP16 for tuning recommendations
    - CSQP020E is issued on the console if a buffer pool is too small
    - In V6, CSQY220I is issued periodically to indicate the amount of storage in use and the amount remaining

Performance Problems
2 – Correct interpretation of log buffer statistics in MP1B

- Customer observation:
  - Using SupportPac MP1B to interpret MQ performance after application changes
  - Noticed that values of "Log Write Force“ (QJSTWRF) and "Read Buf" (QJSTRBUF) had significantly increased
  - Queue Level statistics showed no significant changes
  - SMF Reports showed many commit, put, get, open and close statistics at application and queue level, but no "backout" statistics
  - SupportPac MO12 (Log Extract program) showed commits, but no backouts
    - In V6, CSQ1LOGP contains the extract function
    - In V5, MO12 provides the extract function

Performance Problems

2 – Correct interpretation of log buffer statistics in MP1B

- MQ support comments:
  - An increase in WRF records is usually due to an increase in traffic
    - More local applications running
    - An increase in traffic from the CHIN
    - An increase in client activity
  - Putting or getting more messages out of syncpoint cause an implicit commit which will increase the number of log forces
  - Backouts as well as commits contribute to log forces. These backouts can be explicit or implicit. If the application issues an MQBACK (explicit) then it will show up in the accounting class(3)

Performance Problems

2 – Correct interpretation of log buffer statistics in MP1B

- IBM support comments (2):
  - Channels:
    - A commit initiates a log write force
    - A commit is issued when there are no more messages, or the number of messages reaches the BATCHSZ value
    - With busy channels, it is possible to get more messages in a batch, and so the number of commits may go down
    - If a channel stops mid batch, messages are implicitly rolled back
      - This does not show up in the accounting data since the application did not perform an explicit backout. When the channel restarts, messages are sent again

Performance Problems

2 – Correct interpretation of log buffer statistics in MP1B

- MQ support recommendations:
  - Check MQ joblogs and SYSLOG for any messages relating to channels stopping and restarting abnormally
  - Make sure applications commit messages appropriately
  - If necessary pinpoint which applications incur backouts by extracting data from the MQ log datasets
  - Note that the values of "Log Write Force“ (QJSTWRF) and "Read Buf" (QJSTRBUF) imply backouts from both application and internal queue-manager processing

Performance Problems
3 – CICS transaction slowdown during peak period

- Customer observation:
  - SMF accounting data showed some response times of 12-14 seconds for MQPUT1 requests
  - Problem worsened during times of heavy volume
  - Several CICS regions and various transactions were affected by the problem
  - Other MQ applications appeared to be unaffected
  - MSTR job showed CPU usage high
  - Customer submitted dumps of MSTR, CHIN and CICS address spaces
  - SMF 116 Class 3 data was also submitted

Performance Problems

3 – CICS transaction slowdown during peak period

- MQ support analysis:
  - CICS dispatcher showed several tasks in MQ TASKSWCH
  - Systrace showed EXT 1005 and CLKC interrupts in z/OS virtual storage with the PSW pointing into MQ security modules
  - High use of SP229 KEY7 in the local subpool
  - Dump showed 16844 PHBs
    - These PHBs were for the security RMID and contained HRAI blocks
  - The SMF 116 Class 3 data showed the delays during MQPUT1 processing were due to contention for many security latches associated with a single userid
  - Timeout and Interval settings were at default values

Performance Problems

3 – CICS transaction slowdown during peak period

- MQ support recommendations:
  - Tune the SECURITY settings in MQ. The defaults are TIMEOUT=54 and INTERVAL=12
  - Assign a unique userid to each CICS region. Alternatively, the userid could be given CONTROL access to RESLEVEL
  - If usage varies widely between frequent users and those who only sign on once, consider splitting the work for those users to different queue managers

Channel problems
Before you start…..(1)

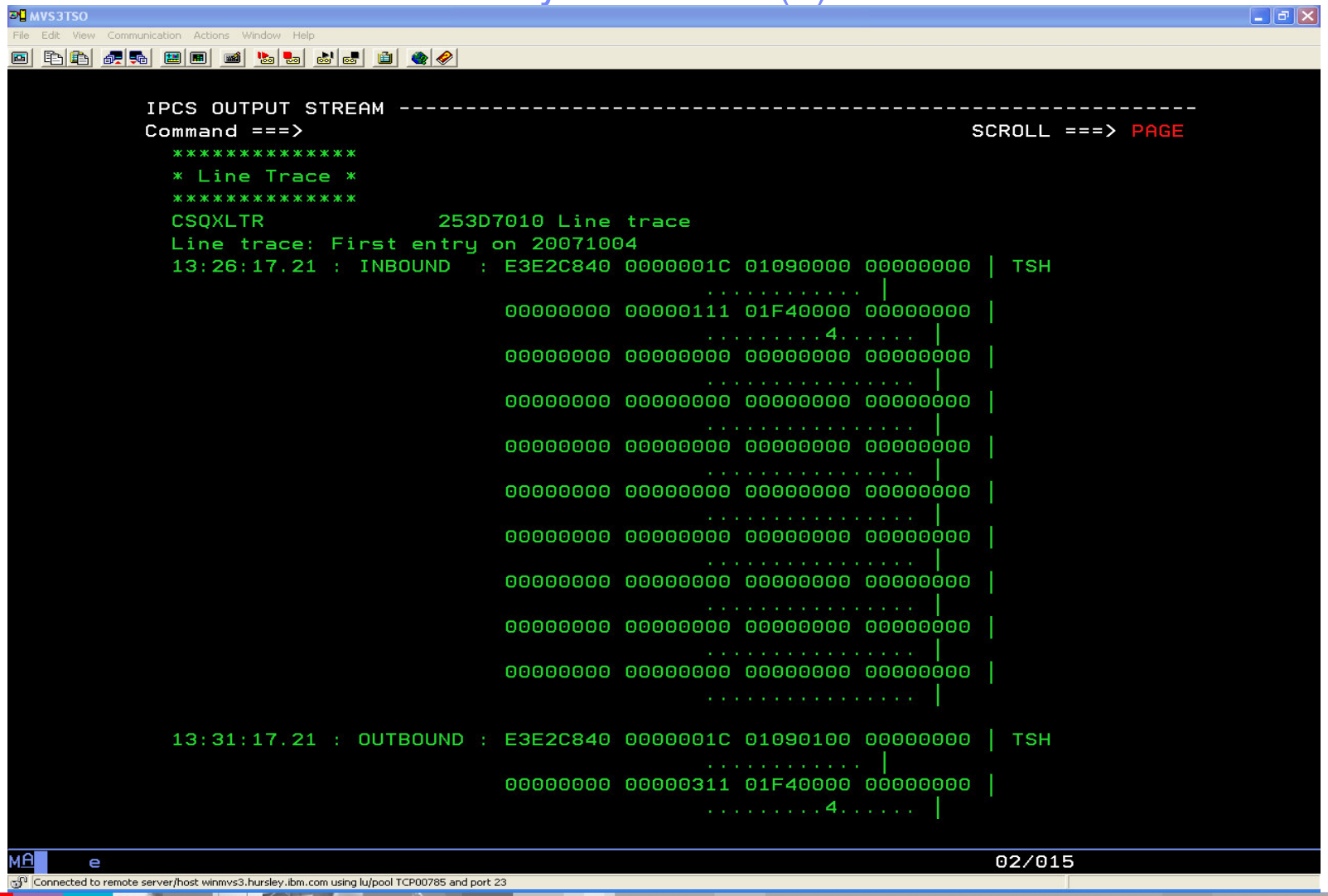
- Distributed queuing messages
  - Are issued to the CHINIT joblog
  - Almost all messages are CSQXnnnx
  - 'X' is the component letter that represents the CHINIT
- CSQX20nx messages:
  - An error in the transport layer
  - Usually not an MQ problem
- Most CSQX6nnx messages are SSL reported errors/return codes
  - An error in system SSL
  - Usually not an MQ problem

Channel problems
Before you start…..(2)

- CHINIT Line Trace
  - Included in the CHINIT address space
    - Cannot disable it
    - Applies to TCP/IP® and LU6.2
  - There are 2 ways to capture it:
    1. Adding CSQSNAP DD statement in the CHINIT JCL
       - START TRACE(CHINIT) CLASS(4) IFCID(202)
       - DISPLAY CHSTATUS(channel) SAVED
    2. Take Dump of CHINIT address space
       - Format the dump with the IPCS command:
         'verbx csqxdprd 'subsys=ssid,chst=3'
  - Contains last INBOUND and OUTBOUND flows on each running channel
  - Ideal for problem where channels appear to be hung or if the problem is in MQ

Channel problems

Before you start…..(3)



Channel problems
1 – Sender channel hang after network outage

- Customer observation:
  - Sender channel on z/OS going to Receiver on Windows®
  - Network error on Windows side
    - The Receiver channel ended abnormally
  - No errors were reported on z/OS side
  - Sender channel seemed to be stuck in a wait
  - Messages remained on the XMIT queue
  - STOP MODE(FORCE) cleared the channel

Channel problems

1 - Sender channel hang after network outage

- MQ support analysis:
  - The following XPARMS settings was found:
    - ADOPTMCA(YES)
    - ADOPTCHK(QMNAME)
    - TCPKEEP(NO)
  - Sender channel definitions were:
    - KAINT(AUTO)
    - HBINT(300)

Channel problems
1 - Sender channel hang after network outage

- MQ support recommendations:
  - If the connection is broken, a combination of the following parameters can be used to clear the channel:
    - Keepalive: TCPKEEP(YES) XPARMS parameter
      - For V6 this is a Qmgr attribute
    - KAINT: channel-level TCP Keepalive interval in seconds
    - HBINT: channel-level Sender check that Receiver is available in seconds when there are no messages on the queue

Channel problems
2 – Orphaned SVRCONN channels

- Customer observation:
  - Users frequently found they had orphaned SVRCONN channels which couldn't be cleared.
  - STOP MODE(FORCE) did not clear the channel

Channel problems
2 – Orphaned SVRCONN channels

- MQ support analysis:
  - Possible reasons:
    - Indefinite or excessive wait by the client for the message
    - Message may not yet have been committed by the putting application
    - Connection broken:
      - TCP/IP will provide MQ with an error response reported on the MQ job log
    - The client may not be performing a disconnect (application error)
    - For V6: Heartbeat is not specified or incorrectly specified
  - STOP MODE(FORCE) may not clear the status

Channel problems
2 – Orphaned SVRCONN channels

- MQ support recommendations:
  - For V5:
    - If the application is issuing an MQGET with an unlimited or excessive wait interval, there is nothing MQ can do to override this wait interval and this could result in orphaned SVRCONN channels.
      - Use much shorter wait intervals
    - If the connection is broken, or the client machine is powered off, TCP/IP Keepalive can be used to clear the channel.
    - Set KeepAlive Interval, KAINT to an appropriate value such as 300 secs
    - For KeepAlive to be used, CSQ6CHIP's TCPKEEP must be set to YES
  - For V6:
    - In addition to the above:
    - Specify an appropriate value for heartbeat (HBINT) on the SVRCONN definition

Channel problems
3 – Channel will not start

- Customer observation:
  - Messages were building up on the Xmit queue
  - CSQX526E was issued on CHINIT joblog
  - Xmit queue had been set to GET(DISABLED)
  - Triggering had been turned off

Channel problems
3 – Channel will not start

- MQ support analysis:
  - MSTR and CHINIT joblogs were examined
    - CSQX517E was found to have been issued for the same channel at CHINIT startup
    - SyncQ message for this channel may be duplicated
  - Possible reasons:
    - User may have cleared the SyncQ
    - User may have changed the channel IP address
    - User may have manually re-set the receiver channel
    - User may have re-defined his sender channel

Channel problems
3 – Channel will not start

- MQ support recommendations:
  - Ensure sequence errors are not returned by:
    - Keeping both sides “clean” when making changes to the configuration
    - In the event of the problem, try the following:
      - RESET CHANNEL to reset sequence number
      - Run CSQ4SYNC / CSQ4SUTL utility from Level 2

Methods for diagnosing SSL issues
1 – Hints when setting up SSL for the first time

*** Keep it SIMPLE ***

- When setting up SSL channels for the first time, most problems encountered are either MQ configuration errors or problems with the SSL setup
- USE SupportPac MO04 as a configuration wizard
- Use self-signed certificates to start with
- Understand the client/server SSL terminology
- Use ONE-way authentication to start with
- Don't use Peer name checking to start with
- Don’t use certificate revocation lists to start with

Methods for diagnosing SSL issues
2 – Fundamental diagnosis guidelines

- Always check logs at both ends
- Look in CHINIT joblog and SYSLOG, and equivalent at other end
- Take a note of any SSL error codes
- Check whether problem is at channel startup, or with message flow
- Always carefully check contents of keyring/key repository
- Carefully check all SSL configuration parameters at both ends

- If you're getting an error trying to use SSL, there will be an error message somewhere

Methods for diagnosing SSL issues
3 – Troubleshooting checklist(1)

- Check Queue Manager settings:
  - SSL turned on
    - Keyring correctly named - SSLKEYR
    - Number of SSL tasks - SSLTASKS
- Check Channel setting:
  - Encryption set on channels - SSLCIPH
  - One-way or two-way authentication? - SSLCAUTH
  - Are you using Peer-level checking? - SSLPEER

Methods for diagnosing SSL issues
3 – Troubleshooting checklist(2)

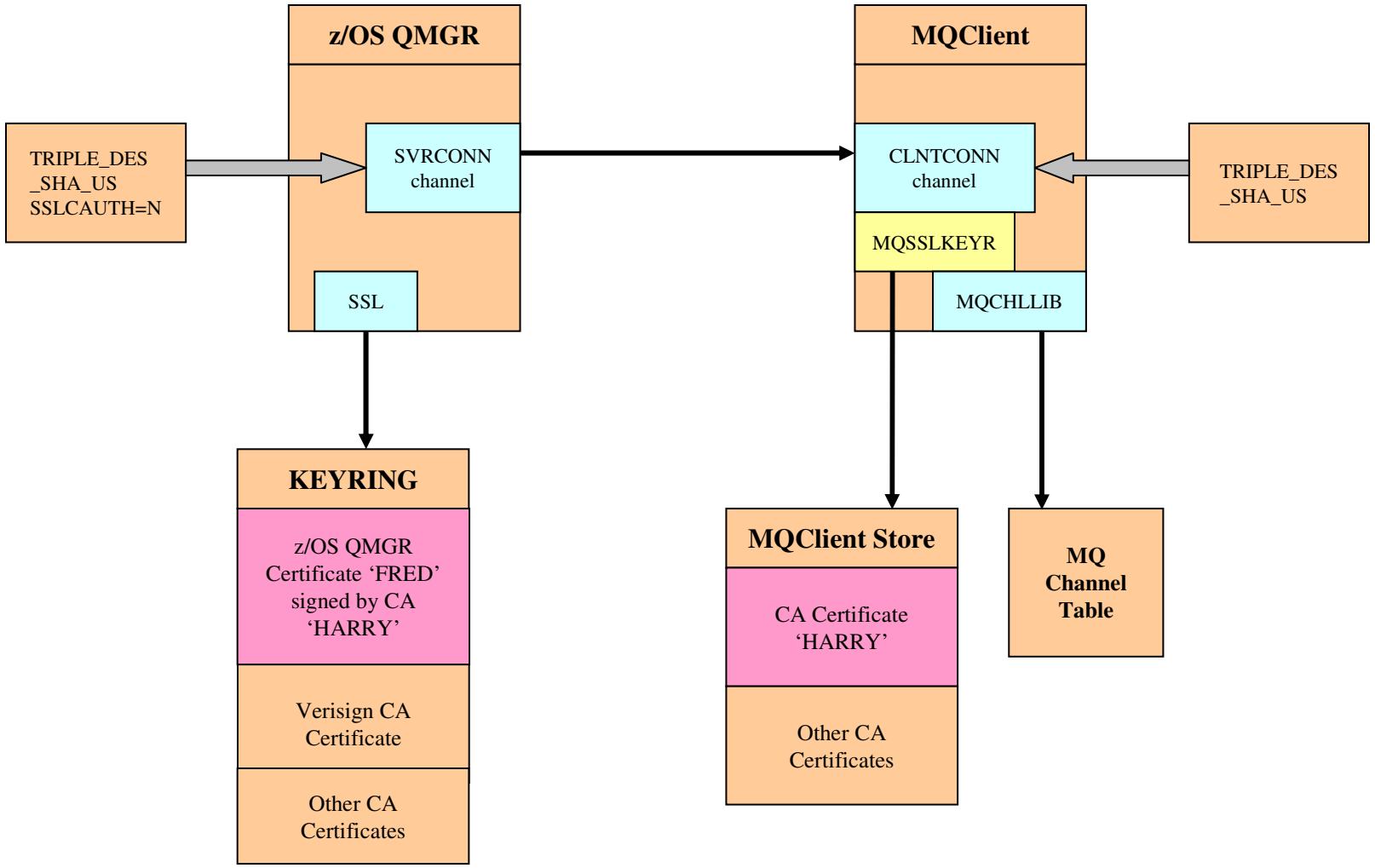
- Check keyring contents:
  - Check queue mgr personal certificate
    - Label name correct and according to MQ standard? – ibmWebSphereMQqmgr-name (for z/OS)
    - Signer/issuer details (DN)
    - Is this a self-signed certificate?
    - Is it set to USAGE=HANDSHAKE?
  - Is the signer certificate available at the other end?
    - CA or self-signed qmgr personal?
    - If it's a CA certificate, is it part of a chain?
    - Is the chain complete?
    - Is the CA certificate, or any in the chain, expired?
    - Is it set to USAGE=CERTAUTH?

Methods for diagnosing SSL issues
3 – Troubleshooting checklist(3)

- Other things to check:
  - The userid owning the key ring should be the userid of the CHINIT unless it is shared
  - Does CHIN userid have read access to IRR.DIGTCERT.LISTRING in FACILITY class?
  - Both ends of the channels must specify the same cipher specification
  - Have you made any changes to the keyring?
    - On V531 you need to recycle the CHIN
    - On V6 it's dynamic
  - Is ICSF and hardware crypto involved?
  - Are you using a Queue Sharing Group?

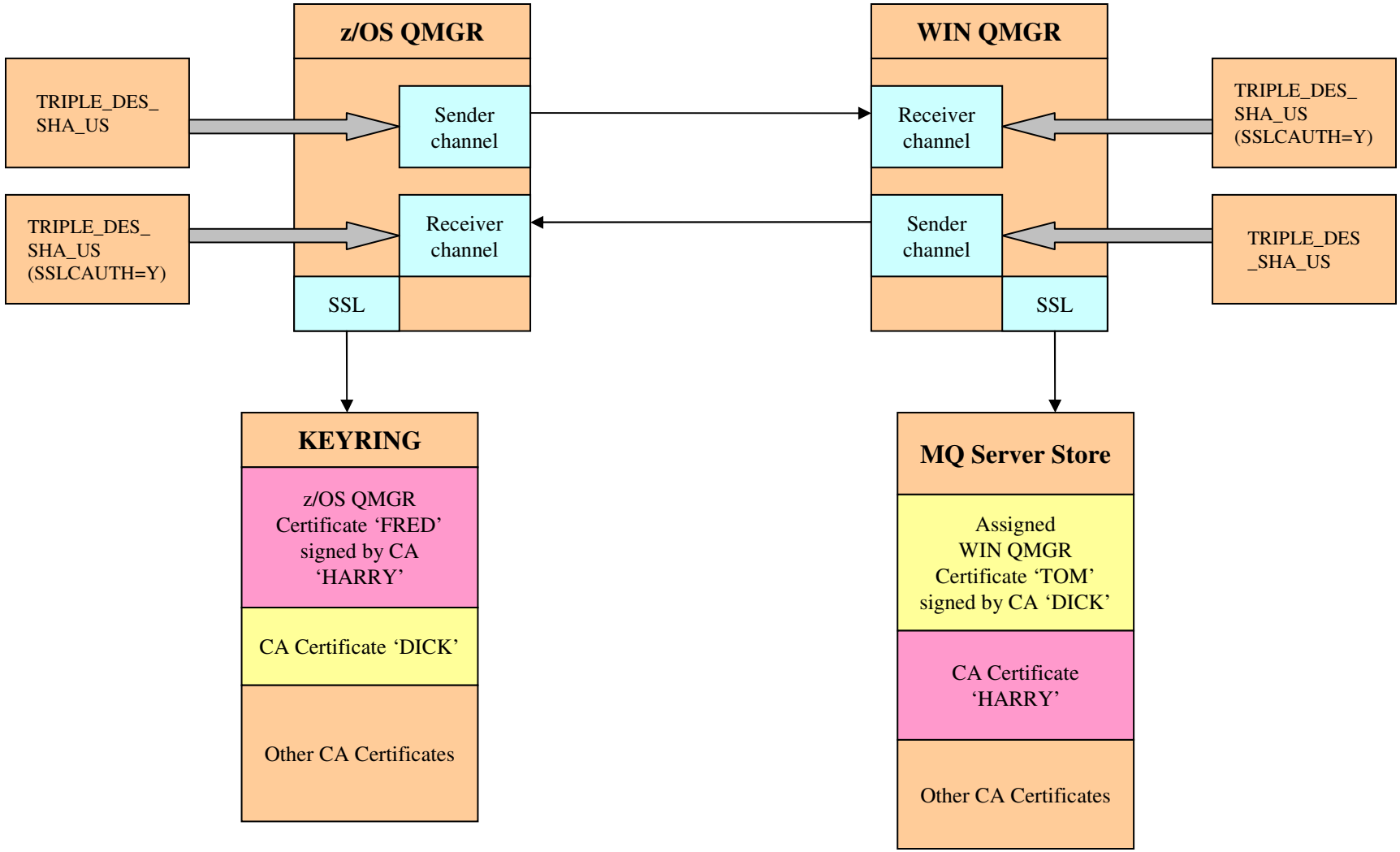
Methods for diagnosing SSL issues

4 – SSL Topology: 1-Way Authentication using an MQ Client



Methods for diagnosing SSL issues

4 – SSL Topology: 2-Way Authentication between MQ Servers



Escalating the problem to Level 2

1 - For performance problems

1. If possible increase MQ trace table to maximum size:
   - If trace is running, issue
     /cpf STOP TRACE(GLOBAL) DEST(RES) CLASS(*) RMID(*)
   - /cpf SET SYSTEM TRACTBL=999
2. Start internal MQ trace as follows:
   /cpf START TRACE(GLOBAL) RMID(*) CLASS(*) DEST(RES)
3. Take Dump of MQ during problem period as follows:
   DUMP COMM=(dump title)
   Rnn,JOBNAME=(ssidMSTR,ssidCHIN),cont
   Rnn,DSPNAME=('ssidCHIN'.CSQXTRDS),cont
   Rnn,SDATA=(CSA,RGN,PSA,SQA,LSQA,TRT,SUM),END
4. Copy ssidMSTR and ssidCHIN job logs
5. Copy any monitoring reports

Escalating the problem to Level 2
2 - For channel problems(1)

- CSQX20nE messages
  - Error condition returned from the transport layer
  - A dump on its own may not show the root of the problem
  - Slip Trap on the CSQX20nE is required:

```
SLIP SET,MSGID=CSQX208E,
JOBNAME=ssidCHIN,A=SVCD,
JOBLIST=(ssidMSTR,ssidCHIN,tcpproc),
DSPNAME=('tcpproc'.TCPIPDS1,'ssidCHIN'.CSQXTRDS),
SDATA=(CSA,RGN,PSA,SQA,LSQA,TRT,SUM),MATCHLIM=1,END
```

  - MSTR and CHIN tracing should be turned on by either:

```
TRACSTR and TRAXSTR to YES
START TRACE(GLOBAL) DEST(RES) CLASS(*) RMID(*)
```

Escalating the problem to Level 2

2 - For channel problems(2)

- For TCP/IP channels
  - Get TCP/IP ctrace and packet trace per doc APAR II12136
  - Turn on MQ internal trace:
    - /cpf START TRACE(GLOBAL) DEST(RES) CLASS(*) RMID(*)
  - Create the problem
  - Take dump of MSTR, CHINIT and TCP and their dataspaces:
    - DUMP COMM=(Dump Title)
    - R xx,SDATA=(CSA,SQA,RGN,TRT,GRSQ),CONT
    - R xx,JOBNAME=(ssidMSTR,ssidCHIN,TCPIP),CONT
    - R xx,DSPNAME=('tcpproc'.TCPIPDS1,'ssidCHIN'.CSQXTRDS),END
  - Stop the MQ trace after the dump has been taken:
    - /cpf STOP TRACE(GLOBAL)
    - /cpf STOP TRACE(CHINIT)

Escalating the problem to Level 2

2 - For channel problems(3)

- For LU62 / SNA channels
  - The logical flow of the data is:

    MQ -- APPC API -- VTAM --***NETWORK*** -- SNA -- APPC API -- MQ
  - On the z/OS side:
    - Start the APPC CTRACE:

      TRACE CT,256M,COMP=SYSAPPC

      R xx,OPTIONS=(GLOBAL),END
    - Take dump during problem
    - Stop the APPC CTRACE:

      TRACE CT,OFF,COMP=SYSAPPC
  - On the distributed side:
    - Collect line trace and an API trace
  - Make sure doc on both sides is collected at the same time
  - Get your networking folks involved

Escalating the problem to Level 2
3 - For SSL channel problems(1)

- CSQX6nnE messages
  - CHIN tracing should be turned on by either:

    Set TRAXSTR to YES

    START TRACE(CHINIT) DEST(RES) CLASS(*) RMID(*)
  - If Slip on CSQX6nnE is required:

    SLIP SET,MSGID=CSQX208E,

    JOBNAME=ssidCHIN,A=SVCD,

    JOBLIST=(ssidMSTR,ssidCHIN,tcpproc),

    DSPNAME=('ssidCHIN'.CSQXTRDS),

    SDATA=(CSA,RGN,PSA,SQA,LSQA,TRT,SUM),MATCHLIM=1,
    END
  - Take documentation from both sides of the SSL connection

Escalating the problem to Level 2

3 - For SSL channel problems(2)

- For problems with the keyring:
  - If you need to trace SSL access to the keyring:
    - Start trace
      1. /S GSKSRVR
      2. /TRACE CT,WTRSTART=GSKWTR
      3. /TRACE CT,ON,COMP=GSKSRVR
      4. /R NN,JOBNAME=(ssidCHIN),OPTIONS=(LEVEL=255),WTR=GSKWTR,END
    - /cpf START QMGR
    - Start channel initiator
    - Stop trace
      1. /TRACE CT,OFF,COMP=GSKSRVR
      2. /TRACE CT,WTRSTOP=GSKWTR
      3. /P GSKSRVR
    - Save trace dataset

Escalating the problem to Level 2

3 - For SSL channel problems(3)

- For problems during initial handshake:
  - If you need to trace SSL handshaking:
    - Start trace:
      1. /S GSKSRVR
      2. /TRACE CT,WTRSTART=GSKWTR
      3. /TRACE CT,ON,COMP=GSKSRVR
      4. /R NN,JOBNAME=(ssidCHIN),OPTIONS=(LEVEL=255),WTR=GSKWTR,END
    - Start problem SSL channel:

      /cpf START CHANNEL(channelname)
    - Stop trace:
      1. /TRACE CT,OFF,COMP=GSKSRVR
      2. /TRACE CT,WTRSTOP=GSKWTR
      3. /P GSKSRVR
    - Save trace dataset

Escalating the problem to Level 2

3 - For SSL channel problems(4)

- To format the saved trace under IPCS:
  - Ensure the following load library is in STEPLIB:
    - hlq.SGSKLOAD for z/OS 1.6 and below
    - hlq.SIEALNKE for z/OS 1.7 and above
  - Issue the IPCS command:
    - CTRACE COMP(GSKSRVR) LOCAL FULL
- Check the flow against the handshaking logic in the WMQ Security manual

Useful WMQ web sites

- Search for known WMQ problems and migration information, and sign up to receive customized weekly e-mails at:

  http://www.ibm.com/software/integration/wmq/support/

- WMQ manuals are at:

  http://www-306.ibm.com/software/integration/wmq/library/library6x.html

- Submit PMRs electronically using the ESR tool at:

  http://ibm.com/software/support/probsub.html

- Order the WebSphere MQ product at:

  http://www.ibm.com/support/docview.wss?uid=swg21212797

  Or it can be done through the Shop zSeries® process

- PSP Buckets are at:

  http://www-1.ibm.com/support/docview.wss?rs=171&uid=swg21201593

Additional WebSphere Product Resources

- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at: www.ibm.com/developerworks/websphere/community/
- Learn about other upcoming webcasts, conferences and events: www.ibm.com/software/websphere/events_1.html
- Join the Global WebSphere User Group Community: www.websphere.org
- Access key product show-me demos and tutorials by visiting IBM Education Assistant: www.ibm.com/software/info/education/assistant
- Learn about the Electronic Service Request (ESR) tool for submitting problems electronically: www.ibm.com/software/support/viewlet/probsub/ESR_Overview_viewlet_swf.html
- Sign up to receive weekly technical My support emails: www.ibm.com/software/support/einfo.html

Questions and Answers