GSE REGION BENELUX,
Enterprise Systems
Security Group

TCP/IP support for RACF remote sharing

Speaker name: Philippe RICHARD (philippe_richard@fr.ibm.com)

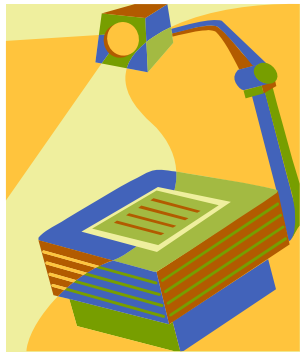
Are you in the right room ?
A simple quizz to start with

1. When was RACF RRSF initially introduced with ?
   - z/OS
   - MVS/ESA
   - OS/390
2. What is the meaning of RRSF ?
   - Resilient Reliable Security Feature
   - RACF Remote Sharing Facility
   - Really Remote Science Fiction

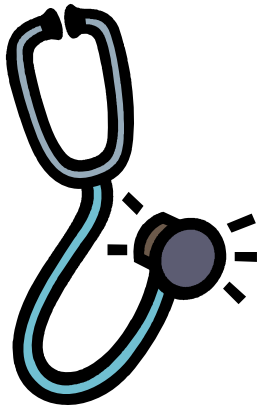
Answers: (did you pass the test ?....No ?)

1. **September 1995,** Version 2 Release 2 (OS/390 R1 security server)
2. **RRSF**: RACF Remote Sharing Facility

TCP/IP support for RACF remote sharing



- Problem
  - Clients do not have the APPC and VTAM skills required to set up and maintain an RRSF network.



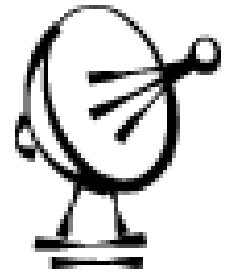
- Solution
  - RRSF will support TCP/IP (IPV4 only) as an alternate transport protocol.

- Benefit
  - Clients already have the skills necessary to maintain a TCP/IP network. In addition, stronger cipher algorithms are supported to protect the data while it crosses the network.

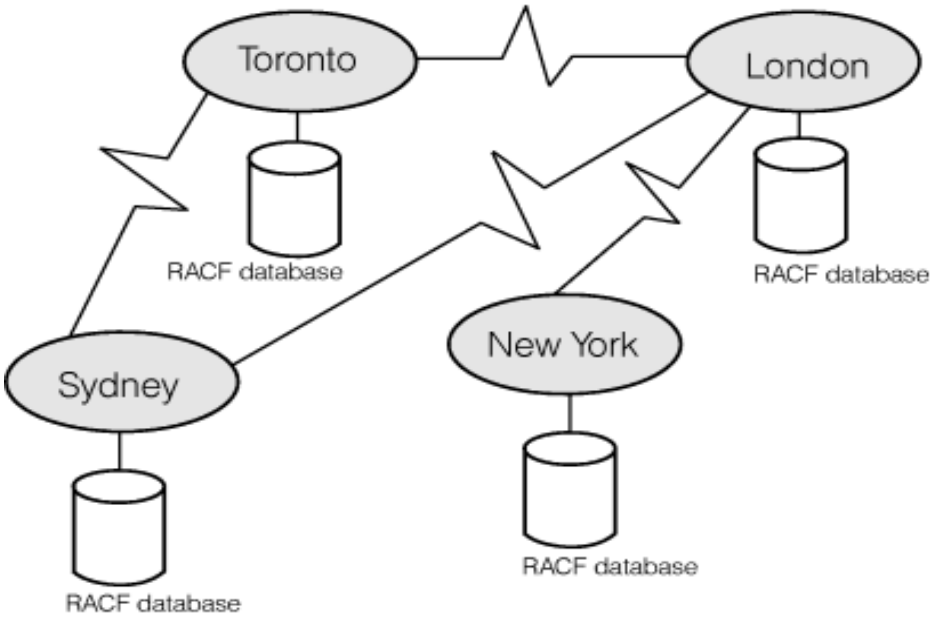
What is RRSF?

- The RACF remote sharing facility (RRSF) allows RACF to communicate with other z/OS systems that use RACF, allowing you to maintain remote RACF databases.
- Benefits of RRSF support for the security administrator include:
  - Administration from anywhere in the RRSF network
  - User ID associations
  - Automatic synchronization of databases
- RRSF is designed in roughly three layers:
  - **Application layer:** Administrative commands and profiles
  - **Presentation layer:** Command execution and return of command output and error and informational messages
  - **Transport layer:** Communication protocol used to transmit requests
- This enhancement deals exclusively with the transport layer.

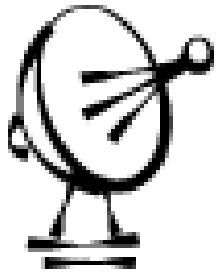
The RRSF network



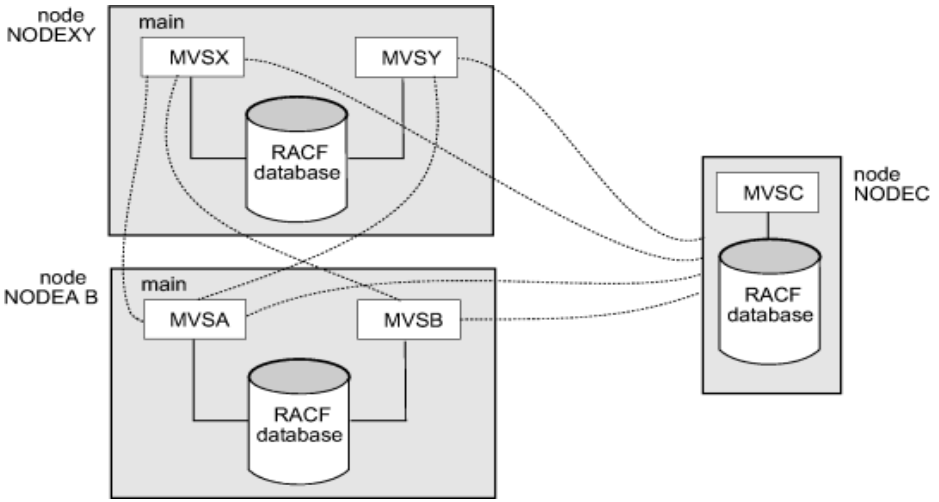
- Consists of *nodes*
  - Local node: The one you are logged on to at the moment
  - Remote nodes (all the others)
  - Local node can run in “local mode” where there are no remote nodes
- The `TARGET` operator command is used to define, modify, delete, and list nodes, as well as to de/activate them.
- `TARGET` commands are contained within the RACF parameter library and are executed automatically when the RACF subsystem starts.
- The RACF parameter library member is specified in your started procedure JCL.
- RACF parameter library members can be “chained together” using the `SET INCLUDE(xx)` command.



Multisystem node

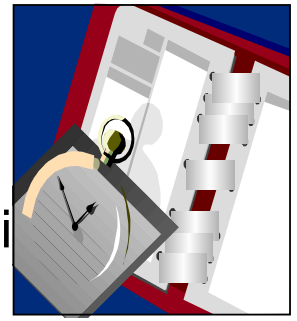


- A set of systems sharing an RACF database (can be in a SYSPLEX or simply on shared DASD)
- Managed with the TARGET command by specifying both NODE and SYSNAME
- All single system nodes (SSNs) send requests only to the *main* system of an MSN
- All peer systems of an MSN send requests only to SSNs and to the MAIN systems of remote MSNs
- Peer systems do not speak with each other, and do not speak with non-MAIN systems of remote MSNs



Sample RRSF network containing two multisystem nodes and a single system node

Workspace data sets (that is, checkpoint files)



- VSAM data sets that RACF uses to temporarily hold data that RACF i sending from one node to another.
- RACF deletes data from the workspace data sets when it receives confirmation that the data has been successfully processed at the receiving node.
- RACF uses two workspace data sets, the INMSG data set and the OUTMSG data set, for the local node and for each of its remote nodes.
  - The INMSG data set is used to temporarily hold requests that are being sent to the local node from itself or from a remote node (for example, commands directed to the local node or output from RACF commands, application updates, and password changes that were directed to a remote node).
  - The OUTMSG data set is used to temporarily hold requests that are being sent to a remote node (for example, commands, application updates, and password changes directed from the local node or output to be returned to a remote node).
- Requests are queued to the files while a connection is DORMANT. Queued work is sent when the connection becomes OPERATIVE ACTIVE.
- Requests are “casually encrypted” while checkpointed.

Usage

- To define and activate a connection using the TCP protocol, you will:
  - Set up all that TLS stuff in Communication Server
  - Establish a socket listener on the local node

```
TARGET NODE(LOCNODE) PROTOCOL(TCP) OPERATIVE

IRRC054I (<) RACF REMOTE SHARING TCP LISTENER HAS BEEN
  SUCCESSFULLY  ESTABLISHED.
```

  - Specify the remote host and make it operative

```
TARGET NODE(REMOTE)
  PROTOCOL(TCP(ADDRESS(remote.pok.ibm.com)))
PREFIX(SYS1.RRSF) WORKSPACE(VOLUME(VOL001)) OPERATIVE

IRRI027I (<) RACF COMMUNICATION WITH TCP NODE REMOTE HAS
  BEEN SUCCESSFULLY ESTABLISHED USING CIPHER ALGORITHM 35
  TLS_RSA_WITH_AES_256_CBC_SHA.
```

- That is all. As simple as that !

Implementation steps

- The implementation steps are roughly the following:
  - Add OMVS segment and UID to RACF subsystem user ID and OMVS segment and GID to its default group.
  - Deploy digital certificates/key rings which are used to authenticate RRSF servers to each other using the TLS protocol.
  - Enable the AT-TLS policy required for RRSF connections (sample provided).
  - Permit the RACF subsystem identity to the necessary resources.
  - Using the TARGET command, establish a socket listener on each RRSF node.
  - Using the TARGET command, activate the connection from both sides of each connection.
  - Harden the TARGET commands in your RACF parameter library.

Setup: OMVS segment for the RACF subsystem

- Use of the TCP protocol requires the use of sockets, which requires UNIX System Services.
- Assign an OMVS segment with a UID to the RACF subsystem user ID using the ALTUSER command.
- Assign an OMVS segment with a GID to its default group using the ALTGROUP command.
- If you do not, you will see an error message when the socket listener attempts to start.
  - Just assign the OMVS segments, and make the local node OPERATIVE again.
  - You do not have to restart the RACF subsystem!

Setup: Digital certificates (1 of 3)

- Network entities authenticate to each other by way of the trust policy established by digital certificates.
  - "I will believe you are who you say you are if someone I trust is vouching for your identity."
  - "I have a list of the people I trust."
- Identities (of the people with whom I talk, and the people who I trust) are represented by digital certificates.
- For a given application (RRSF), I keep my certificate, and those of the people I trust, in a container called a *key ring*.
- The TLS standard requires the server to send its certificate to the client for validation.
  - Also, optionally requires the client to send its certificate to the server for validation (also known as *client authentication*).

Setup: Digital certificates (2 of 3)

- **Question:** In an RRSF network, who is the client and who is the server?

- **Answer:** RRSF runs within the RACF subsystem address space. Each node can initiate a connection or accept it. Therefore, the RACF subsystem address space identity can act as either the client or the server.
  - That is, this is not the traditional client/server model; rather, it is a mesh of peers.

- Therefore, we must enforce client authentication so that both sides of the conversation are authenticated to each other.

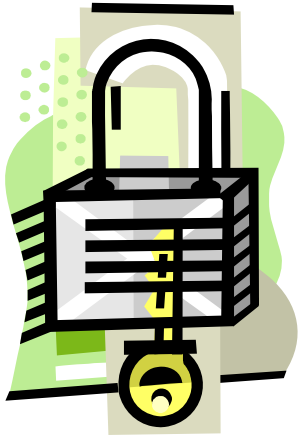
Setup: Digital certificates (3 of 3)

- **Question:** Because I trust the person vouching for your identity, does that make you an RRSF instance?
- **Answer:** No! Not necessarily.

- **Question:** If I knew that the sole purpose in life of somebody I trust is to sign RRSF server certificates and I am presented with a server certificate signed by that person, is that server an RRSF instance?
- **Answer:** Yes.

- **Question:** Whom do I need to trust to make sure that this is the case?
- **Answer:** You! (or your security/network administrator)

Setup: Digital certificates (bottom line)

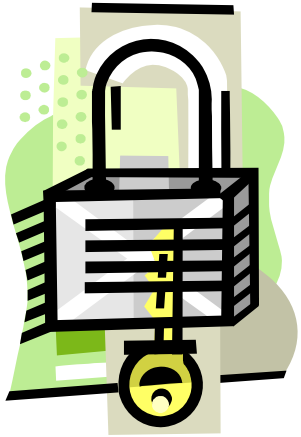


- On each system, the RACF address space must have access to a key ring containing:
  - A server certificate (with private key) for that RRSF instance.
  - The signing certificate (public key only) used to sign the RRSF server certificates and no others.
- In the simplest case, this can be accomplished with a single self-signed certificate added to each key ring (if your security policy allows it).
- Otherwise, create the signing certificate on one of the systems, and use it to create/sign that system's server certificate. On the other systems, generate a certificate request, and send it to the *signer system* where a certificate is generated. Send the certificate back to the original system and add it.
  - Never use the signing certificate to sign anything but RRSF certificates!
- See the *Security Administrator's Guide* for details.

Setup: Digital certificates (addendum)

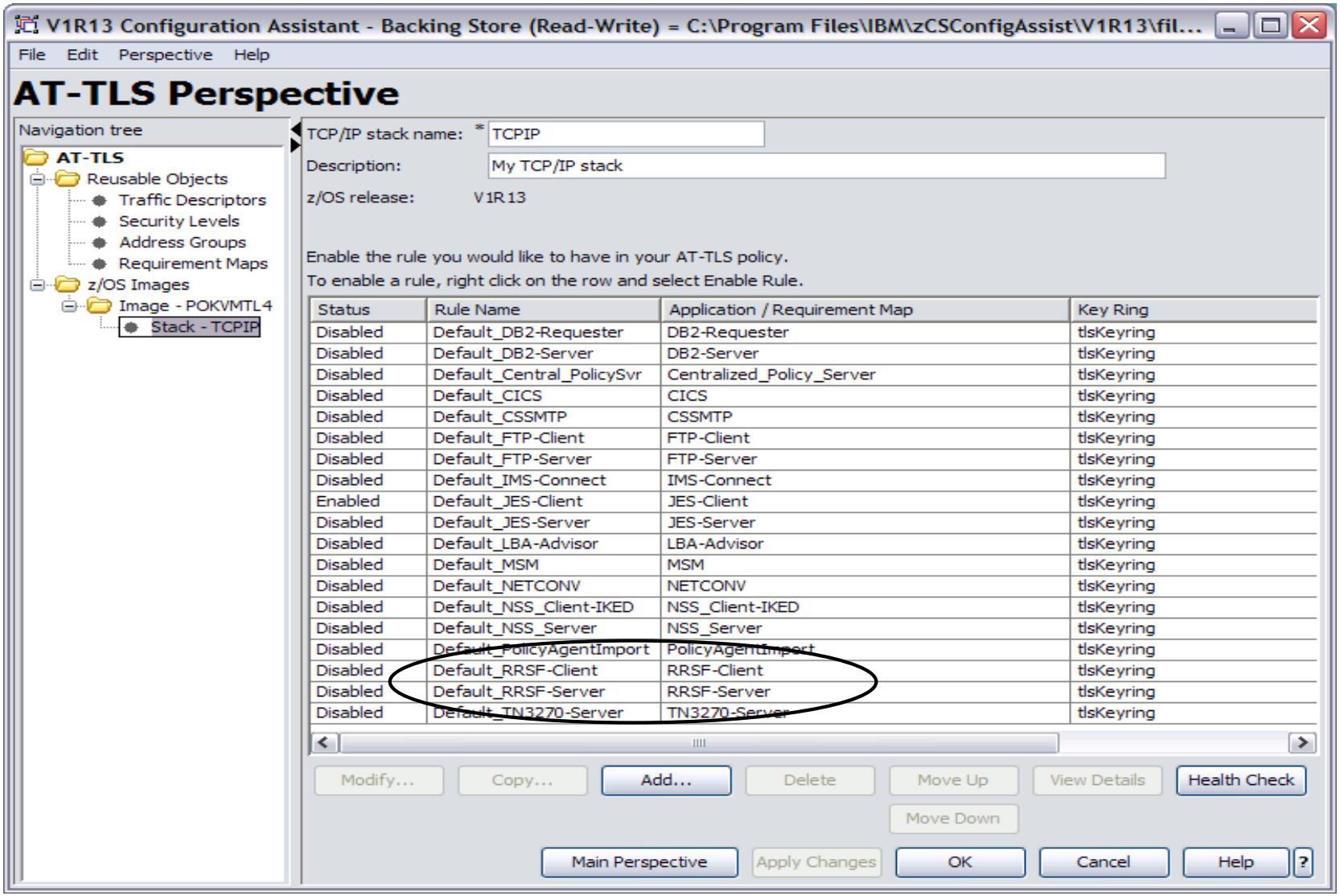
- **Question:** But what if my security policy forces me to obtain digital certificates from an external certificate authority?
- **Answer:** Patiently explain to the “Powers That Be” that you are configuring a RACF-to-RACF function between servers and that the RACDCERT command has already been paid for.
- If this does not work, you can still set up RRSF securely, but it is going to take some more work:
  - Change your AT-TLS policy to specify a client authentication level of SAFCheck (more on AT-TLS policy shortly).
  - “Map” every server certificate to a RACF user ID on every other system (there are multiple ways of accomplishing this).
  - Grant the mapped user ID READ access to `IRR.RRSF.CONNECT` in the RRSFDATA class on each system.

Setup: AT-TLS policy



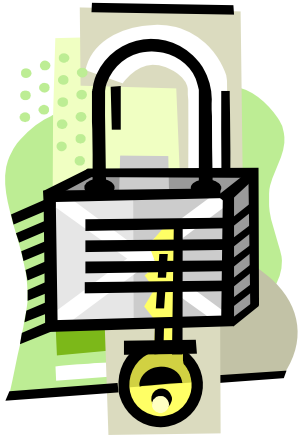
- **Question:** Now that you have your certificates in place, how will the system know to use them?
- **Answer:** Your AT-TLS policy contains the name of the key ring.
- TCP/IP, using System SSL, will use the policy to perform the TLS handshake when one RRSF attempts to connect to another.
- A working sample is provided. It just needs to be enabled and installed into the Policy Agent.
- RRSF is a *TLS-aware application*. RRSF will refuse to connect or accept connections unless adequate policy is in effect.

Setup: AT-TLS policy sample



Screen shot from the Communication Server Configuration Assistant GUI

Setup: AT-TLS policy (1 of 2)



- The sample will satisfy RRSF once (copied and) enabled.
  Notes of interest:
  - It consists of two rules: one when RRSF acts as the client and one when it acts as a server.
  - Client authentication is specified as *Required*.
  - The AES-256 cipher algorithm is specified (strongest currently supported algorithm).
  - A listening port number of 18136 is specified. This number has been reserved with the Internet Assigned Numbers Authority (IANA).
- The policy can optionally be made more specific by adding IP addresses, and user ID or job name, of the endpoints.
  - Would force a “no policy” error rather than a “handshake error” (which will be easier to debug, if it is a bug).

Fun fact:

- You can remember the listening port (not that you have to) by using the digits as an index into the alphabet:

18 = R
1 = A
3 = C
6 = F

- Purely a coincidence, I assure you.

Setup: AT-TLS policy (2 of 2)

- The sample will satisfy RRSF, but in case you modify it, RRSF will enforce the following properties:
  - Policy is in effect for the connection.
    - The TCP/IP stack is enabled for policy (`TCPCONFIG TTLS`).
    - A matching policy rule was found (match is based on target port number).
    - The rule is enabled.
  - Policy specifies a client authentication level of at least *Required*.
  - SSL V3, or TLS, is specified as the protocol level.
  - Application-controlled attribute is OFF.

- **Note:** A minimum encryption level is not enforced, though the sample specifies the strongest level currently available (AES-256), and the value is reported in the connection message and is displayed in `TARGET LIST` output.

Setup: Resource permissions for the RACF address space user ID

- Generally, you run the RACF started task with the TRUSTED attribute, and automatic access is granted to RACF-protected resources.
- However, this does not play well with the SERVAUTH class.
- Therefore, the RRSF tasks that perform TCP/IP communication run under a task-level security environment (ACEE) without the TRUSTED or PIVILEGED attributes.
- As a result, the subsystem user will need to be permitted to whatever SERVAUTH class profiles are protecting resources it is accessing.
  - However, do not permit it to the stack initialization resource (INITSTACK) or remote connections might fail during IPL.
- Permission will also be required to open the key ring.
  - Also, if the server's private key is stored in ICSF, the appropriate CSFKEYS/CSFSERV profiles.

The TARGET command (1 of 3)

New

New

```
subsystem-prefixTARGET
      [ DELETE | DORMANT | OPERATIVE ]
      [ DESCRIPTION('description') ]
      [ LIST ]
      [ LISTPROTOCOL ]
      [ LOCAL ]
      [ MAIN ]
      [ NODE(nodename |*) ]
      [ PREFIX(qualifier ...) ]
      [ PROTOCOL(
        [ APPC(
          [ LUNAME(luname) ]
          [ TPNAME(profile-name) ]
          [ MODENAME(mode-name) ]
          ) ]
        [ TCP(
          [ ADDRESS(host-name) ]
          [ PORTNUM(number) ]
          ) ]
        ) ]
      [ PURGE(INMSG | OUTMSG) ]
      [ SYSNAME(sysname |*) ]
      [ WDSQUAL(qualifier) ]
      [ WORKSPACE( {
        [ STORCLAS(class-name) ]
        [ DATACLAS(class-name) ]
        [ MGMTCLAS(class-name) ]
          | [ VOLUME(volume-serial) ] }
        [ FILESIZE([ nnnnnnnnn | 500 ]
          ) ]
        ) ]
```

TARGET command syntax: The LISTPROTOCOL and TCP keywords are new.

The TARGET command (2 of 3)

- Now that you have all that TLS stuff set up, the actual RRSF/RACF setup is almost as trivial as alluded to previously.
  - Establish the listener on the local node, which must first be dormant:

```
TARGET NODE(LOCNODE) DORMANT
TARGET NODE(LOCNODE) PROTOCOL(TCP) OPERATIVE
```

  - Define remote node:

```
TARGET NODE(REMOTE)
  PROTOCOL(TCP(ADDRESS(remote.pok.ibm.com)))
      PREFIX(SYS1.RRSF) WORKSPACE(VOLUME(VOL001))
  OPERATIVE
```

  - You will get a handshaking error message because you have not performed these steps on the remote system yet. When you have, the connection will complete.
- Now put these TARGET commands in the RACF parameter library so that they are automatically executed on the IPL or restart of the subsystem.

The TARGET command (3 of 3)

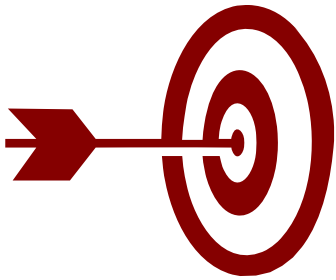
- **Notes:**
  - You can only specify one protocol per TARGET command. Therefore, if you have APPC nodes also (and you will if you wish to communicate with an R12 or older system), do the following in the RACF parameter library for the local node:
    - Remove the OPERATIVE keyword from the existing statement with APPC protocol information.
    - Add your TARGET command for TCP after the APPC command and specify OPERATIVE on this command.
  - For example, before:

```
TARGET NODE(NODE1) PROTOCOL(APPC(LUNAME(MF1AP001)))
PREFIX(NODE1.WORK) - WORKSPACE(VOLUME(TEMP01)
FILESIZE(500)) LOCAL OPERATIVE
```

  - For example, after:

```
TARGET NODE(NODE1) PROTOCOL(APPC(LUNAME(MF1AP001)))
  PREFIX(NODE1.WORK) - WORKSPACE(VOLUME(TEMP01)
  FILESIZE(500)) LOCAL
TARGET NODE(NODE1) PROTOCOL(TCP) OPERATIVE
```

TARGET LIST: Summary version

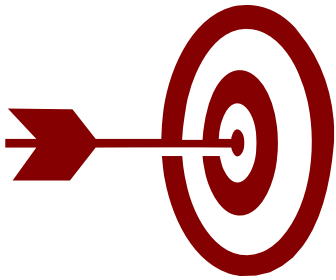


- A new message line, prefixed with IRRM091I, indicates the status of each protocol listener defined to the local node.

```
- NODE1  <target list
- NODE1  IRRM009I (<) LOCAL RRSF NODE NODE1 IS IN THE
  OPERATIVE ACTIVE  STATE.
- IRRM091I (<)       - LOCAL NODE APPC LISTENER IS
  ACTIVE.
- IRRM091I (<)       - LOCAL NODE TCP LISTENER IS
  ACTIVE.
- IRRM009I (<) REMOTE RRSF NODE NODE2 IS IN THE
  OPERATIVE ACTIVE STATE.
```

- Status values are ACTIVE, INACTIVE, and INITIALIZING.

TARGET LISTPROTOCOL

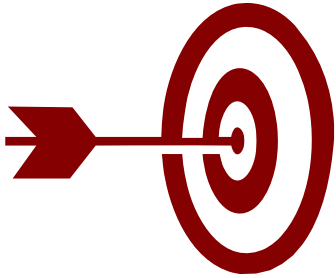


- LISTPROTOCOL is a new keyword that displays the protocol in IRRM009I for remote nodes.

```
- NODE1  <target list
- NODE1  IRRM009I (<) LOCAL RRSF NODE NODE1 IS IN THE
  OPERATIVE ACTIVE STATE.
- IRRM091I (<)       - LOCAL NODE APPC LISTENER IS
  ACTIVE.
- IRRM091I (<)       - LOCAL NODE TCP LISTENER IS
  ACTIVE.
- IRRM009I (<) REMOTE RRSF NODE NODE2 PROTOCOL TCP IS
  IN THE OPERATIVE ACTIVE STATE.
```

- This comes in handy when displaying a mixed-protocol network.

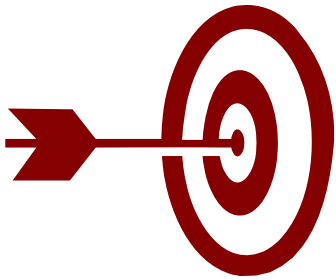
TARGET LIST: Detailed version (1 of 2)



- For the local node, shows protocol information for all defined protocols

```
NODE1  <target list node(node1)
NODE1  IRRM010I (<) RSWJ SUBSYSTEM PROPERTIES OF LOCAL RRSF NODE NODE1:
   STATE       - OPERATIVE ACTIVE
   DESCRIPTION - <NOT SPECIFIED>
   PROTOCOL    - APPC
                 LU NAME          - MF1AP001
                 TP PROFILE NAME  - IRRRACF
                 MODENAME         - <NOT SPECIFIED>
                 LISTENER STATUS  - ACTIVE
   PROTOCOL    - TCP
                 HOST ADDRESS     - 0.0.0.0
                 IP ADDRESS       - 9.57.1.243
                 LISTENER PORT    - 18136
                 LISTENER STATUS  - ACTIVE
   TIME OF LAST TRANSMISSION TO   - <NONE>
   TIME OF LAST TRANSMISSION FROM - <NONE>
   WORKSPACE FILE SPECIFICATION
         PREFIX                   - "NODE1.WORK"
         WDSQUAL                  - <NOT SPECIFIED>
         FILESIZE                 - 500
         VOLUME                   - TEMP01
         FILE USAGE
                "NODE1.WORK.NODE1.INMSG"
                                  - CONTAINS 0 RECORD(S)
                                  - OCCUPIES 1 EXTENT(S)
                "NODE1.WORK.NODE1.OUTMSG"
                                  - CONTAINS 0 RECORD(S)
                                  - OCCUPIES 1 EXTENT(S)
```

TARGET LIST: Detailed version (2 of 2)



- For a connected remote node, shows some AT-TLS information
  - Much more AT-TLS info is available with the NETSTAT command

```
NODE1  <target list node(node2)
NODE1  IRRM010I (<) RSWJ SUBSYSTEM PROPERTIES OF REMOTE RRSF NODE NODE2:
   STATE       - OPERATIVE ACTIVE
   DESCRIPTION - <NOT SPECIFIED>
   PROTOCOL    - TCP
                 HOST ADDRESS     - ALPS4012.POK.IBM.COM
                 IP ADDRESS       - 9.57.1.13
                 LISTENER PORT    - 18136
                 AT-TLS POLICY:
                   RULE_NAME      - DEFAULT_RRSF-CLIENT 1
                   CIPHER ALG     - 35 TLS_RSA_WITH_AES_256_CBC_SHA
                   CLIENT AUTH    - REQUIRED
   TIME OF LAST TRANSMISSION TO   - <NONE>
   TIME OF LAST TRANSMISSION FROM - <NONE>
   WORKSPACE FILE SPECIFICATION
         PREFIX                   - "RSFJ.BRUCE"
         WDSQUAL                  - <NOT SPECIFIED>
         FILESIZE                 - 500
         VOLUME                   - TEMP01
         FILE USAGE
                "RSFJ.BRUCE.NODE1.NODE2.INMSG"
                                  - CONTAINS 0 RECORD(S)
                                  - OCCUPIES 1 EXTENT(S)
                "RSFJ.BRUCE.NODE1.NODE2.OUTMSG"
                                  - CONTAINS 0 RECORD(S)
```

TCP workspace naming convention

- For local node, nothing has changed:

  prefix.sysname_or_wdsqual.INMSG|OUTMSG

- For remote nodes, the current convention uses LU names as qualifiers.

  prefix.local_luname.remote_luname_or_wdsqual.INMSG|OUTMSG

  – This continues to be the convention for APPC nodes

- For remote TCP nodes, the new convention is:

  prefix.local-node-qualifier.wdsqual-or-nodename-or-sysname.INMSG|OUTMSG

- This makes protocol conversions interesting.

Protocol conversions (1 of 5)

- Because of the different workspace file naming conventions, TARGET commands for one protocol cannot derive the names used by the other, and there is no persistent memory across restart/IPL.
- Therefore, when defining the new protocol for a given node, a new set of files will be allocated.
- The following problems must be avoided:
  - We cannot lose whatever work might be queued in the old files.
  - We cannot impose a “quiet time” on the customer to allow the old files to drain before queuing work to the new files.
  - We cannot let requests run out of order.
  - If there is a disruption (subsystem restart), we must continue where we left off when the subsystem resumes.
  - The conversion process should work in either direction.

Protocol conversions (2 of 5)

- Here is the approach:
  - Define protocol information for the local node.
  - For a remote node, enter a TARGET command as though you are defining the node from scratch, specifying the new protocol information (nothing will be copied from the existing protocol).
    - Communication will continue uninterrupted using the old protocol until the new protocol establishes a connection.
    - The new protocol will assume ownership of the old protocol's files.
    - The old protocol instance will be automatically deleted.
    - New requests will be queued to the new files while the old files are draining.
    - When the old files have drained, they will automatically be deallocated and deleted.
    - It will now appear as though the new protocol is the only one that ever existed.

Protocol conversions (3 of 5)

- **Example:** From NODE2, convert NODE1's protocol from APPC to TCP.
- **Assumptions:** TCP listeners have already been established on both systems, and NODE1 has already issued its remote node command.

```
>target node(node1) oper prefix(sys1.rrsf) workspace(volume(temp01))
 protocol(tcp(address(alps4242.pok.ibm.com)))

IRRC057I (>) RRSF PROTOCOL CONVERSION FROM APPC TO TCP FOR NODE
 NODE1 HAS BEEN INITIATED.
IRRM002I (>) RSWK SUBSYSTEM TARGET COMMAND HAS COMPLETED SUCCESSFULLY.
IRRI027I (>) RACF COMMUNICATION WITH TCP NODE NODE1 HAS BEEN
             SUCCESSFULLY ESTABLISHED USING CIPHER ALGORITHM 35
             TLS_RSA_WITH_AES_256_CBC_SHA.
IRRC058I (>) RRSF PROTOCOL CONVERSION FROM APPC TO TCP FOR NODE
 NODE1 IS COMPLETE.
```

- To make the conversion bullet-proof, harden the commands in the parameter library prior to issuing them on the console (see SPG for conversion procedure).

Protocol conversions (4 of 5)

- Before the conversion completes, TARGET LIST will show that the node owns two sets of workspace files.

```
<target list node(node2)
IRRM010I (<) RSWJ SUBSYSTEM PROPERTIES OF REMOTE RRSF NODE NODE2:
   STATE       - OPERATIVE ACTIVE
   ...
   ...
   WORKSPACE FILE SPECIFICATION
         PREFIX                   - "SYS1.RRSF"
         WDSQUAL                  - <NOT SPECIFIED>
         FILESIZE                 - 500
         VOLUME                   - TEMP01
         FILE USAGE
                "SYS1.RRSF.NODE1.NODE2.INMSG"
                                  - CONTAINS 0 RECORD(S)
                                  - OCCUPIES 1 EXTENT(S)
                "SYS1.RRSF.NODE1.NODE2.OUTMSG"
                                  - CONTAINS 0 RECORD(S)
                                  - OCCUPIES 1 EXTENT(S)
         CONVERSION FILE
                INMSG WORKSPACE FILE NOT ALLOCATED
                "SYS1.RRSF.MF1AP001.MF2AP001.OUTMSG"
                                  - CONTAINS 5 RECORD(S)
                                  - OCCUPIES 1 EXTENT(S)
```

Protocol conversions (5 of 5)

- After the conversion completes, there will be no trace left of the APPC node:

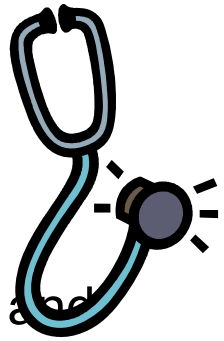
```
<target list node(node2) protocol(appc)
IRRM005I (<) RSWJ SUBSYSTEM TARGET COMMAND WAS UNABLE TO FIND
             DEFINITION OF NODE NODE2 PROTOCOL APPC.

IRRM003I (<) RSWJ SUBSYSTEM TARGET COMMAND ENDED IN ERROR.
```

New and changed messages

- Too numerous to list
- Listener status (successful initialization and termination, and error message when failing to start) are issued to the console.
- Connection status (successful or failed connection and successful or failed termination) are issued to the console.
- On failure, attempts are periodically retried (except for hand shaking errors), but duplicate error messages will not be issued.
  - If unsuccessful after about 30 minutes, a console message is issued and no more retries are attempted.
- Subtask start and end messages are issued to SYSLOG only.
- There will not be one-to-one correspondence with messages issued for APPC.

Diagnostics (1 of 5)

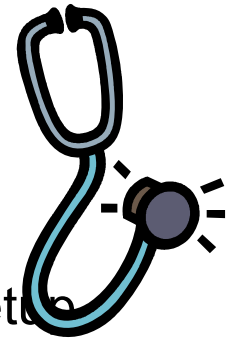


- UNIX System Services interfaces are used. Failing service, return code, and reason code are displayed in error messages.
  - Look in UNIX Messages and Codes for return code (errno) descriptions.
- For AT-TLS return codes, see Communication Server IP Diagnosis Reference for AT-TLS return codes.
- For AT-TLS failures:
  - RRSF issues an explicit message if there is something it does not like about the policy.
  - However, actual TLS handshake failures (happening under RRSF's radar) usually result in:
    - *Socket exception* message, for example:

```
IRRI031I (<) RRSF CONNECTION FROM PEER 9.57.1.13:1231 HAS
  BEEN
      REJECTED BECAUSE RACF COULD NOT VERIFY AT-TLS
  POLICY.
      THE BPX1SEL SERVICE DETECTED A SOCKET EXCEPTION.
```

    - AT-TLS trace messages on the console

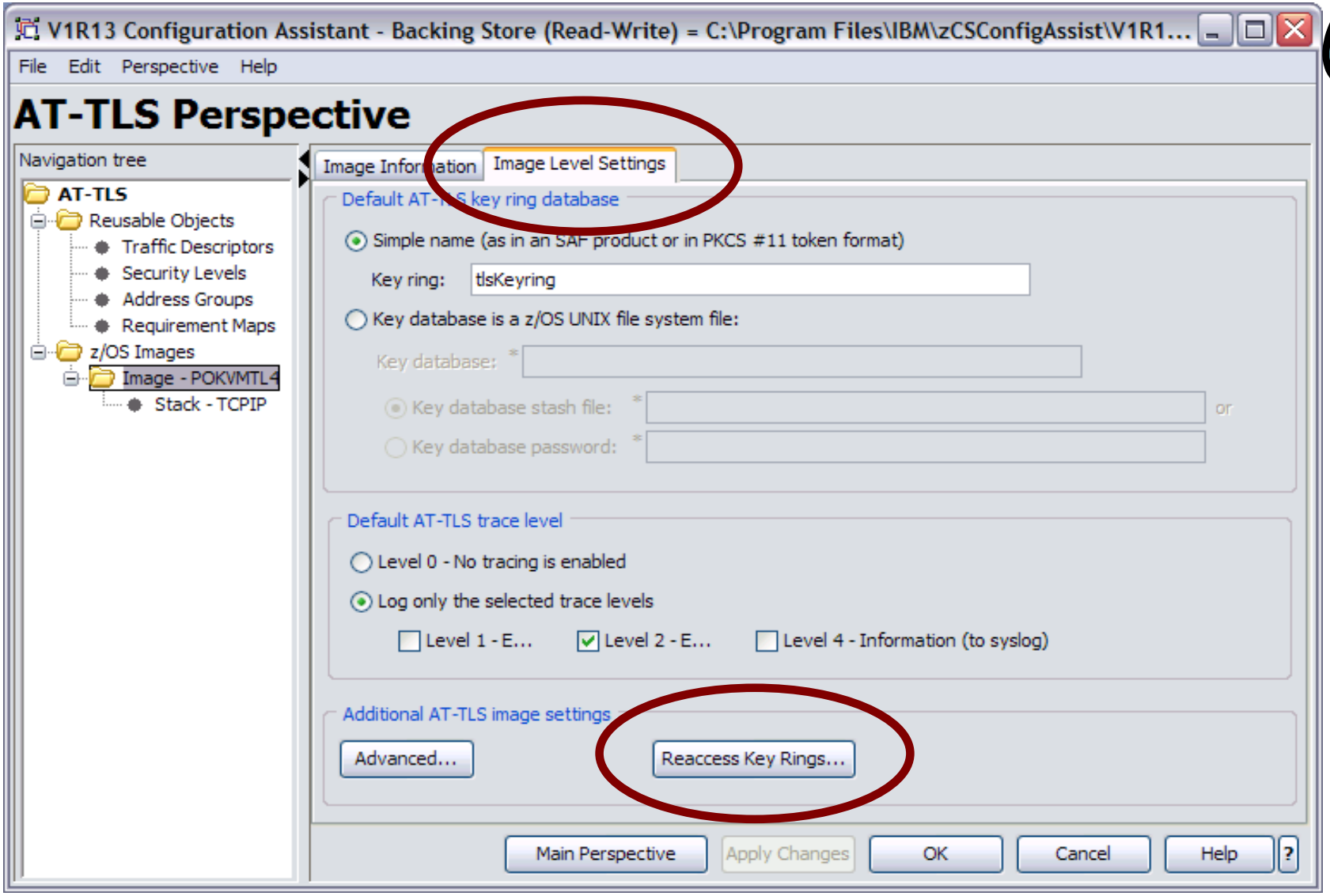
Diagnostics (2 of 5)



- TLS handshake errors are almost always caused by digital certificate setup errors. On both sides of the connection, check that:
  - The key ring specified in AT-TLS policy is defined for the subsystem user ID (case matters).
  - The node's server certificate (or self-signed) is connected as the default.
  - The signing CA cert (if not using self-signed server cert) is connected.
  - The RACF subsystem user has authority to read its key ring (usually implemented with READ access to `IRR.DIGTCERT.LISTRING` in the `FACILITY` class).

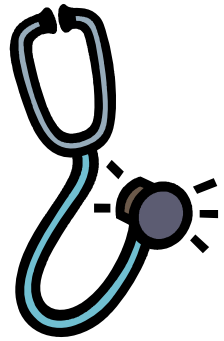
- After making any ring changes (including authority to read it), the policy agent must be notified.
  - Change `EnvironmentUserInstance` value in policy file (or use Reaccess Key Rings button in GUI; see next slide).
  - Refresh policy agent (“`f pagent,refresh`” command from console).

Diagnostics (3 of 5)



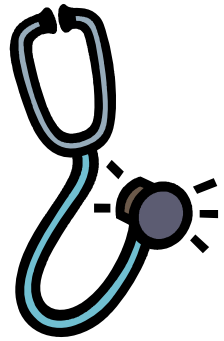
Configuration Assistant screen shot showing the Reaccess Key Rings function

Diagnostics (4 of 5)



- New SET TRACE(RRSF) operator command
- More comprehensive than existing SET TRACE(APPC) and no overlap with it
- Use at direction of RACF Level 2 when trying to debug a problem
- Do not run this way proactively all the time, because it will generate a lot of GTF records!

Diagnostics (5 of 5)



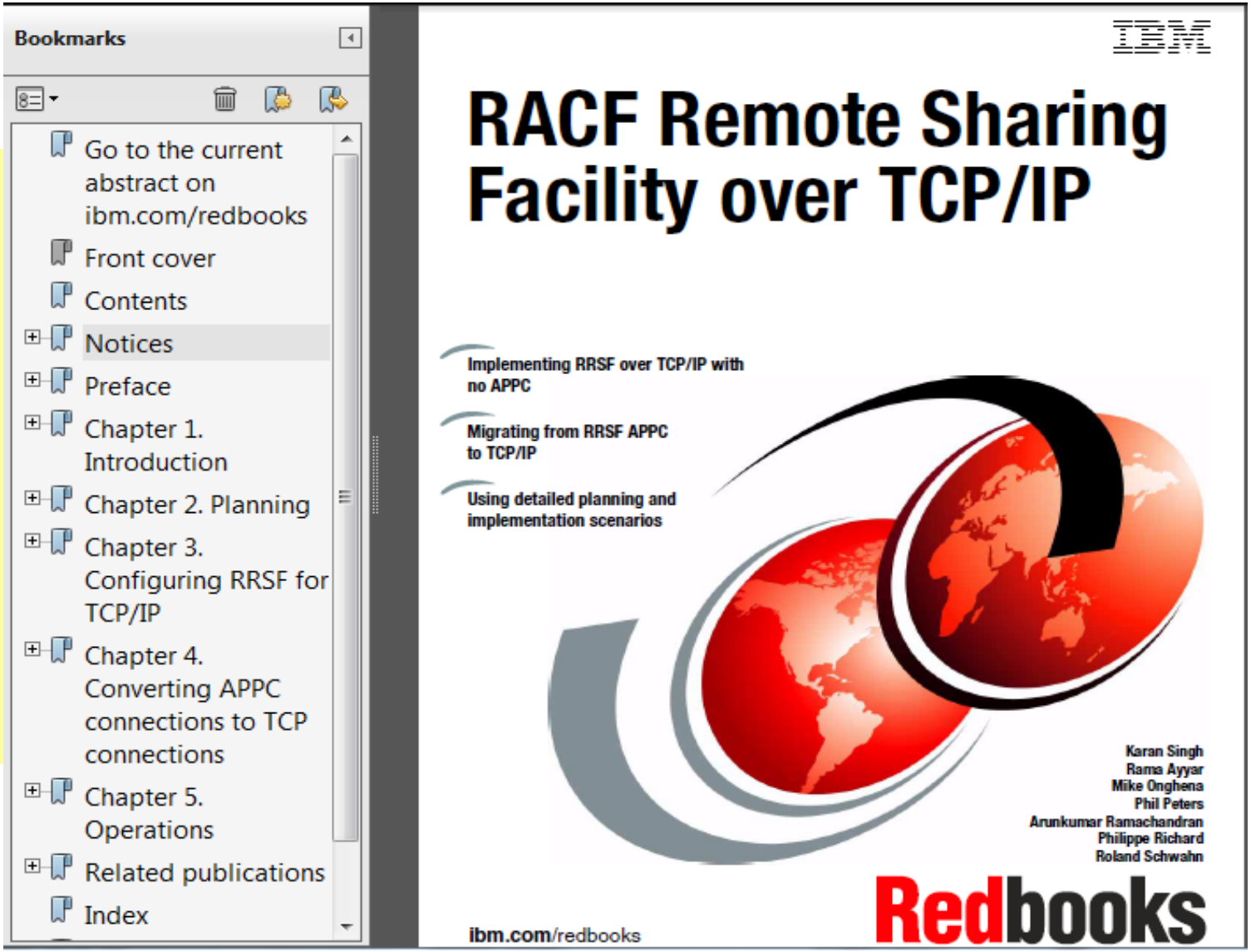
- Network connectivity errors will generally be debugged using Communication Server commands and traces.
- Note that `TARGET LIST` will display what RRSF thinks is the LAST resolved IP address for local and remote nodes.
- Addresses resolved when connection (inbound or outbound) is established (true for local node also).

```
<TARGET LIST NODE(NODE2)
IRRM010I (<) RSWJ SUBSYSTEM PROPERTIES OF REMOTE RRSF NODE NODE2:
   STATE        - DORMANT REMOTE
   DESCRIPTION  - <NOT SPECIFIED>
   PROTOCOL     - TCP
                  HOST ADDRESS      - ALPS4012.POK.IBM.COM
                  IP ADDRESS        - 9.57.1.13
                  LISTENER PORT     - 18136
   TIME OF LAST TRANSMISSION TO   - <NONE>
   TIME OF LAST TRANSMISSION FROM - <NONE>
   WORKSPACE FILE SPECIFICATION
...
```

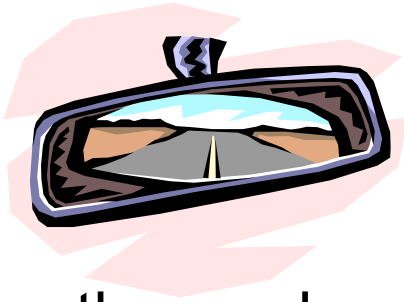
RACF Remote Sharing Facility over TCP/IP

SG24-8041-00

Karan Singh
Rama Ayyar
Mike Onghena
Phil Peters
Arunkumar Ramachandran
Philippe Richard
Roland Schwahn



Checkpoint (1 of 2)



1. True or False: Peer systems do not speak with each other and do not speak with non-MAIN systems of remote MSNs.

2. True or False: In a RRSF MSN, all members must be in the same SYSPLEX.

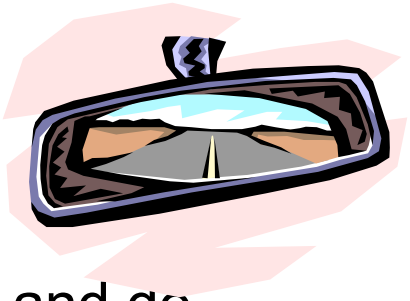
3. In a RRSF network, the RACF address space is the:

   a. Client

   b. Server

   c. Any

Checkpoint solutions (1 of 2)



1. True or False: Peer systems do not speak with each other and do not speak with non-MAIN systems of remote MSNs.

   The answer is true.

2. True or False: In a RRSF MSN, all members must be in the same SYSPLEX.

   The answer is false (RACF DB can be on a shared DASD).

3. In a RRSF network, the RACF address space is the:

   a. Client

   b. Server

   c. Any

   The answer is any.


THANK YOU