Using AT-TLS with HSC/SMC Client/Server z/OS Solution

Implementation Example

Part Number E27193-01

December 2011

Using AT-TLS with HSC/SMC Client/Server z/OS Solution Implementation Example

Part Number E27193-01

Oracle welcomes your comments and suggestions for improving this book. Contact us at STP_FEEDBACK_US@ORACLE.COM. Please include the title, part number, issue date, and revision.

Contents

Introduction

The purpose of this document is to present illustrative implementation concepts for Oracle's StorageTek HSC/SMC secure client/server communication using IBM's z/OS Application Transparent – Transport Layer Security (AT-TLS). The AT-TLS implementation for HSC/SMC communication is dependent on the environmental and business requirements of each individual customer. Depending on your requirements, your HSC/SMC AT-TLS implementation may differ from the example implementation shown in this document.

Oracle tested HSC client/server secure communication with z/OS AT-TLS in its Mainframe Customer Emulation Test Lab. HSC 6.2 was tested under z/OS 1.7, 1.8 and 1.9.

Items to note:

1. HSC 6.2 and SMC 6.2 were tested on z/OS 1.7, 1.8 and 1.9. No other HSC/SMC versions were tested with AT-TLS, and HSC/SMC 6.2 with AT-TLS was not tested on z/OS 1.10.
2. Only the SMC client was tested with AT-TLS. LibraryStation and MVS/CSC were not tested.
3. Only RACF was tested. No other z/OS security packages were tested, such as ACF2 and Top Secret.
4. ACSLS users: ACSLS platforms use encryption techniques different from AT-TLS. Exclude ACSLS IP addresses from the z/OS AT-TLS configuration file to avoid a conflict.

Chapter 1: Overview

AT-TLS is an encryption solution for TCP/IP applications that is completely transparent to the application client and server. Packet encryption and decryption occurs in the z/OS TCPIP address space at the TCP protocol level. The encrypted packet payload is unintelligible when sniffed or traced, but by the time it is delivered to the application the payload is once again readable.

Oracle tested AT-TLS with the StorageTek HSC/SMC 6.2 client/server solution without any changes to the SMC client application or the HSC server application (HSC/HTTP). All necessary modifications, additional parameter files and started tasks were made only to the z/OS TCP/IP facility and the z/OS operating system.

There is overhead associated with encrypting and decrypting the payload contents in the TCP protocol. This overhead was observed as a reduction in the number of HSC mount transactions performed during the test window. Encryption/decryption overhead will vary depending on the number of individual HSC/SMC client/server transactions and might not be observable in low-volume transaction environments.

Implementation

To implement AT-TLS encryption for HSC/SMC client/server communications, the minimum level needed for the Communication Server is z/OS 1.7.

IBM APAR's available should be applied for best performance:

Release 1A0 : UK39417 available 08/10/07 (1000) z/OS 1.10
Release 180 : UK39418 available 08/10/07 (1000) z/OS 1.8
Release 190 : UK39419 available 08/10/07 (1000) z/OS 1.9

See the following IBM publications for detailed information about the IBM z/OS Communications Server Policy Agent configuration and usage:

– IP Configuration Guide, SC31-8775

– IP Configuration Reference, SC31-8776

– IBM Redbook Communications Server TCP/IP Implementation, Volume 4, Policy-Based Network Security, SG24-7172

TCPIP:

The address space where TCPIP policies are specified, which is not necessarily where these policies are enabled. In our example implementation we indicated to TCPIP that TTLS would be used, but not until the PAGENT address space is active will TCPIP actually perform encryption/decryption work. Your implementation may differ depending on your business requirements.

Parmfile:

◊ Indicate where TCPIP address space will obtain certain policy based rules, See below: TCPIP parmfiles

◊ We set up an obeyfile to dynamically modify TCPIP and include TTLS: VARY TCPIP,,O,ZIP.TCPIP.PROFILES(ATTLS), see below: TCPIP Obey file.

Policy Agent (PAGENT):

The address space where the encryption rules are applied.

Parmfile:

◊ Where to find the configuration file and other parameters see: TCPIP parmfiles

```
****** ********************************* Top of Data ************************
000001 ; OSA GIG ETHERNET CARD
000002 DEVICE ECCQD01 MPCIPA   NONROUTER AUTORESTART
000003 LINK   &SYSNAME.MVS IPAQENET ECCQD01
000004
000005 ; OSA 1000BASE-T CARD
000006 DEVICE ECCQA01 MPCIPA   NONROUTER AUTORESTART
000007 LINK   &SYSNAME.2MVS IPAQENET ECCQA01
000008
000009 HOME
000010    10.80.&IPADDR1 &SYSNAME.MVS
000011    10.80.&IPADDR2 &SYSNAME.2MVS
000012
000013 BEGINROUTES
000014 ;       Destination             FirstHop     Linkname       PacketSize
000015   ROUTE 10.80.69.0/24           =            &SYSNAME.MVS   MTU 1492
000016   ROUTE DEFAULT                 10.80.69.254 &SYSNAME.MVS   MTU 1492
000017   ROUTE 10.80.68.0/24           =            &SYSNAME.2MVS  MTU 1492
000018   ROUTE DEFAULT                 10.80.68.254 &SYSNAME.2MVS  MTU 1492
000019 ENDROUTES
000020 INCLUDE USER.TCPIP.PROFILES(COMMON)
000021 START ECCQD01
000022 START ECCQA01
****** ******************************** Bottom of Data *******************
USER.TCPIP.PROFILES(COMMON)
****** ********************************* Top of Data *********************
000001 AUTOLOG
000002   FTPD        ; O/E FTP Server
000003   SMTP        ; Mail Server
000004   RXSERVE     ; Remote Execution Server
000005   PORTMAP     ; Portmap Server
000006 ENDAUTOLOG
000008 PORT
```

```
000009      7 UDP MISCSERV            ; Miscellaneous Server
000010      7 TCP MISCSERV
000011     20 TCP OMVS                ; FTP Server data port
000012     21 TCP OMVS                ; FTP Server control port
000013     23 TCP TN3270    NOAUTOLOG ; TN3270 Server
000014     25 TCP SMTP                ; SMTP Server
000015     53 TCP NAMESRV   NOAUTOLOG ;DOMAIN NAME SERVER
000016     53 UDP NAMESRV   NOAUTOLOG ; DOMAIN NAME SERVER
000017    111 TCP PORTMAP             ; Portmap Server
000018    111 UDP PORTMAP             ; Portmap Server
000019    135 UDP LLBD                ; HSC Location Broker
000020    161 UDP SNMPD               ; SNMP Agent
000021    162 UDP SNMPQE              ; SNMP Query Engine
000022    512 TCP RXSERVE             ; Remote Execution Server
000023    514 TCP RXSERVE             ; Remote Execution Server
000024    515 TCP LPSERVE             ; LPD Server
000025    520 UDP ROUTED              ; RouteD Server
000026    580 UDP NCPROUT   NOAUTOLOG ; NCPROUTE SERVER
000027    750 TCP MVSKERB   NOAUTOLOG ; KERBEROS
000028    750 UDP MVSKERB   NOAUTOLOG ; KERBEROS
000029    751 TCP ADM@SRV   NOAUTOLOG ; KERBEROS ADMIN SERVER
000030    751 UDP ADM@SRV   NOAUTOLOG ; KERBEROS ADMIN SERVER
000031   2049 UDP MVSNFS              ; NFS Server
000032   3000 TCP CICSTCP   NOAUTOLOG ; CICS SOCKET
000043   8000 TCP OMVS                ; Reserved for O/E Users
000044   8000 UDP OMVS                ; Reserved for O/E Users
****** ******************************** Bottom of Data **********************
```

- PAGENT parmfile.

Configuration File:

- Used to indicate to the PAGENT address space who/what/where the encryption is to take place see: AT-TLS client configuration, and AT-TLS server configuration.
- This is an Open Edition (OE) segment file.
- Download IBM Configuration Assistant tool from ‘Downloads’ section at: http://www.ibm.com/software/network/commserver/zos/support/
- Manual documenting use of the configuration assistant tool is at: IBM Configuration Assistant for z/os Communications Server

RACF:

In the z/OS environment, digital certificates are used by AT-TLS to authenticate and encrypt the protocol handshaking messages. An AT-TLS server must send its certificate to the client, and a server can optionally request a certificate from the client. See Chapter 3, “IP Security” in to the IBM Redbook: Communications Server for z/OS V1R8 TCP/IP Implementation Volume 4: Policy-Based Network Security SG24-7342, for information about how to set up digital certificate keys and key rings. See Chapter 3: RACF below used in our example implementation with RACF.

The z/OS Security Access Facility (SAF) is used to protect your network and communications. SAF is the high-level infrastructure that allows you to plug in any commercially available security product. References to RACF apply to any other SAF-compliant security products that provide the required support.

Digital Certificate:

◊ This where you define the certificate to RACF. See Ring creation and certificate creation commands

KeyRing:

◊ Specific name for the ring.

Started Tasks

TCPIP: AT-TLS encrypts the TCP/IP traffic between software clients and servers.

PAGENT: Policy Agent that determines which client, which server, what port, what IP address

Client: In our example the application client is HSC SMC

Server: In our example the application server is HSC HSC/HTTP server, started separately from HSC and SMC.

Media Management Strategy

Our example implementation assigns the Tape Management System to all media management functions. The condition of the media has nothing to do with the control path encryption that is done with AT-TLS.

Chapter 2: Samples

AT-TLS client configuration file

Note that the format of the configuration files generated by the configuration tool assistant is slightly different from what is presented here. We chose to simplify the configuration file for ease of change management. In our example, SMC (the client application) is started as SMC6C2.SMC6, the jobname parameter referred to in the client configuration file below is SMC6.

```
TTLSConfig tmp/t046028/attlc.conf                           <== name of file in OE segment
TTLSRule                          ZIPEMVS-To-ANYHTTP~1      <== Title of the Rule
{
  LocalAddr                       129.80.16.244             <== this hosts' IP addr
  RemoteAddr                      129.80.0.0/16             <== the many hosts that might have HTTP
  LocalPortRange                  1024-65535                <== SMC clients use dynamic ports outbound
  RemotePortRange                 0428                      <== HSC/HTTP uses 1 port, we selected 0428
  Jobname                         SMC6                      <== SMC client jobname was SMC6C2.SMC6
  Direction                       Outbound                  <== clients are outbound
  Priority                        255                       <== many rules can have priorities
  TTLSGroupActionRef              gAct1~SMC-To-HTTP         <== group-Action name, must match below
  TTLSEnvironmentActionRef        eAct1~SMC-To-HTTP         <== environment name, must match below
  TTLSConnectionActionRef         cAct1~SMC-To-HTTP         <== connection name, must match below
}
TTLSGroupAction                   gAct1~SMC-To-HTTP         <== group-Action name
{
  TTLSEnabled                     On                        <== tell PAGENT that TTLS is running
  Trace                           2                         <== 2 to 255, 2 is default
}
TTLSEnvironmentAction             eAct1~SMC-To-HTTP         <== environment name
{
  HandshakeRole                   Client                    <== a client does client handshakes
  EnvironmentUserInstance         0                         <== a single instance
  TTLSKeyringParmsRef             keyR1                     <== name for the Certificate Key Ring
}
TTLSConnectionAction              cAct1~SMC-To-HTTP         <== connection name
{
  HandshakeRole                   Client                    <== again, a client does client handshakes
  TTLSCipherParmsRef              cipher1~AT-TLS__Gold      <== name for below
  TTLSConnectionAdvancedParmsRef  cAdv1~SMC-To-HTTP         <== advanced connection name
  Trace                           2                         <== 2 to 255, like above
}
TTLSConnectionAdvancedParms       cAdv1~SMC-To-HTTP         <== advanced connection name from above
{
  CertificateLabel                CLIENT                    <== matches RACF for certificate name
}
TTLSKeyringParms                  keyR1                     <== Certificate Key Ring
```

```
{
  Keyring                         CLIRING                   <== matches what is RACF for key ring
}
TTLSCipherParms                   cipher1~AT-TLS__Gold      <== name from above
{
  V3CipherSuites                  TLS_RSA_WITH_3DES_EDE_CBC_SHA     <== one encryption algorithm
  V3CipherSuites                  TLS_RSA_WITH_AES_128_CBC_SHA      <== yet another
}
```

AT-TLS server configuration file

In our example, HSC/HTTP (the server application) is started as SVC3C2.SVC3, the jobname parameter referred to in the server configuration file below is SVC3.

```
TTLSConfig tmp/t046028/attls.conf                           <== name of file in OE segment
TTLSRule                          ZIPDMVS-To-ANYSMC~1       <== Title of the Rule
{
  LocalAddr                       129.80.16.123             <== this hosts' IP addr
  RemoteAddr                      129.80.0.0/16             <== the many hosts it can talk to
  LocalPortRange                  0428                      <== HSC/HTTP server uses a specific port
  RemotePortRange                 1024-65535                <== acceptable ports from client
  Jobname                         SVC3                      <== HSC/HTTP jobname was SVC3C2.SVC3
  Direction                       Inbound                   <== servers are inbound
  Priority                        255                       <== many rules can have priorities
  TTLSGroupActionRef              gAct1~HTTP-To-SMC         <== g-name, but must match below
  TTLSEnvironmentActionRef        eAct1~HTTP-To-SMC         <== e-name, but must match below
  TTLSConnectionActionRef         cAct1~HTTP-To-SMC         <== c-name, but must match below
}
TTLSGroupAction                   gAct1~HTTP-To-SMC         <== g-name from above
{
  TTLSEnabled                     On                        <== tell it TTLS is running
  Trace                           2                         <== 2 to 255, 2 is default
}
TTLSEnvironmentAction             eAct1~HTTP-To-SMC         <== e-name from above
{
  HandshakeRole                   Server                    <== a server does server handshakes
  EnvironmentUserInstance         0                         <== a single instance
  TTLSKeyringParmsRef             keyR1                     <== name for the Cerficate Key Ring
}
TTLSConnectionAction              cAct1~HTTP-To-SMC         <== c-name from above
{
  HandshakeRole                   Server                    <== again, a server does server handshakes
  TTLSCipherParmsRef              cipher1~AT-TLS__Gold      <== name for below
  TTLSConnectionAdvancedParmsRef  cAdv1~HTTP-To-SMC         <== name for below
  Trace                           2                         <== 2 to 255, like above
}
```

```
TTLSConnectionAdvancedParms        cAdv1~HTTP-To-SMC          <== name from above
{
  CertificateLabel                 SERVER                     <== matches RACF for certificate name
}
TTLSKeyringParms                   keyR1                      <== name from above
{
  Keyring                          SVRRING                    <== matches what is in RACF for key ring
}
TTLSCipherParms                    cipher1~AT-TLS__Gold       <== name from above
{
  V3CipherSuites                   TLS_RSA_WITH_3DES_EDE_CBC_SHA     <== one encryption algorithm
  V3CipherSuites                   TLS_RSA_WITH_AES_128_CBC_SHA      <== yet another
}
```

TCPIP Obey file

```
****** ********************************* Top of Data ************************
000001 TCPCONFIG TTLS
****** ******************************** Bottom of Data **********************
```

TCPIP parmfiles

```
****** ********************************* Top of Data ************************
000001 ; OSA GIG ETHERNET CARD
000002 DEVICE ECCQD01 MPCIPA   NONROUTER AUTORESTART
000003 LINK   &SYSNAME.MVS IPAQENET ECCQD01
000004
000005 ; OSA 1000BASE-T CARD
000006 DEVICE ECCQA01 MPCIPA   NONROUTER AUTORESTART
000007 LINK   &SYSNAME.2MVS IPAQENET ECCQA01
000008
000009 HOME
000010    10.80.&IPADDR1 &SYSNAME.MVS
000011    10.80.&IPADDR2 &SYSNAME.2MVS
000012
000013 BEGINROUTES
000014 ;       Destination             FirstHop     Linkname       PacketSize
000015   ROUTE 10.80.69.0/24           =            &SYSNAME.MVS   MTU 1492
000016   ROUTE DEFAULT                 10.80.69.254 &SYSNAME.MVS   MTU 1492
000017   ROUTE 10.80.68.0/24           =            &SYSNAME.2MVS  MTU 1492
000018   ROUTE DEFAULT                 10.80.68.254 &SYSNAME.2MVS  MTU 1492
000019 ENDROUTES
000020 INCLUDE USER.TCPIP.PROFILES(COMMON)
000021 START ECCQD01
000022 START ECCQA01
****** ******************************** Bottom of Data *******************
```

```
USER.TCPIP.PROFILES(COMMON)
****** ********************************* Top of Data *********************
000001 AUTOLOG
000002   FTPD         ; O/E FTP Server
000003   SMTP         ; Mail Server
000004   RXSERVE      ; Remote Execution Server
000005   PORTMAP      ; Portmap Server
000006 ENDAUTOLOG
000008 PORT
000009      7 UDP MISCSERV            ; Miscellaneous Server
000010      7 TCP MISCSERV
000011     20 TCP OMVS                ; FTP Server data port
000012     21 TCP OMVS                ; FTP Server control port
000013     23 TCP TN3270    NOAUTOLOG ; TN3270 Server
000014     25 TCP SMTP                ; SMTP Server
000015     53 TCP NAMESRV   NOAUTOLOG ;DOMAIN NAME SERVER
000016     53 UDP NAMESRV   NOAUTOLOG ; DOMAIN NAME SERVER
000017    111 TCP PORTMAP             ; Portmap Server
000018    111 UDP PORTMAP             ; Portmap Server
000019    135 UDP LLBD                ; HSC Location Broker
000020    161 UDP SNMPD               ; SNMP Agent
000021    162 UDP SNMPQE              ; SNMP Query Engine
000022    512 TCP RXSERVE             ; Remote Execution Server
000023    514 TCP RXSERVE             ; Remote Execution Server
000024    515 TCP LPSERVE             ; LPD Server
000025    520 UDP ROUTED              ; RouteD Server
000026    580 UDP NCPROUT   NOAUTOLOG ; NCPROUTE SERVER
000027    750 TCP MVSKERB   NOAUTOLOG ; KERBEROS
000028    750 UDP MVSKERB   NOAUTOLOG ; KERBEROS
000029    751 TCP ADM@SRV   NOAUTOLOG ; KERBEROS ADMIN SERVER
000030    751 UDP ADM@SRV   NOAUTOLOG ; KERBEROS ADMIN SERVER
000031   2049 UDP MVSNFS              ; NFS Server
000032   3000 TCP CICSTCP   NOAUTOLOG ; CICS SOCKET
000043   8000 TCP OMVS                ; Reserved for O/E Users
000044   8000 UDP OMVS                ; Reserved for O/E Users
****** ******************************** Bottom of Data **********************
```

PAGENT parmfile

```
LIBPATH=/lib:/usr/lib:/usr/lpp/ldapclient/lib:
PAGENT_CONFIG_FILE=/SYSTEM/tmp/t046028/attlc.conf <=== see above for more info on the conf file
PAGENT_LOG_FILE=/SYSTEM/tmp/t046028/pagentc.log   <=== file name where PAGENT logs information
PAGENT_LOG_FILE_CONTROL=3000,2
_BPXK_SETIBMOPT_TRANSPORT=TCPIP
TZ=MST7MDT
```

Debugging and PAGENT Logs

It is helpful to have different PAGENT logs during diagnosis; we used pagentc.log for client information and pagents.log for server information. The PAGENT started task reads the parmfile; the parmfile indicates what the library path will be, the config file name and the log file name.

The TCPIP address space also generates a debug.log that resides in /tmp. Information is available there about how TCPIP is processing.

Sample JCL

```
000100 //PAGENT  PROC M='ZIPDCLI'
000200 //* PAGENT PROCEDURE found in USERS.PROCLIB(PAGENT)
000201 //*
000202 //* IBM COMMUNICATIONS SERVER FOR Z/OS
000203 //* SMP/E DISTRIBUTION NAME: EZAPAGSP
000204 //*
000205 //* 5694-A01 (C) COPYRIGHT IBM CORP. 1998, 2006
000206 //* LICENSED MATERIALS - PROPERTY OF IBM
000207 //* "RESTRICTED MATERIALS OF IBM"
000208 //* STATUS = CSV1R8
000209 //*
000210 //PAGENT   EXEC PGM=PAGENT,REGION=0K,TIME=NOLIMIT,
000220 //       PARM='POSIX(ON) ALL31(ON) ENVAR("_CEE_ENVFILE=DD:STDENV")/-D1'
000260 //STDENV   DD DSN=USERS.TCPIP.PROFILES(&M),DISP=SHR
000280 //* SAMPLE HFS FILE CONTAINING ENVIRONMENT VARIABLES:
000290 //SYSPRINT DD SYSOUT=*
000300 //SYSOUT   DD SYSOUT=*
000400 //*
000500 //CEEDUMP  DD SYSOUT=*,DCB=(RECMF=FB,LRECL=132,BLKSIZE=132)
```

The parmfile ‘ZIPDCLI’ is found above: AT-TLS server configuration.

Recommendations

Configuration Overview

- **Understand IP addresses in environment.** Not all IP addresses have to be listed individually in the PAGENT conf files. IP subnet wildcarding can be used to simplify IP address specification.
- **Administer all AT-TLS from 1 permanently mounted OE segment.** This provides for a single point of management that is capable of spanning across multiple site locations, reduces complexity, and enables startup to be more readily shared within the organization.

Benefits of the Recommended Configuration

- **Ease of change management.** One shared OE segment will allow all PAGENT configuration files to be in one location. This is especially useful when debugging a particular host and implementing multiple client traffic to one server. All clients could use the same configuration file if all were to encrypt traffic all the time.

Chapter 3: RACF

Overview

General notes:

The following RACF classes need to be activated:

- DIGTCERT =
- DIGTNMAP = there are no profiles defined
- DIGTRING =
- SERVAUTH CLASS must be RACLISTed in order to prevent PORTMAP and RXSERVE from amending once TCPIP is cycled to include the TCPCONFIG TTLS statement or the TCPIP OBEY statement is issued.

- RACDCERT GENCERT commands generate certificates.
- The WITHLABEL parameter is used in the RACDCERT CONNECT statements to identify which ring the task is associated.
- PAGENT started task is defined and uses the TCPIP userid which has an OMVS UID OF 0.
- STCTASK is the default userid associated with all started tasks which are not defined in the STARTED class.

Activate class commands

Used the RACF panel to activate classes:

DIGTCERT, DIGTNMAP, AND DIGTRING

Used batch commands for the following:

```
SETROPTS RACLIST(SERVAUTH)
RDEFINE SERVAUTH ** UACC(ALTER) OWNER(RACFADM)
RDEFINE STARTED PAGENT*.* OWNER(RACFADM) -
        STDATA(USER(TCPIP) GROUP(STCGROUP))
RDEFINE FACILITY IRR.DIGTCERT.LISTRING  UACC(NONE) OWNER(RACFADM)
RDEFINE FACILITY IRR.DIGTCERT.LIST      UACC(NONE) OWNER(RACFADM)
RDEFINE FACILITY IRR.DIGTCERT.GENCERT   UACC(NONE) OWNER(RACFADM)
```

Ring creation and certificate creation commands

```
RACDCERT ID(STCUSER) ADDRING(SVRRING)
RACDCERT ID(STCUSER) ADDRING(CLIRING)
RACDCERT ID(STCUSER) LISTRING(SVRRING)
RACDCERT ID(STCUSER) LISTRING(CLIRING)
RACDCERT ID(STCTASK) GENCERT                 -
      SUBJECTSDN(CN('SERVER.STORTEK.COM') -
      O('STORAGETEK')                    -
      OU('SERVER STORAGETEK')            -
      C('US'))                     -
      WITHLABEL('SERVER')                -
      TRUST                        -
      SIZE(1024)
 RACDCERT ID(STCTASK) GENCERT                -
      SUBJECTSDN(CN('CLIENT.STORTEK.COM') -
      O('STORAGETEK')                    -
      OU('CLIENT STORAGETEK')            -
      C('US'))                     -
      WITHLABEL('CLIENT')
 RACDCERT ID(STCTASK) ADDRING(CLIRING)
 RACDCERT ID(STCTASK) ADDRING(SVRRING)
 RACDCERT ID(STCTASK) CONNECT(ID(STCTASK) -
      LABEL('CLIENT') RING(CLIRING) -
      DEFAULT USAGE(PERSONAL))
 RACDCERT ID(STCTASK) CONNECT(ID(STCTASK) -
      LABEL('SERVER') RING(SVRRING) -
      DEFAULT USAGE(PERSONAL))
 RACDCERT ID(STCTASK) CONNECT(ID(STCTASK) -
      LABEL('CLIENT') RING(SVRRING) -
      USAGE(PERSONAL))
 RACDCERT ID(STCTASK) CONNECT(ID(STCTASK) -
      LABEL('SERVER') RING(CLIRING) -
      USAGE(PERSONAL))
ADDUSER PAGENT DFLTGRP(STCGROUP) OWNER(RACFADM) -
     OMVS(UID(0) HOME('/'))


SETROPTS RACLIST(STARTED) REFRESH
SETROPTS GENERIC(STARTED) REFRESH
```

Commands to list RACF definitions

```
RLIST STARTED PAGENT.* STDATA ALL
RLIST DIGTRING * ALL
RLIST FACILITY IRR.DIGTCERT.LISTRING ALL
RLIST FACILITY IRR.DIGTCERT.LIST  ALL
RLIST FACILITY IRR.DIGTCERT.GENCERT ALL
RACDCERT ID(STCTASK) LIST
RACDCERT ID(STCTASK) LISTRING(SVRRING)
RACDCERT ID(STCTASK) LISTRING(CLIRING)
```