stonebranch

WORKLOAD AUTOMATION SIMPLIFIED.

Universal Agent for z/OS:

Using IBM System SSL

Contents

Implementation

This document describes a minimum implementation that simply allows utilizing IBM System SSL with no server or user authentication. All certificates and private keys are created in RACF. Certificates and private keys intended for deployment on other systems are exported from RACF and distributed to the target systems.

The implementation described in this document has been simplified in order to effectively communicate how to implement z/OS RACF certificates. The example does not illustrate all the issues that should be considered when implementing a corporate certificate infrastructure.

In this example, certificates shall be implemented on two host systems. The user certificate is shared between three z/OS users, USERZ, USER1, and USER2. All the certificates are created on z/OS and moved to remote systems as needed. The example assumes the following:

- One z/OS system with a host name of SYSZ.
- One UNIX system with host name SYSU.
- z/OS users are user ID's USERZ, USER1, and USER2.
- z/OS Universal Broker user ID is UBRUSR.
- The company name is Acme, Inc. in Annapolis, Maryland USA.

The implementation of z/OS RACF managed certificates for use in the Universal Products components consist of the following high-level steps:

1. Plan and document the certificate infrastructure.
2. Create a trusted certificate authority certificate.
3. Create shared keys and certificate for the z/OS Universal Broker.
4. Create shared keys and a certificate for a z/OS Universal Command users.
5. Transport the shared Universal Broker certificate and private key and the CA certificate to the remote host.
6. Configure Universal Agent components to use the certificates and keys created in prior steps.

Step 1: Planning

The first item to consider is who will be the trusted Certificate Authority (CA). The CA group will be responsible for creating and potentially revoking certificates for the Universal Agent components and users. Above all, they must keep the CA private key secure. If the CA private key becomes compromised, all the certificates the CA has issued are compromised.

The following questions should be considered:

- What naming conventions should be used for the certificate distinguished names?
- What are the certificate expiration practices?
- Are certificates to be used on all servers and by all users?
- Who is responsible for distributing certificates and private keys to remote systems?

Step 2: Create Certificate Authority (CA) Certificate and Keys

The RACDCERT command is used to create the CA certificate and keys. Only the CA group should have authority to create CA certificates. The following command creates the private and public keys and the CA certificate.

```
RACDCERT CERTAUTH GENCERT +
      SUBJECTSDN(CN('Certificate Authority') +
            O('Acme, Inc.') +
            C('US')) +
      NOTAFTER(DATE(2030-01-01)) +
      KEYUSAGE(HANDSHAKE CERTSIGN) +
      WITHLABEL('ACME-CA')
```

CERTAUTH	Specifies that the command is addressing certificate authorities.
GENCERT	Specifies that certificate and public and private key is to be generated.
SUBJECTSDN	Specifies the name of the certificate authority. The subject field identifies who the certificate is for.
NOTAFTER	Specifies the expiration date of the certificate. CA certificates typically have a very long lifetime. In this case, the certificate expiries in 2030.
KEYUSAGE	Specifies for what purposes the certificate can be used. HANDSHAKE and CERTSIGN are required for CA certificates.
WITHLABEL	Specifies an easily remembered name for the certificate. The label name is case sensitive.

Step 3: Create Shared Broker Certificate and Keys

The RACDCERT command is used to create the Broker certificate and key. This certificate and key shall be shared between the z/OS Broker and non-z/OS Brokers. The certificate is signed with the CA certificate created in Step 2.

```
RACDCERT ID(UBRUSR) GENCERT +
      SUBJECTSDN(CN('ubroker.acme.com') +
            O('Acme, Inc.') +
            C('US')) +
      NOTAFTER(DATE(2020-01-01)) +
      KEYUSAGE(HANDSHAKE) +
      WITHLABEL('UBROKER') +
      SIGNWITH(CERTAUTH LABEL('ACME-CA'))
```

ID	Specifies that the command is performing an operation for user ID UBRUSR, which is the RACF user profile with which the z/OS Universal Broker started task executes.
GENCERT	Specifies that certificate and public and private key is to be generated.
SUBJECTSDN	Specifies the name of the z/OS broker. The subject field identifies the z/OS broker. The Common Name (CN) name should be the host name typically, but since this shall be shared between all broker's and will not be used for server authentication, that is not necessary.
NOTAFTER	Specifies the expiration date of the certificate. In this case, the certificate expiries in 2020.
KEYUSAGE	Specifies for what purposes the certificate can be used. HANDSHAKE is required for the certificate to be used by the SSL protocol.
WITHLABEL	Specifies an easily remembered name for the certificate. The label name is case sensitive.
SIGNWITH	Specifies the CA certificate and private key used to sign the certificate. The CA certificate is identified in the certificate being created as the certificate issuer. In this case, the CA certificate is referenced with the label name of the CA certificate created in Step 2.

The NOTAFTER parameter is not specified in the above command. The default value is one year from the date the certificate is created.

Step 4: Create z/OS Broker Key Ring

The RACDCERT command is used to create key rings. A RACF key ring is a RACF definition that groups a set of certificates together for easy reference. A key ring must be created for z/OS Broker user ID.

```
RACDCERT ID(UBRUSR) ADDRING(UBRRING)
```

ID	Specifies that the command is performing an operation for user ID UBRUSR, which is the RACF user profile with which the z/OS Universal Broker started task executes.
ADDRING	Specifies that a key ring is to be created and the name of the key ring. The key ring name is case sensitive.

Step 5: Connect CA Certificate to z/OS Broker Key Ring

The RACDCERT command is used to connect certificates to key rings. The CA certificate that issued the z/OS Broker certificate is connected to the key ring.

```
RACDCERT ID(UBRUSR) CONNECT(CERTAUTH LABEL('ACME-CA') RING(UBRRING))
```

ID	Specifies that the command is performing an operation for user ID UBRUSR, which is the RACF user profile with which the z/OS Universal Broker started task executes..
CONNECT	Specifies that a certificate is being connected to a key ring.
CERTAUTH	Specifies the type of certificate being connected as a CA certificate.
LABEL	Specifies the label name of the certificate being connected. In this case, the label refers to the CA certificate created above.
RING	Specifies the ring name to which the certificate is connected.

Step 6: Connect Shared Broker Certificate to z/OS Broker Key Ring

The RACDCERT command is used to connect certificates to key rings. The shared Broker certificate is connected to the key ring.

```
RACDCERT ID(UBRUSR) CONNECT(LABEL('UBROKER') +
      RING(UBRRING) DEFAULT)
```

ID	Specifies that the command is performing an operation for user ID UBRUSR, which is the RACF user profile with which the z/OS Universal Broker started task executes.
CONNECT	Specifies that a certificate is being connected to a key ring.
LABEL	Specifies the label name of the certificate being connected. In this case, the label refers to the z/OS Broker certificate created above.
RING	Specifies the ring name to which the certificate is connected.
DEFAULT	Specifies the certificate being connected is the default certificate for the key ring.

Step 7: Permit RACF Access to Key Rings

Key ring access is controlled by the resource profile IRR.DIGTCERT.LISTRING in the FACILITY class. In order for the z/OS Broker to use the key ring, it must be permitted appropriate access to the profile.

The following command will list information about the profile (if it is defined) and a list of user profiles with their access level:

```
RLIST FACILITY (IRR.DIGTCERT.LISTRING) ALL
```

If the IRR.DIGTCERT.LISTRING profile is not defined, the following command can be used to define the profile:

```
RDEFINE FACILITY (IRR.DIGTCERT.LISTRING) UACC(NONE)
```

The z/OS Broker user ID must be permitted access to the profile with the following command:

```
PE IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(UBRUSR) ACCESS(READ)
```

Step 8: Create Shared z/OS User Certificate and Keys

The RACDCERT command is used to create a shared user certificate and key. Since the certificate shall be shared between users, it must be a SITE certificate. The certificate is signed with the CA certificate created in Step 2.

```
RACDCERT SITE GENCERT +
      SUBJECTSDN(CN('Shared User') +
            O('Acme, Inc.') +
            C('US')) +
      NOTAFTER(DATE(2020-01-01)) +
      KEYUSAGE(HANDSHAKE) +
      WITHLABEL('UNIVERSAL') +
      SIGNWITH(CERTAUTH LABEL('ACME-CA'))
```

SITE	Specifies that the command is performing an operation for the z/OS SITE.
GENCERT	Specifies that certificate and public and private key is to be generated.
SUBJECTSDN	Specifies the name of the user. The subject field identifies the user by name.
NOTAFTER	The certificate expiration date is set to January 1, 2020.
KEYUSAGE	Specifies for what purposes the certificate can be used. HANDSHAKE is required for the certificate to be used by the SSL protocol.
WITHLABEL	Specifies an easily remembered name for the certificate. The label name is case sensitive.
SIGNWITH	Specifies the CA certificate and private key used to sign the certificate. The CA certificate is identified in the certificate being created as the certificate issuer. In this case, the CA certificate is referenced with the label name of the CA certificate created in Step 2.

Step 9: Permit User Access to Shared Certificate and Key

In order for user IDs not associated with a certificate and key to use them, they must have CONTROL access to the GENCERT profile.

```
PERMIT IRR.DIGTCERT.GENCERT CLASS(FACILITY) +
      ID(USERZ,USER1,USER2) ACCESS(CONTROL)
```

The user ID's USERZ, USER1, and USER2 are permitted access to use the certificate and key.

Step 10: Create Shared z/OS User Key Ring

The RACDCERT command is used to create a shared key ring. A RACF key ring is a RACF definition that groups a set of certificates together for easy reference. A key ring must be created for the z/OS user ID.

```
RACDCERT ID(USERZ) ADDRING(UNVRING)
```

ID	Specifies that the command is performing an operation for user ID USERZ. A key ring must be associated with a user ID even though it will be shared by many.
ADDRING	Specifies that a key ring is to be created and the name of the key ring. The key ring name is case sensitive.

Step 11: Connect z/OS User Certificate to Shared User Key Ring

The RACDCERT command is used to connect certificates to key rings. The z/OS user certificate is connected to the key ring associated with USERZ.

```
RACDCERT ID(USERZ) CONNECT(LABEL('UNIVERSAL') RING(UNVRING)
DEFAULT)
```

ID	Specifies that the command is performing an operation for user ID USERZ.
CONNECT	Specifies that a certificate is being connected to a key ring.
LABEL	Specifies the label name of the certificate being connected. In this case, the label refers to the z/OS user certificate created above.
RING	Specifies the ring name to which the certificate is connected.
DEFAULT	Specifies the certificate being connected is the default certificate for the key ring.

Step 12 Connect CA Certificate to Shared User Key Ring

The RACDCERT command is used to connect certificates to key rings. The CA certificate that issued the User certificate is connected to the key ring.

```
RACDCERT ID(USERZ) CONNECT(CERTAUTH LABEL('ACME-CA')
RING(UNVRING))
```

ID	Specifies that the command is performing an operation for user ID USERZ.
CONNECT	Specifies that a certificate is being connected to a key ring.
CERTAUTH	Specifies the type of certificate being connected as a CA certificate.
LABEL	Specifies the label name of the certificate being connected. In this case, the label refers to the CA certificate created above.
RING	Specifies the ring name to which the certificate is connected.

Step 13 Permit User Access to Shared Key Ring

Key ring access is controlled by the resource profile IRR.DIGTCERT.LISTRING in the FACILITY class. In order for users to use the key ring, they must be permitted appropriate access to the profile.

The following command permits users to the profile:

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) +
      ID(USERZ,USER1,USER2) ACCESS(UPDATE)
```

Step 14 Export the CA Certificate

The CA certificate must be exported so that it can be used by the Broker on the remote system SYSU. The RACDCERT command is used to export the certificate.

```
RACDCERT CERTAUTH EXPORT(LABEL('ACME-CA')) +
      DSN(CA.CERT) FORMAT(CERTB64)
```

CERTAUTH	Specifies that the command is performing an operation for CA certificates.
EXPORT	Specifies that a certificate shall be exported.
LABEL	Specifies the label name of the CA certificate to export.
DSN	Specifies the data set name in which the CA certificate is exported. The data set is deleted and reallocated if it exists.
FORMAT	Specifies the format of the exported certificate. CERTB64 specifies a base64 encoded format. Base64 is compatible with the PEM format.

Step 15 Export the Shared Broker Certificate and Private Key

The Broker certificate and private key must be exported so that it can be used by all Brokers on remote systems. The RACDCERT command is used to export the certificate and private key.

```
RACDCERT ID(UBRUSR) EXPORT(LABEL('UBROKER')) +
      DSN(UBROKER.P12) FORMAT(PKCS12DER) PASSWORD('apwd')
```

ID	Specifies that the command is performing an operation for site certificates.
EXPORT	Specifies that a certificate shall be exported.
LABEL	Specifies the label name of the site certificate to export.
DSN	Specifies the data set name in which the certificate and private key is exported. The data set is deleted and reallocated if it exists.
FORMAT	Specifies the format of the exported certificate and private key. PKCS12DER specifies a DER encoded PKCS #12 package. PKCS #12 format is used when the private key corresponding to the certificate is to be exported. This is a binary format.
PASSWORD	Specifies the PKCS #12 password required to read the exported certificate and private key.

Step 16 Transfer Certificates to the UNIX System

The exported CA certificate and the UNIX Broker certificate and private key must be transferred to the UNIX system SYSU so they can be used by the Universal Broker on that system. The base64 exported files are text files and the DER exported files are binary files.

The following is an FTP transcript used to transfer the files to the UNIX system. From z/OS ISPF, FTP can be entered with the command TSO FTP. The following transcript begins after an FTP session has been successfully established with system SYSU and a user has successfully logged on.

```
ascii
put CA.CERT ca.cert
binary
put UBROKER.P12 sysu.p12
quit
```

Step 17 Import the UNIX Broker Certificate and Private Key

On the UNIX system SYSU, the UNIX Broker certificate and private key must be imported. Universal Certificate is used import the PKCS #12 file exported from the RACF database. The following command places the certificate in file `ubroker.cert` and the private key in `ubroker.pkey`.

```
ucert –create cert –cert_file ubroker.cert \
      –private_key_file ubroker.pkey -transport_file sysu.p12 \
      –transport_file_pwd apwd
```

`-cert_file`	Specifies the file name in which to export the certificate.
`-private_key_file`	Specifies the file name in which to export the private key.
`-transport_file`	Specifies the PKCS #12 file to be imported.
`-transport_file_pwd`	Specifies the password that is protecting the PKCS #12 file.

The private key file, ubroker.pkey, must be placed in a secure location that only the Universal Broker daemon has read access. This implementation example assumes that the Broker certificate and private key files are placed in directory `/etc/universal/keys`, and the CA certificate is placed in directory `/etc/universal/ca`.

The following commands create the designated directories and moves the imported files to the directories. The UNIX user's home directory `/home/useru` is assumed in the example and the ubrokerd daemon is running as user ubroker.

```
su
rootpwd

cd /etc/universal
mkdir keys ca
chmod 700 keys
chmod 755 ca

mv /home/useru/ca.cert ca
chown ubroker ca/ca.cert
chmod 444 ca/ca.cert

mv /home/useru/ubroker.cert keys
chown ubroker keys/ubroker.cert
chmod 444 keys/ubroker.cert

mv /home/useru/ubroker.pkey keys
chown ubroker keys/ubroker.pkey
chmod 400 keys/ubroker.pkey
```

Step 18 Configure UNIX Broker

The UNIX Broker configuration must be updated to make use of the certificates. The following options must be set in the `/etc/universal/ubroker.conf` file:

`ca_certificates`	`/etc/universal/ca/ca.cert`
`certificate`	`/etc/universal/keys/ubroker.cert`
`private_key`	`/etc/universal/keys/ubroker.pkey`

Step 19 Configure z/OS Broker

The z/OS Broker configuration must be updated to make use of the certificates. The following options must be set in the `UNVCONF(UBRCFG00)`file:

`ssl_implementation`	`System`
`saf_key_ring`	`UBRRING`
`Saf_key_ring_label`	`UBROKER`

Step 20 Configure z/OS Universal Data Mover Manager

The z/OS Universal Data Mover Manager configuration must be updated to make use of the certificates. The following options must be set in the `UNVCONF(UDMCFG00)`file:

`ssl_implementation`	`System`
`saf_key_ring`	**`USERZ`**`/UNVRING`
`saf_key_ring_label`	`UNIVERAL`

Note that the saf_key_ring value refers to the user ID associated with the key ring. All users except for USERZ must refer to the key ring using this syntax.

Troubleshooting

Viewing Certificates

RACF certificates are stored in the RACF database. They can be listed with the RACDCERT command to view them.

The following command will list all certificates assigned to user ID UBRUSR:

```
RACDCERT ID(UBRUSR) LIST
```

A label can be specified to list a specific certificate:

```
RACDCERT ID(UBRUSR) LIST (LABEL('BROKER'))
```

A certificate authority certificate can be listed with the following command:

```
RACDCERT CERTAUTH LIST (LABEL('ACME-CA'))
```

Universal Certificate can be used to list PEM formatted certificates:

```
ucert –print cert –cert_file my.cert
```

Viewing Key Rings

RACF key rings are associated with a user profile and stored in the RACF database. They can be listed with the RACDCERT command to view them.

The following command will list all key rings associated with user ID UBRUSR:

```
RACDCERT ID(UBRUSR) LISTRING(*)
```

Certificate Expiration

Certificates are defined with an expiration date. CA certificates are typically defined with a long lifetime. Server and user certificates are typically defined with a lifetime of about one year.

Once a certificate expires, certificate validation will fail when a client attempts to establish an SSL connection with it.

A certificates expiration is specified as its `notAfter` value.

Once a certificate expires, it must be recreated. This typically means that the public and private key must be recreated as well.

Unknown Certificate Authority

A key element to certificate validation is insuring that the CA that issued the certificate is a known and trusted CA. If the CA is unknown, the certificate that it issued cannot be trusted and will fail validation.

There are a few primary reasons for a CA related failure:

- The component configuration does not reference a CA certificate file.
- The certificate being validated was created with a CA certificate that has now been replaced.
- A test certificate created by a test CA is attempting to validate against a production server configured with a product CA certificate, or vice versa.

4550 North Point Parkway, Suite 200
Alpharetta, Georgia 30022
U.S.A.