Scenarios

IBM

Contents

Scenarios

Scenarios demonstrate how to apply technology to accomplish business goals and solve problems. They describe hypothetical business situations to bring the discussions to life.

These scenarios explore some of the first steps and some of the more advanced tasks that you can do by using IBM Security Key Lifecycle Manager. As a prerequisite for these scenarios, install the IBM Security Key Lifecycle Manager server and verify that its components are running.

Scenario: To provide a primary and replica server

To ensure continuous key and certificate availability to encrypting devices, configure a primary and a replica IBM Security Key Lifecycle Manager server for your enterprise. Then, provide repeated backup and restore actions that protect critical data.

On Windows systems and other systems, both systems must have the required memory, speed, and available disk space to meet the workload.

IBM Security Key Lifecycle Manager creates backup files in a manner that is independent of operating systems and directory structure of the application. You can restore the backup files to an operating system that is different from the one it was backed up from.

Figure 1. Primary and replica IBM Security Key Lifecycle Manager server

Before you create a replica server, catalog the requirements in your operation, which might include:

- Disaster recovery procedures that are unique to your site. The procedures might require ad hoc or periodic activities to ensure concurrent availability of a primary and replica IBM Security Key Lifecycle Manager server.

  Your site might require periodic exercises to demonstrate that a simulated failure of a primary IBM Security Key Lifecycle Manager server causes an immediate response from a replica.

  The IBM Security Key Lifecycle Manager server does not provide automatic failover. You must separately set up the necessary device controls to ensure that the replica server is available if the primary server fails.
- Initial installation and configuration of IBM Security Key Lifecycle Manager server and the devices in your installation that require keys and certificates.

You might choose to also install and configure IBM Security Key Lifecycle Manager server and its prerequisites on another server, and set a schedule to back up and restore critical data.

- Cycles of time at which your organization normally changes keys and certificates.

  If your organization replaces keys and certificates on a monthly or quarterly basis, ensure that the key materials and other data are backed up when new keys and certificates begin their usage cycle.
- Events that cause you to create a certificate request and send the request to a certificate authority.

  Use the secure communication process that your site or the certificate authority requires. Run a backup to protect keys and data that are associated with a certificate request until the actual certificate returns.
- Upgrades and related middleware fix packs for the IBM Security Key Lifecycle Manager server.

  Run a backup to ensure that the upgraded IBM Security Key Lifecycle Manager server has the same keys and other critical data that were in use immediately prior to the upgrade.

Backup and restore practices

When a change occurs, such as adding or changing devices, keys, and certificates, you must back up the IBM Security Key Lifecycle Manager critical data. IBM Security Key Lifecycle Manager provides a task that creates a backup file of configuration files, database, and other data. You can restore this backup file to an operating system that is different from the one it was backed up from.

`Failure to back up your critical data properly might result in unrecoverable loss of all access to your encrypted data. Do not encrypt your backup file, or store a backup file on an encrypting device.` Failure to back up data might also result in a later inconsistency of the key manager and potential data loss on the storage device.

You can follow these practices:

- Maintain both a primary IBM Security Key Lifecycle Manager server and at least one replica IBM Security Key Lifecycle Manager server that run concurrently. Ensure that a storage device has access to its keys if the primary server fails.

  The IBM Security Key Lifecycle Manager server does not provide automatic failover. You must separately set up the necessary device controls to ensure that the replica server is available if the primary server fails.
- Run the backup task whenever you add or change devices, keys, or certificates. Restore the IBM Security Key Lifecycle Manager backup file to a replica IBM Security Key Lifecycle Manager server.
- Do not make changes to the IBM Security Key Lifecycle Manager server on the replica computer under normal operating conditions in which a primary server is always available. If failure events cause significant activity on the replica server while the primary server is down, back up the replica server and restore the backup file to the primary server.
- Use only the IBM Security Key Lifecycle Manager backup and restore tasks to create a backup file. Use only IBM Security Key Lifecycle Manager to restore the data that the backup file contains. Do not take other manual steps to back up or to restore files.

- Keep backup files in a safe place, separate from the computer on which the IBM Security Key Lifecycle Manager server runs. Ensure that function can be rebuilt on a replacement server if files on the primary IBM Security Key Lifecycle Manager server are lost. These files might reside at a geographically separate location.

Backup and restore runtime requirements

Backing up and restoring data from backup files for IBM Security Key Lifecycle Manager have several runtime requirements.

Prevent timeout failure by increasing the time interval that is allowed for backup and restore transactions for large key populations. Specify a larger value for the **totalTranLifetimeTimeout** setting in this file:

WAS_HOME/profiles/KLMProfile/config/cells/SKLMCell/nodes/SKLMNode/servers/server1/server.xml

Additionally, these conditions must be true:

- Ensure that the task occurs during a time interval that allows a halt to key serving activity.
- For a backup task, the IBM Security Key Lifecycle Manager server must be running in a normal operational state. The IBM Security Key Lifecycle Manager database instance must be available.
- For a restore task, the IBM Security Key Lifecycle Manager database instance must be accessible through the IBM Security Key Lifecycle Manager data source.

  Before you start a restore task, ensure that you have the password that was used when the backup file was created. Restored files must be written to the same IBM Security Key Lifecycle Manager server from which the data was previously backed up. Alternatively, the restored files must be written to a replica computer.
- Ensure that the directories, which are associated with the **tklm.backup.dir** property exist. Also, ensure read and write access to these directories for the system and IBM Security Key Lifecycle Manager administrator accounts under which the IBM Security Key Lifecycle Manager server and the DB2® server run.

Setting up a replica computer

A replica computer for IBM Security Key Lifecycle Manager must have the same or greater storage capacity and free disk space as the primary computer on which IBM Security Key Lifecycle Manager server customarily runs.

About this task

Use the IBM Security Key Lifecycle Manager installation program and repeat the same steps that you took on the primary computer.

Procedure

1. Obtain a computer that has the same or greater storage capacity and free disk space as the computer on which IBM Security Key Lifecycle Manager server customarily runs.
2. Install and configure an operating system and fixes on the replica computer to match the system on the computer on which IBM Security Key Lifecycle Manager server customarily runs.
3. Complete the installation steps and verification steps that are described in the "Installing and configuring" section on IBM Knowledge Center for IBM Security Key Lifecycle Manager.

What to do next

Configure and test the replica computer after you install and verify the primary computer on which IBM Security Key Lifecycle Manager customarily runs.

Verify that a current backup file that you create on the primary IBM Security Key Lifecycle Manager server can be successfully restored on the replica computer.

Responding after significant replica server activity

A replica server might have significant activity while the primary IBM Security Key Lifecycle Manager server is down. Select an announced maintenance interval, when network traffic is stopped, to back up the replica server and restore the backup file to the primary server.

About this task

No alerts are issued if the replica server provides keys to a device. Validate that there is actually a need to back up the replica computer and then restore the backup file to the primary server. For example, you might determine whether a write request caused a key to be served to a device. Use the **tklmServedDataList** command to query the database and to list served data. Less significant information might be available in the audit log for read requests from devices.

Procedure

1. At an announced time when network traffic is stopped, back up the replica computer.
2. Restore the backup file from the replica computer onto the primary computer on which IBM Security Key Lifecycle Manager server customarily runs.

What to do next

Verify that the primary IBM Security Key Lifecycle Manager server is active and that the backup file was successfully restored.

Scenario: Request for a third-party certificate

IBM Security Key Lifecycle Manager can generate a certificate request in `PKCS #10` format that you can send to a certificate authority. Use the returned CA certificate to protect data on an encryption-enabled device, or for SSL communication.

1. Before you begin, determine whether the usage of the certificate is for SSL authentication, or for secure communication with 3592 tape drives or DS8000 Turbo drives.
2. For each of the certificates that you anticipate in your next business cycle, create a certificate request.

   The generated certificate request files reside in the `SKLM_HOME` directory. For example, a generated certificate request might be a file such as `SKLM_HOME\080419154137-sslcert001.csr`.

   The certificate request file is an encoded, `base64` format, which is not readable with an editor.

   The certificate request file contains the `base64` format information, including:

   - The version number.

- The subject name, which is the `X.500` name of the requestor. For example, an `X.500` name contains values for a common name (cn), organization, and other values that identify the subject.
- The public key data and the algorithm unique identifier. You can use the algorithm, such as `RSA` or `ECDSA`.
- A generated signature for the data that is signed by the private key of the user.

The keystore database contains the private key that was used to generate the signature for the certificate request.

Additionally, information related to the certificate request is stored in the database. The information includes the X.500 subject name, the start, expiration, and retirement date, and other values for other attributes that are normally specified for a certificate, including a pending state for the certificate request. The values are updated when the returned certificate is imported.

3. Protect certificate requests until the certificate returns. It is important to run a backup task for the keystore database after you create and send a certificate request, just as when you change actual keys or certificates in a keystore database.
4. After ensuring that a backup file is in place, manually send a certificate request to your selected certificate authority, by using the secure communication process that your site or the certificate authority requires for e-mail or https transmission.
5. Import a returned certificate that matches an earlier certificate request.

   Upon receipt of a valid request, the certificate authority returns a `DER` or `base64` encoded certificate to you. The certificate contains the public key that was provided in the certificate request, and a signature from the certificate authority, which specify that the public key is valid, and that your enterprise is the authentic owner. The certificate subject name is the `X.500` subject name that you provided in the certificate request.
6. Again back up the keystore database, which contains the new certificate.

Creating a certificate request

Use the Create Certificate dialog, **`tklmCertGenRequest`** command, or **`Certificate Generate Request REST Service`** to create certificate requests.

About this task

Before you begin, determine your site policy and process to obtain certificates that are issued by a certificate authority.

Procedure

1. Navigate to the appropriate page or directory:
   - Graphical user interface:
     a. Log on to the graphical user interface.
     b. In the Key and Device Management section on Welcome page, select the **3592** or **DS8000** device group.
     c. Click **Go to** > **Guided key and device creation**.
     d. Alternatively, right-click **3592** or **DS8000** and select **Guided key and device creation**.
   - Command-line interface:

In the *WAS_HOME*/bin directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive*:\Program Files (x86)\IBM\WebSphere\AppServer\bin directory and type:

– Windows systems:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```

– Systems such as AIX or Linux:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

- REST interface:
  – Open a REST client.

2. Request a certificate:
   - Graphical user interface:
     a. On the Step 1: Create Certificates page, click **Create**.
     b. On the Create Certificate dialog, select a certificate request for a third-party provider.
     c. Specify values for the required and optional parameters.
     d. Click **Create Certificate**.
   - Command-line interface:

     Type tklmCertGenRequest to create a certificate request file. For example:

     – SSL communication

```
print AdminTask.tklmCertGenRequest('[-alias sklmSSLCertificate1
 -cn sklm -ou sales -o myCompanyName -locality myLocation
  -country US -validity 999 -keyStoreName defaultKeyStore
   -fileName mySSLCertRequest1.crt -usage SSLSERVER]')
```

     – 3592 tape drives

```
print AdminTask.tklmCertGenRequest('[-alias sklmCertificate1
 -cn sklm -ou marketing -o CompanyName -locality myLocation
  -country US -validity 999 -keyStoreName defaultKeyStore
   -fileName myCertRequest1.crt -usage 3592]')
```

     – DS8000 Turbo drives

```
print AdminTask.tklmCertGenRequest('[-alias sklmCertificate3
 -cn sklm -ou sales -o myCompanyName -locality myLocation
  -country US -validity 999 -keyStoreName defaultKeyStore
   -fileName myCertRequest3.crt -usage DS8000]')
```

   - REST interface:
     a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.
     b. To invoke **Certificate Generate Request REST Service**, send the HTTP POST request. Pass the user authentication identifier that you obtained in Step a along with the request message as shown in the following example.

     – SSL communication

```
POST https://localhost:9080/SKLM/rest/v1/certificates
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"type":"certreq","alias":"sklmSSLCertificate1","cn":"sklm","ou":
"sales","o":
"myCompanyName","usage":"SSLSERVER","country":"US","validity":"999",
"fileName":
"mySSLCertRequest1.crt","algorithm":"ECDSA"}
```

     – 3592 tape drives

```
POST https://localhost:9080/SKLM/rest/v1/certificates
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"type":"certreq","alias":"sklmCertificate1","cn":"sklm","ou":
"sales","o":
"myCompanyName","usage":"3592","country":"US","validity":"999",
"fileName":
"myCertRequest1.crt","algorithm":"ECDSA"}
```

– DS8000 Turbo drives

```
POST https://localhost:9080/SKLM/rest/v1/certificates
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"type":"certreq","alias":"sklmCertificate3","cn":"sklm","ou":
"sales","o":
"myCompanyName","usage":"DS8000","country":"US","validity":"999",
"fileName":
"myCertRequest1.crt","algorithm":"ECDSA"}
```

3. A success indicator varies, depending on the interface:
   - Graphical user interface:

     The certificate or certificate request appears as an item in the **Certificates** table. Return to the Welcome page. On the Welcome page, in the **Action Items**, the certificate request appears as an item in the **Pending Certificate** table.
   - Command-line interface:

     A completion message indicates success.
   - REST interface:

     The status code `200 OK` indicates success.

What to do next

Manually send the certificate request to a certificate authority, by using the secure communication process that your organization provides. Additionally, retain the alias value of the certificate request, for use when you import the returned certificate, which must match a certificate request.

Importing a certificate

You can use the pending certificates link on the Welcome page of graphical user interface, the **`tklmCertImport`** CLI command, or **`Certificate Import REST Service`** to import a certificate that you earlier requested from a certificate authority.

About this task

Before you begin, ensure that the alias of the incoming certificate matches the alias of a previous certificate request, such as `sklm cert1`. Write the certificate file to a temporary directory.

Retrieve the alias of original certificate request, for use when you import the returned certificate, which must specify the correct alias.

To look up the `X.500` subject name of a certificate request, to determine whether it matches the `X.500` subject name of the certificate, run the **`tklmCertList`** command or **`Certificate List REST Service`**, by specifying the `state` attribute with a value of `pending`.

To look at the subject name of the certificate file, you might take these steps:

- Windows systems:

  Open the certificate file directly. A Windows native utility displays the information in the certificate in readable format.
- Other systems:

  Import the certificate into IBM Security Key Lifecycle Manager by using a new alias. Then, run the **tklmCertList** command or the **Certificate List REST Service**, specifying the alias, to view the certificate information.

Procedure

1. Navigate to the appropriate page or directory:
   - Graphical user interface:

     Log on to the graphical user interface. The Welcome page is displayed.
   - Command-line interface:

     In the *WAS_HOME*/bin directory, start a **wsadmin** session by using Jython. Log on to **wsadmin** with an authorized user ID, such as the SKLMAdmin user ID. For example, on Windows systems, navigate to the *drive*:\Program Files (x86)\IBM\WebSphere\AppServer\bin directory and type:
     - Windows systems:

       ```
       wsadmin -username SKLMAdmin -password mypwd -lang jython
       ```
     - Systems such as AIX or Linux:

       ```
       ./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
       ```
   - REST interface:
     - Open a REST client.
2. Import a certificate:
   - Graphical user interface
     a. In the Action Items section of the Welcome page, in the Key Groups and Certificates area, click **You have pending certificates**.
     b. In the **Pending Certificates** table, select the appropriate pending certificate
     c. Click **Import**.
     d. In the **File Name and location** field, type the path and file name of the certificate request file that are returned by the certificate authority.
     e. Alternatively, click **Browse** to navigate to the certificate request file. For example, you might browse to a pending certificate in the *drive*:\Program Files (x86)\IBM\WebSphere\AppServer\products\sklm directory.
     f. Click **Import**.
   - Command-line interface:

     Type tklmCertImport to import a certificate. For example:
     - SSL communication

       ```
       print AdminTask.tklmCertImport
        ('[-fileName myTempPath\\mySSLCertRequest1.cer
          -alias sklmSSLCertificate1 -format base64
            -keyStoreName defaultKeyStore -usage SSLSERVER]')
       ```
     - 3592 tape drives

       ```
       print AdminTask.tklmCertImport \
        ('[-fileName myTempPath\\myCertRequest2.cer
          -alias sklmCertificate2 -format base64
            -keyStoreName defaultKeyStore -usage 3592]')
       ```
     - DS8000 Turbo drives

```
print AdminTask.tklmCertImport
 ('[-fileName myTempPath\\myCertRequest3.cer
   -alias sklmCertificate3 -format base64
     -keyStoreName defaultKeyStore -usage DS8000]')
```

- REST interface
  a. Obtain a unique user authentication identifier to access IBM Security Key Lifecycle Manager REST services. For more information about the authentication process, see Authentication process for REST services.
  b. To invoke **Certificate Import REST Service**, send the HTTP POST request. Pass the user authentication identifier that you obtained in `Step a` along with the request message as shown in the following example.
     – SSL communication

```
POST https://localhost:9080/SKLM/rest/v1/certificates/import
Content-Type: application/json
Accept : application/json
Authorization: SKLMAuth authId=139aeh34567m
{"fileName":"/mycertfilenam.base64","alias","sklmSSLCertificate1",
"format":"base64",
"usage":"SSLSERVER"}
```

     – 3592 tape drives

```
POST https://localhost:9080/SKLM/rest/v1/certificates/import
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
{"fileName":"/mycertfilenam.base64","alias","sklmSSLCertificate2",
"format":"base64",
"usage":"3592"}
```

     – DS8000 Turbo drives

```
POST https://localhost:9080/SKLM/rest/v1/certificates/import
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
{"fileName":"/mycertfilenam.base64","alias","sklmSSLCertificate3",
"format":"base64",
"usage":"DS8000"}
```

3. A success indicator varies, depending on the interface:
   - Graphical user interface:

     The pending certificate entry is removed from the **Pending Certificates** table on the Welcome page. If there are no more certificates to be imported, the **Pending Certificates** table is removed from the Action Items section of the Welcome page.
   - Command-line interface:

     A completion message indicates success.
   - REST interface:

     The status code `200 OK` indicates success.

What to do next

Ensure that you back up the key materials to protect the certificate. Then, you might associate the certificate with one or more devices.

Certificate request problems

You must solve problems in either creating a certificate request, or enabling a returned certificate for use.

- Before you create a certificate request, solve these problems as administrator:

- **Problem:** You might not have permission to write to the certificate request file. Alternatively, there might not be sufficient free disk space, or the database might not be available.

  Solution: Ensure that your permissions are correct, that there is sufficient free disk space, and that the database connection is available. If not, make the appropriate corrections. Then, try the operation again.
- **Problem:** A value is not specified for the common name. The common name (`cn`) is part of the unique identification for the certificate. For example, the value of `cn` is used in the subject name for a certificate, which can identify whether a certificate that is being imported matches an original certificate request.

  Solution: Specify the common name for the certificate. Then, try the operation again.
- **Problem:** The certificate request file exists.

  Solution: The file name that you specified in the certificate request matches an existing certificate request file name. Specify a different file name for the certificate request. For example, specify `myUniqueRequest.crt`. Then, try the operation again.

• When you import a returned CA certificate, solve these problems:
- **Problem:** The subject name of the certificate that returned from a certificate authority does not match the subject name in the original certificate request.

  Solution: Correct the file name or alias specification. Then, try the import operation again.
- **Problem:** An error occurs while verifying the key and certificate. The certificate request that you submitted to a certificate authority and the certificate that returned, do not match.

  Solution: The problem might be an internal processing error. Collect any information that might be in the audit log and then contact IBM Software Support.
- **Problem:** The key in the certificate to be imported does not match the key in the original certificate request.

  Solution: You attempted to match a returned certificate to an incorrect certificate request. Import the certificate by using an alias that corresponds to this response. Then, try the operation again.
- **Problem:** When you import a certificate with the expiration year greater than 50 years, you might see these messages:

  Using command-line interface
  ```
  CTGKM0002E Command failed: javax.management.MBeanException:
  RuntimeException thrown in RequiredModelMBean while trying to invoke
  operation importCertificate
  ```

  Using graphical user interface
  ```
  Cannot import certificate to the keystore.
  javax.management.MBeanException: RuntimeException thrown in
  RequiredModelMBean while trying to invoke operation
  importCertificate
  ```

  Workaround: The certificate expiration period cannot be greater than 50 years. To modify the expiration period, change the value of the **`maximum.keycert.expiration.period.in.years`** parameter in the`SKLMConfig.properties` file.

Scenario: IBM Security Key Lifecycle Manager bulk replication setup

You can use IBM Security Key Lifecycle Manager to automatically replicate your key materials, configuration files, and other critical information from a primary master server up to 20 secondary clone servers. The automatic replication ensures continuous key and certificate availability to encrypting devices.

Note: The automatic replication process is run only when the new keys are added.

The data replication enables cloning of IBM Security Key Lifecycle Manager environments to multiple servers in a manner that is independent of operating systems and directory structure of the server.

The automatic replication ensures the availability of a backup system when the primary IBM Security Key Lifecycle Manager instance is unavailable. The backup system contains all the required keys and associated data. You can use graphical user interface, CLI commands, or REST interfaces for the following replication tasks:

- Schedule replication process
- Start and stop the replication tasks
- Provide a status on the replication tasks
- Perform replication configuration file functions

Replication configuration files

You can run IBM Security Key Lifecycle Manager replication as a stand-alone task. A valid replication configuration file must be available to start the automated replication process when the new keys are added.

IBM Security Key Lifecycle Manager uses properties in the `<SKLM_HOME>\config\ReplicationSKLMConfig.properties` configuration file to control the replication process. For example, `C:\Program Files (x86)\IBM\WebSphere\AppServer\products\sklm\config\ReplicationSKLMConfig.properties`. You can use the IBM Security Key Lifecycle Manager graphical user interface, command-line interface, or REST interface to change properties of the replication configuration file.

You can classify each system as:

- Master - the primary system that is being replicated.
- Clone - the secondary system that is being copied to.

The replication file of the master system can specify up to 20 clones. Each clone system is identified through an IP address or host name, and a port number. You can replicate IBM Security Key Lifecycle Manager environments to multiple clone servers in a manner that is independent of operating systems and directory structure of the server.

Notes:

- Scheduled replication takes place only when the new keys are added on the master system.
- There can be only one master system with a maximum of 20 clones. Multiple masters are not supported.

You can use the IBM Security Key Lifecycle Manager replication program to schedule automatic backup operation. You must configure properties only for the master server to back up data at regular intervals.

Master configuration file sample

```
replication.role=master
replication.auditLogName=replication.log
replication.MaxLogFileSize=1000
replication.MaxBackupNum=10
replication.MaxLogFileNum=5
replication.BackupDestDir=C\:\\IBM\WebSphere\\AppServer\\products\\sklm\\restore
backup.ClientIP1=myhost1
backup.ClientPort1=2222
backup.EncryptionPassword=password
backup.ReleaseKeysOnSuccessfulBackup=false
backup.CheckFrequency=60
backup.TLSCertAlias=ssl_cert
replication.MasterListenPort=1111
```

- *master* is the default replication role. Specify it to avoid confusion.
- Specify at least one clone with the **backup.ClientIPn** and **backup.ClientPortn** parameter.
- Ensure that the specified ports are available and are not currently in use by IBM Security Key Lifecycle Manager or by any other process.
- You can specify a maximum of 20 clone systems.
- The **backup.TLSCertAlias** parameter must specify a certificate that exists on the master and all clone systems.
- Specify a password to encrypt and decrypt backups. This password becomes obfuscated in the replication configuration file after IBM Security Key Lifecycle Manager reads it for the first time.

Clone configuration file sample

```
replication.role=clone
replication.MasterListenPort=1111
replication.BackupDestDir=C\:\\IBM\WebSphere\\AppServer\\products\\sklm\\restore
replication.MaxLogFileSize=1000
replication.MaxBackupNum=3
replication.MaxLogFileNum=4
restore.ListenPort=2222
```

- On the clone system, specify the parameter value `replication.role=clone`.
- The **restore.ListenPort** parameter must specify the port number that is specified in the **backup.ClientIPn** parameter on the master system.

For complete details of all the available replication configuration parameters, see Reference information section in the IBM Security Key Lifecycle Manager documentation.

Inter-server communication

The Transport Layer Security (TLS) protocol is used for secure communication between the master and clone systems.

An existing private key must be available in the IBM Security Key Lifecycle Manager keystore of the master and all its clone systems. You must set alias of this key on the master system in the **backup.TLSCertAlias** parameter of `ReplicationSKLMConfig.properties` configuration file. If the same key is not available on both the master and clone systems, you cannot start communication between the systems to run the replication task. You can use the graphical user interface, command-line interface, or REST interface to change properties of the replication configuration file.

Replication schedules

Configure the properties of the `ReplicationSKLMConfig.properties` file to schedule IBM Security Key Lifecycle Manager automated replication process.

Use the graphical user interface, command-line interface, or REST interface to configure properties of the replication configuration file for scheduling replication process. Scheduled replication takes place only when the new keys are added on the master system. You can also use the IBM Security Key Lifecycle Manager replication program to schedule automatic backup operation. You must configure properties only for the master server to back up data at regular intervals.

You can configure the schedule so that IBM Security Key Lifecycle Manager checks whether the replication is required periodically, and starts the process if changes are made. You can also specify a time of day to run a replication when required. Configure the **`backup.CheckFrequency`** parameter to specify how often IBM Security Key Lifecycle Manager checks the master system for updates. Replication triggers when the updates take place. The value is set in hours with 1 hour as the default value.

To specify a time of day, configure the **`backup.DailyStartReplicationBackupTime`** parameter. You must specify a time in 24-hour clock format (HH:MM). Replication takes place only when the master system changes since the last replication.

By default, the clone system restores a backup as soon it is received from the master system. To specify the restoration time, add the **`restore.DailyStartReplicationRestoreTime`** parameter in the replication configuration file of the clone system. You must specify time in 24-hour clock format (HH:MM).

You can use the Replication page to force an ad hoc replication to all the defined clones, or a specific replication. Alternatively, you can use the following CLI command or the REST interface:

- **`tklmReplicationNow`**
- **`Replication Now REST Service`**

Replication audit records

IBM Security Key Lifecycle Manager replication records audit information to the IBM Security Key Lifecycle Manager audit log file.

IBM Security Key Lifecycle Manager replication program provides a facility to write replication-specific audit records to its own discrete audit log file. Replication audit log records all the actions that are related to replication process. By default, location of the replication audit log file is `<SKLM_HOME>\logs\replication\replication_audit.log`.

Use the graphical user interface, command-line interface, or REST interface to set audit properties in the `ReplicationSKLMConfig.properties` file. In the configuration file, you can configure audit properties, such as audit log file location, log file name, log file size, maximum number of log files to keep, or maximum number of backup files to keep.

Setting up replication process

You must set up a basic environment in IBM Security Key Lifecycle Manager to run the replication process.

About this task

This topic describes how to set up replication process by using the IBM Security Key Lifecycle Manager command-line interface commands and the REST interfaces for replication. For information about setting up replication by using the graphical user interface, see Replication settings for clone and master servers.

Procedure

1. Set up the IBM Security Key Lifecycle Manager master system.
2. Add keys and devices so that it is ready to serve the needed keys.
3. Specify a `SSLSERVER` certificate for the replication to work. You can create this certificate by using the GUI, CLI command, or the REST interface as shown in the following examples:

   Graphical user interface

   a. Log on to the graphical user interface.
   b. Click **Advanced Configuration** > **Server Certificates**.

   Command-line interface

   Type the **tklmCertCreate** command on one line. For example, to create a self-signed certificate, type:

   ```
   print AdminTask.tklmCertCreate('[('[-type selfsigned -alias
   sklmSSLCertificate -cn sklmssl -ou accounting -o myCompanyName
   -country US -keyStoreName defaultKeyStore -usage SSLSERVER
   -validity 999]')
   ```

   REST interface

   To create a self-signed certificate, you can use **Certificate Generate Request REST Service**. Send the following HTTP request by using a REST client:

   ```
   POST https://localhost: 9080/SKLM/rest/v1/certificates
   Content-Type: application/json
   Accept: application/json
   Authorization: SKLMAuth authId=139aeh34567m
   Accept-Language : en
   {"type":"selfsigned","alias":"sklmCertificate","cn":"sklm","ou":"sales",
   "o":"myCompanyName","usage":"3592","country":"US","validity":"999", "
   algorithm ": " RSA "  }
   ```

4. Create a backup of the master IBM Security Key Lifecycle Manager as shown in the following examples:

   Graphical user interface

   a. Log on to the graphical user interface.
   b. Click **Backup and Restore**.

   Command-line interface

   Type the **tklmBackupRun** command:

   ```
   print AdminTask.tklmBackupRun
   ('[-backupDirectory C:\\wasbak1\\sklmbackup1 -password myBackupPwd]')
   ```

   REST interface

   To create a backup, use **Backup Run REST Service**. Send the following HTTP request by using a REST client:

   ```
   POST https://localhost:9080/SKLM/rest/v1/ckms/backups
   Content-Type: application/json
   Accept: application/json
   Authorization: SKLMAuth authId=139aeh34567m
   Accept-Language : en
   {"backupDirectory":"/sklmbackup1","password":"myBackupPwd"}
   ```

5. Take the backup that is created in step 2 and copy it to each of your IBM Security Key Lifecycle Manager clone systems. Restore this backup to each of these systems as shown in the following examples:

   Graphical user interface

   a. Log on to the graphical user interface.
   b. Click **Backup and Restore**.

   Command-line interface

   Type the **tklmBackupRestoreRun** command on one line:

   ```
   print AdminTask.tklmBackupRunRestore
   ('[-backupFilePath /opt/sklmbackup/sklm_v2.5_20081012074433_backup.jar
     -password myBackupPwd]')
   ```

   REST interface

   To restore a backup, you can use **Backup Run Restore REST Service**. Send the following HTTP request by using a REST client:

   ```
   POST https://localhost:9080/SKLM/rest/v1/ckms/restore
   Content-Type: application/json
   Accept: application/json
   Authorization: SKLMAuth authId=139aeh34567m
   Accept-Language : en
   {"backupFilePath":"/sklmbackup","password":"myBackupPwd"}
   ```

6. Create the `ReplicationSKLMConfig.properties` replication configuration file on the master system. This configuration file must be a text file and you must locate the file in the same directory as the IBM Security Key Lifecycle Manager properties file, for example `C:\Program Files (x86)\IBM\WebSphere\AppServer\products\sklm\config\ReplicationSKLMConfig.properties`.

   The following example shows the fields that are required on the master to allow the replication task to start. You must do the following steps:

   - Set role to master.
   - Identify certificate from Step 1 and provide at least one clone server and port number.
   - Define a master listen port and choose a password.

   ```
   backup.EncryptionPassword=mypassword
   backup.TLSCertAlias=sklmSSLCertificate
   backup.ClientIP1=myhostname
   backup.ClientPort1=2222
   replication.MasterListenPort=1111
   ```

   The **backup.EncryptionPassword** property can contain of characters, numbers, or special characters. The product obfuscates this property when replication is first run. The **backup.TLSCertAlias** property specifies the alias of the certificate and the private key that is used to communicate to the clone created in Step 1.

   The **replication.MasterListenPort** property specifies the port that the master system listens on for certain responses from the clones. The **backup.ClientIP1** and the **backup.ClientPort1** properties define the clone. The **backup.ClientIP1** property can be either a host name or an IP address. The **backup.ClientPort1** property specifies the port that the client is listening on. To define other clones, you must specify the **backup.ClientIP*** and **backup.ClientPort*** properties, where "*" is a number 2 - 5, like you did for the first set.

7. Create the `ReplicationSKLMConfig.properties` replication configuration file on the clone system. This configuration file must be a text file and you must locate the file in the same directory as the IBM Security Key Lifecycle Manager properties file, for example, `C:\Program Files (x86)\IBM\WebSphere\AppServer\products\sklm\config\ReplicationSKLMConfig.properties`.

The following example shows the fields that are required on the clone to allow the replication task to start. You must do the following steps:

- Set role to clone.
- Define a master listen port.
- Define a restore listen port. The port must be the same port number that is coded in the corresponding **backup.ClientIP*** parameter on the master.

```
replication.role=clone
backup.TLSCertAlias=sklmSSLCertificate
replication.MasterListenPort=1111
restore.ListenPort=2222
```

The **replication.role** property is required for clones. By default, the value of this property is master. The **backup.TLSCertAlias** property must set to the certificate created in Step 1 just like on the master. This property is used to send the status of the clone when replication is postponed for a later time, or the restore process takes longer than the master is waiting for a response.

The **replication.MasterListenPort** property specifies the port to send status when replication is postponed for a later time, or the restore process takes longer than the master is waiting for a response. The last property **restore.ListenPort** is the port that the clone listens on for replication requests from the master.

8. Restart IBM Security Key Lifecycle Manager on master and clone systems. You can see the following messages on a clone and master system: Use the **tklmReplicationStatus** CLI command to ensure that the replication task is running. You can see the following messages on a master and a clone system:

Command-line interface

You can use the following CLI command to ensure that the replication task is running:

```
print AdminTask.tklmReplicationStatus()
```

Master system

```
1.CTGKM2215I The Security Key Lifecycle Manager Replication
task is UP. Role set to: MASTER
CTGKM2218I The last completed replication took place at
Thu Jun 19 14:50:59 WST 2015
CTGKM2217I The next scheduled replication is due at
Fri Jun 20 17:03:36 WST 2015
```

Clone system

```
CTGKM2215I The SKLM Replication task is UP. Role set to: CLONE
CTGKM2220I No previous successful replications.
CTGKM2221I No replication currently scheduled.
```

REST interface

Use **Replication Status REST Service** to ensure that the replication task is running. Send the following HTTP request by using a REST client:

```
GET https://localhost:9080/SKLM/rest/v1/replicate/status
Content-Type: application/json
Accept: application/json
Authorization: SKLMAuth authId=139aeh34567m
```

Master system

```
Status Code : 200 OK
Content-Language: en
[
  {code:"CTGKM2215I", "status":"CTGKM2215I The Security Key
Lifecycle Manager  Replication task is UP. Role set to: MASTER"},
  {code:"CTGKM2218I", "status":"CTGKM2218I The last completed
replication took place at Thu Jun 19 14:50:59 WST 2015."} ,
```

```
  {code:"CTGKM2217I", "status":"CTGKM2217I The next scheduled
replication is due at Fri Jun 20 17:03:36 WST 2015." }

]
```

Clone system

```
Status Code : 200 OK
[
  { code:"CTGKM2215I", "status":"CTGKM2215I The Security Key
Lifecycle Manager  Replication task is UP. Role set to: CLONE"}
, { code:"CTGKM2220I", "status":"CTGKM2220I No previous
   successful replications."} ,
  { code:"CTGKM2217I", "status":"CTGKM2221I No replication
   currently scheduled." }

]
```

9. Replication is now set up and replication checks for changes every 60 minutes. You can change this interval, set up a certain time every day for replication to check for changes. You can also use the **tklmReplicationNow** CLI command or **Replication Now REST Service** to run a replication task immediately.

Replication problems and resolution

You must consider possible issues on the clone and master systems when you run the IBM Security Key Lifecycle Manager replication task.

Incomplete replication

- Ensure that the SSL certificate and private key that is specified in the **backup.TLSCertAlias** parameter are available on both the master and clone servers.
- Ensure that the port numbers specified for replication communication are not currently in use by other software products.
- Check the server names or IP addresses specified in the replication configuration file are correct and accessible from the master server.
- Check whether the replication task is up on each server by running the **tklmReplicationStatus** command, **Replication Status REST Service**, or the status on the **Replication** section of IBM Security Key Lifecycle Manager welcome page.
- For DB2 replication, ensure that date/time of master and clone servers are closely synchronized. Large discrepancies can lead to restore failure.
- Check the replication configuration file to ensure that the minimum required parameters are defined, without typographical error.
- Define a maximum of one master and 20 associated clones. At least one clone must be defined.
- Check the replication audit file to get more information about replication failure.

Replication is not taking place at scheduled time

- Scheduled replications take place only when you create **new** key material.
- When both specific replication time and a check interval are set in the master replication configuration file, the time overrides the check interval.

Clone system replication

- The clone system restarts after replication.
- Maintain the availability of your clone servers. You can specify a specific time-of-day to complete the replication with the **restore.DailyStartReplicationRestoreTime** parameter. For example, to run

restores only at 11 p.m., regardless of when the backup file is received, code the following property in the configuration file:

```
restore.DailyStartReplicationRestoreTime=23:00
```

Scenario: Setup for SSL handshake between IBM Security Key Lifecycle Manager server and client device

The SSL handshake enables IBM Security Key Lifecycle Manager server and client devices to establish the connection for secure communication. IBM Security Key Lifecycle Manager provides the Server Configuration Wizard to configure server and the client device for SSL handshake.

You must complete the following steps in the wizard for SSL/TLS handshake:

1. Creating a self-signed SSL/KMIP server certificate.
2. Exporting the SSL/KMIP server certificate that is created in Step 1 to a certificate file in an encoded format for use by the client device. You can also export an existing certificate.
3. Importing client communication certificate to the IBM Security Key Lifecycle Manager server.

Creating a self-signed SSL/KMIP server certificate

As a first activity, you might create an SSL/KMIP server certificate for use with IBM Security Key Lifecycle Manager.

Procedure

1. Log on to the graphical user interface.
2. Click the **Review the configuration parameters and/or create an SSL server certificate** link.

   Immediately after you install IBM Security Key Lifecycle Manager, the **Review the configuration parameters and/or create an SSL server certificate** link is the only available option to configure IBM Security Key Lifecycle Manager for SSL/TLS handshake with the client devices. This link is not visible if you previously created an SSL server certificate.
3. Alternatively, on the Welcome page, click **Configuration** > **SSL/KMIP** > **Launch Server Configuration Wizard**.
4. Click **Create SSL/KMIP Server Certificate**.
5. On the Add SSL/KMIP Certificate dialog, select **Create self-signed certificate**.
6. Specify values for the parameters according to your requirements.
7. Click **Create Certificate**.

What to do next

You might need to export the IBM Security Key Lifecycle Manager SSL/KMIP server certificate that you created to a file in an encoded format for use by the client device. Click the **Export Certificate** link or click the **Export SSL/KMIP Server Certificate** tab. You can also export an existing SSL/KMIP server certificate by selecting **Use an existing certificate**. See “Exporting a server certificate” on page 19.

Exporting a server certificate

You must export the IBM Security Key Lifecycle Manager SSL/KMIP server certificate to a file in an encoded format for use by the client device. The client device imports this certificate for secure communication with the server.

Procedure

1. Log on to the graphical user interface.
2. On the Welcome page, click **Configuration** > **SSL/KMIP** > **Launch Server Configuration Wizard**.
3. To create a self-signed certificate, click **Create SSL/KMIP Server Certificate**. See the "Creating a self-signed SSL/KMIP server certificate" on page 18 topic for more information.
4. Click **Export SSL/KMIP Server Certificate**.
5. On the Export Certificate dialog, specify values for the parameters according to your requirements. For example, you might specify **BASE64** or **DER** for certificate file format.

   Note: If you do not specify the path, certificate is exported to the default location where IBM Security Key Lifecycle Manager is installed.
6. Click **Export Certificate**.

What to do next

You might go the next step to import the client device communication certificate for secure communication between IBM Security Key Lifecycle Manager server and the client device. Click the **Go to Next Step** link or select **Import SSL/KMIP Server Certificate**. See "Importing a client communication certificate."

Importing a client communication certificate

You must import communication certificate to the IBM Security Key Lifecycle Manager server for secure communication with the client device.

Procedure

1. Log on to the graphical user interface.
2. On the Welcome page, click **Configuration** > **SSL/KMIP** > **Launch Server Configuration Wizard**.
3. To create a self-signed certificate, click **Create SSL/KMIP Server Certificate**. See the "Creating a self-signed SSL/KMIP server certificate" on page 18 topic for more information.
4. Click **Export SSL/KMIP Server Certificate** to export the IBM Security Key Lifecycle Manager SSL/KMIP server certificate to a file in an encoded format for use by the client device. See the "Exporting a server certificate" for more information.
5. Click **Import SSL/KMIP Client Certificate**.
6. On the Import Certificate dialog, specify values for the parameters according to your requirements.
7. Click **Import**.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law :

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to

IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Index

W