integrity™ lifecycle manager

Server for IBM Security Access Manager

11.1

UNITED STATES GOVERNMENT RIGHTS

PTC Inc., 140 Kendrick Street, Needham, MA 02494 USA

Contents

1

Introduction

With the 10.9 release of Integrity, PTC provides support for client side certificate-based authentication using IBM Security Access Manager 8.0 (ISAM).

This functionality enables Integrity Lifecycle Manager clients to connect to Integrity Lifecycle Manager server using client side certificate-based authentication. ISAM intercepts all requests from the client and performs authentication. Only on successful authentication, requests are forwarded to the server.

IBM Security Access Manager is a web-based security management system that supports centralized authentication and security policy enforcement. ISAM also provides single sign-on support for Web-based applications.

This document provides details on configuring ISAM, setting the required security properties on the Integrity Lifecycle Manager server, and enabling secure communications. Once the configuration is complete, Integrity Lifecycle Manager is ready to support client side certificate-based authentication using ISAM for authentication.

2

Configuration Overview

A new security scheme called “External Authentication” has been introduced in Integrity Lifecycle Manager to support user authentication using ISAM. This External Authentication is supported for the Integrity Lifecycle Manager’s client (GUI), Web Client, Command Line Interface (CLI), and for APIs using client-side integration point. The implementation uses the MKS Domain for ‘Groups’ management. MKS Domain support for user management is limited to server-side triggers that use a local integration point (trigger running on the same physical machine as that of server).

In this configuration, ISAM server acts a reverse proxy for Integrity Lifecycle Manager server and conceals the identity of the actual server. Integrity Lifecycle Manager clients are not aware about back-end Integrity Lifecycle Manager server details and always connect to ISAM server with virtual host details. Every request intended for Integrity Lifecycle Manager server is intercepted by ISAM server, authenticated as per the scheme and re-routed to the back-end server on successful authentication. In no circumstance, Integrity Lifecycle Manager clients are allowed to make direct connections with the Integrity Lifecycle Manager server.

The following diagram illustrates the general network configuration involving Integrity Lifecycle Manager clients, ISAM Server, and Integrity Lifecycle Manager server. Note that the ISAM server is acting as reverse proxy to two back-end Integrity Lifecycle Manager servers, which are hosted on `host1.ptc.com` and host2.ptc.com respectively. Integrity Lifecycle Manager clients always connect to servers using their virtual host name defined by DNS mapping and virtual host junctions.

Note

When connecting to an Integrity Lifecycle Manager server, the Integrity Lifecycle Manager client checks if the host name used to connect to the server matches the common name (CN) of the ISAM server certificate. If there is a mismatch, the connection fails and the following error message is displayed:

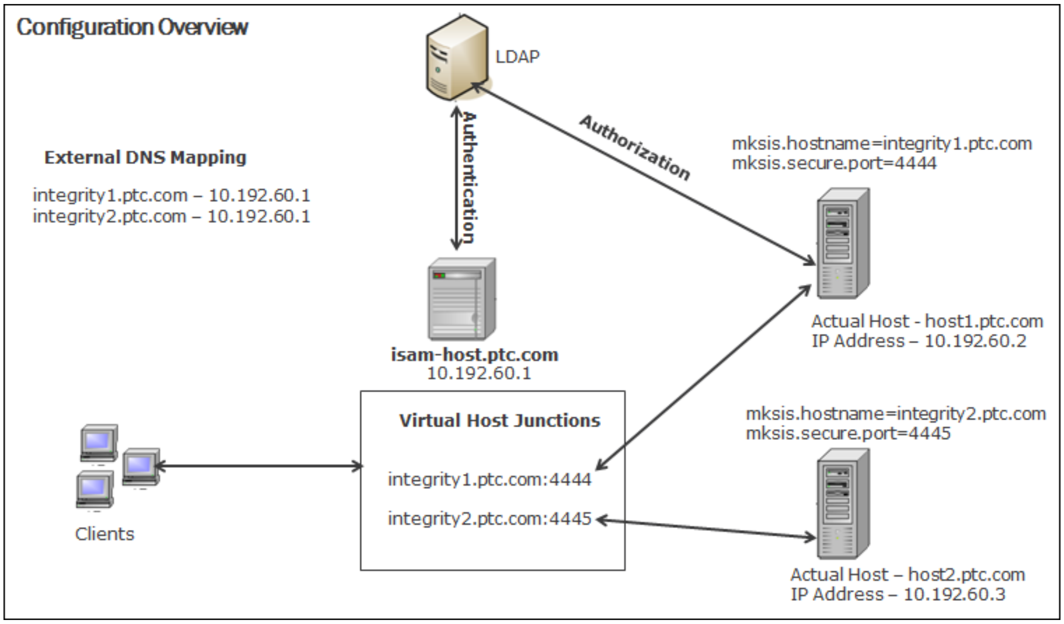
```
Error connecting to <givenhost>:<port_number>:Host name
does not match the certificate subject provided by the
peer (CN= <hostname>)
```

where:

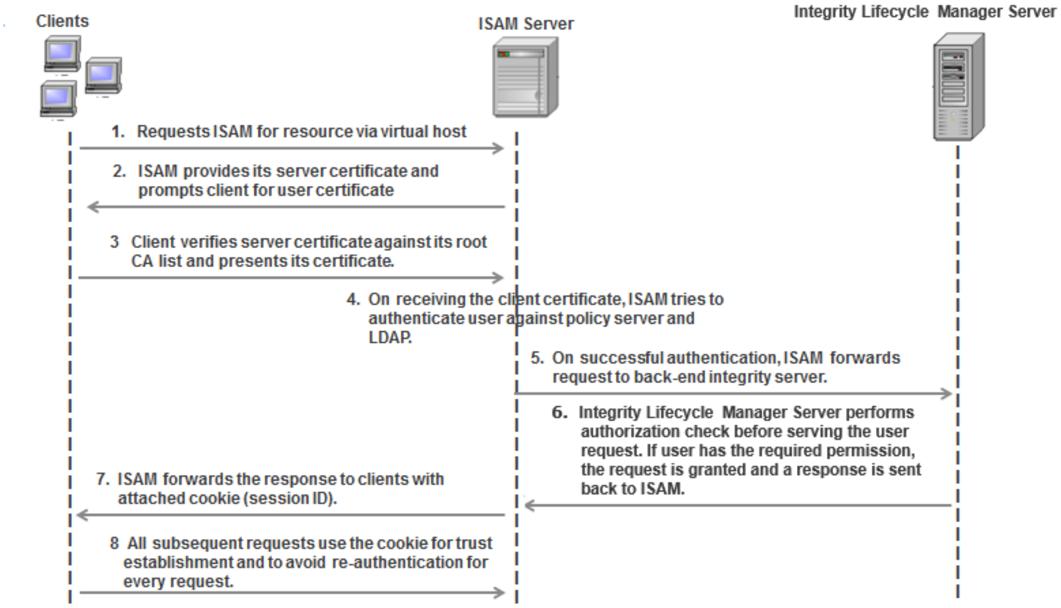
- <givenhost> is the host name used to connect to Integrity Lifecycle Manager
- <hostname> is the CN on the certificate

For example, consider the configuration in the following image. If you connect to Integrity Lifecycle Manager server using https://integrity1.ptc.com:4444 and the CN on the ISAM certificate is isam-host.ptc.com, then you will get an error message as follows:

```
Error connecting to integrity1.ptc.com:4444: Host name
<integrity1.ptc.com> does not match the certificate
subject provided by the peer (CN= isam-host.ptc.com)
```



The following request flow diagram describes initial requests flowing between all three entities involved in the deployment:



1. Requests ISAM for resource via virtual host

2. ISAM provides its server certificate and prompts client for user certificate

3 Client verifies server certificate against its root CA list and presents its certificate.

4. On receiving the client certificate, ISAM tries to authenticate user against policy server and LDAP.

5. On successful authentication, ISAM forwards request to back-end integrity server.

6. Integrity Lifecycle Manager Server performs authorization check before serving the user request. If user has the required permission, the request is granted and a response is sent back to ISAM.

7. ISAM forwards the response to clients with attached cookie (session ID).

8 All subsequent requests use the cookie for trust establishment and to avoid re-authentication for every request.

3

Assumptions

The following assumptions are made in this document:

- ISAM server is already installed at your site and required ISAM instance(s) is created.
- Integrity Lifecycle Manager server and client are installed on respective machines.
- You are familiar with ISAM Local Management Interface (LMI) and you know how to use it for administration tasks such as ISAM instance configuration, junction creation, and junction configuration.
- Client certificates are distributed to end users and are available in end users' personal store.

The focus of the document is to provide details on configuring the ISAM server virtual host junction, configuring ISAM server for certificate-based authentication, setting the required security properties on the Integrity Lifecycle Manager server, configuring Integrity Lifecycle Manager client and enabling secure communications (mutual SSL).

4

Limitations

The following limitations are applicable when using ISAM as an external authenticator:

- Integrity Lifecycle Manager clients are supported only on Windows platform.
- No FSA or Split Server configuration is allowed.
- No support is provided for Workflows and Documents staging functionality.
- This release supports only certificate-based authentication scheme with ISAM.
- Integrations using server-side integration points are not supported. Only client-side integrations that use the common session are supported. For example, Microsoft Visual Studio and Eclipse Platform integrations.
- Only server side triggers that execute on the same physical machine as that of Integrity Lifecycle Manager server are supported. Authentication for these triggers is performed using the MKS Domain security realm.

5

Detailed Configuration

DNS Mapping

ISAM server uses the concept of junctions to logically combine the web space of the back-end server(s) and re-route the requests to an appropriate back-end server. To support different use cases, three types of junctions are available for use. Out of these, Integrity Lifecycle Manager supports only virtual host junctions as they are an ideal fit for this deployment.

Configuration for virtual host junctions requires that the external DNS maps all virtual host names to the IP address (or addresses) of the ISAM server. When the user makes a request to the host name of the junctioned server, the request is routed to ISAM.

For example, in the illustrative diagram used in overview section, a single ISAM server acting as reverse proxy for two backend Integrity Lifecycle Manager servers requires two virtual host entries in DNS mapping table, each pointing to the IP address of the ISAM server.

Table 1: Sample Hosts File Entry for DNS Mapping

IP Address	Virtual Host Name
10.192.60.1	integrity1.ptc.com
10.192.60.1	integrity2.ptc.com

Note

The entries in the table are for illustration purpose only. See the corresponding DNS Server Configuration Guide for actual instructions. On successful configuration, resolve users requests to the ISAM server host.

Configuring the Integrity Lifecycle Manager Server

The server configuration starts with the installation of Integrity Lifecycle Manager server. For details on completing the installation, refer to the *Integrity Lifecycle Manager Installation and Upgrading Guide*.

As this deployment offers only private security scheme, the next step is to enable SSL connection for the Integrity Lifecycle Manager server. For details on configuring Integrity Lifecycle Manager on SSL, refer "Enabling SSL" section of the *Integrity Lifecycle Manager Help Center*.

Upgrading Information

This guide assumes an upgrading path from PTC Integrity 10.x, directly to this release. There are no additional steps required for upgrading to this release.

For general information on upgrading to this release, refer to the *Integrity Lifecycle Manager Installation and Upgrading Guide* available on the PTC Integrity eSupport portal (http://support.ptc.com/integrity.htm).

Configuring the Integrity Lifecycle Manager Server Properties

This section describes the properties required in the Integrity Lifecycle Manager server properties file(`is.properties`) for ISAM configuration.

Perform the following steps to configure the server properties:

1. In a text editor, open the server configuration file `<Integrity Lifecycle Manager server installdir>/config/properties/is.properties` where `<Integrity Lifecycle server installdir>` is the path to the directory where you have installed the server.
2. Add or uncomment the following properties:
   a. Disable the clear port.

   ```
   mksis.clear.port=0
   ```

   b. Ensure that the secure port is set to a valid value.

   ```
   mksis.secure.port=4444
   ```

   Note

   Integrity Lifecycle Manager server can be enabled to listen on both clear and private connections. However, for this deployment, only private connections should be enabled.

   c. Specify one of the virtual host names mentioned in the table “Sample Hosts File Entry for DNS Mapping”. This is the public name which will be used in the notification links, Email links, CP links, and so on.

```
mksis.hostname=integrity1.ptc.com
```

Note

Integrity Lifecycle Manager server should be able to lookup the IP address of the above-mentioned host name. In case the lookup fails, errors are generated. You can do this by either by enabling lookup with the DNS server or through a local hosts file entry.

d. Enable HTTP protocol for server communication. This property allows the Integrity Lifecycle Manager server to use HTTP protocol for communication with ISAM server.

```
mksis.rmi.transport=http
```

Configuring the Integrity Lifecycle Manager Security Properties

This topic describes the properties required in the Integrity Lifecycle Manager server properties file (`security.properties`) to configure the server for ISAM authentication.

Security scheme

To support ISAM authentication, Integrity Lifecycle Manager has introduced a new security scheme called External Authentication. This scheme is represented in security property file as `externalauth_private` since it is combination of LDAP authentication domain and private transport protocol.

Though LDAP authentication domain , which has capability of supporting both users and groups is used by the domain, only users must be maintained in LDAP for this deployment. Groups' definitions are managed in MKS Domain. Hence configuring LDAP properties related to users have impact while group-related properties should not be configured.

Perform the following steps to update the `security.properties` file:

1. In a text editor, open the server configuration file `<Integrity Lifecycle Manager server installdir>/config/properties/security.properties` where `<Integrity Lifecycle Manager server installdir>` is the path to the directory where you have installed the Integrity Lifecycle Manager server.
2. Add or uncomment the following properties:

    a. The following property allows the server to retrieve groups definitions from `mksdomain`. This allows you to use corporate LDAP realm for users. As a result authentication can be taken care by ISAM. Similarly, using `mksdomain` for groups allows Integrity Lifecycle Manager to enforce authorisation by defining various ACLs on them.

    ```
    mks.groupsDomain=mksdomain
    ```

    b. Ensure that the following security scheme-related properties exist and their values are mentioned as follows:

    ```
    mks.security.scheme.externalauth_
    private.description=Authenticate using External
    authenticator over private connection
    ```

    ```
    mks.security.scheme.externalauth_
    private.connectionProvider=
    mks.frame.client.PrivateConnectionProvider
    ```

    ```
    mks.security.scheme.externalauth_
    private.authentication=externalauth
    ```

    ```
    mks.security.scheme.externalauth_private.realm=
    externalauth
    ```

    c. Ensure the following property values correspond to the values mentioned for client identity headers during junction creation. For details, refer section Configuring ISAM Server Instance and Junction Creation.

    ```
    mks.security.externalauth.headers.username=iv-user
    ```

    ```
    mks.security.externalauth.headers.remoteClientIP=
    iv-remote-address
    ```

3. To set up the Integrity Lifecycle Manager server to communicate with your LDAP security realm, uncomment the properties that correspond to your security realm. Refer the *Integrity Lifecycle Manager Help Center* for supported LDAP-compliant security realms and their configuration. Typical settings are pre-configured for each of these supported realms.

    All the limitations and conditions for LDAP realm such as LDAP Referrals, User Fetch limit, and so on also hold true for external authentication domain.

Security Policies

When external authentication scheme is in use, all the incoming requests are expected to be already authenticated by ISAM server. This implies that other traditional security schemes, which expect username-password combination for authentication cannot be configured for incoming user requests from ISAM server. However, server-side triggers residing on the server machine can use `mksdomain_private` scheme provided that `externalauth_private` scheme precedes it in the order.

For example, the following security policy is allowed:

```
mks.security.policy.scheme.default=externalauth_
private,mksdomain_private
```

However, you cannot have the following security scheme:

- A different security scheme based on full or partial client IP address(es).

  ```
  mks.security.policy.scheme.1.2.3.4=mksdomain_private
  ```

  ```
  mks.security.policy.scheme.1.2.3=ldap_private
  ```
- `mksdomain_private` scheme preceding the `externalauth_private` scheme.

  ```
  mks.security.policy.scheme.default= mksdomain_
  private, externalauth_private
  ```

For more details on restrictions related to same `username` in multiple schemes, refer “Server Security” chapter of the *Integrity Lifecycle Manager Help Center*.

Configuring the ISAM Server Instance

Before beginning with the configuration, it is assumed that you have already installed ISAM server and created a server instance that is ready for configuration. If not, refer topics in the BM Security *Access Manager for Web 8.X Appliance Administration* and *IBM Security Access Manager for Web 8.X Web Administration* guides.

This chapter describes only those configuration properties that are required for Integrity Lifecycle Manager server and client deployment. For all other properties that could impact performance tuning, diagnostics, policy management, and so on., refer the appropriate ISAM documentation. To keep this document concise, only necessary details describing properties and their significance are included. For detailed explanation, refer ISAM documentation.

Common Configuration

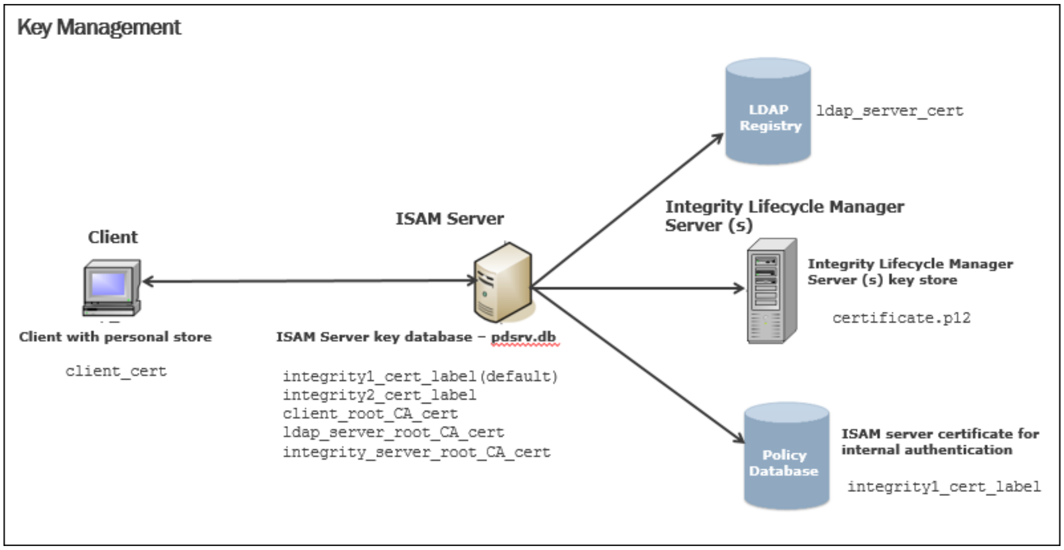
This configuration is applicable to both clients and junctioned Integrity Lifecycle Manager servers. Before beginning with this configuration, refer appropriate sections in the following guides:

- The “Runtime environment configuration” section of the *IBM Security Access Manager for Web 8.X Web Administration* guide for steps involving Policy server and LDAP configuration.
- “WebSEAL instance management” and”"Web server configuration” sections of the *IBM Security Access Manager for Web 8.X Web Reverse Proxy Configuration* guide to familiarize yourself with ISAM configuration files and their locations, and the procedure to modify them.

Key Management

As this deployment supports only client certificate-based authentication, enabling SSL communication between Integrity Lifecycle Manager clients and ISAM server is a pre-requisite that you should meet.

This section explains the administration and configuration tasks required to set up ISAM to handle client-side and server-side digital certificates used for authentication over SSL.


Key Management
LDAP Registry
ldap_server_cert
Integrity Lifecycle Manager Server (s)
Client
ISAM Server
Integrity Lifecycle Manager Server (s) key store
certificate.p12
Client with personal store
client_cert
ISAM Server key database – pdsrv.db
integrity1_cert_label(default)
integrity2_cert_label
client_root_CA_cert
ldap_server_root_CA_cert
integrity_server_root_CA_cert
Policy Database
ISAM server certificate for internal authentication
integrity1_cert_label

For the setup shown in the above image, you need the following three CA-signed certificates. Before you proceed ahead with next configuration steps, ensure you have below signed certificates and root CA certificates used to sign them:

- ISAM Server Certificate—Used to identify to SSL client and junction backend server required for mutual authentication.
- Integrity Lifecycle Manager server Certificate—-Used for SSL communication with ISAM server.
- End user Digital Certificate —Distributed to end users for authentication with ISAM server.

Note

You can use Single server certificate for both ISAM and Integrity Lifecycle Manager servers as they represent one virtual host. But for the purpose of illustration, in this configuration, we are using two separate certificates, one for each server.

ISAM stores its server certificate and CA root certificates in a key database (KDB) file. ISAM refers to its database of Certificate Authority (CA) root certificates to validate the clients accessing its services and to validate junctioned back-end Integrity Lifecycle Manager server.

The ISAM Web Gateway appliance provides a Local Management Interface (LMI) to set up and manage the certificate key database. This database contains one or more ISAM server/client certificates and the CA root certificates. ISAM includes the following components at installation to support SSL authentication using digital certificates:

- A default key database (`pdsrv.kdb`)
- A default key database stash file (`pdsrv.sth`) and password ("`pdsrv`")
- Several common CA root certificates

Configuring the ISAM Key Database File

The configuration provided in the following steps are based on the assumption that the same KDB file is used to authenticate both the clients and the junctioned servers. If your configuration setup requires separate KDB files, then refer the *IBM Security Access Manager for Web 8.X Web Reverse Proxy Configuration* guide to configure the applicable properties.

1. The `webseal-cert-keyfile` property located in the `[ssl]` stanza of the ISAM configuration file identifies the default certificate key database. For example:

```
[ssl]
```

```
webseal-cert-keyfile = pdsrv.kdb
```

You can either use the default key database or create a new one through LMI page as follows:

a. On the **Manage** tab, click **SSL Certificates** under **Secure Settings**.

b. Click **New** in the page that opens and enter a name for new certificate database.

c. If a new KDB is created, associate it with the instance either by modifying the above property directly or through LMI using the following step:

- On the **Secure** tab, select the required ISAM instance and click **Edit**. The **Reverse Proxy Basic Configuration** page opens,
- On the **SSL** tab, select a new KDB file from the **SSL Certificate Key File** list.

2. Import the CA signed ISAM server certificate into the aforementioned key database. If you have used ISAM's GSKit to create Certificate Signing Request (CSR) and received a signed certificate as part of this step, refer ISAM and GSKit documentation on importing the certificate. You can also refer the appropriate documentation for importing any Intermediate CA certificate into the key database.

   To import the certificate through the LMI page, perform the following steps:

   a. On the **Manage** tab, click **SLL Certificates** under **Secure Settings**.

   b. Select the KDB file in the page that opens.

   c. Click **Manage ▸ Edit SSL Certificate Database**

   d. Click the **Personal Certificates** tab in the **Edit SSL Certificate Database** page that opens.

   e. Click **Manage ▸ Import** and browse to the certificate file and enter the password for successful import.

3. Import the root CA certificate used for signing Integrity Lifecycle Manager server certificate and user digital certificate into key database s follows:

   a. On the LMI page, click **Manage ▸ SSL Certificate** and select the KDB file.

   b. Click **Manage ▸ Edit SSL Certificate Database**

   c. Click the **Signer Certificates** tab in the **Edit SSL Certificate Database** page that opens.

   d. Click **Manage ▸ Import** and browse to the certificate file and enter the label for successful import.

For more details on other certificate management tasks, refer 'Key management' section of the *IBM Security Access Manager for Web 8.X Web Reverse Proxy Configuration* guide.

Configuring the ISAM Server for Integrity Lifecycle Manager Client Authentication

This section outlines the configuration steps required to configure ISAM server instance for Integrity Lifecycle Manager client authentication. All the properties mentioned have been either added or modified from the ISAM instance configuration file. You can edit this file through LMI as follows:

1. Click **Secure ▸ Reverse Proxy**.
2. Select the ISAM instance in the **Reverse Proxy** page that opens.
3. Click **Manage ▸ Configuration ▸ Edit Configuration File** and modify the properties.

After modifying the properties, deploy the changes to the selected instance and restart that instance for changes to take effect.

Perform the following configuration steps:

1. If single backend server is used with an ISAM instance, you need not create a separate 'interface' in ISAM for listening on different ports. The following properties under respective stanzas affect instance level behavior:
   - `[server]`

     Enable SSL connections—`https` = `yes`

     Disable plain http connections—`http` = `no`

     Ensure port number is the same as the back-end Integrity Lifecycle Manager server SSL port (mksis.secure.port)—`https-port` = `4443`
   - `[SSL]`

     ISAM certificate keyfile—`webseal-cert-keyfile` = `pdsrv.kdb`

     Stash file which contains the password user to protect the private keys in the keyfile—`webseal-cert-keyfile-stash` = `pdsrv.sth`

     Label of key to use certificate other than the default certificate in the key database—`webseal-cert-keyfile-label` = `integrity1_cert_label`

     Disable all the older SSL protocol versions and turn on only TLSv1.2. This is the most secure for client connections.

     `disable-ssl-v2` = `yes`

     `disable-ssl-v3` = `yes`

     `disable-tls-v11` = `yes`

     `disable-tls-v1` = `yes`

     `disable-tls-v12` = `no`

The following GSK attribute is required if you want to use TLSv1.2 protocol with the Integrity Lifecycle Manager client. This is because Windows MSC-API does not have crypto support for SHA224 with RSA algorithm, so exclude it:

```
gsk-attr-name = string:245:GSK_TLS_SIGALG_ECDSA_
WITH_SHA512,GSK_TLS_SIGALG_ECDSA_WITH_SHA384,GSK_
TLS_SIGALG_ECDSA_WITH_SHA256,GSK_TLS_SIGALG_ECDSA_
WITH_SHA224,GSK_TLS_SIGALG_ECDSA_WITH_SHA1,GSK_
TLS_SIGALG_RSA_WITH_SHA512,GSK_TLS_SIGALG_RSA_
WITH_SHA384,GSK_TLS_SIGALG_RSA_WITH_SHA256,GSK_
TLS_SIGALG_RSA_WITH_SHA1,GSK_TLS_SIGALG_RSA_WITH_
MD5
```

Note

In order to support TLSv1.2, the above-mentioned GSK attribute is required. If for some reason, this property has to be excluded, then you must enable TLSv1.1 and configure Integrity Lifecycle Manager clients need to use TLSv1.1. See section Configuring Integrity Lifecycle Manager client for details.

- `[ba]`

  Disable authentication using the Basic Authentication mechanism
  `ba-auth = none`
- `[forms]`

  Disable authentication using the forms authentication mechanism
  `forms-auth = none`
- `[certificate]`

  Configure the `accept-client-certs` property to either of the following values:

  - `critical`—You should always request for a client certificate. If a valid certificate is not presented, the SSL handshake fails.
  - `required`—You should always request for a client certificate. If a valid certificate is not presented, the SSL handshake will succeed and an error HTTP response will be sent back to the client.

    `accept-client-certs = critical`
- `[authentication-levels]`

Enable authentication strength levels for certificate authentication at the highest level:

- ○ `level = unauthenticated`
- ○ `level = ssl`

2. If more one than one back-end server is used with an ISAM instance, then you must configure 'interfaces' to listen to each these back-end servers. There will be one interface for one virtual host junction that corresponds to one back-end Integrity Lifecycle Manager server.

   For example, in the diagram mentioned in the Configuration Overview on page 9 section, a single ISAM instance is configured to serve two Integrity Lifecycle Manager servers. Hence, this configuration requires two interfaces, listening on two different ports. If a ISAM server is on a multi-homed machine, it is possible to use same ports with different IP addresses.

```
[interfaces]

interface1 = network-interface=10.192.60.1; https-
port=4444; certificate-label=integrity1_cert_label;
accept-client-certs=critical; worker-threads=default

interface2 = network-interface=10.192.60.1; https-
port=4445; certificate-label=integrity2_cert_label;
accept-client-certs=critical; worker-threads=default
```

   The following table lists the available properties and values that are used to configure a custom interface:

Valid properties and Values for Additional Interface Definitions

Property	Values	Description
http-port	• port number • `disabled` (default)	Port number to listen for HTTP requests on the specified network-interface. You can also set the value to disabled. You must specify either the http-port or https-port when you define an interface.
worker-threads	• count • default (`default`)	Number of worker threads that are used to process requests received only on this interface. You can use the default value to specify use of the worker thread pool that belongs to the default interface

Valid properties and Values for Additional Interface Definitions (continued)

Property	Values	Description
network-interface	• IP address • 0.0.0.0 (default)	IP address to listen for requests on the specified http-port or https-port. Both IPv4 and IPv6 formats are supported.
certificate-label	• key-file-label	Label name of a certificate in the pdsrv.kdb key database file. This is valid only when https-port is specified. The server-side certificate ISAM uses to authenticate to the client.
accept-client-certs	• `never` (default) • `required` • `optional` • `prompt_as_ needed` • `critical`	Specifies how ISAM should handle client-side certificates. Only valid when https-port is specified.

Configuring the ISAM Server for Integrity Lifecycle Manager Server Communication

ISAM server uses the concept of junctions to re-route authenticated requests to back-end Integrity Lifecycle Manager server(s). As this deployment supports only virtual host junctions, focus is limited to virtual host junction creation and configuration. The following steps explain properties and procedures required for virtual host junction creation and configuration in ISAM instance.

- Required Properties

  `[junction]`

  The maximum number of persistent connections which will be stored in the cache and used for Integrity Lifecycle Manager server communication.

  You can customize this configuration item for a particular junction by adding the adjusted configuration item to a `[junction:{jct-id}]` stanza, where `{jct-id}` refers to the junction point for a standard junction (including the leading'/'), or the virtual host label for a virtual host junction.

You must set the value to greater than 0 and ensure that it does not exceed worker-threads count.

`max-cached-persistent-connections = 50`

- Virtual Host Junctions

  Each Integrity Lifecycle Manager server requires a virtual host junction and you can have multiple junctions created on a single ISAM instance. You can create virtual host junction from the LMI page or its command line equivalent.

  To create a virtual junction from LMI page, perform the following steps:

  ○ On the LMI page, navigate to your ISAM instances page and select the ISAM instance for which you want to create the junction.

  ○ Click **Manage** ▸ **Junction Management**. The **Junction Management** page opens.

  ○ Click **New** ▸ **Virtual Junction**. The **Create a Virtual Junction** dialog opens.

  ○ Enter the following details for each of the tabs in the dialog:

    ◆ **Junction** Tab.

      An appropriate name for the junction label

      `Junction Label = vhost-junction-integrity1`

      The name of the DNS mapped virtual host as specified in section Configuration Overview on page 9.

      `Virtual Host = integrity1.ptc.com`

      The value of the virtual host port configured for respective interface as specified in step 2 of section Configuring the ISAM Server for Integrity Lifecycle Manager Client Authentication on page 26.

      `Virtual Host Port = 4444`

      Select the type of junction.

      `Junction Type = SSL`

    ◆ **Servers** Tab

      Click **New** to add the back-end Integrity Lifecycle Manager server details, which is the Integrity Lifecycle Manager server physical host name.

      `Hostname = host1.ptc.com`

The SSL port configured in `is.properties` file as `mksis.secure.port`.

`TCP or SSL Port = 4444`

- **Identity** Tab

  Select the `IV-USER` option. This is a mandatory option as the Integrity Lifecycle Manager server needs user name for authorization.

  `HTTP Header Identity Information = IV-USER`

  Optionally, you can send the end-user IP address to the Integrity Lifecycle Manager server by selecting the following option:

  `Insert client IP address`

To create virtual host junction from command line, connect to ISAM host with a terminal. Login as an administrator, and type the following command at the `pdadmin` prompt:

```
server task <ISAM instance name> virtualhost create -t ssl
-h host1.ptc.com -p 4444 -r -c iv-user -v
integrity1.ptc.com:4444 vhost-junction-integrity1.
```

For details on options used for the above command, refer the *IBM Security Access Manager for Web 8.X Command Reference* guide.

Configuring the Integrity Lifecycle Manager Client

Before you start with this configuration, ensure that Integrity Lifecycle Manager client is installed. For details on completing the installation, refer to the *Integrity Lifecycle Manager Help Center*.

Perform the following configuration steps:

1. In a text editor, open the following Integrity Lifecycle Manager client Preferences file to configure the required properties:

   `<home>/IntegrityClient.rc`

   where `<home>` is the home directory of the user.
2. Add the following properties at the end of file:

   This property notifies the client to use Windows store and `crypto` API for user certificate.

   `mks.IntegrityClient.sslProvider=mscapi`

   This property notifies the client to use HTTP as transport protocol for connections to ISAM server.

```
IntegrityClient.rmi.transport=http
```

With this property, the client forces version of TLS protocol to be 1.1 for SSL connections with ISAM server. You should use this property only if the GSK attribute mentioned in section Configuring the ISAM Server for Integrity Lifecycle Manager Client Authentication on page 26 is not configured and TLSv1.1 is enabled at ISAM.

```
IntegrityClient.tls.protocol=TLSv1.1
```

Note

If TLSv1.1 protocol is disabled on ISAM server by using the command `disable-tls-v11 = yes`, then connection will be refused by the server. PTC recommends that you use GSK attribute with TLSv1.2 as it is the latest version of TLS protocol and considered to be most secure.

3. During SSL negotiation, for the client to verify ISAM server certificate's identity, you must import the root CA certificate that is used for signing the ISAM server certificate.

   If the certificate is not present, import it into **Windows Trusted Root Certificates** by using `certmgr` utility. For details, refer https://technet.microsoft.com/en-in/library/cc754489.aspx.

Note

In a production environment, users are expected to use Smart Cards with digital certificates stored on them for authentication. However, for evaluation or testing purpose, you can have user certificates imported to Windows Personal Store and present it as client certificate during authentication with ISAM server.

Sometimes multiple certificates with the same alias may be installed in the system. When using smart cards, if multiple certificates are available, a certificate with the appropriate key usage is selected. For this selection, valid certificates are preferred over expired certificates. The selected certificate is the one that is not expired, and where the value of key usage is `All` or `Client Authentication`. If no right certificate is found, then the certificate selection fails, and a warning is logged in the client logs.

6

Certificate Selection for ISAM Server Authentication

In Integrity Lifecycle Manager 11.1, the client has been enhanced to display a **Certificate Selection** dialog box for ISAM server authentication. Using this dialog box, users can authenticate to a ISAM server with a desired client certificate if more than one eligible certificates are present in the their personal keystore. The behavior of Integrity Lifecycle Manager interfaces during this authentication is described in the following sections:

Integrity Lifecycle Manager Client GUI Behavior

- If a user's personal keystore has only one eligible certificate for client authentication, then that certificate is automatically selected for authentication and no dialog box is displayed for selecting a certificate. If the certificate is protected by a PIN, a prompt to enter the PIN may be displayed.
- If a user's personal keystore has more than one eligible certificate for client authentication, then the **Certificate Selection** dialog box is presented with a list of eligible certificates. The user can select any of the eligible certificates to authenticate with the ISAM server. A PIN prompt may be followed if the certificate is protected by PIN. If the PIN prompt is dismissed, and the user tries to connect to the server, then the **Certificate Selection** dialog box is not displayed as the previous selection is cached.
- If a user's personal keystore has no eligible certificates, or the user dismisses the **Certificate Selection** dialog box, authentication fails, and a handshake failure error message appears.

Note

The **Certificate Selection** dialog box appears only once during the Integrity Lifecycle Manager client process lifecycle. If the user wants to switch the identity (authenticate with a different user certificate), then the client has to be restarted.

Integrity Lifecycle Manager client uses the following list of rules to decide whether a certificate is eligible for client authentication. The **Certificate Selection** dialog box displays only those certificates that fulfill the following rules.:

1. Certificates with current validity
2. Certificates with valid trust chain
3. Certificates out of the Windows keystore (WINDOWS-MY)
4. Certificates for which a private key exists
5. Certificates with the value `digitalsignature` in the `key usage` field (if the field/extension is present)
6. Certificates with OID Smart Card Logon `(1.3.6.1.4.1.311.20.2.2`) or client authentication (`1.3.6.1.5.5.7.3.2`), or any OID (`2.5.29.37.0`) in the `extended key usage` field (if the field/extension is present)
7. Certificates that fulfill requested restrictions (accepted CAs) if requested by the server (`CertificateRequest-message`)

Integrity Lifecycle Manager CLI and Client API Integration Behavior

Both Integrity Lifecycle Manager CLI and Client Integration API interfaces use locally installed Integrity Lifecycle Manager client instance to communicate with the server. Hence, they exhibit the same behavior with respect to **Certificate Selection** dialog box as mentioned in the above section. However, both the interfaces have been enhanced to enforce the `-g` option (GUI) if the command is invoked on an unauthenticated client instance. If a user issues a command that requires server communication, and if the client is not yet authenticated, then the following scenarios need to be considered:

- A user should run the command with a `-g` option. If this option is given, then the **Certificate Selection** dialog box is displayed as mentioned in the above section.
- If the command is not run with the`-g` option, an error message instructing to specify the option is displayed.

After a user is authenticated successfully, successive invocations of the same command, or any other command that requires server communication does not require the `-g` option as long as the client instance is not restarted.

Note

Since not all commands support the `-g` option, it is recommended to stagger the command invocations accordingly for non-interactive use cases to avoid errors. For example, a user can use the `im connect` command with the `-g` option to perform authentication. After the authentication is successful, any other command can be invoked without the `-g` option as long as the client instance is not restarted.

Integrity Lifecycle Manager Gateway Behavior

In most cases, Gateway uses locally installed Integrity Lifecycle Manager client instance to communicate with the server. Hence, if a user is already authenticated with the server using a running client instance, then Gateway does not require additional authentication. However, if Gateway commands involve remote resources (such as remote template configuration or images) hosted on a ISAM protected server, then an additional authentication is required. The **Certificate Selection** dialog box and PIN prompt behavior is same as described in the above section. Any additional authentication performed is effective only for the period of command execution.

The **Certificate Selection** dialog box may appear during any of the following Gateway activities. The dialog box appears only once during the particular command execution:

- Gateway `export` or `import` command with remote template configuration hosted on a ISAM protected server.
- Gateway `export` or `import` of a document containing images hosted on ISAM protected server.

The Gateway standalone command line tool has also been enhanced for ISAM authentication. All the Gateway commands accept two new options to instruct gateway to perform additional ISAM authentication. The options are `sslProvider` and `tlsVersion`.

For example, Gateway tool users can now issue an `export` command as follows:

```
$gateway export --sslProvider=mscapi --tlsVersion=TLSv1.1
<document_id>
```

where:

- `sslProvider` is mandatory and takes `mscapi` as value. If omitted, the export operation may fail if it involves remotely hosted resources on a ISAM protected server.
- `tlsVersion` is optional, and the allowed values are `TLSv1`, `TLSv1.1`, or `TLSv1.2`. If omitted, it defaults to `TLSv1.2` protocol.

Note

JRE provides the `httpsprotocols` property to control TLS protocol version. If this property is present in `Gateway.lax` file, it overrides any user-configured protocol version.

7

Enabling Mutual TLS Authentication

ISAM server intercepts every request intended for the Integrity Lifecycle Manager server and authenticates the end user by prompting for digital certificate. In case of two-factor authentication, Windows also prompts for PIN to access the smart card's certificate. On successful authentication, requests are forwarded to Integrity Lifecycle Manager server with user identity headers. As the requests are already authenticated by ISAM server, Integrity Lifecycle Manager does not re-authenticate incoming requests when configured for External Authentication security scheme. Such a trust establishment using mere HTTP headers is prone to serious attacks. Any adversary with a know-how of these identity headers can carry out impersonation attacks on Integrity Lifecycle Manager server.

To thwart any such attempts and to safeguard Integrity Lifecycle Manager server against any kind of attacks, the server is enhanced to support mutually authenticated TLS connections. Once the mutual TLS feature is enabled, the server can accept connections from only pre-configured peers. Connection attempts from sources such as adversaries or any other source will be terminated at the TLS protocol level. As a result connections cannot be established under any circumstances.

Note

Though enabling mutual TLS is an optional feature, if Integrity Lifecycle Manager server is configured on External Authentication scheme in the production environment, ensure that you always enable TLS. Not doing so, leaves the server open to impersonation attacks.

Establishing mutually authenticated TLS connections between ISAM server and Integrity Lifecycle Manager server involves:

- Configuring the server for mutual authentication
- Creating mutual authentication junctions at ISAM instance

Configuring the Integrity Lifecycle Manager Server for Mutual Authentication

Perform the following steps to configure Integrity Lifecycle Manager server for mutual authentication:

1. Import the root CA certificate into a separate Integrity Lifecycle Manager trust-store. This certificate is used for signing ISAM server certificate

   To verify the digital signature of certificate presented by ISAM during TLS handshake, Integrity Lifecycle Manager server requires CA certificate that was used to sign the ISAM server certificate. Acquire a root CA certificate used for signing ISAM server certificate and execute the following `keytool` command to import it into a new trust store.

   ```
   keytool -importcert -alias <ISAM_cert_alias> -file
   <root_CA_of_ISAM_server_cert> -keystore
   <installDir>/data/tls/clientcacerts
   ```

   where:

   `ISAM_cert_alias` is the alias used for identifying certificate entry.

   `root_CA_of_ISAM_server_cert` is the root CA certificate used to sign ISAM server certificate.

   `installDir` is the Integrity Lifecycle Manager server installation directory.

   The above command creates a new trust store called `clientcacerts` and imports the root CA certificate used for signing ISAM server certificate.

   Note

   The `keytool importcert` command contains the option –`trustcacerts`. If you specify the `-trustcacerts` option , additional certificates are considered for the chain of trust, namely the certificates in a file named `cacerts`. Hence you must not use this option with the above command.

2. Enable mutual TLS authentication and configure client DN(s) for enhanced security.

   By default, mutual authentication is disabled. To enable mutual authentication, you should set the `mksis.secure.clientauth` property to `required` in the Integrity Lifecycle Manager server (is.properties) properties file.

   Set the following property to `required` to enable mutual TLS authentication. The allowed values are:

- `never`—Default value with mutual authentication disabled.
- `required`—Enables mutual authentication.

```
mksis.secure.clientauth=required
```

Note

If mutual authentication is enabled, name the file configured in step A to `clientcacerts` and place it in `data/tls` folder. If this is not done, the server fails to start.

For enhanced security, you can also specify the subject's distinguished name (DN) in the ISAM server certificate. In addition to verifying ISAM server certificate's signature, this enables Integrity Lifecycle Manager server to verify the identity of the certificate presenter by comparing it with a pre-configured DN name. Though this is an optional feature, it is recommended to configure if for enhanced security. If not configured, checks for DN verification are not carried out.

To, configure the subject DN specified in the signed ISAM server certificate, or the one mentioned with –K option during junction creation, refer topic Creating Mutually Authenticated SSL Junctions in ISAM.

The accepted string formats are defined by IETF RFC 1779 and IETF RFC 2253.

```
mksis.secure.clientauth.dn=CN=
integrity1.ptc.com,OU=ptcnet,O=ptc,C=us
```

Note

The above property allows multiple values in enumerated fashion, which is not applicable for the deployment use case provided in this guide. However, if a combination of valid and invalid DN values are used, the server considers only valid values and warnings are logged for incorrectly specified values.

Creating Mutually Authenticated SSL Junctions in ISAM

ISAM supports mutual authentication between an ISAM server and a back-end Integrity Lifecycle Manager server over an SSL junction (–t ssl).

Before you create a mutual SSL junction, ensure the following:

- The Integrity Lifecycle Manager server root CA certificate is imported into the ISAM key database as mentioned in the topic Common Configuration on page 23
- The Integrity Lifecycle Manager server is configured for mutual authentication as mentioned in topic Configuring the Integrity Lifecycle Manager Server for Mutual Authentication on page 39.
- Integrity Lifecycle Manager server is running before you start creating junctions. This avoids warnings during the junction creation process.

You can create mutually authenticated SSL junctions for a backend server either through the LMI page or by using CLI commands as described in the following steps:

1. Perform the following steps on the LMI Page:
    - Click **Secure ▸ Reverse Proxy**and select an instance.
    - Click **Manage ▸ Junction Management**.
    - Click **New ▸ Virtual Junction** in the Junction Management page that appears.

      The **Create a Virtual Junction** dialog opens. Add all the details mentioned in topic Configuring the ISAM Server for Integrity Lifecycle Manager Server Communication on page 29 along with the following additional details:

        - **Servers** Tab

          Click **New**. In the **Add TCP or SSL Servers** dialog that opens, specify the exact DN in Integrity Lifecycle Manager server certificate's subject field.

          During server-side certificate verification, the DN contained in the certificate is compared with the DN defined by the junction. The connection to the Integrity Lifecycle Manager server fails if the two DNs do not match.

          ```
          Distinguished Name (DN) = CN=host1.ptc.com,OU=
          ptcnet,O=ptc,C=us
          ```
        - **Basic Authentication** Tab

          To enable mutual authentication, select **Enable mutual authentication to junctioned WebSEAL servers**.

          From the **Key Label** list, select the ISAM server certificate label `integrity1_cert_label`. This is the label that you want to present to Integrity Lifecycle Manager server during SSL handshake.

2. Perform the following step from the command line.

   Connect to ISAM host with a terminal, login as administrator, then type in the following command:

   ```
   pdadmin sec_master> server task <Your ISAM instance name>
   create -t ssl -h host1.ptc.com -p 4444 -r -c iv-user -K
   "integrity1_cert_label" -D "CN=host1.ptc.com,OU=
   ptcnet,O=ptc,C=us" -v integrity1.ptc.com:4444 vhost-
   junction-integrity1
   ```

   where —`K` specifies the ISAM certificate to be presented during SSL handshake.

   and

   `-D` specifies Distinguished Name to be used for verification of Integrity Lifecycle Manager server certificate.

The following outline summarizes the supported functionality for mutual authentication over SSL:

1. ISAM authenticates the back-end server (normal SSL process)

   - ISAM validates the server certificate from the back-end Integrity Lifecycle Manager server. See “Validation of the back-end server certificate” section of *IBM Security Access Manager for Web 8.X Web Reverse Proxy Configuration topics* guide.

     See “Matching the distinguished name (DN)” section of *IBM Security Access Manager for Web 8.X Web Reverse Proxy Configuration topics* guide.

2. Back-end Integrity Lifecycle Manager server authenticates ISAM

   - The back-end Integrity Lifecycle Manager server validates client certificate from ISAM using –`K` option. See "Authentication with a client certificate" section of *IBM Security Access Manager for Web 8.X Web Reverse Proxy Configuration topics* guide.

8

Troubleshooting

In this deployment, the Integrity Lifecycle Manager server does not require any additional configuration steps for troubleshooting the problems. Refer “Server Troubleshooting” chapter of *Integrity Lifecycle Manager Installation and Upgrading Guide* for available troubleshooting options.

A quick reference on important troubleshooting options is provided in the following sections.

Troubleshooting SSL Related Issues

If you notice SSL or certificate-related errors during the initial setup, you can configure both the Integrity Lifecycle Manager client and server to log the entire request-response exchange flowing through SSL protocol.

To enable SSL logging at client side, append the following logging statement to `lax.nl.java.option.additional` property in `<serverInstallDir>/bin/IntegrityClient.lax` file.

```
-Djavax.net.debug=ssl
```

To enable SSL logging at server side, add the same statement in `<serverInstallDir>/config/mksservice.conf` file as an additional option to server JVM.

`mks.java.additional.<count>=-Djavax.net.debug=ssl` where count is the additional parameter number.

See the article http://docs.oracle.com/javase/8/docs/technotes/guides/security/jsse/ReadDebug.html to understand and analyze the logged statements.

Note

The above logging statements generate huge amount of verbose logging on both the client and server side and can potentially impact the performance. Hence, you should use them only for troubleshooting purpose but not in the production environment.

Troubleshooting ISAM Related Issues

Sometimes you may face problems in deployments involving different types of clients, reverse proxy, LDAP, and back-end servers. Such problems can get worse if SSL protocol is in action. In such cases, ISAM offers excellent troubleshooting options such request-response logging, event logging, Web Interface to monitor http(s) traffic, and so on.

One of the most important troubleshooting options available on ISAM instance is `tracing`. To enable this option, navigate to the **Reverse Proxy** list on LMI page and then go to **Manage** ▸ **Troubleshooting** ▸ **Tracing**. Select the required component and click **Edit**. Set the level to 9 and change the rollover size if you wish to have bigger or smaller size files than the default 2 MB file size.

The following tracing components are frequently used:

- `pdweb.debug`—This component traces the HTTP headers for requests and responses.
- `pdweb.snoop`—This component traces HTTP traffic. It also logs the HTTP headers and the message body for requests and responses.
- `pdweb.wan.ssl`—This component is used to trace the SSL connection between ISAM and junctioned web servers.
- `pdweb.wns.authn`—This component is used to trace the authentication processing.

For more information on other troubleshooting topics, refer *IBM Security Access Manager for Web 8.X Troubleshooting Topics* guide.

Note

The amount of data that is produced by the trace options, especially by the `snoop trace` command can be large. Additionally, the trace might log sensitive information in plain text.

Troubleshooting Network Related Issues

To analyze the network traffic between Integrity Lifecycle Manager client and ISAM server, or between ISAM Server and Integrity Lifecycle Manager server, you can use Wireshark tool. Wireshark is a free and popular open source packet analyzer used for troubleshooting network related issues. See https://www.wireshark.org/ to download and install Wireshark.

You can install Wireshark on both client and server machines to capture end to end network traffic from Integrity Lifecycle Manager client to ISAM server, and to Integrity Lifecycle Manager server. With capture filters, you can capture only interested packets, such as packets between particular IP addresses or on a particular TCP/SSL port. See https://wiki.wireshark.org/CaptureFilters for details.