How to Configure Security for a Web Service
Abstract

You can configure security for web services that run on the Data Integration Service. This article describes the security options available for a web service and solutions to common web service security requirements. It also includes the steps to configure each security option.

Supported Versions

- Data Services 9.1.0

Table of Contents

Web Services Security Overview ......................................................  2
Web Service Security Solutions ......................................................  3
Message Layer Security Configuration ..................................................  3
  User Name Token Requirements ....................................................  4
  Web Service Permissions ........................................................  6
  Enabling WS-Security for a Web Service ...............................................  6
  Assigning Permissions on a Web Service ..............................................  6
Transport Layer Security Configuration ..................................................  7
  Configuring the Data Integration Service to use TLS .......................................  7
  Configuring the Web Service to use TLS ...............................................  8
Pass-Through Security Configuration ...................................................  8
  Adding Pass-Through Security .....................................................  8
HTTP Client Filter Configuration ......................................................  9
  Configuring the Data Integration Service to use a HTTP Client Filter .............................  9

Web Services Security Overview

Informatica Data Services provides data integration functionality through a web service. The Data Integration Service processes web service requests. After you deploy a web service to a Data Integration Service, you can configure web service security.

You can configure the following security options for a web service:

HTTP Client Filter

If you want the Data Integration Service to accept requests based on the host name or IP address of the web service client, use the Administrator tool to configure an HTTP client filter. By default, a web service client running on any machine can send requests.

Message Layer Security

If you want the Data Integration Service to authenticate user credentials in SOAP requests, use the Administrator tool to enable WS-Security and configure web service permissions. The Data Integration Service can validate user credentials that are provided as a user name token in the SOAP request. If the user name token is not valid, the Data Integration Service rejects the request and sends a system-defined fault to the web service client. If a user does not have permission to execute the web service operation, the Data Integration Service rejects the request and sends a system-defined fault to the web service client.
Transport Layer Security (TLS)

If you want the web service and web service client to communicate using an HTTPS URL, use the Administrator tool to enable transport layer security (TLS) for a web service. The Data Integration Service that the web service runs on must also use TLS. An HTTPS URL uses SSL to provide a secure connection for data transfer between a web service and a web service client.

Pass-Through Security

If an operation mapping requires connection credentials, the Data Integration Service can pass credentials from the user name token in the SOAP request to the connection. To configure the Data Integration Service to pass credentials to a connection, use the Administrator tool to configure the Data Integration Service to use pass-through security for the connection and enable WS-Security for the web service.

Note: You cannot use pass-through security when the user name token includes a hashed or digested password.

Web Service Security Solutions

Based on the security requirements, you may want to configure one or more security options.

The following table describes security scenarios and a solution for the requirements of each scenario:

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Solution</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Hypostores customer service representatives want a web service that can access customer details based on the customer ID. The corporate policy requires that data accessed over a network must be secure and authenticated. | - Transport Layer Security  
- Message Layer Security | Transport layer security (TLS) provides secure communication between the web service and the web service client. Message layer security authenticates each web service request and ensure that the user has permission to run the web service. |
| Certain members of the Hypostores finance department require a web service that can access company revenue details. The corporate policy requires that the information should only be accessible to employees that have permission to run the web service and permission to access the database that contains the data. | - Message Layer Security  
- Pass-Through Security | Message layer security authenticates each web service request and ensure that the user has permissions to run the web service. Pass-through security enables the web service operation mapping to connect to a source using the credentials provided in the web service SOAP request. Message layer security and pass-through security together ensure that the user who submits the request has permission to run the web service and access the source data that the web service processes. |
| Hypostores has a two person order processing team that requires access to the latest order data. The corporate policy requires that the information should only be accessible to employees that have permission to run the web service. | - HTTP Client Filter | The HTTP client filter controls which machines can access web services that run on the Data Integration Service. You can specify the host names or the IP addresses of each user that has permission to run the web service. |

Message Layer Security Configuration

If you want the Data Integration Service to authenticate user credentials in SOAP requests, use the Administrator tool to enable WS-Security and configure web service permissions.

When you enable WS-Security, the Data Integration Service uses the credentials provided in the user name token to authenticate web service requests. The Data Integration Service processes the request when the user name token is valid and the user has execute permissions on the web service.
**User Name Token Requirements**

Web service clients must include a user name token header in the SOAP request when a web service requires user authentication. When a web service does not require user authentication, the Data Integration Service ignores the user name token header provided in the SOAP request.

The user name token element in a SOAP request can have one of the following password types:

- Plain text
- Hashed
- Digested

**Plain Text Password**

Include a plain text password in the user name token header of a SOAP request when the user password does not need to be encrypted. The Data Integration Service can process plain text passwords in the UsernameToken element.

When the password is plain text, the UsernameToken element includes the following child elements:

**Username element**

Contains a user name in the Native security domain or any LDAP security domain. The default security domain is the Native security domain. If the user name belongs to the Native security domain, the Username element does not require the name of the security domain. If the user name belongs to an LDAP security domain, the user name must be preceded by the name of the security domain and a slash (/).

**Password element**

Contains the password in plain text. Set the Type attribute of the Password element to "PasswordText."

The following sample SOAP header shows an example of a UsernameToken element with a plain text password:

```xml
<soap:Header>
  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
    <wsse:UsernameToken wsu:Id="UsernameToken-14" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:Username>Administrator</wsse:Username>
      <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">Administrator</wsse:Password>
    </wsse:UsernameToken>
  </wsse:Security>
</soap:Header>
```

**Hashed Password**

Include a hashed password in the user name token header of a SOAP request when the user password must be encrypted. The Data Integration Service can process hashed passwords in the UsernameToken element.

When you use a hashed password, the UsernameToken element includes the following child elements:

**Username element**

Contains a user name in the Native security domain.

**Password element**

Contains a hashed password. The password must be hashed with the MD5 or SHA-1 hash function and encoded to Base64. Set the Type attribute of the Password element to "PasswordText."

The following sample SOAP header shows an example of a UsernameToken element with a hashed password:

```xml
<soap:Header>
  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
    <wsse:UsernameToken wsu:Id="UsernameToken-14" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:Username>Administrator</wsse:Username>
      <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">AmbRoK1UYDdsBUg19Lew</wsse:Password>
    </wsse:UsernameToken>
  </wsse:Security>
</soap:Header>
```
Digested Password

Include a digested password in the user name token header of a SOAP request when the user password is an encrypted password that is hashed with a nonce value and a time stamp. The Data Integration Service can process digested passwords in the UsernameToken element.

When you use a digested password, the UsernameToken element includes the following child elements:

Username element

Contains a user name that can be found in the Native security domain.

Password element

Contains a digested password. The password is the value generated from hashing the password concatenated with the nonce value of the Nonce element and the time stamp in the Created element. The password must be hashed with the SHA-1 hash function and encoded to Base64. For digested password security, set the Type attribute of the Password element to "PasswordDigest."

Nonce element

Contains a nonce value, which is a random value that can be used only once. By default, it is valid for 300 seconds after the time that the request is created, as indicated by the value in the Created element. The client application must send the request within the time that the nonce value is valid. For example, the Created value indicates that the request was created at 10:00 a.m. The request is valid from 10:00 a.m. to 10:05 a.m. If the client application sends the request to the web service before 10:00 a.m. or after 10:05 a.m., then the request and the nonce value are not valid and the request will fail.

Created element

Contains a time stamp value that indicates the time when the request was created. The time stamp uses the UTC format, yyyy-MM-dd'T'HH:mm:ss.SSS'Z'. For example: 2008-08-11T18:06:32.425Z.

The digested password uses the standard OASIS password digest algorithm:

Password_Digest = Base64 ( SHA-1 ( nonce + created + password ) )

You can use any tool to generate the nonce value, time stamp, and the digested password.

The following sample SOAP header shows an example of a UsernameToken element with a digested password:

```xml
<soap:Header>
  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
    <wsse:UsernameToken wsu:Id="UsernameToken-14" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:Username>Administrator</wsse:Username>
      <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordDigest">Ntm58Cxf7SBOqA30lsTv1nD7</wsse:Password>
      <wsse:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">zWELHdoAzNjQQ9xzlIwFZA==</wsse:Nonce>
      <wsu:Created>2010-10-15T20:56:18.633Z</wsu:Created>
    </wsse:UsernameToken>
  </wsse:Security>
</soap:Header>
```
**Web Service Permissions**

End users can send web service requests and receive web service responses through a web service client. Permissions control the level of access that a user has to a web service.

You can assign permissions to users and groups on the following web service objects:

- Web service
- Web service operation

When you assign permissions on a web service object, the user or group inherits the same permissions on all objects that belong to the web service object. For example, you assign a user execute permission on a web service. The user inherits execute permission on web service operations in the web service.

You can deny permissions to users and groups on a web service operation. When you deny permissions, you configure exceptions to the permissions that users and groups might already have. For example, a user has execute permissions on a web service which has three operations. You can deny a user from running one web service operation that belongs to the web service.

**Types of Web Service Permissions**

You can assign the following permissions to users and groups:

- Grant permission. Users can manage permissions on the web service objects using the Administrator tool or using the `infacmd` command line program.
- Execute permission. Users can send web service requests and receive web service responses.

The following table describes the permissions for each web service object:

<table>
<thead>
<tr>
<th>Object</th>
<th>Grant Permission</th>
<th>Execute Permission</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web service</td>
<td>Grant and revoke permission on the web service and all web service operations within the web service.</td>
<td>Send web service requests and receive web service responses from all web service operations within the web service.</td>
</tr>
<tr>
<td>Web service operation</td>
<td>Grant, revoke, and deny permission on the web service operation.</td>
<td>Send web service requests and receive web service responses from the web service operation.</td>
</tr>
</tbody>
</table>

**Enabling WS-Security for a Web Service**

Configure the web service to use WS-Security when you want the web service to authenticate user credentials in the SOAP request.

1. In the Administrator tool, select a Data Integration Service.
2. Click the Application view.
3. Expand the application name in the top panel and select the web service.

**Assigning Permissions on a Web Service**

When you assign permissions on a web service object, you define the level of access a user or group has to the object.

1. On the Domain tab, select the Services and Nodes view.
2. In the Navigator, select a Data Integration Service.
3. In the contents panel, select the Applications view.
4. Select the web service object.
5. In the details panel, select the Group Permissions or User Permissions view.
6. Click the Assign Permission button.
   The Assign Permissions dialog box displays all users or groups that do not have permission on the SQL data service object.
7. Enter the filter conditions to search for users and groups, and click the Filter button.
8. Select a user or group, and click Next.
9. Select Allow for each permission type that you want to assign.
10. Click Finish.

**Transport Layer Security Configuration**

Configure TLS for a web service so that web service request to the Data Integration Service uses the SSL security protocol. When you configure TLS for a web service, the web service client connects to the web service with a HTTPS URL.

To configure TLS for a web service, enable TLS for the web service and the Data Integration Service that runs the web service. When you enable TLS for the Data Integration Service and enable TLS for the web service, the web service uses a HTTPS URL. When you enable TLS for the Data Integration Service and do not enable TLS for the web service, the web service can use the HTTP URL or a HTTPS URL. If you enable TLS for a web service without enabling TLS for the Data Integration Service, the web service will not start.

**Configuring the Data Integration Service to use TLS**

Configure the Data Integration Service to use TLS when you want to use TLS for web services that run on the Data Integration Service.

1. In the Administrator tool, select a Data Integration Service.
2. Click the Processes view.
3. Click Edit in the Data Integration Service Security section.
   The Edit Data Integration Service dialog box appears.
4. Enable transport layer security.
5. Edit the following the Data Integration Service security properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP Port</td>
<td>Unique HTTP port number for the Data Integration Service.</td>
</tr>
<tr>
<td>HTTPS Port</td>
<td>HTTPS port number for the Data Integration Service when you enable the TLS protocol. Use a different port number than the HTTP port number.</td>
</tr>
</tbody>
</table>
6. Edit the following HTTP configuration properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Concurrent Requests</td>
<td>Maximum number of HTTP or HTTPS connections that can be made to this Data Integration Service process. Default is 200.</td>
</tr>
<tr>
<td>Maximum Backlog Requests</td>
<td>Maximum number of HTTP or HTTPS connections that can wait in a queue for this Data Integration Service process. Default is 100.</td>
</tr>
<tr>
<td>Keystore File</td>
<td>Path and file name of the keystore file that contains the keys and certificates required if you enable TLS and use HTTPS connections for the Data Integration Service. You can create a keystore file with a keytool. keytool is a utility that generates and stores private or public key pairs and associated certificates in a keystore file. You can use the self-signed certificate or use a certificate signed by a certificate authority.</td>
</tr>
<tr>
<td>Keystore Password</td>
<td>Password for the keystore file.</td>
</tr>
<tr>
<td>Truststore File</td>
<td>Path and file name of the truststore file that contains authentication certificates trusted by the Data Integration Service.</td>
</tr>
<tr>
<td>Truststore Password</td>
<td>Password for the truststore file.</td>
</tr>
<tr>
<td>SSL Protocol</td>
<td>Secure Sockets Layer protocol to use. Default is TLS.</td>
</tr>
</tbody>
</table>

**Configuring the Web Service to use TLS**

Enable TLS security for a web service to require the web service to use the SSL security protocol.

When you enable TLS for a web service you must also enable TLS for the Data Integration Service that runs the web service.

1. In the Administrator tool, select a Data Integration Service.
2. Click the **Application** view.
3. Expand the application name in the top panel and select the web service.
4. In the **Properties** view, enable transport layer security.

**Pass-Through Security Configuration**

Configure pass-through security and message layer security when you want the web service operation mapping to connect to a source using the user name and password provided in the web service SOAP request.

Pass-through security is the capability to connect to an SQL data service or an external source with the client user credentials instead of the credentials from a connection object.

A web service operation mapping might need to use a connection object to access data. If you configure pass-through security and the web service uses WS-Security, the web service operation mapping connects to a source using the user name and password provided in the web service SOAP request.

Configure pass-through security in the Data Integration Service properties. When you configure pass-through security, you define the connections that allow pass-through security. You can configure the list of connections in the Administrator tool or with infacmd dis UpdateServiceOptions.

**Adding Pass-Through Security**

Select the connections that use pass-through security.
1. In the Administrator tool, select a Data Integration Service.
2. Click the Properties view.
3. Edit the pass-through security options.
   The Edit Pass-through Security Options dialog box appears.
4. To choose pass-through connections, click Select. You can select multiple connections at a time.
5. Select Allow Caching to allow data object caching for the SQL data services and web services that use the connections.
6. Click OK.

You must recycle the Data Integration Service to enable caching for the connections.

### HTTP Client Filter Configuration

Configure HTTP client filter properties to specify web service client machines that can send requests to the Data Integration Service. By default, a web service client running on any machine can send requests.

To specify machines that can send web service request to a Data Integration Service, configure the HTTP client filter properties in the Data Integration Service properties. When you configure these properties, the Data Integration Service compares the IP address or host name of machines that submit web service requests against these properties. The Data Integration Service either allows the request to continue or refuses to process the request.

You can use constants or Java regular expressions as values for these properties. You can include a period (.) as a wildcard character in a value.

**Note:** You can allow or deny requests from a web service client that runs on the same machine as the Data Integration Service. Enter the host name of the Data Integration Service machine in the allowed or denied host names property.

**Example**
The Finance department wants to configure a web service to accept web service requests from a range of IP addresses. To configure the Data Integration Service to accept web service requests from machines in a local network, enter the following expression as an allowed IP address:

```
"192.168.1.\[0-9]\*"
```

The Data Integration Service accepts requests from machines with IP addresses that match this pattern. The Data Integration Service refuses to process requests from machines with IP addresses that do not match this pattern.

### Configuring the Data Integration Service to use a HTTP Client Filter

Configure a HTTP client filter when you configure the Data Integration Service properties.

1. In the Administrator tool, select a Data Integration Service.
2. Click the Properties view.
3. Edit the following HTTP client filter properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allowed IP Addresses</td>
<td>List of constants or Java regular expression patterns compared to the IP address of the requesting machine. Use a space to separate multiple constants or expressions. If you configure this property, the Data Integration Service accepts requests from IP addresses that match the allowed address pattern. If you do not configure this</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>property, the Data Integration Service uses the Denied IP Addresses property to determine which clients can send requests.</td>
<td></td>
</tr>
<tr>
<td>Allowed Host Names</td>
<td>List of constants or Java regular expression patterns compared to the host name of the requesting machine. The host names are case sensitive. Use a space to separate multiple constants or expressions. If you configure this property, the Data Integration Service accepts requests from host names that match the allowed host name pattern. If you do not configure this property, the Data Integration Service uses the Denied Host Names property to determine which clients can send requests.</td>
</tr>
<tr>
<td>Denied IP Addresses</td>
<td>List of constants or Java regular expression patterns compared to the IP address of the requesting machine. Use a space to separate multiple constants or expressions. If you configure this property, the Data Integration Service accepts requests from IP addresses that do not match the denied IP address pattern. If you do not configure this property, the Data Integration Service uses the Allowed IP Addresses property to determine which clients can send requests.</td>
</tr>
<tr>
<td>Denied Host Names</td>
<td>List of constants or Java regular expression patterns compared to the host name of the requesting machine. The host names are case sensitive. Use a space to separate multiple constants or expressions. If you configure this property, the Data Integration Service accepts requests from host names that do not match the denied host name pattern. If you do not configure this property, the Data Integration Service uses the Allowed Host Names property to determine which clients can send requests.</td>
</tr>
</tbody>
</table>

**Author**

Padma Heid  
Lead Technical Writer

**Acknowledgements**

The author would like to thank Sumeet Kumar Agrawal for his technical assistance.