PingFederate®

Kerberos Token Translator

Version 2.0.1

User Guide

Ping Identity®

PingFederate Zendesk Connector Quick Connection Guide
Version 1.0
December, 2014

Ping Identity Corporation
1001 17th Street, Suite 100
Denver, CO 80202
U.S.A.

Phone: 877.898.2905 (+1 303.468.2882 outside North America)
Fax: 303.468.2909
Web Site: www.pingidentity.com

Trademarks

Disclaimer

Document Lifetime

Ping Identity may occasionally update online documentation between releases of the related software. Consequently, if this PDF was not downloaded recently, it may not contain the most up-to-date information. Please refer to documentation.pingidentity.com for the most current information.

From the Web site, you may also download and refresh this PDF if it has been updated, as indicated by a change in this date: **December 18, 2014**.

Contents

Introduction

The PingFederate Kerberos Token Translator provides an Identity Provider (IdP) Token Processor for the PingFederate WS-Trust Security Token Service (STS). The Token Processor allows the STS to accept and validate Kerberos tokens—via a configured Kerberos Realm—from a Web Service Client (WSC) and then map user attributes into a SAML token for the WSC to send to a Web Service Provider (WSP). For more information on Kerberos Realms, see Using AD Domains and Kerberos Realms in the PingFederate *Administrator's Manual*.

Note: Ping Identity provides a Java WS-Trust Client Software Development Kit (SDK) for enabling Web Service applications (Client or Provider) to interact with the PingFederate STS. The SDK is available on the Downloads page (`www.pingidentity.com/en/products/downloads`).

System Requirements

- PingFederate 6.8 or higher

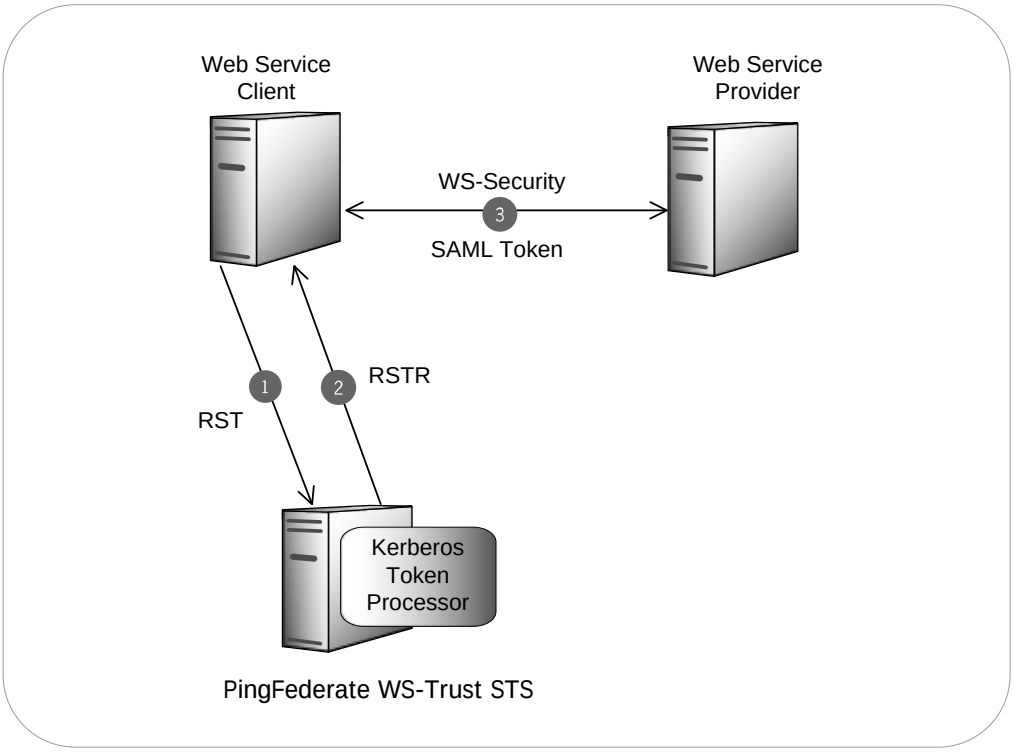
ZIP Manifest

The distribution ZIP file for the Kerberos Token Translator contains the following:

- `ReadMeFirst.pdf` – contains links to this online documentation
- `/dist` – contains libraries needed to run the Token Processor:
  - `pf-kerberos-token-translator-2.0.1.jar` – the Kerberos Token Processor `JAR` file

Process Overview

The following figure shows a basic WS-Trust STS scenario in which PingFederate validates a Kerberos token and issues a SAML token:



Processing Steps

1. A WSC sends a Request Security Token (RST) message containing a Kerberos token as a SOAP request to the PingFederate STS IdP endpoint.
2. PingFederate validates the Kerberos token against the Domain Controller/Key Distribution Center (KDC). If the token is valid, PingFederate issues a SAML token with attributes mapped from the Kerberos token. Next, PingFederate embeds this SAML token into a Request Security Token Response (RSTR). The RSTR token is returned to the WSC as a SOAP response.
3. The WSC binds the issued SAML token into a Web Service Security (WSSE) header and sends a SOAP request to the Web Service Provider (WSP).

Configuration and Processor Setup

This section describes how to configure the Active Directory (AD) domain account, PingFederate access to the domain account, and install and configure the Kerberos Token Processor.

Step One – Configure the AD Domain

1. Create a new domain user account that PingFederate can use to contact the domain controller.

   Note: You must have Domain Administrator permissions.

   Be sure to select **Password never expires**.

   Remember these credentials; you will need them in the next section (see Step Two – Configure PingFederate Access).

2. Register Service Principal Name (SPN) properties using the `setspn` Windows utility by executing the following command:

   ```
   setspn –A HTTP/<pf.domain.name> <domain-account>
   ```

   where:

   - `<pf.domain.name>` is the fully-qualified domain name of the PingFederate STS server.
   - `<domain-account>` is the domain account you created in step 1.

   The `setspn` utility is distributed with Windows Server support tools. The support tools are located on the Windows installation disk or can be downloaded from Microsoft (`www.microsoft.com/download/en/details.aspx?displaylang=en&id=7911`).

Step Two – Configure PingFederate Access

Click Active Directory Domains/Kerberos Realms on the PingFederate Main Menu to configure access for PingFederate to the domain user account you created in the last section (see Step One – Configure the AD Domain). Ping Federate contacts the domain controller(s) or KDC(s) for verifying user authentication (see Using AD Domains and Kerberos Realms in the PingFederate *Administrator's Manual*).

Step Three – Install and Configure the Processor

1. Copy the `pf-kerberos-token-translator-2.0.1.jar` file from the Kerberos Token Processor kit `dist` directory to the `<pf-install>/pingfederate/server/default/deploy` directory of your PingFederate server installation.
2. Start or restart PingFederate.
3. Log on to the PingFederate administrative console and click **Token Processors** under Application Integration Settings in the My IdP Configuration section of the Main Menu.

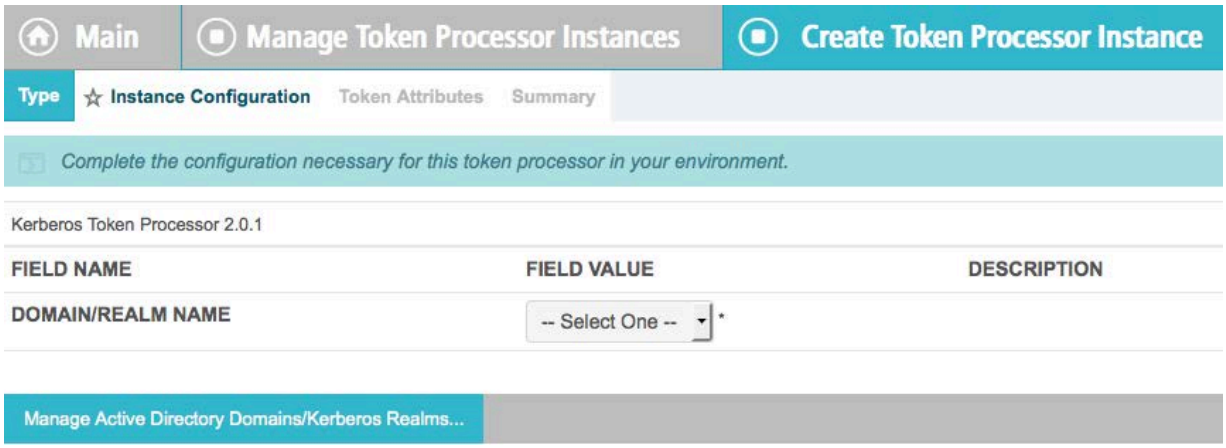
   If you do not see Token Processors on the Main Menu, enable WS-Trust by accessing Roles & Protocols from the Server Settings screen and selecting WS-Trust for the IdP Role.

   Note: To enable token exchange, you may be prompted to provide SAML 1.x and SAML 2.0 federation identifiers for the STS on the Federation Info screen. Refer to the Federation Info screen's Help page for more information.

4. On the Manage Token Processor Instances screen, click **Create New Instance**.

5. On the Type screen, enter an Instance Name and Instance Id, and select **Kerberos Token Processor 2.0.1** as the Type.

6. Click **Next**.



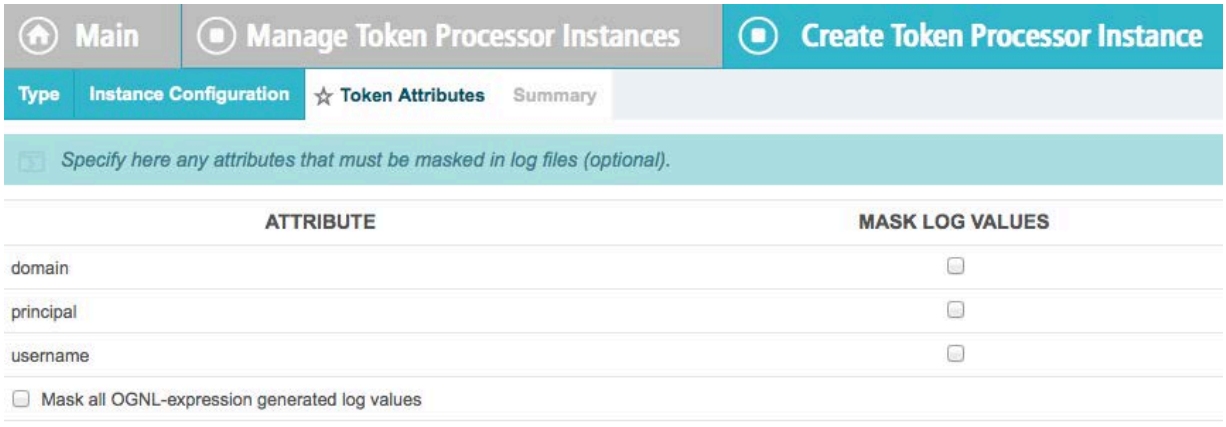
7. On the Instance Configuration screen, choose the applicable Domain or Realm that PingFederate can use to contact Domain Controller(s) or KDC(s).

   A Domain or Realm must be configured for use with a Token Processor. If the Domain or Realm you want does not appear, click **Manage Active Directory Domains/Kerberos Realms** to add it (see Using AD Domains and Kerberos Realms in the PingFederate *Administrator's Manual*).

Note: Kerberos tickets can be accepted from domains other than the domain configured in the Token Processor, as long as there is a transient, two-way trust. This trust exists by default when domains are joined within a single server forest.

---

8. Click **Next**.



9. (Optional) On the Token Attributes screen, select any or all attributes whose values you want to mask in the PingFederate log file.

Note: If OGNL expressions might be used to map derived values into outgoing assertions and you want those values masked, select the associated check box below the Attribute list. (For more information, see Using Attribute Mapping Expressions in the PingFederate *Administrator's Manual*.)

---

10. Click **Next**.
11. On the Summary screen, verify that the information is correct and click **Done**.
12. On the Manage Token Processor Instances screen, click **Save**.

Using the STS Client SDK

Ping Identity provides a Java WS-Trust Client SDK for enabling Web Service applications (Client or Provider) to interact with the PingFederate STS. (The SDK is available for download at `www.pingidentity.com/en/products/downloads`).

The SDK provides functionality for sending a security token to the PingFederate STS for exchange with a returned SAML token, which can then be used to access Web Services across domains. The following code examples show how to send a token and request the exchange. Refer to the SDK documentation for modifications that apply to your site.

Java Sample Code

The code snippet below demonstrates using the PingFederate Java WS-Trust Client SDK to send a Kerberos token to the PingFederate STS:

```
// Example method for obtaining the Kerberos token.
// You will need to implement this for your environment.
String kerbTicket = getKerberosTicket();

// Configure STS Client (IdP side / SP Connection)
STSClientConfiguration stsConfig = new STSClientConfiguration();
stsConfig.setAppliesTo("http://localhost");
stsConfig.setStsEndpoint("https://localhost:9031/idp/sts.wst");
stsConfig.setInTokenType(TokenType.BINARY);
stsConfig.setInTokenValueType("http://docs.oasis-open.org/wss/oasis-wss-
kerberos-token-profile-1.1#GSS_Kerberosv5_AP_REQ");
stsConfig.setIgnoreSSLTrustErrors(true);

//Instantiate the STSClient
STSClient stsClient = new STSClient(stsConfig);

//Register the custom token type URI
stsClient.registerSecurityTokenReference(new
QName(XmlNamespaces.WSSE.uri(), "BinarySecurityToken"), "http://docs.oasis-
open.org/wss/oasis-wss-kerberos-token-profile-1.1#GSS_Kerberosv5_AP_REQ",
new SimpleTokenReference("http://docs.oasis-open.org/wss/oasis-wss-
kerberos-token-profile-1.1#GSS_Kerberosv5_AP_REQ"));

// Send an RST Issue request to PingFederate STS
Element samlToken = stsClient.issueToken(kerbTicket);
```

.NET Sample Code

The code snippet below demonstrates using the Microsoft .NET WCF/WIF framework to send a Kerberos token to the PingFederate STS:

```
//Establish the Kerberos Binding for WS-Trust messaging
KerberosWSTrustBinding kerberosBinding = new KerberosWSTrustBinding()
{
    SecurityMode = SecurityMode.TransportWithMessageCredential,
    TrustVersion = TrustVersion.WSTrust13,
    EnableRsaProofKeys = false
};

//Prepare the WS-Trust Channel Factory providing the previously established
//Kerberos Binding
WSTrustChannelFactory trustChannel = new
WSTrustChannelFactory(kerberosBinding,
"https://idp.domain.com:9031/idp/sts.wst");
trustChannel.Credentials.Windows.ClientCredential.UserName = "user";
trustChannel.Credentials.Windows.ClientCredential.Password = "2Federate";
trustChannel.Credentials.Windows.ClientCredential.Domain = "DOMAIN.COM";

//Set the AppliesTo to the value of the Partner Service Identifier
//configured for the target WS-Trust connection
rst = new RequestSecurityToken(WSTrust13Constants.RequestTypes.Issue)
{
    AppliesTo = new EndpointAddress("http://sp.domain.com")
};

//Issue the request and process the issued SAML token
WSTrustChannel wsTrustChannel =
(WSTrustChannel)trustChannel.CreateChannel();
RequestSecurityTokenResponse rstr = null;
SecurityToken samlToken = idpChannel.Issue(rst, out rstr);
```