Technical Note:

Kerberos Support in Policy Manager 8.0

Table of Contents

Table of Figures

Overview

Policy Manager 8.0 provides the following Kerberos functionality:

Existing Functionality

- The Network Director feature supports the generation of Kerberos tickets for use in communication with downstream services provided it has access to the username and password of the user in need of the Kerberos ticket.
- The Kerberos Impersonation plug-in provides support for Network Director to issue a Kerberos ticket for a user, even without access to a user's password.

New in Policy Manager 8.0

- Ability to protect virtual services with Kerberos authentication using SPNEGO in addition to WS-Security. Policy Manager 7.x SPNEGO support was only available for communicating with downstream services. With Policy Manager 8.0, SPNEGO can also be used for virtual services.

There are two main use cases for protecting a virtual service with Kerberos.

- Authenticate a user using WS-Security and a Kerberos Supporting Token

  The Network Director receives a message protected using WS-Security including a Kerberos ticket in the WS-Security header. The Network Director validates the Kerberos ticket and authenticates the user.

- Authenticate a user using SPNEGO (New in 8.0)

  The Network Director performs an SPNEGO challenge and receives a Kerberos ticket. The Network Director validates the Kerberos ticket and authenticates the user.

There are four main use cases for issuing Kerberos tickets from the Network Director based on the information available to the Network Director identifying the user for whom a ticket is required.

- Generating a Ticket from a Username and Password

  The Network Director receives a user name and password of a user in some manner that can be authenticated using an Active Directory identity system. A downstream service requires Kerberos authentication. The Network Director requests a KDC to issue a Kerberos ticket for the authenticated user.

- Generating a Ticket from a Kerberos Ticket Supporting Delegation

  Kerberos tickets can be generated with an impersonation level supporting delegation, impersonation, or nothing. In this use case the Network Director receives a Kerberos ticket supporting delegation identifying a user that can be authenticated using a Kerberos Identity System. A downstream service requires Kerberos authentication. The Network Director requests a KDC to issue a new Kerberos ticket for the authenticated user ticket using delegation.

- Generating a Ticket from a Kerberos Ticket Supporting Impersonation

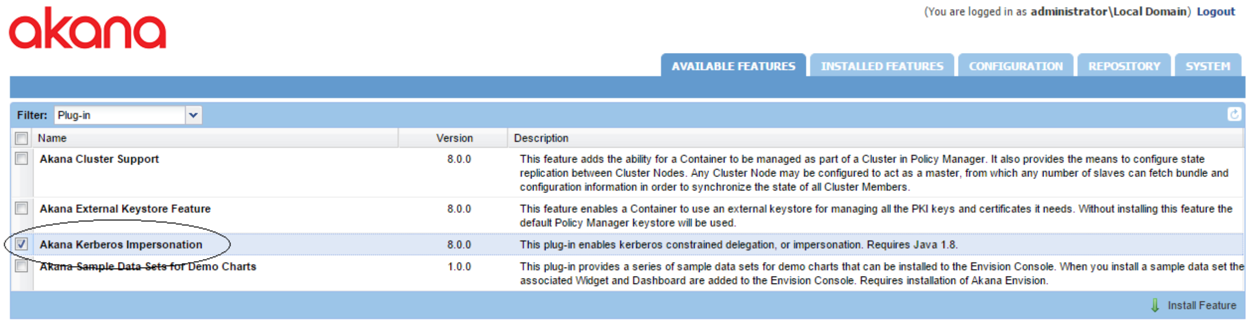
Kerberos tickets can be generated with an impersonation level supporting delegation, impersonation, or nothing. In this use case the Network Director receives a Kerberos ticket supporting impersonation identifying a user that can be authenticated using a Kerberos Identity System. A downstream service requires Kerberos authentication. The Network Director requests a KDC to issue a new Kerberos ticket for the authenticated user ticket using constrained delegation (S4U2Proxy).

- Generating a Ticket from a Username

The Network Director receives user credentials that do not have a password (i.e., SAML assertion or X.509 certificate), which can be authenticated using the corresponding identity system. A downstream service requires Kerberos authentication. The Network Director requests a KDC to issue a Kerberos ticket for the authenticated user using constrained delegation (S4U2Self and S4U2Proxy).

Kerberos Impersonation Plug-in

Constrained delegation requires the Network Director to run using Java 1.8. This is not always an option based on deployment restrictions. Therefore constrained delegation has been separated into its own plug-in named Kerberos Impersonation.



Filter: Plug-in

□	Name	Version	Description
□	Akana Cluster Support	8.0.0	This feature adds the ability for a Container to be managed as part of a Cluster in Policy Manager. It also provides the means to configure state replication between Cluster Nodes. Any Cluster Node may be configured to act as a master, from which any number of slaves can fetch bundle and configuration information in order to synchronize the state of all Cluster Members.
□	Akana External Keystore Feature	8.0.0	This feature enables a Container to use an external keystore for managing all the PKI keys and certificates it needs. Without installing this feature the default Policy Manager keystore will be used.
☑	Akana Kerberos Impersonation	8.0.0	This plug-in enables kerberos constrained delegation, or impersonation. Requires Java 1.8.
□	Akana Sample Data Sets for Demo Charts	1.0.0	This plug-in provides a series of sample data sets for demo charts that can be installed to the Envision Console. When you install a sample data set the associated Widget and Dashboard are added to the Envision Console. Requires installation of Akana Envision.

Figure 1 - Kerberos Impersonation Plug-in

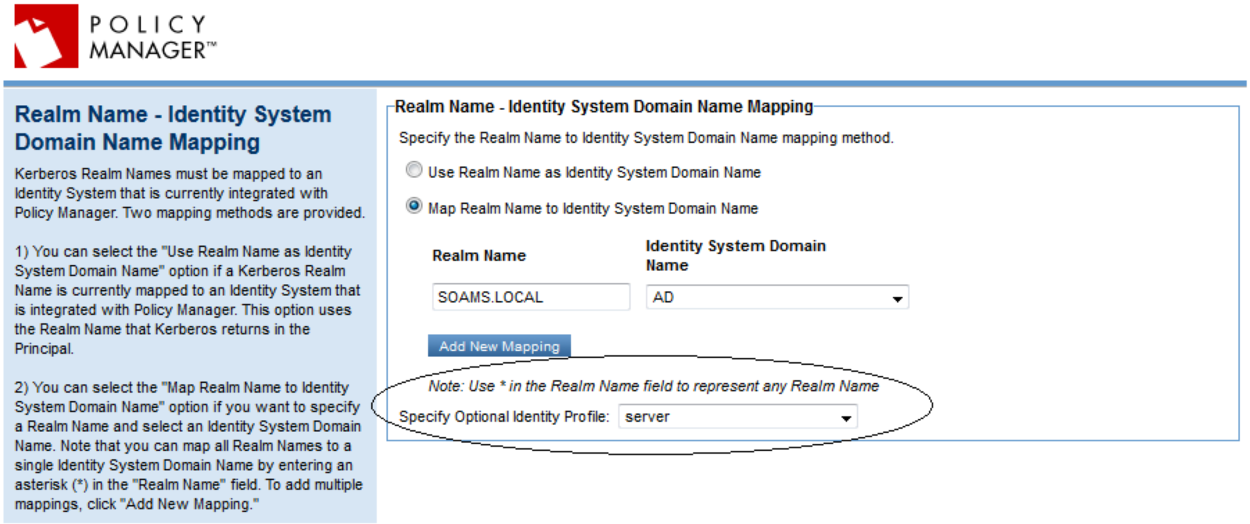
For the use case of creating a ticket with only a username, Java 1.8 should be used and the Kerberos Impersonation plug-in should be installed with the Network Director feature.

Note: This feature is available when the Plug-in filter is selected.

Kerberos Identity System

The Kerberos Identity System supports the issuing of Kerberos tickets in addition to authenticating Kerberos tickets. In order to perform constrained delegation the Network Director must make Kerberos ticket requests to a KDC using the identity of a user with constrained delegation privileges in Active Directory. The identity used is specified in the configuration of the Kerberos Identity System using an

Identity Profile.



POLICY MANAGER™

Realm Name - Identity System Domain Name Mapping

Kerberos Realm Names must be mapped to an Identity System that is currently integrated with Policy Manager. Two mapping methods are provided.

1) You can select the "Use Realm Name as Identity System Domain Name" option if a Kerberos Realm Name is currently mapped to an Identity System that is integrated with Policy Manager. This option uses the Realm Name that Kerberos returns in the Principal.

2) You can select the "Map Realm Name to Identity System Domain Name" option if you want to specify a Realm Name and select an Identity System Domain Name. Note that you can map all Realm Names to a single Identity System Domain Name by entering an asterisk (*) in the "Realm Name" field. To add multiple mappings, click "Add New Mapping."

Realm Name - Identity System Domain Name Mapping

Specify the Realm Name to Identity System Domain Name mapping method.

○ Use Realm Name as Identity System Domain Name

● Map Realm Name to Identity System Domain Name

Realm Name	Identity System Domain Name
SOAMS.LOCAL	AD

*Note: Use * in the Realm Name field to represent any Realm Name*

Specify Optional Identity Profile: server

Figure 2 - Issuing Identity Profile for Kerberos Identity System

When choosing an Identity Profile for this purpose it must be for the same Identity System.

Note: This option is available when you ***modify*** *the identity system only.*

SPNEGO Policy

The Microsoft authored SPNEGO WS-Policy assertion (see http://msdn.microsoft.com/en-us/library/ee525179.aspx) is used in this configuration. When attached to a virtual service the policy instruct the Network Director to issue an SPNEGO challenge and negotiate the passing of a Kerberos ticket.

When attached to a downstream service the policy instructs the Network Director to present a Kerberos ticket when challenged by the downstream service using the SPNEGO negotiate scheme. Often when a web service authored with Microsoft technologies is configured to use SPNEGO that policy assertion will be added to the web service’s WSDL. The policy will be imported into Policy Manager with the WSDL.

SPNs and UPNs

There are multiple options for the Network Director to choose the server to generate a Kerberos ticket for.

When importing the downstream service WSDL an SPN or UPN can be used to identify the server using the standard Microsoft practice of using a WS-Identity Identity element within a WS-Addressing EndpointReference element. There are elements for each SPN and UPN. The following are examples of this approach.

```
<wsdl:port binding="tns:ReadyToUse1" name="BasicWcfServiceEndpoint">
      …
```

```
  <wsa10:EndpointReference>
    <wsa10:Address>https://host/spn/BasicWcfService.svc</wsa10:Address>
    <Identity xmlns="http://schemas.xmlsoap.org/ws/2006/02/addressingidentity">
      <Spn>backend/myhost.acme.local@ACME.LOCAL</Spn>
    </Identity>
  </wsa10:EndpointReference>
</wsdl:port>
```

Figure 3 - SPN in EndpointReference

```
<wsdl:port binding="tns:ReadyToUse1" name="BasicWcfServiceEndpoint">
      …
  <wsa10:EndpointReference>
    <wsa10:Address>https://host/upn/BasicWcfService.svc</wsa10:Address>
    <Identity xmlns="http://schemas.xmlsoap.org/ws/2006/02/addressingidentity">
      <Upn>backend/juser@ACME.LOCAL</Upn>
    </Identity>
  </wsa10:EndpointReference>
</wsdl:port>
```

Figure 4 - UPN in EndpointReference

Another option when using Kerberos for message level security is for an Identity Profile to be associated with the downstream service in the Policy Manager Management Console. The principal name in the Identity Profile will be used as the SPN.

Finally, if neither of the above two options can be used by the Network Director it will use the downstream service's address as a host based SPN.

With all the options above the values chosen by the Network Director for the server must be registered in Active Directory correctly.

Setting up Use Cases

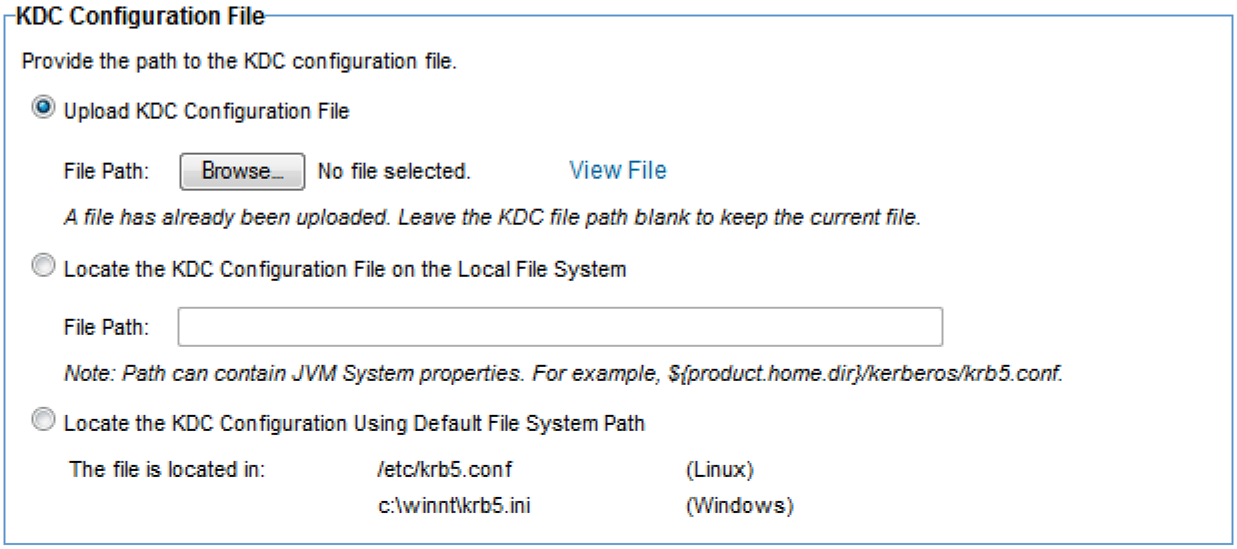
In the following sections the use cases listed previously and some variations will be described in more detail.

Authenticate a User using WS-Security and a Kerberos Supporting Token

The following are the steps to authenticate a user from a WS-Security header with a Kerberos token.

Create the Kerberos Identity System

An Identity System that integrates with a KDC will be needed to authenticate Kerberos tickets. When configuring the identity system a KDC configuration file will be referenced (or loaded) that will provide the needed information for communicating with the KDC. See http://web.mit.edu/kerberos/krb5-1.5/krb5-1.5/doc/krb5-admin/krb5.conf.html for a more in-depth description about the content of this file.



KDC Configuration File

Provide the path to the KDC configuration file.

● Upload KDC Configuration File

File Path: Browse... No file selected. View File

A file has already been uploaded. Leave the KDC file path blank to keep the current file.

○ Locate the KDC Configuration File on the Local File System

File Path:

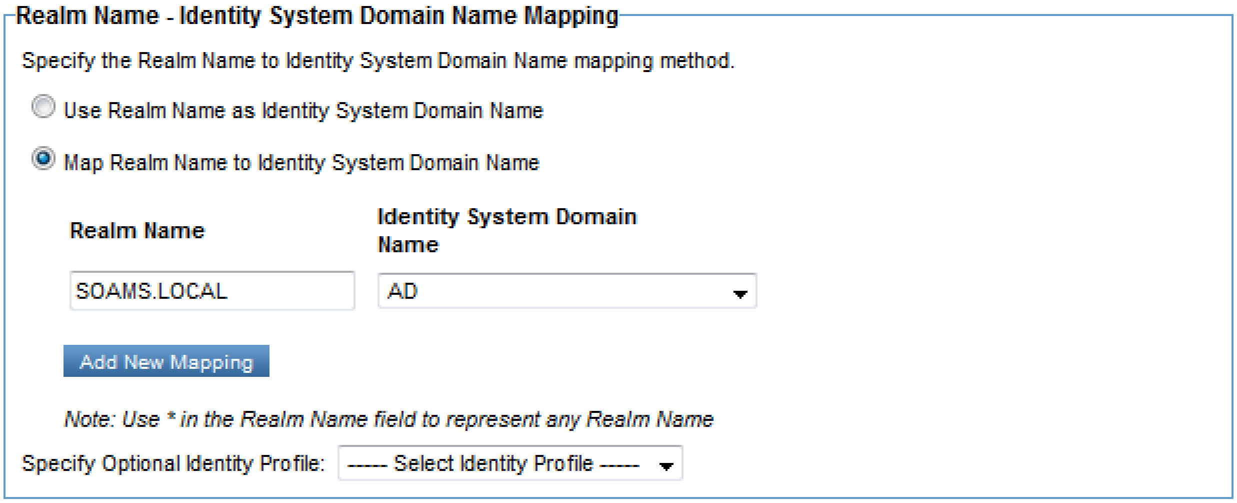
Note: Path can contain JVM System properties. For example, ${product.home.dir}/kerberos/krb5.conf.

○ Locate the KDC Configuration Using Default File System Path

The file is located in:	/etc/krb5.conf	(Linux)
	c:\winnt\krb5.ini	(Windows)

Figure 5 - Adding KDC Configuration File to Kerberos Identity System

In the file you will define the “realms” that are supported by this Identity System. It is important to map the Active Directory Identity System, or Domain, created in the previous step to one of the realms in the KDC configuration file. If more than one realm and/or AD domain is supported multiple mappings may need to be defined.



Realm Name - Identity System Domain Name Mapping

Specify the Realm Name to Identity System Domain Name mapping method.

○ Use Realm Name as Identity System Domain Name

● Map Realm Name to Identity System Domain Name

Realm Name	Identity System Domain Name
SOAMS.LOCAL	AD

Add New Mapping

*Note: Use * in the Realm Name field to represent any Realm Name*

Specify Optional Identity Profile: ----- Select Identity Profile -----

Figure 6 - Mapping Realms to Domains in Kerberos Identity System

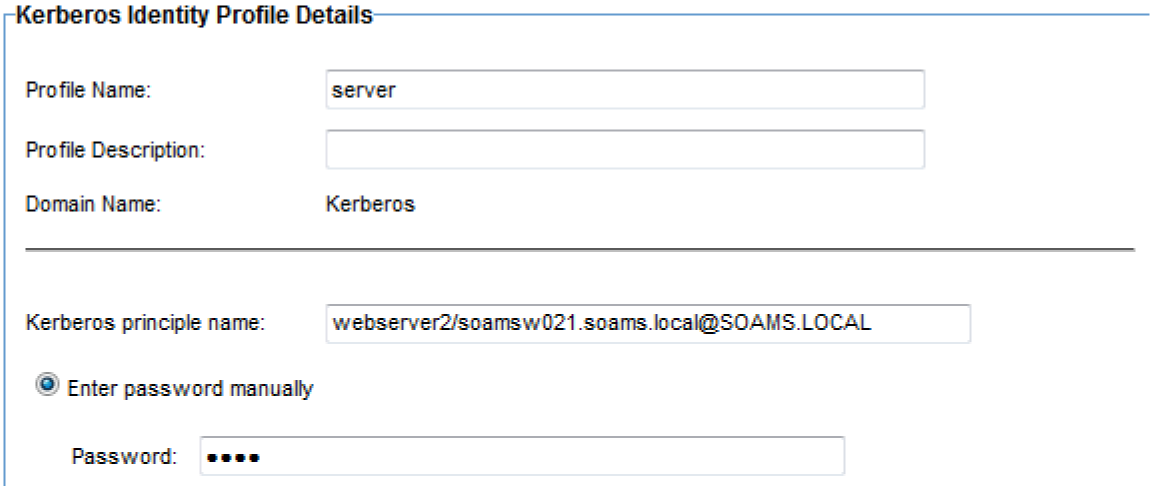
Note: There is no need to specify an Identity Profile for the Identity System for this use case as it does not require constrained delegation.

Note: This screen is available when you ***modify*** *the identity system only.*

Create the Virtual Service Identity Profile

The Network Director will need to verify the correct SPN is referenced in any received Kerberos ticket. When defining a virtual service an SPN will need to be assigned to it. This is done through the use of Kerberos Identity Profiles. A Kerberos Identity Profile will represent an SPN for a particular Kerberos Identity System. The same identity profile can be assigned to multiple virtual services if SPN reuse is required.

To create an identity profile, navigate to the Configure -> Security -> Identity Profiles area and select **Add Identity Profile**.



Kerberos Identity Profile Details

Profile Name: server

Profile Description:

Domain Name: Kerberos

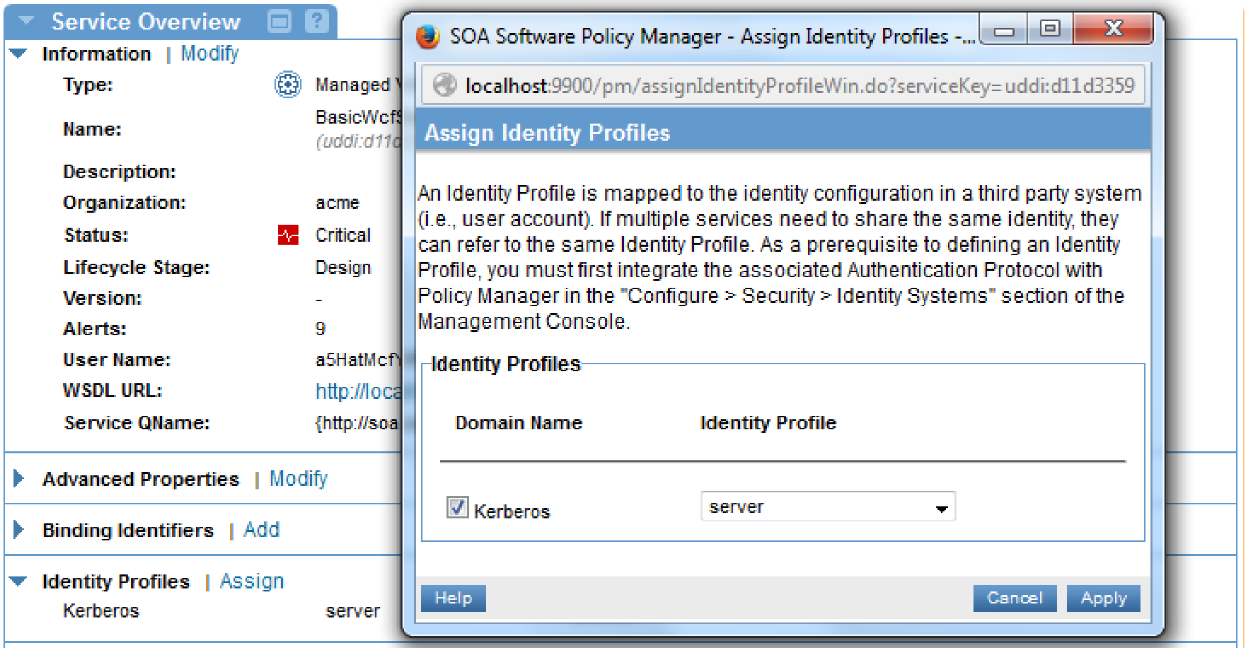
Kerberos principle name: webserver2/soamsw021.soams.local@SOAMS.LOCAL

● Enter password manually

Password: ●●●●

Figure 7 - Creating an Identity Profile

The principal name will be used as the SPN to check for in any received Kerberos ticket. Clients wishing to call the virtual service will need to create tickets with this SPN. The Identity Profile is then associated with the virtual service.



Service Overview

Information | Modify

Type:	Managed \
Name:	BasicWcfS (uddi:d11c
Description:	
Organization:	acme
Status:	Critical
Lifecycle Stage:	Design
Version:	-
Alerts:	9
User Name:	a5HatMcfY
WSDL URL:	http://loca
Service QName:	{http://soa

Advanced Properties | Modify

Binding Identifiers | Add

Identity Profiles | Assign

Kerberos server

SOA Software Policy Manager - Assign Identity Profiles -...

localhost:9900/pm/assignIdentityProfileWin.do?serviceKey=uddi:d11d3359

Assign Identity Profiles

An Identity Profile is mapped to the identity configuration in a third party system (i.e., user account). If multiple services need to share the same identity, they can refer to the same Identity Profile. As a prerequisite to defining an Identity Profile, you must first integrate the associated Authentication Protocol with Policy Manager in the "Configure > Security > Identity Systems" section of the Management Console.

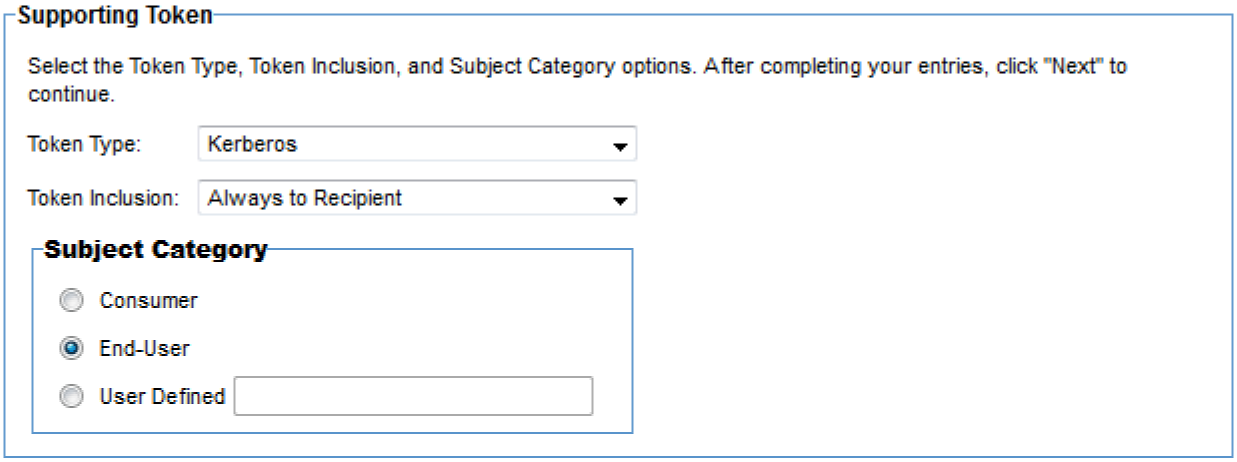
Identity Profiles

Domain Name	Identity Profile
☑ Kerberos	server ▾

Figure 8 - Assigning an Identity Profile to a Service

Create the Kerberos Security Policy

In this use case a Kerberos ticket will be sent to the Network Director by a client. The Kerberos ticket will be presented in a WS-Security SOAP header. So instead of using the HTTP Security Policy with Basic Authentication as used in previous examples a WS-Security Policy will be created. We will rely on the transport for message confidentiality and integrity and simply require a Kerberos Token for authentication purposes. This is implemented using a WS-Security Transport Binding policy and a WS-Security Supporting Tokens policy. In the Supporting Tokens policy the Kerberos Token option is chosen.



Supporting Token

Select the Token Type, Token Inclusion, and Subject Category options. After completing your entries, click "Next" to continue.

Token Type: Kerberos ▾

Token Inclusion: Always to Recipient ▾

Subject Category

○ Consumer

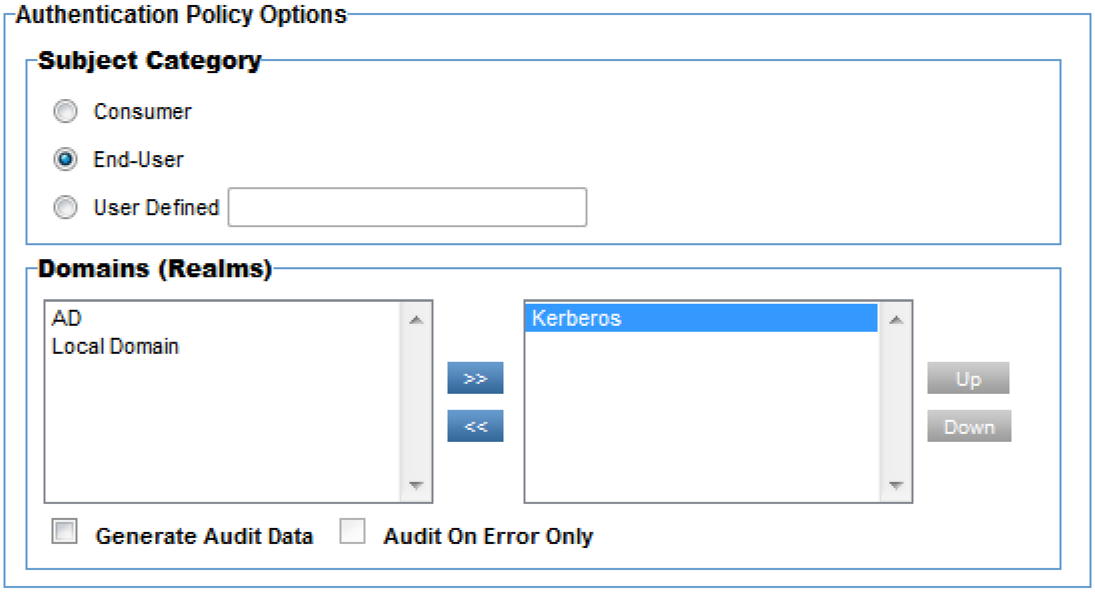
● End-User

○ User Defined

Figure 9 - Configuring the WS-Security Supporting Tokens Policy to Use a Kerberos Token

Create the Authentication Policy

Create an Authentication policy that will authenticate a user against the Kerberos domain created previously.


Authentication Policy Options

Subject Category

○ Consumer
● End-User
○ User Defined

Domains (Realms)

AD
Local Domain

Kerberos

□ Generate Audit Data □ Audit On Error Only

Figure 10 - Configuring the Authentication Policy to Use the Kerberos Domain

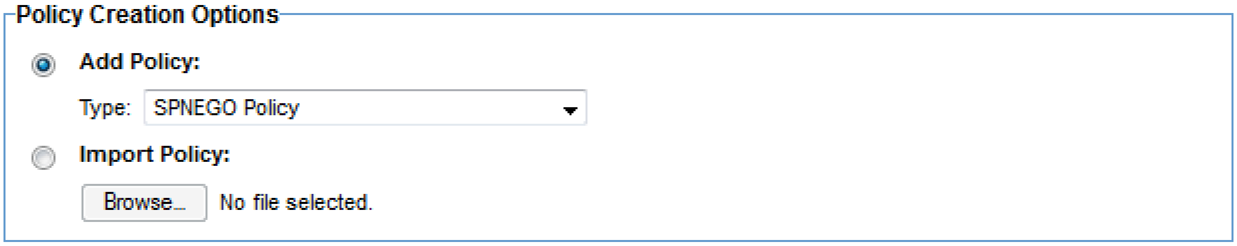
Attach the two policies to the virtual service and the setup is complete. When a client sends a SOAP message with a WS-Security header with a Kerberos Token, the Network Director will authenticate the ticket. The user for which the ticket was created can then be used in authorization or downstream identity requirements, such as requiring a SAML assertion or X.509 certificate, or even another Kerberos ticket.

Authenticate a User using Kerberos and SPNEGO

The following are the steps to authenticate a user from an SPNEGO negotiated Kerberos ticket. The steps are identical to the ones listed for authenticating a user using WS-Security and Kerberos with the following exceptions.

Create the SPNEGO Security Policy

Instead of creating a WS-Security policy and attaching it to the virtual service, create an SPNEGO operational policy in the Policy Manager Management Console. The policy does not have any configurable options. The subject category identifying the identity used to fulfill the SPNEGO challenge is assumed to be "end-user." After creating the policy, activate it.


Policy Creation Options

● Add Policy:

Type: SPNEGO Policy

○ Import Policy:

No file selected.

Figure 11 - Creating the SPNEGO Operational Policy

When a client sends a request to the Network Director for this virtual service the client and Network Director will negotiate using SPNEGO. A Kerberos ticket will be sent to the Network Director and be authenticated. The user for which the ticket was created can then be used in authorization or downstream identity requirements, such as requiring a SAML assertion or X.509 certificate, or even another Kerberos ticket.

Generating a Ticket for SPNEGO from a Username and Password

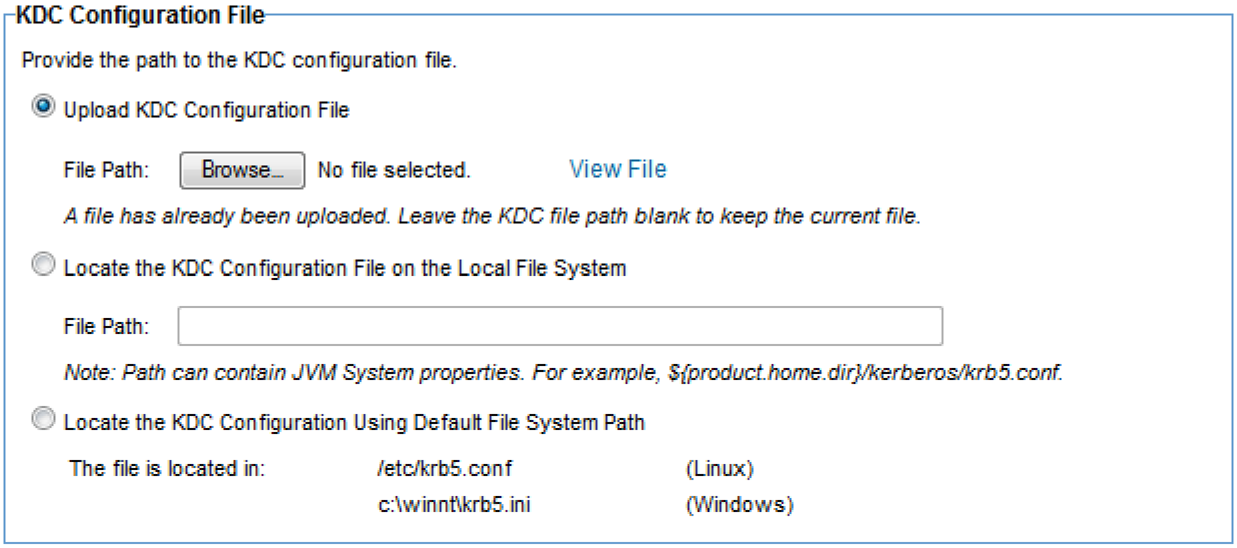
The following are the steps to communicate with a downstream service using SPNEGO using a name and password as input.

Create the Active Directory Identity System

An Identity System that integrates with Active Directory will be needed to authenticate a username and password.

Create the Kerberos Identity System

An Identity System that integrates with a KDC will be needed to issue Kerberos tickets. When configuring the identity system a KDC configuration file will be referenced (or loaded) that will provide the needed information for communicating with the KDC. See http://web.mit.edu/kerberos/krb5-1.5/krb5-1.5/doc/krb5-admin/krb5.conf.html for a more in-depth description about the content of this file.



KDC Configuration File

Provide the path to the KDC configuration file.

● Upload KDC Configuration File

File Path: Browse... No file selected. View File

A file has already been uploaded. Leave the KDC file path blank to keep the current file.

○ Locate the KDC Configuration File on the Local File System

File Path:

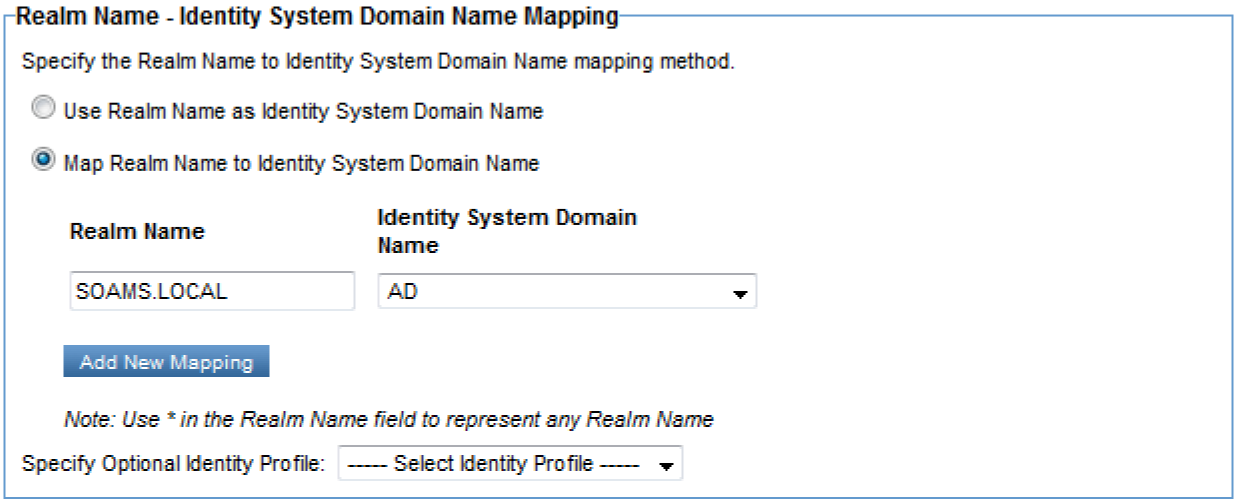
Note: Path can contain JVM System properties. For example, ${product.home.dir}/kerberos/krb5.conf.

○ Locate the KDC Configuration Using Default File System Path

The file is located in: /etc/krb5.conf (Linux)
c:\winnt\krb5.ini (Windows)

Figure 12 - Adding KDC Configuration File to Kerberos Identity System

In the file you will define the “realms” that are supported by this Identity System. It is important to map the Active Directory Identity System, or Domain, created in the previous step to one of the realms in the KDC configuration file. If more than one realm and/or AD domain is supported multiple mappings may need to be defined.



Realm Name - Identity System Domain Name Mapping

Specify the Realm Name to Identity System Domain Name mapping method.

○ Use Realm Name as Identity System Domain Name

● Map Realm Name to Identity System Domain Name

Realm Name	Identity System Domain Name
SOAMS.LOCAL	AD

Add New Mapping

*Note: Use * in the Realm Name field to represent any Realm Name*

Specify Optional Identity Profile: ----- Select Identity Profile -----

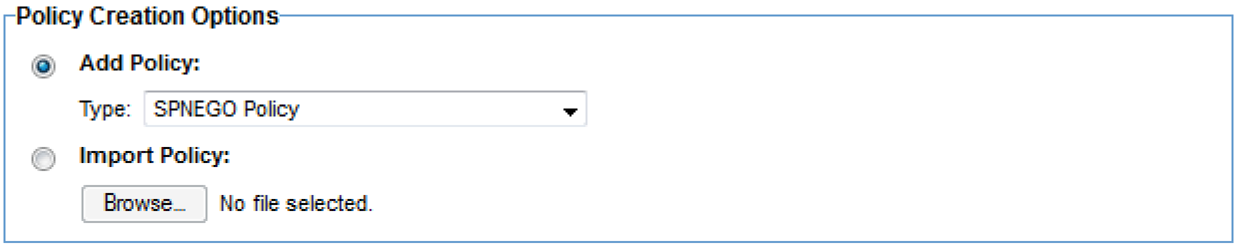
Figure 13 - Mapping Realms to Domains in Kerberos Identity System

Note: There is no need to specify an Identity Profile for the Identity System for this use case as it does not require constrained delegation.

Note: This screen is available when you ***modify*** *the identity system only.*

Create the Downstream Security Policy

In this use case the downstream service will perform an SPNEGO challenge. To accommodate this create an SPNEGO operational policy in the Policy Manager Management Console. The policy does not have any configurable options. The subject category identifying the identity used to fulfill the SPNEGO challenge is assumed to be "end-user." After creating the policy, activate it.



Policy Creation Options

● **Add Policy:**

Type: SPNEGO Policy

○ **Import Policy:**

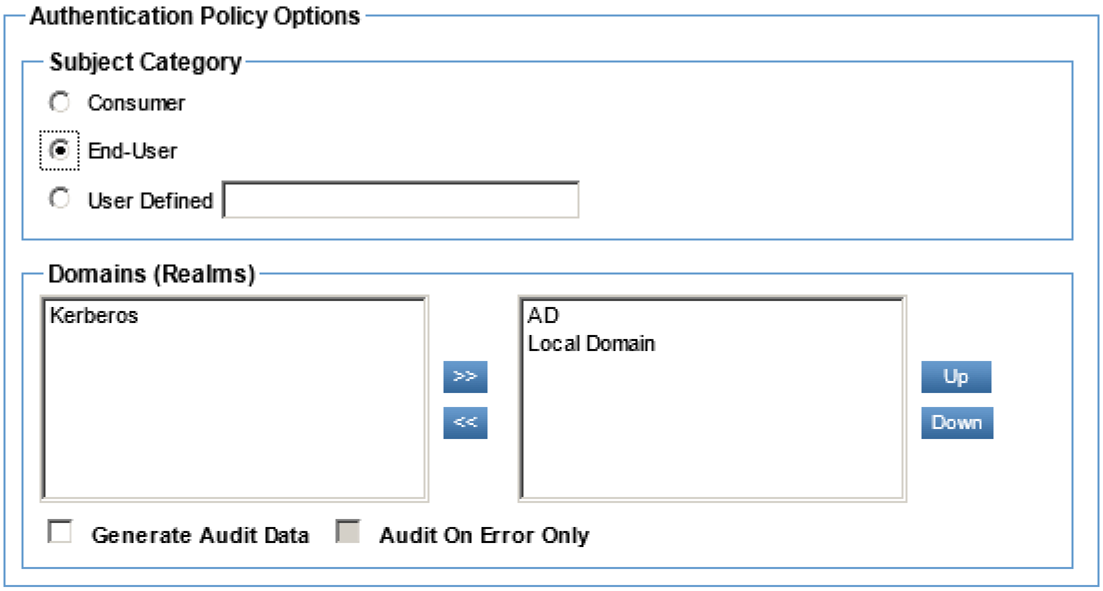
Browse... No file selected.

Figure 14 - Creating the SPNEGO Operational Policy

It is possible that when registering the downstream service (see below) the SPNEGO policy is already defined in the imported WSDL. If this is the case there will not be a need to create the SPNEGO policy as a separate step.

Create the Authentication Policy

Create an Authentication policy that will authenticate a user against the Active Directory domain created previously. After creating the policy, activate it.



Authentication Policy Options

Subject Category

○ Consumer

● End-User

○ User Defined

Domains (Realms)

Kerberos

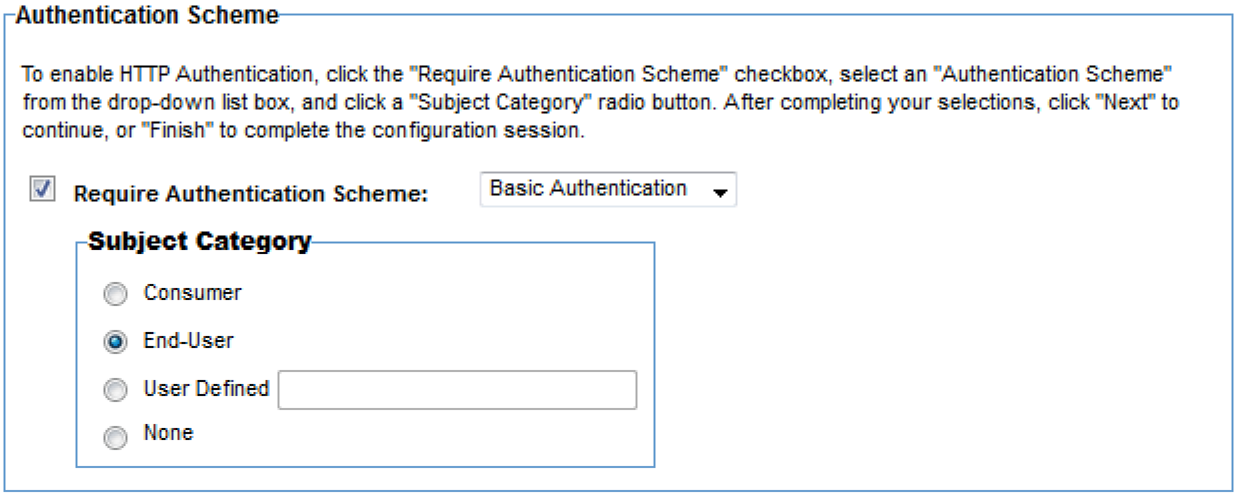
AD
Local Domain

□ Generate Audit Data □ Audit On Error Only

Figure 15 - Configuring the Authentication Policy to Use the AD Domain

Create the Username/Password Security Policy

There is more than one option for security policies that will require and extract usernames and passwords such as the HTTP Security Policy and WS-Security Policy. In this example we will use the HTTP Security Policy configured with the Basic Authentication Scheme option. It is important that the Subject Category selected in this policy match the one chosen in the Authentication Policy. After creating the policy, activate it.



Authentication Scheme

To enable HTTP Authentication, click the "Require Authentication Scheme" checkbox, select an "Authentication Scheme" from the drop-down list box, and click a "Subject Category" radio button. After completing your selections, click "Next" to continue, or "Finish" to complete the configuration session.

☑ Require Authentication Scheme: Basic Authentication

Subject Category

○ Consumer

● End-User

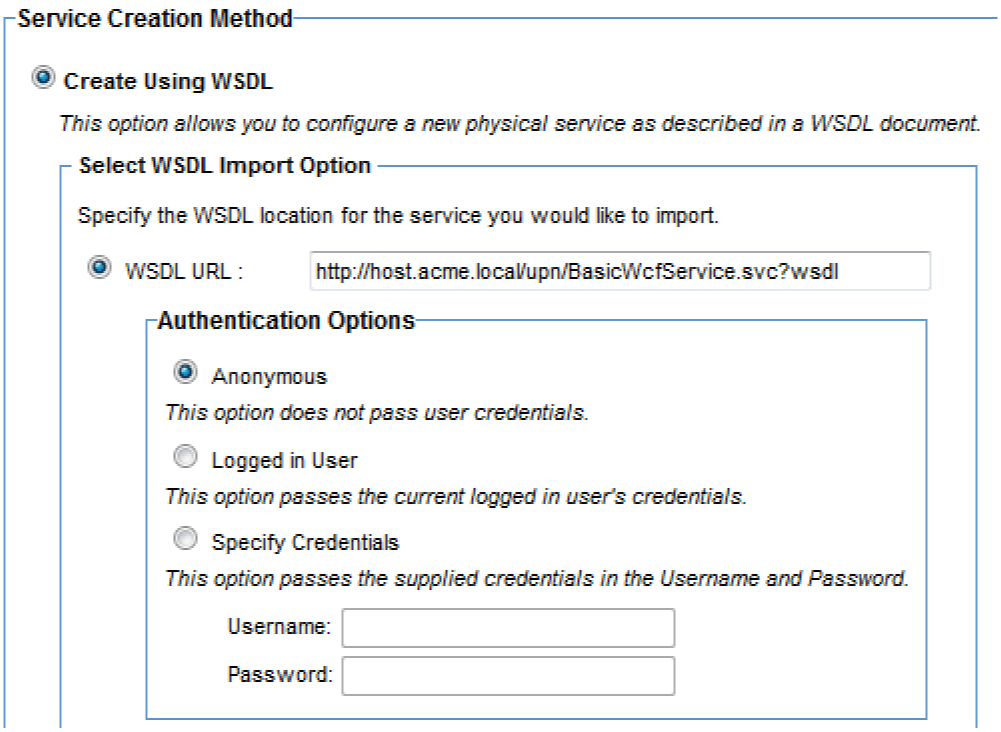
○ User Defined

○ None

Figure 16 - Configuring the HTTP Security Policy

Register the Downstream Service

In this example the downstream service is a SOAP service hosted in IIS. To register the service in Policy Manager its WSDL will be imported. A WSDL can be imported with a URL, file, or zip file.



Service Creation Method

● Create Using WSDL

This option allows you to configure a new physical service as described in a WSDL document.

Select WSDL Import Option

Specify the WSDL location for the service you would like to import.

● WSDL URL : http://host.acme.local/upn/BasicWcfService.svc?wsdl

Authentication Options

● Anonymous

This option does not pass user credentials.

○ Logged in User

This option passes the current logged in user's credentials.

○ Specify Credentials

This option passes the supplied credentials in the Username and Password.

Username:

Password:

Figure 17 - Importing WSDL

If the WSDL has an SPN or UPN it will be used to identify the service to generate Kerberos tickets for. If there is no SPN or UPN the host in the SOAP endpoint in the WSDL will be used.

Once the service is registered, if no SPNEGO policy was imported with the WSDL then the SPNEGO policy created in a previous step should be attached to this service in the Policy Manager Management Console.

Create the Virtual Service

Create a Virtual Service for the previously registered downstream service in the Policy Manager Management Console. Provide it an address that it can be reached at. Attach the username/password security policy and authentication policy.

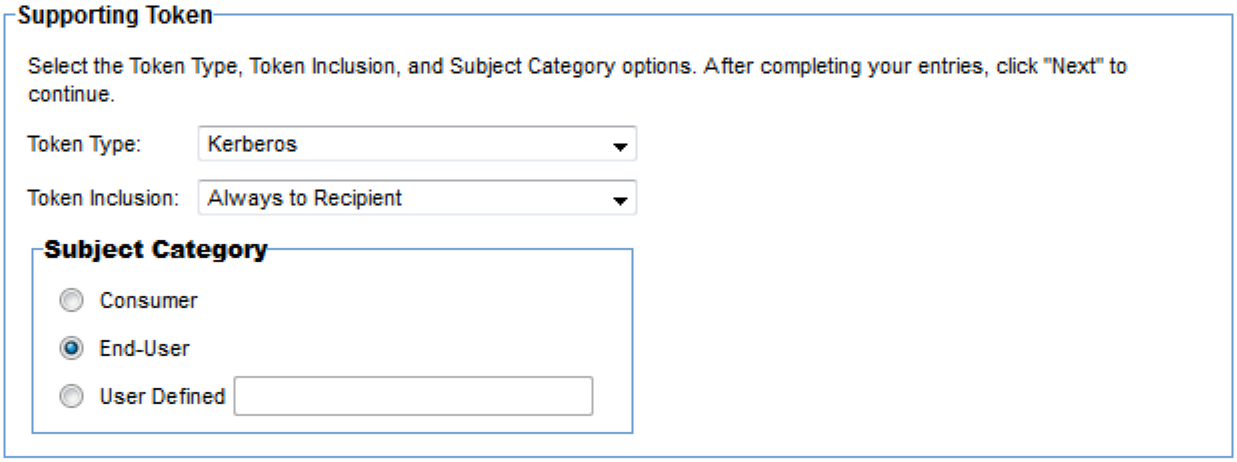
The setup is now complete. When a client sends an HTTP Basic Authentication header with a name and password of an Active Directory user, the Network Director will authenticate the user and then generate a Kerberos ticket for the downstream service in response to an SPNEGO negotiate challenge.

Generating a Ticket for WS-Security from a Username and Password

The following are the steps to communicate with a downstream service using a WS-Security Kerberos Token using a name and password as input. The steps are identical to the ones listed for SPNEGO with the exception of the “Create the Downstream Security Policy” and an additional optional step “Create the Downstream Identity Profile.”

Create the Downstream Security Policy

In this use case the downstream service does not use SPNEGO but expects a WS-Security header with a Kerberos Token. This is accommodated through the definition of a WS-Security Policy. There are a few different options for how that policy can be constructed. In this example we will rely on the transport for message confidentiality and integrity and simply add a Kerberos Token for authentication purposes. This is implemented using a WS-Security Transport Binding policy and a WS-Security Supporting Tokens policy. In the Supporting Tokens policy the Kerberos Token option is chosen. After creating the policy, activate it.



Supporting Token

Select the Token Type, Token Inclusion, and Subject Category options. After completing your entries, click "Next" to continue.

Token Type: Kerberos

Token Inclusion: Always to Recipient

Subject Category

○ Consumer

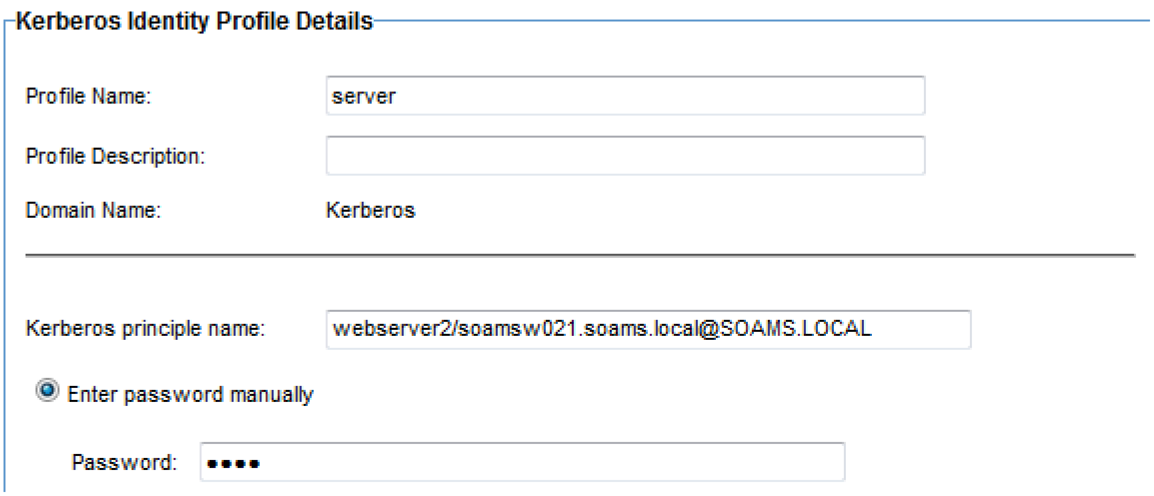
● End-User

○ User Defined

Figure 18 - Configuring the WS-Security Supporting Tokens policy to Use a Kerberos Token

Create the Downstream Identity Profile

This step is not needed if a UPN or SPN in the downstream WSDL is provided or if the host of the downstream endpoint is satisfactory for this use case. However, it is often with message level security as is provided with WS-Security that the true SPN of the Kerberos token may be something unrelated to the downstream service definition. To support that case an Identity Profile can be created for the desired SPN and associated with the downstream service.



Kerberos Identity Profile Details

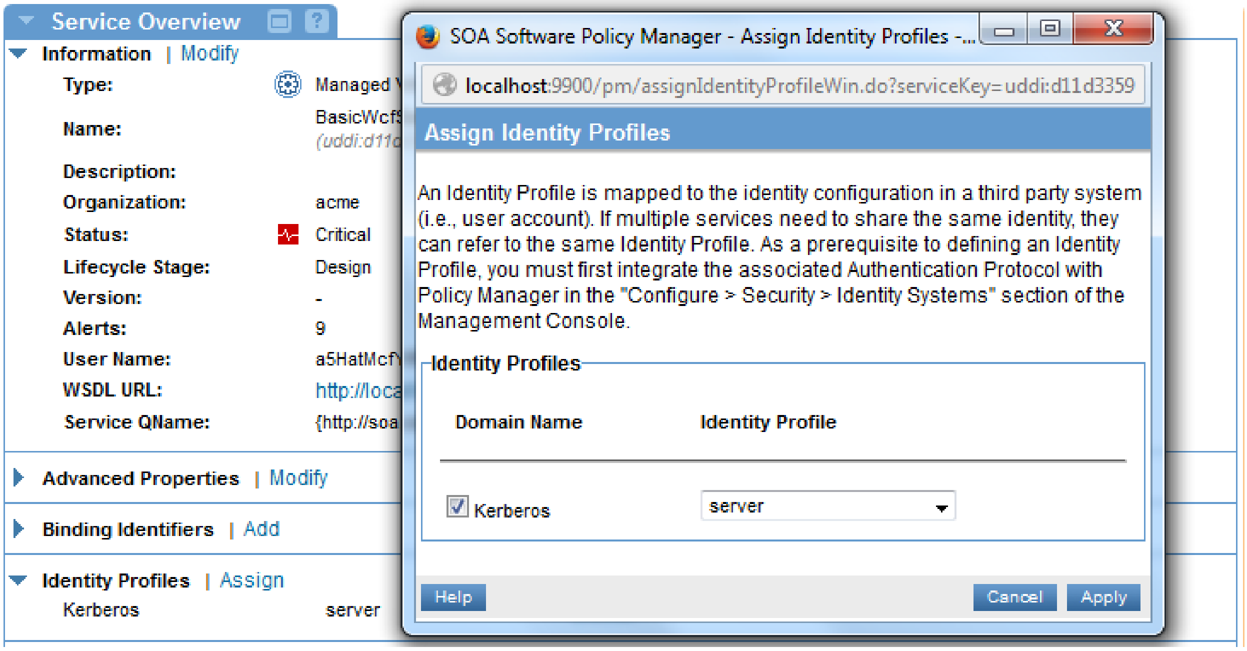
Profile Name:	server
Profile Description:	
Domain Name:	Kerberos
Kerberos principle name:	webserver2/soamsw021.soams.local@SOAMS.LOCAL

● Enter password manually

Password: ••••

Figure 19 - Creating an Identity Profile

The principal name will be used as the SPN for the generated Kerberos ticket. The Identity Profile is then associated with the downstream service.



Information | Modify

Type:	Managed \
Name:	BasicWcf (uddi:d11d
Description:	
Organization:	acme
Status:	Critical
Lifecycle Stage:	Design
Version:	-
Alerts:	9
User Name:	a5HatMcf
WSDL URL:	http://loca
Service QName:	{http://soa

Advanced Properties | Modify

Binding Identifiers | Add

Identity Profiles | Assign

Kerberos server

SOA Software Policy Manager - Assign Identity Profiles -...

localhost:9900/pm/assignIdentityProfileWin.do?serviceKey=uddi:d11d3359

Assign Identity Profiles

An Identity Profile is mapped to the identity configuration in a third party system (i.e., user account). If multiple services need to share the same identity, they can refer to the same Identity Profile. As a prerequisite to defining an Identity Profile, you must first integrate the associated Authentication Protocol with Policy Manager in the "Configure > Security > Identity Systems" section of the Management Console.

Identity Profiles

Domain Name	Identity Profile
☑ Kerberos	server

Figure 20 - Assigning an Identity Profile to a Service

The setup is now complete. When a client sends an HTTP Basic Authentication header with a name and password of an Active Directory user, the Network Director will authenticate the user and then generate a Kerberos ticket for the downstream service using the associated Identity Profile principal as the SPN and send it in a WS-Security header.

Generating a Ticket from a Kerberos Ticket Supporting Delegation

The following are the steps to communicate with a downstream service using SPNEGO by delegating a ticket created with an impersonation level of delegation. The steps are identical to the ones listed for Generating a Ticket for SPNEGO from a Username and Password with the following exceptions.

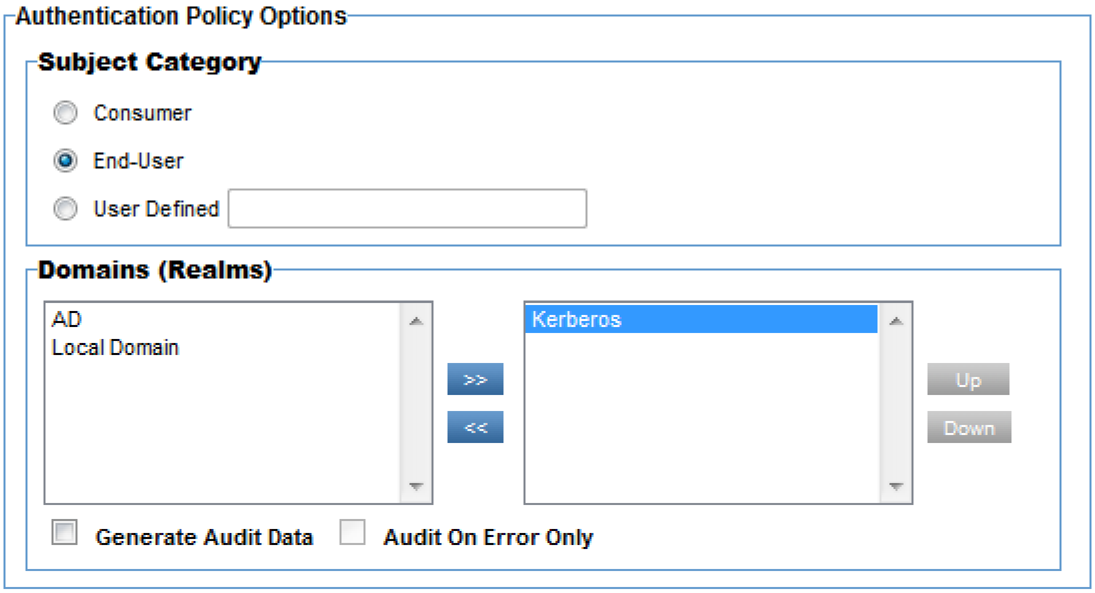
Create Delegating Identity Profile

In order for a ticket to be delegated the recipient service for that ticket must be configured with the ability to perform delegation accordingly in Active Directory. Create an Identity Profile for that service in the Policy Manager Management Console following the steps in the WS-Security with Kerberos token authentication use case above. Assign the Identity Profile to the Virtual Service.

Create the Kerberos Security Policy

Create a WS-Security policy with a Kerberos supporting token as done in the WS-Security with Kerberos Token authentication use case above. Create the Authentication Policy

Whereas in the previous examples the Authentication policy used the Active Directory Domain in this example create an Authentication policy that uses the Kerberos Domain to authenticate the incoming Kerberos ticket.



Authentication Policy Options

Subject Category

○ Consumer

● End-User

○ User Defined

Domains (Realms)

AD
Local Domain

Kerberos

□ Generate Audit Data □ Audit On Error Only

Figure 21 - Configuring the Authentication Policy to Use the Kerberos Domain

The setup is now complete. When a client sends a SOAP message with a WS-Security header with a Kerberos Token supporting delegation for an Active Directory user, the Network Director will authenticate the ticket and then perform delegation using the identity associated with the Virtual Service and generate a new Kerberos ticket for the downstream service.

Generating a Ticket from a Kerberos Ticket Supporting Impersonation

The following are the steps to communicate with a downstream service using SPNEGO by delegating a ticket created with an impersonation level of delegation. The steps are identical to the ones listed for Generating a Ticket from a Kerberos Ticket Supporting Delegation with the following exceptions:

Configure Containers to Use Java 1.8

Install Java 1.8. Edit the setDEMSEnv.bat file in the sm70/bin directory and reference the location of the Java 1.8 JRE instead of the default.

Install the Kerberos Impersonation Plug-in

From the Akana Administration Console of all containers with the Network Director feature install the Kerberos Impersonation Plug-in. There are no configuration options for the plug-in.

The setup is now complete. When a client sends a SOAP message with a WS-Security header with a Kerberos Token supporting impersonation for an Active Directory user, the Network Director will authenticate the ticket and then perform constrained delegation using the identity associated with the Virtual Service and generate a new Kerberos ticket for the downstream service.

Note: This feature is available when the Plug-in filter is selected.

Create Impersonating Identity Profile

In order for a ticket to be delegated the recipient service for that ticket must be configured with the ability to perform impersonation accordingly in Active Directory. Create an Identity Profile for that service in the Policy Manager console. Assign the Identity Profile to the Virtual Service.

Generating a Ticket for SPNEGO from a Username

The following are the steps to communicate with a downstream service using SPNEGO using only a username as input. The steps are identical to the ones listed for Generating a Ticket for SPNEGO from a Username and Password with the following exceptions:

Configure Containers to Use Java 1.8

Install Java 1.8. Edit the setDEMSEnv.bat file in the sm70/bin directory and reference the location of the Java 1.8 JRE instead of the default.

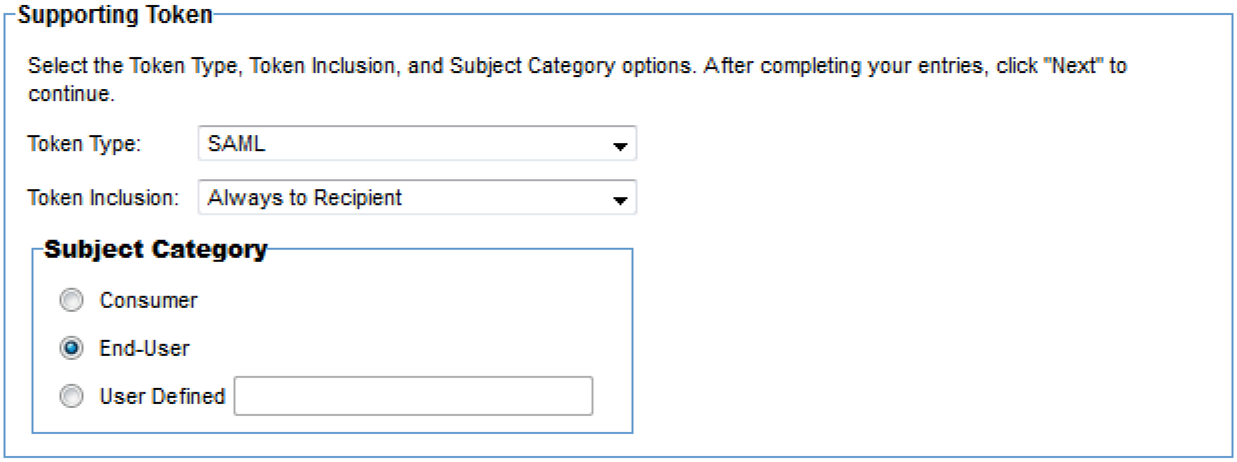
Install the Kerberos Impersonation Plug-in

From the Akana Administration Console of all containers with the Network Director feature install the Kerberos Impersonation Plug-in. There are no configuration options for the plug-in.

Create the Username Security Policy

There is more than one option for security policies that will require different types of tokens that can be authenticated and result in an authenticated username, such as WS-Security Policy using SAML or X.509 Tokens. In this example a SAML Token will be presented by the client in a WS-Security SOAP header. So instead of using the HTTP Security Policy with Basic Authentication as used in previous examples a WS-Security Policy will be created. We will rely on the transport for message confidentiality and integrity and simply require a SAML Token for authentication purposes. This is implemented using a WS-Security

Transport Binding policy and a WS-Security Supporting Tokens policy. In the Supporting Tokens policy the SAML Token option is chosen. After creating the policy, activate it.



Supporting Token

Select the Token Type, Token Inclusion, and Subject Category options. After completing your entries, click "Next" to continue.

Token Type: SAML

Token Inclusion: Always to Recipient

Subject Category

○ Consumer

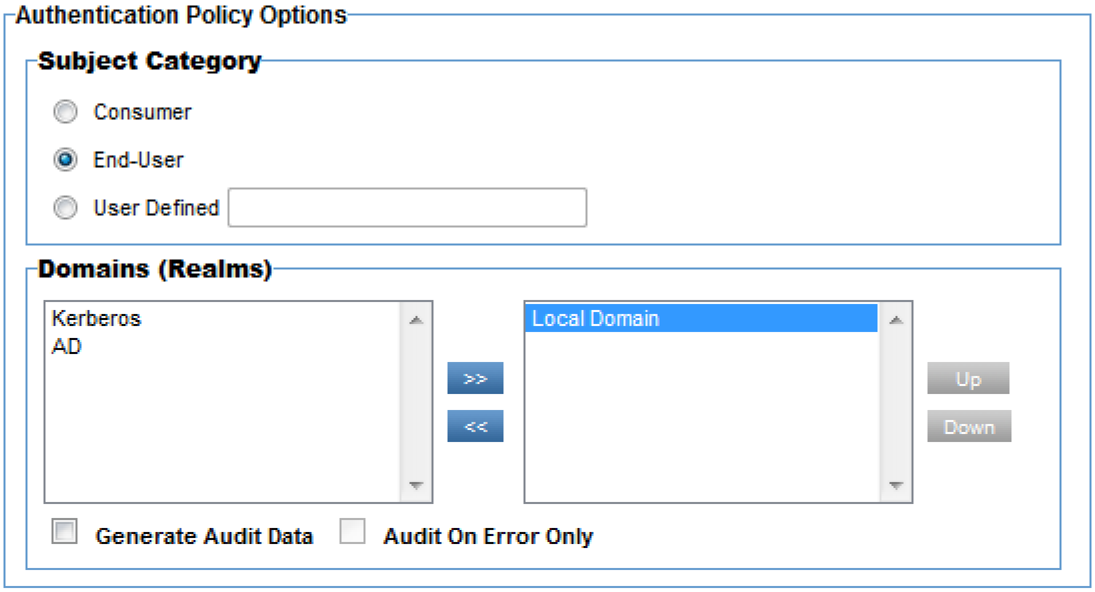
● End-User

○ User Defined

Figure 22 - Configuring the WS-Security Supporting Tokens Policy to Use a SAML Token

Create the Authentication Policy

Whereas in previous examples the Authentication policy used the Active Directory Domain in this example create an Authentication policy that uses the Domain that can authenticate the SAML Token. In this example we will assume the SAML Token was issued using the default Policy Manager Local Domain and therefore the Authentication policy will reference Local Domain. After creating the policy, activate it.



Authentication Policy Options

Subject Category

○ Consumer

● End-User

○ User Defined

Domains (Realms)

Kerberos
AD

Local Domain

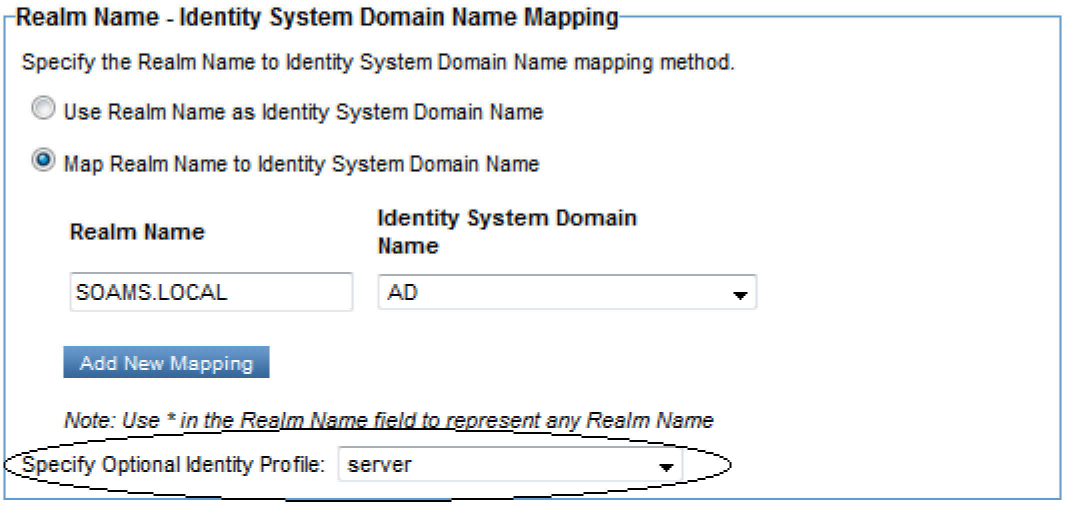
□ Generate Audit Data □ Audit On Error Only

Figure 23 - Configuring the Authentication Policy to Use the Local Domain

It is important to note that although the Local Domain is used to authenticate the SAML Token, the user identified by the SAML Token must be an existing Active Directory user. If not, a Kerberos ticket cannot be generated for that user.

Create the Kerberos Identity System

The same steps for configuring a Kerberos Identity System previously are followed except that for this use case constrained delegation is required and therefore an issuing Identity Profile needs to be associated with the identity system. First a user must be given constrained delegation permissions in Active Directory. Then an Identity Profile should be created for that user in the Policy Manager Management Console. Finally that Identity Profile should be selected in the final page of the Modify Kerberos Identity System Wizard.



Realm Name - Identity System Domain Name Mapping

Specify the Realm Name to Identity System Domain Name mapping method.

○ Use Realm Name as Identity System Domain Name

● Map Realm Name to Identity System Domain Name

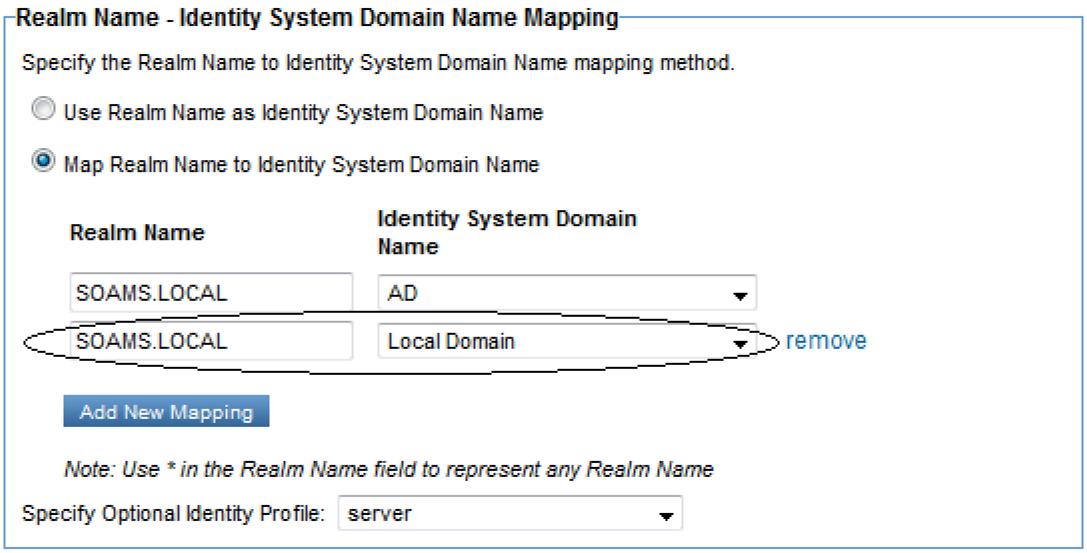
Realm Name	Identity System Domain Name
SOAMS.LOCAL	AD

*Note: Use * in the Realm Name field to represent any Realm Name*

Specify Optional Identity Profile: server

Figure 24 - Adding an Issuing Identity Profile to the Kerberos Identity System

In addition, since SAML Tokens will be authenticated using the Local Domain a new realm mapping needs to be added for the Local Domain. This tells the Network Director that users authenticated against the Local Domain will use the specified realm when issuing a Kerberos ticket.



Realm Name - Identity System Domain Name Mapping

Specify the Realm Name to Identity System Domain Name mapping method.

○ Use Realm Name as Identity System Domain Name

● Map Realm Name to Identity System Domain Name

Realm Name	Identity System Domain Name	
SOAMS.LOCAL	AD	
SOAMS.LOCAL	Local Domain	remove

*Note: Use * in the Realm Name field to represent any Realm Name*

Specify Optional Identity Profile: server

Figure 25 - Adding a mapping for the Local Domain to the Kerberos Identity System

The setup is now complete. When a client sends a SOAP message with a WS-Security header with a SAML Token for an Active Directory user, the Network Director will authenticate the user and then perform constrained delegation as the identity in the Kerberos Identity System and generate a Kerberos ticket for the downstream service.