

# Leveraging JSON Web Tokens In IBM® Security Access Manager

—

## Identity and Access Management Technical Support Webinar

8 August 2019



# Announcing IBM VIP Rewards

Engage. Earn points. Get Rewards.

**IBM VIP Rewards is a way to engage with and recognize the ways that you, the client, add value to IBM.**

**Complete fun challenges and get rewarded for interacting with IBM, learning new technologies and sharing your knowledge.**



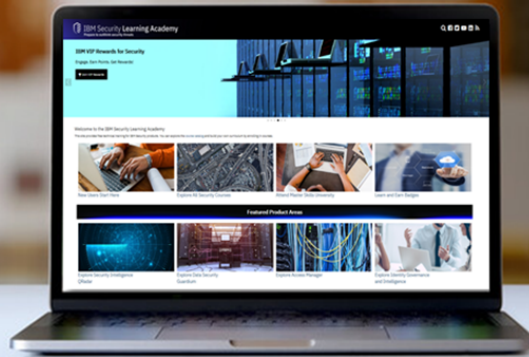
Learn more...  
[ibm.biz/vip-rewards](https://ibm.biz/vip-rewards)

Join IBM VIP Rewards for Security...  
[ibm.biz/JoinIBMVIPRewards-Security](https://ibm.biz/JoinIBMVIPRewards-Security)



IBM VIP Rewards for **Security**

# SecurityLearningAcademy.com



- Courses
- Videos
- Hands-on Labs
- Live Events
- Badges

Learning at no cost.

New content published daily.

# Panel

## Presenter

- Jack Yarborough – ISAM L2 Support

## Panelists

- Annelise Quap – ISAM L2 Support
- Nick Lloyd – ISAM L2 Support

# Goal of session

Understand the extensive ways a JSON Web Token can be utilized in the IBM® Security Access Manager ecosystem

# Agenda

- Overview of OIDC mapping rules and how to manipulate the ID Token claims
- Utilizing Attribute Sources to populate JWT claims
- Sending a JWT to a junction application using SSO Junctions and Trust Chains
- Accepting Authorization headers with JWT content to create an authenticated session
- OAUTH 2.0 JWT Bearer Profile Overview

# Overview of OIDC mapping rules and how to manipulate the ID Token claims

- Review of Mapping Rule Locations
- Support Published Technotes and Open Mic Resources

# Review of Mapping Rule Locations

The screenshot displays the IBM Security Access Manager interface. At the top, the navigation bar includes 'Home Appliance Dashboard', 'Monitor Analysis and Diagnostics', 'Secure Web Settings', 'Secure Access Control', and 'Secure Federation'. A circled '1' points to the 'Secure Federation' tab. Below this, the 'Manage' sidebar contains a tree view with 'Global Settings' and 'Mapping Rules' highlighted by a circled '2' and a circled '3' respectively. The main content area shows the 'Global Keys' section with a 'Mapping Rules' sub-section. A search filter 'oauth' is entered in a text box, and a list of mapping rules is displayed below, all categorized as 'OAUTH'.

Filter with 'oauth'

| Mapping Rules   |
|---|
| azncodeproviderPostTokenGeneration<br>Category: OAUTH |
| azncodeproviderPreTokenGeneration<br>Category: OAUTH  |
| mmfaPostTokenGeneration<br>Category: OAUTH            |
| mmfaPreTokenGeneration<br>Category: OAUTH             |



# Review of Mapping Rule Locations

Filter with  
'oauth'

The screenshot displays the IBM Security Access Manager console interface. The top navigation bar includes 'Home Appliance Dashboard', 'Monitor Analysis and Diagnostics', 'Secure Web Settings', 'Secure Access Control', and 'Secure Federation'. A blue box highlights 'Secure Access Control' with a circled '1' and an arrow pointing to it. Below this, the 'Manage' section is visible, with a blue box highlighting 'Global Settings' and a circled '2' with an arrow pointing to it. Under 'Global Settings', a blue box highlights 'Mapping Rules' with a circled '3' and an arrow pointing to it. On the left, the 'Mapping Rules' section is shown with a search filter 'oauth' in a blue box. Below the filter, four mapping rules are listed, all with 'Category: OAUTH':

- azncodeproviderPostTokenGeneration
- azncodeproviderPreTokenGeneration
- mmfaPostTokenGeneration
- mmfaPreTokenGeneration

A large grey lightning bolt graphic is overlaid on the left side of the console, pointing towards the search filter and the list of mapping rules.

# Review of Mapping Rule Locations

The screenshot displays the IBM Security Access Manager interface. The top navigation bar includes 'Home Appliance Dashboard', 'Monitor Analysis and Diagnostics', 'Secure Web Settings', and 'Secure Access Control' (highlighted with a blue box and labeled 1). The left sidebar contains a 'Policy' menu (labeled 2) with sub-items: Access Control, Authentication, Risk Profiles, Attributes, Obligations, 'OpenID Connect and API Protection' (labeled 3), Information Points, and Extensions. The 'OpenID Connect and API Protection' menu is expanded, showing 'Definitions Resources Clients Mapping Rules' (labeled 4). The 'Mapping Rules' sub-menu is selected, displaying a list of rules: 'azncodeproviderPostTokenGeneration' (Category: OAUTH), 'azncodeproviderPreTokenGeneration' (Category: OAUTH), 'mmfaPostTokenGeneration' (Category: OAUTH), and 'mmfaPreTokenGeneration' (Category: OAUTH). Above the list are action buttons: Add, Import, Edit, Delete, Export, and Replace.

# Support Published Technotes and Open Mic Resources

Fine-Tuning ID Tokens in ISAM Advanced Access Control for OIDC Flows:

<http://www.ibm.com/support/docview.wss?uid=ibm10878999>

- Covers Authorization Code Flow
- Covers Implicit Flow
- Covers Userinfo output for Authorization Code flow
- Covers Userinfo output for Implicit flow

STSEniversalUser Overview:

<http://www.ibm.com/support/docview.wss?uid=ibm10881007&aid=1>

# Utilizing Attribute Sources to populate JWT claims

- Attribute source location
- Creating an attribute source
- Attaching to an API protection
- Confirming attribute presence with trace logs
- Limiting attributes based on scope
- Testing the configuration

# Attribute source location

The screenshot displays the IBM Security Access Manager console interface. The top navigation bar includes the following sections:

- Home: Appliance Dashboard
- Monitor: Analysis and Diagnostics
- Secure: Web Settings
- Secure: Access Control
- Secure Federation (highlighted with a blue box and labeled '1')

The 'Manage' section (labeled '2') is expanded, showing the following sub-items:

- Federations
- Security Token Service
- Attribute Source (highlighted with a blue box and labeled '3')
- Grants
- OpenID Connect and API Protection
- Alias Service Settings

The 'Global Settings' section includes:

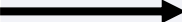
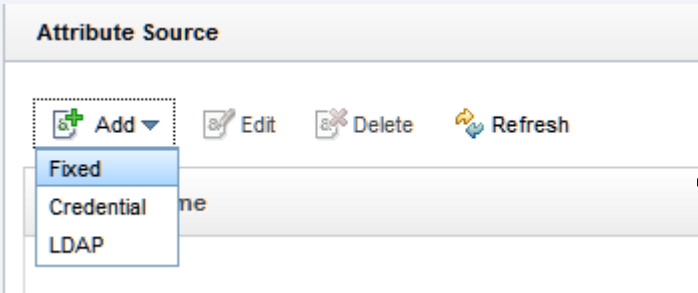
- Advanced Configuration
- User Registry
- Runtime Parameters
- Template Files
- Mapping Rules
- Distributed Session Cache
- Server Connections (highlighted with a grey box)
- Partner Templates
- Point of Contact
- Access Policies

The 'Global Keys' section includes:

- LTPA Keys
- Kerberos Keytab File

# Creating an attribute source

Attribute Sources can hold a fixed value



**Add Attribute Source**

---

Type: Fixed

Attribute Name:

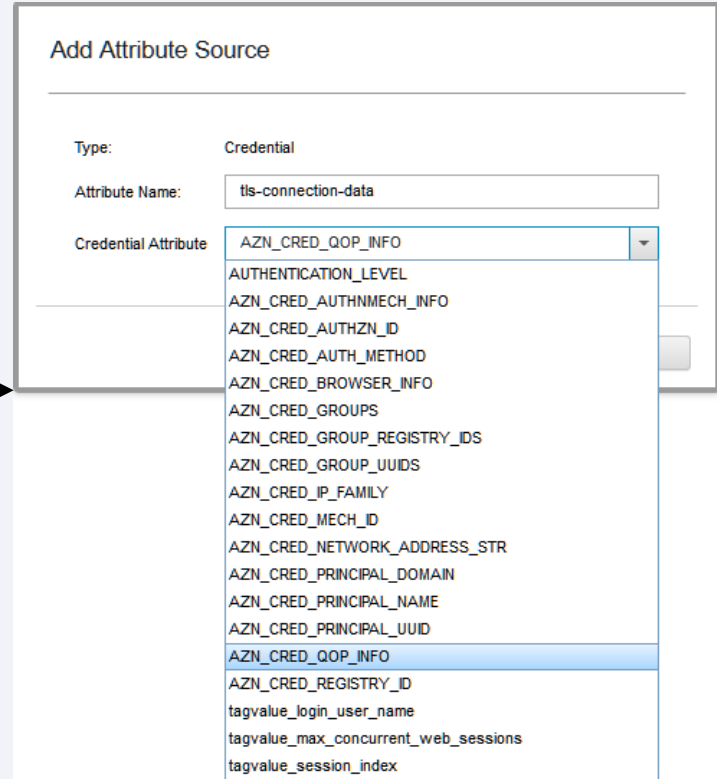
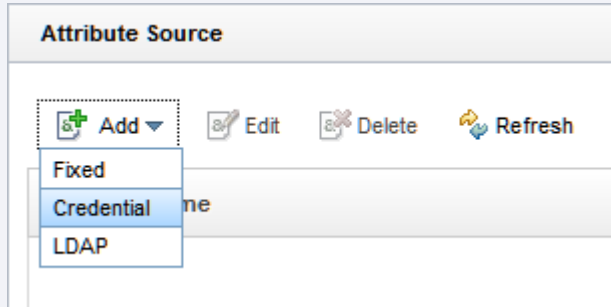
Value:

---

# Creating an attribute source

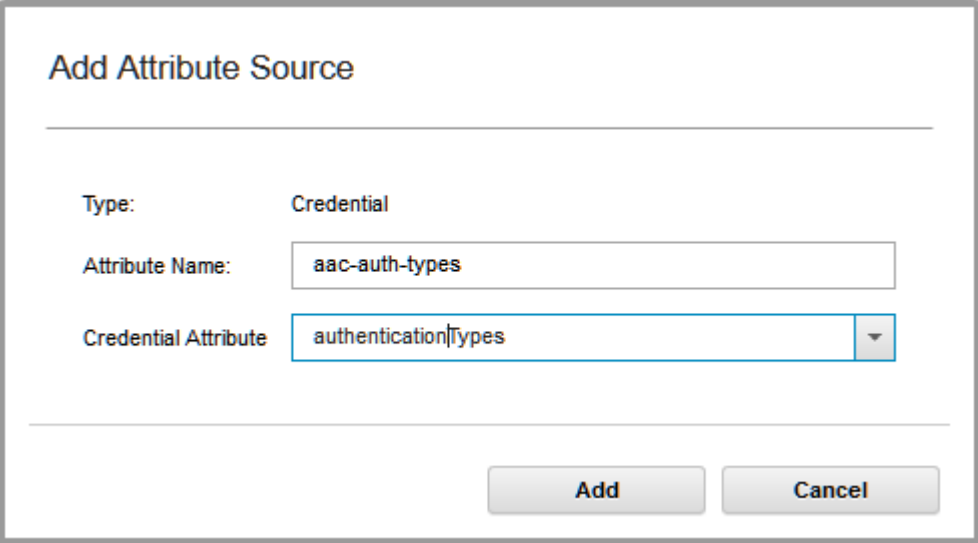
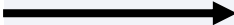
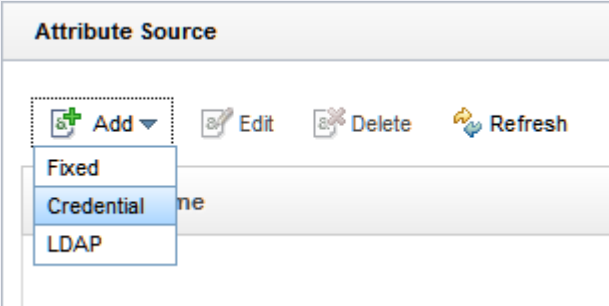
Attribute Sources can be associated with Reverse Proxy credential entries

We've mapped 'AZN\_CRED\_QOP\_INFO'  
to 'tls-connection-data'



# Creating an attribute source

You can also map custom credential attributes by manually specifying a value in the 'Credential Attribute' field



The image shows a dialog box titled "Add Attribute Source". It contains the following fields:

- Type: Credential
- Attribute Name: aac-auth-types
- Credential Attribute: authenticationTypes

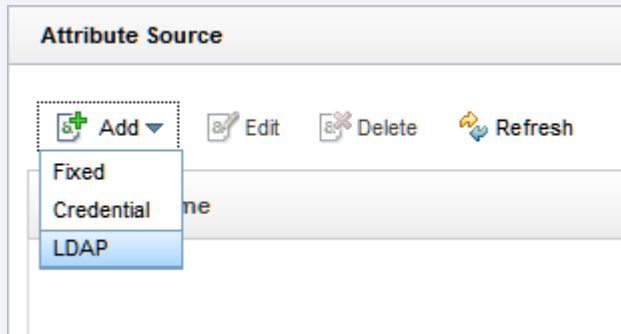
At the bottom of the dialog box are two buttons: "Add" and "Cancel".



# Creating an attribute source

Attribute Sources can retrieve values from LDAP servers as well.

This requires a predefined  
'Server Connection'.



### Add Attribute Source

---

Type: LDAP

Attribute Name:

LDAP Attribute:

Server Connection:

Scope:

Selector:

Search filter:

Base DN:

---

Documentation Reference :

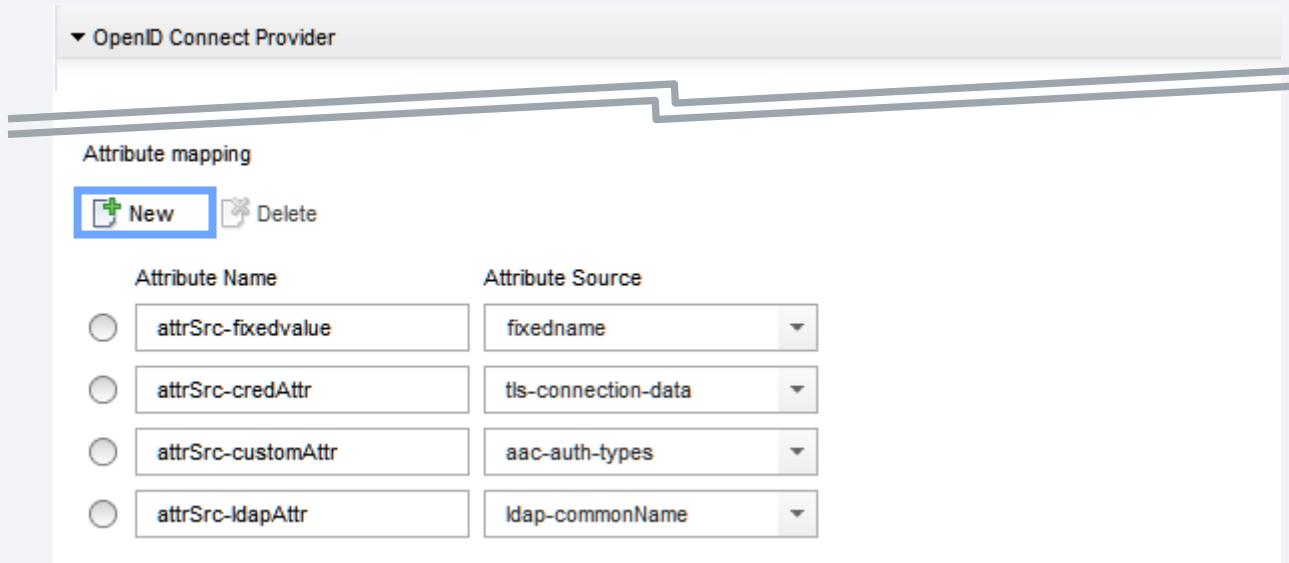
[https://www.ibm.com/support/knowledgecenter/en/SSPREK\\_9.0.7/com.ibm.isam.doc/admin/task/tsk\\_mng\\_attribute\\_sources.html](https://www.ibm.com/support/knowledgecenter/en/SSPREK_9.0.7/com.ibm.isam.doc/admin/task/tsk_mng_attribute_sources.html)

# Attaching to an API protection

The screenshot shows the IBM Security Access Manager console interface. The navigation bar at the top includes 'Home Appliance Dashboard', 'Monitor Analysis and Diagnostics', 'Secure Web Settings', and 'Secure Access Control' (highlighted with a blue box and labeled 1). Below this, the 'Policy' menu is highlighted with a blue box and labeled 2. The 'OpenID Connect and API Protection' option is selected in the left sidebar, highlighted with a blue box and labeled 3. The main content area shows 'OpenID Connect and API Protection' with sub-tabs for 'Definitions', 'Resources', 'Clients', and 'Mapping Rules'. The 'Definitions' tab is active and highlighted with a blue box and labeled 4. An 'Edit Definition' button is highlighted with a blue box and labeled 6. The 'azncodeprovider' definition is highlighted with a blue box and labeled 5. Below the definition name, the text 'mmfa' is visible.

# Attaching to an API protection

The 'Attribute Name' will be the name referenced in the mapping rules



▼ OpenID Connect Provider

Attribute mapping

| Attribute Name                           | Attribute Source    |
|--|---------------------|
| <input type="radio"/> attrSrc-fixedvalue | fixedname           |
| <input type="radio"/> attrSrc-credAttr   | tls-connection-data |
| <input type="radio"/> attrSrc-customAttr | aac-auth-types      |
| <input type="radio"/> attrSrc-ldapAttr   | ldap-commonName     |

# Confirming attribute presence with trace logs

## Example mapping rule syntax:

```
var fixedAttrSrcAttr = stsuu.getAttributeContainer().getAttributeValueByNameAndType("attrSrc-fixedvalue", "urn:ibm:names:ITFIM:5.1:accessmanager");
IDMappingExtUtils.traceString("Fixed Attribute Source Attribute Value: [" + fixedAttrSrcAttr + "]);

var credAttrSrcAttr = stsuu.getAttributeContainer().getAttributeValueByNameAndType("attrSrc-credAttr", "urn:ibm:names:ITFIM:5.1:accessmanager");
IDMappingExtUtils.traceString("Credential Attribute Source Attribute Value: [" + credAttrSrcAttr + "]);

var customCredAttrSrcAttr = stsuu.getAttributeContainer().getAttributeValueByNameAndType("attrSrc-customAttr", "urn:ibm:names:ITFIM:5.1:accessmanager");
IDMappingExtUtils.traceString("Custom Credential Attribute Source Attribute Value: [" + customCredAttrSrcAttr + "]);

var ldapAttrSrcAttr = stsuu.getAttributeContainer().getAttributeValueByNameAndType("attrSrc-ldapAttr", "urn:ibm:names:ITFIM:5.1:accessmanager");
IDMappingExtUtils.traceString("LDAP Attribute Source Attribute Value: [" + ldapAttrSrcAttr + "]);
```

## Trace Specification:

```
com.tivoli.am.fim.trustserver.sts.utilities.IDMappingExtUtils.*=ALL
```

# Confirming attribute presence with trace logs

## Example STSUU contents:

```
<stsuser:AttributeList>
...
  <stsuser:Attribute name="attrSrc-credAttr" type="urn:ibm:names:ITFIM:5.1:accessmanager">
    <stsuser:Value>SSK: TLSV12: 9C</stsuser:Value>
  </stsuser:Attribute>
...
  <stsuser:Attribute name="attrSrc-fixedvalue" type="urn:ibm:names:ITFIM:5.1:accessmanager">
    <stsuser:Value>fixedvalue</stsuser:Value>
  </stsuser:Attribute>
...
  <stsuser:Attribute name="attrSrc-customAttr" type="urn:ibm:names:ITFIM:5.1:accessmanager"/>
...
  <stsuser:Attribute name="attrSrc-ldapAttr" type="urn:ibm:names:ITFIM:5.1:accessmanager">
    <stsuser:Value>Joseph User</stsuser:Value>
  </stsuser:Attribute>
```

# Confirming attribute presence with trace logs

## Trace Output:

```
[8/2/19 3:37:05:376 CDT] 00000426 id=00000000 om.tivoli.am.fim.trustserver.sts.utilities.IDMappingExtUtils >
traceString ENTRY Fixed Attribute Source Attribute Value: [fixedvalue]
[8/2/19 3:37:05:376 CDT] 00000426 id=00000000 om.tivoli.am.fim.trustserver.sts.utilities.IDMappingExtUtils <
traceString RETURN
[8/2/19 3:37:05:376 CDT] 00000426 id=00000000 om.tivoli.am.fim.trustserver.sts.utilities.IDMappingExtUtils >
traceString ENTRY Credential Attribute Source Attribute Value: [SSK: TLSV12: 9C]
[8/2/19 3:37:05:376 CDT] 00000426 id=00000000 om.tivoli.am.fim.trustserver.sts.utilities.IDMappingExtUtils <
traceString RETURN
[8/2/19 3:37:05:377 CDT] 00000426 id=00000000 om.tivoli.am.fim.trustserver.sts.utilities.IDMappingExtUtils >
traceString ENTRY Custom Credential Attribute Source Attribute Value: [null]
[8/2/19 3:37:05:377 CDT] 00000426 id=00000000 om.tivoli.am.fim.trustserver.sts.utilities.IDMappingExtUtils <
traceString RETURN
[8/2/19 3:37:05:377 CDT] 00000426 id=00000000 om.tivoli.am.fim.trustserver.sts.utilities.IDMappingExtUtils >
traceString ENTRY LDAP Attribute Source Attribute Value: [Joseph User]
[8/2/19 3:37:05:377 CDT] 00000426 id=00000000 om.tivoli.am.fim.trustserver.sts.utilities.IDMappingExtUtils <
traceString RETURN
```

# Limiting attributes based on scope

You can extend the '*definitionPreTokenGeneration*' mapping rule logic block on lines 743-750 to extend scope functionality.

## Example code:

```
if (temp_attr.getValues()[scope].includes("fixed") {
    is_fixed_scope = true;
}
if (temp_attr.getValues()[scope].includes("credential") {
    is_cred_scope = true;
}
if (temp_attr.getValues()[scope].includes("custom") {
    is_custom_scope = true;
}
if (temp_attr.getValues()[scope].includes("ldap") {
    is_ldap_scope = true;
}
if (temp_attr.getValues()[scope].includes("all") {
    is_all_scope = true;
}
```

# Limiting attributes based on scope

You would then add logic in the `'definitionPreTokenGeneration'` mapping rule to the `'if (populate_id_token || save_cred_attrs) {'` block to extend the scope functionality.

Example code snippet of 'all' scope logic:

```
if(is_all_scope) {
  if(fixedAttrSrcAttr !=null && fixedAttrSrcAttr != "") {
    stsuu.addAttribute(new com.tivoli.am.fim.trustserver.sts.uuser.Attribute("fixedAttribute", "urn:ibm:jwt:claim",
fixedAttrSrcAttr));
  } else {
    stsuu.addAttribute(new com.tivoli.am.fim.trustserver.sts.uuser.Attribute("fixedAttribute", "urn:ibm:jwt:claim",
"missing"));
  }
}
...
```

Full example JavaScript file located at :

[https://github.com/IBM-Security/isam-support/blob/master/config-example/aac/oauth\\_js/oidc/implicit/oauth-oidc-implicit-preTokenGeneration-attributeSource-with-scope.js](https://github.com/IBM-Security/isam-support/blob/master/config-example/aac/oauth_js/oidc/implicit/oauth-oidc-implicit-preTokenGeneration-attributeSource-with-scope.js)



# Testing the configuration

My test requested used an 'Implicit' flow for simplicity. Scope of 'all'

[https://isam9070.hyperv.lab/mga/sps/oauth/oauth20/authorize?client\\_id=implicit\\_client&scope=openid%20all&response\\_type=id\\_token&redirect\\_uri=https://jwt.io&nonce=blah&state=blah](https://isam9070.hyperv.lab/mga/sps/oauth/oauth20/authorize?client_id=implicit_client&scope=openid%20all&response_type=id_token&redirect_uri=https://jwt.io&nonce=blah&state=blah)

PAYLOAD: DATA

```
{
  "customAttribute": "missing",
  "nonce": "blah",
  "credentialAttribute": "SSK: TLSV12: 9C",
  "iat": 1564737674,
  "iss": "https://isam9070.hyperv.lab/oidc/implicit",
  "fixedAttribute": "fixedvalue",
  "sub": "juser",
  "exp": 1564741814,
  "ldapAttribute": "Joseph User",
  "aud": "implicit_client"
}
```

# Testing the configuration

My test requested used an 'Implicit' flow for simplicity. Scope of 'ldap'

[https://isam9070.hyperv.lab/mga/sps/oauth/oauth20/authorize?client\\_id=implicit\\_client&scope=openid%20ldap&response\\_type=id\\_token&redirect\\_uri=https://jwt.io&nonce=blah&state=blah](https://isam9070.hyperv.lab/mga/sps/oauth/oauth20/authorize?client_id=implicit_client&scope=openid%20ldap&response_type=id_token&redirect_uri=https://jwt.io&nonce=blah&state=blah)

PAYLOAD: DATA

```
{
  "nonce": "blah",
  "iat": 1564738326,
  "iss": "https://isam9070.hyperv.lab/oidc/implicit",
  "sub": "juser",
  "exp": 1564742466,
  "ldapAttribute": "Joseph User",
  "aud": "implicit_client"
}
```

# Testing the configuration

My test requested used an 'Implicit' flow for simplicity. Scope of 'ldap' and 'fixed'

[https://isam9070.hyperv.lab/mga/sps/oauth/oauth20/authorize?client\\_id=implicit\\_client&scope=openid%20fixed%20ldap&response\\_type=id\\_token&redirect\\_uri=https://jwt.io&nonce=blah&state=blah](https://isam9070.hyperv.lab/mga/sps/oauth/oauth20/authorize?client_id=implicit_client&scope=openid%20fixed%20ldap&response_type=id_token&redirect_uri=https://jwt.io&nonce=blah&state=blah)

PAYLOAD: DATA

```
{
  "nonce": "blah",
  "iat": 1564741829,
  "iss": "https://isam9070.hyperv.lab/oidc/implicit",
  "fixedAttribute": "fixedvalue",
  "sub": "juser",
  "exp": 1564745969,
  "ldapAttribute": "Joseph User",
  "aud": "implicit_client"
}
```

# Sending a JWT to a junction application using SSO Junctions and Trust Chains

- Documentation Reference
- Creating an STS Module Template
- Creating an STS Chain
- Configuring the SSO Junction
- Editing the Reverse Proxy Configuration File
- Testing the Configuration

# Documentation Reference

Reverse Proxy SSO Junction related documentation:

## [Single sign-on with the Security Token Service](#)

- The documentation defines how the Trust Service Chain should be configured

## [Stanza Reference: \[tfimssso:/junction\]](#)

- Stanza reference for the [tfimssso] stanza

## [Stanza Reference: \[tfim-cluster:cluster\]](#)

- Stanza reference for the SOAP call to the STS cluster

# Creating an STS Module Template

Navigate to the 'Security Token Service' Menu

The screenshot displays the IBM Security Access Manager console interface. The top navigation bar includes the following items:

- Home Appliance Dashboard
- Monitor Analysis and Diagnostics
- Secure Web Settings
- Secure Access Control
- Secure Federation (highlighted with a blue box and labeled with a circled '1')

The 'Manage' section is highlighted with a blue box and labeled with a circled '2'. It contains the following sub-items:

- Federations
- Security Token Service (highlighted with a blue box and labeled with a circled '3')
- Attribute Source
- Grants
- OpenID Connect and API Protection
- Alias Service Settings

The 'Global Settings' section includes:

- Advanced Configuration
- User Registry
- Runtime Parameters
- Template Files
- Mapping Rules
- Distributed Session Cache
- Server Connections
- Partner Templates
- Point of Contact
- Access Policies

The 'Global Keys' section includes:

- LTPA Keys
- Kerberos Keytab File

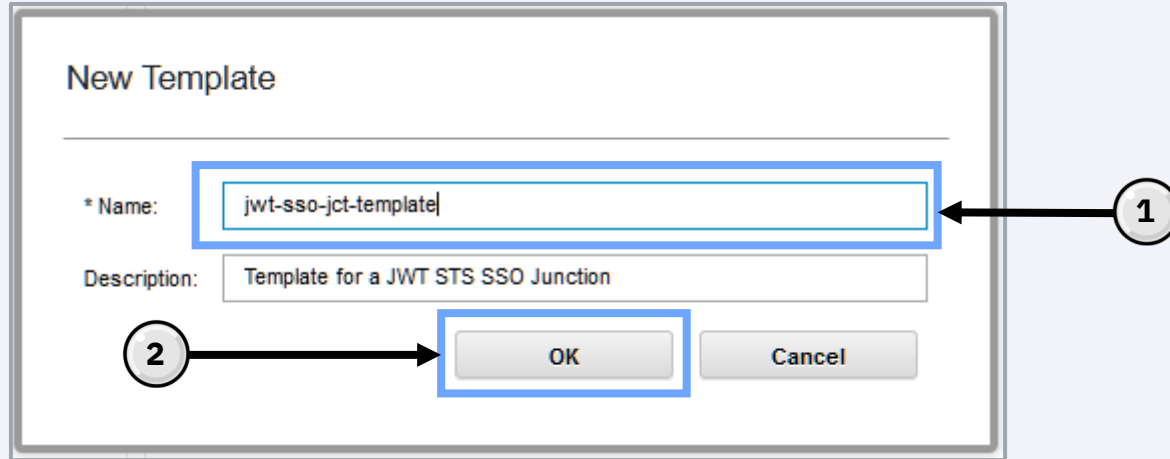
# Creating an STS Module Template

Navigate to the 'Templates' sub menu and 'Add' a new Template

The screenshot displays the IBM Security Access Manager interface. At the top, there is a navigation bar with the title "IBM Security Access Manager" and several menu items: "Home Appliance Dashboard", "Monitor Analysis and Diagnostics", "Secure Web Settings", "Secure Access Control", and "Secure Federation". Below this, a secondary navigation bar contains "Security Token Service", "Module Chains", "Templates", and "Modules". The "Templates" link is highlighted with a blue box and a black arrow labeled "1" points to it from the right. Below the navigation bar, there is a section for "Templates" with three buttons: "Add", "Edit", and "Delete". The "Add" button is highlighted with a blue box and a black arrow labeled "2" points to it from the left. To the right of the buttons is a "Filter" input field. Below the buttons, the text "Templates" is displayed, followed by "No items to display".

# Creating an STS Module Template

Add a 'Name' and 'Description' for your Template



The image shows a 'New Template' dialog box with the following fields and buttons:

- \* Name:** A text input field containing 'jwt-sso-jct-template'. This field is highlighted with a blue border and has a circled '1' with an arrow pointing to it from the right.
- Description:** A text input field containing 'Template for a JWT STS SSO Junction'.
- Buttons:** 'OK' and 'Cancel' buttons are located at the bottom. The 'OK' button is highlighted with a blue border and has a circled '2' with an arrow pointing to it from the left.

Select 'OK' to create the Template



# Creating an STS Module Template

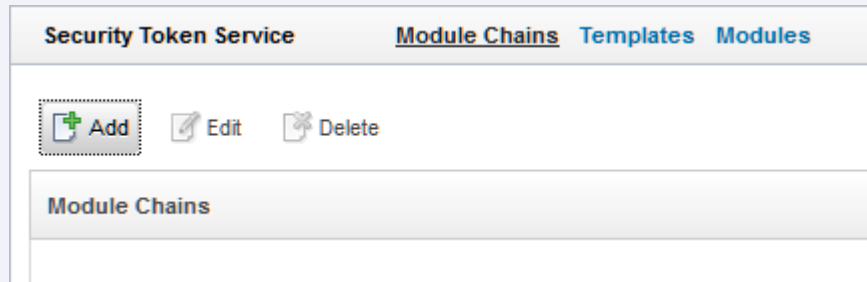
Select your template, add modules, and specify the 'Mode' of operation

The screenshot displays the Security Token Service (STS) interface. At the top, there are navigation tabs for 'Module Chains', 'Templates', and 'Modules'. Below the navigation, there are icons for 'Add', 'Edit', and 'Delete'. A 'Filter' input field is present, with a circled '2' pointing to it. To the right of the filter are icons for 'Add', 'Delete', 'Move Up', and 'Move Down'. The main area is divided into two panes: 'Templates' on the left and 'Template Contents' on the right. In the 'Templates' pane, the 'jwt-ss-jct-template' is selected, with a circled '1' pointing to it. The 'Template Contents' pane shows the current template's configuration: 'Default IVcred Token', 'Default IV Credential Token Instance', and 'Mode: Validate'. An 'Add to Template' dialog box is open in the foreground. It has a title 'Add to Template' and a horizontal line. Below the line, there are two dropdown menus. The first is labeled 'Module Instance' and has 'Default Jwt Module' selected, with a circled '3' pointing to it. The second is labeled 'Mode' and has 'Issue' selected, with a circled '4' pointing to it. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

# Creating an STS Chain

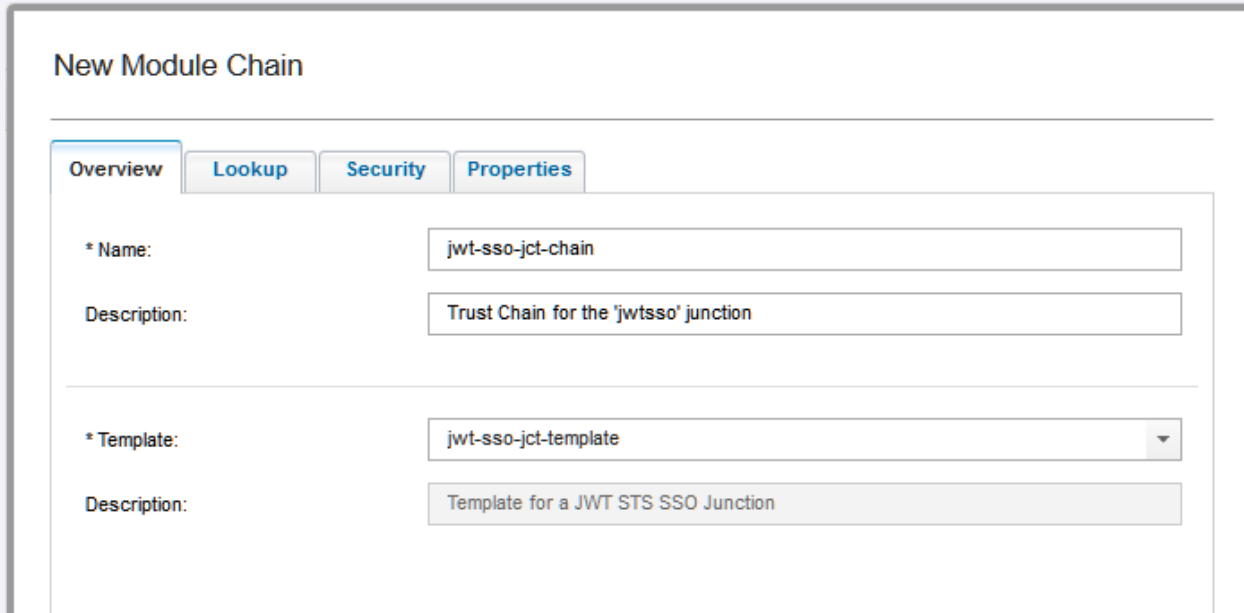
Navigate to the 'Module Chains' sub menu

Select the 'Add' button to create a new STS Chain



# Creating an STS Chain

Use a descriptive naming convention for the 'Name' and provide an optional 'Description'. Select the 'Template' that we created earlier for use.



The screenshot shows a web form titled "New Module Chain". At the top, there are four tabs: "Overview" (selected), "Lookup", "Security", and "Properties". Below the tabs, there are four input fields:

- \* Name:** A text input field containing "jwt-ssso-jct-chain".
- Description:** A text input field containing "Trust Chain for the 'jwtssso' junction".
- \* Template:** A dropdown menu with "jwt-ssso-jct-template" selected.
- Description:** A text input field containing "Template for a JWT STS SSO Junction".

# Creating an STS Chain

The Reverse Proxy client uses WS-Trust 1.3 so the 'Issue (Oasis)' 'Request type' is mandatory.

Customize the 'Applies to' 'Address' for this specific junction.

The 'Issuer' 'Address' is per the documentation.

Choose the 'JWT' token type.

New Module Chain

Overview Lookup Security Properties

\* Request Type: Issue (Oasis)

\* URI: http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue

Lookup Type:  Traditional WS-Trust Elements  XPath

**Applies to**

\* Address: https://isam9070.hyperv.lab/jwtss0

Service Name: :

Port Type: :

**Issuer**

\* Address: amwebrite-sts-client

Service Name: :

Port Type: :

Token Type: JWT

\* URI: urn:ietf:params:oauth:token-type:jwt

# Creating an STS Chain

The only properties that need to be edited are related to the 'Default Jwt Module'

From the 'Properties' tab *optionally* select a 'Signature algorithm', 'Signing shared symmetric key', and when applicable a 'Certificate Database' and 'Certificate Label'

The screenshot shows the 'New Module Chain' configuration interface. The 'Properties' tab is selected and highlighted with a blue border. On the left, under 'Template Contents', the 'Default Jwt Module' entry is highlighted with a blue border. On the right, the 'Default Jwt Module (Issue)' section contains several configuration fields, all of which are highlighted with blue borders: 'Signature algorithm' (set to RS256), 'Signing shared symmetric key' (empty text input), 'Certificate Database' (set to rt\_profile\_keys), and 'Certificate Label' (set to jwtssso-signing-key).

# Creating an STS Chain

Encryption is optional.

I won't be encrypting for my example.

Here we have the 'Claims configuration' where the JWT 'Issuer', 'Subject', 'Audience', 'Expiration' and other JWT related attributes can be specified.

The screenshot displays the 'New Module Chain' configuration window. The 'Properties' tab is selected, showing the 'Claim configuration' section. The 'Default Jwt Module' is highlighted in the left pane. The 'Claim configuration' section includes the following settings:

- Value of 'iss' to include: `https://isam9070.hyperv.lab/jw`
- Value of 'sub' to include: `{AZN_CRED_PRINCIPAL_NAME}`
- Value of 'aud' to include: (empty field)
- Include 'exp' claim
- JWT lifetime: 3600

# Editing the Reverse Proxy Configuration File

The Reverse Proxy configuration file needs to be updated before the junction can be created.

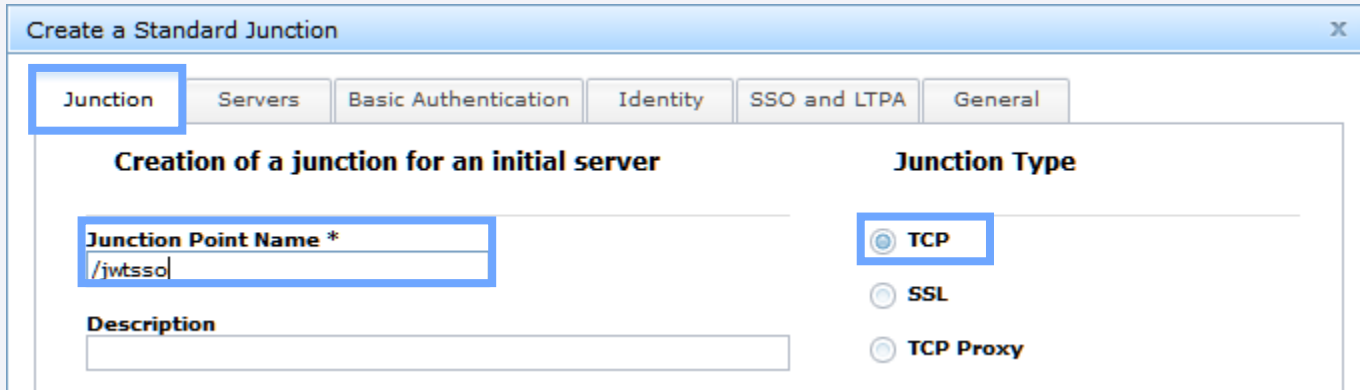
Here are example proxy configuration entries that are required for the SSO junction to work :

```
[tfimssso:/jwtssso]
always-send-tokens = true
applies-to = https://isam9070.hyperv.lab/jwtssso
one-time-token = true
preserve-xml-token = false
token-collection-size = 1
renewal-window = 15
token-type = urn:ietf:params:oauth:token-type:jwt
token-transmit-type = header
token-transmit-name = jwt-authorization
tfim-cluster-name = isam-federation

[tfim-cluster:isam-federation]
server = 9,https://localhost/TrustServerWST13/services/RequestSecurityToken
timeout = 20
handle-pool-size = 10
handle-idle-timeout = 10
basic-auth-user = easuser
basic-auth-passwd = passw0rd
ssl-keyfile = pdsrv.kdb
ssl-keyfile-stash = pdsrv.sth
```

# Configuring the SSO Junction

Create a junction for this JWT SSO solution. Our example will be a 'Standard' type junction making a 'TCP' connection named '/jwtssso'

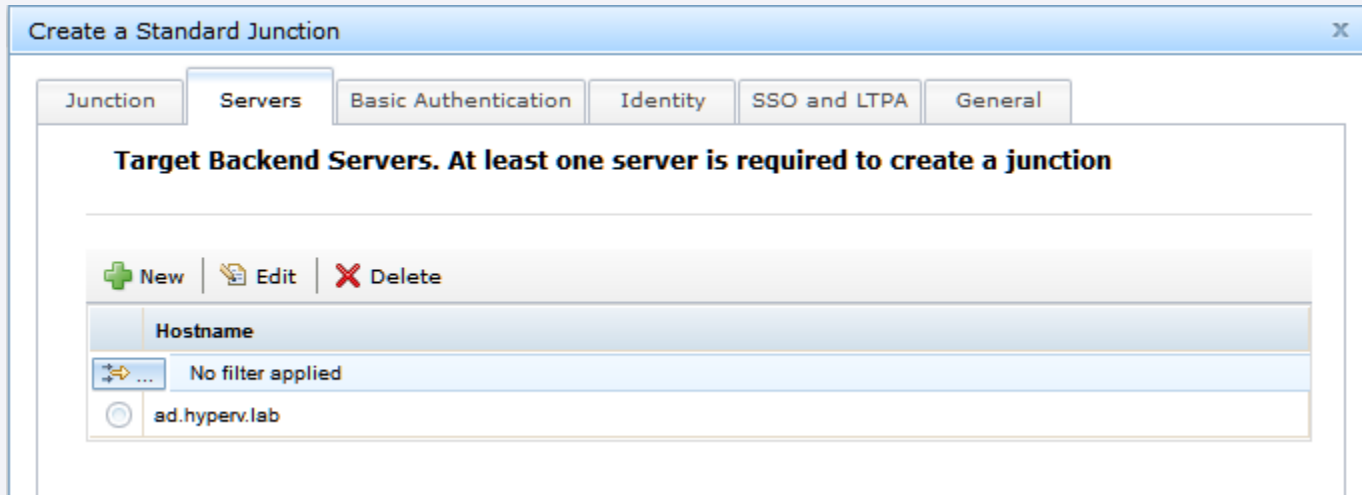


The screenshot shows a dialog box titled "Create a Standard Junction" with a close button (X) in the top right corner. Below the title bar are several tabs: "Junction", "Servers", "Basic Authentication", "Identity", "SSO and LTPA", and "General". The "Junction" tab is currently selected and highlighted with a blue border. The main content area is divided into two sections. The left section, titled "Creation of a junction for an initial server", contains a text input field labeled "Junction Point Name \*" with the value "/jwtssso" entered. Below it is a "Description" field which is currently empty. The right section, titled "Junction Type", contains three radio button options: "TCP" (which is selected and highlighted with a blue border), "SSL", and "TCP Proxy".



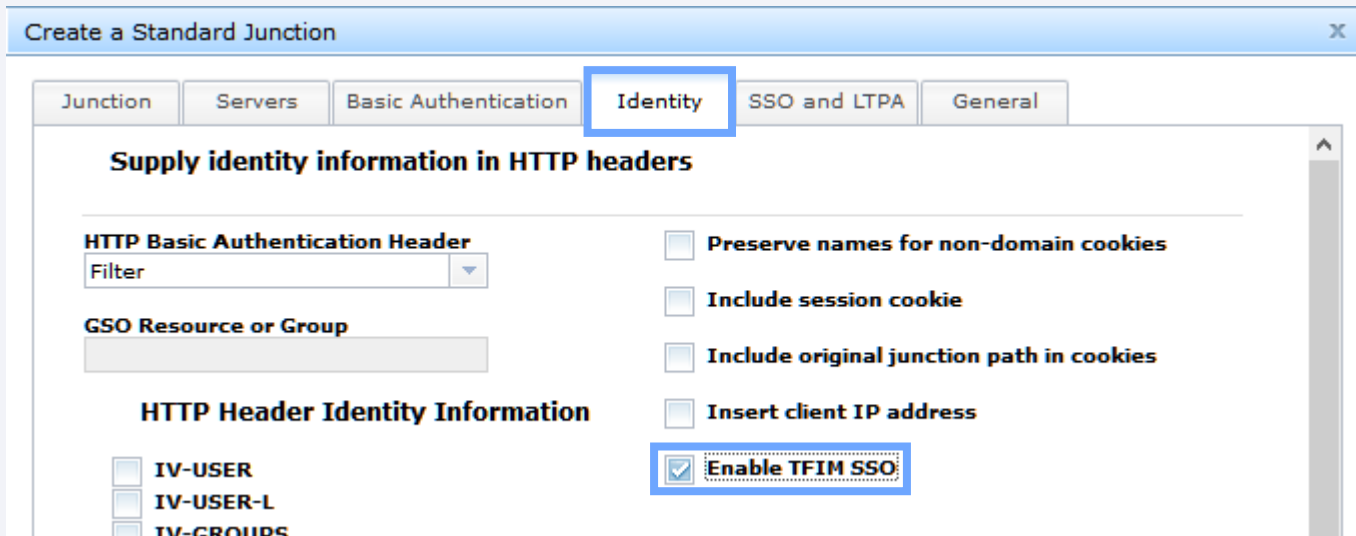
# Configuring the SSO Junction

On the 'Servers' sub menu add the 'Target Backend Server(s)'. Here you can add multiple servers for failover if necessary.



# Configuring the SSO Junction

On the 'Identity' sub menu be sure to select 'Enable TFIM SSO', this is critical to the operation of the STS SSO Junction.



The screenshot shows the 'Create a Standard Junction' configuration window with the 'Identity' tab selected. The window title is 'Create a Standard Junction' with a close button (X) in the top right corner. The tabs are 'Junction', 'Servers', 'Basic Authentication', 'Identity', 'SSO and LTPA', and 'General'. The 'Identity' tab is highlighted with a blue border. The main content area is titled 'Supply identity information in HTTP headers'. It contains several sections: 'HTTP Basic Authentication Header' with a 'Filter' dropdown menu; 'GSO Resource or Group' with an empty text input field; 'HTTP Header Identity Information' with three checkboxes: 'IV-USER', 'IV-USER-L', and 'IV-GROUPS'; and a list of options on the right: 'Preserve names for non-domain cookies', 'Include session cookie', 'Include original junction path in cookies', 'Insert client IP address', and 'Enable TFIM SSO'. The 'Enable TFIM SSO' checkbox is checked and highlighted with a blue border.

At this point we can 'Save' the junction.



# Testing the Configuration

Partial Output from 'https://jwt.io'

PAYLOAD: DATA

```
{
  "emailAddress": "juser@hyperv.lab",
  "AZN_CRED_AUTH_METHOD": "failover-password",
  "tagvalue_user_session_id":
  "aXNhbTkwNzBsbWkuaHlwZXJ2LmxhYi1kZWZhdWx0AA==_XURPMwAAAAIA
  AAawM09EXcjLG9xKfwAAQUNKVE13TWRyOTJoMVhGTepCSThKME5LQzBiNF
  JRdVpzVnZ6Z2hiNzgwTkdr01C:default",
  "AZN_CRED_PRINCIPAL_UUID": "4e7d0eb2-aaba-11e9-8354-
  00155de0219f",
  "AZN_CRED_QOP_INFO": "SSK: TLSV12: 9C",
  "AZN_CRED_PRINCIPAL_DOMAIN": "Default",
  "AUTHENTICATION_LEVEL": "0",
```



# Accepting Authorization headers with JWT content to create an authenticated session

- OAuth: JWT as an Access Token on ISAM
- Supporting both JWT and OAUTH tokens
- Creating the Trust Chain Templates
- Creating the Trust Chains
- Relevant Reverse Proxy settings
- Testing the configuration

# OAuth: JWT as an Access Token on ISAM

Our esteemed Developer Leo Farrell has published the following post to consume JWT and make an ISAM Session:

<https://www.ibm.com/blogs/security-identity-access/oauth-jwt-access-token/>

Cons :

- Only allows for either JWT or OAUTH tokens

Let's improve on the groundwork laid here by supporting both JWT and OAUTH access tokens via a bit more customization.

# Supporting both JWT and OAUTH tokens

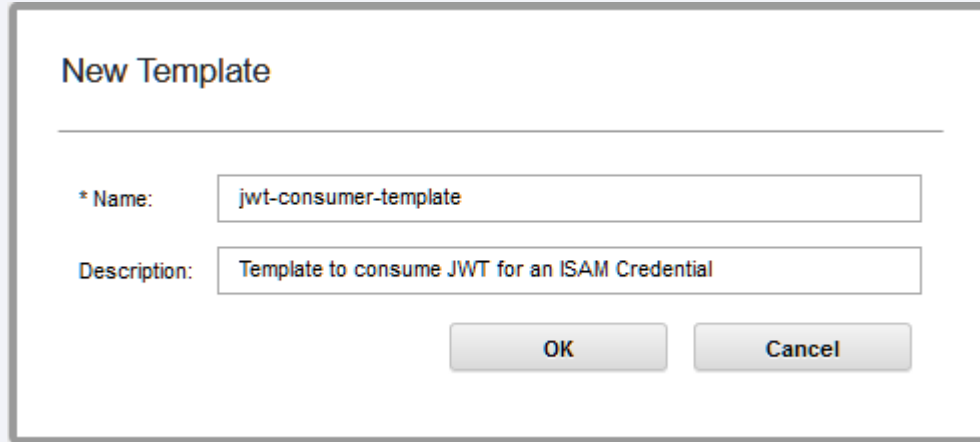
A high level overview of what will be accomplished:

- Create an STS Chain to decide which sub chain to use
  - This allows us to support both JWT and OAUTH tokens
- Create the JWT related chains



# Creating the Trust Chain Templates

We need to make a new Trust Chain template to consume JWT and provide an STS UU XML document back. Provide a 'Name' and 'Description' for the template:



New Template

---

\* Name:

Description:

OK Cancel

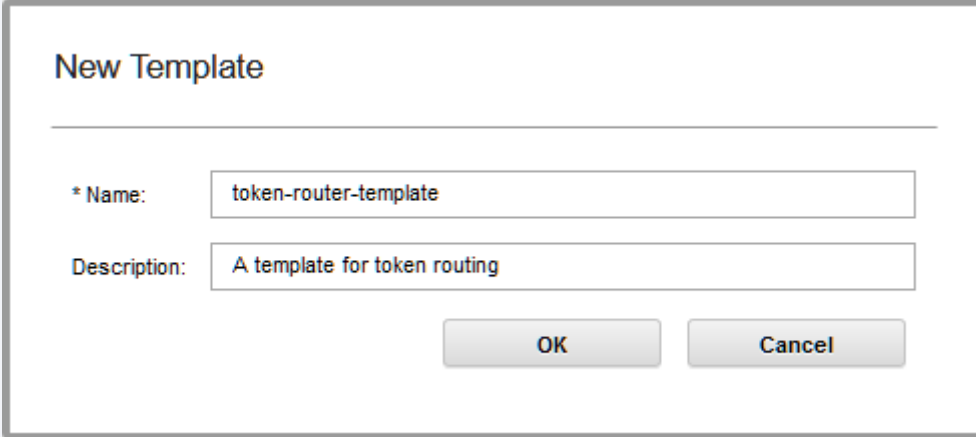
# Creating the Trust Chain Templates

Add the following modules in the specified modes:

The screenshot shows the 'Security Token Service' console with the 'Templates' tab selected. The interface is divided into two main panels: 'Templates' on the left and 'Template Contents' on the right. The 'Templates' panel lists two templates: 'jwt-consumer-template' (Template to consume JWT for an ISAM credential) and 'jwt-sso-jct-template' (Template for a JWT STS SSO Junction). The 'Template Contents' panel shows the configuration for the selected 'jwt-consumer-template', listing 'Default Jwt Module' (Default Jwt Module Instance, Mode: Validate) and 'Default STSUU' (Default STSUU Module Instance, Mode: Issue). The top navigation bar includes 'Security Token Service', 'Module Chains', 'Templates', and 'Modules'. Below the navigation bar, there are action buttons for 'Add', 'Edit', and 'Delete', a 'Filter' input field, and a 'Share' icon. The 'Add' button in the 'Template Contents' section is highlighted with a dashed border.

# Creating the Trust Chain Templates

Make a new Trust Chain template to route between the different Token validators and provide back an STSUU. Provide a 'Name' and 'Description' for the template:



New Template

---

\* Name:

Description:

OK Cancel

# Creating the Trust Chain Templates

Add the following modules in the specified modes:

The screenshot shows the Security Token Service console interface. At the top, there are navigation tabs: "Security Token Service", "Module Chains", "Templates" (which is selected), and "Modules". Below the tabs, there is a toolbar with icons for "Add", "Edit", and "Delete", followed by a "Filter" input field and a refresh icon. To the right of the filter are more icons for "Add", "Delete", "Move Up", and "Move Down".

The main content area is divided into two panes. The left pane is titled "Templates" and contains a list of three templates:

- jwt-consumer-template**: Template to consume JWT for an ISAM credential
- jwt-sso-jct-template**: Template for a JWT STS SSO Junction
- token-router-template**: A template for token routing (This template is highlighted in light blue)

The right pane is titled "Template Contents" and shows the details for the selected "token-router-template":

- Default Map Module**: Default Javascript Mapping Module Instance
- Mode: Map

# Creating the Trust Chains

Add a new chain and specify the 'Overview' properties:

## New Module Chain

---

**Overview**   **Lookup**   **Security**   **Properties**

\* Name:

Description:

---

\* Template:

Description:

# Creating the Trust Chains

Specify the 'Lookup' properties:

For our example we'll use the 'Implicit' OIDC Provider from earlier to create the JWT for consumption.

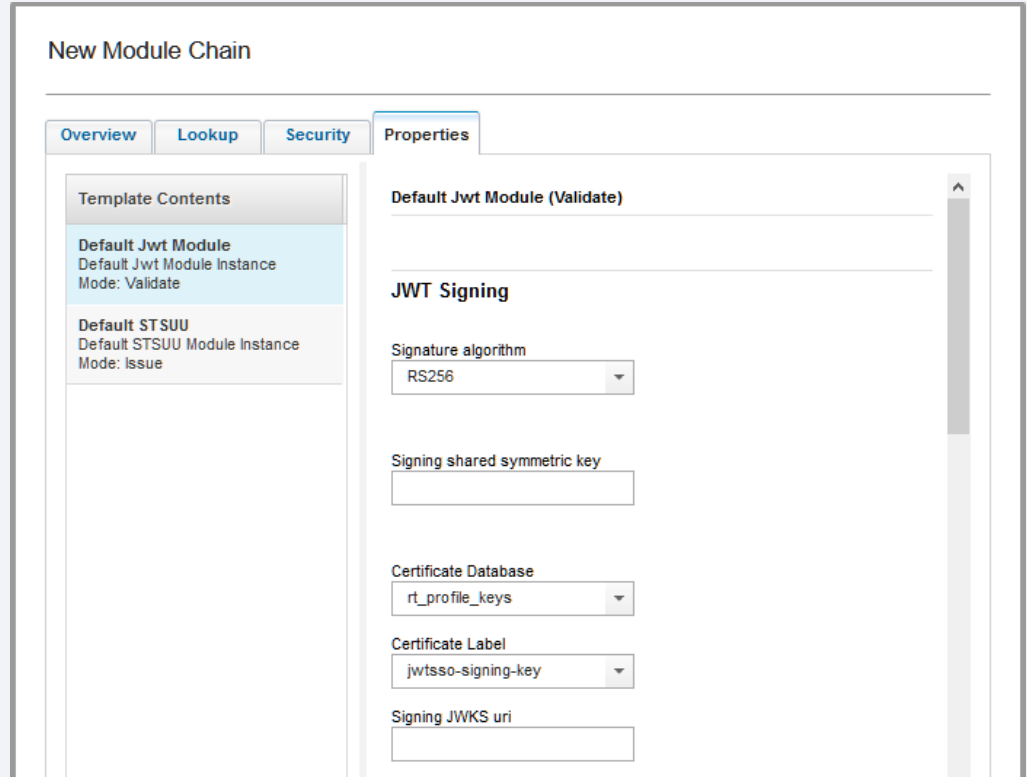
The screenshot shows the 'New Module Chain' configuration window with the 'Lookup' tab selected. The configuration includes the following fields and options:

- \* Request Type:** A dropdown menu with 'Validate' selected.
- \* URI:** A text field containing 'http://schemas.xmlsoap.org/ws/2005/02/trust/Validate'.
- Lookup Type:** Two radio button options: 'Traditional WS-Trust Elements' (selected) and 'XPath'.
- Applies to:**
  - \* Address:** A text field containing an asterisk (\*).
  - Service Name:** A text field.
  - Port Type:** A text field.
- Issuer:**
  - \* Address:** A text field containing 'https://isam9070.hyperv.lab/oidc/implicit'.

# Creating the Trust Chains

Specify the module chain 'Properties'.

Match the 'Implicit' provider properties of the outgoing JWT.

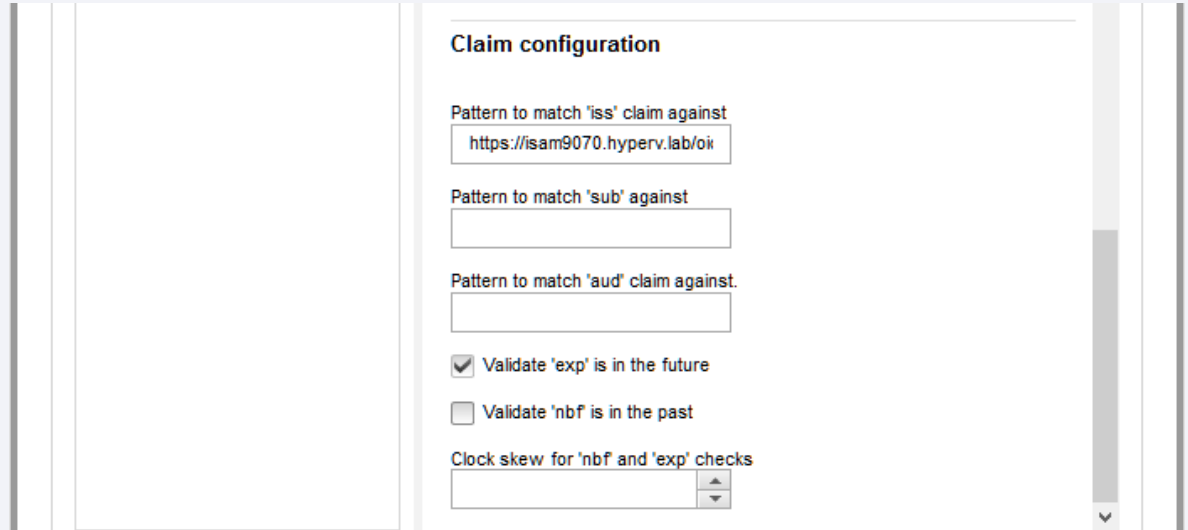


The screenshot displays the 'New Module Chain' configuration interface. It features a navigation bar with tabs for 'Overview', 'Lookup', 'Security', and 'Properties'. The 'Properties' tab is active, showing a list of 'Template Contents' on the left and configuration options on the right. The 'Template Contents' list includes 'Default Jwt Module' (Mode: Validate) and 'Default STSUU' (Mode: Issue). The configuration options on the right include 'Default Jwt Module (Validate)', 'JWT Signing' (with a dropdown for 'Signature algorithm' set to 'RS256'), 'Signing shared symmetric key' (text input), 'Certificate Database' (dropdown set to 'rt\_profile\_keys'), 'Certificate Label' (dropdown set to 'jwtso-signing-key'), and 'Signing JWKS uri' (text input).

# Creating the Trust Chains

Continue with the  
'Claims  
Configuration'  
properties.

We'll explicitly validate  
'Implicit' provider  
created JWT with this  
chain.



**Claim configuration**

Pattern to match 'iss' claim against

Pattern to match 'sub' against

Pattern to match 'aud' claim against.

Validate 'exp' is in the future

Validate 'nbf' is in the past

Clock skew for 'nbf' and 'exp' checks



# Creating the Trust Chain

Create the Trust Chain that will perform the token routing.

Add a new chain and specify the 'Overview' properties:

The screenshot shows a web interface for creating a new module chain. The title is "New Module Chain". Below the title are four tabs: "Overview", "Lookup", "Security", and "Properties". The "Overview" tab is selected. The form contains the following fields:

- \* Name: token-router-chain
- Description: Chain to route tokens and return an STSUU for Reverse Proxy Consumption
- \* Template: token-router-template (dropdown menu)
- Description: A template for token routing

# Creating the Trust Chain

Specify the 'Lookup' Properties

The reverse proxy issues the requests as:

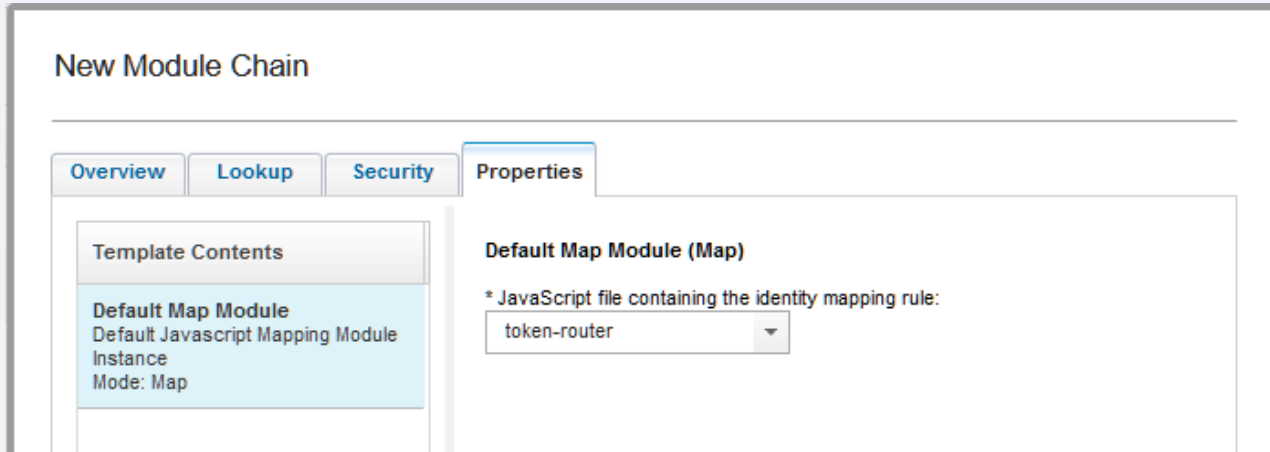
'urn:ibm:ITFIM:oauth20:token:bearer'

The screenshot shows the 'New Module Chain' configuration window with the 'Lookup' tab selected. The configuration includes the following fields and options:

- \* Request Type:** A dropdown menu set to 'Validate'.
- \* URI:** A text input field containing 'http://schemas.xmlsoap.org/ws/2005/02/trust/Validate'.
- Lookup Type:** Two radio button options: 'Traditional WS-Trust Elements' (selected) and 'XPath'.
- Applies to:** A section with the following fields:
  - \* Address:** A text input field containing 'urn:token.router'.
  - Service Name:** Two empty text input fields separated by a colon.
  - Port Type:** Two empty text input fields separated by a colon.
- Issuer:** A section with the following field:
  - \* Address:** A text input field containing 'urn:ibm:ITFIM:oauth20:token:bearer'.

# Creating the Trust Chain

Finally, we specify the mapping rule to be used in the module 'Properties' configuration:



The token-router mapping rule:

<https://github.com/IBM-Security/isam-support/blob/master/config-example/federation/ws-trust/mapping/token-router.js>

# Relevant Reverse Proxy settings

Configure your Reverse Proxy for both 'Browser' and 'API Protection' via the 'OAuth and OpenID Connect Provider Configuration' wizard.

Then, the following is the only Reverse Proxy configuration file change you'll need to make:

```
[oauth]  
...  
default-fed-id = urn:token:router
```



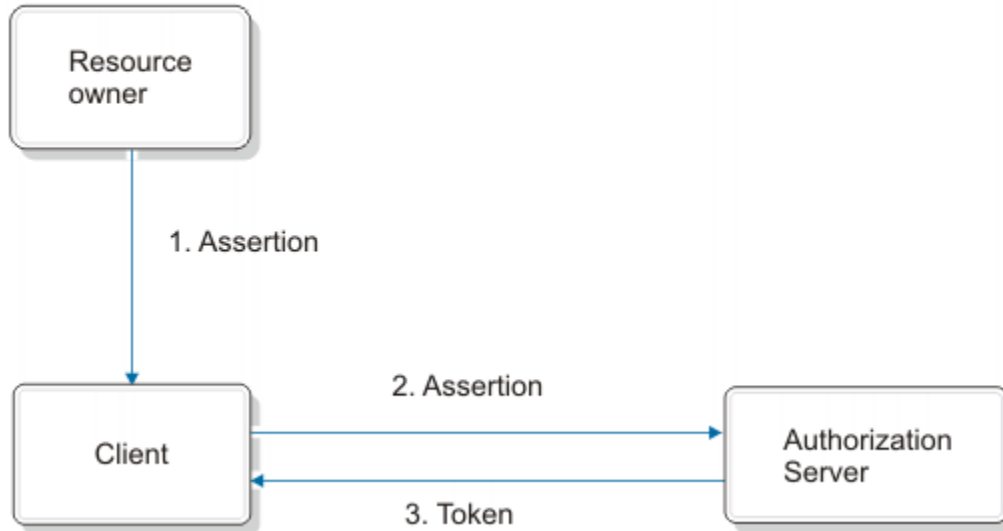
# OAuth 2.0 JWT Bearer Profile Overview

- Specification Reference and Workflow Review
- Creating an API Protection Definition
- Creating a client
- Updating the mapping rules
- Testing the configuration

# Specification Reference and Workflow Review

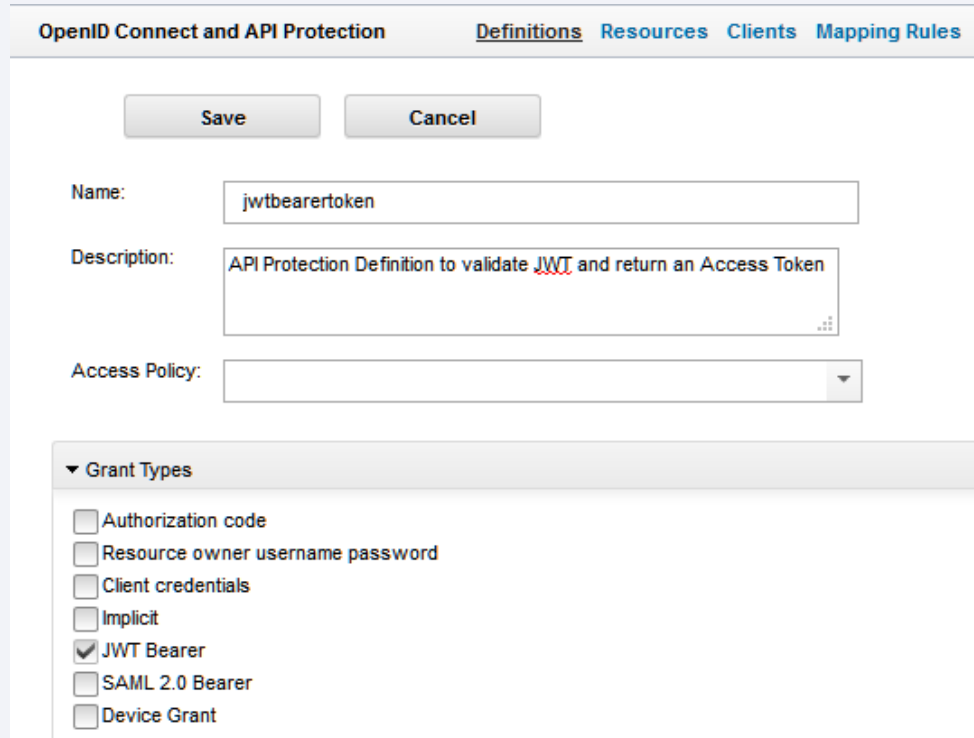
RFC7523 defines the JWT Bearer grant type. This grant type allows the client to submit a JWT Bearer assertion in exchange for an Access Token.

The following diagram describes the steps in the assertion bearer grant type flows:



# Creating an API Protection Definition

Create an API Protection Definition with the type 'JWT Bearer'.



The screenshot shows a web interface for creating an API Protection Definition. At the top, there are navigation tabs: "OpenID Connect and API Protection" (selected), "Definitions", "Resources", "Clients", and "Mapping Rules". Below the tabs are two buttons: "Save" and "Cancel".

The main form contains the following fields:

- Name:** A text input field containing "jwtbearertoken".
- Description:** A text area containing "API Protection Definition to validate JWT and return an Access Token".
- Access Policy:** A dropdown menu that is currently empty.

Below the form is a section titled "Grant Types" with a dropdown arrow. It contains a list of grant types with checkboxes:

- Authorization code
- Resource owner username password
- Client credentials
- Implicit
- JWT Bearer
- SAML 2.0 Bearer
- Device Grant



# Creating a client

Create a client for the JWT Bearer API Protection Definition:

### New Client

**Client Configuration** | Extension Properties

Client ID:

Client name:

API definition:  ▼

Confidential:

Client secret:

Redirect URI:

Company name:

# Updating the mapping rules

Edit the 'preTokenGeneration' and update the following line to enable assertion grant types:

```
var enableAssertionGrants = true;
```

An RFC compliant validation script:

[https://github.com/IBM-Security/isam-support/blob/master/config-example/aac/oauth\\_js/oauth/jwtbearer/assertionGrantValidationTools.js](https://github.com/IBM-Security/isam-support/blob/master/config-example/aac/oauth_js/oauth/jwtbearer/assertionGrantValidationTools.js)

A preTokenGeneration mapping rule that implements it:

[https://github.com/IBM-Security/isam-support/blob/master/config-example/aac/oauth\\_js/oauth/jwtbearer/oauth-jwtbearer-preTokenGeneration.js](https://github.com/IBM-Security/isam-support/blob/master/config-example/aac/oauth_js/oauth/jwtbearer/oauth-jwtbearer-preTokenGeneration.js)



# Questions for the panel

Ask the panelists a question now

**Enter your question in the Q&A widget**

To ask a question after this presentation:

**Ask follow-up questions in the IBM Support Forums**

<https://www.ibm.com/my-support/s/forumshome>

# For more information

- **Product Forum:** <http://ibm.biz/ISAM-support-forum>
- **Security Learning Academy:** <https://ibm.biz/ISAM-LearningAcademy>
- **IBM Knowledge Center:**  
<https://www.ibm.com/support/knowledgecenter/SSPREK/welcome.html>

Useful links:

[Get started with IBM Security Support](#)   [IBM Support](#)  
[Sign up for My Notifications](#)   [IBM Security Community](#)

**Follow us:**



[www.youtube.com/user/IBMSecuritySupport](http://www.youtube.com/user/IBMSecuritySupport)



[twitter.com/askibmsecurity](http://twitter.com/askibmsecurity)



<http://ibm.biz/ISCS-LinkedIn>

# Thank you

Follow us on:

[securitylearningacademy.com](https://securitylearningacademy.com)

[youtube/user/IBMSecuritySupport](https://youtube.com/user/IBMSecuritySupport)

[@AskIBMSecurity](https://twitter.com/AskIBMSecurity)

[ibm.biz/IBMSecurityClientSuccess-LinkedIn](https://ibm.biz/IBMSecurityClientSuccess-LinkedIn)

[securityintelligence.com](https://securityintelligence.com)

[xforce.ibmcloud.com](https://xforce.ibmcloud.com)

[ibm.com/security/community](https://ibm.com/security/community)

© Copyright IBM Corporation 2019. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. Any statement of direction represents IBM's current intent, is subject to change or withdrawal, and represent only goals and objectives. IBM, the IBM logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.

All names and references for organizations and other business institutions used in this deliverable's scenarios are fictional. Any match with real organizations or institutions is coincidental.

Statement of Good Security Practices: IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a lawful, comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. IBM does not warrant that any systems, products or services are immune from, or will make your enterprise immune from, the malicious or illegal conduct of any party.