SPIRION

Spirion Extensions

Single Sign-On for the SDM Console with Shibboleth 3 IdP

Table of Contents

Single Sign-On for the SDM Console with Shibboleth 3 IdP

v20221221

Introduction

Single Sign-On (SSO) centralizes authentication by designating an ***Identity Provider*** (IdP) tasked with processing login requests from authorized, SSO-enabled applications that are referred to as ***Service Providers*** (SPs). Users authenticate into integrated services (SPs) using a single account orchestrated by the IdP, reducing the growth of unnecessary credentials.

Spirion's ***Sensitive Data Manager*** (SDM) Console supports SSO and comes configured by default to function as an SP for federated authentication using SAML 2.0. It is initially set for Microsoft's Active Directory Federation Services (ADFS) as the IdP, though alternative Identity Providers can be configured as well.

This extension describes how ***Shibboleth***, for example, provides an open-source alternative to ADFS that is often deployed as an IdP by higher education institutions to facilitate federated SSO.

Requirements

Before working on the steps outlined in this document, please confirm the following:

- The latest SDM console is accessible.
- HTTPS routing has been enabled for the SDM console.
- A Shibboleth IdP instance is available for configuration.

Important Notes

Software Versions

SDM Console version 11.8.2 was used to perform the steps described in this document.

Shibboleth version 3.4.7 and OpenLDAP version 2.5.8 were used for the SSO integration.

Spirion MSI Files

If an MSI file has not been created yet to deploy the Spirion Agent, please reference the following KB documentation: How to Create a Custom MSI for Windows Client. This process requires:

- Installation media from the Spirion software portal:
  - Windows Agent installation media
  - License Key File
- Registration file obtained from the SDM console.
- The latest Spirion MSI Builder.

Optional Test Environment

This article includes documentation for standing up Shibboleth 3 IdP in a Docker container along with an OpenLDAP container for optional use.

Sections marked ***(DOCKER)*** can be skipped if a test/dev environment is already staged with an easily reconfigurable instance of Shibboleth.

Process

The following attributes are mapped to user accounts in the SDM Console for SSO:

- User Name* – ***MANDATORY*** – typically mapped to "uid" or "sAMAccountName"
- E-mail – *OPTIONAL* – used for SMTP notifications and report exports
- Display Name – *OPTIONAL* – used for notification templates
- Roles/Groups – ***MANDATORY*** – typically mapped to "memberOf"
  - If membership information is not available, accounts belonging to SSO users must have their role added manually after initial SSO logon attempt.

*The included example configuration uses Shibboleth IdP connected to an OpenLDAP directory. Replace all instances of "uid" with "sAMAccountName" in the XML examples below if Shibboleth is instead synced to a Microsoft LDAP environment.

(DOCKER) Build the Docker Images

Each Docker container includes "build.sh" and "run.sh" scripts for easy deployment on Linux systems. If Bash scripting is not available, the scripts' contents should be referenced for the relevant Docker CLI input.

NOTE: Steps 3 through 8 are *optional* if your environment already has an LDAP server available for testing; be sure to update the Shibboleth IdP LDAP configuration accordingly.

1. Clone or download the following GitHub repository: https://github.com/winstonhong/Shibboleth-SAML-IdP-and-SP
2. If only using the Shibboleth IdP container, proceed to step 9. Otherwise, follow steps 3-8 to configure and build the OpenLDAP container as well.
3. From the local root directory of the "Shibboleth-SAML-IdP-and-SP" repository, navigate to the ***LDAP-Dockerized-CentOS*** directory.
4. Edit the accounts specified in ***users.ldif*** as desired for authentication testing.
5. At the end of ***users.ldif***, add a new user/security group object that will be used for role mapping when logging into the SDM Console via SSO, for example:

```
dn: cn=sdm,ou=groups,dc=example,dc=com
objectClass: groupOfNames
cn: sdm
description: SDM Console admins
```

6. Create a new file called ***memberof.ldif*** with the content listed below to enable the OpenLDAP membership overlay:

```
dn: cn=MemberOf Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
```

7. Edit the ***Dockerfile*** to include the following commands inserted after line 24:

```
&& ldapmodify -H ldap:/// -f /memberof.ldif -x -D "cn=Directory Manager" -w
ldap-passwd \
&& sleep 10 \
```

8. Run "`./build.sh`" from the terminal – this can take a few minutes depending on connection speed and system performance.
    a. NOTE: Disregard any errors referencing "cgroup support".
9. Navigate to the ***shibboleth-idp-dockerized*** directory from the local root of the repository and similarly run "`./build.sh`" from the terminal.
10. Run the command "`docker images`" to confirm that the expected images are available – they are labeled ***example/shibboleth-idp*** and ***example/openldap*** by default.

```
REPOSITORY                          TAG       IMAGE ID       CREATED          SIZE
example/shibboleth-idp              latest    b69054041ed1   47 hours ago     931MB
example/openldap                    latest    bac5f5974a40   47 hours ago     464MB
```

Gather Metadata Files

These files contain the information/context needed to establish trust between an Identity Provider and a Service Provider.

1. From the root directory of the Shibboleth IdP installation, browse to the ***metadata directory***.
    a. ***Linux:*** */opt/shibboleth-idp/metadata*
    b. ***Windows:*** *C:\Program Files (x86)\Shibboleth\IdP\metadata*
    c. ***Shibboleth Docker container:*** *../shibboleth-idp-dockerized/ext-conf/metadata* (from the local root directory of the "Shibboleth-SAML-IdP-and-SP" repository).
2. Copy ***idp-metadata.xml*** to the server where the SDM Console is installed, this will be used in a later step.
3. From the SDM Console server, open the Spirion Console Administrator Tool (CAT).
4. Navigate to the ***Authentication and AD Settings*** menu and select ***Configure Single Sign-On***.
5. Click ***Generate SAML Metadata*** and save the XML file as "***sdm-metadata.xml"***.
6. Copy the Spirion XML file to the server running Shibboleth, placing it in the same directory as ***idp-metadata.xml*** (from step 2).

Shibboleth IdP Configuration for Spirion SSO

1. From the Shibboleth IdP's root directory, browse to the ***conf*** folder:
    a. ***Linux:*** */opt/shibboleth-idp/conf*
    b. ***Windows:*** *C:\Program Files (x86)\Shibboleth\IdP\metadata*
    c. ***Shibboleth Docker container:*** *../shibboleth-idp-dockerized/ext-conf/conf* (from the local root directory of the "Shibboleth-SAML-IdP-and-SP" repository).
2. Edit ***metadata-providers.xml*** to add Spirion as a SP:

```
<MetadataProvider xsi:type="FilesystemMetadataProvider" id="SDM"
metadataFile="%{idp.home}/metadata/sdm-metadata.xml"/>
```

3. Edit ***relying-party.xml*** to add Spirion and ***set (only) assertions to be signed*** by including the following flags:

```
<bean parent="RelyingPartyByName" c:relyingPartyIds="https://spirion.com/sp">
  <property name="profileConfigurations">
    <list>
      <bean parent="SAML2.SSO" p:signAssertions="true" p:signResponses="false" />
    </list>
  </property>
</bean>
```

NOTE: Spirion must always be referenced by the entity ID ***https://spirion.com/sp*** when interfacing with Identity Providers.

4. Edit ***idp.properties*** to change the following line from being commented out, and set its value to "true":
   a. BEFORE:
      `#idp.encryption.optional = false`
   b. AFTER:
      `idp.encryption.optional = true`

5. Edit ***attribute-resolver.xml*** – or *attribute-resolver-full.xml* for the Docker container – and ensure the following definitions are included (see attached example for further reference):

EITHER: uid – mapped as user login in non-Microsoft environments:

```
<resolver:AttributeDefinition id="uid" xsi:type="ad:PrincipalName">
    <resolver:AttributeEncoder xsi:type="enc:SAML1String"
name="urn:mace:dir:attribute-def:uid" encodeType="false" />
    <resolver:AttributeEncoder xsi:type="enc:SAML2String"
name="urn:oid:0.9.2342.19200300.100.1.1" friendlyName="uid" encodeType="false" />
</resolver:AttributeDefinition>
```

OR: sAMAccountName – mapped as user login in Microsoft environments:

```
<resolver:AttributeDefinition xsi:type="ad:Simple" id="sAMAccountName"
sourceAttributeID="sAMAccountName">
       <resolver:Dependency ref="myLDAP" />
       <resolver:AttributeEncoder xsi:type="enc:SAML1String"
name="urn:mace:dir:attribute-def:samaccountname" />
       <resolver:AttributeEncoder xsi:type="enc:SAML2String"
name="urn:oid:1.2.840.113556.1.4.221" friendlyName="samaccountname" />
</resolver:AttributeDefinition>
```

memberOf – mapped to set scope of SDM Console access:

```
<resolver:AttributeDefinition xsi:type="ad:Simple" id="memberOf"
sourceAttributeID="memberOf">
       <resolver:Dependency ref="myLDAP" />
       <resolver:AttributeEncoder xsi:type="enc:SAML1String"
name="urn:mace:dir:attribute-def:memberOf" />
       <resolver:AttributeEncoder xsi:type="enc:SAML2String"
name="urn:oid:1.3.6.1.4.1.5923.1.5.1.1" friendlyName="memberOf" />
</resolver:AttributeDefinition>
```

mail – mapped to receive SMTP notifications:

```
<resolver:AttributeDefinition xsi:type="ad:Simple" id="email"
sourceAttributeID="mail">
       <resolver:Dependency ref="myLDAP" />
       <resolver:AttributeEncoder xsi:type="enc:SAML1String"
name="urn:mace:dir:attribute-def:mail" />
       <resolver:AttributeEncoder xsi:type="enc:SAML2String"
name="urn:oid:0.9.2342.19200300.100.1.3" friendlyName="emailaddress" />
</resolver:AttributeDefinition>
```

cn – mapped to Display Name for notification templates:

```
<resolver:AttributeDefinition id="cn" xsi:type="ad:Simple" sourceAttributeID="cn">
    <resolver:Dependency ref="myLDAP" />
    <resolver:AttributeEncoder xsi:type="enc:SAML1String"
name="urn:mace:dir:attribute-def:cn" />
    <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.5.4.3"
friendlyName="cn" />
</resolver:AttributeDefinition>
```

6. Edit ***attribute-filter.xml*** to ensure the attributes added above are released to Spirion (see attached example for further reference):

```
<AttributeFilterPolicy id="Spirion SSO">
    <PolicyRequirementRule xsi:type="Requester" value="https://spirion.com/sp" />
    <AttributeRule attributeID="uid" permitAny="true" />
    <AttributeRule attributeID="displayName" permitAny="true" />
    <AttributeRule attributeID="email" permitAny="true" />
    <AttributeRule attributeID="memberOf" permitAny="true" />
```

(DOCKER) Run the Docker Images

Each Docker container includes “build.sh” and “run.sh” scripts for easy deployment on Linux systems. If Bash scripting is not available, the scripts’ contents should be referenced for the relevant Docker CLI input.

NOTE: Step 2 is *optional* if your environment already has an LDAP server available for testing; be sure to update the Shibboleth IdP LDAP configuration accordingly.

1. Navigate to the local root directory of the "Shibboleth-SAML-IdP-and-SP" repository using a terminal.
2. To start the OpenLDAP container, run the following command:
   `./LDAP-Dockerized-CentOS/run.sh`
3. To start the Shibboleth-IDP container, run the following command:
   `./LDAP-Dockerized-CentOS/run.sh`
4. Run the command "`docker ps a`" to confirm that the expected images are now running – they are named ***example/shibboleth-idp*** and ***example/openldap*** by default.

(DOCKER) Group Membership Adjustment in OpenLDAP

These steps are required to properly map user account group membership info using the *memberOf* attribute. Without proper role mapping, logins using SSO will need to have their SDM Role membership adjusted manually after the initial authentication attempt.

1. ***Replace the UID*** with a user from your ***users.ldif*** file and enter the following command from a terminal on the Docker host:

```
docker exec openldap bash -c "echo 'dn: cn=sdm,ou=groups,dc=example,dc=com
changetype: modify
add: member
member: uid=*INSERT UID*,ou=people,dc=example,dc=com' > addUsersToGroup.ldif"
```

2. This creates a .ldif file that we can use to modify group membership with the following command:

```
docker exec openldap ldapmodify -w ldap-passwd -D "cn=Directory Manager" -f
addUsersToGroup.ldif
```

3. The terminal should return confirmation that it is ***modifying entry…*** after confirming "Type or value exists (20)"
4. Enter the following command to bring up all users and groups in the OpenLDAP test environment:

```
docker exec openldap ldapsearch -w ldap-passwd -D "cn=Directory Manager" -b
dc=example,dc=com
```

5. Here, we should be able to confirm that the "memberOf" attribute is included under the user account added to the group using the steps referenced in this section:

```
memberOf: cn=sdm,ou=Groups,dc=example,dc=com
```

6. If "memberOf" is not referenced for the intended user account, role mapping will not be available. Confirm that the OpenLDAP group membership overlay is enabled and, if it is not, rebuild the container to redeploy after making the necessary changes.

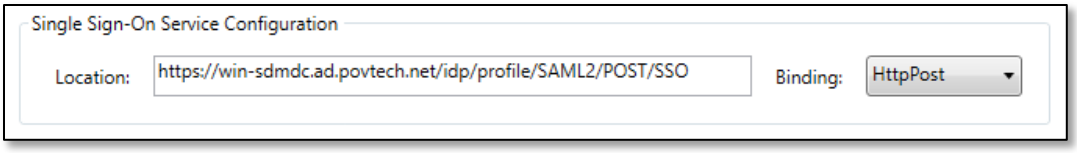
Import IdP Metadata into the SDM Console

1. From the server where the SDM Console is installed, edit the ***idp-metadata.xml*** file copied over in the "Gather Metadata Files" section of this document.
2. Comment out (or remove) the following lines (see attached example for further reference):

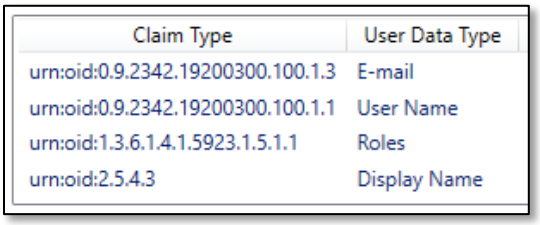
```
<SingleSignOnService Binding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"
Location="https://win-sdmdc.ad.povtech.net/idp/profile/Shibboleth/SSO"/>

<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-
SimpleSign" req-attr:supportsRequestedAttributes="true" Location="https://win-
sdmdc.ad.povtech.net/idp/profile/SAML2/POST-SimpleSign/SSO"/>
```

SDM Console Configuration for SSO

1. From the server where the SDM Console is installed, open the ***Spirion Console Administrator Tool***.
2. Navigate to the ***Authentication and AD Settings*** menu and select ***Configure Single Sign-On***.
3. Ensure that ***Enable Single Sign-On*** is selected.
4. ***Auto Create User*** and ***Auto Update User Data*** should also be selected.
5. ***Auto Sign-On*** is optional, depending on the intended user experience.
6. Under the ***Identity Provider*** section, click ***Load from SAML Metadata*** and select the ***idp-metadata.xml*** file edited in the previous section.
7. Once loaded, a ***Location*** should populate under ***Single Sign-On Service Configuration***: https://[*IdP Server URL or IP Address*]/idp/profile/SAML2/POST/SSO

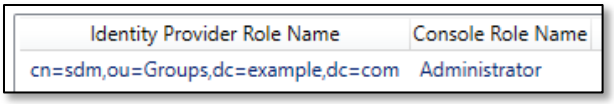


8. Toggle ***Force Authentication*** depending on the intended user experience. If enabled, users will always have to authenticate (via SSO) when logging into the Spirion console even if their credentials are already cached from authenticating to the IdP previously.
9. Expand ***Mappings*** and edit the entries so that the ***Claim Type*** for each ***User Data Type*** matches the attributes configured in the "Shibboleth IdP Configuration: Attributes" section:
    a. User Name
        i. Use ***one*** of the following, depending on LDAP environment:
        ii. Non-Microsoft Active Directory (uid):
            - urn:oid:0.9.2342.19200300.100.1.1
        iii. Microsoft Active Directory (sAMAccountName):
            - urn:oid:1.2.840.113556.1.4.221
    b. E-mail
        i. urn:oid:0.9.2342.19200300.100.1.3
    c. Roles
        i. urn:oid:1.3.6.1.4.1.5923.1.5.1.1
    d. Display Name
        i. urn:oid:2.5.4.3



Claim Type	User Data Type
urn:oid:0.9.2342.19200300.100.1.3	E-mail
urn:oid:0.9.2342.19200300.100.1.1	User Name
urn:oid:1.3.6.1.4.1.5923.1.5.1.1	Roles
urn:oid:2.5.4.3	Display Name

10. Expand ***Role Mappings*** and specify the user group that will be associated with the SDM Console's Administrative role. For example, to map the group specified in the OpenLDAP docker container, "sdm", enter the following: *cn=sdm,ou=Groups,dc=example,dc=com*
    a. This sets account permissions based on the associated Spirion role access while also preventing unauthorized users from logging in at all.



Identity Provider Role Name	Console Role Name
cn=sdm,ou=Groups,dc=example,dc=com	Administrator

11. Click ***Accept*** to finalize the SSO changes.
12. ***Save*** and ***Close*** the Console Administrator Tool.
13. Test logging in via SSO from the Spirion Console login page.

Outcomes

With Single Sign On enabled, the SDM Console login screen will include a separate button for SSO authentication.

Clicking ***Continue*** redirects users to the login screen associated with the configured IdP, where they will authenticate using the federated credentials. SSO authentication generates corresponding user accounts in SDM that are actively synced with the IdP so long as the "Auto Create" and "Auto Update" user flags are enabled via the Console Administrator Tool as outlined above.

Benefits

1. Securely and seamlessly ***integrate domain account authentication*** into Spirion for federated access, which avoids the need for separate SDM Console credentials.
2. Generate SDM Console user and admin accounts automatically with ***role-based access control derived from domain account attributes***.
3. Maintain ***standardized logon processes***, including user account mechanisms like MFA or password resets.