Integrating a third-party identity provider in front of Microsoft ADFS with Parallels Secure Workspace

- Parallels Secure Workspace

Guide

Note: This article provides a demo configuration. The support team will not assist in setting up the link between Microsoft ADFS and the third-party identity provider (IdP). Support is limited to troubleshooting issues between Parallels Secure Workspace and Microsoft ADFS.

Context

Some customers would like to authenticate using their national ID issued by their government.

This scenario could be applied to any situation where there is an Identity Provider providing some form of unique identifier. In practice, this IdP is often a web platform where users can sign in using various methods; for example using some form of national ID. This unique identifier is stored as an attribute value of the Active Directory user object.

Since this unique identifier is not the user principal name (UPN) which is required by Parallels Secure Workspace, Microsoft ADFS will add this information during the authentication flow.

Terminology

To map some terminology between SAML and Active Directory Federated Services:

ADFS	SAML	Basic description
Claims Provider (CP)	Identity Provider (IdP)	Responsible for authenticating the user and providing additional information about the identity of the user.
Relying Party (RP)	Service Provider (SP)	Responsible for providing a service.

In this tutorial, there are 2 relationships:

1. Between **KeyCloak** and **Microsoft ADFS**. In this case, KeyCloak will be the claims provider. Users will authenticate against KeyCloak. KeyCloak provides the identity; while Microsoft ADFS consumes this identity and is thus the relying party or service provider.
2. Between **Microsoft ADFS** and **Parallels Secure Workspace.** In this case, Microsoft ADFS acts as the IdP to Parallels Secure Workspace (SP).

How it works

For the end user, this is the flow:

- The end user navigates to the Workspace URL.
- The Workspace redirects to Microsoft ADFS.
- Microsoft ADFS redirects to the external Identity Provider (in this tutorial, KeyCloak is used as a stand-in for an identity provider where users would authenticate using their national ID or in another way; without a direct link to their UPN).
- User authenticates to KeyCloak.
- KeyCloak sends SAML response; so KeyCloak redirects to Microsoft ADFS.
- Microsoft ADFS provides SAML response; so Microsoft ADFS redirects to Parallels Secure Workspace's Assertion Consumer Service (ACS) URL.

In the background, this is how everything is linked:

External IdP Link: unique identifier **Microsoft ADFS** Link: user principal name **Parallels Secure Workspace**.

Microsoft ADFS will find the proper Active Directory user by issuing a query to find the (only) user who has this unique identifier as the value of one of the Active Directory user attributes. Microsoft ADFS will issue some claims, providing additional info about the user if needed. In this example, this means the user principal name (UPN) and a display name will be passed to Parallels Secure Workspace.

Working setup

This example assumes a “national ID” will be used as the link to identify an authenticated user (through KeyCloak) in Microsoft ADFS.

In reality, for security reasons, this should not be a plain text version but a non-decryptable hash or another unique alternative identifier.

KeyCloak

Note: the initial setup of KeyCloak is not in the scope of this article.

Configuring the client

In the dedicated realm (already created - in this example: **realm-test-nationalid)**, go to **Clients > Import Client**.

The configuration can mostly be imported from the federation metadata XML.

Specify the URL: http://adfs.somedomain.org/FederationMetadata/2007-06/FederationMetadata.xml

It’s possible to tweak the example configuration for debugging (e.g. disable encryption). For the final configuration, be sure to make it as secure as possible.

The imported configuration will look like this:

For **client scopes**, there will be a **dedicated** one that needs to be adjusted. Just click on the name of this dedicated scope.

For each claim (in this example: "nationalID")

Click **Add Mapper > By Configuration > User Attribute** (Map a custom user attribute to a SAML attribute).

Do the same for the given name claim, assuming the given name from the KeyCloak Identity Provider should be used as a display name in Parallels Secure Workspace.
(Use the URI: http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname)

Configuring the user

Go to **Users > select the user > Attributes** and add the custom attributes.

Microsoft ADFS

Configuring KeyCloak as a Claims Provider

In the ADFS console, go to Claims Provider Trusts.
At the right, choose the **Add Claims Provider Trust …** action.

Import the metadata XML which can be obtained in KeyCloak from the Realm Settings:

After import, it should look similar to this:

Hint: go to the Certificates tab to check out the details and import the certificate in a trusted certificate store.

When surfing to the ADFS (https://adfs.somedomain.org/adfs/ls/idpinitiatedsignon.aspx), the **KeyCloak** option should be listed as well now (assuming a default configuration where only Active Directory was present before).

Configuring acceptance claim rules

The acceptance claim rules will be used to find Active Directory users and find their UPN.

Some info could also be passed on (e.g. if the given name provided by the claims provider can stay untouched). Otherwise, adapt and use the same approach as for the UPN to find the given name.

In ADFS: **Claims Provider Trusts >** select KeyCloak: in the actions menu (at the right), click **Edit Claim Rules…**

Configure these two rules:

Configure a rule to find the UPN.

In this **custom** rule, the claim sent by KeyCloak (“nationalID”) will be used to query the Active Directory to find a user principal name.

```
c:[Type == "nationalID"]
 => issue(store = "Active Directory", types = ("http://schemas.xmlsoap.org/ws/2005/05/identity/claims/u
```

Basically, there is an LDAP query to find the user whose "description" (can be any other LDAP attribute) matches the **unique** value of the “nationalID” claim.

Then, Microsoft ADFS issues a new claim: http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn

The structure of the query is: <LDAP_FILTER>;<DESIRED_LDAP_ATT>;<LDAP_USER> .

LDAP filter is an LDAP query. Variables ({0}) are specified near the end.

In this case, only select one desired LDAP attribute: **userprincipalname** .

The LDAP user can be any valid user name. However, it must be specified as **domain\user**.

- **What happens if there is no match (user) for the unique identifier?**
  After authentication, the user is redirected to Parallels Secure Workspace. This error will be shown:

  ```
  {"error": "The status code of the Response was not Success, was
  Requester -> urn:oasis:names:tc:SAML:2.0:status:InvalidNameIDPolicy"}
  ```
- **What happens if there is more than one match for the unique identifier?**
  After authentication, the user is redirected to Parallels Secure Workspace. This error will be shown:
  ```
  {"error": "The status code of the Response was not Success, was
  Responder"}
  ```

Configure a rule to issue the given name claim.

Note: in this example, the given name already comes from KeyCloak. To just pass it through to Parallels Secure Workspace, the following snippet can be used. Alternatively, adjust the rule above to return a different user attribute instead of the "userprincipalname" to provide a display name stored in one of the attributes of the Active Directory user.

```
c:[Type == "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname"]
 => issue(claim = c);
```

Configuring Parallels Secure Workspace as Relying Party Trust

Under **ADFS > Relying Party Trusts**, select the existing Parallels Secure Workspace configuration (or create one as described in the Admin Manual).

Next, adjust the **Issuance Claim Rules** to match this:

Create a “Transform an Incoming Claim” rule:

- **Claim rule name:** UPN (can be anything)
- **Incoming claim type:** UPN
- **Outgoing claim type**: Name ID
- **Outgoing name ID format:** Email
- **Pass through all claim values**

Create two "Passthrough" rules: one for UPN and one for Given Name. Without these rules, there will be no attribute statements in the XML response. In the Parallels Secure Workspace log files, this would lead to a similar error:

```
2023-03-21 13:31:47.418503+00:00 node01 awingu-api.service[/opt/awingu/awingu-core/virtualenv/bin/gunic
NoneType: None
```

Example of a proper SAML response

A proper SAML response should look like this (from Microsoft ADFS to Parallels Secure Workspace):

```
<samlp:Response ID="_50ff6978-ae62-477c-a11e-bdbef3df0dd1" Version="2.0" IssueInstant="2023-03-21T10:32
    <Issuer xmlns="urn:oasis:names:tc:SAML:2.0:assertion">http://JB-ADFS-01.somedomain.com/adfs/service
    <samlp:Status>
        <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </samlp:Status>
    <Assertion ID="_703f3d95-17e0-40a7-af09-309f8c2bb05f" IssueInstant="2023-03-21T10:32:37.897Z" Versi
        <Issuer>http://JB-ADFS-01.somedomain.com/adfs/services/trust</Issuer>
        <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
            <ds:SignedInfo>
                <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
                <ds:Reference URI="#_703f3d95-17e0-40a7-af09-309f8c2bb05f">
                    <ds:Transforms>
                        <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"
                        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                    </ds:Transforms>
                    <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                    <ds:DigestValue>EuoXta/JlrkeYxcW1lI7edG2f7FgDZqm5lRTaCm44n0=</ds:DigestValue>
                </ds:Reference>
            </ds:SignedInfo>
            <ds:SignatureValue>QSDfWQ0SHJW6xq+PtONBPMkpDXnw5/f8gU+1F0O13gJWtl0fUZrlA2T4UfxOX1UV1bMMBPn0
            <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
                <ds:X509Data>
                    <ds:X509Certificate>MIIE9DCCAtygAwIBAgIQGiytnm7ROo9N1HFBhiOedjANBgkqhkiG9w0BAQsFADA
                </ds:X509Data>
            </KeyInfo>
        </ds:Signature>
        <Subject>
            <NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">j@somedomain.com</N
            <SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
                <SubjectConfirmationData NotOnOrAfter="2023-03-21T10:37:37.897Z" Recipient="https://wor
            </SubjectConfirmation>
        </Subject>
        <Conditions NotBefore="2023-03-21T10:32:37.897Z" NotOnOrAfter="2023-03-21T11:32:37.897Z">
            <AudienceRestriction>
                <Audience>Workspace</Audience>
            </AudienceRestriction>
        </Conditions>
        <AttributeStatement>
            <Attribute Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn">
                <AttributeValue>j@somedomain.com</AttributeValue>
            </Attribute>
            <Attribute Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname">
                <AttributeValue>Some User/AttributeValue>
            </Attribute>
        </AttributeStatement>
        <AuthnStatement AuthnInstant="2023-03-21T10:32:37.772Z" SessionIndex="_703f3d95-17e0-40a7-af09-
            <AuthnContext>
                <AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
            </AuthnContext>
        </AuthnStatement>
    </Assertion>
</samlp:Response>
```

Pitfalls

The below error occurred when Parallels Secure Workspace initiated the log-on procedure.
It did not occur when initiating the log-on from ADFS.

```
{"error": "The status code of the Response was not Success, was Requester -> urn:oasis:names:tc:saml:2.
```

In the Windows Event Viewer, the details can be seen in **Applications and Services Logs > AD FS > Admin**.

```
The SAML authentication request had a NameID Policy that could not be satisfied.
Requestor: Workspace
Name identifier format: urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
SPNameQualifier:
Exception details:
MSIS7070: The SAML request contained a NameIDPolicy that was not satisfied by the issued token. Request

This request failed.

User Action
Use the AD FS Management snap-in to configure the configuration that emits the required name identifier
```

To fix this: in **ADFS > Relying Party Trusts**: Select Parallels Secure Workspace. Edit the rule that transforms the UPN (incoming claim type) to Name ID (outgoing claim type). Make sure to specify **Email** as outgoing name ID format.

Hints

If only the KeyCloak claims provider should be used for authentication, the flow for the end user can be simplified and the step where the user has to select either "KeyCloak" or "Active Directory" can be skipped.

On the Microsoft ADFS server, run this PowerShell snippet (adjust the name of the relying party trust and claims provider):

```
Set-ADFSRelyingPartyTrust -TargetName "Workspace" -ClaimsProviderName
@("KeyCloak")
```

Article background information

This documentation was initially created on the 22nd of March 2023.

The setup was performed using:

- KeyCloak 21 running on Ubuntu 22.04.
- The Microsoft ADFS and Active Directory Domain Controllers were running on Microsoft Windows Server 2022.
- Parallels Secure Workspace (Awingu) 5.4 was used.