X.509 PKI AUTHENTICATION SETUP WITH SAML

Security Assertion Markup Language (SAML) (https://www.onelogin.com/learn/saml) is an open standard used for authentication. Based on the Extensible Markup Language (XML) format, web applications use SAML to transfer authentication data between two parties: the identity provider (IdP) and the service provider (SP).

SP is the application that the user is attempting to access. The IdP provides endpoints that an application can use to verify identity. The IdP must be associated with a federated user store so that it can compare authentication information with known users. The federated user store can be Active Directory, LDAP, or a custom user store defined by the IdP.

The steps in this guide cover how to configure your Director for SAML authentication with Active Directory Federation Services (AD FS) using a certificate with the **X.509 PKI** (https://en.wikipedia.org/wiki/X.509) standard.

Federated Server Requirements on AD FS

- **Authentication Context**
  - The federated server must have the necessary authentication context. For example, if authentication is done with SmartCardPKI, the AD FS server must have this setting enabled. To verify that this setting is enabled, run the following command in PowerShell:

    ```
    > Get-ADFSProperties
    ```

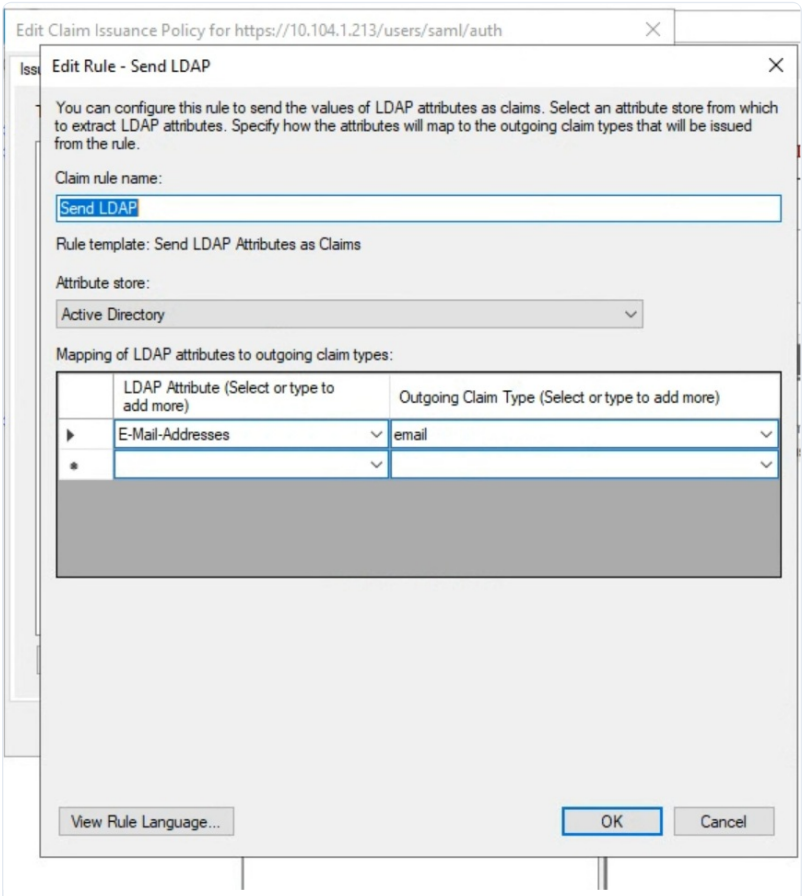
  - Highlighted in this example is the X509 AuthenticationContext for testing purposes (software certificate).

```
PS C:\Windows\system32> Get-AdfsProperties


AcceptableIdentifiers                   : {}
AddProxyAuthorizationRules              : exists([Type ==
                                          "http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid", Value
                                          == "S-1-5-32-544", Issuer =~ "^AD AUTHORITY$"]) => issue(Type =
                                          "http://schemas.microsoft.com/authorization/claims/permit", Value =
                                          "true");
                                                        c:[Type ==
                                          "http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid",
                                          Issuer =~ "^AD AUTHORITY$" ]
                                                                  => issue(store="_ProxyCredentialStore",types=("htt
                                          osoft.com/authorization/claims/permit"),query="isProxyTrustManagerSid({0})
                                          ", param=c.Value );
                                                        c:[Type ==
                                          "http://schemas.microsoft.com/ws/2008/06/identity/claims/proxytrustid",
                                          Issuer =~ "^SELF AUTHORITY$" ]
                                                                  => issue(store="_ProxyCredentialStore",types=("htt
                                          osoft.com/authorization/claims/permit"),query="isProxyTrustProvisioned({0}
                                          )", param=c.Value );
ArtifactDbConnection                    : Data Source=np:\\.\pipe\microsoft##wid\tsql\query;Initial
                                          Catalog=AdfsArtifactStore;Integrated Security=True
AuthenticationContextOrder              : {urn:oasis:names:tc:SAML:2.0:ac:classes:Password,
                                          urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport,
                                          urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient,
                                          urn:oasis:names:tc:SAML:2.0:ac:classes:X509...}
AuditLevel                              : {Basic}
AutoCertificateRollover                 : False
CertificateCriticalThreshold            : 2
CertificateDuration                     : 365
CertificateGenerationThreshold          : 20
CertificatePromotionThreshold           : 5
CertificateRolloverInterval             : 720
CertificateSharingContainer             : CN=e2e72d2e-b688-4d36-9af9-2af6d99a111a,CN=ADFS,CN=Microsoft,CN=Program
                                          Data,DC=mandiant,DC=local
CertificateThresholdMultiplier          : 1440
ClientCertRevocationCheck               : None
ContactPerson                           : Microsoft.IdentityServer.Management.Resources.ContactPerson
DisplayName                             : MandiantAD01
IntranetUseLocalClaimsProvider          : False
ExtendedProtectionTokenCheck            : Allow
FarmRoles                               : Microsoft.IdentityServer.PolicyModel.Configuration.FarmRolesConfiguration
FederationPassiveAddress                : /adfs/ls/
HostName                                : MandiantAD01.mandiant.local
HttpPort                                : 80
HttpsPort                               : 443
TlsClientPort                           : 49443
Identifier                              : http://mandiantad01.mandiant.local/adfs/services/trust
IdTokenIssuer                           : https://mandiantad01.mandiant.local/adfs
InstalledLanguage                       : en-US
LogLevel                                : {Errors, FailureAudits, Information, Verbose...}
```

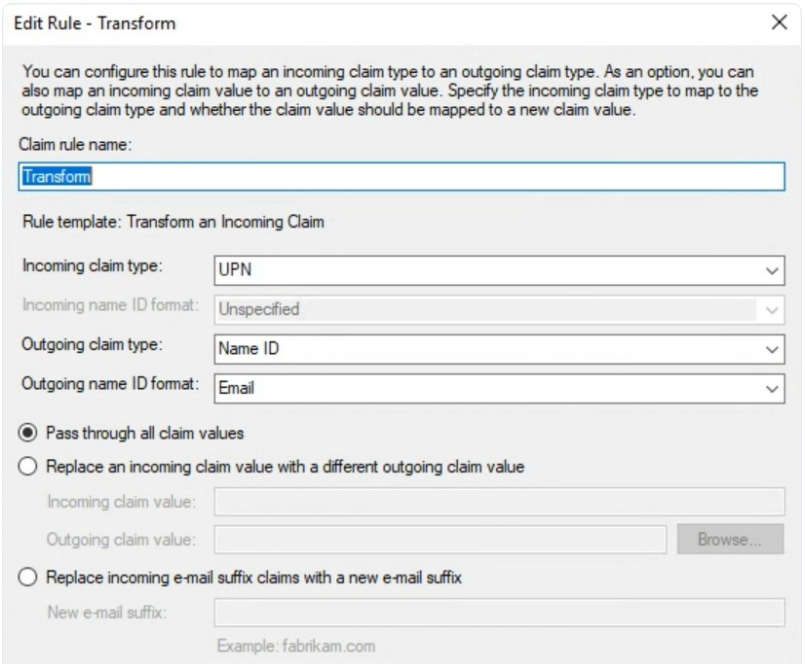
Example Software Certificate with X509 AuthenticationContextOrder Highlighted

- **Claim Issuance Policies**: These policies may be necessary to properly match rules to the necessary format (email address) that the Director requires.
  - **Rule - Send LDAP**: This rule is necessary to send an LDAP attribute as a claim. The Director requires a SAML claim type of email, but that doesn't exactly map to anything in Active Directory. This rule ensures that LDAP knows what we're asking for when we send the claim for email.



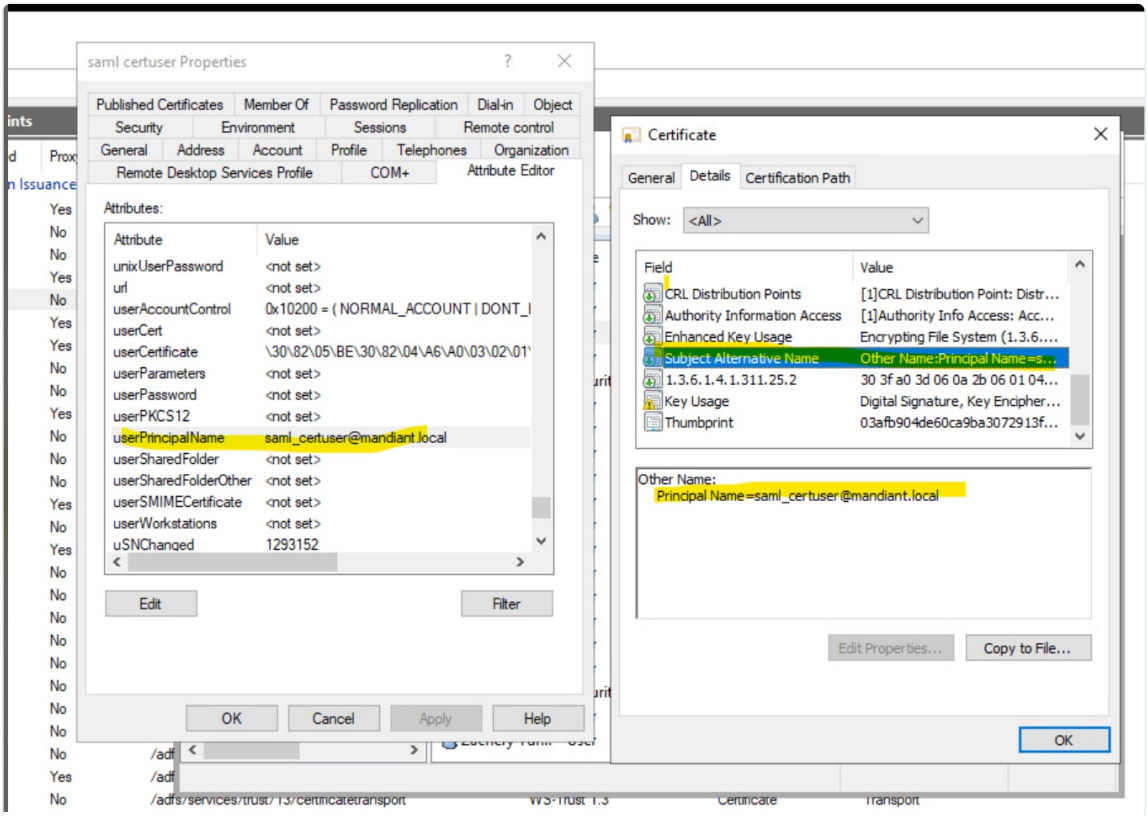
Send LDAP Rule, mapping E-Mail-Addresses attribute to email outgoing claim type.

  - **Transform**: You may also need to ensure that the incoming claim matches an outgoing claim type. Similar to the Send LDAP rule, the languages must match. This match is done by transforming the claim into the proper attribute when sending it over.



Example Transform Claim Rule, used to map UPN to Name ID in Email format

- To understand this concept, look at Active Directory and the smart card certificate.



Smart card certificate in AD FS

Certificate authentication relies on common fields being referenced to authenticate a user. Most often, "UPN - userPrincipalName" appears. In Active Directory, this field is the email address. Similarly, on the certificate,

"SAN > Other Name" is used. Get the attribute from SAN, else use an attribute that is in the email address format. This attribute might be a numeric string, such as `123456700001@example.org`.

Set up Director for SAML with AD FS

1. Go to **Settings** > **User Settings**. The User Settings page opens.
2. Select **Authentication** and then select **SAML**.
3. Optional: If using the Director fully qualified domain name (FQDN) instead of the IP address, enter the FQDN value for **Director Hostname for SAML URLs**.

   FQDNs must comply with RFC 1123, a standard that defines the requirements for FQDNs on the internet. This standard specifies that FQDNs can only contain the following:
   - Letters (A-Z, a-z)
   - Digits (0-9)
   - Hyphens (-)

   Underscores are not permitted.

   For more information, see **RFC 1123: Requirements for Internet Hosts (https://www.rfc-editor.org/rfc/rfc1123.html)**.

4. Define the authentication fields:
   - **SAML - Name Identifier Format**: Leave the default value of `urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress`, because using the `emailAddress` format is a requirement for SAML in Mandiant Security Validation (MSV).
   - **SAML - AuthN Context**: Change this field to SmartCardPKI.

     ```
     urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI
     ```

   - **SAML - IdP SSO URL (Login)**: Use the URL of the AD FS server. For example, `https://example.local/adfs/ls`
   - **SAML - IdP SSO URL (Logout)**: Use the URL of the AD FS server. For example, `https://example.local/adfs/ls`
   - **SAML - Attribute Map**: Leave as the default. Modifying the mapping in MSV has not been tested.
   - **SAML - IdP Certificate**: Enter the SAML IdP certificate (generally the AD server certificate).

5. Click **Update Authentication Settings**. Your Director restarts and switches to SAML for authentication.

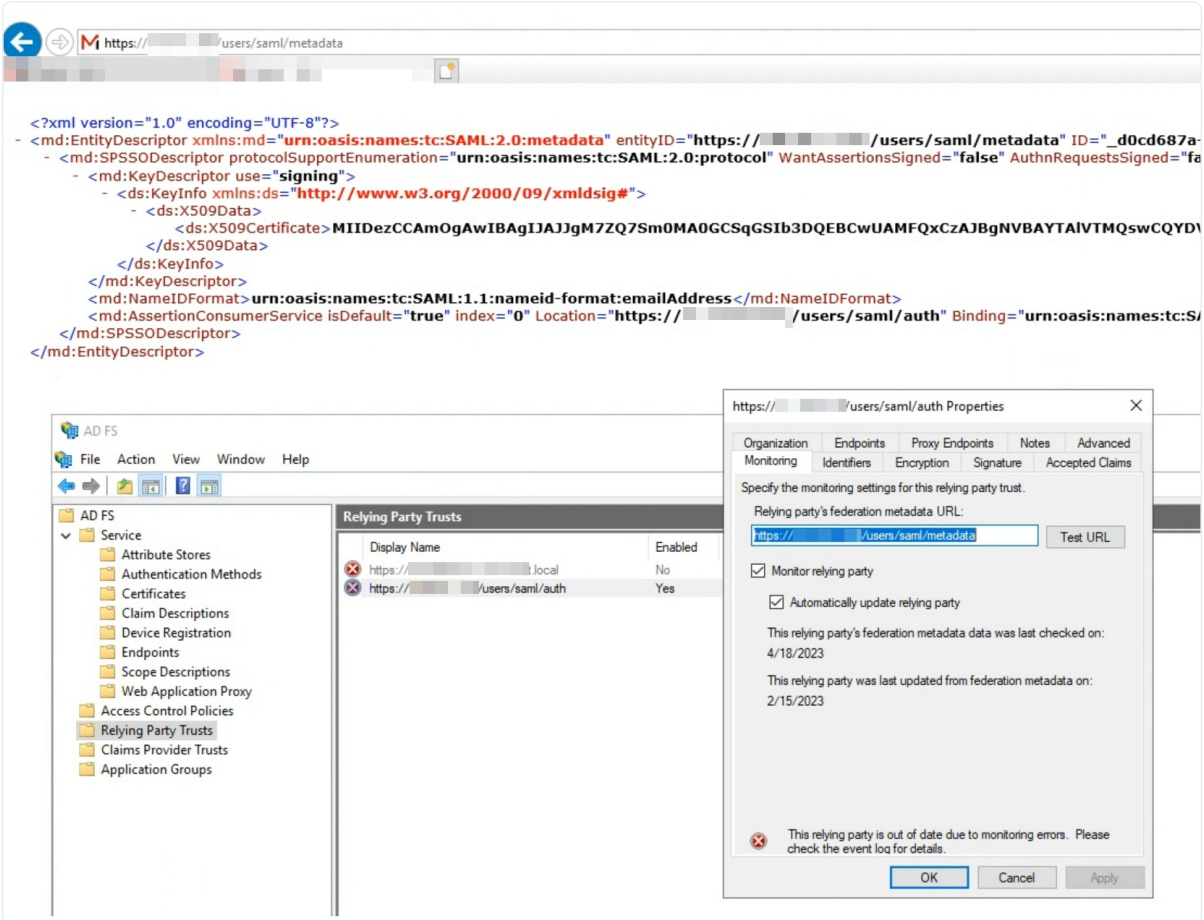
Establish Relying Party Trust

It's important to get the servers to trust one another. To accomplish this goal, you need a signed certificate on the Director. For example, you can issue a certificate using the DC cert services so that it's automatically trusted.

Create a DNS entry if you use the hostname as the subject for the Director.

1. Typically, you put in a request for the AD administrator to perform this task. The administrator needs the metadata URL, formatted as follows: `https://<DIRECTOR_IP>/users/saml/metadata`. This URL, including the IP address, becomes accessible once SAML is enabled on the Director.

Make sure you can access this URL. If you can't, neither can AD FS.



Relying Party Trust Metadata URL

If the test fails, have the AD administrator verify that the certificate is trusted when they browse to the metadata URL in their browser. They may need to use PowerShell or another CLI to verify the cert on the Active Directory server. Browsers tend to be disabled by security policy.

Test SAML Authentication on the Director

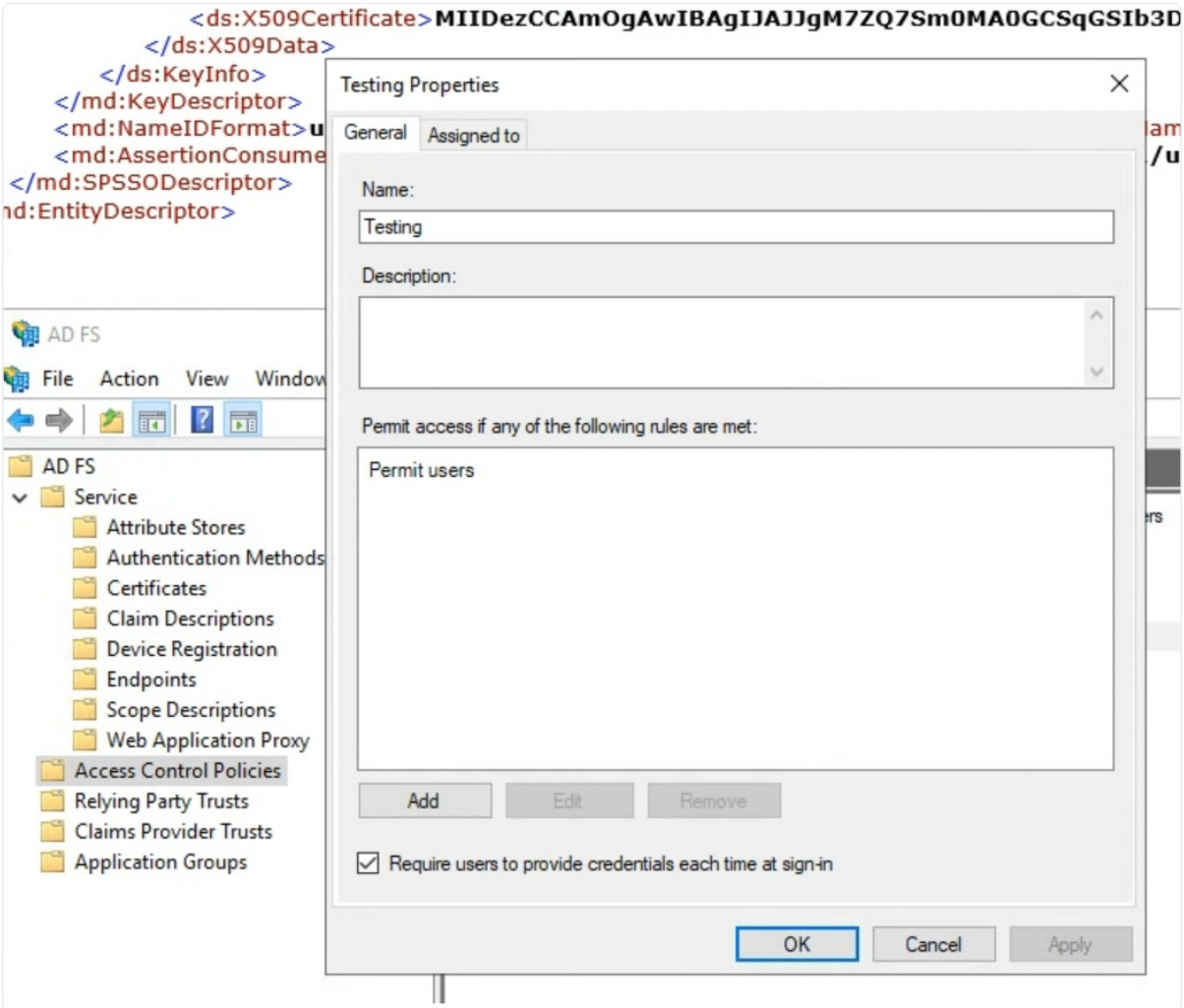
After you complete the setup and you created your user on the Director, you can sign out of the Director and then click **Login with SAML**. The SAML workflow follows these steps:

1. You ask to authenticate to an application.
2. The application asks the IdP to do the verification.
3. The application sends the IdP the SAML assertion fields that it wants to use to verify the user.
4. The application sends the user over to the IdP to enter their credentials.
5. The IdP authenticates the user internally based on the Authentication Context that it was given by the Director in the metadata that was already mentioned. If approved, the IdP redirects the user back to the application in an authenticated session. If failed, the user doesn't get redirected back to the application.

Troubleshoot SAML

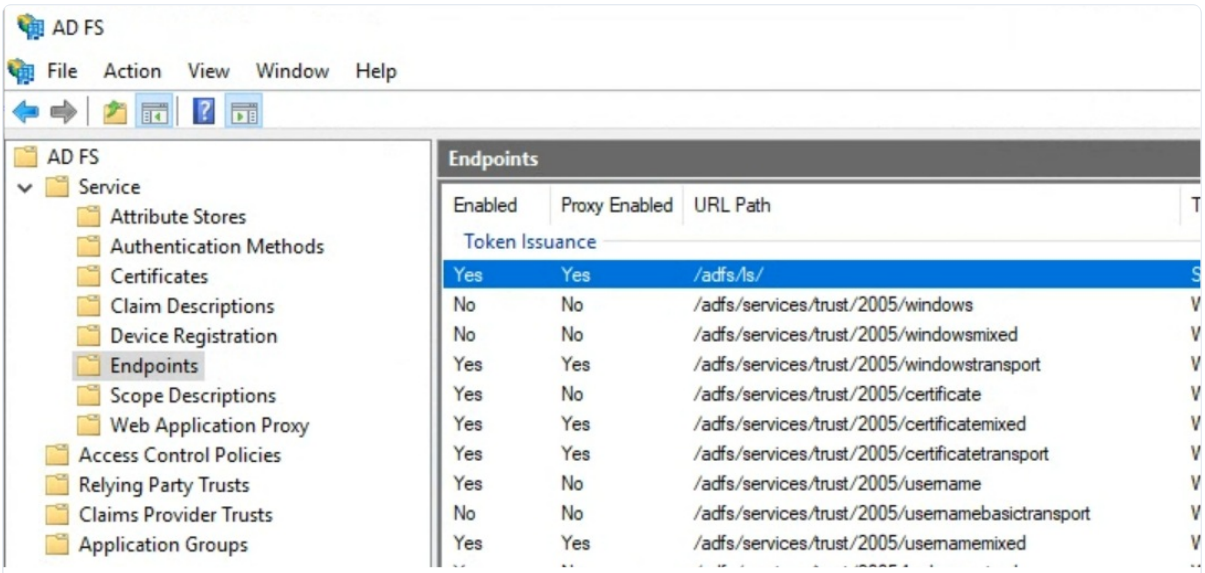
Troubleshooting SAML can be difficult, but some good steps are as follows:

- Use an incognito browser window.
- To reduce the need to open and close your incognito windows, configure a test policy with **Require users to provide credentials at each sign-in**, and link it to your Relying Party trust.



Test policy requiring users to provide credentials for each sign-in

- Make sure you have the correct endpoints enabled. The fields don't match up one for one. To keep things simple, make sure certificate* and /adfs/ls are enabled.



Enabled View of AD FS LS Endpoint

Additional Resources

- **Configuring the SAML Subject and Mapping Attributes** (https://www.ibm.com/docs/en/security-verify?topic=provider-configuring-saml-subject-mapping-attributes)
- **SAML V2.0 Technical Overview** (http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html)