Guide to Deploying NetScaler as an Active Directory Federation Services Proxy

Enabling seamless authentication for Office 365 use cases
# Table of Contents

- Introduction 3
- ADFS proxy deployment 4
- Microsoft recommendations for third-party ADFS proxies 5
- Deployment scenario and access flow with NetScaler as ADFS proxy 5
- Benefits of using NetScaler as ADFS proxy 6
- Configuration and setup details 6
  - Section A: Active clients / internal user configuration flow 7
  - Section B: Passive user configuration flow 20
- Conclusion 26
Recently, more enterprises are migrating to a cloud-based application deployment model. Thanks to cloud-based services such as Microsoft Office 365, this migration has accelerated. Cloud-based app deployment provides significant added value, but at the same time, it changes the underlying infrastructure for the enterprise. One of the critical services enterprise IT teams worry about is authentication for users connecting from within and outside the organization.

When migrating to the cloud, enterprises want to ensure the user experience does not change. However, seamless access to services hosted outside the enterprise data center requires a new component in app deployment design. No one wants the Active Directory password to travel on the wire outside the data center. Therefore, federation becomes a natural and proven alternative. Referring to primarily to Microsoft services, Active Directory Federation Services (ADFS) is the solution you are looking for. The ADFS security token service extends the single sign-on, (SSO) experience for Active Directory-authenticated clients to resources outside the enterprise data center.

An ADFS server farm allows internal users to access external cloud-hosted services. But the moment external users are brought into the mix, they must be given a way to connect remotely and access cloud-based services through federated identity. This is where an ADFS proxy plays a major role – giving external users SSO access to both internal federation-enabled resources as well as cloud resources such as Office 365. The purpose of the ADFS proxy server is to receive and forward requests to ADFS servers that are not accessible from the Internet.

The ADFS proxy plays critical role in remote user connectivity and application access. Citrix® NetScaler® has been playing similar roles – remote user connectivity and application access – for more than a decade. NetScaler has the right technology to enable secure connectivity, authentication and handling of federated identity, and thus it becomes the preferred solution for replacing an existing ADFS proxy or supporting a new ADFS implementation. Most enterprises want to reduce the footprint in the DMZ, and hence, they appreciate the fact that, in addition to its traditional functions, NetScaler can serve as ADFS proxy. This approach avoids the need to deploy an additional component in the DMZ.
Packet flow of how the ADFS proxy helps with external user access:
1. External user accesses internal or external applications enabled by ADFS.
2. User is redirected to the applicable federation service for authentication.
3. User is redirected to the enterprise's internal federation service.
4. User is connected to the ADFS proxy in the DMZ and is presented with a sign-on page.
5. ADFS proxy takes inputs from the external user and connects to the ADFS farm.
6. ADFS proxy presents external user credentials to the ADFS farm.
7. ADFS server authenticates the external user with enterprise Active Directory.
8. ADFS server returns authorization cookie with a signed security token and claims.
9. ADFS proxy sends the token and claim information to external user.
10. User connects to the federation service where the token and claims are verified.
11. Based on validation, the federation service provides the user with a new security token.
12. The external user provides the new authorization cookie with security token to the resource for access.

In most use cases you will run ADFS and the ADFS proxy farm, which would require load balancing and scale with high availability. If you are using the NetScaler ADC for load balancing of your ADFS proxy farm and other key services, only one additional step is needed to set up NetScaler as a replacement for the ADFS proxy farm. This means NetScaler does not just play the ADC role, but also assumes ownership of the processes performed by the ADFS proxy for external user access scenario.

NetScaler is a proven remote access solution for the DMZ. We can use the AAA for Traffic Management (AAA-TM) feature of NetScaler to fulfill the ADFS proxy use case while other product security features add to the overall value of this solution.
Microsoft recommendations for third-party ADFS proxies

<table>
<thead>
<tr>
<th>Microsoft Requirement and Recommendations</th>
<th>NetScaler Competancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proxy must not modify Response body</td>
<td>Yes</td>
</tr>
<tr>
<td>Proxy must pass through all HTTP headers to back-end STS</td>
<td>Yes</td>
</tr>
<tr>
<td>Proxy must not issue HTTP 302 responses</td>
<td>Yes</td>
</tr>
<tr>
<td>All requests must be passed through to ADFS farm</td>
<td>Yes</td>
</tr>
<tr>
<td>All external requests must be rerouted to back-end STS</td>
<td>Yes</td>
</tr>
<tr>
<td>Proxy must persist to same STS for multi-legged NTLM auth flow</td>
<td>Persistency</td>
</tr>
<tr>
<td>All requests to ADFS must be rerouted to same URL on back-end STS</td>
<td>Yes</td>
</tr>
<tr>
<td>Proxy must pass through all query string parameters</td>
<td>Yes</td>
</tr>
<tr>
<td>Proxy may provide form based login</td>
<td>AAA-TM</td>
</tr>
<tr>
<td>Proxy may use credentials to perform NTLM auth on ADFS</td>
<td>SSO</td>
</tr>
<tr>
<td>Proxy may also perform two factor auth as needed</td>
<td>AAA-TM</td>
</tr>
<tr>
<td>For Office 365 access scenarios, Proxy must provide additional info</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Deployment scenario and access flow with NetScaler as ADFS proxy

Packet flow of how NetScaler as ADFS proxy helps with internal/external user access:
1. Internal/external user access to Office 365 application is enabled by ADFS.
2. User is redirected to the applicable federation service for authentication.
3. User is redirected to the enterprise’s internal federation service.
4. Internal user is load balanced to the ADFS farm.
5. External user connects to NetScaler AAA-TM logon page.
6. User is authenticated against Active Directory or similar authentication service.
7. Post authentication, NetScaler does SSO (Kerberos/NTLM) to the ADFS farm.
8. ADFS server validates SSO credentials and returns STS token.
9. External user connects to the federation service where the token and claims are verified.
10. Based on validation, the federation service provides the user with a new security token.
11. External user provides authorization cookie with security token to the resource for access.
Here both internal and external users can go through the NetScaler path with the only difference being that external users are required to pre-authenticate with the NetScaler AAA-TM module. For this access scenario, the AAA-TM vserver must be set up on NetScaler for pre-authentication. Internal users can be directly load balanced to the ADFS server farm.

**Benefits of using NetScaler as ADFS proxy**

1. Caters to both load balancing and ADFS proxy needs
2. Works with both internal and external user access scenarios
3. Supports rich methods for pre-authentication and enables multi-factor authentication
4. Provides an SSO experience for end users
5. Supports both active and passive protocols
   a. Examples of active protocol apps – Outlook, Lync
   b. Examples of passive protocol apps – Outlook web app, browsers
6. Hardened device for DMZ-based deployment
7. Adds value with additional core ADC features
   a. Content Switching
   b. SSL offload
   c. Rewrite
   d. Responder
   e. Rate Limit
   f. Security

Note that for active protocol-based scenarios, users connect to Office 365 and provide their credentials. Microsoft Federation Gateway contacts the ADFS service on behalf of the active protocol client and submits their credentials. Post authentication, the ADFS service provides Federation Gateway with a token, which in turn is submitted to Office 365 to provide client access.

For active protocol-based use cases, clients typically authenticate on NetScaler using 401 NTLM. The configuration section below describes how to set up NetScaler for both active and passive protocol-based use cases.

**Configuration and setup details**

This guide provides the configuration workflow for active clients (Section A) as well as passive clients (Section B). Deployments covering both active and passive clients can follow section A and B sequentially for configuration flow.

The configuration given below is for external users. For internal users, use NetScaler as a load balancing vserv for the ADFS farm. If internal users have to be authenticated at by NetScaler, Section A configuration will suffice for both passive and active clients.

**NOTE:** For information on using NetScaler with Microsoft FAS, please refer to https://docs.citrix.com/en-us/xenapp-and-xendesktop/7-14/secure/federated-authentication-service/fas-architectures/fas-azure-ad.html
Section A: Active clients / internal user configuration flow
1. Create content switching vserver, bind SSL Certkey, bind CA certificate.

Content Switching Virtual Server

Basic Settings

<table>
<thead>
<tr>
<th>Name*</th>
<th>ADPS_Proxy_Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol*</td>
<td>SSL</td>
</tr>
<tr>
<td>IP Address Type*</td>
<td>IP Address</td>
</tr>
<tr>
<td>IP Address*</td>
<td>10.217.22.226 IPv6</td>
</tr>
<tr>
<td>Port*</td>
<td>443</td>
</tr>
</tbody>
</table>

Install Certificate

| Certificate-Key Pair Name* | nssp-dmm12 |
| Certificate and Key files are stored in the folder /nsconfig/ssl on appliance. |
| Certificate File Name* | nssp-dmm12.cer |
| Key File Name | dmm12key |
| Certificate Format | PEM |
| Password | |
| Certificate Bundle | |
| Notify When Expires | On |
| Notification Period | 30 |

Install Close
2. Create AAA vserver, bind SSL certificate, bind negotiate policy, bind session policy for Kerberos SSO. This vserver can be set to a private IP address as it is not accessed externally.

Now bind the server and CA certificate to this vserver as showed in step 1.

Please ensure that the proper DNS server is configured, which is required for client-side NTLM authentication as well as Kerberos SSO. If you have a single DNS server, create a Nameserver pointing to it. In the below configuration we are binding multiple DNS servers as services to the load balancing vserver.
2. Create AAA vserver, bind SSL certificate, bind negotiate policy, bind session policy for Kerberos SSO. This vserver can be set to a private IP address as it is not accessed externally.

Now bind the server and CA certificate to this vserver as showed in step 1.

Please ensure that the proper DNS server is configured, which is required for client-side NTLM authentication as well as Kerberos SSO. If you have a single DNS server, create a Nameserver pointing to it. In the below configuration we are binding multiple DNS servers as services to the load balancing vserver.
Create a negotiate action policy and bind it to the AAA vserver.
### Configure Authentication Negotiate Policy

**Name**
- Negotiate_DM12

**Authentication Type**
- NEGOTIATE

**Request Server**
- Negotiate_DM12

**Expression**
- ns_true

**Policies**

<table>
<thead>
<tr>
<th>Priority</th>
<th>Policy Name</th>
<th>Expression</th>
<th>Request Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Negotiate_DM12</td>
<td>ns_true</td>
<td>Negotiate_DM12</td>
</tr>
</tbody>
</table>

**Actions**
- **Add Binding**
- **Unbind**
- **Edit**

**Expression Editor**
- **Operators**
- **Saved Policy Expressions**
- **Frequently Used Expressions**
- **Clear**
Configure KCD Account

Name
kcd-dmnl2

Use Keytab File

Realm*
DMN12.NSI-TEST.COM

User Realm

Enterprise Realm

Service SPN

User Certificate

Browse

CA Certificate

Browse

Delegated User
Svc_Account_4NS

Password for Delegated User

Password

Confirm Password
Bind the session policy to the AAA vserver.
3. Create a default load balancing vserver that will send 401:Negotiate/NTLM response to authenticate the user and perform Kerberos SSO to the backend.

### Load Balancing Virtual Server

**Basic Settings**

Create a virtual server by specifying a name, an IP address, a port, and a protocol. In the virtual server IP (VIP) address, a public IP address is a public IP address. If the application is on a LAN or wide area network (WAN), the VIP is usually a private (ICANN non-routed) address. You can configure multiple virtual servers to receive client requests, thereby increasing the available capacity.

<table>
<thead>
<tr>
<th>Name*</th>
<th>Active_ADFS_server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol*</td>
<td>SSL</td>
</tr>
<tr>
<td>IP Address Type*</td>
<td>IP Address</td>
</tr>
<tr>
<td>IP Address*</td>
<td>10.217.22.226</td>
</tr>
<tr>
<td>Port*</td>
<td>443</td>
</tr>
</tbody>
</table>

**Create Server**

| Server Name* | ADFS-server |
| IP Address | Domain Name |
| Domain* | windows2012 DemocracyDomain |
| Traffic Domain | |
| Translation IP Address | 0.0.0.0 |
| Translation Mask | 0.0.0.0 |
| Resolve Retry (sec) | 5 |
| IPv6 Domain | |
| Enable after Creating | |
| Comments | |

---
### Load Balancing Service

#### Basic Settings

- **Service Name**: ADFS_service
- **New Server**: Existing Server
- **Server**: adfs-server (windows2012.dmn12)
- **Protocol**: SSL
- **Port**: 443
- **Traffic Domain**:
- **Hash ID**:
- **Server ID**: None
- **Clear Text Port**:
- **Cache Type**: SERVER

<table>
<thead>
<tr>
<th>Name</th>
<th>Active_ADFS_server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>SSL</td>
</tr>
<tr>
<td>State</td>
<td>Up</td>
</tr>
<tr>
<td>IP Address</td>
<td>10.217.222.226</td>
</tr>
<tr>
<td>Port</td>
<td>443</td>
</tr>
<tr>
<td>Traffic Domain</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Services and Service Groups

- **Load Balancing Virtual Server Service Binding**
- **Load Balancing Virtual Server Service Group Binding**
4. Create a load balancing vserver, which will simply pass the requests to the backend and convert the request URL from /adfs/services/trust to /adfs/services/trust/proxymex.
Bind server and CA certificate to the newly created vserver.

**Configure Rewrite Action**

- **Name**: replace_adfs_MEX_request
- **Type**: REPLACE

Use this action type to replace specified text reference with custom text in request/response.

**Expression to choose target location**

- Operators
- Saved Policy Expressions
- Frequently Used Expressions

```
http:requrl
```

**Expression**

```
'/adfs/services/trust/proxymex'
```

**Evaluate**
5. Create content switching policy for requests containing /adfs/services/trust and /federationmetadata/2007-06/federationmetadata.xml to go to the proxy server without any authentication.
6. Set the load balancing vserver with authentication enabled as the default vserver for the content switching vserver.
Section B: Passive user configuration flow
Note: we will use the same content switching vserver created in Section A but have different rules corresponding to passive clients.

1. Create AAA vserver, set authentication domain and bind LDAP policy.
   a. Create a KCD Account for Kerberos impersonation and a session policy for SSO.

<table>
<thead>
<tr>
<th>Server Name</th>
<th>IP Address</th>
<th>Security Type</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>ldap-dmn12</td>
<td>192.217.28.40</td>
<td>PLAINTEXT</td>
<td>389</td>
</tr>
</tbody>
</table>

Connection Settings

Base DN (location of users)
```
dc=dmn12,dc=nsi-test,dc=com
```

Administrator Bind DN
```
Administrator@nsi-test.com
```

Other Settings

Server Logon Name Attribute
```
-<><Name><>-= samaccountname
```

Default Authentication Group
```
ldap-grp
```

Create Authentication LDAP Policy

Name
```
ldap-dmn12
```

Server
```
ldap-dmn12
```

Expression
```
ne,site
```
Bind SSL server and CA certificate to the vserver.

2. Create a KCD account for Kerberos impersonation and ensure that DNS and NTP servers are configured properly. Create a session policy and bind it to the AAA vserver.
Deployment Guide

Deploying NetScaler as an ADFS Proxy

**Create Session Profile**

- **Name**: Passive_Proxy_SSO

  Unchecked Override Global check box indicates that the value is inherited from Global Session Parameters.

  - **Session Time-out (mins)**: 30
  - **Default Authorization Action**: ALLOW
  - **Single Sign-on to Web Applications**: ON
  - **Credential Index**: PRIMARY
  - **Single Sign-on Domain**: 
  - **HTTPOnly Cookie**: YES
  - **Enable Persistent Cookie**: ON
  - **Persistent Cookie Validity**: 3
  - **KCD Account**: impersonation_DMN12

**Configure Session Policy**

- **Name**: Proxy Passive_SSO

  - **Request Profile**: Passive_Proxy_SSO
  - **Expression**: ha_true

  Expression Editor
3. Create a load balancing vserver to handle requests /adfs/ls/auth/integrated (for ADFS 2.0) or /adfs/ls/wia (for ADFS3.0). Enable that vserver for form-based authentication.
4. Create a content switching action and policy and bind it to the content switching vserver.

**Create Content Switching Action**

Name*

ADFSPassive

Target Load Balancing Virtual Server

ADFSPassive

Comment

Create  Close
NetScaler is a proven solution for fast, reliable, high-availability and secure app delivery in remote access cases. Extending these capabilities to include functioning as ADFS proxy increases the total value NetScaler delivers to the enterprise. It becomes a single gateway point for all enterprise user access, including remote access to Office 365. Beyond its core functionality, NetScaler helps to improve the end-user experience and the scalability and stability of the whole deployment. Further, the same NetScaler appliance can also be used for other remote access cases, given that it is deployed in the DMZ. There is great value in consolidating all such remote access and authentication cases through a single NetScaler ADC appliance.
Conclusion
NetScaler is a proven solution for fast, reliable, high-availability and secure app delivery in remote access use cases. Extending these capabilities to include functioning as ADFS proxy increases the total value NetScaler delivers to the enterprise. It becomes single gateway point for all enterprise user access, including remote access to Office 365. Beyond its core functionality, NetScaler helps to improve the end-user experience and the scalability and stability of the whole deployment. Furthermore, the same NetScaler appliance can also be used for other remote access use cases, given that it is deployed in the DMZ. There is great value in consolidating all such remote access and authentication use cases through a single NetScaler ADC appliance.