WEBADM SAML AND OPENID IDP

WebADM SAML and OpenID IDP

Web-Service SSO Federation SAML OpenID

1. Overview

This document will present you how to use WebADM as Identity Provider (IDP) with different Service Provider (SP) which will consume OpenOTP for authentication processes. We will also see how we can configure and return different information per service provider through users/groups and client policies.

The installation of OpenID/SAML IdP is straightforward and only consists of running the self-installer and configure the application in WebADM.

You do not have to modify any files in the OpenID install directory! The web application configurations are managed and stored in LDAP by WebADM. To configure OpenID/SAML, just enter WebADM as super administrator and go to the 'Applications' menu. Click OpenID/SAML to enter the web-based configuration.

OpenID/SAML application logs are accessible in the Databases menu in WebADM.

Note: To be able to use OpenID/SAML, any LDAP user must be a WebADM account. That means usable LDAP accounts are those containing the webadmAccount LDAP object class. You can enable the WebADM features on any LDAP user/group by extending it with the webadmAccount object class (from object extension list).

Inline WebApps:

You can embed a Web app on your website in an HTML iFrame or Object.

```
#Example
<object data="https://<webadm_addr>/webapps/openid?inline=1" />
```

2. WebADM IDP configuration

2.1 SAML

SAML IdP requires a certificate and a private key to be set in the configurations. You can generate an X.509 certificate and private key with OpenSSL:

```
openssl genrsa -out server.key 1024
openssl req -new -key server.key -out server.csr
openssl x509 -req -days 3650 -in server.csr -signkey server.key -out server.crt
```

You can copy/paste the server.crt and server.key contents in your configuration.

The SAML clients (Service Providers) need to know about the SAML IdP endpoints. Most clients will accept the auto-configuration with an XML-based metadata URL. The server metadata URL is: `https://yourserver/webappd/openid/metadata/`. If you need to manually enter the IdP service URLs go to the HTML-based metadata URL with your Web browser: `https://yourserver/webapps/openid/metadata/html/`. You will find client configurations like:

- The SAML entityID of the IdP.
- The SAML server certificate.
- The SingleSignOnService URL.
- The SingleLogoutService URL.

> **Important**
>
> Many SAML Service Providers will require your WebADM to be run with a trusted SSL certificate. To use your own certificate and key, please have a look on Trusted Certificate documentation.

First, we need a WebADM server with *MFA Authentication Server* and *OpenID & SAML Provider*. We can use the appliance or install a new server.

We need also a DNS name for the server. If we can not change the DNS, we can also add the name in `/etc/hosts` or `c:\WINDOWS\system32\drivers\etc\hosts` for testing purpose:

Once the server is up and running, we can configure it as a SAML Identity Provider (IdP).

We connect to the WebADM GUI > `Applications` tab > `Singe Sign-On` > `OpenID & SAML Provider` > `REGISTER`:

We click on `CONFIGURE`:

We add the url of the server in `Issuer URL`:

At `Server Certificate`, we click on `Edit`:

We click on `Generate`:

Now, we have the IdP certificate, we click on `Apply`:

We can add some extra attributes, for example `mail` and `mobile`, and click on `Apply`:

That's all, WebADM is now able to work as an Identity Provider (IdP).

We can check metadata, go to WebADM GUI > `Applications` > `Single Sign-On` > `OpenID & SAML Provider` > `SAML Metadata` and open the link in a new tab:

```
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
entityID="https://webadm.local">
<IdPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
<KeyDescriptor use="signing">
<KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
<X509Data>
<X509Certificate>
MIICujCCAaKgAwIBAgIBAzANBgkqhkiG9w0BAQsFADAkMRIwEAYDVQQDDAlXZWJBRE0gQ0ExDjAMBgNVBAoMBUxv\

</X509Certificate>
<!--
 Cert Fingerprint (SHA1): 802b0a629dfc11a686306a73f8b11b272e1b9ca2
-->
<!--
 Cert Fingerprint (MD5): a0480b3a54a7ea7e2da2d6b9e27fbfbf
-->
</X509Data>
</KeyInfo>
</KeyDescriptor>
<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://webadm.local/webapps/openid/"/>
<SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://webadm.local/webapps/openid/"/>
<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://webadm.local/webapps/openid/"/>
<SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://webadm.local/webapps/openid/"/>
</IdPSSODescriptor>
</EntityDescriptor>
```

2.2 OpenID

Version 1.2x includes the support for OpenID-Connect and OAuth2.

To use your identity provider in OpenID-Connect mode, the client configuration must pass the scope ‘openid’ in the IdP requests. The supported OpenID-Connect scopes are: basic, email, phone and profile.

To use your identity provider in OAuth2 mode, the client must pass the scope ‘profile’ in the IdP requests.

If your client application needs the user’s email address, you can additionally request the openid email scope.

3. Configuration of a Service Provider

3.1 IDP initiated (SAML)

In this scenario, the authentication will be started directly from *OpenID & SAML Provider* web application. We will configure WebADM to manage authentications with Amazon Web Service (AWS). Other Service providers are available but not shown in this HowTo: GSuite, SalesForce, SugarCRM, Zimbra, GoToMeeting, GoToWebinar, GoToTraining and GoToAssist.

3.1.1 AWS SAML integration

3.1.1.1 SAML Configuration on AWS

First, we save the SAML metadata in a file. For our IdP server, we find it in `https://webadm.local/ws/saml/`.

We open AWS console > `IAM` > `Identity providers` > `Create Provider`:

We select `SAML`, add a name, insert the metadata file and click on `Next Step`:

We click on `Create`:

Now, our IdP is added to AWS. We select `Roles`:

We click on `Create Role`:

We click on `SAML`:

We select our SAML provider, select `AWS Management Console access` and click on `Next Permission`:

We select a permission policy and click on `Next: Review`.

We add a name and click on `Create role`:

The role is now created, we can select it to see more details.

3.1.1.2 Configure WebADM IDP for AWS

We need to activate IdP initiated authentication for AWS.

We open the configuration in WebADM GUI > `Applications` > `Single Sign-on` > `CONFIGURE`:

We check `Enable Application SSO` and `AmazonWS`, we add `AWS Account Number` (a numerical value that you can find in the ARN of the AWS role) and `AWS Provider Name` and apply:

We select the test user and click on `WebADM settings: [CONFIGURE]`:

We select `OpenID`, add `AWS Role Names` and `Apply`. We can also add the AWS role to an LDAP group:

3.1.1.3 AWS users/groups/clients policies

See more in section `4. How to create and match a client policy per service provider`. The example used is with AWS.

3.1.1.4 Testing/Debug

To test, open the web application in `https://webadm.local/webapps/openid/` and `Login` with the user:

We select `Application SSO`:

We click on `Amazon WS`:

That's it,we are now connected to AWS:

We can check the log in `/opt/webadm/logs/webadm.log`:

```
[2017-12-22 09:35:17] [192.168.1.220] [OpenID:4JGOGC0T] New login request (OpenOTP)
[2017-12-22 09:35:17] [192.168.1.220] [OpenID:4JGOGC0T] > Username: john
[2017-12-22 09:35:17] [192.168.1.220] [OpenID:4JGOGC0T] > Domain: Default
[2017-12-22 09:35:17] [192.168.1.220] [OpenID:4JGOGC0T] > ANY Password: xxxxxxx
[2017-12-22 09:35:17] [192.168.1.220] [OpenID:4JGOGC0T] Sending openotpSimpleLogin
request
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] New openotpSimpleLogin SOAP
request
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] > Username: john
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] > Domain: Default
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] > Password: xxxxxxx
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] > Client ID: OpenID
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] > Source IP: 192.168.1.220
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] > Context ID:
5cf415099b146265083580f7098f5717
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] Registered openotpSimpleLogin
request
[2017-12-22 09:35:17] [127.0.0.1] [OpenOTP:FFYIGQ6S] Resolved LDAP user: cn=john,o=Root
(cached)
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Started transaction lock for user
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Found 1 user mobiles: 123 456 789
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Found 1 user emails:
john.doe@acme.com
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Found 37 user settings:
LoginMode=LDAP,OTPType=TOKEN,OTPLength=6,ChallengeMode=Yes,ChallengeTimeout=90,MobileTime
1:HOTP-SHA1-6:QN06-
T1M,SMSType=Normal,SMSMode=Ondemand,MailMode=Ondemand,LastOTPTime=300,ListChallengeMode=S

[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Found 2 user data:
LoginCount,RejectCount
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Requested login factors: LDAP
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] LDAP password Ok
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Updated user data
[2017-12-22 09:35:18] [127.0.0.1] [OpenOTP:FFYIGQ6S] Sent success response
[2017-12-22 09:35:18] [192.168.1.220] [OpenID:4JGOGC0T] OpenOTP authentication success
[2017-12-22 09:35:18] [192.168.1.220] [OpenID:4JGOGC0T] Resolved LDAP user:
cn=john,o=Root (cached)
[2017-12-22 09:35:18] [192.168.1.220] [OpenID:4JGOGC0T] Login session started for
cn=john,o=Root
[2017-12-22 09:36:50] [192.168.1.220] [OpenID:4JGOGC0T] Sent SAML success response
```

3.2 SP-Initiated (SAML)

3.2.1 SimpleSAMLPHP

For this test, we are using simplesamplphp.

We install it on another *CentOS 7* server.

We open http port:

```
firewall-cmd --permanent --add-service http
firewall-cmd --reload
```

We disable selinux:

```
setenforce 0
vi /etc/selinux/config
```

We install required packages:

```
yum install wget php php-mbstring php-xml httpd
```

We install *simplesamlphp*:

```
wget "https://simplesamlphp.org/download?latest" -O ssp.tgz
tar xzf ssp.tgz
mv simplesamlphp* /var/simplesamlphp
```

We add a virtual host to *Apache* (replace *sp.local* with the right DNS name who point to this server):

```
vi /etc/httpd/conf.d/saml.conf
```

```
<VirtualHost *>
        ServerName sp.local
        DocumentRoot /var/www/sp.local

        SetEnv SIMPLESAMLPHP_CONFIG_DIR /var/simplesamlphp/config

        Alias /simplesaml /var/simplesamlphp/www

        <Directory /var/simplesamlphp/www>
            Require all granted
        </Directory>
</VirtualHost>
```

We add the Identity Provider. All these values should correspond to the content of metadata from SAML configuration in WebADM:

- *$metadata* corresponds to *entityID*
- *SingleSignOnService* corresponds to *SingleSignOnService Location=*
- *SingleLogoutService* corresponds to *SingleLogoutService Location=*
- *certFingerprint* corresponds to *Cert Fingerprint (SHA1)*

```
vi /var/simplesamlphp/metadata/saml20-IdP-remote.php
```

```
<?php
  $metadata['https://webadm.local'] = array(
    'SingleSignOnService'  => 'https://webadm.local/webapps/openid/',
    'SingleLogoutService'  => 'https://webadm.local/webapps/openid/',
    'certFingerprint'      => '802b0a629dfc11a686306a73f8b11b272e1b9ca2',
);
```

We enable *SAML* in `/var/simplesamlphp/config/config.php`:

```
vi /var/simplesamlphp/config/config.php
```

```
enable.saml20-IdP' => true
```

We start *Apache*:

```
systemctl start httpd
systemctl enable httpd
```

We open `http://sp.local/simplesaml` in a browser:

We click on `Authentication`:

We click on `Test configured authentication sources`:

We click on `default-sp`:

We click on `Select`:

We authenticate with an activated user through WebADM IdP:

It's done, we are authenticated:

We can check the log in `/opt/webadm/logs/webadm.log`:

```
[2017-12-21 11:16:31] [192.168.1.220] [OpenID:Y84I9XHY] User not authenticated
(entering login form)
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] New login request (OpenOTP)
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] > Username: john
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] > Domain: Default
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] > ANY Password: xxxxxxx
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] Sending openotpSimpleLogin
request
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] New openotpSimpleLogin SOAP
request
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] > Username: john
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] > Domain: Default
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] > Password: xxxxxxx
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] > Client ID: OpenID
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] > Source IP: 192.168.1.220
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] > Context ID:
5cf415099b146265083580f7098f5717
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Registered openotpSimpleLogin
request
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Resolved LDAP user: cn=john,o=Root
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Started transaction lock for user
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Found 1 user mobiles: 123 456 789
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Found 1 user emails:
john.doe@acme.com
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Found 37 user settings:
LoginMode=LDAP,OTPType=TOKEN,OTPLength=6,ChallengeMode=Yes,ChallengeTimeout=90,MobileTime
1:HOTP-SHA1-6:QN06-
T1M,SMSType=Normal,SMSMode=Ondemand,MailMode=Ondemand,LastOTPTime=300,ListChallengeMode=S

[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Found 1 user data: LoginCount
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Requested login factors: LDAP
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] LDAP password Ok
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Updated user data
[2017-12-21 11:16:36] [127.0.0.1] [OpenOTP:CADTGBMD] Sent success response
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] OpenOTP authentication success
[2017-12-21 11:16:36] [192.168.1.220] [OpenID:7TWF4J4E] Resolved LDAP user:
cn=john,o=Root (cached)
[2017-12-21 11:16:37] [192.168.1.220] [OpenID:7TWF4J4E] Login session started for
cn=john,o=Root
[2017-12-21 11:16:37] [192.168.1.220] [OpenID:7TWF4J4E] Sent SAML success response
```

3.2.2 Nextcloud

This was tested with Nextcloud 18.

3.2.2.1 Requirements

As a requirement, you need to install two apps in the app section:

- LDAP user and group backend docs.nextcloud.com
- SSO & SAML authentication apps.nextcloud.com

3.2.2.2 Configuration of "LDAP / AD integration" app

Then, you need to configure first the LDAP app to synchronize users stored in your LDAP server.

First, configure the connection to the LDAP server. You can adapt what is showed in the screenshot. You should get a green Configuration OK when settings are well-defined.

Figure 3. LDAP / AD integration (server configuration)

Next, you can adapt the search query in order to get right users from the LDAP.

Figure 4. LDAP / AD integration (user search query configuration)

Finally, configure the login attribute used to get the right username of users.

Figure 5. LDAP / AD integration (Login attribute configuration)

3.2.2.3 Configuration of "SSO & SAML authentication" app

3.2.2.4 Global Settings

On "Global Settings", it is only required to tick "Allow the use of multiple user back-ends (e.g. LDAP)", so IdP login initiation can work (See 2.1.2.4). If you still need to authenticate using a local account of Nextcloud, you can use the following URL to access the direct login mode: `https://yournextcloudserver/login?direct=1`

3.2.2.5 General

In the General section, you can set the following elements:

- Attribute to map the UID to. setting;

- Optional display name of the identity provider (default: “SSO & SAML log in”) setting.

3.2.2.6 Identity Provider Data

In the Identity Provider Data section, you have to set the following elements:

- Identifier of the IdP entity (must be a URI);
- URL Target of the IdP where the SP will send the Authentication Request Message;
- URL Location of the IdP where the SP will send the SLO Request. For these three first settings, you need to set the URL of root of openid (e.g. `https://yournextcloudserver/webapps/openid/`).

In order to set the Public X.509 certificate of the IdP setting, you can open saml URL (e.g. `https://yournextcloudserver/ws/saml/`) and copy and paste value contained in X509Certificate anchor.

3.2.2.7 Attribute mapping

Attribute mapping elements can also be set. Here, you can modify the following:

- Attribute to map the displayname to;
- Attribute to map the email address to;
- Attribute to map the quota to;
- Attribute to map the users groups to;
- Attribute to map the users home to;

Figure 6. SSO & SAML authentication (openid configuration)

3.3 Other examples (OpenID/SAML)

3.3.1 Apache Guacamole

First you need to install the OpenID extension to Apache Guacamole. See Guacamole documentation for instructions.

Please note that the authentication extensions in the GUACAMOLE_HOME/extensions directory are loaded in alphabetical order, so if you have another authentication extension which is alphabetically before the OpenID extension, then the OpenID extension will not be loaded. This is the case for example with guacamole-auth-jdbc-mysql extension. To bypass this issue you can rename the guacamole-auth-openid-1.0.0.jar to for example guacamole-auth-0penid-1.0.0.jar.

Once the extension is installed, you can configure the OpenID settings in GUACAMOLE_HOME/guacamole.properties

```
#OpenID authentication
openid-authorization-endpoint: https://<openotp_server_address>/openid/index.php
openid-jwks-endpoint: https://<openotp_server_address>/openid/certs.php
openid-issuer: https://<openotp_server_address>/webapps/openid/
openid-client-id: Guacamole
openid-redirect-uri: https://<guacamole_server_address>/guacamole/
```

Once the configuration is completed, you need to restart tomcat for it to take effect. If you want to log in as an existing Guacamole Admin user (for example guacadmin) while OpenID is enabled, you need to create that user in WebADM as well.

3.3.2 GitLab

This was tested with GitLab Enterprise Edition 13.2.1.

3.3.2.1 Requirements

The following LDAP attributes must be returned to SAML assertions to GitLab:

- first_name=givenname
- last_name=sn
- mail=mail

It is recommended to add this OpenID setting in a client policy specific to your GitLab instance. First create a client policy (you can name it GitLab) and put the client ID provided by GitLab (this can be found in the webadm.log file) in the “Client Name Aliases” setting:

Figure 1. GitLab (client policy configuration)

Next, still on the client policy, add to the “Forced Application Policies” setting the following to properly configure the returned attributes for the SAML assertion:

OpenID.ReturnAttrs="mail=mail,first_name=givenname,last_name=sn”

Figure 2. GitLab (client policy configuration)

3.3.2.2 Configuring SSO in GitLab

3.3.2.2.1 Enable SSO

First you need to enable SSO, and to permit auto creation of users.

You can add these lines for an Omnibus package installation to `config/gitlab.yml` file:

```
gitlab_rails['omniauth_allow_single_sign_on'] = ['saml']
gitlab_rails['omniauth_block_auto_created_users'] = false
gitlab_rails['omniauth_auto_link_saml_user'] = true
```

You can add these lines for a source installation to `config/gitlab.yml` file:

```
omniauth:
  enabled: true
  allow_single_sign_on: ["saml"]
  block_auto_created_users: false
  auto_link_saml_user: true
```

3.3.2.2.2 Add WebADM IdP

Next, you have to add the configuration of your IdP, still in `config/gitlab.yml` file.

The following parameters must be configured properly:

- **assertion_consumer_service_url**: this must match the URL of your gitlab, appended with `/users/auth/saml/callback`
- **idp_cert_fingerprint**: this is the fingerprint of the certificate provided by the SAML of your openotp. It can be retrieved using this command:

```
curl -ks https://youropenotp/ws/saml | grep SHA1 | awk '{print $5}' |  sed
's/../&:/g;s/:$//'
```

- **idp_sso_target_url**: this must match the URL domain of your openotp, appended with `/webapps/openid/index.php`
- **issuer**: this must be a unique name which will be used by openotp to identify your GitLab.
- **label**: this is the link name displayed on the sign-page to do SSO.

For an Omnibus package installation, add the following and adapt to your needs:

```
gitlab_rails['omniauth_providers'] = [
  {
    name: 'saml',
    args: {
            assertion_consumer_service_url:
'https://yourgitlab/users/auth/saml/callback',
            idp_cert_fingerprint: '43:51:43:a1:b5:fc:8b:b7:0a:3a:a9:b1:0f:66:73:a8',
            idp_sso_target_url: 'https://youropenotp/webapps/openid/index.php',
            issuer: 'https://yourgitlab',
            name_identifier_format: 'urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent'
          },
    label: 'Company Login' # optional label for SAML login button, defaults to "Saml"
  }
]
```

For a source installation, add the following and adapt to your needs:

```
omniauth:
  providers:
    - {
        name: 'saml',
        args: {
               assertion_consumer_service_url:
'https://gitlab.example.com/users/auth/saml/callback',
               idp_cert_fingerprint: '43:51:43:a1:b5:fc:8b:b7:0a:3a:a9:b1:0f:66:73:a8',
               idp_sso_target_url: 'https://youropenotp/webapps/openid/index.php',
               issuer: 'https://yourgitlab',
               name_identifier_format: 'urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent'
             },
        label: 'Company Login' # optional label for SAML login button, defaults to
"Saml"
      }
```

3.3.3 Graphana

First, create a new or update an existing Client Policy in WebADM > Admin > Client Policies. The policy name or friendly name must match the client_id defined in Grafana configuration (see below).

In the client policy, configure Application Settings > Edit > OpenID & SAML Provider > Client Secret. This secret must match the client_secret defined in Grafana.

Once these settings are applied, you can configure Grafana to use OpenOTP IdP for SSO login:

```
[auth.generic_oauth]
enabled = true
name = OpenOTP
allow_sign_up = true
client_id = grafana
client_secret = secret
scopes = openid profile email
auth_url = https://<openotp_server_address>/webapps/openid/index.php
token_url = https://<openotp_server_address>/webapps/openid/index.php
api_url = https://<openotp_server_address>/webapps/openid/index.php
tls_skip_verify_insecure = true
```

3.3.4 OnlyOffice

This was tested with OnlyOffice Enterprise Edition 10.5.3.

3.3.4.1 Requirements

The following LDAP attributes must be returned to SAML assertions to OnlyOffice (Location, Title, and Phone are optional attributes):

- givenName=givenname
- sn=sn
- mail=mail

It is recommended to add this OpenID setting in a client policy specific to your OnlyOffice instance. First create a client policy (you can name it OnlyOffice) and put the client ID provided by OnlyOffice (this can be found in the webadm.log file) in the “Client Name Aliases” setting:

Figure 7. OnlyOffice (client policy configuration)

Next, still on the client policy, add to the “Forced Application Policies” setting the following to properly configure the returned attributes for the SAML assertion:

OpenID.ReturnAttrs="givenName=givenname,sn=sn,mail=mail”

Figure 8. OnlyOffice (client policy configuration)

3.3.4.2 Configuring SSO in OnlyOffice

Open the following URL of your OnlyOffice: `https://youronlyoffice/controlpanel/sso`

Enable SSO, put the URL of your webadm (or waproxy if you have deployed one) in the “URL to IdP Metadata XML” field, and click on Load data button. This will pre-fill other input settings. You can click on the save button.

Figure 9. OnlyOffice (SSO configuration)

4. How to Create and match a client policy per Service Provider

Since the WebADM 1.6.9-x and OpenID/SAML provider 1.3.0, it is possible to create WebADM client policies per Service Provider. That will allow you to return attributes, nameID, attributes mappings, or use a different certificate per client (SP) and not only globally. This feature makes the IDP much more powerful.

4.1 SP Initiated mode

To create a client policy for your SP in SP initiated mode, log in on the WebADM Admin GUI, click on `Admin` tab, `Client Policy` and click on `Add Client`.

Give a name to your `Client Policy` and then click `Proceed` and `Create Object`.

We will now configure the client policy. Many settings can be applied here like which users/groups/networks the client policy will be applied, allowed/excluded hours, which domain… An important setting on this page is the `Client Name Aliases` which will allow us to do the matching between the client policy and the SP. For this, the client policy must be created with the SP issuer URL (Entity ID) as `Client Name Aliases`.

The matching is done, we will now configure the SP policy.

If you scroll down a little bit, you will find the setting named `Forced Application Policies`, click on the `Edit` button and select `OpenID` application in the left box.

Configure your client policy with every setting you need for your SP and then save your configuration.

Your client policy for your SP is now configured. Try an authentication from your SP and check the WebADM logs to be sure that

your policy is applied correctly.

> **Note**
>
> You can not yet apply any OpenOTP settings in the same OpenID/SAML client policy. That part is in the RCDevs roadmap and will be added in the future.

4.2 IDP initiated mode

The way to create create a client policy in IDP initiated mode is similar to SP initiated mode. The matching is done through the issuer value configured in the `app.ini` file located in `/opt/webadm/webapps/openid/apps/<application>.ini`

E.g for Amazon

```
[root@webadm1 ~]# cat opt/webadm/webapps/openid/apps/amazonws.ini

name   = "Amazon WS"
help   = "Amazon Web Services (AWS)"
method = "HTTP-POST"
source = "https://signin.aws.amazon.com/saml"
issuer = "https://signin.aws.amazon.com"
nameid = "Persistent"
```

I can then create my policy for AWS like below :

After creating the client policy object, I configure the client name alias for the matching operate :

In the next section, we show you how to return attributes for AWS SP.

4.3 Returned attributes

Here, I configured some returned attribute to be returned to AWS :

> **Note**
>
> You can not yet apply any OpenOTP settings in the same OpenID/SAML client policy. That part is in the RCDevs roadmap and will be added in the future.

4.4 Test login with AWS

My AWS service provider is now configured with my WebADM IDP. I can perform a login on OpenID & SAML Provider web application and access to AWS :

After a success login on the IDP, if no other SP are configured with your IDP, you are automatically redirected to AWS page :

After the redirection to AWS login page, you are prompted to select the role you want to use with your account. If multiple roles are configued under the user or group, then all role allowed by the user are returned and can be choosen by the end user :

Click `Sign In` button you are now connected to AWS with your account and the associated role.

5. Login debug

5.1 SAML request

To check your configued attributes are well returned by WebADM IDP in the SAML assertion, you can the browser extension SAML Message Decoder available on Chrome. Perform a login request and check the SAML Message Decoder console. You should see something similar :

```
<?xml version="1.0"?>
<samlp:Response Destination="https://signin.aws.amazon.com/saml"
    ID="_f8a62989fac5142a21d93c10fa6882e6f284b0314c" IssueInstant="2020-10-
26T09:26:46Z" Version="2.0"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
    <saml:Issuer>https://192.168.3.64/webapps/openid/</saml:Issuer>
    <samlp:Status><samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/></samlp:Status>
    <saml:Assertion ID="_5490a6d31dd1a3c782a48d0ec1e1541b16756ac843"
IssueInstant="2020-10-26T09:26:46Z"
        Version="2.0">
        <saml:Issuer>https://192.168.3.64/webapps/openid/</saml: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="#_5490a6d31dd1a3c782a48d0ec1e1541b16756ac843">
                    <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>qpLOfz9w9BlUANTvx7C7kB2DiImyIYHWjZYXNRvGPog=</ds:DigestValue>
                </ds:Reference>
            </ds:SignedInfo>
```

```
<ds:SignatureValue>WncS2uxIpx2uKX4MmDlNAXWgjNBS4ZFfNZdFjrp6EXXBUnQkNblL1kCGNWPnCgsbR9pQz

            <ds:KeyInfo>
                <ds:X509Data>

<ds:X509Certificate>MIIDBjCCAe6gAwIBAgIBAjANBgkqhkiG9w0BAQsFADAyMRkwFwYDVQQDDBBXZWJBRE0gC

                </ds:X509Data>
            </ds:KeyInfo>
        </ds:Signature>
        <saml:Subject>
            <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:emailAddress">yoan@rcdevs.com</saml:NameID>
            <saml:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer"><saml:SubjectConfirmationData
InResponseTo="" NotOnOrAfter="2020-10-26T09:27:46Z"

Recipient="https://signin.aws.amazon.com/saml"/></saml:SubjectConfirmation>
        </saml:Subject>
        <saml:Conditions NotBefore="2020-10-26T09:25:46Z" NotOnOrAfter="2020-10-
26T09:27:46Z">
            <saml:AudienceRestriction>
                <saml:Audience>https://signin.aws.amazon.com</saml:Audience>
            </saml:AudienceRestriction>
        </saml:Conditions>
        <saml:AuthnStatement AuthnInstant="2020-10-26T09:26:46Z" SessionIndex="1">
            <saml:AuthnContext>

<saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:Password</saml:AuthnCor

            </saml:AuthnContext>
        </saml:AuthnStatement>
        <saml:AttributeStatement>
            <saml:Attribute Name="uid">
                <saml:AttributeValue>administrator</saml:AttributeValue>
            </saml:Attribute>
            <saml:Attribute Name="domain">
                <saml:AttributeValue>yorcdevs.eu</saml:AttributeValue>
            </saml:Attribute>
            <saml:Attribute Name="group">
                <saml:AttributeValue>organization management</saml:AttributeValue>
                <saml:AttributeValue>group policy creator owners</saml:AttributeValue>
                <saml:AttributeValue>domain admins</saml:AttributeValue>
                <saml:AttributeValue>enterprise admins</saml:AttributeValue>
                <saml:AttributeValue>schema admins</saml:AttributeValue>
                <saml:AttributeValue>administrators</saml:AttributeValue>
                <saml:AttributeValue>denied rodc password replication
group</saml:AttributeValue>
            </saml:Attribute>
            <saml:Attribute Name="https://aws.amazon.com/SAML/Attributes/Role">

<saml:AttributeValue>arn:aws:iam::909745736108:role/112345678,arn:aws:iam::909745736108:s
```

```
provider/webadm1.yorcdevs.eu</saml:AttributeValue>
            </saml:Attribute>
            <saml:Attribute
Name="https://aws.amazon.com/SAML/Attributes/RoleSessionName">
                <saml:AttributeValue>administrator</saml:AttributeValue>
            </saml:Attribute>
            <saml:Attribute
Name="https://aws.amazon.com/SAML/Attributes/SessionDuration">
                <saml:AttributeValue>420</saml:AttributeValue>
            </saml:Attribute>
        </saml:AttributeStatement>
    </saml:Assertion>
</samlp:Response>
```

5.2 Login request on the IDP

The first step is the OpenID login request performed on the OpenID & SAML web application :

5.2.1 OpenID

It start with :

```
[Mon Oct 26 10:35:53.328922 2020] [192.170.3.23] [OpenID:GTZ09PU0] New login request
(OpenOTP)
[Mon Oct 26 10:35:53.328996 2020] [192.170.3.23] [OpenID:GTZ09PU0] > Client ID: OpenID
[Mon Oct 26 10:35:53.329012 2020] [192.170.3.23] [OpenID:GTZ09PU0] > Username:
administrator
[Mon Oct 26 10:35:53.329023 2020] [192.170.3.23] [OpenID:GTZ09PU0] > Domain:
yorcdevs.eu
[Mon Oct 26 10:35:53.329035 2020] [192.170.3.23] [OpenID:GTZ09PU0] > ANY Password:
xxxxxxxx
[Mon Oct 26 10:35:53.329058 2020] [192.170.3.23] [OpenID:GTZ09PU0] Sending
openotpSimpleLogin request
```

The last line of log indicate the login request is sent to OpenOTP. When OpenID call OpenOTP, the session number is the same for the OpenID request and the OpenOTP request (here GTZ09PU0). That allow you to easily identify different requests and products if you need to troubleshoot.

Then, the next part is the OpenOTP request and OpenID request continu after the OpenOTP request.

```
OpenOTP logs available in the next section
```

Below the OpenID session logs after the success login with OpenOTP :

```
[Mon Oct 26 10:35:59.608951 2020] [192.170.3.23] [OpenID:GTZ09PU0] OpenOTP
authentication success
[Mon Oct 26 10:35:59.609206 2020] [192.170.3.23] [OpenID:GTZ09PU0] Resolved LDAP user:
CN=Administrator,CN=Users,DC=yorcdevs,DC=eu (cached)
[Mon Oct 26 10:35:59.609399 2020] [192.170.3.23] [OpenID:GTZ09PU0] Resolved LDAP
groups: organization management,group policy creator owners,domain admins,enterprise
admins,schema admins,administrators,denied rodc password replication group
[Mon Oct 26 10:35:59.609660 2020] [192.170.3.23] [OpenID:GTZ09PU0] Resolved source
location: US
[Mon Oct 26 10:35:59.622375 2020] [192.170.3.23] [OpenID:GTZ09PU0] Login session
started for CN=Administrator,CN=Users,DC=yorcdevs,DC=eu
[Mon Oct 26 10:35:59.830787 2020] [192.170.3.23] [OpenID:GTZ09PU0] Enforcing client
policy: Amazon Web Service
[Mon Oct 26 10:35:59.830849 2020] [192.170.3.23] [OpenID:GTZ09PU0] Returning nameId
value: 'support@rcdevs.com'
[Mon Oct 26 10:35:59.847865 2020] [192.170.3.23] [OpenID:GTZ09PU0] Sent SAML login
success response
```

That part of the logs are important. It shows you the matching with the client policy previously created and the NameID value retuned.

5.2.2 OpenOTP

```
[Mon Oct 26 10:35:53.337483 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] New
openotpSimpleLogin SOAP request
[Mon Oct 26 10:35:53.337509 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] > Username:
administrator
[Mon Oct 26 10:35:53.337516 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] > Domain:
yorcdevs.eu
[Mon Oct 26 10:35:53.337525 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] > Password:
xxxxxxxx
[Mon Oct 26 10:35:53.337531 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] > Client ID: OpenID
[Mon Oct 26 10:35:53.337537 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] > Source IP:
192.170.3.23
[Mon Oct 26 10:35:53.337543 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] > Context ID:
578d78fb7b15a258ea414ffa9db4ebb2
[Mon Oct 26 10:35:53.337601 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Registered
openotpSimpleLogin request
[Mon Oct 26 10:35:53.338238 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Resolved LDAP user:
CN=Administrator,CN=Users,DC=yorcdevs,DC=eu (cached)
[Mon Oct 26 10:35:53.338472 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Resolved LDAP
groups: organization management,group policy creator owners,domain admins,enterprise
admins,schema admins,administrators,denied rodc password replication group
[Mon Oct 26 10:35:53.338718 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Resolved source
location: US
[Mon Oct 26 10:35:53.358316 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Started transaction
lock for user
[Mon Oct 26 10:35:53.370983 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Found user
```

```
fullname: Administrator
[Mon Oct 26 10:35:53.371005 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Found user
language: EN
[Mon Oct 26 10:35:53.371018 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Found 1 user
mobiles: 123456
[Mon Oct 26 10:35:53.371025 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Found 1 user
emails: support@rcdevs.com
[Mon Oct 26 10:35:53.371467 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Found 48 user
settings:
LoginMode=LDAPOTP,OTPType=TOKEN,OTPFallback=MAIL,PushLogin=Yes,ChallengeMode=Yes,Challeng
1:HOTP-SHA1-6:QN06-
T1M,DeviceType=U2F,SMSType=Normal,SMSMode=Ondemand,MailMode=Ondemand,PrefetchExpire=10,La
[5 Items]
[Mon Oct 26 10:35:53.372017 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Found 5 user data:
TokenType,TokenKey,TokenState,TokenID,TokenSerial
[Mon Oct 26 10:35:53.372085 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Found 1 registered
OTP token (TOTP)
[Mon Oct 26 10:35:53.372112 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Requested login
factors: LDAP & OTP
[Mon Oct 26 10:35:53.382710 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] LDAP password Ok
[Mon Oct 26 10:35:53.383006 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Authentication
challenge required
[Mon Oct 26 10:35:53.564385 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Sent push
notification for token #1
[Mon Oct 26 10:35:53.564427 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Waiting 28 seconds
for mobile response
[Mon Oct 26 10:35:59.598111 2020] [192.168.3.56] [OpenOTP:GTZ09PU0] Received mobile
authentication response from 192.170.3.27
[Mon Oct 26 10:35:59.598145 2020] [192.168.3.56] [OpenOTP:GTZ09PU0] > Session:
QIO1HmdExVHo9kr1
[Mon Oct 26 10:35:59.598152 2020] [192.168.3.56] [OpenOTP:GTZ09PU0] > Password: 16
Bytes
[Mon Oct 26 10:35:59.598158 2020] [192.168.3.56] [OpenOTP:GTZ09PU0] Found
authentication session started 2020-10-26 10:35:53
[Mon Oct 26 10:35:59.598252 2020] [192.168.3.56] [OpenOTP:GTZ09PU0] PUSH password Ok
(token #1)
[Mon Oct 26 10:35:59.605533 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Updated user data
[Mon Oct 26 10:35:59.607544 2020] [192.168.3.64] [OpenOTP:GTZ09PU0] Sent login success
response
```