eduGAIN

Overview

eduGAIN is a global federation of identity and service providers, based technically on SAML2.

In order to allow eduGAIN users to access Waldur, there are two steps:

- Waldur deployment must be registered as a service provider in eduGAIN federation.
- Waldur must get a list of identities that are trusted for authentication.

> **Tip**
>
> SAML is a complicated and fragile technology. GEANT provides an alternative to direct integration of SAML - eduTEAMS, which exposes an OpenID Connect protocol for service providers.

Waldur relies on djangosaml2 for the heavylifting of SAML processing, so for fine tuning configuration, contact corresponding project documentation.

Registering Waldur as Service Provider

Add SAML configuration to Waldur Mastermind configuration

Example configuration is below, please adjust to your specific deployment. Once applied, service metadata will be visible at Waldur deployment URL: `https://waldur.example.com/api-auth/saml2/metadata/`. That data needs to be propagated to the federation operator for inclusion into the federation.

> **Tip**
>
> Managed ansible simplifies configuration of the eduGAIN integration and should be a preferred method for all supported deployments.

```
import datetime

import saml2
from saml2.entity_category.edugain import COC


WALDUR_AUTH_SAML2 = {
    # used for assigning the registration method to the user
    'name': 'saml2',
    # full path to the xmlsec1 binary program
    'xmlsec_binary': '/usr/bin/xmlsec1',
    # required for assertion consumer, single logout services and entity
ID
    'base_url': '',
    # directory with attribute mapping
    'attribute_map_dir': '',
    # set to True to output debugging information
    'debug': False,
    # IdPs metadata XML files stored locally
    'idp_metadata_local': [],
    # IdPs metadata XML files stored remotely
    'idp_metadata_remote': [],
    # logging
    # empty to disable logging SAML2-related stuff to file
    'log_file': '',
    'log_level': 'INFO',
    # Indicates if the entity will sign the logout requests
    'logout_requests_signed': 'true',
    # Indicates if the authentication requests sent should be signed by
default
    'authn_requests_signed': 'true',
    # Identifies the Signature algorithm URL according to the XML
Signature specification
    # SHA1 is used by default
    'signature_algorithm': None,
    # Identifies the Message Digest algorithm URL according to the XML
Signature specification
    # SHA1 is used by default
    'digest_algorithm': None,
    # Identified NameID format to use. None means default, empty string
("") disables addition of entity
    'nameid_format': None,
    # PEM formatted certificate chain file
    'cert_file': '',
    # PEM formatted certificate key file
    'key_file': '',
    # SAML attributes that are required to identify a user
    'required_attributes': [],
    # SAML attributes that may be useful to have but not required
    'optional_attributes': [],
    # mapping between SAML attributes and User fields
    'saml_attribute_mapping': {},
    # organization responsible for the service
    # you can set multilanguage information here
    'organization': {},
    # links to the entity categories
    'categories': [COC],
```

Example of generated metadata

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:EntityDescriptor xmlns:ns0="urn:oasis:names:tc:SAML:2.0:metadata"
xmlns:ns1="urn:oasis:names:tc:SAML:metadata:attribute"
xmlns:ns2="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:ns4="urn:oasis:names:tc:SAML:metadata:algsupport"
xmlns:ns5="urn:oasis:names:tc:SAML:metadata:rpi"
xmlns:ns6="urn:oasis:names:tc:SAML:metadata:ui" xmlns:ns7="http://
www.w3.org/2000/09/xmldsig#" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" entityID="https://api.etais.ee/api-auth/saml2/metadata/">
   <ns0:Extensions>
      <ns1:EntityAttributes>
         <ns2:Attribute Name="http://macedir.org/entity-category"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
            <ns2:AttributeValue xmlns:xs="http://www.w3.org/2001/
XMLSchema" xsi:type="xs:string">http://www.geant.net/uri/dataprotection-
code-of-conduct/v1</ns2:AttributeValue>
         </ns2:Attribute>
      </ns1:EntityAttributes>
      <ns4:DigestMethod Algorithm="http://www.w3.org/2000/09/
xmldsig#sha1" />
      <ns4:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#sha224" />
      <ns4:DigestMethod Algorithm="http://www.w3.org/2001/04/
xmlenc#sha256" />
      <ns4:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#sha384" />
      <ns4:DigestMethod Algorithm="http://www.w3.org/2001/04/
xmlenc#sha512" />
      <ns4:SigningMethod Algorithm="http://www.w3.org/2000/09/
xmldsig#dsa-sha1" />
      <ns4:SigningMethod Algorithm="http://www.w3.org/2009/xmldsig11#dsa-
sha256" />
      <ns4:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#ecdsa-sha1" />
      <ns4:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#ecdsa-sha224" />
      <ns4:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#ecdsa-sha256" />
      <ns4:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#ecdsa-sha384" />
      <ns4:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#ecdsa-sha512" />
      <ns4:SigningMethod Algorithm="http://www.w3.org/2000/09/
xmldsig#rsa-sha1" />
      <ns4:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#rsa-sha224" />
      <ns4:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#rsa-sha256" />
      <ns4:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#rsa-sha384" />
      <ns4:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#rsa-sha512" />
      <ns5:RegistrationInfo registrationAuthority="http://taat.edu.ee"
registrationInstant="2017-01-01T00:00:00">
         <ns5:RegistrationPolicy xml:lang="en">http://taat.edu.ee/main/
wp-content/uploads/Federation_Policy_1.3.pdf</ns5:RegistrationPolicy>
      </ns5:RegistrationInfo>
```

Adding trusted identity providers

In order to configure Waldur to use SAML2 authentication you should specify identity provider metadata.

- If metadata XML is stored locally, it is cached in the local SQL database. Usually metadata XML file is big, so it is necessary to use local cache in this case. But you should ensure that metadata XML file is refreshed via cron on a regular basis. A management command `waldur sync_saml2_providers` performs refreshing of the data.
- If metadata XML is accessed remotely, it is not cached in SQL database. Therefore you should ensure that metadata XML is small enough. In this case you should download metadata signing certificate locally and specify its path in Waldur configuration. The certificate is used to retrieve the metadata securely. Please note that security certificates are updated regularly, therefore you should update configuration whenever certificate is updated. By convention, both metadata signing certificate and metadata itself are downloaded to `/etc/waldur/saml2` in Waldur Mastermind instances.

References

TAAT configuration

TaaT certificates can be downloaded from: http://taat.edu.ee/main/dokumendid/sertifikaadid/.

Metadata URL for test hub is https://reos.taat.edu.ee/saml2/idp/metadata.php and for production hub is https://sarvik.taat.edu.ee/saml2/idp/metadata.php. Note, the certificate must correspond to the hub you want connect to.

Using Janus

Janus is a self-service for managing Service Provider records.

- Create a new connection:

New connection ID must be equal to the base_url in saml.conf.py + /apu-auth/saml2/metadata/

- Choose SAML 2.0 SP for connection type.
- Click Create button
- In connection tab select or create ARP. Fields that ARP include must be in the saml_attribute_mapping.

- Navigate to the Import metadata tab and paste same URL as in the first step. Click on the Get metadata.
- Navigate to the Validate tab and check whether all the tests pass. You can fix metadata in Metadata tab.

HAKA configuration

Production hub metadata is described at https://wiki.eduuni.fi/display/CSCHAKA/Haka+metadata.

Test hub metadata is described at https://wiki.eduuni.fi/display/CSCHAKA/Verifying+Haka+compatibility.

FEDI configuration

Production hub metadata is described at https://fedi.litnet.lt/en/metadata.

Discovery is supported: https://discovery.litnet.lt/simplesaml/module.php/discopower/disco.php.

Last update: 2021-05-03