Authorization configuration - CustomerID

Ubisecure CustomerID SSO Adapter is a critical component in the authentication process. If it is not installed or if it is configured incorrectly, authentication will fail. For more detailed information on installing and upgrading Ubisecure CustomerID SSO Adapter, please refer to the pages *Installation - CustomerID and Upgrade - CustomerID* .

Ubisecure CustomerID SSO Adapter can be used as such if no special role sets are required for Ubisecure CustomerID service providers. Often it is required that only certain roles are included to authentication responses for different service providers (applications).

The Ubisecure CustomerID SSO Adapter authorization configuration is located in a file called `eidm2-authorizer.properties`. By default the configuration file is empty, which is sufficient for a standard Ubisecure CustomerID installation which utilizes the Ubisecure Directory for resolving user's roles.

Policies

If needed, different role sets can be defined in the configuration file for use with different services. These sets are called policies and can be defined as follows:

policy.N.name

Defines the name of the policy. The policy name is later used in Ubisecure SSO Management with `eidm:roles:<name>` when defining the authorization policies for service providers.

policy.N.include

Specify the way in which roles are included or excluded. Valid values are *whitelist* and *blacklist*. White listing causes only the defined roles to be included in user's final role set. Black listing causes the defined roles to be omitted from user's roles. If this key is omitted, black listing will be used.

policy.N.roles.M

Roles for a policy are defined using separate key for each role where M starts from 1 onwards.

policy.N.mapping.M

Policies can contain zero to several role mappings. A role mapping translates a role to another role. This allows for instance several roles to be seen as one by the target application. Role mappings can also be used to map any role string to arbitrary role string. Role mapping for a policy are defined with key *policy.N.mapping.M*, where M runs from 1 onwards and keys' values are names of role mappings defined elsewhere in the configuration.

```
# example role mapping
policy.1.mapping.1 = mapping # name of the mapping can be an arbitrary string
policy.1.mapping.2 = mapping2
```

Role Mappings

Role mappings provide a way to translate role names in the Ubisecure CustomerID system to other arbitrary strings in the respective entity scope of the original role. Only mappings used by the policies are read by the authorizer. A mapping is defined by two keys; first key is the name of the mapping as written in a policy definition and its value is the role that should be translated, the second key is the mapping name suffixed with ".name" and its value is the string which will be the resulting role name.

Consider the following mapping:

```
mapping = eIDMUser
mapping.name = normalUser
```

This would translate all the *eIDMUser* roles of a user to *normalUser* where the policy using this mapping is used.

Example Configuration

For instance, consider the following configuration:

```
# policy
policy.1.name =
policy.1.include = whitelist
policy.1.roles.1 = OrganizationMainUser
policy.1.roles.2 = OrganizationUser
policy.1.mapping.1 = defaultUserMapping
policy.1.mapping.2 = adminUserMapping

#role mappings
# OrganizationUser -> defaultUser
defaultUserMapping = OrganizationUser
defaultUserMapping.name = defaultUser
# OrganizationMainUser -> adminUser
adminUserMapping = OrganizationMainUser
adminUserMapping.name = adminUser
```

This defines a policy which only relays user's roles *OrganizationMainUser* and *OrganizationUser* translating them to *adminUser* and *defaultUser* respectively.

Organization Path

Beginning with Ubisecure CustomerID 2.11.1, it is possible to send user's hierarchical organization name to the application that sits behind the Ubisecure SSO web agent using a configured authorization policy. Many times user's organization is in some form of a hierarchy and the application might need to know this hierarchy. To send the hierarchical organization name, use *eidm:organization* in the authorization policy attribute value. Here is an example that would send hierarchical organization name in an attribute called organization:

```
organization eidm:organization
```

You can add this configuration to the Ubisecure SSO's authorization policy using the Ubisecure SSO Management web application's policy editor.

Structured Role Organization Information

Beginning with Ubisecure CustomerID 4.4.0 authorizer can provide structured information about user's roles' respective organizations in a JSON format. These data can be included in the authorization policy by invoking the authorizer with orgclaims parameter, i.e., using 'eidm:orgclaims' as the value in an authorization policy's attribute.

The parameter supports different authorization role filtering and mappings similar to the eidm:roles operation (see the examples above). The desired policy should can be given to the operation as a parameter separated by colon, i.e., eidm:orgclaims:<policy name>.
Using an authorization policy rule:

```
roleorgs eidm:orgclaims
```

will insert a JSON array in the roleorgs attribute of the authentication response with organization information. The response JSON contains basic data and attributes of all the organizations where a user has any roles.

Example Orgclaims Response

User has two roles Customers/1234/Representative and Organizations/OrganizationUser. The following is an example of a JSON response for the user.

```
[
{ "organizationClass" : "CustomerClass", "customerid"
: "1234", "technicalName" : "1234",
"roles": ["Customers\/1234\/Representative"],
"friendlyName" : "Customer 1234",
"entityName":"Customers\/1234" },
{ "organizationClass" : "OrganizationClass",
"technicalName" : "Organizations", "roles" :
["Organizations\/OrganizationUser"], "friendlyName" :
"Organizations container", "entityName":"Organizations"
}
]
```

Structured Role Delegation Information

Beginning with Ubisecure CustomerID 4.6.1 authorizer can provide information about delegated roles to the user in a JSON format. This data can be included in the authorization policy by invoking the authorizer with delegations parameter, i.e. using *eidm:delegations* as the value in an authorization policy's attribute.

Example Delegations Response

The following is an example of a JSON response.

```
[
{
"role":"Services/Service1/Role1",
"mandate":"<mandate ID>",
"organization":"UserOrgs/UserOrg1" },
{
"role":"Services/Service2/Role2",
"mandate":"<mandate ID>",
"organization":"UserOrgs/UserOrg2" }
]
```

Authorization Policies

In this section, we explain how roles and user attributes will be passed on to the applications/services in SAML messages during the authentication and authorization process.

The authorization policy is used to define the data that is delivered to the applications behind Service Providers (SP). In practice, the authorization policy adds attributes with a name and a value to the SAML message for a web agent. No other attributes are delivered to the service provider. In this way, the information exposed about a user to applications can be restricted. For data security, it is best practice to send only the minimum amount of information required by the web agent. For this reason, it is strongly recommended to use an authorization policy for all service providers.

You can add authorization policies using the Ubisecure SSO Management web application's policy editor.
Usually an authorization policy contains all or some of the user's Ubisecure CustomerID roles. It may also contain other application-specific roles that are based on user's group memberships. In addition to role definitions also user attributes are commonly defined in the authorization policy. User attributes can be collected from the main user authentication data repository (default user attributes) or from the internal Ubisecure CustomerID database (custom user attributes). You may also select attributes from the organization under which the user is stored (organization attributes). Ubisecure CustomerID creates an authorization policy for itself but for the application, the configuration of authorization policies is done using the Ubisecure SSO Management application.

> **NOTE:** The use of user or organization custom attributes requires that the Ubisecure CustomerID SSO Adapter has been installed from Ubisecure CustomerID release 3.4.0 or newer and Ubisecure SSO version is 6.3.1 or newer.

Formats of Different Types of Policy Values:

```
Ubisecure CustomerID roles:                         eidm:roles
Ubisecure CustomerID limited roles:                eidm:roles:<policy name>
Default user attributes:                                user:<attribute name>
Custom user attributes:                                 eidm:user:<attribute name>
Organization attributes:                                 eidm:organization:<attribute name>
```

Example Configuration:

```
Name                                Group                   Value
--------------------------------------------------------
email                                 eIDMUser                 user:mail
firstname                         eIDMUser                 user:givenName
surname                         eIDMUser                 user:sn
mobile                                 eIDMUser                 user:mobile
organization                eIDMUser                 user:o
organizationName        eIDMUser                 user:../description
username                         eIDMUser                 user:uid
uniqueid                         eIDMUser                 user:cn
customerid                         eIDMUser                 eidm:customerid
role                                 eIDMUser                 eidm:roles
userid                                 eIDMUser                 eidm:user:userid
```

Another example concerning custom attributes:

```
customerid                                                 eidm:user:customerid
org.companyid                                         eidm:organization:companyid
```