40_configuration

Оглавление

Конфигурирование

Подход к конфигурированию комплекса - Configuration As a Code

Система шаблонизации, принятая по умолчанию

Система шаблонизации для Kubernetes - HELM

Система шаблонизации - Ansible

Файлы конфигурации ПО RooxConfig

config.xml

env.properties

Параметры сервиса аутентификации sso-server

Настройка агентов

Для доступа к RooX UIDM API используются так называемые агенты (они же - clients)

Агенты могут быть следующих типов:

- **m2m** (machine to machine) - для межсерверного взаимодействия
- **m2m public** - для аутентификации без редиректов **oauth2**
- **web** - для реализации стандартных сценариев **oauth2**

Добавление нового агента типа m2m

Настроим для примера агента типа **m2m** с именем **antifraud**:

- в директории /opt/roox-sso-01/conf/ создаем файл с настройками агента

agent_antifraud.conf

```
com.forgerock.openam.oauth2provider.autoGrant=true
clientEntryPointType=oauth2_m2m
clientName=antifraud
com.forgerock.openam.oauth2provider.defaultScopes[0]=cn
com.forgerock.openam.oauth2provider.defaultScopes[1]=sn
```

- в этой же директории в файле openam_tune_batch_2.conf добавим команды по созданию агента (не забыв задать пароль):

```
create-agent --realm /customer --agenttype OAuth2Client --agentname "antifraud" --attributevalues "userpassword=ПАРОЛЬ_АГЕНТА"
update-agent --realm /customer --agentname "antifraud" --attributevalues "sunIdentityServerDeviceKeyValue[0]=ROLE=ROLE_ANTIFRAUD"
update-agent --realm /customer --agentname "antifraud" -s --datafile /opt/roox-sso-01/conf/agent_antifraud.conf
```

ЗАМЕТКА

после добавления\изменения настроек агентов необходимо сделать реконфигурацию RooX UIDM

Параметры сервиса доставки сообщений пользователям webapi-notifier

Подключение к ActiveDirectory либо другим LDAP хранилищам учетных записей пользователей

1. Указать настройки для подключения

Так как UIDM поддерживает работу с несколькими источниками записей, то настройки подключения указываются в формате:

```
com.rooxteam.uidm.ldapv3repository.{source_id}.{property_name}={property_value}
```

где {source_id} - это условное название для набора значений, по которым можно сгруппировать УЗ пользователей. Например: пользователи находятся в разных кустах одного домена, на разных серверах одного леса, либо часть из них в AD, а часть в OpenLDAP.

Указание {source_id} можно опустить, если предполагается работа с одним набором УЗ

Описание настройки	Название	Название для шаблонизации	Пример
Включение источника	com.rooxteam.uidm.ldapv3repository.enabled	.Values.rx.uidm.ldapv3repository.enabled	true
Имя пользователя от имени которого sso-server будет выполнять поиск данных	com.rooxteam.uidm.ldapv3repository.superuserId	.Values.rx.uidm.ldapv3repository.superuserId	cn=uidm_super_user\,cn=Users\,dc=demo\,dc=ex
Пароль для пользователя от имени которого sso-server будет выполнять поиск данных	com.rooxteam.uidm.ldapv3repository.superuserPassword	.Values.rx.uidm.ldapsv3repository.superuserPassword	SuperUserPa$$w0rd
Имя сервера	com.rooxteam.uidm.ldapv3repository.ldapServerName	.Values.rx.uidm.ldapv3repository.ldapServerName	dc.example.com
Порт для подключения к серверу	com.rooxteam.uidm.ldapv3repository.ldapPort	.Values.rx.uidm.ldapv3repository.ldapPort	389
Ветка, от которой начинается поиск данных	com.rooxteam.uidm.ldapv3repository.baseDN	.Values.rx.uidm.ldapv3repository.baseDN	ou=Users\,dc=example\,dc=com
Список аттрибутов, которые sso-server будет синхронизировать для пользователя	com.rooxteam.uidm.ldapv3repository.profileAttributesToFetch	.Values.rx.uidm.ldapv3repository.profileAttributesToFetch	displayName,memberOf,mail,mobile,sAMAccountNar
	com.rooxteam.uidm.ldapv3repository.namingAttribute	.Values.rx.uidm.ldapv3repository.namingAttribute	userPrincipalName
	com.rooxteam.uidm.ldapv3repository.attributes	.Values.rx.uidm.ldapv3repository.attributes	DISPLAY_NAME_ATTRIBUTE,EMAIL_ATTRIBUTE,MOB
Название аттрибута, который будет использоваться для идентификации пользователя	com.rooxteam.uidm.ldapv3repository.attributes.use_as_login	.Values.rx.uidm.ldapv3repository.attributes_use_as_login	userPrincipalName
Строка-фильтр, которая будет использоваться при поиске УЗ в LDAP-репозитории	com.rooxteam.uidm.ldapv3repository.userSearchFilter	.Values.rx.uidm.ldapv3repository.userSearchFilter	
Флаг, который указывает использовать защищенное соединение	com.rooxteam.uidm.ldapv3repository.ssl	.Values.rx.uidm.ldapv3repository.ssl.enabled	false
Указание правил для отражения групп пользователя в набор ролей UIDM	com.rooxteam.uidm.ldapv3repository.groups_to_roles_mapping	.Values.rx.uidm.ldapv3repository.groups_to_roles_mapping	

При сохранении в uidm можно настроить переименование групп находящихся в ldap.

```
com.rooxteam.uidm.ldapv3repository.groups_to_roles_mapping=rooxgroup1=rooxgroup1,rooxgroup_all=rooxgroup_all
# Формат: com.rooxteam.uidm.ldapv3repository.groups_to_org_roles_mapping.{groupName}=orgName1=role1;role2,orgName2=role3;role4
#com.rooxteam.uidm.ldapv3repository.groups_to_org_roles_mapping.GROUP_EMPLOYEE=b2b=adminAllOperations
```

Настройка LDAPS

1. Положить в хранилище сертификат сервера либо авторизационного центра, который выдал сертификат серверу.
   a. для версий, поставляемых как отдельные нативные сервисы, необходимо сделать через команду `keytool import …`
   b. для версий, поставляемых в helm-чартах
      i. `kubectl create secret generic roox-ca-certs --from-file=certfile=./ldap_certificate.cer`
      ii. при установке компонента roox-sso-server указать название созданного `.Values.rx.uidm.ldapv3repository.ssl.kubernetesSecretName: roox-ca-certs`
2. Указать настройку `com.rooxteam.uidm.ldapv3repository.ssl=true`
3. Указать порт `com.rooxteam.uidm.ldapv3repository.ldapPort=636`