SUPER_ADMIN RIGHTS ON ACTIVE DIRECTORY

super_admin rights on Active Directory

Active Directory Proxy User Super Admin

How To configure super_admin rights for Active Directory

There are two things to be considered in order to implement fine-grained LDAP permission for WebADM and its applications.

1. WebADM Proxy user permissions: This system user is used by WebADM to access and manipulate the required LDAP resources without an administrator login, for example, to increase the false authentication counter.
2. Administrator users permissions: These accounts login to the Admin portal in order to manage LDAP resources and registered applications.

These users are defined in `/opt/webadm/conf/webadm.conf` with `proxy_user` and `super_admins`. This documentation is fully dedicated to the super_admin rights. For the proxy_user rights, please have a look at the proxy_user documentation.

1. Global Rights (WebADM/OpenOTP/SpanKey/Self-Services)

When a WebADM administrator login on the WebADM Admin Portal, he always accesses and manages the LDAP resources under his own LDAP permissions. This means the user/group/configuration management permissions are enforced at the LDAP level. For example, a Windows AD Domain Administrator will be able to manage users and groups.

If the WebADM administrator is not an Active Directory administrator, we need to add permissions, depending on what the administrator is allowed to change in the user's attributes.

1.1 Mandatory Attributes used for an Extended Schema

- *webadmData*: is the attribute where the applications store the user data (ex. OpenOTP enrolled Token states).
- *webadmSettings*: is the attribute where WebADM stores user-specific settings (ex. per-user OTP policy).

In this example, we work with the domain *test.local* and the User Search Base configured in WebADM Domain is `CN=Users,DC=test,DC=local`:

```
PS C:\Users\administrator> (Get-ADRootDSE).rootDomainNamingContext
DC=test,DC=local
PS C:\Users\administrator> (Get-WmiObject Win32_NTDomain).DomainName
TEST
```

We set minimal rights easily with Powershell for all groups and users in *Users* container for the super_admin user:

```
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:WPRP;webadmData'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:WPRP;webadmSettings'
```

1.2 Mandatory Attributes used for a Not Extended Schema

- *bootfile*: is the attribute where the applications store the user data (ex. OpenOTP enrolled Token states).
- *bootparameter*: is the attribute where WebADM stores user-specific settings (ex. per-user OTP policy).

In this example, we work with the domain *test.local* and the User Search Base configured in WebADM Domain is `CN=Users,DC=test,DC=local`:

```
PS C:\Users\administrator> (Get-ADRootDSE).rootDomainNamingContext
DC=test,DC=local
PS C:\Users\administrator> (Get-WmiObject Win32_NTDomain).DomainName
TEST
```

We set minimal rights easily with Powershell for all groups and users in *Users* container for the super_admin user:

```
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:WPRP;bootfile'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:WPRP;bootparameter'
```

1.3 Optional Attributes

In this example, we work with the domain *test.local*, `webadm_admins` is the super_admin and the User Search Base configured in WebADM Domain is `CN=Users,DC=test,DC=local`.

For example, if you want that the super_admin user is able to reset users LDAP password through the WebADM Admin GUI, change mobile numbers or email addresses on users account, then the super_admin will need to have write permissions to the corresponding LDAP attributes. The following ones can be configured:

- *mail* (only if Self-Services are used to set email addresses)
- *mobile* (only if Self-Services are used to set mobile numbers)
- *preferredLanguage* (only if Self-Services are used to set user language)
- *userPassword* or *unicodePwd* and *pwdlastset* for Windows AD (only if Self-Services are used to set user password)
- *lockouttime* is used to unlock an AD account at the AD level through WebADM admin GUI or PWReset application.

```
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:WPRP;mail'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:WPRP;mobile'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:WPRP;preferredLanguage'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:WPRP;userPassword'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:WPRP;pwdlastset'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:WPRP;unicodepwd'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:WPRP;lockouttime'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:CA;Reset Password'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:WPRP;userCerticate'
```

These attributes can now be edited by the super_admin `webadm_admins`. Many other attributes can be edited through the WebADM Admin GUI, so if you want to allow your super_admin user to edit other attributes then, give the rights on the attributes you want in the same way.

1.3.1 Voice attribute

The voice authentication has been introduced in version 2. The following attributes are used to store the data of voice enrollment:

- *webadmVoice* attribute when the Schema is extended. This permission needs to be set if voice is activated:

```
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:WPRP;webadmVoice'
```

- *audio* attribute when the Schema is not extended. The following permission needs to be set for Schema not extended if voice is activated:

```
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:WPRP;audio'
```

1.4 Read right on the full user search base

The super_admin needs to read user objects and user attributes. This can be done with the following permission:

```
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:RP'
```

1.5 Users activation

To activate a user through WebADM, you super_admin logged on the WebADM Admin GUI must have the following right:

```
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:WPRP;objectClass'
```

The activation consists of adding a class to the user account. For a schema extended, the class added during the activation is webadmaccount object class. For a not extended schema, the class added during the activation is bootableDevice object class.

1.6 Attribute for Spankey

For Spankey usage, the super_admin needs read/write permissions on the following attributes :

- *uidnumber*: Mandatory to use an AD/LDAP account on UNIX systems,
- *gidnumber*: Mandatory to use an AD/LDAP account on UNIX systems,
- *unixhomedirectory*: Home directory location, mandatory.
- *loginshell*: Login shell location, mandatory.

```
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:WPRP;uidnumber'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:WPRP;gidnumber'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:WPRP;unixhomedirectory'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\webadm_admins:WPRP;loginshell'
```

To extend an account to UNIX (mandatory if you want to use Spankey), the right defined in the 1.5 step is enough. Extend an account to UNIX consist by adding the posixaccount object class to the user account.

2. super_admin permissions on Domain Administrators (AdminSDHolder)

For writing on AD administrators, rights previously settled are not enough because *AdminSDHolder* overwrites these rights every hour. So we need also to apply these rules on *AdminSDHolder* object and wait one hour that it's applied on all admin users and groups of the domain. **These rights must be applied ONLY if the super_admin logged on the WebADM Admin GUI is not a Domain Admin User and you want to perform changes/operations on Domain Admins accounts with RCDevs solutions**.

2.1. Schema Extended

In this example, we work with the domain *test.local* and `webadm_admins` is the super_admin:

```
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /G
'TEST\webadm_admins:WPRP;webadmData'
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /G
'TEST\webadm_admins:WPRP;webadmSettings'
```

2.2 Schema Not Extended

In this example, we work with the domain *test.local* and `webadm_admins` is the super_admin:

```
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /G
'TEST\webadm_admins:WPRP;bootFile'
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /G
'TEST\webadm_admins:WPRP;bootParameter'
```

2.3 Domain Admin User activation

If the super_admin logged on the WebADM Admin GUI is a domain user and you want to Activate a Domain Admin user:

```
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /G
'TEST\webadm_admins:WPRP;objectClass'
```

2.3 super_admin Optional rights

If the super_admin logged on the WebADM Admin GUI is a domain user and you want to edit the following attributes on a Domain Admin user:

```
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /G 'TEST\webadm_admins:WPRP;mail'
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /G
'TEST\webadm_admins:WPRP;mobile'
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /G
'TEST\webadm_admins:WP;preferredLanguage'
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /G
'TEST\webadm_admins:WPRP;userPassword'
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /G 'TEST\webadm_admins:CA;Reset
Password'
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /G
'TEST\webadm_admins:WPRP;pwdlastset'
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /G
'TEST\webadm_admins:WPRP;lockouttime'
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /G
'TEST\webadm_admins:WPRP;userCertificate'
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /G
'TEST\webadm_admins:WPRP;uidnumber'
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /G
'TEST\webadm_admins:WPRP;gidnumber'
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /G
'TEST\webadm_admins:WPRP;unixhomedirectory'
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /G
'TEST\webadm_admins:WPRP;loginshell'
```

In this example, all WebADM configuration containers are under *CN=webadm,DC=test,DC=local*, we add full access to all descendants objects of this container.

2.3.1 Voice attribute

The voice authentication has been introduced in version 2. The following attributes are used to store the data of voice enrollment:

- *webadmVoice* attribute when the Schema is extended. This permission needs to be set:

```
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /G
'TEST\webadm_admins:WPRP;webadmVoice'
```

- *audio* attribute when the Schema is not extended.

The following permission needs to be set for Schema not extended:

```
dsacls "CN=AdminSDHolder,CN=System,DC=test,DC=local" /G 'TEST\webadm_admins:WPRP;audio'
```

3. Rights on the WebADM Container

In this example, we work with the domain *test.local*, `webadm_admins` is the super_admin and `CN=webadm,DC=test,DC=local` is our container defined in `/opt/webadm/conf/webadm.conf`. A super_admin is able to manage application settings (e.g: OpenOTP settings) through the WebADM Admin GUI. Graphical configurations are stored in your LDAP server under the container dedicated to WebADM (container defined in /opt/webadm/conf/webadm.conf). The super_admin must have the grant access on this container/Organizational Unit and all descendent objects.

```
dsacls "CN=webadm,DC=test,DC=local" /I:T /G 'TEST\webadm_admins:GA'
```