PROXY USER PERMISSIONS ON AD

No guarantee is given for the correctness of the information contained in this document. Please send any comments or corrections to info@rcdevs.com.

Proxy User Permissions on AD

Active Directory Proxy User

How to configure proxy_user 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, register token metadata on the user account…
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` settings. This documentation is fully dedicated to the proxy_user rights. For the super_admin rights, please have a look at the super_admin documentation.

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

The proxy user needs to perform a wide LDAP search and reads. It also requires read-only permissions to the WebADM LDAP configurations (i.e. configured containers) and to the user Domains subtrees. The proxy user needs to do some write operations to a few LDAP attributes because it needs to store dynamic application user data into the users.

In some circumstances, the Proxy user will also need to write an application setting on the users and groups. The following attributes are part of the WebADM LDAP schema and need Proxy user read/write permissions:

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 proxy_user user:

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

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

1.3 Optional Attributes (used by WebApps)

If you use WebADM Self-Services and depending on what you allow the users to do within the Self-Service applications, then WebADM proxy_user may need some additional permissions:

For example, if you want users to reset their LDAP password, set their mobile numbers or email addresses, then the Proxy user 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* 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.
- *useraccountcontrol* is used to change the AD accounts flags (Disabled account, Normal account, User can not change password…)
- *userCertificate* is used when the user want to create and register a new user certificate on his account from Web applications.
- *member* and *memberuid* are used when you are using the badging functionality and after a success badging from a user, you

automatically assign it to a group. That permissions must be set at the level of group location (*memberuid* is only required if you implement SpanKey SSH).

```
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\proxy_user:WPRP;mail'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\proxy_user:WPRP;mobile'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\proxy_user:WPRP;preferredLanguage'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\proxy_user:WPRP;userPassword'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\proxy_user:WPRP;unicodepwd'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\proxy_user:WPRP;lockouttime'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\proxy_user:WPRP;useraccountcontrol'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\proxy_user:WPRP;userCertificate'
dsacls "CN=Users,DC=test,DC=local" /I:S /G 'TEST\proxy_user:CA;Reset Password'
dsacls "CN=Groups,DC=test,DC=local" /I:S /G 'TEST\proxy_user:WPRP;member'
dsacls "CN=Groups,DC=test,DC=local" /I:S /G 'TEST\proxy_user:WPRP;memberuid'
```

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\proxy_user: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\proxy_user:WPRP;audio'
```

1.4 Read right on the full user search base

The proxy_user 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\proxy_user:RP'
```

1.5 proxy_user right for Domain Users Activation (Optional)

Usually, the super_admin logged on the WebADM Admin GUI will extend user accounts. If you plan to use the command line tool `/opt/webadm/bin/extend` to activate your users, then the proxy_user will be used for the user extension. To allow the proxy_user to extend your users accounts, you have to set the following right:

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

1.6 Extra rights for SpanKey

The rights defined in steps 1.1, 1.2 and 1.4 are enough for SpanKey. If the delegation has been done properly, then your proxy_user should have the following rights:

1.6.1 Schema extended (Already done in step 1.1)

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

1.6.2 Schema not extended (Already done in step 1.2)

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

1.6.3 Global attributes (Already done in step 1.4)

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

1.7 Extra Rights for Badge-in/Check-in operations

With the RCDevs badging feature, you can group users based on badged-in/check-in operations. The proxy_user is involved in these operations, which is why you need to grant the following permissions to enable the aforementioned operations:

```
dsacls "CN=groups,DC=test,DC=local" /I:S /G 'TEST\proxy_user:RPWP;member'
```

That permission must be applied on groups OUs.

If the group is posix extended, WebADM will also try to modify memberUid attribut of the group. In that case, you also need to grant the following permission:

```
dsacls "CN=groups,DC=test,DC=local" /I:S /G 'TEST\proxy_user:RPWP;memberUid'
```

2. proxy_user permissions for 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 you want to perform OpenOTP logins, Spankey logins or use self-service application with your Domain Admins accounts:

2.1 Extended Schema

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

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

2.2 Not Extended Schema

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

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

2.3 Optional rights

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

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

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 if voice is activated:

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

2.4 proxy_user rights for Domain Admin Activation (Optional)

Usually, the super_admin logged on the WebADM Admin GUI will extend user accounts. If you plan to use the command line tool `/opt/webadm/bin/extend` to activate your domain admin users, then the proxy_user will be used for the user extension. To allow the proxy_user to extend your domain admin users, you have to set the following right:

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

3. Rights on the WebADM Container

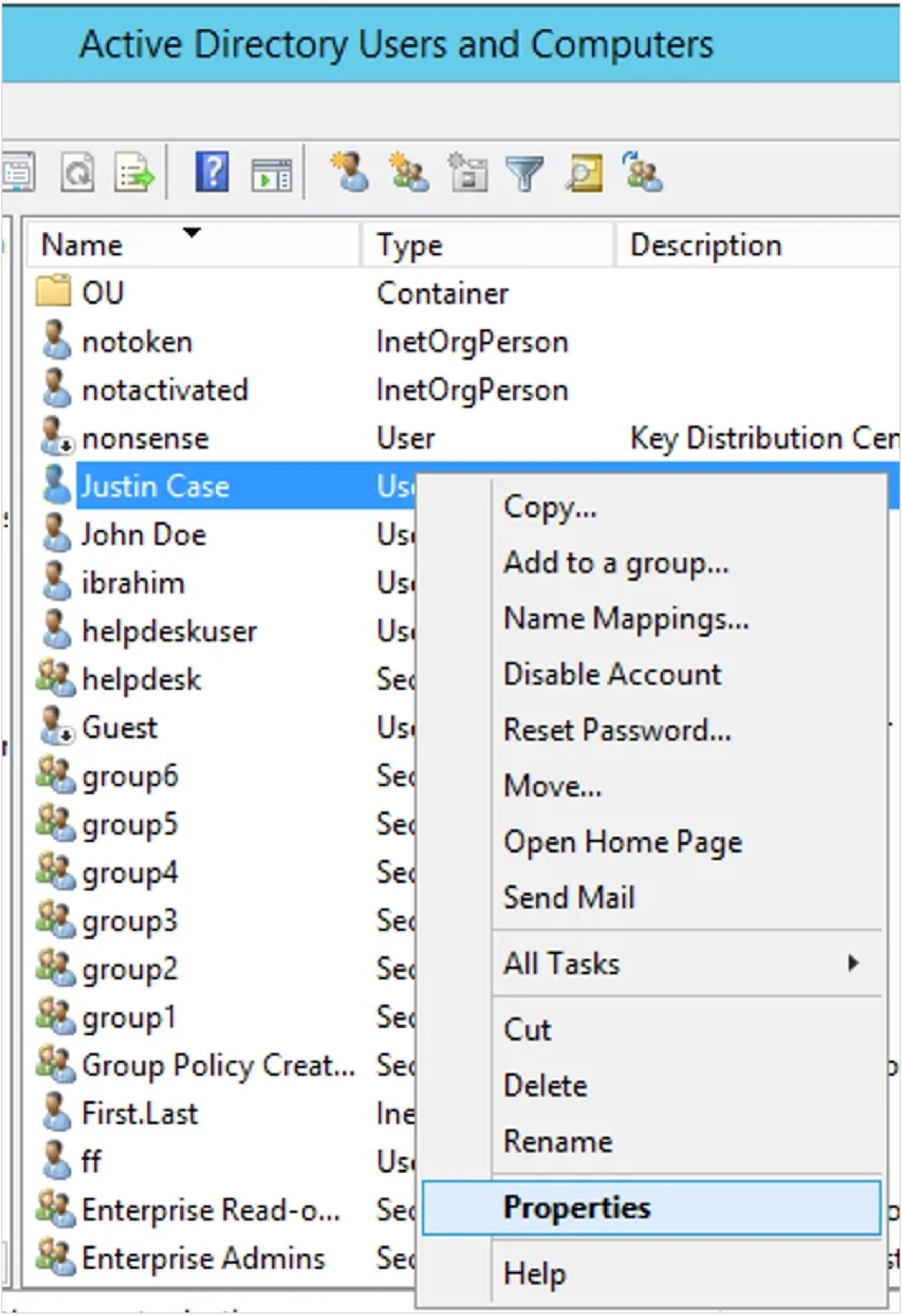
In this example, we work with the domain *test.local*, `proxy_user` is our proxy user and `CN=webadm,DC=test,DC=local` is our container defined in `/opt/webadm/conf/webadm.conf`:

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

4. Viewing effective access

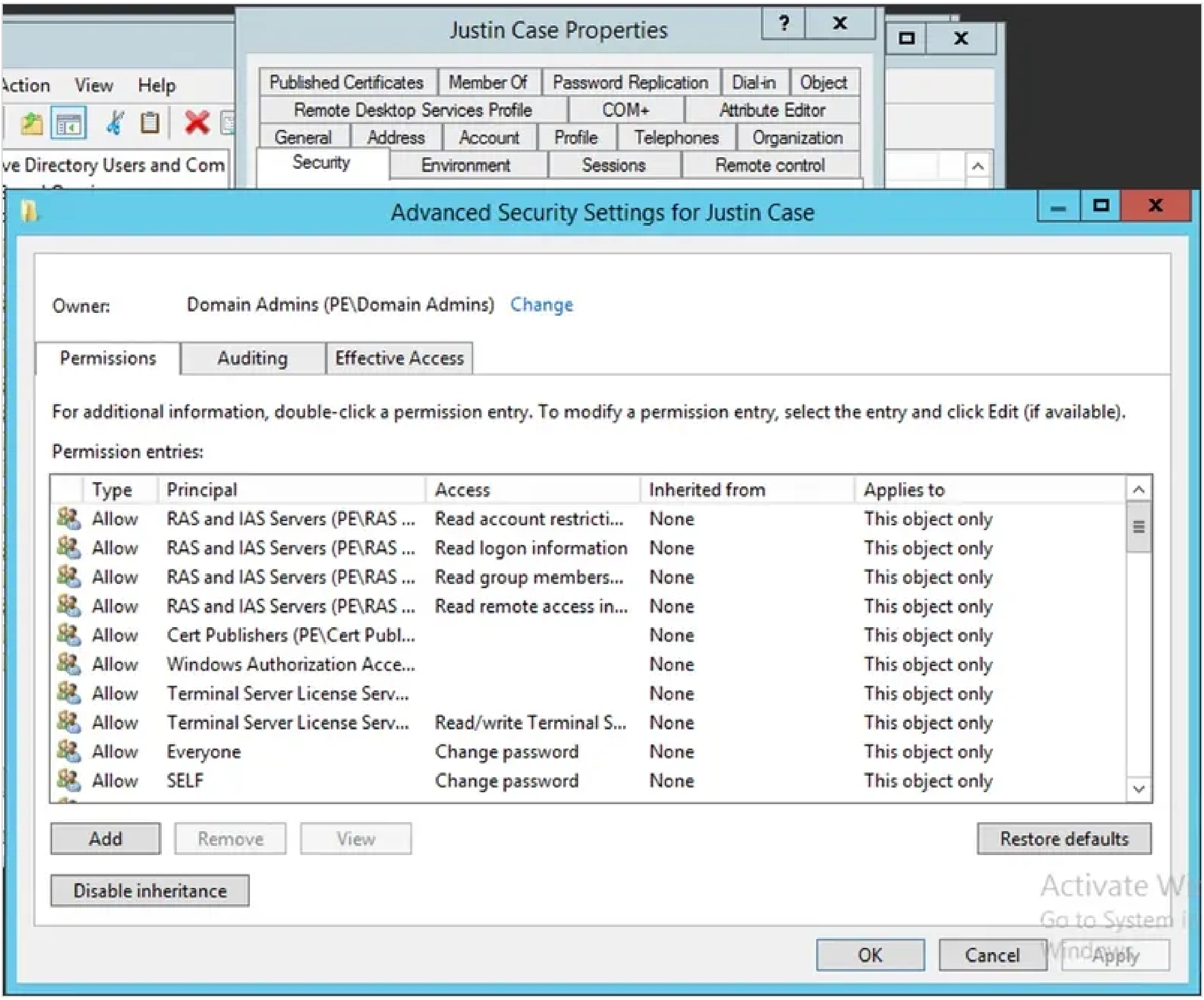
In case you are not sure the permissions are set correctly on a specific user account, you can view the effective access a user has to another account.

First open Active Directory Users and Computers and find the user which has a problem logging in.



Name	Type	Description
OU	Container	
notoken	InetOrgPerson	
notactivated	InetOrgPerson	
nonsense	User	Key Distribution Cen
Justin Case	Us	
John Doe	Us	
ibrahim	Us	
helpdeskuser	Us	
helpdesk	Se	
Guest	Us	
group6	Se	
group5	Se	
group4	Se	
group3	Se	
group2	Se	
group1	Se	
Group Policy Creat...	Se	
First.Last	Ine	
ff	Us	
Enterprise Read-o...	Se	
Enterprise Admins	Se	

Open the “Properties > Security > Advanced Security Settings” page for the user.



Owner: Domain Admins (PE\Domain Admins) Change

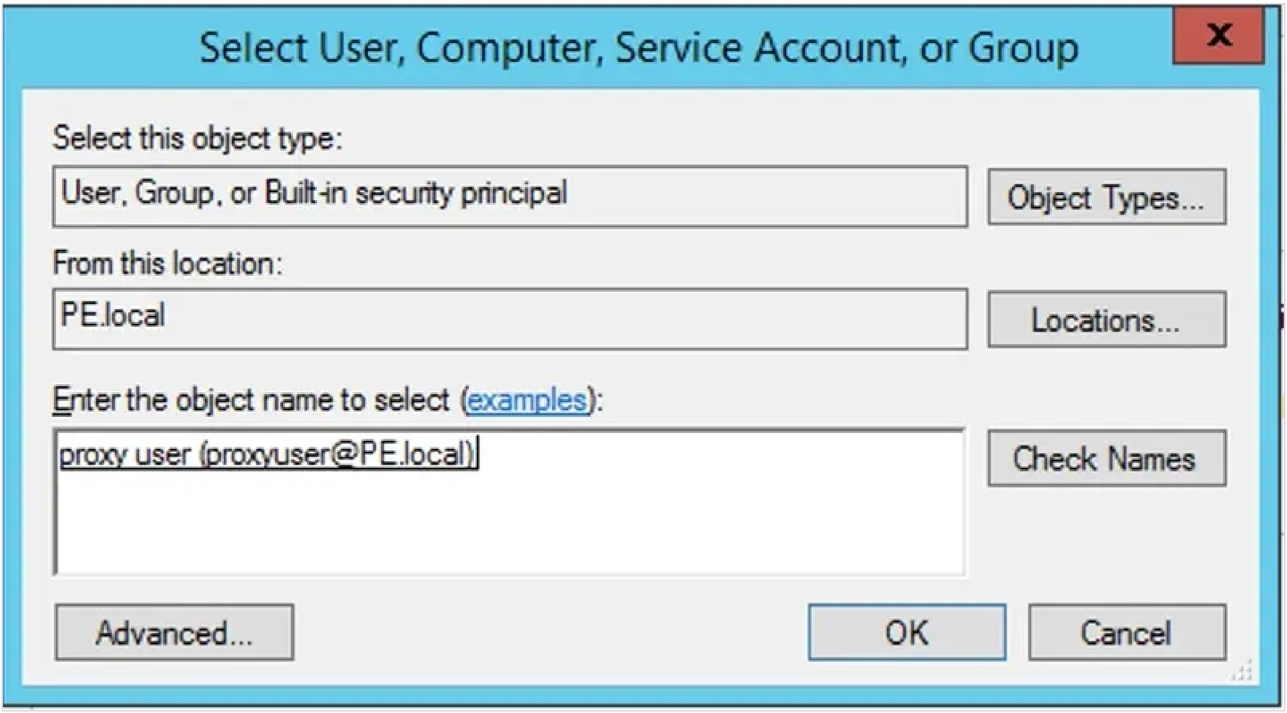
For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

Permission entries:

Type	Principal	Access	Inherited from	Applies to
Allow	RAS and IAS Servers (PE\RAS ...	Read account restricti...	None	This object only
Allow	RAS and IAS Servers (PE\RAS ...	Read logon information	None	This object only
Allow	RAS and IAS Servers (PE\RAS ...	Read group members...	None	This object only
Allow	RAS and IAS Servers (PE\RAS ...	Read remote access in...	None	This object only
Allow	Cert Publishers (PE\Cert Publ...		None	This object only
Allow	Windows Authorization Acce...		None	This object only
Allow	Terminal Server License Serv...		None	This object only
Allow	Terminal Server License Serv...	Read/write Terminal S...	None	This object only
Allow	Everyone	Change password	None	This object only
Allow	SELF	Change password	None	This object only

First check permission inheritance. If you have “Enable inheritance” button, that means the AD object is NOT inheriting permissions from the parent object. This could be for example because the user has previously been in Domain Admin group.

Next, open the effective access page and select the ‘proxy_user’ you have configured in webadm.conf



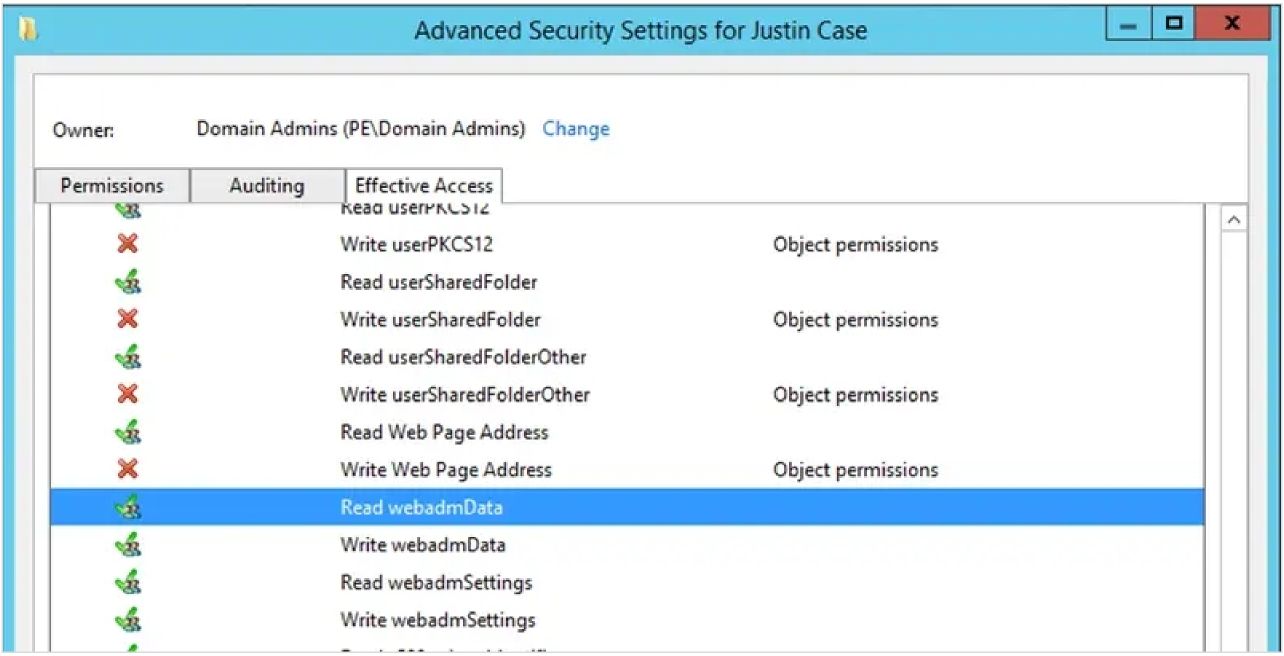
Select this object type:
User, Group, or Built-in security principal

From this location:
PE.local

Enter the object name to select (examples):
proxy user (proxyuser@PE.local)

Then clieck the “View Effective Access” button. This will show you the detailed permissions that the proxy user has to user in question.

In schema extended case you should see read and write access to the *webadmData* and *webadmSettings* attributes.



Advanced Security Settings for Justin Case
Owner: Domain Admins (PE\Domain Admins) Change
Permissions
Auditing
Effective Access

✓	Read userPKCS12	
✗	Write userPKCS12	Object permissions
✓	Read userSharedFolder	
✗	Write userSharedFolder	Object permissions
✓	Read userSharedFolderOther	
✗	Write userSharedFolderOther	Object permissions
✓	Read Web Page Address	
✗	Write Web Page Address	Object permissions
✓	Read webadmData	
✓	Write webadmData	
✓	Read webadmSettings	
✓	Write webadmSettings	

If you have not extended the schem, the corresponding attributes are *bootFile* and *bootParameter*

This manual was prepared with great care. However, RCDevs Security S.A. and the author cannot assume any legal or other liability for possible errors and their consequences. No responsibility is taken for the details contained in this manual. Subject to alternation without notice. RCDevs Security S.A. does not enter into any responsibility in this respect. The hardware and software described in this manual is provided on the basis of a license agreement. This manual is protected by copyright law. RCDevs Security S.A. reserves all rights, especially for translation into foreign languages. No part of this manual may be reproduced in any way (photocopies, microfilm or other methods) or transformed into machine-readable language without the prior written permission of RCDevs Security S.A. The latter especially applies for data processing systems. RCDevs Security S.A. also reserves all communication rights (lectures, radio and television). The hardware and software names mentioned in this manual are most often the registered trademarks of the respective manufacturers and as such are subject to the statutory regulations. Product and brand names are the property of RCDevs Security. © 2024 RCDevs Security S.A., All Rights Reserved