UTILITIES AND COMMAND LINE TOOLS FOR WEBADM

📄 Utilities and Command Line Tools for WebADM

script

1. Introduction

In this How-To, we will demonstrate some useful scripts available for WebADM and how to use them.

2. WebADM Utilities and Scripts

Some scripts are available in:

```
[root@webadm]# cd /opt/webadm/bin/
[root@webadm bin]# ll
total 152
-rwxr-xr-x 1 root root  1809 11 oct.  15:35 backup
-rwxr-xr-x 1 root root  6807 11 oct.  15:35 dbprune
-rwxr-xr-x 1 root root 11215 11 oct.  15:35 encrypt
-rwxr-xr-x 1 root root 10837 11 oct.  15:35 extend
-rwxr-xr-x 1 root root  4442 11 oct.  15:35 pwcrypt
-rwxr-xr-x 1 root root   837 11 oct.  15:35 restore
-rwxr-xr-x 1 root root 17177 11 oct.  15:35 setup
-rwxr-xr-x 1 root root 12626 11 oct.  15:35 verify
-rwxr-xr-x 1 root root 12186 11 oct.  15:35 webadm
-rwxr-xr-x 1 root root 56456 11 oct.  15:35 yubihsm
```

And some other scripts are available in:

```
[root@webadm]# cd /opt/webadm/doc/scripts
[root@webadm]# ll
total 20
-rwxr-xr-x 1 root root  787 11 oct.  19:46 backup_all
-rwxr-xr-x 1 root root 4633 11 oct.  19:46 create_cert
-rwxr-xr-x 1 root root  379 11 oct.  19:46 create_mysqldb
-rwxr-xr-x 1 root root 1655 11 oct.  19:46 repair_perms
```

3. backup

This script allows you to perform a backup of the WebADM configuration files.

```
[root@webadm bin]# ./backup
Usage: ./backup [-f] <BACKUP_FILE>
```

```
[root@webadm bin]# ./backup -f mybackup
Adding conf/webadm.conf... Ok
Adding conf/rsignd.conf... Ok
Adding conf/objects.xml... Ok
Adding conf/servers.xml... Ok
Adding pki... Ok
Adding logs... Ok
Adding temp... Ok
Adding conf/license.key... Ok
Adding conf/webadm.env... Ok
Adding conf/custom.crt... Ok
Adding conf/custom.key... Ok
Adding websrvs/smshub/smsc/custom1.php... Ok
Adding websrvs/smshub/smsc/custom2.php... Ok
Adding websrvs/smshub/smsc/custom3.php... Ok
Adding websrvs/smshub/smsc/custom4.php... Ok
Compressing backup file... Ok

WebADM backup created in mybackup.gz
```

My backup file is created, I can use the restore script to restore my WebADM configuration.

4. restore

Once I have created my backup archive, I can use the restore script to restore my WebADM configuration.

```
[root@webadm bin]# ./restore
Usage: ./restore <BACKUP_FILE>
```

```
[root@webadm tmp]# /opt/webadm/bin/restore mybackup.gz
Are you sure you want to restore WebADM (y/n)? y
Unpacking backup files... Ok
Checking system architecture...Ok
Setting file permissions... Ok
Adding system user to dialout group... Ok
Adding systemd service... Ok
Adding logrotate scripts... Ok
WebADM has successfully been setup.

WebADM backup restored from mybackup.gz
```

5. dbprune

This script allows you to remove DB logs entries older than the specified time passed as a parameter.

```
[root@webadm bin]# ./dbprune
Usage: dbprune [-y] [-t <prune time>] TABLE1 [TABLE2] ... [TABLEX]
Table 1..X is a list of WebADM SQL tables within Admin, Manag, Webapp, Websrv, Alert
and Record.
Options:
  -y : Do not prompt for validation (to be used with cron jobs).
  -t <prune time> : Max age of SQL data retention in the form n[dmy] (ex. 1m for one
month).
                    If no prune time is defined, the default pruning time is 3 month.
```

In my example, I remove DB logs entries older than one day:

```
[root@webadm bin]# ./dbprune -t 1d -y Websrv
Pruning log event older than 1 day in WebSrv SQL table... Ok
```

6. encrypt

This script allows you to encrypt, decrypt, re-encrypt user metadata (WebADM metadata stored on users accounts in the LDAP).

```
[root@webadm bin]# ./encrypt
Usage: encrypt [-e | -d | -r] [-y] [-t] [-f <search filter>] DOMAIN1 [DOMAIN2] ...
[DOMAINX]
Domain 1..X is a list of WebADM Domains where LDAP users will be updated.
Options:
  -e : Encrypt user data
  -d : Decrypt user data (not available with HSM encryption)
  -r : Recrypt user data
  -y : Do not prompt for validation (to be used with cron jobs).
  -t : Only list users to be updated but do not update (testing mode).
  -f <search filter> : Use a custom LDAP search filter.
  -c <search base> : Use a custom LDAP search base (can be used without domain list).
If no search filter is specified then any user object found in the domain(s) and which
includes the webadmAccount LDAP extension will be updated.
```

```
[root@webadm bin]# ./encrypt -d -f \(objectclass=webadmaccount\) rcdevs
This script will help you manage the WebADM user data encryption for the
LDAP users in the provided WebADM Domain(s). Using the script you can:
1) Review user data encryption.
2) Decrypt user data (-d option - not available with HSM encryption).
3) Encrypt user data (-e option).
4) Recrypt user data (-r option).
WebADM always uses the first configured encrypt_key to encrypt user data.
If you want to change the default encrypt_key then set the new key first.

Are you sure you want to update user data (y/n)? y

Entering Domain rcdevs (dc=rcdevs,dc=com).
Decrypting user data for CN=Administrateur,CN=Users,DC=rcdevs,DC=com... Ok
No user data found for CN=YO_AD-DC,OU=Domain Controllers,DC=rcdevs,DC=com (ignoring)
No user data found for CN=krbtgt,CN=Users,DC=rcdevs,DC=com (ignoring)
No user data found for CN=YO_DC,OU=Domain Controllers,DC=rcdevs,DC=com (ignoring)
No user data found for CN=Web ADM,OU=Services,DC=rcdevs,DC=com (ignoring)
No user data found for CN=push user,OU=Utils,DC=rcdevs,DC=com (ignoring)
No user data found for CN=scope,OU=TestScope,DC=rcdevs,DC=com (ignoring)
No user data found for CN=scope2,OU=testscope2,OU=TestScope,DC=rcdevs,DC=com (ignoring)
No user data found for CN=proxyuser,CN=Users,DC=rcdevs,DC=com (ignoring)
No user data found for CN=adfs user,CN=Users,DC=rcdevs,DC=com (ignoring)
Decrypting user data for CN=vagrant,CN=Users,DC=rcdevs,DC=com... Ok
Decrypting user data for CN=yoann,CN=Users,DC=rcdevs,DC=com... Ok
No user data found for CN=test@test,CN=Users,DC=rcdevs,DC=com (ignoring)
No user data found for CN=ff,OU=ADFS,DC=rcdevs,DC=com (ignoring)
Decrypting user data for CN=webmaster,CN=Users,DC=rcdevs,DC=com... Ok
No user data found for CN=OracleGuest,CN=Users,DC=rcdevs,DC=com (ignoring)
Decrypting user data for CN=perttu,CN=Users,DC=rcdevs,DC=com... Ok
Decrypting user data for CN=Admin,OU=localuser,DC=rcdevs,DC=com... Ok
Decrypting user data for CN=testadfs,CN=Users,DC=rcdevs,DC=com... Ok
Decrypting user data for CN=user,CN=Users,DC=rcdevs,DC=com... Ok
Decrypting user data for CN=administrator,OU=localuser,DC=rcdevs,DC=com... Ok

Updated 9 LDAP users in 0 seconds (0 errors).
```

Users data are now decrypted. We will try now to encrypt it again:

```
[root@webadm bin]# ./encrypt -r -f \(objectclass=webadmaccount\) rcdevs
This script will help you manage the WebADM user data encryption for the
LDAP users in the provided WebADM Domain(s). Using the script you can:
1) Review user data encryption.
2) Decrypt user data (-d option - not available with HSM encryption).
3) Encrypt user data (-e option).
4) Recrypt user data (-r option).
WebADM always uses the first configured encrypt_key to encrypt user data.
If you want to change the default encrypt_key then set the new key first.

Are you sure you want to update user data (y/n)? y
Entering Domain rcdevs (dc=rcdevs,dc=com).
Re-encrypting user data for CN=Administrateur,CN=Users,DC=rcdevs,DC=com... Ok
No user data found for CN=YO_AD-DC,OU=Domain Controllers,DC=rcdevs,DC=com (ignoring)
No user data found for CN=krbtgt,CN=Users,DC=rcdevs,DC=com (ignoring)
No user data found for CN=YO_DC,OU=Domain Controllers,DC=rcdevs,DC=com (ignoring)
No user data found for CN=Web ADM,OU=Services,DC=rcdevs,DC=com (ignoring)
No user data found for CN=push user,OU=Utils,DC=rcdevs,DC=com (ignoring)
No user data found for CN=scope,OU=TestScope,DC=rcdevs,DC=com (ignoring)
No user data found for CN=scope2,OU=testscope2,OU=TestScope,DC=rcdevs,DC=com (ignoring)
No user data found for CN=proxyuser,CN=Users,DC=rcdevs,DC=com (ignoring)
No user data found for CN=adfs user,CN=Users,DC=rcdevs,DC=com (ignoring)
Re-encrypting user data for CN=vagrant,CN=Users,DC=rcdevs,DC=com... Ok
Re-encrypting user data for CN=yoann,CN=Users,DC=rcdevs,DC=com... Ok
No user data found for CN=test@test,CN=Users,DC=rcdevs,DC=com (ignoring)
No user data found for CN=ff,OU=ADFS,DC=rcdevs,DC=com (ignoring)
Re-encrypting user data for CN=webmaster,CN=Users,DC=rcdevs,DC=com... Ok
No user data found for CN=OracleGuest,CN=Users,DC=rcdevs,DC=com (ignoring)
Re-encrypting user data for CN=perttu,CN=Users,DC=rcdevs,DC=com... Ok
Re-encrypting user data for CN=Admin,OU=localuser,DC=rcdevs,DC=com... Ok
Re-encrypting user data for CN=testadfs,CN=Users,DC=rcdevs,DC=com... Ok
Re-encrypting user data for CN=user,CN=Users,DC=rcdevs,DC=com... Ok
Re-encrypting user data for CN=administrator,OU=localuser,DC=rcdevs,DC=com... Ok

Updated 9 LDAP users in 0 seconds (0 errors).
```

Data are now encrypted.

7. extend

The user activation/extension can be done through this script. The usage is pretty the same as the encrypt script:

```
[root@webadm bin]# ./extend
Usage: extend [-y] [-t] [-f <search filter>] DOMAIN1 [DOMAIN2] ... [DOMAINX]
Domain 1..X is a list of WebADM Domains where LDAP users will be extended.
Options:
  -y : Do not prompt for validation (to be used with cron jobs).
  -t : Only list users to be extended but do not extend (testing mode).
  -u : Un-extend users (remove the webadmAccount objectclass and attributes).
  -f <search filter> : Use a custom LDAP search filter.
  -c <search base> : Use a custom LDAP search base (can be used without domain list).
  -g <group DN> : Extend only users which are members of the group.
If no search filter is defined, all user objects without the webadmAccount
extension will be extended.
```

This time I use the option -t to list which accounts will be extended if I perform this request.

```
[root@webadm bin]# ./extend -t -f \(objectclass=user\) rcdevs
This script will activate all your LDAP users in the provided Domain(s)
by automatically adding the webadmAccount LDAP objectclass.
Once extended, a user is activated for use with WebADM applications
and it is accounted in the WebADM lincensing and user limit checks.

Are you sure you want to extend users (y/n)? y
Entering Domain rcdevs (cn=users,dc=rcdevs,dc=com).
User CN=yoann,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=test@test,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=Administrateur,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=Invité,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=webmaster,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=OracleGuest,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=perttu,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=test1,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=krbtgt,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=testadfs,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=user,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=proxyuser,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=adfs user,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=vagrant,CN=Users,DC=rcdevs,DC=com will be extended.

Found 14 LDAP users to be extended in 0 seconds.
```

Same way to remove the WebADM account extension, I add the -u option:

```
[root@webadm bin]# ./extend -t -u -f \(objectclass=webadmAccount\) rcdevs
This script will activate all your LDAP users in the provided Domain(s)
by automatically adding the webadmAccount LDAP objectclass.
Once extended, a user is activated for use with WebADM applications
and it is accounted in the WebADM lincensing and user limit checks.

Are you sure you want to extend users (y/n)? y
Entering Domain rcdevs (cn=users,dc=rcdevs,dc=com).
User CN=yoann,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=test@test,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=Administrateur,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=webmaster,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=OracleGuest,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=perttu,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=krbtgt,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=testadfs,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=user,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=proxyuser,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=adfs user,CN=Users,DC=rcdevs,DC=com will be extended.
User CN=vagrant,CN=Users,DC=rcdevs,DC=com will be extended.

Found 12 LDAP users to be extended in 0 seconds.
```

8. pwcrypt

This script allows encrypting some sensitive WebADM configuration settings like user passwords and encryption keys. You can also replace the cleartext passwords and keys with encrypted values in webadm.conf and servers.xml.

> ⚠ !!! Freeware Users !!!
>
> Since WebADM v2.x, freeware users can use pwcrypt tool to encrypt passwords in config files. When you generate a freeware license from `https://cloud.rcdevs.com/freeware-license/`, an email address is asked to provide the license. The email address provided for the license is part of the encryption with pwcrypt tool. After one year, freeware license is expired and you have to renew your license by generating a new one. If you do not use the same email address as for the first license, all passwords encrypted with the first license will not be able to be decrypted with the new license. If you use that tool with freeware version, take care to always use the same email address to generate your freeware license.

```
[root@webadm scripts]#./pwcrypt
Usage: pwcrypt [-p] SECRET
Options:
   -p : Secret is prompted (secret argument is omitted)
Argument is the secret value to be encrypted for use in your config files.
The maximum password or secret length is 250 characters.
```

```
[root@webadm bin]# ./pwcrypt -p
This script allows to encrypt some sensitive WebADM configuration settings
like user passwords and encryption keys. You can also replace the cleartext
passwords and keys with encrypted values in webadm.conf and servers.xml.

Secret: ********
Encrypted: {wcrypt}9k7vvgQQk/XoyNyPISPBSw==
```

For example:

```
-bash-4.2#/opt/webadm/bin/pwcrypt -p
This script allows to encrypt some sensitive WebADM configuration settings
like user passwords and encryption keys. You can also replace the cleartext
passwords and keys with encrypted values in webadm.conf and servers.xml.

Secret: ****************
Encrypted: {wcrypt}Ucw4WJir9VGFzeKoTdYkOWAkO/kXIHSlkl655RyGHJc=

-bash-4.2# vi /opt/webadm/conf/rsignd.conf
#
# WebADM PKI Server Configuration
#
...
#
# Client sections
#
# Declare here the Rsign clients with IP addresses or hostnames.
# In cluster mode, the client WebADM server(s) must be defined here!

client {
 hostname localhost
 secret {wcrypt}Ucw4WJir9VGFzeKoTdYkOWAkO/kXIHSlkl655RyGHJc=
}

-bash-4.2#  vi /opt/webadm/conf/servers.xml
<?xml version="1.0" encoding="UTF-8" ?>

<Servers>

<!--
*****************************************
***  WebADM Remote Server Connections  ***
*****************************************

...

<!--
A PKI server (or CA) is required for signing user certificates.
The RSign PKI server is included in WebADM. So you can keep the
default settings here.
-->

<PkiServer name="PKI Server"
 host="192.168.3.80"
 port="5000"
 secret="{wcrypt}Ucw4WJir9VGFzeKoTdYkOWAkO/kXIHSlkl655RyGHJc="
 ca_file="" />

-bash-4.2# /opt/webadm/bin/webadm restart
```

9. setup

This script is used for the first setup of WebADM and can be re-used at any moment if it's required.

```
[root@webadm bin]# ./setup
```

Just follow the setup script. WebADM services must be stopped before running the setup script.

10. verify

This script does some verifications such as if users metadata are encrypted, if passwords or certificates are expired…

```
[root@webadm bin]# ./verify
Usage: verify [-y] [-f <search filter>] DOMAIN1 [DOMAIN2] ... [DOMAINX]
Domain 1..X is a list of WebADM Domains where LDAP objects will be verified.
Options:
  -y : Do not prompt for validation (to be used with cron jobs).
  -f <search filter> : Use a custom LDAP search filter.
  -c <search base> : Use a custom LDAP search base (can be used without domain list).
If no search filter is defined, all user and group objects will be verified.
```

Here, we do verification for domain rcdevs:

```
[root@webadm bin]# ./verify rcdevs
This script will verify the consistency of your LDAP users and groups in the
provided Domain(s). WebADM LDAP attributes and certificates will be analysed.

Are you sure you want to verify objects (y/n)? y
Entering Domain rcdevs (cn=users,dc=rcdevs,dc=com).
Checking WebADM account CN=yoann,CN=Users,DC=rcdevs,DC=com...
 > WebADM Settings are valid: Yes
 > WebADM Data are valid: Yes (encrypted)
 > Password expiration: Expires in 190 days!
Checking WebADM account CN=test@test,CN=Users,DC=rcdevs,DC=com...
 > Password expiration: Expires in 21 days!
Checking WebADM account CN=Administrateur,CN=Users,DC=rcdevs,DC=com...
 > WebADM Data are valid: Yes (encrypted)
 > Login certificates (1):
   -
/description=USER/CN=rcdevs\administrateur/UID=administrateur/DC=rcdevs/emailAddress=supp
Valid (expires in 208 days)
 > Password expiration: Never expires
Checking user CN=Invité,CN=Users,DC=rcdevs,DC=com...
 > Password expiration: Never expires
Checking WebADM group CN=WinRMRemoteWMIUsers__,CN=Users,DC=rcdevs,DC=com...
```

```
 > Contains members: No
Checking WebADM group CN=master,CN=Users,DC=rcdevs,DC=com...
 > Contains members: Yes (1 members)
Checking WebADM account CN=webmaster,CN=Users,DC=rcdevs,DC=com...
 > WebADM Settings are valid: Yes
 > WebADM Data are valid: Yes (encrypted)
 > Password expiration: Expires in 88 days!
Checking WebADM account CN=OracleGuest,CN=Users,DC=rcdevs,DC=com...
 > Password expiration: Expires in 60 days!
Checking WebADM account CN=perttu,CN=Users,DC=rcdevs,DC=com...
 > WebADM Data are valid: Yes (encrypted)
 > Password expiration: Expires in 190 days!
Checking WebADM group CN=ITWeb,CN=Users,DC=rcdevs,DC=com...
 > Contains members: Yes (2 members)
Checking user CN=test1,CN=Users,DC=rcdevs,DC=com...
 > Password expiration: Expires in 131 days!
Checking WebADM account CN=krbtgt,CN=Users,DC=rcdevs,DC=com...
 > Password expiration: Already expired
Checking WebADM group CN=Ordinateurs du domaine,CN=Users,DC=rcdevs,DC=com...
 > Contains members: No
Checking WebADM group CN=Contrôleurs de domaine,CN=Users,DC=rcdevs,DC=com...
 > Contains members: No
Checking WebADM group CN=Administrateurs du schéma,CN=Users,DC=rcdevs,DC=com...
 > Contains members: Yes (1 members)
Checking WebADM group CN=Administrateurs de l'entreprise,CN=Users,DC=rcdevs,DC=com...
 > Contains members: Yes (1 members)
Checking group CN=Éditeurs de certificats,CN=Users,DC=rcdevs,DC=com...
 > Contains members: Yes (1 members)
Checking WebADM group CN=Admins du domaine,CN=Users,DC=rcdevs,DC=com...
 > Contains members: Yes (4 members)
Checking WebADM group CN=domain admins,CN=Users,DC=rcdevs,DC=com...
 > Contains members: Yes (1 members)
Checking WebADM group CN=Utilisateurs du domaine,CN=Users,DC=rcdevs,DC=com...
 > Contains members: No
Checking group CN=Invités du domaine,CN=Users,DC=rcdevs,DC=com...
 > Contains members: No
Checking WebADM account CN=testadfs,CN=Users,DC=rcdevs,DC=com...
 > WebADM Data are valid: Yes (encrypted)
 > Password expiration: Expires in 158 days!
Checking WebADM group CN=Propriétaires créateurs de la stratégie de
groupe,CN=Users,DC=rcdevs,DC=com...
 > Contains members: Yes (1 members)
Checking WebADM group CN=Serveurs RAS et IAS,CN=Users,DC=rcdevs,DC=com...
 > Contains members: No
Checking WebADM account CN=user,CN=Users,DC=rcdevs,DC=com...
 > WebADM Data are valid: Yes (encrypted)
 > Password expiration: Expires in 165 days!
Checking WebADM group CN=Groupe de réplication dont le mot de passe RODC est
autorisé,CN=Users,DC=rcdevs,DC=com...
 > Contains members: No
Checking WebADM group CN=Groupe de réplication dont le mot de passe RODC est
refusé,CN=Users,DC=rcdevs,DC=com...
 > Contains members: Yes (8 members)
```

```
 > Contains members: Yes (8 members)
Checking group CN=Contrôleurs de domaine en lecture seule,CN=Users,DC=rcdevs,DC=com...
 > Contains members: No
Checking WebADM group CN=Contrôleurs de domaine d'entreprise en lecture
seule,CN=Users,DC=rcdevs,DC=com...
 > Contains members: No
Checking WebADM group CN=Contrôleurs de domaine clonables,CN=Users,DC=rcdevs,DC=com...
 > Contains members: No
Checking group CN=Protected Users,CN=Users,DC=rcdevs,DC=com...
 > Contains members: No
Checking group CN=gpWebADM,CN=Users,DC=rcdevs,DC=com...
 > Contains members: No
Checking WebADM group CN=DnsAdmins,CN=Users,DC=rcdevs,DC=com...
 > Contains members: No
Checking WebADM group CN=DnsUpdateProxy,CN=Users,DC=rcdevs,DC=com...
 > Contains members: No
Checking WebADM account CN=proxyuser,CN=Users,DC=rcdevs,DC=com...
 > Password expiration: Never expires
Checking WebADM account CN=adfs user,CN=Users,DC=rcdevs,DC=com...
 > Password expiration: Never expires
Checking WebADM account CN=vagrant,CN=Users,DC=rcdevs,DC=com...
 > WebADM Settings are valid: Yes
 > WebADM Data are valid: Yes (encrypted)
 > Password expiration: Never expires

Verified 37 LDAP objects in 0 seconds (14 errors).
```

11. webadm

This binary is used to start, stop & restart WebADM services. It can also be used to check the status, version, and updates.

```
[root@webadm bin]# ./webadm
Usage: ./webadm {start|stop|restart|status|version|update}
    start|stop|restart: start, stop and restart WebADM
    status: check the running status of WebADM
    version: display WebADM version information
    update: check if there is an update available online
```

```
[root@webadm scripts]#webadm bin]# ./webadm status
WebADM HTTP server is running with PID 84199.
WebADM Watchd server is running with PID 84194.
WebADM Session server is running with PID 84185.
WebADM PKI server is running with PID 84192.
```

```
[root@webadm bin]# ./webadm version
RCDevs WebADM version 1.6.8 (64bit) built October 8 2018
Copyright (c) 2010-2018 RCDevs SA, All rights reserved
```

12. yubihsm

This script can be used only if you have an HSM plugged and configured with WebADM. This script shows the HSM status, how many HSM are plugged to WebADM servers and the serial numbers.

```
[root@webadm scripts]# /opt/webadm/bin/yubihsm

YubiHSM Unlocker tool version 1.4.3 (382c1ecf4eb60dae767412e424d001cd87ea947e)
Copyright (c) 2010-2017 RCDevs SA, All rights reserved.

Found 1 HSM device:
1) YubiHSM N° 8D9626675057: UNLOCKED
```

13. backup_all

The backup_all script is similar to the previous backup script, but this one includes other RCDevs products installed under /opt/ folder.

```
[root@webadm scripts]# ./backup_all backup
Starting full backup
Are you sure you want to backup OpenOTP LDAP Bridge (y/n)? y
Adding conf/ldproxy.conf... Ok
Adding conf/ldproxy.crt... Ok
Adding conf/ldproxy.key... Ok
Adding temp... Ok
Adding logs... Ok
Adding conf/ldproxy.csr... Missing
Adding conf/ldproxy.env... Missing
Compressing backup file... Ok

OpenOTP LDAP Bridge backup created in /tmp/ldproxy.bkp.gz

Are you sure you want to backup OpenOTP RADIUS Bridge (y/n)? y
Adding conf/clients.conf... Ok
Adding conf/radiusd.conf... Ok
Adding conf/radiusd.crt... Ok
Adding conf/radiusd.key... Ok
Adding logs... Ok
Adding temp... Ok
Adding conf/radiusd.csr... Missing
Adding conf/radiusd.env... Missing
Compressing backup file... Ok

OpenOTP RADIUS Bridge backup created in /tmp/radiusd.bkp.gz

Are you sure you want to backup WebADM (y/n)? y
Adding conf/webadm.conf... Ok
Adding conf/rsignd.conf... Ok
Adding conf/objects.xml... Ok
Adding conf/servers.xml... Ok
Adding pki... Ok
Adding logs... Ok
Adding temp... Ok
Adding conf/license.key... Ok
Adding conf/webadm.env... Ok
Adding conf/custom.crt... Missing
Adding conf/custom.key... Missing
Adding websrvs/smshub/smsc/custom1.php... Ok
Adding websrvs/smshub/smsc/custom2.php... Ok
Adding websrvs/smshub/smsc/custom3.php... Ok
Adding websrvs/smshub/smsc/custom4.php... Ok
Compressing backup file... Ok

WebADM backup created in /tmp/webadm.bkp.gz

Compressing backup file... Ok
Backup created in backup.gz
```

14. create_cert

This script allows you to generate a new self-signed certificate for the WebADM GUI signed by the WebADM CA.

```
[root@webadm scripts]# ./create_cert
Enter the server fully qualified host name (FQDN): webadm.rcdevs.com
Generating HTTP server private key... Ok
Creating HTTP server certificate request... Ok
Signing HTTP server certificate with WebADM CA... Ok
WebADM HTTP certificate successfully updated.
Do you want to restart WebADM (y/n)? y
Stopping WebADM HTTP server... Ok
Stopping WebADM Watchd server......... Ok
Stopping WebADM Session server... Ok
Checking libudev dependency... Ok
Checking system architecture... Ok
Checking server configurations... Ok

Found Trial Enterprise license (RCDEVSSUPPORT)
Licensed by RCDevs SA to RCDevs Support
Licensed product(s): OpenOTP

Starting WebADM Session server... Ok
Starting WebADM PKI server... Ok
Starting WebADM Watchd server... Ok
Starting WebADM HTTP server... Ok

Checking server connections. Please wait...
Connected LDAP server: YO_AD-DC (192.168.3.50)
Connected SQL server: SQL Server (192.168.3.58)
Connected PKI server: PKI Server (127.0.0.1)
Connected Mail server: SMTP Server (78.141.172.203)
Connected Push server: Push Server (91.134.128.157)
Connected Session server: Session Server (127.0.0.1)
Connected License server: License Server (91.134.128.157)

Checking LDAP proxy user access... Ok
Checking SQL database access... Ok
Checking PKI service access... Ok
Checking Mail service access... Ok
Checking Push service access... Ok
```

15. create_mysqldb

This script will create the database, user, password and will set the rights for the user created on the database who will be used by WebADM. Note that MySQL packages should be installed before running this script. You can edit the script to edit the database name, the username and the password. By default, these 3 values are "webadm".

```
[root@webadm scripts]# cat create_mysqldb
#!/bin/bash
# This SQL script initializes the WebADM database on a MySQL Server

DATABASE="webadm"
USERNAME="webadm"
PASSWORD="webadm"

Q1="create database if not exists $DATABASE;"
Q2="grant usage on $DATABASE.* to $USERNAME@localhost identified by '$PASSWORD';"
Q3="grant all privileges on $DATABASE.* to $USERNAME@localhost;"
SQL="${Q1}${Q2}${Q3}"

mysql -u root -p -e "$SQL"
```

To execute this script :

```
[root@webadm scripts]# ./create_mysqldb
```

16. repair_perms

This script can be run when you want to repair permissions on WebADM configuration files.

```
[root@webadm1 scripts]# ./repair_perms
Resetting WebADM file permissions... Ok
```