Rockefeller Archive Center

LDAP SCHEMA, USER/GROUP MANAGEMENT SCRIPTS DOCUMENTATION

Duane K Beyer
MARIST COLLEGE | 2017

Contents

Overview:

This document is an overview of sftp setup on the Rockefeller Archive Center server LXFTPP1. Outside users sftp to the A10 proxy on port 80. The A10 then forwards the packets to port 12060 on the SLES 12 server running on Linux for zSeries,   LXFTPP1. This document details the configuration of the sshd server running on LXFTPP1 and the scripts and programs that setup the users, groups and directories used to sftp files to the server.

Scripts and programs:

The following is a list of bash scripts and programs used to setup the environment. These scripts are located in: /usr/local/bin/ on the server.

RACaddorg - Rockefeller Archive Center - Add Organization Group
RACcreateuser - Rockefeller Archive Center - Create a RAxxxxx User
RACadd2grp - Rockefeller Archive Center - Add user to Group
RACdeluser - Rockefeller Archive Center - Delete User

Assumptions:

Users must exist in the GO LDAP server before they are added with RACadduser.
Groups and users are tracked outside of the scope of these scripts.
LDAP data is updated outside of the scope of these scripts. (scripts are LDAP read only).
User ID's are in the form: RAxxxxx where xxxxx is a number with leading zeros.
    See "uid" in LDAP section for details.
ORG ID's are in the form ogr*xxx* where xxx is a number. Leading zeroes are not used.
    ORG ID's are the Group ID on the Linux system.
The location of the sshd_config file is set in RACaddorg, as following:

```
sshdfile='/etc/ssh2/sshd_config'
```

LDAP schema:

Ldap records for Rockefeller Archive Center are made up of four object Classes.
inetOrgPerson, Person, posixAccount and top.cd .   Items in yellow are the most commonly used.

Person:

```
objectclass ( 2.5.6.6 NAME 'person'
    DESC 'RFC2256: a person'
    SUP top STRUCTURAL
    MUST ( sn $ cn )
    MAY ( userPassword $ telephoneNumber $ seeAlso $ description ) )
```

posixAccount:

```
objectclass ( 1.3.6.1.1.1.2.0 NAME 'posixAccount'
    DESC 'Abstraction of an account with POSIX attributes'
    SUP top AUXILIARY
    MUST ( cn $ uid $ uidNumber $ gidNumber $ homeDirectory )
    MAY ( userPassword $ loginShell $ gecos $ description ) )
```

inetOrgPerson:

```
objectclass     ( 2.16.840.1.113730.3.2.2
    NAME 'inetOrgPerson'
        DESC 'RFC2798: Internet Organizational Person'
    SUP organizationalPerson
    STRUCTURAL
        MAY (
                audio $ businessCategory $ carLicense $ departmentNumber $
                displayName $ employeeNumber $ employeeType $ givenName $
                homePhone $ homePostalAddress $ initials $ jpegPhoto $
                labeledURI $ mail $ manager $ mobile $ o $ pager $
                photo $ roomNumber $ secretary $ uid $ userCertificate $
                x500uniqueIdentifier $ preferredLanguage $
                userSMIMECertificate $ userPKCS12 )
        )
```

RAC LDAP user attributes.

```
ca=Customer Name – Initially set to uid.
uid=RAxxxxx account
          xxxxx=00001-99999
uidNumber= 9xxxxx   (Leading zeros are not supported,  so I lead the number with 9.
          Xxxxx=00001-99999
gidNumber=12345   Not used.  Just a place holder.
homeDirectory=na
```

LDIF files to create dc=ROCK and ou=People.

```
# Add new dc=ROCK  ROCK.org.gke
dn: dc=ROCK,dc=org,dc=gke
objectClass: dcObject
objectClass: organization
dc: ROCK
o: Rockefeller Archive Center

# Add new dc=ROCK  ROCK.org.gke
dn: ou=People,dc=ROCK,dc=org,dc=gke
objectClass: top
objectClass: organizationalUnit
ou: People
```

```
ldapadd -h x.x.x.x -p 389 -D "cn=Manager,ou=Policies,dc=gke"  -w xxxxxxxxxxx -f makeRACusers.ldif
```

Authorizing scripts:

All the scripts and programs need to be run with root authority using sudo. However, sudo requires that you enter a password. By adding the programs to sudo files with NOPASSWD:, the user authorized to run the scripts will not be prompted for a password. In out setup, the scripts will be called by apache who runs under id wwwrun. The following are the updates that need to be made to the sudo file. In our case we are using visudo.

```
#
# The following is used for Rocefeller Archive Center (Must be at Bottom)
#
wwwrun ALL=(ALL) NOPASSWD: /usr/sbin/useradd
wwwrun ALL=(ALL) NOPASSWD: /usr/sbin/groupadd
wwwrun ALL=(ALL) NOPASSWD: /usr/sbin/groupdel
wwwrun ALL=(ALL) NOPASSWD: /usr/local/bin/RACadd2grp
wwwrun ALL=(ALL) NOPASSWD: /usr/local/bin/RACaddorg
wwwrun ALL=(ALL) NOPASSWD: /usr/local/bin/RACcreateuser
wwwrun ALL=(ALL) NOPASSWD: /usr/local/bin/RACdeluser
```

When executing the commands, specify the full path with sudo as follows.

```
        sudo /user/local/bin/urdb/RACaddorg  This is a test organization
```

(You will not be prompted for a password).
Additionally, sudo defaults to root, so you do not need to add “–u root” on the command.

Bash Script: RACaddorg.

Description:

RACaddorg - Rockefeller Archive Center, Add Organization.
This script will create the new organization on the server. On the server, Organizations are represented by Linux Groups. Once the Linux Group has been created, it then needs to update the sshd_config file on the Linux server. The sshd_config file defines the directory for the user will be placed into when sftp*ing* into the server on port 12060. The script will also create the directory structure defined in the sshd_config file. The final step is to restart the sshd server to pick up the new configuration.

Usage:

RACaddorg <Description/Name of Organization>
The description can include spaces. The script will assign the organization the next available ORGxxx value and write it to standard output (&1).

Processing:

1) Create the "Group" orgx
2) Create the Directories for the new org
   /data/orgx
   /data/orgx/upload
3) Set ownership and permissions for new directories
4) Update /etc/ssh2/sshd_config with new org and chown directory
5) Write the organization to standard output.
6) Restart the sshd server.

Logging:

Logging: Output and error messages are placed in the system log.
tail -f /var/log/messages to view output in real time

Output:

The organization (Linux group) will be written to the standard output device.

Return Codes:

1 - General Fail - Operation not completed.
2- No input parameters, Must have at least at least one arguments to run
3 - The group already exits.
4 - Unable to create Directory
8 - Directory or Object exists

C Program: RACcreateuser.c.

Description:

RACcreateuser - Rockefeller Archive Center - Create a RAxxxxx User
This is a “c” program modeled after laddsuer.c
Source file is: RACcreateuser.c

Usage:

```
RACcreateuser  <userid>
         The user must exist in LDAP.
```

Processing:

Logging:

Logging: error messages are logged in the system log.

```
tail -f /var/log/messages to view output in real time
```

Output:

The userid will be written to standard output device.

Return Codes:

1 - General Fail - Operation not completed.
2 – No input parameters, Must have at least at least one argument to run
3 - The user does not exist in LDAP.
6 - useradd failed, see system log for return code from usermod.

Bash Script: RACadd2grp:

Description:

RACadd2grp - Rockefeller Archive Center,   Add a User to an Organization.
This script will add a user to the group that is representing the organization. The group and user must exist before this script can be run.

Groups are created with the RACaddorg script.
Users are created with RACcreateuser.

By adding the user to the group, you are setting the "home" directory for the user when the sftp.

Usage:

```
RACadd2grp <group> <user>
```

Group – The group must be a valid, existing, org$_{xxx}$ group.

User – The user must be a valid RAxxxxx id that exits on the system.

Processing:

Add a USER to a "org" group
1) Verify the user exists
2) Verify the group exists
3) Add the user to the group.

Logging:

Logging: Output and error messages are placed in the the system log.

```
tail -f /var/log/messages to view output in real time
```

Output:

No output, only a Return Code

Return Codes:

1 - General Fail - Operation not completed.
2 - No input parameters, Must have at least at least two arguments to run
3 - The user does not exist.
4 - The Group does not exist.
5 - The user is already part of a "org" group. Only allowed in 1 Rockefeller Archive groupi
6 - usermod failed, see system log for return code from usermod.

Bash Script: RACdeluser.

Description:

- Rockefeller Archive Center, Delete User.

This script will remove a user from the server. The user will remain in LDAP.

Usage:

RACdeluser <user>

Users is a value between RA00001 and RA99999 (upper case).

Processing:

1) Verify the user exists
2) Delete the user from system , not ldap.

Logging:

Output and error messages are placed in the the system log.

tail -f /var/log/messages to view output in real time

Output:

No output, only a Return Code

Return Codes:

1 - General Fail - Operation not completed.
2 - No input parameters, Must have at least at least two arguments to run
3 - The user does not exits.
6 - userdel failed, see system log for return code from usermod.

Sample LDAP output:

The output below was generated with the ldapsearch command:

```
ldapsearch -h xxx.xxx.xxx.xxx -p 389 -D "cn=Manager,ou=Policies,dc=gke" -b dc=gke objectclass=* -w test > file
```

```
# extended LDIF
#
# LDAPv3
# base <dc=ROCK,dc=org,dc=gke> with scope subtree
# filter: objectclass=*
# requesting: ALL
#

# ROCK.org.gke
dn: dc=ROCK,dc=org,dc=gke
objectClass: dcObject
objectClass: organization
dc: ROCK
o: Rockefeller Archive Center

# People, ROCK.org.gke
dn: ou=People,dc=ROCK,dc=org,dc=gke
objectClass: top
objectClass: organizationalUnit
ou: People

# RA00001, People, ROCK.org.gke
dn: uid=RA00001,ou=People,dc=ROCK,dc=org,dc=gke
objectClass: person
objectClass: posixAccount
objectClass: top
objectClass: inetOrgPerson
cn:: UkEwMDAwMSAgICAgICAgIA==
uid: RA00001
sn: na
uidNumber: 900001
gidNumber: 12345
homeDirectory:: bmEgIA==
userPassword:: e1NTSEF9NUV0ZDNQOGJaS2dMNW1RWld4bnBsdUdWQnRkS2FGeDI=

# RA00002, People, ROCK.org.gke
dn: uid=RA00002,ou=People,dc=ROCK,dc=org,dc=gke
objectClass: person
objectClass: posixAccount
objectClass: top
cn: RA00002
```

```
uid: RA00002
sn: na
uidNumber: 900002
gidNumber: 12345
userPassword:: e1NTSEF9WFY3NHlFNTlHRk8xU0hzZ2JseDBpK004TS84UmNyVGY=
homeDirectory: na

# RA00003, People, ROCK.org.gke
dn: uid=RA00003,ou=People,dc=ROCK,dc=org,dc=gke
objectClass: person
objectClass: posixAccount
objectClass: top
cn: RA00003
uid: RA00003
sn: na
uidNumber: 900003
gidNumber: 12345
userPassword:: e1NTSEF9WFY3NHlFNTlHRk8xU0hzZ2JseDBpK004TS84UmNyVGY=
homeDirectory: na

# RA00004, People, ROCK.org.gke
dn: uid=RA00004,ou=People,dc=ROCK,dc=org,dc=gke
objectClass: person
objectClass: posixAccount
objectClass: top
cn: RA00004
uid: RA00004
sn: na
uidNumber: 900004
gidNumber: 12345
userPassword:: e1NTSEF9WFY3NHlFNTlHRk8xU0hzZ2JseDBpK004TS84UmNyVGY=
homeDirectory: na

# RA00005, People, ROCK.org.gke
dn: uid=RA00005,ou=People,dc=ROCK,dc=org,dc=gke
objectClass: person
objectClass: posixAccount
objectClass: top
cn: RA00005
uid: RA00005
sn: na
uidNumber: 900005
gidNumber: 12345
userPassword:: e1NTSEF9WFY3NHlFNTlHRk8xU0hzZ2JseDBpK004TS84UmNyVGY=
homeDirectory: na

# RA00006, People, ROCK.org.gke
dn: uid=RA00006,ou=People,dc=ROCK,dc=org,dc=gke
```

```
objectClass: person
objectClass: posixAccount
objectClass: top
cn: RA00006
uid: RA00006
sn: na
uidNumber: 900006
gidNumber: 12345
userPassword:: e1NTSEF9WFY3NHlFNTlHRk8xU0hzZ2JseDBpK004TS84UmNyVGY=
homeDirectory: na

# RA00007, People, ROCK.org.gke
dn: uid=RA00007,ou=People,dc=ROCK,dc=org,dc=gke
objectClass: person
objectClass: posixAccount
objectClass: top
cn: RA00007
uid: RA00007
sn: na
uidNumber: 900007
gidNumber: 12345
userPassword:: e1NTSEF9WFY3NHlFNTlHRk8xU0hzZ2JseDBpK004TS84UmNyVGY=
homeDirectory: na

# RA00008, People, ROCK.org.gke
dn: uid=RA00008,ou=People,dc=ROCK,dc=org,dc=gke
objectClass: person
objectClass: posixAccount
objectClass: top
cn: RA00008
uid: RA00008
sn: na
uidNumber: 900008
gidNumber: 12345
userPassword:: e1NTSEF9WFY3NHlFNTlHRk8xU0hzZ2JseDBpK004TS84UmNyVGY=
homeDirectory: na

# RA00009, People, ROCK.org.gke
dn: uid=RA00009,ou=People,dc=ROCK,dc=org,dc=gke
objectClass: person
objectClass: posixAccount
objectClass: top
cn: RA00009
uid: RA00009
sn: na
uidNumber: 900009
gidNumber: 12345
userPassword:: e1NTSEF9WFY3NHlFNTlHRk8xU0hzZ2JseDBpK004TS84UmNyVGY=
homeDirectory: na

# RA00010, People, ROCK.org.gke
dn: uid=RA00010,ou=People,dc=ROCK,dc=org,dc=gke
objectClass: person
objectClass: posixAccount
objectClass: top
cn: RA00010
uid: RA00010
sn: na
uidNumber: 900010
gidNumber: 12345
userPassword:: e1NTSEF9WFY3NHlFNTlHRk8xU0hzZ2JseDBpK004TS84UmNyVGY=
homeDirectory: na

# search result
search: 2
result: 0 Success

# numResponses: 13
# numEntries: 12
```

Manually Deleting an Organization:

Organization Groups should never be deleted from the server. However, for testing, it may be necessary to remove a organization. The following outlines the steps to be taking. To complete this task, you need to take three steps.

1) Remove the group with the linux command groupdel
    a. groupdel <groupname>
2) Remove the entry from /etc/ssh2/sshd_config
    a. Use a editor such as vi to remove the information about the group you are deleting. In this example, the group being removed is org1. Delete the block below.

```
# This section was created with the addorg script. Please DO NOT ERASE
# The addorg script is located in /usr/local/bin/addorg.
# This is for group: org1, Monkies in the park.

Match group org1
    ChrootDirectory  /data/org1
    X11Forwarding no
    AllowTcpForwarding no
    ForceCommand internal-sftp
```

3) Remove the directories where the files are stored. In this example /data/org1 is the directory used for org1. Use the Linux command rm –r.
    a. rm –r /data/org1