Rancid lab

Using RANCID

Introduction

Goals

- Gain experience with RANCID

Notes

- Commands preceded with "$" imply that you should execute the command as a general user - not as root.
- Commands preceded with "#" imply that you should be working as root.
- Commands with more specific command lines (e.g. "rtrX>" or "mysql>") imply that you are executing commands on remote equipment, or within another program.

Install rancid

One person in your group should connect srv1 using ssh, and become root:

```
$ sudo -s
#
```

Now install the Subversion Version Control System:

In addition to Subversion we will specify to install telnet and the mutt email client. Both these package may already be installed from prior exercises. If so, don't worry - the apt command will not reinstall them.

```
# apt install subversion telnet mutt
```

Install rancid itself:

```
# apt install rancid
```

- It will prompt with a warning - Select `<OK>` and press ENTER to continue.
- It will give you another warning about making a backup copy of your rancid data. We have no data, so select `<YES>` and press ENTER to continue.

Add alias

Add an alias for the rancid user in `/etc/aliases` file

RANCID by default sends emails to the users rancid-groupname and rancid-admin-groupname. We want them to be sent to the sysadm user instead and use the alias function for this.

```
# nano /etc/aliases
```

Add the following entries.

```
rancid-routers:         sysadm
rancid-admin-routers:   sysadm
```

Save the file, then run:

```
# newaliases
```

Configure rancid

Edit `/etc/rancid/rancid.conf`

```
# nano /etc/rancid/rancid.conf
```

Find these lines in rancid.conf:

```
# list of rancid groups
#LIST_OF_GROUPS="sl joebobisp"; export LIST_OF_GROUPS
```

And, underneath them add the following line:

```
LIST_OF_GROUPS="routers"
```

(with no '#' at the front of line, and aligned to the left)

Find the line with CVSROOT:

```
CVSROOT=$BASEDIR/CVS; export CVSROOT
```

And, change it to:

```
CVSROOT=$BASEDIR/svn; export CVSROOT
```

Note the lowercase "svn".

We want to use Subversion for our Version Control System, and not CVS, so find the line with the parameter RCSSYS:

```
RCSSYS=cvs; export RCSSYS
```

And, change it to:

```
RCSSYS=svn; export RCSSYS
```

Now exit and save the file.

Change to the rancid user

CRITICAL! CRITICAL! CRITICAL!

CRITICAL! CRITICAL! CRITICAL!

CRITICAL! CRITICAL! CRITICAL!

Pay very close attention to what userid you are using during the rest of these exercises. If you are not sure simply type "id" on the command line at any time.

From a root prompt ("#"), switch identity to become the 'rancid' user:

```
# su - rancid
```

Check that you ARE the rancid user:

```
$ id
```

You should see something similar (numbers may be different):

```
uid=117(rancid) gid=123(rancid) groups=123(rancid)
```

IF YOU ARE NOT USER RANCID NOW, do NOT continue!

Create `/var/lib/rancid/.cloginrc`

```
$ nano /var/lib/rancid/.cloginrc
```

Add the following lines to the file:

```
add method * {ssh}
add cyphertype *.ws.nsrc.org {aes128-ctr,3des-cbc}
add user *.ws.nsrc.org nmmlab
add password *.ws.nsrc.org lab-PW lab-EN
```

('nmmlab' is the username, 'lab-PW' is login password and 'lab-EN' is enable password used to login to your router. The star in the name means that it will try to use this username and password for all routers whose names end .ws.nsrc.org. The other settings force ssh, so that rancid will not fall back to insecure telnet)

Exit and save the file.

Now protect this file so that it cannot be read by other users:

```
$ chmod 600 /var/lib/rancid/.cloginrc
```

Test login to the router of your group

Login to your router with clogin. You might have to type yes to the first warning, but should not need to enter a password, this should be automatic.

```
$ /usr/lib/rancid/bin/clogin bdr1.campusY.ws.nsrc.org
```

(replace Y with your group number. So, group 1 is bdr.campus1.ws.nsrc.org)

You should get something like:

```
bdr1.campus6.ws.nsrc.org
spawn telnet bdr1.campus6.ws.nsrc.org
Trying 2001:db8:6::1...
Trying 100.68.6.1...
telnet: Unable to connect to remote host: Connection refused
spawn ssh -c 3des-cbc -x -l nmmlab bdr1.campus6.ws.nsrc.org
Password:
bdr1.campus6>enable
Password:
bdr1.campus6#
```

Exit the from the router login:

```
bdr1.campusY#exit
```

Initialize the SVN repository for rancid

Make sure you are the rancid user before doing this:

```
$ id
```

If you do not see something like

```
uid=117(rancid) gid=123(rancid) groups=123(rancid)
```

then DO NOT CONTINUE until you have become the rancid user. See earlier section for details.

Now initialize the Version Control repository (it will use Subversion):

```
$ /usr/lib/rancid/bin/rancid-cvs
```

You should see something similar to this:

```
Committing transaction...
Committed revision 1.
Checked out revision 1.
Updating '.':
At revision 1.
A         configs
Adding          configs
Committing transaction...
Committed revision 2.
A         router.db
Adding          router.db
Transmitting file data .done
Committing transaction...
Committed revision 3.
```

Do the following ONLY if you have problems

If this does not work, then either you are missing the subversion package, or something was not properly configured during the previous steps. You should verify that subversion is installed and then before running the rancid-cvs command again do the following:

```
$ exit
# apt install subversion
# su - rancid
$ cd /var/lib/rancid
```

```
$ rm -rf routers
$ rm -rf svn
```

Now try running the rancid-cvs command again:

```
$ /usr/lib/rancid/bin/rancid-cvs
```

Create the router.db file

```
$ nano /var/lib/rancid/routers/router.db
```

Add this line (NO spaces at the beginning please):

```
bdr1.campusY.ws.nsrc.org;cisco;up
core1.campusY.ws.nsrc.org;cisco;up
```

(remember to replace Y with your group number as appropriate)

Exit and save the file.

Let's run rancid!

Still as the rancid user:

```
$ rancid-run
```

This may take some time so be patient.

Run it again, since the first time it might not commit correctly:

```
$ rancid-run
```

Check the rancid log files:

```
$ cd /var/lib/rancid/logs
$ ls -l
```

… View the contents of the file(s):

```
$ less routers.*
```

NOTE! Using "less" - to see the next file press ":n". To see the Previous file press ":p". To exit from less press "q".

Look at the configs

```
$ cd /var/lib/rancid/routers/configs
$ less bdr1.campusY.ws.nsrc.org
```

Where you should replace "Y" with your group number.

If all went well, you can see the config of the router.

Tracking changes

Let's change an interface Description on the router

```
$ /usr/lib/rancid/bin/clogin bdr1.campusY.ws.nsrc.org
```

Where you should replace "Y" with your group number.

At the "bdr1.campusY#" prompt, enter the command:

```
bdr1.campusY# conf term
```

You should see:

```
Enter configuration commands, one per line.  End with CNTL/Z.
bdr1.campusY(config)#
```

Enter:

```
bdr1.campusY(config)# interface LoopbackXX      (replace XX with
your server number)
```

You should get this prompt:

```
bdr1.campusY(config-if)#
```

Enter:

```
bdr1.campusY(config-if)# description <put your name here>
bdr1.campusY(config-if)# end
```

You should now have this prompt:

```
bdr1.campusY#
```

To save the config to memory:

```
bdr1.campusY# write memory
```

You should see:

```
Building configuration...
[OK]
```

To exit type:

```
rtrX# exit
```

Now you should be back at your rancid user prompt on your system:

Let's run rancid again

```
$ rancid-run
```

It will take some time to pull the latest router config file. Look at the rancid logs after it's completed.

```
$ ls /var/lib/rancid/logs/
```

You should see the latest rancid execution as a new log file with the date and time in the name.

Let's see the differences

```
$ cd /var/lib/rancid/routers/configs
$ ls -l
```

You should see the router config file for your group:

```
$ svn log bdr1.campusY.ws.nsrc.org
```

(where Y is the number of your group)

Notice the revisions. You should see different revision numbers such as r6, r9 and r8. Choose the lowest and the highest one.

```
------------------------------------------------------------
--------
```

```
r9 | rancid | 2017-02-21 02:20:38 +0000 (Tue, 21 Feb 2017) | 1
line

updates
--------------------------------------------------------------------
--------
r8 | rancid | 2017-02-21 02:02:47 +0000 (Tue, 21 Feb 2017) | 1
line

updates
--------------------------------------------------------------------
--------
r6 | rancid | 2017-02-21 02:01:25 +0000 (Tue, 21 Feb 2017) | 1
line

new router
--------------------------------------------------------------------
--------
```

Let's view the difference between two versions:

```
$ svn diff -r6:9 bdr1.campusY.ws.nsrc.org | less
$ svn diff -r8:9 bdr1.campusY.ws.nsrc.org | less
```

… can you find your changes?

Notice that svn is the Subversion Version Control system command line tool for viewing
Subversion repositories of information. If you type:

```
$ cd /var/lib/rancid/routers
$ ls -lah
```

You will see a hidden directory called `.svn` - this actually contains all the information about the
changes between router configurations from each time you run rancid using rancid-run.

Whatever you do, DO NOT EDIT or touch the `.svn` directory by hand!

Check your mail

Now we will exit from the rancid user shell and the root user shell to go back to being the "sysadm"
user. Then we'll use the "mutt" email client to see if rancid has been sending emails to the sysadm
user.

```
$ exit                    (takes your from rancid to root user)
# exit                    (take you from root to sysadm user)
$ id
```

… check that you are now the 'sysadm' user again;

… if not, log out and in again as sysadm to your virtual host

```
$ mutt
```

(When asked to create the Mail directory, say Yes)

If everything goes as planned, you should be able to read the mails sent by Rancid. You can select an email sent by "rancid@hostX.campusY.ws.nsrc.org" and see what it looks like.

Notice that it is your router description and any differences from the last time it was obtained using the rancid-run command.

Now exit from mutt.

(use 'q' return to mail index, and 'q' again to quit mutt)

Cron configuration

Let's make rancid run automatically every 30 minutes from using cron

cron is a system available in Linux to automate the running of jobs. First we need to become the root user again:

```
$ sudo -s
#
```

Now edit the file /etc/cron.d/rancid:

```
# nano /etc/cron.d/rancid
```

Find this line:

```
#1 * * * * rancid /usr/bin/rancid-run
```

and change it to the following, replacing "X" with your group number:

```
X-3X/30 * * * * rancid /usr/bin/rancid-run
```

(e.g. group 2 would start the line with 2-32/30 , removing the hash as well)

That's it. The command "rancid-run" will execute automatically from now on every 30 minutes all the time (every day, week and month). Each group will start at a slightly different time to avoid overloading the training platform.

More devices

Become the rancid user and update the router.db file:

```
# su -s /bin/bash rancid
$ nano /var/lib/rancid/routers/router.db
```

Add the two remaining devices in your campus so that the file ends up looking like this where "Y" is your campus number.

```
bdr1.campusY.ws.nsrc.org;cisco;up
core1.campusY.ws.nsrc.org;cisco;up
dist1-b1.campusY.ws.nsrc.org;cisco;up
dist1-b2.campusY.ws.nsrc.org;cisco;up
```

(Note that "cisco" means this is Cisco equipment – it tells Rancid that we are expecting to talk to a Cisco device here. You can also talk to Juniper, HP, …).

Be sure the entries are aligned to the left of the file.

Run rancid again (still as the 'rancid' user)

```
$ rancid-run
```

This should take a minute or more now, be patient.

Check out the logs:

```
$ cd /var/lib/rancid/logs
$ ls -l
```

… Pick the latest file and view it

```
$ less routers.YYYYMMDD.HHMMSS
```

This should be the last file listed in the output from "ls -l"

You should notice a bunch of statements indicating that routers have been added to the Subversion version control repository, and much more.

Look at the configs

```
$ cd /var/lib/rancid/routers/configs
$ less *.ws.nsrc.org
```

Press the SPACE bar to scroll through each file and then press ":n" to view the next file. Press "q" to quit at any time.

If all went well, you can see the configs of ALL routers

Re-run rancid

Run RANCID again just in case someone changed some configuration on the router

```
$ rancid-run
```

This could take a few moments, so be patient....

Play with clogin

```
$ /usr/lib/rancid/bin/clogin -c "show clock"
bdr1.campusY.ws.nsrc.org
```

Where "Y" is the number of your group.

What do you notice ?

Even better, we can show the power of using a simple script to make changes to multiple devices quickly:

```
$ nano /tmp/newuser
```

… in this file, add the following commands (COPY and PASTE):

```
configure terminal
username NewUser secret 0 NewPassword
end
write
```

Save the file, exit, and run the following commands from the command line:

```
$ for device in bdr1 core1 dist1-b1 dist1-b2
```

Your prompt will now change to be ">". Continue by typing:

```
> do
> /usr/lib/rancid/bin/clogin -x /tmp/newuser
$device.campusY.ws.nsrc.org
> done
```

Now your prompt will go back to "$" and rancid clogin command will run and execute the commands you just typed above on all of your network devices. This is simple shell scripting in Linux, but it's very powerful.

Q. How would you verify that this has executed correctly ? Hint: "show run | inc"

A. Connect to bdr1 and core1. Type "enable" and then type "show run | inc username" to verify that the NewUser username now exists. Type exit to leave each router. Naturally you could automate this like we just did above.

Install ViewVC

Now we will add the RANCID SVN (Subversion) repository into ViewVC so that you can browse configurations via the web.

If you are still logged in as user rancid, get back to root. Remember you can type "id" to check what userid you are.

```
$ exit
#
```

Install ViewVC:

```
# apt install viewvc
```

Add Rancid SVN report into ViewVC config file. Edit the file

```
# nano /etc/viewvc/viewvc.conf
```

Look for this line in the config file and configure as below;

```
#svn_roots = svn: /var/lib/svn
```

and underneath it add this line:

```
svn_roots = rancid: /var/lib/rancid/svn
```

Save and exit.

Next, we need to add apache web server user (www-data) into rancid group in order to grant access for apache webserver to view rancid's SVN repo.

```
# usermod -a -G rancid www-data
```

Restart apache server:

```
# systemctl restart apache2
```

Browse the rancid files from your Web browser!

- http://oob.srv1.campusY.ws.nsrc.org/cgi-bin/viewvc.cgi/rancid/
  (http://oob.srv1.campusY.ws.nsrc.org/cgi-bin/viewvc.cgi/rancid/)

Browse the files under the 'routers/configs' directory. You can see all your router configuration files here.

Securing ViewVC

You would not want the entire Internet to be able to browse your configuration files. Here are some steps you can take to secure WebSVN access. One step not included is to enforce the use of https (ssl) access. We recommend this for all your web sites wherever possible.

Create an Apache configuration file for ViewVC:

```
# cd /etc/apache2/conf-available
# nano viewvc.conf
```

In this file put:

```
<DirectoryMatch "^/usr/lib/cgi-bin/viewvc">
    AuthName "viewvc RANCID Repository Access"
    AuthType Basic
    AuthUserFile /etc/viewvc/htpasswd
    require valid-user
</DirectoryMatch>
```

Save and exit from the editor. Enable this configuration and restart apache2 for it to be activated:

```
# a2enconf viewvc
# systemctl reload apache2
```

Next we need to create a htpasswd file in the /etc/viewvc directory:

```
# cd /etc/viewvc
# htpasswd -c htpasswd sysadm
```

(The `-c` option should only be used the very first time you create this file, as it will delete any passwords already in it)

Provide a password for the user sysadm (maybe the class password?). You should see this:

```
New password:
Re-type new password:
Adding password for user sysadm
```

Secure the file:

```
# chgrp www-data htpasswd
# chmod 640 htpasswd
```

Try browsing to the WebSVN pages at http://oob.srv1.campusY.ws.nsrc.org/cgi-bin/viewvc.cgi and you should be asked for a username and password to be able to view the pages.

Now you should be able to login with the username and password you created.

Review revisions

ViewVC lets you see easily the changes between versions.

- Browse to http://oob.srv1.campusY.ws.nsrc.org/cgi-bin/viewvc.cgi/rancid/ (http://oob.srv1.campusY.ws.nsrc.org/cgi-bin/viewvc.cgi/rancid/) again, go to `routers/` then `configs/`
- Click on your router file (bdr1.campusY.ws.nsrc.org) name. You will get a new screen with title "Log of /routers/configs/bdr1.campus6.ws.nsrc.org"
- Look for "Diffs between" at the bottom of the screen.
- Key in version no. i.e 3 and 5 and click "Get Diffs"

This will show you the differences between two separate router configurations.

ViewVC is a convenient way to quickly see differences via a GUI between multiple configuration files. Note, this is a potential security hole so you should limit access to the URL http://hostX/cgi-bin/viewvc.cgi using passwords (and SSL) or appropriate access control lists.

Optional: Fetching configs with a non-privileged `rancid` user

In a production environment, we'd probably want to add a "rancid" user on the devices, without config privileges, but able to retrieve do a `show running-config`.

One way to do this, add a user in config mode:

```
bdr1.campusY# conf term
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
bdr1.campusY(config)# username rancid privilege 4 secret
password
bdr1.campusY(config)# privilege exec level 4 show running-config
view full
```

This creates a `rancid` user with privilege level 4. On the next line, we allow that user to execute `show running-config`

You also need to add the username and password to your /var/lib/rancid/.cloginrc

```
add method * {ssh}
add cyphertype *.ws.nsrc.org {aes128-ctr,3des-cbc}
add user *.ws.nsrc.org rancid
add password *.ws.nsrc.org password
add autoenable *.ws.nsrc.org 1
```

The `autoenable` means the user will be in the right privilege level immediately after login and no enable is needed to run `show running-config`

Note: try and look at the `clogin` manpage to find out how you can specify another user (for example: nmmlab) when using clogin interactively, to make changes with -c or -x (as shown above).

Notes

On the use of hostnames in RANCID vs. IP Addresses

Note: it is also allowed to use IP addresses, and one could also write:

```
add user 100.68.* nmmlab
add password 100.68.* lab-PW lab-EN
add user bdr1.ws.nsrc.org nmmlab
add password bdr1.ws.nsrc.org lab-PW lab-EN
```