

Rancid Server Build and Operation Overview

Installation

The installation will cover the setup and configuration of a RANCID server using Fedora 15

Basic List of software

- OS = Fedora15
- Rancid = Really Awesome New Cisco config Differ (<http://www.shrubbery.net/rancid/>) v2.3.6
- CVS = Concurrent Versions System (<http://www.nongnu.org/cvs/>) v1.11.23-15
- ViewVC = Front end to CVS (<http://www.viewvc.org/>) v1.1.11-1
- Python = Programming Language (<http://www.python.org/>) v2.7.1-7
- RCS = Revision Control System (<http://www.cs.purdue.edu/homes/trinkle/RCS/>) v5.7-37
- Apache = Linux Webserver (<http://httpd.apache.org/>) v2.2.17-10
- MySQL = Database (<http://www.mysql.com/>) v5.5.13-1
- MySQLdb = Python connector for mysql (<http://sourceforge.net/projects/mysql-python>)
- PHP = Widely-used general-purpose scripting language that is especially suited for

Web development and can be embedded into HTML (<http://www.php.net/>)

- Pygments – Generic Syntax Highlighter (<http://pygments.org/>)

The basic install is as follows. This is in no way a fully locked down system.

1) Install the base OS

2) Run “yum update” - this will do an upgrade of all packages on the OS

3) Reboot

4) Log into the console

5) Run the following commands (some packages may already be installed)

- yum -y install expect cvs python httpd mysql mysql-server gcc make autoconf gcc++ kernel-devel mod_python
- yum groupinstall “Development Tools” MySQL-python diffutils
- yum install php-common php-gd php-mcrypt php-pear php-pecl-memcache phpmhash php-mysql php-xml
- yum update

6) Reboot

7) Type in

- groupadd netadm
- useradd -g netadm -c "Networking Backups" -d /usr/local/rancid rancid

8) Run the following commands

- mkdir /usr/local/rancid/tar
- cd /usr/local/rancid/tar

9) Download the latest version of rancid and extract it (example below)

- wget: wget ftp://ftp.shrubby.net/pub/rancid/rancid-2.3.6.tar.gz
- tar -xvzf rancid-2.3.6.tar.gz
- cd rancid-2.3.6• ./configure --prefix=/usr/local/rancid/
- make install
- cp cloginrc.sample /usr/local/rancid/.cloginrc
- chmod 0640 /usr/local/rancid/.cloginrc
- chown -R rancid:netadm /usr/local/rancid/
- chmod 770 /usr/local/rancid/

10) Edit /usr/local/rancid/etc/rancid.conf find the line where it starts LIST_OF_GROUPS= to look something like (change the name as necessary, this needs to be noted for further steps)

```
LIST_OF_GROUPS="networking"
```

11) Edit /etc/aliases and add something like the following

```
#  
# Rancid email addresses  
#  
rancid-admin-networking:      rancid-networking  
rancid-networking:           noc  
noc:                          support@test.com
```

12) Type in newaliases

13) Type in cvs (You should get some output re the CVS version, if it doesn't appear run yum install cvs)

14) Type in the following su – rancid

15) Type in /usr/local/rancid/bin/rancid-cvs output should look like the following

```
No conflicts created by this import  
cvs checkout: Updating networking  
cvs checkout: Updating networking/configs  
cvs add: scheduling file `router.db' for addition  
cvs add: use 'cvs commit' to add this file permanently  
RCS file: /usr/local/rancid//var/CVS/networking/router.db,v  
done  
Checking in router.db;  
/usr/local/rancid//var/CVS/networking/router.db,v <-- router.db  
initial revision: 1.1  
done
```

16) Edit the scheduled tasks using crontab –e and add the following lines (the second entry should be on one line)

```
1 * * * * /usr/local/rancid/bin/rancid-run #hourly router dump
50 23 * * * /usr/bin/find /usr/local/rancid/var/logs -type f -mtime +2 -exec rm {} \;
```

17) Now we need to edit the device list and password files

- Edit /usr/local/rancid/var/networking/router.db adding a line similar to

```
192.168.1.1:cisco:up
```

- Edit /usr/local/rancid/.cloginrc add the username and password for the device similar to:

```
add user 192.168.1.1 jim
add password 192.168.1.1 password enablepassword
add method 192.168.100.1 ssh telnet (add this line for ssh only access, for telnet its not needed)
```

18) Now we should have a working copy of rancid to test we need to do the following

bin/clogin 192.168.1.1 (run this from /usr/local/rancid path)

You should see the logon process run by for your device and end up at the enable prompt

Exit this prompt and go back to your bash shell.

19) Now run bin/rancid-run (from /usr/local/rancid path), once this is completed check the log files in the following locations for any errors:

```
/usr/local/rancid/var/logs
```

Successful output will look like the following:

```
starting: Thu Jul 21 02:01:01 PDT 2005
Trying to get all of the configs.
All routers successfully completed.
cvs diff: Diffing .
cvs diff: Diffing configs
cvs commit: Examining .
cvs commit: Examining configs
ending: Thu Jul 21 02:01:06 PDT 2005
```

20) **It should also have created the file 192.168.1.1 under /usr/local/rancid/var/networking/configs this would contain the config of your device**

21) Now it is time to add a user interface to the CVS data base . Switch back to root.

22) Download the latest versions of ViewVC, RCS.

```
yum install rcs viewvc
```

23) Install Pygments

- Type `mkdir /root/python`
- Type `cd ~/python`
- Run: `wget http://peak.telecommunity.com/dist/ez_setup.py`
- Run: `python ./ez_setup.py`
- Run: `python ./ez_setup.py`
- Run: `easy_install babel`
- Run: `easy_install Genshi`
- Run: `easy_install Pygments`
- Run: `easy_install docutils`
- Run: `easy_install textile`

24) configure ViewVC

- Cd into `/etc/viewvc/`
- Edit `/etc/viewvc/viewvc.conf`

Change the following options to look like something below

```
#cvs_roots = cvs:
```

```
root_parents = /usr/local/rancid/var/CVS : cvs
```

```
rsc_path = /usr/local/bin/
```

```
address = <a href=mailto:support@test.com>IT Support</a>
```

```
use_encrypt = 1
```

```
encrypt_path = /usr/bin/
```

```
use_highlight = 1
```

```
highlight_path = /usr/bin
```

- Type `cp /usr/lib/python2.7/site-packages/viewvc/bin/cgi/*.cgi /var/www/cgi-bin`
- Type `chmod +x /var/www/cgi-bin/*.cgi`
- Type `chown apache:apache /var/www/cgi-bin/*.cgi`
- Edit `/etc/httpd/conf/httpd.conf` and make the `scriptaliases` section look similar to

```
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
```

```
ScriptAlias /viewvc /var/www/cgi-bin/viewvc.cgi
```

```
ScriptAlias /query /var/www/cgi-bin/query.cgi
```

- Type `/etc/init.d/httpd restart`

25) Add the apache user to the netadm group in /etc/group

```
vim /etc/group
```

26) Run the following commands

```
chkconfig --levels 2345 mysqld on
```

```
chkconfig --levels 2345 httpd on
```

27) Modify you IPtables rules to allow http from the required devices

28) Create the Check-in Database

- Create a user who has permissions to create a database or just use root (for root just use 2nd link) (<http://dev.mysql.com/doc/refman/5.1/en/adding-users.html> , <http://dev.mysql.com/doc/refman/5.0/en/resetting-permissions.html>)

```
vi /usr/lib/python2.7/site-packages/viewvc/bin/make-database
replace all TYPE= to ENGINE =
```

- Type `/usr/lib/python2.7/site-packages/viewvc/bin./make-database` enter details as required
- Edit `/etc/viewvc/ciewvc.conf` and change the details in the database section to look similar to the following:

```
[cvbdb]
enabled = 1
host = localhost
port = 3306
database_name = ViewVC
user = username
passwd = password
readonly_user = username
readonly_passwd = password
row_limit = 1000
```

- Type (one line) `/usr/lib/python2.7/site-packages/viewvc/bin./cvbdbadmin rebuild`

```
/usr/local/rancid/var/CVS/CVSROOT
```

29) Now reboot, and afterwards we should have a Web interface up and running go to <http://serverip/viewvc> and you should have a web interface up with a Repository Listing of networking

30) Click networking and there should be a sub folder called configs , click into that and you should see the device 192.168.1.1. If you click on this you should be able to navigate to see the config of that device

NB!!

If you want email notifications not to use DNS the following changes need to be made to forward the alerts out via email

- 1) Edit `/etc/mail/mailertable` with something along the lines of

```
.domain.com smtp:[relayaddress]
domain.com smtp:[relayaddress]
```

- 2) Type in `makemap hash /etc/mail/mailertable < /etc/mail/mailertable`

- 3) Type in `/etc/init.d/sendmail start`

- 4) Type in `chkconfig sendmail on`

Important directories

- /usr/local/rancid (where rancid is installed, all files are stored here)
- /usr/local/rancid/bin (the rancid executable directory specific note needs to be made of the following files control_rancid, rancid more on these later)
- /usr/local/rancid/etc (where the rancid config file is stored)
- /usr/local/rancid/var (where the device config files are stored and where the CVS repository is stored)
- /usr/local/viewvc-1.1.6 (this is where view ViewVC is installed and configured)
- /var/www (this is where apache is installed and running from)

Important files

- /usr/local/rancid/.cloginrc (password and per device configuration file)
- /usr/local/rancid/etc/rancid.conf (main rancid config file)
- /usr/local/rancid/var/networking/router.db (list of devices to backup up=backup, down=dontbackup)
- /usr/local/rancid/bin/clogin (application that actually logs into routers)
- /usr/local/rancid/bin/control_rancid (file where you can specify the mail from address, search for sendmail eg sendmail -fsupport@test.com -FRancid -t)
- /usr/local/rancid/bin/rancid (this is main file for rancid, you can add exceptions here eg

Under the Sub ShowFlash

```
#ProcessHistory("FLASH","", "", "!Flash: $_");
```

```
# Modified to exclude crypto_archive on firewalls
```

```
  ^s+(crypto_archive)$/ && next;
```

```
  ^s+crypto_archive\crypto_arch_1.bin$/ && next;
```

```
  ^s+crypto_archive\crypto_arch_2.bin$/ && next;
```

```
#filter coredumpinfo/coredump.cfg
```

```
  ^s+(coredumpinfo)$/ && next;
```

```
^s+coredumpinfo$/ && next;
```

```
^s+coredumpinfo\coredump\.cfg$/ && next;
```

```
ProcessHistory("FLASH", "", "", "!Flash: $_");
```

- /usr/local/rancid/bin/alogin (this is login file for alteon switches, HP blade switches), you can add the username update here

Figure out prompts

```
set u_prompt [find userprompt $router]   if { "$u_prompt" == "" } {  
    set u_prompt "(Username|username|login| Login):"  
} else {  
    set u_prompt [join [lindex $u_prompt 0] ""]
```

References:

http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO:_Ch1:_Network_Backups_With_Rancid

<http://slaptijack.com/system-administration/quick-tip-install-cvs-before-rancid/>

<http://info.routermonkey.org/index.php?op=ViewArticle&articleId=5&blogId=1>

<http://ximbiot.com/cvs/wiki/CVS%20FAQ>

Operation

How to add a Device

1) If adding a new customer edit /usr/local/rancid/etc/rancid.conf (via nano or winscp) Adding to Customers name to the line starting with LIST_OF_GROUPS=

2) If adding a new customer edit /etc/aliases adding lines similar to the following, Just above the last line (helpdesk: support@test.com)

```
rancid-admin-group:      rancid-group  
rancid-group:           helpdesk
```

3) Now if you added a new Customer you need to ssh into the box as root and complete step 7, if not move to step 8

4) Once logged in type newaliases , this should come back with no errors

5) Type in su – rancid

6) Now If you have added a new Customer run `rancid-cvs` , if you have only added a new device move to step 10, The output will be similar to the output below, not the errors (I need to still look into why these are happening but it still works for now so ignore them)

```
[rancid@ran01 ~]$ rancid-cvs
No conflicts created by this import
ERROR(viewvc-logininfo): Bad arguments
cvs checkout: Updating Group
ERROR(viewvc-logininfo): Bad arguments
Directory /usr/local/rancid/var/CVS/Group/configs added to the
repository
cvs commit: Examining configs
cvs add: scheduling file `router.db' for addition
cvs add: use 'cvs commit' to add this file permanently
RCS file: /usr/local/rancid/var/CVS/Group/router.db,v
done
Checking in router.db;
/usr/local/rancid/var/CVS/Group/router.db,v <-- router.db
initial revision: 1.1
done
```

```
ERROR(viewvc-logininfo): Bad arguments
```

7) Now edit, and save, `/usr/local/rancid/.cloginrc` (This is a hidden file) entering the device details as required. There are a number of examples in the file, but basic ally it would look like something below

```
#Customername
add method *.Customername ssh
add user *.Customername username
add password *.Customername userpassword enablepassword
```

8) Now edit, and save, `/usr/local/rancid/var/clientname/router.db` and add the device to be monitored in the following format, `devicename :device type:up eg router:cisco:up`

9) Now from the console that we still have open run `clogin devicename`, This will test that everything is right with the config , you should get enable prompt, it will also add the ssh key (if using ssh) to the `/usr/local/rancid/.ssh/ known_hosts` file

10) Now run `rancid-run Groupname` , this will go and collect the device config

11) Now if you browse to `http://rancidserverip` and login you should be able to navigate to your device and view its config, thats if everything worked of course.

How to run the same commands across multiple devices

This outlines how to save a set of commands and then run this against multiple devices one by one. All commands need to be run as the rancid user

1) Create a file that the rancid user has access to that has a list of commands that you want to enter onto the device, you also need to also need to list any commands to get into config mode etc. Example below

```
conf t
username Fred password hxyksikheosPolst^ privilege 15
end
write mem
```

2) Now run the following command against all devices needed

```
clogin -x /path/to/file.file device.name.in.rancid
```