FreeRADIUS Install and Configuration

Joel Jaeggli
05/04/2006

What is RADIUS?

- A AAA protocol (Authentication, Authorization and Accounting).
- Authentication – Confirmation that the user is who they say they are. Authentication is accomplished through the presentation of credentials.
- Authorization – Granting access to specific types of service or resource.
- Accounting – Tracking the consumption of resources.

What is RADIUS? - Continued

- Radius was originally developed by Livingston for the Portmaster series of network access/terminal servers.
- Remote authentication dial-in user service.
- Eventually it was published as RFC 2058 and 2059. The current incarnation is embodied in RFC 2865.

What does RADIUS do?

- A radius client, which originally would have been a NAS device, but now lots of services can leverage Radius for authentication.
- A radius client takes a user name, some client specific information and a password hashed using a secret shared with the radius server, and uses that to create an authentication request.

What does RADIUS do? - continued

- The server looks up the values presented in the authentication request from flat text files, unix password files, database servers or ldap. Hashes them to compare with the request hashed values, and returns an access-accept packet or reject packet on based on the success or failure of the authentication request.

Why do we need RADIUS?

- Lots of services that you might contemplate deploying require authentication. Maintaining separate sets of authentication information for multiple services has poor scaling properties and creates user unhappiness.
- Centralized management of passwords reduces the number of places in which they have to be stored, and makes them easier to secure.

Why do we need RADIUS? - continued

- AAA services are one of the cores sets of functionality for an ISP.

Other AAA services

- DIAMETER
- TACACS/TAC+
- LDAP – a subset of it's functionality
- Kerberos – identity and authentication

About freeRADIUS...

- FreeRADIUS is the premier open source radius server. In it's simplest form it is similar to Livingston RADIUS 2.0, but is also extensible and has a feature set considerably beyond that of traditional radius servers.
- Also... It's available at no cost.

Plan of Attack

- Build and install freeRADIUS.
- Configure and start the RADIUS server.
- Test authentication
- Convert a service to support Radius.

Installing

- `cd /usr/ports/distfiles`
- lets pre-populate distfiles off the the `e1` noc machine with the packages we need
- the packages are in:
  - ftp://noc.e1.ws.afnog.org/distfiles/freeradius/
- Ok, where in the ports collection is freeradius?
- `/usr/ports/net/freeradius`
- make install
- Select any options you might need (none for now)...
- Watch it build and install...

Configuring – Part 1

- Notice that when freeRADIUS installed everything when in various subdirs of `/usr/local/`, this is typical of FreeBSD ports installations.
- Key in this case are:
  - The rc file in `/usr/local/etc/rc.d`
  - The configuration files located in `/usr/local/etc/raddb`
- Note at a minimum it is necessary to rename some files and enable radiusd in the `/etc/rc.conf` before the service will be able to start.

Configuring – Part 2

- Note, radius is a complex service, while there is copious documentation some of it is only present in the config files themselves which require careful reading.
- One of the most important to tools in understanding how config changes affect the radius server is this ability to run it by hand in debug mode. Debug mode is enabled by running: `radiusd -x`
- If you do that now you will note that it refuses to start.

Configuring – Part 3

- In `/usr/local/etc/raddb` copy:
  - `raddb.conf.sample` to `raddb.conf`
  - `clients.conf.sample` to `clients.conf`
  - `proxy.conf.sample` to `proxy.conf`
  - `snmp.conf.sample` to `snmp.conf`
  - `eap.conf.sample` to `eap.conf`
  - `sql.conf.sample` to `sql.conf`
  - `dictionary.sample` to `dictionary`
  - `huntgroups.sample` to `huntgroups`

Configuring - Part 3 continued

- `hints.sample` to `hints`
- `users.sample` to `users`
- `acct_users.sample` to `acct_users`
- `preproxy_users.sample` to `preproxy_users`

- If you run `radiusd -x` it should indicate if you missed any files you need. If not it should indicate that it's ready to process requests.

Configuring – Part 4

- Lets test the radius server as it is now to see it it will respond to us.
- In another window type:
  - `radtest test test localhost 0 testing123`
- You should see the server receive the access-request and respond with an access-reject.
- Now try it with a user name and password that is valid on your machine.

Configuring – Part 5

- Note, that the shared secret we've been using testing123 is not very secret, so lets change it.
- edit `/usr/local/etc/raddb/clients.conf` note that the client that is currently configured is `127.0.0.1` (localhost)
- A secret can be up to 31 characters in length. Pick one that's more unique than testing123.

Secret (digression)

- From RFC 2865:
  - The secret (password shared between the client and the RADIUS server) SHOULD be at least as large and unguessable as a well-chosen password. It is preferred that the secret be at least 16 octets. This is to ensure a sufficiently large range for the secret to provide protection against exhaustive search attacks. The secret MUST NOT be empty (length 0) since this would allow packets to be trivially forged.
- I tend to prefer large random or pseudo-random numbers for strings.

Configuring - Part 6

- Now run radtest again, using a local username and password and your new secret.

Configuring a client

- Now that we have the server working we can configure a client to query the server.
- We could configure a NAS device if we had one.
- Authenticated services on FreeBSD (and Linux) use a facility called PAM (Pluggable Authentication Modules) which will allow you to query different (or multiple) authentication methods.

PAM – Part 1

- Lets allow the ssh service on our machine to authenticate against our radius server.
- services that leverage PAM have config files in `/etc/pam.d`
- take a look at the one for `sshd`
- add another auth module after pam_nologin
- `auth sufficient pam_radius.so`

Pam – Part 2

- We need to edit the file `/etc/radius.conf`, which probably doesn't exist yet.
- we need to add the line:
  - `auth 127.0.0.1 secret 1`
  - `secret is the better secret you picked`
- Once we've done that we should be able to ssh to localhost enter our password and login, and you should see the results displayed by your radius daemon running in debug mode.

Making radiusd start with FreeBSD

- look at the rc file for radiusd which is located in `/usr/local/etc/rc.d/`
- Notice at the top that it provides instructions.
- Follow them...
- Then kill your current radiusd and start a new one by running `/usr/local/etc/rc.d/radiusd.sh \ start`

What have we achieved?

- We have a radius server that answers authentication queries using the unix password files/database on FreeBSD.
- We can deploy new services, like for example SMTP-AUTH without having to populate them with user credentials.

What more could we do?

- Store credentials in a database such as mysql, or a directory service such as ldap so that we could associate additional meta-data about the user with the account.
- Generate accounting data, so that we could bill for timed access to resources (at a wireless hotspot or a hotel for example).

Bibliography

- FreeRADIUS - http://www.freeradius.org/
- FreeBSD PAM - http://www.freebsd.org/doc/en_US.ISO8859-1/articles/pam/index.html
- PAM RADIUS man page - http://www.freebsd.org/cgi/man.cgi?query=pam_radius&sektion=8