FreeRADIUS Install and Configuration

Frank A. Kuse

27/05/2008

What is Radius?

- Radius is a protocol for carrying information related to authentication, authorization, and Accounting.
- Authentication: This refers to confirmation that a user who is requesting a service is a valid user. It's usually accomplished via the presentation of an identity and credentials. Examples of credentials are username, passwords, digital certificates and phone numbers

What is RADIUS–Continued?

- Authorization: Refers to the granting of specific type of service (including "no service") to a user, based on their authentication. This may actually be based on restrictions, for example time-of-day restrictions, or physical location restrictions, or restriction against multiple logins by the same user. Example of services include IP address filtering, address assignment, route assignment, encryption, bandwidth control/traffic management.

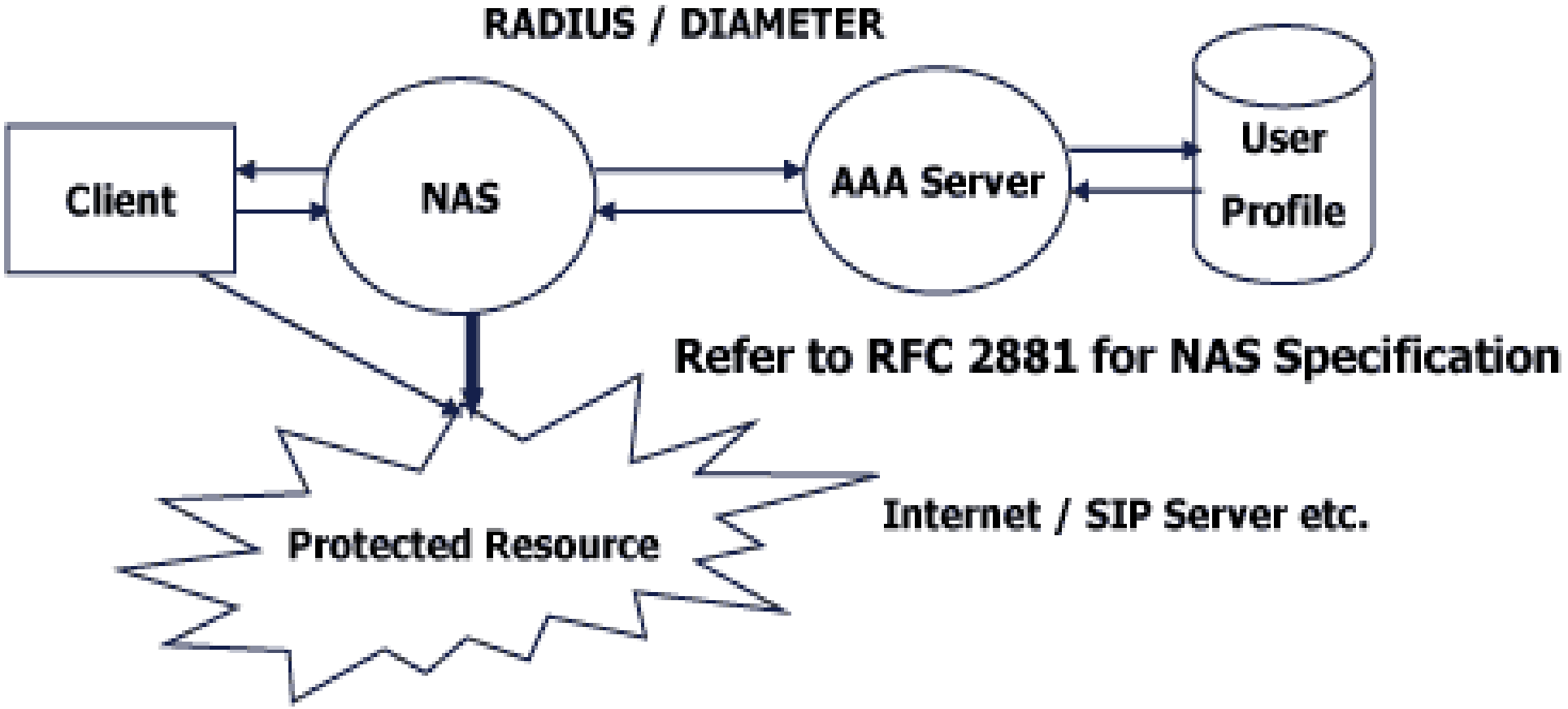
What is RADIUS? – Continued

- Accounting: This refers to the tracking of the consumption of network resources by users. Typical information that is gathered in accounting is the identity of the user, the nature of the service delivered, when the service began and when it ended
- Radius started it's history through Livingston who implemented the prototype Radius client in their "portmaster" dial-in server product and created a simple Radius server software to support it.
- Remote authentication dial-in user service.
- Eventually it was published as RFC 2058 and 2059. The current incarnation is embodied in RFC 2865.

What is (NAS)?

- The Network Access Server (NAS) is a service element that clients dial in to get access to network. A Network Access Server is a device which usually has interfaces both to the backbone and to the Telco and receives calls from hosts that want to access the backbone by dialup services. A NAS is located at an internet provider's point of presence to give their customers internet access.

Basic Architecture of NAS/Radius/AAA



Basic Architecture for NAS/RADIUS/AAA

What does RADIUS do?

- 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.
- 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.
- It provides Flexible Authentication Mechanics as below :
  - Point-to-Point Protocol – PPP
  - Password authentication protocol – PAP
  - Challenge-handshake authentication protocol – CHAP
- Radius is extensible; most vendors of Radius hardware and software implement their own dialets.

Detail Radius Operations

- Before Client starts communicating with Radius server, it is required that shared secret must be shared between client and Server and Client must be configured to use Radius Server to get service.
- Once Client is configured properly then:

  Client starts with Access-Request.

  Server sends either Access-Accept, Access-Reject or Access-Challenge.

  Access-Accept keeps all required attribute to provide a service to user.

Detail Radius Operations – Continued

- Radius Codes are assigned as follows:
  - 1 -- Access-Request
  - 2 -- Access-Accept
  - 3 -- Access-Reject
  - 4 -- Accounting-Request
  - 5 -- Accounting-Response
  - 11 -- Access-Challenge
  - 12 -- Status-Server (experimental)
  - 13 -- Status-Client (experimental)
  - 255 -- Reserved

Detail Radius Operations – Continued

- Radius Packet format is has the following parts.
- Code: This is 1 octet long and identifies various types of packets
- Identifier: This is again 1 octet long and aids in matching responses with requests.
- Length: This is 2 octet long and specify the length of the packet including code, identifier, length and authenticator. (Min packet is 20 octet and max is 4096 octet).
- Authenticator: This is 16 octet long and filled up in case of some request and responses.

Detail Radius Operations – Continued

- List of Attributes: There will be a list of 63+ attributes and a radius attribute will also have defined format as below
  --Type: 1 octet, identifies various types of attribute.
  -- Length: 1 octet, length of the attribute including Type
  -- Value: 0 or more octets, contains information specific to attribute
- Examples of Radius Attributes list are User-Name, User-Password,NAS-IP-Address,NAS-Port,Service-Type, NAS-Identifier,Framed-Protocol,Vender-Specific,Calling-Station-ID ,Called-Station-Id

Other AAA services

- DIAMETER is a proposed next generation protocol specifically designed to meet the requirement of the IETF and TIA for 3GPP, 3GPP2 and IMS AAA requirements.

  Some of it's advantages is as below.

  - Better Proxying
  - Better Session Control
  - Better Security

- TACACS/TACAS+ is a Terminal Access Controller Access Control System is actually a remote authentication protocol that is used to communicate with an authentication server commonly used in unix networks.

Other AAA services – Continued

- LDAP – Lightweight Directory Access Protocol which is also used in unix systems authentication
- Kerberos – is the name of a computer network authentication protocol which allows individuals communication over a non-secure network to prove their identity to on another in a secure manner.

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 sse noc machine with the packages we need
- the packages are in:
  - Ftp ftp://ftpstud:afnog@noc.sse.ws.afnog.org
  - Cd /usr/ports/distfiles
- 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 but for this version of radius it has already been done at compiled time.

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`
- Freeradius should now be started

Configuring – Part 3

- 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.
- For monitoring purposes, we need the same secret on all the machine and that is “afnog”.

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_ssh
- `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.

Pam – Part 3

- Lets test the radius server via ssh to another machine in the class and see if it will request for the radius password which will enable us to log in.
  - Ssh remote_username@remote_ip_address
- In the authentication response, you should be presented with a radius Password request from the remote server

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