FreeRADIUS Install and Configuration

Frank A. Kuse

07/05/2012

Radius Introduction?

- Radius is a protocol that handle authentication but also authorization and accounting as well
- It Helps Centralized Authentication in larger networks
- Most AAA services uses RADIUS
- Reduces loads on front end servers

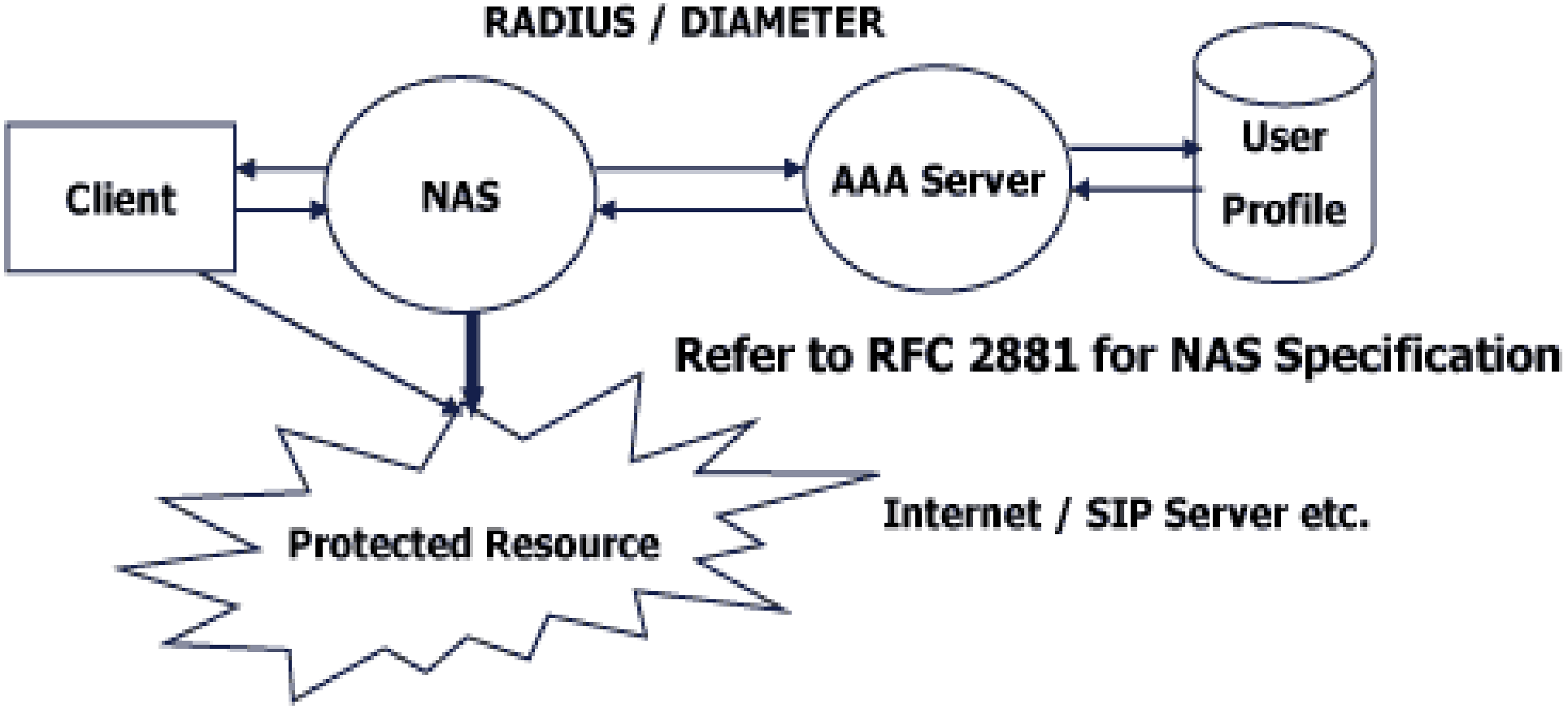
Authentication, Authorization, Accounting?

- Authentication can be posed as a question that is Who are you?
- Authorization can also be posed as a question that is What services am I allowed to give you?
- Accounting can be posed as a question that is what did you do with my services while you were using them?

What is RADIUS? – In Summary

- Authentication is a process of verifying a person's declared identity.
- Authorization is using a set of rules or other templates to decide what an authenticated user can do on a system
- Accounting is measuring and documenting the resources a user takes advantage of during access.
- RFC 2058, 2059 and 3865 contains more documentation on it. carnation is embodied in RFC 2865.

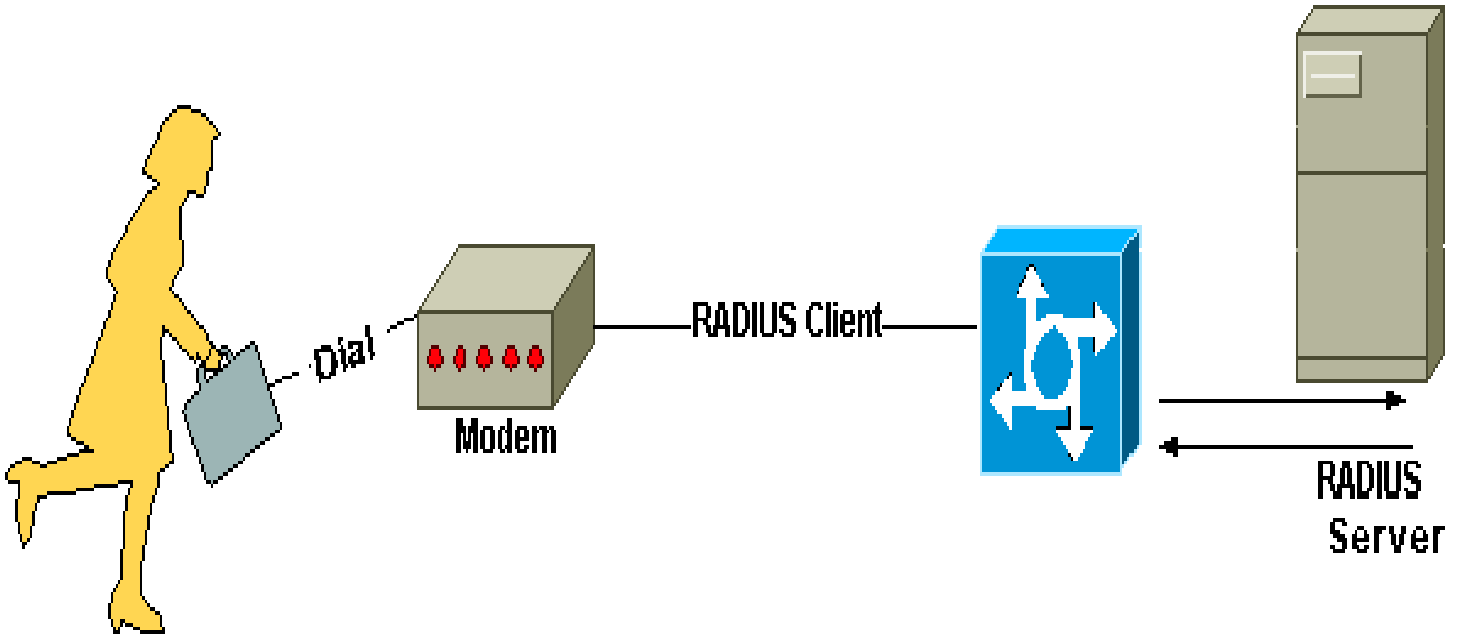
Basic Architecture of NAS/Radius/AAA



Basic Architecture for NAS/RADIUS/AAA

RADIUS Mechanism

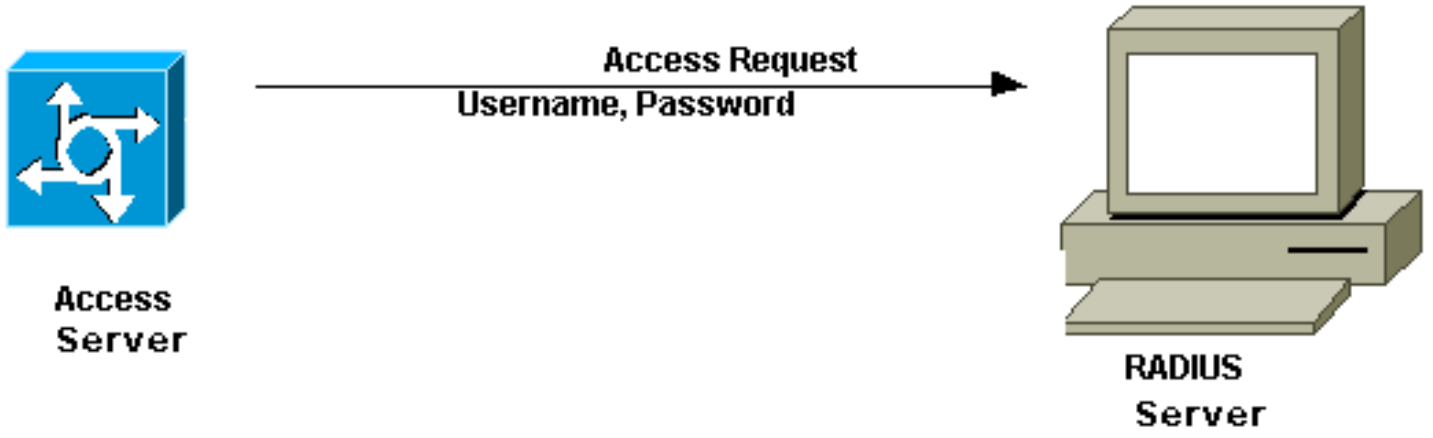
Radius in action



RADIUS Message

- Access Request

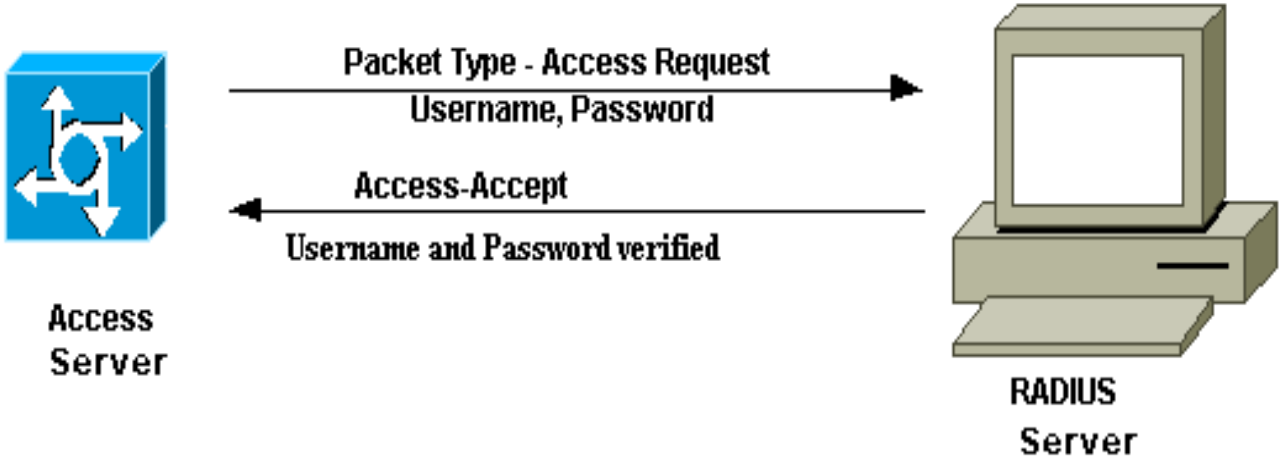
* Generated by the NAS (RADIUS client) towards the server to forward the request from or on behalf of a user.



RADIUS Message Continue

- Access Accept

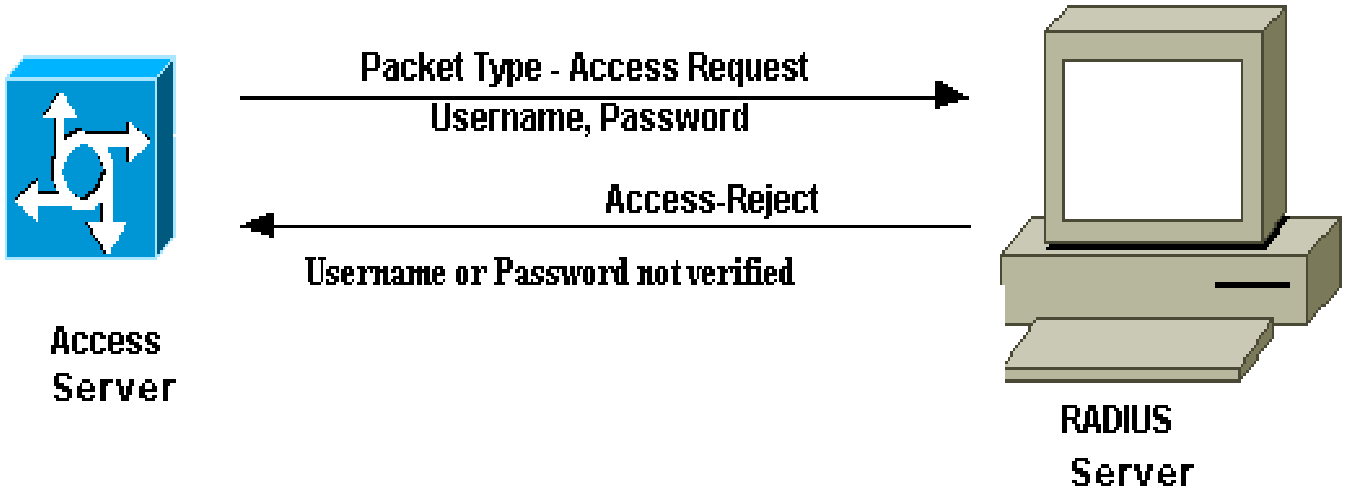
* This message is sent from the RADIUS server to the NAS to indicate a successful completion of the request



RADIUS Message continue

- Access Reject

* This message is sent by the server to indicate the rejection of a request



RADIUS Message Continue

- **Accounting request**
- Sent from the client to the accounting server to convey accounting information regarding the service provided to the user
- **Accounting response**

* Sent by the server to the client (NAS) to acknowledge and indicates the result of the performed accounting function by the server

Detail Radius Operations

- Radius Codes are assigned as follows:

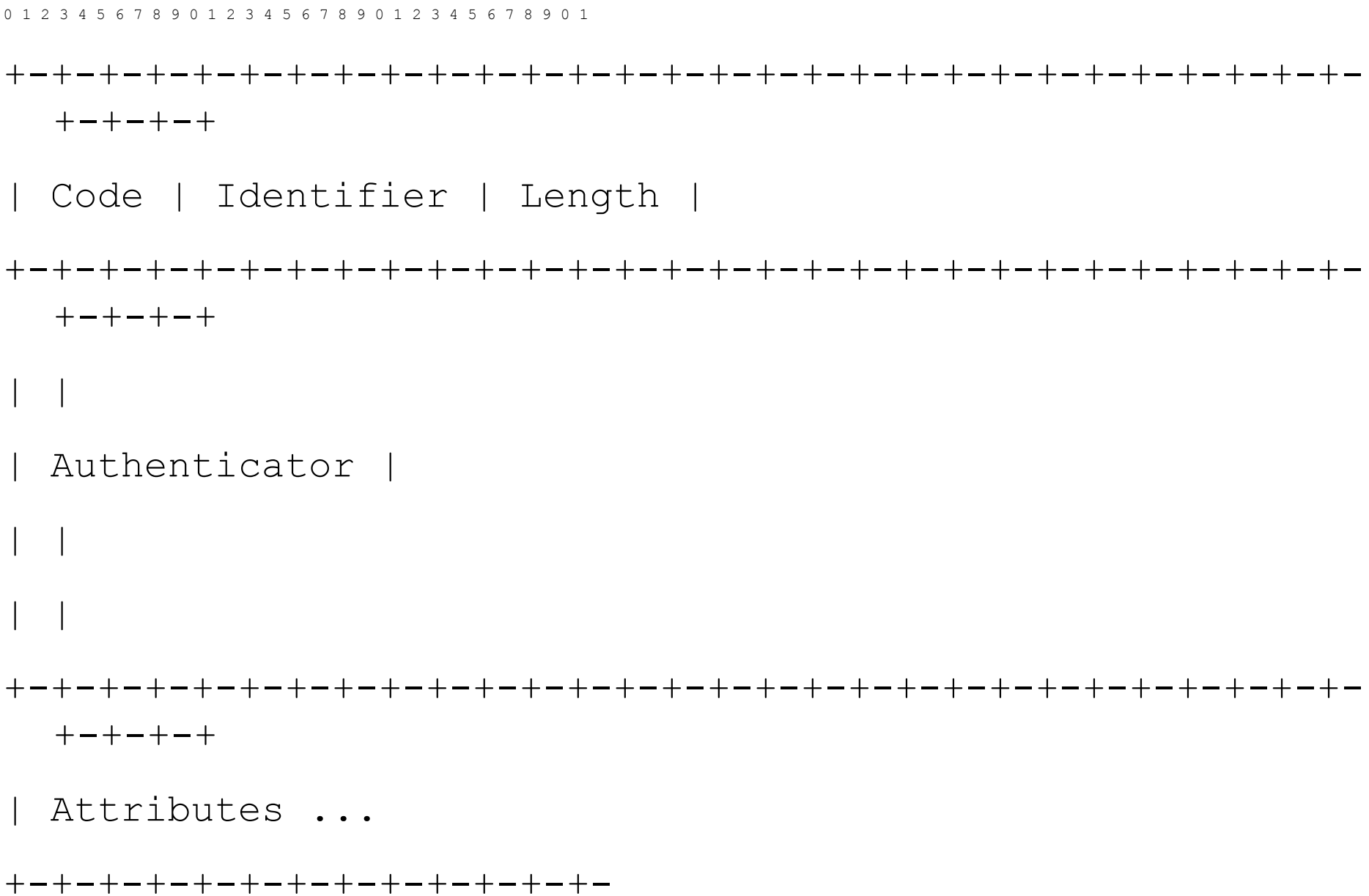
1 -- Access-Request

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



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.

Configuring User Information

The Radius users file is a flat text file on the Radius Server. The users file stores authentication and authorization information for all users authenticated with Radius . For each user, you must create an entry that consists of three parts: the username, a list of check items, and a list of reply items.

```
Franko          Password = 'testing12'
                Service-Type = Frame-User,
                Framed-protocol = PPP,
                Framed-IP-Address =
255.255.255.254
                Framed-IP-Netmask =
255.255.255.255
                Framed-Routing = None,
                Framed-MTU = 1500
```

Franko is the username and password testing12 is a check item and we have Service-Type as the first Reply Item and Framed-IP-Address being the second item.

*Username

The username is the first part of each user entry. Username consist of up to 63 printable,nonspace, ASCII characters.

*Check Items

Check items are listed on the first line of a user entry, separated by commas. For an access request to succeed, all check items in the user entry must be matched in the access request.

N.B: The line in the user entry that contains the username and check items must not exceed 255 characters.

*Reply Items

Reply items give the NAS information about the user's connection. Eg whether to use PPP or SLIP is used or whether the user's IP address is negotiated.

If all check items in the user entry are satisfied by the access-request, the radius server sends the reply items to the NAS to configure the connection.

* Password Locations

Use the Auth-Type check item to specify the type of authentication to use for a particular user. Auth-Type can be either of the following : Local , System or SecureID. If the check Item is omitted the user entry , Local is assumed.

* Local

To indicate that the user's password is stored in the Radius users file, use the Local Auth-Type. To set the user's password, use the Password check Item. An example line from a user entry is displayed below.

Franko Auth-Type = Local, Password = 'test123'

* System

To indicate that the user's password is stored in a system password file, use the System Auth-Type.

System can be a password file in unix such as /etc/passwd, /etc/shadow, a windows NT password database, or a password map in NIS or NIS+ . When the RADIUS server receives a username-password pair from the client, it queries the operating system to determine if there is a matching username-password pair.

Eg.

Franko Auth-Type = System

SecurID

The SecureID Auth-Type indicates that the user's password should be authenticated by a securID Server.

eg.

Franko Auth-Type=SecurID

To receive a passcode from SecurID, the Server software must be running on the same unix host as the radius server.

* Password Expiration Date

To disable logins after a particular date, complete the following steps:

1. Specify the date of expiration using the Expiration Check item. The date must be specified in “Mmm dd yyyy” format;

Eg.

Franko Password =”test12”, Expiration=“May 12 2009”

Edit the Password-Expiration and Password-Warning values in the dictionary to meet your security needs.

```
VALUE        Server-Config Password-Expiration  30
VALUE        Server-Config  Password-Warning          5
```

Configuring Client Information

Use the NAS-IP-Address check item to specify the IP address of a particular NAS. When this setting is used as a check item in a user entry, the user must attempt to start a connection on the specified NAS for the connection to succeed.

Use the NAS-Port check Item to specify a particular NAS port. To be successfully authenticated, the user must attempt to log in to this port.

Use the NAS-Port-Type check item to specify the type of port. Options for the NAS-Port-Type are as follows: Async, Sync, ISDN, ISDN-V120 or ISDN-V110.

Eg to display a user entry containing the NAS-IP-Address and NAS-Port-Type settings.

Franko Password = “test12”, NAS-IP-Address=192.168.2.2, NAS-Port-Type = ISDN

Service-Type = Framed-User,

Framed-Protocol = PPP

Configuring Reply Items

* Service Type

You must specify the type of service provided to the user, called the Service-Type, in each user entry. Service-Type must be set to one of the values show below.

Login-User → User connects via telnet, rlogin

Framed-User → User uses PPP or SLIP for connection

Outbound-User → User uses telnet for outbound connections.

You can get manual for other ones such attributes which are Callback-Login-User , Callback-Framed-User and Administrative-User and NAS-Prompt-User.

Eg of franko's Service-Type which is Framed-User

Franko Auth-Type = System
        Service-Type = Framed-User

. Framed Protocol

When the service-type is a Framed-User, you must include the Framed-Protocol reply item in the user entry to indicate whether PPP or SLIP is used.

Eg for a user franko is a PPP user. His full entry includes the following lines below:

```
Franko          Auth-Type = System
                Service-Type = Framed-User
                Framed-Protocol = PPP
```

Framed-Protocol can also be used as a reply item requiring PPP autodetection by the Portmaster

Franko Auth-Type = System, Framed-Protocol = PPP

Service-Type = Framed-User,

Framed-Protocol = PPP

N.B: To authenticate a user using PAP, set the Auth-Type to any of the following, Local, System or SecureID. To authenticate a user using CHAP, the Auth-Type must turn off PAP.

* Framed IP Address

Use the Framed-IP-Address reply item to specify the user's IP address.

When Framed-IP-Address is set to 255.255.255.255, the NAS negotiates the address with the end-node (dial-in user). When it is set to 255.255.255.254 (or omitted), the NAS assigns an IP address to the dial-in user from the assigned address pool.

* Framed IP Netmask

You must specify a netmask for a user using the Framed-IP-Netmask reply item. If this reply item is omitted, the default subnet mask of 255.255.255.255 is used.

* Framed Route

Use the Framed-Route reply item to add a route to NAS routing table when service to the user begins. Three pieces of information are required: the destination IP address, gateway IP address, and metric.

Eg. is as below

Franko Auth-type = System

Service-Type = Framed-User,

Framed-Protocol = PPP,

Framed-IP-Address = 196.200.219.4

Framed-Route = “196.200.219.0 196.200.219.4 1”

In this eg. 196.200.219.0 is the IP address of a destination network. 196.200.219.4 is the IP address of the gateway for this network.

N.B: If 0.0.0.0 is specified as the gateway IP address, the user's IP address is substituted for the gateway.

You can check your radius server specification documentation for other attributes such as Outbound-User, Callback-Login-User, Callback-Framed-User, Framed-Routing, Filter-Id, Framed-MTU,Session-Timeout,Idle-Timeout, port-limit etc.

Framed User Authenticating with CHAP

The NAS at 192.168.1.16 sends an Access-Request UDP packet to the RADIUS Server for a user named franko logging in on port 20 with PPP, authenticating using CHAP. The NAS sends along the Service-Type and Framed-Protocol attributes as a hint to the RADIUS server that this user is looking for PPP, although the NAS is not required to do so.

The RADIUS server authenticates franko, and sends an Access-Accept UDP packet to the NAS telling it to start PPP service and assign an address for the user out of its dynamic address pool.

Code = 2 (Access-Accept)

ID = 1 (same as in Access-Request)

Length = 56

Response Authenticator = {16-octet MD-5 checksum of the code (2), id (1), Length (56), the Request Authenticator from above, the attributes in this reply, and the shared secret}

Attributes:

Service-Type = Framed-User

Framed-Protocol = PPP

Framed-IP-Address = 255.255.255.254

Framed-Routing = None

Framed-MTU = 1500

Code = 1 (Access-Request)

ID = 1

Length = 71

Request Authenticator = {16 octet random number also used as CHAP challenge}

Attributes:

User-Name = "franko"

CHAP-Password = {1 octet CHAP ID followed by 16 octet CHAP response}

NAS-IP-Address = 192.168.1.16

NAS-Port = 20

Service-Type = Framed-User

Framed-Protocol = PPP

Eg below displays start and stop accounting record in a Radius Server detail file.

Tue May 12 14:12:14 2009

Acct-Session-Id = “25000005”

User-Name = “franko”

NAS-IP-Address = 196.200.219.2

NAS-Port = 1

NAS-Port-Type = Async

Acct-Status-Type = Start

Acct-Authentic = RADIUS

Service-Type = Login-User

Login-Service = Telnet

Login-IP-Host = 196.200.219.254

Acct-Delay-Time = 0

Timestamp = 838763356

Tue May 12 14:11:40 2009

Acct-Session-Id = “25000005”

User-Name = “franko”

NAS-IP-Address = 196.200.219.2

NAS-Port = 1

NAS-Port-Type = Async

Acct-Status-Type = Stop

Acct-Authentic = RADIUS

Service-Type = Login-User

Login-Service = Telnet

Login-IP-Host = 196.200.219.254

Acct-Delay-Time = 0

Timestamp = 838763378

The Acct-Status-Type attribute in the record indicates whether the record was sent when the connection began (Start) or when it ended (stop).

N.B: The Acct-Session-Id matches both the start and stop records indicating that it's the same session.

Plan of Attack

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

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.

Installing

- /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 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