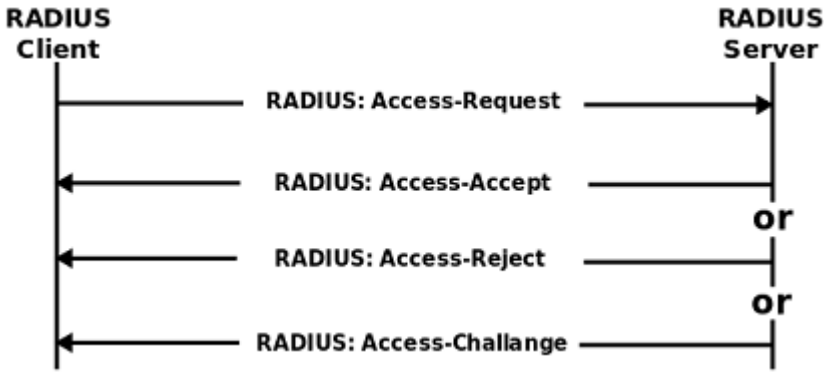
RADIUS Connectivity

Overview

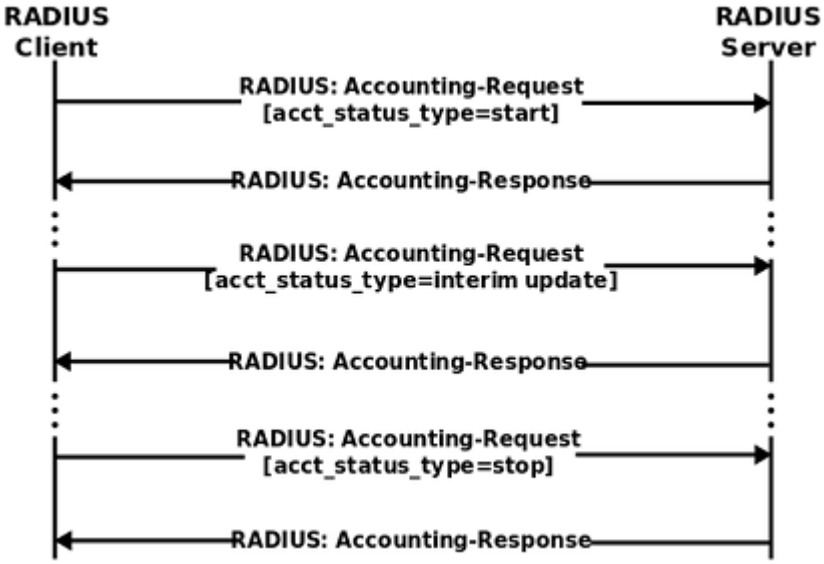
Remote Authentication Dial In User Service(**RADIUS**) is a networking protocol that provides centralized Authentication, Authorization, and Accounting (AAA) management for users that connect and use a network service.

NSC has built-in RADIUS client function, with which you can easily connect to your existing RADIUS services.(NSC doesn't provide RADIUS server function)

RADIUS Authentication and Authorization Flow:

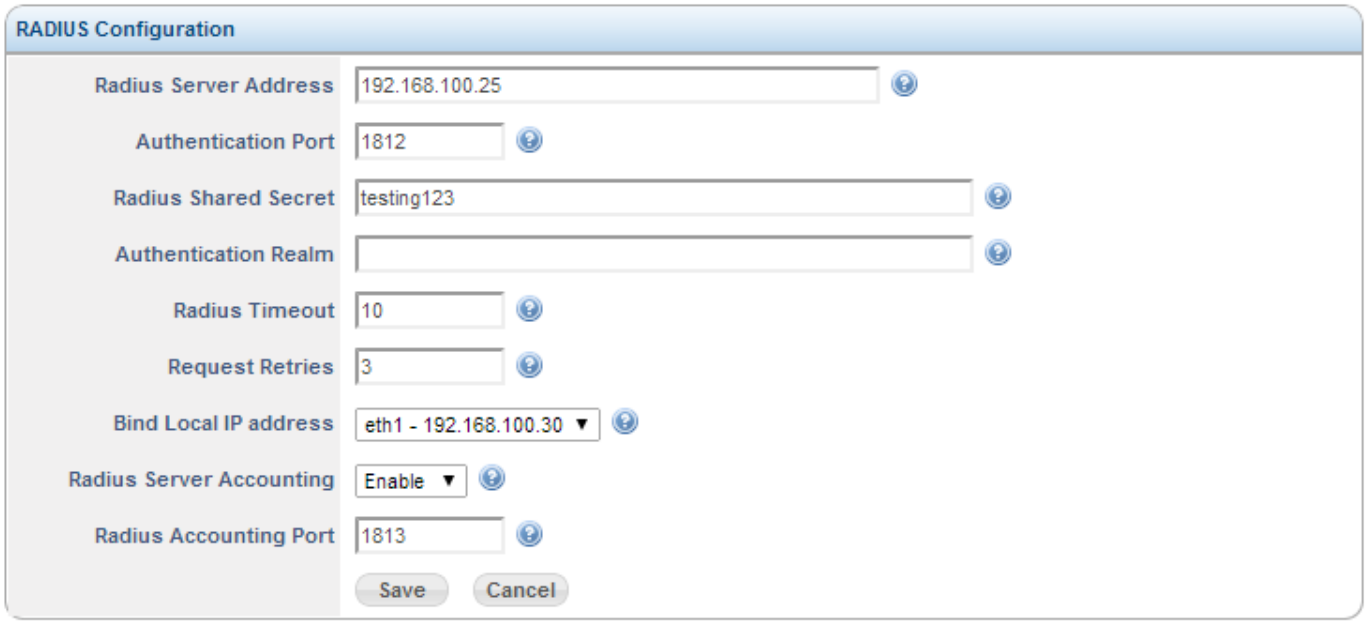


RADIUS Accounting Flow:



Configuration

Go to **"Configuration -> Signalling -> RADIUS"**, edit the **"RADIUS Configuration"**;



- You just need to simply configure the following items:

1. **"Radius Server Address"**: FQDN or IP address of RADIUS service (Note: we only support one RADIUS profile now, so only one address can be configured);
2. **"Authentication Port"** and **"Radius Accounting Port"**: Usually Authentication and Authorization work on port 1812, and Accounting works on port 1813;
3. **"Radius Shared Secret"**: This is the secret to protect the connection itself, please get it from RADIUS administrator;
4. **"Bind Local IP Address"**: From which local NIC to send out the RADIUS request;
5. When Authentication/Authorization service is needed:
    1. Make sure **"Radius Server Address":"Authentication Port"** is reachable;
    2. Add corresponding routing plan rules; Authentication/Authorization can only be used from within routing plan(see next chapter);
6. When Accounting service is needed:
    1. Set "**Radius Server Accounting**" to "Enable";
    2. Make sure **"Radius Server Address":"Radius Accounting Port"** is reachable, or the call will be blocked; if your RADIUS Accounting Server is still not ready yet, please set "**Radius Server Accounting**" to "Disable";
    3. Accounting start RADIUS message is sent to RADIUS server when call is connected, while Accounting Stop RADIUS message is sent when call is disconnected.

Authentication/Authorization from Routing Plan
Here below is an example of how to do Authentication/Authorization from within Routing Plan:

```
<extension name="unitest_rad-ANI-auth">
  <condition field="destination_number" expression="^(601)$">
    <action inline="true" application="set" data="CALLINGNUMBER=${caller_id_number}"/>
    <action inline="true" application="set" data="USERNAME=netborder"/>
    <action inline="true" application="set" data="PASSWD=sangoma"/>
    <action inline="true" application="set" data="DIALED_NUMBER=$1"/>
    <action application="sleep" data="2000"/>
    <action application="auth_function" data="in ${DIALED_NUMBER}, in ${USERNAME}, in ${PASSWD}, out AUTH_RESULT"/>
    <action application="log" data="INFO AUTH_RESULT=${AUTH_RESULT}"/>
  </condition>
</extension>
```

Out channel variable "AUTH_RESULT" has 2 possibilities:

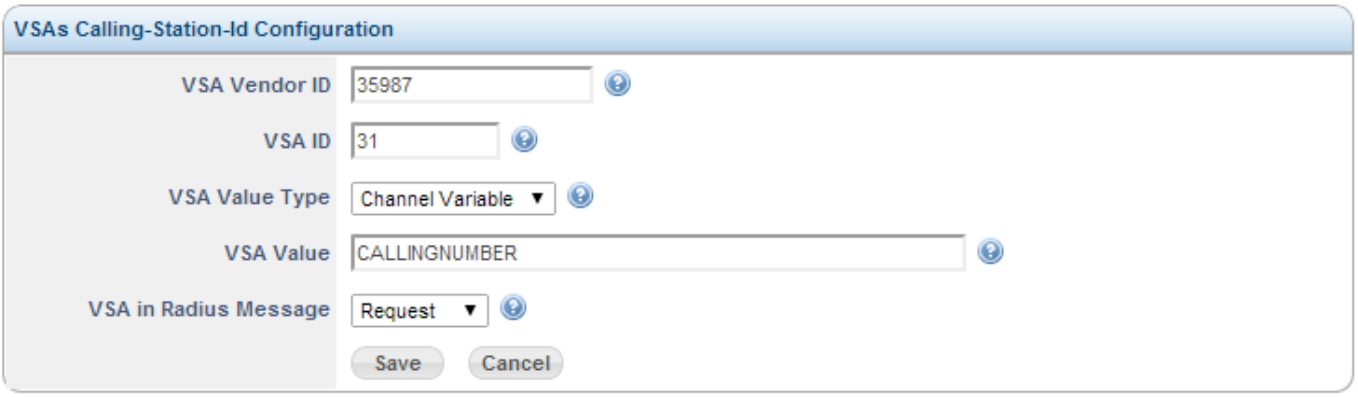
a. "OK": received Access Accpet
b. "NOK": received Access Reject

Adding VSAs for Authencation/Authorization

For Authentication/Authorization(not for Accounting), you can define your own VSAs (in the examples below, I use Sangoma's Vender id 35987 as example; of course you can use your own Vender id):

Define VSAs in the request
E.g. Besides the mandatory "DIALED_NUMBER", "USERNAME" and "PASSWD"; if you want to add your own VSA "Calling-Station-Id" (Vendor id: 35987, VSA id: 31, value is from channel variable "CALLINGNUMBER") in Access-Request, you can do the following:

- Go to **"Configuration -> Signalling -> RADIUS"**, add the **"RADIUS VSAs"** named as "Calling-Station_Id";



Define VSAs in the response and use from routing plan
E.g. In Access-Accpet there is a VSA id = 41, which includes the information for preferred language, you want to put it into channel variable "preferred_lang":

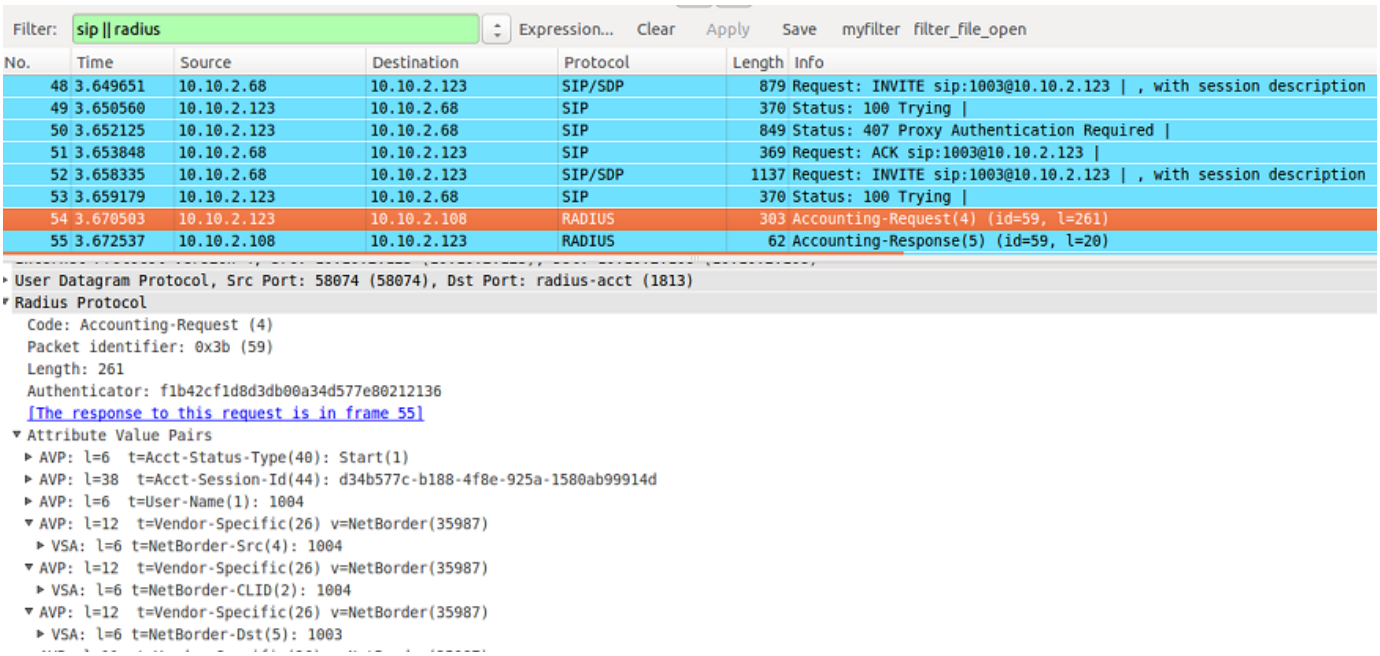
- VSA Vender ID : 35987
- VSA ID : 41
- VSA Value Type : keep it untouched, which is "Direct String Input"
- VSA Value : preferred_lang
- VSA in Radius Message : "Response"
- Here below an example of how to use it in routing plan:

```
<extension name="unitest_rad-ANI-auth">
  <condition field="destination_number" expression="^(601)$">
    ...
    <action application="auth_function" data="in ${DIALED_NUMBER}, in ${USERNAME}, in ${PASSWD}, out AUTH_RESULT"/>
    <action application="log" data="INFO Preferred language of user ${USERNAME} is ${preferred_lang}"/>
  </condition>
</extension>
```

Troubleshooting

- You can easily troubleshoot RADIUS message flow by filtering wireshark pcap trace by filter "radius".
- NSC only has RADIUS client function; for RADIUS server, you can either use your existing RADIUS server, or download and install the the great open source FreeRadius from www.freeradius.org

Here below is the screen capture of one RADIUS Accounting pcap trace:



- VSAs for NetBorder (Vendor ID: 35987) can be found in file dictionary.sangoma
- If VSAs in RADIUS message can not be decoded correctly, maybe it is because that your wireshark does not have the correct radius dictionary, then please do the following:
    - Open Wireshark, go to **"Help -> About Wireshark -> Folders"**, locate where the dictionary.sangma should be copied to (there is a radius sub-folder which contains a bunch of dictionary.* files);
    - Download the above dictionary.sangoma file, make sure the file name is dictionary.sangoma, and then copy into the radius sub-folder
    - Edit radius/dictionary file, add one line "$INCLUDE dictionary.sangoma"

- If your customerized VSA cannot be recognized by wireshark, just simply edit dictionary.sangoma to add the attribute