Authentification Radius sur switch Cisco

>>> Debian 8.7 & Freeradius 2.2.x & Cisco IOS 15.0

Description :

Le but de cette article est de mettre en place un serveur freeradius. Dans un deuxième, temps cette article abordera comment configurer un switch Cisco en IOS 15.0 afin de s'authentifier via le serveur freeradius.

Authentification Radius sur switch Cisco

>>> Debian 8.7 & Freeradius 2.2.x & Cisco IOS 15.0

Sommaire :

I) Introduction

freeradius est un serveur Radius libre permettant de s'authentifier. Le protocole radius permet de se connecter via un échange de paquets UDP, généralement sur le port 1812. Radius intègre également un module d'accounting, permettant par exemple de la facturation. Radius permet de s'authentifier via divers moyens comme une authentification en clair, par adresse MAC, via base MySQL/PgSQL, protocole MSCHAPv1 et MSCHAPv2 ou encore annuaire LDAP.

Radius gère également le 802.1X avec l'authentification via tunnel EAP (PEAP/TLS/TTLS).

freeradius s'appuie sur un système de modules qui sont activés/désactivés lors des phases d'autorisation et d'authentification. Le service peut gérer différents serveurs virtuels, afin de pouvoir gérer plusieurs types d'authentifications conflictuelles, des tunnels internes ou encore des requêtes de proxy radius (en cas de chaînage de différents serveurs radius).

La phase d'autorisation (authorization) définit les modules qui vont intervenir pour autoriser l'utilisateur à utiliser la connexion. La phase d'authentification va s'appuyer sur différents modules pour authentifier l'utilisateur, via son mot de passe ou son adresse MAC par exemple.

AAA (Authentication, Authorization and Accounting) est un protocole qui permet de gérer :

- **Authentication** : Authentification consiste à déterminer si l'utilisateur ou l'équipement est bien celui qu'il prêtant être, cela se fait grâce à une authentification nom d'utilisateur/ mot de passe, ou grâce à un certificat.
- **Authorization** : Autorisation consiste à déterminer les droits de l'utilisateur sur les différentes ressources.
- **Accounting** : Compte permet de garder des informations sur l'utilisation des ressources par

l'utilisateur.

Voici les informations de notre maquette :

- Le serveur radius :
  - Hostname : Deb-Idum-LAB4
  - Adresse IP : 172.16.1.17/24
  - Port d'auth du service Radius : 1812
  - Port accounting du service Radius : 1813
- Le switch :
  - Hostname : sw-Idum-LAB
  - Adresse IP : 172.16.1.253/24
  - Type de switch : WS-C2960-8TC-L
  - Version IOS : 15.0(2)SE4
  - Key radius : bonjour

II) Installation de freeradius

- Commencez par installer le paquet **freeradius**.

```
aptitude install freeradius
```

III) Configuration

Allez dans le répertoire **"/etc/freeradius/"** pour trouver l'ensemble des fichiers de configuration. Sous **freeradius** 2.x (celle qui sera utilisée tout au long de ce tutoriel) la configuration est répartie en dossiers de manière à simplifier l'administration.

Les différents dossiers sont :

- **certs** : les certificats nécessaires pour le 802.1X
- **modules** : l'ensemble des modules avec chacun un fichier de configuration séparée
- **sites-available/sites-enabled** : a l'instar d'Apache, les différents serveurs virtuels radius
- **sql** (si vous avez installé l'un des modules SQL) : fichiers de configuration des différentes bases SQL

La configuration principale est lue à partir du fichier **radiusd.conf**, contenant la définition et les liens vers les modules et les serveurs virtuels.

1) Configuration de radius.conf

- Éditez le fichier **"radius.conf"**.

```
vim /etc/freeradius/radius.conf
```

- Recherchez et modifiez la ligne **"auth_badpass"** et **"auth_goodpass"**, afin de mettre la valeur à **"yes"**.

```
auth_badpass = yes
auth_goodpass = yes
```

- Pour accélérer les réponses de votre serveur radius, désactiver la résolution de nom. (Pas obligatoire)

```
hostname_lookups = no
```

2) Configuration de clients.conf

Les clients radius ne sont pas les utilisateurs, mais les services/binaires qui vont permettre d'authentifier l'utilisateur, comme par exemple : le client d'une borne Wifi, le mod_auth_xradius d'Apache et bien d'autres encore. Ouvrez le fichier clients.conf.

```
vim /etc/freeradius/clients.conf
```

Ce fichier contient les différents clients, identifiés par leur adresse IP ou adresse réseau. Chaque client utilise un secret afin de savoir s'il peut utiliser le serveur radius, et utilise un NASType, option bien souvent inutile ou obsolète. Il est conseillé de la positionner à other, par exemple si vous utilisez des bornes WiFi CISCO.

- Modifiez la clef du serveur radius :

```
client localhost {
secret = coucou
```

Note sur la sécurité : La sécurité du protocole RADIUS dépend COMPLÈTEMENT de cette clef ! Nous vous recommandons d'utiliser une clef complexe composée de :

- Lettres majuscule
- Lettres minuscules
- Nombres
- Symboles

- Ajoutez les lignes ci-dessous pour créer le client **"switch"**.

```
client 172.16.1.253 {
secret = bonjour
shortname = sw-idum-lab
nastype = other
}
```

3) Configuration du fichier users

Dans la forme la plus simple de **freeradius**, vous pouvez utiliser le fichier **"users"** pour déclarer vos utilisateurs (login/password) qui seront autorisés à s'authentifier.
freeradius propose d'autre solution, comme évoqué dans l'introduction :

- Base de données SQL
- Serveur LDAP

- Éditez le fichier **"users"**.

```
vim /etc/freeradius/users
```

- Nous allons créer deux utilisateurs :

- **nsalmon** avec le mot de passe **"hello"** et les droits de niveau 15
- **test-radius** avec le mot de passe **"hello"** et les droits de niveau 2

- Ajoutez les lignes suivantes :

```
nsalmon Cleartext-Password := "hello"
Service-Type = NAS-Prompt-User,
Cisco-AVPair = "shell:priv-lvl=15"

test-radius Cleartext-Password := "hello"
Service-Type = NAS-Prompt-User,
Cisco-AVPair = "shell:priv-lvl=5"
```

4) Derniere étape

- Rechargez le service **freeradius** afin de prendre en compte les modifications.

```
service freeradius reload
```

IV) Configuration du switch

1) Configuration de base du switch

- Définissez un hostname :

```
conf t
hostname sw-idum-lab
```

- Définissez un domaine et le serveur de noms de votre réseau. (Dans notre LAB notre serveur radius est aussi serveur DNS)

```
ip domain-name idum.eu
ip name-server 172.16.1.17
```

- Définissez l'adresse IP de votre switch :

```
interface vlan 1
ip address 172.16.1.253 255.255.255.0
exit
ip default-gateway 172.16.1.254
```

- Générer la crypto key pour le SSH (taille 2048) :

```
crypto key generate rsa
```

- Vous devez obtenir ceci :

```
The name for the keys will be: sw-idum-lab.idum.eu
Choose the size of the key modulus in the range of 360 to 4096 for your
```

```
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.

How many bits in the modulus [512]: 2048
% Generating 2048 bit RSA keys, keys will be non-exportable...
```

- Activez la version 2 de SSH :

```
ip ssh vesion 2
```

- Autoriser les connexions SSH :

```
line vty 0 15
transport input ssh
exit
```

- Définissez un utilisateur local, avec les paramètres ci-dessous :

- nom : admin
- droit : 15
- mot de passe : guten_tag

```
username admin privilege 15 secret guten_tag
```

- Création d'un privilège de niveau 5, autorisant seulement les commandes **"show"** ainsi que la commande **"show running-config"**.

```
privilege exec all level 5 show running-config
privilege exec level 5 show
```

2) Déclaration du serveur radius

- Tapez les lignes suivantes, pour définir le serveur Radius :

```
radius server deb-idum-lab4
address ipv4 172.16.1.17 auth-port 1812 acct-port 1813
key bonjour
exit
```

- Si vous avez plusieurs interfaces IP de configurée sur le switch, vous devez définir l'interface source pour les requêtes radius.

```
ip radius source-interface vlan 1
```

- Rajoutez la ligne suivante, afin d'éviter des erreurs dans les logs :

```
radius-server attribute 6 on-for-login-auth
```

3) Configuration aaa

a) 1ére méthode

Cette première méthode est la plus simple. Nous modifions le type d'authentification par défaut afin qu'il interroge en premier la base locale puis le serveur radius.

- Définissez un nouveau **model aaa**.

```
aaa new-model
```

- Création de l'ensemble **aaa authentication** :

```
aaa authentication login default local group radius
```

- Création de l'ensemble **aaa authorization** :

```
aaa authorization exec default local group radius
```

- Lancez une nouvelle connexion SSH vers le switch, en essayant de vous authentifier avec les utilisateurs radius.
- Lancez une nouvelle connexion SSH vers le switch, en essayant de vous authentifier avec l'utilisateur local.

b) Deuxième méthode

Cette deuxième méthode, permet de déclarer plusieurs scénario d'authentification. Nous voulons par exemple que les connexions en console sur le switch, utilisent seulement les utilisateurs locaux du switch pour s'authentifier. Et nous voulons que les connexions en SSH utilisent les utilisateurs locaux **et** les utilisateurs radius.

- Définissez un nouveau model aaa

```
aaa new-model
```

- Même chose que dans la première méthode, création de l'ensemble **aaa authentication**. Mais cette fois nous remplaçons le terme **Default** par le terme **AuthList1** :

```
aaa authentication login AuthList1 local group radius
```

- Même chose que dans la première méthode, création de l'ensemble **aaa authorization**. Mais cette fois nous remplaçons le terme **Default** par le terme **AuthList1** :

```
aaa authorization exec AuthList1 local group radius
```

- Définissez un ensenble **aaa authentication default** comme ceci :

```
aaa authentication login default local
```

- Création de l'ensemble **aaa authorization default** comme ceci :

```
aaa authorization exec default local
```

- Ensuite dans les interfaces **line vty**, saisissez la liste **AuthList1** :

```
line vty 0 15
login authentication AuthList1
authorization exec AuthList1
exit
line con 0
login authentication default
authorization exec default
```

- Lancez une nouvelle connexion SSH vers le switch, en essayant de vous authentifier avec les utilisateurs radius.
- Lancez une nouvelle connexion SSH vers le switch, en essayant de vous authentifier avec l'utilisateur local.

V) Authentification par adresse MAC

Nous allons maintenant voir comment authentifier une machine (PC, imprimante, ...) via son adresse MAC. Puis dans un deuxième temps, on fera en sorte qu'une fois authentifiée notre machine soit dans un Vlan particulier.

1) Modification du fichier users

- Editez le fichier freeradius **"users"**.

```
vim /etc/freeradius/users
```

- Ajoutez un nouvel utilisateur avec les paramètres suivants :

- nom : adresse_mac_du_PC
- password : adresse_mac_du_PC

```
001c23582eae Cleartext-Password := "001c23582eae"
Service-Type = Framed-User,
```

- Puis rechargez la configuration de **freeradius**.

```
service freeradius reload
```

2) Modification du switch

- Ajoutez les lignes suivantes :

```
aaa authentication dot1x default group radius
```

```
aaa authorization network default group radius local
```

- Puis configurez l'interface que vous voulez sécuriser :

```
interface FastEthernet0/1
switchport mode access
switchport nonegotiate
authentication port-control auto
mab
spanning-tree portfast
```

- Vous n'avez plus qu'à connecter votre PC.
- Vous pouvez observer que l'authentification fonctionne bien grâce à la commande :

```
debug radius
```

3) Authentification MAC et VLAN

Comme évoqué ci-dessus, nous allons faire en sorte qu'une fois authentifiée notre machine soit dans le vlan 10.

- Modifiez le fichier **users**.

```
001c23582eae Cleartext-Password := "001c23582eae"
Service-Type = Framed-User,
Tunnel-type = VLAN,
Tunnel-Medium-Type = 6,
Tunnel-private-Group-ID = 10
```

- Rechargez la configuration de **freeradius**.

```
service freeradius reload
```

- Puis n'oubliez de créer le Vlan sur le switch.

```
vlan 10
name test-radius
```

- Avant de connecter votre machine, tapez la commande **"show vlan"** afin de vérifier que l'interface fa0/1 est bien dans le Vlan par défaut :

```
sw-idum-lab#sh vlan

VLAN Name Status Ports
---- -------------------------------- --------- -------------------------------
1 default active Fa0/1, Fa0/2, Fa0/3, Fa0/4, Fa0/5, Fa0/6, Fa0/7, Fa0/8, Gi0/1
10 test active
1002 fddi-default act/unsup
1003 token-ring-default act/unsup
1004 fddinet-default act/unsup
1005 trnet-default act/unsup

VLAN Type SAID MTU Parent RingNo BridgeNo Stp BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
```

```
1 enet 100001 1500 - - - - - 0 0
10 enet 100010 1500 - - - - - 0 0
1002 fddi 101002 1500 - - - - - 0 0
1003 tr 101003 1500 - - - - - 0 0
1004 fdnet 101004 1500 - - - ieee - 0 0
1005 trnet 101005 1500 - - - ibm - 0 0

Remote SPAN VLANs
------------------------------------------------------------------------------


Primary Secondary Type Ports
------- --------- ----------------- ------------------------------------------
```

- Connectez votre machine sur le port fa0/1, et retapez la commande afin de confirmer qu'elle est bien dans le vlan 10.

```
sw-idum-lab#sh vlan

VLAN Name Status Ports
---- -------------------------------- --------- -------------------------------
1 default active Fa0/2, Fa0/3, Fa0/4, Fa0/5, Fa0/6, Fa0/7, Fa0/8, Gi0/1
10 test active Fa0/1
1002 fddi-default act/unsup
1003 token-ring-default act/unsup
1004 fddinet-default act/unsup
1005 trnet-default act/unsup

VLAN Type SAID MTU Parent RingNo BridgeNo Stp BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1 enet 100001 1500 - - - - - 0 0
10 enet 100010 1500 - - - - - 0 0
1002 fddi 101002 1500 - - - - - 0 0
1003 tr 101003 1500 - - - - - 0 0
1004 fdnet 101004 1500 - - - ieee - 0 0
1005 trnet 101005 1500 - - - ibm - 0 0

Remote SPAN VLANs
------------------------------------------------------------------------------


Primary Secondary Type Ports
------- --------- ----------------- ------------------------------------------
```