Cisco : Authentification radius

>>> Cisco 2960 IOS 15.0

Description :

Le but de cet article est de mettre en place l'authentification radius pour authentifier les utilisateurs qui administre le switch en SSH ou en console.

Cisco : Authentification radius

>>> Cisco 2960 IOS 15.0

Sommaire :

I) Introduction

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
- Paramètres du fichier **client.conf** :

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

II) 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
```

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 modèle 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.