EC2LT
ECOLE CENTRALE DES LOGICIELS LIBRES ET DE TELECOMMUNICATIONS

Un peuple, un but, une foi

Ecole Centrale des Logiciels Libres et de Télécommunications de Dakar

Processus d'authentification SSH avec serveur RADIUS sur un routeur Cisco

Nom : Koty

Prénom : Mahamat Koty

Filière : Télécommunications et Réseaux,

Option Cybersécurité et DevOps.

2024 - 2025

Table des matières

Introduction

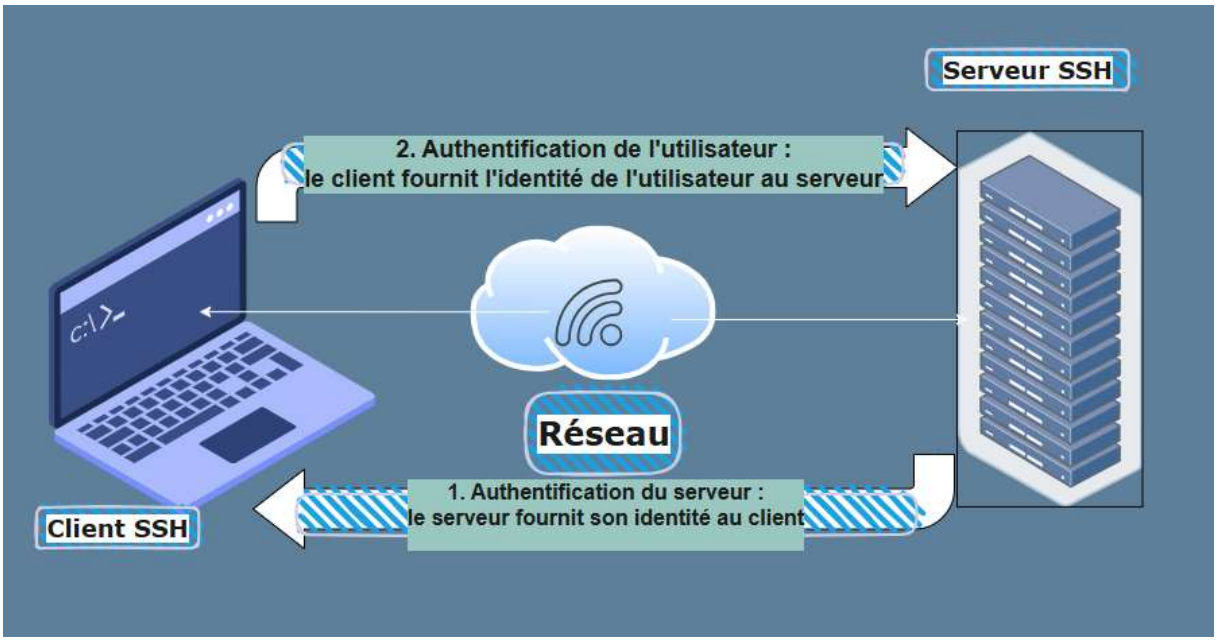
L'authentification centralisée est un élément crucial dans la sécurisation des infrastructures réseau. Ce guide présente l'implémentation d'une solution d'authentification SSH sur un routeur Cisco en utilisant un serveur RADIUS (Remote Authentication Dial-In User Service). Cette architecture permet de centraliser la gestion des utilisateurs et d'améliorer la sécurité des accès aux équipements réseau.

1 Qu’est-ce que le protocole SSH ?

Secure Shell, souvent abrégé en SSH, est un protocole réseau cryptographique. Il est conçu pour la communication sécurisée de données, la connexion à la ligne de commande à distance et l’exécution de commandes à distance. Introduit pour remédier aux failles de sécurité des protocoles antérieurs tels que Telnet, le SSH fournit un canal chiffré pour la communication client-serveur, garantissant la confidentialité et l’intégrité des données.

1.1 Comment fonctionne le protocole SSH ?

Le fonctionnement du protocole SSH repose sur une paire de Clés SSH – une clé publique et une clé privée. Lorsqu’une connexion est établie, ces clés fonctionnent ensemble pour établir un lien chiffré entre le client et le serveur. La clé publique est librement distribuée et peut être utilisée par n’importe qui pour chiffrer des messages. Toutefois, le déchiffrement de ces messages ne peut se faire qu’à l’aide de la clé privée correspondante, qui reste stockée en toute sécurité sur le serveur.



1.2 Capacités du protocole SSH

- Chiffrement des données

Le protocole SSH chiffre les données afin d'empêcher tout accès non autorisé pendant la transmission. Ce chiffrement garantit que les informations sensibles restent confidentielles et à l'abri des regards indiscrets.

- Authentification

Le SSH utilise la cryptographie à clé publique pour l'authentification. Ce mécanisme vérifie l'identité de l'utilisateur ou de l'appareil, ajoutant ainsi une couche de sécurité supplémentaire.

- Exécution de commande

Le protocole SSH permet d'exécuter des commandes sur un serveur distant. Cette capacité permet aux administrateurs de gérer les systèmes à partir de n'importe quel endroit.

- Partage de fichiers

Grâce à des protocoles tels que SFTP (SSH File Transfer Protocol), le SSH permet le transfert sécurisé de fichiers entre différents hôtes.

- Transfert de port

Le SSH peut transférer le trafic d'un port à l'autre, ce qui augmente la flexibilité dans la gestion et l'acheminement du trafic réseau.

1.3 utilisations principales du protocole SSH

- Administration du système

Le SSH est largement utilisé dans l'administration des systèmes. Sa capacité à exécuter des commandes à distance permet aux administrateurs d'accéder à des systèmes situés n'importe où dans le monde.

- Transferts de fichiers sécurisés

Grâce à son rôle intégral dans SFTP, le protocole SSH garantit des transferts de fichiers sécurisés. Il chiffre les fichiers pendant le transfert, protégeant ainsi les données sensibles contre l'interception.

- Gestion du réseau

Le SSH facilite la gestion du réseau grâce à la redirection des ports. Cette fonction permet de réacheminer le trafic, ce qui permet une gestion efficace du trafic sur le réseau.

- Sauvegarde et synchronisation

Le protocole SSH peut être utilisé pour sauvegarder et synchroniser en toute sécurité des fichiers sur plusieurs systèmes. Cette utilisation garantit la redondance et la disponibilité des données.

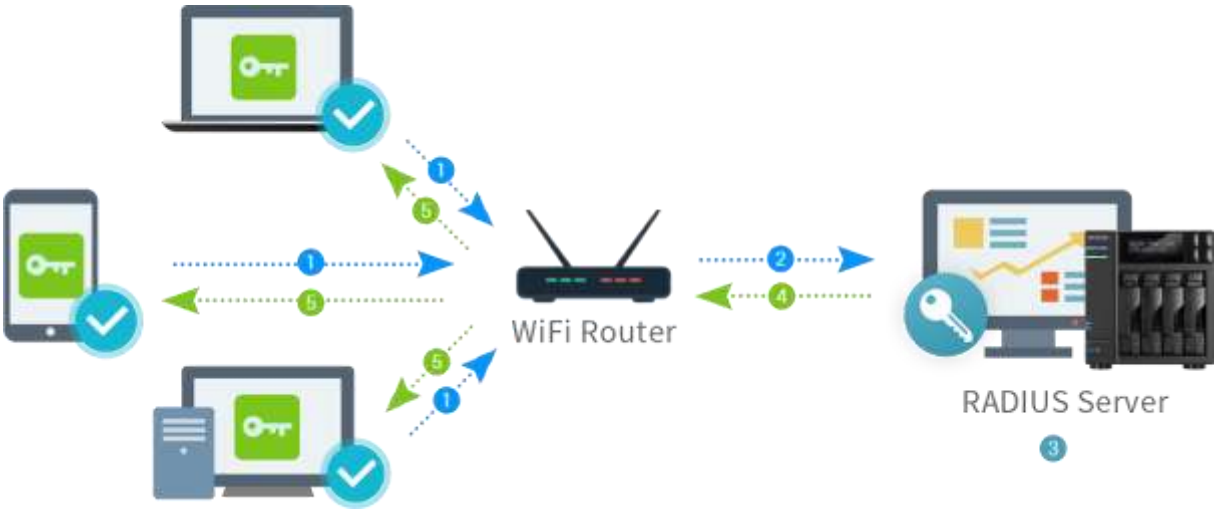
- L’accès à distance

Le SSH fournit un accès à distance sécurisé aux appareils. Cette capacité est vitale pour les professionnels de l’informatique qui doivent gérer des systèmes efficacement et à distance.

2 Qu’est-ce que RADIUS ?

RADIUS est un protocole réseau qui fournit une authentification, une autorisation et une comptabilité centralisées (AAA) pour les utilisateurs qui se connectent à un service réseau et l'utilisent. Il a été développé à l'origine par Livingston Enterprises pour être utilisé avec des réseaux commutés, mais il est depuis devenu une norme largement adoptée pour tous les types de réseaux.

RADIUS est utilisé par les administrateurs réseau pour gérer l'accès aux ressources réseau. Lorsqu'un utilisateur tente de se connecter à un réseau, le serveur RADIUS est contacté pour vérifier les informations d'identification de l'utilisateur. Si les informations d'identification sont valides, le serveur RADIUS envoie un message d'approbation d'accès au serveur d'accès réseau, qui accorde l'accès à l'utilisateur.



2.1 Comment fonctionne RADIUS ?

RADIUS fonctionne en authentifiant les utilisateurs qui tentent d'accéder à une ressource réseau. Pour ce faire, il utilise un nom d'utilisateur et un mot de passe, qui sont transmis au serveur RADIUS pour validation. Le serveur RADIUS renvoie ensuite une réponse au serveur d'accès au réseau, qui accorde ou refuse l'accès.

RADIUS fournit également des services d'autorisation et de comptabilité. L'autorisation consiste à déterminer les ressources auxquelles un utilisateur est autorisé à accéder, en fonction de son profil d'utilisateur. La comptabilité implique le suivi des ressources auxquelles un utilisateur a accédé, y compris le temps qu'il a passé sur le réseau et la quantité de données qu'il a transmises.

2.2 Méthodes d'authentification RADIUS

RADIUS prend en charge diverses méthodes d'authentification, notamment :

Authentification par mot de passe : Il s'agit de la méthode d'authentification la plus courante, où les utilisateurs entrent un nom d'utilisateur et un mot de passe pour accéder au réseau.

Authentification basée sur certificat : Cette méthode utilise des certificats numériques pour authentifier les utilisateurs.

Authentification basée sur des jetons : L'authentification basée sur les jetons utilise un périphérique physique, tel qu'une carte à puce ou un token de sécurité, pour authentifier les utilisateurs.

2.3 Avantages de l'utilisation de RADIUS

- Sécurité améliorée

RADIUS fournit des services d'authentification et d'autorisation forts, ce qui permet de s'assurer que seuls les utilisateurs autorisés peuvent accéder aux ressources réseau. Cela permet d'empêcher tout accès non autorisé et de réduire le risque de violation de données.

- Gestion centralisée

RADIUS permet une gestion centralisée des comptes utilisateurs, ce qui facilite la gestion de l'accès aux ressources réseau. Cela permet de gagner du temps et de réduire le risque d'erreurs.

- Évolutivité

RADIUS est hautement évolutif, ce qui signifie qu'il peut être utilisé pour gérer l'accès à des réseaux de toute taille. Cela en fait une solution idéale pour les petites et les grandes organisations.

- Compatibilité

RADIUS est une norme largement adoptée, ce qui signifie qu'elle est compatible avec une large gamme d'équipements réseau, notamment des serveurs, des routeurs et des points d'accès sans fil.

Mise en œuvre

3 Architecture

L'architecture se compose de trois éléments principaux :

Adresse Réseau : 192.168.122.0/24

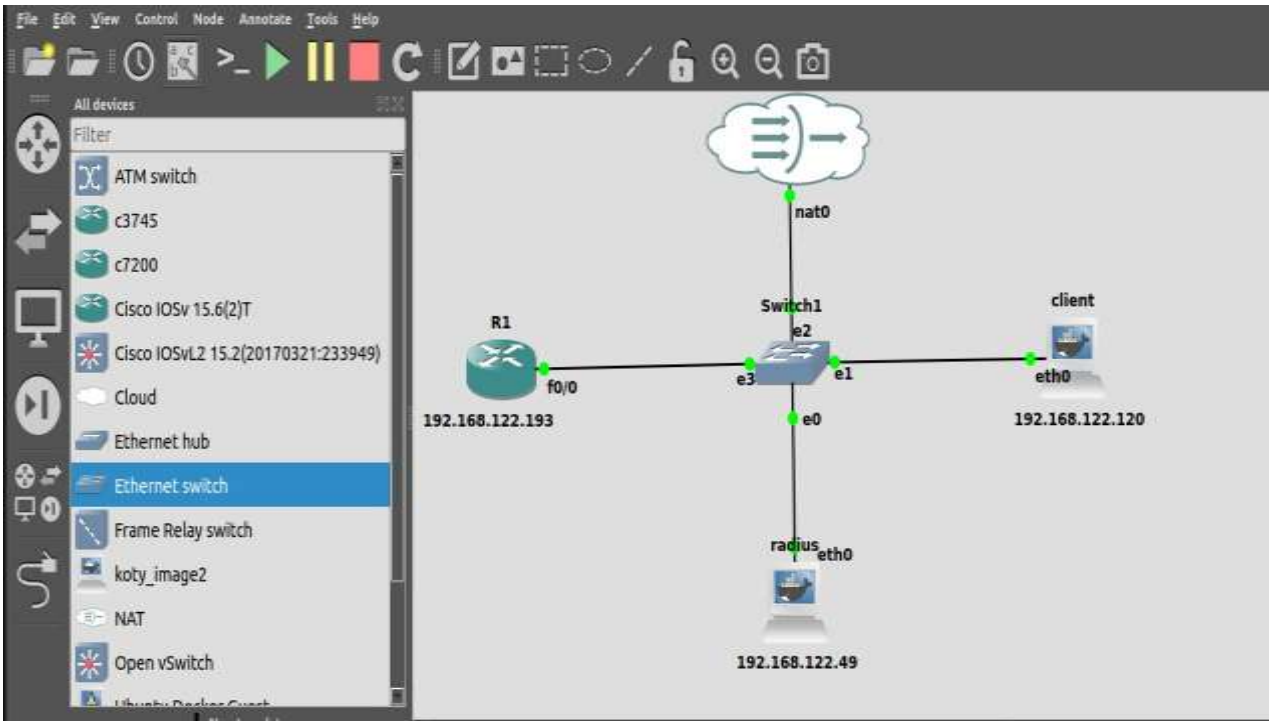
Client : Machine utilisateur (192.168.122.120)

Routeur Cisco : Équipement réseau cible (R1) 192.168.122.193

Serveur RADIUS : Serveur d'authentification FreeRADIUS (192.168.122.49)

Le flux d'authentification suit ce processus :

1. L'utilisateur initie une connexion SSH vers le routeur
2. Le routeur interroge le serveur RADIUS pour vérifier les identifiants
3. Le serveur RADIUS valide ou refuse l'authentification
4. L'accès SSH est accordé ou refusé selon la réponse du serveur



4 Configuration réseau du routeur

Attribution d'une adresse IP par DHCP

R1#conf t

R1(config)#int fa 0/0

R1(config-if)#ip add dhcp

R1(config-if)#no sh

R1(config-if)#

Cette configuration permet au routeur d'obtenir automatiquement une adresse IP dans le réseau 192.168.122.0/24.

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int fa 0/0
R1(config-if)#ip add dhcp
R1(config-if)#no sh
R1(config-if)#
*Apr 13 15:38:44.767: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Apr 13 15:38:45.767: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0
, changed state to up
R1(config-if)#
*Apr 13 15:38:58.367: %DHCP-6-ADDRESS_ASSIGN: Interface FastEthernet0/0 assigned DHCP
 address 192.168.122.193, mask 255.255.255.0, hostname R1

R1(config-if)#
```

5 Installation et configuration du serveur RADIUS

5.1 Attribution d'une adresse IP par DHCP au Serveur RADIUS

- Serveur RADIUS

```
# DHCP config for eth0
auto eth0
iface eth0 inet dhcp
#           hostname koty_image2-2
```

Verification : 192.168.122.49/24

```
root@radius:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.122.49  netmask 255.255.255.0  broadcast 192.168.122.255
        ether 02:42:3b:5b:33:00  txqueuelen 1000  (Ethernet)
        RX packets 85  bytes 9479 (9.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 2016 (2.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

5.2 Attribution d'une adresse IP par DHCP au client

- Client RADIUS

```
# DHCP config for eth0
auto eth0
iface eth0 inet dhcp
#           hostname koty_image2-1
```

Vérification : 192.168.122.120/24

```
root@client:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.122.120  netmask 255.255.255.0  broadcast 192.168.122.255
        ether 02:42:d9:20:4b:00  txqueuelen 1000  (Ethernet)
        RX packets 106  bytes 10941 (10.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10  bytes 1856 (1.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

5.3 Installation de FreeRADIUS

apt install freeradius

```
root@radius:/# apt install freeradius
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
freeradius is already the newest version (3.2.5+dfsg-3~ubuntu24.04.2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@radius:/#
```

Cette commande installe le serveur FreeRADIUS sur la machine Linux qui servira de serveur d'authentification.

5.4 Configuration des clients RADIUS

On edite le fichier de configuration

root@radius:/# nano /etc/freeradius/3.0/clients.conf On renseigne adresse reseau suivi du masque et le secret

client 192.168.122.0/24 {

secret = passer123

}

```
  GNU nano 7.2                 /etc/freeradius/3.0/clients.conf
#               secret = testing123
#       }
#}


client 192.168.122.0/24 {
        secret          = passer123
}
```

- client 192.168.122.0/24 : Définit le réseau autorisé à interroger le serveur RADIUS
- secret = passer123 : Clé partagée entre le serveur RADIUS et les clients (routeurs)

5.5 Création des utilisateurs

Après avoir configurer le fichier clients.conf nous allons creer des utilisateur

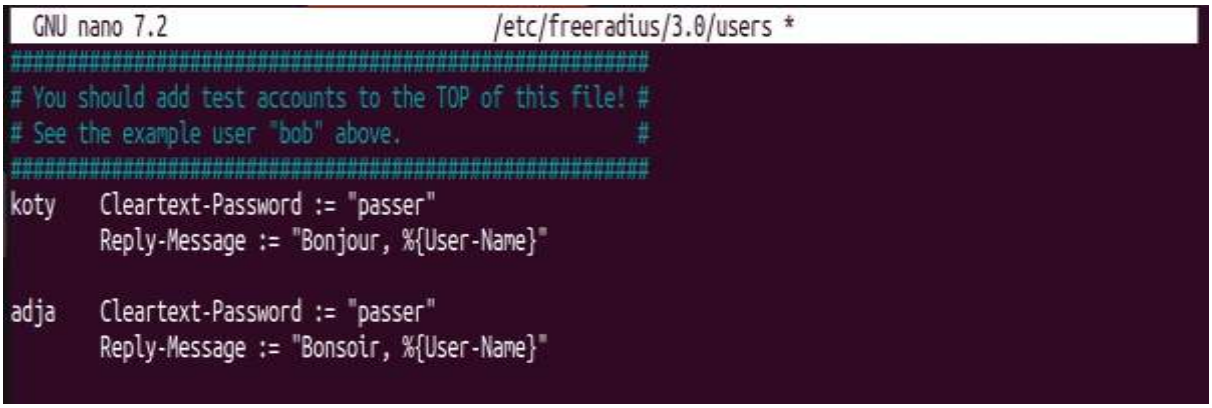
root@radius:/# nano /etc/freeradius/3.0/users

koty Cleartext-Password := "passer"

Reply-Message := "Bonjour, %{User-Name}"

adja Cleartext-Password := "passer"

Reply-Message := "Bonsoir, %{User-Name}"



Cleartext-Password : Définit le mot de passe en texte clair

Reply-Message : Message personnalisé envoyé lors de l'authentification réussie

%{User-Name} : Variable qui sera remplacée par le nom d'utilisateur

- Redémarrage du service

root@radius:/# service freeradius restart

```
root@radius:/# service freeradius restart
 * Checking FreeRADIUS daemon configuration...                    [ OK ]
 * Stopping FreeRADIUS daemon freeradius                          [ OK ]
 * Starting FreeRADIUS daemon freeradius                          [ OK ]
root@radius:/#
```

- Vérification du port d'écoute

root@radius:/# netstat -anp | grep –w 1813

```
root@radius:/# netstat -anp | grep -w 1813
udp        0      0 0.0.0.0:1813            0.0.0.0:*                           350/freeradius
udp6       0      0 :::1813                 :::*                                350/freeradius
root@radius:/#
```

6 Tests de fonctionnement du serveur RADIUS

Test d'authentification

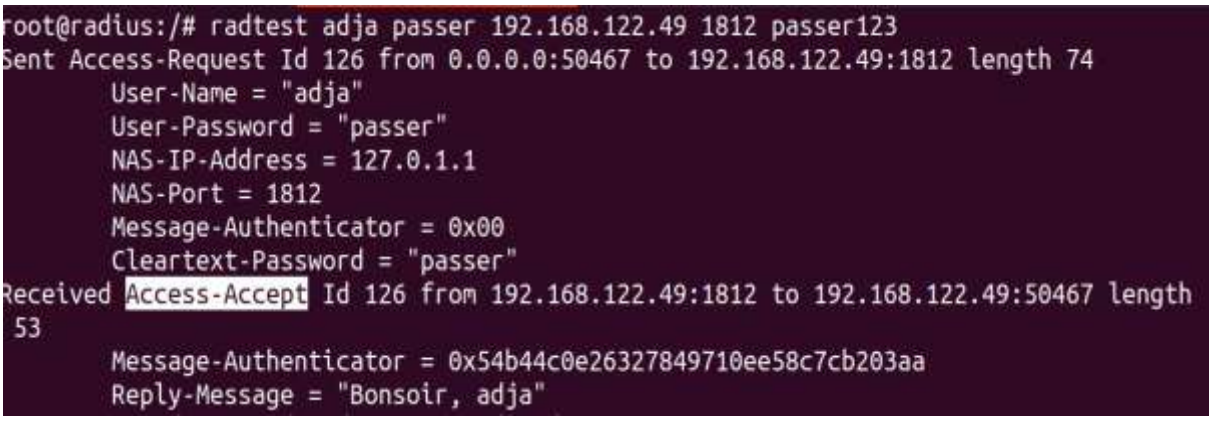
radtest koty passer 192.168.122.49 1812 passer123

radtest adja passer 192.168.122.49 1812 passer123

root@radius:/# radtest koty passer 192.168.122.49 1812 passer123

```
root@radius:/# radtest koty passer 192.168.122.49 1812 passer123
Sent Access-Request Id 151 from 0.0.0.0:51448 to 192.168.122.49:1812 length 74
        User-Name = "koty"
        User-Password = "passer"
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 1812
        Message-Authenticator = 0x00
        Cleartext-Password = "passer"
Received Access-Accept Id 151 from 192.168.122.49:1812 to 192.168.122.49:51448 length
 53
        Message-Authenticator = 0xf8fcece195f6f5c082eb5dac79aee0b8
        Reply-Message = "Bonjour, koty"
root@radius:/#
```

root@radius:/# radtest adja passer 192.168.122.49 1812 passer123



Explication :

radtest : Utilitaire de test RADIUS

koty/adja : Nom d'utilisateur à tester

passer : Mot de passe de l'utilisateur

192.168.122.49 : Adresse IP du serveur RADIUS

1812 : Port d'authentification RADIUS

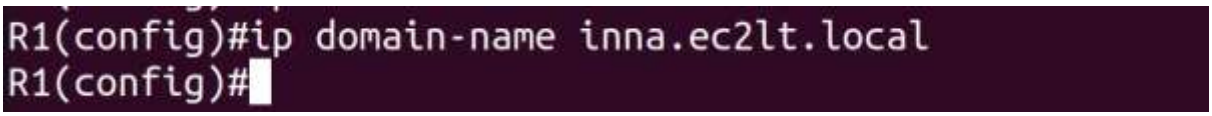
passer123 : Secret partagé

Ces tests permettent de valider que le serveur RADIUS fonctionne correctement avant de configurer le routeur.

7 Configuration SSH sur le routeur Cisco

- Création du domaine inna.ec2lt.local

ip domain-name inna.ec2lt.local



Cette commande définit le nom de domaine nécessaire pour la génération des clés SSH.

- Génération des clés ssh

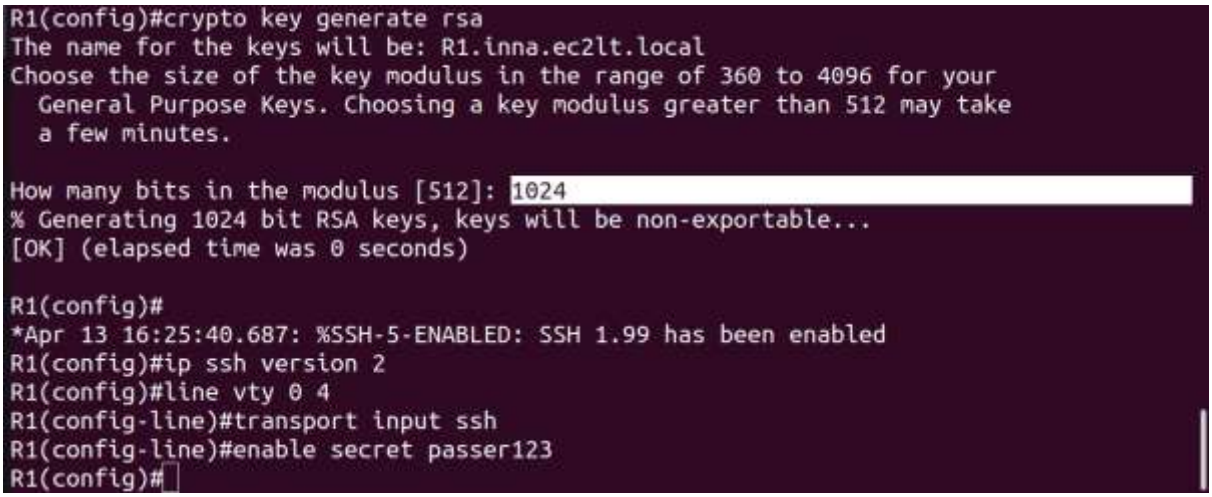
R1(config)#crypto key generate rsa

R1(config)#ip ssh version 2

R1(config)#line vty 0 4

R1(config-line)#transport input ssh

R1(config-line)#enable secret passer123



Explication :

crypto key generate rsa : Génère les clés RSA nécessaires pour SSH

ip ssh version 2 : Force l'utilisation de SSH version 2 (plus sécurisé)

line vty 0 4 : Configure les 5 lignes VTY (0 à 4) pour les connexions distantes

transport input ssh : Autorise uniquement SSH (bloque Telnet)

enable secret passer123 : Définit le mot de passe pour le mode privilégié

8 Configuration AAA (Authentication, Authorization, Accounting)

8.1 Configuration du model d'authentification

R1(config)#aaa new-model

R1(config)#aaa authentication dot1x default group radius

R1(config)#aaa authentication login default group radius

R1(config)#aaa authorization network default group radius

R1(config)#aaa authorization exec default group radius

```
R1(config)#
R1(config)#aaa new-model
R1(config)#aaa authentication dot1x default group radius
R1(config)#aaa authentication login default group radius
R1(config)#aaa authorization  network default group radius
R1(config)#aaa authorization  exec default group radius
R1(config)#
```

Explication :

Aaa new model : active le framework AAA sur le routeur

aaa authentication login default group radius : Configure l'authentification de connexion via RADIUS

aaa authorization exec default group radius : Configure l'autorisation du mode exec via RADIUS

aaa authorization network default group radius : Configure l'autorisation réseau via RADIUS

8.2 Configuration du serveur RADIUS

R1(config)#radius-server host 192.168.122.49 auth-port 1812 key passer123

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#radius-server host 192.168.122.49 auth-port 1812 key passer123
R1(config)#exit
```

Explication :

radius-server host 192.168.122.49 : Spécifie l'adresse IP du serveur RADIUS

auth-port 1812 : Définit le port d'authentification RADIUS

key passer123 : Clé partagée (doit correspondre à celle configurée sur le serveur)

8.3 Test de connectivé vers le serveur

koty & adja

R1#test aaa group radius koty passer legacy

R1#test aaa group radius adja passer legacy

```
R1#test aaa group radius koty passer legacy
Attempting authentication test to server-group radius using radius
User was successfully authenticated.

R1#test aaa group radius adja passer legacy
Attempting authentication test to server-group radius using radius
User was successfully authenticated.
```

Ces commandes testent l'authentification RADIUS directement depuis le routeur pour valider la configuration.

9 Résolution des problèmes de compatibilité SSH

Connexion depuis la machine cliente vers le routeur avec les utilisateurs koty

root@client:/# ssh koty@192.168.122.193

```
root@client:/# ssh koty@192.168.122.193
Unable to negotiate with 192.168.122.193 port 22: no matching key exchange method fou
nd. Their offer: diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1,diffi
e-hellman-group1-sha1
root@client:/#
```

9.1 Modification de la configuration SSH client

Résolvons le problème en éditant le fichier de configuration

root@client:/# nano /etc/ssh/ssh_config

On documente la ligne Ciphers….

Et on ajoute les deux lignes suivantes:

1- **KexAlgorithms diffie-hellman-group-exchange-sha1,diffiehellman-group14-sha1,diffie-hellman-group1-sha1**

2- **HostKeyAlgorithms ssh-rsa**

Ces lignes configurent les algorithmes de chiffrement compatibles avec les anciens équipements Cisco qui ne supportent pas les algorithmes modernes par défaut.

Redémarrage du service SSH

root@client:/# service ssh restart

```
root@client:/# service ssh restart
 * Restarting OpenBSD Secure Shell server sshd
              [ OK ]
```

10 Test de connexion finale

NB: mot de passe enable: passer123

Utilisateur mot de passe : passer

10.1 Connexion utilisateur koty

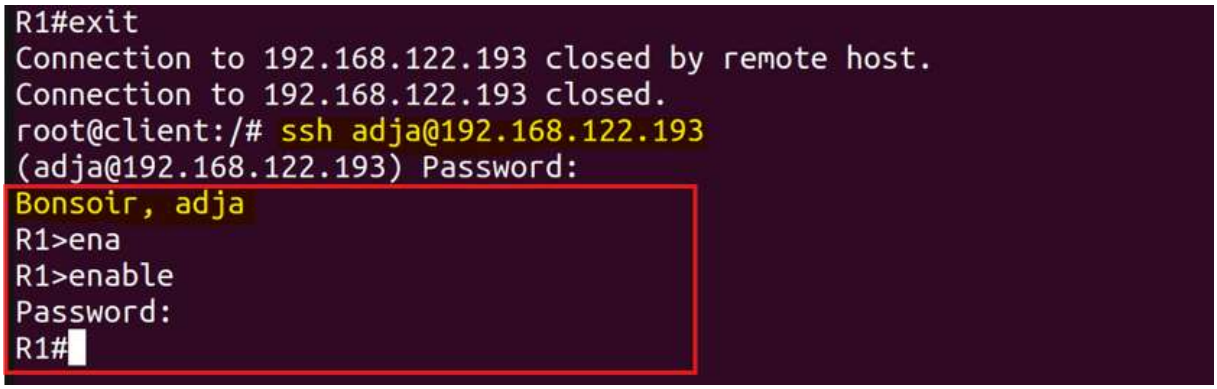
root@client:/# ssh koty@192.168.122.193

```
root@client:/# ssh koty@192.168.122.193
The authenticity of host '192.168.122.193 (192.168.122.193)' can't be established.
RSA key fingerprint is SHA256:yovj+QlRVDe9vZu55JvcXaJOLjICcC0NblHw6AIys3c.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.122.193' (RSA) to the list of known hosts.
(koty@192.168.122.193) Password:
Bonjour, koty

R1>enable
Password:
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#
```

10.2 Connexion utilisateur adja

root@client:/# ssh adja@192.168.122.193



Résumé

a. **Initiation de connexion** : L'utilisateur lance ssh koty@192.168.122.193
b. **Demande d'authentification** : Le routeur reçoit la demande de connexion SSH
c. **Requête RADIUS** : Le routeur interroge le serveur RADIUS (192.168.122.49:1812)
d. **Validation** : Le serveur RADIUS vérifie les identifiants dans sa base
e. **Réponse** : Le serveur renvoie Accept ou Reject
f. **Autorisation** : En cas d'Accept, le routeur autorise la connexion SSH
g. **Session établie** : L'utilisateur accède à l'interface CLI du routeur

Conclusion

Cette configuration démontre l'intégration réussie d'un serveur RADIUS avec un routeur Cisco pour l'authentification SSH. L'architecture mise en place offre une solution robuste et scalable pour la gestion centralisée des accès aux équipements réseau.