Documentation : Attribution d'IP fixes via DHCP avec Radius

Documentation de Patterson CORREA

1. Introduction

Dans le cadre de la M2L, certaines applications nécessitent des adresses IP fixes pour fonctionner correctement, notamment pour la prise en main à distance. Configurer manuellement des adresses statiques sur chaque poste serait fastidieux. La solution retenue consiste à utiliser un serveur DHCP couplé à un serveur Radius pour attribuer dynamiquement des adresses IP fixes en fonction de l'adresse MAC des machines.

2. Technologies utilisées

- **Serveur DHCP** : ISC DHCP ou un serveur DHCP Windows
- **Serveur Radius** : FreeRADIUS (sous Linux)
- **Base de données** : MySQL/MariaDB pour stocker les associations MAC/IP
- **Outils de scan réseau** : `nmap` , `arp-scan`
- **Scripting** : Python/Bash pour automatiser l'import des données

3. Architecture

1. Les postes clients envoient une requête DHCP au serveur DHCP.
2. Le serveur DHCP interroge le serveur Radius pour vérifier si une adresse IP fixe est associée à l'adresse MAC.
3. Si une correspondance est trouvée, l'IP est attribuée au poste.
4. Sinon, une IP dynamique peut être attribuée temporairement.
5. Un scan régulier met à jour la base de données des associations MAC/IP.

4. Installation et configuration

4.1. Installation du serveur DHCP

Sous Debian/Ubuntu :

```
sudo apt update && sudo apt install isc-dhcp-server
```

Configurer `/etc/dhcp/dhcpd.conf` :

```
option domain-name "m2l.local";
option domain-name-servers 192.168.1.1;
default-lease-time 600;
max-lease-time 7200;
authoritative;

subnet 192.168.1.0 netmask 255.255.255.0 {
    range dynamic-bootp 192.168.1.100 192.168.1.200;
    option routers 192.168.1.1;
}
```

4.2. Installation de FreeRADIUS

```
sudo apt install freeradius freeradius-mysql
```

Configurer `/etc/freeradius/3.0/clients.conf` :

```
client dhcp-server {
    ipaddr = 192.168.1.2
    secret = secret123
}
```

4.3. Base de données des adresses MAC/IP

Créer une table dans MySQL/MariaDB :

```
CREATE DATABASE radius;
USE radius;
CREATE TABLE mac_ip (
    id INT AUTO_INCREMENT PRIMARY KEY,
    mac_address VARCHAR(17) UNIQUE NOT NULL,
```

```
ip_address VARCHAR(15) NOT NULL,  
hostname VARCHAR(50)  
);
```

Remplir la table avec les machines connues :

```
INSERT INTO mac_ip (mac_address, ip_address, hostname) VALUES ('00:1  
A:2B:3C:4D:5E', '192.168.1.10', 'poste1');
```

4.4. Script d'importation des adresses MAC

Un script Python peut être utilisé pour scanner le réseau et mettre à jour la base :

```
import subprocess  
import mysql.connector  
  
def scan_network():  
    result = subprocess.run(["arp-scan", "-l"], capture_output=True, text=True)  
    lines = result.stdout.split("\n")  
    devices = []  
    for line in lines:  
        parts = line.split()  
        if len(parts) >= 2:  
            mac, ip = parts[0], parts[1]  
            devices.append((mac, ip))  
    return devices  
  
def update_database(devices):  
    conn = mysql.connector.connect(user='root', password='password', host='localhost', database='radius')  
    cursor = conn.cursor()  
    for mac, ip in devices:  
        cursor.execute("REPLACE INTO mac_ip (mac_address, ip_address) VALUES (%s, %s)", (mac, ip))  
    conn.commit()  
    conn.close()
```

```
if __name__ == "__main__":  
    devices = scan_network()  
    update_database(devices)
```

5. Tests et validation

- Redémarrer le serveur DHCP et vérifier qu'il interagit correctement avec Radius.
- Connecter un poste et s'assurer qu'il obtient bien l'adresse IP fixée.
- Effectuer plusieurs redémarrages pour vérifier la persistance des attributions.

6. Conclusion

Cette solution permet d'automatiser l'attribution d'IP fixes sans configuration manuelle des postes, tout en garantissant une gestion centralisée et scalable des adresses IP. Radius apporte une couche supplémentaire de contrôle et de sécurisation des attributions.