Projet : Réalisation d'un DHCP Server**.**

Réalisé par : Yanis ZEMMOURI & Nazim BENRABAH | 28715827 et 28717146

Langage de programmation : Python

Packages necessaires: *sockets, scapy, random2, binascii, getmac, ipaddress, ipcalc, os, netaddr, json, requests*

Introduction

Ce document a pour but de présenter les fonctionnalités du Serveur DHCP réalisé en expliquant les procédures suivies dans notre programme.

Les codes source sont constitués de deux fichiers python : *dhcp_parameters.py* et *dhcp_server.py* qui représentent respectivement le programme de paramétrisation et le programme de fonctionnement du Serveur.

1. Paramétrisation

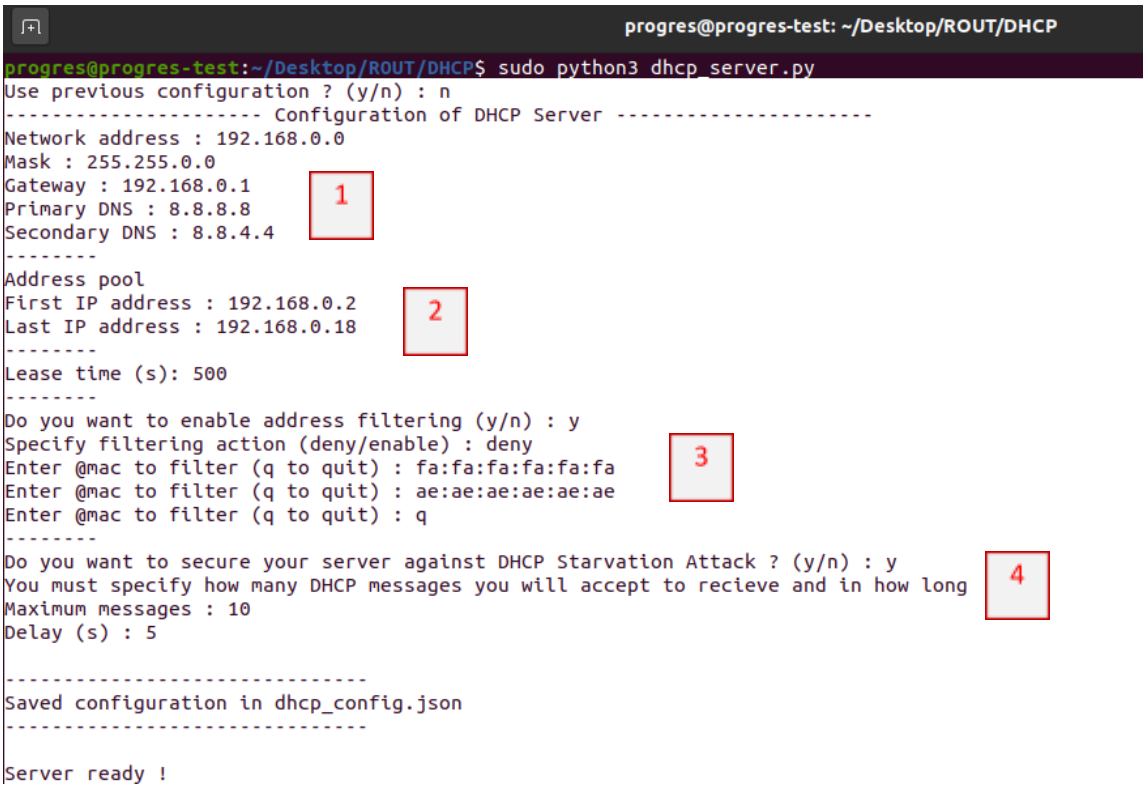
Les paramètres de configuration introduits par l'utilisateur sont récupérés à l'aide du programme *dhcp_parametrisation.py*

En premier lieu, il propose une interface de configuration dans laquelle un utilisateur tape les paramètres du serveur DHCP. Cette configuration est par la suite sauvegardée sous forme d'un fichier json (*dhcp_config.json*).

Grâce à cette sauvegarde, notre programme propose à l'utilisateur s'il désire utiliser la configuration précédemment enregistrée. Si oui, les paramètres du Serveur sont importés directement et ce dernier est lancé. Dans le cas contraire, l'utilisateur retape la configuration.

Aussi, si un utilisateur désire utiliser la même config précédente mais en changeant simplement quelques paramètres, il est ainsi possible de modifier manuellement le fichier *dhcp_config.json*, le sauvegarder, puis de lancer le serveur en spécifiant qu'il désire utiliser la configuration précédente.

Interface :



[1] : L'utilisateur introduit l'adresse réseau, le Mask, la Gateway, et le DNS primaire et secondaire.

Rq : Si la Gateway introduite n'appartient pas au réseau ou qu'elle soit identique à l'adresse Network ou Broadcast, un message d'erreur est affiché et redonne la main pour rectifier l'erreur.

[2] : L'utilisateur introduit la plage d'adresse en spécifiant les deux adresses d'extrémité ainsi que la durée de vie de l'attribution d'une adresse

Rq : Si une des adresses introduites n'appartient pas au réseau ou qu'elle soit identique à l'adresse Network ou Broadcast, un message d'erreur est affiché et redonne la main pour rectifier l'erreur.

[3] : L'utilisateur choisit de filtrer les adresses MAC des client en choisissant comme action de refuser (blacklist) fa:fa:fa:fa:fa:fa et ae:ae:ae:ae:ae:ae. Il est également possible de choisir comme action de n'accepter (whitelist) que ces adresses là en spécifiant « enable » au lieu de « deny ».

[4] : l'utilisateur choisit de protéger le Server DHCP contre une attaque qui vise à épuiser les ressources de la plage d'adresses, en évitant les rafales de requêtes DHCP (Ex : accepter que 10 requêtes pendant 5 secondes).

2. Fonctionnement

Le programme principal *dhcp_server.py* qui permet de fonctionnement du protocole et qui fait appel à l'autre programme pour mettre en arguments paramètres de configuration dans les fonctions qui le constituent.

En premier lieu, le programme importe les paramètres de config sous forme de dictionnaire. A l'aide d'une Socket UDP, il écoute sur le port 67 les messages broadcastés (DHCP Discover/Request des clients).

Si un message lui parvient, il analyse le champ qui comporte la Source MAC de ce dernier. Suivant la paramétrisation du filtrage MAC, si cette adresse n'est pas désirée, le message est ignoré, dans le cas contraire le message est traité.

Le traitement d'un message DHCP accepté par notre serveur consiste à analyser les champs pertinents : Client MAC, Message type, Transaction ID, IP requested.

Ces valeurs-là sont mises comme argument dans la fonction server_respond(). Cette dernière permet soit d'envoyer un OFFER (dans le cas ou le message reçu est un DISCOVER) ou alors un ACK (dans le cas où le message reçu est un Request).

L'envoi d'un message OFFER ou ACK se fait à l'aide du module Scapy, permettant de construire un paquet formaté en superposant : ETHERNET / IP / UDP / BOOTP / DHCP. Ces quatre derniers sont construits respectivement à l'aide des fonction Ether(), IP(), UDP(), BOOTP() et DHCP(), prenant chacune comme argument d'une part les valeurs de notre configuration (Ex : Gateway, broadcast, dns,…Etc) et d'autre part des valeurs extraite du message reçu par notre Serveur (Client_MAC, Transaction ID), et ce dans le but d'avoir un fonctionnement réactif du protocole DHCP.

3. Historique et suivi

Il est possible de suivre un historique horodaté des événements produits à l'aide de la fonctionnalité de Logging stockée dans le fichier *DHCP.log.* Parmi ces évènements on retrouve :

- Démarrage du serveur
- Réception d'un DISCOVER/REQUEST avec Mac client
- Refus d'un message de la part d'un client ayant une MAC filtrée
- Envoi d'un OFFER/ACK avec le Mac client et l'ip attribuée
- Alerte DHCP Starvation Attack.