République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Licence en Informatique

Thème

Administration et configuration d'un système d'authentification FreeRADIUS par PfSense pour sécuriser l'accès à l'internet.

Réalisé par :

- Bouzi Ghizlane.
- Fekih Houaria.

Présenté le 28 mai 2015 devant la commission d'examination composée de MM.

- *BEKARRA C.* *(Examinateur)*
- *BENAISSA Mohamed* *(Encadreur)*
- *DIDI F.* *(Examinateur)*

Année universitaire : 2014-2015

Remerciements

Nous remercions avant tout le Bon Dieu de nous avoir donné la volonté de finir ce mémoire.

Pour commencer, nous voulons adresser nos sincères remerciements et nos gratitude à notre directeur de mémoire, Mr. BENAISSA Mohamed, de nous avoir encadré, ainsi que pour sa grande disponibilité, son orientation, son aide et surtout ses judicieux conseils tout au long de la rédaction de ce mémoire.

Et nous tenons à exprimer nos remerciements aux membres de jury pour l'honneur qu'ils nous ont fait en acceptant de siéger à notre soutenance.

Nous désirons aussi remercier les professeurs du département d'Informatique, intervenants et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé nos réflexions et ont accepté à nous rencontrer et répondre à nos questions.

Finalement, nous tenons à exprimer notre profonde gratitude à nos familles qui nous ont toujours soutenues et à tout ce qui participe de réaliser ce mémoire.

Dédicaces

Je dédie ce modeste travail à:

A la lumière de mes jours, la source de mes efforts, la flamme de mon cœur, ma vie et mon bonheur. Que dieu te garde dans son vaste paradis, maman que j'adore.

La miséricorde de Dieu à mon père.

A mon grand-père que dieu procure bonne santé et longue vie.

Aux personnes dont j'ai bien aimé la présence dans ce jour, à mes frères Hicham et Salim et ma sœurs Iman, et ma nièces Hadjer.

À tous mes proches de la famille BOUZI et MIM.

je dédie ce travail dont le grand plaisir leurs revient en premier lieu pour leurs conseils, aides, et encouragements. Aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui m'ont accompagnaient durant mon chemin d'études supérieures, mes aimables amis, collègues d'étude, et frères de cœur, Zahra , Fadia et Hanane.

A mon binôme Houaria et toute la famille FEKIH.

Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, je vous dis merci.

BOUZI Ghizlane.

Dédicaces

Je dédie ce travail:

A mes chers et respectueux parents Ahmed et Chahira:

Grâce à vos tendres encouragements et vos grands sacrifices, vous avez pu créer le climat affectueux et propice à la poursuite de mes études.

Aucune dédicace ne pourrait exprimer mon respect, ma considération et mes profonds sentiments envers vous.

Je prie le Dieu de vous protéger, vous procurer santé et que la réussite soit toujours à ma porte pour que je puisse vous combler de bonheur, en espérant que vous serez toujours fiers de moi.

A mes frères et ma sœur: Nadia, Amine et Abdelatif à qui je souhaite beaucoup de bonheur, de santés et de succès.

A tous les membres de ma famille pour leur soutien et générosité.

A mes meilleurs amies, merci pour les très bons moments qu'on avait partagé ensemble, Fadia, Hanane et Zahra.

A mon binôme Ghizlane ainsi toute sa famille.

Enfin, je voudrais dédier ce mémoire à tout personnes ayant participé de loin ou de près à la réalisation de ce travaille.

FEKIH Houaria.

Table de matières

Chapitre III: Configuration de serveur freeradius avec Mysql

Chapitre IV: Configuration d'un serveur RADIUS par PfSense

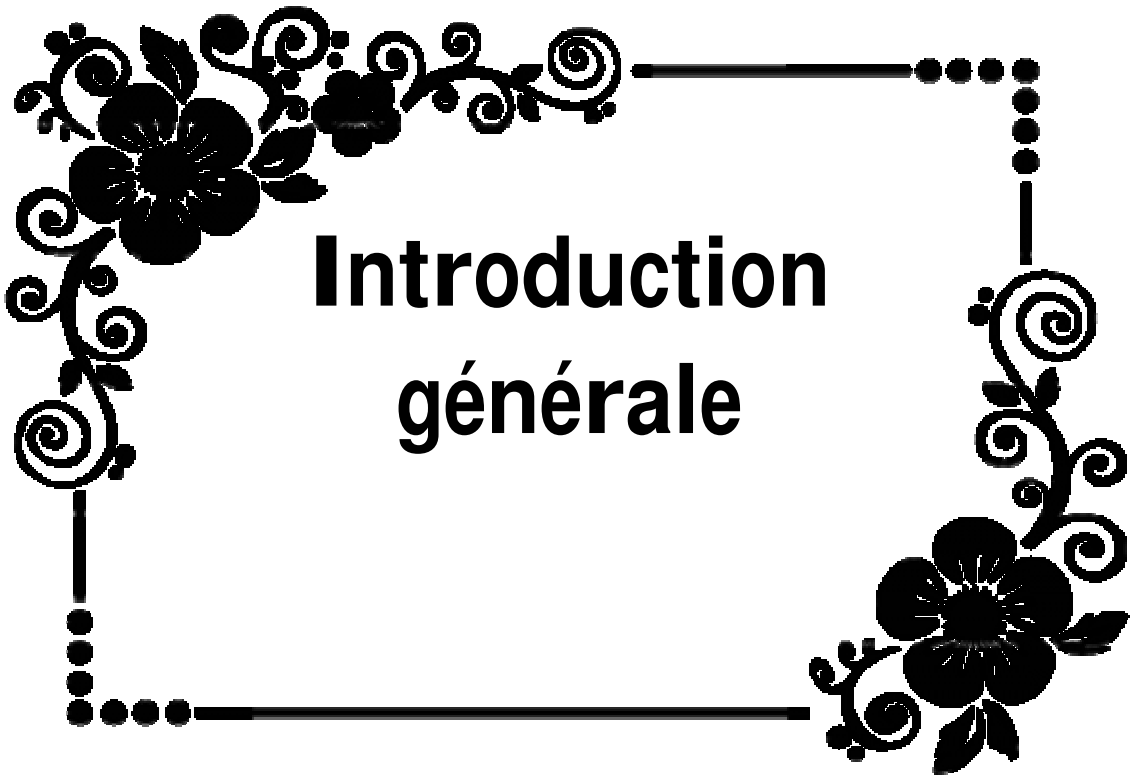
Liste des figures

Liste des Tables

Liste des abréviations:

AAA	Authentication, Authorization and Accounting.
ADSL	Asymmetric Digital Subscriber Line.
BDD	Base De Données.
CHAP	Challenge-Handshake Authentication Protocol.
CPU	Central Processing Unit.
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
EAP	Extended Authentication Protocole.
FAI	Fournisseur d'accès à Internet.
FreeBSD	Free Berkeley Software Distribution.
GTC	Generic Token Card.
GUI	Graphical User Interface.
HTML	HyperText Markup Language.
HTTP	HyperText Transfer Protocol.
IETF	Internet Engineering Task Force.
IP	Internet Protocol.
IPSEC	Internet Protocol Security.
LAN	Local Area Network.
LDAP	Lightweight Directory Access Protocol.
LEAP	Lightweight Extensible Authentication Protocol.
MAC	Message Authentication Codes.
MD5	Message Digest 5.
MS-CHAP	Microsoft Challenge Handshake Authentication Protocol.
NAS	Network Access Server.
NTP	Network Time Protocol.
OS	Operating System.
OVF	Open Virtualization Format.
PAM	Pluggable Authentication Modules.
PAP	Password Authentication Protocol.

PEAP	Protected Extensible Authentication Protocol.
PF	Packet Filter.
PHP	Hypertext Preprocessor.
POP	Post Office Protocol.
PPP	Point-to-Point Protocol.
PPTP	Point-to-Point Tunneling Protocol.
PXE	Preboot Execution Environment.
RADIUS	Remote Authentication Dial-In User Service.
RAM	Random Acces Memory.
RDP	Remote Desktop Protocol.
RFC	Request For Comments.
RTC	Réseau Téléphonique Commuté.
SGBD	Système de Gestion de Base de Données.
SIM	Subscriber Identity Module.
SSH	Secure Shell.
SSL	Secure Sockets Layer.
TLS	Transport Layer Security.
TTL	Tunneled Transport Layer.
TTLS	Tunneled Transport Layer Security.
UDP	User Datagram Protocol.
URL	Uniform Resource Locator.
USB	Universal Serial Bus.
VLAN	Virtual LAN.
VPN	Virtual Private Network.
WAN	Wide Area Network.
WEP	Wired Equivalent Privacy.
WPA	Wi-Fi Protected Access.
WWW	World Wide Web.
XML	Extensible Markup Language.



Introduction générale

Lorsque l'on doit assurer le bon fonctionnement d'un réseau qui dépasse les dimensions du réseau familial, il devient nécessaire de s'assurer aussi que le danger ne vient pas de l'intérieur. Avec la prolifération des ordinateurs personnels portables, le risque de voir une machine inconnue venir polluer le réseau de l'intérieur doit être pris au sérieux.

Pour peu qu'un accès réseau ou wifi soit également disponible, il convient en plus de s'assurer que seules les personnes autorisées puissent s'y attacher. Les premiers réseaux sans fil faisaient donc l'objet de nombreux soucis liés à l'authentification des utilisateurs et au chiffrement des communications.

La première solution à ce problème fut l'apparition des clés WEP qui étaient utilisées à la fois pour authentifier et chiffrer les échanges. Elles deviennent très vite obsolètes car l'algorithme a été rapidement cassé et la méthode d'authentification n'était pas fiable (basée sur une clé partagée). Mais l'avènement du WPA et du WPA2 en 2004 a été d'un grand secours pour les réseaux sans fil.

Pratiquement tous les postes utilisateurs sont équipés d'une couche logicielle appelée supplicant qui prend en compte le WPA qui offre ainsi les fonctions d'authentification et de chiffrement. Pour pouvoir s'authentifier, le supplicant doit envoyer des requêtes à un serveur dédié qui se charge de vérifier les informations et d'accepter ou refuser les demandes.

L'objectif de ce projet de fin d'étude est d'installer et administrer un serveur d'authentification FreeRADIUS pour sécuriser l'accès à un réseau filaire ou sans fil en utilisant PFSense.

Le fonctionnement de RADIUS est basé sur un système client /serveur chargé de définir les accès d'utilisateurs distants à un réseau. Il s'agit de protocole de prédilection des fournisseurs d'accès à internet.

Notre mémoire est structurée comme suite:

Chapitre 1 : introduction aux systèmes d'authentification.

Chapitre 2 : présentation de serveur d'authentification freeradius.

Chapitre 3: présentation les différentes étapes d'installation et configuration de freeradius.

Chapitre 4 : test le système d'authentification freeradius sous pfsense avec l'utilisation un réseau Virtual créer par virtualbox.

Chapitre I

Introduction a un serveur d'authentification RADIUS

I.1. Introduction

La sécurité reste un élément important dans l'établissement de connexions distantes. Certains serveurs et protocoles permettent une sécurité accrue des échanges d'informations.[3]

Parmi ceux là, on distingue un protocole RADIUS permettant de centraliser l'authentification et l'autorisation des utilisateurs distant, à partir d'une seule base utilisateurs.[3]

Il est utilisé depuis longtemps déjà par bon nombre de fournisseurs d'accès à l'internet pour authentifier leurs clients et leur communiquer une configuration IP. RADIUS est également très utile pour sécuriser un réseau Wi-Fi, ou même un réseau filaire. [4]

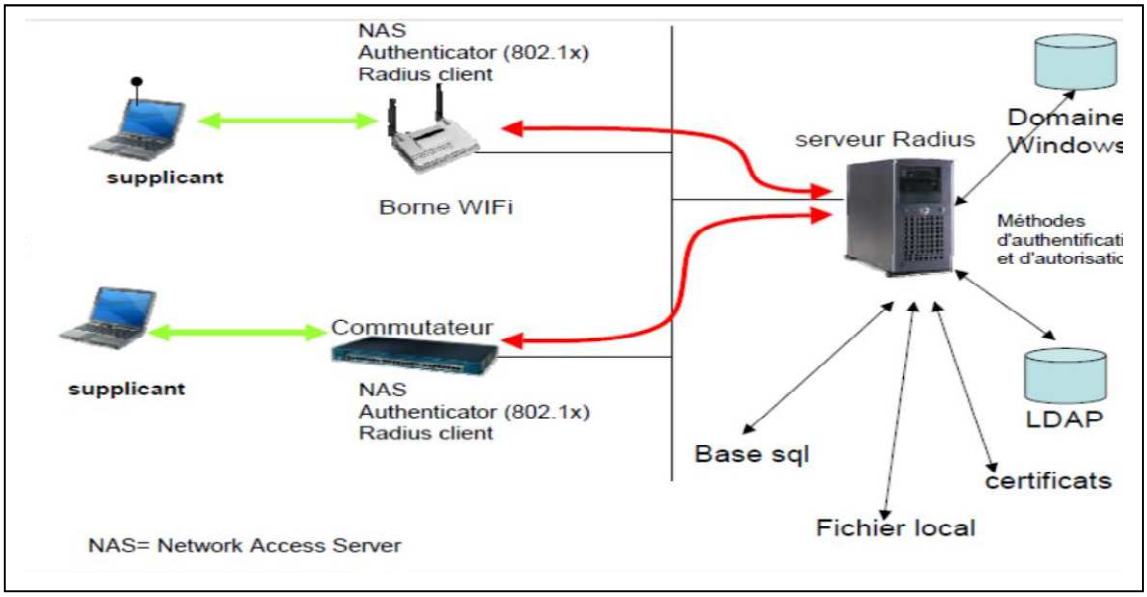


Figure 1.1 : schéma d'authentification par radius.

Ce type de sécurisation s'avère fort intéressant si l'on dispose d'un réseau proposant un accès filaire et Wi-Fi, avec des utilisateurs susceptibles de venir y connecter leur ordinateur portable. [4]

Ainsi RADIUS a deux possibilités d'authentifier les stations soit depuis leur adresse MAC connues sur notre réseau filaire, en utilisant un système de type «login/password », soit avec un certificat réseau Wi-Fi, en utilisant EAP-TLS. [4]

Le serveur RADIUS peut fonctionner en s'appuyant uniquement sur des fichiers texte. Mais pour gérer des multiples clients on aura besoin d'utiliser une base MySQL pour stocker les login et les mots de passe des clients. [4]

I.2. Définition

RADIUS est un protocole d'authentification client/serveur. Un serveur RADIUS dispose d'une base de données de droits utilisateur. Lorsqu'un utilisateur souhaite se connecter à un réseau régi par ce protocole, le NAS, un intermédiaire entre le réseau et l'utilisateur va interroger le serveur RADIUS et attribuer ses droits à l'utilisateur qui pourra dès lors accéder au réseau. [5]

I.3. Historique

Le protocole RADIUS a été développé à l'origine par Steve Willens pour la société Livingston Enterprises en 1991 comme un serveur d'authentification. Depuis devenu la propriété de Lucent Technologie, il appartient maintenant au domaine public.
RADIUS est aujourd'hui le protocole d'authentification le plus utilisé et le plus implémenté par les équipementiers réseaux. En effet bon nombre d'entre eux possèdent leur propre bibliothèque de paramètres RADIUS. [6]

I.4. Utilité

Le but de RADIUS était à l'origine de permettre aux fournisseurs d'accès à Internet d'authentifier les utilisateurs distants utilisant les connexions par modem à partir de multiples serveurs mais d'une seule base utilisateurs. Les noms et mots de passe des utilisateurs devaient être dupliqués dans chaque appareil ayant besoin d'identifier des utilisateurs. le serveur Apache est un des clients RADIUS les plus répandus. C'est toujours l'utilisation la plus courante du protocole RADIUS: nom et mot de passe de connexion à l'Internet, mais de plus en plus les réseaux sans fil ou filaires y ont aussi recours pour identifier les utilisateurs. [7]

I.5. Fonction de RADIUS

Un serveur RADIUS est un serveur qui implémente le protocole RADIUS et qui a principalement trois missions:

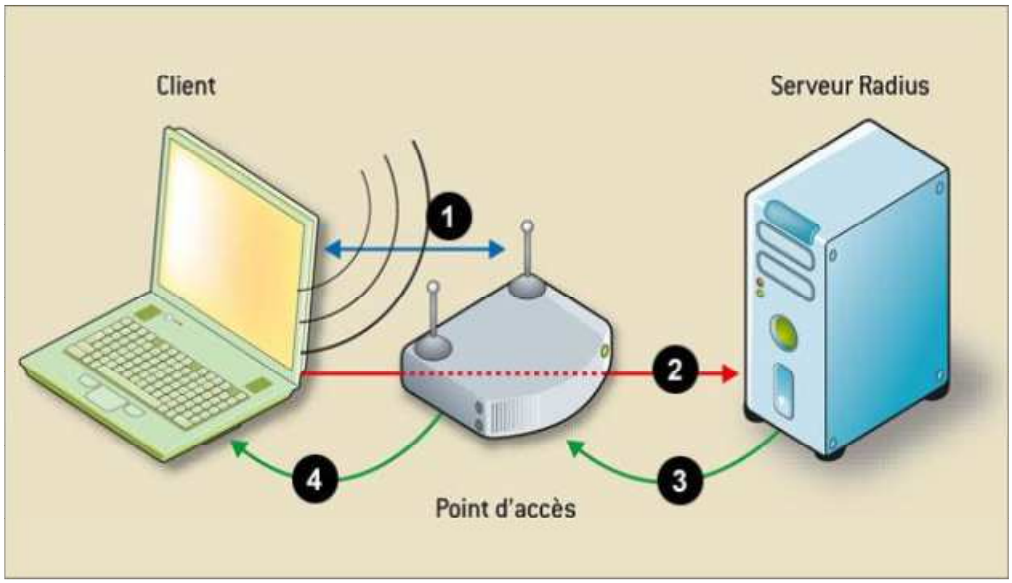


Figure I.2: principe de fonctionnement de RADIUS.

I.5.1. L'authentification

L'authentification est la procédure qui consiste, pour un système informatique, à vérifier l'identité d'une personne ou d'un ordinateur afin d'autoriser l'accès de cette entité à des ressources (systèmes, réseaux, applications…).L'authentification permet donc de valider l'authenticité de l'entité en question. Cette fonction est assurée de deux manières : soit par mot de passe, soit par certificat électronique. [8]

I.5.2. L'autorisation

Une autorisation est la fonction spécifiant les droits d'accès vers les ressources liées à la sécurité de l'information et la sécurité des systèmes d'information en général et au contrôle d'accès en particulier. Plus formellement, "autoriser" consiste à définir une politique d'accès. Par exemple, le personnel des ressources humaines est normalement autorisé à accéder aux informations sur les employés et cette politique est généralement

formalisée dans des règles de contrôle d'accès dans un système informatique. Pendant l'exécution, le système utilise les règles de contrôle d'accès pour décider si une requête d'accès venant d'un client (authentifié) va être approuvée (accordée) ou désapprouvée (rejetée). Les ressource incluent les fichiers, les données, les programmes les appareils et autres fonctionnalités fournies par les applications sur un ordinateur. Le client peut être des utilisateurs, des programmes et d'autres appareils sur l'ordinateur. [9]

I.5.3. La comptabilité

La comptabilité est une discipline pratique, consistant à schématiser, répertorier et enregistrer un certain nombre d'informations pour des fins ultérieures: heure de connexion, numéro de VLAN, adresse MAC… [10]

Il y a lieu de signaler que RADIUS assure une dernière mission qui est le chiffrement des transactions après l'authentification. [10]

I.6. Protocoles et ports de RADIUS

Le protocole est basé sur des échanges requêtes/réponses avec les clients RADIUS, c'est-à-dire les NAS. Il n'y a jamais de communication directe entre le poste de travail et le serveur. [11]

RADIUS connaît nativement deux protocoles de mot de passe: PAP (échange en clair du nom et du mot de passe), et CHAP (échange basé sur un hachage). Le protocole prévoit deux attributs séparés : User-Password et CHAP-Password. [7]

Le protocole établit une couche applicative au-dessus de la couche de transport UDP. Les ports utilisés seront: [11]

• 1812 pour recevoir les requêtes d'authentification et l'autorisation.

• 1813 pour recevoir les requêtes de comptabilité.

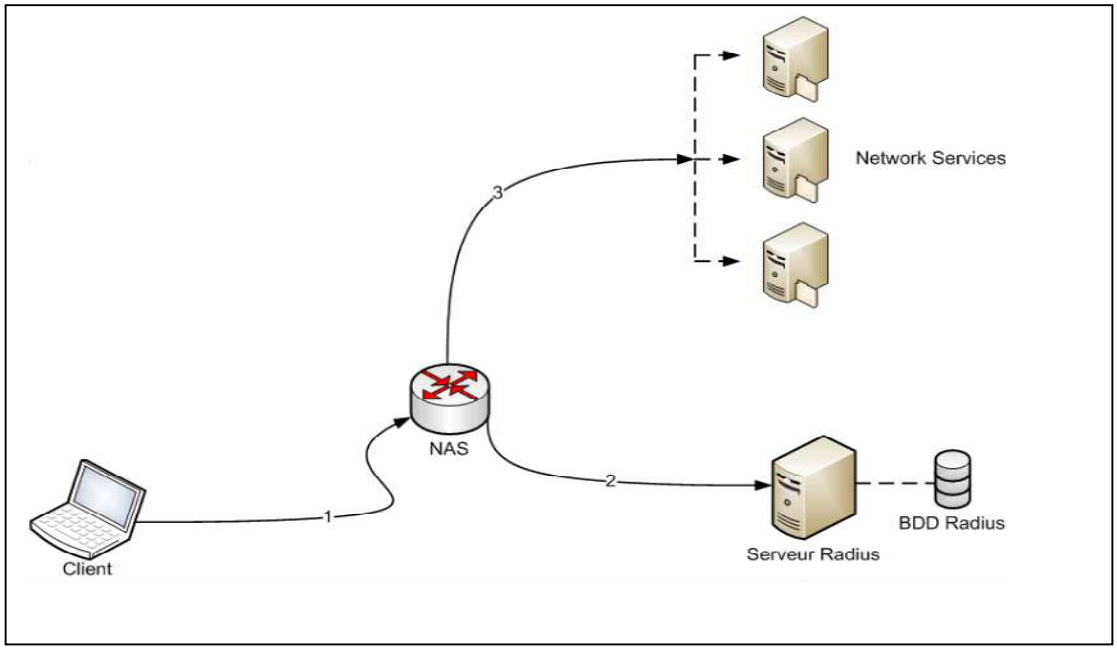


Figure I.3: schéma de connexion du client.

I.7. Les différents types de paquets

Le protocole RADIUS utilise 4 types de paquets différents pour assurer l'authentification:

- Le paquet **Acces-Request**: qui est émis par le NAS vers le serveur RADIUS pour initialiser la conversation. Il contient l'attribut User-Name et d'autres attributs comme Nas-Identifier… [10]
- Le paquet **Acces-Accept**: une réponse Acces-Accept à un paquet Access-Request indique que le serveur RADIUS a authentifié l'utilisateur avec succès.
- Le paquet **Acces-Reject**: une réponse Access-Reject à un paquet Access-Request indique que le serveur RADIUS n'a pas réussi à authentifié l'utilisateur.
- Le paquet **Acces-Challenge**: une réponse Access-Challenge à un paquet Access-Request indique que le serveur RADIUS requiert plus d'informations d'un autre paquet Access-Request avant d'authentifier l'utilisateur.

I.8. Format standard des paquets RADIUS

I.8.1. Structure des trames RADIUS

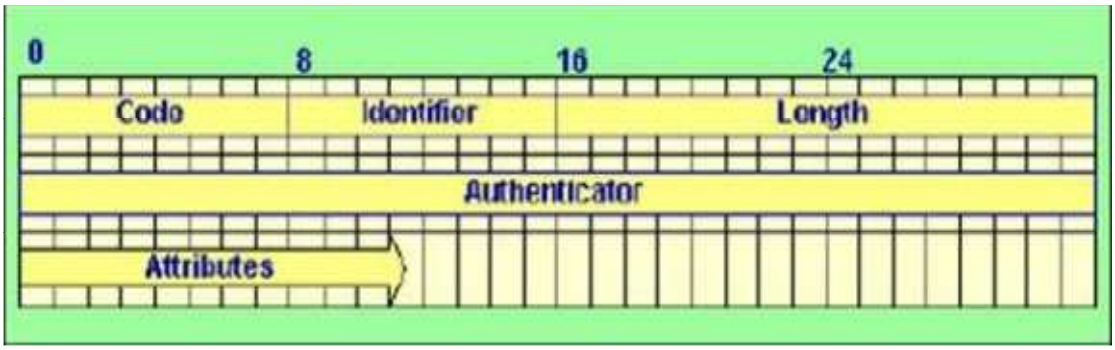


Figure I.4: structure des trames RADIUS.

I.8.2. Champs des trames RADIUS

La trame RADIUS contient cinq champs, le rôle de chacun d'entre eux est explicité ci-dessous:

Le champ code: le champ code d'une longueur d'un octet permet de spécifier le type du message contenu dans la trame. Neuf valeur sont possible pour ce champs chacune correspondant à un type de message possible lors des transactions RADIUS. Ces valeurs et leurs significations sont présentées dans le tableau ci-dessous: [6]

Code	Description
1	Access-Request
2	Access-Accept
3	Access-Reject
11	Access-Challenge

Table I.1 : valeur du champ Code RADIUS.

- **Le champ Identifier**: le champ « Identifier » d'une longueur d'un octet sert à corréler les trames de requête et de réponse au sien des NAS. [6]
- **Le champ Length**: ce champ d'une longueur de 2 octets sert lui à spécifier la longueur de la trame. [6]
- **Le champ Authenticator**: ce champ d'une longueur de 16 octets permet d'authentifier les réponses du serveur RADIUS le plus souvent il consiste en un hachage MD5 du secret. Il permet aussi de préciser le mécanisme d'authentification de l'utilisateur à utiliser. [6]
- **Le champ Attributes**: le champ Attributs et valeurs d'une longueur variable contient tous les tuples d'attributs valeurs de la requêtes ou réponse. Il contient les données transportées par la trame. [6]

I.9. Séquence d'établissement d'une session RADIUS

Le schéma ci-dessous présente les échanges effectués lors de l'établissement d'une session RADIUS classique:

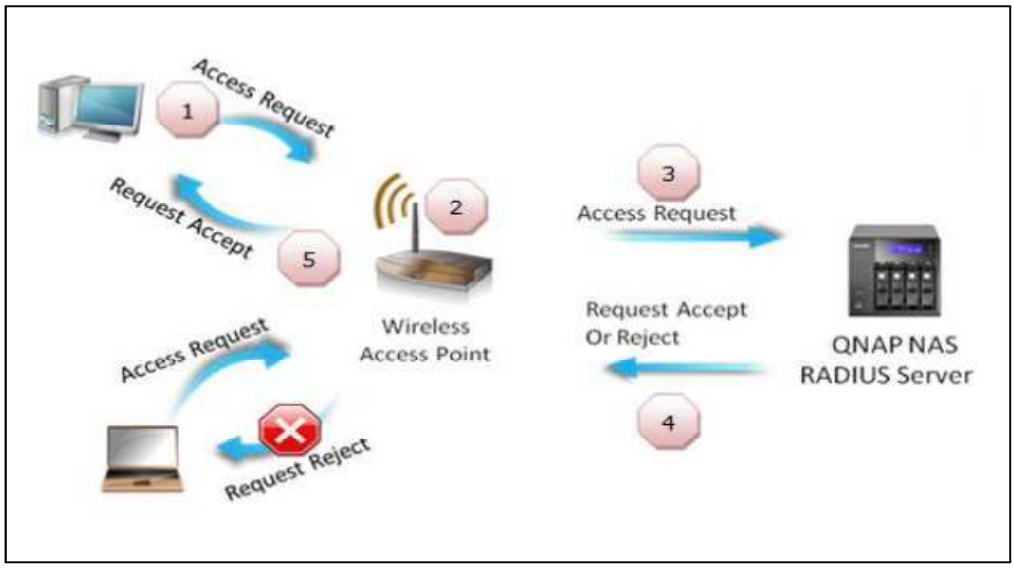


Figure I.5: établissement d'une session RADIUS.

1. Envois du couple «Login/password» crypte avec la clé partagée.
2. Si le couple est valide Acces-accept avec demande d'éventuelles informations supplémentaires (adresse IP, masque de réseau, etc.) qui amènerons à un nouvel échange Acces-request/Acces-accept.

3. Envois d'informations pour la phase comptabilité.
4. Le serveur répond lorsque les informations de compatibilité sont stockées.
5. Envois d'informations pour la phase comptabilité.
6. Le serveur répond lorsque les informations de compatibilité sont stockées. [6]

I.10. Les attributs et leurs valeurs

Les attributs constituent le principe le plus important du protocole RADIUS. Les champs attributs sont le fondement du protocole. Par conséquent, la bonne compréhension de leur signification et de leur rôle est indispensable pour tirer le meilleur parti de RADIUS. Chaque attribut possède un numéro d'attribut, auquel est associé un nom. La valeur d'un attribut peut correspondre à l'un des types suivants: [11]

- adresse IP (4 octets).
- date (4 octets).
- chaîne de caractères (jusqu'à 255 octets) .
- entier (4 octets) .
- valeur binaire (1 bit).
- valeur parmi une liste de valeurs (4 octets).

Il existe un grand nombre d'attributs dans le protocole RADIUS. Parmi eux, les plus intéressants sont:

•**User-Name**: Cet attribut est envoyé par le NAS et contient l'identifiant qui va servir de point d'entrée dans la base du serveur d'authentification. [11]

•**User-Password**: Il s'agit du mot de passe associé à User-Name, transmis par le NAS. Le serveur d'authentification valide ce mot de passe en fonction de la valeur enregistrée dans sa base de données. [11]

•**Nas-IP-Address**: Il s'agit de l'adresse IP du NAS qui communique avec le serveur. Cet attribut est transmis par le NAS lui-même. Son utilisation permettra d'authentifier un poste de travail à la condition qu'il soit connecté sur un NAS qui possède cette adresse IP. [11]

•**Nas-port**: Il s'agit du numéro de port du NAS sur lequel est connecté le poste de travail. Cet attribut est transmis par le NAS. Son utilisation permettra d'authentifier un poste de travail à la condition qu'il soit connecté sur ce numéro de port. [11]

•**Called-Station-Id**: Il s'agit de l'adresse MAC du NAS. Cet attribut est transmis par le NAS et permet d'authentifier un poste en fonction de l'équipement sur lequel il est connecté. [11]

•**Calling-Station-Id**: Il s'agit de l'adresse MAC du poste de travail qui se connecte au réseau. Cet attribut sera l'un des plus utiles. Il permettra d'authentifier un poste par une des méthodes (RADIUS-MAC ou bien EAP) et, en plus, de vérifier (et même d'imposer) que cette authentification s'effectue depuis un poste qui a pour adresse MAC la valeur de cet attribut. [11]

•Les attributs « vendor »: sont des fonctionnalités supplémentaires qui permettent de gérer les fonctions supplémentaires qui ne font pas partie des standards de RADIUS. [11]

Figure I.6: caractéristiques des attributs RADIUS.

I.11. Les différents serveurs RADIUS

Les serveurs RADIUS les plus répandus sont:

• FreeRADIUS.

• Open RADIUS.

• GNU-RADIUS.

• YARD-RADIUS.

I.12. Conclusion

Dans ce chapitre, nous avons présentées les différentes fonctionnalités d'un système d'authentification radius. Le serveur radius est un élément important dans la sécurisation et le contrôle d'accès à un réseau.

Chapitre II

Généralités sur FreeRADIUS

II.1. Introduction

Nous avons vu qu'un système RADIUS était un serveur d'authentification. Le serveur FreeRADIUS nécessite un bon nombre d'outils et de librairie qu'il faut préalablement installer pour que l'installation de celui-ci se passe sans problème et que toutes ses fonctionnalités soient opérationnelles. [12]

Pour tester si l'installation fonctionne, on a besoin d'un client RADIUS. Ceci dit, FreeRADIUS est suffisamment bien fait pour intégrer un client « radtest ». Il permet de simuler une demande d'authentification et donc de tester votre installation du RADIUS. [12]

```
root@serveur-desktop:~# radtest "John Doe" hello 127.0.0.1 1812 testing123
Sending Access-Request of id 239 to 127.0.0.1 port 1812
        User-Name = "John Doe"
        User-Password = "hello"
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 1812
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=239, length=3
        Reply-Message = "Hello, John Doe"
root@serveur-desktop:~#
```

Figure II.1: test de bon fonctionnement du serveur FreeRADIUS.

II.2. Définition

FreeRADIUS offre une alternative aux autres serveurs d'entreprise RADIUS, c'est un des serveurs RADIUS les plus modulaires et riches en fonctionnalités disponibles aujourd'hui. FreeRADIUS est utilisé par des fournisseurs d'accès à l'internet pour authentifier leurs clients et leur communiquer une configuration IP. Il est considéré comme le serveur le plus utilisé dans le monde. [13]

Bases d'authentification	Free RADIUS	Cistron RADIUS	ICRadius	XTRadius	Open RADIUS	GNU-Radius	YARD-Radius
PAM	✓	✓	✓		✓	✓	✓
Unix	✓	✓	✓	✓	✓	✓	✓
MySQL	✓		✓		✓	✓	
PostgreSQL	✓				✓	✓	
Oracle	✓						
LDAP	✓				✓		
Perl DBI	✓		✓				
Perl DBD	✓		✓				
Bekerley DB	✓		✓	✓		✓	✓
SMB	✓				✓		
ODBC	✓					✓	

Table II.1: les avantages de FreeRADIUS.

II.3. Historique

FreeRADIUS est une implémentation de RADIUS élaborée, FreeRADIUS commença à être développé en août 1999 par Alan Dekok et Miquel Van Smoorenburg. Miquel Van Smoorenburg avait précédemment écrit le serveur Cistron RADIUS, qui fut largement utilisé quand le serveur Livingston cessa d'être maintenu. Le serveur gagna rapidement le support de la communauté par l'addition de modules à intégrer avec LDAP, SQL et d'autres bases de données. **[13]**

Le support de EAP fut ajouté en 2001, le serveur supporte désormais la plupart des protocoles d'authentification. [13]

II.4. Les protocoles

Parmi les protocoles d'authentification compatibles citons : [11]

- IEEE 802.1X.
- EAP/TLS.
- EAP/PEAP.
- EAP/TTLS.
- EAP/SIM.
- EAP/GTC.
- EAP/MD5.
- LEAP.

- MS-CHAP.
- CHAP.
- PAM.

Protocole d'authentification	FreeRADIUS	Cistron RADIUS	ICRadius	XTRadius	Open RADIUS	GNU-Radius	YARD-Radius
EAP/SIM	✓						
EAP/TLS	✓						
EAP/TTLS	✓						
EAP/MD5	✓						
CHAP	✓					✓	
PAP	✓					✓	
PEAP	✓						
CISCO/LEAP	✓						
Login/mot de passe	✓	✓	✓	✓	✓	✓	✓

Table II.2: comparaison entre les différents serveurs RADIUS.

II.5. Principe de fonctionnement

Le processus exécuté par FreeRADIUS comprend principalement deux étapes: l'autorisation et l'authentification. Cela puisse paraître, c'est bien dans cet ordre que les opérations vont se dérouler. Bien sûr, FreeRADIUS ne va pas donner d'autorisations avant d'avoir authentifié le client. Il va préparer le terrain en établissant la liste des autorisations qui sera envoyée au NAS quand l'authentification sera positive. [11]

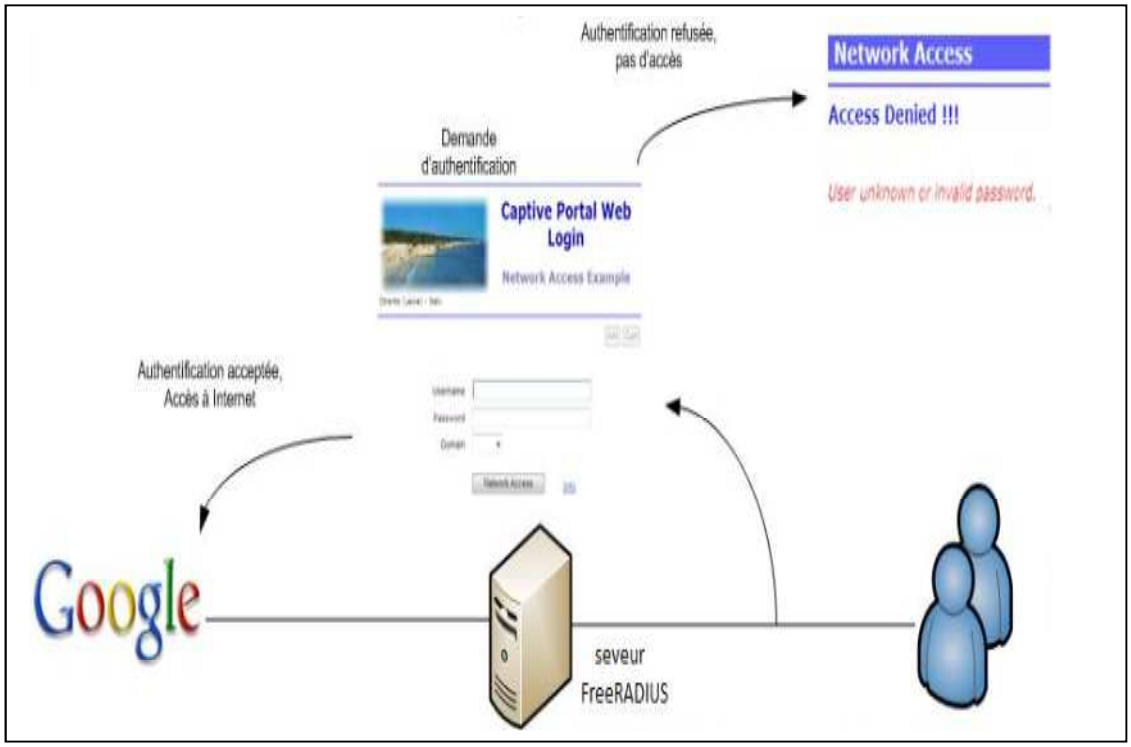


Figure II.2: principe d'authentification pour accès à internet en utilisant un serveur FreeRADIUS.

II.5.1. Soumission d'une requête

Les requêtes arrivent au serveur par le biais de paquets Access-Request pouvant contenir plusieurs attributs. On y trouvera toujours l'attribut User-Name qui contient l'identifiant et aussi des attributs tels que Calling-Station-Id, NAS Identifier, etc. Ces attributs sont appelés request-items. [11]

II.5.2. Recherche dans la base de données

Le nœud central du système FreeRADIUS est la ou les bases de données où les informations d'autorisations et d'authentification sont puisées. Une base d'autorisation est constituée d'une entrée par utilisateur ou par machine. Chaque entrée est caractérisée par un identifiant ainsi que par trois catégories d'attributs appelées check-items(une liste de critères supplémentaires auxquels la requête doit satisfaire), reply-items(attribut auquel une valeur est affectée dans la base d'autorisation) et config-items(les attributs internes du serveur liés à une requête). Une base d'authentification est aussi constituée d'une entrée par utilisateur ou par machine et contient les données d'authentification

(mot de passe). Si base d'authentification et d'autorisation ne font qu'une, ces données d'authentification y seront présentes sous formes de check-items(User-Password). [11]
FreeRADIUS recherche dans la base d'autorisation un identifiant égal à la valeur de User-Name. S'il le trouve, il compare la liste des check-items à la liste des request-items. S'il y a équivalence, l'entrée est validée, sinon l'entrée suivante est vérifiée et ainsi de suite jusqu'à la fin. [11]

II.5.3. Constitution de la liste des autorisations

Quand une entrée a été trouvée, FreeRADIUS en extrait la liste des reply-items et la met provisoirement de côté. Il s'agit là des autorisations qui seront accordées au poste qui se connecte. Quand l'authentification aura été réalisée, les reply-items seront écrits dans le paquet Access-Accept qui sera envoyé au NAS. [11]

II.5.4. Authentification

Après l'étape d'autorisation, FreeRADIUS passe à l'authentification dont le processus demande plus ou moins d'échanges avec le NAS suivant la méthode mise en œuvre (EAP, RADIUS-MAC...). Si l'authentification est positive, un paquet Access-Accept est construit avec la liste des reply-items mise de côté précédemment. Si l'authentification est négative, c'est un paquet Access-Reject qui est envoyé. [11]

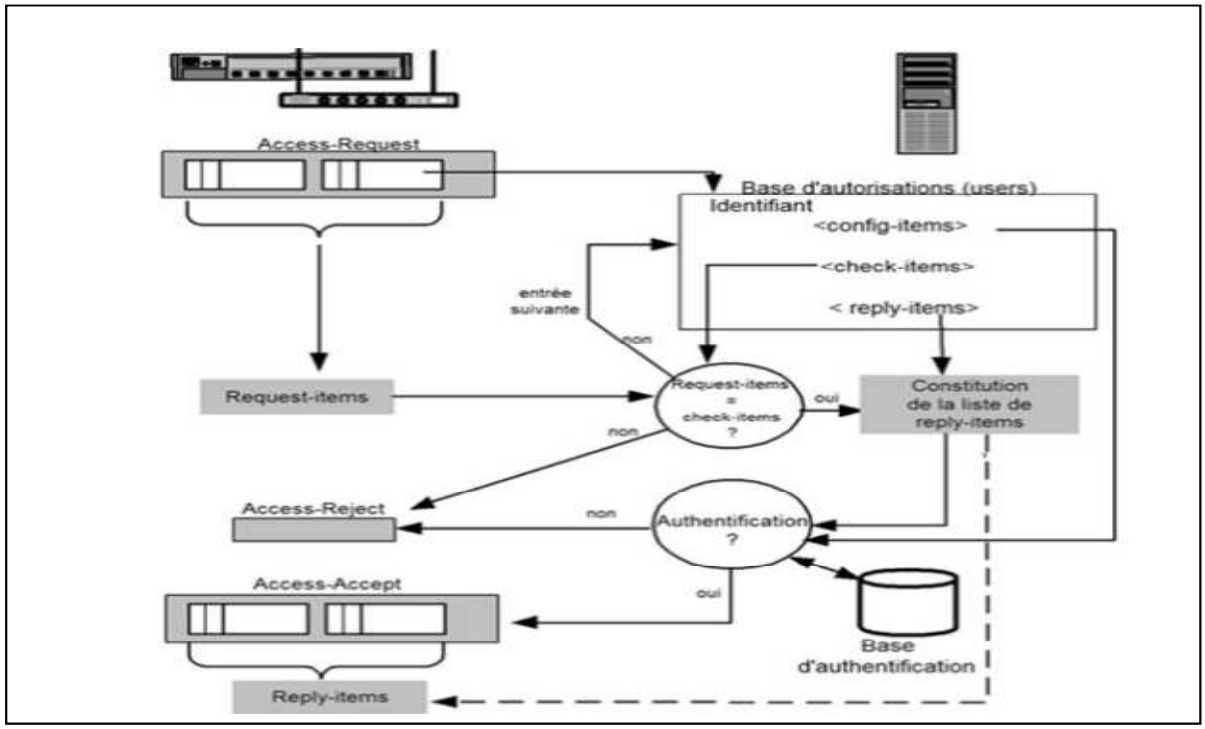


Figure II.3: principe de fonctionnement de FreeRADIUS.

II.6. La base d'authentification

Une base de données est en quelque sorte un gros fichier où sont stockées des données modifiables dynamiquement. L'avantage de ce système est que l'on modifie simplement les données contenues dans la base. [14]

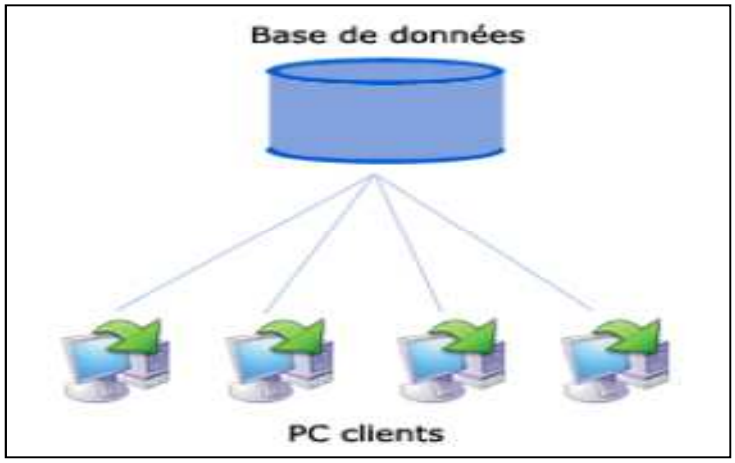


Figure II.4: schéma d'un exemple d'architecture faisant intervenir les bases de données.

La base d'authentification des clients et des utilisateurs sera une base de données MySQL. C'est est un serveur de base de données SQL multi-utilisateurs. Les principaux objectifs de MySQL sont la rapidité, la robustesse et la facilité d'utilisation.

Il puisse gérer des grandes bases de données, sur du matériel bon marché, de manière plus rapide que ce que pouvaient offrir les SGBD commerciaux de l'époque. MySQL est architecturé selon une architecture client/serveur. [15]

La version de MySQL utilisée dans ce PFE est la version 5.1.37-1 ubuntu 9.10

II.7. Conclusion

Nous avons présentées dans ce chapitre les avantages et les caractéristiques de service freeRADIUS qui sera notre système d'authentification pour sécuriser l'accès a notre réseau internet.

Frreradius utilise plusieurs types de protocole d'authentification proposés par EAP.

Administration d'un serveur FreeRADIUS assurant une authentification forte, à l'aide de EAP/TLS, qui sera répartie entre le serveur RADIUS et le point d'accès après avoir généré préalablement les clés et les certificats nécessaires.

Chapitre III

Configuration de serveur freeradius avec Mysql

III.1. Introduction

FreeRADIUS peut fonctionner en s'appuyant uniquement sur des fichiers texte. Nous utiliserons MySQL pour stocker les login et les mots de passe des clients. Dans ce chapitre, nous présentons les différentes étapes de configuration de service d'authentification freeradius avec un serveur de base de données Mysql.

III.2. Pré requis

Afin de faire fonctionner notre serveur, toute une infrastructure autour de lui doit être mise au point.

Nous aurons besoin d'un système d'exploitation et de MySQL (pour gérer nos BDD) doivent être impérativement installé au préalable.

III.3. Installation

L'installation de FreeRADIUS est très simple et classique.

Les paquets à installer sont:

- FreeRADIUS.
- FreeRADIUS-common.
- FreeRADIUS-utils.
- FreeRADIUS-MySQL.
- MySQL-server.

III.4. Configuration de la base de données

Dans un premier temps, on se connecte au serveur MySQL et on crée une nouvelle base de données qui va s'appeler « radius ». Ensuite on va ajouter un utilisateur et un client à la base de données, après avoir importé les différentes tables depuis les SQL que FreeRADIUS propose. [10]

III.4.1.Connexion au serveur MySQL

Pour se connecter à MySQL, il faudra se connecter avec l'utilisateur "root", pour lequel on a défini un mot de passe.

Donc,pour se connecter à MySQL, on tape la commande suivante:

-h: machine hôte.

-u: utilisateur MySQL(pas Unix).

-p: mot de passe MySQL.

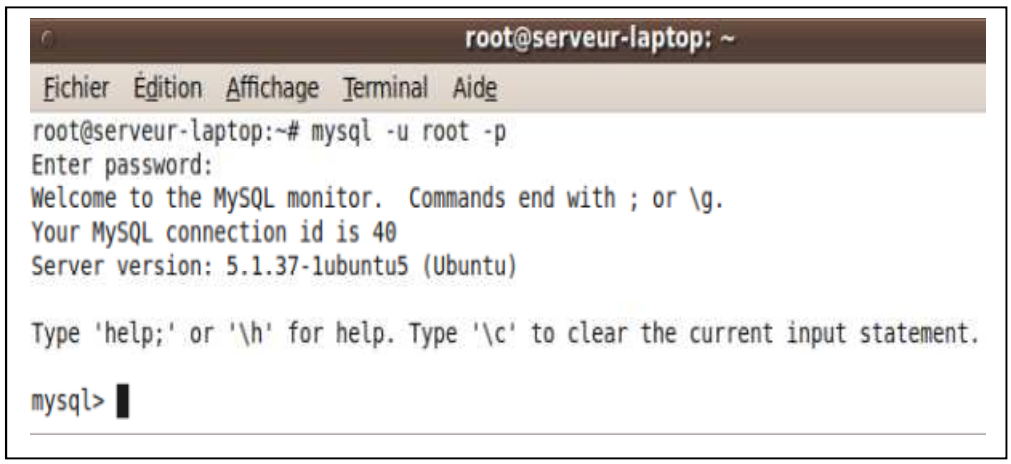


Figure III.1: connexion au serveur MySQL.

III.4.2. Création d'une base de données

Pour créer notre base de données qui sera appelé « radius» il suffit d'utiliser la requête suivante qui est très simple: [16]

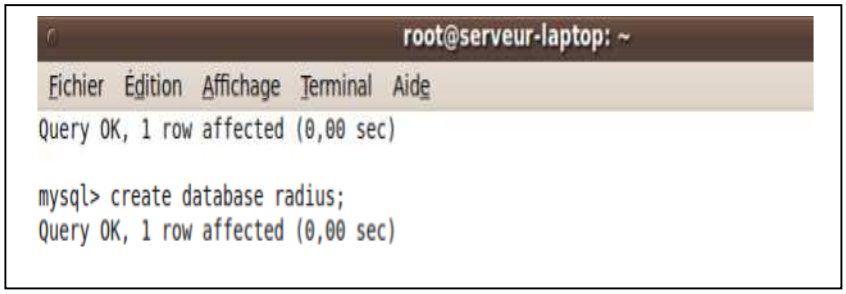


Figure III.2: création d'une BDD «radius».

III.4.3. Création de l'utilisateur RADIUS et attribution des droits

Lors de la création de l'utilisateur MySQL, on lui donne tous les droits sur la base radius. Cet utilisateur est optionnel car on peut aussi utiliser root pour les échanges entre le serveur RADIUS et la base de données. [10]

```
mysql> create user 'radius'@'localhost' identified by 'passer';
Query OK, 0 rows affected (0,02 sec)

mysql> grant all privileges on radius.* to 'radius'@'localhost';
Query OK, 0 rows affected (0,02 sec)
```

Figure III.3: création d'un utilisateur et attribution des droits.

III.4.4. Importation des tables depuis le serveur FreeRADIUS

La commande USE est fournie pour assurer la compatibilité.

```
mysql> use radius;
Database changed
```

Figure III.4: utilisation de la BDD «radius».

Les fichiers de configuration du serveur FreeRADIUS se trouvent dans le répertoire /etc/freeradius: [10]

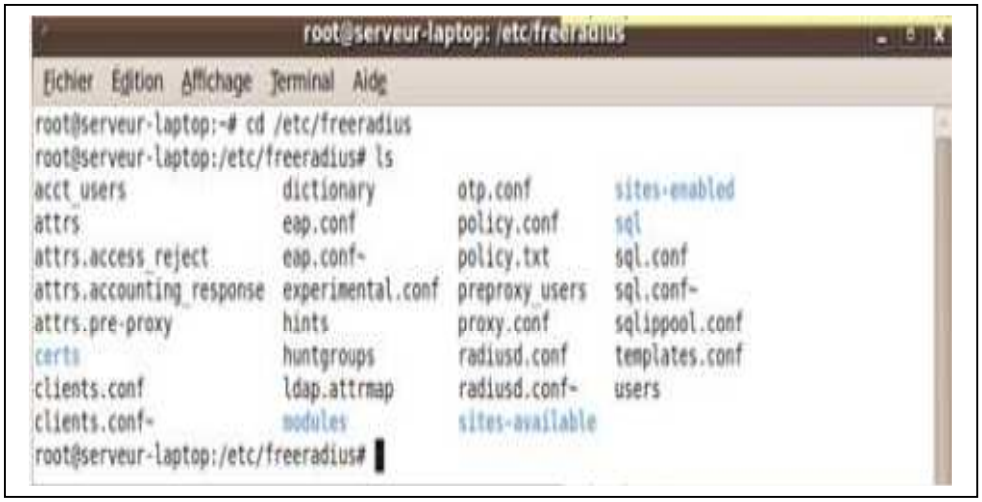


Figure III.5: la liste des fichiers de configuration du seveur FreeRADIUS.

Ce répertoire contient un répertoire /etc/freeradius/sql/mysql qui contient à son tour tous les fichiers SQL qu'on va importer à savoir schema.sql et nas.sql. On va, depuis la base de données, importer les deux fichiers qui vont créer automatiquement les tables dans notre base de données «radius»: [10]

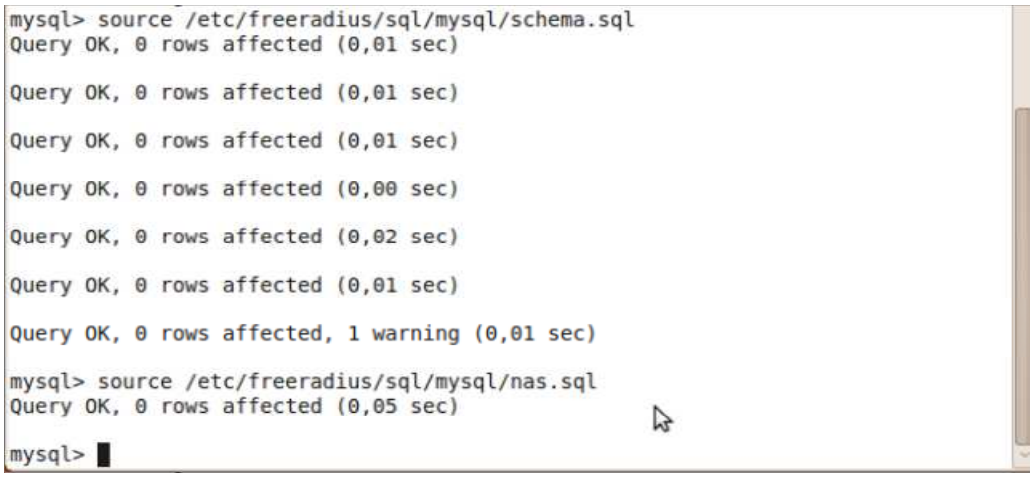


Figure III.6: importation des fichiers schema.sql et nas.sql.

III.4.5. Affichage de la liste de bases de données

La commande show tables; liste les tables disponibles d'une base de donnéesMySQL spécifiée. De plus, par défaut cette commande ne retourne que la colonne du nom des tables. [17]

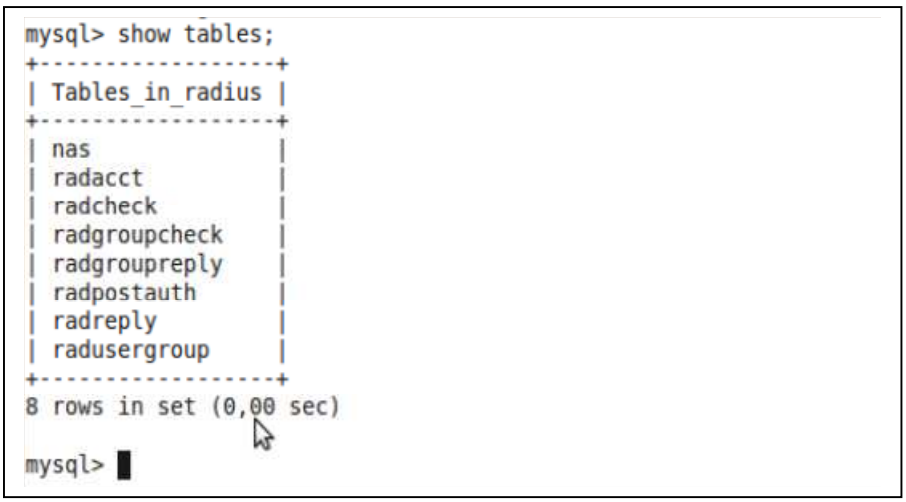


Figure III.7:lister les tables de la BBD «radius».

Une description des différentes tables:

• La table **nas** : contient la liste des NAS, c'est-à-dire des clients RADIUS. Elle remplace le fichier traditionnel clients.conf.

• La table **radacct** : contient les informations qu'un NAS retourne en cas d'accounting.

• La table **radcheck** : permet de vérifier une option d'un utilisateur comme le mot de passe, par exemple, quand on utilise PEAP ou TTL.

• La table **radgroupcheck** : assure la même fonction que radcheck, mais pour une option de groupe.

• La table **radgroupreply** : permet de retourner une option de groupe.

• La table **radpostauth** : contient les informations sur chaque authentification réussie.

• La table **radreply** : permet de retourner une option pour l'utilisateur.

• La table **usergroup** : permet de faire la liaison entre le nom d'utilisateur et son groupe. [10]

Quand on utilise FreeRADIUS avec une base de données, la gestion des utilisateurs est un peu différente: chaque utilisateur est rattaché un groupe. Ce qui fait qu'il y a les options de groupe et les options pour l'utilisateur. [10]

III.4.6. Ajout d'un utilisateur(ou compte utilisateur)

DESCRIBE fournit des informations à propos des colonnes d'une table. Elle fait la liste des colonnes d'une table. [18]

```
mysql> desc radcheck;
+-----------+------------------+------+-----+---------+----------------
| Field     | Type             | Null | Key | Default | Extra
+-----------+------------------+------+-----+---------+----------------
| id        | int(11) unsigned | NO   | PRI | NULL    | auto_increment
| username  | varchar(64)      | NO   | MUL |         |
| attribute | varchar(32)      | NO   |     |         |
| op        | char(2)          | NO   |     | ==      |
| value     | varchar(253)     | NO   |     |         |
+-----------+------------------+------+-----+---------+----------------
5 rows in set (0,00 sec)
```

Figure III.8: liste des colonnes de la table radcheck.

On ajoute le nouvel utilisateur essai avec le type d'authentification que l'utilisateur va utiliser. Dans notre cas, c'est le type MD5, de la sorte:

```
mysql> insert into radcheck (id,username,attribute,op,value) values (NULL,"essai
","MD5-Password",":=",MD5("passer"));
Query OK, 1 row affected (0,03 sec)

mysql>
```

Figure III.9: ajout d'un nouvel utilisateur.

INSERT insère une nouvelle ligne dans une table existante. La forme de INSERT VALUES est basée sur des colonnes explicitement précisée est acceptée avec plusieurs valeurs. [19]

```
mysql> select *from radcheck;
+----+----------+--------------+----+----------------------------------
| id | username | attribute    | op | value
+----+----------+--------------+----+----------------------------------
| 14 | essai    | MD5-Password | := | e7247759c1633c0f9f1485f3690294a9
+----+----------+--------------+----+----------------------------------
1 row in set (0,00 sec)
```

Figure III.10:Affiche le contenu de la table radcheck après insertion d'utilisateur.

III.5. Configuration de FreeRADIUS

On va maintenant s'occuper de la configuration de FreeRADIUS, les fichiers de configuration se trouvent dans /etc/freeradius. On va juste modifier les fichiers suivants:

- radiusd.conf.
- sites-available/default.
- sql.conf.
- eap.conf.

III.5.1. Configuration du fichier radiusd.conf

Ce fichier contient les éléments principaux permettant la configuration de freeRADIUS, ainsi que les modules complémentaires.

Il contient par défaut un tas de lignes (pour la plupart précédées d'un "#"). Ce # signifie que ces lignes sont commentées et donc non-intercepté. Il est donc logique que pour activer les fonctionnalités qui nous intéressent, de décommenter les bonnes lignes (enlever le #).

Il y a dans ce fichier plein de choses que nous pourrions enlever car elles ne nous servent à rien (dans notre cas…). Les lignes suivantes montrent ce qu'il est nécessaire de modifier pour nos besoins.

```
listen {
      type = auth
      ipaddr = *
      port = 0
 }
listen {
      ipaddr = *
      port = 0
      type = acct
}
#$INCLUDE clients.conf
$INCLUDE ${confdir}/modules/
$INCLUDE eap.conf
$INCLUDE sql.conf
$INCLUDE sql/mysql/counter.conf
$INCLUDE sites-enabled/
```

Figure III.11: configuration du fichier radiusd.conf.

III.5.2. Configuration du fichier sites-available/default

Assez peu de choses dans ce fichier, compte tenu de la simplicité de nos besoins. Le module eap doit être présent dans les deux sections Authorize et Authenticate.

```
authorize {
        preprocess
        eap {
                ok = return
        }
        sql
}
authenticate {
        Auth-Type CHAP {
                chap
        }
        eap
}
session {
        sql
}
```

Figure III.12: configuration du fichier sites-available/default.

III.5.3. Configuration du fichier sql.conf

Dans ce fichier on configure l'identifiant et le mot de passe de l'utilisateur de la base de donné MySQL (ici "radius" et "passer").

```
sql {
        #
        #  Set the database to one of:
        #
        #       mysql, mssql, oracle, postgresql
        #
        database = "mysql"

        #
        #  Which FreeRADIUS driver to use.
        #
        driver = "rlm_sql_${database}"

        # Connection info:
        server = "localhost"
        #port = 3306
        login = "root"
        password = "passer"

        # Database table configuration for everything except Oracl
        radius_db = "radius"
```

Figure III.13: configuration du fichier sql.conf.

III.5.4. Configuration du fichier eap.conf

Le fichier eap.conf est inclus dans radiusd.conf au moyen d'une instruction INCLUDE. On y trouve le module eap qui a pour fonction d'implémenter les couches EAP. Ce module est lui-même constitué de sous-modules qui correspondent chacun à un protocole (couche EAP Method). Dans notre cas, on a choisi le protocole MD5.

MD5 C'est une fonction de hachage cryptographique (un algorithme) de calcul d'un hashcode(condensat) du contenu d'un fichier pour chiffrer les mots de passe. Elle permet d'obtenir l'empreinte numérique (hashcode - condensat) d'un fichier. [20]

```
eap {
   default_eap_type = md5
   timer_expire     = 60
   cisco_accounting_username_bug = no
    md5 {
    }
}
```

Figure III.14: configuration du fichier eap.conf.

III.6. Test de fonctionnement en local

La première chose à faire est d'arrêter FreeRADIUS, en utilisant la commande:

Figure III.15: arrêt de FreeRADIUS.

Puis, de le relancer en mode « debug », par la commande :

```
root@serveur-laptop:~# freeradius -X
```

Figure III.16: lancement de FreeRADIUS.

Du côté de la console où on a lancé FreeRADIUS en mode « debug », on a au final:

```
radiusd: #### Opening IP addresses and Ports ####
listen {
        type = "auth"
        ipaddr = *
        port = 0
}
listen {
        type = "acct"
        ipaddr = *
        port = 0
}
Listening on authentication address * port 1812
Listening on accounting address * port 1813
Listening on proxy address * port 1814
Ready to process requests.
```

Figure III.17: FreeRADIUS donne la main pour accepter des requêtes.

Enfin, Dans un autre terminal, on essaie de se connecter en utilisant le compte utilisateur « essai » qu'on a créé précédemment, en utilisant l'outil radtest :

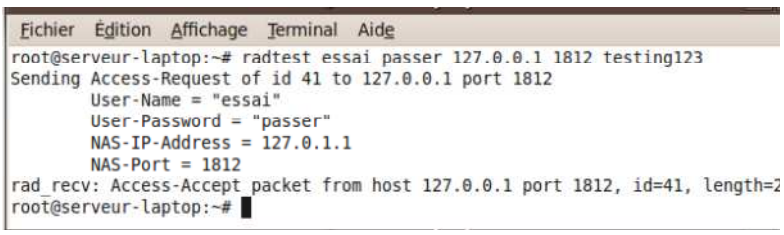


Figure III.18: test de fonctionnement de l'utlisateur essai.

III.7.Conclusion

Dans ce chapitre, nous avons données tous les étapes nécessaires dans la configuration d'un serveur freeradius.

L'objectif de cette configuration est identifié un client à l'aide de Freeradius en utilisant un mode d'authentification basé sur un login et un mot de passe stocké dans une base de données MySQL.

Chapitre IV

Configuration d'un serveur RADIUS sur PfSense

IV.1. Introduction

Dans ce chapitre, nous présentons comment sécurisé l'accès un serveur web sur internet en utilisons un système d'authentification freeradius implémenté sous pfsense.

La création de notre réseau est réalisée d'une façon virtuelle en utilisant un outil de virtualisation virtualbox.

IV.2. Définition de PfSense

PfSense est un routeur / pare-feu opensource basé sur FreeBSD. PfSense peut être installé sur un simple ordinateur personnel comme sur un serveur. Basé sur PF(packetfilter), comme iptables sur GNU/Linux, il est réputé pour sa fiabilité. Après une installation en mode console, il s'administre ensuite simplement depuis une interface web. [21]

Nous avons utilisés dans notre pfe la version 2.2 de pfsense.

IV.3. Définition de FreeBSD

FreeBSD est un système d'exploitation UNIX libre.

L'objectif du projet FreeBSD est de fournir un système qui peut servir à tout, avec le moins de restrictions possibles. Il vise les hautes performances et la simplicité d'utilisation pour l'utilisateur final. Il est l'un des systèmes d'exploitation favoris des fournisseurs de contenu sur le Web. Il fonctionne sur de nombreuses plates-formes. [22]

IV.4. Présentation de PfSense

PfSense est puissante, il est basé sur FreeBSD, mais aussi assez simple d'accès, car il fournit une interface Web pour la configuration, (en plus de l'interface console). Cette interface Web n'est accessible par défaut qu'à partir du LAN. [23]

PfSense est une distribution FreeBSD dédié firewall / routeur.

• Le firewall est basé sur PF. Toute la configuration du système est stockée dans un fichier XML (/cf/conf/config.xml).

• Performances sont liées au matériel.

• L'installation ainsi que la configuration PfSense va se réaliser sur un système d'exploitation de type FreeBSD.

IV.5. Objectif

Passer en revue les principales fonctionnalités de PfSense à travers une analyse détaillée de son interface.

• Apprendre à dimensionner son hardware en fonction de ses besoins.

• Installer et mettre à jour son système sur différents supports.

• S'initier à la pratique de l'outil en présentant les différents modes d'accès envisageables (Web, port série, SSH).

• Procéder aux réglages de base de PfSense (hôte, domaine, serveurs DNS, NTP).

• Manipuler et assigner les interfaces du firewall PfSense.

• Présentation du fichier XML de configuration /cf/conf/config.xml

• Procédure d'urgence: accès SSH, désactivation du firewall, rétablissement de règles opérationnelles.

• Procédure d'installation des paquetages par l'intermédiaire de l'interface graphique.

• Incidence sur le système du déploiement des paquets.

[24]

IV.6. Les avantage de PfSense

Les avantages de PfSense que:

• Il est adapté pour une utilisation en tant que pare-feu et routeur.

• Il comprend toutes les fonctionnalités de pare-feu coûteux commerciales, et plus encore dans de nombreux cas.

• Il peut être installé sur un simple ordinateur personnel comme sur un serveur.

• Il permet d'intégrer de nouveaux services tels que l'installation d'un portail captif, la mise en place d'un VPN, DHCP et bien d'autres.

• Il offre des options de firewalling /routage plus évoluée qu'IPCop. Il permet en autre de réaliser:

• Un portail captif (Lorsqu'un utilisateur ouvre son navigateur internet il est redirigé vers une page lui proposant de s'identifier pour se connecter): solution proposée par les hotspot.

• Un serveur VPN.

• La configuration se fait dans l'interface Web, sans rien toucher à la ligne de commande.

Cela implique une intervention minimum sur les machines sauf pour des maintenances matériels ou de grosse mise à jour qu'il est préférable de faire sur les machines. [21]

IV.7. Pourquoi utiliser PfSense comme un serveur RADIUS?

PfSense fait un excellent hôte pour un serveur RADIUS parce que le service ne nécessite pas beaucoup de ressources système. Le service peut facilement gérer l'authentification pour plusieurs centaines de clients sans compromettre les performances.

Avec le matériel approprié peut facilement être mis à l'échelle pour soutenir des milliers de clients. En effet, PfSense permet également la distance pour fonctionner sur une interface réseau spécifique.

IV.8. Oracle VM VirtualBox

IV.8.1. Virtualisation

VirtualBox ou machine virtuelle est un logiciel de virtualisation de systèmes d'exploitation. En utilisant les ressources matérielles de l'ordinateur (système hôte).

VirtualBox permet la création d'un ou de plusieurs ordinateurs virtuels dans lesquels s'installent d'autres systèmes d'exploitation (systèmes invités).

Les systèmes invités fonctionnent en même temps que le système hôte, mais seul ce dernier a accès directement au véritable matériel de l'ordinateur. Les systèmes invités exploitent du matériel générique, simulé par un « faux ordinateur » (machine virtuelle) créé par VirtualBox. [25]

IV.8.2. Principe de fonctionnement des machines de VirtualBox

VirtualBox propose de virtualiser les systèmes d'exploitation invités sur une machine hôte appelée hyperviseur, l'application supporte les systèmes Windows, Linux, Mac OS X, Solaris, FreeBSD, etc.

Oracle VM VirtualBox, de son nouveau nom, peut créer des machines virtuelles (VM), en exécuter une ou plusieurs en même temps en toute sécurité, en mettre en pause, etc. Il intègre également un accès à distance via protocole HTTP, pratique pour faire des démonstrations sur un système propre. VirtualBox dispose de plusieurs modes de

création de VM à même de satisfaire les utilisateurs experts et guider les novices en matière de virtualisation.

En effet, les systèmes invités n'interagissent pas directement avec le système hôte, et n'interagissent pas entre eux. Le champ d'action des systèmes invités est confiné, limité à leur propre machine virtuelle.

La dernière version de Oracle VM VirtualBox apporte la possibilité d'enrichir l'outil de nouvelles fonctionnalités par le biais d'extensions. Le logiciel améliore également la prise en charge du format OVF(Open Virtualization Format). Celle-ci supporte l'ajout de CPU à chaud et l'accélération vidéo RDP. Dorénavant, VirtualBox sépare les différentes fonctionnalités en paquetages externes. Ainsi, on retrouve un pack pour le support USB 2.0, Serveur RDP et gestionnaire de démarrage PXE. Le célèbre hyperviseur open source dispose également de nouvelles fonctionnalités orientées performances. On relève notamment le "Memory Ballooning" qui consiste à gérer la mémoire de manière dynamique entre chacune des machines virtuelles sur OS 64 bits. Enfin, Oracle VM VirtualBox ajoute la prise en charge des chipsets Intel ICH9 et, surtout, intègre le copier/coller de fichiers de l'hôte vers le client. [26]

IV.8.3. Installation

On peut télécharger Oracle VM VirtualBox à partir de site officiel: https://www.virtualbox.org/wiki/Downloads.

Figure IV.1: site de téléchargement de VirtualBox.

Après le téléchargement, on lance l'installation d'Oracle VM VirtalBox.

Figure IV.2: lancement d'installation de VM VirtualBox.

Figure IV.3: début d'installation de VM VirtualBox.

On continue à cliquer sur next jusqu'à la fin d'installation.

Figure IV.4: fin d'installation de VM VirtualBox.

IV.8.4. Création d'une nouvelle machine virtuelle dans VirtualBox

Pour créer une nouvelle machine virtuelle, on doit démarrer VirtualBox sur l'hôte où on a installé VirtualBox, sélectionner successivement les menus Applications sur le bureau, puis outils système et cliquer sur Oracle VM VirtualBox**.** On peut également exécuter la commande VirtualBox sur un terminal.

Figure IV.5: la fenêtre obtenue au premier démarrage du VM VirtualBox.

Nous devons commencer par créer une nouvelle machine virtuelle, en cliquant sur le bouton « new» en haut à gauche.

On va créer trois machines virtuelles.

Dans un premier temps, on donne un nom aux différentes machines virtuelles, en indiquant le type de système d'exploitation qui sera installé dans la machine virtuelle.

• **Serveur:**Ubuntu 9.10

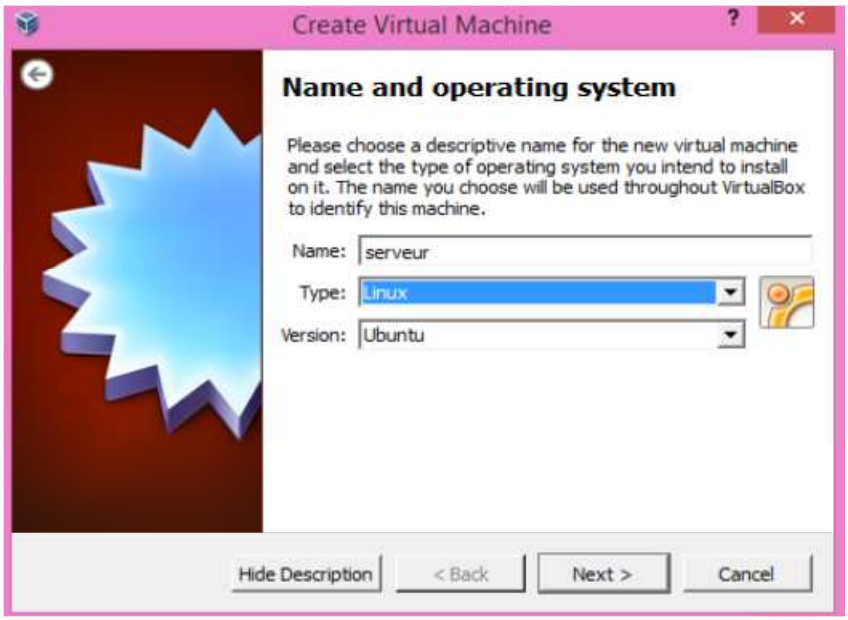


Figure IV.6: attribution d'un nom à la machine virtuelle «serveur».

• **Client:**Ubuntu 12.04

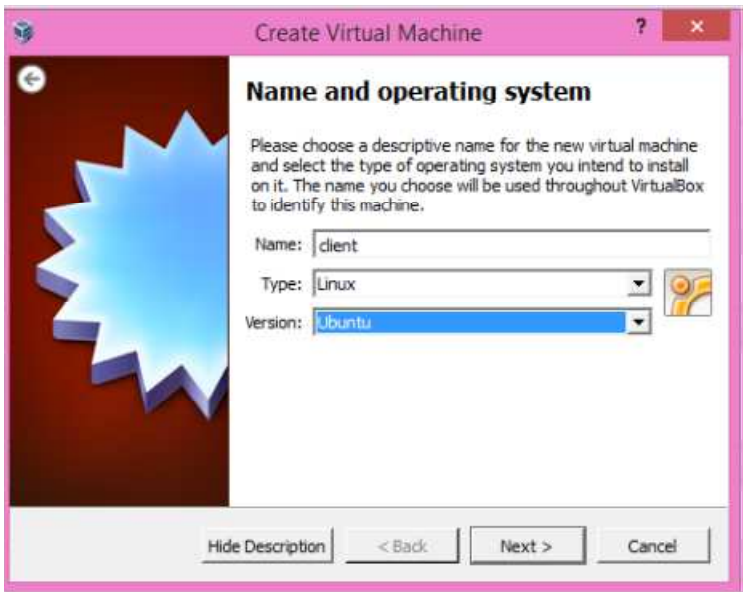


Figure IV.7: attribution d'un nom à la machine virtuelle «client».

• **Gateway:**PfSense 2.2.2

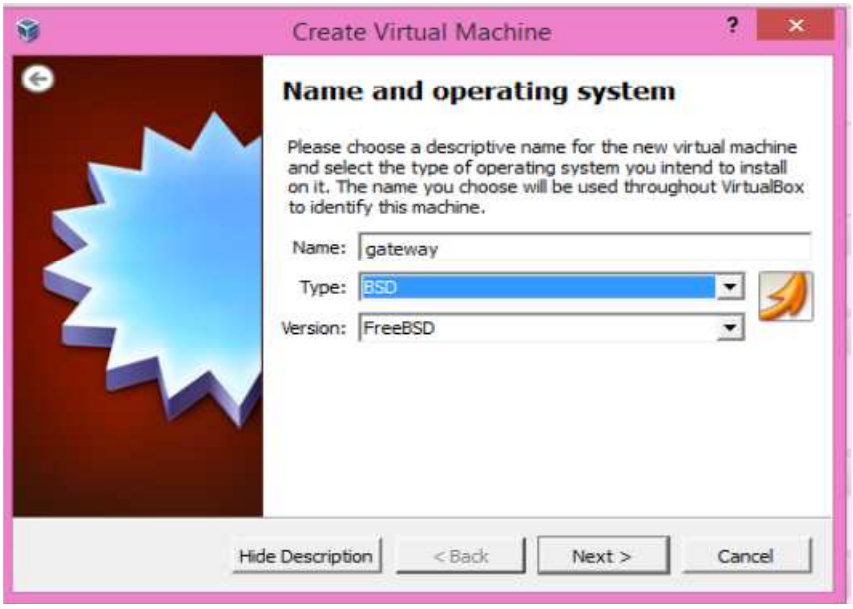


Figure IV.8: attribution d'un nom à la machine virtuelle «gateway».

Les étapes suivantes sont identiques pour la création des trois machines virtuelles:
Nous devons ensuite indiquer quelle quantité de mémoire vive (RAM) nous souhaitons réserver à la machine virtuelle.

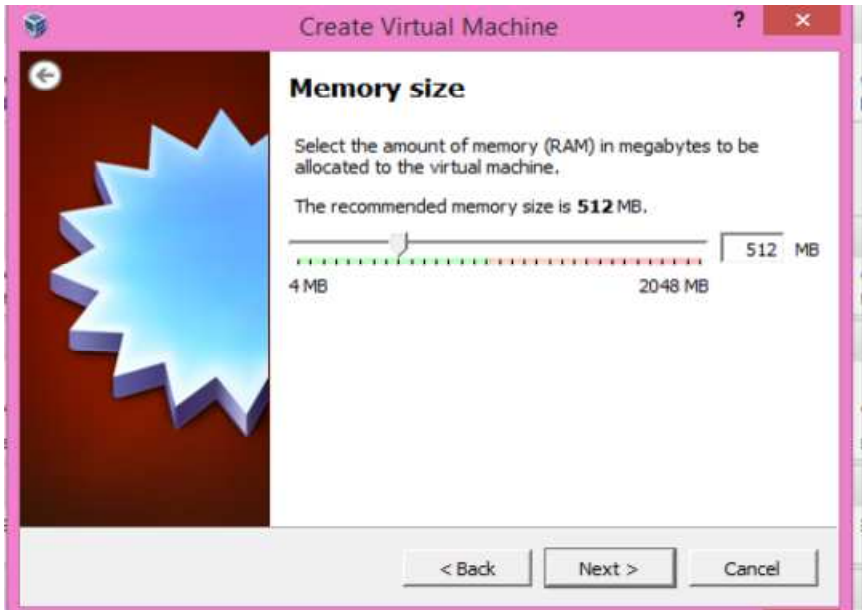


Figure IV.9: la taille mémoire réservée à chaque machine virtuelle.

Il nous reste maintenant à créer le disque dur de la machine virtuelle. VirtualBox va créer une sorte de gros fichier sur notre disque qui représentera le disque dur de la machine. Laissons l'option « Créer un nouveau disque dur » sélectionnée.

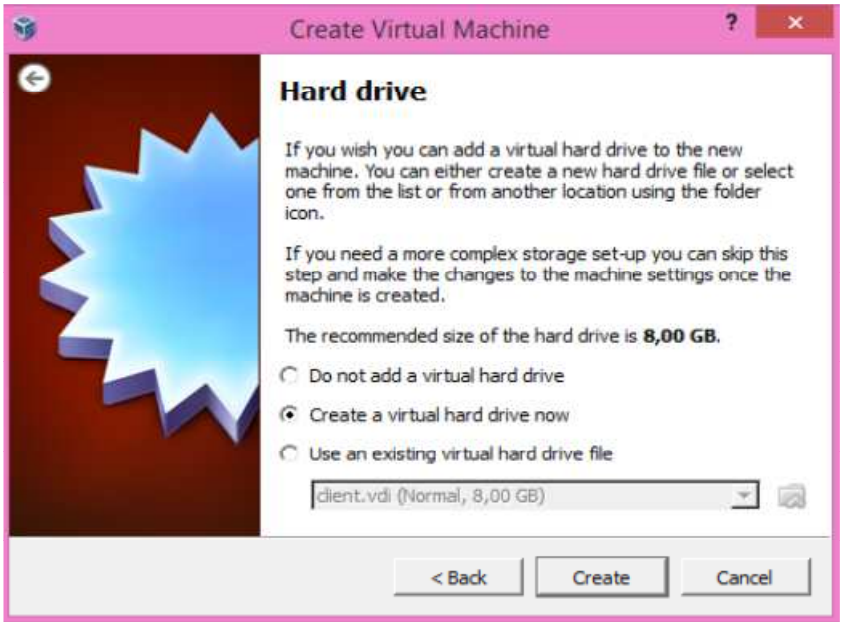


Figure IV.10:création de disque dur virtuel.

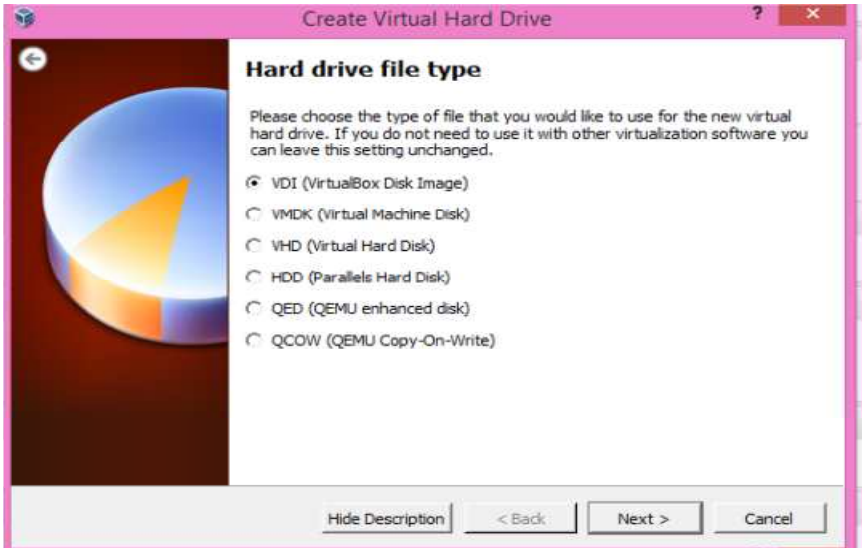


Figure IV.11:choix de type du disque dur virtuel crée.

L'assistant de création de disque dur virtuel nous demande quel type d'image disque nous souhaitons créer. Deux choix s'offrent:

- **Image de taille variable**: le fichier « image » représentant le disque dur virtuel grossira en fonction de l'utilisation du disque dur. C'est l'option recommandée: si le disque virtuel a une taille totale de 8 Go et que seulement 2 Go sont utilisés, le fichier fera 2 Go. [27]
- **Image de taille fixe**: le fichier « image » occupera immédiatement la place maximale. Si le disque virtuel a une taille totale de 8 Go et que seulement 2 Go sont utilisés, le fichier fera tout de même 8 Go. [27]

Figure IV.12:choix de type d'image disque crée.

Nous devrons donner un nom au disque dur virtuel ainsi qu'une taille maximale. Nous recommandons de laisser le nom par défaut (« serveur» ou « client» ou « gateway») et d'indiquer au moins 8 Go.

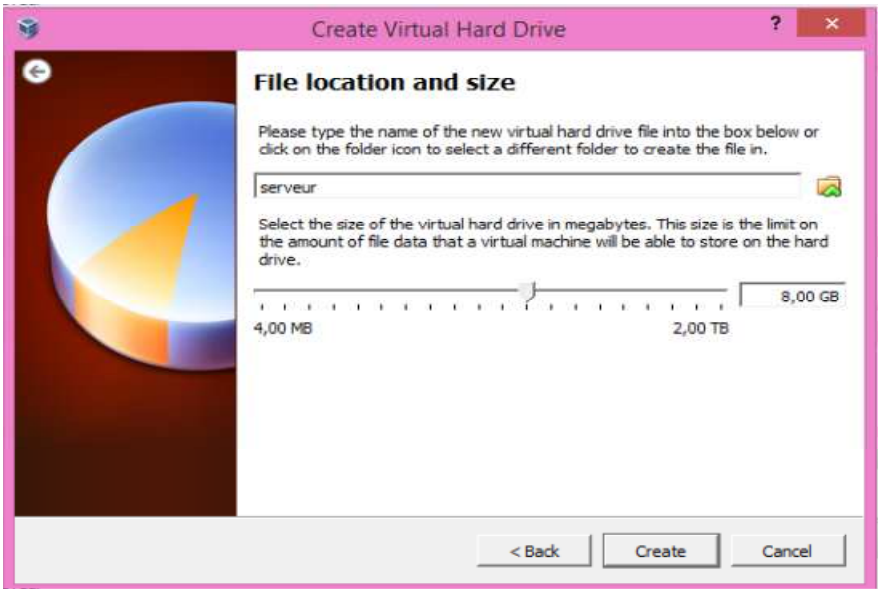


Figure IV.13:choix de nom et taille du disque dur virtuel crée.

L'écran d'accueil de VirtualBox devrait maintenant afficher les machines nommées « serveur», « client » et « gateway » dans la liste de gauche.

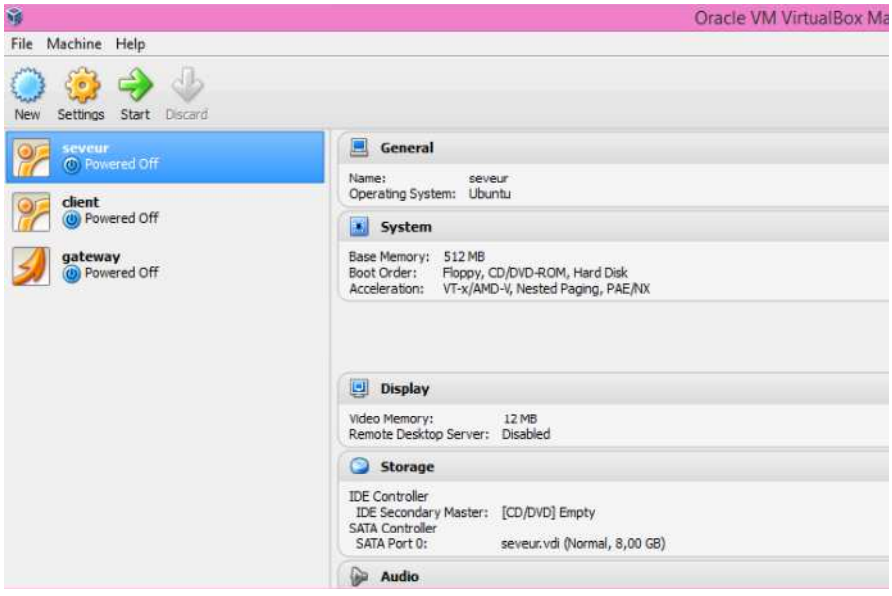


Figure IV.14:écran d'accueil de VirtualBox.

Avant de lancer la machine virtuelle, nous avons besoin du CD d'installation, exactement comme si nous démarrions notre ordinateur pour installer un système d'exploitation. Deux choix s'offrent: [27]

A partir d'un CD gravé: il suffit d'insérer le CD dans le lecteur avant de lancer la machine virtuelle. Il s'agit du cas le plus simple. [27]

A partir d'image (`.iso`) téléchargée mais n'est pas gravée sur CD: inutile d'utiliser un CD pour cela, VirtualBox est capable de lire directement l'image ISO. [27]

Figure IV.15: images ISO des trois machines virtuelles.

Pour lancer la machine, cliquer sur son nom dans la liste à gauche puis sur le bouton « Démarrer », en haut ou aussi double-cliques sur le nom de la machine.

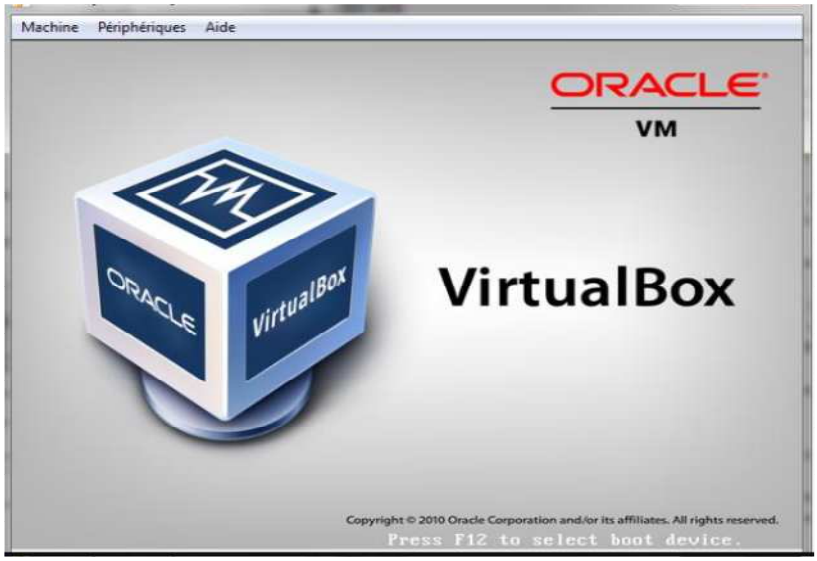


Figure IV.16: fenêtre représente la machine virtuelle s'ouvre.

IV.8.5. Installation de PfSense

La version utilisée dans ce mémoire est 2.2.2 de PfSense.

Tout d'abord, il faut se rendre sur le site http://www.pfsense.org afin de récupérer les images ISO à graver sur CDs pour installer PfSense:

pfSense-LiveCD-2.2.2-RELEASE-i386

Puisqu'on a déjà gravée l'image, on boot sur le CD et on arrive aux menus suivants:

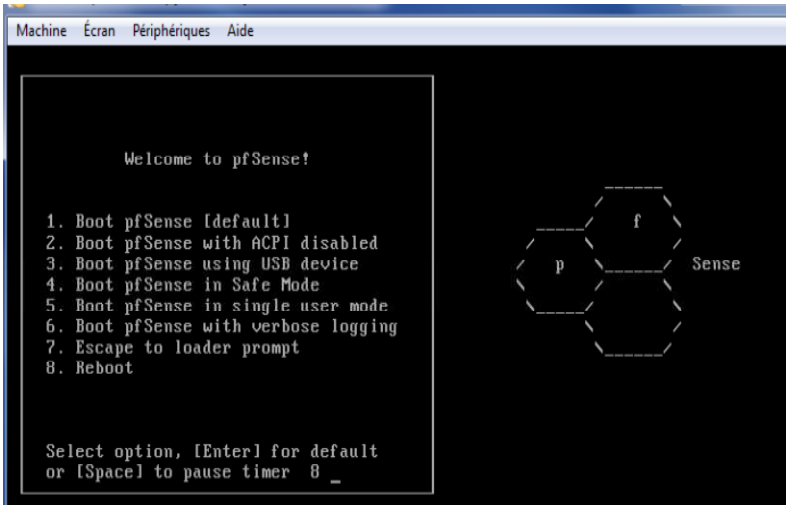


Figure IV.17: écran d'accueil pour installation du PfSense.

On presse « Entrée » pour démarrer sur l'option par défaut (1).

L'installation va se poursuivre un moment, avec un défilement de commandes, jusqu'à s'arrêter sur l'écran suivant: [28]

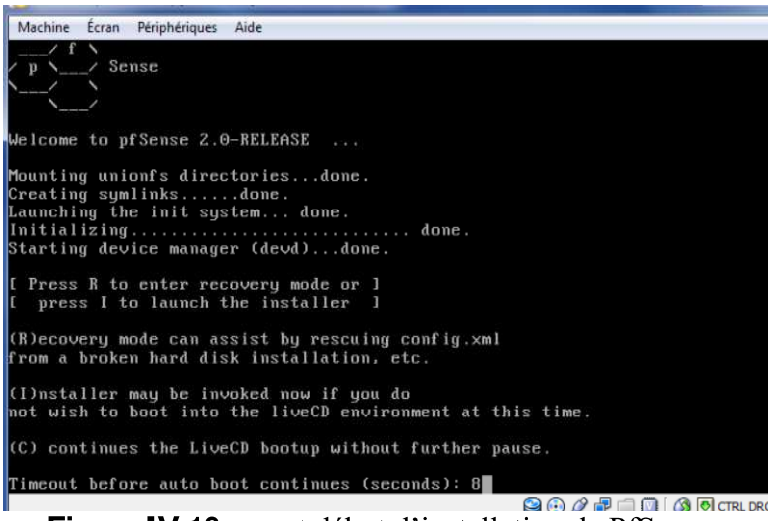


Figure IV.18: avant début d'installation de PfSense.

On tape la lettre I pour lancer l'installation.

On accepte les choix en descendant sur « Accepte these Settings » puis en validant. [28]

Figure IV.19: paramétrage de la console de PfSense.

On opte pour une installation facile puis on confirme à l'écran suivant: [28]

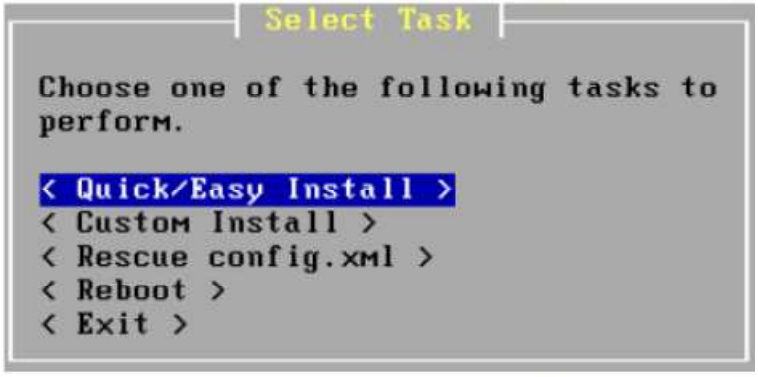


Figure IV.20: type d'installation de PfSense.

Le choix suivant se porte sur le type de processeur: on laisse le choix par défaut puis on valide.

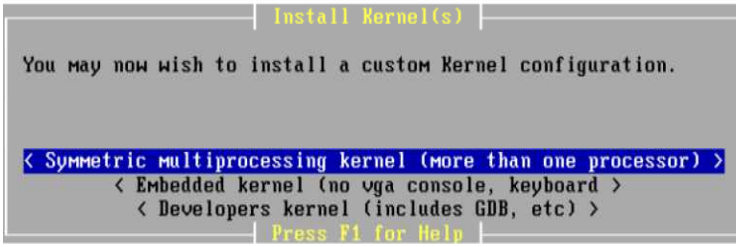


Figure IV.21: type de processeur utilisé.

L'installation terminée.

Si l'installation s'est bien déroulée, la machine démarre sur le nouveau système, et on doit obtenir cet écran: [28]

Figure IV.22: démarrage de machine sur nouveau

On est donc devant l'écran suivant:

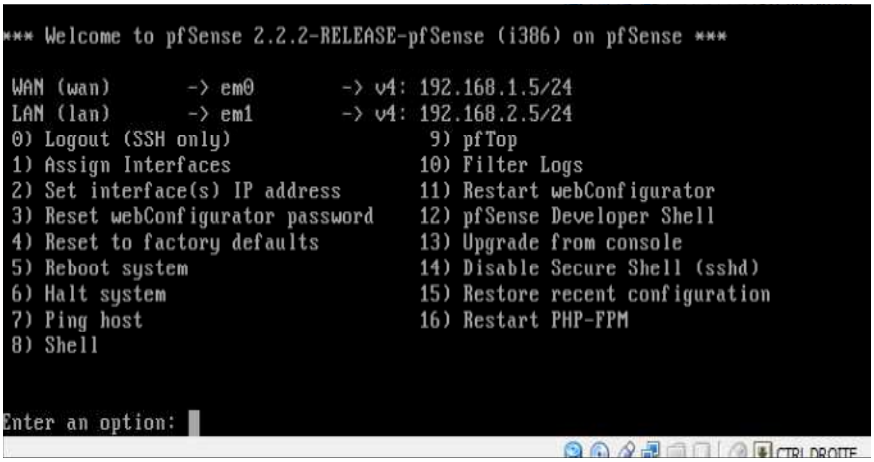


Figure IV.23: menu de PfSense

IV.9. Architecture de notre réseau

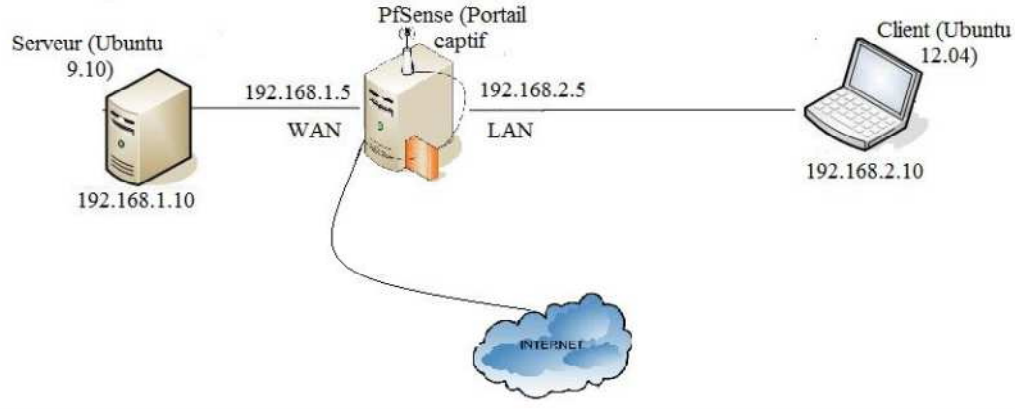


Figure IV.24: architecture serveur/client utilisée.

IV.9.1. Configuration des machines

Après installation des trois machines on va commencer par la configuration des interfaces réseaux:

• La machine virtuelle PfSense: routeur ou passerelle entre le serveur et le client.

Dans la machine virtuelle créer deux cartes réseaux virtuel.

Aller dans la configuration réseau:

Carte réseau1: choisir accès par pont (em0).

Carte réseau2: choisir réseau interne (em1).

Après le démarrage de la machine, il va désormais falloir effectuer quelques paramétrages afin de pouvoir accéder au pare-feu depuis son interface Web (la quasi-totalité de la configuration se fait via navigateur Web).

Le temps de chargement des divers paramètres du système d'exploitation peut être long, mais on arrive ensuite à l'écran suivant:

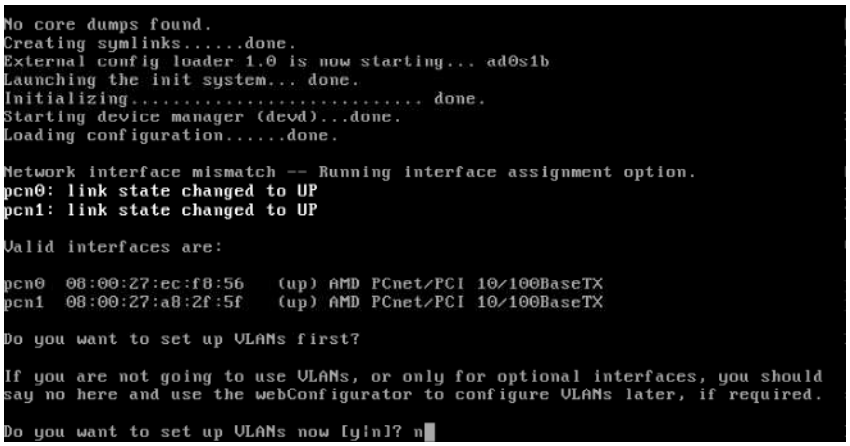


Figure IV.25: demande s'il y a des VLANs à configurer.

Ici, répondre non (à adapter selon le cas bien évidemment).

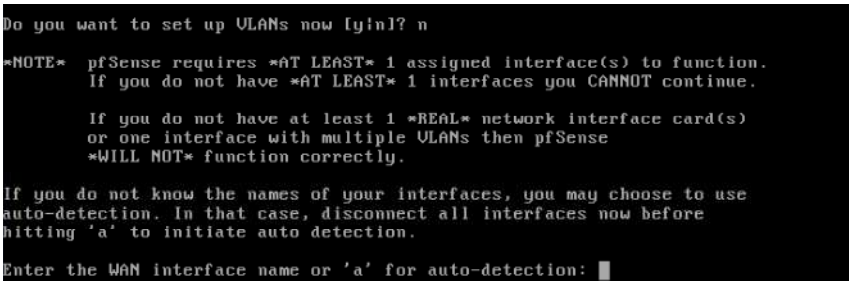


Figure IV.26: demande de saisir une interface WAN.

On tape « a »pour déclencher la détection automatique. Puis, on valide par «Entrée ».On Répète l'opération pour le LAN (et les éventuels autres).

Une fois la configuration des cartes effectuées, on arrive à un écran qui récapitule les différentes cartes réseau et leur association: [28]

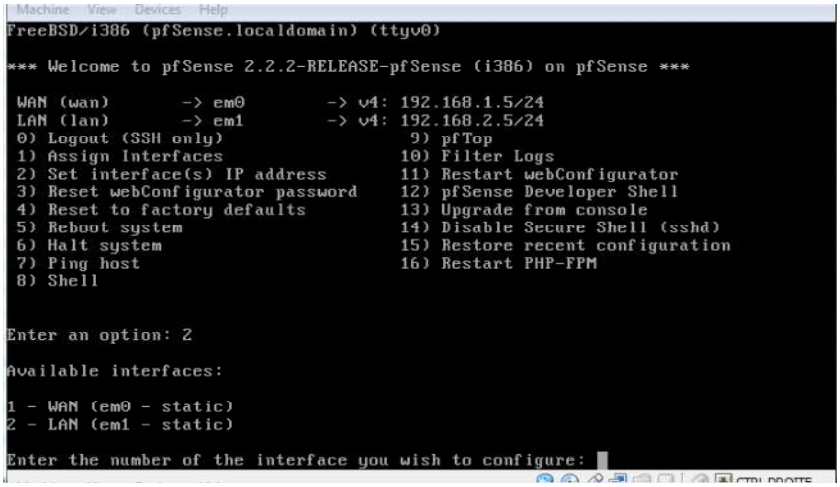


Figure IV.27: configuration des cartes effectuées.

On tape « y » et on valide par « Entrée »

A ce moment-là, PfSense est accessible via son interface Web sur l'adresse IP 192.168.2.5. On va la modifier pour l'intégrer à notre réseau et poursuivre la configuration.

Pour la machine virtuelle serveur:

Aller dans la configuration réseau:

Carte réseau1: choisir réseau interne.

Ensuite, on va démarrer la machine virtuelle et configurer l'adresse IP de notre machine:

Aller vers le fichier /etc/network/interfaces est ajouter les lignes suivantes:

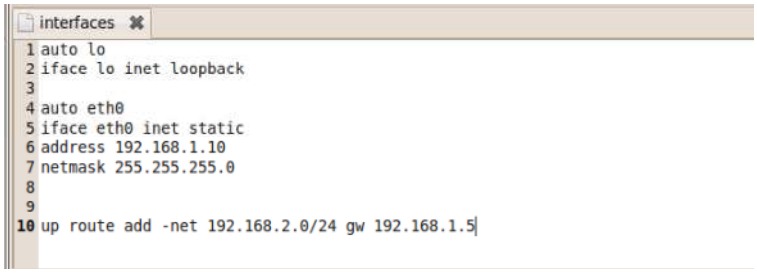


Figure IV.28:configuration d'interface du serveur.

Activation la propriété de routage de la passerelle:

Aller vers le chemin /etc/sysctl.conf est activer la ligne suivante (enlever la #):

net.ipv4.ip_forward=1.

• Pour la machine virtuelle client:

On suit les mêmes étapes que le serveur.

interfaces

```
1 auto lo
2 iface lo inet loopback
3 auto eth0
4 iface eth0 inet static
5 address 192.168.2.10
6 netmask 255.255.255.0
7
8 up route add -net 192.168.1.0/24 gw 192.168.2.5
```

Figure IV.29:configuration d'interface du client.

IV.9.2. Teste de la configuration entre les trois machines

• Gateway:

```
Enter an option: 8

[2.2.2-RELEASE][root@pfSense.localdomain]/root: ping 192.168.1.10
PING 192.168.1.10 (192.168.1.10): 56 data bytes
64 bytes from 192.168.1.10: icmp_seq=0 ttl=64 time=12.768 ms
64 bytes from 192.168.1.10: icmp_seq=1 ttl=64 time=3.238 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=64 time=1.804 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=64 time=1.527 ms
64 bytes from 192.168.1.10: icmp_seq=4 ttl=64 time=1.656 ms
^C
--- 192.168.1.10 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.527/4.199/12.768/4.329 ms
[2.2.2-RELEASE][root@pfSense.localdomain]/root:
```

Figure IV.30: teste ping 192.168.1.10

```
[2.2.2-RELEASE][root@pfSense.localdomain]/root: ping 192.168.2.10
PING 192.168.2.10 (192.168.2.10): 56 data bytes
64 bytes from 192.168.2.10: icmp_seq=0 ttl=64 time=2.796 ms
64 bytes from 192.168.2.10: icmp_seq=1 ttl=64 time=1.883 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=64 time=3.933 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=64 time=2.006 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=64 time=4.004 ms
^C
--- 192.168.2.10 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.883/2.924/4.004/0.909 ms
[2.2.2-RELEASE][root@pfSense.localdomain]/root:
```

Figure IV.31: teste ping 192.168.2.10

• **Serveur:**

root@serveur-laptop: ~

Fichier Édition Affichage Terminal Aide

```
root@serveur-laptop:~# ping 192.168.1.5
PING 192.168.1.5 (192.168.1.5) 56(84) bytes of data.
64 bytes from 192.168.1.5: icmp_seq=1 ttl=64 time=1.69 ms
64 bytes from 192.168.1.5: icmp_seq=2 ttl=64 time=0.576 ms
64 bytes from 192.168.1.5: icmp_seq=3 ttl=64 time=3.03 ms
64 bytes from 192.168.1.5: icmp_seq=4 ttl=64 time=0.600 ms
64 bytes from 192.168.1.5: icmp_seq=5 ttl=64 time=0.820 ms
64 bytes from 192.168.1.5: icmp_seq=6 ttl=64 time=1.37 ms
64 bytes from 192.168.1.5: icmp_seq=7 ttl=64 time=1.13 ms
^C
--- 192.168.1.5 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6014ms
rtt min/avg/max/mdev = 0.576/1.320/3.034/0.796 ms
root@serveur-laptop:~#
```

Figure IV.32: teste ping 192.168.1.5

```
root@serveur-laptop:~# ping 192.168.2.5
PING 192.168.2.5 (192.168.2.5) 56(84) bytes of data.
64 bytes from 192.168.2.5: icmp_seq=1 ttl=64 time=1.66 ms
64 bytes from 192.168.2.5: icmp_seq=2 ttl=64 time=1.50 ms
64 bytes from 192.168.2.5: icmp_seq=3 ttl=64 time=2.38 ms
64 bytes from 192.168.2.5: icmp_seq=4 ttl=64 time=1.47 ms
64 bytes from 192.168.2.5: icmp_seq=5 ttl=64 time=1.10 ms
^C64 bytes from 192.168.2.5: icmp_seq=6 ttl=64 time=0.776 ms
64 bytes from 192.168.2.5: icmp_seq=7 ttl=64 time=1.19 ms
64 bytes from 192.168.2.5: icmp_seq=8 ttl=64 time=1.39 ms
^C
--- 192.168.2.5 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7019ms
rtt min/avg/max/mdev = 0.776/1.438/2.384/0.442 ms
root@serveur-laptop:~#
```

Figure IV.33: teste ping 192.168.2.5

```
root@serveur-laptop: ~
Fichier  Édition  Affichage  Terminal  Aide
root@serveur-laptop:~# ping 192.168.2.10
PING 192.168.2.10 (192.168.2.10) 56(84) bytes of data.
64 bytes from 192.168.2.10: icmp_seq=1 ttl=63 time=8587 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=63 time=7578 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=63 time=6569 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=63 time=5562 ms
64 bytes from 192.168.2.10: icmp_seq=5 ttl=63 time=4553 ms
64 bytes from 192.168.2.10: icmp_seq=6 ttl=63 time=3546 ms
64 bytes from 192.168.2.10: icmp_seq=7 ttl=63 time=2538 ms
64 bytes from 192.168.2.10: icmp_seq=8 ttl=63 time=1529 ms
64 bytes from 192.168.2.10: icmp_seq=9 ttl=63 time=523 ms
64 bytes from 192.168.2.10: icmp_seq=10 ttl=63 time=1.36 ms
64 bytes from 192.168.2.10: icmp_seq=11 ttl=63 time=1.53 ms
64 bytes from 192.168.2.10: icmp_seq=12 ttl=63 time=2.51 ms
64 bytes from 192.168.2.10: icmp_seq=13 ttl=63 time=1.67 ms
64 bytes from 192.168.2.10: icmp_seq=14 ttl=63 time=3.66 ms
64 bytes from 192.168.2.10: icmp_seq=15 ttl=63 time=4.19 ms
^C
--- 192.168.2.10 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14083ms
rtt min/avg/max/mdev = 1.363/2733.678/8587.176/3006.190 ms, pipe 9
root@serveur-laptop:~#
```

Figure IV.34: teste ping 192.168.2.10

- **Client :**

```
root@bouzi:~# ping 192.168.2.5
PING 192.168.2.5 (192.168.2.5) 56(84) bytes of data.
64 bytes from 192.168.2.5: icmp_req=1 ttl=64 time=3418 ms
64 bytes from 192.168.2.5: icmp_req=2 ttl=64 time=2414 ms
64 bytes from 192.168.2.5: icmp_req=3 ttl=64 time=1414 ms
64 bytes from 192.168.2.5: icmp_req=4 ttl=64 time=413 ms
64 bytes from 192.168.2.5: icmp_req=5 ttl=64 time=0.871 ms
64 bytes from 192.168.2.5: icmp_req=6 ttl=64 time=0.821 ms
64 bytes from 192.168.2.5: icmp_req=7 ttl=64 time=1.65 ms
64 bytes from 192.168.2.5: icmp_req=8 ttl=64 time=1.50 ms
64 bytes from 192.168.2.5: icmp_req=9 ttl=64 time=1.76 ms
64 bytes from 192.168.2.5: icmp_req=10 ttl=64 time=1.23 ms
^C
--- 192.168.2.5 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9045ms
rtt min/avg/max/mdev = 0.821/766.828/3418.897/1175.041 ms, pipe 4
root@bouzi:~#
```

Figure IV.35: teste ping 192.168.2.5

```
root@bouzi: ~
root@bouzi:~# ping 192.168.1.5
PING 192.168.1.5 (192.168.1.5) 56(84) bytes of data.
64 bytes from 192.168.1.5: icmp_req=1 ttl=64 time=3074 ms
64 bytes from 192.168.1.5: icmp_req=2 ttl=64 time=2075 ms
64 bytes from 192.168.1.5: icmp_req=3 ttl=64 time=1075 ms
64 bytes from 192.168.1.5: icmp_req=4 ttl=64 time=75.7 ms
64 bytes from 192.168.1.5: icmp_req=5 ttl=64 time=1.48 ms
64 bytes from 192.168.1.5: icmp_req=6 ttl=64 time=1.25 ms
64 bytes from 192.168.1.5: icmp_req=7 ttl=64 time=0.599 ms
64 bytes from 192.168.1.5: icmp_req=8 ttl=64 time=1.34 ms
64 bytes from 192.168.1.5: icmp_req=9 ttl=64 time=2.57 ms
^C
--- 192.168.1.5 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8017ms
rtt min/avg/max/mdev = 0.599/701.036/3074.759/1080.257 ms, pipe 4
root@bouzi:~#
```

Figure IV.36: teste ping 192.168.1.5

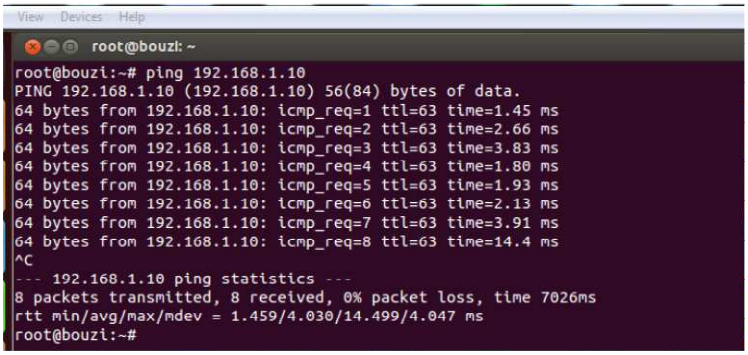


Figure IV.37: teste ping 192.168.1.10

IV.10. Configuration de PfSense

On va configurer une machine sous PfSense pour agir en tant que portail captif pour une LAN. Cette passerelle interface une connexion internet vers la LAN.

L'authentification se fait en FreeRadius (hébergé sur PfSense), qui lui-même interroge une base de données MySQL.

IV.10.1. Premiers paramétrages de PfSense

Maintenant, la configuration se fera depuis le client via un navigateur Web.

Lançons un navigateur Web, et dans la barre d'adresse, saisissons http://192.168.2.5/ puis validons par « Entrée ». On arrive sur la page d'identification suivante: [28]

Figure IV.38: page d'identification PfSense.

Connectons avec les identifiants suivants:

Login: admin et mot de passe: pfsense.

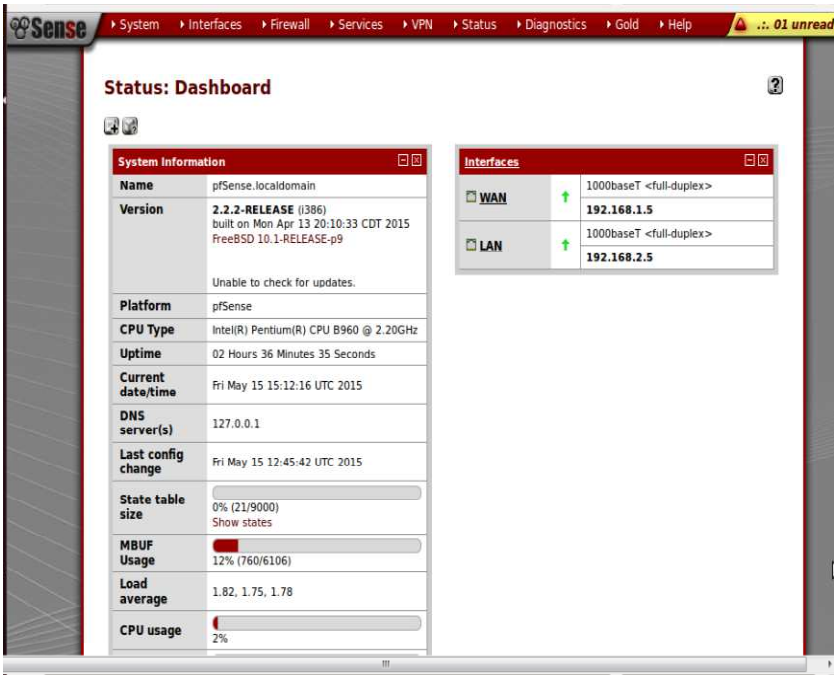


Figure IV.39: tableau de bord de PfSense.

Allons ensuite dans System, puis General Setup.

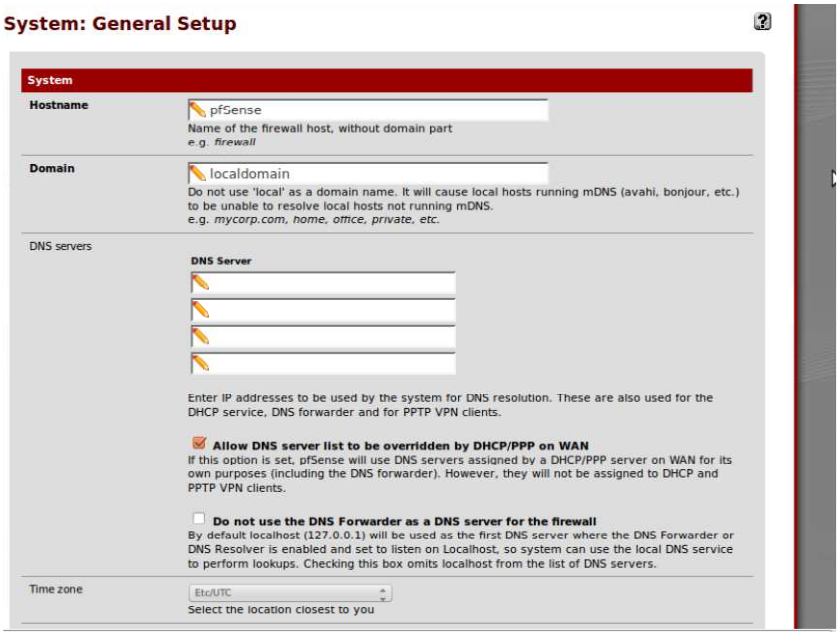


Figure IV.40: configuration générale de PfSense.

Ici se trouve la configuration générale de PfSense. Entrons ici le nom de la machine, le domaine et l'IP du DNS. il vous faut décocher l'option se trouvant dessous (Allow DNS server list to beoverridden by DHCP/PPP on WAN).

En effet, cette option provoque des conflits puisque les DNS des clients n'est plus PfSense, mais un DNS du WAN inaccessible par le LAN.

Ensuite, dans "system", allons dans Advanced. Ici, nous pouvons activer la connexion SSH afin de l'administrer à distance sans passer par l'interface graphique(en effet, pour une configuration accrus, il vaut mieux passer par le Shell). [29]

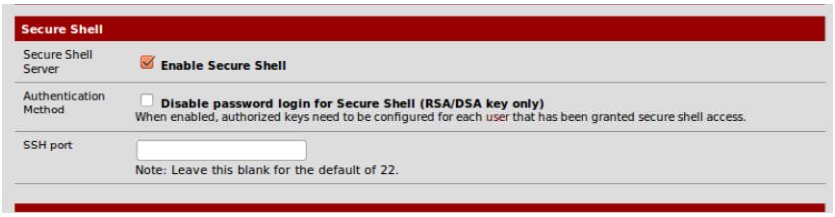


Figure IV.41: activation de connexion SSH.

IV.10.2. Installer le paquet FreeRADIUS

Le gestionnaire de paquets PfSense comprend freeradius2 que des options d'installation qu'on va l'utiliser car il a quelques fonctionnalités supplémentaires ne sont pas présents dans la version précédente.

Le paquet interrompre brièvement le trafic passant par le routeur que le service commence donc soyons prudent lorsqu'on installe sur un système de production.

Ouvrons le gestionnaire de paquets dans le menu du système de l'interface Web. [2]

Figure IV.42:le menu du système de l'interface Web.

Cliquonssur le symbole à côté de freeradius2 pour démarrer l'installation. Ensuite, cliquons sur «OK» pour confirmer l'installation du paquet.

Le processus d'installation va automatiquement télécharger et installer le paquet freeradius2 avec toute cette dépendance. L'installation ne prend que quelques minutes à remplir. [2]

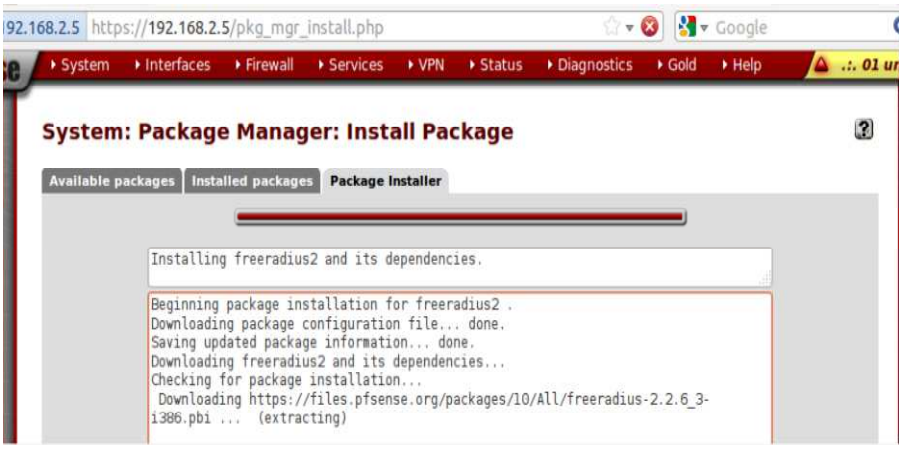


Figure IV.43: installation de freeradius2.

Après cet est fini, il y aura un nouvel élément de menu pour le paquet dans le menu des services. [2]

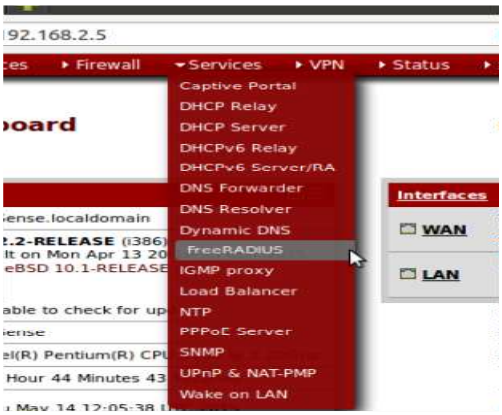


Figure IV.44:nouvel élément FreeRADIUS.

IV.10.3. Configuration des interfaces LAN et WAN

Nous allons maintenant configurer les interfaces LAN et WAN en détail.

Pour cela, allons dans Interface. Ensuite, cliquons sur le signe plus pour ajouter une nouvelle interface. Puis LAN pour commencer. Entrons ici l'adresse IP de la carte réseau coté LAN, ainsi que l'adresse IP de la passerelle. Enfin, cliquons sur Enregistrer. Le reste des paramètres peut être laissé avec les paramètres par défaut. [29]

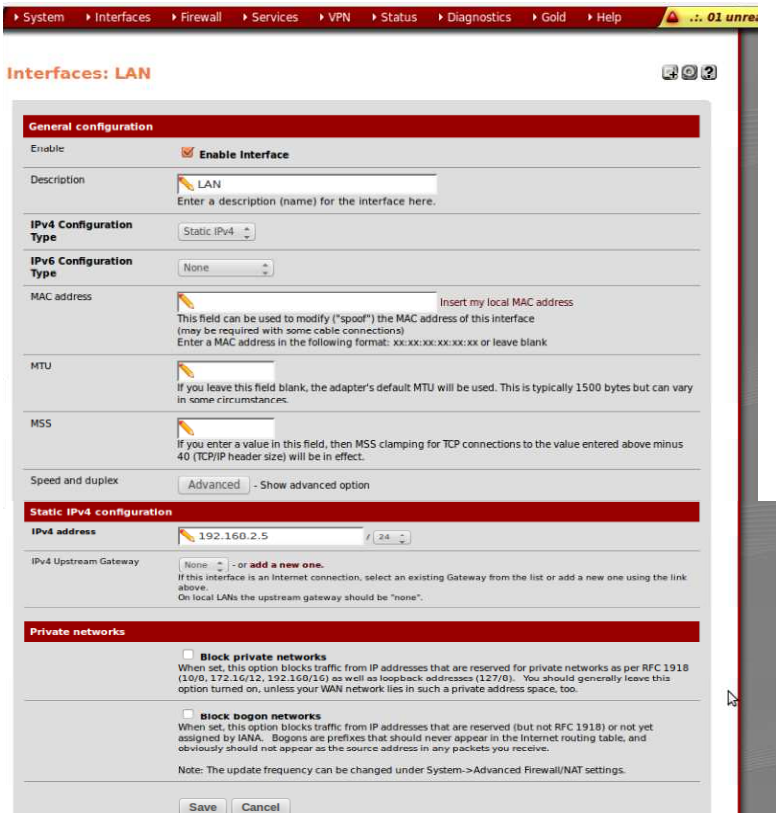


Figure IV.45: configuration de l'interface LAN.

Même méthode pour configurer l'interface WAN.

Figure IV.46: configuration de l'interface WAN.

IV.10.4. Paramétrer les règles de base

L'interface de gestion des règles est joignable de Firewall >rules.

Comme nous voyons, il est possible de définir des règles pour l'interface LAN ainsi que l'interface WAN. [21]

• Pour ajouter une règle.

• Pour modifier une règle.

• Pour supprimer une règle.

On va ajouter deux règles WAN:

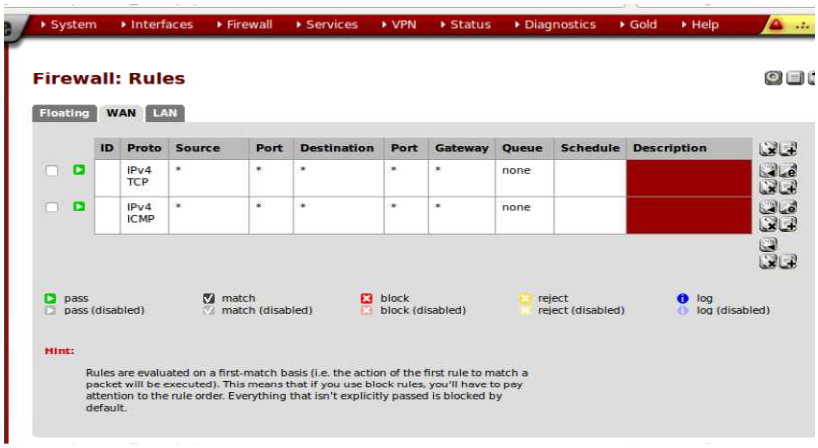


Figure IV.47: règles ajoutées sur l'interface WAN.

Concernant les deux règles par défaut du LAN, il ne faut surtout pas désactiver la règle « Anti-Lockout Rule», qui permet de se connecter à l'interface Web de PfSense via un autre PC (sous peine de devoir reconfigurer voir réinstaller Pfsense).

Par contre, la seconde règle est celle qui autorise tout le trafic. Il faut donc soit la désactiver, soit la supprimer (on va juste désactivée car au besoin, pour juste la réactiver pour avoir de nouveau accès à Internet rapidement). [28]

Et on va ajouté les règles suivantes sur LAN:

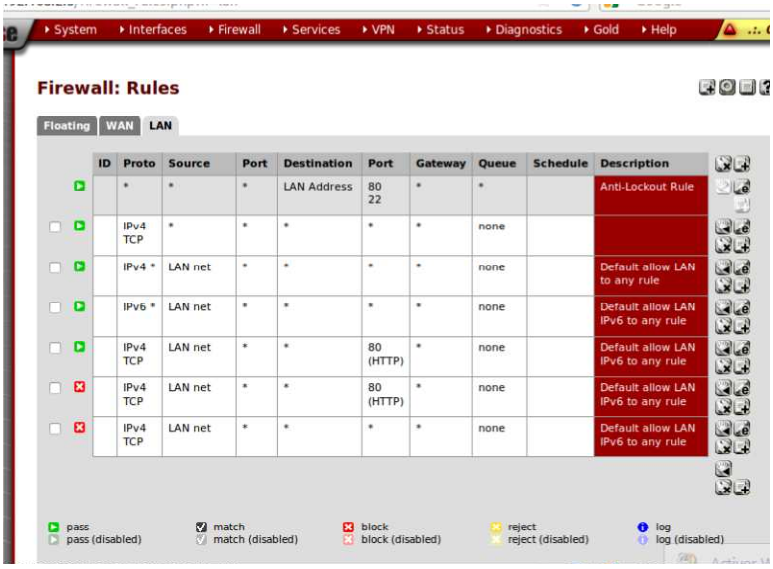


Figure IV.48: règles ajoutées sur l'interface LAN.

IV.10.5. Configuration une interface FreeRADIUS

La première chose que nous devons faire est de spécifier un ou plusieurs interfaces pour le serveur RADIUS. Les paramètres de configuration pour FreeRADIUS sont situés sous le menu des services. [2]

Dans la plupart des cas, on associe le service à l'interface LAN.

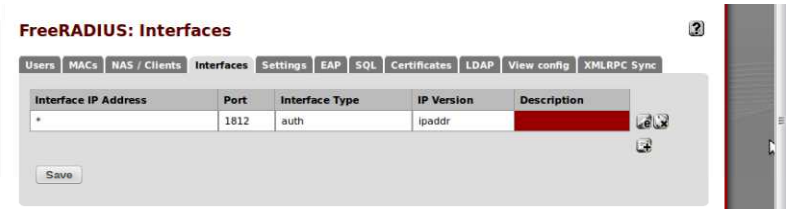


Figure IV.49: les interfaces FreeRADIUS.

IV.10.6. Ajout de clients

L'étape suivante de la configuration du serveur d'authentification est d'ajouter des voix client. Chaque appareil qui utilisera le serveur RADIUS pour l'authentification aura besoin d'avoir un client de voix configuré dans les paramètres.

Au début, on clique sur l'onglet Client/NAS. Après, entrons l'adresse IP de l'appareil où les demandes d'authentification seront en client IP. Et entrons un mot de passe sécurisé dans le client secret partagé. Ceci devra être placé sur le dispositif client ainsi.

Dans la section de configuration miscellany nous devons choisir un type de client à partir du menu déroulant. Si aucun des types énumérés conviennent, nous pouvons sélectionner d'autres. [2]

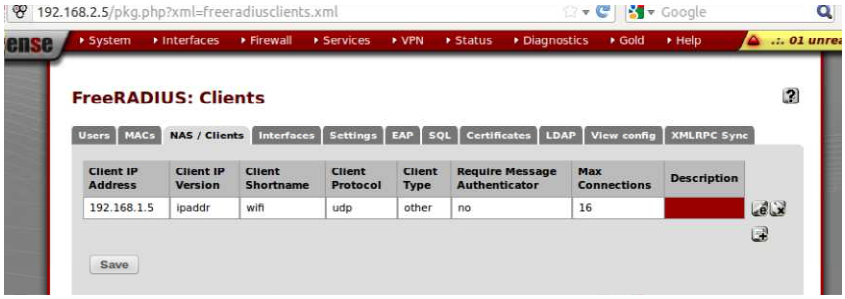


Figure IV.50: les clients FreeRADIUS.

IV.10.7.Création de comptes d'utilisateurs:

La dernière étape consiste à créer des comptes utilisateurs. Pour créer le compte aller à l'utilisateur dans les paramètres de l'emballage et cliquer sur le signe plus pour ouvrir la nouvelle page de création d'utilisateur. [2]

Il y a seulement deux champs obligatoires sur cette page, le nom d'utilisateur et mot de passe. Tous les autres paramètres sont optionnels et s'appliquent surtout aux utilisateurs du portail en captivité. [2]

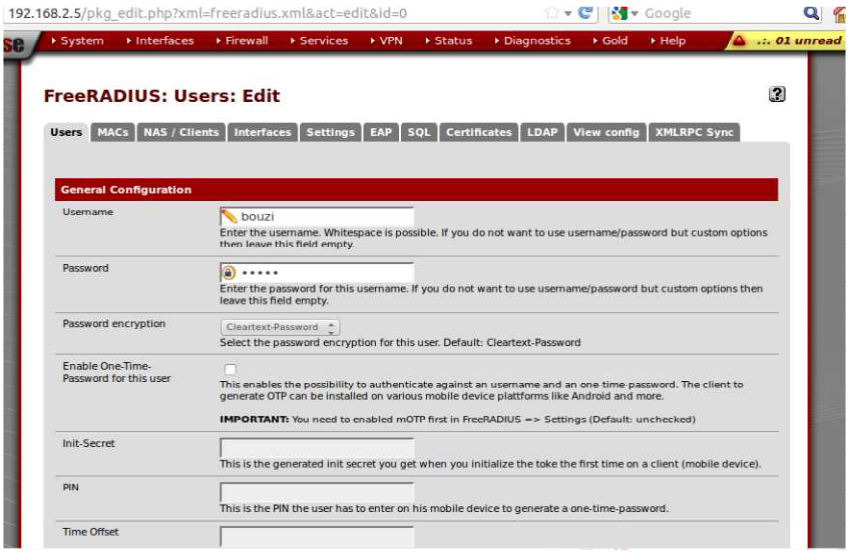


Figure IV.51: création d'un utilisateur FreeRADIUS.

On clique sur save pour enregistrer l'utilisateur.

IV.10.8. Configuration Le portail captif

Nous allons désormais voir la procédure afin de mettre en place le portail captif. Pour cela, allons dans la section Captive portal.

On coche la case « Enable captive portal », puis on choisit l'interface sur laquelle le portail captif va écouter (LAN dans notre cas). [29]

Figure IV.52: le portail captif.

Ensuite vient la méthode d'authentification.

Trois possibilités s'offre à nous :

• Sans authentification, les clients sont libres.

• Via un fichier local.

• Via un serveur RADIUS.

Pour des raisons de sécurités, nous avons mis en place un serveur RADIUS. [29]

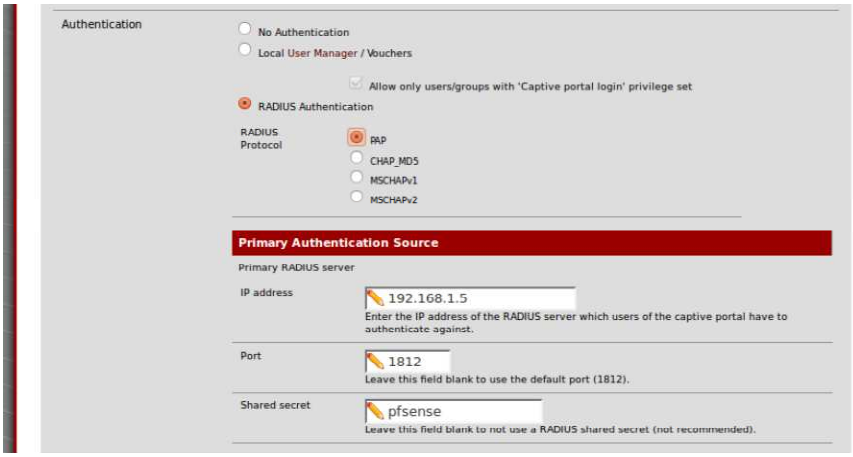


Figure IV.53: méthode d'authentification choisie.

On clique sur save pour enregistrer.

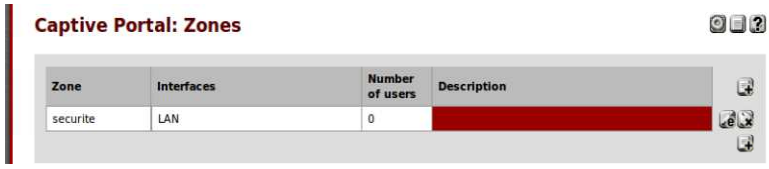


Figure IV.54: zones de portail captif.

Il est possible par la suite de sécuriser l'accès au portail captif.

IV.10.9. Configuration d'un serveur Web avec la méthode de Voucher

La première fois qu'on active la Chèques, une paire de clés RSA sont générés automatiquement.

Figure IV.55: activation de la méthode Vouchers

Les touches RAS de pré généré sont 32 bits. Pour l'instant, nous allons utiliser la valeur par défaut.

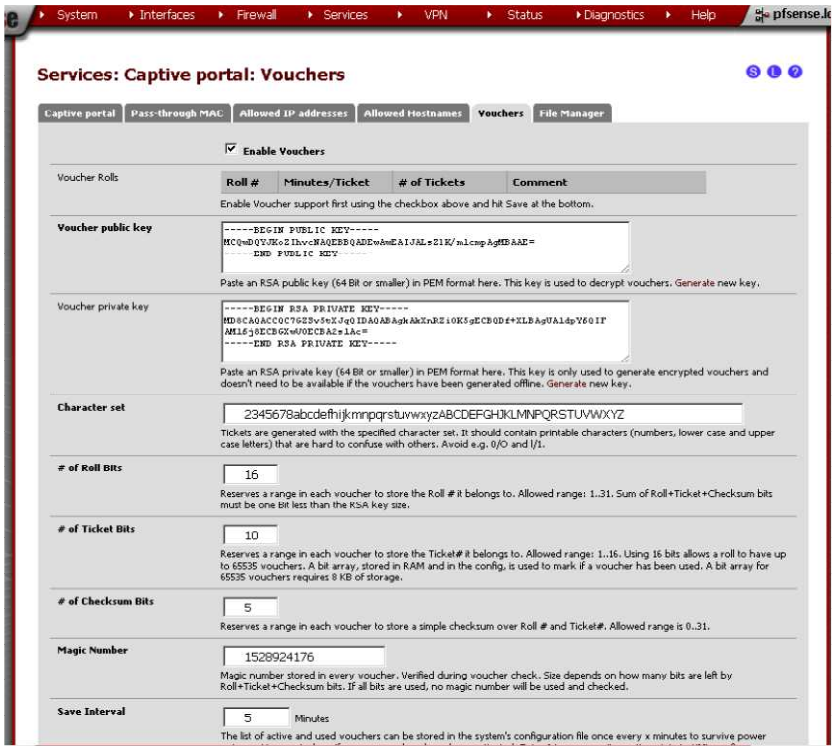


Figure IV.56: changements nécessaires pour activer le Vouchers.

Pour l'Intervalle de sauvegarde, la valeur par défaut est de 5 minutes, mais on ne veut pas l'état des bons pour être conservé dans le fichier de configuration, alors on va le changer à 0. On laisse le reste des champs aux valeurs par défaut. Enregistrer la configuration.

Nous allons générer des bons, dans la section Bon Rolls, cliquer sur le signe "+".

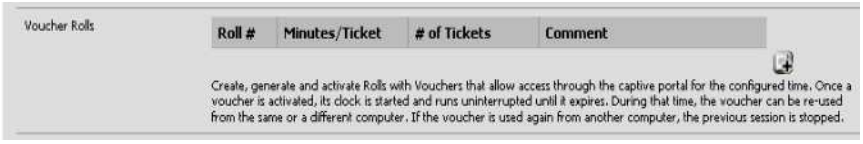


Figure IV.57: la section Vouchers Rolls.

Sur la nouvelle page, entrer:

• Rouleau# : 1

• Procès-verbal par Ticket: 10 minutes.

• Compter: 12 cela est le nombre de chèques générés.

• On peut mettre un peu de commentaire pour référence.

Sauvegarder.

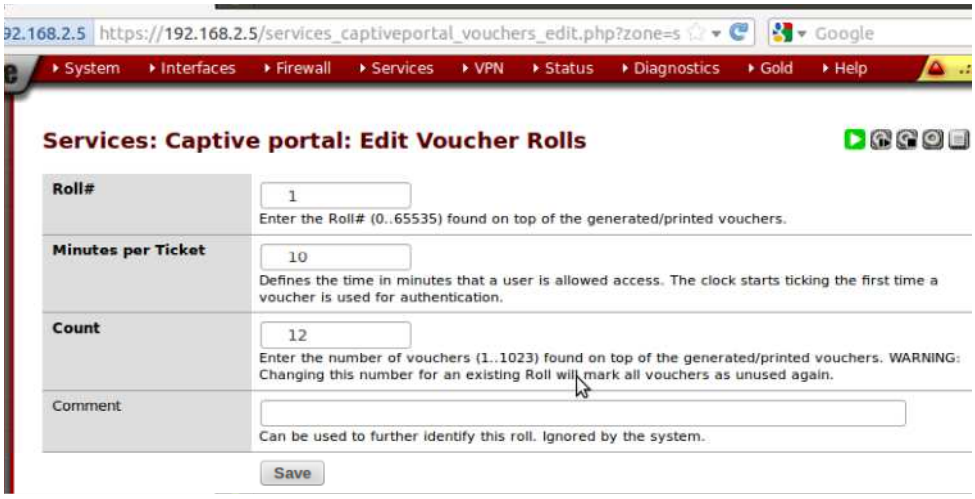


Figure IV.58: ajout d'un Vouchers.

À son retour à Chèques feuille, cliquer sur le cercle de "i" dans d'exporter la liste de pièces justificatives.

Figure IV.59: exportation de pièces justificatives.

Le résultat devrait ressembler à ceci:

Figure IV.60: enregistrement du fichier exporté.

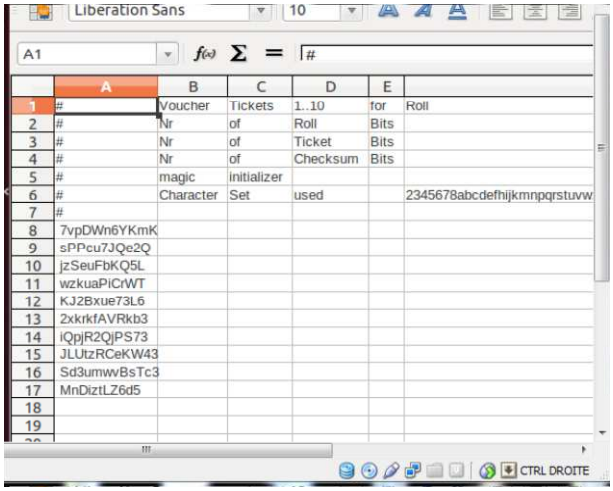


Figure IV.61: ouverture du fichier enregistré.

IV.11.Installation et configuration d'un serveur Web (Apache)

IV.11.1. Définitions

Le terme serveur Web désigne un ordinateur tenant le rôle de serveur informatique sur lequel fonctionne un logiciel serveur HTTP ou le logiciel serveur HTTP lui-même.

Un serveur HTTP est un logiciel servant des requêtes respectant le protocole de communication client-serveur HTTP, qui a été développé pour le WWW. D'autres ressources du Web comme les fichiers à télécharger ou les flux audio ou vidéo sont en revanche fréquemment servies avec d'autres protocoles. La plupart des ordinateurs utilisés comme serveur Web sont reliés à Internet et hébergent des sites Web du WWW. Les autres serveurs se trouvent sur des intranets et hébergent des documents internes d'une entreprise, d'une administration, etc.

Les serveurs Web comportent un dossier contenant les fichiers (pages HTML, images,...) qui constituent les pages Web d'un site. [1]

Un site Web statique est un site ou chacune des pages est créée en HTML. Un ordinateur qui se connecte au serveur, demande une page. Celle-ci lui est directement servie (elle est stockée toute prête sur le serveur):

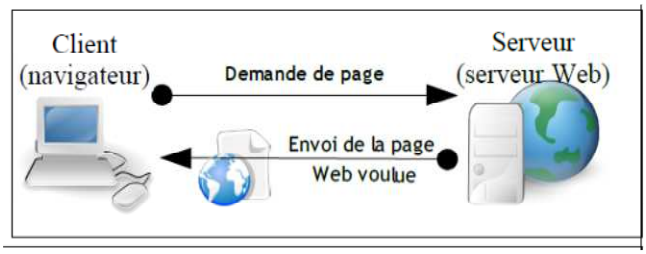


Figure IV.62:site Web statique.

Par opposition, un site Web dynamique est un site Web dont les pages sont générées dynamiquement à la demande.

Le contenu est obtenu en combinant l'utilisation d'un langage de scripts ou de programmation et une base de données.

Il s'agit souvent de PHP pour le langage et MySQL pour la base de données.

Dans les sites dynamiques, le contenu (articles) est séparé de l'habillage (modèles ou squelette). Les avantages sont donc loin d'être négligeables, et les possibilités de dynamisation évoluent de jour en jour. Les rédacteurs du contenu ne sont pas forcément habilités à publier leurs articles. L'administrateur quant à lui peutvalider ou non les articles et changer l'habillage.

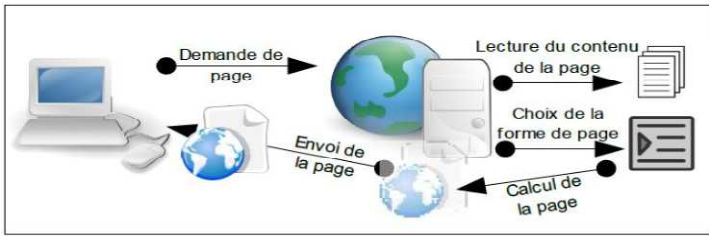


Figure IV.63: site Web dynamique.

IV.11.2. Configuration de serveur Web Apache

Apache est un serveur http libre, c'est un des serveurs http les plus utilisé sur internet.

Pour faire fonctionner un serveur Web Apache2 sur notre serveur Ubuntu 9.10. On doit suivre les étapes suivantes:

1- A partir de terminal, on va installer les paquets .deb nécessaires, en utilisant la commande: dpkg –i nom-paquet.deb

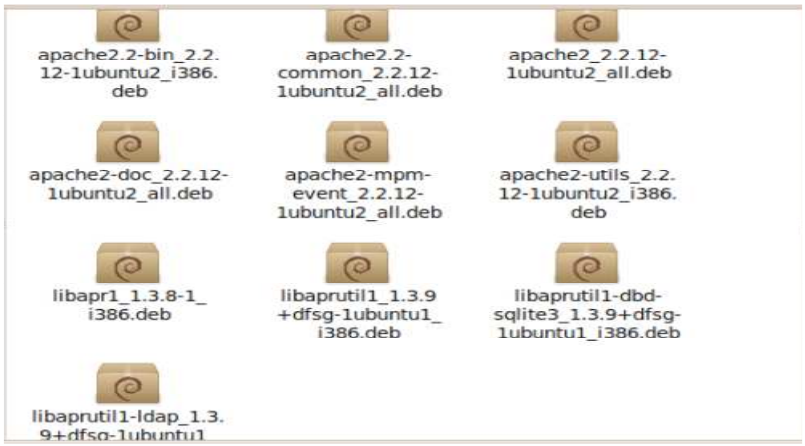


Figure IV.64: paquets nécessaires pour installation d'Apache.

2- Lancer un serveur Apache en écrit la commande suivante:

/etc/init.d/apache2 start.

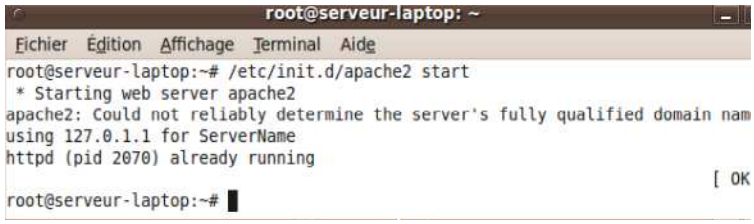


Figure IV.65: démarrage d'Apache.

3- La configuration globale d'apache s'effectue par modification du fichier de configuration/etc/apache2/apache2.conf.

Apache pouvant gérer plusieurs serveurs, on trouve des compléments pour la configuration de chaque serveur dans /etc/apache2/site-enabled.

Le site web par défaut se trouve dans le chemin par défaut : /var/www/ (en trouve le siteindex.html)

Testons le fonctionnement de serveur Web avec URL suivant: http://localhost:80

Pour accès authentifié nous utilisons les deux fichiers suivants : .htpasswd (contient les utilisateurs qui ont le droit de consulter la page web)et .htaccess (fichier .htaccess sera placé dans le répertoire ou se trouve les pages web a protégées).

4- On va créer deux utilisateurs : licence et master par la commande htpasswd:

•Pour créer le premier utilisateur en utilise la commande suivante :

/etc/apache2/htpasswd -c .htpasswd nom-utilisateur

• Pour ajouter un autre utilisateur en utilise uniquement la commande htpasswd sans l'option c :

/etc/apache2/htpasswd .htpasswd nom-utilisateur

5- Créer le fichier .htaccessest le remplir par les informations suivantes:

Authname "securité web"

AuthUserFile /etc/apache2/.htpasswd

AuthType Basic

requirevalid-user |list-user

6- Fichier /etc/apache2/site-enable/default contient les informations sur le chemin ou se trouve les sites web a consultées pas les clients de navigateurs Internet:

Créer un site web appelé index.html

Créer un répertoire web dans le chemin /home/serveur/

Il faut que l'utilisateur serveur existe dans votre home

Mettre le site web index.html dans le répertoire web (/home/serveur/web)

Modifier le fichier : /etc/apache2/site-enable/default est ajouter les lignes suivantes:

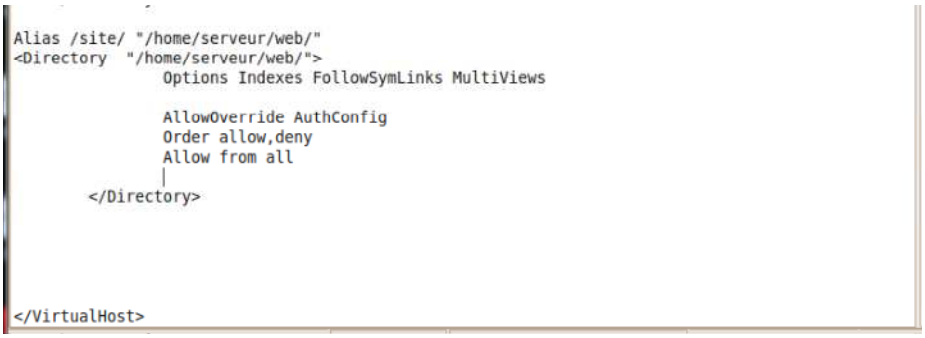


Figure IV.66: champs ajoutés dans /etc/apache2/site-enable/default.

Le site Web a hébergé se trouve dans le chemin /home/serveur/webavec un Raccourci indiqué par /essai/

7- Pour accéder au site Web en écrit sous linux à partir d'un navigateur Internet sur une machine cliente :http://192.168.1.10/essai/index.html

IV.12. Teste d'accès au service Web

Ouvrir un navigateur sur un ordinateur connecté à l'interface des clients de PfSense, entrer une adresse Web, et on devrait être présenté avec la page portail captif.

IV.12.1. Par nom d'utilisateur et mot de passe

Pour le test RADIUS, on entre un nom d'utilisateur et mot de passe.

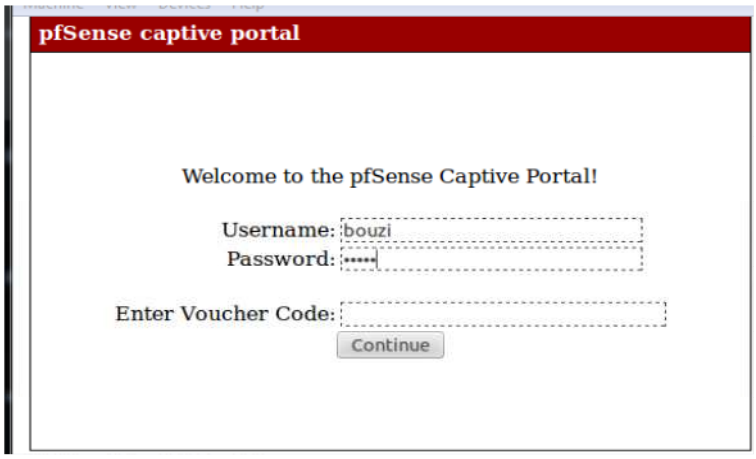


Figure IV.67: test par nom d'utilisateur et mot de passe.

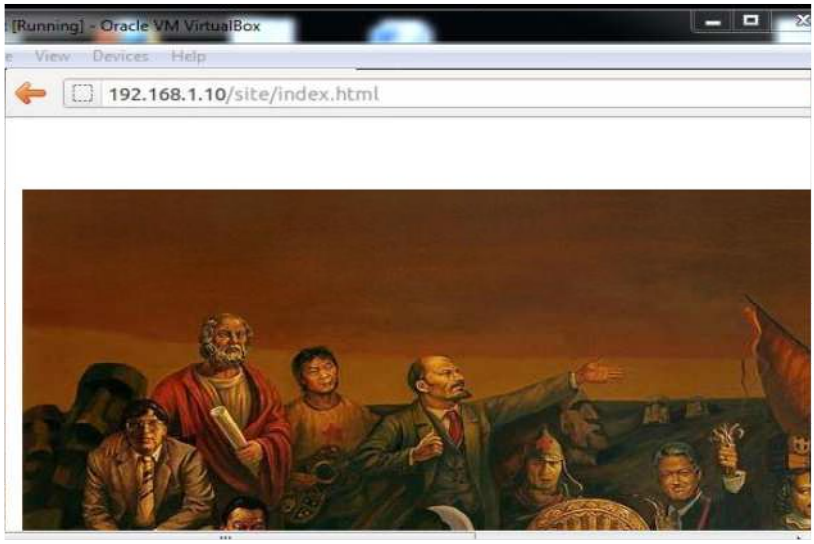


Figure IV.68: accès à la page web de serveur apache.

IV.12.2. Par Vouchers

Pour le système de chèque, copier l'une des lignes du fichier csv, et de le coller dans le champ chèque.

Enfin, nous pouvons importer une page Web qui servira de page d'accueil et on tape l'adresse de serveur 192.168.1.10/site/index.html

Le résultat devrait ressembler à ceci:

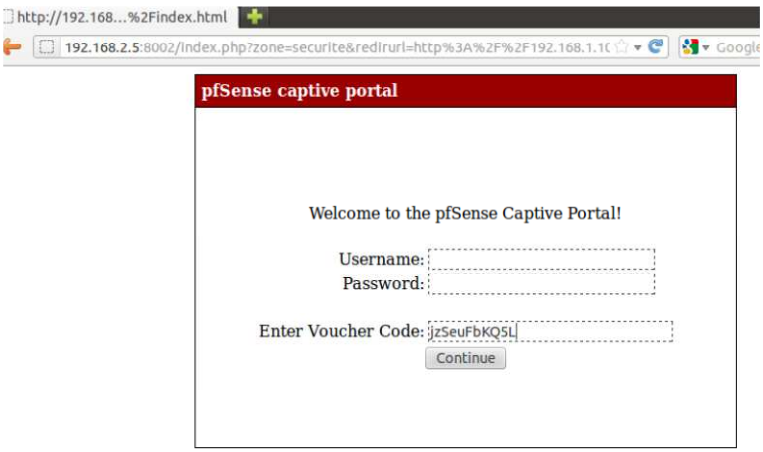


Figure IV.69: test par Vouchers.

Figure IV.70:accès au serveur web par un chèque

IV.13. Conclusion

PfSense est donc un moyen efficace pour gérer et protégé l'accès à l'internet. Les différentes options traitées dans notre mémoire permettent d'intégrer parfaitement PfSense comme une moyenne de sécurité fondamentale, notamment pour l'authentification des utilisateurs via un serveur freeRADIUS.

Conclusion générale

Le système d'authentification est un point fondamental et nécessaire dans le domaine de sécurité des réseaux.
Nous sommes intéressés dans notre projet de fin d'étude au système d'authentification free Radius qui fournit une source centrale d'authentification pour différents appareils et services du réseau.

Authentification central est beaucoup plus facile à gérer que de garder trace des différents comptes locaux à travers des dispositifs distincts dans un réseau.

PfSense est donc un moyen efficace pour gérer et protéger l'accès à l'internet. Les différentes options traitées dans notre mémoire permettent d'intégrer parfaitement PfSense comme une moyenne de sécurité fondamentale, notamment pour l'authentification des utilisateurs via un serveur freeRADIUS.

Au final, il n'y a pas de bonne ou de mauvaise méthode d'authentification. Le choix d'une méthode d'authentification doit se faire avant tout en fonction de son usage et de son contexte. Ce qui est bon pour contrôler l'accès direct à un équipement ou à un local, n'est pas forcément bon pour contrôler l'accès distant sur un réseau comme Internet. Et vice versa. Le niveau d'authentification requis (faible, moyen, fort) doit aussi être adapté aux véritables enjeux de l'identité et des conséquences d'une usurpation de cette identité. Enfin, le nombre d'utilisateurs concernés est aussi un facteur déterminant.

Références bibliographique

Références:

[1]: URL: http://www.cndp.fr/crdp-dijon/Installer-un-serveur-Web.html

[2]: URL: http://fr.pluslib.com/technologie/internet-et-le-web/comment-configurer-un-serveur-radius-sur-pfsense-utilisant-le-paquet-freeradius2.php

[3]: URL: http://www.labo-microsoft.org/articles/win/rad2003/10/

[4]: Younes ASIMI, Yassine SADQI et Mohamed ZAOUI, Etude théorique et pratique de RC5, RC6, SNORT et RADIUS & leurs applications, PFE Master.

[5]: URL: http://www.dicofr.com/cgi-bin/n.pl/dicofr/definition/20061114165935

[6]: URL: http://igm.univ-mlv.fr/~dr/XPOSE2003/Mandille/Radius.htm

[7]: URL: http://fr.wikipedia.org/wiki/Remote_Authentication_Dial-In_User_Service

[8]: URL: http://fr.wikipedia.org/wiki/Authentification

[9]: URL: http://fr.wikipedia.org/wiki/Autorisation

[10]: Gloria Gihanne Agnès et YAKETE-OUALIKETTE, MISE EN PLACE D'UN SERVEUR D'AUTHENTIFICATION FREERADIUS AVEC UNE BASE DE DONNEES MYSQL, PFE master, 26 Avril 2014

[11]: Serge Bordères, Authentification réseau avec Radius, Livre, novembre 2006

[12]: CLEMENÇON Maxime, JUANEDA Matthieu et KERGADALLAN Guillaume, Authentification nomade, RAPPORT FIN DE PROJET.

[13]: URL: http://fr.wikipedia.org/wiki/FreeRADIUS

[14]: URL: http://www.vulgarisation-informatique.com/base-donnees.php

[15]: URL: http://www.linux-france.org/article/cel/SICOMOR/SGBDR/html/Rapport_7-9V10.html

[16]: URL: http://sql.sh/cours/create-database

[17]: URL: http://fr.wikibooks.org/wiki/MySQL/Parcourir_les_bases_de_donn%C3%A9es

[18]: URL: http://tecfa.unige.ch/guides/mysql/fr-man/manuel_DESCRIBE.html#DESCRIBE

[19]: URL: http://tecfa.unige.ch/guides/mysql/fr-man/manuel_toc.html

[20]: URL: http://fr.wikipedia.org/wiki/MD5

[21]: ISMAIL RACHDAOUI – GRT5, Pfsense FreeBSD, PFE, 2013

[22]: URL: http://fr.wikipedia.org/wiki/FreeBSD

[23]: URL: http://www.generation-linux.fr/index.php?post/2009/11/30/Presentation-de-pfSense

[24]: Marwen Ben Cheikh Ali & Khlifa Hammami, Mise en place d'un firewall open source PfSense, PFE, 2012/2013

[25]: URL: http://doc.ubuntu-fr.org/virtualbox

[26]: URL: http://www.01net.com/telecharger/windows/Utilitaire/systeme/fiches/37588.html

[27]: URL: http://openclassrooms.com/courses/reprenez-le-controle-a-l-aide-de-linux/installez-linux-dans-une-machine-virtuelle

[28]: Michel Bonnefond, Guide Pfsense 2.0, document, mai 2012

[29]: Anthony COSTANZO, Damien GRILLAT et Lylian LEFRANCOIS, Etude des principaux services fournis par PfSense, tutorial, 2009.

Résumé :

Le système d'authentification est un point fondamental et nécessaire dans le domaine de sécurité des réseaux.

Nous sommes intéressés dans notre projet de fin d'étude au système d'authentification free Radius qui fournit une source centrale d'authentification pour différents appareils et services du réseau.

L'outil PfSense est un moyen de sécurité efficace notamment pour l'authentification des utilisateurs via un serveur freeRADIUS.

Abstract :

The authentication system is a fundamental and necessary security point in the networking field.

We are interested in our project of end of study at free Radius authentication system which provides a central source of authentication for different devices and network services.

The tool PfSense is an effective means of security including user authentication via a freeRADIUS server.

ملخص :

نظام التوثيق هو نقطة أمنية أساسية وضرورية في مجال الشبكات.

ونحن مهتمون في مشروعنا من نهاية الدراسة في حر نظام التوثيق الشعاع الذي يوفر مصدرا مركزيا للمصادقة الأجهزة المختلفة وخدمات الشبكة.

الأداة **PfSense** وسيلة فعالة للأمن بما في ذلك مصادقة المستخدم عبر خادم **freeRADIUS.**