République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A/Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique

Mémoire de fin de d'étude
en vue de l'obtention du diplôme de Master en Informatique
Option : Administrastion et Sécurité des Réseaux

Thème

Proposition d'une solution VPN
Cas : Université de Béjaïa

Présenté par :
✓ M^{lle} **Ait braham** Assia.
✓ M^{lle} **Medjkoune** Lynda.

Proposé et encadré par :
✓ M^{r} **Sider** Abderrahmane.

Devant le jury composé de :

✓**Président : M^{r} Kadjouh Nabil.**
✓**Examinateur : M^{me} Ghanem Souhila.**

Promotion 2015/2016

Remerciements

Nous tenons en premier lieu à remercier le bon DIEU de nous avoir donné les moyens et l'énergie de réaliser ce travail.

Nous exprimons nos sincères remerciements pour notre promoteur Monsieur Sider Abderrahmane qui nous a orienté dans la mise en oeuvre de ce projet.

Nous remercions aussi Madame Baiche Nabila administratrice réseau du centre de calcule de l'université de béjaia qui a été d'une grande sympathie envers nous, qui nous a apportée son aide et contribuée à l'élaboration de ce projet, grâce à ses précieux conseils qui furent d'un apport considérable.

Que les membres du jury trouvent ici nos remerciements les plus vifs pour avoir accepté d'honorer par leur jugement notre travail.

Tout notre respect à toute personne qui a de près ou de loin contribué à la réalisation de ce modeste travail particulièrement nos chères familles et nos amis(es).

Dédicaces

Nous dédions ce modeste travail à nos chère parents en témoignage de notre gratitude pour leurs dévouement et leurs soutient permanent durant toute nos études, leurs réconfort, leurs sacrifices pour nous voir réussir un jour.
Que dieu les garde.
A nos frères et sœurs.
A nos amis.

TABLE DES MATIÈRES

TABLE DES FIGURES

Liste des abréviation

DNS Domain Name System

ACL Access Control List

LAN Local Area Network

VPN Virtual Private Network

IP Internet Protocol

RLE Réseaux locaux d'entreprise

TCP Transmission Control Protocol

UDP User Datagram Protocol

L2F layer two forwarding

L2TP Layer Tow Tunneling Protocol

IPSec IP Security Protocol

LAC L2TP Access Concentrator

LNS L2TP Network Server

AH (authentification header)

ESP Encapsulating Security Payload

SSL Secure Socket Layer

FAI Fournisseur d'accès Internet

RFC requests for comments

3Com Computers, Communication et Compatibility

GRE Generic Routing Encapsulation

MS-CHAP Microsoft Challenge Handshake Authentication Protocol

PAP Password Authentication Protocol

IETF Internet Engineering Task Force

MD5 Message Digest 5

DES Data Encryption Standard

INTRODUCTION GÉNÉRALE

Les réseaux locaux d'entreprise (LAN ou RLE) sont des réseaux internes à une organisation. Ces réseaux sont de plus en plus souvent reliés à Internet par l'intermédiaire d'équipements d'interconnexion. Il arrive ainsi souvent que ces entreprises éprouvent le besoin de communiquer avec des filiales, des clients ou même du personnel géographiquement éloignés via Internet.

Pour autant, les données transmises sur Internet sont beaucoup plus vulnérables que lorsqu'elles circulent sur un réseau interne à une organisation. En effet, les données empruntent une infrastructure de réseau publique appartenant à différents opérateurs. Ainsi il n'est pas impossible que sur le chemin parcouru, le réseau soit écouté par un utilisateur indiscret ou même détourné. Il n'est donc pas concevable de transmettre dans de telles conditions des informations sensibles pour l'organisation ou l'entreprise. Alors comment une succursale d'une entreprise peut-elle accéder aux données situées sur un serveur de la maison mère distant de plusieurs kilomètres en toute sécurité et à moindre coût ?

La première solution pour répondre à ce besoin de communication sécurisée consiste à relier les réseaux distants à l'aide de liaisons spécialisées. Toutefois la plupart des entreprises ne peuvent pas se le permettre car ces lignes sont coûteuses.

Les VPN ont commencés à être mis en place pour répondre à ce type de problématique. Ainsi grâce au VPN deux entreprises géographiquement distantes peuvent communiquer en toute sécurité par Internet à travers une connexion chiffrée. Mais d'autres problématiques sont apparues et les VPN ont aujourd'hui pris une place importante dans les réseaux informatique et l'informatique distribuée, notamment ces dernières années, ou on assiste à l'apparition d'une nouvelle pratique qui est le télétravail qui offre aux salariés l'alternative de travailler depuis leur domicile sans être présent dans les locaux de l'entreprise, et cela grâce aux technologies de communication et de l'information principalement Internet.

Le télétravail a favorisé le déploiement des VPN d'accès, qui offrent aux salariés (télétravailleurs) la possibilité de se connecter directement sur le réseau interne de l'entreprise

avec une connexion sécurisée afin d'empêcher la divulgation d'informations sensibles sur le réseau. Aujourd'hui ce type de VPN est de plus en plus mis en oeuvre, pas juste par des télétravailleurs mais par n'importe quel utilisateur sur Internet, pour se protéger des risques de piratage de données sur Internet ou bien pour masquer son adresse ip sur Internet...

Le centre de calcul de l'université Abdrerrahmane Mira de Béjaïa fait face ces derniers temps à de plus en plus de demandes de la part du personnel et des enseignants ayant pour objet la permission d'accéder au LAN de l'université depuis leur domicile par Internet, dans le but de continuer leurs travaux (accès aux machines de calcul de la DI, accès aux machines du laboratoire, accès aux machines personnelles, ...)

Pour des raisons de sécurité, ces demandes sont difficiles à accepter par les administrateurs du centre de calcul ; la crainte, très justifiée, que l'utilisateur se fasse dérober son mot de passe lors de sa connexion est forte (dans les applications courantes, le mot de passe circule en clair sur le réseau).

Au cours du stage qu'on a effectué au centre de calcul de l'université de Béjaïa, notre travail consistait à concevoir un VPN d'accès au LAN de l'université pour permettre au personnel d'accéder au données internes du LAN en toute sécurité avec une connexion chiffrée, notre second objectif sera de faire en sorte de faciliter l'administration de ce VPN en mettant en place tout les outils nécessaire pour cela.

Afin de réaliser notre solution, nous avons organisé notre travaille en quatre chapitres :

- Le premier chapitre abordera des notions de sécurité à connaitre pour mettre en place notre VPN.
- Le deuxième chapitre sera consacré à la présentation de généralités sur les VPN et les solutions VPN disponible sur le marché, ainsi que la comparaison de ces différentes solutions. Notre choix s'est porté sur la solution SSL OpenVPN.
- Le troisième chapitre concerne l'étude préalable du réseau existant et suggestion de solution.
- Le quatrième chapitre sera la mise en oeuvre de la solution VPN, la solution sera déployée sur un serveur Ubuntu 14.04.

Nous terminerons par une conclusion et perspective générale de notre travail.

CHAPITRE 1

LA SÉCURITÉ INFORMATIQUE

1.1 Introduction

La sécurité des réseaux informatiques est un sujet essentiel qui favorise le développement des échanges d'information dans tout les domaines. L'expansion de l'importance grandissante des réseaux informatiques a engendrée le problème de sécurité des systèmes de communication. Dans la plupart des organisations informatisées, partager les données directement entre machines est un souci majeur. Il s'avère indispensable de renforcer les mesures de sécurité, dans le but de maintenir la confidentialité, l'intégrité et le contrôle d'accès au réseau pour réduire le risque d'attaque.

Au cours de ce chapitre, nous présenterons les principaux concepts liés à la sécurité des réseaux ainsi que les stratégies et les outils permettant de protéger les infrastructures d'entreprise.

1.2 Définition de la sécurité informatique

On définit la sécurité informatique comme un ensemble de précautions prises, dont l'objectifs principal est de réduire la vulnérabilité d'un système informatique, contre les risques accidentelles ou intentionnelles auxquelles il peut être confronté et qui garantissent de ce fait que les ressources de ce système qu'elles soient matérielles ou logicielles sont utilisées uniquement dans le cadre prévu. [2]

1.2.1 Les objectifs de sécurité

La sécurité d'un système repose sur cinq grands principes : [1]

- **L'intergrité des données :** Il faut garantir qu'au cours de la communication les données n'ont pas étaient altérées. L'intégrité des données doit valider l'intégralité des données, leur précision, l'authenticité et la validité.

- **La confidentialité des données :** Seules les personnes autorisées doivent avoir accès aux données, aucune interception telles qu'elle soit ne doit mettre en danger le système, pour garantir cela les données doivent être cryptées, seules les personnes autorisées possèderont la clé permettant la compréhension de ces données.

- **La disponibilité :** Il faut garantir que le système fonctionne d'une manière optimal et s'assurer que l'accès aux services et aux ressources se fait à n'importe quel moment.

- **La non-répudiation des données :** Une transaction ne peut être niée par aucun des correspondants. La non-répudiation de l'origine et de la réception des données prouve que les données ont bien été reçues.

- **L'authentification :** Elle limite l'accès aux personnes autorisées en s'assurant de l'identité d'un utilisateur avant l'échange de données.

1.2.2 Terminologie de la sécurité informatique

La sécurité informatique utilise un ensemble de termes bien spécifique, que nous énumérons dans ce qui suit : [3]

- **Vulnérabilité (Faille/faiblesse) :** C'est une faille ou un point où le système est susceptible d'être attaqué.
- **Menace :** Ce sont les violations potentielles de sécurité. C'est l'ensemble des personnes, choses, évènements qui posent danger pour un patrimoine en termes de confidentialité, d'intégrité, et de disponibilité. Il existe deux types de menaces, les menaces accidentelles (exposition) et les menaces intentionnelles (attaques).
- **Attaque :** Une attaque désigne un accès ou une tentative d'accès non autorisés à un système.
- **Les contre-mesures :** Ce sont les procédures ou techniques misent au point pour protéger le système en cas de vulnérabilité quelconque ou pour contrer une attaque spécifique.

– **Politique de sécurité :** Elle se fonde sur une analyse des risques décrivant les ressources critique du réseau, ses vulnérabilités, les probabilités d'occurrences des menaces, ainsi que les conséquences. A partir de cette analyse, des outils et des procédures sont définis et déployés afin de protéger le système et répondre aux objectifs de sécurité.

1.3 Les risques de securité

Les auteurs définissent les risques comme "ce qu'on peut perdre en l'absence de moyens adéquats de sécurisation", ils les classent en deux catégories distinctes, qui sont les risques physiques (incendies, explosion, défaillance matérielle...) et les risques logiques qui concernent les données d'un système. Parmi les risques logiques on peut citer le vol d'informations, l'usurpation d'identité, l'intrusion et la mise hors service des ressources systèmes. [2]

1.4 Les attaques

1.4.1 Les différentes étapes d'une attaque

La plupart des attaques, de la plus simple à la plus complexe fonctionnent suivant le même schéma : [4]

1. **Identification de la cible :** Le but est de récolter un maximum de renseignements sur la cible en utilisant des informations publiques et sans engager d'actions hostiles, par exemple l'interrogation des serveurs DNS.
2. **Le scanning :** L'objectif est de récolter des informations plus détaillés que dans la première étape et éventuellement compléter les informations déjà réunies, comme par exemple obtenir les adresses IP utilisées. Pour arriver à ce but les actions engagées peuvent être hostile, en effet certaines techniques de scans sont susceptibles d'entraîner la défaillance de certains systèmes.
3. **L'exploitation :** Le pirate étudie les informations recueillies jusque-là et cherche des failles à exploiter sur la cible. Ces failles peuvent être au niveau des protocoles, des applications, des systèmes d'exploitation, ou autres.
4. **La progression :** C'est la dernière étape qui consiste à la réalisation de l'attaque préparée. Le but du pirate est dans la majorité des cas d'élever ses droits vers root (ou system) sur un système afin d'avoir toutes les autorisations.

1.4.2 Les différents types d'attaques

Il existe trois types d'attaques : [4]

1. **Attaque direct :** Ce type d'attaque n'est pas très courants car le hacker attaque sa victime directement à partir de son ordinateur, ce qui est un peu risqué pour lui. Ce type d'attaque n'est pas utilisé dans les piratages de grande envergure et les programmes de hack utilisés sont généralement peu paramétrables.

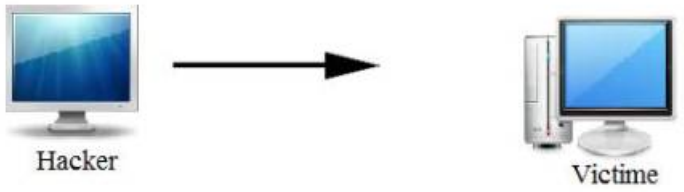

FIGURE 1.1 – Attaque directe.

2. **Les attaques indirectes par rebond :** Cette attaque est beaucoup utilisée par les hackers car elle offre l'avantage de masquer l'identité (l'adresse IP) du hacker. Son principe est d'envoyer les paquets d'attaque à un ordinateur intermédiaire, qui répercute l'attaque vers la cible.

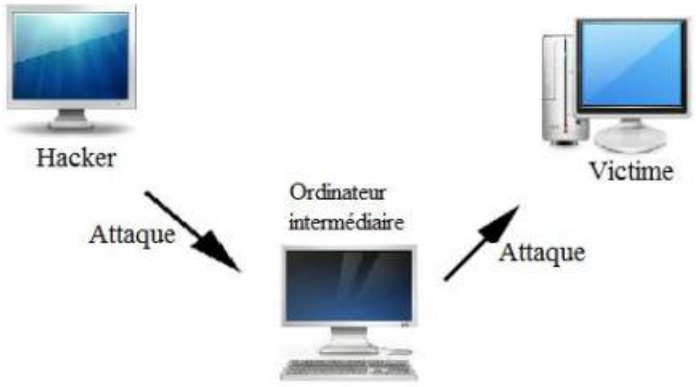

FIGURE 1.2 – Attaque indirectes par rebond.

3. **Les attaques indirectes par réponse :** Elle est une dérivée de l'attaque par rebond. Le hacker envoie une requête à l'ordinateur intermédiaire mais cette fois c'est la réponse à cette requête qui sera envoyer à la victime.

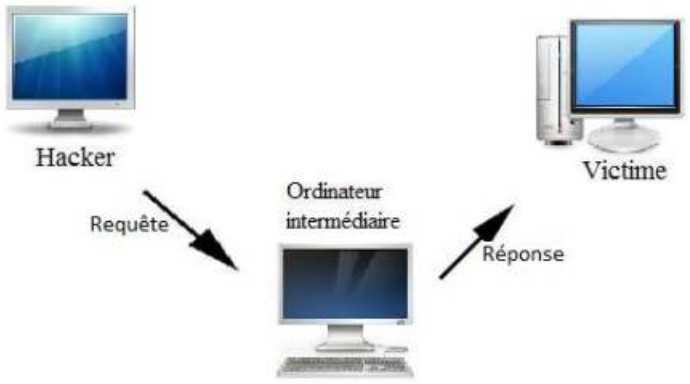

FIGURE 1.3 – Attaque indirectes par réponse.

1.4.3 Quelques attaques courantes

Il existe un grand nombre d'attaques informatiques ayant chacune des objectifs différents, en voici quelques-unes : [4]

- ***Le sniffing***
  Cette attaque se fait avec un logiciel appelé "sniffer" placé sur un ordinateur du même réseau que la machine cible, le sniffer intercepte toutes les trames que la carte réseau d'un ordinateur reçoit et qui ne lui sont pas destinées. Grâce à ça on peut savoir par exemple les pages web que consultent les personnes sur le réseau ou bien les mails envoyés et reçus.

- ***L'IP spoofing***
  Cette attaque consiste à ce faire passer pour une autre machine qui a des privilèges ou droits élevés (root) dans l'accès à un serveur cible en falsifions son adresse IP.

- ***Le DoS (Denial of Service)***
  Cette attaque ne permet pas d'avoir accès aux données mais d'empêcher d'y accéder en générant des arrêts de service, empêchant ainsi le bon fonctionnement du système.

- ***Les programmes cachés ou les virus***
  Il existe de nombreuses variétés de virus parmi lesquels les vers qui se propagent dans le réseau, les troyens qui créent des failles dans un système et les bombes logiques qui se lançent suite à un événement du système.

- ***Le craquage de mots de passe***
  Cette attaque consiste à essayer plusieurs mots de passe sur le système cible à l'aide d'un dictionnaire de mot de passe ou par la méthode de brute force jusqu'à trouver

le bon ou bien jusqu'à l'essai de tous les mots de passe disponibles.

1.5 Stratégies de sécurité

C'est un ensemble de moyens et de dispositifs mis en place pour sécuriser les systèmes d'information, nous citons les principaux : [5]

1.5.1 Le pare-feu

Le firewall est un composant du réseau informatique qui peut être matériel ou logiciel et qui a pour tâche le contrôle du trafic en filtrant les flux qui transitent entre différentes zone de confiance (exemple : entre Internet et un réseau interne). Il fonctionne selon différents modes, qui sont : [5]

1. **Le filtrage de paquets**

   Les paquets sont analysés en les comparants à un ensemble de filtres (règles), ils sont alors soit acceptés soit rejetés. Ce mode agit au niveau des couches transport et réseaux.

2. **La passerelle applicative (Gateway)**

   L'application Gateway permet de limiter les commandes à un service plutôt que de les interdire, ceci limite notamment le trafic direct entre le réseau protégés et Internet qui peut être une source d'intrusion. Ce mode agit au niveau de la couche application.

Le firewall est limité au contrôle d'accès il ne permet pas de se protéger des virus ou bien d'assurer l'authentification, la confidentialité ou l'intégrité des données qui y transitent. [5]

1.5.2 La zone Démilitarisé (DMZ)

Une DMZ est une interface situé entre un réseau connu (réseau interne) et un réseau externe (Internet). Une série de règles de connexion configuré sur le pare-feu font de cette interface une zone physiquement isolée entre les deux réseaux. Cette séparation physique permet d'autoriser les accès Internet à destination des serveurs placés dans la DMZ et non à ceux destinés au réseau privé (interne). La politique de sécurité mise en oeuvre sur la DMZ est généralement la suivante :

- Trafic du réseau externe vers la DMZ **autorisé.**
- Trafic du réseau externe vers le réseau interne **interdit.**
- Trafic du réseau interne vers la DMZ **autorisé.**
- Trafic du réseau interne vers le réseau externe **autorisé.**

- Trafic de la DMZ vers le réseau interne **interdit.**
- Trafic de la DMZ vers le réseau externe **refusé.**

Il est possible de mettre une DMZ en interne pour créer différents niveaux de protection et éviter les attaques en interne. [5]

1.5.3 Les technologies AAA

La signification de ces termes est la suivante : [6]

1. **Authentification**

   L'authentification consiste à vérifier qu'une personne/équipement est bien celle qu'elle prétend être. Ceci est généralement réalisé en utilisant un secret partagé ou à l'aide d'un logiciel approuvé (protocole RADUIS).

2. **Autorisation**

   L'autorisation intervient à l'issue de l'authentification. Une fois l'utilisateur authentifié, il faut s'assurer qu'il est autorisé à accomplir les actions qu'il demande, tels que l'accès aux fichiers, le droit d'écrire, etc. Parmi les mécanismes qui mettent au point l'autorisation on peut citer les listes de contrôle d'accès ACL.

3. **Comptabilité (Accounting)**

   Permet de collecter des informations sur les utilisateurs et les actions qu'ils accomplissent lorsqu'ils sont connectés aux équipements du réseau.

1.5.3.1 Quelques outils pratiques qui utilisent les technologies AAA

1. **LDAP(Lightweight Directory Access Protocol)**

   Le protocole LDAP prend en charge des communications client/serveur ayant comme client un utilisateur quelconque sur le réseau et comme serveur un serveur d'annuaire, plus précisément il fournit au client un ensemble de fonctions de connexion, déconnection, recherche, ... à effectuer sur les annuaires, ainsi que des mécanismes d'authentification, de chiffrement et des règles d'accès pour protéger les données qu'ils contiennent. [16]

   L'annuaire est une base d'information un peu comme une base de données qui stocke différents types d'informations comme par exemple des comptes Unix, dans ce cas l'annuaire pourra servir de base d'authentification pour accéder à des services sur le réseau, ou bien un parc matériel et sera de ce fait une sorte d'inventaire, ou encore des informations personnelles sur des personnes comme par exemple des dossiers scolaire d'élèves d'une écoles, etc. [16]

   Un annuaire n'est pas une base de données, les données qu'il stocke sont peu typées, et leurs organisation est hiérarchique sous forme d'arbre de sorte que l'accès aux

données en lecture soit rapide, c'est la principale différence avec une base de données qui vise une rapidité d'accès aussi bien en lecture qu'en écriture. [16]

Le protocole LDAP propose aussi des fonctions permettant la communication serveur-serveur pour leur permettre d'échanger leurs contenus, ou bien de liés les annuaires entre eux. [16]

LDAP en est actuellement à la version 3 et a été normalisé par l'IETF. [16]

- **Intérêt d'utiliser LDAP**

  Les annuaires LDAP peuvent être utilisés par d'autres applications comme base d'authentification. Dans un réseau ayant plusieurs services qui requièrent une authentification, les comptes des utilisateurs de ces services peuvent être regroupés dans un annuaire LDAP central au lieu d'avoir une liste d'utilisateurs pour chaque service, cela facilite grandement la tâche à l'administrateur réseau d'avoir qu'un seul annuaire à gérer au lieu de plusieurs listes d'utilisateurs. Tout les services du réseau seront désormais en mesure de s'authentifier par l'intermédiaire de ce seul annuaire. Les données contenues dans l'annuaire peuvent être protégées contre toute intrusion grâce à des fonctions SSL/TLS. [16]

2. **PAM(Pluggable Authentification Module)**

   PAM est une suite de librairies ou modules disponible sur les systèmes UNIX qui offrent aux différentes applications ayant des mécanismes d'authentification reposant sur les mots de passe de prendre en charge cette tâche d'authentification. L'application ne va plus utiliser ses mécanismes d'authentification mais confiera cette tâche à PAM. PAM donnent le choix entre plusieurs modules pour mettre au point l'authentification, on pourra par exemple utiliser le module pam-ldap.so pour s'authentifier à partir d'un annuaire LDAP, PAM permet aussi d'implémenter des limites d'accès au cas par exemple ou l'utilisateur n'est pas autorisé à se connecter à cet heure de la journée. [21]

   PAM facilite grandement la tâche de l'administrateur réseau qui ne devra connaitre que PAM et non tous les systèmes d'authentification pour chaque service mis en place. [21]

3. **Postfix**

   Postfix est un serveur de messagerie pour les systèmes Unix. Il est gratuit est open source et permet de créer son propre serveur de messagerie autonome qui se chargera de l'envoie et la réception du courrier de son propre domaine. Il est facile à configurer et à administrer. [17]

   - **Relai SMTP avec Postfix**

   Avec Postfix on a le choix de créer son propre serveur SMTP qui se chargera de l'envoi et de la réception du courrier du domaine ou bien l'utilisation d'un relais

SMTP c'est-à-dire passer par un autre serveur SMTP, les mails seront alors envoyé par Postfix à ce serveur qui se chargera de l'envoi et de la réception. [17]

1.5.4 Liste de contrôle d'accès (ACL)

Les utilisateurs n'ont pas tous les mêmes privilèges d'accès aux différents services ou zones d'un réseau. Afin d'interdire l'accès à certains utilisateurs les routeurs peuvent être utilisés pour mettre en place des ACLs, qui sont un ensemble de conditions appliquées à une interface du routeur qui lui indique les paquets à accepter ou à rejeter, sécurisant de ce fait le trafic en entrée comme en sortie. Il y a d'autre raisons de mettre en place des ACLs comme par exemple limiter le trafic réseau et accroître les performances, en limitant le trafic vidéo. [5]

1.5.5 Proxy

Un proxy est un logiciel qui sert d'intermédiaire, ou qui joue le rôle de relais entre deux autres logiciels pour faciliter leurs communications ou bien pour surveiller leurs échanges. Dans le cadre de la sécurité informatique, un proxy peut être utilisé pour filtrer l'accès à des services comme par exemple interdire la consultation de site web ou au contraire contourner des filtrages. [7]

1.5.6 Les réseaux privés virtuels

Les réseaux privés virtuels sont utilisés pour interconnecter des réseaux locaux à travers un réseau publique comme Internet. Ils reposent sur le principe de création d'un tunnel virtuel dont les extrémités identifiés appartiennent à deux réseaux locaux différents, les données circulent alors dans ce tunnel après avoir était chiffrée, le principale avantage des VPN est d'interconnecter des réseaux à moindre coût à travers un réseau publique au lieu d'utilisation de lignes dédiées très couteuses. [4]

1.5.7 Systèmes de détection d'intrusion (IDS)

Un système de détection d'intrusions (Intrusion Detection System) est un logiciel ou matériel qui a pour rôle la détection d'activités anormales sur un système cible analysé, il peut ainsi protéger un système contre les attaques, ou bien surveiller l'activité du réseau. [4]

1.6 La cryptographie

Il existe à l'heure actuelle deux grands principes de cryptage : le cryptage symétrique et le cryptage asymétrique.

1.6.1 Le cryptage symétrique

Le principe du cryptage symétrique est de crypter et décrypter les messages avec la même clé. Cette clé est partagée entre les parties communicantes. La principale problématique de ce cryptage est : comment transmettre cette clé de façon sûre sur un réseau non sûre pour tous les utilisateurs, surtout quand leurs nombres est très grand, ce qui rend l'utilisation de ce cryptage limité. [8]

Parmi les algorithmes de chiffrement symétrique on cite : Kerberos, DES (Data Encryption Standard), AES , etc.

1.6.2 Le cryptage asymétrique

Avec ce cryptage chaque utilisateur se voit doté de deux clés, une clé privée secrète connu que par lui-même et une clé publique accessible à tous. Un message chiffré avec la clé publique ne peut être déchiffré que pas la clé privé correspondante. Pour communiquer avec quelqu'un, un utilisateur doit chiffrer le message à envoyer avec la clé publique de ce dernier. Le destinataire déchiffre alors le message avec sa clé privée. [8]

Ce mode de fonctionnement résout la problématique de la transmission de clé du cryptage symétrique, en plus de permettre un contrôle d'intégrité sur les données et l'authentification des parties. [8]

Parmi les algorithmes de chiffrement asymétrique on cite : RSA.

1.6.3 Signature numérique

La signature numérique sert à garantir l'intégrité d'une donnée et d'authentifier celui qui la envoyé. Le message se voit appliqué une fonction mathématique qu'on appelle fonction de hachage, ce qui donne comme résultat un code de hachage qui est comme une empreinte digital, impossible à reproduire avec un autre message en lui impliquant la même fonction. [8]

1.7 Conclusion

Dans ce chapitre, nous avons défini les notions fondamentales de la sécurité informatiques et les stratégies à mettre en place pour remédier aux attaques. Le prochain chapitre

sera consacré aux réseaux privés virtuels.

CHAPITRE 2

GÉNÉRALITÉS SUR LES RÉSEAUX PRIVÉS VIRTUELS

2.1 Introduction

Les réseaux locaux type LAN permettent de faire communiquer les ordinateurs d'un site ou d'une société ensemble. Ces réseaux sont relativement sûr car ils sont quasiment toujours derrière une série de pare-feu ou coupés d'Internet et le chemin emprunté par les données ne quitte pas l'entreprise et est connu.

Sur Internet, on ne sait pas par où passent les données car les chemins changent. Ces données peuvent donc être écoutées ou interceptées. Il n'est donc pas envisageable de faire connecter deux LAN entre eux par Internet sans sécuriser le cheminement des données échangées.

Afin de sécuriser les échanges de données entre deux LAN, on peut recourir à deux alternative :

- relier les deux sites par une ligne spécialisée mais hors de prix.
- créer un réseau privé virtuel sécurisé autrement dit un VPN. On encapsule (en anglais tunneling) les données dans un tunnel crypté.

Nous nous intéresserons dans ce chapitre à la deuxième solution, ainsi nous verrons quelles sont les principales caractéristiques des VPN et les protocoles permettant leur mise en place.

2.2 Définition d'un VPN

VPN (Virtual Private Network) correspond à une interconnexion de réseaux locaux en utilisant un réseau publique (ex : Internet) via une technique de " tunnel" c'est-à-dire que seules les ordinateurs des réseaux interconnectés peuvent voir les données échangées. Nous parlons alors de réseau privé virtuel pour désigner le réseau ainsi artificiellement créé. [10]

2.3 Fonctionnement d'un VPN

Le VPN repose sur un protocole de tunnellisation, protocole qui encapsule dans son datagramme un autre paquet de données complet utilisant un protocole de communication différent et qui permet le passage de données cryptées d'une extrémité du VPN (machine client) à l'autre (serveur) grâce à des algorithmes cryptographiques.

Le terme tunnel est employé pour symboliser le fait que les données soient cryptées et de ce fait incompréhensible pour tous les autres utilisateurs du réseau public (ceux qui ne se trouvent pas aux extrémités du VPN).

Dans le cas d'un VPN établi entre deux machines, l'une sera un client VPN (élément permettant de chiffrer et de déchiffrer les données du côté utilisateur) et l'autre le serveur VPN (élément chiffrant et déchiffrant les données du côté de l'organisation). Ainsi lorsque le client nécessite d'accéder au réseau privé virtuel, sa requête va être transmise en clair au système passerelle, qui va se connecter au réseau distant (le serveur) par l'intermédiaire d'une infrastructure de réseau public, puis une fois connecté au serveur il va transmettre la requête de façon chiffrée.

Au cas où le serveur VPN est connecté avec d'autres machines sur son réseau local, le client pourra alors joindre ces machines par l'intermédiaire du serveur VPN. Pour répondre au client ces machines vont alors fournir les données au serveur VPN de leur réseau local qui va transmettre la réponse de façon chiffrée. A la réception sur le client VPN de l'utilisateur, les données seront déchiffrées, puis transmises à l'utilisateur. Pour émuler une liaison point à point, les données sont encapsulées, ou enrobées, à l'aide d'un en-tête qui contient les informations de routage pour leurs permettre de traverser le réseau partagé ou public jusqu'à leur destination finale.

Pour émuler une liaison privée, les données sont cryptées à des fins de confidentialité. Les paquets interceptés sur le réseau partagé ou public restent indéchiffrables sans clé de décryptage.

Ainsi, tous les utilisateurs passent par le même "portail", ce qui permet de gérer la sécurité des accès, ainsi que le trafic utilisé par chacun.

Le tunneling est l'ensemble des processus d'encapsulation, de transmission et de désencapsulation. [11]

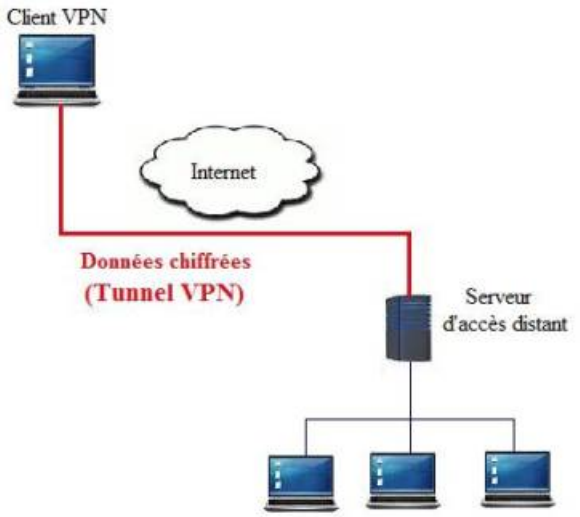

FIGURE 2.1 – connexion VPN entre un client et un serveur.

2.4 Objectifs et caractéristiques des VPN

- ***Confidentialité des données***
  Les VPN visent à protéger le contenu des messages contre toute interception par des sources non authentifiées ou non autorisées. La confidentialité est garantie grâce à l'encapsulation et au chiffrement effectués. [12]

- ***Intégrité des données***
  Pour garantir qu'aucune altération ou modification n'a été apportée aux données lors de leurs parcours entre la source et la destination, les réseaux privés virtuels utilisent des hachages. Un hachage ressemble à une somme de contrôle ou à un sceau garantissant que personne n'a lu le contenu, tout en étant plus robuste. [12]

- ***Authentification***
  Afin de garantir qu'un message provient d'une source authentique, que la personne avec qui la communication est établie est effectivement le destinataire escompté. Les réseaux privés virtuels peuvent utiliser des mots de passe, des certificats numériques

et des cartes à puce pour vérifier l'identité des parties à l'autre extrémité du réseau. [12]

- ***Gestion des clés***
  Les VPN assurent la génération, la distribution, le stockage et la suppression des clés de cryptage pour le client et pour le serveur avec différents mécanismes et protocoles. [12]
- ***Prise en charge multi protocoles***
  La solution VPN doit supporter les protocoles les plus utilisés sur les réseaux publics en particuliers IP. [12]

2.5 Catégories de VPN

Suivant les besoins on référence 3 types de VPN :

2.5.1 Le VPN d'accès

Il est utilisé pour permettre à des utilisateurs itinérants d'accéder au réseau de leurs entreprises. L'utilisateur se sert d'une connexion Internet afin d'établir une liaison sécurisé. [12]

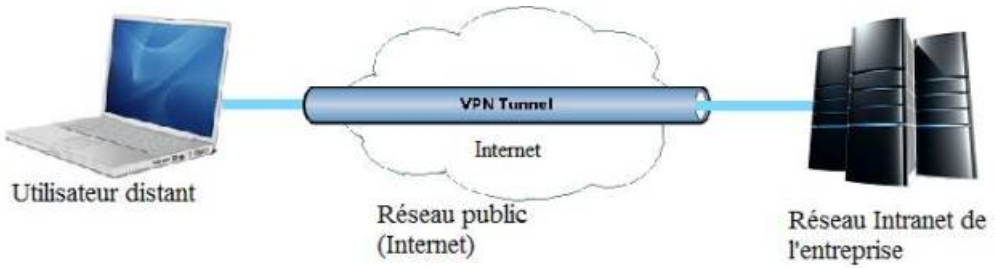

FIGURE 2.2 – VPN d'accès.

2.5.2 L'intranet VPN

Il est utilisé pour relier deux ou plusieurs intranets d'une même entreprise entre eux. Ce type de réseaux est particulièrement utile au sein d'une entreprise possédant plusieurs sites distants. Cette technique est également utilisée pour relier des réseaux d'entreprise, sans qu'il soit question d'intranet (partage de données, de ressources, exploitation de serveur distant, etc). [12]

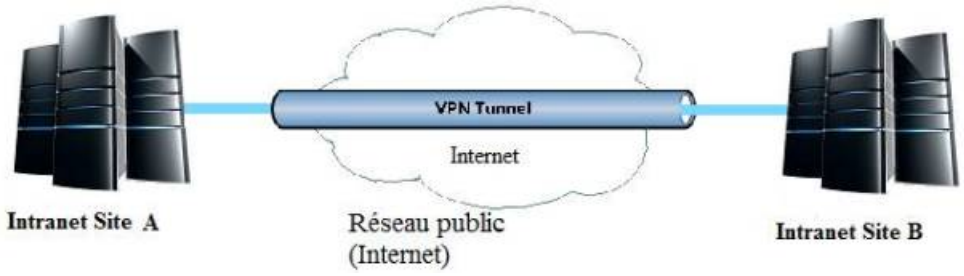

FIGURE 2.3 – VPN intranet.

2.5.3 L'extranet VPN

Une entreprise peut utiliser le VPN pour communiquer avec ses clients et partenaires. Elle ouvre alors son réseau local à ces derniers, dans ce cas il est nécessaire d'avoir une authentification forte des utilisateurs, ainsi qu'une trace des différents accès. De plus, seule une partie des ressources sera partagée, ce qui signifie une gestion rigoureuse des espaces d'échange. [12]

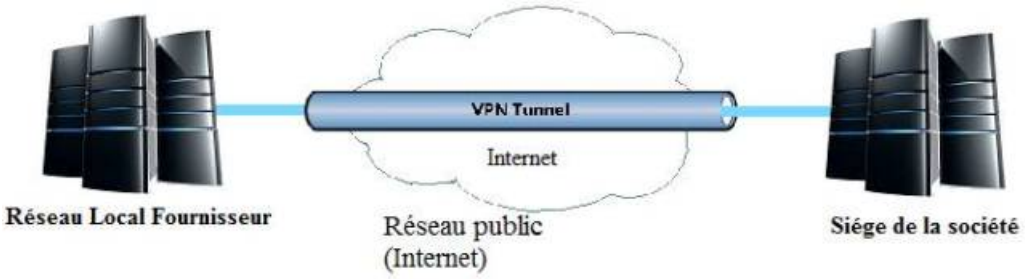

FIGURE 2.4 – VPN extranet.

2.6 Avantage des VPN

Les réseaux privés virtuels offrent les avantages suivants : [12]

1. **Économies**

   Les organisations peuvent utiliser un transport Internet tiers et rentable pour connecter les bureaux et les particuliers à distance au siège, à la différence des liaisons dédiées de réseau étendu qui est une solution trés couteuse. Grâce à la large bande, les réseaux privés virtuels réduisent les coûts de connectivité en augmentant la bande passante de connexions distantes.

2. **Sécurité**

   Les protocoles de chiffrement et d'authentification avancés protègent les données contre tout accès non autorisé.

3. **Évolutivité**

   Les réseaux privés virtuels utilisent l'infrastructure Internet dont les FAI et les opérateurs, facilitant l'ajout de nouveaux utilisateurs pour les entreprises. Ces dernières, quelle que soit leur taille, peuvent augmenter leurs capacités sans élargir sensiblement leur infrastructure.

4. **Simplicité**

   Utilise le circuit de télécommunication classique.

2.7 Les protocoles utilisés

Voici une présentation des protocoles les plus utilisés dans le cadre de VPN. Nous les avons classé selon leur place dans les couches OSI mais ce classement peut se révéler arbitraire pour certains d'entre eux qui recouvrent en fait plusieurs niveaux.

- **Le niveau 2 :** Ces VPN encapsulent les données dans des trames et ce sont ces trames qui seront véhiculées dans le tunnel dans une communication point à point. Nous sommes donc bien dans la couche 2 du modèle OSI. La plupart des protocoles situés ici sont progressivement délaissés au profit de protocoles plus souples comme pouvent l'être ceux des niveaux 3 à 7. [11]
  Parmi ces protocoles : PPTP, L2F, L2TP.
- **Le niveau 3 :** Nous retrouvons ici des protocoles opérant au niveau 3, donc au niveau paquet, ce qui les rend plus souple et explique leur succès croissant. [11]
  Parmi ces protocoles : IPSec.
- **Le niveau 4 :** OpenVPN en SSL. [11]

Les protocoles PPTP et L2TP dépendent des fonctionnalités du protocole PPP d'où la nécessité de voir tout d'abord son fonctionnement. [11]

Le protocole PPP(Point to Point Protocol) : défini dans la RFC 1661 appuyé de la RFC 2153, c'est un protocole qui permet de faire fonctionner l'IP et d'autres protocoles réseau à travers une ligne série, qui est généralement un lien utilisant des modems et les lignes téléphoniques (RTC, PSTN, connexion modem), ou un câble pour des connections Telnet. Il est full duplex et garantit l'ordre d'arrivée des paquets qu'il transmet encapsulés au travers de la liaison point à point. Il permet aussi de transférer des données sur un lien synchrone ou asynchrone. [13]

2.7.1 Le protocole PPTP

Protocole de couche 2 développé par Microsoft, 3Com, Ascend, US Robotics et ECI Telematics et defini par la RFC 2637. [10]

Son fonctionnement se base principalement sur le protocole PPP et consiste en deux flux de communication entre un client et un serveur, d'abord l'établissement d'une connexion normal à Internet par un modem avec PPP sur le port 1723 du serveur en TCP, vient ensuite une deuxième connexion, celle qui établit le VPN, qui représente en fait des trames PPP cryptées encapsulées dans un paqués IP par l'intermédiaire du protocole GRE qui est un protocole permettant d'encapsuler des protocoles de couche 2. [10]

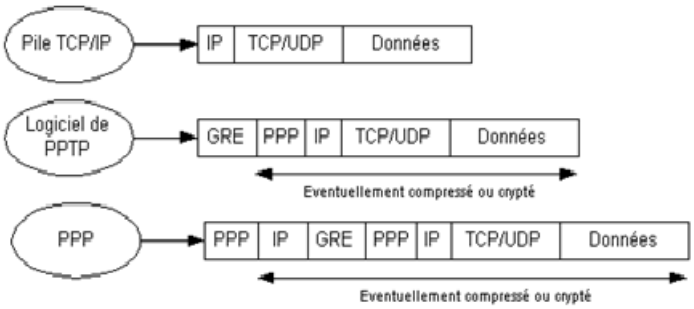

FIGURE 2.5 – Fonctionnement du protocole PPTP. [10]

PPTP ne fait que transporter des données, l'authentification se fait avec l'un des protocoles de PPP, on cite : [10]

- **Password Authentification Protocol (PAP) :** Décrit dans la RFC 1994, les informations d'authentification (nom d'utilisateur et mot de passe) transitent en clair, ce qui n'est pas l'idéal si l'on veut sécuriser au maximum.

- **Challenge Handshake Authentification Protocol (CHAP) :** Il consiste en un mécanisme d'authentification crypté, il est donc sécurisé. Un protocole basé sur ce dernier est aussi utilisé : MS-CHAP.

Le cryptage des données se fait avec Compression Control Protocol qui est aussi un protocole de PPP. Il utilise différents types de cryptage, symétriques ou asymétriques. Les algorithmes RSA, DES, RC4, etc. [10]

2.7.2 Le protocole L2TP

L2TP est une norme préliminaire de l'IETF (RFC 2661) issu de la convergence des protocoles PPTP et L2F, il est actuellement développé et évalué conjointement par Cisco Systems, Microsoft, Ascend, 3Com et d'autres acteurs clés du marché des réseaux.

Son fonctionnement se base sur l'utilisation de deux types de serveur, le LAC (L2TP Access Concentrator) et le LNS (L2TP Network Server). L'utilisateur se connecte d'abord à un LAC qui est un serveur se trouvant dans l'infrastructure du FAI de chaque utilisateur du VPN. Il s'authentifie à ce LAC et ce dernier communique ensuite le login et le mot de passe à un serveur RADUIS. Si les informations sont validées, cela permet au LAC de connaître le LNS auquel l'utilisateur peut se connecter pour être sur le VPN de son entreprise. Le LNS est un autre serveur se trouvant sur le réseau distant. Un tunnel est alors créé entre le LAC et le LNS qui est en fait des trames PPP encapsulées par l'intermédiaire de L2TP dans un entête UDP. [12]

L2TP n'intègre pas directement de protocole pour le chiffrement des données. C'est pourquoi L'IETF préconise l'utilisation conjointe d'IPsec et L2TP. [12]

Plus techniquement, voici l'encapsulation qu'engendre L2TP (de bas en haut, dans le cas d'un HTTP) illustré dans la figure suivante :

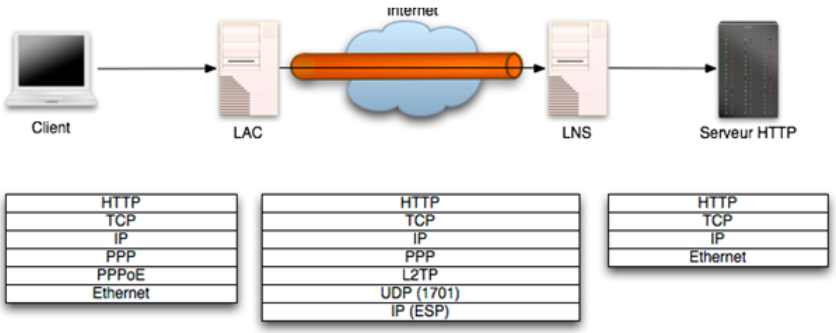

FIGURE 2.6 – Encapsulation engendré par L2TP. [12]

2.7.3 Le protocole IPSEC

IPSec est un protocole défini par l'IETF (RFC 2401) permettant de sécuriser les échanges au niveau de la couche réseau. Il s'agit en fait d'un protocole apportant des améliorations au niveau de la sécurité au protocole IP afin de garantir la confidentialité, l'intégrité et l'authentification des échanges. [14]

Les services de sécurité fournis par IPsec reposent principalement sur deux protocoles qui sont : [14]

- AH : " Authentication Header " (protocole numéro 51) dont la version la plus récente est normalisée par la RFC 4302. Il permet d'assurer l'intégrité, l'authentification et la protection contre le rejeu, mais ne gère pas la confidentialité, c'est pour ça qu'il est moins utiliser qu'ESP.
- ESP : " Encapsulation Security Payload " (protocole numéro 50) dont la version la plus récente est normalisée par la RFC 4303. Il permet d'assurer la confidentialité, l'intégrité et employé avec IKE, l'authentification des données échangées. Il garantit aussi une protection contre le rejeu. Il est possible d'utiliser uniquement les fonctions d'intégrité et d'authentification sans chiffrement (ce qui peut satisfaire la plupart des cas d'usage et justifie donc l'abandon d'AH).

Indépendamment du choix entre AH et ESP, il est possible d'utiliser IPsec dans deux modes distincts : le mode tunnel et le mode transport. [14]

1. **Le mode transport**

   Dans le mode transport, les données associées à AH ou à ESP viennent se greffer sur le paquet IP initial. Le paquet IP résultant contient un paquet AH ou ESP qui contient lui-même le contenu du paquet initial (un segment TCP par exemple). [14]

2. **Le mode tunnel**

   Dans le mode tunnel, un nouveau paquet IP est généré pour contenir un paquet AH ou ESP qui contient lui-même le paquet IP initial sans modification. Il y a donc deux en-têtes IP. L'en-tête externe sera utilisé pour le routage dès l'émission du paquet. L'en-tête interne, qui peut être chiffrée dans le cas où l'on utilise ESP avec le service de confidentialité, ne sera traité que par le destinataire (du paquet externe). [14]

   Nous prendrons le cas d'ESP afin d'illustrer la différence entre ces deux modes dans les figures suivantes :

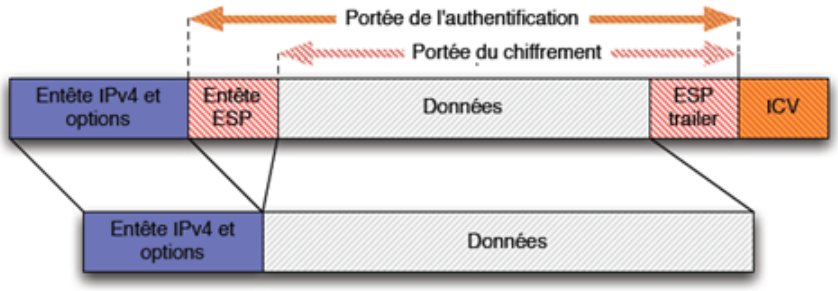

FIGURE 2.7 – Utilisation d'ESP en mode transport. ICV désigne l'Integrity Check Value, valeur utilisée par le mécanisme de contrôle d'intégrité. [14]

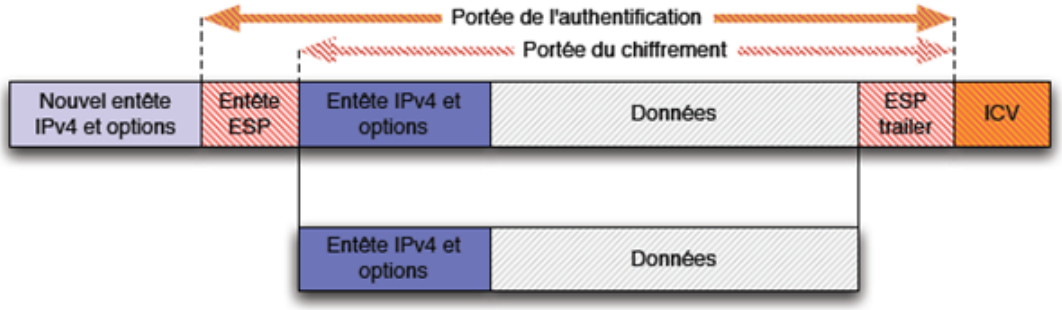

FIGURE 2.8 – Utilisation d'ESP en mode tunnel. [14]

Les protections offertes par IPSec sont basées sur des choix définis par l'administrateur du réseau par le biais de politiques de sécurité, c'est donc par elles que se fait la configuration d'IPSec. L'établissement de ces politiques de sécurité est associés à plusieurs mécanismes (SA, SAD, SPD) qui font appel à la cryptographie et utilisent donc un certain nombre de paramètres (échange préalable d'algorithmes utilisés, échange de clefs, mécanismes sélectionnés...), ces échanges préalables sont gérés automatiquement par le protocole IKE, qui est un protocole développé spécifiquement pour IPSec. [14]

2.7.4 Le protocole SSL

Le protocole SSL (Secure Socket Layer) a été développé par la société Netscape Communications Corporation pour assurer la sécurité des transactions sur Internet (notamment entre un client et un serveur). TLS (Transport Layer Security) est une évolution de SSL réalisée par l'IETF. SSL est un protocole qui s'intercale dans le modèle OSI entre la couche transport et application. [13]

Dans le passé, SSL était un protocole utilisé avec des applications spécifiques comme HTTP. Aujourd'hui il est capable de sécuriser les transactions de n'importe quelles applications à travers Internet et de créer des tunnels sécurisés (VPN) comme peut le faire IPSec, il est notamment implémenté en standard dans les navigateurs modernes depuis 1994 surtout pour la sécurisation des échanges commerciaux sur Internet. [13]

Le principale avantage de SSL face aux autres solutions VPN présentées jusqu'ici est qu'il ne nécessite coté client qu'un navigateur Internet classique. [13]

2.7.4.1 Fonctionnement de SSL

Le fonctionnement de SSL passe par deux protocoles :

1. Le protocole SSL Handshake débute une communication SSL :

– A la réception de la requête du client, le serveur lui envoie son certificat avec la liste des algorithmes qu'il souhaite utiliser. Le client vérifie la validité du certificat du serveur à l'aide de la clé publique de l'autorité de certification contenue dans son navigateur, il verifie aussi la date de validité du certificat et peut également consulter une CRL (Certificate Revocation List). Si le certificat passe toutes les vérifications, le client génère une clé symétrique qu'il chiffre avec la clé publique du serveur et la transmet à ce dernier. [13]
– Avant de commencer à échanger des données le serveur envoie au client un message que le client signera avec la clé privée de son certificat et renverra au serveur afin de s'authentifier, il lui enverra aussi son certificat afin que le serveur le vérifie. [13]

2. Le protocole SSL Records prend en charge l'échange de données cryptées à l'aide de clés dérivées de la clé préalablement échangée entre le client et le serveur. Ce protocole a différentes phases, on cite : [13]
   - Segmentation des paquets en paquets de taille fixe.
   - Compression (mais peu implémenté dans la réalité).
   - Ajout du résultat de la fonction de hachage composé de la clé de cryptage, du numéro de message, de la longueur du message, de données .
   - Chiffrement des paquets et du résultat du hachage à l'aide de la clé symétrique générée lors du Handshake.
   - Ajout d'un en-tête SSL au paquet.

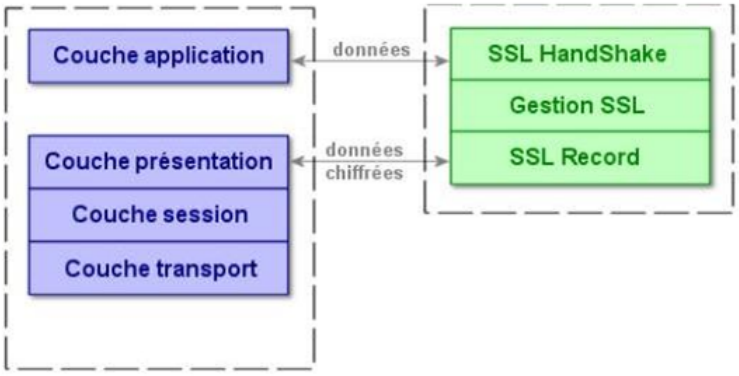

FIGURE 2.9 – SSL et le modèle OSI. [13]

2.7.4.2 La solution OPENVPN

Créé en 2002 par James Yonan, OpenVPN est un logiciel libre et gratuit permettant d'établir des VPN, il est basé sur le protocole SSL/TLS plus précisément OpenSSL pour assurer les fonctions de confidentialité, authentification, intégrité et non-répudiation,

l'authentification se fait à l'aide de certificats électroniques ou de couples de noms d'utilisateur/mot de passe. [15]

OpenVPN utilise un seul port tcp/udp pour tunneliser de manière sécurisée, des données à travers un réseau non sûr comme Internet trés pratique pour passer les pare-feu. [15]

Il est disponible avec une multitude d'environnements tel que Solaris, OpenBSD, FreeBSD, NetBSD, Linux (Debian, Redhat, Ubuntu, etc.), Mac OS X, Windows 2000, XP, Vista, 7, 8 et 10. [15]

1. **OpenSSL :**

   OpenSSL est une version gratuite et open source de SSL pour les tâches de chiffrement et d'authentification. C'est en fait une boîte à outils cryptographiques qui regroupe : [15]

   - Une bibliothèque de programmation en C permettant de réaliser des applications client/serveur sécurisées s'appuyant sur SSL/TLS.
   - Une commande en ligne (openSSL) permettant :
     - La création de clés RSA, DSA (signature).
     - La création de certificats X509.
     - Le calcul d'empreintes (MD5, SHA, RIPEMD160, ...).
     - Le chiffrement et déchiffrement (DES, IDEA, RC2, RC4, Blowfish, ...).
     - La réalisation de tests de clients et serveurs SSL/TLS.
     - La signature et le chiffrement de courriers (S/MIME).

2. **Mode de fonctionnement d'OpenVPN**

   OpenVPN peut être utilisé en mode TUN ou bien TAP. [15]

   - Le mode TAP "Bridged" permet d'établir un pont entre deux réseaux locaux, les deux réseaux seront connectés comme avec un commutateur, le client se verra attribuer la même adresse réseau que le réseau distant. Ce mode permet de transporter n'importe quel protocole réseau (IPv4 ou v6, IPX, etc) et accepte le trafic broadcast. [15]
   - Le mode TUN "routed" ne transporte que le protocole IP (y compris IPv6 depuis OpenVPN 2.2.3) et pas le trafic broadcast. Dans ce mode le client aura une adresse IP appartenant à un réseau different de celui du réseau local auquel il se connecte avec le VPN. Le client se connecte au serveur OpenVPN qui agit alors comme routeur pour relayer les paquets réseau entre le client VPN et le réseau local. [15]

2.8 Comparatif des différents protocoles

Nous allons aborder les points forts et les points faibles des solutions VPN citées précédemment. [13]

2.8.1 Le protocole PPTP

- **Avantages :**
  - Presque tous les matériels et OS sont compatibles avec le PPTP. Vous n'aurez besoin que d'un nom d'utilisateur, d'un mot de passe et de l'adresse du serveur.
  - Le protocole PPTP est très simple à utiliser et à mettre en place.
  - Le plus rapide des protocoles VPN car le cryptage des données se fait sur 128 bit.
- **Inconvénients :**
  - Cryptages des données sur 128 bits et non sur 256 bits ce qui offre une connexion sensiblement plus rapide mais moins sûre.
  - PPTP utilise le port TCP 1723 et avec une valeur GRE de 47. PPTP peut être facilement bloqué par un pare-feu.
  - L'implémentation Microsoft du PPTP possède de sérieux problèmes de sécurité. Le MSCHAP-V2 est vulnérable contre les attaques de type dictionnaire (Brute-force) et l'algorithme RC4 est aussi vulnérable face aux attaques de type Bit-Flipping.

2.8.2 Le protocole L2TP

- **Avantages :**
  - Toutes les versions de Windows à partir de 2000/XP et les MACs avec au moins OSX 10.3 ont un support natif de L2TP. La quasi-totalité des plates-formes mobiles iOS et Android ont un client pré-installé pour L2TP.
  - L2TP est facile à mettre en place.
- **Inconvénients :**
  - L2TP ne garantit pas la confidentialité des données et doit être associé avec un autre protocole pour le cryptage de données (IPSec). Dans ce cas il encapsule les données deux fois, rendant la connexion plus sécurisée mais beaucoup plus lente que les autres protocoles.
  - Le L2TP/IPSec qui est l'implémentation la plus fréquente de L2TP utilise le port UDP 500 pour l'échange des clés. Le port 50 est utilisé pour le cryptage via l'IPSec et le port UDP 1701 est utilisé pour la configuration initiale du L2TP. Enfin, le port UDP 4500 est utilisé pour le transfert NAT. On peut facilement bloquer le L2TP/IPSec parce qu'il se base uniquement sur des protocoles et des ports fixes.

2.8.3 Le protocole IPSEC

- **Avantages :**
  - Intégré dans la plupart des systèmes d'exploitation pour PC, périphériques mo-

biles et tablettes.
- L'IPSec support deux modes de cryptage, le Transport et le Tunnelling. Le cryptage utilise une clé de 256 bits. Il est sécurisé et ne présente pas de faille connu.

- **Inconvénients :**
  - IPSec est un protocole de couche 3. Pour être implémenté, il nécessite une modification de la stack IP au niveau du noyau de l'équipement IPSec. En raison de ce changement dans le noyau, chaque système d'exploitation (Cisco, Windows, Linux, etc. ...) requiert sa propre implémentation d'IPSec et c'est ce qui fait que IPsec authentifie les machines et non pas les utilisateurs.
  - A cause de la lourdeur des opérations de cryptage/décryptage, IPSec réduit les performances globales des réseaux. L'achat de périphériques dédiés couteux est souvent indispensable.
  - IPSec a besoin de changer des règles de Pare-feu (Firewall) pour autoriser les protocoles AH et ESP, et l'ouverture du port 500 pour IKE.
  - La conversion NAT peut casser un tunnel VPN parce qu'elle change l'adresse d'un paquet (et les valeurs de checksum), comme le protocole AH exerce un contrôle d'intégrité sur les entêtes IP, il est donc incompatible avec le NAT. Il faut alors séparer le traitement destiné pour le NAT et celui destiné pour IPSec. Pour régler ce problème, les fabricants de VPN ont commencé à intégrer des capacités de traduction IPSec NAT dans leurs produits.
  - La configuration de IPSec est souvent très complexe à cause des nombreux sous protocoles.

2.8.4 Le VPN SSL OpenVPN

- **Avantages :**
  - C'est une solution open source c'est-à-dire que tout le monde peut contrôler et vérifier le code source, cela permet ainsi de vérifier qu'il n'existe pas de porte dérobé.
  - Son fonctionnement est basée sur le protocole SSL qui est un protocole situé entre la couche transport et application du modèle OSI et va chiffrer la couche application. OpenVPN va se comporter comme une application standard. Il est implémenté dans l'espace utilisateur et a donc l'avantage d'être bien plus sûr.
  - Compression des données : OpenVPN permet la compression des données circulant dans le tunnel, ce qui permet d'améliorer la vitesse de transfert meilleure en mode UDP, plus lente en mode TCP (mais plus sûr).
  - Encapsulation des données dans un unique port TCP ou UDP : OpenVPN pourra donc facilement passer à travers un routeur/firewall sachant faire de la redirection de ports. En effet pour contourner un pare-feu, OpenVPN peut être configuré sur

le port TCP 443 et sera indissociable des requêtes HTTP sur SSL normales, ce qui le rend presque impossible à bloquer et comme la couche 3 (IP) n'est pas modifié par SSL, il n'y a aucun problème avec le protocole NAT.
- Portabilité : OpenVpn peut être exécuté sur la plupart des OS présent sur le marché (Windows 2000/XP, Linux, BSD, Solaris, et Mac OS X). Une infrastructure de tunnel pourra donc être créé entre ces différents systèmes.
- Un seul fichier de paramètre est nécessaire pour configurer OpenVPN. Ce fichier présente toujours le même format quel que soit le système d'exploitation, ce qui facilite grandement l'administration.
- Authentification mutuelles des deux points distants via les certificats et mots de passe : cela permet d'augmenter considérablement la sécurité des échanges et d'authentifier les utilisateurs et les hôtes en même temps.
- Confidentialité des données : OpenVPN laisse le choix entre plusieurs algorithmes de chiffrement et la taille des clefs pour sécuriser les communications.

- **Inconvénients :**
  - Le protocole OpenVPN n'est pas pris en charge par certains appareils mobiles, ce qui peut être un inconvénient important pour l'utilisation d'un VPN mobile.

2.9 Conclusion

Dans ce chapitre nous avons expliqué les notions générales associées au VPN et mis en évidence la forte concurrence entre les différents protocoles pouvant être utilisés pour leurs création. Néanmoins il est possible de distinguer quelques-uns sortant leur épingle du jeu, à savoir SSL, plus précisément la solution OpenVPN. En effet OpenVPN offre une sécurité robuste équivalente à IPSec qui est le standard sur le marché mais avec une configuration plus simple et une transmission plus rapide.
Dans le chapitre qui suivra nous nous intéresserons à la mise en oeuvre d'une solution VPN avec OpenVPN.

CHAPITRE 3

ETUDE PRÉALABLE ET PROPOSITION DE SOLUTION

3.1 Introduction

Dans ce chapitre, nous commencerons par une présentation globale du réseau de l'université de Bejaïa qui est l'infrastructure réseaux sur laquelle nous réaliserons notre projet, cela nous permettra d'identifier les besoins et les points faibles du réseau afin de proposer une solution adéquate et de comprendre la nécessité d'une telle solution.

3.2 Présentation globale du réseau intranet

Le réseau informatique de l'université de Bejaïa est constitué de quatre zones sur le campus Targa Ouzemour et de deux zone au campus Aboudaou, sa topologie physique est en étoile étendue (voir la figure suivante), chaque zone a l'architecture d'un arbre.

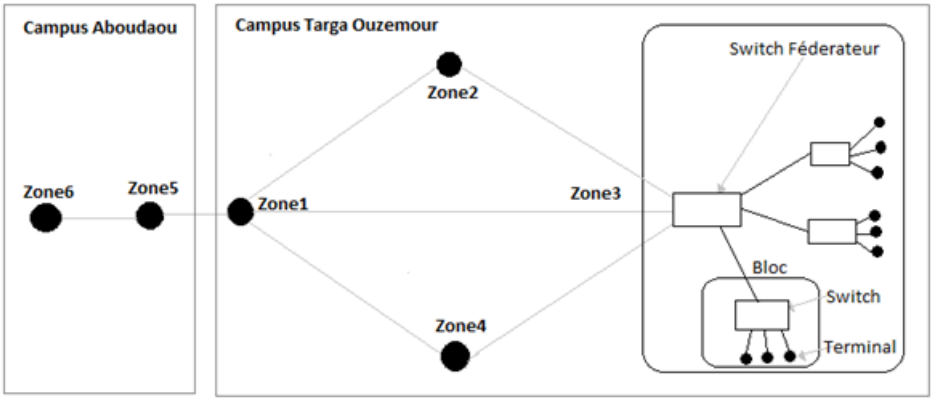

FIGURE 3.1 – La topologie physique du réseau local

La zone 1 est la plus importante car c'est là que se trouve le centre de calcul qui héberge la salle des administrateurs ainsi que tous les serveurs du réseau local. Tandis que le campus Aboudaou est connecté à cette zone par le backbone (fibre optique qui relie la zone 6 à la zone 3).

Chaque zone regroupe des blocs proches les uns des autres en terme physique.

3.2.1 Description détaillé de la zone 1

Notre projet se basera principalement sur la zone 1, elle permet la connexion en amont vers l'extérieur car c'est à son niveau que se trouve le routeur, on trouve aussi le pare-feu qui se charge de filtrer les paquets entrants. Ce système de pare-feu permet aussi le routage inter-LAN car l'une des ses interfaces est reliée directement au switch fédérateur qui offre une liaison vers la zone 2 et 4, ces deux dernières zones sont reliées à leur tour à la zone 3, comme l'illustre la figure 3.2.

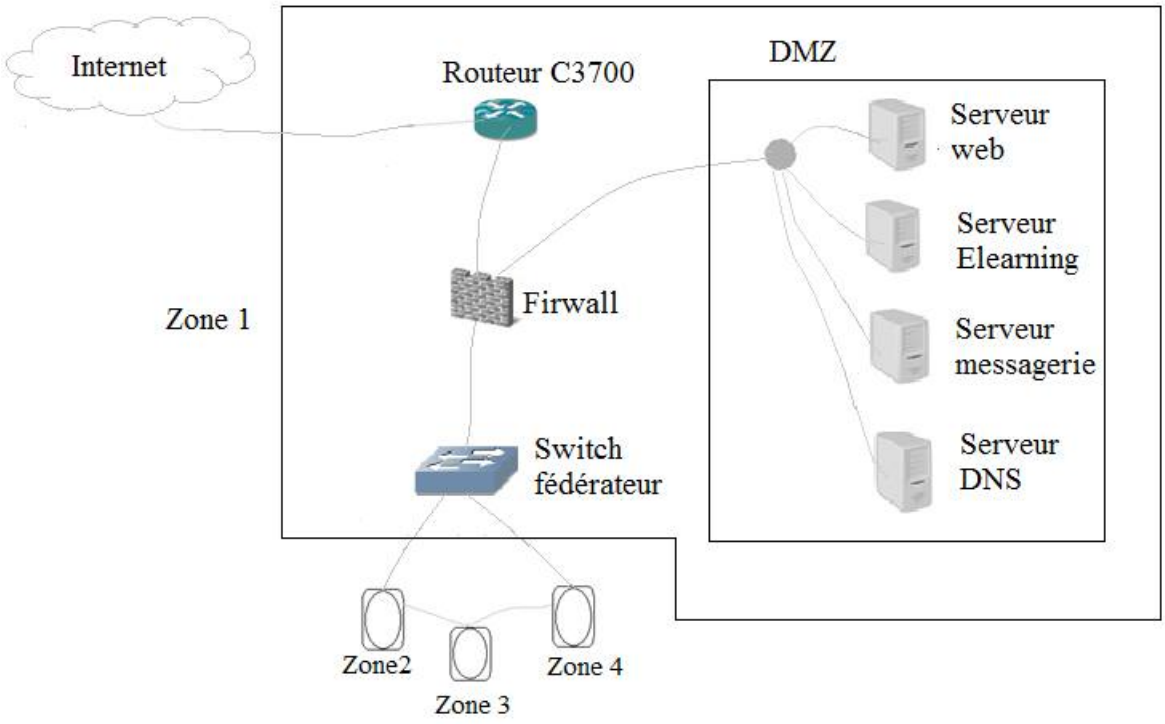

FIGURE 3.2 – Description de la zone1

3.3 Problématique

Ces derniers temps, de plus en plus de personnel et d'enseignants désirent accéder aux ressources internes du LAN depuis l'extérieur (leur domicile ou places publiques) dans le but de continuer leurs travail à distance comme s'ils étaient toujours dans les locaux. Ces ressources internes sont principalement les données/applications présentes sur le réseau de l'université (accès aux machines de calcul, accès aux machines du laboratoire, accès aux machines personnelles,...)

Pour des raisons de sécurité, ces demandes sont difficiles à accepter par les administrateurs informatiques du centre de calcul ; la crainte, très justifiée, que l'utilisateur se fasse dérober son mot de passe lors de sa connexion est forte (dans les applications courantes, le mot de passe circule en clair sur le réseau).

En plus la sécurité dans le réseau local est moindre, en effet n'importe quel utilisateur du réseau local qui possède un logiciel sniffer sur son pc pourra écouter le réseau (écouter le trafic DNS, l'accès aux bases de données...), il faudra donc trouver une solution qui garantit aussi la confidentialité des données pour les utilisateurs locaux.

3.4 Solution proposé

La solution idéale pour ce genre de demandes serait la mise en place d'un VPN pour les accès distants (host to LAN) qui permettra de créer un tunnel chiffré entre une machine sur Internet doté d'une adresse IP quelconque et une passerelle d'accès du réseau privé (réseau de l'université). Cela permettra d'assurer la confidentialité du trafic de l'utilisateur distant au LAN et celui du LAN à la DMZ et Internet en générale (les utilisateurs locaux pourront par exemple masquer le trafic DNS de leurs machines en se connectant au VPN).

Il existe de nombreuses solutions VPN sur le marché, néanmoins il y'a une solution qui a su tirer son épingle du jeu, on parle bien sûr de la solution SSL OpenVPN. Les raisons de ce choix sont nombreuses et ont déjà était expliquées en détails dans le chapitre précédent, on cite ici les principales :

- OpenVPN est une solution open source et gratuite.
- OpenVPN est facile à configurer et à mettre en place en plus d'offrir un niveau de sécurité équivalent à d'autres solutions plus complexes.
- OpenVPN permet d'authentifier les utilisateurs et pas seulement les machines.

Il existe deux produits OpenVPN :

- OpenVPN Access Server.
- OpenVPN Community Software.

Les deux ont les mêmes fonctionnalités et le même niveau de sécurité si ce n'est que OpenVPN Access Server se présente sous forme d'interfaces graphiques plus conviviales qu'OpenVPN Community Software qui s'utilise en ligne de commande. La configuration est donc plus facile avec OpenVPN Access Server, surtout pour les clients, en effet les clients n'auront besoin que d'un mot de passe et l'adresse du serveur pour télécharger l'application cliente qui leur permettra de se connecter au VPN, alors que dans OpenVPN Community Software il faut transporter physiquement les fichiers client (certificat, clé) sur la machine cliente pour lui permettre de se connecter, cependant OpenVPN Access Server nous permet d'avoir que deux utilisateurs connectés simultanément, pour avoir plus d'utilisateurs connectés en simultané il faut payer une licence, c'est le principal inconvénient qui nous a empêché d'opter pour cette solution qui est de ce fait la solution idéale pour une entreprise qui a les moyens.

Nous choisirons donc la solution OpenVPN Community Software, qui est une solution en ligne de commande. Nous mettrons en place les mêmes fonctionnalités qu'OpenVPN Access Server, c'est-à-dire l'authentification par certificat et mot de passe, nous essayerons aussi de faciliter la configuration pour les clients.

Une autre différence entre les deux solutions est qu'OpenVPN Access Server ne peut

pas être déployé sur un serveur Windows, une machine sous Windows ne peut être que cliente alors que ce n'est pas le cas pour OpenVPN Community Software qui peut être déployé en serveur sous Windows.

3.4.1 Architecture proposée

Après ces quelques suggestions sur le réseau actuel, nous aurons donc cette nouvelle architecture qui est représenté dans la figure suivante :

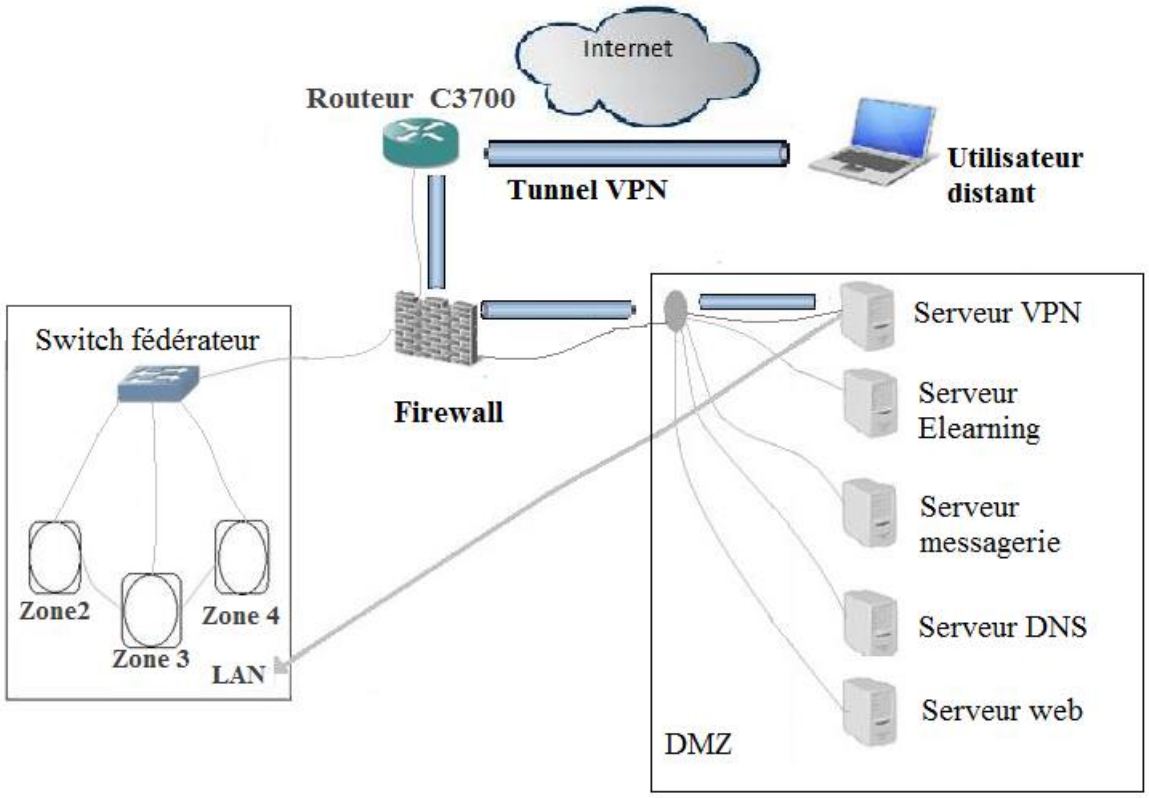

FIGURE 3.3 – Schéma de la nouvelle architecture possible

Le serveur VPN sera mis en place dans la DMZ afin de permettre aux clients de l'atteindre par Internet, il leur donnera accès au réseaux LAN en passant par un switch de niveau 3 qui est connecter à un firewall, qui protège l'accès au LAN, l'ajout de règles de routage sur le firewall est donc nécessaire pour permettre l'accès au LAN.

3.5 Conclusion

Cette études du réseau de l'université nous a permis d'identifier les besoins de l'architecture actuelle du réseau, et de suggérer une solution qui offre sans doute une meilleure sécurité, une meilleure souplesse pour le réseau et à moindre coût.

CHAPITRE 4

RÉALISATION DU VPN

4.1 Introduction

La mise en oeuvre d'un VPN d'accès (host to LAN) est la solution à laquelle nous avons abouti après la concrétisation de notre étude dans le chapitre précédent. Dans ce chapitre nous présentrons les outils utilisés et les principales étapes de configuration pour mettre en place un VPN host to LAN avec OpenVPN.

4.2 Présentation des outils utilisés

Afin de mettre en place notre solution nous avons utilisé les outils suivants :

- **OpenVPN** : présenté dans le chapitre 2.
- **LDAP** : présenté dans le chapitre 1.
- **PAM** : présenté dans le chapitre 1.
- **Postfix** : présenté dans le chapitre 1.

Nous utiliserons aussi un outil de virtualisation qui est VirtualBox que nous n'avons pas encore présenté.

4.2.1 VirtualBox

VirtualBox est un logiciel qui permet l'utilisation des ressources materielles d'un ordinateur (hôte) pour créer un ou plusieurs ordinateurs virtuels sur ce même ordinateur, ayant chacun leur propre système d'exploitation indépendant. [18]

Les systèmes invités et l'hôte fonctionnent en même temps, mais seul l'hôte a accès directement au véritable matériel de l'ordinateur. Les systèmes invités utilisent du materiel

simulés virtuellement par virtualbox. [18]

Le principal intérêt de VirtualBox est de tester des logiciels dans des environnements isolés et sécurisés. Il offre la possibilité de recommencer sans casser le système d'exploitation hôte. Ex : simuler des attaques sur le réseau, simuler un réseau...

4.3 Mise en place du VPN host to LAN avec OpenVPN

Nous allons maintenant passer à la mise en place du VPN avec le logiciel OpenVPN. Le serveur sera installer sur une machine Ubuntu 14.04, les clients seront quant à eux sous Windows. OpenVPN peut fonctionner avec plusieurs types d'authentification. Nous utiliserons l'authentification par clés et certificats, additionné à une authentification par mots de passe.

4.3.1 Mise en place de l'environnement

Pour commencer nous avons donc reproduit l'architecture du réseau de la zone 1 representée à la figure (figure 3.3) avec trois VM (machine virtuelle) (figure 4.1) :

- VM1 : fait office de serveur VPN sous Ubuntu qui sera dans la vrai architecture une machine de la DMZ.
- VM2 : qui fera office de client externe qui sera dans la vrai architecture un utilisateur sur Internet.
- VM3 : qui fera office de machine du réseau LAN qui sera dans la vrai architecture une machine se trouvant sur le LAN derrière le switch fédérateur.

Les commandes effectuées sur les VM seront les mêmes sur les machines réelles, Nous aurons donc l'architecture suivante :

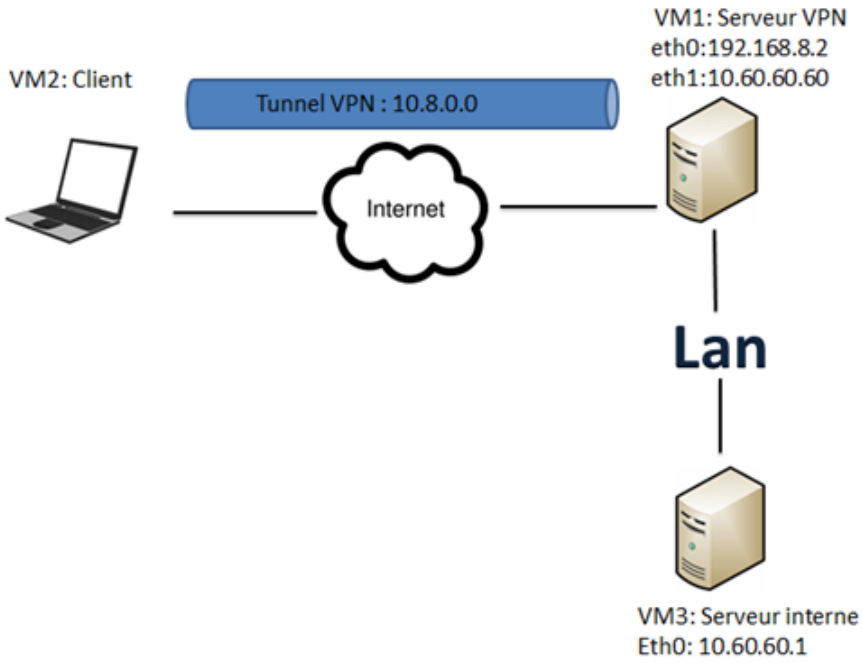

FIGURE 4.1 – Réseau réalisé avec les VM

La VM1 fera aussi office de passerelle pour le réseaux LAN, elle possède pour cela deux interfaces, une interface d'accès au VPN qui est eth0 (c'est par cette interface que les clients pourront joindre le VPN) et une interface connectée au réseau LAN eth1 qui aura donc la même adresse que le LAN. A noter que dans la vrai architecture c'est le routeur qui relie le réseau à Internet, la configuration qu'on va effectuer sur la VM1 sera la même à faire sur le serveur réel, il n'y a rien à configurer sur le routeur, la seule chose à ajouter sur l'architecture réelle par rapport à notre configuration sera des règles de routage sur le firewall pour permettre l'accès du VPN au LAN et du WAN (l'extérieur) au VPN.

4.3.2 Installation du serveur OpenVPN et création des clés et certificats

Taper dans un terminal de la VM1 (seveur) la commande sudo pour se mettre en mode root ou administrateur puis le mot de passe :

#sudo -s

Saisir la commande suivante pour installer OpenVPN sur le serveur :

#apt-get install openvpn

Créer le dossier easy-rsa dans le répertoire /etc/openvpn :

#mkdir /etc/openvpn/easy-rsa/

- **Easy-rsa**
  Easy-rsa est un paquetage de gestion de clefs RSA, fondé sur l'outil en ligne de commande OpenSSL, il contient en fait des scripts qui vont nous permettre de créer plus facilement les certificats clients et serveur. A l'installation d'OpenVPN le dossier easy-rsa sera automatiquement créé dans le répertoire /usr/share/doc/openvpn/examples/ c'est pour cela qu'il faut le copier dans notre répertoire de travaille, ceci n'est pas obligatoire mais ça va organiser notre travail d'avoir tout les fichiers qu'on utilise pour le VPN dans le même répertoire. [19]

Nous copions le contenu du dossier /usr/share/doc/openvpn/examples/easy-rsa dans notre répertoire de travail :
#cp -r /usr/share/doc/openvpn/examples/easy-rsa/* /etc/openvpn/easy-rsa/

Nous configurons en premier lieu l'authentification par clés et certificats. Pour cela, on modifie le fichier /etc/openvpn/easy-rsa/vars :
#nano /etc/openvpn/easy-rsa/vars

Le fichier vars correspond aux paramètres de création de clés et certificats. On laisse les valeurs par défaut de la plupart des options, on modifie principalement les lignes à la fin du fichier correspondant aux valeurs par défaut des informations qui seront dans les certificats. Par défaut elles sont :

```
export KEY_COUNTRY="US"
export KEY_PROVINCE="CA"
export KEY_CITY="SanFrancisco"
export KEY_ORG="Fort-Funston"
export KEY_EMAIL="me@myhost.mydomain"
```

Il faut les modifier par des valeurs correspondant au serveur, dans notre cas :

```
export KEY_COUNTRY="DZ"
export KEY_PROVINCE="06"
export KEY_CITY="bejaia"
export KEY_ORG="univ-bejaia.com"
export KEY_EMAIL="test-vpn@univ-bejaia.com"
```

L'étape suivante correspond à la génération de la clé et du certificat du serveur. On suit les étapes suivantes :

Nous nous déplaçons dans le répertoire de configuration :
#cd /etc/openvpn/easy-rsa/

Nous chargeons les valeurs par défaut du fichier *vars* :
#source vars
Nous effaçons la configuration existante au cas où il y en a une :
#./clean-all

Puis c'est la construction des paramètres Diffie-Hellman utilisés pour l'échange de clés entre le serveur et le client, on peut choisir de générer des paramètres pour une clé sur 1024 bit ou autre et cela en changeant le paramètre KEY_SIZE = dans le fichier vars.
#./build-dh
Génération de la clé privée (ca.key) et le certificat (ca.crt) de l'autorité de certification qui va signer les certificats serveur et clients.
#./pkitool -initca
Génération de la clé privée du serveur (server.key) et son certificat (server.crt).
#./pkitool –server server

Autre sécurité, tant qu'à faire. On ne présente plus les attaques "par déni de service" (DoS, pour Denial of Service) qui se sont popularisées ces dernières années. OpenVPN utilise (si on lui demande de le faire) un système ingénieux : le HMAC (Hash-based Message Authentication Code). C'est un code d'authentification partagé entre le serveur et le client, connu des deux dès le départ, qui précède la réelle authentification du client auprès du serveur. Techniquement, si un client "tiers" demande à s'authentifier auprès de notre serveur sans entamer le dialogue par l'envoi de ce HMAC, OpenVPN ne va même pas entamer la procédure : ce client n'est pas connu, on l'ignore, on ne lance pas la grosse machinerie, donc on ne surcharge pas le serveur. [20]

Pour mettre en place ce HMAC on doit exécuter la commande suivante :
#openvpn -genkey -secret keys/ta.key

Toute ces clés seront créées dans le répertoire ***/etc/openvpn/easy-rsa/keys/*** qui contiendra donc pour le moment : **server.key, server.crt, ca.key, ca.crt, ta.key, dh2048.pem**

Nous copions ensuite les clés et les certificats utiles pour le serveur dans le répertoire **/etc/openvpn/** :
#sudo cp keys/ca.crt keys/ta.key keys/server.crt keys/server.key keys/dh2048.pem /etc/openvpn/

Puis nous génèrons un répertoire **/etc/openvpn/jail** dans lequel le processus OpenVPN

sera chrooté[1] (afin de limiter les dégâts en cas de faille dans OpenVPN)

#mkdir /etc/openvpn/jail

Puis un autre répertoire **(/etc/openvpn/clientconf)** qui contiendra la configuration des clients :

 # mkdir /etc/openvpn/clientconf

4.3.3 Configuration du serveur OpenVPN

Nous créons ensuite le fichier de configuration à proprement dit du serveur toujours dans le répertoire **/etc/openvpn/** :

#nano server.conf

Nous l'éditons avec la configuration suivante (les lignes commençant par # sont des commentaires) :

```
mode server
proto tcp #Le protocole utilisé.
port 443 #Le numero de port à utilisé.
dev tun #Type de VPN, tun (tunnel IP) ou tap (tunnel Ethernet/bridge).

# Les chemins vers les clés utilisées par le serveur.
ca ca.crt
cert server.crt
key server.key
dh dh2048.pem
tls-auth ta.key 1

#Paramétre à ne pas modifier.
key-direction 0

#Choix de l'algorithme de chiffrement symétrique.
cipher AES-256-CBC

#Le réseau du tunnel VPN au sein duquel nous allons attribuer des IP aux
   clients.
server 10.8.0.0 255.255.255.0

#Route vers le réseau local.
```

1. Chroot : le système fichier d'un *NIX est construit autour d'une racine (le /) sur laquelle les partitions sont ensuite "montées" formant ainsi l'espace de fichier accessible. Cette racine est la référence pour tous les chemins absolus utilisés par un processus lui permettant d'accéder aux fichiers (librairies, configurations, etc.) qui lui sont nécessaires. Cette racine est en fait un paramètre du processus qu'il est parfaitement possible de modifier grâce à l'utilitaire chroot Pour faire croire à ce processus que le dossier que nous lui avons arbitrairement fixé comme racine, est l'origine de tous ses chemins absolus. Cela permet par exemple de lancer des processus critiques dans un dossier isolé du reste du système de sorte à rendre plus difficile (mais pas impossible !) la compromission du reste du système de fichier en cas d'exploitation d'une faille de sécurité. C'est ce que l'on appelle mettre en prison le processus (jail).

```
push "route 10.60.0.0 255.255.0.0"

#Pour faire passer tout le traffic client à travers le VPN.
push "redirect-gateway def1 bypass-dhcp"

#Routes vers des serveurs DNS on peut mettre n'importe lequel.
push "dhcp-option DNS 208.67.222.222"
push "dhcp-option DNS 208.67.220.220"

#Pour maintenir la connexion entre le serveur et le client ( ping toutes
    les 10 secondes,
#considéré comme down après 120 secondes sans réponse).
keepalive 10 120

#Resteindre les privilèges du processus OpenVPN.
user nobody
group nogroup

#Chrouter le processus OpenVPN dans le répertoire jail.
chroot /etc/openvpn/jail

#Ne pas relire les fichiers de clés cryptographiques suite au redémarrage.
#Sur signal SIGUSR1 ou sur --ping-restart.
persist-key
# Ne pas fermer et rouvrir le périphérique TUN/TAP ou exécuter les scripts.
# up/down suite au redémarrage sur signal SIGUSR1 ou sur --ping-restart.
persist-tun

#Utiliser la compression dynamique LZO.
comp-lzo

# Régler le niveau de traces à 3. Régler ce qui doit être affiché dans le
    log.
#Comme les informations de connexion et autre.
verb 3

#Ne répète pas plus de 20 fois un message dans le log.
mute 20

#Permet de voir l'état des clients connectés dans le fichier
# openvpn-status.log.
status openvpn-status.log
```

Explication : On a donc un serveur VPN SSL routé basé sur le protocole TCP et utilisant le port HTTPS (443) afin de maximiser son accessibilité depuis des réseaux

sécurisés par des Firewalls. Les clients obtiendrons une nouvelle adresse IP dans le range 10.8.0.0/24.

nous testons la configuration en saisissant les commandes suivantes :

#cd /etc/openvpn/

#openvpn server.conf

Si la configuration du serveur est un succès on doit obtenir les messages suivants :

```
Thu Jun 16 14:11:31 2016 OpenVPN 2.3.2 i686-pc-linux-gnu [SSL (OpenSSL)] [
   LZO] [EPOLL] [PKCS11] [eurephia] [MH] [IPv6] built on Dec  1 2014
Thu Jun 16 14:11:31 2016 PLUGIN_INIT: POST /usr/lib/openvpn/openvpn-plugin-
   auth-pam.so '[/usr/lib/openvpn/openvpn-plugin-auth-pam.so] [openvpn]'
   intercepted=PLUGIN_AUTH_USER_PASS_VERIFY
Thu Jun 16 14:11:31 2016 Diffie-Hellman initialized with 2048 bit key
Thu Jun 16 14:11:31 2016 Control Channel Authentication: using 'ta.key' as
   a OpenVPN static key file
Thu Jun 16 14:11:31 2016 Outgoing Control Channel Authentication: Using 160
    bit message hash 'SHA1' for HMAC authentication
Thu Jun 16 14:11:31 2016 Incoming Control Channel Authentication: Using 160
    bit message hash 'SHA1' for HMAC authentication
Thu Jun 16 14:11:31 2016 Socket Buffers: R=[87380->131072] S
   =[16384->131072]
Thu Jun 16 14:11:31 2016 ROUTE_GATEWAY 192.168.43.1/255.255.255.0 IFACE=
   eth0 HWADDR=08:00:27:66:2a:8a
Thu Jun 16 14:11:31 2016 TUN/TAP device tun0 opened
Thu Jun 16 14:11:31 2016 TUN/TAP TX queue length set to 100
Thu Jun 16 14:11:31 2016 do_ifconfig, tt->ipv6=0, tt->
   did_ifconfig_ipv6_setup=0
Thu Jun 16 14:11:31 2016 /sbin/ip link set dev tun0 up mtu 1500
Thu Jun 16 14:11:32 2016 /sbin/ip addr add dev tun0 local 10.8.0.1 peer
   10.8.0.2
Thu Jun 16 14:11:32 2016 /sbin/ip route add 10.8.0.0/24 via 10.8.0.2
Thu Jun 16 13:11:32 2016 chroot to '/etc/openvpn/jail' and cd to '/'
   succeeded
Thu Jun 16 13:11:32 2016 GID set to nogroup
Thu Jun 16 13:11:32 2016 UID set to nobody
Thu Jun 16 13:11:32 2016 Listening for incoming TCP connection on [undef]
Thu Jun 16 13:11:32 2016 TCPv4_SERVER link local (bound): [undef]
Thu Jun 16 13:11:32 2016 TCPv4_SERVER link remote: [undef]
Thu Jun 16 13:11:32 2016 MULTI: multi_init called, r=256 v=256
Thu Jun 16 13:11:32 2016 IFCONFIG POOL: base=10.8.0.4 size=62, ipv6=0
Thu Jun 16 13:11:32 2016 MULTI: TCP INIT maxclients=1024 maxevents=1028
Thu Jun 16 13:11:32 2016 Initialization Sequence Completed
```

Les messages du début ne sont pas important, il faut juste que la séquence se termine par : Initialization Sequence Completed.

La configuration de notre serveur est donc un succès, nous pouvons à présent le démarrer en saisissant la commande :

#service openvpn start

```
root@server-VirtualBox:/etc/openvpn# service openvpn start
 * Starting virtual private network daemon(s)...
                                                                    *
     Autostarting VPN 'server'
```

Pour s'assurer que le serveur VPN est vraiment en écoute on saisit la commande suivante :

```
root@server-VirtualBox:/etc/openvpn# netstat -ln|grep 443
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN
```

Nous pouvons voir que le VPN est en écoute sur le port tcp numéro 443.
Nous constatons alors l'apparition de l'interface réseau virtuelle tun0 en tapant la commande ifconfig dans un terminal :

```
tun0      Link encap:UNSPEC  HWaddr
   00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet adr:10.8.0.1  P-t-P:10.8.0.2  Masque:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          Packets reçus:0 erreurs:0 :0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:100
          Octets reçus:0 (0.0 B) Octets transmis:0 (0.0 B)
```

A ce stade les machines clientes vont pouvoir se connecter au serveur VPN. Par contre impossible d'aller plus loin que ce dernier car l'adresse 10.8.0.x ne sera par routée en dehors du serveur. Il faut donc configurer le serveur pour qu'il joue le rôle de routeur entre l'interface VPN (tun0) et l'interface physique (eth0) et qu'il fasse le NAT entre les adresses en 10.8.0.x et son adresse IP réelle.

Pour cela nous tapons la commande suivante :
#sh -c 'echo 1 > /proc/sys/net/ipv4/ip_forward'

Pour rendre ce paramétrage de routage permanant (même après un reboot), il faut ajouter la ligne suivante au fichier **/etc/sysctl.conf** :
net.ipv4.ip_forward = 1

Ça y est la configuration du serveur est terminé pour notre configuration sur la machine virtuelle. Pour une configuration sur l'architecture réelle on ajoute en plus des règles de routage sur le firewall pour permettre l'accès des clients au VPN et du VPN au LAN.

4.3.4 Création d'un compte client OpenVPN

Nous allons créé un client qui a pour nom par exemple client1, il suffit alors d'exécuter les commandes suivantes sur le serveur (VM1) :

#cd /etc/openvpn/easy-rsa
#source vars
#./build-key client1

Le script ./build-key va générer 3 fichiers dans le répertoire **/etc/openvpn/easy-rsa/keys** :

- client1.crt : certificat pour le client
- client1.csr : certificat à garder sur le serveur
- client1.key : clé pour le client

Nous copions les fichiers nécessaires au client créé dans un sous répertoire du répertoire **/etc/openvpn/clientconf/** qu'on appellera client1 :
 # mkdir /etc/openvpn/clientconf/client1/
 # cp /etc/openvpn/ca.crt /etc/openvpn/ta.key keys/client1.crt keys/-client1.key /etc/openvpn/clientconf/client1/

Nous allons ensuite dans le répertoire /etc/openvpn/clientconf/client1/ :
#cd /etc/openvpn/clientconf/client1/

Puis nous créons le fichier client.conf, fichier de configuration du client qui va lui permettre de se connecter au serveur :

```
client
dev tun
proto tcp-client
#addresse ip de l'interface où joindre le serveur et le numéro de port.
remote 192.168.8.2 443

#Si l'hôte ne parvient pas à résoudre l'adresse pour le champ remote,
   essayer de nouveau la résolution durant n secondes
#avant de finir en échec, ici n correspend à infinite.
resolv-retry infinite

cipher AES-256-CBC
# Clés
ca ca.crt
cert client1.crt
```

```
key client1.key
tls-auth ta.key 1
key-direction 1
# Sécurité
nobind
persist-key
persist-tun
comp-lzo
verb 3
```

Explication : Il faut bien s'assurer que les options sont identiques entre client et serveur (compression, port, protocole, chiffrement...), car une seule erreur et ça ne fonctionnera pas. Quant au paramètre remote on doit mettre l'adresse de l'interface où joindre le serveur, ici on a mis une adresse privée car on a pas d'adresse publique fix pour joindre notre serveur sur Internet, le vpn est simulé en local. Dans la machine réelle on mettra l'adresse publique du serveur ou bien son nom d'hôte car celui-ci se trouve derrière une passerelle (le routeur) et a une adresse fix.

Pour assurer la compatibilité avec le client Windows OpenVPN, on fait une copie du fichier client.conf vers client.ovpn :
#sudo cp client.conf client.ovpn

On devrait ainsi avoir les fichiers suivants dans le répertoire **/etc/openvpn/clientconf/-client1/** :

- ca.crt : certificat du serveur.
- client.conf : fichier de configuration du client OpenVPN (Linux, BSD, MacOS X).
- client.ovpn : fichier de configuration du client OpenVPN (Windows).
- client1.crt : certificat du client.
- client1.key : clé du client.
- ta.key : clé pour l'authentification HMAC.

Il ne reste plus qu'à mettre ces fichiers dans une archive ZIP à transmettre sur le PC client par exemple avec une clé USB :
#sudo zip client1.zip *.*

4.3.4.1 Configuration d'un client sous windows

Sur la machine cliente (VM2) afin de se connecter au serveur VPN, il suffit d'installer l'application VPN client OpenVPN GUI en la téléchargeant gratuitement sur le site officiel :

http://openvpn.net/index.php/open-source/downloads.html

Puis de prendre le zip client client1.zip téléchargé préalablement à partir du serveur et de le décompresser dans le répertoire d'installation de l'application OpenVPN GUI.

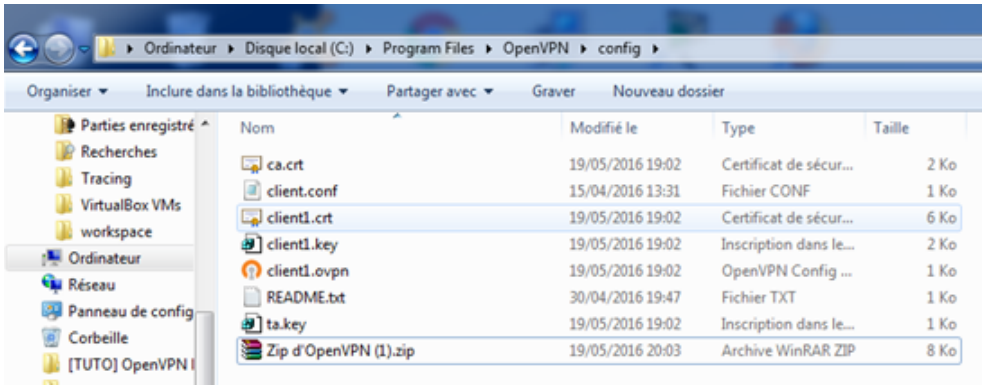

FIGURE 4.2 – Répertoire de configuration d'OpenVPN sur le client

Voilà il ne reste plus qu'à se connecter. Pour cela nous aurons qu'à cliquer sur l'icône de OpenVPN GUI apparu dans le bureau :

FIGURE 4.3 – Icone de l'application OpenVPN GUI

Cela va créer la petite icone suivante dans la barre des tâches de Windows, nous aurons alors qu'à cliquer dessus avec le bouton droit puis sur connecter :

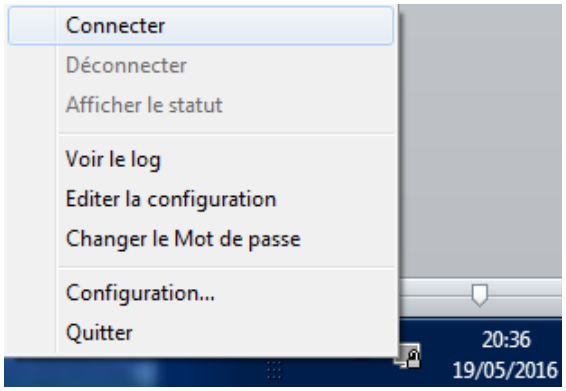

FIGURE 4.4 – Connection au serveur OpenVPN

Le client se connecte alors au serveur VPN comme on voit sur la figure suivante :

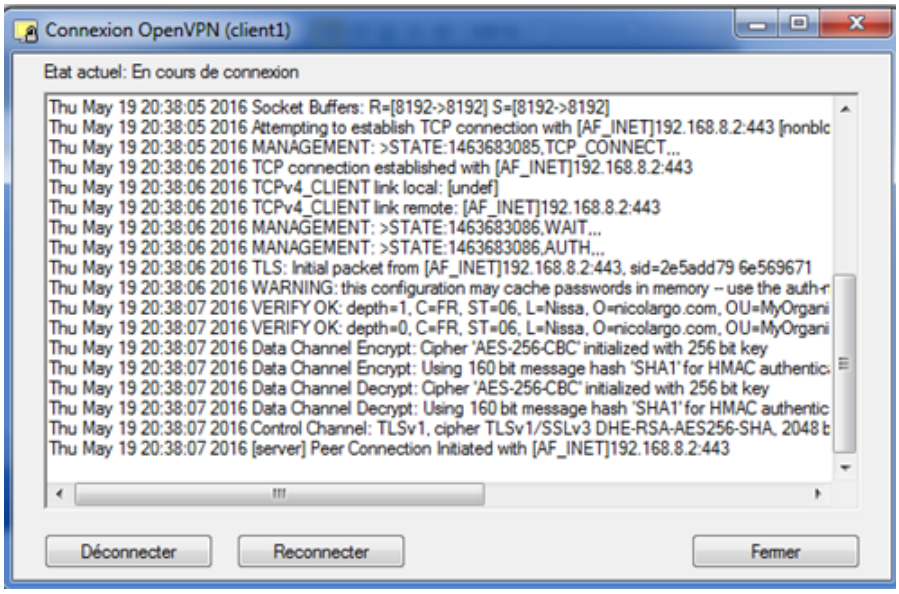

FIGURE 4.5 – Etablissement de la connection

La connexion s'est établis avec succès quand nous voyons l'icône de la barre des tâches devenir verte, on a alors une nouvelle adresse ip assignée par le serveur.

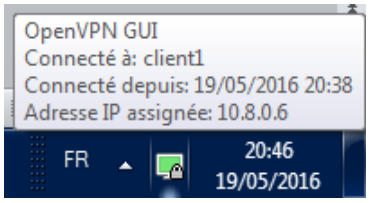

FIGURE 4.6 – Assiagnation d'une adresse

Pour voir sur la machine cliente si nous passons vraiment par le VPN pour notre connexion Internet nous faisons un tracert en ligne de commande :

```
C:\Users\msia>tracert www.google.fr

Détermination de l'itinéraire vers www.google.fr [41.201.129.25]
avec un maximum de 30 sautsă:

  1     6 ms     4 ms     5 ms  10.8.0.1
  2     *        *        *     Délai d'attente de la demande dépassé.
  3     *        *        *     Délai d'attente de la demande dépassé.
  4     *      963 ms   620 ms  10.110.1.49
  5   260 ms   228 ms   227 ms  10.110.1.49
  6   100 ms    66 ms    75 ms  172.17.3.25
  7   115 ms   149 ms    81 ms  172.29.23.1
  8    79 ms    69 ms   342 ms  10.104.25.5
  9     *      510 ms   669 ms  10.104.25.6
 10   157 ms   181 ms   234 ms  41.201.129.25

Itinéraire déterminé.
```

Nous voyons ici que la première route de l'itinéraire est l'adresse 10.8.0.1, qui est l'adresse de notre serveur VPN, le client passe donc bien par notre serveur VPN.
Nous essayons maintenant de pinger une machine du réseau local pour voir si on y accède.

```
C:\Users\msia>tracert 10.60.60.1

Détermination de l'itinéraire vers 10.60.60.1 avec un maximum de 30 sautsă:

  1     6 ms     5 ms     4 ms  10.8.0.1
  2     8 ms     5 ms     5 ms  10.60.60.1

Itinéraire déterminé.
```

Nous voyons bien qu'on accède à la machine du réseau local et qu'on passe par le serveur VPN 10.8.0.1 pour y accéder.

4.3.5 Administration et révocation de certificat

Nous pouvons révoquer le certificat d'un client on tapant la commande :
#source ./var
#./revoke-full nomducertificat

Un fichier nommé **crl.pem** est créé dans le dossier **/etc/openvpn/easy-rsa/keys/**, il contient la liste des certificats révoqués.

Nous copions le fichier **crl.pem** dans le répertoire chrouté **/etc/openvpn/jail** car la directive **crl-verify** regarde par défaut dans ce répertoire.
#cp /etc/openvpn/easy-rsa/keys/crl.pem /etc/openvpn/jail/

Nous ajoutons ensuite la ligne suivante dans le fichier de configuration serveur **server.conf** :
crl-verify /etc/openvpn/jail/crl.pem

Ainsi le serveur VPN vérifiera la liste des clients révoqués à chaque connexion client en consultant le fichier **crl.pem.**

Sur le serveur, nous pouvons voir les clients connectés en tapant la commande suivante :
 # nano /etc/openvpn/openvpn-status.log

```
OpenVPN CLIENT LIST
Updated,Thu Jun 16 14:17:01 2016
Common Name,Real Address,Bytes Received,Bytes Sent,Connected Since
client1,192.168.8.101:52110,1533710,3148927,Thu Jun 16 13:56:14 2016
ROUTING TABLE
Virtual Address,Common Name,Real Address,Last Ref
10.8.0.6,client1,192.168.8.101:52110,Thu Jun 16 14:16:47 2016
GLOBAL STATS
Max bcast/mcast queue length,0
END
```

nous voyons ainsi que le client **client1** est actuellement connecté.
Nous pouvons voir aussi les certificats créés et révoqués en tapant la commande :
#nano /etc/openvpn/easy-rsa/keys/index.txt

```
V       260407175131Z           01      unknown /C=DZ/ST=06/L=bejaia/O=
    univ-bejaia.com/OU=MyOrganizationalUnit/CN=server/name=EasyRSA/
```

```
    emailAddress=test-vpn@univ-bejaia.com
V       260407182047Z           02      unknown /C=DZ/ST=06/L=bejaia/O=univ
    -bejaia.com/OU=MyOrganizationalUnit/CN=pcportablenicolargo/name=EasyRSA/
    emailAddress=test-vpn@univ-bejaia.com
V       260511162924Z           03      unknown /C=DZ/ST=06/L=bejaia/O=univ
    -bejaia.com/OU=MyOrganizationalUnit/CN=bob/name=EasyRSA/emailAddress=
    test-vpn@univ-bejaia.com
V       260511165232Z           04      unknown /C=DZ/ST=06/L=bejaia/O=univ
    -bejaia.com/OU=MyOrganizationalUnit/CN=k/name=EasyRSA/emailAddress=test-
    vpn@univ-bejaia.com
V       260511173252Z           05      unknown /C=DZ/ST=06/L=bejaia/O=univ
    -bejaia.com/OU=MyOrganizationalUnit/CN=l/name=EasyRSA/emailAddress=test-
    vpn@univ-bejaia.com
V       260511180047Z           06      unknown /C=DZ/ST=06/L=bejaia/O=univ
    -bejaia.com/OU=MyOrganizationalUnit/CN=lynda/name=EasyRSA/emailAddress=
    test-vpn@univ-bejaia.com
V       260511181405Z           07      unknown /C=DZ/ST=06/L=bejaia/O=univ
    -bejaia.com/OU=MyOrganizationalUnit/CN=m/name=EasyRSA/emailAddress=test-
    vpn@univ-bejaia.com
V       260517180213Z           08      unknown /C=DZ/ST=06/L=bejaia/O=univ
    -bejaia.com/OU=MyOrganizationalUnit/CN=client1/name=EasyRSA/emailAddress
    =test-vpn@univ-bejaia.com
V       260521124002Z           09      unknown /C=DZ/ST=06/L=bejaia/O=univ
    -bejaia.com/OU=MyOrganizationalUnit/CN=bobi/name=EasyRSA/emailAddress=
    test-vpn@univ-bejaia.com
R       260521170218Z   160523195858Z   0A      unknown /C=DZ/ST=06/L=
    bejaia/O=univ-bejaia.com/OU=MyOrganizationalUnit/CN=client2/name=EasyRSA
    /emailAddress=test-vpn@univ-bejaia.com
```

On voit sur notre fichier index.txt que nous avons créé 10 certificats en tout et que le numéro 10 qui s'appelle client2 est révoqué, on voit bien la lettre R qui indique cela au début de la ligne.

4.3.6 Authentication centralisée avec LDAP

4.3.6.1 Mise en place de l'annuaire LDAP

Avec la configuration que nous avons fait précédemment les clients se connecte sans problème et les connexions sont chiffrées, cependant l'authentification par certificat permet d'authentifier juste la machine pas l'utilisateur, il faut donc trouver un moyen d'authentifier l'utilisateur en cas de vol par exemple de la machine.

Pour cela on ajoutera une authentification par login et mot de passe en plus de l'authentification par certificat, cette authentification sera centralisée grâce à un annuaire

LDAP, comme ça les clients pourront se connecter au VPN avec le même mot de passe et login qui leur permet déjà de s'authentifier aux différents services du LAN de l'université, cela rendra aussi la tâche plus facile pour l'administrateur d'avoir une seule base d'utilisateurs à gérer au lieu de plusieurs.

OpenLDAP version gratuite et open source du service LDAP sera installer sur le serveur (VM1), dans l'architecture réelle le service LDAP peut s'installer sur la même machine que le serveur VPN ou bien une autre.

Pour installer OpenLDAP et tous les package requis sur la machine serveur (VM1) tapez :
#apt-get install ssh nmap
#apt-get install slapd ldap-utils

Le mot de passe de l'administrateur de l'annuaire sera demandé durant l'installation. On definit ce mot de passe et on valide. L'installation se termine.
Nous testons si OpenLDAP est en marche sur la machine :
#nmap -p 389 localhost

```
Starting Nmap 6.40 ( http://nmap.org ) at 2016-06-16 15:27 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00077s latency).
PORT    STATE SERVICE
389/tcp open  ldap

Nmap done: 1 IP address (1 host up) scanned in 0.94 seconds
```

Nous voyons qu'il est en marche en regardant le champ **state : open**.
Lors de l'installation d'OpenLDAP un annuaire est créé par défaut avec un utilisateur qui est l'utilisateur admin, pour voir un aperçu du contenu de l'annuaire on tape la commande :
#slapcat

```
dn: dc=nodomain
objectClass: top
objectClass: dcObject
objectClass: organization
o: nodomain
dc: nodomain
structuralObjectClass: organization
entryUUID: dc6e54b6-a41b-1035-8a50-c7cbf06ccd5b
creatorsName: cn=admin,dc=nodomain
createTimestamp: 20160501190840Z
entryCSN: 20160501190840.995507Z#000000#000#000000
modifiersName: cn=admin,dc=nodomain
modifyTimestamp: 20160501190840Z
```

```
dn: cn=admin,dc=nodomain
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e1NTSEF9SmxLblZTaThKSmZLMG9LcHVYNSt2MGE0N01jUXdoUk8=
structuralObjectClass: organizationalRole
entryUUID: dc8c232e-a41b-1035-8a51-c7cbf06ccd5b
creatorsName: cn=admin,dc=nodomain
createTimestamp: 20160501190841Z
entryCSN: 20160501190841.190841Z#000000#000#000000
modifiersName: cn=admin,dc=nodomain
modifyTimestamp: 20160501190841Z
```

Nous ne pouvons pas expliquer le fonctionnement détaillé d'un annuaire LDAP car ce serait trop long, en revanche on peut aborder les points essentielles à connaitre pour arriver à authentifier un client VPN à partir d'un annuaire LDAP.

Tout d'abord, un annuaire LDAP est structurée sous forme d'arbre qui part d'une base, cette base est défini par le champ **dc** et est souvent dérivée d'un nom de domaine. Ici notre base s'appelle ***nodomain*** comme on peut le voir sur le résultat de la commande slapcat plus haut.

Chaque objet stocké dans la base LDAP a un **DN** ***(Distinguished Name)*** et ce DN l'identifie de manière unique. Le DN est formé de plusieurs composants, chaque composant étant un doublet attribut=valeur.

Par exemple sur le résultat de la commande slapcat plus haut, on a **dn : cn=admin, dc=nodomain** est un DN qui distingue de façon unique l'administrateur.

Nous pouvons changer le nom de notre base en éditant le fichier **/etc/ldap/ldap.conf** qui est le fichier de configuration de l'annuaire en local c'est à dire juste sur la machine serveur.

LDAP est en écoute mais il faut tester si on est en mesure de faire une requête sur l'annuaire, par exemple pour la recherche d'un utilisateur, pour cela on utilise la requête **ldapsearch**.

Nous essayons de lister tous les utilisateurs de l'annuaire par exemple :
 # ldapsearch -xLLL

Cette requête ne marchera pas, car pour faire des requêtes sur l'annuaire il faut tout d'abord éditer le fichier **/etc/ldap/ldap.conf**

```
# LDAP Defaults
```

```
#

# See ldap.conf(5) for details
# This file should be world readable but not world writable.

BASE    dc=nodomain
URI     ldap://localhost:389  ldap://ldap-master.example.com:666
```

- **Le champ dc** correspond au nom de la base LDAP ici nodomain.
- **Le champ URI** correspond au nom d'hôte du serveur ici on a mis localhost suivit du port d'écoute.

nous redémarrons le service slapd :
#service slapd restart

Après ça nous pourrons effectuer des requêtes sans problème sur l'annuaire.

```
root@server-VirtualBox:/etc/openvpn# ldapsearch -xLLL
dn: dc=nodomain
objectClass: top
objectClass: dcObject
objectClass: organization
o: nodomain
dc: nodomain

dn: cn=admin,dc=nodomain
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
```

L'annuaire est pour l'instant vide il n'y a que l'utilisateur admin, nous allons donc ajouter un autre utilisateur.

La première étape est de créer la structure basic de l'arbre, pour cela nous ajoutons ce qu'on appelle des unités d'organisation auxquelles appartiendront nos utilisateurs.
Pour cela nous créons un fichier **/tree.ldif** dans un répertoire quelconque (on est pas obligé de l'appeler tree.ldif on peut l'appeler unit.ldif...)
#nano tree.ldif
Avec le texte suivant :

```
dn: ou=personnes, dc=nodomain
ou: les gens
objectClass: organizationalUnit

dn: ou=groupes, dc=nodomain
```

```
ou: les groupes
objectClass: organizationalUnit
```

Nous ajoutons cette information à la base LDAP avec la commande **ldapadd** :
#ldapadd -cxWD cn=admin,dc=nodomain -f /tree.ldif

Nous allons ajouter maintenant un nouvel utilisateur qui appartient aux unités d'organisation qu'on vient de créer, avec le login client1 à définir dans le (champ **uid**) et le mot de passe client1 à définir dans le (champ **sn**).
Pour cela nous allons créer un fichier par exemple client1.ldif dans un répertoire quelconque avec le texte suivant :

```
dn: cn=client1,ou=groupes,dc=nodomain
cn: client1
gidNumber: 20000
objectClass: top
objectClass: posixGroup

dn: uid=client1,ou=personnes,dc=nodomain
uid: client1
uidNumber: 20000
gidNumber: 20000
cn: client1
sn: client1
objectClass: top
objectClass: person
objectClass: posixAccount
objectClass: shadowAccount
loginShell: /bin/bash
homeDirectory: /home/client1
userPassword: client1
```

Nous ajoutons cette information à la base LDAP avec la commande **ldapadd**.

```
root@server-VirtualBox:/etc/openvpn/auth# ldapadd -cxWD cn=admin,dc=
    nodomain -f client1.ldif
Enter LDAP Password:
adding new entry "cn=client1,ou=groupes,dc=nodomain"
adding new entry "uid=client1,ou=personnes,dc=nodomain"
```

Bien sûr l'ajout se fait avec le compte administrateur en entrant le mot de passe, on peut s'assurer que les entrées ont bien été ajoutées en saisissant la commande :
#ldapsearch -xLLL

```

dn: dc=nodomain
objectClass: top
```

```
objectClass: dcObject
objectClass: organization
o: nodomain
dc: nodomain

dn: cn=admin,dc=nodomain
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

dn: ou=personnes,dc=nodomain
ou: les gens
ou: personnes
objectClass: organizationalUnit

dn: ou=groupes,dc=nodomain
ou: les groupes
ou: groupes
objectClass: organizationalUnit

dn: cn=client1,ou=groupes,dc=nodomain
cn: client1
gidNumber: 20000
objectClass: top
objectClass: posixGroup

dn: uid=client1,ou=personnes,dc=nodomain
uid: client1
uidNumber: 20000
gidNumber: 20000
cn: client1
sn: client1
objectClass: top
objectClass: person
objectClass: posixAccount
objectClass: shadowAccount
loginShell: /bin/bash
homeDirectory: /home/client1
```

Pour tester si notre client s'authentifie en local nous exécutons la commande suivante :

```
root@server-VirtualBox:/etc/openvpn/auth# ldapwhoami -xD uid=client1,ou=
    personnes,dc=nodomain -w client1
dn:uid=client1,ou=personnes,dc=nodomain
```

Maintenant que nous avons un service d'annuaire LDAP fonctionnel, il ne reste plus qu'à

configurer l'authentification LDAP pour OpenVPN.

4.3.6.2 Mise en place de l'authentification LDAP avec OpenVPN

Pour éviter de mettre du code LDAP dans chaque application qui fait de l'authentification , Unix utilise en général le système PAM dans lequel l'application appelle un greffon PAM pour les différentes opérations. PAM à son tour charge, selon sa configuration, du code LDAP (ou bien utilisant d'autres techniques d'authentification).

Dans la terminologie PAM, configurer une application revient à configurer l'accès au service. Tout les services sont configurés dans le répertoire /etc/pam.d/ où chaque fichier détaille les politiques d'authentification liées à ce service.

Nous allons donc installer le support LDAP de PAM. Sur Ubuntu, c'est le paquetage **libpam-ldap**.
#apt-get install libpam-ldap

Il faut le configurer pour lui donner les informations nécessaires pour se connecter à l'annuaire. Cela se fait dans **/etc/ldap.conf** dont voici l'extrait :

```
# The distinguished name of the search base.
base dc=nodomain

# Another way to specify your LDAP server is to provide an
uri ldapi://localhost:389
# The LDAP version to use (defaults to 3
# if supported by client library)
ldap_version 3
```

Nous spécifions la même chose que dans le fichier **/etc/ldap/ldap.conf**, le champ nom de la base **dc**=nodomaine, **uri**=adresse du serveur LDAP, ici c'est sur la machine locale donc on met l'adresse localhost.

Nous créons ensuite le fichier appelé openvpn dans le répertoire /etc/pam.d/. C'est en fait ce fichier qui permet à OpenVPN d'utiliser PAM pour l'authentification LDAP. Tout les services qui utilisent PAM pour l'authentification ont un fichier à leurs noms dans ce dossier on trouvera par exemple : login, sudo, samba...
Le fichier openvpn est le suivant :
#nano /etc/pam.d/openvpn

```
account required pam_ldap.so debug
auth required pam_ldap.so debug
password required pam_ldap.so debug
```

Ces lignes veulent dire :

- **Ligne1 :** vérification du compte de l'utilisateur dans l'annuaire LDAP (expiration, plannings, etc).
- **Ligne2 :** l'authentification avec LDAP est nécessaire.
- **Ligne3 :** il permet de mettre à jour le jeton d'authentification (en général un mot de passe), soit parce qu'il a expiré, soit parce que l'utilisateur souhaite le modifier.

Le paramètre **debug** permet juste d'avoir des traces des opérations effectuées sur l'annuaire dans le fichier de log qui est dans le répertoire **/var/log/**.

La derniére étape, nous mettons à jour le fichier **/etc/openvpn/server.conf** en ajoutant deux lignes :

```
#authentification ldap
username-as-common-name
plugin /usr/lib/openvpn/openvpn-plugin-auth-pam.so openvpn
```

Explication : Lors du lancement de OpenVPN, l'option **"plugin/usr/lib/openvpn/openvpn-auth-pam.so openvpn"** indique à OpenVPN de chercher le fichier **"openvpn"** dans **/etc/pam.d** et d'exécuter ce qu'il y a dedans.

Maintenant il ne reste plus qu'à ajouter la directive **auth-user-pass** dans le fichier de configuration client qui devient donc ainsi :

```
client
dev tun
proto tcp-client
remote 192.168.8.2 443
resolv-retry infinite
cipher AES-256-CBC
; client-config-dir ccd
# Clés
ca ca.crt
cert  client1.crt
key  client1.key
tls-auth ta.key 1
key-direction 1
# Sécurité
auth-user-pass
nobind
persist-key
persist-tun
comp-lzo
```

```
verb 3
```

Nous redémarrons le service slapd.

#service slapd restart

Nous redémarrons le service VPN.

#service openvpn restart

Notre configuration est términé.

- **Test**

Nous essayons de nous connecter une nouvelle fois à notre serveur VPN à partir de notre client.

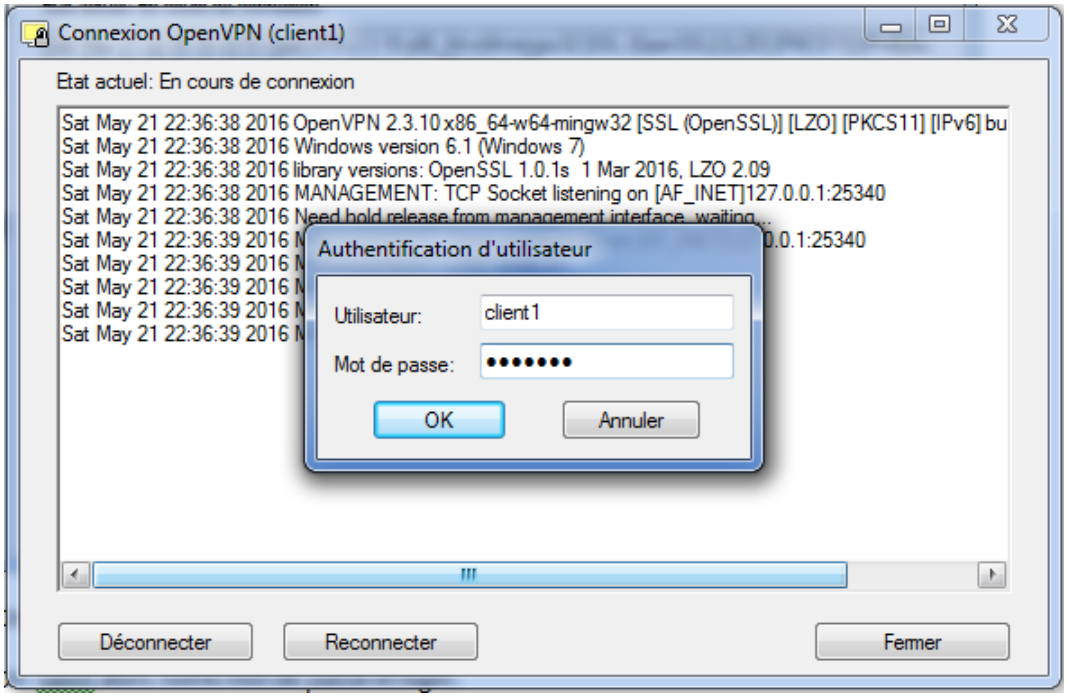

FIGURE 4.7 – Authentification avant la connection au serveur OpenVPN

Nous voyons alors apparaitre une fenêtre qui nous demande de nous authentifier. Nous saisissons notre mot de passe et login.

Pour vérifier que nous accèdons bien à la base LDAP pour s'authentifier, nous regardons dans le log de slapd **/var/log/slapd.log**

```
[16-06-2016 16:56:17] slapd debug  conn=1005 fd=19 ACCEPT from IP
   =127.0.0.1:51472 (IP=0.0.0.0:389)
[16-06-2016 16:56:17] slapd debug  conn=1005 op=0 BIND dn="" method=128
[16-06-2016 16:56:17] slapd debug  conn=1005 op=0 RESULT tag=97 err=0 text=
```

```
[16-06-2016 16:56:17] slapd debug  conn=1005 op=1 SRCH base="dc=nodomain"
   scope=2 deref=0 filter="(uid=client1)"
[16-06-2016 16:56:17] slapd debug  <= bdb_equality_candidates: (uid) not
   indexed
[16-06-2016 16:56:17] slapd debug  conn=1005 op=2 BIND dn="uid=client1,ou=
   personnes,dc=nodomain" method=128
[16-06-2016 16:56:17] slapd debug  conn=1005 op=2 BIND dn="uid=client1,ou=
   personnes,dc=nodomain" mech=SIMPLE ssf=0
[16-06-2016 16:56:17] slapd debug  conn=1005 op=2 RESULT tag=97 err=0 text=
[16-06-2016 16:56:17] slapd debug  connection_input: conn=1005 deferring
   operation: binding
[16-06-2016 16:56:17] slapd debug  conn=1005 op=3 BIND anonymous mech=
   implicit ssf=0
[16-06-2016 16:56:17] slapd debug  conn=1005 op=3 BIND dn="" method=128
[16-06-2016 16:56:17] slapd debug  conn=1005 op=3 RESULT tag=97 err=0 text=
[16-06-2016 16:56:17] slapd debug  conn=1005 op=4 UNBIND
[16-06-2016 16:56:17] slapd debug  conn=1005 op=1 SEARCH RESULT tag=101 err
   =0 nentries=1 text=
[16-06-2016 16:56:17] slapd debug  conn=1005 fd=19 closed
```

Nous voyons bien qu'une requête **bind** (requête de connexion) s'effectue avec le nom d'utilisateur **client1** qui confirme que nous accédons bien à l'annuaire.

4.3.7 Création automatique des fichiers client

La création de comptes clients (certificat et clé) et la mise en place de la configuration sur la machine cliente est une tâche contraignante pour l'administrateur, en effet il doit copier tout les certificats et clés nécessaires sur la machine cliente pour lui permettre de se connecter.

Nous avons essayé de faciliter ces opérations avec un script qui permet la création automatique des certificats et clés clients et leur rangement dans un dossier spécifique pour chaque client, avec toute la configuration nécessaire, (le fichier de config est créé automatiquement ainsi que toutes les clés et certificats).

Pour la mise en place de la configuration sur la machine cliente, ce script permettra d'envoyer la configuration client dans un zip automatiquement par mail au compte client. Dans ce zip il y'aura un fichier **readme.txt** qui expliquera la procédure à suivre pour mettre en place l'accès au VPN, une procédure très facile.

Nous avons donc le script **autoclient.sh** que nous avons créé dans le dossier **/etc/openvpn/**

```
#!/bin/bash
#Script pour crée automatiquement des certificats et des clefs pour les
   clients OpenVPN
```

```
#Initialisation des variables
echo "Nom du certificat: (ex: client4 ou bob)"
read cn

#Création des certificats
cd /etc/openvpn/easy-rsa/
source vars
./build-key $cn
cd /etc/openvpn/clientconf/
mkdir $cn
#Création du fichier de configuration ovpn pour les clients Windows et conf
    pour linux
echo "client
dev tun
proto tcp-client
remote 192.168.8.2 443
resolv-retry infinite
cipher AES-256-CBC
; client-config-dir ccd
# Cles
ca ca.crt
cert  $cn.crt
key  $cn.key
tls-auth ta.key 1
key-direction 1
# Securite
auth-user-pass
nobind
persist-key
persist-tun
comp-lzo
verb 3" > $cn/$cn.ovpn

echo "client
dev tun
proto tcp-client
remote 192.168.8.2 443
resolv-retry infinite
cipher AES-256-CBC
; client-config-dir ccd
# Cles
ca ca.crt
cert  $cn.crt
key  $cn.key
tls-auth ta.key 1
key-direction 1
```

```
# Securite
auth-user-pass
nobind
persist-key
persist-tun
comp-lzo
verb 3" > $cn/$cn.conf


#Création de l'archive avec tous les fichiers
cp /etc/openvpn/easy-rsa/keys/{$cn.crt,$cn.key,ca.crt,ta.key} $cn/
cp /etc/openvpn/readme $cn/
zip -r $cn.zip $cn

#Envoie des fichiers par mail
echo "Adresse E-mail a qui envoyer les certificats"
read mail
echo "Configuration d'OpenVPN" | mutt -x -s "Zip d'OpenVPN" $mail -a $cn.
    zip

exit 0
```

Bien sûr, pour envoyer les fichiers par mail il faut avoir un serveur SMTP fonctionnel, dans le cas de l'architecture réelle nous aurons qu'à utiliser le serveur SMTP de l'université pour envoyer ces mails. Mais comme nous sommes sur une machine virtuelle et que nous n'avons pas de serveur SMTP à notre disposition nous utiliserons le serveur SMTP de GMAIL comme relais.

- **Création du relais SMTP**

Installer Postfix

#sudo apt-get install postfix

Lorsque l'assistant se lance, un premier panneau nous explique chaque configuration possible. Donc faisons OK. Nous devons ensuite choisir différents paramètres, nous citons ici les plus importants pour avoir un service fonctionnel :

1. La première question correspond justement au type d'installation de Postfix que nous souhaitons effectuer, ici Système Satellite car nous voulons faire un relais SMTP.
2. La question suivante permet de choisir le nom suivant le symbole @ pour les expéditeurs du courriel. Nous mettons le nom de notre machine.
3. Ensuite, cette question est le point clé de l'assistant, elle permet de saisir le serveur SMTP que nous voulons utiliser comme relais. Donc, nous mettons l'adresse du serveur SMTP de Gmail, suivi du numéro de port qu'on veut utiliser.

4. Le courrier de root doit être envoyé en alias sur un autre utilisateur, nous saisissons donc ce nom d'utilisateur à la question suivante (au quotidien, lorsque l'utilisateur root du système enverra un mail, il sera transféré sous le nom de l'utilisateur choisi ici). Nous mettons l'adresse de notre compte utilisateur sur Gmail pour que les emails de root soient directement renvoyés vers notre boite mail où il seront expédiés par notre boite mail. Ici une adresse de type : XXXX@gmail.com
5. Ensuite, nous choisirons les noms d'hôtes qui seront acceptés lorsqu'un utilisateur local souhaitera envoyer un courrier (le nom de notre machine devrait suffire ici).
6. On répond "Oui" à la question "Forcer des mises à jour synchronisées de la file d'attente des courriels ?"
7. Il est fortement recommander de n'accepter les envois de courrier que sur la boucle locale, à savoir "127.0.0.0/8" pour la question suivante.
8. Enfin, tous les protocoles Internet méritent d'être activés, nous choisissons "tous" à la question suivante.

Le serveur Postfix va alors redémarrer, et l'envoi de mail sera désormais fonctionnel.

Une fois cela fait, nous allons configurer Postfix :

Nous editons le fichier de configuration ***/etc/postfix/main.cf*** et on ajoute ou modifie les lignes suivantes :

```
relayhost = [smtp.gmail.com]:587
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_sasl_security_options =
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
smtp_use_tls = yes
```

Puis, nous allons configurer les informations utilisées sur le SMTP de Gmail :

Nous éditons ou créons le fichier ***/etc/postfix/sasl_passwd*** et on ajoute nos informations personnelles :

```
[smtp.gmail.com]: 587    user.name@gmail.com:password
```

Nous restreignons les droits :

#sudo chmod 400 /etc/postfix/sasl_passwd

#sudo postmap /etc/postfix/sasl_passwd

#sudo chown postfix /etc/postfix/sasl_passwd*

Et nous finissons l'installation en relançant Postfix :

#/etc/init.d/postfix reload

Maintenant nous allons tester notre relai SMTP, pour cela nous installons le package mailutils.

#apt-get install mailutils

Ceci va nous permettre d'utiliser la commande mail pour tester l'envoi d'un mail.

#mail adressemail

```
Cc:
Subject: mail
mail depuis une console
```

On appuis sur Ctrl+d pour envoyer.

Pour les Vérifications d'erreur nous pouvons regarder dans le fichier de log **(/var/log/-mail.log)**.

Le programme utilisé pour envoyer des pièces jointes dans notre script est mutt. On l'installe avec la commande suivante :

#apt-get install mutt

Maintenant notre script est prêt à être utiliser.

- **Test**

Nous rentrons dans le dossier /etc/openvpn/ et nous exécutons notre script.

#./autoclient.sh

```
root@server-VirtualBox:/etc/openvpn# ./autoclient.sh
Nom du certificat: (ex: client4 ou bob)
client3
NOTE: If you run ./clean-all, I will be doing a rm -rf on /etc/openvpn/easy
   -rsa/keys
Generating a 2048 bit RSA private key
.....................................+++
.............+++
writing new private key to 'client3.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
. . . .

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
  adding: client3/ (stored 0%)
```

```
  adding: client3/client3.ovpn (deflated 30%)
  adding: client3/client3.key (deflated 23%)
  adding: client3/client3.conf (deflated 30%)
  adding: client3/ta.key (deflated 39%)
  adding: client3/client3.crt (deflated 47%)
  adding: client3/ca.crt (deflated 36%)
  adding: client3/readme (deflated 35%)
Adresse E-mail a qui envoyer les certificats
sss@live.fr
```

Nous voyons bien la création d'un zip client3 avec tous les fichiers nécessaires et le fichier **readme** qui explique la procédure à suivre pour l'installation des certificats et de l'application VPN. Et enfin le script qui nous demande de rentrer l'adresse de destination.

4.4 conclusion

Notre objectif dans ce chapitre était la mise en place d'un VPN d'accès avec une facilité d'administration et d'entretient pour l'administrateur réseau. Pour cela nous avons présenté une solution facile et à moindre coût qui est OpenVPN avec une authentification centralisée à travers un annuaire LDAP et mis en place quelques outils pour une configuration plus aisée.

CONLUSION GÉNÉRALE ET PERSPECTIVE

Notre projet de fin de cycle consistait en la mise en oeuvre d'un VPN d'accès pour l'université Abderrahmane Mira de Béjaïa, ayant comme objectif principal d'offrir un moyen sécurisé et sûre qui permettra au personnel (enseignants, administrateurs...) d'accéder au réseau interne de l'université par Internet, depuis l'extérieur du réseau (leur domicile, places publiques) afin de poursuivre leurs activités sans pour autant risquer de compromettre la sécurité des ressources internes du réseau.

Au travers de notre étude, les concepts présentés dans l'ensemble de l'ouvrage ont étaient méthodiquement appliqués. Ainsi la mise en oeuvre du VPN s'est basée sur la solution SSL OpenVPN qui a l'avantage d'offrir une facilité de configuration couplé à une sécurité optimal qui l'a distinguée des autres solutions VPN tout aussi efficaces mais plus complexes. OpenVPN offre une authentification en deux niveaux, au niveau de la machine avec des certificats SSL et au niveau des utilisateurs avec un login et mot de passe.

La configuration du serveur a été réalisée sur une machine Ubuntu 14.04 et afin de faciliter l'administration du VPN nous avons mis en place une gestion centralisée des logins et mots de passe client avec un annuaire LDAP, ainsi on aura pas besoin de créer une base des utilisateurs VPN pour l'authentification, nous réutiliserons directement les comptes déjà présents dans l'annuaire qui permettent déjà à ces utilisateurs de s'authentifier aux autres services du réseau de l'université.

La création des certificats client et leurs déploiement sur la machine cliente est souvent une tâche contraignante, pour pallier cette difficulté nous avons mis en place un script qui automatise cette opération, celui-ci utilise un relais SMTP pour l'envoi automatique des fichiers de configuration par mail au client concerné avec des explications aisées pour l'installation de ces fichier.

Au terme de notre travail nous avons un service VPN fonctionnel avec tous les outils

nécessaires permettant son administration et son debugging en cas de bug. Néanmoins des perspectives d'améliorations et d'extensions de notre configuration restent envisageables en tenant compte des insuffisances observées, nous pouvons ajouter des fonctionnalités supplémentaires telle qu'une interface graphique pour la surveillance du VPN (clients connectés, certificats valides et révoqués...), plus conviviale que sur console, ou bien une interface pour permettre au client de télécharger directement leurs certificats en ligne...

La réalisation de ce projet a était bénéfique à plus d'un titre. Le système a exigé des connaissances solides et avancées en administration et sécurité réseau ce qui nous a permis de mettre en pratique les stratégies étudiées dans notre cursus universitaire, dans des livres, sur internet ou même en stage.

Pour conclure, ce contexte nous a permis de réaliser un travail collectif dans des conditions similaires à celle d'un vrai poste dans l'administration réseau et faisant appel aux connaissances de chacun.

BIBLIOGRAPHIE

[1] Joëlle Musset ,"Sécurité informatique Ethical Hacking", Editions ENI, Octobre 2009.

[2] Aman Vladimir, "Concevoir la Sécurité Informatique en Entreprise", Publié sous licence Creative Commons, 2014.

[3] Lourent Bloch, Christophe Wolfhgel, "Sécurité informatique", 2éme edition EYPRLLES, 1998.

[4] Nicolas Baudoin, Marion Karle, " NT Réseaux : IDS et IPS ", Rapport Ingéniorat, 2003/2004.

[5]] Abid Y, Belhocine M, " Proposition d'une architecture réseaux sécurisé pour l'université de Bejaia", Mémoire de fin d'étude Master 2 , 2015.

[6] G Blum, F Lasowy, C Guerin, C Pfeiffer, " Protocole AAA Principes et implantation", rapport de stage, 2002.

[7] Mohamed Douhaji, " Exemple d'attaque par DOS + la sécurité ", cour, disponible sur : http ://eventus-networks.blogspot.com/2013/10/exemple-dattaque-par-dos.html, avr 2016.

[8] Guillaume Desgeorge, " La sécurité des réseaux ", Cour, 2000.

[9] Université de lille 1, "Qu'est-ce qu'un FireWall ?", Cour, disponible sur : https ://wapiti.telecom-lille.fr/commun/ens/peda/options/st/rio/pub/exposes/exposesrio 2002/vanoudendycke-delaby/ firewall.htm, avr 2016.

[10] Jean-Paul ARCHIER, "Les VPN fonctionnement, mise en oeuvre et maintenance des Réseaux privés virtuels", Editions ENI , Juin 2010.

[11] Marc Boget, "Les protocoles réseaux", cour, disponible sur : http ://marc.boget.free.fr/stage-html2.html, avr 2016.

[12] "Les VPN", cour, disponible sur : http ://perso.modulonet.fr/placurie /Ressources/BTS2-AMSI/Chap-8-Les-VPN.pdf, avr 2016.

[13] Xavier Lasserre, Thomas Klein, "Réseaux Privés Virtuels", cour, disponible sur : http ://www.frameip.com/vpn/3.6.1-Fonctionnement,avr 2016.

[14] Agence nationale de la sécurité des systèmes d'information, "Recommandations de sécurité relatives à IPsec", Note technique, disponible sur : www.ssi.gouv.fr/uploads /2012/09/NT_IPsec.pdf, avr 2016.

[15] Site officiel de OpenVPN, disponible sur : https ://openvpn.net/howto.html, avr 2016.

[16] Jérôme Fenal, " Introduction à LDAP et déploiment de OpenLDAP ", article, disponible sur : http ://articles.mongueurs.net/magazines/linuxmag65.html, avr 2016.

[17] "Relais SMTP", guide, disponible sur : http ://www.serversmtp.com/, avr 2016.

[18] Site officiel d'ubuntu, disponible sur :https ://doc.ubuntu-fr.org/virtualbox, avr 2016.

[19] Site officiel d'openVPN ,disponible sur : https ://openvpn.net/easyrsa.htm, avr 2016.

[20] Maxime Auvy, " Monter son VPN perso avec OpenVPN ", article, disponible sur : open-freax.fr/monter-vpn-openvpn/, avr 2016.

[21] Site de l'institut d'électronique et d'informatique Gaspard-Monge (IGM),"Pluggable Authentification Module", cour, disponible sur : http ://www-igm.univ-//mlv.fr/ dr/X-POSE2003/augereau/2.html, avr 2016.

Résumé

Depuis quelque temps le centre de calcule de l'université de Béjaïa fait face à de plus en plus de demandes du personnel relatant de la permission d'accéder au réseau locale de l'université par Internet depuis leur domicile ou places publiques. Compte tenus des importants risques de sécurité encouru (dans les applications courantes, le mot de passe circule en clair sur le réseau), ces demandes sont difficiles à accepter.

Le présent travail consiste en la mise en place d'un VPN d'accès au LAN de l'université pour permettre au personnel d'accéder au données internes en toute sécurité avec une connexion chiffrée sans risquer de compromettre la sécurité du réseau.

Pour ce fait nous avons choisis la solution SSL OpenVPN qui offre une facilité de configuration et une sécurité optimale renforcées avec une double authentification par mot de passe login et certificats SSL. Afin de faciliter l'administration du VPN nous avons mis en place une authentification centralisée au travers d'un annuaire LDAP, et mit en oeuvre des outils pour l'automatisation de certaines taches contraignantes. La configuration s'est faite sur un serveur Ubuntu 14.04 avec des clients Windows.

Mot clés : VPN d'accès, LDAP, relai SMTP, Ubuntu 14.04, PAM, OpenVPN