ÉCOLE PRATIQUE DES HAUTES ETUDES COMMERCIALES

Avenue du Ciseau, 15
1348 Louvain-la-Neuve

Comparatif des solutions d'accès à distance aux ressources informatiques

---

Travail de fin d'études présenté en vue de l'obtention du diplôme de bachelier en Informatique et Systèmes : finalité Technologie de l'informatique

Auteur :
Christian JUCKLER 3TL1
Rapporteur :
Virginie Van den Schriek

Année Académique 2014-2015

Résumé

Test

Table des matières

Liste des tableaux

Table des figures

Introduction

Première partie

La problématique des accès distants en entreprise

Chapitre 1

Les besoins habituels

Chapitre 2

Les besoins spécifiques

Chapitre 3

La situation actuelle

FIGURE I – Test figure

Chapitre 4

Méthodologie et objectifs

Deuxième partie

Technologies d'accès distant

Chapitre 5

Description des technologies

Dans ce chapitre, je présente les technologies d'accès distants. Dans un premier temps, je commence par expliquer les notions des VPN et de tunnel VPN. De plus, j'y expose les protocoles de tunneling et de chiffrement. Par la suite, je détaille les protocoles qui peuvent former des tunnels VPN sécurisés. Finalement, je passe en revue les solutions propriétaires et celles des fournisseurs d'accès pour les accès distants.

5.1 Virtual Private Network - VPN

Le terme VPN est un acronyme pour « Virtual Private Network ». Un VPN est par définition un réseau virtuel qui transfère des données privées en créant un tunnel à travers un réseau public.

Historiquement, les VPN étaient construits sur des lignes louées, mais le coût de ces infrastructures était trop important. Maintenant, les VPN sont construits sur l'Internet. L'avantage principal est son faible coût. Mais en utilisant l'Internet, les données sont accessibles à tout le monde. Il a donc été nécessaire de fournir des protocoles permettant d'assurer la confidentialité et l'intégrité des données à travers le réseau public.

Un tunnel VPN est monté entre deux passerelles VPN, ces passerelles sont d'un point de vue logique connectées directement l'une à l'autre comme illustré sur la figure. Le tunnel permet d'envoyer des données du réseau privé à travers le réseau public. Le tunnel encapsule les données dans un protocole compris par les deux passerelles. La passerelle émettrice encapsule les données et la passerelle destinataire récupère les données. En plus de l'encapsulation des données, les tunnels VPN peuvent réaliser du chiffrement. Le chiffrement permet de rendre les données inutilisables en cas de vol.

5.1.1 Les types de VPN

Il existe deux grands types de VPN : les VPN site-à-site et les VPN client-à-site.

Les VPN site-à-site

Ils sont utilisés pour connecter des sites entre eux. Le tunnel est monté entre deux passerelles VPN dont les configurations sont connues. Le réseau Internet opère comme une liaison WAN entre les sites. Les employées peuvent échanger des informations entre les différents sites comme s'ils sont connectés sur le site distant. Dans ce cas-ci, nous utilisons des tunnels VPN IPsec. Je discute d'IPsec plus loin dans ce chapitre.

Les VPN client-à-site

Ils sont généralement utilisés par des travailleurs pour accéder aux ressources de l'entreprise depuis des emplacements non fiables. L'utilisateur se connecte via son ordinateur ou son smartphone à la passerelle VPN de son entreprise. Dans ce cas, la configuration n'est pas connue, car selon la localisation

de l'utilisateur, les paramètres de connexion changent. Il est souvent nécessaire d'installer sur l'appareil mobile un client VPN. Nous utilisons des tunnels VPN SSL pour la simplicité de configuration. Je discute de SSL plus loin dans ce chapitre.

5.1.2 Les protocoles de tunneling

Ces protocoles permettent de transférer les paquets d'un protocole à l'intérieur d'un autre protocole. On parle d'encapsulation. Il est nécessaire que les deux extrémités du tunnel comprennent le protocole encapsulé. Il existe plusieurs protocoles, je ne discuterai que de ceux utilisés actuellement.

Point-to-Point Tunneling Protocol (PPTP)

PPTP est un protocole de tunneling. Il permet de router n'importe qu'elle protocole à travers le réseau IP. Il fonctionne en quatre phases dont une est optionnelle : Link Establishment Phase, Authentication Phase, Callback Control Phase, Network Control Phase.

1. La phase une sert à établir, maintenir et terminer la connexion physique entre les deux hôtes. C'est aussi à ce moment que les protocoles d'authentification sont choisis.
2. La phase une sert à établir, maintenir et terminer la connexion physique entre les deux hôtes. C'est aussi à ce moment que les protocoles d'authentification sont choisis.
3. La phase trois est optionnel et elle permet une sécurité accrue. Elle déconnecte le client et le serveur. Ensuite, le serveur rappelle le client.
4. La dernière phase sert à négocier et implémenter les protocoles de compression et de chiffrement.

Layer 2 Tunneling Protocol (L2TP)

Ce protocole a été créé en utilisant les avantages des protocoles PPTP et L2F. Mais il ne permet toujours pas la confidentialité du trafic. Il est possible de faire de l'authentification et du chiffrement avec les paquets PPP, mais la connexion reste vulnérable au niveau de la couche transport. Il est donc intéressant d'associer L2TP avec un autre protocole de sécurité comme IPSec.

5.2 Internet Protocol Security - IPSec

IPsec est un ensemble de protocole visant à sécuriser les données au niveau de la couche réseau. Il est composé de trois protocoles : AH (*Authentication Header*), ESP (*Encapsulating Security Payload*) et IKE (*Internet Key Exchange*). IKE est utilisé lors de la négociation des paramètres du tunnel VPN. Les deux autres protocoles fournissent la sécurité des données en les encapsulant au sein du tunnel VPN.

5.2.1 Les protocoles AH et ESP

Le protocole AH fournit une authentification sur la source du paquet, l'intégrité des données et une protection contre les attaques par rejeu. Le protocole ESP fournit les mêmes sécurités que AH, sauf qu'il n'authentifie l'en-tête IP des paquets. Il fournit en plus la confidentialité des données en chiffrant une partie du paquet. AH et ESP peuvent travailler en mode tunnel ou en mode transport.

Le mode transport est utilisé lorsque les terminaisons du tunnel VPN sont les destinataires finaux de la communication. Ce mode offre une sécurité de bout en bout. Au niveau des paquets, les en-têtes AH et ESP sont placés après l'en-tête IP, il n'y a donc qu'un seul en-tête IP.

Le mode tunnel est plus souple, mais il consomme plus de bande passante. Les destinataires finaux sont connectés via des passerelles VPN. Il n'y a pas de différence entre les VPN site-à-site et les VPN client-à-site. Les passerelles VPN ont pour objectif d'encapsuler les paquets pour les faire passer dans le tunnel VPN. Nous trouvons donc dans le paquet encapsulé l'en-tête IP du paquet d'origine (voir Fig.II p.10). Comme le paquet est passé à travers la passerelle, il possède un deuxième en-tête IP qui sert au routage entre les deux passerelles.

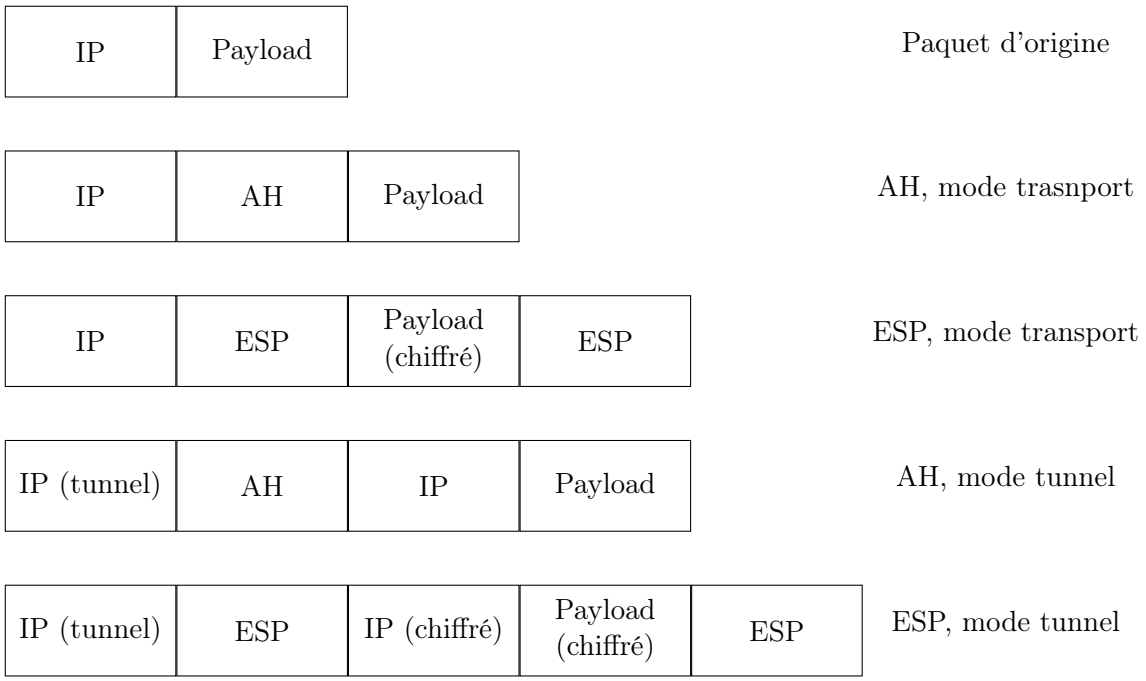

FIGURE II – Schéma des encapsulations IPSec

5.2.2 Security association

Quand nous parlons de monter un tunnel, en réalité, nous synchronisons un état partagé entre les terminaisons du tunnel. Cet état partagé se nomme une SA (*security association*) en IPSec. Une SA contient l'algorithme de chiffrement utilisé et les clés utilisées, l'algorithme d'authentification, un numéro d'identifiant, le *security parameter index* (SPI), ... Plus d'autres paramètres qui servent à maintenir les tunnels VPN. Les SA peuvent être créées manuellement ou gérées par l'IKE. Chaque terminaison possède deux SA, une pour le trafic entrant et une pour le trafic sortant. De plus, chaque pair est lié à un protocole. Les SA se caractérisent par un triplet formé du SPI, de l'adresse de destination et du protocole. Les SA sont stockées dans une SAD (*security association database*). Cette SAD est utilisée pour déterminer quel protocole est utilisé pour les paquets sortants et pour fournir les paramètres pour déchiffrer et/ou authentifier les paquets entrants. Il est possible de combiner les SA pour créer des tunnels VPN complexe.

Les SA sont des éléments simples, c'est-à-dire qu'elles traitent tous les paquets de la même manière. Pour un réglage plus fin, IPSec utilise des policies. Ces policies se basent sur les champs suivants des en-têtes du paquet.

— L'adresse de destination
— L'adresse source
— Le protocole de la couche transport
— Le port source
— Le port de destination

Elles servent à déterminer quels paquets à émettre sur quel tunnel, à dropper les paquets ne correspondant à aucune des règles décrites dans les policies. De la même manière que les SA, les policies sont stockées dans une SPD (*security policy database*). Le fonctionnement est similaire, pour chaque paquet entrant ou sortant, le système consulte la SPD pour déterminer les règles à appliquer au paquet. Si une règle est trouvé, le système cherche après la SA correspondante.

5.2.3 Le protocole IKE

IKE a un seul objectif : procéder à des échanges de clé Diffie-Hellman pour sécuriser un tunnel VPN. Il négocie le chiffrement, l'authentification nécessaire au tunnel, qui satisfont les policies.

IKE dérive du *Internet Security Association and Key Management* Protocol (ISAKMP). ISAKMP est un framework qui fournit des outils pour la sécurisation des échanges et l'échange de clé. De plus, IKE utilise différents mode du protocole OAKLEY. Il établit une SA en deux phases et il possède cinq modes d'échange, dont trois découlent ISAKMP. Les deux derniers modes ne sont utilisés que lors de la phase deux.

La phase 1 d'IKE

La phase 1 crée un canal sécurisé entre les terminaux du tunnel pour déterminer les SA. Le canal sécurisé est créé après l'authentification des terminaux. Les SA de la phase 1 sont bidirectionnelles, c'est-à-dire qu'une SA sécurise le trafic entrant et sortant. Pour l'échange des SA de la phase 1, IKE possède deux modes d'échanges :

— Main mode
— Agressive mode

Pour l'authentification d'un terminal, il existe quatre méthodes :

— a shared secret
— a digital signature
— public key encryption
— revised public key encryption

Le mode *"main"* d'IKE travaille en trois étapes (voir Fig.III p.11). Premièrement, l'initiateur envoie un

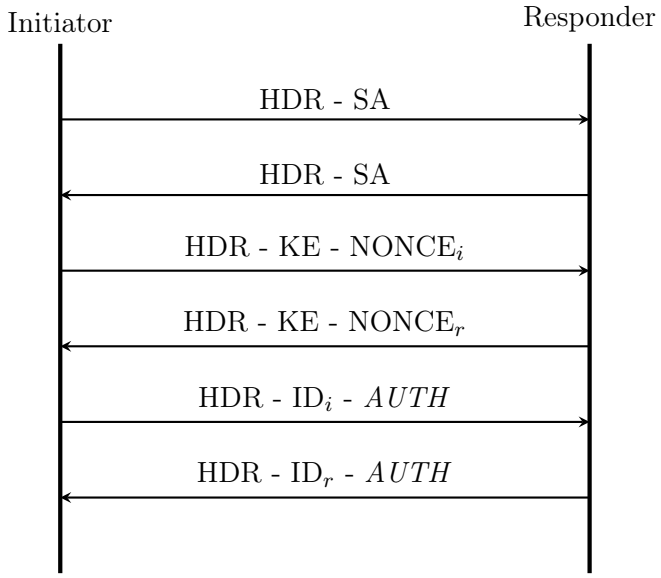

FIGURE III – IPsec : mode "main"

message contenant une liste des méthodes de sécurisation qu'il utilise. Le receveur choisit dans la liste reçue la méthode correspondant à ses policies et envoie sa décision à l'initiateur. Ensuite, ce dernier envoie sa clé privée pour créer le secret partagé de l'algorithme de Diffie-Hellman. Le receveur fait de même. Ils sont donc capables tous les deux de créer les clés. Les clés dépendent des méthodes d'authentification choisies. Finalement, l'initiateur envoie son identité et des informations sur l'authentification. Ces messages sont chiffrés et masquent donc l'identité des terminaux. L'échange se fait en six messages.

À la fin de ce mode, les terminaux sont d'accord sur les algorithmes de chiffrement et de confidentialité des données. Ils possèdent également les clés pour les algorithmes sélectionnés.

Le mode *"agressive"* fait le même travail de façon plus rapide, il n'utilise que trois messages (voir Fig.IV p.12). Lors du premier envoi, l'initiateur émet la liste des méthodes de sécurisation, son identité et

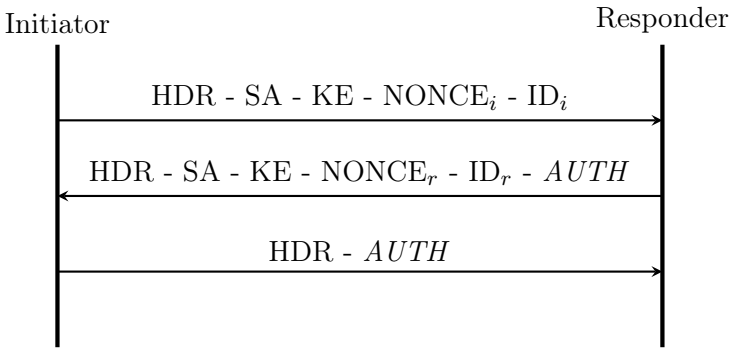

FIGURE IV – IPsec : mode "agressive"

sa clé. Le receveur répond par son choix de sécurisation, sa clé, son identité et ses identifiants. Finalement l'initiateur s'authentifie auprès du receveur. Ce dernier message peut être chiffré.

La phase 2 d'IKE

Une fois la SA établie, les terminaux peuvent l'utiliser pour négocier les SA de phase 2. La phase 2 est un échange en Quick mode. L'échange se fait en trois messages. Lors de l'échange, il est possible de négocier plusieurs SA en même temps.

5.3 Secure sockets layer - SSL

Netscape a lancé SSL 1 en 1994, dans le but de sécuriser des transactions réalisées avec leur navigateur. Dans la même année, SSL 2 était déjà en route. Mais le protocole montrait déjà des problèmes de sécurité. Fin 1995, SSL 3 était lancé. Il s'agissait d'une version complètement récrite de SSL, qui introduisait de nouvelles fonctionnalités issues de PCT [1]. Bien que les machines actuelles intègrent SSL 3, elles essaient d'abord de négocier une connexion en SSL 2.

Dans un effort de standardisation de SSL, l'IETF a lancé le protocole TLS [2]. Il se base principalement sur SSL 3 bien qu'il ne soit pas compatible avec ce dernier.

SSL utilise des suites de chiffrement. Ces suites se composent de trois fonctions de chiffrement : la méthode d'échange de clé, l'algorithme de chiffrement et une méthode de hachage. Il existe un large ensemble de suite, les client ont donc un mécanisme pour signaler les suites qu'ils gèrent et qu'ils utilisent.

OpenSSL est l'implémentation la plus courante de SSL. Cette implémentation possède un interface en ligne de commande, qui permet de générer des clés RSA, signer des certificats, calculer des valeurs de hash, ...

5.3.1 Le protocole SSL

SSL est un protocole de la couche transport, il utilise donc les protocoles de cette couche pour le transfert des données. Pour éviter des problèmes lors de la transmission des données, SSL utilise le protocole TCP.

De manière analogue à TCP, une session SSL se divise en trois phase (voir Fig.V p.13) :

1. Établissement de la connexion
2. Transfert des données
3. Clôture de la connexion.

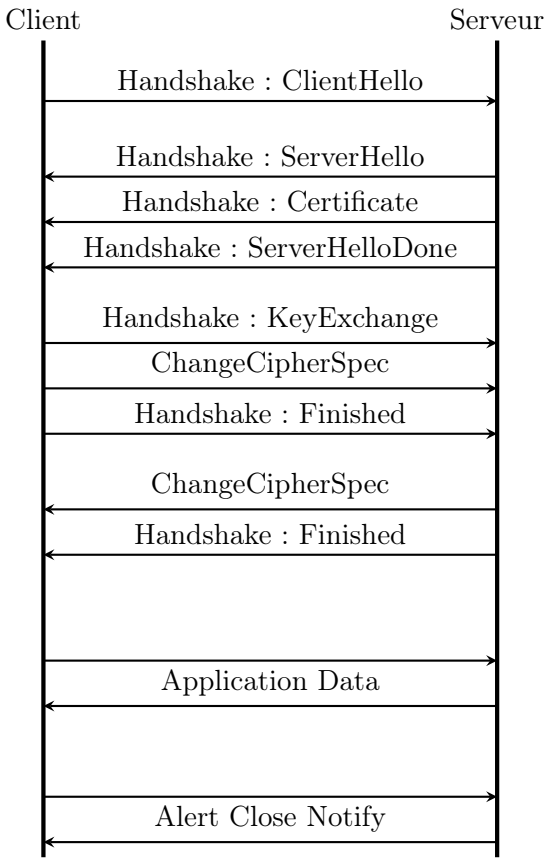

FIGURE V – Session SSL

La session commence par le *triple handshake*. Le client envoie un message ClientHello, qui indique la version de SSL supportée, la liste des suites de chiffrement et les algorithmes de compression. La version de SSL est signalée par deux champs dans l'en-tête : version mineur et version majeure. SSL3 a une version majeure de 3 et une version mineure de 0, et TLS a un version majeure de 3 et une version mineure de 1.

Le serveur répond par trois messages.

1. Le message *ServerHello* indique au client la suite de chiffrement et l'algorithme de compression à utiliser.
2. Le certificat du serveur permet au client de vérifier l'identité du serveur et contient la clé publique du serveur. Cette clé va servir à générer les différents clés pour la session.
3. Le message *ServerHelloDone* précise la fin de la séquence *Hello*.

Suite à ces trois messages, le client donne au serveur des inputs pour la génération des clés (*ClientKeyExchange*), signal au serveur qu'il utilise les nouvelles clés pour le chiffrement et l'authentification (*ChangeCipherSpec*) et qu'il a fini le handshake (*Finished*). Le serveur répond avec son message *ChangeCipherSpec* et son *Finished*.

Le client et le serveur sont capables de s'échanger des données de façon sécurisé.

Finalement, la session est clôturée.

5.4 Secure Shell - SSH

SSH a pour objectif de créer une connexion sécurisée. De la même manière que SSL, SSH est un protocole de la couche transport et utilise TCP. Par contre les applications ne doivent pas forcément

1. Microsoft's *Private Communications Technology*
2. Transport Layer Security

intégrer SSH pour être utilisées via SSH.

SSH est principalement utilisé pour remplacer `telnet`, mais il est également possible de faire du VPN. En effet, SSH fournit de l'authentification et du chiffrement pour les communications entres les machines.

Les tunnels SSH sont peu utilisés, car ils manquent de performance. Mais ils sont simple à mettre en place.

5.5 Secure Socket Tunneling Protocol - SSTP

SSTP est utilisé pour transporter du trafic PPP/L2TP via du SSL3. Son avantage réside dans le fait qu'il peut passer à travers les NAT, les proxys et les firewalls.

5.6 HTTP over TLS/SSL - HTTPS

5.7 Propriétaires

5.8 Fournisseurs d'accès

Chapitre 6

Comparaison théorique

Dans ce chapitre, je réalise un comparatif théorique des technologies décrites dans le chapitre précédent. Je précise, dans un premier temps, les critères de comparaison. Puis, je présente le tableau. Finalement, je réalise une analyse de ce tableau.

6.1 Les critères de comparaison

Tous les protocoles décrits dans le chapitre 5 visent un même but, mais en ayant des implémentations différentes. Pour pouvoir les différencier, il est utile de réaliser un comparatif sur des critères pertinents. Les critères que j'ai sélectionné sont :

— La facilité de configuration
— Le type de VPN
— La couche de protection
— L'intégrité des données
— L'authentification
— Le chiffrement

D'un point de vue purement technique, la facilité de configuration précise le degré de complexité pour mettre en place un tunnel VPN en utilisant une technologie.

6.2 Tableau de comparaison

Protocoles	Facilité de configuration
IPSec (AH, tunnel)	++
IPSec (ESP, tunnel)	++
IPSec (AH, transport)	+++
IPSec (ESP, transport)	+++
SSL/TLS	++++
SSH	++++
SSTP	++++
HTTPS	+++++

Table 6.1 – Tableau de comparaison théorique des technologies d'accès à distance

Chapitre 7

Architecture

Troisième partie

Comparaison des solutions commerciales

Chapitre 8

Choix des solutions

Chapitre 9

Installation des solutions

Chapitre 10

Scénario de test

Chapitre 11

Critères de comparaison

Test	de	tableau

Table 11.1 – Test de tableau

Conclusion