PPE 3.3 SECURISATION DES COMMUNICATIONS ENTRE SITES

I-) Contexte

a) Presentation de stadium company

Stadium Company est une société qui s'occupe de la gestion des stades , qui souhaite moderniser l'infrastructure réseau de son stade , la rendre scalable au fur et à mesure que ses ambitions augmentent . Elle emploie 170 personnes à plein temps et 80 à temps partiels . Elle projette ajouter des fonctions haute technologie afin de pouvoir prendre en charge l'organisation des concerts . La direction de Stadium Company n'étant pas compétente en matière d'Infrastructure réseau décide de faire appel à une entreprise de consultants réseaux qui s'en occupera de la conception à la mise en oeuvre en passant par la gestion du projet . Il sera mis en oeuvre en trois phases à savoir :

- La planification du projet et la préparation de la conception réseau haut niveau
- Le développement de la conception reseau détaillée
- La mise en oeuvre de la conception

b) Presentation NetworkingCompany

Le prestataire désigné pour l'exécution de ce projet est la société Networking Company, une firme locale spécialisée dans la conception d'infrastructure réseau et le conseil . C'est une société Cisco Premier Partner qui emploie 20 ingénieurs réseaux certifiés et expérimentée dans ce secteur.

Dans le but de concevoir le nouveau plan de l'infrastructure de haut niveau , Networking Company a décrit un profil de l'organisation et des installations sur la base des études menées par ses experts notamment en interrogeant le personnel.

Mission 3

Solution permettant l'administration à distance sécurisées et la sécurisation des interconnexions

· La sécurité du système d'information devra être renforcée entre les différents sites

· Sécurisation des interconnexions entre le site du stade et les sites distants Billetterie et Magasin.

· La solution retenue devra être administrable à distance via un accès sécurisé par SSH

Solutions

Test et comparaison des solutions :

3.1 Acces distant :

Consulter son courriel, transférer des fichiers ou simplement se connecter sur les ordinateurs , sur les serveurs étant à l'extérieur impose que les échanges d'informations se fassent en utilisant des réseaux externes. En particulier, en utilisant les protocoles standards (**telnet** , **rlogin , rcp , rsh**) non sécurisé , la phase d'authentification est transmise en clair. Une personne mal intentionnée peut ainsi capter le login et le mot de passe. Elle pourra ensuite l'utiliser à sa convenance. Il est donc recommandé de mettre en place des procédures permettant de limiter ce risque.

- **SSH protocole**

SSH™ (ou *S*ecure *SH*ell) est un protocole qui facilite les connexions sécurisées entre deux systèmes à l'aide d'une architecture client/serveur et permet aux utilisateurs de se connecter à distance à des systèmes hôte de serveurs.Toutefois, contrairement à d'autres protocoles de communication à distance, tels que FTP ou Telnet, SSH crypte la session de connexion et empêche ainsi tout agresseur de recueillir des mots de passe non-cryptés.
SSH est conçu pour remplacer les applications de terminal plus anciennes et moins sécurisées qui sont utilisées pour se connecter à des hôtes distants, comme **telnet** ou **rsh**. Un programme similaire appelé **scp** remplace des programmes moins récents conçus pour copier des fichiers entre des hôtes, tels que **rcp**. Étant donné que ces applications plus anciennes ne cryptent pas les mots de passe entre le client et le serveur, il est recommandé d'éviter autant que possible de les utiliser. En effet, l'utilisation de méthodes sécurisées pour se connecter à des systèmes distants, réduit les risques aussi bien pour le système client que pour l'hôte distant.

- **Fonctionnement**

SSH offre les précautions suivantes au niveau de la sécurité :

- Après avoir effectué une connexion initiale, le client peut s'assurer que sa connexion est établie avec le même serveur que lors de sa session précédente.
- Le client transmet ses données d'authentification au serveur au moyen d'un cryptage solide 128 bits.
- Toutes les données envoyées et reçues lors d'une session sont transférées au moyen d'un cryptage 128 bits, rendant ainsi le décryptage et la lecture de toute transmission interceptée extrêmement difficile.
- Le client peut retransmettre des applications X11 depuis le serveur. Cette technique appelée *retransmission X11*, fournit un moyen d'utiliser en toute sécurité des applications graphiques sur un réseau.

Étant donné que le protocole SSH crypte tout ce qu'il envoie et reçoit, il peut être utilisé pour sécuriser des protocoles autrement vulnérables. Grâce à la technique de *retransmission de port*, un serveur SSH peut être employé pour sécuriser des protocoles non-sécurisés tels que POP, augmentant ainsi la sécurité globale du système et de ses données.

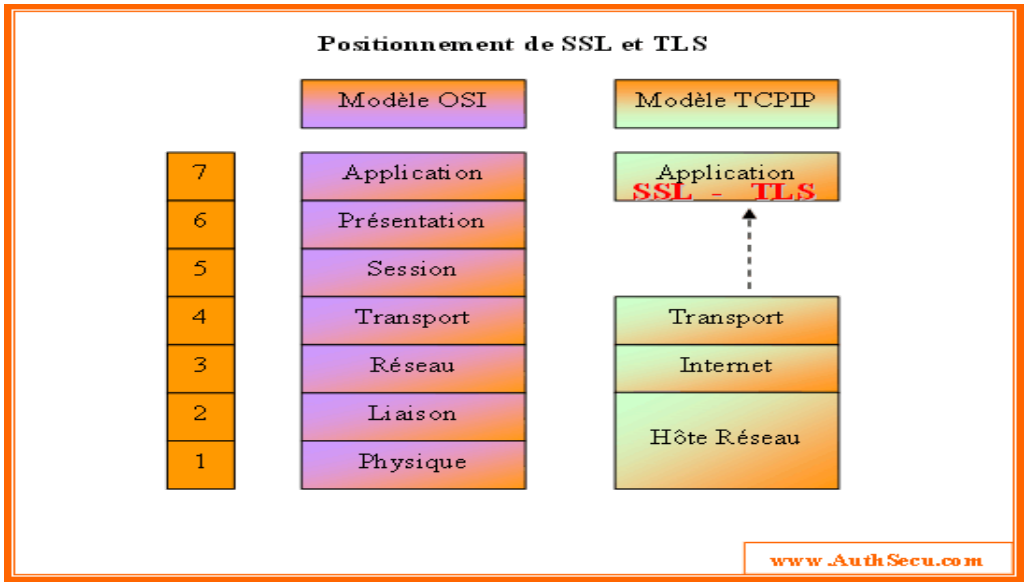
Les utilisateurs d'ordinateurs malintentionnés disposent d'une variété d'outils pour interrompre, intercepter et réacheminer le trafic réseau afin de s'octroyer l'accès à un système. D'une manière générale, ces menaces peuvent être répertoriées de la manière suivante :

- *Interception d'une communication entre deux systèmes* — Dans ce scénario, le pirate peut se trouver quelque part sur le réseau entre les entités qui communiquent, pouvant ainsi copier toute information qui est transmise entre elles. Le pirate peut intercepter et garder les informations ou peut les modifier avant de les envoyer au destinataire prévu. Cette attaque peut être orchestrée en utilisant un programme renifleur — un utilitaire réseau courant.
- *Usurpation de l'identité d'un hôte* — Grâce à cette technique, le système d'un agresseur est configuré de telle manière qu'il apparaît comme étant le destinataire souhaité d'une transmission. Si cette stratégie fonctionne, le système de l'utilisateur ne détecte pas qu'il communique en fait avec le mauvais hôte.
  Ce type d'attaque peut être organisé grâce à l'utilisation de techniques appelées empoisonnements DNS ou usurpation d'adresse IP.

Ces deux techniques permettent d'intercepter des informations potentiellement confidentielles et si cette interception est effectuée pour des raisons hostiles, le résultat peut être catastrophique. L'utilisation du protocole SSH pour effectuer une connexion au shell à distance ou pour copier des fichiers permet de réduire considérablement ces menaces au niveau de la sécurité. En effet, le client et serveur SSH utilisent des signatures numériques pour vérifier leur identité respectives. En outre, toute communication entre le système client et le système serveur est cryptée. Toute tentative d'usurpation d'identité à une extrémité ou à une autre de la communication est difficilement possible puisque chaque paquet est crypté à l'aide d'une clé connue seulement par le système local et le système distant.

- **Protocole TLS et SSL**

SSL signifie Secure Sockets Layer et son équivalent actuel TLS signifie Transport Layer Security. Ils sont tous les deux des protocoles situés entre le niveau Transport et Application. SSL et TLS se comportent en effet comme une couche intermédiaire supplémentaire, car ils sont indépendants du protocole utilisé au niveau application. Cela signifie donc qu'il peut aussi bien être employé pour sécuriser une transaction web, l'envoi ou la réception d'email, etc. SSL et TLS sont donc transparents pour l'utilisateur et ne nécessitent pas l'emploi de protocoles de niveau Application spécifiques.



- **Fonctionnalités des protocoles TLS et SSL**

SSL et TLS proposent les fonctionnalités suivantes :

- Authentification – Le client doit pouvoir s'assurer de l'identité du serveur. Depuis SSL 3.0, le serveur peut aussi demander au client de s'authentifier. Cette fonctionnalité est assurée par l'emploi de certificats.
- Confidentialité – Le client et le serveur doivent avoir l'assurance que leur conversation ne pourra pas être écoutée par un tiers. Cette fonctionnalité est assurée par un algorithme de chiffrement.
- Identification et intégrité – Le client et le serveur doivent pouvoir s'assurer que les messages transmis ne sont ni tronqués ni modifiés (intégrité), qu'ils proviennent bien de l'expéditeur attendu. Ces fonctionnalités sont assurées par la signature des données

SSL et TLS reposent donc sur la combinaison de plusieurs concepts cryptographiques, exploitant à la fois le chiffrement asymétrique et le chiffrement symétrique.
SSL et TLS se veulent en outre évolutif, puisque le protocole est indépendant des algorithmes de chiffrement et d'authentification mis en oeuvre dans une transaction. Cela lui permet de s'adapter aux besoins des utilisateurs et aux législations en vigueur. Cela assure de plus une meilleure sécurité, puisque le protocole n'est pas soumis aux évolutions théoriques de la cryptographie (Si un chiffrement devient obsolète, le protocole reste exploitable en choisissant un chiffrement réputé sûr).

- **Fonctionnement de TLS et SSL**

Les protocoles SSL et TLS se décomposent en deux couches principales (quatre en réalité) :

- SSL et TLS Handshake Protocol choisit la version de SSL et TLS qui sera utilisée, réalise l'authentification par l'échange de certificats et permet la négociation entre le client et le serveur d'un niveau de sécurité au travers du choix des algorithmes de cryptage. C'est le protocole de configuration de la transaction.

- SSL et TLS Record Protocol encapsule et fragmente les données. C'est le protocole de transmission des données.

Dans une première phase, le client et le serveur vont effectuer la négociation afin de configurer la transaction et d'échanger les clés de chiffrement. Puis ils effectueront l'échange de données proprement dit.

- **Implementation de TLS et SSL**

- **Implémentations dans les navigateurs web**
  La majeure partie des implémentations de SSL et TLS se trouve dans les navigateurs et serveurs web. Le serveur Apache, notamment, peut exploiter SSL grâce une implémentation basée sur OpenSSL.
- **OpenSSL**
  Implémenté en C, OpenSSL est une boîte à outils de chiffrement comportant deux bibliothèques (une de cryptographie générale et une implémentant le protocole SSL), ainsi qu'une commande en ligne. OpenSSL supporte SSL 2.0, SSL 3.0 et TLS 1.0. OpenSSL est distribué sous une licence de type Apache.
- **GnuTLS**
  Le projet GnuTLS propose une implémentation du protocole TLS conforme aux spécifications de l'IETF. GnuTLS supporte TLS 1.1, TLS 1.0, SSL 3.0 et les extensions TLS. Il permet l'authentification via les certificats X509 et PGP. A la différence d'OpenSSL, GnuTLS est compatible avec les licences GPL.

- **Comparaison de TLS et SSL avec les autres solutions**

D'autres protocoles permettent d'assurer la sécurité sur le réseau. Bien que proposant des fonctionnalités concurrentes de SSL et TLS, ils sont plutôt considérés comme complémentaires.

- SSH
  SSH est un protocole de niveau application qui propose une alternative sécurisée aux utilitaires classiques (rlogin, rsh, telnet) qui n'offrent pas de confidentialité. La possibilité d'exploiter un mécanisme de tunneling rend SSH, comme SSL et TLS compatible avec les autres protocoles de niveau application déjà existant. Tout comme SSL et TLS, SSH assure l'authentification des machines, la confidentialité et l'intégrité des données. Il assure aussi l'authentification des utilisateurs par mot de passe.
  SSH souffre de faiblesses par rapport à SSL et à TLS : il n'intègre pas la notion de certificats X509 v3, nécessite l'installation d'une application cliente spécifique (pas de transparence).De plus, la notion de tunneling reste difficile à appréhender.
  Cependant, SSH est moins vulnérable que SSL et TLS en matière d'identification du client. En effet, la protection du certificat sur un poste client ne peut pas toujours être correctement assurée.
- IPSec
  IPSec fournit un mécanisme de sécurisation au niveau de la couche réseau (IP). Il est utilisé notamment pour la mise en oeuvre de réseaux privés virtuels (VPN). Les fonctionnalités d'IPSec sont l'authentification des machines, la confidentialité et l'intégrité des transactions.
  Son implémentation indissociable de la prochaine version du protocole IP, IPv6, entre en concurrence avec les fonctionnalités de confidentialité et d'intégrité de SSL et TLS.

Elle offre en outre une sécurisation du réseau dans sa globalité et non des applications au cas par cas.
A ce jour, donc, les fonctionnalités de sécurité d'IPSec et IPv6 sont vues comme un important complément la sécurité offerte par SSL et TLS.

- SET/3D-Secure
  Basé sur SSL et TLS, les protocoles SET (aujourd'hui obsolète) et 3D-Secure proposent une authentification validée par un tiers. Ces protocoles sont principalement destinés aux applications de paiement en ligne (ils sont développés par des institutions bancaires). Si SSL , TLS et SET/3D-Secure assurent chacun un haut degré de confidentialité, seul le SET permet une pleine identification réciproque des deux parties grâce un tiers de confiance,en l'occurrence la banque du vendeur. Ainsi, elle rassure le vendeur que la carte est bonne et qu'elle n'a pas été volée et le client qu'aucune utilisation malveillante ne sera faite de ces informations.

On voit ici que, quoique souffrant de limitations, l'univers SSL et TLS est vaste et stable.

- **Implementation SSL et TLS**

SSL est souvent implémenté au niveau des serveurs et site web . Le serveur Apache exploite aussi ses protocoles en OpenSSL.

OpenSSL est un utilitaire cryptographique qui implémente les protocoles réseau Secure Sockets Layer (SSL v2/v3, Couche de sockets sécurisés) et Transport Layer Security (TLS v1, sécurité pour la couche de transport) ainsi que les standards cryptographiques liés dont ils ont besoin.
Le **openssl** est un outil de ligne de commande pour utiliser les différentes fonctions cryptographiques de la librairie **crypto** d'OpenSSL à partir du shell . Il peut être utilisé pour :
Création de paramètres des clefs RSA, DH et DSA
Création de certificats X.509, CSRs et CRLs
Calcul de signature de messages
Chiffrement et Déchiffrement
Tests SSL/TLS client et server
Gestion de mail S/MIME signé ou chiffrés

Il est distribué sous une licence de type Apache.

Acces internet :

- **Le NAT :**

Les ordinateurs d'un réseau interne n'étant pas directement reliés à l'extérieur. Alors tout passe par la passerelle qui fera le lien entre le réseau interne et internet le réseau externe . Elle traduira les adresses internes en adresses externes et les adresses externes en adresses internes en retour . C'est le Network Address Translation (NAT) ; Traduction d'adresse réseau . Il renforce la sécurité du réseau en protégeant les adresses privées du réseau interne . Il sert de filtre en ne faisant passer que les paquets étant des réponses aux emission des requêtes des hôtes internes . Il est important pour la sécurité du réseau et des serveurs des entreprises du fait de ses changements d'adresses réguliers .

Il existe deux types de **NAT :**

- **NAT Statique :** On parle de NAT ou de translation d'adresse statique quand il s'agit d'effectuer une conversion des paquets d'un point à un autre de façon constante et systématique. Techniquement, le routeur va, à la réception d'un paquet depuis l'extérieur, modifier le champ "IP destination" qui va passer de l'IP externe du routeur à l'IP du serveur en interne, c'est ici qu'agit la translation d'adresse. On peut également parler de NAT de destination.Le NAT statique permet donc de rendre une machine présente dans un LAN ou une DMZ disponible depuis internet. Cela ne va pas dans le sens premier de la création du NAT qui est d'économiser des adresses IPv4 car l'association d'une IP publique vers une IP privée est en un à un. On parle également de **redirection de port** lorsque l'on va rediriger uniquement un port de l'IP externe vers un port (le même ou un autre) d'une IP interne. En gros **n adresses privées = n adresses publiques .** C'est une technique devenue obsolete.

- **NAT Dynamique :** La translation d'adresse dynamique fonctionne elle dans l'autre sens et c'est le but premier de la création du NAT. Il permet de mettre, aux yeux des éléments qui sont du côté de l'interface externe, un ensemble de machine derrière une ou plusieurs IPs. Dans son utilisation courante il peut également être appelé NAT de source car on va effectuer une translation d'adresse (un changement d'information dans les paquets IP) de l'adresse **source** des paquets, les faisant passer des IPs internes vers l'IP externe du routeur.Le routeur va alors avoir une **table de translation** qui va être générée via un mécanisme de **PAT** (Port Address Translation), on va affecter un échange depuis une IP interne vers une IP externe à un port sur l'interface externe. Le bon fonctionnement nous permet d'avoir plusieurs postes "se faisant passer" pour une seule IP côté internet, une correspondance IP_externe:port <--> IP interne est faite pour chaque requête à l'inverse du NAT statique ou une correspondance IP à IP ou IP:port à IP:port est faite de façon automatique et constante. En somme elle donne naissance à un **pool d'adresses .**

- **Le PAT**

Le PAT (Port Address Translation) est la transformation d'un pool d'adresse (NAT dynamique) internes en une seule adresse externe, qui donne la relation n<=>1 (plusieurs-à-un) en utilisant des ports. Dans ce cas, la différenciation des connexions se fait en modifiant le port source.Il est utiliser pour mapper l'adresse privée (locale) non enregistrée d'un réseau interne vers une adresse publique enregistrée d'un réseau externe avant de transférer le paquet. Le **PAT** est la traduction d'adresse de port c'est un type de **NAT** dynamique grâce auquel la traduction d'adresse peut être configurée au niveau du port, et l'utilisation de l'adresse IP est optimisée. **PAT** met en correspondance plusieurs adresses locales et ports sources avec une adresse IP publique et un port à partir d'une liste d'adresses IP routables sur le réseau de destination. Ici, l'adresse IP de l'interface est utilisée en combinaison avec le numéro de port et plusieurs hôtes peuvent avoir la même adresse IP avec un numéro de port unique.
Il utilise une adresse de port source unique sur l'adresse IP globale interne pour identifier des traductions distinctes. Le nombre total de traductions **NAT** pouvant être exécutées est 65536 car le numéro de port est codé sur 16 bits.

Les ACL :

- **Definition et categories**

Une ACL sur un pare-feu ou un routeur filtrant, est une liste d'adresses ou de ports autorisés ou interdits par le dispositif de filtrage.
Les *Access Control List* sont divisés en trois grandes catégories, l'ACL standard, l'ACL étendue et la nommée-étendue.

- L'ACL standard ne peut contrôler que deux ensembles : l'adresse IP source et une partie de l'adresse IP de destination, au moyen de masque générique.
- L'ACL étendue peut contrôler l'adresse IP de destination, la partie de l'adresse de destination (masque générique), le type de protocole (TCP, UDP, ICMP, IGRP, IGMP, etc.), le port source et de destination, les flux TCP, IP TOS (*Type of service*) ainsi que les priorités IP.
- L'ACL nommée-étendue est une ACL étendue à laquelle on a affecté un nom.

Les ACL conviennent bien à des protocoles dont les ports sont statiques , mais ne suffisent pas avec les ports variables.

- **Verifications des paquets et compatibilites**

L' ACL IP est un ensemble séquentiel de permissions ou de restrictions applicables à un paquet IP. Le routeur teste les paquets en fonction des conditions présentes dans l'ACL, les unes après les autres.
La première correspondance détermine si le logiciel Cisco IOS® doit accepter ou refuser le paquet. Puisque le Logiciel Cisco IOS arrête le test des conditions après la première correspondance, l'ordre des conditions est essentiel. Si aucune condition ne possède de correspondance, le routeur refuse le paquet en raison d'une clause implicite de refus de tous les paquets.
Voici des exemples d'ACL IP qui peuvent être configurées dans le logiciel Cisco IOS :

- ACL standards
- ACL étendus
- ACL dynamiques (verrou et clé)
- ACL nommées IP
- Listes de contrôle d'accès réflexives
- ACL basées sur l'heure, qui utilisent des plages temporelles
- Entrées de liste de contrôle d'accès IP commentées
- ACL basées sur le contexte
- Proxy d'authentification
- Listes de contrôle d'accès turbo
- Listes de contrôle d'accès basées sur l'heure distribuées

Les ACL standards comparent l'adresse de la source du paquet IP aux adresses configurées dans l'ACL afin de contrôler le trafic.
Les ACL étendus comparent l'adresse de la source du paquet IP et son adresse de destination aux adresses configurées dans l'ACL afin de contrôler le trafic. Vous pouvez également rendre les ACL étendues plus précises et les configurer pour le filtrage du trafic selon des critères tels que :

- **Protocol**
- Numéros de port
- Valeur de point de code de services différenciés (DSCP - Differentiated services code point)

- Valeur de priorité
- État du bit de numéro de séquence (SYN - Synchronize Sequence Number)

- **Principe de fonctionnement des ACL :**

Exemples de configuration pour les listes de contrôle d'accès (ACL) fréquemment utilisées, qui filtrent les paquets IP sur la base des éléments suivants :

- Adresse source
- Adresse de destination
- Type de paquet
- Toute combinaison de ces éléments

Pour filtrer le trafic réseau, les listes de contrôle d'accès vérifient si les paquets sont acheminés ou bloqués au niveau de l'interface du routeur. Le routeur examine chaque paquet afin de déterminer s'il doit être acheminé ou abandonné, selon les critères spécifiés dans l'ACL (Access Control List, liste de contrôle d'accès). Les critères de la liste ACL incluent :

- Adresse source du trafic
- Adresse de destination du trafic
- Protocole de la couche supérieure

- **Creation des ACL :**
  Créez une liste ACL.
  Appliquez l'ACL à une interface.

- **VPN**

Un réseau privé virtuel (*Virtual Private Network* , *VPN*) est vu comme une extension des réseaux locaux et préserve la sécurité logique que l'on peut avoir à l'intérieur d'un réseau local. Il correspond en fait à une interconnexion de réseaux locaux via une technique de «tunnel». La technique consiste à utiliser Internet comme support de transmission en utilisant un protocole de « tunnellisation » (*tunneling*), c'est-à-dire encapsulant les données à transmettre de façon chiffrée. On parle alors de VPN pour désigner le réseau ainsi artificiellement créé. Ce réseau est dit virtuel car il relie deux réseaux « physiques » (réseaux locaux) par une liaison non fiable (Internet), et privé car seuls les ordinateurs des réseaux locaux de part et d'autre du VPN peuvent accéder aux données en clair.
Le VPN vise à apporter certains éléments essentiels dans la transmission de données :

- l'authentification (et donc l'identification) des interlocuteurs,
- la confidentialité des données (le chiffrement vise à les rendre inutilisables par quelqu'un d'autre que le destinataire).

Il devient quasiment indispensable d'utiliser un VPN pour sécuriser les données. De cette façon, aucun hacker ne peut décrypter les données qui transitent via Le VPN, ce qui vous offre une sécurité optimum dans le transfert des données.
Il existe plusieurs types de protocole VPN dont IPsec , VPN SSL/TLS et OpenVPN

- **IPSec :**

IPSec (Internet Protocol Security, RFC 2401) est un protocole de la couche 3 du modèle OSI, tout comme IP. IPSec est un protocole destiné à fournir différents services de sécurité. Son intérêt principal reste sans conteste son mode dit de tunneling, c'est-à-dire d'encapsulation d'IP qui lui permet entre autres choses de créer des réseaux privés virtuels.
les services de sécurité proposés par IPSec :

- Authentification des extrémités : cette authentification mutuelle permet à chacun de s'assurer de l'identité de son interlocuteur. Rappelons tout de même qu'IPSec est un protocole de niveau 3 et qu'il ne fournit donc qu'une authentification de niveau égal, c'est-à-dire une authentification des machines mettant en oeuvre le protocole plutôt que des personnes utilisant réellement la machine. Nous verrons techniquement comme l'authentification est effectuée dans les paragraphes suivants.
- Confidentialité des données échangées : IPSec permet si on le désire de chiffrer le contenu de chaque paquet IP pour éviter que quiconque ne le lise.
- Authenticité des données : IPSec permet de s'assurer, pour chaque paquet échangé, qu'il a bien été émis par la bonne machine et qu'il est bien à destination de la seconde machine.
- Intégrité des données échangées : IPSec permet de s'assurer qu'aucun paquet n'a subit de modification quelconque (attaque dite active) durant son trajet.
- Protection contre les écoutes et analyses de trafic : IPSec permet de chiffrer les adresses IP réelles de la source et de la destination, ainsi que tout l'en-tête IP correspondant. C'est le mode de tunneling, qui empêche tout attaquant à l'écoute d'inférer des informations sur les identités réelles des extrémités du tunnel, sur les protocoles utilisés au-dessus d'IPSec, sur l'application utilisant le tunnel (timing-attacks et autres)...
- Protection contre le rejeu : IPSec permet de se prémunir contre les attaques consistant à capturer un ou plusieurs paquets dans le but de les envoyer à nouveau (sans pour autant les avoir déchiffrés) pour bénéficier des même avantages que l'envoyeur initial.

PSec repose en fait sur plusieurs protocoles différents -dont certains existent à part entière hors d'IPSec- qui lui offrent en retour une grande souplesse d'utilisation.

Le protocole initial et principal est le protocole **IKE** (Internet Key Exchange, RFC 2409). Appliqué à IPSec, ce protocole a pour objectif dans un premier temps d'établir un premier tunnel entre les 2 machines (le tunnel IKE), que l'on pourra qualifier de "tunnel administratif". C'est la phase 1 du protocole IKE. Ce protocole est dit administratif car il ne sert pas à la transmission des données utilisateur; il est utilisé pour gérer les tunnels secondaires, leur création, le rafraîchissement des clés, etc... La phase 2 du protocole IKE consiste en effet à établir autant de tunnels secondaires que nécessaire pour la transmission des données utilisateur entre les 2 machines. Notez qu'il est possible de recourir à une authentification manuelle à la place d'IKE, mais comme ce dernier permet bien plus de choses que de l'authentification, cela s'avère beaucoup plus difficile à utiliser. Le protocole IKE est décrit dans le paragraphe sur l'initialisation d'IPSec.

Les tunnels destinés aux échanges de données vont s'appuyer sur 2 protocoles différents suivant les besoins en sécurité des utilisateurs. Le premier est le protocole **AH** (Authentication Header, RFC 2402) qui vise à établir l'identité des extrémités de façon certaine. Il ne garantit aucune confidentialité (chiffrement) des données. Le deuxième protocole est le protocole **ESP** (Encapsulating Security Payload, RFC 2406) qui a pour but de chiffrer les données, avec ou sans les entêtes des paquets si l'on souhaite le mode tunneling. Il garantit également l'authenticité des données et, à ce niveau, peut introduire de la redondance par rapport à AH. Les aspects cryptographiques de ces protocoles seront décrits dans un paragraphe dédié.

Ces 2 protocoles, AH et ESP, peuvent d'autre part être utilisés séparément ou combinés.

- **Mode**

Le mode **Transport** ne modifie pas l'en-tête initial; il s'intercale entre le protocole réseau (IP) et le protocole de transport (TCP, UDP...). Plusieurs variantes existent, conformément aux protocoles décrits plus haut :
Le mode **Tunnel** remplace les en-têtes IP originaux et encapsule la totalité du paquet IP. Par exemple, l'adresse IP_A externe pourra être celle de la passerelle de sécurité implémentant IPSec, et l'adresse IP_B interne sera celle de la machine finale, sur le réseau derrière la passerelle.
Le mode de **Nesting** est hybride puisqu'il utilise les 2 modes cités précédemment. Il s'agit bien d'encapsuler de l'IPSec dans de l'IPSec; nous verrons plus tard les répercussions au niveau implémentation (bundle de SAs)

- **Contraintes :**

IPSec n'a pas été conçu initialement pour ce type d'accès distant. En effet IPSec a été à l'origine pensé pour un monde où les PCs possèdent des adresses statiques. De nos jours la mobilité est très importante et les utilisateurs changent d'adresse IP même si ils se connectent toujours depuis le même endroit. De plus IPSec nécessite qu'un logiciel spécifique (client VPN) soit installé sur l'ordinateur utilisé pour la connexion. Cette obligation est contraignante et il existe des problèmes de compatibilités entre OS et clients VPN. Des add-on de sécurité sont très souvent installés en complément du client VPN pour garantir un niveau de sécurité maximum. Toutes ces contraintes font d'IPSec un protocole peu adapté dans le cadre d'accès distant. IPSec ne permet d'identifier que des machines et non pas des utilisateurs. Ceci est particulièrement problématique pour les utilisateurs itinérants. Il faut donc prévoir un service

d'authentification des utilisateurs. Dans le cas de connexion dial-up c'est l'identifiant de connexion qui sera utilisé pour authentifier l'utilisateur. Mais dans le cas de connexion via Internet il faudra prévoir une phase d'authentification supplémentaire à l'établissement du tunnel.
D'autre part IPSec n'offre aucun mécanisme de Qos Ce qui limite ses applications : toutes les applications de voix sur Ip ou de vidéo sur Ip sont impossibles ou seront amenées à être complètement dépendantes des conditions de trafic sur l'internet public.
Enfin IPSec à cause de la lourdeur des opérations de cryptage/décryptage réduit les performances globales des réseaux. L'achat de périphériques dédiés, coûteux est souvent indispensable.

- **VPN SSL**

Le client commence la session
- Indique la version de SSL
- Génère un nombre aléatoire (client_random)
- Suites de chiffrement supportées
Le serveur confirme et envoie son certificat

- Choix de la version SSL
- Génère un nombre aléatoire (server_random)
- Choix de la suite de chiffrement
- Certificat X.509
- Le client vérifie le certificat (date d'expiration, vérification du nom de domaine du serveur, vérification du certificat par la clé publique de l'autorité de certification)
Le client établit le canal sécurisé
- Génère un paramètre PreMasterSecret qu'il chiffre avec la clé publique du serveur et qu'il envoie.
- Le serveur décrypte le PremasterSecret, génère le MasterSecret et calcule les clés secrètes.
- Le client génère le même MasterSecret et calcule les même clés.
- F(client_random, server_random,PreMasterSecret) = MasterSecret
- Puis F(client_random,server_random,MasterSecret) = Clés Secrètes
- Clés secrètes= Clés de chiffrement, Vecteurs d'initialisation et clés de hachage.
Le cryptage SSL, fonctionne par le choix aléatoire de deux nombres premiers, qui multipliés entre eux forment un très grand nombre. Ce dernier constitue la clé de cryptage. Sans la connaissance des deux nombres premiers ayant servis à générer cette clé, il n'est pas possible de pouvoir décrypter un message. En réalité une manière possible serait de défactoriser le nombre afin de retrouver les deux nombres premiers, mais les nombres sont tellement grands, que cela n'est pas à la portée d'ordinateurs conventionnels. Il est à relever que déjà à partir de sa version 2.0, SSL a commencé à être utilisé. Raison pour laquelle encore aujourd'hui, la plupart des intervenants SSL, tentent lors de la phase d'établissement, de dialoguer avec le

protocole v3.0, mais si l'un des deux partenaires ne supporte que la version 2.0, et bien c'est cette version antérieure qui va être utilisée. A noter que cette ancienne version contient des clés de cryptage considérées aujourd'hui comme peu sûres. (Par exemple au niveau de la longueur de la clé, il y a un passage de 40 à 128 bits et plus, pour la version v3.0).
SSL se subdivise en quatre sous protocoles; le SSL record Protocol, et le SSL handshake protocol. Plus deux autres protocoles, mais qui ont un rôle moins essentiel, c'est le SSL Change Cipher Spec, et le SSL Alert.
Le SSL record protocol définit le format qui sera utilisé pour l'échange des données. Alors que le SSL handshake se charge des différents échanges de messages entre le client et le serveur, au moment où ils établissent la connexion comme l'authentification, le version de protocole, l'algorithme de cryptage, …

- **Contraintes :**

L'algorithme AES pour le chiffrement des données alors que la majorité des applications SSL ne l'ont pas encore implémenté.
Présentée comme la solution miracle pour permettre aux itinérants de se connecter aux applications réparties de l'entreprise les VPN-SSL souffrent de problèmes principalement liés aux navigateurs web utilisés due aux attaques tunnels à travers leurs vulnérabilités.
Le but d'utiliser des navigateurs web est de permettre aux utilisateurs d'utiliser un outil dont ils ont l'habitude et qui ne nécessite pas de configuration supplémentaire. Le renouvellement d'un certificat expiree par l'utilisateur manuellement peut poser problème aux utilisateurs novices. La majorité des navigateurs web la consultation des listes de certificats révoqués n'est pas activée par défaut : toute la sécurité de SSL reposant sur ces certificats ceci pose un grave problème de sécurité.
Enfin Un autre problème lié à l'utilisation de navigateurs web comme base au VPN est leur spécificité au monde web. En effet par défaut un navigateur n'interceptera que des communication Https ou éventuellement Ftps. Toutes les communications venant d'autre type d'applications (MS Outlook, ou une base de données par exemple) ne sont pas supportées. Ce problème est généralement contourné par l'exécution d'une applet Java dédiée dans le navigateur. Mais ceci implique également la maintenance de cette applet (s'assurer que le client possède la bonne version, qu'il peut la re-télécharger au besoin)
L'idée suivant laquelle le navigateur web est une plate-forme idéale pour réaliser des accès VPN est donc sérieusement à nuancer.

- **OpenVPN :**

Le Protocole **Openvpn** est une application informatique ouverte pour la mise en place de techniques de réseaux privés virtuels (VPN, Virtual Private Network), avec des connexions sécurisées point-par-point ou site-par-site, pour des configurations via routage ou pont, ainsi que pour les accès à distance. Il exploite un protocole de sécurité sur mesure qui utilise SSL/TLS pour les échanges clés. Il est capable de traverser des transpondeurs de réseaux d'adresses (NAT : Network Address Translators) et des pare-feu. Il a été défini par James Yonan et est publié sous licence publique GNU en tant que GPL (General Public License).
Un protocole Openvpn permet à des homologues de s'authentifier mutuellement en utilisant une clé secrète pré-partagée, des certificats ou un nom d'utilisateur / mot de passe. Lorsqu'il est utilisé dans une configuration multi client-serveur, il permet au serveur de libérer un certificat d'authentification pour chaque client, en utilisant la signature et l'autorité de certification. Ce système utilise en grande partie la base de cryptage OpenSSL, ainsi que le protocole SSLv3/TLSv1 et contient de nombreuses fonctionnalités de sécurité et de contrôle.
il s'agit d'une technologie à source ouverte qui utilise les ports TCP UDP pour la transmission. OpenVPN windows offre également aux utilisateurs un cryptage AES 256 bits. En raison de sa nature open source, toute vulnérabilité de sécurité rapportée publiquement est généralement corrigée par la communauté open source.
Ce protocole est largement utilisé en raison des divers avantages qu'il offre. Pour commencer, contrairement aux autres protocoles courants, OpenVPN est compatible avec les systèmes d'exploitation mobiles tels qu'Android et iOS. Plus important encore, il peut contourner n'importe quel pare-feu et accéder à un certain nombre de ports pour la communication.

- **Avantages :**

Open VPN offre la meilleure vitesse et une plus grande sécurité pendant votre connexion VPN et il permet de passer la plupart des pare-feu et des restrictions réseaux/FAI.

- Il est soutenu par la communauté open source qui détecte et corrige rapidement tous les problèmes de sécurité.
- Capable de contourner tout pare-feu car il prend en charge plusieurs ports.
- Extrêmement sécurisé car il utilise un cryptage AES jusqu'à 256 bits.
- Fournit le meilleur mélange de sécurité et de rapidité.
- Compatible avec les principaux systèmes d'exploitation, notamment Windows , Mac, Linux, Android et iOS.

- **Inconvenients :**

Openvpn est moins facile à installer que d'autres protocoles VPN, et l'installation d'un VPN avec OpenVPN peut être complexe pour certains clients, vu qu'elle nécessite d'installer une application spéciale pour le client. Le protocole Openvpn n'est pas non plus pris en charge par certains appareils mobiles, ce qui peut être un inconvénient important pour l'utilisation d'un VPN mobile.

- Nécessite que des logiciels et des applications tiers soient exécutés sur tous les systèmes d'exploitation, car ils ne peuvent pas être configurés directement.
- L'installation n'est pas facile et nécessite des connaissances techniques.

Choix de solution

- **Acces distant :**

Nous utiliserons le **SSH** pour une connexion sécurisée à distance en mode terminal ou console de manière sécurisée grâce aux algorithmes .

- **Acces internet :**

Pour bénéficier des avantages du **NAT statique** sans les inconvénients du **NAT dynamique** on optera pour un mix **NAT/PAT** .

- **ACL**

- **VPN :**

Quand au **VPN** on optera pour le protocole **IPSec** seul capable de crypter les données échangées.

Mise en place VLAN , attribution port et port trunk

```
Switch>en
Switch#
00:01:54: %LINK-5-CHANGED: Interface Vlan1, changed state to administratively down
00:01:55: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
Switch#
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#hostname com1-srv
com1-srv(config)#vlan 10
com1-srv(config-vlan)#name administration
com1-srv(config-vlan)#exit
com1-srv(config)#vlan 11
com1-srv(config-vlan)#name equipes
com1-srv(config-vlan)#exit
com1-srv(config)#vlan 12
com1-srv(config-vlan)#name wifi-stades
com1-srv(config-vlan)#exit
com1-srv(config)#exit
```

Verification mise en place VLAN

```
com1-srv#show vlan

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                 Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                 Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                 Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                 Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                 Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                 Gi0/1, Gi0/2
10   administration                    active
11   equipes                           active
12   wifi-stades                       active
1002 fddi-default                      act/unsup
1003 token-ring-default                act/unsup
1004 fddinet-default                   act/unsup
1005 trnet-default                     act/unsup

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        0      0
10   enet  100010     1500  -      -      -        -    -        0      0
11   enet  100011     1500  -      -      -        -    -        0      0
12   enet  100012     1500  -      -      -        -    -        0      0
1002 fddi  101002     1500  -      -      -        -    -        0      0
1003 tr    101003     1500  -      -      -        -    -        0      0
1004 fdnet 101004     1500  -      -      -        ieee -        0      0
1005 trnet 101005     1500  -      -      -        ibm  -        0      0

Remote SPAN VLANs
------------------------------------------------------------------------------


Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------------
```

Attribution de Ports

```
com1-srv(config)#interface range fa0/1 - 6
com1-srv(config-if-range)#switchport access vlan 10
com1-srv(config-if-range)#no sh
com1-srv(config-if-range)#exit
com1-srv(config)#
com1-srv(config)#interface range f0/7 - 12
com1-srv(config-if-range)#switchport access vlan 11
com1-srv(config-if-range)#no sh
com1-srv(config-if-range)#exit
com1-srv(config)#
com1-srv(config)#interface range fa0/13 - 14
com1-srv(config-if-range)#switchport access vlan 12
com1-srv(config-if-range)#no sh
com1-srv(config-if-range)#exit
com1-srv(config)#
com1-srv(config)#exit
```

Verification attributions de port

```
com1-srv#show vlan

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/15, Fa0/16, Fa0/17, Fa0/18
                                                 Fa0/19, Fa0/20, Fa0/21, Fa0/22
                                                 Fa0/23, Fa0/24, Gi0/1, Gi0/2
10   administration                    active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                 Fa0/5, Fa0/6
11   equipes                           active    Fa0/7, Fa0/8, Fa0/9, Fa0/10
                                                 Fa0/11, Fa0/12
12   wifi-stades                       active    Fa0/13, Fa0/14
1002 fddi-default                      act/unsup
1003 token-ring-default                act/unsup
1004 fddinet-default                   act/unsup
1005 trnet-default                     act/unsup

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        0      0
10   enet  100010     1500  -      -      -        -    -        0      0
11   enet  100011     1500  -      -      -        -    -        0      0
12   enet  100012     1500  -      -      -        -    -        0      0
1002 fddi  101002     1500  -      -      -        -    -        0      0
1003 tr    101003     1500  -      -      -        -    -        0      0
1004 fdnet 101004     1500  -      -      -        ieee -        0      0
1005 trnet 101005     1500  -      -      -        ibm  -        0      0

Remote SPAN VLANs
------------------------------------------------------------------------------


Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------------
```

Attribution Ports Trunk

```
com1-srv(config)#interface range fa0/22 - 24
com1-srv(config-if-range)#switch mode trunk
com1-srv(config-if-range)#no sh
00:20:13: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24, changed state to down
com1-srv(config-if-range)#
00:20:16: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24, changed state to up
com1-srv(config-if-range)#exit
com1-srv(config)#
com1-srv(config)#exit
```

Configuration interfaces routeurs :

```
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname r-stade
r-stade(config)#interface fa0/1
r-stade(config-if)#ip address dhcp
r-stade(config-if)#no sh
r-stade(config-if)#exit
r-stade(config)#
r-stade(config)#interface fa 0/0
r-stade(config-if)#no shutdown
r-stade(config-if)#exit
r-stade(config)#
r-stade(config)#
r-stade(config)#
*Nov  7 08:12:02.631: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Nov  7 08:12:03.631: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
r-stade(config)#
r-stade(config)#interface fa0/0.10
r-stade(config-subif)#description vlan 10
r-stade(config-subif)#encapsulation dot1Q 10
r-stade(config-subif)#ip address 172.20.0.1 255.255.255.0
r-stade(config-subif)#no shutdown
r-stade(config-subif)#exit
r-stade(config)#
r-stade(config)#interface fa0/0.11
r-stade(config-subif)#description vlan 11
r-stade(config-subif)#encapsulation dot1Q 11
r-stade(config-subif)#ip address 172.20.1.1 255.255.255.0
r-stade(config-subif)#no shutdown
r-stade(config-subif)#exit
r-stade(config)#
r-stade(config)#interface f0/0.12
r-stade(config-subif)#description vlan 12
r-stade(config-subif)#encapsulation dot1Q 12
r-stade(config-subif)#ip address 172.20.2.1 255.255.255.0
r-stade(config-subif)#no shutdown
r-stade(config-subif)#exit
r-stade(config)#
r-stade(config)#
r-stade(config)#exit
```

Verification des interfaces :

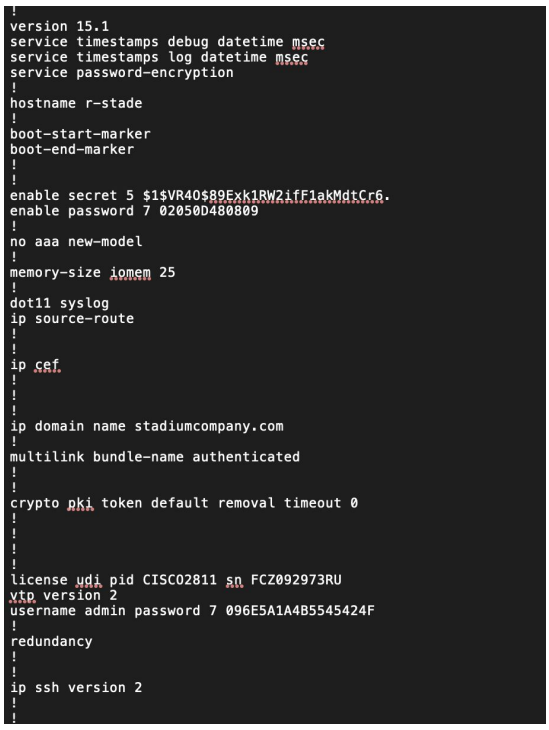
```
r-stade#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            unassigned      YES unset  up                    up
FastEthernet0/0.10         172.20.0.1      YES manual up                    up
FastEthernet0/0.11         172.20.1.1      YES manual up                    up
FastEthernet0/0.12         172.20.2.1      YES manual up                    up
FastEthernet0/1            172.20.87.44    YES DHCP   up                    up
r-stade#
*Nov  7 08:25:08.875: %DHCP-6-ADDRESS_ASSIGN: Interface FastEthernet0/1 assigned DHCP address 172.20.87.44, mask 255.255.240.0, hostname r-stade
```

Configuration ssh :

r-stade (config) # enable password cisco
enable secret cisco123
service password-encryption
r-stade(conf) # ip domain-name stadiumcompany.com
crypto key generate rsa
r-stade(config) # username admin password Bts2020$
r-stade(config) # line vty 0 4
login local
transport input SSH
exit
r-stade(config) # ip ssh version 2
ip ssh authentication-retries 3
ssh time-out 120

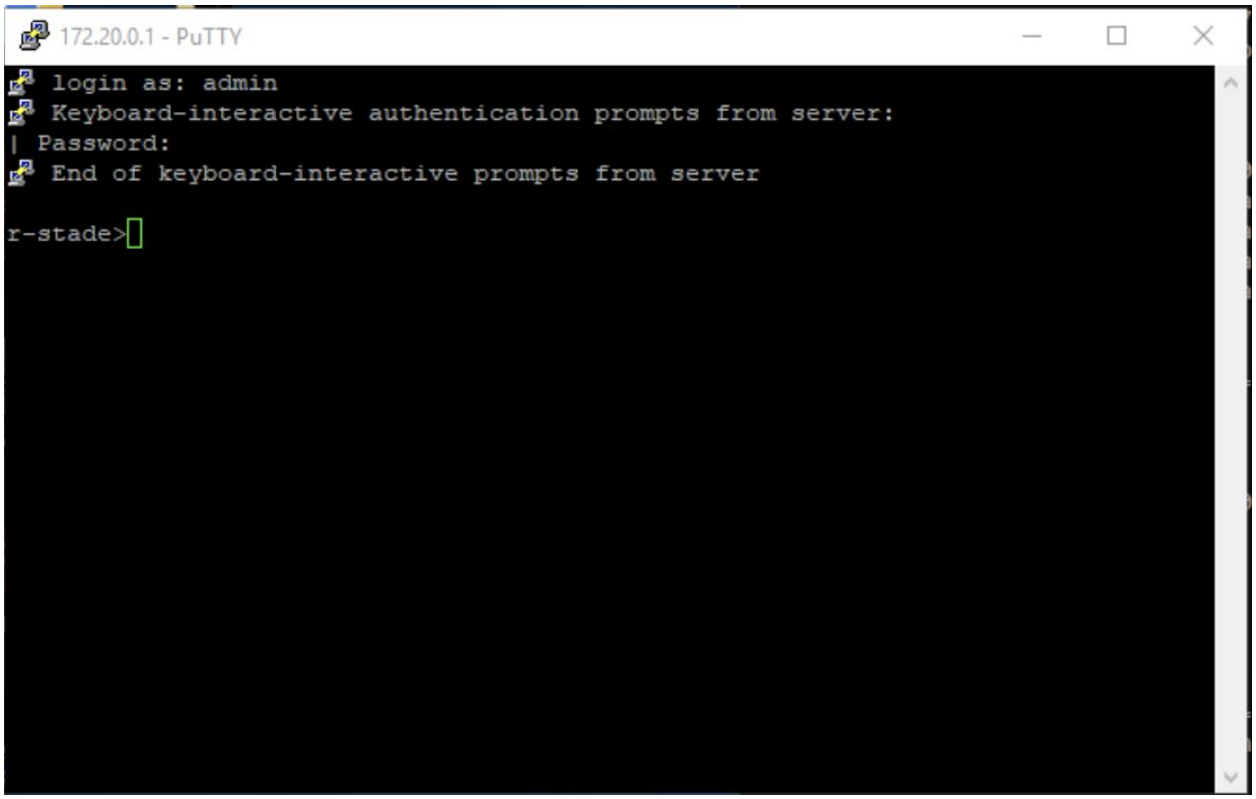
Cle de chiffrement :

Verification fonctionnement SSH



Verification SSH via Putty en se connectant a l'adresse du routeur 172.20.0.1
Alerte **cle de chiffrement (la cle de chiffrement est unique)**

Demande d'identification pour la connexion securisee



Mise en place du NAT / PAT

Parametrage interface

```
r-stade>en
Password:
r-stade#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
r-stade(config)#interface fa0/0.10
r-stade(config-subif)#ip nat inside

*Nov  7 09:43:38.395: %LINEPROTO-5-UPDOWN: Line protocol on Interface NVI0, chan
r-stade(config-subif)#
r-stade(config-subif)#
r-stade(config-subif)#
r-stade(config-subif)#exit
r-stade(config)#
r-stade(config)#interface fa0/0.11
r-stade(config-subif)#ip nat inside
r-stade(config-subif)#exit
r-stade(config)#interface fa0/0.12
r-stade(config-subif)#ip nat inside
r-stade(config-subif)#exit
r-stade(config)#
r-stade(config)#interface fa0/1
r-stade(config-if)#ip nat outside
r-stade(config-if)#exit
```

Creations d'ACL

```
r-stade(config)#access-list 10 permit 172.20.0.0 0.0.0.255
r-stade(config)#access-list 11 permit 172.20.0.0 0.0.0.255
r-stade(config)#access-list 12 permit 172.20.2.0 0.0.0.127
r-stade(config)#access-list 11 permit 172.20.1.0 0.0.0.255
r-stade(config)#
r-stade(config)#ip nat inside source list 10 interface fa0/1 overload
r-stade(config)#ip nat inside source list 11 interface fa0/1 overload
r-stade(config)#ip nat inside source list 12 interface fa0/1 overload
```

Verification ACL

Vérification Traduction d’adresse

F- Les activites

A1.1.1 Analyse du cahier des charges d'un service à produire
A1.2.1 Élaboration et présentation d'un dossier de choix de solution technique
A1.2.2 Rédaction des spécifications techniques de la solution retenue (adaptation d'une solution existante ou réalisation d'une nouvelle solution)
A1.2.3 Evaluation des risques liés à l'utilisation d'un service
A1.2.4 Détermination des tests nécessaires à la validation d'un service
A1.3.1 Test d'intégration et d'acceptation d'un service
A1.3.4 Deploiement d'un service
A1.4.1 Participation a un projet
A3.1.1 Proposition d'une solution d'infrastructure
A3.1.2 Maquettage et prototypage d'une solution d'infrastructure
A3.1.3 Prise en compte du niveau de sécurité nécessaire à une infrastructure
A3.2.1 Installation et configuration d'éléments d'infrastructure
A3.2.2 Planification des sauvegardes et gestion des restaurations
A3.3.1 Administration sur site ou à distance des éléments d'un réseau , de serveurs , de services et d'équipements terminaux
A4.1.9 Rédaction d'une documentation technique
A5.1.2 Recueil d'informations sur une configuration et ses éléments
A5.1.5 Évaluation d'un élément de configuration ou d'une configuration
A5.2.1 Exploitation des referentiels , normes et standards adoptés le prestataire informatique
A5.2.2 Veille technologique
A5.2.4 Étude d'une technologie , d'un composant , d'un outil ou d'une méthode