Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de génie électrique

MÉMOIRE DE MASTER

Sciences et Technologies
Télécommunications
Réseaux et télécommunications

Réf. : Entrez la référence du document

Présenté et soutenu par :
MEZHOUDI YAZID

Le : jeudi 26 avril 2018

Voix sur IP Sécurisée

Jury :

Mme.	HEMAIZIA Zohra	MCA	Université de Biskra	Président
M.	GUESBAYA Tahar	MCA	Université de Biskra	Rapporteur
M.	TERRISSA Sadek Labib	MCA	Université de Biskra	Examinateur

Année universitaire : 2017/2018

Dédicace

Je dédie ce mémoire à :

Mes très chers parents qui ont toujours été là pour moi

Mes frères et sœurs qui sont restées toujours à mes coté.

Mes collègues de travail à l'ENSP.

Mes professeurs de l'Université Mohamed KHEIDER de Biskra

Ma femme.

Et surtout à ma petite fillette Meriem.

Y.MEZHOUDI

Remerciements :

Tout d'abord, je remercie le Dieu, notre créateur de m'avoir donné la forces, la volonté et le courage afin de reprendre mes études après une interruption de plus de vingt ans, ainsi que d'accomplir ce travail malgré mes obligations professionnelles avec la société où je travail.

J'adresse le grand remerciement à mon encadreur Mr. GUESBAYA, pour ses conseils et ses dirigés du début à la fin de ce travail.

Egalement je tiens à remercier messieurs les membres de jury pour l'honneur qu'ils acceptant de siéger à ma soutenance.

Finalement, j'exprime mais profonde gratitude à ma famille et à tout ce qui participe de réaliser ce mémoire.

Résumé

La VoIP permet aux gens d'utiliser le protocole IP, employé dans les réseaux locaux et Internet, comme moyen de transmission pour les communications vocales. Plusieurs protocoles peuvent être envisagés pour cette technologie (SIP, H323, etc.).

Astersik est un outil qui permet de fournir à Linux un commutateur téléphonique complet et totalement libre. Dans ce mémoire, nous avons configuré un serveur SIP pour la VoIP sous Asterisk tout en présentant toute les étapes de création et aussi en considérant les problèmes de sécurité. En effet, le nombre d'attaques possibles sur ce type de serveur est considérable. Sécuriser le serveur n'est pas seulement une nécessité mais plutôt une obligation. Nous avons présenté les principales solutions qui permettent de sécuriser le serveur (sécurisation des comptes SIP, sécurisation de la communication entre les clients, …etc.). Nous avons aussi utilisé des outils pour montrer que notre serveur pour la VoIP est sécurisé.

ملخص

لقد تمكنت تقنية استعمال البروتوكول IP من نقل الصورة و الصوت بعد ما كانت مقتصرة على نقل البيانات بفضل الابحاث المتعددة في مجال VOIP , و ذلك باستحداث ادوات و بروتوكولات تتماشى و طبيعة الشبكة الناقلة . من بين هذه البروتكولات و التي تعتبر اكثر رواجا نجد كل من SIP و H323

بل اكثر من ذلك قد ظهرت على شبكة الإنترنت بعض المحولات الهاتفية المجانية المتكاملة ابرزها Asterisk الذي أنشئ خصيصا للعمل تحت نضام اللينوكس .

في هذا البحث قمنا بتنصيب سيرفر لنقل الصوت عن طريق Asterisk يعمل بالبرتكول SIP. حيث عرضنا جميع مراحل تنصيب السيرفر , كما تطرقنا لمشاكل تامين هذا السيرفر وكذلك لمختلف الهجمات التي قد يكون عرضت لها .

و لقد تأكدنا انه صار لزاما تامين هذا السيرفر لذا قدمنا اهم الحلول الموجودة في هذا المجال (تامين الحسابات sip و تامين المكالمات بين المستعملين). كذلك استعملنا بعض الادوات للتأكد من مدى حماية سيرفر Asterisk.

Sommaire

Liste des figures

Liste des tableaux

Liste des abréviations

A

ACL = Access Control List

ADSL= Asymmetric digital subscriber link

AH = Authentication Header

ARP = Address Resolution Protocol

C

CLI = Command Line Interface

D

DDoS = Distributed Denial of Service

DHCP = Dynamic Host Configuration Protocol

DMZ = Demilitarized Zone

DNS = Domain Name System

DoS = Deny of Service

DSP= digital signal processing

DTMF = Dual-Tone Multi-Frequency

E

ESP = Encapsulated Security Payload

F

FTP = File Transfer Protocol

G

GSM = Global System for Mobile Communications

H

HTTP = Hyper Text Transfer Protocol

I

IAX = Inter-Asterisk Exchange

ICMP = Internet Control Message Protocol

IETF = Internet Engineering Task Force

IGMP = Internet Group Management Protocol

IGRP = Interior Gateway Routing Protocol

IM = Instant Message

IP = Internet Protocol Security

IPsec = Internet Protocole de sécurité

ISDN = Integrated Service Data Network

ITU = International Telecommunications Union

L

LAN = Local Area Network

M

MD5 = Message Digest 5

MKI = Master Key identifier

N

NAT = Network Address Translation

O

OS = Operating System

P

PABX = Private Automatic Branch exchange

PBX = Private Branch exchange

PSTN = Public Switched Telephone Network

Q

QoS = Quality of Service

R

RFC = Requests For Comment

RNIS = Réseau Numérique à Intégration de Service

RTC = Réseau Téléphonique Commuté

RTCP = Real-time Transport Control Protocol

RTP = Real-Time Transport Protocol

RTSP = Real Time Streaming Protocol

S

SIP = Session Initiation Protocol

SNMP = Simple Network Management Protocol

SRTP = Secure Real-time Transport Protocol

T

TCP = Transport Control Protocol

TDM = Time Division Multiplexing

TFTP = Trivial File Transfer Protocol

TLS = Transport Layer Security

ToIP = Telephony over Internet Protocol

U

UAC = User Agent Client

UAS = User Agent Server

UDP = User Datagram Protocol

URL = Uniform Resource Locator

V

VoIP = Voice over Internet Protocol

VPN = Virtual Private Network

W

WAN = World Area Network

Introduction générale

La technologie de commutation de circuits est utilisée depuis longtemps par les opérateurs de réseau téléphonique public (RTCP) pour acheminer le trafic vocal.

Dans ce type de technologie, avant que les utilisateurs puissent communiquer, un canal ou circuit dédié est établi entre l'expéditeur et le récepteur et ce chemin est sélectionné sur la route la plus efficace en utilisant des commutateurs intelligents.

En conséquence, il n'est pas nécessaire qu'un appel téléphonique du même expéditeur au même destinataire prenne le même trajet chaque fois qu'un appel téléphonique est effectué.

Lors de l'installation de l'appel, une fois l'itinéraire déterminé, ce trajet ou circuit reste fixe tout au long de la communication, et les ressources nécessaires sur le trajet sont allouées à l'appel téléphonique du début à la fin.

Le circuit établi ne peut pas être utilisé par d'autres appelants tant que il n'est pas relâché, il reste indisponible pour d'autres utilisateurs, même si aucune communication réelle est en cours.

L'inconvénient majeur de la commutation de circuits est la mauvaise exploitation des circuits mis en place pour la durée de la communication. Dans une communication téléphonique, par exemple, il est rare d'avoir que les communicants parlent simultanément. Les interlocuteurs parlant généralement l'un après l'autre. Le circuit est donc utilisé, au plus, la moitié du temps. Si on enlève les blancs du signale parole, l'utilisation réelle du circuit est encore plus faible.

Une technologie alternative aux réseaux téléphoniques à commutation de circuits pour acheminer le trafic vocal consiste à utiliser des réseaux de commutation de paquets centrés sur les données tels que les réseaux IP (Internet Protocol). Dans la technologie de commutation par paquets, aucun circuit n'est construit entre l'expéditeur et le destinataire. Les paquets sont envoyés sur la route la plus efficace au moment du dialogue ; par conséquent des paquets différents peuvent emprunter des routes différentes venantes du même expéditeur allant vers le même destinataire.

CHAPITRE I: Généralités sur la voix sur IP

I.1. Introduction

En plein développement depuis quelques années, la VoIP représente une alternative à la communication téléphonique traditionnelle. Elle fournit des services que n'offre pas la téléphonie classique et propose certaines prestations à des coûts bien inférieurs aux tarifs pratiqués par certains opérateurs téléphoniques.

La VoIP (Voice over IP), ou Voix sur IP en français, désigne une technique permettant la communication par la voix sur des réseaux compatibles IP. La voix peut être une voix humaine, mais aussi un flux audio ou vidéo. Les réseaux compatibles IP se résument aux réseaux privés, Internet, Wi-Fi, satellite, ADSL, GSM, etc. La VoIP est la technologie chargée de transporter la voix sur l'un de ces réseaux.

I.2. Principe et fonctionnement

I.2.1. Principe

Avant de commencer la transmission audio sur le réseau IP, vous devez convertir les vibrations sonores générées par les différents objets en un signal gérable par un tel réseau. Pour cela :

En émission le signal doit être en premier lieu convertir en signal électrique analogique, en suite en passe à leur numérisation et compression selon les codecs utilisés. Le signal obtenu est découpé en paquets, à chaque paquet on ajoute les entêtes propres au réseau (IP, UDP, RTP....) et pour finir il est envoyé sur le réseau.

A l'arrivée, les paquets transmis sont réassemblés en supprimant d'abord les entêtes. Les paquets de données ainsi obtenu est décompressé puis converti en signal analogique afin que l'utilisateur puisse écouter le message d'origine à travers un composent qui permis la transformation d'un signal électrique a un signal acoustique.

I.2.2. Architecture de transmission VoIP

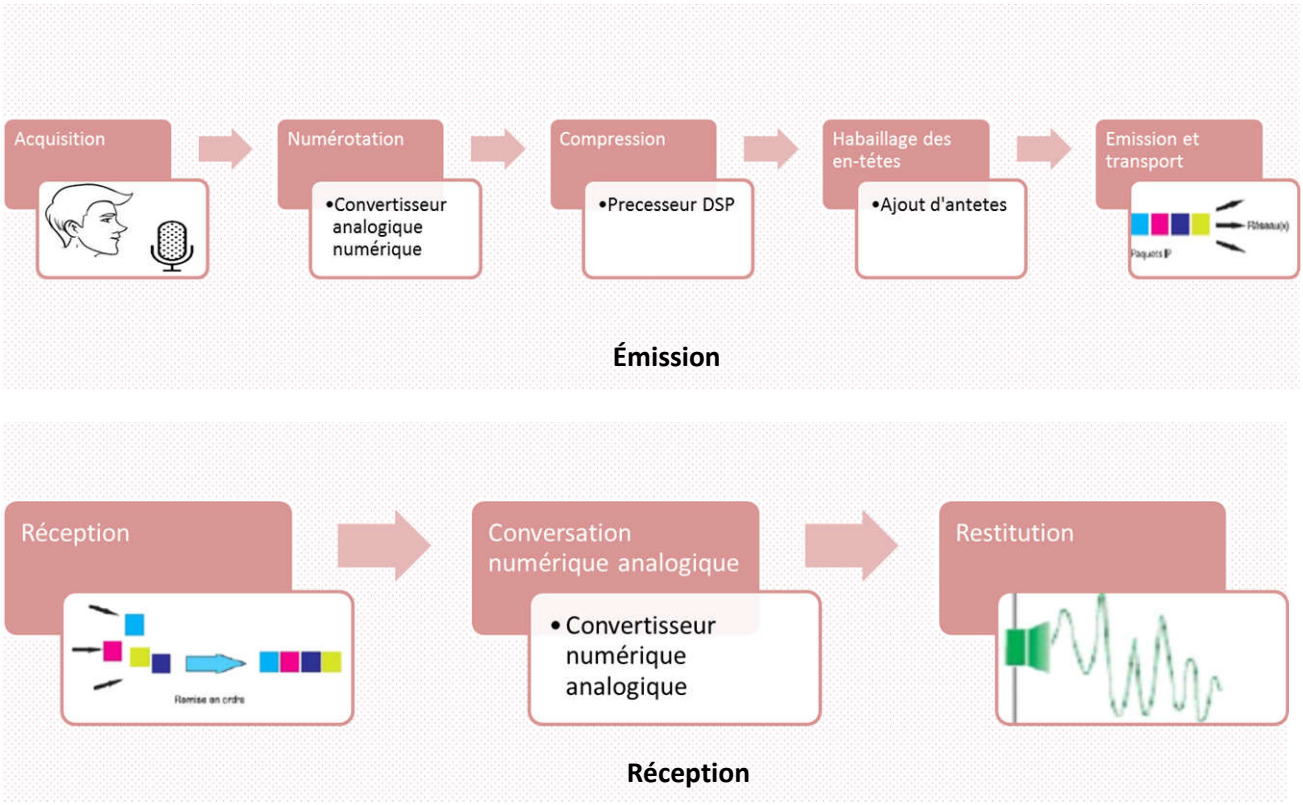


Figure 1-1 : Architecture de transmission VoIP

I.2.2.1. Acquisition du signal

La première étape consiste naturellement à capter la voix à l'aide d'un micro, qu'il s'agisse de celui d'un téléphone ou d'un micro casque.

Le microphone est un dispositif permettant de capter une onde sonore et de la « transformer » en signal électrique. Le micro contient une partie mobile, appelée membrane. Celle-ci est très légère et se trouve à l'avant du micro, afin de réagir à l'onde sonore. Quand celle-ci percute la membrane, cette dernière rentre en mouvement. En bougeant elle provoque une variation de courant, qui permet de former un signal électrique. L'énergie passe du domaine acoustique (onde) au domaine mécanique (mouvement de la membrane) puis au domaine électrique (signal).

I.2.2.2. Numérisation du signal

La conversion d'un signal analogique sous forme numérique implique une double approximation. D'une part, dans l'espace des temps, le signal fonction du temps s(t) est remplacé par sa valeur s(nT) à des instants multiples entiers d'une durée T; c'est l'opération

d'échantillonnage. D'autre part, dans l'espace des amplitudes, chaque valeur s (nT) est approchée par un multiple entier d'une quantité élémentaire q; c'est l'opération de quantification. La valeur approchée ainsi obtenue est ensuite associée à un nombre; c'est le codage, ce terme étant souvent utilisé pour désigner l'ensemble, c'est-à-dire le passage de la valeur s(nT) au nombre qui la représente.

- **L'opération d'échantillonnage :**

La première étape de numérisation consiste à échantillonner, c'est-à-dire prendre en compte seulement l'amplitude du signal à des intervalles de temps régulier T. La fréquence d'échantillonnage est donc de f=1/T. Pour être capable de reconstituer le signal d'origine, le théorème de l'échantillonnage fourni par la théorie du signal dit que la fréquence d'échantillonnage doit être supérieure ou égale à 2fmax, ainsi, pour un signal de la parole où l'information est contenue dans une bande de 4000 Hz (0-4000), un échantillonnage à 8000 Hz suffit (c'est-à-dire toutes les 125 µs).

Echantillonner à une fréquence plus faible conduit à un signal restitué de mauvaise qualité, et un échantillonnage plus élevé augmente le volume de données à transmettre sans augmentation significative de la qualité.

- **Quantification et codage** :

Il faut ensuite quantifier le signal échantillonné, c'est-à-dire lui associer une valeur parmi un ensemble fini de valeurs. La quantification peut se faire sur 256 niveaux.

Le codage du niveau, troisième étape, est finalement effectué sur 1 octet. La numérisation d'un signal vocal produit donc un flux régulier d'informations numériques de 1 octet toutes les 1/8 ms, soit un débit de 64 kbit/s. Cette technique, appelée MIC (Modulation par Impulsion et Codage), est utilisée dans le réseau téléphonique.

I.2.2.3. Compression

Le signal une fois numérisé peut être traité par un DSP (Digital Signal Processor) qui va le compresser, c'est-à-dire réduire la quantité d'informations (bits) nécessaire pour l'exprimer. Plusieurs normes de compression et décompression (Codecs) sont utilisées pour la voix (tableau 1-1). L'avantage de la compression est de réduire la bande passante nécessaire pour transmettre le signal.

Codec	débit (Kb/s)	taille de l'échantillon (octets)	durée de l'échantillonnage (ms)	Mean Opinion Score (MOS)
G711	(64 Kb/s)	80	10	4,1
G.729	(8 Kb/s)	10	10	3,92
G.723.1	(6,3 Kb/s)	24	30	3,9
G723.1	(5,3 Kb/s)	20	30	3,8
G.726	(32 Kb/s)	20	5	3,85
G.726	(24 Kb/s)	15	5	3,65
G.728	(16 Kb/s)	10	5	3.61
G.722	(64 Kb/s)	80	10	4.13
ILBC	(15.2 Kb/s)	38	20	4.1

Tableau 1-1 : Principaux codec utilisé dans la VOIP

I.2.2.4. Habillage des en-têtes

Les données «brutes» qui sortent du DSP doivent encore être enrichies en informations avant d'être converties en paquets de données à expédier sur le réseau. Trois «couches» superposées sont utilisées pour cet habillage :

- ✓ La couche RTP (Real Time Protocol) / RTCP (Real Time Control Protocol)

Pour pallier l'absence de fiabilité d'UDP, un formatage RTP est appliqué de surcroît aux paquets. Il consiste à ajouter des entêtes d'horodatage et de synchronisation pour s'assurer du réassemblage des paquets dans le bon ordre à la réception. RTP est souvent renforcé par RTCP qui comporte, en plus, des informations sur la qualité de la transmission l'identité des participants à la conversation.

- ✓ La couche UDP

La couche, UDP, consiste à formater très simplement les paquets. Si l'on restait à ce stade, leur transmission serait non fiable : UDP ne garantit ni le bon acheminement des paquets, ni leur ordre d'arrivée.

- ✓ La couche IP

La couche IP correspond à l'assemblage des données en paquets. Chaque paquet commence par un en-tête indiquant le type de trafic concerné, ici du trafic UDP.

Entête IP	Entête UDP	Entête RTP (12 octets)	Audio Data

Figure 1-2 : Format du paquet voix

I.2.2.5. Emission et transport

Une fois que la voix est transformée en paquets IP, ces paquets IP identifiés et numérotés peuvent transiter sur n'importe quel réseau IP (ADSL, Ethernet, Satellite, routeurs, switchs, PC, Wifi…etc.,), pour atteindre le point de réception sans qu'un chemin précis soit réservé pour leur transport. Ils vont transiter sur le réseau en fonction des ressources disponibles et arriver à destination dans un ordre indéterminé.

I.2.2.6. Réception

Lorsque les paquets arrivent à destination, il est essentiel de les replacer dans le bon ordre (rôle du protocole RTP) et assez rapidement. Faute de quoi une dégradation de la voix se fera sentir.

I.2.2.7. Conversion numérique analogique

La conversion numérique analogique est l'étape réciproque de l'étape 2, qui permet de transformer les données reçues sous forme de série discrète en un signal électrique «continu»

I.2.2.8. Restitution

Le signal électrique analogique reproduit par le convertisseur doit être transformé en un signal acoustique audible par l'oreille humaine. Cette transformation est assurée par un haut-parleur ou ca fonctionnement est opposite de celle du microphone. L'énergie passe du domaine électrique (signal) au domaine mécanique (mouvement de la membrane) puis au domaine acoustique (onde).

I.3. Protocoles

Un protocole est un langage commun utilisé par l'ensemble des acteurs de la communication pour échanger des données. Toutefois son rôle ne s'arrête pas là. Un protocole permet aussi d'initialiser la communication, d'échanger de données. Il faut distinguer plusieurs types de protocoles :

- ✓ Les protocoles de transport de la voix, transportent l'information sur un réseau IP. Ce type de protocoles est spécifique à la voix sur IP et aux applications nécessitant le transit de l'information en temps réel comme par exemple, la vidéo conférence.

- ✓ Les protocoles de signalisation, chargés de régir les communications, de déterminer les appelés, de signaler les appelants, de gérer les absences, les sonneries ; mais aussi de négocier quel codec pourra être utilisé.

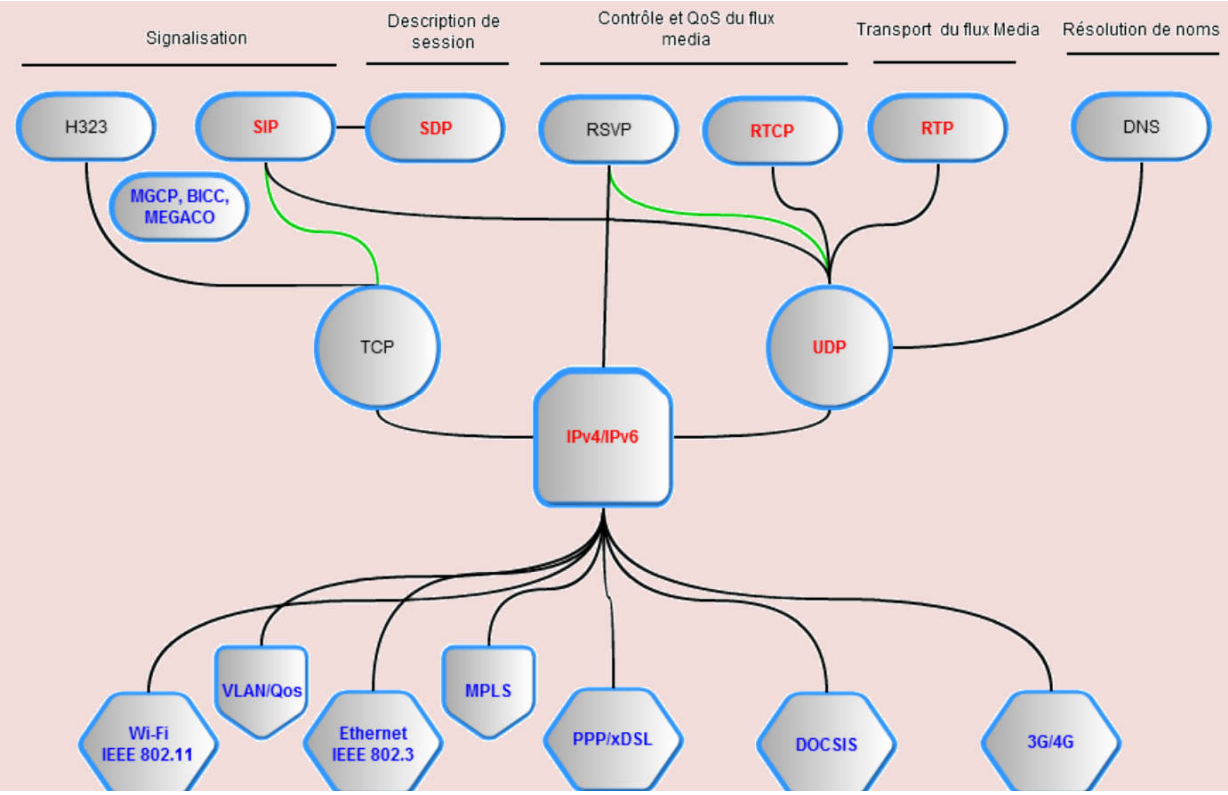


Figure 1- 3 : les protocoles VOIP

I.3.1. Le Protocole IP (Internet Protocol)

Internet Protocol, généralement abrégé IP, est un protocole de communication de réseau informatique, il correspond à un protocole de niveau 3 dans le modèle OSI et niveau 2 du modèle TCP/IP (figure : 1-3) permettant un service d'adressage unique pour l'ensemble des terminaux connectés.

Le protocole IP permet aux paquets de se déplacer sur le réseau Internet, indépendamment les uns des autres, sans liaison dédiée. Chacun d'entre eux, envoyé sur le réseau, se voit attribuer une adresse IP. Cette dernière est un en-tête accolé à chaque paquet et contenant

certaines informations, notamment, l'adresse destinataire, sa durée de vie, le type de service désiré… etc.

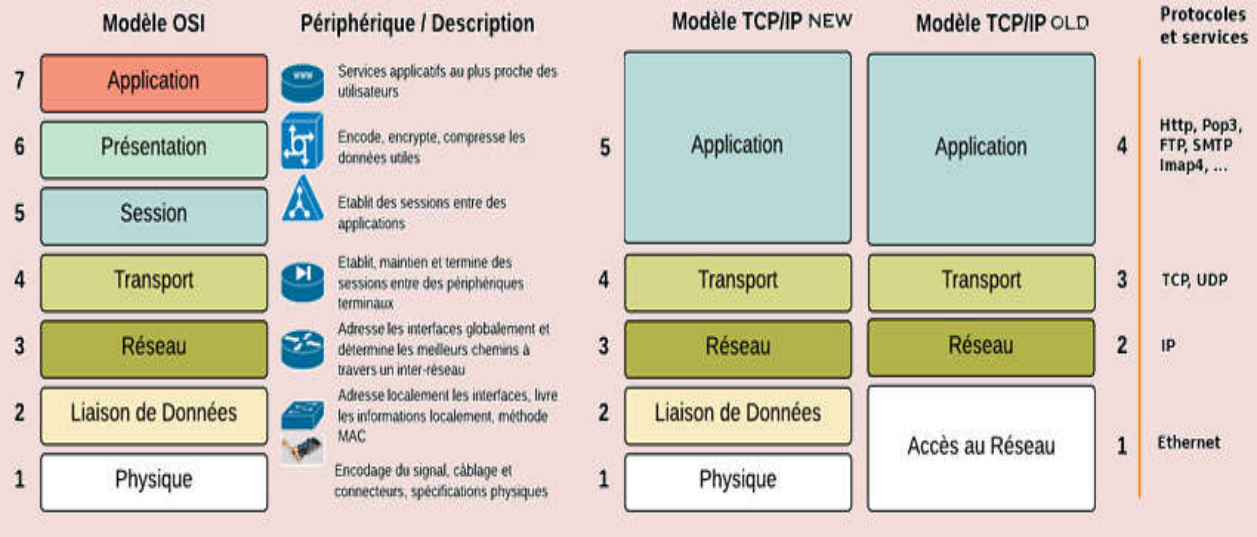


Figure1- 4 : Modèle OSI (Open Systems Interconnection) & du modèle TCP/IP

I.3.2. Les protocoles de transport

Pour transporter la voix ou la vidéo sur IP, le protocole IP (Internet Protocol) au niveau 3 et le protocole UDP (User Datagram Protocol) au niveau 4 sont utilisés. En effet, si TCP (Transmission Control Protocol) présente l'avantage de gérer un transfert fiable (renvoi des paquets IP en cas d'erreur), il est malheureusement incompatible avec un flux temps-réel dans la mesure où les mécanismes de TCP prévoient une réduction automatique du débit accordé à l'émetteur en cas de congestion du réseau et une remontée lente vers le débit nominal (ceci afin de protéger le réseau de soubresauts d'émetteurs qui chercheraient tous en même temps à tirer parti de la bande passante disponible).

Mais ces deux protocoles UDP et IP ne suffisent pas à assurer le transport de la voix. De fait, UDP est un protocole sans correction d'erreur, et à aucun moment l'arrivée des paquets dans leur ordre d'émission est assurée. Pour le transport de données temps réel telles que la voix ou la vidéo, il est nécessaire d'utiliser deux protocoles supplémentaires : RTP (Real Time transport Protocol) et RTCP (RTP Control Protocol).

I.3.2.1. UDP

Le protocole de datagramme utilisateur (UDP) est le protocole de transport sans confirmation. UDP est un protocole simple qui permet aux applications d'échanger des datagrammes sans accusé de réception ni remise garantie

UDP n'utilise ni fenêtrage, ni accusés de réception, il ne reséquence pas les messages, et ne met en place aucun contrôle de flux. Par conséquent, la fiabilité doit être assurée par les protocoles de couche application. Les messages UDP peuvent être perdus, dupliqués, remis hors séquence ou arriver trop tôt pour être traiter lors de leurs réception. UDP est un protocole particulièrement simple conçu pour des applications qui n'ont pas à assembler des séquences de segments. Son avantage est un temps d'exécution court qui permet de tenir compte des contraintes de temps réel ou de limitation d'espace mémoire sur un processeur, contraintes qui ne permettent pas l'implémentation de protocoles beaucoup plus lourds comme TCP.

Dans des applications temps-réel, UDP est le plus approprié, cependant il présente des faiblesses dues au manque de fiabilité. Des protocoles de transport et de contrôle temps-réel sont utilisés au-dessus du protocole UDP pour remédier à ses faiblesses et assurer sa fiabilité.

I.3.2.2. RTP et RTCP

RTP et RTCP sont des protocoles qui se situent au niveau de l'application et s'appuient sur le protocole de transport UDP. RTP et RTCP peuvent utiliser aussi bien le mode Unicast (point à point) que le mode Multicast (multipoint).

RTP et RTCP utilisent des ports différents. RTP utilise un numéro de port pair, et RTCP le numéro de port impair qui suit directement. Lorsqu'une session RTP est ouverte, alors une session RTCP est aussi ouverte de manière implicite.

Les numéros de port utilisés par RTP et RTCP sont compris entre 1025 et 65535. Les ports

RTP et RTCP par défaut sont respectivement 5004 et 5005.

- **RTP :**

Le but de RTP et de fournir un moyen uniforme de transmettre sur IP des données soumises à des contraintes de temps réel (audio, vidéo, etc.).

En générale RTP permet :

• d'identifier le type de l'information transportée,

• d'ajouter des marqueurs temporels permettant d'indiquer l'instant d'émission du paquet. L'application destinataire peut alors synchroniser les flux et mesurer les délais et la gigue.

• d'inclure des numéros de séquence à l'information transportée afin de détecter l'occurrence de paquets perdus et de délivrer les paquets en séquence à l'application destinataire.

De plus, RTP peut être véhiculé par des paquets multicast afin d'acheminer des conversations vers des destinataires multiples

L'en-tête d'un paquet RTP (RTP header) est obligatoirement constitué de 12 octets, éventuellement suivi d'une liste d'identificateurs de sources contributives CSRCs dans le cas d'un mixer.

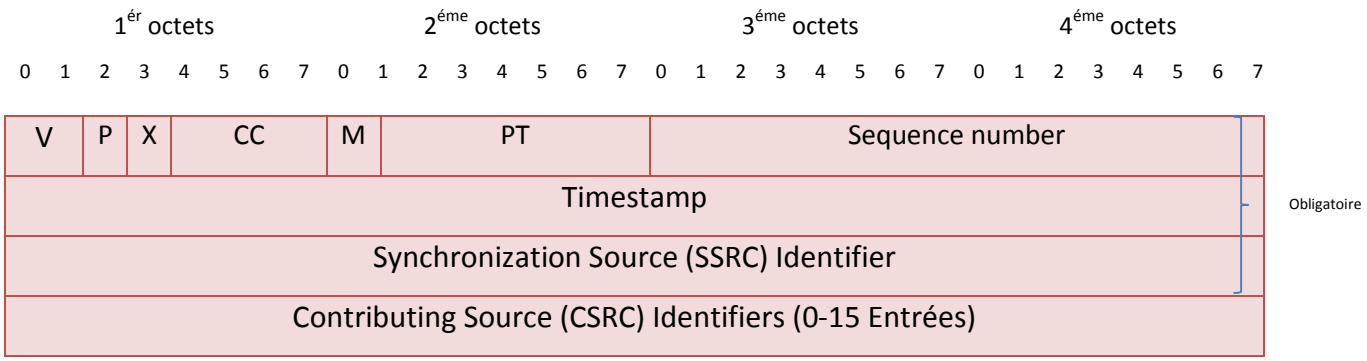


Figure 1-5 : En-tête RTP

- **V** : Ce champ, codé sur 2 bits, permet d'indiquer la version de RTP. Actuellement, V=3.
- **P** : Ce bit indique, s'il est à 1, que les données possèdent une partie de bourrage.
- **X** : Ce bit spécifie, s'il est à 1, que l'entête est suivi d'un entête supplémentaire.
- **CC** : Ce champ, codé sur 4 bits, représente le nombre de CSRC qui suit l'entête.
- **M** : Ce bit, lorsqu'il est à 1, définie que l'interprétation de la Marque est par un profil d'application.
- **PT** : Basé sur 7 bits, ce champ identifie le type du payload (audio, vidéo, image, texte, html, etc.).

- **Numéro de séquence** : Ce champ, d'une taille de 2 octets, représente le numéro d'ordre d'émission des paquets. Sa valeur initiale est aléatoire et il s'incrémente de 1 à chaque paquet envoyé, il peut servir à détecter des paquets perdus.
- **Timestamp**: Ce champ horodatage, de 4 octets, représente l'horloge système ou l'horloge d'échantillonnage de l'émetteur. Elle doit être monotone et linéaire pour assurer la synchronisation des flux.
- **SSRC** : Basé sur 4 octets, ce champ identifie de manière unique la source de synchronisation, sa valeur est choisie de manière aléatoire par l'application.
- **CSRC** : Ce champ, sur 4 octets, identifie les sources de contribution. La liste des participants ayant leur contribution (audio, vidéo) aux donnée du paquet.

- **RTCP :**

RTCP (Real Time Control Protocol, RFC 1889) est un protocole de contrôle utilisé conjointement avec RTP pour contrôler les flux de données et la gestion de la bande passante.

RTCP permet de contrôler le flux RTP, et de véhiculer périodiquement des informations de bout en bout pour renseigner sur la qualité de service de la session de chaque participant à la session. Des quantités telles que le délai, la gigue, les paquets reçus et perdus sont très important pour évaluer la qualité de service de toute transmission et réception temps réelles.

C'est le protocole sous-jacent (UDP par exemple) qui permet grâce à des numéros de ports différents et consécutifs (port pair pour RTP et port impair immédiatement supérieure pour RTCP) le multiplexage des paquets de données RTP et des paquets de contrôle RTCP.

Le protocole RTCP remplit trois fonctions :

- L'information sur la qualité de service : RTCP fournit, en rétroaction des informations sur la qualité de réception des données transmises dans les paquets RTP. Cette information est utilisée par la source émettrice pour adapter le type de codage au niveau des ressources disponibles.

- L'identification permanente : RTCP transporte un identificateur original de la source RTP c'est à dire la provenance du flux, appelé CNAME (Canonical name).

Cet identificateur permet une identification permanente de chacun des flux multimédia entrants.

- Le calibrage de la fréquence d'émission : La réception des feed-back et la connaissance du nom permanent servent à ajuster la fréquence d'envoi des paquets à la bande passante mise à la disponibilité de l'utilisateur situé à l'autre extrémité.

Il existe cinq types de paquets de contrôle (Tableau II). Chaque paquet commence par un en-tête fixe suivi d'éléments structurés qui peuvent être de longueur variable selon le type de paquet

SR	Rapport Émetteur
RR	Rapport Récepteur
SDES	Description de Source
BYE	Fin de session
APP	Nouveau champ

Tableau 1-2: Messages RTCP

- RR : contient le rapport de la qualité de la livraison des données des participants passifs (récepteur) incluant le nombre de paquets reçus, le nombre de paquets perdus, la gigue et l'horodatage qui permet le calcul du délai total de transmission entre les deux parties.

- SR :contient le rapport de la qualité de livraison des données des participants actifs (émetteurs). Il contient les champs du RR, et des informations sur l'émetteur, la synchronisation (pour synchroniser deux sources de données), un compteur cumulatif de paquets et le nombre d'octets envoyés.

- SDES : contient l'information concernant les émetteurs comme le nom canonique

(CNAME), et le nom d'usager (NAME), le numéro de téléphone (PHONE) et d'autres informations concernant les participants.

- BYE : indique la fin de la participation d'une des parties.

-APP : Dans le cas des applications spécifiques, les données peuvent être transmises dans des paquets spécifiques de type APP.

Le Figure 1-4 détaille l'en-tête ce message commun à tous les paquets RTCP :

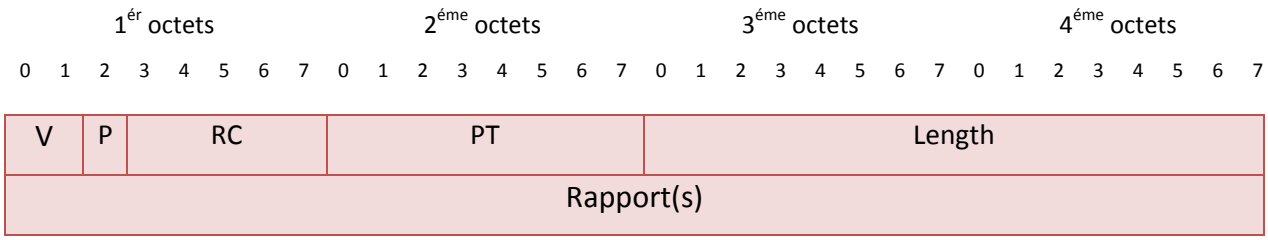


Figure 1-6 : En-tête des paquets RTCP

Avec :

- **V** : Ce champ, codé sur 2 bits, permet d'indiquer la version de CRTP. Actuellement, V=3.
- **P** : Ce bit indique, s'il est à 1, que les données possèdent une partie de bourrage dont la taille est indiquée dans le dernier octet.
- **RC :** Ce champ, codé sur 5 bits, Il contient le nombre de rapports contenus dans le paquet (Un paquet pour chaque source)**.**
- **PT** : Ce champ, d'une taille d'un octet; Il donne le type de rapport du paquet (PT=SR=200, PT=RR=201, PT=SDES=202, PT=BYE=203, PT=APP=204)
- **Length** : Ce champ, de 4 octets indique la longueur totale du paquet en mots de 32 bits (entête et bourrage compris).

I.3.3. Protocoles de signalisation

La signalisation désigne la transmission d'un ensemble de signaux et d'informations de contrôle échangés entre les intervenants d'une communication. Ces intervenants peuvent être des entités en bout de liaison (terminaux) ou des entités intermédiaires de contrôle et de gestion des communications. Leurs échanges permettent l'initiation, la négociation, l'établissement, le maintien et la fermeture de la connexion.

Dans le cas typique d'une application de téléphonie, lorsqu'une personne en appelle une autre, elle n'a initialement pas de « données » à lui transmettre, mais veut simplement être mise en relation avec son correspondant. Cette mise en relation nécessite d'abord de localiser l'appelé, puis de faire sonner son poste, afin de lui signaler l'appel. Pour la localisation comme pour l'avertissement d'appel, on parle de signalisation.

En VoIP, il existe plusieurs protocoles de signalisation tel que : SIP, H.323, MGCP, IAX, SCCP (propriétaire Cisco), MEGACO,BICC ,….

La figure1-6 illustre quelques exemples de messages de signalisation transportant des requêtes et réponses à caractère descriptif.

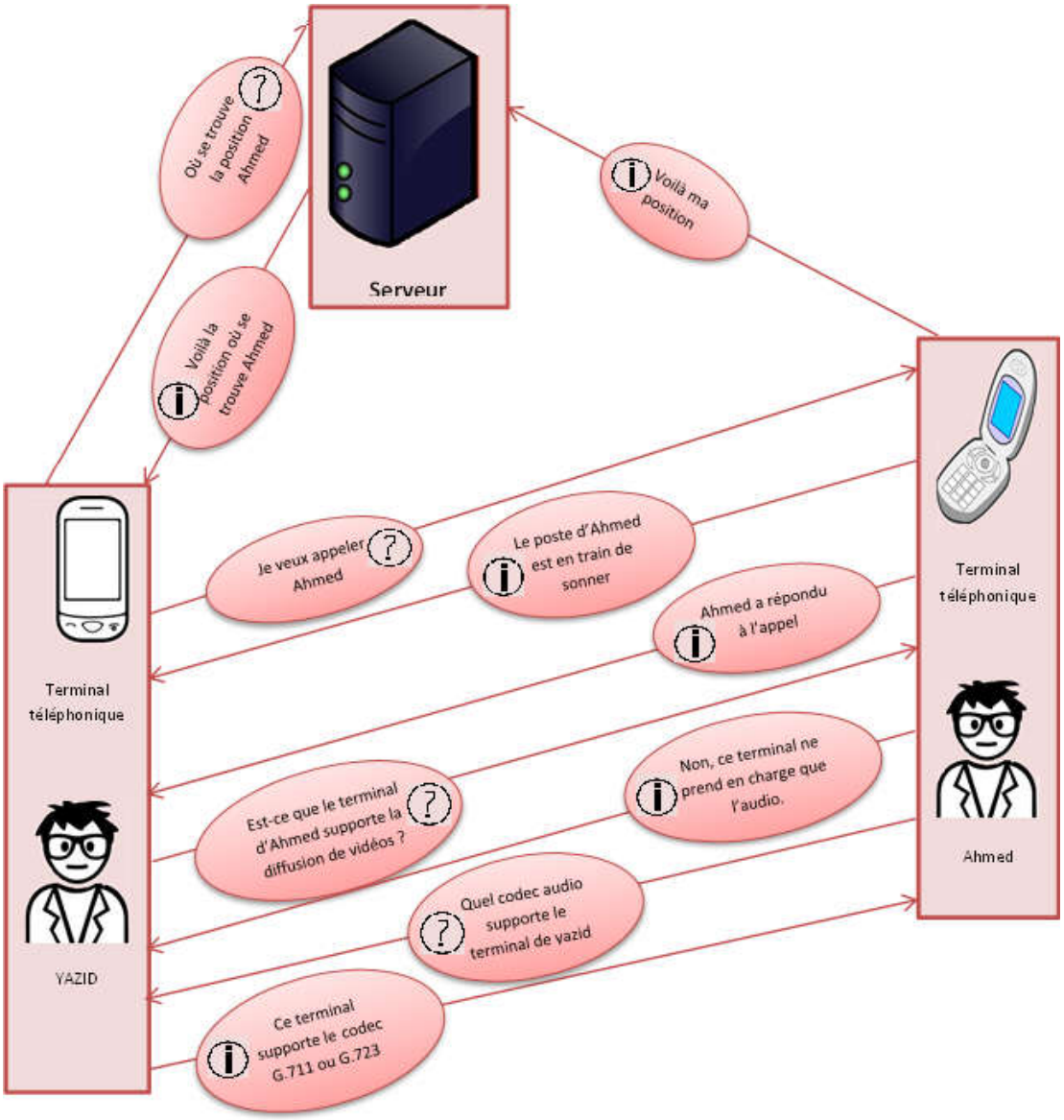


Figure 1-7 : Exemples de messages de signalisation transportant

I.3.3.1. Le protocole H.323

Développé par l'Union International des Télécommunications (ITU).Le H323 est un ensemble de protocoles utilisés pour la voix sur IP (figure : 1-7), qui peuvent être regroupés en trois catégories : signalisation, négociation et transport.

Lors d'un appel, il est utilisé en premier lieu le protocole H225 pour la signalisation de l'appel. Puis vient le H245 pour la négociation, et enfin le RTP pour le transport de la voix. Ces trois protocoles sont de la couche 5 et reposent sur le protocole TCP pour les deux premiers, et UDP pour le dernier.

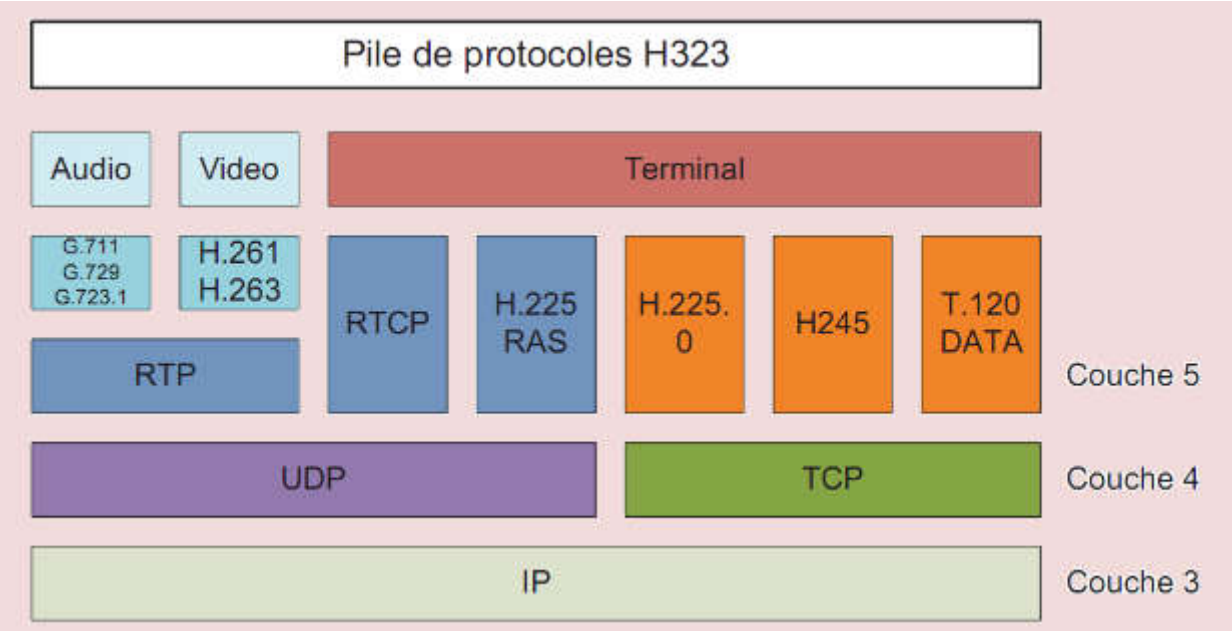


Figure 1-8 : Pile de protocole H323

- **Les éléments du H323**

Les entités H.323 sont regroupées dans des zones (Figure 1-8). Une zone est un ensemble de terminaux, passerelles (Gateway, GW) et ponts de conférence (Multipoint Control Unit, MCU) gérés par un même portier (Gatekeeper, GK). La zone comprend au moins un terminal et, éventuellement, des Gateways ou des MCUs. Une zone n'a qu'un seul Gatekeeper

- **Terminal**:

Un terminal est un endpoint qui permet des communications temps réels avec d'autres endpoints. Il s'agit d'un équipement utilisateur tel qu'un PC ou un téléphone IP qui supporte au moins un codec audio et éventuellement d'autres codecs audio et vidéo.

- **Passerelle**:

Est un endpoint du réseau qui assure en temps réel des communications bidirectionnelles entre des terminaux H.323 et d'autres terminaux (terminaux RTC, RNIS, GSM……).

- **Gardes-barrières**:

Le garde-barrière est un élément vital dans un système H323. Il joue le rôle de contrôleur pour tous les appels à l'intérieur de la zone H323 (une zone H323 est une agrégation de garde-barrière et de tous les autres éléments terminaux et MCU qui sont enregistré auprès de lui). Il fournit les services aux éléments qui sont enregistrés auprès de lui tel que la conversion des adresses, le contrôle d'admission, la gestion de la bande passante et la capacité de routage.

- **Unîtes de contrôle multipoint (MCU)**:

Un MCU est un terminal qui supporte des conférences entre 3 (ou plus) terminaux. Il peut s'agir d'un équipement indépendant, ou peut être intégré dans un Gateway, un Gatekeeper ou un terminal. Un MCU est composé de deux modules, le contrôleur multipoint (Multipoint Controller, MC) et le processeur multipoint (Multipoint Processor, MP).

Le module MC met en œuvre le contrôle et la signalisation pour le support de la conférence alors que le module MP reçoit les flux des terminaux, les traites, et les renvoie aux terminaux participant à la conférence.

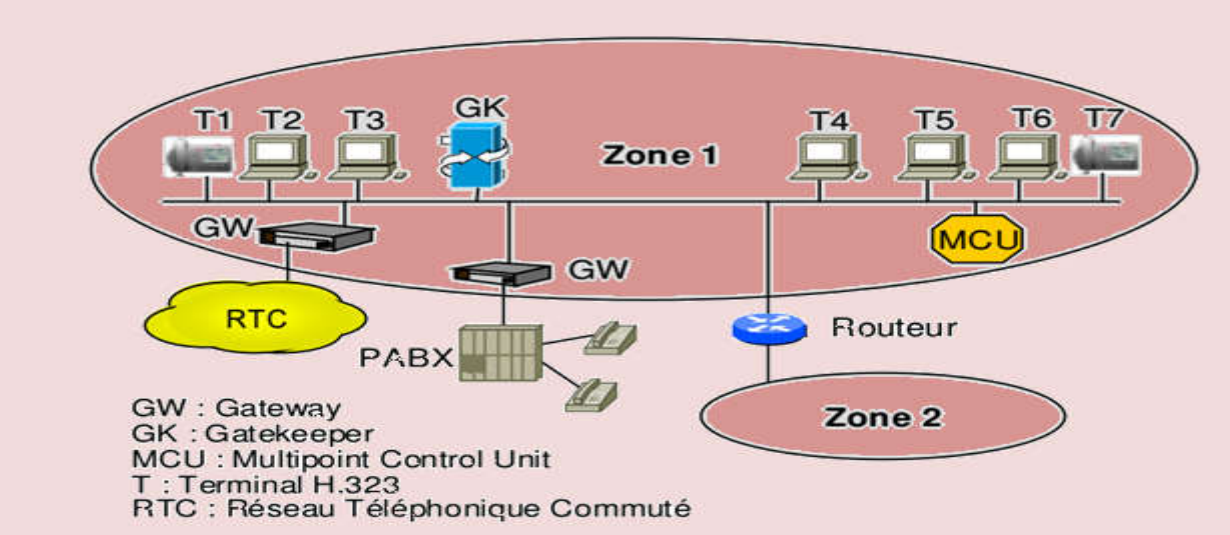


Figure 1-9 : Zone H 323

- **Protocoles et procédures**

La recommandation H323 inclus d'autres recommandations supplémentaires pour permettre les communications en temps réel.

Le tableau 1-3 résume quelques-unes d'entre elles.

Recommandation	Aperçu
Codecs Audio	
G.711	Encode le signal selon les lois A ou ì en 64 Kbit/s
G.723.1	Encode et compresse le signal vocal en 5.3 et 6.4 Kbit/s Encode et
G.729	compresse le signal vocal en 8 et 13 Kbits/s
Codecs Vidéo	
H.261	Encode et compresse la vidéo en 64 kb/s
H.263	Encode et compresse à faible taux de compression
Communication de données	
T.120	Protocole de données pour les conférences multimédia
Contrôle	
H.245	Protocole de contrôle pour les communications de données
H.225.RAS et H225.Q931	Protocole de signalisation des appels
Transport temps réel	
RTP / RTCP	Protocoles de transport et de contrôle en temps réel
Sécurité	
H.235	Sécurité, cryptage pour les terminaux des séries H32x
Services supplémentaires	
H.450.1	Protocole pour les services supplémentaires
H.450.2 et 450.3	Transfert d'appels et autres services supplémentaires

Tableau 1-3 : Recommandations du H323

- **Mode de signalisation** (Figure 1-9)

Le mode de signalisation détermine quels messages de signalisation sont échangés entre les endpoints via le Gatekeeper, et lesquels sont directement échangés.

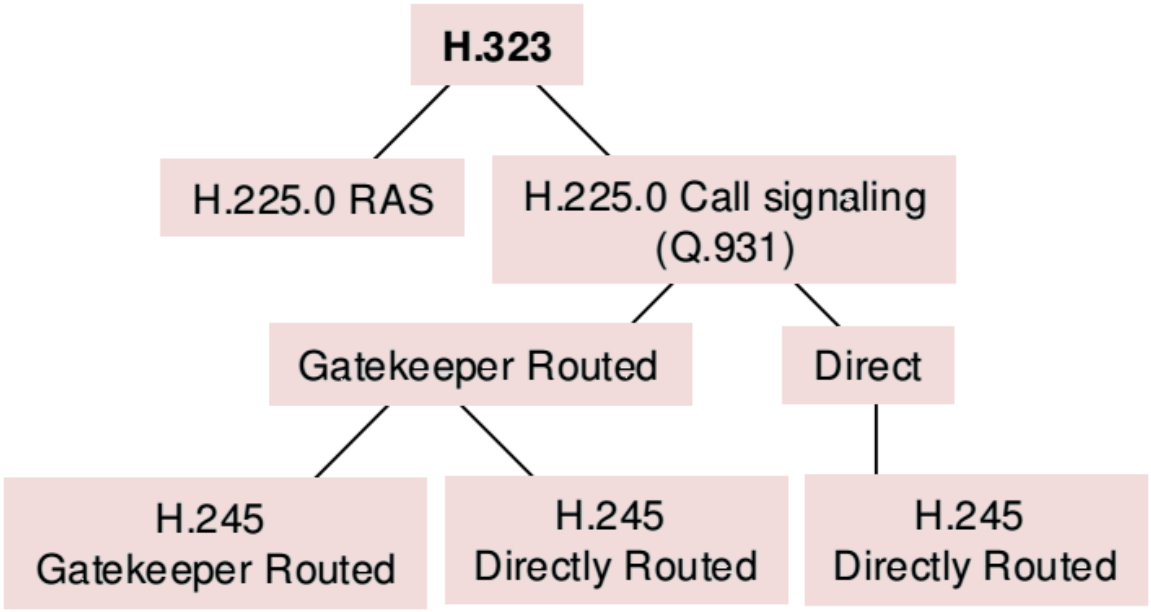


Figure1- 10 : Modes de signalisation H.323

- **La signalisation RAS :**

Les messages H.225.0 RAS (registration, admission, status) définissent une communication entre les terminaux et un garde-barrière. H.225.0 RAS s'occupe de la communication entre le garde-barrière et les différents terminaux. Elle gère les opérations suivantes : l'inscription, le contrôle d'admission, la gestion de la bande passante. Un canal de signalisation est utilisé afin de transporter les différents messages RAS

-**La signalisation d'appel Q.931 (Call Signaling) :** La signalisation des appels est importante pour établir et libéré une connexion entre deux entités ; est échangée soit directement entre les endpoints (Direct Routed), soit entre endpoints à travers le Gatekeeper (Gatekeeper Routed).

- **La signalisation H.245 :**

La flexibilité de H.323 nécessite que les différends terminaux négocient les capacités avant que les liens de la communication audio, vidéo et/ou données ne soit établit. H.245 utilise les messages de contrôle et de commandes qui sont échangés durant l'appel. Ces messages permettent:

• L'échange de capacités multimédia (audio, vidéo), afin d'assurer une transmission selon un mode audio, vidéo particulier.

• La détermination du terminal maître et du terminal asservi afin d'éviter tout conflit dans le contrôle d'une conférence,

• L'établissement et la libération de canaux logiques permettant le transfert de données multimédia,

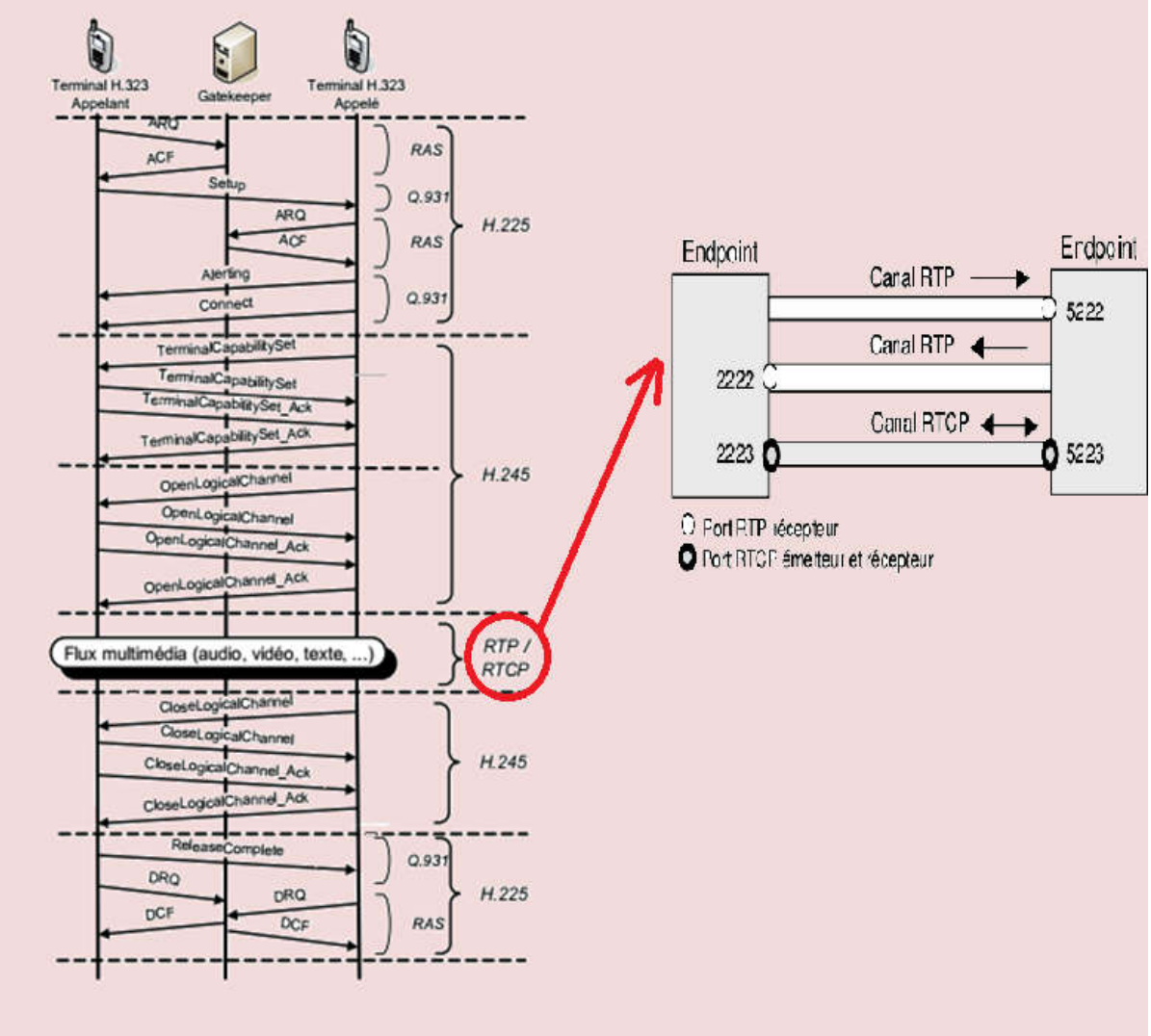


Figure 1-11 : Exemple de signalisation d'appel suivant le mode Direct Routed

I.3.3.2. Le protocole SIP (Session Initiation Protocol)

Session Initiation Protocol (dont l'abréviation est SIP) est un protocole normalisé et standardisé par le RFC 3261. Il a été conçu pour établir, modifier et terminer des sessions multimédia. Il hérite de certaines fonctionnalités des protocoles http (Hyper Text Transport Protocol) utilisé pour naviguer sur le WEB, et SMTP (Simple Mail Transport Protocol) utilisé pour transmettre des messages électroniques (E-mails). SIP s'appuie sur un modèle transactionnel client/serveur comme HTTP. L'adressage utilise le concept d'URL SIP (Uniform Resource Locator) qui ressemble à une adresse E-mail.

Chaque participant dans un réseau SIP est donc adressable par une URL SIP. Par ailleurs, les requêtes SIP sont acquittées par des réponses identifiées par un code numérique.

D'ailleurs, la plupart des codes de réponses SIP ont été empruntés au protocole HTTP. Par exemple, lorsque le destinataire n'est pas localisé, un code de réponse « 404 Not Found » est retourné. Une requête SIP est constituée de headers comme une commande SMTP.

Enfin SIP comme SMTP est un protocole textuel.

- **Architecture de SIP**

Contrairement à H.323, largement fondé sur une architecture physique, le protocole SIP s'appuie sur une architecture purement logicielle.

L'architecture de SIP s'articule principalement autour des cinq entités suivantes :

• terminal d'utilisateur ;

• serveur d'enregistrement ;

• serveur de localisation ;

• serveur de redirection ;

• serveur proxy.

La figure 1-11 illustre de façon générique les communications entre ces éléments. Un seul terminal étant présent sur cette figure, aucune communication n'est possible. Nous nous intéressons en fait ici aux seuls échanges entre le terminal et les services que ce dernier est susceptible d'utiliser lors de ses communications.

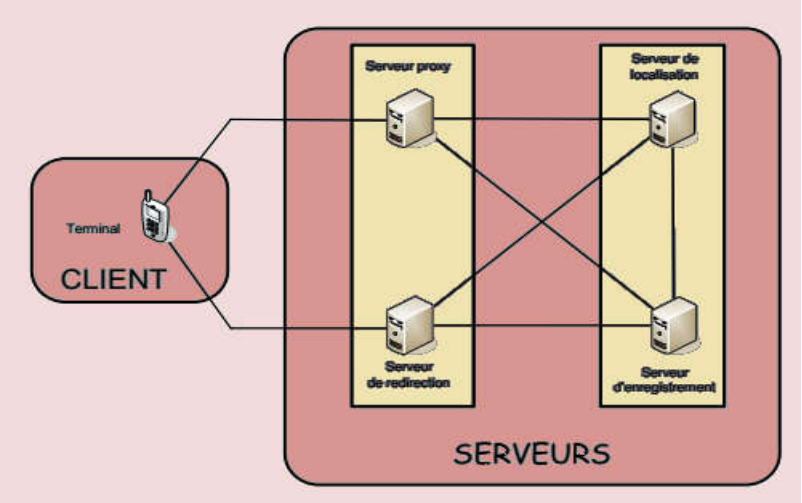


Figure 1-12 : Architecture de SIP

- **Terminal :**

Le terminal est l'élément dont dispose l'utilisateur pour appeler et être appelé. Il doit donc permettre de composer des numéros de téléphone. Il peut se présenter sous la forme d'un composant matériel (un téléphone) ou d'un composant logiciel (Softphone).

Le terminal est appelé UA (User Agent). Il est constitué de deux sous-entités

• Une partie cliente, appelée UAC (User Agent Client), chargée d'émettre les requêtes. C'est l'UAC qui initie un appel.

• Une partie serveur, appelée UAS (User Agent Server), qui est en écoute, reçoit et traite les requêtes. C'est l'UAS qui répond à un appel.

- **Serveur d'enregistrement :**

Deux terminaux peuvent communiquer entre eux sans passer par un serveur d'enregistrement, à la condition que l'appelant connaisse l'adresse IP de l'appelé. Cette contrainte est fastidieuse, car un utilisateur peut être mobile et donc ne pas avoir d'adresse IP fixe, par exemple s'il se déplace avec son terminal ou s'il se connecte avec la même identité à son travail et à son domicile. En outre, l'adresse IP peut être fournie de manière dynamique par un serveur DHCP.

Le serveur d'enregistrement (Registrar Server) offre un moyen de localiser un correspondant avec souplesse, tout en gérant la mobilité de l'utilisateur. Il peut en outre supporter l'authentification des abonnés.

Dans la pratique, lors de l'activation d'un terminal dans un réseau, la première action initiée par celui-ci consiste à transmettre une requête d'enregistrement auprès du serveur d'enregistrement afin de lui indiquer sa présence et sa position de localisation courante dans le réseau. C'est la requête REGISTER, que nous détaillons plus loin, que l'utilisateur envoie à destination du serveur d'enregistrement. Celui-ci sauvegarde cette position en l'enregistrant auprès du serveur de localisation.

L'enregistrement d'un utilisateur est constitué par l'association de son identifiant et de son adresse IP. Un utilisateur peut s'enregistrer sur plusieurs serveurs d'enregistrement en même

temps. Dans ce cas, il est joignable simultanément sur l'ensemble des positions qu'il a renseignées.

- Serveur de localisation

Le serveur de localisation (Location Server) joue un rôle complémentaire par rapport au serveur d'enregistrement en permettant la localisation de l'abonné.

Ce serveur contient la base de données de l'ensemble des abonnés qu'il gère. Cette base est renseignée par le serveur d'enregistrement. Chaque fois qu'un utilisateur s'enregistre auprès du serveur d'enregistrement, ce dernier en informe le serveur de localisation.

- Serveur de redirection

Le serveur de redirection (Redirect Server) agit comme un intermédiaire entre le terminal client et le serveur de localisation. Il est sollicité par le terminal client pour contacter le serveur de localisation afin de déterminer la position courante d'un utilisateur.

L'appelant envoie une requête de localisation d'un correspondant (il s'agit en réalité d'un message d'invitation, qui est interprété comme une requête de localisation) au serveur de redirection. Celui-ci joint le serveur de localisation afin d'effectuer la requête de localisation du correspondant à joindre. Le serveur de localisation répond au serveur de redirection, lequel informe l'appelant en lui fournissant la localisation trouvée. Ainsi, l'utilisateur n'a pas besoin de connaître l'adresse du serveur de localisation.

- Serveur proxy

Le serveur proxy (parfois appelé serveur mandataire) permet d'initier une communication à la place de l'appelant. Il joue le rôle d'intermédiaire entre les terminaux des interlocuteurs et agit pour le compte de ces derniers.

Le serveur proxy remplit les différentes fonctions suivantes :

• localiser un correspondant ;

• réaliser éventuellement certains traitements sur les requêtes ;

• initier, maintenir et terminer une session vers un correspondant.

Lorsqu'un utilisateur demande à un serveur proxy de localiser un correspondant, ce dernier effectue la recherche, mais, au lieu de retourner le résultat au demandeur (comme le ferait un serveur de redirection), il utilise cette réponse pour effectuer lui-même l'initialisation de la communication en invitant le correspondant à ouvrir une session.

Bien que fournissant le même type de service de localisation qu'un serveur de redirection, un serveur proxy va donc plus loin que la simple localisation en initiant la mise en relation des correspondants de façon transparente pour le client. Il peut acheminer tous les messages de signalisation des terminaux, de l'initialisation de la communication à sa terminaison, en passant par sa modification.

- **Fonctionnement du protocole SIP**

Le RFC 3261 définit six requêtes ou méthodes SIP (tableau 1-4)

Méthode	Utilisation
INVITE	Utilisée afin d'établir une session entre UAs, contient les informations sur l'appelante et l'appelé et sur le type de flux qui seront échangés
ACK	Confirme que le client a reçu une réponse positive à une requête INVITE
BYE	Permet la libération d'une session préalablement établie. Un message BYE peut être émis par l'appelant ou l'appelé
REGISTER	Est utilisée par un UA afin d'indiquer au Registrar la correspondance entre son adresse SIP et son adresse IP
CANCAL	Cette méthode annule une requête dont la réponse n'est pas encore parvenue au demandeur. Elle ne permet pas d'interrompre une session, mais indique que la réponse n'est plus attendue et qu'il n'est donc pas nécessaire de traiter la requête.
OPTIONS	Est utilisée afin d'interroger les capacités et l'état d'un User agent ou d'un serveur

Tableau 1- 4 : les requêtes SIP

Après avoir reçu et interprété une requête SIP, le destinataire de cette requête retourne une réponse SIP. Il existe six classes de réponses (tableau 1-5)

Classe	Définition de la famille de réponse	Principales réponses
1xx	Réponse intermédiaire d'information (traitement en cours)	100 Trying ,180 Ringing
2xx	Succès	200 OK
3xx	Redirection	301 Moved permanently 302 Moved temporarily
4xx	Erreur client	400 Bad Request 401 Unauthorized
5xx	Erreur serveur	500 Server Internal Error 501 Not Implemented
6xx	Echec global du traitement	600 Buzy Everywhere 603 Decline

Tableau 1- 5 : Les classes de réponses SIP

- **Exemple d'établissement et libération de session SIP :**

Dans l'exemple suivant (figure 1-12), l'appelant a pour URL SIP sip: mary.taylor@tn.com, alors que celle de l'appelé est sip:mark.rich@tn.com

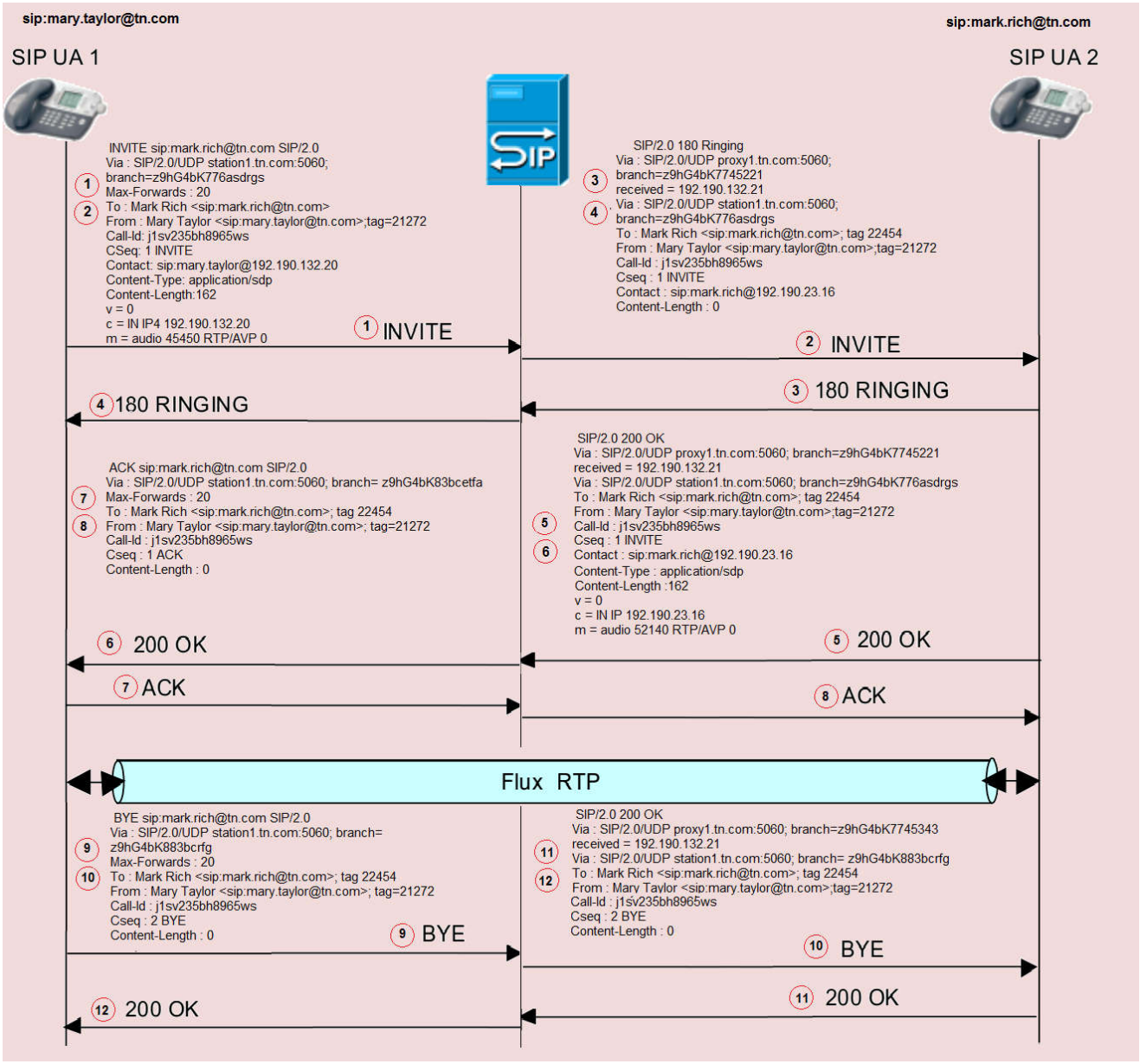


Figure 1-13 : Etablissement d'une session audio avec le protocole SIP

I.4. Conclusion

À travers notre étude de ce chapitre, nous avons remarqué qu'il y a un développement continu de cette technologie pour répondre aux besoins des utilisateurs, en termes de coûts, de qualité et de services supplémentaires.

De nombreuses entreprises et organisations internationales ont développé des protocoles conçus spécifiquement pour s'adapter à la nature de ce réseau, y compris des protocoles propriétaires pour un constricteur spécifique tel que SCCP pour Cisco, et d'autres protocoles ouverts tels que SIP et H323.

Cependant, aucun modèle uniforme de cette technologie n'a encore été atteint, mais cela ne signifie pas qu'il est risqué de parier sur ces critères spécifiques, en raison de la mesure dans laquelle ils sont acceptés par l'ensemble de la communauté de la téléphonie.

Dans le chapitre qui suit, nous allons présenter en détail les étapes nécessaires à la création d’un serveur pour la VoIP en utilisant Asterisk l’un des plus populaires open sources IPBX.

CHAPITRE II : installation et configuration d'une solution de VoIP basée sur l'outil Asterisk

II.1. Introduction

Asterisk est un projet lancé en 1999 par Mark Spencer. Son but était de fournir à L'Unix un commutateur téléphonique complet et gratuit (publié sous licence GPL).

Le nom **Asterisk** fait référence au symbole "*****" qui signifie "**wildcard**" en ligne de commande **Unix** et **DOS**. Cette sélection est faite parce que l'astérisque est conçu pour fournir une grande flexibilité dans les réseaux de voix.

Asterisk comprend un nombre très élevé de fonctions, tel que les appels téléphoniques, la messagerie vocale, les files d'attentes, les conférences, etc. Il implémente plusieurs protocoles H.320, H.323, SIP et IAX.

Durant ce chapitre, on montrera les étapes d'installation et de configuration d'Asterisk sous le système d'exploitation Linux, ainsi que l'installation et la configuration des téléphones VoIP softphone, freeware tel-que Zoiper et 3CX.

II.2. Architecture interne d'Asterisk

Asterisk se compose d'un noyau de commutation central, et de quatre API (Application Programming Interface) de chargement modulaire d'applications téléphoniques, d'interfaces matérielles, de traitement de format de fichier et de codecs. Il offre une commutation transparente entre toutes les interfaces prises en charge, permettant à cette commutation d'interconnecter une variété de systèmes téléphoniques en un seul réseau commuté.

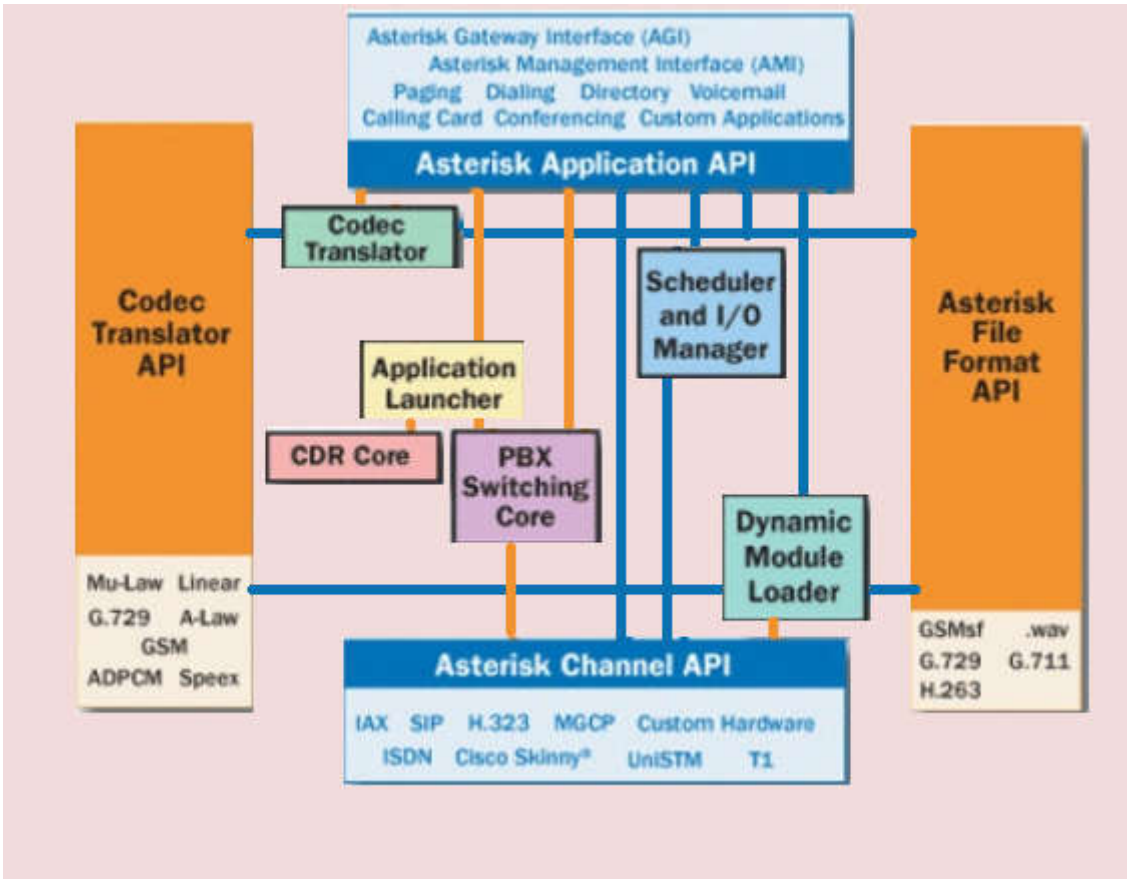


Figure 2-1: architecture interne d'Asterisk

Principales fonctions : Les principale-s tâches d'Asterisk sont:

-Noyau central de commutation:

- **PBX Switching Core** : Un système de commutation PBX, qui connecte d'une manuaire automatique les appels entre différents utilisateurs .Le noyau du commutateur connecte d'une manière transparente les appels à diverses interfaces matérielles et logicielles.
- **Application Launcher:** lance les applications qui fournissent des services aux utilisateurs, telles que la messagerie vocale, la lecture des messages et le répertoire téléphonique (annuaire).
- **Codec Translator :** Utilise des modules de codec pour le codage et le décodage de divers formats de compression audio utilisés dans l'industrie de la téléphonie. Un certain nombre de codecs sont disponibles pour répondre à différents besoins et atteindre le meilleur équilibre entre la qualité sonore et l'utilisation de la bande passante.

- **Scheduler & I/O Manager:** gérer la planification des tâches de bas niveau et la gestion du système pour optimiser les performances dans toutes les conditions de charge.
- **Dynamic Module Loader :** charge les pilotes (lors de la 1ère exécution d'Asterisk, configure les pilotes et crée le lien avec les API appropriées). Une fois les pilotes chargés, les appels sont acceptés et acheminés en faisant sonner les téléphones.

- Les APIs (Application Programming Interface)

Le dépouillement de matériel et de protocole passe par l'utilisation de 4 API :

- **Asterisk Application API :** permet de lancer différentes unités de tâches pour exécuter différentes fonctions. communications, audioconférence, pagination, la liste des répertoires, la messagerie vocale, transmission de données intégrée, et toute autre tâche qu'un système PBX exécute actuellement ou dans l'avenir, sont mises en œuvre par ces distincts unités.
- **Asterisk Translator API**: charge les modules de codec pour prendre en charge les différents formats de codage et de décodage audio tels que GSM, Mu-Law, A-Law et MP3.
- **Asterisk Channel API :** Cette API gère le type de connexion à laquelle l'appelant se connecte, que ce soit une communication VoIP, RNIS, PRI, ou toute autre technique. Les modules dynamiques sont chargés pour gérer les détails de la couche inférieure de ces connexions.
- **Asterisk File Format API :** Permet de lire et d'écrire de divers formats de fichiers pour stocker des données dans le système de fichiers.

II.3. Mise en place d'un PABX-IP avec Asterisk

II.3.1. Architecture du réseau

La figure 2.2 montre l'architecture adoptée au cours de la configuration de la solution de VoIP à base d'Asterisk

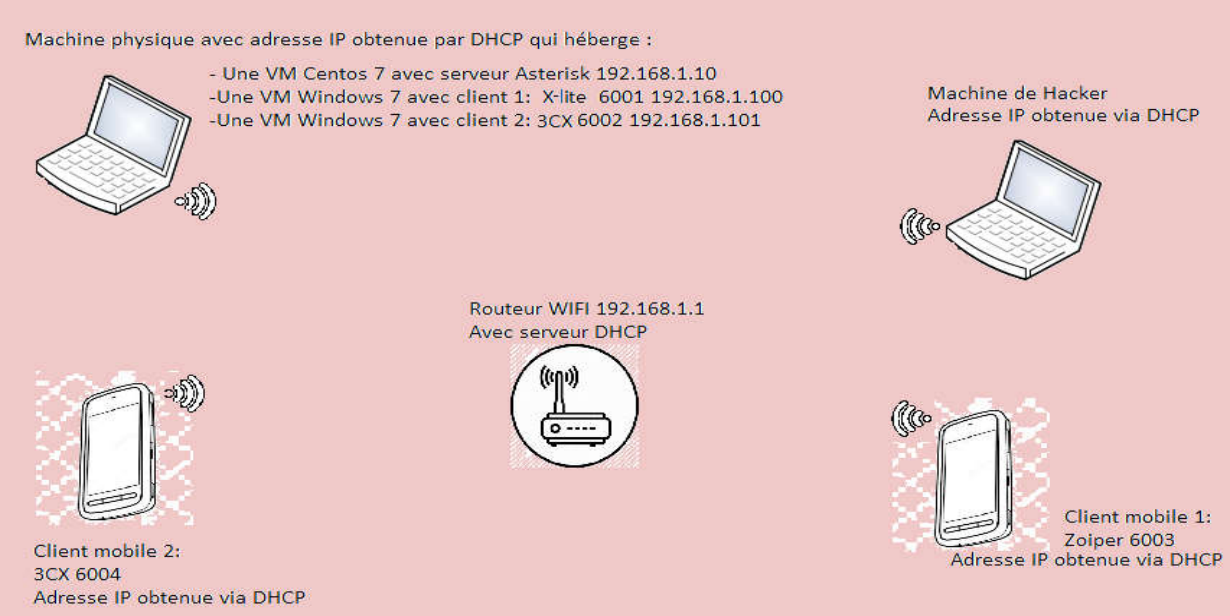


Figure 14: Architecture du réseau

Le serveur qui héberge la plate-forme Asterisk est un serveur GNU/Linux fonctionnant avec un

Système d'exploitation CentOS release 7.4.1708 (Core) installer dans une machine VM.

Les étapes pour l'installer sont les suivantes:

II.3.2. Installation du prérequis

-Mettre à jour notre distribution Linux

```
[root@localhost ymezhoudi]#  yum update
```

Mise à jour du système

- Procédons ensuite à l'installation des dépendances nécessaires à la compilation d'Asterisk.

```
[root@localhost ymezhoudi]#  yum install gcc gcc-c++ php-xml php php-mysql php-pear
php-mbstring mariadb-devel mariadb-server mariadb sqlite-devel lynx bison gmime-devel
psmisc tftp-server httpd make ncurses-devel libtermcap-devel sendmail sendmail-cf
caching-nameserver sox newt-devel libxml2-devel libtiff-devel audiofile-devel gtk2-devel
uuid-devel libtool libuuid-devel subversion kernel-devel kernel-devel-$(uname -r) git
subversion kernel-devel php-process crontabs cronie cronie-anacron wget vim
```

Installation des dépendances d'Asterisk

-Pré-configurations de MariaDB.

Ensuite nous allons configurer MariaDB ; par défaut, MariaDB permet une connexion à la base de données sans mot de passe. Nous allons donc activer le service MariaDB puis ajouter un mot de passe de connexion.

[root@localhost ymezhoudi]# systemctl enable mariadb
Permettre à MariaDB de se lancer

[root@localhost ymezhoudi]# systemctl start mariadb
Lancer mariadb manuellement

[root@localhost ymezhoudi]# systemctl status mariadb
Connaitre l'état du service

[root@localhost ymezhoudi]# mysql_secure_installation
Définir un mot de passe de root, supprimer les utilisateurs anonymes et interdire la connexion des utilisateurs distants

Installation de JANSSON :

Ensuite on télécharge la bibliothèque Jansson .Jansson est une bibliothèque C pour le codage, le décodage et la manipulation des données JSON

[root@localhost ymezhoudi]# wget http://www.digip.org/jansson/releases/jansson-2.11.tar.gz
Téléchargement de la bibliothèque jansson

[root@localhost ymezhoudi]# tar -zxvf jansson-2.11.tar.gz
Extraction des fichiers

[root@localhost ymezhoudi]# cd jansson-2.11 **[root@localhost jansson-2.11]# ./configure --prefix=/usr**
lancement du script de configuration

```
[root@localhost jansson-2.11 ]#  make clean
[root@localhost jansson-2.11 ]# make
[root@localhost jansson-2.11 ]# make install
[root@localhost jansson-2.11 ]# ldconfig
```

compilation et installation de jansson

II.3.3. Installation d'Asterisk 14

Les prérequis étant installés, nous allons passer à l'installation d'Asterisk

```
[root@localhost ymezhoudi]# wget
http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-14-current.tar.gz
```

Récupération de l'archive d'Asterisk

```
[root@localhost ymezhoudi]# tar -zxvf asterisk-14-current.tar.gz
```

Extraction des fichiers

```
[root@localhost ymezhoudi]# cd asterisk-14.7.6
[root@localhost asterisk-14.7.6]# ./configure--libdir=/usr/lib64
```

lancement du script de configuration

A la fin du script de configuration, on voit apparaitre le logo d'Asterisk, nous allons maintenant sélectionner les modules d'Asterisk. Il faut lancer le menu de sélection

```
[root@localhost asterisk-14.7.6]#  make menuselect
```

Sélection des modules

- Aller dans Core sound packages, décocher core-sound-en-gsm et cocher core-sounds-fr-ulaw
- Aller dans Music on hold file packages cocher moh-opsound-ulaw et décocher les autres
- Aller dans Extra-sounds packages, cocher extra-sounds-fr-ulaw
- Save & Exit

```
[root@localhost asterisk-14.7.6]#contrib/scripts/get_mp3_source.sh
```

lancement le script de chargement de la libraire de décodage mp3

On peut maintenant compiler et installer Asterisk

```
[root@localhost asterisk-14.7.6]# make
[root@localhost asterisk-14.7.6]#  make install
```

compilation et installation d'Asterisk

Asterisk est maintenant installé, cependant le dossier des fichiers de configurations est vide. Nous allons installer des exemplaires de fichiers de configuration

```
[root@localhost asterisk-14.7.6]# make samples
[root@localhost asterisk-14.7.6]# make config
```

installation des exemplaires de fichiers de configuration

Maintenant il faut installer Asterisk Database, on se connecte à la base de données en tant que root.

```
[root@localhost asterisk-14.7.6]# mysql -u root -p
```

Se connecter à MariaDB

On crée ensuite l'utilisateur 'asterisk' avec un mot de passe.

```
MariaDB [(none)]> create user 'asterisk'@'localhost' identified by'abc@123';
```

Création de l'utilisateur de la base de données nommé asterisk

```
MariaDB [(none)]> create database asterisk;
MariaDB [(none)]> create database cdrdb;
```

Création les bases de données asterisk et cdrdb

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON asterisk.* TO asterisk@localhost IDENTIFIED BY
'abc@123';
MariaDB [(none)]>  GRANT ALL PRIVILEGES ON cdrdb.* TO asterisk@localhost IDENTIFIED
BY 'abc@123';
```

Donner l'accès de la base 'asterisk ' et la base 'cdrdb' à l'utilisateur ' asterisk'

```
MariaDB [(none)]> FLUSH PRIVILEGES;
```

Appliqué les modifications et recharger les privilèges utilisateurs.

On peut maintenant lancer asterisk et se connecter à la console

```
[root@localhost ymezhoudi]# asterisk -r
[root@localhost ymezhoudi]# asterisk -cvvvv
```

Lancement d'asterisk

Asterisk peut être lancé de 2 manières:

- En mode serveur (usage normal, à l'écoute des requêtes des clients).

Syntaxe: asterisk -cvvvv

-Chaque 'v' demande un degré de verbosité, de verbose à very very verbose.

- 'c' nous donnera accès à une invite de commande (nommée CLI pour commande line interface) qui permettra de dialoguer avec le serveur une fois celui-ci lancé.

- En mode client (en accédant à une console permettant de dialoguer avec le serveur).

Syntaxe: asterisk -r

II.4. Configuration d'Asterisk et création des comptes utilisateurs

II.4.1. Configuration des comptes user

La création des utilisateurs se fait dans le fichier **sip.conf** ,

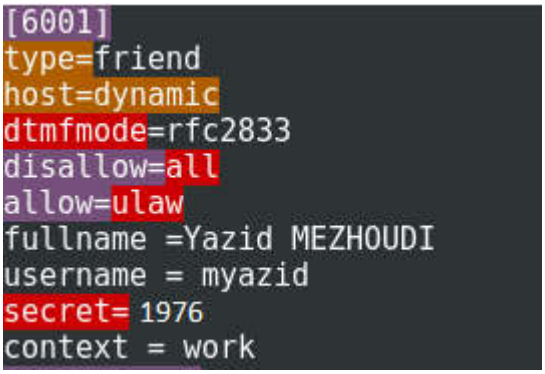
```
[root@localhost ymezhoudi]#  vim /etc/asterisk/sip.conf
```

Ouverture du fichier Sip.conf dans la console pour modifications

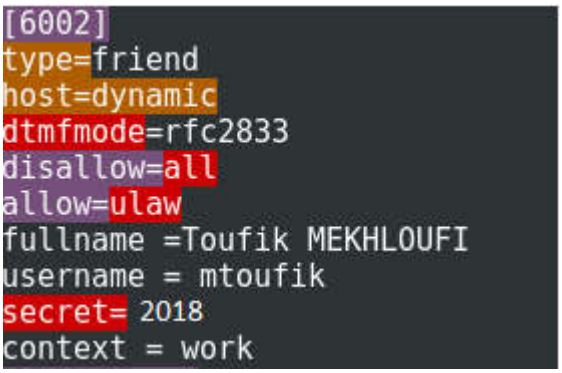
Voici un exemple pour des modifications apportés au fichier sip.conf pour la création de deux utilisateurs Yazid MEZHOUDI et Toufik MEKHLOUFI avec comme numéro SIP le 6001 et le 6002.

```
[general]
hasvoicemail = yes
hassip = yes
hasiax = yes
callwating = yes
threewaycalling = yes
callwaitingcallerid = yes
transfer = yes
canpark = yes
canallforward = yes
callreturn = yes
callgroup = 1
pickupgroup = 1
nat = yes
```

La rubrique générale



Utilisateur 6001



Utilisateur 6002

Plusieurs options permettent de définir et de paramétrer un client parmi eux :

Type => - user = peut appeler mais ne peut pas recevoir d'appel

- peer = peut recevoir des appels

- friend = peut appeler et recevoir des appels

Host => - dynamic : pout connecté à ce compte SIP à partir de n'importe quelle adresse IP

- nom d'hôte : Nom d'hôte du client

- adresse IP : Adresse IP du client.

dtmfmode=> rfc2833 : définit des formats des paquets de RTP utilisés pour transporter des chiffres DTMF

disallow=>all : Désactivation de tous les codecs

allow=>ulaw : Activation du codec µlaw

fullname => Prénom et NOM de l'utilisateur (ce qui sera afficher sur le téléphone lors d'un appel)

Username => Si Asterisk agit entre un client **SIP** et un serveur **SIP** distant, ce champ est utilisé pour authentifier le message **INVITE** envoyé par **Asterisk** au serveur (Identifiant de l'utilisateur)

Secret => Mot de passe de l'utilisateur

Context => Contexte (Utiliser dans le fichier **extensions.conf**)

Une fois le fichier Sip.conf enregistré allez dans la console Asterisk, tapez reload et après on tape la commande sip show users, les deux comptes utilisateurs que nous venons de créer devrait y apparaître

```
*CLI> sip show users
Username                 Secret           Accountcode      Def.Context     ACL  Forcerport
6002                     2018                              work            No   No
6001                     1976                              work            No   No
```

II.4.2. Configuration du Dialplan

La création des utilisateurs se fait dans le fichier **extensions.conf** ,

[root@localhost ymezhoudi]# vim /etc/asterisk/extentions.conf

Ouverture du fichier extentions.conf dans la console pour modifications

```
[work]
exten => 6001,1,dial(SIP/6001,20)
exten => 6001,2,Hangup()
exten => 6002,1,dial(SIP/6002,20)
exten => 6002,2,Hangup()
```

[Work] : est le contexte, c'est une sorte de conteneur dans lequel les utilisateurs faisant partie de ce contexte pourront communiquer entre eux.

Lors de la création de nos deux utilisateurs nous avons spécifié le contexte work.

Exten : déclare l'extension (on peut aussi simplement dire numéros)

6001,6002 : Prend les extensions (ou numéros) 6001 et 6002

1 : Ordre de l'exécution

Dial : application qui va être utilisée (ici appel)

SIP: Protocol qui va être utilisé

20: temps d'attente avant de passer à l'étape suivante.

Donc la ligne **exten => 6001(ou 6002), 1, Dial (SIP/6001(ou 6002) ,20)** se traduit par :

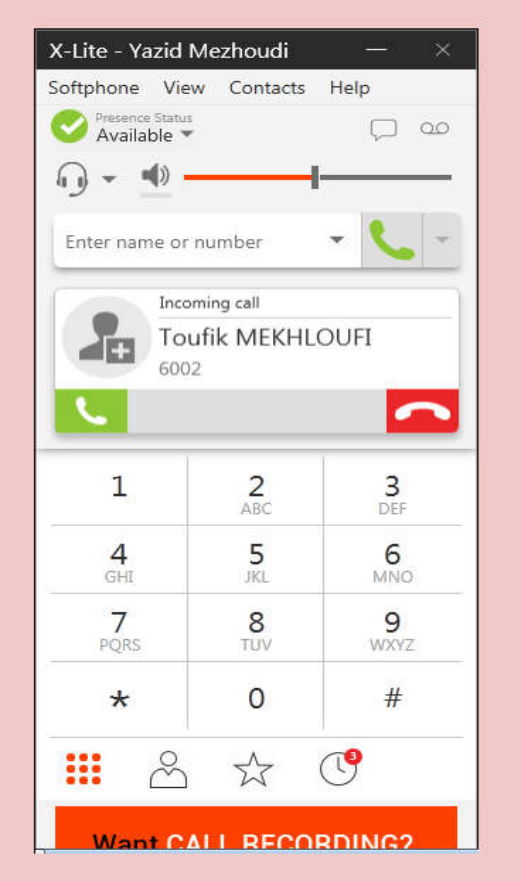
Quand on compose le numéro (par exemple) 6001, on appelle le numéro 6001 et si au bout de 20 secondes il n'y a pas de réponse on passe à la ligne du dessous (ou l'ordre d'exécution =2).

exten => 6001(ou 6002), 2, Hangup() qui permet de raccrocher s'il n'y a pas de réponse au bout de 20 secondes.

En peut utiliser une formule générale qui regroupe une suite d'extensions (6000 à 6999):

```
[work]
exten => _6XXX,1,dial(SIP/${EXTEN},20)
exten => _6XXX,2,Hangup()
```

Pour faire les tests, il suffit d'appeler par exemple à partir du compte de Toufik MEKHLOUFI (6002) le numéro 6001.



Simulation d'un appel

II.5. Conclusion

Dans ce chapitre, nous avons présenté l'environnement physique du travail, ainsi que les divers programmes open source et leurs configurations approuvées pour mettre en œuvre les services VoIP (Voice over Internet Protocol) sous la plate-forme Astérisk.

CHAPITRE III : Performances, Attaque & sécurité VOIP

III.1. Introduction:

La voix sur IP, a fourni de nombreux avantages à ses utilisateurs, et les a permirent de bénéficier de nouveaux services de télécommunication tel que la vidéoconférence a un prix réduit .

Cette technique basé sur des systèmes informatiques, pouvant faire sujet d'attaque des pirates, au niveau spécifique des réseaux IP ou la VoIP proprement dite.

Cela présente un certain nombre de faiblesses en termes de: protocole, logiciel, système d'exploitation, infrastructure physique et erreur humaine.

Ces vulnérabilités doivent être soigneusement examinées afin d'établir une protection efficace contre les attaques.

Pour faire face à ces attaques, La sécurité du réseau VoIP doit être basée sur deux types de protection:

- La sécurité traditionnelle des réseaux informatiques (Firewall, IPS, IDS, Antivirus,…..)

- La sécurité spécifique VoIP

Dans ce chapitre, on va voir quelques attaques qui menacent la VoIP. Et les bonnes pratiques pour sécuriser les communications de type voix sur IP.

Les menaces peuvent être vues comme des violations potentielles de la sécurité qui existent en raison des vulnérabilités du système. Les menaces envers un système VOIP comprennent les éléments suivants : - Destruction d'information; Corruption ou modification d'informations ; Vol, suppression ou perte d'informations; Divulgation d'informations ; et Interruption de service. Les menaces peuvent être classées en menaces accidentelles ou menaces intentionnelles (les attaques) qui peuvent être actives ou passives.

Les attaques qui menacent la VoIP peuvent être classées en deux types : les attaques internes et les attaques externes. Les attaques externes sont lancées par des personnes autres que

celle qui participe à l'appel, et ils se produisent généralement quand les paquets VoIP traversent un réseau peu fiable et/ou l'appel passe par un réseau tiers durant le transfert des paquets. Les attaques internes s'effectuent directement du réseau local dans lequel se trouve l'attaquant

III.2. Attaques sur le protocole

La vulnérabilité protocolaire s'effectue au niveau des protocoles qu'utilise la VoIP. Étant donné que les protocoles de la VoIP utilisent TCP et UDP, comme moyen de transport et par conséquent elles sont aussi vulnérables à toutes les attaques contre ces protocoles, telles que le détournement de session(TCP) et la mystification(UDP).

Les attaques les plus fréquentes contre le système VoIP sont

III.2.1. Sniffing

Le reniflement de trafics (Sniffing) constitue l'une des méthodes couramment utilisée par les pirates pour espionner le trafic sur le réseau. Dans la pratique, les hackers ont généralement recours à ce procédé pour détecter tous les messages circulant sur le réseau en récupérant des mots de passe et des données sensibles. Les pirates utilisent des sniffers réseau pour pouvoir surveiller le réseau et soustraire frauduleusement les différents types de données confidentielles susceptibles de les intéresser.

III.2.2. Suivie des appels

Appelé aussi **Call tracking** en Anglais, cette attaque se fait au niveau du réseau LAN/VPN et cible les terminaux (soft/hard phone).elle a pour but de connaitre qui est en train de communiquer à quelle heure, et pendant combien de temps. L'attaquant doit récupérer les messages INVITE et BYE en écoutant le réseau

III.2.3. Injection de paquet RTP

L'attaque par injection de paquet RTP est destinée à perturber une communication en cours. L'attaquant doit d'abord écouter le flux RTP de l'appelant vers l'appelé, analyser son contenu

et générer un paquet RTP avec un en-tête similaire mais avec un numéro de séquence et un horodatage (timestamp) plus grands, pour que ce paquet soit reproduit avant les autres paquets (s'ils sont déjà reproduits). La communication sera interrompue et l'appel ne pourra pas continuer correctement.

Pour effectuer cette attaque, l'attaquant doit pouvoir écouter le réseau pour localiser une communication et ainsi déterminer les horodatages des paquets RTP.

Il doit également pouvoir d'insérer des messages RTP qu'il avait créés avec un horodatage modifié.

III.2.4. Les Spam

Les lignes VoIP sont la cible d'actions marketing indésirables qui leur sont propres, plus connues sous le nom de "SPIT" (Spam over Internet Telephony).

Et il existe trois types de SPAM.

- Call Spam : Ce type de spam est défini comme une masse de tentatives d'initiation de session (des requêtes INVITE) non sollicitées. Généralement c'est un UAC (User Agent Client) qui lance, en parallèle, un grand nombre d'appels. Si l'appel est établi, l'application génère un ACK, rejoue une annonce préenregistrée, et ensuite termine l'appel.

- IM (Instant Message) Spam : Ce type de spam est semblable à celui de l'e-mail. Il est défini comme une masse de messages instantanés non sollicitées. Les IM spams sont pour la plupart envoyés sous forme de requête SIP. Ce pourraient être des requêtes INVITE avec un entête « Subject » très grand, ou des requêtes INVITE avec un corps en format texte ou HTML.

  Bien-sûr, l'IM spam est beaucoup plus intrusif que le spam email, car dans les systèmes actuels, les IMs apparaissent automatiquement sous forme de pop-up à l'utilisateur

- Présence Spam : Ce type de spam est semblable à l'IM spam. Il est défini comme une masse de requêtes de présence (des requêtes SUBSCRIBE) non sollicitées. L'attaquant fait ceci dans le but d'appartenir à la " white list " d'un utilisateur afin de lui envoyer des messages instantanés ou d'initier avec lui d'autres formes de communications. L'IM Spam est

différent du Presence Spam dans le fait que ce dernier ne transmet pas réellement de contenu dans les messages

III.2.5. Le déni de service (DOS : Denial of service)

L'attaque de déni de service consiste à augmenter et surcharger le serveur VOIP avec des requêtes jusqu'à ce qu'elle ne puisse plus suivre et s'arrêter. Il est envisageable de saturer les réseaux des sociétés équipées de la voix sur IP, bloquant ainsi les communications internes et externes ainsi que le système d'information. Les attaques par déni de service prennent plusieurs formes. Les plus courants sont ceux qui visent à utiliser toute la bande passante disponible ou à exploiter les problèmes de TCP / IP d'une manière inappropriée pour empêcher les tentatives de connexion. Dans le cadre de la solution VoIP, de nombreux éléments tels que le téléphone, le réseau, le système d'exploitation, l'application, etc. peuvent être attaqués.

Dans une attaque de type Dos, les ressources d'un serveur ou d'un réseau sont épuisées par un flot de paquets. Un seul attaquant visant à envoyer un flot de paquets peut être identifié et isolé assez facilement. Cependant l'approche de choix pour les attaquants a évolué vers un déni de service distribué (DDoS). Une attaque DDoS repose sur une distribution d'attaques DoS, simultanément menées par plusieurs systèmes contre un seul. Cela réduit le temps nécessaire à l'attaque et amplifie ses effets.

III.2.6. Détournement d'appel (Call Hijacking)

Le Call Hijacking consiste à détourner un appel. De nombreux fournisseurs de services VoIP utilisent le Web en tant qu'interface utilisateur pour accéder à leur système téléphonique.

Un utilisateur authentifié peut échanger les paramètres de ses conversions d'appel via cette interface Web. Cela peut être approprié, mais un utilisateur malveillant peut utiliser les mêmes moyens pour mener une attaque.

Exemple: lorsqu'un agent SIP envoie un message INVITE pour lancer un appel,

L'attaquant envoie un message de redirection 3xx indiquant que l'appelé s'est déplacé, et donne en même temps sa propre adresse en tant qu'adresse de transfert. A partir de ce moment, tous les appels entrants vers l'utilisateur seront reçus par l'attaquant.

Un appel piraté est lui-même un problème, mais il est plus dangereux lorsqu'une entreprise transfère des informations sensibles et confidentielles.

III.2.7. L'écoute clandestine

L'**eavesdropping** est l'écoute clandestine d'une conversation téléphonique. Un attaquant ayant accès à un réseau VoIP peut sniffer le trafic et décoder la conversation vocale

Le principe de l'écoute clandestine est montré dans la figure 2-1 comme suit :

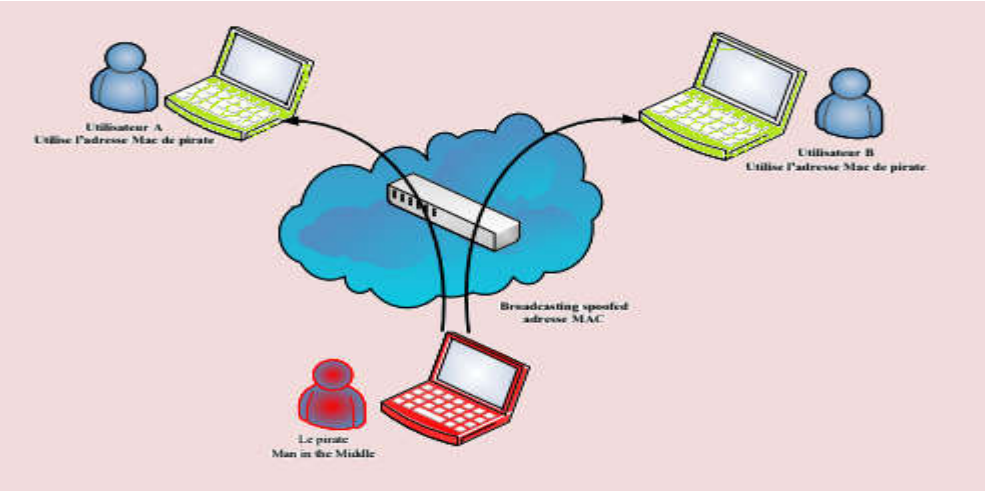


Figure 15: Exemple de détournement d'appel " Man in the middle"

1. déterminer les adresses MAC des victimes (client-serveur) par l'attaquant

2. Envoi d'une requête ARP non sollicités au client, pour l'informer du changement de l'adresse

MAC du serveur VoIP à l'adresse MAC.

3. Envoi d'une requête ARP non sollicités au serveur, pour l'informer du changement de l'adresse

MAC du client à l'adresse MAC.

4. Désactiver la vérification des adresses MAC sur la machine d'attaque afin que le trafic puisse circuler entre les 2 victimes

III.3. Les vulnérabilités de l'infrastructure

L'infrastructure VoIP comprend des téléphones IP, des Gateway, des serveurs (proxy, registrer, etc.). Chaque composant, que ce soit un système embarqué ou d'un serveur avec système d'exploitation standard, est accessible à travers le réseau comme n'importe quel autre ordinateur.

Chacun comporte un processeur qui exécute des programmes pouvant être attaqués ou utilisés comme points de lancement pour une attaque plus profonde.

III.3.1. Faiblesses dans la configuration des dispositifs de la VoIP

De nombreux appareils VoIP, dans leur configuration par défaut, peuvent avoir une variété de ports TCP et UDP ouverts. Les services qui opèrent sur ces ports peuvent être vulnérable aux attaques DoS ou au buffer overflow .

Également Plusieurs dispositifs de la VoIP exécutent un serveur WEB d'administration à distance. Cela peut être vulnérable aux attaques buffer overflow et à la divulgation d'informations.

Si les services accessibles ne sont pas configurés avec un mot de passe, un attaquant peut obtenez un accès non autorisé à cet appareil.

Les services SNMP (Simple Network Management Protocol) fournis par ces périphériques Peut être vulnérable aux de reconnaissance ou attaques d'overflow..

Aussi plusieurs appareils VoIP sont configurés pour être téléchargés périodiquement leur fichier de configuration depuis un serveur TFTP ou d'autres mécanismes. L'attaquant peut potentiellement détourner ou mystifier cette connexion et tromper l'appareil, qui téléchargera un fichier de configuration illicite au lieu du fichier réel.

III.3.2. Les téléphones IP

Un attaquant pourrait compromettre un périphérique IP, tel qu'un téléphone IP, un softphone ou un autre logiciel ou matériel client. En général, il obtient des privilèges qui lui permettent de contrôler entièrement les fonctionnalités de l'appareil.

Le contrôle fraude sur le terminal (téléphone IP) peut être effectuée à distance ou par un accès physique à l'appareil. Hacker peut modifier les aspects opérationnels d'un tel appareil:

Également un firmware modifié de manière malveillante peut être téléchargé et installé, permettre de :

•rediriger les appels entrants vers un autre point final sans connaissance de l'utilisateur.

• surveiller les appels.

• Pour les informations de signalisation et / ou les paquets contenant la voix d'être dirigée vers un autre appareil ainsi que d'être enregistrés et/ou modifiés.

De rompre la disponibilité du point final. Par exemple, ce dernier peut refuser automatiquement tous les appels, ou élimine toute notification sonore ou visuelle lorsqu'un appel arrive. Les appels peuvent également être interrompus de manière inattendue (certains téléphones IP le permettent via l'interface Web).

Toutes les informations utilisateur stockées sur l'appareil peuvent être extraites.

L'accès non autorisé à l'appareil téléphonique IP peut être dû à un autre objet piraté sur le réseau IP ou à des informations collectées sur le réseau.

Les softphones ne réagissent pas de la même manière aux attaques que leurs homologues sur les téléphones IP. Ils sont plus vulnérables aux attaques en raison du nombre de vecteurs répertoriés dans le système, à savoir les vulnérabilités du système d'exploitation, les vulnérabilités des applications, les vulnérabilités du service, les vers, les virus, etc. En outre, le softphone reste dans le volet de données, est ainsi sensible aux attaques lancées contre ce segment et pas seulement contre l'hôte qui héberge l'application softphone.

Les téléphones IP exécutent leurs systèmes d'exploitation avec un nombre limité de services pris en charge et présentent donc moins de vulnérabilités

III.3.3. Les serveurs

Un attaquant pourrait cibler les serveurs qui fournissent la téléphonie IP.

Une renonciation à une telle entité met généralement en danger l'ensemble du réseau de téléphonie dont le serveur fait partie.

Par exemple, si le serveur de signalisation est compromis, un attaquant peut entièrement contrôler les informations de signalisation pour différents appels. Ces informations sont acheminées via le serveur piraté. Avoir le contrôle de l'information de signalisation permet à l'attaquant de changer n'importe quel paramètre lié à l'appel.

Si un serveur IP est installé sur un système d'exploitation, il peut être la cible de virus, de vers ou de code malveillant.

III.3.4. Les vulnérabilités du système d'exploitation

Les vulnérabilités du système d'exploitation sont principalement liées au manque de sécurité lors de la phase de développement initiale du système d'exploitation et ne sont découvertes qu'après le lancement du produit.

L'une des principales vulnérabilités des systèmes d'exploitation est le buffer overflow. Cela permet à un attaquant de prendre le contrôle partiel ou complet de la machine.

Les périphériques VoIP tels que les téléphones IP, Call Managers, Gateway et les serveurs proxy héritent les mêmes vulnérabilités du système d'exploitation ou du firmware sur lequel ils s'exécutent.

Il y a une centaine de vulnérabilités qui peuvent être exploitées à distance sur Windows et même sur Linux. Beaucoup de ces exploits sont disponibles gratuitement et prêts à être téléchargés sur Internet.

Peu importe comment, une application de VoIP s'avère sûre, elle devient menacée si le système d'exploitation sur lequel elle s'exécute est compromis

III.4. Sécurisation et bonne pratiques

Nous avons déjà vu que des vulnérabilités existent au niveau du protocole, des applications et des systèmes d'exploitation. Pour cela, la sécurité a été divisée en trois niveaux: la sécurité du protocole, la sécurité de l'application et la sécurisation du système d'exploitation.

III.4.1. L'authentification

L'une des méthodes les plus importantes pour anticiper une attaque sur un système de téléphonie consiste à identifier clairement l'identité des dispositifs ou des personnes participant à la conversation.

Plusieurs solutions simples sont implémentées pour cela, il est recommandé d'utiliser des mots de Pass complexe lors de la configuration des clients SIP En effet, sachez que certains hackers développer des robots pour sonder les réseaux informatiques et dès que l'un d'eux répond au protocole SIP, un algorithme sophistiqué est engagé et teste toutes les combinaisons mots de passe possibles. Donc, il faut éviter

- Les Mots de passe trop courts
- Les suites numériques (123456) ou alphabétiques (abcd)
- Les séquences logiques telles que les prénoms ou les dates
- Un mot de passe unique pour toutes les extensions SIP
- Un mot de passe similaire pour le système Linux, la base de données MySql et Asterisk

Un mot de passe recommandé doit être, complètement aléatoire de 8 caractères minimum, impliquant une combinaison de caractères spéciaux, de majuscules, de lettres minuscules, les

chiffres ne sont pas suivis. Pour proscrire, utilisez 1 et 1 (L minuscule) ainsi que 0 (zéro) et O d'Oscar.

La confidentialité des mots de passe est primordiale: lors de la configuration des téléphones ou softphones sur site, il est impératif d'être discret lors de la saisie des mots de passe, et bien entendu de ne pas les communiquer aux utilisateurs.

III.4.2. Sécurisation protocolaire

La prédominance et la facilité de sniffer des paquets et d'autres techniques de capture de paquets IP sur un réseau VoIP rendent le chiffrement indispensable. La sécurisation de la VoIP consiste à protéger les personnes qui sont interconnectées.

IPsec peut être utilisé pour atteindre deux objectifs. Garantir l'identité des deux extrémités et protéger la voix. VOIPsec (VoIP utilisant IPsec) permet de réduire les menaces, les sniffeurs de paquets et de nombreux types de trafic " vocal analyze". Associé à un pare-feu, IPsec rend la VOIP plus sûre qu'une ligne téléphonique traditionnelle. Il est important de noter, cependant, que IPsec n'est pas toujours un bon moyen pour certaines applications, et que certains protocoles doivent continuer à s'appuyer sur leurs propres fonctionnalités de sécurité.

III.4.2.1. VPN VoIP

La VoIP VPN combine la voix sur IP et la technologie de réseau virtuel privé pour fournir une méthode de préservation de la transmission de la voix. Comme la VoIP transmet la voix numérisée dans un flux de données, la solution VoIP VPN semble la plus appropriée puisqu'elle offre un cryptage des données grâce aux mécanismes de cryptage, puisqu'elle permet d'offrir l'intégrité des paquets VoIP.

Cryptage aux points terminaux

Puisque notre objectif est d'assurer la confidentialité et l'intégrité des clients, la nécessité de concevoir des mécanismes d'authentification et de cryptage pour IP. Parce qu'il sécurise le paquet dans son ensemble (contrairement au mode transport qui ne sécurise que la charge utile IP). Le mode tunnel (réseau privé virtuel sécurisé), est basé sur 'l'encapsulation de tout le

paquet IP et ajoute un nouvel en-tête pour le routage de ce dernier, et l'authenticité des paquets reçus assurée par IPSec (Internet Protocole de sécurité) entre les machines impliquées.

Ce mode est généralement utilisé pour un routeur à routeur. Ses avantages sont sa capacité à l'utiliser uniquement sur des communications spécifiques (sans perturber d'autres communications), et puisque IPSec est en dessous de la couche de transport (TCP, UDP); il est donc transparent aux applications (possibilité d'augmenter la sécurité sans modifier les applications de plus haut niveau). Une fois implémenté, IPSec est transparent pour les utilisateurs.

Donc IPSec fournit :

- Un protocole de d'authentification indiqué par l'en-tête d'authentification (AH

(Authentification Header)).

- Contrôle d'accès.
- Authentification de l'origine des données.
- Rejet de paquets rejoués.

- Un protocole combiné chiffrement authentification (ESP (Encapsulating Security Payload)).

- Confidentialités (chiffrement).
- Confidentialités limitée au flot du trafic.

Peut être utilisé avec différents mécanismes de sécurité pour assurer la confidentialité, l'originalité et l'intégrité, ainsi que pour fournir une gestion automatique des clés (Figure 2-2).

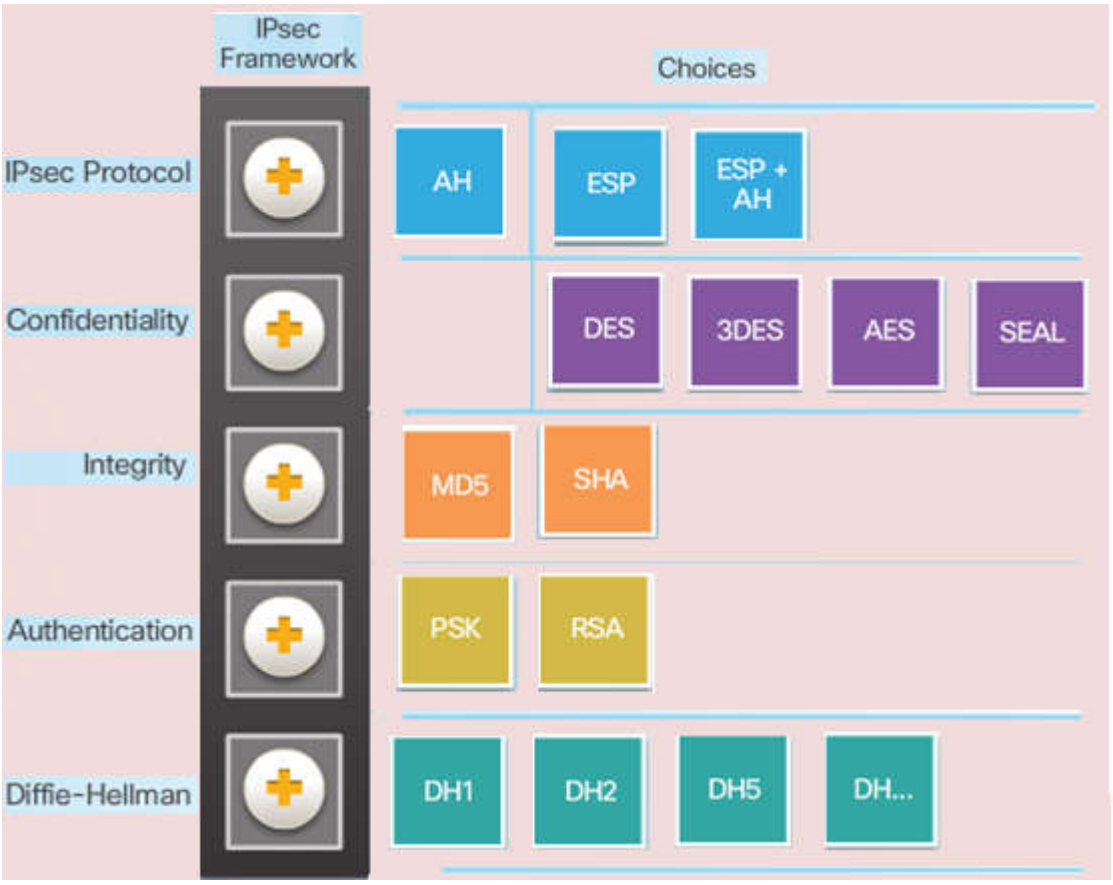


Figure 16: IPsec Technologies

III.4.2.2. Protocole TLS

Le protocole TLS est utilisé pour sécuriser les échanges au niveau de la couche de transport (**TLS : Transport Layer Security**). TLS, anciennement connu sous le nom de **Secure Sockets Layer (SSL)**, est un protocole de sécurisation des échanges sur internet. C'est un protocole standard conçu pour sécuriser l'échange de données entre le client et le serveur indépendamment de tout type d'application. TLS agit comme une couche supplémentaire au-dessus de TCP.

Les protocoles SSL / TLS peuvent être divisés en 2 couches.

La première couche est constituée par des protocoles de négociation (Handshake, Cipher, Alert) et la deuxième couche est le protocole Record. (Figure 2.3).

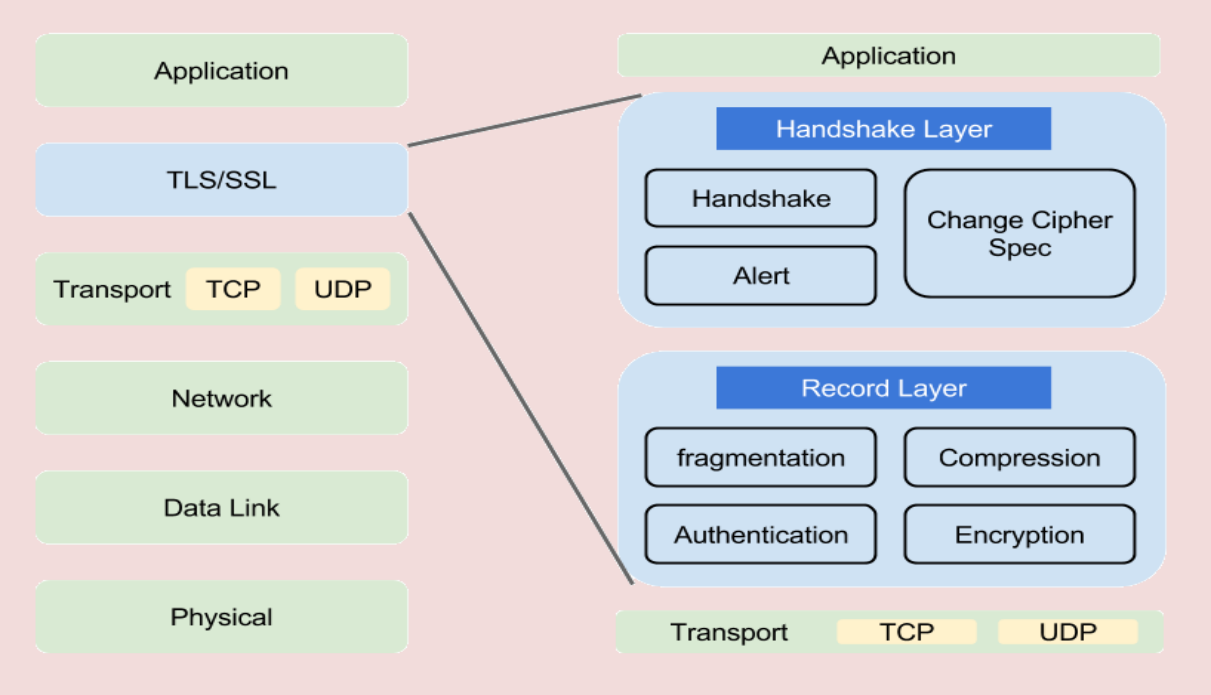


Figure 17 : Empilement des sous-couches protocolaires de TSL/SSL

- **Le protocole Hanshake**: ce protocole permet au serveur et au client de :

- s'authentifier mutuellement ;
- négocier :
  - Les algorithmes de chiffrement ;
  - Les algorithmes de MAC (Message Authentification Code) ;
  - Les clés symétriques qui vont servir au chiffrement avant que l'application ne transmette son premier octet.

- Le protocole Change Cipher Spec : Ce protocole contient un seul message : change_cipher_spec. Il est envoyé par les deux parties à la fin du protocole de négociation. Ce message transite chiffré par l'algorithme symétrique précédemment négocié.

- Le protocole Alert: Ce protocole spécifie les messages d'erreur que peuvent s'envoyer clients et serveurs. Les messages sont composés de deux octets. Le premier est soit warning soit fatal. Si le niveau est fatal, la connexion est abandonnée. Les autres connexions sur la même session ne sont pas coupées mais on ne peut pas en établir de nouvelles. Le deuxième octet donne le code d'erreur.

Les erreurs fatales	Les warnings
Unexpected_message : **indique que le message n'a pas été reconnu**	Close_notify : annonce la fin d'une connexion
Bad_record_mac : **signale une signature MAC incorrecte**	No_certificate : répond une demande de certificat s'il n'y en a pas
Decompression_failure : **indique que la fonction de décompression a reçu une mauvaise entrée**	Bad_certificate : le certificat reçu n'est pas bon (par exemple, sa signature n'est pas valide)
Handshake_failure : **impossible de négocier les bons paramètres**	Unsupported_certificate : le certificat reçu n'est pas reconnu
Illegal_parameter : **indique un champ mal formaté ou ne correspondant à rien**	Certificate_revoked : certificat révoqué par l'émetteur
	Certificate_expired : certificat expiré
	Certificate_unknown : pour tout problème concernant les certificats et non listé ci-dessus.

Tableau 3-1 : Liste des erreurs SSL/TLS

- **Protocole Record:** Ce protocole supervise les autres protocoles TLS, en fournissant une interface unifiée pour la transmission de données**.**

- Rôle

Encapsulation : Permet la transmission et l'identification des données TLS sous une forme Homogène.

Confidentialité : Assure que le contenu du message ne soit pas lisible par un tiers: Les données sont cryptées à l'aide des clés produites lors de la négociation.

Intégrité et Identité : Permet de vérifier la validité des données transmises, grâce aux signatures MAC: Cette signature est également créée à l'aide des clés produites lors de la négociation.

- Processus d'encapsulation :

Segmentation : Les données sont découpées en blocs de taille inférieure à 16 384 octets

Compression : Les données sont compressées en utilisant l'algorithme choisi lors de la négociation. A partir de SSL 3.0, il n'y a plus de compression.

Chiffrement: Le paquet obtenu est chiffré à l'aide l'algorithme de chiffrement. Le choix peut se faire entre RC2, RC4, DES avec une clef de taille 40 bits ou de 64 bits, ou l'algorithme de **Fortezza**. Ce dernier algorithme est un algorithme secret défense aux États Unis. Notons, qu'il est possible de choisir des échanges en clair; Algorithme de hachage utilisé, qui peut être soit le MD5, soit le SHA. Il est possible de ne choisir aucun algorithme de hachage; La négociation de cette suite de chiffrement se fait en clair pendant l'établissement de la session.

Le **tableau 3-2** donne l'ensemble des algorithmes supportés par SSL tandis que **tableau 3-3** contient les suites de chiffrement reconnues.

Fonction	**Algorithme**
Echanges de clefs	RSA, Fortezza, Diffie-Hellman
Chiffrement symétrique à la volée	RC4 avec clefs de 128 bits ou de 40bits
Chiffrement symétrique en blocs	DES, DES40, 3DES RC2, IDEA,Fortezza
Hachage	MD5, SHA

Tableau 3-2 : algorithmes supportés par SSL

Echange de clefs	**Chiffrement symétrique**	**Hachage**	**Signature**
	Sans chiffrement RC4-40	MD5 ou SHA MD5	
RSA	RC4-128 RC2 CBC 40 IDEA CBC	MD5 ou SHA MD5 SHA	

	DES40 CBC DES CBC 3DES EDE CBC	SHA SHA SHA	
Diffie-Hellman	DES40 CBC DES CBC 3DES, EDE CBC	SHA SHA SHA	DSS ou RSA DSS ou RSA DSS ou RSA
Diffie-hellman ephémère	DES40 CBC DES CBC 3DES EDE CBC	SHA SHA SHA	DSS ou RSA DSS ou RSA DSS ou RSA

Tableau 3-3: les suites de chiffrement reconnues par SSL

- **Réception des paquets**

A la réception des paquets, le destinataire effectue les opérations suivantes :

1- Vérification de l'entête SSL

2- Déchirage du paquet

3- Vérification du champ HMAC (en appliquant la même fonction que ci-dessus aux données déchiffrées puis en comparant le résultat au HMAC reçu)

4- Décompression des données

5- Réassemblage des parties

Si au cours de ces vérifications se passe mal, une alarme est générée.

III.4.2.3. Secure RTP ou SRTP

SRTP est conçu pour sécuriser les échanges multimédias sur les réseaux IP. Il couvre les lacunes de protocoles de sécurité existants comme IPsec (IP Security), dont le mécanisme d'échanges de clés est trop lourd. Il aussi est bâti sur le protocole temps réel RTP (Real Time Transport Protocol). Il associe aussi une demi-douzaine de protocoles complémentaires. Il est donc compatible à la fois avec des protocoles d'initiation de session de voix sur IP tel que SIP (Session Initiation Protocol), ainsi que le protocole de diffusion de contenu multimédia en temps réel RTSP (Real Time Streaming Protocol). Mais, surtout, il s'adjoint les services du protocole de gestion de clé MIKEY (Multimedia Internet KEYing).

- **Principe de fonctionnement de SRTP**

Avec une gestion de clé appropriée, SRTP est sécurisé pour les applications unicast et multicast de RTP. En théorie, SRTP est une extension du protocole RTP dans lequel ont été rajoutée des options de sécurité. En effet, il a pour but d'offrir plusieurs implémentations de cryptographie tout en limitant l'overhead lié à l'utilisation des chiffrements. Il propose des algorithmes qui monopoliseront au minimum les ressources et l'utilisation de la mémoire.

Surtout, il permet de rendre RTP indépendant des autres couches en ce qui concerne l'application de mécanismes de sécurité.

Pour implémenter les différents services de sécurité précités, SRTP utilise les composants principaux suivants :

- Une clé maîtresse utilisée pour générer des clés de session; Ces dernières seront utilisées pour chiffrer ou pour authentifier les paquets.
- Une fonction utilisée pour calculer les clés de session à partir de la clé maîtresse.
- Des clés aléatoires utilisées pour introduire une composante aléatoire afin de contrer les éventuels rejeu ou effets de mémoire.

SRTP utilise deux types de clés : clef de session et clef maîtresse. Par « clef de session » nous entendons une clef utilisée directement dans les transformations cryptographiques; et par « clef maîtresse », nous entendons une chaîne de bit aléatoire à partir desquelles les clefs de sessions sont dérivées par une voie sécurisé avec des mécanismes cryptographiques.

- **Service de sécurités offertes par SRTP**

Les principaux services offerts par SRTP sont :

- Rendre confidentielles les données RTP, que ce soit l'en-tête et la charge utile ou seulement la charge utile.
- Authentifier et vérifier l'intégrité des paquets RTP. L'émetteur calcule une empreinte du message à envoyer, puis l'envoie avec le message même.

- La protection contre le rejeu des paquets. Chaque récepteur tient à jour une liste de tous les indices des paquets reçus et bien authentifiés.

- **Format du paquet SRTP**

Un paquet SRTP est généré par transformation d'un paquet RTP grâce à des mécanismes de sécurité. Donc le protocole SRTP effectue une certaine mise en forme des paquets RTP avant qu'ils ne soient sur le réseau. La figure suivante présente le format d'un paquet SRTP

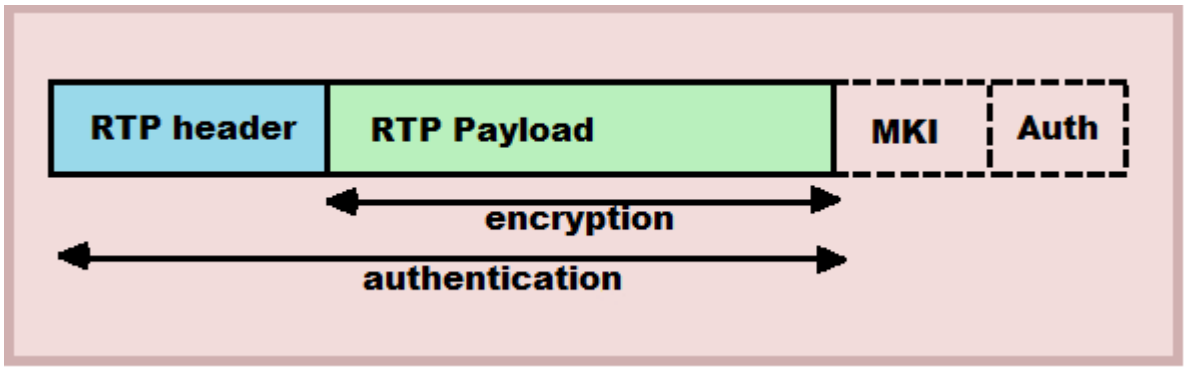


Figure 18: Format d'un paquet SRTP

On remarque que le paquet SRTP est réalisé en rajoutant deux champs au paquet RTP :

- SRTP MKI (SRTP Master Key identifier) : sert à re-identifier une clef maîtresse particulière dans le contexte cryptographique. Le MKI peut être utilisé par le récepteur pour retrouver la clef primaire correcte quand le besoin d'un renouvellement de clefs survient.

- Authentication tag : est un champ inséré lorsque le message a été authentifié. Il est recommandé d'en faire usage. Il fournit l'authentification des en-têtes et données RTP et indirectement fournit une protection contre le rejeu de paquets en authentifiant le numéro de séquence.

III.4.3. Sécurisation de l'application

Plusieurs méthodes peuvent être appliquées pour sécuriser l'application, ces méthodes varient selon le type d'application (serveur ou client). Pour sécuriser le serveur il faut :
- L'utilisation d'une version stable, Il est bien connu que toute application non stable contient surement des erreurs et des vulnérabilités. Pour minimiser les risques, il est impératif d'utiliser une version stable.

- Tester les mises à jour des softwares dans un laboratoire de test. Il est très important de tester toute mise à jour de l'application dans un laboratoire de test avant de les appliquer sur le système en production

- Ne pas tester les correctifs sur le serveur lui-même:

- Ne pas utiliser la configuration par défaut qui sert juste à établir des appels. Elle ne contient aucune protection contre les attaques.

- Ne pas installer une application client dans le serveur.

Certains paramètres doivent être appliqués de manière sélective. Ces paramètres renforcent la sécurité de l'application, on peut les activer ou les interdire sur la configuration générale de l'application, comme on peut juste utiliser les paramètres nécessaires pour des clients bien déterminé et selon le besoin bien sûr. Ces paramètres protègent généralement contre le déni de service et ces différentes variantes. Il est conseiller d'utiliser les paramètres qui utilise le hachage des mots de passe, et cela assure la confidentialité

III.4.4. Sécurisation du système d'exploitation

Il est très important de sécuriser le système sur lequel est implémenté le serveur de VoIP. En effet, si le système est compromis, l'attaque peut se propager sur l'application serveur. Celle-ci risque d'affecter les fichiers de configuration contenant des informations sur les clients enregistrés.

Il y a plusieurs mesures de sécurités à prendre pour protéger le système d'exploitation :
- utiliser un système d'exploitation stable. Les nouvelles versions toujours contiennent des bugs et des failles qui doivent être corrigés et maîtrisés avant.

- mettre à jour le système d'exploitation en installant les correctifs de sécurité recommandé pour la sécurité.

- Ne pas mettre des mots de passe simple et robuste. Ils sont fondamentaux contre les intrusions. Et ils ne doivent pas être des dates de naissances, des noms, ou des numéros de téléphones. Un mot de passe doit être assez long et former d'une combinaison de lettre, de chiffres et ponctuations.

- Ne pas exécuter le serveur VoIP avec un utilisateur privilège. Si un utilisateur malveillant arrive à accéder au système via une exploitation de vulnérabilité sur le serveur VoIP, il héritera tous les privilèges de cet utilisateur.

III.5. Conclusion

Dans ce chapitre, nous avons présenté les faiblesses les plus importantes et la façon de les traiter, alors que nous appelons un certain nombre de mesures de sécurité doivent être prises en compte pour assurer la qualité du service dans le réseau ainsi que la sécurité.

CHAPITRE IV : Sécurisation de la solution mise en place

IV.1. Introduction

Après avoir étudié les protocoles de VoIP, identifié les attaques qui menacent les Systèmes VoIP et les bonnes solutions pour sécuriser le serveur Asterisk. Dans ce chapitre, nous effectuerons une analyse de vulnérabilité et identifierons les faiblesses et les insuffisances dans les mesures de sécurité pour simuler et corriger les faiblesses et mettre en œuvre des mesures de sécurité garantissant une communication fiable et robuste pour sécuriser notre solution VoIP basée sur le serveur Asterisk.

Ce chapitre se compose de deux parties principales. Dans le premier, nous utiliserons des scénarios d'attaques par divers programmes de piratage. Dans la deuxième partie, nous présenterons les solutions mises en place pour sécuriser la solution déployée.

IV.2. Attaques Simulées

Pour la simulation des attaques on a installé le système Kali Linux sur une machine VMware dans un pc dédié pour le hacker.

IV.2.1. Machine Kali Linux

Kali Linux est une distribution Linux basée sur Debian, gratuite et Open source, son but principal est d'aider les professionnels travaillant dans la sécurité informatique afin de pouvoir réaliser des tests de pénétrations avancés ainsi que des audits de sécurité. Contient plusieurs centaines d'outils orientés pour différentes tâches de sécurité informatique, tel que les tests de pénétrations, la recherche en sécurité, le Forensics et l'ingénierie inversé.

Parmi ces outils, et ceux qui nous intéresse ici en trouve :

- SIPVicious
- Ettercap
- Wireshark
- Inviteflood

IV.2.1.1. SIPVicious

La suite SIPVicious est un ensemble d'outils pouvant être utilisés pour l'audit des systèmes VoIP basés sur SIP. Il se compose actuellement de quatre outils:

- svmap : c'est un scanner sip, qui Liste les périphériques SIP trouvés sur une plage IP
- svwar : permet de identifie les extensions actives sur un PBX
- svcrack : permet de cracker les mots de passes des utilisateurs
- svreport : gère les sessions et exporte les rapports vers différents formats

IV.2.1.2. Ettercap

Est un logiciel libre d'analyse du réseau informatique. Il est capable d'effectuer des attaques sur le protocole ARP pour se positionner comme "homme au milieu"(Man In The Middle) afin de :

- Faire passer le flux à travers l'ordinateur pirate
- Infecter, remplacer et supprimer des données dans une connexion
- Découvrir les mots de passe pour les protocoles comme FTP, HTTP, SSH1 ...

IV.2.1.3. Wireshark

Wireshark est un logiciel libre d'analyse de protocole, utilisé dans le dépannage et l'analyse de réseaux informatiques, le développement de protocoles, l'éducation et la rétro-ingénierie, mais aussi le piratage. C'est l'analyseur réseau le plus populaire du monde. Cet outil extrêmement puissant fournit des informations sur des protocoles réseaux et applicatifs à partir de données capturées sur un réseau.

L'utilisation de Wireshark dans notre projet est pour la détection des vulnérabilités dans le réseau VoIP. Nous essayerons de capturer les paquets qui circulent pour déterminer quelques informations telles que les adresses IP, les numéros de ports, et d'autres informations qui servent au piratage (vol d'identité, déni de service, etc.). Ainsi que nous pouvons écouter une communication entre deux clients en décodant les paquets RTP (écoute clandestine).

IV.2.1.4. Inviteflood

Cet outil peut être utilisé pour inonder une cible avec INVITE il peut être utilisé pour cibler SIP passerelles/proxy et les softphone SIP.

IV.2.2.Simulation

IV.2.2.1. Attaque usurpation d'identité

C'est une attaque qui permettre de récupérer les informations nécessaire d'un utilisateur, entre autre son identifiant et son mot de passe.

Pour ce type d'attaque, nous allons utiliser l'outil **SIPVicious** en suivant ces étapes :

- scanner une plage d'adresse IP à la recherche de dispositifs VOIP avec **Svmap.**

```
root@kali:~# svmap 192.168.1.0/24
| SIP Device          | User Agent          | Fingerprint |
-----------------------------------------------------------
| 192.168.1.10:5060 | Asterisk PBX 14.7.6 | disabled    |
```

- déterminer la liste des extensions actives sur le serveur Asterisk avec **Svwar**.

```
root@kali:~# svwar -e4000-7000 -m invite 192.168.1.10
WARNING:TakeASip:using an INVITE scan on an endpoint (i.e. SIP phone) may cause
it to ring and wake up people in the middle of the night
| Extension | Authentication |
-----------------------------
| 6004      | reqauth        |
| 6002      | reqauth        |
| 6003      | reqauth        |
| 6001      | reqauth        |
```

- Cracker les mots de passes des utilisateurs avec **Svcrack**

Dans la pratique en utilise une attaque par dictionnaire des mots de passe enregistré dans le PC sous forme d'un fichier texte, il existe dans le net plusieurs fichiers qui sont près à l'emploi, permis eux celui du hacker **Stun** qui liste plus de 1.5 milliard mots de passe (15 GB d'espace).

Ce type d'attaque peut être inefficace avec des mots de passe longs et complexes, et cela peut prendre beaucoup de temps, voire des années, pour trouver le mot de passe. Mais si des mots simples et familiers sont utilisés, cette attaque peut fonctionner et nous donner un résultat dans un temps relativement court.

Dans notre cas et pour simuler l'attaque rapidement.

Soit en créant notre propre dictionnaire qui contient nos mots de passe qui sont des chiffres.

```
root@kali:~# svcrack -u6003  -d dictionary.txt 192.168.1.10
| Extension | Password |
------------------------
| 6003      | 1962     |
```

Soit en utilisant des options propres à la commande Svcrack qui fonctionnent avec des mots composés de nombres.

```
root@kali:~# svcrack -u6003 -r1-9999 -z4 192.168.1.10
| Extension | Password |
------------------------
| 6003      | 1962     |
```

IV.2.2.2. Attaque Eavesdropping

Pour réaliser cette attaque, nous utiliserons les outils **Ettercap** et **Wirshark**.

Tout d'abord, nous devons exécuter une attaque de type MITM par l'outil Ettercap pour rediriger le trafic vers le hacker en procédant comme suit:

- On lance Ettercap, Puis nous choisissons notre interface réseau pour lancer le Sniffing. Pour se faire on clique l'onglet Sniff puis en sélectionnant Unified sniffing (Figure)

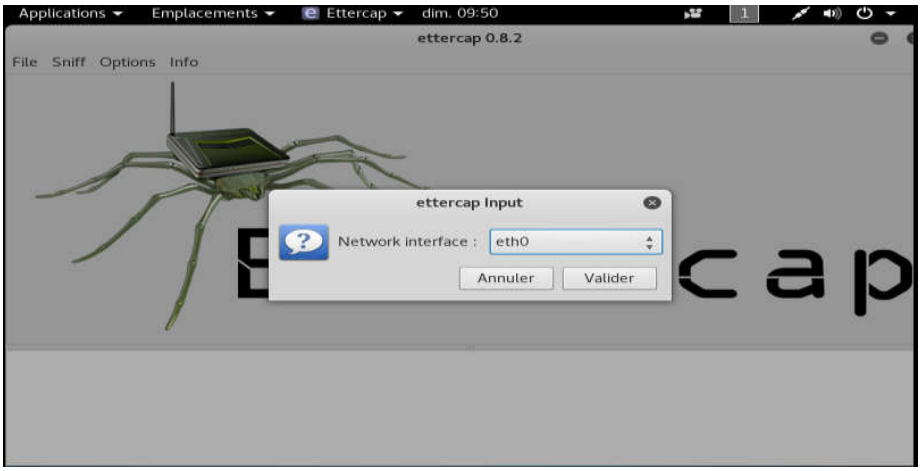


Figure 19: Lancement d'outil ETTERCAP et choix de l'interface

- Ensuite en scan le réseau (1-Scan for hosts) pour visualiser les machines connectés sur ce réseau (2-Hostes List) (Figure)

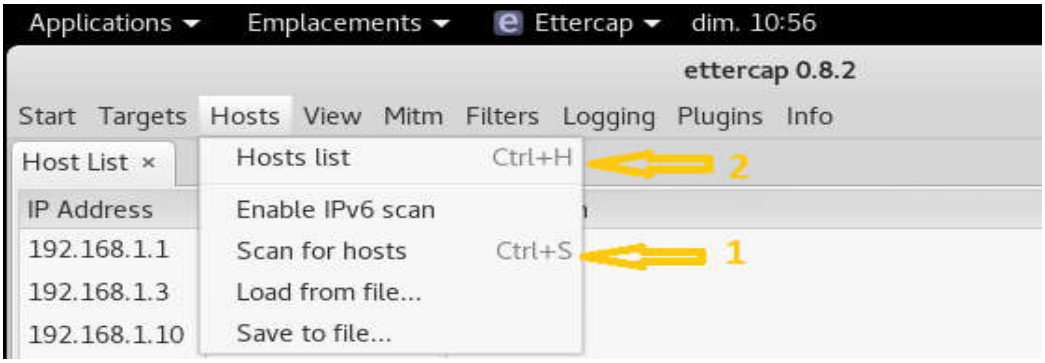


Figure 20: scans du réseau

- Après il est nécessaire de choisir le type d'attaque:

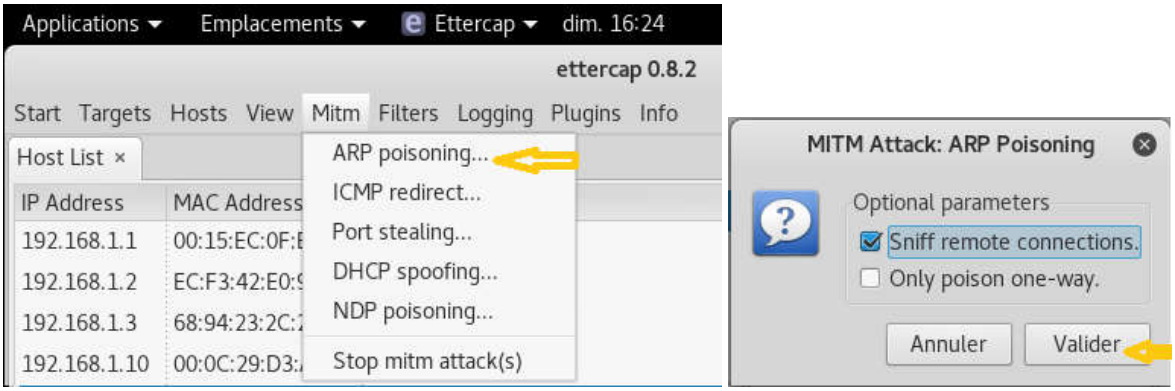


Figure 21: choix de type d'attaque

- Enfin, nous commençons le processus de sniffing.

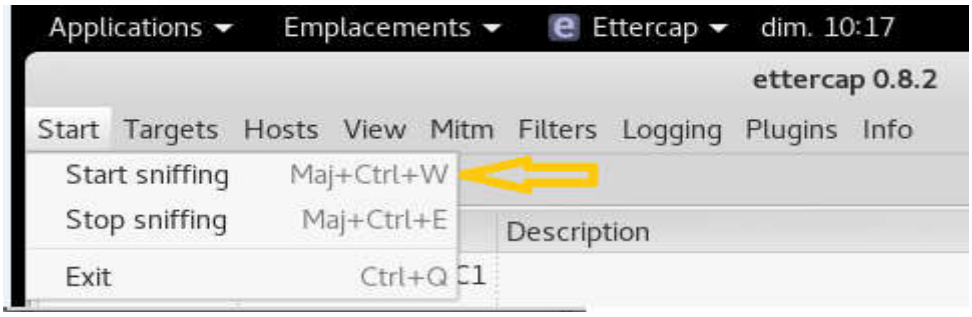
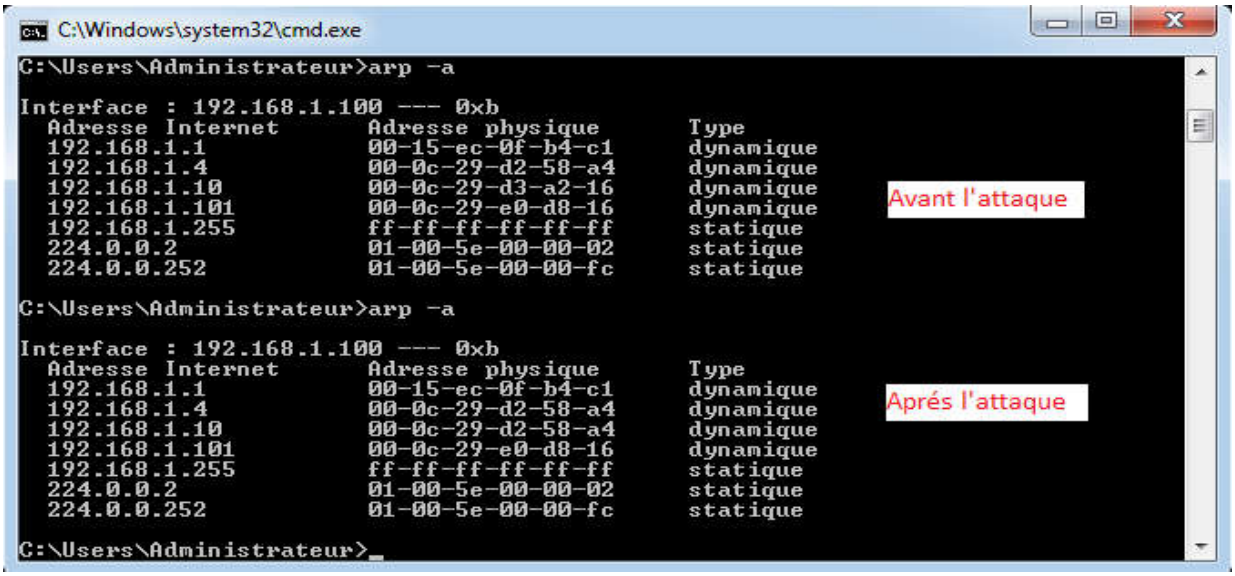


Figure 22: Commencement du sniffing

Après avoir effectué ces étapes l'architecture logique du réseau devient comme suit :

L'attaque Man in the Middle est réalisée et la machine hacker identifiée par l'IP 192.168.1.4 est placée entre le serveur Asterisk identifié par l'adresse IP 192.168.1.10 et la machine virtuel identifiée par l'adresse IP 192.168.1.100.

Le Pirate à l'adresse MAC 00-0c-29-d2-58-a4 et le serveur Asterisk à l'adresse MAC 00-0c-29-d3-a2-16, la table adresse MAC chez le client 192.168.1.100 avant et après l'attaque ressemble donc à cela :



Donc, la table ARP de notre cible est falsifiée, il va former ces trames avec l'adresse IP du serveur, mais finira par les envoyer au pirate car il formera ses requêtes avec comme adresse MAC de destination celle du pirate.

Maintenant il vient le rôle de **Wireshark** pour sniffer le trafic et capturer les paquets échangés, pour ce :

- Après le lancement de Wireshark, nous sélectionnons l'interface réseau sur laquelle nous allons recevoir les paquets échangés.

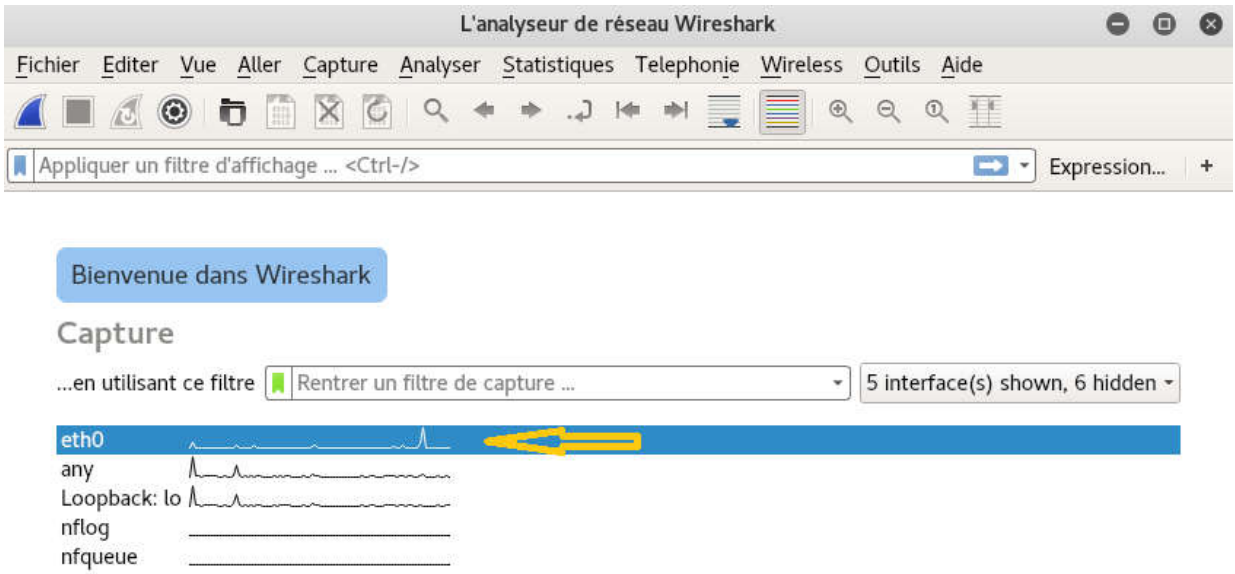


Figure 4-5: Lancement d'outil Wireshark et choix de l'interface

- En filtrer les paquets à capturer et se limiter au protocole « RTP »

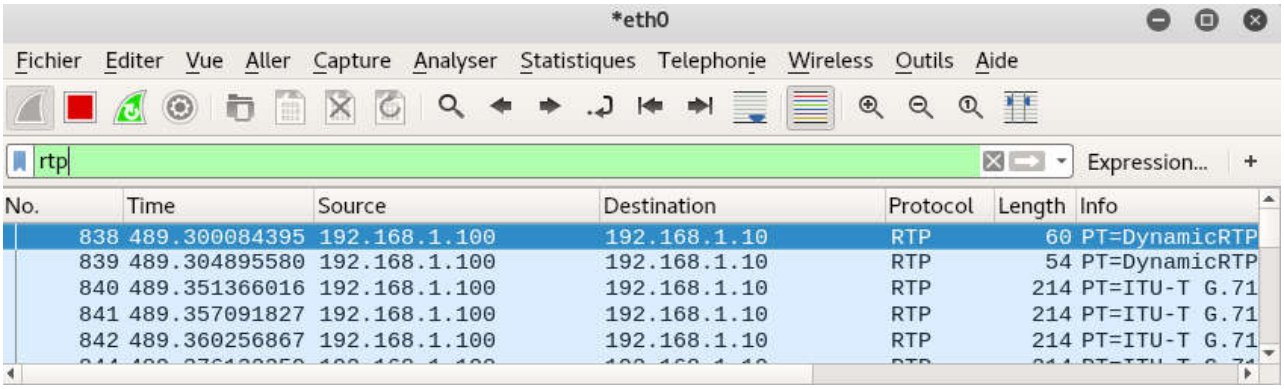


Figure 4-6: choix du protocole RTP dans Wireshark

- Nous allons maintenant analyser les paquets capturés et les enregistrer sous forme de fichier audio pour une lecture ultérieure par un lecteur multimédia.

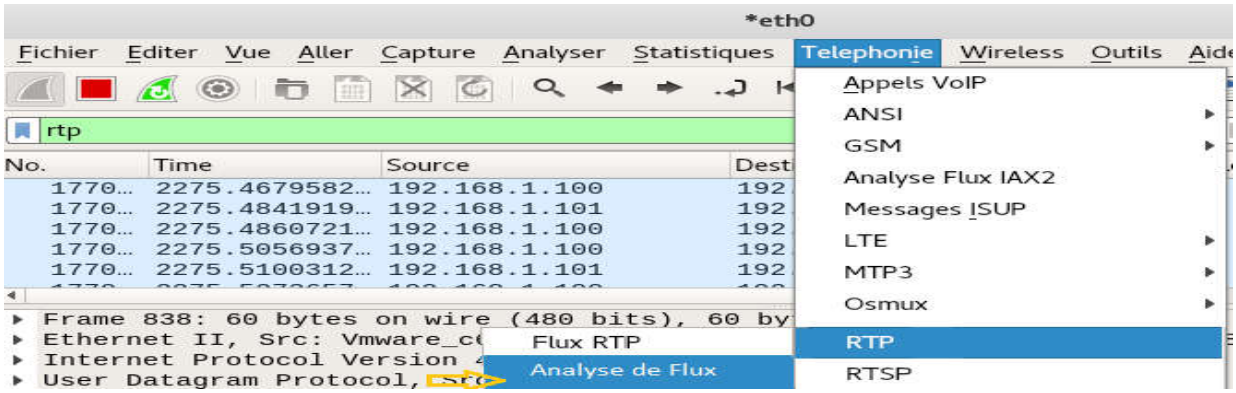


Figure 4-7: analyses de flux RTP par Wireshark

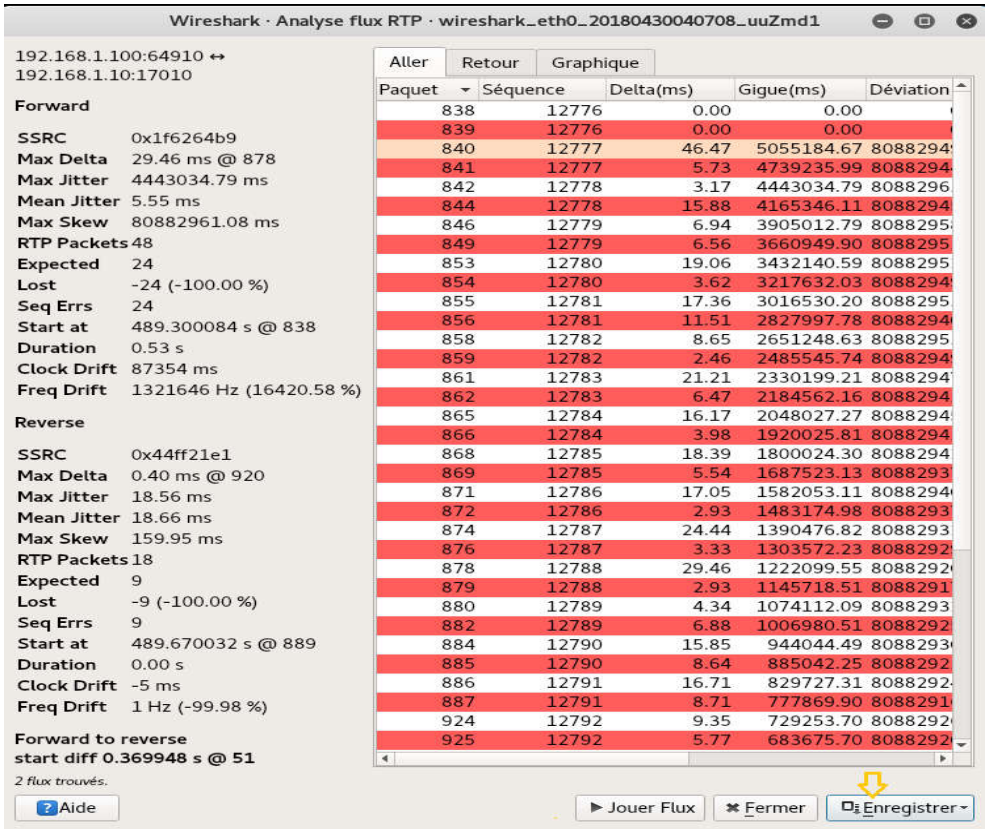


Figure 4-8: enregistrements de flux RTP analysé par Wireshark

IV.2.2.3. Attaque Denis de service (Dos)

Pour ce type d'attaque, nous utiliserons l'outil **Inviteflood**

La formule d'utilisation est:

inviteflood eth0 target_extension target_domain target_ip number_of_packets

```
root@kali:~# inviteflood eth0 6001 192.168.1.10 192.168.1.10 100000000

inviteflood - Version 2.0
              June 09, 2006

source IPv4 addr:port    = 192.168.1.4:9
dest   IPv4 addr:port    = 192.168.1.10:5060
targeted UA              = 6001@192.168.1.10

Flooding destination with 100000000packets
sent: 100000000
```

On a lancé le ping à partir d'une machine client identifiée par l'adresse IP 192.168.1.2 et on a constaté que :

Le serveur Asterisk répond à la requête de ping, une fois l'attaque lancée, on remarque une perturbation dans le réseau du serveur et après un certain temps ce dernier sera hors service.

Lorsqu'on a vérifié le serveur, on a un dysfonctionnement total.

IV.3. Choix et implémentation des bonnes pratiques

Pour protéger contre les attaques, et même d'autres attaques similaires, nous avons choisi une gamme de solutions qui peuvent aider à réduire les menaces, nous ne pouvons pas dire que les solutions proposées et mises en œuvre sont efficaces car il y a toujours des problèmes de sécurité

IV.3.1. Bonne pratique contre l'attaque usurpation d'identité

IV.3.1.1. Création d'un mot de passe crypté

Une meilleure pratique pour mieux assurer la sécurité du serveur Asterisk est de crypter le mot de passe du client à l'aide de la méthode MD5. Grâce au cryptage, le mot de passe du client devient illisible dans le cas où une personne malveillante accède au fichier sip.conf.

Et cela est effectué grâce à la commande suivante appliquée sur chaque client :

echo -n "utilisateur:Asterisk:motdepasse" | md5sum

```
[ymezhoudi@localhost ~]$ echo -n "6001:Asterisk:1976" | md5sum
88962f3de6c2d17138cc4ac047ff40b5  -
[ymezhoudi@localhost ~]$ echo -n "6002:Asterisk:2018" | md5sum
cdee17f1dc0c85c60ab338ee3ff35bdf  -
[ymezhoudi@localhost ~]$ echo -n "6003:Asterisk:1962" | md5sum
8fd8761f7cb7b7c64389b2ec7f0bcd65  -
[ymezhoudi@localhost ~]$ echo -n "6004:Asterisk:1954" | md5sum
8193975cbc0eefd4b89bfa3235a97991  -
```

Des modifications seront nécessaires dans sip.conf pour que la configuration soit fonctionnelle. En remplacent la ligne secret par md5secret

```
[6001]
type=friend
host=dynamic
dtmfmode=rfc2833
disallow=all
allow=ulaw
fullname =Yazid MEZHOUDI
username = myazid
md5secret=88962f3de6c2d17138cc4ac047ff40b5
context = work
```

IV.3.1.2. Utilisation de l'outil « Fail 2 Ban »

Fail 2 Ban est un outil propre aux systèmes Linux, permettant de se protéger contre les attaques de brute force ayant pour but de permettre à un attaquant de s'authentifier.

Il est possible de configurer Fail2Ban pour qu'il protège Asterisk.

1. Fonctionnement

Fail2Ban n'est donc pas un outil propre à Asterisk. Il est utilisé pour se protéger contre les attaques de brute force d'authentification (SSH, Apache, FTP, etc…).

Dans le cas d'Asterisk, Fail2Ban va analyser les logs d'Asterisk, à la recherche de tentatives de connexions échouées.

Si Fail2Ban détecte plus d'un certain nombre de connexions échouées, il va bloquer l'IP source du client qui tente de se connecter.

Le blocage se fait à l'aide d'une règle IP table.

Il est bien entendu possible de paramétrer le nombre de connexion menant à un blocage. De même qu'il est possible de paramétrer le temps de blocage, ou l'intervalle de temps de recherche de Fail2Ban. Par exemple : bloquer un client pendant 1h après 5 tentatives d'authentifications manquées en 10 minutes.

La configuration va se faire dans les deux fichiers suivants :

- /etc/fail2ban/filter.d/asterisk.conf
- /etc/fail2ban/jail.conf

De plus, le fichier de log d'Asterisk sera utilisé pour chercher les tentatives de connexion.

Le fichier asterisk.conf correspond au filtre de Fail2Ban pour Asterisk. Il permet de définir les logs qui correspondent à des erreurs de connexion.

Le fichier jail.conf permet de dire à Fail2Ban d'analyser le fichier de log d'Asterisk à l'aide des filtres définis dans asterisk.conf.

2. installation et Configuration

- Installation :

Tout d'abord, il nous faut installer Fail2Ban.

Pour cela, entrer la commande suivante :

[root@localhost ymezhoudi]# yum install fail2ban
Installer fail2ban

Ensuite, il faut installer iptables.

```
[root@localhost ymezhoudi]#  yum install iptables
```

Installer iptables

- Configuration :

La première chose à faire est de vérifier le fichier /etc/fail2ban/filter.d/asterisk.conf, celui qui apprend à Fail2ban ce qu'il faut surveiller dans les fichiers journaux Asterisk.

```
[INCLUDES]

# Read common prefixes. If any customizations available -- read them from
# common.local
before = common.conf

[Definition]

_daemon = asterisk

__pid_re = (?:\[\d+\])

iso8601 = \d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}\.\d+[+-]\d{4}

# All Asterisk log messages begin like this:
log_prefix= (?:NOTICE|SECURITY|WARNING)%(__pid_re)s:?(?:\[C-[\da-f]*\])? [^:]+:\d*(?:(?: in)? \w+:)?

failregex = ^%(__prefix_line)s%(log_prefix)s Registration from '[^']*' failed for '<HOST>(:\d+)?' - (Wrong
password|Username/auth name mismatch|No matching peer found|Not a local domain|Device does not match
ACL|Peer is not supposed to register|ACL error \(permit/deny\)|Not a local domain)$
^%(__prefix_line)s%(log_prefix)s Call from '[^']*' \(<HOST>:\d+\) to extension '[^']*' rejected because extension
not found in context
^%(__prefix_line)s%(log_prefix)s Host <HOST> failed to authenticate as '[^']*'$
^%(__prefix_line)s%(log_prefix)s <HOST> failed to authenticate as '[^']*'$
^%(__prefix_line)s%(log_prefix)s No registration for peer '[^']*' \(from <HOST>\)$
^%(__prefix_line)s%(log_prefix)s Host <HOST> failed MD5 authentication for '[^']*' \([^)]+\)$
^%(__prefix_line)s%(log_prefix)s Failed to authenticate (user|device) [^@]+@<HOST>\S*$
^%(__prefix_line)s%(log_prefix)s hacking attempt detected '<HOST>'$
^%(__prefix_line)s%(log_prefix)s
SecurityEvent="(FailedACL|InvalidAccountID|ChallengeResponseFailed|InvalidPassword)",EventTV="([\d-
]+|%(iso8601)s)",Severity="[\w]+",Service="[\w]+",EventVersion="\d+",AccountID="(\d*|<unknown>)",SessionID
=".+",LocalAddress="IPV[46]/(UDP|TCP|WS)/[\da-fA-
F:.]+/\d+",RemoteAddress="IPV[46]/(UDP|TCP|WS)/<HOST>/\d+"(,Challenge="[\w/]+")?(,ReceivedChallenge="\
w+")?(,Response="\w+",ExpectedResponse="\w*")?(,ReceivedHash="[\da-f]+")?(,ACLName="\w+")?$
^%(__prefix_line)s%(log_prefix)s "Rejecting unknown SIP connection from <HOST>"$
^%(__prefix_line)s%(log_prefix)s Request (?:'[^']*' )?from '[^']*' failed for '<HOST>(?::\d+)?'\s\(callid: [^\)]*\) -
(?:No matching endpoint found|Not match Endpoint(?: Contact)? ACL|(?:Failed|Error) to authenticate)\s*$
ignoreregex =
```

Configuration /etc/fail2ban/filter.d/asterisk.conf

Le fichier de configuration suivant qui doit être vérifier est le fichier de configuration de journalisation Asterisk(/etc/asterisk/logger.conf). Au moins la dateformat et les messages sont définis, car ceux-ci sont requis pour Fail2ban:

```
[general]
dateformat = %F %T
[logfiles]
console => notice,warning,error,debug
messages => notice,warning,error
```

En passe à la création de la prison (jail), celle-ci a pour but de faire le lien entre le filtre et le fichier de log. C'est aussi elle qui définit les paramètres du blocage.

Assurez-vous de configurée /etc/fail2ban/jail.local comme ceci:

```
[asterisk]
filter = asterisk
enabled = true
port    = 5060,5061
action  = %(banaction)s[name=%(__name__)s-tcp, port="%(port)s", protocol="tcp",
chain="%(chain)s", actname=%(banaction)s-tcp]
        %(banaction)s[name=%(__name__)s-udp, port="%(port)s", protocol="udp",
chain="%(chain)s", actname=%(banaction)s-udp]
        %(mta)s-whois[name=%(__name__)s, dest="%(destemail)s"]
logpath  = /var/log/asterisk/messages
maxretry = 5
findtime = 3600
bantime = 120
```

Configuration /etc/fail2ban/jail.local

Cette prison définit un maximum de 5 tentatives manquées dans l'espace de 2 minutes. Au-delà, l'utilisateur est bloqué pour 1 heure.

IV.3.2. Bonne pratique contre l'attaque Eaveasdropping

Pour sécuriser les appels, il faut chiffrer deux choses :

- Le trafic SIP

- Le trafic RTP

IV.3.2.1. Chiffrement du trafic SIP avec TLS

Transport Layer Security (TLS) fournit un cryptage pour la signalisation d'appel. C'est un moyen pratique d'empêcher les personnes qui ne sont pas Autorisé de savoir qui vous appelez. La configuration de TLS entre Asterisk et un client SIP implique la création de fichiers clés, la modification de la configuration SIP d'Asterisk pour activer TLS, la modification du client SIP pour se connecter à Asterisk via TLS.

- Générer les clés :

1. D'abord, faisons une place à nos clés.

```
[root@localhost ymezhoudi]# mkdir /etc/asterisk/keys
```

Création fishier keys

2. Ensuite, nous utilisons le script "ast_tls_cert" dans le répertoire source " asterisk-14.7.6/contrib / scripts" d'Asterisk pour créer une autorité de certification auto-signée et un certificat Asterisk.

```
[root@localhost scripts]# ./ast_tls_cert -C 192.168.1.10  -O "BISKRA UNIV" -d /etc/asterisk/keys
```

Création une autorité de certification

- C : permet de spécifier le nom d'hôte ou l'adresse IP du serveur Asterisk.

- O : permet de définir le nom de l'organisation

- d : permet de spécifier le dossier de sortie

A l'exécution du script,

1/ Il vous sera demandé d'entrer un mot de passe pour /etc/asterisk/keys/ca.key.

2/ Ceci créera le fichier /etc/asterisk/keys/ca.crt.

3/ Vous serez invité à entrer à nouveau le mot de passe, puis le fichier /etc/asterisk/keys/asterisk.key sera créé.

4/ Le fichier /etc/asterisk/keys/asterisk.crt sera généré automatiquement.

5/ Il vous sera demandé d'entrer le mot de passe une troisième fois, et le fichier /etc/asterisk/keys/asterisk.pem sera créé, une combinaison de Fichiers asterisk.key et asterisk.crt.

Les fichiers suivants devraient être créés :

```
[root@localhost keys]# ls
asterisk.crt  asterisk.csr  asterisk.key  asterisk.pem  ca.cfg  ca.crt  ca.key  tmp.cfg
```

3. Et puis nous allons créer des clés et des certificats pour les clients.

[root@localhost scripts]#./ast_tls_cert -m client –c /etc/asterisk/keys/ca.crt -k /etc/asterisk/keys/ca.key -C 192.168.1.100 -O " BISKRA UNIV" –d /etc/asterisk/keys/ -o mezhoudi
Création clé et certificat pour un client

Voici le détail des options :

- **m** : indique qu'il faut créer un certificat client

- **c** : permet de spécifier le chemin vers le certificat de l'autorité de certificat

- **k** : permet de spécifier le chemin vers la clé privée de l'autorité de certificat

- **C** : permet de spécifier le nom d'hôte du poste du client ou l'adresse IP

- **O** : permet de définir le nom de l'organisation

- **d** : permet de spécifier le dossier de sortie

- **o** : permet de choisir le nom de la clé à créer

L'opération est à répéter pour tous les clients devant bénéficier de TLS.

- La configuration SIP d'Asterisk

Maintenant, configurons Asterisk pour utiliser TLS.

Dans le fichier de configuration sip.conf, nous définissons les éléments suivants:

```
[general]
tlsenable=yes
tlsbindaddr=0.0.0.0
tlscertfile=/etc/asterisk/keys/asterisk.pem
tlscafile=/etc/asterisk/keys/ca.crt
tlscipher=ALL
tlsclientmethod=tlsv1
```

Ensuite, nous devons autoriser les clients à utiliser TLS, toujours dans le fichier SIP.conf, on ajoute la ligne transport=tls pour tous les clients. Voici un exemple:

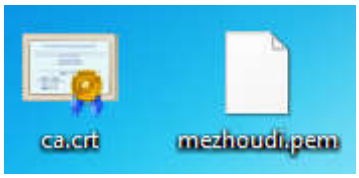
```
[6001]
type=friend
host=dynamic
dtmfmode=rfc2833
disallow=all
allow=ulaw
fullname =Yazid MEZHOUDI
username = myazid
secret=1976
context = work
transport=tls
```

Et enfin, nous pouvons configurer les postes des clients.

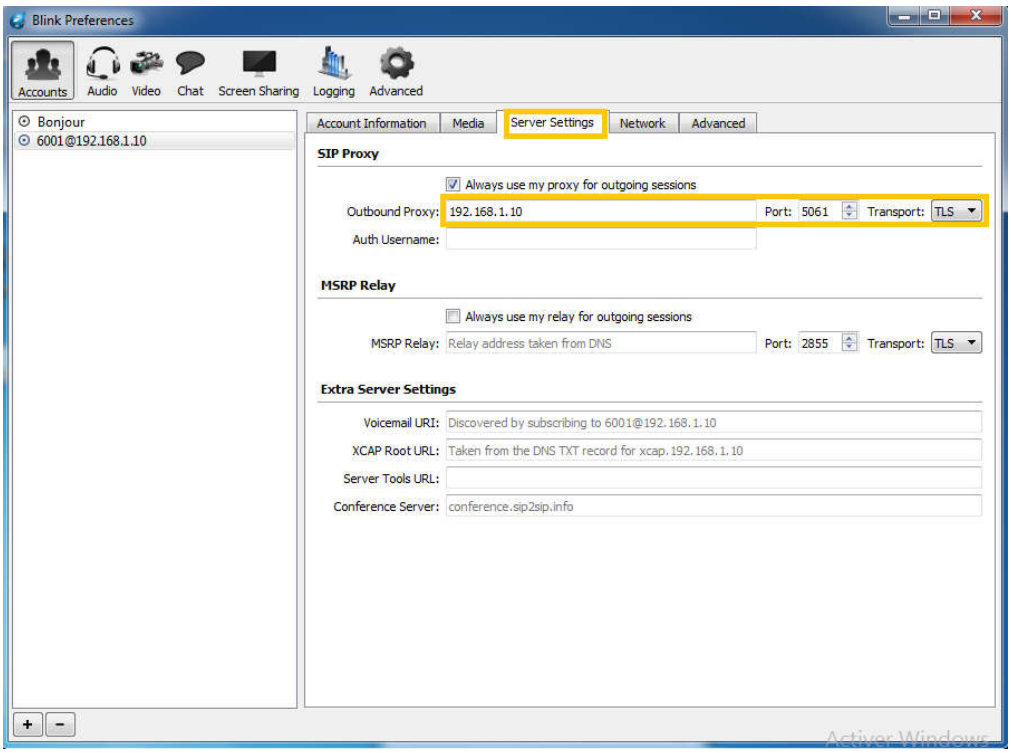
Pour cette démonstration, on a utilisé le softphone Blink.

Chaque client doit posséder 2 fichiers : ca.crt et client.pem

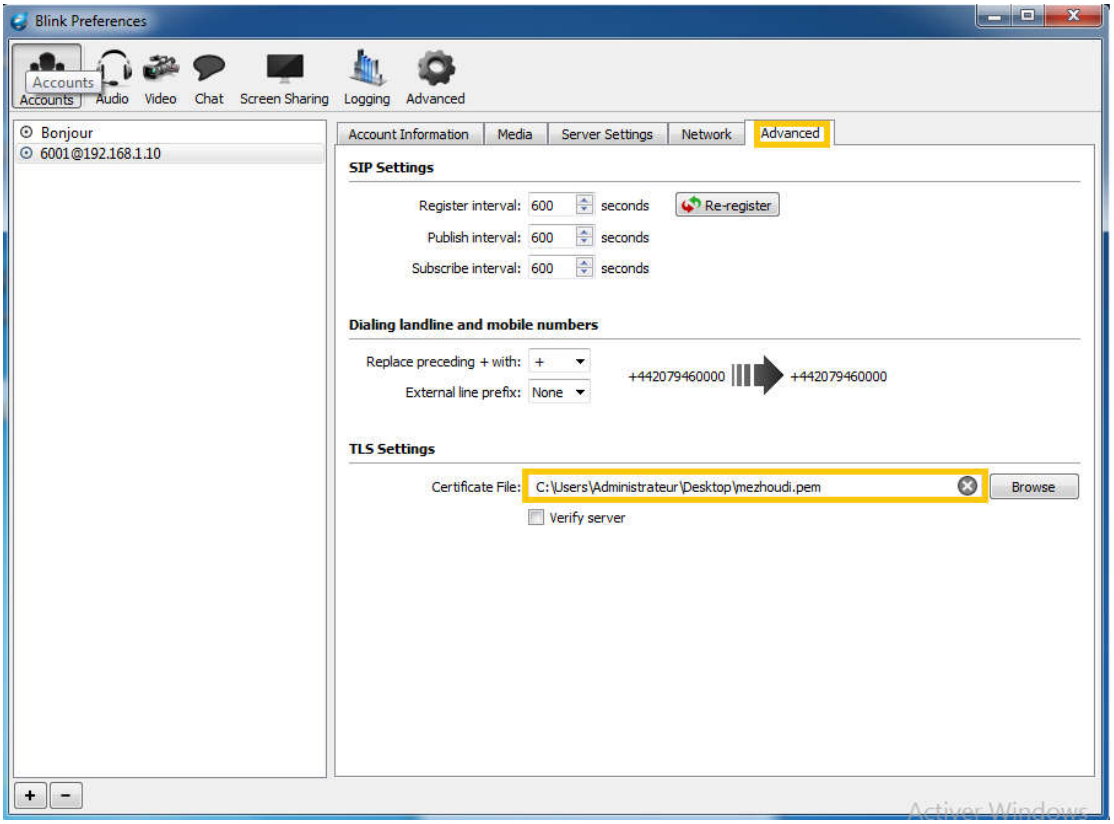
Client 6001 :

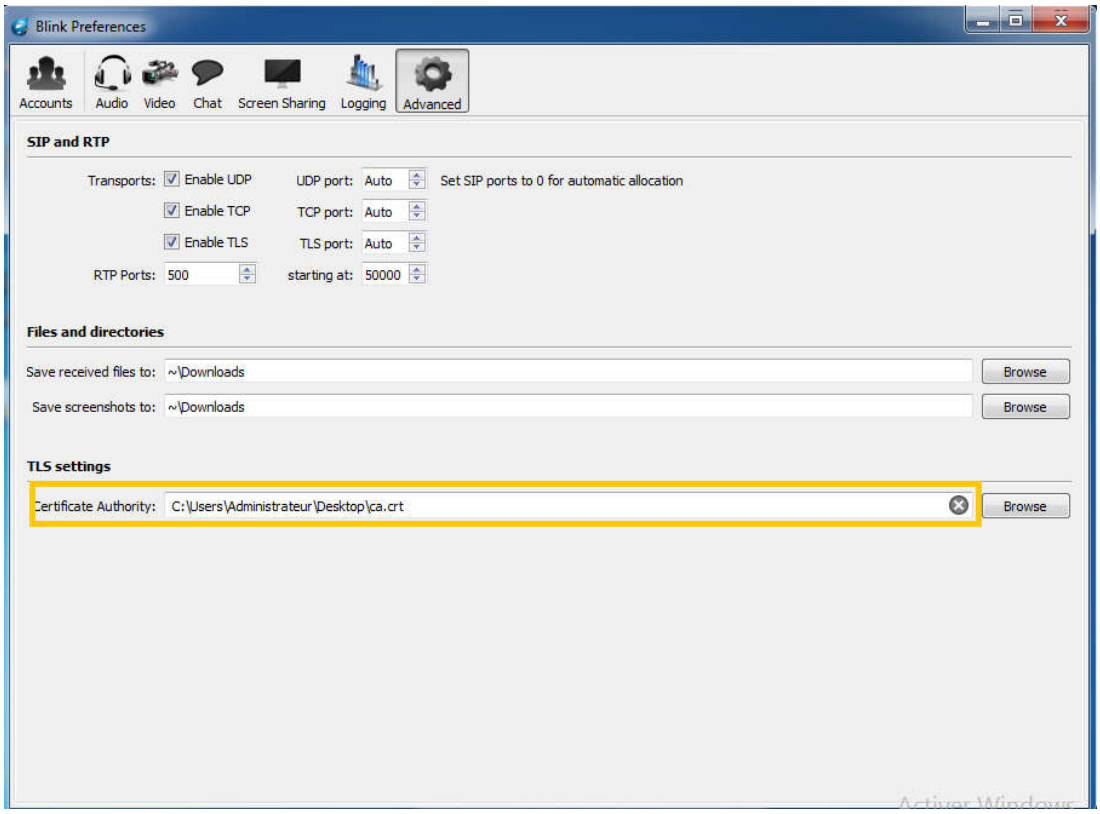


Nous devons modifier les préférences de compte, et sous les paramètres SIP, nous devons définir le proxy sortant pour se connecter au port 5061 avec type de transport TLS sur notre serveur Asterisk



Renseigner le chemin des fichiers (Certificat et Clé du client)

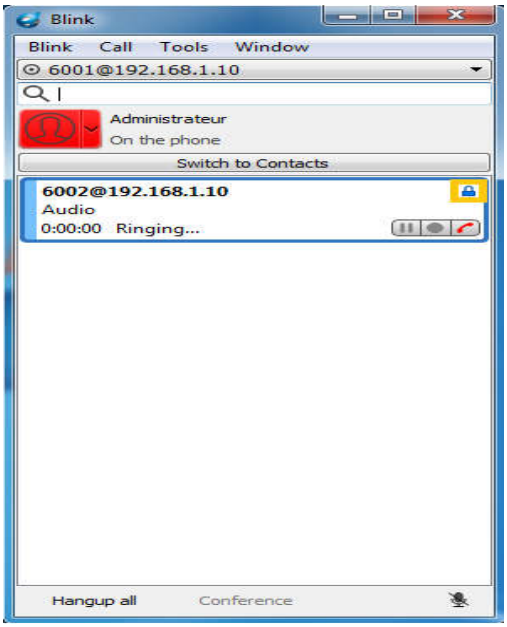




Dans le serveur Asterisk, on peut vérifier si le TLS était bien configuré :

```
*CLI> Reloading SIP
  == Parsing '/etc/asterisk/sip.conf': Found
  == Parsing '/etc/asterisk/users.conf': Found
  == Using SIP CoS mark 4
  == TLS/SSL ECDH initialized (automatic), faster PFS ciphers enabled
  == TLS/SSL certificate ok
  == Parsing '/etc/asterisk/sip_notify.conf': Found
```

Pour assurer le chiffrement de la signalisation de l'appel un cadenas bleu apparaitre dans le softphone Blink



Après une configuration réussie de SIP / TLS, Asterisk doit écouter sur le nouveau port dédié pour les connexions TLS 5061(sip-tls) :

```
[root@localhost ymezhoudi]# netstat
Connexions Internet actives (sans serveurs)
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat
tcp        0      0 localhost.local:sip-tls 192.168.1.100:49457     ESTABLISHED
```

IV.3.2.2. Chiffrement du trafic RTP avec SRTP

La prise en charge de SRTP est fournie par libsrtp.

La première étape consistera à télécharger et installer les bibliothèques SRTP

[root@localhost ymezhoudi]# yum install libsrtp libsrtp-devel
installer les bibliothèques SRTP

Nous devons aussi assurer que le module SRTP est sélectionné dans l'Asterisk, pour ce faire, nous devons aller dans le menuselect

[root@localhost asterisk-14.7.6]# make menuselect
Sélection des modules

Aller à la section modules de ressources et vérifier que res_srtp est sélectionné.

Pour qu'Asterisk prenne en charge SRTP, il nous faut le réinstaller.

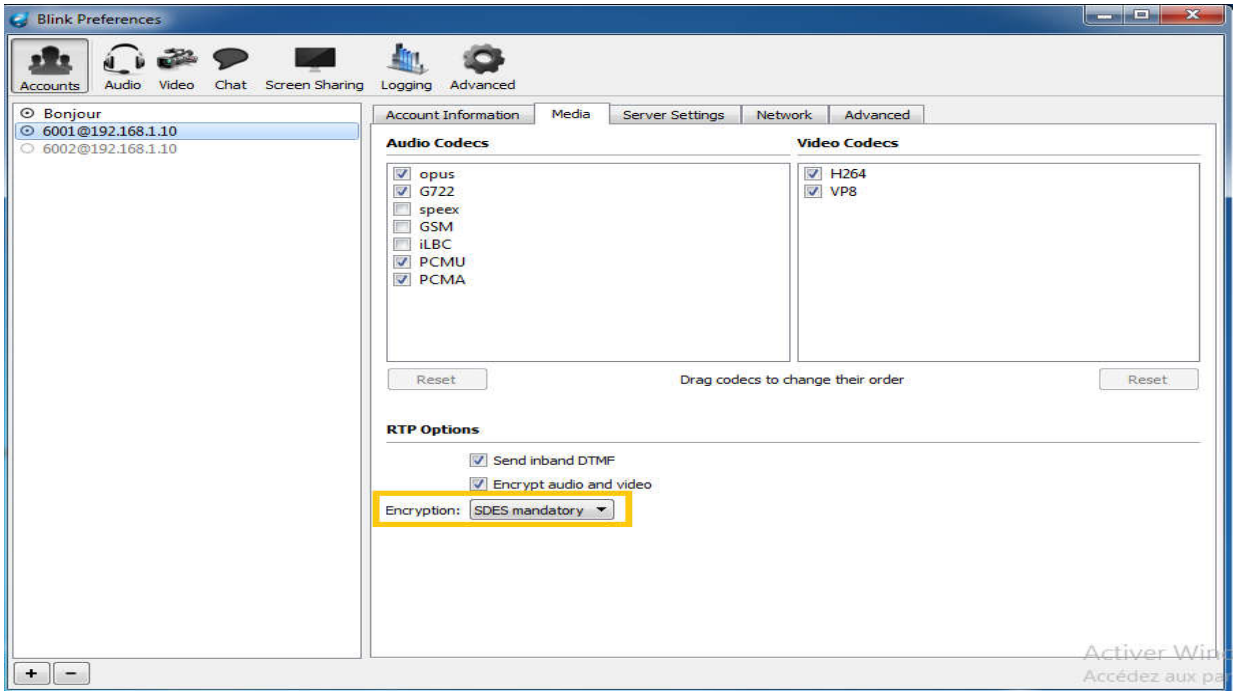
[root@localhost ymezhoudi]# cd asterisk-14.7.6 **[root@localhost asterisk-14.7.6]# ./configure--libdir=/usr/lib64**
lancement du script de configuration

[root@localhost asterisk-14.7.6]# make && make install **[root@localhost asterisk-14.7.6]# make samples && make config**
compilation et installation d'Asterisk

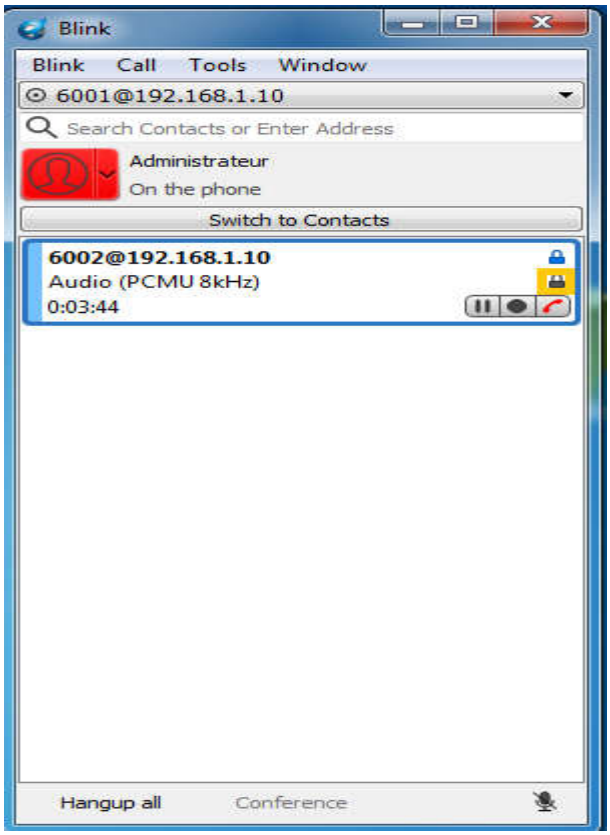
Finalement Il faut forcer l'utilisation de SRTP sur les clients voulus. Pour cela, ajouter la ligne encryption=yes chez les utilisateurs concernés dans SIP.conf.

```
[6001]
type=friend
host=dynamic
dtmfmode=rfc2833
disallow=all
allow=ulaw
fullname =Yazid MEZHOUDI
username = myazid
secret=1976
context = work
transport=tls
encryption=yes
```

Et en configure softophone Blink de poste client pour qu'il utilise SRTP.



Dans la fenêtre de Blink, un cadenas noir devrait apparaitre.



IV.3.3.Autre aspect de Sécurité

IV.3.3.1. Implémentation d'un firewall

Dans le cadre de notre projet le firewall va nous permettre de minimiser le trafic entrant au serveur Asterisk est cela pour limiter les attaques de types DoS. En effet notre objectif est de ne laisser passer que le trafic VoIP et plus exactement les paquets basés sur le protocole SIP et le protocole RTP, qui sont utilisés par notre serveur Asterisk pour le trafic VoIP.

```
[root@localhost ymezhoudi]#  systemctl enable firewalld
```

Activation de firewall

```
[root@localhost ymezhoudi]#  firewall-cmd --zone=public --add-port=5060/udp --permanent
[root@localhost ymezhoudi]#  firewall-cmd --zone=public --add-port=5060/tcp --permanent
[root@localhost ymezhoudi]# firewall-cmd --zone=public --add-port=5061/udp --permanent
[root@localhost ymezhoudi]#  firewall-cmd --zone=public --add-port=5061/tcp --permanent
[root@localhost ymezhoudi]#   firewall-cmd --zone=public --add-port=4569/udp --permanent
[root@localhost ymezhoudi]#  firewall-cmd --zone=public --add-port=5038/tcp --permanent
[root@localhost ymezhoudi]#  firewall-cmd --zone=public --add-port=10000-20000/udp --permanent
[root@localhost ymezhoudi]#  firewall-cmd --reload
```

Permission d'accéder aux ports utilisés par asterisk

IV.3.3.2. Exécuter Asterisk sous un utilisateur non privilégié

Parmi les bonnes pratiques pour sécuriser notre serveur Asterisk est de changer l'utilisateur sur lequel Asterisk tourne. Le principal objectif de cette sécurisation est si le serveur Asterisk est compromis au niveau de sa sécurité ceci ne doit en aucun cas affecter toute la machine sur laquelle tourne le serveur. Idéalement, la compromission ne devrait pas permettre d'éditer les fichiers de configuration. Voici les étapes à suivre pour ce changement :

Tous d'abord il faut arrêter le service Asterisk ; Ensuit en va ajouter un utilisateur **asterisk**, et de le définir comme propriétaire des fichiers et dossiers de l'installation

```
[root@localhost asterisk-14.7.6]# useradd -m asterisk
```

Ajouter un utilisateur asterisk

```
[root@localhost asterisk-14.7.6]# chown  -R asterisk.asterisk /var/run/asterisk
[root@localhost asterisk-14.7.6]# chown -R asterisk.asterisk /etc/asterisk
[root@localhost asterisk-14.7.6]# chown -R asterisk.asterisk /var/{lib,log,spool}/asterisk
[root@localhost asterisk-14.7.6]# chown -R asterisk.asterisk /usr/lib64/asterisk
```

Définir l'utilisateur asterisk comme propriétaire des fichiers et dossiers de l'installation

Maintenant il faut redémarrer Asterisk.

IV.3.3.3. Implémentation des ACLs

Limitez les adresses IP sources. Le fichier de configuration sip.conf d'Asterisk peut contenir des règles d'accès (ACLs avec "permit" et "deny") pour restreindre les clients qui sont autorisés à envoyer des messages SIP INVITE.

Syntax: permit=<ip addresse>/<network mask>

deny=<ip addresse>/<network mask>

Par exemple:

```
[6001]
type=friend
host=dynamic
dtmfmode=rfc2833
disallow=all
allow=ulaw
fullname =Yazid MEZHOUDI
username = myazid
secret=1976
context = work
transport=tls
deny=0.0.0.0/0.0.0.0
permit=192.168.1.102/255.255.255.255
```

Refuser toutes les adresses IP sauf pour l'adresse IP 192.168.102 est autorisé.

En vérifiant fichiers log Asterisk, que visualisent les messages similaires à ceci:

```
[2018-05-21:21:05:42] NOTICE[1892]: acl.c:748 ast_apply_acl: SIP Peer ACL: Rejecting '192.168.1.108' due to a failure to pass ACL '(BASELINE)'
[2018-05-21:21:05:42] NOTICE[1892]: chan_sip.c:28172 handle_request_register: Registration from '"Test1"<sip:6001@192.168.1.10>' failed for '19
2.168.1.108:7944' - Device does not match ACL
```

IV.4. Conclusion

Tout au long de ce chapitre, nous avons simulés des nombreuses attaques et des tests afin de découvrir les failles et les vulnérabilités possibles dans ce système de communication et les

mesures de sécurité à prendre pour les éviter. Mais il faut savoir qu'il est impossible d'obtenir une sécurité totale dans le réseau VoIP et généralement sur tous les réseaux.

Conclusion Général

La VoIP est une technologie émergente que de nombreuses entreprises tentent de l'exploiter en raison des avantages qu'elles offrent.

En Algérie, cette technologie n'est pas encore très développée, compte tenu de l'absence de fournisseurs VoIP. Cependant, il est possible de déployer certaines applications de cette technologie au sein des entreprises multi-sites (par exemple l'entreprise national des services aux puits ENSP, qui est doter d'une solution VOIP Cisco), ce qui permettra de migrer les communications du réseau traditionnel RTC vers le réseau IP.

Dans le premier chapitre, nous nous intéressons à l'étude de cette technologie avec ses différents protocoles et standards.

Au deuxième chapitre, nous avons installé et configuré une solution VoIP, en utilisant comme serveur un IPBX Asterisk et en tant que clients, nous avons utilisé des différents types de Softphones.

Comme dans le troisième chapitre, nous avons étudié les problèmes de sécurité de la VoIP, les attaques, les vulnérabilités à différents niveaux et les bonnes pratiques possibles pour les attaques mentionnées.

Et dans le dernier chapitre, nous avons fait des attaques contre la solution installée, puis nous avons proposé et mis en place des mécanismes et des protocoles pour la sécuriser.

En conclusion, nous avons vu que la solution open-source proposée par Asterisk nous permet de bénéficier librement des services d'un standard téléphonique moderne et intégré pouvant être exploité par les entreprises pour les appels téléphoniques internes en utilisant le réseau informatique interne existant, Pour les appels externes dont nous n'avons pas discuté dans notre recherche, des équipements spéciaux devraient être mis en place pour connecter ce système aux fournisseurs de téléphonie traditionnels. Néanmoins, il est indispensable de sécurisé le maximum cette installation avec des différents mécanismes et outillés.

Bibliographie

[1] C. Servin, RÉSEAUX ET TÉLÉCOMS, Paris: DUNOD, 2003.

[2] O. Laurent et G. Pujolle, Téléphonie sur IP <<2e édition>>, Paris: Eyrolles, 2011.

[3] P. Ledru, Téléphonie sur IP, Eni, 2011.

[4] A. BENCHIKH et K. MECHERNENE, Etude de la sécurité dans la VOIP,Master, Tlemcen: Université Abou Bakr Belkaid, 2015.

[5] R.BOUZIDA, Etude et mise en place d'une solution VoIP sécurisée, Master Professionnel, Tunis, Université virtuelle de Tunis, 2011.

[6] «H.323 Architecture et Protocoles,» 2008. [En ligne]. Disponible sur le lien: http://www.efort.com/r_tutoriels/H323_EFORT.pdf . [Accès le 24 02 2018].

[7] «Le Protocole SIP Avancé et ses Extensions,» 2011. [En ligne]. Disponible sur le lien: http:// www.efort.com/r_tutoriels/SIP2_EFORT.pdf . [Accès le 26 02 2018].

[8] R. m. perea, Internet Multimedia Communication Using SIP, Burlington: Morgan Kaufmann, 2008.

[9] «Asterisk Administrator Guide,» . [En ligne]. Disponible sur le lien: https://wiki.asterisk.org/wiki/download/attachments/19005471/Asterisk-Admin-Guide-14.pdf?version=1&modificationDate=1469484505517&api=v2 . [Accès le 21 04 2018].

[10] «VoIP Hacking Techniques,» . [En ligne]. Disponible sur le lien: https://hakin9.org/voip-hacking-techniques/ . [Accès le 26 04 2018].

[11] «Sécurité des réseaux IPSec,» . [En ligne]. Disponible sur le lien: http://dept-info.labri.fr/~guermouc/SR/SR/cours/cours3.pdf . [Accès le 29 04 2018].

[12] «SSL and TLS An Overview of A Secure Communications Protocol,» . [En ligne]. Disponible sur le lien: http://horms.net/projects/ssl_and_tls/stuff/ssl_and_tls.pdf .[Accès le 02 05 2018].

[13] D.Endler et M.Collier, Hacking VoIP Exposed

[14] « Kali Linux,». [En ligne]. Disponible sur le lien: https://www.kali.org/ . [Accès le 04 05 2018].

[15] «L'outil sipviciouse,». [En ligne]. Disponible sur le lien: https://github.com/EnableSecurity/sipvicious / . [Accès le 09 05 2018].

[16] «MITM avec ettercap sous Kali Linux,» . [En ligne]. Disponible sur le lien: https://www.supinfo.com/articles/single/1569-tutomitm-avec-ettercap-kali-linux-105 . [Accès le 13 05 2018].

[17] « L'outil Wireshark,» . [En ligne]. Disponible sur le lien: Http://www.wireshark.org . [Accès le 16 05 2018].

[18] «Kalli tools ,inviteflood,» . [En ligne]. Disponible sur le lien: https://tools.kali.org/sniffingspoofing/inviteflood . [Accès le 19 05 2018].

[19] «La solution Fail2Ban,». [En ligne]. Disponible sur le lien: http://www.fail2ban.org . [Accès le 24 05 2018].

[20] «VoIP Security with Asterisk,» . [En ligne]. Disponible sur le lien: https://ritcsec.wordpress.com/2017/05/19/voip-security-with-asterisk/ . [Accès le 28 05 2018].