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

Mémoire de fin d'études

pour l'obtention du diplôme de Master (2) en Informatique

Option: *Réseaux et Systèmes Distribués (R.S.D)*

Thème

Etude de la sécurité dans la VOIP

Réalisé par :

- **BENCHIKH Aissa**
- **MECHERNENE Karima**

Présenté devant le jury composé de :

- *Mme. Labraoui* *(Président)*
- *Mr. Benaissa* *(Encadreur)*
- *Mr. Benmouna* *(Examinateur)*
- *Mr. Belhoucine* *(Examinateur)*

Année universitaire : 2014-2015

Remerciements

Tout d'abord, nous remercions le Dieu, notre créateur de nous avoir donné la forces, la volonté et le courage afin d'accomplir ce travail.

Nous adressons le grand remerciement à notre encadreur **M^{r}. Benaissa** qui a proposé le thème de ce mémoire, pour ses conseils et ses dirigés du début à la fin de ce travail.

Nous tenons également à remercier messieurs les membres de jury pour l'honneur qu'ils nous ont fait en acceptant de siéger à notre soutenance.

Finalement, nous tenons à exprimer notre profonde gratitude à nos familles qui nous ont toujours soutenues et à tout ce qui participe de réaliser ce mémoire. Ainsi que l'ensemble des enseignants qui ont contribué à notre formation.

Dédicace

A mes très chers parents qui ont toujours été la pour moi et j'espère qu'ils trouveront dans ce travail toute ma connaissance et tout mon amour.

A mes chers sœurs et frères :Hakima , Esma , Redouane & Yassine.

A mon binôme Aissa BENCHIKH.

A tout mes amis.

MECHERNENE Karima

TABLE DE MATIERE

❖ Introduction

A partir de 1995, Internet a été utilisé afin de diminuer les coûts des communications téléphoniques longues distances nationales et internationales. Les communications ainsi établies coûtent seulement le prix de deux communications locales (une communication locale à chaque extrémité).

Toutefois, les enjeux de la voix sur IP sont aussi techniques et dépassent la simple idée de la communication téléphonique à moindre coût. Du fait de la convergence voix, données et images, il devient plus facile de gérer un support de transmission unique pour l'ensemble des services (tout sur IP).

En effet les entreprises dépensent énormément en communication téléphonique, or le prix des communications de la **VoIP** est dérisoire en comparaison. Il suffit simplement de louer des hébergeurs pour transmettre les communications sans avoir à payer tous les services. Cela offre donc une grande indépendance.

Le transport se faisant aussi par le biais du réseau informatique, il n'est donc pas nécessaire de devoir mettre en place un réseau téléphonique à part. Le tout est alors centralisé sur une même entité.

De plus, en positionnant la voix comme une application supplémentaire sur les réseaux **IP**, l'entreprise ne va pas uniquement substituer un transport opérateur **RTC** à un transport **IP**, mais simplifié la gestion de la voix, des données et vidéo par ce seul transport.

Il est important de noter que certaines fonctions de liaison entre postes, nécessaires dans certains cas au fonctionnement opérationnel des services utilisateurs, peuvent présenter un risque potentiel de malveillance et de vulnérabilité.

Si la sécurité des systèmes d'information est souvent vue comme une contrainte, c'est probablement parce que trop souvent, ses spécialistes oublient qu'après l'exposition des vulnérabilités, leur rôle est de fournir des solutions, adaptées aux besoins des utilisateurs, efficaces et économiques, ces solutions permettent d'encadrer de manière sécurisée le déploiement et l'utilisation de nouvelles technologies. Cette sécurité-là est force d'avancées, d'accompagnement, et non de blocage.

L’objectif principal de notre projet est basé sur le test et la simulation des différentes attaques qui nous permet de découvrir les différentes failles de sécurité au niveau du service de **VoIP** sous la plate-forme **Astérisk**.

Notre mémoire est structuré comme suit :

Chapitre 1 : C’est une introduction générale sur la technologie de la **VOIP**.

Chapitre 2 : Concerne la thématique de sécurité au niveau de la **VOIP.**

Chapitre 3 : Installation et configuration de la plate-forme **Astérisk.**

Chapitre 4 : Test et simulation des différentes attaques qui peuvent arriver sur le système de communication par **VOIP.**

Chapitre I

Généralités sur la VoIP

1. Introduction

La voix sur IP (Internet Protocol) est un sujet vaste et très riche. Pour commencer il convient de définir les concepts cachés derrière le terme générique « VoIP » (Voice Over IP).

La " **VoIP** " ou littéralement " **voix sur IP** " en Français désigne l'ensemble des technologies permettant de communiquer oralement via un réseau utilisant le protocole IP. Le terme "**VoIP**" est en général utilisé pour décrire des communications "Point à Point". Pour la diffusion de son sur IP en multipoints, on parlera plutôt de streaming (comme les radios sur Internet, par exemple).

2. Définitions

- **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 du modèle TCP/IP **(Figure I.1)** permettant un service d'adressage unique pour l'ensemble des terminaux connectés.

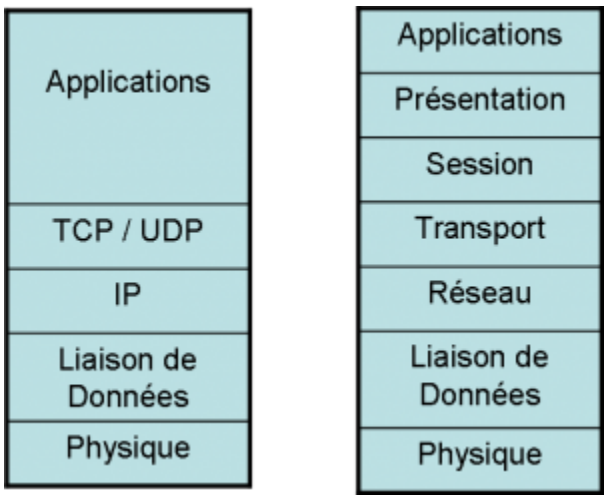


Figure I.1: Modèle OSI (Open Systems Interconnection) & du modèle TCP/IP [1]

- **VoIP (Voice Over IP)**

La voix sur IP (VoIP) regroupe l'ensemble des techniques permettant de faire transiter de la voix sur un réseau informatique. La voix sur IP comprend ainsi les communications de PC à PC. Pour ce type de communication, chaque utilisateur doit disposer d'un logiciel approprié. Si la connexion passe par le réseau Internet, on parle alors de **VoIP**, la téléphonie par Internet. Deuxième catégorie de voix sur

IP, les communications de PC à téléphone (PC to Phone). Dans les deux cas, le PC communicant est appelé Softphone, terme qui insiste sur l'émulation du PC en téléphone grâce à un logiciel.

La **ToIP** s'inscrit dans la troisième catégorie de communications en voix sur IP, les échanges de téléphone à téléphone. Les postes sont alors baptisés IP-Phone pour les distinguer de leurs homologues standards. Un téléphone IP doit en effet être alimenté par courant au contraire des téléphones classiques. Il est capable de numériser la voix pour la transmettre sur des réseaux IP et peut, à l'inverse, rassembler les paquets entrants pour interpréter la voix reçue. La téléphonie sur IP circule sur des réseaux privés **LAN** (Local Area Network), **VPN** (Virtual Private Network) ou publics **(Figure I.2)**.

Dans la littérature il arrive souvent de tomber sur le terme « **ToIP** » qui désigne les mêmes technologies que la **VoIP**, **ToIP** signifie « Telephony Over IP » ou, plus prosaïquement les techniques employées pour faire circuler la voix sur un réseau informatique dont la couche transport est prise en charge par le protocole IP.

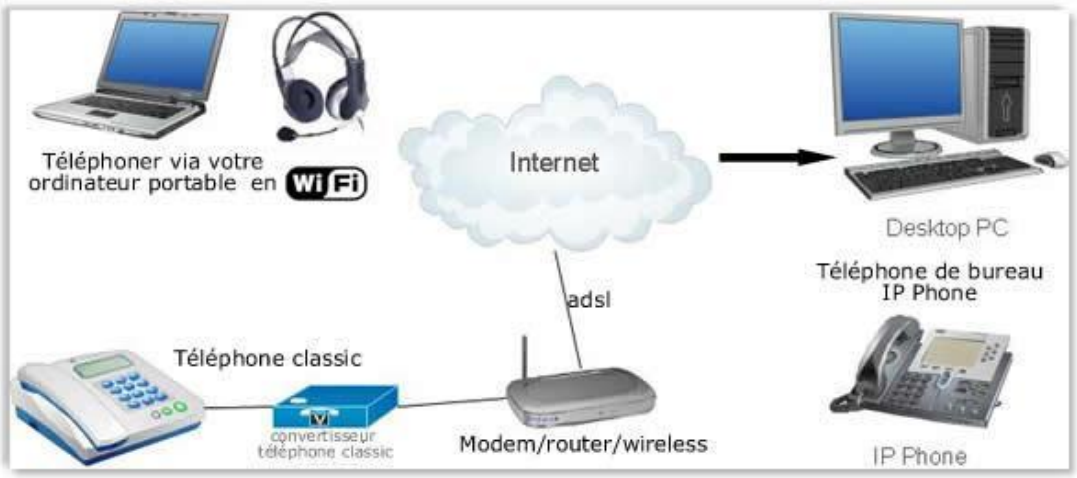


Figure I.2 : Equipements utilisés pour VoIP [2]

- **PABX**

PABX (signifie : **Private Automatic Branch eXchange**) sert principalement à relier les postes téléphoniques d'un établissement (lignes internes) avec le réseau téléphonique public (lignes externes). Il permet en plus la mise en œuvre d'un certain nombre de fonctions, notamment :

1. Relier plus de lignes internes qu'il n'y a de lignes externes.
2. Permettre des appels entre postes internes sans passer par le réseau public.
3. Programmer des droits d'accès au réseau public pour chaque poste interne.

4. Proposer un ensemble de services téléphoniques (conférences, transferts d'appel, renvois, messagerie, appel par nom…).
5. Gérer la ventilation par service de la facture téléphonique globale (taxation).
6. Apporter des services de couplage téléphonie-informatique (CTI).
7. Gérer les appels d'urgence dans les structures d'accueil hospitalières, maisons de retraite, etc.
8. Gérer un portier interphone d'immeuble et commander une gâche électrique

Il s'agit en quelque sorte d'un Switch doté de fonctionnalités particulières et peut être considéré comme étant le cœur d'un réseau privé de téléphonie (**Figure I.3**).

Le terme "IP-PABX" (ou "**IPBX**") désigne un PABX (ou PBX) utilisant la technologie IP pour accéder à son réseau.

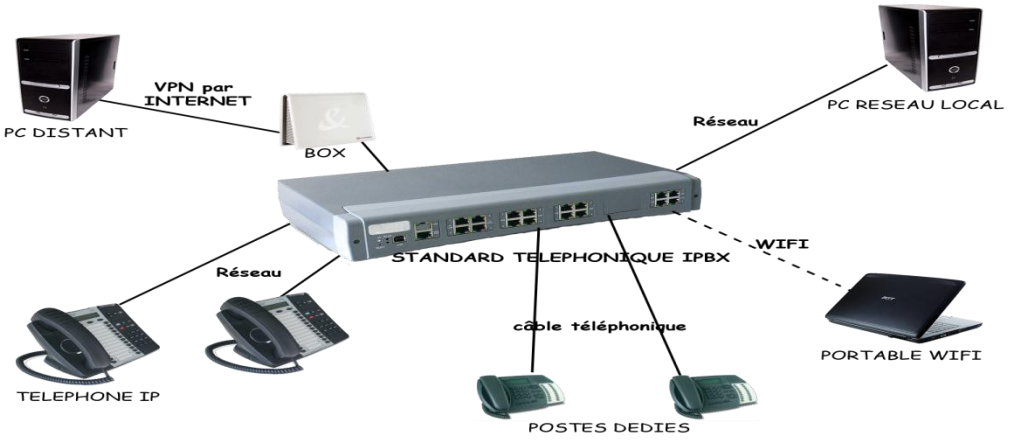


Figure I.3 : Standard Téléphonique IPBX [3]

3. Principe de transformation de la voix en IP

La bande voix (qui est un signal électrique analogique) est d'abord échantillonnée numériquement par un convertisseur puis compressée selon une certaine norme de compression variable selon les codecs utilisés, puis ensuite on peut éventuellement supprimer les pauses de silences observées lors d'une conversation, pour être ensuite habillée **RTP**, **UDP** et enfin en **IP**. 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.,) (**Figure I.4**)

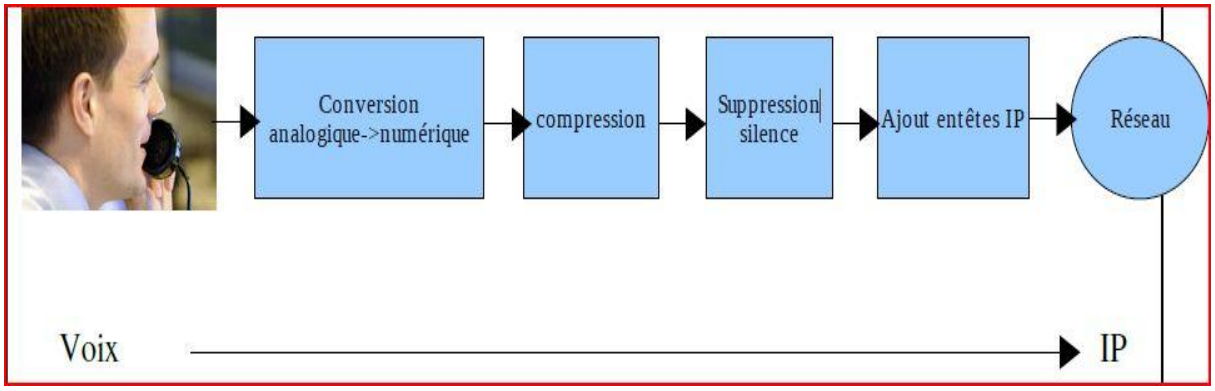


Figure I.4 : Schéma de principe pour transformer la voix en IP [4]

4. Protocoles

1. Introduction

A chaque besoin son protocole, c'est en général la règle qui s'applique à tous les protocoles. Les fournisseurs **ADSL** (Asymetric Digital Subscriber Line) utilisent en règle générale le protocole **MGCP** (Media Gateway Control Protocol) qui est un protocole asymétrique (le serveur contrôle le téléphone de l'abonné), alors que les logiciels de Voix sur IP utilisent en général SIP qui est un protocole symétrique. Historiquement le protocole H.323 ayant été développé et adopté massivement en premier lieu, il reste très présent mais sur le déclin. Ainsi on peut grossièrement identifier certains acteurs historiques se lançant dans l'aventure VoIP au moyen de protocoles plus anciens (H323) et les nouveaux arrivés utilisant les derniers protocoles (SIP par exemple).

2. RTP (Real-time Transfert Protocol)

Le but du protocole RTP et de fournir un moyen uniforme de transmettre sur IP des données soumises à des contraintes de temps réel (audio, vidéo, ...) (**Figure I.5**). Le rôle principal de RTP consiste à mettre en œuvre des numéros de séquence de paquets IP pour reconstituer les informations de voix ou vidéo même si le réseau sous-jacent change l'ordre des paquets.

Plus généralement, RTP permet :

- D'identifier le type de l'information transportée,
- D'ajouter des marqueurs temporels et des numéros de séquence à l'information transportée,
- De contrôler l'arrivée à destination des paquets.

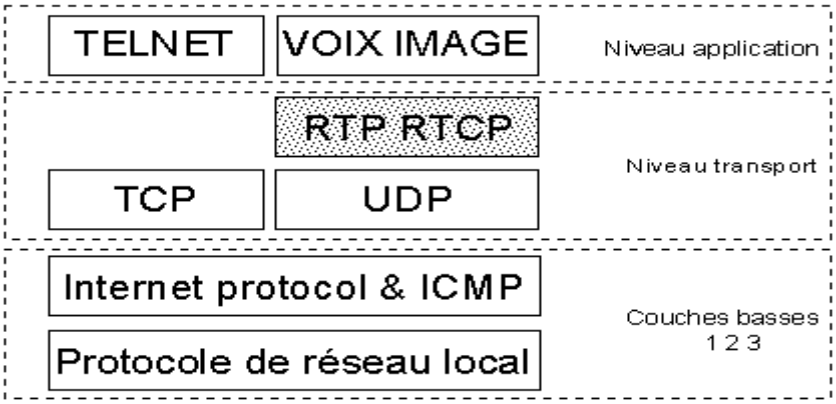


Figure I.5 : Protocole RTP dans le modèle OSI [5]

3. RTCP (Real-time Transfert Control Protocol)

Le protocole RTCP est basé sur des transmissions périodiques de paquets de contrôle par tous les participants dans la session.

C'est un protocole de contrôle des flux RTP, permettant de véhiculer des informations basiques sur les participants d'une session, et sur la qualité de service.

4. ICMP (Internet Control Message Protocol)

Le protocole ICMP (Internet Control Message Protocol) est un protocole qui permet de gérer les informations relatives aux erreurs aux machines connectées. Etant donné le peu de contrôles que le protocole IP réalise, il permet non pas de corriger ces erreurs mais de faire part de ces erreurs aux protocoles des couches voisines. Ainsi, le protocole ICMP est utilisé par tous les routeurs, qui l'utilisent pour signaler une erreur (appelé Delivery Problem).

5. UDP (User Datagram Protocol)

Le rôle de ce protocole est de permettre la transmission de données de manière très simple entre deux entités, chacune étant définie par une adresse IP et un numéro de port. Contrairement au protocole TCP, il fonctionne sans négociation : il n'existe pas de procédure de connexion préalable à l'envoi des données. Donc UDP ne garantit pas la bonne livraison des datagrammes à destination, ni leur ordre d'arrivée.

6. Protocole de signalisation

Le concept de protocole de signalisation va être utilisé dans les chapitres suivants, voici donc une petite explication préliminaire : 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 intermédiaire de contrôle et de gestion de communications. Leurs échanges permettent l'initiation, la négociation, l'établissement, le maintien et la fermeture de la connexion.

Il convient de distinguer deux types de transferts pour comprendre à quoi correspond la signalisation:

- Le transfert de données brutes.
- Le transfert d'informations de contrôle.

Le transfert de données brutes concerne les échanges de données binaires d'un poste vers un autre. L'objectif de ce transfert est de reproduire à l'identique des données en les faisant transiter par un réseau. Par exemple, deux correspondants peuvent s'échanger un fichier audio **MP3** ou des images **JPEG**. De la même façon, si on considère une conversation téléphonique en cours, les intervenants produisent des sons qui doivent être recomposés et diffusés chez leurs correspondants.

Dans tous les cas, seul l'envoi des données à de l'importance, ce qui relève d'un transport d'informations.

Le transfert d'informations de contrôle concerne les échanges de type protocolaires exécutant une action prédéfinie, et donc nécessairement limitée en possibilités. L'objectif de ce transfert est d'assurer la maitrise et la gestion du flux.

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.

De la même manière, lorsque la sonnerie d'appel retentit dans le terminal appelé, l'appelant en est immédiatement informé par une tonalité particulière sur son terminal téléphonique. Il s'agit là aussi d'une information de signalisation.

Si un correspondant ne répond pas à un appel, il est probable que sa messagerie téléphonique va s'enclencher. Cette redirection d'appel du poste appelé vers sa messagerie est également une information de signalisation. Elle ne transporte aucune information de données brute, mais vise à signaliser à l'appelant que l'appelé n'est pas disponible et que sa messagerie est opérationnelle.

Ces deux catégories de transfert sont liées : ce n'est que lorsque l'appelé a répondu à l'appel que commence le transfert d'information brutes, c'est-à-dire le transport de la voix, qui doit être

fidèlement retransmise d'un correspondant à l'autre. La signalisation n'est que l'étape préalable qui a mis en place la connexion entre les différents utilisateurs pour permettre la communication.

Dans le modèle **OSI**, la signalisation téléphonique correspond à une fonctionnalité de niveau 7 (applicatif). Elle n'est donc jamais assurée par des entités réseau de routage pur, comme les routeurs ou switchs, qui fonctionnent à des couches inférieures. Des entités dédiées sont exploitées à ces fins : il s'agit de serveurs au niveau du cœur du réseau et des terminaux (téléphone, ordinateur ou PDA par exemple) en bordure de réseau, au niveau de l'utilisateur.

Pour être comprise et correctement interprétée de l'ensemble des entités participant aux mécanismes de signalisation, celle-ci doit respecter une syntaxe particulière. C'est tout l'objet de la spécification d'un protocole de signalisation.

- **H.323**

Le protocole **H.323** figure parmi les plus réputés des protocoles de signalisation pour la téléphonie sur IP. H.323 n'est en réalité que la référence du protocole. Son nom complet est *Packet-based Multimedia Communications Systems*. Comme son nom l'indique, il peut être utilisé pour tous les réseaux à commutation de paquets, en particulier IP.

Ce protocole est spécifié pour le traitement de la signalisation des données multimédia avec de fortes contraintes temporelles, comme la voix ou la vidéo.

- **Famille de protocole H.323**

H.323 se dessine en 3 grandes parties **(Figure I.6)**. En effet, pour établir une communication audio ou vidéo sur IP, le signal doit être encodé en utilisant des codecs normalisés définis dans la norme H.323, qui normalise aussi la signalisation à utiliser pour l'établissement d'une communication. La voix ou la vidéo est transmise en utilisant le protocole UDP, associé aux protocoles RTP et RTCP pour le transfert des données en temps réel.

- Parmi les codecs possibles figurent G.711, G.723 et G.729 pour les signaux audio, H.261 et H.263 pour les signaux vidéo.
- La signalisation pour l'établissement des appels est mise en œuvre à l'aide de trois protocoles:
    - **H.225 RAS** (**Registration, Admission and Status**) : La signalisation **RAS** est utilisée entre les endpoints (Terminal) et le Gatekeeper qui les contrôle. RAS permet donc au **Gatekeeper** de contrôler les **endpoints** présents dans sa zone. Autrement dis la gestion du trafic entre le client et le serveur de communication.

- **H.225 Call signaling** : Cette signalisation permet d'établir et de libérer des connexions entre **endpoints** H.323.
- **H.245** : Lorsque l'appelé décroche, le protocole H.245 permet l'établissement de canaux RTP/RTCP permettant le transfert de données multimédia et le contrôle de ce transfert.

- Les protocoles temps réel sur IP utilisés sont RTP et RTCP. RTP fournit un transport de bout en bout sur un réseau pour les applications transmettant des données en temps réel, telles que la voix ou la vidéo, en unicast et en multicast. RTP ne se préoccupe pas de la réservation de ressources et ne garantit pas la qualité de service des transferts de données en temps réel. Le transport des données bénéficie aussi du protocole de contrôle RTCP qui fournit un contrôle minimal et des fonctions d'identification particulièrement utiles dans le cas de réseaux multicast. RTP et RTCP sont conçus pour être indépendants des réseaux sous-jacents.

Pour le contrôle et la signalisation : H.225, H.245, RTCP.

Pour la voix : G.711, G.722, G.723, G.726, G.728, G.729.

Pour la vidéo : H.261, H.263, H.263+, H.264.

Pour les données : T.123, T.124, T.125.

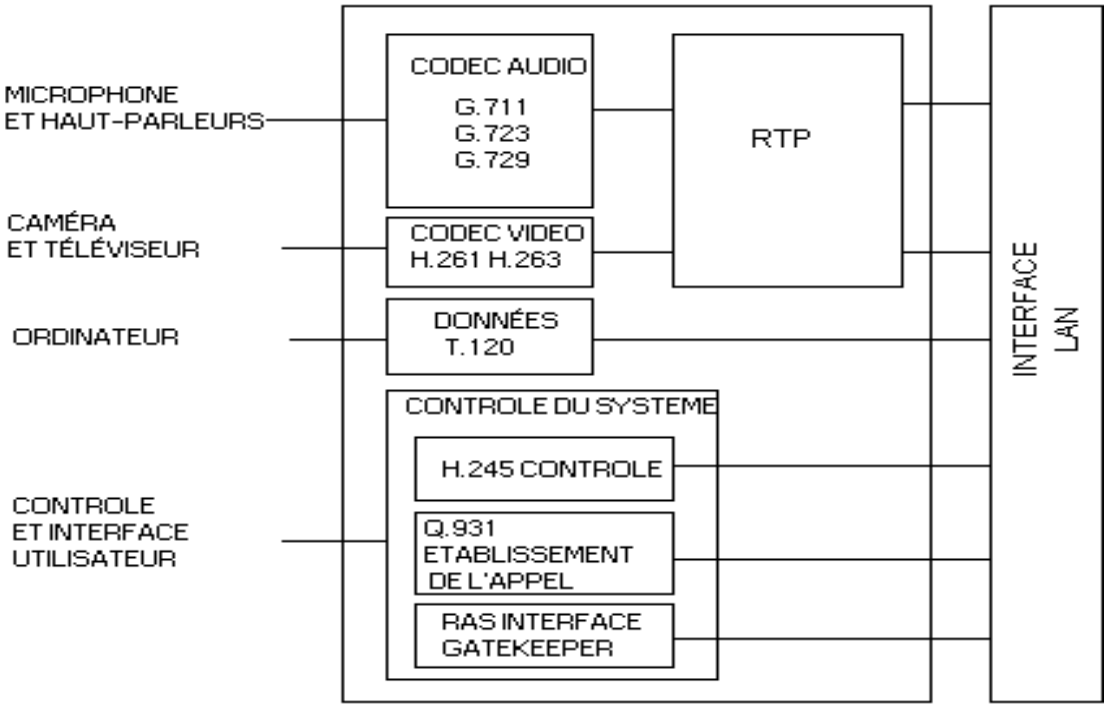


Figure I.6 : Architecture client H.323 [6]

- **SIP (Session Initiation Protocol)**

Ce que nous intéresse beaucoup, c'est ce protocole de signalisation de vidéo et voix sur IP, SIP (Session Initiation Protocol) est un protocole développé par l'Internet Engineering Task Force (IETF) permettant la négociation et l'établissement de sessions VoIP. SIP est un protocole de couche 5 du modèle OSI, dite de session. Il s'appuie généralement sur une couche de transport UDP, bien qu'il soit possible d'augmenter sa fiabilité en l'appliquant sur du TCP. Le port par défaut de SIP est le 5060. SIP ne traite que l'établissement de session. Il ne transporte pas les données échangées pendant la communication, ce rôle étant joué par RTP (Real-time Transport Protocol). On distingue différents acteurs dans le protocole SIP. (**Figure I.7**)

- **SDP**, abréviation pour (**Session Description Protocol**), est un format pour décrire les paramètres d'initialisation des flux média. **SDP** ne fournit pas le média en soi, mais il négocie plutôt les points de terminaisons du format du média en multidiffusion. Le Streaming media est le contenu qui peut être visualisé ou entendu lors de la transmission. SDP est conçu pour être extensible afin de supporter de nouveaux types de média et formats. SDP a trouvé des usages avec le RTP et le SIP, malgré son origine en tant que protocole d'annonce de session **SAP** (**Session Announcement Protocol**). **[7]**

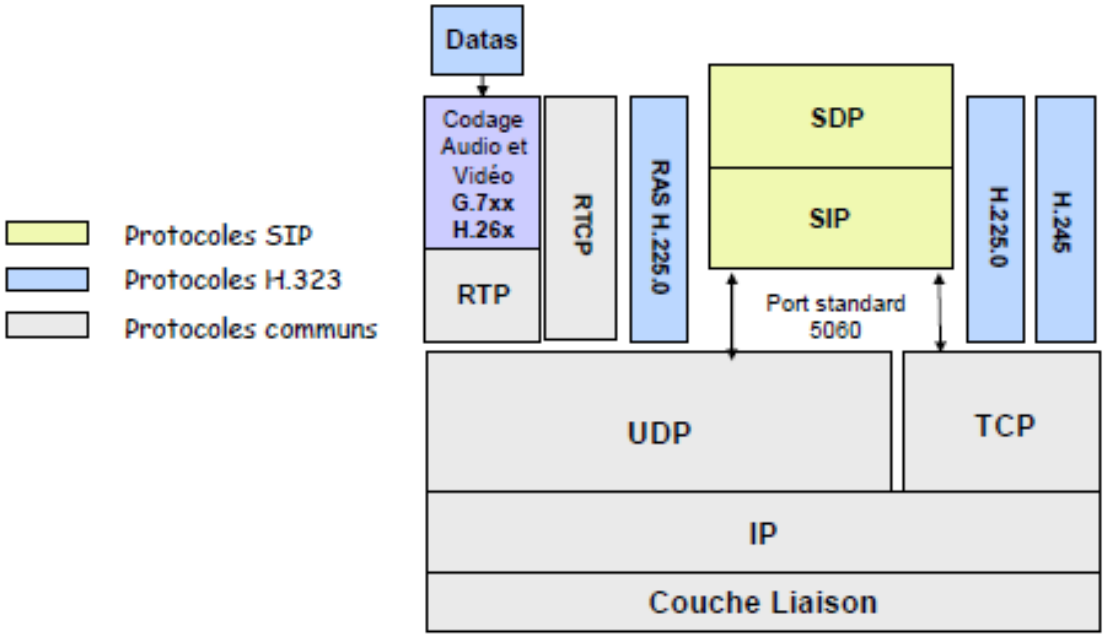


Figure 1.7 : L'encapsulation du protocole SIP en comparaison avec H.323. [6]

Dans la négociation et l'établissement de session on trouve l'User Agent, il peut s'agir d'un téléphone IP, d'un téléphone analogique (RTC) relié à un boîtier ATA (Analog Telephony Adapter) ou encore d'un softphone. C'est l'équipement manipulé par l'usager. Un élément fondamental est le Registrar Server. Il établit la correspondance entre une adresse à long terme (une URI ou un numéro de téléphone) et une adresse à court terme, typiquement une adresse IP. **[12]**

Le dernier élément très important d'une architecture SIP est le serveur proxy. Il relaye les messages des User Agent à leur destination. Sa raison d'être est que les User Agent ne peuvent pas toujours joindre directement les autres périphériques, notamment les User Agent hors de leur réseau.

Les messages utilisés par SIP sont volontairement similaires à ceux utilisés par le HTTP. Ils sont codés en ASCII et utilisent des codes proches de ceux du http (**Figure I.8)**. Différents messages sont utilisés par SIP, les plus importants étant les suivants :

· **REGISTER** : Le client envoie ce message à son registrar pour s'enregistrer, c'est-à-dire pour donner son URI et son adresse IP au registrar.

· **INVITE** : Ce message permet à un client de demander l'établissement d'une nouvelle session. Il peut être utilisé également en cours de communication pour modifier la session.

· **ACK** : Le message ACK confirme l'établissement d'une session **SIP**, suite à un message INVITE.

· **CANCEL** : Ce message annule une demande de session précédemment effectuée avec un INVITE.

· **BYE** : termine une session en cours. Contrairement au message **CANCEL**, la session **SIP** doit être active pour pouvoir envoyer un message **BYE**. Même si son comportement peut sembler similaire à **CANCEL**, une différence fondamentale existe : le message **BYE** représente un succès (la communication a eu lieu et est désormais terminée) alors que **CANCEL** est, du point de vue de l'usager, un échec : l'appelé n'a pas répondu à temps, l'appelant a donc raccroché, la session n'a donc pas abouti.

· **Des codes de réponse** : Des codes à trois chiffres, similaires à ceux du HTTP, sont envoyés en réponse à un précédent message. Le premier chiffre détermine le type de réponse, les deux suivants donnent une indication plus précise.

- **Classe 1xx -** Réponse temporaire : la requête est en cours de traitement.
- **Classe 2xx -** Succès : l'action demandée a été reçue, comprise et acceptée.
- **Classe 3xx -** Redirection : une autre action auprès d'un autre équipement est nécessaire.
- **Classe 4xx -** Erreur du client : la requête est mal formée
- **Classe 5xx -** Erreur du serveur : le serveur n'a pas réussi à répondre correctement à la requête.

- **Classe 6xx** - Autre erreur, problème global.

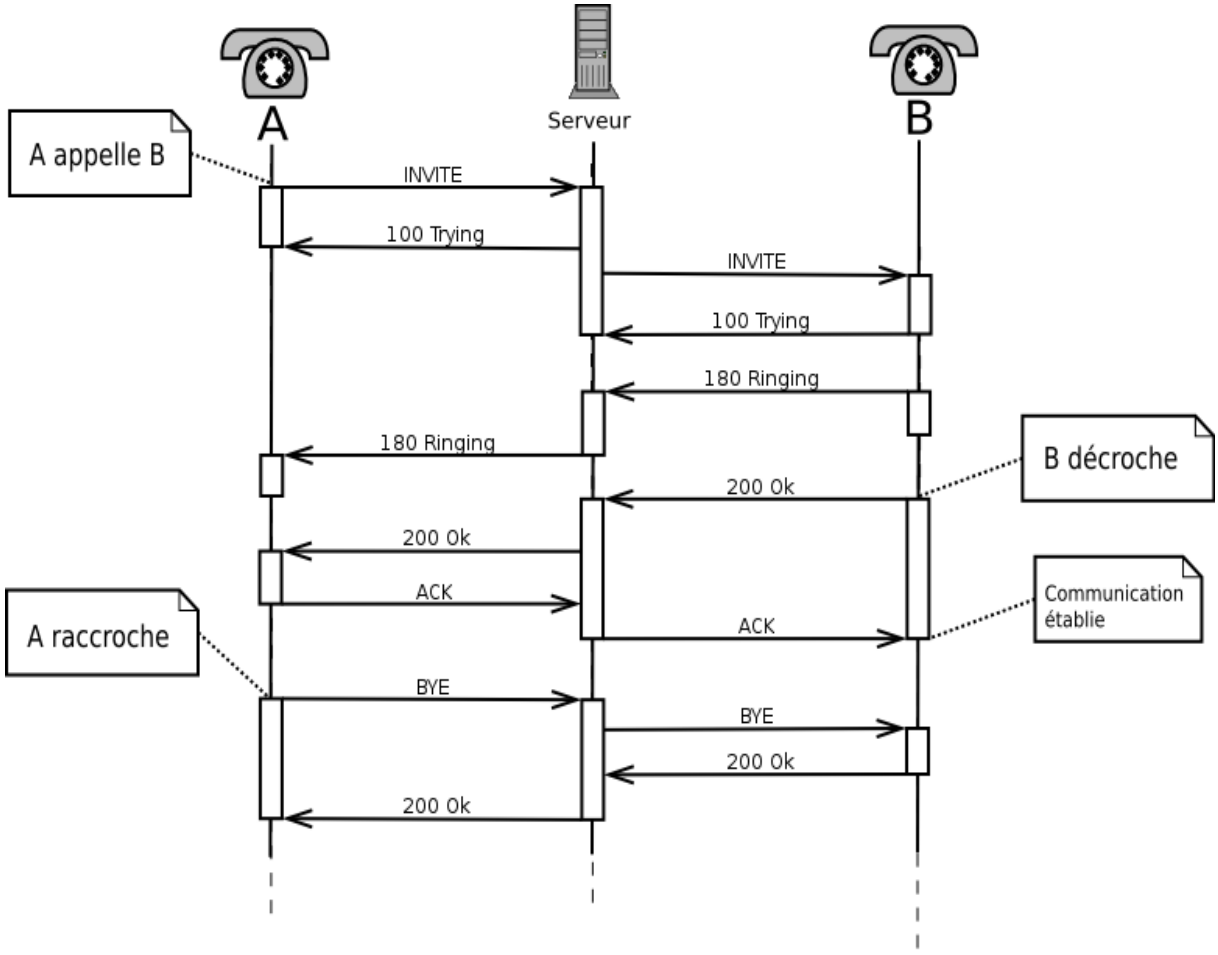


Figure I.8 : Procédure d'établissement de session SIP [9]

- **Les avantages SIP**

L'implémentation de la **VoIP** avec le protocole de signalisation SIP (Session Initiation Protocol) fournit un service efficace, rapide et simple d'utilisation. SIP est un protocole rapide et léger.

Les utilisateurs s'adressent à ces serveurs Proxy pour s'enregistrer ou demander l'établissement de communications. On peut s'enregistrer sur le Proxy de son choix indépendamment de sa situation géographique. L'utilisateur n'est plus "attaché" à son autocommutateur. Une entreprise avec plusieurs centaines d'implantations physiques différentes n'a besoin que d'un serveur Proxy quelque part sur l'Internet pour établir "son" réseau de téléphonique "gratuit" sur l'Internet un peu à la manière de l'émail.

- **Les inconvénients SIP**

L'une des conséquences de cette convergence est que le trafic de voix et ses systèmes associés sont devenus aussi vulnérables aux menaces de sécurité que n'importe quelle autre donnée véhiculée par le réseau.

En effet **SIP** est un protocole d'échange de messages basé sur HTTP. C'est pourquoi SIP est très vulnérable face à des attaques de types **DoS** (dénis de service), détournement d'appel, trafic de taxation,… etc.

- **IAX (Inter Asterisk eXchange)**

Protocole de signalisation de voix/ vidéo sur **IP. IAX** (**Inter Asterisk eXchange**), utilisé par Asterisk (Asterisk est un PABX (Private Automatic Branch eXchange) et surtout un IPBX (Internet Protocole Branche eXchange)). Ce protocole fonctionne sur le port 4569 en UDP et transporte à la fois les données (voix) et la signalisation. L'intérêt principal de ce protocole est d'être fait pour traverser le NAT (Network Address Translation un routeur fait la traduction d'adresse réseau) et qu'il est possible de créer des trunks IAX (appelés également Canaux (définissant le nombre d'appels simultanés entre un opérateur IP et un IPBX)), entre les serveurs dans lesquels les communications RTP sont multiplexées ainsi on économise les surcharges d'entêtes IP (**Figure I.9**).

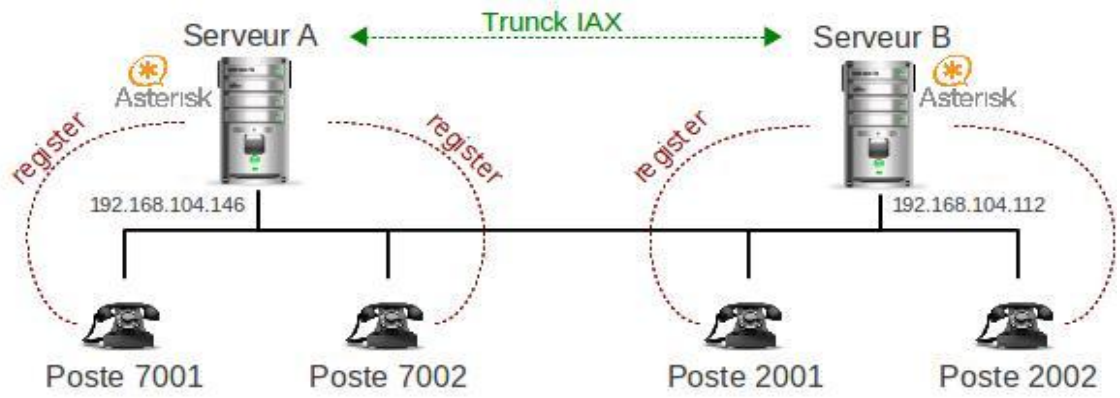


Figure I.9 : Rôle du protocol IAX [10]

L'avantage principal d'IAX est qu'il permet à plusieurs appels d'être rassemblés dans un seul ensemble de paquets IP, transportant des informations concernant plusieurs appels en cours. Et ça réduit la consommation de bande passante pour un ensemble d'appels (par l'utilisation du trunking).**[8]**

En bref, la simplicité, la rapidité et la légèreté d'utilisation, tout en étant très complet, du protocole **SIP** sont autant d'arguments qui pourraient nous permettre d'opter pour son choix. De plus, ses avancées en matière de sécurité des messages sont un atout important par rapport à ses concurrents.

7. Les Codecs

Codec est une abréviation pour Codeur/Décodeur. Un codec est basé sur un algorithme qui permet la compression des données qu'on lui donne. Il s'agit d'un procédé permettant de compresser et de décompresser un signal, de l'audio ou de la vidéo, le plus souvent en temps réel, permet une réduction de la taille du fichier original. Le codec numérise et compresse la voix de l'émetteur, ainsi les données numériques sont encapsulées dans des paquets IP et acheminées vers le destinataire. A l'arrivés au destinataire, ce dernier grâce au même codec décompresse et restitue le son. Il On distingue des codecs à pertes et codecs sans pertes. Un codec à pertes distingue les parties moins importantes des informations et les supprime pour gagner en taille.

8. Qualité de la voix

Dans la téléphonie sur IP, les différents codecs retransmettent plus ou moins bien le signal original (**Figure I.10**). Pour mesurer la qualité de la voix restituée, on parle de score MOS (Mean Opinion Score). C'est une note comprise entre 1 et 5 et attribuée par des auditeurs jugeant de la qualité de ce qu'ils entendent. Pour la VoIP, plusieurs codecs peuvent servir. Voici leurs détails :

G.711 : Ce codec est le premier à avoir été utilisé dans la VoIP. Même si il existe maintenant des codecs nettement plus intéressants, celui-ci continue d'être implémenté dans les équipements à des fins de compatibilité entre marques d'équipements différentes.

G.722 : A la différence du G.711, ce codec transforme le spectre jusqu'à 7kHz ce qui restitue encore mieux la voix. Les débits que ce codec fournit sont 48,56 ou 64kbit/s. Une des particularités est de pouvoir immédiatement changer de débit. Ceci est fortement appréciable lorsque la qualité du support de transmission se dégrade.

G.722.1 : Dérivé du codec précédent, celui-ci propose des débits encore plus faibles (32 ou 24kbit/s). Il existe même des versions propriétaires de ce codec fournissant un débit de 16kbit/s.

G.723.1 : C'est le codec par défaut lors des communications à faible débit. Deux modes sont disponibles. Le premier propose un débit de 6,4kbit/s et le deuxième un débit de 5,3kbit/s. **[13]**

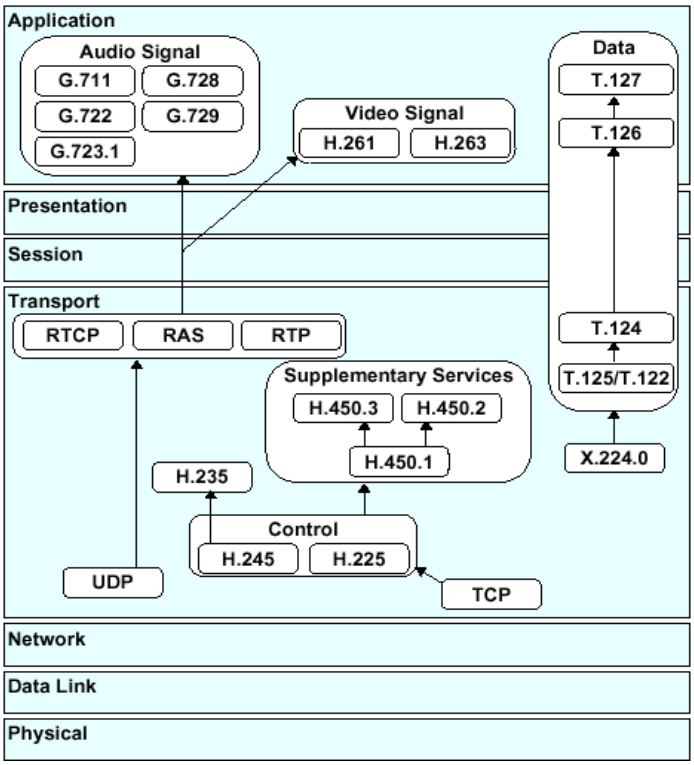


Figure I.10 : Différents codecs dans le modèle OSI [14]

- **Compression du silence**

Une des méthodes utilisées par les **codecs** pour réduire la quantité de données à transmettre et de détecter les silences. Dans une conversation téléphonique, chaque locuteur ne parle que 1/3 du temps en moyenne. Ce qui fait que 1/3 du temps d'une conversation est constitué de silence facilement reproductible et donc non codé par le codec. Ce mécanisme s'appelle VAD (Voice Activity Détection - DAV : Détection d'activité de la voix).

- **Génération de bruits de confort**

Pendant une conversation où les silences sont effacés, l'absence de bruit chez le récepteur peut vite se révéler inconfortable. Dans cette optique, les codecs disposent d'un générateur de bruits de confort visant à simuler des bruits de fond pour améliorer le confort des utilisateurs.

- **Robustesse sur la perte de paquets**

Si les conditions de circulations sur le réseau se dégradent, certains paquets contenant de l'information peuvent se perdre ou arriver trop tard. Ce problème est en partie compensé par l'utilisation des buffers, mais la gigue peut être telle que le codec soit obligé de retransmettre au récepteur un paquet, alors qu'il n'est pas arrivé. Il existe plusieurs méthodes pour palier à ce

problème: Il est possible par exemple de simplement répéter le contenu du dernier paquet pour combler le vide. On peut aussi répartir l'information sur plusieurs paquets de façon à introduire une redondance des données. En cas de pertes de paquets, le codec dispose ainsi d'une copie du paquet à retransmettre.

5. Application

Le but de ce type d'application est de fournir des services téléphoniques à des utilisateurs locaux ou distants avec le principe de la **VoIP**, qui réduit considérablement les coûts téléphonique de l'entreprise (réseau convergé, capacité liaisons **WAN** utilisées, équipe technique unique) ou de l'utilisateur (utilisation d'Internet au lieu du réseau **RTC** ce qui réduit les coûts aussi).

Notre application va permettre, à travers l'étude du logiciel **OpenSource Asterisk [15]**, de faire un tour d'horizon de cette technologie et des possibilités offertes par cet outil. Par ailleurs cela nous aura appris les notions de **VoIP** et d'utilisation d'**Asterisk** qui pourront être mise en avant et faire la sécurisation de voix aux attaques de déni de services et autres vulnérabilités causées par des intrus.

Donc les objectifs de ce travail consistent à valider et assurer que les appels émis ou reçus, par la société ou l'individu qui sont internes ou externes soient conformes et corrects aux différentes règles compris y leurs sécurités.

- **Cadre de l'application**

  - **Identification de l'application**

Implémentation du serveur de téléphonie **Asterisk** dans le cadre de projet de création d'un centre service client.

  - **Définition de l'application**

Il est question dans ce projet de mettre en place dans un environnement précis (centre service client), une architecture de communication reposant sur le F composée de plusieurs serveurs.

  - **Caractéristiques**

- Mise en place d'une solution **VoIP** basée sur un **IPBX** Open source **Asterisk**.
- Proposer une solution technique à l'administrateur système et réseau d'admettre les appels émis intra-site, inter-site et vers/de l'extérieur.

  - **Contraintes**

La solution que nous proposons doit être sécurisée et doit être correctement dimensionnée pour éviter toutes pertes d'appels ou intrusion.

6. Asterisk

Le nom **Asterisk** fait référence au symbole "*****" qui signifie "**wildcard**" en ligne de commande **Unix** et **DOS**. Ce choix a été fait car **Asterisk** est conçu dans le but d'offrir une très grande souplesse dans des réseaux de voix.

Asterisk est un autocommutateur téléphonique privée (**PABX**) open source pour les systèmes d'exploitation **UNIX**, il est publié sous licence GPL.

Le **PBX** open source **Asterisk** a vu le jour quand Mark Spencer, a voulu acquérir un PBX traditionnel pour sa société. Le créateur d'**Asterisk**, trouvant que le prix d'acquisition d'un PBX traditionnel était démesuré, initia un projet open source. Il a donc commencé à développer Asterisk.

- **Rôle**

Asterisk est un **IP-PBX** qui transforme un ordinateur en "central telephonique" ou "***PABX***"(***Private Automatic Branch eXchange***), qui est un **autocommutateur téléphonique privé » [11]**. Ce PBX est un commutateur qui relie dans une entreprise les appels d'un poste quelconque vers un autre (appels internes) ou avec un réseau téléphonique public (appels externes).

Asterisk a le rôle d'un middleware entre les technologies de téléphonie VOIP (TDM, SIP …) et les applications (conférence, messagerie vocale, …).

Ce PBX est base sur le protocole IP. Donc les communications et les paquets échanges sont transportés sous forme plusieurs protocoles de la voix qu'on veut (*SIP, IAX, H.323, ADSI, MGCP*).

- **Caractéristiques**

Asterisk offre toutes les fonctions d'un PBX et ses services associent :

- La conférence téléphonique.
- Les répondeurs interactifs.
- La mise en attente d'appels.
- La distribution des appels.
- Les mails vocaux.
- La musique d'attente.
- La génération d'enregistrement d'appels pour l'intégration avec des systèmes de facturation.

Asterisk fonctionne sur les principaux systèmes d'exploitation (Linux, BSD, Windows, Mac OS X).**[11]**

- **Architecture interne d'Asterisk**

Asterisk est composé d'un noyau central de commutation, de quatre API (Interface de programmation d'applications) de chargement modulaire des applications téléphoniques, des interfaces matérielles, de traitement des formats de fichier, et des codecs. Il assure la commutation transparente entre toutes les interfaces supportées, permettant à cette commutation de relier entre eux une diversité de systèmes téléphoniques en un unique réseau commuté.

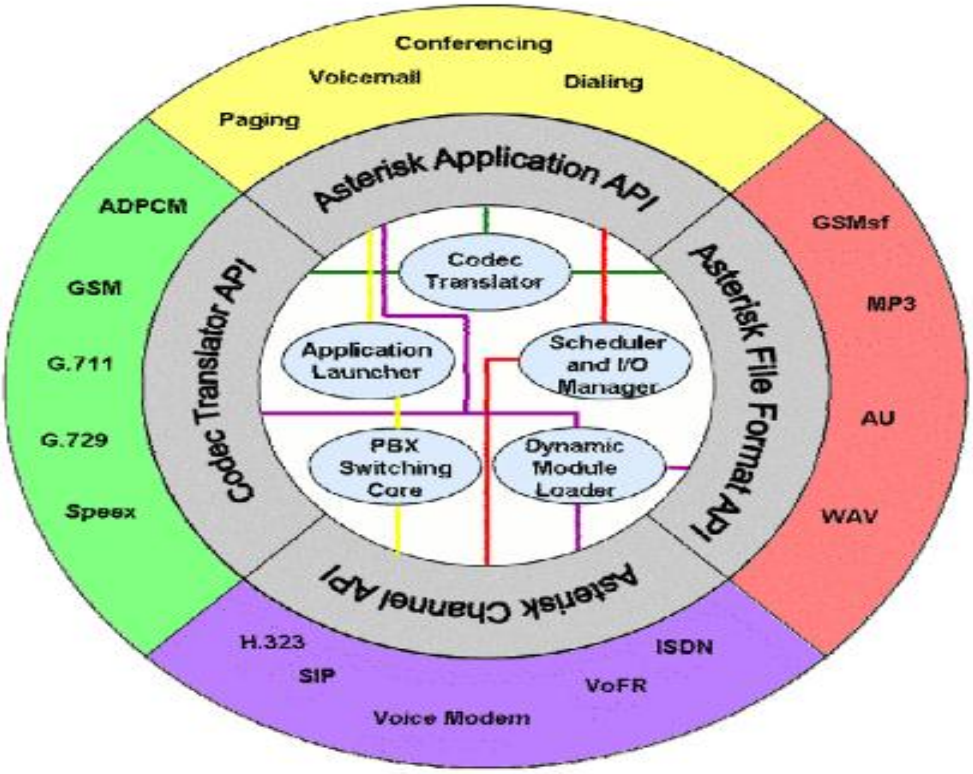


Figure I.11 : architecture interne d'Asterisk [11]

- **Principales fonctions**

Les principales fonctions d'Asterisk sont **:**

- ✓ **La commutation de PBX (**PBX Switching Core) : Système de commutation de central téléphonique privé, reliant ensemble les appels entre divers utilisateurs et des tâches automatisées. Le noyau de commutation relie d'une manière transparente des appels arrivant sur divers interfaces de matériel et de logiciel.

- ✓ **Lanceur d'applications (**Application Launcher) : Lance les applications qui assurent des services pour des usagers, tels que la messagerie vocale, la lecture de messages et le listage de répertoires (annuaires).
- ✓ **Traducteur de codec (**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 pallier aux divers besoins et pour arriver au meilleur équilibre entre la qualité audio et l'utilisation de la bande passante.
- ✓ **Planificateur Manager d'I/O (**Scheduler & I/O Manager) : Ils traitent la planification des tâches de bas niveau et la gestion du système pour une performance optimale dans toutes les conditions de charge.
- ✓ **Dynamic Module Loader :** charge les pilotes (lors de la 1ère exécution d'Asterisk, il initialise les pilotes et fait le lien avec les APIs appropriés). Après que les pilotes soient chargés (DML), les appels commencent à être acceptés (PBXSC) et redirigés en faisant sonner les téléphones (AL).

- **Les APIs (Application Programming Interface)**

L'abstraction matérielle et protocolaire passe par l'utilisation de **4 APIs [11]** :

- ✓ **Asterisk Application API :** Elle autorise différents modules de tâches à être lancé pour exécuter diverses fonctions. Communication, audioconférence, pagination, liste d'annuaire, messagerie vocale, transmission de données intégrée, et n'importe quelle autre tâche qu'un système PBX standard exécute actuellement ou exécutera dans l'avenir, sont mises en œuvre par ces modules distincts.
- ✓ **Asterisk Translator API:** Charge les modules de codec pour supporter divers formats de codage et de décodage audio tels que le GSM, la Mu-Law, l'A-Law, et même le MP3.
- ✓ **Asterisk Channel API:** Cette API gère le type de raccordement sur lequel arrive un appelant, que ce soit une connexion VoIP, un RNIS, un PRI, une signalisation de bit dérobé, ou une autre technologie. Des modules dynamiques sont chargés pour gérer les détails de la couche basse de ces connexions.
- ✓ **Asterisk File Format API:** Elle permet la lecture et l'écriture de divers formats de fichiers pour le stockage de données dans le file system. Sa particularité modulaire permet à Asterisk d'intégrer de façon continue le matériel de commutation téléphonique actuellement mise en

œuvre, et les technologies de Voix par paquet en constante augmentation, émergeant aujourd'hui.

7. Conclusion

La **VoIP** est une technologie révolutionnaire qui défie les règles édictées par la téléphonie RTC. Elle est plus souple, conviviale, ne nécessite pas un investissement lourd, coûte moins chère, propose de nouveaux services et beaucoup d'autres avantages, si bien que toute entreprise qui se veut compétitive et moderne aujourd'hui, jette son dévolu sur la téléphonie sur IP pour gérer ses communications tant internes qu'externes. Elle vise principalement à améliorer le cadre de travail des employés de l'entreprise en libérant l'utilisateur du lieu d'implantation du poste téléphonique.

Chapitre II

Installation et configuration D'Astérisk pour la VoIP.

1. Introduction

Dans ce chapitre nous proposons de réaliser une plateforme intégrant certains services de voix sur IP (**VoIP)** offerts par **Asterisk**.

Cette réalisation consiste à la mise en place et la configuration d'une machine contenant le serveur **Asterisk**, d'autre part l'installation et la configuration du client **X-Lite**, **Blink** et la configuration de certains services.

2. Architecture du réseau

La figure **II.1** montre l'architecture adoptée au cours de la configuration de la solution de VoIP à base d'Asterisk.

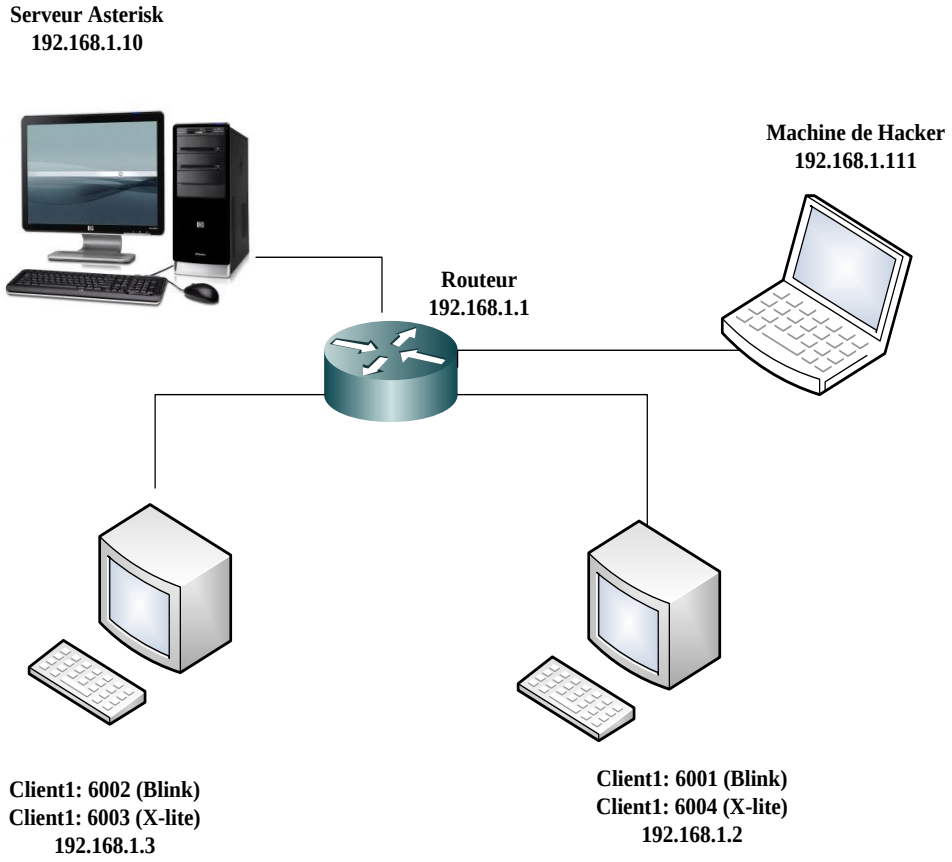


Figure II.1 : Architecture du réseau.

3. Matériel requis

- Un serveur sur lequel sont installé asterisk.
- Deux clients **SIP** : Sont des machines sur lesquelles installé le système d'exploitation **windows** et un client **X-Lite** et **Blink**.
- Un routeur.

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

1. Installation du système

Le serveur qui héberge la plate-forme Asterisk est un serveur GNU/Linux fonctionnant avec un système d'exploitation Ubuntu 14.04 LTS.

2. Installation d'Asterisk

- **Préparation à l'installation**

On commence par mettre à jour notre distribution et installer les dépendances nécessaires à la compilation d'Asterisk.

```
root@ubuntu:/# apt-get update && apt-get upgrade
root@ubuntu:/# apt-get install build-essential libxml2-dev libncurses5-dev linux-headers-
`uname -r` libsqlite3-dev libssl-dev
```

- **Téléchargement**

On crée un dossier ou nous allons placer les sources d'Asterisk dans /usr/src

```
root@ubuntu:/# mkdir /usr/src/asterisk
root@ubuntu:/usr/src/asterisk#
```

On télécharge l'Asterisk, la version installée d'Asterisk était la version 11.10.2.

```
root@ubuntu:/usr/src/asterisk# wget
http://downloads.asterisk.org/pub/telephony/asterisk/releases/asterisk-11.10.2.tar.gz
root@ubuntu:/usr/src/asterisk# tar xvzf asterisk-11.10.2.tar.gz
root@ubuntu:/usr/src/asterisk# cd asterisk-11.10.2/
```

- **Compilation et installation**

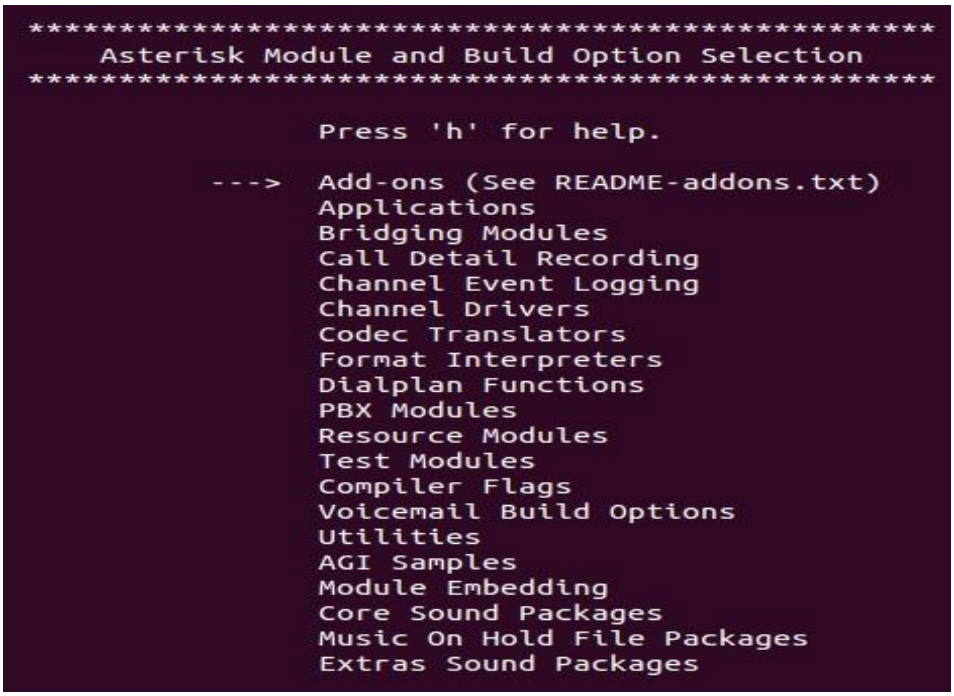
Construction d'un nouveau fichier **makefile** qui contient les instructions à exécuter à partir des commandes, **./configure, make, make install, make config**, etc.

```
root@ubuntu:/usr/src/asterisk/asterisk-11.10.2# ./configure
```

La commande **make menuselect** permet d'installer des modules supplémentaires.

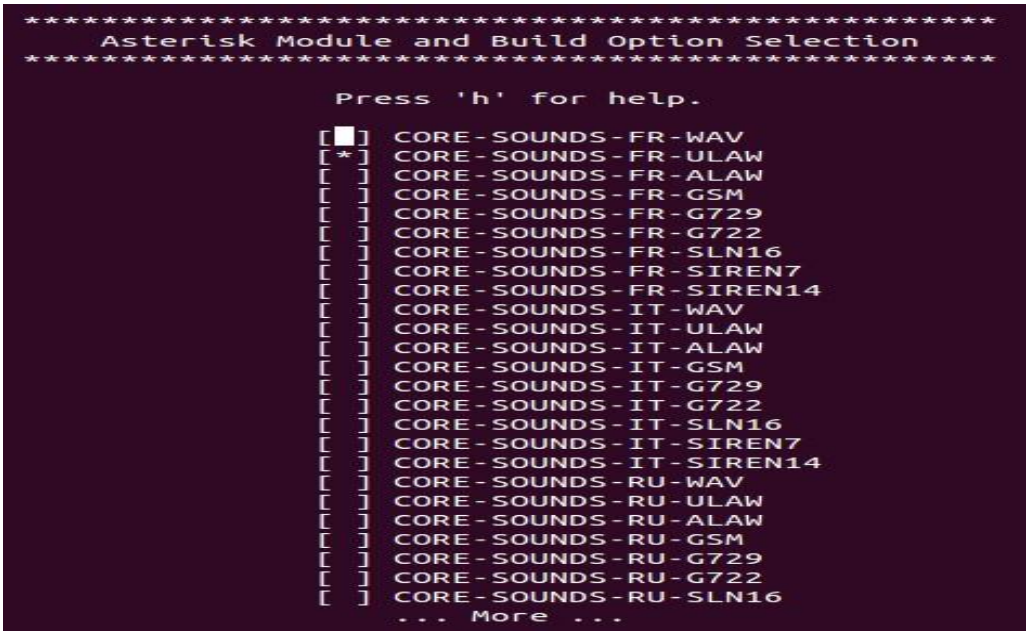
```
root@ubuntu:/usr/src/asterisk/asterisk-11.10.2# make menuselect
```

On a une fenêtre comme suit apparaitre :

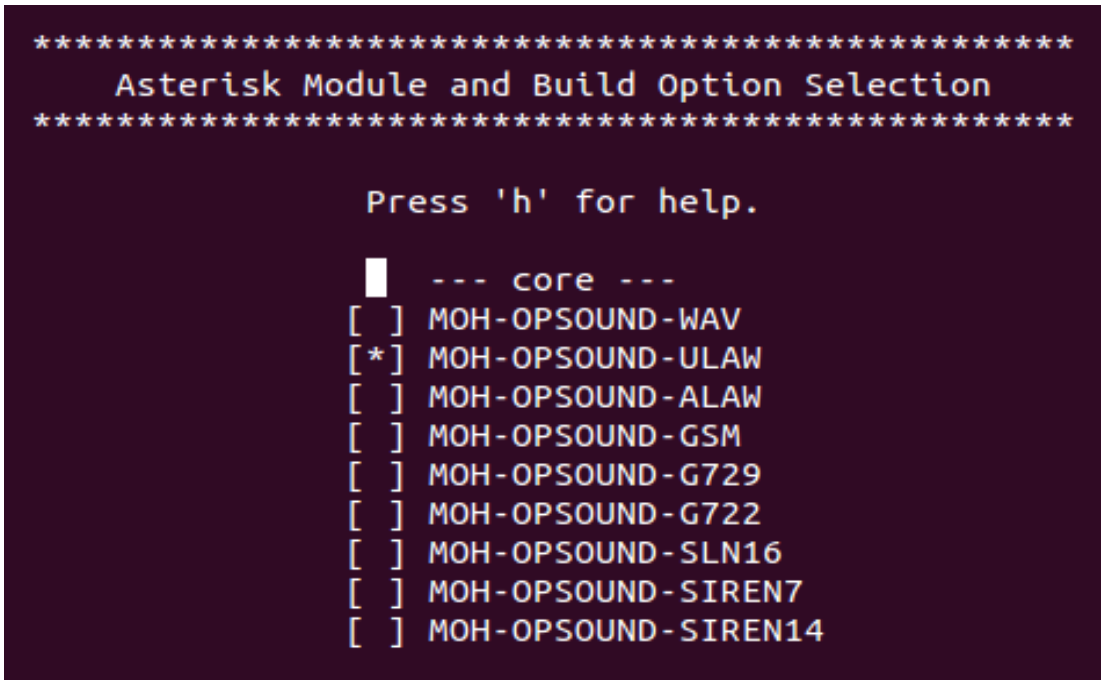


On installe les sons français pour **Asterisk** au format **µ-law**.

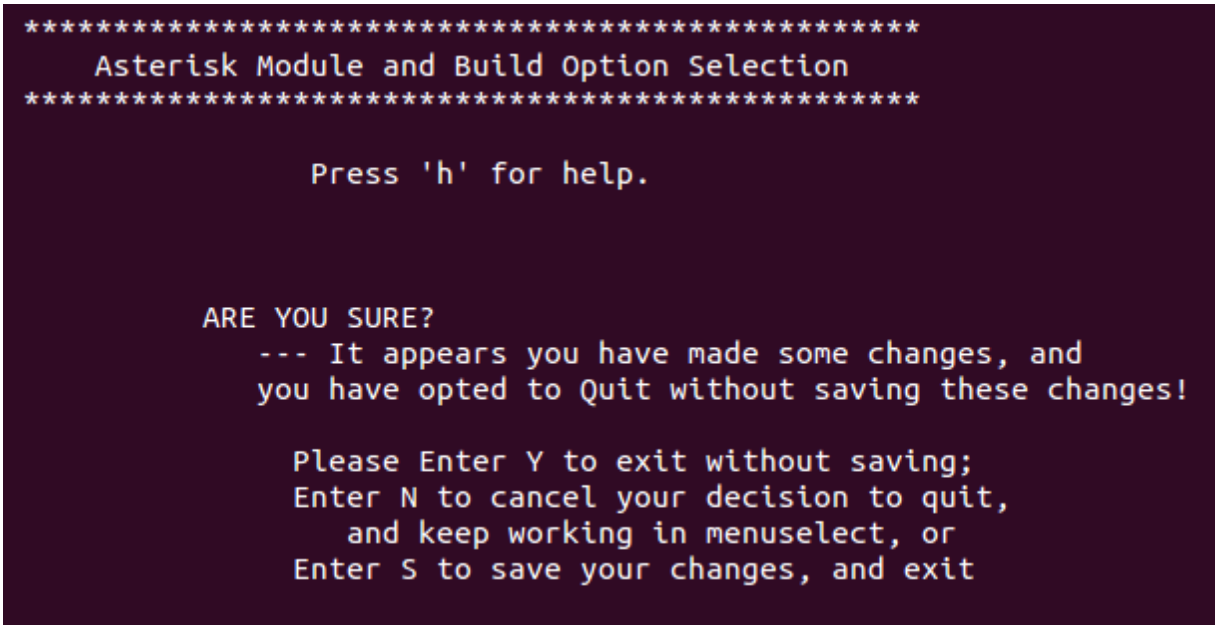
Dans **Core Sound Package** nous allons cocher la case **CORE-SOUNDS-FR-ULAW** avec la touche **Espace** puis appuyez sur **Echap** pour retourner à l'écran précédent.



Puis dans **Music On Hold File Packages** cochez **MOH-OPSOUND-ULAW** (Décochez celui en WAV), appuyez sur **Echap**



Sur l'écran principal refaites **Echap** et appuyez sur **S** pour sauvegarder les changements.



Enfin tapez les commandes suivantes pour terminer l'installation:

```
root@ubuntu:/usr/src/asterisk/asterisk-11.10.2# make  (compilation du code source)
root@ubuntu:/usr/src/asterisk/asterisk-11.10.2# make install
root@ubuntu:/usr/src/asterisk/asterisk-11.10.2# make samples
```

make samples (optionnelle) permet de créer des fichiers de configuration de base. (Elle peut être utile par la suite pour revenir à une configuration d'origine d'asterisk).

```
root@ubuntu:/usr/src/asterisk/asterisk-11.10.2# make config
```

make config charge le serveur **Asterisk** au démarrage du système.

- **Remarque *:*** Pendant le **make install** assurez-vous que votre serveur soit connecté à internet parce que celui-ci va télécharger les package de langue **FR** que nous avons sélectionné.

- **Démarrage du serveur Asterisk**

Lancement d'Asterisk à l'aide de la commande suivante:

```
root@ubuntu:/usr/src/asterisk/asterisk-11.10.2# /etc/init.d/asterisk start
```

Nous pouvons vérifier le bon fonctionnement de votre serveur Asterisk avec la commande suivant qui sert à afficher la console d'Asterisk.

```
root@ubuntu:/etc/asterisk# asterisk -cvvvvvvvvvvr
```

On a l'affichage suivant: (c.-à-d. que le serveur Asterisk est démarré et fonctionnel)

```
root@ubuntu:/etc/asterisk# asterisk -cvvvvvvvvvvr
Asterisk 11.10.2, Copyright (C) 1999 - 2013 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=========================================================================
Connected to Asterisk 11.10.2 currently running on ubuntu (pid = 1038)
ubuntu*CLI>
```

Asterisk peut être lancé de 2 manières:

- En mode serveur (usage normal, à l'écoute des requêtes des clients).
  Syntaxe: **$ asterisk -vvvc**
  - 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.
  - '**d**' mode **debug**.
- En mode client (en accédant à une console permettant de dialoguer avec le serveur).
- Syntaxe: **$ asterisk -r**

Cette console peut permettre d'obtenir des informations sur l'activité du serveur, comme par exemple la liste des clients connectés (commande '**sip show peers**' en l'occurrence).

- **Commandes du serveur**

Pour connaître l'ensemble des commandes du serveur, vous pouvez taper : **help**, une fois connecté à la console **Asterisk**.

A chaque modification du fichier **sip.conf**, vous devez exécuter la commande **"sip reload"** pour recharger le fichier.

A chaque modification du fichier **extensions.conf**, vous devez exécuter la commande **"extensions reload"** pour recharger le fichier.

Pour recharger l'ensemble du serveur, tapez la commande : **reload**

- **Identification des fichiers de configuration**

Une fois l'installation d'Asterisk est effectuée, plusieurs fichiers sont créés :

- **/usr/sbin/** : Contient le fichier binaire d'Asterisk (programme principal).
- **/usr/lib/asterisk/** : Contient les fichiers binaires qu'Asterisk utilise pour fonctionner.
- **/usr/lib/asterisk/modules/** : Contient les modules pour les applications, les codecs, et les drivers.
- **/var/lib/asterisk/sounds/** : Contient les fichiers audio utilisés par Asterisk, par exemple pour les invites de la boite vocale.
- **/var/run/asterisk.pid** : Fichier contenant le numéro du processus Asterisk en cours.
- **/var/spool/asterisk/outgoing/**: Contient les appels sortants d'Asterisk.
- **/etc/asterisk/** : Contient tous les fichiers de configuration.

  Ce dossier nous intéresse vu qu'il contient les fichiers de configuration du serveur Asterisk, parmi ces fichiers on trouve :

  - ***asterisk.conf* :** Définit certaines variables pour l'utilisation d'Asterisk. Il sert essentiellement à indiquer à Asterisk où chercher certains fichiers et certains programmes exécutables.
  - ***extensions.conf* :** Configure le comportement d'Asterisk.
  - ***iax.conf* :** Configure les conversations VoIP en utilisant le protocole Inter-Asterisk-Exchange (IAX).
  - ***rtp.conf :*** Ce fichier de configuration définit les ports à utiliser pour le protocole RTP (Real-Time Protocol). Il faut noter que les numéros listés sont des ports UDP.
  - ***sip.conf :*** Il contient les informations relatives aux téléphones sip.

Il est composé de plusieurs sections:

- une section **[general]** qui va définir les propriétés générales pour l'ensemble des clients SIP.
- une section par client SIP dans lesquelles seront définit les propriétés de chaque poste.

3. Configuration d'Asterisk

Dans un premier temps, les 2 fichiers qui vont nous intéresser sont **'sip.conf'** et **'extensions.conf'**.

- **Création des comptes utilisateurs**

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

On commence de mettre les sons par défaut en Français.

```
[general]
Language=fr
```

Plusieurs options permettent de définir et de paramétrer un client :

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)

Type => il existe 3 types d'utilisateurs :

- **user** = peut appeler mais ne peut pas recevoir d'appel
- **peer** = peut recevoir des appels
- **friend** = peut appeler et recevoir des appels

Secret => Mot de passe de l'utilisateur

Host => - **dynamic** : Le client s'enregistre auprès du serveur
- **nom d'hôte** : Nom d'hôte du client
- **adresse IP** : Adresse IP du client.

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

Voici un exemple de fichier **sip.conf** avec deux utilisateurs ***karima MECHERNENE*** et ***aissa BENCHIKH*** avec comme numéros respectifs le **6001** et le **6002**.

```
[6001]
Type=friend
Host=dynamic
fullname= karima MECHERNENE
username = karima
secret=1234
context = work
;*********************************************************************************
[6002]
Type=friend
Host=dynamic
fullname= aissa BENCHIKH
username = aissa
secret=1234
context = work
```

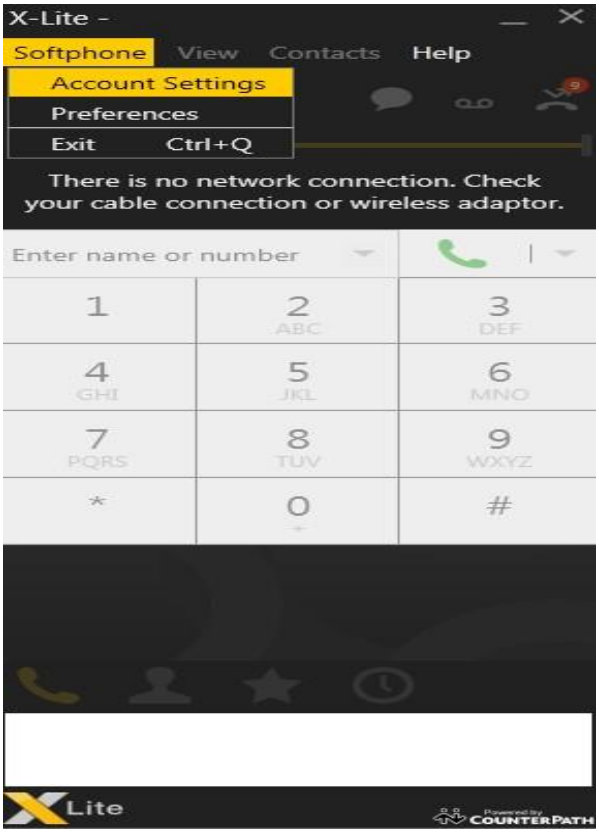
Une fois le fichier **Sip.conf** enregistré allez dans la console **Asterisk**, tapez **reload** et après on tape la commande **sip show users**, les deux comptes utilisateurs que nous venons de créer devrait y apparaître.

```
karima-HP-620*CLI> sip show users
Username                Secret          Accountcode     Def.Context     ACL  Forcerport
6002                    1234                            work            No   No
6001                    1234                            work            No   No
```

- **Configuration de X-Lite**

X-Lite est un freeware, son utilisation est simple, il est disponible pour les différents systèmes d'exploitation **Windows**, **Mac** et **Linux**. On a utilisé la version de **X-lite : 4.7.0**

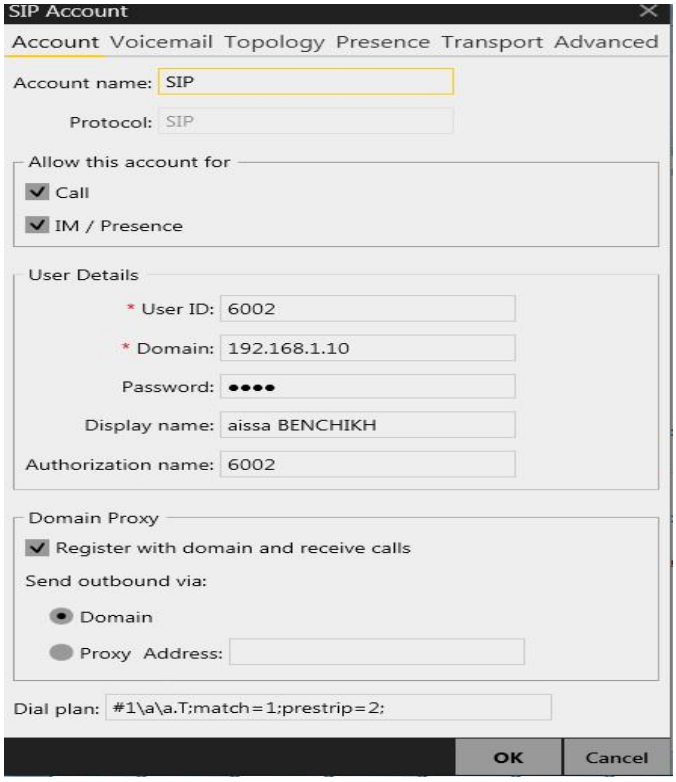
Pour configurer le client **X-Lite** l'utilisateur « **6001** » et aussi « **6002** » doivent accéder au menu « **Softphone** » puis de ce menu vers le sous menu « **Account Settings**».



Fenêtre du softphone X-lite.

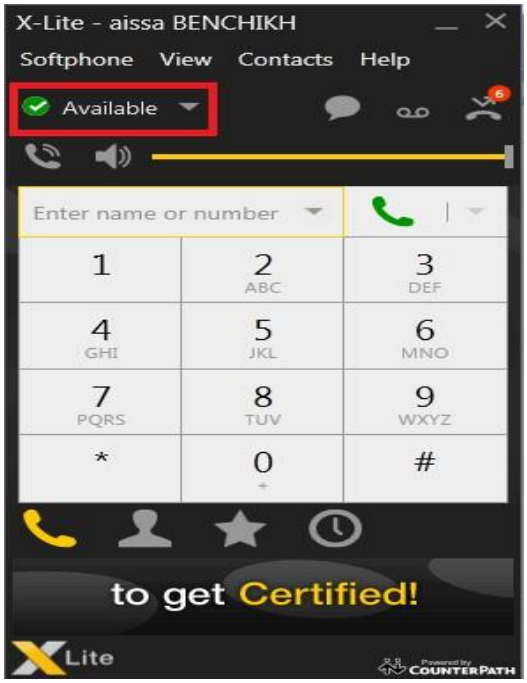
Dans la fenêtre qui s'ouvre, il suffit de remplir les champs illustré suivant des deux utilisateurs : L'utilisateur **6002** :

- **User ID** : 6002
- **Domain** : 192.168.1.10
- **Password** : 1234
- **Display name** : aissa BENCHIKH
- **Nom** sous lequel l'autorisation d'accès est possible (Authorization name) : 6002



Configuration du compte du client 6002

- **Remarque :** l'authentification soit possible, ces valeurs doivent être conformes à celles saisies dans le fichier **sip.conf** du serveur **Asterisk**. Une fois la configuration est achevée, le **softphone** se connectera automatiquement au serveur et s'enregistrera. Un message « **Available** » s'affichera, indiquant que les communications sont désormais possibles. Sinon, un message d'erreur explique le motif qui a fait échouer le processus.



Le softphone X-lite est connect

Dans la console Asterisk, on tape la commande **sip show peers**, les deux comptes utilisateurs que nous venons de créer devrait y apparaître avec leur adresse ip.

```
karima-HP-620*CLI> sip show peers
Name/username             Host                                  Dyn Forcerport Comedia    ACL Port     Status      Description

6001/karima               192.168.1.2                            D  Auto (No)  No             49240    Unmonitored

6002/aissa                192.168.1.55                           D  Auto (No)  No             45748    Unmonitored

2 sip peers [Monitored: 0 online, 0 offline Unmonitored: 2 online, 0 offline]
```

Enregistrement d'un client SIP

4. Fonctionnalités

a) Appel

- **Configuration du Dialplan**

Nous allons donc configurer **Asterisk** de telle sorte que l'utilisateur **6001** puisse appeler le numéro **6002**.

La syntaxe du fichier **extensions.conf** est sous le format suivant :

Exten => extension, priorité, commande (paramètre)

- ***Exten :*** permet de définir une nouvelle extension.
- ***Extension*** : C'est généralement le numéro de téléphone ou le nom du client.
- ***Priorité*** : C'est un numéro qui indique la priorité de la commande, le serveur prend en considération la priorité de la commande en utilisant le numéro inscrit dans la syntaxe.
- ***Commande*** : C'est la commande qui peut exister, comme la commande dial (appel), voicemail (boîte vocale), etc.

On peut utiliser plusieurs options pour un seul numéro d'appel, on peut mettre par exemple un transfert d'appel vers un autre numéro ou vers la boite vocale selon des priorités.
Voici donc le contenu du notre fichier **extensions.conf**

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

Donc la ligne : **exten => _6XXX,1,Dial(SIP/${EXTEN},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 réponses on passe à la ligne du dessous.

Dans le cas du numéros **6001** la ligne devient comme ceci : **exten => 6001,1,Dial(SIP/6001,20)**, mais l'avantage de la ligne précédente est qu'elle permet d'appeler les numéros de 6000 a 6999.
La seconde ligne : **exten => _6XXX,2,Hangup()** permet de raccrocher si il n'y a pas de réponses au bout des 20 secondes.
Dans chaque ajout ou modification d'un client, il faut mettre à jour le serveur **Asterisk** en utilisant la commande suivante dans la console : **reload**
Pour faire les tests, il suffit d'appeler par exemple à partir du compe de karima MECHERNENE (**6001**) le numéro **6002**.

Simulation d'un appel

b) Mise en place des boîtes vocales

On va éditer le fichier /etc/asterisk/voicemail.conf

Voici le contenu du fichier :

```
[general]
format=wav49|gsm|wav
maxsilence=10
silencethreshold=128
maxlogins=3
[work]
6001 => ,karima MECHERNENE
6002 => ,aissa BENCHIKH
```

- **format :** Il est possible de définir les différents formats dans lesquels seront sauvegardés les messages vocaux. (plusieurs formats séparées par un | permettent de sauvegarder dans plusieurs format)
- **[work]** : Contexte dans lequel se trouvent nos utilisateurs
- **6001 =>** : Numéro de l'utilisateur

- **,** : on n'utilise pas un mot de passe pour accéder à la boite vocale
- **karima MECHERNENE**: Prénom et nom de l'utilisateur

Maintenant nous allons éditer le fichier **extensions.conf** pour configurer deux choses :

- après certain temps **Asterisk** bascule sur la boite vocale de l'utilisateur si celui-ci ne répond pas
- Création d'une extension pour créer le numéro qui servira à consulter la boite vocale.

Voici le fichier **extensions.conf** modifié :

```
[work]
exten => _6XXX,1,Dial(SIP/${EXTEN},20)
exten => _6XXX,2,VoiceMail(${EXTEN}@work)
;Numéro de la boite vocale
exten => 600,1,VoiceMailMain(${CALLERID(num)}@work)
```

On a remplacé la ligne : **exten => _6XXX,2,Hangup()**

Par : **exten => _6XXX,2,VoiceMail(${EXTEN}@work)**

Cette ligne permet donc de basculer sur la boîte vocale de l'utilisateur appelé s'il n'y a pas de réponse au bout de 20 secondes. Explication de cette ligne:

exten => : On déclare l'extensions

_6XXX : Toujours les numéros de 6000 à 6999

2 : Priorité

VoiceMail: On lance l'application **Voicemail**

${EXTEN}@work : On récupère la numéro mis en variable et on se connecte a la boite vocale associé du contexte work (comme précisé dans le fichier **voicemail.conf**)

On a aussi rajouté la ligne suivante:

exten => 600,1,VoiceMailMain(${CALLERID(num)}@work)

Cette ligne sert à mettre en place un numéro pour que nos utilisateurs puissent consulter leurs boîtes vocales.

exten => : On déclare l'extension

600 : On déclare le numéro 600 comme numéro de messagerie

1 : Priorité

VoiceMailMain : On appelle l'application VoiceMailMain

${CALLERID(num)} : On récupère le numéros de l'utilisateur qui appelle pour se connecter à sa boîte vocale

@work : on est toujours dans le contexte work

Lorsqu'on fait un **reload** de notre serveur **Asterisk**, on peut déposer des messages vocaux entre nos utilisateurs et faires en sortes qu'ils consultent les messageries en appelant le **600**.

5. Conclusion

Dans ce chapitre nous avons présenté l'environnement matériel du travail, ainsi que les différents logiciels open sources et leurs configurations adoptées pour mettre en place les services de voix sur IP (**VoIP)** sous la plate-forme **Astérisk**. Cette étape nécessite la compréhension du principe de routage des appels à travers ce serveur.

Chapitre III

Différents risques et méthodes de sécurité de VoIP

1. Introduction

La **VoIP**, actuellement en plein développement, pouvant faire sujet d'attaque des pirates informatiques, au niveau spécifique des réseaux **IP** ou la **VoIP** proprement dite.

Celle-ci présente un nombre de vulnérabilités en terme : de protocole, de logiciel, de système d'exploitation, d'infrastructure physique et d'erreur humaine.

Il convient d'étudier avec précaution ses vulnérabilités dans le but d'établir une protection efficace contre les attaques.

Pour faire face à ces attaques, la sécurité du réseau **VoIP** doit s'appuient sur deux types de sécurités :

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

2. Principaux risques

Cette étape permet de déterminer les menaces et d'identifier les sources de menaces. Une menace est une violation potentielle de la politique de service à cause d'une attaque ou d'une action involontaire ou négligente qui compromet la sécurité. **[9]**

Les sources de menaces 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. **[10]**

A. Attaques sur le protocole

Les types d'attaques les plus fréquentes contre un system VoIP sont : **[10]**

1. Sniffing

Un **reniflage** (**Sniffing**) peut avoir comme conséquence un vol d'identité et la révélation d'informations confidentielles. Il permet également aux utilisateurs malveillants perfectionnés de rassembler des informations sur les systèmes VoIP.

2. Suivie des appels

Appelé aussi Call tracking, cette attaque cible les terminaux (soft/hard phone). Elle a pour but de connaître qui est en train de communiquer et quelle est la période de la communication. L'attaquant doit récupérer les messages **INVITE** et **BYE** en écoutant le réseau et peut ainsi savoir qui communique, à quelle heure, et pendant combien de temps.

Pour réaliser cette attaque, L'attaquant doit être capable d'écouter le réseau et récupérer les messages INVITE et BYE.

3. Injection de paquet RTP

Cette attaque pour but de perturber une communication en cours. L'attaquant devra tout d'abord écouter un flux RTP de l'appelant vers l'appelé, analyser son contenu et générer un paquet RTP contenant un en-tête similaire mais avec un plus grand numéro de séquence et **timestamp** afin que ce paquet soit reproduit avant les autres paquets (s'ils sont vraiment reproduits). Ainsi la communication sera perturbée et l'appel ne pourra pas se dérouler correctement.

Pour réaliser cette attaque, l'attaquant doit être capable d'écouter le réseau afin de repérer une communication et ainsi repérer les timestamps des paquets RTP.

Il doit aussi être capable d'insérer des messages RTP qu'il a généré ayant un timestamp modifié.

4. Le déni de service (DOS : Denial of service)

C'est, d'une manière générale, l'attaque qui vise à rendre une application informatique ou un équipement informatique incapable de répondre aux requêtes de ses utilisateurs et donc hors d'usage. Une machine serveur offrant des services à ses clients (par exemple un serveur web) doit traiter des requêtes provenant de plusieurs clients.

Lorsque ces derniers ne peuvent en bénéficier, pour des raisons délibérément provoquées par un tiers, il y a déni de service.

Dans une attaque de type **DoS flood attack**, 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.

Une attaque de type DoS peut s'effectuer à plusieurs niveaux soit:

Couche réseau :

- **IP Flooding** : Le but de l'**IP Flooding** est d'envoyer une multitude de paquets IP vers une même destination de telle sorte que le traitement de ces paquets empêche une entité du réseau (un routeur ou la station destinatrice) de traiter les paquets IP légitimes. Si l'IP Flooding est combiné à l'IP Spoofing, il est impossible, pour le destinataire, de connaître l'adresse source exacte des paquets IP.

De ce fait, à moins que le destinataire ne limite ses échanges avec certaines stations, il lui est impossible de contrer ce type d'attaques

Couche transport :

- **UDP Flooding Attacks**: Le principe de cette attaque est qu'un attaquant envoie un grand nombre de requêtes **UDP** vers une machine. Le trafic **UDP** étant prioritaire sur le trafic **TCP**, ce type d'attaque peut vite troubler et saturer le trafic transitant sur le réseau et donc de perturbe le plus la bande passante.

Presque tous les dispositifs utilisant le protocole **SIP** fonctionnent au-dessus du protocole **UDP**, ce qui en fait d'elles des cibles. De nombreux dispositifs de **VoIP** et de systèmes d'exploitation peuvent être paralysés grâce à des paquets **UDP Flooding** visant l'écoute du port **SIP** (**5060**) ou d'autres ports.

- **TCP SYN floods :** est une attaque visant le protocole **TCP** et plus exactement la phase d'établissement de connexion. Celle-ci consiste en trois sous étapes :

1. Le client envoie un paquet **SYN** au serveur.
2. Le serveur répond avec un paquet **SYN-ACK**.
3. Le client envoie un paquet **ACK** au serveur.

L'attaque consiste en l'envoie d'un grand nombre de paquets **SYN**. La victime va alors répondre par un message **SYN-ACK** d'acquittement. Pour terminer la connexion **TCP**, la victime ensuite va attendre pendant une période de temps la réponse par le biais d'un paquet ACK. C'est là le cœur de l'attaque parce que les ACK final ne sont jamais envoyés, et par la suite, la mémoire système se remplit rapidement et consomme toutes les ressources disponibles à ces demandes non valides. Le résultat final est que le serveur, le téléphone, ou le routeur ne sera pas en mesure de faire la distinction entre les faux SYN et les SYN légitimes d'une réelle connexion **VoIP**.

Couche applications :

- **SIP Flooding** : Dans le cas de **SIP**, une attaque **DoS** peut être directement dirigée contre les utilisateurs finaux ou les dispositifs tels que téléphones IP, routeurs et proxy SIP, ou contre les serveurs concernés par le processus, en utilisant le mécanisme du protocole SIP ou d'autres techniques traditionnelles de **DoS**.

Voyons maintenant en détail les différentes formes d'attaque **DoS** :

- **DoS-CANCEL** : C'est un type de déni de service lancé contre l'utilisateur. L'attaquant surveille l'activité du proxy SIP et attend qu'un appel arrive pour un utilisateur spécifique. Une fois que le

dispositif de l'utilisateur reçoit la requête INVITE, l'attaquant envoie immédiatement une requête CANCEL. Cette requête produit une erreur sur le dispositif de l'appelé et termine l'appel. Ce type d'attaque est employé pour interrompre la communication.

- **DoS-REGISTER :** Le serveur d'enregistrement lui-même est une source potentielle de déni de service pour les utilisateurs. En effet ce serveur peut accepter des enregistrements de tous les dispositifs. Un nouvel enregistrement avec une «*» dans l'entête remplacera tous les précédents enregistrements pour ce dispositif.

C'est un mécanisme très pratique pour les utilisateurs mais également pour les pirates.

5. Détournement d'appel (Call Hijacking)

Le **Call Hijacking** consiste à détourner un appel. Plusieurs fournisseurs de service VoIP utilisent le web comme interface permettant à l'utilisateur d'accéder à leur système téléphonique.

Un utilisateur authentifié peut changer les paramètres de ses transferts d'appel à travers cette interface web. C'est peut être pratique, mais un utilisateur malveillant peut utiliser le même moyen pour mener une attaque.

Exemple: quand un agent **SIP** envoie un message **INVITE** pour initier un appel, l'attaquant envoie un message de redirection 3xx indiquant que l'appelé s'est déplacé et par la même occasion donne sa propre adresse comme adresse de renvoi. A partir de ce moment, tous les appels destinés à l'utilisateur sont transférés et c'est l'attaquant qui les reçoit.

Un appel détourné en lui-même est un problème, mais c'est encore plus grave quand il est porteur d'informations sensibles et confidentielles.

6. L'écoute clandestine

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

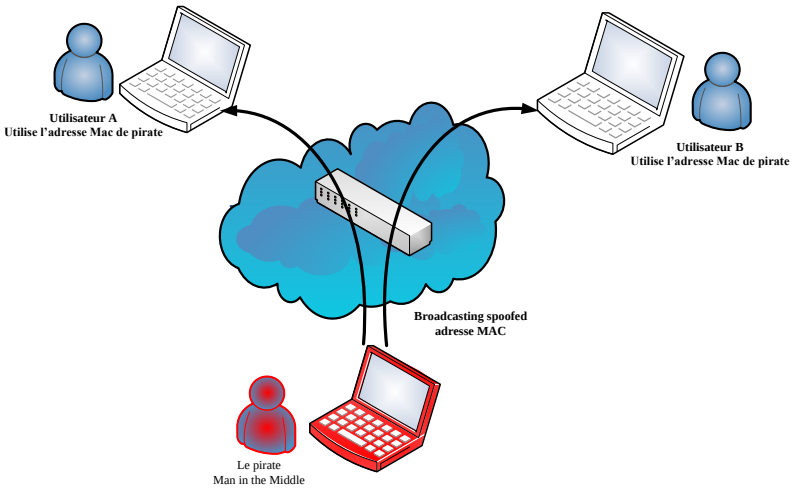


Figure III.1 : **Exemple de détournement d'appel " Man in the middle"** [10]

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

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

B. Les vulnérabilités de l'infrastructure

Une infrastructure VoIP est composée de téléphones IP, Gateway, serveurs. Ces derniers tournant sur un système d'exploitation, est accessible via le réseau comme n'importe quel ordinateur et comportent un processeur qui exécute des logiciels qui peuvent être attaqués ou employés en tant que points de lancement d'une attaque plus profonde.

1. Faiblesses de configuration des dispositifs VoIP

Plusieurs dispositifs de la VoIP, dans leur configuration par défaut, peuvent avoir une variété de ports TCP et UDP ouverts. Les services fonctionnant sur ces ports peuvent être vulnérables aux attaques DoS ou buffer overflow.

Plusieurs dispositifs de la VoIP exécutent également un serveur WEB pour la gestion à distance qui 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 acquérir un accès non autorisé à ce dispositif.

Les services SNMP (Simple Network Management Protocol) offerts par ces dispositifs peuvent être vulnérables aux attaques de reconnaissance ou attaques d'overflow.

Plusieurs dispositifs de la VoIP sont configurés pour télécharger périodiquement un fichier de configuration depuis un serveur par TFTP ou d'autres mécanismes. Un attaquant peut potentiellement détourner ou mystifier cette connexion et tromper le dispositif qui va télécharger un fichier de configuration malveillant à la place du véritable fichier.

2. Les téléphones IP

Un pirate peut compromettre un dispositif de téléphonie sur IP, par exemple un téléphone IP, un softphone et autres programmes ou matériels clients. Généralement, il obtient les privilèges qui lui permettent de commander complètement la fonctionnalité du dispositif.

Compromettre un point final (téléphone IP) peut être fait à distance ou par un accès physique au dispositif. Le pirate pourrait modifier les aspects opérationnels d'un tel dispositif:

La pile du système d'exploitation peut être changée. Ainsi la présence de l'attaquant ne sera pas remarquée.

Aussi un firmware modifié de manière malveillante peut être téléchargé et installé. Les modifications faites à la configuration des logiciels de téléphonie IP peuvent permettre:

- Aux appels entrants d'être réorientés vers un autre point final sans que l'utilisateur soit au courant.
- Aux appels d'être surveillés.
- A l'information de la signalisation et/ou les paquets contenant de la voix d'être routés vers un autre dispositif et également d'être enregistrés et/ou modifiés.

De compromettre la disponibilité du point final. Par exemple, ce dernier peut rejeter automatiquement toutes les requêtes d'appel, ou encore, éliminer tout déclenchement de notification tel qu'un son, une notification visuelle à l'arrivée d'un appel. Les appels peuvent également être interrompus à l'improviste (quelques téléphones IP permettent ceci via une interface web).

Toutes les informations concernant l'utilisateur qui sont stockées sur le dispositif pourraient été extraites.

L'acquisition d'un accès non autorisé sur un dispositif de téléphonie IP peut être le résultat d'un autre élément compromis sur le réseau IP, ou de l'information récoltée sur le réseau.

Les softphones ne réagissent pas de la même façon aux attaques comparés à leur homologues téléphones IP. Ils sont plus susceptibles aux attaques dues au nombre de vecteurs inclus dans le système, à savoir les vulnérabilités du système d'exploitation, les vulnérabilités de l'application, les vulnérabilités du service, des vers, des virus, etc. En plus, le softphone demeure sur le segment de données, est ainsi sensible aux attaques lancées contre ce segment et pas simplement contre l'hôte qui héberge l'application softphone.

Les téléphones IP exécutent quant à eux leurs propres systèmes d'exploitation avec un nombre limité de services supportés et possèdent donc moins de vulnérabilités.

3. Les serveurs

Un pirate peut viser les serveurs qui fournissent le réseau de téléphonie sur IP.

Compromettre une telle entité mettra généralement en péril tout le réseau de téléphonie dont le serveur fait partie.

Par exemple, si un serveur de signalisation est compromis, un attaquant peut contrôler totalement l'information de signalisation pour différents appels. Ces informations sont routées à travers le serveur compromis. Avoir le contrôle de l'information de signalisation permet à un attaquant de changer n'importe quel paramètre relatif à l'appel.

Si un serveur de téléphonie IP est installé sur un système d'exploitation, il peut être une cible pour les virus, les vers, ou n'importe quel code malveillant.

4. Vulnérabilités du système d'exploitation

Ces vulnérabilités sont pour la plupart relatives au manque de sécurité lors de la phase initiale de développement du système d'exploitation et ne sont découvertes qu'après le lancement du produit.

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

Les dispositifs de la 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 tournent.

Il existe une centaine de vulnérabilités exploitables à distance sur Windows et même sur Linux. Un grand nombre de ces exploits sont disponibles librement et prêts à être téléchargés sur l'Internet.

Peu importe comment, une application de la VoIP s'avère être sûre, celle-ci devient menacé si le système d'exploitation sur lequel elle tourne est compromis.

3. Eléments de sécurité

On a déjà vu que les vulnérabilités existent au niveau protocolaire, application et systèmes d'exploitation. Pour cela, on a découpé la sécurisation aussi en trois niveaux : Sécurisation protocolaire, sécurisation de l'application et sécurisation du système d'exploitation. **[10]**

1. Sécurisation protocolaire

La prévalence et la facilité de sniffer des paquets et d'autres techniques pour la capture des paquets IP sur un réseau pour la voix sur IP fait que le cryptage soit une nécessité. La sécurisation de la VoIP est à la protection des personnes qui sont interconnecté.

IPsec peut être utilisé pour réaliser deux objectifs. Garantir l'identité des deux points terminaux et protéger la voix. VOIPsec (VoIP utilisant IPsec) contribue à réduire les menaces, les sniffeurs de paquets, et de nombreux types de trafic « vocal analyze ». Combiné avec un pare-feu, IPsec fait que la VOIP soit plus sûr qu'une ligne téléphonique classique. Il est important de noter, toutefois, que IPsec n'est pas toujours un bon moyen pour certaines applications, et que certains protocoles doivent continuer à compter sur leurs propres dispositifs de sécurité.

a. VoIP VPN

Un VPN VoIP combine la voix sur IP et la technologie des réseaux virtuels privés pour offrir une méthode assurant la préservation de la prestation vocale. Puisque la VoIP transmet la voix numérisée en un flux de données, la solution VPN VoIP semble celle la plus approprié vu qu'elle offre le cryptage des données grâces a des mécanismes de cryptages, puisqu'elle permet d'offrir l'intégrité des paquets VoIP.

Cryptage aux points terminaux : Vu que notre objectif est d'assurer la confidentialité et l'intégrité des clients, la nécessité de concevoir des mécanismes d'authentifications et

de chiffrement pour IP. Puisqu'il sécurise le paquet comme un tout (contrairement en mode transport qui ne sécurise que le **Payload IP**). Le mode tunnel (réseau privé virtuel sécurisé)**[10],** se base sur l'encapsulation de tout le paquet IP et ajoute un nouvel entête pour l'acheminement de ce dernier, et l'authenticité des paquets reçus assurée par l'utilisation d'**IPSec** (Internet Protocol Security) entre les machines concernées.

Ce mode est généralement utilisé pour les routeur-to-routeur (**Figure III.2)**. Ses avantages est sa possibilité de l'utiliser uniquement sur des communications spécifiques (sans perturber les autres communications), et puisque **IPSec** est au-dessous de la couche de transport (TCP, UDP); il est donc

transparent aux applications (possibilité d'accroître la sécurité sans modifier les applications de plus haut niveau). Une fois mis en place, IPSec est transparent aux utilisateurs.

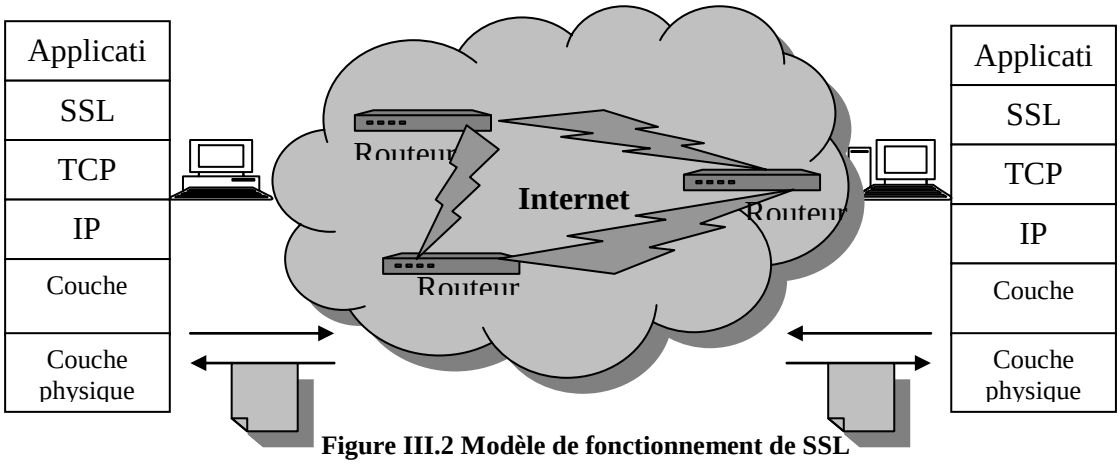


Figure III.2 Modèle de fonctionnement de SSL

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

On a choisi le protocole **ESP** qui lui a son tour va assurer le cryptage des données et donc la confidentialité contrairement au protocole AH qui lui ne permet que l'authentification des paquets et non le cryptage.

Dans ce cas, la solution qu'on propose est ESP mode tunnel qui sera appliqué uniquement sur les points de terminaison à la voix IP, c'est-a-dire le routeur. Ceci nous permettra donc de minimiser le nombre de machines qui seront impliquées dans le traitement engendré par la sécurité. De plus le nombre des clés nécessaires sera réduit.

b. Protocole TLS

C'est un protocole de sécurisation des échanges au niveau de la couche transport (**TLS : Transport Layer Security**). TLS, anciennement appelé **Secure Sockets Layer** (**SSL**), est un

protocole de sécurisation des échanges sur Internet. C'est un protocole modulaire dont le but est de sécuriser les échanges des données entre le client et le serveur indépendamment de tout type d'application. TLS agit comme une couche supplémentaire au-dessus de TCP. **[9]**

Le protocole SSL et TLS est subdivisé en quatre sous protocoles (**Figure III.3)** :

- **Le protocole Hanshake :** C'est un protocole qui permet au client et au serveur de s'authentifier mutuellement, de négocier les algorithmes de chiffrement, de négocier les algorithmes de MAC (Message Authentification Code) et enfin de négocier les clés symétriques qui vont servir au chiffrement.
- **Le protocole Change Cipher Spec :** Ce protocole contient un seul message : change_cipher_spec. Il est envoyé par les deux parties au 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.
- **Le protocole Record :** Ce protocole chapeaute les autres protocoles de **SSL** et **TLS**, en fournissant une interface unifiée pour la transmission des données.

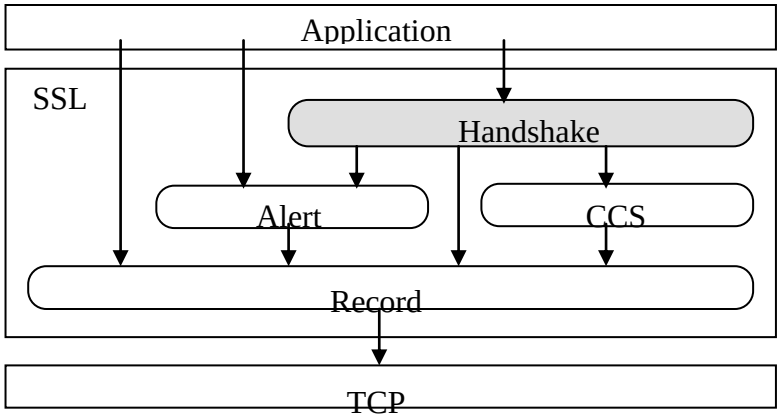


Figure III.3 : Empilement des sous-couches protocolaires de SSL

- **Rôle**

Encapsulation : Permet aux données SSL et TLS d'être transmises et reconnues sous une forme homogène.

Confidentialité : Assure que le contenu du message ne peut pas être lu par un tiers : les données sont chiffrées en utilisant les clés produites lors de la négociation.

Intégrité et Identité : Permet de vérifier la validité des données transmises, grâce aux signatures MAC: cette signature est elle aussi générée 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 III.4** donne l'ensemble des algorithmes supportés par SSL tandis que **tableau III.5** 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 40 bits
Chiffrement symétrique en blocs	DES, DES40, 3DES RC2, IDEA, Fortezza
Hachage	MD5, SHA

Table III.4 Algorithmes négociés par le protocole Handshake

Echange de clefs	*Chiffrement symétrique*	*Hachage*	*Signature*
	Sans chiffrement RC4-40	MD5 ou SHA MD5	
RSA	RC4-128 RC2 CBC 40 IDEA CBC DES40 CBC DES CBC 3DES EDE CBC	MD5 ou SHA MD5 SHA SHA SHA SHA	
Diffie-Hellman	DES40 CBC	SHA	DSS ou RSA

	DES CBC 3DES, EDE CBC	SHA SHA	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

Table III.5 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.

c. Secure RTP (SRTP)

SRTP est conçu pour sécuriser la multiplication à venir des échanges multimédias sur les réseaux. 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).

- **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.

- **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 a 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.

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.

- **Format du paquet SRTP**

Un paquet SRTP est généré par transformation d'un paquet RTP grâce à des mécanismes de sécurité (**Figure III.6)**. 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

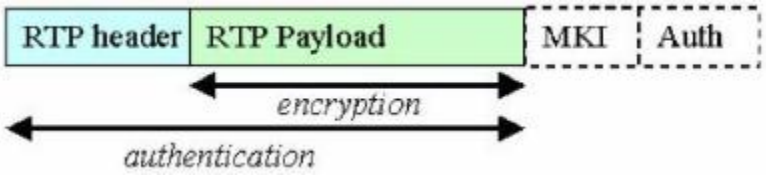


Figure III.6 : Format d'un paquet SRTP

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.

2. L'authentification

L'une de méthode les plus importantes pour anticiper une attaque sur un système de téléphonie est de déterminer clairement l'identité des périphériques ou des personnes participant à la conversation. Plusieurs solutions simples sont mises en œuvre pour cela, il est recommandé d'utiliser des mots de passe complexes lors de la configuration des clients **SIP** ; en effet, il faut savoir que certains hackers développent des robots en charge de sonder les réseaux informatiques et dés que l'un d'entre eux réponds au protocole SIP, un algorithme sophistiqué est engagé et teste toutes les combinaisons possibles de mots de passe. Ainsi, il faut éviter

• Les mots de passes trop courts

• Les suites numériques (123456) ou alphabétiques (abcd)

• Les suites logiques tels prénoms ou 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

On ne saurait trop recommander un mot de passe complètement aléatoire de 8 caractères au minimum, faisant intervenir une combinaison de caractères spéciaux, lettres majuscules, lettres minuscules, chiffres non suivis. A proscrire, l'utilisation de 1 et de l (L minuscule) ainsi que de 0 (zéro) et O de Oscar.

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

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.

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.

- Asterisk in CHROOT : empêcher le serveur VoIP d'avoir une visibilité complète de l'arborescence du disque, en l'exécutant dans un environnement sécurisé qui l'empêche d'interagir librement avec le système.

- Sauvegarde des fichiers log à distance : les fichiers log sont très importants, il est conseillé de les enregistrer sur un serveur distant.
- Installer seulement les composants nécessaires : pour limiter les menaces sur le système d'exploitation. Il vaut mieux installer sur la machine le système d'exploitation et le serveur.
- Supprimer tous programmes, logiciels ou des choses qui n'ont pas d'importance et qui peuvent être une cible d'attaque pour accéder au système.
- Renforcer la sécurité du système d'exploitation en installant des patches qui permettent de renforcer la sécurité générale du noyau.

On peut aussi utiliser les pare feu ou/et les ACL pour limiter l'accès à des personnes bien déterminé et fermer les ports inutiles et ne laisser que les ports utilisés (5060, 5061, 4569,…). Le pare feu (firewall) est un software ou hardware qui a pour fonction de sécuriser un réseau ou un ordinateur contre les intrusions venant d'autres machines. Le pare feu utilise le système de filtrage de paquet après analyse de l'entête des paquets IP qui s'échange entre les machines.

On aura besoin d'ACL pour donner des droits à des personnes bien déterminés selon leurs besoins et leurs autorités.

Pour un serveur VoIP, il est important d'implémenter les ACL pour sécuriser le serveur en limitant l'accès à des personnes indésirables. Par exemple, seuls les agents enregistrés peuvent envoyer des requêtes au serveur.

La liste de contrôle d'accès peut être installée en réseau sur les pare feu ou les routeurs, mais aussi ils existent dans les systèmes d'exploitation.

4. Conclusion

Dans ce chapitre, nous avons présentés les vulnérabilités les plus importants et comment doit-on procéder pour y faire faces, tout en préconisant un certain nombres de mesures de sécurités devant être prises en compte dans le but de garantir la qualité de service du réseau ainsi que sa sécurité.

Chapitre IV

Sécurisation de la solution VoIP sur Asterisk

1. Introduction

Dans ce chapitre nous réaliserons des analyses de vulnérabilités et identifier les faiblesses et les défauts des mesures de sécurité, afin de simuler et corriger les points faibles et mettre en place des mesures de sécurité qui garantit une communication fiable et robuste.

2. Attaques Simulées :

A. Attaque sur les mots de passe

- **Procédure**

Les mots de passe des clients SIP sont stockés en clair dans le serveur Asterisk, et peuvent être identifiés par plusieurs méthodes :

- Dans le fichier ***SIP.conf***, lorsqu'on définit les mots de passe des utilisateurs:

Dans notre exemple le mot de passe de client 6001 est 1234

```
[6001]
Type=friend
Host=dynamic
fullname= karima MECHERNENE
username = karima
secret=1234
context = work
```

- La commande console : **sip show users**

```
karima-HP-620*CLI> sip show users
Username                  Secret          Accountcode      Def.Context      ACL  Forcerport
6003                      1234                             work             No   No
6002                      1234                             work             No   No
6001                      1234                             work             No   No
6004                      1234                             work             No   No
```

- **Mécanismes pour sécuriser l'authentification**

- **Création d'un mot de passe crypté**

Les mots de passe sont cryptés à l'aide de la méthode MD5, utilisé comme couche de sécurité de base. Voici la structure nécessaire pour la génération du hash MD5 :

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

```
root@karima-HP-620:/etc/asterisk# echo -n "6001:asterisk:1234" | md5sum
8ae80145b304007cb19732b6aafa517c  -
root@karima-HP-620:/etc/asterisk# echo -n "6002:asterisk:1234" | md5sum
0c8f41fc699b9efc957ce1dcf059aa41  -
```

Des modifications seront nécessaires dans sip.conf pour que la configuration soit fonctionnelle.

Dans notre cas on a crypté les deux clients 6001 & 6002, donc on change la ligne **secret** par

md5secret.

```
6001]
Type=friend
Host=dynamic
fullname= karima MECHERNENE
username = karima
md5secret=8ae80145b304007cb19732b6aafa517c
context = work
;*********************************************************************
[6002]
Type=friend
Host=dynamic
fullname= aissa BENCHIKH
username = aissa
md5secret=0c8f41fc699b9efc957ce1dcf059aa41
```

Le résultat est le suivant :

```
karima-HP-620*CLI> sip show users
Username                Secret          Accountcode     Def.Context     ACL  Forcerport
6003                    1234                            work            No   No
6002                                                    work            No   No
6001                                                    work            No   No
6004                    1234                            work            No   No
```

Dans les lignes des deux clients 6001 & 6002 les mots de passe ne sont pas visibles.

- **Outil « Fail 2 Ban »**

Fail 2 Ban est un outil propre aux systèmes Linux, pour de protéger 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

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

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**.

Il faut installer Fail2Ban : **apt-get install fail2ban**

Pour ne pas se bloquer, mettre l'IP (ou une autre) en liste blanche.

Pour cela, éditer le fichier **jail.conf**, et ajouter l'IP dans **ingnoreip**.

Dans **/etc/fail2ban/jail.conf**

[DEFAULT]

ignoreip = 127.0.0.1 192.168.1.200

Ensuite, il faut modifier le format de date des logs d'Asterisk.

Dans le fichier **/etc/Asterisk/logger.conf**, documenter la ligne : **dateformat=%F %T**

Ainsi, Asterisk utilisera le format suivant pour les logs : yyyy-mm-dd HH:MM:SS

Puis redémarrer Asterisk: **Asterisk -rvv**

logger reload

A présent, nous pouvons créer le filtre Fail2Ban pour Asterisk.

Le filtre est à créer dans le fichier **/etc/fail2ban/filter.d/Asterisk.conf**.

[INCLUDES]

[Definition]

failregex = NOTICE.* .*: Registration from '.*' failed for '<HOST>:.*' - Wrong password

NOTICE.* .*: Registration from '.*' failed for '<HOST>:.*' - No matching peer found

NOTICE.* .*: Registration from '.*' failed for '<HOST>:.*' - No matching peer found

NOTICE.*.*:Registration from'.*' failed for '<HOST>:.*'-Username/auth name mismatch

NOTICE.* .*: Registration from '.*' failed for '<HOST>:.*' - Device does not match ACL

NOTICE.*.*:Registration from'.*' failed for '<HOST>:.*' -Peer is not supposed to register

NOTICE.* .*: Registration from '.*' failed for '<HOST>:.*' - ACL error (permit/deny)

NOTICE.* .*: Registration from '.*' failed for '<HOST>:.*' - Device does not match ACL

NOTICE.*.*:Registration from '\".*\".*' failed for '<HOST>:.*' - No matching peer found

NOTICE.* .*: Registration from '\".*\".*' failed for '<HOST>:.*' - Wrong password

NOTICE.* <HOST> failed to authenticate as '.*'$

NOTICE.* .*: No registration for peer '.*' \(from <HOST>\)

NOTICE.* .*: Host <HOST> failed MD5 authentication for '.*' (.*)

NOTICE.* .*: Failed to authenticate user .*@<HOST>.*

NOTICE.* .*: <HOST> failed to authenticate as '.*'

NOTICE.* .*: <HOST> tried to authenticate with nonexistent user '.*'

VERBOSE.*SIP/<HOST>-.*Received incoming SIP connection from unknown peer

ignoreregex =

Il faut maintenant créer la prison.

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 de blocage.

Dans le fichier **/etc/fail2ban/jail.conf**, ajouter la section suivante :

[Asterisk-iptables]

enabled = true

filter = Asterisk

action = iptables-allports[name=ASTERISK, protocol=all]

logpath = /var/log/Asterisk/messages

maxretry = 5

bantime = 3600

findtime = 120

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

La personnalisation de ces paramètres est libre.

Puis démarrer Fail2Ban: **/etc/init.d/fail2ban start**

```
root@karima-HP-620:/etc/fail2ban/filter.d# /etc/init.d/fail2ban start
 * Starting authentication failure monitor fail2ban                    [ OK ]
root@karima-HP-620:/etc/fail2ban/filter.d#
```

Pour tester Fail2Ban sur Asterisk, entrer la commande suivante :

fail2ban-regex /var/log/Asterisk/messages /etc/fail2ban/filter.d/Asterisk.conf

Cela va demander à Fail2Ban d'analyser le fichier de log, pour voir si le filtre détecte des erreurs de d'authentification.

Exemple : On a essayé d'authentifié par des faux mots de passe à plusieurs fois

Parmi les résultats, on retrouve la partie suivante, qui indiquera les erreurs d'authentification détectées :

```
root@karima-HP-620:/etc/fail2ban# fail2ban-regex /var/log/asterisk/messages /etc/fail2ban/filter.d/asterisk.conf

Running tests
=============

Use   failregex file : /etc/fail2ban/filter.d/asterisk.conf
Use         log file : /var/log/asterisk/messages


Results
=======

Failregex: 24 total
|-  #) [# of hits] regular expression
|   1) [12] NOTICE.* .*: Registration from '.*' failed for '<HOST>:.*' - Wrong password
|  10) [12] NOTICE.* .*: Registration from '\".*\".*' failed for '<HOST>:.*' - Wrong password
`-

Ignoreregex: 0 total

Date template hits:
|- [# of hits] date format
|  [6122] Year-Month-Day Hour:Minute:Second
|  [711] MONTH Day Hour:Minute:Second
`-

Lines: 6833 lines, 0 ignored, 12 matched, 6821 missed
Missed line(s):: too many to print.  Use --print-all-missed to print all 6821 lines
```

B. Attaque MITM (Man In The Middle) (Homme Du Milieu)

- **Procédure**

- *Pré-requis :*
    - **Machine de Hacker :** On a installé le système Backtrack sur Vmware.
    - BackTrack : est une distribution GNU/Linux basée sur la distribution Slackware, elle est née de la fusion de Whax et Auditor. Son objectif est de fournir une distribution regroupant l'ensemble des outils nécessaires aux tests de sécurité d'un réseau.**[9]**
    - **Ettercap :** Il est capable d'accomplir des attaques sur le protocole ARP pour se positionner en tant que "man in the middle". Une fois placé en tant que tel, il permet:

  - De faire transiter le flux par l'ordinateur pirate
  - d'infecter, de remplacer et de supprimer des données dans une connexion
  - de découvrir des mots de passe pour des protocoles comme FTP, HTTP, SSH1… **[9]**

- *Topologie du réseau :* On doit ajouter une machine pour le hacker (**Figure IV.1)**

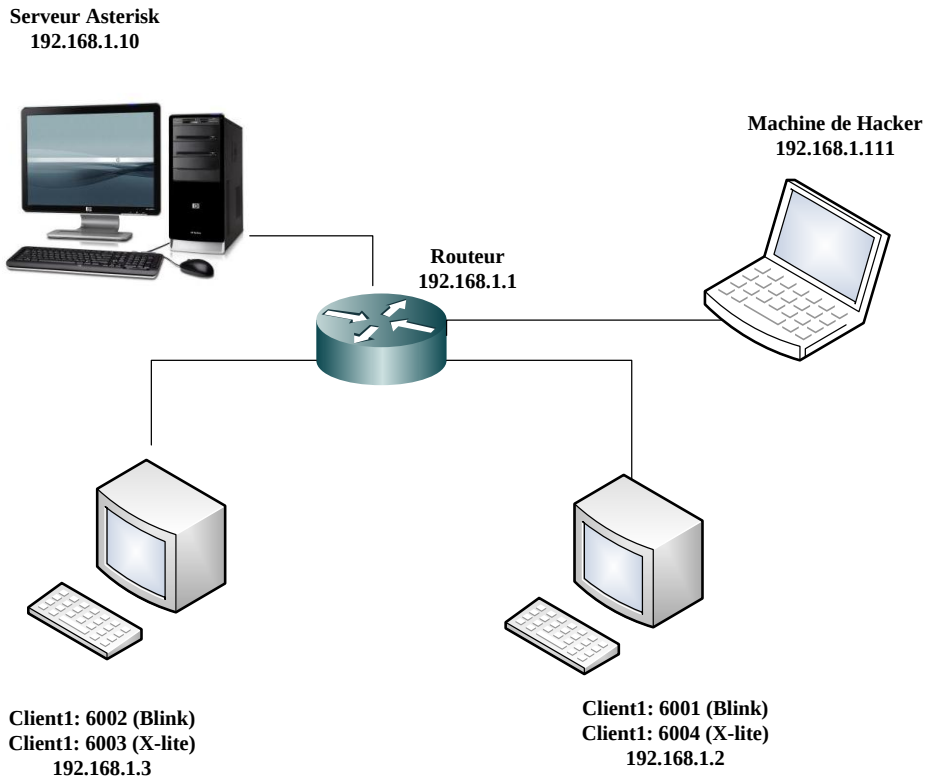


Figure IV.1 : Intrusion initial du Hacker

On remarque que chaque hôte à une adresse mac différent aux autres.

On lance ettercap en mode graphique afin de scanner le réseau (**Figure IV.2)** avec la commande suivante : **ettercap –G**

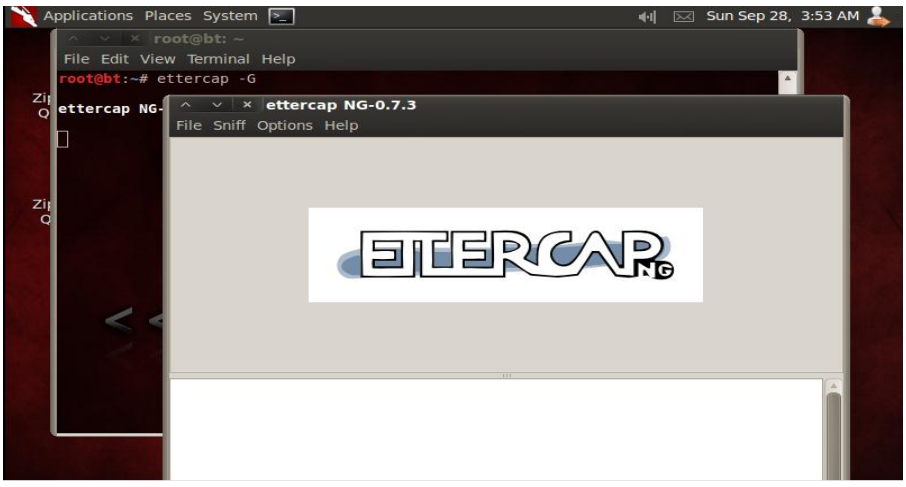


Figure IV.2 : Lancement d'outil ETTERCAP

Puis nous choisissons notre interface réseau pour lancer le Sniffing. Pour se faire on clique l'onglet **Sniff** puis en sélectionnant **Unified sniffing (Figure IV.3).**

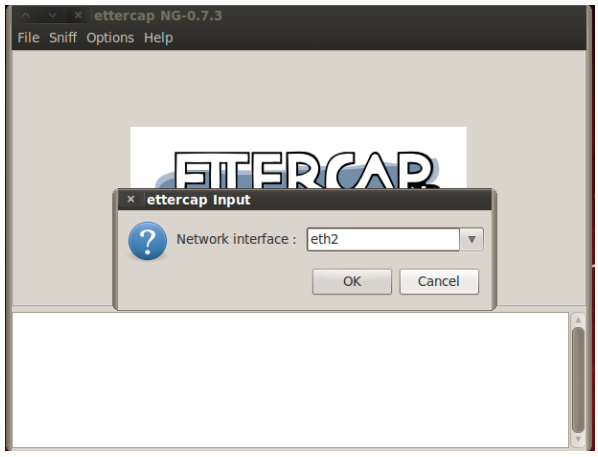


Figure IV.3 : Le choix de la passerelle ou l'interface du réseau.

Scan du réseau et ajouts d'hôtes :

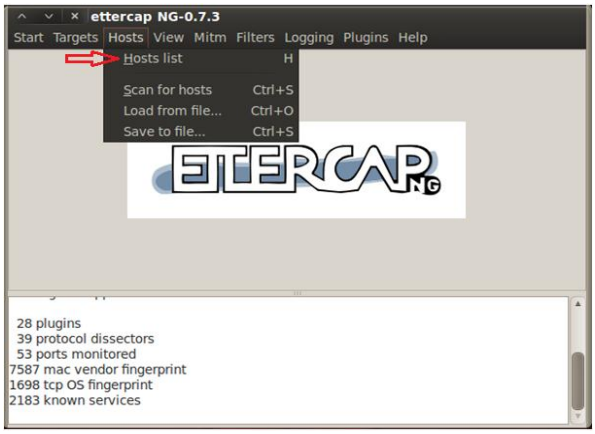
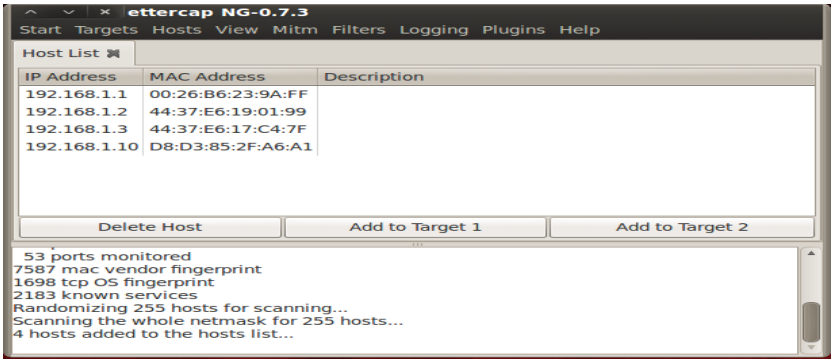
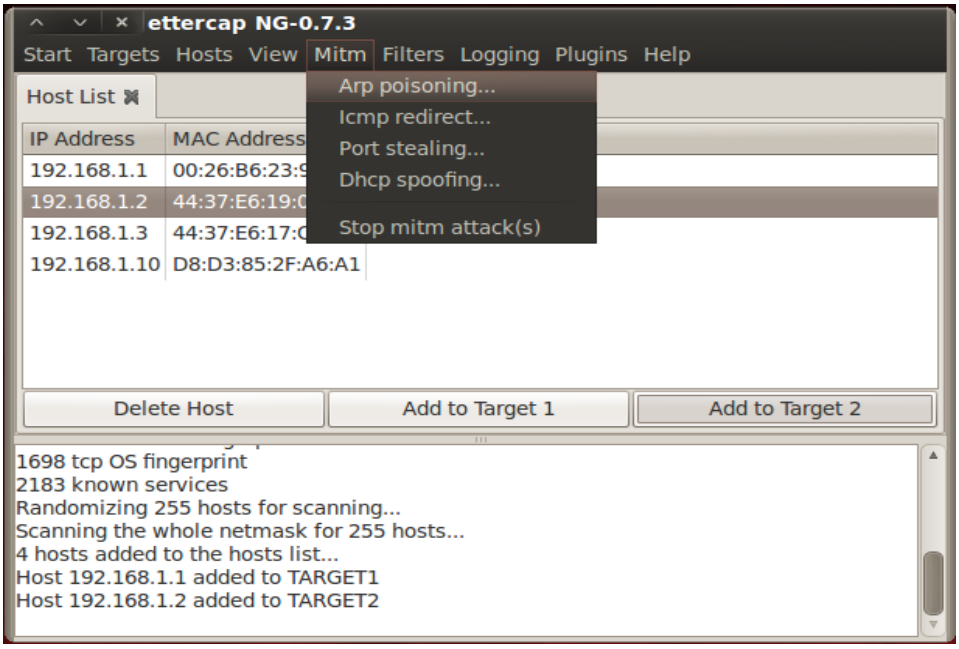


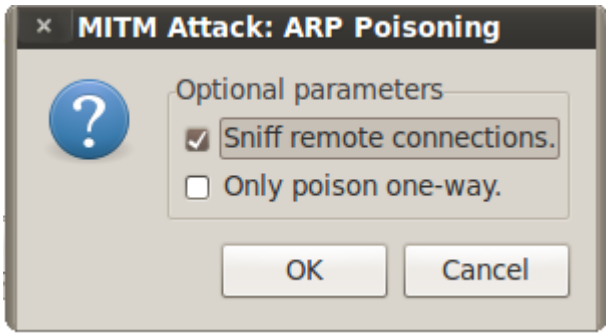
Figure IV.4 : Commencement du sniffing

Une fois la découverte d'hosts terminée, retourner dans l'onglet **Hosts** et sélectionner **Hosts list**, cette étape permet de visualiser les différentes machines connectées au réseau.

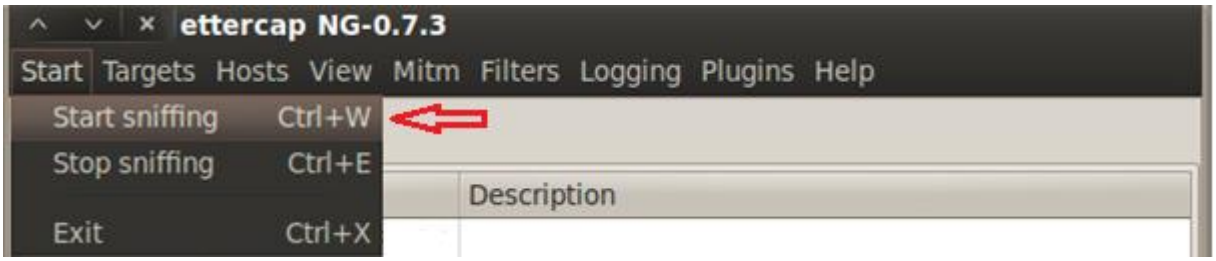


Choisir le type de l'attaque :





Début du Sniffing :



Après avoir effectué ces étapes l'architecture du réseau devient comme suit (**Figure IV.5)**

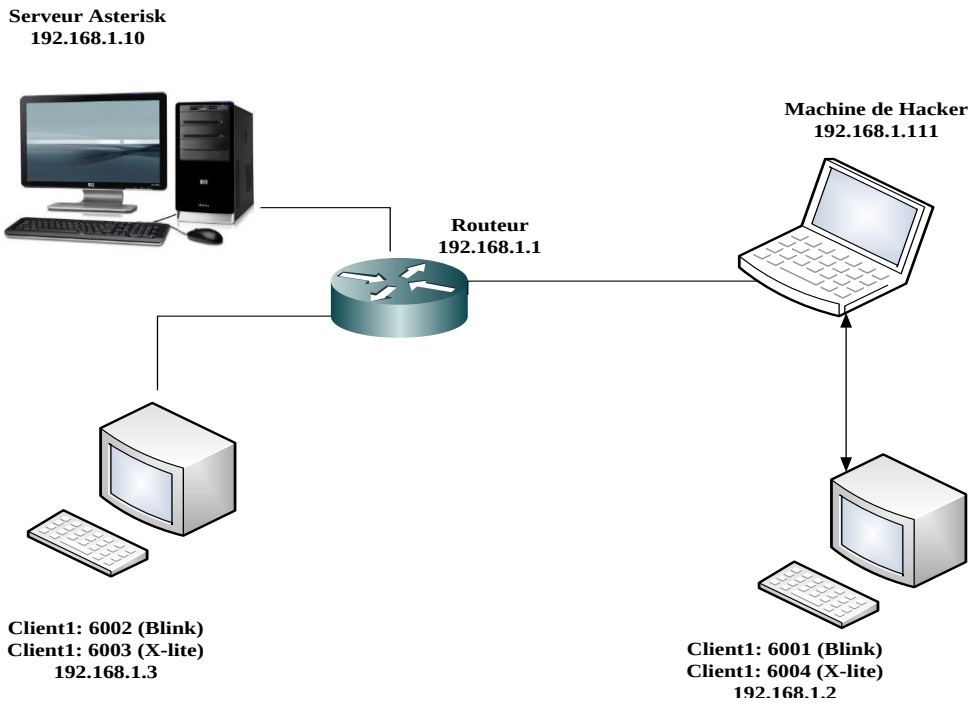


Figure IV.5 : Le hacker (intru) a pu jouer le rôle du serveur

L'attaque Man in the Middle est réalisée et la machine hacker identifiée par l'IP 192.168.1.111 est placée entre le serveur d'Asterisk identifié par l'adresse IP 192.168.1.10 et la machine identifiée par l'adresse IP 192.168.1.2.

Le Pirate à l'adresse MAC **00-0c-29-8c-1e-88** et le serveur d'Asterisk à l'adresse MAC **d8-d3-85-2f-a6-a1**, la table adresse MAC chez le client 192.168.1.2 ressemble donc à cela :

```
C:\Users\karima>arp -a

Interface : 192.168.1.2 --- 0xa
  Adresse Internet      Adresse physique      Type
  192.168.1.1           00-26-b6-23-9a-ff     dynamique
  192.168.1.3           44-37-e6-17-c4-7f     dynamique
  192.168.1.10          d8-d3-85-2f-a6-a1     dynamique
  192.168.1.111         00-0c-29-8c-1e-88     dynamique
  192.168.1.255         ff-ff-ff-ff-ff-ff     statique
```

Figure IV.6 : Table ARP du client avant attaque

On voit donc ici les correspondances IP – MAC qui correspondent à notre schéma. Si le pirate envoie des paquets au client avec l'adresse IP source du serveur mais en laissant son adresse MAC, la table ARP de notre client va donc enregistrer le couple suivant :

192.168.1.10 - 00-0c-29-8c-1e-88

La table ARP de notre client va donc ressembler à cela :

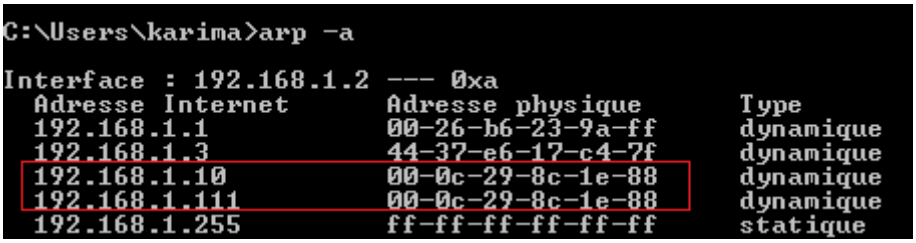


Figure IV.7 : Table ARP après l'attaque

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

C. Attaque usurpation d'identité

- **Procédure :**

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

- Pré-requis
  - **SIPVicious** est un ensemble d'outils qui peuvent être utilisés pour vérifier les systèmes de VoIP basés sur SIP. Il est composé de plusieurs outils.**[9]**

Télécharger sipvicious sur la machine hacker qui contient le système Backtrack5

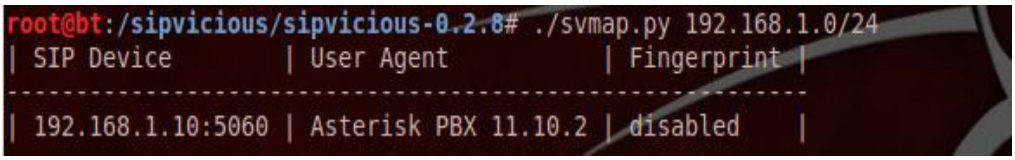
root@bt:/sipvicious# wget http://sipvicious.googlecode.com/files/sipvicious-0.2.8.tar.gz

root@bt:/sipvicious# tar xzvf sipvicious-0.2.8.tar.gz

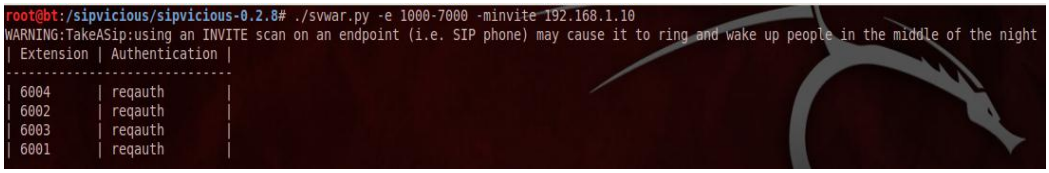
root@bt:/sipvicious# cd sipvicious-0.2.8/

root@bt:/sipvicious/sipvicious-0.2.8#

Svmap : permet de scanner une plage d'adresse IP à la recherche de dispositifs VOIP.



Svwar : permet de déterminer la liste des extensions actives sur le serveur Asterisk



Svcrack : permet de cracker les mots de passes des utilisateurs.

On a utilisé une attaque par dictionnaire

```
root@bt:/sipvicious/sipvicious-0.2.8# ./svcrack.py -u6001 -d dictionnaire.txt 192.168.1.10
WARNING:root:found nothing
root@bt:/sipvicious/sipvicious-0.2.8# ./svcrack.py -u6002 -d dictionnaire.txt 192.168.1.10
WARNING:root:found nothing
root@bt:/sipvicious/sipvicious-0.2.8# ./svcrack.py -u6003 -d dictionnaire.txt 192.168.1.10
| Extension | Password |
-----------------------
| 6003      | 1234     |

root@bt:/sipvicious/sipvicious-0.2.8# ./svcrack.py -u6004 -d dictionnaire.txt 192.168.1.10
| Extension | Password |
-----------------------
| 6004      | 1234     |
```

Remarque : Les mots de passe des utilisateurs 6001 & 6002 sont crypté au paravent donc l'outil **svcrack** n'a pas réussi de cracker contrairement aux autres utilisateurs 6003 & 6004.

- **Mécanisme de sécurité**

b.1. On doit ajouter dans le fichier sip.conf les lignes suivantes :

```
[general]
allowguest=no
alwaysauthreject = yes
```

Après cette modification de sip.conf ; on lance la même commande .Dans ce cas on obtient une erreur :

```
root@bt:/sipvicious/sipvicious-0.2.8# ./svwar.py -e 1000-7000 -minvite 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
ERROR:TakeASip:SIP server replied with an authentication request for an unknown extension. Set --force to force a scan.
WARNING:root:found nothing
```

allowguest=no: Ceci interdit les connexions SIP invité. La valeur par défaut est de permettre les connexions des clients. SIP nécessite normalement l'authentification, mais peuvent être acceptés les appels des utilisateurs qui ne prennent pas en charge l'authentification.

alwaysauthreject = yes: Si cette option est activée, chaque fois que Asterisk rejette une invitation, il sera toujours rejeté en envoyant un message ' 401 Unauthorized message' au lieu de laisser l'appelant savoir s'il y avait un utilisateur correspondant pour sa demande

D. Attaque Eaveasdropping

- **Procédure**

Cette attaque est utilisée pour écouter et enregistrer les conversations entre les interlocuteurs mais aussi de récupérer un ensemble d'informations confidentielles

- Pré-requis

  Il faut réalisé préalablement une attaque de type MITM pour rediriger le trafic vers la machine de pirate

  Wireshark : C'est l'un des logiciels les plus utilisés pour ce qui est de la surveillance, intrusion et capture dans les réseaux. Non seulement il peut capturer et analyser des trames, mais aussi, dans le cas de conversations téléphoniques, convertir ces trames en fichiers audio.[9]

Lancer Wireshark (**Figure IV.8)**:

Il suffit de taper dans le terminal de la machine pirate : wireshark

Choix de l'interface réseau sur laquelle on va effectuer la capture des paquets échangés :

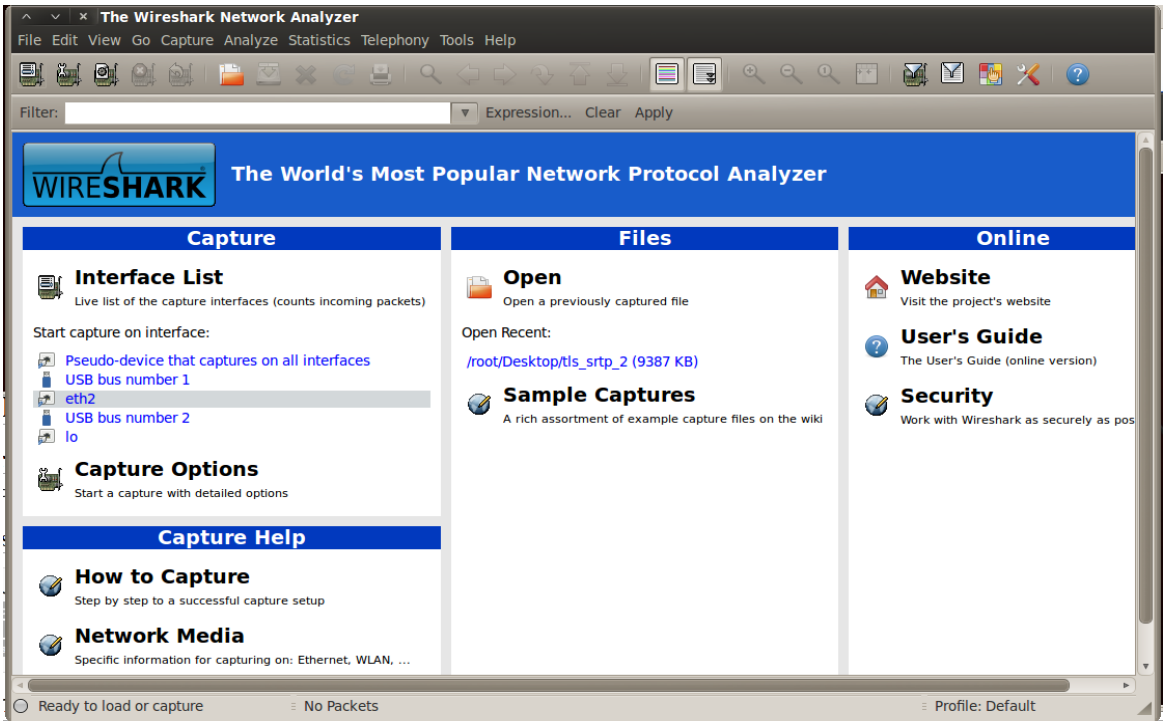


Figure IV.8 : Lancement d'outil WireShark sur la passerelle eth0

Maintenant, nous allons commencer la capture d'une partie du trafic. Pour commencer de sniffer le trafic, lorsque le client 6004 appelle le client 6003 (**Figure IV.9**).

Dans wireshark, filtrer les paquets à capturer et se limiter au protocole « RTP »

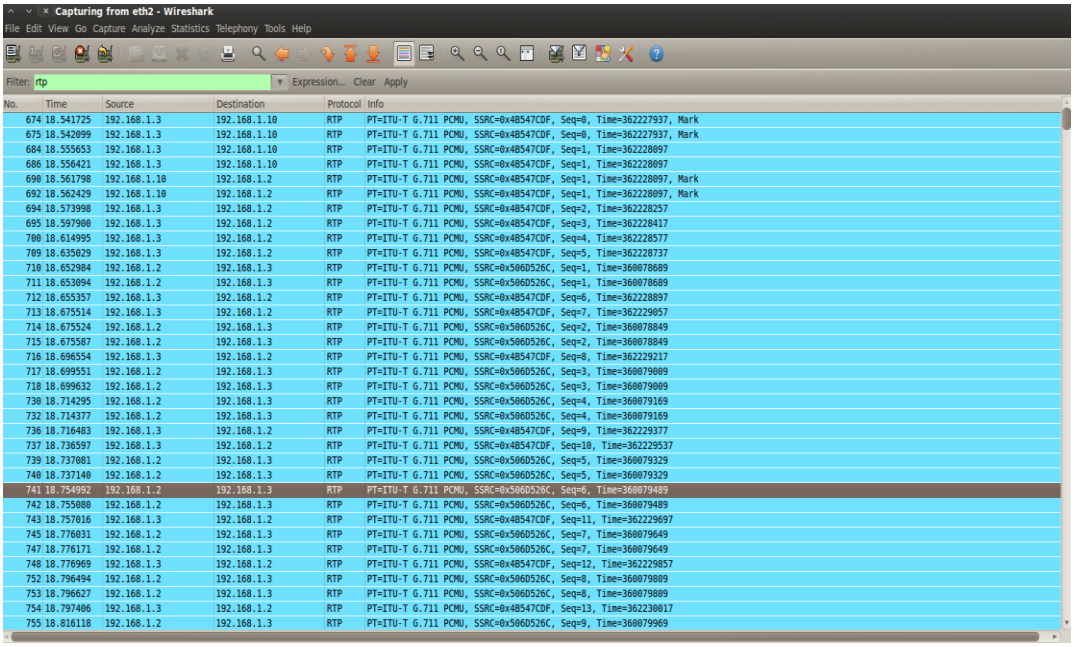
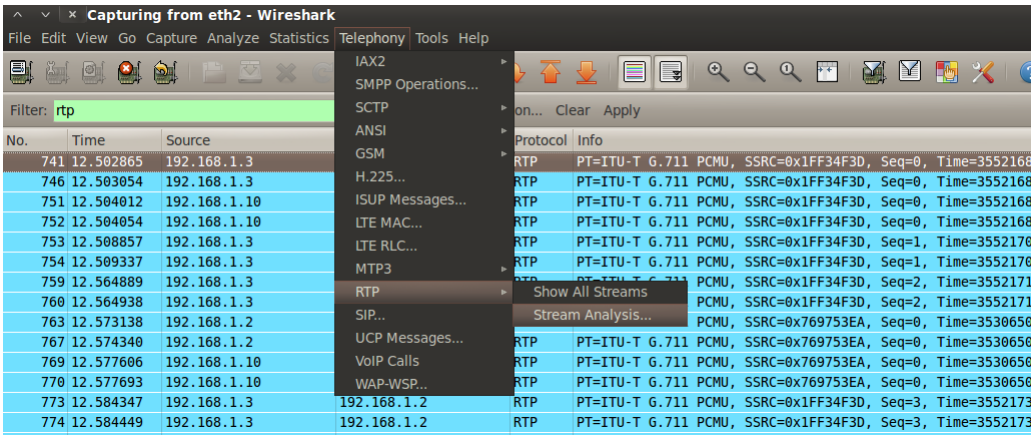
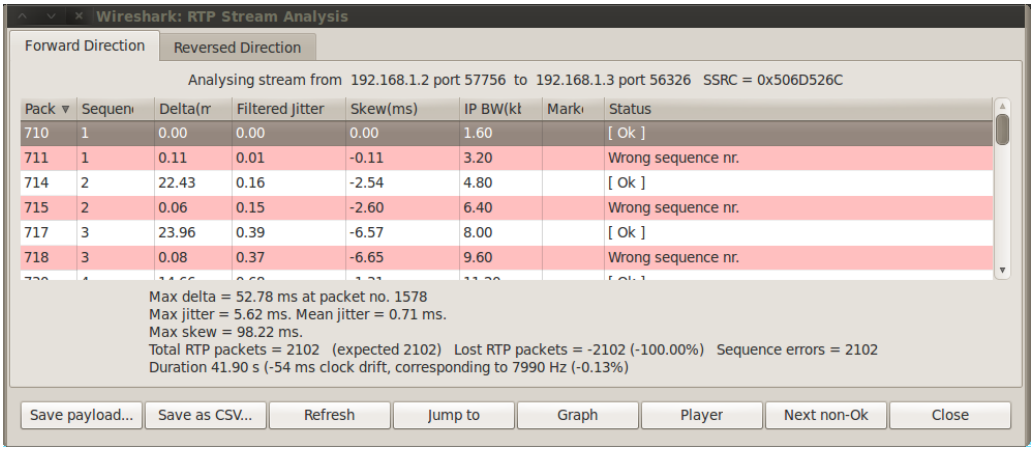


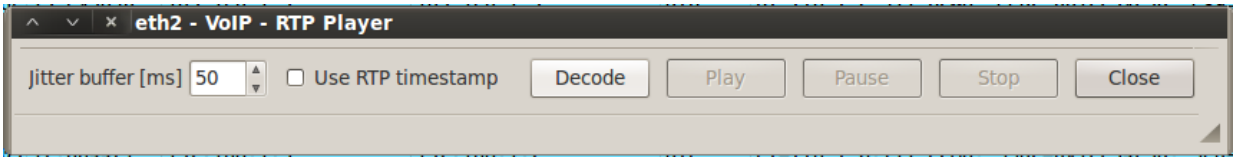
Figure IV.9 : L'analyse des paquets RTP récupérés.

Analyse des paquets RTP capturés :





On clique sur le bouton Player



Ensuite, on clique sur le bouton Decode

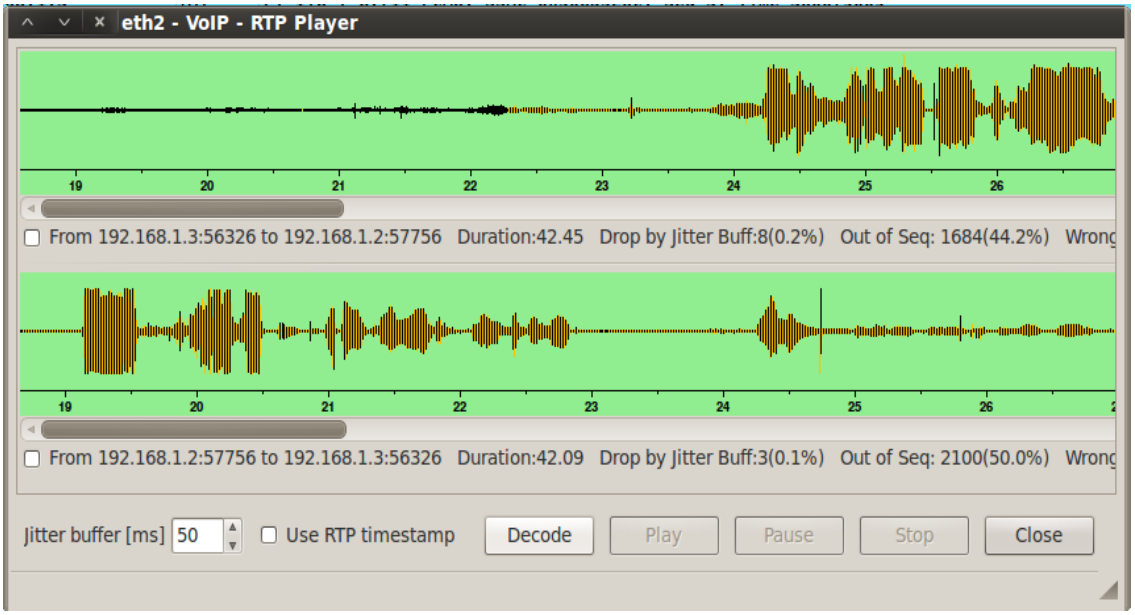


Figure IV.10 : Décodage du flux récupéré

Dans cette capture, on constate qu'une conversation entre les clients 6004 et 6003 a été décodée. Pour l'écouter, il suffit de cliquer sur le bouton Player.

- **Mécanisme de sécurité (Chiffrement des Appels)**

Pour sécuriser les appels, il faut chiffrer deux choses :

- Le trafic SIP
- Le trafic RTP

- **Chiffrement du trafic SIP avec TLS**

Pour cela, nous allons faire appel au protocole **TLS – Transport Layer Security**.

Générer les clés :

Créer un dossier qui contiendra les clés : **mkdir /etc/Asterisk/keys**

Le script permettant de créer les clés se trouve dans le dossier suivant :

root@karima-HP-620:~# cd /usr/src/Asterisk/Asterisk-11.10.2/contrib/scripts/

Le script s'exécute comme ceci :

root@karima-HP-620:/usr/src/Asterisk/Asterisk-11.10.2/contrib/scripts#./ast_tls_cert-C

192.168.1.10 -O "Master_RSD" -d /etc/Asterisk/keys/

Voici le détail des options :

- **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, il est demandé de spécifier le mot de passe des fichiers.

Les fichiers suivants devraient être créés :

```
root@karima-HP-620:/etc/asterisk/keys# ls
asterisk.crt  asterisk.csr  asterisk.key  asterisk.pem  ca.cfg  ca.crt  ca.key  tmp.cfg
```

Nous avons donc un certificat self-signed pour l'autorité de certificat, et un certificat pour le serveur Asterisk.

Nous avons aussi une clé privée pour l'autorité de certificat et une pour Asterisk.

Les fichiers PEM regroupent la clé privée et le certificat.

Les fichiers CSR sont des fichiers de requête de certificat (nous n'en avons pas besoin).

En suite, créer les clés et les certificats pour les clients.

root@karima-HP-620:/usr/src/Asterisk/Asterisk-11.10.2/contrib/scripts# ./ast_tls_cert -m client -c /etc/Asterisk/keys/ca.crt -k /etc/Asterisk/keys/ca.key -C 192.168.1.10 -O "Master_RSD" -d /etc/Asterisk/keys/ -o 6001

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.

A présent, nous devons configurer Asterisk pour autoriser l'utilisation de TLS.

Dans le fichier **sip.conf**, apporter les modifications suivantes :

```
[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.

```
[6001]
Type=friend
Host=dynamic
fullname= karima MECHERNENE
username = karima
md5secret=8ae80145b304007cb19732b6aafa517c
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
- client.pem

Client 6001 :

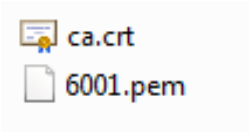


Figure IV.11 : Certificat et clé pour chiffrer la communication.

Ajouter l'adresse IP du serveur Asterisk, spécifier le port 5061 et choisit le type de transport TLS

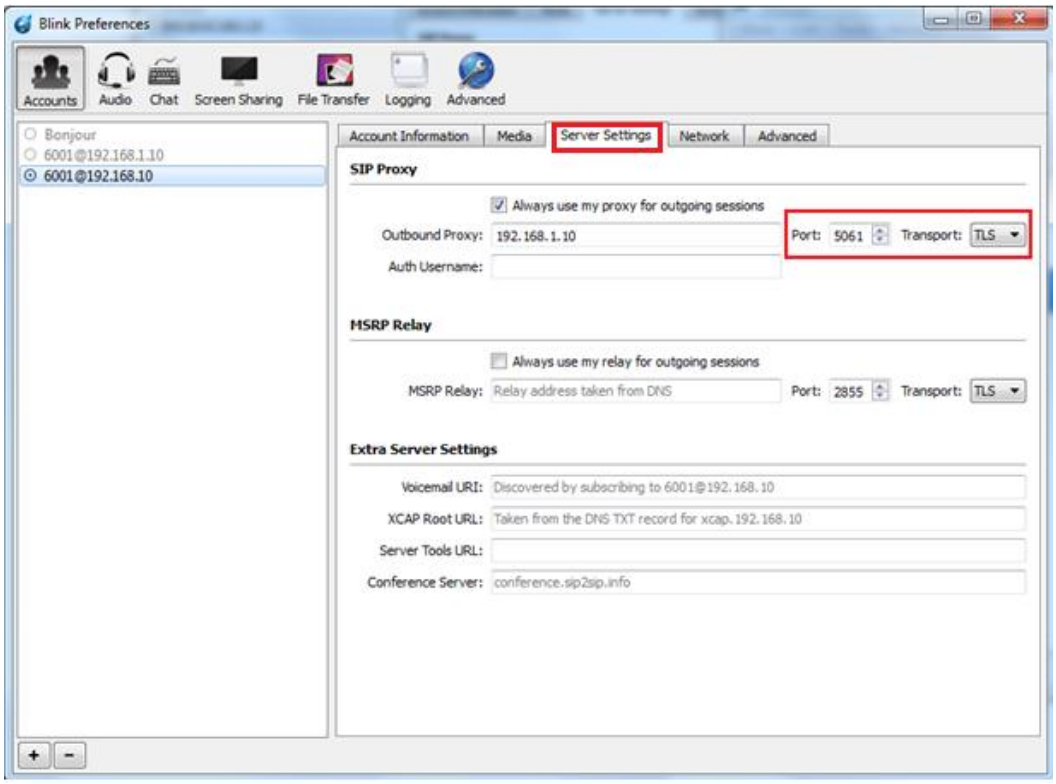
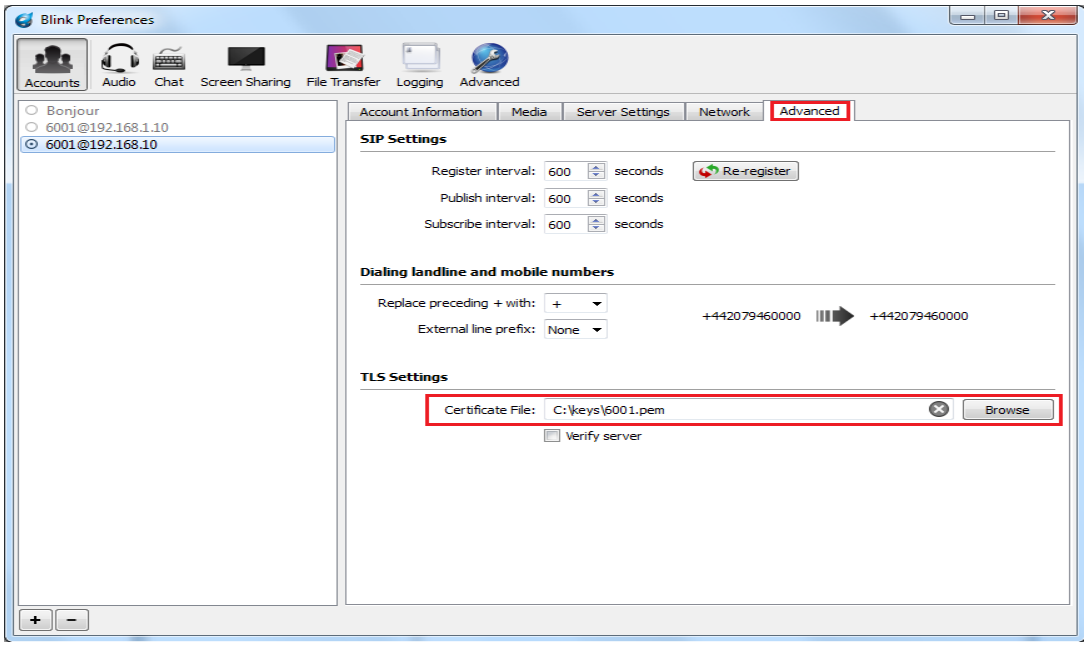
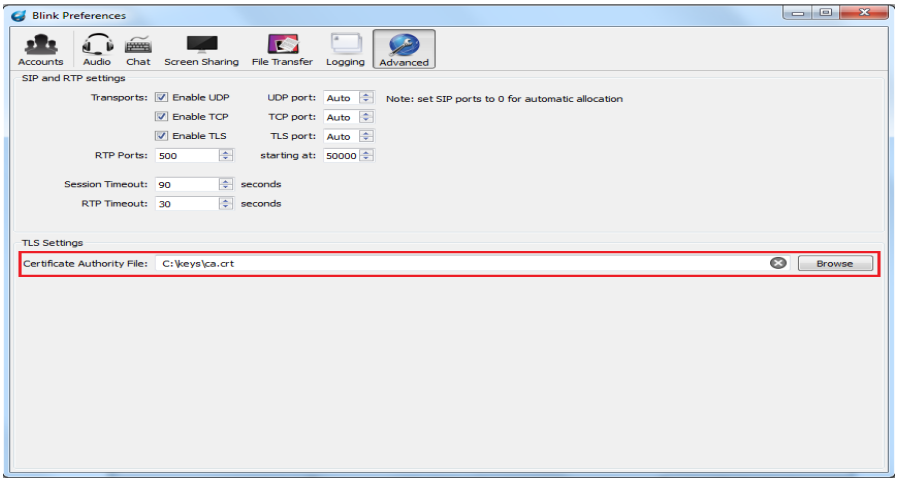


Figure IV.12 : Configuration du client avec le nouveau certificat et clé de chiffrement

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





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

```
karima-HP-620*CLI> sip reload
 Reloading SIP
   == Parsing '/etc/asterisk/sip.conf': Found
   == Parsing '/etc/asterisk/users.conf': Found
   == Using SIP CoS mark 4
SSL certificate ok
   == Parsing '/etc/asterisk/sip_notify.conf': Found
karima-HP-620*CLI>
```

Pour assurer le chiffrement de la signalisation de l'appel un cadenas bleu apparaitre dans le softphone Blink (**Figure IV.13**) :

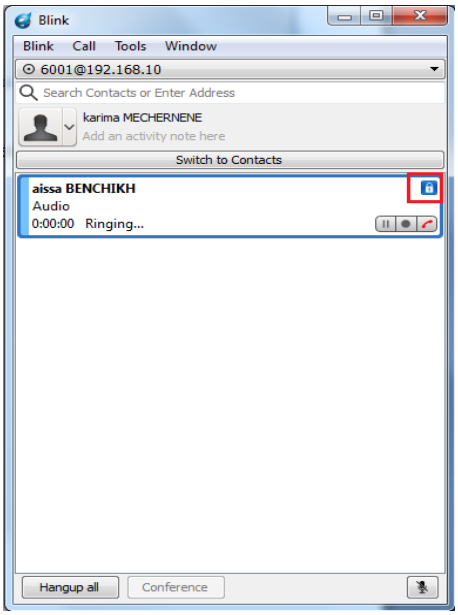


Figure IV.13 : 1ère étape pour chiffrer une Appel.

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

```
root@karima-HP-620:/etc/asterisk# netstat
Connexions Internet actives (sans serveurs)
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat
tcp        0      0 karima-HP-620:sip-tls   192.168.1.2:52127       ESTABLISHED
tcp        0      0 karima-HP-620:sip-tls   192.168.1.3:53875       ESTABLISHED
```

- **Chiffrement du trafic RTP avec SRTP**

Pour cela, nous allons utiliser le protocole SRTP.

Les premières étapes consisteront à télécharger et installer les bibliothèques SRTP

root@karima-HP-620:/# cd /usr/src/

root@karima-HP-620:/usr/src# wget http://srtp.sourceforge.net/srtp-1.4.2.tgz

root@karima-HP-620:/usr/src#tar xfzv srtp-1.4.2.tgz

root@karima-HP-620:/usr/src#cd srtp

root@karima-HP-620:/usr/src/srtp #./configure --prefix=/usr

Compiler et installer

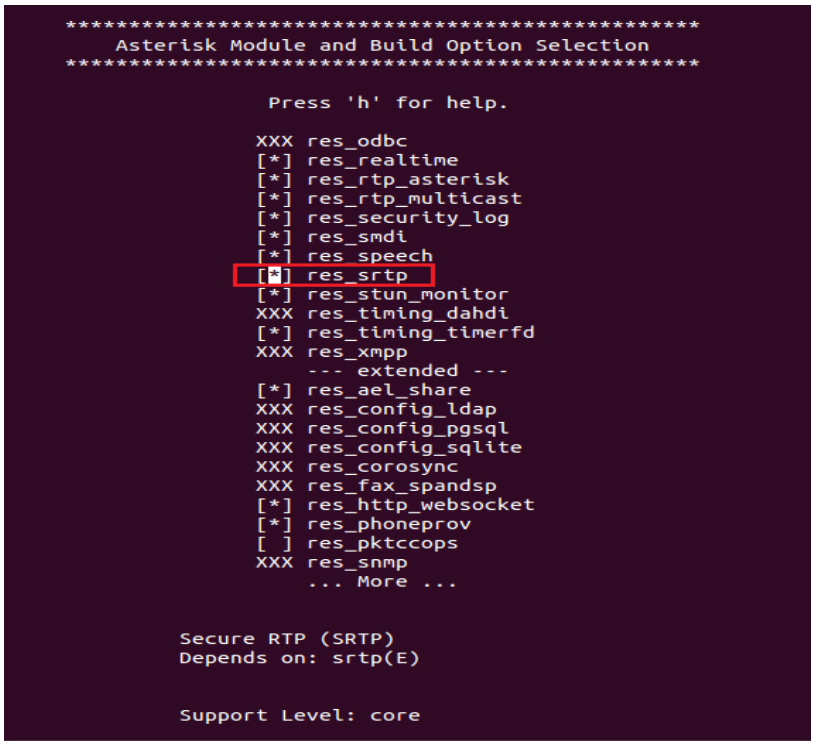
root@karima-HP-620:/usr/src/srtp# make

root@karima-HP-620:/usr/src/srtp# make install

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

root@karima-HP-620:/usr/src/Asterisk/Asterisk-11.10.2# make menuselect

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@karima-HP-620:/usr/src/Asterisk/Asterisk-11.10.2#**make clean**

root@karima-HP-620:/usr/src/Asterisk/Asterisk-11.10.2# **./configure**

root@karima-HP-620:/usr/src/Asterisk/Asterisk-11.10.2# **make**

root@karima-HP-620:/usr/src/Asterisk/Asterisk-11.10.2# **make install**

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
fullname= karima MECHERNENE
username = karima
md5secret=8ae80145b304007cb19732b6aafa517c
context = work
transport=tls
encryption=yes
```

En fin on configure softophone Blink de poste client pour qu'il utilise SRTP (**Figure IV.14**)

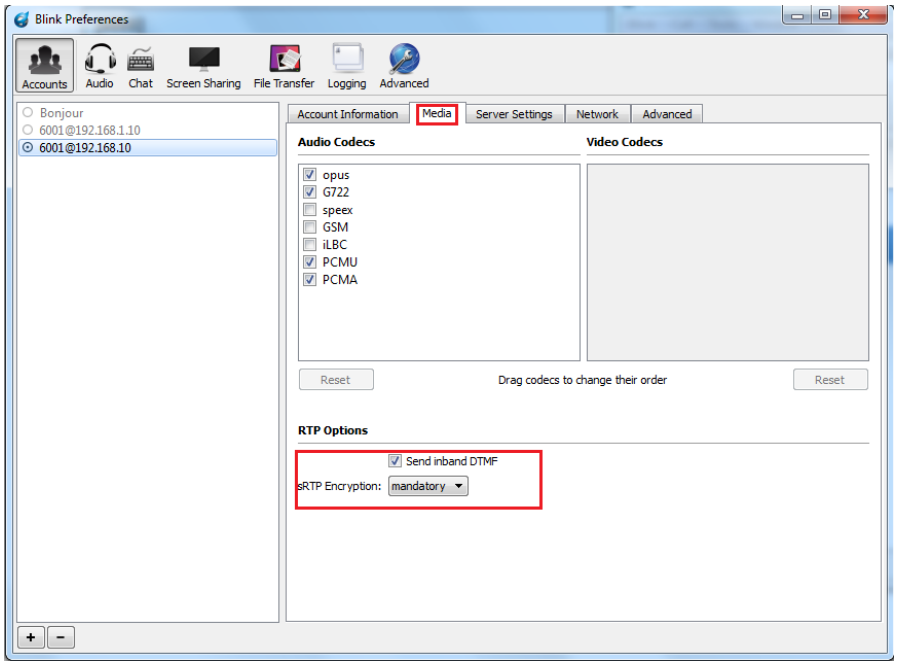


Figure IV.14 : Sécurisation du flux RTP.

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

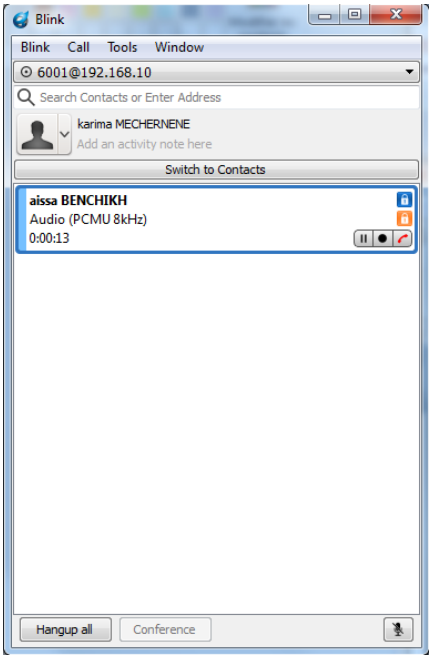


Figure IV.15 : Appel sécurisé.

Refaire l'attaque vu auparavant de l'écoute et analyse des flux RTP.

On remarque que la forme d'audio est déformé, en plus lorsqu'on clique sur le bouton play on écoute juste de bruit (**Figure IV.16)**.

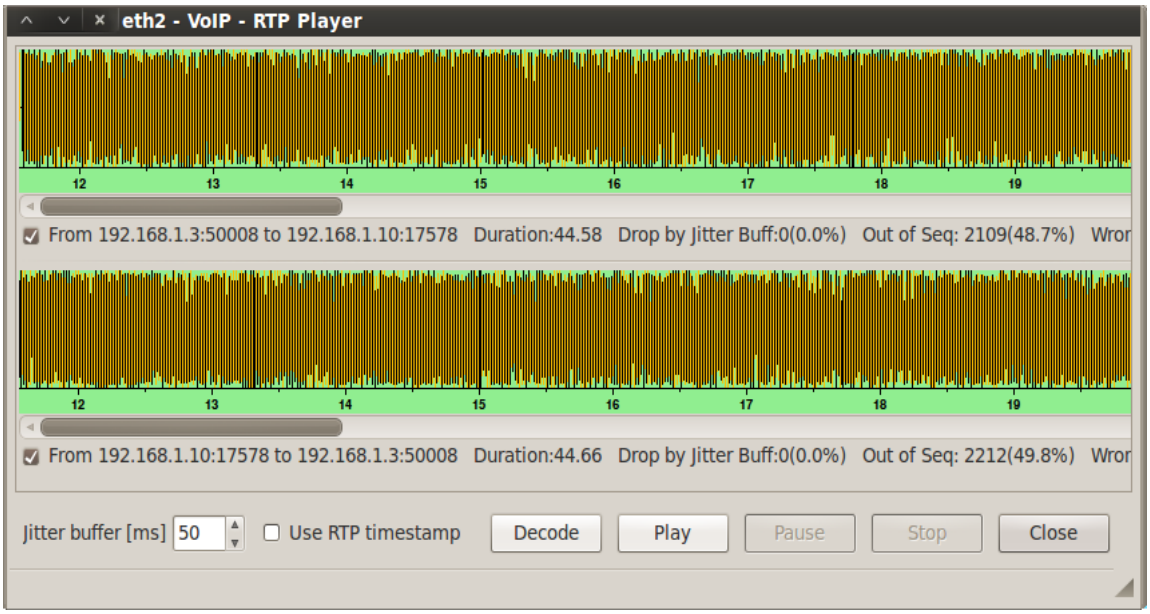


Figure IV.16 : Appel sécurisé et écoute parasitée

E. Attaque Denis de service (Dos)

- **Procédure**

  Dans cette attaque, en envoyant une multitude de requêtes SIP flooding jusqu'à saturation des services.

- Pré-requis

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.

Une syntaxe de l'utilisation est :

./inviteflood eth0 target_extension target_domain target_ip number_of_packets

Inviteflood est disponible dans le répertoire: **root @ bt:~# cd / pentest / VoIP /**

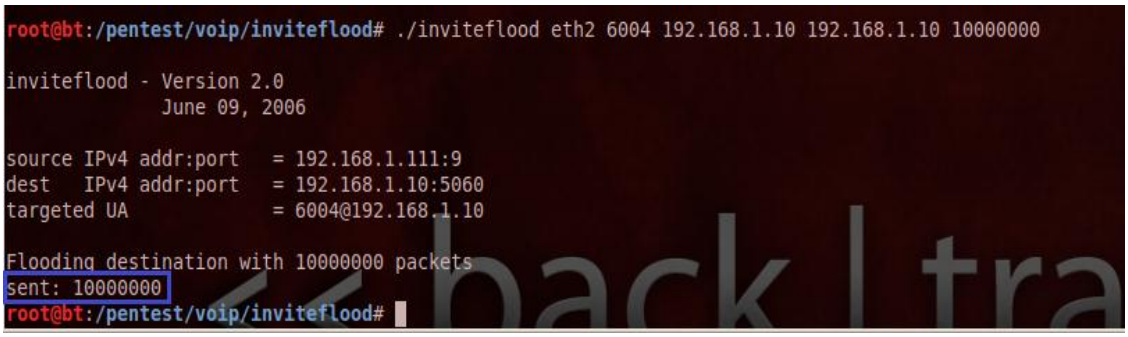
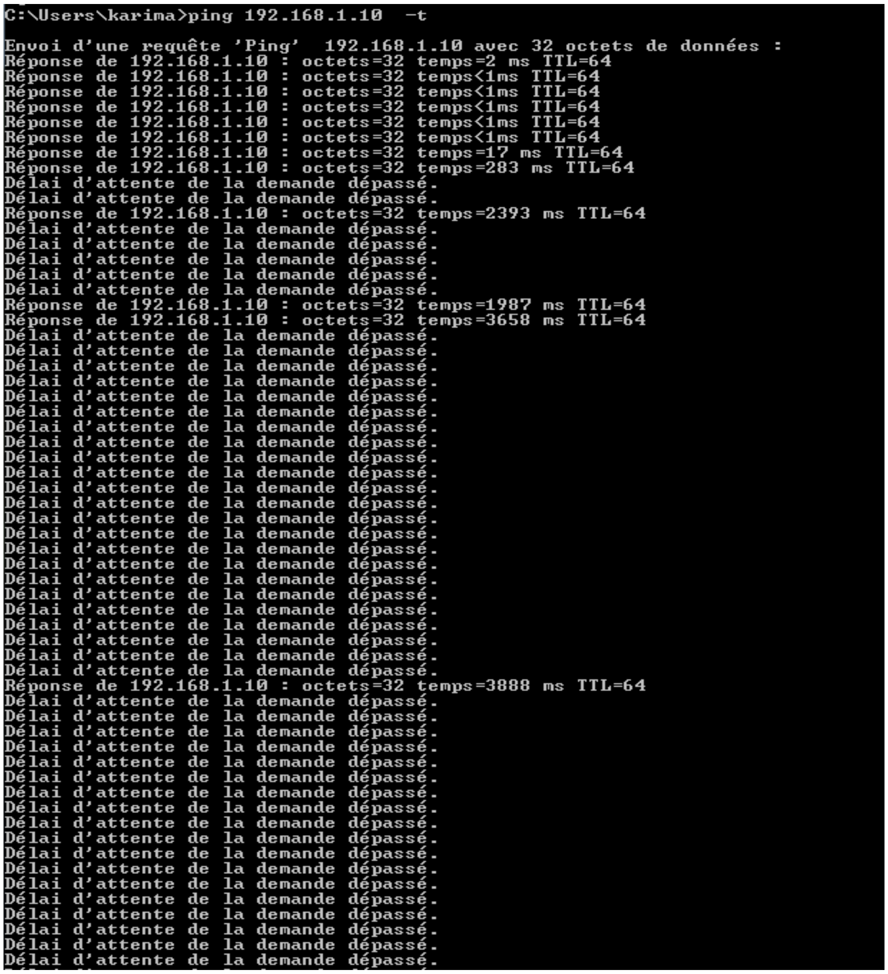


Figure IV.17 : Attaque DoS.

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.



On peut aussi de suivre la traçabilité de l'attaque au niveau de fichier log d'Asterisk :

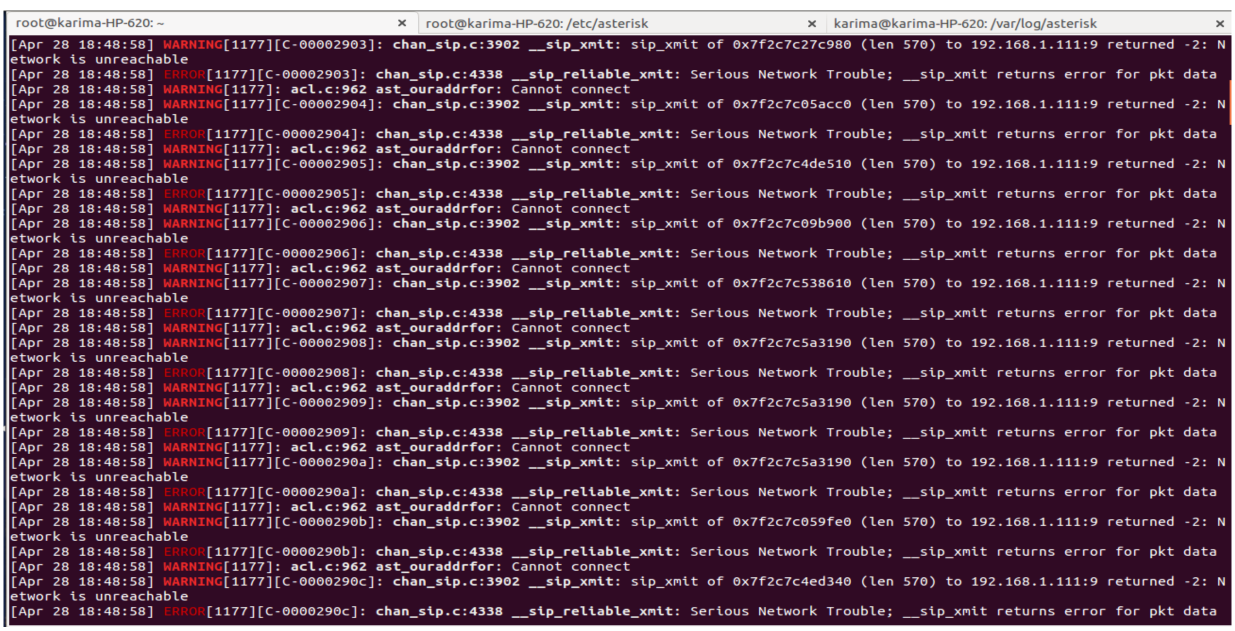


Figure IV.18 : Fichier log de l'attaque DoS

- **Autre aspect de Sécurité**
    - **Surveiller son système en lisant les logs**

Les journaux d'activité d'Asterisk (ou fichiers de log) sont classés dans le répertoire /var/log/Asterisk.

On y retrouve des informations importantes, comme une erreur survenue dans le programme ou dans un de ses modules, les connexions qui arrivent à notre serveur Asterisk. Cela permet aussi de voir si quelqu'un essaye de nous attaquer.

Ces fichiers sont en fait des indicateurs de santé du serveur Asterisk.

- **Configuration**

Le fichier le plus important pour l'administration et la surveillance du serveur Asterisk est /var/log/Asterisk/messages.

La configuration des journaux d'activité s'effectue dans le fichier /etc/Asterisk/logger.conf.

Configuration du journal d'activité (fichier /etc/Asterisk/logger.conf)

[general]]

[logfiles]

console => notice,warning,error,debug,verbose,dtmf

messages => notice,warning,error

- **Archivage**

Pour éviter des fichiers de logs trop volumineux pour Asterisk, il est judicieux de mettre en place une rotation de ces fichiers. Pour l'archivage du journal d'activité logrotate est utilisé. Ce programme vient en général s'installer avec le système d'exploitation

On trouvera donc un fichier /etc/logrotate.d/Asterisk destiné à archiver le journal d'activité. Archivage du journal d'activité (fichier /etc/logrotate.d/Asterisk)

compress
/var/log/Asterisk/messages {
daily
missingok
rotate 30
sharedscripts
postrotate
/usr/sbin/rAsterisk -x 'logger reload' >/dev/null 2>/dev/null || true
endscript }

Quelques explications sur les instructions de ce fichier s'imposent.

Les archives sont compressées par le programme gzip et suffixées par .gz.

Avec le journal d'activité, le fichier messages existant dans la repertoir /var/log/Asterisk est archivé. Les archives sont réalisées quotidiennement et trente jours d'enregistrement d'activité sont conservés.

Une fois les archives créées, c'est-à-dire après l'exécution périodique du programme logrotate, la commande console logger reload est exécutée (via le programme rAsterisk) afin de créer un nouveau fichier /var/log/Asterisk/message. L'instruction sharedscripts permet de n'exécuter cette commande qu'une fois, plutôt que pour chaque archive enregistrée.

On peut ensuite tester la rotation des logs en exécutant logrotate manuellement:

```
root@karima-HP-620:/# cd /var/log/asterisk/
root@karima-HP-620:/var/log/asterisk# ls
cdr-csv  cdr-custom  cel-custom  messages
root@karima-HP-620:/var/log/asterisk# logrotate -vf /etc/logrotate.d/asterisk
```

Cette figure montre l'archive de fichier messages

```
root@karima-HP-620:/var/log/asterisk# ls
cdr-csv  cdr-custom  cel-custom  messages  messages.1.gz  queue_log
```

- **Solution CDR (CALL Detail Record) dans Asterisk**

CDR (*Call Detail Records*) sont indispensables à tout système de téléphonie d'entreprise, principalement pour des raisons de sécurité. Ils contiennent en effet les informations sur les appels traités par Asterisk (numéro appelant, destination, durée de l'appel, etc.).

Avec l'authentification et le contrôle d'accès, la traçabilité est le troisième pilier qui soutient la politique de sécurité du système d'information dans l'entreprise.

Pour le système de téléphonie, les tickets de taxation sont les éléments de base qui permettront, en cas de besoin, de vérifier l'utilisation du système, et donc d'assurer la traçabilité des communications téléphoniques.

Asterisk dispose de plusieurs interfaces d'enregistrement de tickets de taxation, auxquelles accèdent différents modules, présentés dans le tableau ci-après.

Module	Format	Fichier de Configuration	Bibliothèque externe
cdr_csv.so	CSV (Comma Separated Values) : chaque ticket est enregistré dans une ligne d'un fichier texte dont les champs sont séparés par des virgules.	/etc/Asterisk/ cdr.conf : attention, la section [csv] doit contenir au moins une ligne non commentée, sinon Asterisk ne charge pas le module.	Non
cdr_custom.so	CSV : l'administrateur choisit les champs qui seront enregistrés.	/etc/Asterisk/ cdr_custom.conf	Non
cdr_radius.so	RADIUS : les tickets sont envoyés à un serveur RADIUS.	/etc/Asterisk/ cdr.conf : la section [radius] contient le fichier de configuration définissant la connexion RADIUS.	radiusclient-ng

Tableau IV.19: Interfaces d'enregistrement des CDR

On a choisi d'utiliser l'interface d'enregistrement basique pour les CDR.

La configuration de base du CDR est déjas présente dans le fichier cdr.conf

Le module cdr_csv sera chargé au démarrage d'Asterisk,sinon il est possible de le charger comme ceci : Asterisk –rvvvvvvv

Module load cdr-csv.so

Pour confirmer que le module apparaît bien via la commande console **cdr show status**.

```
karima-HP-620*CLI> cdr show status

Call Detail Record (CDR) settings
----------------------------------
  Logging:                      Enabled
  Mode:                         Simple
  Log unanswered calls:         No
  Log congestion:               No

* Registered Backends
  --------------------
    cdr-custom
    csv
```

Les CDR seront enregistrés dans le répertoire **/var/log/Asterisk/cdr-csv/Master.csv.**

La figure suivante montre le format général du fichier Master.csv

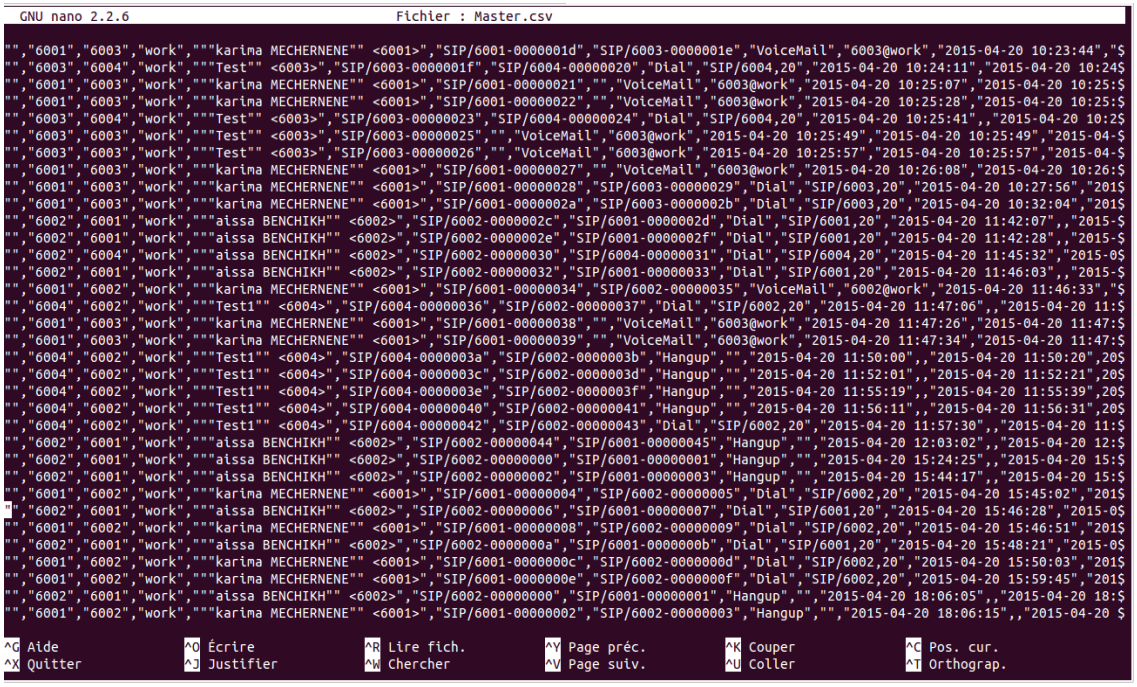


Figure IV.20 : Format du fichier Master.csv du CDR

Parmi les colonnes, nous retrouvons :

- **Clid** : l'ID de l'appelant
- **Start** : date de début de l'appel
- **Answer** : date de la réponse
- **End** : date de fin de l'appel
- **Duration** : durée total de l'appel

- **Billsec** : durée de l'appel après décrochage

- **Contournement 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.

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

deny=<ip addresse>/<network mask>

Par exemple :

```
[6001]
Type=friend
Host=dynamic
fullname= karima MECHERNENE
username = karima
md5secret=8ae80145b304007cb19732b6aafa517c
context = work
deny=0.0.0.0/0.0.0.0
permit=192.168.1.2/255.255.255.255
```

Refuser toutes les adresses IP sauf pour l'adresse IP 192.168.1 est autorisé.

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

```
[2015-04-30 21:05:42] NOTICE[1092]: acl.c:748 ast_apply_acl: SIP Peer ACL: Rejecting '192.168.1.5' due to a failure to pass ACL '(BASELINE)'
[2015-04-30 21:05:42] NOTICE[1092]: chan_sip.c:28172 handle_request_register: Registration from '"Test1"<sip:6004@192.168.1.10>' failed for '19
2.168.1.5:7044' - Device does not match ACL
```

Différenciez vos noms d'utilisateurs de vos extensions SIP.

- Il est conseillé de choisir un nom d'utilisateur SIP différent de l'extension.

3. Conclusion

Avec la complexité des protocoles et l'absence de sécurité fiable sous la plate-forme VoIP, on a simulé des attaques et des tests afin de découvrir les failles et les vulnérabilités de ce système de communication, et pour but corriger ces erreurs et réaliser une communication moderne, fiable et accessible.

❖ Conclusion générale

L'important de notre travail est consisté à étendre notre solution de gestion de risques à des infrastructures **VoIP**, notamment celles s'appuyant sur le protocole **SIP**.
La gestion des risques est un processus qui permet d'identifier, d'évaluer et de traiter les risques notamment dans les réseaux et les services. L'objectif principal de ce projet consiste à protéger les services et leur capacité à réaliser leur mission dans les infrastructures **VoIP** qui offrent des services interactifs et variété de stratégies de traitement comme l'évitement ou l'optimisation des risques qui permet d'adapter l'exposition du réseau et des services en fonction de la potentialité de la menace et des coûts induits.
Nous avons étudié dans ce cadre différent scénarios d'attaques, notamment :
La substitution : la fonction a pour objet d'obtenir à partir d'un autre poste les fonctionnalités de son propre poste ou celle d'un autre poste à partir de son propre poste.
La malversation classique : est l'usurpation d'identité réalisée par une personne depuis son poste, mais attribuée à la personne dont le poste a été substitué.
L'écoute de conversation : l'implémentation de la fonction permet « l'entrée en tiers » sans bip d'annonce sur le poste que l'on souhaite écouter quand celui-ci est en communication.
La malversation évidente : est l'utilisation malveillante de certaines fonctionnalités à des fins d'espionnage.

Et comme conclusion, notre but principal de ce travail est d'assurer une communication sécurisée sur **IP** (**VoIP**), et pour cela on a testé et simulé les différentes attaques afin de remédier ces failles de sécurité et proposer un système de communication sûr et robuste.

❖ Références

[1]: Piere LEDRU, Téléphonie sur IP, Edition Eni, 2011.

[2]: Philippe-ATTELIN & José-DORDOIGNE, Réseau informatique, Edition Eni, 2006.

[3]: Theodore-WALLINGFORD & James-GUERIN, Switching to VoIP, Edition O'Reilly Media, 2005.

[4]: Lawrence-HARTE & Robert-FLOOD, Introduction to Private Telephone Systems, Edition Althos Publishing, 2005.

[5]: Bernard-VIAL, TCP/IP pratique, Edition Micro Application, 2009.

[6]:Ahmed-MEDDAHI & Gilles-VANWORMHOUDT, Téléphonie SIP, Edition Hermes-science, 2012.

[7]: M. Handley & V. Jacobson, The Internet Society, Edition ISI/LBNL, April 1998

[8]: IAX: Inter-Astersik eXchange version 2, Internet Engineering Task Force, RFC 5456

[9]: Hamid Mahamat Abdoulaye, Audit d'un réseau VoIP et implémentation d'un client SIP sécurisé, Master, TDSI, 2013

[10]: Rebha Bouzaida, Étude et Mise en place d'une Solution VOIP Sécurisée, Master Professionnel, université de Tunis, 2011

[11]: Yannick YANI KALOMBA, Etude et mise au point d'un système de communication VOIP, Université protestante de Lubumbashi, Ingénieur en réseaux et télécoms, 2009

Site Web à consulté :

[12]: https://www.terena.org/activities/tf-vvc/voip-wsh/SIP-tutorial.pdf

[13]: http://docs.polycom.com/global/documents/whitepapers/codecs_white_paper.pdf

[14]: http://www.frameip.com/voip/

[15]: http://www.linuxgull.ch/rsrc/GULLServeurWWW/GestionActiviteDocumentCours

[16]: VoIPAsterisk20071002/asterisk_voip_slides.pdf

[17]: http://www2.cndp.fr/notestech/39/nt039_C.htm

[18]: http://wiki.ncad.fr/index.php?title=Asterisk_IAX

[19]: http://www.ficome.fr/campus/CR5_universite_ete/asterisk.pdf

[20]: http://read.pudn.com/downloads105/AsteriskWin32/Patrick_Deruel_EUR06.pdf

[21]: http://www.memoireonline.com/08/11/4644/Etude-et-mise-au-point-dun-systeme-de-communication-VOIP--application-sur-un-PABX-IP-open-source.html

[22]: http://www.nolot.eu/Download/Cours/reseaux/m2pro/SESY0708/securite_voip.pdf

[23]: http://docnum.univ-lorraine.fr/public/DDOC_T_2013_0044_DABBEBI.pdf

[24]: http://www.madpowah.org/textes/Man_In_The_Middle.pdf

[25]: http://www.adwfr.com/memo/Sage/PDF/TO_securite_VOIP_V5.0.pdf

[26]: http://repo.zenksecurity.com/Protocoles_reseaux_securisation/Securité.pdf

[27]: http://dept-info.labri.fr/~guermouc/SR/SR/cours/cours3.pdf

[28]: http://www.ssi.gouv.fr/uploads/SSL_TLS_etat_des_lieux_et_recommandations.pdf

[29]: http://horms.net/projects/ssl_and_tls/stuff/ssl_and_tls.pdf

[30]: http://www.idconline.com/Spec_Alert_and_Handshake_Protocol.pdf

[31]: http://igm.univ-mlv.fr/~duris/NTREZO/20022003/SSH_SSL_TLS.pdf

[32]: http://anonoups.unblog.fr/2012/10/27/comment-installer-backtrack-5/

[33]: http://www.giac.org/paper/gsec/3492/arp-cache-poisoning-ettercap/105198

[34]: https://www.ict.tuwien.ac.at/lva/384.081/datacom/A2_Standard_Applications_ss2012_v4-5.pdf

[35]: http://blog.sipvicious.org/2007/09/sipvicious-tools-in-works.html

[36]: http://cours.darties.fr/wp-content/uploads/sites/6/Fiche-Wireshark.pdf

[37]: ftp://ftp2.ephonenet.com/ephonenetj/public/eequip/fixe/xlite4/ephonenet/xlite4_configuration.pdf

[38]: http://iso.framadvd.org/standard/content/Data/Documents/pdf/fiche4-gnulinux.pdf

[39]: https://wiki.noojee.com.au/@api/deki/pages/136/pdf

[40]:http://www.authsecu.com/ssl-tls/ssl-tls.php#Le_protocole_SSL_et_TLS

❖ Résumé

Le but de ce projet est de mettre en place une plate-forme de redirections d'appels **SIP (Session Initiation Protocol)**, d'assurer la sécurisation du trafic, de garantir une qualité de service.

La plate-forme doit recevoir des appels en **SIP** (SIP est un protocole utilisé en voix sur **IP** permettant de transférer de la voix, de la vidéo ou des données à travers un réseau.) et les rediriger vers les passerelles correspondantes. Le système choisit pour effectuer le routage des appels **SIP** est le système **Asterisk (est un autocommutateur téléphonique privé (PABX))**.

Etant donné que ce serveur reçoit du trafic venant d'internet, il est important de mettre en place une politique de sécurisation des flux à traiter, pour cela, il fut décidé de faire des tests et des simulations afin de découvrir ses failles et ses inconvénients et par la suite la proposition des solutions et des remèdes permettant de garantir une bonne qualité de service à nos utilisateurs et entreprises.

❖ Resume

The purpose of this project is to develop a platform **SIP (Session Initiation Protocol)** call redirections ensure secure traffic to guarantee service quality.

The platform must receive calls **SIP** (SIP is a protocol used in VoIP to transfer voice, video or data over a network.) And redirect them to the corresponding gateways. The system chooses to perform the routing of SIP calls is the **Asterisk** system. (**Asterisk** is a **private automatic branch exchange (PABX))**

Since this server receives traffic from the Internet, it is important to establish a flow security policy to deal with, for this; it was decided to make tests and simulations to discover its flaws and drawbacks and subsequently the proposal of solutions and remedies to ensure a good quality of service to our users and companies.

❖ ملخص

الغرض من هذا المشروع هو تطوير منصة **SIP (Session Initiation Protocol)** وإعادة توجيه النداء لضمان المرور الآمن و ضمان جودة الخدمة.

المنصة تستقبل المكالمات **SIP** (SIPهو بروتوكول مستخدم في الاتصالات يستعمل في الإنترنت لنقل الصوت والفيديو أو البيانات عبر الشبكة) و توجيهها إلى بوابات المقابلة . يختار **SIP** لأداء توجيه النداءات النظام النجمة (**Asterisk**) **private automatic branch exchange** (**تبادل فرع التلقائي الخاص**)

لأن هذا الخادم يتلقى حركة المرور من الإنترنت، فمن المهم وضع سياسة أمنية لتدفق التعامل معها، لهذا فقد تقرر إجراء اختبارات و عمليات المحاكاة لاكتشاف عيوبها و بعد ذلك اقتراح الحلول والمعالجات لضمان نوعية جيدة من الخدمات للمستخدمين والشركات.