République **A**lgérienne **D**émocratique et **P**opulaire
Ministère de L'**E**nseignement **S**upérieur et de la A **R**echerche **S**cientifique

UNIVERSITE **M**OULOUD **M**AMMERI DE **T**IZI-**O**UZOU

FACULTE DE **G**ENIE **E**LECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

Mémoire de Fin d'Etude de MASTER ACADEMIQUE

Domaine : **Sciences et Technologies**
Filière **: Génie électrique**
Spécialité : **Télécommunication et Réseaux**

Présenté par
Anis AMZIANE
Hakim AGDOUR

Thème

Mise en place et sécurisation d'une plateforme VoIP basée sur la solution open source Asterisk

Mémoire soutenu publiquement le 17 juillet 2016 devant le jury composé de :

M Mr Fethi OUALOUCHE
Maitre de conférences B, à l'UMMTO, Président

M Slimane HAMEG
Maitre assistant A, à l'UMMTO, Examinateur

M Mourad LAZRI
Maitre de conférences A, à l'UMMTO, Promoteur

Remerciements

Au terme de ce projet de fin d'études, nous adressons nos sincères remerciements à Monsieur LAZRI MOURAD, notre encadreur pour ses directives précieuses, et pour la qualité de son suivi durant le travail effectuer pour ce projet.

Nos remerciements s'adressent également à l'administration et aux professeurs de département d'électronique pour les moyens qu'ils ont mis à notre disposition afin d'élaborer ce travail.

Nous souhaitons exprimer nos gratitudes et nos vifs remerciements à nos familles pour leurs soutiens.

Nous voulons exprimer nos reconnaissances envers les amis et collègues qui nous ont apporté leurs support moral et intellectuel tout au long de notre démarche.

Pour finir, nous remercions les membres du jury qui ont accepté d'évaluer notre projet. Nous leurs présentons toute nos gratitudes et nos profonds respects.

Nous remercions ALLAH le tous puissant de nous avoir donné la force et le courage afin d'accomplir ce travail.

Je tien à exprimer mes chaleureuses dédicaces à mes chers parents que Dieu les bénissent
A mes très chers grands parents que je site chaleureusement.
A mes sœurs et frères
A toute ma famille paternelle et maternelle.

A mon binôme Hakim avec qui j'ai partagé ce modeste travaille

Anis

Nous remercions ALLAH le tous puissant de nous avoir donné la force et le courage afin d'accomplir ce travail.

Je tien à exprimer mes chaleureuses dédicaces à mes chers parents que Dieu les bénissent
A mes très chers grands parents que je site chaleureusement.
A mes sœurs et frères
A toute ma famille paternelle et maternelle.

A mon binôme Anis avec qui j'ai partagé ce modeste travaille

Hakim

Liste des figures

Figure 1.1 : Architecture générale de la voix sur IP

Figure 1.2 : étapes de la transmission de VOIP

Figure 1.3 : Les composants de l'architecture H.323

Figure 1. 4 : La zone H.323

Figure 1.5 :Enregistrementd'UNutilisateur

Figure 1.6 : Principe du protocole SIP

Figure 1.7 : Session SIP à travers un proxy

Figure 2.1 : Architecture du réseau VoIP à réaliser

Figure 2.2 : configuration d'Asterisk réussie

Figure 2.3 : installation d'asterisk réussie

Figure 2.4 : Démarrage d'asterisk

Figure 2.5 : accès au mode console

Figure 2.6 : création des comptes utilisateurs

Figure 2.7 : Configuration du compte du client « 6001 »

Figure 2.8 : appel de l'utilisateur 6001 vers 6002

Figure 2.9 : appel de l'utilisateur 6002 vers 6003

Figure 3.1 : Exemple d'une écoute clandestine " Man in the middle"

Figure 3.2 : Attaque DoS via une requête CANCEL

Figure 3.3 : processus d'authentification SIP entre client/server

Figure 3.4 : paramètre du paquet « challenge »

Figure 3.5: Exemple d'un hachage MD5

Figure 3.6 : test de joignabilité du serveur SIP

Figure 3.7 : scan de ports sur le serveur SIP

Figure 3.8 : scan de ports sur les postes clients

Figure 3.9 : outils contenue dans le dossier VoIP

Figure 3.10 : détection des périphériques SIP activé

Figure 3.11 : récolte d'information sur le serveur avec SMAP

Figure 3.12 : contenue de la suite SIPVicious et usage de svwar

Figure 3.13 : Enumération des utilisateurs

Figure 3.14 : exécution de l'ARP spoofing

Figure 3.15 : Table Mac avant ARP spoof

Figure 3.16: Table Mac après ARP spoof

Figure 3.17: lancement de Wireshark

Figure 3.18 : Ecran de capture Wireshark

Figure 3.19 : Exemple de paquet qui contient une requête INVITE

Figure 3.20: Capture d'une communication téléphonique

Figure 3.21: Décodage : Bouton VoIP Calls

Figure 3.22 : communication téléphonique détectés

Figure 3.23 : Fenêtre RTP Player

Figure 3.24 : Communication téléphonique décodé

Figure 3.25 : UDP flood en utilisant Hping3

Figure 3.26 : Exécution de l'Invite Flood

Figure 3.27 : message d'erreur, échec de l'appel

Figure 3.28 : capture de la réponse d'authentification en utilisant SIPDUMP

Figure 3.29 : capture de l'authentification SIP digest

Figure 3.30 : exécution de sipcrack

Figure 3.31 : crack de l'authentification SIP haché

Figure 3.32 : attaque brute force en utilisant SVCRAC

Figure 3.33 : attaque par brute force réussie

Figure 4.1 : emplacement de snort

Figure 4.2 : installation de SNORT réussie

Figure 4.3 : lancement de SNORT en mode NIDS

Figure 4.4 : alertes générées par SNORT

Figure 4.5 : configuration d'arpwatch

Figure 4.6 : redémarrage d'arpwatch

Figure 4.7 : Lancement d'arpwatch

Figure 4.8 : message signalant un changement d'adresse mac

Figure 4.9: envoie de la clé publique au client

Figure 4.10 : Format d'un paquet SRTP

Figure 4.11: exécution du script et création des clés

Figure 4.12: création des clés et des certificats

Figure 4.13 : création des clés et des certificats pour les clients

Figure 4.14: configuration de Blink

Figure 4.15 : enregistrement du certificat .pem

Figure 4.16: enregistrement du fichier ca.crt

Figure 4.17 : activation du TLS sur Blink

Figure 4.18: activation de TLS et de SRTP

Glossaire

ACE = Access Control Entry

ACL = Access Control List

AH = Authentification Header

ARP = Address Resolution Protocol

CAN = Convertisseur analogique numérique

CLI = Command Line Interface

DDoS = Distributed Denial of Service

DHCP = Dynamic Host Configuration Protocol

DMZ = Démilitarized Zone

DNS = Domain Name System

DoS = Deny of Service

DTMF = Dual-Tone Multi-Frequency

ESP = Encapsulated Security Payload

FTP = File Transfer Protocol

GSM = Global System for Mobile Communications

HTTP = HyperText Transfer Protocol

IAX = Inter-Asterisk eXchange

IAX = Inter-Asterisk Exchange

ICMP = Internet Control Message Protocol

IETF = Internet Engineering Task Force

IGMP = Internet Group Management Protocol

IGRP = Interior Gateway Routing Protocol

IM = Instant Message

IP = Internet Protocol

ISDN = Integrated Service Data Network

ITU = International Telecommunications Union

LAN = Local Area Network

MD5 = Message Digest 5

MIKEY = Multimedia Internet KEYing

MKI = Master Key identifier

NAT = Network Address Translation

OS = Operating System

PABX = Private Automatic Branch eXchange

PBX = Private Branch eXchange

PSTN = Public Switched Telephone Network

QoS = Quality of Service

RFC = Requests For Comment

RNIS = Réseau Numérique à Intégration de Service

RTC = Réseau Téléphonique de Commuté

RTCP = Real-time Transport Control Protocol

RTP = Real-Time Transport Protocol

RTSP = Real Time Streaming Protocol

SIP = Session Initiation Protocol

SNMP = Simple Network Management Protocol

SRTP = Secure Real-time Transport Protocol

TCP = Transport Control Protocol

TDM = Time Division Multiplexing

TFTP = Trivial File Transfert Protocol

TLS = Transport Layer Security

ToIP = Telephony over Internet Protocol

UAC = User Agent Client

UAS = User Agent Server

UDP = User Datagram Protocol

URL = Uniform Resource Locator

VoIP = Voice over Internet Protocol

WAN = World Area Network

ZRTP= Zimmermann Secure RTP

SOMMAIRE

INTRODUCTION..... *1*

CHAPITRE 1 : GENERALITES SUR LA VOIX SUR IP

1.1 Préambule 5

1.2 présentation de la voix sur IP 5

1.2.1 Définition 5

1.2.2 Architecture 6

1.2.3 Principe de fonctionnemet 7

1.3 Protocole H323 10

1.3.1 Description générale du protocole H323 10

1.3.2 Rôles des composants 11

1.3.3 Avantages et inconvénients de la technologie H323 14

1.4 Protocole SIP 15

1.4.1 Description générale du protocole SIP 15

1.4.2 Principe de fonctionnement 16

1.4.3 Rôles des composants 18

1.4.4 Avantages et inconvénients 20

1.5 Protocoles de transport 21

1.5.1 Le protocole RTP 21

1.5.1.1 Description générale de RTP 21

1.5.1.2 Les fonctions de RTP 21

1.5.1.3 Avantages et inconvénients 22

1.5.2 Le protocole RTCP 22

1.5.2.1 Description générale de RTCP 22

1.5.2.2 Point fort et limite Du protocole RTCP 24

1.6 Points forts et limite de la voix sur IP 24

1.7 Discussion 26

CHAPITRE 2 : INSTALLATION ET CONFIGURATION D'UNE SOLUTION VoIP BASEE SUR LE SERVEUR ASTERISK

2.1 Préambule 28

2.2 Présentation d'Asterisk 28

2.3 Architecture du réseau VoIP déployé 29

2.4 Installation d'Asterisk 11.22.0 29
2.4.1 Téléchargements des pré-requis 30
2.4.2 Téléchargement des codes sources 30
2.4.3 Extractions des paquetages 30
2.4.4 Compilation et installation 31
2.5 Configuration d'Asterisk 34
2.5.1 Identification des fichiers de configuration 35
2.5.2 Configuration des comptes 35
2.5.3 Configuration des extensions 37
2.6 Installation et configuration des softphone 39
2.6.1 Installation de ZOIPER 39
2.6.2 Configuration de ZOIPER 39
2.6.3 Simulation d'un appel avec ZOIPER 40
2.7 Discussion 43

CHAPITRE 3 : ATTAQUES, VULNERABILITES ET HACKING DANS LES RESEAUX DE VOIP

3.1 Préambule 45
3.2 Attaques sur la VoIP 46
3.2.1 Sniffing 47
3.2.2 L'ARP poisoning 48
3.2.3 L'ARP flooding 48
3.2.4 L'écoute clandestine 48
3.2.5 Suivie des appels 49
3.2.6 Injection de paquet RTP 50
3.2.7 Les Spam 50
3.2.8 Le déni de service (DOS : Denial Of Service) 51
3.2.9 Détournement d'appel (Call Hijacking) 54
3.2.10 Attaque visant l'authentification SIP 55
3.3 Les vulnérabilités de l'infrastructure 57
3.3.1 Faiblesses dans la configuration des dispositifs de la VoIP 57
3.3.2 Les téléphone IP 58
3.3.3 Les serveurs 59
3.3.4 Les vulnérabilités du système d'exploitation 59
3.4 Mise en œuvre des attaques contre le réseau VoIP déployé 60
3.4.1 Localisation des serveurs VoIP 60

3.4.2 Arp poisoning 66
3.4.3 Capture du trafic et écoute clandestine en utilisant Wireshark 68
3.4.3.1 captures de trames 69
3.4.3.2 démonstration de l'écoute clandestine aves Wireshark 71
3.4.4 Déni de services 74
3.4.4.1 Hping 74
3.4.4.2 Invite Flood 75
3.4.5 Attaque sur l'authentification SIP 76
3.4.5.1 Capture de l'authentification SIP en utilisant Sipdump 77
3.4.5.2 Crack de la réponse d'authentification SIP 78
3.4.5.3 Attaque par force brute 79
3.5 Discussion 81

CHAPITRE 4 : SECURISATION ET BONNES PRATIQUES

4.1 Préambule 83
4.2 Implémentation des bonnes pratiques de sécurisation 83
4.2.1 Exécuter Asterisk sous un utilisateur non privilégié 83
4.2.2 Implimentation d'un système de détection d'intrusion 85
4.2.2.1 Définition de SNORT 86
4.2.2.2 Emplacement de Snort 87
4.2.2.3 Installation des dépendances nécessaire 87
4.2.2.4 Configuration de SNORT en mode NIDS 89
4.2.2.5 Création des règles pour SNORT 91
4.3 Bonne pratique contre l'Arp poisoning 95
4.4 Bonne pratique contre l'écoute clandestine 99
4.4.1 TLS (Transport Layer Security 99
4.4.1.1 Principe de Fonctionnement 100
4.4.2 SRTP 101
4.4.2.1 Service de sécurités offertes par SRTP 101
4.4.2.2 Principe de fonctionnement de SRTP 102
4.4.2.3 Format du paquet SRTP 102
4.4.3 Chiffrement de la signalisation 103
4.4.4 Chiffrement du flux audio 109
4.5 Implémentation d'un firewall Netfilter 111
4.6 Discussion 114

CONCLUSION..... 115

BIBLIOGRAPHIE/ WEBOGRAPHIE

Introduction

La Voix sur IP constitue actuellement une des plus importantes évolutions dans le domaine des Télécommunications. Il y a quelques années, la transmission de la voix sur le réseau téléphonique classique ou RTC constituait l'exclusivité des télécommunications. Aujourd'hui, la donne a changé. La transmission de la voix via les réseaux IP constitue une nouvelle évolution majeure comparable à la précédente. Au-delà de la nouveauté technique, la possibilité de fusion des réseaux IP et téléphoniques entraîne non seulement une diminution de la logistique nécessaire à la gestion de deux réseaux, mais aussi une baisse importante des coûts de communication ainsi que la possibilité de mise en place de nouveaux services utilisant simultanément la voix et les données. Plus, récemment, Internet s'est étendu partiellement dans l'Intranet de chaque organisation, voyant ainsi le trafic total basé sur un transport réseau de paquets IP surpasser le trafic traditionnel du réseau voix (réseau à commutation de circuits).
La migration des entreprises vers ce genre de technologie n'est pas pour rien. Le but est principalement de : minimiser le coût des communications, utiliser le même réseau pour offrir des services de données, de voix, et d'images, et simplifier les coûts de configuration et d'assistance.

Plusieurs fournisseurs offrent certaines solutions qui permettent aux entreprises de migrer vers le monde IP. Des constructeurs de PABX tels que Nortel, Siemens, et Alcatel préfèrent la solution de l'intégration progressive de la VoIP en ajoutant des cartes extensions IP. Cette approche facilite l'adoption du téléphone IP surtout dans les grandes sociétés possédant une plateforme classique et voulant bénéficier de la voix sur IP. Mais elle ne permet pas de bénéficier de tous les services et la bonne intégration vers le monde des données.

Le développement des PABXs software, est la solution proposée par des fournisseurs tels que Cisco et Asterisk. Cette approche permet de bénéficier d'une grande flexibilité, d'une

très bonne intégration au monde des données et de voix, et surtout d'un prix beaucoup plus intéressant.

Cette solution, qui est totalement basée sur la technologie IP, est donc affectée par les vulnérabilités qui menacent la sécurité de ce protocole et l'infrastructure réseau sur laquelle elle est déployée.

Pour cela la sécurité du réseau VoIP n'est pas seulement une nécessité mais plutôt une obligation, avec laquelle on peut réduire, au maximum, le risque d'attaques sur les réseaux de VoIP.

La sécurité d'une solution de VoIP doit couvrir toute l'infrastructure réseau, incluant les outils et les équipements de gestion des communications et des utilisateurs, le système d'exploitation sur lesquels sont installés ces outils, et les protocoles de signalisation et de transport de données. Il faut même se protéger contre les personnes malveillantes. Mieux on sécurise, moins il y a de risques.

Ce travail a pour objectif : l'étude des protocoles de VoIP et des architectures proposées ,l'étude des vulnérabilités et des attaques de sécurités surs les divers composants d'une infrastructure VoIP dans des réseaux LAN ; et la mise en place d'une solution de VoIP sécurisée basée sur des outils open source, précisément le serveur Asterisk et le softphone Zoiper.Les entreprises, bénéficiant de notre solution, seront capables de mettre en place une plateforme de VoIP assez flexible, peu couteuse, et protégée contre les attaques de sécurité del'intérieur du réseau comme de l'extérieur aussi.

Ce mémoire se compose de quatre chapitres et une conclusion :
Le premier chapitre introduit la voix sur IP et ces éléments, décrit et explique son architecture et ces protocoles, et énumère les majeurs points forts de cette technologie ainsi que ses faiblesses.

Le deuxième chapitre s'intéresses à la mise en place d'une solution de VoIP pour les réseaux locaux basés sur le serveur Asterisk et le client Zoiper. Les différentes étapes

d'installation et de configuration ainsi que les pré-requis et les librairies nécessaires seront définies.

Le troisième chapitre s'intéresse aux vulnérabilités et aux attaques qui menacent la sécurité d'un réseau de VoIP, nous détaillerons quelques-unes. Nous finirons par une simulation d'un test de pénétration sur le réseau de VoIP déployée en s'appuyant sur les attaques qu'on aura définies.

Le dernier chapitre du rapport, s'intéresse aux solutions, mécanismes et configurations à mettre en place dans le but de sécuriser la solution VoIP basé sur le serveur Asterisk contre les différentes techniques de hacking que nous avons menés dans le chapitre 3.

Nous terminons notre travail par une conclusion tout en donnant les perspectives.

Chapitre 1

Généralités sur la voix sur IP

1.1. Préambule :

La voix sur IP constitue actuellement l'évolution la plus importante du domaine des Télécommunications. Ces dernières années, les réseaux de données ont augmenté à un rythme hallucinant, en grande partie en raison de l'évolution d'Internet. Selon certains experts, le trafic de données est prévu pour dépasser bientôt le trafic vocal traditionnel. En conséquence, de plus en plus d'entreprises se sont intéressés à la mise en œuvre de cette technologie qui va leur permettre de fusionner le trafic voix et le trafic de donné au sein d'un seul et même réseau.

L'objectif de ce chapitre est l'étude de cette technologie et de ses différents aspects. On parlera en détail de l'architecture de la VoIP, ses éléments et son principe de fonctionnement. On détaillera aussi des protocoles VoIP de signalisation et de transport ainsi que leurs principes de fonctionnement et de leurs principaux avantages et inconvénients.

1.2. Présentation de la voix sur IP

1.2.1. Définition [14] [19]

VoIP signifie Voice over Internet Protocol ou Voix sur IP. Comme son nom l'indique, la VoIP permet de transmettre des sons (en particulier la voix) dans des paquets IP sur des réseaux compatibles avec le protocole IP que ce soit des réseaux privé ou internet, filaire (câble, ADSL, fibre optique) ou non filaire (wifi, GSM, satellite). La VoIP comprend donc les communications de pc à pc dont lequel chaque utilisateur utilise un logiciel appelé softphone pour effectuer des appels, communications de pc à téléphone IP ou bien de téléphone IP à téléphone IP. La VoIP consiste à échanger deux types d'informations entre les terminaux communicants : la voix (les données utiles) et la signalisation.

1.2.2. Architecture

La VoIP étant une nouvelle technologie de communication, elle n'a pas encore de standard unique. En effet, chaque constructeur apporte ses normes et ses fonctionnalités à ses solutions.

Les trois principaux protocoles sont H.323, SIP et MGCP/MEGACO. Il existe donc plusieurs approches pour offrir des services de téléphonie et de visiophonie sur des réseaux IP.

Certaines placent l'intelligence dans le réseau alors que d'autres préfèrent une approche égale à égale avec l'intelligence répartie à la périphérie. Chacune ayant ses avantages et ses inconvénients.

La figure 1.1 décrit, de façon générale, la topologie d'un réseau de téléphonie IP. Elle comprend toujours des terminaux, un serveur de communication et une passerelle vers les autres réseaux.

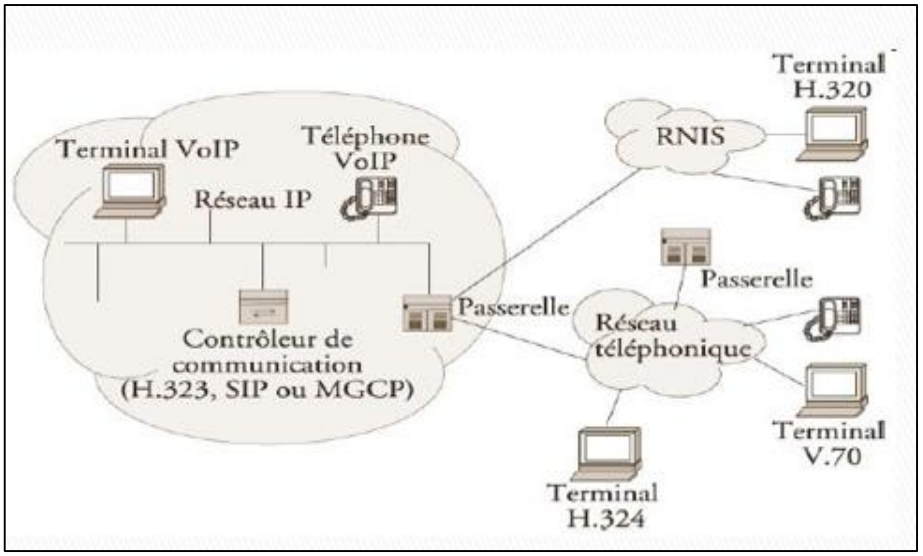

Figure 1.1 : Architecture générale de la voix sur IP. [19]

Chaque norme a ensuite ses propres caractéristiques pour garantir une plus ou moins grande qualité de service. L'intelligence du réseau est aussi déportée soit sur les terminaux, soit sur les passerelles/ contrôleur de commutation, appelées Gatekeeper. On retrouve les éléments communs suivants :

- Le routeur : permet d'aiguiller les données et le routage des paquets entre deux réseaux.

  Certains routeurs permettent de simuler un Gatekeeper grâce à l'ajout de cartes spécialisées supportant les protocoles VoIP.

- La passerelle : permet d'interfacer le réseau commuté et le réseau IP.
- Le PABX : est le commutateur du réseau téléphonique classique. Il permet de faire le lien entre la passerelle ou le routeur, et le réseau téléphonique commuté (RTC). Toutefois, si tout le réseau devient IP, ce matériel devient obsolète.

- Les Terminaux : sont généralement de type logiciel (software phone) ou matériel (Hard-Phone), le softphone est installé dans le PC de l'utilisateur. L'interface audio peut être un microphone et des haut-parleurs branchés sur la carte son, même si un casque est recommandé. Pour une meilleure clarté, un téléphone USB ou Bluetooth peut être utilisé.
- Le Hard-Phone est un téléphone IP qui utilise la technologie de la Voix sur IP pour permettre des appels téléphoniques sur un réseau IP tel que l'Internet au lieu de l'ordinaire système PSTN. Les appels peuvent parcourir par le réseau internet comme par un réseau privé. Un terminal utilise des protocoles comme le SIP (Session Initiation Protocol) ou l'un des protocoles propriétaire tel que celui utilisée par Skype.

1.2.3 Principe de fonctionnement

Le principe de la téléphonie sur IP est la numérisation de la voix, c'est-à-dire le passage d'un signal analogique à un signal numérique [11]. Celui-ci est compressé en fonction des codecs choisis, cette compression a comme but de réduire la quantité d'information qui est transmise sur le réseau. Le signal obtenu est découpé en paquets, à chaque paquet on ajoute les entêtes propres au réseau (IP, UDP, RTP…) et pour finir il est envoyé sur le réseau.

Le processus de la numérisation de la voix est schématisé par la figure suivante (voir figure 1.2) :

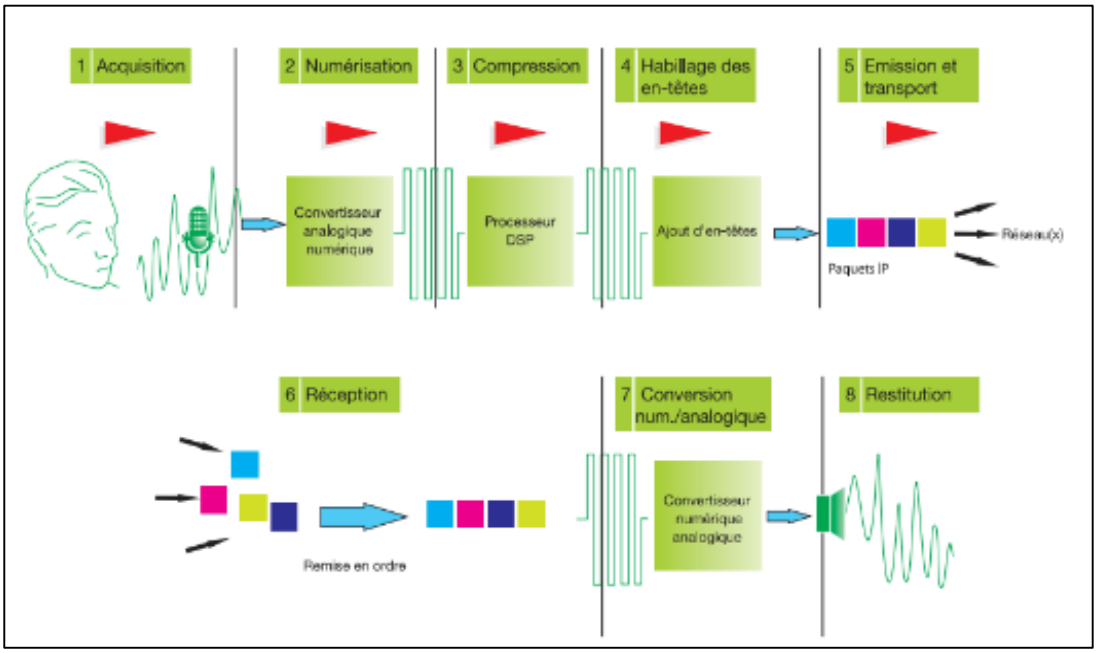

Figure 1. 2 : étapes de la transmission de VOIP. [11]

A. Acquisition du signal :

La VoIP suppose la transformation d'un signal continu analogique (la voix) en un signal discret numérique (composé d'une série de chiffres). La première étape consiste naturellement à capter la voix à l'aide d'un micro, qu'il s'agisse de celui d'un téléphone ou d'un micro casque.

B. Numérisation :

La voix passe alors dans un convertisseur analogique numérique qui réalise deux tâches distinctes :

- L'échantillonnage du signal sonore, c'est-à-dire un prélèvement périodique de ce signal.
- La quantification, qui consiste à affecter une valeur numérique (en binaire) à chaque échantillon. Plus les échantillons sont codés sur un nombre de bits important, meilleure sera la qualité (on parle de « résolution ») de la conversion. Généralement, la voix est échantillonnée à 8 kHz et chaque échantillon est codé sur 8 bits, ce qui donne un débit de 64 kbit/s (norme G711).

C. Compression :

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

D. Habillage des en-têtes :

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

- La couche IP :

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

- La couche UDP :

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

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

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

E. Emission et transport :

Les paquets sont acheminés depuis le point d'émission pour atteindre le point de réception sans qu'un chemin précis soit réservé pour leur transport. Ils vont transiter sur

le réseau (réseau local, réseau étendu voire Internet) en fonction des ressources disponibles et arriver à destination dans un ordre indéterminé.

F. Réception :

Lorsque les paquets arrivent à destination, il est essentiel de les replacer dans le bon ordre et assez rapidement. Faute de quoi une dégradation de la voix se fera sentir. Ce point sera détaillé plus loin.

G. Conversion numérique analogique :

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

H. Restitution :

Dès lors, la voix peut être retranscrite par le haut-parleur du casque, du combiné téléphonique ou de l'ordinateur. Les réseaux TCP/IP sont des supports de circulation de paquets IP contenant un en-tête (pour contrôler la communication) et une charge utile pour transporter les données.

Il existe plusieurs protocoles qui peuvent supporter la voix sur IP tel que le H.323, SIP et MGCP. Les deux protocoles les plus utilisées actuellement dans les solutions VoIP présentes sur le marché sont le H.323 et le SIP.

1.3. Protocole H.323

1.3.1. Description générale du protocole H.323

Le standard H.323 fournit, depuis son approbation en 1996, un cadre pour les communications audio, vidéo et de données sur les réseaux IP. Il a été développé par l'ITU (International Télécommunications Union) pour les réseaux qui ne garantissent pas une qualité de service (QoS), tels qu'IP IPX sur Ethernet, Fast Ethernet et Token Ring. Il est présent dans plus de 30 produits et il concerne le contrôle des appels, la gestion multimédia, la gestion de la bande passante pour les conférences point-à-point

et multipoints. H.323 traite également de l'interfaçage entre le LAN et les autres réseaux. Le protocole H.323 fait partie de la série H.32x qui traite de la vidéoconférence au travers différents réseaux. Il inclue H.320 et H.324 liés aux réseaux ISDN (Integrated Service Data Network) et PSTN (Public Switched Téléphone Network). Plus qu'un protocole, H.323 crée une association de plusieurs protocoles différents et qui peuvent être regroupés en trois catégories : la signalisation, la négociation de codec, et le transport de l'information.

- Les messages de signalisation sont ceux envoyés pour demander la mise en relation de deux clients, qui indique que la ligne est occupée ou que le téléphone sonne, etc. En H.323, la signalisation s'appuie sur le protocole RAS pour l'enregistrement et l'authentification, et le protocole Q.931 pour l'initialisation et le contrôle d'appel.
- La négociation est utilisée pour se mettre d'accord sur la façon de coder les informations à échanger. Il est important que les téléphones (ou systèmes) utilisent un langage commun s'ils veulent se comprendre. Il s'agit du codec le moins gourmand en bande passante ou de celui qui offre la meilleure qualité. Il serait aussi préférable d'avoir plusieurs alternatives de langages. Le protocole utilisé pour la négociation de codec est le H.245.
- Le transport de l'information s'appuie sur le protocole RTP qui transporte la voix, la vidéo ou les données numérisées par les codecs. Les messages RTCP peuvent être utilisés pour le contrôle de la qualité, ou la renégociation des codecs si, par exemple, la bande passante diminue.

Une communication H.323 se déroule en cinq phases : l'établissement d'appel, l'échange de capacité et réservation éventuelle de la bande passante à travers le protocole RSVP (Ressource reservation Protocol), l'établissement de la communication audio-visuelle, l'invocation éventuelle de services en phase d'appel (par exemple, transfert d'appel, changement de bande passante, etc.) et enfin la libération de l'appel.

1.3.2 Rôle des composants

L'infrastructure H.323 repose sur quatre composants principaux : les terminaux, les Gateways, les Gatekeepers, et les MCU (Multipoint Control Units),(voir figure 1.3).

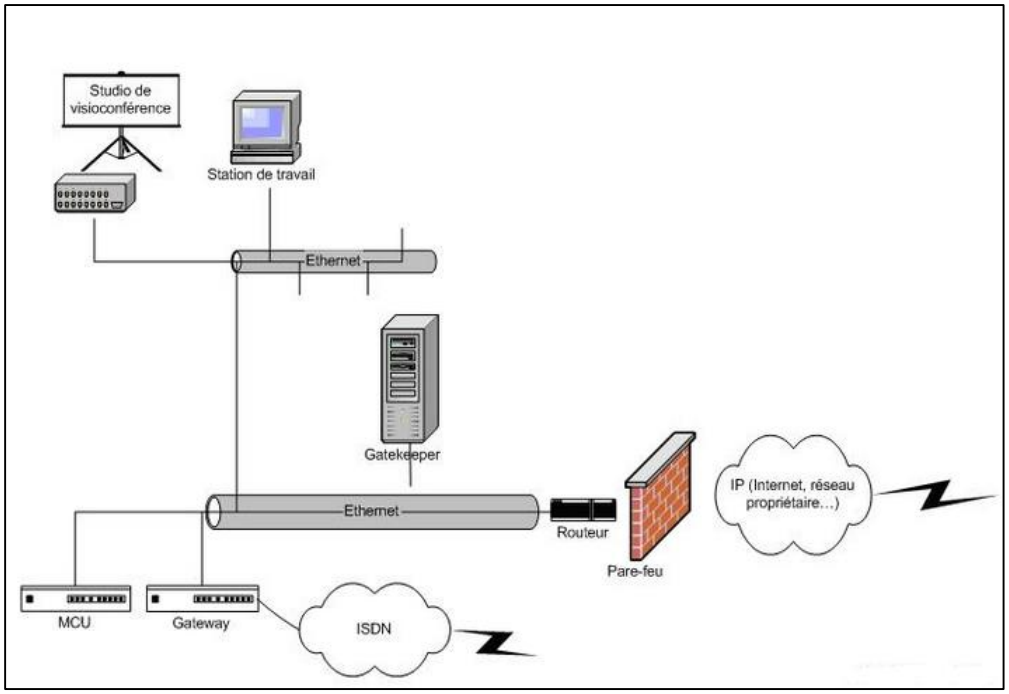

Figure 1.3 : Les composants de l'architecture H.323. [14]

- **Les terminaux H.323**

Le terminal peut être un ordinateur, un combiné téléphonique, un terminal spécialisé pour la vidéoconférence ou encore un télécopieur sur Internet. Le minimum imposé par H.323 est qu'il mette en œuvre la norme de compression de la parole G.711, qu'il utilise le protocole H.245 pour la négociation de l'ouverture d'un canal et l'établissement des paramètres de la communication, ainsi que le protocole de signalisation Q.931 pour l'établissement et l'arrêt des communications.

Le terminal possède également des fonctions optionnelles, notamment, pour le travail en groupe et le partage des documents. Il existe deux types de terminaux H.323, l'un de haute qualité (pour une utilisation sur LAN), l'autre optimisé pour de petites largeurs de bandes (28,8/33,6 kbit/s – G.723.1 et H.263).

- **Gateway ou les passerelles vers des réseaux classiques (RTC, RNIS, etc.)**

Les passerelles H.323 assurent l'interconnexion avec les autres réseaux, ex:(H.320/RNIS), les modems H.324, téléphones classiques, etc. Elles assurent la correspondance de signalisation de Q.931, la correspondance des signaux de

contrôle et la cohésion entre les médias (multiplexage, correspondance des débits, transcodage audio).

- **Gatekeeper ou les portiers**

Dans la norme H323, Le Gatekeeper est le point d'entrée au réseau pour un client H.323. Il définit une zone sur le réseau, appelée zone H.323 (voir figure 1.4 ci-dessous), regroupant plusieurs terminaux, Gateways et MCU dont il gère le trafic, le routage LAN, et l'allocation de la bande passante. Les clients ou les Gateway s'enregistrent auprès du Gatekeeper dès l'activation de celui-ci, ce qui leur permet de retrouver n'importe quel autre utilisateur à travers son identifiant fixe obtenu auprès de son Gatekeeper de rattachement.

Le Gatekeeper a pour fonction :

- La translation des alias H.323 vers des adresses IP, selon les spécifications RAS (Registration/Admission/Status) .
- Le contrôle d'accès, en interdisant les utilisateurs et les sessions non autorisés.
- La gestion de la bande passante, permettant à l'administrateur du réseau de limiter le nombre de visioconférences simultanées. Concrètement une fraction de la bande passante est allouée à la visioconférence pour ne pas gêner les applications critiques sur le LAN et le support des conférences multipoint ad-hoc.

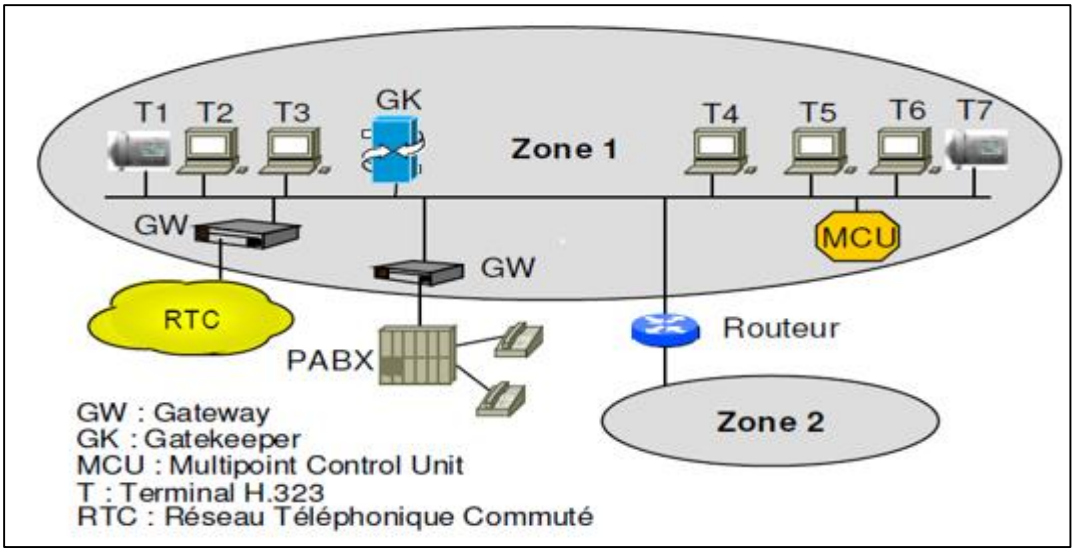

Figure 1. 4 : La zone H.323. [16]

- **Les MCU**

Les contrôleurs multipoint appelés MCU (Multipoint Control Unit) offrent aux utilisateurs la possibilité de faire des visioconférences à trois terminaux et plus en « présence continue » ou en « activation à la voix ». Une MCU consiste en un Contrôleur Multipoint (MC), auquel est rajouté un ou plusieurs Processeurs Multipoints (MP). Le MC prend en charge les négociations H.245 entre tous les terminaux pour harmoniser les paramètres audio et vidéo de chacun. Il contrôle également les ressources utilisées. Mais le MC ne traite pas directement avec les flux audio, vidéo ou données, c'est le MP qui se charge de récupérer les flux et de leurs faire subir les traitements nécessaires. Un MC peut contrôler plusieurs MP distribués sur le réseau et faisant partie d'autres MCU.

1.3.3. Avantages et inconvénients de la technologie H323 [19]

La technologie H.323 possède des avantages et des inconvénients. Parmi les avantages, nous citons :

- ***Gestion de la bande passante*** : H.323 permet une bonne gestion de la bande passante en posant des limites au flux audio/vidéo afin d'assurer le bon fonctionnement des applications critiques sur le LAN. Chaque terminal H.323 peut procéder à 'ajustement de la bande passante et la modification du débit en fonction du comportement du réseau en temps réel (latence, perte de paquets et gigue).
- ***Support Multipoint*** : H.323 permet de faire des conférences multipoint via une structure centralisée de type MCU (Multipoint Control Unit) ou en mode ad-hoc.
- ***Support Multicast*** : H.323 permet également de faire des transmissions en multicast.
- ***Interopérabilité :*** H.323 permet aux utilisateurs de ne pas se préoccuper de la manière dont se font les communications, les paramètres (les codecs, le débit…) sont négociés de manière transparente.
- ***Flexibilité :*** une conférence H.323 peut inclure des terminaux hétérogènes (studio de visioconférence, PC, téléphones…) qui peuvent partager selon le cas, de la voix de la vidéo et même des données grâce aux spécifications T.120.

Les inconvénients de la technologie H.323 sont :

- La complexité de mise en œuvre et les problèmes d'architecture en ce qui concerne la convergence des services de téléphonie et d'Internet, ainsi qu'un manque de modularité et de souplesse.
- Comprend de nombreuses options susceptibles d'être implémentées de façon différente par les constructeurs et donc de poser des problèmes d'interopérabilité.

1.4 Protocole SIP

1.4.1. Description générale du protocole SIP

Le protocole SIP (Session Initiation Protocol) est un protocole normalisé et standardisé par l'IETF (décrit par le RFC 3261 qui rend obsolète le RFC 2543, et complété par le RFC 3265) qui a été conçu pour établir, modifier et terminer des sessions multimédia [15]. Il se charge de l'authentification et de la localisation des multiples participants. Il se charge également de la négociation sur les types de média utilisables par les différents participants en encapsulant des messages SDP (Session Description Protocol) [1]. SIP ne transporte pas les données échangées durant la session comme la voix ou la vidéo. SIP étant indépendant de la transmission des données, tout type de données et de protocoles peut être utilisé pour cet échange. Cependant le protocole RTP (Real-time Transport Protocol) assure le plus souvent les sessions audio et vidéo. SIP remplace progressivement H323.

SIP est le standard ouvert de VoIP, interopérable, le plus étendu et vise à devenir le standard des télécommunications multimédia (son, image, etc.). Skype par exemple, qui utilise un format propriétaire, ne permet pas l'interopérabilité avec un autre réseau de voix sur IP et ne fournit que des passerelles payantes vers la téléphonie standard. SIP n'est donc pas seulement destiné à la VoIP mais pour de nombreuses autres applications telles que la visiophonie, la messagerie instantanée, la réalité virtuelle ou même les jeux vidéo [16].

1.4.2. Principe de fonctionnement

Puisque on choisira le protocole SIP pour effectuer notre travail, on s'approfondira à expliquer les différents aspects, caractéristiques qui font du protocole SIP un bon choix pour l'établissement de la session, les principales caractéristiques du protocole SIP sont :

- **Fixation d'un compte SIP**

Il est important de s'assurer que la personne appelée soit toujours joignable. Pour cela, un compte SIP sera associé à un nom unique. Par exemple, si un utilisateur d'un service de voix sur IP dispose d'un compte SIP et que chaque fois qu'il redémarre son ordinateur, son adresse IP change, il doit cependant toujours être joignable. Son compte SIP doit donc être associé à un serveur SIP (proxy SIP) dont l'adresse IP est fixe. Ce serveur lui allouera un compte et il permettra d'effectuer ou de recevoir des appels quelques soit son emplacement. Ce compte sera identifiable via son nom (ou pseudo).

- **Changement des caractéristiques durant une session**

Un utilisateur doit pouvoir modifier les caractéristiques d'un appel en cours. Par exemple, un appel initialement configuré en (voix uniquement) peut être modifié en (voix + vidéo).

- **Différents modes de communication**

Avec SIP, les utilisateurs qui ouvrent une session peuvent communiquer en mode point à point, en mode diffusif ou dans un mode combinant ceux-ci.

- **Mode Point à point *:*** on parle dans ce cas-là de « unicast » qui correspond à la communication entre deux machines.
- **Mode diffusif** : on parle dans ce cas-là de « multicast » (plusieurs utilisateurs via une unité de contrôle MCU – Multipoint Control Unit).
- **Combinatoire *:*** combine les deux modes précédents. Plusieurs utilisateurs interconnectés en multicast via un réseau à maillage complet de connexion.

- **Gestion des participants**

Durant une session d'appel, de nouveaux participants peuvent joindre les participants d'une session déjà ouverte en participant directement, en étant transférés ou en étant mis en attente (cette particularité rejoint les fonctionnalités d'un PABX par exemple, où l'appelant peut être transféré vers un numéro donné ou être mis en attente).

- **Négociation des médias supportés**

Cela permet à un groupe durant un appel de négocier sur les types de médias supportés. Par exemple, la vidéo peut être ou ne pas être supportée lors d'une session.

- **Adressage**

Les utilisateurs disposant d'un numéro (compte) SIP disposent d'une adresse ressemblant à une adresse mail (sip:numéro@serveursip.com). Le numéro SIP est unique pour chaque utilisateur.

- **Modèle d'échange**

Le protocole SIP repose sur un modèle Requête/Réponse. Les échanges entre un terminal appelant et un terminal appelé se font par l'intermédiaire de requêtes. La liste des requêtes échangées est la suivante :

- **Invite** : cette requête indique que l'application (ou utilisateur) correspondante à l'url SIP spécifié est invité à participer à une session. Le corps du message décrit cette session (par exemple : médias supportés par l'appelant). En cas de réponse favorable, l'invité doit spécifier les médias qu'il supporte.
- **Ack *:*** cette requête permet de confirmer que le terminal appelant a bien reçu une réponse définitive à une requête Invite.
- **Options *:*** un proxy server en mesure de contacter l'UAS (terminal) appelé, doit répondre à une requête Options en précisant ses capacités à contacter le même terminal.
- **Bye *:*** cette requête est utilisée par le terminal de l'appelé afin de signaler qu'il souhaite mettre un terme à la session.
- **Cancel *:*** cette requête est envoyée par un terminal ou un proxy server à fin d'annuler une requête non validée par une réponse finale comme, par exemple, si une machine ayant été invitée à participer à une session, et ayant accepté l'invitation ne reçoit pas de requête Ack, alors elle émet une requête Cancel.
- **Register *:*** cette méthode est utilisée par le client pour enregistrer l'adresse listée dans l'URL TO par le serveur auquel il est relié.
- **Codes d'erreurs**

Une réponse à une requête est caractérisée, par un code et un motif, appelés respectivement code d'état et raison phrase. Un code d'état est un entier codé sur 3 digits indiquant un résultat à l'issue de la réception d'une requête. Ce résultat est

précisé par une phrase, textbased (UTF-8), expliquant le motif du refus ou de l'acceptation de la requête. Le code d'état est donc destiné à l'automate gérant l'établissement des sessions SIP et les motifs aux programmeurs. Il existe 6 classes de réponses et donc de codes d'état, représentées par le premier digit :

- 1xx = Information - La requête a été reçue et continue à être traitée.
- 2xx = Succès - L'action a été reçue avec succès, comprise et acceptée.
- 3xx = Redirection - Une autre action doit être menée afin de valider la requête.
- 4xx = Erreur du client - La requête contient une syntaxe erronée ou ne peut pas être traitée par ce serveur.
- 5xx = Erreur du serveur - Le serveur n'a pas réussi à traiter une requête apparemment correcte.
- 6xx = Echec général - La requête ne peut être traitée par aucun serveur.

1.4.3. Rôle des composants

Dans un système SIP on trouve deux types de composantes, les agents utilisateurs (UAS, UAC) et un réseau de serveurs (Registrar, Proxy).

L'**UAS** (User Agent Server) représente l'agent de la partie appelée. C'est une application de type serveur qui contacte l'utilisateur lorsqu'une requête SIP est reçue. Et elle renvoie une réponse au nom de l'utilisateur.

L'**U.A.C** (User Agent Client) représente l'agent de la partie appelante. C'est une application de type client qui initie les requêtes.

Le **Registrar** est un serveur qui gère les requêtes REGISTER envoyées par les Users Agents pour signaler leur emplacement courant. Ces requêtes contiennent donc une adresse IP, associée à une URI, qui seront stockées dans une base de données (figure 1.5).

Les **URI SIP** sont très similaires dans leur forme à des adresses email :sip:utilisateur@domaine.com. Généralement, des mécanismes d'authentification permettent d'éviter que quiconque puisse s'enregistrer avec n'importe quelle URI.

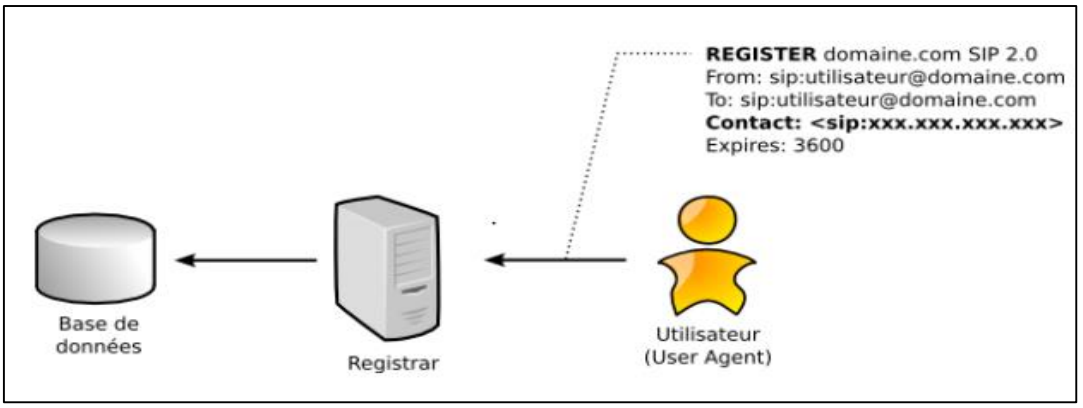

Figure 1.5 : Enregistrement d'un utilisateur. [5]

Un **Proxy SIP** sert d'être l'intermédiaire entre deux User Agents qui ne connaissent pas leurs emplacements respectifs (adresse IP). En effet, l'association URI-Adresse IP a été stockée préalablement dans une base de données par un Registrar. Le Proxy peut donc interroger cette base de données pour diriger les messages vers le destinataire. La figure 6 montre les étapes de l'interrogation du proxy la base de données.

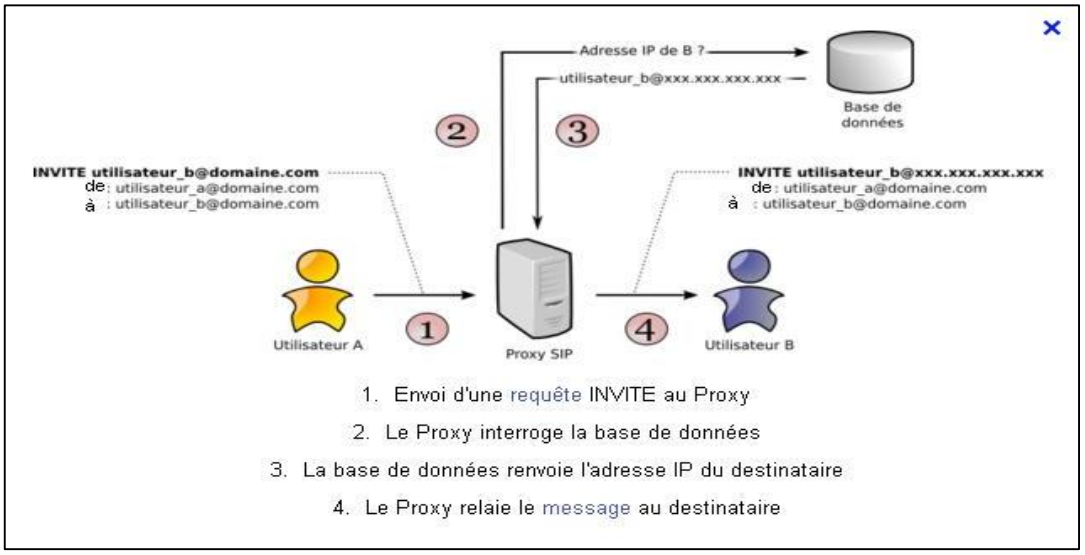

Figure 1.6 : Principe du protocole SIP. [5]

Le Proxy se contente de relayer uniquement les messages SIP pour établir, contrôler et terminer la session (voir figure 1.7). Une fois la session établie, les données, par exemple le flux RTP pour la VoIP, ne transitent pas par le serveur Proxy. Elles sont échangées directement entre les User Agents.

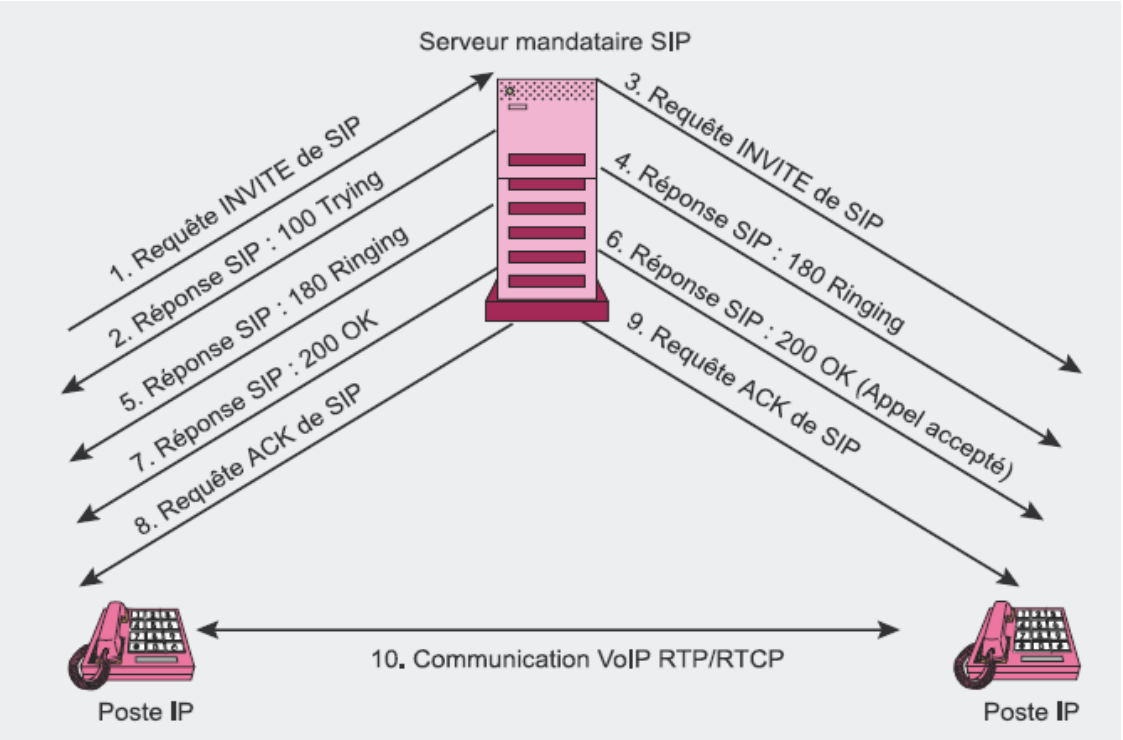

Figure 1.7 : Session SIP à travers un proxy. [17]

1.4.4. Avantages et inconvénients

Ouvert, standard, simple et flexible sont les principaux atouts du protocole SIP, voilà en détails ces différents avantages :

- Ouvert : les protocoles et documents officiels sont détaillés et accessibles à tous en téléchargement.
- Standard : l'IETF a normalisé le protocole et son évolution continue par la création ou l'évolution d'autres protocoles qui fonctionnent avec SIP.
- Simple : SIP est simple et très similaire à http.
- Flexible : SIP est également utilisé pour tout type de sessions multimédia (voix, vidéo, mais aussi musique, réalité virtuelle, etc.).
- Téléphonie sur réseaux publics : il existe de nombreuses passerelles (services payants) vers le réseau public de téléphonie (RTC, GSM, etc.) permettant d'émettre ou de recevoir des appels vocaux.
- Points communs avec H323 : l'utilisation du protocole RTP et quelques codecs son et vidéo sont en commun.

Par contre une mauvaise implémentation ou une implémentation incomplète du protocole SIP dans les User Agents peut perturber le fonctionnement ou générer du trafic superflu sur le réseau. Un autre inconvénient est le faible nombre d'utilisateurs : SIP est encore peu connu et utilisé par le grand public, n'ayant pas atteint une masse critique, il ne bénéficie pas de l'effet réseau.

1.5. Protocoles de transport

Nous décrivons deux autres protocoles de transport utilisés dans la voix sur IP à savoir l'RTP et le RTCP

1.5.1. Le protocole RTP

1.5.1.1. Description générale de RTP

RTP (Real time Transport Protocol), standardisé en 1996, est un protocole qui a été développé par l'IETF afin de faciliter le transport temps réel de bout en bout des flots données audio et vidéo sur les réseaux IP [19], c'est à dire sur les réseaux de paquets. RTP est un protocole qui se situe au niveau de l'application et qui utilise les protocoles sous-jacents de transport TCP ou UDP. Mais l'utilisation de RTP se fait généralement au-dessus d'UDP ce qui permet d'atteindre plus facilement le temps réel. Les applications temps réels comme la parole numérique ou la visioconférence constitue un véritable problème pour Internet. Qui dit application temps réel, dit présence d'une certaine qualité de service (QoS) que RTP ne garantit pas du fait qu'il fonctionne au niveau Applicatif. De plus RTP est un protocole qui se trouve dans un environnement multipoint, donc on peut dire que RTP possède à sa charge, la gestion du temps réel, mais aussi l'administration de la session multipoint.

1.5.1.2. Les fonctions de RTP

Le protocole RTP a pour but d'organiser les paquets à l'entrée du réseau et de les contrôler à la sortie. Ceci de façon à reformer les flux avec ses caractéristiques de départ. RTP est un protocole de bout en bout, volontairement incomplet et malléable pour s'adapter aux besoins des applications. Il sera intégré dans le noyau de l'application. Il laisse la responsabilité du contrôle aux équipements d'extrémité. Il est

aussi un protocole adapté aux applications présentant des propriétés temps réel. Il permet ainsi de :

- Mettre en place un séquencement des paquets par une numérotation et ce afin de permettre ainsi la détection des paquets perdus. Ceci est un point primordial dans la reconstitution des données. Mais il faut savoir quand même que la perte d'un paquet n'est pas un gros problème si les paquets ne sont pas perdus en trop grands nombres. Cependant il est très important de savoir quel est le paquet qui a été perdu afin de pouvoir pallier à cette perte.
- Identifier le contenu des données pour leurs associer un transport sécurisé et reconstituer la base de temps des flux (horodatage des paquets : possibilité de resynchronisation des flux par le récepteur)
- L'identification de la source c'est à dire l'identification de l'expéditeur du paquet. Dans un multicast l'identité de la source doit être connue et déterminée.
- Transporter les applications audio et vidéo dans des trames (avec des dimensions qui sont dépendantes des codecs qui effectuent la numérisation). Ces trames sont incluses dans des paquets afin d'être transportées et doivent, de ce fait, être récupérées facilement au moment de la phase de segmentation des paquets afin que l'application soit décodée correctement.

1.5.1.3. Avantages et inconvénients

Le protocole RTP permet de reconstituer la base de temps des différents flux multimédia (audio, vidéo, etc.) ; de détecter les pertes de paquets ; et d'identifier le contenu des paquets pour leur transmission sécurisée. Par contre, il ne permet pas de réserver des ressources dans le réseau ou d'apporter une fiabilité dans le réseau. Ainsi il ne garantit pas le délai de livraison.

1.5.2. Le protocole RTCP

1.5.2.1. Description générale de RTCP

Le protocole RTCP est fondé sur la transmission périodique de paquets de contrôle à tous les participants d'une session. C'est le protocole UDP (par exemple) qui permet le multiplexage des paquets de données RTP et des paquets de contrôle RTCP.

Le protocole RTP utilise le protocole RTCP, Real-time Transport Control Protocol, qui transporte les informations supplémentaires suivantes pour la gestion de la session.

Les récepteurs utilisent RTCP pour renvoyer vers les émetteurs un rapport sur la QoS. Ces rapports comprennent le nombre de paquets perdus, le paramètre indiquant la variance d'une distribution (plus communément appelé la gigue : c'est à dire les paquets qui arrivent régulièrement ou irrégulièrement) et le délai aller-retour. Ces informations permettent à la source de s'adapter, par exemple, de modifier le niveau de compression pour maintenir une QoS.

Parmi les principales fonctions qu'offre le protocole RTCP sont les suivants :

- Une synchronisation supplémentaire entre les médias : Les applications multimédias sont souvent transportées par des flots distincts. Par exemple, la voix, l'image ou mê me des applications numérisées sur plusieurs niveaux hiérarchiques peuvent voir les flots gérés et suivre des chemins différents.
- L'identification des participants à une session : en effet, les paquets RTCP contiennent des informations d'adresses, comme l'adresse d'un message électronique, un numéro de téléphone ou le nom d'un participant à une conférence téléphonique.
- Le contrôle de la session : en effet le protocole RTCP permet aux participants d'indiquer leur départ d'une conférence téléphonique (paquet Bye de RTCP) ou simplement de fournir une indication sur leur comportement.

Le protocole RTCP demande aux participants de la session d'envoyer périodiquement les informations citées ci-dessus. La périodicité est calculée en fonction du nombre de participants de l'application. On peut dire que les paquets RTP ne transportent que les données des utilisateurs. Tandis que les paquets RTCP ne transportent en temps réel, que de la supervision. On peut détailler les paquets de supervision en 5 types :

- SR (Sender Report) : Ce rapport regroupe des statistiques concernant la transmission (pourcentage de perte, nombre cumulé de paquets perdus, variation de délai (gigue), etc.). Ces rapports sont issus d'émetteurs actifs d'une session.
- RR (Receiver Report) : Ensemble de statistiques portant sur la communication entre les participants. Ces rapports sont issus des récepteurs d'une session.

- SDES (Source Description) : Carte de visite de la source (nom, e-mail, localisation).
- BYE : Message de fin de participation à une session.
- APP : Fonctions spécifiques à une application.

1.5.2.2 Point fort et limite du protocole RTCP [12] [13]

Le protocole de RTCP est adapté pour la transmission de données temps réel. Il permet d'effectuer un contrôle permanant sur une session et ces participants. Par contre il fonctionne en stratégie bout à bout. Et il ne peut pas contrôler l'élément principal de la communication, le réseau.

1.6. Points forts et limites de la voix sur IP

Différentes sont les raisons qui peuvent pousser les entreprises à s'orienter vers la VoIP comme solution pour la téléphonie. Les avantages les plus marqués sont :

- **Réduction des coûts *:*** En effet le trafic véhiculé à travers le réseau RTC est plus couteux que sur un réseau IP. Réductions importantes pour des communications internationales en utilisant le VoIP, ces réductions deviennent encore plus intéressantes dans la mutualisation voix/données du réseau IP intersites (WAN). Dans ce dernier cas, le gain est directement proportionnel au nombre de sites distants.
- **Standards ouverts *:*** La VoIP n'est plus uniquement H323, mais un usage multi- protocoles selon les besoins de services nécessaires. Par exemple, H323 fonctionne en mode égale à égale alors que MGCP fonctionne en mode centralisé. Ces différences de conception offrent immédiatement une différence dans l'exploitation des terminaisons considérées.
- **Un réseau voix, vidéo et données (à la fois) *:*** Grace à l'intégration de la voix comme une application supplémentaire dans un réseau IP, ce dernier va simplifier la gestion des trois applications (voix, réseau et vidéo) par un seul transport IP.

  Une simplification de gestion, mais également une mutualisation des efforts financiers vers un seul outil.

- **Un service PABX distribué ou centralisé *:*** Les PABX en réseau bénéficient de services centralisés tel que la messagerie vocale et la taxation. Cette même centralisation continue à être assurée sur un réseau VoIP sans limitation du nombre de canaux. Il convient pour en assurer une bonne utilisation de dimensionner convenablement le lien réseau. L'utilisation de la VoIP met en commun un média qui peut à la fois offrir à un moment précis une bande passante maximum à la donnée, et dans une autre période une bande passante maximum à la voix, garantissant toujours la priorité à celle-ci.

Les points faibles de la voix sur IP sont :

- **Fiabilité et qualité sonore *:*** un des problèmes les plus importants de la téléphonie sur IP est la qualité de la retransmission qui n'est pas encore optimale. En effet, des désagréments tels la qualité de la reproduction de la voix du correspondant ainsi que le délai entre le moment où l'un des interlocuteurs parle et le moment où l'autre entend peuvent être extrêmement problématiques. De plus, il se peut que des morceaux de la conversation manquent (des paquets perdus pendant le transfert) sans être en mesure de savoir si des paquets ont été perdus et à quel moment.
- **Dépendance de l'infrastructure technologique et support administratif exigeant *:*** les centres de relations IP peuvent être particulièrement vulnérables en cas d'improductivité de l'infrastructure. Par exemple, si la base de données n'est pas disponible, les centres ne peuvent tout simplement pas recevoir d'appels. La convergence de la voix et des données dans un seul système signifie que la stabilité du système devient plus impo rtante que jamais et l'organisation doit être préparée à travailler avec efficience ou à encourir les conséquences.
- **Vol :** les attaquants qui parviennent à accéder à un serveur VoIP peuvent également accéder aux messages vocaux stockés et au même au service téléphonique pour écouter des conversations ou effectuer des appels gratuits aux noms d'autres comptes.
- **Attaque de virus *:*** si un serveur VoIP est infecté par un virus, les utilisateurs risquent de ne plus pouvoir accéder au réseau téléphonique. Le virus peut également infecter d'autres ordinateurs connectés au système.

1.7. Discussion

Comme on a pu le voir tout au long de ce chapitre, la VoIP est la solution la plus rentable pour effectuer des conversations.

Aujourd'hui, la technologie VoIP est généralement utilisé dans tous les bureaux d'entreprise pour effectuer des appels à longue ou à courte distance.

Comme le trafic de données continue d'augmenter et dépasser celui du trafic voix, la convergence et l'intégration de ces technologies non seulement va continuer de s'améliorer, mais va aussi ouvrir la voie à un moyen vraiment unifiés de communication. Mise en œuvre, la VoIP peut offrir des avantages significatifs et des économies pour ses utilisateurs.

Chapitre 2

Installation et configuration d'une solution de VoIP basée sur le serveur Asterisk

2.1. Préambule

Dans ce chapitre, nous allons réaliser une solution assurant la transmission de la voix sure le un réseau IP. Pour se faire nous allons utiliser la solution libre Asterisk.
On montrera les étapes d'installation et de configuration d'asterisk sous le système d'exploitation linux Ubuntu ainsi que l'installation et la configuration de ZOIPER softphone qui est un logiciel gratuit pour la téléphonie sur IP.

2.2. Présentation d'Asterisk [9] [11]

Asterisk est un logiciel open source qui transforme un ordinateur en un autocommutateur téléphonique PABX (Private Automatic Branche eXchange) à part entière mais d'implémentation logicielle, compatible linux, qui s'interconnecte avec quasiment tous les équipements de téléphonie de base standard et peu coûteux. Il a été créé par Mark Spencer de la société Digium en 1999, et publié sous licence GPL. Son objectif était alors de fournir à Linux un commutateur téléphonique complet et totalement libre.
Aujourd'hui Asterisk est un PABX (Private Automatic Branche eXchange) d'une rare puissance et souplesse, il a été conçu pour une flexibilité maximale et reste un système ouvert à de nouvelles applications. Il fournit donc toutes les fonctionnalités attendues d'un PABX tel que les appels téléphoniques, la messagerie vocale, les fils d'attentes, les conférences, le renvoi d'appel mais aussi la voix sur IP et n'a besoin d'aucun matériel supplémentaire pour l'assurer. Il implémente plusieurs protocoles H.320, H.323, SIP et IAX.

2.3. Architecture du réseau VoIP déployé

La figure 2.1 montre l'architecture adoptée au cours de la configuration de la solution de VoIP à base d'Asterisk

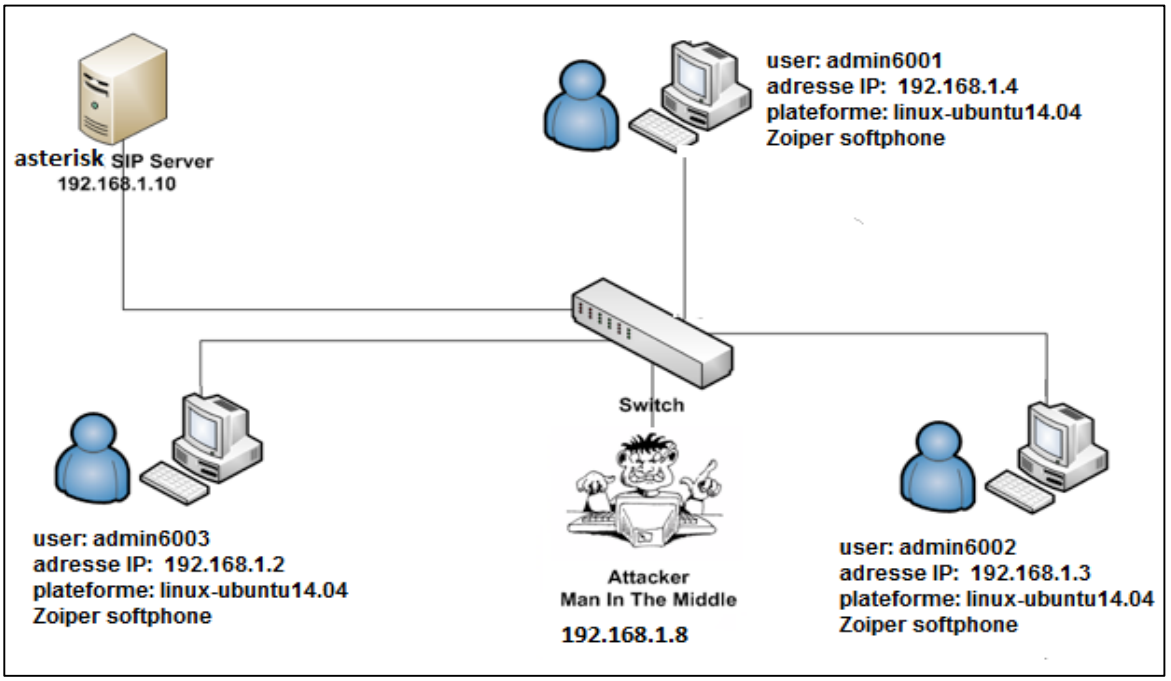

Figure 2.1 : Architecture du réseau VoIP à réaliser

- **Les trois clients SIP** : Sont des machines sur lesquelles est installé le système d'exploitation Linux-Ubuntu 14.04 et un client Sip utilisant ZOIPER softphone
- **Poste attaquant** : Dans lequel on a installé BackTrack 5 r3 pour réaliser les attaques.
- **Machine serveur** : Sur laquelle installé un système d'exploitation Linux Ubuntu 14.04, et le serveur de VoIP Asterisk

2.4. Installation d'Asterisk 11.22.0

Avant d'installer Asterisk, il faut préparer le système sous lequel on installera notre serveur. Pour cela, il faut installer tout d'abord les pré-requis nécessaires.

2.4.1. Téléchargement des pré-requis

Les pré-requis nécessaires pour que l'installation du serveur Asterisk s'accomplit avec succès, Peuvent être téléchargé avec la commande suivante :

```
# apt-get install gcc make pkg-config build-essential wget libssl-dev libncurses5-dev
  libnewt-dev libxml2-dev linux-headers-$(uname -r) uuid-dev libsqlite3-dev
  libjansson-dev
```

Le meilleur chemin pour obtenir le code source d'Asterisk et ces paquetages est de les télécharger à partir du site web www.asterisk.org. On peut aussi les télécharger directement du serveur de Digium.

2.4.2. Téléchargement des codes sources

Voici les lignes de commandes nécessaires pour le téléchargement d'Asterisk, Libpri et dahdi, on identifie l'url. Après on télécharge via la commande **wget**

```
# cd /usr/src/                                    accès au dossier src (source)
# sudo wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-11- current.tar.gz
# sudo wget http://downloads.digium.com/pub/libpri/libpri-1.4-current.tar.gz
# sudo wget http://downloads.asterisk.org/pub/telephony/dahdi-linux-complete/dahdi-linux-
complete-current.tar.gz
```

2.4.3. Extraction des paquetages

Les paquetages téléchargés sont des archives compressées qui contiennent le code source, on aura besoin de les extraire, en utilisant la commande **tar**, avant de les compiler.

```
# cd /usr/src/
# tar zxvf dahdi-linux-complete-current.tar.gz
# tar zxvf libpri-1.4-current.tar.gz
# tar zxvf asterisk-11-current.tar.gz
```

2.4.4. Compilation et installation :

Le paquetage DAHDI pour Digium Asterisk Hardware Device Interface est un composant lié à Asterisk pour gérer la communication entre Asterisk et les différents types de cartes physique qu'on peut connecter au serveur. [9]

Voici les lignes de commandes nécessaires pour le compiler :

```
# cd  /usr/src/dahdi*          =accès au dossier du dahdi
# make clean                   =supprime les fichiers inutiles après installation.
# ./configure                  =construction d'un nouveau fichier makefile.
# make menuselect              =execution de la partie menuselect dans le fichier make file
# make                         =compilation du code source
# make install                 =execution de la partie install dans makefile.
```

Makefile est un fichier qui contient les instructions à exécuter à partir des commandes,
./configure, make, make install, make config, etc. chacune de ces commandes exécute le code approprié à elle dans ce fichier

Libpri est utilisé par les décideurs du multiplexage temporel (TDM) des appareils VoIP, mais même s'il n'y a pas le matériel installé, il est conseillé de compiler et installer cette bibliothèque. Elle doit être compilé et installé avant Asterisk, car elle sera détecté et utilisélorsqu'Asterisk est compilé.

```
#  cd /usr/src/libpri-1.4
#  make clean
#  make
#  make install
```

Maintenant que tous les paquetages nécessaires sont installés, on install asterisk avec les commandes suivantes :

# cd /usr/src/asterisk-11.22.0	accès au fichier asterisk
# make clean	
# ./configure	configuration
# make menuselect	
# make install	installation
# make samples	
# make config	cette commande charge le serveur Asterisk au démarrage du système

Les figures 2.2 et 2.3 montrent qu'Asterisk s'est bien configurer et installer

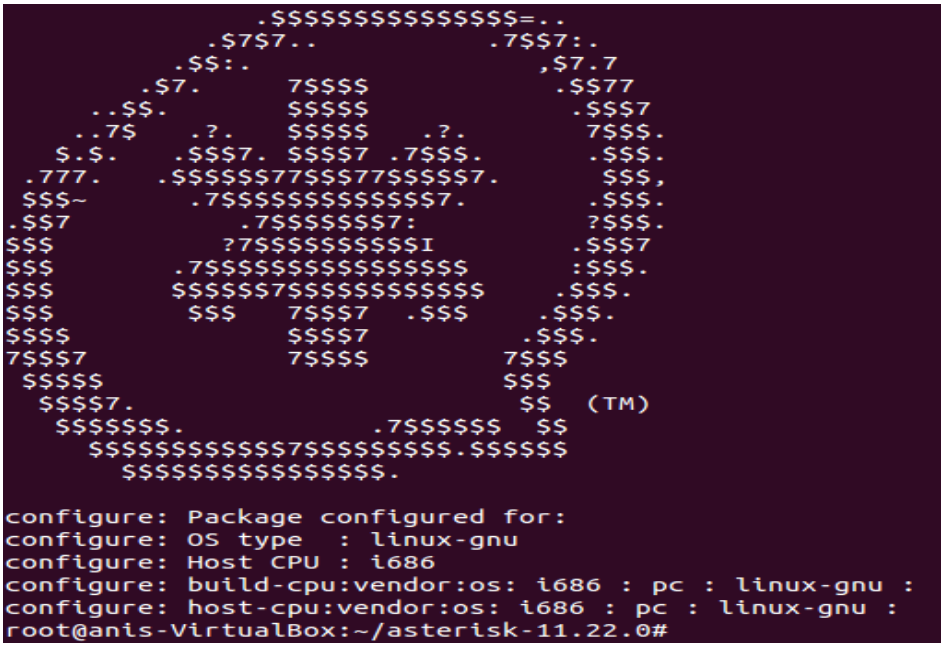

Figure 2.2: configuration d'Asterisk réussie

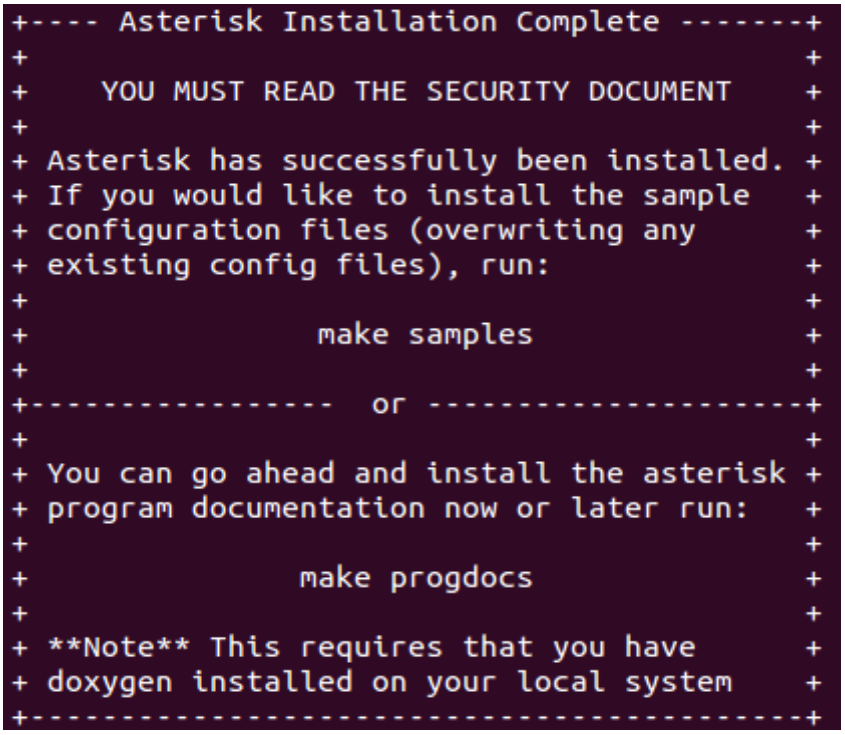

Figure 2.3: installation d'asterisk réussi

Il suffit maintenant de lancer le serveur, (voir figure 2.4 et 2.5) et de se connecter à la console CLI (Command Line Interface), via les commandes suivantes :

# /etc/init.d/asterisk start	Démarrage d'asterisk
# asterisk -r	accès à la console

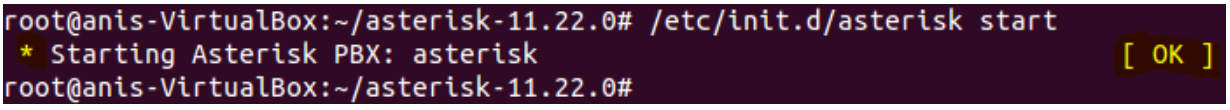

Figure 2.4: Démarrage d'asterisk

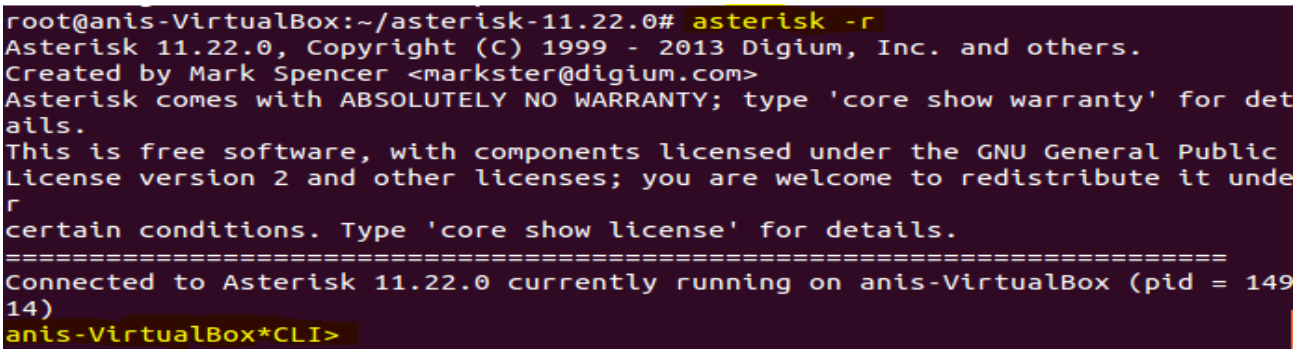

Figure 2.5: accès au mode console

2.5. Configuration d'Asterisk

2.5.1. 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/ : Continent les appels sortants d'Asterisk d'Asterisk.
- **/etc/asterisk/** : Contient tous les fichiers de configurations

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

- **agents.conf:** Contient la configuration de l'utilisation des agents, comme dans le cas d'un centre d'appel. Ceci nous permet de définir les agents et de leurs assigner des ID et des mots de passe.

- **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. C'est le fichier qui nous

intéresse le plus dans ce travail.

- **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:** Définit les utilisateurs du protocole SIP et leurs options. On peut aussi définir d'autres options globales pour SIP telles que, quels ports utiliser et les timeouts qu'on va imposer. Nous focalisons sur ce fichier puisque notre solution est basée sur le protocole SIP.
- **zapata.conf:** Configure les paramètres de l'interface téléphonique Zapata

2.5.2. Configuration des comptes

Les fichiers à configurer sont sip.conf, et extensions.conf. Dans le fichier sip.conf, on créera des utilisateurs utilisant le protocole SIP pour l'établissement de la connexion, on accède au fichier sip.conf en tapant **nano /etc/asterisk/sip.conf** sur le terminal **.**

Voici les lignes à introduire dans le fichier sip.conf pour la création de trois clients SIP :

[6001]	(Numéro Sip du client admin6001)
type=friend	(spécifie le type d'utilisateur)
secret=secret	(mot de passe du compte sip)
host=dynamic	(pour se connecter au compte à partir de n'importe qu'elle adresseIP)
dtmfmode=rfc2833	(type de rfc utilise)
disallow=all	(désactivation de tout les codecs)
allow=ulaw	(activation du codecs ulaw)
fullname= admin 6001	(prénom et nom de l'utilisateur qui va être affiché surl'écran)
username=admin6001	(nom de l'utilisateur)
context=work	(spécifie le type de routage à utiliser)
[6002]	(Numéro Sip du client admin6002)
type=friend	
secret=secret	
host=dynamic	
dtmfmode=rfc2833	
disallow=all	
allow=ulaw	
fullname= admin 6002	
username=admin6002	
context=work	

```
[6003]                                  (Numéro Sip du client admin6003)

type=friend
secret=secret
host=dynamic
dtmfmode=rfc2833
disallow=all
allow=ulaw
fullname= admin 6003
username=admin6003
context=work
```

Une fois le fichier sip.conf enregistré allez dans la console Asterisk, tapez **sip reload** ensuite **reload** enfin tapez la commande **sip show users**.Les comptes utilisateurs que nous venons de créer devrait y apparaitre comme indiqué dans la figure suivante

```
*CLI> sip show users
Username                        Secret           Accountcode      De
f.Context       ACL   Forcerport
6003                            secret                            wo
rk              No    No
6002                            secret                            wo
rk              No    No
6001                            secret                            wo
rk              No    No
```

Figure 2.6 : création des comptes utilisateurs

Maintenant que les utilisateurs sont créés, passant à la configuration du Dial Plan ou le plan d'appel dans le fichier **extensions.conf**

2.5.3. Configuration des extensions :

La configuration des extensions s'effectue dans le fichier **extensions.conf**

```
[work]                                  (il faut saisir le nom du context entre crochet)

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

exten => : déclare l'extension (on peut aussi simplement dire numéros)

_6XXX : Prend les extensions (ou numéros) de 6000 a 6999 , le « _ » permet d'utiliser des regex

1 : Ordre de l'extension

Dial : application qui va être utilisé

SIP: Protocol qui va être utilisé

${EXTEN} : variable de l'extension composé, si on appelle le 6001 la variable ${EXTEN} prendra comme valeur

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

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.

Il existe d'autres options qu'on peut ajouter dans le fichier extensions.conf, telles que la boîte vocale et le renvoi d'appel.

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.

```
exten => _6001,1,Answer
exten => _6001,2,Playback(répondeur)
exten => _6001,3,Voicemail(9)          (9 est le numéro de la boîte vocale)
exten => _6001,4,Hangup()
```

Dans chaque ajout ou modification d'un client, il faut mettre à jour le serveur Asterisk en utilisant les commandes suivantes :

```
*CLI> sip reload
*CLI> dialplan reload
*CLI> reload
```

2.6. Installation et configuration des softphone :

Zoiper est un softphone freeware(gratuit), son utilisation est simple, il est disponible pour les différents systèmes d'exploitation, Windows, Mac et Linux, téléchargeable depuis le lien suivant :
http://www.zoiper.com/en/voip-softphone/download/zoiper3#linux/step3

2.6.1. Installation de ZOIPER :

L'installation sous Windows est classique, mais pour linux, il faut décompresser le fichier, accéder au répertoire et lancer l'exécutable, comme indiquer ci-dessous :

# tar -zxvf Zoiper_3.2_linux_Free_32Bit_64Bit.tar.gz	(décompression du fichier Zoiper Avec la commande **tar**)
# cd Zoiper_3.2_linux_Free _32Bit_64Bit	(accès au fichier de Zoiper)
# sudo ./ Zoiper3.2_linux_Free_32Bit.run	(installation de Zoiper)

Après l'installation, on accède à Zoiper depuis l'icône sur le bureau d'Ubuntu.

2.6.2. Configuration de Zoiper

Pour configurer le client Zoiper les utilisateurs doivent accéder au menu « Settings » puis de ce menu vers le sous menu « create a new account ». Dans la fenêtre qui s'ouvre, il suffit de remplir les champs avec les paramètres de l'utilisateur concerné.
Prenant l'exemple de l'utilisateur 6001 (voir figure 2.7) :

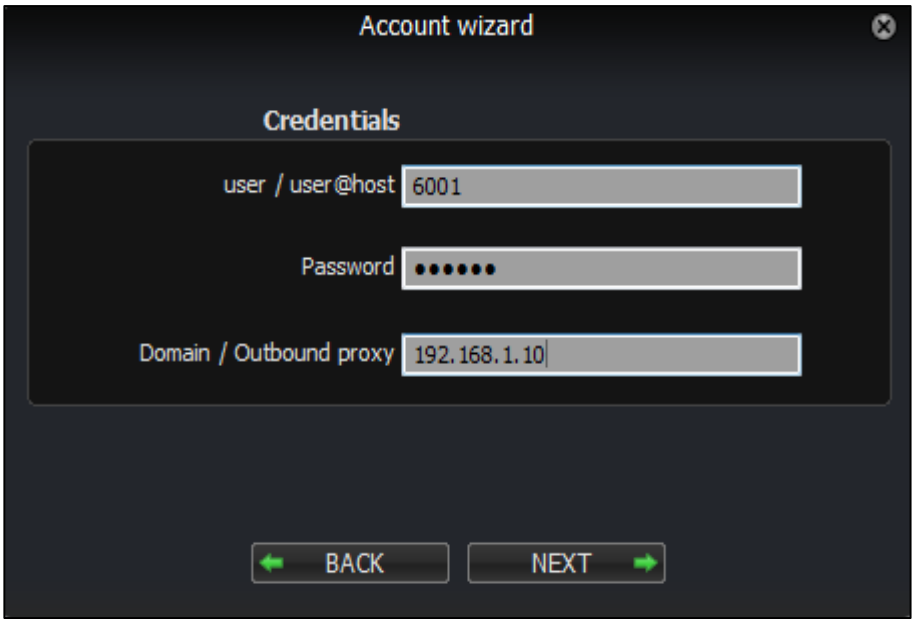

Figure 2.7 : Configuration du compte du client « 6001 »

-**User** : le nom d'utilisateur qui va être affiché à l'écran du softphone

-**Domain** : l'adresse IP du server asterisk (SIP server)

-**Password** : le mot de passe (secret)

Il est à noter qu'afin que 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 « Registration succeded » s'affichera, indiquant que les communications sont désormais possibles. Sinon, un message d'erreur explique le motif qui a fait échouer le processus.

Même procédures pour les autres utilisateurs.

2.6.3. Simulation d'un appel avec Zoiper

Après avoir terminé l'installation et la configuration du serveur Asterisk et des comptes utilisateur, nous avons testé le réseau de VoIP déployé en effectuant un appel de l'utilisateur (client) 6001 vers l'utilisateur 6002 et de l'utilisateur 6002 vers l'utilisateur 6003 (voir figure 2.8 et 2.9)

Figure 2.8 : appel de l'utilisateur 6001 vers 6002

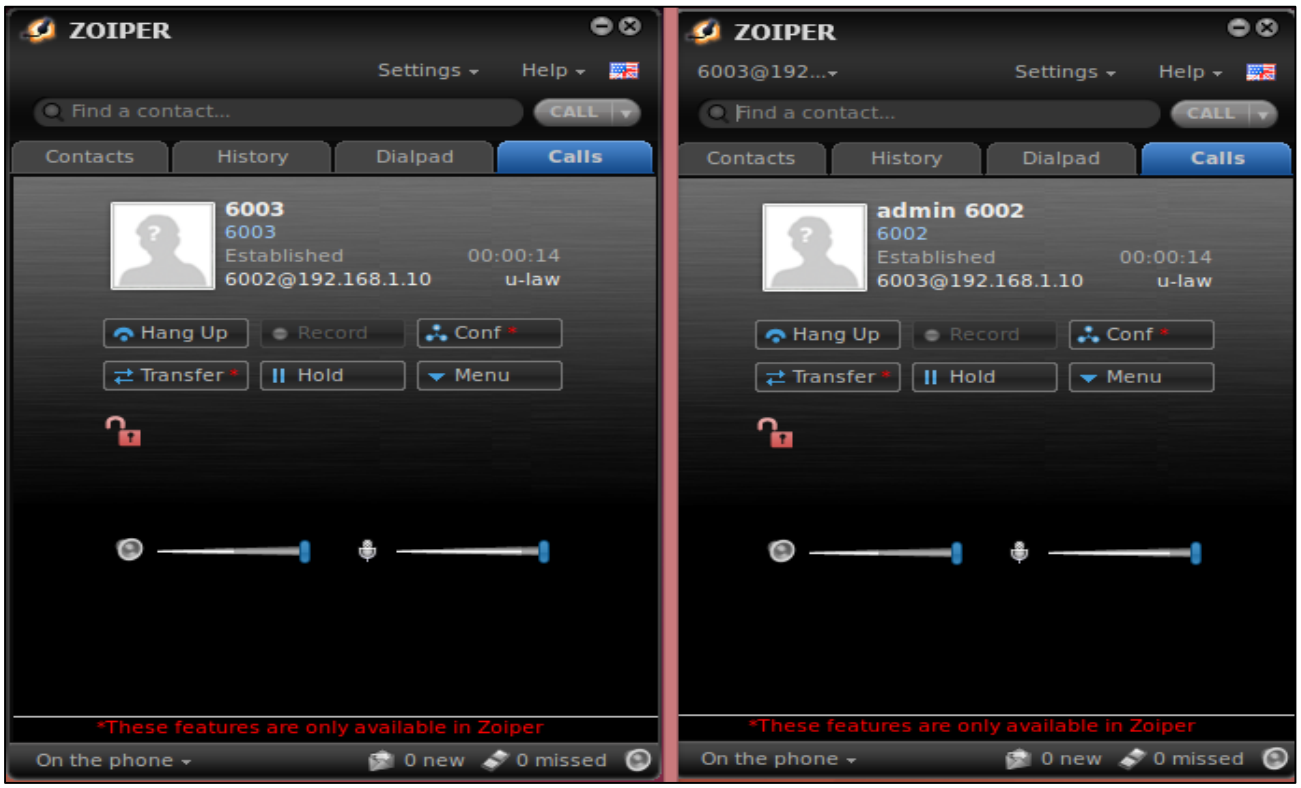

Figure 2.9 : appel de l'utilisateur 6002 vers 6003

Les tests d'appels entre les postes clients ont été effectués avec succès, le service de VoIP est opérationnel et a été bien mis en place dans notre réseau, maintenant les clients peuvent s'appeler gratuitement et facilement depuis leurs ordinateurs.

2.7. Discussion :

Dans ce chapitre, nous avons vu les étapes d'installation et de configuration de la solution libre Asterisk ainsi que le softphone Zoiper, suivis de testes d'appels entre les différents clients.

Après avoir expérimenté Asterisk, nous pouvons dire que c'est la solution libre la plus efficaces et rentable pour effectuer des conversations de qualité mais surtout gratuits au sein d'un réseau IP.

L'évolutivité, et la professionnalisation de ce système en font une solution concurrente pour les PBX Hardware

Chapitre 3

Attaques, vulnérabilités et hacking dans les réseaux VOIP

3.1. Préambule :

L'opportunité de migrer de la téléphonie classique vers la téléphonie IP, a offert plusieurs avantages pour les entreprises, et leurs a permis de bénéficier de nouveaux services tel que la vidéoconférence et la transmission des données. Cependant, avec la mise en place de cette nouvelle architecture, la voix devient aussi vulnérable que les applications classiques aux failles de sécurité sur le réseau, posant ainsi une nouvelle problématique L'intégration de ces services dans une seule plateforme nécessite plus de sécurité

Dans ce chapitre, nous présenterons des attaques qui menacent la VoIP, et nous détaillerons quelques-unes. Nous passerons ensuite à la réalisation de quelques techniques de hacking sur le réseau de voix sur IP déployé en s'appuyant sur les attaques qu'on aura définie.

Le système VoIP utilise l'Internet, et particulièrement le protocole IP. De ce fait les vulnérabilités de celui-ci.

Les attaques sur les réseaux VoIP peuvent être classées en deux types : les attaques internes et les attaques externes. Les attaques externes sont lancées par des personnes situées dans un autre réseau, 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.

Il existe deux principales classes de vulnérabilités sur un environnement VoIP. La première dépend des protocoles utilisés (SIP, H.323…) et la deuxième est reliée aux systèmes sur lesquels les éléments VoIP sont implémentés. Chaque protocole ou service a ses propres vulnérabilités.

3.2. Attaques sur la VoIP :

Un appel téléphonique VoIP est constitué de deux parties : la signalisation, qui instaure l'appel, et les flux de media, qui transporte la voix.

La signalisation, en particulier SIP, transmet les entêtes et la charge utile (Payload) du paquet en texte clair, ce qui permet à un attaquant de lire et falsifier facilement les paquets. Elle est donc vulnérable aux attaques qui essaient de voler ou perturber le service téléphonique, et à l'écoute clandestine qui recherche des informations sur un compte utilisateur valide, pour passer des appels gratuits par exemple.

La signalisation utilise, en général, le port par défaut UDP/TCP 5060. Le firewall doit être capable d'inspecter les paquets de signalisation et ouvre ce port afin de leurs autoriser l'accès au réseau. Un firewall qui n'est pas compatible aux protocoles de la VoIP doit être configuré manuellement pour laisser le port 5060 ouvert, créant un trou pour des attaques contre les éléments qui écoutent l'activité sur ce port.

Le protocole RTP, utilisé pour le transport des flux multimédia, présente également plusieurs vulnérabilités dues à l'absence d'authentification et de chiffrage. Chaque entête d'un paquet RTP contient un numéro de séquence qui permet au destinataire de reconstituer les paquets de la voix dans l'ordre approprié.

Cependant, un attaquant peut facilement injecter des paquets artificiels avec un numéro de séquence plus élevé. En conséquence, ces paquets seront diffusés à la place des vrais paquets.

Généralement, les flux multimédias contournent les serveurs p r o x y e t circulent directement entre les points finaux. Les menaces habituelles conte le flux de la voix sont l'interruption de transport et l'écoute clandestine.

Les protocoles de la VoIP utilisent TCP et UDP comme moyen de transport et par conséquent sont aussi vulnérables à toutes les attaques contre ces protocoles, telles le détournement de session (TCP) (session Hijacking) et la mystification (UDP) (Spoofing), etc. Les types d'attaques les plus fréquentes contre un system VoIP sont :

3.2.1. Sniffing

Le Sniffing ou reniflement de trafics constitue l'une des méthodes couramment utilisées par les pirates informatiques pour espionner le trafic sur le réseau. Dans la pratique, les hackers ont généralement recours à ce procédé pour détecter tous les messages circulant sur le réseau en récupérant des mots de passe et des données sensibles. Ces informations peuvent par exemple être employées pour mettre en place une attaque contre d'autres systèmes ou données.

Plusieurs outils requis pour le sniffing tel que Ettercap, y compris pour le protocole H.323 et des plugins SIP, sont disponibles en open source

3.2.2. L'ARP Poisoning (Empoisonnement ARP) [20] [21]

L'ARP est un protocole de couche 2 du modèle OSI et TCP/IP, lors des échanges sur un réseau, les adresses MAC sont utilisées pour la formation des trames Ethernet et le rôle de l'ARP est donc de fournir, à partir d'une adresse IP, l'adresse MAC correspondante.

Par défaut, une carte réseau refuse les paquets reçus n'ayant pas son adresse MAC comme adresse MAC de destination.

On pourrait comparer cela au fait de recevoir une lettre alors que le nom du destinataire n'est pas le nôtre, nos cartes réseaux, en individus sages et réfléchis, refusent donc d'ouvrir ces lettres. Une trame Ethernet complète doit donc forcément avoir un couple IP-MAC correcte dans les champs destination pour transiter sans encombre d'un hôte A à un hôte B.

Cependant, le fonctionnement d'ARP sur les ordinateurs est le suivant : à chaque réception d'un paquet, la carte réseau va vérifier le couple IP-MAC qu'il contient et mettre à jour sa table ARP (aussi appelée cache ARP) si le couple trouvé n'est pas enregistré. Ceci dans le but de ne pas faire de requête ARP à chaque échange et de remplir son cache ARP de façon dynamique**.**

Le principe de l'ARP *ou du* MAC spoofing *(spoofing voulant dire* "falsifier", "usurper") est d'envoyer des informations à un système afin de lui faire enregistrer des informations qui ne sont pas les bonnes et qui usurpent l'identité (la relation IP-MAC) d'un autre système, ainsi les trames vont transiter par la machine du pirate car les

victimes vont former leurs trames avec l'adresse MAC de cette dernière, ce qui va permettre à l'attaquant de capturer les paquets de signalisation et media circulant entre les machines victimes de l'ARP spoofing et de les décoder ensuite avec des logiciels appropriés

3.2.3 L'ARP Flooding [4] [22]

L'ARP Flooding ou saturation de la table ARP est une technique employée pour Saboter un commutateur en lui envoyant des milliers d'adresses MAC jusqu'à saturation. Les commutateurs tiennent à jour une table d'apprentissage (MAC) qui associe les adresses MAC Ethernet aux différents ports physiques du commutateur. Cela permet au commutateur d'envoyer les trames Ethernet directement aux machines à qui elles sont destinées. Dans une attaque par saturation, le pirate envoie de nombreuses trames Ethernet au commutateur, chacune d'entre elles ayant une adresse MAC source différente. Le but est de remplir l'espace nécessairement limité de la mémoire du commutateur consacré à la table MAC et lorsque ce dernier se sature, il fonctionne comme un concentrateur (Hub) qui va envoyer les trames sur tous les ports et c'est l'effet désiré par l'attaquant pouvant ainsi écouter le réseau et capturer les paquets qui est circule.

Un réseau de VoIP est très sensible à ce genre d'attaque visant la confidentialité et l'intégrité des informations qui sont échangé entre les différents clients.

L'ARP Poisoning et l'ARP Flooding sont des attaques souvent utilisées avant d'effectuer d'autre attaques ″Man in the middle″ notamment l'écoute clandestine ou le sniffing.

3.2.4. 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 [10]. Des outils tels que WIRESHARK ou VOMIT (Voice Over Misconfigured Internet Téléphones) permettent de réaliser cette attaque. VOMIT convertit les paquets sniffés en fichier.wav qui peut être réécouté avec n'importe quel lecteur de fichier, la figure 3.1 illustre un exemple d'un détournement d'appel.

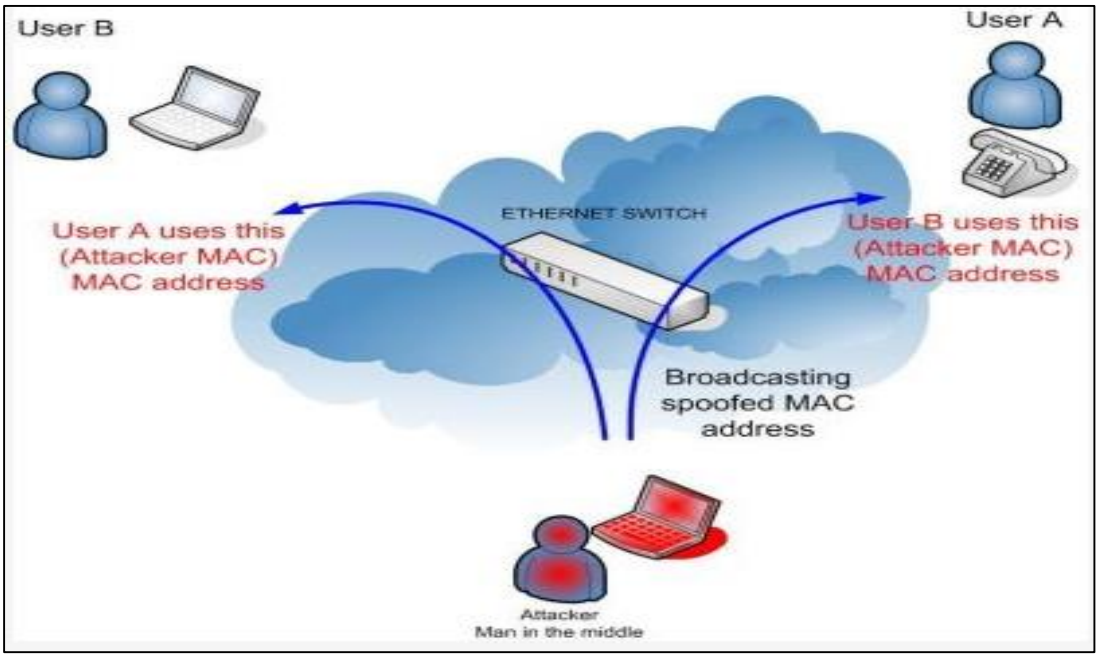

Figure 3.1 : Exemple de détournement d'appel" Man in the middle". [19]

Le principe de l'écoute clandestine est montré dans la figure (3.2) 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 de la machine pirate.
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 de la machine pirate.
4. Désactiver la vérification des adresses MAC sur la machine d'attaque afin que le trafic puisse circuler entre les 2 victimes

3.2.5. Suivie des appels

Appelé aussi Call tracking, cette attaque se fait au niveau du réseau LAN/VPN et 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. [14]

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

3.2.6. Injection de paquet RTP [15] [7]

Cette attaque se fait au niveau du réseau LAN/VPN. Elle cible le serveur registrar, et a 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.

3.2.7. Les Spam [14] [19]

Trois formes principales de spams sont jusqu'à maintenant identifiés dans SIP :

-Call Spam : Ce type de spam est défini comme une masse de tentatives d'initiation de session (des requêtes INVITE) non sollicitées.
Généralement c'est un UAC (User Agent Client) qui lance, en parallèle, un grand nombre d'appels. Si l'appel est établi, l'application génère un ACK, rejoue une annonce préenregistrée, et ensuite termine l'appel.
-IM (Instant Message) Spam : Ce type de spam est semblable à celui de l'e-mail. Il est défini comme une masse de messages instantanés non sollicitées. Les IM spams sont pour la plupart envoyés sous forme de requête SIP. Ce pourraient être des requêtes INVITE avec un entête « Subject » très grand, ou des requêtes INVITE avec un corps en format texte ou HTML. Bien-sûr, l'IM spam est beaucoup plus intrusif que le spam email, car dans les systèmes actuels, les IMs apparaissent automatiquement sous forme de pop-up à l'utilisateur.

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

3.2.8. Le déni de service (DOS : Denial of service) [1] [4] [8]

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. Dans ce type d'attaque les pirates se dissimulent parfois grâce à des machines-rebonds (ou machines zombies), utilisées à l'insu de leurs propriétaires. Un ensemble de machines-rebonds, est contrôlable par un pirate après infection de chacune d'elles par un programme de type porte dérobée (backdoor).

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

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.

- Fragmentation des paquets IP : Par la fragmentation des paquets, il est possible de rendre hors service de nombreux systèmes d'exploitation et dispositif VoIP par le biais de la consommation des ressources. Il existe de nombreuses variantes d'attaques par fragmentation, parmi les plus populaires teardrop, opentear, nestea, jolt, boink, et Ping of death.

Couche transport :

- L'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 :

✓ 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.

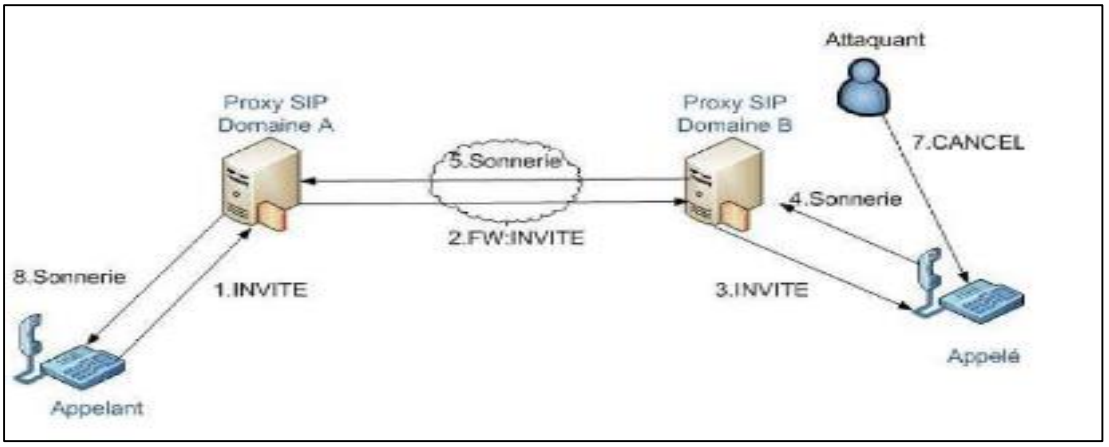

Figure 3.2 : Attaque DoS via une requête CANCEL. [10]

La figure suivante montre un scénario d'attaque DoS CANCEL, l'utilisateur toto initie l'appel, envoie une invitation (1) au proxy auquel il est rattaché. Le proxy du domaine A achemine la requête (2) au proxy qui est responsable de l'utilisateur titi. Ensuite c'est le proxy du domaine B qui prend le relais et achemine la requête INVITE (3) qui arrive enfin à destination. Le dispositif de titi, quand il reçoit l'invitation, sonne (4).

Cette information est réacheminée jusqu'au dispositif de toto. L'attaquant qui surveille l'activité du proxy SIP du domaine B envoie une requête CANCEL (7) avant que titi n'ait pu envoyer la réponse OK qui accepte l'appel. Cette requête annulera la requête en attente (l'INVITE), l'appel n'a pas lieu.

- ✓ 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.

Les attaquants, de cette façon, peuvent supprimer l'enregistrement de quelques-uns des utilisateurs, ou tous, dans un domaine, empêchant ainsi ces utilisateurs d'être invités à de nouvelles sessions.

Notez que cette fonction de suppression d'enregistrement d'un dispositif au profit d'un autre est un comportement voulu en SIP afin de permettre le transfert d'appel. Le dispositif de l'utilisateur doit pouvoir devenir le dispositif principal quand il vient en ligne. C'est un mécanisme très pratique pour les utilisateurs mais également pour les pirates.

3.2.9. Détournement d'appel (Call Hijacking) [4] [15]

Le Call Hijacking consiste à détourner un appel en. 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.

3.2.10. Attaque visant l'authentification SIP

Le protocole SIP peut être sensible à 2 types d'attaques, avant de voir ce type d'attaques, essayons de comprendre comment l'enregistrement SIP et le processus d'authentification a lieu. SIP utilise une authentification digest (haché) qui est un mécanisme que le protocole HTTP utilise et connu sous le nom HTTP Digest [23]. Parce que SIP est un protocole basé sur le code ASCII, les informations d'authentification sont hachées afin de ne pas les transporter en texte clair. Quand un client SIP (User Agent) veut s'authentifier avec un serveur SIP, le serveur génère et envoie un défi pour le client, ce dernier répond par une valeur dérivée du défi et un secret (mot de passe) qu'il partage. La figure suivante illustre ce processus ainsi que ses paramètres.

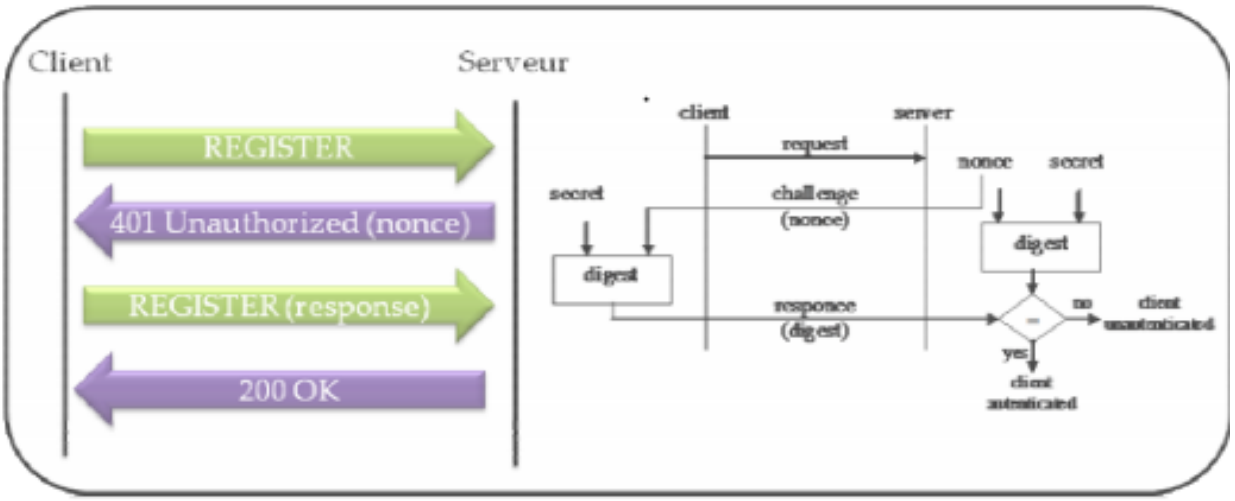

Figure 3.3 : processus d'authentification SIP entre client/server. [24]

Le défi ou le challenge généré par le serveur contient les paramètres suivants (voir figure 3.4)

```
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP 192.168.1.101;branch=z9hG4bKwpyxpiud;received=192.168.1.101;rport=5060
From: "NightRanger" <sip:200@192.168.1.104>;tag=qiqel
To: "NightRanger" <sip:200@192.168.1.104>;tag=as375e8fdb
Call-ID: vdyozxbralxnufk@BlackBox
CSeq: 961 REGISTER
User-Agent: Asterisk PBX
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY
Supported: replaces
WWW-Authenticate: Digest algorithm=MD5, realm="asterisk", nonce="421d0ea6"
Content-Length: 0
```

Figure 3.4 : paramètre du paquet « challenge ». [23]

- **Real :** utilisé pour identifier les informations d'identification au sein de message SIP, il est généralement le domaine SIP.
- **Nonce :** c'est une chaîne unique qui est généré par le serveur pour chaque demande d'enregistrement, il est fabriqué à partir d'un horodatage et une phrase secrète pour assurer qu'elle a une durée de vie limitée et pourrait ne pas être utilisé à nouveau. Une fois que le client reçoit le digest et l'utilisateur entre ses lettres de créance le client utilise le nonce pour générer une réponse digest et l'envoie au serveur.
- **Digest algorithme MD5 :** c'est l'algorithme utilisé pour la fonction de hachage.

Voici un exemple d'une réponse d'authentification haché en utilisant **MD5** envoyé par le client pour s'authentifier auprès du serveur SIP (voir figure 3.5) :

Figure 3.5: Exemple d'un hachage MD5 [23]

Donc la première attaque visant l'authentification consiste à capturer la réponse d'authentification SIP haché envoyé au serveur par le client et de la cracker ensuite. La deuxième c'est une attaque par « **brute force** » visant directement le serveur SIP.

L'attaque par **force brute** est une méthode utilisée en cryptanalyse pour trouver un mot de passe ou une clé. Il s'agit de tester une à une, toutes les combinaisons possibles. Cette méthode est en général considérée comme la plus simple concevable. Elle permet de casser tout mot de passe en un temps fini indépendamment de la protection utilisée, mais le temps augmente avec la longueur du mot de passe [25]. En théorie la complexité d'une attaque par **force brute** est une fonction exponentielle de la longueur du mot de

passe. Dans le cas d'un réseau de VOIP il s'agit d'envoyer au server SIP une multitude de requête REGISTAR ou INVITE avec un mot de passe différent à chaque tentatives en utilisant un fichier qui contient une large liste de mot de passe, si le mot de passe utilisé par le client n'est pas assez fort, un hacker peut facilement le trouver.

3.3. Les vulnérabilités de l'infrastructure

Une infrastructure VoIP est composée de téléphones IP, Gateway, serveurs (proxy, register, etc…). Chaque élément, que ce soit un système embarqué ou un serveur standard tournant sur un système d'exploitation, est accessible via le réseau comme n'importe quel ordinateur.Chacun comporte un processeur qui exécute des logiciels qui peuvent être attaqués ou employés en tant que points de lancement d'une attaque plus profonde.

3.3.1 Faiblesses dans la configuration des dispositifs de la 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 assez fort, 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.

3.3.2 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 (micro logiciel) 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 paquets contenant 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).

D'autres conséquences possibles sont :

- Des backdoors pourraient été installés.
- 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.

3.3.3. 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.

3.3.4. Les 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 (débordement de tampon). 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.

3.4. Mise en œuvre des attaques contre le réseau VoIP déployé

Après avoir étudié les protocoles de la VoIP, identifié les vulnérabilités et les attaques qui menacent les systèmes de VoIP. Nous allons simuler les techniques de hacking les plus envisageable sur un réseau de VOIP en utilisant **Backtrack 5 r3**.

Backtrack est une distribution basée sur Debian GNU / Linux, distribution destinée au piratage et à l'utilisation de tests de pénétration [27]. La version qu'on va utiliser est **BackTrack 5 r3**. Elle est basée sur Ubuntu 10.04 (Lucid) LTS et appartient à la famille Debian, téléchargeable depuis le lien suivant :
http://www.nextleveltricks.com/download-backtrack-5-r3-iso-free-64-32-bit/
BackTrack propose à ses utilisateurs un large panel d'outils (programme) en partant du simple scanner de ports jusqu'aux crackers de mot-de-passe.

3.4.1. Localisation des serveurs VoIP

Toute bonne attaque commence par une étape qui établit le profil de la cible connu sous le nom de profiling ou encore foot printing. Une empreinte englobe les informations sur la cible qui déploie le serveur VoIP et ces paramètres de sécurité.
Il existe plusieurs méthodes pour la collecte des informations en voici quelques-unes des plus utilisées notamment le **scan de réseau**.

3.4.1.1. Scan de réseau

Afin d'analyser un réseau, l'outil nécessaire pour cela est un scanner de réseau. C'est un utilitaire permettant de réaliser un audit de sécurité d'un réseau en effectuant un balayage des ports (en anglais *port scanning*) sur une machine donnée ou sur un réseau tout entier permettant de découvrir les équipements présents sur un réseau et les services qu'il offre [7]. Le scanner est souvent utilisé par les administrateurs réseau au cours de test de sécurité. Son principe de fonctionnement est de tester chaque adresse IP et chaque port TCP, UDP en envoyant des requêtes successives sur les différents ports et analyse les réponses afin de déterminer lesquels sont actifs.

Backtrack inclut de bons scanners de réseaux tel que **: NMAP, SMAP, SIPSAK, SIPVICIOUS,** qu'on va utiliser pour récolter des informations sur notre réseau VoIP.

On peut utiliser ces outils sur le terminal comme suite :

- **Nmap -sS -sU 192.168.1.1/24** : fournir les ports a exploités sur toutes les machines Connectées sur le réseau.
- **Nmap -sS -sU 192.168.1.10 :** fournit les ports exploitables sur le serveur
- **Smap 192.168.1.1/24** : détermine les périphériques SIP active sur le réseau
- **Smap -l 192.168.1.10 :** fournit des informations sur le serveur
- **Ping 192.168.1.x** : vérifier que le serveur est bien connecté dans le réseau

Avant de Commencer le scan, nous allons d'abord vérifier que le serveur Asterisk est bien connecté sur le réseau en utilisant la commande **Ping** comme l'indique la figure suivante :

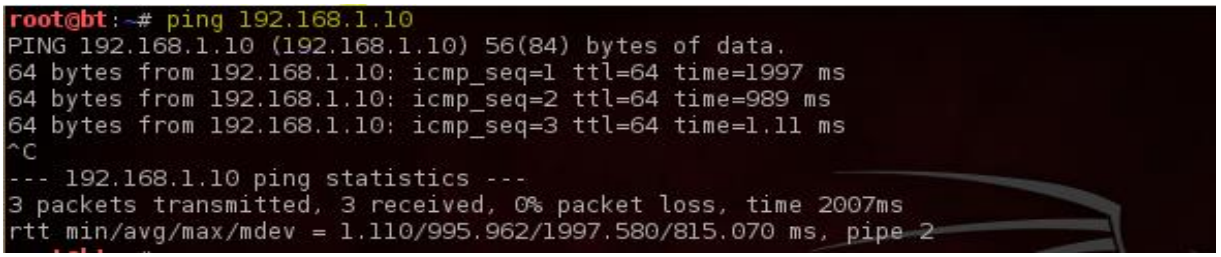

Figure 3.6 : test de joignabilité du serveur SIP

Voilà, maintenant qu'on sait que le serveur SIP est bien joignable, nous pouvons effectuer le scan en utilisant l'outil **nmap** comme indiqué dans la figure suivante **:**

Figure 3.7 : scan de ports sur le serveur SIP

L'option « **-sS** » permet de réaliser un scan TCP tandis que « **-sU** » permet de réaliser un scan UDP.

D'après les résultats de la figure 3.7, Nmap a pu déterminer les ports TCP et UDP ouverts sur le serveur SIP, ainsi que le service utilisé sur chaque port, dans la VoIP, le protocole SIP utilise le port 5060 par default et c'est ce qui nous intéresse, d'après le résultat du scan ce port est ouvert, nous pouvons exploiter cette information pour réaliser différentes attaques notamment le DOS (SYN flood ou UDP flood).

Ensuite nous avons effectués un autre scan de ports avec **nmap** pour déterminer les ports ouverts sur les postes Clients (voir figure 3.8)

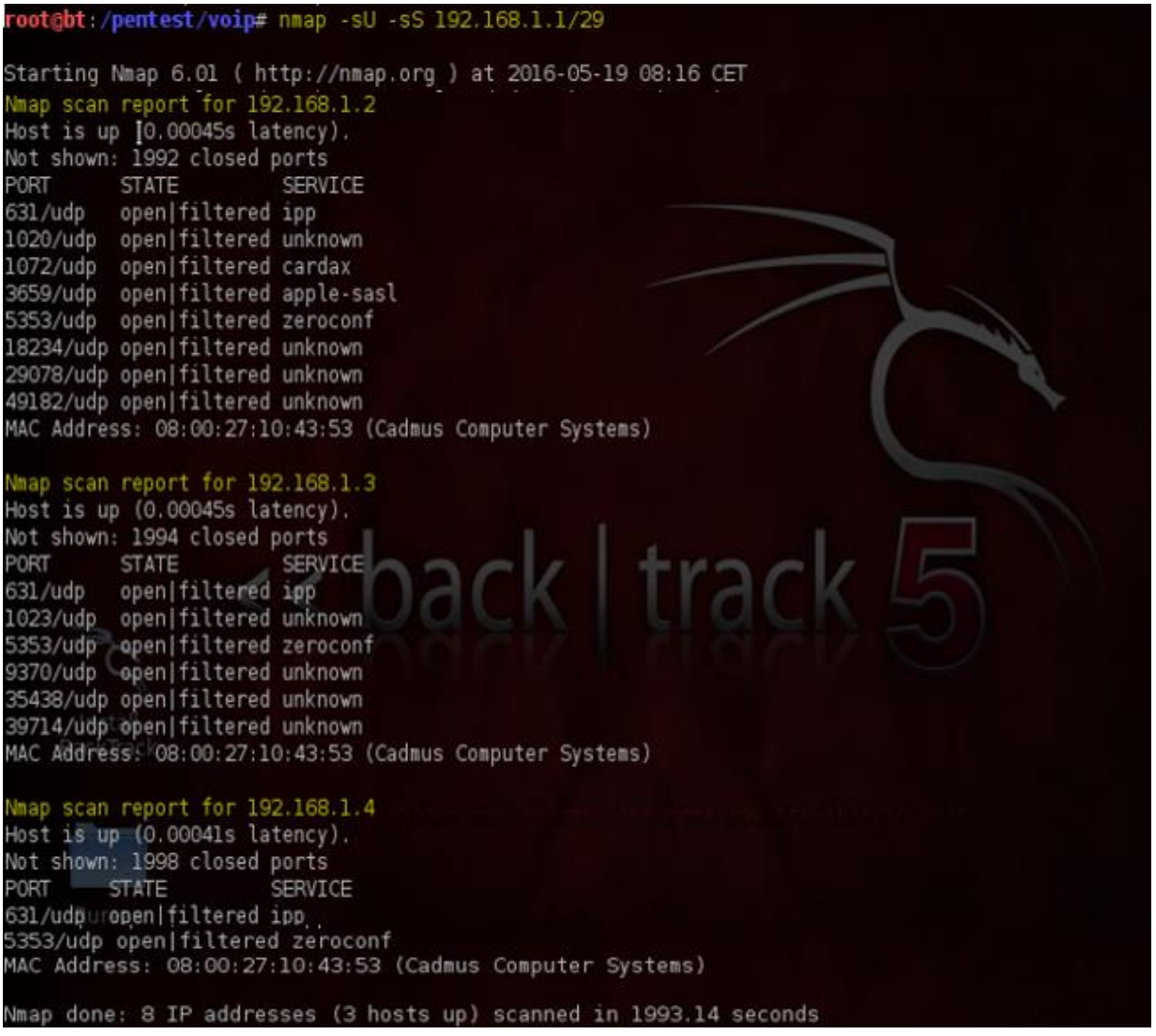

Figure 3.8 : scan de ports sur les postes clients

Nmap a pu également détectes les ports ouverts ainsi que les services utilisés sur chaque poste client de notre réseau, maintenant on connait les ports et le protocole qu'utilise chaque machine.

Backtrack inclue dans le dossier **pentest/VoIP** différents outils qui permettent d'effectuer différents types d'attaques sur un réseau de VoIP (voir figure 3.9).

Figure 3.9 : outils contenus dans le dossier VoIP

SMAP est un scanner qui permet de déterminer les périphériques SIP activé en envoyant des requêtes SIP**,** la figure suivante indique l'usage de SMAP ainsi que les périépherique SIP détécté :

Figure 3.10 : détection des périphériques SIP activé

Maintenant que nous avons identifié les hôtes SIP active, dans notre cas c'est le serveur Asterisk, nous pouvons utiliser l'option **« -l »** pour récoltés des informations sur le serveur (type et version utilisé), la figure suivante indique les résultats obtenus avec cette option.

Figure 3.11 : récolte d'information sur le serveur avec SMAP

Voilà, Maintenant nous avons déterminer le type et la version du serveur SIP qui dans notre cas est Asterisk PBX 11.22.0.

A présent nous allons déterminer combien d'extensions existe dans le fichier extensions.conf à l'aide de l'outil **SVWAR**.

SVWAR est un outil très intéressant, il fait partie de la suite d'outils **SIPVicious**.Il permet d'énumérer les extensions en utilisant une gamme d'extensions avec l'option « -e » ou bien en utilisant un fichier svwar dictionnaire (fichier.txt) avec l'option « -d ».

La figure suivante indique le contenue de la suite SIPvicious ainsi que l'usage de svwar (voir figure 3 .12).

Figure 3.12 : contenue de la suite SIPVicious et usage de svwar

Maintenant procédons à l'énumération des extensions, la figure suivante indique l'exécution de svwar en utilisant une gamme d'extensions, la méthode d'énumération par default se fait par des requetés REGISTAR, nous pouvons spécifier d'autres méthode avec l'option « **-m** ».

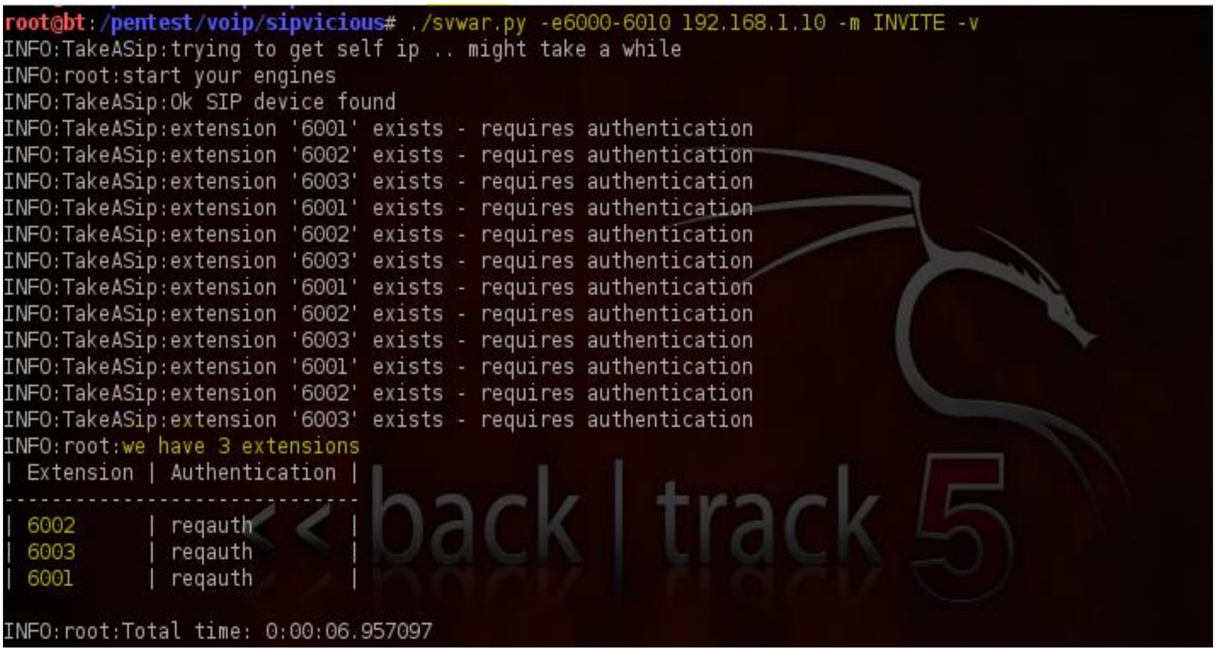

Figure 3.13 : Enumération des utilisateurs

Comme l'indique la figure, **Svwar** a pu déterminer toutes les extensions qui existe dans le fichier **extensions.conf** ainsi que le numéro de chaque extension.

Maintenant, qu'on a assez d'informations sur notre réseau de VOIP, nous pouvons réaliser Les attaques qui vont suivre.

3.4.2. Arp poisoning

Comme nous l'avons déjà expliqué auparavant **L'*ARP spoofing*** (« usurpation » ou « parodie ») ou ***ARP poisoning*** (« empoisonnement ») est une technique utilisée en informatique pour attaquer tout réseau local utilisant le protocole de résolution d'adresse ARP, les cas les plus répandus étant les réseaux Ethernet et Wi-Fi. Cette technique permet à l'attaquant de détourner des flux de communications transitant entre une machine cible et une passerelle : routeur, box, etc. L'attaquant peut ensuite écouter, modifier ou encore bloquer les paquets du réseau.

Pour réaliser cette attaque nous allons utiliser l'outil **ArpSpoofing** qui peut être invoqué juste en tapant **ArpSpoofing** sur le terminal, mais avant de pouvoir l'utiliser on doit activer l'IP forwarding avec la commande suivante :

```
#    echo 1 > /proc/sys/net/ipv4/ip_forward
```

Sans cette commande, les paquets ne pourront pas transiter à travers le système de notre machine pirate et ainsi atteindre leur cible originelle.

Maintenant que l'IP forwarding est activé, nous allons utiliser *"arpspoof"* pour envoyer continuellement des paquets ARP qui vont falsifier la table ARP de notre serveur SIP.

La ligne de commande à introduire pour exécuté arpspoof est la suivante :

```
# arpspoof   -i  interface réseau   -t   IP de la cible     IP du host (client)
```

La figure suivante indique l'exécution de de l'ArpSpoofing sur le serveur :

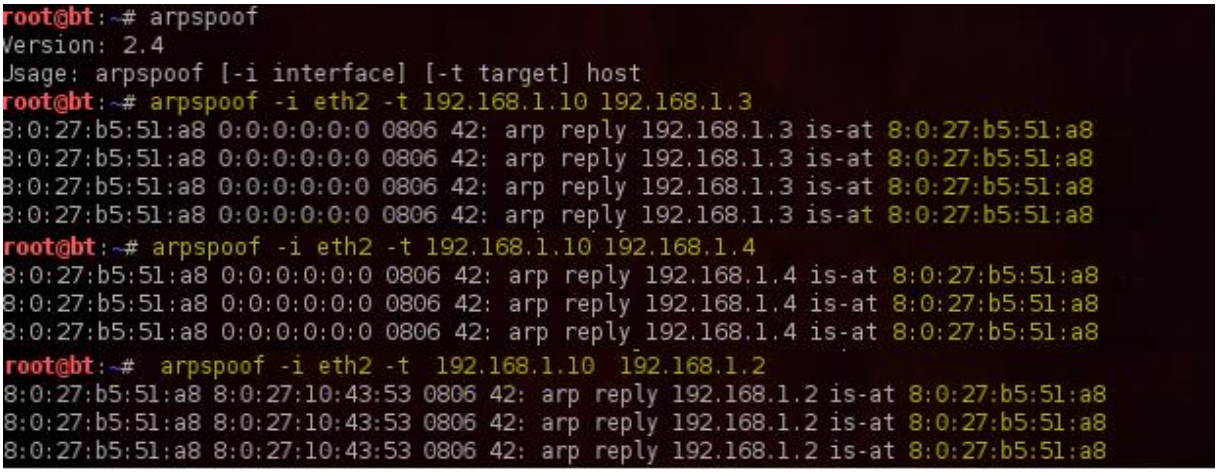

Figure 3.14 : exécution de l'ARP spoofing

La première commande peut se traduit par "fait moi passer pour **192.168.1.3** auprès de **192.168.1.10**, la deuxième par fait moi passer pour **192.168.1.4** auprès de **192.168.1.10** tandis que la troisième de traduit par fait moi passer pour **192.168.1.2** auprès de **192.168.1.10 .**

On remarque l'envoi de paquets ARP reply indiquant au serveur que les clients 6001 et 6002 et 6003 ayant les adresses IP respectives (***192.168.1.4 et 192.168.1.3 et 192.168.1.2***) ont maintenant l'adresse MAC (***8:0:27:b5:51:a8***) , il s'agit ici d'une donnée qui est fausse et falsifiée par l'action de notre machine pirate.

Maintenant comparons la table MAC du serveur SIP avant et après **ARP Spoofing :**

-Table MAC du serveur avant l'attaque d'ARP Spoofing (voir figure 3.15) :

```
root@anis-VirtualBox:~# arp -a
? (192.168.1.3) à 08:00:27:10:43:53 [ether] sur eth1
? (192.168.1.4) à 08:00:27:10:43:53 [ether] sur eth1
? (192.168.1.8) à 08:00:27:b5:51:a8 [ether] sur eth1
? (192.168.1.2) à 08:00:27:11:40:50 [ether] sur eth1
```

Figure 3.15 : Table Mac avant ARP spoof

-Table MAC du serveur après l'attaque arpspoofing (voir figure 3.16) :

```
root@anis-VirtualBox:~# arp -a
? (192.168.1.3) à 08:00:27:b5:51:a8 [ether] sur eth1
? (192.168.1.4) à 08:00:27:b5:51:a8 [ether] sur eth1
? (192.168.1.8) à 08:00:27:b5:51:a8 [ether] sur eth1
? (192.168.1.2) à 08:00:27:b5:51:a8 [ether] sur eth1
```

Figure 3.16: Table Mac après ARP spoof

On voit bien que la table ARP de notre serveur a été **falsifiée,** il va former des trames avec l'adresse IP des clients mais va en fin de compte les envoyer au pirate car il formera ses requêtes avec comme adresse MAC de destination celle de la machine pirate. Donc les données vont transiter à travers notre machine pirate qui va faire suivre les paquets au destinataire sans perturbations ou coupure car l'IP forwarding est activé.

Nous allons également utiliser arpspoof sur les postes clients pour que tout le trafic transite par notre machine pirate en utilisant les lignes de commandes suivantes :

```
#  echo 1 > /proc/sys/net/ipv4/ip_forward
#  arpspoof  -i  interface  -t  192.168.1.3  192.168.1.4
#  arpspoof  -i  interface-t  192.168.1.4  192.168.1.3
#  arpspoof  -i  interface  -t  192.168.1.3  192.168.1.2
#  arpspoof  -i  interface  -t  192.168.1.4  192.168.1.2
```

Maintenant nous allons laisser Arp Spoof s'exécuter en arrière-plan, tandis que nous effectuerons d'autre attaques

3.4.3. Capture du trafic et écoute clandestine en utilisant Wireshark

Wireshark est un logiciel libre d'analyse de protocole, utilisé dans le dépannage et l'analyse de réseaux informatiques, le développement de protocoles, l'éducation et la rétro - ingénierie, mais aussi le piratage [19]. C'est l'analyseur réseau le plus populaire du monde. Cet outil extrêmement puissant fournit des informations sur des protocoles réseaux et applicatifs à partir de données capturées sur un réseau.

Nous essayerons de capturer les paquets qui circulent dans notre réseau et spécialement les paquets SIP et RTP qui sont utilisé dans la communication VOIP pour déterminer quelques informations telles que les adresses IP, les numéros de ports, et d'autres informations qui servent au piratage (vol d'identité, déni de service, etc.). Ainsi nous pouvons écouter une communication entre deux clients en décodant les paquets RTP (écoute clandestine).

Wireshark est inclue par défaut dans **Backtrack**, on peut en faire usage en tapant simplement **Wireshark** sur notre terminal (voir la figure 3.17) :

Figure 3.17: lancement de Wireshark

3.4.3.1. Captures de trames :

Wireshark va sniffer le trafic circulant dans notre réseau local. Nous avons lancé au début, la capture des trames ensuite on a initialisé une connexion entre deux clients, « 6001 » ayant comme adresse IP 192.168.1.4 et «6002 » ayant comme adresse IP 192.168.1.3. Voici le résultat de la capture (figure) :

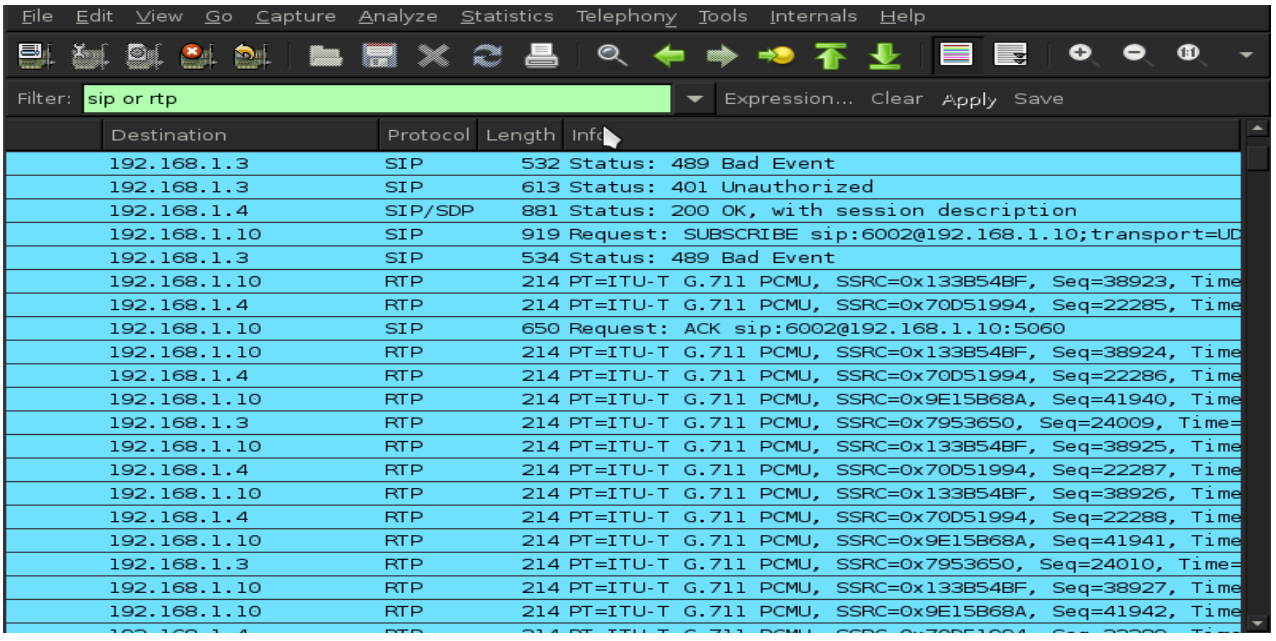

Figure 3.18 : Ecran de capture Wireshark

Comme nous pouvons le voir dans la figure (3.18), la conversation entre ces deux hôtes a été Capturée. La fenêtre principale de Wireshark comprend deux grandes parties. Dans la première partie, nous voyons les différentes étapes de connexion entre les deux clients. Dans la deuxième partie, celle la plus Intéressante (voir figure 3.19), nous pouvons lire le contenue des paquets et donc collecter des informations très indispensables pour effectuer une bonne attaque.

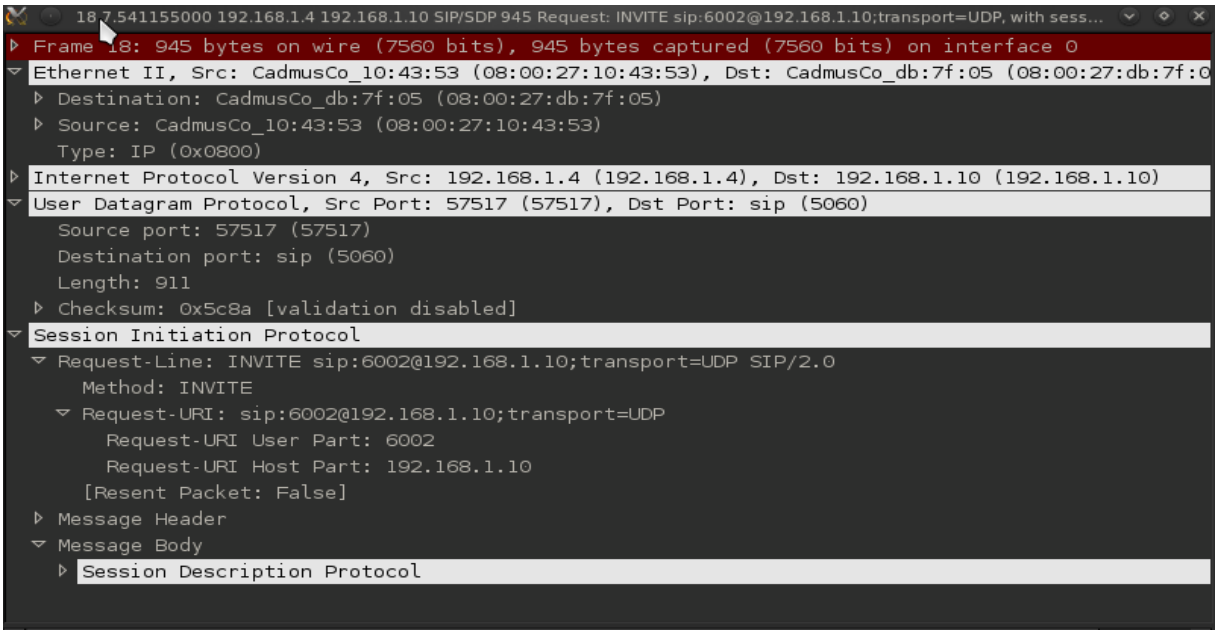

Figure 3.19 : Exemple de paquet qui contient une requête INVITE

Voilà en plus grand, dans la figure 3.19, le paquet que nous avons choisi pour examiner. Celui-ci est un paquet utilisant le protocole SIP contenant une requête INVITE.

Cette requête contient des informations indispensables dans le cas où nous voulons effectuer une attaque basée sur le protocole SIP. Par exemple dans le cas où nous voulons exécuter une attaque de type DoS en utilisant le protocole SIP, nous aurons besoin de connaitre le user agent (client). Dans cet exemple il n'est autre que le serveur Asterisk, l'adresse SIP de notre victime, son identité et d'autres paramètres.

3.4.3.2. Démonstration de l'écoute clandestine avec Wireshark :

Cette attaque consiste à capturer les trames circulant entre deux machines effectuant une conversation VoIP (voir figure 3.20), et décoder par la suite les paquets afin d'écouter la conversation effectuée.

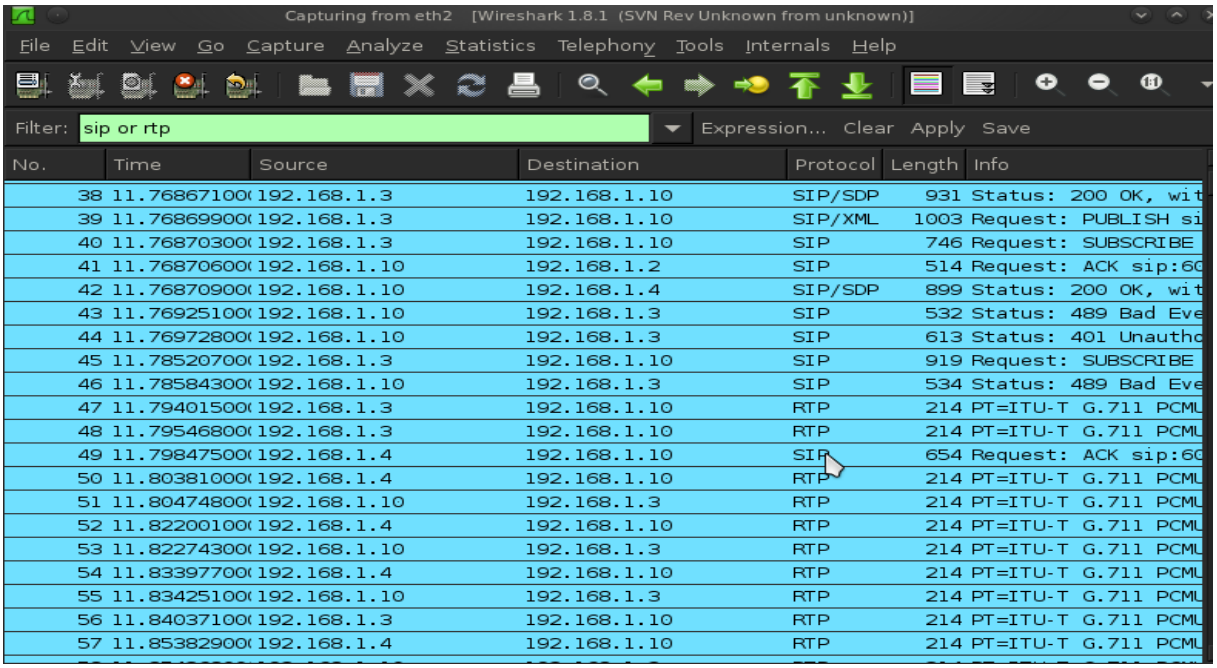

Figure 3.20: Capture d'une communication téléphonique

Le principe est le suivant. Un client nommé admin6001 ayant comme adresse IP 192.168.1.4 va appeler le client nommé admin6002 ayant comme adresse 192.168.1.3. Il faut savoir que ces deux clients utilisent le serveur Asterisk qui a été préalablement configuré pour effectuer leurs appels. Avant d'effectuer cet appel, il faut tout d'abord activer Wireshark afin de sniffer le trafic depuis la machine pirate qui n'est pas autorisé à passer des appels à travers le serveur Asterisk puisqu'elle n'est pas configurée dans les fichiers de ce dernier. Toutes ces machines sont installées sous le même réseau. Durant la capture nous pouvons voir les différentes phases d'appel, la signalisation et le transport des paquets.

A la fin de l'appel, nous aurions sniffé tous les paquets dont nous aurions besoin pour l'écoute clandestine, les paquets les plus importants sont ceux basés sur le protocole RTP vu qu'ils contiennent les conversations audio entre les deux clients comme l'indiqué la figure 3.20.

Après avoir identifié les paquets RTP, nous allons maintenant procéder au décodage de l'appel. Dans le menu de Wireshark, nous cliquons sur le bouton « Téléphony », puis ensuite le bouton « VoIP Calls » comme l'indique la figure suivante.

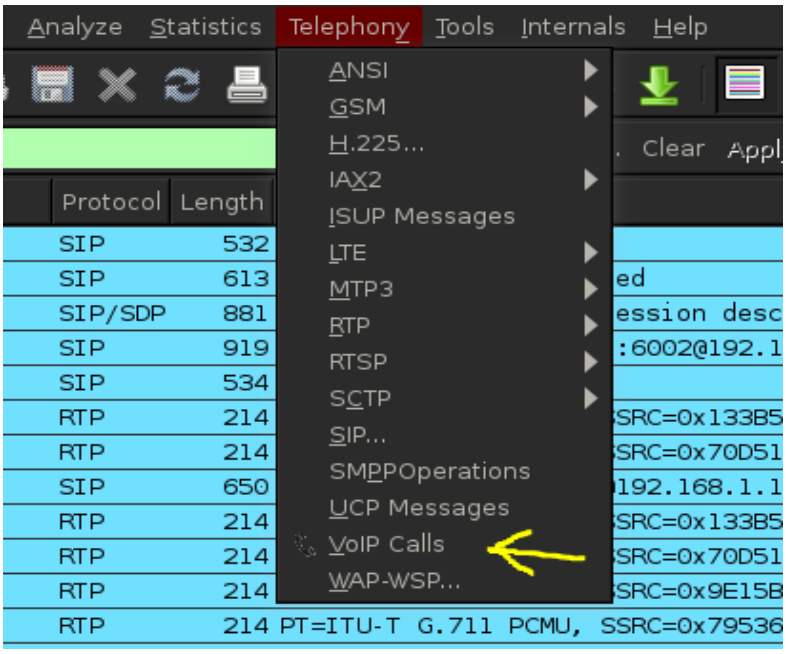

Figure 3.21: Décodage : Bouton VoIP Calls

Une deuxième fenêtre s'ouvre (voir figure 3.22) contenant les communications dans les deux sens, du client 192.168.1.4 vers 192.168.1.3, et inversement

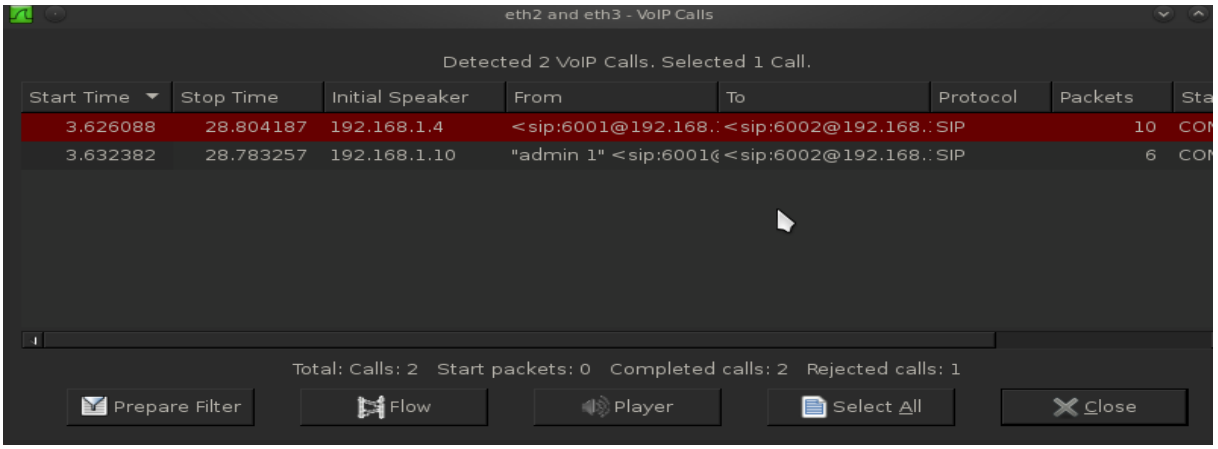

Figure 3.22 : communication téléphonique détectes

Nous choisissons une des communications détectées et nous cliquons sur le bouton « Player », une fenêtre « RTP Player » s'ouvre pour le décodage (figure 3.23). nous cliquons sur « Décode » pour que l'opération commence.

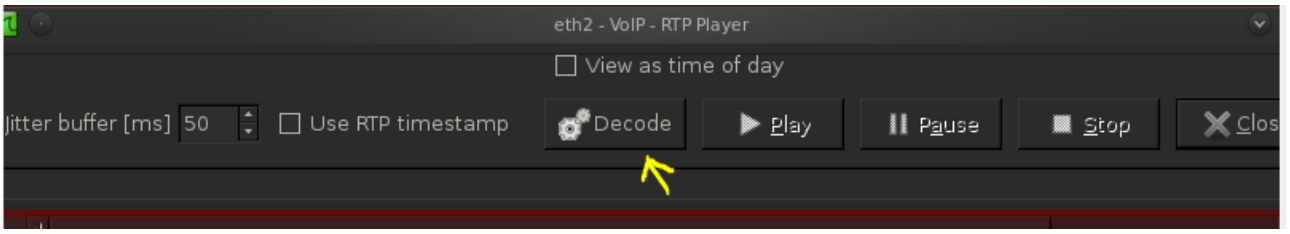

Figure 3.23 : Fenêtre RTP Player

Maintenant que le décodage des paquets RTP a abouti, nous pouvons aussi voir sur la figure 25 que le son est décodé et qu'il est prêt à être écouté.

Pour l'écoute, il faut choisir le parcours de la communication, plus précisément, il faut choisir la direction de la communication. Nous avons choisi ici d'écouter la communication qui se dirige de l'adresse IP192.168.1.4 vers l'adresse IP 192.168.1.10. La communication est de durée 16,27s.

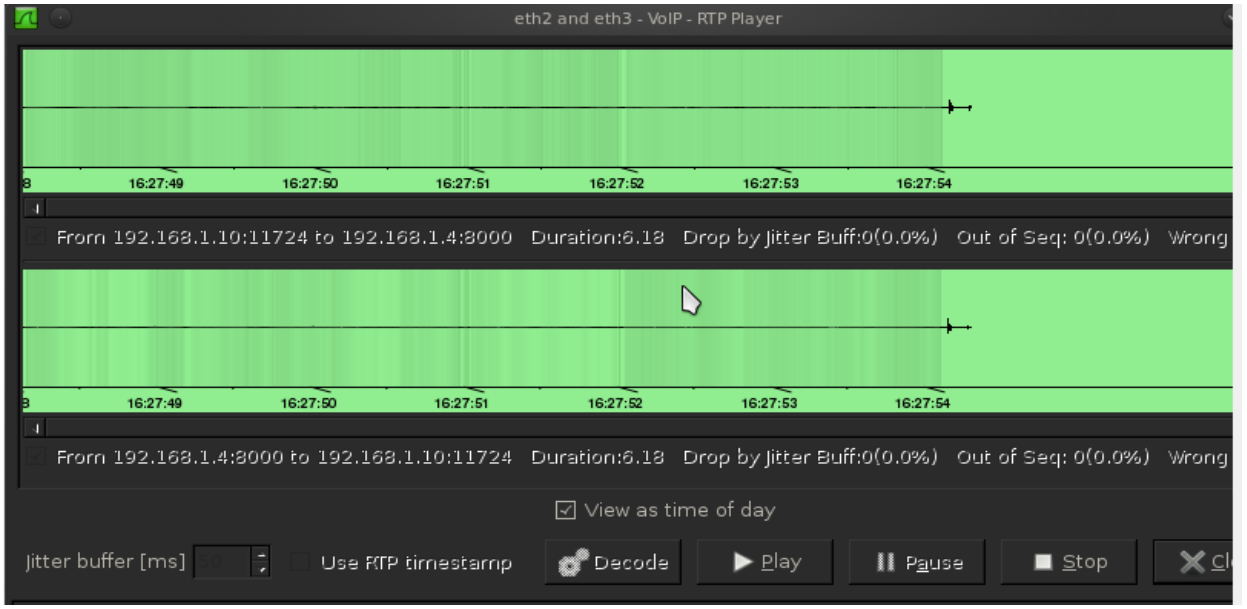

Figure 3.24 : Communication téléphonique décodé

3.4.4. Déni de services

Un déni de service sur le service VoIP peut le rendre inutilisable en provoquant un endommagement intentionnel au réseau et la disponibilité des systèmes VoIP. Cette attaque peut se produire sur deux niveaux, attaques de dos standard ou attaques spécifiques visant un certain protocole. En général, nous ferons parvenir des tonnes de données en inondant la cible et l'inciter à consommer toutes ses ressources ou un protocole spécifique afin de la mettre hors service. Plusieurs outils sont disponibles sur backtrack, nous allons utiliser **Hping3** et **Inviteflood**

3.4.4.1. Hping3

Hping3 est un outil capable de généré des paquets TCP / IP. Il est utilisé par les experts en sécurité comme un scanner de réseaux mais aussi dans les attaques de types dos (déni de service) bien que hping3 offre une grande variété d'options, nous allons utiliser le peu d'entre eux y compris :

--flood : Envoyés des paquets aussi vite que possible, sans prendre soin de montrer les paquets.

-I: Interface à utiliser (utilisé si on est connecté à plusieurs interfaces)

-1: Mode ICMP

-2: Mode UDP

-a: Fake Host Name(utiliser une fausse adresse IP)

-p: port de destination

-S: pour envoyer des paquets SYN

Nous allons exploiter les informations récoltées préalablement par le scan et effectuer une attaque DOS de type UDP flood sur le proxy SIP, la syntaxe de la commande à exécuter sur le terminal est comme suite :

Figure 3.25 : UDP flood en utilisant Hping3

L'attaque UDP flood à débuter, Hping3 a commencé à envoyer un grand nombre de paquets UDP au serveur SIP à travers le port 5060, l'option **« -a »** permet de masquer la source des paquets UDP et rester anonyme.

3.4.4.2. Invite Flood

Cet outil peut être utilisé pour inonder une cible avec des requêtes INVITE, il peut être utilisé pour cibler des passerelles SIP / proxies et des téléphones SIP. Cette fois ci nous allons cibler l'utilisateur 6001.L'utilisation de cet outil se fait comme suite :

./inviteflood interface réseau extension de la cible domaine de la cible IP cible nombre de paquets à envoyer

Exécution du DOS (voir figure 3.26) :

Figure 3.26 : Exécution de l'inviteflood

La figure indique que l'attaque a été bien mené et que la destination est inondée avec 1000000 de requêtes INVITE.

Au moment de l'exécution de l'inviteflood, on a essayé d'effectuer un appel de l'utilisateur 6002 vers l'utilisateur 6001.

La figure 3.27 indique le message affiché par le softphone de l'utilisateur 6002

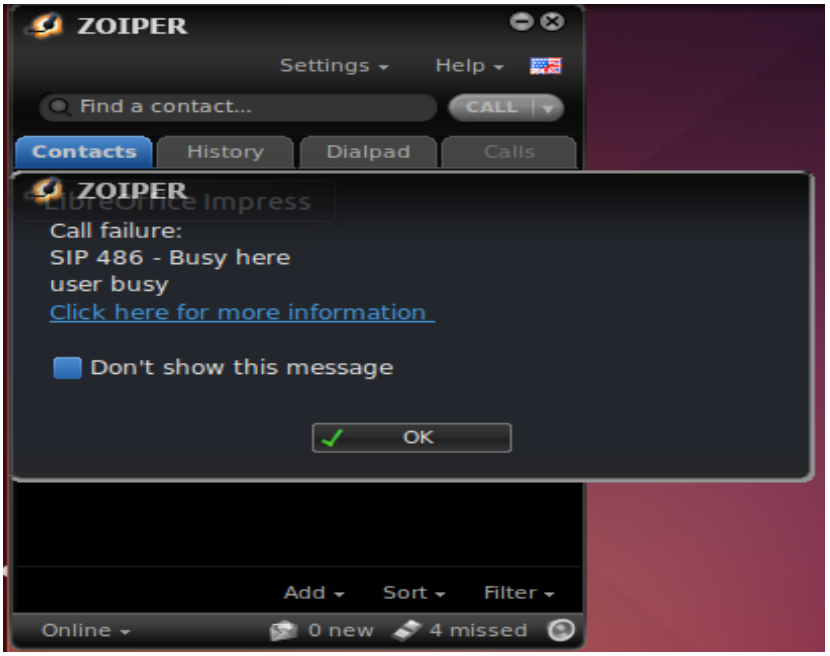

Figure 3.27 : message d'erreur : échec de l'appel

Un message « call faillure » apparait sur le soft phone de l'utilisateur 6002 indiquant un échec d'appel, impossible d'établir une connexion avec l'utilisateur 6001, le service est occupé car il est inondé par un flux de donnée important.

3.4.5. Attaque sur l'authentification SIP

Comme nous l'avons vue auparavant Le protocole SIP peut être sensible à 2 types d'attaques d'authentification, la première consiste à capturer la réponse d'authentification haché envoyé du client vers le serveur et du cracker ensuite, la deuxième consiste à attaquer le serveur directement par force brute en envoyant une multitude de requêtes registar ou invite avec des mots de passe différents à chaque fois.

Backtrack fournit une excellente suite d'outils appelée **SIPCrack** qui va nous permettre de réaliser ces attaques. La réalisation de la première attaque visant l'authentification va se faire en deux parties, la première consiste à capturer la réponse d'authentification SIP haché en utilisant **sipdump** et la deuxième étape consiste cracker avec **sipcrack**.

3.4.5.1 Capture de l'authentification SIP en utilisant Sipdump

SIPDump fait partie de l'outil **SIPCrack**, il permet d'effectuer une capture en temps réel de l'authentification SIP digest (haché) ou il peut vider une session précédemment capturée à partir d'un fichier PPCE. **Sipdump** s'utilise comme suite :

./sipdump **-i** interface réseau chemin du fichier ou enregistré les captures

La figure suivante indique l'exécution de sipdump :

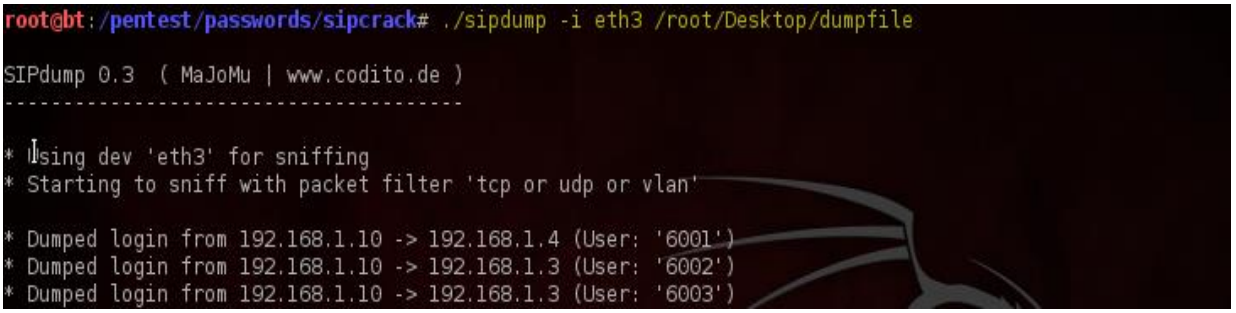

Figure 3.28 : capture de la réponse d'authentification en utilisant SIPDUMP

Dès qu'un client s'enregistre sur le serveur ou bien effectue un appel sipdump va capturer La réponse d'authentification SIP envoyé par client au serveur et va l'enregistrer dans un fichier que nous avons nommé **dumpfile**, voici le contenu de ce fichier (voir figure 3.29) :

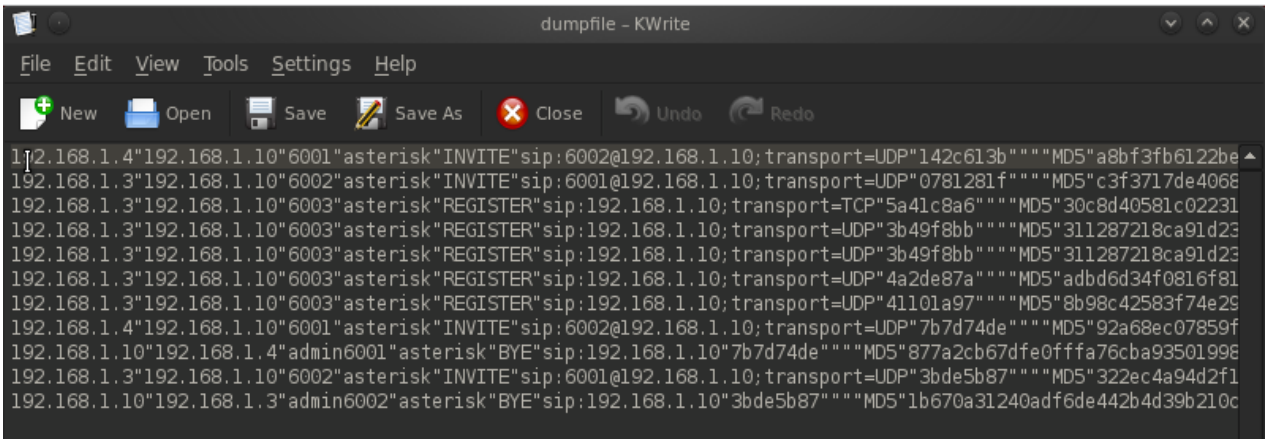

Figure 3.29 : capture de l'authentification SIP digest

Maintenant nous allons cracker l'authentification SIP digest capturé.

3.4.5.2. Crack de la réponse d'authentification SIP :

Pour cela nous allons utiliser **sipcrack** et un dictionnaire sous forme d'un fichier.txt qui se trouve sous **pentest/passwords/wordlists/rockyou.txt** et qui contient une large liste de mot de passe, c'est une attaque MD5 Bruter en quelque sorte qui consiste à casser le hache MD5. Sipcrack va générer un hache MD5 pour chaque mot de passe du fichier **rockyou.txt** et les comparer un par un au hache de la réponse SIP capturé qui se trouve dans le fichier que nous avons nommé dumpfile jusqu'à ce qu'il trouve un hache similaire à celui de la réponse SIP.

L'utilisation de sipcrack se fait comme suite :

# cd **/pentest/passwords/sipcrack/**	accès au dossier sipcrack
# **.**/sipcrack chemin du fichier **dumpfile** -w chemin du fichier **rockyou.txt**	

L'exécution ainsi que les résultats de cette commande sont indiquées dans la figure 3.30

```
root@bt:/pentest/passwords/sipcrack# ./sipcrack /root/Desktop/dumpfile -w /pentest/passwords/wordlists/rockyou.
txt

SIPcrack 0.3  ( MaJoMu | www.codito.de )
----------------------------------------

* Found Accounts:

Num     Server          Client          User    Hash|Password

1       192.168.1.4     192.168.1.10    6001    a8bf3fb6122be2f9d305f302508e29b7
2       192.168.1.3     192.168.1.10    6002    c3f3717de4068837d4499ac82e53f49c
3       192.168.1.3     192.168.1.10    6003    30c8d40581c02231e2724f2a1faa032c
4       192.168.1.3     192.168.1.10    6003    311287218ca91d2399fc01e5c3ed0b7d
5       192.168.1.3     192.168.1.10    6003    311287218ca91d2399fc01e5c3ed0b7d
6       192.168.1.3     192.168.1.10    6003    adbd6d34f0816f81bd418dc50e334e2b
7       192.168.1.3     192.168.1.10    6003    8b98c42583f74e29ee8000e37b71e85b
8       192.168.1.4     192.168.1.10    6001    92a68ec07859f44223cc3704012c4333
9       192.168.1.10    192.168.1.4     admin6001       877a2cb67dfe0fffa76cba9350199807
10      192.168.1.3     192.168.1.10    6002    322ec4a94d2f158edc2f6c95ce7e2a72
11      192.168.1.10    192.168.1.3     admin6002       1b670a31240adf6de442b4d39b210c5b

* Select which entry to crack (1 - 11): 1
```

Figure 3.30 : exécution de sipcrack

Après avoir éxécuté la commande,on choisit le hash a cracker dans la liste qui s'affiche ,on achoisit de cracker celui de l'utilisateur 6001 ,voici le résultat obtenue (voir figure 3.31) :

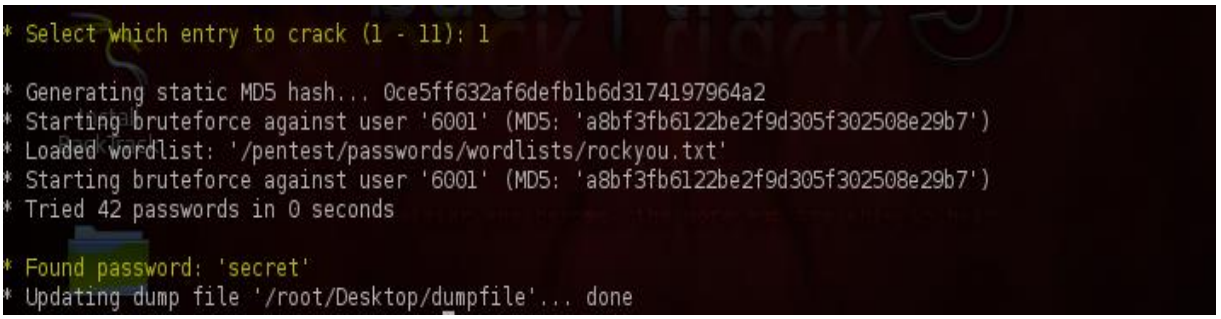

Figure 3.31 : crack de l'authentification SIP haché

On est parvenue à cracker la réponse d'authentification et à trouver le mot de passe, l'utilisateur 6001 utilise le mot **« secret »** comme mot de passe pour s'authentifier auprès du serveur Asterisk. Maintenant qu'on a toutes les informations qu'il faut sur cet utilisateur, on peut se faire passer pour lui et effectuer des appels en son nom.

3.4.5.3. Attaque par force brute

Comme nous l'avons expliqué auparavant, l'attaque par brute force est une méthode utilisée par les pirates et en cryptanalyse pour découvrir un mot de passe ou une clef. Il s'agit d'essayer toutes les combinaisons possibles, dans notre cas l'attaque consiste à trouver le mot de passe utilisé par le client pour s'authentifié auprès du serveur SIP.

Nous allons utiliser **SVCRACK** qui fait partie de la suite d'outil **SIPVICIOUS** pour effectuer cette attaque et un dictionnaire « rockyou.txt » se trouvant dans le dossier **pentest /passwords/wordlists**/ qui contient une large liste de mot passe, cette fois ci nous allons essayer de trouver le mot de passe du compte utilisateur 6002 (admin6002), l'usage de cet outil se fait sur terminal comme suite :

cd /pentest/voip/sipvicious (accès à la suite sipvicious)
./svcrack.py **-u** extensions client **-d** le chemin du fichier.txt **IP** du serveur (cible)

La figuré suivante indique l'exécution de l'attaque par brute force :

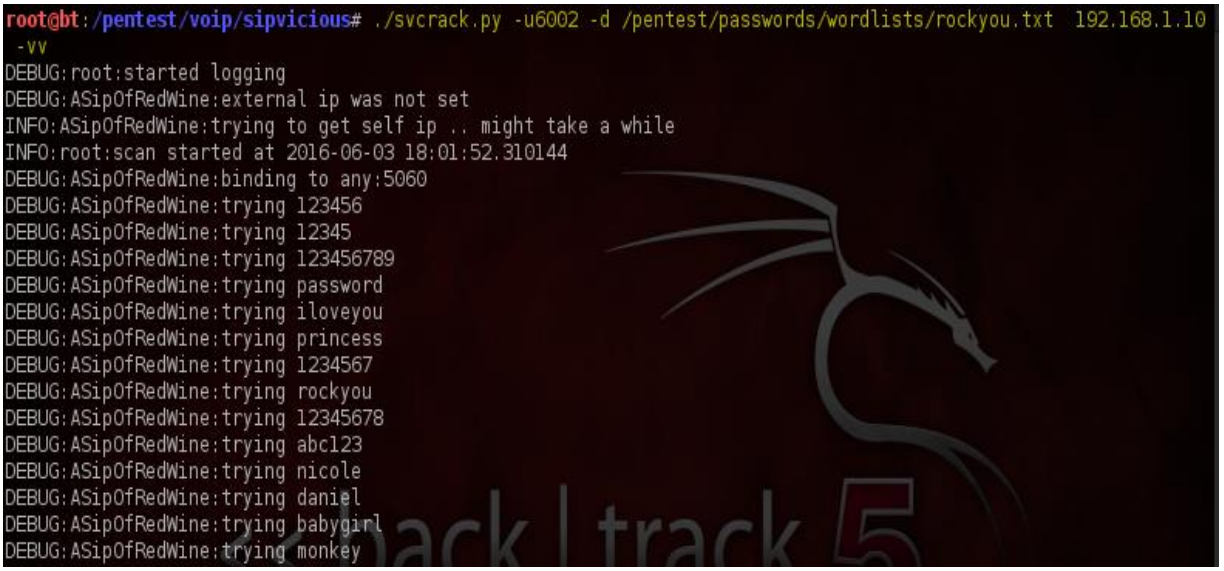

Figure 3.32 : attaque brute force en utilisant SVCRACK

SVCRACK va tester une par une les différentes combinaisons qui existe dans le fichier rockyou.txt jusqu'à ce qu'il parvienne à trouver la combinaison correcte qui lui permet de s'authentifier auprès du serveur Asterisk, la figure suivante indique que le brute force a pu déterminer le mot de passe du compte utilisateur 6002

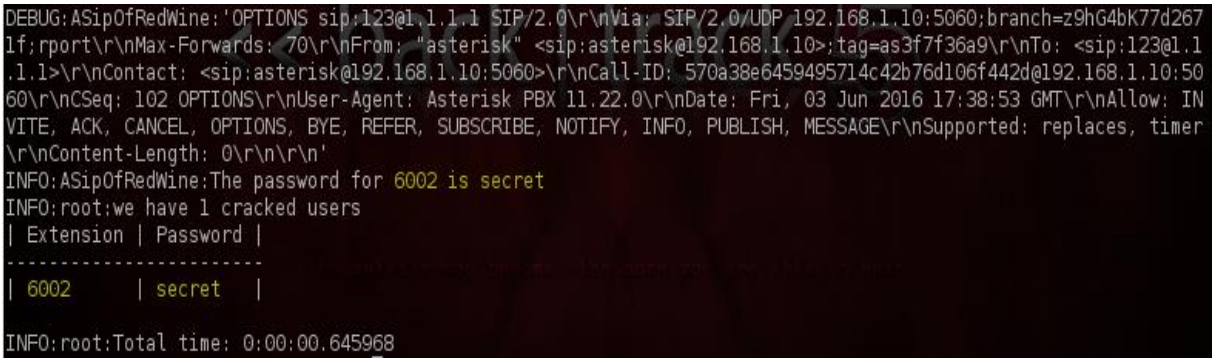

Figure 3.33 : réussite du de l'attaque par brute force

D'après la figure, L'utilisateur 6002 utilise « **secret** » comme mot de passe pour s'authentifier auprès du serveur Asterisk. Maintenant qu'on a toutes les informations nécessaires sur ce client (numéro d'extension et mot de passe), on pourra effectuer des appels en son nom ce qui représente un réel danger surtout dans le cadre d'une entreprise ou ce client détient une certaine notoriété et privilège.

3.5 Discussion

Dans ce chapitre nous avons présenté les attaques qui menacent un réseau de VoIP et nous avons détaillé quelques-unes ainsi que les vulnérabilités d'une telle infrastructure. Ensuite nous avons réalisé quelques techniques utilisées pour pirater un réseau de VoIP en se basant sur les attaques présentées.

La voix sur IP devient jour après jour plus ciblée. Il existe plusieurs autres attaques qui menace la sécurité de la VoIP, les attaques cités et réalisés dans ce chapitre sont les plus fameuses et courantes dans les réseaux VoIP.

Mais en suivant certaines bonnes pratiques qu'on va présenter et installer dans le chapitre qui suit, on peut déployer un réseau de VoIP bien sécurisé.

Chapitre 4

Sécurisation et bonnes Pratiques

4.1. Préambule :

Après avoir étudié les protocoles de la VOIP, identifié les vulnérabilités et simulé des attaques sur le réseau de VOIP déployé, nous allons nous intéresser aux solutions, mécanismes et configurations à mettre en place dans le but de sécuriser la solution VOIP basée sur le serveur asterisk contre les différentes techniques de hacking que nous avons simulées précédemment.

4.2. Implémentation des bonnes pratiques de sécurisation :

La sécurité de la VoIP n'est pas une nécessité mais plutôt une obligation. Pour se protéger contre les attaques réalisées et même d'autres attaques similaires, nous avons choisi un ensemble de solutions qui peuvent aider à réduire voir bloquer certaines attaques ainsi rendre la tache encore plus difficile pour le hacker.
Puisque la majorité des attaques cible toujours les conversations et le serveur SIP dans un réseau de VoIP, nous allons donc proposer un ensemble de solutions pour les sécuriser.

4.2.1 Exécuter Asterisk sous un utilisateur non privilégié :

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

Tous d'abord il faut arrêter le service Asterisk :

```
# /etc/init.d/asterisk stop #> Shutting down asterisk:                    OK
```

Ensuite il faut créer un utilisateur depuis lequel Asterisk va démarrer. Nous avons choisi le nom du groupe asterisk et comme nom d'utilisateur asterisk.

```
# /usr/sbin/groupadd asterisk
#/usr/sbin/useradd -d /var/lib/asterisk -g asterisk asterisk
```

Ensuite, il faut attribuer les droits d'accès vu qu'Asterisk en a besoin. Les fichiers dans le répertoire /var/spool/asterisk doivent être la propriété de l'utilisateur Asterisk et accessibles en écriture. Les commandes suivantes sont celles qui ont été exécutées pour le répertoire /var/lib/asterisk. Les mêmes commandes sont appliquées pour les répertoires suivant : /var/log/asterisk, /var/run/asterisk, /var/spool/asterisk, /usr/lib/asterisk et le dossier /dev/zap.

```
# chown --recursive asterisk:asterisk /var/lib/asterisk
# chmod --recursive u=rwX,g=rX, /var/lib/asterisk
```

(L'option --recursive permet de modifier les permissions d'un répertoire et de ses sous-répertoires. Ainsi grâce à la commande chown le propriétaire du répertoire /var/lib/asterisk et ses sous-répertoires est devenue asterisk)

Asterisk a besoins de lire le répertoire /etc/asterisk et son contenu afin de le modifier.

```
# chown --recursive root:asterisk /etc/asterisk
# chmod --recursive u=rwX,g=rX, /etc/asterisk
```

Ensuite il faut changer le répertoire d'Asterisk afin qu'il puisse démarrer du nouveau chemin créé :

```
# cp /etc/asterisk/asterisk.conf /etc/asterisk/asterisk.conf.org
# vi /etc/asterisk/asterisk.conf
```

Modifier le chemin en changeant la ligne suivante de :

```
astrundir => /var/run à astrundir => /var/run/asterisk
```

Ensuite il faut activer le nouveau groupe que nous venons de créer et le nouvel utilisateur aussi :

```
# cp  /etc/init.d/asterisk /etc/init.d/asterisk.org

# vi /etc/init.d/asterisk
```

Changer la ligne suivante afin d'informer Asterisk de son nouvel utilisateur :

```
#AST_USER="asterisk"

#AST_GROUP="asterisk"

à

AST_USER="asterisk"

AST_GROUP="asterisk"
```

Maintenant il faut redémarrer Asterisk avec les nouveaux paramètres, à savoir le groupe Asterisk et le nom d'utilisateur Asterisk

```
#  /etc/init.d/asterisk restart

#  asterisk -U asterisk -G asterisk
```

4.2.2 Implémentation d'un système de détection d'intrusion :

Un système de détection d'intrusion (ou IDS : *Intrusion Détection System*) est un mécanisme destiné à repérer des activités anormales ou suspectes sur la cible analysée (un réseau ou un hôte).

Il permet ainsi d'avoir une connaissance sur les tentatives réussies comme échouées des intrusions. Il existe trois grandes familles distinctes d'IDS :

- Les NIDS (*Network Based Intrusion Détection System*), qui surveillent l'état de la sécurité au niveau du réseau.

- Les HIDS (*Host Based Intrusion Détection System*), qui surveillent l'état de la sécurité au niveau des hôtes.
- Les IDS hybrides, qui utilisent les NIDS et HIDS pour avoir des alertes plus pertinentes.

Dans notre cas, nous allons utiliser **SNORT** qui est un NIDS open source.

4.2.2.1. Définition de SNORT [10] [26]

SNORT est un système de détection d'intrusions (ou NIDS) open source publié sous licence GNU/GPL à l'origine écrit par Martin Roesch, il appartient actuellement à Sourcefire. Des versions commerciales intégrant du matériel et des services de supports sont vendus par Sourcefire, son code source est accessible et modifiable depuis l'URL : https://www.snort.org/. Snort est un des plus actifs NIDS Open Source et possède une communauté importante contribuant à son succès.

Snort est capable d'effectuer en temps réel des analyses de trafic et de logger les paquets sur un réseau IP. Il peut effectuer des analyses de protocole, recherche/correspondance de contenu et peut être utilisé pour détecter une grande variété d'attaques et de sondes comme des dépassements de buffers, scans, attaques sur des CGI (Common Gateway Interface), sondes SMB (Server Message Block), essai d'OS fingerprintings et bien plus. Snort pour effectuer ces analyses se fonde sur des règles. Celles-ci sont écrites par Sourcefire ou bien fournies par la communauté. Snort est fourni avec certaines règles de base mais cependant, comme tout logiciel, Snort n'est pas infaillible et demande donc une mise à jour régulière. Le choix de Snort, en particulier, est justifié par le fait que nous trouvons que cette solution peut être adaptée pour un réseau quel que soit sa taille (TPE, PME …), mais aussi la disponibilité de nombreuses règles dans la communauté de Snort et des projets comme Oinkmaster qui gèrent la mise à jour des signatures et fournissent un nombre important de règles optimisées et efficaces pour la détection des intrusions.

4.2.2.2. Emplacement de Snort

Dans le cas réelle la machine qui va héberger snort se positionnera entre le serveur et les postes clients (voir figure 4.1) Ces derniers seront connecté à une interface de l'IDS à

travers le switch et le proxy à une autre interface ,l'IDS va jouer le role d'un pont entre eux,la machine qui héberge Snort va étre configuré en bridge , de cette façon tout les message en directions du serveur passeront d'abord par Snort qui va les éxaminer et détécter d'eventuelles menaces.

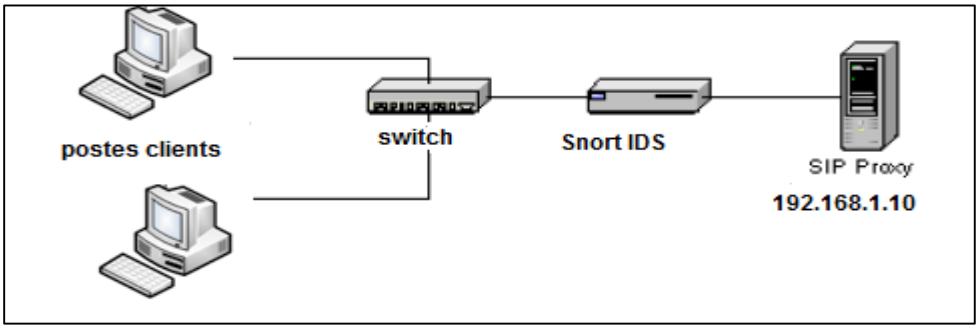

Figure 4.1 : emplacement de SNORT IDS

Mais puisque nos testes ont été effectuer dans un environnement virtuelle , nous avons directement installer Snort sur notre serveur SIP .

4.2.2.3. Installation des dépendances nécessaire :

Snort a quatre principales dépendances qu'il faut installer :

- Pcap (libpcap-dev) disponible dans ubuntu repository
- PCRE (libpcre3-dev) disponble dans Ubuntu repository
- Libdnet (libdumbnet-dev) disponible dansUbuntu repository
- DAQ (http://www.snort.org/downloads/) à compiler depuis la source

Nous devons d'abord installer tous les outils nécessaires à la construction de logiciels. Le paquet build-essentials fait cela pour nous :

```
# sudo apt-get install -y build-essential
```

Une fois fait, on installe les dépendances nécessaires à la compilation de snort :

```
# sudo apt-get install -y libpcap-dev libpcre3-dev libdumbnet-dev libssl-dev bison flex
```

Ensuite on doit créer un dossier pour snort en utilisant la commande mkdir :

```
# sudo mkdir /etc/snort
```

On accéde au dossier créé, ensuite on télécharge DAQ (Data Acquisition Library*)* qui est indispensable pour Snort puis on l'installe avec les commandes suivantes :

```
# cd /etc/snort
# wget https://www.snort.org/downloads/snort/daq-2.0.6.tar.gz
# tar -xvzf daq-2.0.6.tar.gz
# cd daq-2.0.6
# ./configure && make && sudo make install
```

Maintenant tout est prêt pour télécharger et installer snort :

```
# cd /etc/snort
# wget https://snort.org/downloads/snort/snort-2.9.8.0.tar.gz
# tar -xvzf snort-2.9.8.0.tar.gz
# cd snort-2.9.8.0
# ./configure --enable-sourcefire && make && sudo make install
```

On exécute la commande suivante pour mettre à jour les bibliothèques partagées (On obtient une erreur en essayant d'exécuter Snort sans cette étape) :

```
# sudo ldconfig
```

Maintenant testons snort avec la commande **snort -v**, si tout est bien installé, la figure 4.2 va s'afficher :

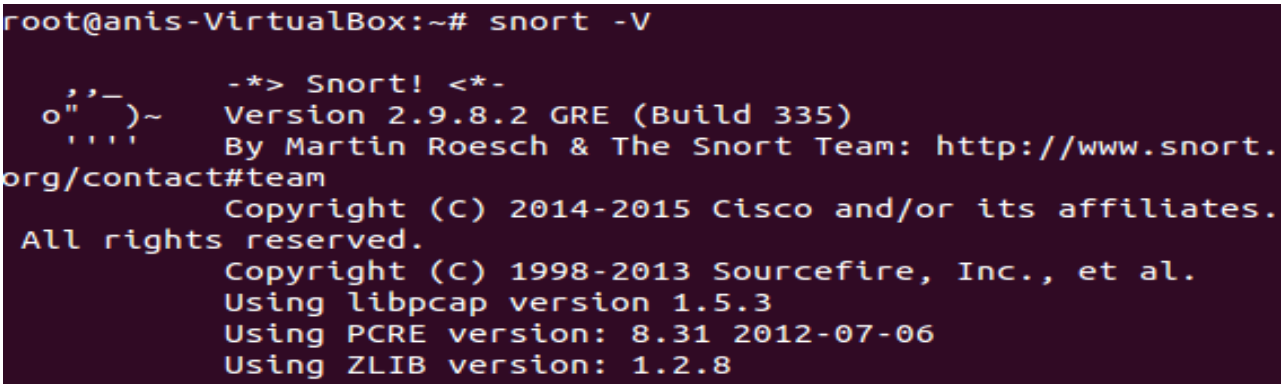

Figure 4.2 : installation de SNORT réussi

4.2.2.4. Configuration de snort en mode NIDS :

Snort dispose de plusieurs modes de fonctionnements qui sont les suivants :

- **Mode "écoute" :** Ce mode permet de lancer snort en mode “sniffer” et permet d'observer les paquets que l'IDS perçoit ("snort -v")
- **Mode "log de paquets " :** Le log de paquet permet l'archivage des paquets circulant sur le réseau de l'IDS. Il permet, grâce à ses arguments des opérations intéressantes permettant de limiter les logs à certains critères, comme une plage d'adresse IP (ex : "snort -l .../log/snort -h 192.168.0.0/24")
- **Mode "détection d’intrusion" :** Le mode NIDS permet à snort d'adopter un comportement particulier en cas de détection d'une (succession) de chaînes de caractères dans les paquets interceptés ; selon les règles définies dans les fichiers d'extension ".rules" du répertoire /rules ("snort -A full -d -l ../log -c $SNORTPATH/snort.conf").

Étant donné que nous voulons surveiller le serveur Asterisk, nous allons le configurer en mode NIDS, mais d’abord ne voulons pas que Snort fonctionne en tant que root, nous avons besoin de créer un compte non privilégié et un groupe pour le processus daemon exécuté sous (snort : snort). Nous allons également créer un certain nombre de fichiers et de et répertoires requis pour Snort, et définir des autorisations sur ceux-là. Snort aura les répertoires suivants : **Configuration** et **Rule,** dans / etc / snort. Les alertes seront écrites dans / var / log / snort. Les règles compilé (.so règles) seront stockées dans / Usr / local / lib / snort DYNAMICRULES

Création du compte snort et group :

```
#  sudo groupadd snort
#  sudo useradd snort -r -s /sbin/nologin -c SNORT_IDS -g snort
```

Création des répertoires Snort :

```
#  sudo mkdir /etc/snort
#  sudo mkdir /etc/snort/rules
#  sudo mkdir /etc/snort/rules/iplists
#  sudo mkdir /etc/snort/preproc_rules
#  sudo mkdir /usr/local/lib/snort_dynamicrules
#  sudo mkdir /etc/snort/so_rules
```

Création des fichiers qui stockent quelque règles et listes IP :

```
#  sudo touch /etc/snort/rules/iplists/black_list.rules
#  sudo touch /etc/snort/rules/iplists/white_list.rules
#  sudo touch /etc/snort/rules/local.rules
#  sudo touch /etc/snort/sid-msg.map
```

Changer de propriété sur les dossiers :

```
#  sudo chown -R snort:snort /etc/snort
#  sudo chown -R snort:snort /var/log/snort
#  sudo chown -R snort:snort /usr/local/lib/snort_dynamicrules
```

Nous allons maintenant modifier manuellement certains paramètres dans le ficher snort.conf :

```
#   sudo vi /etc/snort/snort.conf                    accès au fichier snort.conf
```

Puis on modifie les lignes suivantes pour répondre à notre environnement :

Ligne 45, HOME_NET doit correspondre à notre réseau interne. Dans l'exemple ci-dessous notre HOME_NET est 192.168.1.0 avec un masque de sous-réseau 24 bits (255.255.255.0)

```
ipvar  HOME_NET  192.168.1.0/24
```

On ajoute les chemins suivants dans **snort.conf**, en commençant à la ligne 104 :

```
var  RULE_PATH /etc/snort/rules
var  SO_RULE_PATH /etc/snort/so_rules
var  PREPROC_RULE_PATH /etc/snort/preproc_rules
var  WHITE_LIST_PATH /etc/snort/rules/iplists
var  BLACK_LIST_PATH /etc/snort/rules/iplists
```

Ensuite nous allons inclure le chemin de notre règle à la ligne 660 :

```
include  $RULE_PATH/voip-alert.rules
```

4.2.2.5. Création des règles pour SNORT

Les règles de snort sont décrites dans un langage simple et suivent le schéma suivant :

L'en-tête de règle qui contient :

- L'action de la règle (la réaction de snort).
- Le protocole qui est utilisé pour la transmission des données (snort en considère trois : TCP, UDP et ICMP).
- Les adresses IP source et destination et leur masque.
- Les ports source et destination sur lesquels il faudra vérifier les paquets.

Les options de la règle (entre parenthèse) qui contiennent :

- Le message d'alerte.
- Les conditions qui déterminent l'envoi de l'alerte en fonction du paquet inspecté.

Voici un simple exemple d'une règle qui permet de détecter les tentatives de login sous l'utilisateur root, pour le protocole ftp (port 21) :

```
alert  tcp  any  any -> 192.168.1.0/24 21 (content : "USER root"; nocase; msg: "Tentative d'accès au FTP pour l'utilisateur root";)
```

Les messages en direction de cette plage d'adresse IP effectuant une tentative de login root ("USER root" contenu dans le paquet) auront pour conséquence la génération de l'alerte "Tentative d'accès au FTP pour l'utilisateur root".

Maintenant nous allons créer des règles dans le fichier **voip-alert.rules** pour détecter les scans de ports, les tentatives de dos et le brute force sur le serveur Asterisk

# cd /etc/snort/rules	accès au dossier rules
# nano voip-alert.rules	accès au fichier ou vont être crée nos règles

1ere règle :

```
alert icmp 192.168.1.8 any -> 192.168.1.10 any (msg:"Getting pings from 192.168.1.8"; sid:1000004;)
```

Une alerte sera générée si le serveur reçoit des pings provenant de l'adresse IP 192.168.1.8 qui est celle de la machine pirate.

2éme règle :

```
alert icmp any any -> 192.168.1.10 any (msg:"ICMP PING NMAP"; dsize:0; itype:8; reference:arachnids,162; classtype:attempted-recon; sid:469; rev:3;)
```

Cette règle permet de détecter les scans nmap sur n'importe quel port du serveur.

3éme règle :

```
alert udp any any -> 192.168.1.10 any (msg:"UDP flood attack detected"; threshold: type threshold, track by_dst, count 20 , seconds 1 ; sid: 5000003; rev:1;)
```

Puisque par défault le serveur asterisk utilise UDP comme protocole de transport, cette règle permet de détecter un DOS de type UDP flood, si plus de 20 paquets UDP sont envoyé vers notre serveur en moins d'une seconde, snort va considérer cela comme une tentative de DOS et va générer une alerte contenant le message "UDP flood attack detected".

4éme règle :

```
alert tcp any any -> 192.168.1.10 80 (flags: S; msg:"Possible TCP DoS Detected"; flow: stateless ; threshold: type both, track by_src, count 100,seconds 10; sid:10001;rev:1;)
```

Cette règle permet de détecter une attaque TCP DOS sur le port 80 du serveur, si plus de 70 paquets TCP sont reçu en moins de 10 secondes par le serveur SIP, snort va considérer cela comme une éventuelle tentative de DOS et va générer une alerte sous forme d'un message « Possible TCP Dos detected ».

5éme règle :

```
alert ip any any -> 192.168.1.10  any (msg:"INVITE message flooding"; content:"INVITE";
depth:6; threshold: type both, track by_src, count 200,seconds 60;sid:1000100; rev:1;)
```

Cette règle permet de détecter des attaques par brute force sur le serveur SIP, si plus de 200 requêtes INVITE sont envoyé au serveur en moins de 60 seconds snort va considérer cela comme une éventuelle tentative de brute forcing et va générer une alerte.

6éme règle :

```
alert tcp any any ->  192 .168.1.10  any (msg:"NULL Scan";flags: 0;sid:1000017;)
```

Le scan Null est un type de TCP scan que les pirates - à la fois éthiques et malveillants - utilisent pour identifier les ports d'écoute TCP. Une analyse Null est une série de paquets TCP qui contiennent une séquence de 0 et aucun drapeau. Il peut parfois pénétrer les pare-feu et les routeurs de bord qui filtrent les paquets entrants avec des drapeaux particuliers. Le résultat attendu d'un scan Null sur un port ouvert est sans réponse. Comme il n'y a pas de drapeaux fixés. Cette règle permet de détecter ce type de scan.

Nous allons maintenant lancer SNORT en mode NIDS (network intrusion détection system) en utilisant la commande suivante sur le terminal :

```
#     snort  -A  console  -i  eth1  -c  /etc/snort/snort.conf  -l  /var/log/snort  -K  ascii
```

-A console : permet d'afficher les messages sur l'écran.

-i eth1 : spécifie l'interface d'écoute de snort

-c /etc/snort/snort.conf : spécifie le fichier de configuration que nous avons utilisé, cela comprendra les règles personnalisées que nous avons ajouté plus tôt

-l /var/log/snort : spécifie le répertoire où seront situés les logs

-K ascii : spécifie la façon dont les fichiers journaux seront écrits, ascii peut facilement être ouvert par un éditeur de texte ou par la commande "cat"

Après le lancement de snort en mode NIDS (voir figure 4.3), nous avons retenté un scan de port en utilisant nmap, un DOS (SYN flood et UDP flood) et une attaque par brute force :

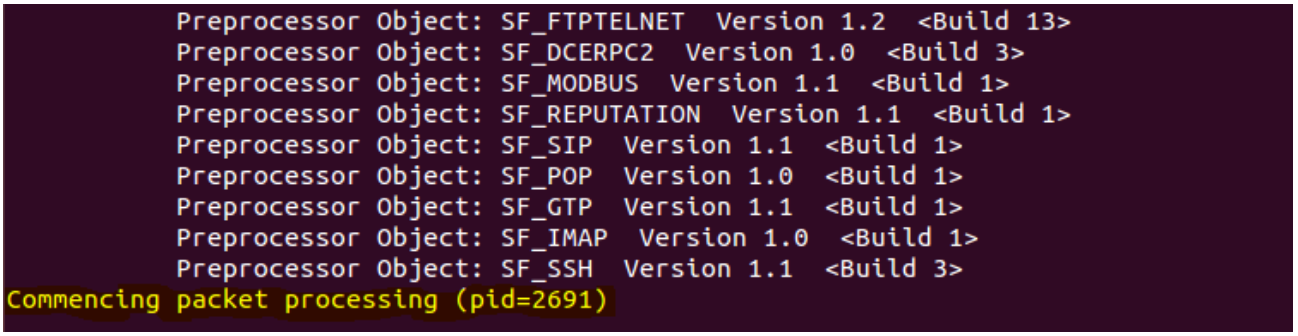

Figure 4.3 : lancement de snort en mode nids

La figure suivante, indique les alertes que snort a générées après l'exécution des attaques (scan de ports, DOS et brute force) :

Figure 4.4 : alertes générées par snort

Snort a pu détecter les scans de ports, les tentatives de DOS (SYN Flood et UDP Flood) et le brute force par des requêtes INVITE, ainsi nous pouvons stopper ses attaques,

bloquer les adresses IP suspectes et filtrer le trafic entrant vers le serveur, tout cela peut se fera en utilisant le pare-feu Netfilter.

4.3. Bonne pratique contre l'Arp poisoning :

Pour détecter ce genre d'attaque, il existe un utilitaire nommée **Arpwatch.** Ce dernier permet de surveiller en temps réel l'activité du protocole ARP dans un réseau informatique en vérifiant les correspondances adresses IP-MAC. ARPWATCH va loguer les événements dans /var/log/syslog et /var/log/messages et pourra nous notifier par email de chaque changement du couple adresse MAC-IP, ce qui s'avère être très pratique lorsqu'on est loin de notre réseau et qu'on désire toujours garder un œil dessus. Pour l'installer, il suffit de taper la commande suivante sur le terminal :

```
# apt-get install   arpwatch
```

La configuration d'arpwatch se fait dans **/etc/arpwatch.conf**, on peut y accéder en utilisant l'éditeur de texte **nano** (voir figure 4.5), ensuite il faut ajouter la ligne suivante :

```
Interface carte réseau  -a  -n  192.168.1.0/24  -m   adresse email
```

```
root@anis-VirtualBox:~# nano /etc/arpwatch.conf

  GNU nano 2.2.6          Fichier : /etc/arpwatch.conf


# or, if you have an MTA configured for plussed addressing:
#
#eth0    -m root+eth0
#eth1    -m root+eth1
#eth2    -m root+eth2

eth0 -a -n 192.168.1.0/24 -m anisamziane6810@gmail.com
eth1 -a -n 192.168.1.0/24 -m anisamziane6810@gmail.com
```

Figure 4.5 : configuration d'arpwatch

Explication des variables :

eth0 : le nom de votre interface réseau.

-a : pour dire qu'arpwatch fait un rapport de toutes les nouvelles adresses IP.

-n : permet de spécifier l'adresse réseau et le masque de sous-réseau

- 192.168.1.0/24: correspond à l'adresse de notre réseau et le masque de sous-réseau (/24 correspond à 255.255.255.0)

-m : permet d'envoyer le rapport à l'adresse email spécifiée.

Une fois la configuration effectuée, on redémarre arpwatch (voir figure 4.6).

Ensuite on lance le service (voir figure 4.7) :

Figure 4.6 : redémarrage d'arpwatch

Figure 4.7 : Lancement d'arpwatch

Bien sûr pour qu'arpwatch puisse envoyer des emails, il faut que notre serveur soit configuré pour on envoyer. sSMTP est un petit utilitaire qui permet de le faire.
Pour l'installer rien de plus simple. sSMTP se trouve dans les dépôts, donc il suffit de taper dans le terminal la commande suivante pour l'installer :

```
# sudo apt-get install ssmtp
```

Ensuite on doit configurer sSMTP dans les fichier suivants /etc/ssmtp/ssmtp.conf et /etc/ssmtp/revaliases :

```
# nano /etc/ssmtp/ssmtp.conf                    accès au fichier ssmtp.conf
```

```
  ### CONFIGURATION GENERALE ###

 MailHub=smtp.gmail.cim :587     # Serveur SMTP vers lequel forwarder les
                                    mails
RewriteDomain=                   # Domaine depuis lequel est envoyé le
                                    mail(on peut le laisser vide)
Hostname=nom.de.votre.serveur    # Nom de la machine
FromLineOverride=yes            # Ré-écriture de l'expéditeur (champ from)

Root=anisamziane6810@gmail.com    # redirige les mails à destination de
                                    "root" vers srv-ced@ced-info.com


 ### CONFIGURATION DE L'AUTHENTIFICATION ###

UseTLS=yes                      # Utilisation d'une connexion sécurisée SSL
                                  ou TLS (si votre FAI l'utilise sinon
                                                Mettre no)

AuthUser=anisamziane6810@gmail.com      # Nom d'utilisateur pour
                                          L'authentification SMPT
AuthPass :                              # Le mot de passe correspondant
```

Une fois ce fichier de configuration sauvegardé nous allons pouvoir modifier le fichier /etc/ssmtp/revaliases.

```
#  nano  /etc/ssmtp/revaliases                    accès au fichier revaliases
```

```
#  sSMTP aliases

#  Format :       local_account : outgoing_address : mailhub
# Example : root : your_login@your.domain : mailhub.your. domain [:po$
#  where [: port] is an optional port number that defaults to 25.

        anis:anisamziane6810@gmail.com:smtp.gmail.com :587
```

Voilà, maintenant le serveur peut envoyer des mails. Nous avons tenté une nouvelle fois l'ARP poisoning, voici les messages envoyés par notre serveur (voir figure 4.8) :

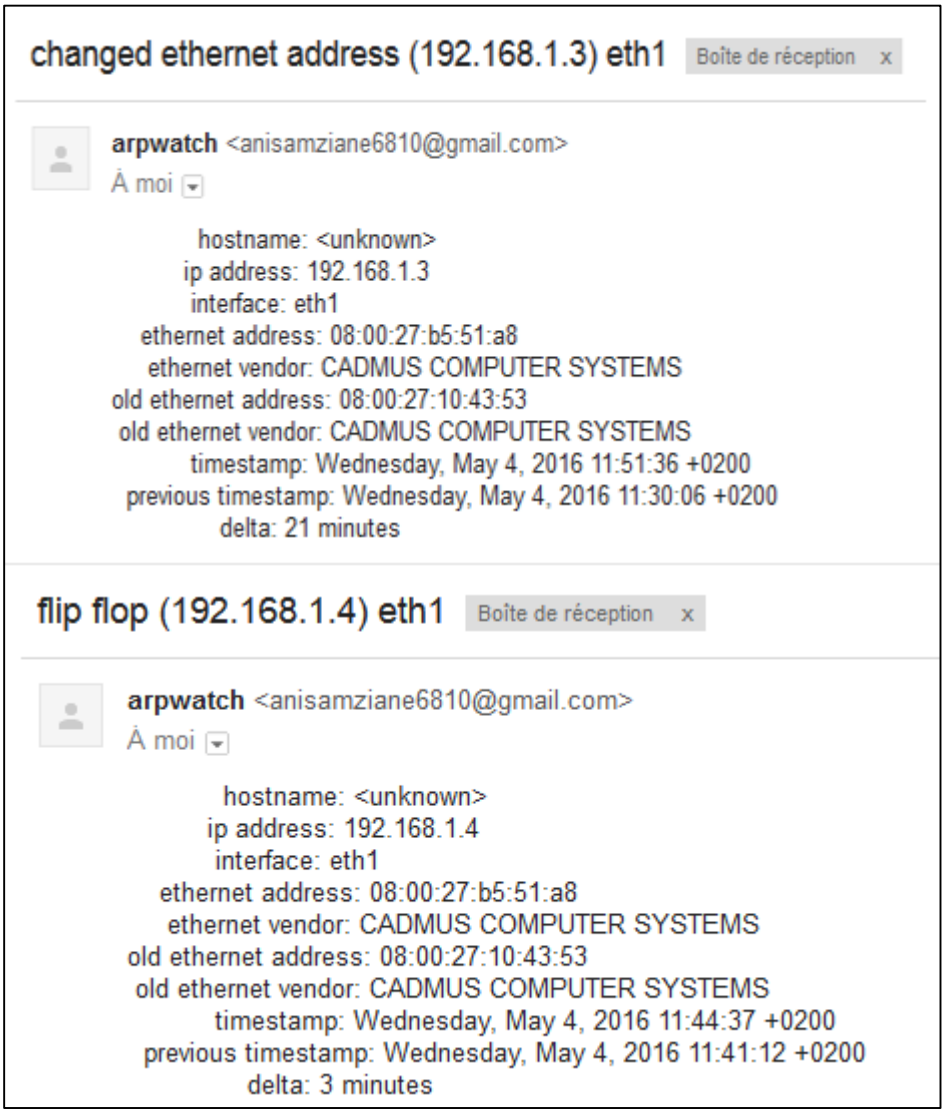

Figure 4.8 : message signalant un changement d'adresse mac

Les deux messages nous notifient d'un changement dans le cache ARP du serveur indiquant l'ancienne adresse mac des deux clients 6001 et 6002 ayant les adresse IP 192.168.1.4 et 192.168.1.3 et leur nouvelle adresse mac (08 :00 :27 :b5 :51 :a8) qui est bien évidemment celle de la machine pirate. Ainsi même si on est loin de notre réseau en peut détecter et se protéger de ce type d'attaque.

Une autre solution peut s'avérer être utile, c'est de mettre l'enregistrement des adresse MAC à l'état statique pour que la cache ARP ne soit pas mise à jour à chaque réception de paquets présentant des couples IP-MAC différents. Ces enregistrements statiques ne seront ainsi plus falsifiables via des requêtes ARP malicieuses**.**

Cela peut se faire avec la commande suivante :

```
#     arp  -s  adresse IP   adresse  MAC
```

Ici, l'option « **-s** » indique le renseignement de l'association IP-MAC en *"static",* une attaque par ARP spoofing ne pourra plus la modifier. Le seul inconvénient de cette méthode c'est qu'il faut changer l'enregistrement statique sur tous les hôtes clients dans le cas d'un changement d'interface réseau ou bien remplacement d'un matériel

4.4. Bonne pratique contre l'écoute clandestine

La sécurisation des appels peut être intéressante à mettre en place, afin de protéger nos appels téléphoniques. Nous pouvons chiffrer le flux de signalisation ainsi que le flux audio. De cette manière, nous pouvons assurer la confidentialité des appels. Donc pour sécuriser les appels, il nous faut chiffrer deux choses :

- Le flux SIP (la signalisation)
- Le flux RTP (la voix)

Pour réaliser cela on va utiliser le protocole TLS et le protocole SRTP.

4.4.1. TLS (Transport Layer Security) :

Transport Layer Security (TLS) est un protocole cryptographique qui permet de crypter les segments de connexions (informations) échangées entre un client et un serveur, en utilisant la cryptographie asymétrique pour l'échange de clés, le chiffrement symétrique pour la confidentialité, et les codes d'authentification de message pour l'intégrité du message. En utilisant TLS pour crypter la signalisation SIP, des paramètres tels que les numéros de téléphone et les noms d'utilisateur envoyés entre le serveur Asterisk et l'application softphone ne peuvent pas être capturés. TLS aide également à empêcher des tiers de l'écoute ou la falsification des messages. Le protocole TLS se décompose en deux couches principales :

- **TLS Handshake Protocol** : choisit la version de TLS qui sera utilisée, réalise l'authentification par l'échange de certificats et permet la négociation entre le

client et le serveur d'un niveau de sécurité au travers du choix des algorithmes de cryptage. C'est le protocole de configuration de la transaction.

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

4.4.1.1. Principe de Fonctionnement

La sécurisation des transactions par TLS est basée sur un échange de clés entre le client et le serveur. La transaction sécurisée par TLS se fait selon le modèle suivant :

- Dans un premier temps, le client se connecte au site marchand sécurisé par SSL et lui demande de s'authentifier. Le client envoie également la liste des cryptosystems qu'il supporte, triée par ordre décroissant selon la longueur des clés.

- Le serveur a réception de la requête envoie un certificat au client, contenant la clé publique du serveur (voir figure 4.9), signée par une autorité de certification (CA), ainsi que le nom du cryptosystem le plus haut dans la liste avec lequel il est compatible (la longueur de la clé de chiffrement – 40 bits ou 128 bits - sera celle du cryptosystem commun ayant la plus grande taille de clé).

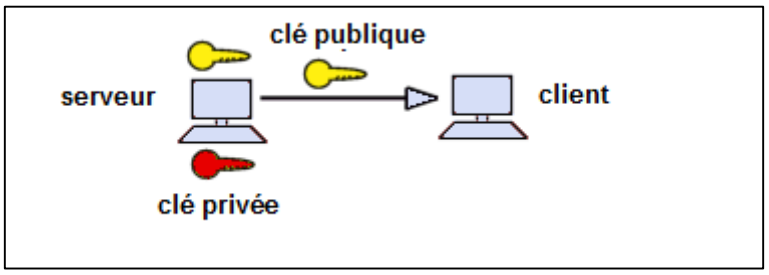

Figure 4.9: envoie de la clé publique au client. [28]

- Le client vérifie la validité du certificat (donc l'authenticité du marchand), puis crée une clé secrète aléatoire (plus exactement un bloc prétendument aléatoire), chiffre cette clé à l'aide de la clé publique du serveur, puis lui envoie le résultat (la clé de session).

- Le serveur est en mesure de déchiffrer la clé de session avec sa clé privée. Ainsi, les deux entités sont en possession d'une clé commune dont ils sont seuls connaisseurs. Le reste des transactions peut se faire à l'aide de clé de session, garantissant l'intégrité et la confidentialité des données échangées.

4.4.2. SRTP

Secure Real-time Transport Protocol définit un profil de RTP qui a pour but d'apporter le chiffrement, l'authentification et l'intégrité des messages et la protection du flux de données RTP. RTP fonctionne à la fois en envoi ciblé et en multidiffusion. Pour le chiffrement et le déchiffrement du flux de données (et donc pour assurer la confidentialité des flux de données), SRTP (avec SRTCP) utilise AES (Advanced Encryption standard) comme chiffrement par défaut. 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 est aussi 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)

4.4.2.1. 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 même message.
- La protection contre le rejet (replay) des paquets. Chaque récepteur tient à jour une liste de tous les indices des paquets reçus et bien authentifiés.

4.4.2.2. 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.
- **Des clés aléatoires utilisées** pour introduire une composante aléatoire afin de contrer les éventuels rejet ou effets de mémoire.

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

4.4.2.3. Format du paquet SRTP :

Un paquet SRTP est généré par transformation d'un paquet RTP grâce à des mécanismes de sécurité. Donc le protocole SRTP effectue une certaine mise en forme des paquets RTP avant qu'ils ne soient sur le réseau. La figure 4.10 présente le format d'un paquet SRTP.

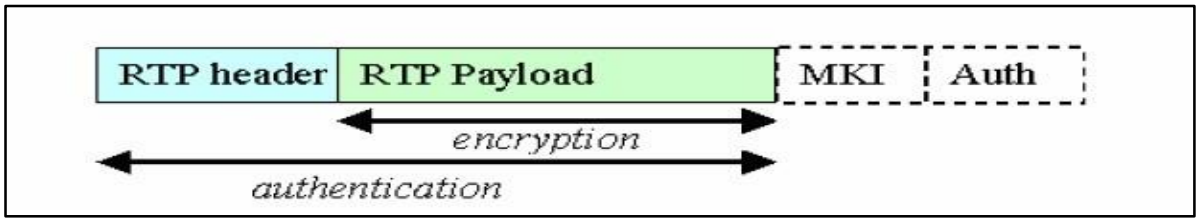

Figure 4.10 : Format d'un paquet SRTP. [15]

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

- SRTP MKI (SRTP Master Key identifier) : sert à ré-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 rejet de paquets en authentifiant le numéro de séquence.

4.4.3. Chiffrement de la signalisation

Pour cela, nous allons faire appel au protocole TLS **(**Transport Layer Security). Tout d'abord, il va nous falloir créer des clés pour Asterisk et les clients profitant du chiffrement. Ensuite, nous devrons autoriser Asterisk à utiliser TLS pour les échanges SIP, puis nous devrons choisir les clients à sécuriser. Enfin, il faudra activer le chiffrement sur le poste du client, et lui fournir les fichiers de clé.

Commençons donc par générer les clés

```
# mkdir  /etc/asterisk/keys                    (création du fichier Keys)
```

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

```
# cd  /asterisk/asterisk-11.22.0/contrib/scripts/
```

Ensuite, on exécute le script avec la commande suivante :

```
# ./ast_tls_cert -c asterisk.Networklab -o " Networklab" –d /etc/asterisk/keys
```

La figure suivante montre l'exécution du script :

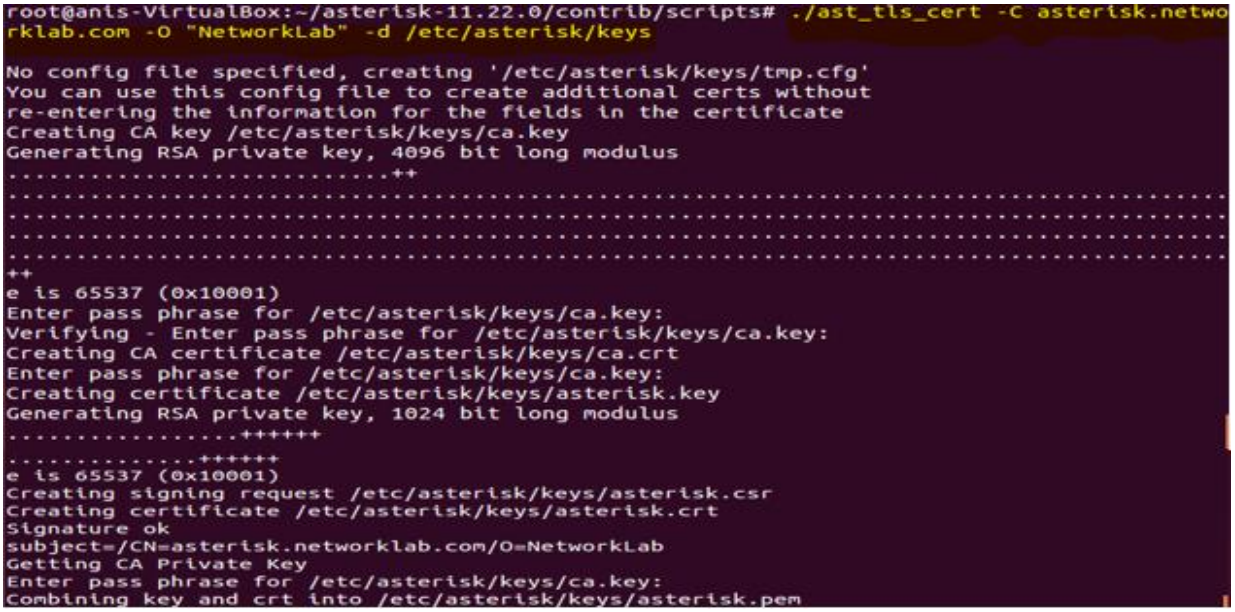

Figure 4.11: exécution du script et création des clés.

Voici les détails des options :

- **-c** : permet de spécifier le nom d'hôte du serveur Asterisk. A défaut d'un nom, vous pouvez spécifier une IP.
- **-o** : permet de définir le nom de l'organisation
- **-d** : permet de spécifier le dossier de sortie

A l'exécution du script, il vous sera demandé de spécifier le mot de passe des fichiers. Retenez bien le mot de passe entrer pour chaque fichier, ou à défaut, utilisez le même à chaque fois. Les fichiers suivants devraient être créés (voir figure 4.12) :

Figure 4.12: création des clés et des certificats

-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.

-Un fichier PEM regroupant la clé privée et le certificat.

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

A présent, nous devons créer les clés et certificats pour le client admin6001

```
# ./ast_tls_cert -m client -c /etc/asterisk/keys/ca.crt -k /etc/asterisk/keys/ca.key -C
  6001.networklab.com -O "Networklab" -d /etc/asterisk/keys -o 6001
```

Pour les autres utilisateurs on doit juste changer 6001 par 6002 ou par 6003 selon leur extension dans le fichier **sip.conf**

Voici les détails de la commande :

- **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. Il est possible de spécifier une 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.

Ainsi nous retrouvons 4 nouveaux fichiers dans le dossier **Keys** que nous avons créé (voir figure 4.13)

```
root@anis-VirtualBox:~# ls /etc/asterisk/keys/
6001.crt  6002.crt  6003.crt  asterisk.crt  ca.cfg
6001.csr  6002.csr  6003.csr  asterisk.csr  ca.crt
6001.key  6002.key  6003.key  asterisk.key  ca.key
6001.pem  6002.pem  6003.pem  asterisk.pem  tmp.cfg
```

Figure 4.13 : création des clés et des certificats pour les clients

L'opération est à répéter pour tous les clients devant bénéficier du TLS.A présent, nous devons configurer Asterisk pour autoriser l'utilisation de TLS.

Dans le fichier **sip.conf**, on doit ajouter les lignes 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, on ajoute la ligne **transport=tls** pour tous les clients concernés :

```
[6001]
type=friend
secret=secret
host=dynamic
dtmfmode=rfc2833
disallow=all
allow=ulaw
fullname= admin 1
username=admin6001
context=work
transport=tls
```

```
[6002]
type=friend
secret=secret
host=dynamic
dtmfmode=rfc2833
disallow=all
allow=ulaw
fullname= admin 1
username=admin6001
context=work
transport=tls
```

Enfin, nous pouvons configurer les postes des clients. Pour cette démonstration, on a choisi d'utiliser le softphone **Blink** pour sa facilité de configuration. Blink est téléchargeable depuis le lien suivant : **http://icanblink.com/download/**

Activation du TLS sur le poste du Client **6001 :**

Le client doit posséder 2 fichiers :

- ca.crt
- 6001. pem

Voici la configuration à appliquer pour Blink (voir figure) :

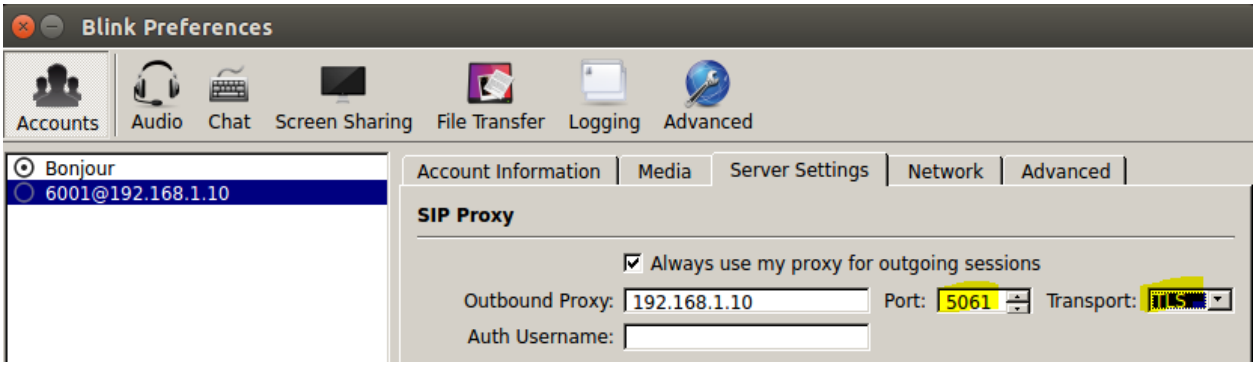

Figure 4.14: configuration de Blink

Ensuite renseigner le chemin du fichier PEM (certificat + clé privée du client) (voir figure 4.15)

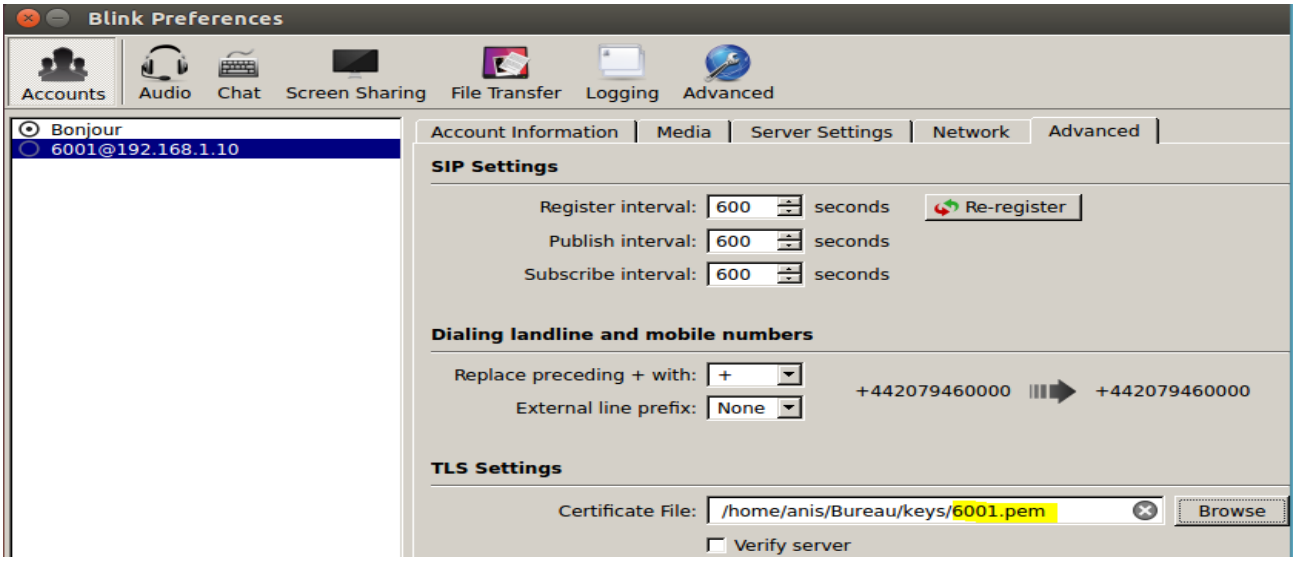

Figure 4.15 : enregistrement du certificat .pem

Puis renseigner le chemin du fichier ca.crt (voir figure 4.16)

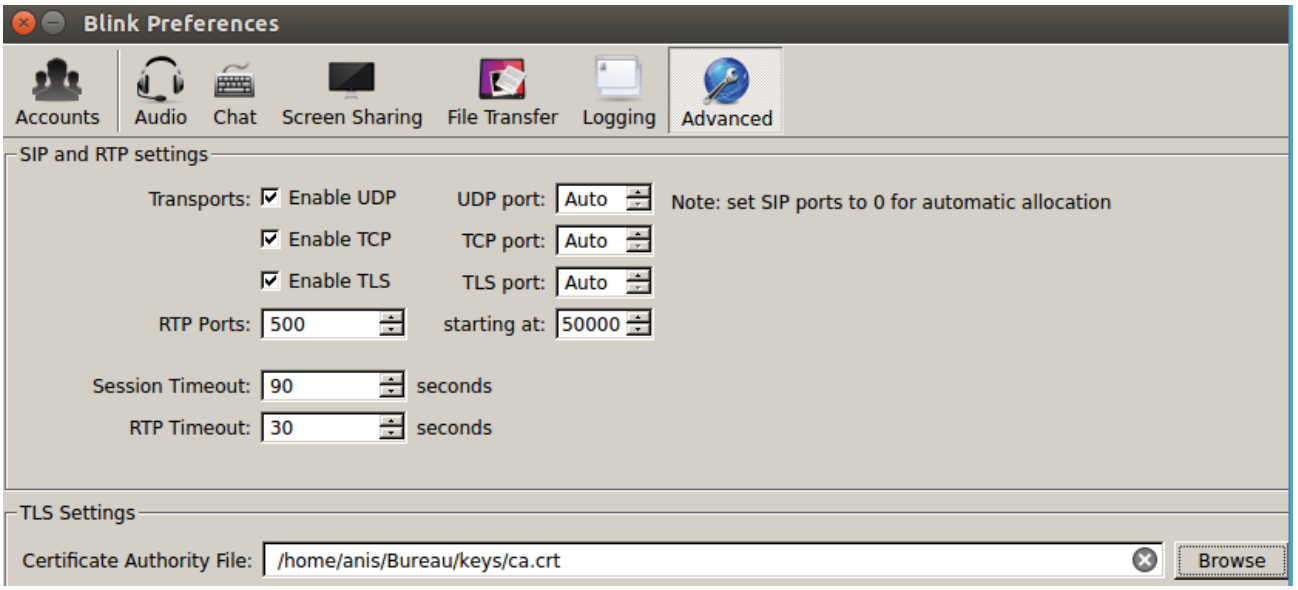

Figure 4.16: enregistrement du fichier ca.crt

A présent, la signalisation de l'appel devrait être chiffrée (petit cadenas vert dans Blink) (voir figure 4.17).

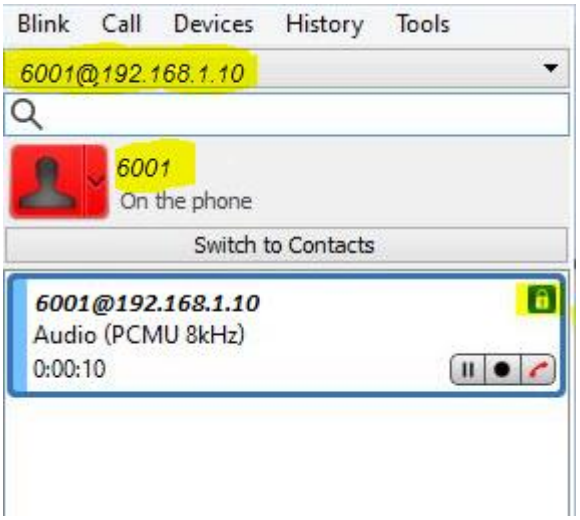

Figure 4.17 : activation du TLS sur Blink

A présent que la signalisation est chiffrée, il nous faut encore chiffrer le flux audio. Pour cela, nous allons utiliser le protocole SRTP.

4.4.4. Chiffrement du flux audio :

La première étape consiste à ajouter le support de SRTP à Asterisk. Commençons par télécharger la librairie SRTP. Le lien vers la dernière version est à cette adresse : **http://srtp.sourceforge.net/download.html**

Le téléchargement de LIBSRTP se fait come suite :

```
#  cd  /usr/src
#  wget http://srtp.sourceforge.net/srtp-1.4.2.tgz
#  tar -xvzf srtp-1.4.2.tgz
#  rm srtp-1.4.2.tgz
```

Après le téléchargement de LIBSRTP on Procède à son installation comme suit :

```
#  cd srtp                                          accès au fichier srtp
#  ./configure CFLAGS=-fPIC --prefix=/usr           configuration
#   make                                            installation
#   make install
```

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

```
# cd /usr/src/asterisk/asterisk-11.22.0/
# make clean
# ./configure
# make
# make install
```

Ensuite, charger le module SRTP dans Asterisk :

```
# asterisk -rv
# module load res_srtp.so
```

A présent, il faut forcer l'utilisation de SRTP sur les clients voulus. Pour cela, il faut ajouter la ligne **encryption=yes** chez les utilisateurs concernés dans **users.conf**.

Maintenant le flux RTP devrait être chiffré (voir figure 4.18)

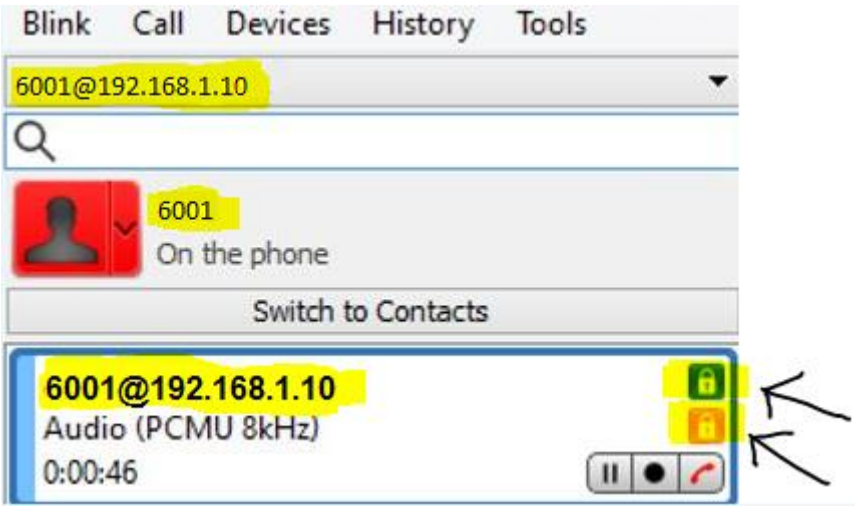

Figure 4.18: activation de TLS et de SRTP

Apparition de deux cadenas sur notre softphone indiquant que le flux SIP et le flux RTP sont désormais cryptés, ainsi nous pouvons nous protéger contre l'écoute clandestine.

4.5. Implémentation d'un firewall Netfilter :

Un firewall doit être imprenable car sinon le réseau entier est compromis. Un firewall efficace doit posséder plusieurs interfaces réseaux pour pouvoir faire un filtrage entre plusieurs zones.

Dans le cadre de notre projet le firewall va nous permettre de stopper certaines attaques et minimiser le trafic entrant au serveur Asterisk. En effet notre objectif est de ne laisser passer que le trafic VoIP et plus exactement les paquets basés sur le protocole SIP et le protocole RTP, qui sont utilisés par le serveur Asterisk pour le trafic VoIP. C'est pour cela que nous avons choisi de mettre un firewall au niveau du serveur et de cette façon toutes les requêtes en direction du serveur Asterisk passeront automatiquement par le firewall. La plupart des firewalls propre à la VoIP sont à titre commercial et donc non libre.

IPtables est la commande permettant de paramétrer le filtre Netfilter du noyau Linux et donc de configurer le Firewall. IPtables est par default pré- installé sur une distribution Linux, si ce n'est pas le cas il suffit de l'installer simplement en tapant la commande suivante sur le terminal :

```
# sudo apt-get install iptables
```

IPtables fonctionne selon un système de tables, ces tables sont composées de chaînes. Dans le cadre de la configuration et de l'utilisation de Netfilter comme pare-feu, c'est la table Filter qui est utile, elle permet de filtrer les paquets réseaux. Tout paquet entrant est analysé afin de déterminer notamment sa source et sa destination.

Elle est composée de trois sortes de chaîne :

- **INPUT** : Permet d'analyser les paquets entrants. Si le paquet est adressé au poste, il est confronté au filtre INPUT.
- **FORWARD** : Permet d'analyser et d'autoriser les trames à passer d'une interface à une autre, seulement dans le cadre d'une interface réseau servant de passerelle.
- **OUTPUT** : Permet d'analyser les paquets sortants. Si le paquet sort du poste, il passera par la chaîne OUTPUT.

Dans l'exemple qui suit nous avons programmé notre firewall pour qu'il puisse laisser passer que le trafic VoIP au niveau du serveur Asterisk et de bloquer tous le trafic restant. Voici les commandes exécutées :

```
# iptables -A INPUT -p udp -m udp --dport 5060 -j ACCEPT
```

Cette commande va permettre d'accepter le trafic UDP entrant du port 5060. Ce numéro de port n'est autre que celui du protocole SIP.

```
# iptables -A INPUT -p udp -m udp --dport 10000:20000 -j ACCEPT
```

Cette commande va permettre d'accepter le trafic UDP entrant du protocole RTP.

Ensuite il faut attribuer une règle par défaut pour bloquer tous le trafic restant et qui passe par UDP :

```
# iptables -A INPUT -p UDP -j DROP
```

En ce qui concerne le flood ou dos (déni de service), nous pouvons implémenter les règles suivantes pour limiter le nombre de demandes de connexion au minimum vital :

```
# iptables -A INPUT -p tcp --syn -m limit --limit 1/s -j ACCEPT
```

Cette règle va laisser passer que le trafic de synchronisation en limitant la réception des requêtes de synchronisation à une requête par secondes. Ainsi nous pouvons éviter les attaques de type SYN flood.

```
# iptables -A FORWARD -p udp -m udp --dport 5060 -m limit --limit 1/second -j
  ACCEPT
```

Celle-ci va donc limiter le trafic UDP entrant sur le port 5060 a une requête par seconde.

On a vue auparavant que le scan de ports consiste à tester tout ou une partie des ports d'une machine pour déterminer ceux ouverts et donc potentiellement les services "attaquables".

On peut donc en bloquer certains en utilisant la commande suivante :

```
# iptables -A FORWARD -p tcp --tcp-flags SYN,NULL,ACK,FIN,RST -j DROP
```

Tous les scans utilisant des paquets ayant un drapeau de type SYN, NULL, ACK, FIN, RST seront bloqué.

Pour les attaques par brute force, on peut utiliser les deux lignes de commandes suivantes pour limiter le nombre de connexions sur le port 5060 :

```
# iptables -A INPUT -p udp --dport 5060 -i eth0 -m state --state NEW -m recent --set
# iptables -A INPUT -p udp --dport 5060 -i eth0 -m state --state NEW -m recent --update --seconds 5 --hitcount 5 -j DROP
```

L'option « --set » dans la première ligne permet d'ajouter l'adresse IP de l'hôte qui a initié la connexion au port 5060 du serveur à la « liste récente ». Ensuite dans la deuxième ligne, cette adresse IP sera bloquée si elle introduit plus de 5 mot de passe erronés en moins de 5 secondes.

On peut également utiliser la commande suivante pour bloquer une adresse IP jugé suspecte, par exemple L'IP « 192.168.1.8 » :

```
# iptables -A INPUT -s 192.168.1.8 -j DROP
```

Ainsi, tous les paquets entrants en provenance de la machine ayant l'adresse IP 192.168.1.8 seront systématiquement bloqués.

4.6. Discussion

Tout au long de ce chapitre, nous avons proposé et implémenté au sein du réseau VoIP déployé des mesures de sécurités à prendre, afin de se protéger des attaques évoquées précédemment et d'autres attaques similaires. Les différentes attaques simulées ont été détecter et bloquer.

Nous pensons avoir apporté aux utilisateurs de la VoIP des solutions efficaces pour mieux sécuriser le serveur SIP ainsi que leurs conversations téléphoniques.

Conclusion

L'objectif de ce travail est de mettre en place une solution libre permettant la transmission de la voix sur des réseaux utilisant le protocole IP et de la sécuriser en élaborant différents mécanismes et protocoles.

Dans une première étape, nous nous sommes intéressés à l'étude générale de la VoIP avec ses différents protocoles et standards. Dans une deuxième étape, nous avons réalisé une solution assurant la transmission de la voix sur le réseau IP à base de serveur ASTERISK, nous avons montré les étapes d'installation et de configuration d'asterisk sous le système d'exploitation linux Ubuntu ainsi que l'installation et la configuration de ZOIPER softphone qui est un logiciel gratuit pour la téléphonie sur IP suivis de testes d'appel entre les différents clients SIP. Comme troisième étapes, nous avons présenté les vulnérabilités et les attaques qui menacent la VoIP. Nous sommes passés ensuite à la réalisation de quelques attaques sur le réseau de voix sur IP déployé en s'appuyant sur les attaques que nous avons définies. En dernière étape, nous nous sommes intéressés aux solutions, mécanismes et configurations à mettre en place dans le but de sécuriser la solution VoIP contre les différentes techniques de hacking que nous avons simulées.

Selon les testes que nous avons réalisée, toutes les attaques ont été détectées et bloquées à l'aide de l'IDS Snort, l'application ARPWATCH, protocoles TLS et SRTP et un pare-feu Netfilter. En effet, l'IDS Snort a permis de détecter les scans de ports, les attaques par déni de service et l'attaque par brute force, alors que l'application ARPWATCH a identifié les attaques par empoisonnement ARP en nous notifiant par mail. Les protocoles TLS et SRTP ont contribué aux chiffrements de la voix et de la signalisation permettant ainsi de se protéger de l'écoute clandestine. Le pare-feu permet de filtrer le trafic entrant au serveur et choisis le type de paquets à accepter limitant ainsi les attaques de type DOS, scans de ports et brute force.

Après l'implémentation de ses solutions, nous avons abouti à une sécurisation appréciable grâce à la complémentarité des différentes applications. Selon tous ces résultats, nous pouvons dire que nous avons apporté aux utilisateurs de la VOIP une solution de sécurisation fiable.

Pour optimiser cette solution, il est intéressant d'utiliser aussi le protocole ZRTP. Ce dernier est une version améliorée du protocole SRTP, plus souple, il utilise directement le flux multimedia plutôt que le flux de signalisation pour établir les clés de chiffrement SRTP. Cela rend ZRTP un choix idéal pour une utilisation sur des réseaux où la signalisation est traitée par des dispositifs intermédiaires et où il est important d'assurer la confidentialité des appels.

Bibliographie/Webographie :

[1] : L. Ouakil–G. Pujolle, Téléphonie sur IP, éd.Eyrolles, 2007,458 p.

[2]: HimanshuDwivedi, Hacking VoIP, Protocols, Attacks, and Countermeasures, 2009,211 p.

[3]: Peter Thermos and Ari Takanen: Securing VoIP networks, threats, vulnerabilities, and counter measures, Addison-Wesley, 2007,384 p .

[4]: David Endler and Mark Collier, Hacking Exposed VoIP: Voice Over IP Security, Secrets &Solutions, McGraw-Hill/Osborne, 2009, 574 p.

[5]: Tobias Glemser et Reto Lorenz, Sécurité pour le système Voice over IP – protocolesSIP et RTP, Article publié dans le numéro 5/2005 du magazine *hakin9*

[6]: R. Rehman, "Intrusion Detection Systems with Snort. New Jersey: Prentice Hall PTR, 2003.

[7]: James **F. Ransome, John W. Rittinghouse**, VoIP Security, Elsevier Digital Press, 2005, 402 p.

[8]: D. Sisalem, J. Kuthan, T. Elhert, Denial of Service Attacks Targeting SIP VoIP Infrastructure: Attack Scenarios and Prevention Mechanisms. *IEEE Network*, 2006.

[9]: ABDELRAHIM Ibrahim Mahamat, "Mise en place d'un système de Téléphonie sur IP basé sur le logiciel Asterisk.pdf",2013-2014, URL : http://abdelrahim.sunulab.com/wp-content/uploads/2014/12/PRESENTATION-DE-ASTERISK. . Date de dernier accès 05/06/2016.

[10]: M. Fathi BEN NASR et Mme Alia KHESSAIRI ABBASSI, mémoire, "Mise en place d'une sonde SNORT", Université de la Manouba Ecole Nationale des Sciences de l'Informatique 2004-2005.

[11]: Didi Souhila et Guerriche Meryem, mémoire, "La Téléphonie sur IP (ToIP) ",

http://dspace.univ-tlemcen.dz/bitstream/112/6352/1/La-Telephonie-sur-IP"

.pdf. Date de dernier 05/06/2016

[12]: Willy Wandji, mémoire, "Mise en place d'une solution VoIP au sein de l'autorité aéronautique camerounaise", Institut Africain d'Informatique 2012. http://www.memoireonline.com/10/13/7498/Mise-en-place-dune-solution-VoIP-au-sein-de-lautorite-aeronautique-camerounaise.html. Dernier accès le 05/06/2016

[13]: Yannick YANI KALOMBA, mémoire, " Etude et Mise Au Point Dun système de Communication VOIP : Application Sur Un PABX-IP Open Source „Cas de L'agence En Douane Getrak". http://www.memoireonline.com/08/11/4644/m_Etude-et-mise- au-point-dun-systeme-de-communication-VOIP--application-sur-un-PABX-IP-open-source10.html.Dernier accès le 05/06/2016.

[14]: Rabha Bouzaida, mémoire, "Étude et Mise en place d'une Solution VOIP Sécurisée ", 2010-2011, URL : http://pf-mh.uvt.rnu.tn/620/1/%C3%89tude_et_Mise_en_place_d%27une_Solution_VOIP_S%C3%A9curis%C3%A9e.pdf . Dernier accès le 05/06/2016.

[15]: Denis TSHIMANGA, mémoire, "Etude d'implémentation d'une solution VOIP Sécurisée dans un réseau informatique d'entreprise. Cas de l'ISTA de Kinshasa", Institut supérieur de techniques appliquées de Kinshasa 2012, URL : http://www.memoireonline.com/09/13/7361/m_Etude-dimplementation-dune-solution-VOIP-securisee-dans-un-reseau-informatique-dentrepr0.html , dernier accès le 05/06/2016.

[16]: "H323_EFORT.pdf", URL: http://www.efort.com/r_tutoriels/H323_EFORTpdf

[17]: Le Protocole H323 : Equipements, Avantages et Inconvénients." Par *SAeeeD Blog*. URL : http://www.blog.saeeed.com/2011/03/h323-protocole-protocole-gateway-gatekeeper- h-323/, Dernier accès le 05/06/2016

[18]: SebF, Voix sur IP, URL : http://www.frameip.com/voip/ ,accès 01/06/2016

[19]: Beshir&Aziz, Mise en place d'une solution VoIP sécurisée, URL : http://ts5ri-voip-pfe.fr.gd/ , date de dernier accès 05/06/2016

[20]: Mao, introduction to arp Poison Routing, URL :http://www.oxid.it/downloads/apr-intro.swf Date de dernier accès le 05/06/2016

[21]: Mickael Dorigny, Comprendre les attaques via ARP spoofing (MITM, DOS), URL :https://www.information-security.fr/attaque-man-in-the-middle-via-arp-spoofing , dernier accès le 05/06/2016

[22]: Un article de Wikipédia, l'encyclopédie libre, Saturation de la table D'apprentissage, URL : https://fr.wikipedia.org/wiki/Saturation_de_la_table_d%27apprentissage **,** date de Dernier accès 05/06/2016.

[23]: NightRang3r**,** Penetration Testing VOIP with Backtrack, URL: http://www.backtrack-linux.org/wiki/index.php/Pentesting_VOIP, date de dernier accès 05/06/2016

[24] : Thomas Guillet et Ahmed Serhrouchni, Authentification HTTP Digest SIP Renforcée, URL :http://docplayer.fr/16163844-Authentification-http-digest-sip-renforcee.html, date de dernier accès 05/06/2016

[25] : Un article de Wikipédia, l'encyclopédie libre**,** Attaque par force brute, URL : https://fr.wikipedia.org/wiki/Attaque_par_force_brute

[26] : Un article de Wikipédia, l'encyclopédie libre, Snort, URL : https://fr.wikipedia.org/wiki/Snort, date de dernier accès 05/06/2016

[27] : Loic FONTAINE, Une distribution spécialisée dans la sécurité informatique, Linux Backtrack 5, URL :http://www.lolokai.com/blog/2011/06/14/linux-backtrack-5-une-distribution-specialisee-dans-la-securite-informatique ,date de dernier accès le 05/06/2016

[28] : Jean-François PILLOU**,** SSL - Secure Sockets Layers, URL : http://www.commentcamarche.net/contents/215-ssl-secure-sockets-layers, Jean-François PILLOU**,** dernier accès le 05/06/2016.