La République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université SAAD DAHLEB de BLIDA
Faculté des sciences
Département d'Informatique

Mémoire de fin d'études présenté par :
Mlle. MIMOUNI Sarah
Mr. BOUSSAÏDI Mohammed El-Amine El-Djazaïri
pour l'obtention du diplôme de Master 2 en Informatique
Domaine : Informatique
Filière : Informatique
Option : Ingénierie du logiciel

Thème

La Mise en place d'un serveur d'authentification et d'un Pare-feu

Soutenu le :
Devant le jury composé de :

- Le président : Mme. BENSTETI
- Le rapporteur : Mlle. BOUSTIA
- Les examinateurs : Mr. BOUKABOU

Le promoteur : Mr. HEDDACHE Mohammed

Promotion 2010

Le résumé :

Le travail présenté dans ce mémoire porte sur la mise en place d'un système d'authentification et la mise en place d'un pare-feu. Notre travail est scindé en trois parties :

La première traite l'authentification prise en charge par le protocole RADIUS, la seconde représente l'autorisation prise en charge par le pare-feu noyau de LINUX "NETFILTER" conformément à la politique de sécurité décidée par l'entreprise utilisatrice, et enfin la dernière qui consiste en la capture des paquets pris en charge par l'outil JPCAP.

Ce système permet l'évaluation des politiques de sécurité de n'importe quels réseaux fonctionnels.

Mot clés : Sécurité informatique, RADIUS, LINUX, NETFILTER, JPCAP

Abstract:

The work presented in this memory concerns the installation of a system of authentification and the installation of a fire wall. Our work is divided into three parts: The first treats the authentification taken of load by the protocol RADIUS, the second represents the authorization taken of load by the fire wall core of LINUX "NETFILTER" in accordance with the security policy decided by the company user, and finally the last which consists of the capture of the packages dealt with by tool JPCAP.

This system allows the evaluation of the security policies of any functional networks.

Key words: Computer security, RADIUS, LINUX, NETFILTER, JPCAP

ملخص:

العمل المقدم في هذه المذكرة يتعلق بوضع و إنشاء نظام تعرف آلي و كذلك نظام صد . عملنا ينقسم إلى ثلاثة أجزاء: الأول يعالج التعرف الآلي بواسطة البروتوكول RADIUS، أما الثاني فيتعلق برخصة المرور المقدمة من طرف جدار الصد لنظام LINUX 'NETFILTER' وفقا لنظام الأمن لمقرر من طرف الشركة المستغلة. الجزء الثالث و الأخير يتعلق برصد الحزم المعالجة من طرف النظام JPCAP.

هذا النظام يسمح بتقييم السياسات الأمنية لأي شبكة معلوماتية قيد الاستعمال.

كلمات رئيسية: الأمن المعلوماتي، JPCAP, NETFILTER, LINUX, RADIUS.

REMERCIEMENTS

Louange à notre Seigneur « ALLAH » qui nous a doté de la merveilleuse faculté de raisonnement.
Louange à notre Créateur qui nous a incité à acquérir le savoir.
C'est à lui que nous adressons, en premier lieu, toute notre gratitude.

En second lieu, nous tenons à exprimer nos vifs remerciements et reconnaissances envers notre promoteur monsieur HEDDECHE Mohamed, de l'Université de Boumerdes, pour ses conseils et sa disponibilité.

Nous tenons aussi à remercier Monsieur BOUKABOU Mohamed pour son entière disponibilité, ses conseils, ses encouragements et sa rigueur tout au long de notre projet.

Nous remercions monsieur MESSIED Mohamed, Chef du Département Informatique, pour sa générosité et l'ensemble des enseignants pour la qualité de la formation qu'ils nous ont prodigué.

Nos remerciements s'adressent également à messieurs DABOUZ M'hamed, KASBADJI Réda, LATTEB Abdelhamid, et mademoiselle BELABIDI Faïza de la Direction Générale d'ALGERIE Télécom qui nous ont permis d'effectuer le stage et nous ont encadrés tout le long de notre stage dans leur entreprise.

Comme nous tenons à remercier messieurs BENYAHIA Jamel et BOUSAHOUA Mohamed ainsi que Samir d'ALGERIE Télécom pour leur disponibilité et leur soutien durant le stage.
Enfin que tous ceux qui nous avons involontairement oublié, et qui ont participé de près ou de loin à la réussite de ce travail, nous en excusent et trouvent ici l'expression de notre gratitude.

Nous remercions également madame la présidente du jury ainsi que tous les membres de notre jury qui nous ont honorés de leur présence et leurs jugements constructifs.

SOMMAIRE

CHAPITRE IV

CHAPITRE V

CHAPITRE VI

LISTE DES FIGURES

LISTE DES TABLEAUX

Introduction générale

I. INTRODUCTION GÉNÉRALE

Au cours des dernières années, les systèmes d'information (SI) ont acquis dans tous les domaines professionnels une importance capitale, grâce à la simplicité de leurs usages et du nombre de tâches qu'ils permettent de réaliser. Cependant, l'importance acquise par ces systèmes les transforme également en cibles potentielles pour des attaques et des tentatives d'intrusions de plus en plus sophistiquées.

D'autre part, la plupart des entreprises ont actuellement tendance à faire confiance aux utilisateurs à cause des exigences du marché et du besoin d'améliorer leur productivité et leur compétitivité en restreignant l'utilisation de leurs équipements et leurs ressources à l'infrastructure du réseau local.

Etant donné que cette situation pourrait favoriser les attaques des systèmes d'information, ces mêmes entreprises se doivent de protéger leur S.I contre les anomalies de fonctionnement provenant soit d'une manœuvre intentionnellement malveillante d'un utilisateur, soit d'une faille technique rendant le système vulnérable.

A cet effet, notre entreprise d'accueil "Algérie Télécom" nous a proposé un sujet de Recherche et Développement ayant pour objet de sécuriser l'accès à ses ressources. Ce sujet qui fera l'objet de notre projet de fin d'études portera sur **"la mise en place d'un Serveur d'authentification et d'un Pare- feu"**.

Notre projet sera constitué principalement de deux (02) parties :

- La première partie concernera le volet théorique et renfermera l'étude sur l'architecture des systèmes d'authentification et les principes de filtrage,
- La seconde partie concernera le volet pratique et consistera en la conception de la solution, la réalisation du prototype, son intégration aux systèmes en exploitation et enfin son évaluation technique pour apprécier les résultats et éventuellement proposer des améliorations.

La première partie sera précédée par la présentation de la problématique et les objectifs attendus à la fin de ce travail.

Enfin, notre travail sera clôturé par une conclusion sur les résultats de nos travaux et les perspectives qui peuvent découler de ces mêmes résultats.

II. PRÉSENTATION D'ALGÉRIE TELECOM (A.T)

ALGERIE TELECOM (A.T), qui est notre entreprise d'accueil pour la réalisation de ce projet, a le statut d'une entreprise publique économique sous la forme juridique d'une société par actions SPA opérant sur le marché des réseaux et services de communications électroniques.

L'organigramme détaillé de l'entreprise est présenté en annexe 1.

III. PROBLEMATIQUE ET OBJECTIFS A ATTEINDRE

III-1 Problématique

L'entreprise nationale ALGERIE TELECOM (A.T) qui dispose d'un grand parc informatique et qui accorde un intérêt primordial pour le contrôle d'accès à ces ressources a relevé qu'elle ne dispose d'aucun moyen lui permettant de s'assurer que la personne qui se connecte est réellement celle qu'elle prétend être. L'absence d'un tel système rend ses équipements réseaux vulnérables aux différentes menaces qui peuvent nuire au bon fonctionnement du système. Pour ce faire la question fondamentale à laquelle il faut nécessairement répondre est la suivante:

« Quelles sont les principales fonctionnalités que doit comprendre un système d'authentification et un mécanisme de filtrage, afin que ces deux dernières soient fiables et conformes aux exigences de sécurité au niveau des réseaux de l'entreprise Algérie Telecom ? »

III-2 Objectifs à atteindre

Ainsi, il devient nécessaire, pour cette entreprise de se protéger des intrusions réseaux en installant un dispositif de protection. Un système d'authentification et un mécanisme qui renforcent la sécurité permettant de se protéger et se prémunir contre les intrusions, en filtrant les données entrantes et sortantes. Pour résoudre ce problème, Algérie Telecom nous a fixé les objectifs suivants :

1. la mise en place d'une politique de sécurité informatique d'accès efficace pour simplifier la gestion des droits du personnel d'Algérie TELECOM, en regroupant ses applications et ses services par famille avec des procédures sécurisées,
2. le déploiement d`un serveur d`authentification pour la gestion d`accès aux services d`A.T,

3. la mise en place d'un mécanisme de filtrage basé sur une politique de droit d'accès pré établi par A.T.

Pour cela, le plan de travail retenu pour atteindre ses objectifs s'articule autour du traitement des points suivants:

1. L`état de l`art sur la sécurité informatique,
2. L'étude sur le système d`authentification RADIUS,
3. L'étude sur les (pare-feux) firewalls,
4. Les outils de comptabilisation,
5. La conception da la solution,
6. La réalisation du système et son application sur le réseau d`Algérie Telecom,
7. L'évaluation du système réalisé et l'analyse des résultats.

La première partie, théorique, traitera les points 2, 3 et 4 cités précédemment.

La seconde partie, pratique, renfermera les points 5, 6 et 7 cités précédemment.

Partie Théorique

Chapitre I

Introduction à la sécurité Informatique

I. L`ETAT DE L`ART SUR LA SECURITE INFORMATIQUE

A travers la problématique et les objectifs retenus, il apparaît clairement que les questionnements fondamentaux pivotent autour de la vulnérabilité des systèmes informatiques et que les solutions traiteront cette vulnérabilité et renforceront par la même voie, cette même sécurité recherchée. Pour cela, il parait important d'introduire les différents reliefs de la sécurité.

I-1. La Sécurité Informatique [1]

Le terme "Système Informatique, (S.I)" désigne, dans notre contexte, tout système dont le fonctionnement est destiné à élaborer, traiter, stocker, acheminer ou présenter de l'information. Les systèmes d'information s'appuient, en règle générale, sur des systèmes informatiques pour leur mise en œuvre. De tels systèmes se prêtent à des menaces de types divers, susceptibles soit, d'altérer ou de détruire l'information (intégrité), soit de la révéler à des tiers qui ne doivent pas en avoir connaissance (confidentialité), soit de porter atteinte à sa disponibilité (disponibilité).

L'accès rapide aux informations, la rapidité et l'efficacité des traitements, les partages de données et l'interactivité ont augmenté de façon considérable mais c'est également le cas des pannes, des indisponibilités, des incidents, des erreurs, des négligences et des malveillances, en particulier avec l'ouverture sur Internet où de plus en plus d'entreprises ouvrent leur système d'information à leurs partenaires ou leurs fournisseurs.

Il est donc très important, essentiel même, de connaître les ressources de l'entreprise à protéger et de maîtriser le contrôle d'accès et les droits des utilisateurs du système d'information, particulièrement avec l'ouverture de l'accès de l'entreprise sur internet.

Par ailleurs, avec le nomadisme, consistant à permettre aux personnels de se connecter au système d'information à partir de n'importe quel endroit, les personnels sont amenés à « transporter » une partie du système d'information hors de l'infrastructure sécurisé de l'entreprise.

Afin de pouvoir sécuriser un système, il est nécessaire d'identifier les menaces potentielles, et donc de connaître et de prévoir la façon de procéder de l'ennemi.

Ainsi, la sécurité informatique, d'une manière générale, consiste à assurer l'utilisation des ressources matérielles ou logicielles d'une organisation uniquement dans le cadre prévu.

I-1-2 Objectifs de la sécurité informatique

Dans l'entreprise, La sécurité informatique doit remplir certains objectifs, qui se traduisent par un ensemble de services ; ils représentent les fondements et les pièces maîtresses de la sécurité sur les S.I. Les différents services assurés par la sécurité informatique sont:

1. **L'intégrité :** L'intégrité des données signifie que l'information ne peut être modifiée que par les personnes autorisées ou seulement par les moyens autorisés. L'intégrité reste un domaine très large couvrant à la fois les modifications, les moyens de modification mais également l'après modifications et donc la consistance contre les fraudes actives (brouillage, modification des données ou de l'identité, déguisement en émission ou en réception). Ce service détecte les altérations partielles ou intégrales des données entre l'émetteur et le récepteur.
2. **La confidentialité :** La confidentialité des données représente le fait que les données informatiques ne sont accessibles que par les personnes autorisées. L'objectif de ce service est d'empêcher les données d'être compréhensibles par une entité tierce non autorisée.
3. **La disponibilité :** La disponibilité se reflète dans l'information et dans les services. Ces derniers permettent d'assurer qu'un objet soit accessible et utilisable sur demande par une entité autorisée.
4. **La non répudiation de l'information :** Un mécanisme de non répudiation permet d'empêcher une personne de nier le fait qu'elle a effectué une opération. En fait, c'est surtout le récepteur qui est visé, il ne peut pas « répudier » son message, c'est-à-dire « qu'il ne l'a pas reçu ».
5. **L'authentification :** Le service d'authentification assure que le message provient de l'endroit d'où il prétend venir. Dans le cas d'un échange bidirectionnel, deux aspects sont présents. Il faut s'assurer que les deux entités sont bien celles qu'elles affirment être. De plus, le service d'authentification doit montrer que la connexion ne peut pas être brouillée par une troisième entité essayant de se faire passer pour l'un des deux correspondants (l'homme ou le milieu).

I-1-3 Politique de la Sécurité Informatique [2]

La politique de sécurité définit un certain nombre de règles, de procédures et de bonnes pratiques permettant d'assurer un niveau de sécurité conforme aux besoins de

l'organisation. En d'autres termes c'est « **décider entre ce qui est autorisé et ce qui ne l'est pas** ». A cette politique viennent bien entendu s'ajouter les lois et les règlements en vigueur, qui s'imposent à tous.

Dans notre cas, la sécurité des Systèmes Informatiques s'attelle généralement à garantir les droits d'accès aux données et ressources d'un système en mettant en place des mécanismes d'authentification et de contrôle permettant d'assurer que les utilisateurs des dites ressources possèdent uniquement les droits qui leurs ont été octroyés. Pour atteindre un bon niveau de sécurité, il faut l'aborder à travers plusieurs approches

- L'approche préventive cœrcitive : Obliger les usagers à utiliser la sécurité,
- L'approche préventive analytique : Identifier les menaces,
- L'approche curative : Enregistrer tout ou partie des actions accomplies sur le Système Informatique pour pouvoir l'analyser.

I-1-4 Nécessité d'une approche globale

Le niveau de sécurité d'un système est caractérisé par celui du maillon le plus faible. Cela signifie que la sécurité doit être abordée dans un contexte global notamment par :

- **La sensibilisation des utilisateurs** : Une politique de sécurité doit se faire avec les utilisateurs. Ces derniers doivent comprendre cette politique et respecter un certain nombre de règles en relation avec elle.
- **La sécurité logique** : C'est-à-dire la sécurité au niveau des données, en utilisant la cryptographie pour la confidentialité des informations et la signature électronique et le contrôle d'accès aux ressources.
- **La sécurité des applications** : Elle peut se faire avec les logiciels anti-virus (deux sur trois des attaques sont causées par des virus) et les programmes de tests de vulnérabilité et d'erreurs de configuration.
- **La sécurité physique** : La sécurité au niveau des infrastructures matérielles qui va de la fermeture des portes à clefs jusqu'à la mise en place d'une garde armée.

I-1-5 Etapes types d'établissement d'une politique de sécurité

Lorsqu'une défaillance apparaît dans un Système Informatique, celui-ci devient vulnérable. Dans ce cas, une intrusion peut facilement réussir. De là, l'établissement d'une politique de sécurité est nécessaire.

La mise en œuvre de la politique de sécurité se fait selon les quatre étapes suivantes :

1. Identifier les besoins en termes de sécurité, les risques informatiques pesant sur l'entreprise et leurs éventuelles conséquences,
2. Elaborer des règles et des procédures à mettre en œuvre dans les différents services de l'organisation pour les risques identifiés,
3. Surveiller et détecter les vulnérabilités du système d'information et se tenir informer des failles sur les applications et matériels utilisés,
4. Définir les actions à entreprendre et les personnes à contacter en cas de détection d'une menace.

I-2 Les Risques [3]

La Sécurité Informatique vise à se protéger contre les risques liés à l'informatique, pouvant être fonction de plusieurs éléments :

- Les menaces qui pèsent sur les actifs à protéger,
- Les vulnérabilités de ces actifs,
- La sensibilité des actifs, qui est la conjonction de différents facteurs (Disponibilité, Intégrité et confidentialité).

Le risque est le produit des trois éléments cités. Si l'un des éléments est nul, le risque n'existe plus. L'équation des risques est généralement représentée par :

Risque = (Menace x Vulnérabilité) / Contres mesures

I-2-1 Menaces

Les menaces peuvent être vues comme des transgressions potentielles de la sécurité. Les motivations des menaces envers un système de communication de données peuvent être de différentes sortes tels que :

- Obtenir un accès au système ,
- Voler des informations (secrets industriels ou propriétés intellectuelles),
- Recueillir des informations personnelles sur un utilisateur,
- Récupérer des données bancaires,
- S'informer sur l'organisation,
- Troubler le bon fonctionnement d'un service,
- Utiliser le système de l'utilisateur comme « rebond » pour une attaque,

- Utiliser les ressources du système de l'utilisateur, notamment lorsque la bande passante du réseau utilisé est élevée.

Il existe plusieurs critères de classification des menaces (nature, objectif, etc.) ; une des classifications est faite selon leurs natures :

✓ **Les menaces accidentelles**

Ce sont des faits involontaires ou erreurs pouvant constituer une menace. Des exemples de menaces accidentelles sont : les catastrophes naturelles (feu, inondation, etc.), les actes humains involontaires (mauvaise entrée de données, erreur de frappe ou de configuration), les performances imprévues des systèmes (erreur de conception dans le logiciel ou matériel, erreur de fonctionnement dans le matériel)

✓ **Les menaces délibérées**

Les menaces délibérées peuvent aller de l'examen déroutant, utilisant des outils de contrôle facilement disponibles, aux attaques sophistiquées utilisant une connaissance spéciale du système. Les menaces délibérées peuvent être passives ou actives.

- Les menaces passives sont celles qui, si elles se réalisent, ne produiraient aucune modification d'informations contenues dans le système et avec lesquelles ni le fonctionnement, ni l'état du système ne change. Il est très difficile de détecter ce type de menaces car elles sont inoffensives par rapport aux fonctions normales du système.
- Les menaces actives ou attaques envers un système comprennent l'altération d'informations contenues dans ce système, ou des modifications de l'état ou du fonctionnement du système. Les menaces actives sont, contrairement aux menaces passives, plus faciles à détecter si des précautions appropriées ont été prises au préalable. Les exemples d'attaques sont la destruction, la modification, la fabrication, l'interruption ou l'interception de données. Le résultat d'une attaque peut être une divulgation de l'information, une violation de la confidentialité de l'objet, une modification des objets, une violation de l'intégrité de l'objet, un déni de service , une violation de la disponibilité.

I-2-2 Vulnérabilités

Les vulnérabilités représentent les failles ou les faiblesses des entités qui composent ou interagissent avec le Système Informatique. Elles sont susceptibles d'être

exploitées par des éléments menaçants, utilisant une méthode d'attaque pour consulter, détruire, usurper ou modifier un bien.

Une vulnérabilité est une propriété intrinsèque d'une entité. Une même vulnérabilité peut exister pour plusieurs entités, en général, de type similaire. Cependant, une vulnérabilité peut s'appliquer seulement à une entité donnée, voire à un exemplaire particulier d'une entité. Une même vulnérabilité peut être exploitée dans le cadre de plusieurs méthodes d'attaques.

Il existe plusieurs critères de classification des vulnérabilités (par moment de leur introduction (réalisation, installation, et environnement), par faiblesse technique, par origine (intentionnelle ou non). Nous allons définir les différentes vulnérabilités, du point de vue de leurs origines. On distingue les grands types de vulnérabilités suivants :

- ✓ **Les vulnérabilités de conception**

Elles résultent d'un choix initial du concepteur (choix d'une technologie) ; ces vulnérabilités ne peuvent pas être supprimées sans remettre en cause l'entité elle – même.

- ✓ **Les vulnérabilités de réalisation**

Elles résultent des principes de fabrication (mauvais codage) ; ces vulnérabilités peuvent être réduites ou supprimées par des opérations correctrices à la charge du réalisateur.

- ✓ **Les vulnérabilités liées aux conditions d'emploi des entités,**

Il peut s'agir de l'environnement ou du processus d'installation (mauvais paramétrage) ; ces vulnérabilités peuvent être réduites ou supprimées par des opérations correctrices à la charge de la mise en œuvre.

- ✓ **Les vulnérabilités liées à l'usage des entités**

Les entités sont les ensembles qui composent ou interagissent avec le Système Informatique; elles peuvent être réduites ou supprimées par une action au niveau des utilisateurs.

I-2-3 la contre-mesure

C'est l'ensemble des actions mises en œuvre en prévention de la menace. Les contre-mesures à mettre en œuvre ne sont pas uniquement des solutions techniques mais

également des mesures de formation et de sensibilisation à l'intention des utilisateurs, ainsi qu'un ensemble de règles clairement définies.

Ainsi, le risque est l'effet néfaste de l'exploitation d'une vulnérabilité ou de la réalisation d'une menace. Dans certains contextes, le risque se mesure par sa probabilité d'occurrence ou par les conséquences quantifiées qui suivent sa réalisation. Donc, le risque mesure le coût de succès d'une attaque.

Une fois que toutes les étapes citées seront réalisées, il sera possible de mettre en place les solutions techniques appropriées à la défense du périmètre selon la politique choisie. Mais il est, bien entendu, évident que les dispositifs techniques ne pourront jamais résoudre tous les problèmes de sécurité.

Le système de sécurisation recherché au niveau d'AT repose sur trois niveaux

1. Le premier qui traite l'aspect « **Authentification** » va permettre de comprendre comment se fait l'accès au réseau à distance et qu'elle est le meilleur moyen d'authentification pour s'assurer que la personne qui se connecte est réellement celle qu'elle prétend être.
2. Le deuxième qui traite « **le filtrage**» va permettre de comprendre le principe de **l'autorisation** et ainsi le fonctionnement de NETFILTER.
3. Le troisième qui traite « **la comptabilisation** » sera le niveau où on va introduire l'outil qui sera utilisé dans notre projet « JPCAP ».

Ces niveaux seront présentés dans les trois chapitres suivants.

Chapitre II

L'AUTHENTIFICATION

II. LES PROTOCOLES AAA [4], [5]

L'accès à n'importe quel réseau se fait traditionnellement depuis le domicile, l'université, le bureau, ou les salles de conférence. Dans chacun de ces cas, la station d'accès est un équipement fixe ou éventuellement mobile dans une faible mesure (inférieur à 100 mètres pour l'Ethernet sans fil). Avec le déploiement des mobiles, il est devenu nécessaire de développer des protocoles permettant à des utilisateurs de se déplacer de réseau en réseau.

Les protocoles AAA (Authentication, Authorization, Accounting) qui permettent aux opérateurs d'authentifier des utilisateurs, de leur autoriser certains services et de collecter des informations sur l'utilisation des ressources sont présentés ci après.

II-1 Les concepts

Authentification (Authentification) (A) : Qui me parle ? ; C'est authentifier l'identité du client,

Autorisation (Authorization) (A) : Quelles autorisations lui accorder ? ; accorder des droits au client,

Comptabilisation (Accounting) (A) : Que fait le client ? ; enregistrer les données de comptabilité de l'usage du réseau par le client,

En pratique, les serveurs **"AAA"**, dans les domaines mère et visités, permettent de gérer les utilisateurs. Les clients **"AAA"** sont hébergés sur des routeurs ou sur des serveurs d'accès au réseau.

Les protocoles implémentant du **"AAA"** sont essentiellement utilisés par des opérateurs offrant des services de télécommunications à des utilisateurs. Ces protocoles leur permettent de contrôler l'accès à leurs réseaux et de connaître l'utilisation de leurs ressources. Ils peuvent ainsi facturer selon le temps de connexion ou selon la quantité d'informations téléchargées. La **figure 1** représente le schéma de l'architecture AAA la plus commune.

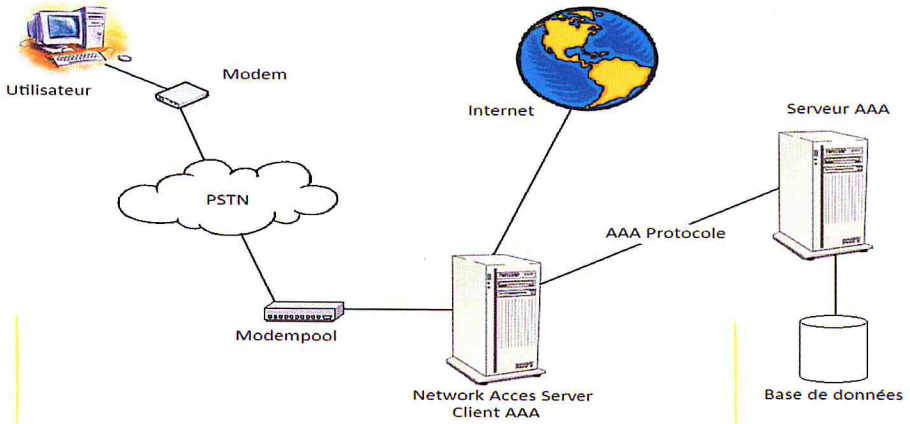

Figure 1: L'architecture AAA

II-2 Les protocoles

Il existe quatre protocoles AAA : TACACS, TACACS+, RADIUS et DIAMETER.

✓ **Le protocole TACACS + [6]**

Le protocole TACACS+ (Terminal Access Controller Access Control System Plus) est un protocole de sécurité inventé à la fin des années 90 par CISCO Systems. Même s'il a fini par remplacer les protocoles TACACS et XTACACS, TACACS+ n'est pas basé sur ces derniers. Ce protocole se situe au niveau de la couche transport. Il utilise le port 46 via le protocole TCP.

TACACS+ permet de vérifier l'identité des utilisateurs distants mais aussi, grâce au modèle AAA, d'autoriser et de contrôler leurs actions. Contrairement au protocole Radius qui fait l'authentification et l'autorisation en même temps, TACAS+ sépare les deux derniers.

Les protocoles RADIUS et DIAMETER sont de dernière génération.

✓ **Le protocole RADIUS [7]**

Le protocole RADIUS était, initialement, destiné pour répondre aux problèmes d'authentification pour des accès distants, par liaison téléphonique, vers les réseaux des fournisseurs d'accès ou des entreprises. C'est de là qu'il tient son nom qui signifie "**Remote Access Dial In User Service**". Au fil du temps, il a été enrichi et on peut envisager aujourd'hui de l'utiliser pour authentifier les postes de travail sur les réseaux locaux, qu'ils soient filaires ou sans fil.

C'est un protocole d'authentification standard dont le fonctionnement est basé sur un système "client/serveur" chargé de définir les accès d'utilisateurs distants à un réseau. Ce protocole repose principalement sur le serveur RADIUS, relié à une base d'identification (base de données, annuaire LDAP, etc.) et un client RADIUS, appelé **NAS** (Network Access Server), faisant office d'intermédiaire entre l'utilisateur final et le serveur. Il n'y a jamais de communication directe entre le poste de travail et le serveur. Le serveur RADIUS peut faire office de proxy, c'est-à-dire transmettre les requêtes du client à d'autres serveurs RADIUS.

✓ **Le protocole DIAMETER [8]**

Le protocole DIAMETER successeur du protocole RADIUS est un protocole AAA. Son nom provient d'un jeu de mot, signifiant diamètre en anglais, qui est le double du rayon (radius en anglais).

Le protocole DIAMETER a été conçu comme une version améliorée du protocole RADIUS. C'est la génération suivante du protocole RADIUS. Un des objectifs de cette conception était de maximiser la compatibilité et faciliter la migration de RADIUS à DIAMETER.

Le protocole DIAMETER est défini à travers un protocole de base et un ensemble d'applications. Cette conception permet une extension du protocole de base pour de nouvelles applications. Le protocole de base fournit des mécanismes pour un transport fiable, la livraison des messages et le traitement des erreurs.

Il était prévu d'utiliser, dans notre projet, le protocole DIAMETER pour l'authentification. L'opération d'installation du protocole OpenDiameter version 1.0.7-i a été réalisée avec succès, sauf que l'exécution du service du protocole OpenDiameter sur le système d'exploitation UBUNTU 9.10 n'a pas eu lieu pour erreur de pagination générée par la compilation du protocole lui-même.

L'erreur a été signalée au niveau du laboratoire responsable du développement du protocole « Diameter » à l'adresse suivante : mailto:diameter-developers@lists.sourceforge.net

```
Fichier Édition Affichage Terminal Onglets Aide
ibeaparchie.a ../../libdiametereap/.libs/libdiametereap.a ../../libdiameter/.libs/libdiameter.a ../../libdiamparser/.libs/lib
diamparser.a ../../libodutl/.libs/libodutl.a /usr/local/lib/libACEXML_Parser.so /usr/local/lib/libACEXML.so /usr/local/lib/li
bACE_SSL.so /usr/local/lib/libACE.so -lcrypto -lrt -ldl -lssl -Wl,--rpath -Wl,/usr/local/lib -Wl,--rpath -Wl,/usr/local/lib
make[2]: quittant le répertoire « /home/amine/opendiameter-1.0.7-i/applications/aaa »
Making all in pana
make[2]: entrant dans le répertoire « /home/amine/opendiameter-1.0.7-i/applications/pana »
depbase=`echo pacd.o | sed 's|[^/]*$|.deps/&|;s|\.o$||'`; \
        if g++ -DPACKAGE_NAME=\"OpenDiameter\" -DPACKAGE_TARNAME=\"opendiameter\" -DPACKAGE_VERSION=\"1.0.7-i\" -DPACKAGE_STR
ING=\"OpenDiameter\ 1.0.7-i\" -DPACKAGE_BUGREPORT=\"vfajardo@tari.toshiba.com\" -DPACKAGE=\"opendiameter\" -DVERSION=\"1.0.7-
i\" -DSTDC_HEADERS=1 -DHAVE_SYS_TYPES_H=1 -DHAVE_SYS_STAT_H=1 -DHAVE_STDLIB_H=1 -DHAVE_STRING_H=1 -DHAVE_MEMORY_H=1 -DHAVE_ST
RINGS_H=1 -DHAVE_INTTYPES_H=1 -DHAVE_STDINT_H=1 -DHAVE_UNISTD_H=1 -DHAVE_DLFCN_H=1 -DHAVE_IFADDRS_H=1 -DHAVE_SHADOW_H=1 -DHAV
E_BOOL=1 -DHAVE_STDBOOL_H=1 -DHAVE_GETIFADDRS=1 -I. -I.  -Wall -I/usr/local -I/usr -g -fno-strict-aliasing -I../../include -
I../../libodutl/include -I../../libdiamparser -I../../libeap/include -I../../libpana/include -I../common -I../../libpana/test
 -DOS_LINUX   -g -O2 -MT pacd.o -MD -MP -MF "$depbase.Tpo" -c -o pacd.o pacd.cxx; \
        then mv -f "$depbase.Tpo" "$depbase.Po"; else rm -f "$depbase.Tpo"; exit 1; fi
depbase=`echo pacd_config.o | sed 's|[^/]*$|.deps/&|;s|\.o$||'`; \
        if g++ -DPACKAGE_NAME=\"OpenDiameter\" -DPACKAGE_TARNAME=\"opendiameter\" -DPACKAGE_VERSION=\"1.0.7-i\" -DPACKAGE_STR
ING=\"OpenDiameter\ 1.0.7-i\" -DPACKAGE_BUGREPORT=\"vfajardo@tari.toshiba.com\" -DPACKAGE=\"opendiameter\" -DVERSION=\"1.0.7-
i\" -DSTDC_HEADERS=1 -DHAVE_SYS_TYPES_H=1 -DHAVE_SYS_STAT_H=1 -DHAVE_STDLIB_H=1 -DHAVE_STRING_H=1 -DHAVE_MEMORY_H=1 -DHAVE_ST
RINGS_H=1 -DHAVE_INTTYPES_H=1 -DHAVE_STDINT_H=1 -DHAVE_UNISTD_H=1 -DHAVE_DLFCN_H=1 -DHAVE_IFADDRS_H=1 -DHAVE_SHADOW_H=1 -DHAV
E_BOOL=1 -DHAVE_STDBOOL_H=1 -DHAVE_GETIFADDRS=1 -I. -I.  -Wall -I/usr/local -I/usr -g -fno-strict-aliasing -I../../include -
I../../libodutl/include -I../../libdiamparser -I../../libeap/include -I../../libpana/include -I../common -I../../libpana/test
 -DOS_LINUX   -g -O2 -MT pacd_config.o -MD -MP -MF "$depbase.Tpo" -c -o pacd_config.o pacd_config.cxx; \
        then mv -f "$depbase.Tpo" "$depbase.Po"; else rm -f "$depbase.Tpo"; exit 1; fi
/bin/sh ../../libtool --tag=CXX --mode=link g++  -g -O2 -L/usr/local/ace  -o pacd  pacd.o pacd_config.o -lssl -lcrypto -lACE_
SSL -lACE -lACEXML -lACEXML_Parser ../../libpana/libpana.la ../../libeap/libeap.la ../../libeap/libeaparchie.la ../../libdiam
parser/libdiamparser.la ../../libodutl/libodutl.la
mkdir .libs
g++ -g -O2 -o pacd pacd.o pacd_config.o  -L/usr/local/ace /usr/local/lib/libACE_SSL.so -lssl -lcrypto /usr/local/lib/libACEXM
L_Parser.so /usr/local/lib/libACEXML.so /usr/local/lib/libACE.so -lrt -ldl ../../libpana/.libs/libpana.a ../../libeap/.libs/l
ibeap.a ../../libeap/.libs/libeaparchie.a ../../libdiamparser/.libs/libdiamparser.a ../../libodutl/.libs/libodutl.a -Wl,--rpa
th -Wl,/usr/local/lib -Wl,--rpath -Wl,/usr/local/lib
make[2]: quittant le répertoire « /home/amine/opendiameter-1.0.7-i/applications/pana »
make[2]: entrant dans le répertoire « /home/amine/opendiameter-1.0.7-i/applications »
make[2]: Rien à faire pour « all-am ».
make[2]: quittant le répertoire « /home/amine/opendiameter-1.0.7-i/applications »
make[1]: quittant le répertoire « /home/amine/opendiameter-1.0.7-i/applications »
make[1]: entrant dans le répertoire « /home/amine/opendiameter-1.0.7-i »
make[1]: Rien à faire pour « all-am ».
make[1]: quittant le répertoire « /home/amine/opendiameter-1.0.7-i »
amine:/home/amine/opendiameter-1.0.7-i# aaad
Erreur de segmentation
amine:/home/amine/opendiameter-1.0.7-i#
```

L'installation du protocole OpenDiameter tentée à maintes reprises est présentée en annexe 2.

Après maintes tentatives toutes échouées, ALGERIE Télécom nous a recommandé l'utilisation du protocole RADIUS pour notre application.

Après la décision d'utiliser le protocole RADIUS pour notre application, nous allons détailler le fonctionnement de ce dernier.

II-2-1 Le protocole RADIUS [9]

Ses principales caractéristiques sont :

- Modèle Client/Server,
- Sécurité Réseau,
- Mécanismes Flexibles d'authentification,
- Protocole Extensible.

II-2-1-1 La méthode d'authentification

Pour pouvoir mettre en place une architecture s'adaptant à l'environnement existant, ou faire évoluer ce dernier afin d'obtenir une solution d'authentification plus complète et plus robuste, le traitement des deux questions suivantes s'imposent.

1. Que veut-on authentifier?

La question qui mérite d'être posée: Veut-on authentifier des utilisateurs ou des machines? En effet, selon la réponse, les choix de mise en œuvre seront différents.

- Si on authentifie des utilisateurs, cela signifie que l'authentification, et donc le VLAN (Virtual Local Area Network ou Virtual LAN, en français Réseau Local Virtuel) offert, sera dépendante de l'utilisateur qui exploite la machine. Une même machine sera connectée à tel ou tel VLAN suivant son utilisateur. Le VLAN est un réseau local regroupant un ensemble de machines de façon logique et non physique. Cela signifie donc que soit l'authentification (et donc l'établissement de la liaison réseau) se fasse au moment où chaque utilisateur se connecte dans sa session, soit c'est l'identité d'un seul utilisateur qui est considérée au moment du démarrage de.la machine.

- Si on authentifie des machines, cela signifie que quels que soient leurs utilisateurs, les machines seront toujours placées sur le même VLAN. La machine disposera alors d'une identité propre et unique qui peut être utilisée pour l'authentifier au moment du démarrage, ou au moment de la connexion d'un utilisateur.

2. Comment faire l'authentification?

De quels éléments dispose-t-on pour authentifier un utilisateur ou une machine ? L'authentification se fera avec l'un des trois moyens suivants :

- l'adresse Ethernet (adresse MAC),
- le certificat électronique,
- l'identifiant et mot de passe.

Ce dernier moyen (l'identifiant et mot de passe) a été choisi pour assurer l'authentification.

II-2-1-2 La mise en œuvre de l'authentification

Toute la difficulté de la mise en œuvre d'une solution d'authentification réside dans le fait qu'il s'agit d'un fonctionnement tripartite: les équipements réseau, le serveur d'authentification et le poste utilisateur (le client). Les moyens dont il faut disposer sont les équipements réseau, le serveur d'authentification et les postes clients.

✓ **Les équipements réseau**

L'élément pivot de tout le dispositif est l'équipement réseau, c'est-à-dire le commutateur. Dans la terminologie RADIUS, ces équipements sont appelés **NAS** (Network Access Server). Ils sont aussi nommés **clients RADIUS** et fonctionnent en tant que client **RADIUS**. Le client est responsable pour passer l'information de l'utilisateur vers les serveurs **RADIUS**, et puis d'effectuer les traitements en conséquence en fonction de la réponse qui est retournée puisque ce sont eux qui soumettent des requêtes au serveur.

✓ **Le serveur d'authentification**

Même avec la mise en œuvre de plusieurs protocoles, un seul serveur d'authentification sera nécessaire. Dans notre projet, ce sera le serveur d'authentification du type RADIUS qui sera utilisé. Les serveurs RADIUS seront chargés de (d'):

- recevoir des demandes de connexion d'utilisateur,
- authentifier l'utilisateur,
- renvoyer toute l'information de configuration nécessaire pour que le client puisse fournir le service à l'utilisateur.

Il faudra choisir le système d'exploitation et l'implémentation de RADIUS. Pour l'implémentation, il existe plusieurs solutions sur le marché, certaines sont commercialisées, d'autres libres. Citons à titre d'exemples :

- ACS (Access Control Server de Cisco sous Windows),
- Aegis (de MeetingHouse sous Linux),
- IAS (Internet Authentication Service de Microsoft sous Windows),
- Open RADIUS (libre sous Linux),

- Free RADIUS (libre sous Linux, BSD, Solaris et Windows) ; celle-ci servira à l'implémentation et sera détaillée dans le chapitre VI qui traite la partie concernant l'implémentation.

L'utilisation du système d'exploitation Linux nous a été recommandée par l'organisme d'accueil.

L'installation et la configuration de RADIUS seront abordées dans le chapitre VI "Implémentation ".

✓ **Poste clients**

Ce poste représente les utilisateurs.

II-2-1-3Le fonctionnement du système d'authentification

Le fonctionnement de RADIUS est basé sur un scénario qui se rapproche du cheminement suivant :

1. Le Client se connecte à NAS (équipement d'accès),
2. Le NAS demande au Client de s'authentifier,
3. Le Client donne son (identifiant + son mot de passe) au NAS,
4. Le NAS envoie une requête RADIUS au serveur,
5. Le Serveur RADIUS répond favorablement ou négativement,
6. Le NAS autorise le Client à se connecter ou rejettera sa demande,

Pour permettre la communication entre le serveur RADIUS et le NAS, les données sont échangées en paquets RADIUS.

✓ **Paquets RADIUS**

Les informations contenues dans un paquet RADIUS sont schématisées dans la **figure 2**

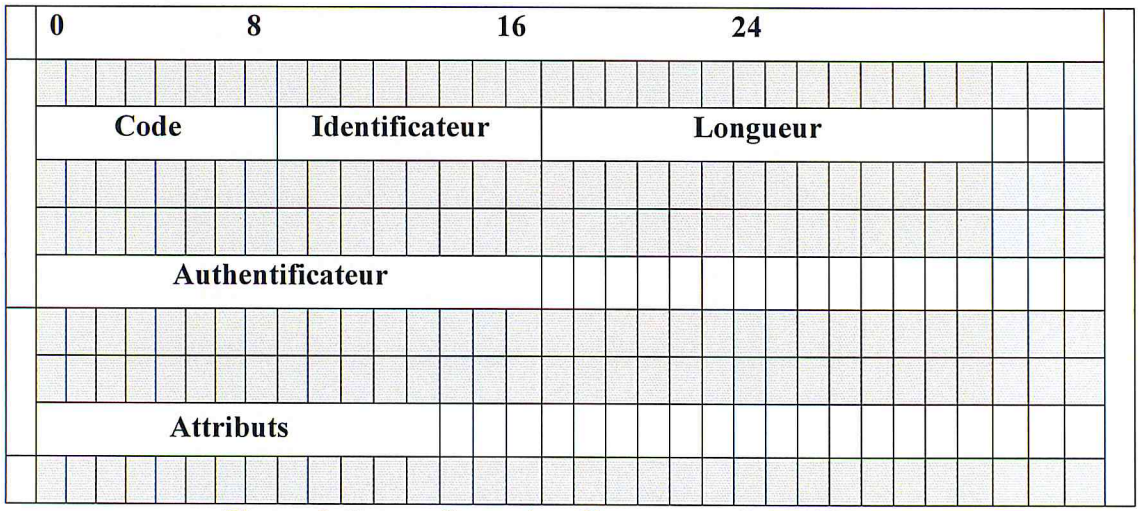

Figure 2: La configuration d'un paquet RADIUS

Le Code

Ce champ d'un seul octet contient une valeur qui identifie le type du paquet. La RFC définit 255 types de paquets. Cependant, trois d'entre eux seront suffisants pour les problèmes sujets de notre système. Il s'agit de :

- Access-Request (code=1) ;
- Access-Accept (code=2) ;
- Access-Reject (code=3) ;

✓ **L'identificateur, ID**

Ce champ, d'un seul octet, contient une valeur permettant au client RADIUS d'associer les requêtes et les réponses.

✓ **La longueur**

C'est le champ de seize octets contenant la longueur totale du paquet.

✓ **L'authentificateur**

Ce champ de seize octets a pour but de vérifier l'intégrité des paquets. On distingue l'authentificateur de requête et l'authentificateur de réponse. Le premier est inclus dans les paquets de type Access-Request envoyés par les NAS. Sa valeur est calculée de façon aléatoire. L'authentificateur de réponse est présent dans les paquets de réponse de type Access- Accept ou Access-Reject. Sa valeur est calculée par le serveur à partir d'une formule de hachage MD5 sur une chaîne de caractères composée de la concaténation des champs code, ID, longueur, authentificateur de requête et attributs ainsi que d'un secret partagé. Il s'agit d'un mot de passe connu à la fois par le serveur et le NAS. Ce dernier peut alors exécuter le même calcul que le serveur sur cette chaîne pour s'assurer qu'il obtient bien la valeur de l'authentificateur de réponse. Si c'est le cas, il peut considérer que la réponse lui vient bien du serveur auquel il a soumis la requête et qu'elle n'a pas été modifiée pendant la transmission.

✓ **Les Attributs et valeurs**

Ce champ du paquet est de longueur variable et contient la charge utile du protocole, c'est-à-dire les attributs et leur valeur qui seront envoyés soit par le NAS en requête, soit par le serveur en réponse.

Les attributs constituent le principe le plus important du protocole. Les champs attributs sont le fondement du protocole.

Chaque attribut possède un numéro d'attribut, auquel est associé un nom. La valeur d'un attribut peut correspondre à l'un des types suivants :

- adresse IP (4 octets),
- date (4 octets),
- Chaîne de caractères (jusqu'à 255 octets),
- Entier (4 octets),
- Valeur binaire (1 bit),
- Valeur parmi une liste de valeurs (4 octets).

Le nom de l'attribut n'est jamais présent dans les paquets. Seul son numéro apparaît. La correspondance entre un numéro d'attribut et son nom sera faite grâce à un dictionnaire.

Le dictionnaire d'attributs est une simple table qui permet de faire la correspondance entre leur numéro et leur nom respectif. Cette table contient trois champs par attribut le numéro d'attribut, son nom et son type.

Il existe un grand nombre d'attributs dans le protocole RADIUS, mais peu d'entre eux sont utiles dans notre cas, c'est-à-dire pour l'authentification sur réseau local. Nous allons définir les plus importants attributs que nous utiliserons dans notre système.

- **User-Name**

Dans le cas d'une authentification, cet attribut est envoyé par le NAS et contient l'identifiant qui va servir de point d'entrée dans la base du serveur d'authentification.

- **User-Password**

Il s'agit du mot de passe associé à User-Name, transmis par le NAS. Le serveur d'authentification valide ce mot de passe en fonction de la valeur enregistrée dans sa base de données.

- **Nas-IP-Address**

Il s'agit de l'adresse IP du NAS qui communique avec le serveur. Cet attribut est transmis par le NAS lui-même. Son utilisation permettra d'authentifier un poste de travail à la condition qu'il soit connecté sur un NAS qui possède cette adresse IP.

- **Nas-port**

Il s'agit du numéro de port du NAS sur lequel est connecté le poste de travail. Cet attribut est transmis par le NAS. Son utilisation permettra d'authentifier un poste de travail à la condition qu'il soit connecté sur ce numéro de port.

✓ **Les différents types de paquets**

L'authentification RADIUS se déroule suivant un dialogue entre le NAS et le serveur, qui met en jeu quatre types d'échanges. Chacun est véhiculé au moyen d'un paquet spécifique.

- **Access-Request**

  La conversation commence toujours par un paquet Access-Request émis par le NAS vers le serveur. Il contient au moins l'attribut User-Name et une liste d'autres attributs tels que Nas-Identifier, etc.

- **Access-Accept**

  Ce paquet est renvoyé au NAS par le serveur RADIUS si l'authentification transmise par l'Access-Request a été correctement validée.

- **Access-Reject**

  Ce paquet est renvoyé par le serveur RADIUS au NAS si l'authentification a échoué.

II-3 Les normes qui complètent le protocole RADIUS

Pour pouvoir bien comprendre tous les mécanismes mis en œuvre depuis le branchement d'un poste de travail sur le réseau, jusqu'au moment de son authentification, nous présenterons quelques protocoles d'authentification et quelques autres du mode de transport.

II-3-1 Les protocoles d'authentification [10]

Avant d'établir définitivement une session avec une entité distante, il convient de vérifier si cette entité est bien celle qu'elle annonce être.

Les protocoles se chargent de définir comment l'information est cryptée et comment fonctionne le mécanisme de « clé privée / clé publique » sur le réseau.

Cette authentification peut être assurée par l'un des protocoles suivants:

1. PAP (Password Authentification Protocole) : protocole classique avec utilisation d'un mot de passe statique,
2. CHAP (Challenge Handshake),

3. MSCHAP : la version CHAP de Microsoft pour ses réseaux Windows,
4. MS-CHAP V2 (Microsoft Challenge Handshake Authentication Protocol version 2).

Le quatrième protocole, baptisée MS-CHAP V2, qui sera utilisé dans notre système, est la version 2 du protocole MS-CHAP. Il a été défini en Janvier 2000 dans la RFC 2759. Cette nouvelle version du protocole définit une méthode dite « d'authentification mutuelle », permettant au serveur d'authentification et à la machine distante de vérifier leurs identités respectives. Le mécanisme d'authentification de cryptage a été mis à jour avec beaucoup plus de sécurité en particulier lorsque le nom d'utilisateur et mot de passe peuvent désormais être échangées avec la détermination des clés de chiffrement. Par conséquent, c'est un comportement où le client et le serveur, à la fois, s'authentifient mutuellement. En outre, il existe deux types de clés de chiffrement utilisées, l'une de l'envoi des données et l'autre de la réception des données.

Le processus d'authentification mutuelle de MS-CHAP V2 fonctionne de la manière suivante :

1. Initialement, le serveur d'authentification envoie à l'utilisateur distant une demande de vérification (ID) composée d'un identifiant de session et d'une chaîne aléatoire (un défi).
2. L'utilisateur distant utilise l'algorithme de hachage de réponse retour au défi de la chaîne du serveur NAS avec le cryptage, l'ID de session, son propre défi par les pairs et le mot de passe utilisateur. L'utilisateur distant répond avec :
   - son nom d'utilisateur,
   - un haché contenant la chaîne arbitraire fournie par le serveur d'authentification, l'identifiant de session ainsi que son mot de passe,
   - une chaîne aléatoire.
3. A l'étape suivante, le serveur NAS vérifie l'information de l'utilisateur distant et répond avec l'autre ID précisant la raison si cette connexion a été un succès ou un échec sur la base des informations comme le type de chiffrement négocié, et la décision sur la récusation serveur NAS (le mot de passe client à

fournir). Après la vérification de la réponse de l'utilisateur distant, il renvoie à son tour les éléments suivants :

- La notification de succès ou d'échec de l'authentification,
- Une réponse chiffrée sur la base de la chaîne aléatoire fournie par l'utilisateur distant ; cette réponse fournie est le mot de passe de l'utilisateur distant.

4. L'utilisateur distant vérifie enfin à son tour la réponse avec l'information qu'il a envoyée avant et se connecter au serveur NAS. L'utilisateur distant, en cas de réussite, établit la connexion.

Les trois premiers protocoles d'authentification précités sont présentés en annexe 3.

II-3-2 Les protocoles de transport

✓ Le protocole de transport UDP [11], [12]

L'UDP est un des principaux protocoles de télécommunication. Un paquet RADIUS est encapsulé dans un paquet **UDP (**User Datagram Protocol).

L'UDP est un protocole de transport (couche 4 du modèle OSI) qui est apparu avec le développement des réseaux locaux. Il sert à transmettre des données de manière très simple entre deux entités, chacune étant définie par une adresse IP et un numéro de port.

L'UDP est utilisé par les applications qui ne nécessitent pas le niveau de service du protocole TCP ou que souhaitent utiliser les services de communication qui ne disposent pas de TCP (par exemple, le multicast ou de livraison de diffusion). Il est aussi utilisé quand il est nécessaire soit de transmettre très rapidement des données, soit de transmettre des petites quantités de données, là où la connexion TCP serait trop lourde.

Contrairement au protocole TCP, il travaille en mode non connecté entre des processus s'exécutant sur des machines interconnectées sous IP (couche 3 du modèle OSI). C'est un protocole simple à mettre en œuvre, il est aussi très simple étant donné qu'il ne fournit pas de contrôle d'erreurs.

Les messages qu'envoie l'UDP sont appelés datagrammes, et chaque datagramme émis par UDP est encapsulé dans un datagramme IP en y fixant à 17 la valeur du protocole.

L'UDP, qui n'est pas indépendant d'IP gère donc :

- Un service de désignation s'appuyant sur les portes,
- Un service de détection d'erreur qui utilise une fonction prenant en considération la désignation complète de la source ainsi que les données transportées.

✓ **Structure d'un datagramme UDP**

Lors d'une transmission, les données traversent chacune des couches au niveau de la machine émettrice. A chaque couche, une information est ajoutée au paquet de données, il s'agit d'un **en-tête**, ensemble d'informations qui garantit la transmission.

Figure 3: Schéma d'un datagramme

A chaque niveau, le paquet de données change d'aspect, car on lui ajoute un en-tête, ainsi les appellations changent suivant les couches :

- Le paquet de données est appelé **message** au niveau de la couche Application **(données)**,
- Le message est ensuite encapsulé sous forme de **segment** dans la couche Transport **(paquet UDP)**,
- Le segment une fois encapsulé dans la couche Internet **(paquet IP)** prend le nom de **datagramme**
- Enfin, on parle de **trame** au niveau de la couche Accès réseau.

Au niveau de la machine réceptrice, lors du passage dans chaque couche, l'en-tête est lu, puis supprimé. Ainsi, à la réception, le message est dans son état original.

Le paquet UDP contient les 4 champs suivants :

Le Port Source : Il indique depuis quel port le paquet a été envoyé.

Le Port de Destination : Il indique à quel port le paquet doit être envoyé.

La Longueur : Elle indique la longueur totale du segment UDP (en-tête et données). La longueur minimale est celle de l'en-tête donc de 8 octets.

La Somme de contrôle : Celle-ci (de type checksum) permet de s'assurer de l'intégrité du paquet reçu. Elle est calculée sur l'ensemble de l'en-tête UDP et des données, mais aussi sur un pseudo en-tête (extrait de l'en-tête IP).

Port Source (16 bits)	Port Destination (16 bits)
Longueur (16 bits)	Somme de contrôle (16 bits)
Données (longueur variable)	

Figure 4: Schéma du segment (paquet) UDP

La Figure 5 décrit les champs utilisés pour le calcul de la somme de contrôle UDP. Les indices négatifs correspondent au pseudo en-tête IP.

<table>
<tr><th>+</th><th>Bits 0 - 7</th><th>8 - 15</th><th>16 - 23</th><th>24 - 31</th><th>Observation</th></tr>
<tr><td>-96</td><td colspan="4">Adresse Source</td><td rowspan="3">(paquet IP)</td></tr>
<tr><td>-64</td><td colspan="4">Adresse Destination</td></tr>
<tr><td>-32</td><td>Zéros</td><td>Protocole</td><td colspan="2">Taille UDP</td></tr>
<tr><td>0</td><td colspan="2">Port Source</td><td colspan="2">Port Destination</td><td rowspan="3">Segment (paquet) UDP</td></tr>
<tr><td>32</td><td colspan="2">Longueur</td><td colspan="2">Somme de contrôle</td></tr>
<tr><td>64</td><td colspan="4">Données</td></tr>
</table>

Figure 5: Schéma d'un datagramme

L'installation et la configuration du protocole FreeRadius est présentée en annexe 4.

Chapitre III

Le Filtrage

III. LE SYSTEME PARE FEU (FIREWALL) [13], [14]

Un pare-feu est un équipement réseau qui contrôle le trafic réseau au niveau transport ou inférieur. Il utilise les informations d'un paquet IP et celles du niveau protocolaire supérieur comme TCP ou UDP (ports source et destination) pour garantir le respect d'une politique de sécurité.

Le pare-feu est donc un système permettant de filtrer les paquets de données échangés avec le réseau.

III-1 Fonctionnement d'un système pare-feu

Un système pare-feu contient un ensemble de règles prédéfinies permettant de (d')

- autoriser la connexion (allow),
- bloquer la connexion (deny),
- rejeter la demande de connexion sans avertir l'émetteur (drop).

L'ensemble de ces règles permet de mettre en œuvre une méthode de filtrage dépendant de la politique de sécurité adoptée par l'entité. On distingue habituellement deux types de politiques de sécurité permettant :

- soit d'autoriser uniquement les communications ayant été explicitement autorisées,
- soit d'empêcher les échanges qui ont été explicitement interdits.

La première méthode est sans nul doute la plus sûre, mais elle impose toutefois une définition précise et contraignante des besoins en communication.

Pour filtrer les paquets on va utiliser Netfilter.

- **Netfilter [15], [16], [17]**

Netfilter est l'implémentation noyau du firewall sous Linux. Un certain codeur talentueux du nom de Rusty Russel a estimé qu'il était nécessaire d'avoir un remplaçant au fameux « IPFILTER » et de pouvoir rivaliser voir dépasser largement les firewalls commerciaux.

Netfilter permet de faire beaucoup plus de choses en matière de filtrage de paquets et de translation d'adresses que ses prédécesseurs, Il fournit ainsi les fonctions de pare-feu, comme l'autorisation du trafic réseau qui sera utile pour notre système ce qui fait de Linux 2.6 un outil de choix pour la réalisation de passerelles dans notre projet.

✓ **Le principe de fonctionnement**

Netfilter travaille sur des paquets réseaux. Il s'agit de parties des informations transmises. Par exemple, pour télécharger un fichier, celui-ci est découpé en

plusieurs paquets avant de transiter sur le réseau. Chacun de ces paquets comporte en plus des données, des informations ajoutées par les couches réseaux. Ce sont sur ces informations que s'effectueront les tests de filtrage.

La couche réseau Linux présente plusieurs points d'accès (en anglais hook). Netfilter dispose de fonctions de rappel (callback). Celles-ci sont des suites d'instructions qui précisent ce qui doit être fait lorsque survient un événement.

Concrètement, lorsqu'un paquet réseau atteint un de ces points d'accès, il est passé à Netfilter par l'intermédiaire de sa fonction de rappel. Il est alors examiné pour prendre une décision concernant son traitement futur. Netfilter se comporte comme un automate qui compare le paquet successivement à plusieurs règles. Et selon le résultat du test, le paquet est traité ou transmis au test suivant. La configuration de Netfilter sera présentée en annexe 5.

✓ **Les tables**

Aux points d'accès sont associées des **chaînes** de traitement. C'est dans celles-ci que sont effectués les tests. Elles sont regroupées par **tables** selon le type de traitement.

Les tables sont ajoutées par des modules. Il en existe 3 principales pouvant être utilisées. Voici ci-après leurs noms et la tâche à laquelle elles sont destinées.

Table	Description
Filter	Cette table permet de filtrer les paquets. Typiquement ce sera pour les accepter ou non.
Nat	Avec cette table, on peut réaliser des translations d'adresse (ou de ports). Ceci sera notamment utile pour partager une connexion.
Mangle	elle sert pour modifier les en-têtes des paquets. On la rencontrera parfois pour marquer des paquets afin que d'autres applications puissent les reconnaître.

Tableau 1: Les Tables de Netfilter

✓ **Les chaînes**

A l'intérieur d'une table, on peut trouver plusieurs chaînes. Ce sont elles qui contiendront les règles à appliquer aux paquets. Ces règles seront évaluées séquentiellement. On trouve deux types de chaînes. Tout d'abord celles qui sont associées aux différents points d'entrées existants. Un paquet atteignant un de ces

points sera envoyé vers la chaîne associée. Ce sont les fonctions de rappel évoquées précédemment qui réalisent cela. Elles effectuent les uns après les autres les tests que contient la chaîne. Ces chaînes sont en nombre fini et ne sont pertinentes que pour certaines tables. Le tableau suivant les liste, en indiquant quelle table a une chaîne de ce type.

Chaîne	Table	Description
PREROUTING	Nat, Mangle	Par cette chaîne passeront les paquets entrants dans la machine avant routage.
INPUT	Filter	Cette chaîne traitera les paquets entrants avant qu'ils ne soient passés aux couches supérieures (Les applications).
FORWARD	Filter	Ce sont les paquets uniquement transmis par la machine sans que les applications n'en aient connaissance.
OUTPUT	Filter, Nat, Mangle	Cette chaîne sera appelée pour des paquets envoyés par des programmes présents sur la machine.
POSTROUTING	Nat	Les paquets prêts a être envoyés (soit transmis, soit générés) seront prise en charge par cette chaine.

Tableau 2: Chaîne de Netfilter

✓ **Les cibles**

Enfin, le dernier élément important est la notion de cible. Il s'agit du traitement que l'on décide d'appliquer au paquet. C'est la cible qui se chargera de faire les opérations nécessaires. En plus de celles prédéfinies, il est possible d'indiquer comme cible une chaîne utilisatrice. Cela permet d'imbriquer différents tests et traitements.

Chaque chaîne peut être vue comme un ensemble de tests, chacun ayant pour résultat l'envoi du paquet vers la cible spécifiée si la condition est vérifiée. Si ce n'est pas le cas, on passe à la suivante. En arrivant à la fin d'une des chaînes du tableau précédent, une cible par défaut est utilisée. A la fin d'une chaîne utilisatrice, si aucune décision n'a été prise, on revient à la chaîne appelante.

Voici les cibles prédéfinies les plus courantes :

Cible	Description
ACCEPT	Les paquets envoyés vers cette cible seront tout simplement acceptés et pourront poursuivre leur cheminement au travers des couches réseaux.
DROP	Cette cible permet de jeter des paquets qui seront donc ignorés.
REJECT	Permet d'envoyer une réponse à l'émetteur pour lui signaler que son paquet a été refusé.
LOG	Demande au noyau d'enregistrer des informations sur le paquet courant. Cela se fera généralement dans le fichier /var/log/messages.
MASQUERADE	Cible valable uniquement dans la chaîne POSTROUTING de la table Nat. Elle change l'adresse IP de l'émetteur par celle courante de la machine pour l'interface spécifiée. Cela permet de masquer des machines et de faire par exemple du partage de connexion.
SNAT	Egalement valable pour la chaîne POSTROUTING de la table Nat seulement. Elle modifie aussi la valeur de l'adresse IP de l'émetteur en la remplaçant par la valeur fixe spécifiée.
DNAT	Valable uniquement pour les chaînes PREROUTING et OUTPUT de la table Nat. Elle modifie la valeur de l'adresse IP du destinataire en la remplaçant par la valeur fixe spécifiée.
RETURN	Utile dans les chaînes utilisateurs. Cette cible permet de revenir à la chaîne appelante. Si RETURN est utilisé dans une des chaînes de base précédente, cela est équivalent à l'utilisation de sa cible par défaut.

Tableau 3: Cibles prédéfinies

- **IPTABLE [15]**

Pour pouvoir configurer Netfilter et le manipuler, il existe l'outil IPTABLE, intégré la plupart du temps dans le noyau Linux, et qui est l'interface "ligne de commande".

IPTABLE est l'outil qui est fourni à l'administrateur pour agir sur tous les concepts vus précédemment et pour modifier les règles de filtrage donc.

La première option à connaître est "**-t**" qui permet de spécifier le nom de la table sur laquelle portera les autres paramètres. Si cette option n'est pas spécifiée, ce sera par défaut la table filter. On peut aussi demander à IPTABLE de charger un module particulier avec l'option

La deuxième option à connaître est "**-m**" ; Ce module peut ajouter de nouvelles tables ou de nouvelles manières de tester les paquets.

Il faut ensuite indiquer une commande pour indiquer par exemple qu'une nouvelle règle doit être ajoutée dans la chaîne spécifiée. Ci-après la liste des options les plus courantes pour spécifier une commande. Une seule à la fois peut être présente, et toutes devront être suivies du nom de la chaîne à prendre en compte.

Option	Rôle
-L	Affiche toutes les règles de la chaîne indiquée.
-F	Supprime toutes les règles de la chaîne. Si aucune chaîne n'est spécifiée, toutes celles de la table sont vidées.
-N	Crée une nouvelle chaîne utilisateur avec le nom passé en paramètre.
-X	Supprime la chaîne utilisatrice. Si aucun nom n'est spécifié, toutes les chaînes utilisatrices seront supprimées
-P	Modifie la politique par défaut de la chaîne. Il faut indiquer en plus comme paramètre la cible à utiliser.
-A	Ajoute une règle à la fin de la chaîne spécifiée.
-I	Insère la règle avant celle indiquée. Cette place est précisée par un numéro qui fait suite au nom de la chaîne. La première porte le numéro 1. Si aucun numéro n'est indiqué, la règle est insérée au début.
-D	Supprime une règle de la chaîne. Soit un numéro peut être précisé, soit la définition de la chaîne à supprimer (ses tests de concordance et sa cible).

Tableau 4: Options d'IPTABLE

Le schéma suivant décrit le fonctionnement de NETFILTER.

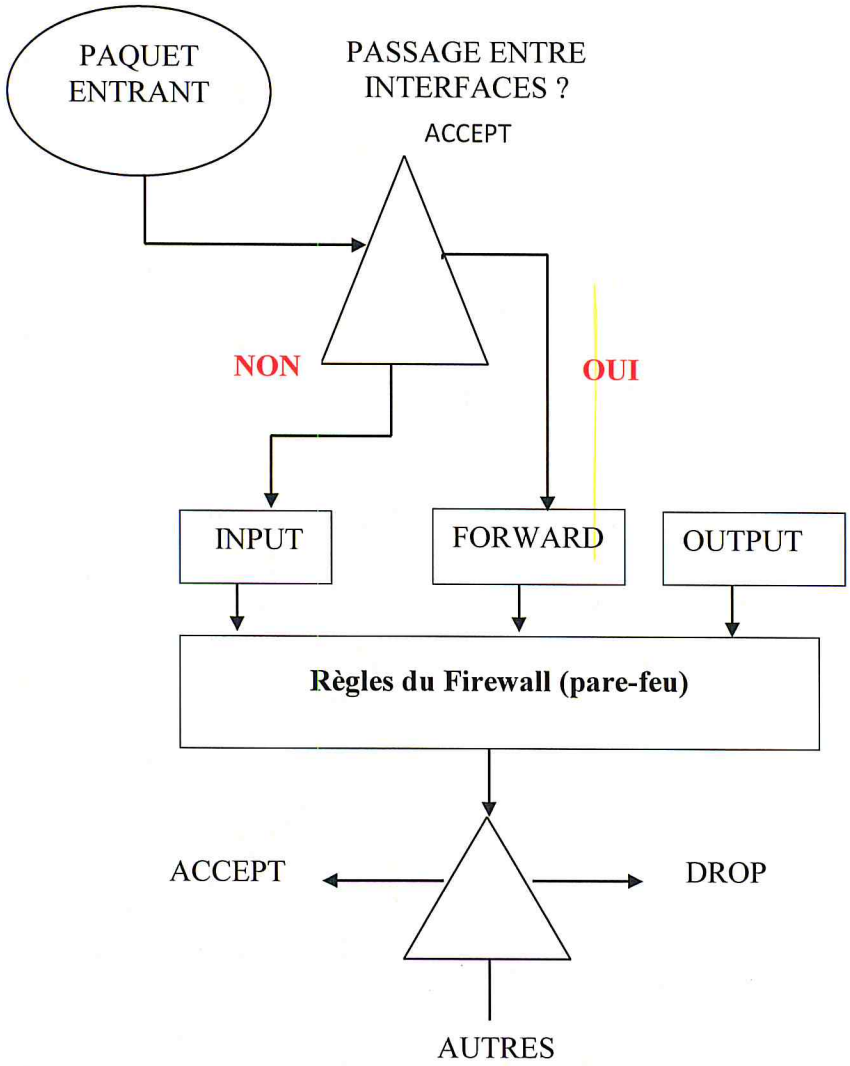

Figure 6: Le cheminement des paquets à travers les chaînes

Chapitre IV

La Comptabilisation

IV - 1 INTRODUCTION

Dans le monde de la Sécurité Informatique et de l'intrusion, nous avons tous entendu un jour ou l'autre parler de Snort, Ethereal ou même de TcpDump pour ne citer qu'eux.

Ces utilitaires qui sont des outils de capture et d'analyse de trames ont tous une particularité commune, ils utilisent la librairie Libpcap.

Ecrite en « C », cette librairie existe sous Windows et Linux et est conjointement utilisée par les développeurs réseau afin d'intercepter des paquets, débugger leur programme et par les administrateurs souhaitant forger leur propre paquets afin de tester la sécurité et la robustesse de leurs protocoles et équipements. Cependant la Libpcap présente un inconvénient majeur qui est qu'elle n'est écrite et exploitable qu'en « C » limitant ainsi son utilisation par des programmeurs confirmés adeptes de compilations croisées et obligés d'être au fait des divers soucis liés au portage de l'application d'un système vers un autre. Certes d'autres utilisateurs moins bien intentionnés utilisent également la librairie Libpcap à des fins néfastes.

C'est pourquoi le projet JPCAP a vu le jour. Ceci afin de pouvoir exploiter la libpcap dans un environnement de programmation relativement simple portable et puissant tel que le propose JAVA.

IV - 2 LES RESEAUX ET LES MODES DE CAPTURE DE PAQUETS

IV - 2 -1 Les domaines de collisions

Un réseau est scindé en domaine de collisions. Ce dernier est un segment du réseau dans lequel chaque trame émise est visible par tous les hôtes du domaine de collisions. Cependant seul l'ordinateur désigné comme destinataire de la trame prendra connaissance de la trame et la désencapsulera pour la passer à sa couche 3 du modèle OSI après consultation du champ adresse de destination de la trame codée sur 6 octets (48 bits). Fixée par les constructeurs d'interfaces réseaux, cette adresse est unique et est appelée adresse MAC ou adresse physique.

IV - 2 -2 Les différents modes de capture de paquets

La capture réseau que l'on peut également appeler « sniffing » consiste à relever chaque trame entrante et sortante au niveau de notre carte réseau et peut fonctionner selon deux schémas distincts :

1 - Ecoute des trames émises et reçues par notre ordinateur seul.

Ce mode de fonctionnement ne permet de voir que les trames issues et à destination de notre interface réseau. Nous pouvons ainsi analyser les données envoyées et reçues que ce soit pour du débug ou pour analyser la sécurité de notre ordinateur. Ce mode ne nécessite pas de configuration particulière de la carte réseau.

2 - Ecoute de toutes les trames émises sur notre domaine de collision.

Ce deuxième mode permet de voir toutes les trames émises sur notre segment du réseau.

Une telle écoute du réseau est interdite sur un réseau public. C'est ce genre de mode d'écoute qu'utilisent les IDS (Intrusion Detection System) tel que Snort. Ce mode d'écoute nécessite qu'on configure la carte réseau en mode « promiscuous » afin que cette dernière récupère toutes les trames qu'elle voit passer et non pas uniquement celles qui lui sont destinées.

IV - 2 -3 Le domaine d'application de « JPCAP » [18]

Jpcap est une bibliothèque open source pour la capture et l'envoi de paquets réseau à partir d'applications Java. Elle offre des installations à:

- Capturer des paquets bruts en direct du fil.
- Mettre les paquets capturés dans un fichier en mode hors connexion, et de les lire.
- identifier automatiquement les types de paquets et de générer des objets Java correspondant (pour Ethernet, IPv4, IPv6, ARP / RARP, TCP, UDP, et ICMPv4 paquets).

Jpcap est basé sur libpcap/WinPcap, et est implémenté en C et Java. Jpcap a été testé sur Microsoft Windows (98/2000/XP/Vista), Linux (Fedora, Ubuntu), Mac OS X (Darwin), FreeBSD et Solaris.

Le projet Jpcap, contrairement à certaines idées reçues, n'est absolument pas une alternative au projet Libpcap. La librairie Jpcap repose sur la librairie Libpcap et fournit au programmeur JAVA les « API » (Application Programming Interface) nécessaires à la capture et à la forge de paquets.
Deux versions de ce projet existent à ce jour portant tous les deux le même nom « Jpcap »: Une interface de programmation (API) est une interface fournie par un programme informatique. Elle permet l'interaction des programmes les uns avec les

autres de manière analogue à une interface homme-machine rend possible l'interaction entre un homme et une machine.

Du point de vue technique une API est un ensemble de fonctions, procédures ou classes mises à disposition par une bibliothèque logicielle, un système d'exploitation ou un service. La connaissance des API est indispensable à l'interopérabilité entre les composants logiciels.

1. Le premier de sourceforge, « http://sourceforge.net/projects/JPCAP » sous licence » Mozilla Public License » interdisant la fonctionnalité d'émission de paquets.
2. Le deuxième de Keita comportant moins de fonctionnalités et de mises à jour, « Fujii,http://netresearch.ics.uci.edu/kfujii/JPCAP/doc/index.html » mais disposant de la capacité d'émettre des paquets sur le réseau.

Concernant le domaine d'utilisation de Jpcap, il fonctionne au niveau de la couche 2 (Liaison de donnée) du modèle OSI (Open System Interconnection). La PDU (Protocol Data Unit) utilisée pour les données capturées est la trame. Les trames capturées peuvent contenir des paquets de type « arp, rarp, icmpv4 et IPv4&6 ».

Jpcap nous permet de manipuler des objets des types suivants :

- ARPPacket,
- EthernetPacket,
- ICMPPacket,
- IPPacket,
- TCPPacket,
- UDPPacket.

Jpcap possède également d'autres fonctionnalités, tels que la possibilité d'enregistrer les paquets capturés dans un fichier pour les analyser plus tard. C'est justement l'objectif dans notre projet.

L'installation de Jpcap sous LINUX est présentée en annexe 6.

CONCLUSION GENERALE DE LA PARTIE THEORIQUE

L'objectif de cette partie était de comprendre le système de sécurisation recherchée au niveau d'AT. Nous avons commencé par comprendre les différents concepts fondamentaux liés à la sécurité informatique. Nous avons ensuite défini le protocole d'authentification « RADIUS ». Nous avons constaté qu'il offre une solution fiable pour centraliser les informations d'authentification.

Concernant le fonctionnement du pare-feu « Netfilter », il est, actuellement, l'un des pare-feu (firewall) les plus puissants du marché ; il est souple, et très largement supporté par la communauté.

Nous avons fini par introduire l'outil Jpcap pour la capture des paquets car il apporte la puissance à la bibliothèque libpcap de la plateforme Java. Il est évident que la portabilité et la relative simplicité du langage Java en font l'instrument idéal pour la gestion des réseaux.

La deuxième partie, pratique, sera consacrée à la description globale et la conception de notre système.

Partie Pratique

Chapitre V

La Conception

La phase la plus importante, dans le processus de développement d'une application est la conception car elle fournit une architecture simple et détaillée de la solution d'un problème donné.

Dans ce chapitre, nous introduirons brièvement le langage de modélisation, à savoir UML et le processus de développement que nous avons adopté, à savoir le processus UP.

Nous détaillerons, ensuite, la plateforme à mettre en œuvre, tout en respectant le découpage en modules pour la flexibilité de notre système.

V-1 Présentation globale de la plateforme

L'ensemble de la plateforme sera décomposée sur plusieurs stations. Elle se compose de trois parties essentielles :

1. la première est celle qui représente un réseau (LAN1) dans l'entreprise à partir duquel nous simulons des tentatives de connexions,
2. la seconde partie concerne le réseau (LAN2) et qui sera par la suite la cible de nos tentatives de connexion,
3. la dernière partie concerne les machines à utiliser la première porte le serveur radius, la deuxième porte le pare- feu Netfilter et les deux seront reliés à un routeur qui, lui, joue le rôle du Nas (client radius).

La configuration matérielle du réseau de l'entreprise comporte un ensemble de stations reliées par un switch. La figure suivante illustre l'architecture globale du réseau.

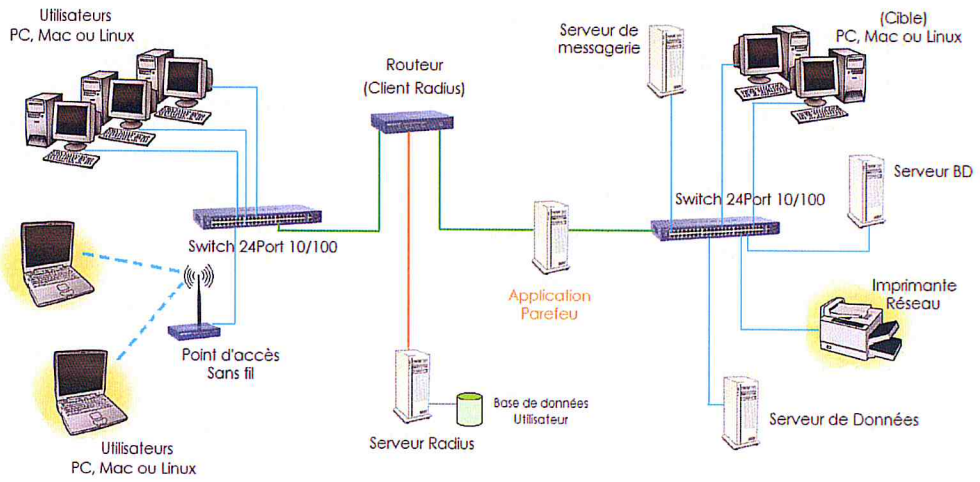

Figure 7: Architecture globale de la plate-forme

V-2 Conception de la base de données de la plate forme

Notre plate-forme est composée d'un ensemble de machines qui sont chacune caractérisée par un identifiant, un nom, une adresse IP et le type, Chacune de ces machines a des règles propres à elle caractérisées par un identifiant, adresse IP source, adresse IP destination, port source, port destination, type protocole et action qui lui permet de se connecter à des machines de type ressources. La figure suivante donne une description des différentes entités de la base.

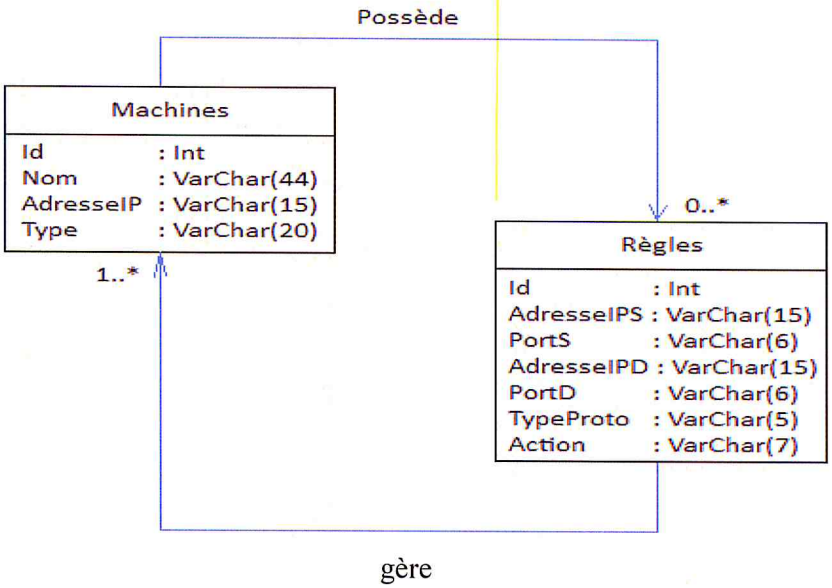

Figure 8 : Modèle conceptuel de données de la plate-forme

Architecture logicielle du système

Les objets de notre Système se regroupent en quatre principaux paquetages :

1. Application de paramétrage du pare- feu : elle représente le noyau du système qui est chargé de la gestion de la plate-forme,
2. Connecteur JDBC : il permet de se connecter a la base de données,
3. les logiciels de configuration,
4. Base de données MYSQL : elle est utilisée pour stocker les différentes informations de la plate-forme.

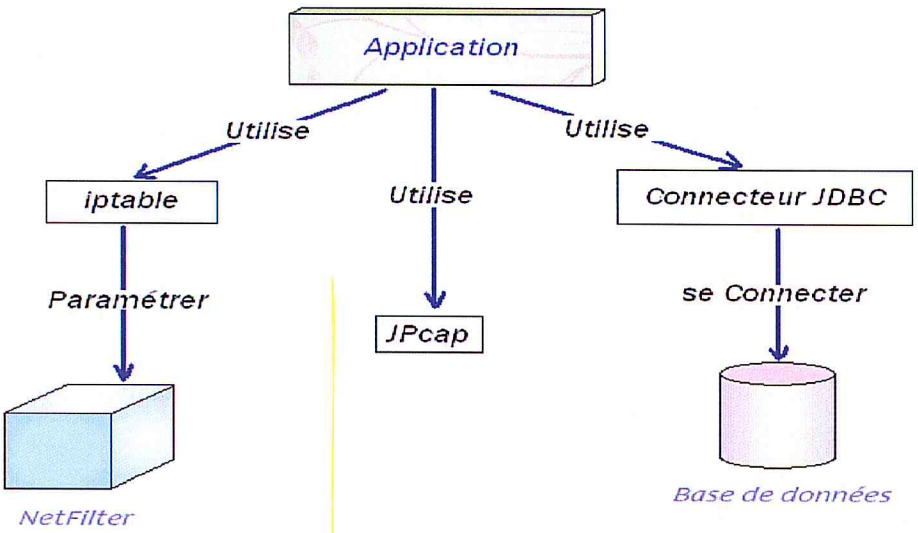

Figure 9: Architecture logicielle du système

Afin de mieux représenter la conception du système, nous l'avons modélisé en utilisant un langage de référence UML :

V-3 L'étude conceptuelle

V-3-1 Langage de modélisation UML [19], [20], [21]

L'approche objet est loin d'être une idée récente. Simula, premier langage de programmation à implémenter le concept de type abstrait à l'aide de classes, date de 1976 ; Smalltalk implémenta les concepts fondateurs de l'approche objet :

Encapsulation, agrégation, héritage, le début des années 80 a connu la naissance des premiers compilateurs c++.

Or, rien dans les concepts de base de l'approche objet ne dicte comment modéliser la structure objet d'un système de manière pertinente. De plus connaître C++ ou Java n'est pas une fin en soi, il faut aussi savoir se servir de ces langages à bon escient. La question est donc de savoir qui va nous guider dans l'utilisation des concepts objet, si ce ne sont pas les langages orientés objet ? Il faut aussi avoir un outil pour comparer deux solutions objet pour un système ?

Pour remédier à ces problèmes, il nous faut donc un langage qui doit permettre de représenter des concepts abstraits, de limiter les ambiguités (parler un langage commun indépendant des langages orientés objet) et simplifier la comparaison et l'évaluation de solutions.

Un langage unifié pour la modélisation a été développé : UML (Unified Modeling Language). Il s'agit d'un langage graphique de modélisation objet permettant de spécifier, construire, visualiser et décrire les détails d'un système logiciel. Il est adapté à la modélisation de tous types de systèmes.

✓ **Le modèle conceptuel d'UML**

Le modèle conceptuel d'UML comprend les notions de base génériques du langage. Il définit trois sortes de « briques » :

- Les éléments ; qui sont les abstractions essentielles à un modèle,
- Les relations ; qui constituent des liens entre ces éléments,
- Les diagrammes ; qui regroupent des éléments et des liens au sein de divers ensembles.

1. **Les éléments** : Il existe quatre types d'éléments dans UML :
   - Les éléments structurels (classe, interface, collaboration…) ;
   - Les éléments comportementaux (interaction, automate à états finis) ;
   - Les éléments de regroupement (package) ;
   - Les éléments d'annotation (note) ;
2. **Les relations** : Il existe aussi quatre types de relations dans UML : la dépendance, l'association, la généralisation et la réalisation.
3. **Les diagrammes** : UML exprime ses concepts à travers différents diagrammes graphiques qu'on classifie en deux grandes catégories :
   - Diagrammes statiques (diagrammes structurels) : diagrammes de classes, objets, de composants, de déploiements et de cas d'utilisation.
   - Diagrammes dynamiques (diagrammes comportementaux) : diagrammes d'activités, de séquences, d'états de transitions et de collaborations.

✓ **La représentation graphique**

Nous donnons, ci après, les différents diagrammes de la représentation UML :

- **Les diagrammes de classes :** ils illustrent la vue conceptuelle statique d'un système, avec ses package, ses classes, ses interfaces, ses collaborations et leurs relations respectives.

- **Les diagrammes d'objets :** ils fournissent des instantanés statiques des instances d'éléments figurant dans les diagrammes de classes, en particuliers du point de vue cas réels ou prototypiques.
- **Les diagrammes de composants :** ils montrent la vue d'implémentation statique d'un système ainsi qu'un ensemble de composants avec leurs relations. Un composant représente une implémentation physique des abstractions logiques d'un modèle, comme les classes et leurs interactions.
- **Les diagrammes de déploiement :** ils illustrent la connectivité des nœuds physiques dans une vue architecturale du système. Un nœud est une ressource de calcul fournissant un environnement d'exploitation physique pour l'exécution d'un ou de plusieurs composants.
- **Les diagrammes de cas d'utilisation :** ils modélisent le comportement d'un système ou d'une classe en expliquant les relations existantes entre un ensemble de cas d'utilisation et leurs acteurs.
- **Les diagrammes d'activités :** ils présentent le flot d'activité d'un système, y compris le flot séquentiel ou de branchement d'activité en activité, ainsi que les objets qui agissent ou qui sont touchés par ces activités.
- **Les diagrammes états transition :** ils illustrent une machine à états et se composent d'états, de transitions, d'évènements et d'activités. Ils servent le plus souvent à modéliser le comportement d'objet lorsque ceux-ci sont soumis à des évènements.
- **Les diagrammes de séquences :** ils décrivent l'interaction entre objets en mettant en relief la séquence temporelle des messages. Ces objets sont en général des instances de classes, mais ils peuvent également représenter d'autres classificateurs tels que des acteurs, des composants ou des nœuds.
- **Les diagrammes de collaboration :** ils montrent également l'interaction entre objets, mais ils insistent sur l'organisation structurelle des objets émetteurs et récepteurs de messages.

V-3-2 La méthode UP [22]

Le Processus Unifié (UP) est un processus de développement logiciel « itératif et incrémental » ; Il s'agit d'une démarche s'appuyant sur la modélisation **UML** pour la description de l'architecture du logiciel (fonctionnelle, logicielle et physique) et la

mise au point de cas d'utilisation permettant de décrire les besoins et exigences des utilisateurs.

- **Itératif et incrémental** : le projet est découpé en itérations de courte durée (environ 1 mois) qui permettent de mieux suivre l'avancement global. A la fin de chaque itération, une partie exécutable du système final est produite, de façon incrémentale.
- **Centré sur l'architecture** : tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML et pas seulement documentée en texte.
- **Piloté par les risques** : les risques majeurs du projet doivent être identifiés au plus tôt mais surtout levés le plus rapidement possible. Les mesures à prendre dans ce cadre déterminent l'ordre des itérations.
- **Conduit par les cas d'utilisation** : le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d'utilisation du futur système sont identifiés, décrits avec précision et priorisés.

La gestion d'un tel processus est organisée suivant les **quatre phases** suivantes : initialisation, élaboration, construction et transition.

1. La phase d'**initialisation** conduit à définir la « vision » du projet, sa portée, sa faisabilité et son « business case » afin de pouvoir décider au mieux de sa poursuite ou de son arrêt.
2. La phase d'**élaboration** poursuit trois objectifs principaux en parallèle, à savoir:

- Identifier et décrire la majeure partie des besoins utilisateurs,
- Construire (et pas seulement décrire dans un document) l'architecture de base du système,
- Lever les risques majeurs du projet.

3. La phase de **construction** consiste surtout à concevoir et implémenter l'ensemble des éléments opérationnels (autres que ceux de l'architecture de base). C'est la phase la plus consommatrice en ressources et en effort.

4. Enfin, la phase de **transition** permet de faire passer l'application des développeurs aux utilisateurs finaux. Les mots-clés sont : conversion des données, formation utilisateurs, déploiement, béta-tests.

Les activités de développement sont définies par cinq disciplines fondamentales qui décrivent la capture des exigences, l'analyse et la conception, l'implémentation, le test et le déploiement. La modélisation métier est une discipline amont optionnelle et transverse aux projets. Enfin, trois disciplines appelées « de support » complètent le tableau : gestion de projet, gestion du changement et de la configuration, ainsi que la mise à disposition d'un environnement complet de développement incluant aussi bien des outils informatiques que des documents et des guides méthodologiques.

Le processus de développement UP apporte une méthodologie qui permet de disséquer un projet en plusieurs parties fortement réutilisables.

V-4 La conception

La première étape de notre processus de développement consiste à :

- définir la « vision » du projet, sa portée et sa faisabilité,
- effectuer un premier repérage des besoins fonctionnels et opérationnels,
- préparer les activités plus formelles de capture des besoins fonctionnels et de capture des besoins techniques.

Comme on l'a constaté dans les précédents chapitres, notre projet est scindé en trois phases l'authentification, l'autorisation et la comptabilisation, et afin de pouvoir bien comprendre le bon déroulement de chaque phase nous les aborderons une par une.

V-4-1 Différentes phases du projet

V-4-1.1 Authentification

L'installation et la configuration de Radius sont présentées en annexe IV.

Le diagramme de séquence suivant nous permet de comprendre comment se fait l'authentification en montrant les différentes interactions entre l'utilisateur et le serveur radius

Le déroulement de l'authentification est basé sur un scénario qui se rapproche du cheminement suivant :

- L'utilisateur envoie une demande de connexion,
- Le NAS envoie une demande d'identification à l'utilisateur,

- L'utilisateur envoie son identifiant et son mot de passe au NAS,
- Le NAS achemine la demande au serveur RADIUS.

Le serveur RADIUS consulte la base de données d'identification et valide ou refuse l'authentification.

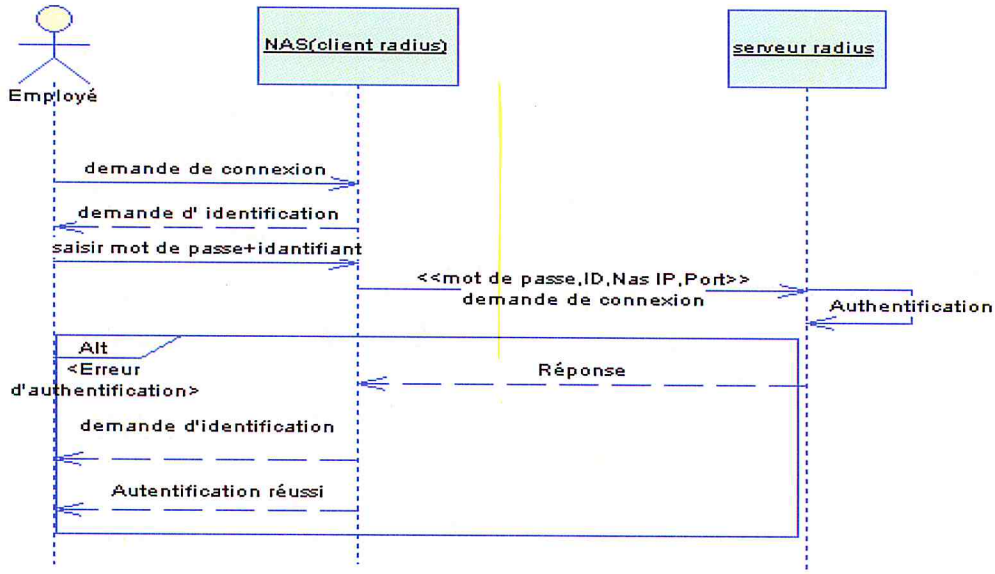

Figure 10: Diagramme de séquence « Authentification »

V-4-1.2 Autorisation

Pour la deuxième étape qui est l'autorisation prise en charge par Netfilter, elle est basée sur des règles de sécurité prédéfinie par l'administrateur.

Remarque : la configuration de Netfilter est présentée en annexe V.

V-4-1.3 La comptabilisation

La troisième étape qui est la comptabilisation, est prise en charge par Jpcap et le déroulement de ces deux dernières phases (autorisation, comptabilisation) est basé sur un scénario qui se rapproche du cheminement suivant :

Après l'authentification de l'utilisateur le NAS ouvre le port de connexion pour laisser passer le paquet de demande de connexion envoyé par l'utilisateur vers le pare-feu, ce dernier intercepte le paquet l'analyse et le traite selon les règles de filtrage et de routage existant dans la base de données.

En même temps Jpcap capte tous les paquets qui passent par l'interface réseau d'entrée du pare-feu et en garde une copie. Après la vérification des règles de

sécurité, le pare-feu autorise ou bloque le passage des paquets de connexion vers les ressources.

Le diagramme de séquence suivant nous permet de comprendre comment se fait l'autorisation en montrant les différentes interactions entre les acteurs.

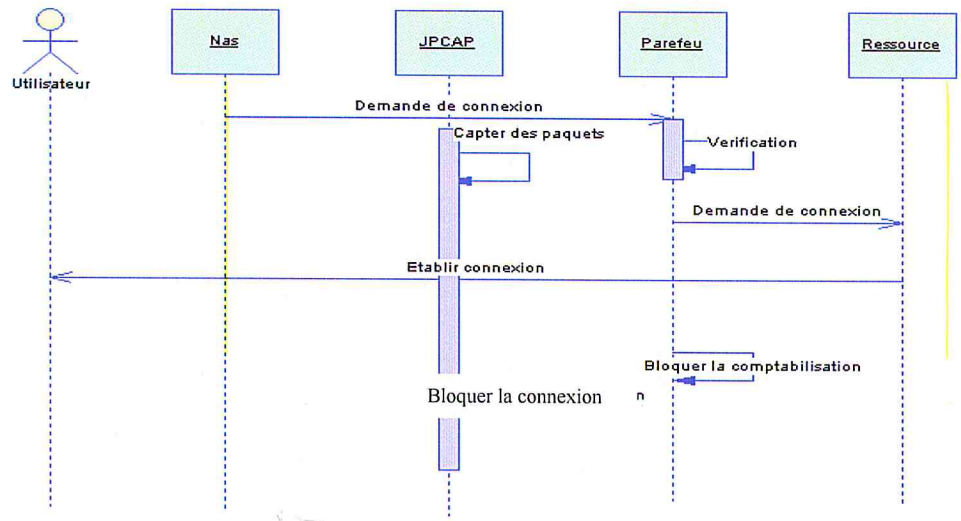

Figure 11: Diagramme de séquence « Autorisation et comptabilisation »

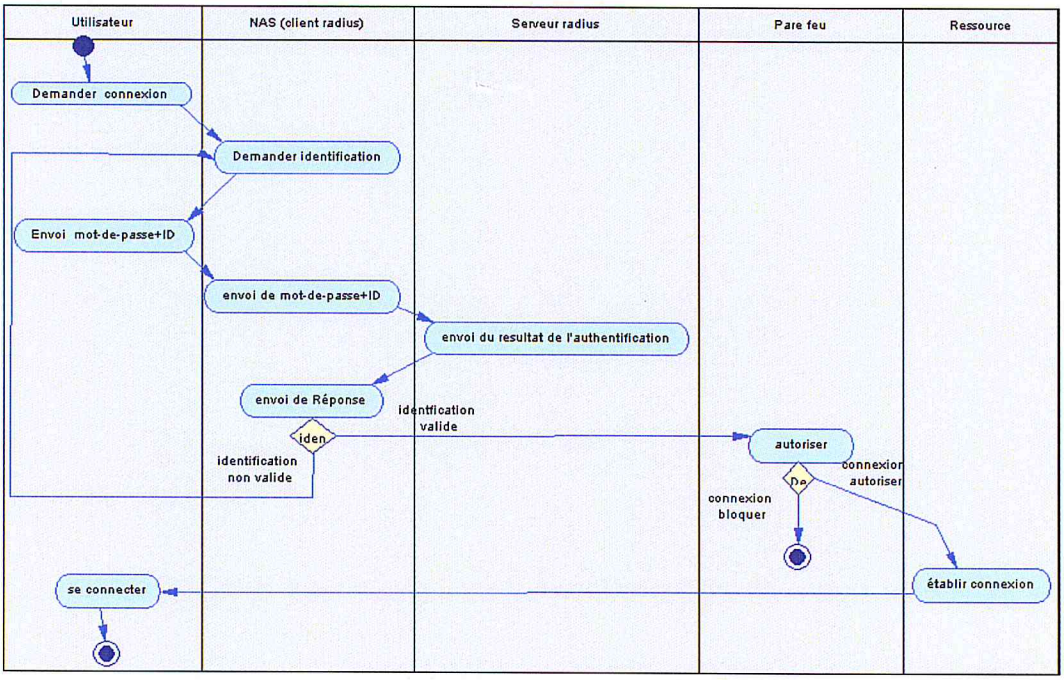

Figure 12: Diagramme d'activité

Afin d'assurer une bonne gestion des règles de sécurité et des ressources nous avons conçu une base de données MYSQL pour la sauvegarde des informations concernant chaque station,

V - 5 Détermination des cas d'utilisation

Les diagrammes de cas d'utilisation se composent d'acteurs (représentés par des silhouettes) et des cas d'utilisation (représentés par des ellipses). Les traits entre les cas d'utilisation et les acteurs représentent les interactions.

Ces diagrammes décrivent sous la forme d'actions et de réactions, le comportement d'un système du point de vue d'un utilisateur. Ils permettent de définir les limites du système et les relations entre le système et l'environnement.

L'étude des cas d'utilisation a pour objectif de déterminer ce que chaque acteur attend du système. La détermination des besoins est basée sur la représentation de l'interaction entre l'acteur et le système.

Cette approche a l'avantage de forcer l'utilisateur à définir précisément ce qu'il attend du système.

Après interview des utilisateurs, il ressort que l'administrateur est le seul acteur de notre plate-forme. Lui seul s'occupe de toutes les opérations. (**Tableau 1**)

Acteur	Rôle
Administrateur	C'est la personne responsable de la gestion des ressources, des règles et des captures.

Tableau 5: Les acteurs et leurs rôles

Après avoir définit l'acteur principal de notre système, il nous reste à déterminer les cas d'utilisations :

Acteur	Cas d'utilisation
Administrateur	Activer le pare- feu
	Désactiver le pare- feu
	Ajouter une ressource
	Supprimer une ressource
	Ajouter une règle
	Supprimer une règle
	Modifier une règle
	Consulter les captures
	Paramétrage du pare- feu

Tableau 6 : cas d'utilisation

V- 6 Description des cas d'utilisation

Les cas d'utilisation sont décrits de manière textuelle, argumentés de quelques diagrammes d'interaction. A ce stade de la modélisation, les interactions représentent les principaux événements qui se produisent dans le domaine de l'application. Plus tard, lors de la conception, ces événements sont traduits en messages qui déclenchent des opérations. Dans un premier temps, On va décrire seulement les scénarios nominaux (cas normal).

Pour chaque scénario, nous allons décrire son déroulement et concevoir un diagramme de séquence. Un diagramme de séquence est une représentation graphique qui met l'accent sur l'ordre chronologique des interactions entre les acteurs et le système.

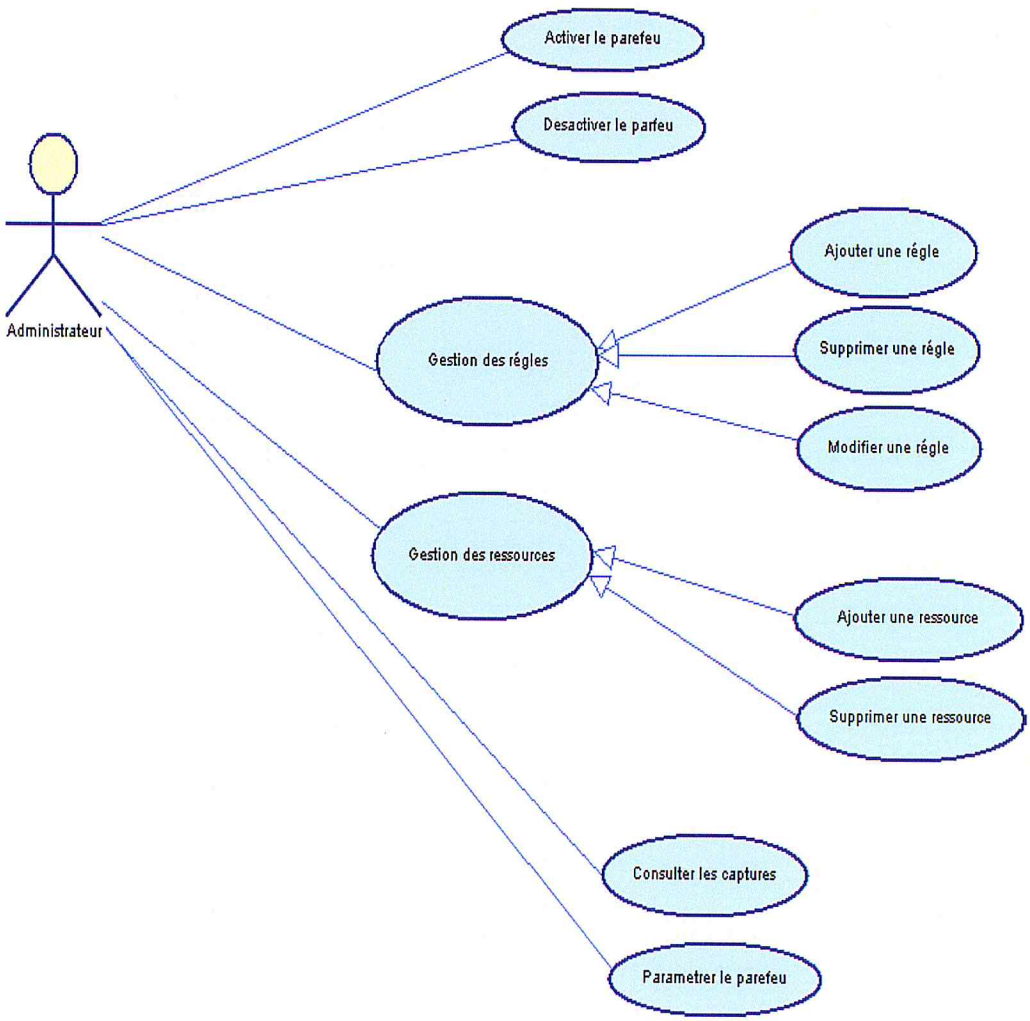

Figure 13: Diagramme des Cas d'utilisation de la plate-forme

V- 6-1 Les cas d'utilisations et scénaris principaux

V- 6-1-1 Cas d'utilisation « Activer le pare- feu»

✓ **Scénario**

N° Acheminement	Action Acteur et action Système
1	L'administrateur envoie une demande d'activation du pare- feu au système
2	Le système envoie une demande d'activation au pare-feu et au Jpcap au même temps
3	Le pare- feu s'active et envoie un message d'activation
4	Le Jpcap commence la capture

Tableau 7: Scénario « Activer le pare- feu»

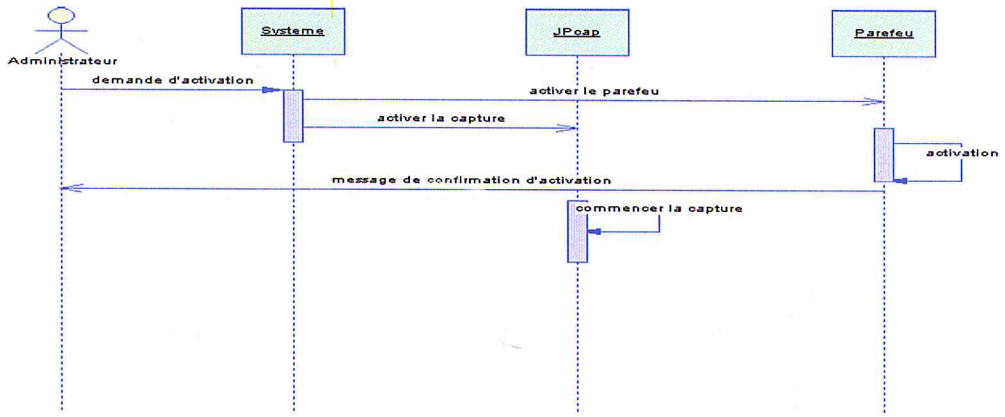

Figure 14: Diagramme de séquence « Activer le pare- feu »

V- 6-1-2 Cas d'utilisation « Désactiver le pare- feu»

✓ **Scénario**

N° Acheminement	Action Acteur et action Système
1	L'administrateur envoie une demande de désactivation du pare- feu au système
2	Le système envoie une demande de désactivation au pare-feu et au Jpcap au même temps
3	Le pare- feu se désactive et envoie un message de confirmation
4	Le Jpcap Arrête la capture

Tableau 8: Scénario «Désactiver le pare- feu »

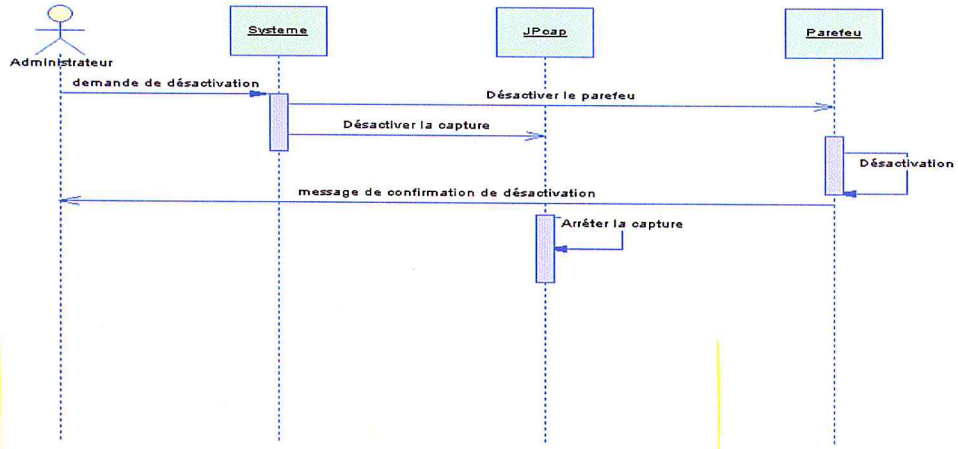

Figure 15 : Diagramme de séquence « Désactiver le pare- feu »

V- 6-1-3 Cas d'utilisation « Gestion des ressources »

Cas d'utilisation	Gestion des ressources
Acteurs	L'administrateur
But	Permettre à l'administrateur de gérer l'ensemble des ressources du réseau.
Résumé métier	Le système offre à l'administrateur la possibilité de gérer les ressources de ce système en lui permettant l'ajout, la modification et la suppression des utilisateurs.

Tableau 9 : Cas d'utilisation « Gestion des ressources »

✓ **Scénario « Ajouter une ressource »**

N° Acheminement	Action Acteur et action Système
1	L'administrateur demande d'ajouter une ressource.
2	Le système affiche le formulaire de saisie contenant les champs suivants : Identifient /Masque/ Adresse IP/ type.
3	L'administrateur remplit les champs et valide.
4	Le système enregistre et affiche un message de confirmation

Tableau 10 : Scénario « Ajouter une ressource »

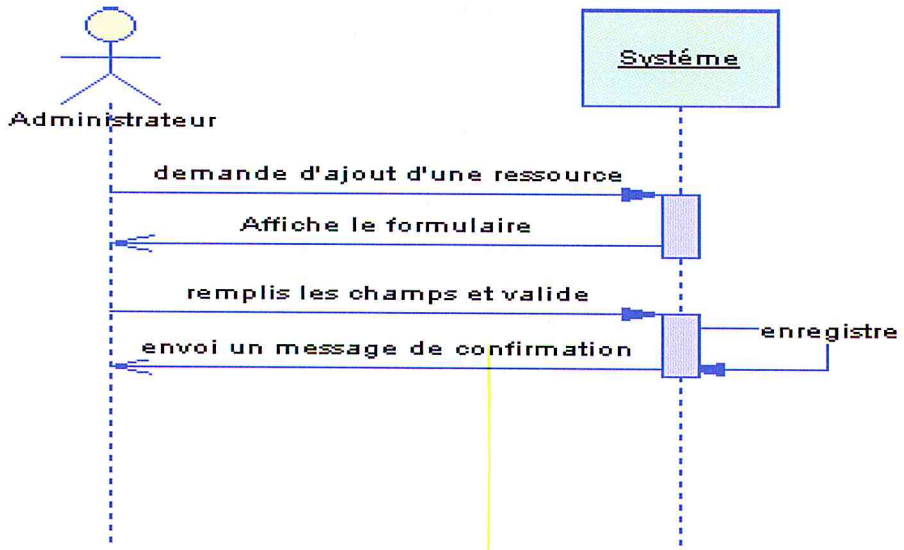

Figure 16: Diagramme de séquence « Ajouter une ressource »

✓ **Scénario « Supprimer une ressource »**

N° Acheminement	Action Acteur et action Système
1	L'administrateur demande la suppression d'une ressource
2	le système affiche un formulaire
5	L'administrateur sélectionne la ressource à supprimer
6	Le système supprime et envoie un message de confirmation

Tableau 11: Scénario « Supprimer une ressource »

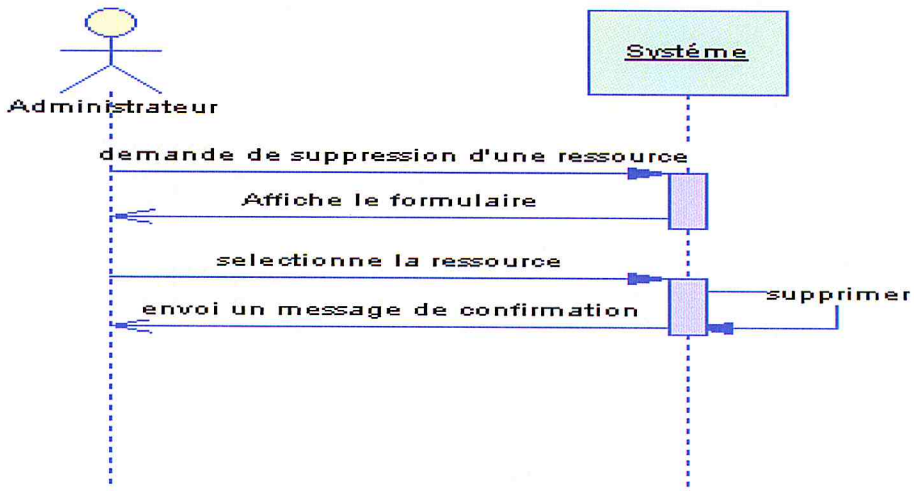

Figure 17 : Diagramme de séquence « supprimer une ressource »

V- 6-1-4 Cas d'utilisation de la gestion des règles

Cas d'utilisation	Gestion des règles
Acteurs	L'administrateur
Buts	Permettre à l'administrateur de gérer l'ensemble des règles.
Résumé métier	Le système offre à l'administrateur la possibilité de gérer les règles de sécurité de ce système en lui permettant l'ajout, la modification et la suppression des règles.

Tableau 12: Cas d'utilisation de la gestion des règles

✓ **Scénario «Ajouter une règle »**

N° Acheminement	Action Acteur et action Système
1	L'administrateur demande d'ajouter une règle.
2	Le système affiche le formulaire.
3	L'administrateur remplit les champs et valide.
4	Le système enregistre et envoie une confirmation.

Tableau 13: Scénario «Ajouter une règle »

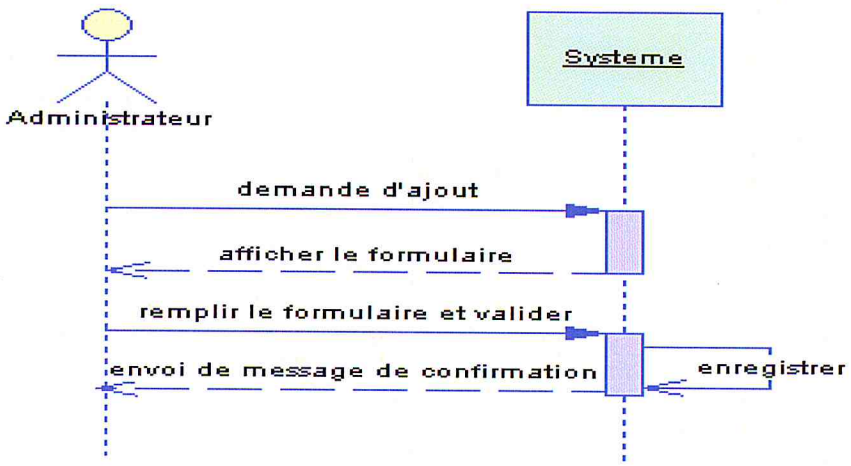

Figure 18 : Diagramme de séquence « ajouter une règle »

✓ **Scénario «Modifier une règle »**

N° Acheminement	Action Acteur et action Système
1	L'administrateur demande de modifier une règle.
2	Le système affiche le formulaire.
3	L'administrateur modifie les champs et valide.
4	Le système enregistre et envoie une confirmation.

Tableau 14 : Scénario «Modifier une règle »

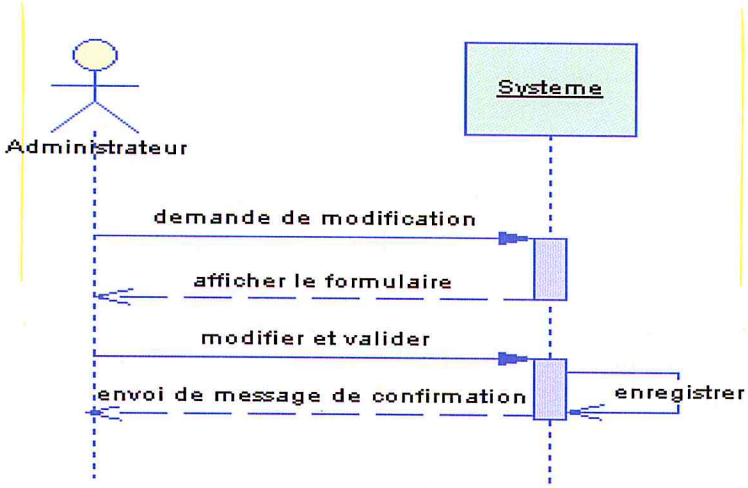

Figure 19 : Diagramme de séquence « modification d'une règle »

✓ **Scénario « Supprimer une règle »**

N° Acheminement	Action Acteur et action Système
1	L'administrateur envoie une demande de suppression
2	Le système affiche le formulaire
3	L'administrateur remplit les champs
4	Le système recherche et affiche le résultat
5	L'administrateur sélectionne la règle
6	Le système effectue la suppression et envoie un message de confirmation

Tableau 15: Scénario « Supprimer une règle »

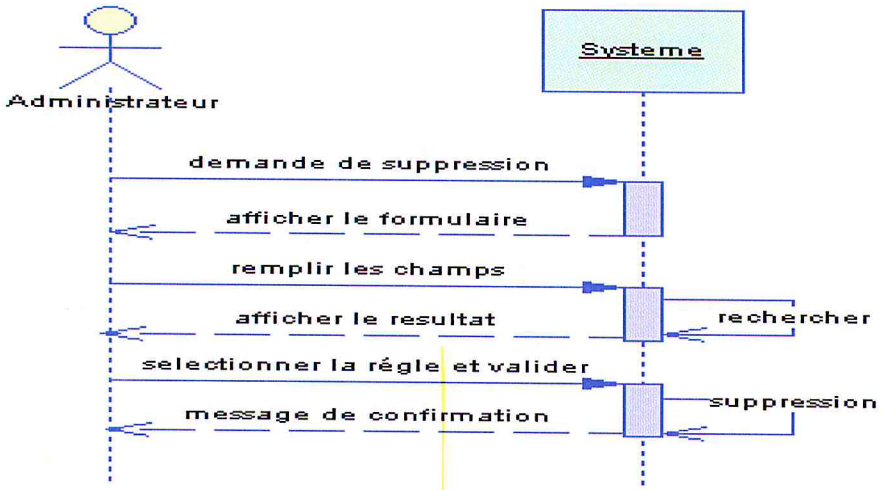

Figure 20: Diagramme de séquence « supprimer une règle »

V- 6-1-5 Cas d'utilisation « Consulter les captures »

✓ **Scénario**

N° Acheminement	Action Acteur et action Système
1	L'administrateur envoie une demande de consultation des captures au système
2	Le système afficher le formulaire
3	L'administrateur remplit les champs
4	Le système recherche et affiche le résultat

Tableau 16 : Scénario « Consulter les captures »

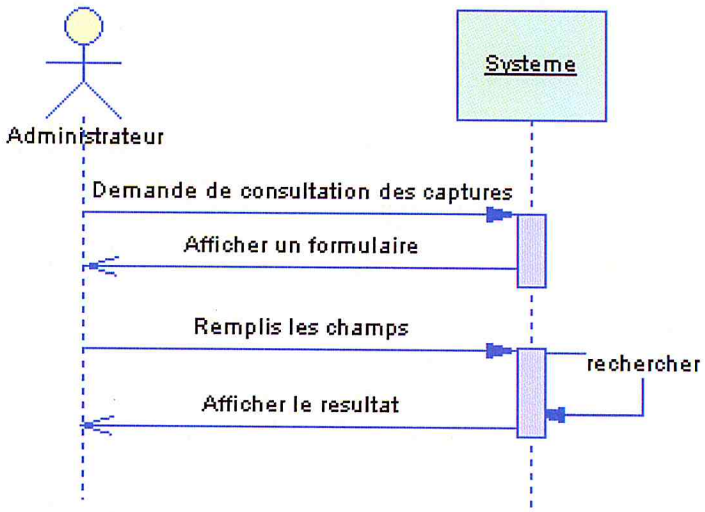

Figure 21 : Diagramme de séquence « Consulter les captures »

V - 7 Description de collaboration

Dans cette section, nous nous intéressons à la description de l'intérieur du système. Nous décrivons l'interaction et l'échange des messages entre les objets du système pour réaliser un cas d'utilisation. Comme la plupart des cas d'utilisation se ressemble, nous nous contenterons d'un seul diagramme de collaboration.

Ajout d'une règle

Le diagramme suivant représente la collaboration entre les objets pour réaliser le cas " Ajout d'une règle".

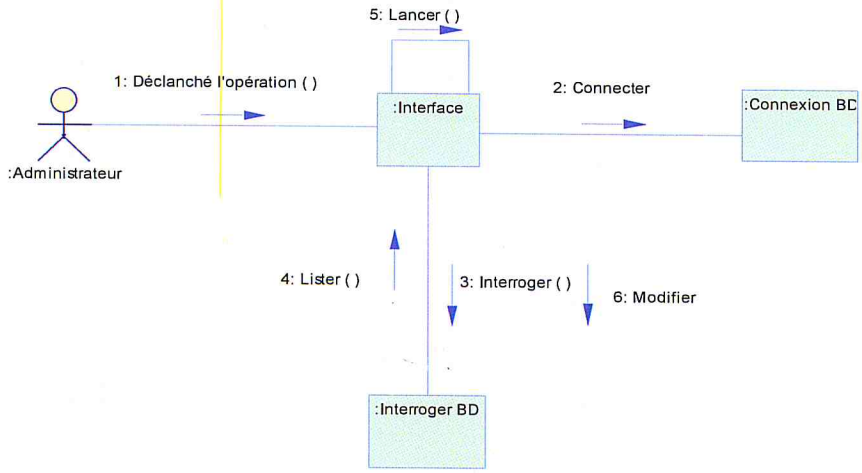

Figure 22: Diagramme de collaboration

L'utilisateur déclenche « l'ajout » par le biais de notre interface. L'interface de l'ajout s'affiche, l'utilisateur remplit les champs et valide ensuite cette opération. L'ajout s'effectue et la mise à jour se fait dans la base.

V - 8 Le Diagramme de déploiement

Les diagrammes de déploiement montrent la disposition physique des différents matériels (les nœuds) qui entrent dans la composition d'un système et la répartition des instances de composants, processus et objets qui « vivent » sur ces matériels.

Les diagrammes de déploiement sont donc très utiles pour modéliser l'architecture physique d'un système.

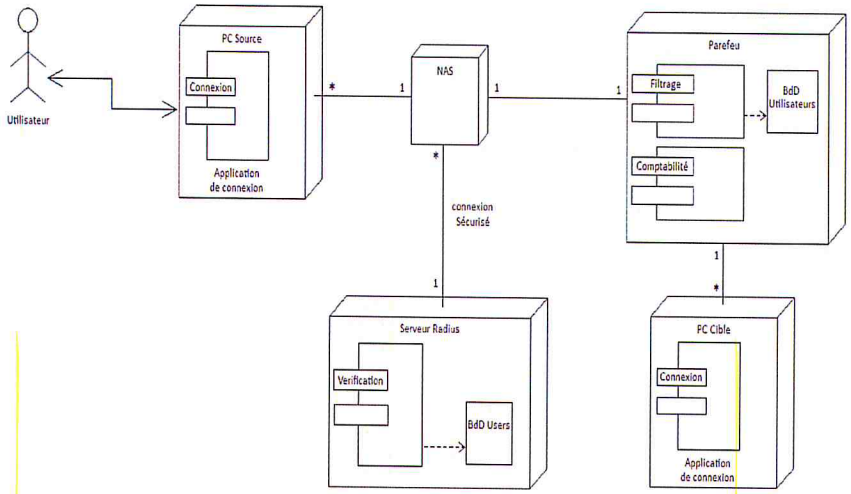

Figure 23 : Diagramme de déploiement

V - 9 Le Diagramme de classe

Après une étude approfondie du fonctionnement global de notre système, nous aboutissons à une représentation statique du système matérialisé par le diagramme de classe.

Ce diagramme nous permet d'avoir une vue statique de l'application. Il nous montre les relations entre les différentes entités (classes) composant notre application. Il nous mène vers une abstraction transparente de la solution finale.

A partir de ce diagramme, on retrouve les corps des différentes classes de notre application.

Pour des raisons d'espace et une meilleure représentation, on va utiliser un formalisme de représentation simplifié des classes dans le diagramme de classes, en mettant juste le nom de la classe, et en détaillant les classes par une description qui va spécifier les attributs et les opérations pour chaque classe.

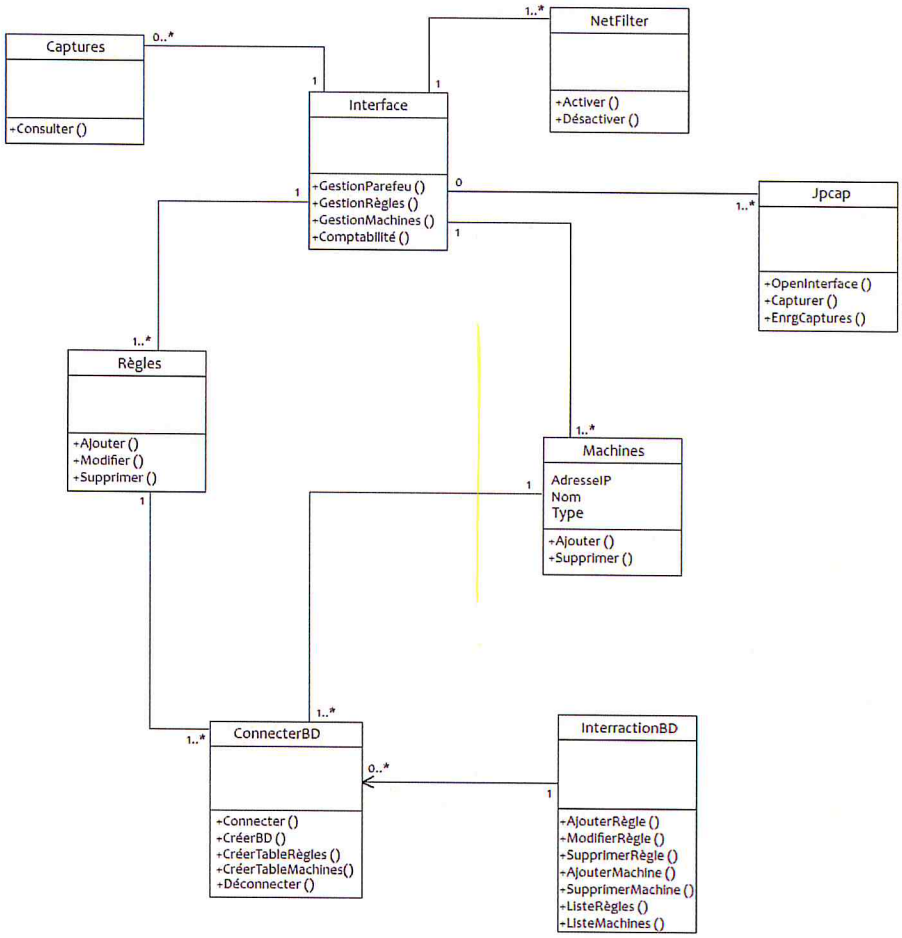

Figure 24 : Le diagramme de classe

V - 10 Conclusion

Nous avons présenté dans ce chapitre le mode de fonctionnent d'un système d'authentification et d'un pare feu (firewall) pour renforcer la sécurité. Ce système permet à Algérie Télécom (AT) de définir des politiques flexibles. Pour gérer l'accès à ces ressources, grâce à l'architecture de ce système, il est possible de supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modification majeure. On parle alors d'un réseau évolutif.

Le chapitre suivant va présenter la concrétisation de cette étude pour aboutir à l'implémentation des interfaces qui nous permettra par la suite de gérer ce système.

Chapitre VI

L'Implémentation

VI - L'IMPLEMENTATION

VI - 1Introduction

Il est, bien entendu, évident qu'une conception bien faite facilite énormément la réalisation du projet. Le découpage modulaire permet, en plus, d'avoir une vue plus détaillée sur la structure interne du logiciel à réaliser.

Dans ce chapitre nous allons décrire les choix techniques, les détails de l'implémentation, la configuration nécessaire de notre produit et nous donnerons quelques explications nécessaires pour comprendre son fonctionnement.

VI - 2 Description de l'environnement de développement

VI - 2 -1 Aspect logiciel

✓ **Le Système d'exploitation LINUX**

Pour l'implémentation de notre système, nous avons opté pour Linux comme système d'exploitation. Linux est un système d'exploitation multiutilisateur et multitâche.

- Le système multitâche est facile à expliquer ; c'est en fait un système d'exploitation où les applications sont exécutées en parallèle les unes aux autres, ce qui fait que si une application plante, le reste du système est très peu ralenti et aucune autre application n'est perturbée.
- Le système multiutilisateur permet d'avoir différents profils sur un même poste, et en plus, les utilisateurs ne peuvent pas, sauf autorisation, lire ou modifier les fichiers des autres utilisateurs.

La différence de Linux par rapport aux autres systèmes d'exploitation (O.S.) réside dans les paramètres suivants:

1. **La Puissance** qui permet de faire beaucoup de choses sur la machine.
   - Mémoire maximum supportée par le noyau de Linux: **64,**
   - Nombre processeurs supportés par Linux: **254**.
2. **L'efficacité** ; contrairement à des systèmes beaucoup plus répandus, il n'utilise pour ses besoins propres que très peu de ressources. Les logiciels qui seront utilisés pour le travail disposeront donc de beaucoup plus de puissance pour fonctionner.
3. **La fiabilité** ; une machine sous Linux fonctionne 24h/24 si besoin est sans se plaindre, particulièrement sur le plan thermique.

4. **La robustesse** ; une erreur d'un utilisateur ou un ``plantage" éventuel d'une application n'affecte pas le reste du système. D'autre part, il est exceptionnel de devoir l'arrêter ; la quasi-totalité des opérations de configuration, mise au point, etc.., ne nécessitent pas l'arrêt du système.

5. **Le coût** ; il est très bon marché ; le prix demandé par les sociétés qui vendent Linux sur CDROM ne sert qu'à couvrir leurs frais et à leur permettre de financer dans une certaine mesure la poursuite de cette activité. Linux étant développé par des passionnés pour le plaisir, personne n'a à supporter le coût de son développement.

En outre, alors que Linux lui-même est disponible gratuitement sur Internet, différentes sociétés ont construit ce que l'on appelle des distributions, que l'on peut assimiler à des versions "boîtes" de Linux. Elles comprennent le noyau Linux et l'environnement GNU, incluant un logiciel d'installation et peuvent comporter des logiciels propriétaires et un support.

Ainsi, Linux est un système exceptionnel donnant satisfaction aussi bien sur des machines anciennes ou bas de gamme que sur des machines puissantes très sollicitées ou devant remplir des fonctions importantes.

Le système utilisé est UBUNTU 10.04 LTS

✓ **Netfilter**

Depuis que Netfilter a fait son apparition en standard avec le noyau 2.4, il est possible de réellement se protéger avec son firewall sous Linux. Netfilter est un dispositif de filtrage introduisant :

-La notion d'état permettant de savoir si un paquet appartient à une communication en cours ou non,

- Le NAT : la translation d'adresse,

- La possibilité de modifier des paquets à la volée.

L'applicatif est aussi appelé Iptables qui est le successeur des précédents systèmes Linux 2.2.x ipchains et Linux 2.0.x ipfwadm.

VI – 2-2 L'aspect langage et outils de programmation/ Java

Dans le cadre de l'implémentation de notre application, nous avons opté pour le langage orienté objet Java ; ce dernier possédant de nombreuses caractéristiques qui en font un des langages de choix. En effet, Java a été conçu pour mettre en œuvre des applications susceptibles de s'exécuter sur n'importe qu'elle plateforme, ce qui est très avantageux dans le cas d'environnements hétérogènes. Il offre, en même temps, une souplesse relative grâce aux opérations de "casting". En plus du fort typage, Java met à disposition plusieurs niveaux de protection des données, ainsi qu'un mode de traitement des exceptions très sophistiqué. En outre , Java propose des outils et des mécanismes qui permettent au programmeur de faire de la programmation concurrente (multi-threaded), de réaliser des applications Internet, et de mettre en œuvre des interfaces utilisateurs très sophistiquées, ainsi que la manipulation de base de données en tout genre.

- ✓ **Base de données MYSQL**

MYSQL est un moteur de base de données multi plateformes de niveau industriel qui peut être utilisé pour d'énormes applications de base de données contenant des tables dotées de millions de lignes. Il a prouvé, à de nombreuses reprises, qu'il était parfaitement adapté à la plupart des applications.

- ✓ **FreeRADIUS version 2.1.8**

Comme il a été précisé dans le chapitre authentification, il existe plusieurs solutions sur le marché, certaines commerciales, d'autres libres. Notre choix s'est porté sur FreeRadius pour la simple raison qu'il est libre sous Linux.

Il présente aussi l'avantage d'être très stable, de bénéficier d'une communauté très active au travers d'un site web et de listes de diffusion. Il s'agit d'une solution très largement répandue sur divers systèmes d'exploitation et apte à s'intégrer dans des environnements très variés grâce à sa compatibilité avec les standards les plus courants.

FreeRADIUS est une implémentation de RADIUS élaborée, à la suite du projet Cistron, par un groupe de développeurs. La scission entre les deux projets date de 1999. C'est un projet Open Source sous licence GPL. Le site officiel du projet est http://www.freeRADIUS.org.

FreeRADIUS doit son succès à sa compatibilité avec un grand nombre de standards

couvrant les systèmes d'exploitation, les protocoles et les bases d'authentification.

Il est annoncé comme testé et fonctionnel sur les systèmes Linux (Toutes distributions), FreeBSD, NETBSD et Solaris.

VI - 2 -3 L'aspect matériels

Les matériels et logiciels nécessaires sont :

- Les routeurs niveau 3 supportant le protocole d'authentification RADIUS. (NAS),
- Les serveurs avec architecture x86 ; interface réseau. (Serveur Radius),
- Un micro-ordinateur avec architecture x86 ; deux interfaces réseau. (Passerelle).

VI – 3 L'implémentation

Avant de présenter notre produit avec ses différentes interfaces graphiques, on doit préciser quelques notions et détails sur l'installation et la configuration de notre protocole d'authentification ainsi des détails sur Netfilter et tout au long de ces explications, nous allons détailler l'implémentation des points suivants :

1. La gestion des ressources,
2. La gestion des règles,
3. La consultation des captures.

Fenêtre principale du pare- feu

La fenêtre principale nous permet la gestion des différentes opérations suivantes :

1: Démarrer le Filtrage.
2 : Arrêter le Filtrage,
3 : Afficher les paramètres de la gestion des machines,
4: Afficher les paramètres de la gestion des règles,
5 : Consulter les captures.
6 : Accéder au Journal du de l'application.

Figure 25: Fenêtre principale du pare-feu

Fenêtre de Gestion des machines

Cette fenêtre nous permet de choisir, à l'aide des boutons, soit d'ajouter, modifier ou supprimer une règle.

7 : le tableau d'affichage de la liste des machines,

8 : Le bouton "Ajouter une machine" nous affiche la fenêtre "d'ajout"

9 : Le bouton " supprimer une machine" pour valider le choix de la machine à supprimer parmi la liste affichée,

10 : Le bouton "quitter" pour fermer la fenêtre.

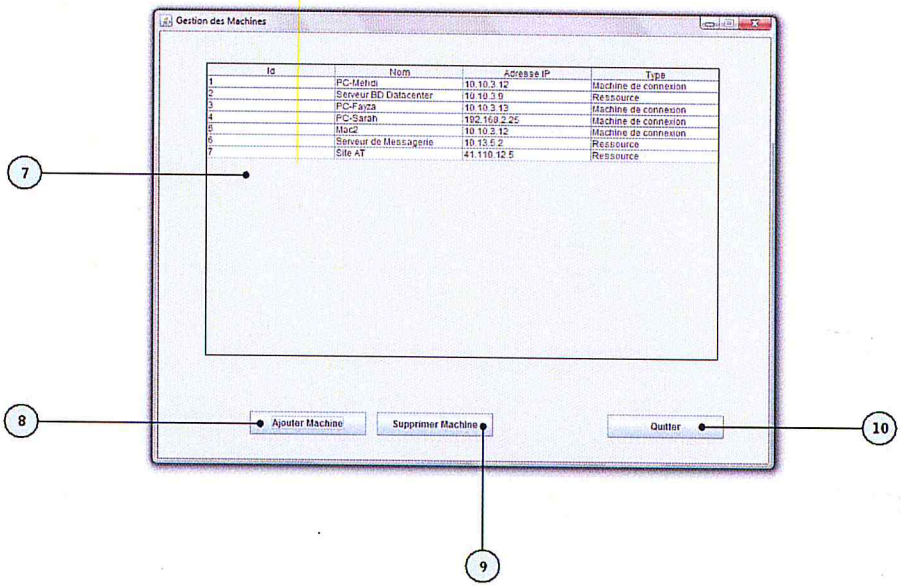

Figure 26: Fenêtre Gestion des machines

Fenêtre Ajout d'une machine

Cette fenêtre nous permet d'ajouter une nouvelle machine dans la base figure.

11 : Pour introduire le nom de la machine,

12 : Pour introduire l'adresse IP de la machine,

13 : Pour choisir le type de la machine " ressource" où " machine de connexion ",

14: Bouton pour valider les données,

15 : Bouton pour annuler et fermer cette fenêtre,

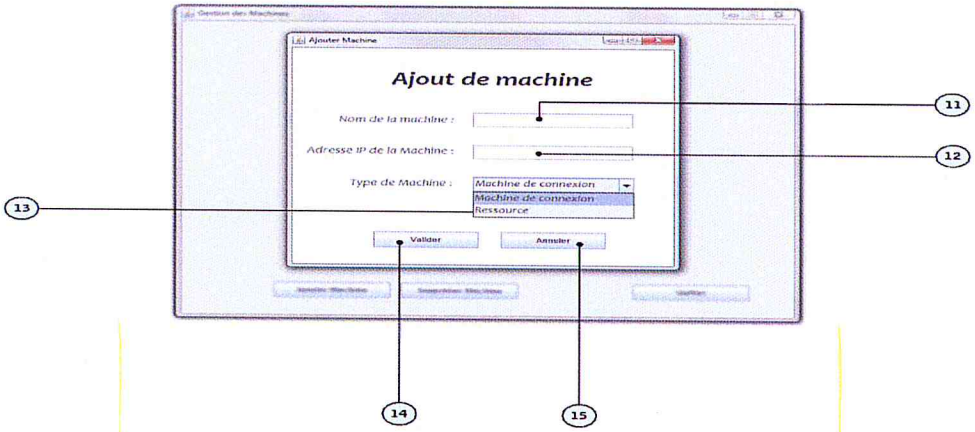

Figure 27 : Fenêtre Ajout d'une machine

Fenêtre suppression d'une machine

La fenêtre " Gestion des machines", figure 26, nous permet la suppression d'une machine de la base. Pour la suppression on choisit la machine sur la liste et avec le bouton valider on confirme la suppression

Fenêtre de gestion des règles

Cette fenêtre nous permet de choisir d'ajouter, supprimer ou modifier une règle de sécurité de la base figure.

16 : Le tableau d'affichage de la liste des règles de sécurité,

17 : Le bouton "Ajouter une règle" affiche la fenêtre "d'ajout",

18 : Le bouton "quitter" pour fermer la fenêtre,

19 : pour valider le choix de la règle à supprimer parmi la liste affichée.

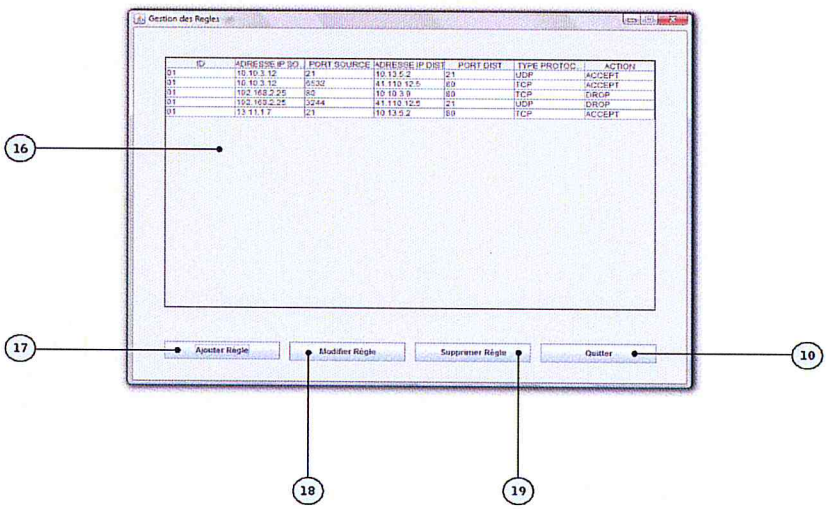

Figure 28: Fenêtre de gestion des règles

Fenêtre ajout d'une règle

Cette fenêtre nous permet d'ajouter une nouvelle règle dans la base figure.

20 : Choisir l'adresse IP de la machine de connexion dans la liste déroulante,

21 : Introduire le port de la machine de connexion,

22 : Choisir l'adresse IP destination de la ressource destination dans la liste Déroulante,

23 : Introduire le port de la machine de connexion,

24 : Choisir le type de protocole qui sera utilisé pour la communication,

25 : Choisir de l'autoriser ou lui bloquer cette connexion.

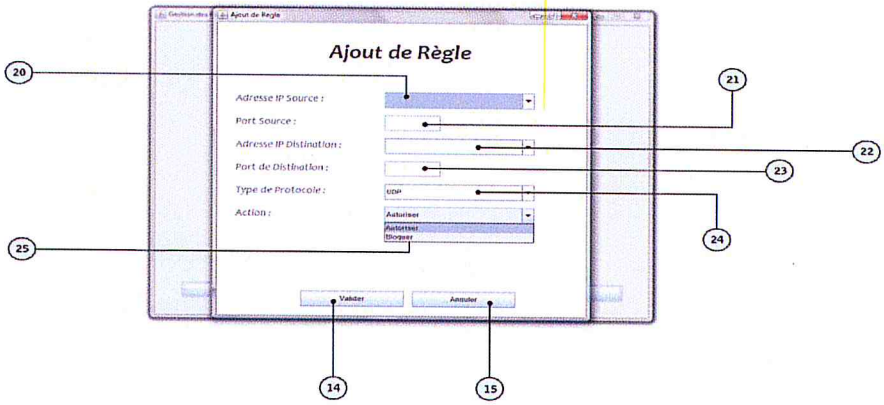

Figure 29: Fenêtre ajout d'une règle

Fenêtre suppression d'une règle

La fenêtre "Gestion des règles", figure 28, nous permet la suppression d'une règle de la base.

On choisit la règle de sécurité sur la liste et on confirme la suppression avec le bouton valider.

Fenêtre modification d'une règle

La fenêtre "ajout d'une règle", figure 29, nous permet de modifier une règle dans la base figure.

On choisit la règle de sécurité sur la liste sur la fenêtre "ajout d'une règle" et on confirme la modification avec le bouton valider.

Pour la modification, on ne peut modifier que l'action, par exemple, si une machine est autorisée à se connecter à une ressource, on peut lui enlever ce privilège et donc lui bloquer la connexion.

VI – 4 Conclusion

Dans ce chapitre nous venons de présenter une phase très importante de notre projet, à savoir la plateforme de la gestion du pare- feu (firewall). Nous avons commencé, par décrire les composants logiciels et matériels qui ont été exploités. Par la suite, nous avons illustré les différentes interfaces constituant notre application afin de voir toutes les fonctionnalités qu'elle nous offre.

Cette phase de notre processus de réalisation, a été l'étape la plus exigeante en efforts et en temps. Elle nous a permis d'approfondir et d'enrichir nos connaissances dans le domaine des réseaux informatiques ainsi que leur sécurité.

Lors de la phase des tests, de nombreuses difficultés ont entravé le bon déroulement du projet. Pourtant, ces désagréments se sont révélés être en fait l'opportunité qui nous a permis d'effectuer des recherches et de découvrir de nouveaux concepts. En d'autres termes, la plupart des outils de développement et des équipements de test étant nouveaux du point de vue pratique (routeur, switchs), leurs installations, configurations et intégrations n'ont pas été aussi faciles que l'on pouvait le croire.

Conclusion générale et perspectives

Le manque de sécurité reste le vrai problème des entreprises et particulièrement Algérie Télécom. Et la question qui se pose est comment savoir se protéger contre les attaques malveillantes ?

A travers notre projet, nous avons voulu montrer que la prise en charge de ce phénomène nécessite:

- l'utilisation et la connaissance des outils pouvant servir,
- l'adoption d'une bonne politique de sécurité approuvée par des spécialistes du domaine.

Pour notre part nous nous sommes proposés, à travers notre travail, de concevoir et réaliser une solution pour sécuriser le réseau de l'entreprise d'accueil, ALGERIE Télécom.

Pour atteindre notre objectif, nous avons organisé, dans le présent mémoire, l'enchaînement suivant pour pouvoir parvenir à la "**Réalisation de notre solution de sécurisation**".

1. Ainsi, nous avons commencé par introduire la sécurité informatique.
2. Nous avons, ensuite, abordé les protocoles d'authentification et les pare-feux pour passer à l'outil Jpcap qui nous a servi pour la capture des paquets.

Ces travaux ont constituée la première phase dite théorique.

Après cette phase, nous sommes passés à la conception UML de notre solution afin de faciliter la phase de réalisation.

Il était prévu d'utiliser, dans notre projet, le protocole DIAMETER pour l'authentification, mais un problème est survenu au niveau de l'exécution du service du protocole OpenDiameter sur le système d'exploitation UBUNTU 9.10 (erreur de segmentation).

Après plusieurs essais non réussis durant tout un trimestre, Algérie Télécom nous a recommandé l'utilisation du protocole RADIUS pour notre application.

Notre solution, avec l'utilisation de Radius, assure l'authentification et l'autorisation des utilisateurs ainsi que la gestion des règles de sécurité et des ressources.

La phase d'authentification nous permet de contrôler l'accès au réseau et de s'assurer que la personne qui se connecte est réellement celle qu'elle prétend être.

La phase d'autorisation consiste à protéger les Systèmes Informatiques et la confidentialité des données qui ne seront accessibles que par les personnes autorisées. Ainsi, il empêche que les données soient compréhensibles par une entité tierce non autorisée.

La phase de comptabilisation nous permet de collecter des informations sur l'utilisation des ressources.

Après plusieurs recherches soutenues d'essais, les résultats obtenus nous ont encouragés à développer des combinaisons entre les différents outils utilisés et des protocoles existants pour aboutir à un seul système très efficace pour sécuriser le réseau d'Algérie Télécom.

Le travail effectué et présenté dans ce mémoire :

- représente notre contribution à la conception et la mise en place d'une plateforme,

Comme il nous a permis de,

- approfondir nos connaissances concernant la sécurité informatique, l'architecture, le fonctionnement et le déploiement réels des réseaux informatiques avec leurs différents services,
- appliquer également les principes et les méthodes théoriques acquis durant notre cursus de formation master, en l'occurrence UML , POO , les bases de données,
- nous familiariser, particulièrement, avec un langage de programmation puissant et ouvert, à savoir JAVA (ECLIPSE).

En terme de perspectives et de continuation de ce modeste travail, il serait très intéressant et très prometteur de :

- réaliser l'authentification avec le protocole DIAMETER,
- réaliser l'autorisation et la comptabilisation avec le protocole RADIUS et DIAMETER,
- développer un module de filtrage du trafic qui intervient au niveau de la couche N°3 du modèle OSI « couche réseau » en utilisant Jpcap,
- développer la gestion du pare-feu en lui ajoutant des modules qui permettent de faire les statistiques sur le trafic réseau.

Annexes

ANNEXE N° 1

ANNEXE 1

PRÉSENTATION D'ALGÉRIE TELECOM

ALGERIE TELECOM, est une société par actions à capitaux publics opérant sur le marché des réseaux et services de communications électroniques. Entrée officiellement en activité à partir du 1er janvier 2003, elle s'engage dans le monde des Technologies de l'Information et de la Communication.

Sa naissance a été consacrée par la loi 2000/03 du 5 août 2000, relative à la restructuration du secteur des Postes et Télécommunications, qui sépare notamment les activités Postales de celles des Télécommunications. ALGERIE TELECOM est donc régie par cette loi qui lui confère le statut d'une entreprise publique économique sous la forme juridique d'une société par actions SPA.

Son ambition est d'avoir un niveau élevé de performance technique, économique, et sociale pour se maintenir durablement leader dans son domaine, dans un environnement devenu concurrentiel. Son souci consiste, aussi, à préserver et développer sa dimension internationale et participer à la promotion de la société de l'information en Algérie.

I-1- Missions et objectifs

L'activité majeure d'Algérie Télécom est de :

- Fournir des services de télécommunication permettant le transport et l'échange de la voix, de messages écrits, de données numériques, d'informations audiovisuelles ;
- Développer, exploiter et gérer les réseaux publics et privés de télécommunications ;
- Etablir, exploiter et gérer les interconnexions avec tous les opérateurs des réseaux.
- Accroître l'offre de services téléphoniques et faciliter l'accès aux services de télécommunications au plus grand nombre d'usagers, en particulier en zones rurales ;
- Accroître la qualité de services offerts et la gamme de prestations rendues et rendre plus compétitifs les services de télécommunications ;
- Développer un réseau national de télécommunication fiable et connecté aux autoroutes de l'information.

I-2- Organisation d'Algérie Télécom

ALGERIE TELECOM est organisée en Directions Centrales, Régionales et Directions Opérationnelles de Wilaya autour de ses métiers fixes et services et d'autre part des fonctions supports réseaux. A cette structure s'ajoutent une filiale mobile et deux Directions de Projets chargées l'une de l'Internet et l'autre des Télécommunications Spatiales.

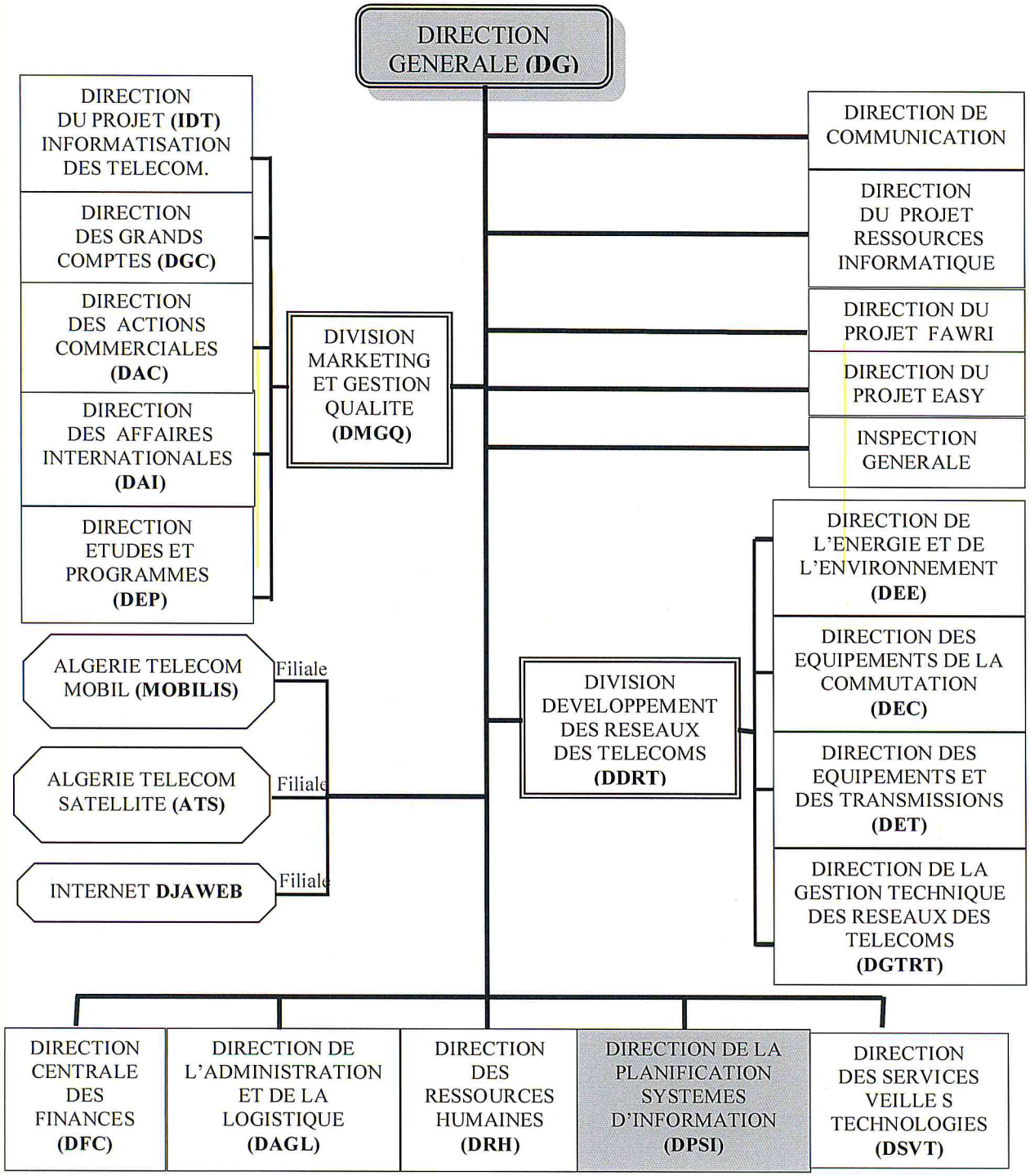

Figure 1 : Organigramme de la DG d'AT

II- Organisation du DPSI

Le schéma ci après illustre l'organisation de la direction de la planification systèmes d'information (DPSI)

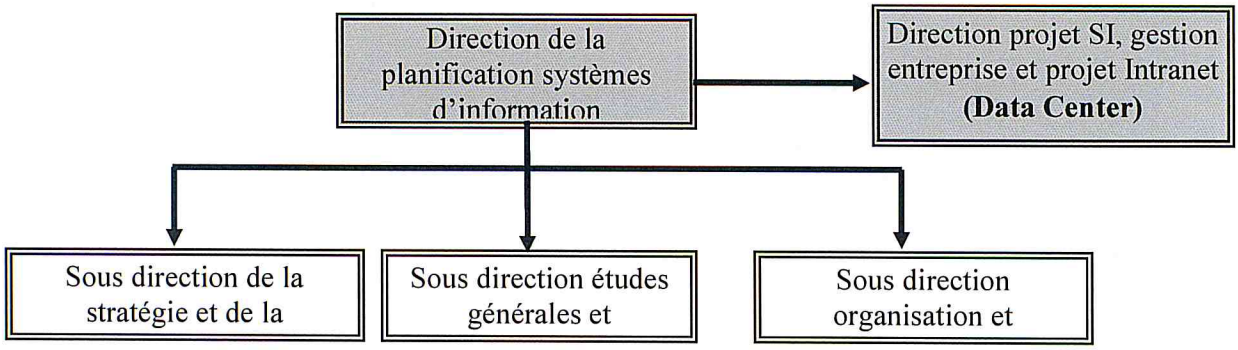

Figure 1.2 : Organigramme du DPSI

II-1 Présentation du Data Center

Le Data Center constitue le noyau central du Système d'Information Global d'Algérie Télécom. à vocation nationale il prend en charge les plateformes de gestion, le réseau intranet et la sécurité informatique de l'entreprise. Ses missions principales sont :

- Héberger et sécuriser les applications de gestion.
- Mettre à la disposition des directions métiers les moyens nécessaires à la gestion (accès aux applications métiers, Accès Intranet et Internet).
- Assurer la maintenance préventive et corrective du logiciel et du matériel composant le réseau.
- Suivre l'évolution sur la technologie de l'information (matérielle, logicielle).
- Élaborer un programme d'extension des systèmes en cas de saturation en fonction des besoins.
- Assurer un support technique pour l'ensemble du personnel d'Algérie Télécom.
- Mettre en place des architectures réseaux.
- Mettre en place des solutions de sécurité informatique.

II-2- Organisation du Data Center

Le Data Center est organisé comme suit :

1) **Département système :** Ce département a pour mission d'assurer le bon fonctionnement des plateformes de production, de développement et de sauvegarde, il est composé de :
   - Service des systèmes d'impressions.
   - Service backup /archive.

2) **Département bases de données :** Ce département a pour mission d'assurer le bon fonctionnement des bases de données de production et de développement. Il a un seul service :

- Service de développement.

3) **Département réseaux :** Ce département a pour mission d'assurer le bon fonctionnement du réseau INTRANET d'Algérie Télécom et de prendre en charge son évolution dans le temps. Il est composé de :

- Service messagerie.
- Service help desk.

4) **Département sécurité informatique :** Ce département a pour mission d'assurer la sécurité des systèmes informatiques de l'entreprise. Il est chargé de (d') :

- héberger et sécuriser les applications de gestion,
- mettre en place des solutions de sécurité informatique,
- mettre à la disposition des directions métiers les moyens nécessaires à la gestion (accès aux applications métiers, Accès Intranet et Internet).

ANNEXE N° 2

ANNEXE 2

INSTALLATION D'OPENDIAMETER

1- Installation d'openDiameter

La mise en place du protocole Opendiameter requis l'installation des libréries externes et les outils suivants :

1.1 GNU g++ versions

1.2 Xerces C++ XML Parser

1.3 ACE library

1.4 BOOST library

1.5 OpenSSL library

1.6 Autoconf/Automake/libtool versions

Description des outils et des librairies externes nécessaire à l'installation du protocole :

1.1 GNU g++ versions

g + + est le surnom traditionnel de GNU C + +, un compilateur C + + librement redistribuable. Il fait partie de GCC.

Version actuelle 4.5.0

1.2 Xerces C++ XML Parser

Xerces-C + + est un analyseur de validation XML écrit dans un sous-ensemble portable de C + +. Xerces-C + + permet de donner facilement aux applications la capacité de lire et d'écrire des données XML. Une bibliothèque partagée est prévu pour l'analyse, la production, la manipulation et la validation des documents XML en utilisant le DOM, SAX, et les API SAX2.

L'analyseur fournit de hautes performances, la modularité et l'évolutivité Ver actuelle 3.1.0

1.3 ACE library

The ADAPTIVE Communication Environment (ACE), open-source orienté objet. ACE offre un riche ensemble d'environnements et emballage C++ réutilisables. Permet aux logiciels de communication de gérer le démultiplexage, la gestion d'événements d'expédition, traitement du signal, l'initialisation du service, la communication interprocessus, la gestion partagée de la mémoire, le routage des messages, configuration dynamique des services distribués, l'exécution simultanée et la synchronisation des processus.

Version actuelle 5.7

1.4 OpenSSL

Le projet OpenSSL est un effort collaboratif pour développer un solide, de qualité commerciale, complet, et Open Source.

OpenSSL est basé sur l'excellente bibliothèque SSLeay développée par Eric A. Young et Tim J.Hudson. La boîte à outils OpenSSL est autorisée en vertu d'une licence de type Apache, ce qui signifie essentiellement la libre utilisation pour des fins non commerciales et des fins commerciales sous réserve de certaines conditions de licence simple.

Version actuelle 1.0.0

Les versions actuelles des autres librairies sont :

- Boost: version 1.0.4
- Autoconf: version 2.57
- Automake: version 1.7.6

2- Les librairies d'openDiameter

1. libdiamparser - Diameter message parser
2. libdiameter - Diameter base protocol
3. libeap - EAP protocol implementation
4. libeaparchie - EAP Archie implementation
5. libpana - PANA protocol implementation
6. libdiametereap - Diameter/EAP library
7. libdiameternasreq - Diameter/NASREQ library
8. libdiametermip4 - Diameter/MIP4 library
9. libodutl - General support library for all protocols

ANNEXE N° 3

ANNEXE 3

2-1 Protocole PAP

(Password Authentication Protocol)

Le protocole PAP transmet simplement un mot de passe, sous la forme d une chaîne, de l'ordinateur de l'utilisateur final au périphérique du serveur d'accès au réseau. Lorsque ce dernier envoie le mot de passe, celui ci est crypté à l'aide du secret partagé de RADIUS sous la forme d'une clé de cryptage. Le protocole PAP est la solution la plus souple, car la communication d'un mot de passe « en clair » au serveur d'authentification permet à celui ci de le comparer avec pratiquement n'importe quel format de mémoire. Les mots de passe PAP peuvent être comparés avec ces chaînes en reproduisant la méthode de cryptage. Étant donné qu il utilise la version « en clair » du mot de passe, le protocole PAP présente quelques points faibles en termes de sécurité. Bien que le protocole RADIUS crypte le mot de passe pour le transmettre via Internet, il est transmis « en clair » sur la connexion à distance et décrypté au niveau du proxy RADIUS et du serveur RADIUS.

2-2 Protocole CHAP

(Challenge Handshake Authentication Protocol)

Le protocole CHAP est destiné à résoudre le problème des mots de passe transmis « en clair ». Dans le protocole CHAP, le périphérique du serveur d'accès au réseau envoie un défi sous forme de numéro aléatoire à l'ordinateur de l'utilisateur final. Le numéro aléatoire est alors crypté, selon un algorithme de hachage, avec le mot de passe de l'utilisateur. L'ordinateur client envoie le résultat obtenu comme réponse au défi du serveur d'accès au réseau et le périphérique du serveur d'accès au réseau transmet la réponse et le défi au moyen du protocole RADIUS.

Lorsque le serveur d'authentification reçoit le lot RADIUS, il utilise le défi et le mot de passe de l'utilisateur pour créer sa propre version de la réponse. Si les calculs du serveur correspondent à la réponse proposée par l'ordinateur de l'utilisateur, la requête d'accès est acceptée.

Le protocole CHAP réduit le problème lié à la transmission d'un mot de passe «en clair» sur Internet, car le mot de passe obtenu par l'algorithme de hachage et le défi ne peuvent pas être décryptés. Les réponses CHAP ne peuvent pas être réutilisées, car les périphériques du serveur d'accès au réseau envoient un défi unique à chaque tentative de connexion d'un ordinateur client.

Étant donné que l'algorithme de calcul des réponses CHAP est connu, il est très important que les mots de passe CHAP soient choisis avec soin et soient suffisamment longs. Les mots de passe qui correspondent à des noms communs ou propres sont vulnérables aux attaques effectuées à l'aide de

dictionnaires ; en effet, ils peuvent être découverts en calculant la réponse au défi CHAP pour chaque entrée d'un dictionnaire. Les mots de passe trop courts peuvent être découverts en calculant par essais successifs la réponse CHAP, et ce jusqu'à ce que l'une des réponses corresponde à celle de l'utilisateur.

Le protocole CHAP est le protocole d'authentification à distance le plus couramment utilisé, mais Il n'est pas pris en charge par tous les serveurs d'authentification. Lorsque le serveur ne stocke pas le même mot de passe que celui utilisé pour calculer la réponse CHAP, il ne peut pas calculer la réponse équivalente. Comme les clients CHAP utilisent un mot de passe « en clair » pour créer la réponse au défi CHAP, les mots de passe « en clair »doivent figurer sur le serveur pour que la réponse équivalente puisse être calculée. Par exemple, les serveurs Windows NT et UNIX stockent tous deux les valeurs hachées des mots de passe des utilisateurs et, par conséquent, ne peuvent pas reproduire le mot de passe « en clair » utilisé par l'ordinateur client pour créer la réponse au défi. Le serveur UNIX sera utilisé dans notre cas.

2-3 Protocole MS-CHAP

(Microsoft Challenge Handshake Authentication Protocol)

Microsoft a mis au point une version spécifique de CHAP, baptisée MS-CHAP (Microsoft Challenge Handshake Authentication Protocol version 1, noté parfois MS-CHAP-v1), améliorant globalement la sécurité.

En effet, le protocole CHAP implique que l'ensemble des mots de passe des utilisateurs soient stockés en clair sur le serveur, ce qui constitue une vulnérabilité potentielle.

Ainsi MS-CHAP propose une fonction de hachage propriétaire permettant de stocker un hash intermédiaire du mot de passe sur le serveur. Lorsque la machine distante répond au défi, elle doit ainsi préalablement hacher le mot de passe à l'aide de l'algorithme propriétaire.

Le protocole MS-CHAP-v1 souffre malheureusement de failles de sécurité liées à des faiblesses de la fonction de hachage propriétaire.

Les solutions apportées par MS-CHAP v 2 aux problèmes rencontrés dans MS-CHAP sont les suivantes :

1. Avec le **MS-CHAP**, Le codage par LAN Manager de la réponse utilisée pour la compatibilité avec des versions antérieures des clients d'accès distant Microsoft est faible du point de vue cryptographique, alors que **MS-CHAP v2** ne permet plus le codage des réponses par LAN Manager.

2. Avec le **MS-CHAP**, le codage par LAN Manager des modifications de mots de passe est faible du point de vue cryptographique, alors que le **MS-CHAP v2** ne permet plus le codage des modifications de mots de passe par LAN Manager.
3. Avec le **MS-CHAP**, seule une authentification unidirectionnelle est possible. Le client d'accès distant ne peut pas vérifier s'il essaie de se connecter au serveur d'accès distant de son entreprise ou à un serveur d'accès distant fictif, alors que le **MS-CHAP v2** offre une authentification bidirectionnelle, également désignée par le terme « authentification mutuelle ». Le client d'accès distant reçoit la confirmation que le serveur d'accès distant auquel il tente de se connecter a accès au mot de passe de l'utilisateur.
4. Avec le **MS-CHAP**, grâce au cryptage 40 bits, la clé cryptographique repose sur le mot de passe de l'utilisateur. Chaque fois que l'utilisateur se connecte à l'aide d'un même mot de passe, la même clé cryptographique est générée, alors qu'avec le **MS-CHAP v2** la clé cryptographique repose toujours sur le mot de passe de l'utilisateur et une chaîne d'interrogation arbitraire. Chaque fois que l'utilisateur se connecte à l'aide d'un même mot de passe, une clé cryptographique différente est utilisée.
5. Avec le **MS-CHAP**, Une clé cryptographique unique est utilisée pour les données envoyées dans les deux sens de la connexion, alors qu'avec le **MS-CHAP v2** des clés cryptographiques différentes sont générées pour les données envoyées et reçues.

Annexe N° 4

ANNEXE 4

L'INSTALLATION ET LA CONFIGURATION DU PROTOCOLE RADIUS

Comme on l'a précisé dans le chapitre précédent, il existe plusieurs solutions sur le marché, certaines commerciales, d'autres libres notre choix c'est port » sur FreeRADIUS pour la simple raison qu'il est libre sous Linux.

Il présente aussi l'avantage d'être très stable, de bénéficier d'une communauté très active au travers d'un site web et de listes de diffusion. Il s'agit d'une solution très largement répandue sur divers systèmes d'exploitation et apte à s'intégrer dans des environnements très variés grâce à sa compatibilité avec les standards les plus courants.

FreeRADIUS :

C'est une implémentation de RADIUS élaborée, à la suite du projet Cistron, par un groupe de développeurs. La scission entre les deux projets date de 1999. C'est un projet Open Source sous licence GPL.Le site officiel du projet est http://www.freeRADIUS.org

FreeRADIUS doit son succès à sa compatibilité avec un grand nombre de standards couvrant les systèmes d'exploitation, les protocoles et les bases d'authentification.Il est annoncé comme testé et fonctionnel sur les systèmes Linux (Toutes distributions), FreeBSD, NETBSD et Solaris.

Installation de FreeRADIUS :

L'installation de FreeRADIUS est simple et classique.

Le fichier archive se présente sous le nom : *freeRADIUS-version.tar.gz* et il s'installe comme suite :

$ Tar xvzf freeRADIUS-2.1.8

$ cd freeRADIUS-2.1.8

$ ./configure

$ make

$ make install

FreeRADIUS existe aussi sous forme de paquetage disponible suivant les distributions.

L'installation de FreeRADIUS sous une distribution UBUNTU se fait par la commande suivante :

$ apt-get install freeRADIUS

Soumission d'une requête :

Les requêtes arrivent au serveur par le biais de paquets Access-Request pouvant contenir plusieurs attributs. On y trouvera toujours l'attribut *User-Name* qui contient l'identifiant. On y trouve aussi des attributs tels que *Calling-Station-Id, NAS-Identifier,* etc...

Ces attributs sont appelés ***request-items.***

Recherche dans la base de données :

Le nœud central du système FreeRADIUS est la ou les bases de données où les informations d'autorisations et d'authentification sont puisées.

Par défaut la base d'autorisation de FreeRADIUS est le fichier *users.* Il s'agit d'un fichier plat et d'une forme simpliste de base de données séquentielle. Le fichier *users* peut être utilisé pour cumuler l'authentification et l'autorisation ou uniquement pour l'une ou l'autre.

Une base d'authentification est constituée d'une entrée par utilisateur ou par machine et contient les données d'authentification (mot de passe). Si base d'authentification et d'autorisation ne font qu'une, ces données d'authentification y seront présentes sous formes de check-items (User-Password).

FreeRADIUS recherche dans la base d'autorisation un identifiant égal à la valeur de *User-Name*. S'il le trouve, il compare la liste des *check-items* à la liste des *request-items*. S'il y a équivalence, l'entrée est validée, sinon l'entrée suivante est vérifiée et ainsi de suite jusqu'à la fin.

Les *check-items* constituent donc une liste de critères supplémentaires auxquels la requête doit satisfaire. Chaque attribut de la liste des *request-items* devra avoir pour valeur la valeur du *check-item* correspondant. Pour qu'une entrée soit validée, il ne suffit donc pas de trouver l'identifiant mais il faut que les *request-items* soient équivalents aux *check-items*. Ce qui veut dire qu'il est parfaitement possible d'avoir le même identifiant pour plusieurs entrées mais avec des check-items différents.

Les Principaux fichiers de configuration :

FreeRADIUS dispose de quelques fichiers de configuration pour décrire son environnement. Parmi ceux-ci, *clients.conf, RADIUSdconf, et* la base *users* sont incontournables. Tous ces fichiers sont placés, par défaut, dans le répertoire */etc/raddb.* Ils sont très commentés. Parmi les nombreuses options, seules celles qui ont le plus d'importance pour la compréhension des mécanismes seront détaillées.

Le fichier « Clients.conf » :

Ce fichier a pour fonction de définir les secrets partagés avec chaque équipement réseau. Cela revient à déclarer quels matériels (NAS) peuvent soumettre des requêtes au serveur FreeRADIUS. Tout autre matériel sera refusé et le message suivant sera émis dans le fichier de journalisation :

Ignoring request from unknown client adresse-ip-du-NAS Pour chaque NAS

La syntaxe est la suivante :

```
Client adresse-ip {
        Secret = le-secret-partagé
        shortname = nom
}
```

- *adresse-ip* est l'adresse IP du commutateur ou de la borne.
- *le-secret-partagé* est le secret partagé entre le serveur et cet équipement. On peut définir un secret différent pour chaque équipement. Il faudra enregistrer le même secret dans l'équipement.
- *nom* est un alias que l'on donne à cet équipement. Il peut être choisi librement et n'est pas forcément égal au nom DNS de l'équipement. Il est important de donner un nom différent pour chaque équipement car ils apparaîtront dans les journaux de FreeRADIUS pour indiquer à partir d'où un poste a réussi à s'authentifier.

La base « users » :

Le fichier *users* est la base de données locale. Elle est utilisée soit comme base d'autorisations, soit comme base d'authentification ou les deux à la fois. Il s'agit d'un simple fichier texte.

Quel que soit le type de base de données que l'on souhaitera utiliser au final, la connaissance du format et de l'usage du fichier *users* est fondamentale.

Format :

Ce fichier est constitué d'une liste d'entrées, chacune correspondant à un utilisateur ou à un utilisateur ou à une machine.

Le format de ces entrées est :

identifiant *<config-items>,<check-item>.....<check-item>*
reply-item,
reply-item,
.....
reply-item

Une entrée est composée de deux parties : la première ligne et les lignes suivantes, en retrait, jusqu'à la première ligne de l'entrée suivante.

Première ligne

identifiant est l'identité véhiculée par l'attribut *User-Name* dans un paquet *Access-Request*. FreeRADIUS balaye le fichier avec cette identité comme clé de recherche.

Les *config-items* sont toujours écrits sur la première ligne et sont séparés par une virgule.

Les *check-items* sont eux aussi toujours sur la première ligne, séparés par une une virgule. La ligne se termine sans virgule.

Lignes suivantes

Les *reply-items* sont écrits, à raison d'un par ligne, en retrait (espaces ou tabulations) par rapport à la première ligne. Chaque ligne se termine par une virgule sauf la dernière.

La figure suivante montre l'exemple d'une entrée dans le fichier *users :*

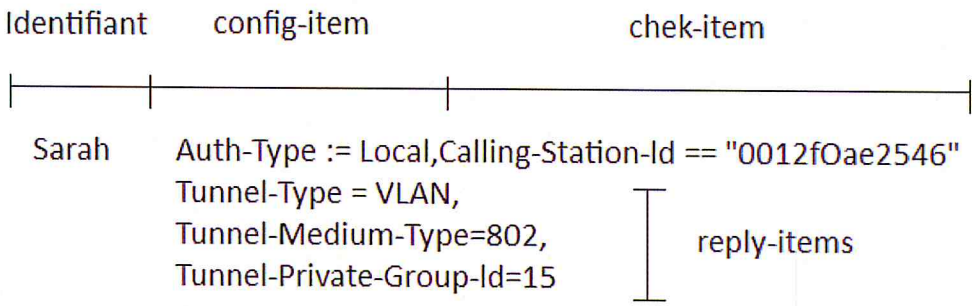

Figure N° 1

Le fichier « RADIUSd.conf » :

Ce fichier de configuration regroupe l'ensemble des paramètres nécessaires pour décrire le type de fonctions souhaitées.

RADIUSd.conf 'est composé de plusieurs parties :

- Paramètres du service RADIUSd.
- Section de déclaration des modules.
- Section Instantiate.
- Section Authorize.
- Section Authenticate.
- Section Pre-Acct.
- Section Post-Auth.
- Section Pré-Proxy.
- Section Post-Proxy.

Paramètres du service RADIUSd

On trouve ici des paramètres directement liés au fonctionnement de base du processus (daemon) RADIUSd :

- Déclaration des chemins d'accès : où se trouvent les programmes (prefix), les fichiers de configuration (sysconfig).
- Déclaration du port d'écoute : normalement 1812 ou bien 0 pour utiliser le port défini dans le fichier */etc/services.*
- Déclaration du nombre minimal de processus lancés au démarrage et du nombre maximal possible (thread_pool).
- Inclusion du fichier *clients, conf.*

Déclaration des modules :

Cette section commence par le mot-clé *modules* et contient la déclaration de tous les modules qui seront utilisés pendant l'exploitation.

Dans *RADIUSd.conf,* il existe un grand nombre de modules prédéfinis. Parmi ceux, celui qui nous intéresse ici :

- *files*, qui définit et charge le fichier *users.*

Section Instantiate :

Cette section n'est pas obligatoire et sert à précharger des modules comme *exec* qui permet l'exécution de programmes externes.

Section Authorize

Cette section contient la liste des modules qui doivent être exécutés afin de constituer la liste des *reply-items* et de préparer le terrain avant la section *Authenticate*. Ils positionnent le *config-item Auth-Type*, si cela n'a pas déjà été fait explicitement dans le fichiers *users,* et interrogent une base de données d'autorisation. Par exemple,

```
authorize {
        chap
        eap
        files
}
```

files: Le fichier *users* est parcouru à la recherche de l'identifiant contenu dans l'attribut *User-Name* et la liste *replay-items* est constituée.

Les autres sections :

La section *Post-Auth* intervient après l'authentification et permet d'y exécuter d'autres modules. Par exemple, il est possible d'y placer un programme externe qui enregistrera des informations dans un journal.

La section *Pré-Proxy* est exécutée avant d'envoyer une requête vers un autre serveur.

La section *Post-Proxy* est appelée lorsqu'une réponse revient.

Les autres fichiers de configuration :

Il existe d'autres fichiers de configuration dont l'usage dépond des spécificités du site sur lequel FreeRADIUS sera déployé.

ANNEXE N° 5

ANNEXE 5

LA CONFIGURATION DE NETFILTER

La configuration du firewall Netfilter reste très complexe sachant les différentes situations, conditions et architectures.

et sécurisé. Nous verrons au fur et à mesure comment protéger un poste utilisateur, un serveur et enfin un réseau.

Pour pouvoir utiliser les fonctionnalités Netfilter du noyau Linux, il faut activer les options suivantes lors de la configuration du noyau.

Nous avons utiliser le kernel 2.6.10, notamment pour sa robustesse avec le patch GRsec 2.1.0 qui est disponible pour cette version du kernel .

Grsec est sur http://www.grsecurity.net/:

http://www.grsecurity.net/grsecurity-2.1.0-2.6.10-200501081640.patch (pour le 2.1.0 pour le kernel 2.6.10)

Le kernel est disponible sur : ftp.kernel.org

ftp://ftp.kernel.org/pub/linux/kernel/v2.6/linux-2.6.10.tar.gz (pour le kernel 2.6.10)

Ensuite, on a déplacer les fichers grsecurity -2.1.0-2.6.10-200501081640.patch et linux-2.6.10.tar.gz archives dans /usr/src puis on tape :

tar zxvf linux-2.6.10.tar.gz

patch -p0 < grsecurity-2.1.0-2.6.10-200501081640.patch

Maintenant on a eu une arborescence des sources du kernel patchées avec GRsec.

Ensuite :

cd linux-2.6.10 && make menuconfig

On a Configurer notre kernel selon les besoins de notre machine et on a Activés les options suivantes:

Device Drivers -> Networking support -> Networking Options -> Network Packet Filtering (replace ipchains) -> IP : Netfilter configuration

Toutes les options et sous options, pas de soucis, même si globalement iptables support / NAT / MASQUERADE / IP range match support / Limit match support / connection state / Reject / NAT of local connexion / Log.

(Si vous avez patché avec GRsec) :

Security Options

-> GRsecurity

-> Adress Space Protection

-> Deter Exploit brutforcing

-> FileSystem protection

-> Chroot jail restriction (tout)

-> executable protection

-> randomized PID

-> Network protection

-> (larger entropy pool, truly random tcp isn, random IP ids)

-> sysctl support

-> sysctl support

-> sysctl support

-> PaX

-> PaX control

-> Use ELF program header Marking

-> Non executable pages

-> enforce non-executable pages

-> segmentation based non-exec pages

-> enforce non-executable kernel pages

-> address space layout randomization

-> Randomize kernel stack base.

Enfin :

make bzImage && make install

Annexe N° 6

ANNEXE 6

COMMENT INSTALLER JPCAP

A PARTIR DU CODE SOURCE SOUS LINUX

JPCap est un open - source Java bibliothèque publié sous la GNU LGPL et conçu pour permettre la capture de paquets réseau en Java.

JPCap est développé par Keita Fujii à l'Université de la Californie et Irvine.

La page d'accueil est JPCap http://netresearch.ics.uci.edu / kfujii / jpcap / doc /

JPCap est une bibliothèque Java Native de mise en œuvre (JNI) de la bibliothèque libpcap populaire et doit à cet effet travailler sur n'importe quel OS qui supporte libpcap. libpcap est un utilisateur

- la capture de paquets au niveau de bibliothèque qui fournit un système commun.

- API indépendante pour le niveau réseau de surveillance-bas. libpcap est également open - source.

Logiciel développé et maintenu par TCPDump.org.

Nous avons utilisé JPCap et libpcap sur le système d'exploitation Linux, et les bibliothèques de travail sur Linux.

JPCap supporte les types suivants de données réseau:

Niveau 2: Ethernet datagrammes

Niveau 3: IPv4 et IPv6 ARP / RARP

TCP, UDP et ICMPv4.

JPCap reconnaît les types de paquets énumérés ci-dessus, mais peut capturer n'importe quel type de trafic réseau en tant que (C.-à-premières de paquets, comme une instance de la classe Packet) qui contient l'ensemble des données par paquets. Cette fonctionnalité permet aux applications Java d'analyser n'importe quel type de paquet.

Étape 1:

Préparation du système

Voici les instructions pour installer JPCap sur une nouvelle installation de Ubuntu 7.10.

On a installé les logiciels nécessaires au développement pour créer un environnement de développement utilisables:

On a installé GNU du compilateur de base des bibliothèques de la % sudo apt - get install build-essential

On a installé la bibliothèque pcap linux

% sudo apt - get install libpcap0.8

On a Installé le SDK Java de Sun

% sudo apt - get install soleil - java6-sdk

Télécharger la source JPCap et l'extraire dans le répertoire de travail

http://netresearch.ics.uci.edu / kfujii / jpcap / jpcap - 0.7.tar.gz

Étape 2:

Construction et installation de la bibliothèque JNI JPCap

Dans une fenêtre de terminal, on a accédés à l' l'nagivate] jpcap [/ répertoire / c src. Par exemple:

/ Jpcap - 0.7/src/c

[IMPORTANT] Modifier le Makefile

Avec la commande suivante:

$ Find / usr - jni.h nom

/ Usr / lib / jvm / java-6 - soleil - 1.6.0.03/include / jni.h

/

Le Makefile doit alors lire

= / Usr / lib / jvm / java-6 - soleil JAVA_DIR - 1.6.0.03

J

Dans la fenêtre de terminal dans le cadre du] jpcap [/ src / répertoire de type C qui permet de créer la bibliothèque partagée

$ Make

Cela va créer le fichier 'libjpcap.so.

Le système de copie »libjpcap.so JNI répertoire de la bibliothèque Java

« [Java-dir] / jre / lib / <arch> 'où <arch> est soit« i386 »ou« sparc ».

Étape 3:

Construction et installation du fichier JAR JPCap

Dans une fenêtre de terminal, à l'nagivate] jpcap [/ répertoire / src java. Par exemple:

/ Jpcap - 0.7/src/java

Il devrait y avoir un sous-répertoire nommé 'jpcap'

Tous les fichiers. Java dans le dossier des «jpcap» et «jpcap Compiler / paquet"répertoires.

$ Find. - "*. Java" - exec (javac) \ nom;

On a Créer le fichier JAR

jar - jpcap jpcap.jar FC

j

On a vérifié le contenu du fichier JAR avec la commande suivante

jar - jpcap.jar tvf

j

on a Copier le nouveau fichier jpcap.jar aux extensions Java répertoire

«Jpcap.jar cp $ [Java] dir / jre / lib / ext /

Maintenant on est prêt à commencer l'utilisation de JPCAP Java Native Interface Library

Author:

Keita Fujii (kfujii@uci.edu)

Home page:

http://netresearch.ics.uci.edu/kfujii/jpcap/doc/index.html

GLOSSAIRE

AAA	Authentification, Autorisation Comptabilisation (Accounting)
ACS	Access Control Server (De Cisco Sous Windows)
ADIF	Accounting Data Interchange Format
API	Application Programming Interface
AT	Algérie Telecom
CHAP	Challenge Handshake
IAS	Internet Authentication Service
JRE	Java Runtime Environment
JSP	Javaserver Pages
JVM	Java Virtual Machine
NAS	Network Access Server
OS	Systèmes D'exploitation
OSI	Open System Interconnection
PAP	Password Authentification Protocole)
SI	Systèmes D'information
TACACS+	Terminal Access Controller Access Control System Plus
UDP	User Datagram Protocol
UML	Unified Modeling Language
UP	Processus Unifié
VLAN	Virtual Local Area Network

BIBLIOGRAPHIE

[1] http://www.commentcamarche.net/contents/secu/secuintro.php3

[2] http://www.securite-reseaux.com/politique_securite.html(politique

[3] **CHARMAT H. et OUALI A.**, Réalisation d'une plateforme de simulation d'attaque informatique, PFE, EMP Bordj El Bahri, 2010

[4] http://www.loria.fr/~ichris/Teaching/ESIAL/ESIAL3/TPESR/AAA.pdf

[5] http://fr.wikipedia.org/wiki/Protocole_AAA

[6] wikipedia.org/wiki/TACACS

[7] http://dictionnaire.sensagent.com/protocole+aaa/fr-fr

[8] http://diameter.sourceforge.net/

[9] **Serge Bordères** ; Authentification réseau avec Radius : 802.1x, EAP, FreeRadius ; Eyrolles ; 23 novembre 2006.

[10] http://fr.wikipedia.org/wiki/Cat%C3%A9gorie:Protocole_d'authentification (CHAP)

[11] http://www.irisa.fr/prive/bcousin/Cours/03-UDP.fm.pdf

[12] http://www.lifl.fr/~carle/dnld/iut/RSX06-TCPUDP_6ppp.pdf

[13] http://www.commentcamarche.net/contents/protect/firewall.php3

[14] http://depinfo.mines.inpl-nancy.fr/Members/lahmadi/cours.Firewall.pdf

[15]http://www.netfilter.org/

[16] http://www.netfilter.org/projects/iptables/index.html

[17] http://www.c-sait.net/cours/iptables.php

[18] http://netresearch.ics.uci.edu/kfujii/jpcap/doc/index.html

[19] Rémy Fannader Hervé Leroux, UML Principes de modélisation,Edition.DUNOD

[20] **Joseph Gabay**, MERISE vers OMT et UML, InterEditions

[21] **Michel Lai,** Penser Objet avec UML et JAVA, InterEditions

[22] **Piechocki, Laurent.** UML, le langage de modélisation objet unifié. s.l. : Developpez.com, Octobre 2007.

[23] WaveRider by Vecima Support: http://www.wr.vecimasupport.com

[24] FreeRADIUS web site: http://www.freeradius.org/

[25] http://www.penguin-soft.com/penguin/man/8/freeradius.html

[26] FreeRADIUS Configuration example: http://www.frontios.com/freeradius.html

[27] MySQL web site: http://www.mysql.com/

[28] MySQL tutorial: http://dev.mysql.com/doc/refman/5.0/en/tutorial.html

[29] MySQL statement syntax: http://dev.mysql.com/doc/refman/5.0/en/sql-syntax.html

[30] https://help.ubuntu.com/community/CronHowto

[31] http://crunchbang.org/archives/2007/10/26/howto-setup-a-crontab-file/

[32] Ubuntu cron help. [En ligne] https://help.ubuntu.com/community/CronHowto.

[33] **Damoiseaux, Jean-Luc.** TP WIFI Radius-EAP. s.l. : Dpt R&T IUT Aix en Provence, Juillet 2009.

[34] **Adhara.** Systèmes et Réseaux - Conception d'une infrastructure réseau Windows Server 2008. Avril 2010.

[35] **Cariou, Eric.** Sockets TCP/UDP et leur mise en oeuvre en Java. s.l. : Université de Pau et des Pays de l'Adour Département Informatique, Décembre 2006.

[36] **Micro, Trend.** SÉCURITÉ DES POINTS FINAUX. 2008.

[37] **CHRISMENT, Isabelle.** Protocole AAA Principes et implantations. s.l. : ESIAL, Avril 2003.

[38] **MORELLE, Albin.** Poly Java v 3.6. s.l. : Esiee, Juillet 2009.

[39]. **Erve, John Vant.** Nortel IP Telephony Deployment Technical Configuration Guide. s.l. : Nortel IP Telephony Deployment, Septembre 2009.

[40] **RICHARD, Abederahim BIBA Fabien GRABHERR Eric LARONDE-LARRETCH Yohann.** Modélisation avec UML. s.l. : ISTIA (DESS QUASSI), Février 2000.

[41] **Bezsilko, Quentin.** Mise en place d'un service de téléphonie dans un réseau local personnalisé. s.l. : IUT A de Lille I, Juin 2009.

[42] **CARON, Dominique.** Mise en place d'un serveur Radius pour authenti cation sur une serveur VPN sous LinuX . s.l. : Université MONTPELLIER II, Avril 2008.

[43] **N., Melab.** Les sockets java . s.l. : Université de Lille1, Juillet 2002.

[44] **Bulfone, Christian.** La mise en place d'un routeur Parefeu entre des réseaux privés. s.l. : Université Pierre Mondès FRANCE, 2009.

[45] **Cocquebert, Cédric.** L'API – Socket en JAVA. s.l. : Supélec, Novembre 2003.

[46] **Tomac, Siniša, Skorin-Kapov, Marko Sikirica Lea et Skorin-Kapov, Marko Sikirica Lea.** Implementation of the Diameter-based Cx Interface in the IP Multimedia Subsystem. s.l. : Université de Zagreb, Avril 2006.

[47] **BARRIER, François.** Guide d'installation de FreeRadius avec EAP-TLS + MySQL. s.l. : La Goutte Alexis, Avril 2005.

[48] **Razi, Csar.** Gestion d'identité et contrôle d'accès. s.l. : CISCCO, Janvier 2010.

[49] www.ubuntu.com

[50] www.eclipse.org