Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université Virtuelle de Tunis

UVT
جامعة تونس الافتراضية
UNIVERSITÉ VIRTUELLE DE TUNIS

MEMOIRE

DE STAGE DE FIN D'ETUDES

Pour l'obtention du

«Mastère professionnel en Nouvelles Technologies des Télécommunications et Réseaux (N2TR)»

Présenté par :

Raja Homri

Conception et Développement d'un Système d'analyse et Reporting des Fichiers Logs

Devant le jury :

Encadreur : Mme Ahlem Ben Hassine……

Rapporteur :……

Membre : ……

Année Universitaire : 2015 / 2016

www.uvt.rnu.tn
14, Rue Yahia Ibn Omar - 1082 Mutuelleville - Tunis
Tel. : +216 71 28 99 81 / +216 71 89 17 31 - Fax : +216 71 89 26 25

Dédicaces

À mes bien-aimés parents,

Pour leur soutien sans faille,

À mes chères sœurs,

À toute mon adorable famille,

À tous ceux qui m'ont aidé

À franchir les épreuves difficiles,

Aimablement…

Remerciements

Il nous est agréable d'exprimer notre gratitude à tous ceux qui ont participé de près ou de loin à la réalisation de ce projet de fin d'études.

Nos remerciements s'adressent en premier lieu à mon encadreur Mme Ahlem Ben Hassine de l'honneur qu'elle m'a fait en encadrant mon projet, de la gentillesse qu'elle a éprouvé, de ses idées et de l'attention qu'il porte à ses stagiaires, tout en conduisant leurs travaux dans le respect mutuel et la confiance.

C'est aussi avec un grand plaisir que nous pressentons spécialement nos chaleureux remerciements à Monsieur Yassine Bouchakoua qui m'a soutenu tout au long de l'élaboration de mon travail et qui n'a pas hésité à me fournir les documents et les informations nécessaires à cette fin.

Enfin je tiens également à remercier tous nos enseignants du département informatique, pour leurs précieux conseils et l'effort qu'ils n'ont cessé de nous prodiguer au cours de notre formation, aussi bien les membres du jury qui ont l'aimablement d'évaluer notre travail.

تلخيص

إن مشروع ختم الدروس يتمثل في صياغة و تطوير نسق تصرف تحليلي لمختلف ملفات تسجيل الأثر "لوج" للجدران النارية. هذا الإنتاج من شانه أن يوفر معلومات و تحاليل حول تصفية المبادلات بين الشبكات. و يمكن أيضا أن يقوم بمراقبة و تحليل مختلف أنواع "لوج".

الكلمات المفاتيح: "جدار ناري""فبروول""لوج""جفا"

RESUME :

Notre projet consiste à concevoir et élaborer un système de gestion et d'analyse des fichiers journaux des pares feu. Ce produit peut fournir des informations et des statistiques sur le trafic filtré sur le réseau. Il peut également contrôler et analyser de divers types de fichiers journaux des pares feu. Ce projet a été réalisé dans l'environnement JAVA.

MOTS CLES : fichier journal, pare feu, JAVA.

ABSRACT:

Our project consists to conceive and elaborate a system of management and analyzes of firewall's log file. This product can give information and statistics about network traffic. It can also manage and analyze various types of firewall's log file. This project was realized in JAVA environment.

KEY-WORDS: Log file, firewall, JAVA

Sommaire

Liste des Figures

Introduction Générale

Avec la croissance explosive d'Internet, la sécurité informatique est devenue de plus en plus importante pour les entreprises et les organisations qui ne cessent de stocker et de transférer un tas immense de données critiques à travers leurs réseaux locaux ou étendus.

Même ces organisations qui n'ont pas de secrets spécifiques, constituent une cible potentielle pour plusieurs types d'attaques qui peuvent aboutir à une perte considérable de données ou de services.

Les systèmes informatiques sont devenus de plus en plus complexes et hétérogènes : des serveurs de constructeurs différents se côtoient, ainsi que des routeurs, des commutateurs des pares feux et des systèmes de détection d'intrusion. Tous ces équipements fonctionnent avec des systèmes de plus en plus variés et complexes et les réseaux viennent interconnecter tous ces éléments entre eux. C'est ainsi que dans chaque élément, chaque système et chaque réseau, se posent de nombreux problèmes de sécurité.

Une maîtrise de la sécurité de fonctionnement des systèmes d'information est une garantie de la légalité, la fiabilité et la pertinence des transactions opérées. Cela nécessite un contrôle s'appuyant nécessairement sur l'enregistrement systématique et temporaire d'un certain nombre d'informations qui caractérisent chaque transaction.

Les systèmes d'information permettent cette facilité en enregistrant leurs transactions dans des fichiers appelés fichiers journaux ou fichiers traces (en anglais log files). Ces traces ont plusieurs objectifs :

Tout d'abord, **la métrologie du réseau** qui consiste à contrôler le volume d'utilisation des ressources, détecter des anomalies, faire évoluer les équipements en fonction des besoins. Puis **la vérification** que les règles en matière de système de sécurité informatique sont correctement appliquées, et que la sécurité d'information et du réseau, telle qu'elle a été définie par la

politique de sécurité de l'unité, est assurée. Ensuite, **la détection de toute défaillance ou anomalie de sécurité**, volontaire ou accidentelle, d'origine matérielle ou humaine. Enfin, **la détection de toute violation de la loi ou tout abus d'utilisation** des moyens informatiques.

Les objectifs précités imposent d'aller au-delà là d'un simple enregistrement des données des fichiers journaux. Ils impliquent nécessairement l'enregistrement, la conservation temporaire et l'exploitation des données collectées afin de générer des rapports statistiques permettant de contrôler l'efficacité des ressources matérielles et logicielles de l'organisme.

Notre projet de fin d'études consiste à concevoir et mettre en place une application d'analyse des fichiers journaux des pares feu dans le but de favoriser le contrôle des ressources matérielles et logicielles de l'organisme.

Afin de présenter ce rapport, nous avons choisi de le découper en quatre chapitres :

Dans le **premier chapitre**, présentation générale, nous mettons notre sujet dans son contexte général. Ensuite nous présentons l'organisme d'accueil; enfin, nous présentons notre application avec ses problématiques différentes.
Dans le **deuxième chapitre,** étude préalable et spécification des besoins, nous analysons les besoins fonctionnels et non fonctionnels de notre application ainsi que nous faisons une étude détaillée des différents cas d'utilisation possibles avec leurs descriptions
Le **troisième chapitre**, la conception, nous présentons l'architecture de notre application et l'étude détaillée de chaque package avec les différents scénarios possibles afin de concevoir une application réalisable, fiable et efficace.
Le **quatrième chapitre**, la réalisation, la réalisation, nous présentons l'environnement matériel et logiciel dans lequel le projet a été réalisé. Ainsi que les différents choix techniques liés à cette application, et les contraintes de réalisation. Nous terminons par la présentation de l'interface de l'application.

Chapitre I

Présentation Générale

I.1. Introduction

Dans ce chapitre, nous allons présenter la société d'accueil « Telecom Information Services **T.I.S »** au sein de laquelle nous avons effectué notre projet. Ensuite la description du contexte du projet, la problématique et le travail à réaliser. Enfin, nous exposons la méthodologie de travail admise.

I.2. Présentation l'organisme d'accueil

I.2.1.Présentation

Telecom Information Services **T.I.S** est une société d'expertise en ingénierie informatique et elle offre une large gamme de solutions complètes parfaitement adaptées aux besoins des entreprises.

Depuis sa création, elle a pris en charge des études et des réalisations de plus en plus complexes dans le domaine de l'expertise et des solutions informatiques pour les entreprises. Elle est basée sur l'expertise de ces ingénieurs qui lui permet la prise en compte globale du système d'information de l'entreprise : la rédaction d'un cahier de charge, en passant par la fourniture de matériels ou logiciels, le développement logiciel et l'ingénierie.

I.2.2.Objectifs et orientations de l'entreprise

Telecom Information Services opère sur plusieurs domaines d'activités. Parmi ces aspects nous pouvons citer :

- **Sécurité informatique**

Les prestations de sécurité informatique, selon Telecom Information Services, couvre :

- l'audit, l'élaboration et la conception de réseaux informatiques sécurisés ;
- la formation spécialisée pour administrateurs de réseaux et auditeurs ;

-la définition d'une politique de sécurité (règles d'implantation, comportement, architecture des réseaux et systèmes, accès aux applications et à Internet, gestion des emails, sécurité des postes de travail et des portables et préventions des virus).

- **Travaux d'analyse et Développement**

Telecom Information Services propose à toute entreprise de déléguer la gestion de leurs projets, mais aussi lui confier des tâches d'analyse telle que : l'évaluation des offres de services du marché nécessaires à la mise à niveau des équipements.

- **Interventions sur site**

L'équipe technique de Telecom Information Services assure les prestations de l'installation complète de systèmes informatiques et les services d'intervention sur site.

I.2.3.Organigramme

La figure I.1 représente l'organigramme de la société.

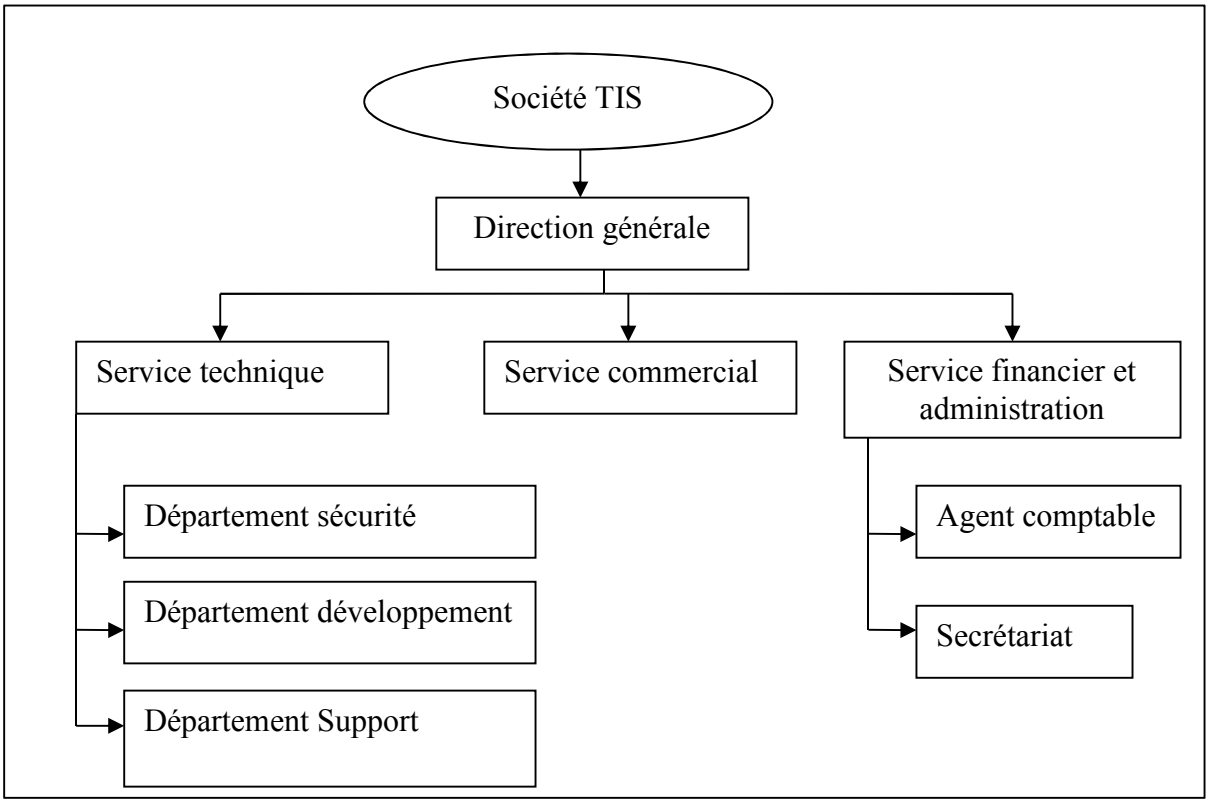


Figure I.1 Organigramme de l'organisme d'accueil

I.3.Contexte de projet

La sécurité est devenue une composante cruciale des systèmes d'information. Cependant, les organisations sont peu ou pas protégées contre les attaques, et leurs réseaux sont exposés aux

menaces externes et internes. En effet, la politique de sécurité est complexe, on y trouve notamment des faiblesses dues à la gestion et à la configuration des systèmes.

Un système d'analyse des fichiers journaux permet de collecter toutes les informations utiles sur la tentative, les sauvegardes pour des corrélations ultérieures. Ces fichiers logs seront enregistrés et traités en vue de réaliser des statistiques et fournir un tableau de bord qui aidera le responsable de sécurité au contrôle et à la prise de décision.

Dans ce contexte, ce projet vise à développer un logiciel de gestion de pare feu qui tourne sur n'importe quelle plateforme et qui prend en considération la diversité des types de pares feu pouvant être déployés sur le réseau.

I.4.Problématique

Dans cette nous allons exposer les risques et les problèmes de non surveillance de réseau et la complexité des systèmes existants :

-**Non surveillance des risques encourus** : Les services en ligne, utilisant Internet ou les technologies IP, mis en place par les entreprises ne cessent de se développer. Le besoin de sécurité et de traçabilité pour minimiser la vulnérabilité d'un système contre les menaces accidentelles ou intentionnelles.

D'une manière générale, les experts ont classé les risques informatiques que l'organisation peut courir en deux types selon les natures et leurs causes :

- Les attaques intentionnelles : Tout équipement informatique connecté à un réseau informatique peut être potentiellement vulnérable à une attaque :

Sur Internet, des attaques ont lieu en permanence, à raison de plusieurs attaques par minute sur chaque machine connectée. Ces attaques sont en généralement lancées automatiquement à partir de machines infectées (virus, chevaux de Troie, etc.), à l'insu de leur propriétaire et plus rarement par des pirates informatiques. C'est la raison pour laquelle il est absolument impératif d'installer des dispositifs de sécurisation, à savoir, des pare-feu afin de faire barrière entre l'ordinateur et le réseau.

- Les défaillances logicielles et matérielles : A part les attaques volontairement lancées contre les systèmes informatiques des organismes visant à compromettre leur fonctionnement normal, ces systèmes restent toujours vulnérables vis-à-vis des défaillances logicielles et matérielles.

-**Non surveillance des politique des sécurités** : Une politique de sécurité bien définie, bien mise en place peut ne servir à rien et devenir obsolète si on ne peut pas garantir qu'elle est bien respectée par le personnel de l'organisme, car même les théories les plus pertinentes ne valent

rien si elles sont mal appliquées. La surveillance du respect de la politique de sécurité se fait à travers l'identification des failles de sécurité issues d'une mauvaise configuration des systèmes et l'analyse des activités des utilisateurs et du système afin de détecter d'éventuels problèmes.

La complexité et la grande taille des fichiers Log : Les évènements journalisés générés par les équipements deviennent de plus en plus difficiles d'exploiter ou d'en faire un reporting pertinent vu en plus, les structures propriétaires des fichiers log du différent firewall nécessiteront la mise en place d'un outil à la fois efficace et générique pour le filtrage des logs et la gestion centralisée et en temps réel des informations de sécurité.

I.5.Travail à réaliser

L'objectif de ce travail est de concevoir et mettre en place une application permettant une analyse centralisée des fichiers journaux. Ces derniers seront intégrés dans une base de données relationnelle. Leur analyse et la génération de rapports adéquats permettent aux administrateurs d'avoir une vue claire sur l'utilisation des ressources et les guident dans l'élaboration d'une politique de sécurité adaptée aux besoins

I.6.Méthodologie de conception à utiliser

Avant la réalisation d'un projet informatique, il est nécessaire de fixer une méthodologie de travail afin d'aboutir à la fin à un logiciel fiable.

Le model du cycle de vie adopté pour la réalisation de ce projet est le model en V comme le montre la figure I.2.

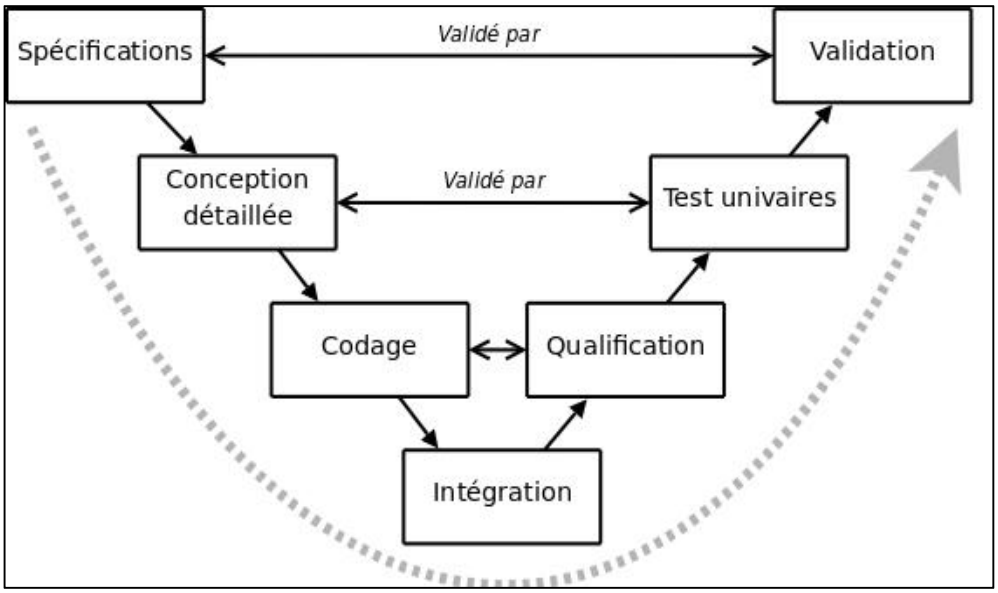


Figure I.2 Le model du cycle de vie en V

Le modèle en V demeure actuellement le cycle de vie le plus connu et certainement le plus utilisé. Il s'agit d'un modèle en cascade dans lequel le développement des tests et des logiciels sont effectués de manière synchrone.

Le principe de ce modèle est qu'avec toute décomposition doit être décrite la recomposition et que toute description d'un composant est accompagnée de tests qui permettront de s'assurer qu'il correspond à sa description. Ceci rend explicite la préparation des dernières phases (validation-vérification) par les premières (construction du logiciel), et permet ainsi d'éviter un écueil bien connu de la spécification du logiciel : énoncer une propriété qu'il est impossible de vérifier objectivement après la réalisation.

La mise en place de notre application passe par les phases suivantes :

-Spécification des besoins : Où nous se proposons de recueillir et de déterminer les fonctionnalités du système et ces interactions avec ces utilisateurs.

-Analyse des besoins : Dans cette étape, notre tâche principale sera de spécifier l'architecture logicielle de notre application et de la décomposer tout en spécifiant les interactions de ces composants logiciels entre eux pour subvenir aux besoins des acteurs du système.

-Conception : Cette étape sera réservée à la conception proprement dite du système en affinant la spécification des composants de l'application et leurs interactions à travers des diagrammes permettant de représenter ces derniers en précisant la chronologie des échanges des messages durant l'exécution du système et de représenter l'exécution des opérations relatives à une utilisation spécifique du système.

-Réalisation : Cette étape est réservée à l'implémentation du système dans l'environnement logiciel choisi, ainsi qu'au déploiement de l'application développée dans l'environnement des utilisateurs.

Après le choix de la méthodologie, nous avons besoins, d'un langage de modélisation unifiée pour la modélisation de notre projet. Pour concevoir notre système, nous avons choisi UML comme un langage de modélisation.

Notre choix s'est basé sur les points forts de ce langage notamment sa standardisation et les divers diagrammes qu'il propose. Aussi UML présente le meilleur outil pour schématiser des systèmes complexes sous un format graphique et textuel simplifié et normalisé.

En effet, UML n'est ni un processus ni une démarche, d'où il fallait choisir un langage de conception et de développement que nous devons l'adopter.

I.7.Conclusion

Dans ce chapitre nous avons présenté le cadre général de notre projet, ainsi que la méthodologie de travail. Dans le chapitre suivant nous nous focaliserons sur la spécification des besoins et des exigences fonctionnelles et non fonctionnelles de notre projet.

Chapitre II

Etude Préalable et Spécification des Besoins

II.1.Introduction

Dans ce chapitre, nous introduisons la première étape de notre processus de développement, en présentant l'étude préalable et la spécification des besoins. Tout au long de ce chapitre, nous allons décrire l'existant, définir les besoins fonctionnels et non fonctionnels du système, puis nous allons essayer d'exprimer les besoins sous forme de diagrammes de cas d'utilisation.

II.2.Etude préalable

Dans cette section, nous allons définir et critiquer la méthode de travail utilisée au niveau de l'entreprise

II.2.1.Description de l'existant

Les responsables de sécurité du système d'information des entreprises font l'erreur de non contrôler leurs pares feu. Ils les déploient, les testent puis les oublient. La mise en place d'une procédure de révision des journaux des pare-feu aidera à identifier les éventuels problèmes de configuration

Les entreprises sont confrontées quotidiennement à différents types de menaces tel que :

- Les tentatives d'intrusion non ciblées (virus).
- Les scans des machines.
- Des attaques de dénies de services à travers l'ouverture successive des ports due à la réception de nombre important de mail sans sujet ni message.

Ces attaques provoquent une activité intense des pares feu ainsi que des risques de ralentissement des machines dans les réseaux.

Contre ces menaces, les responsables de sécurité déploient plusieurs méthodes telles que le filtrage, l'antivirus, les systèmes de détection d'intrusion (IDS) et l'analyse des journaux.

Les fichiers journaux permettent :

- D'indiquer une activité hostile ou inhabituelle.
- Une analyse complète des paquets entrants et sortants.
- L'analyse des en-têtes.
- La détermination des règles ayant généré le blocage.
- D'avoir une vue des scans qu'une entreprise peut subir depuis l'extérieur du réseau.
- De pouvoir déceler les erreurs de configuration des machines du réseau.
- De pouvoir déceler les problèmes de filtrage des pares feu.

Il existe plusieurs types de pares feu tel que les pares feu de filtrages, de paquets, de type session et les pares feu Proxy dont leurs mise en place au sein d'un réseau local permettra d'hiérarchiser les niveaux de sécurité.

Chaque type de pares feu générera des fichiers journaux de structure propriétaire et non formalisé, ce qui rendra encore plus complexe la tâche d'analyse et l'exploitation des journaux.

En plus, la mise en œuvre des pares feu selon les départements et les directions dans le réseau local des entreprises impliquera la nécessité d'un processus de collecte, de gestion et d'analyse centralisée des événements journaliers (signalés par les éléments de sécurité) et ainsi automatiser le contrôle et le suivi continu de la politique de sécurité.

Nous allons décrire des exemples des solutions existent sur le marché tel que Firewall Analyser et FortiAnalyzer :

- **Firewall Analyzer** est un agent d'analyse des journaux et des logiciels de gestion de configuration qui permet aux administrateurs réseau de manière centralisée collecter, archiver , d'analyser les logs et de générer les rapports. Firewall Analyzer est indépendante du fournisseur, il prend en charge la quasi-totalité des pare-feu comme Check Point, d'autres dispositifs de sécurité connexes Juniper Fortinet, Snort, SonicWALL, Palo Alto et Cisco. Cependant, ses principales limites sont :

  - Rapport qualité/prix.
  - Très difficile à configurer.
  - Peut parfois classer des connexions légitimes en tant qu'illégitimes.

- **FortiAnalyzer** consolide les fonctions de journalisation, d'analyse et de reporting réseau au sein d'un seul système pour assurer une analyse centralisée des évènements de sécurité, et offrir des fonctions d'expertise post-incident (forensics), de reporting, d'archivage de contenu, de datamining, de mise en quarantaine des fichiers infectés et de gestion des vulnérabilités. Cependant, ses principales limites sont :

  - Rapport qualité/prix.

  -Logiciel fait pour les équipements fortigate.

II.2.2.Critiques de l'existant

L'étude de l'analyse des fichiers journaux dans les organisations nous a permis de déceler des anomalies dans la discipline, qui peut réduire l'efficacité de l'administration du système d'information et même mettre en péril la politique de sécurité mise en place. Ces lacunes sont essentiellement dues à la nature même des fichiers de journalisation et aux approches adoptées pour les analyser et les gérer. Parmi ces lacunes nous citons :

- **La négligence des fichiers journaux** : Beaucoup d'administrateurs de système d'information ne donnent aucune importance à la collecte, la centralisation et l'analyse des fichiers journaux de leurs équipements parfois par manque de sensibilisation à leurs importances et souvent à cause de l'absence d'outils adéquats pour la tâche.
- **La diversité phénoménale des formats des fichiers journaux** : Une simple recherche permet de déceler le nombre impressionnant des formes des fichiers journaux. Ces formats, souvent incompatibles et différents, nécessitent tous d'être analysés. Le problème ici est que les outils d'analyse offerts par les constructeurs ne permettent presque jamais l'analyse que du format de leurs produits et si nous suivons cette tendance nous aurons des dizaines d'outils d'analyse des fichiers journaux installés chez chaque administrateur du système d'information d'une organisation. De ce fait un outil générique d'analyse des fichiers journaux s'avère indispensable.
- **Le manque d'automatisation de l'analyse** : La tâche de l'analyse des fichiers journaux reste une activité essentiellement manuelle : c'est l'administrateur qui doit à chaque fois lancer le logiciel d'analyse et effectuer la gestion de ces fichiers. Donc on peut rendre les choses beaucoup plus aisées par la génération de ces fichiers journaux d'une façon continuelle. Certes, on ne peut pas éliminer l'intervention humaine définitivement, mais nous devrons penser à diminuer au maximum les tâches répétitives.

- **Le non suivi des fichiers journaux d'une source donnée** : La plupart des approches d'analyse des fichiers journaux présentées précédemment, permettent d'avoir des rapports pertinents à partir des journaux, mais elles ne permettent pas un suivi efficace des fichiers générés par une source donnée dans le sens où à chaque fois que nous voulons analyser les fichiers journaux on doit faire comme si c'était la première analyse.

II.2.3.Solution proposée et retenue

Ce projet permet le développement d'un outil graphique de gestion des fichiers Log (fichier d'historique ou journal) des firewalls d'un réseau local.

Notre système traite de façon simplifiée, générique et centralisée la collecte et l'analyse des logs et des événements sur une station de supervision à partir des différents types de firewall mis en place dans le réseau.

Cet outil assure les fonctionnalités configurables suivantes:

-Ecouter sur le réseau local les paquets IP émis par les firewalls et la création des fichiers Log correspondants.

-Sauvegarder les informations des fichiers Log des différents types de firewall dans une base de données indépendamment de la structure de ces fichiers.

-Consultation et analyse à travers des graphes et des statistiques des différents types et catégories d'intrusion et d'attaques externes à partir des informations stockées dans la base de données.

II.3.Spécification des besoins

L'étape de spécification est déterminante pour la suite de notre projet. En effet elle permet de préciser les fonctionnalités que le système doit fournir à ses utilisateurs et le lien entre ces différentes fonctionnalités.

Dans cette étude, nous allons tous d'abord spécifier les besoins fonctionnels et non fonctionnels de notre application, ensuite nous allons dresser le diagramme de cas d'utilisation correspondant.

II.3.1.Besoins fonctionnels

La pertinence du produit réalisée ne peut avoir de sens sans une analyse préalable des besoins. Le type des besoins identifiés est principalement une collecte centralisée et générique des événements journaliers signalés par les pare-feu et une gestion et analyse des activités du

réseau pour une reconfiguration et mise à jour des règles de sécurité. Ces besoins se présentent par module:

- **Gestion Event Listener**

  -Identifier le pare-feu à écouter : assurer l'identification de chaque pare-feu à écouter en mettant l'adresse IP et le numéro de port du firewall.

  -Créer les fichiers de journalisation : permet de créer un fichier logs qui englobe les évènements journaux.

  -Ecouter les événements Logs : assurer l'écoute des événements Logs générés sur chaque firewall du réseau local en temps réel.

- **Gestion Event Check**

  -Sauvegarder les informations des fichiers logs : Recenser l'ensemble des logs éparpillés, pour centraliser ces contenus dans une base de donnés bien définie.

- **Gestion Event Report**

  -Consulter le tableau de bord : L'interface graphique permettant l'analyse en temps réel des intrusions et des attaques, doit être claire et fiable. Elle présente au responsable de sécurité du système d'information de l'entreprise, des informations pertinentes et actualisée des activités filtrées par les pare-feu.

  -Générer les rapports : assurer la génération des rapports et statistique détaillés qui englobe les informations journaux selon des critères bien définies pour mesurer les performances et planifier les mises à jour ou prendre connaissance des intrusions ou des attaques externes.

  -Gestion des rotations des fichiers journaux : Le responsable de sécurité du système d'information peut planifier les rotations des fichiers logs ou les supprimer manuellement.

II.3.2.Besoins non fonctionnels

Notre système de gestion et d'analyse des fichiers journaux doit répondre aux contraintes de :

Portabilité : Aucune supposition sur la plate-forme sur laquelle doit tourner l'application.

Réutilisation : Le code doit être facile à réutiliser, à modifier et à étendre.

Maniabilité : C'est un critère important dans la mesure où la clarté et la visibilité des informations journalisées permettront de comprendre à tout moment l'activité du réseau.

II.4.Diagramme du cas d'utilisation

Cette partie définit les acteurs ainsi que les cas d'utilisation.

-Les acteurs :

L'étude des besoins a révélé la présence d'un acteur principal du système : Le responsable de sécurité du système d'information de tout organisme. Un acteur est par définition, en relation avec le système.

-Les cas d'utilisation :

Un cas d'utilisation spécifie une séquence d'actions selon le point de vue d'une catégorie d'utilisateurs.

Le mécanisme générique de regroupement d'éléments en UML s'appelle-le package.

Nous allons y recourir afin de structurer notre ensemble de cas d'utilisation

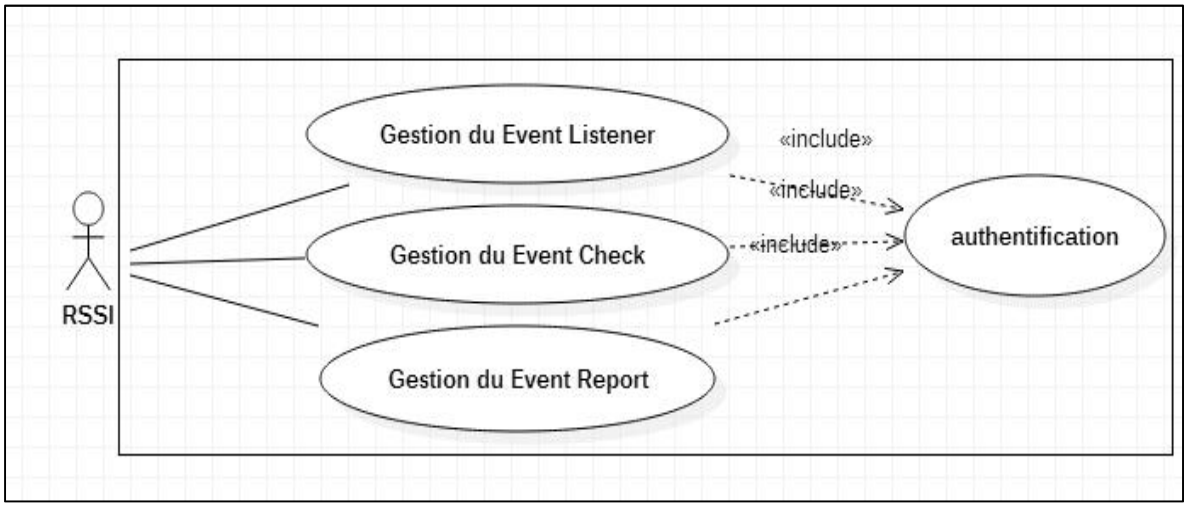


Figure II.1 Diagramme de cas d'utilisation général

Le déroulement de l'application est représenté selon la figure II.1 par un seul utilisateur qui se présente par le responsable de sécurité du système d'information ainsi que par les cas d'utilisation suivants:

- **Authentification:**

- Le responsable de sécurité du système d'information doit s'authentifié (saisir un login et un mot de passe) pour réaliser toutes les opérations sur le système.

- **Gestion du l'Event Listener**

L'Event Listener qui assurera l'écoute des événements Logs générés sur chaque firewall du réseau local en temps réel.

- **Gestion du l'Event Check**

L'Event Check permettra de parcourir l'ensemble des fichiers journaux, créés après avoir lancé l'Event Listener sur les pares feu du réseau dans le but de stocker dans une base de données les informations pertinentes des attaques et des événements signalés.

- **Gestion du l'Event Report**

L'Event Report permettra à travers une interface graphique de fournir des informations claires et fiables et des statistiques actualisées sur les données extraites de la base de données des fichiers journaux.

II.4.1.Raffinement du cas d'utilisation « Gestion du l'Event Listener »

Event Listener se charge d'écouter un port précis des pares feu, qui peut être différent selon leurs types. En général, le pare feu envoi sur ce port tous les événements journaux. Le fonctionnement de ce module consiste à écouter le port préalablement précisé, en utilisant les principes des sockets, ensuite d'avoir une copie de l'information envoyée par le pare feu et la mettre dans un fichier journal qui est créé au début de l'application.

La figure II.2 permet de nous décrire l'ensemble des traitements effectués à travers le package Event Listener

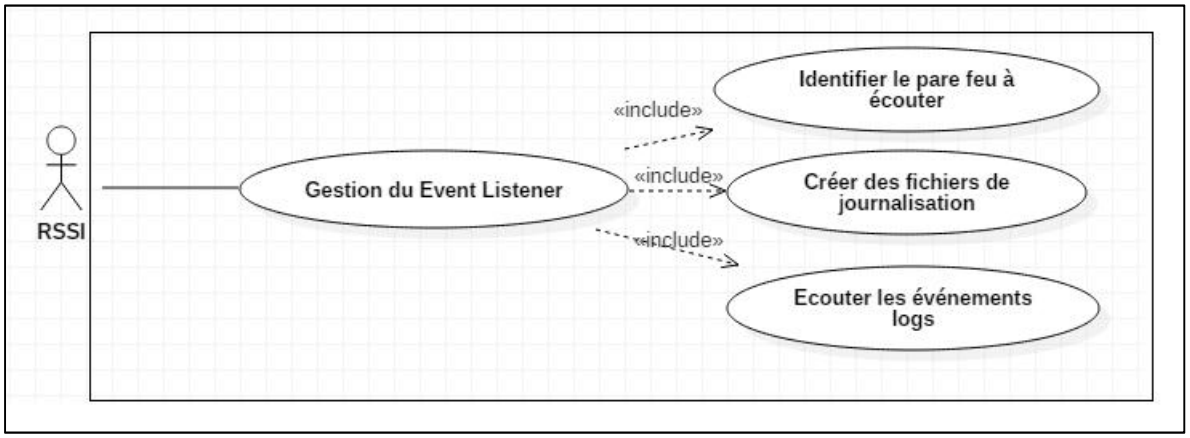


Figure II.2 Diagramme de cas d'utilisation « Gestion du l'Event Listener »

Le responsable de sécurité du système d'information permet d'identifier le pare feu à écouter à travers un adressage bien définie ensuite il crée un fichier pour sauvegarder les événements Logs afin de le traiter

II.4.2.Raffinement du cas d'utilisation « Gestion du l'Event Check »

Event Check consiste à extraire les données qui existent dans le fichier log et les organiser dans une base de données comme le présente la figure II.3.

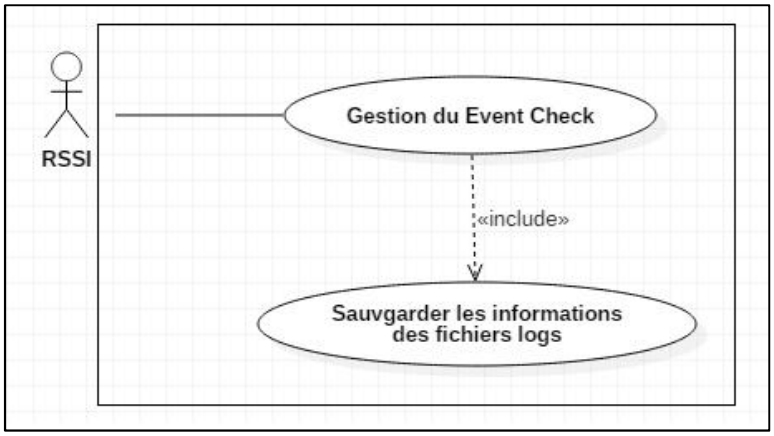


Figure II.3 Diagramme de cas d'utilisation « Gestion du l'Event Check »

Le responsable de sécurité du système d'information doit s'authentifier pour chaque intégration et sauvegarde des donnés à partir des fichiers logs

II.4.3.Raffinement du cas d'utilisation « Gestion du l'Event Report»

Event Report fournit une interface graphique affichant des informations et des statistiques sur les données extraites de la base de données. Ces informations peuvent être de plusieurs types :

- Des adresses IP (sources ou destinataires).
- Date et heure d'attaques.
- Protocoles utilisés.
- Type de l'événement journalisé.

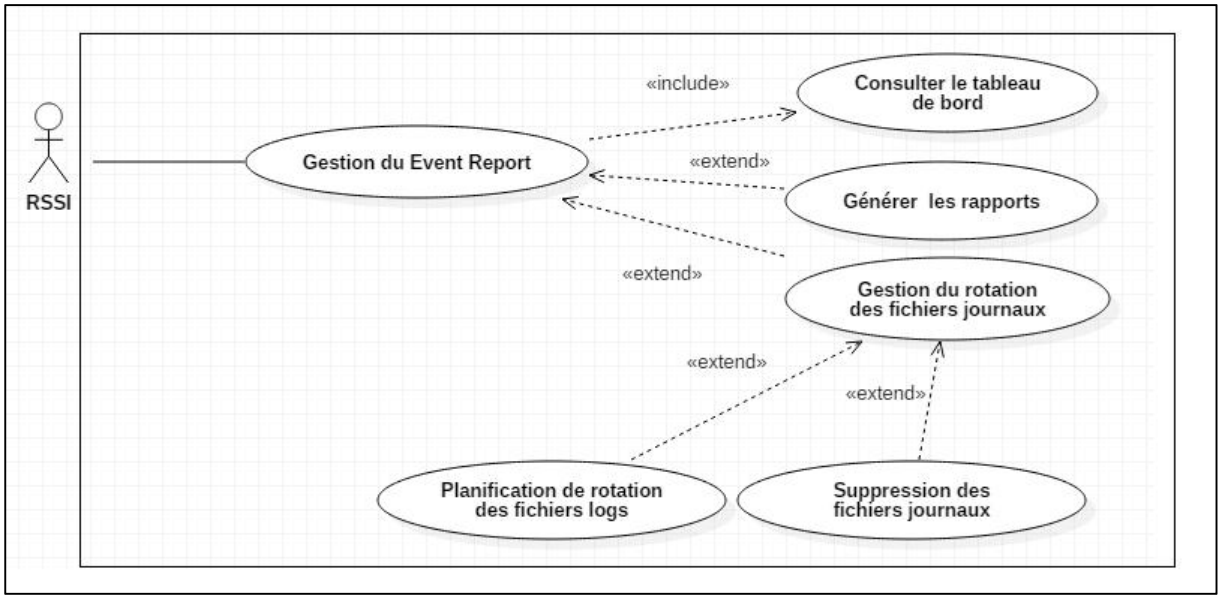


Figure II.4 Diagramme de cas d'utilisation « Gestion du l'Event Report »

Le responsable de sécurité de système d'information selon la figure II.4 consulte le tableau de bord ainsi qu'il peut générer des rapports HTML personnalisés. Il gère aussi la rotation des fichiers journaux. Cette rotation peut être configurée selon la taille des fichiers journaux ou selon une fréquence de rotation (quotidienne, hebdomadaire) ainsi le responsable peut consulter les caractéristiques des fichiers journaux (Nom, Taille, date de dernière modification) et choisir les fichiers à supprimer.

II.5.Conclusion

Au cours de ce chapitre, nous avons analysé les besoins fonctionnels de notre système, nous avons cerné les différentes relations et interactions. Dans le chapitre suivant nous allons entamer la phase de conception

Chapitre III

Conception

III.1.Introduction

Ce chapitre sera consacré à une description détaillée des échanges effectués entre l'utilisateur et les différents composants de notre produit. Cette description sera explicite notamment à partir des diagrammes des séquences et des diagrammes des classes.

III.2. Architecture Logicielle

-L'architecture logicielle du système regroupe les composants physiques et logiques et les relations entre ces composants.

III.2.1.Architecture Physique

L'architecture utilisée est à 2 tiers comme le montre la figure III.1 et elle permet de spécifier les composants physiques nécessaires pour l'application.

- Tiers 1 : Les firewalls à écouter
- Tiers 2 : Une machine d'application et base de données de notre application.

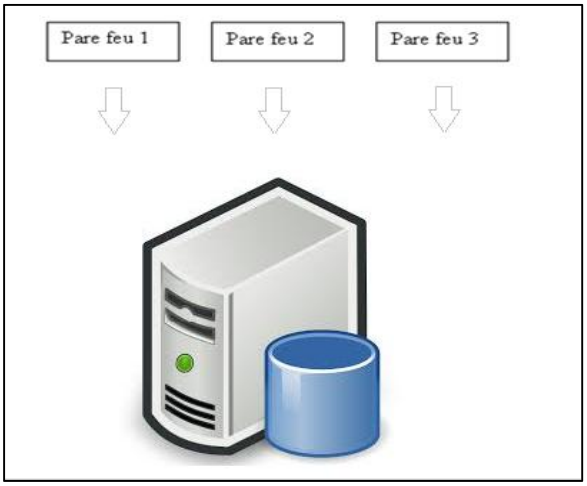


Figure III.1 Architecture physique

III.2.2.Architecture Logique

L'architecture logique permet de structurer et décomposer de façon logique chaque application en couches.

L'architecture utilisée est une architecture MVC (modèle, vue et contrôleur) comme l'illustre la figure III.2.

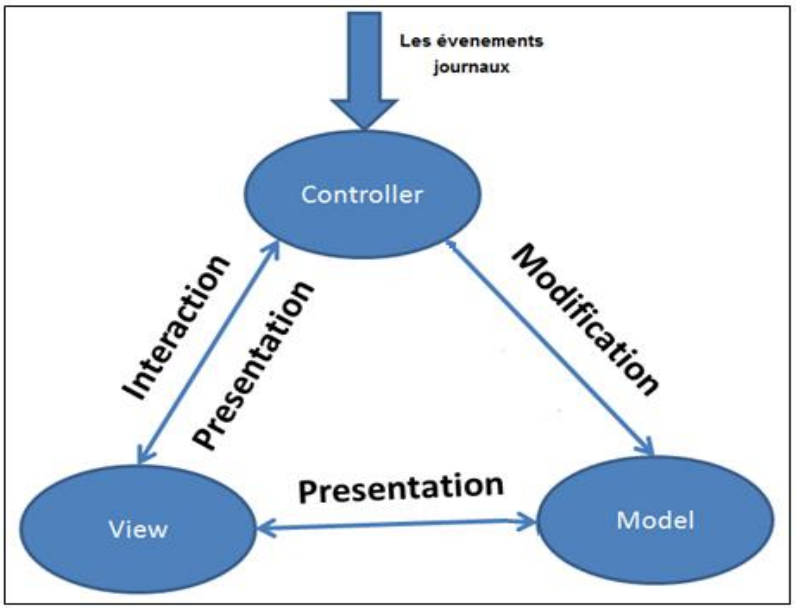


Figure III.2 Architecture logique

L'architecture MVC est basée sur trois couches :

-Le modèle

Il décrit les traitements des données et les interactions avec la base de données. Il contient les données manipulées par l'application. Il assure la gestion de ces données et garantit leur intégrité.

-La vue

Elle représente l'interface avec laquelle l'utilisateur interagit. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toutes les actions de l'utilisateur. La vue n'effectue aucun traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur.

-Le contrôleur

Il prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande

la modification des données au modèle, ce dernier avertit la vue que les données ont changée pour qu'elle se mette à jour. Certains événements de l'utilisateur ne concernent pas les données mais la vue. Dans ce cas, le contrôleur demande à la vue de se modifier. Le contrôleur n'effectue aucun traitement, ne modifie aucune donnée. Il analyse la requête du client et se contente d'appeler le modèle adéquat et de renvoyer la vue correspondant à la demande. Par exemple, dans le cas d'une base de données gérant les emplois du temps des professeurs d'une école, une action de l'utilisateur peut être l'entrée (saisie) d'un nouveau cours. Le contrôleur ajoute ce cours au modèle et demande sa prise en compte par la vue. Une action de l'utilisateur peut aussi être de sélectionner une nouvelle personne pour visualiser tous ses cours. Ceci ne modifie pas la base des cours mais nécessite simplement que la vue s'adapte et offre à l'utilisateur une vision des cours de cette personne. Quand un même objet contrôleur reçoit les événements de tous les composants, il lui faut déterminer quelle est l'origine de chaque événement. Ce tri des événements peut s'avérer fastidieux et peut conduire à un code pas très élégant C'est pourquoi le contrôleur est souvent scindé en plusieurs parties dont chacune reçoit les événements d'une partie des composants.

III.3.Conception de l’aspect statique

Dans cette partie, nous allons présenter Le diagramme de package de l’application et les diagrammes des classes des trois packages Event Listener, Event Check et Event Report.

III.3.1. Diagramme de package

Notre produit étant composé de 3 packages, l’Event Listener, l’Event Check et l’Event Report.

Nous avons vu qu’il était utile de regrouper les classes fortement couplées en unités plus grandes. Le couplage s’exprime à la fois structurellement par des associations, des agrégations ou des généralisations, mais aussi dynamiquement par les interactions qui se produisent entre les instances des classes. Nous utiliserons le concept de package pour regrouper les classes à forte cohérence interne.

La répartition des classes candidates illustrée dans la figure III.3 offre une vue globale des packages et de leurs propres classes.

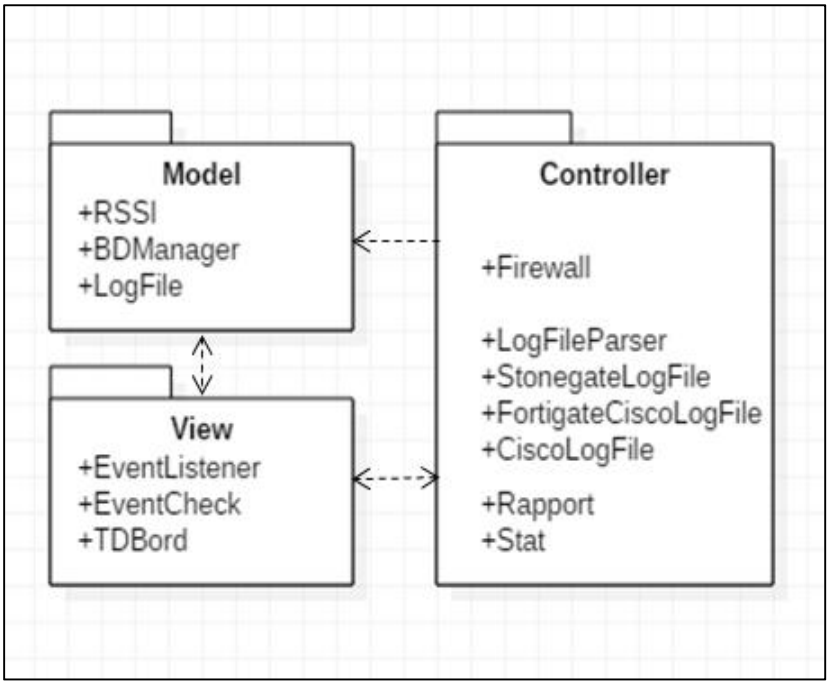


Figure III.3 Découpage en package de système de gestion et d'analyse de fichier journaux.

III.3.2. Diagramme de classe

Dans la figure III.4, nous allons déterminer les diagrammes de classe par package.

-Package Controller

Le diagramme de classe indiqué dans la figure III.4 est du package Controller :

- Une classe Firewall représente le pare feu déployé sur le réseau local.
- La classe abstraite LogFileParser contiendra les différentes méthodes pour parcourir les fichiers journaux et extraire les informations pertinentes.
- La classe abstraite FortigateCiscoLogFile contient des méthodes abstraites pour extraire les données communes des fichiers de type Cisco Pix et FortiGate.
- La classe StonegateLogFile contient des méthodes pour extraire les données des fichiers de type StoneGate.
- La classe FortigateLogFile contient des méthodes pour extraire les données des fichiers de type FortiGate.
- La classe CiscoLogFile contient des méthodes pour extraire les données des fichiers de type Cisco Pix.
- La classe Rapport identifie chaque rapport généré par son nom et son emplacement.

- La classe GestionLog assure la suppression des fichiers journaux sélectionnés par le RSSI, ainsi que le lancement de tache planifié concernant la rotation des fichiers journaux.
- La classe Stat assure les statiques selon des critères bien définies.

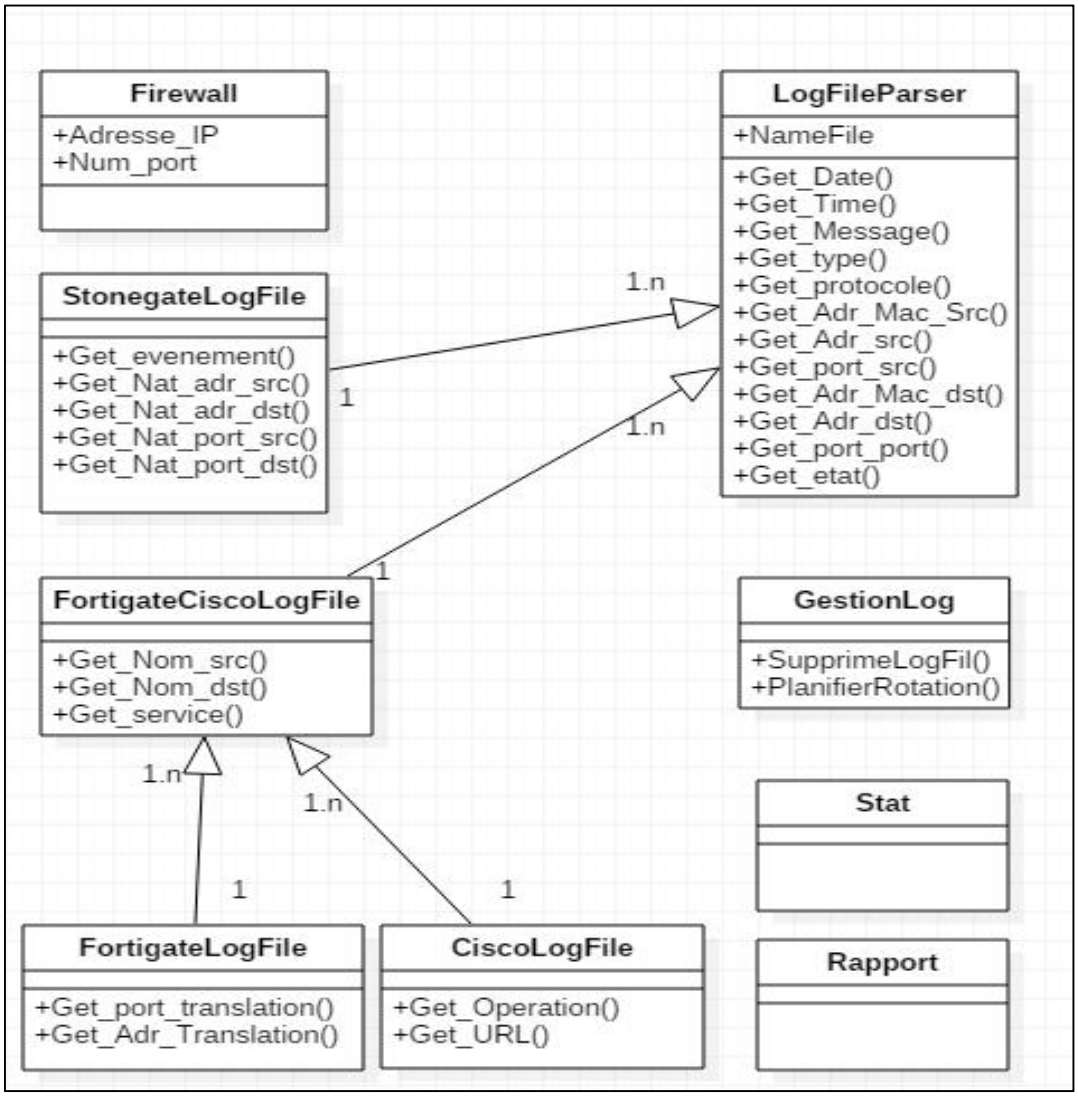


Figure III.4 Diagramme de classe du package Controller

-Package View

Le diagramme de classe du package View illustré dans la figure III.5 comporte :

- La classe Menu_Principale présente l'interface principale de l'application.
- La classe Event check permet de déterminer l'ensemble des fichiers journaux créés.
- La classe EventListener présente l'interface ou le responsable de sécurités peut mettre les informations nécessaires pour écouter un pare-feu.

- La classe TDBord permettra de représenter les informations des fichiers journaux et d'afficher les statistiques sur l'état du trafic réseau

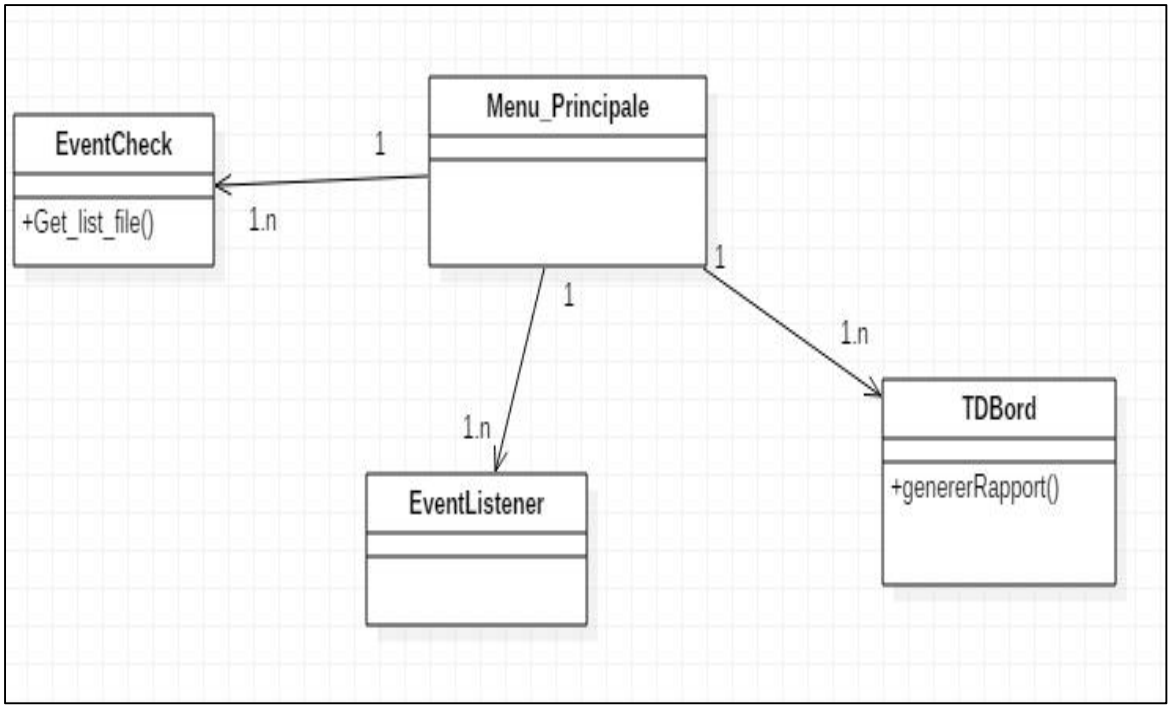


Figure III.5 Diagramme de classe de module View

-Package Model

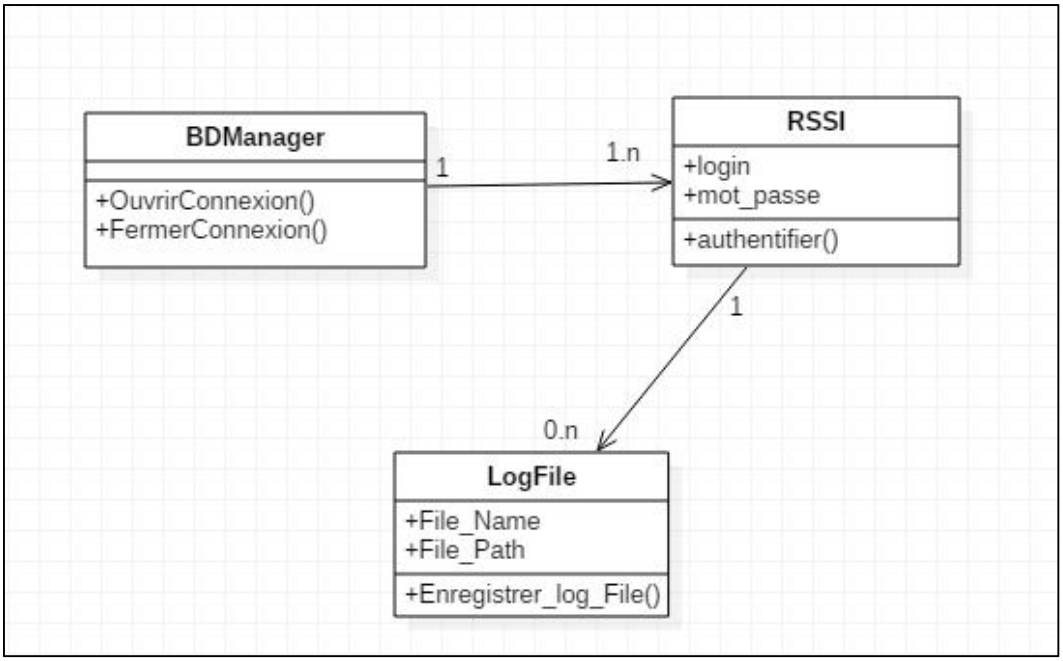


Figure III.6 Diagramme de classe de package Model

Le diagramme de classe présenté dans la figure III.6 du package Model comporte :

- La classe RSSI qui identifie le responsable de sécurité du système d'information par son login et son mot de passe.
- La classe BDManager assurera l'ouverture et la fermeture des connexions vers une base de données spécifiée par son nom et l'extraction des informations voulue à partir de la base de données.
- Une classe LogFile qui représente les fichiers journaux créent.

III.4. Conception de l'aspect dynamique

Dans cette partie, nous allons présenter les diagrammes des séquences des trois packages Event Listener, Event Check et Event Report ainsi que le diagramme de classe de notre application

III.4.1. Diagrammes des séquences

Dans cette partie nous allons commencer par détailler les diagrammes des séquences.

- **Event Listener**

Pour l'identification des pares feu à écouter, la figure III.7 représente les méthodes implémentées au sein de l'application afin d'identifier un pare feu du réseau.

Nous pouvons expliciter le diagramme de séquence comme suit :

- L'utilisateur doit d'abord s'authentifier avant d'utiliser le système.
- Il fournit ensuite l'adresse IP et le numéro de port du pare feu.
- Il indique la catégorie et le type du pare feu à écouter.
- Une indication sur l'état du pare feu (disponible ou non) sera envoyée à l'utilisateur.

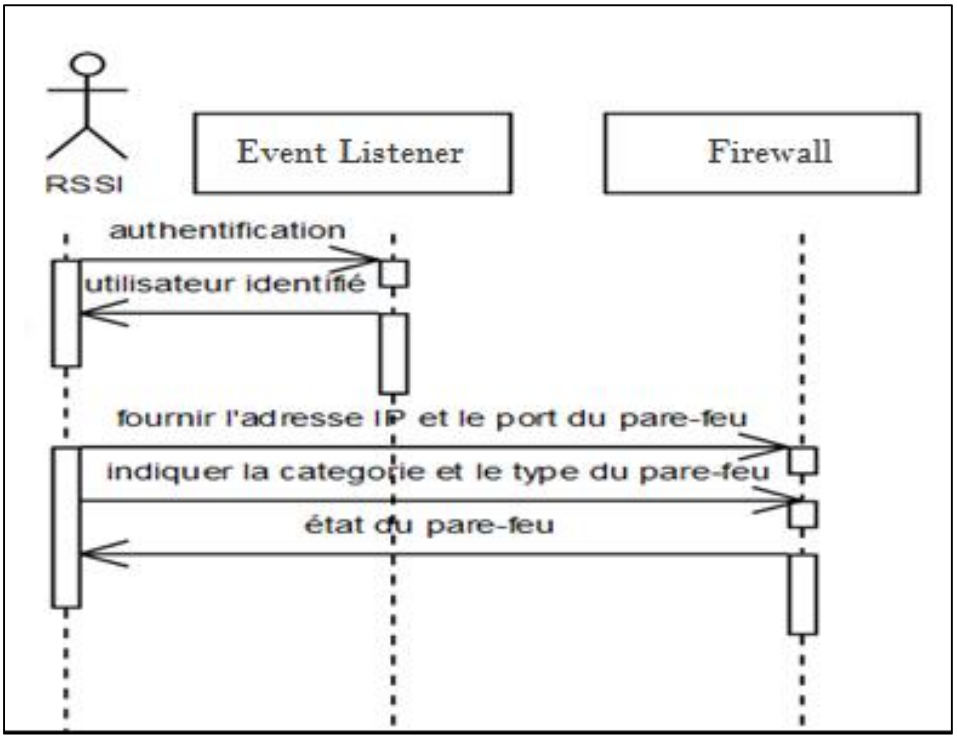


Figure III.7 Identification des pares feu à écouter

- **Event Check**

Pour la sauvegarde des informations des fichiers journaux des pares feu, le responsable de sécurité du système d'information indiquera à travers une interface graphique les emplacements des fichiers journaux créés pour chaque types des pares feu comme illustré dans la figure III.8.

La sauvegarde des informations journalisées dans une base de données s'effectuera après l'établissement de connexion avec une base de données spécifiée par son nom. Une fois l'enregistrement effectué, la connexion vers la base sera fermée.

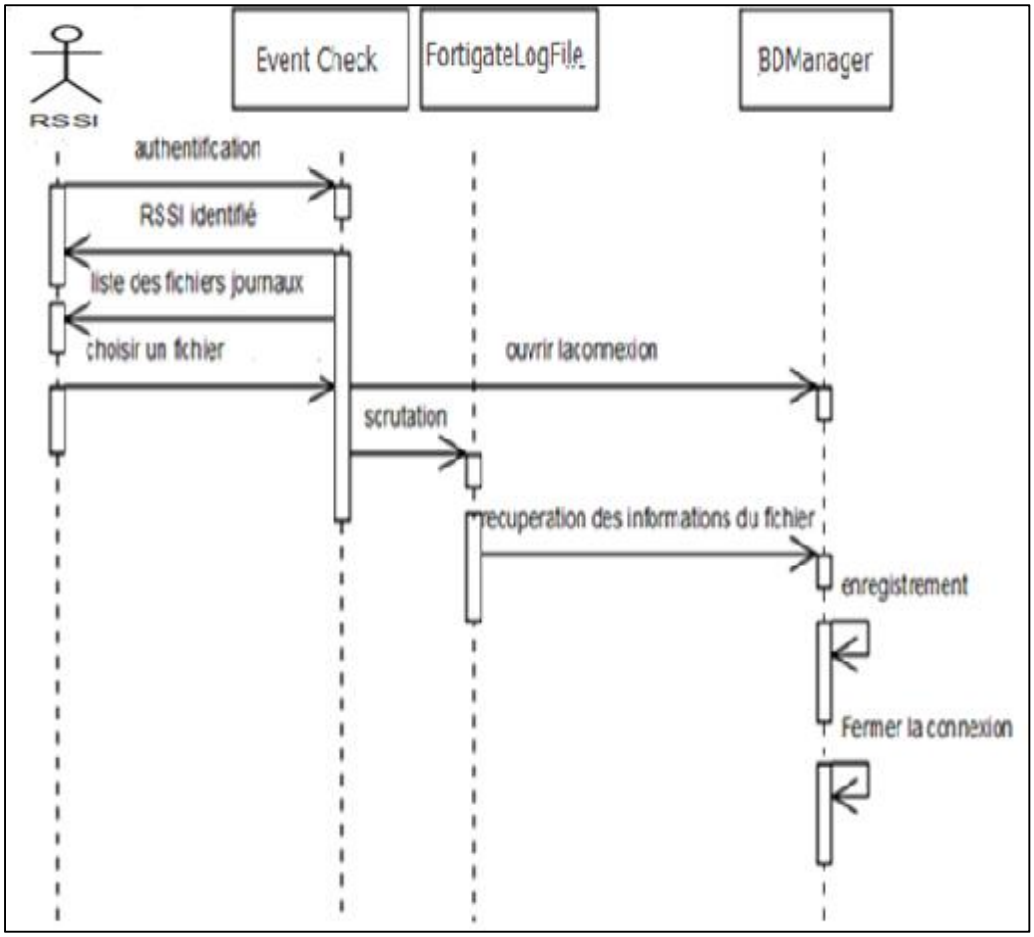


Figure III.8 Sauvegarde des informations des fichiers journaux des pares feu

- **Module Event Report**

Pour la consultation du tableau de bord et la génération des rapports nous allons expliciter le diagramme de séquence dans la figure III.9 :

- L'utilisateur doit d'abord s'authentifier avant d'utiliser le système.
- Il indique au système les informations et l'intervalle de temps sur lesquels portent les statistiques à afficher.
- Un objet de connexion à la base de données est créé pour extraire les données.
- Affichage des statistiques et des informations des entrées journaux.
- Génération des rapports.
- Fermeture de la connexion à la base de données.

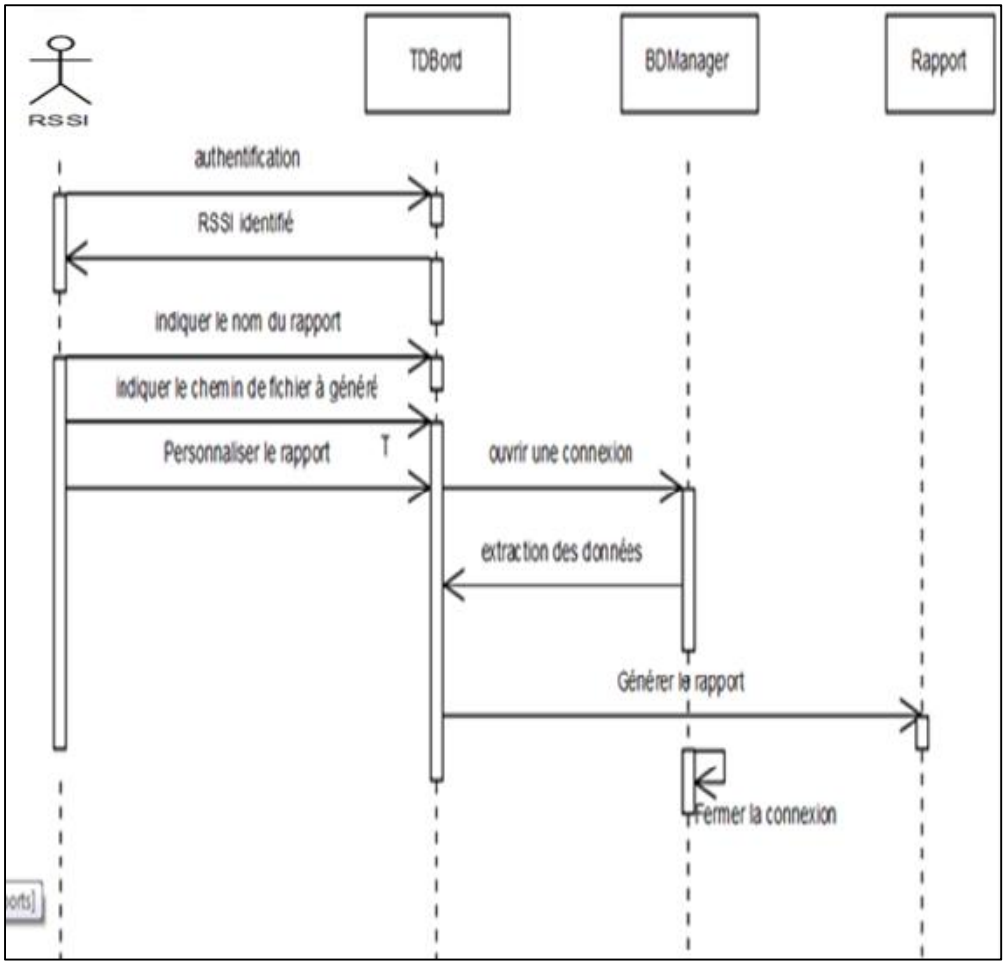


Figure III.9 Consultation et génération de rapport du reporting

Pour la suppression des fichiers journaux nous allons expliciter le diagramme de séquence dans la figure III.11 :

- L'utilisateur doit d'abord s'authentifier avant d'utiliser le système.
- Il va demander la liste des fichiers journaux et il va sélectionner le fichier journal à supprimer.
- Suppression du fichier journal.

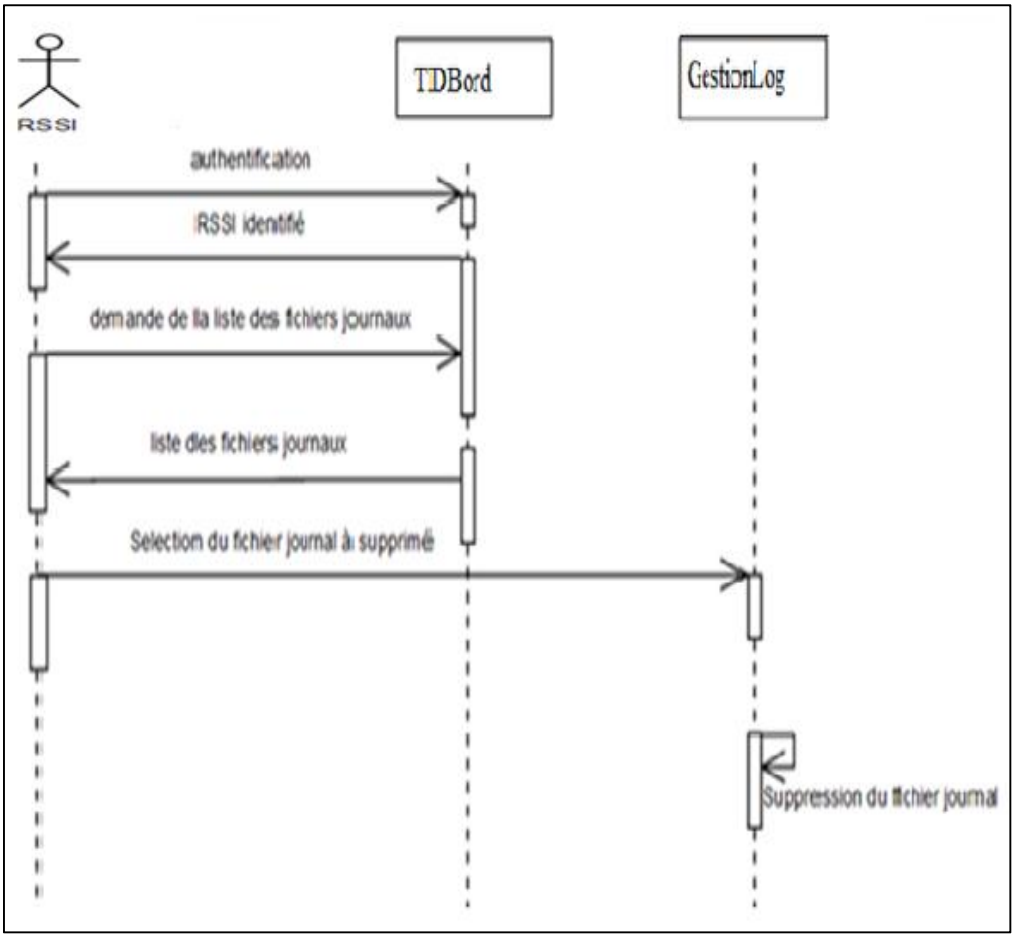


Figure III.11 Suppression des fichiers journaux

III.5.Conclusion

Lors de la conception, nous avons présenté les composants logiciels collaborant pour la réalisation des fonctionnalités de notre système tout en nous concentrant sur les cas d'utilisation les plus significatifs et représentatifs.

Lors du chapitre suivant, la réalisation, nous présentons l'environnement matériel et logiciel dans lesquels notre projet peut être exécuté.

Chapitre IV

Réalisation

IV.1.Introduction

Une manipulation souple et aisée reste toujours parmi les critères les plus décisifs pour le succès de tout projet. C'est pourquoi le choix des outils de programmation doit être bien étudié.

Après avoir réalisé la conception et le développement de notre application d'analyse et de reporting des fichiers journaux des firewalls, nous allons présenter au cours de ce chapitre les résultats de l'implémentation de notre application.

Nous commençons tous d'abord par une description de l'environnement de travail puis une description de la mise en place de l'application

IV.2.Environnement de travail

Dans cette partie, nous présentons les différents outils requis pour le développement et la mise en œuvre de notre application.

IV.2.1.Environnements Matériels

Nous avons utilisé principalement un seul ordinateur portable dont leur configuration est la suivante :

- Un ordinateur tournant sous Microsoft Windows 7 Professionnel 64 bit.
  Processeur Intel core i7-6500U
  8 GO de RAM.
  1TERA de disque dur.

De plus nous avons accès aux équipements suivants pour l'obtention des fichiers journaux pour les tests de l'application:

- ✓ Le pare feu FORTIGATE.
- ✓ Le pare feu STONEGATE.
- ✓ Le pare feu CISCO PIX.

IV.2.2.Environnement Logiciel

Dans cette section, nous présentons les différents outils de développement ainsi que le SGBD (Système de Gestion de Base de Données) et l'outil de conception, qui sont liés à notre projet.

Nous avons utilisé l'environnement de développement **Eclipse Standard** pour le développement des différents modules de notre application.

- **LE SGBD MYSQL**

Le serveur de base de données MySQL est très rapide, fiable et facile à utiliser. Il dispose aussi de fonctionnalités pratiques, développées en coopération avec ces utilisateurs puisqu'il est Open Source. Le serveur MySQL a été développé à l'origine pour des grandes bases de données plus, et a été utilisé avec succès dans des environnements de production très contraints et très exigeant, depuis plusieurs années.
Bien qu'en développement, le serveur MySQL offre des nombreuses fonctions puissantes. Ses possibilités de connexion, sa rapidité et sa sécurité font du serveur MySQL un serveur hautement adapté au développement des applications.

- **StarUML**

StarUML est un logiciel de modélisation UML (Unified Modeling Language), cédé comme open source par son éditeur, à la fin de son exploitation commerciale, sous une licence modifiée de GNU GPL. Etant simple d'utilisation, nécessitant peu de ressources système, supportant UML, ce logiciel constitue une excellente option pour une familiarisation à la modélisation.

- **L'API JfreeChart :**

JFREECHART est une bibliothèque Open Source qui permet de créer des données statistiques sous la forme de graphiques. Elle possède plusieurs formats dont le camembert, les barres ou les lignes, et propose de nombreuses options de configuration pour personnaliser les graphiques. Elle peut être utilisée dans des logiciels ou des applications Web et permet également d'exporter le graphique sous la forme d'une image.

- **L'API MySQL Connector/J:**

C'est une bibliothèque Open Source contenant un ensemble de classes pour la connexion à des bases de données MySQL. Elle fournit un pilote JDBC pour le développement d'application JAVA interagissant avec des serveurs MySQL.

IV.3.Implémentation du Package Event Listener

Le module regroupe les différentes interfaces graphiques de notre application, dont les principales sont détaillées dans la figure IV.1.

La figure IV.1 montre l'aspect de l'interface au moment du début de l'exécution. Dans cette étape, le responsable de sécurité du système d'information peut choisir l'un des packages à exécuter (Event Listener, Event Check et Event Report).

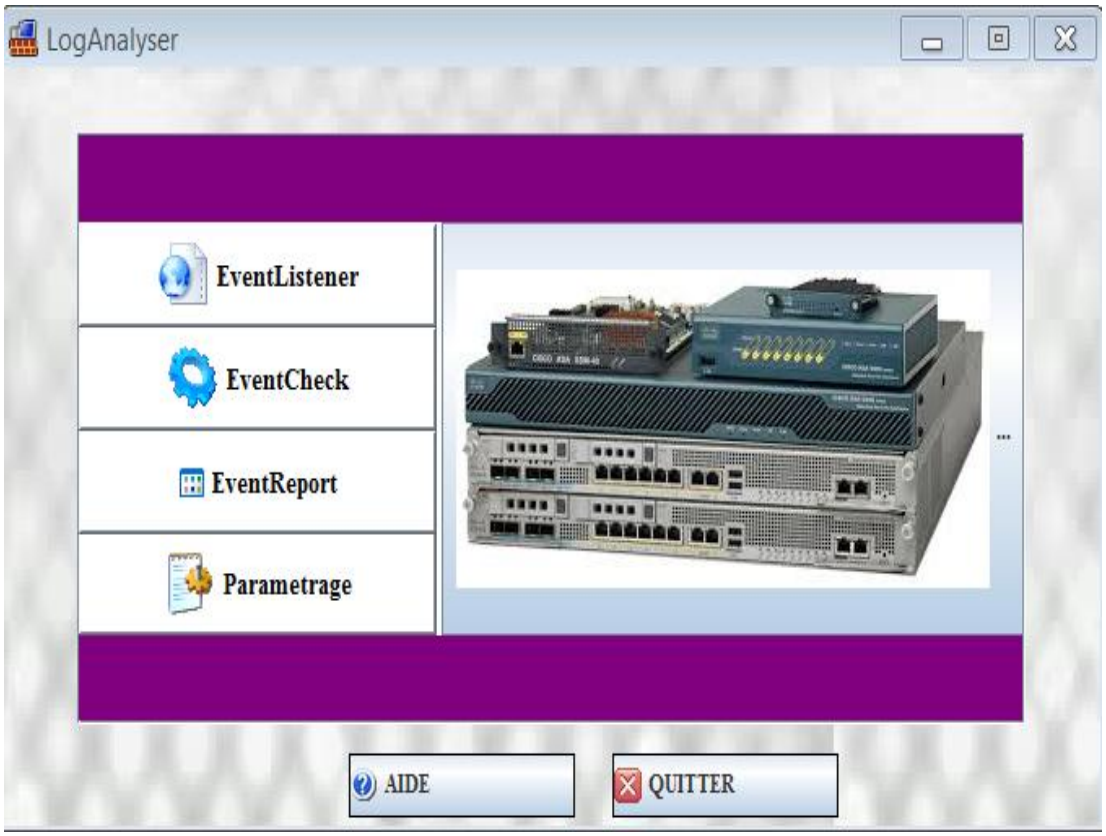


Figure IV.1 Menu principal de l'application

La figure IV.2 montre une étape de l'exécution du package Event Listener : L'utilisateur mentionne l'adresse IP et le numéro de port du pare feu à écouter, choisit le nom et le chemin du fichier journal à créer et sélectionne le type et la catégorie du pare feu en question.

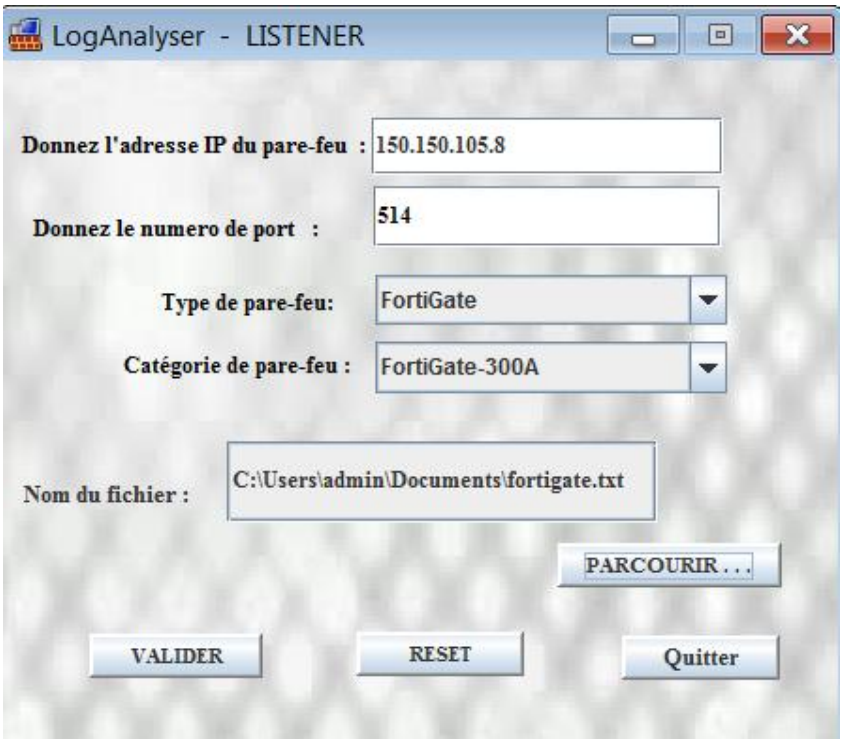


Figure IV.2 Interfaces du package EventListener

⇨ Lorsqu'on tape sur le bouton « valider », tout d'abord, nous allons vérifier l'état du firewall : S'il est accessible, le listener va être lancé, et il commence à récupérer les paquets.

```
package loganalyser;
public class Firewall extends Thread {
        String Adresse_IP;
        LogFile F;
        int Port;
        public Firewall(String _Adr, int _port, LogFile _F) {
                Adresse_IP = _Adr;
                Port = _port;
                F = _F;
        }
        public boolean Etat_Firewall(){}

        public void Get_Log_message() {}
```

Figure IV.3Implémentations de la classe Firewall

⇨ La classe Firewall illustré dans la figure IV.3 représente un pare feu déployé sur le réseau local, la méthode « public Etat_Firewall () »est une méthode implémentée pour avoir l'état de firewall et la méthode « public Void Get_Log_message ()» est une méthode implémentée pour la récupération des logs.

```
public void Get_Log_message()
 {
   String s;
   try {
   feed=new FeedBack(Adresse_IP,this);
   Socket cl = new Socket(Adresse_IP,Port);
   BufferedReader lire = new BufferedReader(new InputStreamReader(cl.getInputStream()));
```

Figure IV.4 Extrait de la méthode « Get_Log_message () »

⇨ La méthode Get_Log_message () illustré dans la figure IV.4 est responsable de la récupération des logs.

La figure IV.5 montre la visualisation en temps réel des événements journaux écoutés du pare feu indiqué.

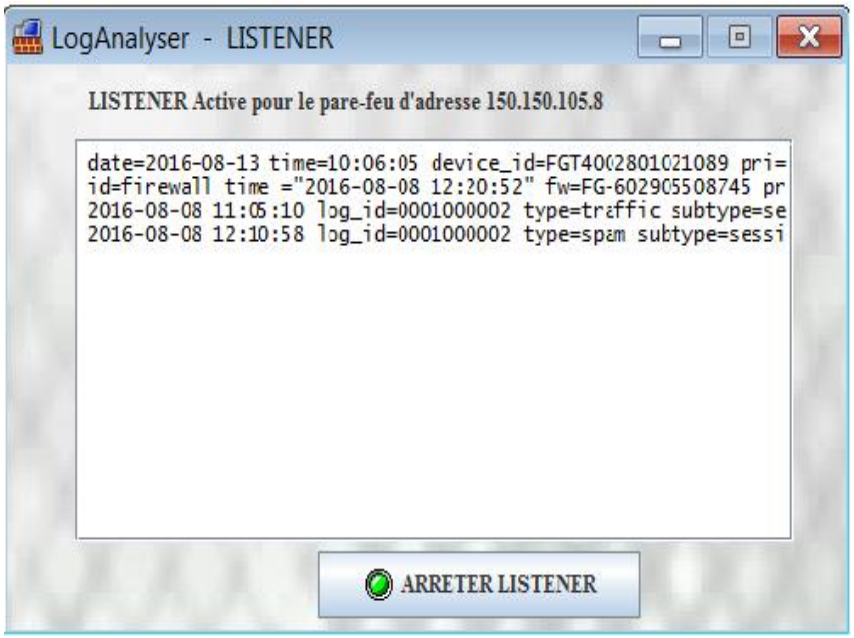


Figure IV.5 Interface de visualisation des événements journaux.

IV.4.Implémentation du package Event Check

Une fois le fichier créé et rempli, le responsable de sécurité du système d'information peut sauvegarder les informations journaux du fichier créé en lançant la scrutation du fichier en question à travers le package Event Check comme indiqué dans la figure IV. 6. Il indiquera pour cela son login, son mot de passe, le nom de la base de données pour s'y connecter et le chemin du fichier journal.

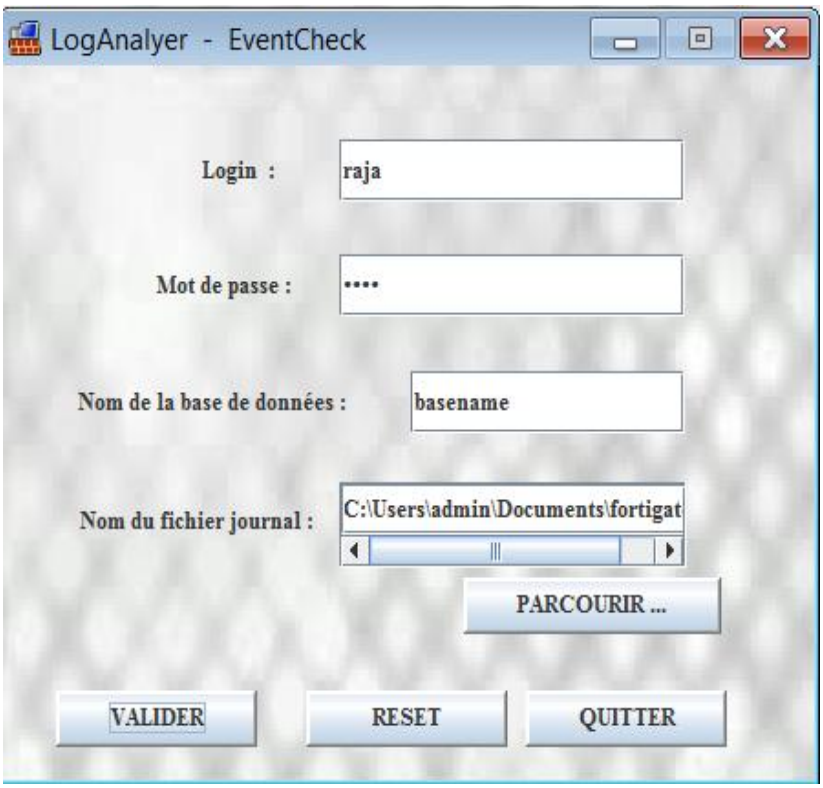


Figure IV. 6 Interfaces du package Event check

Lorsque nous cliquons sur le bouton « valider », nous allons vérifier et distinguer le type du pare feu selon l'emplacement du fichier log (déjà mentionné dans le package Event Listener : Fortigate, Cisco Pix ou bien Stonegate) que nous allons par suite insérer ces données dans la base de données associée.

```
package loganalyser;

public class EventCheck {
  String Nom_fichier="";
  BufferedReader fichier = null;
  BDManager bd;
  public EventCheck(BDManager _base)
  {
bd=_base;
 try {fichier = new BufferedReader(new FileReader(Nom_fichier));
 }catch (FileNotFoundException ex) {}
}
public void GetTypeFile(String FileName){}
}
```

Figure IV.7 Implémentation de la classe Event Check

⇨ La figure IV.7 présente la classe Event Check qui permet de déterminer l'ensemble des fichiers journaux créés.

⇨ La méthode « public Void GetTypeFile () »est implémentée pour savoir le type des fichiers journaux (s'ils sont Fortigate, CiscoPix ou Stonegate).

```
int categorie=0;
  String type="";
  try {
       BufferedReader logf = new BufferedReader(new FileReader(FileName));
       String aux="";
       while((aux=logf.readLine()).equals("")){}
       if(!aux.equals(""))
       {
         try
{

         ResultSet resultSet = null;
         bd.statement
                 .execute("SELECT * FROM categorie_firewall WHERE Categorie='"
                                                              + aux.split("=")[1] + "'");
```

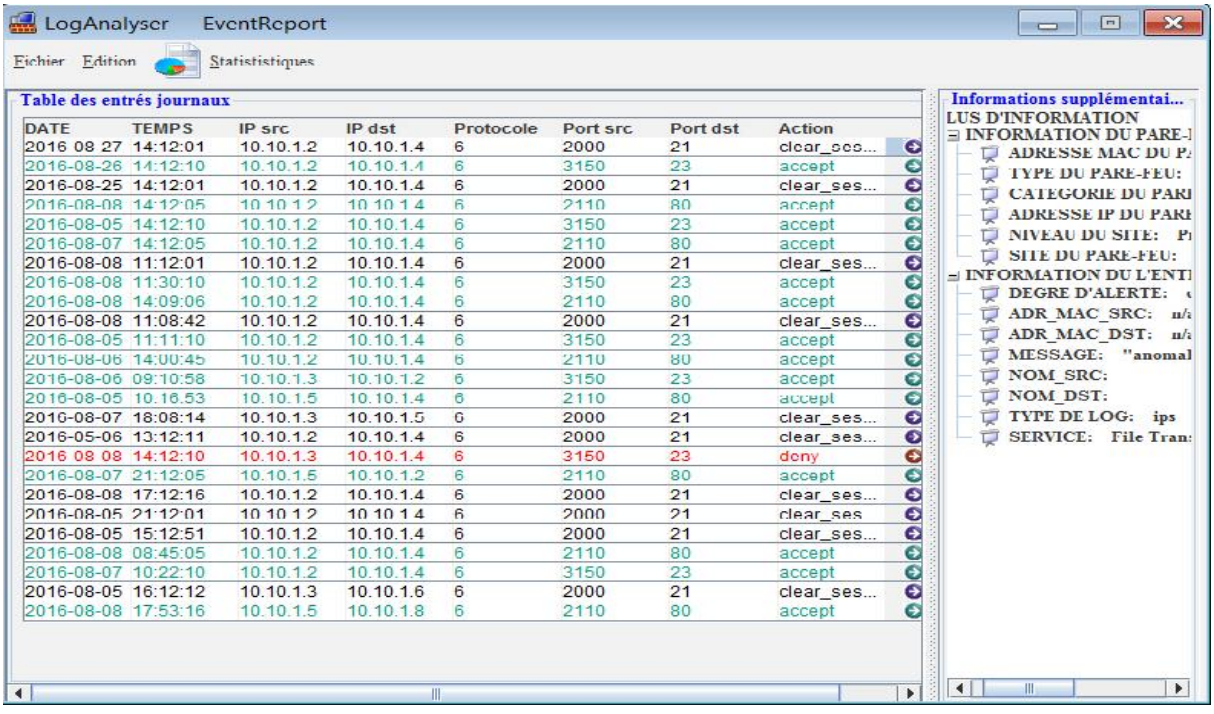
Figure IV.8 Extrait de la méthode « GetTypeFile () »

⇨ La figure IV.8 montre la méthode GetTypeFile () qui permet d'identifier le type de fichier log des firewalls (Fortigate, CiscoPix Pix, Stonegate), pour commencer la scrutation de ces fichiers, et insérer les paramètres dans la base de données.

IV.5.Implémentation du package Event Report

La figure IV.9 présente l'interface principale du package Event Report.

Pour visualiser des informations supplémentaires d'une entrée journal (le pare feu déclencheur, son site, son niveau, etc.). L'utilisateur peut sélectionner dans l'interface principale l'entrée voulue. Les informations complémentaires seront affichées en arborescence.



DATE	TEMPS	IP src	IP dst	Protocole	Port src	Port dst	Action
2016 08 27	14:12:01	10.10.1.2	10.10.1.4	6	2000	21	clear_ses...
2016-08-26	14:12:10	10.10.1.2	10.10.1.4	6	3150	23	accept
2016-08-25	14:12:01	10.10.1.2	10.10.1.4	6	2000	21	clear_ses...
2016-08-08	14:12:05	10.10.1.2	10.10.1.4	6	2110	80	accept
2016-08-05	14:12:10	10.10.1.2	10.10.1.4	6	3150	23	accept
2016-08-07	14:12:05	10.10.1.2	10.10.1.4	6	2110	80	accept
2016-08-08	11:12:01	10.10.1.2	10.10.1.4	6	2000	21	clear_ses...
2016-08-08	11:30:10	10.10.1.2	10.10.1.4	6	3150	23	accept
2016-08-08	14:09:06	10.10.1.2	10.10.1.4	6	2110	80	accept
2016-08-08	11:08:42	10.10.1.2	10.10.1.4	6	2000	21	clear_ses...
2016-08-05	11:11:10	10.10.1.2	10.10.1.4	6	3150	23	accept
2016-08-06	14:00:45	10.10.1.2	10.10.1.4	6	2110	80	accept
2016-08-06	09:10:58	10.10.1.3	10.10.1.2	6	3150	23	accept
2016-08-05	10.16.53	10.10.1.5	10.10.1.4	6	2110	80	accept
2016-08-07	18:08:14	10.10.1.3	10.10.1.5	6	2000	21	clear_ses...
2016-05-06	13:12:11	10.10.1.2	10.10.1.4	6	2000	21	clear_ses...
2016 08 08	14:12:10	10.10.1.3	10.10.1.4	6	3150	23	deny
2016-08-07	21:12:05	10.10.1.5	10.10.1.2	6	2110	80	accept
2016-08-08	17:12:16	10.10.1.2	10.10.1.4	6	2000	21	clear_ses...
2016-08-05	21:12:01	10.10.1.2	10.10.1.4	6	2000	21	clear_ses
2016-08-05	15:12:51	10.10.1.2	10.10.1.4	6	2000	21	clear_ses...
2016-08-08	08:45:05	10.10.1.2	10.10.1.4	6	2110	80	accept
2016-08-07	10:22:10	10.10.1.2	10.10.1.4	6	3150	23	accept
2016-08-05	16:12:12	10.10.1.3	10.10.1.6	6	2000	21	clear_ses...
2016-08-08	17:53:16	10.10.1.5	10.10.1.8	6	2110	80	accept

Figure IV.9 Interface principale du package EventReport.

La figure IV.10 présente le menu de l'Event Report contient des rubriques accessibles pour l'utilisateur. Ces rubriques ne sont accessibles qu'après authentification. Le menu est présenté dans la figure suivante.

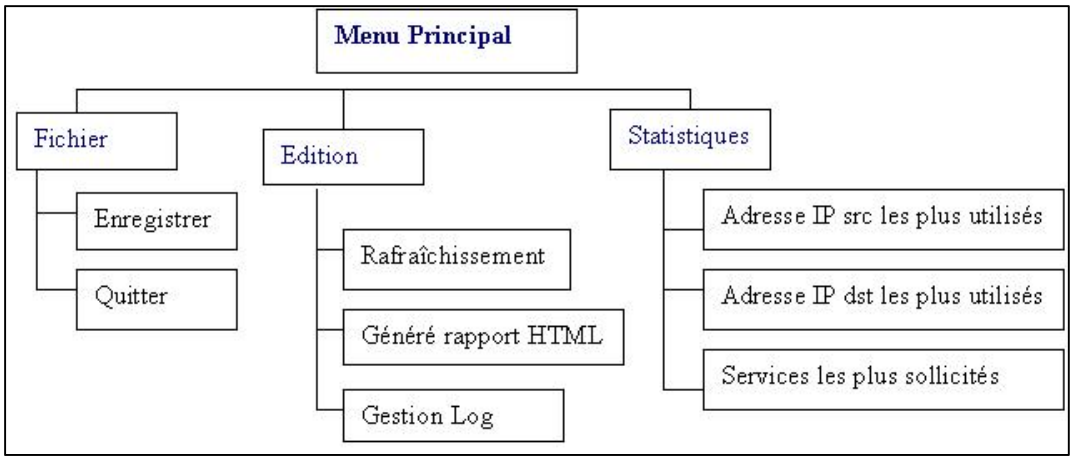


Figure IV.10 Menu principal du package Event Report

Le responsable de sécurité du système d'information peut visualiser les graphiques des statistiques reflétant l'activité de filtrage des pares feu du réseau, comme l'indique les figure IV.11 et IV.12, qui représentent des graphiques des services et adresses IP les plus sollicités de l'extérieur du réseau. Dans la première statistique par exemple, les ports 80 et 21 sont signalés, ce qui signifie que le réseau possède des serveurs Web. Dans la deuxième statistique on remarque que l'adresse source 10.10.1.2 est la plus sollicité depuis les 7 derniers jours

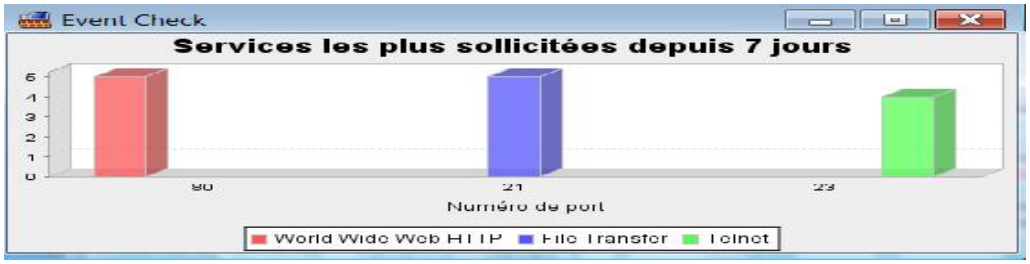


Figure IV.11 Graphique de statistique des services les plus sollicités.

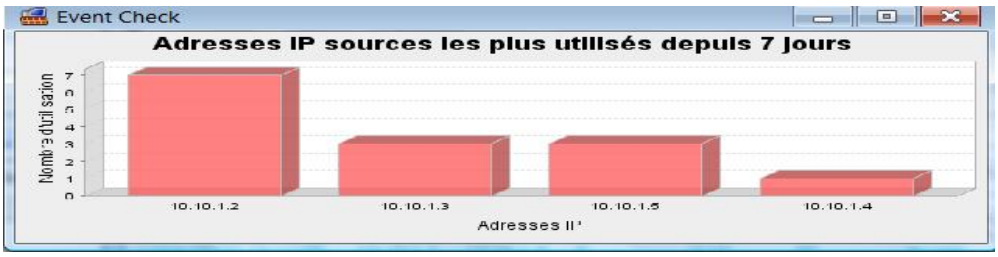


Figure IV.12 Graphique de statistique des adresses IP source les plus sollicitées

Comme illustré dans la figure IV.13, la méthode « statistique_adr_src » permet d'afficher les statistiques selon les adresses IP source les plus utilisés.

```
public void Staististique_Adr_src()
{
 ResultSet resultSet=null;
 largeur=0;
 barChart=null;

String List_num_entrés=this.Creation_des_stat();
 try{

  boolean  rs = bd.statement.execute("SELECT Num_entre,Adr_src,count(Adr_src) FROM `log_table`
                                     WHERE Num_entre IN '+List_num_entrés+" GROUP BY 'Adr_src' ORDER BY 3 DESC");
    resultSet =bd.statement.getResultSet();
```

Figure IV.13 Extrait de la méthode « statistique_Adr_src() »

Pour gérer les fichiers journaux, le responsable de sécurité du système d'information ouvrira la fenêtre « GESTION DES JOURNAUX ». À partir de cette fenêtre, il pourra sélectionner les fichiers journaux qu'il veut supprimer. Cette gestion est illustrée dans la figure IV.14.

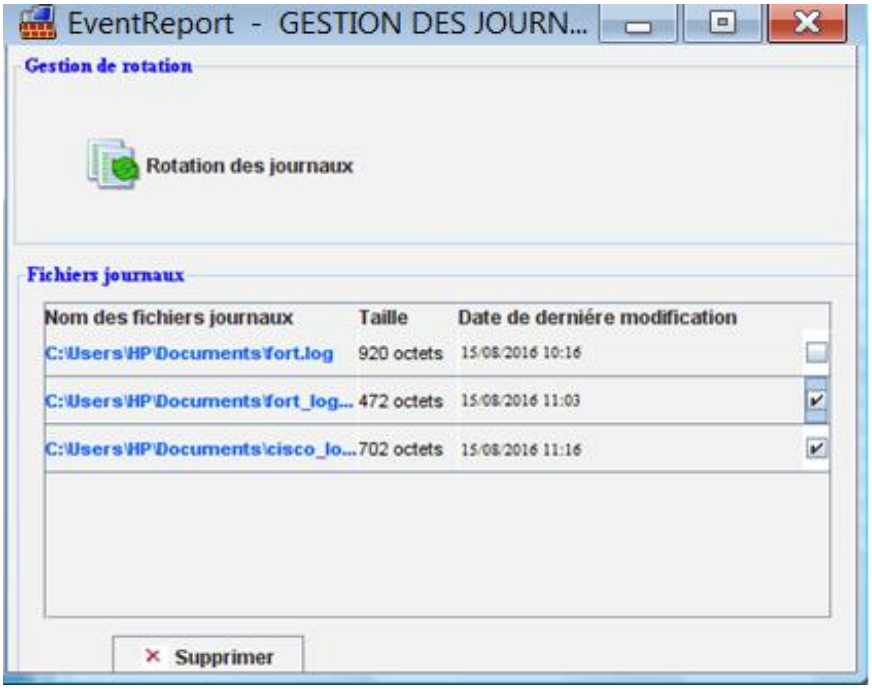


Figure V.14 Interface de gestion des fichiers journaux

Une fois sélectionnés, les fichiers journaux seront automatiquement supprimés du disque, et la fenêtre « GSTION DES JOURNAUX » ne présentera que les fichiers existants.

En cliquant sur le bouton « Rotation des journaux », une interface de planification de rotation des fichiers journaux des pares feu est présentée à l'utilisateur. Le choix de condition de rotation peut s'effectuer suivant la taille des journaux ou selon une fréquence de temps (hebdomadaire ou quotidienne), comme l'indique la figure IV.15.

Figure IV.15 Interface de gestion de rotation des fichiers journaux.

IV.6.Conclusion

Nous avons entamé dans ce dernier chapitre, la partie consacrée aux détails de fonctionnement de notre système de gestion et d'analyse de fichiers journaux en expliquant le fonctionnement de quelques menus et interfaces.

Conclusion Générale

Dans un environnement informatique de plus en plus évolutif et risqué, la bonne gestion des ressources communes et le contrôle des paramètres de sécurité s'avèrent indispensables aux organisations. Pour cela, les administrateurs des systèmes d'information sont amenés à examiner et explorer régulièrement les traces électroniques de fonctionnement des différents services.Notre projet s'intéresse en particulier à la problématique des fichiers journaux générés de façon systématique par les différents pares feu. Chaque activité, même aussi simple que le fait de se connecter à Internet, d'envoyer un email ou d'imprimer un fichier, laisse plusieurs traces à partir desquelles il est possible de reconstruire qui a fait quoi et à quel moment. En fait, la plupart des logiciels écrivent des fichiers journaux, une simple recherche de tous les fichiers «*.Log », sur le disque dur, permet de se rendre compte de l'ampleur de la situation.

Dans ce stage que nous avons effectué à la société **Telecom Information Service**, nous avons tout d'abord fait notre investigation pour nous familiariser avec la discipline d'analyse des fichiers journaux (surtout ceux des pares feu), et ce en s'intéressant à la classification de ces fichiers selon le type, la source, le format ou le contenu, ce qui nous a permis de nous rendre compte de la diversité phénoménale de ces fichiers.

Une fois la problématique est définie, la solution que nous avons proposée consiste à concevoir et élaborer une application permettant une analyse centralisée des fichiers journaux des pares feu, et ce en les centralisant, les intégrant dans une base de données relationnelle, les analysant, et en générant des rapports adéquats qui permettent aux administrateurs d'avoir une vue claire sur l'utilisation de leurs pares feu, et de les guider dans l'élaboration d'une politique de sécurité adaptée aux besoins.

En complément à la formation essentiellement, théorique et pratique dispensée au sein de **l'Université Virtuelle de Tunis**, ce stage représente une opportunité de choix pour mettre nos connaissances en relief, les consolider et surtout en acquérir d'autres sur le plan pratique. Nous pensons que notre stage de PFE était fructueux et très intéressant, entre autres pour les raisons suivantes :

▪ **Le contact avec la vie réelle des développeurs** : Notre stage nous a permis de nous rendre compte que le plus difficile dans la vie d'un informaticien n'est pas les examens ou les contrôles continus, mais les problèmes rencontrés lors du développement d'une application réelle. La connaissance de ces problèmes d'ordre technique, ajoutés aux contraintes de satisfaire les exigences des utilisateurs dans les meilleurs délais tout en respectant les budgets, était très instructive.

▪ **L'utilisation et l'apprentissage de nouveaux outils de développement** : Pour la réalisation de ce projet, nous étions obligés de nous familiariser avec des outils et des composants que nous n'avons pas – réellement – eu une connaissance suffisante sur leurs fonctionnements. Ce qui nous a pris un certain temps pour les maîtriser.

▪ **L'acquisition du réflexe de recherche et d'autoformation** : Les problèmes rencontrés lors de la réalisation de ce projet nous ont souvent poussés à effectuer des recherches, de consulter plusieurs sources de documentation afin de trouver des réponses aux questions posées. Ces recherches nous ont permis d'une part d'avancer dans la réalisation de notre projet, et d'autre part d'acquérir le reflex de la recherche opérationnelle efficace et indispensable pour s'auto former.

Malgré son intérêt comme solution logicielle d'analyse des fichiers journaux des pares feu, nous ne pouvons prétendre que ce travail pourrait être une solution complète et définitive pour une gestion efficace de la journalisation au sein d'une organisation. Ceci est essentiellement dû à la durée relativement courte du stage, qui ne nous a pas permis de cerner tous les aspects d'un sujet aussi vaste que la gestion des fichiers journaux des pares feu des grandes organisations.

Malgré la difficulté de la tâche d'autocritique, voilà un ensemble de limites qui pourraient être un point de référence pour améliorer notre travail :

- **La non-couverture du cycle de vie des fichiers journaux des pares feu**: En effet, notre travail se limite à l'activité d'analyse des fichiers journaux, or un fichier journal (pas nécessairement d'un pare feu) a un cycle de vie beaucoup plus long. D'où il faut penser à informatiser la collecte, le transfert de ces traces d'activités des systèmes d'information.

- **La non-connaissance automatique des formats des fichiers journaux** : Certes, notre application offre la possibilité d'analyser les fichiers journaux des pares feu indépendamment de leurs formats, mais un mécanisme de reconnaissance et d'aide à la définition des formats des fichiers journaux des pares feu les plus populaires serait d'un grand intérêt.

Bibliographie

[1] Hainaut, Jean-Luc: « Base de données et Modéles de calcul - outils et méthodes pour l'utilisateur », Dunod, 2002.

[2] La, Michel: « Penser objet avec UML et JAVA », Dunod, 2004.

[3] Ramez Elmasri et Shamkant Navathe: «Conception et architecture des bases de données », PearsonEducation, 2004.

[4] Pascal Roques et Franck Vallée: « UML 2 en action. De l'analyse des besoins à la conception J2EE », EYROLLES, 2004.

Netographie

http://support.microsoft.com consulter le 01/06/2016
http://www.fortinet.com consulter le 07/04/2016
http://www.cisco.com consulter le 07/04/2016
http://etudiant.univ-mlv.fr consulter le15/06/2016
http://www.mysql.com/ consulter le 15/05/2016
http://fadace.developpez.com/sgbdcmp consulter le 18/05/2016
http://kc.forticare.com/ consulter le 18/07/2016
http://www.frameip.com/firewall consulter le 12/06/2016
http://fr.wikipedia.org/ consulter 04/04/2016
http://laurent-audibert.developpez.com/Cours-UML/ consulter le 11/06/2016
http://www.jmdoudoux.fr/java consulter le 08/07/2016
http://www.unity3d-france.com/unity consulter le 30/08/2016

ANNEXE A : FORMATS DES FICHIERS JOURNAUX

Pour donner une idée sur la diversité phénoménale des formats des fichiers journaux, nous présentons ci-dessous les listes des différents champs des fichiers journaux des pares feu Cisco Pix, FortiGate et StoneGate :

Cisco Pix	FortiGate	StoneGate
Message	device ID	log ID
message code	Type	node ID
Protocol	Subtype	facility
source IP	Priority	Type
destination IP	Duration	event
source hostname	policy ID	action
destination hostname	source IP	protocol
source port	source hostname	source IP
destination port	source port	destination IP
source side	source interface	source port
destination side	destination IP	destination port
geographic location	destination hostname	rule ID
Interface	destination port	NAT source IP
Direction	destination interface	NAT destination IP
foreign IP	translated IP	NAT source port
foreign port	translated port	NAT destination port
global IP	icmp type	flags
global port	icmp code	source interface
local IP	Status	protocol agent

local port	Protocol	alert name
service name	Service	syslog message
URL	Message	icmp type
Flags	Accesses	icmp code
User	Visitors	ICMP ID
Command	Sent	IPSEC SPI
Type	Received	RTT
List	Duration	authenticated name
Events	Received packets	source VLAN
Page views		destination VLAN
Unique source IP		firewall engine ID
Bytes		info message
Destination bytes		Hits
Duration		visitors
		Bytes received

ANNEXE B : EXEMPLES DES FICHIERS JOURNAUX

Nous présentons dans cette partie des exemples des fichiers journaux générés par les pares feu Cisco Pix et FortiGate :

```
August 25 2016 09:55:55: %PIX-6-302005: Built UDP connection for faddr
203.124.140.107/12520 gaddr 10.0.0.187/53 laddr 192.168.0.2/53

August 25 2016 09:55:56: %PIX-6-302005: Built UDP connection for faddr
195.146.160 3/16708 gaddr 10.0.0.187/53 laddr 192.168.0.2/53

August 25 2016 09:56:00: %PIX-6-106015: Deny TCP (no connection) from
192.168.0.2/2796 to 192.168.80.1/1719 flags SYN ACK on interface inside

August 25 2016 09:56:02: %PIX-6-302005: Built UDP connection for faddr
194.114.201.14/46474 gaddr 10.0.0.187/53 laddr 192.168.0.2/53

August 29 2016 09:56:02: %PIX-6-302006: Teardown UDP connection for faddr
```

Exemples de fichier logs d'un pare feu de type Cisco Pix.

```
2016-08-25 21:12:01 log_id=0420073001 type=ips subtype=anomaly pri=critical
attack_id=100663399 src=10.10.1.2 dst=10.10.1.4 src_port=2000 dst_port=21
interface=external src_int=n/a dst_int=n/a status=clear_session proto=6 service=ftp
msg="anomaly: syn_fin[Reference: http://www.fortinet.com/ids/ ID100663399]"

2016-08-25 21:12:05 log_id=0420073002 type=traffic subtype=violation pri=alert
attack_id=100663399 src=10.10.1.2 dst=10.10.1.4 src_port=2110 dst_port=80
interface=external src_int=10-22-BB-99-CC-2B  dst_int=00-50-BF-9E-C3-2B status=accept
proto=6 service=http

2016-08-25 21:12:10 log_id=0420073003 type=event subtype=system pri=Warning
attack_id=100663399 src=10.10.1.2 dst=10.10.1.4 src_port=3150 dst_port=23
interface=external src_int= 23-26-FE-91-13-CB dst_int=00-50-BF-9E-C3-2B status=accept
proto=6 service=telnet
```

Exemples de fichier logs d'un pare feu de type FortiGate.

ANNEXE C : LES DIFFERENTS TYPES DES PARES FEU

1. Les pares feu bridge

Ces derniers sont relativement répandus. Ils agissent comme des vrais câbles réseaux avec la fonction de filtrage en plus, d'où leur appellation de pare feu. Leurs interfaces ne possèdent pas d'adresses IP, et ne font que transférer les paquets d'une interface à une autre en leurs appliquant des règles prédéfinies. Cette absence est particulièrement utile, car cela signifie que le pare feu est indétectable pour un hacker (pirate). En effet, quand une requête ARP est émis sur le câble réseau, le pare feu ne répondra jamais. Son adresse Mac ne circulera jamais sur le réseau, et co mme il ne fait que « transmettre » les paquets, il sera totalement invisible sur le réseau. Cela rend impossible toute attaque dirigée directement contre le pare feu, étant donné qu'aucun paquet ne sera traité par ce dernier comme étant sa propre destination. Donc, la seule façon de le contourner est de passer outre ses règles de drop. Ces pares feu se trouvent typiquement sur les commutateurs.

❖ **Avantages**

- Impossible de l'éviter (les paquets passeront par ses interfaces).
- Peu coûteux.

❖ **Inconvénients**

- Possibilité de le contourner (il suffit de passer outre ses règles).
- Configuration souvent contraignante.
- Les fonctionnalités présentes sont très basiques (filtrage sur adresse IP, port, etc.).

2. Les pares feu matériels

Ils se trouvent souvent sur des routeurs achetés dans le commerce par des grands constructeurs comme Cisco ou Nortel. Intégrés directement dans la machine, ils font office de

« boite noire », et ont une intégration parfaite avec le matériel. Leurs configurations sont souvent relativement ardues, mais leurs avantages : Leurs interactions avec les autres fonctionnalités du routeur sont simplifiées par leur présence sur le même équipement réseau. Souvent relativement peu flexible en termes de configuration, ils sont aussi peu vulnérables aux attaques. De plus, étant souvent très liés au matériel, l'accès à leurs codes est assez difficile, et le constructeur a eu toute latitude pour produire des systèmes de codes « signés » afin d'authentifier le logiciel (système RSA ou assimilé). Ce système n'est implanté que dans les hauts de gamme, car cela évite un remplacement du logiciel par un autre non produit par le fabricant, ou toute modification de ce dernier, rendant ainsi le pare feu très sûre. Son administration est souvent plus aisée que les pares feu bridges, les grandes marques des routeurs utilisant cet argument comme argument de vente. Leurs niveaux de sécurité sont de plus très bon, sauf découverte de faille éventuelle comme tout pare feu. Néanmoins, il faut savoir que nous la somme totalement dépendant du constructeur du matériel pour sa mise à jour, ce qui peut être, dans certains cas, assez contraignant. Enfin, seules les spécificités prévues par le constructeur du matériel sont implémentées. Cette dépendance induite que si une possibilité intéresse une certaine organisation sur un pare feu d'une autre marque, son utilisation est impossible. Il faut donc bien déterminer à l'avance ses besoins et choisir le constructeur du routeur avec soin.

❖ **Avantages**

- Intégré au matériel réseau.
- Administration relativement simple.
- Bon niveau de sécurité.

❖ **Inconvénients**

- Dépendant du constructeur pour les mises à jour.
- Souvent peu flexibles.

3. Les pares feu logiciels

Présents à la fois dans les serveurs et les routeurs « faits maison », on peut les classer en plusieurs catégories :

-Les pares feu personnels

Ils sont assez souvent commerciaux et ont pour but de sécuriser un ordinateur particulier, et non pas un groupe d'ordinateurs. Souvent payants, ils peuvent être contraignants

et quelquefois très peu sécurisés. En effet, ils s'orientent plus vers la simplicité d'utilisation plutôt que vers l'exhaustivité, afin de rester accessible à l'utilisateur final.

❖ **Avantages**

- Sécurité en bout de chaîne (le poste client).
- Personnalisation assez facile.

❖ **Inconvénients**

- Facilement contournable.
- Difficile à départager.

Les pares feu professionnels

Tournant généralement sous Linux, car cet OS offre une sécurité réseau plus élevée et un contrôle plus adéquat, ils ont généralement pour but d'avoir le même comportement que les pares feu matériels des routeurs, à ceci prêt qu'ils sont configurables à la main. Le plus courant est iptables (anciennement ipchaines), qui utilise directement le noyau Linux. Toute fonctionnalité des pares feu des routeurs est potentiellement réalisable sur une telle plateforme.

❖ **Avantages**

- Personnalisables.
- Niveau de sécurité très bon.

❖ **Inconvénients**

- Nécessite une administration système supplémentaire.

⇨ Ces pares feu logiciels ont néanmoins une grande faille : Ils n'utilisent pas la couche basse réseau. Il suffit donc de passer outre le noyau en ce qui concerne la récupération de ces paquets, en utilisant une librairie spéciale, pour récupérer les paquets qui auraient été normalement « droppés » par le pare feu. Néanmoins, cette faille induite de s'introduire sur l'ordinateur en question pour y faire des modifications... chose qui induit déjà une intrusion dans le réseau, ou une prise de contrôle physique de l'ordinateur, ce qui est déjà synonyme d'inefficacité de la part des pares feu.