REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEINGENEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE SAAD DAHLEB - BLIDA 1 -

FACULTE DES SCIENCES

DEPARTMENT D'INFORMATIQUE

Mémoire de fin d'études

Présenté en vue de l'obtention du diplôme de master

Domaine : Informatique

Filière : Informatique

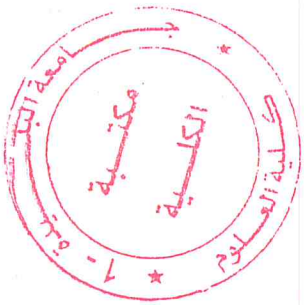
Spécialité : Sécurité des systèmes d'information

Thème: CONCEPTION ET RÉALISATION D'UN SYSTÈME DE SÉCURITÉ A L'AIDE D'OUTILS DE GESTION DES ÉVÉNEMENTS (SIEM)

SESSION: JUIN 2017

Présenté par:

- BENCHERCHALI Nasreddine
- BENBADA Abdessalam



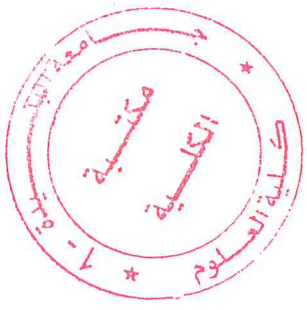
Devant le jury:

Président de jury: Mr. CHIKHI NACIM FATEH

Examinateur: Mme. TOUAHRI DALILA

Prometteur : Mr. ZAIR MUSTAPHA

Encadreur: Mr. BELMAHDI EMIR FARES


جامعة الإخوة منتوري قسنطينة
مكتبة
كلية العلوم

REMERCIMENTS

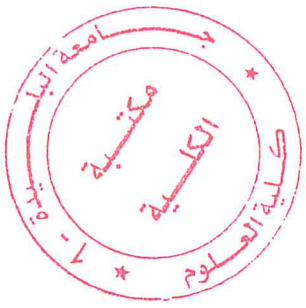
Nous remercions très vivement, nos enseignants d'avoir accepté d'être membres du jury de ce mémoire.

La réalisation de ce rapport ayant été rendue possible grâce au concours de plusieurs personnes, nous voudrions leur témoigner notre reconnaissance.

Nos remerciements vont particulièrement et en premier lieu à Monsieur le Directeur Général d'UNIDEES et tout son personnel pour avoir accepté de nous confier cette mission et surtout pour leurs disponibilités pendant le stage.

Nous souhaiterions adresser notre gratitude à Monsieur **BELMAHDI Emir Fares**, notre encadreur, qui a non seulement dirigé nos travaux mais a pour son mérite, soutenu une logique de travail ayant abouti à la cristallisation de nos espérances. Notre reconnaissance lui étant destinée, sera sans fin.

Un grand merci à notre prometteur monsieur **M.ZAIR** qui nous a été d'une très grande aide concernant la correction et la rédaction de ce document, et aux camarades et amis qui nous ont apporté leur soutien en lieu et date, tout au long de notre travail.



RÉSUMÉ

La sécurité informatique fait à face à plusieurs défis ; Les exploits gagnent en agressivité avec une vitesse incroyable, et il est devenu plus crucial que jamais de sécuriser nos infrastructures, mais cela risque d'être plus compliqué qu'on l'imagine.

Puisque les attaques modernes utilisent des méthodes complexes, moyennant plusieurs vecteurs d'attaque pour atteindre leur but, le seul moyen pour espérer contrer cela, est d'avoir une vue globale sur le système, et de savoir traduire et corréler les informations contenues dans les évènements détectés par les dispositifs de sécurité, en des indications de menace potentielle.

Pour cette fin, il existe une source d'information rarement exploitée, pourtant facilement accessible : les logs.

En effet, on peut utiliser les logs pour extraire des informations concernant le lancement des services, l'activité des utilisateurs, les erreurs d'exécutions...etc. mais aussi les évènements de sécurité.

Si on peut automatiser la tâche de collection, d'analyse, et d'extraction d'information depuis les logs puis les corréler avec d'autres données depuis d'autres sources, on pourrait détecter, ou même prévenir une attaque beaucoup plus facilement.

Dans ce document, on va présenter une solution de surveillance et de détection d'incidents à base de fichiers journaux en utilisant des outils open source.

On va montrer les difficultés liées à la gestion des logs, les contraintes de conservation d'information, ainsi que la complexité de la corrélation des évènements.

On va aussi détailler la démarche d'implémentation, à savoir le concept, la logique, et les outils utilisé pendant celle-ci.

Enfin, on va conclure avec la présentation de quelques perspectives pour l'amélioration du projet dans des versions futures.

Mots clés : SIEM, Logs, Sécurité, Détection

SIGLES ET ABRÉVIATION:

DOS: Denial of Service.

DDOS: Distributed Denial of Service.

SIEM: Security Information Management System.

IDS: Intrusion Detection System.

IPS: Intrusion Prevention System.

UNIDEES: Univers d'idées.

HTML: Hyper Text Markup Language.

CSS: Cascading Style Sheet.

AJAX: Asynchronous JavaScript and Xml.

UTM: Unified Threat Management.

ISO: International Organization for Standardization.

PMP: Project Management Professional.

ITIL: Information Technology Infrastructure Library.

GIF: Graphics Interchange Format.

NRT : Near Real Time.

RSSI : Résponsable de sécurité des systèmes d'information

SOMMAIRE

TABLE DES FIGURES ET ILLUSTRATIONS

INTRODUCTION

Il existe une nouvelle ère dans la sécurité informatique : des exploits fatals, qui peuvent détruire toute une industrie, des malwares qui peuvent se répandre mondialement sont en train d'être découverts chaque semaine, nous obligeons à repenser ce que nous jugeons comme "sécurisé".

Et tant que les cyber-attaques continuent à gagner en ampleur et en sophistication, les entreprises doivent être capables à adapter leurs systèmes et leurs politiques de sécurité pour suivre cette évolution continue.

Face à cette réalité, les entreprises, ont évidemment besoin de sécuriser leurs réseaux, pour protéger leur biens critiques, pour cela, ils implémentent typiquement des IDS, des IPS, des pare-feu, et des technologies similaires, dans un effort pour atténuer les risques. Or ils oublient souvent une étape importante : la gestion des journaux.

Les serveurs, les pare-feu et autres équipements informatiques conservent des fichiers journaux qui enregistrent les transactions et les événements détectés. Ces informations peuvent fournir des indices importants concernant les activités hostiles affectant le réseau à l'intérieur et à l'extérieur. Les données contenues dans ces fichiers peuvent également fournir des informations pour identifier et résoudre les problèmes d'équipement, y compris les problèmes de configuration et les pannes matérielles.

Mais comme les journaux enregistrent toute action prise sur un réseau, et sont générés par virtuellement tout type de dispositif sur le réseau, leur volume risque d'être très important, au point où il devient très difficile de les gérer et donc, de les exploiter facilement.

De tous ces problèmes, le besoin d'une solution sécurité qui automatise l'exploitation des logs tout en fournissant plus de visibilité sur le système, et est né. Et le SIEM est la réponse à ce besoin.

C'est dans ce sens que la Société UNIDEES (Univers d'idées) Algérie nous a ouvert ses portes, pour la réalisation de notre stage de fin de formation sur le thème « CONCEPTION ET RÉALISATION D'UN SYSTÈME DE SÉCURITÉ A L'AIDE D'OUTILS DE GESTION DES ÉVÉNEMENTS ».

1. Problématique

L'un des principaux facteurs qui contribuent à l'échec de la détection des attaques est un manque général de visibilité sur le système qu'on essaye de sécuriser. En effet, le fait de n'avoir aucun moyen facile de détecter, suivre, et d'analyser le flux de données dans notre système peut gravement paralyser notre capacité à répondre et à remédier aux incidents en temps.

En outre, ils existent d'autres problèmes liés à la sécurité qui complique le travail d'un RSSI (responsable de la sécurité des systèmes d'information), tel que :

1.1.La Gestion des logs

Les Logs sont difficiles à gérer; En effet, dans un environnement typique de travail, Ils existent des pare-feu, des systèmes de détection et de prévention des intrusions (IDS / IPS), de contrôles d'accès, des systèmes de surveillance, et des systèmes de gestion de menace (Unified Threat Management - UTM) ... etc.

Chacun de ces systèmes produit des journaux d'événements. La plupart ont leurs propres formats d'expression de journal. Le nombre de journaux (ou logs) générés par ces systèmes est très important, au point où il devient très peu pratique d'essayer de les lire manuellement. En outre, en raison des différents formats de logs provenant de différentes sources, il peut être très difficile et/ou très couteux en termes de temps de les analyser.

1.2.Manque de communication entre les terminaux

Différents dispositifs sur le réseau utilisent différents protocoles et différentes techniques pour traiter l'information, et si un problème se présente, chaque dispositif essaye de travailler dessus individuellement, seulement avec les informations qu'il a localement. Cela peut fonctionner dans certains cas, mais le plus souvent, le problème nécessite la collaboration de nombreuses parties du système afin d'identifier et neutraliser la menace sans un système centralisé, où toutes les données sont corrélées, les informations compris dans les logs peuvent être perdues avant d'être exploitées.

1.3.Lenteur de la réponse aux incidents

Les problèmes susmentionnés font en sorte qu'un responsable de sécurité des systèmes d'information (RSSI) perd tellement de temps à lire, analyser et tracer la source des problèmes, que le dommage peut déjà être déjà fait, avant qu'il réussisse à répondre au problème, ou une différente menace peut avoir pénétré le système alors qu'il était occupé avec les logs.

2. Objectifs

L'objectif de ce travail est d'implémenter un système de détection, d'analyse, et de signalisation d'évènements à base de fichiers journaux.

Le produit final devrait être en mesure de:

- Gérer les fichiers journaux (collection, centralisation, stockage)
- Détecter les incidents de sécurité.
- Détecter des incidents relatifs à des besoins spécifiques
- Corréler les informations requises
- Identifier les menaces
- Offrir des outils nécessaires pour:
    - la détection
    - La signalisation
    - Le suivi, et la remédiation des problèmes détectés
- Etre complètement configurable pour permettre un niveau de personnalisation et d'adaptabilité à la solution
- Archiver les évènements.

3. Discussion

Avec les problèmes mentionnés ci-dessus, et le reste des problèmes dans le sphère de la sécurité informatique, qui vont de la sécurité des applications Web, à l'infection de logiciels malveillants, à l'ingénierie sociale, avoir la possibilité de suivre l'état de sécurité de votre système en temps réel n'est pas juste un bonus, c'est une nécessité. L'une des solutions proposées sur le marché aujourd'hui pour prévenir et de minimiser ces risques est le SIEM (System Information and Event Manager).

4. Organisation du mémoire :

Le mémoire est articulé en 4 parties principales :

Dans le premier chapitre on va introduire des notions de base sur les SIEM, ensuite on va donner un aperçu sur les différents types d'attaques qui existent, ainsi que sur quelques méthodes de détection, on va ensuite parler sur quelques solutions SIEM et faire une petite comparaison de ces fonctionnalités et on terminera par une discussion sur les difficultés rencontrés au début de la phase de conception. SIEM.

Dans le deuxième chapitre on va détailler la conception et l'architecture de notre système, ainsi que la méthodologie que nous avons adopté.

Dans le troisième chapitre on va présentera les outils utilisé dans le développement de notre application leur fonctionnement, leur configuration et la manière dont ils ont été implémentés, ainsi qu'une évaluation des performances de notre application.

Dans le quatrième chapitre on va donner un aperçu des diverses fonctionnalités de notre l'application, illustrée avec des prise d'écran.

Dans le cinquième chapitre est dernier chapitre on terminera par une conclusion et perspective, pour des améliorations future.

ÉTAT DE L'ART (Partie I)

Dans cette première partie du chapitre, on va découvrir qu'est-ce qu'un SIEM, puis on va analyser l'anatomie de ce dernier, à savoir ses composants, ses capacités, et comment il fonctionne typiquement. On va aussi mettre la lumière sur les logs, les soucis liés à leur gestion ainsi que leur relation avec différents composants du SIEM.

1. C'est quoi un SIEM ?

Le SIEM est une solution qui vise à donner une vue d'ensemble du système de sécurité de l'information d'une organisation en regroupant toutes les données vers une interface centralisée, où les informations sur les événements de sécurité peuvent être analysées et corrélées, pour mieux détecter et signaler les menaces en temps-réel [1].

« SIEM » est un acronyme pour "System information and Event Management" et est la combinaison de deux approches existantes dans la sécurité adaptative :

- SIM (Security Information Management) : Un type de logiciel qui automatise la collecte, la normalisation, et l'analyse des données des journaux d'événements à partir des dispositifs de sécurité tels que les pare-feu, les serveurs proxy, les systèmes de détection d'intrusion et les logiciels antivirus [1].
- SEM (Security Event Management) : Ce produit fournit une gestion forte des événements, une analyse des menaces en temps réel, la visualisation, billetterie, et la réponse aux incidents, ainsi que des opérations de sécurité [1].

Un SIEM propose en plus de cela, la corrélation des évènements collectés, l'analyse en temps-réel, et l'archivage.

2. Bénéfices

Une solution SIEM offre de nombreux avantages si elle est déployée correctement. Principalement en sécurité, mais s'étend aux avantages économiques et stratégiques :

- Maintenir la stabilité du réseau.
- Détecter les incidents qui, autrement, ne seraient pas détectés.
- Améliorer l'efficacité de la gestion des incidents.

3. Fonctionnalités

Un SIEM standard a typiquement les fonctionnalités suivantes (sans lesquelles un système ne peut pas être considéré comme un SIEM) :

1. Collection de logs.
2. Analyse/Normalisation des logs.
3. Moteur de Corrélation.
4. Stockage et Archivage.
5. Tableau de bord (Interface de surveillance).

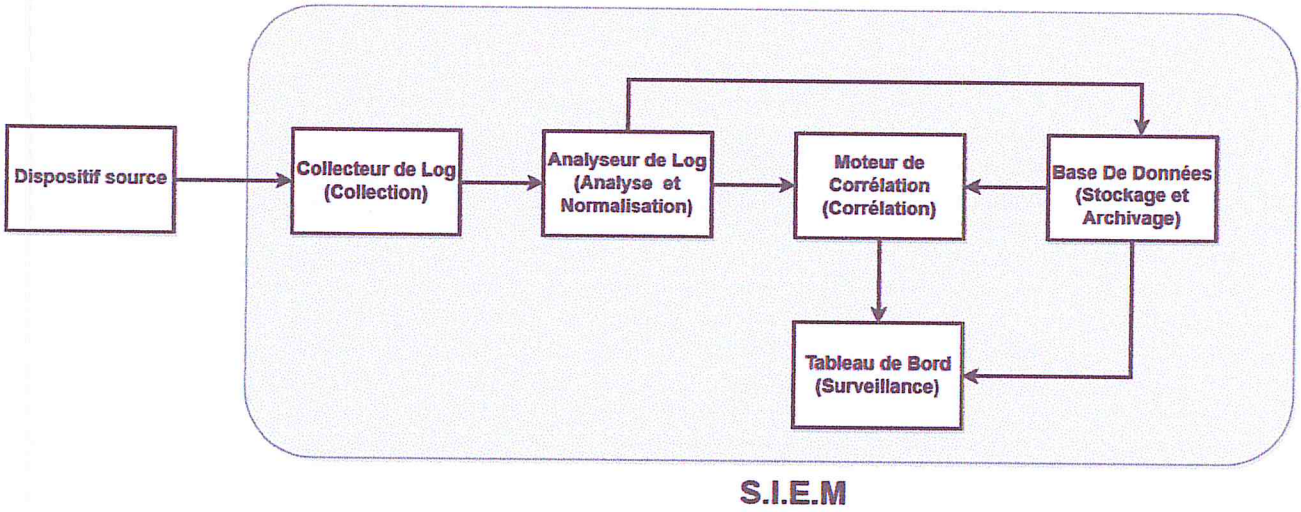


Figure 1 : Architecture global du SIEM

3.1. Collection des logs

3.1.1. Définition des Logs

Les Logs (fichiers journaux en français) sont des fichiers qui enregistrent chaque action prise dans un système, que ce soit un OS ou un logiciel simple, chaque système génère son propre type de logs suivant une structure définie. Un pare-feu génère un type spécifique de logs (Access Logs), une machine Linux génère des « system logs » et une machine Windows peut générer un « event log », etc., [2].

La gestion des journaux est la première clé de toute solution SIEM. Réussir à collecter les journaux pertinents pour la sécurité, produits par les dispositifs sur le réseau, nous aidera à être en mesure d'extraire des informations de sécurité importantes et, par conséquent, d'agir en temps pour sécuriser notre système. Sans une bonne gestion des journaux, la fonctionnalité SIEM sera Impossible à réaliser.

Lorsqu'on commence à considérer les besoins de gestion des journaux, il est nécessaire de définir les limites à l'intérieur desquelles la solution sera contrainte [2]. À cette fin, on devrait définir les éléments suivants :

3.1.1.1. Conservation (Pour combien de temps)

Des règlements ou lois, ainsi que des exigences fonctionnelles peuvent vous obliger à conserver certains types de données pour une période donnée (rétention) [1], [2].

Ils peuvent également dicter comment vous disposez de l'information après une période de temps donnée (destruction) [1], [2].

Vous devez prendre en considération les deux points susmentionnés ainsi que les besoins fonctionnels de votre entreprise, quand vous déterminez la durée de temps pendant laquelle vous allez conserver vos logs [1], [2].

3.1.1.2. Taille (Combien)

Même dans une petite entreprise, la quantité d'évènements produits par les dispositifs sur le réseau est probablement très importante, et peut facilement accabler l'espace de stockage réservé pour la journalisation, vous devez donc décider la quantité de logs que vous voulez retenir, en définissant une politique de rotation de logs [1], [2].

3.1.1.3. Type (Quoi)

Les logs se présentent sous nombreuses tailles et formes, et ils sont générés par presque tout type de dispositifs. Il est essential pour la gestion des logs, ainsi que pour le bon fonctionnement du SIEM en général de déterminer quels types d'évènements faut-il sauvegarder [1], [2].

3.1.2. Comment collecter ?

Lorsque les journaux sont générés, ils sont stockés dans des fichiers à l'intérieur du système en question. Pour les collecter, les solutions SIEM utilisent généralement une de ces méthodes :

3.1.2.1. Pousser (Push)

Cette méthode consiste à mettre en place un récepteur, puis à pointer le périphérique source vers ce récepteur, qui finira par envoyer des journaux à notre SIEM. Cette méthode a l'avantage de la facilité de déploiement et configuration. Cependant, elle a quelques inconvénients, par exemple : l'utilisation de certains protocoles tels qu'UDP qui peut introduire certaines vulnérabilités de sécurité. Donc, savoir exactement quel périphérique envoie quels journaux est crucial pour cette solution [1], [2].

3.1.2.2. Tirer (Pull)

Cette méthode est à l'opposé de la méthode push, dans le fait qu'au lieu de laisser les périphériques source nous envoyer des journaux, nous configurons le SIEM pour se connecter au périphérique source et commencer à récupérer des journaux à partir de là. L'inconvénient de cette méthode est qu'il y a un laps de temps considérable entre les actes de récupération et de vérification des journaux, ce qui cause les journaux à être retardé, et donc, brise l'aspect « temps réel » du SIEM [1], [2].

3.1.2.3. Personnalisé

Parfois, il est nécessaire de construire sa propre méthode pour collecter les journaux. Construire sa propre méthode de collecte et de synthèse des journaux peut être intensif en main-d'œuvre et nécessite beaucoup de temps, mais si cela se fait correctement, cela fera que les journaux seront tirés directement de leur système natif vers le SIEM. Un avantage de créer sa propre méthode de collecte serait qu'on aurait le contrôle sur tous les processus de récupération et d'analyse qui ont lieu, ainsi que la flexibilité pour extraire les journaux qui ne sont pas directement supportés, ce qui élargit considérablement la fonctionnalité du SIEM [1], [2].

3.2. Analyse & Normalisation

Les logs se présentent sous différents formats, chaque système a sa propre méthode pour les structurer.

Citons quelques-uns :

La figure suivante présente une prise d'écran d'un fichier log du serveur web Apache.

```
::1 - - [10/Mar/2016:20:10:54 +0100] "GET / HTTP/1.1" 302 - "-" "Mozilla/5.0 (Windows NT 10.0;
::1 - - [10/Mar/2016:20:10:54 +0100] "GET /dashboard/ HTTP/1.1" 200 6904 "-" "Mozilla/5.0 (Wind
::1 - - [10/Mar/2016:20:10:54 +0100] "GET /dashboard/stylesheets/normalize.css HTTP/1.1" 200 68
::1 - - [10/Mar/2016:20:10:54 +0100] "GET /dashboard/stylesheets/all.css HTTP/1.1" 200 481308 "
::1 - - [10/Mar/2016:20:10:54 +0100] "GET /dashboard/javascripts/all.js HTTP/1.1" 200 189003 "h
::1 - - [10/Mar/2016:20:10:54 +0100] "GET /dashboard/javascripts/modernizr.js HTTP/1.1" 200 513
::1 - - [10/Mar/2016:20:10:55 +0100] "GET /dashboard/images/bitnami-xampp.png HTTP/1.1" 200 221
::1 - - [10/Mar/2016:20:10:55 +0100] "GET /dashboard/images/fastly-logo.png HTTP/1.1" 200 1770
::1 - - [10/Mar/2016:20:10:55 +0100] "GET /dashboard/images/xampp-logo.svg HTTP/1.1" 200 5427 "
::1 - - [10/Mar/2016:20:10:55 +0100] "GET /dashboard/images/social-icons.png HTTP/1.1" 200 3361
::1 - - [10/Mar/2016:20:10:55 +0100] "GET /dashboard/images/favicon.png HTTP/1.1" 200 2508 "-"
::1 - - [10/Mar/2016:20:11:49 +0100] "GET / HTTP/1.1" 302 - "-" "Mozilla/5.0 (Windows NT 10.0;
::1 - - [10/Mar/2016:20:19:59 +0100] "GET /htdocs HTTP/1.1" 404 1150 "-" "Mozilla/5.0 (Windows
::1 - - [10/Mar/2016:20:20:01 +0100] "GET /favicon.ico HTTP/1.1" 200 30894 "-" "Mozilla/5.0 (Wi
::1 - - [10/Mar/2016:20:20:23 +0100] "GET / HTTP/1.1" 302 - "http://localhost:8080/htdocs" "Moz
::1 - - [10/Mar/2016:20:20:44 +0100] "GET /phpmyadmin/ HTTP/1.1" 200 12599 "http://localhost:80
::1 - - [10/Mar/2016:20:20:47 +0100] "GET /phpmyadmin/themes/pmahomme/jquery/jquery-ui-1.11.2.c
::1 - - [10/Mar/2016:20:20:47 +0100] "GET /phpmyadmin/js/codemirror/addon/hint/show-hint.css?v=
::1 - - [10/Mar/2016:20:20:47 +0100] "GET /phpmyadmin/js/codemirror/addon/lint/lint.css?v=4.5.1
::1 - - [10/Mar/2016:20:20:47 +0100] "GET /phpmyadmin/themes/pmahomme/css/printview.css?v=4.5.1
::1 - - [10/Mar/2016:20:20:47 +0100] "GET /phpmyadmin/js/codemirror/lib/codemirror.css?v=4.5.1
```

Figure 2: Log d'un serveur APACHE (ACCESS LOG)

La figure suivante présente une prise d'écran d'un fichier log du Windows Log.

```
2017-03-01 11:22:02, Info                  CBS    Appl: detectParent (exact match): Parent: Microsoft-Windc
2017-03-01 11:22:02, Info                  CBS    Appl: detect Parent, Package: Package_for_KB3199209~31bf3
2017-03-01 11:22:02, Info                  CBS    Appl: detectParent (exact match): Parent: Microsoft-Windc
2017-03-01 11:22:02, Info                  CBS    Appl: detect Parent, Package: Package_for_KB3199209~31bf3
2017-03-01 11:22:02, Info                  CBS    Appl: detectParent (exact match): Parent: Microsoft-Windc
2017-03-01 11:22:02, Info                  CBS    Appl: detect Parent, Package: Package_for_KB3199209~31bf3
2017-03-01 11:22:02, Info                  CBS    Appl: detectParent (exact match): Parent: Microsoft-Windc
2017-03-01 11:22:02, Info                  CBS    Appl: detect Parent, Package: Package_for_KB3199209~31bf3
2017-03-01 11:22:02, Info                  CBS    Appl: detectParent (exact match): Parent: Microsoft-Windc
2017-03-01 11:22:02, Info                  CBS    Appl: detect Parent, Package: Package_for_KB3199209~31bf3
2017-03-01 11:22:02, Info                  CBS    Appl: detectParent (exact match): Parent: Microsoft-Windc
2017-03-01 11:22:02, Info                  CBS    Appl: detect Parent, Package: Package_for_KB3199209~31bf3
2017-03-01 11:22:02, Info                  CBS    Appl: detectParent (exact match): Parent: Microsoft-Windc
2017-03-01 11:22:02, Info                  CBS    Appl: detect Parent, Package: Package_for_KB3199209~31bf3
2017-03-01 11:22:02, Info                  CBS    Appl: detectParent (exact match): Parent: Microsoft-Windc
2017-03-01 11:22:02, Info                  CBS    Appl: detect Parent, Package: Package_for_KB3199209~31bf3
2017-03-01 11:22:02, Info                  CBS    Appl: detectParent (exact match): Parent: Microsoft-Windc
2017-03-01 11:22:02, Info                  CBS    Appl: detect Parent, Package: Package_for_KB3199209~31bf3
2017-03-01 11:22:02, Info                  CBS    Appl: detectParent (exact match): Parent: Microsoft-Windc
```

Figure 3: Log CBS (Log Windows)

La figure suivante présente une prise d'écran d'un fichier log d'un pare-feu.

```
Oct 18 23:41:42 Retina-MacBook-Pro socketfilterfw[320] <Info>: smbd: Allow TCP CONNECT (in:1 out:0)
Oct 19 16:37:17 Retina-MacBook-Pro socketfilterfw[320] <Info>: iTunes: Allow TCP LISTEN  (in:0 out:1)
Oct 21 14:29:57 Retina-MacBook-Pro socketfilterfw[320] <Info>: smbd: Allow TCP LISTEN  (in:0 out:2)
Oct 21 14:29:57 Retina-MacBook-Pro socketfilterfw[320] <Info>: smbd: Allow TCP CONNECT (in:1 out:0)
Oct 21 17:06:59 Retina-MacBook-Pro socketfilterfw[320] <Info>: iTunes: Allow TCP LISTEN  (in:0 out:1)
Oct 21 17:11:35 Retina-MacBook-Pro socketfilterfw[320] <Info>: iTunes: Allow TCP LISTEN  (in:0 out:1)
Oct 21 19:30:59 Retina-MacBook-Pro socketfilterfw[320] <Info>: iTunes: Allow TCP LISTEN  (in:0 out:1)
Oct 22 14:40:03 Retina-MacBook-Pro socketfilterfw[311] <Info>: iTunes: Allow TCP LISTEN  (in:0 out:1)
Oct 22 15:14:03 Retina-MacBook-Pro socketfilterfw[311] <Info>: iTunes: Allow TCP LISTEN  (in:0 out:1)
Oct 22 17:25:38 Retina-MacBook-Pro socketfilterfw[311] <Info>: iTunes: Allow TCP LISTEN  (in:0 out:1)
Oct 22 23:23:41 Retina-MacBook-Pro socketfilterfw[311] <Info>: smbd: Allow TCP LISTEN  (in:0 out:2)
Oct 22 23:23:41 Retina-MacBook-Pro socketfilterfw[311] <Info>: smbd: Allow TCP CONNECT (in:1 out:0)
Oct 26 13:45:43 Retina-MacBook-Pro socketfilterfw[311] <Info>: smbd: Allow TCP LISTEN  (in:0 out:2)
Oct 26 13:45:43 Retina-MacBook-Pro socketfilterfw[311] <Info>: smbd: Allow TCP CONNECT (in:1 out:0)
Oct 26 14:45:43 Retina-MacBook-Pro socketfilterfw[311] <Info>: smbd: Allow TCP CONNECT (in:1 out:0)
Oct 26 15:45:44 Retina-MacBook-Pro socketfilterfw[311] <Info>: smbd: Allow TCP CONNECT (in:1 out:0)
Oct 26 19:46:15 Retina-MacBook-Pro socketfilterfw[311] <Info>: smbd: Allow TCP CONNECT (in:3 out:0)
Oct 30 12:54:29 Retina-MacBook-Pro socketfilterfw[311] <Info>: smbd: Allow TCP LISTEN  (in:0 out:2)
Oct 30 12:54:29 Retina-MacBook-Pro socketfilterfw[311] <Info>: smbd: Allow TCP CONNECT (in:6 out:0)
Oct 30 12:54:59 Retina-MacBook-Pro socketfilterfw[311] <Info>: smbd: Allow TCP CONNECT (in:6 out:0)
Oct 31 13:16:19 Retina-MacBook-Pro socketfilterfw[311] <Info>: iTunes: Allow TCP LISTEN  (in:0 out:1)
Nov  2 11:14:01 Retina-MacBook-Pro socketfilterfw[311] <Info>: launchd: Allow TCP LISTEN  (in:0 out:1)
Nov  2 11:14:01 Retina-MacBook-Pro socketfilterfw[311] <Info>: launchd: Allow TCP LISTEN  (in:0 out:1)
Nov  2 11:14:31 Retina-MacBook-Pro socketfilterfw[311] <Info>: kdc: Allow TCP LISTEN  (in:0 out:2)
Nov  5 14:58:33 Retina-MacBook-Pro socketfilterfw[311] <Info>: launchd: Allow TCP LISTEN  (in:0 out:1)
Nov  5 14:58:33 Retina-MacBook-Pro socketfilterfw[311] <Info>: launchd: Allow TCP LISTEN  (in:0 out:1)
Nov  5 15:57:52 Retina-MacBook-Pro socketfilterfw[311] <Info>: launchd: Allow TCP LISTEN  (in:0 out:2)
Nov  9 16:43:41 Retina-MacBook-Pro socketfilterfw[311] <Info>: iTunes: Allow TCP LISTEN  (in:0 out:1)
Nov 12 11:32:57 Retina-MacBook-Pro socketfilterfw[311] <Info>: iTunes: Allow TCP LISTEN  (in:0 out:1)
Nov 18 11:37:49 Retina-MacBook-Pro socketfilterfw[311] <Info>: iTunes: Allow TCP LISTEN  (in:0 out:1)
```

Figure 4: Log d'un pare-feu

En raison de la nature hétérogène de la façon avec laquelle les différents programmes génèrent et stockent l'information, lorsqu'on essaye de centraliser les journaux, il devient vite évident que la lecture, la comparaison et la corrélation de ces résultats manuellement prend trop de temps, et le traitement des journaux avec les mêmes critères n'est pas possible. Par conséquent, On doit les faire suivre le même format et la même structure en les normalisant, et en suivant une norme établie.

3.2.1. Standard de journaux (Syslog)

Syslog est une norme pour la journalisation des messages. Elle permet de séparer le logiciel qui génère les messages, le système qui les stocke et le logiciel qui les enregistre et les analyse [3].

Syslog utilise trois couches :

- La couche "Syslog content" est l'information de gestion contenue dans un message Syslog.
- La couche "Syslog application" gère la génération, l'interprétation, le routage et le stockage des messages Syslog.
- La couche "Syslog transport" met des messages sur le fil et les retire du fil.

3.2.2. Format de message Syslog

Le message Syslog à la définition ABNF [RFC5234] suivante :

SYSLOG-MSG = HEADER SP STRUCTURED-DATA [SP MSG]

HEADER = PRI VERSION SP TIMESTAMP SP HOSTNAME - SP APP-NAME SP PROCID SP MSGID
PRI = "<" PRIVAL ">"
PRIVAL = 1*3DIGIT ; range 0... 191
VERSION = NONZERO-DIGIT 0*2DIGIT
HOSTNAME = NILVALUE / 1*255PRINTUSASCII

APP-NAME = NILVALUE / 1*48PRINTUSASCII
PROCID = NILVALUE / 1*128PRINTUSASCII
MSGID = NILVALUE / 1*32PRINTUSASCII

TIMESTAMP = NILVALUE / FULL-DATE "T" FULL-TIME
FULL-DATE = DATE-FULLYEAR "-" DATE-MONTH "-" DATE-MDAY
DATE-FULLYEAR = 4DIGIT
DATE-MONTH = 2DIGIT; 01-12
DATE-MDAY = 2DIGIT; 01-28, 01-29, 01-30, 01-31 based on; month/year

FULL-TIME = PARTIAL-TIME TIME-OFFSET
PARTIAL-TIME = TIME-HOUR ":" TIME-MINUTE ":" TIME-SECOND [TIME-SECFRAC]

TIME-HOUR = 2DIGIT; 00-23
TIME-MINUTE = 2DIGIT; 00-59
TIME-SECOND = 2DIGIT; 00-59

```
TIME-SECFRAC    = "." 1*6DIGIT
TIME-OFFSET     = "Z" / TIME-NUMOFFSET
TIME-NUMOFFSET = ("+" / "-") TIME-HOUR ":" TIME-MINUTE
STRUCTURED-DATA = NILVALUE / 1*SD-ELEMENT
SD-ELEMENT      = "[" SD-ID *(SP SD-PARAM) "]"
SD-PARAM        = PARAM-NAME "=" %d34 PARAM-VALUE %d34
SD-ID           = SD-NAME
PARAM-NAME      = SD-NAME
PARAM-VALUE     = UTF-8-STRING; characters '"', '\' and; ']' MUST be escaped.

SD-NAME         = 1*32PRINTUSASCII; except '=', SP, ']', %d34 (")

MSG             = MSG-ANY / MSG-UTF8
MSG-ANY         = *OCTET; not starting with BOM
MSG-UTF8        = BOM UTF-8-STRING
BOM             = %xEF.BB.BF

UTF-8-STRING    = *OCTET; UTF-8 string as specified; in RFC 3629

OCTET           = %d00-255
SP              = %d32
PRINTUSASCII    = %d33-126
NONZERO-DIGIT   = %d49-57
DIGIT           = %d48 / NONZERO-DIGIT
NILVALUE        = "-"
```

3.3. Moteur de Corrélation

Un seul événement peut ne pas être pertinent pour la sécurité par lui-même, mais lorsqu'il est corrélé avec d'autres événements, cela peut conduire à une détection précoce des menaces. La corrélation d'événements est une technique permettant d'analyser un grand nombre d'événements tout en essayant d'identifier les événements les plus importants dans une grande masse d'information. Ceci est accompli en cherchant et en analysant les relations entre les événements, qui est à son tour fait en établissant des règles de corrélation, puis en comparant les événements détectés avec ces règles. S'il y a une correspondance et qu'une règle est déclenchée, la réponse correspondante (lancement d'une alerte, par exemple) sera activée [1].

Le moteur de corrélation est la partie la plus importante de la solution SIEM, sans elle un SIEM sera simplement un lieu pour la centralisation des événements [1].

Un moteur de corrélation est basé sur des règles de corrélation. Ces règles peuvent être très simples, comme élever un avertissement si quelqu'un a essayé un mauvais mot de passe pour un compte plus de 5 fois de suite ou très complexe, comme le blocage d'une adresse IP de

quelqu'un qui a essayé d'accéder à un compte d'administrateur à partir de différents adresses IP a des différents horodatages avec des techniques différentes [1].

Le moteur de corrélation, ou plus spécifiquement les règles de corrélation, sont la "sauce secrète" de toute solution SIEM. Les fournisseurs commerciaux de SIEM consacrent des ressources considérables et du temps à les créer et à les entretenir [1].

3.4. Stockage

Pour travailler avec les volumes de journaux qui entrent dans le SIEM, on a besoin d'un moyen de les stocker à des fins de conservation et pour l'analyse des tendances. Il existe généralement trois façons dont le SIEM peut stocker ses journaux : dans une base de données, un fichier texte ou un fichier binaire [1].

3.4.1. Base de données

Le stockage des journaux dans une base de données est la façon dont la plupart des SIEM stockent leurs journaux. La base de données est habituellement une plate-forme de BD standard telle qu'Oracle, MySQL, Microsoft ...etc. Cette méthode permet une interaction et une récupération faciles des données stockées car les appels de base de données font partie de l'application. [1]

3.4.2. Fichier texte

Un fichier texte plat est juste un fichier texte standard qui stocke l'information dans un format lisible par l'homme. Le fichier doit contenir un genre de délimiteur, qu'il s'agisse de virgule, d'onglet ou d'un autre caractère, afin que les événements peuvent être séparés les uns des autres, et par conséquent, d'être lus correctement. [1]

3.4.3. Fichier binaire

C'est un fichier utilisant un format personnalisé pour stocker des informations binaires qui sont utilisées uniquement par le SIEM spécifique qu'il a généré. Le SIEM est la seule application qui sait lire et écrire sur ce fichier hautement propriétaire. [1]

3.5. Tableau de bord (Dashboard)

Même après la collecte, la normalisation et la corrélation des logs, les données obtenues ne signifient rien à moins d'être exploitées, c'est pourquoi nous avons besoin d'une interface dans laquelle toutes les informations de sécurité sont rassemblées et peuvent être aperçues sous forme statistique.

Cette interface de surveillance (appelée désormais « tableau de bord ») devrait, à première vue, présenter un aperçu général de l'ensemble de l'environnement qui comprend des informations agrégées telles que le nombre d'événements, leur répartition dans système, leur temps de détection, ainsi que toute information jugée pertinente pour notre système de surveillance.

Les alertes peuvent être filtrées selon les règles appliquées, la date, la source, le type, la catégorie, sévérité...etc.

Dans le tableau de bord, un RSSI devrait également pouvoir ajouter, activer et paramétrer des règles de corrélation soit pour mettre à jour la politique de sécurité, soit pour le suivi d'une vulnérabilité, ou à des fins de débogage.

ÉTAT DE L'ART (Partie II)

Dans cette deuxième partie, on va donner un aperçu sur les différents types d'attaques qui existent, ainsi que sur quelques méthodes de détection, on va ensuite parler sur quelques solutions SIEM et faire une petite comparaison de ces fonctionnalités et on terminera par une discussion sur les difficultés rencontrées au début de la phase de conception.

1. Détection des attaques

Depuis leur création, les ordinateurs ont été victime de plusieurs attaques informatiques, depuis le premier virus en 1986 au Ransomware WannaCry de 2017, la sophistication et l'intensité des attaques ont augmenté au fil des années [4].

La figure suivante représente l'évolutivité des attaques depuis 1997 au 2017

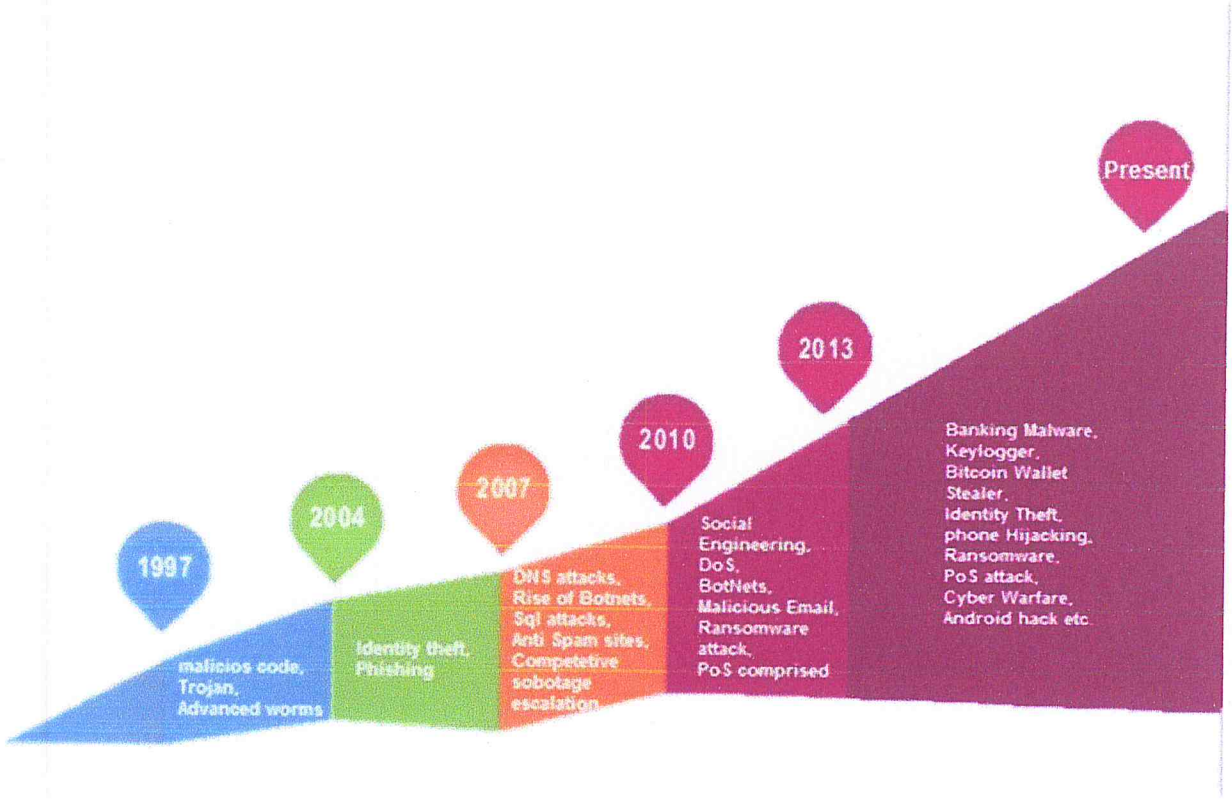


Figure 5: L'évolutivité des attaques depuis 1997 à 2017 « figure tirée de [4] »

Dans le chapitre suivant, nous allons définir c'est quoi une attaque informatique, les différentes types d'attaque et comment détecter une attaque.

1.1. Les attaques informatiques

Une attaque informatique (ou cyber-attaque) est une tentative d'accès non autorisé aux services, ressources, ou informations d'un système, ou une tentative visant à compromettre l'intégrité de celui-ci. En d'autres termes : c'est toute activité malveillante qui tente de collecter, perturber, nier, dégrader ou détruire les ressources d'un système d'information ou l'information qu'il contient [5].

1.2. Différentes types d'attaques

Les systèmes d'information sont des systèmes assez complexes, ils contiennent plusieurs composants et opèrent sur plusieurs niveaux d'abstractions.

Du matériel, au logiciel, au personnel, aux procédés de traitement de l'information, une cyber-attaque peut viser n'importe quel maillon de cette chaîne, pour peu qu'il existe une vulnérabilité exploitable, et chaque maillon de ce système peut être vecteur d'un tas d'attaques différentes, telles que :

1.2.1. Attaque Physique

Elles peuvent être d'origine criminelle, naturelle ou accidentelle. On peut citer notamment les désastres naturels, les pannes ou casses matérielles, le feu, le vandalisme ou les coupures électriques.

1.2.2. Attaque Réseaux

Sont des attaques qui se propagent par réseau, citons :

1.2.2.1. Attaque de l'homme du milieu (Man in the middle)

Est une attaque ou l'attaquant écoute secrètement la communication qui peut même être modifie par ce dernier sans que les deux communiquant ne se rendent en compte elles [6].

1.2.2.2. Usurpation d'adresse IP (IP spoofing)

Est la création des paquets IP avec une fausse adresse IP source, afin de cacher l'identité de l'expéditeur ou de se faire passer pour une autre personne ou un autre système informatique [6].

1.2.3. Attaque par exploitation de failles

Des attaques causées par des failles, ou des bogues, dans un logiciel ou un système que l'attaquant va pouvoir utiliser pour exécuter ses plans [6].

Il existe plusieurs attaques de ce type :

1.2.3.1. Dépassement de l'espace du tampon (Buffer Overflow)

Une attaque qui exploite un bogue par lequel un processus, lors de l'écriture dans un tampon (un espace mémoire utilisé pour sauvegarder des données temporairement pendant un transfert), écrit à l'extérieur de l'espace alloué à celui-ci, écrasant ainsi d'autres données, qui peuvent être cruciales pour le fonctionnement du système [6].

1.2.3.2. Escalade de privilège (Privilege Escalation)

Ça consiste à exploiter un bogue, un défaut de conception ou une erreur de configuration dans un système d'exploitation ou une application logicielle, pour obtenir un accès non légitime à des ressources normalement non accessibles. Le résultat est qu'une application avec plus de privilèges que prévu par le développeur d'applications ou l'administrateur système, peut effectuer des actions non autorisées [6].

1.2.4. Attaque par déni de service (Dos)

C'est une attaque ayant pour but de rendre indisponible un service, ou d'empêcher les utilisateurs légitimes d'un service de l'utiliser.

Quand ces attaques sont exécutées en utilisant plusieurs machines en même temps, on parle alors d'un déni de service distribué (DDOS) [6].

Il existe plusieurs type d'attaque Dos, nous allons citer quelques-unes :

1.2.4.1. Inondation Ping (ICMP flood)

L'une des formes les plus simple d'attaque par déni de service, où l'attaquant inonde le serveur cible de requêtes Ping (ICMP) [6].

1.2.4.2. SYN flood

L'une des attaques les plus répandues, dans laquelle l'auteur envoie une succession de requêtes SYN au système-cible afin de consommer suffisamment de ressources du serveur pour que le système devienne incapable de répondre au trafic légitime [6].

1.3. Reconnaissance et détection des attaques informatiques

Détecter une attaque une fois qu'elle a été exécuté est plus facile que de la détecté quand on est en train d'être attaqué.

Lorsqu'on subit une attaque de type DOS par exemple, on ne se rend compte qu'après un service est devenu hors ligne, similairement, lorsqu'on subit une attaque par force brute, on peut ne pas détecter un accès illégitime à un compte, qu'après l'attaquant commence faire des activités suspectes.

L'une des défis les plus difficiles auxquelles les entreprises sont confrontées est de savoir comment détecter une attaque le moment où elle survienne

Dans n'importe quel un système d'information, toute action que nous effectuons sur un ordinateur laisse une trace quelque part, et l'un des endroits pour trouver ces traces, est les fichiers journaux.

Chaque action prise, dans un réseau ou un logiciel est enregistrée dans des fichiers logs, que nous pouvons consulter.

1.3.1. Détection d'une attaque de par force brute (Brute-Force)

C'est une attaque qui consiste à essayer plusieurs fois un mot de passe ou phrase passe pour obtenir un accès à un compte ou un système quelconque.

Une particularité de ce type d'attaque c'est le grand nombre d'essai à faire pour obtenir un résultat.

La figure suivante représente une capture de log d'un serveur Apache

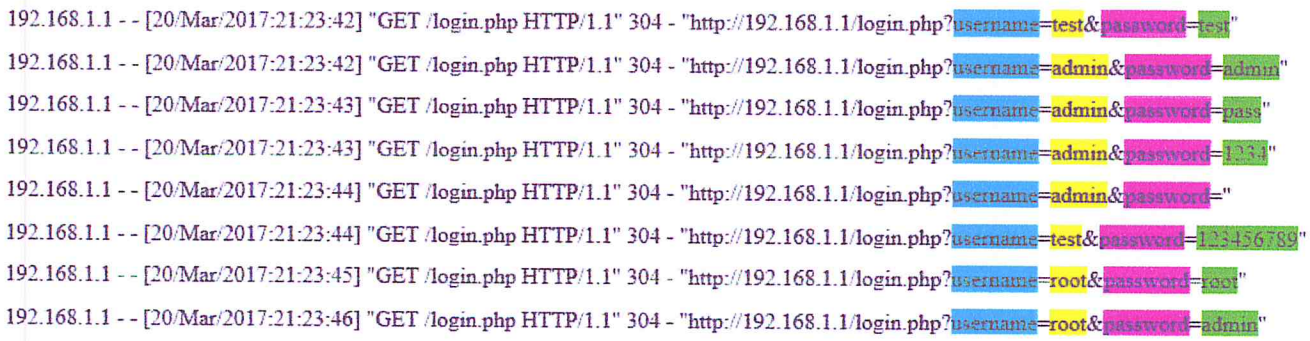


Figure 6: Prise d'écran d'une capture de log d'un serveur web

La figure montre qu'il y a eu de multiples tentatives pour accéder à la page « login.php » avec différents noms d'utilisateurs et différents mots de passe.

L'attaquant dans un espace de quatre seconde a fait huit tentatives de connexion, ce qui est considéré comme un comportement suspect pour un utilisateur typique, donc on peut en déduire que l'attaquant veut accéder à ce compte d'une manière illégale.

Dans ce cas, on peut détecter l'intrusion relativement facile.

1.4. Discussion

La détection d'attaques n'a pas une méthode bien définie à suivre, on a juste des lignes directrices nommées auparavant qui nous aideront à programmer un algorithme de détection, car dans la vie réelle une attaque est beaucoup plus sophistiquée où l'attaquant utilise plusieurs méthodes pour cacher son identité ou bien simulé un accès légitime à un compte ou un service, d'où la détection devient très difficile et les algorithmes montent en complexité.

2. Le marché des systèmes de sécurité à l'aide d'outils de gestion des évènements (SIEM)

Le besoin d'un marché efficace pour les solutions des systèmes de sécurité à l'aide d'outils de gestion des événements augmente constamment à cause de la propagation massive des cyber-attaques et des violations de la conformité qui affectent énormément les entreprises, mais le marché des SIEM avance et évolue rapidement [7].

Les fournisseurs des produit SIEM ne se concentrent pas seulement sur la présentation de leurs produits en tant que plates-formes offrant des analyses de sécurité et d'applications, mais aussi de se concentrer d'avantage sur l'expansion des déploiements technologiques dans des comptes clients existants et nouveaux [7].

MarketsandMarkets s'attend à ce que le marché total de l'information sur la sécurité et de la gestion des événements passe de 2,57 milliards de dollars en 2014 à 4,54 milliards de dollars en 2019 à raison de 12,0% au cours de la période de prévision [7].

Particulars	2014	2015	2016	2017	2018	2019	CAGR (2014-2019)
SIEM Solutions	xxx.x	xxx.x	xxx.x	xxx.x	xxx.x	xxx.x	xx.xx
SIEM Services	xxx.x	xxx.x	xxx.x	xxx.x	xxx.x	xxx.x	xx.xx
Global SIEM Market	xxx.x	xxx.x	xxx.x	xxx.x	xxx.x	xxx.x	xx.xx
Y-o-Y (%)		xx.xx%	xx.xx%	xx.xx%	xx.xx%	xx.xx%	

Figure 7: Taille du marché SIEM, 2014-2019 (en millions de dollars) « figure tirée de [7] »

Figure 8: Taille du marché SIEM et taux de croissance, 2014-2019 (en millions de dollars) « figure tirée de [7] »

2.1. Les différents SIEM disponibles sur le marché

Vu cette croissance rapide et continue des SIEM, beaucoup de solutions ont immergé dans le marché, nous allons citer quelqu'une :

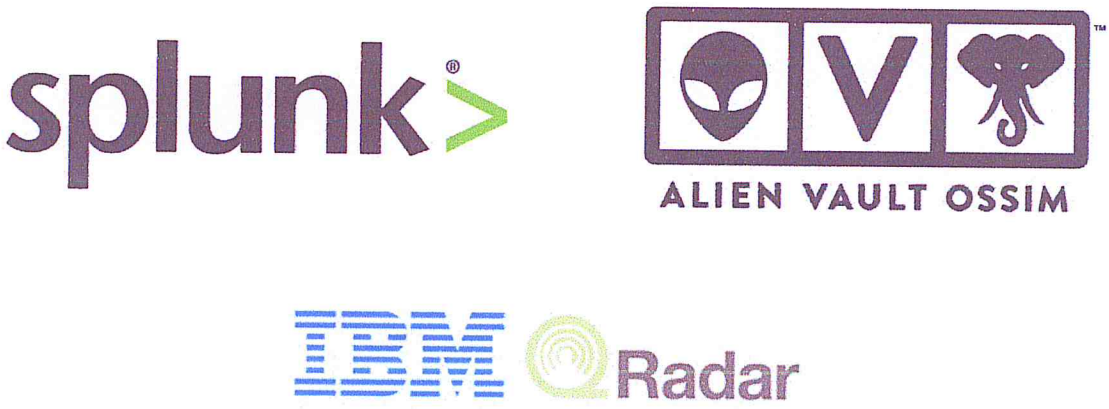


Figure 9: Logo des solutions présentées

2.1.1. Splunk Entreprise

Est une solution SIEM offerte par l'entreprise américaine Splunk avec toutes les fonctionnalités SIEM basiques qui peuvent être étendues à l'aide d'extension. C'est une solution payante mais qui offre une version d'essai limité de 60 jours [8].

2.1.2. AlienVault OSSIM

Est une solution SIEM open-source offerte par AlienVault basé sous forme serveur. La société offre aussi une version commerciale à 5000 $ par an et qui est plus robuste, plus performante et la version recommandée pour les grandes entreprises [8].

2.1.3. IBM Security QRadar

Est une solution proposé par IBM, basée sur le principe de composant ; elle peut être déployé comme un matériel, logiciel ou produit virtuel, elle propose une version d'essai et une version commercial trop chère, qui calcule le montant à base d'utilisation par minute, heure ou jour [8].

2.2. Comparaison des solutions disponibles sur le marché

Chaque solution SIEM disponible sur le marché diffère d'une autre sur plusieurs plans, car chaque fabricant vise une catégorie bien spécifique du marché.

Nous allons présenter des comparaisons des différentes SIEM disponible sur le marché sous des plans différents :

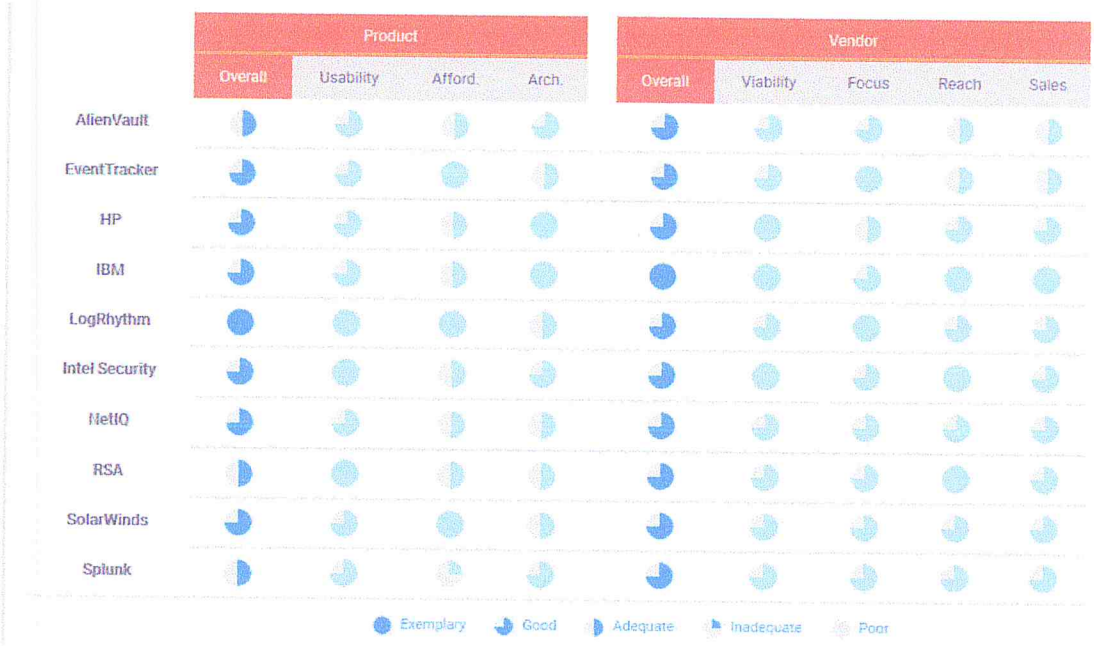


Figure 10: Comparaison de plusieurs SIEM pour une petite entreprise de communication « figure tirée de [9] »

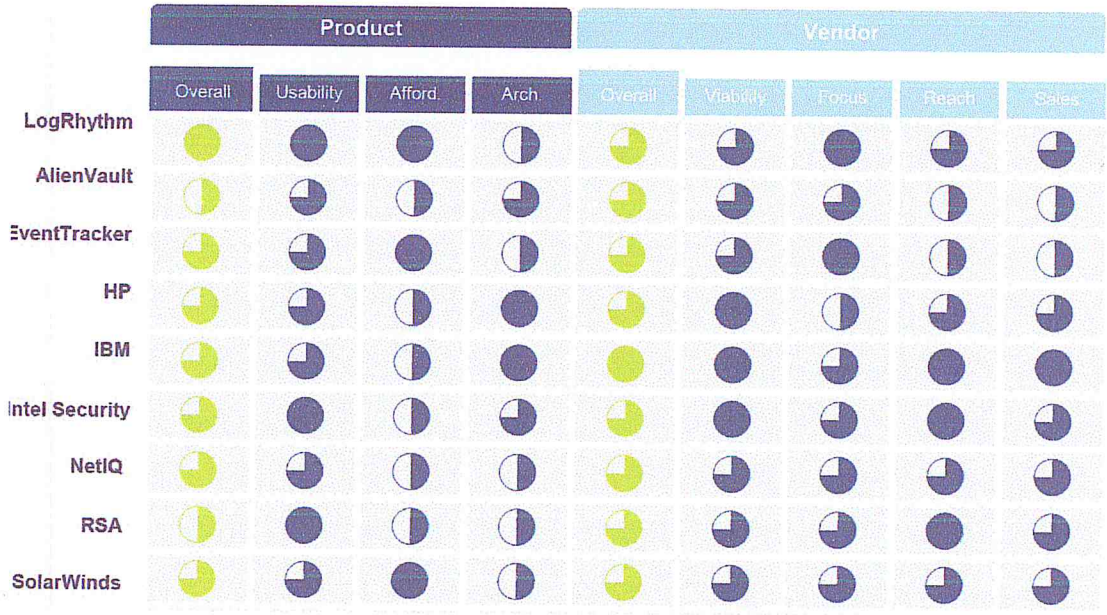


Figure 11: Comparaison de plusieurs SIEM pour une grande entreprise de technologie « figure tirée de [9] »

2.3. Discussion

Un des problèmes du développement d'une solution SIEMs est la disponibilité du code source, la plupart des solutions SIEMs sont payante, et par conséquent leur code est propriétaire et closed-source, du coup on ne peut pas les étudier pour comprendre ou pour s'inspirer, ils sont comme des boites noires qu'on peut utiliser sans jamais comprendre le fonctionnement.

Quelques solution open-source existent, mais même ceux-ci sont mal-documentés, et détaillent rarement bien leur fonctionnement.

Tout cela peut laisser quelqu'un qui veut développer une solution SIEM dans un vrai dilemme ; entre le manque de références et la difficulté de la tâche, il n'a pas de point d'appui pour commencer.

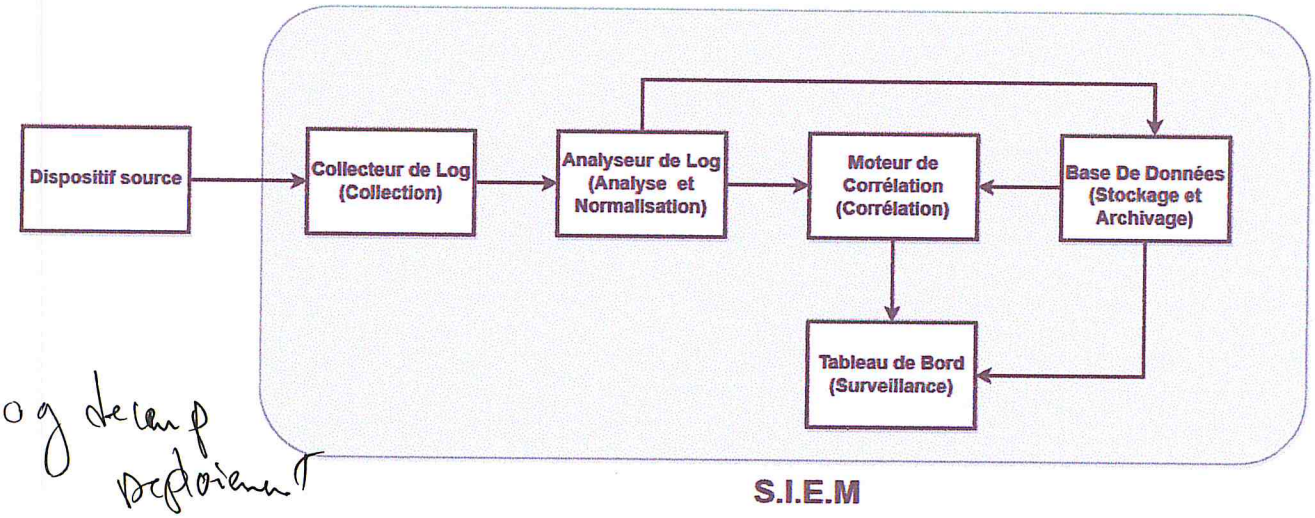
CONCEPTION DE L'APPLICATION

Après avoir défini le cadre de notre application, nous nous orientons à travers le présent chapitre, vers la conception de notre système.

La modélisation que nous avons adoptée, assure que le système restera dynamique et indépendant des outils utilisés lors de l'implémentation et que chaque module soit fonctionnellement indépendant des autres.

Nous allons commencer par détailler l'architecture de notre solution :

1. Schéma générale



Diag de dep deploiement

Figure 12: Architecture globale du SIEM

La figure ci-dessus représente le schéma global ou bien l'architecture globale que nous avons conçue pour la réalisation de notre SIEM. Chaque composant sera détaillé maintenant :

1.1. Les dispositifs Sources

Ce sont des systèmes qui génèrent des fichiers journaux (logs), ils peuvent être des systèmes matériels, des serveurs, des pare-feu, logiciels, des systèmes d'exploitation, des antivirus… etc.

1.2. Collection des logs

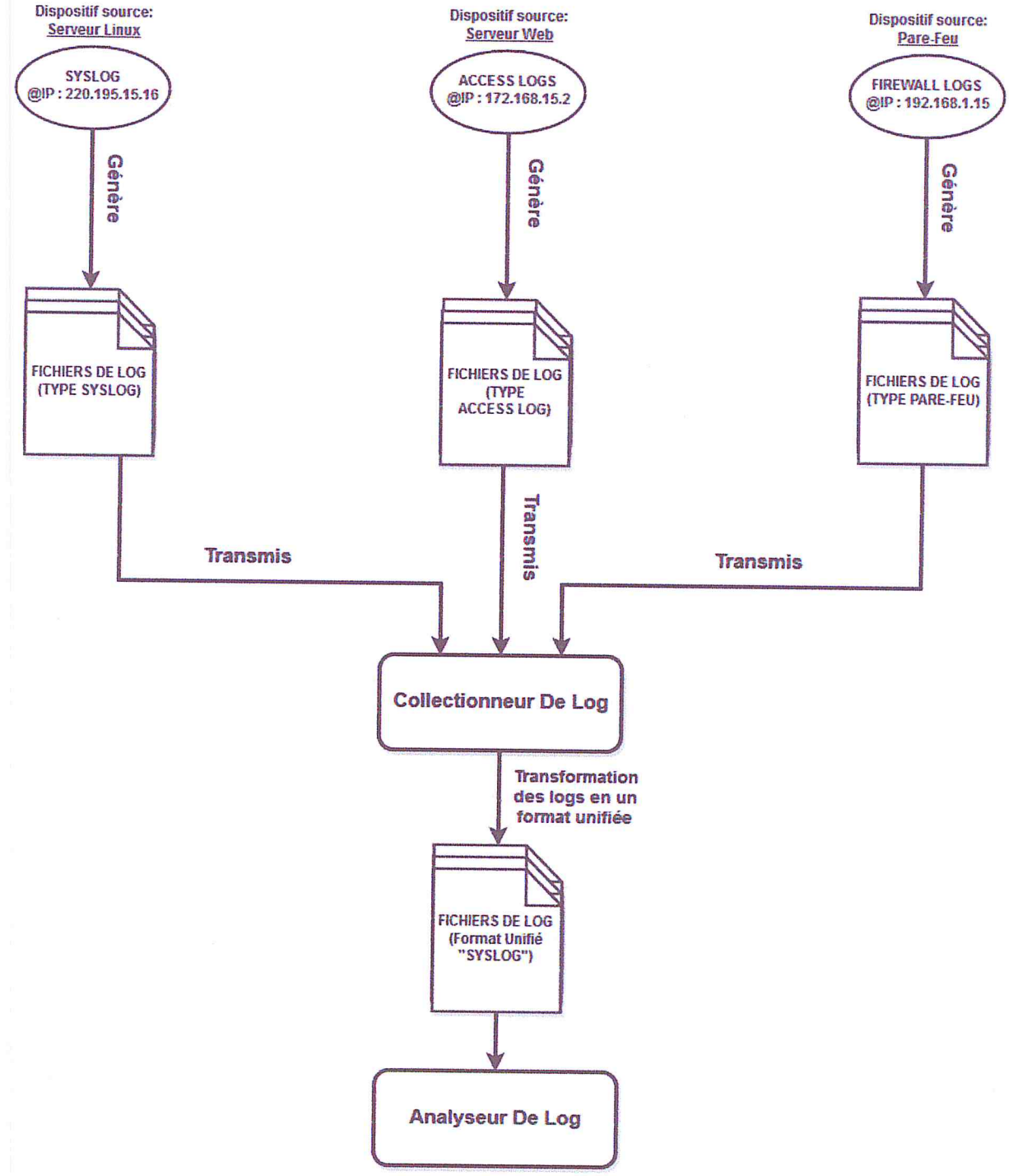


Figure 13: Collection des logs

La figure ci-dessus montre la procédure de collection des logs depuis des différents dispositifs (Serveur Linux, Serveur Web, Pare feu).

Les différents dispositifs envoient une copie non normalisée de leurs fichiers journaux (log) en se connectant avec un serveur central (Receveur ou collecteur des logs).

Lors de l'implémentation du collecteur, on a mis l'accent sur l'optimisation de la latence entre le serveur de collection et le serveur d'analyse, avec comme but de rendre la durée du transfert quasi-négligeable.

Pour ce faire, a besoin d'un mécanisme de réduction de nombre de logs. Et vu qu'un système génère des logs pour toute action prise sur celui-ci, il est évident que chaque type des logs n'est pas être toujours pertinent pour la sécurité

Donc, on a appliqué des filtres pour les logs reçus depuis les dispositifs sources pour n'envoyer au serveur d'analyse que les logs nécessaires pour la détection des attaques.

1.3. Analyse et normalisation

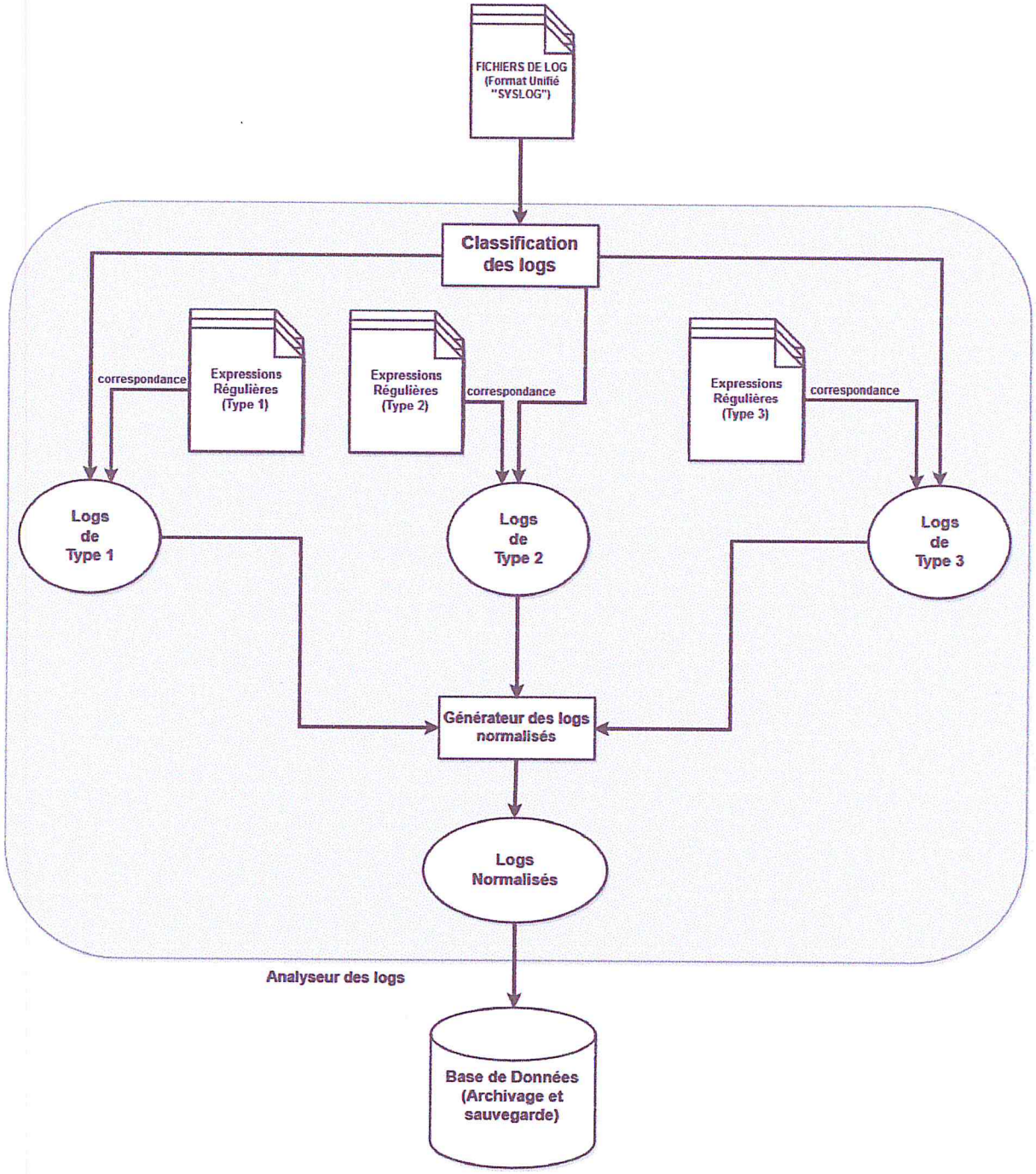


Figure 14: Analyse et normalisation des logs

La figure ci-dessus montre la procédure d'analyse et la normalisation des logs reçus depuis le collecteur des logs.

Le serveur de collection (Collecteur des logs) va rediriger les logs reçus vers un autre serveur (Analyseur des logs) qui va faire le traitement et la normalisation de ces logs comme suit :

1.3.1. Classification des logs

Le system analyse les logs reçus et les classifie selon leur type, pour ensuite correspondre chaque log avec son fichier de configuration approprié.

1.3.2. Correspondance entre log et expression régulière

L'analyseur des logs utilise un système de fichiers de configurations, ou chaque fichier de configuration contient un nombre d'expressions régulières qui vont permettre d'extraire les données présentes dans un log, et de les organiser en champs bien déterminés.

Les fichiers de configuration ont été conçus pour être dynamiques, c'est-à-dire même si un log non reconnu par le système a été envoyé depuis le serveur de collection, on peut toujours ajouter un fichier de configuration qui contient les expressions régulières nécessaires pour l'analyser.

1.3.3. Génération des logs normalisés

Après avoir fait le découpage des champs, le système va par la suite générer un log normalisé pour chaque log envoyé, qui contiendra des champs spécifiques et bien définis, que le moteur de corrélation va utiliser par la suite pour faire son analyse.

La figure suivante représente un exemple de normalisation :

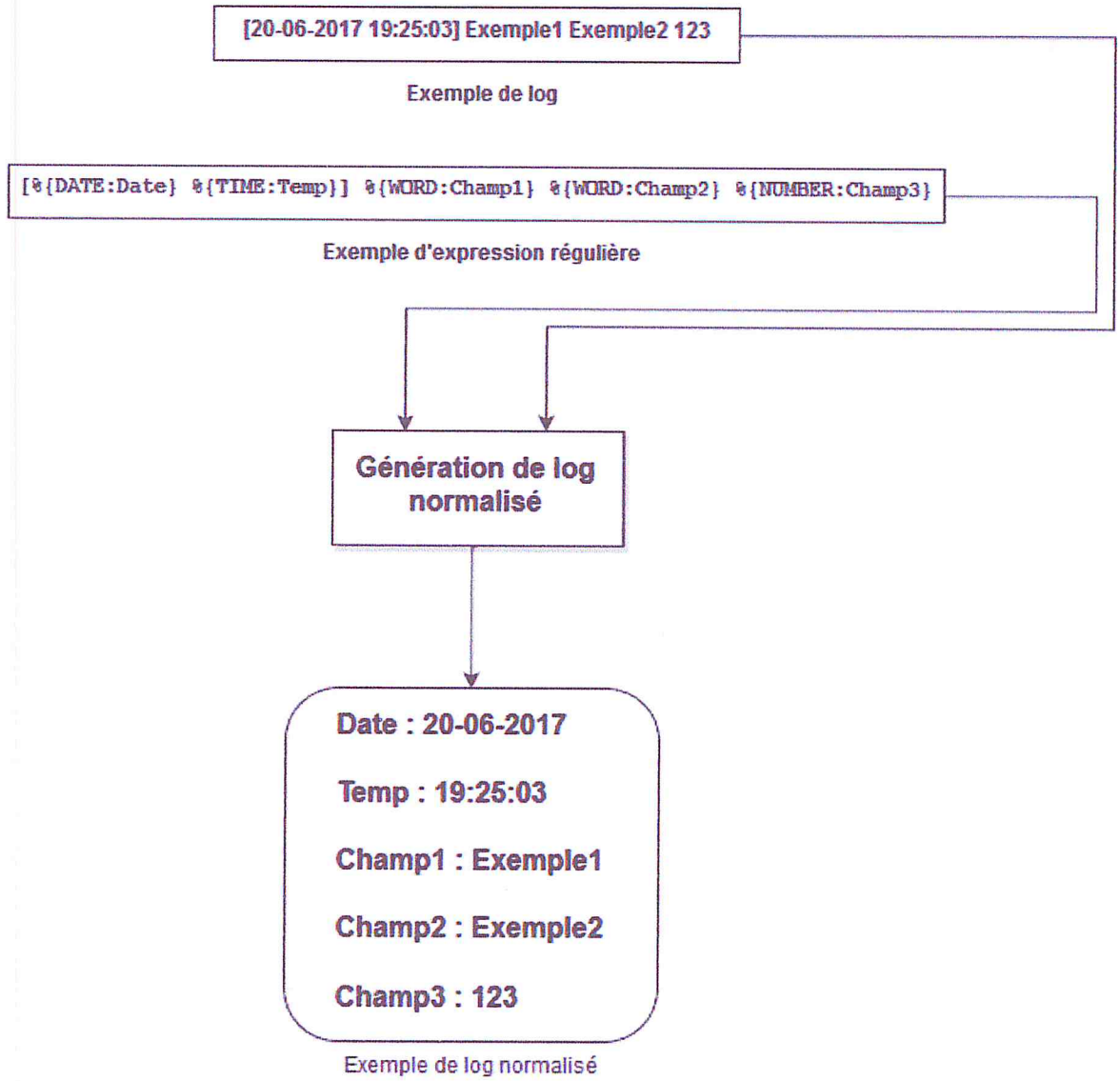


Figure 15: Exemple de normalisation d'un log

Une fois les logs normalisés sont générés, ils seront envoyés vers une base de données, pour un futur traitement.

1.4. Moteur de corrélation

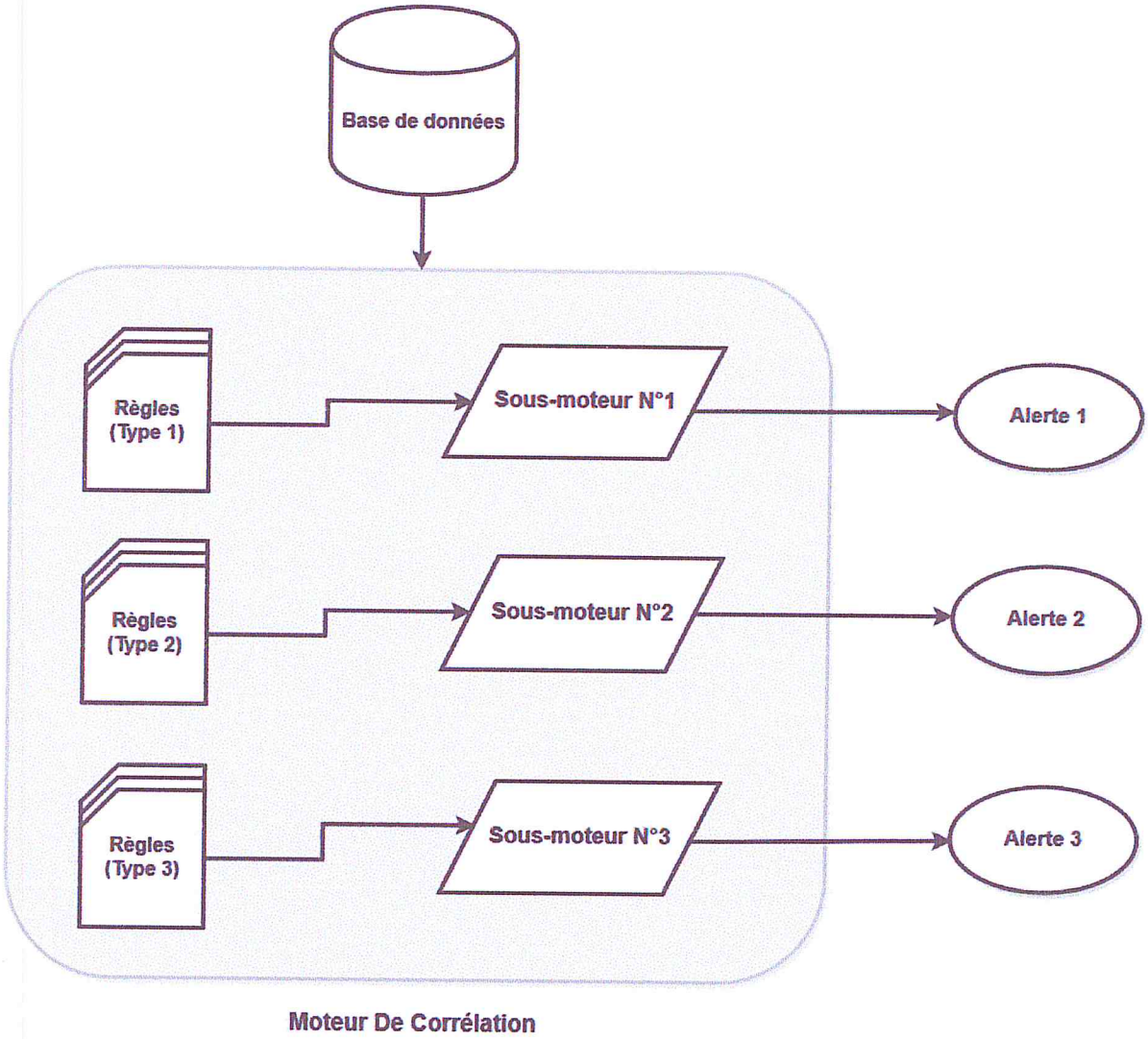


Figure 16: Moteur de corrélation

Le moteur de corrélation va permettre la détection des attaques en temps réel et cette fonctionnalité est réalisée à l'aide de plusieurs composants :

1.4.1. Règles de corrélation

La méthode de détection peut varier de complexité ; par exemple : pour une règle qui compte combien de fois un utilisateur a accédé à une certaine page, ajouter une contrainte de temps peut faire la différence entre un algorithme simple et un algorithme compliqué.

De plus, il n'y a pas de modèle standard pour la création d'une règle, pour créer une, il faut développer son propre algorithme de détection, qui peut varier selon le contexte, le besoin, et les intentions du développeur.

Donc vu les difficultés liées à la conception d'une règle de détection, nous n'avons pris en compte que deux type de règles, à savoir ceux de :

- La détection d'une attaque DOS à base des logs pare-feu.
- La détection d'une attaque Force-Brute à base des logs d'un serveur-web (Apache)

1.4.1.1. Conception des règles de corrélation

Lors de la conception des règles de corrélation deux contraintes ont été prises en compte :

- **La terminaison :** Un algorithme doit toujours retourner une réponse dans un temps fini.
- **Les ressources consommées :** Un algorithme ne doit pas consommer plus de ressources qu'il en a besoin pour trouver une réponse.

Algorithm 1 Règle générique de détection d'une attaque

```
1: procedure ExempleDétectionAttaque(Data)
2:     compteur : entier;
3:     compteur = 0;
4:     for each element i in Base de données do
5:         if i == Data then
6:             compteur = compteur + 1;
7:         end if
8:     end for
9:     if compteur >= seuil then
10:        Return "Alert d'une attaque";
11:    else
12:        Return 0;
13:    end if
14: end procedure
```

Figure 17: Exemple d'une règle de corrélation générale

La figure ci-dessus représente le pseudocode de l'algorithme de détection d'une attaque générique.

Il décrit l'approche utilisé pour la détection des attaques :

- La fonction de détection prend des données en entrée sous la forme de « Data » qui représente les informations nécessaires à la détection.
- Ensuite, un compteur est créé pour sauvegarder le nombre de tentatives détectés.
- Une comparaison suit cette étape, entre la donnée d'entrée et chaque élément de la base de données, et s'il y a une égalité, le compteur est incrémenté.
- A la fin, on fait un test pour déterminer si le compteur a atteint un certain seuil, et si c'est le cas, on déclenche une alerte.

Ce qu'on vient de voir est une règle de corrélation générale, pour mieux comprendre on va présenter dans ce qui suit deux règles particulières.

1.4.1.2. La conception d'une règle de détection d'une attaque DOS à base des logs pare-feu

La figure suivante représente un organigramme détaillant le fonctionnement de la règle.

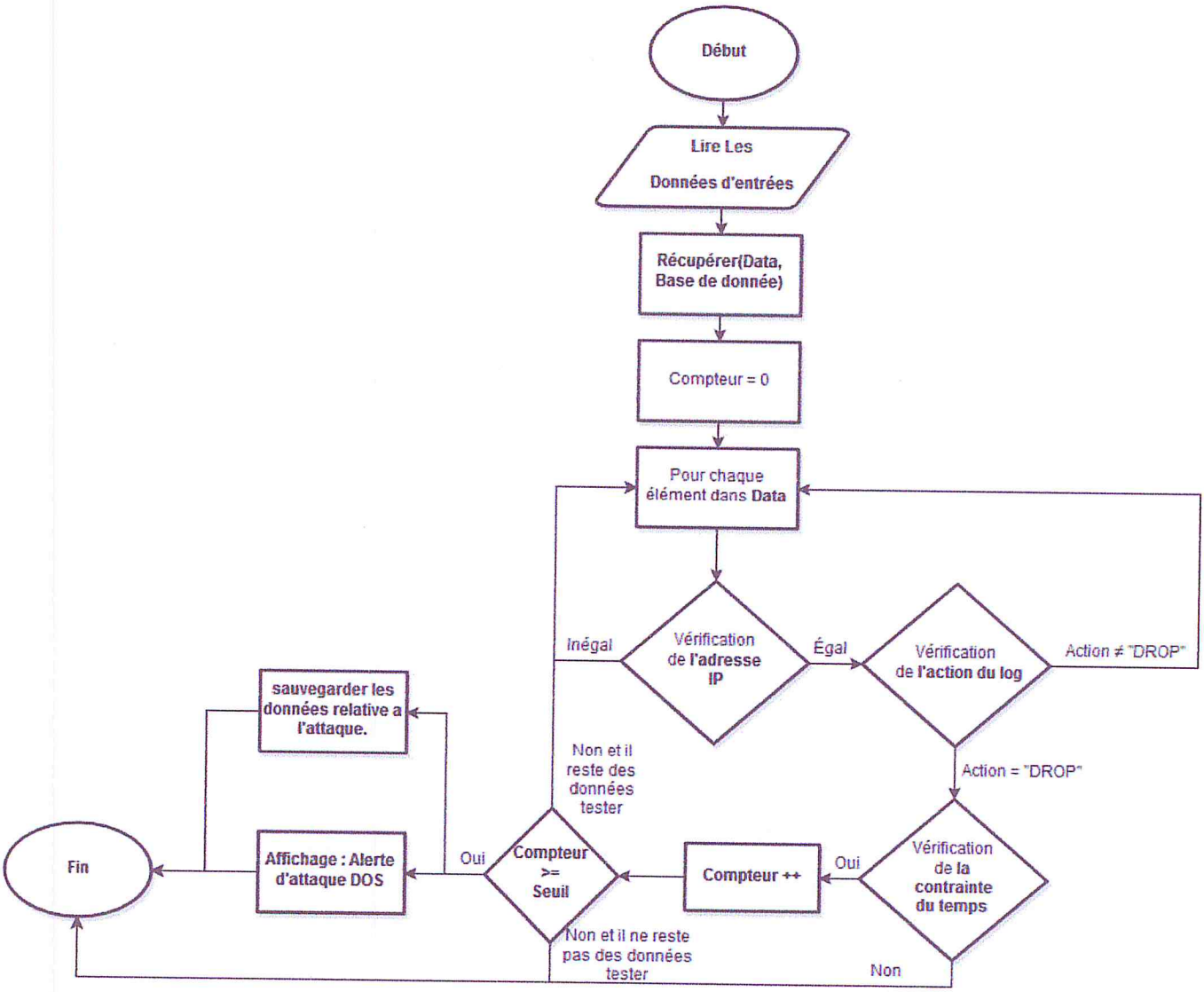


Figure 18: Organigramme d'une règle de détection d'une attaque DOS

1.4.1.3. La conception d'une règle de détection d'une attaque Force-Brute à base des logs web (Apache)

La figure suivante représente un organigramme détaillant le fonctionnement de la règle

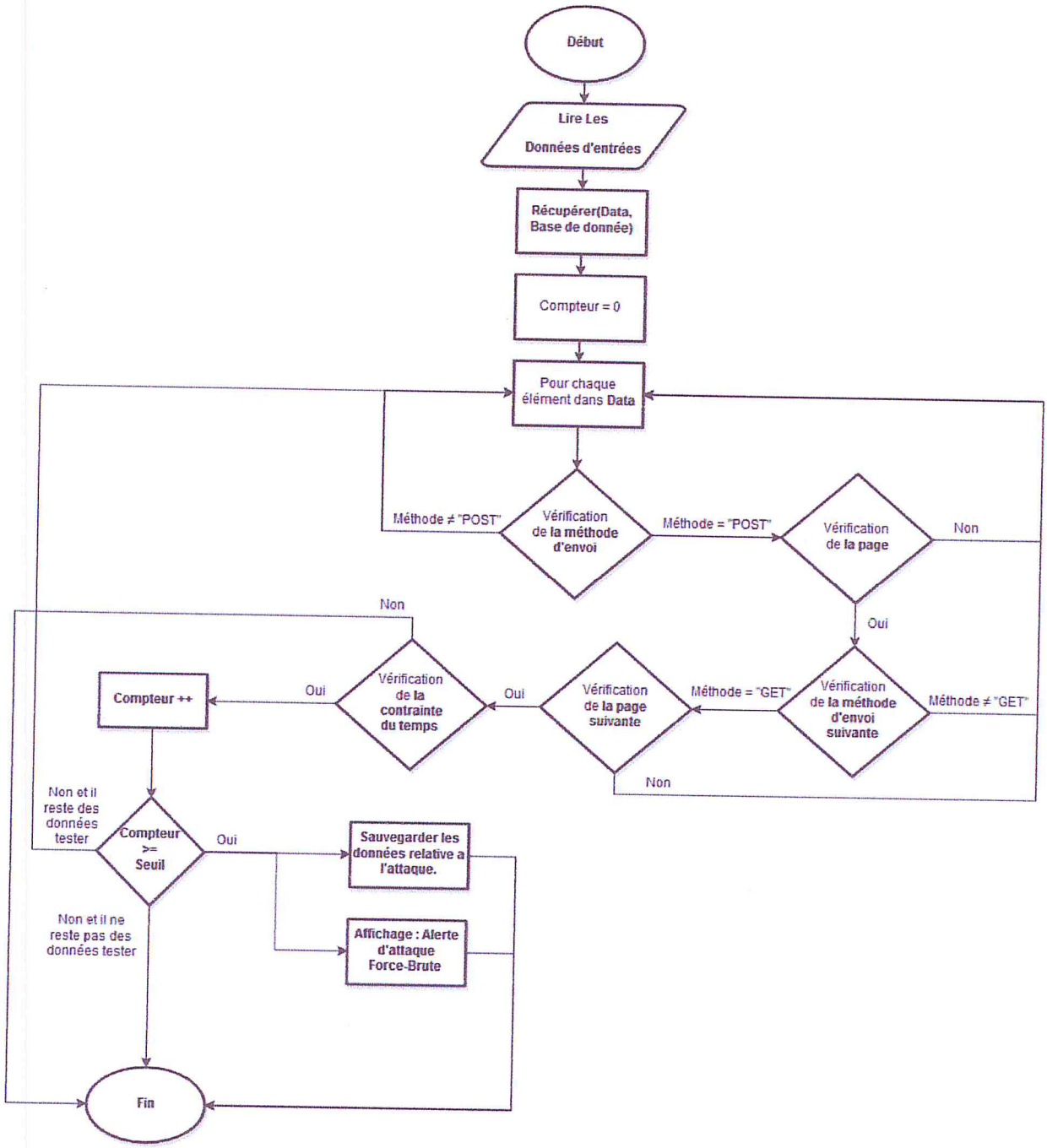


Figure 19: Organigramme d'une règle de détection d'une attaque force-brute

1.4.2. Sous-moteur de corrélation :

Les règles sont classifiées par catégorie en fonction du type de l'attaque à détecter (attaque DOS, attaque force-brute ... etc.).

Chaque sous moteur exécute une catégorie de règles permettant ainsi la détection d'un seul type d'attaque.

Pourquoi le découpage en sous-moteur ?

- Chaque sous-moteur est indépendant des autres et s'exécute en parallèle, cela est fait pour assurer que le cœur du SIEM (le moteur de corrélation) est toujours fonctionnel même si un de ses sous-moteurs éprouve une erreur d'exécution ou s'arrête de fonctionner complètement.

- Si une règle n'a pas abouti à un résultat dans un temps fini, au lieu de redémarrer tout le moteur (ce qui causera une perte de temps, sans évoquer le risque qu'une attaque arrive pendant ce temps-là), on redémarrera juste la partie qui est a causé l'erreur.

1.4.3. Moteur de corrélation

Le dernier composant c'est le moteur de corrélation lui-même, qui va s'occuper des tâches suivantes :

- La collection des données depuis la base des données.
- Activer / Désactiver les sous-moteur de corrélation.
- Suivre l'état de chaque sous-moteur et le réactiver dans le cas échéant.

Tout résultat obtenu par le moteur de corrélation va être envoyé en temps réel sous forme d'une alerte indiquant le type d'attaque.

1.5. Archivage et sauvegarde des données

Les logs sauvegardés vont être envoyés depuis le serveur de normalisation.

Lors de la sauvegarde des données le SIEM prend en considération deux types de données :

- **Les logs normalisés (Corrélation et recherche)**

  Qui vont être sauvegardés dans une base de données non relationnelle pour faciliter et augmenter la vitesse de recherche, et qui vont servir comme des données d'entrée pour le moteur de corrélation.

- **Les logs non normalisés (Archivage)**

  Qui vont eux aussi être sauvegardés dans une base non relationnelle (NOSQL), sauf que ceux-ci vont être utilisés pour des recherches légales au cas d'une investigation.

1.5.1. Choix de la base de données

Compte tenu de la quantité et le type de données que nous traitons dans ce projet, une base de données relationnelle a été un premier choix à considérer, mais a été rapidement rejetée pour sa limitation vis-à-vis l'hétérogénéité des types des données et leurs tailles importantes (Big Data) [10].

1.5.2. Base de données relationnelle(SQL) VS Base de données non relationnelle (NoSQL)

Il existe des différences dans de nombreux aspects entre les bases de données relationnelles et non relationnelles qui sont les suivantes :

- **Types de données**

Les données incluent maintenant des types de données riches : tweets, vidéos, podcasts, gifs animés, qui sont difficiles, voire impossible, à stocker dans une base de données relationnelle, Les base de données non-relationnelle d'autre part, peuvent intégrer tous types des données, tout en fournissant toutes les fonctionnalités nécessaires pour créer des applications riches en contenu [10].

- **Évolutivité**

La mise à l'échelle d'une base de données relationnelle n'est pas triviale et trop cher. D'autres part, Les Base de données NoSQL sont conçues pour prendre en charge le stockage te le traitement une quantité importante de données [10].

- **Vitesse**

Les bases de données relationnelles nécessitent un degré plus élevé de normalisation, c'est-à-dire que les données doivent être divisées en plusieurs petites tables logiques pour éviter la redondance, mais la complexité d'étendre plusieurs tables liées impliquées dans la normalisation entrave les performances du traitement de données dans des bases de données relationnelles [10].

D'autre part, dans les bases de données NoSQL, les données sont stockées sous la forme de collections plates où ces données sont dupliquées à plusieurs reprises et une seule donnée est rarement partitionnée, mais elle est conservée sous la forme d'une entité. Par conséquent, les opérations de lecture ou d'écriture à une seule entité sont devenues plus faciles et plus rapides [10].

Les bases de données NoSQL peuvent également stocker et traiter des données en temps réel quelque chose que SQL n'est pas capable de le faire.

Et comme l'objectif de notre projet est de proposer une solution fiable et compétitive (sur le marché Open-source), notre choix s'est orienté vers les bases de données non relationnelles libres pour leur bonne réputation dans la mise à l'echelle, et l'optimisation pour le Big Data .

1.6. Tableau de bord (La communication de données et Surveillance)

Ce tableau de bord va contenir toutes les informations nécessaires pour gérer toutes les fonctionnalités présentes partant de la création des règles, corrélation des logs, recherche des logs, jusqu'à l'affichage des notifications en temps réels.

Conclusion

Ce chapitre a présenté comment nous avons conçu l'architecture conceptuelle de notre application en tenant en compte la collection, la normalisation, la corrélation. Nous avons aussi détaillé les considérations et les choix conceptuels qu'on a pris, tel que les contraintes de choix de la base de données et le découpage en sous-moteurs de corrélation. On va découvrir à travers le prochain chapitre, comment nous avons implémenté cette conception.

MISE EN ŒUVRE

Ce chapitre présentera les outils utilisés pour le développement de notre application leur fonctionnement et la manière dont ils ont été implémentés.

1. Présentation des outils

Pour le développement de cette application nous avons utilisé plusieurs outils, que nous allons présenter :

1.1. La pile Elastique (Elastic Stack)

Est un groupe de produits open source d'Elastic conçu pour prendre des données à partir de n'importe quel type de source et dans n'importe quel format et rechercher, analyser et visualiser ces données en temps réel. (Voir figure 20)

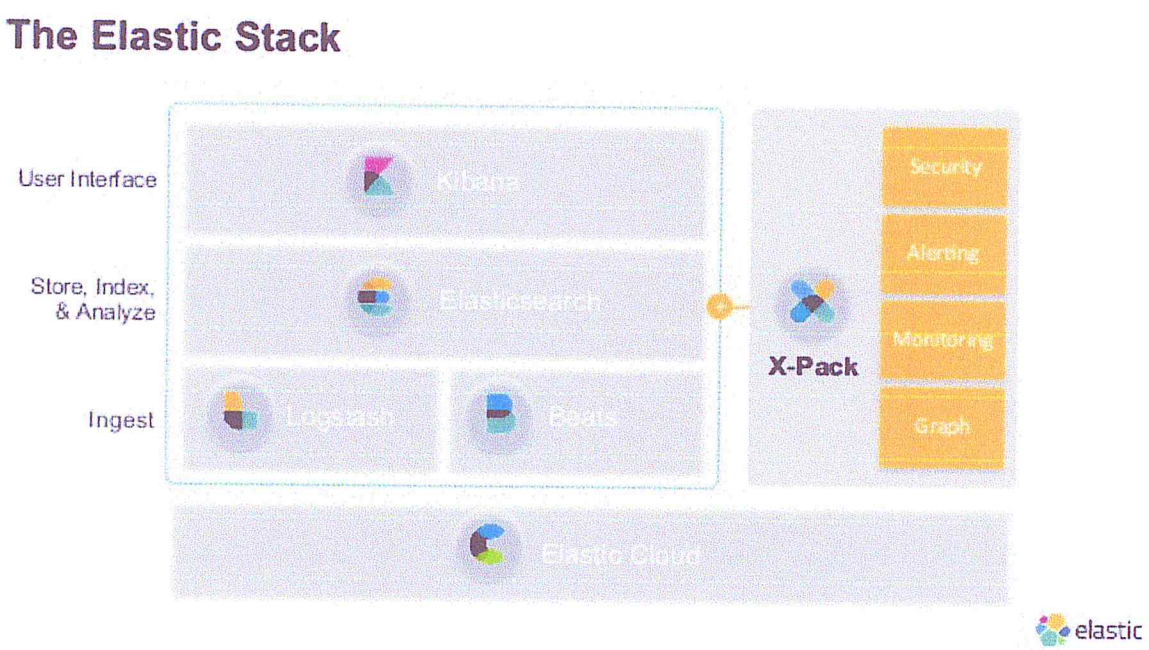


Figure 20: Les différents produits Elastic

On va s'intéresser à deux de ces solutions : Logstash et Elasticsearch, que nous allons introduire ci-dessous :

1.1.1. Logstash

Est un moteur de collecte de données open source avec des capacités de pipeline en temps réel. Logstash peut unifier dynamiquement les données provenant de différentes sources et normaliser les données vers n'importe quel format de sortie [11].

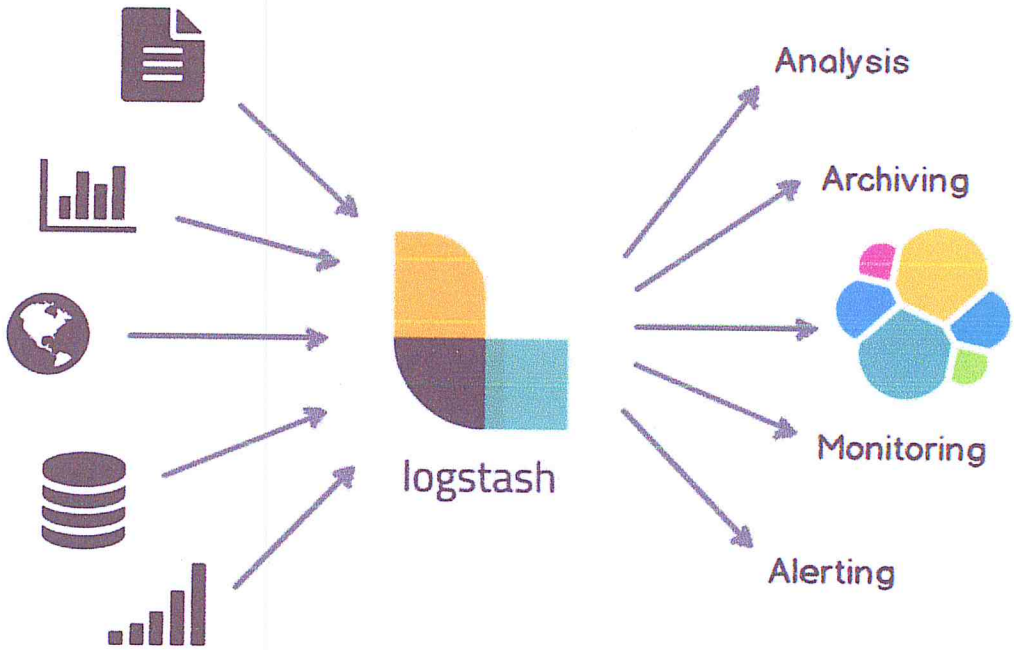


Figure 21: Principe de Logstash « figure tirée de [11] »

1.1.2. Elasticsearch

Est un moteur de recherche et d'analyse à texte intégral hautement évolutif. Il permet de stocker, de rechercher et d'analyser de gros volumes de données rapidement et en temps quasi réel. Il est généralement utilisé comme moteur / technologie sous-jacent qui permet aux applications qui ont des fonctionnalités de recherche complexes et des exigences. Élément clé de la suite Elastic, il stocke de manière centralisée les données et permet d'être préparé en toutes circonstances, il permet d'effectuer et de combiner des recherches variées sur des données structurées, non-structurées, de géolocalisation ou indicateurs [12].

Il est basé sur certain concept que nous allons introduire :

- **Temps quasi réel (Near Real Time)**

Cela signifie, qu'il y a une légère latence (normalement une seconde) à partir du moment où on indexe un document jusqu'au moment où il devient consultable [12].

- **Cluster**

Un cluster est une collection d'un ou plusieurs nœuds (serveurs) qui, ensemble, détiennent toutes les données et fournissent des fonctions d'indexation et de recherche fédérées sur tous les nœuds [12].

- **Nœud**

Un nœud est un serveur unique qui fait partie du cluster, stocke les données et participe aux fonctionnalités d'indexation et de recherche du cluster [12].

- **Index**

Un index est une collection de documents ayant des caractéristiques un peu semblables [12].

1.2. Nxlog

En concept, NXLog est similaire à syslog-ng ou rsyslog, qui sont tous les deux des collecteurs de logs mais il n'est pas limité à Unix et à Syslog uniquement. Il prend en charge différentes plates-formes, log sources et formats [13].

Il peut collecter des journaux à partir de fichiers dans différents formats, recevoir des journaux du réseau à distance sur UDP, TCP ou TLS / SSL sur toutes les plates-formes prises en charge. Il prend en charge les sources spécifiques à la plate-forme telles que le journal des événements Windows, les journaux du noyau Linux, les journaux des périphériques Android, le Syslog local, etc. L'écriture et la lecture des journaux dans les bases de données sont également prises en charge pour de nombreux serveurs de base de données. Les journaux collectés peuvent être stockés dans des fichiers, des bases de données ou transmis à un serveur de journaux distants en utilisant différents protocoles [13].

Quelques fonctionnalités :

- Open source.
- Multi-plate-forme.
- Architecture modulaire grâce à des plugins dynamiquement chargeables.
- E / S évolutive et performante.
- Pas de messages perdus ou abandonnés.
- Tâches programmées et rotation logarithmique intégrée.
- Prise en charge de différents formats tels que Syslog, CSV, GELF, JSON, XML, EventLog de Windows et même des formats personnalisés.
- Transport réseau sécurisé sur SSL.

1.3. PyCharm

C'est un environnement de développement intégré (EDI), spécifiquement pour le langage Python. Il fournit une analyse de code, un débogueur graphique, un testeur d'unité intégré, une intégration avec des systèmes de contrôle de version (VCSes) et prend en charge le développement Web avec Django [14].

1.4. JAVA

Java est un ensemble de logiciels et de spécifications. Créer le 17 novembre 2006, développés par Sun Microsystems, qui a ensuite été acquis par Oracle Corporation, qui fournit un système pour développer des logiciels d'application et le déployer dans un environnement informatique multiplateforme.

1.5. Kali Linux

C'est un système d'exploitation de la distribution GNU/Linux, il fournit une solution regroupant l'ensemble des outils nécessaires aux tests de sécurité d'un système d'information, notamment le test d'intrusion [15].

1.6. XAMPP

XAMPP est un ensemble de logiciels permettant de mettre en place facilement un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Créer le 22 mai 2002, il s'agit d'une distribution de logiciels libres (X (cross) Apache MariaDB Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la

portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus. [16].

2. Présentation des langages

Afin de réaliser notre système, nous avons utilisé plusieurs langages qui nous ont permis de mettre au point un système dynamique et interactif, ces langages sont :

2.1. Python

Python est un langage de programmation objet, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions [17].

2.2. AJAX (Asynchronous JavaScript and XML)

L'architecture informatique Ajax (acronyme d'Asynchronous Javascript and Xml) permet de construire des applications Web et des sites web dynamiques interactifs sur le poste client en se servant de différentes technologies ajoutées aux navigateurs web entre 1995 et 2005, il communique vers et à partir d'un serveur / base de données sans avoir besoin d'un retour ou d'un rafraîchissement complet de la page [18].

2.3. CSS3 (Cascading Style Sheets)

Est un langage de feuille de style utilisé pour décrire la présentation d'un document écrit en HTML ou XML, il décrit comment les éléments doivent être rendus à l'écran, sur papier, dans la parole ou sur d'autres supports [19].

2.4. HTML5 (HyperText Mark-up Language)

HTML 5 est une révision de Hypertext Markup Language (HTML), le langage de programmation standard pour décrire le contenu et 'l'apparence' des pages Web [20].

Il a été conçue pour prendre en compte la nouvelle version du Web, il contient beaucoup d'amélioration et de nouveauté par rapport à la précédente version notamment un ensemble de nouveaux éléments de formulaire qui devraient faciliter considérablement le développement d'applications web.

2.5. JavaScript

JavaScript est un langage de programmation Web qui permet la création et la mise à jour dynamique du contenu, l'animation des images et de contrôler la multimédia. Il peut être mis en œuvre dans toute application disposant d'un interpréteur pour ce langage [21].

2.6. Bootstrap

Bootstrap est un Framework de type "Front-End Framework", c'est une compilation de plusieurs éléments et fonctions Webdesign personnalisables, ces éléments sont une combinaison de HTML, CSS et JavaScript [22].

2.7. Choix du Framework (Django)

Un Framework n'est pas indispensable pour la création de notre système mais un outil d'un grand aide pour développer une application qui respecte pleinement les règles métier, qui est structurée et qui est à la fois maintenable et évolutive [23].

Vu que nous avons choisie python pour le développement de nos règles de corrélation, le Framework Django est le choix le- plus logique.

Django est un Framework Web gratuit et open source, écrit en Python, qui suit le modèle d'architecture Modèle-Vue-Template (MVT) qui est dérivé du modèle d'architecture MVC [24].

3. Installation et configuration des outils

L'installation des outils se fait après avoir télécharger les différents logiciels sur leur site officiel:

- https://www.elastic.co/guide/en/elasticsearch/reference/current/_installation.html
- https://www.elastic.co/guide/en/logstash/current/installing-logstash.html
- https://nxlog.co/products/nxlog-community-edition/download
- https://www.python.org/downloads/windows/
- https://www.jetbrains.com/pycharm/download/#section=windows
- https://www.djangoproject.com/download/

Après avoir téléchargé ces logiciels, l'installation et la configuration initiale peut se faire aisément en suivant les tutoriaux disponibles de chacun d'eux.

3.1. Configuration de Nxlog

Quand l'installation est terminée on va ensuite passer à la configuration de Nxlog qui consistera à saisir le numéro de port d'écoute du serveur de traitement et son adresse IP.

Les logs venant des différentes sources seront saisis par l'utilisateur de l'application et seront considérée comme des entrées de l'application.

3.2. Configuration de Logstash

Pour la configuration de Logstash on a besoin de créer deux fichiers initiaux qui vont déterminer d'où viennent les logs et où les envoyer après avoir fait le traitement et la normalisation.

Les fichiers configuration qui vont traiter les logs sont considérés comme des entrées de l'application et donc saisie par l'utilisateur.

3.3. Configuration d'Elasticsearch

Pour la configuration d'Elasticsearch, il faut déterminer le nombre de nœuds nécessaires et à quel cluster ils appartiennent.

4. Implémentation

Dans cette partie nous allons détailler comment on a utilisé les outils pour réaliser notre SIEM.

4.1. Collection des logs (Nxlog)

Nxlog utilise un fichier de configuration pour savoir quels logs à collecter, ainsi que comment le faire (sources et modules utilisés).

Un fichier de configuration contient les informations relatives aux sources de logs ; à savoir l'adresse IP de l'hôte, le port, le protocole utilisé, le module ...etc.

Un périphérique duquel on veut collecter les logs est configuré préalablement avec Nxlog pour envoyer ses logs vers une machine de collection.

Nxlog lit alors sa configuration, reçoit les logs depuis la source en utilisant son adresse IP et son port, applique une règle de traitement (s'il y en a une), puis redirige vers une autre partie du système ou vers une autre machine dans le réseau (dans notre cas, vers Logstash)

Une fois les logs sont collectés, ils sont redirigés vers la prochaine étape qui est la normalisation (Logstash).

Note : cela est possible car Nxlog peut jouer le rôle d'un serveur (réception) ou d'un client (envoie) selon la configuration.

```
<Input firewall>
Module im_file
File "/root/firewall.log"
SavePos FALSE
Recursive FALSE
</Input>

<Input AccessLog_lampp>
Module im_file
File "/opt/lampp/logs/access_log"
SavePos FALSE
Recursive FALSE
</Input>

<Route 1>
Path firewall, AccessLog_lampp => logstash
</Route>
```

Figure 22: Prise d'écran du fichier de configuration Nxlog (1)

```
User nxlog
Group nxlog

LogFile /var/log/nxlog/nxlog.log
LogLevel INFO

<Extension _json>
Module  xm_json
</Extension>

<Extension syslog>
Module  xm_syslog
</Extension>


<Output logstash>
    Module      om_tcp
    Host        192.168.190.128
    Port        10515
    Exec        parse_syslog();to_json();
</Output>

<Output out2>
    Module      om_tcp
    Host        192.168.190.128
    Port        10516
    Exec        parse_syslog();to_json();
</Output>
```

Figure 23: Prise d'écran du fichier de configuration Nxlog (2)

4.2. Normalisation des logs (Logstash)

Quand les logs arrivent à Logstash, leur contenu est encore dans son format d'origine ; et si on a des logs provenant de différentes sources, ils auront des formats différents, ce qui posera un problème dans l'une des prochaines étapes qui est la corrélation.

On devrait donc sauvegarder les logs qu'on a collectés sous un format unifié (par type) pour qu'on puisse leur appliquer des opérations en vrac et c'est exactement le rôle de Logstash.

Logstash a les fonctionnalités nécessaires pour trier, filtrer, modifier, et envoyer les logs en ajoutant ou en modifiant les informations contenues dans ceux-ci (organisés dans des champs), cela se fait, en fournissant Logstash avec des fichiers de configuration contenant entre autres :

- **Fichier de configuration des données d'entrée**

Un fichier qui permet de déterminer les entrées du serveur de normalisation, on donne l'adresse et le numéro de port des logs envoyé par le serveur de collection et un type qui va nous permettre de déterminer si on applique les filtres de normalisation ou non et cela pour séparer les logs qu'on veut normaliser et ceux qu'on veut archiver.

```
input {
  tcp {
    port => 10515
    type => "siem_logs"
    codec => json_lines { charset => CP1252 }
  }
  tcp {
    port => 10516
    type => "archive"
    codec => json_lines { charset => CP1252 }
  }

}
```

Figure 24: Prise d'écran du ficher de configuration des données d'entrée

- **Fichier de configuration pour normaliser les logs**

Un fichier qui contient les expressions régulières nécessaires pour normaliser un log.

On génère ce fichier automatiquement lorsque l'utilisateur transmet un log au SIEM qui n'est pas déjà pris en charge.

Les expressions régulières vont faire la correspondance de motif "Pattern Matching" et c'est ce dernier qu'on va utiliser pour identifier et modifier les champs pour construire notre format unifié propre à chaque type et catégorie de log.

On peut ajouter et supprimer et modifier des champs comme il est expliqué auparavant.

```
filter {
  if "AccessLog_" in [SourceModuleName] {

    json { source => "message" }

    mutate { rename => { "Message" => "message" } }

    grok {
      match => { "message" => "- - \[%{DATA:timestamp2}\] \"%{DATA:method} %{DATA:PAGE}\"
      %{POSINT:response} %{GREEDYDATA:bytes}"}
      match => { "message" => "%{COMBINEDAPACHELOG}" }
      add_field => [ "parsed", "NO" ]
    }

    mutate {

      remove_field => "SourceModuleType"
      remove_field => "timestampe2"
      remove_field => "@version"
      remove_field => "message"
      remove_field => "EventTime"
      remove_field => "EventReceivedTime"

    }
  }
  }
```

Figure 25: Prise d'écran du fichier de configuration du filtre des logs web

- **Fichier de configuration des données de sortie**

Ce fichier sert à déterminé ou envoyer les logs après l'étape de normalisation, dans notre les logs vont être redirigé vers Elasticsearch.

4.3. Sauvegarde des logs (Elasticsearch)

On donne l'adresse IP du nom de l'index d'Elasticsearch et Logstash va automatiquement envoyer les logs vers l'instance d'Elasticsearch.

```
output {
    elasticsearch{
          hosts => ["192.168.190.128:9200"]
          index => "siem"
    }

    stdout{codec => rubydebug}
  }
```

Figure 26: Prise d'écran du fichier de configuration de sortie

4.4. Corrélation des logs

Pour la corrélation des logs nous avons implémenté des algorithmes pour détecter certains types d'attaque, tous les algorithmes ont été conçus et implémenté en programmation multithread pour assurer la rapidité nécessaire.

Parlons du fonctionnement de quelqu'un :

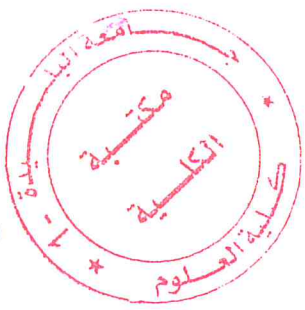
- **Détection d'une attaque de type DOS dans des logs d'un pare-feu**

La première étape consiste a détecté si le sous-moteur de corrélation du type DOS est activé avec la fonction écrite « **check_firewall_rule** » (voir figure 27).

```
def check_firewall_rule():
    while True:
        if Settings.correlation_engine_dos_rules:
            all_rules = RulesTable.objects.all()
            for rule in all_rules:
                if rule.status and rule.log_type == "firewall" and rule.rule_type
                    thread = StartFirewallRule()
                    thread.start()
        else:
            break
```

Figure 27: Code source de la fonction de détection DOS (1)

Si c'est le cas on lance le thread correspondant à ce type d'attaque, qui lui-même va exécuter une fonction « **dos_firewall_rule** » (voir figure 28)



```
209 def dos_firewall_rule():
210     global list_of_ips
211     client = Elasticsearch("192.168.190.128")
212     q = Q("match", SourceModuleName="firewall".decode('utf-8'))
213     while True:
214         i = elasticsearch_dsl.Search(using=client, index="siem").query(q)
215         i = i[0:2000]
216         response = i.execute()
217         for s in response:
218             if s.SRCIP not in list_of_ips:
219                 if s.parsed == "NO":
220                     list_of_ips.append(s.SRCIP)
221                     Time = datetime.datetime.strptime(str(s.EventTime), '%Y-%m-%d %H:%M:%S')
222                     thread = DosFirewallThread(s.SRCIP, Time, s, s.meta.id)
223                     thread.start()
224
```

Figure 28: Code source de la fonction de détection DOS (2)

Qui va s'exécuter en boucle en collectent les informations nécessaires depuis la base de données (Elasticsearch) avec la bibliothèque de python d'Elasticsearch (ligne 211-216), une fois les informations obtenu, on fait un test sur l'adresse IP (ligne 218) qui se trouve dans le log avec une liste qui contient les adresses IP déjà analysé si c'est le cas on ignore le log et on passe au suivante, si c'est le contraire on vérifie si on a analysé déjà (ligne 219) ce log en vérifiant le champ « PARSED » dans celui-ci, si c'est pas le cas alors on ajoute l'adresse IP a la liste, on collecte l'heure d'évènement du log (ligne 220-222), ensuite on lance un autre thread qui va traiter le log pour vérifier si l'adresse IP soupçonné est en train de lancer une attaque DOS avec la fonction « DosFirewallThread » (voir figure 29) qui prend quatre arguments qui sont :

- L'adresse IP du log.
- L'heure de l'évènement.
- Les autres informations du log
- L'identificateur du log.

```
DosFirewallThread(s.SRCIP, Time, s, s.meta.id)
```

Figure 29: Entête fonction DosFirewallThread

Le thread se lance avec les informations envoyées depuis la fonction précédente, on redemande les informations de la base de données des logs au moment d'un changement depuis le lancement du thread, on vérifie l'adresse IP si elle est égale à celle qu'on est en train d'analysé, si ce n'est pas le cas on ignore le log et on passe au prochaine. Si l'inverse on vérifie si le champ « ACTION » est égale à « DROP » car une des caractéristiques des logs d'un pare-feu que nous allons utiliser c'est, lorsque un pare-feu refuse une connexion il marque cette tentative par un « DROP » dans ces fichiers log. Si c'est le cas on va calculer la différence de temps entre le premier log et de celle qu'on est en train d'analyser en vérifiant **une contrainte de temps** émise par l'utilisateur, si elle dépasse alors on annule la recherche et on marque le log qui a lancé le thread comme « PARSED », si elle ne dépasse pas on vérifie une deuxième contrainte qui est **le nombre de tentative** qui déterminera si on déclare une alerte d'attaque ou pas. Si la contrainte est respectée on sauvegarde les logs dans la base de données de l'application avec tous les logs qui ont déclenché cette alerte pour permettre à l'administrateur du système de connaitre exactement la cause de l'alerte (voir figure 30).

```
client.update(index="siem", doc_type="siem_logs", id=doc_id,
              body={"script": {"inline": "ctx._source.parsed = params.prs",
                               "lang": "painless",
                               "params": {"prs": "YES"}}})

correlated_event = CorrelatedEvents(severity="Danger", source_ip=original_source.SRCIP,
                                    source_port=original_source.SRCPORT,
                                    destination_ip=original_source.DSTIP,
                                    destination_port=original_source.DSTPORT,
                                    event_time=original_source.EventTime,
                                    event_id=doc_id)
correlated_event.save()

for element in data_list:
    connected_event = ConnectedEvents(c_severity=element.SyslogSeverity,
                                      c_source_ip=element.SRCIP,
                                      c_source_port=element.SRCPORT,
                                      c_destination_ip=element.DSTIP,
                                      c_destination_port=element.DSTPORT,
                                      c_event_time=element.EventTime,
                                      c_event_id=doc_id)
    connected_event.save()

notification = Notifications(message="Dos Attack - Firewall Rule Triggered",
                             severity="Danger")
notification.save()
```

Figure 30: Code source de la fonction de détection DOS (3)

Mais si la contrainte n'est pas respectée on marque le log comme « PARSED » et dans les deux cas le thread lancé est tué.

Cette procédure est répétée pour chaque log dans la base de données des logs qui respecte les conditions mentionnées au-dessus.

5. Evaluation de l'application

Les tests faits sur l'application ont été réalisé sur un ordinateur avec un processeur I5 7200 et une mémoire de 8GB de Ram.

PS : Les performances de l'application dépendent de l'implémentation des outils et des algorithmes mais aussi sur les capacités de l'ordinateur, un ordinateur plus performant donnera des résultats plus rapides.

5.1. Vitesse de collection des logs

Le test a été réalisé en alimentant le système par des tranches de fichier de log et mesurer le temps nécessaire à la collection de celle-ci par le système.

La figure suivante représente un graphe de temps en fonction de nombres de logs.

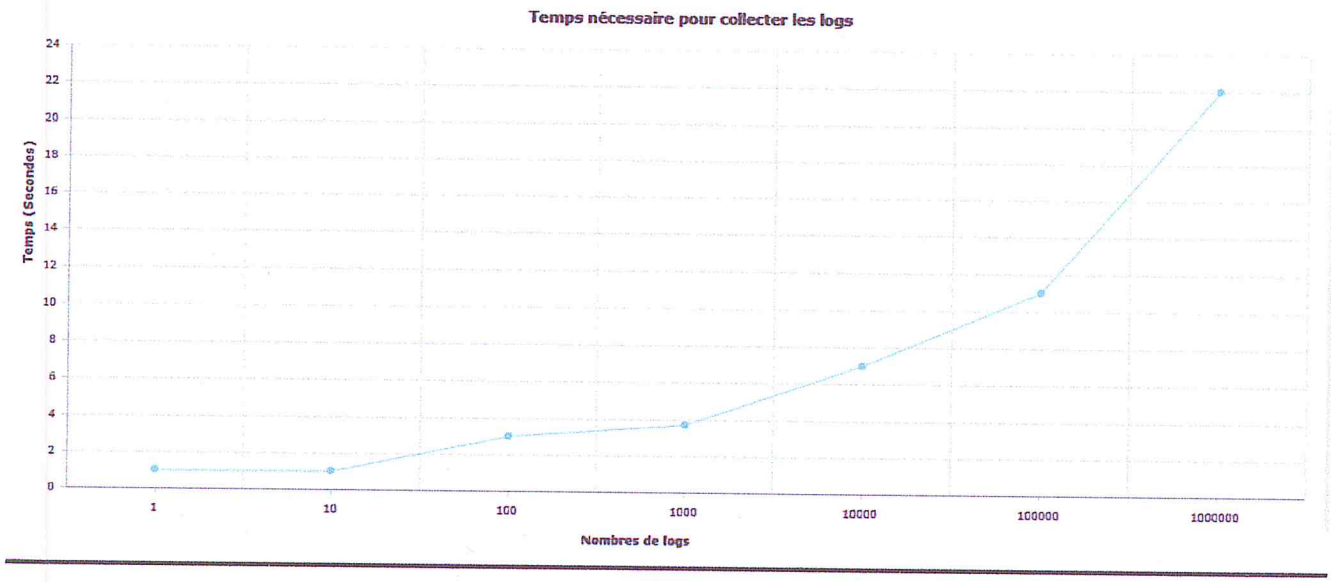


Figure 31: Temps Nécessaire pour collecter les logs

On voit clairement que le temps de réponse (collection) reste trop rapide malgré l'augmentation apparente en nombre de logs.

5.2. Le temps de réponse à une attaque

Le temps de réponse du moteur de corrélation dépend de plusieurs facteurs :

- **Performance du matériel**

Plus le matériel est performant plus le temps de réponse est réduit.

- **La complexité des algorithmes de corrélation**

Tous les algorithmes conçus pour cette application sont d'une complexité linéaire, car les algorithmes ne contiennent pas des boucles imbriquées.

- **La complexité des algorithmes de recherche de base de données**

Elasticsearch qui est la base de données des logs utilisé dans cette solution a une complexité polynomiale.

Par conséquent le temps de réponse à une attaque par le moteur de corrélation est polynomial.

5.3. Consommation des Ressources :

Les tests ont été réalisés en lançant l'application avec toutes ces fonctionnalités au même temps et en sauvegardant le résultat à chaque fois ensuite faire une moyenne, la figure suivante montre le résultat :

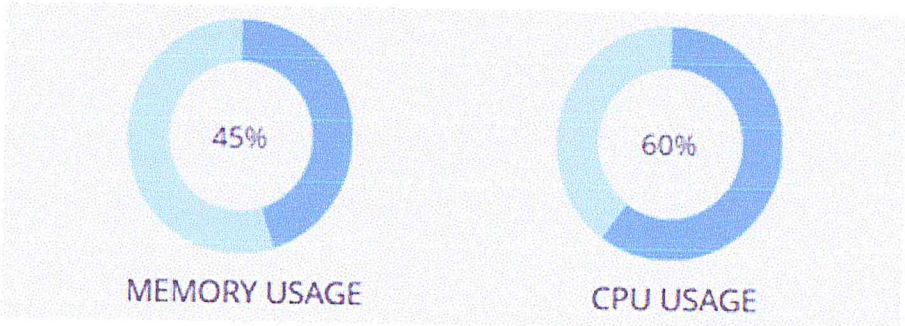


Figure 32: Consommation moyenne de l'application

5.4. Prix et licenciement

La solution proposée a été basé sur des algorithmes personnels et des outils open source.

6. Conclusion

Ce chapitre a présenté les différents outils, technologies, et concepts utilisés pendant le développement de ce projet, ainsi que comment ceux-ci ont été utilisés pour implémenter les différents composants de l'application, le tout avec des prises d'écran depuis cette dernière.

APERÇU DE L'APPLICATION

Dans ce chapitre nous allons illustrer les diverses fonctionnalités de notre application on fournissant des prises de vue de chaque fonctionnalité implémentée à partir de l'interface de notre application.

1. Tableau de bord (Dashboard):

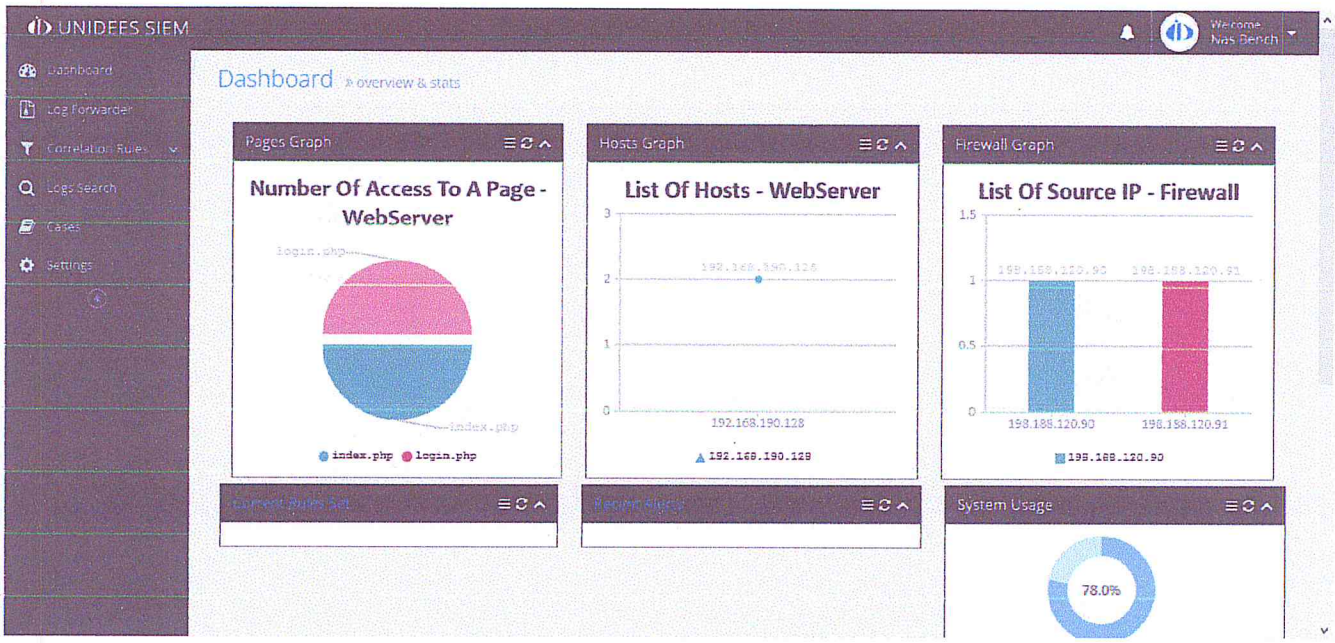


Figure 33: Prise d'écran du tableau de bord

La figure ci-dessus montre l'interface principale de l'application elle contient les fonctionnalités suivantes :

- La possibilité de visualiser trois graphes qui vont afficher les informations présentes dans le SIEM et la possibilité de les agrandir dans une page séparé pour plus de clarté.
- La possibilité de visualiser les règles de corrélations couramment activée et la possibilité de les modifier.
- La possibilité de visualiser les alertes récentes qui ont été déclenché par le moteur de corrélation.
- La possibilité de visualiser en temps réels la consommation des ressources du CPU et de RAM.
- La possibilité de visualiser les évènements corrélés en temps réels et d'interagir avec eux.
- Recevoir des notifications au cas d'une règle déclenchée, en temps réels.

2. Expéditeur de log :

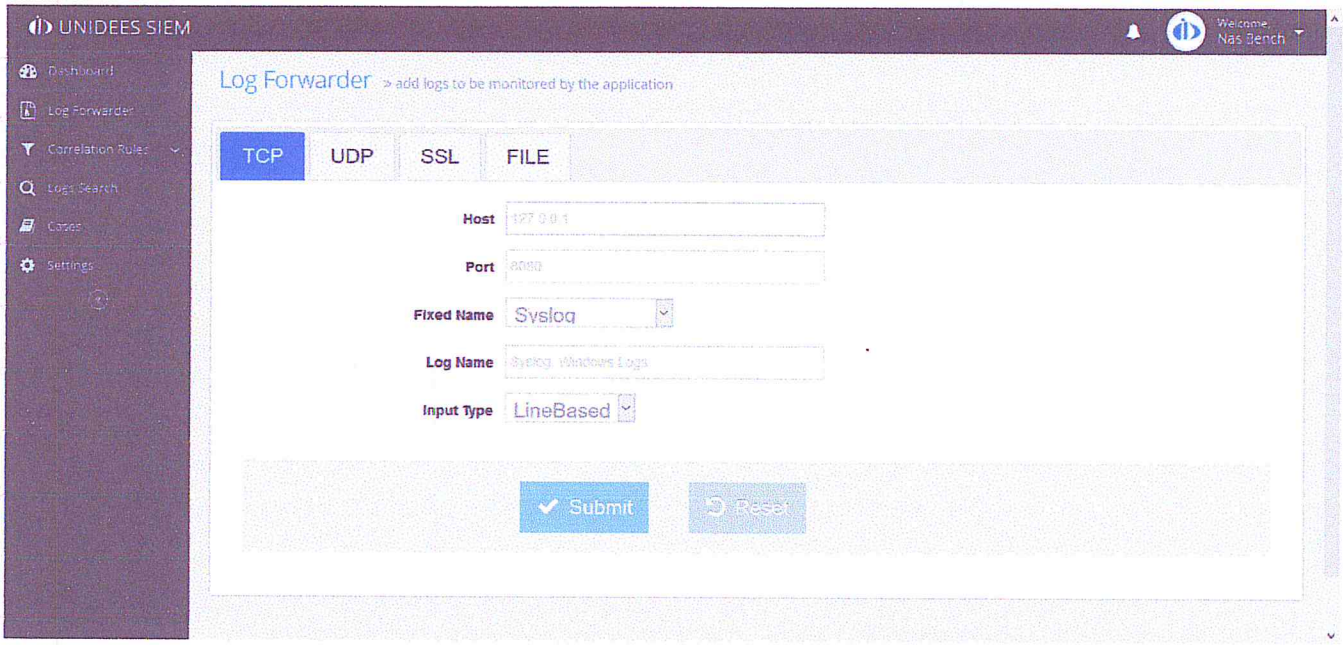


Figure 34: Prise d'écran de l'expéditeur de log

La figure ci-dessus montre l'interface qui va permettre à l'utilisateur de l'application d'envoyer les logs vers le serveur de collection des logs et de choisir une méthode d'envoi parmi les quatre disponibles (UDP, TCP, SSL, Fichier)

3. Interface de recherche :

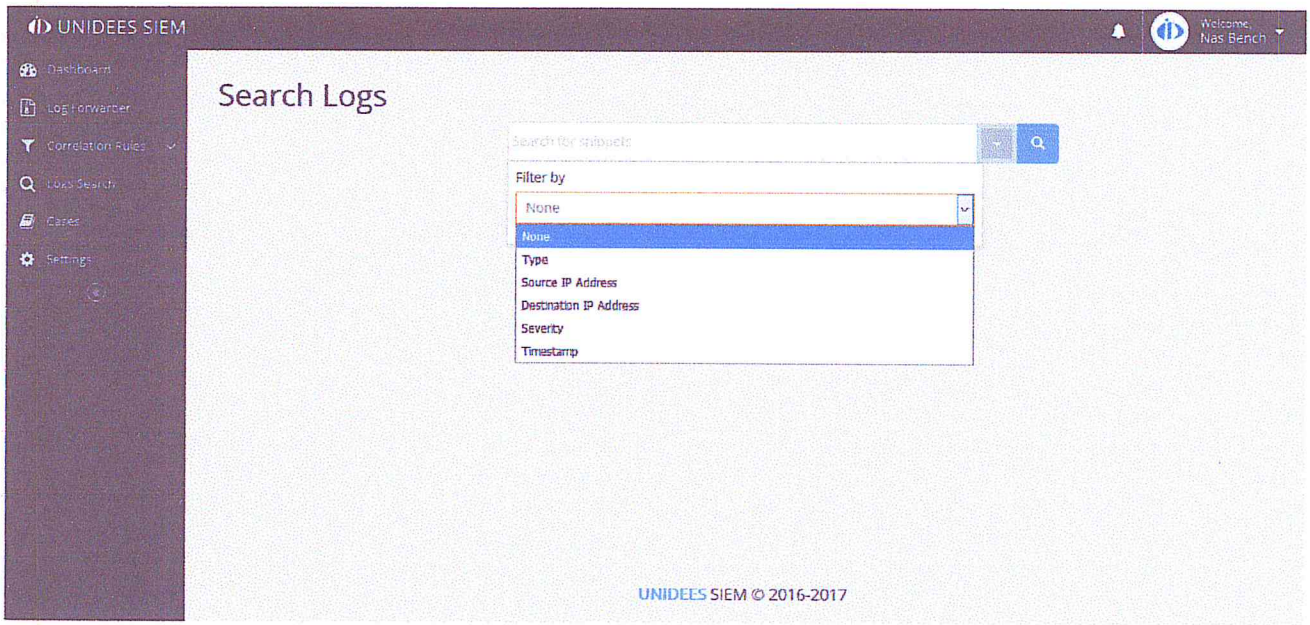


Figure 35: Prise d'écran de l'interface de recherche

La figure ci-dessus montre l'interface de recherche qui va offrir les fonctionnalités suivante :

- La possibilité de faire une recherche générale de tous les logs.
- La possibilité de filtrer le résultat de recherche à travers plusieurs filtres.

4. Interface des règles de corrélations :

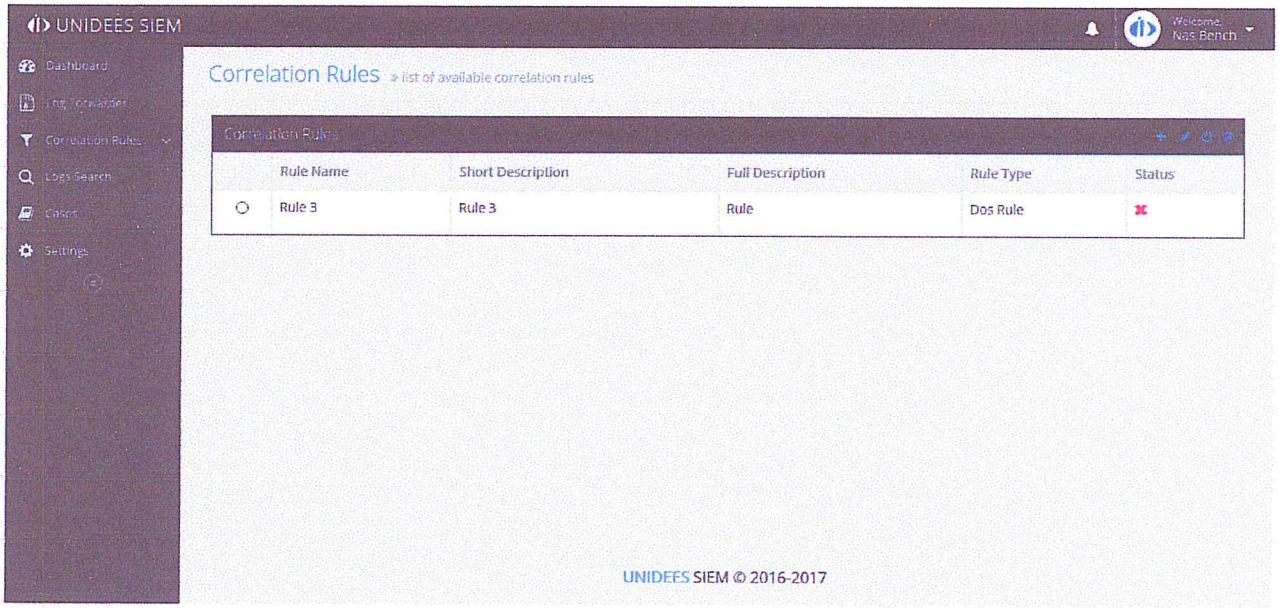


Figure 36: Prise d'écran de l'Interface des règles de corrélations

La figure ci-dessus montre l'interface des règles de corrélation qui offre les fonctionnalités suivantes :

4.1. L'ajoute d'une règle

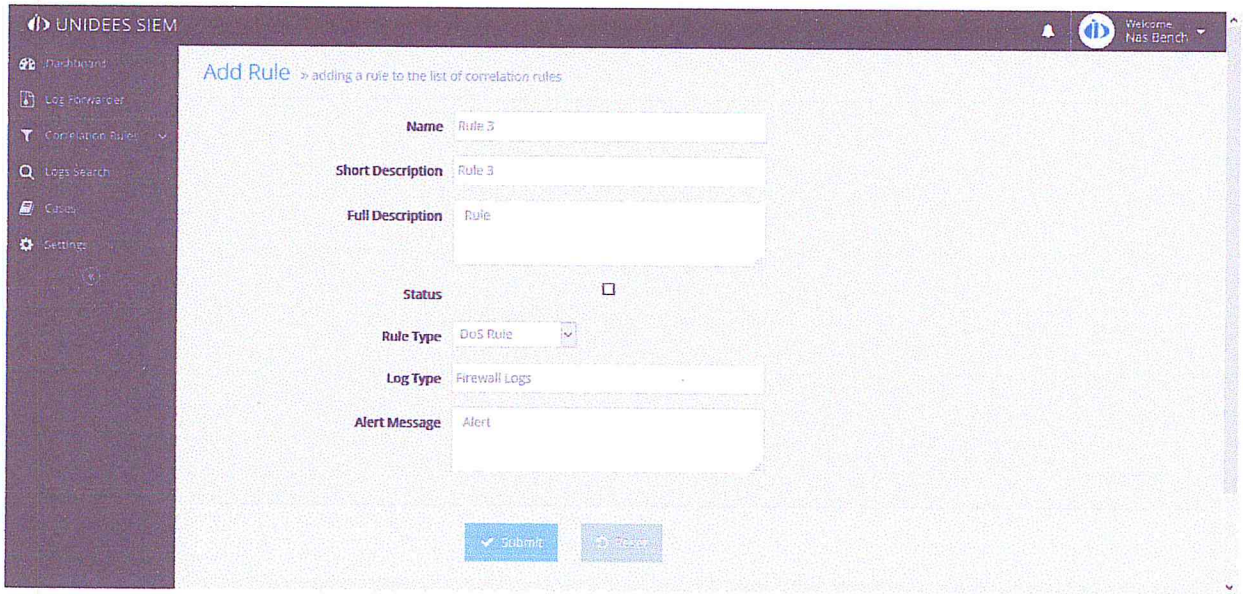


Figure 37: Prise d'écran de l'interface d'ajoute des règles

4.2. La modification d'une règle

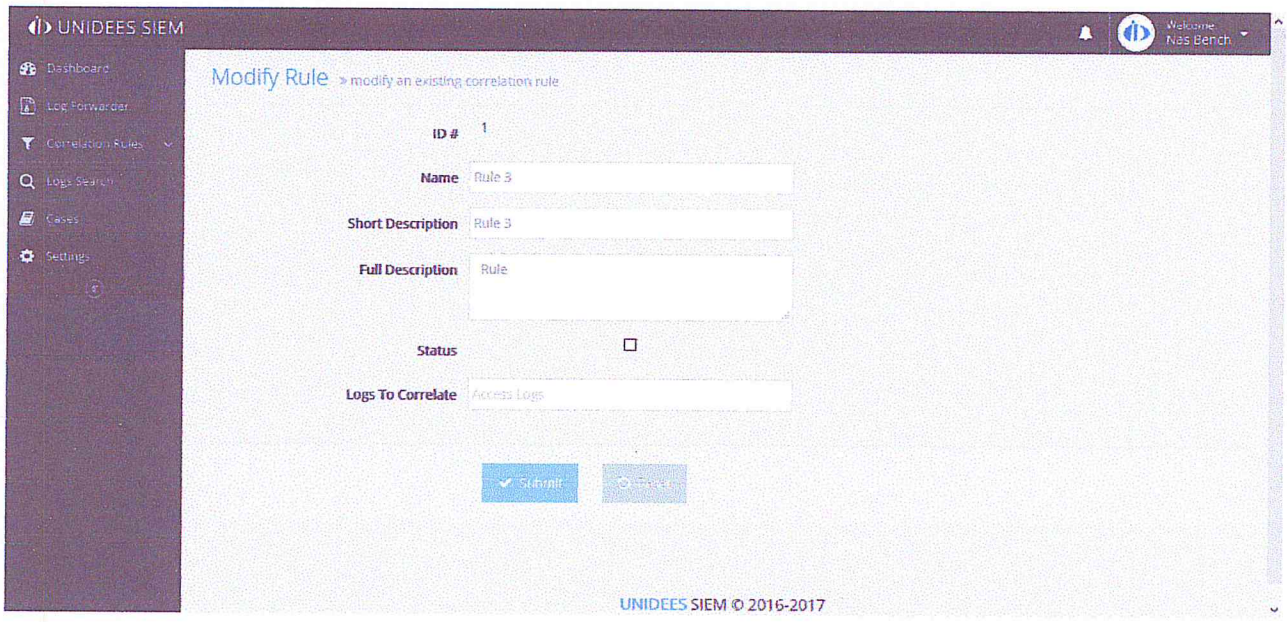


Figure 38: Prise d'écran de l'interface de modification des règles

En plus de ces deux fonctionnalités l'interface nous permet :

- D'activer ou désactiver une règle.
- La possibilité de supprimer une règle déjà créé.

5. Interface des paramètres :

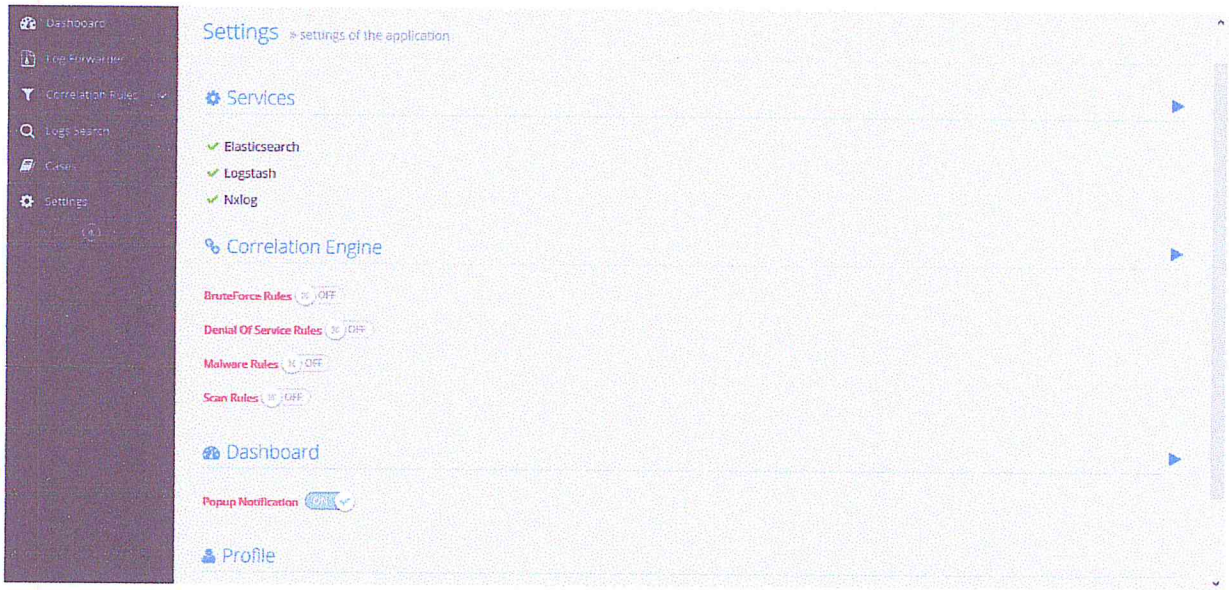


Figure 39: Pise d'écran de l'interface des paramètres

La figure ci-dessus montre l'interface des paramètres de l'application qui offre les fonctionnalités suivantes :

- La possibilité de visualiser les services active ou de les activés
- Les paramètres du moteur de corrélation quel type de moteur activé ou désactivé.
- La possibilité d'activer ou désactiver les notifications pop-up dans le Dashboard.
- La possibilité de consulter le profil de l'utilisateur pour modifier les informations présentes (voir figure 40).

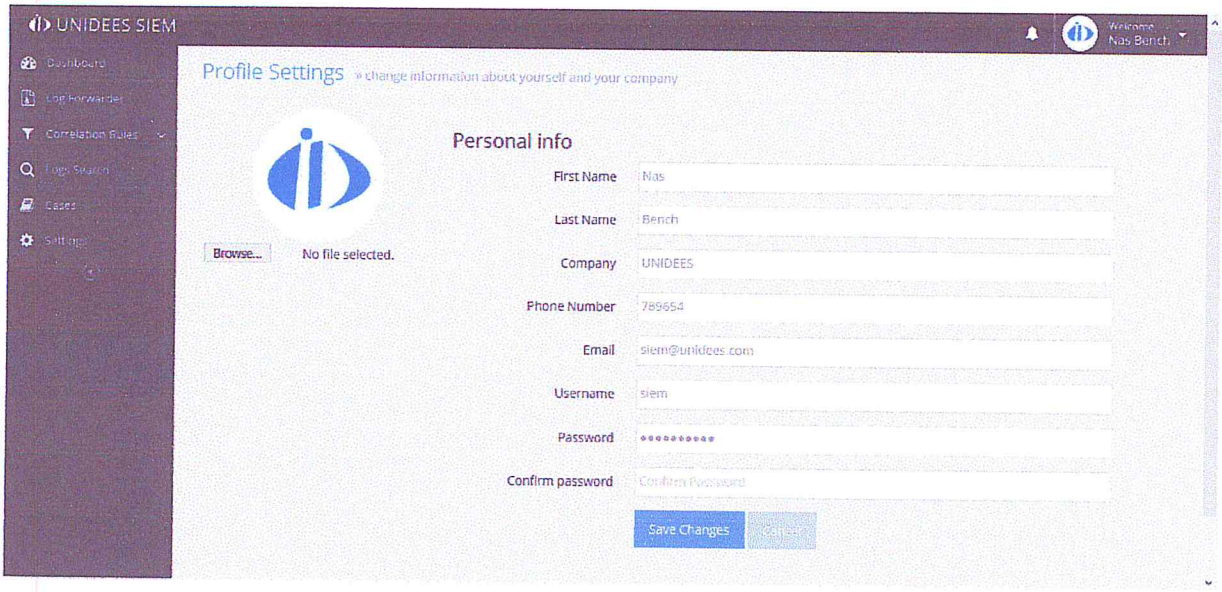


Figure 40: Prise d'écran de la page profile de l'utilisateur

7. Conclusion :

Ce chapitre a présenté les différentes fonctionnalités implémentées dans cette première version de l'application et il a illustré le tout avec des prises d'écran fournis depuis cette dernière.

CONCLUSION ET PERSPECTIVES

De nos jours, le système de sécurité à l'aide d'outils de gestion des évènements (SIEM) constitue un outil très performant pour la détection des attaques en temps réels et une aide très précieux pour la sécurité informatique.

Le thème que nous avons traité a été très enrichissant pour nous. En effet, il nous a permis de découvrir les difficultés liées à la détection des attaques et l'analyse en temps réels.

Sur le plan professionnel, ce stage nous a permis d'avoir une idée des réalités d'un monde autre que celui académique.

Sur le plan technique, ce stage a été une opportunité pour nous de mettre en pratique les connaissances acquises au cours de notre formation.

En termes de perspective on peut noter qu'il est possible de perfectionner et d'optimiser chaque composant du S.I.E.M pour cela on peut ajouter les éléments suivants :

- Optimisation des algorithmes de corrélations. Ce qui entraînera la réduction du taux de faux positives, l'amélioration de la vitesse de réponse, une consommation meilleure des ressources et une amélioration générale des performances de l'application.
- Rendre les règles de corrélations encore plus dynamiques pour permettre une détection plus performante.
- Améliorer le service de recherche des logs, en ajoutant plus de filtre et plus de restriction.
- Ajouter plus de fonctionnalités, comme la possibilité de créer des « cas » qui vont permettre à un utilisateur de l'application de créer un « dossier spécifique » pour une adresse IP et de suivre ces actions au fil du temps (est ce que cette adresse a déjà déclenché une alerte ou à essayer de faire une action suspect sur le réseau).
- Ajouter une fonctionnalité pour rechercher les logs non normalisé.

BIBLIOGRAPHIE

[1] A. A. H. C. B. S. V. S. H. David R. Miller, Security Information and Event Management (SIEM) Implémentation, McGraw-Hill Education, 2010.

[2] K. Karen et S. Murugiah, «Guide to Computer Security Log Management,» *NIST Publications*, p. 72, 2006.

[3] G. Rainer, «RFC 5424 - The Syslog Protocol,» Mars 2009. [En ligne]. Available: https://tools.ietf.org/html/rfc5424.

[4] I. Shakeel, «Evolution in the World of Cyber Crime,» Info Sec Institute, 28 Juin 2016. [En ligne]. Available: http://resources.infosecinstitute.com/evolution-in-the-world-of-cyber-crime/.

[5] N. C. C. Defence, «Cyber Definitions | CCDCOE,» [En ligne]. Available: https://ccdcoe.org/cyber-definitions.html.

[6] Techtarget, «Techtarget Definitions,» [En ligne]. Available: http://searchsecurity.techtarget.com/definition/.

[7] marketsandmarkets.com, «Security Information and Event Management Market by Services & Solutions - 2019 | MarketsandMarkets,» MARKETS AND MARKETS, Février 2014. [En ligne]. Available: http://www.marketsandmarkets.com/Market-Reports/security-information-event-management-market-183343191.html.

[8] «The top SIEM products: A buyer's guide,» [En ligne]. Available: http://searchsecurity.techtarget.com/essentialguide/The-top-SIEM-products-A-buyers-guide.

[9] I.-T. Research, «SIEM Comparison,» [En ligne]. Available: http://siemcomparison.com/.

[10] «NoSQL vs SQL- 4 Reasons Why NoSQL is better for Big Data applications,» 19 Mars 2015. [En ligne]. Available: https://www.dezyre.com/article/nosql-vs-sql-4-reasons-why-nosql-is-better-for-big-data-applications/86.

[11] e. company, «Logstash Introduction - Logstash Reference [5.4],» [En ligne]. Available: https://www.elastic.co/guide/en/logstash/current/introduction.html.

[12] e. company, «Basic Concepts - Elasticsearch Reference [5.4],» [En ligne]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/_basic_concepts.html.

[13] n. company, «NXLog Community Edition | High Performance Log Management Solutions,» [En ligne]. Available: https://nxlog.co/products/nxlog-community-edition.

[14] J. BRAINS, «PyCharm :: Download Latest Version of PyCharm,» JET BRAINS, Juillet 2010. [En ligne]. Available: https://www.jetbrains.com/pycharm/download/#section=windows.

[15] Wikipédia, «Kali Linux — Wikipédia,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Kali_Linux.

[16] apachefriends, «About the XAMPP project,» [En ligne]. Available: https://www.apachefriends.org/fr/index.html.

[17] Python, «General Python FAQ - Python 2.7.13 documentation,» [En ligne]. Available: https://docs.python.org/2/faq/general.html.

[18] Wikipedia, «Ajax (programming) - Wikipedia,» [En ligne]. Available: https://en.wikipedia.org/wiki/Ajax_(programming).

[19] W. L. Hakon et B. Bos, «Cascading Style Sheets, level 1,» W3C, 11 Avril 2008. [En ligne]. Available: https://www.w3.org/TR/CSS1/.

[20] I. Hickson, B. Robin, F. Steve, L. Travis, D. N. Erika, O. Edward et P. Silvia, «HTML5,» W3C, 28 Octobre 2014. [En ligne]. Available: https://www.w3.org/TR/html5/.

[21] M. D. Network, «JavaScript | MDN,» [En ligne]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript.

[22] M. Otto, J. Thornton et Bootstrap, «Bootstrap - The world's most popular mobile-first and responsive front-end framework,» [En ligne]. Available: http://getbootstrap.com/.

[23] «FAQ: General | Django documentation | Django,» [En ligne]. Available: https://docs.djangoproject.com/en/1.11/faq/general/.

[24] N. Big, «The Model-View-Controller Design Pattern,» [En ligne]. Available: http://djangobook.com/model-view-controller-design-pattern/.

[25] S. Banon, «Python Elasticsearch Client - Elasticsearch 5.4.0 documentation,» [En ligne]. Available: https://elasticsearch-py.readthedocs.io/en/master/.

[26] K. Scarfone, «Comparing the best SIEM systems on the market,» [En ligne]. Available: http://searchsecurity.techtarget.com/feature/Comparing-the-best-SIEM-systems-on-the-market.

[27] J. Sweeny, «Creating Your Own SIEM and Incident Response Toolkit Using Open Source Tools,» 20 Juin 2011. [En ligne]. Available: https://www.scribd.com/document/283430812/Creating-Siem-Incident-Response-Toolkit-Open-Source-Tools-33689.

GLOSSIARE

Terme	Définition
Big Data	Big Data (parfois appelées données massives), désignent des ensembles de données qui deviennent tellement volumineux qu'ils en deviennent difficiles à travailler avec des outils classiques de gestion de base de données ou de gestion de l'information.
Evènement	Un événement, dans le contexte informatique, est une occurrence identifiable qui a une signification pour le matériel, le logiciel, ou le réseau du système
Open-source	cette désignation s'applique aux logiciels dont la licence respecte des critères précisément établis par l'Open Source Initiative, c'est-à-dire les possibilités de libre redistribution, d'accès au code source et de création de travaux dérivés. Mis à la disposition du grand public, ce code source est généralement le résultat d'une collaboration entre programmeurs
Ransomware	Ransomware ou logiciel de rançon est un logiciel malveillant qui prend en otage des données personnelles. Pour ce faire, un Ransomware chiffre des données personnelles puis demande à leur propriétaire d'envoyer de l'argent en échange de la clé qui permettra de les déchiffrer
Expression régulière	Les expressions régulières sont des chaines de caractères ou des écritures compactes pour représenter un ensemble variable de séquences de caractères semblables en utilisant une syntaxe précise.
Reporting	Le Reporting est la génération des tableaux de bord et de notification
Requête	Une requête est une demande effectuée par un système vers un serveur lorsqu'il souhaite recevoir des données.
Rétention	Rétention est le stockage continu des données d'une organisation pour des raisons de conformité ou des raisons commerciales.
Rotation de logs	La rotation de logs c'est le faute de supprimer, ou déplacer des logs qui ont un certain âge vers un autre emplacement.