**UNIVERSITE SAAD DAHLAB BLIDA 1**

**Faculté des sciences**

Département informatique

# MEMOIRE DE MASTER

Option : sécurité des systèmes d'information

# CONCEPTION ET REALISATION D'UN SYSTEME DE GESTION ET D'ANALYSE DE LOGS

Mémoire Présenté par :

**SI-AHMED Ayoub** et **SI-AHMED Idris**

devant le jury composé de :

| | | |
|---|---|---|
| M.OULED-KHAOUA | Professeur, U. de Blida | Président |
| S. BENAISSI | Maître assistant B, U. de Blida | Examinateur |
| N.CHIKHI | Maître conference A, U. de Blida | Promoteur |

Blida, juillet 2019

**Table des matières**

Liste des figures ..... VI

Liste des tableaux ..... VIII

RESUME ..... IX

REMERCIEMENTS ..... XII

PRESENTATION DE L'ORGANISME D'ACCUEIL [38] ..... XIII

INTRODUCTION GENERALE ..... 1

CHAPITRE 1 : ETAT DE L'ART SUR LA GESTION DES LOGS ..... 3

1. GENERALITES SUR LES LOGS ..... 3

1.1. Définition des termes utilisés [33] ..... 3

1.2. Les classes de messages journaux ..... 4

1.3. Contenu de base d'un message journal ..... 5

2. LA COLLECTE ET LA TRANSMISSION DES FICHIERS JOURNAUX ..... 5

2.1. syslog [33] ..... 5

2.2. SNMP (Simple Network Management Protocol) [34] ..... 7

2.3. Windows Event Log [33] ..... 7

3. TECHNOLOGIES DE STOCKAGE DES LOGS ..... 8

3.1. Fichiers textes plats ..... 8

3.2. Fichiers logs binaires ..... 8

3.3. Bases de données ..... 8

3.4. Bases de données réparties ..... 9

4. LES ATTAQUES CONTRE LES SYSTEMES DE JOURNALISATION ..... 10

4.1. Attaques contre la confidentialité ..... 11

4.2. Attaques contre l'intégrité du système de journalisation ..... 12

4.3. Attaques sur la disponibilité du système de journalisation ..... 15

5. PRESENTATION DES TRAVAUX ANTERIEURS ..... 17

5.1. Forward Integrity For Secure Audit Logs [4] ..... 17

5.2. Logcrypt: Forward Security and Public Verification for Secure Audit Logs: [5] ..... 18
5.3. Secure Audit Logs to Support Computer Forensics [3] ..... 20
5.4. A New Approach to Secure Logging [2] ..... 22
5.5. On the Security of Public Key Protocols [6] ..... 24
5.6. The BSD (Berkeley Software Distribution) Syslog Protocol [10] ..... 26
5.6. IETF (Internet Engineering Task Force) syslog protocole [42] ..... 27
5.8. Safeguarding Cryptographic Keys [8] ..... 28
5.9. Proactive Secret Sharing [45] ..... 30
5.10. Secure Logging as a Service- Delegating log management to the cloud [9] ..... 30
6. CONCLUSION ..... 32
CHAPITRE 2 : ETAT DE L'ART SUR L'ANALYSE DES LOGS. ..... 34
1. ANALYSE MANUEL [33] ..... 34
1.1. Outils utilisé pour l'analyse manuelle ..... 34
1.2. Limites de l'examen du journal manuel ..... 35
2. ANALYSE DESCRIPTIFS ..... 36
3. FILTRER, NORMALISER ET CORRELER [33] ..... 37
3.1. Filtrer ..... 38
3.2. Normaliser ..... 38
3.3. Corréler ..... 38
4. PRESENTATION DES TRAVAUX ANTERIEURS ..... 39
4.1. Méthodes de filtrage ..... 39
4.2. Méthodes de préparation de fichiers journaux ..... 40
4.2.1. Méthodes heuristiques ..... 40
4.2.2. Méthodes de clustering approche IPLoM : [29] ..... 41
4.2.3. Méthodes basées sur le code source ..... 44
4.3. Techniques et méthodes d'analyse des journaux ..... 44

4.3.1. Apprentissage de règles associatives ..... 45
4.3.2. Test du $x2$ : [22] ..... 45
4.3.3. Apprentissage par arbre de décision [30] ..... 46
4.3.4. Classification hiérarchique ..... 47
4.3.5. Naive Bayes ..... 49
4.3.6. Machine à vecteurs de support ..... 50
5. AGIR SUR LES LOGS [33] ..... 52
5.1. Logs critiques ..... 52
5.2. Logs non critiques ..... 52
6. CONCLUSION ..... 54
CHAPITRE 03 : CONCEPTION ET IMPLEMENTATION DE LA SOLUTION 55
1. SPECIFICATION ET ANALYSE DES BESOINS ..... 55
1.1. Besoins fonctionnels ..... 55
1.2. Besoins non fonctionnels ..... 56
1.3. Architecture générale de notre système ..... 56
2. PARTIE CLIENT ET RELAI ..... 58
2.1. Syslog-ng ..... 59
2.1.1. Introduction ..... 59
2.1.2. Installation ..... 59
2.1.3. Synchronisation d'horloge ..... 60
2.1.4. Format syslog ..... 60
2.1.5. Options globales de syslog-ng ..... 61
2.1.6. Collecte des messages de journal ..... 61
2.1.7. Transfert sécurisé avec TLS (Transport Layer Security) ..... 62
2.1.8. Envoi et stockage des messages de journal ..... 66
2.1.9. Filtrer et classer ..... 68
2.1.10. Préparation et réécriture ..... 70

2.1.11. Chemins de journal ..... 73
2.1.12. La rotation des fichiers logs ..... 74
3. PARTIE SERVEUR ..... 77
3.1. Graylog ..... 77
3.2. Architecture Graylog ..... 77
3.3. Composants de Graylog ..... 78
3.3.1. MongoDB ..... 78
3.3.2. Elasticsearch ..... 78
3.3.3. Serveur Graylog ..... 78
3.4. Activation du HTTPS ..... 79
3.5. Ajout des inputs ..... 80
3.6. Configuration des clients ..... 82
3.7. Recherche ..... 83
3.8. Flux ..... 84
3.9. Alerte ..... 85
3.9.1. Conditions d'alerte ..... 85
3.9.2. Etat d'alerte ..... 86
3.10. Tableaux de bord ..... 87
3.10.1. Histogramme ..... 89
3.10.2. Valeurs statistiques ..... 89
3.10.3. Résultats de quick values ..... 90
3.11. Utilisateur et rôle ..... 91
3.12. Diagramme de déploiement ..... 92
4. ANALYSE BASEE SUR L'APPRENTISSAGE AUTOMATIQUE ..... 93
4.1. Approches en clustering ..... 93
4.2. Description de l'environnement de test ..... 94
4.3. Analyse des résultats ..... 96

5. CONCLUSION ..... 98

CONCLUSION GENERALE ..... 99

REFERENCE ..... VIII

## Liste des figures

Figure *5.1.1* : *Sécurité de transfert simple à l'aide de MAC (Message Authentication Codes)*……..…*18*

*Figure 5.2.1 : Sécurité avancée avec vérifiabilité*……..……..*19*

*Figure 5.2.2 : vérification des entrées dans le schéma à*…….....*19*

Figure 5.3.1*: Schéma Schneier-Kelsey*.....21

Figure 5.1.1*: Architecture système* pour *la journalisation sécurisée dans le cloud* ..... **Erreur ! Signet non défini.**

Figure 4.2.2.1*: Partition par position de jeton.* .....42

Figure 4.2.2.2 *: Partitionnez en recherchant bijection.* .....43

Figure 4.3.1.1*: Exemple de signature* .....45

Figure 4.3.3.1*: Simple Decision Tree*.....47

Figure 4.3.4.1*: La structure globale de LogCluster* .....49

Figure 4.3.6.1*: distribution spatiale des données d'apprentissage* .....50

Figure 4.3.6.2*: séparation entre 2 classes par un hyperplan* .....51

Figure 4.3.6.3*: Nuage de données d'apprentissage avec un Hyperplan optimal et une marge maximale*.....51

Figure 1.1.1 : *Diagramme de cas d'utilisation d'un administrateur*.....55

Figure 1.1.2 : *Diagramme de cas d'utilisation d'un superviseur*.....56

Figure 1.3.1 : *architecture générale.* .....57

Figure 2.1*: partie syslog-ng client [32].*.....58

Figure 2.2 *: partie syslog-ng Relai [32].*.....58

Figure 3.2.1 : *ARCHITECTURE montrant la configuration de Graylog*.....77

Figure 3.3.3*: capture d'écran de l'interface web* .....79

Figure 3.5.1 *: capture d'écran montrant l'état du fonctionnement de l'input.*..82

Figure 3.6.1*: capture d'écran montrant l'arrivé des messages log au serveur.* .....83

Figure 3.7.1*: capture d'écran du moteur de recherche Graylog.*.....83

Figure 3.8.1*: capture d'écran montrant le résultat de la création de flux.*.....85

Figure 3.9.2.1*: capture d'écran montrant une notification d'alerte non résolu.* .....87

Figure 3.10.1*: capture d'écran montrant le résultat de la création de tableaux de bord.*.....88
Figure 3.10.1.1*: capture d'écran montrant un histogramme des messages log reçu.*.....89
Figure 3.10.2.1*: capture d'écran montrant les différentes informations statistiques sur un champ particulier.*.....89
Figure 3.10.2.2 : *capture d'écran montrant respectivement la sévérité et le nombre d'application générant des messages log.*.....90
Figure 3.10.3.1*: capture d'écran montrant respectivement les nombres rapides des champs sévérité, nom d'application et fonction (facility).*.....91
Figure 3.11.1 : *CAPTURE d'écran montrant les différents utilisateurs.*.....92
Figure 3.12.1*: diagramme de déploiement.* .....92
*Figure 4.2.1 : Nombre d'instances dans le jeu de données d'apprentissage.* 95
*Figure 4.2.2 : Nombre d'instances dans le jeu de données de test.*.....96
*Figure 4.3.1 :REPRESENTATION DE DEUX TYPES DE CLASSES. NORMALE ET EN ANOMALIE* .....96

## Liste des tableaux

*Tableau 5.1 : Tableau comparatif des outils d'analyse les plus utiliser*.....37
Tableau 2.1.6.1*: Résumé des actions à entreprendre sur les messages critiques du journal*.....52
Tableau 2.1.8.1*: pilote source syslog-ng [32]..* .....62
Tableau 2.1.9.1 : pilote destination syslog-ng *[32].* .....66
Tableau 2.1.9.2*: valeur gravité syslog-ng [32].*.....68
Tableau 2.1.9.3 *: facilités syslog-ng [32].* .....69
Tableau 4.2.1 : lISTE D'ATRRIBUTS.....95
Tableau 4.3.1 : *DISTRIBUTION D'INSTANCES DANS LES CLUSTERS.* ....96

# RESUME

Les logs générés par les différents composants d'un système d'information sont souvent utilisés pour le contrôle du bon fonctionnement de ce même système. Dans le domaine de la sécurité, les logs s'avèrent être une source précieuse et incontournable pour la détection et la prévention d'éventuels risques.

La gestion et l'exploitation de ce type d'informations s'avèrent délicates en pratique en raison de la grande quantité et de l'hétérogénéité des logs générés par les différents systèmes routeurs, pare-feu, SGBD, serveur web, etc.

Notre travail se divise en deux parties

Dans la première partie nous avons conçu et réalisé un système robuste qui est capable de gérer de grandes quantités de messages logs ayant des formats différents en utilisant deux outils qui se complètent entre eux. syslog-ng pour la collecte et le prétraitement des logs et Graylog pour la gestion et l'analyse descriptive de ces messages, tout en définissant des alertes de sécurité, pour aider le responsable de la dite sécurité à y identifier d'éventuelles failles.

En ce qui concerne la deuxième partie nous avons effectué une analyse basée sur l'apprentissage automatique, plus précisément avec l'algorithme K-means. Cela a pour but de différencier un comportement normal d'un comportement anormal.

**Mots clés :** gestion de logs, analyse de logs, apprentissage automatique, Graylog, syslog-ng, K-Means.

# SUMMARY

The logs generated by the various components of an information system are often used to check the proper functioning of the same system. In the field of security, the logs prove to be a valuable and essential source for the detection and the prevention of possible risks.

The management and exploitation of this type of information is tricky in practice because of the large quantity and heterogeneity of the logs generated by the various routers, firewalls, DBMS, web server, etc.

Our work is divided into two parts

In the first part we have designed and realized a robust system that is able to handle large amounts of message logs having different formats using two tools that complement each other. syslog-ng for the collection and preprocessing of logs and Graylog for the management and the descriptive analysis of these messages, while defining security alerts, to help the person in charge of said security to identify any faults.

Regarding the second part we performed an analysis based on machine learning, more precisely with the K-means algorithm. This is intended to differentiate normal behavior from abnormal behavior.



**Keywords :** log management, log analysis, machine learning, Graylog, syslog-ng, K-Means.

# المستخلص

غالبًا ما تستخدم السجلات التي يتم إنشاؤها بواسطة المكونات المختلفة لنظام المعلومات للتحقق من حسن سير العمل في النظام نفسه. في مجال الأمن، كما تثبت السجلات أنها مصدر قيّم وضروري للكشف عن المخاطر المحتملة ومنعها.

تعد إدارة واستغلال هذا النوع من المعلومات أمرًا صعبًا في الممارسة نظرًا للكمية الكبيرة للمعلومات، وعدم تجانس السجلات الناتجة عن مختلف أجهزة التوجيه و حواجز الحماية وقواعد إدارة البيانات وخادم الويب، الخ routeurs, pare-feu, SGBD, serveur web.

و في هذا السياق قمنا بتقسيم العمل إلى جزئين كما يلي:

في الجزء الأول قمنا بإعداد و تصميم نظامًا قويًا قادرًا على التعامل مع كميات كبيرة من سجلات الرسائل ذات تنسيقات مختلفة باستخدام أداتين يكمل كل منهما الآخر.Syslog-ng لجمع السجلات المسبقة ومعالجتها وGraylog للإدارة والتحليل الوصفي لهذه الرسائل، مع تحديد تنبيهات الأمان، لمساعدة الشخص المسؤول عن الأمان على تحديد الأخطاء المحتملة. بينما في الجزء الثاني، أجرينا تحليلًا معتمدين على التعلم الآلي، وبشكل أكثر دقة قمنا باستخدام خوارزمية الوسائل k-means الذي يهدف إلى التمييز بين السلوك الطبيعي والسلوك غير الطبيعي

**الكلمات المفتاحية:** إدارة السجل ، تحليل السجل ، التعلم الآلي ، k-Means, Syslog-ng, Graylog

# REMERCIEMENTS

Nous remercions Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce travail de master.

Nous tenons à remercier chaleureusement monsieur le professeur CHIKHI Nacim Fateh notre promoteur, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter notre réflexion et à structurer notre travail.

Nous remercions notre encadrant monsieur ZOUGAR abdellah chef de cellule réseaux a Algérie télécom, qui nous a fait découvrir l'opportunité de notre sujet et qui nous a guidé dans notre Projet.

Un grand merci à monsieur HAMOUDA mourad directeur a Algérie Telecom qui nous a accueilli et apporté son soutien tout au long de notre recherche.

Nous remercions nos très chers parents, qui ont toujours été là pour nous ainsi que mes frères et ma petite sœur.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Enfin, nous adressons nos sincères remerciements à tous nos professeurs qui nous ont formés durant notre cursus universitaire et à tous les intervenants, qui, par leurs conseils et leurs critiques ont aiguisé notre réflexion durant nos recherches.

À tous, nous présentons notre gratitude et notre respect.

# PRESENTATION DE L'ORGANISME D'ACCUEIL [38]

« ALGERIE TELECOM » est leader sur le marché Algérien des télécommunications qui connait une forte croissance. Offrant une gamme complète de services de voix et de données aux clients résidentiels et professionnels. Cette position s'est construite par une politique d'innovation forte adaptée aux attentes des clients et orientée vers les nouveaux usages. Entrée officiellement en activité à partir du 1er janvier 2003, elle s'engage dans le monde des Technologies de l'Information et de la Communication avec trois objectifs :

- Rentabilité.
- Efficacité.
- Qualité de service.

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

# INTRODUCTION GENERALE

Les fichiers logs sont des enregistrements historiques de l'état de fonctionnement du matériel et des logiciels. Ils stockent des informations sur leur utilisation, les erreurs qui se produisent et les événements spécifiques à l'application, qui détaillent la manière dont les utilisateurs interagissent avec eux. L'examen régulier de ces informations peut fournir aux administrateurs système et aux équipes de sécurité informatique un aperçu de l'efficacité du fonctionnement de l'entreprise et des erreurs de configuration susceptibles de générer des problèmes sur le réseau, de sorte qu'ils puissent être résolus avant qu'ils aient un impact plus important.

Dans une étude récente [1], l'examen des logs n'a révélé que 6% des violations du réseau alors que 86% des victimes détenaient des preuves de la violation dans leurs fichiers logs. Cette statistique montre que les enregistrements de logs constituent également une source d'informations extrêmement précieuse pour la sécurité informatique, mais leur valeur en tant que partie intégrante d'un processus de détection d'intrusion et de réponse aux incidents est en grande partie mal comprise par de nombreuses organisations. Les logs ne sont pas collectés du tout ou sont collectés sans considération de la manière dont ils pourraient être utilisés en cas d'incident.

Développer une stratégie de gestion des logs pour améliorer la sécurité informatique d'une entreprise n'est pas une tâche aisée et un certain nombre d'obstacles doivent être surmontés. Le nombre de périphériques générant des journaux sur un réseau peut être considérable, et les stratégies de stockage peuvent être à la fois coûteuses et complexes à mettre en œuvre. Comment garantir l'intégrité, la confidentialité et la disponibilité des donnés log ? Quels journaux devraient être collectés ? Pendant combien de temps devraient-ils être stockés ? Comment doivent-ils être stockés pour assurer au mieux la sécurité et l'intégrité des données ? Comment les fichiers doivent-ils être utilisés de manière proactive pour rechercher les « aiguilles » malveillantes dans une « botte de foin » des données disponibles ? La leçon pour les organisations est simple : les journaux sont une source de preuves

d’une importance vitale, mais des efforts sont nécessaires pour bien les exploiter.

Dans ce contexte l’objectif de ce mémoire est de concevoir et d’implémenter un système de traitement centralisé des logs.Le systèmedevra en outre offrir un outil permettant d’analyser les logs afin de renforcer la politique de sécurité d’Algérie Télécom.

Le mémoire est divisé en trois chapitres.Le premier chapitre parlera des généralités sur les logs, des technologies de collecte, transmission et de stockage des fichiers journaux. Nous donnerons ensuite les différentes attaques que peut subir un système de journalisation. En continuité, nous exposerons les travaux antérieurs qui ont été menés pour assurer la sécurité des différentes phases de gestion de log.

Dans le deuxième chapitre nous présenterons d’abord les techniques manuelles d’analyse des fichiers logs, puis des outils utilisé pour faire une analyse descriptive des logs, en définitive nous parlerons des techniques plus avancées basées sur des méthodes de préparation et d’analyse automatique de logs. Nous parachèverons ce chapitre par les actions qui peuvent être entreprise pour répondre à certaine situation.

Le dernier chapitre décrira la conception et l’implémentation de notre solution qui utilisera deux outils. Le premier outil « syslog-ng » pour la préparation et le transfert des logs. Le deuxième outil « Graylog » pour le stockage et la visualisation. Enfin l’analyse s’effectuera avec l’application d’un algorithme d’apprentissage automatique en utilisant la méthode de clustering pour la détection d’anomalies.

La conclusion générale quant à elle aura pour double fonction de présenter les limites de notre solution et d’en exposer des axes d’ouverture afin de l’améliorer.

# CHAPITRE 1 : ETAT DE L'ART SUR LA GESTION DESLOGS

## 1. GENERALITES SUR LES LOGS

### 1.1. Définition des termes utilisés [33]

Évènement : est une occurrence unique dans un environnement, impliquant généralement une tentative de changement d'état. Un événement inclut généralement une notion de temps, d'occurrence et tout détail explicitement lié à l'événement ou à l'environnement pouvant aider à expliquer ou à comprendre les causes ou les effets de l'événement.

Champ d'événement : décrit une caractéristique d'un événement. Les exemples de champ d'événement incluent la date, l'heure, l'adresse IP source, l'identification de
L'utilisateur et l'identification de l'hôte.

Entrée de journal ou bien enregistrement de journal (event record) : est une collection de champs d'événement qui ensemble décrivent un seul événement.

Log : est une collection d'enregistrement de journal. Les termes tels que «journal de données», «journal d'activité», «journal d'audit", "fichier journal", "journal des événements» sont souvent utilisés pour signifier la même chose que log (journal).

Fichiers log : ce sont des ensembles séquentiels de messages émis par un programme informatique ou un périphérique qui rendent compte de leur exécution.

Un audit :est le processus d'évaluation des journaux dans un environnement (par exemple, dans un système électronique). L'objectif principal d'un audit est d'évaluer l'état général ou de déterminer toute activité notable ou problématique.

Journalisation : c'est le fait de stocker les fichiers logs dans une base de données ou autres supports.

Alerte ou Alarme : est une action effectuée en réponse à un événement, généralement destiné à attirer l'attention de quelqu'un (ou de quelque chose).

### 1.2. Les classes de messages journaux

Ils peuvent être classés dans les catégories générales suivantes [33] :

Informatif :Les messages de ce type sont conçus pour que les utilisateurs et les administrateurs sachent qu'un événement bénin s'est produit.

Débogage :les messages de débogage sont généralement générés à partir de systèmes logiciels afin d'aider les développeurs à résoudre et à identifier les problèmes liés à l'exécution du code de l'application.

Avertissement :les messages d'avertissement concernent des situations dans lesquelles des éléments peuvent manquer ou être nécessaires pour un système, mais leur absence n'aura pas d'incidence sur le fonctionnement du système.

Erreur :ces messages sont utilisés pour relayer les erreurs qui se produisent à différents niveaux dans un système informatique ; de nombreux messages d'erreur ne donnent qu'un point de départ pour expliquer pourquoi ils se sont produits. Une enquête plus approfondie est souvent nécessaire afin de rechercher la cause fondamentale de l'erreur.

Alerte : une alerte indique qu'un événement intéressant s'est produit.

### 1.3. Contenu de base d'un message journal

Un message journal contient généralement les informations suivantes [33] :

Timestamp : indique l'heure à laquelle le message de journal a été généré.

Source : est le système qui a généré le message de journal.

Données : il n'existe pas de format standard pour la représentation des données dans un message de journal. Parmi les éléments de données les plus courants que l'on peut trouver dans un message de journal figurent les adresses IP source et destination, les ports source et destination, les noms d'utilisateurs, les noms de programmes etc.

*Exemple :*

```
Jul 13 19:17:23 10.240.46.16 15: *Mar  1 00:16:17: %LINEPROTO-5-UPDOWN: Line
protocol on Interface FastEthernet0/0, changed state to up
```

## 2. LA COLLECTE ET LA TRANSMISSION DES FICHIERS JOURNAUX

Après la génération des logs par les divers périphériques et ordinateurs, les fichiers journaux doivent être transmis et collectés dans un endroit que l'on l'appelle un collecteur (loghost).

Un collecteur est un système informatique, généralement un système Unix ou Windows server, où les messages de journalisation sont collectés [33].

Les protocoles utilisés pour la transmission des fichiers logs sont :

### 2.1. syslog [33]

C'est un protocole qui se compose d'une partie *cliente* et d'une partie *serveur*. La partie cliente émet les informations sur le réseau, via le port UDP 514. Les serveurs collectent l'information et se chargent de créer les journaux.

L'intérêt de syslog est donc de centraliser les journaux d'événements, permettant de repérer plus rapidement et efficacement les défaillances d'ordinateurs présents sur un réseau.

Cependant il n'y a pas de distribution fiable des messages de journal. Le principal inconvénient de syslog est qu'il ne protège pas les enregistrements de journal pendant la transmission ou le transfert point à point.
Il existe différentes implémentations du protocole syslog :

- syslog-ng : est une amélioration de syslog. Certaines des fonctionnalités incluent la prise en charge du protocole IPv6, la possibilité de transférer des messages de journal avec fiabilité à l'aide de TCP et le filtrage du contenu des journaux à l'aide d'expressions régulières. Il utilise SSL pour une transmission sécurisée des fichiers journaux. syslog-ng ne protège pas contre les modifications de données de journal lorsqu'il est placé sur un nœud final.
- Syslog-sign : est une forme améliorée de syslog qui ajoute l'origine de l'authentification, l'intégrité des messages, la résistance à la lecture, le séquencement des messages et la détection des messages manquants à l'aide de deux messages supplémentaires : "blocs de signature" et "blocs de certificats". Malheureusement, ils sont supprimés après l'authentification, par conséquent l'intégrité du transfert n'est que partiellement remplie. De plus, syslog-sign ne fournit aucune confidentialité lors de la transmission.
- syslog-pseudo : est également une amélioration de syslog qui propose une architecture de journalisation pour pseudonymiser les fichiers journaux. L'idée principale est que les enregistrements du journal sont d'abord traités par le pseudonymiseur avant d'être archivés. Le pseudonymiseur consiste à remplacer des champs spécifiques par des pseudonymes soigneusement conçus. Par conséquent les messages log générés ne sont pas identiques à ceux qui sont stockés et l'origine des messages logs ne peut être restaurée pour les besoins d'investigation. syslog-pseudo ne protège pas les enregistrements de journal des attaques contre la sécurité c'est-à-dire des violations de la confidentialité et de l'intégrité.
- Reliable-syslog : a pour objectif la mise en œuvre d'une livraison fiable des messages syslog, qui repose sur les blocs du protocole BEEP

(Extensible Exchange Protocol) qui s'exécute sur TCP pour fournir le service de livraison fiable requis. Il permet l'authentification des périphériques et incorpore des mécanismes pour protéger l'intégrité des messages de journal contre les attaque de rejeu et autres attaques. Cependant il n'assure pas la confidentialité lors de la transmission des messages logs.

## 2.2. SNMP (Simple Network Management Protocol) [34]

SNMP a été créé à l'origine pour être utilisé dans la gestion des périphériques réseau. Cependant, au fil des années, de nombreux systèmes qui ne sont pas mis en réseau ont adopté SNMP comme moyen d'émettre des messages de journalisation. Le noyau de SNMP est constitué d'un ensemble simple d'opérations (et des informations qu'elles recueillent) qui permet aux administrateurs de modifier l'état de certains périphériques SNMP. Nous pouvons, par exemple, utiliser SNMP pour fermer une interface de notre routeur ou vérifier la vitesse à laquelle notre interface Ethernet fonctionne. Celui-ci peut même surveiller la température sur notre commutateur et nous avertir quand elle est trop élevée.

De nombreux périphériques réseau sont capables d'envoyer des informations sur les événements via syslog, mais pas tous, en particulier lorsqu'on utilise les anciens périphériques. Par conséquent, les interruptions et les notifications SNMP permettent d'obtenir des informations sur les événements des périphériques que nous n'aurions pas pu collecter.

## 2.3. Windows Event Log [33]

Format de journalisation propriétaire de Microsoft. Microsoft a décidé il y a longtemps d'inventer son propre système de collecte et d'enregistrement de journaux. Ce système s'appelle le journal des événements. Il a évolué au fil des ans et existe depuis presque aussi longtemps que Windows. Le journal des événements d'aujourd'hui dispose de fonctionnalités avancées. Le journal des événements est utilisé pour collecter et examiner principalement deux Types de journaux : les Journaux Windows et Les Journaux d'application.

# 3. TECHNOLOGIES DE STOCKAGE DESLOGS

Nous allons aborder un certain nombre de formats de stockage de journaux. Les périphériques réseau, applications et systèmes d'exploitation produisent une multitude de formats différents et dans de nombreux cas, les journaux seront stockés dans divers formats : texte, binaire ou compressé [33].

## 3.1. Fichiers textes plats

Les fichiers texte plats, sont à bien des égards, des fichiers sans schéma qui peuvent suivre un modèle commun ou être de forme libre. Généralement, un système crée un nouveau fichier journal et continue d'ajouter des données tant qu'il y a de l'espace libre ou jusqu'à ce qu'un processus système, tel que logrotate (rotation de log), indique au système de commencer un nouveau fichier journal et d'archiver celui-ci.

Ce format a tendance à être dans l'ordre chronologique avec les événements les plus anciens au début du fichier et dans l'activité la plus récente à la fin de celui-ci.

## 3.2. Fichiers logs binaires

Comme leur nom l'indique, ce sont des fichiers journaux illisibles ou destinés pour les machines. Ils sont générés par les applications qui nécessitent des outils spéciaux pour les lire et les traiter. Voici quelques exemples de fichiers journaux binaires les plus courants : les journaux IIS (Microsoft Internet Information Server) et les journaux des événements Windows.

## 3.3. Bases de données

L'un des principaux avantages de l'utilisation d'une base de données pour la conservation des journaux est la facilité avec laquelle on peut utiliser des requêtes SQL standard pour rechercher et récupérer rapidement des enregistrements de journaux.

L’inconvénient majeur est que l'écriture de données dans la base sera considérablement plus lente que l'écriture dans un fichier texte local sur le disque en raison de la latence du réseau, de l'analyse SQL de la base de données, des mises à jour d'index et de la validation des informations sur le disque. Les besoins en espace disque pour le stockage des journaux seront également plus importants dans une base de données en raison du nombre de fichiers d'index nécessaires pour effectuer une recherche et une extraction rapides, mais avec des options disponibles limitées.

## 3.4. Bases de données réparties

La recherche sur des milliards et des milliards de lignes peut devenir lente et fastidieuse. C’est pourquoi de nombreux systèmes de base de données prennent en charge le partitionnement. Le partitionnement permet à une table de base de données logiquement unique d'être divisée en fragments plus petits, avec les données de journal. Partitionner une table de base de données en fonction de la date et de l'heure est une approche logique qui offre les avantages suivants :

- La vitesse d'insertion des données est améliorée.
- Les performances des requêtes peuvent être améliorées pour certaines requêtes car de petits fragments de données sont examinés et filtrés par le système de base de données.

  La suppression en bloc peut être réalisée en supprimant une partition de données lorsqu'un ensemble de données de journal dépasse la période de rétention. Cela améliore considérablement la destruction des données de journal lors de suppressions individuelles dans une seule base de données volumineuse.
- Certains systèmes autorisent la migration de partitions rarement utilisées vers des options de stockage moins coûteuses et plus lentes.

# 4. LES ATTAQUES CONTRE LES SYSTEMES DE JOURNALISATION

Il existe diverses raisons pour lesquelles des attaquants pourraient cibler les journaux. Tout d'abord, l'attaquant intelligent veut généralement éviter de se faire prendre. Comme les données du journal fourniront la preuve de son activité, il voudrait empêcher que les informations ne soient retrouvées en modifiant ou en supprimant ces journaux. Un attaquant encore plus intelligent peut vouloir cacher ses traces en induisent l'observateur à penser qu'il se passe autre chose.

En plus de masquer les pistes, les attaquants peuvent trouver des informations dans les journaux qui sont utiles en elles-mêmes, ou des informations pouvant aider à attaquer d'autres systèmes, tels que des mots de passe ou des numéros de compte.

Les attaques contre l'infrastructure de journalisation peuvent être effectuées à tout moment et peuvent cibler n'importe quelle entité de l'infrastructure [33]:

La source : le ou les hôtes sur lesquels les messages de journal sont générés.

Lors du transfert du fichier journal : au niveau du réseau entre les sources et l'hôte-journal, ou entre le traitement et le stockage.

Le collecteur ou l'agent qui collecte les journaux : là où sont collectés les journaux de différentes sources.

Le stockage des données de journalisation : le système de base de données sur lequel sont stockés les informations de journalisation (si l'on utilise une base de données à cette fin). Cela peut également s'appliquer à n'importe quel mécanisme utilisé pour l'archivage des données de journal.

Au moment de l'analyse : le système où l'analyse est effectuée et le processus d'analyse lui-même.

Enfin, on peut s'attaquer à la partie inévitable de toute gestion de journaux : un analyste humain examinant des données et prenant des décisions.

Dans ce qui va suivre nous allons explorer les attaques qui peuvent être menées sur les systèmes de journalisation ainsi que les mesures de protection qui peuvent être prises pour les en empêcher.

### 4.1. Attaques contre la confidentialité

L'attaque contre la confidentialité peut être basée sur la collecte des renseignements confidentiels par l'attaquant.

Confidentialité à la source : le moyen le plus simple est d'obtenir des données de journaux et d'avoir accès à ces données là où elles sont générées. Des autorisations excessives ou un accès indirect peuvent permettre à un attaquant de lire les données du journal. L'exemple le plus simple est que le fichier journal soit lisible par tout le monde. Exemple :

```
#ls -l /var/log/messages
-rw-r--r-- 1 root root 327533 Oct 9 15:25 /var/log/messages
```

Confidentialité lors du transfert des fichiers journaux : Si l'on transfère des données de journal vers un hôte de journal central, un attaquant disposant d'un accès au chemin réseau entre la source et l'hôte de connexion pourrait intercepter les données lors du transfert. Ce type d'attaque peut être effectué sur l'un des points d'extrémité, avec un accès au périphérique réseau sur l'un des hôtes ou par «détection» (trafic d'interception) sur l'un des segments du réseau entre les deux hôtes.

Exemple simple de détection du trafic syslog à l'aide de tcpdump:

```
# tcpdump -s 1600 -w syslog-dump.dmp src host 172.16.90.128 and proto
   UDP and port 514
```

Cette ligne collecte tous les paquets UDP du protocole de datagramme utilisateur sur le port 514 (le port syslog) à partir de l'hôte 172.16.90.128 et les réécrit dans le fichier syslog-dump.dmp.

Confidentialité chez le collecteur (loghost) : tous les problèmes liés à la protection de la confidentialité des données à la source s'appliquent au loghost. De plus, comme le loghost ne doit normalement être utilisé que pour collecter et éventuellement analyser les données de journal, on peut restreindre les comptes de l'hôte de journalisation aux seuls utilisateurs ayant besoin d'accéder aux données du journal, ou d'administrer l'hôte de journalisation.

Pour certains environnements, un loghost furtif peut être souhaitable. Un hôte de journalisation furtif n'est pas visible sur le réseau, mais les hôtes de ce réseau peuvent toujours lui transférer leurs journaux.

Confidentialité de la base donnée (Log Store) : Le serveur de base de données doit être considéré comme aussi critique que le loghost et protégé de la même manière.

La plupart des bases de données fournissent des moyens de contrôler l'accès à la base de données, à la fois par l'hôte et par l'utilisateur. MySql utilise le même mécanisme pour les deux, en se servant de l'instruction GRANT:

```
mysql> GRANT INSERT, UPDATE PRIVILEGES ON logdb TO
logwriter@loghost.example.com;
mysql> GRANT SELECT PRIVILEGES ON logdb TO logreader@analsysis-station.
   example.com;
```

Cet exemple permet à un utilisateur, *logwriter*, d'écrire dans la base de données et uniquement à partir de l'hôte log. Un autre utilisateur, *logreader*, est autorisé uniquement à interroger la base de données à partir de la station d'analyse des fichiers journaux, mais pas à apporter des modifications. Notons que l'utilisateur de *logwriter* ne peut pas interroger les données, mais seulement insérer et modifier des données.

La confidentialité à l'analyse : les systèmes d'analyse ont besoin d'accéder aux données de journalisation. Ils constituent donc également un point d'attaque pour l'accès aux données de journalisation et pour les informations sur la manière dont les journaux sont analysés. Comme pour les bases de données et les hôtes de journal, la station d'analyse doit être protégée en limitant l'accès aux seuls utilisateurs des outils, tout en vérifiant les autorisations de fichiers et en chiffrant le trafic.

### 4.2. Attaques contre l'intégrité du système de journalisation

Une attaque contre l'intégrité du journal est la possibilité de corrompre des données réelles en écrasant ou en insérant de fausses données, ou en supprimant des données. Un attaquant corrompt souvent des données afin de dissimuler des preuves de son activité. L'approche la plus courante consiste simplement à supprimer les données en question. Un attaquant plus malin pourrait supprimer uniquement le message du journal indiquant son activité

sur le système ou tout simplement modifier les messages pour qu'ils semblent anodins et n'attirent pas l'attention. Un autre type d'attaque consiste à créer une distraction en insérant des messages factices indiquant qu'une autre activité est en cours, ce qui peut amener à négliger son activité. Les fichiers journaux ne constituent pas de preuve si l'attaquant arrive à démontrer que l'intégrité des journaux n'était pas sécurisée.

Intégrité à la source : Comme pour les problèmes de confidentialité que nous avons déjà explorés, des autorisations inappropriées peuvent permettre à un attaquant de modifier directement les données du journal à mesure qu'elles sont stockées. Par exemple, disons qu'un utilisateur «anton» sur un système fait quelque chose de néfaste et ne veut pas que l'activité lui soit associée. Si anton peut modifier les fichiers journaux où ils sont stockés, il pourrait remplacer toutes les occurrences de «anton» dans les journaux par «marcus», en transférant la responsabilité de l'activité sur quelqu'un d'autre. Cela peut être facilement fait ceci :

```
% sed 's/anton/marcus/g' < /var/log/logfile > /var/log/logfile.x; /bin/
   mv /var/log/logfile.x /var/log/logfile
```

Les fichiers journaux des évènements sont en binaire, ce qui rend l'attaque précédente plus difficile, mais un simple outil de conversion pourrait surmonter cette difficulté puisque le format du fichier est assez documenté.

Une solution possible est que les journaux soient transférés vers un hôte central permettant alors de créer un autre ensemble de journaux qu'il faudrait modifier pour que l'attaquant puisse cacher efficacement ses traces.

Un utilisateur ordinaire peut accidentellement supprimer les fichiers journaux. Une solution possible sous Unix, est de configurer les fichiers pour être append-only (ajoutés uniquement). Ainsi, seuls les utilisateurs privilégiés peuvent supprimer les fichiers logs.

La génération de faux messages log est un problème difficile à résoudre. Il est possible de supprimer le système de journalisation mais l'attaquant pourrait apporter sa propre copie.

Notons par ailleurs qu'il existe aussi des attaques de synchronisation.

Intégrité lors du transfert : les messages Syslog Unix standard sont transportés via UDP. Il n'y a pas d'authentification des adresses IP dans les

en-têtes de paquets UDP, et le serveur n'envoie aucun accusé de réception au client. Un attaquant ayant accès au réseau peut injecter des messages Syslog falsifiés semblant provenir d'hôtes légitimes sur le réseau. A titre d'exemple l'utilisation de l'outil Packit qui permet de créer des paquets réseau arbitraire. Il est aussi possible d'intercepter et de modifier le trafic IP («pirater la session») en utilisant une attaque par l'homme-au-milieu Spoofing ARP. La meilleure défense contre toutes ces attaques basées sur le réseauconsiste àutiliser un mécanisme qui vérifie l'intégrité des messages.
Le protocole syslog-ng fournit ce mécanisme.

L'intégrité chez le collecteur : les attaques d'intégrité sur le collecteur sont fondamentalement les mêmes que les attaques à la source et les mêmes techniques de protection s'y appliquent. L'impact de telles attaques, cependant, est supérieur à l'impact sur une source unique, car l'intégrité de toutes les données collectées par le serveur est en jeu.

Intégrité de la base de données : la base de données soulève deux défis majeurs en matière de protection des données selon leurs caractéristiques, à savoir «données en mouvement» et «données en attente». Les premières font référence aux données au fur et à mesure de leur arrivée dans la base de données, tandis que les secondes font référence aux données stockées dans la base de données. La sécurité des bases de données est un vaste domaine de connaissances et nous n'aborderons que brièvement certains des défis liés à la protection des bases de données avec celles des données de journal. Les données en mouvement sont généralement protégées en chiffrant la connexion de l'application à la base de données. L'exemple à citer ici est l'utilisation de «SQLS» pour transférer des commandes SQL et des données via SS, mais il n'est pas très utilisé. Le cryptage de base de données est le seul moyen fiable de protéger les données stockées. Peu de bases de données prennent en charge le cryptage de manière native et des logiciels tiers sont alors requis.

Intégrité à l'analyse : les atteintes à l'intégrité de l'analyse cherchent à miner la confiance dans la précision du système d'analyse. Il existe trois situations possibles pour l'attaque de l'intégrité du système d'analyse :

- les données en cours de traitement (l'entrée) : en modifiant les données en entrée. L'attaquant peut amener le système à signaler des fausses attaques ou à ne pas signaler les attaques réelles. La méthode utilisée par le système d'analyse pour accéder aux données doit comporter des mécanismes de protection de l'intégrité. A titre d'exemple pour un serveur de fichiers, le transport doit être sécurisé contre le détournement de session et la corruption de flux de données.
- les outils utilisés pour l'analyse (le système) :en ciblant le système d'analyse lui-même, un attaquant pourrait modifier les outils d'analyse afin que le système se comporte différemment que prévu. Le système d'analyse doit avoir un accès limité aux seuls utilisateurs autorisés et le logiciel utilisé pour l'analyse doit être protégé de manière appropriée contre toute modification. Il ya lieu d'envisager des privilèges distincts pour les utilisateurs du logiciel par rapport à ceux qui peuvent le modifier. Un contrôle de révision ou un autre logiciel doit être implémenté pour suivre les modifications apportées au logiciel (y compris les fichiers de configuration).L'intégrité du logiciel doit être vérifiée régulièrement.
- la présentation des résultats (la sortie) :enfin, le système utilisé pour présenter les données peut être altérée pour modifier ou supprimer l'affichage des données à l'analyste. La sécurité des données en sortie doit être égale à celle des données en entrée ; ce qui permet de limiter les personnes ayant un accès en écriture et celles ayant accès au système.

### 4.3. Attaques sur la disponibilité du système de journalisation

Les attaques sur la disponibilité visent à empêcher les utilisateurs légitimes d'accéder aux données ou au système. Les attaques contre la disponibilité peuvent être considérées comme un type d'attaque par déni de service. Dans ce cas, «service» peut signifier ressources humaines ainsi que ressources informatiques.

Disponibilité à la source : le déni de disponibilité le plus simple est la suppression des données de journal à la source. Exemple :

```
# killall syslog
# rm ~/.bash_history
# ln -s ~/.bash_history /dev/null
# zap /var/log/messages
```

La première commande consiste à supprimer le démon syslog, indiquant qu'il souhaite arrêter toute journalisation distante éventuelle. Il supprime le fichier .bash_history de la racine pour détruire l'enregistrement de toutes les commandes qu'il a tapées. Il installe ensuite un *rootkit* qui écrase par zéros les données pertinentes dans le fichier / var / log / messages. La remise à zéro des données, par opposition à leur suppression, écrase les données stockées sur le disque, rendant leur récupération impossible.

Un autre exemple si un attaquant sait que les journaux sont soumis à une rotation lorsqu'ils atteignent une taille particulière, il peut détruire les preuves de son activité en générant des messages de journal afin de générer suffisamment de rotation pour supprimer les preuves de son activité.Les méthodes permettant d'atténuer les abus de rotation consistent simplement à disposer de suffisamment d'espace disque.Les méthodes d'atténuation des attaques impliquant la suppression de données sont exactement les mêmes que pour les attaques d'intégrité.

Disponibilité lors du transfert : comme mentionné précédemment, le mécanisme de transport standard syslog utilise le protocole UDP, appelé couramment transport «non fiable». Un trafic réseau important peut entraîner le rejet de paquets UDP entre la source et le serveur. Un attaquant pourrait essayer de masquer ses traces en inondant le réseau de trafic, ce qui provoquerait l'abandon de paquets qu'il ne souhaite pas être découverts.

Un des moyens les plus simples d'inonder un réseau qui peut facilement passé inaperçu est le « ping flooding», exemple :

```
# ping -f loghost
```

« -f » indique au programme de sortir des paquets en nombre (pas moins de 100 paquets par seconde). L'inondation du réseau est difficile à atténuer, car il est difficile de différencier entre le trafic légitime et le trafic factice.

Disponibilité chez le collecteur (*loghost*) : en plus d'inonder le réseau, l'hôte de journal peut être inondé, soit avec de fausses données de journal, soit simplement avec un trafic factice.Les mêmes recommandations que pour les attaques d'intégrité s'appliquent ici.

Disponibilité à l'analyse : une attaque contre la disponibilité, remarquable mais souvent négligée, concerne la disponibilité de l'analyste. Un attaquant peut échapper à la notification en exécutant suffisamment d'attaques différentes ou en injectant des données pour donner l'impression que des attaques sont en cours.Il submerge ainsi l'analyste de la sécurité ou lui faire perdre du temps à examiner une autre activité qui semble plus sérieuse. La collecte et l'analyse des événements de journalisation doivent être organisées de manière à minimiser la capacité des personnes extérieures à injecter des données falsifiées. Généralement, la cryptographie aide à résoudre le problème. Cela résoudra, au moins, le problème de la falsification des données lors de la transmission de la source au point de collecte. Par exemple, remplacer syslog basé sur UDP par un syslog fiable mis en tunnel via SSL, SSH ou IPSec.

# 5. PRESENTATION DES TRAVAUXANTERIEURS

## 5.1. Forward Integrity For Secure Audit Logs [4]

Les deux chercheurs proposent une solution pour garantir l'intégrité du système de journal d'audit, qui consiste en l'utilisation d'un mécanisme de clé secrète ; point de départ d'une chaîne de hachage. La chaîne de hachage est générée par une fonction unidirectionnelle forte dans laquelle la clé est modifiée pour chaque entrée du journal comme montré dans la figure 5.1.1.

Un petit secret est établi au moment de la création du journal et stocké dans un endroit sûr (bout de papier enfermé dans un coffre-fort ou sur un ordinateur séparé et de confiance).

Le secret stocké sur l'ordinateur est le point de départ d'une chaîne de hachage, il est modifiée via une fonction cryptographique unidirectionnelle chaque fois qu'une entrée est écrite dans le journal. Ce secret est utilisé pour calculer un code d'authentification de message cryptographique (MAC) pour le journal chaque fois qu'une entrée est ajoutée, voire pour chiffrer le journal. Si le système est compromis, l'attaquant ne dispose d'aucun moyen pour récupérer les secrets utilisés pour créer les codes MAC ou les clés du déchiffrement des journaux.

Il peut effacer le journal entièrement, mais ne peut pas le modifier sans détection.

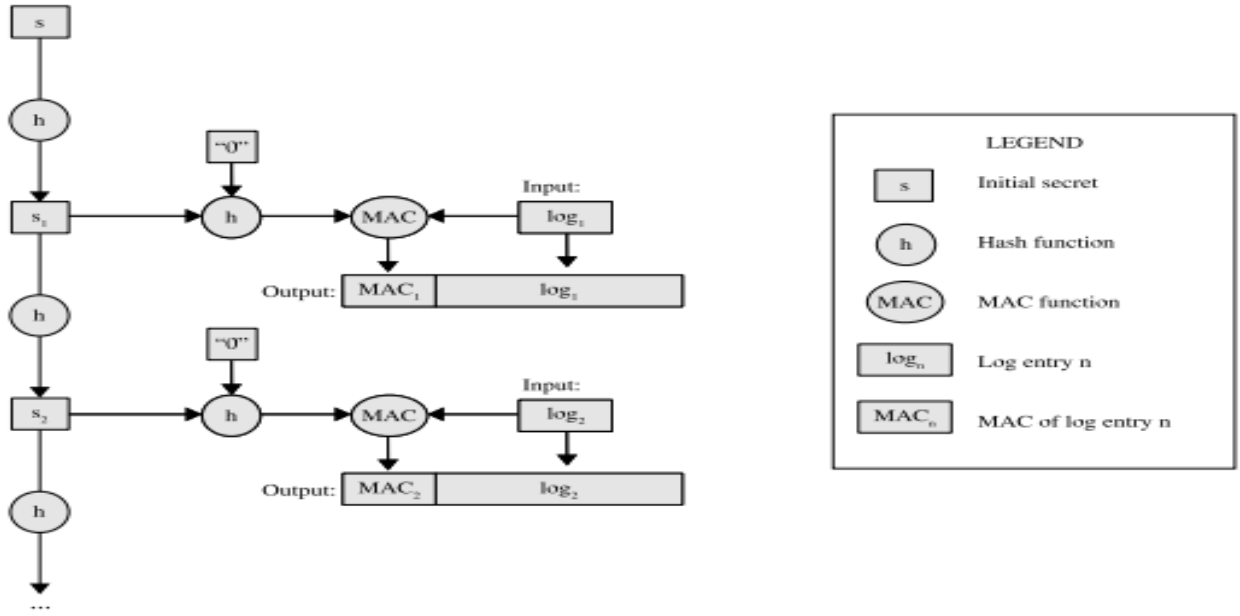



***Figure 5.1.2 : SECURITE DE TRANSFERT SIMPLE A L'AIDE DE MAC (MESSAGE AUTHENTICATION CODES)***

## 5.2.Logcrypt : Forward Security and Public Verification for Secure Audit Logs:[5]

Le principal inconvénient du système symétrique que nous venons de décrire est que la vérification d'un MAC nécessite la même clé que celle utilisée pour le créer. Cela signifie que toute personne ayant la possibilité de

vérifier une entrée de journal particulière pourrait créer des entrées alternatives arbitraires qui sembleraient également correctes.

La cryptographie à clé publique permet de séparer la signature de la vérification et le cryptage du décryptage.

Ces deux figure 5.2.1 et 5.2.2 montre comment séparer la signature de la vérification et le cryptage du décryptage. Le Sig remplace le MAC. Il a ajouté une méta-entrée qui répertorie les n-1 clé publique. Chaque entrée du journal est cryptée avec une clé privée. La dernière clé privée, Private n, est utilisée pour signer le méta-entry listant les n prochaines clés publiques. Pour faire la vérification, il suffit de décrypter le sig avec une clé publique et faire une comparaison avec l'entrée du journal. Si elles sont identiques, l'intégrité est alors vérifiée.

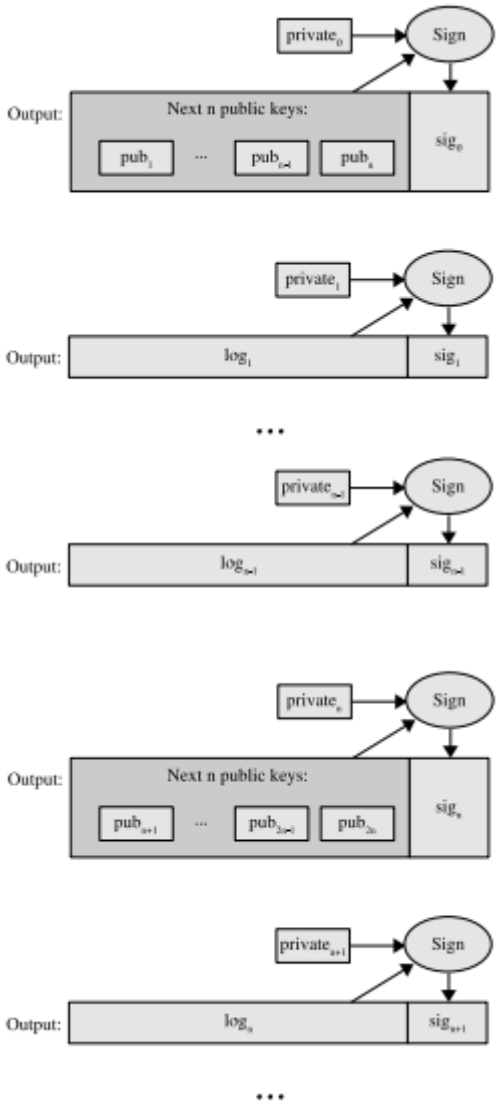



***Figure 5.2.1 : SECURITE AVANCEE AVEC VERIFIABILITE***

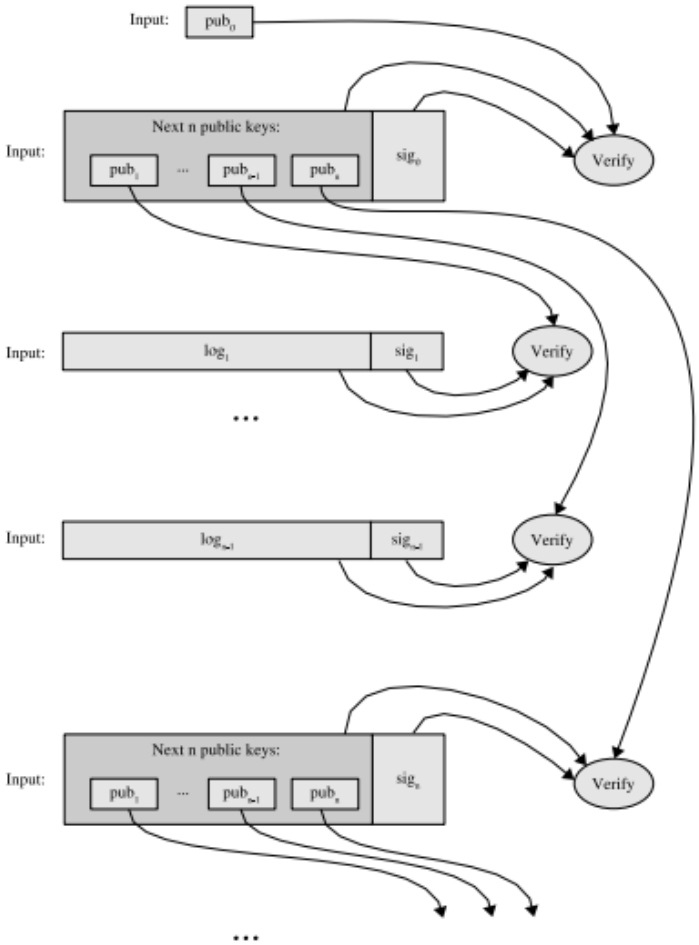



***Figure 5.2.2 : VERIFICATION DES ENTREES DANS LE SCHEMA A***

## 5.3. Secure Audit Logs to Support Computer Forensics [3]

L'architecture proposée par Bruce Schneier John Kelsey dans la figure 5.3.1 est composée de trois notions :

1. T est la machine de confiance. Cela peut généralement être considéré comme un serveur dans un emplacement sécurisé, bien que son implémentation puisse être mise en œuvre de différentes manières

2. U est la machine non fiable sur laquelle le journal doit être conservé.

3. V est un vérificateur moyennement digne de confiance, qui sera autorisé à examiner certains types d'enregistrement mais ne pourra pas effectuer de modification.

Les différentes notations utilisées dans le schéma Schneier Kelsey sont les suivantes :

1. Dj représente les données à saisir dans la jième entrée du journal.

2.Wj est le type d'entrée de journal de la jième entrée de journal. Ce type sert de masque d'autorisations pour V, T sera autorisé à contrôler les types d'entrée de journal auxquels un V particulier sera autorisé à accéder.

3.Aj est la clé d'authentification pour la jième entrée du journal. C'est le secret de base qui fournit toute la sécurité de ce schéma. Notez que U doit générer un nouveau A0 avant de lancer le fichier journal, A0 peut être donné à U par T au démarrage, ou U peut le générer de manière aléatoire puis le transmettre de manière sécurisée à T.

4. Kj = hash (\ Encryption Key ", Wj, Aj). Il s'agit de la clé utilisée pour chiffrer la jième entrée du journal. Notez que Wj est utilisé dans la dérivation de clé pour empêcher le vérificateur d'obtenir des clés de décryptage pour les types d'entrée de journal auxquels il n'est pas autorisé à accéder.

5. Yj = hash (Yj-1, EKj (Dj), Wj). C'est la chaîne de hachage que nous maintenons pour permettre aux utilisateurs partiellement approuvés, Vs, de vérifier des parties du journal via une connexion à faible bande passante avec la machine approuvée, T. Yj est basé sur EKj (Dj) au lieu de Dj afin que la chaîne puisse être vérifiée sans connaître l'entrée de journal.

6. Zj = MACAj (Yj).

7. Lj = Wj, EKj (Dj), Yj, Zj, où Lj est la jième entrée du journal.

8. Aj + 1 =(Increment Hash ", Aj).

Dans le schéma Schneier-Kelsey, une machine de journalisation U ouvrant un nouveau journal d'audit établit d'abord une clé secrète partagée A0 avec un serveur distant de confiance T. Une fois chaque entrée d'audit générée, la clé secrète actuelle Ai évolué en Ai+1 via une fonction unidirectionnelle. Les entrées de journal sont liées à l'aide d'une chaîne de hachage. Chaque entrée de journal Li contient trois parties : EKj(Dj), Yj, Zj.

U ferme le fichier journal en écrivant un message spécial d'enregistrement final Df et en effaçant Af ainsi que d'autres secrets, le cas échéant. Il n'y a pas de constant canal à bande passante élevée entre U et T. Il est supposé que U communique rarement les entrées de journal à T. Parfois, une personne ou une machine de confiance moyenne, appelée V, peut avoir besoin de vérifier ou de lire le journal d'audit, alors qu'il est toujours sur U. V reçoit de U une copie du journal d'audit, [L0; L1;……; Lf], où f est la valeur d'index du dernier enregistrement, à partir de U. V parcourt la chaîne de hachage dans les entrées du journal (les valeurs Yi), en vérifiant que chaque entrée de la chaîne de hachage est correcte. V envoie ensuite Yf et Zf à T, T sait A0 pour pouvoir calculer Af, cela lui permet de vérifier que Zf = MACAf (Yf).

T informe V du résultat de la vérification et V découvre si la copie reçue a été corrompue ou non.

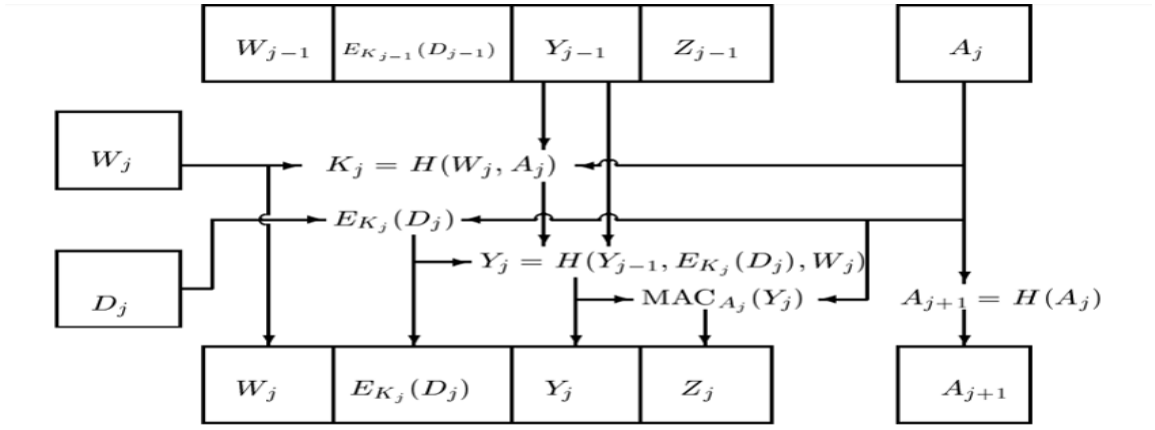



***Figure5.3.1 : SCHÉMA SCHNEIER-KELSEY***

## 5.4.A New Approach to Secure Logging[2]

Un schéma FssAgg comprend les composants suivants :

[FssAgg.Kg] - algorithme de génération de clé utilisé pour générer des paires de clés publique / privée.

[FssAgg.Asig] - algorithme de signature et d'agrégation qui prend en entrée une clé privée, un message à signer et une signature d'agrégation des autres signatures.

FssAgg.Asig est une procédure de mise à jour de clé FssAgg.Upd qui prend en entrée la clé de signature de la période en cours et renvoie la nouvelle clé de signature de la période suivante.

[FssAgg.Aver] - algorithme de vérification qui, lors de la saisie d'une signature globale putative, d'un ensemble de messages distincts supposément signés et d'une clé publique, génère une valeur binaire indiquant si la signature est valide.

### a) Schéma MAC FssAgg immuable

Initialisation du fichier journal : Avant que le système de journalisation ne démarre, nous exigeons que T soit accessible à U et supposons que U n'est pas (encore) compromis. U génère deux clés symétriques aléatoires, A1 et B1. Ensuite, il valide ces clés auprès de T, ainsi que les autres informations relatives au fichier journal spécifique et à l'intervalle de mise à jour de la clé UPD. Nous ne nous intéressons pas aux détails du processus d'engagement. Il suffit de dire qu'après le processus d'engagement, T peut passer hors ligne et U peut être déployé dans un environnement accusatoire et sans surveillance.

Pendant ce temps, U crée l'entrée de journal «fictive» initiale L1 qui valide un message fixe (par exemple, réglé sur «START») et calcule deux MAC sur L1 avec les clés A1 et B1, respectivement : UT 1 = macA1 (L1) et UV 1 = macB1 (L1). Ensuite, U fait évoluer ses clés au moyen d'une fonction à sens unique F: A2 = F (A1) et B2 = F (B1).

Grâce à l'interaction initiale, T sait que U a démarré un fichier journal à l'instant t avec les secrets initiaux A1 et B1. T stock ces valeurs dans sa base

de données et sait par la suite qu'un journal valide doit exister sur U et que ce journal doit contenir au moins une entrée de journal L1.

Le but de cette étape initiale d'engagement est d'empêcher une attaque par suppression totale.

L'extension d'immutabilité du schéma MAC *FssAgg* vérifiable de manière privée est très simple : U génère un MAC «fantôme» et le place comme premier composant de l'agrégat au démarrage du système ; ce MAC "fantôme" est ensuite effacé juste après l'agrégation.

Nous modifions l'initialisation du fichier journal (section 4) comme suit :

1. U calcule μ1 = macA1 (L1) sur un message d'initialisation fixe (connu) L1 ;
2. U évolue A1 en A2 ;
3. Lorsque la première entrée de journal réelle L2 apparaît, U génère μ2 = macA2(L2) et agrégats :

μV 2 = H (μ1 || μ2) ;

4. U stocke à la fois μ2 et μV 2 dans le fichier journal, convertit A2 en A3, efface de manière sécurisée μ1 et A2 et passe officiellement à la période suivante.

Notez que nous ne changeons rien à l'utilisation de B1 et des clés ultérieures Bi impliquées dans la génération de MAC pour T. En d'autres termes, seul le calcul des valeurs vérifiables par V est modifié.

Le format du fichier journal résultant de la prise en charge de l'immutabilité et de la vérification (efficace) individuelle des entrées de journal est le suivant :

{[L0, (L1, μ1), (L2, μ2), · · ·, (Lf, μf)], μV f, μT f}

Si V ne souhaite que vérifier une entrée de journal particulière Li, il utilise directement μi sans impliquer l'agrégat μV f. Cependant, notez que, pour vérifier Li, V doit toujours recalculer Ai qui nécessite i opérations de hachage.

### b) A Public-Verifiable Scheme

Dans ce schéma, nous n'avons plus besoin d'un serveur de confiance T. Au lieu de cela, nous avons besoin d'une autorité de certification (CA) capable de certifier / d'enregistrer la clé publique de U. La portée de V passe d'un petit groupe privé d'entités semi-fiables au domaine public, c'est-à-dire que toute personne disposant d'une copie du fichier journal peut le vérifier.

### c) Scheme Description

Un fichier journal authentifié dans le présent schéma se compose de deux parties : les entrées de journal [L1, · · ·, Lf] et une seule signature FssAgg, σ1 f.

Initialisation du fichier journal Pour créer un fichier journal, U utilise F ssAgg.Kg pour générer la clé secrète initiale sk1 et la clé publique pk. Ensuite, il enregistre pk auprès d'une autorité de certification publique. Le certificat du fichier journal de U doit contenir au moins ces informations essentielles telles que le journal créateur, l'ID de journal, l'heure de début et la clé publique. Par exemple, la signature de l'autorité de certification dans le certificat de U pour IDlog de fichier journal peut se présenter comme suit :

CERT (IDlog) = SIGNCA (U, IDlog, t, T, pk, horodatage, · · ·)

U garde sk1. Ensuite, il crée l'entrée de journal initiale L1 qu'il définit sur le certificat CERT (IDlog). Ensuite, U génère une signature σ1, 1 sur L1 avec F ssAgg.Asig en utilisant la clé privée initiale sk1. Enfin, U met à jour sa clé de sk1 à sk2 et efface de manière sécurisée toutes les copies de sk1.

Génération d'entrées de journal Avant que la i-ème entrée ne se produise, le fichier journal contient [L1, · · ·, Li − 1] et une signature *FssAgg* σ1, i − 1. La clé secrète actuelle de U est ski.

Fermeture du fichier journal : Comme dans le schéma de contrôle privé, U ferme le fichier journal en créant un message de fermeture spécial en tant qu'entrée de journal finale Lf, en mettant à jour la signature *FssAgg* en conséquence et en effaçant de manière sécurisée sa clé secrète.

Validation du fichier journal : Après avoir reçu une copie du fichier journal, V extrait les clés publiques de CERT (*IDlog*) contenues dans l'entrée de journal initiale L1 et V vérifie la signature de l'AC sur CERT (*IDlog*). Ensuite, V valide le fichier journal réel à l'aide de la fonction d'agrégation F *ssAgg.Aver*.

## 5.5. On the Security of Public Key Protocols [6]

L'utilisation de chiffrement à clé publique pour assurer la communication réseau sécurisé a suscité une attention considérable. Ils sont efficaces contre l'écoute passive, qui consiste à intercepter un message chiffré et essayer

ensuite de le déchiffrer a l'exemple des protocoles Needham Schroeder et Diffie Hellman…etc. Cependant, il est vulnérable face à un intrus actif, qui consiste à se faire passer pour l'un des utilisateurs ; soit l'émetteur, soit le récepteur du message voire modifié un message transmis.

L'auteur a établi des hypothèses de base sur lesquelles le system va être modélisé :

a) Un système à clé publique parfait :
    - Les fonctions à sens unique utilisées sont incassables ;
    - L'annuaire public est sécurisé et ne peut pas être falsifié ;
    - Tout le monde peut avoir accès à la fonction de cryptage $E_X$.
    - Seul X connait la fonction de décryptage D.
b) L'assistance d'un tiers en décryptage ou cryptage n'est pas nécessaire.
c) Protocole uniforme : le même format est utilisé par chaque paire d'utilisateurs souhaitant communiquer.
d) Prendre en considération seulement les intrus actifs :
    - Il peut obtenir n'importe quel message transitant par le réseau ;
    - Il est un utilisateur légitime du réseau et peut donc notamment engager une conversation avec un autre utilisateur ;

En prenant en considération ces hypothèses, l'auteur a défini deux modèles :

a) Les protocoles en cascade : Il s'agit de protocoles dans lesquels les utilisateurs peuvent appliquer les opérations à clé publique de cryptage-décryptage pour former des messages. L'auteur à prouver que le protocole en cascade est sur si et seulement si les messages transmis entre A et B contiennent toujours des couches de fonctions de chiffrement $E_B$ ou$E_A$. En générant un message de réponse, chaque participant A (A = X, Y) n'applique jamais D (la fonction de déchiffrement) sans appliquer également E (la fonction de chiffrement). Il a proposé aussi un algorithme pour déterminer si le protocole en cascade est sûr.

b) name-stamp protocols: Il s'agit des protocoles dans lesquels les utilisateurs sont autorisés à ajouter, supprimer et vérifier les noms cryptés

avec le texte en clair. Un autre algorithme a été proposé pour déterminer si le protocole name-stamp est sûr, avec un temps polynomial.

## 5.6. The BSD (Berkeley Software Distribution) Syslog Protocol[10]

Jusque dans les années 1980, la journalisation des événements était principalement réalisée en écrivant des événements dans un fichier résidant dans un système de fichiers local ou dans un autre moyen de stockage d'événements local (par exemple, une mémoire tampon en anneau basée sur la mémoire). Dans les années 1980, Eric Allman a proposé le protocole BSD syslog pour la collecte du journal des événements, initialement utilisé dans le système de mailing de sendmail, mais il a ensuite été progressivement accepté par la plupart des fournisseurs. Selon le protocole BSD syslog, les messages de journal sont envoyés sur le réseau sous forme de datagrammes UDP, chaque message étant encapsulé dans un paquet UDP séparé. Chaque message a une certaine facilité et sévérité, où facilité indique le type de l'expéditeur. Le protocole BSD définit 24 ressources et 8 valeurs de sévérité, comprises entre 0 et 23 et entre 0 et 7 respectivement.

Comme BSD syslog est basé sur UDP, il s'agit d'un protocole léger et efficace, qui consomme très peu de bande passante réseau et de ressources système. D'autre part, le protocole présente un certain nombre d'inconvénients qui sont résumés ci-dessous :

1) pas de support pour la transmission de messages fiable sur TCP.

2) pas de support pour le cryptage et l'authentification.

3) les horodatages ne sont pas assez spécifiques, il manque le numéro de l'année, les informations de fuseau horaire et des fractions de seconde.

4) À part le nom du programme d'envoi et son ID de processus, le champ MSG n'a pas de structure.

Afin de résoudre le premier problème, une variante TCP du protocole BSD Syslog a été proposée au cours de la décennie précédente, où un flux de messages au format BSD Syslog est envoyé via une connexion TCP.

Avec un caractère de nouvelle ligne (ASCII 10) faisant office de séparateur entre les messages. Cette variante de protocole permettait également d'utiliser d'autres utilitaires (par exemple, Stunnel) pour configurer des tunnels sécurisés pour la journalisation, et donc pour résoudre le second problème.

L'architecture introduit trois notions qui sont le collecteur dont l'objectif est de collecter les fichiers logs, terminale c'est-à-dire là où les fichiers log vont être crées, relai qui est un intermédiaire entre le terminale et le collecteur.

Il peut y avoir un relai par département dans une entreprise, qui authentifie tous les appareils du département, et qui à son tour s'authentifie auprès d'un collecteur général. Plusieurs canaux fonctionnant sous le même profil peuvent être ouverts entre deux homologues, avec les messages syslog de priorité supérieure, acheminés sur un canal auquel on donne plus de bande passante.

## 5.6. IETF(Internet Engineering Task Force) syslog protocole [42]

En 2009, le protocole syslog de l'IETF a été proposé pour résoudre les inconvénients du syslog BSD (voir [RFC5424-5426]). Le syslog IETF prend en charge la transmission sécurisée de messages sur TLS, mais également la transmission non cryptée sur UDP. En outre, il utilise un nouveau format de message avec des horodatages RFC3339 plus détaillés (voir [RFC3339]) et des blocs de données structurés. L'exemple suivant décrit l'exemple d'un message dans un nouveau format :

*<28>1 2012-11-17T12:33:59.223+02:00 myhost2 ids 1299 - [timeQuality tzKnown="1" isSynced="1"][origin ip="10.1.1.2"] port scan from 192.168.1.102*

La spécification de priorité <28> est immédiatement suivie du numéro de version du protocole (actuellement défini sur 1). En outre, l'expéditeur transmet deux blocs de données structurés [timeQuality tzKnown = "1" isSynced = "1"] et [origin ip = "10.1.1.2"] à l'expéditeur, le premier indiquant que l'horloge de l'expéditeur est synchronisée. Source d'heure fiable externe, la seconde indiquant l'adresse IP de l'expéditeur.

## 5.8. Safeguarding Cryptographic Keys [8]

Certaines clés cryptographiques sont importantes, exemple la clé qui permet de calculer l'exposant de décodage utilisé dans le cryptosystème RSA. Le dilemme c'est que si trop de copies sont distribuées, on peut en égarer une. Si trop peu de copies sont réalisées, elles risquent toutes d'être détruites. Dans un cryptosystème typique en plus d'utiliser des clés volatiles stockées dans des emplacements mémoire, sécurisés, mais qui risquent d'être détruites par un intrus, il est utile de confier une ou plusieurs clés non volatiles à des personnes fiables ou à des emplacements sécurisés. Les copies non volatiles ou les informations utilisées pour reconstruire les clés doivent être protégées contre trois types d'incident qui sont :

- Un incident d'abnégation est un événement après lequel une information non volatile n'est plus récupérable dans son intégralité par l'organisation. Les types d'abnégation sont :
    - Destruction de l'information non volatile. Par exemple, une personne portant une copie d'un numéro peut rencontrer un accident inattendu, au cours duquel la copie est détruite.
    - Dégradation de l'information non volatile. Par exemple, une personne peut perdre sa copie du numéro et, dans l'embarras et la confusion, générer un autre numéro à la demande.
    - Défection avec les informations non volatiles. Par exemple, la personne avec la copie du numéro peut le divulguer à l'opposition et refuser de le communiquer à l'organisation qui le lui a confié.
- Un incident de trahison est un événement au terme duquel une information non volatile est parfaitement connue de l'opposant de l'organisation, les types de trahison sont :
    - Défection comme expliqué précédemment.
    - Manquement (Déréliction) c'est le fait de révéler la clé non volatile à l'opposant sans que l'organisation ne le découvre. Exemple

continué à jouer le rôle d'un gradient fidèle après avoir révélé la clé non volatile à l'opposant.

- Un incident combiné est un incident d'abnégation qui est également un incident de trahison. Le principal type d'incident de combinaison est la défection.

L'auteur n'a pas pris en considération la simple perte de l'information non volatile car dans ce cas ni l'organisation ni l'opposant ne sont susceptibles d'obtenir l'information, l'information devient sans valeur.

Il y a deux méthodes pour compter les incidents :

La première est la loi d'inclusion et d'exclusion de Boole. Supposons qu'une organisation envoie des informations non volatiles aux gardes et attende une période de temps modeste au cours de laquelle des incidents se produisent parfois. Soit a le nombre d'incidents d'abnégation, b le nombre d'incidents de trahison et c le nombre d'incidents de combinaison. Le nombre total d d'incidents est $d = a + b-c$ car un incident combiné est compté deux fois, une fois par a et une fois par b.

Le deuxième principe est que les incidents sont si rares, que la possibilité que deux incidents distincts se produisent avec le même élément d'information non volatile est généralement considérée comme absurde pour des raisons probabilistes. Des deux principes de comptage ci-dessus, il s'ensuit que : $a+ b-MIN \{a,b\} \leq d \leq a+b$

L'homme prudent quand il conçoit un système de protection de clé doit prendre en considération que c peut être égale à 0 donc $d= a+b-c = a+b-0$. Le system doit générer $a+b+1$ information différente qui permet la reconstruction de clé et les distribuer aux différents gardes. La clé doit être reconstructible à partir de n'importe quel $b + 1$de ces pièces (mesure contre la destruction et la dégradation). Il ne doit y avoir aucune information sur la clé pouvant être déduite de la connaissance de seulement $b$ de ces pièces (il s'agit d'une protection contre les incidents de trahison).

## 5.9. Proactive Secret Sharing [45]

Les systèmes de partage de secrets protègent les secrets en les distribuant à différents endroits (détenteurs d'actions). La sécurité est assurée si, pendant toute la durée de vie du secret, l'adversaire est limité à une compromission inférieure à k (seuil) des n lieux. Pour les secrets sensibles et de longue durée, cette protection peut être insuffisante.

L'auteur propose un système de partage de secret proactif efficace, dans lequel les secrets partagés sont renouvelés périodiquement (sans changer de secret) de manière à ce que les informations obtenues par l'adversaire au cours d'une période donnée soient inutiles pour attaquer le secret une fois les actions partager renouvelées. Par conséquent, l'adversaire désireux d'apprendre le secret doit percer tous les k endroits au cours de la même période (par exemple, un jour, une semaine, etc.).

Afin de garantir la disponibilité et l'intégrité du secret, l'auteur a fourni des mécanismes pour détecter les secrets partagés corrompus de manière malveillante (ou accidentelle), ainsi que des mécanismes pour récupérer les secrets partagés lorsque la modification est détectée.

## 5.10. Secure Logging as a Service- Delegating log management to the cloud [9]

Architecture du système :

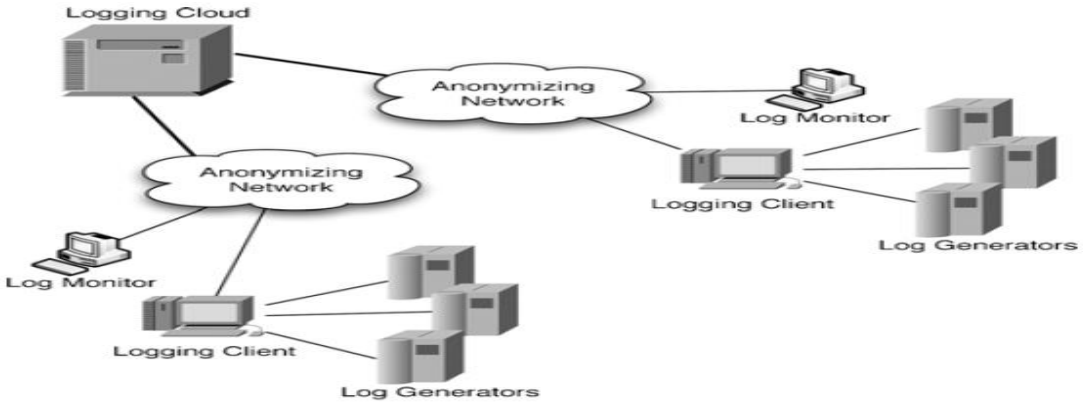



***Figure 5.1.1: ARCHITECTURE SYSTEME POUR LA JOURNALISATION SECURISEE DANS LE CLOUD***

Ce système comporte quatre principaux composants fonctionnels :

- Générateurs de journaux (Log Generators)
- Client de journalisation ou relai de journalisation (Logging Client) : le client de journalisation est un collecteur qui reçoit des groupes d'enregistrements de journal, générés par un ou plusieurs générateurs de journal et prépare les données de journal de manière à ce qu'elles puissent être transférées vers le cloud pour un stockage à long terme.
- Cloud de journalisation (Logging Cloud) : Le Cloud de journalisation fournit un service de stockage et de maintenance à long terme permettant de journaliser les données reçues de différents clients de journalisation appartenant à différentes organisations.
- log monitor (Log Monitor) : Il s'agit des hôtes utilisés pour surveiller et examiner les données du journal. Ils peuvent générer des requêtes pour extraire les données de journal du cloud. Sur la base des données de journal extraites, ces moniteurs effectueront une analyse plus approfondie si nécessaire. Ils peuvent également demander au nuage de journaux de supprimer les données de journal de manière permanente ou de faire pivoter les journaux.

Tous les messages du client de journalisation ou du moniteur de journalisation sont transmis au cloud de journalisation via le réseau Tor, ils ne peuvent pas être liés directement à leurs sources.

#### 5.10.1. Préparation du fichier journal pour le stockage sécurisé

La préparation du fichier journal se base sur le principe Schéma MAC FssAgg immuable, mais elle est adaptée pour cette architecture, le partage de secret se fait en appliquant le principe Proactive secret sharing**.**

#### 5.10.2. Téléchargement, récupération et suppression anonymes

Dans cette étape quatre protocoles différents sont développé pour le téléchargement, la récupération et la suppression anonymes de données de journal.

- Génération de balises de téléchargement anonyme
- Téléchargement anonyme
- Récupération anonyme
- Suppression anonyme

## 6. CONCLUSION

L'étude des travaux antérieurs à montrer qu'il existe trois types d'architecture possible pour la gestion des logs, qui sont une architecture centralisée, distribuée ou bien déléguée au cloud. L'étude a montré aussi que la gestion des logs consiste en quatre phases qui sont la génération, le transfert, la collecte et le stockage des logs.

Les logs sont critiques, car ils contiennent des informations relevant de la sécurité du système ainsi que des preuves d'activité malveillante. Ils peuvent donc être sujets à des attaques.

Des solutions ont été proposées pour assurer la sécurité des quatre phases.Dans deux des quatre phases (la première et la troisième)qui sont la génération et la collecte de logs,la solution consiste en l'attribution de signature pour chaque entrée log. Deux méthodes ont été proposées, l'utilisation d'un système de cryptage symétrique ou asymétrique. Ces deux méthodes présentent l'avantage de garantir l'intégrité et la confidentialité de ces entrées logs. Toute insertion, modification ou suppression du message log est difficile à réaliser sans être repérée. L'inconvénient avec la méthode symétrique est que la clé qui permet de faire la vérification de la signature est la même que celle utilisée pour la créer. Dans la méthode asymétrique par contre la signature est séparée de la vérification et le cryptage est séparé du décryptage.

La signature des fichiers logs implique la génération de clés qui doivent être gérées. Les travaux mentionnés précédemment ont proposé deux solutions qui sont le partage des secrets sans renouvèlement, ou, avec renouvèlement périodique. La première solution assure la sécurité si le nombre de secrets partagés compromis est inférieur à un seuil defini. Toutefois cette protection est insuffisante s'il s'agit d'un secret sensible et de longue durée. La deuxième solution pour parerà cette faiblesse consiste a renouveler le secret partagé périodiquement.

Dans la deuxième phase qui est le transfert des logs, deux protocoles ont été proposés : le protocole BSD-syslog (RFC 3162) basé sur UDP et le protocole IETF-syslog basé sur TCP. L'outil Syslog-ng propose l'utilisation de

SSL (Secure Socket Layer) pour un transfert sécurisé de la source à la destination.

Dans la quatrième phase qui est le stockage des logs, celui-ci représente la partie dite de confiance d'unsystème de gestion des logs, ou seuls les administrateurs ont accès afin d'y effectuer les analyses.

# CHAPITRE 2 : ETAT DE L'ART SUR L'ANALYSE DES LOGS.

## 1. ANALYSE MANUEL [33]

L'analyse des fichiers journaux nous permettra de vérifier la conformité à la réglementation, de prendre conscience de l'état du réseau, de mesurer la sécurité et d'apporter une réponse appropriée à un incident.

Une des techniques utilisée est de faire l'analyse manuellement, c'est-à-dire parcourir les fichiers journaux ligne par ligne et essayer de trouver des anomalies ou des manquements à la politique de sécurité mise en place par l'organisation. Toutefois, ce processus peut devenir vite fastidieux s'il s'agit de millions de lignes de journaux.

### 1.1. Outils utilisé pour l'analyse manuelle

Plusieurs outils peuvent nous aider dans l'analyse des journaux sur la plupart des versions Unix et Linux. Certains aident à visualiser les journaux en temps réel, tandis que d'autres aident à consulter l'historique. Exemple de visualisation en temps réel :

```
# tail -f /var/log/messages
```

Cette commande affiche les dernières lignes du fichier de messages Unix standard et les nouvelles lignes qui apparaissent après le lancement de la commande.

En ce qui concerne la visualisation de l'historique, n'importe quel éditeur de texte sur la plate-forme de notre choix peut être utilisé pour consulter les journaux de textes bruts. Lorsque les fichiers journaux sont volumineux, certains éditeurs de textes ne vont pas pouvoir les gérer efficacement (voire pas du tout).Exemple sous Windows, il existe un outil qui permet de consulter l'historique, il s'agit de « Event Viewer », il se trouve au niveau du panneau de configuration. Il permet :

- De parcourir les trois journaux Windows standard (Application, Systèmes et Sécurité).
- D'attendre de nouveaux événements.
- De les afficher au fur et à mesure de leur arrivée (en cliquant sur ''Actualiser'').
- De les filtrer et de les trier.

Dès les fichiers journaux stockés, nous pouvons les manipuler.

Sous UNIX, il existe des commandes qui permettent le filtrage pour ne garder que les éléments spécifiques. *Grep* en combinaison avec *tail,* ainsi que le Reformatage permettent de modifier la vue des fichiers journaux et de faire un résumé pour avoir une vue condensée.

Exemple : pour une recherche des enregistrements avec une chaîne "*ailed*" visant à intercepter les deux commandes "*Failed*" et "*failed*"(car*grep* est sensible à la casse) dans les 1000 dernières lignes d'un fichier journal :

```
# tail - 1000 /var/log/messages | grep ailed
```

## 1.2. Limites de l'examen du journal manuel

Les limitations évidentes incluent :

- L'examen manuel des journaux n'est pas adapté à l'augmentation de la taille des fichiers journaux. Des outils simples et une révision manuelle ne donneront probablement jamais une image complète de ce qui se passe dans l'environnement informatique.
- L'analyse nécessite souvent la mise en relation des journaux provenant de plusieurs sources. Une telle activité peut en effet être réalisée manuellement, mais le temps nécessaire augmentera considérablement (par rapport à un seul fichier journal). Une telle corrélation devrait être laissée aux outils automatisés.
- Lorsque le fichier est assez volumineux, explicite et vous dit que quelque chose de spécifique a échoué, il peut être interprété manuellement. Toutefois, dans le cas d'une source de journal non documentée, l'effort nécessaire pour interpréter chaque ligne passera d'une seconde de réflexion rapide à des heures de recherche en ligne et dans d'autres sources.

## 2. ANALYSE DESCRIPTIFS

Ici nous allons procéder à une comparaison entre les trois outils les plus populaires pour l'analyse des logs :

| | splunk> | | graylog |
|---|---|---|---|
| **installation** | Très simple d'installation : création d'un compte et récupération du fichier d'installation sur le site officiel de Splunk. | L'installation est plus complexe que Splunk mais reste relativement simple grâce à la documentation en ligne. | Installation similaire à ELK (Graylog utilise également ElasticSearch). Disponibilité d'une bonne documentation. |
| **Configuration** | Configuration simple qui se fait depuis l'interface Web (configuration de port d'écoute, ajout de données…) | Configuration plus complexe car il faut configurer Logstash (il faut donc maîtriser un minimum de langages de script) | Configuration simple et similaire à Splunk car elle se fait là aussi depuis l'interface web. |
| **Recherche** | Simple pour une utilisation basique. Il suffit de taper le mot clé recherché pour qu'il s'affiche en surbrillance. Recherche avancée basée sur la syntaxe de recherche SPL (Splunk Search Processing Language) | Simple également pour une basique utilisation. Similaire à Splunk (mot clé = en surbrillance). Syntaxe de recherche avancée basée sur la syntaxe Lucene. | Utilisation basique simple, similaire à Splunk et ELK. Syntaxe proche de Lucene. |
| **Graphiques** | Les graphiques se créent depuis la recherche et grâce aux champs disponibles. Graphiques facilement réalisables et très élaborés. | Ils se créent aussi depuis la recherche et les champs disponibles mais cela nécessite une bonne configuration de Logstash. On peut également créer les graphiques depuis le menu « visualise » en appliquant les filtres que l'on souhaite. Graphiques légèrement moins complets que Splunk. | Graphiques facilement réalisables depuis la recherche et les champs. (similaire à Splunk). Graphiques cependant moins complets que Splunk et ELK. |
| **Tableau de bord (Dashboard)** | Dashboard non interactif. Barre de recherche et temps non disponible par défaut. Il faut configurer les dashboards pour les rendre compatibles avec les visualisations ce qui peut être vite contraignant.<br>Possibilité de mettre un Dashboard en page | Dashboard interactif par défaut. Barre de recherche et barre de temps toujours disponibles. Les dashboards s'adaptent en fonction des termes de recherche ou de la plage de temps sélectionnée. Dashboard facile à créer et à modifier. | Dashboard facile à créer et à modifier mais ces deux aspects ne sont pas interactifs et la barre de recherche / temps n'est pas disponible ; Point faible de Graylog. |

| | | | |
|---|---|---|---|
| | d'accueil. | Point fort d'ELK. | |
| **Alertes** | Nécessite la version « Splunk Enterprise ». | Nécessite le « X-Pack » et donc la souscription à un abonnement. | Alertes disponibles gratuitement. Point fort de Graylog. |
| **Identification et gestion des utilisateurs** | Nécessite la version « Splunk Enterprise » pour créer des utilisateurs et gérer leurs droits. Possibilité d'intégration Active Directory / LDAP. | Nécessite le « X-Pack »pour bénéficier de la fonction d'identification et la gestion des utilisateurs. | Gestion des utilisateurs disponible gratuitement. Intégration Active Directory / LDAP possible également. Point fort de Graylog. |
| **Coût** | Version gratuite limitée à 500 Mo de logs/jour.<br>Nécessite également la version de Splunk Enterprise pour bénéficier des alertes, monitoring, support… | Open source sponsorisé par la société Elastic.<br>Nécessite l'achat d'une licence (X-Pack) pour bénéficier de toutes les fonctionnalités (identification, alerting, monitoring…) et du support. | Open Source. Possibilité de souscrire à un abonnement pour bénéficier d'un support. Dans ce cas, le prix varie en fonction de la quantité de données que l'on envoie à Graylog. |

***Tableau 2.1 : TABLEAU COMPARATIF DES OUTILS D'ANALYSE LES PLUS UTILISER***

D'après le tableau comparatif, nous remarquons que Splunk est un outil payant, tandis que ELK est gratuit, open source et ne traite pas seulement les fichiers logs. Cependant, pour avoir des fonctionnalités de sécurité, il faut souscrire à un abonnement X-Pack qui est payant.

Graylog est un outil dédié seulement à l'analyse des fichiers logs. Il est open source et entièrement gratuit. Il intègre les fonctionnalités de sécurité de manière native qui sont la configuration d'alertes, la gestion des utilisateurs et la possibilité de sécuriser l'interface web en activant le HTTPS.

## 3. **FILTRER, NORMALISER ET CORRELER** [33]

Dans cette partie nous allons parler de trois grands concepts : filtrage, normalisation et corrélation. Le filtrage consiste à collecter des données de journal brutes afin de déterminer si vous souhaitez les conserver. La sortie du filtrage est une donnée de journal normalisée. Ces données sont une entrée pour la corrélation. La corrélation est l'acte qui consiste à faire correspondre

une donnée unique normalisée, ou une série de données de pièces, dans le but d'entreprendre une action.

### 3.1. Filtrer

Après avoir reçu les fichiers logs à l'état brut, ces mêmes fichiers sont filtrés et ne sont gardés que ceux qu'on juge importants. Les autres peuvent être supprimés.

Marcus Ranum a inventé la phrase ''Artificial Ignorance'' (ignorance artificielle) en 1997. Ce concept repose sur un mécanisme permettant de rechercher des données de journal que nous connaissons afin que nous puissions trouver des choses que nous ne connaissons pas encore.

### 3.2. Normaliser

Les étapes permettant de normaliser un message de journal brut sont les suivantes :

1. Obtenir de la documentation sur le ou les produits que nous utilisons.
2. Lire la documentation pour obtenir une description de la nature des données de journal brutes et de la nature de chaque champ.
3. Proposer l'expression d'analyse appropriée pour normaliser les données.

La plupart des systèmes d'analyse de journaux utilisent une implémentation d'expression régulière pour analyser les données (il s'agit d'une chaîne de caractères, qui décrit, selon une syntaxe précise, un ensemble de chaînes de caractères possible).

4. Tester la logique d'analyse sur des exemples de données de journal brutes.
5. Déployer la logique d'analyse.

### 3.3. Corréler

La corrélation des journaux a pour objectif le regroupement de plusieurs événements similaires ou différents dans un seul élément de connaissance, par opposition à une vue incomplète de ce qui se passe en regardant des événements isolés. Ceci est accompli par la création de règles dans une sorte de langage, qui modélise des situations qui intéressent les administrateurs et

les analystes de la sécurité et du réseau. Par exemple, si nous recevons des données de journal à partir de pare-feu et d'IDS, nous pouvons capturer les tentatives de reconnaissance suivies d'une violation de politique de pare-feu avec la règle suivante :

```
If the system sees an event E1 where E1.eventType=portscan
followed by
an event E2 where E2.srcip=E1.srcip and E2.dstip=E1.dstip and
E2.eventType=fw.reject then
doSomething
```

Cette règle, exprimée en pseudocode, explique en détail comment deux événements disjoints peuvent être liés. La règle examine deux événements différents provenant de deux systèmes de sécurité réseau différents. Le premier est un balayage de port, qui serait détecté par IDS. L'événement suivant, noté E2, est généré par un pare-feu. Cela indique qu'une tentative d'accès à un hôte doté d'une stratégie de pare-feu exclut tous les hôtes approuvés, à l'exception de certains. Notez que la règle contient la phrase «followed by» (suivis de), cela ne signifie pas nécessairement que cet événement doit suivre immédiatement, bien queça pourrait l'être. Cela signifie que, à un moment donné, l'événement *fw.reject* doit se produire pour que la règle exécute l'action *doSomething* (Email, alerte, etc.).

# 4. PRESENTATION DES TRAVAUX ANTERIEURS

## 4.1. Méthodes de filtrage

Lorsqu'une anomalie survient dans un système et génère des avertissements ou des défaillances, ces anomalies ont tendance à apparaître jusqu'à ce que le problème soit résolu. Les événements du même type qui se produisent dans une courte période de temps n'introduisent éventuellement aucune nouvelle information.

La quantité d'informations redondante peut être filtrée pour réduire la charge lors des étapes de traitement ultérieures. De plus, dans les systèmes multi-nœuds, des avertissements ou des erreurs peuvent être générés à plusieurs endroits qui peuvent être filtrés.

Une approche similaire est appliquée par [18] et [19] qui, en fonction d'une valeur de seuil, utilisent l'intervalle entre les événements similaires pour déterminer les journaux à filtrer. De plus, [19] filtrent les messages similaires produits par plusieurs nœuds et sont capables de filtrer plus de 99,97% de ces messages.

## 4.2. Méthodes de préparation de fichiers journaux

La préparation de journal consiste à transformer les journaux bruts non structurés en événements spécifiques représentant un point de journalisation dans une application. Généralement, le contenu brut comprend une partie constante et une partie variable. La partie constante est le texte brut fixe de l'instruction de journal qui reste identique pour chaque message de journal correspondant à cette instruction.

La préparation de journal a pour objectif de séparer les parties constantes et variables du message de journal. Son résultat est communément appelé modèle de message ou événement de journal représentant le journal. Nous caractérisons quatre groupes différents de méthodes de préparation de journaux : méthodes heuristiques, méthodes de clustering, méthodes basées sur le code source et les autres méthodes.

### 4.2.1. Méthodes heuristiques

Une heuristique est une technique utilisée pour approcher une solution lorsqu'une solution exacte ne peut pas être trouvée dans un délai raisonnable. [12] utilise l'heuristique qui dit que les valeurs numériques représentent généralement des variables telles que les identificateurs de processus, les valeurs de paramètre et les adresses IP. Ils remplacent ces variables par des marques substitutives, pour réduire le nombre de messages d'un ensemble de données de télécommunication de 99% en un ensemble de modèles de journaux. [13] utilisent la même heuristique dans leur technique de regroupement qui divise les journaux en fonction du nombre maximal de mots constants dans une position donnée. Pour améliorer l'efficacité, ils accordent

une faible priorité aux valeurs numériques car ces jetons ont le plus de chances d'être variables. Outre les valeurs numériques, les messages de journalisation ont également tendance à contenir des symboles spéciaux utilisés pour présenter les variables. Ces symboles spéciaux incluent les deux points, les accolades et les signes égaux. Dans le message de journal «*Montant invalide pour la devise : EUR avec montant = 100*», les valeurs de variables trouvées seraient «*EUR*» et «*100*». [14] : Les messages de journalisation ont également tendance à contenir des symboles spéciaux utilisés pour présenter les variables. Ces symboles spéciaux incluent les deux points, les accolades et les signes égaux. Cette heuristique a permis de découvrir 95% des modèles de message journal. [15] utilisent une technique similaire dans laquelle ils considèrent que le signe égal est utilisé pour séparer une description statique d'une valeur variable. De plus, ils introduisent une heuristique qui reconnaît des expressions telles que *"[is|are|was|were] valeur*" pour capturer plus de variables.

### 4.2.2. Méthodes de clustering approche IPLoM :[29]

L'algorithme IPLoM est conçu comme un algorithme de clustering de données de journal. Cela fonctionne en partitionnant de manière itérative un ensemble de messages de journal. Les quatre étapes suivies par IPLoM sont les suivantes :

- Partition par nombre de jetons.
- Partition par position de jetons.
- Partition par recherche de bijection.
- Découverte des descriptions de cluster / formats de ligne.

A. Partition par nombre de jetons **:** La première étape du processus de partitionnement part du principe que les messages de journalisation ayant le même format de ligne auront probablement la même longueur de jeton. Pour cette raison, la première étape d'IPLoM utilise l'heuristique de comptage de jetons pour partitionner les messages du journal comme ce qui est monté dans la figure 4.2.2.1.

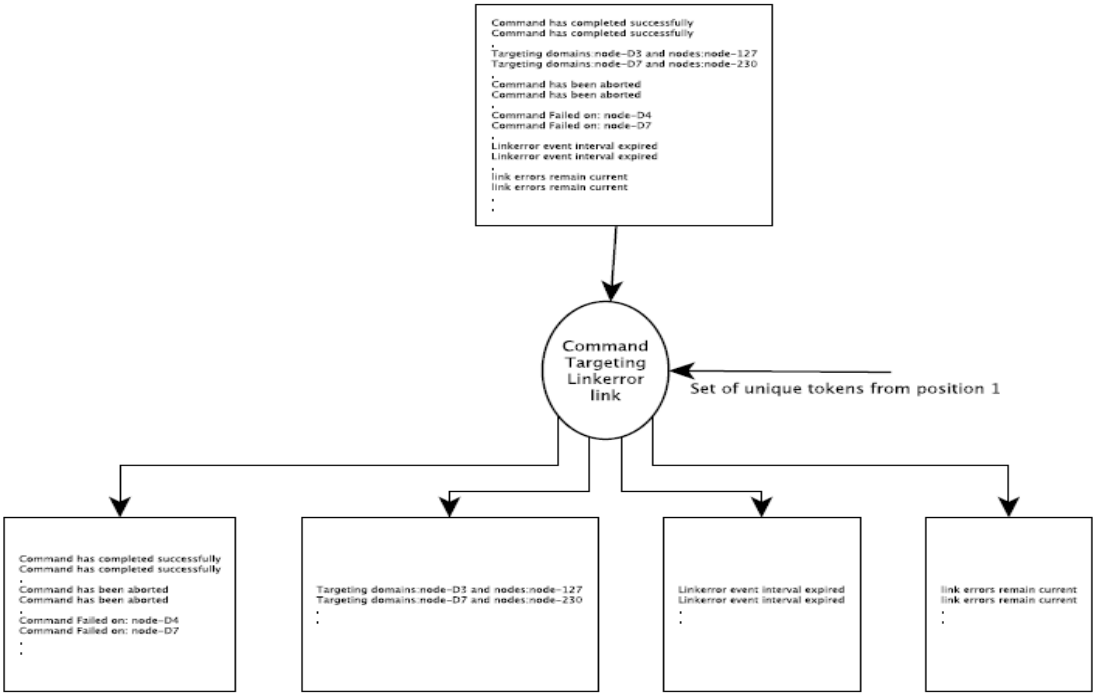



***Figure4.2.2.1 : PARTITION PAR POSITION DE JETON.***

B. Partition par position de jetons : À ce stade, chaque partition des données de journal contient des messages de journal de même longueur et peut donc être visualisée sous la forme de *n-uplets* ; n étant la longueur du jeton des messages de journal de la partition. Cette étape de l'algorithme part de l'hypothèse que la colonne contenant le plus petit nombre de variables (mots uniques) est susceptible de contenir des mots qui sont constants à cette position des formats de ligne qui les ont produits. Notre heuristique consiste donc à trouver la position du jeton avec le plus petit nombre de valeurs uniques et à scinder chaque partition en utilisant les valeurs uniques dans cette position de jeton. Chaque partition résultante ne contiendra donc qu'une de ces valeurs uniques dans la position de jeton découverte.

C. Partition par recherche de bijection : Lorsqu'il existe une bijection entre deux éléments des ensembles de jetons, cela implique généralement l'existence d'une relation étroite entre eux. Les messages de journalisation contenant ces valeurs de jeton dans les positions de jeton correspondantes sont alors séparées dans une nouvelle partition. Parfois,

les relations trouvées ne sont pas 1-1 mais 1-M, M-1 et M-M comme ce qui est montré dans la figure 4.2.2.2

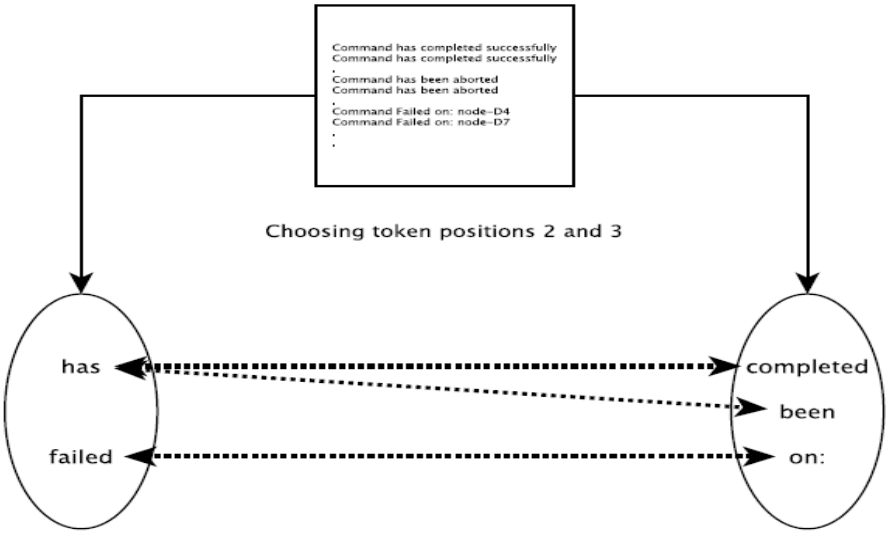



***Figure 4.2.2.2 : PARTITIONNEZ EN RECHERCHANT BIJECTION.***

Avant le passage des partitions par le processus de partitionnement de l'étape 3 de l'algorithme, elles sont évaluées pour voir si elles forment déjà de bons clusters. Pour ce faire, un seuil de qualité de cluster est introduit dans l'algorithme. Le seuil de validité du cluster est le rapport entre le nombre de positions de jeton ayant une seule valeur unique et la longueur de jeton totale des lignes de la partition. Les partitions dont la valeur est supérieure au seuil de validité du cluster sont considérées comme de bonnes partitions et ne sont plus partitionnées à cette étape.

D. Découverte des descriptions de cluster (formats de ligne) de chaque partition : Dans cette étape de l'algorithme, le partitionnement est terminé et nous supposons que chaque partition représente un cluster, c'est-à-dire que chaque message de journal de la partition a été produit en utilisant le même format de ligne. Une description de cluster ou un format de ligne consiste en une ligne de texte dans laquelle les valeurs constantes sont représentées littéralement et les valeurs de variables à l'aide de caractères génériques.

Cela se fait en comptant le nombre de jetons uniques dans chaque position de jeton d'une partition. Si une position de jeton n'a qu'une seule valeur, elle est considérée comme une valeur constante dans le format de ligne. Si elle est supérieure à une, elle est considérée comme une variable.

### 4.2.3. Méthodes basées sur le code source

Au lieu d'analyser les messages de journal pour obtenir un ensemble de modèles de message, il est possible d'analyser le code source pour atteindre le même objectif. Considérez l'instruction Java suivante du journal.

```
log.info("Machine " + m + " has state " + STATE);
```

Plusieurs difficultés surgissent dans l'analyse de cette déclaration. L'analyseur doit savoir quelle bibliothèque de *logging* est utilisée Par exemple, log.info (). Dans ce cas, STATE semble être une variable, mais m pourrait être une instanciation d'une classe d'ordinateur qui remplace la méthode par défaut *toString ()*. Cette méthode elle-même peut renvoyer une partie constante mélangée à des variables, qui sont plus difficiles à analyser. Ce dernier problème a tendance à se produire dans les langages orientés objet (par exemple, Java). [16] proposent l'utilisation de l'arbre de syntaxe abstraite ASA (*abstract syntax tree*, ou AST, en anglais), Leur analyseur de journal requiert la classe de journalisation en tant qu'entrée facultative. Ils prennent les modèles de message partiels des instructions de journalisation et les complètent avec des définitions *toString ()* et saisissent des informations sur la hiérarchie pour résoudre le problème décrit. Le résultat obtenu est l'appariement de 99,98% des messages de journalisation dans un modèle. De manière similaire, [17] utilisent des AST pour créer des modèles de message et extraire des variables de message.

## 4.3. Techniques et méthodes d'analyse des journaux

Les techniques et méthodes utilisées dans l'analyse des journaux ne s'appliquent pas uniquement à l'une des catégories d'analyse. Pour cette raison, nous présentons d'abord les techniques et les méthodes et expliquons comment elles peuvent être utilisées dans l'analyse des journaux.

### 4.3.1. Apprentissage de règles associatives

L'apprentissage de règles associatives est une technique d'apprentissage supervisé qui permet de découvrir des relations fortes entre variables [20]. L'idée clé derrière les règles d'association est que si un certain ensemble d'événements s'est produit, un autre événement est également susceptible de se produire. La technique a été introduite pour exploiter les relations entre les éléments d'une base de données, par exemple les produits d'épicerie. Un cas d'utilisation [20] donne toutes les règles qui ont pour conséquence le «coke diète», ce qui aide à planifier ce qu'un magasin devrait faire pour augmenter les ventes de coke diète.

Dans l'analyse des journaux, l'apprentissage par règles associatives peut être utilisé pour traduire des séquences d'événements en règles [21]. Par exemple, trois événements qui se produisent dans un intervalle de temps proche peuvent indiquer une probabilité élevée qu'un quatrième événement se produise. En exploitant les règles associatives à partir des événements de journal, il est possible de capturer le comportement d'exécution d'une application. De manière similaire, les règles associatives peuvent caractériser les défaillances récurrentes qui se manifestent via une signature d'événement répétitif comme montré dans la figure suivante.

| | | | |
|---|---|---|---|
| If | xxxxxxxx1 Warning from Unit A is Present AND<br>xxxxxxxx2 Error from Unit A is Present AND<br>xxxxxxxx4 Warning from Unit B is Present AND<br>xxxxxxxx4 Warning from Unit A is Present AND<br>xxxxxxxx5 Error from Unit A is Present | Then | Faulty |
| If | xxxxxxxx1 Warning from Unit A is Present AND<br>xxxxxxxx2 Error from Unit A is Present AND<br>xxxxxxxx4 Warning from Unit B is Present AND<br>xxxxxxxx6 Warning from Unit A is Present AND<br>xxxxxxxx7 Error from Unit A is Present | Then | Faulty |

***Figure 4.3.1.1 : EXEMPLE DE SIGNATURE***

### 4.3.2. Test du $x^2$ :[22]

Est un test d'hypothèse statistique simple pouvant être utilisé pour calculer la probabilité que deux vecteurs de données proviennent de distributions différentes. Une application dans l'analyse du journal du test du $x^2$ est la

détection d'anomalie. En testant si un échantillon de nouveaux événements de journal provient d'une distribution différente d'événements qui représentent le comportement normal du système, il est possible de déterminer si cet échantillon est une anomalie. La loi utilisée est $\sum_{i=1}^{j} \frac{(o_i - e_i)^2}{e_i}$ avec **o** évènement observé et**e** évènement espéré.

### 4.3.3. Apprentissage par arbre de décision [30]

Un arbre de décision est une structure récursive simple permettant d'exprimer un processus de classification séquentiel dans lequel un cas, décrit par un ensemble d'attributs, est affecté à l'un des ensembles de classes disjoints. Chaque feuille de l'arbre désigne une classe. Un nœud intérieur désigne un test sur un ou plusieurs attributs avec un arbre de décision subsidiaire pour chaque résultat possible du test. Pour classer un cas, nous commençons à la racine de l'arbre. S'il s'agit d'une feuille, le cas est attribué à la classe désignée, s'il s'agit d'un test, le résultat de ce test est déterminé et le processus poursuivi avec l'arborescence secondaire appropriée à ce résultat.

Cette approche peut classer les nouveaux cas même lorsque le résultat de tests cruciaux est inconnu. Il existe au moins deux types de circonstances dans lesquelles cette condition Xi doit être supprimée du côté gauche de la règle.

La première survient généralement avec des concepts disjonctifs [43] dans lesquels un cas appartient à une classe particulière chaque fois qu'une expression logique disjonctive de la forme Y ∪ Z est satisfaite. Un arbre de décision pour une telle tâche de classification peut commencer par un test pertinent pour Y mais non pour Z, de sorte que les feuilles associées à Z disjoint génèrent des règles prototypes contenant des conditions non pertinentes. Si Xi est une telle condition, son élimination produira une règle plus générale sans diminution de la précision.

Deuxièmement, la présence de Xi dans la partie gauche de la règle peut donner une plus grande précision apparente, mais cette précision peut provenir de caractéristiques fortuites (Qui arrivent par hasard, d'une manière imprévue).

La figure 1 montre un arbre décisionnel non trivial pour un aspect du diagnostic de maladie thyroïdienne, pour simplifier l'impression, l'arbre a été tourné sur le côté.

Les feuilles apparaissent en gras et les résultats possibles au niveau d'un nœud intérieur sont représentés par des expressions logiques avec une indentation égale comme montré dans la figure suivante.

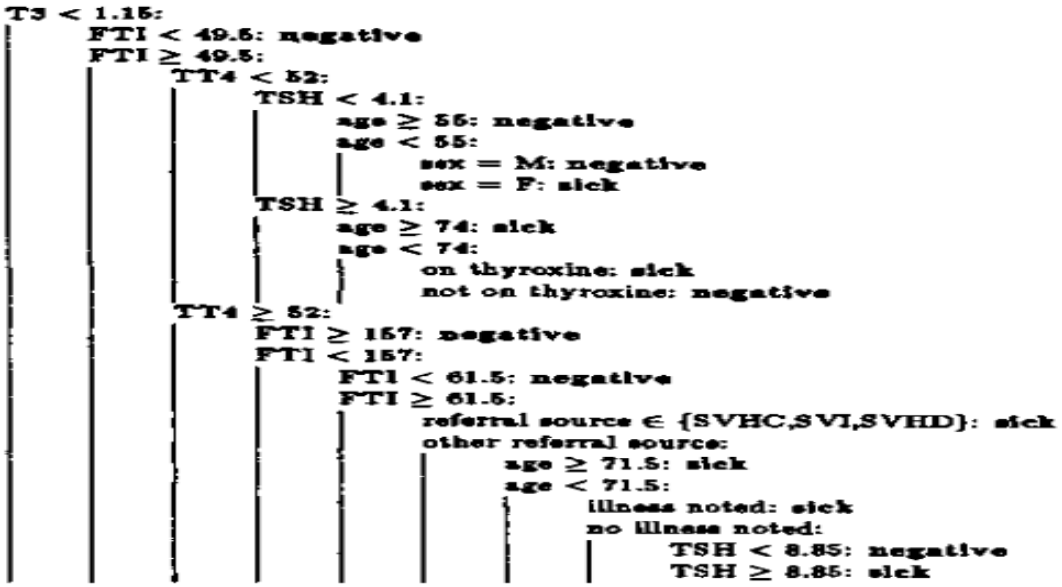



***Figure 4.3.3.1 : SIMPLE DECISION TREE***

Le lecteur peut se demander pourquoi nous utilisons l'arbre de décision au lieu de développer des règles directement à partir des cas de formation. Travailler à partir de l'arbre présente deux avantages majeurs. Les tâches de classification les plus intéressantes impliquent des attributs avec des valeurs continues qui doivent être transformés en tests par le développement de seuils appropriés. L'approche divisée pour mieux régner qui est couramment utilisée par les algorithmes pour la construction d'arbres de décision constitue un moyen puissant et sensible au contexte pour faire face à ce problème, par ailleurs complexe.

Deuxièmement, même un long chemin dans un arbre de décision n'implique généralement qu'une faible proportion des attributs possibles.

### 4.3.4. Classification hiérarchique

[23] C'est une technique d'apprentissage non supervisée. Elle est basée sur l'idée que les objets proches les uns des autres sont davantage liés que les

objets plus éloignés comme montré dans la figure 4.3.4.1, en fonction d'un ou de plusieurs attributs. Ce que sont ces attributs et comment ils sont calculés diffère d'un cas d'utilisation à l'autre. Cette technique est utilisée lorsque l'on ne veut pas émettre d'hypothèses sur le nombre de grappes, contrairement à des techniques telles que K-Means[24]. Il existe deux stratégies en matière de classification hiérarchique : regroupement par *agglomération* et regroupement par *division* [23].Le regroupement par *agglomération* est une approche ascendante dans laquelle chaque objet commence en tant que grappe et le nombre de grappes est réduit en combinant deux grappes. Le regroupement par *division* est une approche descendante dans laquelle tous les objets commencent sous la forme d'un cluster, qui est divisé à plusieurs reprises en plusieurs groupes. La classification est non seulement utilisée dans la préparation des journaux, mais elle a également une application dans l'analyse des journaux.

[25] proposent « *LogCluster* » ; une approche d'identification des problèmes basée sur le regroupement de journaux qui prend en compte toutes les caractéristiques des journaux des systèmes de service en ligne. Dans leur travail, ils affectent des poids pour consigner les messages et groupent les séquences de journaux similaires en clusters. Ils extraient ensuite une séquence de journal représentative de chaque cluster. Le fonctionnement de *LogCluster* peut être divisé en deux phases : la phase de construction et la phase de production. Lors de la phase de construction, ils utilisent les séquences de journaux collectées à partir de l'environnement de test et les regroupent pour constituer une base de connaissances initiale. Au cours de la phase de production, ils analysent les séquences de journaux collectées à partir de l'environnement de production réel, les mettent en cluster et vérifient si les clusters peuvent être trouvés dans la base de connaissances. De cette manière, le développeur n'a qu'à examiner un petit nombre de séquences de journaux représentatives des grappes précédemment invisibles. *LogCluster* réduit en conséquence davantage le nombre total de journaux à examiner manuellement et améliore l'efficacité de l'identification des problèmes.

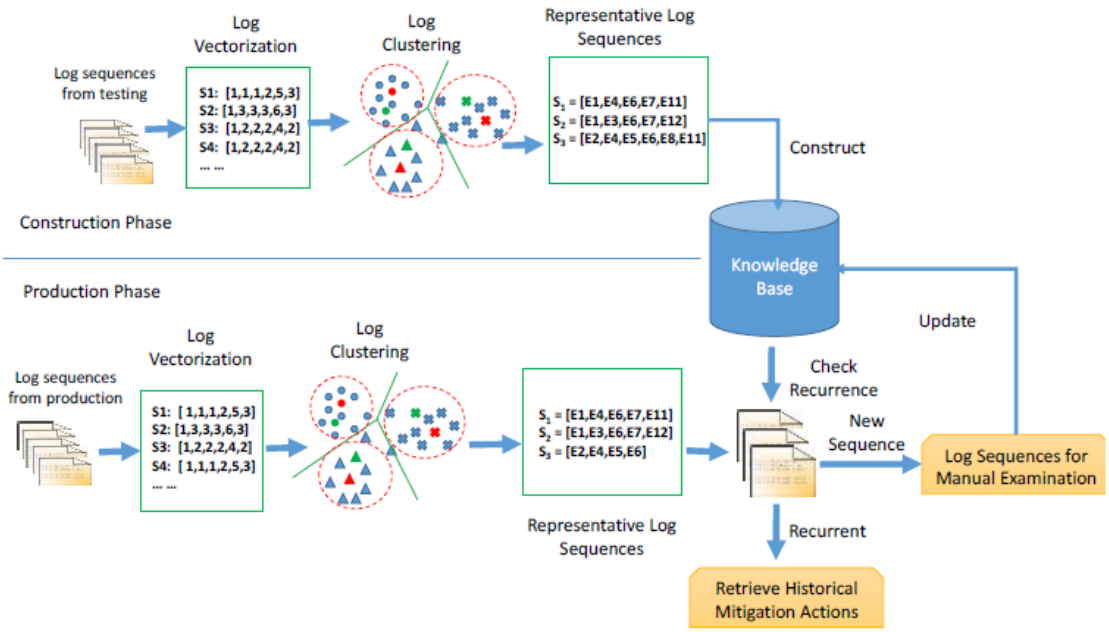



***Figure 4.3.4.1 : LA STRUCTURE GLOBALE DE LOGCLUSTER***

### 4.3.5. Naive Bayes

[26], une famille de classificateurs probabilistes simples, permet de classer en fonction de caractéristiques considérées comme indépendantes les unes des autres. Les applications de l'apprentissage supervisé vont de la détection du spam à la catégorisation. Un classificateur Naive Bayes considère que chacune des caractéristiques d'un ensemble contribue à la probabilité qu'un élément entre dans une classification. Le problème connu de Naive Bayes est que, même si les caractéristiques dépendent les unes des autres, elles seront considérées comme indépendantes. Son principal avantage est qu'il peut largement surpasser les algorithmes plus sophistiqués en termes d'exécution et que, malgré son inconvénient, il fonctionne bien dans la pratique. Son application à la détection du spam explique intuitivement sa pertinence dans la détection des anomalies [27]. Dans la détection d'anomalies, la probabilité exacte de savoir si quelque chose est une anomalie ou non, n'est pas cruciale ; une estimation brute est suffisante. Cela rend Naïve Bayes suffisamment robuste pour la classification des événements anormaux. Voici la formule : $P(C|X)=\frac{P(C|x)P(C)}{P(X)}$ avec **c** c'est la classe, **x** c'est la valeur du prédicteur et **p** c'est la probabilité.

### 4.3.6. Machine à vecteurs de support

Définition : Une machine à vecteurs de support (SVM) [28], utilisée pour la classification, est un modèle d'apprentissage supervisé qui, à partir d'un ensemble d'apprentissage, construit un modèle de prédiction. Sur la base d'un ensemble d'exemples étiquetés, il tente de trouver un moyen de séparer les données en deux groupes (problème linéaire). Ce modèle est utilisé sur les nouvelles données pour déterminer s'il convient de l'étiqueter comme l'une des deux catégories ; souvent le succès et l'échec. Son objectif de classer les données dans l'une des deux catégories fait que les SVM sont utiles pour la détection des défaillances.

Approche linéaire : Le cas est présenté par les n exemples d'apprentissage (Training Set Data) sous formes de pairs {Xi, Yi} pour i allons de 1 à n avec Xi∈R et Yi ∈ {-1,1}, les points rouges sont représentés pour Yi =1 et les points bleus pour Yi =-1 comme montré dans la figure suivante.

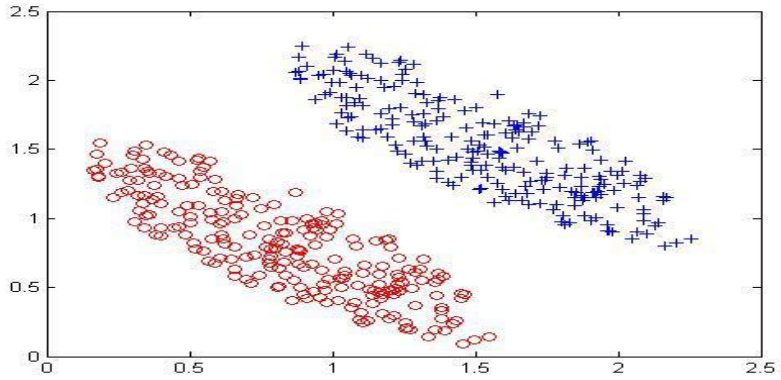



***Figure4.3.6.1 : DISTRIBUTION SPATIALE DES DONNEES D'APPRENTISSAGE***

La fonction classifiant linéaire (discriminante) est la suivante : $y(x)=w^t x + w_0$

Le $w_0$ représente le biais et x représente l'entrée et $w^t$ représente le vecteur de poids comme montré dans la figure suivante.

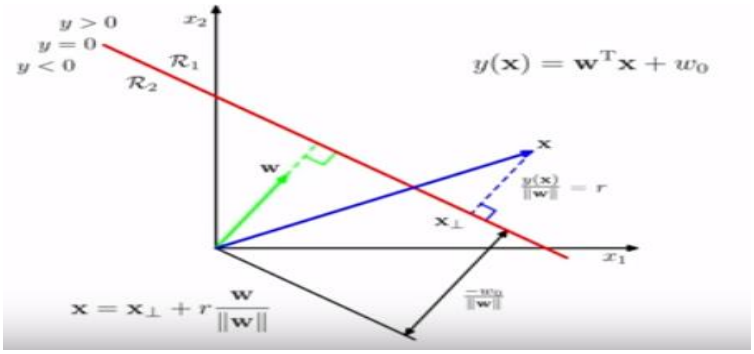



***Figure 4.3.6.2 :SEPARATION ENTRE 2 CLASSES PAR UN HYPERPLAN***

La droite qui sépare les données est appelée surface de décision, la fonction discriminante prend en entrée (x) et donne en sortie sa classe ($c_k$), si le résultat de la fonction discriminante y(x)>0, cela correspond à la classe ($c_1$), si y(x)<0 nous sommes alors dans la classe ($c_2$).

Pour calculer la distance entre un point (x) et la surface de décision t. $\frac{y(x)}{||w||}$ =r, la cible (t) sera égale à (t=1) si y(x)>0 sinon si (y(x)<0) alors (t=-1), (r) peut être appelé aussi la marge de la plus petite distance signée entre la surface de décision et les entrées de l'ensemble d'entrainement. La fonction de la marge est représentée comme suit :

arg max(w, $w_0$)={$\frac{1}{||w||}$ min[$t_n(w^t x + w_0)$]}

les valeurs recherchées sont (w, $w_0$).

Comme ce qui est montrer dans la figure suivante

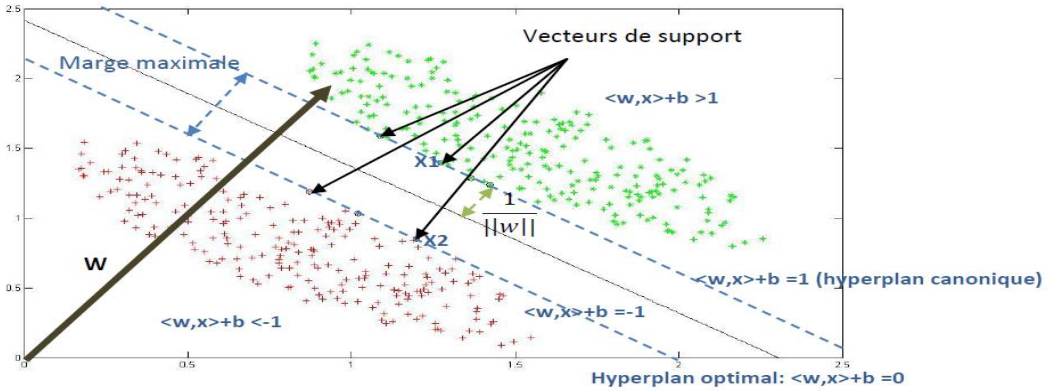



***Figure 4.3.6.3 : NUAGE DE DONNEES D'APPRENTISSAGE AVEC UN HYPERPLAN OPTIMAL***

# 5. AGIR SUR LES LOGS [33]

## 5.1. Logs critiques

Le tableau suivant, présente des exemples d'actions à entreprendre sur les journaux critiques. Notons que les actions du tableau ne sont pas fournies comme un « Guide de conduite » complet, mais pour illustrer le fait que les journaux peuvent souvent être traités.

| Fichier log critique | Action requise |
|---|---|
| Défauts pouvant affecter les opérations du système. | En cas de perte du fonctionnement du système, un système de sauvegarde doit être mis en place. |
| Des attaques réussies. Ou des attaques qui ont de grandes chances de réussir | Lancer le processus de réponse aux incidents et de récupération pour la machine compromise |
| Système atteignant la capacité ou le maximum d'une certaine valeur. | Augmenter la capacité ou risquer que le système tombe en panne. |
| Modifications du système pouvant entraîner des problèmes de sécurité et de disponibilité. | Annuler les modifications non autorisé pour éviter les problèmes. |
| Crash du système. | Un système de sauvegarde doit être mis en place. |
| De nombreuses connexions ont échouées. | Recherchez des signes qu'un attaquant à pu deviner le mot de passe. |
| Erreur matérielle. | En cas de perte du fonctionnement du système, un système de sauvegarde doit être mis en place. |
| Modification de la configuration liée à la sécurité. | Annuler les modifications non autorisé. |
| Connexion non autorisée détectée. | Enquêter sur qui a eu accès au système et pourquoi. |

***TABLEAU 5.1 : RESUME DES ACTIONS A ENTREPRENDRE SUR LES MESSAGES CRITIQUES DU JOURNAL***

## 5.2. Logs non critiques

Il existe plusieurs chemins allant des journaux apparemment non exploitables à l'action. Nous résumons ci-dessous certaines des plus courantes :

- Un grand nombre de messages ne donnant pas lieu à une action, se traduisent souvent par une action, en raison d'une tendance en développement ou de nouvelles connaissances déduites du résumé.
- Agir sur un événement corrélé ou un groupe d'événements : Souvent, une combinaison connue d'événements non critiques s'inscrit dans un schéma critique en raison de leur séquence et de leur synchronisation.
- Agir sur un modèle découvert ou sur une séquence inhabituelle d'événements non critiques : à l'aide d'outils d'analyse avancés de

journaux, il est parfois possible d'extraire un nouveau modèle suspect à partir d'enregistrements de journaux qui sont en apparence normaux et inoffensifs.

## 6. CONCLUSION

L'étude de l'existant montre que l'analyse des logs peut se faire de deux manières, manuelle ou automatique. L'analyse manuelle consiste à parcourir les fichiers logs ligne par ligne, mais ce processus peut se révéler fastidieux s'il s'agit de millions de lignes. L'analyse automatique au contraire permet de traiter beaucoup plus de fichiers logs et avec beaucoup plus de précision. Elle se divise en deux parties, une partie pré- analyse et une partie analyse. La partie pré-analyse consiste au filtrage et à la normalisation des logs. La partie analyse consiste à appliquer un algorithme d'apprentissage automatique sur les logs.

Plusieurs méthodes ont été proposées pour effectuer l'analyse. Le filtrage dans la partie pré-analyse consiste à supprimer les fichiers logs redondants. Pour la normalisation, les méthodes proposées consistent en la séparation des parties constantes du fichier log des parties variables.

En somme, les méthodes proposées pour effectuer l'analyse consistent en l'application d'algorithmes d'apprentissage automatique supervisé ou non supervisé.

De nombreux outils d'analyse de logs existent sur le marché. Certains d'entre eux sont payants d'autres sont gratuits. L'utilisation de ces outils par les entreprises et les organisations est différente, quand bien même certains outils sont plus flexibles que d'autres. Chaque organisation peut avoir des exigences et des préférences différentes.

# CHAPITRE 03 : CONCEPTION ET IMPLEMENTATION DE LA SOLUTION

Dans ce chapitre nous décrivons notre contribution qui consiste en la conception et l'implémentation d'un système qui permet la gestion sécurisée des logs à des fins d'analyse.

## 1. SPECIFICATION ET ANALYSE DES BESOINS

### 1.1. Besoins fonctionnels

Notre système comprend une partie qui permet l'analyse des logs, ce qui implique une interaction avec des utilisateurs. Dans notre cas, nous pouvons distinguer deux types d'utilisateurs : administrateur et superviseur. Notre système doit aussi permettre d'effectuer certaines tâches que nous décrivons sà travers des diagrammes de cas d'utilisation :

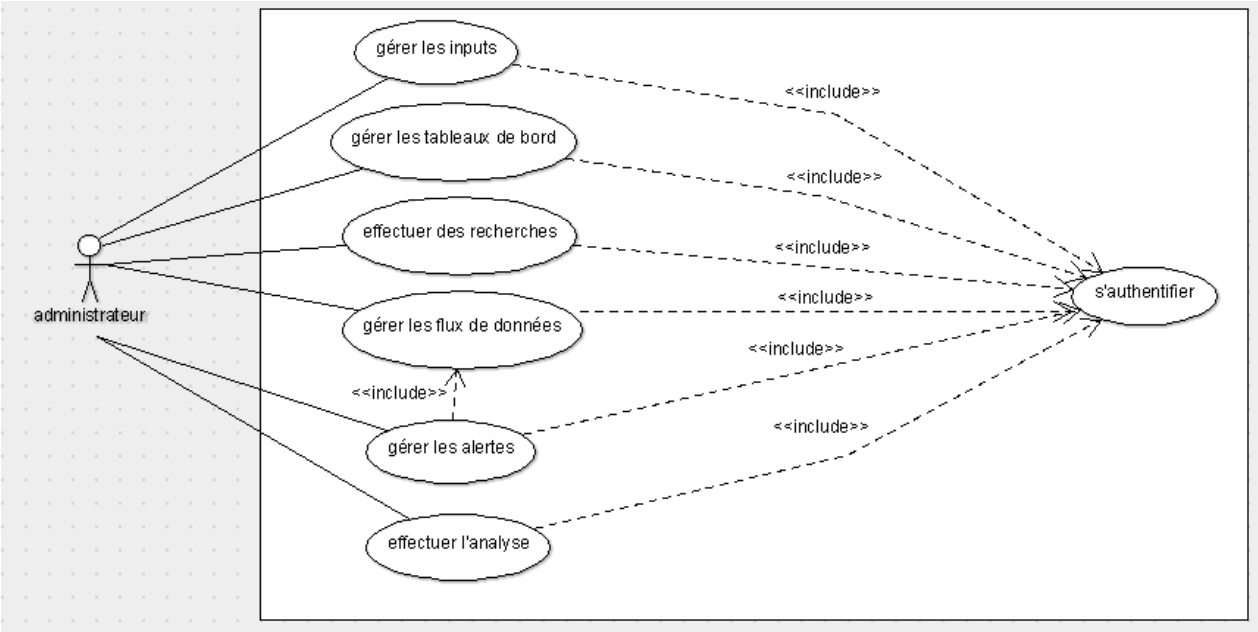



***Figure 1.1.1 : Diagramme de cas d'utilisation d'un administrateur***

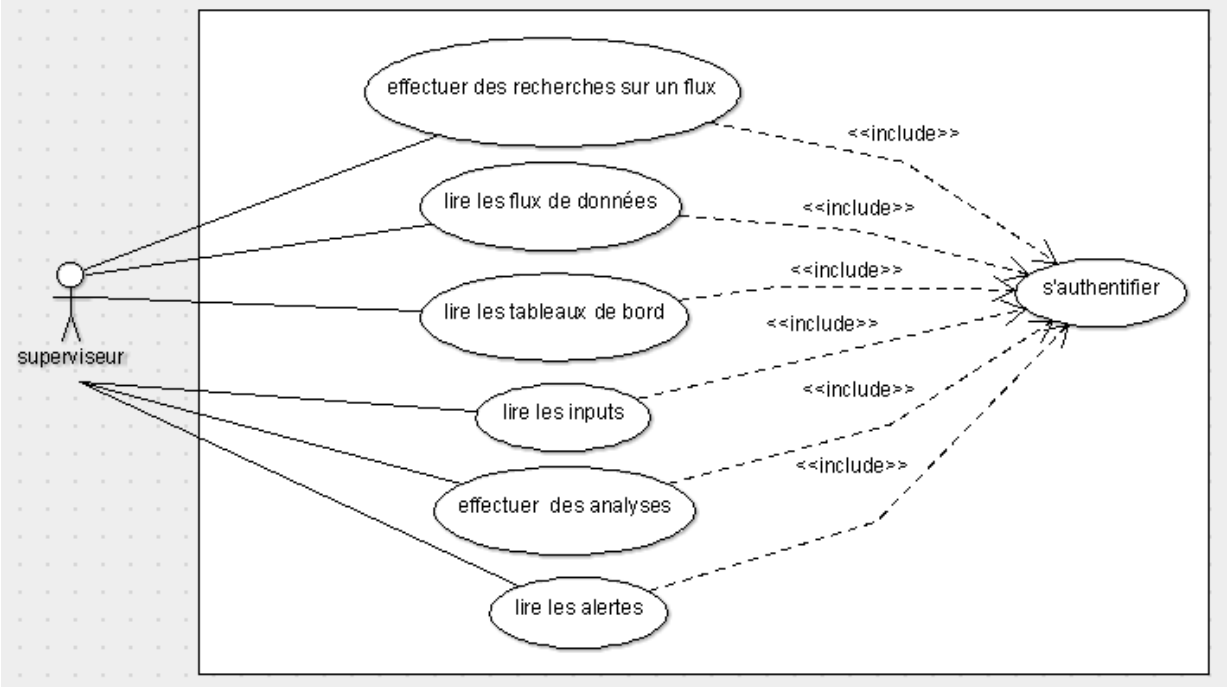



***Figure 1.1.2: Diagramme de cas d'utilisation d'un superviseur***

## 1.2. Besoins non fonctionnels

Notre système doit se prémunir contre les attaques qui peuvent être subies par les différentes entités de l'infrastructure. Ces attaques peuvent être de trois types :

- Attaque contre la confidentialité : c'est la collecte d'informations confidentielles par l'attaquant.
- Attaque contre l'intégrité : c'est le fait de corrompre les données en les modifiant ou en les supprimant.
- Attaque contre la disponibilité : c'est le fait d'empêcher les utilisateurs légitimes d'accéder aux données ou aux systèmes.

## 1.3. Architecture générale de notre système

L'analyse des besoins fonctionnels et non fonctionnels nous a conduits à proposer une architecture qui permet d'y répondre. La figure ci-dessous décrit l'architecture de notre système.

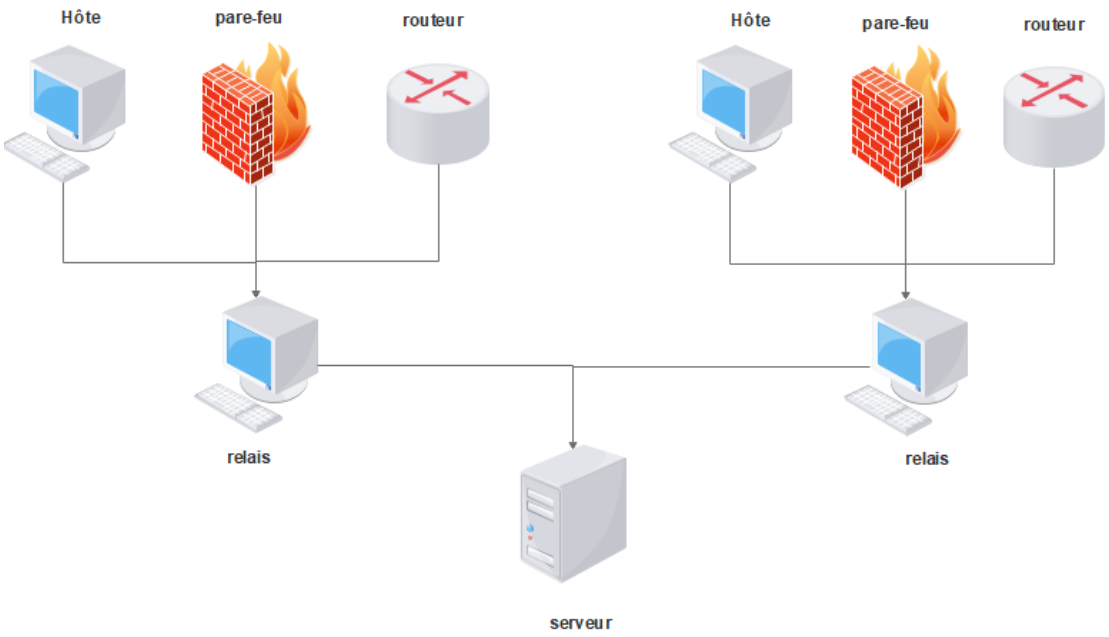



***Figure 1.3.1 : ARCHITECTURE GENERALE.***

Notre système utilise une architecture centralisée. L'avantage de la centralisation des fichiers logs est d'avoir une vue d'ensemble des équipements d'un système d'information pour y mener des traitements tels que :

- Faciliter les recherches et effectuer des statistiques.
- Détecter des intrusions.
- Diagnostiquer une défaillance du système.
- Générer des alertes en temps réel permettant de surveiller en permanence les fichiers journaux à la recherche de signatures préconfigurées

Cette architecture montre aussi l'utilisation de relais. Il permet de diminuer la charge au niveau du serveur ; il permet également d'effectuer une vérification en temps réel de l'intégrité des messages logs que nous allons détailler par la suite.

L'implémentation de notre système est faite en utilisant deux outils : le premier est syslog-ng pour la collecte et le prétraitement des logs ; le second est Graylog pour l'analyse des logs. Notons au passage que la partie gestion des logs est réalisée conjointement par les deux outils.

## 2. PARTIE CLIENT ET RELAI

Pour la mise en œuvre des parties client et relai, notre choix s'est porté sur l'utilisation de l'outil syslog-ng.

La figure suivante montre l'utilisation de syslog-ng au niveau du client :

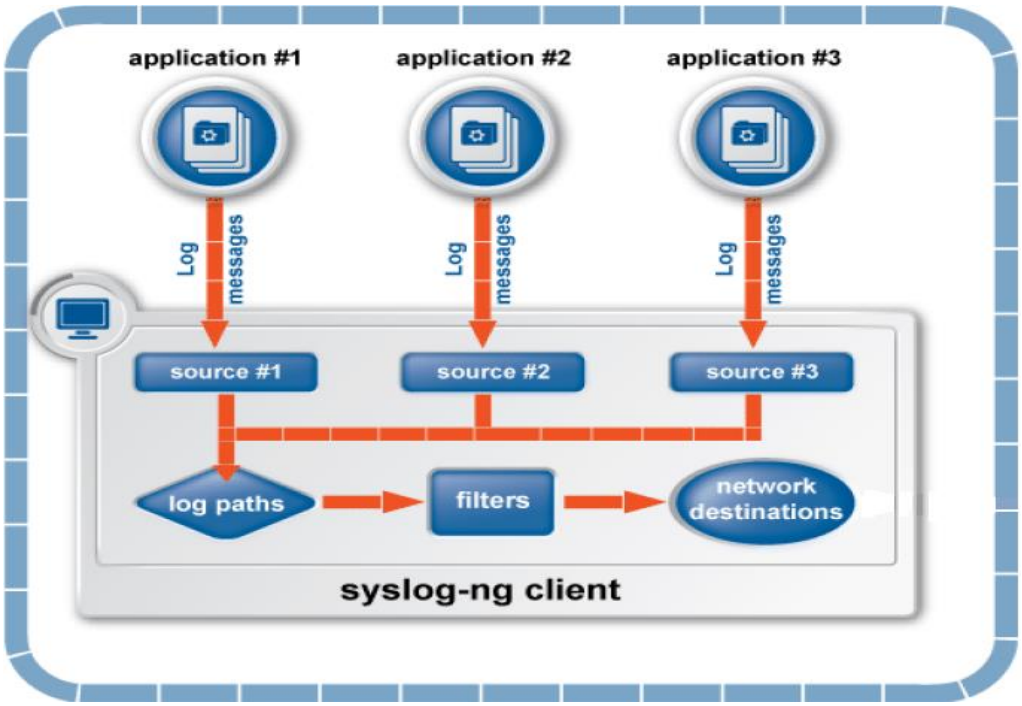



*FIGURE 2.1: PARTIE SYSLOG-NG CLIENT[32].*

Ce schéma résume la collecte des messages logs des différentes applications d'un client avec la définition des sources. "log paths" permet de regrouper les différentes sources en appliquant différents filtres pour envoyer l'ensemble sur le réseau.

Le schéma suivant montre l'utilisation de syslog-ng au niveau du relai :

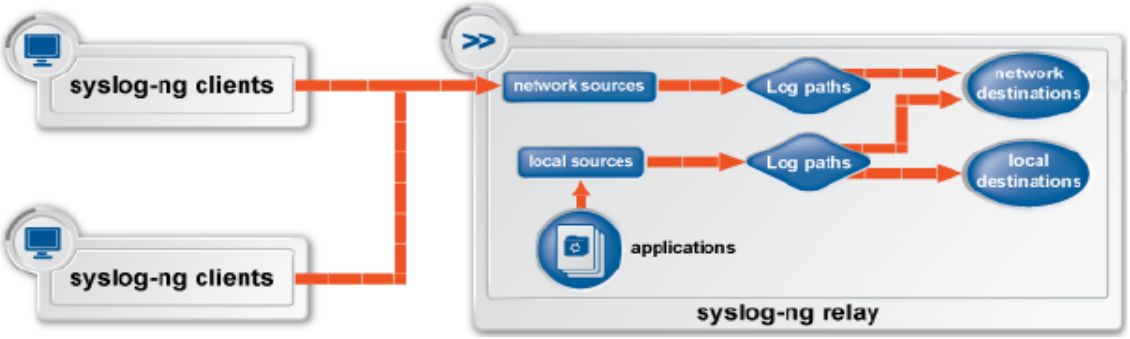



*FIGURE 2.2 : PARTIE SYSLOG-NG RELAI[32].*

En mode relai, syslog-ng reçoit les journaux via le réseau des clients syslog-ng et les transmet au serveur central via une connexion réseau.

Les relais consignent également les messages de l'hôte relai dans un fichier local ou les transmettent au serveur central.

## 2.1. syslog-ng

### 2.1.1. Introduction

Le programme syslog est une interface de journalisation distribuée fournissant un cadre normalisé dans lequel les programmes (système d'exploitation et applications) peuvent envoyer des messages pour être stockés sur le système local ou envoyés à un hôte distant. Cependant, à l'origine, écrit pour UNIX, syslog est devenu un standard pour de nombreux périphériques réseau et est désormais également disponible pour la plate-forme Windows.

Parmi les nombreuses variantes de syslog, syslog-ng est une solution de journalisation centralisée flexible et hautement évolutive créée par une société spécialisée dans le domaine de la sécurité informatique BalaBit. Il fournit un système de journalisation centralisé de tous les périphériques du réseau, quelle que soit la plate-forme. syslog-ng intègre également une multitude de fonctionnalités puissantes, notamment le filtrage en fonction du contenu du message, ainsi que des fonctionnalités personnalisables d'organisation et d'exploration de données.

syslog-ng a apporté beaucoup de nouvelles fonctionnalités par rapport à syslog classique. Parmi les améliorations les plus importantes citons l'ajout de fonctionnalité de sécurité et la prise en charge de TCP. Cependant syslog-ng n'est pas un outil d'analyse des logs. Il ne permet pas l'interprétation de la signification des messages, ni la reconnaissance des modèles d'occurrence des différents messages.

### 2.1.2. Installation

Nous avons choisi comme environnement de travail le système UNIX plus précisément Ubuntu, pour sa facilité de déploiement et de configuration de syslog-ng.

syslog-ng est disponible dans les dépôts, pour l'installer il suffit de taper la commande suivante :

```
apt-get install syslog-ng
```

Le comportement de syslog-ng est contrôlé par un fichier de configuration, généralement stocké dans le fichier /etc/syslog.conf dans un système UNIX.

### 2.1.3. Synchronisation d'horloge

Pour une corrélation correcte des entrées de journal sur les différents relais, l'heure de tous les périphériques doit être synchronisée. Pour cela, nous avons utilisé un serveur NTP pour réaliser la synchronisation horaire appropriée.

### 2.1.4. Format syslog

Nous avons choisi de sauvegarder les logs dans des fichiers texte au format syslog. Il comporte trois parties :

- Un timestamp qui est la date et l'heure du log. Il est au format "Mmm dd hh:mm:ss". « Mmm » correspond aux trois premières lettres du mois, « dd » au numéro du jour du mois, puis l'heure qui est constituée des heures (hh), minutes (mm) et secondes (ss) sur 24 heures.
- Un identifiant de la machine qui a généré le log. Cela peut être le nom de la machine (son hostname) ou son adresse IPv4 ou IPv6.
- Les fonctionnalités(Facility) sont des catégories de log. Ses catégories correspondent à un type de service. Les facilities nous permettrons principalement de filtrer nos logs, ce qui nous permettra de les ranger.
- Les sévérités (priorities) sont les niveaux de gravité des erreurs de log. Il en existe 8. Plus le niveau de priority est proche de 0, plus l'erreur est grave. Plus le niveau de sévérité est haut (proche de 7), plus la machine enverra de log.

- Le message, qui est un message texte comportant des informations.

Exemples :

```
Sep 26 19:42:20 10.0.0.146 sshd[438]: Server listening on 0.0.0.0 port 22.

Sep 26 19:42:20 10.0.0.146 sshd[438]: Server listening on :: port 22.
```

### 2.1.5. Options globales de syslog-ng

L'application syslog-ng dispose d'un certain nombre d'options globales régissant l'utilisation de DNS, le format d'horodatage utilisé et d'autres points d'ordre général. Chaque option peut avoir des paramètres similaires aux spécifications du pilote [32].

Les options les plus importantes que nous avons définies pour le client et pour le relai et qui sont nécessaires au bon déroulement de notre approche sont les suivantes :

"use_dns(no) " :permet de désactiver la résolution du nom de domaine DNS.

"owner("root")" :permet de donner pour tout type d'opération le privilège root.

"create-dirs(yes) " : permet de donner le droit de créer des dossiers en cas de nécessité.

"keep-hostname(yes) " :permet de garder le nom de l'hôte distant qui a envoyer les fichiers logs.

" keep-timestamp(yes)" :spécifie si syslog-ng doit accepter l'horodatage reçu de l'application ou du client d'envoi. Si elle est désactivée, l'heure de réception sera utilisée à sa place.

### 2.1.6. Collecte des messages de journal

Une source est l'endroit où syslog-ng reçoit les messages de journal. Les sources sont composées d'un ou de plusieurs pilotes, chacun définissant où et comment les messages sont reçus [32].

Le tableau suivant répertorie les pilotes sources utilisée dans notre fichier de configuration syslog-ng.

| Nom | Description |
|---|---|
| file() | Ouvre le fichier spécifié et lit les messages. |
| internal() | Messages générés en interne dans syslog-ng. |
| network() | Reçoit les messages des hôtes distants utilisant le protocole BSD-syslog sur IPv4 et IPv6. Prend en charge les protocoles réseau TCP, UDP et TLS. |
| syslog() | Écoute les messages entrants à l'aide du nouveau protocole syslog standard IETF. |
| system() | Détecte automatiquement la plate-forme sur laquelle syslog-ng est exécuté et collecte les messages de journal natifs de cette plate-forme. |

***Tableau2.1.6.1 : PILOTE SOURCE SYSLOG-NG[32].***

### 2.1.7. Transfert sécurisé avec TLS (Transport Layer Security)

L'application syslog-ng peut envoyer et recevoir des messages de journal de manière sécurisée sur le réseau avec l'utilisation de TLS (Transport Layer Security) à l'aide des pilotes network () et syslog ().

TLS utilise des certificats pour authentifier et chiffrer la communication. Le client authentifie le relai en demandant son certificat et sa clé publique. Le relai peut également demander un certificat au client ; ce qui permet également une authentification mutuelle. Toutefois dans le cas où le nombre de clients est important alors cette méthode a pour effet d'engendrer un trafic important de données qui circulent dans le réseau. L'origine de ce problème est l'échange de certificats et de la clé publique. Pour cette raison, nous avons choisi uniquement la méthode de l'authentification du relai. Le principe est que le relai envoie son certificat au client qui contient les informations sur l'identité du relai (ip, localisation…) qui servent à identifier le Relai par le client ; la clé publique servant pour le chiffrement des données par ce même client. Le relai déchiffre enfin ces données avec sa clé privée.

La configuration ci-dessous chiffre les messages de journal à l'aide de TLS et les envoie au port 5514 / TCP du Relai syslog-ng possédant l'adresse IP 192.168.43.34.

```
destination demo_tls_syslog_destination {
syslog("192.168.43.34" port(5514)
transport("tls")
tls(ca-dir("/opt/syslog-ng/etc/syslog-ng/ca.d"))  ); };
```

Le répertoire "/opt/syslog-ng/etc/syslog-ng/ca.d" contient le certificat auto-signé du Relai syslog-ng.

- Configuration de TLS sur le relais syslog-ng :La source suivante reçoit les messages de journal chiffrés à l'aide de TLS sur le port 5514/ TCP de toute interface du relai syslog-ng utilisant le protocole IETF-syslog :

```
source demo_tls_syslog_source {
syslog(ip(0.0.0.0) port(5514)
transport("tls")
tls(  key-file("/opt/syslog-ng/etc/syslog-ng/key.d/syslog-ng.key")
cert-file("/opt/syslog-ng/etc/syslog-ng/cert.d/syslog-ng.cert")
peer-verify(optional-untrusted))   );  };
```

"tls(  peer-verify(optional-untrusted)" :cette instruction a pour rôle de désactiver l'authentification mutuelle pour la source.

"key-file()" et "cert-file()" :ces deux instructions spécifient les fichiers de clé et de certificat du relai.

- Fichier de configuration de stunnel relai : Pour pouvoir transférer et recevoir les logs de manière sécurisée, syslog-ng à besoin d'un module

externe qui est un fichier de configuration qui s'exécute avec un outil qui s'appelle stunnel.

```
debug = 7

cert = /etc/stunnel/cert.pem

key = /etc/stunnel/key.pem

CAfile = /etc/stunnel/cert.pem

verify = 3

pid = /var/run/stunnel4.pid

[5149]

accept = 5149

verifypeer = no

connect = 127.0.0.1:514
```

"cert = /etc/stunnel/cert.pem" :cette instruction indique le chemin menant vers le certificat relai.

"key = /etc/stunnel/key.pem" :cette instruction indique le chemin menant vers la clé privée.

"CAfile = /etc/stunnel/cert.pem" :cette instruction indique le chemin menant vers l'autorité de certification du relai.

"verify = 3" et "debug = 7" :ces instructions nous permettent d'être en courant en cas ou une erreur venait à se produire en affichant des messages d'erreurs sur le terminal.

"pid = /var/run/stunnel4.pid" :cette instruction permet de lancer un processus qui a le nom de stunnel4.pid.

" [5149] " : le [5149] représente le numéro de port de destination.

"connect = 127.0.0.1:514" : cette adresse représente l'adresse au niveau local avec le numéro de port source 514.

"verifyPeer = no" : cette instruction a pour rôle de désactiver l'authentification mutuelle pour la source.

- Configuration de TLS sur le client syslog-ng :

```
destination demo_tls_destination { syslog("192.168.43.34" port(5150)
transport("tls")
tls( ca-dir("/etc/syslog-ng/ssl")) ); };
```

"tls( ca-dir("/etc/syslog-ng/ssl"))" :cette instruction donne le chemin du dossier qui contient le certificat du relai qui est dans notre exemple "ssl".

- Fichier de configuration de stunnelclient :Le fichier de configuration de stunnel coté client contient les informations suivantes :

```
client = yes
CAfile = /etc/stunnel/cert.pem
verify = 3
pid = /var/run/stunnel4.pid
[5149]
accept = 127.0.0.1:514
verifyPeer = no
connect = 192.168.43.34:5149
```

"client = yes" :cette instruction indique que c'est un terminal client.

"CAfile = /etc/stunnel/cert.pem" :cette instruction indique le chemin menant vers le certificat relai.

"verify = 3" :cette instruction nous permet d'être en courant en cas ou une erreur venait à se produire en affichant des messages d'erreurs sur le terminal.

"pid = /var/run/stunnel4.pid" :cette instruction permet de lancer un processus qui a le nom de stunnel4.pid.

" [5149] " : le [5149]représente le numéro de port de destination.

"accept = 127.0.0.1:514" :cette adresse représente l'adresse au niveau local avec le numéro de port source 514.

"verifyPeer = no" :cette instruction a pour rôle de désactiver l'authentification mutuelle pour la source.

"connect = 192.168.43.34:5149" :cette adresse représente l'adresse de destination avec son numéro de port de destination qui est le 5149.

### 2.1.8. Envoi et stockage des messages de journal

Une destination est l'endroit où un message de journal est envoyé si les règles de filtrage correspondent. De la même manière que pour les sources, les destinations sont composées d'un ou de plusieurs pilotes, chacun définissant où et comment les messages sont envoyés [32].

Le tableau suivant répertorie les pilotes de destination utilisés dans notre configuration syslog-ng.

| Nom | Description |
|---|---|
| file() | Écrit des messages dans le fichier spécifié. |
| syslog() | Envoie des messages à l'hôte distant spécifié à l'aide du protocole IETF-syslog. La norme IETF prend en charge le transport des messages syslog () à l'aide des protocoles de réseau UDP, TCP et TLS. |
| usertty() | Envoie des messages au terminal de l'utilisateur spécifié, si l'utilisateur est connecté. |
| Pipe() | Écrit des messages dans le canal nommé spécifié. |

***Tableau2.1.8.1 : PILOTE DESTINATION SYSLOG-NG[32].***

Voici quelque exemple de destinations utilisées dans notre approche coté relai:

```
############################
# Destinations
############################
destination d_auth { file("/var/log/$HOST/auth.log"); };
destination d_myapp { file("/var/log/$HOST/myapp.log"); };
destination hash_myappp { file("/var/log/$HOST/hach_myappp.log"); };
destination d_ufw { file("/var/log/idris/ufw.log"); };
destination d_python { file("/var/log/$HOST/python.log"); };
destination d_host { file("/var/log/$HOST/$HOST.log"); };
```

Une destination se compose d'une variable qui est précédée dans notre configuration par «d_ » par exemple dans la première ligne de notre fichier de configuration :

"destination d_auth { file("/var/log/$HOST/debug.log") ; } ; " : d_auth représente une variable de " destination " et "{ file("/var/log/$HOST/debug.log") ; } ; " représente la valeur de cette variable .

"file("/var/log/$HOST/debug.log")" :cette instruction permet de spécifier le chemin qui est "/var/log/$HOST/" vers le fichier log "debug.log".

"$HOST" :cette variable prend la valeur de l'hôte émetteur.

"destination d_ufw" :correspond aux messages logs du firewall.

"destination d_python" :correspond aux messages logs générés par un programme python qui s'exécute en arrière-plan, qui a pour rôle de s'assurer de l'intégrité des données en vérifiant leur hache.

"destination d_host" :correspond aux messages logs générés par un hôte distant.

## 2.1.9. Filtrer et classer

L'application OSE syslog-ng peut trier les messages de journal entrants en fonction de leur contenu et de divers paramètres tels que l'hôte source, l'application et la priorité. Il offre aussi la possibilité de créer des répertoires, des fichiers et des tables de base de données de manière dynamique à l'aide des macros. Le filtrage complexe à l'aide d'expressions régulières et d'opérateurs booléens offre une flexibilité presque illimitée pour transférer uniquement les messages de journal importants aux destinations sélectionnées [32].

Le tableau suivant répertorie les valeurs de gravité :

| Code numérique | Gravité |
|---|---|
| 0 | Urgence: le système est inutilisable |
| 1 | Alerte: il faut agir immédiatement |
| 2 | Critical: conditions critiques |
| 3 | Erreur: conditions d'erreur |
| 4 | Attention: conditions d'avertissement |
| 5 | Avis: état normal mais significatif |
| 6 | Informationnel: messages d'information |
| 7 | Debug: messages au niveau du débogage |

***Tableau2.1.9.1 : VALEUR GRAVITE SYSLOG-NG[32].***

Nous avons utilisé dans notre configuration les fonctions suivantes :

| Nom | Description |
|---|---|
| facility() | Filtrer les messages en fonction de l'installation d'envoi. |
| filter() | Appelle une autre fonction de filtrage. |
| netmask() | Filtre les messages en fonction de l'adresse IP de l'hôte d'envoi. |
| level() or priority() | Filtrer les messages en fonction de leur priorité. |

***Tableau 2.1.9.2 : FONCTION SYSLOG-NG (SYSLOG-NG 2018).***

Nous avons utilisé dans notre configuration les facilités suivantes :

| Code numérique | Nom du Facility | Facility |
|---|---|---|
| 0 | kern | messages du noyau |
| 2 | mail | système de messagerie |
| 3 | daemon | démons système |
| 4 | auth | messages de sécurité / autorisation |
| 5 | syslog | messages générés en interne par syslogd |
| 6 | lpr | sous-système imprimante en ligne |
| 7 | news | sous-système de nouveaux réseau |
| 8 | uucp | Sous-système UUCP |
| 9 | cron | démon d'horloge |
| 10 | authpriv | messages de sécurité / autorisation |
| 16-23 | local0..local7 | Installations utilisées localement (local0-local7) |

***Tableau2.1.9.3 : FACILITES SYSLOG-NG[32].***

Voici quelque filtre utilisés dans notre approche coté relai :

```
############################
# Filters
############################
filter f_dbg { level(debug); };
filter f_info { level(info); };
filter f_notice { level(notice); };
filter f_warn { level(warn); };
filter f_err { level(err); };
filter f_crit { level(crit .. emerg); };
```

```
filter f_debug { level(debug) and not facility(auth, authpriv, news, mail); };
filter f_error { level(err .. emerg) ; };
filter f_myapp { filter(f_error) and not filter(f_debug); };
filter f_python { facility(local0) and not level(debug); };
filter f_ipHost { netmask(192.168.1.48/24); };
```

L'instruction "filter f_myapp { filter(f_error) and not filter(f_debug); };"est un exemple de l'utilisation d'opérateurs booléens. Elle va filtrer les messages logs dont la valeur du filtre est " filter(f_error)" c'est-à-dire les messages logs qui correspondent à une gravité de type erreur, alerte, critique ou urgent et ne va pas filtrer les fichiers "filter(f_debug)" c'est-à-dire les fichiers logs de niveaux debug.

L'instruction "filter f_ipHost { netmask(192.168.1.48/24); };" peut être utilisée pour filtrer les fichiers logs qui ont été émis d'un hôte avec la valeur de son adresse ip 192.168.1.48/24.

### 2.1.10. Préparation et réécriture

Syslog-ng propose de nombreuses fonctionnalités en ce qui concerne la gestion des logs, cependant il ne propose aucun mécanisme pour assurer l'intégrité des données. S'assurer de la fiabilité des logs c'est permettre d'utiliser ces logs comme preuve en cas d'enquête (forensic) sur une éventuelle intrusion ou tout autre acte malveillant. La cryptographie est une technique importante utilisée pour la communication sécurisée en présence d'une tierce entité indiscrète. Il fournit tous les aspects primordiaux de la sécurité de l'information tels que la confidentialité, l'intégrité, l'authentification et la non-répudiation des données. Les méthodes de cryptographie sont : cryptosystème à clé symétrique, cryptosystème à clé asymétrique et cryptosystème hybride.

❖ Les inconvénients d'un algorithme à clé symétrique sont les suivants :
- L'échange de clés de manière sécurisée devient un problème.
- Les dommages causés lorsqu'une entité met la main sur une clé

symétrique, car elle peut déchiffrer tout ce qui est cryptéavec cette clé.

• l’augmentation du nombre de participants utilisant la clé secrète augmente le risque de dommages et les conséquences de ces dommages augmentent aussi.

- Les inconvénients d'un algorithme à clé asymétrique sont ainsi :

• Comme les clés asymétriques doivent être plus longues que laclé secrète dans l'algorithme à clé symétrique, elles sont plus coûteuses en calcul.

• Ils sont sensibles aux attaques en moins de temps que la force brute.

• Il est également vulnérable aux attaques de l'homme du milieu.

• Il existe de nombreux systèmes de clés publiques qui font appel à une tierce partie pour certifier leur fiabilité.

- pour s’assurer de l’intégrité des entrées logs, notre choix s’est porté sur Les fonctions de hachage à sens unique, car elles répondent à tous les aspects de la sécurité de l’information en raison des caractéristiques suivantes :

• Il est facile de calculer le hash d’un message donné.

•Il ne peut jamais y avoir deux messages associés aux mêmes hash.

•Il est, par définition, impossible de modifier un message sans changer sa valeur de hachage.

• Il est, par définition, impossible, pour une valeur de hachage donnée, de construire un message ayant cette valeur de hachage.

- Les deux algorithmes de hachage cryptographiques largement connus sont:MD5,SHA :

<u>MD5</u>

•Il est facile de produire une collision dans MD5.

•Il y a des attaques par collision dévastatrice sur le MD5. Ces attaques signifient que MD5 ne fournit essentiellement aucune sécurité contre les collisions.

<u>SHA</u>

•Il n'est pas facile de produire des collisions SHA-1.

• Il n’existe aucune collision pour SHA-1 qui a été produite. SHA-256, qui est beaucoup plus "massif" (beaucoup plus d'opérations que SHA-1, mais

avec une structure similaire), est actuellement non cassé.
- Il est plus rapide que MD5 pour l'assemblage.
- Il est beaucoup plus sécurisé que les autres algorithmes. Bien que certaines attaques sur SHA1 soient connues, elles sont beaucoup moins graves que les attaques sur MD5.

De nos jours, au lieu d'utiliser MD5 ou SHA1, sur lequel il y a des attaques connues, il y a probablement encore de meilleures fonctions de hachage modernes, comme SHA256. Ceux-ci n'ont pas d'attaques connues ayant une pertinence pratique [41].

Dans notre approche. Chaque entrée log reçue par syslog-ng est normalisé sous format syslog et se voit attribuer une signature via la fonction sha-256(256 bits), et ceci est fait en temps réel, ce qui permet de ne pas laisser du temps à un intrus de modifier les messages logs.
La fonction de signature qui est présent coté client et coté relai :

```
rewrite msghashing {
set("$(sha256 $MESSAGE)", value("MYMESS"));
  set ("°$MESSAGE° hash: $MYMESS", value("MESSAGE"), on-error("fallback-to-string"));  };
```

"°$MESSAGE°" : le "$MESSAGE" est une variable qui contient la valeur du message qui est délimitée par "°".

"$MYMESS" : cette variable contient la valeur du hash.

"$(sha256 $MESSAGE)" : cette instruction signifie que nous allons appliquer à la variable "$MESSAGE" la fonction du hash sha 2 de 256 bit.

Voici un example de sortie

```
Jun 10 17:14:59 idris systemd[1]: °Reloading System Logger Daemon.° hash: 17079db6a820b7b6b6e4109e072dd6c8c7d6f48886e2349c67b0bd70214db65a
```

```
Jun 10 17:14:59 idris systemd[1]: °Reloaded System Logger Daemon.° hash:
96669a7162e80be37e231882e871317551770bd9a6017abea45e600116197e
f9
```

Inverser le hachage :afin de valider le fait que les données n'ont pas été modifiées, un nouveau hachage est généré avec les données reçues et est mis en correspondance avec le hachage d'origine. Grâce à un script python qui s'exécute en arrière-plan, en parcourant le fichier ou sont envoyés les messages logs de haut en bas, le programme va attendre qu'une nouvelle ligne soit insérée dans le fichier pour pouvoir vérifier la correspondance des résultats. Cela représente une vérification en temps réel. Si le programme détecte que les données ne correspondent pas avec leur hash, le programme va générer un message log alertant l'administrateur et contenant les informations nécessaires pour agir immédiatement en cas de nécessité.

### 2.1.11. Chemins de journal

Les chemins de journal déterminent ce qui se passe avec les messages de journal entrants. Les messages provenant des sources répertoriées dans l'instruction de journal et correspondant à tous les filtres sont envoyés aux destinations répertoriées [32].

Pour chaque chemin nous pouvons définir plusieurs sources tout en lui appliquant un certain nombre de filtres de fonction rewrite, suivis des destinations ou les fichiers log vont être stockés.

Nous avons utilisé dans notre solution la configuration suivante pour le relai :

```
log { source(s_src); source(demo_tls_source); destination(hash_myappp); };

log { source(s_src); source(demo_tls_source); filter(f_ipHost);
destination(d_host); };

log { source(s_src); filter(f_myapp); rewrite(msghashing);
destination(d_myapp); };
```

```
log { source(s_ufw); rewrite(msghashing); destination(d_ufw); };

log { source(s_src); filter(f_python); rewrite(msghashing);
destination(d_python); };
```

Et pour le client nous avons utilisé la configuration suivante :

```
log { source(s_src); source(demo_tls_source); destination(hash_myappp); };
```

L'exemple ci-dessous montre un exemple de résultatd'un chemin du journal :

```
Jun 10 17:14:58 idris sudo[8819]: °   idris : TTY=pts/0 ; PWD=/home/idris ;
USER=root ; COMMAND=/bin/systemctl reload syslog-ng.service° hash:
c6ff6f38bb3ccd48fbeb784710dacb85195cda19316298372def9431708a2ce8

Jun 10 17:14:58 idris sudo[8819]: °pam_unix(sudo:session): session opened
for user root by (uid=0)° hash:
332cbfe512457408c120173fa352aedc85899d675da9e88fc923555cf5e897dc
```

### 2.1.12. La rotation des fichiers logs

La rotation par définition peut archiver un fichier journal sous un autre nom et supprimer la plus ancienne archive.

Dans notre solution les fichiers logs sont envoyés et conservés dans Elasticsearch suivant une politique de conservation bien définie, ce qui implique que la conservation au niveau local n'est plus nécessaire. Notre choix de configuration s'est alors porté sur la rotation quotidienne.

Les configurations et les options par défaut de l'utilitaire Logrotate sont disponibles dans le fichier */etc/logrotate.d/*

Voici la solution que nous avons définie au niveau du fichier de configuration et qui est la suivante :

```
#Logrotate for app
/var/log/Relai/*.log {
su root syslog
copytruncate
dateext
dateformat _%Y-%m-%d
extension .log
  daily
 notifempty
  missingok
  rotate 0   }

/var/log/client/*.log {
su root syslog
copytruncate
dateext
dateformat _%Y-%m-%d
extension .log
  daily
 notifempty
  missingok
  rotate 0 }
```

Quelques-uns des réglages importants à configurer sont :

- Pour les fichiers logs du relai et du client, Logrotate surveille les fichiers /var/log/Relai/*.log et /var/log/client/*.log et génère une rotation quotidienne.
- 'su root syslog' signifie la délégation de tous les priviléges root à syslog.
- 'rotate 0'signifie qu'à chaque intervalle, on ne conserve pas les fichiers de journalisation.
- 'missingok'permet au processus de ne pas s'arrêter à chaque erreur et de poursuivre avec le fichier de log suivant.
- 'notifempty' empêche la rotation de s'effectuer si le fichier de log est vide.

Nous avons vu comment nous pouvons configurer nos clients et relai pour centraliser tous nos logs. Logs sont tous regroupés dans un relai et rangés comme nous le souhaitons. Le problème est que ces logs ne sont pas très lisibles et plaisants à lire. Certains logiciels peuvent nous aider à parcourir les logs grâce à une interface comme php-syslog-ng et Glogg.

Php-syslog-ng utilise une base de données pour stocker les logs (MySQL) et nous permet de les visualiser par l'intermédiaire d'une interface web. Nous pouvons faire des recherches, selon le type d'erreur, sur une période définie, que ce soit par client ou par sévérité (facility).

Glogg, lui est fonctionnel sur Windows Linux et Mac. Il nous permet de parcourir des fichiers logs volumineux et de faire des recherches en utilisant des expressions régulières.

Les logs sont consultables à tout moment mais c'est à nous d'aller les lire lorsque nous nous rendons compte qu'il y a un problème. Il existe des logiciels qui lisent ces logs à notre place et qui nous avertissent lorsqu'une erreur survient comme par exemple Logwatch, qui envoie un mail lorsqu'une erreur grave survient.

Il existe d'autres logiciels qui vont plus loin comme ELK et Graylog. Ils peuvent tout comme php-syslog-ng stocker les logs et faire des

recherches, mais ils proposent aussi une interface qui permet d'analyser les logs en construisant des graphiques, camemberts ou tableaux.

# 3. PARTIE SERVEUR

## 3.1. Graylog

Graylog est une solution open-source de gestion de messages/texte/logs. Chaque message reçu par un nœud serveur est enregistré dans une base de données Mongo-DB indexée via Elasticsearch. Une interface web permet de gérer et d'analyser les données.

Graylog est découpé en 2 parties principales : Graylog-server et Graylog-web. La première est une application Java qui accepte les messages sur différents protocoles : UDP, TCP, GELF, AMQP, …. Une API Rest est également intégrée à l'outil et est notamment utilisée par la partie web-interface. Celle-ci (la deuxième partie), nous permet de gérer des utilisateurs, des flux et des tableaux de bord.

## 3.2. Architecture Graylog

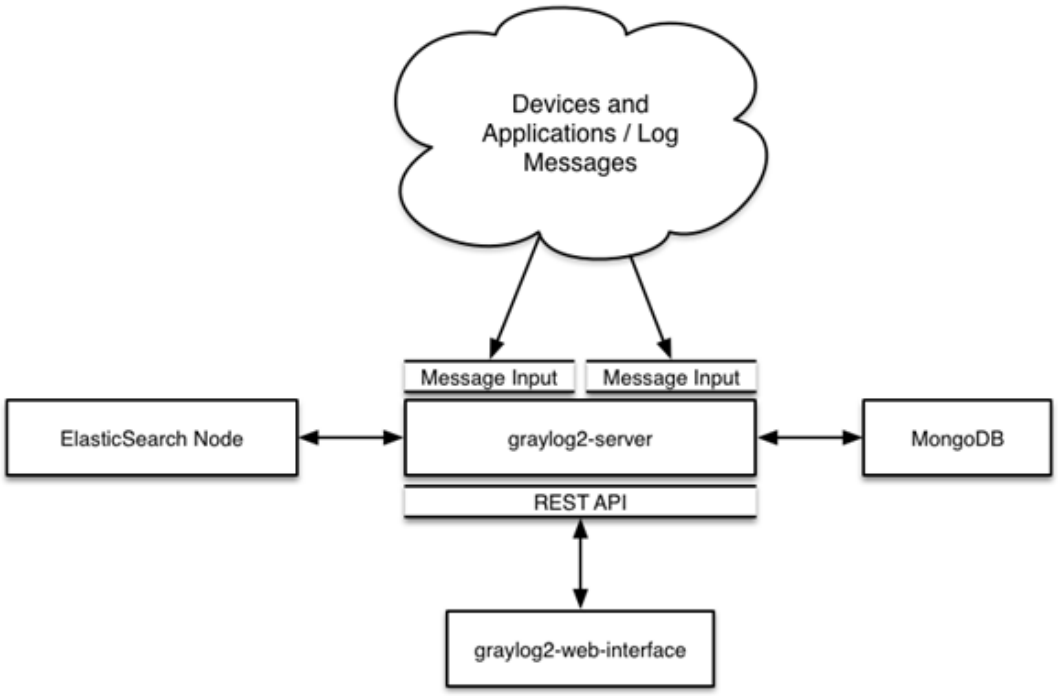



***Figure3.2.1 : ARCHITECTURE MONTRANT LA CONFIGURATION DE GRAYLOG***

## 3.3. Composants de Graylog

Nous avons choisi comme OS d'utiliser Ubuntu, car il est stable, documenté et sa procédure d'installation est simple. Il offre aussi la possibilité de chiffrer la partition sur laquelle Ubuntu est installée. Comme le montre la figure 22 le serveur Graylog se compose d'une base de données MongoDB et d'un moteur de recherche Elasticsearch.

### 3.3.1. MongoDB

MongoDB est une base de données orientée documents (mongodb). Graylogl'utilise pour stocker nos données de configuration, pas nos données de journal. Seules les métadonnées sont stockées, telles que les informations utilisateur ou lesconfigurations de flux. Aucun de nos messages de journal n'est stocké dans MongoDB. MongoDB s'exécutera simplement en parallèle des processus de notre serveur Graylog et n'utilisera pratiquement aucune ressource [35].

Graylog utilise MongoDB pour stocker nos données de configuration. Telles que les informations utilisateur ou les configurations de flux.

### 3.3.2. Elasticsearch

Elasticsearch est un moteur de recherche et d'analyse distribué, open source, RESTful, basé sur Apache Lucene. Depuis sa publication en 2010, Elasticsearch est rapidement devenu le moteur de recherche le plus populaire. Il est couramment utilisé pour l'analyse de journaux, la recherche en texte intégral, les analyses à des fins commerciales [34].

### 3.3.3. Serveur Graylog

Graylog est une solution de gestion de journaux centralisée de premier plan construite selon des normes ouvertes pour la capture, le stockage et l'analyse en temps réel de téraoctets de données machine.

L'installation de Graylog nécessite la configuration d'un accès avec mot de passe. Pour cela nous avons modifié le fichier du serveur qui se trouve au

niveau de */etc/Graylog/server/server.conf* pour ajouter « password_secret » et« root_password_sha2 »qui permettront l'authentification de l'utilisateur.

Le serveur écoute par défaut sur le port : http://127.0.0.1:9000/

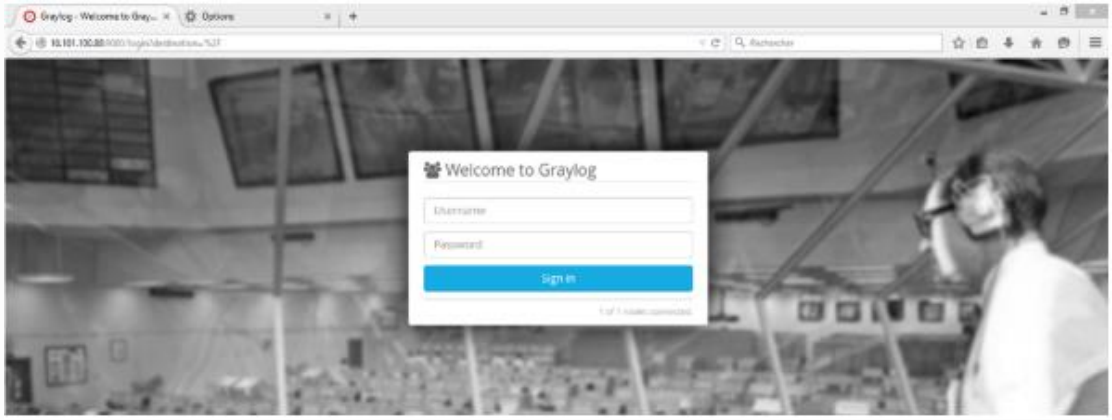



***Figure 3.3.3 : CAPTURE D'ECRAN DE L'INTERFACE WEB***

## 3.4. Activation du HTTPS

Afin de sécuriser notre installation Graylog, nous avons utilisé SSL/TLS pour nous assurer qu'aucune donnée sensible n'est envoyée en clair sur le réseau. Pour que cela fonctionne nous devons activer*http_enabel_tls*. Nous devons également nous assurer que nous disposons des certificats appropriés et validés par les clients.

La première étape consiste à créer un fichier nommé*openssl_Graylog_cnf* contenant les informations suivantes :

```
[req]
distinguished_name = req_distinguished_name
x509_extensions = v3_req
prompt = no

# Details about the issuer of the certificate
[req_distinguished_name]
C = US
ST = Some-State
L = Some-City
O = My Company
OU = My Division
CN = graylog.example.com

[v3_req]
keyUsage = keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth
subjectAltName = @alt_names

# IP addresses and DNS names the certificate should include
# Use IP.### for IP addresses and DNS.### for DNS names,
# with "###" being a consecutive number.
[alt_names]
IP.1 = 192.168.43.34
DNS.1 = graylog.example.com
```

Puis nous avons créé la clé privée*pkcs#5* et le certificat *x.509* :

```
idris@idris:~$ openssl req -x509 -days 365 -nodes -newkey rsa:2048 -config openssl-graylog.cnf -keyout pkcs5-plain.pem -out cert.pem
```

Nous avons ensuite converti la clé*pkcs#5* en une clé privée*pkcs#8*, car Graylog n'accepte que ce format :

```
idris@idris:~$  openssl pkcs8 -in pkcs5-plain.pem -topk8 -nocrypt -out pkcs8-plain.pem
```

Nous pouvons utiliser la clé privée *pkcs#8* résultante et le certificat *x.509* pour activer la connexion chiffrée dans le fichier de configuration Graylog :

```
################
# HTTPS settings
################

#### Enable HTTPS support for the HTTP interface
#
# This secures the communication with the HTTP interface with TLS to prevent request forgery and eavesdropping.
#
# Default: false
http_enable_tls = true

# The X.509 certificate chain file in PEM format to use for securing the HTTP interface.
http_tls_cert_file = /home/idris/certs/cert.pem

# The PKCS#8 private key file in PEM format to use for securing the HTTP interface.
http_tls_key_file = /home/idris/certs/pkcs8-plain.pem
```

La dernière étape consiste à ajouter le certificat auto-signé au magasin de clé java (le mot de passe par défaut est « changeit ») :

```
idris@idris:~$ keytool -importcert -keystore /path/to/cacerts.jks -storepass changeit -alias graylog-self-signed -file cert.pem
```

C'est fait HTTPS est activé :

https://192.168.43.34:9000/gettingstarted

L'étape suivante consiste à intégrer des messages logs dans notre serveur Graylog.

## 3.5. Ajout des inputs

Graylog est capable d'accepter et d'analyser les messages syslog conformes aux normes RFC 5424 et RFC 3164 et prend en charge le transport (TCP/UDP) avec les méthodes de comptage d'octets ou de caractères de fin. En règle générale, les messages transmis par syslog-ng sont généralement analysés sans faille.

Nous allons utiliser le Protocol syslog, donc des Inputs syslog pour écouter et collecter les messages :

Nous utiliserons **« syslog TCP** » avec le port 5000 pour les messages syslog puis « **launch new input** ».


Voici les étapes de la configuration de l'input syslog TCP :

- Nous avons coché la case « Global Input » pour le lancer sur tous les nœuds, bien que nous n'ayons qu'un seul nœud.
- Nous avons nommé explicitement l'input « **tcp input** ».
- Nous avons renseigné le port TCP choisi (5000).
- Nous avons mis le bind sur 0.0.0.0 pour Graylog écoute sur tous les réseaux.
- Nous autorisons de remplacer la date reçue par celle du serveur si elle n'est pas lisible dans le message reçu.
- Nous avons laissé les autres options par défaut, leurs valeurs nous conviennent.
- On clique sur « **Launch** » pour lancer la capture.

Notre input est lancé comme montré dans la figure ci-dessous.

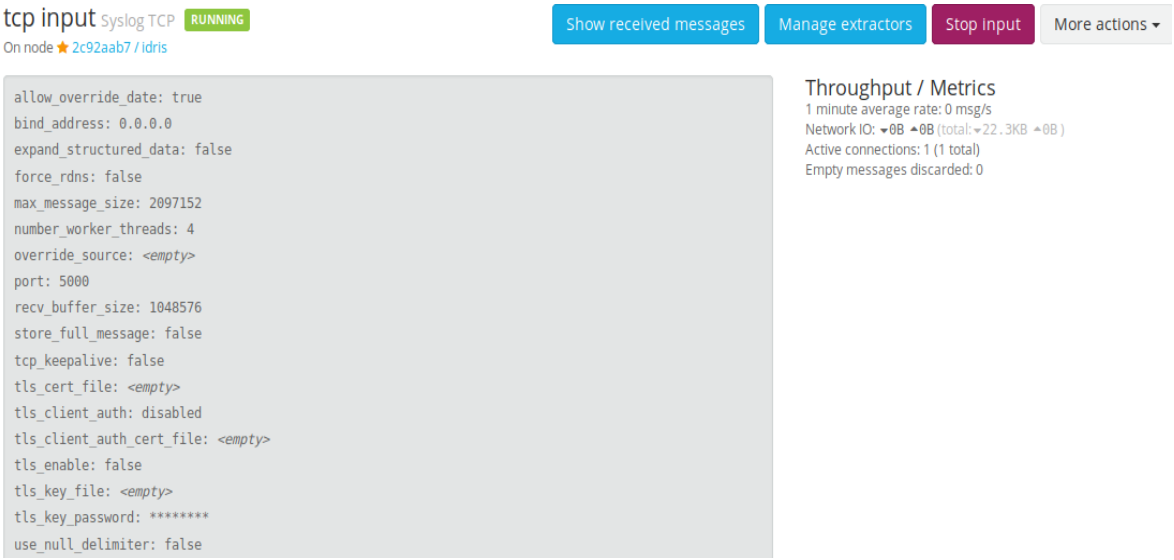



***Figure 3.5.1 : CAPTURE D'ECRAN MONTRANT L'ETAT DU FONCTIONNEMENT DE L'INPUT.***

## 3.6. Configuration des clients

Maintenant que notre serveur Graylog est prêt à recevoir des messages pour nos équipements grâce à ses inputs, nous allons pouvoir configurer les hôtes pour envoyer leurs logs à notre serveur.

syslog-ng est modulable. Pour transférer ses logs à notre serveur Graylog, il faut ajouter ces lignes suivantes dans « /etc/syslog-ng/syslog-ng.conf » :

```
destination d_net { syslog(192.168.43.34 transport("tcp") port(5000) ); };
```

```
log { source(s_src); source(demo_tls_source); destination(d_net); };
```

La 1ère ligne permet de déterminer la destination des messages logs en utilisant le protocole IETF-syslog basé le TCP, en y indiquant l'adresse IP et le port. La 2ème ligne indique aux logs générés par les applications en interne et les logs reçus de la part des hôtes distants d'être envoyés au serveur Graylog.

Nous redémarrons par la suite le démon syslog-ng :

```
idris@idris:~$ sudo systemctl reload syslog-ng.service
```

Les logs commencent à arriver comme montré dans la figure ci-dessous :


***Figure 3.6.1 : CAPTURE D'ECRAN MONTRANT L'ARRIVE DES MESSAGES LOG AU SERVEUR.***

## 3.7. Recherche

Pour la recherche des données, Graylog dispose d'un moteur de recherche très simple qui est représenter dans la figure 3.7.1. Il est possible d'utiliser les opérateurs classiques des moteurs de recherche : jointure AND et OR, NOT et de spécifier une plage horaire. Cela permettra à Graylog de rechercher uniquement dans les index pertinents et réduira considérablement la charge du système et les ressources requises. Le *time frame* offre trois manières différentes de sélectionner une plage de temps et est essentiel pour la vitesse de recherche :

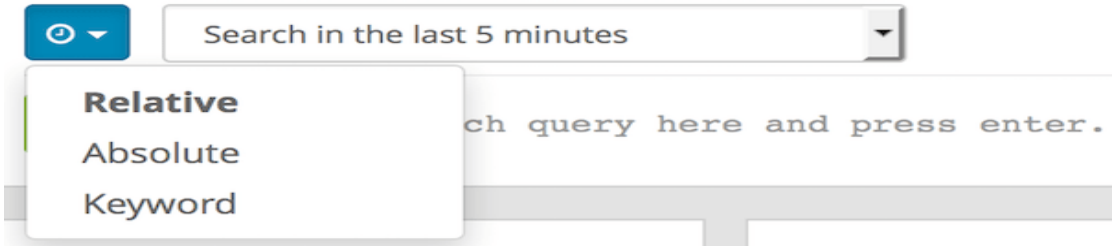



***Figure3.7.1: CAPTURE D'ECRAN DU MOTEUR DE RECHERCHE GRAYLOG.***

- Sélecteur de période relative :

Le sélecteur de cadre temporel relatif nous permet de rechercher des messages de l'option sélectionnée à l'heure à laquelle nous appuyons sur le bouton de recherche. Le sélecteur offre une large gamme de délais relatifs qui répondent à la plupart de nos besoins de recherche.

- Sélecteur de délai absolu :

Lorsque nous connaissons exactement les limites de notre recherche, nous utilisons le sélecteur de délai absolu. Il suffit d'introduire les dates et heure de

la recherche manuellement ou de cliquer dans le champ de saisie pour ouvrir un calendrier dans lequel il est possible de choisir le jour avec une souris

- Sélecteur de délai de mot clé :

Graylog propose un sélecteur de délai de mot-clé qui nous permet de spécifier le délai de la recherche en langage naturel, exemple la « dernière heure » ou les « 90 derniers jours ».

## 3.8. Flux

Les flux Graylog sont un mécanisme permettant de router les messages en catégories et en temps réel pendant leur traitement. Nous définissons des règles qui indiquent à Graylog quel message acheminer dans quels flux.
Pour créer un flux il faut :

1. Accéder à la section *Streams* dans la barre de navigation supérieure.
2. Cliquer sur « *Create Stream* ».
3. Enregistrer le flux après avoir saisi un nom et une description. Le flux est alors enregistré mais pas encore activé.
4. Cliquer sur « *Manage Rules* » pour le flux que nous venons de créer. Une page est alors ouverte où nous pourrons gérer et tester les règles de flux.
5. Choisir comment nous souhaitons évaluer les règles de flux pour déterminer les messages à inclure dans le flux :
   - Un message doit correspondre à toutes les règles suivantes (ET logique) : Les messages ne seront routés dans le flux que si toutes les règles du flux sont satisfaites. Ceci est le comportement par défaut
   - Un message doit correspondre à au moins une des règles suivantes (OU logique) : Les messages seront acheminés dans le flux si une ou plusieurs règles du flux sont satisfaites.
6. Ajouter des règles de flux en indiquant le champ que nous souhaitons vérifier et la condition à remplir. Puis nous devons cliquer sur « i'm done !».
7. Le flux est toujours en pause, nous devons cliquer sur le bouton « Start Stream» pour activer le flux.

Dans notre solution, nous avons configuré des flux pour chaque périphérique, plus un autre flux pour les messages logs générés par le serveur Graylog. La règle qui a permis la description de ces flux est basée sur le champ source des messages syslog. Elle s'applique au niveau du flux par défaut, là où tous les messages logs générés par les différents équipements réseaux et terminaux vont être reçus et traités.

La création de ces flux va nous permettre par la suite de définir des conditions d'alerte personnalisées selon le périphérique, d'analyser leur activité séparément via des tableaux de bord et de faciliter la maintenance du serveur Graylog comme montré dans la figure suivante.

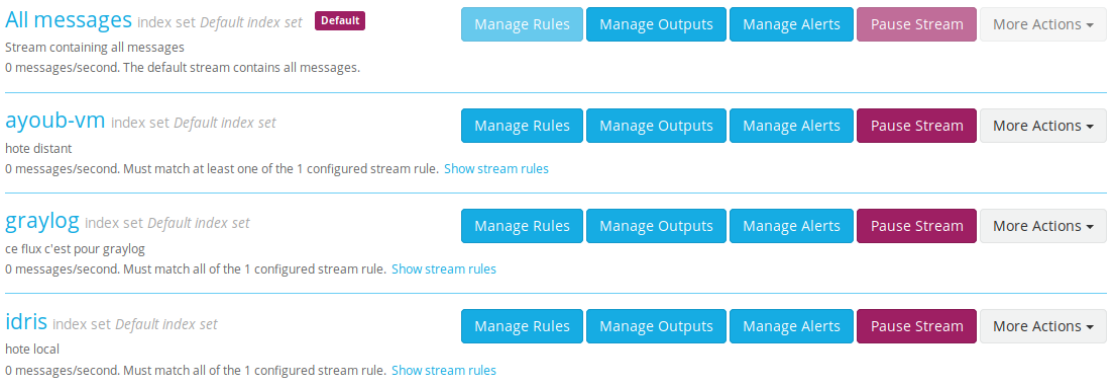



***Figure3.8.1 : CAPTURE D'ECRAN MONTRANT LE RESULTAT DE LA CREATION DE FLUX.***

## 3.9. Alerte

Les alertes sont toujours basées sur des flux. Nous pouvons définir des conditions qui déclenchent des alertes. En accédant à la section des *alertes* qui se trouve dans la barre de navigation supérieure, nous allons voir les alertes déjà configurées, les alertes déclenchées dans le passé et la possibilité de configurer de nouveaux critères d'alerte.

### 3.9.1. Conditions d'alerte

La première étape de la gestion des alertes consiste à définir des conditions d'alerte. Les conditions d'alerte spécifient les recherches que

Graylog exécutera périodiquement, ainsi que les circonstances dans lesquelles Graylog doit considérer ces résultats de recherche comme exceptionnels ; ce qui déclenche une alerte dans ce cas.

Les types de conditions d'alertes sont :

- Condition du nombre de messages :Cette condition se déclenche chaque fois que le flux reçoit plus de X messages au cours des Y dernières minutes.
- Condition d'agrégation de champ :Cette condition se déclenche chaque fois que le résultat d'un calcul statistique d'un champ de message numérique dans le flux, est supérieur ou inférieur à un seuil donné. Cette condition est la mieux adaptée pour surveiller les problèmes de performance.
- Condition du contenu du champ :Cette condition se déclenche à chaque fois que le flux reçoit au moins un message dans lequel un champ contient une valeur donnée.

### 3.9.2. Etat d'alerte

Les alertes sont des recherches périodiques pouvant déclencher certaines notifications lorsqu'une condition définie est satisfaite. Les alertes peuvent avoir deux états comme montré dans la figure 3.9.2.1 :

- Etat non résolu :Les alertes ont un état non résolu tant que la condition définie est satisfaite. Toute nouvelle alerte est déclenchée dans cet état. Ces alertes nécessitent généralement une action de notre part.
- Etat résolu :L'état résolu implique que la condition d'alerte n'est plus satisfaite. Une fois l'alerte résolue, Graylogappliquera le *délai de grâce*, en attendant un certain temps avant de créer une nouvelle alerte pour cette condition.

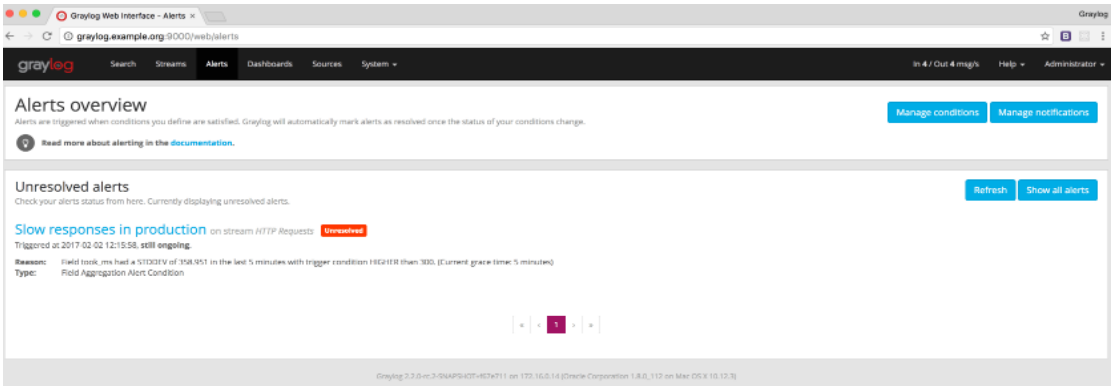



***Figure3.9.2.1: CAPTURE D'ECRAN MONTRANT UNE NOTIFICATION D'ALERTE NON RESOLU.***

Dans notre approche nous avons utilisé la condition du contenu du champ pour configurer les alertes sur les différents flux créés précédemment. Sur l'ensemble des flux nous avons spécifié des conditions d'alerte qui se déclenchent si les champs sévérité (level) du message syslog contient 0, 1 ou 2 qui signifient comme suit urgence pour 0, alerte pour 1 et critique pour 2. Nous avons aussi ajouté aux flux associés aux terminaux des conditions d'alerte qui se déclenchent si le champ fonction (facility) du message syslog contient 4, 13 ou 14 qui signifient comme suit 4 pour sécurité / autorisation (security / authorization), 13 pour log audit et 14 pour log alerte. En plus de ces conditions nous avons ajouté une autre condition d'alerte pour le flux assimilé au relaiqui se déclenchera si le message syslog contient le mot *hashingError* qui indique que le l'intégrité du message est peut-être compromise.

Graylog fournit plusieurs outils pour analyser nos résultats de recherche. Il est possible de sauvegarder ces analyses dans des tableaux de bord afin de pouvoir les vérifier au fil du temps de manière plus pratique.

## 3.10. Tableaux de bord

L'utilisation de tableaux de bord nous permet de créer des vues prédéfinies sur nos données. Elle permet de conserver une vue d'ensemble et de vérifier rapidement que tout fonctionne. Elle permet de s'organiser par équipes (équipe statistiques, équipe sécurité...).Elle permet, toujours à

titre d'exemple,l'autorisation pour un utilisateur de consulter certaines informations globales, tout en n'ayant accès qu'à certains flux.

Pour créer un tableau de bord vide il suffit d'accéder à la section *Dashboard* à l'aide du lien situé dans la barre de menus supérieure de notre interface Web Graylog. La page répertorie tous les tableaux de bord que nous sommes autorisés à afficher. Ensuite il faut appuyez sur le bouton « *Create dashboard* » pour créer un nouveau tableau de bord vide. Les seules informations requises sont un *titre* et une *description* du nouveau tableau de bord.

Dans notre solution nous avons créé des tableaux de bord pour chaque périphérique générant des messages logs, ainsi qu'un tableau de bord qui contiendrait les informations jugées pertinentes sur l'ensemble des machines connectées au serveur Graylog comme montré dans la figure suivante.

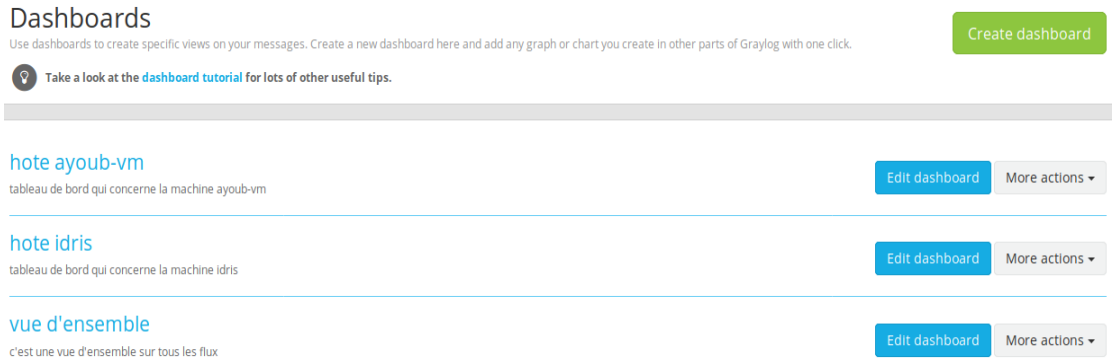



***Figure3.10.1: CAPTURE D'ECRAN MONTRANT LE RESULTAT DE LA CREATION DE TABLEAUX DE BORD.***

Les tableaux de bord crées sont initialement vides. Ils sont enrichis par l'ajout des résultats de nos recherches en quelques clics.

Nous avons ajouté les types de résultats de recherche suivants aux tableaux de bord : histogramme, valeurs statistiques et quick values.

## 3.10.1. Histogramme

La page de recherche comprend un histogramme des résultats de la recherche dans lequel nous pouvons visualiser de manière concise le nombre de messages reçus regroupés selon une période donnée comme montré dans la figure 3.10.1.1.

Dans notre solution, nous avons ajouté des histogrammes de l'ensemble des flux aux tableaux de bord qui contiennent les informations sur toutes les machines connectées au serveur Graylog. A partir des flux créés pour chaque périphérique, nous avons ajouté les histogrammes aux tableaux de bord de chaque machine.

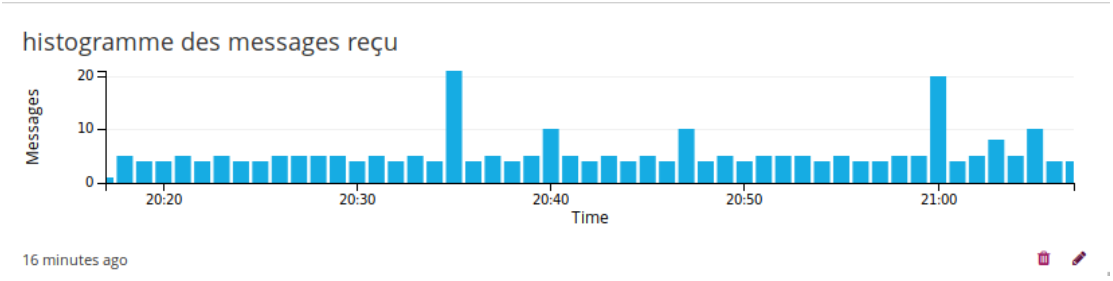
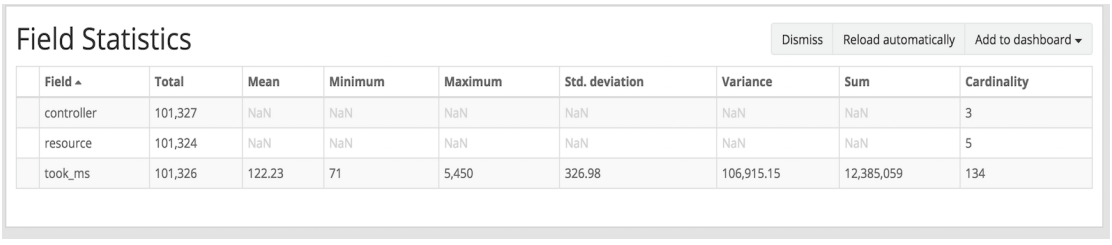



***Figure3.10.1.1: CAPTURE D'ECRAN MONTRANT UN HISTOGRAMME DES MESSAGES LOG REÇU.***

## 3.10.2. Valeurs statistiques

Les valeurs statistiques offrent la possibilité de Calculer différentes statistiques sur nos champs afin de nous aider à mieux les synthétiser et les comprendre.

Les informations statistiques comprennent les paramètres suivants : total, moyenne, minimum, maximum, écart-type, variance, somme et cardinalité comme montré dans la figure 3.10.2.1. Sur les champs non numériques c'est le nombre de valeurs uniques qu'il possède qui est considéré.




| Field | Total | Mean | Minimum | Maximum | Std. deviation | Variance | Sum | Cardinality |
|---|---|---|---|---|---|---|---|---|
| controller | 101,327 | NaN | NaN | NaN | NaN | NaN | NaN | 3 |
| resource | 101,324 | NaN | NaN | NaN | NaN | NaN | NaN | 5 |
| took_ms | 101,326 | 122.23 | 71 | 5,450 | 326.98 | 106,915.15 | 12,385,059 | 134 |

***Figure 3.10.2.1 : CAPTURE D'ECRAN MONTRANT LES DIFFERENTES INFORMATIONS STATISTIOUES SUR UN CHAMP PARTICULIER.***

Dans notre solution nous avons créé une valeur statistique qui indique le nombre d'applications en train de générer des messages logs. Il en est aussi du degré de gravité le plus bas retrouvé parmi les messages reçus. Lorsque le nombre est inférieur à trois cela indique que le système rencontre un problème et que l'administrateur devra agir rapidement. Ces valeurs sont calculées à partir des flux correspondant à chaque périphérique. Ces données statistiques sont ensuite transmises aux tableaux de bord associés à chaque machine ainsi qu'au tableau de bord principal ayant une vue d'ensemble sur les périphériques.


***Figure 3.10.2.2: CAPTURE D'ECRAN MONTRANT RESPECTIVEMENT LA SEVERITE ET LE NOMBRE D'APPLICATION GENERANT DES MESSAGES LOG.***

## 3.10.3. Résultats de quick values

Les « Quick values »ou valeurs rapides, sont des données représentées sous la forme d'un graphe type camembert nous aidant à déterminer la répartition des valeurs pour un champ. En plus d'une représentation graphique des valeurs communes contenues dans un champ, Graylog affiche un tableau avec toutes les valeurs différentes, nous permettant de voir le nombre de fois qu'elles apparaissent et les classer ainsi selon leur ordre d'apparition comme montré dans la figure 3.10.3.1.

Dans notre solution, nous avons créé des valeurs rapides (quick values) à partir des champs : sévérité (level), noms d'application (application_name) et fonction (facility) qui représente pour cette dernière la partie du système qui envoie le message. La création de ces valeurs rapides est faite à partir des flux concernant chaque périphérique qui envoie des messages logs au

serveur Graylog. Elles sont ainsi ajoutées au tableau de bord associé à chaque flux.

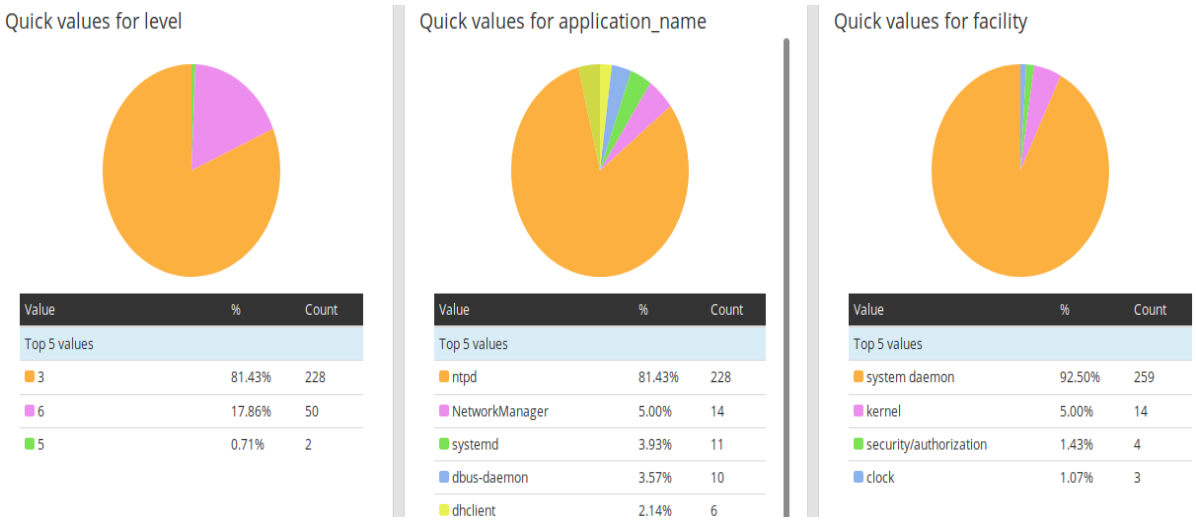



***Figure 3.10.3.1: CAPTURE D'ECRAN MONTRANT RESPECTIVEMENT LES NOMBRES RAPIDES DES CHAMPS SEVERITE, NOM D'APPLICATION ET FONCTION (FACILITY).***

## 3.11. Utilisateur et rôle

Graylog dispose d'un système de permissions granulaires qui sécurisent l'accès à ses fonctionnalités. Chaque interaction pouvant examiner des données ou modifier la configuration dans Graylog doit être effectuée en tant qu'utilisateur authentifié.

Chaque utilisateur peut avoir différents niveaux d'accès aux fonctionnalités de Graylog, qui peuvent être contrôlées en attribuant des rôles aux utilisateurs.

Sous Graylog il y a deux types d'utilisateur :

- Les administrateurs ayant tous les droits sur l'interface web :


- Les utilisateurs, qui n'ont que des droits (lecture ou modification) explicites sur des flux ou des *Dashboard*, et ne peuvent pas utiliser la recherche globale.


Dans notre solution nous avons configuré deux utilisateurs, un administrateur et un superviseur, l'administrateur à tous les droits sur l'interface web, tandis

que le superviseur n'aura que le droit de lecture comme montré dans la figure ci-dessous.

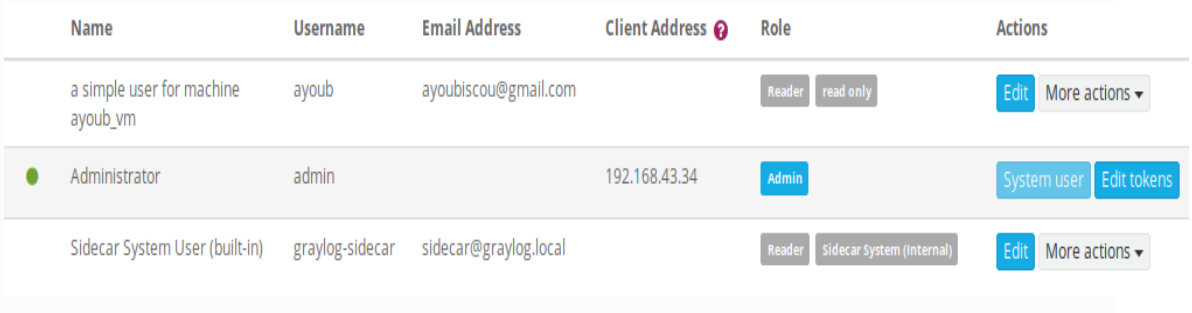



***Figure 3.11.1 : CAPTURE D'ECRAN MONTRANT LES DIFFERENTS UTILISATEURS.***

## 3.12. Diagramme de déploiement

Nous présentons ci-dessous le diagramme de déploiement qui résume la mise en place de notre système de gestion des logs :

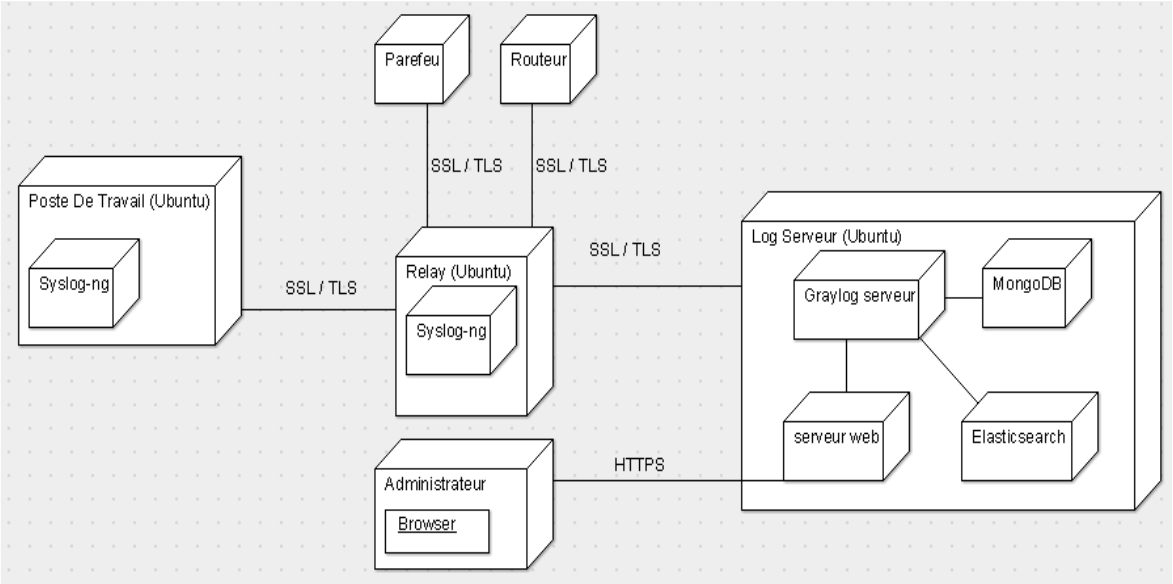



***Figure 3.12.1 : DIAGRAMME DE DEPLOIEMENT.***

# 4. ANALYSE BASEE SUR L'APPRENTISSAGE AUTOMATIQUE

## 4.1. Approches en clustering

Graylog fournit une option pour exporter les résultats d'une recherche sous format CSV qui est un séparateur à caractères spéciaux. Nous pouvons également choisir les champs qui doivent apparaitre dans le document CSV.

Le document CSV une fois établi, peut ne pas contenir de données relatives aux failles de sécurité. Cela ne permet pas de tester l'efficacité d'un algorithme d'apprentissage automatique dans la détection d'anomalies. Nous avons donc choisi d'utiliser le jeu de données NSL-KDD (known as Knowledge Discovery in Databases).

NSL-KDD est un ensemble de données de référence efficaces pour aider les chercheurs à comparer les différentes méthodes de détection d'intrusion [39].

Parmi les différentes techniques d'apprentissage automatique, nous avons choisi d'appliquer l'algorithme de clustering en utilisant la technique de K-Means clustering. Les travaux menés par [25] ont montré toute l'efficacité du clustering dans la détection d'anomalies, aboutissant à une véritable «success stories».

K-Means clustering :K-Means est l'un des algorithmes d'apprentissage non supervisé permettant de résoudre le problème bien connu du clustering.
La procédure suit un moyen simple et facile de classer un ensemble de données établies à travers un nombre de grappes fixé au préalable (exemple : K grappes). L'idée principale est de définir k centres soit un pour chaque cluster. Ces centres doivent être placés le plus loin possible les uns des autres.
L'étape suivante consiste à prendre chaque point appartenant à un ensemble de données et à l'associer au centre le plus proche. La première itération est terminée. À ce stade, nous devons recalculer k nouveaux centres de grappes résultant de l'étape précédente. Une fois que nous avons ces k nouveaux centres, une nouvelle liaison doit être établie entre les mêmes points de jeu

de données et le nouveau centre le plus proche. Une boucle a été générée. À la suite de cette boucle, nous pouvons remarquer que les k centres changent progressivement d'emplacement, jusqu'à ce qu'aucune autre modification ne puisse être apportée. Enfin, cet algorithme vise à minimiser une fonction objective appelée fonction d'erreur quadratique.

## 4.2. Description de l'environnement de test

Les expériences sont effectuées dans l'environnement WEKA (Waikato Environment for knowledge analysis) en utilisant 20% du jeu de données NSL-KDD sur une machine autonome dotée d'un processeur Core-i7 avec 8 Go de RAM.

WEKA est une collection d'algorithmes d'apprentissage automatique. Il contient des outils pour la préparation, la classification, la régression, la mise en cluster, l'exploration de règle d'association et la visualisation des données [40]. C'est un outil développé en java et peut être intégré dans Graylog.

Il y a 41 attributs décrivant différentes caractéristiques de la connexion et une étiquette attribuée à chacun d'eux en tant qu'anomalie ou en tant que normale. Le tableau 4.2.1 présente tous les attributs présents dans le jeu de données NSL-KDD.

| Total Attributs | | |
|---|---|---|
| Duration | su_attempted | same_srv_rate |
| protocol_type | num_root | diff_srv_rate |
| service | num_file_creation | srv_diff_host_rate |
| flag | num_shells | dst_host_count |
| src_byte | num_access_file | dst_host_srv_count |
| dst_byte | num_outbound cmds | dst_host_same_srv_rate |
| land | is_host_login | dst_host_diff_srv_rate |
| wrong_fragment | is_gust_login | dst_host_same_src_port_rate |
| urgent | Count | dst_host_srv_diff_host_rate |
| hot | srv_count | dst_host_serror_rate |

| | | |
|---|---|---|
| num_failed_login | serror_rate | dst_host_srv_serro_rate |
| logged in | srv_serror_rate | dst_host_rerror_rate |
| num_compromised | rerror_rate | dst_host_srv_rerror_rate |
| root_shell | srv_rerror_rate | class |

***Tableau 4.2.1 : LISTE D'ATRRIBUTS***

Le jeu de données d'apprentissage est composé de 21 anomalies sur les 37 présentes dans le jeu de données de test. Les anomalies connues sont celles qui sont présentes dans le jeu de données d'apprentissage. Les nouvelles anomalies sont des anomalies supplémentaires de l'ensemble de données de test, c'est-à-dire non disponibles dans les jeux de données d'apprentissage.

Le jeu de données d'apprentissage est constitué de 25 192 instances parmi lesquelles 13 449 sont normales et 11 743 sont des anomalies figure 4.2.1alors que les données de test sont composées de 9 711 normales et 12 833 anomalie figure 4.2.2.

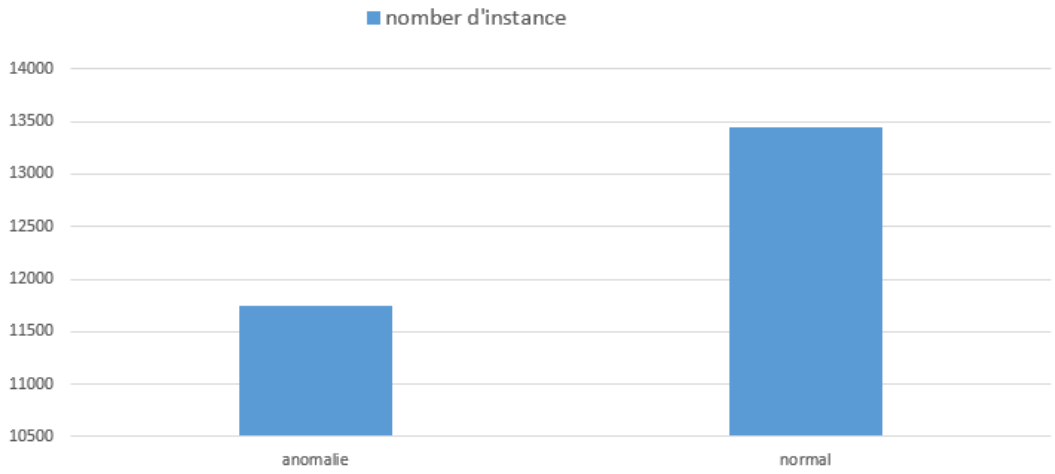



***Figure 4.2.1 : NOMBRE D'INSTANCES DANS LE JEU DE DONNEES D'APPRENTISSAGE.***

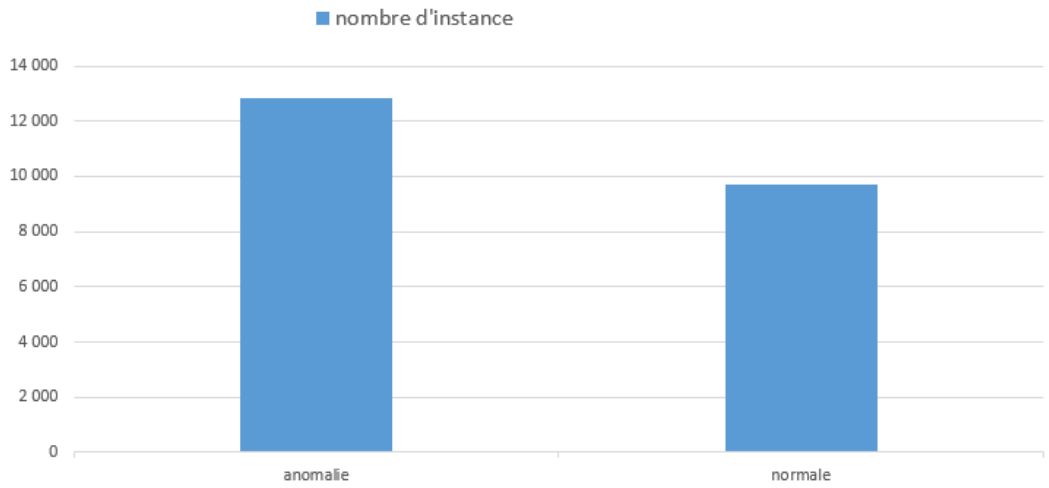



***Figure 4.2.2 : NOMBRE D'INSTANCES DANS LE JEU DE DONNEES DE TEST.***

## 4.3. Analyse des résultats

Nous analysons l'ensemble de données NSL-KDD à l'aide de l'algorithme de classification simple K-Means. La performance de l'algorithme K-Means est évaluée à l'aide de la mesure de distance euclidienne. Les instances sont réparties sur deux clusters :

- Cluster 1 : représente des logs de type anomalie.
- Cluster 2 : représente des logs normaux.

L'algorithme K-Means a mis 0,88 secondes pour traiter le jeu de données d'apprentissage et 0,82 secondes pour traiter le jeu de données de test. Le nombre d'itérations est de 8 dans chaque étape.

Le tableau suivant représente le nombre d'instance normale et anomalie dans chaque cluster.

| cluster | Nombre d'instance dans cluster | % | normal | anomalie |
|---|---|---|---|---|
| **Cluster 0** | 9555 | 38% | 65 | 9490 |
| **Cluster 1** | 15637 | 62% | 13384 | 2253 |

***Tableau 4.3.1 : DISTRIBUTION D'INSTANCES DANS LES CLUSTERS.***

Le nombre d'instances mal regroupées est de 2318.0 qui représentent un taux de 9.2%

Les résultats obtenus sont décrits dans la figure ci-dessous

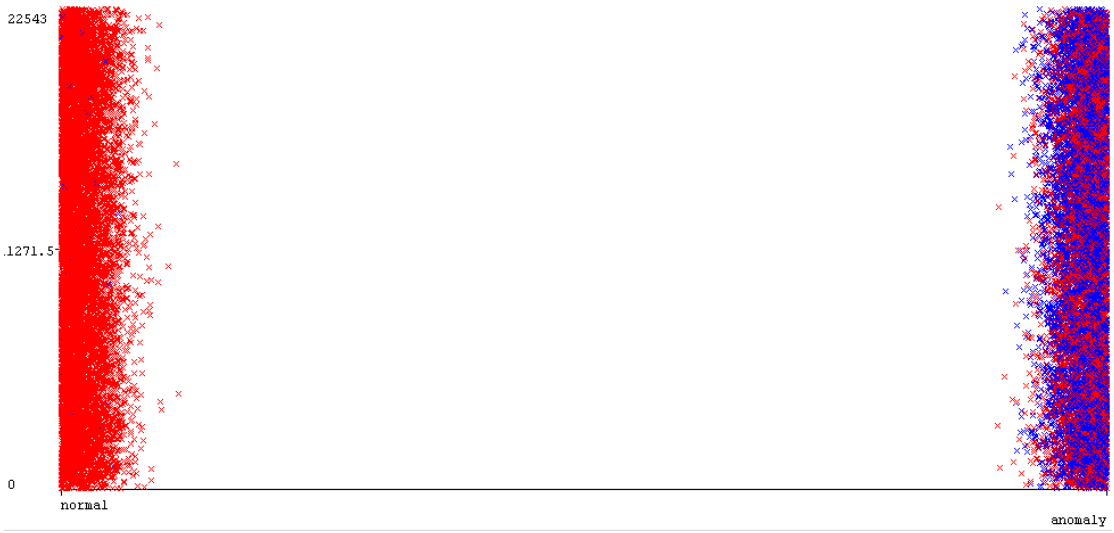



***Figure 4.3.1 : REPRESENTATION DE DEUX TYPES DE CLASSES. NORMALE ET EN ANOMALIE***

# 5. CONCLUSION

Notre système de gestion centralisé des fichiers logs comporte deux outils qui sont syslog-ng etGraylog.

syslog-ng est utilisé pour la collecte et le prétraitement de logs incluant le filtrage, la normalisation et la sécurité.

La sécurité comprise tant au niveau local en vérifiant l'intégrité des données qu'au niveau du transfert en assurant la confidentialité des messages logs, dans le respect de notre politique de journalisation.

Graylog est utilisé pour l'analyse descriptive, qui est effectuée grâce à ses nombreuses fonctionnalités intégrées nativement (tableaux de bord par utilisateur, valeur statistique, valeur rapide, histogramme). Le centralisateur utilisant ElasticSearch pour stocker l'ensemble des informations.

La configuration, la personnalisation et l'exploitation se font à travers une interface Web. Il est possible de collecter différents formats en entrée (syslog, GELF, etc).

Graylog permet de mettre en place des flux (streams) qui ont pour fonction de classer les logs selon des critères multiples. Ils permettent aussi de créer des alertes.

La partie interface Web de Graylog dispose d'une authentification où il est possible de mettre en place des accès profilés. Il est ainsi possible de présenter des tableaux de bord ou des flux (streams) à l'ensemble des personnes accédant à la plate-forme ou seulement à un groupe d'utilisateurs.

Enfin nous avons appliqué l'algorithme K-Means sur le jeu de données NSL-KDD. L'algorithme a permis de détecter efficacement les nouvelles anomalies avec un taux de succès de plus de 90%.

# CONCLUSION GENERALE

L'objectif de ce travail consisté à concevoir et à réaliser un système capable de gérer et d'analyser d'une manière sécurisée de grandes quantités de logs ayant des formats différents.

L'étude bibliographique a montrait que la gestion des logs se composait de trois phases, qui sont la génération de logs par les périphériques finaux et réseaux, leur transfert puis leur stockage à des fins d'analyse. L'étude arévélé aussi qu'il existe des outils sur le marché qui sont « open source » et qui peuvent être adaptés selon notre politique de management sécurisée des logs.

En ce qui concerne la partie hôte et relai nous avons utilisé syslog-ng qui nous a permis la collecte, le filtrage et la réécriture des différents messages logs générés par les différentes applications en format syslog. Nous lui avons attribué un champ qui contient la signature du message log ; ce qui nous a permis de vérifier leur intégrité en temps réel. syslog-ng nous a permis aussi d'utiliser SSL/TLS pour assurer la confidentialité des logs lors du transfert vers un serveur distant en passant par un relai.

Dans la partie serveur nous avons utilisé le serveur Graylog qui nous a autorisés à centraliser le stockage des logs afin de les analyser selon des méthodes statistiques, histogrammes et valeurs rapides sur les différents champs des messages logs. Il nous a été possible aussi de créer des alertes suivant notre politique de sécurité. Tout cela a été réalisé grâce à une interface web sécurisée avec des accès profilés après avoir activé le HTTPS.

Enfin nous avons appliqué un algorithme d'apprentissage automatique K-Means aux messages logs. L'algorithme K-Means qui est un algorithme de clustering, nous a permis de détecter les messages de type anomalies des messages de type normaux avec une efficacité remarquable.

A l'issue de notre travail, nous pouvons envisager l'utilisation du format GELF (Graylog Extended Log Format) qui apporte de nombreuses améliorations dont celui d'être facilement extensible par rapport au format classique syslog. Nous prévoyons également d'intégrer l'analyse avec K-Means dans la partie Graylog.

# REFERENCE


[1] Wade Baker, Mark Goudie, Alexander Hutton, C. David Hylender, Jelle Niemantsverdriet, Christopher Novak, David Ostertag, Christopher Porter, Mike Rosen, Bryan Sartin, Peter Tippett, M.D., Ph.D, 2010 Data Breach Investigations Report.

[2] Ma D, Tsudik G. A new approach to secure logging. *ACM transactions storage* 2009; **5**,1:2:1–2:21.

[3] Schneier B, Kelsey J. Security audit logs to support computer forensics. *ACM transactions on information system security* 1999;**2**,2: 159–176.

[4] Bellare M, Yee BS. Forward integrity for secure audit logs. Department of computer science, University of California, San Diego, Technical Reports, November 1997.

[5] HoltJ. E. Logcrypt : Forward security and public verification for secure audit logs. *In proceedings of the 4th australasian information security workshop*2006; 203–211.

[6] Dolev D, Yao Y. On the security of public key protocols. *IEEE transactions on information theory* 1983;**29**:2,198–208.

[7] Shamir A. How to share a secret.*ACM journal* 1979;**22**,11:612–613.

[8] Blakley G.R. Safeguarding cryptographic keys. *In Proceedings of the National Computer Conference* 1979; 313-314.

[9] Ray I, Belyaev K, Strizhov M, Mulamba D, Rajaram M. Secure logging as a service—delegating log management to the cloud. *Systems Journal, IEEE* 2013;**7**,2:323-334.

[10] C. Lonvick, "*The BSD Syslog Protocol"*, Request for Comment RFC 3164,Internet Engineering Task Force, Network Working Group, Aug. 2001.

[11] ] Indrajit Ray, Kirill Belyaev, Mikhail Strizhov, DieudonneMulamba, MariappanRajaram "Secure Logging As a Service—Delegating Log Management to the Cloud" IEEE SYSTEMS JOURNAL, VOL. 7, NO. 2, JUNE 2013.

[12] Felix Salfner and Steffen Tschirpke. Error log processing for accurate failure prediction. In USENIX Conference on Analysis of system logs, 2008.

[13] Ana Gainaru, Franck Cappello, Stefan Trausan-Matu, and Bill Kramer. Event log mining tool for large scale hpc systems. In European Conference on Parallel Processing. Springer, 2011.

[14] Qiang Fu, Jian-Guang Lou, Yi Wang, and Jiang Li. Execution anomaly detection in distributed systems through unstructured log analysis. In IEEE International Conference on Data Mining, 2009.

[15] Zhen Ming Jiang, Ahmed E Hassan, Gilbert Hamann, and Parminder Flora. An automated approach for abstracting execution logs to execution events. Journal of Software Maintenance and Evolution : Research and Practice, 20(4), 2008b.

[16] Wei Xu, Ling Huang, Armando Fox, David Patterson, and Michael I Jordan. Detecting large-scale system problems by mining console logs. In ACM SIGOPS : Symposium on Operating systems principles, 2009.

[17] Ding Yuan, Haohui Mai, Weiwei Xiong, Lin Tan, Yuanyuan Zhou, and Shankar Pasupathy. Sherlog: error diagnosis by connecting clues from run-time logs. In ACM SIGARCH : Computer Architecture News, volume 38, 2010.

[18] Xiaoyu Fu, Rui Ren, Jianfeng Zhan, Wei Zhou, Zhen Jia, and Gang Lu. Logmaster: mining event correlations in logs of large-scale cluster systems. In IEEE Symposium on Reliable Distributed Systems, 2012.

[19] Ziming Zheng, Zhiling Lan, Byung H Park, and Al Geist. System log pre-processing to improve failure prediction. In IEEE International Conference on Dependable Systems & Networks, 2009.

[20] Rakesh Agrawal, Tomasz Imieli´ nski, and Arun Swami. Mining association rules between sets of items in large databases. In ACM SIGMOD: International Conference on Management of Data, 1993.

[21] Jian-Guang Lou, Qiang Fu, Shengqi Yang, Ye Xu, and Jiang Li. Mining invariants from console logs for system problem detection. In USENIX Annual Technical Conference, 2010.

[22] Karl Pearson. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 50(302), 1900.

[23] Oded Maimon and Lior Rokach. Data mining and knowledge discovery handbook, volume 2. Springer, 2005.

[24] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In Berkeley Symposium on Mathematical Statistics and Probability, volume 1, 1967.

[25] Qingwei Lin, Hongyu Zhang, Jian-Guang Lou, Yu Zhang, and Xuewei Chen. Log clustering based problem identification for online service systems. In ACM International Conference on Software Engineering Companion, 2016.

[26] Stuart Jonathan Russell, Peter Norvig, John F Canny, Jitendra M Malik, and Douglas D Edwards. Artificial intelligence: a modern approach, volume 2. Prentice hall Upper Saddle River, 2003.

[27] Peter Bodic, Greg Friedman, Lukas Biewald, Helen Levine, George Candea, Kayur Patel, Gilman Tolle, Jonathan Hui, Armando Fox, Michael I Jordan, et al. Combining visualization and statistical analysis to improve operator confidence and efficiency for failure detection and localization. In IEEE International Conference on Autonomic Computing, 2005.

[28] Vladimir Vapnik. The nature of statistical learning theory. Springer Science & Business Media, 2013.

[29] Adetokunbo Makanju, A. Nur Zincir-Heywood, Evangelos E. Milios, clustering Event Logs Using Iterative Partitioning, 2009.

[30] J. R. Quinlan, Generating Production Rules From Decision Trees, 1987.

[31] Surbhi Aggarwal, Neha Goyal, Kirti Aggarwal, A review of Comparative Study of MD5 and SHA Security Algorithm, 2014.

[32] The syslog-ng Open Source Edition 3.13 Administrator Guide, June 19, 2018, p637

[33] Dr. Anton A. Chuvakin Kevin J. Schmidt Christopher Phillips Partricia Moulder Technical Editor. Logging and Log Management. 2014.

[34] Douglas R. Mauro and Kevin J. Schmidt. Essential SNMP. 2005.

[35] Amazon : https://aws.amazon.com/fr/elasticsearch-service/what-is-elasticsearch/

[36] mongodb: https://www.mongodb.com/what-is-mongodb?lang= fr-fr

[37] The syslog-ng Open Source Edition 3.13 Administrator Guide. Publication date June 19, 2018.

[38] Source pour la description d'algerie telecome https://www.algerietelecom.dz/fr/page/le-groupe-p2

[39] Jeu de données poue le K-Means : https://www.unb.ca/cic/datasets/nsl.html

[40] Description du weka : https://www.cs.waikato.ac.nz/ml/weka/

[41] Surbhi Aggarwal, Neha Goyal, Kirti Aggarwal. A review of Comparative Study of MD5 and SHA Security Algorithm. International Journal of Computer Applications (0975 – 8887) Volume 104 – No.14, October 2014.

[42] Risto Vaarandi, Paweł Niziński. A Comparative Analysis of Open-Source Log Management Solutions for Security Monitoring and Network Forensics. 2013.

[43] Bundy, A., Silver, B. and Plummer, D. (1985), An analytical comparison of some rule-learning programs, *Artificial Intelligence 27,* pp 137-181.

[44] J. R. Quinlan. Generating Production Rules From Decision Trees. 1987.

[45] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, “Proactive secret sharing or: How to cope with perpetual leakage,” in *Proc. 15th Ann. Int. Cryptology Conf.*, pp. 339–352. Aug. 1995.