RÉPUBLIQUE DU BÉNIN
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

**UNIVERSITÉ D'ABOMEY-CALAVI**

**INSTITUT DE FORMATION ET DE RECHERCHE EN INFORMATIQUE**

BP 526 Cotonou Tel : +229 21 14 19 88
http://www.ifri-uac.net Courriel : contact@ifri.uac.bj

# MÉMOIRE

pour l'obtention du

**Diplôme de Licence en Informatique**

**Option :** Sécurité Informatique

**Présenté par** :
Farid M. **GNONLONFOUN**

# Mise en place d'une solution HIDS open source dans une organisation : Cas de WAZUH au MDGL

**Sous la supervision** :
**Dr M. A. Tidjani SANDA MAHAMA**

**Membres du jury** :

| | | | |
|---|---|---|---|
| Eugène C. EZIN | Professeur | IFRI | Président |
| Hervé Guy M. AKAKPO | Dr | IFRI | Examinateur |
| Tidjani SANDA MAHAMA | Dr | IDRI | Rapporteur |

Année Académique : **2023-2024**

# Sommaire


Dédicace 2

Remerciements 3

Résumé 4

Abstract 5

Table des figures 6

Liste des tableaux 7

Liste des acronymes 8

Glossaire 9

Introduction 2

1 Revue de littérature 4

2 Choix de la solution et Implémentation 10

3 Tests, Résultats et Discussion 25

Conclusion 32

Bibliographie 33

Webographie 34

Table des matières 35

# Dédicace

Mes dédicaces *A mon père* ***Ganiou GNONLONFOUN***

*A ma mère* ***Victorine AMOUSSOU***

*A mes soeurs et à leurs époux*

A tous ceux qui œuvrent activement pour le développement de l'informatique en Afrique et particulièrement au Bénin.

# Remerciements

Je tiens particulièrement à remercier :

- L'administration de l'IFRI et tout le corps professoral pour la qualité de la formation reçue ces trois années ;
- Monsieur SANDA TIdjani, mon maître mémoire qui m'a supervisé tout au long de la rédaction de ce mémoire ;
- Monsieur ADIO Didier, mon maître de stage pour le temps et l'attention qu'il m'a accordés ;
- Mes chers parents qui ont toujours été là pour moi, ainsi que mes sœurs pour leur soutien et leurs encouragements ;
- Mes amis qui ont toujours été là pour moi, pour leur soutien et leurs encouragements ;
- Ainsi que tous ceux qui nous ont aidés, de près ou de loin.

# Résumé

Il est souhaitable que les équipements ou ordinateurs dans notre réseau répondent aux règles ou aux configurations qui leur sont appliquées. Il faut garantir une continuité de service et protéger aussi bien les équipements que les données. Ceci est d'autant plus important dans un environnement tel que celui du MDGL. Nous avons donc proposé la mise en place d'une solution de sécurité, un IDS basé sur l'hôte (HIDS) pour améliorer la sécurité des actifs du MDGL. La solution proposée permet la détection et une réponse aux menaces tout en assurant l'intégrité des données d'un système par l'analyse des flux relatifs à une machine ainsi que des journaux. Nous avons opté pour la solution de sécurité open source Wazuh. Après son installation et sa configuration, nous avons fait quelques tests pour évaluer sa performance.

**Mots clés** : Sécurité, IDS, HIDS, Wazuh, MDGL

# Abstract

It is desirable that the equipment or computers in our network respond to the rules or configurations applied to them. It is necessary to guarantee continuity of service and protect both equipment and data. This is all the more important in an environment such as that of the MDGL. We therefore proposed the implementation of a security solution, a host-based IDS (HIDS) to improve the security of MDGL assets. The proposed solution allows detection and response to threats while ensuring the integrity of a system's data by analyzing flows relating to a machine as well as logs. We opted for the open source security solution Wazuh. After its installation and configuration, we did some tests to evaluate its performance.

**Key words**: Sécurity, IDS, HIDS, Wazuh, MDGL

# Table des figures


1.1 Architecture NIDS[2] . . . 6
1.2 Architecture HIDS[3] . . . 6
1.3 Architecture Hybride[2] . . . 7

2.1 Flux de données[11] . . . 14
2.2 Architecture avec la solution implémentée . . . 17
2.3 Architecture avec la solution implémentée . . . 18
2.4 Architecture simplifiée . . . 18
2.5 Interface serveur Wazuh . . . 19
2.6 Tableau de bord . . . 20
2.7 Echange de clé entre le client Windows et le Serveur . . . 20
2.8 Connexion avec le serveur . . . 21
2.9 Vue des agents sur le Tableau de bord . . . 21
2.10 Création du Bot . . . 22
2.11 Vue de la plateforme avec un ID obtenue . . . 23
2.12 Un aperçu du script python avec l'ID du chat insérer . . . 23
2.13 Code d'intégrations au fichier ossec.conf avec l'api key ajoutés . . . 24
2.14 Aperçu des premiers alertes reçu dans le groupe Telegram . . . 24

3.1 Modification de la règle Wazuh . . . 25
3.2 Script bash nommée perso.sh . . . 26
3.3 Configuration du fichier ossec.conf du manager pour la gestion du AR . . . 26
3.4 Brute force SSH avec hydra sur kali . . . 27
3.5 Fichier créé grâce à l'AR . . . 27
3.6 Contenu des deux fichiers avec cat . . . 27
3.7 Alertes Telegram après l'attaque . . . 28
3.8 Code ajouté au fichier ossec.conf . . . 28
3.9 Fichier de log . . . 28
3.10 Fichier redaction.docx et classeur.xlsx ajoutés . . . 29
3.11 Suppression du fichier . . . 29
3.12 Fichier après modification . . . 29
3.13 Les alertes reçues sur Telegram . . . 30
3.14 Tableau de bord avec toutes les alertes recues . . . 30

# Liste des tableaux

1.1 Comparaison des types d'IDS [4] . . . . . 8

2.1 Comparaison entre HIDS [6] . . . . . 11
2.2 Comparaison des types d'IDS . . . . . 12

# Listes des acronymes

**API :** Application Programming Interface 7

**HIDS :** Host Intrusion Detection System 7

**HIPAA :** Health Insurance Portability and Accountability Act 7

**IDS :** Intrusion Detection System 7

**JSON :** JavaScript Object Notation 7

**MDGL :** Ministère de la Décentralisation et de la Gouvernance Locale 7

**NIDS :** Network Intrusion Detection System 7

**PCI DSS :** Payment Card Industry Data Security Standard 7

**PSSIE :** Politique de Sécurité des Systèmes d'information de l'Etat 7

**SI :** Système d'Information 7

**SIEM :** Security Information and Event Management 7

**SMTP :** Simple Mail Transfer Protocol 7

**SSH :** Secure Shell 7

**TCP/IP :** Transmission Control Protocol/Internet Protocol 7

**XDR :** Extended Détection Response 7

# Glossaire

**Agent :** un agent est un logiciel qui agit de façon autonome. C'est un programme qui accomplit des tâches à la manière d'un automate et en fonction de ce que lui a demandé son auteur. 7

**Cloud :** Le terme « cloud » désigne les serveurs accessibles sur Internet, ainsi que les logiciels et bases de données qui fonctionnent sur ces serveurs 7

**Cluster :** Dans un système informatique, un agrégat, ou « cluster », est un groupe de ressources, telles que des serveurs. Ce groupe agit comme un seul et même système Au sein d'un système informatique, un cluster de serveurs est un groupe de serveurs et d'autres ressources indépendantes fonctionnant comme un seul système 7

**Conteneur :** Un conteneur est une enveloppe virtuelle qui permet de distribuer une application avec tous les éléments dont elle a besoin pour fonctionner : fichiers sources, environnement d'exécution, librairies, outils et fichiers. Ils sont assemblés en un ensemble cohérent et prêt à être déployé sur un serveur et son système d'exploitation (OS) 7

**Firewall :** un logiciel ou un matériel dont la principale fonction est d'assurer la sécurité d'un réseau 7

**GPG :** GPG est l'acronyme de GNU Privacy Guard. Il permet le chiffrement et la signature de données 7

**Hacker :** un hacker, francisé hackeur ou hackeuse, est un spécialiste d'informatique, qui recherche les moyens de contourner les protections logicielles et matérielles.Il peut agir par curiosité, par quete de gloire, par conscience politique, contre rémunération, ou bien par vangeance ou envie de nuire 7

**Logs :** un type de fichier ou une entité équivalente, dont la mission principale consiste à stocker un historique des événements 7

**MITRE ATTACK :** MITRE ATT&CK est une base de connaissances accessible dans le monde entier sur les tactiques et techniques de l'adversaire, basée sur des observations du monde réel 7

**Protocole :** est un ensemble de règles qui régissent les échanges de données ou le comportement collectif de processus ou d'ordinateurs en réseaux ou d'objets connectés. 7

**Rootkit :** Un rootkit ou simplement « kit » (aussi appelé « outil de dissimulation d'activité », « maliciel furtif », « trousse administrateur pirate », est un ensemble de techniques mises en œuvre par un ou plusieurs logiciels, dont le but est d'obtenir et de pérenniser un accès (généralement non autorisé) à un ordinateur le plus furtivement possible à la différence d'autres logiciels malveillants 7

**Script Kiddies :** est un terme designant un débutant dépourvu de principales compétences en sécurité informatique, qui essaie d'infiltrer des systèmes en se servant de programmes efficaces – souvent des scripts –, simples d'utilisation, mais qu'il ne comprend pas 7

**Serveur :** C'est un dispositif informatique (matériel et logiciel) qui offre des services à un ou plusieurs clients 7

**Syscheck :** Syscheck est le nom du processus de vérification d'intégrité au sein d'OSSEC 7

**Syslog :** C'est un protocole définissant un service de journaux d'événements d'un système informatique 7

**Zero-day :** une faille / vulnérabilité zero-day — également orthographiée 0-day — ou faille / vulnérabilité du jour zéro est une vulnérabilité informatique n'ayant fait l'objet d'aucune publication ou n'ayant aucun correctif connu. L'existence d'une telle faille sur un produit informatique implique qu'aucune protection n'existe, qu'elle soit palliative ou définitive 7

# Introduction Générale

L'interconnexion des réseaux facilite la communication entres différents réseaux et engendre malheureusement les risques dans le domaine de la sécurité. Face à l'évolution des techniques d'attaques, il est de plus en plus difficile d'assurer la protection des systèmes informatiques. Il faut alors mettre un accent sur la protection préventive. Pour cela, une surveillance et une analyse de la sécurité en temps réel sont nécessaires.

## 1 Contexte et justification

Avec la forte digitalisation de nos administrations (publiques et privées), la sécurité des systèmes d'information devient de plus en plus indispensable car la survie de l'organisation en dépend. Conscient de ce fait, l'on ne lésine plus sur les moyens afin de protéger les infrastructures informatiques. Le parc informatique du MDGL, est constitué de terminaux interconnectés à travers plusieurs réseaux locaux. La taille du réseau le rend donc vulnérable aux attaques mettant ainsi en péril le patrimoine de l'état. .

## 2 Problématique

La sécurité des réseaux informatiques est un enjeu de taille pour les administrateurs réseaux. Les antivirus et les pares-feux, malgré leur efficacité sont limités face à l'évolution des techniques d'attaque et d'intrusion qui prennent pour cible les équipements, actifs ou terminaux dans un réseau. L'adoption d'une politique de sécurité garantissant l'intégrité des actifs du réseau surveillé est alors indispensable. Le MDGL, ainsi que le recommande la PSSIE doit donc mettre en œuvre des actions de protection de ses actifs.

## 3 Objectifs

L'objectif de notre travail est de proposer une solution de détection d'intrusion basée sur l'hôte(la machine).
De façon spécifique, nous aurons à :

- Identifier la meilleure solution de protection des systèmes ;
- Installer et configurer les composants de la solution ;
- Intégrer la fonctionnalité de réception des alertes par Télégram ;
- Evaluer la solution .

# 4 Organisation du mémoire

Le présent travail s'articule autour de trois grands chapitres. Le premier chapitre aborde la revue de littérature sur le thème. Quant au deuxième chapitre, il présente la solution choisie et son implémentation. Enfin, le dernier et troisième chapitre montre les résultats des simulations et tests.

Chapitre 1

# Revue de littérature

## Introduction

Pour mener à bien le présent projet, il est nécessaire d'effectuer un état de l'art qui est une connaissance existante, à un moment donné, un objet d'étude. Dans ce chapitre, nous allons commencer par donner une vue d'ensemble de la sécurité informatique avant de nous plonger dans l'univers des systèmes de détection d'intrusion.

## 1 Généralités

### 1.1 Défintion

- **Réseau informatique**
  Est un ensemble d'équipements reliés entre eux grâce à des protocoles pour échanger des informations et partager des ressources
- **Sécurité informatique**
  C'est l'ensemble des moyens mis en œuvre pour minimiser la vulnérabilité d'un système contre des menaces accidentelles ou intentionnelles.
- **Parc informatique [1]**
  Désigne la globalité des ressources qui composent le système informatique d'une entreprise.
- **Système d'information**
  Est un ensemble de ressources et de dispositifs permettant de collecter, stocker, traiter et diffuser les informations nécessaires au fonctionnement d'une organisation.

### 1.2 Les objectifs de la sécurité informatique

La sécurité informatique, de manière générale, vise à s'assurer que les ressources matérielles ou logicielles d'une organisation ou entreprise sont utilisées par les personnes légitimes, aux heures prescrites et dans le cadre prévu.
La sécurité informatique vise généralement cinq principaux objectifs :

- **Intégrité** garantir que les informations ne seront modifiées que par les personnes autorisées
- **Confidentialité** garantir que les informations qui se trouvent dans le système soient lues que par les personnes autorisées
- **Disponibilité** garantir que les informations qui se trouvent dans le système soient disponibles pour les personnes autorisées
- **Non répudiation** permet de garantir qu'une transaction ou opération ne sera pas niée
- **Authentification** consiste à s'assurer que seules les personnes autorisées aient accès aux ressources

## 1.3 Système de détection d'intrusion

### 1.3.1 Définition

Un système de détection d'intrusion (ou IDS) est un mécanisme destiné à repérer des activités anormales ou suspectes sur la cible analysée (un réseau ou un hôte). Il permet ainsi d'avoir une connaissance sur les tentatives réussies comme échouées des intrusions.

### 1.3.2 Fonctionnement des IDS

Un système de détection dispose de deux fonctions : le mode de détection utilisé et la réponse apportée par le système de détection d'intrusion lors de la détection d'une intrusion.

#### 1.4.2.1 Méthodes de détection

Pour le mode de détection, nous avons la détection d'anomalies (encore appelée approche comportementale) et la reconnaissance de signature (encore appelée approche par scénario). Cependant ces méthodes peuvent être combinées au sein d'un même système afin d'accroitre la sécurité

- **Approche par scénario ou signature**
  Elle consiste à rechercher dans l'activité de l'élément surveillé, les empreintes d'attaques connues. L'IDS est purement réactif, il ne peut détecter que les attaques dont il possède la signature. Pour cela, il nécessite des mises à jour fréquentes. L'efficacité de ce système de détection réside dans la précision de sa base de signature.
- **Approche comportementale ou anomalie**
  La détection d'anomalies consiste à détecter des anomalies par rapport à un trafic habituel. La mise en place comprend toujours une phase d'apprentissage au cours de laquelle les systèmes de détection d'intrusion vont découvrir le fonctionnement normal des éléments surveillés. Ils sont en mesure de signaler les divergences par rapport au fonctionnement normal.

#### 1.3.2.2 Type de réponses

Il existe deux types de réponses, suivant les IDS utilisés. La réponse passive est disponible pour tous les IDS, la réponse active est plus ou moins implémentée.

- **Réponse passive**
  La réponse passive consiste à enregistrer les intrusions détectées dans un fichier log qui sera analysé par le responsable sécurité. Lorsqu'une attaque est détectée, le système d'intrusion ne prend aucune action ; il génère seulement une alarme en direction de l'administrateur système sous forme d'une alerte lisible qui contient les informations à propos de chaque attaque.

- **Réponse active**
  La réponse active au contraire a pour but de stopper une attaque au moment de sa détection. Elle implique des actions automatisées prises par un IDS qui permet de couper rapidement une connexion suspecte quand le système détecte une intrusion.

### 1.3.3 Différents type d'IDS

Les différents IDS se caractérisent par leur domaine de surveillance. Il existe trois grandes familles distinctes d'IDS :

#### 1.3.3.1 Systèmes de détection d'intrusion réseau(NIDS)

Les NIDS sont des IDS utilisés pour protéger un réseau. Ils comportent généralement une sonde (machine par exemple) qui écoute et surveille en temps réel tout le trafic réseau, puis analyse et génère des alertes s'il détecte des intrusions ou des paquets semblent dangereux

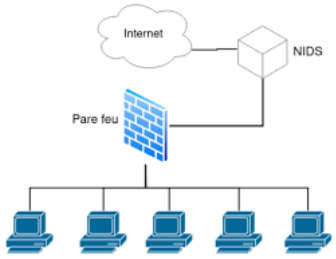



FIGURE 1.1 : Architecture NIDS[2]

#### 1.4.3.2 Système de détection d'intrusion basé sur l'hôte(HIDS)

La détection d'intrusion basée sur l'hôte analyse l'information concernant cet hôte. Le but de ce type de système de détection d'intrusion est d'assurer l'intégrité des données d'un système et d'analyser le flux relatif à une machine ainsi que ses journaux

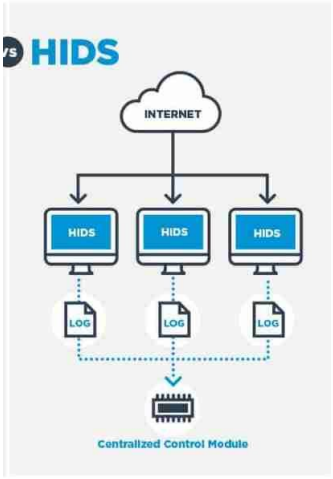



FIGURE 1.2 : Architecture HIDS[3]

#### 1.4.3.3 Système de détection Hybride

Les IDS hybrides rassemblent les caractéristiques des NIDS et HIDS. Ils permettent de surveiller le réseau et les terminaux. Les sondes sont placées en des points stratégiques et agissent comme NIDS et/ou HIDS suivant leurs emplacements. Toutes ces sondes remontent alors les alertes à une machine qui va centraliser le tout, et lier les informations d'origines multiples. Ainsi, on comprend que les IDS

hybrides sont basés sur une architecture distribuée, où chaque composant unifie son format d'envoi. Cela permet de communiquer et d'extraire des alertes plus exactes.

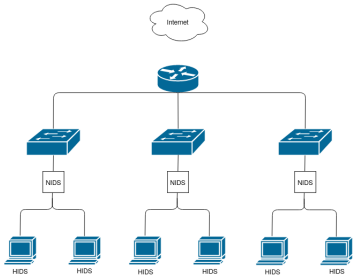



FIGURE 1.3 : Architecture Hybride[2]

**Comparaison entre types IDS**

TABLE 1.1 : Comparaison des types d'IDS [4]

| Types IDS | Avantages | Inconvénients |
|---|---|---|
| **NIDS** | - Détecter plus facilement les scans grâce aux signatures ;<br>- Surveiller un grand réseau ;<br>- Identifier les attaques de multiples hôtes ;<br>- Assurer la sécurité contre les attaques puisqu'il est invisible. | - La probabilité de faux négatifs (attaques non détectées) est élevée et il est difficile de contrôler le réseau entier ;<br>- Ils doivent principalement fonctionner de manière cryptée, d'où une complication de l'analyse des paquets ;<br>- Ne permet ni de savoir si une tentative d'attaque est couronnée de succès, ni connaitre ces impacts. |
| **HIDS** | - Contrôler les activités locales des utilisateurs avec précision ;<br>- Capable de déterminer si une tentative d'attaque est couronnée de succès ;<br>- Détecter des attaques qui ne sont pas vues par NIDS. | - Ils ont moins de facilité à détecter les scans ;<br>- Incapable de détecter des attaques contre de multiples cibles dans le réseau ;<br>- La difficulté de déploiement et de gestion, surtout lorsque le nombre d'hôtes qui ont besoin de protection est large. |
| **HYBRIDE** | - Moins de faux positifs ;<br>- Meilleure corrélation (la corrélation permet de générer de nouvelles alertes à partir de celles existantes) ;<br>- Possibilité de réaction sur les analyseurs. | |

## Conclusion

Dans ce chapitre, nous avons présenté un aperçu sur la sécurité informatique en précisant ces objectifs avant de proposer quelques solutions existantes afin de se protéger, dont notre solution à développer dans le prochain chapitre.

Chapitre 2

# Choix de la solution et Implémentation

## Introduction

Comme nous l'avons vu dans le chapitre qui précède, les IDS font partie des outils utilisés pour la sécurisation d'un réseau ou d'un hôte. Dans ce chapitre, nous présenterons d'abord les systèmes de détection d'intrusion basés sur l'hôte. Ensuite, nous aurons à présenter la solution de notre choix et enfin, nous installerons les différents composants du HIDS ainsi que toutes les configurations nécessaires.

## 1 Choix de la Solution Implémentée

### 1.1 Présentation de quelques HIDS Open source les plus connus [5].

Face aux menaces d'intrusion, il existe plusieurs solutions concernant le choix d'un IDS. Il existe des solutions commerciales aussi bien qu'Open Source(gratuit). Comme solution Open Source, nous avons :

- **OSSEC (Open Source HIDS SECurity)** est un système de détection d'intrusion basé sur l'hôte (HIDS) gratuit et open source. Il effectue une analyse des journaux , une vérification de l'intégrité, une surveillance du registre Windows , une détection des rootkits , une alerte basée sur le temps et une réponse active. Il fournit une détection d'intrusion pour la plupart des systèmes d'exploitation, y compris Linux , OpenBSD , FreeBSD , OS X , Solaris et Windows.
- **Tripwire** Comme son nom l'indique, il s'agit d'un système de détection basé sur l'hôte gratuit et open-source développé par Tripwire. Le package Tripwire open source fonctionne exclusivement sur presque toutes les distributions Linux. Connu pour ses grandes capacités à assurer l'intégrité des données , il aide les administrateurs système à détecter les altérations des fichiers système et les avertit en cas de fichiers corrompus ou falsifiés
- **Wazuh** est une autre solution de surveillance open source pour la surveillance de l'intégrité, la réponse aux incidents et la conformité. Wazuh utilise des méthodes de détection d'anomalies

et de signatures pour détecter les rootkits en plus d'effectuer une analyse des journaux, une vérification de l'intégrité, une surveillance du registre Windows et une réponse active. Ses agents fonctionnent sur les systèmes d'exploitation Windows, Linux, Solaris, BSD et Mac

- **Samain** est un HIDS open source avec une gestion centralisée qui vous aide à vérifier l'intégrité des fichiers, à surveiller les fichiers journaux et à détecter les processus cachés. Cette solution multi-plateforme fonctionne sur les systèmes POSIX ( Unix, Linux, Cygwin/Windows ).

**Comparaison entre les HIDS**

TABLE 2.1 : Comparaison entre HIDS [6]

| Solutions | Mise à jour | Dépôts officiels d'Ubuntu | Dépôts Officiel CentOS | Dossier | Réseau | Journaux | Configuration |
|---|---|---|---|---|---|---|---|
| **Wauh** | 2022 | Non | Non | Oui | Oui | Oui | Oui |
| **OSSEC** | 2019 | Non | Non | Oui | Oui | Oui | Oui |
| **Samhain** | 2021 | Oui | Non | Oui | Non | Partiel | |
| **Tripwire** | 2018 | Oui | Oui | Oui | Non | Non | |

## 1.2 Critère de choix d'un IDS [7]

Les systèmes de détection d'intrusion sont devenus indispensables lors de la mise en place d'une infrastructure de sécurité opérationnelle. Ils s'intègrent donc toujours dans un contexte et dans une architecture imposante des contraintes très diverses. Certains critères imposant le choix d'un IDS peuvent être dégagés :

- **Fiabilité :**Les alertes générées doivent être justifiées et aucune intrusion ne doit pouvoir lui échapper.
- **Réactivité :**Un IDS doit être capable de détecter les nouveaux types d'attaques le plus rapidement possible ; pour cela, il doit rester constamment à jour. Des capacités de mise à jour automatique sont indispensables.
- **Facilité de mise en œuvre et adaptabilité :**Un IDS doit être facile à mettre en œuvre et surtout s'adapter au contexte dans lequel il doit opérer. Il est inutile d'avoir un IDS émettant des alertes en moins de 10 secondes si les ressources nécessaires à une réaction ne sont pas disponibles pour agir dans les mêmes contraintes de temps.
- **Performance :**La mise en place d'un IDS ne doit en aucun cas affecter les performances des systèmes surveillés. De plus, il faut toujours avoir la certitude que l'IDS a la capacité de traiter toute l'information à sa disposition (par exemple un IDS réseau doit être capable de traiter l'ensemble du flux pouvant se présenter à un instant donné sans jamais supprimer de paquets) car dans le cas contraire, il devient trivial de masquer les attaques en augmentant la quantité d'information.

- **Multicanal :** Un bon IDS doit pouvoir utiliser plusieurs canaux d'alerte (mail, pager, téléphone, fax...) afin de pouvoir garantir que les alertes seront effectivement émises. Information : L'IDS doit donner un maximum d'information sur l'attaque détectée afin de préparer la réaction.

- **Classification :**il doit être aisé de hiérarchiser la gravité des attaques détectées afin d'adapter le mode d'alerte.

**Comparaison par rapport aux critères**

TABLE 2.2 : Comparaison des types d'IDS

| Critères | Fiabilité | Réactivité | Facilité de mise en œuvre et adaptabilité | Performance | Multicanal | Classification |
|---|---|---|---|---|---|---|
| **Wauh** | Oui | Oui | Oui | Oui | Oui | Oui |
| **OSSEC** | Oui | Oui | Oui | Oui | Oui | Oui |
| **Samhain** | Oui | Oui | Non | Oui | Oui | Oui |
| **Tripwire** | Oui | Oui | Oui | Oui | Oui | |

**Autres Critère de choix [8]**
Il est important de choisir une solution IDS qui :
- Répond à nos besoins uniques.
- Peut être pris en charge par votre réseau.
- Convient à votre budget.
- Est compatible avec les systèmes filaires et sans fil.
- Peut être mis à l'échelle.
- Permet une interopérabilité accrue.
- Comprend des mises à jour de signature.
Dans notre production, nous avons pris Wazuh comme solution.

**Quelques Avantages de wazuh [9]**
Certains des avantages les plus appréciés :

- Pas de verrouillage du fournisseur
- Pas de frais de licence
- Utilise des agents légers et multiplateformes
- Assistance communautaire gratuite active
- Il est très facile d'intégrer Wazuh à d'autres environnements, les services cloud, les conteneurs et applications sur site. L'avantage est donc qu'il est facile à mettre en œuvre et à intégrer à d'autres solutions.

- Il peut être utilisé, agrégé et analysé les données provenant de source externe
- Fiabilité et évolutivité
- Utilise des méthodes de détection d'anomalies et de signatures pour détecter les rootkits
- Wazuh fournit les contrôles de conformité des normes telles que PCI DSS, HIPAA, GDPR et autres
- Il étant les fonctionnalités de OSSEC
- Intègre à la suite ELK (Elasticsearch Logstash Kibana)

**Choix de solution**

Après la comparaison (en utilisant les critères de choix) des différentes solutions sur les intrusions basées sur l'hôte, nous avons décidé de mettre en place Wazuh. Il est un logiciel open source de détection d'intrusion (HIDS). Il présente plus d'avantages et moins d'inconvénients (certains inconvénients des HIDS), extrêmement fiables, facile à mettre en place, une interface de lecture facile pour l'administrateur. C'est pour cette raison que nous l'avons choisi comme solution de sécurité open source.

## 1.3 Présentation générale de Wazuh [10]

Wazuh est une plate-forme de sécurité gratuite et open source qui fournit des fonctionnalités XDR et SIEM pour protéger vos charges de travail cloud, conteneur et serveur. Celles-ci incluent l'analyse des données de journal, la détection des intrusions et des logiciels malveillants, la surveillance de l'intégrité des fichiers, l'évaluation de la configuration, la détection des vulnérabilités et la prise en charge de la conformité réglementaire.

### 1.3.1 Composants

La solution Wazuh est basée sur l'agent Wazuh, qui est déployé sur les terminaux surveillés, et sur trois composants centraux : le serveur Wazuh, l'indexeur Wazuh et le tableau de bord Wazuh. Outre les capacités de surveillance basées sur les agents, la plate-forme Wazuh peut surveiller les périphériques sans agent tels que les pares-feux, les commutateurs, les routeurs ou les IDS réseau, entre autres.

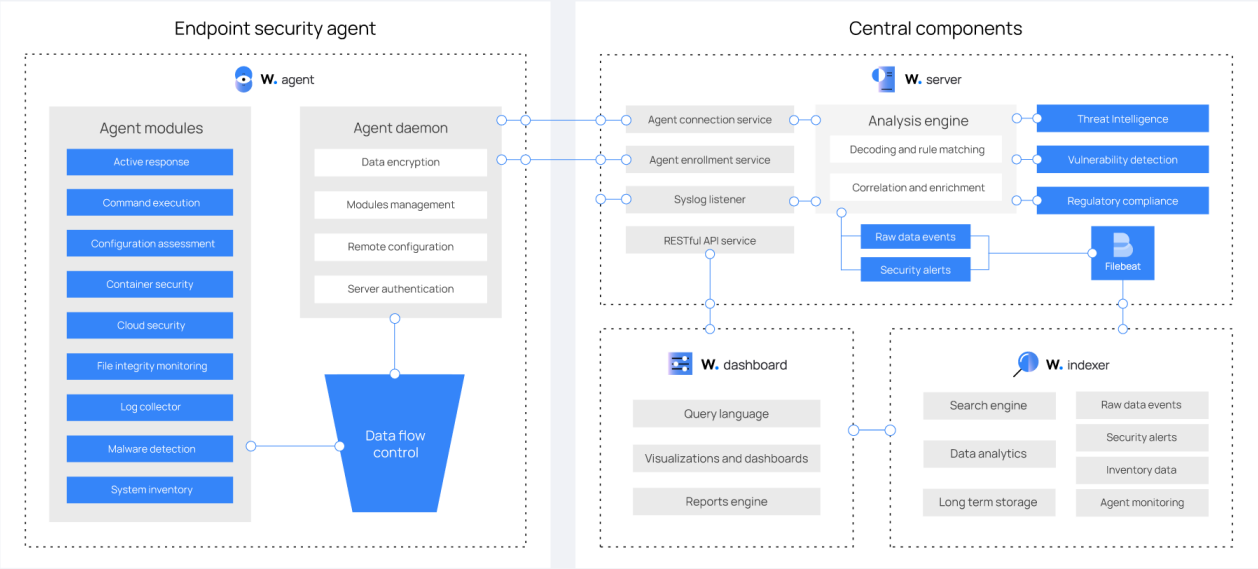



FIGURE 2.1 : Flux de données[11]

#### 2.4.1.1 Indexeur Wazuh

L'indexeur Wazuh est un moteur de recherche et d'analyse en texte intégral hautement évolutif. Ce composant central Wazuh indexe et stocke les alertes générées par le serveur Wazuh et fournit des capacités de recherche et d'analyse de données en temps quasi réel.

#### 2.3.1.2 Serveur Wazuh

Le composant serveur Wazuh analyse les données reçues des agents, déclenchant des alertes lorsque des menaces ou des anomalies sont détectées. Il est également utilisé pour gérer la configuration des agents à distance et surveiller leur état.

**Architecture serveur**

Le serveur Wazuh exécute le moteur d'analyse, l'API RESTful Wazuh, le service d'inscription d'agent, le service de connexion d'agent, le démon de cluster Wazuh et Filebeat. **Composants du serveur**
Le serveur Wazuh comprend plusieurs composants répertoriés ci-dessous qui ont différentes fonctions, telles que l'inscription de nouveaux agents, la validation de l'identité de chaque agent et le cryptage des communications entre l'agent Wazuh et le serveur Wazuh.

- Service d'inscription d'agent : Il est utilisé pour inscrire de nouveaux agents.
- Service de connexion des agents : Ce service reçoit les données des agents.
- Moteur d'analyse : il s'agit du composant serveur qui effectue l'analyse des données.
- API Wazuh RESTful : ce service fournit une interface pour interagir avec l'infrastructure Wazuh.
- Démon de cluster Wazuh : ce service est utilisé pour mettre à l'échelle horizontalement les serveurs Wazuh, en les déployant en tant que cluster.
- Filebeat : Il est utilisé pour envoyer des événements et des alertes à l'indexeur Wazuh.

#### 2.3.1.3 Tableau de bord Wazuh

Le tableau de bord Wazuh est une interface utilisateur Web flexible et intuitive pour l'exploration,

l'analyse et la visualisation des événements de sécurité et des données d'alerte.

**Visualisation et analyse des données**

L'interface Web aide les utilisateurs à naviguer à travers les différents types de données collectées par l'agent Wazuh, ainsi que les alertes de sécurité générées par le serveur Wazuh.

**Surveillance et configuration des agents**

Le tableau de bord Wazuh permet aux utilisateurs de gérer la configuration des agents et de surveiller leur statut.

**Gestion de la plateforme**

Le tableau de bord Wazuh fournit une interface utilisatrice dédiée à la gestion de votre déploiement Wazuh.

**Outil de développement**

Le tableau de bord Wazuh comprend un outil de test d'ensemble de règles qui peut traiter les messages du journal pour vérifier comment il est décodé et s'il correspond ou non à une règle de détection des menaces.

#### 2.3.1.4 Agent Wazuh

L'agent Wazuh est un logiciel de surveillance unique et léger qui s'exécute sur Linux, Windows, macOS, Solaris, AIX et d'autres systèmes d'exploitation. Il peut être déployé sur des ordinateurs portables, des ordinateurs de bureau, des serveurs, des instances cloud, des conteneurs ou des machines virtuelles.

**Architecture des agents**

L'agent Wazuh a une architecture modulaire. Chaque composant est responsable de ses propres tâches, notamment la surveillance du système de fichiers, la lecture des messages du journal, la collecte des données d'inventaire, l'analyse de la configuration du système et la recherche de logiciels malveillants.

**Modules agents**

Tous les modules d'agent sont configurables et exécutent différentes tâches de sécurité. Ci-dessous, vous pouvez en savoir plus sur les différents objectifs de tous les modules d'agent.

- Collecteur de journaux : ce composant d'agent peut lire les fichiers journaux plats et les événements Windows, en collectant les messages du système d'exploitation et du journal des applications.
- Exécution des commandes : les agents exécutent périodiquement des commandes autorisées, collectent leur sortie et la signalent au serveur Wazuh pour une analyse plus approfondie.
- Surveillance de l'intégrité des fichiers (FIM) : ce module surveille le système de fichiers, signalant la création, la suppression ou la modification de fichiers.
- Evaluation de la conformité de la sécurité (SCA) : ce composant fournit une évaluation continue de la configuration, en utilisant des vérifications prêtes à l'emploi basées sur les références du Center of Internet Security (CIS).
- Inventaire du système : ce module d'agent exécute périodiquement des analyses, collectant des données d'inventaire telles que la version du système d'exploitation, les interfaces réseau, les processus en cours d'exécution, les applications installées et une liste des ports ouverts.

- Détection des malwares : Utilisant une approche non basée sur les signatures, ce composant est capable de détecter les anomalies et la présence éventuelle de rootkits.
- Réponse active : ce module exécute des actions automatiques lorsque des menaces sont détectées, déclenchant des réponses pour bloquer une connexion réseau, arrêter un processus en cours d'exécution ou supprimer un fichier malveillant.
- Surveillance de la sécurité des conteneurs : ce module d'agent est intégré à l'API Docker Engine pour surveiller les modifications dans un environnement conteneurisé.
- Surveillance de la sécurité du cloud : ce composant surveille les fournisseurs de cloud tels qu'Amazon AWS, Microsoft Azure ou Google GCP.

**Communication avec le serveur Wazuh**
L'agent Wazuh communique avec le serveur Wazuh pour expédier les données collectées et les événements liés à la sécurité.

### 1.3.2 Architecture et fonctionnement de Wazuh

L'architecture Wazuh est basée sur des agents, exécutés sur les terminaux surveillés, qui transmettent les données de sécurité à un serveur central. Les périphériques sans agent tels que les pares-feux, les commutateurs, les routeurs et les points d'accès sont pris en charge et peuvent soumettre activement des données de journal via Syslog, SSH ou à l'aide de leur API.

**Agent Wazuh - Communication serveur Wazuh**
L' agent Wazuh envoie en continu des événements au serveur Wazuh pour analyse et détection des menaces. Les événements peuvent être mis en file d'attente dans l'un ou les deux fichiers suivants, selon qu'une règle est déclenchée ou non

- Le fichier **/var/ossec/logs/archives/archives.json** contient tous les événements, qu'ils aient déclenché une règle ou non.
- Le fichier **/var/ossec/logs/alerts/alerts.json** contient uniquement les événements qui ont déclenché une règle avec une priorité suffisamment élevée (le seuil est configurable).

**Serveur Wazuh - Communication avec indexeur Wazuh**
Le serveur Wazuh utilise Filebeat pour envoyer des données d'alerte et d'événement à l'indexeur Wazuh. Filebeat lit les données de sortie du serveur Wazuh et les envoie à l'indexeur Wazuh. Une fois les données indexées par l'indexeur Wazuh, le tableau de bord Wazuh est utilisé pour extraire et visualiser les informations. Le tableau de bord Wazuh interroge l'API Wazuh RESTful pour afficher les informations de configuration et d'état du serveur et des agents Wazuh .

## 1.4 Mise en place de la solution

### 1.4.1 Architecture

**Architecture simplifiée du réseau interne du MDGL**
Nous avons proposé une architecture simplifiée du réseau interne du MDGL. Cette architecture va énumérer les équipements dans le réseau

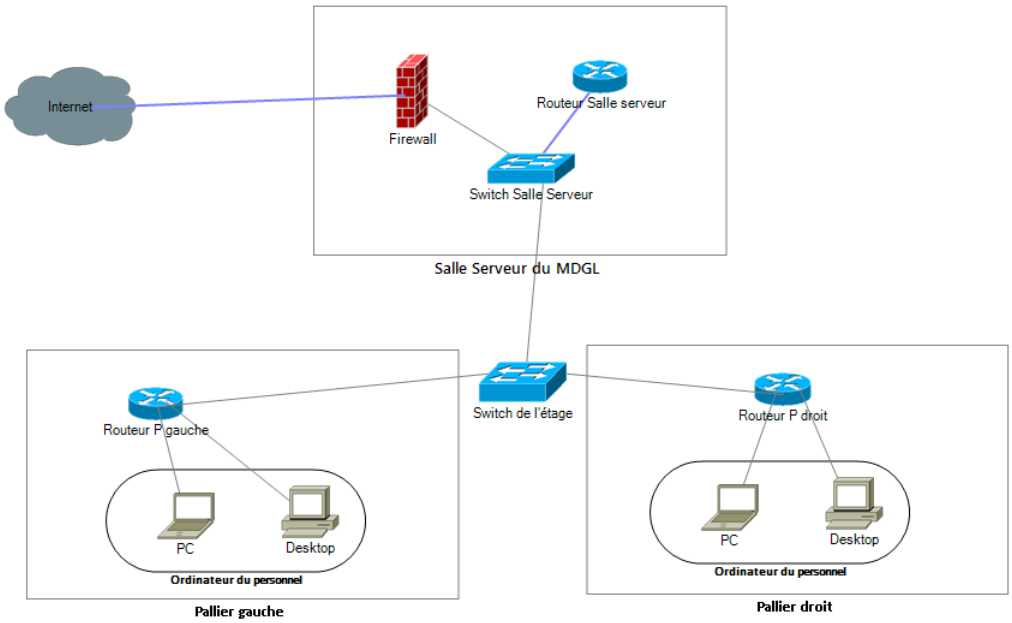



FIGURE 2.2 : Architecture avec la solution implémentée

Pour une description un peu globale du réseau du MDGL, nous pouvons dire qu'il est composé essentiellement de switch, de routeur, d'un firewall et des machines. L'internet arrive dans le réseau (sur le switch de la salle serveur) par une fibre optique et un pare-feu s'occupe de filtrer le trafique entrant et sortant du réseau. Le réseau est subdivisé en plusieurs sous-réseau, chaque sous-réseau se trouve sur l'un des six étages que compte le bâtiment du MDGL. Dans chaque sous-réseau se trouve au moins un Switch qui sert d'élément centrale et au moins de deux routeurs à laquelle sont reliés directement ou indirectement les ordinateurs du personnels (par câble Ethernet ou par WIFI).

**Proposition d'architecture avec la solution implémentée**

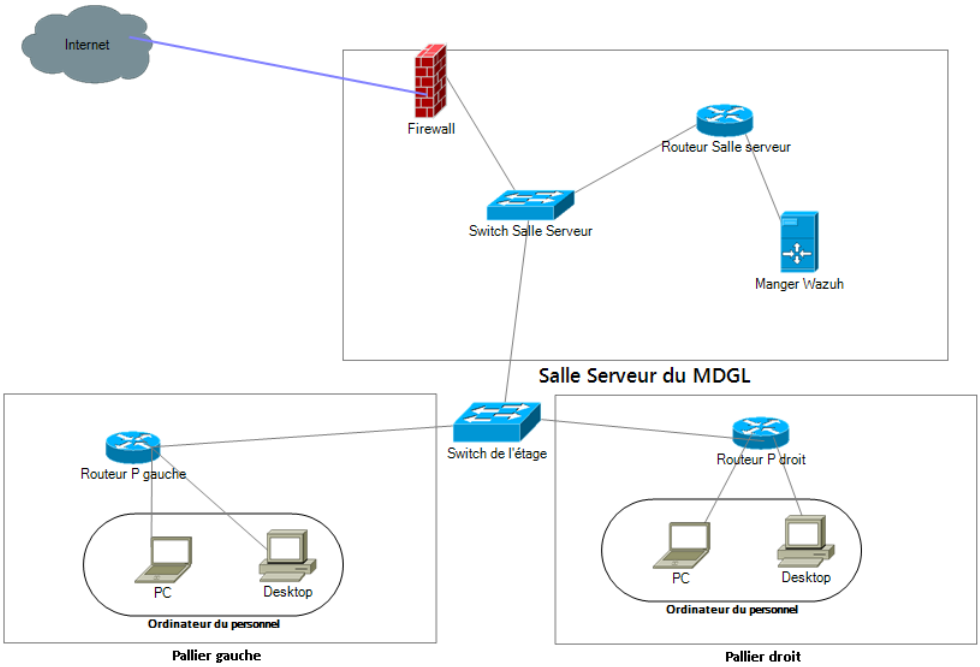



FIGURE 2.3 : Architecture avec la solution implémentée

Etant donné que notre solution est basée sur des agents qui surveillent les actifs et un gestionnaire qui analyse les données reçues des agents et déclenchant des alertes lorsque des menaces ou des anomalies sont détectées, il nous faudra juste installer les agents sur les actifs à surveiller et mettre notre serveur dans un endroit isolé de sorte que seul l'administrateur puisse y accéder tout en assurant qu'il y a échange entre les agents et le serveur

**Architecture simplifié**

Pour notre test, nous avons installé sur notre ordinateur personnel avec un système win10, le serveur wazuh comme machine virtuelle ainsi que deux autres machines virtuelle (Ubuntu et kali). La machine kali nous servira d'attaquant et nous connecterons les deux autres au serveur (le système principal win10 et l'Ubuntu).

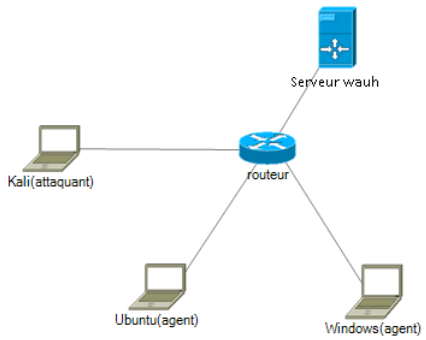



FIGURE 2.4 : Architecture simplifiée

### 1.4.2 Outils utilisés

- VMware Workstation 16 Pro
- Machine virtuelle OVA de Wazuh qui contient les composants suivants :
  o Cent OS 7
  o Gestionnaire Wazuh
  o Indexeur Wazuh
  o Filebeat
  o Tableau de bord Wazuh
- Wazuh agent pour Windows et Ubuntu
- Ubuntu OS en machine virtuel
- Machine kali jouant le rôle d'attaquant (en virtuel)

### 1.4.3 Installation de l'environnement et configuration

**1. Importation du serveur Wazuh**
Nous avons fait l'importation de la machine virtuelle à utiliser
**2. Interface du serveur Wazuh**
Après l'importation, nous avons :
Lancé le démarrage du serveur
Ensuite, nous avons entré le login et le mot de passe du serveur
Et nous avons obtenu l'adresse ip du serveur avec la commande **ip add** (**ip a** en abrégé)

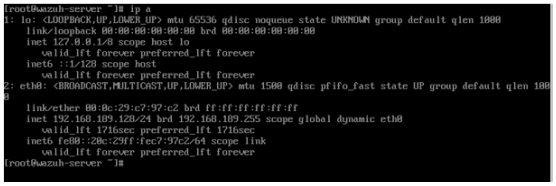



FIGURE 2.5 : Interface serveur Wazuh

### 3. Lancement du Tableau de Bord

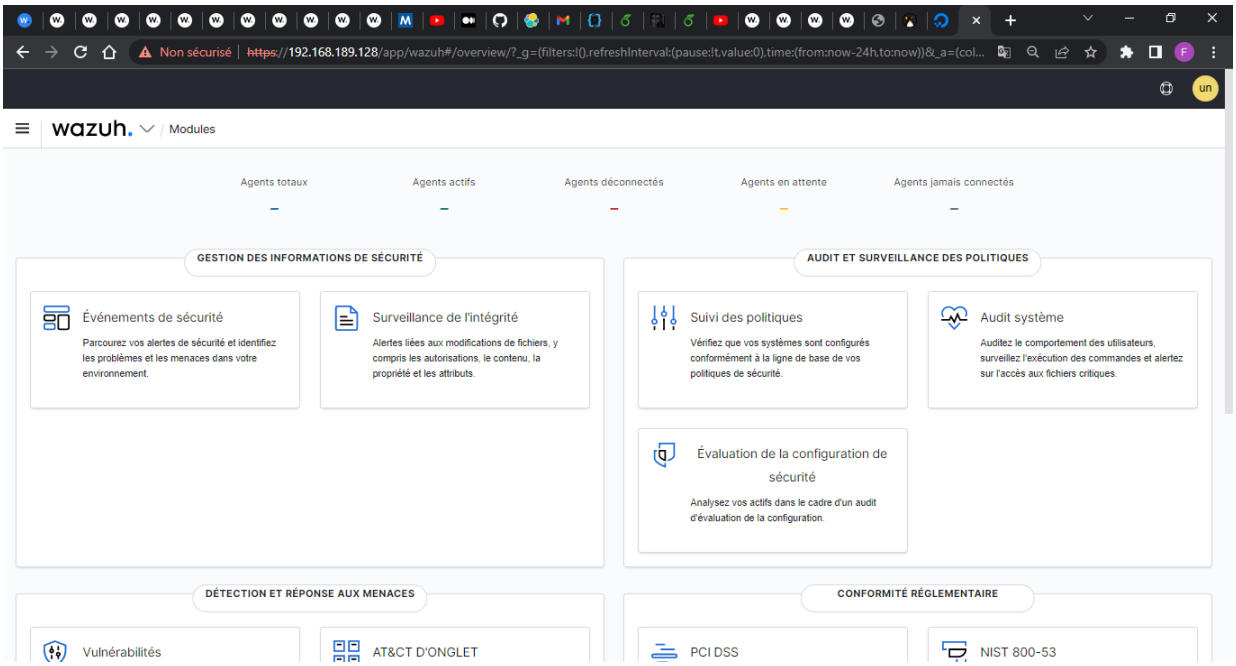



FIGURE 2.6 : Tableau de bord

### 4. Ajout des agents

- **Windows**
  Pour l'agent de windows, nous avons téléchargé le package wazuh-agent.msi et ensuite nous avons procédé à son installation. Echange de clé entre le client Windows et le Serveur

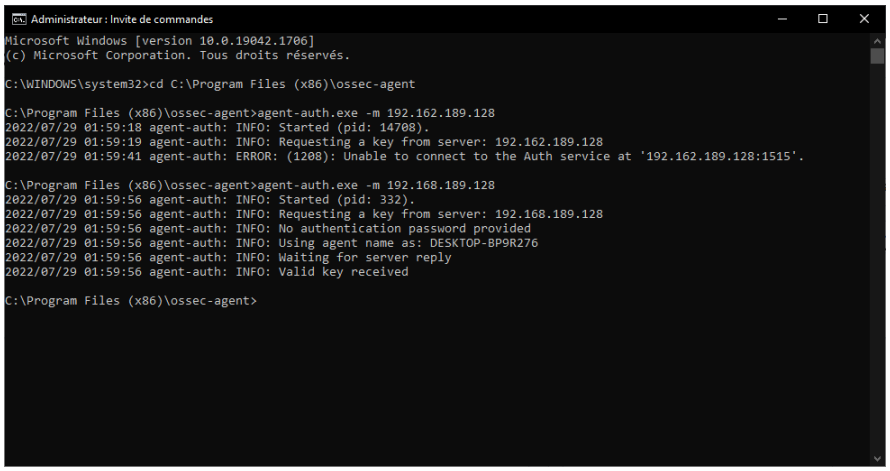



FIGURE 2.7 : Echange de clé entre le client Windows et le Serveur

- **Ubuntu**
  Pour l'agent linux, il faut ajouter le référentiel Wazuh pour télécharger les packages officiels
  - Installer la clé GPG avec
  **# curl -s https ://packages.wazuh.com/key/GPG-KEY-WAZUH | apt-key-add -**
  - Ensuite, ajoutez le dépôt avec
  **# echo "deb https ://packages.wazuh.com/4.x/apt/ stable main" | tee -a /etc/apt/sources.list.d/wazuh.list**

- Enfin, mettre à jour les informations sur le package avec

**# apt-get update**

- Déploiement de l'agent avec

**# WAZUH_MANAGER="<adresse_ip_manager>" apt-get install wazuh-agent**

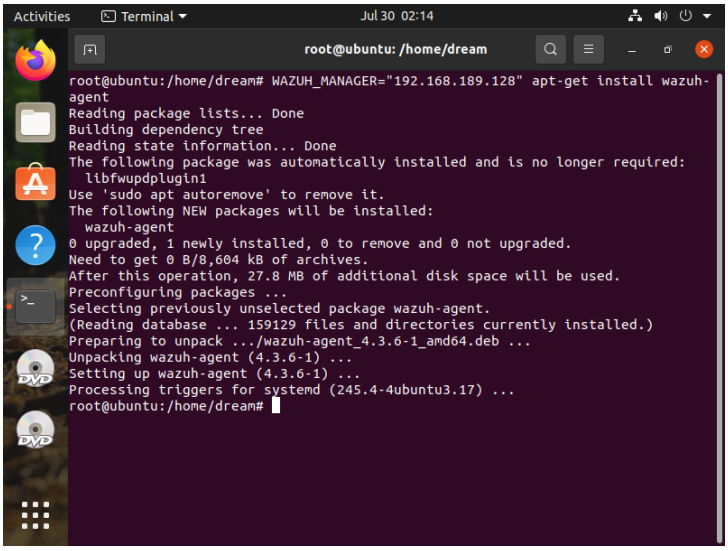



FIGURE 2.8 : Connexion avec le serveur

## 5. Liste des agents connectés au serveur

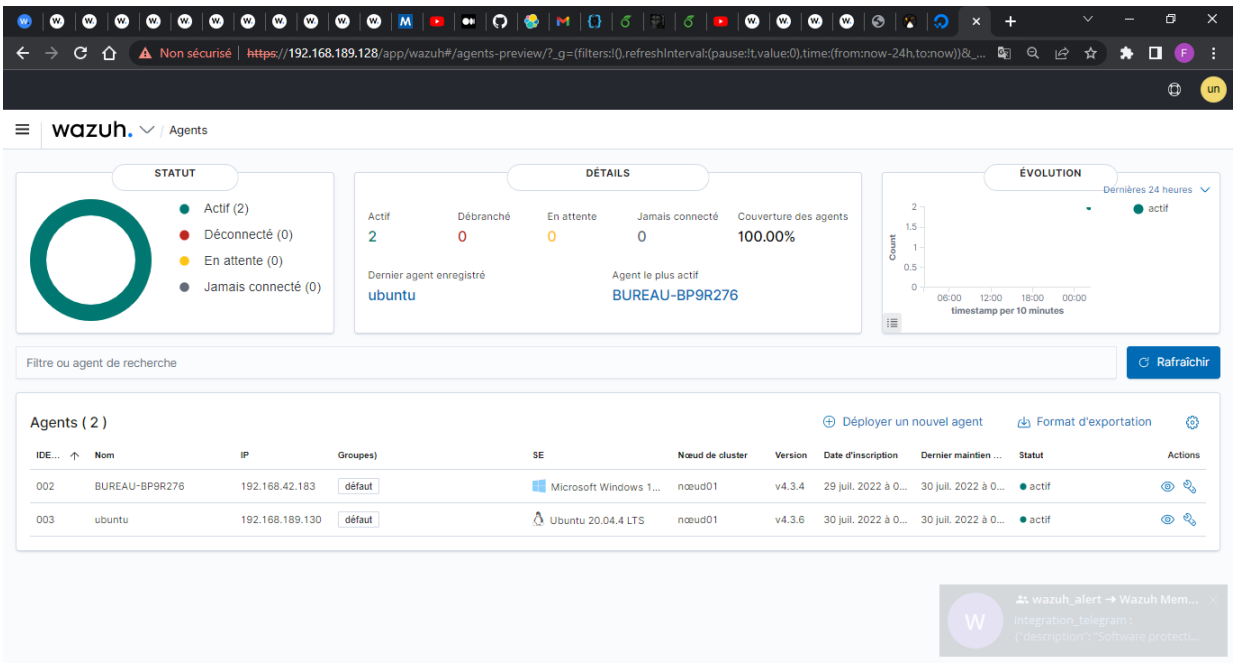



FIGURE 2.9 : Vue des agents sur le Tableau de bord

### 5. Intégration de Telegram

Telegram est une application de messagerie instantanée sécurisée hébergée dans un cloud. Pour commencer, il nous faut :

- Sur Telegram

Créer un compte Telegram sur lequel on procédera à la création d'un bot qui se chargera de la diffusion des alertes obtenues à partir du manager wazuh Pour cela, nous utiliserons BotFather, un outil Telegram qui permet la création de bot

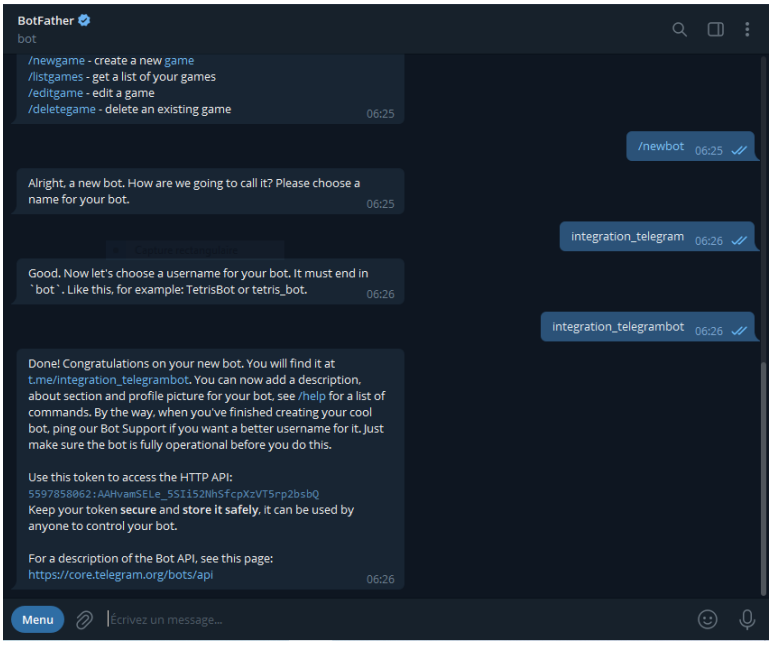



FIGURE 2.10 : Création du Bot

Création du bot nommé integration_telegram avec le nom d'utilisateur integration_telegrambot La création du bot nous donne un token qu'on utilisera pour notre intégration au niveau du serveur wazuh et aussi pour obtenir notre ID de chat. Nous allons démarrer notre bot créé et ensuite, nous allons créer un groupe Telegram nommé wazuh_alert dans lequel nous allons recevoir les alertes en temps réel. Dans le groupe, nous avons ajouté le bot Telegram comme membre du groupe pour qu'il s'occupe de la diffusion des alertes. Nous avons ensuite utilisé un site `https://sean-bradley.medium.com/get-telegram-chat-id-80b575520659` sur lequel nous mis le token obtenu après la création du bot pour obtenir l'ID du chat(groupe).

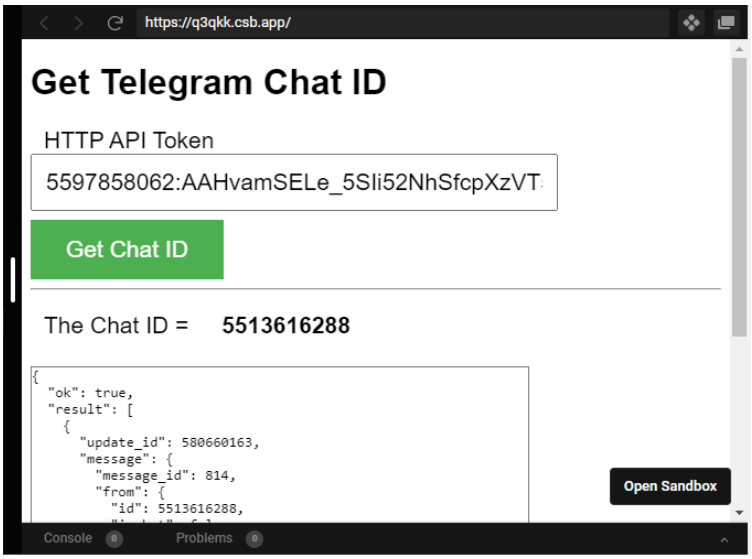



FIGURE 2.11 : Vue de la plateforme avec un ID obtenue

- Sur le manager de wazuh

Nous ferons l'intégration en utilisant deux scripts trouvés à cette adresse https://github.com/OpenSecureCo/Demos/blob/main/Telegram%20Integration. L'un écrit en Shell qui gère l'intégrations dans un nouveau fichier nommé **custom-telegram** qui sera à l'adresse **/var/ossec/integrations/**. L'autre écrit en python qui s'occupe de la gestion des messages envoyés vers le groupe Telegram nommé **custom-telegram.py** qui sera aussi à l'adresse **/var/ossec/integrations** Dans le fichier **custom-telegram.py** nous ajouterons l'ID du chat(groupe)

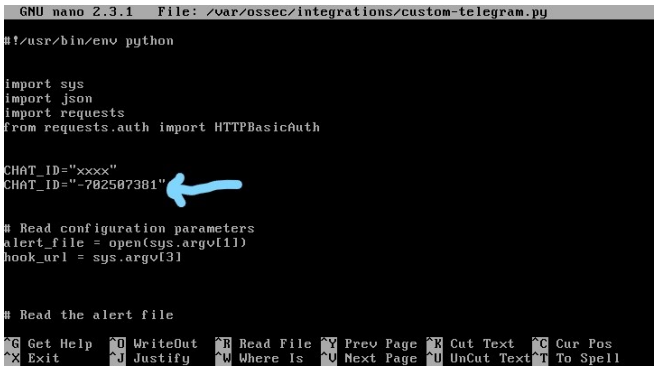



FIGURE 2.12 : Un aperçu du script python avec l'ID du chat insérer

A présent, il nous faut changer le groupe des deux fichiers avec **# chown root :wazuh /var/ossec/integrations/custom-telegram*** et modifier l'accès avec **# chmod 750 /var/ossec/integrations/custom-telegram***

Pour finir, il nous faut ajouter le code ci-dessous au fichier de configuration **ossec.conf** du manager qui se trouve au répertoire **/var/ossec/etc/** en remplaçant la partie ***YOUR API KEY*** par le token obtenu après la création du bot Telegram.

```
<integration>
      <name>custom-telegram</name>
      <level>3</level>
      <hook_url>https://api.telegram.org/bot5580417850:AAEBi2664aBTHRfdNZjgkDCMMNjH6AdunrI/sendMe$
      <alert_format>json</alert_format>
</integration>
```

FIGURE 2.13 : Code d'intégrations au fichier ossec.conf avec l'api key ajoutés

Après l'ajout du code, nous allons redémarrer le manager avec la commande **# systemctl restart wazuh-manager**. S'il n'y a pas eu d'erreur après le redémarrage, nous devrons commencer par recevoir les alertes dans notre groupe Telegram comme sur la figure ci-dessous

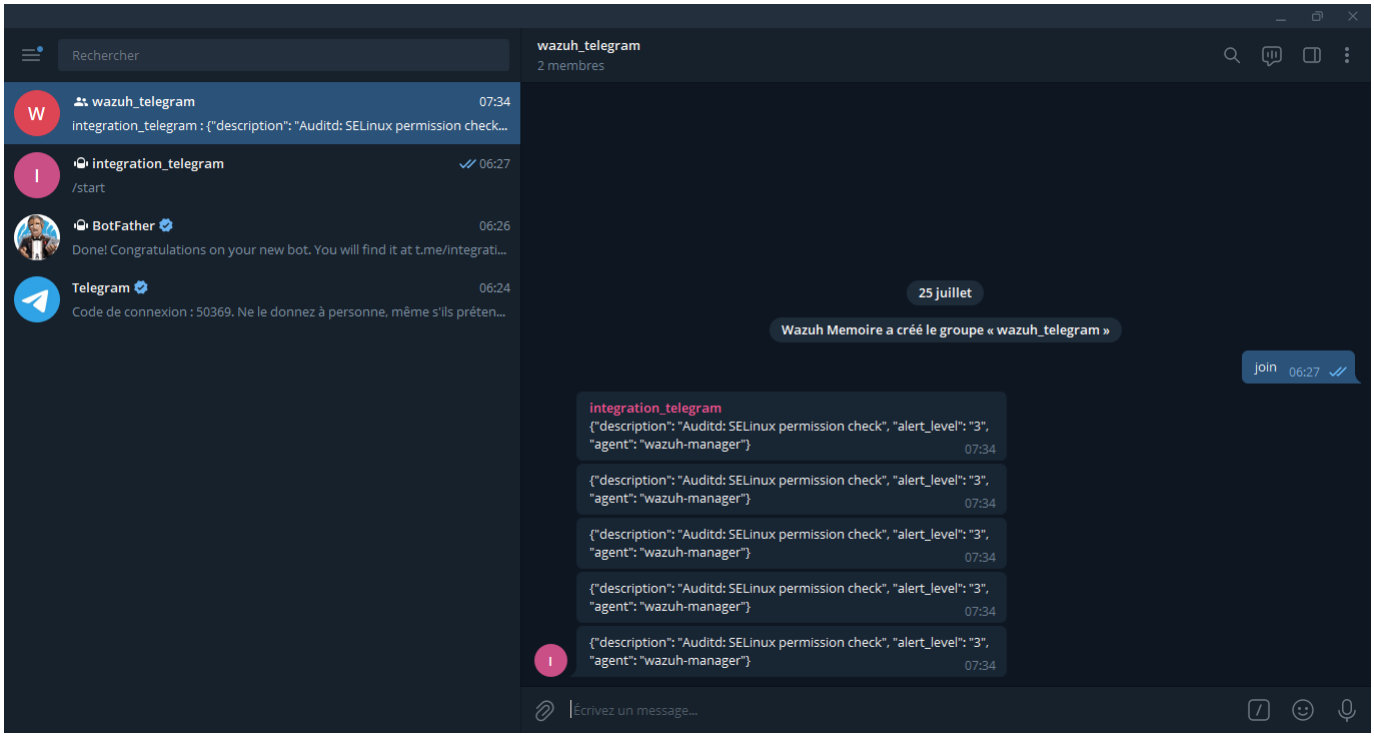



FIGURE 2.14 : Aperçu des premiers alertes reçu dans le groupe Telegram

## Conclusion

Wazuh est un outil puissant et facile à déployer qui sécurise notre réseau. Un seul serveur Wazuh peut analyser les données de centaines ou de milliers d'agents, évoluer lorsqu'il est configuré en mode cluster et même faciliter pour ses intégrations et configurations. Sa mise en place a été effectuée avec succès comme décrit ce chapitre.

Chapitre 3

# Tests, Résultats et Discussion

## Introduction

Dans ce chapitre, nous allons faire quelques cas pratiques concernant Wazuh et faire l'étude des résultats après test.

## 1 Tests et résultats

Nous allons tester quelques cas d'utilisation de Wazh et afficher les résultats.

### 1.1 Scénario 1 : Détecter une attaque par force brute SSH et réponse active

Il nous faut d'abord un accès distant par force brute SSH de la machine Kali à la machine Ubuntu avec l'outil hydra et exécution d'une réponse en réaction à l'attaque SSH. Pour commencer, nous allons modifier la configuration des règles Wazuh qui signalent si une tentative SSH est une attaque force brute. Les règles utilisées par Wazuh sont au répertoire **/var/ossec/ruleset/rules** et nous nous intéressons aux règles SSH qui se trouvent dans le fichier **0095-sshd_rules.xml**, plus précisément la règle avec **l'ID 5712** qui concerne les attaques SSH par force brute. Par défaut, la règle est configurée de sorte que 8 tentatives SSH échouées en moins de 120s soit 2mn puissent être considérés comme une attaque force brute SSH. Nous avons modifié la règle de sorte que 3 tentatives SSH échouées en moins de 120s puissent être considérés comme une attaque SSH par force brute comme montré à la figure ci-dessous

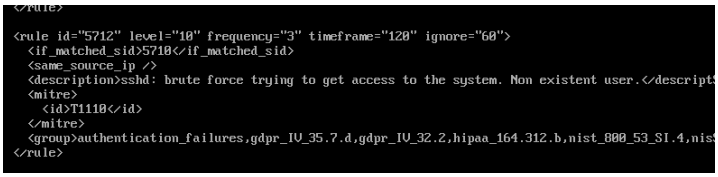



FIGURE 3.1 : Modification de la règle Wazuh

Maintenant, nous allons nous occuper de la partie réponse active. La réponse active est l'une des fonctionnalités intégrées à Wazuh. Ils exécutent un script en réponse au déclenchement d'alertes spé-

cifiques en fonction du niveau d'alerte ou du groupe de règles. N'importe quel nombre de scripts peut être lancé en réponse à un déclencheur et peut être écrit dans n'importe quel langage de programmation. Cependant, ces réponses doivent être considérées avec soin. Une mauvaise mise en œuvre des règles et des réponses peut accroître la vulnérabilité du système.
Pour notre exemple, nous avons écrit un petit script en langage bash qui sera exécuté lorsque la règle 5712 sera déclenché. Les scripts pour la réponse active doivent se trouver au répertoire **/var/ossec/active-response/bin**
. Notre script aura juste à créer deux fichiers, dans l'un(reponse-text), nous aurons un petit texte qui dira ''réponse déclenchée avec succès'' et l'autre (reponse-json) nous auront un aperçu du code json générée par alerte

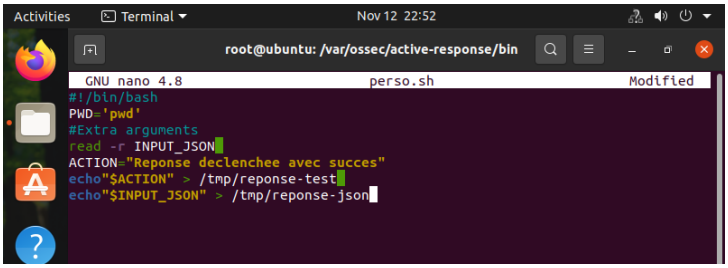



FIGURE 3.2 : Script bash nommée perso.sh

Ensuite, il nous faut une configuration au niveau du manager qui ordonne à la machine concernée l'action à effectuer quand telle règle est déclenchée à son niveau. La configuration est simple. Nous avons juste dit au manager de déclencher le script perso.sh qui se trouve au répertoire **/var/ossec/active-response/bin** de l'agent qui a déclenchée la règle avec **ID 5712**.

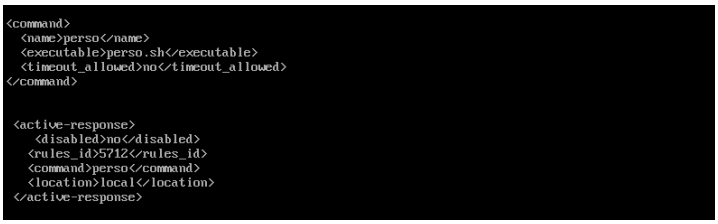



FIGURE 3.3 : Configuration du fichier ossec.conf du manager pour la gestion du AR

Après les modifications effectuées, il faut redémarrer le manager wazuh avec **# systemctl restart wauh-manager**

- **Test**

  Sur notre machine Kali qui sert d'attaquant, nous avons utilisé l'outil hydra pour lancer l'attaque SSH vers la machine Ubuntu.

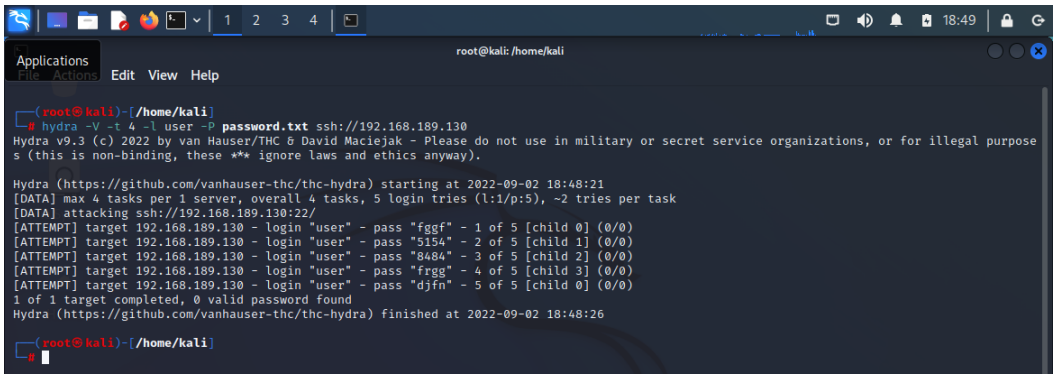



FIGURE 3.4 : Brute force SSH avec hydra sur kali

- **Resultat**
  Sur notre agent, nous pouvons voir les deux fichiers créés en réponse au déclenchement de la règle 5712

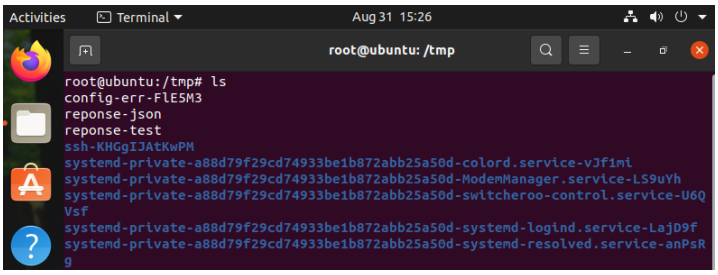



FIGURE 3.5 : Fichier créé grâce à l'AR

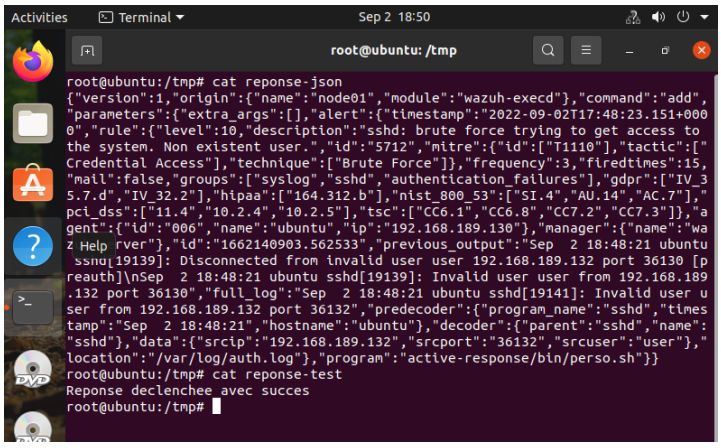



FIGURE 3.6 : Contenu des deux fichiers avec cat

Et pour finir, les alertes envoyées dans le groupe Télégram et nous pouvons voir qu'à la troisième tentative, il a été reconnu comme une brute force attaque comme indiqués dans la règle que nous avons modifiée plus tôt.

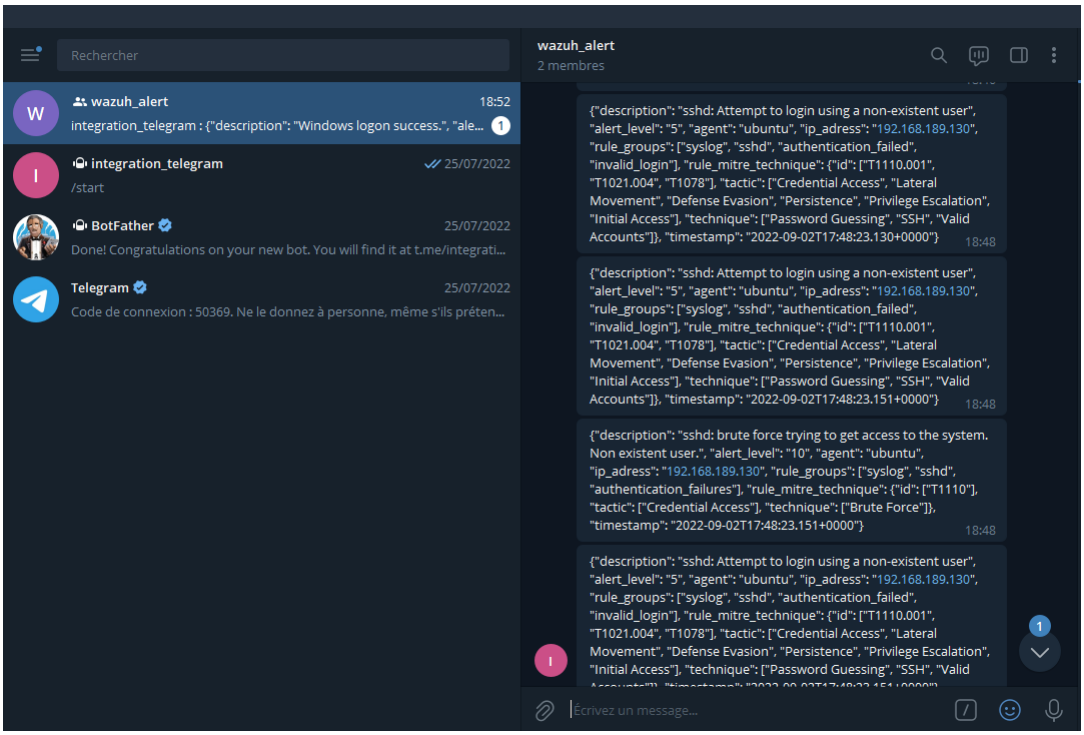



FIGURE 3.7 : Alertes Telegram après l'attaque

## 1.2 Scénario 2 :Détecter les modifications du système de fichiers

Wazuh possède des fonctionnalités de monitoring files FIM (File Integrity Monitoring) que nous allons mettre à l'épreuve. Pour notre test avons créé un fichier nommé « mémoire » afin d'assurer la surveillance de l'intégrité de ce dernier.
Le code suivant a été ajouté au fichier ossec.conf de l'agent afin d'assurer la surveillance.

```
<syscheck>
    <disabled>no</disabled>
    <scan_on_start>yes</scan_on_start>
    <directories check_all="yes" realtime="yes" report_changes="yes">C:/memoire</directories>
</syscheck>
```

FIGURE 3.8 : Code ajouté au fichier ossec.conf

Ici, nous demandons à l'outil de surveillance FIM de garder un œil sur le fichier "mémoire" en temps réel.

```
2022/08/18 16:24:58 wazuh-agent: INFO: (6002): Monitoring registry entry: 'HKEY_LOCAL_MACHINE\Software\Microsoft\Active Set
2022/08/18 16:24:58 wazuh-agent: INFO: (6003): Monitoring path: 'c:\memoire', with options 'size | permissions | owner | gr
2022/08/18 16:24:58 wazuh-agent: INFO: (6003): Monitoring path: 'c:\programdata\microsoft\windows\start menu\programs\start
```

FIGURE 3.9 : Fichier de log

Au niveau de l'image ci-dessus, nous pouvons voir le fichier log de l'agent qui montre à la deuxième ligne que le fichier est effectivement en monitoring

- **Test**
  Ajout de fichier au répertoire en monitoring

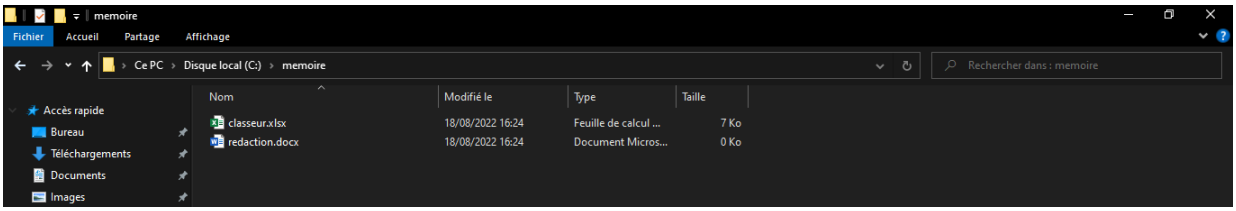



FIGURE 3.10 : Fichier redaction.docx et classeur.xlsx ajoutés

- **Test**
  Suppression du fichier

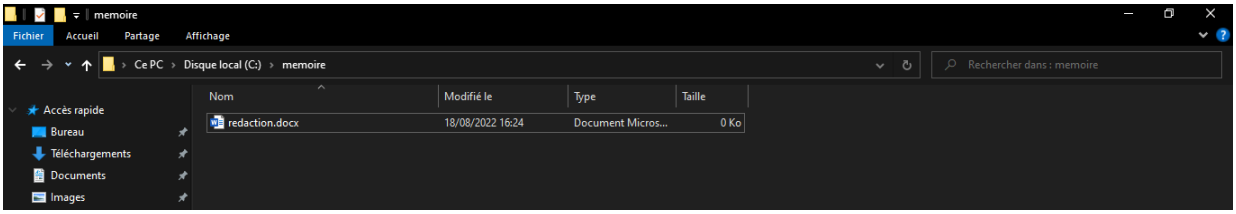



FIGURE 3.11 : Suppression du fichier

- **Test**
  Modification du fichier redaction.docx du répertoire en monitoring

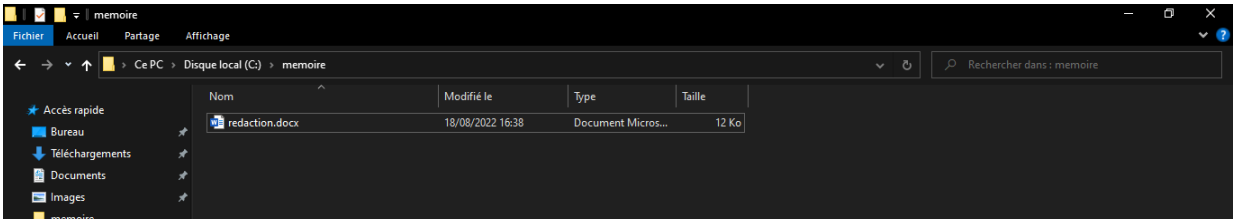



FIGURE 3.12 : Fichier après modification

- **Resultat**

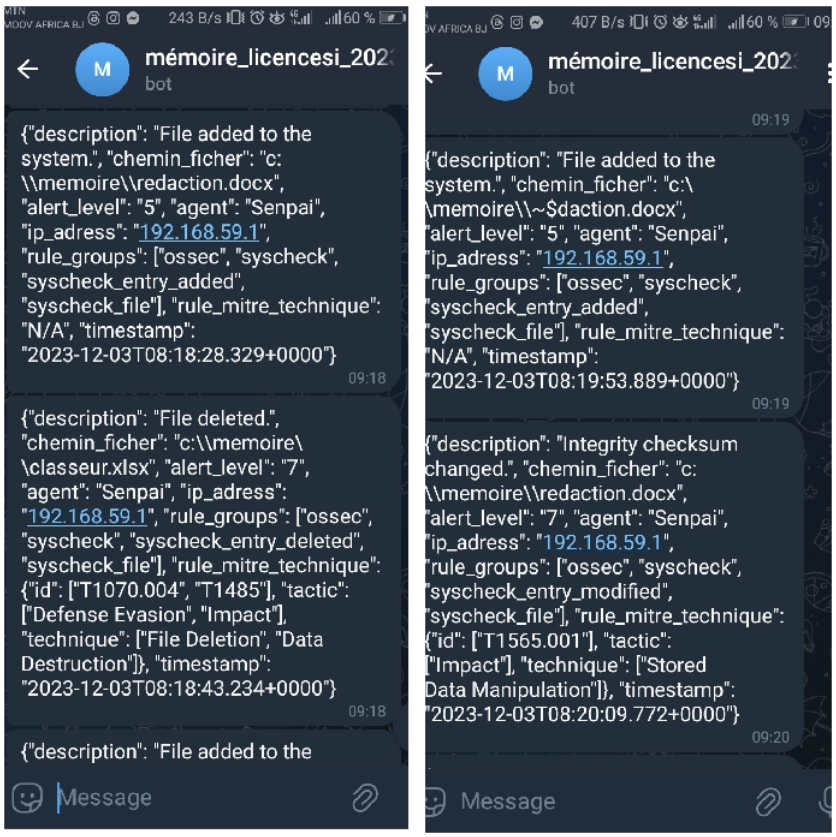



FIGURE 3.13 : Les alertes reçues sur Telegram

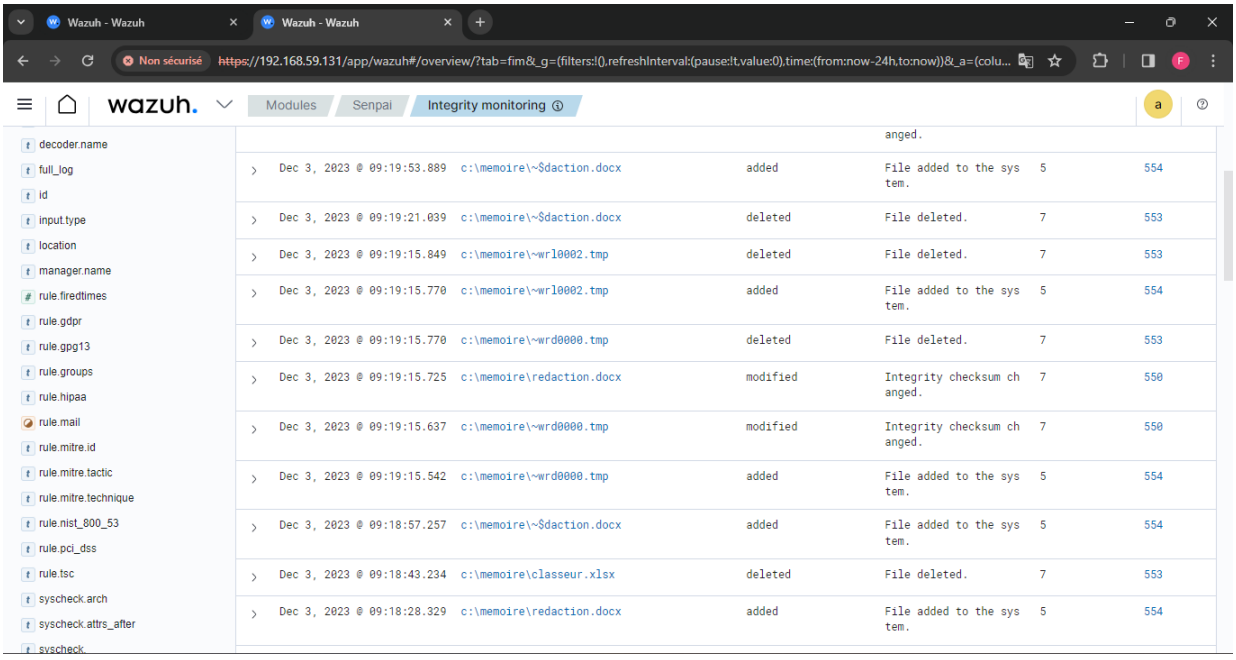



FIGURE 3.14 : Tableau de bord avec toutes les alertes recues

## 2 Discussion

La mise en place de notre solution HIDS open source WAZUH au MDGL répond aux préoccupations de départ qui sont celles de l'amélioration de la sécurité équipements dans le parc informatique. Les fonctionnalités de wazuh, au regard des tests que nous avons effectués, prouvent qu'il est capable de surveiller les connexions des utilisateurs aux agents, de réagir en de cas violation des règles qui lui sont configurées et aussi d'assurer la surveillance(l'intégrité) de n'importe quel fichier.

## Conclusion

A la vue de la réussite des tests effectués et les résultats obtenus, nous pouvons dire que Wazuh fonctionne et est efficace.

# Conclusion Générale

La sécurité des réseaux informatiques demeure encore et toujours un sujet très sensible, voire complexe pour les acteurs du monde informatique. Même si on note une évolution des technniques de défense, l'évolution des techniques d'attaques ne sont pas à négliger. Il est d'ailleurs difficile voire impossible de garantir une sécurité à cent pourcent. L'objectif de notre mémoire était de présenter un des solutions existantes et pouvant fournir aux administrateurs réseaux de surveillance en temps réel les équipements dans un réseau afin de garantir leur sécurité. Notre travail consistait à mettre en place Wazuh qui est une plateforme de sécurité open source ou aussi un outil de détection d'intrusion système basé sur l'hôte. Dans un premier lieu, nous avons pu étudier notre solution et nous avons présenté les avantages qui nous ont motivés à faire ce choix. Dans la partie réalisation, nous avons mis en place notre solution WAZUH et configuré les différents agents sur les différents hôtes pour leur surveillance ainsi que l'alerter par Telegram en cas d'atteinte aux règles qui ont été configurées. Conscient que notre travail pour la protection d'un réseau est perfectible, des perspectives sont donc envisagées en vue de l'améliorer. Ainsi nous envisageons :

- Renforcer la détection en modifiant les configurations et les règles de notre solution pour être conformes à la politique établie ;
- Automatiser la réponse aux menaces en exécutant des actions lorsqu'elles sont détectées ;
- Intégrer l'outil SURICATA qui se chargera de la détection des menaces en surveillant le trafic réseau ;
- La supervision d'autres types équipement réseau utiles en entreprise.

# Bibliographie


[4] Memoire de MARCELLE HOUNSA. “Mise en place d’un système de detection d’intrusion : cas de SNORT à SAHAM ASSURANCE Bénin, (IFRI-UAC)”. In : (2020).

[9] Projet Tutoré de GANKPE ESDRAS BÉNIS. “Analyse et comparaison des détecteurs d’intrusions (IDS) WAZUH et SURICATA, (Université de Perpignan Via Domitia PERPIGNAN au maroc)”. In : (2020).

[12] Memoire de SAMIOU WARAKPE. “Implémentation de SNORT/BASE pour la détection avec alerte par SMS : Cas de FECECAM-BENIN, (IFRI-UAC)”. In : (2019).

[13] Memoire de ELEGBA DURAND. “Mise en place d’un système de detection d’intrusion et simulation d’attaque : cas de Bro, (IFRI-UAC)”. In : (2019).

## Webographie


[1] *Parc informatique*. https://codeo.com/gestion-parc-informatique-tout-savoir/#:~:text=Qu%27est-ce%20que%20le,travail%2C%20et%20de%20ressources%20logicielles, consulté en Mars 2022.

[2] *Architecture Nids et Hybride*. https://codeo.com/gestion-parc-informatique-tout-savoir/#:~:text=Qu%27est-ce%20que%20le,travail%2C%20et%20de%20ressources%20logicielles, consulté en Mars 2022.

[3] *Architecture hids*. https://www.google.com/search?q=hids&client=ms-android-huawei&sca_esv=587313471&tbm=isch&prmd=isvn&sxsrf=AM9HkKkADwJ8cE7JTllHms-cnmMChTHLRA:1701536143600&source=lnms&sa=X&ved=2ahUKEwi-pqWLnPGCAxUIVEEAHWpWBVAUoAXoECAIQAQ&biw=360&bih=560&dpr=2#imgrc=VxIaFnljELkl3M, consulté en Mars 2022.

[5] *Top 5 Open-source HIDS systems*. https://logz.io/blog/open-source-hids/, consulté en Mars 2022.

[6] *Comparaison entre HIDS*. https://en.m.wikipedia.org/wiki/Host-based_intrusion_detection_system_comparison, consulté en Mars 2022.

[7] *Critère de choix d'un IDS*. https://www.securiteinfo.com/conseils/choix_ids.shtml#:~:text=R%C3%A9activit%C3%A9%20%3A%20Un%20IDS%20doit%20%C3%AAtre,sont%20pour%20ainsi%20dire%20indispensables, consulté en Mars 2022.

[8] *Autres critères de choi*. https://fre.myservername.com/top-10-best-intrusion-detection-systems, consulté en Mars 2022.

[10] *Documentation de wazuh*. https://documentation.wazuh.com/current/index.html, consulté en Mars 2022.

[11] *Flusx de données*. https://documentation.wazuh.com/current/getting-started/components/index.html, consulté en Mars 2022.

[14] *Systeme de détection d'intrusion*. https://fr.m.wikipedia.org/wiki/Syst%C3%A8me_de_d%C3%A9tection_d%27intrusion, consulté en Mars 2022.

[15] *Site de Wazuh*. https://wazuh.com, consulté en Mars 2022.

[16] *Code source de WAZUH*. https://github.com/wazuh/wazuh/tree/4.3, consulté en Mars 2022.

# Table des matières

**Dédicace** **2**

**Remerciements** **3**

**Résumé** **4**

. . . . . 4

**Abstract** **5**

. . . . . 5

**Table des figures** **6**

**Liste des tableaux** **7**

**Liste des acronymes** **8**

**Glossaire** **9**

**Introduction** **2**

1 Contexte et justification . . . . . 2
2 Problématique . . . . . 2
3 Objectifs . . . . . 2
4 Organisation du mémoire . . . . . 3

**1 Revue de littérature** **4**

Introduction . . . . . 4
1 Généralités . . . . . 4
1.1 Défintion . . . . . 4
1.2 Les objectifs de la sécurité informatique . . . . . 4
1.3 Système de détection d'intrusion . . . . . 5
1.3.1 Définition . . . . . 5
1.3.2 Fonctionnement des IDS . . . . . 5
1.3.3 Différents type d'IDS . . . . . 6

**2 Choix de la solution et Implémentation** **10**

Introduction . . . . . 10
1 Choix de la Solution Implémentée . . . . . 10
1.1 Présentation de quelques HIDS Open source les plus connus [5]. . . . . 10
1.2 Critère de choix d'un IDS [7] . . . . . 11
1.3 Présentation générale de Wazuh [10] . . . . . 13

1.3.1 Composants . . . 13
1.3.2 Architecture et fonctionnement de Wazuh . . . 16
1.4 Mise en place de la solution . . . 16
1.4.1 Architecture . . . 16
1.4.2 Outils utilisés . . . 19
1.4.3 Installation de l'environnement et configuration . . . 19

**3 Tests, Résultats et Discussion** **25**
Introduction . . . 25
1 Tests et résultats . . . 25
1.1 Scénario 1 : Détecter une attaque par force brute SSH et réponse active . . . 25
1.2 Scénario 2 :Détecter les modifications du système de fichiers . . . 28
2 Discussion . . . 31

**Conclusion** **32**

**Bibliographie** **33**

**Webographie** **34**

**Table des matières** **35**