



Ministère de l'enseignement supérieur
et de la recherche scientifique

Université de Carthage



Institut Supérieur des Sciences Appliquées
et de Technologie de Mateur

Mémoire de Mastère Professionnel

Présenté en vue de l'obtention du diplôme national de Mastère
professionnel en

Réseaux et Télécoms

Parcours : Services et Sécurité des Réseaux

2.1 Déploiement d'une plateforme XDR intelligent

Elaboré par :

Taam Khaoula



Soutenu le 18/06/2026 devant le jury composé de :

Président : **Dr. Feltekh Kais**
Rapporteur : **Dr. Najjar Abdelhalim**
Encadrant : **Dr. Ben Said Yosr**
Encadrant entreprise : **M. Jlailia Imed**

Année universitaire : 2025/2026

Table des matières

Chapitre 1 : contexte de projet	4
1.1 Introduction	4
1.2 Cadre du projet	4
1.2.1 Présentation de l'organisme d'accueil	5
1.2.2 Organigramme de SOTETEL	6
1.3 Contexte technique de l'environnement d'étude	7
1.3.1 Etude de l'existant	8
1.3.2 Critique de l'existant	9
1.3.3 Solution proposée	9
1.4 Conclusion	10
Chapitre 2 : les principes fondamentaux	11
2.1 Introduction	11
2.2 Attaques DDoS	11
2.2.1 Définition d'une attaque DDoS	11
2.2.2 Les attaques Zero-Day	12
2.2.3 Impacts et risques des attaques DDoS	12
2.2.4 Défis et Complexités des Attaques DDoS	13
2.3 Les systèmes de détection traditionnels	14
2.3.1 Systèmes de détection basés sur les signatures	14
2.3.2 Définition des IDS	15
2.3.3 Définition des SIEM	15
2.3.4 Définition des EDR	16
2.4 Limites des solutions traditionnelles	16
2.4.1 Limites des méthodes de détection basées sur les signatures	16
2.4.2 Limites des IDS	17
2.4.3 Limites des SIEM	17
2.4.4 Limites des EDR	18
2.5 Problème de fragmentation des outils	18
2.6 L'intelligence artificielle (IA)	19
2.6.1 Intelligence artificielle pour la détection DDoS	19
2.6.1.1 Définition de l'intelligence artificielle	19

2.6.1.2	L'apprentissage automatique	19
2.6.2	Détection des anomalies basée sur l'IA	20
2.6.3	Intelligence artificielle dans la cybersécurité	22
2.7	XDR et corrélation des événements	22
2.7.1	Définition du XDR	23
2.7.2	Architecture générale du XDR	23
2.7.3	Centralisation des alertes	24
2.7.4	Corrélation des événements	24
2.7.5	Réponse automatisée	25
2.7.6	Comparaison des solutions de sécurité : IDS, SIEM, EDR et XDR	25
2.8	Conclusion	27
Chapitre 3 : Implémentation de la plateforme XDR intelligente		29
3.1	Introduction	29
3.2	Analyse des besoins	29
3.2.1	Besoins fonctionnels	29
3.2.2	Besoins non fonctionnels	30
3.3	Architecture globale du système	31
3.4	Environnement de travail	32
3.4.1	Matériel utilisé	32
3.4.2	Environnement logiciel	33
3.4.3	Données utilisées pour l'expérimentation	33
3.4.3.1	Types d'attaques DDoS répertoriés dans les données	34
3.4.3.2	La préparation des données	36
3.4.3.3	Réduction des données	36
3.4.3.4	Division des données	38
3.4.3.5	Bibliothèques et technologies utilisées	39
3.4.3.6	Features sélectionnées	40
3.4.3.7	Interprétation des features	41
3.4.4	Type d'apprentissage utilisé	41
3.5	Développement du modèle de détection DDoS	42
3.5.1	• Extraction et préparation des données	43
3.5.2	Analyse des données	43
3.5.3	• Normalisation des données	43

3.5.4	Évaluation du modèle	45
3.6	Mise en place de l'environnement XDR	47
3.6.1	Configuration de l'environnement Ubuntu/WSL	47
3.6.2	Déploiement et configuration de la plateforme Wazuh sous Docker	48
3.7	Conclusion	51
	 Chapitre 4. Mise en œuvre, tests et résultats	53
4.1	Introduction	53
4.2	Scénarios de tests	53
4.3	Évaluation des performances	54
4.4	Mécanisme de défense	57
4.5	Dashboard et visualisation	58
4.5.1	Règles iptables	59
4.5.2	Journal des alertes	60
4.5.3	Statistique	60
4.6	Conclusion	61
	Conclusion générale	613
	Perspective	64
	Webographie	65

Liste des figures

Figure 1:Logo de SOTETEL	6
Figure 2 :Organigramme de SOTETEL	7
Figure 4 : Schéma de fonctionnement du système	31
Figure 5 : Taxonomie des attaques DDoS [17]	35
Figure 6 : Extrait du code de prétraitement des données pour le XDR intelligent	37
Figure 7 : Code d'entraînement d'un modèle de détection binaire sous Keras	44
Figure 8 : Performances du modèle de classification binaire	46
Figure 9 : configuration de l'intégration de Docker Desktop	48
Figure 10 : Conteneurs Docker Wazuh opérationnels	49
Figure 11 : chargement de modèle	50
Figure 12 : Exécution du script test_model_csv.py	51
Figure 13 : Attaque DDoS	54
Figure 14 : Détection des attaques en temps réel	54
Figure 15 : Journalisation des alertes de sécurité	56
Figure 16 : Détection d'attaque et réponse automatique	57
Figure 17 : Tableau de bord " XDR SOC DASHBOARD "	58
Figure 18 : Règles iptables et journal des alertes	59
Figure 19 : Statistiques du dashboard XDR	60

Liste des tableaux

Tableau 1:détection d'attaque par signature vs. Machine learning	21
Tableau 2: Étude comparative des mécanismes de détection et de réponse aux incidents de sécurité	25
Tableau 3: Configuration de la machine	32
Tableau 4:: Résultats détectés le 03/11/2019	35
Tableau 5: Résultats détectés le 01/12/2019	36
Tableau 6: bibliothèques utilisées	39
Tableau 8:: catégories features	41

Dédicace et Remerciements

Je dédie ce travail avec tout mon amour, ma gratitude et ma profonde reconnaissance :

À mon cher père **Mohamed**, qui a toujours été un pilier de force et de sagesse, et qui a cru en moi même dans les moments de doute,

À ma chère mère **Jamila**, dont l'amour inconditionnel, les sacrifices et les prières ont constitué ma plus grande source de courage et de réussite,

À mes sœurs **Intyssar** et **Marwa**, pour leur soutien constant, leur présence réconfortante et leur affection qui m'ont toujours encouragée à avancer,

À mon ami **Med Rayen**, pour sa fidélité, son aide précieuse et son soutien sincère tout au long de ce parcours.

J'exprime également ma sincère reconnaissance :

À mon encadrante académique **Mme Yosr Ben Saïd**, pour son encadrement bienveillant, sa disponibilité et la pertinence de ses conseils qui ont enrichi ce travail,

À mon encadrant au sein de la société **SOTETEL**, **M. Imed Jlailia**, pour son accompagnement, sa confiance et son soutien précieux durant toute la période de stage.

Enfin, je tiens à me remercier moi-même pour ma persévérance, ma patience et mes efforts continus, qui m'ont permis de mener ce travail à terme.

Introduction générale

Dans le contexte actuel de la transformation numérique, les organisations reposent de plus en plus sur des systèmes d'information interconnectés et exposés à Internet. Cette évolution, bien qu'elle ait considérablement amélioré la performance, la flexibilité et la connectivité des infrastructures, s'accompagne d'une augmentation significative des menaces en cybersécurité. En effet, les cyberattaques deviennent de plus en plus sophistiquées et diversifiées, incluant notamment les attaques par déni de service distribué (DDoS), les intrusions, les logiciels malveillants ainsi que les violations de données.

Traditionnellement, la sécurité des systèmes d'information repose sur des solutions telles que les antivirus, les pare-feux et les systèmes de détection d'intrusion.

Toutefois, ces mécanismes fonctionnent souvent de manière isolée, ce qui limite leur efficacité face aux attaques complexes, coordonnées et multi-vectorielles. Par conséquent, ces approches ne permettent pas toujours une corrélation globale des événements de sécurité ni une réponse rapide et adaptée aux incidents.

Dans ce contexte, l'émergence du concept de XDR (Extended Detection and Response) constitue une évolution majeure dans le domaine de la cybersécurité. Le XDR vise à unifier et centraliser les données issues de différentes sources (terminaux, réseaux, serveurs et applications), afin d'offrir une visibilité étendue et corrélée sur l'ensemble du système d'information. Cette approche permet d'améliorer la détection des menaces, d'accélérer l'investigation des incidents et de renforcer les capacités de réponse.

Par ailleurs, l'intégration de l'intelligence artificielle et du machine Learning dans les systèmes de cybersécurité représente une avancée déterminante. Ces technologies permettent d'analyser de grands volumes de données en temps réel, de détecter des comportements anormaux et d'anticiper les attaques potentielles. Elles contribuent ainsi à renforcer l'efficacité des solutions XDR en améliorant la précision de la détection et en automatisant les mécanismes de réponse.

C'est dans ce cadre que s'inscrit notre projet de fin d'études, réalisé au sein de SOTETEL, intitulé :

« Conception et implémentation d'un système XDR intelligent basé sur l'intelligence artificielle pour la détection et la réponse aux cyberattaques ».

L'objectif principal de ce projet est de concevoir et de développer une plateforme intelligente capable de centraliser, corréler et analyser les données de sécurité issues de sources hétérogènes, afin de détecter automatiquement les menaces et de déclencher des mécanismes de réponse appropriés. Cette solution vise à renforcer la sécurité des infrastructures informatiques en adoptant une approche proactive, intelligente et automatisée face aux cyberattaques.

Une attention particulière est accordée à la détection et à la mitigation des attaques DDoS, considérées comme l'une des menaces les plus critiques, en raison de leur capacité à saturer les ressources réseau et à perturber les services.

Ce rapport est structuré comme suit :

Le premier chapitre présente l'organisme d'accueil, SOTETEL, ainsi que l'étude de l'existant et la problématique du projet.

Le deuxième chapitre introduit les concepts fondamentaux liés à la cybersécurité, au XDR et à l'intelligence artificielle.

Le troisième chapitre décrit l'architecture de la solution proposée ainsi que les outils et technologies utilisés.

Enfin, le quatrième chapitre est consacré à l'implémentation, aux tests réalisés et à l'analyse des résultats obtenus.

Chapitre 1 : contexte de projet

3.1 Introduction

Dans le contexte actuel de la transformation numérique, les systèmes d'information des entreprises sont de plus en plus exposés à des menaces cybernétiques sophistiquées, persistantes et en constante évolution. La multiplication des attaques ciblant les infrastructures réseau, notamment les attaques par déni de service distribué (DDoS), les intrusions avancées et les logiciels malveillants, rend indispensable la mise en place de mécanismes de détection et de réponse à la fois efficaces et réactifs. Ce chapitre a pour objectif de présenter le contexte général du projet ainsi que l'environnement dans lequel il s'inscrit. Il met en évidence les principaux enjeux liés à la cybersécurité au sein des infrastructures modernes, caractérisées par une forte interconnexion et une grande complexité. Par la suite, une analyse de l'existant est réalisée afin d'identifier les limites des solutions de sécurité actuelles, souvent hétérogènes et insuffisamment corrélées.

Cette étude critique permet de justifier le choix d'une approche basée sur un système XDR intelligent, capable d'améliorer la visibilité globale, d'optimiser la détection des menaces et de renforcer la capacité de réponse face aux incidents de sécurité.

3.2 Cadre du projet

Ce projet de fin d'études s'inscrit dans le cadre de l'obtention du diplôme de Master professionnel en Services de Sécurité Réseau à l'Institut Supérieur des Sciences Appliquées et de Technologie de Mateur (ISSATM).

Il a été réalisé au sein de la société SOTETEL sur une période de trois mois. Ce stage constitue une opportunité d'application et de mise en pratique des connaissances théoriques acquises au cours de la formation, notamment dans les domaines de la cybersécurité, des réseaux informatiques et des systèmes de détection d'intrusions. L'objectif principal de ce projet est de concevoir et de mettre en œuvre une solution de sécurité basée sur une architecture XDR (Extended Detection and Response), capable d'améliorer la détection des menaces ainsi que la réponse aux incidents dans un environnement réseau complexe et en constante évolution.

3.2.1 Présentation de l'organisme d'accueil

La SOTETEL (Société Tunisienne d'Entreprise de Télécommunications) est une entreprise fondée en 1981, reconnue pour son expertise et son approche innovante dans le domaine de la maintenance et du déploiement des réseaux de télécommunications, tant au niveau national qu'international. Elle intervient aussi bien dans les infrastructures de réseaux publics que privés, contribuant ainsi au développement et à la modernisation du secteur des télécommunications en Tunisie et à l'étranger.

En tant que filiale de Tunisie Télécom, la SOTETEL occupe une position stratégique dans le paysage des entreprises tunisiennes du secteur. Elle est spécialisée dans l'installation, la maintenance et l'optimisation des infrastructures de télécommunications.

Son capital est réparti entre plusieurs actionnaires, dont Tunisie Télécom (35 %), El Athir Funds (7,47 %), ainsi que d'autres actionnaires divers représentant 57,33 %.



Figure 1:Logo de SOTETEL

3.2.2 Organigramme de SOTETEL

La SOTETEL est structurée en cinq directions centrales, chacune ayant des missions spécifiques contribuant au bon fonctionnement de l'entreprise :

- D.C.F (Direction Centrale Financière) : elle est chargée de la gestion financière, de la comptabilité ainsi que des tâches administratives liées à la gestion des ressources de l'entreprise.
- D.C.C (Direction Centrale Commerciale) : elle assure la gestion des activités commerciales, notamment le développement des ventes, le suivi du chiffre d'affaires ainsi que les actions de marketing.
- D.C.R.H (Direction Centrale des Ressources Humaines) : elle prend en charge la gestion du personnel, incluant le recrutement, l'intégration, la formation, la gestion administrative, la paie ainsi que la communication interne.
- D.C.S.E (Direction Centrale Solutions d'Entreprise) : elle est dédiée à l'étude, à l'installation et à la maintenance des réseaux privés destinés aux clients entreprises.

- D.C.RX (Direction Centrale des Réseaux) : elle est responsable de la mise en œuvre et de la maintenance des infrastructures de réseaux, incluant les réseaux de transmission et les réseaux d'accès.

La structure organisationnelle de la SOTETEL est représentée par l'organigramme ci-dessous, illustrant les relations hiérarchiques et fonctionnelles entre les différentes directions.

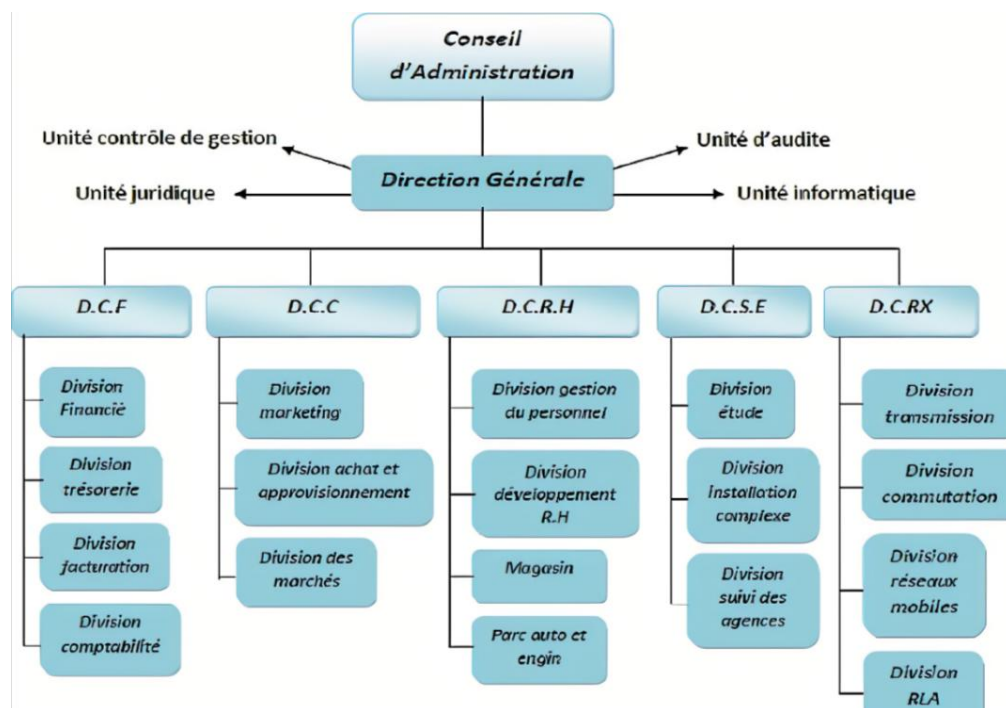


Figure 2 :Organigramme de SOTETEL

3.3 Contexte technique de l'environnement d'étude

L'environnement informatique de l'entreprise repose sur une infrastructure réseau distribuée, permettant l'accès à un ensemble d'applications métiers essentielles. Cette architecture joue un rôle fondamental dans la continuité des services ainsi que dans l'échange et la circulation des données entre les différents systèmes internes.

Cependant, cette interconnexion croissante entraîne une augmentation significative de la surface d'attaque, exposant ainsi le système d'information à divers risques de sécurité, tels que les intrusions non autorisées, les attaques par déni de service distribué (DDoS), l'interception de données sensibles ou encore la compromission des accès.

Dans ce contexte, la sécurité du système d'information devient un enjeu critique, nécessitant la mise en place de mécanismes de surveillance continue ainsi que de capacités avancées de détection et de réponse face aux incidents de sécurité.

3.3.1 Etude de l'existant

L'infrastructure actuelle repose sur des mécanismes de sécurité traditionnels qui assurent la surveillance du réseau et la protection des données. Toutefois, face à l'évolution rapide et à la sophistication croissante des cybermenaces, ces solutions montrent plusieurs limites.

En effet, la surcharge des équipes d'administration réseau, la difficulté de mise en place d'une surveillance centralisée et continue, ainsi que l'exploitation insuffisante des journaux de sécurité (logs) réduisent significativement l'efficacité de la supervision et de l'analyse des incidents.

Par ailleurs, la détection tardive des attaques, la vulnérabilité aux attaques distribuées telles que les DDoS, ainsi que le manque de corrélation entre les événements de sécurité rendent l'identification des menaces complexes plus difficile.

Ces insuffisances impactent directement la capacité de réaction du système de sécurité et soulignent la nécessité d'adopter des solutions plus avancées et intelligentes.

3.3.2 Critique de l'existant

L'évolution rapide des réseaux informatiques a augmenté la complexité des infrastructures de la SOTETEL, entraînant une surcharge des administrateurs réseau et des limites dans la supervision globale. Cette situation accroît la vulnérabilité du système d'information face à diverses menaces telles que les attaques DDoS, les intrusions et les attaques de type Man-in-the-Middle.

Par ailleurs, la difficulté de surveillance continue du trafic, l'exploitation insuffisante des logs, ainsi que le manque de corrélation des événements de sécurité réduisent l'efficacité de la détection des incidents. De plus, l'absence de mécanismes de protection avancés et la détection tardive des attaques compromettent la réactivité du système de sécurité.

Ainsi, ces limitations mettent en évidence la nécessité de renforcer le dispositif de sécurité de la SOTETEL à travers des solutions plus avancées de supervision, de corrélation et de réponse aux menaces.

3.3.3 Solution proposée

Afin de répondre aux limites identifiées, nous proposons la mise en place d'une plateforme de sécurité basée sur une architecture XDR (Extended Detection and Response). Cette solution vise à centraliser et corréler les données issues de sources hétérogènes telles que le réseau, les systèmes, les journaux de sécurité et les mécanismes d'authentification, afin de fournir une analyse globale et cohérente des événements de sécurité.

L'intégration de techniques d'analyse avancées et d'intelligence artificielle permet d'assurer une détection proactive des menaces, une réduction significative du temps de réponse aux incidents, ainsi qu'une meilleure visibilité sur les attaques complexes et multi-vectorelles. Elle favorise également l'automatisation des actions de réponse, améliorant ainsi l'efficacité globale du dispositif de sécurité.

Ainsi, la solution proposée offre une approche unifiée, intelligente et évolutive de la cybersécurité, parfaitement adaptée aux environnements informatiques modernes.

3.4 Conclusion

Ce chapitre a permis de présenter le contexte général du projet ainsi que l'analyse de l'environnement d'étude. L'étude de l'existant a mis en évidence plusieurs limites des solutions de sécurité traditionnelles, notamment en matière de visibilité, de corrélation des événements et de réactivité face aux incidents de sécurité.

Ces insuffisances soulignent la nécessité d'adopter une approche plus évoluée et intelligente basée sur un système XDR (Extended Detection and Response). Une telle solution permettrait d'améliorer significativement la détection des menaces, la corrélation des événements et la rapidité de réponse au sein des infrastructures modernes.

Chapitre 2 : les principes fondamentaux

4.1 Introduction

L'évolution des cybermenaces, notamment les attaques DDoS, impose le développement de solutions de sécurité plus intelligentes et plus efficaces. Les approches traditionnelles comme les IDS, les SIEM et les EDR présentent aujourd'hui certaines limites face aux attaques modernes. Dans ce contexte, l'intelligence artificielle et les architectures XDR apportent une approche plus avancée basée sur la détection intelligente, la corrélation des événements et l'automatisation de la réponse aux incidents.

Ce chapitre présente les notions essentielles liées aux attaques DDoS, aux systèmes de détection traditionnels, à l'intelligence artificielle appliquée à la cybersécurité ainsi qu'aux principes du XDR, qui constituent la base de notre solution proposée.

4.2 Attaques DDoS

4.2.1 Définition d'une attaque DDoS

L'attaque par déni de service distribué (DDoS) est une forme de cyberattaque dans laquelle un grand nombre de systèmes compromis, généralement infectés par des logiciels malveillants et contrôlés à distance, sont utilisés pour saturer volontairement les ressources d'une cible, telle qu'un serveur web, un service en ligne ou une infrastructure réseau.

L'objectif principal de ce type d'attaque est de perturber le fonctionnement normal des services en générant un volume massif de trafic malveillant. Cette saturation peut

entraîner des ralentissements importants, des interruptions de service, voire une indisponibilité totale des services pour les utilisateurs légitimes.[19]

4.2.2 Les attaques Zero-Day

Les attaques dites zero-day exploitent des vulnérabilités encore inconnues des éditeurs de logiciels et pour lesquelles aucun correctif n'est disponible au moment de l'attaque.

En raison de leur caractère inédit, ces menaces sont particulièrement difficiles à détecter et à anticiper par les solutions de sécurité traditionnelles.

Elles représentent ainsi un risque critique, car elles peuvent compromettre un système avant même que les vulnérabilités ne soient identifiées et corrigées, laissant les infrastructures exposées à des attaques potentiellement graves.[2]

4.2.3 Impacts et risques des attaques DDoS

Les attaques par déni de service distribué (DDoS) ont des conséquences significatives sur les systèmes d'information ainsi que sur les activités des organisations. Leur principal impact réside dans la perturbation ou l'interruption des services en ligne, rendant les applications, sites web ou infrastructures réseau inaccessibles aux utilisateurs légitimes.

Sur le plan opérationnel, ces attaques entraînent une dégradation notable des performances des systèmes, se traduisant par une augmentation du temps de réponse et une surcharge des ressources serveurs. Dans les cas les plus critiques, elles peuvent conduire à une indisponibilité totale des services.

D'un point de vue économique, les impacts sont également considérables, notamment en raison des pertes financières liées à l'interruption des services, en particulier pour

les organisations fortement dépendantes des plateformes en ligne. À cela s'ajoutent les coûts supplémentaires associés à la mitigation des attaques et au renforcement des dispositifs de sécurité.

Enfin, les attaques DDoS peuvent également porter atteinte à la réputation de l'organisation, entraînant une perte de confiance de la part des clients et des partenaires, surtout lorsque les incidents sont répétés ou prolongés.[3]

4.2.4 Défis et Complexités des Attaques DDoS

Les attaques par déni de service distribué (DDoS) constituent une menace majeure en cybersécurité en raison de leur impact important sur la disponibilité des systèmes et la continuité des services.

- Les attaques DDoS génèrent un volume massif de trafic malveillant provenant de sources multiples et distribuées.
- Leur nature distribuée rend leur détection, leur analyse et leur neutralisation particulièrement complexes.
- Elles reposent sur des réseaux de machines compromises (botnets), exploitées à l'insu de leurs propriétaires.
- La dispersion des sources complique l'identification et l'attribution des attaquants.
- Elles peuvent provoquer une saturation des ressources réseau et serveurs, entraînant une dégradation des performances ou une indisponibilité des services.

- Même sans arrêt complet, elles peuvent causer des ralentissements importants et affecter la qualité de service.
- Elles entraînent des impacts opérationnels et économiques significatifs ainsi qu'un risque pour la réputation des organisations.

Ainsi, les attaques DDoS représentent un défi majeur pour la sécurité des systèmes modernes, nécessitant des mécanismes de protection avancés et efficaces.[4]

4.3 Les systèmes de détection traditionnels

4.3.1 Systèmes de détection basés sur les signatures

Les systèmes de détection basés sur les signatures constituent une approche classique en cybersécurité, largement utilisée pour l'identification des menaces connues. Leur principe repose sur la comparaison des activités observées au sein d'un système d'information avec une base de données contenant des modèles de comportements malveillants déjà identifiés, appelés signatures.

Lorsqu'une activité réseau ou un comportement correspond à une signature existante, le système déclenche automatiquement une alerte signalant la présence potentielle d'une attaque. Cette méthode s'avère particulièrement efficace pour la détection de menaces déjà répertoriées, telles que certains logiciels malveillants ou exploits connus. Cependant, cette approche présente une limite majeure : elle est incapable de détecter les attaques nouvelles ou inconnues (zero-day), dans la mesure où celles-ci ne disposent pas encore de signatures dans la base de données. Par conséquent, son efficacité dépend fortement de la mise à jour régulière et continue des signatures.[5]

4.3.2 Définition des IDS

Un système de détection d'intrusion (IDS) est un dispositif de sécurité conçu pour surveiller en continu les activités d'un système d'information ou d'un réseau, afin d'identifier les comportements anormaux ainsi que les tentatives d'intrusion.

Il repose sur l'analyse des flux de données et des événements système dans le but de détecter toute violation des politiques de sécurité et de générer des alertes appropriées.

Contrairement aux systèmes de prévention, l'IDS n'intervient pas directement pour bloquer les attaques. Il joue néanmoins un rôle essentiel dans l'amélioration de la visibilité sur les menaces et dans la détection précoce des incidents de sécurité, facilitant ainsi leur analyse et leur prise en charge.[6]

4.3.3 Définition des SIEM

Un système de gestion des informations et des événements de sécurité (SIEM – Security Information and Event Management) est une solution de cybersécurité permettant de collecter, centraliser et analyser en temps réel les données de sécurité issues de différentes sources d'un système d'information, telles que les serveurs, les équipements réseau, les applications et les utilisateurs.

Son objectif principal est de fournir une visibilité globale sur l'état de sécurité d'une organisation, en corrélant les événements afin de détecter les menaces, d'identifier les incidents de sécurité et de faciliter leur analyse ainsi que leur traitement.

Ainsi, le SIEM constitue un outil essentiel pour améliorer la détection des attaques, accélérer la réponse aux incidents et renforcer la gestion globale de la sécurité au sein des infrastructures informatiques.[7]

4.3.4 Définition des EDR

L'EDR (Endpoint Detection and Response) est une solution avancée de cybersécurité conçue pour protéger les terminaux, tels que les ordinateurs, les serveurs et les appareils mobiles. Elle repose sur une surveillance continue des activités des endpoints afin de détecter, analyser et répondre aux menaces informatiques, notamment les logiciels malveillants et les ransomwares.

Cette solution collecte et analyse en permanence les événements et comportements des systèmes à l'aide de techniques avancées, notamment l'analyse comportementale, afin d'identifier toute activité suspecte ou anormale. Elle fournit également des informations contextuelles détaillées, facilitant ainsi l'investigation des incidents et la mise en œuvre de mesures de remédiation appropriées.

Ainsi, l'EDR joue un rôle essentiel dans la détection proactive des menaces, l'amélioration de la visibilité sur les activités des systèmes et le renforcement de la capacité de réponse face aux cyberattaques.[8]

4.4 Limites des solutions traditionnelles

4.4.1 Limites des méthodes de détection basées sur les signatures

Les systèmes de détection basés sur les signatures permettent d'identifier efficacement les menaces connues en comparant les activités du système à une base de données de signatures malveillantes. Toutefois, cette approche présente plusieurs limites importantes :

- Incapacité à détecter les attaques inconnues ou de type zero-day, en raison de l'absence de signatures correspondantes dans la base de données.

- Risque de laisser passer des menaces non répertoriées, ce qui réduit la couverture de sécurité.
- Nécessité d'une mise à jour constante des bases de signatures afin de maintenir un niveau d'efficacité élevé face à l'évolution des cybermenaces.
- Diminution rapide de la performance de détection en cas de retard ou d'absence de mises à jour.
- Génération potentielle d'un volume important d'alertes, incluant des faux positifs.
- Surcharge pour les analystes en sécurité, compliquant l'analyse et la priorisation des incidents.[9]

4.4.2 Limites des IDS

Les systèmes de détection d'intrusion (IDS) présentent plusieurs limites importantes. Bien qu'ils soient efficaces pour la surveillance du trafic réseau et l'identification des comportements suspects, ils reposent généralement sur des règles ou des signatures, ce qui limite leur capacité à détecter des attaques nouvelles ou inconnues.

Par ailleurs, les IDS génèrent souvent un volume élevé d'alertes, dont une partie peut correspondre à des faux positifs. Cette situation complique le travail des équipes de sécurité, entraîne une surcharge d'analyse et peut réduire l'efficacité globale de la réponse aux incidents.[10]

4.4.3 Limites des SIEM

Les solutions SIEM permettent de centraliser et d'analyser les événements de sécurité provenant de multiples sources. Toutefois, elles présentent certaines contraintes

importantes. Leur efficacité dépend fortement de la qualité et de la configuration des sources de logs, ce qui peut impacter directement la pertinence des analyses.

Par ailleurs, l'exploitation des données requiert une expertise avancée, notamment pour l'interprétation des corrélations entre événements, ce qui peut ralentir la détection et la prise en charge des incidents. Enfin, la gestion de volumes importants de logs engendre une complexité opérationnelle accrue, ainsi que des coûts élevés en matière de stockage et de traitement des données.[11]

4.4.4 Limites des EDR

Les solutions EDR offrent une protection avancée au niveau des terminaux, notamment les postes de travail, les serveurs et les appareils mobiles. Toutefois, leur périmètre d'action reste limité aux endpoints, sans couverture globale de l'ensemble de l'infrastructure réseau.

Par ailleurs, leur efficacité dépend fortement de l'installation et de la configuration correcte de l'agent sur chaque terminal, ce qui peut représenter une contrainte importante dans les environnements complexes, distribués ou hybrides.[12]

4.5 Problème de fragmentation des outils

L'un des principaux défis des solutions traditionnelles de cybersécurité réside dans la fragmentation des outils. En effet, des technologies telles que les IDS, les SIEM et les EDR fonctionnent généralement de manière indépendante, avec leurs propres sources de données et mécanismes d'alerte. Cette séparation limite la capacité à obtenir une vision globale et cohérente des menaces, tout en compliquant la corrélation des événements de sécurité.

En conséquence, les équipes de sécurité doivent gérer plusieurs plateformes distinctes, ce qui augmente la complexité opérationnelle et peut ralentir la détection ainsi que la réponse aux incidents.

Cette limitation met en évidence la nécessité d'adopter une approche unifiée et intelligente, capable de centraliser les informations, d'automatiser la détection et d'optimiser la réponse aux menaces, à l'image des solutions XDR.

4.6 L'intelligence artificielle (IA)

4.6.1 Intelligence artificielle pour la détection DDoS

4.6.1.1 Définition de l'intelligence artificielle

L'intelligence artificielle (IA) [13] désigne un ensemble de techniques permettant à un système informatique d'analyser automatiquement de grandes quantités de données afin d'identifier des comportements anormaux, détecter des menaces et assister la prise de décision. Dans le domaine de la cybersécurité, l'IA est principalement utilisée pour améliorer la détection des cyberattaques grâce à des modèles d'apprentissage capables de reconnaître des schémas suspects dans le trafic réseau. Dans le cadre de notre projet, l'intelligence artificielle est exploitée pour analyser les flux réseau et détecter automatiquement les attaques DDoS en temps réel.

4.6.1.2 L'apprentissage automatique

L'apprentissage automatique [18], également appelé machine learning ou apprentissage statistique, constitue un domaine fondamental de l'intelligence artificielle. Il repose sur des approches mathématiques et statistiques permettant aux systèmes informatiques d'apprendre à partir des données.

Concrètement, les algorithmes d'apprentissage automatique sont capables d'améliorer leurs performances dans l'exécution de tâches sans être explicitement programmés pour chaque situation. Ce domaine englobe plusieurs aspects, notamment la conception, l'analyse, l'optimisation et l'implémentation de modèles d'apprentissage. Il est souvent qualifié de discipline statistique, dans la mesure où il vise à construire des modèles capables de minimiser l'erreur sur un ensemble de données, afin de produire des prédictions aussi précises que possible.

4.6.2 Détection des anomalies basée sur l'IA

Comparaison : détection d'attaque par signature vs. Machine learning

Face à l'évolution continue des cyberattaques, les approches traditionnelles de détection, notamment celles basées sur les signatures, montrent aujourd'hui leurs limites. Elles deviennent de moins en moins efficaces face à la complexité croissante des attaques et à leur capacité d'adaptation rapide.

Dans ce contexte, le tableau suivant met en évidence les différences entre la détection basée sur les signatures et les approches fondées sur l'apprentissage automatique, en termes de méthodologie, d'efficacité et de capacité d'adaptation aux nouvelles menaces.[19]

Tableau 1:détection d'attaque par signature vs. Machine learning

Détection basée sur les signatures	Apprentissage automatique
Basée sur des modèles d'actions ou de codes malveillants	Se base sur l'analyse comportementale des logiciels
Efficace pour identifier les menaces connues	Capable d'identifier de nouvelles formes de logiciels malveillants
Limitée aux menaces pour lesquelles une signature existe	Peut détecter les menaces non signées et émergentes
Nécessite des mises à jour constantes des bases de signatures	S'adapte et apprend en fonction des données qu'il analyse
Dépendance à la maintenance manuelle des bases de signatures	Peut développer des modèles avancés de détection sans intervention humaine
Vulnérable aux nouvelles formes de logiciels malveillants	Capable de reconnaître les variations subtiles dans les comportements malveillants
Analyse statique du code	Analyse dynamique et adaptable des comportements malveillants
Peut manquer les logiciels malveillants ayant muté	Capable de détecter les logiciels malveillants même s'ils ont muté
Limite la détection à des correspondances de code spécifiques	Offre une détection plus dynamique et adaptable

Cette comparaison démontre que l'apprentissage automatique surpasse la détection basée sur les signatures en proposant une approche plus dynamique et adaptable pour identifier les menaces émergentes. Cela renforce la sécurité des organisations dans un environnement cybernétique en constante évolution.

4.6.3 Intelligence artificielle dans la cybersécurité

L'intelligence artificielle occupe aujourd'hui une place centrale dans l'évolution des systèmes de cybersécurité modernes. Elle permet d'analyser et d'exploiter de grandes quantités de données issues des réseaux et des systèmes d'information afin de détecter, en temps réel, les menaces potentielles ainsi que les comportements anormaux.

Grâce à ses capacités d'apprentissage et d'adaptation, l'IA améliore significativement la détection des attaques, qu'elles soient connues ou totalement nouvelles. Elle est également capable d'identifier des schémas complexes difficilement détectables par les approches traditionnelles basées sur les signatures.

Par ailleurs, l'intelligence artificielle joue un rôle essentiel dans l'automatisation de la réponse aux incidents de sécurité. Elle contribue à réduire le temps de détection et d'intervention, tout en augmentant la précision des alertes et en limitant l'impact des attaques sur les systèmes d'information.

Ainsi, l'intégration de l'IA dans les solutions de cybersécurité renforce la protection globale des infrastructures numériques et optimise la gestion des menaces dans des environnements de plus en plus complexes et dynamiques.[15]

4.7 XDR et corrélation des événements

4.7.1 Définition du XDR

Le XDR (Extended Detection and Response) est une solution avancée de cybersécurité qui permet de collecter, corréler et analyser les données de sécurité issues de différentes sources, telles que les terminaux, les réseaux, le cloud, les systèmes d'identité et les applications, au sein d'une plateforme unifiée.

Cette approche intégrée offre une visibilité globale sur l'ensemble du système d'information et facilite la détection, l'investigation et la réponse aux menaces en temps réel. Elle s'appuie généralement sur des techniques d'analyse avancées, notamment l'intelligence artificielle et l'automatisation.

Ainsi, le XDR vise à améliorer l'efficacité des opérations de sécurité en simplifiant les outils utilisés, en facilitant la corrélation des événements de sécurité et en accélérant la réponse face aux cyberattaques.[16]

4.7.2 Architecture générale du XDR

Une architecture XDR repose généralement sur plusieurs composants interconnectés permettant la collecte, l'analyse et la réponse aux incidents de sécurité. Elle comprend principalement:

- Des sources de données telles que les équipements réseau, les endpoints, les serveurs et les journaux de sécurité ;
- Un moteur de collecte et de centralisation chargé de regrouper les événements provenant des différentes sources ;
- Un système d'analyse utilisant des mécanismes de corrélation et d'intelligence artificielle pour identifier les comportements suspects ;

- Un module de réponse automatisée permettant d'appliquer des actions de protection et de mitigation.

Cette architecture permet d'obtenir une vision globale et centralisée des activités du système d'information.

4.7.3 Centralisation des alertes

La centralisation des alertes constitue l'un des principaux avantages du XDR. Les événements de sécurité générés par différentes solutions sont regroupés dans une plateforme unique afin de simplifier leur gestion et leur analyse.

Cette centralisation améliore la visibilité sur les incidents et réduit la complexité liée à l'utilisation de plusieurs outils indépendants. Elle permet également aux analystes de sécurité d'identifier plus rapidement les activités malveillantes et de mieux comprendre le contexte des attaques.

4.7.4 Corrélation des événements

La corrélation des événements consiste à analyser et relier plusieurs alertes provenant de différentes sources afin d'identifier une attaque ou un comportement suspect.

Grâce à cette approche, le XDR est capable de détecter des menaces complexes qui pourraient passer inaperçues lorsqu'elles sont analysées séparément. La corrélation permet également de réduire le nombre de faux positifs en regroupant les événements réellement liés à une activité malveillante.

L'intégration de techniques d'intelligence artificielle et de machine learning améliore davantage cette analyse en permettant la détection de comportements anormaux dans le trafic réseau.

4.7.5 Réponse automatisée

Le XDR intègre des mécanismes de réponse automatisée permettant de réagir rapidement aux incidents détectés. Lorsqu'une menace est identifiée, le système peut automatiquement appliquer des actions de protection, telles que :

- Le blocage d'une adresse IP suspecte.
- Isolation d'un équipement compromise.
- La génération d'alertes de sécurité.
- L'enregistrement des événements dans des journaux d'analyse.

Dans le cadre de notre projet, cette réponse automatisée est réalisée à travers des règles iptables permettant de bloquer automatiquement les adresses IP générant des attaques DDoS détectées par le modèle d'intelligence artificielle.

4.7.6 Comparaison des solutions de sécurité : IDS, SIEM, EDR et XDR

Tableau 2: Étude comparative des mécanismes de détection et de réponse aux incidents de sécurité

Critère	IDS	SIEM	EDR	XDR
Définition	Système de détection	Plateforme de centralisation et	Solution de protection et de	Plateforme unifiée de détection et

	d'intrusion surveillant le réseau ou les systèmes	d'analyse des logs de sécurité	réponse au niveau des terminaux	réponse étendue et intelligente
Objectif principal	Détecter les intrusions et générer des alertes	Corréler et analyser les événements de sécurité	Détecter et répondre aux menaces sur les endpoints	Détection globale, corrélation avancée et réponse automatisée aux attaques
Périmètre de couverture	Réseau / systèmes	Infrastructure complète (logs, réseaux, systèmes)	Endpoints (PC, serveurs, mobiles)	Réseau + endpoints + cloud + applications
Vision de sécurité	Locale	Centralisée	Orientée endpoints	Globale et unifiée (360°)
Analyse des données	Signatures ou anomalies	Corrélation de logs	Analyse comportementale	Analyse intelligente basée sur IA et corrélation multi- sources
Réponse aux incidents	Alerte uniquement	Partielle	Oui sur endpoints	Réponse automatisée et orchestrée sur tout le système

Automatisation	Faible	Moyenne	Élevée	Très élevée
Force principale	Détection simple des intrusions	Centralisation des événements	Protection avancée des terminaux	Intégration complète + intelligence + rapidité de réponse
Limite principale	Pas de réponse active	Complexité et surcharge de logs	Limité aux endpoints	Solution complexe mais la plus complète et évolutive

Notre projet de XDR intelligent s’inscrit dans une évolution des solutions traditionnelles telles que les IDS, SIEM et EDR, en proposant une approche unifiée, centralisée et intelligente de la cybersécurité. Il permet une corrélation avancée des événements de sécurité, une détection proactive des menaces ainsi qu’une réponse automatisée en temps réel.

Ainsi, cette solution contribue à renforcer significativement la posture de sécurité globale du système d’information, en améliorant la visibilité, la rapidité de réaction et l’efficacité de la gestion des incidents.

4.8 Conclusion

L’évolution rapide et la sophistication croissante des cyberattaques, notamment les attaques par déni de service distribué (DDoS), obligent les organisations à renforcer en permanence leurs mécanismes de détection et de réponse afin de garantir la sécurité et la disponibilité de leurs systèmes d’information. Les approches

traditionnelles, bien qu'efficaces dans certains contextes, montrent aujourd'hui leurs limites face à la complexité et à l'interconnexion des environnements numériques modernes.

Dans ce cadre, notre projet de plateforme XDR intelligente s'inscrit dans une approche évolutive, centralisée et orientée vers l'automatisation de la cybersécurité. Il vise à intégrer différentes sources de données de sécurité afin d'assurer une détection proactive des menaces, une corrélation avancée des événements ainsi qu'une réponse automatisée en temps réel.

Ainsi, cette solution permet de renforcer la visibilité globale sur le système d'information et d'améliorer significativement la capacité de protection face aux cyberattaques, en particulier les attaques DDoS.

Chapitre 3 : Implémentation de la plateforme XDR intelligente

5.1 Introduction

Dans ce chapitre, nous présentons la conception et l'implémentation de la solution proposée pour la détection et la mitigation des attaques DDoS basée sur l'intelligence artificielle. L'objectif principal de cette solution est d'analyser le trafic réseau en temps réel afin d'identifier automatiquement les comportements malveillants et de déclencher des mécanismes de réponse adaptés.

Nous présentons d'abord les besoins fonctionnels et non fonctionnels du système, puis l'architecture globale de la solution ainsi que les différents outils matériels et logiciels utilisés. Ensuite, nous détaillons les étapes de collecte et de prétraitement des données, la conception du modèle d'intelligence artificielle ainsi que le mécanisme de détection et de réponse automatique basé sur iptables. Enfin, nous présentons le tableau de bord permettant la supervision et la visualisation des alertes générées par le système.

5.2 Analyse des besoins

L'analyse des besoins permet d'identifier les principales fonctionnalités attendues du système ainsi que les contraintes techniques nécessaires à son bon fonctionnement.

Cette étape constitue une base importante pour la conception de la solution proposée.

5.2.1 Besoins fonctionnels

Les besoins fonctionnels représentent les principales fonctionnalités que le système doit assurer. Dans le cadre de notre projet, la solution doit permettre de :

- Capturer le trafic réseau en temps réel ;
- Collecter les données réseau à partir des interfaces de communication ;

- Prétraiter les données afin de les rendre exploitables par le modèle d'intelligence artificielle ;
- Détecter automatiquement les attaques DDoS ;
- Générer des alertes lors de la détection d'un comportement suspect ;
- Bloquer automatiquement les adresses IP malveillantes à l'aide des règles iptables ;
- Enregistrer les événements et les alertes dans des fichiers de logs ;
- Afficher les informations de supervision dans un tableau de bord.

5.2.2 Besoins non fonctionnels

Les besoins non fonctionnels définissent les contraintes techniques et les critères de qualité que doit respecter le système afin d'assurer son efficacité et sa fiabilité.

La solution proposée doit principalement :

- Offrir une détection rapide des attaques afin de réduire le temps de réponse ;
- Garantir une bonne précision de détection avec un faible taux de faux positifs ;
- Assurer la disponibilité du système même en présence d'un trafic important ;
- Permettre l'analyse des données en temps réel ;
- Fournir une interface simple pour la visualisation des alertes ;
- Assurer la stabilité et la fiabilité du mécanisme de blocage automatique ;
- Etre évolutive afin de permettre l'intégration future d'autres types d'attaques ou de nouvelles fonctionnalités inspirées des architectures XDR.

5.3 Architecture globale du système

L'architecture de la solution proposée repose sur plusieurs modules permettant la capture, l'analyse et la détection des attaques DDoS en temps réel. Le système utilise des techniques d'intelligence artificielle afin d'identifier les comportements suspects dans le trafic réseau et d'appliquer automatiquement des mécanismes de mitigation.

L'ensemble des composants communiquent de manière séquentielle afin d'assurer le traitement des données depuis la capture du trafic jusqu'à la génération des alertes et le blocage automatique des adresses IP malveillantes.

Le schéma suivant présente l'architecture globale de la solution proposée ainsi que les différentes étapes de traitement du trafic réseau.

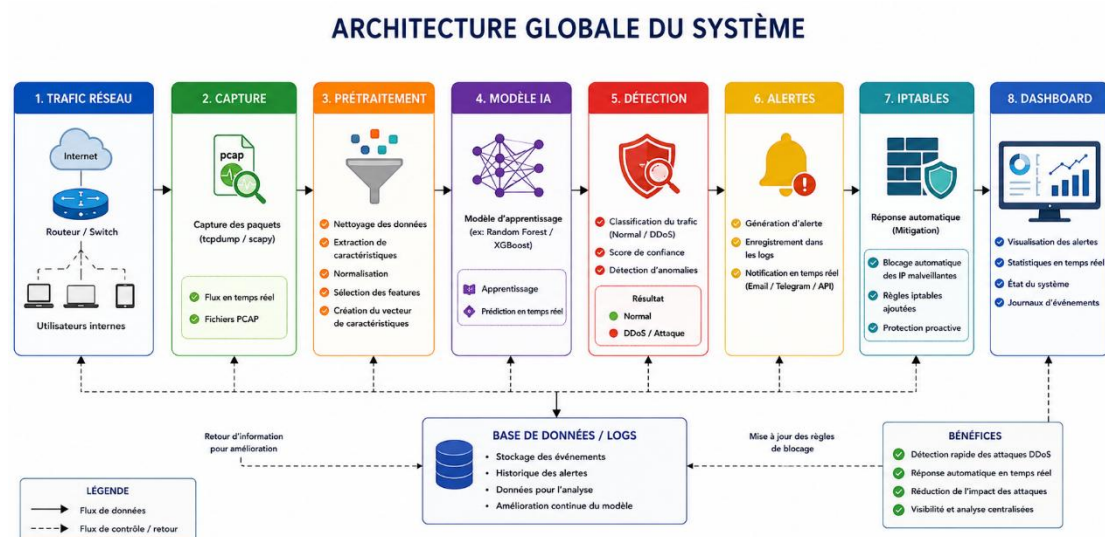


Figure 3 : Schéma de fonctionnement du système

L'architecture globale du système proposé repose sur une chaîne de traitement composée de huit étapes principales :

- Trafic réseau : génération et circulation des données sur le réseau.
- Capture : interception et collecte des flux réseau en temps réel.

- Prétraitement : nettoyage, transformation et préparation des données pour l'analyse.
- Modèle IA : application du modèle d'intelligence artificielle pour analyser les caractéristiques du trafic.
- Détection : identification des comportements normaux ou malveillants (attaques DDoS).
- Alertes : génération d'alertes en cas de détection d'une activité suspecte.
- Iptables : application automatique de règles de blocage pour neutraliser les adresses IP malveillantes.
- Dashboard : visualisation et supervision des alertes et des événements de sécurité en temps réel.

5.4 Environnement de travail

5.4.1 Matériel utilisé

La configuration de la machine sur laquelle notre système a été développé est la suivante :

Tableau 3: Configuration de la machine

Nom de l'appareil	ASUS
Edition	Windows 11 Famille Unilingue
Types de Système	Système d'exploitation 64 bits, processeur x64
Mémoire RAM	8,00 Go
Processeur	11th Gen Intel(R) Core (TM) i5-1135G7 @ 2.40GHz 2.42 GHz

5.4.2 Environnement logiciel

Le développement de la plateforme XDR intelligente a été réalisé principalement avec Python en utilisant l'éditeur Visual Studio Code (VS Code). Cet environnement de développement facilite l'écriture, l'exécution et le débogage du code grâce à ses fonctionnalités intégrées.

Python a été choisi en raison de sa simplicité, de sa flexibilité et de la richesse de ses bibliothèques spécialisées dans le traitement des données et l'intelligence artificielle, notamment NumPy, Pandas, TensorFlow et Scikit-learn. Ces outils ont permis de développer efficacement le modèle de détection des attaques DDoS intégré dans la plateforme XDR.

5.4.3 Données utilisées pour l'expérimentation

L'étude utilise le jeu de données CIC-DDoS-2019 [17], fourni par l'Institut canadien de cybersécurité (CIC), qui contient des données de flux réseau réelles, comprenant divers types d'attaques DDoS courantes et récentes. Les données sont présentées dans des formats condensés, principalement sous forme de métadonnées sur les connexions réseau. Chaque donnée représente un ensemble de paquets partageant des propriétés spécifiques dans une fenêtre temporelle donnée, sans inclure de charge utile.

Le jeu de données se décline en deux versions : des données PCAP brutes et des données CSV. Les auteurs ont utilisé l'analyseur de trafic CICFlowMeter-V3 pour extraire plus de 80 caractéristiques à partir des fichiers PCAP. Ces caractéristiques ont été ensuite sauvegardées dans des fichiers CSV structurés et étiquetés par l'Université du Nouveau-Brunswick.

L'ensemble de données comprend également 12 types d'attaques DDoS différents. Les données CIC-DDoS-2019 ont été collectées sur une période de deux jours. Les tableaux suivants fournissent des informations sur les différents types d'attaques, ainsi que leur taille en kilooctets, pour chaque jour.

5.4.3.1 Types d'attaques DDoS répertoriés dans les données

Les attaques DDoS basées sur la réflexion : [17] sont caractérisées par la dissimulation de l'identité de l'attaquant grâce à l'utilisation de composants tiers légitimes. Les paquets sont dirigés vers des serveurs réflecteurs par les attaquants, ces paquets ayant une adresse IP source falsifiée pour correspondre à celle de la victime visée. L'objectif est de submerger la victime avec des paquets de réponse. Ces attaques peuvent être réalisées via des protocoles de couche applicative utilisant les protocoles de transport TCP et UDP, ou une combinaison des deux. Dans cette catégorie, les attaques TCP incluent MSSQL et SSDP, tandis que les attaques UDP (ou CharGen), NTP et TFTP. Certaines attaques peuvent utiliser à la fois TCP et UDP, comme DNS, LDAP, NETBIOS et SNMP.

Les attaques DDoS basées sur l'exploitation : [17] sont caractérisées par le camouflage de l'identité de l'attaquant grâce à l'utilisation de composants tiers légitimes. Les paquets sont envoyés vers des serveurs réflecteurs par les attaquants, ces paquets ayant une adresse IP source falsifiée pour correspondre à celle de la victime visée. L'objectif est également de submerger la victime avec des paquets de réponse. Ces attaques peuvent être réalisées via des protocoles de couche applicative utilisant les protocoles de transport TCP et UDP. Les attaques basées sur TCP

comprennent l'inondation SYN, tandis que les attaques basées sur UDP comprennent l'inondation UDP et l'inondation UDP-Lag.

La figure ci-dessous illustre la répartition des attaques DDoS

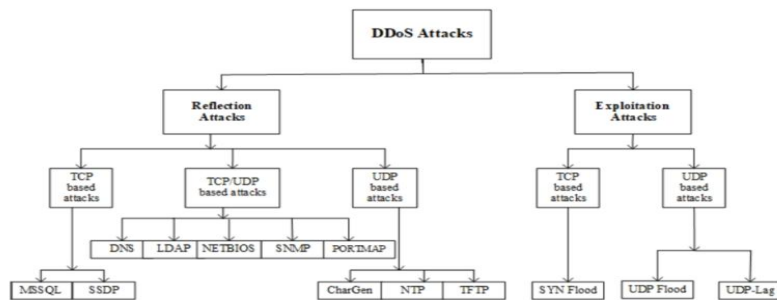


Figure 4 : Taxonomie des attaques DDoS [17]

Tableau 4: Résultats détectés le 03/11/2019

Nom d'Attaques	Taille en ko
DrDoS_DNS.csv	2 083 309
DrDoS_LDAP.csv	895 804
DrDoS_MSSQL.csv	1 844 905
DrDoS_NetBIOS.csv	1 657 696
DrDoS_NTP.csv	629 892
DrDoS_SNMP.csv	2 121 660
DrDoS_SSDP.csv	1 223 327
DrDoS_UDP.csv	1 470 742
Syn.csv	622 376
TFTP.csv	695 388
UDPLag.csv	17 963

Tableau 5: Résultats détectés le 01/12/2019

Nom d'attaques	Taille en ko
LDAP.csv	850 976
MSSQL.csv	2 330 299
NetBIOS.csv	1 385 223
Portmap.csv	76 769
Syn.csv	1 833 372
UDP.csv	1 750 779
UDPLag.csv	312 299

Une fois que j'ai téléchargé ces données depuis le site

<https://www.unb.ca/cic/datasets/DDoS-2019.html> (17/05/2026), je procède à une

étape cruciale de prétraitement des données. Cette étape revêt une grande importance

pour la mise en place d'une plateforme de détection d'attaques, car le Deep learning

exige des données bien organisées et nettoyées.

5.4.3.2 La préparation des données

L'efficacité des algorithmes de machine learning repose en grande partie sur la qualité

et la quantité des données d'apprentissage. Un ensemble de données volumineux et de

haute qualité permet d'obtenir des résultats plus précis et fiables. Bien que nous

disposions d'un volume de données adéquat, un déséquilibre entre les classes

nécessite une réduction des données.

5.4.3.3 Réduction des données

La préparation des données implique deux étapes essentielles : la réduction des

données et la correction des labels. En raison du nombre important d'échantillons dans les tableaux 2 et 3, une réduction significative est nécessaire. Pour ce faire, nous traitons les 17 fichiers CSV afin de gérer les différentes erreurs de données telles que les valeurs manquantes, les valeurs aberrantes et les doublons. Une analyse minutieuse est ensuite effectuée pour identifier toute occurrence de "NAN" (Not A Number) ou "INF" (Infinite Value). Les colonnes et les lignes contenant ces valeurs sont ensuite supprimées. Les algorithmes de machine learning sont sensibles à ces valeurs et peuvent affecter négativement les performances des modèles finaux.

```
# Nettoyage noms colonnes
df.columns = df.columns.str.strip()

# Conversion labels
df["Label"] = df["Label"].apply(
    lambda x: 0 if x == "BENIGN" else 1
)

# Séparation X / y
y = df["Label"].values
X = df.drop("Label", axis=1)

# Conversion numérique
X = X.apply(pd.to_numeric, errors='coerce')

# Remplacement valeurs infinies
X = X.replace(np.inf, max_val)
X = X.replace(-np.inf, max_val)

# Valeurs manquantes
X = X.fillna(0)

return X, y

# =====
# LECTURE FICHIERS TRAIN
# =====
with open('train_files.txt') as f:
    train_files = [line.strip() for line in f]

with open('test_files.txt') as f:
    test_files = [line.strip() for line in f]
```

Figure 5 : Extrait du code de prétraitement des données pour le XDR intelligent
Cette figure illustre les étapes de préparation des données avant leur injection dans le modèle de détection d'intrusions. On y distingue trois blocs fonctionnels :

Nettoyage et conversion des labels

Les espaces dans les noms de colonnes sont supprimés, et la colonne `Label` est

binarisée (0 = BENIGN, 1 = attaque). Cette transformation est essentielle pour l'apprentissage supervisé.

Prétraitement des caractéristiques (features)

Conversion forcée en type numérique (`pd.to_numeric`) pour gérer les valeurs non numériques issues des logs bruts.

Remplacement des valeurs infinies (provenant par exemple de divisions par zéro) par une valeur maximale prédéfinie `max_val`.

Remplissage des valeurs manquantes par 0 pour éviter les erreurs lors de l'entraînement.

Gestion des sources de données

La lecture des fichiers `train_files.txt` et `test_files.txt` permet de charger dynamiquement les chemins des fichiers d'entraînement et de test, facilitant ainsi le passage à l'échelle dans un contexte XDR (volumes de données importants).

Ce prétraitement garantit que les données sont propres, homogènes et prêtes à être utilisées par des modèles de machine learning

5.4.3.4 Division des données

La division des données est une étape cruciale dans le développement d'un modèle de détection d'attaques DDoS. Les données sont divisées en un ensemble d'entraînement, utilisé pour entraîner le modèle, et un ensemble de test, réservé à l'évaluation de ses performances. Dans ce cas, l'ensemble d'entraînement contient les données du 03/11/2019 présentées dans le tableau 4, tandis que l'ensemble de test comprend celles du 01/12/2019 présentées dans le tableau 5. Cette division garantit une évaluation

objective du modèle avant son déploiement.

La préparation et la division des données sont des étapes essentielles dans le développement d'un système de détection d'attaques DDoS basé sur la machine learning. En suivant ces étapes, nous pouvons nous assurer que les données sont de haute qualité et prêtes à être utilisées par les algorithmes de machine learning pour construire un modèle précis et fiable.

5.4.3.5 Bibliothèques et technologies utilisées

La mise en œuvre de la plateforme XDR intelligente repose sur un ensemble de bibliothèques Python spécialisées dans le traitement de données, l'apprentissage automatique et le deep learning. Le tableau ci-dessous récapitule les principales bibliothèques utilisées et leurs rôles respectifs.

Tableau 6: bibliothèques utilisées

Bibliothèque	Rôle
Pandas	Manipulation, nettoyage et analyse des DataFrames
NumPy	Calculs numériques et gestion des tableaux multidimensionnels
Scikit-learn	Prétraitement, normalisation, sélection de features, métriques d'évaluation
TensorFlow/Keras	Construction, entraînement et sauvegarde des réseaux de neurones
Matplotlib	Visualisation des résultats et des courbes d'apprentissage

Justification des choix

- Pandas/NumPy : Le jeu de données CIC-DDoS-2019 est volumineux (plusieurs millions de lignes). Ces bibliothèques offrent des performances optimales pour le chargement, le filtrage et la transformation des données.
- Scikit-learn : Elle fournit des outils essentiels comme `QuantileTransformer` (normalisation robuste aux valeurs aberrantes), `train_test_split` (division des données) et les métriques d'évaluation.
- Keras/TensorFlow : Framework de deep learning permettant de construire rapidement des réseaux de neurones denses (fully connected) avec des couches de dropout pour éviter le surapprentissage.

5.4.3.6 Features sélectionnées

Après l'étape d'exploration et de sélection décrite précédemment, 20 caractéristiques ont été retenues pour l'entraînement des modèles. Ces features sont listées dans le fichier `features.txt` :

```
Fwd Packet Length Min
Fwd Packet Length Std
Flow IAT Mean
Flow IAT Max
Fwd IAT Mean
Fwd IAT Max
Fwd Header Length
Fwd Packets/s
Min Packet Length
Max Packet Length
Packet Length Std
ACK Flag Count
Average Packet Size
Fwd Header Length.1
Subflow Fwd Bytes
Init_Win_bytes_forward
min_seg_size_forward
Label
```

5.4.3.7 Interprétation des features

Ces caractéristiques peuvent être regroupées en quatre catégories :

Tableau 7: catégories features

Catégorie	Features concernées	Rôle dans la détection DDoS
Durée et temps	Flow Duration, Flow IAT Mean/Max, Fwd IAT Mean/Max	Une attaque DDoS modifie les intervalles inter-arrivées des paquets
Taille des paquets	Fwd Packet Length Min/Max/Std, Min/Max Packet Length, Packet Length Std	Les paquets malveillants ont souvent des tailles anormales
Flags TCP	ACK Flag Count	Les attaques SYN flood génèrent un déséquilibre des flags
Débit	Fwd Packets/s, Average Packet Size, Subflow Fwd Bytes	Le trafic DDoS sature la bande passante

5.4.4 Type d'apprentissage utilisé

Deux approches d'apprentissage supervisé ont été développées et comparées :

Classifieur binaire (ML_binary_classifier.py)

- Objectif : Distinguer le trafic légitime (BENIGN) du trafic malveillant (toute attaque DDoS).
- Type : Classification binaire supervisée.
- Encodage des labels : 0 pour BENIGN, 1 pour toute autre attaque.
- Utilisation : Détection rapide, idéale pour une première alerte.

Classifieur multi-classes (ML_category_classifier.py)

- Objectif : Identifier le type spécifique d'attaque DDoS (Syn, UDP-lag, DrDoS_NTP, etc.).
- Type : Classification multi-classes supervisée.
- Encodage des labels : One-hot encoding (chaque catégorie d'attaque est convertie en vecteur binaire).
- Utilisation : Analyse forensique et réponse adaptée au type d'attaque.

Justification du choix supervisé

L'apprentissage supervisé a été retenu car :

- Le jeu de données CIC-DDoS-2019 est parfaitement étiqueté (ground truth disponible).
- Les attaques DDoS ont des signatures comportementales bien définies.
- L'objectif opérationnel est de classer le trafic en temps réel.

5.5 Développement du modèle de détection DDoS

Le développement du modèle de détection des attaques DDoS s'est déroulé en plusieurs étapes afin d'obtenir un système efficace basé sur l'intelligence artificielle.

5.5.1 • Extraction et préparation des données

Cette étape consiste à utiliser le dataset CIC-DDoS-2019. Les données sont d'abord nettoyées en supprimant les valeurs manquantes ou incorrectes comme NaN et Infinity.

Ensuite, les 20 caractéristiques les plus importantes sont sélectionnées pour l'apprentissage du modèle.

Les données sont divisées en deux parties :

- 80 % pour l'entraînement
- 20 % pour le test

5.5.2 Analyse des données

Cette étape permet de vérifier la répartition des données entre trafic normal et attaques.

Les étiquettes sont transformées en :

- 0: trafic normal (BENIGN)
- 1: attaque

Pour les attaques multiples, un encodage one-hot est utilisé.

5.5.3 • Normalisation des données

Les données sont normalisées avec QuantileTransformer afin de mettre toutes les variables sur une même échelle et améliorer l'apprentissage du modèle.

• Entraînement du modèle IA

Le modèle est basé sur un réseau de neurones (Keras / TensorFlow).

- Il contient plusieurs couches de neurones ($64 \rightarrow 64 \rightarrow 32$).
- La fonction ReLU est utilisée pour l'apprentissage.

- Le Dropout est utilisé pour éviter le surapprentissage.
- L'optimiseur utilisé est Adam.

Deux modèles sont développés:

- Un modèle binaire (normal / attaque)
- Un modèle multi-classes (types d'attaques)

```
ta.py > ...
import os
import numpy as np
from keras.models import Sequential, load_model
from keras.layers import Dense

# -----
# MODEL
# -----
model = Sequential([
    Dense(12, input_dim=8, activation='relu'),
    Dense(8, activation='relu'),
    Dense(1, activation='sigmoid')
])

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# -----
# DATA (juste test)
# -----
X_train = np.random.rand(100, 8)
y_train = np.random.randint(2, size=(100, 1))

# -----
# TRAIN
# -----
model.fit(X_train, y_train, epochs=10, batch_size=10)

# -----
# SAVE MODEL (CORRECT)
# -----
output_dir = 'C:/Users/khaou/OneDrive/Bureau/ddos1/output'
os.makedirs(output_dir, exist_ok=True)

model_path = os.path.join(output_dir, "model.keras")
model.save(model_path)

print("Model saved at:", model_path)
```

Figure 6 : Code d'entraînement d'un modèle de détection binaire sous Keras

Ce code construit un petit réseau de neurones pour la classification binaire (attaque/normal).

Le modèle est un perceptron multicouche avec une couche d'entrée de 12 neurones, une couche cachée de 8 neurones, et une couche de sortie à 1 neurone avec activation sigmoid. Il est compilé avec la perte `binary_crossentropy` et l'optimiseur `adam`

Des données d'entraînement sont générées aléatoirement (100 échantillons, 8 caractéristiques) à titre de test. Le modèle est entraîné pendant 10 époques avec des lots de 10 échantillons.

Enfin, le modèle entraîné est sauvegardé au format. Keras dans un dossier output pour une réutilisation ultérieure dans le pipeline du XDR.

5.5.4 Évaluation du modèle

Le modèle est évalué avec plusieurs métriques :

- accuracy
- précision
- rappel
- F1-score
- matrice de confusion

Les résultats montrent une bonne performance avec une accuracy de 99,89 %.

```
ML_binary_classifier.py X
ddos1 > ML_binary_classifier.py > ...
1 import os
2 import pandas as pd
3 import numpy as np
4
5 from sklearn.preprocessing import QuantileTransformer
6 from sklearn.model_selection import train_test_split
7 from sklearn.metrics import (
8     accuracy_score,
9     confusion_matrix,
10    classification_report,
11    f1_score,
12    recall_score,
13    precision_score
14 )
15
16 from sklearn.utils.class_weight import compute_class_weight
17 from imblearn.over_sampling import SMOTE
18
19 from keras.models import Sequential
20 from keras.layers import Dense, Dropout

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

PS C:\Users\khaou\OneDrive\Bureau\ddos1> python ML_binary_classifier.py

RESULTATS
Accuracy: 0.99957
F1: 0.9997796442535833
Recall: 0.9996157215979833
Precision: 0.9999436206799346

Confusion Matrix:
[[ 4818    11]
 [  75 195096]]
```

Figure 7 : Performances du modèle de classification binaire

La figure présente les performances du modèle de détection d'attaques DDoS évalué sur l'ensemble de test. Les résultats obtenus montrent une **très haute performance globale**, avec une accuracy de **99,95 %**, ce qui indique que le modèle est capable de classifier correctement la grande majorité des flux réseau.

L'analyse de la matrice de confusion montre que le modèle a correctement identifié **4 818 flux BENIGN** et **195 096 attaques (ATTACK)**. Cependant, on observe la présence de **11 faux positifs**, où des flux normaux ont été classés comme attaques, ainsi que **75 faux négatifs**, correspondant à des attaques non détectées.

Ces erreurs restent relativement faibles par rapport au volume total de données, ce qui explique les valeurs élevées de **précision (99,99 %)** et de **rappel (99,96 %)**. Le F1-score de **99,97 %** confirme également un excellent équilibre entre précision et rappel.

Cependant, malgré ces performances élevées, la présence de faux négatifs reste un point critique dans un contexte de cybersécurité, car elle représente des attaques potentiellement non détectées par le système.

En conclusion, le modèle présente des performances très satisfaisantes pour une tâche de détection d'intrusion, mais peut encore être optimisé afin de réduire davantage les faux négatifs.

5.6 Mise en place de l'environnement XDR

5.6.1 Configuration de l'environnement Ubuntu/WSL

Cette section présente la mise en place de l'environnement de travail utilisé pour le développement de la plateforme XDR. Une machine virtuelle Ubuntu, accessible via WSL sous Windows, a été configurée afin de simuler un environnement Linux dédié à la cybersécurité et à l'analyse des flux réseau.

L'environnement déployé repose sur une architecture de sécurité complète intégrant plusieurs outils essentiels. **Elasticsearch** assure le stockage et l'indexation des logs de sécurité, permettant une centralisation efficace des événements et une analyse en temps réel. Les outils **tcpdump** et **Zeek** sont utilisés pour la capture et l'analyse du trafic réseau, où tcpdump récupère les paquets bruts et Zeek les transforme en logs structurés exploitables.

Pour la gestion et la transmission des logs, **Filebeat** est utilisé afin d'automatiser la collecte des données et leur envoi vers Elasticsearch. Enfin, **Kibana** permet la visualisation et la supervision des données à travers des tableaux de bord interactifs facilitant l'analyse des incidents de sécurité.

Dans l'ensemble, cette architecture permet de construire une chaîne complète de collecte, de traitement et de visualisation des événements de sécurité. Elle constitue une base solide pour la détection des attaques DDoS et le développement de la plateforme XDR intelligente.

5.6.2 Déploiement et configuration de la plateforme Wazuh sous Docker

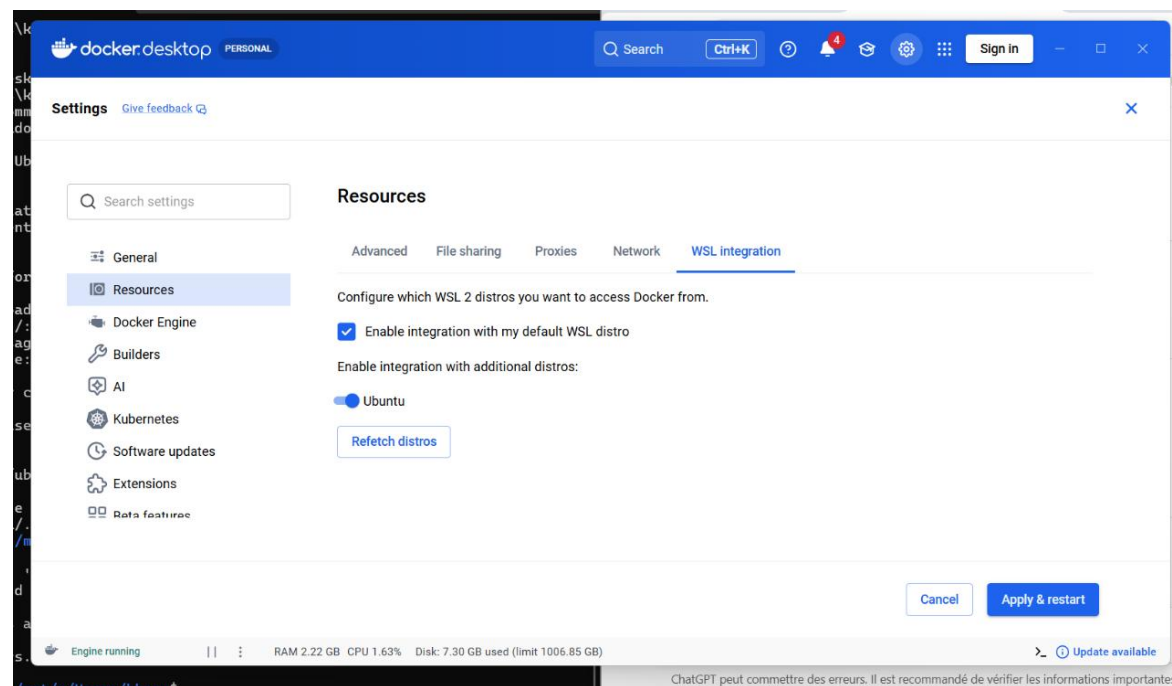


Figure 8 : configuration de l'intégration de Docker Desktop

Cette figure illustre la configuration de l'intégration de Docker Desktop avec l'environnement WSL 2 (Windows Subsystem for Linux). L'option d'intégration avec les distributions Linux supplémentaires a été activée pour Ubuntu, permettant l'exécution des conteneurs Docker directement depuis l'environnement Linux utilisé dans le projet.

```

khaou@ASUS:/mnt/c/Users/khaou$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATE
STATUS        PORTS                    NAMES
dee10cf53137   wazuh/wazuh-dashboard:4.7.5        "/entrypoint.sh"        10 hou
rs ago        Up 10 hours             0.0.0.0:5601->5601/tcp, [::]:5601->5601/tcp  wazuh-d
ashboard
f474bcf2b9c4   wazuh/wazuh-manager:4.7.5          "/init"                  10 hou
rs ago        Up 10 hours             1514-1516/tcp, 514/udp, 55000/tcp          wazuh-m
anager
50cadbb4600e   wazuh/wazuh-indexer:4.7.5          "/entrypoint.sh open..." 10 hou
rs ago        Up 10 hours             0.0.0.0:9200->9200/tcp, [::]:9200->9200/tcp wazuh-i
ndexer
khaou@ASUS:/mnt/c/Users/khaou$

```

Figure 9 : Conteneurs Docker Wazuh opérationnels

Cette figure présente le résultat de la commande `docker ps`, utilisée pour afficher les conteneurs Docker actifs dans l'environnement de test. La capture met en évidence le bon fonctionnement des principaux composants de la plateforme Wazuh déployée dans le cadre de l'étude des solutions XDR open source.

Les conteneurs opérationnels sont les suivants :

- Wazuh-indexer : responsable du stockage et de l'indexation des données de sécurité à l'aide du moteur OpenSearch. Le port 9200 est exposé afin de permettre l'accès aux services d'indexation et de recherche des logs.
- Wazuh-manager : composant central de la plateforme chargé de la collecte, de l'analyse et de la corrélation des événements de sécurité. Il assure également la gestion des alertes générées par les agents et les outils de supervision.
- Wazuh-dashboard : interface web de visualisation permettant la consultation des alertes, des événements réseau et des statistiques de sécurité via le port 5601.
- L'état des conteneurs affiché dans la colonne STATUS confirme leur stabilité opérationnelle, avec une exécution continue durant plusieurs heures sans interruption. Les ports réseau exposés assurent la communication entre les

différents composants de la plateforme ainsi que l'accès aux services de supervision et d'administration.

```
Skipping registering GPU devices...
>>>
>>> print("Model chargé avec succès")from tensorflow.keras.models import load_model
File "<stdin>", line 1
    print("Model chargé avec succès")from tensorflow.keras.models import load_model
                                         ^^^^^
SyntaxError: invalid syntax
>>>
>>> model = load_model("/mnt/c/Users/khaou/OneDrive/Bureau/ddos1/output/model.keras")
>>>
>>> print("Model chargé avec succès")
Model chargé avec succès
>>> import numpy as np
>>> sample = np.random.rand(1, 8)
>>> prediction = model.predict(sample)
1/1 ----- 1s 746ms/step
>>> print(prediction)print(prediction)
File "<stdin>", line 1
    print(prediction)print(prediction)
                        ^^^^^
SyntaxError: invalid syntax
>>> print(prediction)
[[0.50973594]]
>>>
```

Figure 10 : chargement de modèle

Cette figure présente la phase de validation du modèle d'intelligence artificielle après son entraînement. Le modèle sauvegardé a été chargé avec succès à l'aide de TensorFlow/Keras, puis testé sur un échantillon réseau aléatoire contenant plusieurs caractéristiques de trafic.

Le résultat obtenu représente la probabilité qu'un flux réseau soit considéré comme une attaque DDoS. La valeur prédite étant supérieure au seuil défini, le trafic est classé comme potentiellement malveillant.

Cette étape confirme le bon fonctionnement du modèle ainsi que sa capacité à effectuer des prédictions sur de nouvelles données avant son intégration dans la plateforme XDR intelligente.

L'étude de cette architecture a mis en évidence l'intégration de mécanismes d'intelligence artificielle permettant d'améliorer la détection des menaces, ainsi que l'automatisation des processus de réponse aux incidents. Cette approche vise à renforcer la visibilité globale sur le système d'information et à optimiser la réactivité face aux cyberattaques.

Ainsi, ce chapitre constitue une étape essentielle dans la réalisation du projet, en préparant la phase d'implémentation et de tests qui sera détaillée dans le chapitre suivant.

Chapitre 4. Mise en œuvre, tests et résultats

6.1 Introduction

Ce chapitre est consacré à la phase de validation de la plateforme XDR intelligente développée dans ce projet. L'objectif est d'évaluer le comportement du système dans un environnement de test simulé et de mesurer les performances du modèle de détection des attaques DDoS.

Les tests réalisés permettent de vérifier l'ensemble de la chaîne de traitement, depuis la capture du trafic réseau jusqu'à la génération des alertes et l'application des mécanismes de réponse automatique.

Le système est évalué à travers plusieurs scénarios de trafic réseau, incluant du trafic normal et des comportements anormaux simulés, afin d'analyser sa précision, sa rapidité de détection et sa robustesse.

6.2 Scénarios de tests

Dans notre environnement de test, le projet est organisé et exécuté dans un répertoire local situé sur la machine hôte Windows, puis accessible via l'environnement WSL (Windows Subsystem for Linux) dans le chemin

`/mnt/c/Users/khaou/OneDrive/Bureau/DDoS1`. Cette configuration permet d'exécuter le système XDR dans un environnement Linux tout en conservant les fichiers du projet sur Windows, offrant ainsi une architecture hybride adaptée aux tests de cybersécurité. Après la préparation et l'organisation du dossier de travail ainsi que le lancement de l'environnement d'exécution, une simulation de trafic réseau est réalisée à l'aide de l'outil `hping3` afin de générer un flux réseau intensif dans un contexte

contrôlé. La figure ci-dessous présente les résultats de cette exécution, où l'on observe la vérification de la version de l'outil ainsi que le lancement du mode de génération de trafic en flood SYN. Le message affiché indique le fonctionnement normal de l'outil en mode intensif, sans retour de réponses, ce qui permet de simuler un comportement de saturation réseau utilisé pour tester la capacité de détection et de réaction de la plateforme XDR développée.

```
(venv) khaou@ASUS:/mnt/c/Users/khaou/OneDrive/Bureau/ddos1$ hping3 --version
hping3 version 3.0.0-alpha-2 ($Id: release.h,v 1.4 2004/04/09 23:38:56 antirez Exp $)
This binary is TCL scripting capable
(venv) khaou@ASUS:/mnt/c/Users/khaou/OneDrive/Bureau/ddos1$ sudo hping3 --flood -S 127.0.0.1
HPING 127.0.0.1 (lo 127.0.0.1): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Figure 12 : Attaque DDoS

6.3 Évaluation des performances

```
khaou@ASUS:/mnt/c/Users/kl x khaou@ASUS:/mnt/c/Users/l x + - - □ x
2026-05-08 03:24:15.113885 - 🚨 ATTACK DETECTED
2026-05-08 03:24:15.131124 - ✅ BENIGN
2026-05-08 03:24:15.131228 - ✅ BENIGN
2026-05-08 03:24:15.131242 - ✅ BENIGN

📡 Nouveau trafic réseau reçu
7/7 0s 3ms/step
2026-05-08 03:24:17.324461 - 🚨 ATTACK DETECTED
2026-05-08 03:24:17.326343 - ✅ BENIGN
2026-05-08 03:24:17.326454 - ✅ BENIGN
2026-05-08 03:24:17.326478 - ✅ BENIGN
2026-05-08 03:24:17.326487 - ✅ BENIGN
2026-05-08 03:24:17.326497 - 🚨 ATTACK DETECTED
2026-05-08 03:24:17.343998 - 🚨 ATTACK DETECTED
2026-05-08 03:24:17.360666 - 🚨 ATTACK DETECTED
2026-05-08 03:24:17.402273 - ✅ BENIGN
2026-05-08 03:24:17.402447 - 🚨 ATTACK DETECTED

📡 Nouveau trafic réseau reçu
7/7 0s 3ms/step
2026-05-08 03:24:19.585047 - 🚨 ATTACK DETECTED
2026-05-08 03:24:19.586959 - 🚨 ATTACK DETECTED
2026-05-08 03:24:19.606662 - 🚨 ATTACK DETECTED
2026-05-08 03:24:19.616440 - 🚨 ATTACK DETECTED
2026-05-08 03:24:19.625675 - 🚨 ATTACK DETECTED
2026-05-08 03:24:19.669483 - ✅ BENIGN
2026-05-08 03:24:19.669603 - ✅ BENIGN
2026-05-08 03:24:19.669615 - ✅ BENIGN
2026-05-08 03:24:19.669640 - 🚨 ATTACK DETECTED
2026-05-08 03:24:19.680037 - ✅ BENIGN

📡 Nouveau trafic réseau reçu
7/7 0s 3ms/step
2026-05-08 03:24:22.456320 - 🚨 ATTACK DETECTED
2026-05-08 03:24:22.914168 - ✅ BENIGN
2026-05-08 03:24:22.914301 - ✅ BENIGN
2026-05-08 03:24:22.914320 - ✅ BENIGN
2026-05-08 03:24:22.914329 - ✅ BENIGN
2026-05-08 03:24:22.914337 - ✅ BENIGN
2026-05-08 03:24:22.914344 - 🚨 ATTACK DETECTED
2026-05-08 03:24:22.935781 - 🚨 ATTACK DETECTED
```

Figure 13 : Détection des attaques en temps réel

La figure représente les résultats en temps réel générés par la plateforme XDR intelligente lors de l'analyse du trafic réseau. Chaque ligne correspond à un flux réseau traité par le modèle d'intelligence artificielle, avec une classification automatique en deux catégories : **BENIGN** (trafic normal) et **ATTACK DETECTED** (trafic suspect).

On observe que le système alterne entre des flux normaux et des flux malveillants détectés, ce qui montre que le modèle est capable de différencier efficacement les comportements réseau en fonction des caractéristiques extraites.

Les horodatages affichés indiquent que chaque flux est traité de manière quasi instantanée, avec un temps moyen d'environ **3 ms par étape**, ce qui confirme la capacité du système à fonctionner en temps réel.

La présence répétée de messages **ATTACK DETECTED** montre que le modèle identifie correctement les anomalies dans le trafic simulé. À l'inverse, les prédictions **BENIGN** correspondent à des flux considérés comme normaux par le système.

Dans l'ensemble, ces résultats confirment :

- la capacité du modèle à effectuer une classification binaire efficace
- la détection en temps réel des comportements suspects
- la stabilité du pipeline de traitement (capture → analyse → prédiction)

Ainsi, la plateforme XDR démontre une bonne réactivité et une efficacité satisfaisante dans la détection des attaques DDoS simulées.

```

PS C:\Users\khaou> cd C:\Users\khaou\OneDrive\Bureau\ddos1
PS C:\Users\khaou\OneDrive\Bureau\ddos1> cat alerts.log
2026-05-08 03:21:15.412074 - 0Y$" ATTACK DETECTED
2026-05-08 03:21:17.524996 - 0Y$" ATTACK DETECTED
2026-05-08 03:21:19.596780 - 0Y$" ATTACK DETECTED
2026-05-08 03:21:25.831501 - 0Y$" ATTACK DETECTED
2026-05-08 03:21:52.981850 - 0Y$" ATTACK DETECTED
2026-05-08 03:22:01.308106 - 0Y$" ATTACK DETECTED
2026-05-08 03:22:17.948585 - 0Y$" ATTACK DETECTED
2026-05-08 03:22:26.271597 - 0Y$" ATTACK DETECTED
2026-05-08 03:22:28.350937 - 0Y$" ATTACK DETECTED
2026-05-08 03:22:36.660857 - 0Y$" ATTACK DETECTED
2026-05-08 03:22:36.665797 - 0Y$" ATTACK DETECTED
2026-05-08 03:22:40.835414 - 0Y$" ATTACK DETECTED
2026-05-08 03:22:47.068259 - 0Y$" ATTACK DETECTED
2026-05-08 03:23:03.691284 - 0Y$" ATTACK DETECTED
2026-05-08 03:23:14.107416 - 0Y$" ATTACK DETECTED
2026-05-08 03:23:16.197468 - 0Y$" ATTACK DETECTED
2026-05-08 03:23:16.201744 - 0Y$" ATTACK DETECTED
2026-05-08 03:23:18.308524 - 0Y$" ATTACK DETECTED
2026-05-08 03:23:22.494295 - 0Y$" ATTACK DETECTED
2026-05-08 03:23:22.497778 - 0Y$" ATTACK DETECTED
2026-05-08 03:23:32.965517 - 0Y$" ATTACK DETECTED
2026-05-08 03:23:32.970012 - 0Y$" ATTACK DETECTED
2026-05-08 03:23:32.979975 - 0Y$" ATTACK DETECTED
2026-05-08 03:23:35.078494 - 0Y$" ATTACK DETECTED
2026-05-08 03:23:35.081832 - 0Y$" ATTACK DETECTED
2026-05-08 03:23:52.021602 - 0Y$" ATTACK DETECTED
2026-05-08 03:24:01.664720 - 0Y$" ATTACK DETECTED
2026-05-08 03:24:05.998918 - 0Y$" ATTACK DETECTED
2026-05-08 03:24:08.237395 - 0Y$" ATTACK DETECTED
2026-05-08 03:24:08.255660 - 0Y$" ATTACK DETECTED
2026-05-08 03:24:08.265568 - 0Y$" ATTACK DETECTED
2026-05-08 03:24:08.277791 - 0Y$" ATTACK DETECTED
2026-05-08 03:24:08.289378 - 0Y$" ATTACK DETECTED
2026-05-08 03:24:08.303741 - 0Y$" ATTACK DETECTED
2026-05-08 03:24:08.508752 - 0Y$" ATTACK DETECTED
2026-05-08 03:24:08.623156 - 0Y$" ATTACK DETECTED
2026-05-08 03:24:08.639318 - 0Y$" ATTACK DETECTED

```

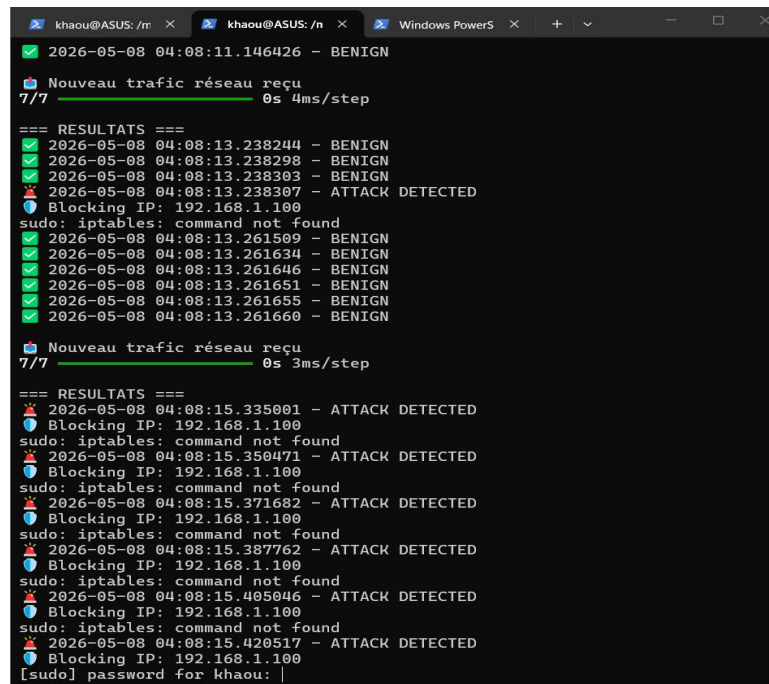
Figure 14 : Journalisation des alertes de sécurité

Cette figure présente le contenu du fichier alerts.log, généré par la plateforme XDR lors de la détection d'activités suspectes. Chaque ligne du journal correspond à une alerte de sécurité contenant un horodatage précis et l'indication ATTACK DETECTED.

Les événements enregistrés montrent une détection continue et répétée d'attaques, ce qui confirme le fonctionnement du système de surveillance en temps réel. Ce fichier de logs joue un rôle essentiel dans la traçabilité des incidents, l'analyse post-attaque et la corrélation des événements de sécurité.

Cette journalisation constitue une base importante pour le module de supervision du système XDR, permettant le suivi historique des attaques et l'aide à la prise de décision en matière de réponse aux incidents.

6.4 Mécanisme de défense



```
khaou@ASUS: /m x khaou@ASUS: /n x Windows PowerS x + v - □ x
2026-05-08 04:08:11.146426 - BENIGN
Nouveau trafic réseau reçu
7/7 0s 4ms/step

=== RESULTATS ===
2026-05-08 04:08:13.238244 - BENIGN
2026-05-08 04:08:13.238298 - BENIGN
2026-05-08 04:08:13.238303 - BENIGN
2026-05-08 04:08:13.238307 - ATTACK DETECTED
Blocking IP: 192.168.1.100
sudo: iptables: command not found
2026-05-08 04:08:13.261509 - BENIGN
2026-05-08 04:08:13.261634 - BENIGN
2026-05-08 04:08:13.261646 - BENIGN
2026-05-08 04:08:13.261651 - BENIGN
2026-05-08 04:08:13.261655 - BENIGN
2026-05-08 04:08:13.261660 - BENIGN
Nouveau trafic réseau reçu
7/7 0s 3ms/step

=== RESULTATS ===
2026-05-08 04:08:15.335001 - ATTACK DETECTED
Blocking IP: 192.168.1.100
sudo: iptables: command not found
2026-05-08 04:08:15.350471 - ATTACK DETECTED
Blocking IP: 192.168.1.100
sudo: iptables: command not found
2026-05-08 04:08:15.371682 - ATTACK DETECTED
Blocking IP: 192.168.1.100
sudo: iptables: command not found
2026-05-08 04:08:15.387762 - ATTACK DETECTED
Blocking IP: 192.168.1.100
sudo: iptables: command not found
2026-05-08 04:08:15.405046 - ATTACK DETECTED
Blocking IP: 192.168.1.100
sudo: iptables: command not found
2026-05-08 04:08:15.420517 - ATTACK DETECTED
Blocking IP: 192.168.1.100
[sudo] password for khaou: |
```

Figure 15 : Détection d'attaque et réponse automatique

Cette figure illustre le fonctionnement de la plateforme XDR en mode de détection et de réponse en temps réel. Le système analyse de manière continue le trafic réseau et classe chaque flux en deux catégories : **BENIGN** pour le trafic légitime et **ATTACK DETECTED** pour les comportements suspects, avec une faible latence d'environ 3 à 4 millisecondes par traitement. Lorsqu'un flux malveillant est identifié, la plateforme déclenche automatiquement un mécanisme de réponse visant à limiter l'impact de l'attaque, notamment par le blocage de l'adresse IP source via des règles de filtrage au niveau du pare-feu (iptables). Cette étape représente le module de

réponse automatique intégré dans le pipeline XDR, assurant une réaction rapide sans intervention humaine. Cette capture met ainsi en évidence deux aspects essentiels : la capacité du modèle à assurer une détection en temps réel des anomalies réseau, ainsi que l'importance de la configuration correcte des mécanismes de défense afin de garantir une réponse efficace et automatisée face aux menaces détectées.

6.5 Dashboard et visualisation

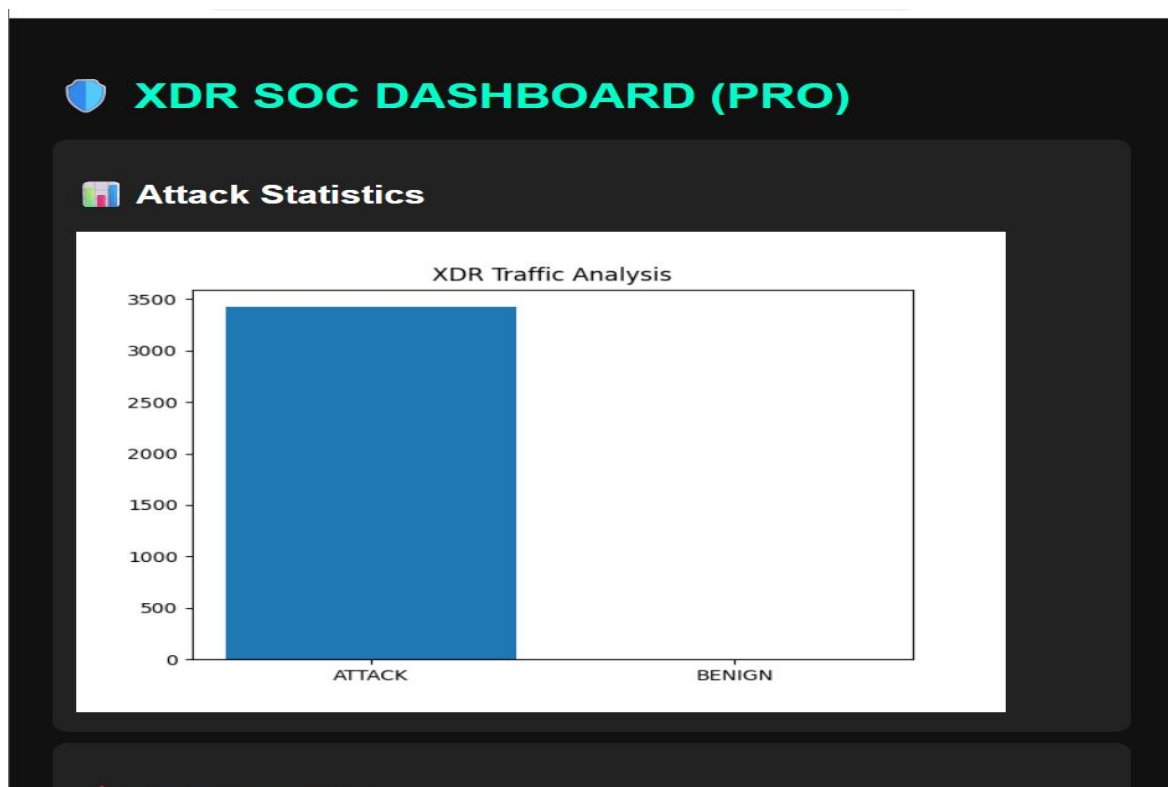


Figure 16 : Tableau de bord " XDR SOC DASHBOARD "

La figure présente le tableau de bord "XDR SOC DASHBOARD (PRO)" développé à l'aide de Kibana. Le graphique "XDR Traffic Analysis" permet de comparer les volumes de trafic **ATTACK** (malveillant) et **BENIGN** (légitime) détectés par le modèle d'intelligence artificielle. Cette interface offre une visualisation en temps réel de l'état du réseau, facilitant la supervision par les analystes SOC. Elle permet

notamment d'identifier rapidement les pics d'activité malveillante et d'évaluer l'efficacité globale du système de détection mis en place.

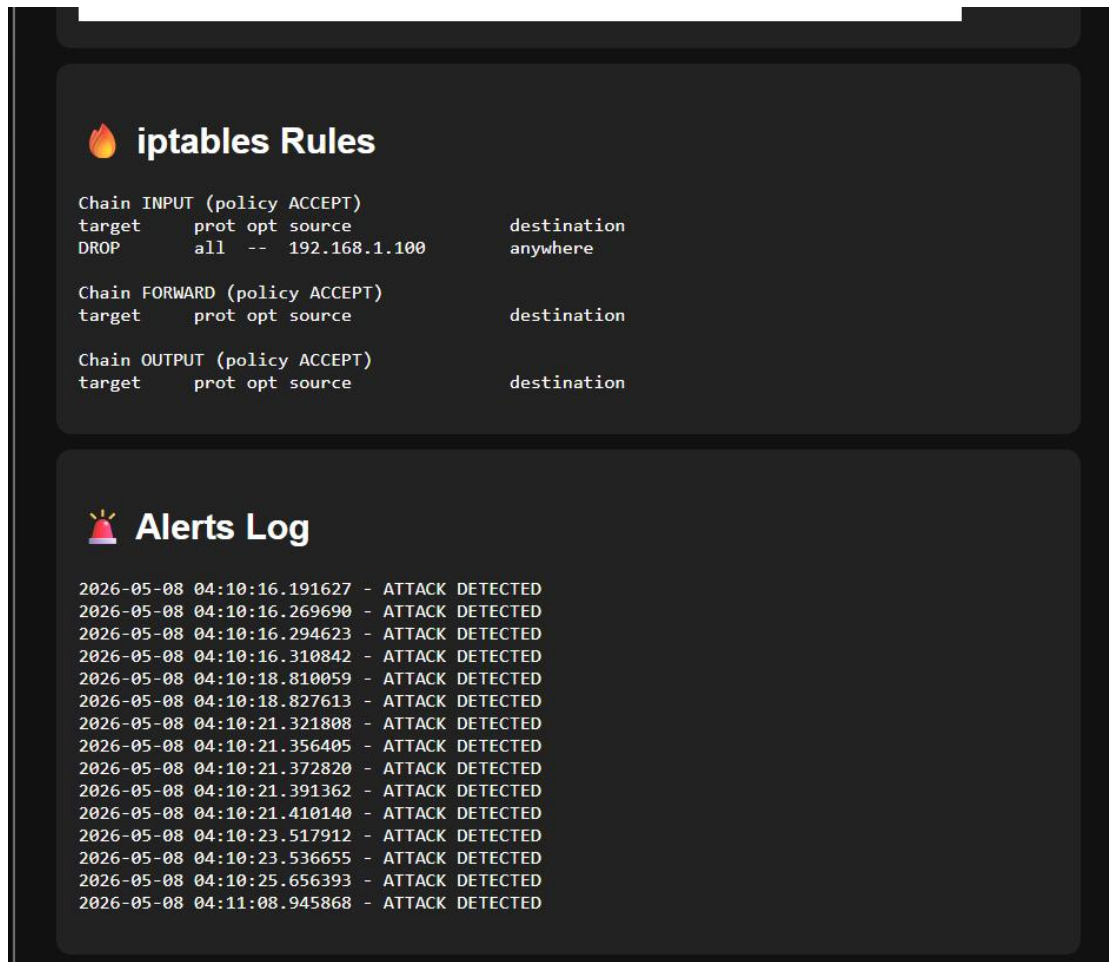


Figure 17 : Règles iptables et journal des alertes

La figure illustre deux composants essentiels de notre plateforme XDR : le mécanisme de blocage réseau (iptables) et le journal des détections (alerts.log).

6.5.1 Règles iptables

La commande iptables -L permet d'afficher les règles de filtrage actives sur le système. Dans le cadre de la plateforme XDR, une règle de type **DROP** est appliquée afin de bloquer le trafic provenant d'une adresse IP identifiée comme malveillante. Ce

mécanisme est déclenché automatiquement par le système lorsqu'une activité suspecte est détectée, permettant ainsi une mitigation en temps réel sans intervention humaine. Cette approche renforce la capacité du système à réagir rapidement face aux attaques réseau.

6.5.2 Journal des alertes

Le fichier alerts.log constitue le journal central des événements de sécurité générés par la plateforme. Il enregistre de manière chronologique les détections effectuées par le modèle d'intelligence artificielle, avec un horodatage précis pour chaque événement.

Les entrées du journal indiquent la succession des alertes générées en fonction du trafic analysé, ce qui permet de suivre l'évolution des comportements réseau en temps réel. Chaque occurrence de l'événement **ATTACK DETECTED** correspond à une prédiction du modèle de classification intégré dans le système.

6.5.3 Statistique

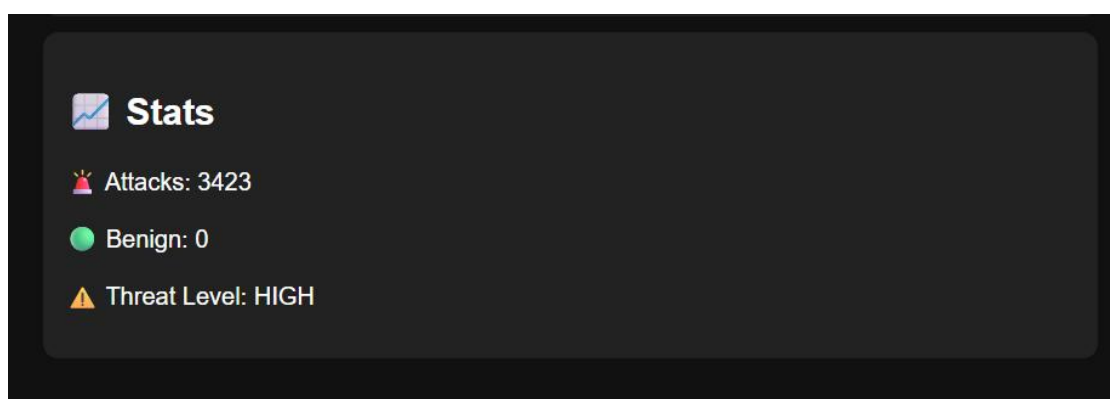


Figure 18 : Statistiques du dashboard XDR

La figure présente les principaux indicateurs du tableau de bord de la plateforme XDR.

On y observe notamment le nombre total d'attaques détectées (**Attacks : 3423**), ainsi

que le trafic légitime (**Benign : 0**), ce qui s'explique par la nature du scénario de test basé principalement sur des flux malveillants simulés. Le niveau de menace (**Threat Level : HIGH**) est automatiquement généré par le système en fonction du volume élevé d'activités suspectes détectées.

Ces résultats montrent la capacité de la plateforme à quantifier et à qualifier le niveau de risque en temps réel. Le système fournit ainsi une vision globale de l'état de sécurité du réseau et permet d'alerter immédiatement l'analyste en cas de forte activité malveillante.

Par ailleurs, le tableau de bord permet la visualisation centralisée des alertes de sécurité, des statistiques de trafic et des événements enregistrés par la plateforme XDR. Cette interface améliore la supervision en temps réel et facilite l'analyse des incidents de sécurité.

6.6 Conclusion

Ce chapitre a permis de valider la plateforme XDR développée à travers une série de tests fonctionnels réalisés dans un environnement contrôlé. Les scénarios de simulation, notamment ceux basés sur des flux de type SYN Flood, ont mis en évidence la capacité du système à détecter en temps réel les comportements réseau anormaux.

Les résultats obtenus confirment la bonne performance du modèle d'intelligence artificielle ainsi que le bon fonctionnement de la chaîne complète de traitement, allant de la capture des flux réseau jusqu'à la génération des alertes et leur visualisation dans le tableau de bord.

Par ailleurs, l'intégration des mécanismes de réponse et la centralisation des journaux via la stack Elastic ont renforcé la visibilité et la fiabilité du système. Ainsi, la plateforme XDR développée répond aux objectifs fixés en matière de détection, d'analyse et de réponse aux incidents de sécurité réseau.

Conclusion générale

Ce projet a permis de concevoir et de mettre en œuvre une plateforme XDR intelligente dédiée à la détection des attaques DDoS en temps réel. L'objectif principal était de proposer une solution capable de collecter, analyser et corréler les flux réseau afin d'identifier automatiquement les comportements malveillants et de renforcer la sécurité des systèmes d'information.

Pour atteindre cet objectif, une architecture complète a été mise en place en intégrant des outils de capture et d'analyse du trafic réseau tels que Zeek et tcpdump, ainsi que des solutions de centralisation et de visualisation des logs comme Elasticsearch et Kibana. Cette architecture a été complétée par un modèle d'intelligence artificielle basé sur TensorFlow/Keras, permettant la classification des flux réseau en trafic normal et trafic malveillant.

Les expérimentations réalisées dans un environnement contrôlé ont montré que la plateforme est capable de détecter efficacement les attaques simulées, de générer des alertes en temps réel et de déclencher des mécanismes de réponse automatique. Les résultats obtenus confirment la bonne performance du modèle ainsi que la fiabilité de la chaîne de traitement implémentée.

Ainsi, la solution développée répond aux objectifs fixés en assurant la détection, la supervision et la réponse aux incidents de sécurité. Ce travail constitue un prototype fonctionnel d'une approche XDR orientée intelligence artificielle pour la cybersécurité.

Perspective

Comme perspective, il serait intéressant d'étendre la plateforme vers une architecture XDR plus complète intégrant plusieurs sources de données de sécurité telles que les endpoints, les serveurs et les services cloud. L'intégration de techniques avancées de deep learning ainsi que l'utilisation de données temps réel permettraient également d'améliorer davantage les performances de détection et la réduction des faux positifs.

Webographie

- [13] <https://www.coursera.org/fr-FR/articles/what-is-artificial-intelligence> Consulté le : 17/05/2026
- [15] <https://www.fortinet.com/fr/resources/cyberglossary/artificial-intelligence-in-cybersecurity> Consulté le : 10/05/2026
- [6] <https://www.cyber-securite.fr/quest-ce-que-sont-les-ids-et-comment-ils-protagent-votre-systeme/> Consulté le : 05/05/2026
- [7] <https://www.microsoft.com/fr-fr/security/business/security-101/what-is-siem> Consulté le : 01/05/2026
- [8] <https://www.crowdstrike.com/fr-fr/cybersecurity-101/endpoint-security/endpoint-detection-and-response-edr/> Consulté le : 04/05/2026
- [1] https://www.researchgate.net/publication/51988470_Distributed_Denial_of_Service_DDOS_Attacks_Detection_Mechanism Consulté le : 25/04/2026
- [4] [https://www.cloudflare.com/fr-fr/the-net/small-DDoS-attacks/\(20/04/2026\)](https://www.cloudflare.com/fr-fr/the-net/small-DDoS-attacks/(20/04/2026))
- [2] <https://www.ibm.com/topics/zero-day> Consulté le : 20/04/2026
- [3] <https://www.cloudflare.com/fr-fr/learning/DDoS/what-is-a-DDoS-attack/> Consulté le : 19/04/2026
- [5] <https://www.fortinet.com/fr/resources/cyberglossary> Consulté le : 18/04/2026
- [9] <https://www.fortinet.com/fr/resources/cyberglossary> Consulté le : 17/04/2026
- [10] <https://www.fortinet.com/fr/resources/cyberglossary/intrusion-detection-system-ids> Consulté le : 15/04/2026

[11] <https://www.microsoft.com/fr-fr/security/business/security-101/what-is-siem>

Consulté le : [10/04/2026](#)

[12] <https://www.crowdstrike.com/fr-fr/cybersecurity-101/endpoint->

[security/endpoint-detection-and-response-edr/](#)Consulté le : 09/04/2026

[\[17\]](#)

https://www.researchgate.net/publication/336953914_Developing_Realistic_Distributed_Denial_of_Service_DDoS_Attack_Dataset_and_Taxonomy Consulté le : 03/04/2026