الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoire de Master

Filière : Télécommunication
Spécialité : Réseaux et Télécommunication

Présenté par

BOUGHABA AKRAM

&

GHRIBI AOUATIF

Détection Et Prévention Des Attaques DDOS Avec Le Système IDPS Suricata

Proposé par : M. Bensebti

Année Universitaire 2023-2024

Remerciements

Avant tout, nous tenons à remercier ***Allah*** *Tout Puissant de nous avoir donné la force, la santé, la connaissance, la capacité et l'opportunité d'entreprendre cette étude de recherche et de persévérer pour la compléter de manière satisfaisante.*

Nous tenons en tout premier lieu à exprimer notre profond respect et nos sincères remerciements à notre encadrant, ***Monsieur Bensebti****, pour sa contribution, sa confiance et son soutien indéfectible. Sa présence et ses conseils ont été essentiels à la réussite de ce travail.*

Nous souhaitons également exprimer notre gratitude à ***Monsieur Mehdi Marouan*** *pour son aide pratique et son précieux soutien tout au long de la réalisation de notre étude.*

Nos remerciements vont également à ***Madame Benachour Lina*** *pour son soutien et ses précieux conseils ses recommandations, son aide pratique tout au long de notre recherche.*

Enfin, nous tenons à exprimer notre profonde reconnaissance envers tous nos enseignants du département d'électronique de ***l'Université de Blida 1****, dont le dévouement exceptionnel envers notre éducation a été une source constante d'inspiration et a grandement enrichi notre parcours académique.*

Dédicaces

Du fond de mon cœur, je dédie ce travail à tous ceux qui me sont chers :

A mon cher Père, mon précieux don du dieu, celui qui me soutient sans limites, et qui me donne sans récompense, le premier soutien dans ma carrière, qui est à mes côtés chaque jour.

A ma chère mère, Quoi que je fasse ou que je dise, je ne saurai point te remercier comme il se doit. Ton affection me couvre, ta bienveillance me guide et ta présence à mes côtés à mes côtés a toujours été ma source de force pour affronter les différents obstacles., j'espère que votre bénédiction m'accompagne toujours.

A Ma sœur et Mes frères pour leurs soutiens et pour l'amour J'espère que nos liens fraternels seront renforcés et dureront plus longtemps.

Sans oublier mon binôme AKRAM pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.

A tous mes camarades, mes connaissances et a tous ceux qui ont une relation de proche ou de loin avec la réalisation de ce travail.

GHRIBI AOUATIF.

Dédicaces

Du fond de mon cœur, je dédie ce travail :

A mes chers parents, qui ont toujours été mon pilier de soutien et ma source d'inspiration. Votre amour inconditionnel, vos sacrifices et votre encouragement constant m'ont permis de réaliser ce projet. Merci pour votre patience et votre croyance en mes capacités.

A mon frère, pour sa camaraderie, ses conseils avisés et son soutien indéfectible. Ta présence à mes côtés m'a souvent donné la force de persévérer.

A toute ma famille, pour leur amour, leur soutien et leurs encouragements constants tout au long de ce parcours académique. Vous avez tous contribué, chacun à votre manière, à la réalisation de ce mémoire. Avec toute ma gratitude et mon affection.

BOUGHABA AKRAM.

ملخص :تأمين الشبكة يتضمن تقليل جميع المخاطر المحتملة. تستعرض هذه الأطروحة اكتشاف ومنع هجمات DDOS باستخدام Suricata، وهو نظام لمنع الاختراق. يقدم المفاهيم المتعلقة بأمن المعلومات وهجماتDDOS ، ثم يوضح إعداد بيئة اختبار تشمل تثبيت Suricataومحاكاة الهجمات لتقييم أدائه. تُظهر النتائج أن Suricata يكتشف ويمنع هجمات DDOS بكفاءة في الوقت الفعلي. أخيرًا، يتم تقديم توصيات لتعزيز أمان الشبكة، مثل تحديث قواعد الاكتشاف.

كلمات المفاتيح: الحرمان من الخدمة الموزعة(DDOS) ، نظام(Suricata) ، الأمن السيبراني، لوقاية من التطفل(IPS) ، محاكاة الهجمات، حماية الشبكة.

Résumé : La sécurisation d'un réseau implique de minimiser tous les risques envisageables. Ce mémoire examine la détection et la prévention des attaques DDOS à l'aide de Suricata, un système de prévention d'intrusion. Il présente les concepts de sécurité informatique et des attaques DDOS, puis détaille la mise en place d'un environnement de test, incluant l'installation de Suricata et la simulation d'attaques pour évaluer ses performances. Les résultats montrent que Suricata détecte et bloque efficacement les attaques DDOS en temps réel. Enfin, des recommandations sont fournies pour renforcer la sécurité réseau, telles que la mise à jour des règles de détection.

Mots clés : DDOS, IPS, Suricata, sécurité informatique, simulation d’attaques, protection réseau.

Abstract: Securing a network involves minimizing all conceivable risks. This thesis examines the detection and prevention of DDOS attacks using Suricata, an intrusion prevention system. It presents the concepts of cybersecurity and DDOS attacks, then details the setup of a test environment, including the installation of Suricata and the simulation of attacks to evaluate its performance. The results show that Suricata effectively detects and blocks DDOS attacks in real-time. Finally, recommendations are provided to strengthen network security, such as updating detection rules.

Keywords: DDOS, IPS, Suricata, cybersecurity, attack simulation, network protection.

Listes des acronymes et abréviations

ARP : Address Resolution Protocol**.**

CPU: Central Processing Unit.

DOS: Denial of Service.

DDOS: Distributed Denial of Service.

DNS: Domain Name System.

Gbps: Gigabits per second.

HIPS: Host-based Intrusion Prevention System**.**

HIDS: Host-based Intrusion Detection System.

HULK: HTTP Unbearable Load King.

HTTP: HyperText Transfer Protocol.

ICMP: Internet Control Message Protocol**.**

IDS: Intrusion Detection System.

IDPS: Intrusion Detection and Prevention System.

IPS: Intrusion Prevention System.

IP: Internet Protocol.

LAN: Local Area Network.

LOIC: Low Orbit Ion Cannon.

NIPS: Network-based Intrusion Prevention System.

NIDS: Network-based Intrusion Detection System.

OISF: Open Information Security Foundation.

PA: Push Acknowledgement.

PC: Personal Computer.

PCAP: Packet Capture.

PPS: Packet per second.

RJ45: Registered Jack 45.

RPS: Request per second.

SSH: Secure Shell.

SYN: Synchronize.

TCP: Transmission Control Protocol.

TOR: The Onion Router.

UDP: User Datagram Protocol.

VPN: Virtual Private Network.

Table des matières

Liste des Figures

Liste des tableaux

Introduction générale

Le développement rapide de la technologie et l'expansion des réseaux ont certes créé de nouvelles opportunités, mais ont également exposé les systèmes informatiques à des menaces de plus en plus complexes, parmi lesquelles les attaques par déni de service distribué (DDOS) se révèlent particulièrement préoccupantes. Ces attaques, en saturant les services en ligne avec un flux massif de trafic, rendent les ressources indisponibles pour les utilisateurs légitimes. L'augmentation spectaculaire de ces attaques met en évidence l'urgence de développer des méthodes efficaces de détection et de réponse. À titre d'exemple, l'attaque DDOS de septembre 2017, qui a ciblé les services de Google avec un volume de trafic atteignant 2,54 Tb/s, a révélé la vulnérabilité des infrastructures critiques face à de telles menaces. Cet incident a ainsi motivé l'étude présentée dans ce mémoire [1].

Dans le cadre du projet de fin d'étude, nous allons concevoir un système capable de détecter et de bloquer les attaques DDOS de manière efficace. Notre travail est structuré comme suit :

Le premier chapitre offre une vue d'ensemble de l'état de l'art en matière de sécurité informatique, ainsi qu'une revue de la littérature sur les attaques DOS/DDOS et les solutions de détection disponibles, avec un focus sur l'IPS Suricata.

Le deuxième chapitre se concentre sur la conception et l'implémentation de l'environnement de test, en abordant l'architecture, les choix logiciels et la configuration des outils.

Enfin, le troisième chapitre présente les résultats des tests et des expérimentations, analyse les performances de l'IPS Suricata et propose des recommandations pour l'amélioration des mécanismes de détection des attaques DOS/DDOS.

Chapitre 1 État de l'art

1.1 Introduction

Dans un paysage où la sécurité informatique revêt une importance cruciale pour les entreprises, les gouvernements et les particuliers, les attaques informatiques sont devenues un problème majeur. Parmi ces attaques, certaines se démarquent par leur dangerosité, notamment les attaques par déni de service (DOS) et les attaques par déni de service distribué (DDOS).

Dans ce chapitre, nous examinons l'état actuel de la sécurité informatique, en mettant l'accent sur ces deux types d'attaques. Nous examinons également les moyens de protection contre ces attaques et soulignons l'importance des systèmes de détection et de prévention des intrusions pour la sécurité des réseaux informatiques. Enfin, nous présenterons les logiciels de détection d'intrusion, open source qui jouent un rôle crucial dans la protection contre les attaques informatiques.

1.2 Sécurité Informatique

La sécurité informatique est un aspect crucial des technologies numériques modernes, englobant différentes mesures et techniques pour préserver les systèmes informatiques et les données des attaques internes et externes [2].

1.2.1 Objectifs de la sécurité informatique

Les principes fondamentaux de la sécurité informatique reposent sur la garantie de la confidentialité, de l'intégrité, de la disponibilité des données, de l'authenticité et de la non-répudiation [3].

- **Confidentialité**

La confidentialité des données est fondamentale pour la sécurité, limitant l'accès aux données sensibles aux seules personnes autorisées. Cela est garantie grâce à l'emploi d'un algorithme de chiffrement spécifique, ce qui rend les données transmises incompréhensibles pour les personnes non autorisées [3].

- **Intégrité**

L'intégrité des données est cruciale pour assurer la véracité et la fiabilité des Informations stockées. Cela signifie que les données ne peuvent être altérées, supprimées ou modifiées de Manière non autorisée [3].

- **Authenticité**

L'authentification des données est une mesure de sécurité essentielle pour prévenir l'accès non autorisé aux informations confidentielles.

Il s'agit de vérifier l'identité d'un utilisateur ou d'un système avant de lui offrir l'accès aux données [3].

- **Disponibilité**

La disponibilité des systèmes d'information vise à assurer leur fonctionnement continu et leur accessibilité constante aux utilisateurs autorisés [3].

- **Non-répudiation**

La non-répudiation assure qu'une fois qu'une transaction a été effectuée, elle Ne peut être niée par l'une des parties impliquées. En d'autres termes, chaque partie est tenue Responsable de ses actions et ne peut pas rejeter sa participation à la transaction [3].

1.3 Attaques par Déni De Service DOS/DDOS

Les attaques par déni de service (DOS) et par déni de service distribué (DDOS) visent à perturber le fonctionnement normal des systèmes informatiques en les submergeant de demandes. Elles peuvent cibler divers systèmes, comme des sites Web, des services en ligne, et des infrastructures critiques.

Il existe deux principales façons de mener une attaque par déni de service [4] :

- **Inondation de trafic :** Il s'agit de l'envoi d'une quantité massive de données à un réseau, un ordinateur ou une application, à une vitesse que le système ne peut pas gérer.
- **Paquets malveillants :** consiste à envoyer intentionnellement des paquets de données mal formés à un destinataire, comme un ordinateur ou une application, ce qui le rend incapable de les traiter correctement.

1.3.1 Déni de Service (DOS)

Une attaque DOS (Denial of Service) consiste en ce qu'un seul attaquant inonde le système cible avec un trafic excessif, comme un flux continu de paquets de données, empêchant ainsi les utilisateurs légitimes d'accéder aux ressources du système [5].

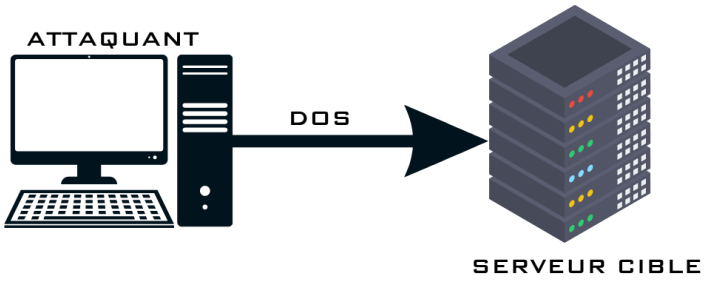

Figure 1.1 : Inondation de requêtes DOS.

1.3.2 Déni De Service Distribue (DDOS)

Une attaque DDOS (Distributed Denial of Service) est une forme plus complexe et puissante d'une attaque DOS. L'attaquant utilise un réseau d'ordinateurs compromis, appelé botnet, pour inonder la cible de trafic. Étant donné que l'attaque provient de multiples sources, il est beaucoup plus difficile de s'en défendre [5].

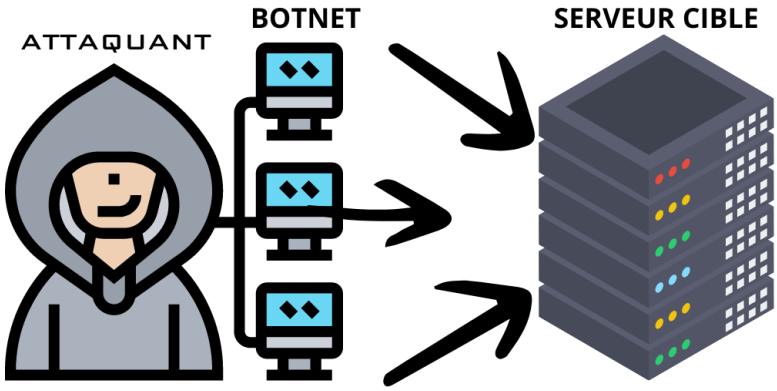

Figure 1.2 : Inondation de requêtes DDOS.

1.4 Principales catégories d'attaques DOS/ DDOS

Les attaques DOS/DDOS peuvent être classées de différentes manières, selon l'aspect de l'attaque considéré. Voici une typologie de quelques classifications courantes :

1.4.1 Attaque volumétrique

Elles visent à submerger la cible en lui envoyant une quantité massive de données, mesurée en bits ou en gigabits par seconde (Gbps).

Cela inclut des attaques telles que les attaques par inondation UDP et les attaques par inondation ICMP [5].

1.4.2 Attaque de protocole

Elles exploitent les faiblesses des protocoles réseau pour générer un grand nombre de demandes de connexion, mesuré en paquets par seconde (PPS).

Cela inclut par exemple les attaques par inondation SYN et les attaques Ping de la mort [5].

1.4.3 Attaque de la couche application

Elles ciblent des vulnérabilités spécifiques dans des applications ou des sites Web, mesurées en requêtes par seconde (RPS).

Elles peuvent être plus difficiles à détecter car elles peuvent ressembler à un trafic légitime, comme par exemple saturant les ressources d'un serveur Web via des requêtes HTTP [5].

1.5 Types d’attaques DOS/DDOS

Ces attaques peuvent prendre différentes formes et utilisent diverses techniques telles que :

Tableau 1.1 : Types d'attaques DDOS et leurs descriptions [6][7].

Nom de l’attaque	**Explication du principe de l’attaque**
UDP FLOOD	Submerge la bande passante du serveur avec des paquets UDP volumineux sans nécessité de préétablissement de connexion.
SYN FLOOD	Exploite le processus de poignée de main TCP pour épuiser les ressources du serveur.
ICMP FLOOD	Envoie des requêtes Ping ICMP ininterrompues pour surcharger le système.
AMPLIFICATION DNS	Exploite les serveurs DNS pour amplifier le trafic vers la cible.
HTTP GET/POST	Envoie un grand nombre de requêtes HTTP vers la cible pour surcharger le serveur Web. Les navigateurs envoient des requêtes GET pour récupérer du contenu statique et des requêtes POST pour des interactions dynamiques avec le serveur.
ATTAQUE PING OF DEATH	Envoie des paquets ICMP malformés pour planter les systèmes cibles.
SMURF	Une attaque ICMP en broadcast consiste à usurper l'adresse source pour rediriger les multiples réponses vers la victime.
LAND	Envoie des paquets SYN avec les adresses source et destination configurées à l'adresse de la cible, perturbant le serveur et consommant des ressources.
ATTAQUE ARP SPOOFING	Empoisonne le cache ARP des réseaux pour rediriger le trafic vers un attaquant.

1.6 Outils d'attaque DOS/DDOS

Un outil d'attaque par déni de service distribué est un logiciel ou un script spécialement conçu pour lancer des attaques DDOS.

Tableau 1.2 : Présente des outils DOS/DDOS, leurs protocoles ciblés et catégories d'attaque.

Logiciel	Protocol	Catégorie D'Attack	Référence
LOIC (Low Orbit Ion Cannon)	UDP/TCP/ HTTP	1.Attaques volumétriques (UDP). 2. Attaques de la couche d'application (HTTP). 3. Attaques de protocole (TCP).	**[7]**
HPING3	ICMP/UDP/ TCP	1.Attaques volumétriques (UDP/ICMP). 2. Attaques de protocole (TCP).	**[7]**
SLOWLORIS	HTTP	Attaques de la couche d'application (HTTP).	**[7]**
HULK	UDP	Attaques volumétriques (UDP).	**[8]**
TORSHAMMER	HTTP	Attaques de la couche d'application (HTTP).	**[9]**
SLOWHTTPTEST	HTTP	Attaques de la couche d'application (HTTP).	**[10]**
GOLDENEYE	HTTP	Attaques de la couche d'application (HTTP).	**[11]**

1.7 Méthodes de protection

La variété et la disponibilité des outils d'attaques augmentent le risque des intrusions. Par conséquent les administrateurs s'appuient sur diverses solutions dans le but de maintenir la protection du réseau informatique**.**

1.7.1 Un antivirus

Un logiciel conçu et une barrière de protection pour prévenir, détecter et éliminer les virus informatiques ainsi que d'autres programmes malveillants.

Il surveille les comportements des programmes et analyse les données afin de repérer tout signe de menace.

1.7.2 Pare-feu

Un pare-feu est un système de sécurité, qu'il soit matériel ou logiciel, qui surveille et contrôle le flux de données entrant et sortant d'un réseau. Il utilise des règles définies pour bloquer les accès non autorisés et filtrer le trafic.

1.7.3 Réseau privé virtuel (VPN)

Une technologie de réseau qui crée un tunnel sécurisé et chiffré entre deux réseaux physiques ou entre un appareil connecté à Internet et un réseau privé, Les données échangées à travers le VPN sont cryptées, garantissant ainsi leur confidentialité et leur intégrité [11].

1.7.4 Système de détection d'intrusions (IDS)

Un Système de Détection d'Intrusion (IDS) est un dispositif ou logiciel passif qui surveille le trafic réseau pour identifier des activités suspectes ou anormales. Il analyse le trafic pour repérer des menaces potentielles et alerte les administrateurs en cas de détection de comportements non autorisés. Les IDS sont déployés de manière à observer le trafic sans l'interférer directement [12].

Il existe plusieurs catégories principales de systèmes IDS [13] :

1. **NIDS :** Analyser le trafic sur l'ensemble du réseau à partir d'un point spécifique pour détecter les activités suspectes sur tous les terminaux.
2. **HIDS :** Examiner le trafic en provenance ou à destination de terminaux individuels dans le réseau, en ignorant les autres terminaux.

1.7.5 Système de prévention d'intrusion (IPS)

Un Système de Prévention d'Intrusion (IPS) est un dispositif ou logiciel actif qui détecte et prévient les menaces sur un réseau en temps réel. Contrairement aux IDS, les IPS sont déployés en ligne avec les flux réseau, permettant d'intervenir directement pour bloquer ou filtrer le trafic malveillant. Ils utilisent des mécanismes de détection similaires aux IDS mais peuvent appliquer des mesures correctives pour empêcher les attaques de réussir [12].

Il existe principaux types d'IPS [13] :

1. **NIPS :** surveille et sécurise le trafic au sein de l'ensemble du réseau.
2. **HIPS :** surveille et analyse les événements se produisant sur un hôte spécifique.

1.8 Logiciels IDS/IPS pour la détection des attaques DOS/DDOS

Les logiciels de détection d'attaques jouent un rôle crucial dans l'identification et la neutralisation des menaces avant qu'elles n'occasionnent des dommages considérables. Parmi les solutions : Suricata, Snort et Zeek se distinguent par leurs fonctionnalités avancées et leur flexibilité d'utilisation.

- **Suricata :** un moteur open source spécialisé dans la détection des menaces, offrant des fonctionnalités avancées telles que la détection d'intrusions (IDS), la prévention d'intrusions (IPS) et la surveillance de la sécurité réseau. Grâce à son architecture multithread, il est capable d'effectuer une analyse parallèle efficace. Les capacités de détection de Suricata couvrent à la fois les signatures spécifiques et les anomalies, prenant en charge une gamme étendue de protocoles réseau tels que IPv4, IPv6, TCP, UDP et ICMP [14].
- **Snort :** Est un système de détection et de prévention d'intrusions open-source utilisé pour surveiller et analyser le trafic réseau en temps réel. Il utilise des techniques de détection de signatures et d'analyse protocolaire pour identifier les comportements suspects et les attaques potentielles [14].
- **Zeek :** Zeek est un logiciel open source conçu pour l'analyse passive du trafic réseau. Il fonctionne comme un "capteur" sur une plateforme matérielle ou logicielle, observant discrètement le trafic réseau sans l'interrompre. Il analyse les données qu'il observe pour générer des journaux de transactions détaillés, des informations sur le contenu des fichiers, et des sorties personnalisées [14].

1.9 Règle Suricata

Suricata intègre un ensemble de règles simples, légères, flexibles et puissantes permettant de détecter les violations de politique et les comportements malveillants, Elles sont divisées en deux sections [15] :

- **Entête de la règle :** Elle contient l'action de la règle, le protocole, les adresses IP, les ports et la direction du trafic ciblé par la règle.
    - Action : C'est la partie qui nous informe quoi faire quand Suricata trouve un paquet qui correspond aux critères de la règle. Il existe quatre actions par défaut dans Suricata : Alerte, Passer, Drop et Rejeter.
    - Protocole : Définit les protocoles auxquels les règles seront appliquées.

- Adresses IP : Définit les IP ou les plages d'IP pour correspondre à l'hôte source ou de destination.
- Ports : Ce sont les interfaces d'entrée/sortie sur lesquelles sont définis les ports source ou de destination, ou les plages de ports.
- Direction du trafic : Désigne la direction dans laquelle la règle doit être évaluée.

- **Options de la règle :** qui contient les messages d'alerte et les informations sur la partie du paquet qui doivent être inspectées pour déterminer si l'action de la règle doit être acceptée.
  - Msg : Informations textuelles envoyées lorsque l'alerte se déclenche.
  - Sid : Métadonnées de l'ID de la signature.
  - Rev : Métadonnées de révision de la signature.
  - Flow : Correspond aux connexions établies vers le serveur.
  - Content : Contenu pour que la signature corresponde.

Exemple : nous présentons un modèle basique de règle IDS.

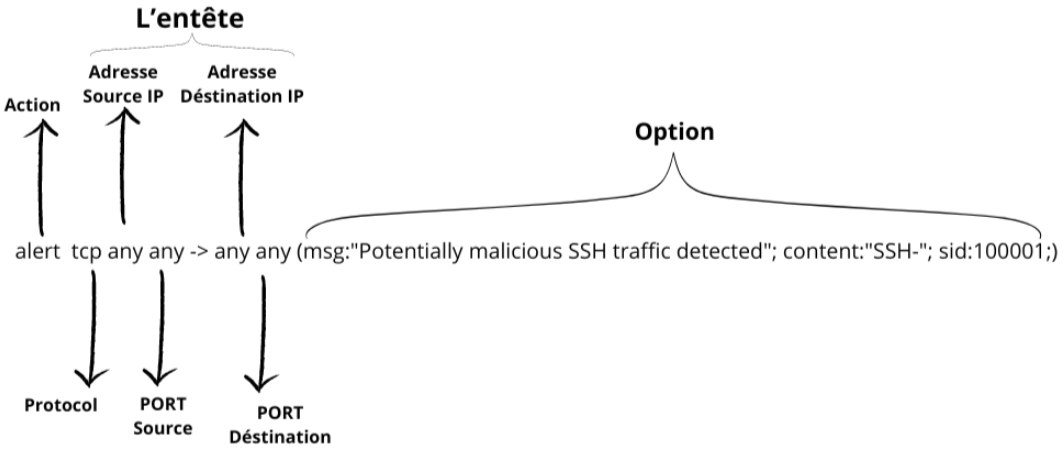

Figure 1.3 : Format d'une règle de détection d'intrusion SSH.

1.10 Conclusion

Dans ce chapitre, nous avons constaté que les attaques par déni de service distribué peuvent entraîner des dommages considérables, soulignant ainsi l'importance cruciale de la préparation et de la coopération en matière de cyber sécurité. En renforçant les défenses contre de telles attaques et en développant des stratégies de réponse adaptées, les organisations peuvent mieux se protéger et réduire l'impact de ces incidents sur leurs activités et leurs utilisateurs. Les prochains chapitres exploreront en détail chacun de ces aspects.

Chapitre 2 Conception de l'environnement de travail

2.1 Introduction

Les organisations sont de plus en plus préoccupées par la sécurité de leurs systèmes informatiques, en raison de l'augmentation des attaques par déni de service distribué qui peuvent perturber les infrastructures informatiques. La création d'environnements simulés est essentielle pour mieux comprendre ces attaques et s'en protéger, ce qui permet de tester les défenses et de développer des contre-mesures efficaces.

Ce chapitre explique comment mettre en place un tel environnement, en détaillant chaque étape De la sélection de matériel et de logiciels jusqu'aux scénarios d'attaque. En adoptant cette approche, les organisations peuvent évaluer leur capacité à résister aux attaques DDoS et à renforcer leur sécurité.

2.2 Environnement de travail

Dans notre simulation d'attaque DDOS dans un environnement LAN réel, notre architecture comprend trois PC interconnectés via des câbles RJ45 à un routeur. Deux de ces PC fonctionnent en tant qu'attaquant, tandis que le troisième agit soit en tant que victime.

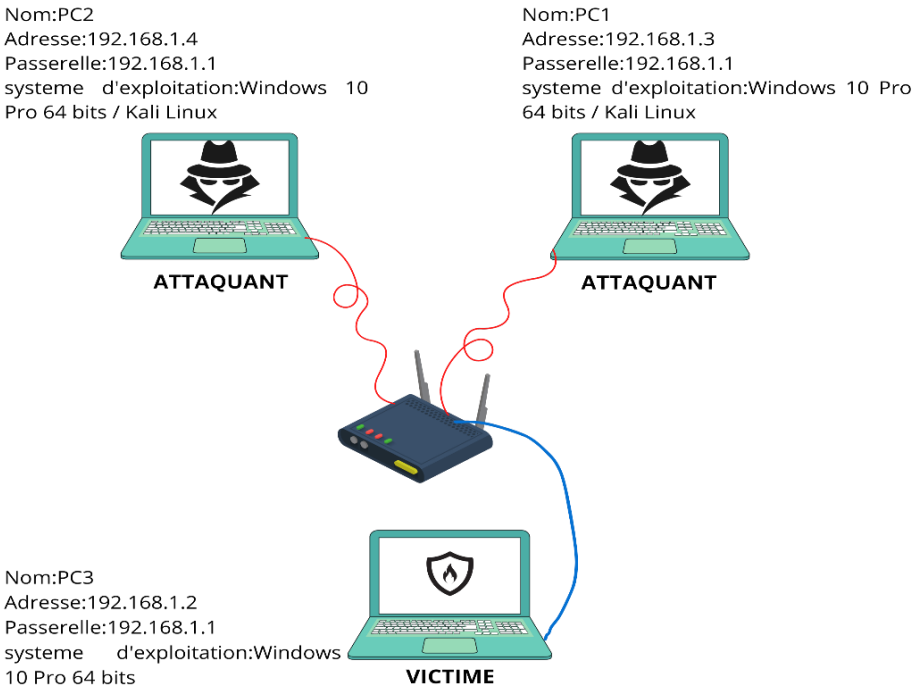

Figure 2.1 : Architecture du réseau.

Tableau 2.1: Caractéristique des équipements de travail.

Appareil	PC1 (HP)	PC2 (DELL)	PC3 (HP)
Type	Attaquant	Attaquant	Victime
Système d'exploitation	Windows 10 Pro 64 bits Kali Linux	Windows 10 Pro 64 bits Kali Linux	Windows 10 Pro 64 bits
Ram	8GB	8GB	8GB
Rom	256 GB SSD	256 GB SSD	256 GB SSD
Processeur	I7 6th Gen	I3 7th Gen	I5 11 Gen
Adresse IP	192.168.1.3	192.168.1.4	192.168.1.2

2.3 Environnement logiciel

Il est crucial d'installer divers logiciels et outils pour simuler une attaque entre la victime et l'attaquant, garantissant ainsi le bon déroulement de l'attaque et de sa détection.

Chaque machine, qu'elle soit du côté de l'attaquant ou la victime, nécessite la mise en place de logiciels et d'outils spécifiques.

2.3.1 Attaquant

Pour l'attaquant, les outils nécessaires sont :

- **Outils de simulation d'attaque** : Nous avons fait le choix de sept programmes spécifiques : HULK, GOLDENEYE, HPING3, LOIC, SLOWHTTPTEST, SLOW LORIS et TORSHAMMER.

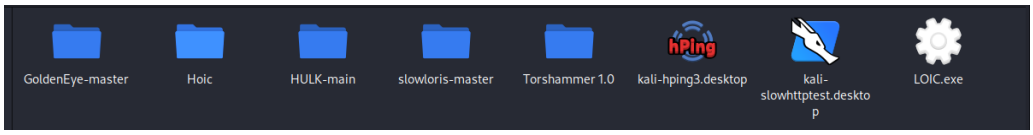

Figure 2.2 : Outils de Simulation d'Attaques.

- **Python** : L'attaquant doit installer Python sur ses machines. Il peut télécharger les versions Python 2 et 3, car certains outils d'attaque utilisent Python 2, par exemple TorsHammer, tandis que d'autres requièrent Python 3.

2.3.2 Victime

Le système de la victime requiert divers logiciels et outils pour détecter et prévenir les attaques.

- Suricata : principal outil de détection des attaques.
- Npcap : fournit la capture de paquets réseau nécessaire.

- Notepad++ : facilite la configuration des règles de détection d'intrusion et de Suricata.yaml.
- Wireshark : offre une analyse détaillée du trafic.

2.4 Test la connectivité entre les machines

Avant de procéder aux simulations complètes d'attaques, des vérifications initiales sont effectuées pour garantir la connectivité entre les PC.

- **PC1 (Attaquant) :**

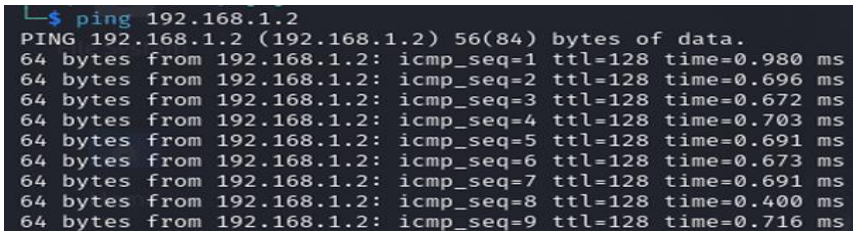

Figure 2.3 : Ping entre PC1 et PC3.

- **PC2 (Attaquant) :**

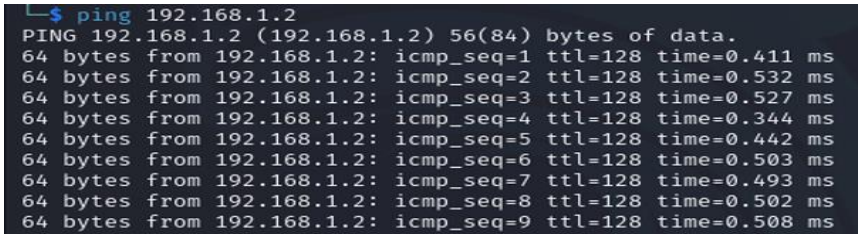

Figure 2.4 : Ping entre PC2 et PC3.

- **PC3 (Victime) :**

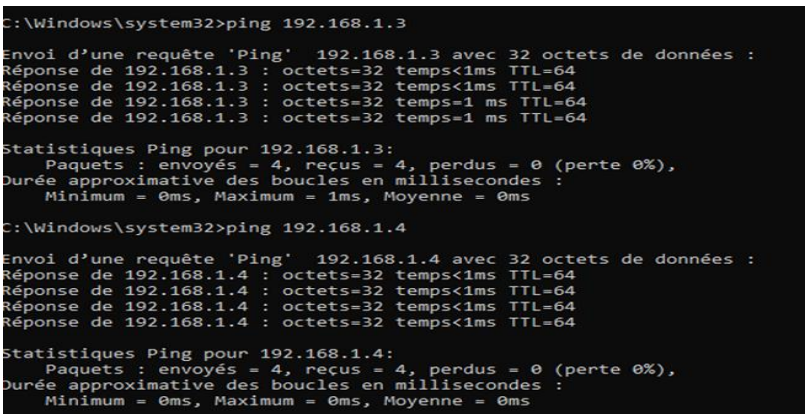

Figure 2.5 : Ping entre PC3 et PC1,2.

2.5 Simulation des attaques

Nous procédons à une analyse détaillée des différents types d'attaques, incluant TCP, UDP, ICMP et HTTP, ainsi que de leur implémentation à travers les outils sélectionnés.

Ensuite, nous conduisons une étude approfondie des conséquences de chaque type d'attaque sur les ressources du système et du réseau.

2.5.1 Attaque Inondation TCP

a. HPING3

L'utilisation de HPING3 déclenche une attaque SYN flood ciblant l'adresse IP 192.168.1.2 sur le port 80 à partir de deux ordinateurs attaquants compromis. Les paquets sont envoyés avec les mêmes adresses source et destination vers la machine cible.

La commande: **hping3 –syn --flood -p 80 192.168.1.2 -a 192.168.1.2**

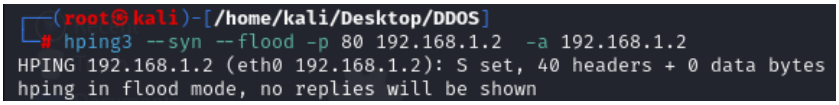

Figure 2.6 : Attaque Inondation TCP avec HPING3.

- **--flood :** Envoie les paquets aussi rapidement que possible sans attendre de réponse.
- **-a :** Spécifie l'adresse IP source des paquets envoyés.
- **--syn :** Indique que HPING3 enverra des paquets avec le flag SYN activé.
- **-p 80 :** Spécifie le port de destination des paquets.

➢ **Analyse du trafic**

Durant l'attaque, nous avons employé l'outil Wireshark sur le PC cible afin d'observer et d'analyser le trafic réseau engendré.

Au cours de l'analyse de trafic, nous remarquons que la cible reçoit multiples demandes de connexion TCP indiquées par le flag SYN=1.

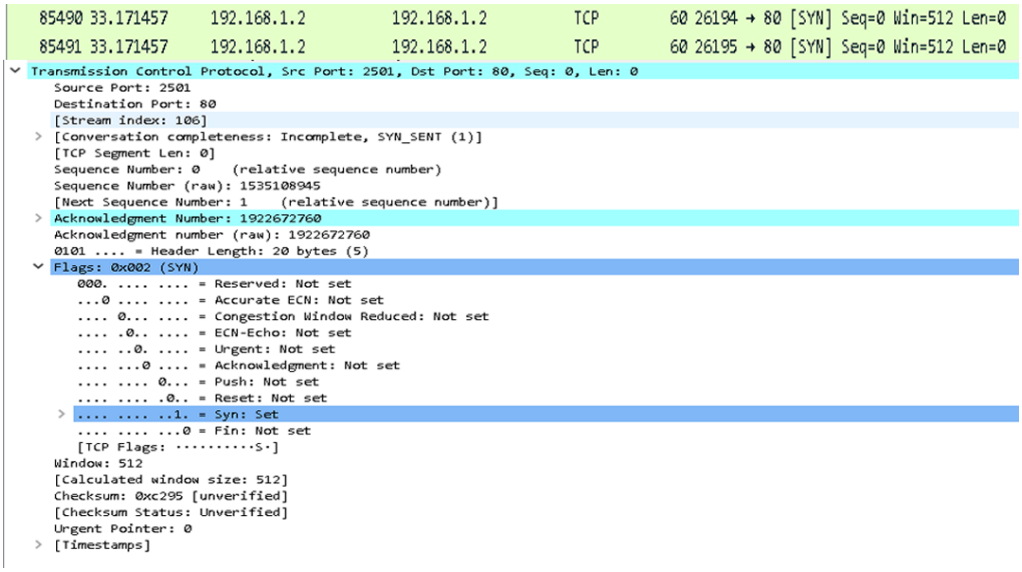

Figure 2.7 : Analyse des paquets d'attaque Inondation TCP_HPING3.

Dans le graphique représentant le nombre de paquets par seconde, nous avons remarqué que la quantité de paquets TCP s'élevait à environ 57500 par seconde pendant l'attaque. Par la suite, nous avons constaté une diminution constante jusqu'à la terminaison de l'attaque.

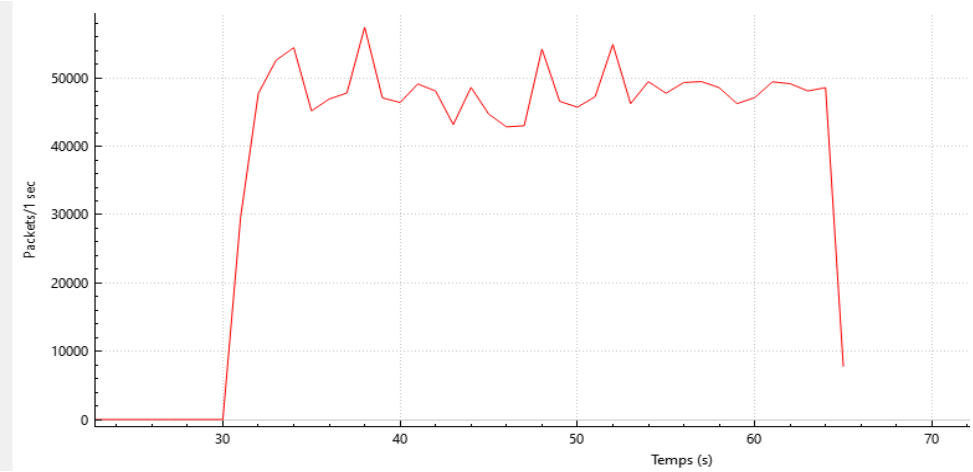

Figure 2.8 : Nombre de Paquets/s Inondation TCP _HPING3.

b. LOIC (Low Orbit Ion Cannon)

Pour chaque PC attaquant, exécutez l'outil, entrez l'adresse IP cible (192.168.1.2), sélectionnez l'attaque TCP et le port 80, puis lancez l'attaque.

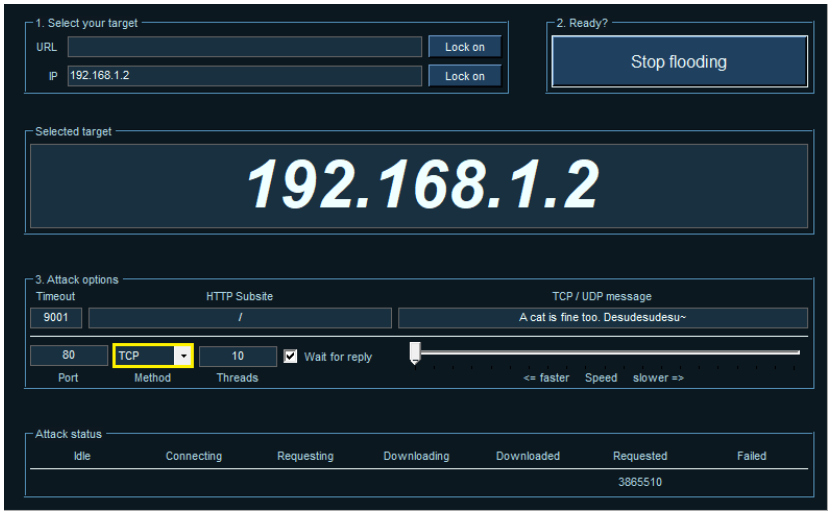

Figure 2.9 : Attaque Inondation TCP_LOIC.

➢ **Analyse du trafic**

Lors de l'analyse du trafic avec Wireshark, nous avons observé que la cible reçoit un maximum de 17 500 paquets TCP par seconde. Ces paquets, identifiés par les indicateurs PUSH et ACK, contiennent les données spécifiques "desudesudesu".

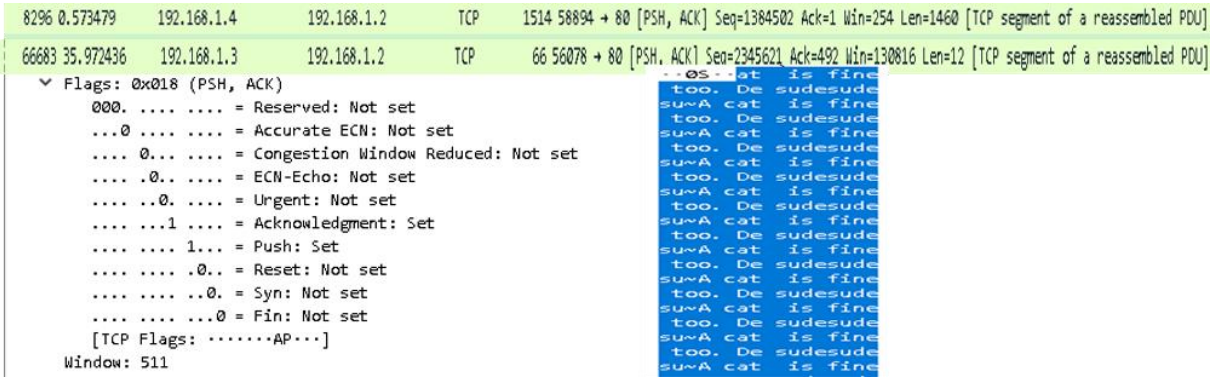

Figure 2.10 : Analyse des paquets d'attaque Inondation TCP _LOIC.

c. TORSHAMMER

À partir de cet outil, nous lançons une attaque TCP en envoyant un grand nombre de requêtes à l'adresse IP cible 192.168.1.2 sur le port 80.

La commande **: python2 torshammer-py -t 192.168.1.2 -p 80 -r 5000000**

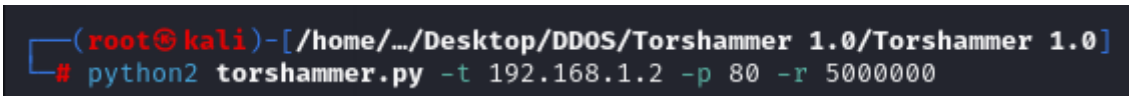

Figure 2.11 : Attaque Inondation TCP _TORSHAMMER.

- **Python2 torshammer-py :** Utilise le script Python torshammer-py pour lancer l'attaque
  - **-r 5000000 :** Cette option spécifie le nombre de requêtes par thread.
  - **-t 192.168.1.2 :** Cette option spécifie l'adresse IP du serveur cible à attaquer.
  - **-p 80 :** Cette option spécifie le port du serveur cible à attaquer.

➢ **Analyse du trafic**

Au cours de notre analyse, nous avons observé que le nombre de paquets TCP atteignait environ 8 000 par seconde, avec les indicateurs PUSH et ACK. Ces paquets transportaient des données spécifiques de type "application/x-www-form-urlencoded" et provenaient des adresses IP des attaquants (192.168.1.3 et 192.168.1.4).

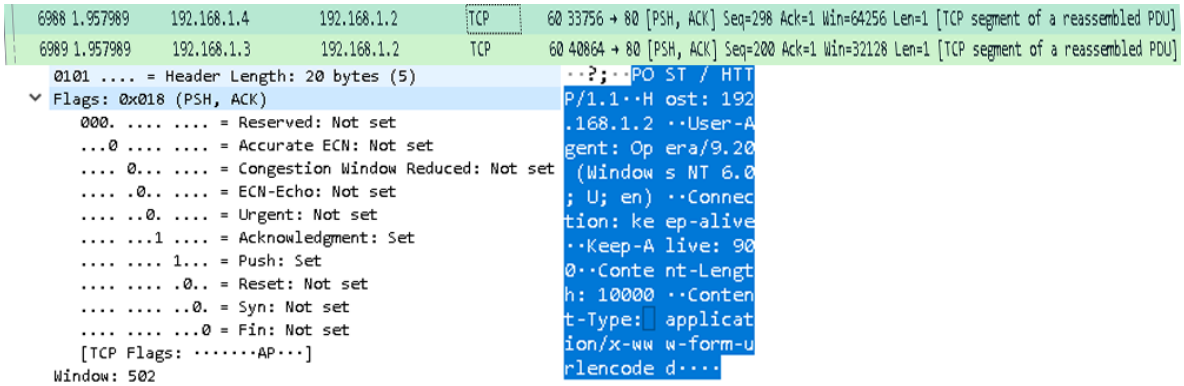

Figure 2.12 : Analyse des paquets Inondation TCP _TORSHAMMER.

d. SLOW LORIS

À l'aide de ce logiciel, nous avons utilisé la commande suivante pour cibler l'adresse 192.168.1.2 : **python slowloris.py -p 80 -s 10000 --sleeptime 0 192.168.1.2**

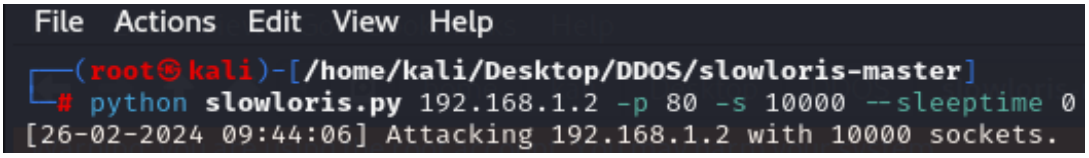

Figure 2.13 : Attaque Inondation TCP _SLOW LORIS.

- **-p 80 :** Cette option spécifie le port du serveur cible à attaquer.
- **-s 10000 :** Cette option spécifie le nombre de sockets à utiliser pour l'attaque.
- **--sleeptime 0 :** Cette option spécifie le temps de sommeil entre chaque envoi de données.

➢ **Analyse du trafic**

Au cours de notre analyse, le nombre de paquets par seconde indiquait environ 9500 paquets TCP avec les indicateurs PA. Ces paquets contenaient des données spécifiques telles que "GET /?" et "HTTP/1.1", avec des longueurs respectives de 21 et 20 octets. Ils incluaient également des informations utilisateur telles que "User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36", "Accept-language: en-US,en;q=0", ainsi que la chaîne "X-a:". Ces paquets étaient émis à partir des adresses IP des attaquants, à savoir 192.168.1.3 et 192.168.1.4.

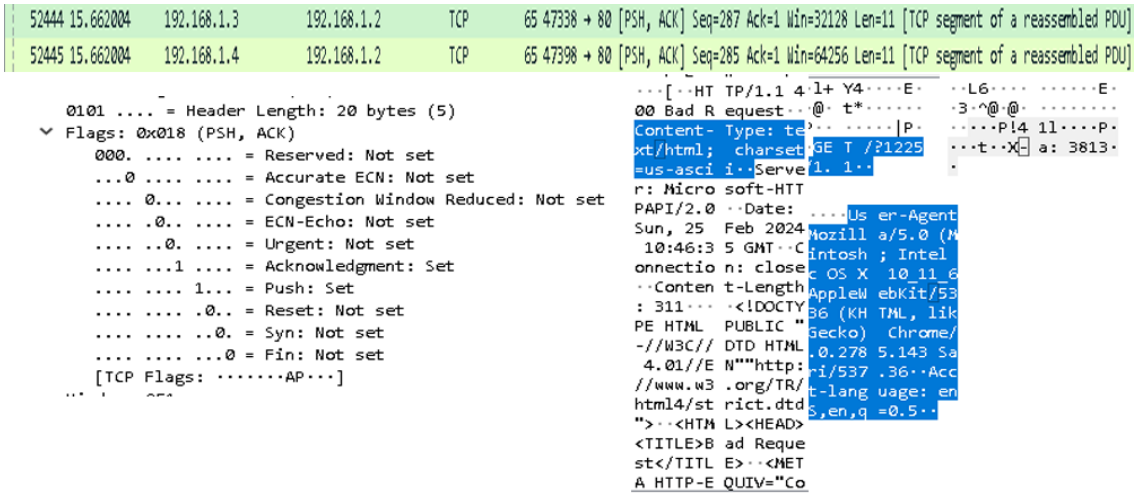

Figure 2.14 : Analyse des paquets Inondation TCP_SLOWLORIS.

➢ **Impact des attaques TCP**

En l'absence d'attaque, le système fonctionne à un niveau d'activité minimal, avec une utilisation du CPU de seulement 2% et une bande passante du réseau quasi inexistante, mesurée à 0,008 Mbps.

- Attaque TORSHAMMER : Présente une légère augmentation de l'utilisation du CPU à 7%, mais son impact le plus significatif est sur la bande passante du réseau, atteignant 1 Mbps. Cette attaque vise à épuiser les ressources de la connexion réseau en envoyant un grand nombre de requêtes TCP de manière répétitive.
- Attaque HPING3 Land : A un impact plus important sur la bande passante du réseau, mesurée à 35,5 Mbps, tandis que l'utilisation du CPU atteint 20%.
- Attaque SLOWLORIS : Épuise les ressources du serveur en maintenant des connexions HTTP ouvertes le plus longtemps possible, ce qui entraîne une utilisation élevée du CPU à 23% et une augmentation notable de la bande passante du réseau, atteignant 48 Mbps.

À partir de ces résultats, nous pouvons conclure que l'effet des attaques diffère considérablement en ce qui concerne l'utilisation du CPU et la bande passante du réseau.

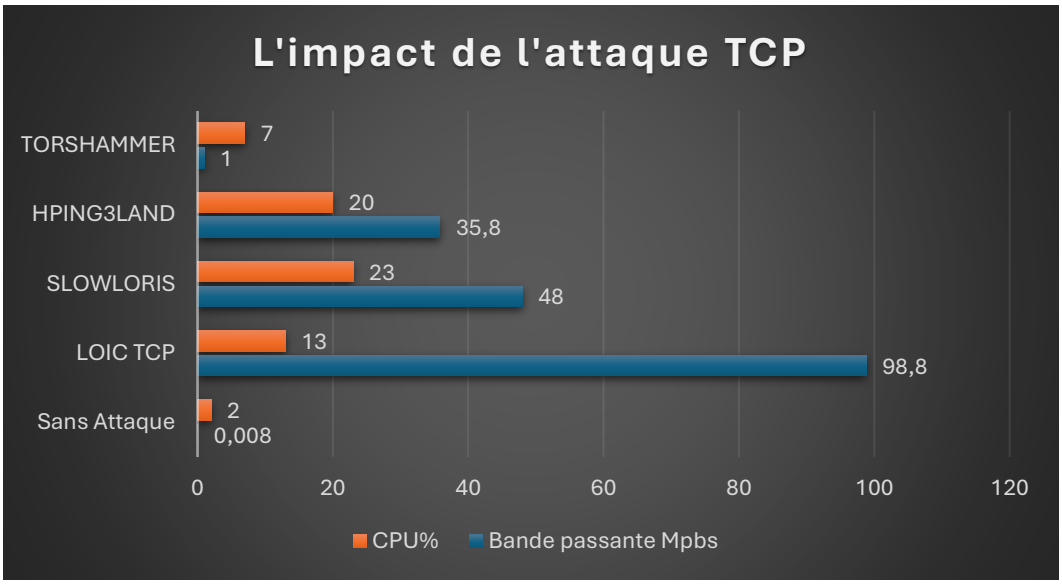

Figure 2.15 : Impact d'une attaque Inondation TCP sur les performances d'un serveur.

2.5.2 Attaque Inondation UDP

Nous avons exécuté une attaque UDP en utilisant les outils HULK et LOIC.

a. HULK

Lorsque l'outil HULK est en cours d'exécution, il suffit d'ajouter l'adresse cible, 192.168.1.2.

```
[+] HULK is attacking server 192.168.1.2

[+] Successfully sent 1 packet to 192.168.1.2 throught port:80
```

Figure 2.16 : Attaque Inondation UDP_HULK.

➢ **Analyse du trafic**

Au cours de l'analyse, une accumulation significative de paquets UDP a été identifiée, provenant des adresses IP des attaquants 192.168.1.3 et 192.168.1.4. Ces paquets transportaient des données spécifiques, caractérisées par une taille de 1490 octets et un protocole IP de type 17 (UDP). Un pic a été observé dans le nombre de paquets par seconde, oscillant entre environ 7500 et 7900 paquets UDP par seconde pendant l'attaque, suivi d'une diminution constante jusqu'à la terminaison de l'attaque.

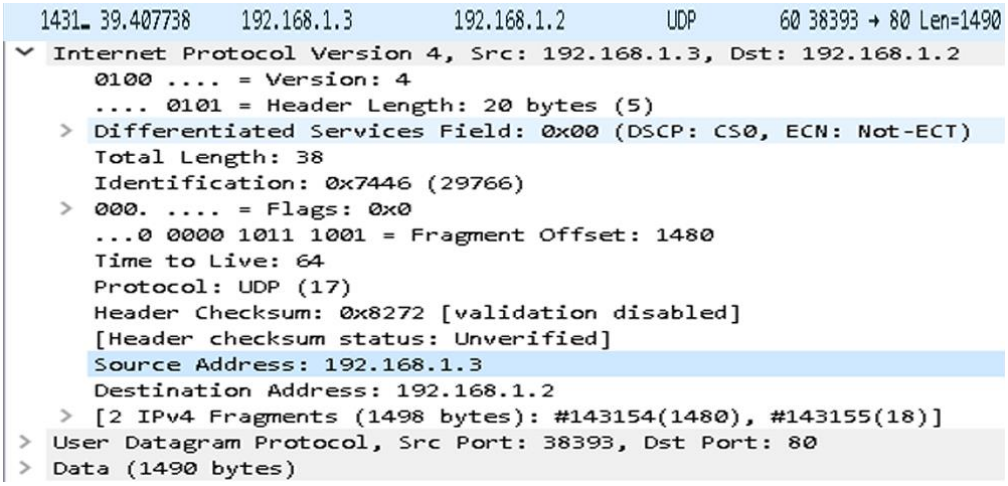

Figure 2.17 : Analyse des paquets Inondation UDP_HULK.

b. LOIC

Nous allons suivre la même procédure que celle utilisée lors de la visualisation des paquets TCP pour obtenir la signature de l'attaque UDP. Pour ce faire, nous allons sélectionner le type d'attaque comme UDP.

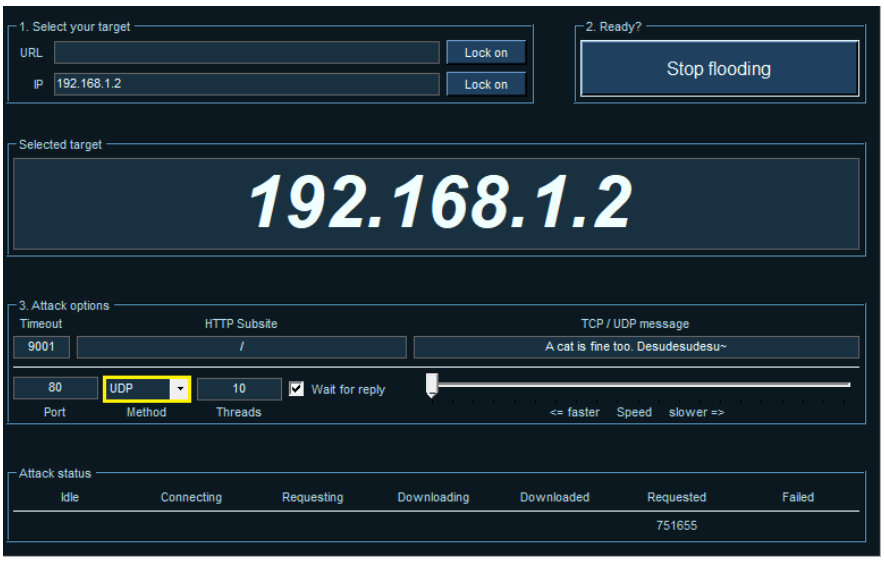

Figure 2.18 : Attaque Inondation UDP_LOIC.

➢ **Analyse du trafic**

Au cours de l'analyse, une accumulation significative de paquets UDP a été identifiée, provenant des adresses IP des attaquants 192.168.1.3 et 192.168.1.4. Ces paquets transportent des données spécifiques, contenant la chaîne "A cat is fine too. Desudesudesu". En ce qui concerne le nombre de paquets UDP reçus par seconde, un maximum de 12 500 paquets par seconde a été constaté pendant l'attaque, suivi d'une stabilisation, puis d'une diminution à la fin de celle-ci.

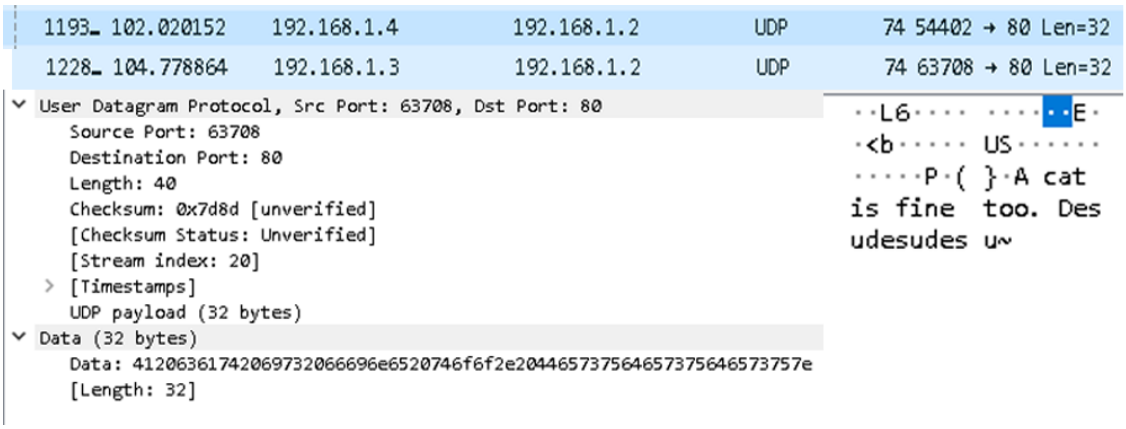

Figure 2.19 : Analyse des paquets Inondation UDP_LOIC.

➢ **Impact des attaques UDP**

L'analyse a été réalisée en utilisant des mesures du processeur (CPU) et de la bande passante du réseau (LAN).

- Absence d'attaque UDP : Lorsqu'il n'y a pas d'attaque UDP en cours, l'utilisation du processeur (CPU) reste basse, ne dépassant pas 2%, tandis que la bande passante du réseau (LAN) demeure presque inexistante, mesurée à 0,008 Mbps, indiquant une activité minimale du système.
- Attaque LOIC UDP : Une augmentation significative de l'utilisation du CPU à 9% est observée, accompagnée d'une forte augmentation de la bande passante du réseau à 76 Mbps. Cette observation suggère que l'attaque LOIC UDP consomme davantage de ressources réseau que de ressources CPU.
- Attaque HULK UDP : Présentant des caractéristiques similaires à l'attaque Loïc UDP, avec une utilisation du CPU également à 9%, mais une augmentation encore plus importante de la bande passante du réseau, atteignant 97 Mbps. Cela indique que l'attaque HULK UDP est particulièrement gourmande en bande passante réseau.

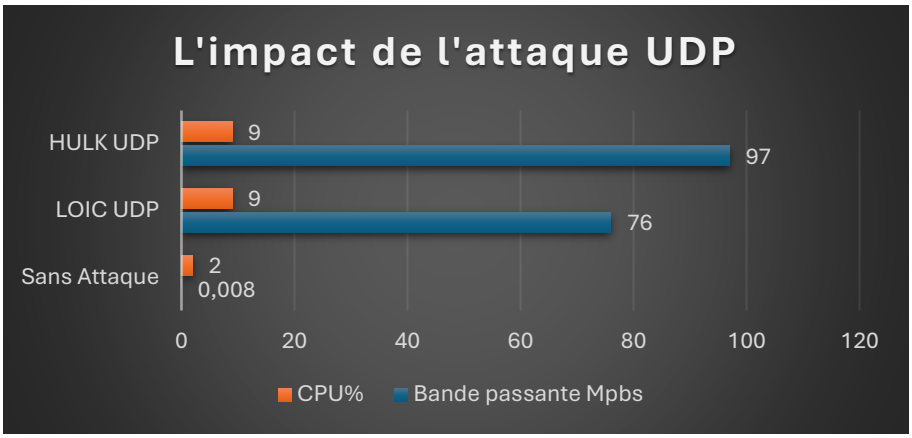

Figure 2.20 : Impact d'une attaque Inondation UDP sur les performances d'un serveur.

2.5.3 Attaque Inondation HTTP

a. GOLDENEYE

Pour lancer l'outil GOLDENEYE avec différentes techniques d'attaque, il vous suffit d'utiliser les commandes suivantes :

La commande: **python goldeneye.py http://192.168.1.2:80/ -s 5 -m post**

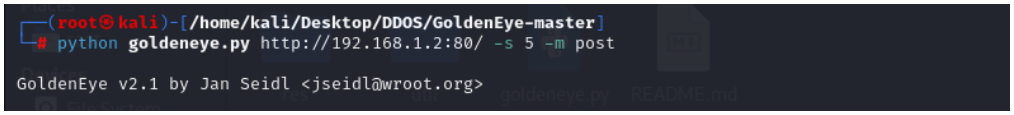

Figure 2.21: Attaque HTTP POST_GOLDENEYE.

La commande**: Python goldeneye.py http://192.168.1.2:80/ -s 5 -m get**

Figure 2.22 : Attaque HTTP GET_GOLDENEYE.

La commande: **python goldeneye.py http://192.168.1.2:80/ -s 5 -m random**

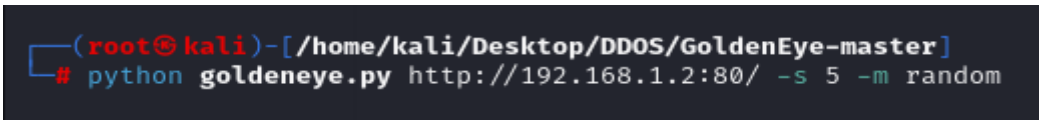

Figure 2.23 : Attaque HTTP Random_GOLDENEYE.

- -s 5 : Cette option spécifie le nombre de threads à utiliser pour l'attaque.
- -m : Cette option spécifie le type de méthode HTTP à utiliser pour l'attaque.

➢ **Analyse du trafic**

Durant l'analyse de l'attaque aléatoire (Random), caractérisée par une utilisation imprévisible et mélangée des méthodes POST et GET, une importante concentration de paquets

HTTP a été observée. Ces paquets, provenant des adresses IP des attaquants 192.168.1.3 et 192.168.1.4, véhiculent des données spécifiques, incluant les en-têtes "POST /?" et "GET /?", accompagnés de "HTTP/1.1". Il a été constaté que le nombre de paquets TCP est supérieur à celui des paquets HTTP, avec environ 56 000 paquets par seconde pour TCP contre seulement 12 000 paquets par seconde pour HTTP. Cette différence s'explique par le fait que chaque requête et réponse HTTP est encapsulée dans des segments TCP pour le transport sur le réseau.

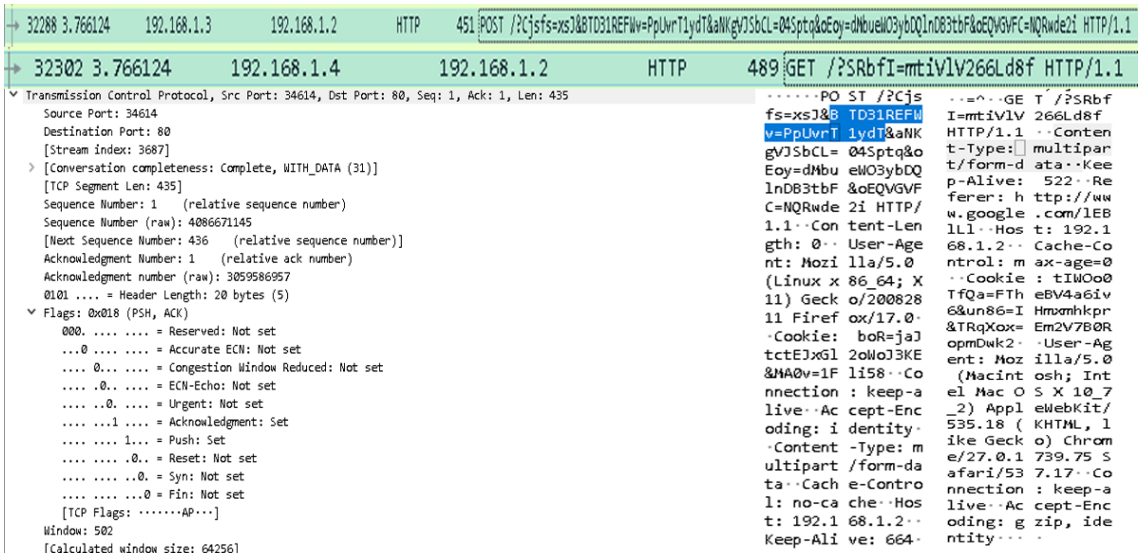

Figure 2.24 : Analyse des paquets HTTP _GOLDENEYE.

b. SLOWHTTPTEST

Pour simuler une attaque de type GET et POST en utilisant l'outil Slowhttptest, vous pouvez utiliser la commande suivante :

La commande **: slowhttptest -c 10000 -i 1 -l 240 -r 10 -s 8000 -t get -x 2000 -u http://192.168.1.2**

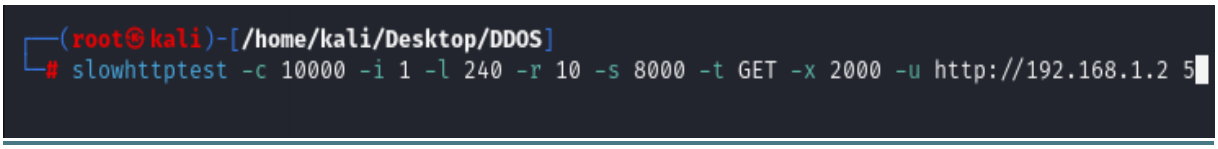

Figure 2.25 : Attaque HTTP GET_SLOWHTTPTEST.

La commande **: slowhttptest -c 10000 -i 1 -l 240 -r 10 -s 8000 -t post -x 2000 -u http://192.168.1.2**

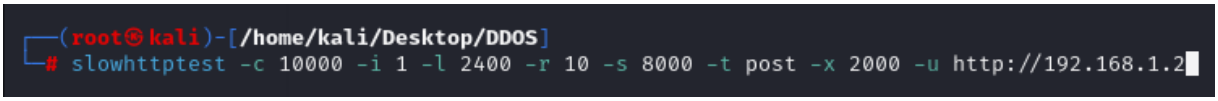

Figure 2.26 : Attaque HTTP POST_SLOWHTTPTEST.

- -c 10000 : Cette option spécifie le nombre de connexions à établir avec le serveur cible pendant l'attaque.

- -i 1 : Cette option spécifie l'intervalle entre chaque envoi de données de suivi.
- -l 240 : Cette option spécifie la durée totale du test en secondes.
- -r 10 : Cette option spécifie le taux de connexion (connexions par seconde).
- -s 8000 : Cette option spécifie la taille du contenu à envoyer dans la requête.
- -t : Cette option spécifie le type de requête HTTP à envoyer.
- -x 2000 : Cette option spécifie la longueur maximale de chaque paire nom/valeur de données de suivi par tick.
- -u http://192.168.1.2 : Cette option spécifie l'URL cible du serveur à attaquer.

➢ **Analyse du paquet**

Au cours de l'analyse, une accumulation significative de paquets HTTP a été détectée, provenant des adresses IP des attaquants 192.168.1.3 et 192.168.1.4. Ces paquets contenaient des données spécifiques, notamment une requête HTTP GET avec un en-tête "Referer" incluant le texte "TESTING_PURPOSES_ONLY". Une nette supériorité du nombre de paquets TCP par rapport aux paquets HTTP a été observée, avec environ 750 paquets par seconde pour TCP, tandis que le nombre de paquets HTTP était beaucoup plus réduit. Cette disparité s'explique par le fait que chaque tentative de connexion HTTP nécessite l'envoi d'un segment TCP pour établir et maintenir la connexion avec le serveur cible.

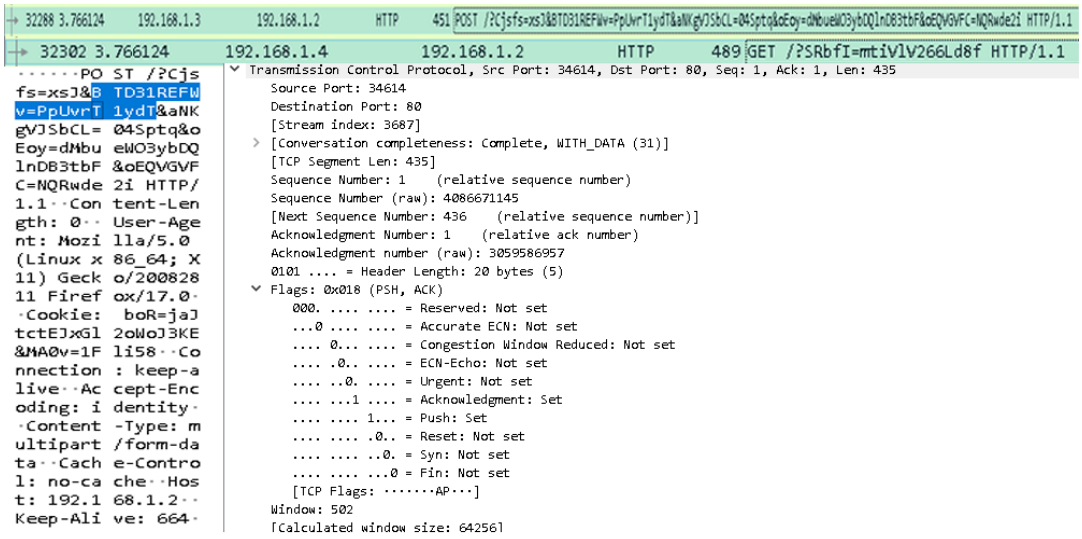

Figure 2.27 : Analyse des paquets HTTP _SLOWLORIS.

c. LOIC

Nous utilisons la même méthode que pour les attaques précédentes TCP et UDP. Nous sélectionnons le type d'attaque HTTP et spécifions l'adresse cible 192.168.1.2 et le port d'attaque 80.

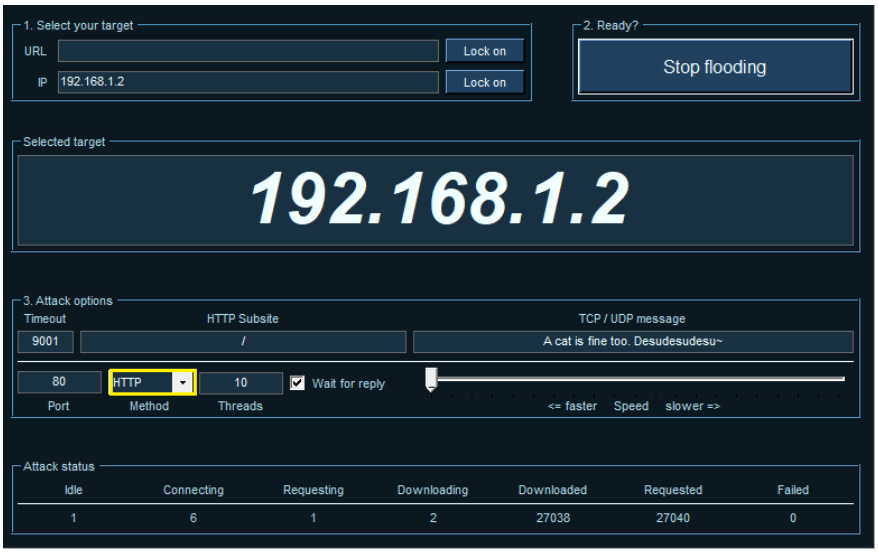

Figure 2.28 : Attaque Inondation HTTP _LOIC.

- **Analyse du paquet**

D'après l'analyse effectuée à l'aide de Wireshark, des paquets HTTP provenant des adresses IP des attaquants 192.168.1.3 et 192.168.1.4 ont été identifiés. Ces paquets contiennent une requête HTTP avec le contenu "GET / HTTP/1.0". Sur le graphique, une nette différence entre le nombre de paquets TCP et celui de paquets HTTP est observée. Environ 41 000 paquets par seconde sont attribués à TCP, tandis que le nombre de paquets HTTP est très limité. Cette disparité peut être expliquée par le fait que chaque requête HTTP nécessite l'envoi d'un segment TCP pour établir et maintenir la connexion avec le serveur, ce qui entraîne un nombre significatif de paquets TCP par rapport aux paquets HTTP.

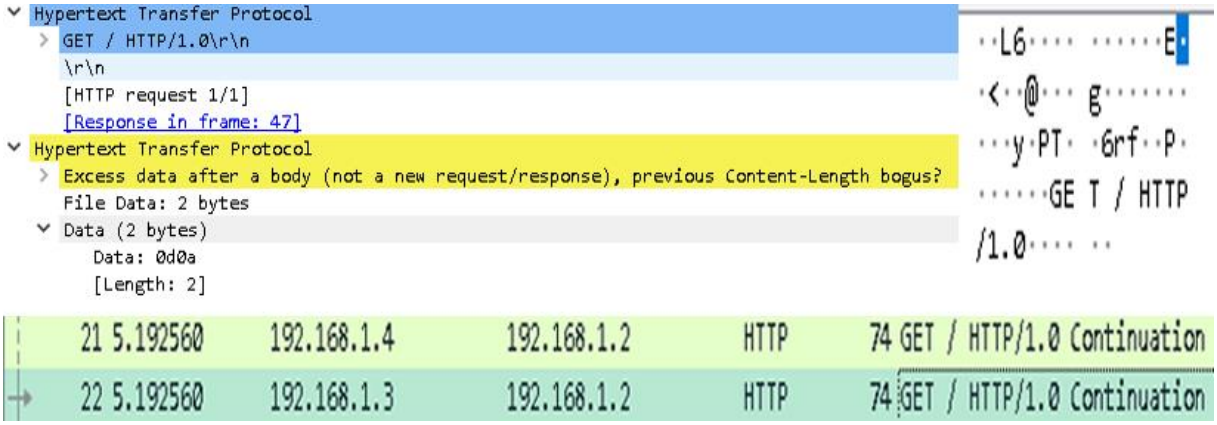

Figure 2.29 : Analyse des paquets HTTP _LOIC.

- **Impact des attaques HTTP**

L'analyse de la situation sans attaque HTTP en cours révèle que l'utilisation du processeur (CPU) est minime, s'élevant seulement à 2%. Cela indique une activité négligeable.

De plus, la bande passante du réseau local (LAN) est presque inexistante, mesurée à 0,008 Mbps.

- Attaque LOIC HTTP : Cette attaque se caractérise par une utilisation élevée du CPU, atteignant 51%. Son mode opératoire consiste à inonder le serveur avec un grand nombre de requêtes HTTP, ce qui nécessite un traitement intensif par le CPU pour y répondre.
- Attaque SLOWHTTPTEST : Bien que cette attaque n'affecte que légèrement l'utilisation du CPU, soit environ 8%, elle entraîne une diminution notable de la bande passante du réseau, chutant à 3,7 Mbps. Son objectif est de maintenir les connexions HTTP ouvertes le plus longtemps possible, saturant ainsi progressivement la capacité du serveur à répondre à de nouvelles demandes de connexion.
- Attaque GOLDENEYE : Cette attaque a un impact élevé à la fois sur l'utilisation du CPU, atteignant 30%, et sur la bande passante du réseau, montant jusqu'à 14 Mbps. Elle est probablement conçue pour épuiser simultanément les ressources du CPU et de la bande passante en inondant le serveur avec un grand volume de trafic HTTP.

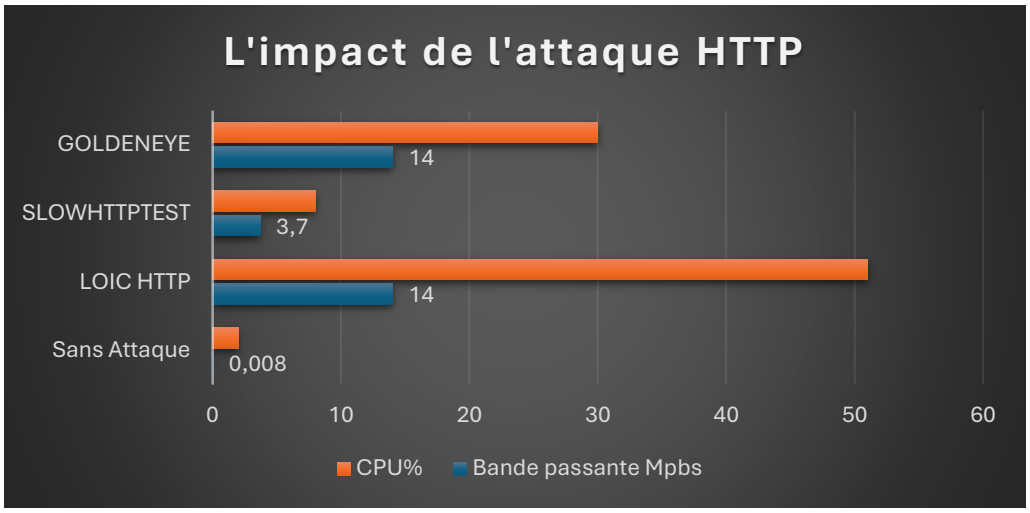

Figure 2.30 : Impact d'une attaque Inondation HTTP sur les performances d'un serveur.

2.5.4 Attaque Inondation ICMP

a. HPING3

Formule utilisée : hping3 -1 -flood 192.168.1.2 -a 192.168.1.255

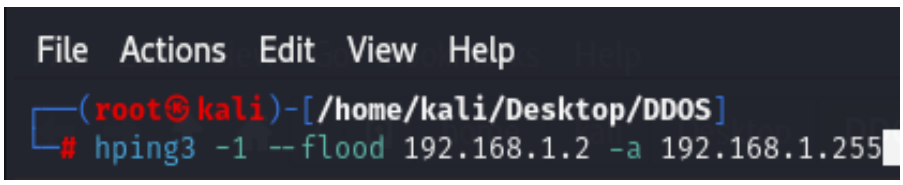

Figure 2.31 : Attaque Inondation ICMP_HPING3.

- -1 : Indique que HPING3 enverra des paquets ICMP.
- --flood : Envoie les paquets aussi rapidement que possible sans attendre de réponse.
- --a : Spécifie l'adresse IP source des paquets envoyés.

➢ **Analyse du paquet**

Durant l'incident, l'application de l'outil Wireshark sur l'hôte cible a été déployée dans le dessein d'observer et d'examiner le flux de données réseau engendré. Cette attaque a été caractérisée par l'exploitation d'une adresse cible en tant qu'adresse source pour l'émission ou la diffusion de multiples requêtes de ping ICMP. Les spécificités de ces requêtes incluaient une taille de données (dsize) de 0, un code (icode) de 0, et un type (itype) de 8. Le nombre de paquets ICMP a été particulièrement notable, atteignant jusqu'à 130 000 paquets par seconde (pps), suivi d'une fluctuation persistante jusqu'à la cessation de l'attaque. Ce modus operandi, généralement orchestré par un nombre considérable de dispositifs compromis, consiste en l'émission simultanée de quantités substantielles de trafic vers le réseau ciblé. Cette activité induit une déplétion des ressources du réseau, induisant une incapacité à répondre aux requêtes légitimes, provoquant ainsi des interruptions de service pour les usagers réguliers.

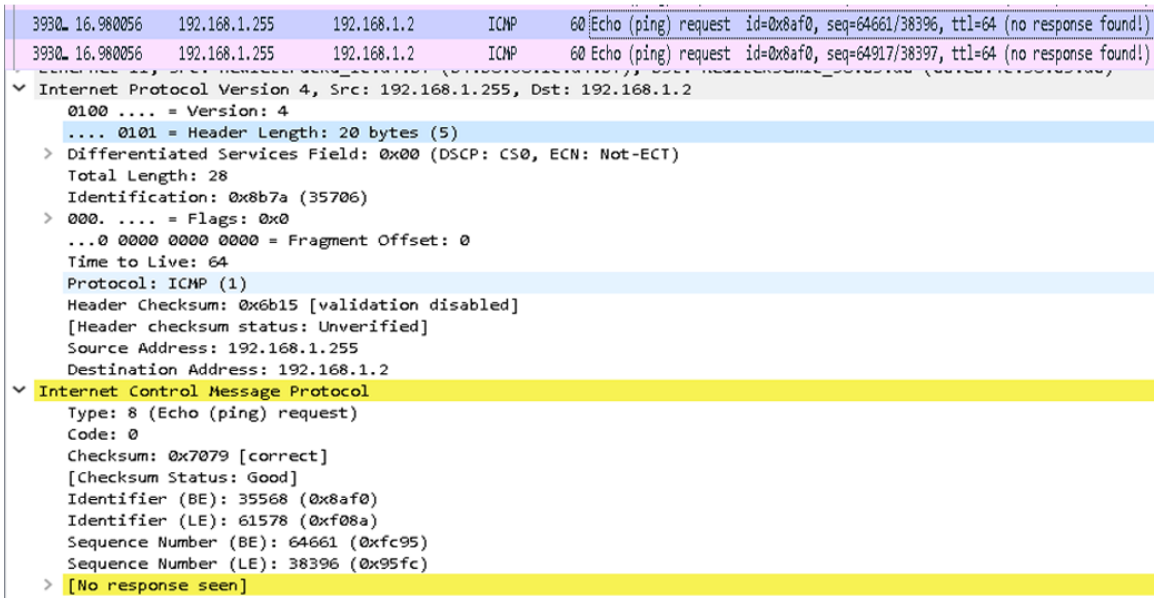

Figure 2.32 : Analyse des paquets ICMP _HPING3.

➢ **Impact de l'attaque ICMP**

Lors de l'attaque ICMP de HPING3, une augmentation significative de l'utilisation du CPU est observée, atteignant 25%. Cette augmentation peut être attribuée à la charge supplémentaire imposée au système pour répondre aux requêtes ICMP générées par l'attaque.

De plus, une augmentation importante de la bande passante du réseau est remarquée, évaluée à 70 Mbps. Cette hausse peut être attribuée au trafic ICMP généré par l'attaque, qui occupe une part importante de la capacité du réseau.

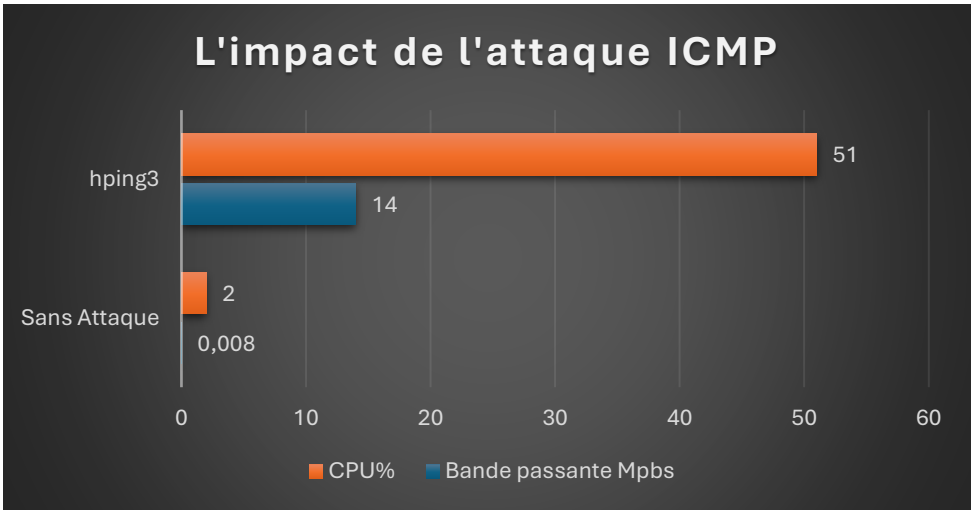

Figure 2.33 : Impact d'une attaque Inondation ICMP sur les performances d'un serveur.

2.6 Détections des signatures

Les signatures jouent un rôle crucial dans la défense contre les attaques DDoS. Elles permettent de repérer rapidement et avec précision les comportements suspects sur le réseau, ce qui est essentiel pour protéger les infrastructures et les services en ligne des organisations. En étant constamment mises à jour, elles peuvent s'adapter aux nouvelles menaces qui émergent, assurant ainsi une défense efficace contre les attaques malveillantes.

Tableau 2.2 : Signatures d'attaques.

Outil d'Attaque	Signature
LOIC UDP	Supérieur à 125 000 paquets/s UDP. Payload=32bytes content de paquet : Desudesudesu
LOIC TCP	Supérieur à 15 500 paquets/s TCP. Content de paquet :desudesudesu FLAGS=(PA) PUSH=1 et ACK=1
LOIC HTTP	Content de paquet : GET /http/1.0 Supérieur à 10 000 paquets/s HTTP. Supérieur à 36 000 paquets/s TCP FLAGS=(PA) PUSH=1 et ACK=1 ;
SLOWLORIS	Supérieur à 70 000 paquets/s TCP.

	Content de Paquet :"HTTP/1.1";"GET /?""User-Agent|3a| Mozilla/5.0 (Macintosh";"Intel Mac OS X 10_11_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36";"Accept-language: en-US,en,q=0.5";"X-a:"; Payload=20,21bytes; FLAGS=(PA) PUSH=1 et ACK=1;
TORHOMERS	Supérieur à 8 000 paquets/s TCP. FLAGS=(PA) PUSH=1 et ACK=1 ; Content de Paquet:"Content-Type: application/x-www-form-urlencoded";"p".
SLOWHTTPTEST GET	Supérieur à 600 paquets/s TCP. FLAGS=(PA) PUSH=1 et ACK=1 ; Content de paquet : "Referer: TESTING_PURPOSES_ONLY", content:"GET / HTTP/1.1".
SLOWHTTPTEST POST	Supérieur à 600 paquets/s TCP. FLAGS=(PA) PUSH=1 et ACK=1; Content de Paquet: "Referer: TESTING_PURPOSES_ONLY";"GET / HTTP/1.1"; Content de Paquet: "Referer: TESTING_PURPOSES_ONLY";"POST / HTTP/1.1".
HULK	Supérieur à 7 000 paquets/s UDP. Payload=1490bytes
HPING3 SMURF	Supérieur à 100 000 paquets/s ICMP payload:0 bytes code:0 ; type:8 ;
HPING3 LAND	Supérieur à 35 000 paquets/s SYN TCP FLAG (S) SYN=1 payload:0 bytes window:512
GOLDENEYE GET	Supérieur à 10 000 paquets/s HTTP. Supérieur à 50 000 paquets/s TCP. FLAG=(PA) PUSH=1, ACK=1 ;

	Content de Paquet: "GET /?";"HTTP/1.1";"Connection: keep-alive".
GOLDENEYE POST	Supérieur à 10 000 paquets/s HTTP. Supérieur à 50 000 paquets/s TCP. FLAG=(PA) PUSH=1, ACK=1 ; Content de Paquet: "POST /?";"HTTP/1.1";"Connection: keep-alive".

2.7 Conclusion

Dans cette section de notre étude, nous avons présenté une analyse détaillée de diverses attaques DDOS, mettant en lumière les différentes formes qu'elles peuvent revêtir. À travers une série de simulations, nous avons pu capturer les signatures de ces attaques, utilisant une gamme variée d'outils pour maximiser notre collecte de données. Cette approche nous permettra de concevoir nos propres règles de détection, lesquelles seront ensuite soumises à des tests rigoureux avec l'IPS SURICATA dans le prochain chapitre.

Chapitre 3 Détection et prévention des attaques DDOS

3.1 Introduction

Dans ce chapitre, nous explorons un exemple concret d'utilisation de systèmes IDS/IPS, en mettant l'accent sur l'application Suricata. Nous analyserons les paramètres clés pour nous assurer de leur efficacité. En développant nos propres règles de détection et de prévention à partir de signatures préalables, nous évaluerons ensuite les performances de notre solution en simulant des attaques DDOS.

3.2 Implémentation des Suricata

Nous avons mis en place un schéma de protection utilisant les mécanismes de sécurité les plus couramment utilisés, tels que Suricata.

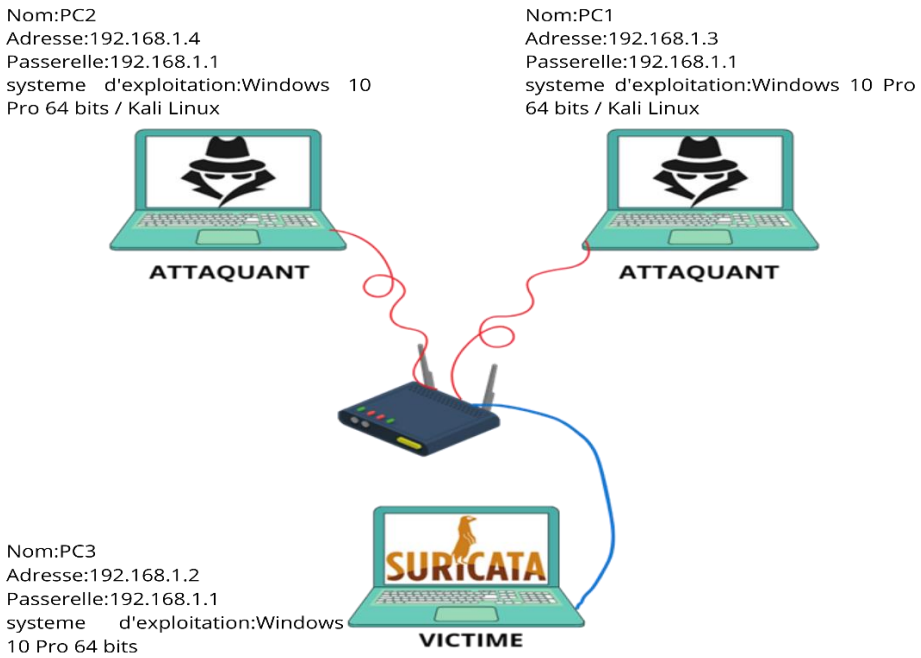

Figure 3.1 : Architecture de Protection.

3.3 Configuration Suricata IPS

Après avoir installé Suricata, la configuration correcte devient cruciale pour assurer son efficacité.

1. Sélection de l'Interface Réseau

La définition des interfaces réseau sur lesquelles Suricata va écouter le trafic est une étape primordiale.

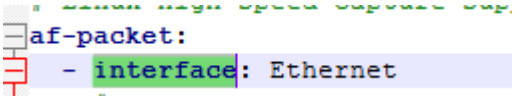

Figure 3.2 : configuration de l'interface.

2. Configuration des Réseaux Internes

Nous avons attribué à HOME_NET le réseau 192.168.0.0/16 pour couvrir notre réseau interne.

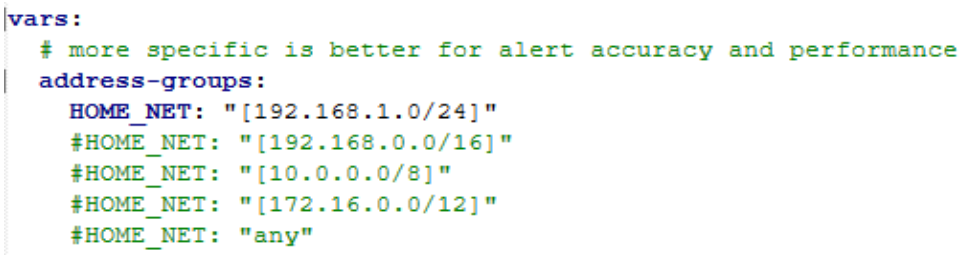

Figure 3.3 : Paramétrage des Plages IP Suricata.

3. Chargement des Règles

La définition des règles de détection spécifiques aux besoins de notre réseau. Suricata dispose d'une large bibliothèque de règles préchargées, mais le développement de règles sur mesure permet de cibler des menaces propres à notre contexte.

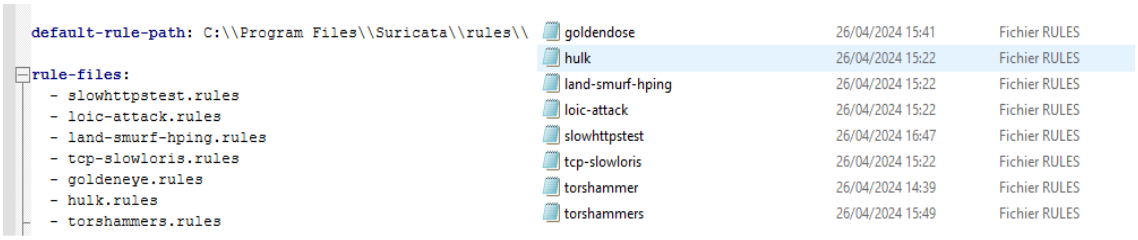

Figure 3.4 : Path Rule avec documents.

3.4 Implémentation des règles

L'application précise des règles dans les outils de défense tels que Suricata est essentielle, elles permettent l'analyse et le filtrage des paquets d'attaque, garantissant ainsi la sécurité du système. Cependant, des règles mal définies peuvent entraîner des risques en générant de fausses alertes ou en identifiant incorrectement les attaques. Pour remédier à cela, nous avons créé nos propres règles basées sur les signatures de ces attaques, assurant ainsi une détection plus fiable et une réduction des faux positifs.

3.4.1 Attaque TCP HPING3 LAND

- **Règle**

```
drop tcp any any -> any any (msg:"HPING3 LAND TCP Packet with SYN Flag Set"; flags: S; dsize: 0; window:512; sid:1000002;)
```

- **Explication**

Cette règle intercepte et bloque les paquets TCP où le drapeau SYN est activé sans contenu de données. Elle vise à prévenir les attaques LAND, caractérisées par des paquets envoyés à la même adresse source et destination.

3.4.2 Attaque TCP LOIC

- **Règle**

```
drop tcp any any -> any any (msg:"TCP LOIC ATTACK"; flow:to_server,established; content:"desudesudesu"; nocase; threshold: type limit,track by_src,seconds 1, count 1000; sid:1000008;)
```

- **Explication**

Cette règle est conçue pour bloquer les paquets TCP suspects qui correspondent à un modèle utilisé typiquement par LOIC pour générer un volume élevé de trafic vers un serveur cible. En filtrant les paquets contenant la chaîne spécifique "desudesudesu" et en appliquant un seuil de détection (1000 paquets par seconde par source), cette règle vise à empêcher les attaques TCP DDoS qui pourraient saturer et déstabiliser les serveurs.

3.4.3 Attaque TCP TORSHAMMER

- **Règle**

```
drop tcp any any -> any any (msg:"TCP TORHAMMER ATTACK"; flow:established,to_server; content:"Content-Type: application/x-www-form-urlencoded"; http_header; nocase; sid:1000013; threshold:type limit, track by_src, count 100, seconds 1;)

drop tcp any any -> any any (msg:"TCP TORHAMMER ATTACK"; flags: PA; dsize: 1; content:"p"; sid:10000014; threshold:type limit, track by_src, count 100, seconds 1;)
```

- **Explication**

Ces règles visent à détecter et bloquer les attaques TORHAMMER, qui cherchent à surcharger les serveurs web.

Elles filtrent les paquets contenant des requêtes HTTP spécifiques "Content-Type : application/x-www-form-urlencoded" et des paquets TCP particuliers avec le drapeau "Push Acknowledgement" (PA) et une taille de données de 1 octet contenant le caractère "p" ; Si un seul hôte envoie plus de 100 requêtes ou paquets correspondants en une seconde, les règles déclenchent un blocage, protégeant ainsi les serveurs contre une saturation dommageable.

3.4.4 Attaque SLOWLORIS

- **Règles**

Plusieurs configurations pour identifier et stopper les attaques Slowloris, qui exploitent les limites des serveurs en maintenant ouvertes des connexions HTTP avec des en-têtes incomplets.

```
drop tcp any any -> any any (msg:"SLOWLORIS ATTACK"; flags: PA; content:"GET /?"; content:"HTTP/1.1"; nocase; dsize:21; threshold: type limit, track by_src, count 100, seconds 1; sid:1000003;)
```

```
drop tcp any any -> any any (msg:"SLOWLORIS ATTACK"; flags: PA; content:"GET /?"; content:"HTTP/1.1"; nocase; dsize:20; threshold: type limit, track by_src, count 100, seconds 1; sid:1000004;)
```

```
drop tcp any any -> any any (msg:"SLOWLORIS ATTACK"; content:"User-Agent|3a| Mozilla/5.0 (Macintosh"; nocase; content:"Intel Mac OS X 10_11_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36"; content:"Accept-language: en-US,en,q=0.5"; threshold: type limit, track by_src, count 100, seconds 1; sid:1000005;)
```

```
drop tcp any any -> any any (msg:"SLOWLORIS ATTACK"; content:"X-a:";nocase; flags: PA; threshold: type limit, track by_src, , count 100, seconds 1; sid:1000006;)
```

- **Explication**

Ces règles sont essentielles pour prévenir cette approche d'attaque qui utilise peu de ressources de l'attaquant pour avoir un impact significatif sur la disponibilité du serveur cible. Elles empêchent les connexions TCP qui contient des demandes HTTP GET de 21 octets ou plus avec le drapeau PA et les demandes de 20 octets ou plus afin de repérer des variantes de l'attaque, repèrent les demandes HTTP avec un User-Agent spécifique souvent lié à Slowloris, et recherchent une chaîne spécifique "X-a:" dans le flux TCP. Les connexions sont également limitées à 100 par adresse source en une seconde.

3.4.5 Attaques par inondation UDP

- **Règle par inondation UDP HULK**

```
drop udp any any -> any any (msg:"HULK ATTACK"; dsize:1490; threshold: type threshold, track by_src, count 100, seconds 1; sid:1000012;)
```

- **Règle par inondation UPD LOIC**

```
drop udp any any -> any any (msg:"Attaque UDP_flood LOIC"; content:"Desudesudesu"; dsize:32; threshold: type threshold, track by_src, count 1000, seconds 1; sid:1000007;)
```

- **Explication**

Ces règles bloquent les grandes quantités de trafic UDP générées par des outils comme HULK et LOIC, qui sont utilisés pour mener des attaques par inondation afin de saturer les réseaux et serveurs ciblés.

Les règles "HULK" visent les attaques HULK qui inondent les serveurs avec des paquets UDP de 1490 octets de taille, tandis que les règles "LOIC" repèrent les attaques UDP_flood de LOIC en empêchent les paquets UDP contenant la chaîne "Desudesudesu" et de 32 octets de taille. Les deux ensembles de règles sont limités respectivement à 100 et 1000 connexions par adresse source en une seconde.

3.4.6 Attaque GOLDENEYE (HTTP GET et POST)

- **Règle GET**

```
drop http any any -> any any (msg:"HTTP GET GOLDENEYE"; content:"GET /?"; content:"HTTP/1.1"; nocase; content:"Connection: keep-alive"; http_header; sid:1000015;)
```

- **Règle POST**

```
drop http any any -> any any (msg:"HTTP POST GOLDENEYE"; content:"POST /?"; content:"HTTP/1.1"; nocase; content:"Connection: keep-alive"; http_header; sid:1000016;)
```

- **Explication**

Ces règles sont conçues pour détecter les requêtes HTTP spécifiques qui pourraient signaler une attaque par déni de service distribué (DDOS) à l'aide de l'outil GoldenEye, qui exploite la persistance des connexions HTTP pour épuiser les ressources du serveur ; Elles ont pour objectif de prévenir de telles attaques en repérant et en bloquant les requêtes HTTP malveillantes.

La règle "GET" cible les requêtes HTTP GET contenant "GET /?" et "HTTP/1.1", avec l'en-tête "Connection: keep-alive" tandis que La règle "POST" cible les requêtes HTTP POST similaires, mais avec "POST /?" .

3.4.7 Attaque SLOWLORISHTTPTST(GET/POST)

- **Règles GET**

```
drop http any any -> any any (msg:"SLOWHTTPTEST GET request"; content:"GET / HTTP/1.1"; nocase; content:"Referer: TESTING_PURPOSES_ONLY"; sid:1000010;)
```

```
drop tcp any any -> any any (msg:"SLOWHTTPTEST GET request"; content:"get / HTTP/1.1"; nocase; content:"Referer: TESTING_PURPOSES_ONLY"; sid:1000033;)
```

- **Règles POST**

```
drop http any any -> any any (msg:"SLOWHTTPTEST GET request"; content:"GET / HTTP/1.1"; nocase; content:"Referer: TESTING_PURPOSES_ONLY"; sid:1000010;)
```

```
drop http any any -> any any (msg:"SLOWHTTPHTTP POST request"; content:"POST / HTTP/1.1"; nocase; content:"Referer: TESTING_PURPOSES_ONLY"; sid:1000011;)
```

- **Explication**

Ces règles sont conçues pour détecter et bloquer les requêtes HTTP générées par des outils de test de HTTP lent (SLOWHTTPTEST), utilisés pour mener des attaques visant à saturer les serveurs web en épuisant leurs ressources par des connexions HTTP incomplètes.

b. Attaque HTTP GET LOIC

- **Règle**

```
drop tcp any any -> any any (msg:"HTTP GET request LOIC"; content:"GET / HTTP/1.0"; nocase; sid:1000009;)
```

- **Explication**

Cette règle vise à bloquer les requêtes TCP qui ressemblent à des requêtes HTTP GET générées par l'outil LOIC (Low Orbit Ion Cannon), une technique couramment utilisée dans les attaques DDOS. Elle cible spécifiquement les requêtes HTTP GET utilisant le protocole HTTP/1.0, moins courant dans le trafic web moderne (HTTP/1.1 et HTTP/2 sont plus fréquents), afin de prévenir les tentatives d'exploitation de cette méthode et de limiter le risque de surcharge des serveurs causé par ces attaques.

c. Attaque SMURF via ICMP (HPING3)

- **Règle**

```
drop icmp any any -> any any (msg:"Possible ICMP_hping3 SMURF"; dsize:0; icode:0; itype:8; threshold: type threshold, track by_src, count 500, seconds 1; sid:1000001;)
```

- **Explication**

Cette règle bloque les paquets ICMP de type 'echo request' qui n'ont pas de taille de données. Elle est utilisée pour prévenir les attaques SMURF, où l'attaquant inonde le

réseau victime avec de multiples requêtes ICMP pour provoquer une surcharge En mettant en place cette mesure, on cherche à maintenir la stabilité et la disponibilité du réseau en prévenant ce type d'attaque.

3.5 Évaluation des Règles

Après la mise en place et la configuration de Suricata, il est nécessaire de procéder à des tests de simulation d'attaques réseau. Ces tests cibleront différents protocoles comme UDP, TCP, ICMP ainsi que les attaques par inondation HTTP. Ils permettront d'évaluer la capacité de Suricata à détecter et bloquer en temps réel les activités malveillantes.

- **Validation de la Règle de Blocage pour l'Attaque TCP HPING3 LAND**

Suricata a détecté une attaque HPING3 LAND ciblant l'adresse IP 192.168.1.2 sur le port 80, provenant des adresses IP 192.168.1.3 et 192.168.1.4. Pour contrer cette menace, il a immédiatement bloqué le trafic malveillant, assurant ainsi la protection et la stabilité du réseau.

```
04/28/2024-14:41:11.341276  [Drop] [**] [1:1000002:0] HPING3 LAND TCP Packet with SYN Flag Set  [**] [Classification: (null)] [Priority
: 3] {TCP} 192.168.1.2:25636 -> 192.168.1.2:80
04/28/2024-14:41:11.337013  [Drop] [**] [1:1000002:0] HPING3 LAND TCP Packet with SYN Flag Set  [**] [Classification: (null)] [Priority
: 3] {TCP} 192.168.1.2:25029 -> 192.168.1.2:80
04/28/2024-14:41:11.318472  [Drop] [**] [1:1000002:0] HPING3 LAND TCP Packet with SYN Flag Set  [**] [Classification: (null)] [Priority
: 3] {TCP} 192.168.1.2:22441 -> 192.168.1.2:80
```

Figure 3.5 : Suivi des alertes et des drops en temps réel pour l'Attaque TCP HPING3 LAND.

- **Validation de la Règle de Blocage pour l'Attaque TCP LOIC**

Suricata a détecté une attaque TCP LOIC qui vise l'adresse IP 192.168.1.2 sur le port 80, en provenance des adresses IP 192.168.1.3 et 192.168.1.4. Afin de faire face à cette menace, il a immédiatement arrêté le trafic malveillant, garantissant ainsi la sécurité et la stabilité du réseau.

```
04/28/2024-15:00:20.916249  [Drop] [**] [1:1000008:0] TCP LOIC ATTACK [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.1.3:557
38 -> 192.168.1.2:80
04/28/2024-15:00:21.621211  [Drop] [**] [1:1000008:0] TCP LOIC ATTACK [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.1.4:175
31 -> 192.168.1.2:80
04/28/2024-15:00:21.946439  [Drop] [**] [1:1000008:0] TCP LOIC ATTACK [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.1.3:557
39 -> 192.168.1.2:80
```

Figure 3.6 : Suivi des alertes et des drops en temps réel l'Attaque TCP LOIC.

- **Validation de la Règle de Blocage pour l'Attaque TCP TORSHAMMER**

Suricata a détecté une attaque TCP TORSHAMMER visant l'adresse IP 192.168.1.2 sur le port 80, provenant des adresses IP 192.168.1.3 et 192.168.1.4. Pour contrer cette menace, il a immédiatement bloqué le trafic malveillant, assurant ainsi la protection et la stabilité du réseau.

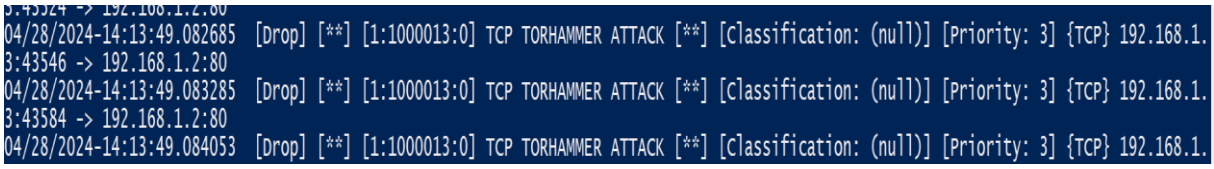

Figure 3.7 : Suivi des alertes et des drops en temps réel l'Attaque TCP TORSHAMMER.

- **Validation de la Règle de Blocage pour l'Attaque TCP SLOWLORIS**

Suricata a repéré une attaque TCP SLOWLORIS qui vise l'adresse IP 192.168.1.2 sur le port 80, en provenance des adresses IP 192.168.1.3 et 192.168.1.4. Afin de faire face à cette menace, il a immédiatement arrêté le trafic malveillant, garantissant ainsi la sécurité et la stabilité du réseau.

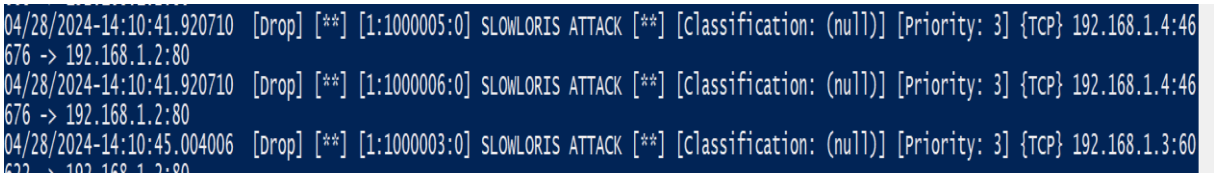

Figure 3.8 : Suivi des alertes et des drops en temps réel l'Attaque TCP SLOWLORIS.

- **Validation de la Règle de Blocage pour l'Attaque UDP HULK**

Suricata a détecté une attaque UDP HULK ciblant l'adresse IP 192.168.1.2 sur le port 80, provenant des adresses IP 192.168.1.3 et 192.168.1.4. Pour contrer cette menace, il a immédiatement bloqué le trafic malveillant, assurant ainsi la protection et la stabilité du réseau.

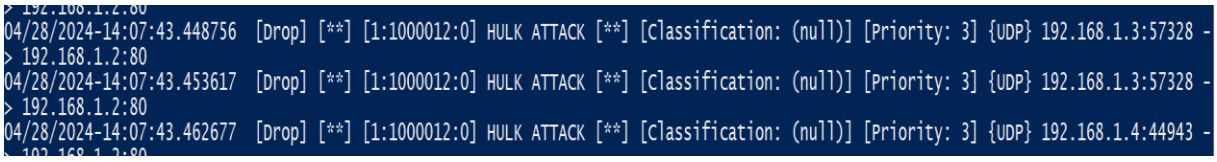

Figure 3.9 : Suivi des alertes et des drops en temps réel l'Attaque UDP HULK.

- **Validation de la Règle de Blocage pour l'Attaque UDP LOIC**

Suricata a détecté une attaque UDP LOIC qui vise l'adresse IP 192.168.1.2 sur le port 80, en provenance des adresses IP 192.168.1.3 et 192.168.1.4. Afin de faire face à cette menace, il a immédiatement arrêté le trafic malveillant, garantissant ainsi la sécurité et la stabilité du réseau.

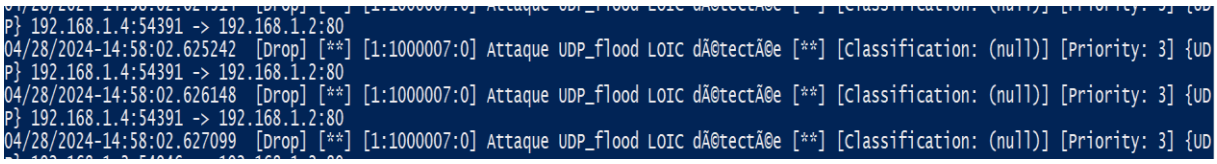

Figure 3.10 : Suivi des alertes et des drops en temps réel l'Attaque TCP UDP LOIC.

- **Validation de la Règle de Blocage pour l'Attaque HTTP GOLDENEYE GET**

Suricata a détecté une attaque HTTP GOLDENEYE GET visant l'adresse IP 192.168.1.2 sur le port 80, provenant des adresses IP 192.168.1.3 et 192.168.1.4. Pour contrer cette menace, il a immédiatement bloqué le trafic malveillant, assurant ainsi la protection et la stabilité du réseau.

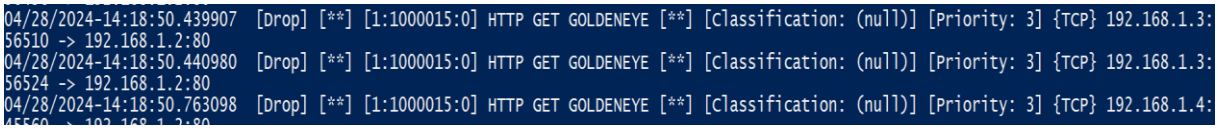

Figure 3.11 : Suivi des alertes et des drops en temps réel l'Attaque GOLDENEYE GET.

- **Validation de la Règle de Blocage pour l'Attaque HTTP GOLDENEYE POST**

Suricata a détecté une attaque HTTP GOLDENEYE POST visant l'adresse IP 192.168.1.2 sur le port 80, provenant des adresses IP 192.168.1.3 et 192.168.1.4. Pour contrer cette menace, il a immédiatement bloqué le trafic malveillant, assurant ainsi la protection et la stabilité du réseau.

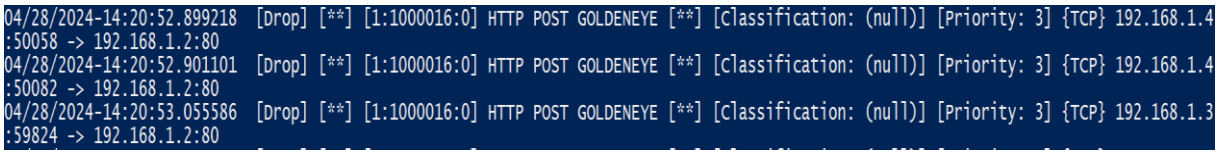

Figure 3.12 : Suivi des alertes et des drops en temps réel l'Attaque GGOLDENEYE POST.

- **Validation de la Règle de Blocage pour l'Attaque SLOWHTTPTEST GET**

Suricata a détecté une attaque HTTP SWLOHTTPTEST GET visant l'adresse IP 192.168.1.2 sur le port 80, provenant des adresses IP 192.168.1.3 et 192.168.1.4. Pour contrer cette menace, il a immédiatement bloqué le trafic malveillant, assurant ainsi la protection et la stabilité du réseau.

Figure 3.13 : Suivi des alertes et des drops en temps réel l'Attaque SLOWHTTPTEST GET.

- **Validation de la Règle de Blocage pour l'Attaque SLOWHTTPTEST POST**

Suricata a détecté une attaque HTTP SWLOHTTPTEST POST visant l'adresse IP 192.168.1.2 sur le port 80, provenant des adresses IP 192.168.1.3 et 192.168.1.4. Pour contrer

cette menace, il a immédiatement bloqué le trafic malveillant, assurant ainsi la protection et la stabilité du réseau.

Figure 3.14 : Suivi des alertes et des drops en temps réel l'Attaque SLOWHTTPTEST POST.

- **Validation de la Règle de Blocage pour l'Attaque HTTP LOIC**

Suricata a détecté une attaque HTTP LOIC qui vise l'adresse IP 192.168.1.2 sur le port 80, en provenance des adresses IP 192.168.1.3 et 192.168.1.4. Afin de faire face à cette menace, il a immédiatement arrêté le trafic malveillant, garantissant ainsi la sécurité et la stabilité du réseau.

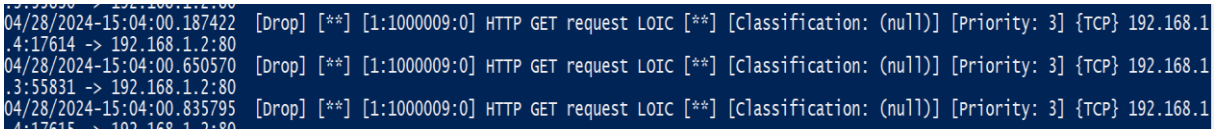

Figure 3.15 : Suivi des alertes et des drops en temps réel l'Attaque HTTP LOIC.

- **Validation de la Règle de Blocage pour l'Attaque ICMP HPING3 SMURF**

Suricata a détecté une attaque HPING3 SMURF ciblant l'adresse IP 192.168.1.2 sur le port 80, provenant des adresses IP 192.168.1.3 et 192.168.1.4. Pour contrer cette menace, Suricata a immédiatement bloqué le trafic malveillant, assurant ainsi la protection et la stabilité du réseau.

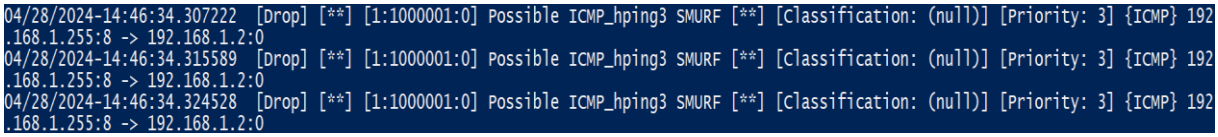

Figure 3.16 : Suivi des alertes et des drops en temps réel l'Attaque ICMP HPING3 SMURF.

3.6 Analyse des résultats obtenus

D'après les données recueillies par les capteurs, Suricata a réussi à contrer les attaques de manière exhaustive en interdisant non seulement les actions offensives, mais aussi en entravant la transmission des paquets vers leur destinataire et toute réponse éventuelle de la victime à l'agresseur. Parmi les attaques totalement neutralisées, on peut mentionner TCP LOIC, TCP TORSHAMMER, TCP SLOWLORIS, GOLDENEYE et HTTP LOIC. En revanche, d'autres formes d'attaques telles que Slowhttptest, UDP LOIC, HULK, ICMP

SMURF, HPING3 en TCP et LAND HPING3 ont simplement vu leurs communications bloquées.

L'action de Suricata s'est déroulée sans générer d'effets négatifs perceptibles, tels que des interruptions involontaires lors de la mise en "drop", ce qui confirme son efficacité dans la protection des serveurs contre les attaques.

3.7 Discussion

La détection et le blocage efficaces des attaques par Suricata reposent en grande partie sur la création de règles basées sur des signatures d'attaques connues. Ces règles permettent à Suricata d'identifier et de neutraliser efficacement les menaces, comme le montrent les résultats obtenus. Toutefois, un problème subsiste : La détection de nouvelles formes d'attaques. Les systèmes basés sur des signatures comme Suricata peuvent rencontrer des difficultés lorsqu'ils sont exposés à de nouvelles attaques ou à des variations de techniques d'attaque existantes.

Pour surmonter cette limitation, il est nécessaire d'envisager des tâches futures, telles que :

- **Intégrer l'apprentissage automatique et l'apprentissage profond** : Ces technologies peuvent analyser les modèles de trafic en temps réel et détecter les anomalies qui pourraient indiquer des attaques nouvelles ou inconnues. Elles peuvent créer des modèles adaptatifs capables d'évoluer avec les techniques d'attaque.

3.8 Conclusion

Dans ce chapitre, nous avons exploré l'efficacité de Suricata en tant que système de détection et de prévention des intrusions dans un contexte réseau.

Nous avons réalisé une série de tests et d'analyses pour évaluer sa capacité à détecter et à contrer différentes attaques informatiques. En élaborant nos propres règles de détection et en les implémentant dans Suricata, nous avons confirmé sa capacité à détecter en temps réel des attaques DDOS, tout en évitant les fausses alertes.

La réalisation réussie de cette mise en place met en évidence l'importance des systèmes de prévention d'intrusions dans la sécurité globale d'un réseau, garantissant une protection cruciale contre les risques cybernétiques.

Conclusion Générale

Les réseaux informatiques sont particulièrement vulnérables aux attaques par déni de service (DOS) et par déni de service distribué (DDOS), qui peuvent paralyser les services en ligne en les saturant de trafic malveillant. Ces attaques soulignent l'importance cruciale de la sécurisation des réseaux pour garantir leur disponibilité et leur intégrité.

Pour faire face à ces menaces, nous avons utilisé le système de détection et de prévention des intrusions (IDPS) de Suricata. Notre étude a consisté à configurer et à personnaliser Suricata, à développer des règles de détection spécifiques et à évaluer ses performances en simulant des attaques DDOS.

Les résultats ont montré que Suricata est capable de détecter et de bloquer les attaques DDOS en temps réel sans générer de fausses alertes importantes, et qu'il gère efficacement un grand volume de trafic. Des règles personnalisées ont permis de contrer des attaques du type TCP LOIC, SLOWLORIS et UDP HULK, démontrant ainsi l'efficacité de notre solution.

Pour améliorer encore la détection, nous recommandons de mettre régulièrement à jour les règles de détection et d'optimiser les configurations réseau. L'intégration de techniques d'apprentissage profond peut également augmenter la précision et l'efficacité de Suricata.

Les tâches futures devraient inclure l'intégration d'autres outils de sécurité, des tests dans différents environnements de production et le développement de modèles d'apprentissage en profondeur pour détecter des attaques plus intelligentes.

Bibliographie

[1] [Cloudflare], "Learning DDoS: Famous DDoS Attacks," https://www.cloudflare.com/fr-fr/learning/ddos/famous-ddos-attacks/. [Consulté le : 30 décembre 2023].

[2] [Smartyou], "Définition de la sécurité informatique," https://www.smartyou.ch/definition-securite-informatique/. [Consulté le : 14 mars 2024].

[3] [Sécurité Info], "Les 5 principes fondamentaux de la cybersécurité," https://www.securiteinfo.com/conseils-cybersecurite/les-5-principes-fondamentaux-cybersecurite-dican.shtml. [Consulté le : 14 mars 2024].

[4] [SOS Ransomware], "Comprendre les attaques par déni de service (DOS) et DDOS," https://sosransomware.com/cybersecurite/comprendre-les-attaques-par-deni-de-service-dos-et-ddos/. [Consulté le : 14 mars 2024].

[5] [Cyber.gc.ca], "Défense contre les attaques par déni de service distribué (DDOS)," https://www.cyber.gc.ca/fr/orientation/defense-contre-attaques-deni-service-distribue-ddos. [Consulté le : 16 mars 2024].

[6] [Akamai], "Qu'est-ce qu'un DDOS ?," https://www.akamai.com/fr/glossary/what-is-ddos. [Consulté le : 16 mars 2024].

[7] D. K. Bhattacharyya and J. K. Kalita, *DDOS Attacks: Evolution, Detection, Prevention, Reaction, and Tolerance*. CRC Press, 2016. [Consulté le : 16 mars 2024].

[8] [Wallarm], "What is HULK (HTTP Unbearable Load King) ?," https://www.wallarm.com/what/what-is-hulk-http-unbearable-load-king. [Consulté le : 16 mars 2024].

[9] [Radware], "Tor's Hammer," https://www.radware.com/security/ddos-knowledge-center/ddospedia/tors-hammer/. [Consulté le : 16 mars 2024].

[10] [Diva Portal], "An Investigation of Slow HTTP DoS attacks on Intrusion Detection Systems," https://www.diva-portal.org/smash/get/diva2:1740303/FULLTEXT02. [Consulté le : 17 mars 2024].

[11] [Guru99], "Outils d'attaque DDoS," https://www.guru99.com/fr/ddos-attack-tools.html. [Consulté le : 17 mars 2024].

[12] [LeBigData], "VPN contre attaques DDoS," https://www.lebigdata.fr/vpn-contre-attaques-ddos. [Consulté le : 20 mars 2024].

[13] [Pluralsight], "Vidéo sur les attaques DDoS," https://app.pluralsight.com/ilx/video-courses/clips/50753b79-5e9e-435b-a76d-285797d67e68. [Consulté le : 20 mars 2024].

[14] Okta, "IDS vs IPS," https://www.okta.com/fr/identity-101/ids-vs-ips/?id=countrydropdownheader-FR. [Consulté le : 20 mars 2024].

[15] Suricata Documentation, "Introduction aux règles Suricata," https://docs.suricata.io/en/suricata-7.0.3/rules/intro.html. [Consulté le : 20 mars 2024].