République Démocratique Populaire Algérien

Ministère de L'enseignement Supérieur Et de La Recherche Scientifique

Université 20 Aout 1955

Faculté de Science

Département d'Informatique

Spécialité : Réseaux Et Systèmes Distribués

Rapport de projet de fin d'études pour l'obtention du diplôme de

Master professionnel en informatique

Thème

Système de Détection d'Intrusion

Basé sur les Algorithmes Génétiques

Encadré par :

Touil Ghassen

Rédigé par :

Boulares Achref

Mallem Salah Eddine

Année universitaire:2022/2023

Remerciements

Tout d'abord, nous tenons à remercier Dieu pour nous avoir accordés la santé, la détermination et la persévérance nécessaires pour mener à bien ce projet. Sa grâce et Sa guidance nous ont accompagnés tout le long de ce parcours.

Nous adressons également nos remerciements les plus sincères à notre encadreur, M. Touil Ghassen, pour ses conseils éclairés, son expertise et sa disponibilité. Ses orientations précieuses nous ont permis d'avancer dans la bonne direction et de surmonter les difficultés rencontrées. Sa confiance en nos capacités nous a encouragés à donner le meilleur de nous-mêmes.

Nous tenons à exprimer notre gratitude envers les membres du jury qui ont accepté de consacrer leur temps et leurs compétences pour évaluer ce mémoire. Leurs remarques et suggestions constructives ont contribué à l'amélioration de notre travail et à notre progression académique.

Un merci spécial à nos parents et à nos proches, qui ont été d'un soutien indéfectible tout au long de cette aventure. Leur amour, leur compréhension et leurs encouragements ont été une source de motivation essentielle. Leur présence et leur soutien inconditionnel ont été une force qui nous a permis de surmonter les défis et d'atteindre nos objectifs.

Résumé

Ce mémoire se concentre sur le développement d'un système de détection d'intrusion basé sur les algorithmes génétiques pour résoudre le problème de l'usurpation d'adresse IP, également connu sous le nom de « IP spoofing ». L'usurpation d'adresse IP est une technique couramment utilisée par les attaquants pour dissimuler leur identité en falsifiant l'adresse IP source d'un paquet réseau. Pour résoudre ce problème, nous utilisons une approche basée sur les algorithmes génétiques, qui sont des techniques d'optimisation inspirées par la théorie de l'évolution. Notre système utilise une représentation des solutions sous forme de structures chromosomiques, qui encodent les différentes combinaisons possibles d'adresses IP. Ces structures chromosomiques sont ensuite soumises à des opérations génétiques telles que la sélection, le croisement et la mutation, afin de générer une population de solutions potentielles.

ملخص

تركز هذه الرسالة على تطوير نظام كشف التسلل يعتمد على الخوارزميات الجينية لحل مشكلة انتحال عنوان IP، والمعروف أيضًا باسم انتحال IP. انتحال عنوان IP هو أسلوب شائع يستخدمه المهاجمون لإخفاء هويتهم عن طريق انتحال عنوان IP المصدر لحزمة الشبكة. لحل هذه المشكلة، نستعمل نهجًا يعتمد على الخوارزميات الجينية، وهي تقنيات تحسين مستوحاة من نظرية التطور. يستخدم نظامنا تمثيلاً للحلول في شكل هياكل كروموسومية، والتي تقوم بترميز المجموعات المختلفة الممكنة لعناوين IP. تخضع هذه الهياكل الصبغية بعد ذلك لعمليات وراثية مثل الانتقاء والتقاطع والطفرة ، من أجل توليد مجموعة من الحلول المحتملة.

TABLE DES MATIERES

Chapitre 2: Systèmes de détection d'intrusion

Chapitre 3: Algorithmes génétiques

Chapitre 4: Implémentation

TABLE DES FIGURES

Introduction Générale

Introduction Générale:

La sécurité des systèmes informatiques est devenue une préoccupation majeure dans notre société numérique en constante évolution. Les attaques informatiques, qu'elles soient ciblées ou généralisées, représentent une menace sérieuse pour la confidentialité, l'intégrité et la disponibilité des données sensibles. Dans ce contexte, les systèmes de détection d'intrusion (IDS) jouent un rôle crucial pour identifier et prévenir les activités malveillantes au sein des réseaux informatiques.

Ce projet de mémoire se concentre spécifiquement sur l'utilisation des algorithmes génétiques (AG) dans les systèmes de détection d'intrusion. Les AG sont une classe d'algorithmes inspirés de la théorie de l'évolution qui permettent de résoudre des problèmes complexes d'optimisation et de recherche. Leur application dans le domaine de la sécurité informatique offre des perspectives prometteuses pour améliorer la détection des intrusions et renforcer la résilience des réseaux.

Ce rapport se divise en quatre chapitres qui abordent différentes facettes de notre projet. Dans le premier chapitre, nous présenterons une vue d'ensemble de la cybersécurité, en mettant en évidence les principales menaces et les enjeux auxquels sont confrontés les systèmes informatiques de nos jours. Nous examinerons également les concepts fondamentaux de la sécurité informatique, les principes de base de la protection des réseaux et les différents outils et techniques utilisés pour contrer les attaques.

Le deuxième chapitre se concentrera sur les systèmes de détection d'intrusion (IDS). Nous examinerons leur rôle essentiel dans la détection précoce des activités malveillantes et la prévention des atteintes à la sécurité des réseaux. Nous discuterons des différentes approches utilisées par les IDS, telles que la détection basée sur les signatures, la détection basée sur les anomalies et la détection basée sur les comportements. Nous analyserons également les défis auxquels sont confrontés les IDS traditionnels et la nécessité de nouvelles approches pour renforcer leur efficacité.

Dans le troisième chapitre, nous plongerons dans les détails des algorithmes génétiques. Nous expliquerons leur structure, leur fonctionnement et les principes fondamentaux qui guident leur utilisation. Nous nous concentrerons particulièrement sur leur rôle dans les systèmes de détection d'intrusion, en présentant une structure de chromosome adaptée à la détection des activités malveillantes et une fonction de fitness pour évaluer la qualité des solutions proposées.

Enfin, dans le quatrième chapitre, nous présenterons l'implémentation de notre application basée sur les algorithmes génétiques pour la détection d'intrusion. Nous décrirons l'architecture du système, les étapes de développement, les choix technologiques et les résultats obtenus. Nous évaluerons également les performances de notre application et discuterons des perspectives d'amélioration et des travaux futurs.

Ce projet de mémoire vise à appliquer et exploiter les avantages des algorithmes génétiques dans les systèmes de détection d'intrusion. En combinant les connaissances en cybersécurité, les concepts des IDS et les principes des AG, nous fournissons une solution efficace pour la détection précoce des intrusions et la protection des réseaux contre les attaques malveillantes.

Chapitre 1

Cybersécurité

1- Introduction:

La cybersécurité désigne l'ensemble des technologies, des processus et des pratiques conçus pour protéger les réseaux, les appareils, les logiciels et les données. Ceux-ci sont protégés des attaques, dommages ou accès non autorisé.

2- Cybersécurité:

La cybersécurité est le processus de protection des informations électroniques par l'atténuation des risques et des vulnérabilités de l'information. Ces risques peuvent inclure l'accès, l'utilisation, la divulgation, l'interception ou la destruction non autorisés des données. Les données peuvent inclure les informations confidentielles des entreprises ou des utilisateurs individuels.

La cybersécurité couvre un large éventail d'activités, notamment **:**

- **Le contrôle d'accès :** Il vous permet de vous assurer que seuls les utilisateurs autorisés peuvent avoir accès aux systèmes, aux données et aux ressources.
- **L'identité et l'authentification :** Elles vous aident à vérifier l'identité et les dispositifs de l'utilisateur avant de lui accorder l'accès aux systèmes, aux données et aux ressources.
- **Sécurité des données :** Elle protège les données contre toute utilisation, accès, divulgation, interception ou destruction non autorisés.
- **Réponse aux incidents :** Elle vous permet d'identifier, de contenir et d'éradiquer les cyber-menaces.
- **Gestion des risques :** Il s'agit du processus d'évaluation, d'identification et de hiérarchisation des risques pour les actifs, les systèmes et les données de l'organisation.

2.2- Inconvénients de la cybersécurité:

La cybersécurité présente plusieurs inconvénients, tels que :

- Coût élevé.
- Complexité.
- Faux sens de sécurité.

- Lenteur.
- Réactivité.
- Menaces internes.

2.3- Avantages de la cybersécurité:

La cybersécurité présente plusieurs avantages, tels que :

- Protection des données.
- Maintien de la confidentialité.
- Maintien de l'intégrité des données.
- Continuité des activités.
- Réduction des coûts.
- Amélioration de la confiance.

2.4- Types de cybersécurité:

- La sécurité du réseau.
- La sécurité des applications.
- La sécurité des données.
- La sécurité du Cloud.
- La sécurité de l'Internet des objets (IoT).

2.5- Principes de sécurité de l'information:

- **La confidentialité:**

Au sens non technique, la confidentialité équivaut au terme vie privée que nous utilisons dans la vie quotidienne. Techniquement, la confidentialité est définie comme la caractéristique à laquelle seules les personnes autorisées ont accès à la ressource. Ainsi, c'est l'attribut de dissimuler les ressources des entités non autorisées. En termes de sécurité de l'information, c'est l'assurance que l'information ne serait accessible qu'aux personnes autorisées. Son interprétation peut être étendue de plusieurs manières, qui sont les suivantes :

• Les informations seront stockées sous une forme autorisée uniquement, que ce soit physique ou électronique, et seules les personnes autorisées auront accès aux magasins de données.

• Les informations seront stockées dans le format autorisé uniquement.

• les informations seront disponibles via des supports autorisés seul.

Dans le cas de notre communication via des réseaux informatiques, il existe deux aspects majeurs de la confidentialité. Premièrement, la confidentialité des communications, c'est-à-dire uniquement ceux qui sont autorisés doit être conscient que la communication est en cours ou a eu lieu.

Deuxièmement, la confidentialité du contenu de la communication, c'est-à-dire que seules les personnes autorisées devraient pouvoir voir ce qui a été communiqué ou est en cours de communication. **[1]**

- **L'intégrité:**

L'accès non autorisé à une ressource peut entraîner le problème de mal l'utiliser ou le corrompre. En sécurité de l'information, l'intégrité est la caractéristique que les informations ne sont pas modifiées de manière non autorisée. Ainsi, d'une part, c'est l'assurance que l'information est en son état d'origine sans aucune corruption. D'autre part, c'est aussi l'information que toute modification requise a été faite ou seront effectués en utilisant une approche autorisée, par exemple uniquement par des personnes et uniquement par des moyens autorisés. **[1]**

- **La disponibilité:**

La ressource doit être disponible pour les utilisateurs autorisés en cas de besoin.

La disponibilité est l'attribut par lequel le système fournit les informations aux utilisateurs ou spectateurs autorisés chaque fois qu'ils en ont besoin, généralement tout le temps. La disponibilité de l'information exige indirectement que :

• Le système stockant les données/informations doit être disponible.

• Le système utilisé pour récupérer les données/informations doit être disponible.

• Le système utilisé pour mettre à jour les données/informations doit être disponible.

• Un plan de sauvegarde doit être disponible et, en cas d'échec, le système de sauvegarde doit pouvoir prendre le contrôle immédiatement. **[1]**

2.6- Importance de cybersécurité:

La cybersécurité est essentielle pour:

- protéger les données contre les pertes, les vols ou les fuites. Les attaques informatiques peuvent avoir des conséquences graves, y compris la perte de données client, la divulgation de données confidentielles, la perte de propriété intellectuelle, etc.
- assurer la continuité des activités en cas d'attaque ou de panne de système.
- protéger la réputation d'une entreprise et maintenir la confiance de ses parties prenantes.
- assurer la conformité réglementaire et éviter les conséquences négatives.

3- Cyberattaque:

Une cyberattaque est une action malveillante visant à compromettre la sécurité des systèmes informatiques, des réseaux, des appareils électroniques ou des données numériques. Elle est généralement menée par des individus malintentionnés, des groupes de hackers ou même des États-nations dans le but de causer des dommages, de voler des informations confidentielles, de perturber les opérations ou d'obtenir un avantage illégitime.

3.1- Types d'attaques:

Il existe essentiellement 06 types d'attaques :

3.1.1- Attaque réseau:

Une attaque réseau est une tentative d'obtenir un accès non autorisé au réseau d'une organisation, dans le but de voler des données ou d'effectuer d'autres activités malveillantes.

Il existe deux principaux types d'attaques réseau :

- **Passif :** les attaquants accèdent à un réseau et peuvent surveiller ou voler des informations sensibles, mais sans apporter aucune modification aux données, en les laissant intactes.
- **Actif :** les attaquants obtiennent non seulement un accès non autorisé, mais modifient également les données, en les supprimant, en les cryptant ou en les endommageant d'une autre manière.

3.1.2- Attaque applicative:

Une attaque d'application consiste en l'accès de cybercriminels à des zones non autorisées. Les attaquants commencent le plus souvent par jeter un coup d'œil à la couche d'application, à la recherche de vulnérabilités d'application écrites dans le code

3.1.3- Déni de Service (Dos):

Une attaque par déni de service (Dos) consiste à générer un trafic énorme sur un service afin d'épuiser les ressources et la bande passante d'un réseau informatique ou d'un serveur. Le service en question se retrouve alors saturé par de nombreuses requêtes utilisateurs et devient inaccessible.

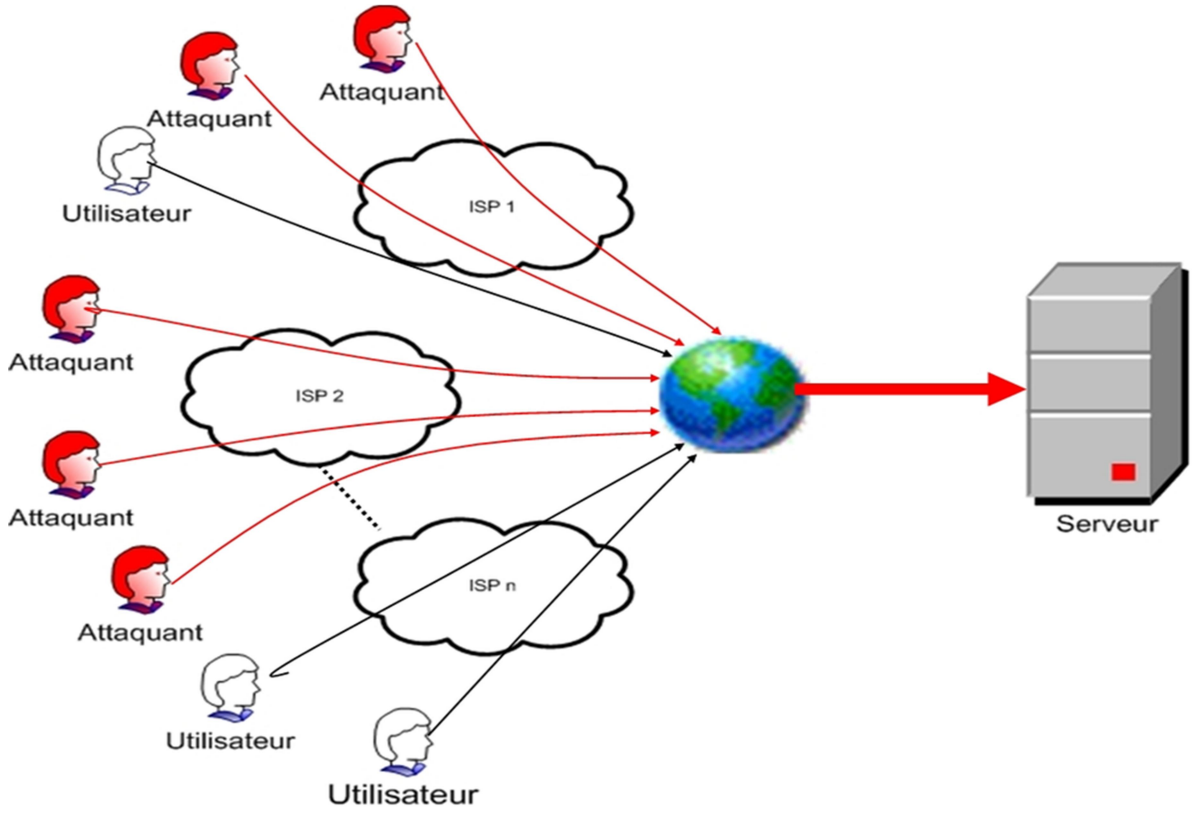

Figure 1.1: Attaque par dénis de service

3.1.4- Attaque de données:

Les données transportées par le protocole applicatif peuvent constituer une menace pour l'intégrité du système qui les reçoit. Les principales attaques de ce type, on retrouve : virus, vers, applet Java, chevaux de Troie etc. désignés par des codes malveillants ou Malware.

3.1.5-Attaque Phishing:

Une attaque par hameçonnage consiste à envoyer des e-mails ou des messages frauduleux pour inciter les utilisateurs à fournir des informations confidentielles, telles que des noms d'utilisateur, des mots de passe ou des informations bancaires.

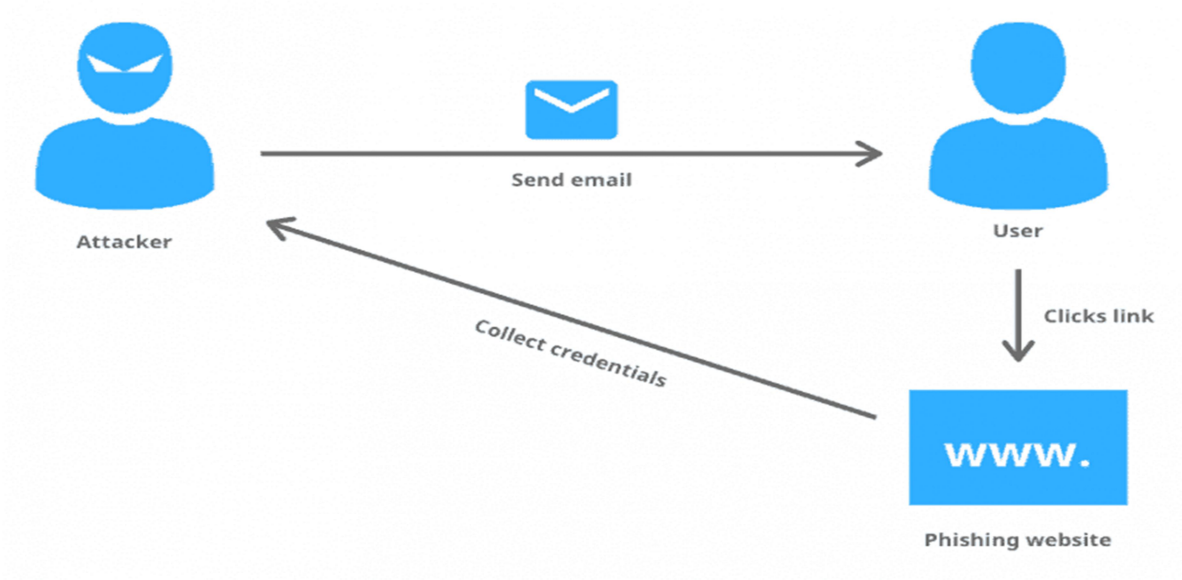

Figure 1.2: Attaque de Phishing.

3.1.6-TCP SYN Flood Attaques:

Un attaquant exploite l'utilisation de l'espace tampon lors du handshake d'initialisation de session TCP. La machine de l'attaquant inonde de demandes de connexion la petite file d'attente de traitement du système cible, mais elle ne réagit pas lorsque le système cible répond à ces demandes. Le système cible se met alors à temporiser en attendant la réponse de la machine de l'attaquant, ce qui fait planter le système ou le rend inutilisable lorsque la file d'attente de connexion se remplit. [2]

4-Les Dispositifs de protection:

Les mécanismes de protection dans la cybersécurité sont des mesures et des techniques utilisées pour prévenir, détecter et répondre aux attaques et aux menaces informatiques. Voici quelques-uns des mécanismes de protection couramment utilisés dans la cybersécurité :

- **Pare-feu (Firewall) :**

Un pare-feu est un dispositif ou un logiciel qui contrôle le trafic réseau en filtrant les connexions et en bloquant les communications non autorisées. Il peut être configuré pour

surveiller et contrôler les accès entrants et sortants, en fonction de règles de sécurité prédéfinies.

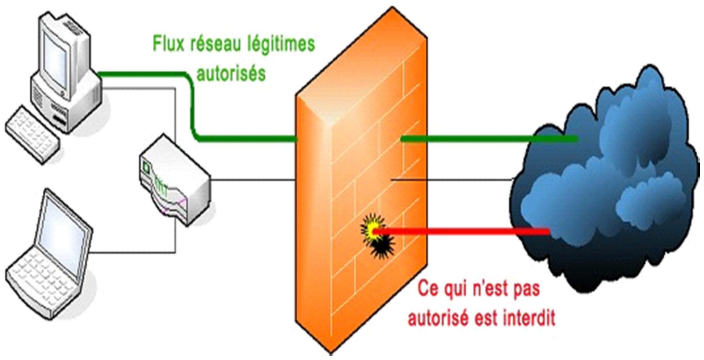

Figure 1.3: l'emplacement d'un pare-feu dans un réseau

- **Systèmes de détection et de prévention d'intrusion (IDS/IPS) :**

Les systèmes de détection d'intrusion (IDS) et de prévention d'intrusion (IPS) surveillent le trafic réseau à la recherche de comportements anormaux et d'activités suspectes. Ils détectent les intrusions potentielles et prennent des mesures pour les bloquer ou les prévenir.

- **Logiciel antivirus et anti-malware :**

Les logiciels antivirus et anti-malware sont conçus pour détecter, prévenir et éliminer les logiciels malveillants tels que les virus, les vers, les chevaux de Troie, les ransomwares, etc. Ils analysent les fichiers et les activités du système à la recherche de signatures ou de comportements malveillants.

- **Cryptographie :**

La cryptographie est utilisée pour protéger la confidentialité et l'intégrité des données en les chiffrant. Elle utilise des algorithmes de chiffrement pour transformer les données en un format illisible, sauf pour les destinataires autorisés qui possèdent la clé de déchiffrement appropriée.

- **Serveur proxy :**

Un serveur proxy appelé aussi serveur mandataire, est un composant logiciel informatique qui joue le rôle de l'intermédiaire entre deux machines pour surveiller leurs échanges.

La plupart du temps le serveur proxy est utilisé pour le web, il s'agit alors d'un proxy HTTP. Toutefois il peut exister des serveurs proxy pour chaque protocole applicatif (FTP, ...etc.)

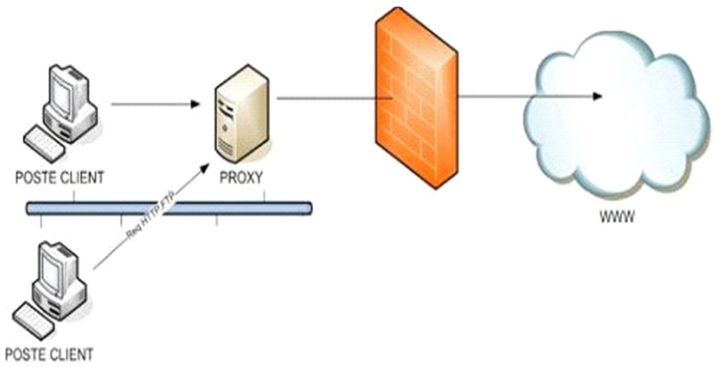

Figure 1.4: l'emplacement d'un proxy dans un réseau.

5- Evaluation des risques:

L'évaluation des risques est un processus d'identification et d'analyse des risques pour déterminer les mesures à prendre pour les atténuer ou les éliminer. Dans le domaine de la sécurité de l'information, l'évaluation des risques est essentielle pour protéger les informations sensibles et les actifs informatiques contre les menaces et les vulnérabilités. Voici les étapes clés de l'évaluation des risques dans le domaine de la sécurité de l'information :

- **Identification des actifs :** Cette étape consiste à identifier tous les actifs informatiques tels que les données, les applications, les systèmes, les réseaux, etc. qui sont essentiels à l'organisation et qui peuvent être la cible d'attaques.
- **Identification des menaces :** Cette étape consiste à identifier les menaces qui pourraient affecter les actifs informatiques. Les menaces peuvent être internes ou externes et peuvent inclure des cyberattaques, des catastrophes naturelles, des erreurs humaines, etc.
- **Évaluation des vulnérabilités :** Cette étape consiste à identifier les vulnérabilités des actifs informatiques. Les vulnérabilités peuvent inclure des failles de sécurité, des lacunes dans les politiques de sécurité, des défauts dans les systèmes, etc.
- **Évaluation de la probabilité :** Cette étape consiste à évaluer la probabilité que les menaces exploiteront les vulnérabilités pour affecter les actifs informatiques**.**
- **Évaluation de l'impact :** Cette étape consiste à évaluer l'impact potentiel sur l'organisation si les menaces exploitent les vulnérabilités pour affecter les actifs informatiques.

- **Analyse des risques :** Cette étape consiste à combiner les résultats des étapes précédentes pour déterminer les risques et les priorités pour l'organisation.
- **Élaboration d'un plan de gestion des risques :** Cette étape consiste à élaborer un plan pour gérer les risques identifiés. Le plan peut inclure des mesures pour atténuer ou éliminer les risques, des mesures pour détecter les attaques et y répondre, ainsi que des mesures pour récupérer après une attaque.

5.1- Types d'audits:

Il existe plusieurs types d'audits dans le domaine de la sécurité de l'information, chacun ayant ses propres objectifs et méthodologies. Voici quelques exemples de types d'audits :

5.1.1- Audit de conformité :

Cet audit vise à évaluer si l'organisation respecte les exigences réglementaires et légales relatives à la sécurité de l'information. Cela peut inclure des normes telles que le Règlement général sur la protection des données (RGPD), la norme ISO/IEC 27001, etc.

5.1.2- Audit de sécurité :

Cet audit vise à évaluer les contrôles de sécurité en place pour protéger les actifs informatiques de l'organisation. Cela peut inclure des tests de pénétration pour évaluer la résistance du système aux attaques, des évaluations de vulnérabilité, etc.

5.1.3- Audit de gestion de la continuité d'activité :

Cet audit vise à évaluer la capacité de l'organisation à gérer les incidents et à maintenir la continuité des opérations en cas de catastrophe naturelle, de cyberattaque ou d'autres perturbations majeures**.**

5.1.4- Audit de conformité à la politique de sécurité de l'information :

Cet audit vise à évaluer si l'organisation respecte les politiques internes relatives à la sécurité de l'information. Cela peut inclure des évaluations de la gestion des accès, de la classification des données, etc.

5.1.5- Audit de conformité à la norme ISO/IEC 27001 :

Cet audit vise à évaluer si l'organisation respecte les exigences de la norme ISO/IEC 27001, qui est une norme internationale de gestion de la sécurité de l'information.

5.2- Outils d'audit de sécurité:

Il existe de nombreux outils d'audit de sécurité disponibles qui peuvent aider les professionnels de la sécurité à évaluer la sécurité des systèmes informatiques et des applications. Voici quelques exemples d'outils couramment utilisés :

- **Nmap** : un scanner de ports qui peut être utilisé pour identifier les services et les applications qui s'exécutent sur un système.
- **Nessus** : un outil de vulnérabilité largement utilisé qui peut scanner un réseau pour identifier les vulnérabilités connues et les failles de sécurité.
- **Open VAS** : un scanner de vulnérabilité open source qui peut détecter les vulnérabilités connues dans les systèmes et les applications.
- **Metasploit** : un Framework d'exploitation qui peut être utilisé pour tester les vulnérabilités et les failles de sécurité.
- **Wireshark** : un outil de capture et d'analyse de paquets qui peut être utilisé pour surveiller le trafic réseau et détecter les activités suspectes.
- **Aircrack-ng** : un ensemble d'outils pour tester la sécurité des réseaux sans fil, y compris la détection de réseaux sans fil cachés, la capture de paquets et la récupération de clés de chiffrement.
- **John the Ripper** : un outil de craquage de mot de passe qui peut être utilisé pour tester la force des mots de passe.

6- Différence entre pare-feu et système de détection d'intrusion:

Les pare-feux et les systèmes de détection d'intrusion (IDS) sont deux types de systèmes de sécurité informatique qui jouent des rôles différents dans la protection d'un réseau.

Un pare-feu est un système de sécurité qui permet de contrôler les flux de trafic entrants et sortants sur un réseau. Il peut être configuré pour autoriser ou bloquer certains types de trafic en fonction de règles spécifiques, telles que des adresses IP, des ports ou des protocoles. Les pare-feux sont souvent utilisés comme une première ligne de défense contre les attaques, pour empêcher les tentatives d'accès non autorisées à un réseau.

Un système de détection d'intrusion (IDS) est un système de sécurité qui détecte les tentatives d'intrusion ou les comportements malveillants sur un réseau. Il surveille le trafic réseau pour détecter des anomalies et signaler les activités suspectes. Les IDS peuvent être basés sur des règles, des modèles ou des techniques d'apprentissage automatique pour détecter les anomalies. Les IDS sont souvent utilisés pour détecter les tentatives d'attaques qui ont réussi à contourner les pare-feux.

Conclusion:

Dans ce chapitre, nous avons présenté un aperçu de la sécurité informatique dans un réseau et l'importance de mettre en place une politique de sécurité en traçant les besoins et les objectifs recherchés afin de remédier aux menaces constantes que subit un réseau informatique. Ces menaces se manifestent généralement sous la forme d'attaques informatiques que nous avons illustrées afin de montrer l'intensité du danger. Enfin, nous avons proposé quelques solutions existantes afin de se protéger et réduire les risques.

Figure 1.5: La cybersécurité.

Chapitre 2

Systèmes de Détection d'Intrusion

1- Introduction:

Les systèmes de détection d'intrusion (IDS, pour Intrusion Detection Systems) sont des outils essentiels de sécurité informatique utilisés pour surveiller et détecter les activités malveillantes ou suspectes au sein d'un réseau ou d'un système informatique. Leur objectif principal est d'identifier les intrusions, les tentatives d'exploitation, les comportements anormaux et les violations de la politique de sécurité.

2- Définition de l'intrusion:

L'intrusion, en matière de sécurité informatique, est l'action de pénétrer illégalement dans un système informatique, un réseau ou un service, avec l'intention de compromettre la confidentialité, l'intégrité ou la disponibilité des informations qui y sont stockées ou transitent.

3- Définition d'un système de détection d'intrusion:

Un système de détection d'intrusion (IDS, pour Intrusion Détection System en anglais) est un dispositif ou un logiciel de sécurité informatique qui surveille le trafic réseau ou les activités des utilisateurs pour détecter les tentatives d'intrusion ou les comportements anormaux.

- **Faux positif :** une alerte d'un IDS, mais pas une véritable attaque.
- **Faux négatif :** une intrusion réelle qui n'a pas été détectée par l'IDS.

4- Comment fonctionne un IDS ?

Le fonctionnement d'un système de détection d'intrusion (IDS) peut varier en fonction de la technologie et de l'architecture utilisées. Cependant, en général, voici les étapes de base du fonctionnement d'un IDS :

- **Surveillance des événements :** L'IDS surveille les événements sur le réseau ou sur l'hôte protégé, tels que les connexions entrantes et sortantes, les paquets de données, les connexions à des services réseau, les requêtes de protocoles, etc.
- **Analyse des événements :** L'IDS analyse les événements pour identifier les comportements suspects ou malveillants, tels que les activités de scan de ports, les tentatives d'authentification non autorisées, les anomalies dans les protocoles de communication, les tentatives de modification de fichiers système, etc.

- **Détection des intrusions :** Si l'IDS détecte une activité suspecte, il peut envoyer une alerte au personnel de sécurité ou déclencher une action préventive, telle que le blocage de l'adresse IP d'où provient l'activité suspecte, ou la désactivation du compte d'utilisateur concerné.
- **Journalisation des événements :** L'IDS conserve un historique des événements de sécurité afin que les administrateurs puissent analyser les activités passées et identifier les tendances et les schémas d'attaque.
- **Mise à jour des règles et des signatures :** Les IDS utilisent des règles et des signatures pour identifier les activités malveillantes. Ces règles et signatures sont régulièrement mises à jour pour inclure de nouvelles menaces et les techniques d'attaques les plus récentes.

5- Types de systèmes de détection d'intrusion:

- **IDS basé sur les signatures :** Ce type d'IDS utilise une base de données de signatures ou de modèles d'attaque connus pour identifier les activités malveillantes. Il compare le trafic réseau ou les événements système à la base de données de signatures et génère des alertes ou prend des mesures si une correspondance est trouvée.
- **IDS basé sur le comportement :** L'IDS basé sur le comportement se concentre sur la surveillance du comportement des utilisateurs et des entités au sein d'un réseau ou d'un système. Il établit des modèles de comportement attendus pour différents utilisateurs ou entités et déclenche des alarmes si les activités diffèrent de ces modèles. Il aide à détecter les menaces internes ou les comptes d'utilisateurs compromis.
- **IDS basé sur le réseau (NIDS) :** NIDS surveille le trafic réseau en temps réel et analyse les paquets qui circulent dans le réseau. Il détecte les modèles suspects, les signatures ou les anomalies qui peuvent indiquer une intrusion. NIDS peut fonctionner en mode promiscues (surveillance passive) ou en mode inline (blocage actif du trafic). [3]

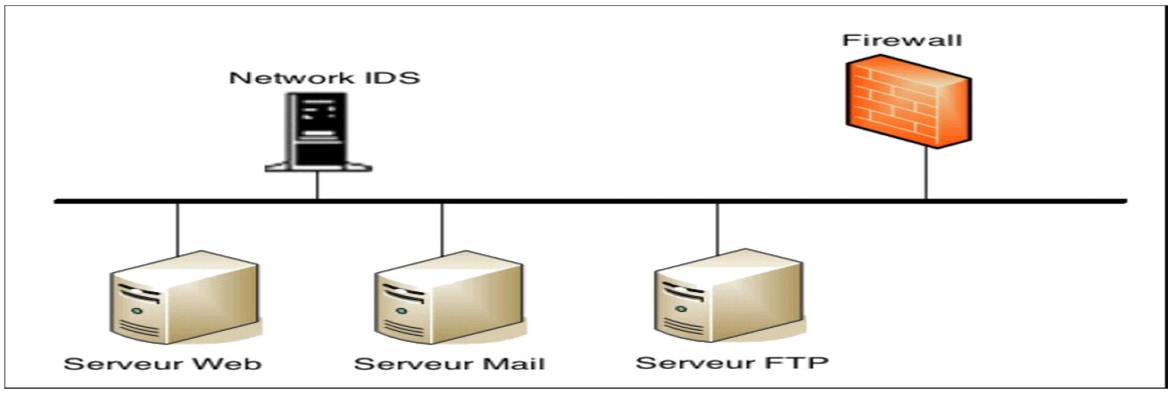

Figure 2.1 : Système de détection d'intrusion réseau.

- **IDS basé sur l'hôte (HIDS) :** HIDS est installé sur des systèmes d'hôtes individuels et surveille les journaux système, l'intégrité des fichiers et d'autres activités liées à l'hôte. Il recherche des signes d'accès non autorisé, de modifications de fichiers critiques ou de comportement anormal au niveau de l'hôte. [3]

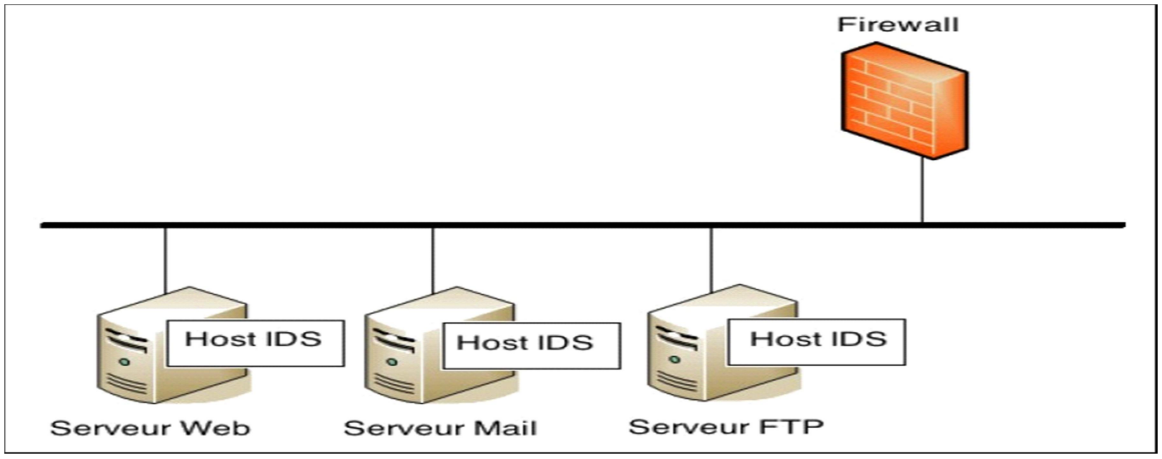

Figure 2.2 : Système de détection d'intrusion hôte.

6- Caractéristiques des IDS:

Parmi les caractéristiques souhaitables trouvées dans un système de détection d'intrusion nous pouvons citer :

- Résister aux tentatives de corruption, c'est-à-dire, il doit pouvoir détecter s'il a subi lui-même une modification indésirable.
- Utiliser un minimum de ressources de système sous surveillance.
- S'adapter au cours du temps aux changements du système surveillé et du comportement des utilisateurs.

- Etre facilement configurable pour implémenter une politique de sécurité spécifique d'un réseau.

7- L'architecture d'un IDS:

Cette section décrit les trois composants qui constituent classiquement un système de détection d'intrusions. La Figure illustre les interactions entre ces trois composants

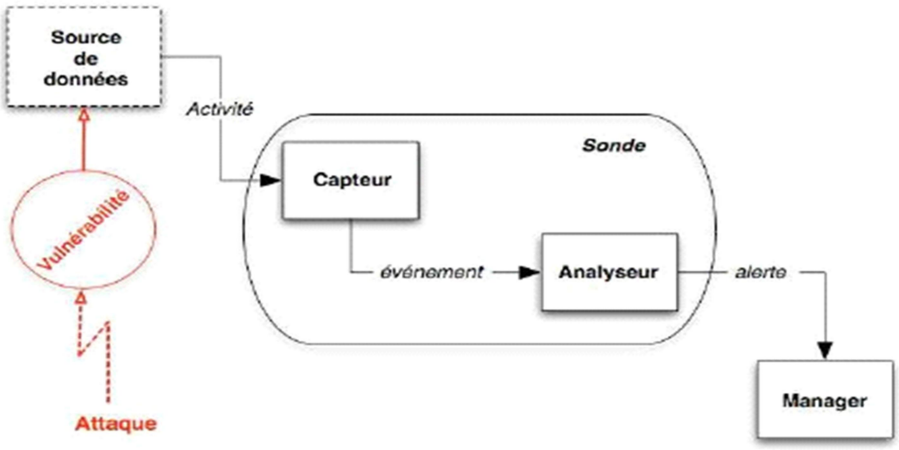

Figure 2.3: L'architecture d'un IDS.

7.1- Capteur:

Capteurs (Sensors) : Les capteurs sont les composants qui collectent les données pour l'analyse. Dans le cas d'un IDS réseau (NIDS), les capteurs sont positionnés sur le réseau et surveillent le trafic en temps réel. Ils peuvent être déployés de manière distribuée sur plusieurs emplacements pour une couverture étendue. Dans le cas d'un IDS hôte (HIDS), les capteurs sont installés directement sur les hôtes individuels pour surveiller les activités et les événements sur ces hôtes.

7.2- Analyseur:

L'objectif de l'analyseur est de déterminer si le flux d'événements fourni par le capteur contient des éléments caractéristiques d'une activité malveillante.

7.3- Gestionnaire :

Le manager collecte les alertes produites par le capteur, les met en forme et les présente à l'opérateur. Éventuellement, le manager est chargé de la réaction à adopter qui peut être :

- Confinement de l’attaque, qui a pour but de limiter les effets de l’attaque.
- Eradication de l’attaque, qui tente d’arrêter l’attaque.
- Recouvrement, qui est l’étape de restauration du système dans un état sain.
- Diagnostic, qui est la phase d’identification du problème. [4]

8- Mise en place d'un IDS:

8.1- Le positionnement de l'IDS:

Il existe plusieurs endroits stratégiques où il convient de placer un IDS. Le schéma suivant illustre un réseau local ainsi que les trois positions que peut y prendre un IDS :

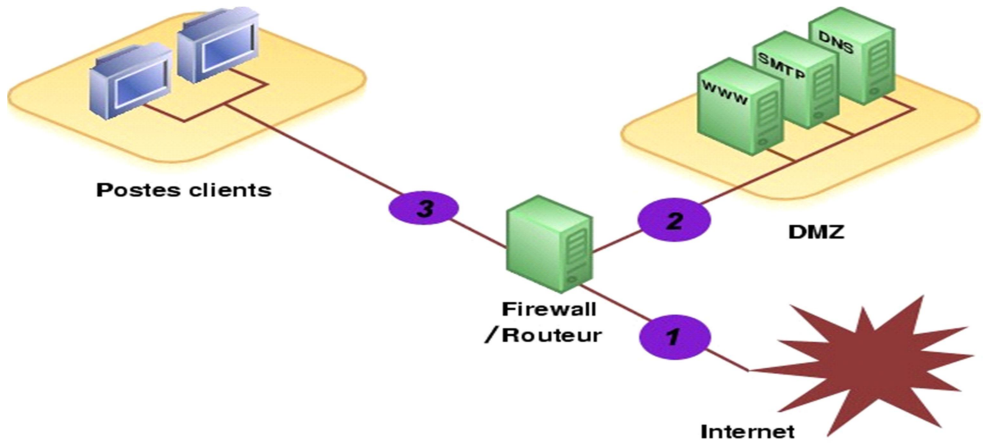

Figure 2.4: La position des IDS.

- **Position (1):** Sur cette position, l'IDS va pouvoir détecter l'ensemble des attaques frontales, provenant de l'extérieur, en amont du firewall. Ainsi, beaucoup d'alertes seront remontées ce qui rendra les logs difficilement consultables.
- **Position (2):** Si l'IDS est placé sur la DMZ, il détectera les attaques qui n'ont pas été filtrées par le firewall et qui relèvent d'un certain niveau de compétence. Les logs seront ici plus clairs à consulter puisque les attaques ne seront pas recensées.
- **Position (3):** L'IDS peut ici rendre compte des attaques internes, provenant du réseau local de l'entreprise. Il peut être judicieux d'en placer un à cet endroit étant donné le fait que 80% des attaques proviennent de l'intérieur. De plus, si des

trojans ont contaminé le parc informatique (navigation peu méfiante sur internet) ils pourrontêtre ici facilement identifiés pour être ensuite éradiqués.

9- Critères pour tester un IDS:

Lors de la mise en place d'un système de détection d'intrusion (IDS), il est important de tester son efficacité et ses performances pour s'assurer qu'il fonctionne conformément aux attentes. Voici quelques critères importants à prendre en compte lors du test d'un IDS :

- **Détection des attaques connues :** Vérifiez si l'IDS est capable de détecter les attaques connues pour lesquelles il dispose de signatures ou de modèles de détection. Effectuez des tests en utilisant des exemples d'attaques connues et vérifiez si l'IDS génère des alertes appropriées.
- **Détection des attaques inconnues :** Testez la capacité de l'IDS à détecter les attaques inconnues ou émergentes. Utilisez des techniques d'attaques avancées ou des comportements anormaux pour évaluer la capacité de l'IDS à détecter ces activités suspectes.
- **Taux de faux positifs :** Évaluez le taux de faux positifs de l'IDS, c'est-à-dire le nombre d'alertes générées par l'IDS qui sont en réalité des activités légitimes. Un taux de faux positifs élevé peut entraîner une surcharge d'alertes et une perte d'efficacité de l'IDS.
- **Taux de faux négatifs :** Évaluez le taux de faux négatifs de l'IDS, c'est-à-dire les attaques ou activités malveillantes qui ne sont pas détectées par l'IDS. Un taux de faux négatifs élevé peut indiquer des lacunes dans la capacité de détection de l'IDS.
- **Performances et scalabilité :** Testez les performances de l'IDS pour vous assurer qu'il peut gérer la charge de trafic réseau prévue. Évaluez également sa capacité à s'adapter à l'évolution des besoins et de la taille de l'infrastructure.
- **Capacité de réaction aux incidents :** Testez la capacité de l'IDS à générer des alertes en temps réel et à fournir des informations détaillées sur les incidents de sécurité détectés. Évaluez également la capacité de l'IDS à prendre des mesures automatiques ou à fournir des recommandations pour atténuer les incidents détectés.

- **Intégration avec d'autres outils de sécurité :** Si l'IDS est intégré à d'autres outils de sécurité tels que les systèmes de gestion des journaux (SIEM) ou les systèmes de prévention d'intrusion (IPS), assurez-vous que l'intégration fonctionne correctement et que les informations sont partagées de manière efficace entre les différentes solutions

10- Les méthodes de détections :

Les systèmes de détection d'intrusion (IDS) utilisent différentes méthodes pour détecter les attaques et les activités suspectes. Voici les principales méthodes de détection utilisées par les IDS :

10.1- Détection basée sur l'anomalie :

Dans ce mode de fonctionnement, l'IDS analyse le trafic réseau ou les activités des hôtes pour identifier les comportements anormaux ou inhabituels. Il établit un profil du trafic ou des activités normales, puis compare en temps réel les modèles observés avec ce profil. Lorsqu'une déviation significative est détectée, l'IDS génère une alerte pour signaler une possible intrusion ou une activité suspecte.

10.2- Détection basée sur les signatures:

Dans ce mode de fonctionnement, l'IDS utilise des signatures préalablement définies pour identifier les attaques connues. Il compare le trafic réseau ou les activités des hôtes avec une base de signatures d'attaques connues et génère des alertes lorsque des correspondances sont détectées. Les signatures peuvent être basées sur des modèles de paquets réseau, des séquences d'octets spécifiques, des comportements de processus, etc.

11- Techniques anti-IDS:

Les techniques anti-IDS, également appelées techniques d'évasion, sont utilisées par les attaquants pour éviter la détection des systèmes de détection d'intrusion (IDS). Voici quelques-unes des techniques couramment utilisées :

- **Fragmentation des paquets :** Les attaquants peuvent fragmenter les paquets réseau afin de rendre plus difficile la reconstitution et l'analyse par les IDS. En fragmentant les paquets, ils peuvent éviter la détection des attaques ou rendre les signatures d'attaques plus difficiles à identifier.
- **Encodage ou compression des données :** Les attaquants peuvent utiliser des techniques d'encodage ou de compression pour masquer le contenu malveillant des paquets. Cela peut rendre difficile la détection des attaques par les IDS qui se basent sur l'analyse du contenu des paquets.
- **Utilisation de tunnels ou de protocoles chiffrés :** Les attaquants peuvent utiliser des protocoles ou des tunnels chiffrés, tels que les connexions VPN ou le protocole SSL/TLS, pour masquer leurs activités et éviter la détection par les IDS. Le chiffrement du trafic rend difficile l'analyse du contenu des paquets par les IDS.

12- Les limites de l'IDS

Comme tout système informatique, les IDS ont des limites. On peut en citer :

- **Pollution/surcharge** : Les IDS peuvent être pollués ou surchargés, par exemple par la génération d'un trafic important (le plus difficile et lourd possible à analyser). Une quantité importante d'attaques peut également être envoyée afin de surcharger les alertes de l'IDS. Des conséquences possibles de cette surcharge peuvent être la saturation de ressources (disque, CPU, mémoire), la perte de paquets, le déni de servicepartiel ou total … .
- **Consommation de ressources** : outre la taille des fichiers de logs (de l'ordre du Go), la détection d'intrusion est excessivement gourmande en ressources. En effet un système NIDS doit générer des journaux des comptes-rendus d'activité anormale ou douteuse sur le réseau.
- **Perte de paquets (limitation des performances)** : les vitesses de transmission sont parfois telles qu'elles dépassent largement la vitesse d'écriture des disques durs, ou même la vitesse de traitement des processeurs. Il n'est donc pas rare que

des paquets ne soient pas traités par l'IDS, et que certains d'entre eux soient néanmoins reçus par la machine destinataire.

Vulnérabilité aux dénis de service : un attaquant peut essayer de provoquer un déni de service au niveau du système de détection d'intrusion, ou pire au niveau du système d'exploitation de la machine supportant l'IDS. Une fois que l'IDS est désactivé (« hors service »), l'attaquant peut tenter tout ce qui luiconvient.

Conclusion:

Dans ce chapitre, nous avons parlé sur les systèmes de détection d'intrusion et leur rôle essentiel dans la sécurité des systèmes informatiques et des réseaux. Leur objectif principal est de détecter les activités suspectes, les attaques et les comportements malveillants, afin de protéger les infrastructures et les données sensibles.

Chapitre 3

Algorithmes Génétiques

1-Introduction :

Les algorithmes génétiques (AG) constituent une classe d'algorithmes d'optimisation et de recherche qui s'inspirent de la théorie de l'évolution biologique pour résoudre des problèmes complexes. Ils ont été largement utilisés dans de nombreux domaines, y compris l'informatique, l'ingénierie, l'optimisation des processus et la science des données. Leur application la plus courante se trouve dans le domaine de l'intelligence artificielle, où ils offrent une approche puissante pour trouver des solutions optimales à des problèmes difficiles.

L'idée fondamentale derrière les algorithmes génétiques repose sur le principe de sélection naturelle et de reproduction. Tout comme dans la nature, où les espèces évoluent et s'adaptent à leur environnement au fil du temps, les AG cherchent à optimiser des solutions en évaluant, en sélectionnant et en croisant des individus représentant ces solutions. Les individus sont généralement représentés sous forme de chromosomes, composés de gènes qui codent les caractéristiques des solutions potentielles.

Le processus d'évolution dans les AG se déroule en plusieurs étapes clés. Tout d'abord, une population initiale d'individus est générée de manière aléatoire ou selon des critères spécifiques. Ensuite, ces individus sont évalués en utilisant une fonction de fitness qui mesure leur qualité par rapport à l'objectif de l'optimisation. Les individus les mieux adaptés sont alors sélectionnés pour la reproduction, où leurs gènes sont combinés et modifiés pour générer une nouvelle génération d'individus.

Cette nouvelle génération subit ensuite des processus de mutation et de croisement, qui introduisent des variations génétiques dans la population. Cela permet d'explorer de nouvelles régions de l'espace de recherche et d'éviter de rester bloqué dans des optima locaux. Ce processus d'évolution est répété sur plusieurs générations, avec une sélection et une reproduction basées sur les performances des individus, jusqu'à ce qu'une solution satisfaisante soit atteinte ou qu'un critère d'arrêt prédéfini soit rempli.

2-Algorithmes génétiques :

Les algorithmes génétiques sont des méthodes de recherche et d'optimisation inspirées du processus de sélection naturelle et de l'évolution biologique. Ils sont largement utilisés pour résoudre des problèmes complexes et trouver des solutions efficaces dans de nombreux domaines tels que l'ingénierie, la finance, l'intelligence artificielle, la biologie et bien d'autres.

Le concept central des algorithmes génétiques repose sur la notion d'évolution. Comme dans la nature, les algorithmes génétiques partent d'une population initiale d'individus (souvent représentés sous forme de chaînes de bits ou de vecteurs) et utilisent des opérations génétiques telles que la reproduction, la mutation et la recombinaison pour générer de nouvelles générations d'individus. Ces nouvelles générations sont soumises à un processus de sélection basé sur leur adaptation à un environnement donné, déterminé par une fonction d'évaluation ou de fitness.

3-Structure des algorithmes génétiques:

Les algorithmes génétiques sont mis en œuvre sous forme de structures de données de type chromosome.

Un algorithme génétique possède de nombreux paramètres, opérateurs et processus qui décident de son arrivée à une solution optimale. Une courte description des paramètres, des opérateurs et des processus est la suivante :

- **Fonction de fitness :** La fonction de fitness est la mesure de la qualité d'une solution particulière. La fonction fitness est utilisée pour déterminer la solution la plus optimale parmi un certain nombre de solutions dans une population.
- **Sélection :** Le processus de sélection dans les algorithmes génétiques est utilisé pour sélectionner la solution la plus optimale déterminée en utilisant la fonction de remise en forme. Les solutions qui ne sont pas optimale sont rejetées.
- **Crossover :** Le processus de croisement dans les algorithmes génétiques est utilisé pour échanger des caractéristiques entre deux solutions. Les paires de solutions pour échanger des caractéristiques sont sélectionnées au hasard et continuent d'échanger des caractéristiques, jusqu'à ce que une toute nouvelle génération de solutions est obtenue.
- **Mutation :** Le processus de mutation dans les algorithmes génétiques modifie certains bits aléatoires dans une solution. Le changement de bits se traduit par la diversité génétique des algorithmes mutés. [5]

4-Fonctionnement des algorithmes génétique :

Les algorithmes génétiques commencent le traitement en sélectionnant initialement une population aléatoire de chromosomes.

Chaque chromosome est composé d'un nombre fini de gènes, qui est prédéfini dans chaque implémentation Ces chromosomes sont les données représentant le problème.

Cette population initiale est raffinée en une population de chromosomes de haute qualité, où chaque chromosome satisfait une fonction de fitness prédéfinie. Selon les exigences de la solution nécessaire, les positions des différents gènes dans un chromosome sont codées sous forme de nombres, de bits ou de caractères. Chaque population est raffinée en appliquant la mutation, processus de croisement, d'inversion et de sélection. Le pseudo code générique d'un algorithme génétique tiré de est donné ci-dessous :

InitPopulation (P)
Forme physique(P)
Tandis que Max Génération Not Reached fait
pour i = 0 à facteur x faire
p1 = Sélection(P)
p2 = Sélection(P)
(o1, o2) = croisement(p1, p2)
Surpeuplement(p1, p2, o1, o2)
fin pour
pour i = 0 à d factor faire
p = Sélection(P)
Chute(P)
fin pour
pour i = 0 à m factor faire
p = Sélection(P)
Mutation(p)
fin pour
Forme physique(P)
fin tandis que
Sélection BestIndividual(P)

5-Limites des algorithmes génétiques :

Les algorithmes génétiques sont efficaces, mais en pratique ils ont certaines limites :

- Il n'est pas toujours facile de trouver une fonction fitness. Représenter un espace de problèmes dans les algorithmes génétiques est très complexe.
- Dans de nombreux cas, les algorithmes génétiques convergent prématurément vers une solution.
- Il est difficile de choisir les paramètres optimaux pour un algorithme génétique.
- Les algorithmes génétiques doivent être couplés à une technique de recherche locale pour un fonctionnement efficace.
- Les algorithmes génétiques nécessitent un grand nombre d'évaluations de fonctions de fitness.
- Il n'est pas facile de configurer un système basé sur un algorithme génétique. [6]

6- Différence entre algorithmes génétiques et méthodes conventionnelles :

Les algorithmes génétiques (AG) se distinguent des méthodes conventionnelles d'optimisation de plusieurs façons. Voici quelques-unes des principales différences :

- **Inspiration biologique :** Les AG sont inspirés par les principes de l'évolution biologique, de la génétique et de la sélection naturelle. Ils imitent le processus de reproduction, de sélection et de variation génétique observé dans la nature. En revanche, les méthodes conventionnelles d'optimisation ne sont pas basées sur ces principes biologiques.
- **Exploration de l'espace des solutions :** Les AG sont généralement plus efficaces pour explorer l'espace des solutions, en particulier lorsque celui-ci est complexe et vaste. Les AG utilisent des opérateurs de croisement et de mutation pour générer de nouvelles solutions potentielles, permettant ainsi d'explorer différentes régions de l'espace des solutions. Les méthodes conventionnelles, telles que les méthodes de descente de gradient, peuvent être plus limitées dans leur capacité à explorer efficacement l'espace des solutions.

- **Recherche multi-objectif :** Les AG sont bien adaptés à la résolution de problèmes multi-objectifs, où plusieurs critères doivent être optimisés simultanément. Les AG utilisent des techniques de sélection et de domination pour maintenir un ensemble de solutions diversifiées et non dominées, appelé "front de Pareto". Les méthodes conventionnelles sont généralement plus axées sur l'optimisation d'un seul objectif et peuvent avoir du mal à gérer efficacement les problèmes multi-objectif.
- **Adaptabilité et robustesse :** Les AG ont une certaine capacité d'adaptation et de robustesse intégrée. Grâce à la sélection et à la variation génétique, ils peuvent s'ajuster aux changements dans l'environnement du problème ou aux perturbations. Les méthodes conventionnelles sont souvent moins flexibles et peuvent nécessiter une adaptation manuelle en cas de changements.
- **Utilisation de la connaissance experte :** Les méthodes conventionnelles d'optimisation dépendent souvent de modèles mathématiques et d'informations sur le problème pour guider la recherche. Les AG, en revanche, peuvent fonctionner sans connaître explicitement le modèle du problème et peuvent être utilisés lorsque l'information experte est limitée.
- **Exploration globale vs locale :** Les AG sont généralement plus adaptés à l'exploration globale de l'espace des solutions, cherchant à trouver de bonnes solutions dans différentes régions. Les méthodes conventionnelles sont souvent plus efficaces pour l'optimisation locale, se concentrant sur la recherche d'une solution optimale dans une région spécifique. [6]

7- Algorithmes génétique et les systèmes de détection d'intrusion :

Cette section débute par une introduction au fonctionnement des algorithmes génétiques appliqués à la détection d'intrusions et un vue d'ensemble d'un algorithme de détection d'intrusion mis en œuvre à l'aide de la technique de l'algorithme génétique. Ensuite, le rôle joué par les algorithmes génétiques dans la détection d'intrusion est discuté. Au final, les avantages de la mise en place d'une détection d'intrusion des systèmes utilisant des algorithmes génétiques sont présentés.

Le fonctionnement d'un algorithme génétique lorsqu'il est appliqué à la détection d'intrusion peut être considéré comme une séquence d'étapes suivantes :

- Le module de capture de paquets ou renifleur présent dans le système de détection d'intrusion collecte les informations sur le trafic réseau ou les journaux.
- Le système de détection d'intrusion applique des algorithmes génétiques aux données capturées. L'algorithme génétique à ce stade a des règles de classification tirées des informations recueillies.
- Le système de détection d'intrusion applique alors l'ensemble de règles produit dans la phase précédente au trafic entrant.

L'application de règles aux données capturées entraîne l'initialisation de la population, qui à son tour entraîne la création d'une nouvelle population avec de bonnes qualités. Cette population est ensuite évaluée et une nouvelle génération avec de meilleures qualités est créée. Ensuite, les opérateurs génétiques sont appliqués à la génération nouvellement créée jusqu'à ce que l'individu le plus approprié soit trouvé. [6]

La figure, fournit un exemple d'implémentation d'algorithme génétique dans les systèmes de détection d'intrusion :

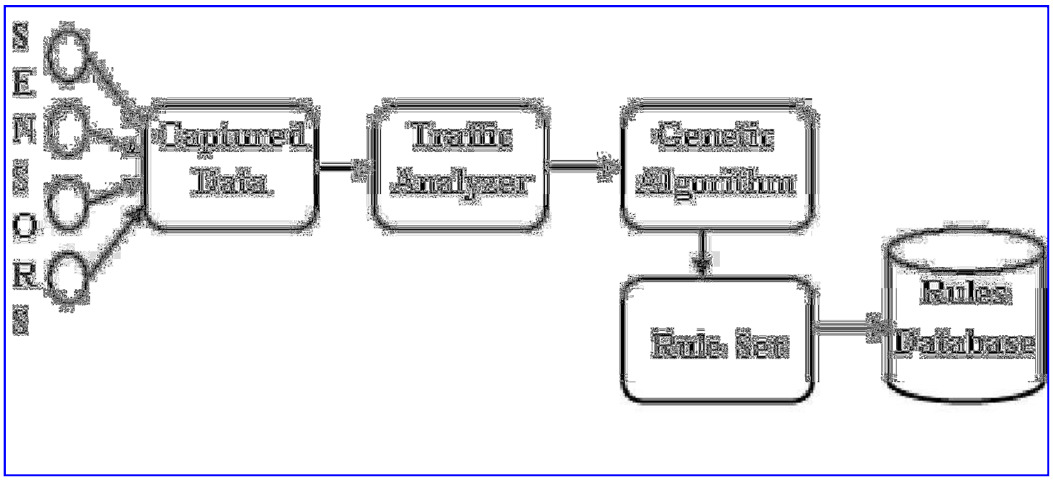

Figure 3 .1 : Implémentation d'algorithmes génétiques dans les systèmes de détection d'intrusion.

8- Les principes généraux des algorithmes génétiques (AG) :

Les principes généraux des algorithmes génétiques (AG) sont inspirés du processus de l'évolution biologique et sont utilisés pour résoudre des problèmes d'optimisation et de recherche. Voici les principes fondamentaux des algorithmes génétiques :

- **Codage des individus** : Les solutions potentielles à un problème sont représentées sous forme d'individus, souvent appelés "chromosomes". Ces chromosomes sont généralement encodés sous forme de chaînes de bits, mais ils

peuvent également être représentés sous d'autres formes, telles que des vecteurs ou des arbres.

- **Population initiale :** Une population initiale d'individus est générée de manière aléatoire. Chaque individu de la population représente une solution potentielle au problème.
- **Fonction d'évaluation :** Une fonction d'évaluation, également appelée fonction d'adaptation ou fonction de fitness, est définie pour mesurer la qualité de chaque individu dans la population. Cette fonction attribue une valeur numérique à chaque individu en fonction de son aptitude à résoudre le problème.
- **Sélection :** Les individus les mieux adaptés, c'est-à-dire ceux ayant les valeurs de fitness les plus élevées, sont sélectionnés pour la reproduction. Les individus moins adaptés ont une probabilité plus faible d'être sélectionnés, mais ils ne sont pas complètement exclus.
- **Croisement (cross over) :** Les individus sélectionnés sont combinés par des opérations de croisement afin de créer de nouvelles solutions potentielles, appelées descendants. Le croisement implique l'échange de parties des chromosomes parentaux pour créer des chromosomes descendants.
- **Mutation :** Les descendants nouvellement créés subissent des mutations aléatoires avec une faible probabilité. La mutation introduit une petite modification aléatoire dans le chromosome d'un individu, ce qui permet d'explorer de nouvelles régions de l'espace de recherche.
- **Remplacement :** Les individus de la population parentale sont remplacés par les descendants créés par croisement et mutation. Cette étape garantit que la population évolue vers de meilleures solutions au fil du temps.
- **Critère d'arrêt :** L'algorithme génétique s'exécute pendant un certain nombre d'itérations ou jusqu'à ce qu'un critère d'arrêt prédéfini soit atteint. Le critère d'arrêt peut être basé sur le nombre d'itérations, la convergence des solutions ou d'autres conditions spécifiques au problème. [5]

9- Les Avantages des algorithmes génétiques :

Les algorithmes génétiques sont des méthodes d'optimisation inspirées par le processus de l'évolution biologique. Ils sont utilisés pour résoudre des problèmes d'optimisation combinatoire et offrent plusieurs avantages :

- **Exploration de l'espace de recherche étendu :** Les algorithmes génétiques sont efficaces pour explorer un grand espace de recherche en générant et en évaluant une population de solutions potentielles. Ils peuvent rechercher des solutions dans des domaines complexes et multidimensionnels.
- **Recherche globale :** Les algorithmes génétiques cherchent généralement des solutions globales plutôt que de se contenter d'une solution locale optimale. En utilisant des techniques telles que la sélection, le croisement et la mutation, ils peuvent parcourir différentes régions de l'espace de recherche pour trouver des solutions potentiellement meilleures.
- **Parallélisme et évolutivité :** Les algorithmes génétiques peuvent être parallélisés, ce qui signifie qu'ils peuvent être exécutés sur plusieurs processeurs ou machines en même temps. Cela permet d'accélérer le processus de recherche et de traiter des problèmes de grande taille.
- **Adaptabilité :** Les algorithmes génétiques peuvent s'adapter à différents types de problèmes d'optimisation combinatoire. Ils peuvent être personnalisés en fonction des caractéristiques spécifiques du problème, tels que la représentation des solutions, les opérateurs de croisement et de mutation, les critères de sélection, etc.
- **Robustesse aux optima locaux :** Les algorithmes génétiques ont la capacité d'éviter de rester bloqués dans des optima locaux en introduisant des mécanismes d'exploration de l'espace de recherche. Les opérations de croisement et de mutation permettent d'introduire de nouvelles solutions potentielles et de les évaluer.
- **Adaptation à des contraintes multiples :** Les algorithmes génétiques peuvent être étendus pour traiter des problèmes avec des contraintes multiples. En utilisant des techniques telles que les fonctions de pénalité, les méthodes de domination ou les méthodes basées sur le classement, ils peuvent prendre en compte plusieurs objectifs et trouver des solutions satisfaisantes.

10- Les inconvénients des algorithmes génétiques :

Bien que les algorithmes génétiques présentent de nombreux avantages, ils ont également quelques inconvénients à prendre en compte :

- **Temps de calcul élevé :** Les algorithmes génétiques peuvent nécessiter un temps de calcul important, en particulier pour les problèmes de grande taille. La génération et l'évaluation de nombreuses solutions potentielles, ainsi que les opérations de croisement et de mutation, peuvent être coûteuses en termes de temps de calcul.
- **Dépendance aux paramètres :** Les performances des algorithmes génétiques dépendent de l'ajustement adéquat des paramètres tels que la taille de la population, le taux de croisement, le taux de mutation, etc. La sélection de paramètres inappropriés peut conduire à des performances médiocres ou à une convergence prématurée.
- **Difficulté de représentation des solutions :** La représentation des solutions dans un algorithme génétique peut être un défi, en particulier pour les problèmes complexes. Le choix d'une représentation adéquate peut avoir un impact significatif sur l'efficacité de l'algorithme.
- **Risque de convergence prématurée :** Il existe un risque que l'algorithme génétique converge vers une solution suboptimale sans explorer suffisamment l'espace de recherche. Cela peut se produire si les opérations de sélection, de croisement et de mutation ne sont pas bien équilibrées, ou si la diversité de la population diminue trop rapidement.
- **Sensibilité aux conditions initiales :** Les performances des algorithmes génétiques peuvent varier en fonction des conditions initiales, notamment de la population initiale. Des populations initiales mal choisies peuvent entraîner une convergence rapide vers une région sous-optimale de l'espace de recherche.
- **Représentation binaire limitée :** Les algorithmes génétiques traditionnels utilisent souvent une représentation binaire pour les solutions, ce qui peut être limitant pour certains problèmes où les variables sont continues ou discrètes. Des techniques supplémentaires, telles que le codage réel ou les algorithmes génétiques basés sur des arbres, peuvent être nécessaires pour résoudre ces types de problèmes.
-

11. Rôle des algorithmes génétiques dans la détection d'Intrusion :

Les algorithmes génétiques (AG) jouent un rôle important dans la détection d'intrusion en renforçant la capacité des systèmes de détection à identifier et à réagir aux activités malveillantes. Voici quelques rôles clés que les AG peuvent jouer dans la détection d'intrusion :

- **Sélection de caractéristiques :** Les AG peuvent être utilisés pour sélectionner les caractéristiques les plus pertinentes à partir des données de trafic ou des journaux système. En utilisant des techniques d'évaluation de fitness appropriées, les AG peuvent identifier les caractéristiques les plus discriminantes qui contribuent à la détection des intrusions. Cela permet de réduire la dimensionnalité des données et d'améliorer l'efficacité de la détection.
- **Optimisation des paramètres :** Les AG peuvent être utilisés pour optimiser les paramètres des systèmes de détection d'intrusion. Cela inclut les paramètres des modèles d'apprentissage automatique utilisés par les IDS, tels que les réseaux de neurones, les arbres de décision, etc. Les AG peuvent rechercher les combinaisons optimales de paramètres qui maximisent la sensibilité et la spécificité de la détection, améliorant ainsi les performances globales du système.
- **Génération de règles d'analyse** : Les AG peuvent être utilisés pour générer automatiquement des règles d'analyse qui permettent de détecter des comportements suspects ou des schémas d'attaques connus. En utilisant des mécanismes de sélection et de recombinaison génétiques, les AG peuvent évoluer et affiner les règles d'analyse au fil du temps pour s'adapter à de nouvelles formes d'attaques ou à des changements dans les modèles de trafic.
- **Détection d'anomalies :** Les AG peuvent être utilisés pour construire des modèles de comportement normal à partir de données historiques. Ces modèles peuvent être utilisés pour détecter des comportements anormaux qui pourraient indiquer des intrusions. Les AG peuvent optimiser les paramètres de ces modèles et identifier les schémas d'anomalies complexes qui peuvent échapper à des méthodes de détection conventionnelles.
- **Adaptation et évolutivité :** Les AG peuvent permettre aux systèmes de détection d'intrusion de s'adapter et de s'améliorer en fonction des changements dans l'environnement ou des nouvelles menaces. Les AG peuvent être utilisés pour ajuster automatiquement les paramètres, les règles d'analyse et les modèles en

réponse à l'évolution des attaques ou des modèles de trafic. Cela permet de maintenir la performance et l'efficacité de la détection au fil du temps.

12- Avantages des algorithmes génétique pour les systèmes de détection d'intrusion :

La mise en œuvre d'algorithmes génétiques offre de nombreux avantages aux systèmes de détection d'intrusion. Les avantages d'utiliser algorithmes génétiques pour la détection d'intrusion peuvent être résumés comme suit :

- Les algorithmes génétiques offrent aux systèmes de détection d'intrusion un parallélisme intrinsèque.
- Les algorithmes génétiques sont capables de travailler dans plusieurs directions simultanément. Cela les rend bénéfiques pour analyser les énormes volumes de données multidimensionnelles à traiter par un système de détection d'intrusion.
- Les algorithmes génétiques fonctionnent avec des populations de solutions plutôt qu'avec une solution unique. Cela les rend aptes à détection d'intrusion basée sur le comportement, où les attributs de comportement peuvent présenter des valeurs variables.
- Les algorithmes génétiques sont hautement ré-entraînables. Par conséquent, l'utilisation d'algorithmes génétiques pour la détection d'intrusion ajoutera à l'adaptabilité du système.
- Les algorithmes génétiques évoluent avec le temps en utilisant le croisement et la mutation. La propriété d'évoluer dans le temps fait un bon choix pour la génération dynamique de règles. [6]

Conclusion :

En conclusion, le chapitre 3 a fourni une vue d'ensemble approfondie des systèmes de détection d'intrusion basés sur les algorithmes génétiques (AG). Ce dernier offre une approche prometteuse pour la détection d'intrusion, grâce à leur capacité à évoluer et à s'adapter aux nouvelles menaces. Cependant, il est important de prendre en compte les limitations et les défis liés à leur mise en œuvre.

Chapitre 4

Implémentation et Tests

1- Introduction :

Le présent système de détection d'intrusion basé sur l'hôte est développé à l'aide du langage de programmation Java pour aider à détecter les paquets ayant des adresses IP usurpées. Il sniffe en premier lieu les paquets entrants sur le système hôte et ensuite les analyses afin de détecter une intrusion. Compte tenu du fait que ce processus de détection est une opération de bas niveau, l'application Java utilise la bibliothèque de capture de paquets Java (JpCap) qui fonctionne en conjonction avec la bibliothèque de capture de paquets Windows (WinpCap).

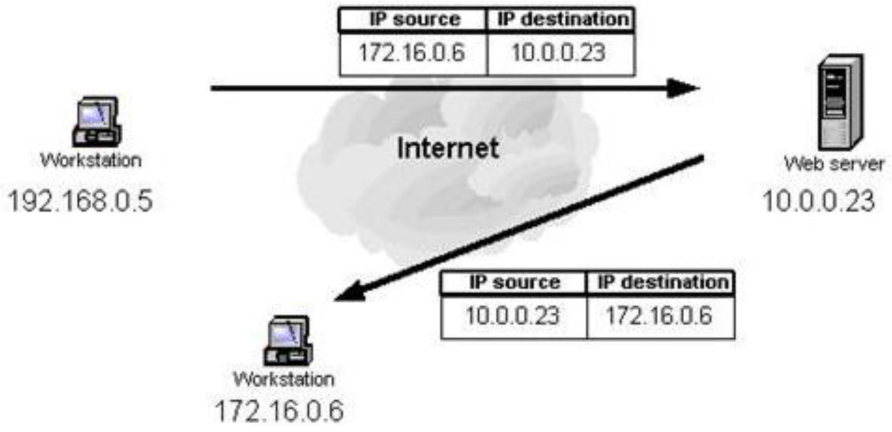

Figure 4.1 : Diagramme montrant l'usurpation d'adresse IP.

2- À propos de jpCap:

JpCap est une bibliothèque open source de capture de paquets réseau basée sur les bibliothèques LibpCap et WinpCap. Il est utilisable avec Java pour capturer et afficher le trafic réseau sur les ordinateurs LINUX, Windows et Macintosh. JpCap capture les types de paquets suivants et peut même analyser chaque en-tête de paquet et la charge utile des données.

- Ethernet
- TCP
- UDP
- IPv4
- IPv6
- ARP/RARP

- Paquets ICMPv4

JpCap capture les paquets bruts en direct du câble, identifie automatiquement ses types de paquets et génère les objets Java correspondants. Il peut également filtrer les paquets en fonction des règles spécifiées par l'utilisateur avant de les envoyer à l'application. JpCap peut également envoyer des paquets bruts au réseau, enregistrer et lire des paquets capturés vers et depuis un fichier hors ligne.

3- À propos de WinPcap :

WinPcap est une bibliothèque logicielle open-source spécifiquement conçue pour la capture et l'analyse de paquets réseau sous le système d'exploitation Windows. Elle permet aux développeurs d'interagir avec les interfaces réseau de Windows pour capturer et analyser les paquets qui transitent sur le réseau.

4- Fonctionnement de l'application:

4.1-Charger et sélectionner les interfaces réseau disponibles sur l'ordinateur :

Comme nous le savons, pour capturer des paquets à partir d'un réseau, la première chose à faire est d'obtenir la liste des interfaces réseau fonctionnelles sur l'ordinateur. Pour ce faire, JpCap fournit la méthode JpcapCaptor.getDeviceList(). Il renvoie un tableau d'objets d'interfaces réseau. Par conséquent, la première opération importante que le système effectue est de permettre à l'utilisateur de charger les interfaces réseau disponibles sur l'ordinateur afin qu'il puisse choisir l'interface souhaitée dont les paquets doivent être reniflés et analysés. La méthode de classe Java écrite dans la base de code 1 ci-dessous permet de réaliser cette opération

Code base 1 : Méthode Java pour ouvrir les interfaces disponibles

```
public synchronizedvoidGetAvailableInterfaces() {
IDS.Interfaces= JpcapCaptor.getDeviceList();
IDS.TotalNumberOfInterfaces= IDS.Interfaces.length;
IDS.MyMacAddresses= new byte[IDS.TotalNumberOfInterfaces][];
Vector list = new Vector();
list.clear();
MainFrame.TextArea.setText("");
```

```
intcounter = 0;

for(inti = 0; i <IDS.Interfaces.length; i++) {

counter = counter + 1;

print.TextAreaAppend("-----------------------INFORMATION ON NETWORK
INTERFACE " + counter + "-----------------------");

print.TextAreaAppend("\nName: " + IDS.Interfaces[i].name);

print.TextAreaAppend("\nDataLink Name: " + IDS.Interfaces[i].datalink_name);

print.TextAreaAppend("\nDataLink Description: " +
IDS.Interfaces[i].datalink_description);

print.TextAreaAppend("\nGeneral Description: " + IDS.Interfaces[i].description);

list.add("INTERFACE " + counter + ": " + IDS.Interfaces[i].description);

print.TextAreaAppend("\nLoop Back: " + IDS.Interfaces[i].loopback);

print.TextAreaAppend("\nIP Address: ");

for(NetworkInterfaceAddress c : IDS.Interfaces[i].addresses) {

print.TextAreaAppend(c.address.toString());

}

print.TextAreaAppend("\nMAC Address: ");

for(byte c : IDS.Interfaces[i].mac_address) {

print.TextAreaAppend(Integer.toHexString(c & 0xff) + ":");

}

print.TextAreaAppend("\n");

print.TextAreaAppend("\n");

}

MainFrame.InterfacesList.setListData(list);

    }
    Fin de code
```

Après le processus d'ouverture d'interface, l'utilisateur est alors autorisé à sélectionner l'interface souhaitée à renifler ou la combinaison d'interfaces à renifler. Les processus de sélection simples permettent au système d'obtenir une instance de JpcapCaptor, comme on peut le voir à la ligne 1 de la base de code 1.

4.2-Commencer l'opération de reniflement :

Lors de l'activation, c'est-à-dire une fois que vous avez obtenu une instance de JpcapCaptor, vous pouvez capturer des paquets à partir de l'interface. Il existe deux approches principales pour capturer des paquets lors de l'utilisation d'une instance JpcapCaptor et elles sont :

a. Utiliser une méthode de rappel

b. Capturer les paquets un par un

4.2.1-Utilisation d'une méthode de rappel :

Dans cette approche, vous implémentez une méthode de rappel pour traiter les paquets capturés, puis transmettez la méthode de rappel à JpCap afin que JpCap la rappelle chaque fois qu'il capture un paquet. Voyons comment vous pouvez adopter cette approche en détail.

Tout d'abord, vous implémentez une méthode de rappel en définissant une nouvelle classe qui implémente l'interface PacketReceiver. L'interface PacketReceiver définit une méthode receivePacket(), vous devez donc implémenter une méthode receivePacket() dans votre classe. La classe suivante implémente une méthode receivePacket() qui imprime simplement un paquet capturé.

Code base 2 : classe java qui implémente l'interface PacketReceiver

```
class Packet Printer implements Packet Receiver {
public void receive Packet(Packet packet) {
System.out.println(packet);
}
}
```

Fin de code

Une fois que la classe dans la base de code 2 ci-dessus a été configurée, vous pouvez appeler les méthodes JpcapCaptor.processPacket() ou JpcapCaptor.loopPacket() pour commencer la capture à l'aide de la méthode de rappel. Lors de l'appel de la méthode processPacket() ou loopPacket(), vous pouvez également spécifier le nombre de paquets à capturer avant le retour de la méthode. Vous pouvez spécifier -1 pour continuer à capturer les paquets à l'infini.

Code base 3 : code pour capturer/renifler le trafic

```
public void Capture(){
JpcapCaptor captor=JpcapCaptor.openDevice(device[index], 65535, true, 5000);
while (true){
captor.processPacket(10,new PacketPrinter());
captor.close();
}
}
```

Fin de code

Les deux méthodes de rappel, processPacket() et loopPacket(), sont très similaires. Habituellement, vous voudrez peut-être utiliser processPacket() car il prend en charge le délai d'attente et le mode non bloquant, contrairement à loopPacket().

4.2.2-Capturer les paquets un par un :

Capturer les paquets un par un L'utilisation d'une méthode de rappel est un peu délicate car vous ne savez pas quand la méthode de rappel est appelée par Jpcap. Si vous ne souhaitez pas utiliser de méthode de rappel, vous pouvez également capturer des paquets à l'aide de la méthode JpcapCaptor.getPacket(). La méthode getPacket() renvoie simplement un paquet capturé. Vous pouvez (ou devez) appeler la méthode getPacket() plusieurs fois pour capturer des paquets consécutifs

Codebase 4 : méthode Java montrant comment capturer le trafic un par un

```
public void CaptureOneByOne{
JpcapCaptor captor=JpcapCaptor.openDevice(device[index], 65535, true, 5000);
for(int i=0;i<10;i++){
```

```
//capture a single packet and print it out
System.out.println(captor.getPacket());
}
captor.close();
}
```

Fin de code

4.3-Initialiser la détection d'usurpation d'adresse IP :

L'application effectue la détection d'usurpation à l'aide de l'algorithme génétique. L'algorithme génétique utilise simplement l'examen des chromosomes pour détecter les changements de phénotype et les mutations. On extrait des gènes sélectionnés des paquets IP, puis on combine ces gènes dans un chromosome de vérification appelé le chromosome des paquets nécessaires. Tant que le mécanisme de détection d'usurpation d'adresse IP est activé, l'application examine l'en-tête de chaque paquet et extrait les attributs/gènes suivants :

- IP source
- Adresse Mac source
- Durée initiale de vie
- Type de protocole de comptage de sauts
- ID de paquet

Tous ces gènes représentent des variables/gènes qui peuvent être falsifiés si un intrus essaie d'usurper. Par conséquent, ce que fait l'IDS est de convertir tous ces gènes en leur équivalent binaire et de les concaténer pour former le chromosome. Voir la structure chromosomique ci-dessous.

4.4-Structure chromosomique :

Adresse MAC source en binaire +
Adresse MAC de destination en binaire +
TTL en binaire +
Nombre de sauts en binaire +
Validité de l'ID de paquet ;

4.4.1-Exemple de chromosome :

Prenons par exemple un paquet avec les détails suivants :

MAC source : 02:00:4c:4f:4f:50

MAC de destination : 01:00:5e:7f:ff:fa

TTL: 32

Nombre de sauts: 1

Identifiant du paquet: 288

Ce paquet aura le chromosome suivant :

1001001100100111110011111010000101011110111111111111111111110101111111

Notez que le dernier bit du chromosome représente la validité de l'ID de paquet. Si l'ID du paquet est supérieur à celui envoyé précédemment, la validité du paquet est mise à 1 sinon, c'est a. L'ID de paquet est une valeur croissante, en tant que telle, la seule raison pour laquelle un paquet suivant peut avoir un ID de paquet inférieur est principalement parce qu'un usurpateur falsifie des éléments de l'autre côté.

Si l'application reçoit un paquet pour la première fois d'une adresse IP particulière, elle enregistre son chromosome avec 100 % de fitness. Cependant, le chromosome des paquets suivants reçus de cette adresse IP source est comparé au chromosome du paquet le plus adapté dans la base de données. Si le niveau de fitness du nouveau paquet reçu est inférieur au paquet précédemment reçu, cela indique une falsification. A ce stade, l'application alerte l'administrateur réseau d'une intrusion.

Le niveau de fitness minimum autorisé par défaut défini par notre application est de 65 % de fitness chromosomique. Cependant, l'utilisateur peut l'ajuster en fonction de l'environnement réseau.

4.5-Comment les niveaux de fitness des chromosomes sont générés :

Par exemple, si mon système reçoit pour la première fois un paquet d'une adresse IP source, l'application extrait les gènes comme indiqué ci-dessus et génère le chromosome. Prenons par exemple que le chromosome d'un paquet est : 1010101010. De plus, si le système reçoit un autre paquet de cette même adresse IP source, on s'attend à ce que le nouveau chromosome qui sera généré soit identique ou presque identique au précédent car l'adresse MAC source l'adresse, l'adresse MAC de destination, la durée de vie initiale et le nombre de sauts doivent avoir à peu près la même valeur. De plus, l'ID de paquet devrait être supérieur à celui du paquet précédemment reçu. Si tout cela est vrai, alors le deuxième paquet reçu devrait avoir à peu près le même chromosome. Cependant, dans le cas où le chromosome du second paquet reçu est 1100101001, on voit par comparaison que

	1	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**
	1	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**1**
Le résultat	**M**	**N**	**N**	**M**	**M**	**M**	**M**	**M**	**N**	**N**

Où :

- **M = Correspondance**
- **N = Aucune correspondance**

D'après ce qui précède, nous pouvons voir qu'il existe 4 différences dans le chromosome, ce qui signifie que le paquet B a été grossièrement altéré, n'ayant que 60% de fitness. À ce stade, le système de messagerie de l'application enverra à l'administrateur un message d'alerte.

4.5.1- digramme de classe pour cette phase :

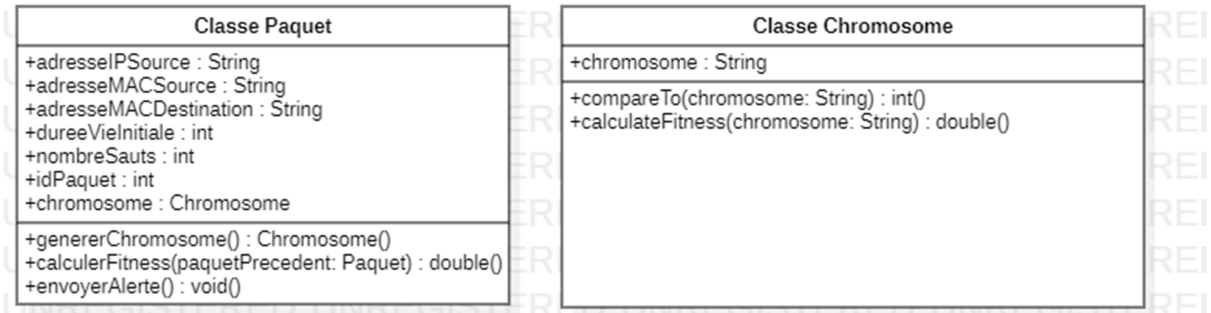

Figure 4.2 : digramme de classe.

Dans ce diagramme de classes simplifié, nous avons deux classes principales : "Chromosome" et "Paquet". La classe "Chromosome" représente un chromosome et possède une variable d'instance "chromosome" pour stocker la séquence binaire. Elle a deux méthodes publiques : "compareTo" pour comparer deux chromosomes et "calculateFitness" pour calculer le niveau de fitness.

La classe "Paquet" représente un paquet de données et possède plusieurs variables d'instance pour stocker les différentes informations comme l'adresse IP source, les adresses MAC source et destination, la durée de vie initiale, le nombre de sauts, l'ID du paquet et le chromosome correspondant. Elle a des méthodes pour générer un chromosome, calculer le niveau de fitness en comparant avec le paquet précédent, et envoyer une alerte si le niveau de fitness est inférieur à un seuil.

4.6- Enregistrer dans la base de données :

À ce stade, l'application enregistre les détails du paquet dans la base de données afin que les futurs paquets entrants puissent être comparés aux paquets déjà enregistrés.

4.7- Enregistrer dans un fichier :

Le système enregistre les paquets capturés dans un fichier binaire afin que vous puissiez les récupérer ultérieurement Pour enregistrer les paquets capturés, le système ouvre d'abord un fichier en appelant la méthode JpcapWriter.openDumpFile() avec une instance de JpcapCaptor qui a été utilisée pour capturer les paquets et un nom de fichier String . Le code ci-dessous explique explicitement ce processus.

Codebase 5 : code montrant comment JpCap écrit dans un fichier

```
JpcapCaptorcaptor = JpcapCaptor.openDevice(IDS.Interfaces[InterfaceToPrint], 65535, true, 5000);

JpcapWriter writer=JpcapWriter.openDumpFile(captor,"DumpFile"+InterfaceToPrint+".txt");

writer.writePacket(Pack);

    writer.close();
```

Fin du code

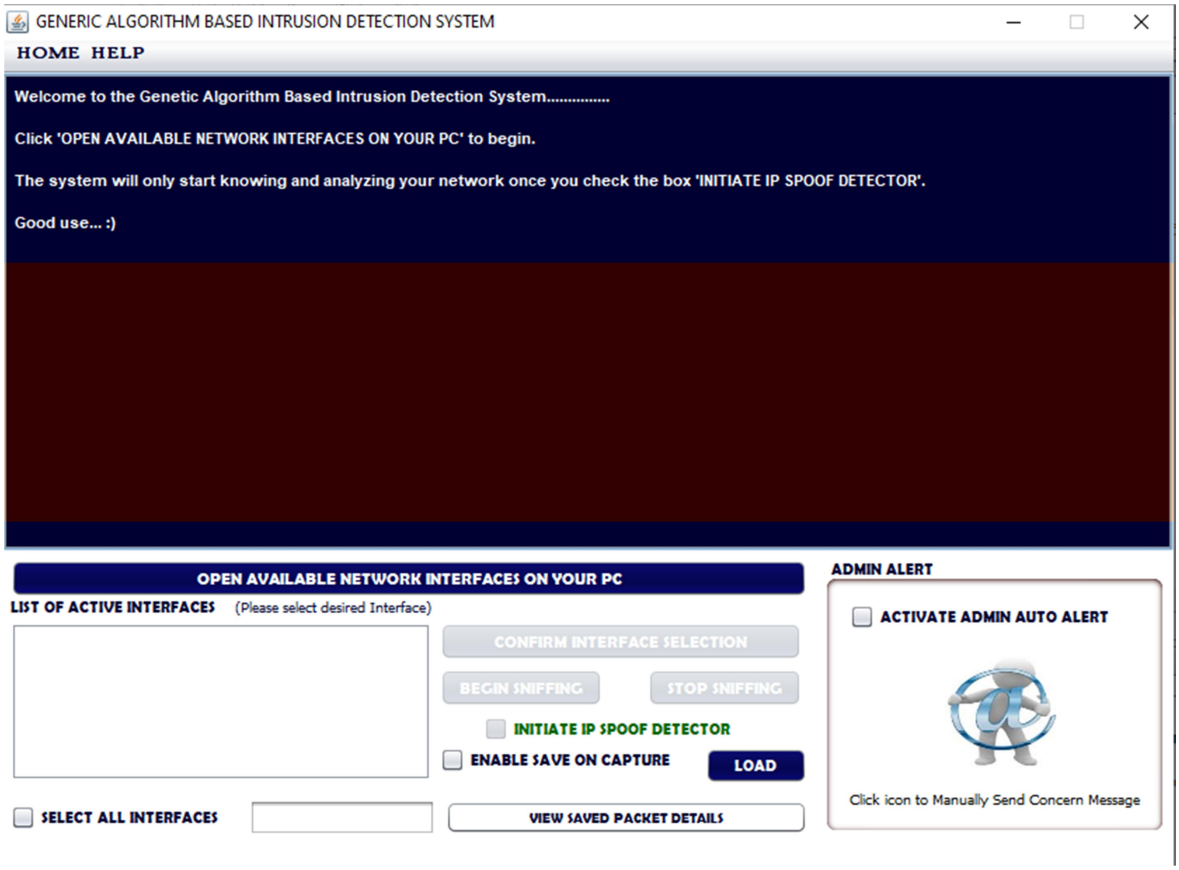

Figure 4.3 : l'interface 1.

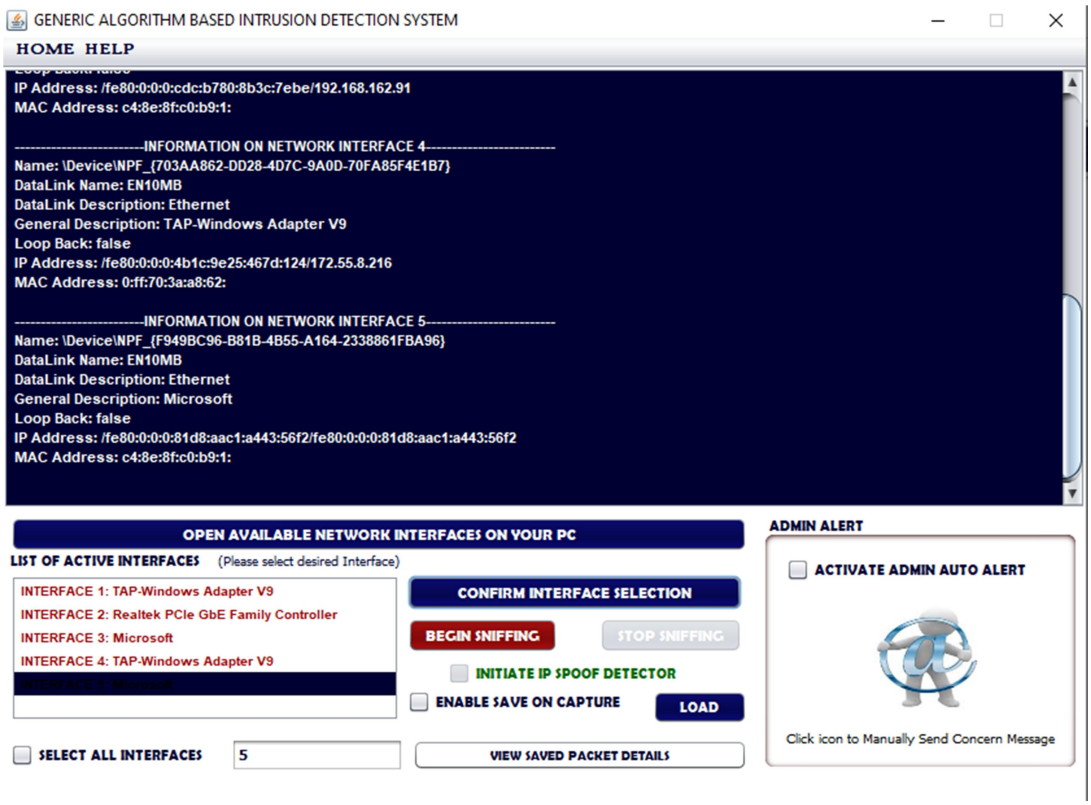

Figure 4.3 : l'interface 2**.**

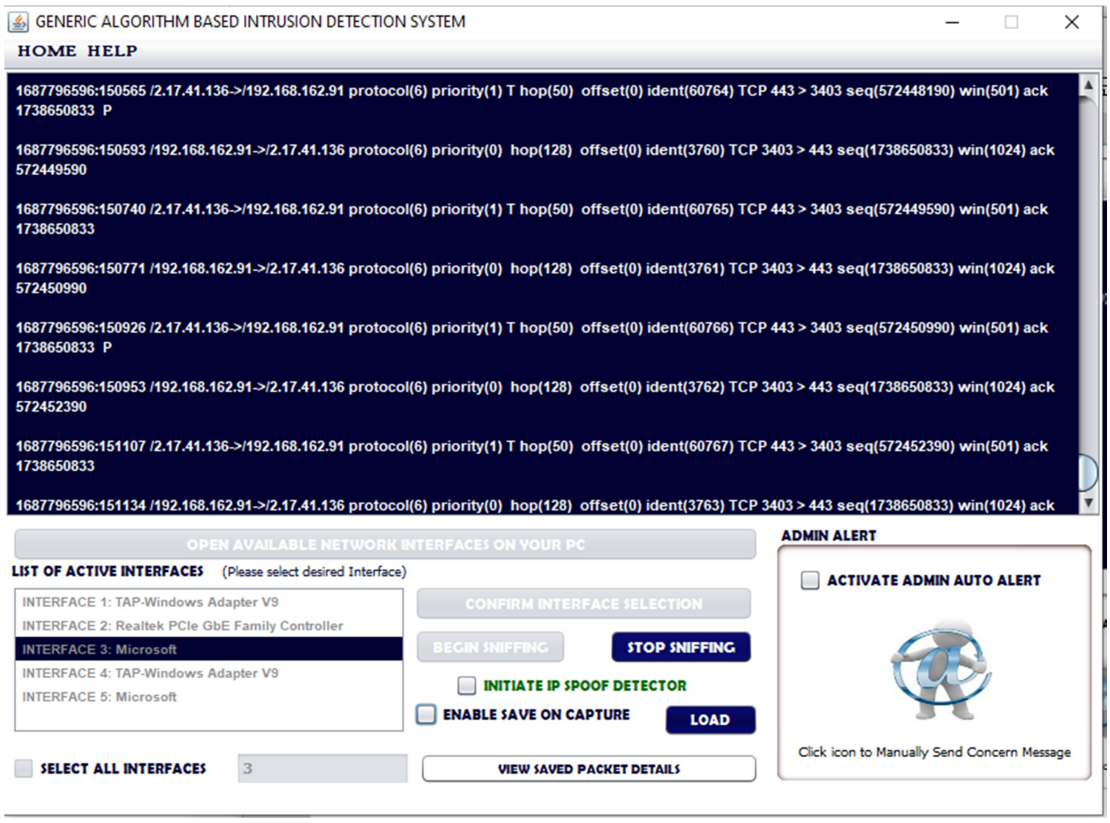

Figure 4.4 : l'interface 3**.**

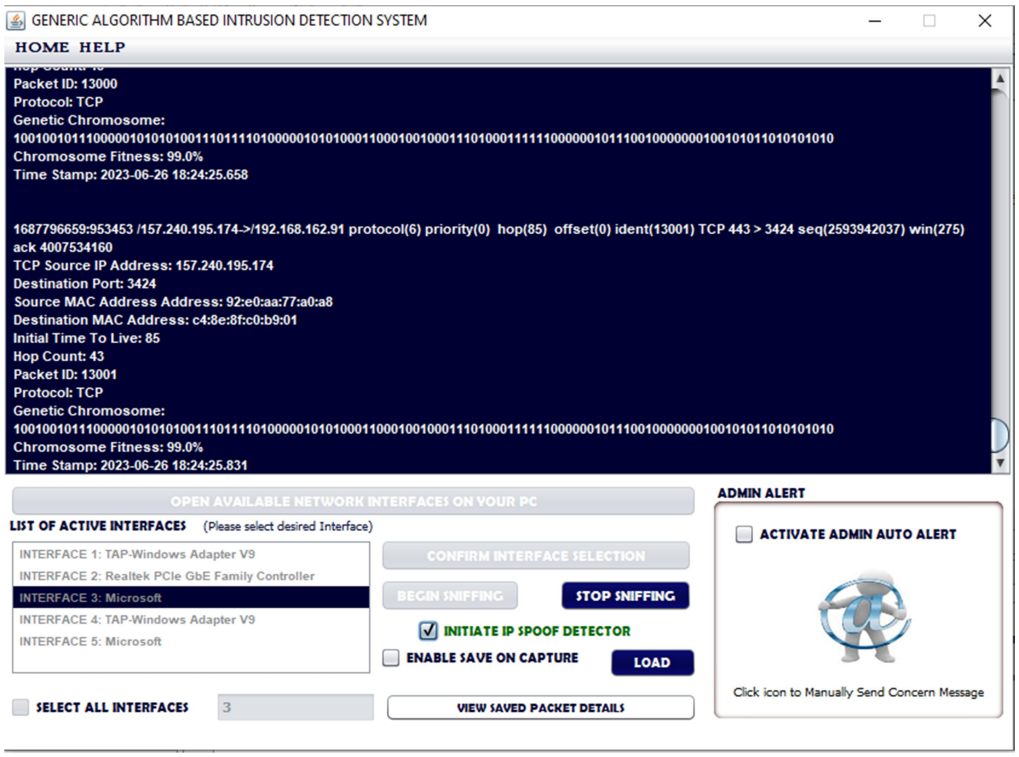

Figure 4.5 : l'interface 4**.**

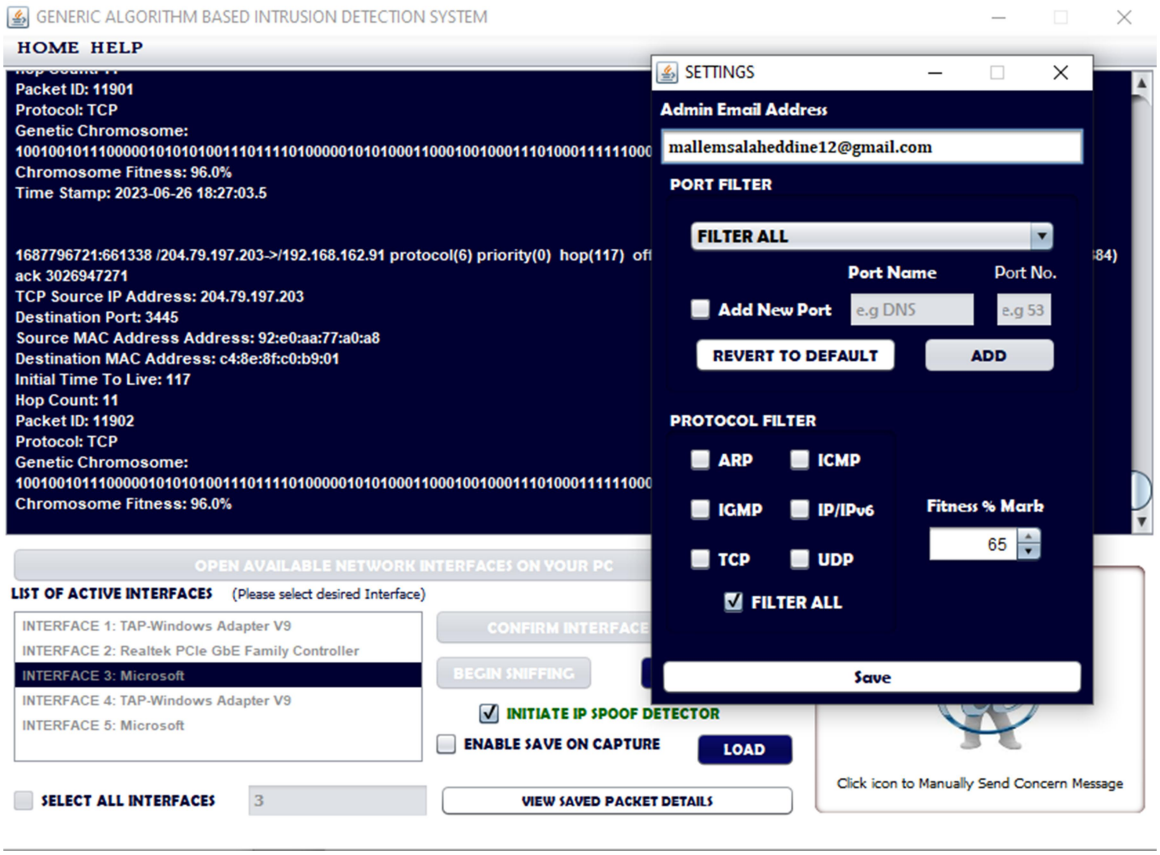

Figure 4.6: l'interface 5**.**

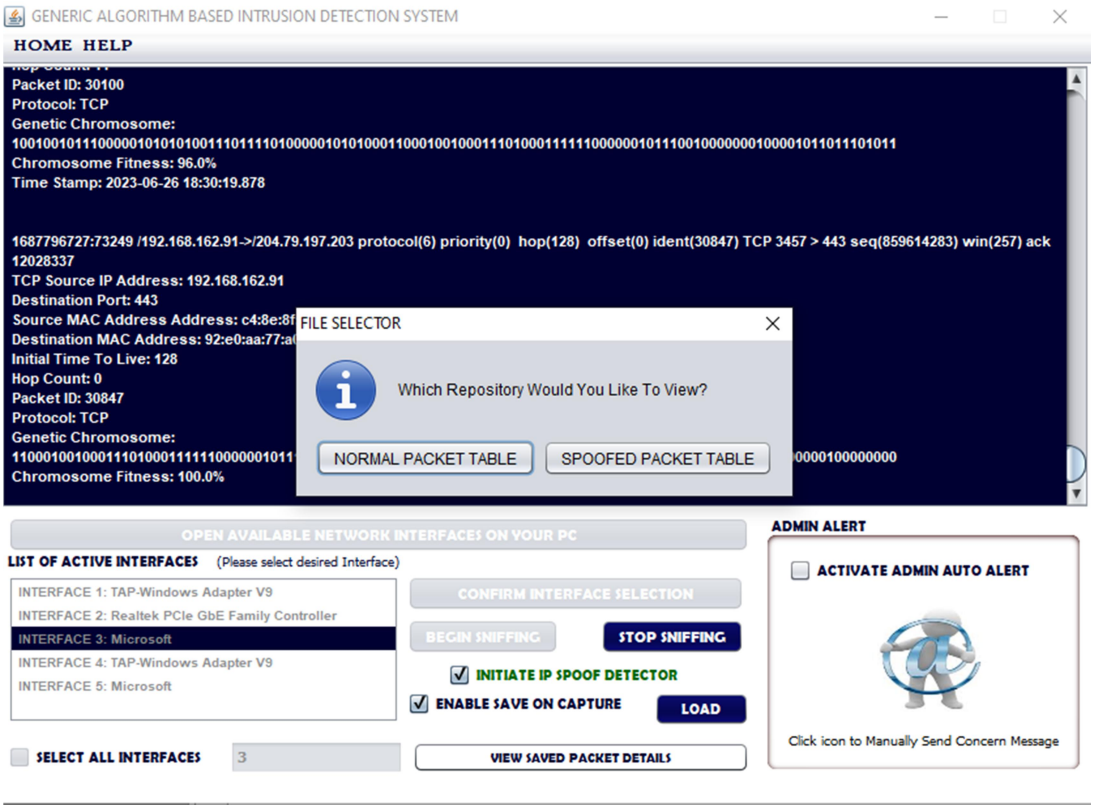

Figure 4.7: l'interface 6.

RECEIVED PACKETS DETAILS

S/N	SOURCE IP	SOURCE MAC	DESTINATION M...	INITIAL TTL	HOP COUNT	PACKET ID	PROTOCOL	CHROMOSOME	FITNESS	TIME RECEIVED
1	62.252.168.153	c0:05:c2:3d:0c:88	192.168.0.42	1c:39:47:d4:c2:e6	59	5	33402	TCP	1100000000000...	100.0
2	62.252.168.153	c0:05:c2:3d:0c:88	192.168.0.42	1c:39:47:d4:c2:e6	59	5	8421	TCP	1100000000000...	99.0
3	192.168.0.42	1c:39:47:d4:c2:e6	62.252.168.153	c0:05:c2:3d:0c:88	128	0	26134	TCP	0001110000111...	100.0
4	192.168.0.42	1c:39:47:d4:c2:e6	62.252.168.153	c0:05:c2:3d:0c:88	128	0	26135	TCP	0001110000111...	100.0
5	192.168.0.42	1c:39:47:d4:c2:e6	62.252.168.153	c0:05:c2:3d:0c:88	128	0	26136	TCP	0001110000111...	100.0
6	192.168.0.42	1c:39:47:d4:c2:e6	62.252.168.153	c0:05:c2:3d:0c:88	128	0	26137	TCP	0001110000111...	100.0
7	192.168.0.24	b2:b8:88:89:21:8b	239.255.255.250	01:00:5e:7f:ff:fa	1	31	28099	UDP	1011001010111...	100.0
8	192.168.0.34	a0:2b:b8:4b:bb:99	224.0.0.251	01:00:5e:00:00:fb	255	0	23151	UDP	1010000000101...	100.0
9	192.168.0.24	b2:b8:88:89:21:8b	239.255.255.250	01:00:5e:7f:ff:fa	1	31	28100	UDP	1011001010111...	100.0
10	192.168.0.10	38:63:bb:b9:c3:42	230.82.31.145	01:00:5e:52:1f:91	1	31	25669	UDP	0011100001100...	100.0
11	192.168.0.42	1c:39:47:d4:c2:e6	216.58.212.101	c0:05:c2:3d:0c:88	128	0	23137	TCP	0001110000111...	99.0
12	216.58.212.101	c0:05:c2:3d:0c:88	192.168.0.42	1c:39:47:d4:c2:e6	57	7	12108	TCP	1100000000000...	100.0
13	192.168.0.42	1c:39:47:d4:c2:e6	157.55.235.158	c0:05:c2:3d:0c:88	128	0	27625	UDP	0001110000111...	100.0
14	157.55.235.158	c0:05:c2:3d:0c:88	192.168.0.42	1c:39:47:d4:c2:e6	53	11	6393	UDP	1100000000000...	100.0
15	192.168.0.42	1c:39:47:d4:c2:e6	216.58.204.14	c0:05:c2:3d:0c:88	128	0	30826	UDP	0001110000111...	100.0
16	192.168.0.42	1c:39:47:d4:c2:e6	216.58.204.14	c0:05:c2:3d:0c:88	128	0	30827	UDP	0001110000111...	100.0
17	216.58.204.14	c0:05:c2:3d:0c:88	192.168.0.42	1c:39:47:d4:c2:e6	57	7	0	UDP	1100000000000...	100.0
18	216.58.204.14	c0:05:c2:3d:0c:88	192.168.0.42	1c:39:47:d4:c2:e6	57	7	0	UDP	1100000000000...	100.0
19	216.58.204.14	c0:05:c2:3d:0c:88	192.168.0.42	1c:39:47:d4:c2:e6	57	7	0	UDP	1100000000000...	100.0
20	192.168.0.42	1c:39:47:d4:c2:e6	216.58.204.14	c0:05:c2:3d:0c:88	128	0	30828	UDP	0001110000111...	100.0
21	192.168.0.42	1c:39:47:d4:c2:e6	216.58.204.14	c0:05:c2:3d:0c:88	128	0	30829	UDP	0001110000111...	100.0
22	192.168.0.42	1c:39:47:d4:c2:e6	216.58.204.14	c0:05:c2:3d:0c:88	128	0	30830	UDP	0001110000111...	100.0
23	216.58.204.14	c0:05:c2:3d:0c:88	192.168.0.42	1c:39:47:d4:c2:e6	57	7	0	UDP	1100000000000...	100.0
24	216.58.204.14	c0:05:c2:3d:0c:88	192.168.0.42	1c:39:47:d4:c2:e6	57	7	0	UDP	1100000000000...	100.0
25	216.58.204.14	c0:05:c2:3d:0c:88	192.168.0.42	1c:39:47:d4:c2:e6	57	7	0	UDP	1100000000000...	100.0
26	192.168.0.42	1c:39:47:d4:c2:e6	216.58.204.14	c0:05:c2:3d:0c:88	128	0	30831	UDP	0001110000111...	100.0
27	192.168.0.10	38:63:bb:b9:c3:42	239.192.152.143	01:00:5e:40:98:8f	255	0	10777	UDP	0011100001100...	100.0
28	192.168.0.42	1c:39:47:d4:c2:e6	194.168.4.100	c0:05:c2:3d:0c:88	128	0	9343	UDP	0001110000111...	99.0
29	194.168.4.100	c0:05:c2:3d:0c:88	192.168.0.42	1c:39:47:d4:c2:e6	252	3	40895	UDP	1100000000000...	100.0

PRINT

Figure 4.8 : l'interface 7.

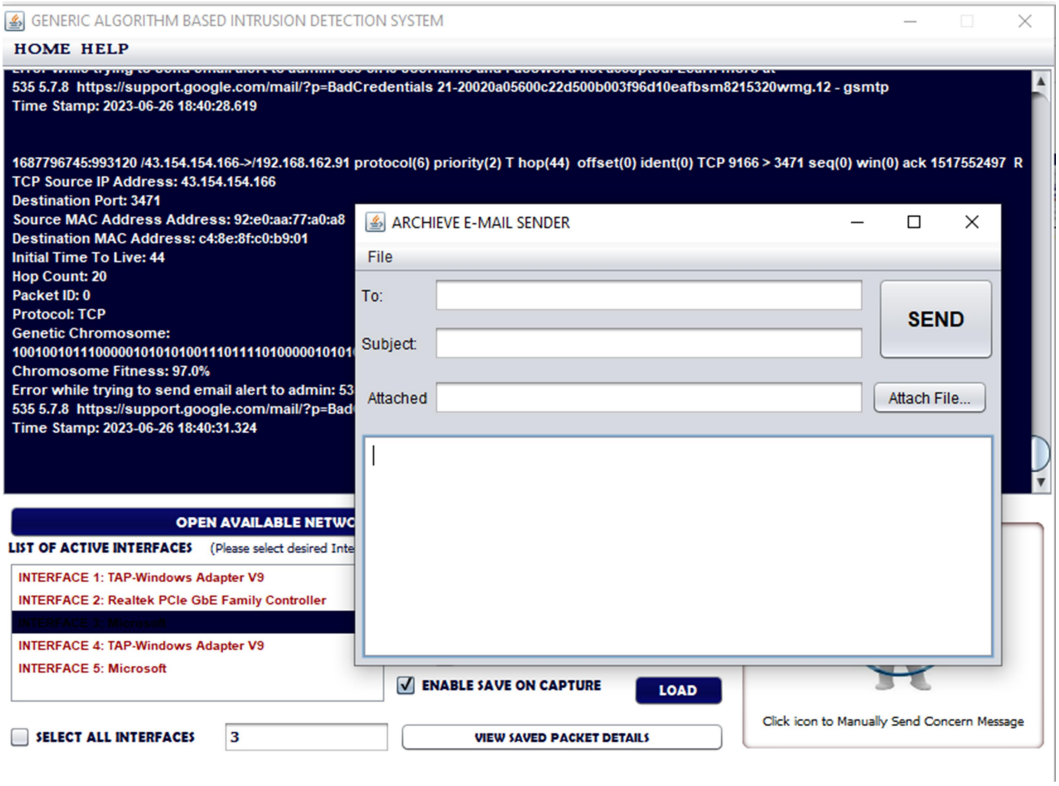

Figure 4.9 : L'interfacé 8**.**

Conclusion :

Dans ce chapitre, nous avons présenté l'implémentation de notre application basée sur les algorithmes génétiques pour la détection d'intrusion. L'objectif principal était de mettre en œuvre notre approche théorique et d'évaluer ses performances dans un environnement réel.

Nous avons utilisé le langage de programmation Java pour développer notre application, en exploitant les fonctionnalités et les bibliothèques disponibles pour la manipulation des données réseau, la génération de populations génétiques et l'évaluation des fonctions de fitness.

Conclusion Générale

Conclusion générale

En conclusion, ce mémoire a présenté une approche de détection d'intrusion basée sur les algorithmes génétiques pour résoudre le problème de l'usurpation d'adresse IP, également connu sous le nom d'IP spoofing. L'utilisation des algorithmes génétiques offre une méthode efficace pour générer et évaluer des solutions potentielles en utilisant une structure chromosomique et une fonction de fitness.

L'utilisation du langage de programmation Java pour implémenter notre système a été un choix judicieux en raison de sa popularité et de sa polyvalence. Java a permis une manipulation aisée des structures de données nécessaires à notre approche basée sur les algorithmes génétiques, et ses bibliothèques ont facilité le développement et l'optimisation du système de détection d'intrusion.

Cette recherche ouvre des perspectives intéressantes pour l'amélioration de la sécurité des réseaux informatiques. L'approche utilisée, basée sur les algorithmes génétiques, peut être utilisée comme une couche de sécurité supplémentaire pour contrer les attaques d'IP spoofing, renforçant ainsi la protection des systèmes et des données sensibles.

Cependant, il convient de noter que notre système de détection d'intrusion présente également certaines limites. Il est essentiel de continuer à travailler sur l'optimisation des performances et l'adaptation à des environnements de réseau complexes. De plus, l'évolution constante des attaques et des techniques de contournement nécessite une vigilance continue pour maintenir l'efficacité du système de détection.

Bibliographie

Bibliographie

[1]: Mohammed, Mohssen & Rehman, Habib-ur. (2015). Honeypots and Routers: Collecting Internet Attacks.

[2]: Kaspersky (2023) What is cyber security?, www.kaspersky.com. Disponible sur: https://www.kaspersky.com/resource-center/definitions/what-is-cyber-security (Accédé le : 30 Juin 2023).

[3]: Firewalls and Internet Security: Repelling the Wily Hacker, 2nd Edition, William Cheswick, Steven Bellovin, Aviel Rubin, John Fuller.

[4]: Christophe Bidan, Guillaume Hiet, Ludovic Mé, Benjamin Morin, Jacob Zimmermann. Vers une détection d'intrusions à fiabilité et pertinence prouvables. La Revue de l'électricité et de l'électronique, 2006, 9, 13 p.

[5]: Melanie Mitchell. 1998. An Introduction to Genetic Algorithms. MIT Press, Cambridge, MA, USA.

[6]: Majeed, P.G. & Kumar, S.. (2014). Genetic algorithms in intrusion detection systems: A survey. International Journal of Innovation and Applied Studies. 5. 233-240.

[7]: “Cyber security and Cyber war: What Everyone Needs to Know” par P.W. Singer et Allan Friedman (2014)

[8]: “Cyber security: A Practical Guide to the Law of Cyber Risk” par Paul Rosenzweig (2016)

[9] : “Cyber security: The Beginner's Guide” par Raef Meeuwisse (2018).

[10] : “Intrusion Detection Systems” par Rebecca Gurley Bace (2000).

[11]: “Intrusion Detection Systems: Concepts and Techniques” par Nong Ye (2012).

[12]: “An Overview of Intrusion Detection Systems: Categories and Approaches” par Wafaa Mousa et al. (2018)

[13]: “A Survey of Anomaly Detection Methods in Intrusion Detection Systems” par V. R. Dixit et al. (2017)

[14]: “Genetic Algorithms: Concepts and Designs”, par G.S. Mani et al. (2017)

[15]: “Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications” par Michael Affenzeller et al. (2009)

[16]: “Intrusion Detection System using Genetic Algorithm and Rule-Based Classifier” par Parag Thool et al. (2015)

[17]: “Intrusion Detection System using Genetic Algorithm for Feature Selection” par S. M. Mahajan et al. (2016)

[18] : “Genetic Algorithm based Intrusion Detection System: A Review” par K. Arul Kumar et al. (2017)

[19]: https://www.oracle.com/

[20] : https://www.microsoft.com/