République Algérienne Démocratique et Populaire
Ministére de l'enseignement supérieur et de la Recherche Scientifique

Université Abderrahmane Mira de Béjaïa
Faculté des Sciences Exactes
Département Informatique

Mémoire de fin de cycle
en vue de l'obtention du diplôme de master
en Administration et Sécurité des Réseaux

Filière : Informatique
Option : Administration et Sécurité des Réseaux

Thème :

Détection d'attaque APT en utilisant les techniques d'IA

Présenté par :
M. Zineddine MADI
M. Lounes MEDJKOUNE

Encadré par :
Mme. Malika YAICI

Maitre du Stage :
M. Seghir RAHMANI (Sonatrach)

Soutenu devant le jury composé de :

HOUHA Amel	Université de Béjaia	Présidente
BOUDRIES Abdelmalek	Université de Béjaia	Examinateur
CHEKRID Mohamed	Université de Béjaia	Examinateur
BATTAT Nadia	Université de Béjaia	Examinatrice

Année universitaire 2024–2025

Table des matières

Table des figures

Liste des abréviations

AI / IA	Artificial Intelligence / Intelligence Artificielle
APL	Armée Populaire de Libération (Chine)
APT	Advanced Persistent Threat (Menace Persistante Avancée)
CatBoost	Categorical Boosting
CNN	Convolutional Neural Network (Réseau de Neurones
CTI	Cyber Threat Intelligence
DL	Deep Learning (Apprentissage profond)
DNN	Deep Neural Network (Réseau de Neurones Profond) Convolutif)
HIDS	Host-based Intrusion Detection System
IDS	Intrusion Detection System
IP	Internet Protocol
K-NN	K-Nearest Neighbors (K plus proches voisins)
LightGBM	Light Gradient Boosting Machine
MITM	Man In The Middle (Attaque de l'homme du milieu)
ML	Machine Learning (Apprentissage automatique)
NIDS	Network-based Intrusion Detection System
OSINT	Open Source Intelligence
PCA	Principal Component Analysis (Analyse en Composantes Principales)
SHAP	SHapley Additive exPlanations
SIEM	Security Information and Event Management
SQL	Structured Query Language

SVM	Support Vector Machine (Machine à Vecteurs de Support)
XGBoost	Extreme Gradient Boosting

Introduction Générale

Les cybermenaces deviennent plus sophistiquées et ciblées. Les défenses traditionnelles ne suffisent plus.. – Eugene Kaspersky [10]

Nous vivons aujourd'hui dans un monde de plus en plus connecté, où Internet est devenu une infrastructure essentielle pour de nombreuses opérations : financières, commerciales, sanitaires, militaires ou éducatives. L'intégration des réseaux dans les systèmes d'information des entreprises et des organisations a conduit au concept de réseau d'entreprise, permettant aux utilisateurs d'accéder facilement à toutes les ressources internes, indépendamment de leur localisation.

Cependant, cette interconnexion généralisée a pour conséquence une exposition accrue aux cyberattaques. Les systèmes d'information sont devenus des cibles privilégiées pour des acteurs malveillants, souvent organisés, cherchant à voler des données sensibles, perturber les services ou nuire à l'image d'une entité.

Parmi les cybermenaces les plus sophistiquées et redoutées aujourd'hui, on trouve les attaques persistantes avancées (APT – Advanced Persistent Threats). Ces attaques ciblées se caractérisent par leur furtivité, leur durée et leur capacité à contourner les mécanismes de sécurité traditionnels. Contrairement aux attaques classiques, les APT sont souvent soutenues par des États ou des groupes hautement organisés, et elles visent à infiltrer un réseau sur le long terme pour y exfiltrer des données stratégiques sans être détectées.

Les solutions de sécurité classiques – telles que les antivirus ou les pare-feux – ne suffisent plus. Il est donc nécessaire de développer des mécanismes de détection intelligents capables d'identifier des comportements anormaux même en l'absence de signatures connues. L'intelligence artificielle (IA), et plus particulièrement les techniques d'apprentissage automatique (Machine Learning), s'imposent comme une solution prometteuse pour détecter ces menaces complexes en temps réel.

Dans ce contexte, l'objectif de ce mémoire est de concevoir, mettre en œuvre et évaluer une approche basée sur l'intelligence artificielle pour détecter les attaques persistantes avancées (APT) dans les réseaux d'entreprise. Après une revue des travaux existants sur le sujet, nous proposons une simulation réaliste d'une attaque APT dans un environnement virtuel, afin de

tester un système de détection intelligent capable d'identifier, en temps réel, des comportements anormaux malgré l'usage d'identifiants légitimes par l'attaquant.

Organisation du mémoire :
Ce mémoire est structuré comme suit :

- **Chapitre 1 :** Nous présentons les concepts fondamentaux de la sécurité des systèmes d'information, les objectifs de sécurité, ainsi qu'un panorama des principales cyberattaques. Une attention particulière est accordée aux attaques persistantes avancées (APT), en mettant en lumière leurs caractéristiques, leur cycle de vie, et quelques exemples réels.

- **Chapitre 2 :** Ce chapitre est consacré aux systèmes de détection d'intrusion (IDS). Nous y étudions leurs types (NIDS, HIDS), leurs méthodes de détection (signature, anomalie), ainsi que leurs limites face aux APT. Ensuite, nous introduisons les techniques d'intelligence artificielle (notamment le machine learning) utilisées pour améliorer la détection, avec un survol des modèles les plus pertinents.

- **Chapitre 3 :** Ce dernier chapitre adopte une approche pratique. Nous y résumons quatre articles scientifiques récents portant sur la détection d'APT à l'aide de l'intelligence artificielle. Sur cette base, nous développons une simulation ou une expérimentation visant à reproduire une détection d'APT dans un réseau d'entreprise. Ce chapitre présente la méthodologie suivie, les outils utilisés (ex : VirtualBox.), ainsi que les résultats obtenus.

On termine ce memoire par une conclusion et des perspectives.

CHAPITRE 1

Chapitre 1 : Fondements de la sécurité des systèmes informatiques

1.1 Introduction

La sécurité des systèmes d'information constitue aujourd'hui une préoccupation majeure pour les entreprises, les gouvernements et les particuliers . Les cybermenaces ne cessent d'évoluer, exploitant la moindre faille pour compromettre la confidentialité, l'intégrité et la disponibilité des données. Ce premier chapitre vise à introduire les notions fondamentales de la sécurité informatique, en mettant l'accent sur la sécurité des réseaux, les objectifs clés à atteindre, ainsi que les principales typologies d'attaques auxquelles un système peut être exposé. Une attention particulière sera portée à l'analyse des attaques persistantes avancées (APT), considérées comme l'une des formes les plus sophistiquées et redoutables de cybermenace.

1.2 Securité informatique

La sécurité informatique est l'ensemble des moyens mis en œuvre pour réduire la vulnérabilité d'un système contre les menaces accidentelles ou intentionnelles.il convient d'identifier les exigences fondamentales en sécurité informatique. Elles caractérisent ce a quoi s'attendent les utilisateurs de systèmes informatique en regard de la sécurité.

1.3 Sécurité des réseaux

La securite des réseaux est le domain de la cybersecurité qui se concentre sur la protection des reseaux et des systemes informatiques contre les cybermenaces et les cyberattaques interne ou externe. Elle s'articule autour de trois objectifs principaux : empecher l'acces non autorisé aus ressources du reseau, detecter et neutraliser les cyberattaques et les violations de securité en cours, et veiller est ce que les utilisateurs autorisés puissent acceder en toute securité aux ressources du reseau dont ils ont besoin, au moment ou ils en ont besoin. La securite reseau

protege l'integrité de l'infrastructur, des ressources et du trafic des reseaux afin de contrecarrer ces attaques et d'en minimiser les repercussions financieres et operationnelles [25].

1.4 Objectifs de la sécurité

Il convient d'identifier les exigences fondamentales en sécurité informatique, qui caractérisent ce à quoi s'attendent les utilisateurs de systèmes informatiques au regard de la sécurité :

1.4.1 La confidentialité

Seules les personnes habilitées doivent avoir accès aux données. Toute interception ne doit pas être en mesure d'aboutir, les données doivent être cryptées, seuls les acteurs de la transaction possèdent la clé de compréhension.

1.4.2 L'intégrité

Il faut garantir à chaque instant que les données qui circulent sont bien celles que l'on croit, qu'il n'y a pas eu d'altération (volontaire ou non) au cours de la communication. L'intégrité des données vise à garantir que celles-ci n'ont pas été altérées, volontairement ou non, pendant le stockage ou la transmission. Elle assure que les données restent exactes, complètes et fiables, telles qu'elles ont été créées ou reçues, leur précision, l'authenticité et la validité.

1.4.3 La disponibilité

Il faut s'assurer du bon fonctionnement du système, de l'accès à un service et aux ressources à n'importe quel moment. La disponibilité d'un équipement se mesure en divisant la durée durant laquelle cet équipement est opérationnel par la durée durant laquelle il aurait dû être opérationnel.

1.4.4 La non-répudiation

Une transaction ne peut être niée par aucun des correspondants. La non-répudiation de l'origine et de la réception des données prouve que les données ont bien été reçues. Cela se fait par le biais de certificats numériques grâce à une clé privée.

1.4.5 L'authentification

Elle limite l'accès aux personnes autorisées. Il faut s'assurer de l'identité d'un utilisateur avant l'échange de données.

1.5 Type D'attaques

1.5.1 Attaque par men-in-the-middle

Les attaques Man-in-the-Middle (MitM) représentent une violation de la cybersécurité permettant à un attaquant d'intercepter et de modifier les communications entre deux parties sans

leur consentement .

1.5.2 Attaque de phishing

Les attaques de phishing sont des tentatives malveillantes visant à tromper les utilisateurs en les incitant à divulguer des informations personnelles, telles que des mots de passe, des numéros de carte de crédit ou des données bancaires.
Elles se manifestent souvent sous forme de mails ou de messages imitant ceux d'organisations légitimes.

1.5.3 Attaque de Whaling

Le whale-phishing, ou hameçonnage de baleines, est une forme ciblée de phishing qui vise spécifiquement les hauts dirigeants ou les membres clés d'une organisation.
Ces attaques sont conçues pour dérober des informations hautement sensibles ou obtenir un accès non autorisé à des systèmes critiques.
Les cybercriminels utilisent des techniques d'ingénierie sociale sophistiquées, s'appuyant sur des recherches approfondies pour personnaliser leurs messages et les rendre plus convaincants. xLe but est souvent d'exploiter l'autorité et l'accès privilégié de ces individus pour mener à bien des fraudes financières ou des fuites de données.

1.5.4 Attaque par injection SQL

L'injection SQL (SQL Injection) est une technique d'attaque qui exploite une faille de sécurité dans l'application d'une base de données d'un site web.
Elle permet à l'attaquant d'insérer ou « injecter » un code SQL malveillant dans une requête SQL, modifiant ainsi la logique de programmation prévue pour exécuter des actions non autorisées, telles que l'accès à des données sensibles, leur modification ou leur suppression .
Cette attaque peut avoir des conséquences désastreuses, notamment la fuite d'informations confidentielles comme les données personnelles des utilisateurs, les mots de passe et les informations de carte de crédit [13].

1.5.5 Attaque APT

Une attaque persistante avancée (APT) est une forme de cyberattaque sophistiquée et ciblée, menée sur une longue période, qui vise principalement à infiltrer un système informatique de manière discrète afin d'y rester cachée tout en exfiltrant progressivement des informations sensibles. Contrairement aux attaques classiques, souvent rapides et visibles, une APT se distingue par sa durée, sa furtivité et l'usage de techniques élaborées permettant à l'attaquant de maintenir un accès continu et durable aux ressources de la victime.

1.6 Attaque persistante APT

Parmi les nombreuses formes de cyberattaques, certaines se distinguent par leur complexité, leur discrétion et leur durée d'action. C'est notamment le cas des attaques persistantes avancées

(APT), qui représentent aujourd'hui une menace majeure pour les organisations stratégiques. Dans ce qui suit, nous allons nous intéresser en détail à ce type d'attaque, en mettant en lumière ses caractéristiques, ses objectifs, ainsi que les techniques utilisées par les attaquants.
Une menaces persistantes avancée fait reference a une attaque qui continue, secretement, en utilisant des methodes de piratages innovantes pour acceder a un système et rester a l'interieur pendant une longue periode.
Comprendre ce qu'est une APT implique egalment de connaitre ses cibles. Les attaques APT sont connus pour s'emparer des pays et des grandes organisations, ainsi que des grandes entreprises, pour exfilter des grandes informations, progressivement et systematiquement, sur de longues periodes avant de se retirer . Le temps passé au sein du système informatique d'une organisation est appelée « temps d'arret ».
Souvent, ces attaquants se concentrent sur les organisations ou les entreprises aux États-Unis ou dans d'autres pays développés. Bien qu'ils puissent s'attaquer à des cibles de grande valeur, les attaquants essaient souvent d'y accéder en utilisant des entreprises ou organisations plus petites que les cibles utilisent pour faire des affaires. Il peut s'agir d'entreprises le long de leurs lignes d'approvisionnement ou d'organisations avec lesquelles elles collaborent pour atteindre leurs objectifs.
Les attaquants APT ont tendance à se pencher sur l'obtention de renseignements ou d'autres informations vitales pour endommager un système plus grand, exploiter ou faire paraître une organisation mauvaise, ou obtenir un avantage concurrentiel.
Les attaques APT peuvent être lancées par une seule personne ou par un groupe plus important. Dans certains cas, l'attaque est réalisée par une agence parrainée par le gouvernement. Ils se concentrent généralement sur l'attaque de la capacité d'une organisation à fonctionner efficacement ou à atteindre ses objectifs. Ils pourraient également viser à obtenir des renseignements qui peuvent être utilisés pour nuire à la cible ou pour renforcer la mission du groupe, ou celle de son employeur.
Il est important de noter que même les petites et moyennes entreprises doivent prendre des précautions. Les entreprises de tous les principaux secteurs d'activité ont été ciblées. L'avantage qu'un sponsor d'attaque peut gagner sur une entreprise ou au sein d'un segment de marché vaut souvent le prix qu'il paie pour une attaque APT.

1.6.1 Processus de l'attaque

Les auteurs d'attaques persistantes avancées (APT) suivent généralement une série d'étapes méthodiques et coordonnées dans le but d'établir et de maintenir un accès durable et discret au sein du réseau ciblé. Ces étapes, soigneusement planifiées, leur permettent d'exfiltrer des données sensibles tout en échappant à la détection des mécanismes de sécurité traditionnels :

1.6.1.1 Accès au réseau

Pour atteindre la cible, les systèmes cibles APT sont regroupés sur Internet, soit en envoyant un message de phishing ciblé (une attaque personnaliser), soit en utilisant une faille de sécurité qui leur permet d'introduire des logiciels malveillants.

1.6.1.2 Établir un point d'ancrage

Une fois l'accès initial au système obtenu, les attaquants approfondissent leur reconnaissance interne et commencent à exploiter les logiciels malveillants qu'ils ont déployés. Leur objectif est d'installer des mécanismes de persistance (techniques permettant de rester dans le système sur la durée), notamment des portes dérobées (backdoors, accès cachés non autorisés) et des tunnels de communication chiffrés (connexions secrètes protégées par un codage des données), leur permettant de maintenir une présence continue sans être détectés.
Pour dissimuler leurs activités, ils ont recours à des techniques avancées d'évasion (stratégies pour ne pas être repérés par les systèmes de sécurité), telles que la réécriture de code, le chiffrement des communications et la modification des fichiers système.

1.6.1.3 Propagez l'attaque

Après avoir pénétré dans le réseau cible, les acteurs de l'APT peuvent, entre autres, pirater le mot de passe des privilèges administratifs, ce qui leur permet de mieux contrôler le système et d'accéder à d'autres niveaux.

1.6.1.4 Se déplacer dans le système

Après avoir violé les systèmes cibles et obtenu des privilèges administratifs, ils peuvent se déplacer dans le réseau d'entreprise comme ils le souhaitent. En outre, ils peuvent tenter d'accéder à d'autres serveurs ou à d'autres zones protégées du réseau

1.6.1.5 Déployer l'attaque

À ce stade, les pirates centralisent, cryptent et compriment les données pour les filtrer.

1.6.1.6 Filtrage des données

Les pirates informatiques collectent les données et les transmettent à leur propre système.

1.6.1.7 L'accès n'est pas détecté

Les cybercriminels peuvent répéter ce processus pendant longtemps tout en restant invisibles, ou ils peuvent créer une porte dérobée pour accéder au système s'ils le souhaitent.

Contrairement à la plupart des cyberattaques classiques, qui utilisent des outils automatisés visant un grand nombre de victimes de manière opportuniste, les campagnes d'attaques persistantes avancées (APT) reposent sur des méthodes ciblées, minutieusement conçues pour infiltrer des organisations précises. Ces attaques sont généralement précédées d'une phase de reconnaissance approfondie (collecte d'informations sur la cible), permettant aux attaquants d'adapter leurs techniques à l'environnement spécifique du système visé.
Les attaques APT se distinguent également par leur durée : elles s'étendent souvent sur plusieurs mois, voire des années, les assaillants cherchant à rester invisibles le plus longtemps possible

pour extraire un maximum de données sensibles ou compromettre durablement les systèmes. Cette persistance, alliée à leur sophistication technique, rend les attaques APT beaucoup plus difficiles à détecter et à neutraliser que les attaques ordinaires.

1.6.2 Exemples connus d'attaques APT

Les informations suivantes sont principalement tirées du rapport publié par l'entreprise Mandiant [14], l'un des premiers à documenter en profondeur les activités d'un groupe APT.

- **Stuxnet (2010)** Attaque contre le programme nucléaire iranien.
- **APT1 (attribuée à un groupe chinois, 2013)** Espionnage de plus de 140 entreprises à travers le monde.
- **Sony Pictures Hack (2014)** Attribuée à la Corée du Nord.
- **SolarWinds (2020)** Attaque sur la chaîne d'approvisionnement touchant des agences gouvernementales américaines.
- **Equation Group (lié à la NSA)** Utilisation de malwares très sophistiqués.

APT1 – Un des premiers groupes APT révélés publiquement

a) **Présentation générale**

- Nom de l'attaque / groupe : APT1 (Advanced Persistent Threat 1).
- Alias : Comment Crew, Comment Group, Shanghai Group, Byzantine Candor.
- Origine présumée : Chine
- Organisation liée : Unité 61398 de l'Armée Populaire de Libération (APL), basée à Shanghai
- Révélé par : Mandiant, entreprise américaine de cybersécurité, dans un rapport publié en février 2013 [15]
- Durée estimée des opérations : 2006 à 2013 (au moins)
- Nombre de victimes : Environ 141 organisations dans 20 secteurs industriels à travers le monde

b) **Cibles visées**

APT1 ciblait des entreprises et des organisations stratégiques dans les secteurs suivants :

- Énergie
- Aérospatial
- Télécommunications
- Finance
- Technologies de l'information
- Transports
- Défense et sécurité
- Produits chimique
- Fabrication industrielle

- Services juridiques

Les cibles étaient principalement situées aux **États-Unis**, mais aussi en **Europe** et en **Asie**.

c) **Techniques utilisées**

i) Ingénierie sociale et spear phishing (Hameçonnage ciblé) **Ingénierie sociale et spear phishing (Hameçonnage ciblé)**
L'ingénierie sociale désigne un ensemble de techniques psychologiques utilisées par les cyberattaquants pour manipuler les individus et les inciter à divulguer des informations sensibles ou à exécuter des actions compromettantes, comme ouvrir une pièce jointe malveillante ou cliquer sur un lien piégé. Contrairement aux attaques purement techniques, cette approche cible le maillon le plus vulnérable de la chaîne de sécurité : l'humain.
Dans ce contexte, les groupes APT utilisent une méthode appelée spear phishing, ou hameçonnage ciblé. Contrairement au phishing classique, qui envoie des messages génériques à un grand nombre de personnes, le spear phishing consiste à envoyer des emails spécifiquement conçus pour une personne ou un groupe restreint. Ces messages sont souvent très convaincants, car ils s'appuient sur des informations recueillies à l'avance (nom, fonction, projet en cours, etc.).
Les techniques employées incluent notamment :

- L'envoi d'emails personnalisés contenant des pièces jointes piégées (fichiers infectés par des malwares) ou des liens vers des sites frauduleux, déguisés en documents légitimes (factures, rapports, demandes internes, etc.).
- Le ciblage stratégique de personnes clés ayant accès à des systèmes critiques (comme les administrateurs réseau, les responsables financiers ou les dirigeants), afin de maximiser les chances de compromission d'un système ou d'exfiltration de données sensibles.

Cette étape constitue souvent le point d'entrée initial dans les attaques APT, ouvrant la voie à une compromission plus large du réseau.

ii) Exploitation de failles logicielles **Exploitation de failles logicielles**
Les groupes APT exploitent souvent des vulnérabilités (ou failles de sécurité) présentes dans les logiciels couramment utilisés, dans le but de contourner les protections et de prendre le contrôle d'un système cible.

- **Vulnérabilités non corrigées (zero-day)**
  Une faille zero-day est une vulnérabilité inconnue du public et des développeurs du logiciel au moment de son exploitation par les attaquants. Le terme "zero-day" signifie que les éditeurs de logiciels ont zéro jour pour corriger le problème avant qu'il ne soit exploité. Ces failles sont extrêmement dangereuses, car aucun correctif ou antivirus ne peut les détecter immédiatement.
- **Exemple :**
  En 2021, une faille zero-day dans Microsoft Exchange Server (CVE-2021-26855)

a été exploitée pour permettre à des attaquants de lire des emails et d'exécuter du code à distance sur les serveurs vulnérables.

- **Exploitation de failles connues**
  Outre les zero-days, les APT utilisent aussi des failles déjà connues, souvent présentes sur des systèmes qui n'ont pas été mis à jour à temps. Ces vulnérabilités sont documentées publiquement, et des patchs de sécurité sont généralement disponibles — mais de nombreuses organisations négligent leur installation rapide.
- **Exemples de failles connues dans Windows**

  - CVE-2017-0144 (EternalBlue) : utilisée dans l'attaque WannaCry. Elle permettait une exécution de code à distance via SMBv1 (protocole de partage de fichiers).
  - CVE-2019-0708 (BlueKeep) : vulnérabilité critique dans le protocole RDP (Remote Desktop Protocol), pouvant être exploitée pour lancer une attaque sans interaction de l'utilisateur.
  - CVE-2020-0601 (CurveBall) : faille dans la validation des certificats numériques dans Windows, affectant la cryptographie et permettant des attaques de type "man-in-the-middle".

  Ces failles sont souvent exploitées par les attaquants APT pour pénétrer dans un réseau, installer des logiciels malveillants et prendre progressivement le contrôle du système.

iii) **Installation de malwares**

- Les malwares typiques utilisés incluent :
  - **WEBC2 :** Outil de Command and Control (C2) basé sur le web. Il permet aux attaquants de communiquer avec les machines compromises via des requêtes HTTP, souvent en se cachant dans du trafic web légitime.
  - **GETMAIL :** Malware conçu pour intercepter et extraire des courriels depuis des boîtes mail compromises. Il est utilisé pour voler des informations sensibles échangées par email.
  - **MCCLIENT :** Un cheval de Troie d'accès à distance (RAT), qui offre aux attaquants un contrôle complet à distance du système infecté, y compris l'exécution de commandes et le transfert de fichiers.
  - **NEWSREELS :** Utilisé pour la surveillance du réseau local, ce malware aide les attaquants à détecter d'autres systèmes vulnérables dans le réseau et à planifier leur propagation.
  - **HOOKBOX :** Outil d'exfiltration de données, conçu pour extraire discrètement des fichiers depuis les systèmes compromis vers les serveurs des attaquants.
- Ces outils servaient à :
  - Maintenir l'accès au réseau

- Exfiltrer des données
- Contrôler les systèmes compromis

iv) **Commande et Controle**

- Les machines infectées communiquaient avec des serveurs distants pour recevoir des ordres .
- Ces serveurs utilisaient souvent des noms de domaine déguisés (DNS dynamique, hébergement sur des serveurs compromis).

v) **Exfiltration de données**

- Données compressées et chiffrées avant l'exfiltration.
- Transfert en dehors des heures de bureau pour éviter la détection.

vi) **Persistance dans les systèmes**

- Création de comptes administrateurs cachés.
- Modification des politiques de sécurité du système.
- Utilisation de tunnels chiffrés pour maintenir une présence invisible.

d) **Étapes typiques de l'attaque APT1**

- **Reconnaissance :** Identification des cibles via recherche OSINT (Open Source INTelligence) (réseaux sociaux, documents publics…).
- **Intrusion initiale :** Spear phishing ou exploitation d'une vulnérabilité.
- **Installation :** Déploiement de malware et prise de contrôle.
- **Mouvement latéral :** Propagation vers d'autres machines ou serveurs internes.
- **Escalade de privilèges :** Pour obtenir un contrôle total.
- **Exfiltration de données :** Transfert discret vers les serveurs des attaquants.
- **Maintien de l'accès :** Installation de backdoors ou d'outils d'accès à distance.

e) **Révélation par l'entreprise Mandiant (spécialisée en cybersécurité)**
En février 2013, l'entreprise américaine Mandiant, spécialisée dans la cyberdéfense et l'analyse des menaces, publie un rapport choc de 60 pages intitulé :
"APT1 : Exposing One of China's Cyber Espionage Units" (APT1 : Révélation d'une unité de cyberespionnage chinoise)
Ce rapport constitue une étape majeure dans l'histoire de la cybersécurité, car il fournit pour la première fois des preuves techniques et circonstanciées liant un groupe de cyberattaquants à un État-nation.

Contenu et éléments clés du rapport

- Attribution claire des attaques à l'unité 61398 de l'Armée Populaire de Libération (APL) chinoise, localisée à Shanghai.
- Corrélations techniques précises : adresses IP utilisées dans les attaques, noms de domaine contrôlés par les attaquants, et heures d'activité correspondant au fuseau horaire de Shanghai, suggérant une opération de bureau organisée.
- Identification de membres spécifiques du groupe APT1, avec des photos, pseudonymes, profils d'activité en ligne et éléments de leur infrastructure numérique.

- Estimation des dégâts : des années de vol de propriété intellectuelle touchant au moins 141 entreprises dans 20 secteurs industriels, entraînant des pertes de millions à plusieurs milliards de dollars.
- Ciblage stratégique d'organisations américaines et internationales dans des domaines sensibles comme l'énergie, la défense, les technologies de l'information, ou encore la finance.

Cette publication a eu un retentissement international, car c'était la première fois qu'un acteur privé accusait publiquement un État (la Chine) de cyberespionnage à grande échelle, en fournissant des preuves détaillées et vérifiables.

f) **Conséquences de l'affaire APT1**

i) **Politique**

- Tension majeure entre les États-Unis et la Chine.
- Première fois qu'un État désignait ouvertement une force militaire étrangère comme responsable d'une campagne de cyberespionnage.

ii) **Cybersécurité**

- Révélation d'un nouveau niveau de menace étatique.
- Les entreprises commencent à prendre la cybersécurité plus au sérieux.
- Développement de nouvelles techniques de défense contre les APT.

iii) **Réaction chinoise**

- Déni total de toute implication.
- Accusation de "propagande américaine".

g) **Ce qu'on peut apprendre d'APT1**

- Les APT ne sont pas que des menaces techniques : ce sont des outils géopolitiques.
- Même des entreprises bien protégées peuvent être infiltrées par un adversaire persistant.
- Il faut une cybersécurité proactive et multi-niveaux : détection, réponse, prévention, sensibilisation.
- Le cyberespace est devenu un champ de bataille invisible entre puissances mondiales.

1.7 Conclusion

Ce chapitre a permis de poser les bases essentielles pour comprendre les enjeux de la sécurité informatique, notamment à travers les principes fondamentaux comme la confidentialité, l'intégrité ou encore l'authentification. En explorant les types d'attaques les plus fréquents, telles que le phishing ou l'injection SQL, nous avons pris conscience de la diversité et de la complexité des menaces. L'étude approfondie des attaques persistantes avancées (APT) révèle à quel point certaines cybermenaces peuvent être stratégiques, discrètes et destructrices à long terme. Ces éléments justifient l'importance d'une vigilance constante et d'une adaptation continue des dispositifs de sécurité, notamment dans les réseaux d'entreprise qui constituent des cibles privilégiées.

CHAPITRE 2

Chapitre 2 : L'évolution des systèmes de détection d'intrusion face aux cybermenaces modernes

2.1 Introduction

Dans un contexte où les cyberattaques deviennent de plus en plus sophistiquées et fréquentes, les mécanismes classiques de détection d'intrusions tels que les IDS (Intrusion Detection Systems) représentent un pilier fondamental de la cybersécurité. Ces systèmes permettent de surveiller en permanence les flux de données et les activités des systèmes afin de repérer tout comportement anormal ou malveillant. Cependant, face à des menaces toujours plus évoluées, les méthodes traditionnelles trouvent parfois leurs limites. C'est dans ce cadre que l'intelligence artificielle, et en particulier l'apprentissage automatique, offre de nouvelles perspectives pour renforcer l'efficacité des systèmes de détection. Ce chapitre propose ainsi un panorama des systèmes de détection classiques, puis explore les apports des techniques d'intelligence artificielle dans l'identification proactive des intrusions.

2.2 Systèmes de détection classique des attaques

Avec l'augmentation constante des cyberattaques, la sécurité des systèmes d'information est devenue un enjeu majeur pour les entreprises et les organisations. Parmi les solutions de protection, les systèmes de détection d'intrusion (IDS - Intrusion Detection Systems) jouent un rôle crucial en surveillant les réseaux et les systèmes informatiques afin d'identifier les activités malveillantes. Ces systèmes analysent le trafic, détectent les comportements suspects et alertent les administrateurs en cas de menace potentielle. Selon leur mode de fonctionnement, ils peuvent être basés sur l'analyse des signatures d'attaques connues ou sur la détection d'anomalies comportementales. Bien que les IDS constituent une première ligne de défense efficace, ils sont souvent combinés avec d'autres mécanismes de sécurité pour améliorer leur capacité à contrer les cybermenaces émergentes.

2.2.1 Système de detection d'intrusion (IDS)

Un système de détection d'intrusions (« Intrusion Détection System » ou IDS) est un appareil ou une application qui alerte l'administrateur en cas de faille de sécurité, de violation de règles ou d'autres problèmes susceptibles de compromettre son réseau informatique. Un IDS est un mécanisme écoutant le trafic réseau de manière furtive afin de repérer des activités anormales ou suspectes et permettant ainsi d'avoir une action de prévention sur les risques d'intrusion [3, 24].

2.2.2 Objectif d'un système de detection des intrusions

L'objectif principal d'un système de détection des intrusions est de renforcer la sécurité de votre infrastructure réseau. Il identifie les potentielles violations de sécurité avant qu'elles ne causent des dommages importants.
Avec la complexité et la fréquence croissantes des cyberattaques, le NIDS est devenu essentiel pour protéger votre infrastructure numérique.

2.2.3 NIDS (Système de détection d'intrusion basé sur le réseau

Un NIDS est un IDS qui surveille le trafic circulant sur un réseau pour identifier des attaques ou des comportements malveillants.

— **Caractéristiques**

- Analyser le trafic en temps réel.
- Détecter les attaques reseaux.
- surveiller les paquets entrants et sortants pour identifiers les anomalies.

Fonctionne sur un point stratégique du réseau (routeur, switch).

2.2.3.1 Fonctionement des systèmes de détection d'intrusion réseau

Un système de détection des intrusions surveille le trafic réseau, analyse les paquets et les compare à des signatures d'attaques connues ou à des schémas comportementaux suspects. Voici les principales étapes et composants impliqués :

a) **Collecte des données**
Le système de détection des intrusions capture le trafic grâce à des capteurs intégrés dans l'infrastructure réseau. Ces capteurs peuvent être matériels ou des logiciels installés sur des dispositifs existants.

b) **Analyse du trafic**
Une fois les données collectées, le système de détection des intrusions utilise des algorithmes avancés pour analyser les paquets réseau. Cette analyse peut comprendre :

- l'examen des en-têtes des paquets pour détecter des adresses IP ou des numéros de port suspects ;

- l'inspection du contenu des paquets pour identifier des schémas d'attaques connus ou du contenu malveillant ;
- l'analyse des flux de trafic pour repérer des comportements anormaux.

c) **Méthodes de détection**
Le système de détection des intrusions utilise différentes méthodes pour identifier les menaces potentielles. Ces méthodes seront détaillées plus loin dans l'article. En général, il fonctionne soit de manière passive, soit active. Lorsqu'il fonctionne passivement, il n'interfère pas avec le trafic réseau. En mode actif, il modifie le trafic pour bloquer les activités malveillantes, ce qui peut toutefois perturber le trafic légitime [13, 16].

d) **Génération d'alertes**
Lorsqu'une activité suspecte est détectée, le NIDS génère des alertes avec des détails sur la menace identifiée. Ces alertes sont généralement envoyées vers une console de gestion centralisée ou un système SIEM (Security Informatique and Event Management) pour une analyse approfondie et une réponse rapide.

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

Un HIDS est un IDS installé directement sur un hôte (ordinateur, serveur) et qui surveille les activités internes du système.

— **Caractéristiques**

- Analysez les fichiers journaux (logs) du système.
- Surveillez les modifications des fichiers critiques.
- Détectez les tentatives de connexion suspectes sur la machine.
- Protège contre les menaces internes (malware, accès non autorisé).

2.3 Système de détection des attaques base sur l'IA

Pour la détection des menaces en cybersécurité avec l'IA, on utilise souvent des techniques supervisées ou non supervisées d'apprentissage automatique, ainsi que des modèles de deep learning capables d'analyser en temps réel des flux de données massifs.

2.3.1 L'apprentissage automatique(ML)

L'apprentissage automatique (Machine Learning, ML) apporte une dimension supplémentaire dans la détection des attaques. En s'appuyant sur des algorithmes avancés, les WAF équipés de ML analysent des volumes massifs de données pour déceler des modèles subtils et complexes indiquant des comportements malveillants [6, 15, 22].
Cette technologie permet :

- **Une personnalisation accrue :** Adaptation aux spécificités de chaque application.
- **Une détection proactive :** Identification des menaces émergentes avant qu'elles ne deviennent critiques.

- **Une évolution continue :** Amélioration constante de l'efficacité de détection au fil du temps.

L'intégration de ces différentes méthodes dans les WAF constitue une approche multidimensionnelle, essentielle pour contrer efficacement la diversité et la complexité des attaques web actuelles.

2.3.2 Méthodes d'Intelligence Artificielle utilisées pour la détection d'intrusions

L'IA ne se limite pas au machine learning ; il existe plusieurs techniques qui peuvent être intégrées dans un IDS pour améliorer la précision, réduire les faux positifs, et détecter des attaques inconnues. Voici quelques-unes des méthodes les plus utilisées :

2.3.2.1 Apprentissage supervisé

L'apprentissage supervisé repose sur un ensemble de données d'apprentissage étiquetées contenant des exemples d'activités normales et malveillantes. Des algorithmes tels que :

- SVM (Support Vector Machines)
- Random Forest
- k-Nearest Neighbors (k-NN) sont couramment utilisés pour classer le trafic réseau comme étant sûr ou dangereux.

2.3.2.2 Apprentissage non supervisé

Cette méthode est utile lorsque les données ne sont pas étiquetées. Les algorithmes essaient de détecter des anomalies par regroupement ou réduction de dimension :

- Clustering (K-means, DBSCAN)
- PCA (Analyse en Composantes Principales) Ce type d'apprentissage permet de détecter des comportements inhabituels sans avoir besoin d'exemples d'attaques.

2.3.2.3 Apprentissage semi-supervisé

Combinaison des deux méthodes précédentes, il est efficace quand on dispose de peu de données étiquetées, en les complétant par des données non étiquetées pour améliorer la détection.

2.3.2.4 Réseaux de neurones et Deep Learning

Les réseaux de neurones profonds (DNN) ou les réseaux convolutifs (CNN) sont capables d'extraire automatiquement des caractéristiques complexes à partir des données réseau, rendant possible la détection de menaces très subtiles, voire inédites. Ces approches sont particulièrement efficaces sur de grands volumes de données.

2.3.2.5 Systèmes hybrides

Les systèmes hybrides combinent plusieurs techniques d'IA, ou une méthode classique avec une méthode basée sur l'IA, afin de bénéficier à la fois de la précision des signatures et de la détection des anomalies non connues.

2.4 Conclusion

Ce chapitre a mis en lumière les mécanismes classiques de détection des intrusions à travers les IDS, qu'ils soient basés sur le réseau (NIDS) ou sur l'hôte (HIDS), ainsi que leur rôle fondamental dans la protection des infrastructures informatiques. Néanmoins, avec l'évolution constante des techniques d'attaque, il est devenu essentiel de recourir à des approches plus intelligentes et adaptatives. L'intégration de l'intelligence artificielle, en particulier via le machine learning, permet de repérer des menaces complexes et inconnues avec une meilleure précision. Cela marque une évolution majeure vers des systèmes de cybersécurité plus autonomes, capables d'anticiper et de neutraliser des intrusions toujours plus furtives.

CHAPITRE 3

Chapitre 3 : Etat de l'art

3.1 Introduction et problématique

Aujourd'hui, les attaques persistantes avancées (APT) représentent l'une des menaces les plus redoutables pour la sécurité des systèmes d'information des entreprises. Contrairement aux attaques classiques, souvent bruyantes et rapidement détectables, les APT se distinguent par leur furtivité, leur persistance dans le temps, et leur capacité à exfiltrer des données sensibles sans éveiller les soupçons. Elles visent généralement des organisations spécifiques — institutions gouvernementales, entreprises stratégiques ou centres de recherche — avec pour objectif l'espionnage, le sabotage ou le vol d'informations critiques.

Ces attaques s'appuient fréquemment sur des techniques d'ingénierie sociale, telles que le phishing, pour piéger les utilisateurs légitimes et obtenir leurs identifiants d'accès. Une fois infiltré, l'attaquant se dissimule dans le trafic réseau légitime, adopte un comportement proche de celui d'un utilisateur normal, et échappe ainsi aux mécanismes classiques de détection. C'est précisément cette capacité à opérer sous couverture qui rend les APT particulièrement difficiles à détecter, même avec des outils de sécurité traditionnels.

Face à ce constat, une question centrale émerge : Comment un système d'information peut-il détecter, en temps réel, une compromission d'un compte utilisateur suite à une attaque de type APT, alors que l'attaquant utilise des identifiants parfaitement valides ?

Pour tenter d'apporter une réponse concrète à cette problématique, nous avons conçu et mis en œuvre un environnement de simulation réaliste, basé sur VirtualBox[19], dans lequel une attaque APT est reproduite au sein d'un réseau d'entreprise virtuel. Cette simulation, s'appuyant sur des machines virtuelles configurées avec pfSense[17], Ubuntu Server[4] et Kali Linux[18], permet de tester différentes méthodes de surveillance et de détection, notamment à travers l'analyse des journaux d'authentification SSH, l'examen des logs système (syslog), et la surveillance des connexions réseau.

Avant d'entrer dans cette phase pratique, nous avons effectué une revue de littérature à partir de plusieurs articles scientifiques, rapports de recherche et retours d'expérience. L'objectif était de comprendre en profondeur les techniques utilisées par les attaquants APT, d'identifier les faiblesses des systèmes classiques de protection, et de découvrir les nouveaux leviers technologiques, tels que l'intelligence artificielle, les IDS (Intrusion Detection Systems), ou les plateformes SIEM (Security Information and Event Management).

3.2 Analyse des approches existantes

Le premier article étudié présente RANK, une architecture innovante basée sur l'intelligence artificielle, conçue pour améliorer la détection des attaques persistantes avancées dans les réseaux d'entreprise, en automatisant la corrélation et la priorisation des alertes de sécurité

3.2.1 RANK – AI-assisted End-to-End Architecture for Detecting Persistent Attacks in Enterprise Networks

Dans le contexte actuel marqué par la croissance du volume d'alertes de sécurité et la complexité croissante des cyberattaques, l'identification efficace des attaques persistantes avancées (APT) est devenue un enjeu majeur. L'article présente RANK, une architecture de bout en bout intégrant des techniques d'intelligence artificielle pour automatiser la détection des incidents dans les réseaux d'entreprise[23]. À travers l'utilisation de graphes et de modèles probabilistes, cette architecture vise à optimiser la corrélation et la priorisation des alertes, tout en réduisant la charge cognitive des analystes en cybersécurité.

1. **Objectif**
   L'objectif est de proposer RANK, une architecture automatisée de bout en bout basée sur l'IA pour :
   - - Réduire le nombre d'alertes que les analystes en cybersécurité doivent examiner.
   - Extraire automatiquement des incidents cohérents à partir d'alertes multiples générées par les systèmes IDS/UEBA.
   - Utiliser des graphes pour représenter et corréler les alertes, puis scorer les incidents en fonction de leur pertinence dans un contexte MITRE ATTCK
2. **Methodes IA utilisées**

   - Alert Templating (clustering intelligent) pour fusionner les alertes redondantes.
   - Graphes d'alerte et partitionnement de graphe optimisé.
   - Factor Graphs (graphe factoriel probabiliste) : utilisés pour calculer un score probabiliste pour chaque tactique MITRE présente dans un incident, en tenant compte de l'ordre logique des événements.
   - Utilisation de l'apprentissage non supervisé et graph-based optimization pour segmenter efficacement les alertes en incidents exploitables.

3. **Dataset**
    - DARPA 2000 Intrusion Detection Dataset
        - Gratuit et public
        - Classique mais limité en scénarios réalistes modernes.
    - Dataset privé d'un réseau d'entreprise
        - Non public
        - Composé d'environ 77 000 alertes collectées sur un mois via Suricata (IDS open-source)
4. **Programme**
    - Étapes de l'architecture RANK
        - Templating and Merging (modélisation et fusion) : réduction massive des alertes (facteur 100).
        - Construction de graphe d'alerte (Alert graph) : relations pondérées selon les tactiques MITRE et corrélations IP.
        - Partitionnement de graphe (Graph partitioning) :
            - Méthode classique de détection de communautés (community detection) via Louvain.
            - Méthode d'optimisation (optimization approach) avec contraintes sur :
                - le nombre d'alertes par incident,
                - la diversité tactique (tactics),
                - les hôtes impliqués.
    - Technologies utilisées
        - Apache Spark (framework Big Data) pour la phase MapReduce.
        - NetworkX pour le traitement de graphes.
        - Solveur CBC pour les modèles MILP.

Le deuxième article analysé combine des techniques de Boosting avec l'intelligence artificielle explicable(XAI), afin d'améliorer la détection des attaques APT tout en assurant la transparence des décisions prises par les modèles d'IA.

3.2.2 Advanced Persistent Threat Identification with Boosting and Explainable

La compréhension du comportement des modèles d'intelligence artificielle appliqués à la cybersécurité est essentielle pour assurer leur adoption et leur fiabilité. Cet article propose une approche novatrice combinant des algorithmes de Boosting avec des méthodes d'intelligence artificielle explicable (XAI). L'objectif est non seulement d'améliorer la détection des APT[9] grâce à des modèles performants tels que XGBoost et CatBoost, mais également de fournir des explications compréhensibles sur les décisions prises par ces modèles, notamment via la méthode SHAP.

1. **Objectif**
   Ce travail vise à devlopper un modele de detection efficace pour les APT en conbinant :
   - Des méthodes d'ensemble basées sur le boosting (boosting ensemble methods),
   - Et l'intelligence artificielle explicable (XAI – Explainable Artificial Intelligence).

   Le but est :
   - Améliorer la précision des systèmes de détection,
   - Et fournir des explications interprétables aux résultats pour les experts en cybersécurité
2. **Methodes IA utilisées**
   Les autres ont exploré plusieurs algorithmes de Boosting (renforcement) :
   - AdaBoost (Adaptive Boosting),
   - Gradient Boosting,
   - CatBoost (Category Boosting),
   - LightGBM (Light Gradient Boosting Machine),
   - XGBoost (Extreme Gradient Boosting).

   En plus ils ont utilisés XAI (Intelligence Artificielle Explicable), notamment :
   - **SHAP (SHapley Additive exPlanations),** une méthode d'explication basée sur la théorie des jeux (game theory) qui permet de mesurer l'importance de chaque caractéristique.

3. **Dataset**
   SCVIC-APT-2021
   - Gratuit et accessible publiquement
   - Contient 315 607 lignes, 84 variables, et 6 classes :
     - Exfiltration de données (*data exfiltration*),
     - Compromission initiale (*initial compromise*),
     - Mouvement latéral (*lateral movement*),
     - Trafic normal (*normal traffic*),
     - Reconnaissance (*reconnaissance*),
     - Pivot (*pivoting*).
   - Provenant d'un benchmark réaliste pour les APTs
4. **Programme**

   - **Pretraitements des données (data preprocessing)** :
     - Suppression des colonnes inutiles,
     - Nettoyage des valeurs infinies et manquantes (imputation à 0),
     - Encodage des variables catégorielles (label encoding),
     - Normalisation Min-Max pour homogénéiser les plages de données.
   - **Apprentissage et experimentation** :
     - Division 80/20 entre jeu d'entraînement/test.
     - Utilisation de plusieurs métriques de performance :
       - Précision (precision),
       - Rappel (recall),
       - Score F1 (F1 score),
       - MCC (Matthews Correlation Coefficient),
       - Cohen's Kappa,
       - Hamming Loss.
   - **IA explicable** :
     Visualisation des SHAP values :
     - Montre quelles caractéristiques ont le plus influencé la prédiction.
     - Par exemple : "Idle Max" pour XGBoost et "Fwd Header Length" pour CatBoost.

Pour élargir notre compréhension des différentes approches de détection des attaques APT, nous nous sommes également appuyés sur une revue de littérature approfondie [8], qui propose une classification complète des techniques existantes, en mettant l'accent sur l'intégration de l'intelligence artificielle et les futurs axes de recherche.

3.2.3 A Comprehensive Survey on Advanced Persistent Threat (APT) Detection Techniques.

Avec l'évolution continue des cybermenaces, les chercheurs ont entrepris de nombreuses initiatives pour développer des techniques de détection des APT toujours plus performantes. Cet article constitue une étude de synthèse approfondie sur les différentes approches de détection, en mettant en lumière à la fois les avancées récentes en intelligence artificielle et les limites des approches existantes. Il propose également une classification des méthodes [12], identifie les jeux de données utilisés et suggère des pistes pour les recherches futures dans le domaine de la cybersécurité intelligente.

1. **Objectif**
   Une etude exhaustive sur les techniques de detection des attaques de types APT, avec un focus sur :
   - L'analyse du processus d'attaque APT.
   - L'examen des techniques utilisées pour détecter et atténuer ces attaques.
   - L'identification des limitations des approches existantes.
   - La mise en avant des futurs axes de recherche en cybersécurité
2. **Methodes IA utilisées**
   - **Apprentissage automatique :** modèles de classification (SVM, Random Forest, etc.), détection d'anomalies.
   - **Apprentissage profond :** un réseau neuronal profond multi-couches (MLP-DNN) a été utilisé pour détecter les événements anormaux dans le réseau.
   - **Apprentissage fédéré avec méta-apprentissage (Federated Meta Learning) :** introduit pour détecter les APT à faible latence tout en préservant la confidentialité.
3. **Dataset**
   - **UNSW-NB15 :** utilisé pour entraîner et tester le modèle MLP-DNN dans des scénarios de forensique réseau liés aux APT.
   - **Jeux de données semi-synthétiques :** pour certains modèles de détection statistiques ou hybrides.
   - **Logs système et trafic réseau :** pour l'analyse comportementale et la corrélation d'événements.

   USNW-NB15 est entierement gratuit et couramment utilisé dans la recherche.

4. **Programme**

   - **Prétraitement des données :** réduction de dimensionnalité, extraction de caractéristiques (usage CPU, ports ouverts, etc).
   - **- Étapes de l'entraînement :**
     - Extraction des features (système et reseau) sur système sain.
     - Injection de code malveillant.
     - Extraction de nouvelles features.
     - Apprentissage supervisé pour entraîner un modèle de classification.

L'intégration de la Threat Intelligence avec les techniques de détection avancée des APT représente une approche complémentaire très prometteuse. L'article explore cette synergie en proposant une architecture modulaire combinant intelligence artificielle, analyse comportementale et renseignement sur les menaces pour renforcer la réponse aux incidents de cybersécurité.

3.2.4 Enhancing Cybersecurity Incident Response through Threat Intelligence and Advanced Persistent Threat Detection

Face à la sophistication croissante des attaques ciblées, les capacités de réponse rapide et contextuelle deviennent aussi cruciales que la détection elle-même [1]. Cet article propose un modèle d'intégration entre la Threat Intelligence (renseignement sur les menaces) et les technologies d'IA afin de créer un système cohérent de détection et d'intervention. En adoptant une architecture modulaire, cette approche permet de traiter les alertes dans un cadre dynamique et de prioriser les réponses selon la gravité et le contexte opérationnel.

1. **Objectif**
   renforcer la reponse aux incidents de cybersecurité en integrant :
   - La cyber threat intelligence (CTI) ou renseignement sur les menaces cybernetiques
   - La détection avancée des APT (Advanced Persistent Threats).
   - Une meilleure coordination entre détection des menaces et intervention en cas d'incident.
2. **Methodes d'IA**

   - **Apprentissage automatique supervisé** pour la classification d'événements anormaux.
   - **Réseaux bayésiens** pour l'évaluation des risques et des probabilités d'attaque.
   - **Analyse comportementale** basée sur des profils utilisateurs ou réseau.
   - **Systèmes d'alerte fondés sur l'IA** capables de générer des priorités d'action en fonction du contexte de l'incident.
   - **Fuzzy Logic** pour modéliser les incertitudes dans l'identification des APT

3. **Dataset**
    - Données issues des systèmes SIEM (Security Information and Event Management).
    - Flux de logs système, événements réseau, et alertes IDS.
    - Données tirées de menaces documentées (CTI).
    - Informations issues de sources OSINT (Open Source Intelligence).

    Les **datasets** évoqués sont en grande partie issue de système opérationnels **non publics,** bien que certains composants **OSINT** soient **partiellement accessible gratuitement**
4. **Programme**
    - Architecture modulaire pour intégrer :
        - Des modules de collecte CTI.
        - Un moteur de corrélation IA.
        - Des mécanismes d'automatisation de la réponse.
    - Pipeline automatisé ou chaine d'etapes de détection via :
        - Normalisation des données,
        - Extraction de caractéristiques (features),
        - Détection via des règles adaptatives et modèles entraînés.
    - Déploiement dans un environnement de simulation opérationnelle pour tester la robustesse des réponses.

3.2.5 Conclusion

À travers l'analyse de ces quatre contributions scientifiques, il apparaît que la détection des attaques persistantes avancées (APT) représente aujourd'hui un défi central dans le domaine de la cybersécurité. Les approches les plus prometteuses reposent sur l'utilisation conjointe de techniques d'intelligence artificielle (IA), d'analyses comportementales, de corrélation d'alertes[11] et, dans certains cas, de renseignements sur les menaces (Threat Intelligence).

Ces travaux ont mis en évidence plusieurs leviers techniques pour améliorer la détection d'APT : la réduction du bruit d'alerte, l'explicabilité des décisions prises par les modèles, l'utilisation de graphes pour relier les événements, et l'intégration de données hétérogènes issues de SIEM[5] ou d'OSINT[2]. La combinaison de méthodes supervisées et non supervisées, de boosting, ou encore d'approches fédérées, offre un éventail de stratégies adaptables selon le contexte et les objectifs de sécurité.

Cette revue de littérature fournit une base solide pour orienter notre propre démarche expérimentale. En nous inspirant des architectures et concepts proposés, nous allons dans la suite de ce chapitre mettre en œuvre une simulation réaliste d'une attaque APT dans un environnement virtuel, et évaluer l'efficacité d'un système de détection basé sur l'IA.

CHAPITRE 4

Chapitre 4 : Simulation d'une attaque APT et mise en œuvre d'un système de détection

4.1 Partie 01 : Scénario d'attaque simulée et son implémentation

4.1.1 Introduction

Dans cette section, nous présentons un scénario réaliste d'attaque de type APT (Advanced Persistent Threat) dans un réseau d'entreprise, puis nous détaillons son implémentation pratique. L'objectif est de démontrer comment un attaquant peut compromettre un système en exploitant des identifiants légitimes et exfiltrer des données sensibles sans déclencher d'alerte immédiate. Notre simulation repose sur trois étapes principales : compromission initiale, infiltration, et exfiltration des données, dans un environnement virtualisé comprenant un serveur d'entreprise, un poste employé, un routeur pfSense (avec DHCP) et une machine attaquante Kali Linux.

4.1.2 Étapes du scénario d'attaque

1. **Phishing et compromission initiale**
   Objectif : récupérer des accès valides pour se connecter au réseau interne.
   L'attaquant cible un employé via une campagne de phishing, réussissant à obtenir ses identifiants (nom d'utilisateur et mot de passe).
2. **Connexion au serveur en SSH**
   Objectif : obtenir un accès interactif au serveur.
   À l'aide des identifiants volés, l'attaquant se connecte au serveur Ubuntu en SSH. Cette connexion ressemble à une activité normale d'un employé, ce qui complique la détection.
3. **Exploration et collecte des données**
   Objectif : localiser les données à exfiltrer.

Après avoir établi la connexion, l'attaquant explore le serveur pour identifier des informations critiques. Dans notre simulation, il s'agit d'un dataset sensible stocké dans le server.

4. **Exfiltration des données**
   Objectif : transférer les données sans être détecté.
   L'attaquant copie le dataset vers sa machine locale (Kali) via SCP (Secure Copy).

4.1.3 Implémentation pratique et configuration

L'implémentation du scénario repose sur un environnement virtualisé avec VirtualBox, comprenant :

1. **pfSense**
   - **Version utilisée :** 2.7.2 Community Edition
   - **Role :** Utilisé comme passerelle réseau, serveur DHCP et proxy de journalisation.
   - **Fonctionnalités exploitées :**
     - Attribution dynamique des adresses IP via **DHCP**.
     - Filtrage du trafic.
     - Envoi des logs vers le serveur central (via Syslog).
2. **Serveur Ubuntu**
   - **Version utilisée :** 24.04.2 LTS
   - **Role :**
     Serveur central dans notre réseau d'entreprise fictif.
     - Contient le dataset sensible.
     - Fournit un accès SSH aux utilisateurs.
   - **Services instalés**
     - *openssh-server* pour la connexion distante.
     - *rsyslog* pour la journalisation système.
     - *cron* pour la surveillance périodique des connexions.
3. **Poste employé (Ubuntu)**
   - **Version utilisée :** 22.04.5 LTS
   - **Role :** Poste de travail de l'utilisateur légitime (employe), utilisé pour se connecter au serveur et consulter les données via SSH.
4. **Kali Linux**
   - **Version utilisée :**2025.2
   - **Role :** Machine de l'attaquant, utilisée pour simuler l'exploitation d'un compte compromis.
   - **Utilisation :**
     - Reconnaissance réseau (nmap, netdiscover).
     - Connexion SSH avec identifiants volés.

4.1.4 Configuration réseau avec pfSense

Activer le DHCP dans pfSense

- Accéder à l'interface web pfSense via l'IP par défaut (192.168.1.1).
- Services > DHCP Server > LAN.
- Activer le service DHCP et définir la plage d'adresses :

```
Range :  192.168.1.100 - 192.168.1.150
Subnet : 255.255.255.0
Getway : 192.168.1.1
```

- Sauvegarder les paramètres.
- Verification des adresses IP sur les machine avec ***ip a***

4.1.5 Configuration du serveur Ubuntu

1. **Mise à jour et installation de rsyslog**

```
sudo apt update && sudo apt upgrade
-y sudo apt install rsyslog net-tools -y
```

2. **Création d'un répertoire pour le dataset**

```
mkdir ~/entreprise_data
```

3. **Importation du dataset (depuis la machine physique via VirtualBox Shared Folder)**

```
cp /media/sf_dataset2/DSRL-APT-2023.csv ~/entreprise_data/
```

4. **Création du compte employé**

```
sudo adduser employe
```

Saisir un mot de passe (*zineddine*), ignorer les autres champs en appuyant sur Entrée.

5. **Attribution des droits d'accès au dataset**

```
sudo chown employe :employe ~/entreprise_data/DSRL-APT-2023.csv
sudo chmod 600 ~/entreprise_data/DSRL-APT-2023.csv
```

La premiere commande pour définir l'utilisateur et le groupe comme employe.
La seconde, pour donner la permission de lecture(4) et écriture (2) **rw-** au propriétaire (employé).
Seul **l'employé** qui peut lire et écrire ce fichier sécurisé.

4.1.6 Configuration du poste employé (Desktop)

- Installer OpenSSH :

```
sudo apt install openssh-client -y
```

- Connexion SSH au serveur :

```
ssh employe@192.168.1.100
```

192.168.1.100 : c'est l'adresse du server.
Saisir le mot de passe défini (*zineddine*).

4.1.7 Préparation de la machine Kali (attaquant)

- Verifier la connectivité réseau :

```
ping 192.168.1.100
nmap -p 22 192.168.1.100
```

4.1.8 Étapes de l'attaque simulée

1. **Phase de reconnaissance**
   - L'attaquant identifie le serveur accessible sur le port SSH (22).
   - Scanner le réseau avec nmap : la commande nmap permet de scanner un réseau et découvrir les hôtes actifs sur le réseau.

```
nmap -sP 192.168.1.0/24
```

Résultat

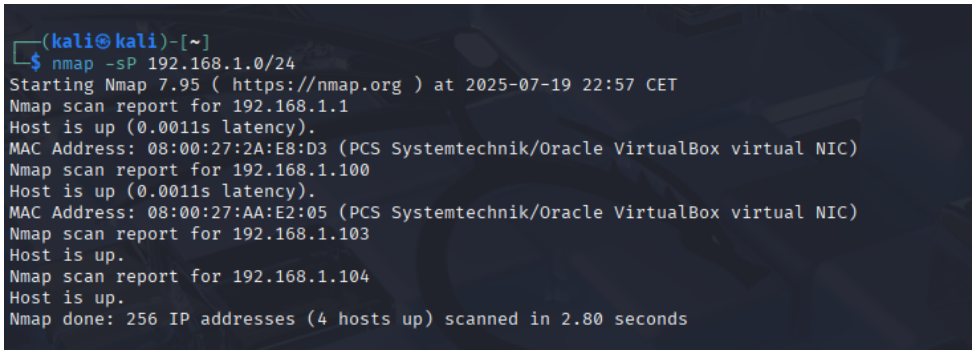

FIGURE 4.1 – Affichager les machines actives

- Scanner le réseau en ciblant port 22 SSH : permet d'affiche le statut du port 22 ssh

```
nmap -p 22 192.168.1.0/24
```

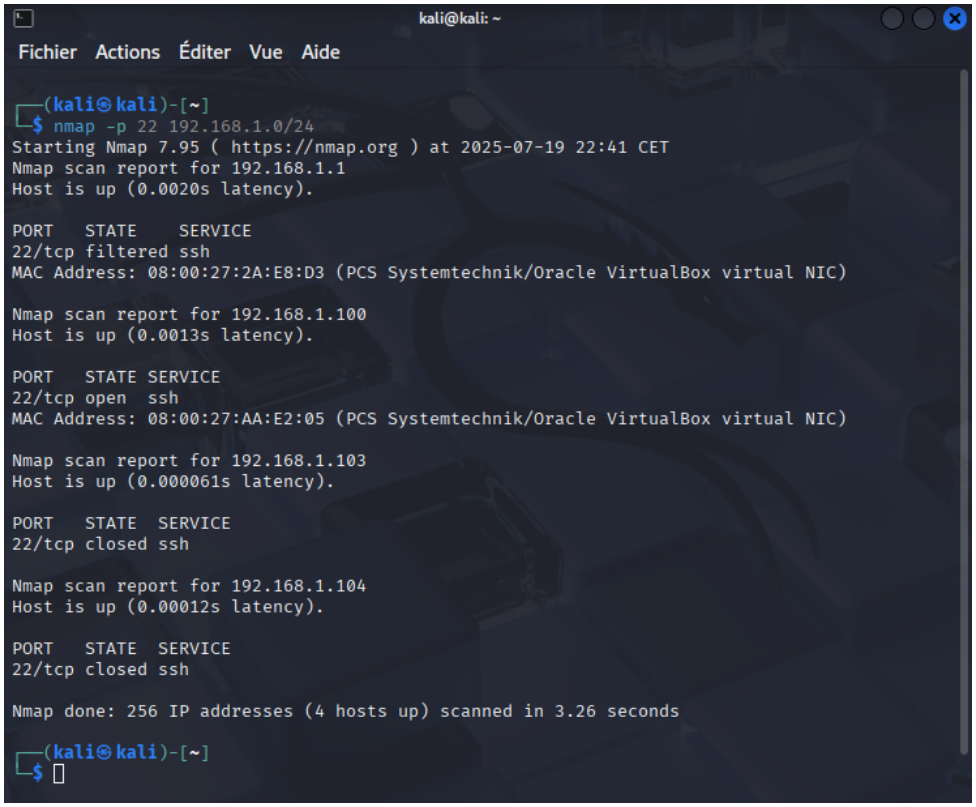

FIGURE 4.2 – afficher le statut du port 22 ssh dans les machines

- Vérifier la connectivité avec un ping :

```
ping 192.168.1.100
```

2. **Phase de compromission**
   - Suppose que l'attaquant obtient les identifiants : ***employe / zineddine***
3. **Connexion illégitime au serveur**
   - Se connecter au serveur en SSH avec les identifiants volés :

```
ssh employe@192.168.1.100
```

4. **Mouvement latéral et recherche de fichiers**
   - Ces trois lignes exécutent successivement la navigation vers le dossier entreprise_data, l'affichage détaillé de la liste des fichiers qu'il contient avec **ls -l**, puis la visualisation du contenu du fichier DSRL-APT-2023.csv à l'aide de **cat**.

```
cd ~/entreprise_data
ls -l
cat DSRL-APT-2023.csv
```

5. **Exfiltration des données via SCP**
   - cette commande permet de copier un fichier de server vers la machine kali

```
scp employe@192.168.1.100 :/home/employe/DSRL-APT-2023.csv /home/kali/
```

   Remarque : Pour executer cette commande on doit quitté l'accès à distance SSH.

6. **Deconnexion**
   - Pour la deconnexion

```
exit
```

4.1.9 Conclusion

La mise en place de ce scénario d'attaque APT nous a permis de reproduire, dans un environnement contrôlé, les principales étapes d'une compromission réelle dans un réseau d'entreprise. En utilisant un serveur Ubuntu contenant des données sensibles, un poste employé et une machine attaquante Kali Linux, nous avons simulé l'ensemble de la chaîne d'intrusion : reconnaissance, obtention d'identifiants via phishing, accès illégitime au serveur et exfiltration de données.

Cette simulation met en évidence la facilité avec laquelle un attaquant, une fois en possession d'identifiants valides, peut accéder aux ressources internes sans déclencher d'alertes apparentes. Elle démontre également les limites des mécanismes de sécurité traditionnels lorsqu'ils ne sont pas complétés par une analyse des journaux et un suivi des comportements réseau.

Dans la prochaine section, nous allons détailler les mécanismes de détection que nous avons mis en place, en nous basant sur l'analyse des logs SSH, la surveillance en temps réel des connexions et l'automatisation de la collecte d'événements, afin de repérer toute activité suspecte ou non autorisée.

4.2 Partie 02 : Detection de l'attaque

4.2.1 Introduction

Après avoir simulé une attaque APT réussie au sein de notre réseau d'entreprise, l'étape suivante consiste à mettre en place des mécanismes de détection afin d'identifier toute activité suspecte. L'objectif est de montrer comment un système d'information peut repérer un accès illégitime même si l'attaquant utilise des identifiants valides de l'employé compromis.

Pour cela, nous avons configuré un système de collecte et d'analyse des logs d'authentification (SSH) et des journaux système (syslog) sur le serveur. Nous avons également automatisé la surveillance grâce à des scripts et des outils natifs de Linux (comme grep, tail, cron) afin de générer des alertes et de conserver un historique complet des connexions réussies et échouées. La détection repose sur deux mécanismes complémentaires :

Surveillance en temps réel des journaux, pour repérer les tentatives de connexion suspectes.

Création d'un historique des authentifications, où chaque événement (connexion réussie, échec de connexion, IP source, horodatage) est stocké dans un fichier (hist_auth.txt) et consultable par l'administrateur.

Dans ce cadre, nous avons choisi rsyslog pour centraliser les logs et avons configuré des filtres spécifiques afin de détecter uniquement les événements liés à SSH.

4.2.2 Objectif de la détection

L'objectif de la détection est de surveiller et identifier toute activité anormale dans le réseau, en particulier lorsqu'un compte utilisateur légitime, comme celui de l'employé, est compromis. Face aux attaques persistantes avancées (APT), qui exploitent des identifiants valides et se fondent dans le trafic normal, il est crucial de mettre en place un système capable de suivre en temps réel les connexions (via les logs SSH et syslog), d'enregistrer les tentatives d'accès réussies ou échouées, et d'alerter rapidement en cas de comportement suspect.

Ce mécanisme de détection complète les mesures de prévention en fournissant une traçabilité précise (utilisateur, adresse IP, date et heure) et permet aux administrateurs de réagir avant que l'attaque ne progresse.

4.2.3 Configuration de la détection

La configuration de la détection repose sur la mise en place d'un système centralisé de collecte et d'analyse des logs dans le serveur Ubuntu (192.168.1.102). L'objectif est de tracer toutes les connexions et tentatives d'accès des différents hôtes du réseau (poste employé, Kali, etc.) afin de détecter d'éventuelles intrusions. Cette configuration s'effectue en plusieurs étapes :

1. **activation de *rsyslog* sur le serveur Ubuntu**
   - Commande utilisée pour activer syslog (démarre automatiquement)

```
sudo systemctl enable rsyslog
sudo systemctl start rsyslog
```

- Vérification que le port 514 (utilisé par rsyslog) est bien ouvert

```
sudo netstat -tulnp | grep 514
```

```
serverubuntu@serverubuntu:~/folder$ sudo netstat -tulnp | grep 514
[sudo] password for serverubuntu:
tcp        0      0 0.0.0.0:514             0.0.0.0:*               LISTEN
778/rsyslogd
tcp6       0      0 :::514                  :::*                    LISTEN
778/rsyslogd
udp        0      0 0.0.0.0:514             0.0.0.0:*
778/rsyslogd
udp6       0      0 :::514                  :::*
778/rsyslogd
serverubuntu@serverubuntu:~/folder$
```

FIGURE 4.3 – Le service rsyslog est en train d'écouter sur le port 514

2. **Configuration de rsyslog pour accepter les logs distants**
   - On tape cette commande pour acceder au fichier de configuration

```
nano /etc/rsyslog.conf
```

   - Le fichier de configuration /etc/rsyslog.conf a été modifié pour activer les modules d'écoute TCP et UDP

```
# provides UDP syslog reception
module(load="imudp")
input(type="imudp" port="514")

# provides TCP syslog reception
module(load="imtcp")
input(type="imtcp" port="514")

# provides kernel logging support and enable non-kernel klog messages
module(load="imklog" permitnonkernelfacility="on")
```

FIGURE 4.4 – Activer la réception de logs via les protocoles UDP et TCP

   - Apres la modification de *rsyslog* on redémarre avec :

```
sudo systemctl restart rsyslog
```

3. **Surveillance des logs l'authentification**

   La sauvegarde des événements du système dans un fichier dédié, offre plusieurs avantages. Elle permet d'assurer une traçabilité complète des tentatives de connexion, en conservant un historique clair et consultable à tout moment. Cette méthode facilite la détection rapide d'activités suspectes, comme les tentatives d'accès non autorisées, et constitue un support précieux pour les audits et les enquêtes post-incident. De plus, en centralisant les informations importantes (ex. connexions réussies ou échouées), elle simplifie l'analyse des logs et permet d'automatiser certaines tâches de surveillance, garantissant ainsi une meilleure sécurité et un suivi régulier du système.

- **Surveillance avec l'enregisterement d'un fichier**
  - On tape cette commande :

```
nano ~/extract_auth.sh
```

- on écrit ça dans fichier extract_auth.sh
  Cet algorithme surveille les connexions SSH réussies et échouées du système et

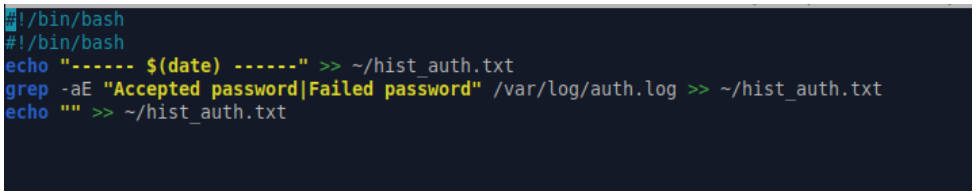

FIGURE 4.5 – Le contenu du fichier extract_auth.sh

enregistre ces informations dans un fichier d'historique (hist_auth.txt) avec un horodatage.
- On verifie que le programme est éxecutable avec :

```
chmod +x ~/extract_auth.sh
```

-On ouvre l'éditeur pour configurer les tâches planifiées avec :

```
crontab -e
```

-Puis, on ajoute la ligne suivante à la fin :

```
* * * * * /home/serverubuntu/extract_auth.sh
```

Donc, le programme éxecute chaque minute
-Pour afficher le contenu du fichier (logs) hist_auth.txt, on tape :

```
cat hist_auth.txt
```

On aura ce résultat :
La capture ci-dessus montre l'historique des tentatives de connexion SSH collectées par le script extract_auth.sh et enregistrées dans le fichier hist_auth.txt. Chaque bloc commence par un horodatage (par exemple, dim. 20 juil. 2025 01 :56 :01 CET) indiquant le moment où le script a été exécuté. On observe des connexions réussies (Accepted password for employe) et des tentatives échouées (Failed password for employe) provenant de l'adresse IP 192.168.1.104. Ces informations permettent de suivre les activités d'authentification en temps réel et de détecter les tentatives suspectes.

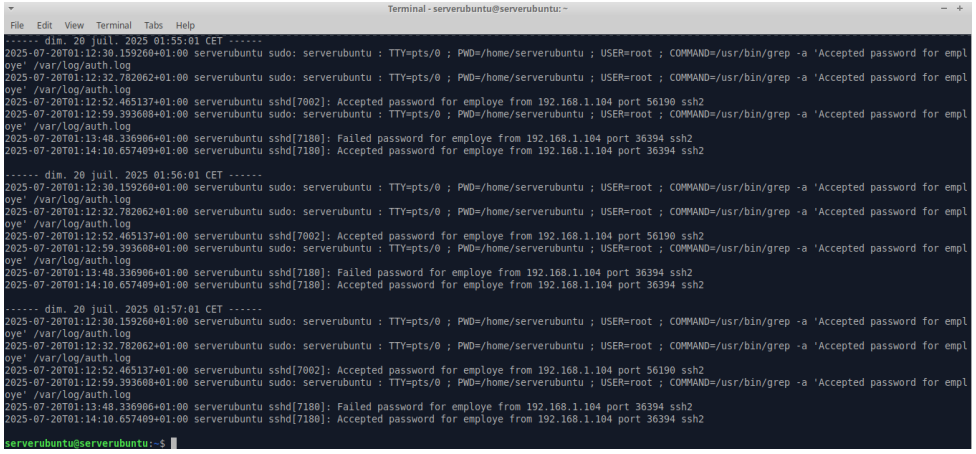

FIGURE 4.6 – Capture d'écrans des logs sauvegarder dans le fichier hist_auth.txt

- **Surveillance en temps réel**
  - Pour surveiller en temps réel on utilise :

```
sudo tail -f /var/log/auth.log
```

tail -f : permet de suivre les nouvelles lignes ajoutées en continu.
Ce qu'on va voir dans terminal du server

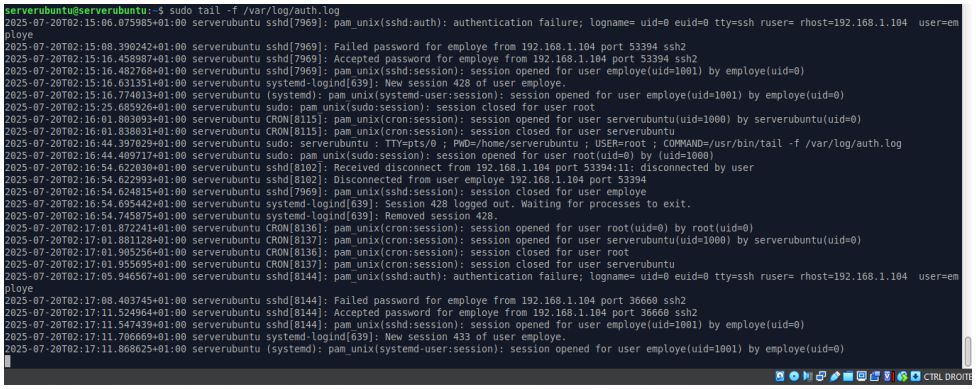

FIGURE 4.7 – Capture d'écrans des logs affiché dans le server en temps réel

Cette capture montre une surveillance en temps réel du fichier */var/log/auth.log* grâce à la commande sudo *tail -f*. On peut observer plusieurs tentatives d'authentification SSH de l'utilisateur employe (ou l'attaquant) depuis l'adresse IP *192.168.1.104*. Certaines connexions ont échoué (*authentication failure et Failed password*), suivies de connexions réussies (*Accepted password*). Le log indique également l'ouverture et la fermeture de sessions (*session opened / session closed*) ainsi que des exécutions automatiques via cron. Ces informations permettent d'analyser les activités des utilisateurs et de détecter d'éventuelles intrusions.

Filtrer les tentatives d'authentification :
- Pour voir seulement les tentatives de connexion SSH (réussites et échecs) :

```
sudo tail -f /var/log/auth.log | grep "sshd"
```

- Pour uniquement les connexions réussies :

```
sudo tail -f /var/log/auth.log | grep "Accepted password"
```

- Pour uniquement les échecs :

```
sudo tail -f /var/log/auth.log | grep "Failed password"
```

4.2.4 Intégration de l'algorithme XGBoost pour la détection des anomalies

1. **Langage de programmation utilisé : Python**
   Le langage Python a été utilisé pour l'ensemble du traitement, du prétraitement des données jusqu'à l'entraînement du modèle.
   Ses avantages sont nombreux :
   - Simplicité de syntaxe rendant le code plus lisible.
   - Large écosystème de bibliothèques spécialisées (pandas, scikit-learn, xgboost, joblib...).
   - Support communautaire important et documentation riche.

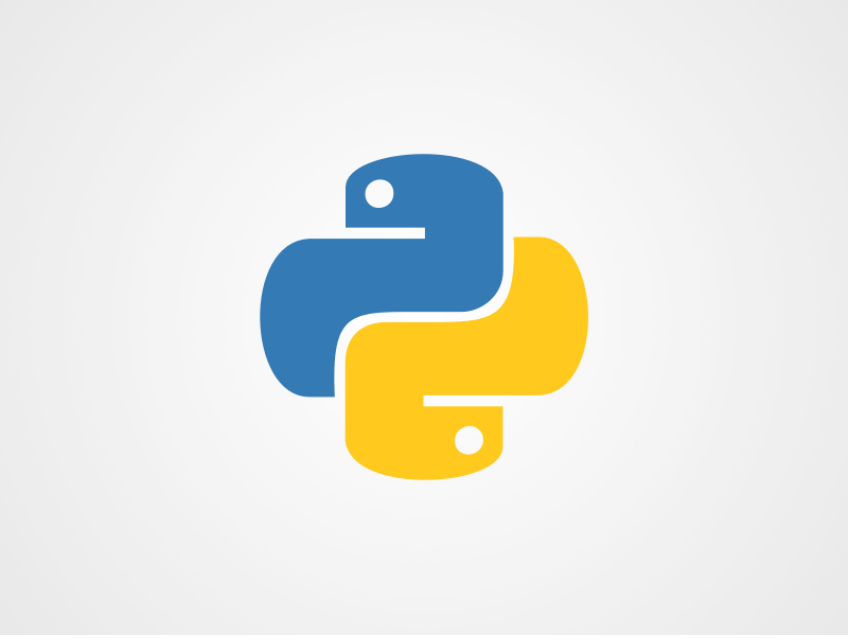

FIGURE 4.8 – Logo du language Python

2. **Présentation et justification du choix de XGBoost**
   XGBoost (Extreme Gradient Boosting) est un algorithme d'apprentissage supervisé basé sur la technique de Gradient Boosting appliquée aux arbres de décision.
   Ses points forts sont :
   - **Haute performance** en termes de précision et de rapidité d'entraînement.
   - **Robustesse** face aux valeurs manquantes et aux données bruitées.
   - **Optimisation mémoire** et parallélisation native.
   - **Adaptabilité** à divers types de problèmes, y compris la classification binaire, la classification multi-classes et la régression.

   Justification du choix :
   Par rapport à d'autres modèles comme SVM ou les réseaux neuronaux, **XGBoost** offre :
   - Un excellent compromis entre performance et vitesse.
   - Une capacité d'explication plus claire grâce à l'importance des variables.
   - Une meilleure gestion des données tabulaires hétérogènes, fréquentes dans les réseaux.

3. **Nature binaire de la classification**
   Dans notre projet, la tâche consiste à prédire si un trafic réseau est normal ou malveillant.
   - **Classe 0 :** Comportement normal (Benign).
   - **Classe 1 :** Anomalie ou attaque APT détectée (Reconnaissance).

   Ce type de problème est appelé **classification binaire**.
   XGBoost gère efficacement ce format en utilisant des fonctions de perte adaptées.

   Pour obtenir une détection complète d'une attaque APT, il faudrait entraîner des modèles séparés pour :
   - Benign vs Reconnaissance.
   - Benign vs Exfiltration.
   - Benign vs Mouvement latéral.
   - Benign vs establish foothold (établir une base).

4. **Prétraitement et filtrage des données**
   Avant l'entraînement, un filtrage a été réalisé pour ne conserver que les données pertinentes (classes *benign et reconnaissance*) à partir du jeu de données **DSRL-APT-2023** [21].

   Le dataset **DSRL-APT-2023** contient un ensemble riche de caractéristiques (features) décrivant les flux réseau capturés. Ces attributs peuvent être regroupés en plusieurs catégories :

   a) **Informations d'identification du flux :**
   - *Flow ID, Src IP, Src Port, Dst IP, Dst Port, Protocol, Timestamp* $\rightarrow$ identifient de manière unique un flux réseau et précisent son contexte (source, destination, protocole, moment de capture).

b) **Durée et taille du flux :**
- *Flow Duration, Total Fwd Packets, Total Bwd Packets, Total Length of Fwd/Bwd Packets* → mesurent la durée et le volume des paquets transmis dans chaque direction.

c) **Statistiques sur les paquets envoyés et reçus :**
- *Fwd Packet Length Max/Min/Mean/Std, Bwd Packet Length Max/Min/Mean/Std* → décrivent la distribution des tailles de paquets en avant (Fwd) et en arrière (Bwd).

d) **Caractéristiques temporelles (IAT – Inter Arrival Time) :**
- *Flow IAT Mean/Std/Max/Min, Fwd IAT Mean, Bwd IAT Mean* → indiquent la régularité ou la variabilité du trafic, souvent révélatrice d'un comportement automatisé ou malveillant.

e) **Indicateurs liés aux drapeaux TCP :**
- *FIN Flag Count, SYN Flag Count, RST Flag Count, PSH Flag Count, ACK Flag Count, URG Flag Count* → comptent le nombre de paquets contenant certains flags TCP, utiles pour détecter des anomalies ou attaques réseau.

f) **Mesures de débit et ratios :**
- *Flow Bytes/s, Flow Packets/s, Down/Up Ratio* → évaluent l'intensité et la symétrie du flux.

g) **Caractéristiques de sous-flux (Subflow) :**
- *Subflow Fwd Packets, Subflow Bwd Packets* → fractionnent un flux en sous-séquences pour observer la dynamique d'échange.

h) **Fenêtre et activité TCP :**
- *Fwd Init Win Bytes, Bwd Init Win Bytes* → valeurs initiales de fenêtre TCP.
- *Active Mean/Std/Max/Min, Idle Mean/Std/Max/Min* → périodes actives et inactives d'un flux, pertinentes pour repérer des communications furtives.

i) **Colonnes de classification :**
- *Activity* → type d'activité réseau (ex. reconnaissance, mouvement latéral, exfiltration). - *Stage* → étape de l'attaque APT associée (par exemple : reconnaissance, exploitation, persistance).

```
 GNU nano 7.2                                                    alg filter.py
import pandas as pd

# === Étape 1 : Charger le fichier CSV ===
fichier_entree = "DSRL-APT-2023.csv"
df = pd.read_csv(fichier_entree)

# === Étape 2 : Nettoyer la colonne 'Stage' ===
# Convertir en string, retirer les espaces, et passer en minuscule
df['Stage'] = df['Stage'].astype(str).str.strip().str.lower()

# === Étape 3 : Filtrer les lignes avec 'benign' ou 'reconnaissance' ===
df_filtré = df[df['Stage'].isin(['benign', 'reconnaissance'])]

# === Étape 4 : Sauvegarder dans un nouveau fichier ===
fichier_sortie = "filtrer.csv"
df_filtré.to_csv(fichier_sortie, index=False)

# === Étape 5 : Afficher le résumé ===
print(f" {len(df_filtré)} lignes filtrées ont été enregistrées dans '{fichier_sortie}'.")
print("\n Détail des classes conservées :")
print(df_filtré['Stage'].value_counts())
```

FIGURE 4.9 – Algorithme qui filtrer les donnée

5. **Séparation en ensembles d'entraînement et de test** Pour entraîner et évaluer le modèle, les données filtrées ont été séparées en deux ensembles :
   - **80%** pour l'entraînement
   - **20%** pour le test

```
 GNU nano 7.2                                                    amg separation.py
import pandas as pd
from sklearn.model_selection import train_test_split

# Charger les données filtrées
df = pd.read_csv("filtrer.csv")

# Séparer en 80% entraînement et 20% test
df_train, df_test = train_test_split(df, test_size=0.2, random_state=42, shuffle=True)

# Sauvegarder les deux sous-fichiers
df_train.to_csv("fichier80.csv", index=False)
df_test.to_csv("fichier20.csv", index=False)

# Afficher un résumé
print(f" Données divisées avec succès :")
print(f"- Entraînement (80%) : {len(df_train)} lignes → fichier80.csv")
print(f"- Test (20%)         : {len(df_test)} lignes → fichier20.csv")
```

FIGURE 4.10 – Algorithme qui sépare les données

6. **Entraînement et validation du modèle XGBoost**
   L'entraînement du modèle XGBoost inclut :
   - Encodage des étiquettes (LabelEncoder).
   - Normalisation des données (StandardScaler).
   - Construction et entraînement du modèle.
   - Évaluation via un rapport de classification.

4.2.5 Résultats expérimentaux et interprétation

Après l'entraînement du modèle XGBoost sur l'ensemble d'apprentissage (80 % du dataset) et sa validation sur l'ensemble de test (20 %), les résultats obtenus sont présentés dans le rapport

```
 GNU nano 7.2                                                    alg_entrai.py
import pandas as pd
import xgboost as xgb
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.metrics import classification_report, accuracy_score, log_loss
import joblib

# === 1. Charger les données d'entraînement ===
df_train = pd.read_csv("fichier80.csv")
df_train = df_train.dropna()

# === 2. Supprimer colonnes inutiles ===
colonnes_a_supprimer = ["Flow ID", "Src IP", "Dst IP", "Timestamp"]
df_train = df_train.drop(columns=[col for col in colonnes_a_supprimer if col in df_train.columns])

# === 3. Nettoyage de la colonne cible ===
df_train['Stage'] = df_train['Stage'].astype(str).str.strip().str.lower()

# === 4. Encoder la cible ===
label_encoder = LabelEncoder()
df_train['Label'] = label_encoder.fit_transform(df_train['Stage'])

# === 5. Séparation features / target ===
X_train = df_train.drop(columns=["Label", "Stage"]).select_dtypes(include=["float64", "int64"])
y_train = df_train["Label"]

# === 6. Normalisation ===
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
```

FIGURE 4.11 – Algorithme de l'entraînement : Prétraitement et préparation des données

```
 GNU nano 7.2                                                    alg_entrai.py

# === 7. Entraînement du modèle ===
model = xgb.XGBClassifier(use_label_encoder=False, eval_metric="logloss")
model.fit(X_train_scaled, y_train)

# === 8. Sauvegarde ===
joblib.dump(model, "xgboost_model.pkl")
joblib.dump(scaler, "scaler.pkl")
joblib.dump(label_encoder, "label_encoder.pkl")

print(" Modèle entraîné et sauvegardé.")

# === 9. TEST SUR LE FICHIER 20% ===
df_test = pd.read_csv("fichier20.csv")
df_test = df_test.dropna()
df_test = df_test.drop(columns=[col for col in colonnes_a_supprimer if col in df_test.columns])
df_test['Stage'] = df_test['Stage'].astype(str).str.strip().str.lower()

# Utiliser le label_encoder déjà entraîné pour transformer les labels
df_test['Label'] = label_encoder.transform(df_test['Stage'])

X_test = df_test.drop(columns=["Label", "Stage"]).select_dtypes(include=["float64", "int64"])
y_test = df_test["Label"]

# Appliquer le scaler déjà entraîné
X_test_scaled = scaler.transform(X_test)
```

FIGURE 4.12 – Algorithme de l'entraînement : Entraînement et sauvegarde du modèle

```
# Prédiction
y_pred = model.predict(X_test_scaled)
y_pred_proba = model.predict_proba(X_test_scaled)

# Évaluation
print("\n Rapport sur ensemble de test :\n")
print(classification_report(y_test, y_pred, target_names=label_encoder.classes_))
print(f"Accuracy : {accuracy_score(y_test, y_pred):.4f}")
print(f"Log-loss : {log_loss(y_test, y_pred_proba):.4f}")
```

FIGURE 4.13 – Algorithme de l'entraînement : Évaluation et test du modèle

suivant :

```
serverubuntu@serverubuntu:~/folder/folder2$ python3 alg_entrai.py
 Modèle entraîné et sauvegardé.

 Rapport sur ensemble de test :

              precision    recall  f1-score   support

       benign       1.00      1.00      1.00      2010
reconnaissance       1.00      1.00      1.00      2864

     accuracy                           1.00      4874
    macro avg       1.00      1.00      1.00      4874
 weighted avg       1.00      1.00      1.00      4874

Accuracy : 1.0000
Log-loss : 0.0003
serverubuntu@serverubuntu:~/folder/folder2$
```

FIGURE 4.14 – Rapport de classification sur l'ensemble de test

Deux métriques globales viennent compléter ce rapport :

- **Précision (Precision) = 1.0000 :** la précision indique la proportion des prédictions positives qui sont correctes. Une valeur de 1.00 signifie que le modèle n'a généré aucun faux positif : chaque instance prédite comme attaque ou bénigne correspondait bien à la réalité.
- **Rappel (Recall) = 1.0000 :** le rappel mesure la capacité du modèle à identifier correctement toutes les occurrences positives. Avec une valeur de 1.00, cela veut dire qu'aucune attaque ni aucun trafic bénin n'a été oublié (aucun faux négatif).
- **Score F1 (F1-score) = 1.0000 :** le F1-score combine la précision et le rappel en une seule mesure harmonisée. Une valeur de 1.00 traduit un équilibre parfait entre précision et rappel, confirmant la robustesse du modèle.
- **Exactitude (Accuracy) = 1.0000 :** l'exactitude représente la proportion totale de prédictions correctes sur l'ensemble des 4874 échantillons testés. Un score parfait de 1.00 indique que toutes les instances ont été classées correctement.
- **Moyenne macro (Macro avg) = 1.0000 :** cette moyenne donne le même poids à chaque classe, indépendamment de leur fréquence. Une valeur de 1.00 montre que le modèle est tout aussi performant pour chaque catégorie.
- **Moyenne pondérée (Weighted avg) = 1.0000 :** cette moyenne prend en compte le déséquilibre entre les classes (2010 échantillons bénins et 2864 d'attaque). Une valeur de 1.00 confirme que le modèle reste excellent même en présence d'un déséquilibre de données.
- **Log-loss binaire = 0.0003 :** le log-loss évalue la qualité probabiliste des prédictions. Plus la valeur est proche de zéro, plus le modèle attribue une probabilité élevée à la bonne classe. Ici, 0.0003 indique une très forte confiance et une grande fiabilité du modèle.

Analyse des résultats

Ces résultats indiquent que le modèle a atteint un taux de reconnaissance parfait (100 %) sur l'ensemble de test, aussi bien pour la classe *benign* que pour la classe *reconnaissance*. Les métriques de précision, rappel et F1-score étant toutes égales à 1.00, cela signifie qu'aucune erreur

de classification n'a été détectée.
L'interprétation combinée de l'accuracy et du log-loss montre que le modèle ne se contente pas seulement de prédire correctement la classe finale, mais qu'il le fait avec une très forte probabilité associée, ce qui renforce la fiabilité de ses décisions.

Discussion
Bien que les résultats obtenus soient extrêmement satisfaisants, il convient d'adopter une interprétation prudente :

- Le score parfait peut être le signe que le dataset est relativement simple à séparer ou que les classes présentes dans ce sous-ensemble de test sont fortement discriminantes.
- Il existe un risque de surapprentissage (overfitting), le modèle ayant peut-être mémorisé les données d'entraînement. Pour évaluer sa robustesse, il serait nécessaire de tester le modèle sur d'autres phases d'attaque APT (comme *lateral movement, establish foothold, ou exfiltration*) ou sur un dataset totalement différent.
- Néanmoins, dans le cadre de ce projet, ces résultats démontrent que l'intégration de l'algorithme XGBoost est efficace et qu'il constitue une solution prometteuse pour la détection des anomalies en environnement réseau simulé.

4.2.6 Conclusion

La mise en œuvre d'un mécanisme de surveillance, combinée à l'intégration de l'algorithme XGBoost, a permis de simuler et de détecter efficacement une attaque APT au sein de l'environnement expérimental. Les scripts de collecte et de filtrage des logs ont assuré une préparation rigoureuse des données, tandis que l'entraînement supervisé a montré des performances remarquables, avec une exactitude de 100 % et un log-loss quasi nul. Ces résultats confirment la pertinence de l'approche, tout en soulignant la nécessité d'évaluations complémentaires sur d'autres phases d'attaque et jeux de données afin de garantir la robustesse et la généralisation du modèle.

Conclusion

À travers ce travail, nous avons étudié la problématique de la détection des attaques persistantes avancées (APT) dans les réseaux d'entreprise, en mettant en évidence leurs caractéristiques, leur complexité ainsi que les limites des approches classiques de détection. Nous avons montré que les systèmes traditionnels, bien qu'efficaces face aux menaces connues, demeurent insuffisants pour contrer des attaques sophistiquées capables d'exploiter des identifiants légitimes et de se fondre dans le trafic normal.

Pour répondre à cette problématique, nous avons conçu et implémenté une simulation réaliste d'une attaque APT, puis développé un système de détection basé sur l'algorithme XGBoost. Les résultats obtenus confirment que l'intelligence artificielle, et plus particulièrement le machine learning, constitue une solution prometteuse pour améliorer la précision, réduire les faux positifs et renforcer la capacité de détection en temps réel des comportements anormaux.

Cependant, ce travail présente certaines limites, notamment la dépendance aux jeux de données utilisés pour l'entraînement, la complexité de la mise en œuvre de certains modèles et la nécessité d'une adaptation continue face à l'évolution rapide des techniques d'attaque. Ces limites ouvrent la voie à plusieurs perspectives intéressantes, telles que l'enrichissement des jeux de données par des scénarios plus variés, l'exploration d'approches hybrides combinant différentes méthodes d'IA, l'intégration du modèle dans des solutions SIEM pour une meilleure corrélation des alertes, ou encore le recours à l'IA explicable afin de fournir des explications claires aux analystes de sécurité.

En définitive, ce mémoire met en évidence l'importance de développer des solutions de cybersécurité intelligentes et adaptatives, capables non seulement de détecter les menaces les plus avancées, mais aussi d'évoluer avec elles. Il constitue une contribution à la compréhension et à l'amélioration des mécanismes de détection des APT, tout en ouvrant des perspectives prometteuses pour la recherche et l'application pratique dans le domaine de l'administration et de la sécurité des réseaux.

En conclusion, ce mémoire s'inscrit dans une double dynamique : d'une part, analyser et comprendre les défis que posent les attaques persistantes avancées (APT) aux réseaux d'entreprise ; d'autre part, expérimenter l'apport de l'intelligence artificielle dans la détection de ces menaces. La simulation d'un scénario d'APT, combinée à l'application de techniques de

machine learning, constitue une contribution concrète à l'étude des mécanismes de défense modernes.

À plus long terme, ce travail pourra être enrichi par plusieurs perspectives :

- explorer des modèles d'apprentissage plus avancés tels que le deep learning, les réseaux neuronaux récurrents ou les graph neural networks ;
- utiliser des jeux de données plus riches et représentatifs pour améliorer la robustesse des modèles ;
- intégrer les mécanismes de détection proposés dans des environnements de production réels, afin d'évaluer leur efficacité dans des contextes opérationnels.

Ces pistes ouvrent la voie vers le développement de systèmes de sécurité intelligents, capables de s'adapter en permanence et de répondre de manière proactive aux menaces émergentes.

Bibliographie

[1] Ali, G., Shah, S., and ElAffendi, M. (2025). Enhancing cybersecurity incident response : Ai-driven optimization for strengthened advanced persistent threat detection. *Results in Engineering*, 25 :104078.

[2] Bazzell, M. (2023). *Open Source Intelligence Techniques : Resources for Searching and Analyzing Online Information*. Independently published, 10th edition edition.

[3] Bhuyan, M. H., Bhattacharyya, D. K., and Kalita, J. K. (2013). Network anomaly detection : methods, systems and tools. *Ieee communications surveys & tutorials*, 16(1) :303–336.

[4] Canonical Ltd. (2023). *Ubuntu Server Guide*. Disponible sur https://ubuntu.com/server/docs. Consulté le 30 juin 2025.

[5] Chuvakin, A., Schmidt, K., and Phillips, C. (2012). *Security Information and Event Management (SIEM) Implementation*. McGraw-Hill Education.

[6] Dhanabal, L. and Shantharajah, S. (2015). A study on nsl-kdd dataset for intrusion detection system based on classification algorithms. *International journal of advanced research in computer and communication engineering*, 4(6) :446–452.

[7] Fornaro, C. (2002). Network intrusion detection : An analyst's handbook. *COMPUTER JOURNAL*, 45(4) :473–473.

[8] Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., and Vázquez, E. (2009). Anomaly-based network intrusion detection : Techniques, systems and challenges. *computers & security*, 28(1-2) :18–28.

[9] Hasan, M. M., Islam, M. U., and Uddin, J. (2023). Advanced persistent threat identification with boosting and explainable ai. *SN Computer Science*, 4(3) :271.

[10] Kaspersky, E. (2013). The threats of tomorrow. Conférence RSA Europe, Amsterdam. Consulté le 24 juillet 2025.

[11] Khan, N., Alsaqour, R., and et al. (2024). A survey of machine learning for big data processing. *Journal of Big Data*, 11(1) :1–45.

[12] Krishnapriya, S. and Singh, S. (2024). A comprehensive survey on advanced persistent threat (apt) detection techniques. *Computers, Materials & Continua*, 80(2).

[13] Kshetri, N. (2013). *Cybercrime and cybersecurity in the global south*. Springer.

[14] Mandiant (2013). Apt1 : Exposing one of china's cyber espionage units. Technical report, Mandiant Corporation. Consulté le 30 juin 2025.

[15] McWhorter, D. (2013). Apt1 : Exposing one of china's cyber espionage units. *Mandiant. com*, 18.

[16] Mishra, N. and Mishra, S. (2024). A review of machine learning-based intrusion detection system. *EAI Endorsed Transactions on Internet of Things*, 10.

[17] Netgate (2023). *pfSense Documentation*. Disponible sur https://docs.netgate.com/pfsense/en/latest/. Consulté le 18 mai 2025.

[18] Offensive Security (2023). *Kali Linux Documentation*. Disponible sur https://www.kali.org/docs/. Consulté le 30 juin 2025.

[19] Oracle Corporation (2023). *Oracle VM VirtualBox User Manual*. Version 7.0. Disponible sur https://www.virtualbox.org. Consulté le 14 avril 2025.

[20] Scarfone, K. and Mell, P. (2007). Guide to intrusion detection and prevention systems (idps). *NIST special publication*, 800(2007) :94.

[21] Shadabfar, H. and Dehghan, M. (2023). Dsrl-apt-2023 : Dataset for advanced persistent threat detection. https://github.com/shadab75/DSRL-APT-2023. Consulté le 15 aout 2025.

[22] Shankar, V. and Muralidhar, N. (2025). Securing hardware with ai : Intrusion detection, threat mitigation, and trust assurance. *International Journal of Computer Sciences and Engineering*, 13(4) :92–98.

[23] Soliman, H. M., Sovilj, D., Salmon, G., Rao, M., and Mayya, N. (2023). Rank : Ai-assisted end-to-end architecture for detecting persistent attacks in enterprise networks. *IEEE Transactions on Dependable and Secure Computing*, 21(4) :3834–3850.

[24] Sommer, R. and Paxson, V. (2010). Outside the closed world : On using machine learning for network intrusion detection. In *2010 IEEE symposium on security and privacy*, pages 305–316. IEEE.

[25] Stallings, W. (2014). *Network Security Essentials : Applications and Standards*. Pearson, Boston, USA, 5th edition.

Résumé

Ce mémoire porte sur la détection des attaques persistantes avancées (APT) dans un réseau d'entreprise à l'aide de l'intelligence artificielle. Les attaques APT constituent une menace majeure car elles sont discrètes, évolutives et difficiles à détecter avec les solutions de sécurité classiques. Pour répondre à cette problématique, nous avons conçu une architecture de simulation représentant un réseau d'entreprise vulnérable et y avons reproduit un scénario complet d'attaque APT. Ce scénario a permis de collecter un jeu de données réel (DSRL-APT-2023), qui constitue la base de nos expérimentations.

Nous avons ensuite appliqué plusieurs techniques d'apprentissage supervisé, dont le modèle XGBoost, afin de classifier les événements et détecter les activités malveillantes. Les résultats obtenus mettent en évidence une précision élevée dans l'identification des phases d'attaque, ainsi qu'une bonne robustesse face aux faux positifs. Ces performances démontrent l'efficacité des approches supervisées pour renforcer la sécurité des réseaux d'entreprise.

Enfin, nous discutons des perspectives futures, telles que l'amélioration de la qualité des données, l'enrichissement du scénario de simulation avec davantage de phases d'attaque, et l'optimisation des modèles supervisés afin de faciliter une détection plus rapide et plus fiable.
Mots clés : attaques persistantes avancées (APT), sécurité des réseaux, détection d'intrusions, XGBoost, analyse du trafic réseau.

Abstract

This thesis focuses on the detection of Advanced Persistent Threats (APTs) in an enterprise network using Artificial Intelligence. APTs represent a major threat since they are stealthy, evolving, and difficult to detect with traditional security solutions. To address this issue, we designed a simulation architecture that reproduces a vulnerable enterprise network and implemented a complete APT attack scenario. This scenario enabled the collection of a real dataset (DSRL-APT-2023), which served as the foundation for our experiments.

We then applied several supervised learning techniques, including the XGBoost model, to classify events and detect malicious activities. The results demonstrate a high accuracy in identifying attack phases as well as strong robustness against false positives. These findings highlight the effectiveness of supervised approaches in strengthening the security of enterprise networks.

Finally, we discuss future perspectives, such as improving data quality, enriching the simulation scenario with additional attack phases, and optimizing supervised models to achieve faster and more reliable detection.
Keywords : Advanced Persistent Threats (APT), network security, intrusion detection, XGBoost, network traffic analysis.