الجمـــهورية الـــــجزائرية الديمقـــــراطية الشعـــــــبية
République Algérienne Démocratique et Populaire
وزارة التعليـــــــــم العالـــــــــي و البحث العلمـــــــي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université 20 Août 1955 – Skikda		جـامعة 20 أوت 1955- سكيكدة
Faculté des Sciences		كـــــلية العلـــــوم
Departement d'informatique		قسم الإعـــــلام الآلـــي

Mémoire De fin d'étude en vue de l'obtention du Diplôme de Master en Informatique

Option : *Systèmes Informatique (SI)*

Thème :

Détection De Malwares Par Les Techniques De Machines Learning

Réalisé par :

Mosbah Asma

Bouchoukh Amina

Encadré par :

Dr. Chikh Ramdane

Année Universitaire 2022-2023

Remerciement

*Au terme de ce travail, on tient à remercier **ALLAH** le tout puissant de nous avoir donné la foi et de nous avoir permis d'en arriver là.*

*Nous voudrions également témoigner notre gratitude à notre encadreur **Dr. Chikh Ramdane** pour sa présence, sa patience et son soutien et ses conseils qui nous ont été précieux afin de mener notre travail à bon port.*

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail Et de l'enrichir par leurs propositions. Les mots ne suffisent pas pour remercier nos parents et toutes nos familles pour leur soutien et pour avoir toujours cru en nous. Ce travail est aussi le fruit de leur patience. Merci. Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Dédicaces

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,

A mes chers frères, « Alla » & « Imad », pour leur appui et leur encouragement,

A toute ma famille pour leur soutien tout au long de temps,

A mes chers amis : Rourou, Nardjes, Aya, Rania, Khadidja et Racha, Anfal, pour leur appui,

A ma cher binôme Amina, avec qui j'ai passé du bon temps

Toutes les étudiantes de la promotion 2022/2023.

Merci à touts

Asma

Dédicaces

Je dédie ce travail

*A mon père *Amar* qui ma toujours soutenue dans moments difficiles et pour ses sacrifices et ses encouragements dieu me le garde.*

*A ma mère *Dalila* qui a sacrifié sa vie pour mon bien être lui dédie ce modeste mémoire en souhaitant de tout mon cur dieu me la garde.*

A ma cher frère "Amir"

A mes chères sœurs "Ikram" "Amira"

*Mon amie *Asma*, qui a été mon binôme durant toute notre cursus.*

A toutes mes amies "Louiza, Maissa, Rabiaa, Meryem, Ghada "

A tous mes camarades de la promotion 2023 : Master Systèmes Informatiques.

A tous ceux qui me sont chers et que j'ai involontairement oublié A tous mes professeurs.

Merci !

Amina

ملخص :

الهجمات السيبرانية على أنظمة الكمبيوتر لا تزال تتزايد يوما بعد يوم. غالبًا ما يستخدم المهاجمون البرامج الضارة لتنفيذ أنشطتهم الخبيثة. يعد فهم البرامج الضارة التي يستخدمها المهاجمون وتحليلها أمرًا أساسيًا لاكتشاف طرق تسللها ومن تم القضاء عليها. لا تزال تقنيات التقليدية للكشف عن البرامج الضارة غير قادرة على اكتشاف البرامج الضارة الجديدة. للتعامل مع هذه المشكلة، يُقترح استخدام تقنيات التعلم الآلي للكشف عن البرامج الضارة والحميدة وتصنيفها.

الكلمات المفتاحية: *الأمن السيبراني، الكشف، التعلم الالي، البرامج الضارة، الملف التنفيذي*

Résumé :

Les attaques contre les systèmes informatiques sont encore en constante augmentation. Les attaquants utilisent souvent des logiciels malveillants pour mener à bien leurs activités malveillantes. Comprendre et analyser les logiciels malveillants utilisés par les attaquants est essentiel pour détecter ces actions et les éliminer. Les techniques classiques de détection de malwares sont encore incapables de découvrir les nouvelles malwares. Pour traiter ce problème, il est proposé d'utiliser des techniques d'apprentissage automatique pour la détection et la classification des logiciels malveillants et bénins.

Mots Clés : *Sécurité Informatique, Détection, Machine Learning, Malware, Fichier* PE

Abstract:

Attacks against computer systems are still increasing. Attackers often use malware to carry out their malicious activities. Understanding and analyzing the malware used by attackers is key to detect and eliminate these actions. Conventional malware detection techniques are still unable to discover new malware. To deal with this problem, it is proposed to use machine learning techniques for the detection and classification of malicious and benign software.

Keywords: *computer security, detection, machine learning, malware, PE file*

Table des matières

Table des figures

Liste des tableaux

Introduction générale

Contexte

Internet et les systèmes d'information sont devenus une technologie incontournable dans notre vie privée comme dans notre vie professionnelle. Ils facilitent la vie et économise les efforts, l'argent et du temps. Malgré ces avantages, ils présentent des risques et des menaces pour les utilisateurs d'ordinateurs. Puisqu'ils peuvent être comme un moyen pour lancer les programmes et des fichiers malveillants qui pouvant endommager et détruire les systèmes informatiques et volent les données d'individus et d'organismes. Pour sécurité nos système et réduire les risques d'attaquants, plusieurs approches et techniques ont été proposées et réalisées par les spécialistes. Notamment celles à base de l'apprentissage automatique pour leurs avantages de détecter les nouvelles malwares.

Problématique

En général, l'industrie des logiciels malveillants est devenue un marché lucratif pour investir et mettre en œuvre des technologies de pointe pour échapper à la détection traditionnelle, tandis que les fournisseurs d'anti-malware dépensent des milliers, voire des millions de dollars pour arrêter la propagation des logiciels malveillants, car cela ne fera pas qu'entraîner des pertes financières, mais aussi émotionnelles.

Objectif

Nous essayons donc dans ce projet de concevoir un système à base d'analyse statique de fichiers binaires en utilisant l'apprentissage numériques, le but ultime est de détecter les programmes malveillants avant qu'ils ne soient exécutés en mémoire, notamment en détectant les nouveaux contenus suspects.

Contenu et organisation

Chapitre01 :

Ce chapitre présente la sécurité informatique, y compris les attaques et les logiciels malveillants, en mentionnant leurs types, les sources de d'infection et les approches de détection, suivi d'une étude sur le format PE.

Chapitre02 :

Dans ce chapitre, nous avons fourni une brève définition de l'apprentissage automatique et de ses types, ainsi que certaines des techniques et les algorithmes utilisés pour détecter les logiciels malveillants.

Chapitre03 :

Ce chapitre est consacré à l'analyse des logiciels malveillants ainsi que de l'architecture du système proposé, en fournissant une comparaison entre les études précédentes et une description de l'ensemble de données utilisé dans cette analyse.

Chapitre04 :

Dans le dernier chapitre, une description des outils et des programmes utilisés dans cette recherche et quelques captures d'écran du système final sont fournies.

Chapitre 01 : la sécurité informatique Et la détection des malwares

Introduction

La gestion des réseaux d'entreprise et le nombre croissant d'utilisateurs se connectant à l'Internet font actuellement face à de sérieux défis en matière de sécurité informatique. Les idées fondamentales de la sécurité des systèmes informatiques, les nombreux types d'attaques qui peuvent avoir lieu, et les outils et méthodes qui pourraient être utilisés pour assurer la sécurité sont tous couverts dans ce chapitre d'introduction. Le transfert de données privées et la nécessité d'en protéger le secret sont devenus des éléments essentiels du développement des réseaux informatiques.

La sécurité informatique

Définition

Le concept de sécurité informatique couvre tous les moyens, technologies et méthodes pour garantir que seul le personnel ou d'autres systèmes autorisés participent au système et disposent de données sensibles.

Avec le rôle des systèmes informatiques ouverts à l'extérieur et le soutien stratégique actuellement détenu par ce dernier, la sécurité est un thème majeur, qui est rarement pris en compte dans la juste valeur. [1]

Figure 1:sécurité informatique [65]

Les types de la sécurité informatique

Sécurité du réseau :

Pour empêcher les utilisateurs malveillants ou non autorisés d'accéder à votre réseau et d'utiliser la sécurité du réseau. Cela favorise l'utilité, la fiabilité et l'intégrité du réseau. Afin d'empêcher les pirates d'accéder à des données sur le réseau, cette protection est requise. De plus, cela les empêche de modifier l'accès au réseau ou à l'utilisation des utilisateurs. [2]

Sécurité Internet :

La sécurité Internet comprend des sauvegardes de la sécurité réseau pour les applications Web et les données fournies et reçues dans les navigateurs. Ces mesures de protection visent à accorder une attention particulière au mauvais trafic et aux logiciels malveillants dans le trafic Internet constant. Les pare-feu, les logiciels anti-malicious et les logiciels anti-espy peuvent assurer cette sécurité. [2]

Sécurité de point d'extrémité :

La sécurité au niveau de l'appareil est assurée via la sécurité des terminaux. Les téléphones portables, les tablettes, les ordinateurs portables et les ordinateurs de bureau font partie des gadgets qui peuvent bénéficier de la sécurité des terminaux. Vos appareils ne peuvent pas accéder à des réseaux hostiles qui pourraient mettre en danger votre entreprise grâce à la sécurité des terminaux. [2]

Figure2 :les types de sécurité informatique [66]

Sécurité du nuage :

Les utilisateurs peuvent se connecter directement à Internet sans être sécurisés par l'architecture de sécurité classique grâce à la migration des applications, des données et des identités vers le cloud. La sécurité du réseau augmente la sécurité de l'utilisation des réseaux publics et des applications SaaS (logiciel en tant que service). Pour protéger le nuage, il est possible d'utiliser un système unifié de gestion des menaces, une passerelle Internet sécurisée ou un portail de sécurité de l'information. [2]

Sécurité des conteneurs :

Le processus continu de défense des conteneurs contre les cybermenaces comprend le pipeline de conteneurs, l'infrastructure de déploiement et la chaîne d'approvisionnement. [2]

Sécurité des applications :

Afin de rendre l'application aussi sûre que possible et d'éviter les attaques, la sécurité des applications est utilisée pour coder spécifiquement les applications pendant la création. Une autre couche protectrice consiste à vérifier le code source de l'application pour rechercher des défauts logiciels possibles. [2]

Sécurité de l'internet of things (lot) :

La protection, la surveillance et la neutralisation des attaques visant l'Internet des objets (IoT) et son réseau d'appareils IoT connectés qui collectent, stockent et partagent des données en ligne sont couvertes par la sécurité IoT, un sous-ensemble de la cybersécurité. [2]

Les approches de la sécurité informatique

Approche préventive

Un outil important dans cette catégorie est le contrôle d'accès. Le contrôle d'accès permet de limiter et de gérer l'accès aux ressources critiques. Il est supposé qu'un attaquant n'est pas légitimement autorisé à utiliser l'objet cible et se voit donc refuser l'accès à la ressource. Comme l'accès est une condition préalable à une attaque, toute interférence possible est empêchée.
La forme la plus courante de contrôle d'accès utilisée consiste à vérifier l'identité

d'un utilisateur, ceci est fait par un processus d'authentification qui nécessite généralement un nom et un mot de passe. Un pare-feu est un autre contrôle d'accès important du système sur la couche réseau L'idée d'un pare-feu est basée sur la séparation d'un réseau interne d'ordinateurs de confiance sous un seul contrôle administratif d'un réseau extérieur hostile potentiel. Donc, le pare-feu empêche les attaques de l'extérieur contre les machines du réseau interne en refusant les tentatives de connexion provenant de parties non autorisées situées à l'extérieur. **[3]**

Approche détective

Toutes les approches et procédures sont combinées dans une approche de détective pour identifier rapidement les accès non autorisés aux systèmes informatiques et l'altération des données protégées. Cette méthode repose généralement sur la simulation du comportement typique du système avant de considérer tout écart comme une action ou une attaque indésirable. Le mécanisme de base de cette méthode est le système de détection d'intrusion (IDS).

Enfin, le principal avantage de cette stratégie est sa capacité à identifier de nouvelles incursions sans avoir besoin de connaissances préalables sur les attaques. [3]

Les mécanismes de sécurité

Les systèmes de détection d'intrusion (IDS)

Un ensemble de composants matériels et/ou logiciels connus sous le nom de système de détection d'intrusion (IDS) est utilisé pour identifier et évaluer toute tentative d'infiltration intentionnelle. [4]

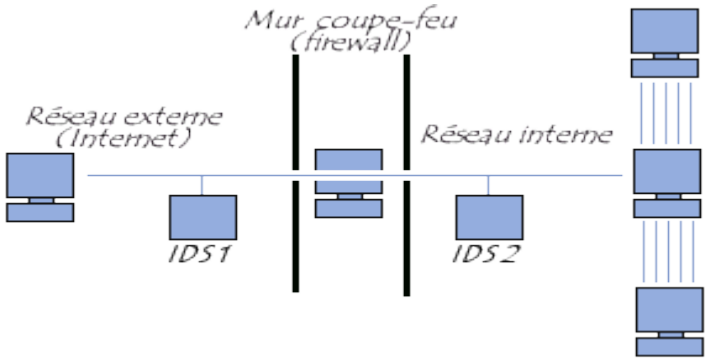


Figure03 : Protection par IDS [67]

- **Systèmes de détection d'intrusion de type "hôte" (HIDS)**

Les techniques de détection d'hôte de type HIDS existent avant le NIDS. Un HIDS est un agent logiciel qui, dans la plupart des cas, est placé sur le système à protéger et qui analyse en temps réel les flux relatifs à cette machine. [4]

- **Systèmes de détection d'intrusion de type "réseaux" (NIDS)**

Le but des NIDS est de regarder les paquets transitant dans le réseau pour déterminer si une attaque a lieu. Ainsi, un NIDS ne pourra protéger que les ordinateurs se trouvant sur le même réseau que lui. Un désavantage des NIDS est que tout paquet chiré ne pourra pas être compris par les NIDS, qui doivent lire le contenu des paquets. Ce désavantage peut disparaître en utilisant des HIDS qui vont obtenir les informations après leur déchirement sur la machine. Néanmoins, les NIDS sont faciles à installer pour la détection d'intrusion d'un ensemble de machines. [4]

- **Système de détection d'intrusion hybride (Hybrid IDS)**

Dans ce type d'IDS, les sources d'information viennent à la fois du réseau et des machines sur le réseau. Ainsi, la complexité du système grandit mais les avantages des NIDS et des HIDS sont combinés. De plus, ils permettent une meilleure détection d'attaques distribuée. [4]

LES FIREWALLS (PARE-FEU)

Un pare-feu est un dispositif de sécurité qui surveille le trafic réseau entrant et sortant et autorise ou bloque les paquets de données en fonction d'un ensemble de règles de sécurité. Son objectif est de créer une barrière entre votre réseau interne et le trafic provenant de sources externes (telles que Internet) afin de bloquer le trafic malveillant tel que les virus et les pirates. [5]

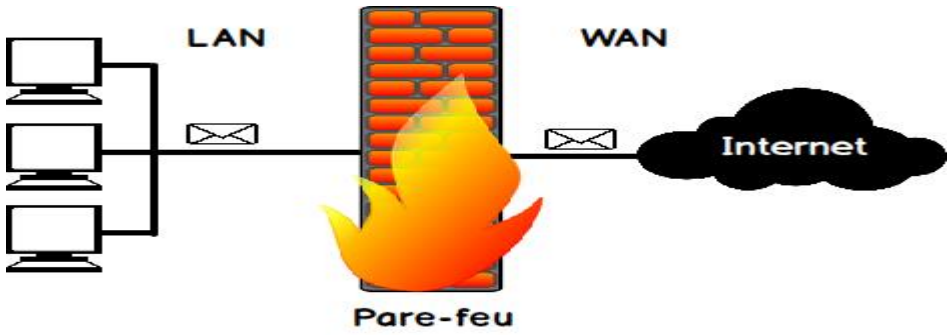


Figure04 : les Pares Faux [68]

Le cryptage

Le cryptage, ou chiffrement, sert à des fins de sécurité et de confidentialité des données. Bien avant l'arrivée de l'informatique et des réseaux, le chiffrement existait déjà, sous d'autres formes. [6]

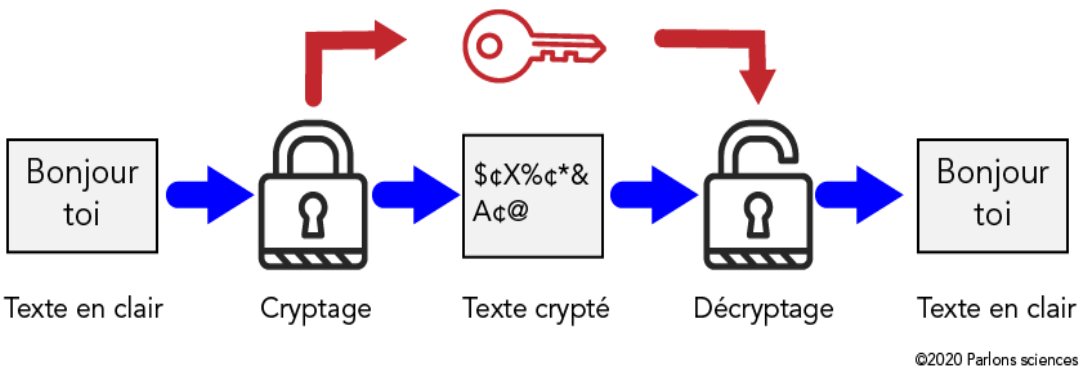


Figure05 : Protection par le cryptage[69]

Le cryptage va prendre des données et les passer dans une sorte de moulinette qui va les rendre incompréhensibles. Cette moulinette est appelée « algorithme de cryptage » ou « algorithme de chiffrement ». Cela va permettre d'envoyer des données, et que seuls ceux qui sont en possession de la clé de chiffrement pourront lire. [7]

Anti-virus

Les antivirus sont des logiciels conçus pour identifier, neutraliser et éliminer des logiciels malveillants. Ceux-ci peuvent se baser sur l'exploitation de failles de sécurité, mais il peut également s'agir de programmes modifiant ou supprimant des fichiers, que ce soit des documents de l'utilisateur de l'ordinateur

infecté, ou des fichiers nécessaires au bon fonctionnement de l'ordinateur. La plupart des antivirus sont basés sur l'analyse de signature des fichiers, la base des signatures doit donc être très régulièrement mise à jour sur le site de l'éditeur (des procédures automatiques sont généralement possibles).

Deux modes de protection :

- Généralisation de l'antivirus sur toutes les machines, il faut absolument prévoir une mise à jour automatique de tous les postes via le réseau.
- Mise en place d'un antivirus sur les points d'entrée/sortie de données du réseau après avoir parfaitement identifiés tous ces points. La rigueur de tout le personnel pour les procédures doit être acquis. [4]

Privacy

Toutes les données peuvent être sensibles, de la rentabilité de l'entreprise aux chiffres de vente ou aux feuilles de route des produits. Les données personnelles relatives à toute personne connue ou identifiable sont parmi les plus sensibles. Les informations personnelles identifiables (PII) peuvent être presque n'importe quoi. Les PII ne sont généralement pas aussi évidents qu'un nom ou un numéro de sécurité sociale. Parfois, il s'agit d'une autre identification, telle qu'une adresse IP ou des informations sur les cookies. Les données personnelles sont des informations qui peuvent être utilisées pour identifier un individu sur la base d'un champ de données ou d'un enregistrement. [58]

Les attaques

Définition

Une « attaque » est l'exploitation d'une faille d'un système informatique à des fins non connues par l'exploitant du système et généralement préjudiciables.

Sur internet des attaques ont lieu en permanence, à raison de plusieurs attaques par minute sur chaque machine connectée. Ces attaques sont pour la plupart lancées automatiquement à partir de machines infectées, à l'insu de leur propriétaire. [8]

Figure 06 : l'attaque [79]

Les types d'attaques

Dans l'environnement numérique de la connexion actuelle, les criminels de réseau utilisent des outils complexes pour lancer des cyberattaques contre l'entreprise. Les type d'attaques de réseau actuel est le suivant :

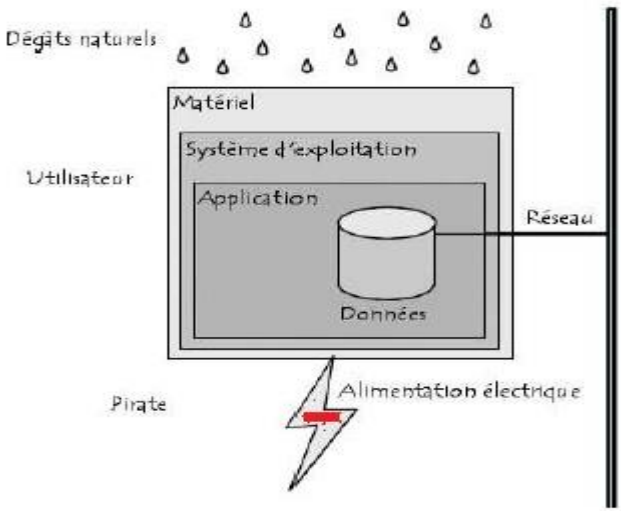


Figure07 :les types attaque [79]

Les chevaux de Troie de porte dérobée :

Un cheval de Troie crée une porte dérobée vulnérable dans le système la victime, permettant au pirate d'en obtenir le contrôle à distance et presque total. Fréquemment utilisé pour relier un groupe d'ordinateurs de victimes dans un réseau de bots ou réseau de zombies. [9]

Attaque par déni de service (DoS) :

Les attaques DoS, ou « Denial of Service », ont pour but de perturber un système en l'inondant d'une foule de requêtes. Les attaques DDoS adoptent le même principe, mais sont lancées depuis plusieurs machines simultanément. Ce type de cyberattaque ne permet pas aux pirates de voler des données ou de contrôler un système ; il sert uniquement à paralyser un site internet ou un système. [10]

Tunnellisation des systèmes de noms de domaines (DNS):

Les cybercriminels utilisent la tunnellisation DNS, un protocole transactionnel, pour échanger des données d'applications, comme l'extraction de données en mode silencieux ou l'établissement d'un canal de communication avec un serveur inconnu, à l'image de l'échange de commande et de contrôle (C&C) à titre d'exemple. [9]

Logiciels malveillants :

Il s'agit d'un logiciel malveillant qui peut rendre les systèmes infectés inopérants. La plupart des variantes de logiciels malveillants détruisent les données en supprimant ou en effaçant les fichiers essentiels au fonctionnement du système d'exploitation. [9]

Hameçonnage :

L'hameçonnage, ou Phishing en anglais, n'est pas une attaque en soi, mais plutôt un moyen pour préparer une attaque future du type piratage de compte, intrusion voire rançongiciel.

Il s'agit de se faire passer dans un email pour une source fiable et digne de confiance afin de tromper les victimes et obtenir ainsi des informations confidentielles, comme des codes d'accès, ou les inciter à agir : clic vers un site malveillant, ouverture d'une pièce jointe, installation d'un logiciel, saisie d'un formulaire, … [11]

Exploit zero-day :

Les attaques zero-day tirent avantage des faiblesses inconnues du matériel et du logiciel. Ces vulnérabilités peuvent exister pendant des jours, des mois ou des années avant que les développeurs ne prennent connaissance de ces failles. [9]

Injection SQL :

Les attaques par injection de langage de requête structurée (langage SQL) intègrent un code malveillant dans des applications vulnérables, produisant des résultats finaux de requêtes de bases de données et exécutant des commandes ou des actions similaires que l'utilisateur n'a pas demandées. [9]

Rançongiciel :

Le rançongiciel est un logiciel malveillant sophistiqué qui tire avantage des faiblesses du système, en utilisant un chiffrement renforcé pour retenir les données ou la fonctionnalité du système en otage. Les cybercriminels se servent du rançongiciel pour exiger un paiement en échange de la libération du système. [9]

Attaque de script intersite (XSS) :

Les attaques cross-site scripting insèrent un code malveillant dans un site web ou un script d'application légitime dans le but d'obtenir les informations d'un utilisateur, souvent à l'aide de ressources web tierces. Les pirates se servent fréquemment de JavaScript pour les attaques XSS, mais Microsoft VCScript, ActiveX et Adobe Flash peuvent également être utilisés. [9]

Attaque par mot de passe :

Étant donné que les mots de passe sont le mécanisme le plus couramment utilisé pour authentifier les utilisateurs de systèmes informatiques, l'obtention de mots de passe est une méthode d'attaque courante et efficace. [9]

L'attaque anniversaire :

Une attaque d'anniversaire Il est un type d'attaque cryptographique utilisé pour cryptanalyse des algorithmes de chiffrement ; Il est ainsi nommé parce qu'il utilise les principes mathématiques de paradoxe d'anniversaire en la théorie des probabilités. [12]

Attaque par téléchargement :

Les opérateurs Coreflood ont utilisé des sites Web piégés pour infecter les ordinateurs des utilisateurs via une technique appelée téléchargement furtif (Drive by download). [13]

Les malwares

Définition :

Tout logiciel malveillant destiné à accéder secrètement à un appareil est appelé malware. Les logiciels malveillants se présentent sous une grande variété de formes, chacune employant une stratégie unique pour atteindre ses objectifs malveillants. Mais il y a deux choses que chaque type de virus partage... [14]

Figure08: les malwares [70]

Les types des malwares :

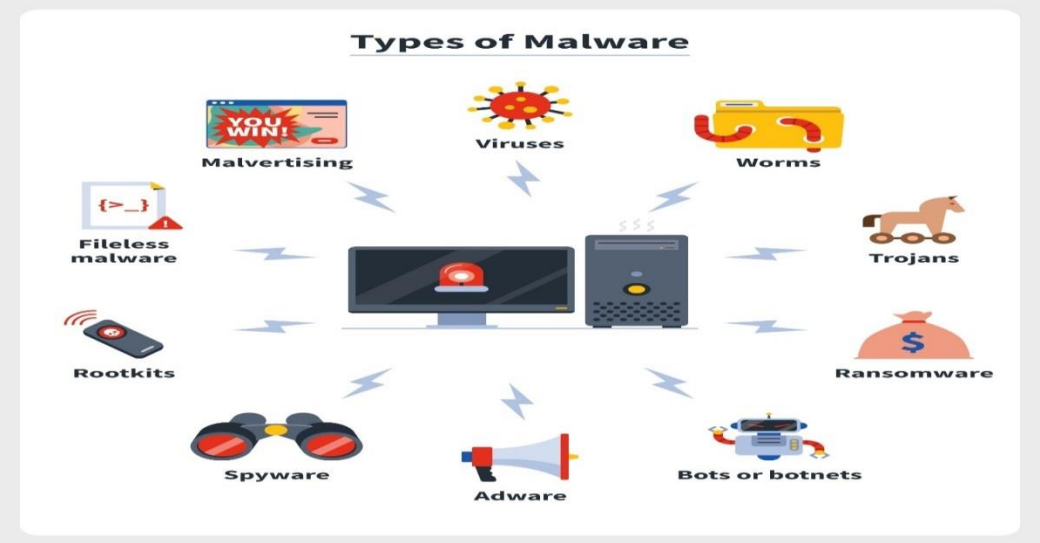


Figure09 : les types des malwares [70]

- **Les chevaux de Troie**

Les chevaux de Troie sont des programmes nuisibles qui se cachent dans d'autres applications. Il se faufile sur les PC en se cachant dans des applications réputées comme les économiseurs d'écran. Le système d'exploitation est ensuite modifié avec un code qui permet aux pirates d'accéder à l'ordinateur compromis. En règle générale, les chevaux de Troie ne se propagent pas d'eux-mêmes. Ils sont transmis par des vers, des virus ou des logiciels téléchargés. [15]

- **Les adwares**

Le but de l'adware est de faire gagner de l'argent à ses créateurs en faisant en sorte que les victimes voient des publicités indésirables. Les jeux gratuits ou les barres d'outils du navigateur sont des exemples de types de logiciels publicitaires courants. Ils recueillent des informations sensibles sur leurs victimes et les utilisent pour adapter les publicités qu'ils affichent. Même si la majorité des logiciels publicitaires peuvent être installés légalement, cela peut toujours être très gênant. [16]

- **Les spywares**

Les logiciels espions, sont exactement ce que leur nom l'indique : des logiciels malveillants conçus pour surveiller l'activité de votre ordinateur à votre insu. Parce que les logiciels espions peuvent collecter tant d'informations sur vous, de vos e-mails aux numéros de carte de crédit, mots de passe, etc. Ils constituent une menace énorme pour la confidentialité et la sécurité. [17]

- **Les Ransomware**

Le malware de rançonnage, ou ransomware, est un type de malware qui empêche les utilisateurs d'accéder à leur système ou à leurs fichiers personnels et exige le paiement d'une rançon en échange du rétablissement de l'accès. Les premières versions de ransomwares ont été développées à la fin des années 1980. À cette époque, la rançon devait être envoyée par courrier postal. Aujourd'hui, les auteurs de ransomwares demandent à être payés en cryptomonnaies ou par carte de crédit. [18]

- **Les virus**

Un virus est un morceau de code qui s'insère dans une application et s'exécute au démarrage de l'application. Une fois à l'intérieur d'un réseau, un virus peut être utilisé pour voler des données une fois que cela se produit, le virus se réplique et se propage dans votre système. Confidentielles, lancer une attaque DDoS ou exécuter une attaque de ransomware. Généralement propagés via des sites Web infectés, des partages de fichiers ou le téléchargement de pièces jointes à des e-mails, les virus restent inactifs jusqu'à ce qu'un fichier ou un programme hôte infecté soit activé [19] , exemple : I love you , CryptoLocker [40].

- **Les macrovirus**

Peut-être le type de virus informatique le plus courant, un virus de macro s'attache aux fichiers créés dans des programmes qui prennent en charge les macros et les séquences de commandes à une seule touche. Ces virus se trouvent le plus souvent dans les documents Microsoft Word et les feuilles de calcul Excel. [20]

- **Boot Sector**

C'est un type de virus qui infecte le secteur de démarrage des disquettes ou

le Master Boot Record (MBR) des disques durs. Le secteur Boot comprend tous les fichiers nécessaires au démarrage du système d'exploitation de l'ordinateur. Le virus écrase le programme existant ou se copie sur une autre partie du disque**. [21]**

Virus infectant les fichiers

Virus qui s'attache à un programme exécutable. Il est également appelé virus parasite qui infecte généralement les fichiers avec les extensions .exe ou .com. Certains infecteurs de fichiers peuvent écraser les fichiers hôtes et d'autres peuvent endommager le formatage de votre disque dur. [22]

- **Les vers**

Un ver est un type de logiciel malveillant qui se propage dans le monde entier via des connexions au réseau à la recherche d'une cible. Les vers sont dangereux car ils exploitent des vulnérabilités informatiques connues (par exemple, des problèmes avec les systèmes de sécurité informatique) pour s'adapter aux machines. Une fois qu'ils y sont, il est très difficile de les empêcher de rôder vers leurs cibles. [23]

- **Vers internet**

Ce sont les vers qui ciblent les sites internet qui ne sont pas très bien protégés. Ils exploitent une vulnérabilité sur le site web pour se déposer sur le serveur. [24]

- **Vers courriel**

Des vers courriel possèdent généralement une double extension (.mp4.exe ou .avi.exe par exemple). Ils se propagent par mail, à travers une pièce jointe. Il est aussi probable que le corps même du message contienne un lien. [24]

- **Vers de messagerie instantanée**

Les vers de messagerie instantanée se propagent en envoyant des liens vers la liste de contacts des applications de messagerie instantanée telles que Messenger, WhatsApp, Skype, etc. [25]

- **Vers partage de fichier**

Les vers de partage de fichiers s'intègrent et se déguisent sous la forme de

fichiers multimédias bénins, qu'un utilisateur inconscient téléchargera ensuite sur son appareil, permettant ainsi au ver de s'infecter. Une fois que le ver s'est propagé sur l'appareil, il peut alors voler des informations privées que l'instigateur malveillant peut vendre à d'autres attaquants ou utiliser directement à son avantage. [26]

- **Vers IRC**

Comme les vers de messagerie instantanée et de courrier électronique, les vers IRC se propagent via des pièces jointes et des liens malveillants et accèdent ensuite à la liste de contacts d'un utilisateur infecté pour se propager davantage. [26]

- **Le backdoor**

Une backdoor, ou porte dérobée, est un logiciel qui permet à un attaquant de prendre le contrôle d'un ordinateur à distance. Les fonctionnalités de base d'une backdoor permettent de pouvoir copier des fichiers sur la machine infectée et d'en lancer l'exécution. [27]

- **Spam**

L'envoi de spam est une méthode visant à submerger Internet avec des copies d'un même message. La plupart des spams sont des offres publicitaires envoyées aux utilisateurs sous la forme d'un message électronique non sollicité. Les spams sont aussi appelés courrier indésirable. [28]

- **Rootkits**

Sont un type de malware qui permet aux cybercriminels de contrôler à distance les appareils des victimes, souvent à leur insu. Étant donné que les rootkits sont conçus pour rester cachés, ils peuvent détourner ou subvertir les logiciels de sécurité. [29]

PE format

Historique :

Microsoft migra vers le format PE avec l'introduction de Windows NT 3.1 OS. Toutes les versions suivantes de Windows, incluant Windows 95/98/ME, supportent ce format. Auparavant ils étaient au format NE — New Executable File Format, « New » faisant référence à CP/M, et aux fichiers .com .

La création du format PE a été induite par le fait que Microsoft souhaitait créer une structure de fichier portable, de sorte qu'elle puisse s'adapter aux différents systèmes de Windows NT, car il faut savoir que Windows NT était au départ capable de supporter d'autres architectures que le x86 d'Intel, Power PC et Motorola 68000 en faisaient partie. L'idée fût donc de créer une structure commune à ces architectures. [30]

Définition

Le format de fichier PE est une structure de données qui contient les informations nécessaires au chargeur du système d'exploitation Windows pour gérer le code exécutable encapsulé. Avant le fichier PE, il existait un format appelé COFF utilisé dans les systèmes Windows NT. [31]

Le schéma du format PE

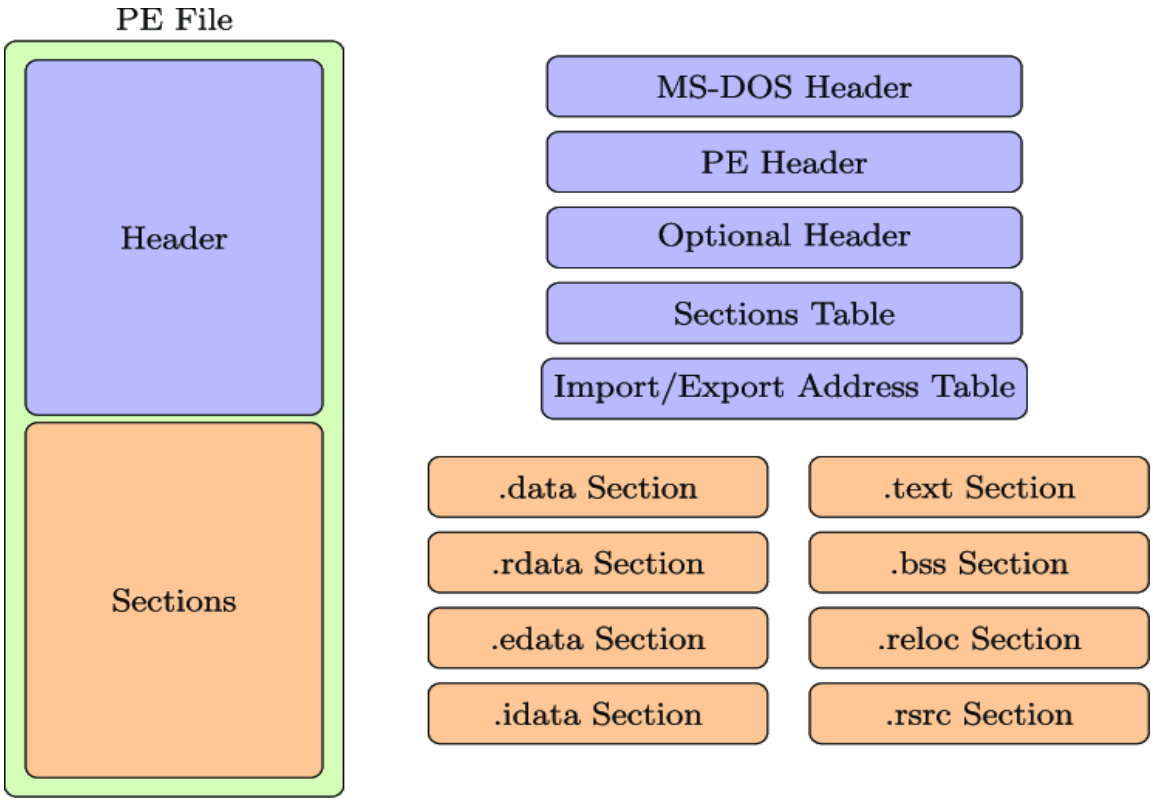


Figure10 : Structure du fichier PE [71]

En-tête MZ sous MS-DOS

Chaque fichier PE commence par une structure de 64 octets appelée en-tête DOS, c'est ce qui fait du fichier PE un exécutable MS-DOS. [32]

```
typedef struct _IMAGE_DOS_HEADER {      // DOS .EXE header
    WORD   e_magic;                     // Magic number
    WORD   e_cblp;                      // Bytes on last page of file
    WORD   e_cp;                        // Pages in file
    WORD   e_crlc;                      // Relocations
    WORD   e_cparhdr;                   // Size of header in paragraphs
    WORD   e_minalloc;                  // Minimum extra paragraphs needed
    WORD   e_maxalloc;                  // Maximum extra paragraphs needed
    WORD   e_ss;                        // Initial (relative) SS value
    WORD   e_sp;                        // Initial SP value
    WORD   e_csum;                      // Checksum
    WORD   e_ip;                        // Initial IP value
    WORD   e_cs;                        // Initial (relative) CS value
    WORD   e_lfarlc;                    // File address of relocation table
    WORD   e_ovno;                      // Overlay number
    WORD   e_res[4];                    // Reserved words
    WORD   e_oemid;                     // OEM identifier (for e_oeminfo)
    WORD   e_oeminfo;                   // OEM information; e_oemid specific
    WORD   e_res2[10];                  // Reserved words
    LONG   e_lfanew;                    // File address of new exe header
  } IMAGE_DOS_HEADER, *PIMAGE_DOS_HEADER;
```

Figure11 : En-tête PE sous MS-DOS [72]

Segment DOS

Après l'en-tête DOS vient le stub DOS qui est un petit exécutable compatible MS DOS 2.0 qui imprime simplement un message d'erreur disant "Ce programme ne peut pas être exécuté en mode DOS" lorsque le programme est exécuté en mode DOS.

En-tête PE

La partie NT Headers contient trois parties principales :

- **PE signature:**

Une signature de 4 octets qui identifie le fichier en tant que fichier PE.

- **File Header:**

Un en-tête de fichier COFF standard. Il contient des informations sur le fichier PE.

- **Optional Header:**

L'en-tête le plus important des en-têtes NT, son nom est l'en-tête facultatif car certains fichiers comme les fichiers objets ne l'ont pas, mais il est requis pour les fichiers image (fichiers comme les fichiers .exe). Cet en-tête fournit des informations importantes au chargeur du système d'exploitation**. [32]**

Table de section

La table de section suit immédiatement l'en-tête facultatif, c'est un tableau d'en-têtes de section d'image, il y a un en-tête de section pour chaque section du fichier PE. Chaque en-tête contient des informations sur la section à laquelle il se réfère. [32]

Sections

Les sections sont l'endroit où le contenu réel du fichier est stocké, celles-ci incluent des éléments tels que les données et les ressources utilisées par le

programme, ainsi que le code réel du programme. Il existe plusieurs sections, chacune ayant son propre objectif. [32]

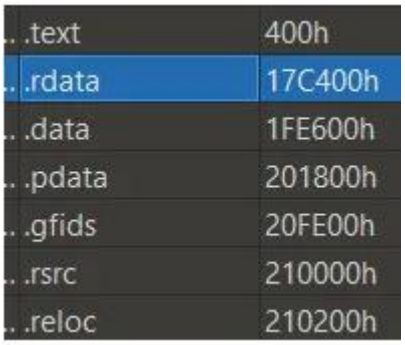


Figure 12: les sections [73]

Les principales sources d'infection

Email :

Si votre adresse électronique a été piratée, un programme malveillant peut forcer votre ordinateur à envoyer des emails avec des pièces jointes infectées ou des liens vers des sites Web malveillants. Lorsque le destinataire ouvre la pièce jointe ou clique sur le lien, le programme malveillant est installé sur son ordinateur, et le cycle se répète.

Supports physiques :

Les pirates informatiques peuvent charger des programmes malveillants sur des clés USB et attendre que des victimes peu méfiantes les branchent sur leur ordinateur. Cette technique est souvent utilisée dans l'espionnage d'entreprise.

Vulnérabilités :

Un défaut de sécurité dans un logiciel peut permettre à un programme malveillant d'obtenir un accès non autorisé à l'ordinateur, au matériel ou au réseau.

Téléchargements furtifs :

Téléchargements involontaires de logiciels effectués à l'insu ou non de l'utilisateur final.

Homogénéité :

Si tous les systèmes utilisent le même système d'exploitation et sont connectés au même réseau, le risque de propagation d'un ver à d'autres ordinateurs est plus élevé. [35]

Les approches de détection de malware

Analyse des malwares

L'analyse des logiciels malveillants consiste à utiliser des outils et des procédures pour comprendre le comportement et l'objectif d'un fichier suspect. Le processus vise à détecter et à atténuer toute menace potentielle. Ce processus pratique permet aux analystes de comprendre les fonctions, les objectifs et l'impact potentiel du logiciel malveillant. Pour y parvenir, les équipes de sécurité utilisent des outils d'analyse des logiciels malveillants. Ils évaluent et évaluent des échantillons de logiciels malveillants spécifiques, généralement dans un environnement confiné appelé sandbox. [36]

Approche statiques

Une analyse statique classique ne requiert pas l'exécution du code. Elle examine le fichier à la recherche de tout signe d'intention malveillante. Elle peut s'avérer utile pour identifier des infrastructures, des bibliothèques ou des fichiers compressés malveillants.

Les indicateurs techniques identifiés, comme les noms de fichier, les hachages, les chaînes (adresses IP, par exemple), les domaines et les données d'en-tête, peuvent être utilisés pour déterminer si un fichier est malveillant. Il est en outre possible d'utiliser des outils tels que des désassembleurs et des analyseurs de réseau pour observer le logiciel malveillant sans l'exécuter afin de recueillir des informations sur son fonctionnement.

Toutefois, dans la mesure où l'analyse statique n'exécute pas le code, les malwares sophistiqués peuvent présenter un comportement malveillant à l'exécution susceptible de passer inaperçu. Par exemple, une analyse statique pourrait ne pas détecter qu'un fichier génère une chaîne dynamique qui est ensuite

utilisée pour télécharger un fichier malveillant. C'est pourquoi les entreprises privilégient de plus en plus les analyses dynamiques pour bénéficier d'un meilleur éclairage sur le comportement des fichiers. [37]

Approche dynamique

L'analyse dynamique, également appelée analyse du comportement des logiciels malveillants, exécute le programme malveillant pour examiner son comportement. Bien entendu, l'exécution d'un logiciel malveillant comporte toujours un certain risque, de sorte que l'analyse dynamique doit être effectuée dans un environnement sûr. Un environnement « sandbox » est un système virtuel isolé du reste du réseau et qui peut exécuter des logiciels malveillants sans risque pour les systèmes de production. Une fois l'analyse terminée, le bac à sable peut être restauré à son état d'origine sans dommage permanent.

Lorsqu'un logiciel malveillant est exécuté, des indicateurs techniques apparaissent et fournissent une signature de détection que l'analyse dynamique peut identifier.

Un logiciel d'analyse dynamique surveille le système sandbox pour voir comment le logiciel malveillant le modifie. Les modifications peuvent inclure de nouvelles clés de registre, des adresses IP, des noms de domaine et des emplacements de chemin de fichier. L'analyse dynamique révélera également si le logiciel malveillant communique avec le serveur externe d'un pirate. Le débogage est une autre technique d'analyse dynamique utile. Pendant l'exécution du logiciel malveillant, un débogueur peut se concentrer sur chaque étape du comportement du programme pendant le traitement des instructions.

Les cybercriminels ont développé des techniques pour déjouer l'analyse dynamique. Un logiciel malveillant peut refuser de s'exécuter s'il détecte un environnement virtuel ou un débogueur. Le programme peut retarder l'exécution de sa charge utile nuisible ou nécessiter certaines entrées de l'utilisateur. Pour parvenir à la meilleure compréhension d'une menace de logiciel malveillant particulière, une combinaison d'analyses statiques et dynamiques est la plus efficace. [38]

Approche hybride

L'analyse statique de base n'est pas suffisamment fiable pour détecter le code malveillant sophistiqué et les logiciels malveillants avancés peuvent parfois rester dissimulés pour échapper à la technologie de sandbox. En combinant les techniques d'analyse statique et d'analyse dynamique, l'analyse hybride offre aux équipes de sécurité le meilleur des deux mondes, notamment parce qu'elle est capable de détecter le code malveillant qui tente de se dissimuler, puis d'extraire de nombreux indicateurs de compromission supplémentaires grâce à l'analyse statique de code jusque-là inconnu. L'analyse hybride contribue donc à détecter les menaces inconnues, y compris celles émanant des malwares les plus sophistiqués. [39]

Conclusion

Dans ce chapitre, nous avons donné les principales connaissances dans le domaine de la sécurité informatique, en précisant les différentes étapes d'une attaque ainsi que les différentes techniques utilisées par les attaquants et les outils et mécanismes de protection pouvant être employés contre ces attaques. [8] Ces informations sont très nécessaires à connaître pour la génération de scénarios d'attaque. En effet, c'est ces informations qu'un intrus peut exploiter pour réaliser son attaque. La génération des scénarios d'attaques est le domaine de recherche de plusieurs travaux.

Chapitre 02 : les techniques de machine Learning

Introduction

Avec leurs percées perturbatrices, l'apprentissage automatique et l'intelligence artificielle ont radicalement changé notre environnement. Les robots apprennent à distinguer les tâches grâce aux nombreuses techniques d'apprentissage en profondeur qui ont été développées et qui sont influencées par le réseau neuronal du cerveau humain. Il existe des télécommandes et des commandes de Smartphone disponibles. L'intelligence artificielle, l'apprentissage automatique et d'autres sujets seront abordés dans ce chapitre.

Intelligence Artificielle :

définition:

L'intelligence artificielle (IA) est une technique qui imite l'intelligence humaine et qui repose sur le développement et l'utilisation d'algorithmes exécutés dans un environnement informatique dynamique. Son objectif est de donner aux ordinateurs la capacité de penser et de se comporter comme des personnes. Trois éléments sont nécessaires pour y parvenir : des appareils électroniques des systèmes de gestion des données des algorithmes d'IA modernes (code) L'intelligence artificielle nécessite beaucoup de puissance de traitement et de données afin de reproduire le plus fidèlement possible le comportement humain.[42]

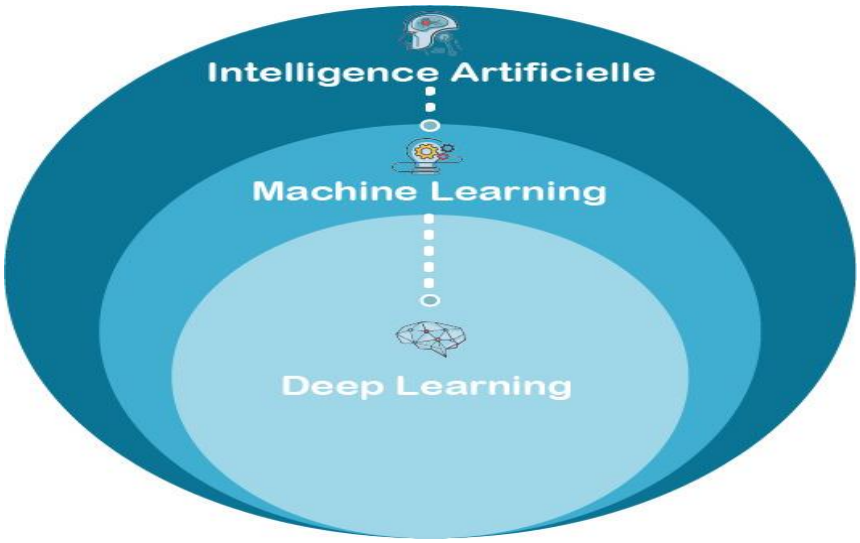


Figure 13: Schéma de l'intelligence artificielle [74]

Apprentissage:

Historique:

Trois périodes distinctes peuvent être identifiées dans le développement de l'apprentissage automatique. De nombreux algorithmes, y compris des réseaux de neurones et des machines à vecteurs de support, ont été créés dans la période initiale. Dans la deuxième étape, des puces dédiées à la formation de réseaux de neurones appelées Tensor Processing Units (TPU) et Graphics Processing Units (GPU) ont été créées pour prendre en charge ces méthodes. Enfin, des frameworks logiciels comme Apache MXNet et TensorFlow ont été développés dans la troisième phase pour simplifier la conception d'applications par les programmeurs en tirant parti de ces nouvelles approches et éléments matériels. [40]

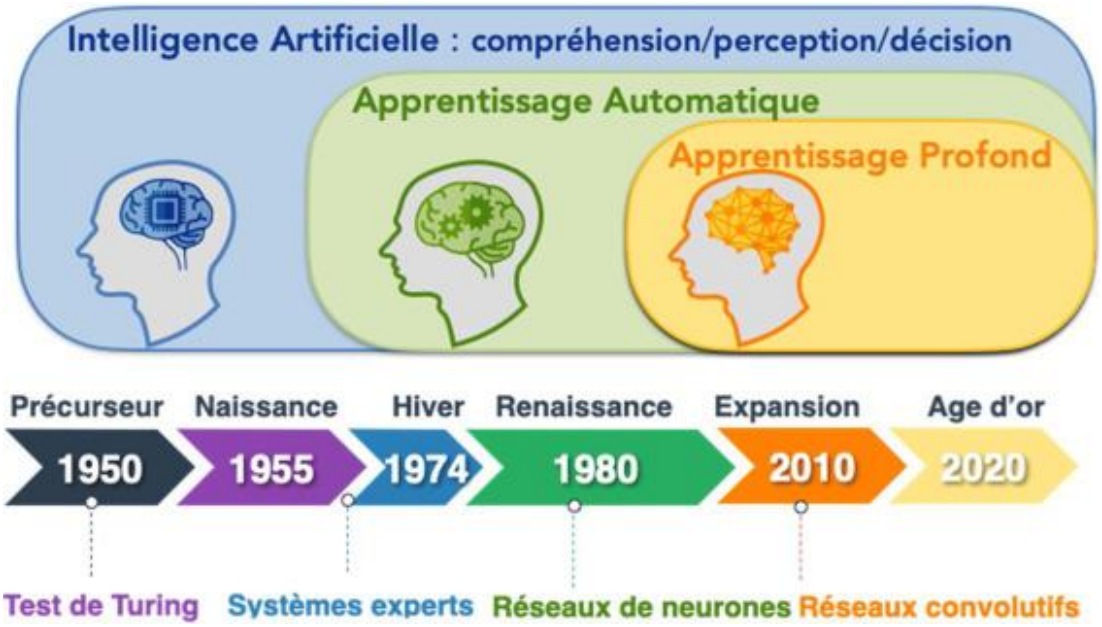


Figure 14: L'histoire de machine Learning [75]

Definition:

Une technologie en développement appelée apprentissage automatique permet aux ordinateurs d'apprendre de manière autonome à partir de données historiques. L'apprentissage automatique utilise une variété de techniques pour créer des modèles mathématiques et faire des prédictions basées sur des informations ou des

données antérieures. De nos jours, il est utilisé pour de nombreuses choses différentes, notamment les systèmes de recommandation, le filtrage des e-mails, le marquage automatique de Facebook, l'identification d'images et la reconnaissance vocale.[43].

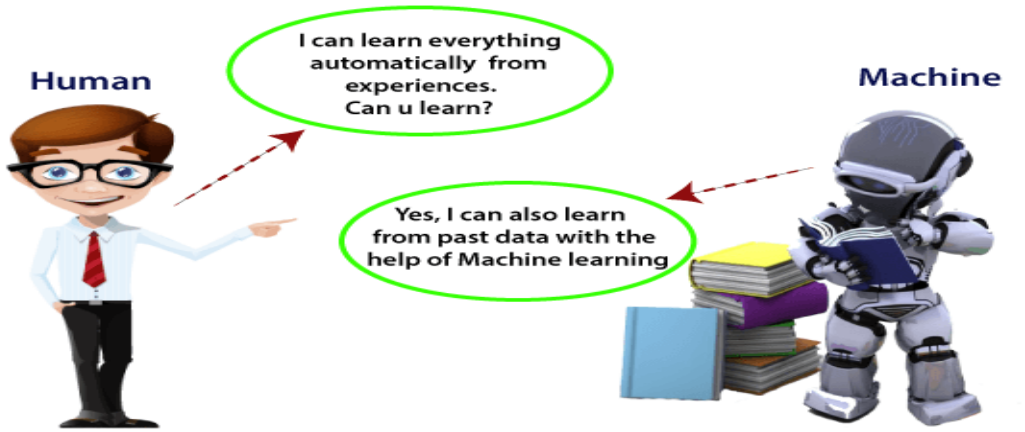


Figure15:machine learning [76]

Les types de machine learning:

1.1.1.Machine Learning supervisé

De nombreuses instances (entrées) sont fournies à l'algorithme pour apprendre, et ces exemples sont étiquetés, ce qui signifie que nous les associons à un résultat souhaité (sorties). L'objectif suivant de l'algorithme est d'identifier la loi qui permet de déterminer la sortie en fonction des entrées. Trouver la fonction optimale f(x) pour connecter l'entrée (x) et la sortie (y) est l'objectif (y). Deux catégories fondamentales de problèmes peuvent être résolues par l'apprentissage supervisé : les problèmes de classification et les problèmes de régression. [41]

1.1.2.Machine Learning non supervisé

L'algorithme, qui détermine la structure distinctive de l'entrée sans l'aide de l'homme, ne reçoit aucune étiquette. L'algorithme créera sa propre représentation, qu'une personne pourrait trouver difficile à comprendre. Afin de créer des regroupements homogènes à partir des observations, des modèles communs sont trouvés. Le regroupement et les relations sont les deux autres sous-catégories de

l'apprentissage non supervisé. Trouver des points communs dans les données pour créer des clusters est la notion sous-jacente au clustering. Les algorithmes d'association se concentrent sur la découverte de modèles dans les données, ce qui est un peu différent du clustering. Ces lois pourraient être exprimées par "Si X et Y sont vrais, alors l'événement Z peut se produire". [41]

1.1.3. Machine Learning par renforcement

Il se situe entre les deux premiers algorithmes en tant qu'intermédiaire. Cette méthode fonctionne via le mécanisme de récompense d'expérience plutôt que l'évaluation de données étiquetées. Pour améliorer les critères de décision et proposer une meilleure réponse au problème, la procédure est évaluée et réintroduite dans l'algorithme d'apprentissage. Cet apprentissage est basé sur l'expérience et orienté vers la décision (échecs et succès). [41]

Classification machine Learning :

L'objectif de la classification, une technique d'apprentissage automatique supervisé, est de prédire l'étiquette appropriée pour certaines données d'entrée. Dans la classification, le modèle est soigneusement entraîné à l'aide des données d'entraînement avant d'être évalué à l'aide des données de test, puis utilisé pour faire des prédictions sur des données fraîches et non contaminées.[44]

Classification binaire :

Les problèmes de classification d'apprentissage automatique qui n'ont que deux résultats possibles sont appelés problèmes binaires. À titre d'illustration, « oui » ou « non », « vrai » ou « faux », « spam » ou « pas de spam », etc. Bayes simples, la régression logistique, les arbres de décision et les machines à vecteurs de support sont quelques classifications binaires populaires algorithmes . [45]

Classification multi-classes :

Les problèmes de classification multi-classes n'ont pas l'idée de résultats normaux et anormaux et ont plus de deux résultats. Dans ce cas, chaque résultat est associé à une seule étiquette. Par exemple, catégoriser les visages, les espèces et les

photos, entre autres. Les arbres de choix, le renforcement progressif, les k voisins les plus proches et la forêt brute sont quelques exemples d'algorithmes multi-classes populaires. [45]

Classification multi-étiquettes :

Plusieurs étiquettes de classe peuvent être appliquées aux données dans une tâche de classification multi-étiquettes. Le modèle produira plusieurs résultats dans ce cas. Par exemple, un livre, un film ou une image peut appartenir à plusieurs genres ou contenir différents objets. Les arbres de décision multi-étiquettes, l'amplification de gradient multi-étiquettes et les forêts aléatoires multi-étiquettes sont quelques exemples d'algorithmes multi-étiquettes populaires. [45]

Classification déséquilibré :

La plupart des méthodes d'apprentissage automatique distribuent présumément les données de manière égale. Le déséquilibre résulte d'une répartition inégale des données. Un problème de classification connu sous le nom de problème de classification déséquilibrée survient lorsque la distribution de l'ensemble de données est faussée ou biaisée. Pour modifier la composition des échantillons de données, cette méthode utilise des procédures spécialisées. Le filtrage des spams, l'identification des maladies et la détection des fraudes sont quelques exemples de catégorisation déséquilibrée. [45]

Les différentes techniques de machine Learning :

Random Forest (Forêt d'arbres décisionnels)

Une méthode d'apprentissage basée sur des ensembles basée sur des arbres de décision est appelée forêts aléatoires ou forêts d'arbres de décision. La construction de nombreux arbres de décision pour le modèle de forêt aléatoire implique l'utilisation d'ensembles de données fractionnés à partir des données d'origine. et à chaque branche de l'arbre de décision, traduit un sous-ensemble des variables au hasard. Le mode de chaque prédiction de chaque arbre de décision est ensuite choisi par le modèle. [46]

Machine à vecteurs de support (SVM)

La machine à vecteurs de support est une autre approche moderne d'apprentissage automatique (SVM). Les machines à vecteurs de support dans l'apprentissage automatique sont des modèles d'apprentissage supervisés avec des algorithmes d'apprentissage associés qui examinent les données utilisées pour l'analyse de régression et de classification. Les SVM peuvent effectuer efficacement une classification non linéaire via ce que l'on appelle l'astuce du noyau, traduisant implicitement leurs entrées dans des espaces de caractéristiques de grande dimension, en plus d'effectuer une classification linéaire. Essentiellement, il s'agit de délimiter les classes. Les marges sont tracées pour réduire l'erreur de classification en maximisant la distance entre la marge et les classes. [47]

Algorithm KNN (K- Nearest Neighbors)

Les problèmes impliquant la classification et la regression peuvent tous deux être résolus en utilisant cette approche. Il est plus fréquemment utilisé dans l'industrie de la science des données pour résoudre les problèmes de classification. Il s'agit d'un algorithme simple qui garde une trace de tous les exemples disponibles et classe tout nouveau cas en demandant à ses k voisins leur approbation. La classe avec laquelle il partage le plus de caractéristiques reçoit alors le cas. Ce calcul est effectué à l'aide d'une fonction de distance. [48]

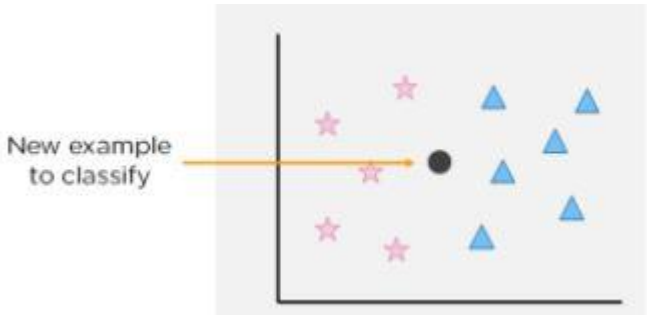


Figure 16: Data to be classified [77]

Figure 17: Classification using K-Nearest Neighbours[77]

Arbre de décision

Une approche d'apprentissage supervisé non paramétrique appelée arbre de décision est utilisée à la fois pour les problèmes de classification et de régression. Sa structure hiérarchique arborescente comprend un nœud racine, des branches, des nœuds internes et des nœuds feuilles.

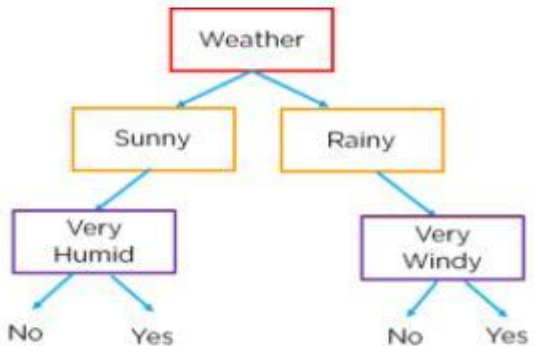


Figure 18 : Arbre de décision [82]

Le nœud racine d'un arbre de décision est le premier nœud et n'a pas de branches sortantes. Les nœuds internes, parfois appelés nœuds de décision, sont les endroits où vont les branches quittant le nœud racine. Les deux types de nœuds évaluent des sous-ensembles homogènes, appelés nœuds feuilles ou nœuds feuilles, en fonction des attributs disponibles. Les nœuds feuilles du fichier reflètent tous les résultats potentiels. [49]

Logistic Regression

Lorsque la variable dépendante est binaire, vous devez utiliser la régression logistique comme méthode d'analyse de régression (binaire). La régression logistique est une analyse prédictive, comme toutes les autres méthodes de régression. Une variable dépendante binaire et une ou plusieurs variables indépendantes nominales, ordinales, d'intervalle ou de niveau de rapport sont analysées à l'aide de la régression logistique pour caractériser les données et expliquer la relation entre les deux.[50]

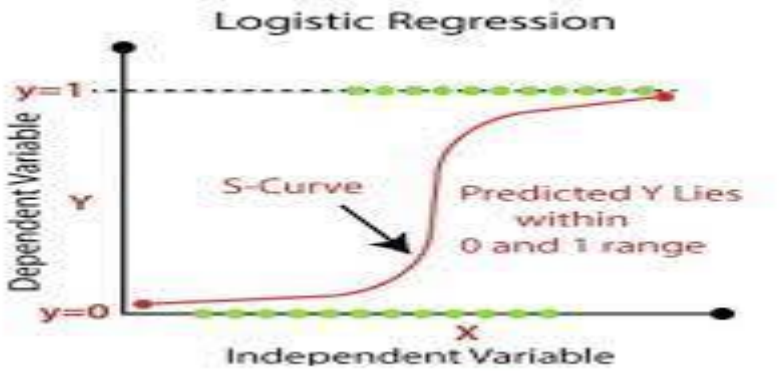


Figure 19: régression logistique [83]

Naïve Bayes

Naive Bayes est un algorithme d'apprentissage simple qui utilise la règle de Bayes et fait l'hypothèse forte que les attributs sont conditionnellement indépendants lorsqu'on leur donne des catégories. Bien que cette hypothèse d'indépendance soit souvent violée dans la pratique, Naive Bayes fournit encore souvent une précision de classification compétitive. Couplé à son efficacité de calcul et à de nombreuses autres propriétés souhaitables, cela a conduit à l'utilisation généralisée de Naive Bayes dans la pratique

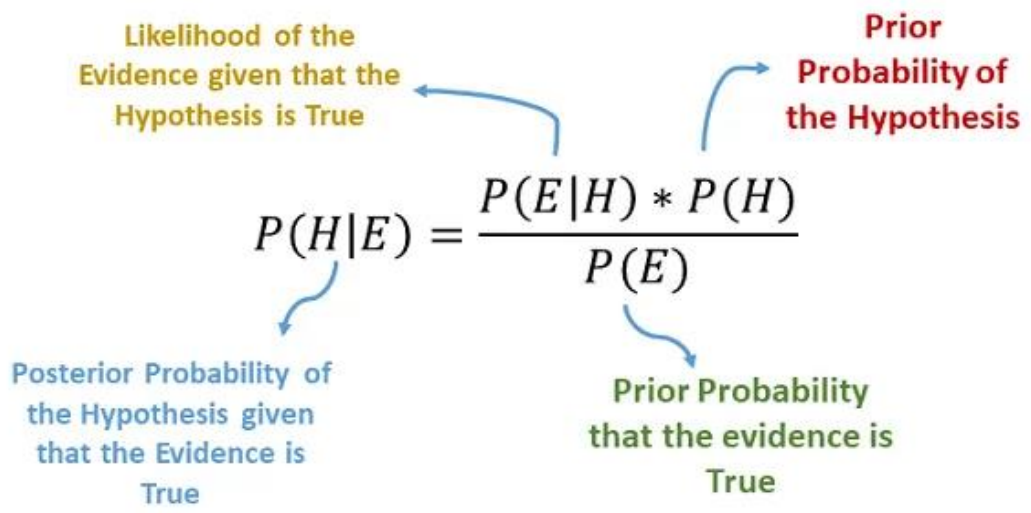


Figure 20 :naïve bayes [84]

Multi layer perceptron

Le Perceptron Multicouche (MLP) est l'un des modèles de réseaux de neurones les plus couramment utilisés dans le domaine de l'apprentissage en profondeur. Souvent appelés réseaux de neurones "vanille", les MLP sont plus simples que les modèles complexes de l'ère actuelle. Cependant, les techniques qu'il a introduites ont ouvert la voie à d'autres réseaux de neurones avancés.

Les perceptrons multicouches (MLP) sont utilisés dans diverses tâches, telles que l'analyse d'inventaire, la reconnaissance d'images, la détection de spam et la prédiction. [88]

Gradient Boosting :

Un algorithme Boosting spécifique est le Gradient Boosting. Le Boosting consiste à combiner plusieurs « élèves faibles » pour en faire un « élève fort », c'est-à-dire à combiner plusieurs algorithmes faibles pour en faire un beaucoup plus efficace et

satisfaisant. Pour estimer une variable d'intérêt, les « élèves faibles » sont regroupés en « élèves forts ».

Le principe d'une régression consiste à estimer nos résultats à l'aide du modèle 1, puis à utiliser les résidus de ce modèle comme variable cible du modèle 2, etc. [89]

$$x \rightarrow \text{model 1 } (\hat{y}),\quad x \rightarrow \text{model 2 } (\widehat{y-\hat{y}}),\quad \text{- - - } x \rightarrow \text{model N } (\hat{y}_n)$$

$$\tilde{y} = \sum_{i=1}^{n} \hat{y}_i$$

$$\text{Si } n = 2 \text{ par exemple: } \tilde{y} = \hat{y} + \widehat{y-\hat{y}}$$

$$x, w_0 \rightarrow \text{model 1 } (\hat{y}),\quad x, w_1 \rightarrow \text{model 2 } (\widehat{y-\hat{y}}),\quad \text{- - - } x, w_2 \rightarrow \text{model N } (\hat{y}_n)$$

Conclusion :

Faire des prédictions à partir de données est une forte utilisation de l'apprentissage automatique. Pourtant, il est crucial de garder à l'esprit que l'apprentissage automatique n'est aussi efficace que les données utilisées pour former les algorithmes. [51] Il est crucial d'utiliser des données de haute qualité qui indiquent les données du monde réel sur lesquelles l'algorithme sera utilisé afin de créer des prédictions précises.

Chapire03 : Description de projet

1. Introduction :

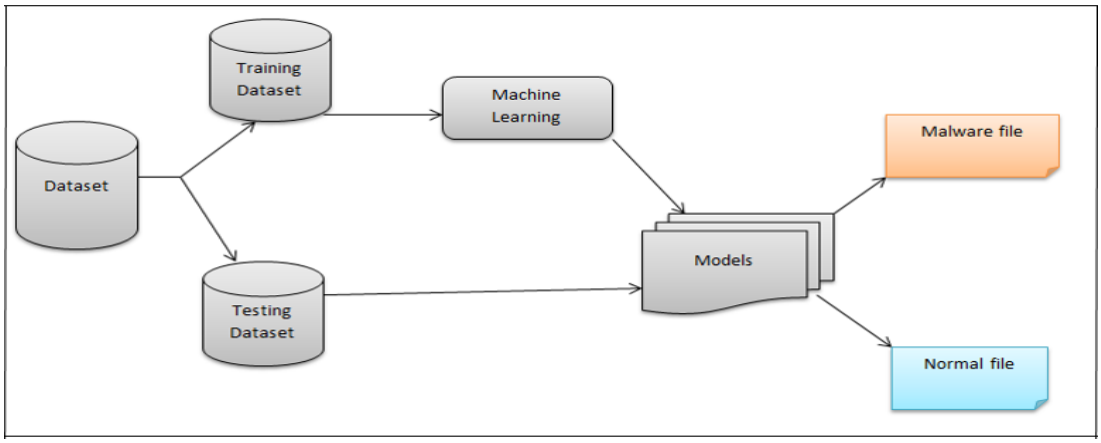
Les systèmes de détection de logiciels malveillants sont l'une des considérations les plus importantes en matière de sécurité réseau, car ils aident à détecter si un logiciel est malveillant ou non. Dans ce chapitre, nous présentons quelques techniques d'apprentissage automatique qui ont été appliquées dans le domaine de la détection de logiciels malveillants. Nous fournissons également une description complète du projet. Nous définissons les objectifs, la structure du système et l'ensemble de données utilisé. Nous avons également étudié et analysé de travaux la détection des logiciels malveillants fonctionne sur la base de techniques d'apprentissage automatique.

2. Objectif :

Nous visons dans notre projet à développer un modèle efficace basé sur des techniques de machine learning pour prédire les malwares dans les fichiers PE, ça se fait dans plusieurs étapes:

1. Préparation de dataset
2. Apprentissage : Ceci est accompli par l'utilisation d'algorithmes de classification tels que DT, KNN, RF, LR, SVM et autres.
3. Test et classification.

3. Architecture du système :



L’architecture du système.

4. Travaux connexes :

Dans cette section, nous mettons en évidence un certain nombre d'études qui ont appliqué des algorithmes d'apprentissage automatique pour détecter les logiciels malveillants :

Hussain, Abrar, et al [59] : Cet article propose un modèle de détection de logiciels malveillants de fichiers exécutables Windows basé sur l'apprentissage automatique. Le modèle proposé extrait les caractéristiques de l'en-tête du fichier PE et utilise six algorithmes d'apprentissage automatique (Random Forest, Support Vector Machine (SVM), Decision Tree, AdaBoost, Gaussian Naive Bayes (GNB) et Gradient Boosting) pour déterminer si le fichier est mal intentionné. Ensuite, les résultats de classification de ces algorithmes sont comparés et analysés pour sélectionner l'algorithme le plus adapté, les résultats de la comparaison montrent que l'algorithme de forêt aléatoire a les meilleures performances par rapport aux autres algorithmes, avec un taux de précision de 99,4%.

Akhtar, Muhammad Shoaib, and Tao Feng [60]: Cet article vise à améliorer le mécanisme de détection des logiciels malveillants polymorphes qui modifient constamment leurs propriétés de signature pour éviter d'être identifiés par des modèles de détection de logiciels malveillants basés sur les signatures. Lorsque l'expérience repose sur la mesure de la précision de reconnaissance du classificateur d'apprentissage automatique, en utilisant une analyse statique pour extraire des caractéristiques basées sur les données PE et en les comparant avec d'autres classificateurs, les résultats montrent que la méthode d'apprentissage automatique DT a une précision plus élevée que tout autre classificateur.

Xing, Xiaofei, et al [61]: Cet article vise à trouver des moyens de détecter les logiciels malveillants qui ne sont pas basés sur des algorithmes d'apprentissage automatique, propose ainsi un bon modèle qui combine la représentation d'image en niveaux de gris des logiciels malveillants avec un réseau de chiffrement

automatique dans un modèle d'apprentissage en profondeur et une analyse d'images en niveaux de gris des logiciels malveillants Faisabilité de la méthode. Reconstruire les erreurs basées sur les auto-encodeurs et exploiter la capacité de réduction de la dimensionnalité des auto-encodeurs pour obtenir une bonne identification des logiciels malveillants

Les résultats expérimentaux montrent que cette méthode est la plus précise (96%) par rapport aux autres méthodes basées sur des algorithmes d'apprentissage automatique, nécessite moins de temps de formation et de temps pour détecter les logiciels malveillants.

Moubarak, Joanna, and Tony Feghali [62]: Cet article vise à présenter l'utilisation d'algorithmes d'apprentissage automatique pour l'analyse des logiciels malveillants et comment utiliser la science des données pour détecter les logiciels malveillants. De nombreux algorithmes ont été implémentés, formés et testés, et les courbes Roc montrent que certains algorithmes fonctionnent mieux que d'autres. Random Forest fonctionne de manière satisfaisante par rapport aux autres algorithmes

Ganta, Venkata Gopi, et al [63]: Cet article vise à développer des méthodes de détection des ransomwares avec des techniques avancées pour échapper aux programmes antivirus traditionnels, puisque la plupart de ces programmes reposent sur des signatures. Par conséquent, certaines personnes proposent d'utiliser la forêt aléatoire, l'arbre de décision, la régression logistique, l'algorithme KNN et d'autres algorithmes d'apprentissage automatique pour classer les fichiers exécutables afin de déterminer s'ils sont infectés par un ransomware.

Choudhary, Sunita, and Anand Sharma [64]: Dans ce travail, la détection de programmes malveillants par apprentissage automatique dans la sécurité des systèmes informatiques, qu'ils soient nuisibles ou non, et la détermination de leur comportement par analyse statique ou dynamique, en appliquant des techniques d'apprentissage automatique, ont également été abordées. L'algorithme d'apprentissage automatique dans la détection des antivirus et des logiciels malveillants joue un rôle majeur dans la détection des logiciels malveillants, et

avec le développement et les progrès rapides du Web, les logiciels malveillants constituent une menace majeure.

5. Dataset :

Cet article vise à étudier la classification des malwares en fonction du contenu et des caractéristiques des fichiers. Ainsi, ClaMP_Integrated [88] dataset avait été utilisé, où ce jeu de données comprend 5184 instances.

En outre, comme mentionné précédemment, le total des échantillons était de 5184, dont (2722) des malwares et le reste des fichiers de (2488) bénins. De plus, le nombre total de caractéristiques était de (69), dont (54) des caractéristiques Row et (15) d'entre elles des caractéristiques Derived.

Afin de classer la présence ou l'absence de contenu malveillant dans les fichiers de l'ensemble de données en fonction du contenu et des caractéristiques des fichiers PE, différents algorithmes sont utilisés, ils sont; SVM, LG, NB, KNN, MLP, DT et Random Forêt.

6. Mesures de la performance de votre modèle :

- **Précision(Pr) :** La précision est la mesure des positifs corrects parmi tous les positifs prédits, y compris les vrais positifs et les faux positifs.

$$\boldsymbol{Pr = \frac{TP}{(TP + FP)}}$$

- **Recall (Rc) :** le rappel, la sensibilité, le taux de vrais positifs (TPR) ou la probabilité de détection est le rapport entre les prédictions positives correctes (TP) et le taux positif total (c'est-à-dire TP et FN).

$$\boldsymbol{R = \frac{TP}{(TP + FN)}}$$

- **F1-score (F 1) :** F-score ou F1-score, qui est une moyenne harmonique plus précise car elle ne varie pas beaucoup pour des valeurs extrêmement élevées. Un score F plus élevé (maximum 1) indique un meilleur modèle. [55]

$$\text{Score F} = \frac{2 * \text{Précision} * \text{Rappel}}{(\text{Rappel} + \text{Précision})}$$

- **Matrice de confusion :** Une matrice de confusion est un résumé des résultats de prédiction pour un problème de classification. Les prédictions correctes et incorrectes sont mises en évidence et ventilées par catégorie. Les résultats sont ainsi comparés avec les valeurs réelles. [57]

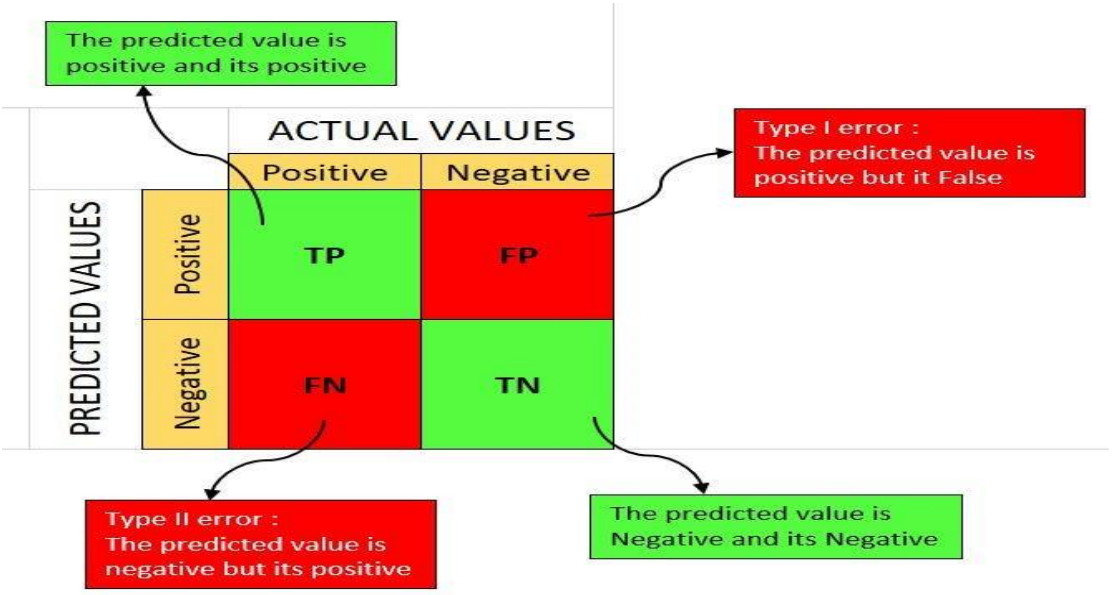
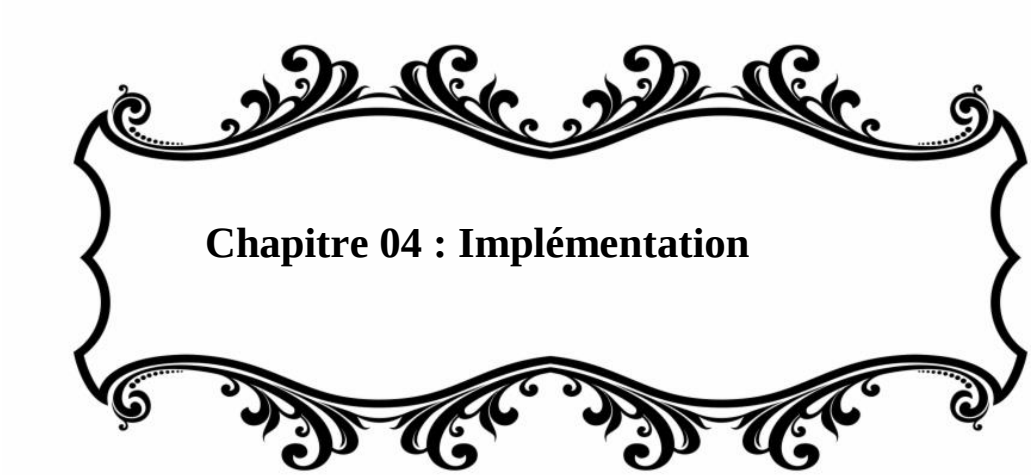


Figure 21: Matrice de confusion [85]

7. Conclusion :

Dans ce chapitre, nous avons présenté le fonctionnement général du système de détection de logiciels malveillants et appliqué différentes méthodes et techniques d'apprentissage automatique pour détecter les logiciels malveillants. Ces algorithmes d'apprentissage automatique ont des performances différentes selon les ensembles de données saisis. Cependant, l'utilisation les mêmes méthodes d'apprentissage automatique et techniques ne garantissent pas toujours les mêmes résultats et l'exposition aux attaques potentielles.


Chapitre 04 : Implémentation

1. Introduction

Ce chapitre détaille les nombreuses étapes de la mise en œuvre de notre algorithme, les résultats et une liste des outils qui ont été utilisés.

2. Outils de développement :

2.1. Langage Python :

Python est un langage de programmation multiplateforme orienté objet. Python peut être utilisé dans une variété de contextes, y compris le développement de logiciels, l'analyse de données et la gestion d'infrastructure, grâce à des bibliothèques spécialisées. Ainsi, contrairement au langage HTML, il n'est pas uniquement utilisé pour le développement Web. Python est un langage de programmation interprété qui permet d'exécuter du code sur n'importe quelle machine. Python rend simple et rapide la construction de programmes, et il peut être utilisé par les programmeurs novices et expérimentés. [52]

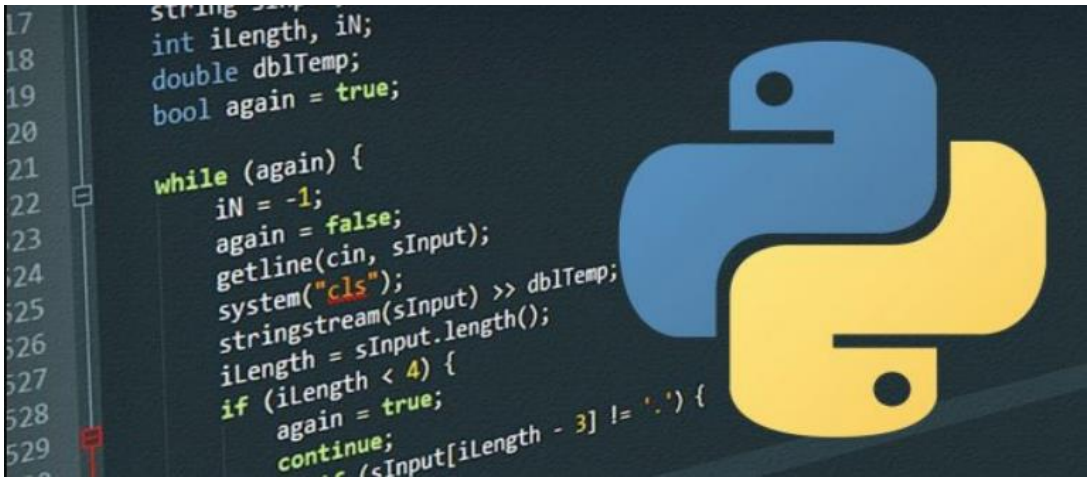


Figure22 : langage python [86]

3. Environnement de travail (Colab):

3.1. Anaconda Navigator :

Conçu pour faciliter la gestion et le déploiement des packages, Anaconda est une distribution open source des langages de programmation de science des données Python et R. Conda, le système de gestion de packages utilisé par Anaconda, est chargé de maintenir les versions des packages. Avant de lancer une installation, il examine l'environnement pour s'assurer qu'aucun autre framework ou package ne sera affecté. Plus de 250 éléments sont installés automatiquement dans la distribution Anaconda. Plus de 7500 packages open source supplémentaires, en plus du gestionnaire de packages conda et du gestionnaire d'environnement virtuel, peuvent être installés via PyPI. Une interface graphique (interface utilisateur graphique), Anaconda Navigator, est également mise à disposition en tant que substitut graphique de l'interface de ligne de commande. Anaconda Navigator, un composant de la distribution Anaconda, permet aux utilisateurs d'exécuter des applications et de gérer les packages, les environnements et les canaux Anaconda sans avoir besoin d'outils de ligne de commande. Les fonctionnalités de Navigator incluent la recherche de packages, leur installation sur un système, leur exécution et leur mise à niveau. [53]

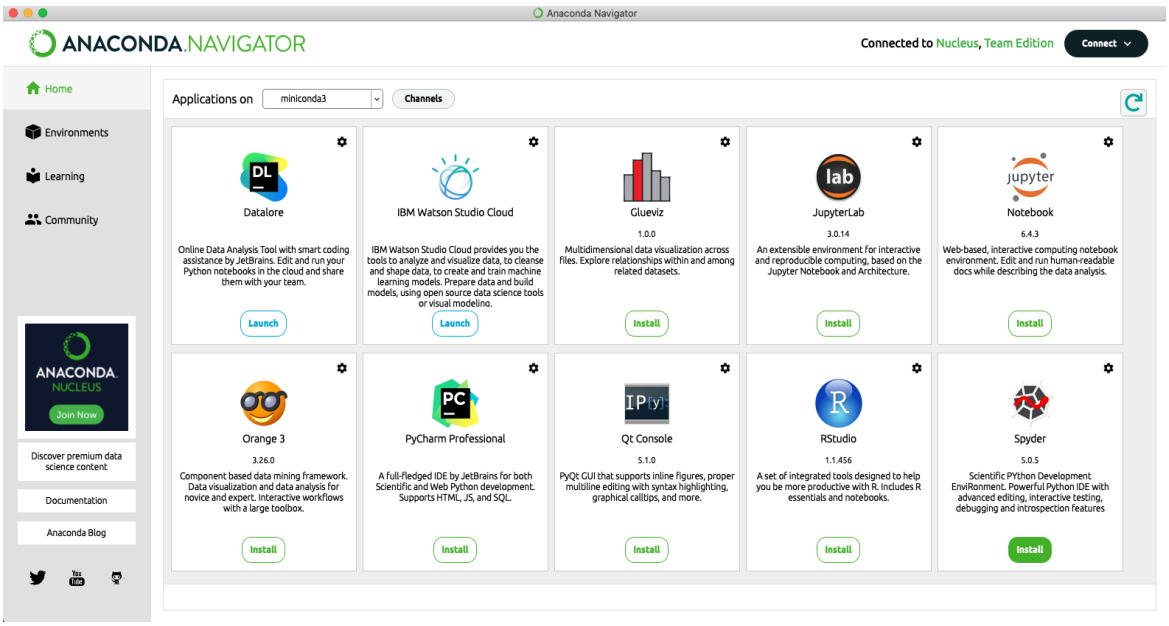


Figure 23: Représentation de l'interface d'anaconda. [87]

3.2. JupyterLab:

Un succès récent dans l'espace de développement interactif est Jupyter, qui s'exécute dans le navigateur. En exécutant des blocs de code et en connectant une sortie rich media, Jupyter modélise les programmes sous forme de blocs de code et simplifie la création interactive de blocs de code. Cependant, les espaces de noms et les systèmes de modules ne sont pas pris en charge par Jupyter. Étant donné que Jupyter utilise des blocs de code linéaires qui existent dans l'espace de noms global, il peut être difficile de créer d'énormes applications qui nécessitent une modularisation. Par conséquent, Jupyter n'est utilisé que comme couche de surface, tandis que les projets à code volumineux sont toujours développés dans des fichiers texte conventionnels.[54]

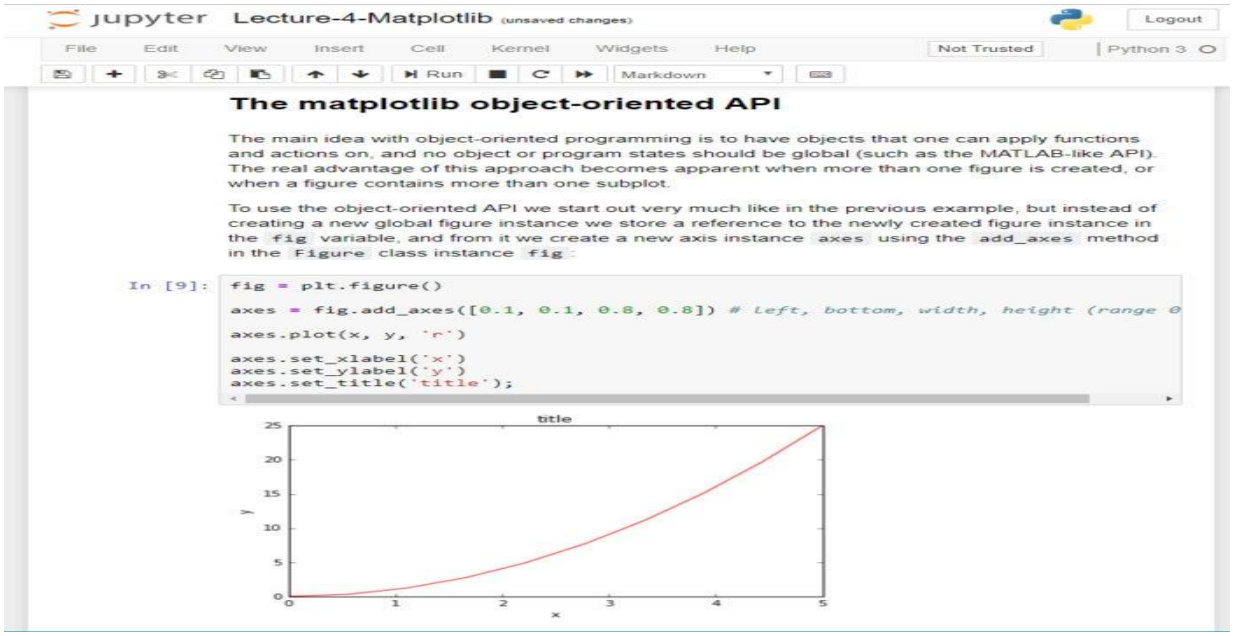
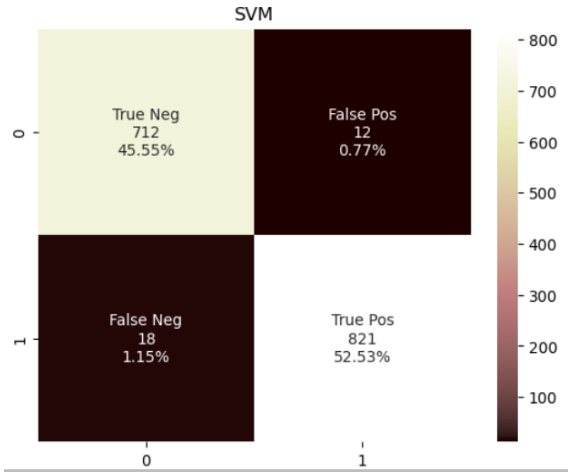
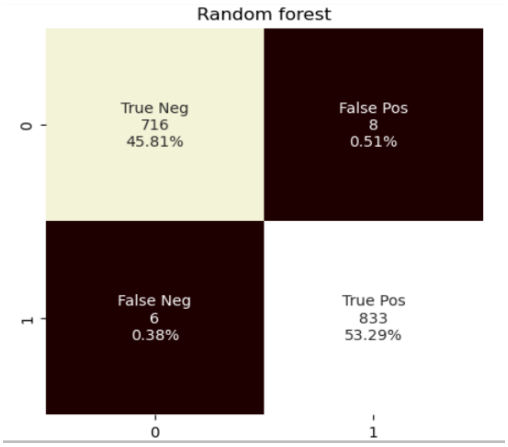
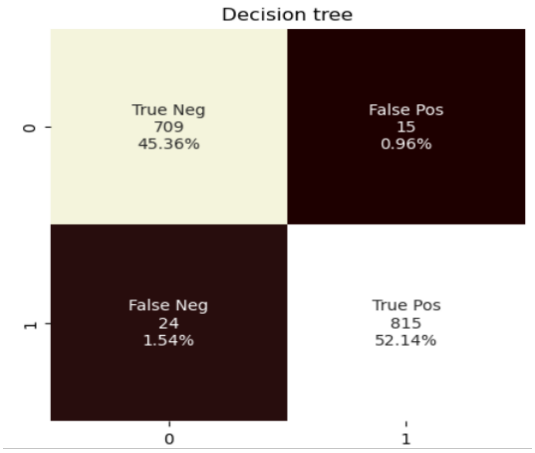


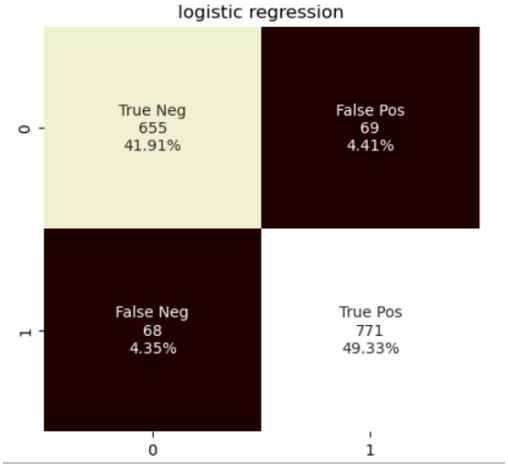
Figure24 : Représentation de la fenêtre de JupyterLab[88]

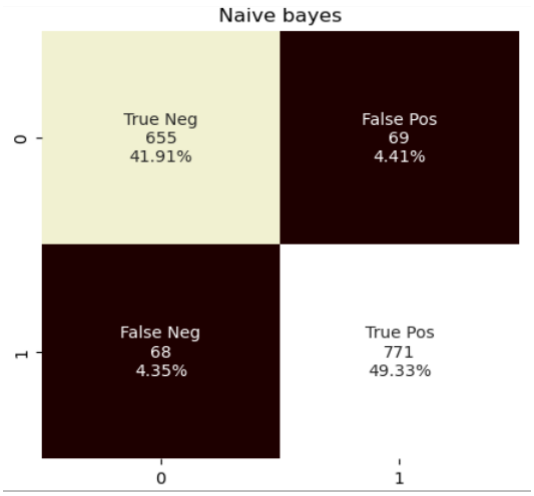
4. Résultats et discussions :

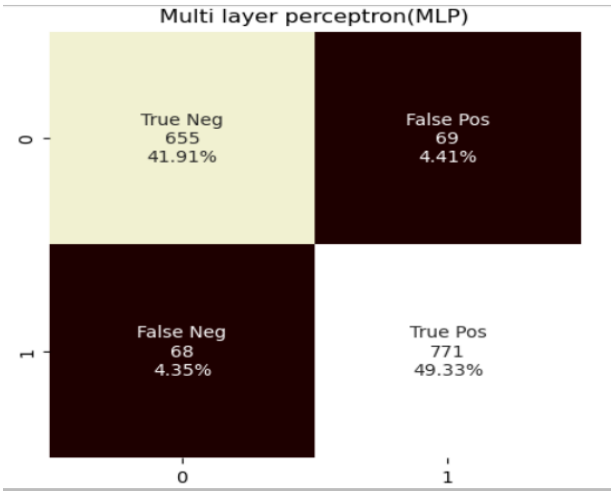












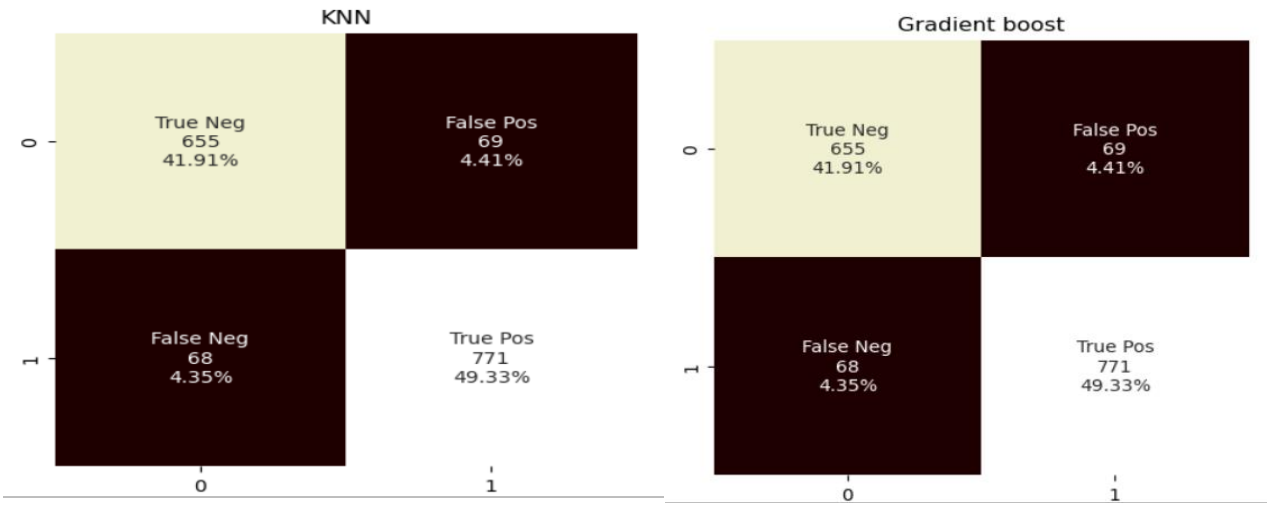


Figure 25 : Matrices de confusions les prédictions par rapport aux données réelles sur les données de test

Le tableau suivant représente les résultats obtenus :

Classifier	**Accuracy**	**Score F1**	**Précision**
SNV	77,99%	77,43%	0.98%
Rendom forest	99,10%	99,10%	0.99%
Decesion tree	97,50%	97,50%	0.98%
Logistic regression	79,33%	78,89%	0.75%
Naive bayes	57,00%	49,68%	0.97%
MLP	82,02%	81,94%	0.81%
Knn	91,23%	91,23%	0.91%
Gradient boosting	96,22%	96,21%	0.91%

Table 1 : les résultats obtenus du test

Les résultats de tableau 1 montrent que le classificateur Rendom Forest est meilleur que les autres classificateurs sur l'ensemble de données ClaMP_Integrated en fonction des paramètres sélectionnés. La précision du classificateur RF est de 99,10%.

5. Les codes source de notre projet :

```
from sklearn.svm import SVC
pipeSfrom sklearn.svm import SVC
pipeStd = Pipeline([('scaler', StandardScaler()), ('svm', SVC(kernel='poly', random_state=0))])
pipeStd.fit(x_train, y_train)
param_grid = {'svm__C':[0.1, 1, 10, 100, 200, 300],
    'svm__gamma':[0.001, 0.005, 0.01, 0.1, 1]}

grid = GridSearchCV(pipeStd, param_grid, cv = 5, n_jobs = -1)
grid.fit(x_train.to_numpy(), y_train)
print('SVC score after StdScaler: {:.3f}'.format(grid.score(x_test.to_numpy(), y_test)))
print("SVC's best score on cross validation: {:.3f}".format(grid.best_score_))
print("Classifier's best parameters: {}".format(grid.best_params_))
pred_val = grid.predict(x_test.to_numpy())
print(classification_report(y_test, pred_val, target_names=['benign', 'malicious'], digits=3))
```

Figure 26 : svm

```
#Random forest
from sklearn.ensemble import RandomForestClassifier
import time
print("Starting Random forest...")
classifier_rf = RandomForestClassifier(verbose=0,random_state=2)
start1 = time.time()
classifier_rf.fit(x_train, y_train)
end = time.time()
diffRF=end-start1
print("Training time: " + str(diffRF))
accuracy_test = classifier_rf.score(x_test,y_test)
print ("accuracy_test_RF=", accuracy_test*100)
starttest = time.time()
y_pred_rf = classifier_rf.predict(x_test)
endtest =time.time()
difftestRF = endtest-starttest
print("Test time: " + str(difftestRF))
print("RF accuracy: " + str(metrics.accuracy_score(y_test, y_pred_rf))
+ " F1 score:" + str(metrics.f1_score(y_test, y_pred_rf,average='weighted')))
matrixnv = confusion_matrix(y_test,y_pred_rf)
print(matrixnv)
print(classification_report(y_test, y_pred_rf))
```

Figure 27 : Rendom forest

```
#Decision tree
print("Starting Decision tree")
from sklearn.tree import DecisionTreeClassifier
Classifier_dt = DecisionTreeClassifier(random_state=2)
start1 = time.time()
clf = Classifier_dt.fit(x_train,y_train)
end = time.time()
diffDT=end-start1
print("Training time: " + str(diffDT))
starttest = time.time()
y_pred_dt = Classifier_dt.predict(x_test)
endtest =time.time()
difftestDT = endtest-starttest
print("Test time: " + str(difftestDT))
print("Decision Tree, accuracy: " + str(metrics.accuracy_score(y_test, y_pred_dt))
+ " F1 score:" + str(metrics.f1_score(y_test, y_pred_dt,average='weighted')))
matrixdt = confusion_matrix(y_test,y_pred_dt)
print(matrixdt)
print(classification_report(y_test, y_pred_dt))
```

Figure 28 : Decision tree

```
#Naive bayes
from sklearn.naive_bayes import GaussianNB
print("Starting Naive Bayes")
gnb = GaussianNB()
start1 = time.time()
gnb.fit(x_train, y_train)
end = time.time()
diffNB=end-start1
print("Training time: " + str(diffNB))
starttest = time.time()
y_pred_nb = gnb.predict(x_test)
endtestNB =time.time()
difftestNB = endtestNB-starttest
print("Test time: " + str(difftestNB))
print("Naive Bayes, accuracy: " + str(metrics.accuracy_score(y_test, y_pred_nb))
+ " F1 score:" + str(metrics.f1_score(y_test, y_pred_nb,average='weighted')))
matrixnv = confusion_matrix(y_test,y_pred_nb)
print(matrixnv)
print(classification_report(y_test, y_pred_nb))
```

Figure 29 :Naive bayes

```
#Logistic regression
import time
print("Starting Logistic regression All ")
start = time.time()
Classifier_LR = LogisticRegression ( random_state=0)
Classifier_LR.fit(x_train, y_train)
end = time.time()
diffLR=end-start
print("Training time: " + str(diffLR))
starttest = time.time()
y_pred_lr = Classifier_LR.predict(x_test)
endtest =time.time()
difftestLR = endtest-starttest
print("Test time: " + str(difftestLR))
print("Logistic Regression, accuracy: " +
str(metrics.accuracy_score(y_test, y_pred_lr)) +
" F1 score:" + str(metrics.f1_score(y_test, y_pred_lr,average='weighted')))
cm = confusion_matrix(y_test,y_pred_lr)
print("Confusion Matrix=\n",cm)
print("\n")
print(classification_report(y_test, y_pred_lr))
```

Figure 30 : Logistic regression

```
#Multi layer perceptron(MLP)
from sklearn.neural_network import MLPClassifier
print("Starting Multi layer perceptron")
start1 = time.time()
classifier_mlp = MLPClassifier( hidden_layer_sizes=(20,20,20),max_iter=500,
batch_size=1000, alpha=1e-4, activation = 'relu',solver='adam', verbose=2, tol=1e-4, random_state=2)
classifier_mlp.fit(x_train, y_train)
end = time.time()
diffMLP=end-start1
print("Training time: " + str(diffMLP))
starttest = time.time()
y_pred_mlp = classifier_mlp.predict(x_test)
endtest =time.time()
difftestMLP = endtest-starttest
print("Test time: " + str(difftestMLP))
print("MultiLayerPerceptron, accuracy: " + str(metrics.accuracy_score(y_test, y_pred_mlp)) +
" F1 score:" + str(metrics.f1_score(y_test, y_pred_mlp,average='weighted')))
matrixml = confusion_matrix(y_test,y_pred_mlp)
print(matrixml)
print(classification_report(y_test, y_pred_mlp))
```

Figure 31 : Multi layer perceptron

```
#KNN
from sklearn.neighbors import KNeighborsClassifier
classifier_knn= KNeighborsClassifier(n_neighbors=1, p=2)
start1 = time.time()
classifier_knn.fit(x_train,y_train)
end = time.time()
diffKNN=end-start1
print("Training time: " + str(diffKNN))
starttest = time.time()
y_pred_knn= classifier_knn.predict(x_test)
endtest =time.time()
difftestKNN = endtest-starttest
print("Test time: " + str(difftestKNN))
print("Knn, accuracy: " + str(metrics.accuracy_score(y_test, y_pred_knn))
+ " F1 score:" + str(metrics.f1_score(y_test, y_pred_knn,average='weighted')))
matrixnv = confusion_matrix(y_test,y_pred_knn)
print(matrixnv)
print(classification_report(y_test, y_pred_knn))
```

Figure 32 : KNN

```
#Gradient boost
from sklearn.ensemble import GradientBoostingClassifier
print("Starting Gradient boost")
start1 = time.time()
classifier_gb = GradientBoostingClassifier(n_estimators=10,
random_state=0,verbose=0)
classifier_gb.fit(x_train, y_train)
end = time.time()
diffGB=end-start1
print("Training time: " + str(diffGB))
starttest = time.time()
y_pred_gradient = classifier_gb.predict(x_test)
endtest =time.time()
difftestGB = endtest-starttest
print("Test time: " + str(difftestGB))
print("GradienBoost, accuracy: " + str(metrics.accuracy_score(y_test, y_pred_gradient))
+ " F1 score:" + str(metrics.f1_score(y_test, y_pred_gradient,average='weighted')))
matrixgb = confusion_matrix(y_test,y_pred_gradient)
print(matrixgb)
print(classification_report(y_test, y_pred_gradient))
```

Figure 33 : Gradient boost

6. Conclusion :

Dans ce chapitre, nous avons présenté les résultats des différentes expérimentations de notre système de détection, en ClaMP_Integrated dataset pour la détection des malwares.

Conclusion générale

Les logiciels malveillants sont la première menace de cyber-sécurité à laquelle nous serons confrontés à l'avenir, et ils deviennent chaque jour plus gros, plus dangereux et plus intelligents, et enfin, nous examinons tout ce que nous avons développé dans cet article. Un petit résumé visant à mettre en œuvre un système d'analyse des logiciels malveillants

Premièrement : Nous avons couvert la sécurité informatique en passant en revue les différents concepts de sécurité informatique, puis nous avons parlé des attaques logicielles, en nous concentrant sur les attaques de logiciels malveillants, et nous avons également abordé la structure des fichiers exécutables et leurs concepts.

Deuxièmement, nous avons examiné les bases et les méthodologies d'apprentissage automatique.

Troisièmement : Nous avons travaillé sur l'analyse et la classification des logiciels malveillants à l'aide de techniques d'apprentissage automatique.

En fin, les résultats auxquels nous sommes parvenus au terme de notre étude sont actuellement théoriquement et scientifiquement satisfaisants dans une certaine mesure.

Dans les années à venir, les logiciels malveillants cibleront non seulement de petites cibles, mais cibleront également des infrastructures et des dispositifs de contrôle de toutes sortes.

Dans les travaux futurs, nous espérons :

- Utiliser un ensemble de données contenant un plus grand nombre de fichiers

- Mettre en place un mécanisme pour analyser et tester de nouveaux logiciels malveillants

- Proposer une stratégie de test et de validation de logiciels

Web-graphiques

[1] :https://www.piloter.org/systeme-information/securite-informatique.htm?fbclid=IwAR0FkVwSrwP31-vuhcKIAiCDrLqqr01I9xJOvv-b9WCQZcAFBp3dFlGGQ34

[2]: https://www.cisco.com/c/fr_ca/products/security/what-is-it-security.html#~how-it-security-works

[3]: https://support.microsoft.com/fr-fr/topic/virus-informatiques-description-pr%C3%A9vention-et-r%C3%A9cup%C3%A9ration-53dc9904-0baf-5150-6e9a-e6a8d6fa0cb5

[4]:file:///C:/Users/asma/Downloads/Documents/Security_Mechanisms.pdf

[5]: https://waytolearnx.com/2019/06/qu-est-ce-qu-un-pare-feu.html

[6]: https://www.cyberpreventys.com/conseils-informatique-cybersecurite/quest-ce-que-cryptage-chiffrement/

[7] : https://culture-informatique.net/comment-ca-marche-cryptage/

[8] : https://web.maths.unsw.edu.au/~lafaye/CCM/attaques/attaques.htm

[9] : https://www.ibm.com/fr-fr/topics/cyber-attack

[10]:https://www.ipe.fr/quels-sont-les-differents-types-dattaques-informatiques/?fbclid=IwAR3_zL10IEI69TCuJhYFTSW0c8zUYNdQg_vpnHFa9wukaEF2XYTd-_nwClk

[11]: https://www.lesdigiteurs.fr/s-informer/8-types-de-cyberattaque

[12] : https://boowiki.info/art/methodes-de-cryptanalyse/l-attaque-anniversaire.html

[13] : https://fr.techdico.com/traduction/francais anglais/t%C3%A9l%C3%A9chargement+furtif.html

[14] : https://www.avast.com/fr-fr/c-malware

[15] :https://support.microsoft.com/fr-fr/topic/virus-informatiques-description-pr%C3%A9vention-et-r%C3%A9cup%C3%A9ration-53dc9904-0baf-5150-6e9a-e6a8d6fa0cb5

[16] : https://www.avast.com/fr-fr/c-malware

[17] : https://www.le-vpn.com/fr/principaux-types-de-malwares/

[18] : https://fr.malwarebytes.com/ransomware/

[19] : https://www.proofpoint.com/fr/threat-reference/computer-virus

[20] : https://softwarelab.org/fr/virus-informatique

[21] : https://byjus.com/govt-exams/computer-virus/

[22] : https://uniserveit.com/blog/what-are-the-different-types-of-computer-viruses

[23] : https://fr.norton.com/blog/privacy/what-is-a-computer-worm

[24] : https://www.securiteinfo.com/attaques/malwares-virus-spam-logiciels-indesirables/vers.shtml

[25] : https://www.educba.com/types-of-computer-worms/

[26] : https://www.makeuseof.com/types-of-computer-worms/

[28]: https://www.websecurity.digicert.com/fr/ca/security-topics/what-are-malware-viruses-spyware-and-cookies-and-what-differentiates-them

[29] : https://us.norton.com/blog/malware/types-of-malware#fileless

[30] :https://fracademic.com/dic.nsf/frwiki/1358550?fbclid=IwAR2WMaub5zHEGjrxcfWSaZ949maQfwQQb2XXDJjRO_-qU2nwFFt2jKFcV-Y#Historique

[31] :https://resources.infosecinstitute.com/topic/2-malware-researchers-handbook-demystifying-pe-

file/?fbclid=IwAR3MvbQ0ZHVhFnXB5l_dsyAOo2EZcX9DrCJNpb2CVQB5sUBdsRCsqqdSpts

[32] : https://0xrick.github.io/win-internals/pe2/#dos-header

[33] : https://medium.com/ax1al/a-brief-introduction-to-pe-format-6052914cc8dd

[34] : https://fr.wikipedia.org/wiki/Portable_Executable

[35] : https://www.kaspersky.fr/resource-center

[36] :https://www.intezer.com/blog/incident-response/the-role-of-malware-analysis-in-cybersecurity/

[37] : https://www.crowdstrike.fr/cybersecurity-101/malware/malware-analysis/

[38] : https://www.n-able.com/blog/malware-analysis-steps

[39] : https://www.crowdstrike.fr/cybersecurity-101/malware/malware-analysis/

[40]: https://www.intelligence-artificielle-school.com/machine-learning-quest-ce-que-cest/

[42] https://www.netapp.com/fr/artificial-intelligence/what-is-artificial-intelligence/

[43] https://www.expert.ai/blog/machine-learning-definition/

[44] : https://www.datacamp.com/blog/classification-machine-learning

[45] : https://emeritus.org/blog/artificial-intelligence-and-machine-learning-classification-in-machine-learning/

[46]: https://moncoachdata.com/blog/modeles-de-machine-learning-expliques/

[48]: https://mobiskill.fr/blog/conseils-emploi-tech/quels-sont-les-differents-algorithmes-en-machine-learning/

[49]: https://www.ibm.com/fr-fr/topics/decision-trees

[50]: https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/what-is-logistic-regression/

[51] https://www.alibabacloud.com/topic-center/tech/19tggrvkimpl-conclusion-of-machine-learning-alibaba-cloud

[52] https://www.futura-sciences.com/tech/definitions/informatique-python-19349/

[55] https://geekflare.com/fr/confusion-matrix-in-machine-learning/?fbclid=IwAR3XYTRuiI6h32QewQOrn9UUVf8YUGcmPQAI5EoJ12oXzAVfbW-N7K1JLQo

[57] https://www.lebigdata.fr/confusion-matrix-definition

[58] https://www.talend.com/resources/data-privacy/

[65] https://fr.123rf.com/photo_16573899_sch%C3%A9ma-de-la-s%C3%A9curit%C3%A9-informatique.html

[66]https://www.istockphoto.com/fr/vectoriel/ensemble-dic%C3%B4nes-de-ligne-de-cybers%C3%A9curit%C3%A9-illustration-vectorielle-contour-gm1338003094-418723303

[67] :https://waytolearnx.com/2019/06/qu-est-ce-qu-un-pare-feu.html?fbclid=IwAR1t9dhqujvmI7m3xHex2M82B2X5zgfeDUCP_MgF1Uk9BvVPJ_S8BjmBmw

[68]:https://parlonssciences.ca/ressources-pedagogiques/documents-dinformation/protection-des-donnees-introduction-au-cryptage?fbclid=IwAR1eMWUQW_kiHlBf1rTrX2XO06qWxvCUmgSXrSPZ0Ghn7nTug1pPSyWvn18

[69] https://www.axis-solutions.fr/cyberattaques-les-5-types-les-plus-courants/

[70]https://apcpedagogie.com/introduction-aux-attaques-informatiques/

[71]:https://www.bbc.co.uk/bitesize/topics/zv63d2p/articles/zcmbgk7

[72]: https://us.norton.com/blog/malware/types-of-malware#fileless

[73] https://fr.wikipedia.org/wiki/Portable_Executable

[74] https://medium.com/ax1al/a-brief-introduction-to-pe-format-6052914cc8dd

[75] https://mbamci.com/intelligence-artificielle-benefices-sante/

[76]https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.sciencedirect.com%2Fscience%2Farticle%2Fpii%2FS1957255720300407&psig=AOvVaw087ZcpjmrO4fX5lCqCH6cx&ust=1685898098245000&source=images&cd=vfe&ved=0CBMQjhxqFwoTCJiO06rKp_8CFQAAAAAdAAAAABAE

[77] :https://www.javatpoint.com/machine-learning

[78] : https://www.javatpoint.com/machine-learning

[79] :https://www.simplilearn.com/tutorials/machine-learning-tutorial/classification-in-machine-learning

[80] :https://www.simplilearn.com/tutorials/machine-learning-tutorial/classification-in-machine-learning

[81] : https://www.simplilearn.com/tutorials/machine-learning-tutorial/classification-in-machine-learning

[82]:https://sparkbyexamples.com/machine-learning/logistic-regression-explained-with-examples/

[83]:https://medium.com/analytics-vidhya/na%C3%AFve-bayes-algorithm-5bf31e9032a2

[84] :https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5

[85] :https://www.tresfacile.net/le-langage-python-les-principale-raisons-qui-poussent-a-apprendre-ce-langage/

[86]: https://github.com/napari/packaging/issues/22

[87]: https://www.infoworld.com/article/3347406/what-is-jupyter-notebook-data-analysis-made-easier.html

[88]: https://www.kaggle.com/datasets/saurabhshahane/classification-of-malwares

Bibliographiques:

[27]: Avoine, Gildas, et al. "Sécurité informatique." Vuibert, (2010).

[41]: Moubarak, Joanna, and Tony Feghali. "Comparing Machine Learning Techniques for Malware Detection." ICISSP 10 (2020): 0009373708440851.

[47]: [Mahesh, Batta. "Machine learning algorithms-a review." International Journal of Science and Research (IJSR).[Internet] 9 (2020): 381-386.]

[53] Jadhav, P., Mulla, A., Bhoi, G., Raj, S., & Nambiar, S. Mobile Botnet Detection.

[54] Li, H., Bao, F. S., Xiao, Q., & Tian, J. (2023). Codepod: A Namespace-Aware, Hierarchical Jupyter for Interactive Development at Scale. *arXiv preprint arXiv:2301.02410*.

[56] Rahmad, F., Y. Suryanto, and K. Ramli. "Performance comparison of anti-spam technology using confusion matrix classification." IOP Conference Series: Materials Science and Engineering. Vol. 879. No. 1. IOP Publishing, 2020.

[59] Hussain, Abrar, et al. "Malware detection using machine learning algorithms for windows platform." *Proceedings of International Conference on Information Technology and Applications: ICITA 2021*. Singapore: Springer Nature Singapore, 2022.

[60] Akhtar, Muhammad Shoaib, and Tao Feng. "Malware Analysis and Detection Using Machine Learning Algorithms." *Symmetry* 14.11 (2022): 2304.

[61] Xing, Xiaofei, et al. "A malware detection approach using autoencoder in deep learning." *IEEE Access* 10 (2022): 25696-25706

[62] Moubarak, Joanna, and Tony Feghali. "Comparing Machine Learning Techniques for Malware Detection." *ICISSP* 10 (2020): 0009373708440851

[63] Ganta, Venkata Gopi, et al. "Ransomware Detection in Executable Files Using Machine Learning." *2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*. IEEE, 2020.

[64] Choudhary, Sunita, and Anand Sharma. "Malware detection & classification using machine learning." *2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)*. IEEE, 2020