

Département Génie Informatique

Mémoire de projet de fin d'études

**Diplôme Universitaire de
Technologie
Filière : Génie logiciel**

**Présenté par :
Abdelkrim BELLAGNECH
Yassine BOUSMARA**

Thème

**Conception et Implémentation d'un Pare-feu
d'Application Web Entraîné par Machine
Learning pour la Détection des Attaques par
Injection SQL**

**Encadré par :
Mr. Abdallah RHATTOY**

Soutenu le 15/06/2023 devant le jury

**Pr. A. RHATTOY
Pr. M. L. HASNAOUI**

**UMI-EST-Meknès
UMI-EST-Meknès**

Année universitaire : 2022/2023

REMERCIEMENT



Cher Monsieur Abdallah RHATTOY et toute l'équipe administrative et professorale de l'ESTM,

Je tiens à vous adresser mes sincères remerciements pour votre soutien tout au long de ce travail. Votre contribution a été d'une valeur inestimable et j'aimerais exprimer ma profonde gratitude envers vous tous.

En premier lieu, je souhaite remercier Monsieur Abdallah RHATTOY , mon Encadrant de stage, pour sa confiance en moi. Je suis extrêmement reconnaissant(e) de m'avoir offert cette opportunité et d'avoir été là pour me guider tout au long du processus. Votre compréhension et votre générosité ont été des sources d'inspiration et d'encouragement constantes, et je vous en suis sincèrement reconnaissant(e).

Je tiens également à remercier chaleureusement tout le corps administratif et professoral de l'ESTM. Votre dynamisme et votre savoir-faire ont joué un rôle essentiel dans notre formation. Je suis particulièrement reconnaissant(e) envers les enseignants pour leur sympathie et l'aide précieuse qu'ils nous ont apportée. Leur passion pour l'enseignement a eu un impact significatif sur notre parcours académique, et je leur en suis très reconnaissant(e).

LISTE DES MATIERES

Introduction générale.....	02
CHAPITRE 1 : Etude préalable	04
I. Présentation.....	05
II. Objectifs	05
CHAPITRE 2 : Contexte général du projet.....	06
I. Introduction.....	07
II. Présentation du projet.....	07
1. Problématique.....	07
2. Solution envisagée.....	08
3. Contraintes.....	08
4. Planning prévisionnel du déroulement du Projet.....	08
5. Conclusion.....	09
CHAPITRE 3: Le Web.....	10
I. Introduction.....	11
II. Applications web.....	11
2.1 Fonctionnement d'une application web.....	11
2.2 Architecture des applications web	12
2.3 Vulnérabilités des applications web	14
III. Les attaques web	15
1. Attaques par déni de service (DoS) et par déni de service distribué (DDoS)...	15
2. Attaque de l'homme du milieu (MitM)	16
3. Attaque par injection SQL	17
4. Cross-Site Scripting (Attaque XSS).....	18
5 Attaque Brute-Force.....	18
IV- Solutions de filtrage Web et problèmes de contournement des signatures	19
V- Techniques furtives d'évasion aux systèmes de filtrage	20
1- Variation de la casse(MAJUSCULE - minuscule)	21
2- Espacement	21
3- Commentaires	21
VI- Conclusion	22
CHAPITRE 4: Machine Learning	23

I- Introduction	24
II- Techniques d'apprentissage automatique	24
III- Différents types d'apprentissage	24
1- L'apprentissage supervisé.....	24
2- L'apprentissage insupervisé.....	26
3- Apprentissage par renforcement	27
IV- Arbres de décision	28
1- Principe général	28
2- Exemple introductif	29
3- Données	30
4- Construction de l'arbre.....	31
4.1- Mesure de la pureté des feuilles	31
4.2- Algorithme de construction	32
5- Élagage de l'arbre	32
6- Gestion des données manquantes	33
V- Conclusion	34
CHAPITRE 5 : Conception et mise en œuvre du système	35
I- Expression des besoins	36
II- Explication du système réalisé	36
CHAPITRE 6 : Réalisation	38
I. Langages et outils du travail utilisé	39
1- Python	39
2- Anaconda	39
3- Jupyter notebook	40
4- Burpsuite	40
5- Acunetix	42
6- CSVCleaner	43
7- Visual Studio Code	43
II- Importation des bibliothèques et des modules nécessaires.....	44
III- Datasets	45
1- Génération des datasets	45
2- Importation des datasets	46
3- Dataset principale	47
IV- Clustering	47
1- API fonctionnelle	48
2- Kmeans Clustering	49
3- Création du modèle	49
V- Proxy réalisé	51
VI- Déploiement et expérimentation	51
1- Cas de détection d'une attaque SQLInjection	54
2- Cas normal	54
Conclusion	56
REFERENCES	58

LISTE DES FIGURES

FIGURE 1. Les attaques web détectées par Kaspersky en avril 2022	07
FIGURE 2. Diagramme de GANTT	08
FIGURE 3. Fonctionnement d'une application web	12
FIGURE 4. Niveaux d'une application web	13
FIGURE 5. Application à n-niveaux	14
FIGURE 6. Vulnérabilités d'une application web	15
FIGURE 7. Attaques DoS et DDoS	16
FIGURE 8. Détournement de session	17
FIGURE 9. Injection SQL	17
FIGURE 10. Attaque XSS	18
FIGURE 11. Attaque Brute Force	19
FIGURE 12. Phases d'analyse et de log de ModSecurity source	19
FIGURE 13. Variation de la casse	21
FIGURE 14. Utilisation de la tabulation	21
FIGURE 15. Suppression des espaces	21
FIGURE 16. Commentaire pour déguiser la requête SQL	21
FIGURE 17. L'apprentissage supervisé	25
FIGURE 18. Cas de régression	25
FIGURE 19. L'apprentissage de classification	26
FIGURE 20. Algorithmes utilisant l'apprentissage non supervisé	26
FIGURE 21. Apprentissage par renforcement	28
FIGURE 22. Arbre de décision pour l'étude sur la réussite des étudiants	35
FIGURE 23. Bibliothèques utilisées	44
FIGURE 24. Echantillonnage des datasets	46
FIGURE 25. Exemple d'un api fonctionnel	48
FIGURE 26. Création du modèle	49
FIGURE 27. Visualisation graphique 3d du modèle	50
FIGURE 28. Visualisation graphique 2d du modèle	50
FIGURE 29. Site web concerné par la protection	52
FIGURE 30. Configuration du proxy	52
FIGURE 31. Exécuter le notebook IPS proxy	53
FIGURE 32. Cas d'une attaque SQLInjection	54
FIGURE 33. Cas normal	55

Résumé

Internet comptait 4,95 milliards d'utilisateurs actifs en janvier 2021, soit l'équivalent de 62,5% de la population mondiale. Ces statistiques témoignent de l'accessibilité publique du web au cours des années précédentes. Cependant, Amazon Web Services (AWS) a prétendu s'être défendu en février 2020 contre une attaque DDoS (Déni de Service Distribué) de 2,3 téraoctets par seconde.

Dans ce projet, nous nous intéressons à la détection d'attaques web, plus précisément SQLInjection à l'aide de l'apprentissage automatique (Machine Learning). Nous allons d'abord construire un système basé sur l'apprentissage automatique qui analyse le trafic réseau entrant afin de distinguer ce qui est bénin de ce qui ne l'est pas.

La cybercriminalité a constamment augmenté ces dernières années, ce qui oblige les administrateurs à sécuriser correctement leurs réseaux informatiques. Ils ont généralement recours à des systèmes de détection d'intrusion pour repérer les attaques qui pénètrent dans le réseau ou celles qui ont échappé aux pare-feu et aux routeurs de filtrage.

Les attaques SQL Injection peuvent mettre en péril votre système. Apprenez à vous en protéger dès maintenant.



Introduction générale aux attaques web

Introduction générale



La sécurité des systèmes informatiques consiste à protéger l'accès et la manipulation des données et des ressources d'un système par des mécanismes d'authentification, d'autorisation, de contrôle d'accès, etc. Cependant, avec l'ouverture des entreprises et des personnes à Internet, l'assurance de la sécurité des systèmes devient très difficile, du fait que les attaques et les intrusions augmentent de plus en plus et elles deviennent de plus en plus complexes et difficiles à éviter. Ce chapitre introductif présente les notions de base de la sécurité des systèmes informatiques et les différentes attaques possibles qui peuvent se produire, ainsi que les mécanismes et outils pouvant être mis en place pour assurer la sécurité.

La sécurité informatique est l'ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires et mis en place pour réduire la vulnérabilité d'un système contre les menaces accidentelles et intentionnelles. La sécurité consiste à assurer :

- **La confidentialité** : assurer que l'information ne sera lue que par les personnes autorisées.
- **L'intégrité** : assurer que les informations ne peuvent être modifiées ou altérées que par les personnes autorisées.
- **La disponibilité** : assurer que l'information est disponible pour les personnes autorisées.
- **L'authentification** : vérifier l'identité d'un utilisateur pour lui associer des droits d'accès.
- **La non-répudiation** : garantir qu'aucun des correspondants ne pourra nier la transaction (l'envoi ou la réception des données).

Une attaque informatique est l'exploitation d'une faille d'un système informatique à des fins non connues par l'exploitant du système et est généralement préjudiciable. Les objectifs d'un attaquant sont divers : vols d'informations, terrorisme, espionnage, chantage, avantages concurrentiels et bénéfices financiers, attirer l'attention ou vérification de l'insécurité d'un système.

Pour faire face aux différentes attaques web, on fait appel aux systèmes de détection d'intrusion. Ce sont des outils ayant pour objectif de détecter des activités malveillantes sur la cible qu'ils surveillent. Une alerte sera déclenchée dès qu'un comportement malveillant est détecté. Les systèmes de détection d'intrusion sont utilisés en complément des solutions traditionnelles telles que les pare-feu, pour détecter différents types d'utilisation malveillante de leur cible qui ne peuvent être détectées par ces dernières. Pour cela, de nombreux paramètres doivent être pris en compte selon ce que l'on cherche à surveiller. En effet, le système de détection d'intrusion ne se placera pas au.

Même endroit dans l'architecture réseau, les systèmes de détection d'intrusion peuvent être placés en coupure du réseau ou sur un hôte. La temporalité de l'analyse est également un paramètre important, pouvant être réalisée en temps réel ou à posteriori. Ces systèmes se basent sur l'écriture de règles de filtrage par les utilisateurs pour effectuer leurs analyses.

On classe généralement les systèmes de détection d'intrusion en deux catégories : la détection par signatures et la détection par anomalies.

Les systèmes de détection d'intrusion par signatures reposent sur des bibliothèques de description des attaques (signatures). Lors de l'analyse du flux réseau, le système examine chaque événement et déclenche une alerte en cas de détection d'une signature correspondante. Cette approche est efficace si la base de signatures est régulièrement mise à jour, mais elle peut être inefficace face à des attaques inconnues. Elle dépend donc de l'environnement et peut être limitée.

Les systèmes de détection d'intrusion par anomalies, quant à eux, ne se basent pas sur des bibliothèques de description des attaques. Ils cherchent à détecter des comportements anormaux lors de l'analyse du flux réseau. Ces systèmes passent par deux phases : une phase d'apprentissage où ils étudient les comportements normaux du flux réseau, et une phase de détection où ils analysent le trafic pour identifier les événements anormaux en se basant sur leurs connaissances. Cette approche, qui utilise des techniques d'apprentissage supervisé telles que les réseaux de neurones artificiels, les modèles de Markov cachés ou les machines à vecteurs de support, est la plus efficace.

Chapitre 1 : Etude préalable

I. Présentation

Notre projet est un pare-feu applicatif Web efficace (Web Application Firewall ou WAF) qui est spécifiquement conçue pour détecter les attaques par injection SQL. Les attaques par injection SQL sont une menace majeure pour les applications Web, et notre WAF offre une protection robuste en identifiant et en bloquant ces attaques malveillantes. On va vous expliquer comment notre WAF fonctionne, ses principales caractéristiques et les avantages qu'il offre pour la sécurité de vos applications Web.

II. Objectifs

Le pare-feu applicatif Web (Web Application Firewall) a pour objectif principal d'analyser les vulnérabilités de sécurité liées à l'injection SQL dans une application web, et d'évaluer l'efficacité d'un pare-feu d'application web (WAF) dans la prévention de ces attaques.

Voici nos objectifs :

- Comprendre les vulnérabilités de l'injection SQL : Il est important de bien comprendre comment fonctionnent les attaques par injection SQL et les différentes techniques utilisées par les attaquants.
- Identifier les différentes formes d'injection SQL : Il existe de nombreuses variantes d'injections SQL, telles que les attaques basées sur les chaînes de caractères, les commentaires SQL, les opérations booléennes, etc. Identifiez-les et comprenez comment elles peuvent être utilisées pour exploiter les failles de sécurité.
- Mettre en place des règles de sécurité : Établissez des règles de sécurité pour détecter et bloquer les tentatives d'injection SQL. Cela peut inclure l'identification des motifs ou des séquences de caractères associés à une attaque par injection SQL et la mise en place de mesures pour les contrer.
- Effectuer des tests de validation : Testez votre WAF en utilisant des scénarios d'attaque prédéfinis et des données d'injection SQL connues pour vérifier son efficacité. Assurez-vous qu'il bloque les attaques et ne génère pas de faux positifs.

Chapitre 2 :Contexte **général du projet**

I. Introduction

Ce chapitre a pour objectif de présenter le contexte général du projet. Il commence d'abord par présenter le cadre général du sujet, pour enchaîner avec la problématique et nos objectifs du projet, et finir par une description du déroulement du projet et le processus de développement adopté ainsi que la planification du sujet.

II. Présentation du projet

1. Problématique

Les attaques web ont vu le jour parallèlement avec le web et ne datent pas d'hier. Cependant, elles continuent de s'avérer dangereuses et menaçantes pour notre sécurité en tant qu'internautes.

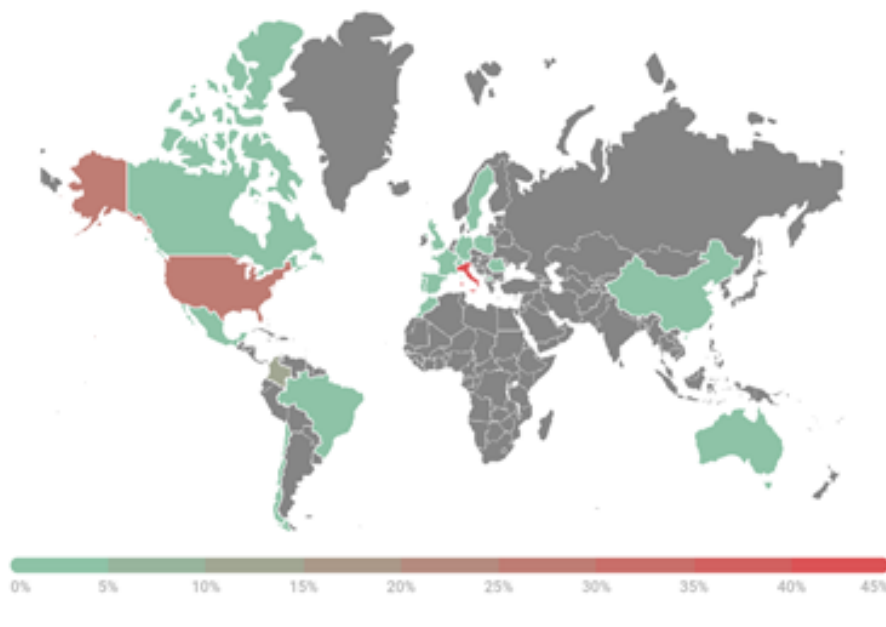


Figure 1 : Les attaques web détectées par Kaspersky en avril 2022

Ainsi, les solutions traditionnelles se basent sur l'analyse statique du trafic entrant. Cette solution inclut tout d'abord la création d'une signature qui décrit les caractéristiques de l'attaque que l'on veut détecter, et lors de la détection de la signature, le trafic entrant sera bloqué par le FIREWALL ou par une interface de sécurité. Ces solutions traditionnelles ont l'avantage d'être plus rapides et peuvent être implémentées en temps réel pour protéger les ressources réseau, mais elles restent incapables face aux attaques inconnues, car elles nécessitent un travail préalable pour définir le malicieux du bénin.

2. Solution envisagée

Pour lutter contre le développement des attaques web et être en mesure de détecter des attaques même si elles sont inconnues, nous proposons dans ce projet d'utiliser des algorithmes de Machine Learning pour analyser le trafic réseau et distinguer le bénin du malicieux.

3. Contraintes

L'utilisation des techniques du Machine Learning pour la détection des attaques web est efficace contre le développement continu des attaques, mais cela pose une panoplie de contraintes auxquelles nous devons faire face lors de l'application de la solution envisagée. Premièrement, l'emploi des techniques du Machine Learning nécessite une certaine performance du système, d'où la nécessité de choisir des algorithmes capables d'atteindre une performance système satisfaisante. Deuxièmement, les recherches dans le domaine de la détection des attaques web affirment que l'efficacité de notre solution repose sur l'existence de bons ensembles de données (data sets), d'où la nécessité de mettre en place une architecture réseau permettant une bonne capture de trafic ou d'emprunter des ensembles de données provenant de sources crédibles.

4. Planning prévisionnel du déroulement du Projet

La figure ci-dessous illustre les différentes étapes que nous avons suivies tout au long de la réalisation de ce projet. La première étape consiste en une étude sur les applications web en général et leurs vulnérabilités en particulier, ainsi que sur les différents types d'attaques web. Ensuite, il serait judicieux de consacrer du temps à l'apprentissage du langage Python, qui est indispensable pour passer à l'étape suivante, à savoir l'étude des différents algorithmes et techniques du Machine Learning. Enfin, l'étape ultime serait la conception et la réalisation de notre système de détection des attaques web basé sur les techniques du Machine Learning.

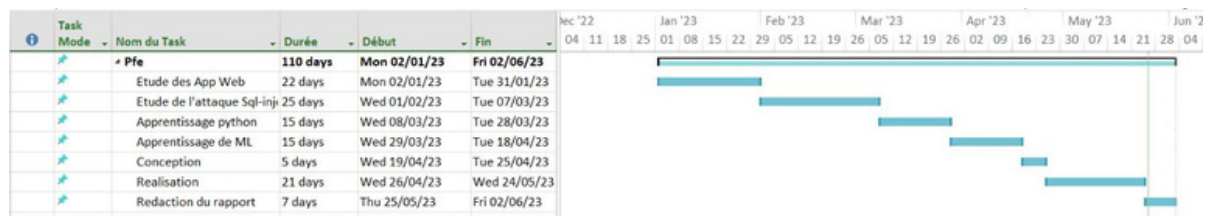


Figure 2 : Diagramme de GANTT

5. Conclusion

Dans cette partie, nous avons présenté le sujet de notre travail dans son contexte général afin de pouvoir l'aborder de la meilleure manière possible et d'assurer une maîtrise totale de l'objectif visé.

Chapitre 3 :Le Web

I. Introduction

Dans ce chapitre, nous présentons une introduction à l'architecture et au fonctionnement des applications Web. Ainsi, nous étudions et analysons les différents problèmes de sécurité auxquels font face ces applications en présentant leurs vulnérabilités et les attaques qui les ciblent .

II. Applications web

Les évolutions des systèmes de télécommunications et de l'informatique ont permis l'émergence de nouvelles technologies pour répondre aux besoins grandissants en termes de connectivité et de partage de données. Vers la fin des années 80s, un physicien du CERN, Tim Berner-Lee, initia le projet World Wide Web (la toile d'araignée mondiale) après avoir déjà travaillé sur le projet précurseur du Web ENQUIRE depuis le début des années 80s. L'objectif du projet WWW ou W3 était de "fournir un système d'information collaboratif, indépendamment des plateformes matériels et logiciels, et de la localisation géographique. Le projet décrivait un paradigme englobant un fonctionnement distribué en mode client/serveur, du contenu en hypertexte et un protocole de transfert des documents hypertexte basé sur des URLS.

Le Web est un système distribué basé sur le modèle client/serveur ouvert, où les ressources se trouvent du côté du serveur et les utilisateurs pouvant les exploiter à travers des clients qu'on appelle navigateurs. La conception de ce modèle est basée sur le principe des documents en hypertexte interconnectés par des hyperliens. Pour la description de ces documents, il a fallu développer un nouveau langage qui structure la sémantique du contenu par un système de balisage. Le langage HTML (HyperText Markup Language), descendant directement du langage SGML, permet la création et la mise en forme des documents qu'on appelle pages Web. Ces pages sont échangées entre le client et le serveur par le biais d'un système de requête/réponse appelé protocole HTTP (HyperText Transfert Protocol).

2.1 Fonctionnement d'une application web

Les applications web sont de plus en plus utilisées, pour diverses causes et emplois qui facilitent nos vies.

Une application web est l'ensemble des pages web, des fichiers, des données et des services web situés sur le serveur web de compagnie.

Souvent , une application web est présentée en trois niveaux:

Un niveau de présentation (le navigateur web), un niveau logique (langage de programmation tel que PHP, JAVA, .NET, etc.). Le navigateur web (niveau présentation) envoie des demandes au niveau logique, qui les traite et formule des requêtes pour les envoyer au niveau stockage (base de données ou server).

Une application web est une application qui n'a besoin que du protocole HTTP pour être pilotée par un utilisateur. Celui-ci utilise un simple navigateur web ou bien une application propriétaire utilisant le protocole http.

Les données saisies par l'utilisateur sont envoyées à l'application web via les deux fonctions POST et GET.

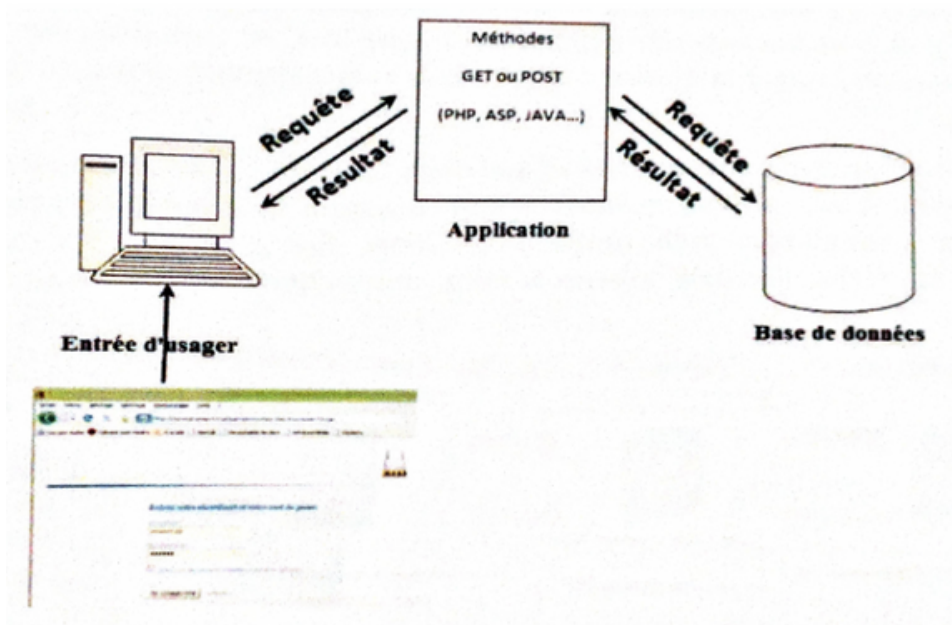


Figure 3 : Fonctionnement d'une application web

2.2 Architecture des applications web

Comme déjà mentionné, une application web qui utilise une base de données est composée souvent de trois tiers: présentation, logique, stockage. La figure ci-dessous les illustre.

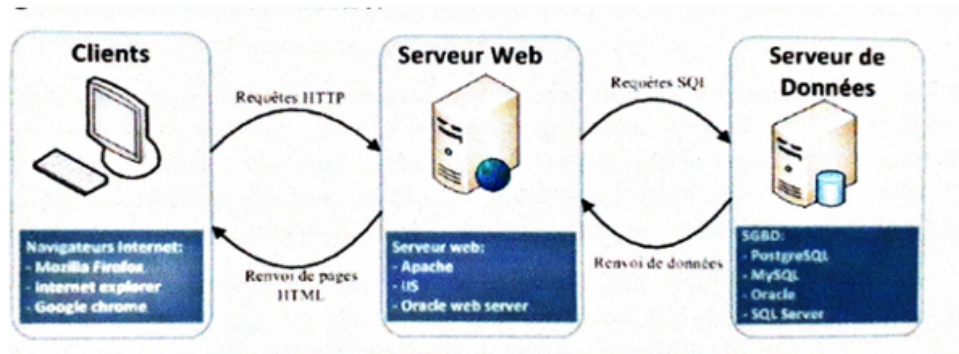


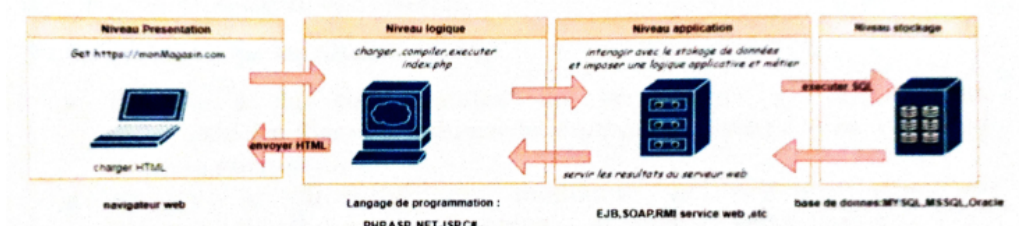
Figure 4 : Niveaux d'une application web

Le niveau présentation est le niveau supérieur de l'application. C'est dans celui-ci que les informations relatives au service voulu s'affichent, tel que les informations sur les identifiants d'un utilisateur, ses paramètres et préférences, etc.

Le niveau logique contrôle les fonctionnalités de l'application, pendant que le niveau data représente les serveurs de base de données, dans lesquelles les informations sont stockés et récupérés.

Dans la figure précédente, le navigateur envoie des requêtes au niveau logique, qui à son rôle les translate en requêtes exploitables par la base de données. La règle fondamentale donc de cette architecture c'est que le premier niveau (présentation) ne peut pas communiquer directement avec le niveau stockage et toute communication doit passer à travers le nœud intermédiaire (logique).

L'architecture à 3 niveaux présentée précédemment n'est pas évolutive, pour cela on a pensé à un n-niveau modèle. Dans ce modèle, une solution à 4 niveaux a été conçue, elle inclue l'utilisation d'un middleware, appelé généralement serveur d'application, entre le serveur Web et le serveur de base de données.



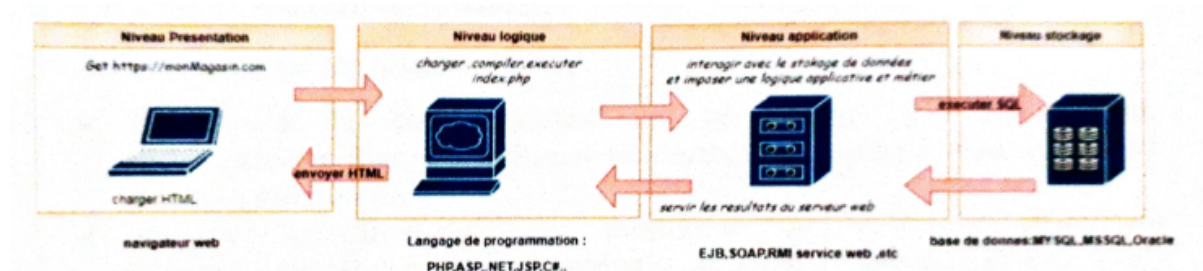


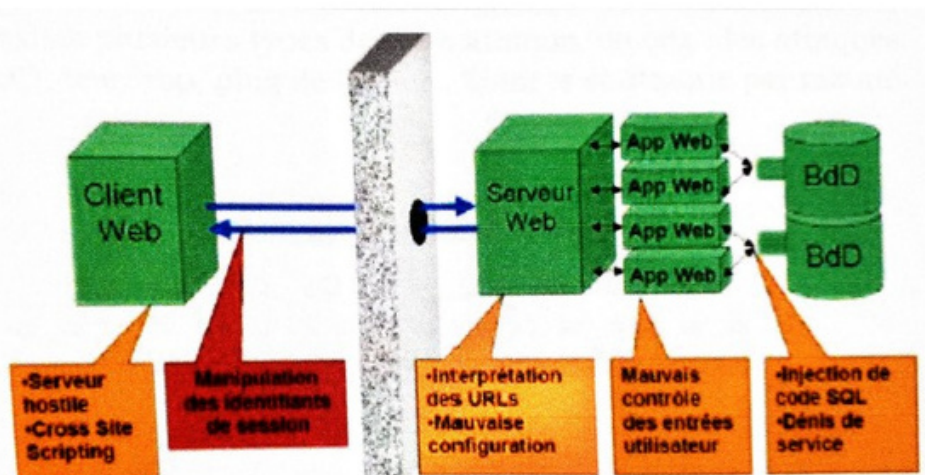
Figure 5 : Application à n-niveaux

Le serveur d'application dans une architecture à n-niveaux, est un serveur qui est responsable de l'hébergement d'un API (application programming interface) qui a pour rôle l'exposition du logique métier et les processus métier. Ce serveur est capable de communiquer avec les différentes sources, parmi eux, les bases de données, mainframes, etc.

Dans cette architecture, le niveau présentation (navigateur web) envoie des requêtes au niveau logique qui communique avec le serveur d'application à travers l'API public. Le principe général de cette architecture, c'est que chaque application web est divisée en blocs logiques ou couches, chacune a un rôle, et plus on utilise de niveaux, plus le rôle de chaque niveau est plus spécifique.

2.3 Vulnérabilités des applications web

L'avantage majeur d'une application web est qu'elle est utilisable à distance sans la nécessité d'installer ni de déployer aucun logiciel sur le poste client. Les visiteurs des applications web sont nombreux. Avec cette variété des utilisateurs et ses différents comportements, il est impossible de distinguer entre légitimes ou non. Les visiteurs inconnus peuvent représenter un risque pour les applications web et donc altérer les informations, la qualité des traitements et les opérations qui y sont conduites.



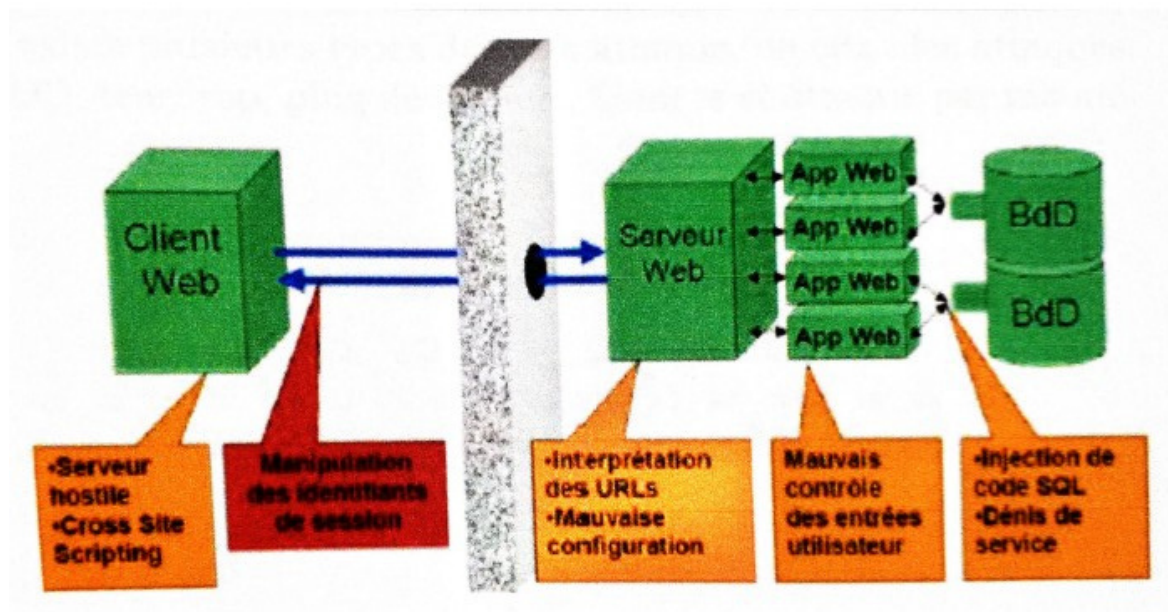


Figure 6 : Vulnérabilités d'une application web

Les différents points d'attaques possibles sur les applications web sont illustrés dans la figure ci-dessous :

Parmi les objectifs des attaques :

- Atteinte à la confidentialité en récupérant des informations confidentielles telles que des secrets industriels, des annonces commerciales, des résultats ou des données clients.
- Atteinte à l'intégrité en modifiant le contenu, entraînant des résultats incorrects ou incohérents, des comptes falsifiés, des défigurations ou des bases de données corrompues.
- Atteinte à la disponibilité en rendant le service indisponible pour les utilisateurs légitimes pendant une durée temporaire ou permanente, souvent appelée attaque par déni de service (DoS). Cela peut entraîner des pertes financières ou bloquer la force de travail.

III. Les attaques web

1. Attaques par déni de service (DoS) et par déni de service distribué (DDoS)

Une attaque par déni de service submerge les ressources d'un système de manière à ce qu'il ne puisse pas répondre aux demandes de service.

Une attaque DDoS vise également les ressources d'un système, mais elle est lancée à partir d'un grand nombre d'autres machines hôtes infectées par un logiciel malveillant contrôlé par l'attaquant.

Contrairement aux attaques conçues pour permettre à un attaquant d'obtenir ou de faciliter l'accès, le déni de service ne procure pas d'avantage direct aux attaquants. Il s'agit plutôt d'une satisfaction personnelle pour les pirates. Cependant, son objectif peut être de mettre un système hors ligne afin de lancer un autre type d'attaque, notamment le détournement de session.

Il existe plusieurs types d'attaques de déni de service (DoS), notamment les attaques SYN Flood, Teardrop, Ping de la mort, les botnets et les attaques par rebond. La figure 6 illustre ces attaques DoS et DDoS.

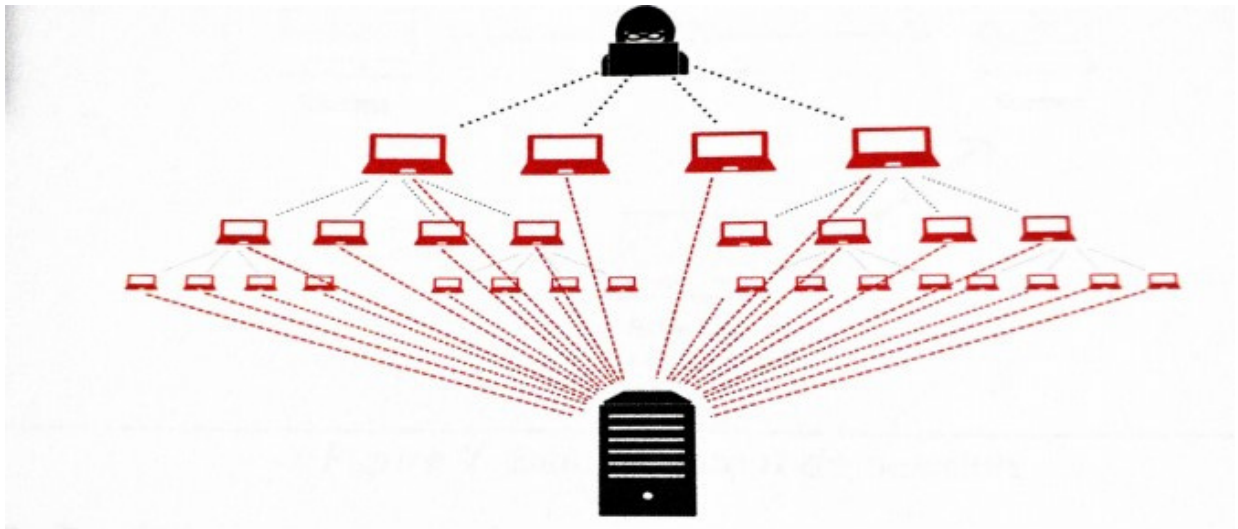


Figure 7 : Attaques DoS et DDoS

2. Attaque de l'homme du milieu (MitM)

Une attaque de l'homme du milieu consiste en l'interception des communications entre un client et un serveur par un pirate. Parmi les types courants d'attaques de l'homme du milieu, on trouve :

- **Détournement de session** : L'attaquant intercepte une session entre un client de confiance et un serveur réseau. L'ordinateur de l'attaquant substitue son adresse IP à celle du client de confiance, tandis que le serveur continue de poursuivre la session, croyant qu'il communique avec le client. Le déroulement de cette attaque peut être le suivant :
 - a. Un client se connecte à un serveur.
 - b. L'ordinateur de l'attaquant prend le contrôle du client.
 - c. L'ordinateur de l'attaquant déconnecte le client du serveur.
 - d. L'ordinateur de l'attaquant remplace l'adresse IP du client par sa propre adresse IP et son propre nom de domaine, et usurpe les numéros de séquence du client.
 - e. L'ordinateur de l'attaquant continue le dialogue avec le serveur, qui pense toujours communiquer avec le client.

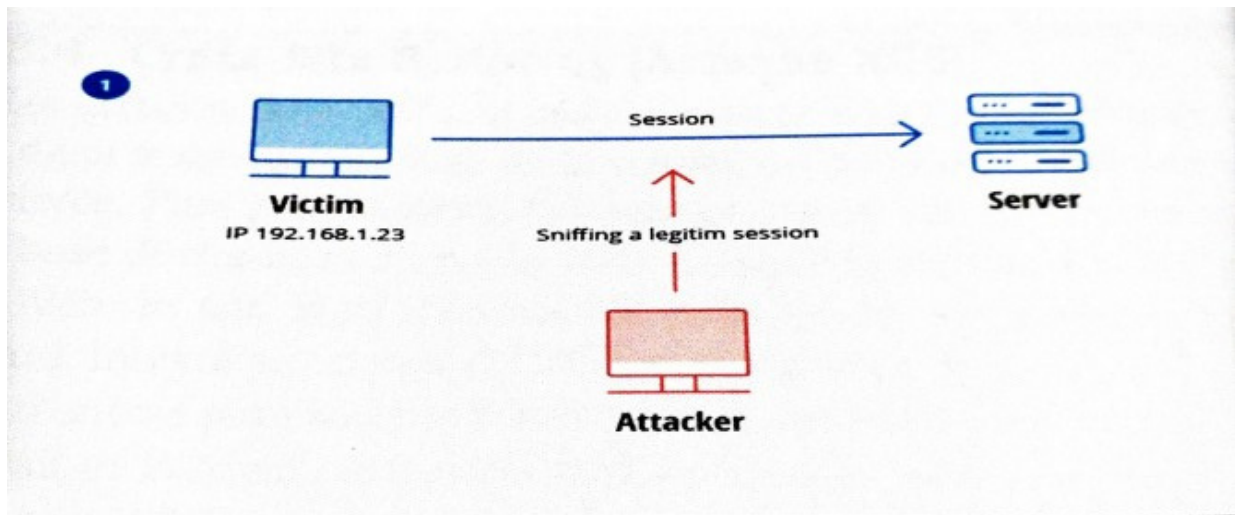


Figure 8 : Détournement de session

3. Attaque par injection SQL

L'injection SQL est devenue un problème courant qui affecte les sites Web exploitant des bases de données. Elle se produit lorsqu'un attaquant exécute une requête SQL sur la base de données en utilisant les données entrantes du client vers le serveur. Des commandes SQL sont insérées dans la saisie des données, par exemple, à la place du nom d'utilisateur ou du mot de passe, dans le but d'exécuter des commandes SQL prédéfinies. Une attaque d'injection SQL réussie peut permettre la lecture de données sensibles de la base de données, la modification (insertion, mise à jour ou suppression) des données de la base de données, l'exécution d'opérations d'administration de la base de données (par exemple, la fermeture de la base de données), la récupération du contenu d'un fichier spécifique, et, dans certains cas, l'envoi de commandes au système d'exploitation.

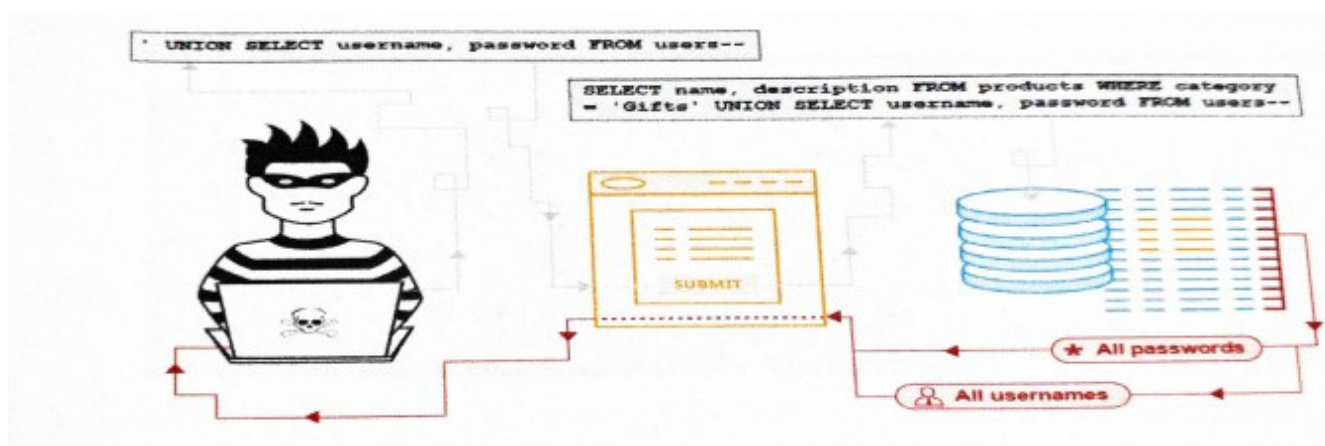


Figure 9 : Injection SQL

4. Cross-Site Scripting (Attaque XSS)

Les attaques XSS utilisent des ressources Web tierces pour exécuter des scripts dans le navigateur Web de la victime ou dans une application pouvant être scriptée. Plus précisément, l'attaquant injecte un JavaScript malveillant dans la base de données d'un site Web. Lorsque la victime demande une page du site Web, le site Web transmet la page à son navigateur avec le script malveillant intégré dans le corps HTML. Le navigateur de la victime exécute ce script, qui peut par exemple envoyer le cookie de la victime au serveur de l'attaquant, qui l'extraît et l'utilise pour détourner la session. Les conséquences les plus graves se produisent lorsque XSS est utilisé pour exploiter des vulnérabilités supplémentaires. Ces vulnérabilités peuvent non seulement permettre à un attaquant de voler des cookies, mais aussi d'enregistrer les frappes de touches et des captures d'écran, de découvrir et de collecter des informations sensibles, et d'accéder et de contrôler à distance l'ordinateur de la victime.

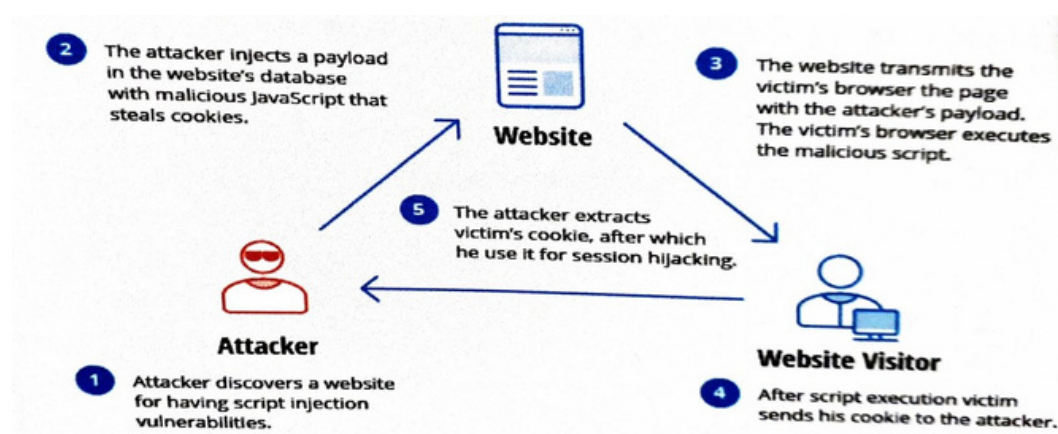


Figure 10 : Attaque XSS

5 Attaque Brute-Force

L'attaque brute-force est une méthode d'essais et d'erreurs utilisée par les pirates pour deviner des informations d'identification ou des données cryptées, telles que des informations de connexion, des mots de passe ou des clés de cryptage, en utilisant un effort exhaustif (en utilisant la force brute) dans l'espoir de trouver éventuellement la bonne combinaison.

```
[80][http-get-form] host: 192.168.100.155 login: admin password: password
[80][http-get-form] host: 192.168.100.155 login: admin password: password
[80][http-get-form] host: 192.168.100.155 login: admin password: 12345
[80][http-get-form] host: 192.168.100.155 login: admin password: 1234567890
[80][http-get-form] host: 192.168.100.155 login: admin password: Password
[80][http-get-form] host: 192.168.100.155 login: admin password: 123456
[80][http-get-form] host: 192.168.100.155 login: admin password: 1234567
[80][http-get-form] host: 192.168.100.155 login: admin password: 12345678
[80][http-get-form] host: 192.168.100.155 login: admin password: 1q2w3e4r
[80][http-get-form] host: 192.168.100.155 login: admin password: 123
[80][http-get-form] host: 192.168.100.155 login: admin password: 1
[80][http-get-form] host: 192.168.100.155 login: admin password: 12
1 of 1 target successfully completed, 12 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-07-27 15:28:24
```



```

[80][http-get-form] host: 192.168.100.155 login: admin password: password
[80][http-get-form] host: 192.168.100.155 login: admin password: p@ssword
[80][http-get-form] host: 192.168.100.155 login: admin password: 12345
[80][http-get-form] host: 192.168.100.155 login: admin password: 1234567890
[80][http-get-form] host: 192.168.100.155 login: admin password: Password
[80][http-get-form] host: 192.168.100.155 login: admin password: 123456
[80][http-get-form] host: 192.168.100.155 login: admin password: 1234567
[80][http-get-form] host: 192.168.100.155 login: admin password: 12345678
[80][http-get-form] host: 192.168.100.155 login: admin password: 1q2w3e4r
[80][http-get-form] host: 192.168.100.155 login: admin password: 123
[80][http-get-form] host: 192.168.100.155 login: admin password: 12
1 of 1 target successfully completed, 12 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-07-27 15:28:24

```

Figure 11 : Attaque Brute Force

IV- Solutions de filtrage Web et problèmes de contournement des signatures

Dans le contexte de la détection des attaques sur les applications Web, les modèles de sécurité actuels recommandent le déploiement en amont d'un pare-feu d'application Web (Web Application Firewall - WAF) en mode coupe-feu devant le serveur Web hébergeant ces applications. Cela se traduit, dans la plupart des cas, par l'intégration d'un module dans le serveur Web lui-même ou, mieux encore, sur un serveur mandataire inversé (reverse-proxy) fonctionnalités, mais cela peut également rendre leur configuration complexe. Les administrateurs doivent prendre des décisions difficiles pour optimiser les performances du WAF tout en assurant une protection adéquate contre les attaques.

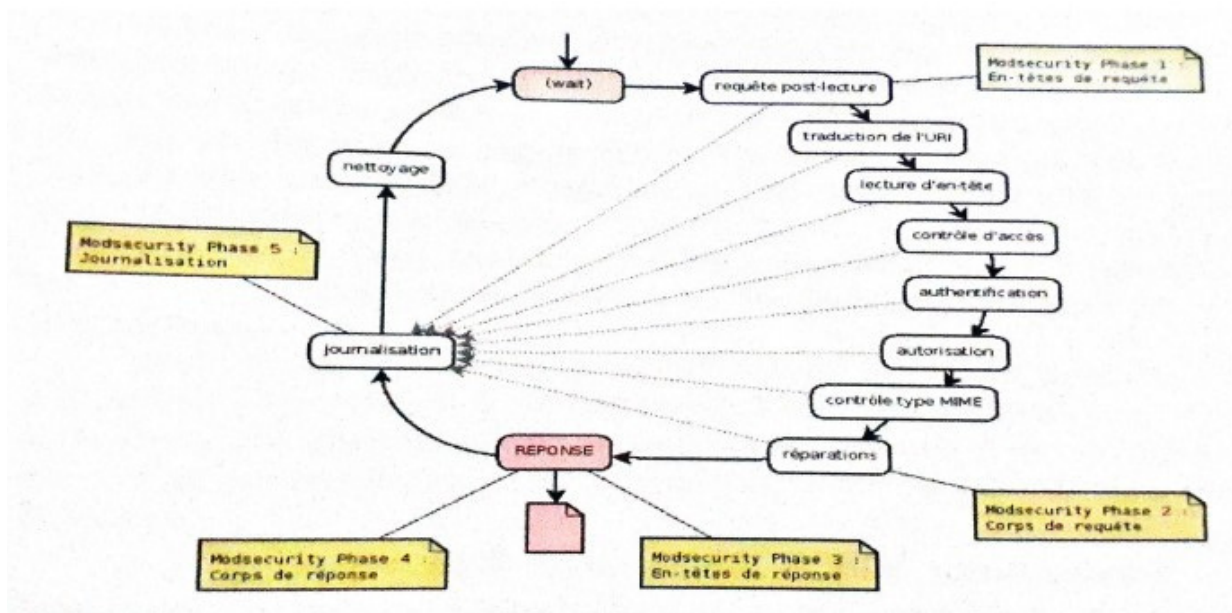


Figure 12 : Phases d'analyse et de log de ModSecurity source

Ce mode de fonctionnement permet un découpage des règles de sécurité en quatre parties. Dans la première phase, le serveur Web traite les en-têtes de la requête et passe la main à l'IPS proxy pour analyser l'ensemble des en-têtes comme étant une seule chaîne de caractères. Cette dernière charge uniquement les règles déclarées sur cette phase. Si aucune règle ne correspond au contenu analysé, alors l'IPS proxy repasse la main au serveur Web pour la suite de la requête.

Dans le cas où une seule règle a déclenché une correspondance, l'IPS proxy informe le serveur Web avec une décision adéquate et enregistre l'événement. Le serveur Web doit suivre la décision envoyée par son IPS proxy. Le traitement des autres phases est identique à la première phase. Les règles de sécurité de chaque phase sont chargées en mémoire pour des raisons de performance de fonctionnement. Cependant, si le nombre de règles devient trop important, les performances (mémoire et CPU) se dégradent rapidement.

Pour éviter la surcharge, les administrateurs sont confrontés à des choix difficiles. Ils peuvent ignorer certaines phases, comme les phases 3 et 4, ou réduire le nombre de règles de sécurité de chaque phase, au risque de laisser passer des attaques à travers l'IPS proxy. Ce problème est lié au fait que le serveur Web ne permet pas un découpage fin de la requête et de la réponse. La requête est considérée comme une entité à deux blocs : les en-têtes et le corps. L'IPS proxy est donc contraint d'analyser chaque bloc sans considération pour la sémantique intrinsèque aux en-têtes et aux autres composantes du corps.

Un deuxième problème est lié à l'expression des règles de sécurité. Les langages des systèmes de filtrage applicatif sont certes riches en syntaxe et en fonctionnalités, mais ils nécessitent une expertise et une précision dans l'expression des attaques. Malheureusement, la richesse de la sémantique des systèmes d'informations à protéger d'un côté, et la sophistication des outils utilisés par les attaquants, rendent la maîtrise du processus d'élaboration et de maintien des règles de sécurité une tâche difficile et complexe.

V- Techniques furtives d'évasion aux systèmes de filtrage

Dans le domaine de la détection des attaques sur les applications Web, il existe des outils capables de détecter les systèmes de défense installés en frontal des applications. Des outils tels que WAFulz, SqlMap et nmap sont en mesure de repérer l'empreinte du pare-feu d'application Web (WAF), ainsi que de déterminer si un serveur mandataire inversé (reverse-proxy) et un système d'équilibrage de charges sont présents. Ces outils offrent aux attaquants une visibilité sur les approches à adopter pour contourner ces systèmes de défense.

Dans cette section, nous abordons les attaques de type SQLI et les techniques utilisées par les attaquants pour obfusquer et masquer du code SQL malveillant. De nombreux systèmes de détection des attaques basés sur des signatures utilisent des algorithmes de recherche de motifs (Pattern-Matching). Pour échapper à ces algorithmes, les attaquants recourent à des techniques d'encodage qui modifient complètement la chaîne de caractères du code malveillant.

1- Variation de la casse(MAJUSCULE - minuscule)

Cette technique est la plus basique des techniques d'échappement, mais elle peut être utilisée dans certains filtres qui sont sensibles à la casse, ce qui n'est pas le cas du langage SQL. Des mots clés tels que UNION et SELECT peuvent s'écrire sous plusieurs formes:



```
# Union ou uNiOn, SeLeCT ou sELEcT,...etc.
```

Figure 13 : Variation de la casse

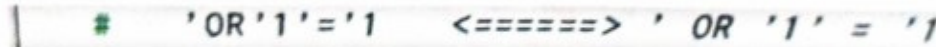
2- Espacement

La flexibilité du langage SQL en termes d'utilisation des espaces entre les opérateurs, ainsi que l'interprétation de certains caractères spéciaux tels que le retour-chariot (CR), retour à la ligne (LF) et la tabulation comme étant des espaces, offre aux attaquants une multitude de combinaisons possibles pour faire varier la forme du schéma d'attaque.



```
# ' OR '1' = '1' <=====> ' OR '1' = '1'
```

Figure 14 : Utilisation de la tabulation

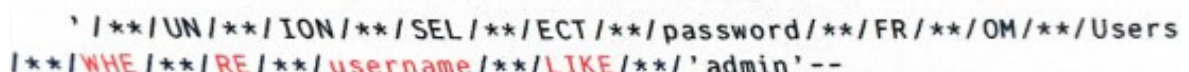


```
# 'OR'1'='1' <=====> ' OR '1' = '1'
```

Figure 15 : Supression des espaces

3- Commentaires

Dans le langage SQL, les commentaires sont délimités par /* et */. Tout ce qui se trouve entre ces balises de commentaire sera éliminé par l'interpréteur du langage. Ainsi, si la chaîne SEL/"texte"/ECT est passée à SQL, ce dernier l'interprétera comme SELECT en fusionnant les parties situées avant et après les balises de commentaire.



```
' /*UN/*ION/*SEL/*ECT/*password/*FR/*OM/*Users  
/*/WHE/*/RE/*/username/*/LIKE/*/'admin'--
```

Figure 16 : Commentaire pour déguiser la requete SQL

VI- Conclusion

Dans ce chapitre, nous avons exposé les problèmes auxquels les applications Web font face à travers quelques exemples d'attaques. Nous avons démontré que les attaquants manipulent les entrées des applications Web pour y sur la sécurité de l'application Web du côté serveur, mais aussi sur la sécurité des utilisateurs de cette application. Nous avons aussi présenté une solution de filtrage des attaques basée sur les signatures d'attaques. Cette solution est représentative des problèmes d'ordonnancement des règles de sécurité qui ne se base pas la sémantique applicative, mais sur un simple regroupement relatif au cycle de fonctionnement des serveurs Web. Ensuite, nous avons démontré par l'exemple que cette solution est vulnérable à des techniques de changement de forme ou d'évasion utilisées par les attaquants pour contourner les signatures déployées par les administrateurs.

Chapitre 4 : Machine learning

I- Introduction

Dans ce chapitre, nous allons présenter le Machine Learning en spécifiant les types d'apprentissage qu'il utilise, les techniques et les algorithmes offerts pour résoudre les différents problèmes, ainsi que des métriques d'évaluation des performances qui facilite le choix du modèle le plus adéquat.

II- Techniques d'apprentissage automatique

Machine Learning est une technique de découverte des modèles d'une façon automatique. Elle est définie aussi comme la science qui étudie les algorithmes informatiques qui s'améliorent avec l'expérience. Ces algorithmes construisent des modèles mathématiques en se basant sur des données d'entraînement. Ces modèles vont permettre aux algorithmes de faire des prédictions et décisions sans être programmés pour les faire. De nos jours, le Machine Learning est devenu l'une des branches informatiques les plus importantes et les plus demandées, et ses applications sont classées les plus utilisées dans tout secteurs d'activités.

Dans notre contexte, celui de la sécurité réseau, machine Learning est utilisé pour trouver et déterminer le moyen optimal de classer le trafic réseau en normal ou malveillant, ainsi de faire la prédiction des attaques.

Le développement d'un modèle Machine Learning repose sur quatre phases:

- Sélection et préparation d'un ensemble de données d'entraînement(Data Set).
- Sélection d'un algorithme à utiliser.
- Entraînement de l'algorithme.
- L'utilisation et l'amélioration du modèle.

III- Différents types d'apprentissage

1- L'apprentissage supervisé

L'apprentissage par supervision signifie qu'il y a toujours une personne qui va juger votre travail, de même, dans le domaine du Machine Learning, il signifie qu'on va fournir aux algorithmes pour réaliser des modèles doivent être étiquetées. Une donnée étiquetée signifie que chaque donnée d'apprentissage inclue une étiquette qui contient la bonne réponse que l'algorithme doit trouver. Par exemple, un ensemble de données étiquetées d'images d'animaux contient des étiquettes qui indiquent quel type d'animal est dans l'image. Le modèle compare chaque image aux données d'apprentissage afin de prédire d'une façon correcte l'étiquette de l'image.

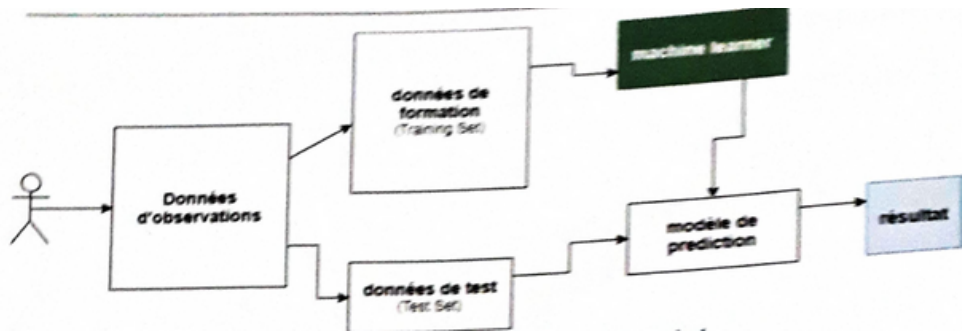


Figure 17 : L'apprentissage supervisé

L'apprentissage supervisé s'avère utile pour résoudre plusieurs problèmes, parmi eux ceux de classification et de régression.

Dans la figure suivante, les données d'apprentissage sont marquées par des cercles, et la ligne bleue représente le modèle parfait qui sera généré avec ces données-là, tandis que les autres lignes sont des modèles réels qui ont été générés avec les mêmes données d'apprentissage,

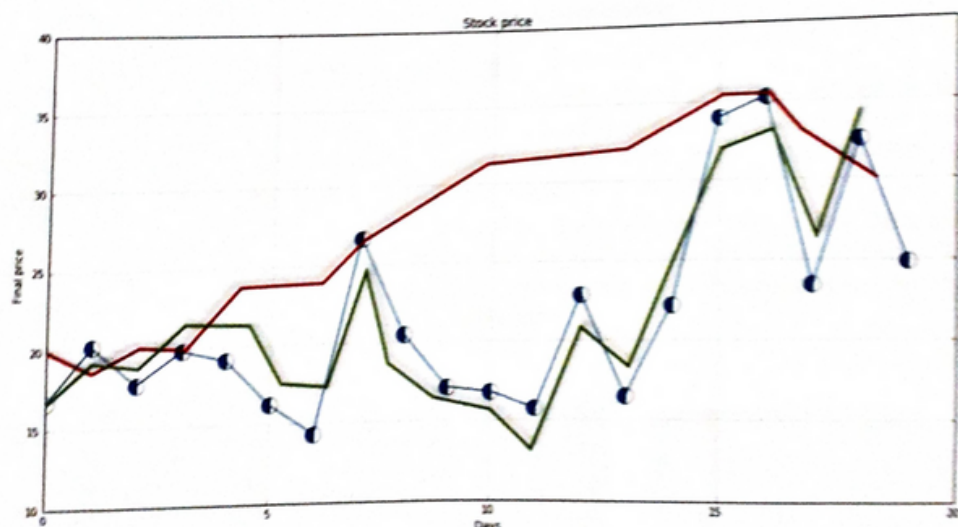


Figure 18 : Cas de régression

Dans la figure ci-dessous, il s'agit d'une classification, car il a juste un nombre discret de résultats possibles appelées catégories.

Dans la classification, l'algorithme utilisé pour effectuer la prédiction d'une valeur et déterminer sa classe ou son groupe est appelé "algorithme de classification".

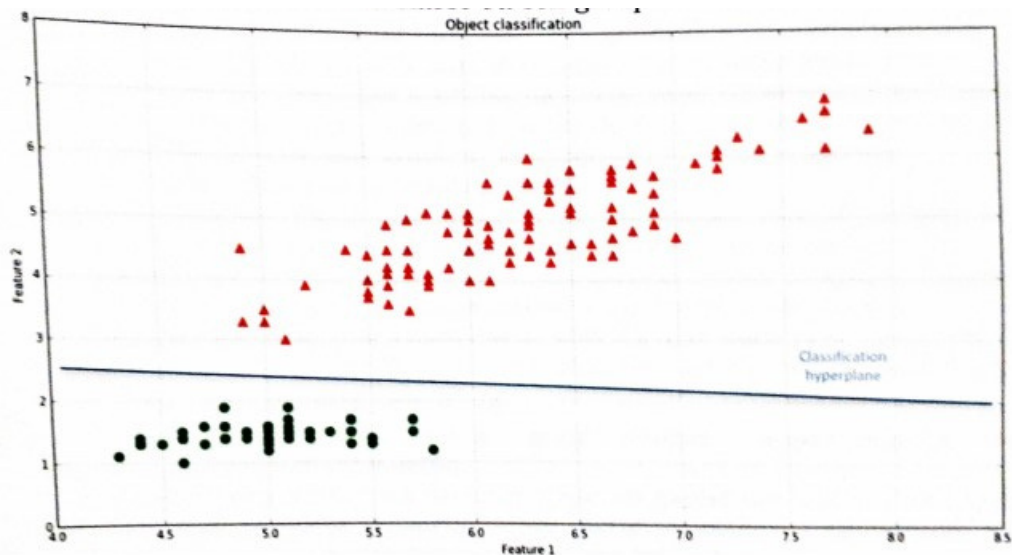


Figure 19 : L'apprentissage de classification

2- L'apprentissage insupervisé

Dans l'apprentissage supervisé, nous pouvons utiliser des données étiquetées, c'est-à-dire des données pour lesquelles les classes ou les catégories sont connues. Cependant, trouver de telles données étiquetées peut être difficile. C'est pourquoi nous avons recours à l'apprentissage non supervisé.

Dans l'apprentissage non supervisé, les algorithmes traitent des données non étiquetées, c'est-à-dire des données pour lesquelles les classes ne sont pas connues à l'avance. L'algorithme ne reçoit pas d'instructions spécifiques sur ce qu'il doit rechercher ou prédire dans ces données.

Un algorithme d'apprentissage non supervisé analyse les données pour découvrir des structures, des schémas ou des regroupements intrinsèques aux données, sans se baser sur des informations externes ou des étiquettes préexistantes. Il organise les données de différentes manières, par exemple en identifiant des groupes similaires ou en réduisant la dimensionnalité.

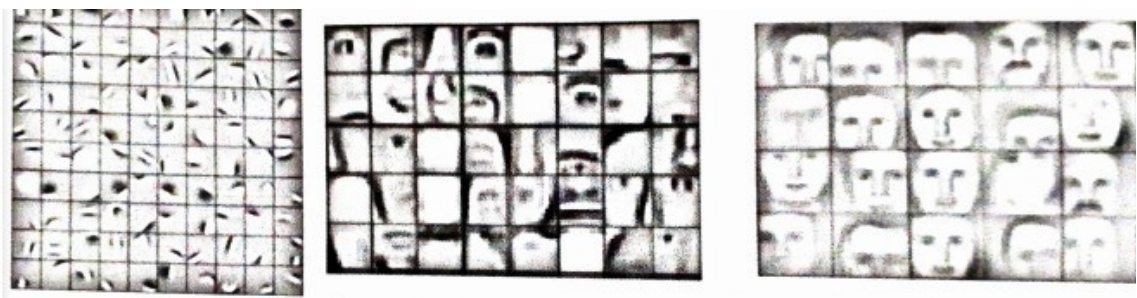


Figure 20 : Algorithmes utilisant l'apprentissage non supervisé

Clustering : Pour trier par exemple une collection de photos d'animaux par espèce, il n'est pas nécessaire d'être un expert dans le domaine, il suffit de s'appuyer sur des indices tels que la fourrure, la couleur, la forme, etc. C'est ainsi que fonctionnent les algorithmes qui utilisent l'apprentissage non supervisé. Le modèle cherche des données de formation similaires et les regroupe ensemble.

Détection d'anomalies : Les banques détectent généralement des actes de fraude en étudiant le comportement des clients. Par exemple, si un client utilise sa carte à Casablanca et à Bogota le même jour, cela suscite des soupçons. De la même manière, les algorithmes utilisant l'apprentissage non supervisé détectent des anomalies dans les données.

Association : Dans une bibliothèque en ligne, lorsque nous achetons un livre sur le Machine Learning, le site nous recommande automatiquement d'autres articles dans le même domaine. C'est un exemple d'association. L'algorithme utilisant l'apprentissage non supervisé prend les attributs clés d'une donnée et cherche d'autres données qui ont les mêmes attributs.

3- Apprentissage par renforcement

L'exemple le plus facile pour définir l'apprentissage par renforcement est les jeux vidéo.

Dans un jeu, lorsque nous terminons un niveau, nous gagnons des points, mais si nous tombons dans un piège, nous perdons la partie. Cela incite le joueur à développer ses compétences, à suivre des étapes et à éviter les actions aléatoires. L'apprentissage par renforcement fonctionne de la même manière. Lorsque l'agent (par exemple, un algorithme) suit des étapes pour atteindre un objectif, il reçoit une récompense.

Dans la figure suivante, un réseau neuronal a été développé pour jouer à un ancien jeu Atari. Les captures du jeu sont fournies en tant que données d'entrée à l'algorithme. Ces captures sont traitées par le réseau neuronal pour générer des règles à suivre dans le jeu. En appliquant ces règles, le jeu réagit en donnant une forme de récompense. Ce processus est répété jusqu'à ce que le résultat devienne stable, c'est-à-dire que l'algorithme choisit toujours les actions correctes et optimales dans le jeu.

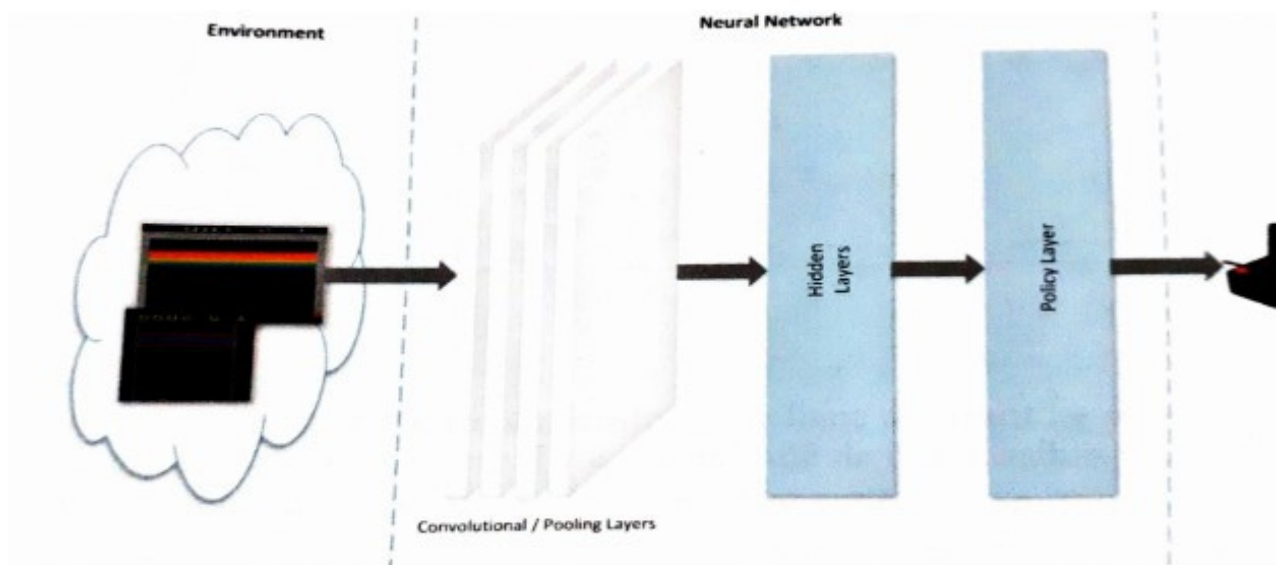


Figure 21 : Apprentissage par renforcement

IV- Arbres de décision

1- Principe général

Les arbres de décision sont une méthode récente et efficace pour explorer les données et prédire une variable qualitative en utilisant des variables de différents types (qualitatives et/ou quantitatives). Cette flexibilité constitue un avantage par rapport à certains outils de classification qui sont conçus pour des prédicteurs d'un seul type.

Il s'agit d'une méthode itérative appelée "partitionnement récursif des données". Elle construit des classes d'individus aussi homogènes que possible en posant une série de questions binaires (oui/non) sur les attributs de chaque individu.

Contrairement à de nombreux autres outils de classification tels que la régression logistique ou les machines à vecteurs de support (SVM), les arbres de décision sont extrêmement intuitifs et fournissent une représentation graphique claire et facile à lire du processus de classification des individus. Cette représentation graphique prend la forme d'un arbre avec des feuilles terminales représentant les classes d'individus, et en suivant un chemin à travers les nœuds de l'arbre, chaque nœud correspondant à une question binaire basée sur une variable du jeu de données.

Les arbres de décision permettent donc d'identifier rapidement les variables les plus discriminantes d'un jeu de données, en fonction de leur apparition éventuellement répétée le long des nœuds de l'arbre.

Une extension de cette méthode consiste à construire plusieurs arbres pour former une forêt aléatoire, ce qui améliore la capacité de discrimination [Bre01]. Cette approche n'est pas traitée dans ce document, mais le package R "randomForest" [LW02], bien documenté, permet de réaliser de telles analyses. Les arbres de décision et les forêts aléatoires sont largement utilisés en médecine [KTK08, MSR11] et constituent des outils très efficaces pour de nombreuses problématiques en anthropologie biologique [CHSD13, Lac13].

2- Exemple introductif

Un arbre de décision est un modèle analytique utilisé pour trouver la relation entre une variable d'intérêt qualitative Y et plusieurs prédicteurs X_1, X_2, \dots, X_p . Il peut servir à atteindre deux objectifs principaux :

1. Exploration : permet de comprendre la structure d'un jeu de données en identifiant les relations entre les différentes variables. Il permet également de déterminer les variables les plus discriminantes dans le jeu de données.
2. Prédiction : en utilisant les règles de décision générées par l'arbre, il est possible de prédire la valeur d'un nouvel individu (par exemple, s'il sera admis ou ajourné) en fonction des scores obtenus sur les variables prédictives X_1, X_2, \dots, X_p .

Dans cet exemple, nous disposons d'un jeu de données fictif qui recense les résultats (admis/ajourné) de 39 étudiants à l'issue d'une deuxième année de mathématiques dans une petite université. Les variables étudiées sont les suivantes :

- Sexe : le sexe de l'étudiant (F pour femme, M pour homme).
- Vit_Parents : variable binaire indiquant si l'étudiant vit chez ses parents.
- Taux_Presence : pourcentage de présence de l'étudiant aux cours magistraux (CM) et travaux dirigés (TD) au cours de l'année.
- Mention_Bac : variable ordinale, échelonnée de 0 (sans mention) à 3 (Très Bien), qui représente la mention obtenue par l'étudiant à son baccalauréat.
- Boursier : variable binaire indiquant si l'étudiant bénéficie d'une bourse sur critères sociaux.
- Salarié : variable binaire indiquant si l'étudiant a un emploi à temps partiel pendant ses jours de cours ou les week-ends.

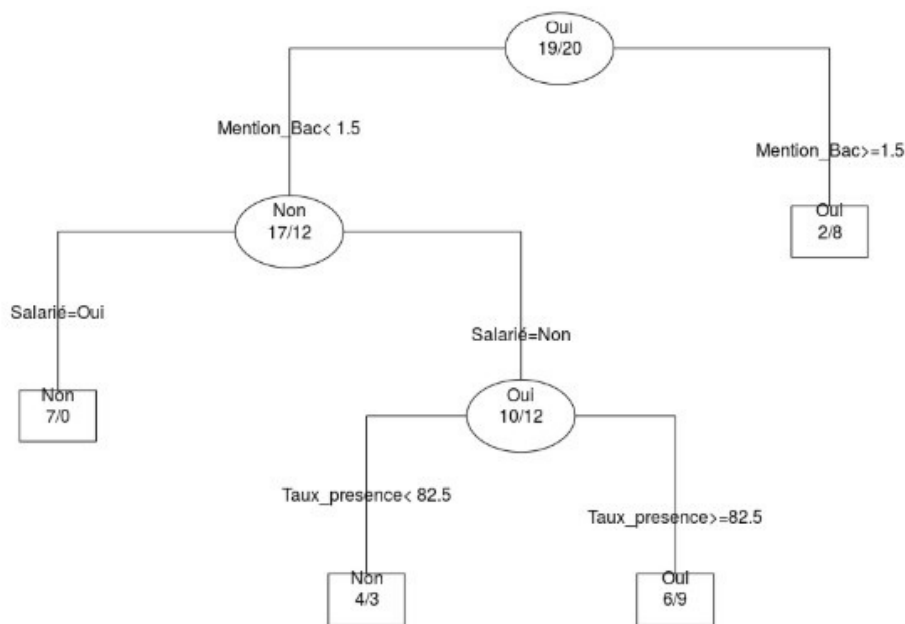


Figure 22 : Arbre de décision pour l'étude sur la réussite des étudiants

La lecture de l'arbre de décision est intuitive, car elle procède par une série de dichotomies à partir de la racine de l'arbre (située en haut du graphique). L'arbre fournit une série de règles de classification conditionnelles :

- Le critère le plus discriminant parmi les variables étudiées est la mention au baccalauréat : les étudiants ayant obtenu une bonne mention (Bien ou Très Bien) au lycée réussissent généralement leur deuxième année de mathématiques (L2).
- Pour les étudiants n'ayant pas obtenu une telle mention, le fait d'être salarié conduit à un échec en L2.
- Enfin, pour les étudiants sans bonne mention au baccalauréat et qui ne sont pas salariés, le critère discriminant est le taux de présence en cours.

Dans la figure 21, les feuilles terminales, représentant les classes d'affectation des individus, sont indiquées par des rectangles, tandis que la racine et les nœuds intermédiaires sont encadrés. La règle de décision associée à une feuille est le chemin parcouru pour atteindre cette feuille.

3- Données

Un arbre de décision construit avec l'algorithme CART peut traiter tous types de variables : qualitatives, ordinales et quantitatives continues. C'est l'une des grandes forces de cette méthode, car elle permet de créer des règles de décision combinant différents types d'informations, ce qui est particulièrement utile en anthropologie biologique. Par exemple, il est possible de prédire le sexe d'un individu en prenant en compte des mesures métriques, l'observation de caractères biologiques discrets ainsi que des relevés archéologiques indiquant la présence de mobilier.

Cette capacité à intégrer des données de différents types permet d'obtenir une vision plus complète et plus précise lors de la prise de décision, en prenant en compte diverses sources d'information.

4- Construction de l'arbre

4.1- Mesure de la pureté des feuilles

Lors de la construction d'un arbre de décision, le choix des questions les plus discriminantes pour créer les nœuds de l'arbre peut être effectué en utilisant différents critères. L'algorithme CART utilise l'indice de Gini, tandis que l'algorithme C4.5 utilise l'entropie probabiliste. Ces deux outils mathématiques visent à évaluer la "pureté" de chaque feuille de l'arbre.

L'entropie probabiliste, introduite par Claude Shannon en 1948, est une mesure du désordre dans une variable aléatoire discrète. Elle est utilisée pour évaluer la pureté d'une feuille dans un arbre de décision. L'entropie est maximale lorsque la variable est uniformément répartie, c'est-à-dire lorsque toutes les valeurs ont la même probabilité. En revanche, elle est minimale lorsque nous connaissons avec certitude la valeur de la variable. L'entropie est calculée en utilisant la formule $H(X) = - \sum p_i * \log_b(p_i)$, où p_i représente la probabilité de chaque valeur x_i de la variable et b est la base du logarithme (généralement 2).

L'indice de Gini est une mesure de la dispersion ou de l'inégalité d'une distribution. Il est souvent utilisé pour évaluer les inégalités sociales dans des domaines tels que l'économie ou la sociologie. Dans le contexte des arbres de décision, l'indice de Gini est utilisé pour évaluer la pureté d'une feuille en mesurant la probabilité qu'un élément choisi aléatoirement dans la feuille soit incorrectement classé. L'indice de Gini varie entre 0 (feuille pure) et 1 (feuille impure).

L'objectif de ces critères (entropie probabiliste et indice de Gini) est de créer des nœuds de l'arbre qui génèrent des feuilles aussi homogènes que possible, en séparant les données de manière à maximiser le gain d'information. Cela permet de construire un arbre de décision qui peut prendre en compte différents types d'informations et créer des règles de décision mixtes, en combinant des variables qualitatives, ordinales et quantitatives continues.

4.2- Algorithme de construction

L'algorithme de construction d'un arbre de décision peut être résumé comme suit:

1. Vérifier si la situation actuelle satisfait à un critère d'arrêt. Si oui, arrêter la construction de l'arbre.
2. Si aucun critère d'arrêt n'est satisfait, procéder comme suit:
 - a. Parmi toutes les variables disponibles dans le jeu de données, sélectionner celle qui permet de poser la question offrant le gain d'information maximal. Cela peut être déterminé en utilisant des mesures telles que l'entropie probabiliste ou l'indice de Gini.
 - b. Partitionner les individus du jeu de données en fonction de la question sélectionnée. Créer un nœud correspondant à cette question sur l'arbre.
 - c. Appliquer récursivement l'algorithme à chaque sous-ensemble de données résultant de la partition, en partant des nouveaux nœuds créés.

Ce processus est répété jusqu'à ce que les critères d'arrêt soient satisfaits, tels que le nombre minimal d'individus dans une feuille, le nombre maximal de feuilles autorisées ou lorsque la division supplémentaire n'améliore pas significativement la qualité discriminante de l'arbre.

L'objectif est de trouver un équilibre entre la complexité de l'arbre (pour éviter le surajustement) et sa capacité à représenter les modèles et les relations dans les données d'apprentissage. Une fois que l'arbre est construit, il peut être utilisé pour classer de nouveaux exemples en les faisant passer à travers les questions posées aux différents nœuds, jusqu'à atteindre une feuille qui correspond à une décision ou une prédiction.

5- Élagage de l'arbre

Pour éviter le sur-ajustement dans un arbre de décision et obtenir un modèle qui généralise bien, il est courant d'utiliser une technique appelée validation croisée pour élaguer l'arbre.

La validation croisée implique la division du jeu de données en sous-ensembles distincts, qui sont utilisés tour à tour comme ensemble d'apprentissage et ensemble de validation. Par exemple, si vous divisez le jeu de données en 3 sous-ensembles (appelés plis), vous pouvez suivre les étapes suivantes :

1. Utiliser les deux premiers plis comme ensemble d'apprentissage pour construire l'arbre de décision.

2. Calculer le taux d'erreur de prédiction en utilisant le troisième pli comme ensemble de validation.
3. Inverser les rôles des ensembles d'apprentissage et de validation. Les deux derniers plis sont utilisés comme ensemble d'apprentissage, tandis que le premier pli est utilisé comme ensemble de validation.
4. Calculer à nouveau le taux d'erreur de prédiction.
5. Répéter les étapes 1 à 4 en permutant les plis utilisés pour l'apprentissage et la validation à chaque itération.

Cela permet d'obtenir plusieurs estimations du taux d'erreur de prédiction pour différentes tailles de l'arbre (c'est-à-dire différents nombres de feuilles terminales). En comparant les taux d'erreur obtenus, vous pouvez identifier le niveau d'élagage de l'arbre qui donne la plus faible erreur de prédiction globale.

L'objectif est de trouver un compromis entre la complexité de l'arbre et sa capacité à généraliser correctement. Si l'arbre est trop complexe, il peut mémoriser des particularités spécifiques aux données d'apprentissage, ce qui entraîne une mauvaise généralisation. En élaguant l'arbre au niveau offrant l'erreur minimale, vous pouvez obtenir un modèle plus général et éviter le sur-ajustement.

6- Gestion des données manquantes

Lorsqu'il y a des données manquantes dans un jeu de données utilisé pour construire un arbre de décision, plusieurs problèmes se posent. Voici quelques solutions courantes pour gérer les données manquantes dans CART (Classification and Regression Trees) :

1. Suppression des individus : Si les valeurs manquantes sont rares et le nombre d'individus est important, une approche simple consiste à supprimer les individus qui ont au moins une valeur manquante. Cela permet de travailler avec un jeu de données complet, mais cela peut ne pas être approprié si le nombre d'individus est faible ou si les valeurs manquantes sont très fréquentes.
2. Utilisation de variables substituts : CART utilise une technique appelée "surrogate splits" ou "variables-substituts" pour contourner le problème des valeurs manquantes. Lorsqu'un split est effectué en utilisant une variable Z , il est possible qu'une autre variable $Z1$ puisse produire une partition similaire, c'est-à-dire les mêmes groupes. De même, une troisième variable $Z2$ peut produire une autre partition similaire. Ainsi, lorsque de nouveaux individus avec des valeurs manquantes pour la variable Z arrivent dans un nœud, ils sont envoyés à gauche ou à droite en fonction des valeurs qu'ils ont pour la variable substitut $Z1$ (ou $Z2$ si $Z1$ est également manquante chez eux).

3. Approches probabilistes : Il existe d'autres approches probabilistes pour traiter les données manquantes dans un arbre de décision. Ces approches font appel à des méthodes plus avancées et sont un sujet de recherche actif en statistique. Une référence pour explorer ces méthodes est l'article [Haw08].

Il est important de noter que le choix de la méthode de gestion des données manquantes dépend du contexte spécifique, de la fréquence des valeurs manquantes et de la disponibilité des données. Il est recommandé de choisir la méthode la plus appropriée en fonction de ces considérations.

V- Conclusion

En conclusion, nous avons examiné les différents types d'apprentissage, notamment l'apprentissage supervisé et l'apprentissage non supervisé. Nous nous sommes concentrés sur l'algorithme de CART, basé sur les arbres de décision, et avons utilisé la méthode des forêts aléatoires pour détecter les attaques web. Nous avons souligné l'importance des métriques de performance pour évaluer les modèles, telles que l'exactitude, la précision et le rappel. L'utilisation de ces techniques de Machine Learning nous permet d'améliorer la détection des attaques et d'obtenir des résultats plus précis.

Chapitre 5 : Conception et mise **en œuvre** **du système**

I- Expression des besoins

Notre équipe s'est engagée dans la conception et la mise en œuvre d'un système de détection des attaques web basé sur les techniques du Machine Learning. Conscients des risques d'attaques, notamment les attaques de type SQL Injection, auxquels les utilisateurs du web sont confrontés, notre objectif est de renforcer la sécurité en utilisant des méthodes avancées de détection. En analysant le trafic réseau, notre système est capable de distinguer et d'identifier les flux de données comme étant soit bénins, soit malveillants, en mettant notamment l'accent sur la détection des attaques de type SQL Injection.

Les attaques de type SQL Injection représentent une menace majeure pour la sécurité des applications web, car elles exploitent les vulnérabilités dans les requêtes SQL pour accéder, modifier ou supprimer des données sensibles dans les bases de données. En utilisant des techniques de Machine Learning, notre système peut apprendre à détecter les schémas et les comportements caractéristiques des attaques SQL Injection, tels que l'injection de code SQL malveillant dans les paramètres de requête. Cela permet une détection précoce et une réponse rapide pour protéger les applications et les données sensibles contre ce type d'attaque spécifique.

L'approche basée sur le Machine Learning offre ainsi une solution avancée et efficace pour la détection des attaques web, en particulier les attaques de type SQL Injection. Grâce à l'apprentissage des modèles et des comportements associés à ces attaques, notre système peut fournir une protection proactive et une défense solide contre les vulnérabilités exploitées par les attaquants.

II- Explication du système réalisé

Nous avons mis en place un système complet pour la détection des attaques web, qui se compose de deux notebooks distincts. Le premier notebook est dédié au processus d'entraînement et de test du modèle à l'aide de la bibliothèque PyCaret. Nous utilisons ce notebook pour charger le fichier CSV généré à partir des logs, qui contient les données nécessaires pour l'entraînement et le test du modèle. Le processus de génération du fichier CSV implique l'analyse des logs générés à l'aide d'outils tels que BurpSuite et Acunetix, leur extraction des informations pertinentes, puis leur structuration dans un format tabulaire avec des colonnes telles que la méthode, le chemin, le corps, les quotes simples, les quotes doubles, les tirets, les accolades, les espaces, les mots inappropriés et la classe (indiquant si la ligne est "bonne" ou "mauvaise").

Une fois le fichier CSV généré, nous utilisons PyCaret pour effectuer le processus d'entraînement du modèle. PyCaret automatise plusieurs étapes du pipeline de Machine Learning, telles que la préparation des données, le choix du modèle, l'optimisation des hyperparamètres et l'évaluation des performances. Dans notre cas, nous utilisons le modèle CART (Classification and Regression Trees) pour la détection des attaques SQL Injection. Le premier notebook nous permet également de visualiser les clusters générés à l'aide de l'algorithme de clustering K-means, ainsi que d'évaluer l'exactitude (accuracy) du modèle entraîné.

Le deuxième notebook de notre système contient le modèle entraîné et implémente un simple proxy pour le serveur local (localhost) à l'adresse IP 127.0.0.1. Ce proxy est conçu pour intercepter les requêtes HTTP qui sont dirigées vers les sites hébergés sur cette adresse IP locale. Il analyse les requêtes reçues et utilise le modèle entraîné pour détecter les attaques de type SQL Injection. Si une attaque est détectée, une alerte est générée pour avertir l'utilisateur. Ce proxy offre ainsi une protection en temps réel contre les attaques SQL Injection sur les sites hébergés localement, contribuant ainsi à renforcer la sécurité de notre système et à prévenir les vulnérabilités potentielles dans les applications web.

Chapitre 6 : Réalisation

I. Langages et outils du travail utilisé

1- Python :



Python est un langage de programmation flexible, puissant et largement utilisé dans les domaines de la cybersécurité et de l'intelligence artificielle. Sa simplicité et son écosystème riche en bibliothèques spécialisées en font un choix idéal pour les professionnels travaillant dans ces domaines, leur permettant d'automatiser des tâches, d'analyser des données et de développer des applications d'IA avancées.

2- Anaconda :



Anaconda est une distribution Python complète et conviviale, spécialement conçue pour les scientifiques des données, les chercheurs et les ingénieurs. Elle facilite l'installation, la gestion des packages et la création d'environnements de développement isolés, ce qui en fait un outil puissant pour travailler sur des projets de science des données et de calcul scientifique.

3- Jupyter notebook:



Jupyter Notebook est un environnement de développement interactif qui combine du code exécutable, des visualisations et du texte explicatif dans des documents appelés notebooks. Il permet d'exécuter du code, de créer des graphiques interactifs et de documenter les analyses de données de manière claire et reproductible. C'est un outil précieux pour les scientifiques des données, les développeurs et les chercheurs travaillant dans le domaine de la programmation et de l'analyse de données.

4- Burpsuite :



Burp Suite est une suite d'outils de test de sécurité des applications Web développée par PortSwigger. Elle est largement utilisée par les professionnels de la sécurité pour évaluer la vulnérabilité des applications web et effectuer des tests de pénétration. L'outil principal de Burp Suite est le "Proxy", qui agit comme un intermédiaire entre le navigateur web et le serveur cible, permettant de capturer, d'analyser et de modifier les requêtes et les réponses HTTP.

5- Acunetix :



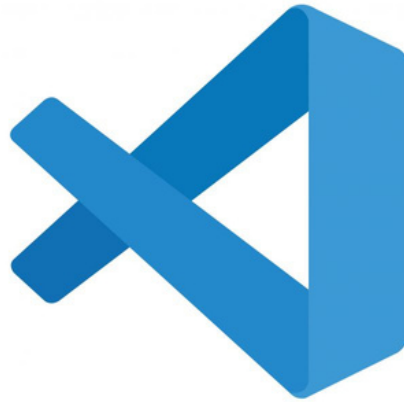
Acunetix est un outil puissant utilisé pour détecter et prévenir les vulnérabilités de sécurité dans les applications web, en particulier en ce qui concerne les injections SQL. L'injection SQL est une technique couramment utilisée par les pirates informatiques pour exploiter les failles de sécurité des applications web. Elle consiste à insérer du code SQL malveillant dans les entrées utilisateur, ce qui peut permettre aux attaquants de manipuler la base de données sous-jacente et d'accéder à des informations sensibles ou de prendre le contrôle de l'application.

6- CSVCleaner:



CSVcleaner, un outil puissant et facile à utiliser pour nettoyer et transformer les fichiers CSV. Que vous soyez un professionnel des données ou simplement quelqu'un qui travaille avec des fichiers CSV, CSVcleaner peut vous aider à simplifier et à optimiser votre travail.

7- Visual Studio Code:



Visual Studio Code est un puissant éditeur de code source, utilisé par de nombreux développeurs à travers le monde pour créer des applications dans divers langages de programmation. Sa polyvalence, ses fonctionnalités avancées et son écosystème d'extensions en font un outil populaire pour le développement logiciel.

8- PyCharm



PyCharm est un environnement de développement intégré (IDE) puissant et convivial conçu spécifiquement pour la programmation en Python. Il offre un large éventail de fonctionnalités qui facilitent la création, l'édition et le débogage de code Python. PyCharm dispose d'un éditeur de code avancé avec une coloration syntaxique, une complétion intelligente et une mise en forme automatique pour améliorer la productivité du développeur. L'IDE propose également des outils de débogage, de

profilage et de test qui permettent de détecter et de résoudre les erreurs plus facilement. PyCharm offre également une intégration avec des systèmes de contrôle de version tels que Git, ce qui facilite le travail collaboratif sur des projets. En résumé, PyCharm est un outil essentiel pour les développeurs Python, offrant une expérience de développement fluide, des fonctionnalités avancées et une productivité accrue.

9- Firefox



Firefox est un navigateur web populaire qui offre de nombreuses fonctionnalités, dont la possibilité de configurer un proxy intégré. Un proxy est un serveur intermédiaire qui agit comme un intermédiaire entre votre navigateur et Internet, vous permettant de masquer votre adresse IP réelle et de sécuriser votre connexion. En utilisant le paramètre de configuration de proxy intégré de Firefox, vous pouvez spécifier les détails du proxy, tels que l'adresse IP et le port, pour rediriger votre trafic Internet via ce serveur. Cela peut être utile pour contourner les restrictions géographiques, accéder à des sites web bloqués ou protéger votre vie privée en ligne. La configuration du proxy dans Firefox est simple et vous permet de personnaliser vos paramètres de navigation en fonction de vos besoins spécifiques. En utilisant cette fonctionnalité intégrée, vous pouvez bénéficier des avantages d'un proxy sans avoir à installer de logiciels supplémentaires.

II- Importation des bibliothèques et des modules nécessaires

```
from http.server import SimpleHTTPRequestHandler , HTTPServer
from urllib import request , error
import pandas as pd
from pycaret.clustering import *
import urllib.parse
import sys
```

Figure 23 : Bibliothèques utilisées

La bibliothèque **http.server** fournit des classes telles que `SimpleHTTPRequestHandler` et `HTTPServer`, qui sont utilisées pour créer des serveurs HTTP simples en Python. Elles vous permettent de gérer les requêtes HTTP entrantes et de fournir des réponses appropriées. Cela peut être utile lorsque vous souhaitez créer un serveur web léger pour servir des fichiers statiques ou répondre à des requêtes spécifiques.

Le module **urllib.request** fournit des fonctionnalités pour envoyer des requêtes HTTP vers des URL spécifiées. Il vous permet de récupérer des données à partir d'URL distantes en utilisant différentes méthodes telles que GET et POST. Il facilite également la gestion des cookies et des en-têtes HTTP. Le module `urllib.error` est utilisé pour gérer les erreurs lors de l'accès aux ressources distantes, telles que les erreurs de connexion ou les erreurs HTTP.

La bibliothèque **pandas** est largement utilisée pour la manipulation et l'analyse des données en Python. Elle offre des structures de données puissantes, notamment les `DataFrames`, qui permettent de manipuler facilement des données tabulaires. Avec `pandas`, vous pouvez effectuer des opérations de filtrage, de tri, de regroupement et de calcul statistique sur les données. Elle est souvent utilisée pour préparer et nettoyer les données avant l'analyse, ainsi que pour effectuer des tâches d'exploration et de visualisation des données.

La bibliothèque **pycaret** est un outil puissant pour le développement de modèles d'apprentissage automatique. Elle simplifie et accélère le processus de création de modèles en fournissant une interface conviviale pour les tâches courantes, telles que la préparation des données, la sélection des caractéristiques, la création et l'évaluation des modèles, ainsi que la comparaison des performances entre plusieurs modèles. `Pycaret` prend en charge un large éventail d'algorithmes d'apprentissage automatique et automatise de nombreuses étapes du flux de travail, ce qui en fait un outil précieux pour les professionnels du ML.

Le module **urllib.parse** est utilisé pour analyser et manipuler les URL. Il fournit des fonctions pour extraire des informations spécifiques à partir d'une URL, telles que le schéma, l'hôte, le chemin, les paramètres de requête, etc. Il permet également de coder et décoder les paramètres d'URL, ce qui est utile lors de la construction d'URL dynamiques ou de l'envoi de données via des requêtes GET ou POST.

Enfin, le module intégré `sys` fournit des fonctionnalités pour interagir avec l'environnement d'exécution du script Python. Il permet d'accéder à des variables et à des fonctions spécifiques au système, ainsi que de manipuler les arguments de ligne de commande passés au script. Il est couramment utilisé pour récupérer des informations sur le système d'exploitation, l'environnement Python ou les paramètres de ligne de commande, ce qui facilite la gestion du script en fonction du contexte d'exécution.

III- Datasets

Parler des ensembles de données est essentiel dans le domaine de la science des données et de l'apprentissage automatique. Un ensemble de données est un ensemble structuré de données qui représente une collection d'observations, d'informations ou de faits. Il peut être présenté sous différentes formes, telles que des tableaux, des fichiers CSV, des bases de données ou même des flux de données en temps réel. Les ensembles de données servent de base pour la modélisation et l'analyse statistique, permettant aux praticiens de tirer des informations précieuses et des insights exploitables. Ils jouent un rôle crucial dans le développement et l'évaluation de modèles d'apprentissage automatique, car la qualité et la représentativité des données peuvent avoir un impact significatif sur les résultats et les performances des modèles.

La préparation des données est une étape essentielle dans l'analyse des ensembles de données. Elle comprend des tâches telles que le nettoyage des données, le traitement des valeurs manquantes, la normalisation des données et la création de variables ou de caractéristiques supplémentaires pour améliorer la performance des modèles. Les ensembles de données peuvent varier considérablement en termes de taille, de complexité et de domaines d'application. Certains ensembles de données sont relativement petits et bien structurés, tandis que d'autres peuvent être extrêmement volumineux et complexes, nécessitant des techniques de gestion et de manipulation des données plus avancées. Les praticiens des données doivent avoir une compréhension approfondie des ensembles de données avec lesquels ils travaillent, y compris leur signification, leurs caractéristiques spécifiques, leur qualité et leurs biais potentiels. Cela leur permet de prendre des décisions éclairées lors de l'analyse, de l'exploration et de la modélisation des données.

1- Génération des datasets

Le mécanisme de génération de datasets en fichiers CSV à partir de logs générés à l'aide de BurpSuite et Acunetix comprend plusieurs étapes clés. Tout d'abord, BurpSuite et Acunetix sont des outils de sécurité utilisés pour effectuer des tests sur des applications web, et ils produisent des logs contenant des informations précieuses sur les requêtes, les réponses, les erreurs et les vulnérabilités détectées lors des tests. Pour générer un fichier CSV à partir de ces logs, un programme spécifique peut être développé. Ce programme analyse les logs générés par BurpSuite et Acunetix, extrait les informations pertinentes et les structure dans un format tabulaire avec des colonnes appropriées.

Les informations extraites peuvent inclure des détails tels que les URL, les méthodes HTTP, les codes de réponse, les vulnérabilités détectées et les paramètres de requête. Les bibliothèques spécifiques de manipulation de fichiers CSV, comme la bibliothèque csv en Python, peuvent être utilisées pour faciliter cette tâche.

Une fois que le fichier CSV est généré, il peut nécessiter un processus de nettoyage pour garantir la qualité des données. Cela implique l'élimination des données redondantes, des valeurs manquantes ou erronées, et l'application de transformations supplémentaires pour préparer les données à des fins d'analyse. Des outils tels que CSV Cleaner peuvent être utilisés pour simplifier le processus de nettoyage. CSV Cleaner est un site web qui propose des fonctionnalités de nettoyage de fichiers CSV en ligne, permettant d'appliquer des opérations de filtrage, de suppression des doublons, de normalisation des valeurs, etc. Une fois le nettoyage terminé, le fichier CSV est prêt à être utilisé dans des tâches d'analyse des données ou de création de modèles.

En complément, il est courant d'ajouter un champ supplémentaire appelé "classe" dans le fichier CSV généré à partir des logs. Ce champ "classe" permet de désigner si chaque ligne de données correspond à une entrée "bad" ou "good". Il est utilisé pour étiqueter les données et les classer en fonction de leur caractère potentiellement malveillant ou non.

La présence de ce champ "classe" dans le fichier CSV permet d'enrichir les données avec des informations supplémentaires pour les analyses ultérieures. Il facilite l'identification des schémas, des tendances et des modèles associés aux vulnérabilités ou aux comportements malveillants.

2- Importation des datasets

```

import pandas as pd
http = pd.read_csv(r'C:\Users\lenovo\Documents\PFE\legit\allall.csv')

test_http = pd.read_csv(r'C:\Users\lenovo\Documents\PFE\legit\allall.csv')
test_http.head()

```

path	body	single_q	double_q	dashes	braces	spaces	badwords	class
/sendFeedback submit=%20Submit%20&file=comments.txt&comment...		1.0	0.0	1.0	1.0	5.0	2.0	bad
/survey_questions.jsp	NaN	0.0	0.0	0.0	0.0	0.0	0.0	good
/Privacypolicy.jsp	NaN	0.0	0.0	0.0	0.0	0.0	0.0	good
0documents/JohnSmith/Bank%20Site%20Docume...	NaN	0.0	0.0	0.0	0.0	3.0	0.0	bad
/images	NaN	0.0	0.0	0.0	0.0	0.0	0.0	good

Figure 24 : Echantillonnage des datasets

3- Dataset principale

Le dataset principal utilisé dans notre étude comprend 45 000 logs clean provenant des tests effectués à l'aide de BurpSuite et Acunetix. Ces logs ont été préalablement nettoyés à l'aide d'outils tels que CSV Cleaner pour garantir la qualité des données. Pour analyser et exploiter ces données de manière efficace, nous avons utilisé la bibliothèque pycaret, qui permet d'automatiser les étapes de prétraitement, de formation et de test des modèles.

En utilisant pycaret, nous avons pu effectuer le training et le testing du modèle de classification CART (Classification and Regression Trees) de manière automatique. Pycaret offre une interface conviviale et simplifie le processus en gérant les tâches courantes telles que la séparation des données en ensembles d'entraînement et de test, la sélection des caractéristiques, le réglage des hyperparamètres et l'évaluation des performances du modèle.

En nous basant sur les lignes du dataset, nous avons également utilisé pycaret pour effectuer du clustering. Le clustering consiste à regrouper les données similaires dans des clusters distincts. Dans notre cas, pycaret a créé deux clusters : un cluster pour les requêtes bénignes (ou "good") et un autre cluster pour les requêtes malveillantes (ou "bad"). Cette approche de clustering nous permet de détecter les schémas et les comportements associés aux requêtes malveillantes, ce qui peut être précieux dans l'identification et la prévention des attaques potentielles.

IV- Clustering

Le module de regroupement de PyCaret est un module d'apprentissage automatique non supervisé qui effectue la tâche de regrouper un ensemble d'objets de manière à ce que les objets du même groupe (également appelé cluster) soient plus similaires entre eux que ceux des autres groupes.

1- API fonctionnelle

Le terme "Functional API" se réfère à une approche spécifique pour construire et manipuler des modèles d'apprentissage automatique à l'aide de bibliothèques telles que Keras ou TensorFlow. En utilisant l'API fonctionnelle, les modèles sont créés en définissant un graphe acyclique dirigé de couches (layers) qui représentent les opérations de transformation des données. Contrairement à l'approche séquentielle, où les couches sont empilées les unes sur les autres, l'API fonctionnelle permet des architectures de modèles plus complexes avec des connexions entre les couches non linéaires.

Dans le contexte du clustering dans PyCaret, l'API fonctionnelle peut être utilisée pour spécifier et configurer des modèles de clustering personnalisés. PyCaret est une bibliothèque Python pour l'apprentissage automatique automatisé, qui fournit des fonctionnalités pour la préparation des données, la sélection de modèles, l'optimisation des hyperparamètres et l'évaluation des modèles. En utilisant l'API fonctionnelle, vous pouvez créer des pipelines de clustering personnalisés en spécifiant les étapes de prétraitement des données, la configuration du modèle et les évaluations nécessaires.

Notez que les termes et les concepts spécifiques à l'apprentissage automatique peuvent varier en fonction du domaine et des outils utilisés.



Figure 25 : Exemple d'un api fonctionnelle

2- Kmeans Clustering

K-means clustering est un algorithme populaire utilisé dans le domaine de l'apprentissage automatique non supervisé pour regrouper un ensemble de données en différents clusters. Dans la bibliothèque Python PyCaret, l'algorithme K-means clustering est implémenté pour faciliter le processus de clustering.

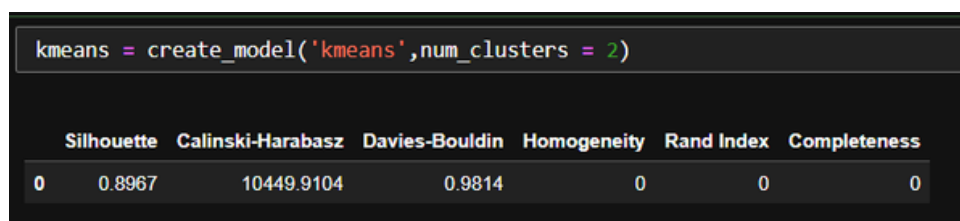
L'algorithme K-means commence par sélectionner au hasard k centres de cluster dans l'espace des données, où k est le nombre de clusters souhaité. Ensuite, pour chaque point de données, l'algorithme attribue le point au centre de cluster le plus proche en termes de distance euclidienne. Cette étape est appelée l'étape d'affectation.

Après l'étape d'affectation, les centres de cluster sont mis à jour en calculant la moyenne des points de données assignés à chaque cluster. Ce nouveau centre devient le point de référence pour les itérations suivantes. Ces étapes d'affectation et de mise à jour des centres sont répétées jusqu'à ce que les centres de cluster ne changent plus ou que le nombre d'itérations maximum soit atteint.

PyCaret simplifie l'utilisation de l'algorithme K-means clustering en fournissant une interface conviviale

3- Création du modèle

Cette fonction entraîne et évalue les performances d'un modèle donné. Les métriques évaluées peuvent être obtenues à l'aide de la fonction `get_metrics`. Des métriques personnalisées peuvent être ajoutées ou supprimées à l'aide des fonctions `add_metric` et `remove_metric`. Tous les modèles disponibles peuvent être consultés à l'aide de la fonction `models`.



```
kmeans = create_model('kmeans', num_clusters = 2)
```

	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
0	0.8967	10449.9104	0.9814	0	0	0

Figure 26 : Création du modèle

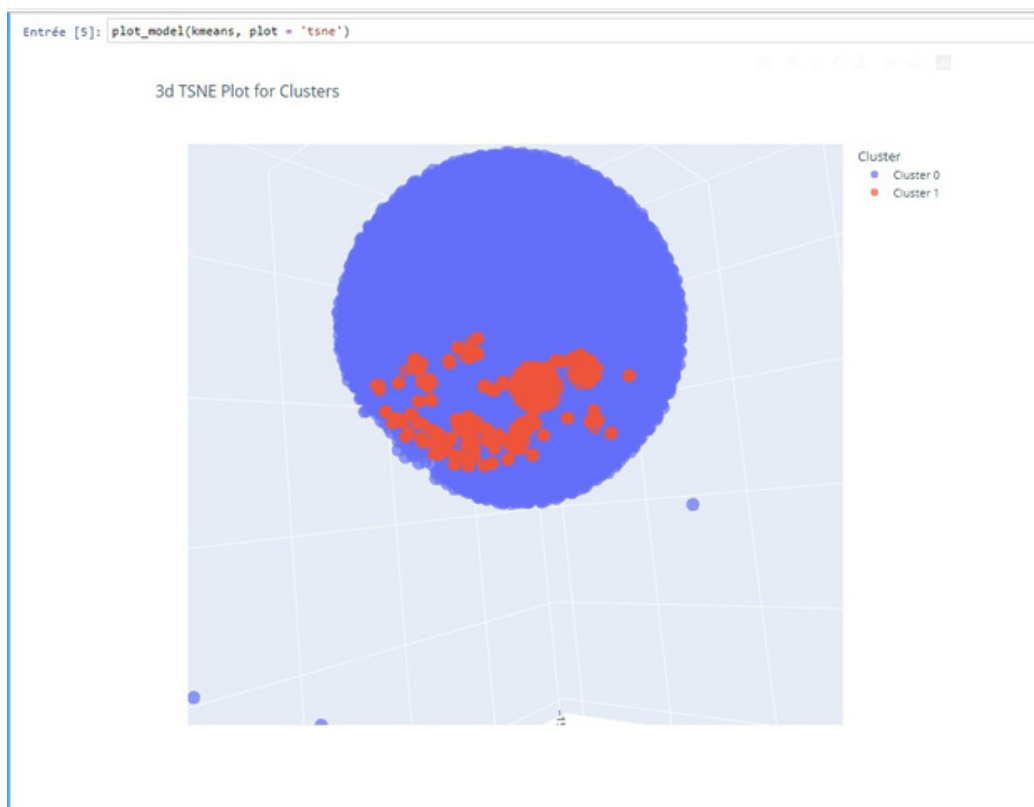


Figure 27 : visualisation graphique 3d du modèle

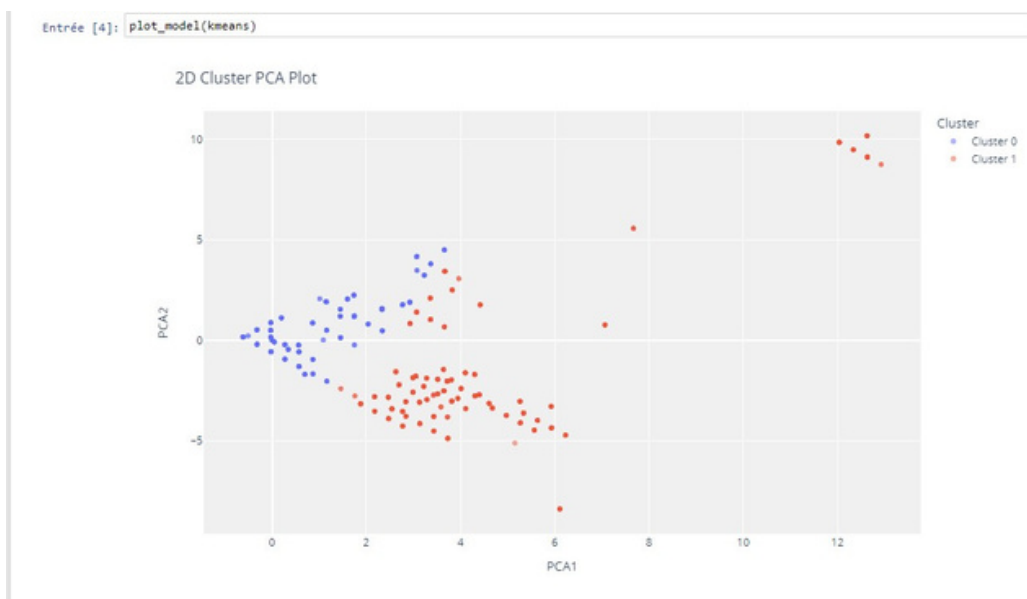


Figure 28 : visualisation graphique 2d du modèle

La fonction `plot_model` de PyCaret avec l'argument `plot='tsne'` génère un graphique de visualisation appelé t-SNE (t-Distributed Stochastic Neighbor Embedding) pour le modèle KMeans.

Le t-SNE est une technique de réduction de dimensionnalité qui permet de représenter des données multidimensionnelles de manière graphique en deux dimensions. Dans ce cas, il est utilisé pour visualiser les clusters (groupes) créés par l'algorithme de KMeans.

Le graphique t-SNE montre les points de données projetés dans un espace bidimensionnel, où des points similaires sont regroupés ensemble. Cela permet de visualiser les similarités et les différences entre les différents clusters formés par le modèle KMeans.

V- Proxy réalisé

Le système de proxy que nous avons développé intègre un modèle de prévention des attaques de type SQL Injection. Ce proxy est configuré pour fonctionner sur le port 8090 du serveur local, à l'adresse IP 127.0.0.1. Son objectif principal est de détecter et de prévenir les attaques SQL Injection sur les sites hébergés localement.

Lorsqu'une requête HTTP est dirigée vers un site hébergé sur le serveur local via le port 8090, le proxy analyse cette requête en utilisant le modèle de prévention des attaques SQL Injection que nous avons développé. Le modèle a été entraîné à reconnaître les schémas caractéristiques des attaques SQL Injection, permettant ainsi de détecter les tentatives malveillantes.

Si le modèle détecte une tentative d'attaque SQL Injection dans la requête, le proxy prend des mesures appropriées pour bloquer cette requête et prévenir toute exploitation de vulnérabilités potentielles dans le site ciblé. Il peut générer une alerte pour informer l'utilisateur de la tentative d'attaque et bloquer l'accès au site pendant un certain laps de temps afin de prévenir tout accès malveillant.

En utilisant le port 8090 pour notre proxy, nous pouvons rediriger les requêtes vers les sites hébergés localement vers notre système de prévention des attaques SQL Injection. Cela permet de renforcer la sécurité des sites web en détectant et en prévenant les attaques potentielles, offrant ainsi une protection supplémentaire contre les vulnérabilités liées aux attaques SQL Injection.

VI- Déploiement et expérimentation

Le déploiement et l'expérimentation de notre projet se déroulent de la manière suivante :

Tout d'abord, nous ouvrons le navigateur Firefox et accédons au site web Altoro à l'adresse <http://demo.testfire.net/>. Altoro est un site web largement reconnu dans les tests d'attaques web, en particulier les attaques de type SQL Injection. Cela nous permet de simuler un environnement réaliste pour nos expérimentations.

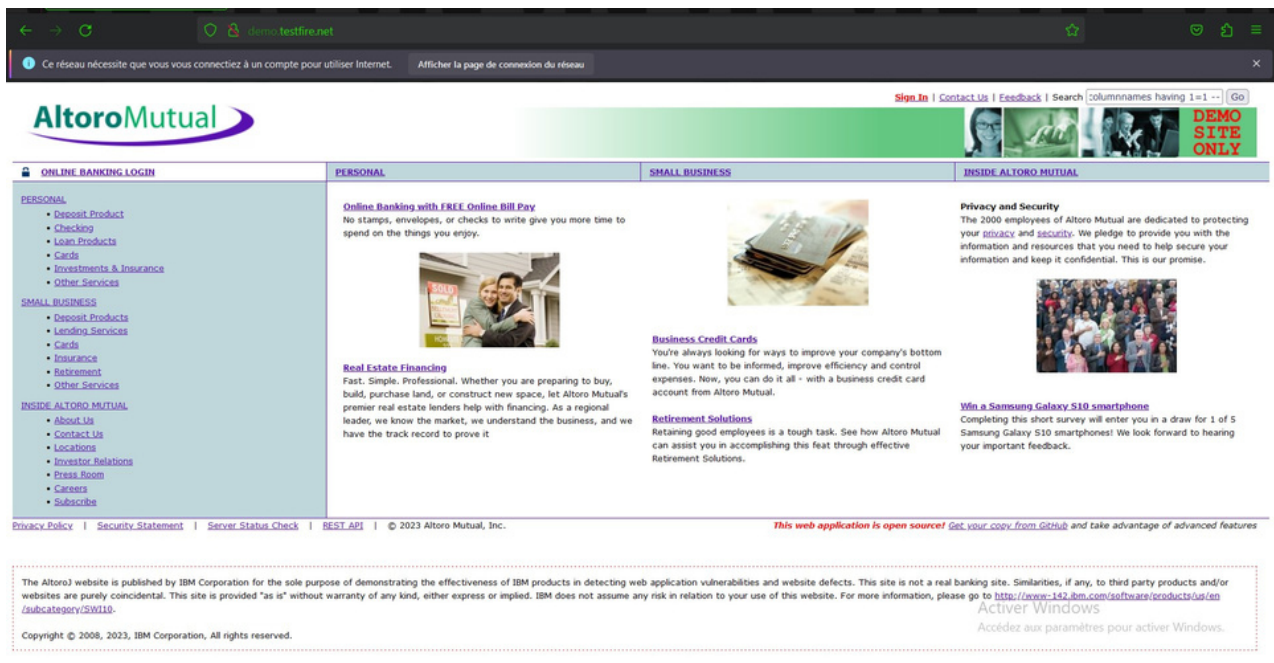


Figure 29 : Site web concerné par la protection

Ensuite, nous configurons les paramètres du proxy dans Firefox pour rediriger le trafic vers notre propre proxy. Nous spécifions l'adresse IP 127.0.0.1 et le port 8090 comme le proxy à utiliser. Cela permet à notre notebook IPS Proxy de capturer et d'analyser le trafic réseau provenant du navigateur.

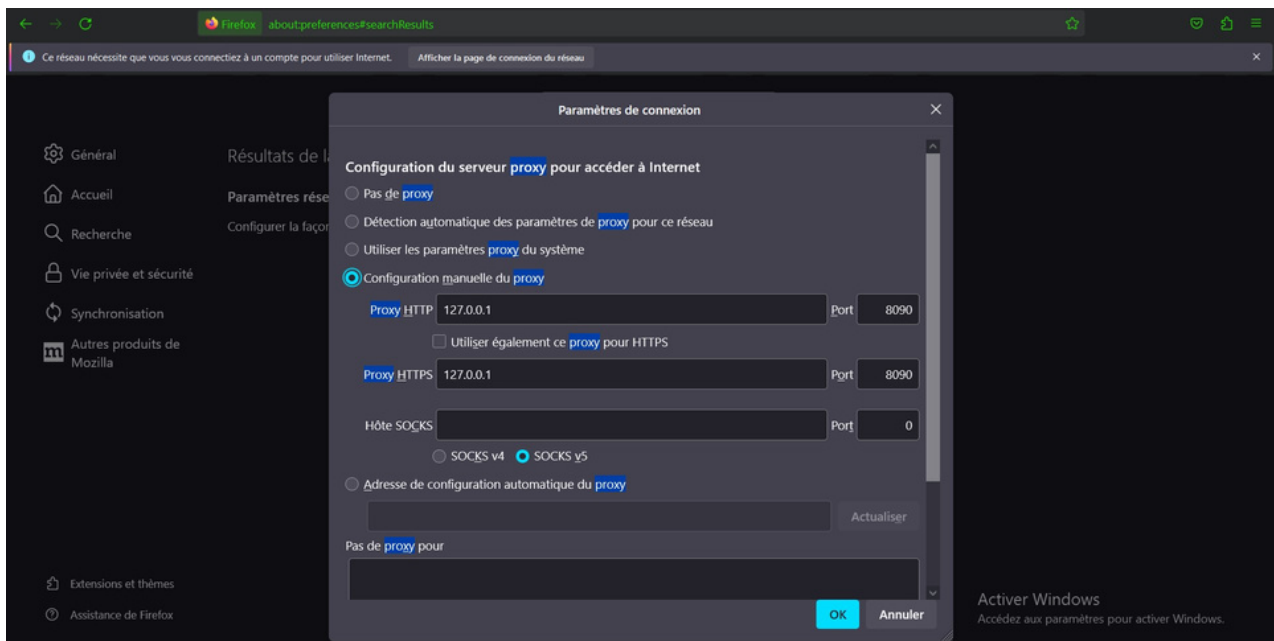


Figure 30 : Configuration du proxy

Nous exécutons ensuite notre notebook IPS Proxy qui se met à l'écoute du trafic réseau. Le proxy analyse chaque requête HTTP entrante et sortante et applique notre modèle de prévention des attaques SQL Injection pour détecter les requêtes malveillantes. Les requêtes jugées malveillantes sont bloquées, tandis que les requêtes jugées bénignes sont autorisées à passer.

```
Entrée [*]: class SimpleHTTPProxy(SimpleHTTPRequestHandler):
    proxy_routes = {}
    @classmethod
    def set_routes(cls, proxy_routes):
        cls.proxy_routes = proxy_routes

    def do_GET(self):
        #print (self.proxy_routes)
        #print (self.path)
        parts = self.path.split('/')
        print (parts)
        live_data = ExtractFeatures(parts[3])
        result = predict_model(kmeans , data = live_data)
        print(result['Cluster'][0])
        if result['Cluster'][0] == "Cluster 1":
            print('Intrusion Detected !')
            #print (parts, len(parts), part[1])
            if len(parts) >= 2:
                #url = self.proxy_routes[parts[2]] + '/' + parts[2:]
                #print(url)
                self.proxy_request('http://' + parts[2] + '/')
            else:
                super().do_GET()

    def proxy_request(self, url):
        #print('proxy req')
        #exit()
        try:
            response = request.urlopen(url)
        except error.HTTPError as e:
            #print('error')
            self.send_response_only(e.code)
            self.end_headers()
            return
        self.send_response_only(response.status)
        for name, value in response.headers.items():
            self.send_header(name, value)
        self.end_headers()
        self.copyfile(response, self.wfile)
SimpleHTTPProxy.set_routes({'proxy_route': 'http://demo.testfire.net/'})
with HTTPServer(('127.0.0.1', 8090), SimpleHTTPProxy) as httpd :
    host , port = httpd.socket.getsockname()
    print(f'Listening on http://{host}:{port}')
    try:
        httpd.serve_forever()
    except KeyboardInterrupt:
        print("\nKeyboard interrupt received , exiting .")

Listening on http://127.0.0.1:8090
```

Figure 31 : Executer le notebook IPS proxy

Cependant, étant donné que nous travaillons en localhost, il est impossible d'établir une connexion directe avec le serveur web d'Altoro. Par conséquent, notre tâche principale consiste à analyser les requêtes qui transitent par notre proxy et à évaluer l'efficacité et l'exactitude de notre modèle de prévention des attaques SQL Injection. Nous mesurons l'exactitude de notre modèle en comparant les prédictions du modèle aux résultats réels des attaques SQL Injection connues sur Altoro.

Cette approche nous permet de déployer notre projet dans un environnement contrôlé, en simulant des attaques réelles tout en protégeant les systèmes réels d'Altoro. Nous pouvons ainsi expérimenter et évaluer l'efficacité de notre modèle de prévention des attaques SQL Injection dans des conditions proches de la réalité.

1- Cas de détection d'une attaque SQLInjection

En cas d'une attaque web, le modèle implémenté dans le notebook IPS Proxy est capable de la détecter. Cependant, dans notre configuration actuelle en localhost, nous ne générons pas d'alerte ou de notification à l'utilisateur. Notre objectif principal est de réaliser la détection dans le backend du système, en analysant le trafic des requêtes et en utilisant le modèle de prévention des attaques SQL Injection. Par conséquent, le proxy fonctionne comme une couche de sécurité supplémentaire en filtrant les requêtes malveillantes et en permettant uniquement le passage des requêtes légitimes vers le serveur web local.

```
['http:', '', 'demo.testfire.net', 'search.jsp?query=%27+OR+%27%27+%3D+%27']  
[4, 0, 0, 0, 0, 0]  
Cluster 1  
Intrusion Detected !  
['http:', '', 'demo.testfire.net', 'search.jsp?query=%27+OR+%27%27+%3D+%27']  
[4, 0, 0, 0, 0, 0]  
Cluster 1  
Intrusion Detected !  
['http:', '', 'demo.testfire.net', 'search.jsp?query=%7C%7C']  
[0, 0, 0, 0, 0, 0]  
Cluster 0  
['http:', '', 'demo.testfire.net', 'search.jsp?query=%27+GROUP+BY+columnnames+having+1%3D+--']  
[1, 0, 1, 0, 0, 0]  
Cluster 1  
Intrusion Detected !
```

Activer V

Figure 32 : Cas d'une attaque SQLInjection

2- Cas normal

Dans le cas d'un scénario normal, où il n'y a pas de détection d'attaque web, le modèle dans le notebook IPS Proxy ne détecte aucune activité suspecte. Le système fonctionne en tant que proxy sécurisé, permettant le passage transparent des requêtes légitimes entre le navigateur et le serveur web local. Les requêtes sont analysées, mais aucune anomalie ou tentative d'attaque n'est identifiée par le modèle. Cela indique que le trafic est considéré comme sûr et conforme aux règles prédéfinies. Le proxy joue alors un rôle essentiel dans la sécurisation des communications entre l'utilisateur et le serveur web, offrant une protection contre les attaques potentielles sans perturber l'expérience utilisateur normale.

```
['http:', '', 'detectportal.firefox.com', 'success.txt?ipv4']  
[0, 0, 0, 0, 0, 0]  
Cluster 0  
['http:', '', 'detectportal.firefox.com', 'success.txt?ipv6']  
[0, 0, 0, 0, 0, 0]  
Cluster 0  
['http:', '', 'detectportal.firefox.com', 'success.txt?ipv4']  
[0, 0, 0, 0, 0, 0]  
Cluster 0
```

Figure 33 : Cas normal

Conclusion

Les attaques web, par four type, ne cessent de nuire aux différents utilisateurs d'internet et sont désormais la tache noire du web qui offre un apport énorme à l'humanité toute

Dans ce projet, on a mis en œuvre un système qui analyse un trafic réseau pour détecter les SQL injections dans ceci et les dénoncer comme requêtes malicieuses, on a découvert les différents algorithmes du Machine Learning, les linéaires parmi eux qui ne servaient pas à la résolution de notre problème, ainsi que ceux de classification comme l'arbre de décision qui est avéré très puissant et offrait des résultats en termes de précision très satisfaisants.

Comme perspectives, on suggère aux promotions suivantes d'améliorer encore ce système en employant un ensemble de données plus large et à jour, étendre ce modèle à faire une classification multi classe et définir le type d'attaque émanant. Ainsi, il serait très utile d'employer les techniques du Deep Learning pour un temps de réponse réduit et plus d'accuracy face à un trafic réseau très grand.

Nous remercions également sincèrement notre respecté professeur Abdallah RHATTOY pour tous les efforts précieux et toutes les informations qu'il n'a pas épargné pour nous fournir, et nous le remercions également d'avoir ouvert la porte pour communiquer avec nous à tout moment.

REFERENCES

- <https://stackoverflow.com>
- <https://www.anaconda.com/about-us>
- <https://pycaret.gitbook.io/docs/get-started/quickstart>
- <https://www.acunetix.com/about/>
- <https://portswigger.net/about>
- <https://code.visualstudio.com/learn>