Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études
pour l'obtention du diplôme de Master en Informatique

Option: *Réseaux et Systèmes Distribués (R.S.D)*

Thème

Implémentation d'un IDS hybride dans le cadre d'une application Web

Réalisé par :

- **M^r^ DELLAL Mohamed Seddik**
- **M^r^ BENAHMED DAHO Mourad**

Présenté le 24 Juin 2014 devant le jury composé de MM.

- *M^r^ LEHSAINI Mohamed* *(Président)*
- *M^r^ BENMAMMAR Badr* *(Encadreur)*
- *Mme LABRAOUI Nabila* *(Examinateur)*
- *M^r^ BELHOUCINE Amin* *(Examinateur)*

Année universitaire : 2013-2014

Remerciements

*En premier lieu, nous remercions **DIEU** le très haut de nous avoir aidé et donner la force et la volonté pour achever ce modeste travail.*

*Nous tenons à remercier en cette occasion tout le corps professoral et administratif de département d'informatique de l'université **ABOU BAKR BELKAID** de Tlemcen.*

*Nous tenons à remercier sincèrement **Mr Badr Benmammar** d'avoir proposé et encadré ce sujet. Nous lui exprimons notre profonde gratitude pour nous avoir fait profiter de ses connaissances, mais aussi de ses méthodes de travail, et surtout de son aide et ses conseils précieux. Grâce à lui, nous avons découvert un domaine de développement qui aujourd'hui nous passionne.*

Nos remerciements vont également aux membres du jury qui ont accepté de juger ce travail.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours soutenue et encouragé au cours de la réalisation de ce mémoire.

Merci à tous et à toutes.

Dédicace

A l'aide de DIEU tout puissant, qui trace le chemin de ma vie, Nous avons pu arriver à réaliser cet humble travail que je dédie :

A mes très chers parents pour leur amour et sacrifices,

A mes adorables frères, sœur pour leur patience,

A toute ma grande famille pour leur encouragement,

A mes proches amis pour leurs soutient,

A Hafsa Benammar,

A mon binôme Mourad,

A mes enseignants et surtout Mr Badr Benmammar,

A toutes les personnes qui me connaissent de proche ou de loin.

Mohamed Seddik DELLAL

Dédicace

Je dédie ce mémoire :

A une personne que j'ai tant aimé qu'il assiste à ma soutenance : le regretté mon cher père.

A mon adorable mère qui m'a beaucoup donné de l'amour et l'espoir.

A mes chers frères Omar, Youcef, Abdelhak et ma sœur Fatna el Hadja … Pour leur soutien moral

et leurs sacrifices le long de ma formation.

A mon cher binôme Seddik.

A mon cher encadreur BENMAMMAR Badr.

A mes chers amis Abdellah, Sofiane, Houcème, son oubliant Ganda et Désiré.

A mon adorable enseignante OUED EL ARBI Fadhila.

Enfin je le dédie à ma famille et à tous mes amis que je n'ai pas cités et à tous ceux qui me connaissent, en particulier mes amis africains, sahariens et ceux de la ville

d'Ain Temouchent.

Qu'ils trouvent à travers ce travail ma sincère reconnaissance.

BENAHMED DAHO Mourad

Résumé :

Un système de détection d'intrusions (IDS) est un système capable de scruter l'accès et le flux d'information, collecter tous les événements, les analyser et générer des alarmes en cas d'identification de tentatives malveillantes. Deux approches coexistent dans la détection d'intrusions : l'approche par signatures et l'approche comportementale. Chacune des deux présentent des points forts, mais aussi des faiblesses qui sont les faux positifs et les faux négatifs. Notre objectif est de sécuriser une application Web (boutique en ligne) en se basant sur les deux approches et en utilisant une méthode de classification afin de cumuler les forces et d'éliminer les faiblesses.

Mots-clefs : IDS, approche comportementale, approche par signatures, approche hybride, faux positifs, faux négatifs, clustering, CAH.

Abstract :

An intrusion detection system (IDS) is a system able to scan the access and the flow of information, to collect all the events, to analyse them and raise the alarm when there is identification of malicious attempts. Two approaches coexist in the intrusion detection: signature-based approach and behavior-based approach. Each of the two has strengths, but also weaknesses that are false positives and false negatives. Our goal is to secure a web application (online shop) based on the two approaches and using a classification method in order to increase the strengths and to eliminate the weaknesses.

Keywords: IDS, Behavior-based IDS, signature-based IDS, hybrid approach, false positives, false negatives, clustering, CAH.

ملخص :

نظام كشف التسلل (الهوية) هو نظام قادر على فحص الوصول الى تدفق المعلومات، جمع جميع الأحداث مع تحليلها و توليد الإنذارات في حالة تحديد محاولات معادية. تعتمد أساليب كشف التسلل على نهجين رئيسيين : نهج السلوكية و نهج السيناريو. كل منهما لديه نقاط قوة و لكن أيضا نقاط ضعف و التي تتمثل في الأخطاء الإيجابية و الأخطاء السلبية. هدفنا هو إدارة تطبيق ويب (متجر على شبكة الأنترنيت) مستندين على كلا النهجين، و باستخدام طريقة تصنيف للجمع بين نقاط القوة و القضاء على نقاط الضعف.

الكلمات المفتاحية: نظام كشف التسلل، نهج السلوكية، نهج السيناريو، نهج هجين، الأخطاء الإيجابية، الأخطاء السلبية، جميع، التصنيف الهرمي.

Table des matières

Chapitre 3: Implémentation d'une Application Web

Liste des Figures

Liste des Abréviations

BDD	**B**ase **D**e **D**onnées
CDDL	**C**ommon **D**evelopment and **D**istribution **L**icense
HIDS	**H**ost **I**ntrusion **D**etection **S**ystem
HTML	**H**yper Text **M**arkup **L**anguage
HTTP	**H**yper **T**ext **T**ransfer **P**rotocol
IDE	**I**ntegrate **D**evelopment **E**nvironment
IDS	**I**ntrusion **D**etection **S**ystem
IP	**I**nternet **P**rotocol
JSP	**J**ava **S**erver **P**age
MVC	**M**odel **V**iew **C**ontroller
NIDS	**N**etwork **I**ntrusion **D**etection **S**ystem
SQL	**S**tructured **Q**uery **L**anguage
SI	**S**ystème d'**I**nformation
XML	e**X**tensible **M**arkup **L**anguage
CAH	**C**lassification **A**scendante **H**iearchique

Introduction Générale

Introduction générale :

L'informatique et en particulier Internet joue un rôle très important dans notre société. Malheureusement, le développement de l'informatisation des échanges s'accompagne du développement d'activités malveillantes qui s'évolue d'une manière exponentielle avec le temps. Car de nombreuses applications critiques d'un point de vue de leur sécurité sont déployées dans divers domaines comme le domaine militaire, la santé, le commerce électronique, etc.

Les attaques et les menaces à distance sont devenues de plus en plus grave à cause de la connectivité croissante à l'internet. Cette croissance de connectivité sur internet pose des problèmes à savoir l'augmentation de l'information qui sera difficile à contrôler, les vulnérabilités et failles continuellement découvertes dans les systèmes informatiques (systèmes d'exploitation, applications, protocoles de communication, etc.).

De nos jours, une des solutions est le pare-feu, il permet juste de réduire partiellement les risques, cependant un réseau protégé par un pare-feu demeure tout de même pénétrable.

Une solution plus perfectionnée est l'utilisation des IDS (Intrusion Detection System), la détection d'intrusions consiste à scruter le trafic réseau, collecter tous les événements, les analyser et générer des alarmes en cas d'identification de tentatives malveillantes. Le classement des IDS se fait généralement par les techniques de détection ou leurs architectures. Malgré leur utilité, en pratique, le nombre important de faux positifs et de faux négatifs pose un problème délicat pour les IDS.

- ✓ Les faux positifs, c'est-à-dire les fausses alertes sont générées lorsque l'IDS identifie des activités normales comme des intrusions.
- ✓ les faux négatifs correspondent aux attaques ou intrusions qui ne sont pas détectées, aucune alerte n'est générée.

Les techniques de détection d'intrusions se répartissent en deux grandes classes : approche comportementale, appelée aussi détection d'anomalies, et approche par signatures, nommé aussi, détection d'attaques.

- L'approche par scénario : se base sur des signatures d'attaques déjà connues et qui peuvent être fournies par l'expert du domaine. L'inconvénient majeur de cette

approche est son incapacité à détecter les nouvelles attaques (dont les signatures sont encore inconnues).

- L'approche comportementale : suppose que l'activité normale est différente de l'activité intrusive. Il suffit alors d'élaborer un profil pour l'activité normale et un mécanisme permettant de comparer l'activité courante au profil établi pour détecter des écarts sensibles qui seront considérés des éventuelles intrusions. Un tel profil peut être obtenu en observant, pour une durée suffisante, l'activité normale au sein du système d'informations. C'est en ce sens que l'approche comportementale peut détecter les nouvelles attaques.

Ces deux approches seront, dans ce qui suit, les approches sur lesquelles on se base pour proposer un modèle d'IDS hybride, dans l'objectif est de sécuriser une application Web en utilisant en particulier la méthode de classification (CAH).

Pour cela, on a réparti notre travail en trois chapitres, comme suite :

Le premier chapitre se compose de deux parties, la première s'articule sur une généralité sur la sécurité, et la deuxième partie est consacrée aux IDS. Nous présentons la définition, l'architecture globale et le mode de fonctionnement des IDS, ainsi que la classification de ses derniers et enfin les méthodes de détection des intrusions.

Le deuxième chapitre est consacré aux méthodes de classification. En premier lieu, nous allons commencer par les définitions, puis les types de méthodes de classification. Ensuite, nous détaillerons les méthodes de classification.

Le troisième chapitre présente les différents outils qui vont servir à l'implémentation de notre projet, ainsi que l'implémentation de ce dernier. Nous avons par la suite élaboré un système de détection d'intrusion hybride en se basant sur deux approches : une approche comportementale optimisée par l'algorithme CAH et une approche par signature, afin de minimiser les fausses alertes.

Chapitre 1

Système de Détection d'Intrusion

I. Introduction :

Dans un contexte de connectivité croissante des réseaux informatiques, la sécurisation des systèmes d'informations est devenue un enjeu majeur. Ainsi, les systèmes et réseaux informatiques sont déployés dans différents domaines comme la banque, les assurances, la médecine ou encore le domaine militaire. L'accroissement de l'interconnexion de ces divers systèmes et réseaux, les a rendus accessibles par une population diversifiée d'utilisateurs qui ne cesse d'augmenter. Ces utilisateurs, connus ou non, ne sont pas forcément pleins de bonnes intentions vis-à-vis de ces réseaux. En effet, ils peuvent essayer d'accéder à des informations sensibles pour les lire, les modifier ou les détruire ou encore tout simplement pour porter atteinte au bon fonctionnement du système. Dès lors que ces réseaux sont apparus comme des cibles d'attaques potentielles, les sécuriser est devenu un enjeu incontournable.

De nombreux mécanismes ont été développés pour assurer la sécurité des systèmes d'informations et tout particulièrement pour prévenir les intrusions.

Malheureusement, ces mécanismes ont des limitations. En effet, les systèmes informatiques présentent des failles de conception, d'implémentation et de configuration qui permettent à des attaquants de contourner les mécanismes de prévention. Pour cela une seconde ligne de défense est nécessaire, la détection d'intrusions.

Ce chapitre est subdivisé en deux partie, la première représente des définitions et des stratégies de la sécurité, dans la deuxième sera posé la notion de la détection d'intrusions, les systèmes de détection d'intrusions seront ensuite abordés en présentant les approches de détections, la classification des IDS selon différents critères et terminant par la détection d'intrusion web.

II. Généralités sur la sécurité informatique :

II.1. La sécurité informatique :

La sécurité informatique est l'ensemble des moyens matériels et logiciels mis en œuvre pour minimiser la vulnérabilité d'un système contre des menaces accidentelles ou

intentionnelles, permettant ainsi aux systèmes informatiques de fonctionner normalement.

Elle consiste, aussi, à s'assurer que celui qui modifie ou consulte les données du système en a l'autorisation et qu'il peut le faire car le service est disponible. [1]

Sécuriser les données, c'est garantir : [1]

- **L'authentification :** le but de l'authentification est de garantir l'origine:
  - D'une information : Prouver qu'une information provient de la source annoncée (auteur, émetteur).
  - D'une personne (ou machine, groupe ou organisation): Prouver que l'identité est bien celle annoncée.
- **L'intégrité :** l'intégrité est d'assurer que les données n'ont pas été modifiées et empêcher toute modification (intentionnelle ou accidentelle) non explicitement requise par une entité habilitée. Cela permet, par exemple, au récepteur d'un message d'être raisonnablement assuré que le message reçu est le même que le message envoyé. Donc, l'intégrité des données est la propriété qui assure qu'une information n'est modifiée que dans des conditions prédéfinies (selon des contraintes précises).
- **La confidentialité :** la confidentialité est de garder secret le contenu de l'information et empêcher (ou prévenir) sa divulgation à des entités (sites, organisation, personnes, etc.) non habilitées à le connaître. Seuls les destinataires prédéterminés doivent être capables de lire le contenu du message.
- **La non répudiation :** pour éviter la contestation par l'émetteur de l'envoi de données, la non répudiation est une propriété qui assure que l'auteur d'un acte ne peut ensuite nier l'avoir effectué (signature de l'acte) et que le récepteur ne peut ultérieurement dénier avoir reçu un message (exemple exécution d'un ordre boursier, d'une commande…).

Définir une politique de sécurité pour un système d'information revient à élaborer l'organisation et les mécanismes à mettre en place et l'ensemble des mesures à prendre en vue de :

- Réduire dans la mesure du possible l'utilisation frauduleuse, l'altération et l'accès non autorisé au système et ses ressources (machines, réseau, etc.).
- Détecter aussi rapidement que possible toute activité, malveillante ou accidentelle, pouvant porter atteinte à la confidentialité, intégrité et

disponibilités des ressources et services.

- Prendre aussi efficacement que possible des contre-mesures afin de limiter les conséquences et poursuivre éventuellement l'auteur du forfait.

II.2. Protection des systèmes d'informations :

Pour atteindre les objectifs de la sécurité, une stratégie de sécurité doit minutieusement évaluer ce qui est à protéger dans le système d'informations, les risques potentiels, les vulnérabilités du système à sécuriser, pour dégager de cette analyse les mécanismes et moyens nécessaires, tenant compte bien entendu d'autres considérations telles que le coût de déploiement et la complexité de cette stratégie. Entre autres mécanismes utilisés pour la sécurité des systèmes d'informations, ci-après les plus répandus [1]:

- **Authentification et contrôles d'accès aux ressources** : il s'agit en premier lieu de la sécurité physique des équipements et installations. L'authentification est un mécanisme permettant de prouver l'identité d'un utilisateur (mots de passe, cartes à puce, méthodes biométriques, etc.) et de lui accorder uniquement les privilèges nécessaires pour l'accomplissement de ses tâches.
- **Scanners de vulnérabilités** : les scanners de vulnérabilités automatisent la découverte des failles de sécurité. Ils sont utilisés par les attaquants pour localiser les faiblesses du réseau cible. De plus les administrateurs peuvent en tirer profit pour corriger les vulnérabilités de leurs systèmes informatique. Cependant, malgré le grand nombre de vulnérabilités détectées, les scanners d'aujourd'hui sont inaptes à déterminer toutes les faiblesses possibles. De plus, la mise à jour de ces produits ne suit pas le rythme de la découverte des nouvelles vulnérabilités.
- **La cryptographie** : les informations sensibles et confidentielles sont souvent cryptées pour empêcher leur lecture même si elles étaient dérobées ou accédées frauduleusement. La cryptographie garantit la confidentialité, l'intégrité, la non répudiation et l'authenticité des données mais elle ne constitue pas une solution unique et suffisante de sécurité. Effectivement, diverses implémentations des protocoles de sécurité se sont révélées vulnérables. De plus la sécurité peut être rompue via plusieurs types d'attaques (par exemple : l'homme du milieu (MITM), les courts et les simples mots de passes utilisés sont facilement

cassables, le risque de vol des clés privées).

- **Les pare-feu [Firewall]** : un pare-feu (ou coupe-feu) est un dispositif matériel et logiciel d'interfaçage entre le réseau à sécuriser et un autre réseau externe, jugé moins sûr, et potentiellement source d'attaques. Un pare-feu assure d'abord une fonction de filtrage des flux entrants et sortants puisqu'il est un passage obligé pour tout échange. Le risque d'attaques distantes est alors réduit puisque le pare-feu ne laisse passer que le trafic d'une certaine plage d'adresses IP et relatif à certains ports et services. Malgré leurs grands intérêts, les pare feux présentent quelques lacunes. En effet, un attaquant peut exploiter les ports laissés ouverts pour pénétrer au réseau local. Les scripts constituent aussi des sources d'intrusion que les pare feux échouent à détecter. Ainsi l'opération supplémentaire d'encapsulation/décapsulation des données permet à l'attaquant de contourner le pare feu.
- **Protection anti-virus** : les logiciels anti-virus sont largement utilisés pour les stations de travail et ordinateurs personnels. Ils détectent et protègent contre les virus pouvant se propager à travers des fichiers, courrier électronique, etc.
- **Audit de sécurité et détection d'intrusions** : la plupart des systèmes d'exploitations et applications sont dotés de mécanismes d'audit permettant d'enregistrer tout ou une partie des événements (exécutions de commandes, utilisation de ressources, etc.) ayant lieu sur le système. Ces données peuvent faire l'objet d'une analyse en temps réel ou en différé pour détecter des activités malveillantes, suspectes ou toute activité pouvant violer la politique de sécurité. Cependant, ces données contiennent beaucoup d'informations normales et anormales. La taille énorme des fichiers contenant ces données pose souvent des problèmes de stockage et d'exploration du contenu. Les administrateurs fournissent aussi l'effort pour localiser les activités anormales, comprendre les objectifs des attaquants et déterminer les vulnérabilités exploitées du système.

II.3. Nécessité de la détection d'intrusions :

L'importance particulière qu'il convient selon [2] d'apporter à la détection d'intrusions tient à trois raisons :

- De nombreux systèmes existants sont vulnérables aux intrusions externes (attaques), et internes (abus de la part d'utilisateurs habilités à se servir du

système).

- Les systèmes existants ne sont pas toujours remplaçables par des systèmes plus sécurisés, soit pour des raisons de coût (développer un système hautement sécurisé est une tâche difficile), soit car l'apport de sécurité se fait au détriment du confort d'utilisation du système (voire de sa philosophie, c'est notamment le cas du système UNIX).
- Même les systèmes les plus sécurisés sont vulnérables aux abus de la part des utilisateurs habilités et aux canaux cachés. Il est important de noter que ces deux risques, ne violant aucune des règles de la politique de sécurité, ne sont détectables que par audit.

III. Définition d'un IDS :

Le concept de système de détection d'intrusions a été introduit en 1980 par James Anderson [3]. Mais le sujet n'a pas eu beaucoup de succès. Il a fallu attendre la publication d'un modèle de détection d'intrusions par Denning en 1987 [4] pour marquer réellement le départ du domaine.

La détection d'intrusion est devenue une industrie mature et une technologie éprouvée : à peu près tous les problèmes simples ont été résolus, et aucune grande avancée n'a été effectuée dans ce domaine ces dernières années, les éditeurs de logiciels se concentrant plus à perfectionner les techniques de détection existantes.

Un IDS est un système informatique, composé généralement de logiciel et éventuellement de matériel, dont le rôle est la détection d'intrusions. Par définition, un IDS n'a pas de vocation préventive ou réactive dans la mesure où il n'empêche pas une intrusion de se produire.

Il se contente plutôt d'analyser certaines informations en vue de détecter d'éventuelles activités malveillantes qu'il aura à notifier dans les plus brefs délais au responsable de la sécurité du système. C'est pour cette raison que la majorité des IDS opèrent en temps réel.

Toutefois, il y'a des IDS qui réagissent suite à la détection d'une intrusion en mettant fin par exemple à une connexion suspecte.

Les IDS traditionnellement suivent deux critères:

- **Fiabilité :** toute intrusion doit effectivement donner lieu à une alerte. Une intrusion non signalée constitue une défaillance de l'IDS, appelée faux négatif. (voirFigureI.1)
- **Pertinence des alertes:** toute alerte doit correspondre à une intrusion effective. Toute « fausse alerte » (appelée également faux positif) diminue la pertinence de l'IDS. (voirFigureI.1)

Un IDS est parfaitement fiable en absence de faux négatif; il est parfaitement pertinent en l'absence de faux positif.

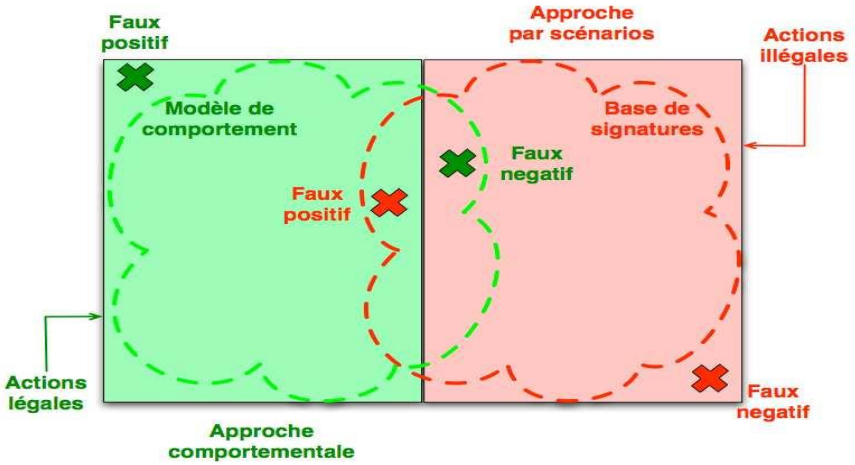


Figure I.1. Problèmes des IDS [5]

Les IDS propose les fonctions suivantes :

- ✓ Détection d'attaques (actives ou passives) ;
- ✓ Génération des rapports ;
- ✓ Outil de corrélation avec d'autres éléments d'architecture de sécurité ;
- ✓ Réaction aux attaques par le blocage de route ou la fermeture de connexion ;
- ✓ Transfer d'activités.

IV. Architecture d'un IDS :

Nous décrivons dans cette section les trois composants qui constituent classiquement un système de détection d'intrusions [5]. La FigureI.2 illustre les interactions entre ces trois composants.

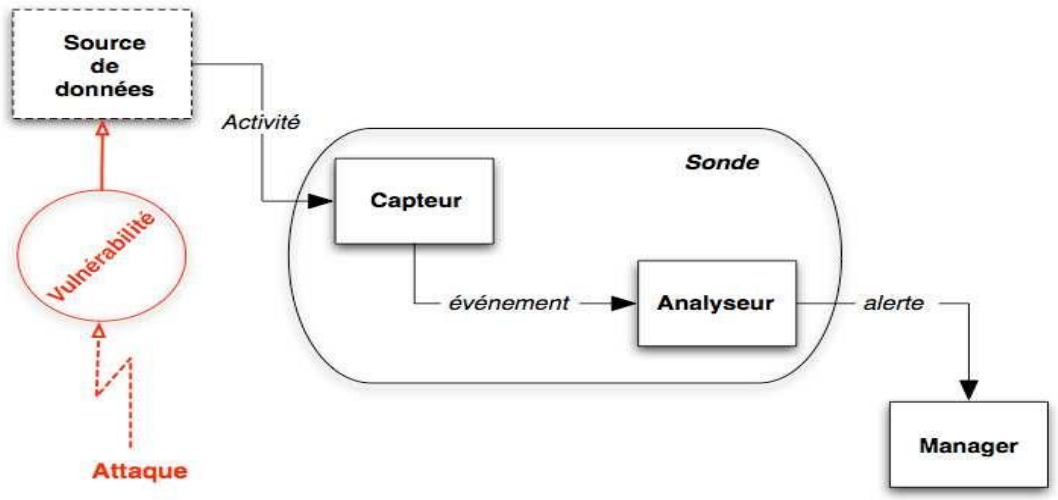


Figure I.2. Architecture classique d'un IDS [5]

IV.1. Capteur :

Le capteur observe l'activité du système par le biais d'une source de données et fournit à l'analyseur une séquence d'événements qui renseignent de l'évolution de l'état du système. Le capteur peut se contenter de transmettre directement ces données brutes, mais en général un prétraitement est effectué. [6]

On distingue classiquement trois types de capteurs en fonction des sources de données utilisées pour observer l'activité du système: les capteurs système, les capteurs réseau et les capteurs applicatifs.

IV.2. Analyseur :

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

IV.3. Manager :

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

qui peut être:

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

Du fait du manque de fiabilité des systèmes de détection d'intrusions actuels, les réactions sont rarement automatisées, car elles peuvent se traduire par un déni de service en cas de faux positif.

V. Méthodes d'analyses :

La technologie des systèmes de détection d'intrusions permet d'analyser les données recueillies de trois façons [7]:

V.1. Analyse centralisée :

L'IDS possède plusieurs capteurs, il centralise les alertes pour les analyser sur une seule machine. Ce type d'analyse présente l'avantage d'avoir une vue globale sur toutes les machines protégées. Toutefois, il a l'inconvénient d'occupation très longue du réseau pour acheminer l'information.

V.2. Analyse locale:

Chaque machine dispose d'un capteur et analyse l'information à son niveau. Avec ce type d'analyse le trafic réseau est diminué mais les attaques distribuées peuvent échapper à la détection.

V.3. Analyse distribuée :

Des petits programmes appelés agents sont déployés sur les nœuds du réseau.

Pour les besoins d'analyse un agent est envoyé sur une machine pour traiter l'information.

VI. Mode de fonctionnement d'un IDS :

Il faut distinguer deux aspects dans le fonctionnement d'un IDS : le mode de détection utilisé et la réponse apportée par l'IDS lors de la détection d'une intrusion. Il existe deux modes de détection, la détection d'anomalies et la reconnaissance de signatures. De même, deux types de réponses existent, la réponse passive et la réponse active. [8]

VI.1. Mode de détection :

a) La détection d'anomalies : elle consiste à détecter des anomalies par rapport à un profil "de trafic habituel". La mise en œuvre comprend toujours une phase d'apprentissage au cours de laquelle les IDS vont "découvrir" Le fonctionnement "normal" des éléments surveillés. Ils sont ainsi en mesure de signaler les divergences par rapport au fonctionnement de référence. [8].

b) La reconnaissance de signature : cette approche consiste à rechercher dans l'activité de l'élément surveillé les empreintes (ou signatures) d'attaques connues. Ce type d'IDS est purement réactif ; il ne peut détecter que les attaques dont il possède la signature. De ce fait, il nécessite des mises à jour fréquentes. [8].

De plus. L'efficacité de ce système de détection dépend fortement de la précision de sa base de signature.

Une signature permet de définir les caractéristiques d'une attaque, au niveau des paquets ou au niveau protocole.

VI.2. Réponse passive et active :

Il existe deux types de réponses, la réponse passive et la réponse active.

a) La réponse passive : la réponse passive d'un IDS consiste à enregistrer les intrusions détectées dans un fichier de log qui sera analysé par le responsable sécurité. Certain IDS permettent de logger l'ensemble d'une connexion identifiée comme malveillante. Ceci permet de remédier aux failles de sécurité pour empêcher les attaques

enregistrées de se reproduire, mais n'empêche pas directement une attaque de se produire. [8].

b) La réponse active : la réponse active au contraire a pour but de stopper une attaque au moment de sa détection. [8].

VII. Classification des IDS :

Plusieurs critères permettent de classer les systèmes de détection d'intrusion, la méthode d'analyse étant le principal. Deux méthodes dérivant de cette dernière existent aujourd'hui : l'approche comportementale et l'approche par scénario.

On peut citer aussi d'autres critères de classification des IDSs : la fréquence d'utilisation, les sources de données à analyser, le comportement de l'IDS après intrusion. [9]

VII.1. Approche comportementale :

Une approche proposée par James Anderson [3] puis reprise et étendue par Denning [4] consiste à utiliser des méthodes basées sur l'hypothèse selon laquelle l'exploitation d'une vulnérabilité du système implique un usage anormal de celui-ci. Une intrusion est donc identifiable en tant que déviation par rapport au comportement habituel d'un utilisateur. Bien sur une telle déviation peut avoir une autre cause qu'une attaque du système par exemple un changement de fonction de l'utilisateur au sein de l'entreprise. On s'attachera donc a trouver des méthodes possédant le plus fort taux de discrimination possible (c'est-à-dire ayant le plus fort taux de détection d'intrusions et le plus faible taux de fausses alarmes). De plus on se référera à un seuil au-delà duquel on considérera que le comportement est intrusif.

Cette approche, dont la question de base est « le comportement actuel de l'utilisateur est-il cohérent avec son comportement passé ? », est appelée approche comportementale. Pour caractériser le comportement normal d'un utilisateur (on parle de *profil*), des techniques employées pour modéliser le comportement se basent sur des méthodes statistiques, il est également possible d'envisager l'utilisation de systèmes experts ou de réseaux de neurones.

VII.2. Approche par scénario :

Contrairement à un système de détection d'anomalies, ce type de détecteur d'intrusions nécessite une maintenance active : puisque par nature il ne peut détecter que les attaques dont les signatures sont dans sa base, cette base doit être régulièrement (sans doute quotidiennement) mise à jour en fonction de la découverte de nouvelles attaques. Aucune nouvelle attaque ne peut par définition être détectée, ce qui implique un taux plus élevé de faux négatifs.

De manière générale, les détecteurs de scénario se montrent fiables pour signaler les attaques référencées dans la base. Théoriquement, leur taux de faux positifs devrait rester très faible, car par définition une alerte n'est levée que dans le cas où la signature d'une attaque est observée.

Cependant, pour des raisons de performance, les signatures sont souvent trop simples, peuvent donc leur correspondent des actions tout à fait légitimes. Le taux de faux positif reste donc élevé avec les outils existant aujourd'hui.

De plus, une éventuelle connaissance de la base de signatures (particulièrement dans le cas des *patterns*) permet en principe à l'attaquant de construire précisément un scénario non détectable, cela ne fait que renforcer encore l'exigence de maintenir régulièrement la base.

Ce type de détecteur reste assez facile à mettre en œuvre, ne nécessitant pas de phase d'apprentissage.

VII.3. Autres critères :

Parmi les autres critères de classification existants, nous pouvons citer entre autres :

VII.3.1. Les sources de données à analyser :

Les sources possibles de données à analyser sont une caractéristique essentielle des systèmes de détection d'intrusions. Les données proviennent, soit de fichiers générés par le système d'exploitation, soit de fichiers générés par des applications, soit encore d'informations obtenues en écoutant le trafic sur le réseau. [10].

VII.3.2. Le comportement de l'IDS après intrusion :

Une autre façon de classer les systèmes de détection d'intrusions, consiste à voir quelle est leur réaction lorsqu'une attaque est détectée. Certains se contentent de déclencher une alarme (réponse passive). [10].

VII.3.3. La fréquence d'utilisation :

Une autre caractéristique des systèmes de détection d'intrusions est leur fréquence d'utilisation : périodique ou continue. Certains systèmes de détection d'intrusions analysent périodiquement les traces d'audit à la recherche d'une éventuelle intrusion ou anomalie passée. Cela peut être suffisant dans des contextes peu sensibles.

La plupart des systèmes de détection d'intrusions récents effectuent leur analyse des traces d'audit ou des paquets réseau de manière continue afin de proposer une détection en quasi temps réel. Cela est nécessaire des contextes sensibles (confidentialité) ou commerciaux (confidentialité, disponibilité).C'est toutefois un processus couteux en temps de calcul car il faut analyser à la volée tout ce qui se passe sur le système. [10].

VIII. Détection d'intrusions Web :

Les serveurs Web sont un environnement de test intéressant pour la détection d'intrusions, d'une part, par leur importance et par l'universalité du protocole http et d'autre part, par le nombre de vulnérabilités les frappant. [11]

Les serveurs Web sont la vitrine des entreprises, associations, états, voir des individus par l'intermédiaire des blogs sur Internet. Ils sont, dans certains cas, une source de revenus importants (commerce en ligne par exemple). De plus en plus d'applications Web sont déployées sur Internet.

Les outils de détection d'intrusions utilisés pour détecter les attaques contre les serveurs Web utilisent principalement une approche par signatures bien que des approches comportementales soient apparues récemment. Nous allons donc présenter les différents IDS spécifiques au Web suivant leur approche de détection.

VIII.1. Approche par signatures :

Les IDS par signatures spécifiques au Web sont pour la plupart des HIDS au niveau applicatif. Ils évitent certains écueils des NIDS: reconstruction des paquets, perte de paquets en cas de charge, vulnérabilité aux techniques d'évasion, gestion de la cryptographie, etc. La plupart de ces outils utilisent les fichiers d'audit des serveurs Web comme source d'événements.

Almgren et al. [12] ont proposé également un système analysant les logs générés par le serveur Web, recherchant des motifs qui correspondent à des attaques connues. L'analyse peut se réaliser en temps réel et n'affecte pas les performances du serveur Web. Ce système est également capable d'apprendre de nouvelles attaques en surveillant plus particulièrement les activités des clients considérés comme suspects.

Un autre système WebSTAT [13] Utilise un langage de haut-niveau pour décrire les attaques en termes d'états et de transitions. La détection d'intrusions se fait en temps réel et plusieurs autres sources d'événements peuvent être utilisées.

Almgren et Lundqvist [14] ont proposé un système de détection d'intrusions intégré à un serveur Apache. L'avantage de cette solution réside dans sa capacité à détecter les intrusions à différents stades du traitement de la requête.

L'IDS présenté ci-dessus est à rapprocher de ModSecurity [15] qui est un module pour Apache permettant d'écrire des règles pour détecter, bloquer et modifier les requêtes parvenant au serveur Apache.

VIII.2. Approche comportementale :

Bien que les approches par signatures soient effectives, elles posent certains problèmes. La plupart des applications Web sont spécifiques et sont développées rapidement sans souci de sécurité particulier. Il est difficile d'écrire des signatures pour ces applications car il n'y a pas forcément de caractéristiques communes. L'approche comportementale semble ici adaptée à la nature des vulnérabilités.
On étudiant [11], nous avons constaté qu'il y a beaucoup de travaux qui sont réalisés dans ce contexte :

Breach Security propose un produit nommé *WebDefend* [16] qui modélise le trafic normal à destination et en provenance des applications Web protégées et ensuite

capable de bloquer les attaques.

Kruegel et al. [17][18] ont proposé le premier IDS comportemental spécifique aux serveurs Web. Il utilise les fichiers d'audit comme source d'événements. Il ne traite cependant que les requêtes de type GET comportant des paramètres et se concentre donc sur la détection d'attaques contre les scripts CGI.

Robertson et al. [19] présentent une amélioration des travaux précédents en ajoutant à la détection d'anomalies deux composants: un composant permettant la génération de signatures d'alertes et groupant les alertes suivant les signatures et un composant permettant d'identifier les anomalies suivant des heuristiques.

Valeur et al. [20] proposent également une amélioration des travaux de Kruegel et Vigna [21]. Leur approche permet de limiter l'influence des faux positifs dans un reverse proxy qui redirige les requêtes http anormales vers un serveur Web ayant un accès plus restreint aux parties critiques du site.

Ingham et al. [22] proposent également une méthode portant sur toute la requête HTTP. Après l'application de certaines heuristiques, ils modélisent les requêtes HTTP grâce à des automates à états finis déterministes. [23]

VIII.3. Approche hybride :

Une approche hybride a été proposée par Tombini et al. [24][25]. Cette approche consiste en la sérialisation d'un IDS comportementale suivi d'un IDS par signatures. L'IDS comportementale permet de filtrer les requêtes normales et ainsi seules les requêtes détectées comme anormales sont passées à l'IDS par signatures. Bien que l'IDS comportementale utilisé soit simple, ceci permet de réduire le nombre de faux positifs générés globalement. La source d'entrées est le fichier d'audit du serveur Web. Cet IDS est donc soumis aux mêmes problèmes que les autres utilisant cette source de données.

IX. Conclusion :

Nous avons présenté dans ce chapitre les qualités requises des systèmes de détection d'intrusions. Afin de remplir ces objectifs, diverses méthodes de détection d'intrusions ont été proposées. Elles se basent principalement sur deux principes de détection : la détection par anomalie et la détection par la connaissance. Nous avons expliqué ces deux principes de détection.

Dans le chapitre suivant nous présentons quelques méthodes d'intelligence artificielle utilisées dans ce contexte, plus particulièrement les méthodes de classification non supervisée, qui ont été utilisées pour la détection d'intrusions.

Chapitre 2

Méthode de Classification

I. Introduction :

La détection des tentatives d'attaques est une problématique très importante dans le domaine de la sécurité informatique. Les technologies classiques de protection de type Firewall filtrant sont en effet inefficaces contre la plupart des attaques actuelles. Aussi sont apparus de nouveaux équipements réseaux pour prendre en compte ces carences, systèmes de détection d'intrusions, dont le but est de détecter les tentatives d'attaque qu'un firewall ne peut pas bloquer. Malheureusement, en pratique, les IDS soit génèrent un nombre trop élevé d'alarmes ou ils sont incapables de détecter de nouvelles attaques. L'utilisation des méthodes de la classification non supervisée pour modéliser le comportement des utilisateurs peut être efficace pour construire des IDS générant le minimum de fausses alertes.

Nous reviendrons donc, dans ce chapitre, tout d'abord sur les définitions et les concepts de partitionnement de données, nous passerons ensuite à quelques types de méthodes de classification qui se basent sur l'approche de *clustering*(classification non supervisée), nous allons entamer la partie codage (prétraitement des données) ensuite la présentation de quelques méthodes de la classification non supervisée, et nous terminons par une conclusion.

II. Concepts et définitions :

II.1. La classification :

La classification est le processus qui permet de regrouper un ensemble d'objets en sous-ensembles de tel sorte que les objets appartenant au même sous ensemble présentent la plus grande similarité, et que deux objets de sous-groupes différents soient le moins similaire possible ou le plus dissimilaire [26].

Par classification, on entend l'identification de la classe d'un objet à partir des attributs le décrivant. Il s'agit, à titre d'exemple, d'identifier la maladie (classe) d'un patient (objet) à partir des symptômes (attributs descriptifs) qu'il manifeste. On peut l'assimiler à une fonction associant à une description d'un objet, la classe correspondante [1].

La classification est à la base de la plus part des applications en reconnaissance des formes. Celles-ci sont nombreuses et s'intéressent à des domaines aussi variés que l'analyse de l'écriture, la détection automatique d'objets particuliers dans des images (des visages par exemple), l'analyse des données bancaires, du marketing, la mesure d'audiences sur Internet ou enfin la détection d'intrusions dans des réseaux informatiques.

La classification, bien que primordiale dans toutes les applications citées précédemment, n'est pas la seule phase à considérer avec attention lors du développement d'un outil d'extraction et d'analyse de l'information [27]. La figure suivante montre le parcours de l'information à classifier.

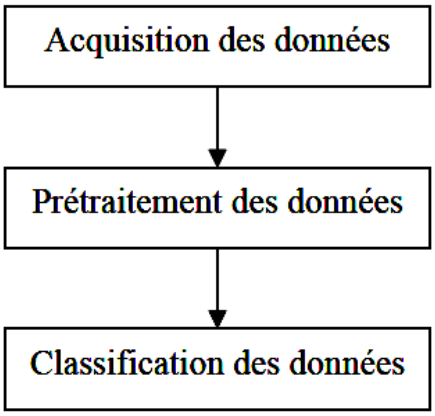


Figure II.1. Le parcours de l'information à classifier

 Acquisition des données : d'une manière générale, il s'agit à ce niveau de mettre en place l'ensemble de l'instrumentation (capteurs, matériel d'acquisition,…), de façons à reproduire le phénomène observé le plus fidèlement possible. C'est l'opération de transformation de l'information à traiter en signaux numériques manipulables par ordinateurs ou bien la numérisation de l'information.

 Prétraitement des données : cette phase correspond au filtrage des informations en ne conservant que ce qui est pertinent dans le contexte d'étude, c'est la partie segmentation dans la classification d'images.

 Classification des données : Elle correspond à l'étape de décision et pour cela plusieurs méthodes se présentent pour la résolution.

II.2. Mesures d'éloignement :

Notons Ω = *{i=1,..., n}* l'ensemble des individus. Cette section se propose de définir sur Ω * Ω différentes mesures d'éloignement entre deux individus. Les hypothèses et propriétés étant de plus en plus fortes. [28]

II.2.1. Indice de ressemblance, ou similarité :

C'est une mesure de proximité définie de Ω * Ω dans *R*+ et vérifiant :

$$\begin{aligned} s(i,j) &= s(j,i), \forall (i,j) \in \Omega \times \Omega : \quad \text{symétrie ;} \\ s(i,i) &= S > 0, \forall i \in \Omega \ : \ \text{ressemblance d'un individu avec lui-même ;} \\ s(i,j) &\leq S, \forall (i,j) \in \Omega \times \Omega : \text{la ressemblance est majorée par } S. \end{aligned}$$

Un indice de ressemblance normé *s** est facilement défini à partir de *s* par :

$$s^*(i,j) = \frac{1}{S} s(i,j), \forall (i,j) \in \Omega \times \Omega ;$$

*s** est une application de Ω * Ω dans [0; 1].

II.2.2. Indice de dissemblance, ou dis similarité :

Une dis similarité est une application *d* de Ω * Ω dans *R*+ vérifiant :

$$\begin{aligned} \forall (i,j) &\in \Omega \times \Omega \\ d(i,j) &= d(j,i), : \quad \text{symétrie ;} \\ d(i,j) = 0 &\Leftrightarrow i = j. \end{aligned}$$

Les notions de similarité et dis similarité se correspondent de façon élémentaire. Si *s* est un indice de ressemblance, alors

$$d(i,j) = S - s(i,j), \forall (i,j) \in \Omega \times \Omega$$

Est un indice de dissemblance. De façon réciproque, si d est un indice de dissemblance avec

$D = \sup_{(i,j) \in \Omega \times \Omega} d(i,j)$, alors $s(i,j) = D - d(i,j)$ est un indice de ressemblance. Comme *s**, un indice de dissemblance normé est défini par :

$$d^*(i,j) = \frac{1}{D} d(i,j), \forall (i,j) \in \Omega \times \Omega$$

Avec *d* = 1 – s* et s* = 1 – d**. Du fait de cette correspondance immédiate, seule la notion de dissemblance, ou dis similarité, normée est considérée par la suite.

II.2.3. Distance :

Une distance sur Ω est, par définition, une dis similarité vérifiant en plus la propriété d'*inégalité triangulaire*. Autrement dit, une distance *d* est une application Ω * Ω dans *R*+ vérifiant [29] :

$$\begin{aligned} d(i,j) &= d(j,i), \ \forall (i,j) \in \Omega \times \Omega ; \\ d(i,i) &= 0 \Longleftrightarrow i = j ; \\ d(i,j) &\leq d(i,k) + d(j,k), \ \forall (i,j,k) \in \Omega^3. \end{aligned}$$

Si Ω est fini, la distance peut être normée.

Une mesure générale de la distance dans le cas des données numériques est donnée par l'indice de *Mikowski* défini de la manière suivante : $\forall\, x, y \ \in I$

$$d(x,y) = [\sum_{i=1}^{p} | x_i - y_i |^q]^{\frac{1}{q}}$$

Les trois distances suivantes *d1*, *d2* et *d3* sont des cas particuliers de l'indice de *Mikowski* :

$$d_1(x,y) = \sqrt{\sum_{i=1}^{p} (x_i - y_i)^2}$$

$$d_2(x,y) = \sum_{i=1}^{p} | x_i - y_i |$$

$$d_3(x,y) = \max\{| x_i - y_i |\} \qquad i = 1, \ldots\ldots, p$$

1. $q = 1 \rightarrow d = d2 \left(distance\ Manhattan(city - block)\right)$
2. $q = 2 \rightarrow d = d1\ (distance\ Euclidienne)$
3. $q = \infty \rightarrow d = d3\ (distance\ Tchebycev\ )$

III. Les types des méthodes de classification :

Il a été possible de les regrouper sous la forme d'une hiérarchie de méthodes appelée taxonomie. Nous présentons ci-après une taxonomie dérivée de celle de Jain et Dubes.[30]

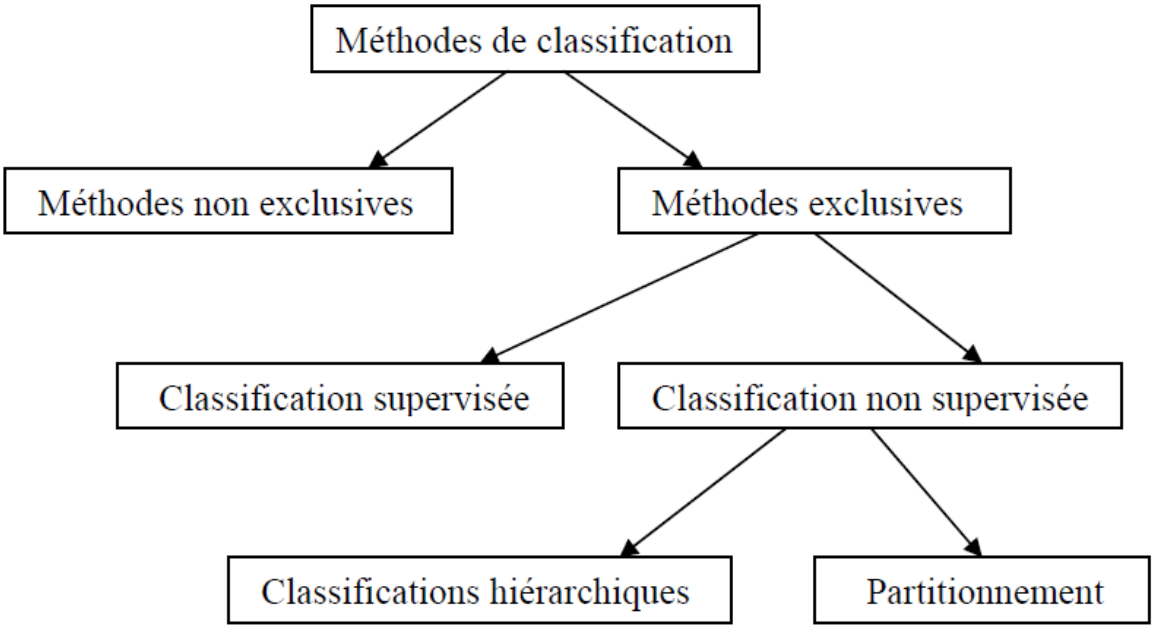


Figure II.2. Les méthodes de classification [31]

- **Classification exclusive/non exclusive :** Une classification exclusive est un partitionnement des objets : un objet n'appartient qu'à une classe et une seule. Au contraire, une classification non exclusive autorise qu'un objet appartienne à plusieurs classes simultanément. Les classes peuvent alors se recouvrir, on parle de la classification floue.

- **Classification supervisée / non supervisée :** Dans la classification non supervisée, aucune information n'est fournie à la méthode (les objets ne sont pas étiquetés), ici on cherche à découvrir de nouveaux groupes d'objets. En classification supervisée, les objets sont étiquetés, il s'agit d'utiliser les groupes connus pour classer les nouveaux objets.

 Classification hiérarchique/partitionnement : Une méthode de classification hiérarchique construit une séquence de partitions imbriqués, que l'on visualise par exemple par un dendrogramme (figure II.3), alors qu'un *partitionnement* ne construit qu'une seule partition des données.

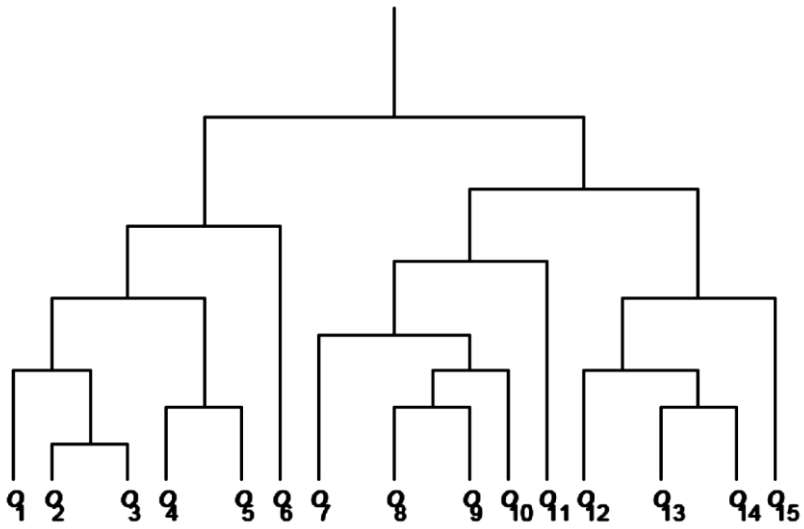


Figure II.3. Exemple d'un dendrogramme.

IV. Classification supervisée et classification non supervisée :

La classification peut être appliquée selon deux motivations différentes. Elle peut être utilisée tout d'abord comme un outil de partitionnement. Dans ce cas le bénéfice de la classification et d'autoriser une étude plus concise de données initiales en essayant de trouver le plus petit nombre de groupes de données qui va le mieux possibles conserver ou mettre en lumière l'information retenue dans ces même données. Les groupes alors formés sont les plus cohérents et homogènes possibles et sont appelés des *clusters*. Dans ce cas, le problème est alors de définir la similarité entre objets. Typiquement, la similarité entre objets est estimée par une fonction calculant la distance entre ces objets. Une fois cette fonction distance définie, la tâche de *clustering* consiste à réduire au maximum la distance entre membres d'un même cluster, tout en augmentant au maximum la distance entre clusters **[**27**]**.

Ces systèmes de classification ne reçoivent aucune information extérieure (provenant d'un opérateur humain par exemple) pour créer des partitions. Cette approche porte en conséquence la dénomination de *classification non supervisée.* De

nombreuses techniques peuvent être envisagées: classification hiérarchique, classification par partition, classification statistique, classification floue (Fuzzy clustering), classification par réseaux de neurones ou classification par algorithmes génétiques.

La deuxième approche nécessite l'intervention d'un système extérieure (qui se compose le plus souvent d'un expert humain) pour étiqueter les informations avant leur classification. Dans ce cas, la classification va apprendre à trouver les groupes de données en fonction des étiquetages précédents et va donc s'entraîner à reconnaître les données. L'objectif de cette démarche est ensuite de pouvoir fournir au système entraîné des données non étiquetées pour qu'il puisse continuer à les répartir dans les groupes adéquats sans autre intervention humaine. On parle alors d'une approche *d'apprentissage supervisé*. On peut citer comme méthodes de classification supervisées toutes les méthodes basées sur un ensemble d'apprentissage tel que la plupart des méthodes du Data Mining (Arbre de décision, Raisonnement à base de cas, Réseaux de neurones…).

V. Le codage de l'information :

A l'origine, l'information obtenue n'est pas forcément manipulable. Pour cette raison, on doit apporter quelques modifications (formaliser et unifier les données) sur ces informations afin qu'elles soient traitable.

V.1. Les types de données :

En classification, on distingue deux types de données :

- **Les données individus-variables :** elles se présentent sous forme d'un tableau qui présente un ensemble de variables observées sur plusieurs unités statiques (individus). Les types des variables peuvent être quantitatifs, c'est-à-dire qu'elles prennent des valeurs numériques ou bien qualitatives, dans ce cas, elles prennent des valeurs symboliques qui déterminent des catégories.

- **Les données de proximité :** elles présentent les distances ou les similitudes entre les éléments d'un groupe, c'est une matrice symétrique par rapport à la diagonale.

V.2. Notion de similarité sur les variables :

La similarité comme elle est définie précédemment est un facteur qui se mesure pour chaque couple de données et varie entre 0 et 1, il est égal à 1 si les données sont similaires et 0 sinon. Et cela selon le type :

- Les variables disjonctives : elle est égale à 1 si les deux objets présentent la même caractéristique.

- Les variables qualitatives : elle est égale à 1 si les deux objets présentent la même variable (couleur, sexe,…).

- Les variables quantitatives : elle mesure l'écart entre les deux variables de manière relative à l'intervalle des variables.

V.3. La relation entre la similarité et la distance :

La notion de similarité est complémentaire à la notion de distance, comme le montre la formule suivante :

Distance (A, B) = 1 – Similarité (A, B).

Deux objets similaires veulent dire que la distance qui les sépare est nulle.

VI. Méthodes de classification :

Formalisation générale du problème de la classification :

Soit *I* un ensemble des individus (objets) *I= {w1, w2,..., wn}*. A partir de critères donnés, le problème consiste à réaliser une partition de *P= {C1, C2,..., Ck}* tell que :

$$C_1 \cup C_2 \ \ldots \ \cup C_k = \mathcal{P}$$
$$C_i \cap C_j = \phi \text{ avec } i, j = 1, 2, \ldots, k$$

Suivant la nature du problème posé, on se fixe à l'avance ou non le nombre *K* des classes *Ci* désirées.

VI.1. Les algorithmes de clustering :

On se limitera ici aux deux techniques de classification non supervisée, et dans chaque type on va présenter seulement un algorithme :

- Le clustering hiérarchique : la classification ascendante hiérarchique (*CAH*).
- Le clustering par partition : les centres mobiles (*K-Means*).

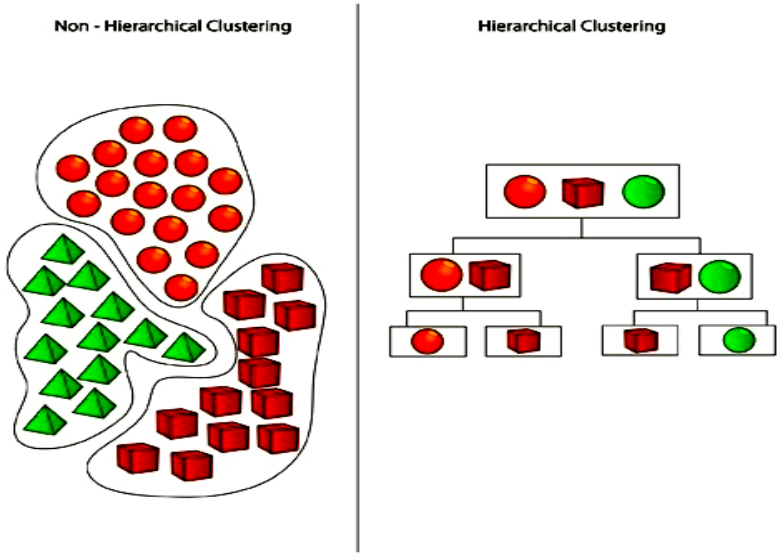


Figure II.4. Les deux types de clustering non-hiérarchique/hiérarchique.

VI.2. Les centres mobiles : (K-Means)

L'algorithme des centres mobiles est un algorithme de partitionnement itératif connu sous le nom de *K-Means*. Il a été introduit par MacQueen en 1967. Cet algorithme peut être utilisé sur des jeux de données volumineux. L'algorithme K_Means utilise l'erreur quadratique comme critère d'évaluation des partitions. En premier temps les objets sont regroupés autour de K centres arbitraires en affectant chacun des objets au centre de gravité le plus proche. On calcule les nouveaux centres de gravité et on refait l'opération d'affectation jusqu'à ce que les objets se stabilisent [27]. La figure suivante donne un exemple d'une partition associée à trois centres dans le plan.

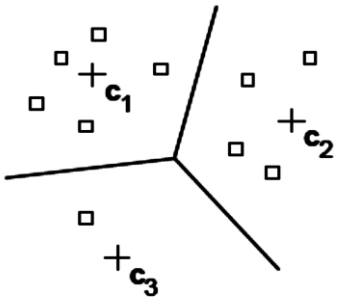


Figure II.5. Exemple de partition obtenue par les centres mobiles

Algorithme : [32]

(1) **Tantque** l'inertie intraclasse ne s'est pas stabilisée **faire**

(2) Générer une nouvelle partition *P'* en affectant chaque objet à la classe dont le centre est le plus proche.

(3) Calculer les centres de gravité des classes de la nouvelle partition *P'*.

(4) $P \leftarrow P'$.

(5) **Fin Tantque**

(6) **retourner***P*.

Principe à travers un exemple :

4 types de médicaments avec chacun deux modalités, la concentration et l'efficacité, on veut créer deux classes => K=2 (voir la figure ci-dessous)

Médicament	Concentration	Efficacité
A	1	1
B	2	1
C	4	3
D	5	4

Figure II.6. Exemple *K-Means*

- ✓ **Etape 1 :** On désigne aléatoirement A et B comme centre de classes. *C1* = A *C2* = B.
- ✓ **Etape 2 :** On assigne chaque point à une des classes. On commence par D :

$$d(D,c_1) = \sqrt{(5-1)^2 + (4-1)^2} = 5$$
$$d(D,c_2) = \sqrt{(5-2)^2 + (4-1)^2} = 4.24$$

- ✓ **Etape 3 :** Calcul les nouveaux centres de classe compte tenu de la nouvelle classification.

$$c_1 = (1,\ 1)$$
$$c_2 = \left(\frac{2+4+5}{3},\ \frac{1+3+4}{3}\right) = (11/3,\ 8/3) = (3.67,\ 2.67)$$

=> *C1* = (1 , 1) et *C2* = (3.67 , 2.67)

On commence la deuxième itération de l'algorithme, on réassigne chaque médicament à une classe en calculant la distance les séparant des nouveaux centres de classe, on repart à l'étape 2, on répète les étapes jusqu'à convergence.

Le résultat final est donc :

- Classe1 = {A, B} avec comme centre de classe *c1* = (1.5, 1).
- Classe2 = {C, D} avec comme centre de classe *c2* = (4.5, 3.5).

Les étapes sont illustrées par la figure ci-dessous

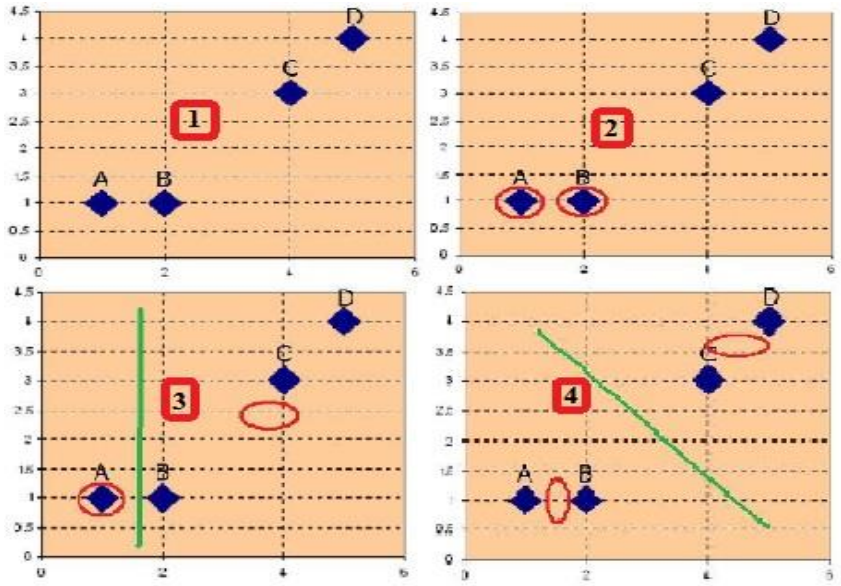


Figure II.7. Les étapes de l'algorithme des centres mobiles

Discussion :

Cette méthode est la plus populaire des méthodes de clustering, malgré ça, un de ses problèmes majeur est qu'il tend à trouver des classes sphériques de même taille. En plus K-means est connu par sa complexité de « NP-difficile ». Il est donc fréquemment faire appeler une heuristique en pratique, ce qui explique qu'elle est convergente et surtout avantageuse du point de vue calcul mais elle dépend essentiellement de la partition initiale (des initialisations différentes peuvent mener à des clusters différents «problèmes de minima locaux ») cela risque d'obtenir une partition qui ne soit pas optimale pourtant qu'elle donne surement une partition meilleure que la partition initiale. De plus, la définition de la classe se fait à partir de son centre, qui pourrait ne pas être un individu de l'ensemble à classer, d'où le risque d'obtenir des classes vides.

VI.3. Méthode hiérarchique : (CAH)

Le processus basique des méthodes hiérarchiques a été donné par [33] [34], ce type de clustering consiste à effectuer une suite de regroupements en Clusters de moins en moins fines en agrégeant à chaque étape les objets (simple élément) ou les groupes d'objets les plus proches. Ce qui nous donne une arborescence de clusters [35].

La méthode CAH suppose qu'on dispose d'une mesure de dis similarité entre les individus, dans le cas de points situés dans un espace euclidien, on peut utiliser la *distance* comme mesure de dis similarité.

La classification ascendante hiérarchique est dite ascendante car elle part d'une situation où tous les individus sont seuls dans une classe, puis sont rassemblés en classes de plus en plus grandes.

Algorithme :

1) **Initialisation** :

   Chaque individu est placé dans son propre cluster, Calcul de la matrice de ressemblance *M* entre chaque couple de clusters (ici les points).

2) **Répéter** :

   - Sélection dans *M* des deux clusters les plus proches *Ci* et *Cj*.
   - Fusion de *Ci* et *Cj* par un cluster *CG* plus général.

- Mise à jour de *M* en calculant la ressemblance entre *CG* et les clusters existants Jusqu'à fusionner les 2 derniers clusters.

Dans la figure suivante, on représente une illustration du principe de CAH et la hiérarchie finale obtenue où les liens hiérarchiques apparaissent clairement.

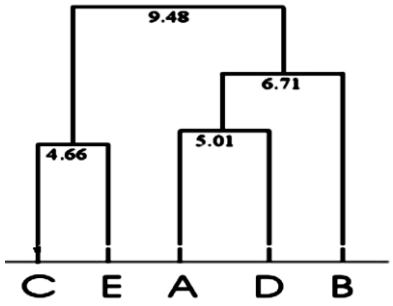


Figure II.8. Le principe de CAH

Les critères d'agrégation :

De nombreux critères ont été proposés, les plus connus sont :

- **Le critère du saut minimal :** la distance entre deux classes *C1* et *C2* est définie par la plus courte distance séparant un individu de *C1* et un individu de *C2*.

$$D(C_1, C_2) = min(\{d(x, y)\}, x \in C_1, y \in C_2)$$

- **Le critère du saut maximal :** la distance entre deux classes *C1* et *C2* est définie par la plus grande distance séparant un individu de *C1* et un individu de *C2*.

$$D(C_1, C_2) = max(\{d(x, y)\}, x \in C_1, y \in C_2)$$

- **Le critère de la moyenne :** ce critère consiste à calculer la distance moyenne entre tous les éléments de *C1* et tous les éléments de *C2*.

$$D(C_1, C_2) = \frac{1}{n_{C_1} n_{C_2}} \sum_{x \in C_1} \sum_{y \in C_2} d(x, y)$$

- **Le critère de Ward :** ce critère ne s'applique que si on est muni d'un espace euclidien la dis similarité *p* entre 2 individus doit être égale à la

moitié du carré de la distance euclidienne *d*. Ce critère consiste à choisir à chaque étape le regroupement de classes tel que l'augmentation de l'inertie intra-classe soit minimal.

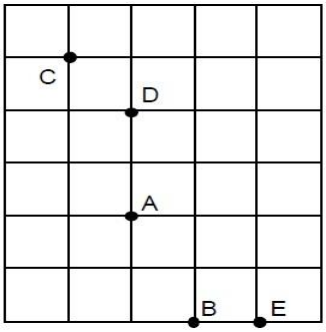
$$D(C_1, C_2) = \frac{n_{C_1} n_{C_2}}{n_{C_1} + n_{C_2}} d^2(g_{C_1}, g_{C_2})$$

La difficulté du choix du critère réside dans le fait que ces critères peuvent déboucher sur des résultats différents. Selon les plus parts des références le critère le plus couramment utilisé est celui du Ward.

Principe dans un exemple :

Considérons les 5 points suivants dans ***R*R*** : A=(2,2), B=(3,0), C=(1,5), D=(2,4), E=(4,0). On désire effectuer une classification CAH en utilisant la distance euclidienne comme mesure de dis similarité entre les points et la stratégie d'agrégation du saut maximal.

L'initialisation est illustrée par la figure ci-dessous :

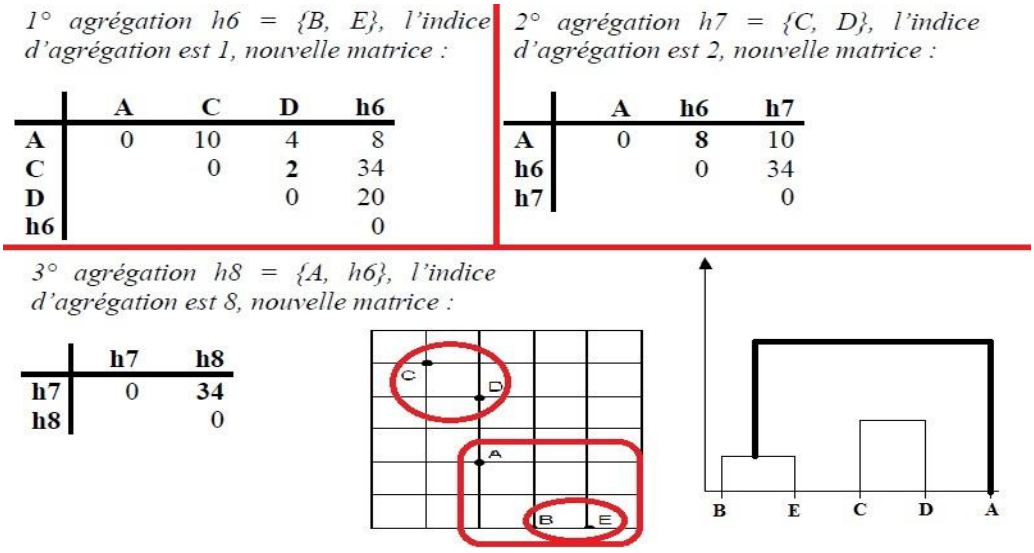


Matrice de dissimilarité initiale :

	A	B	C	D	E
A	0	5	10	4	8
B		0	29	17	**1**
C			0	2	34
D				0	20
E					0

Figure II.9. L'initialisation dans CAH

Les étapes sont illustrées par la figure ci-dessous :



	A	C	D	h6
A	0	10	4	8
C		0	**2**	34
D			0	20
h6				0

	A	h6	h7
A	0	**8**	10
h6		0	34
h7			0

	h7	h8
h7	0	**34**
h8		0

Figure II.10. Les étapes de l'algorithme de CAH

Discussion :

La CAH ne nécessite pas de connaître le nombre de clusters a priori. De plus, il n'y a pas de fonction d'initialisation, ainsi une seule construction d'un cluster (équivalent à une itération pour les méthodes de partitionnement).

En ce qui concerne généralement les méthodes hiérarchiques le problème qu'on peut rencontrer réside dans la sélection d'une ultra-métrique (distance pour calculer la similarité entre clusters) soit la plus proche de la métrique utilisée pour les individus, car ces méthodes sont heuristiques, pour cela il y a plusieurs techniques permet de le faire : Saut minimal (single linkage) ; Saut maximal (complete linkage) ; Saut moyen ; Barycentre...

Une autre faiblesse est : la complexité de temps d'au moins *O(n2)*, où *n* est le nombre d'objets au total.

Il est difficile parfois d'apporter une justification aux méthodes hiérarchique (CAH, CDH..), Cependant, dans [36], une interprétation probabiliste de la CAH, basée sur une estimation par maximum de vraisemblance des modèles de mélange, est proposée comme solution pour mieux interpréter les résultats.

Afin d'améliorer la qualité d'une classification hiérarchique, on peut profiter de deux techniques

- Analyser attentivement les liens entre objets à chaque étape [37] et [38].

- Améliorer la partition obtenue avec une méthode de deuxième type de clustering (partitionnement) [39].

VII. Conclusion :

Deux méthodes sont proposées pour le problème général de la classification non supervisée, ils diffèrent par les mesures de proximité qu'ils utilisent, la nature des données qu'ils traitent et l'objectif final de la classification. Chacune de ces méthodes possède ses points forts et ses points faibles.

C'est donc, le choix d'une méthode approprié dépend fortement de l'application, la nature des données et les ressources disponibles. Une analyse attentive des données aide à bien choisir le meilleur algorithme. Il n'existe pas un algorithme qui peut répondre à toutes les demandes.

Pour cela, le chapitre suivant sera consacré à la construction d'un IDS hybride non seulement détectant toute utilisation malveillante du système mais aussi génère le minimum de faux positifs et faux négatif en utilisant l'algorithme CAH.

Chapitre 3

Implémentation de l'Application Web

I. Introduction :

Ce chapitre présente notre contribution dans le cadre de ce PFE à travers la description de la solution que nous proposons pour sécuriser une application web. Ce travail comprend trois étapes. La première étape consiste à décrire la réalisation en détail de cette application Web (boutique en ligne) en utilisant le paradigme MVC 2 à travers la version 2 du Framework Struts. La deuxième étape consiste à sécuriser l'application Web réalisée on se basant sur deux approches des IDS :

- L'approche comportementale optimisée par l'algorithme CAH (dont l'objectif est de diminuer le nombre des *faux-positifs*).
- L'approche par signatures.

Enfin la dernière étape qui consiste à faire une hybridation entre les deux approches afin de diminuer au maximum le nombre des *faux-alertes*.

II. Outils de réalisation :

Cette partie présente les principaux outils utilisés pour la mise en place de l'application. La réalisation de cette dernière a été faite sous la plateforme Java on se basant sur le Framework Struts 2 avec l'utilisation de MYSQL comme serveur de base de données.

II.1. Langage de programmation java :

Les modules conçus ont été réalisés sous Java dont les principales vertus, sont résumées dans les points suivants [1] :

- Java est un langage simple : plus simple que le C ou le C++ car on lui a retiré les caractéristiques peu utilisées ou difficile à utiliser.
- Java est un langage orienté objet : une classe contient des attributs et des méthodes (principe de l'encapsulation).
- Java est un langage distribué : il est facile de mettre en place une architecture *Client-Serveur* pour travailler avec des fichiers situés sur un ordinateur distant.

- Java est un langage multithread : pour l'exécution simultanée de plusieurs processus. Java est fourni avec un jeu de primitives qui facilitent l'écriture de ce genre de programmes.
- Java est langage robuste : le typage des données est très strict, tant à la compilation qu'à l'exécution. Les pointeurs utilisés dans Java ne sont pas accessibles au programmeur.
- Java est un langage dynamique : les classes de Java peuvent être modifiées sans modification du programme qui les utilise.
- Java est un langage portable : le compilateur java fabrique du *byte-code* "universel". Pour l'exécuter sur une machine quelconque.
- Sécurité : la sécurité dans Java est un aspect primordial, il ne supprime pas tous les problèmes de sécurité mais les réduits fortement.

Nous avons utilisé un éditeur de Java appelé NetBeans version 7.0.1 qui est placé en open source par Sun sous licence CDDL (*Common Development and Distribution License*). En plus de Java, NetBeans permet également de supporter différents langages, comme Python, C, C++, XML et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

II.2. Choix du Framework Struts 2 :

Le Framework Web Apache Struts est un logiciel gratuit open-source pour créer des applications Web Java basées sur JSP (*Java Server Pages*).

Struts 2 implémente le modèle d'architecture dit MVC 2 (Modèle-Vue-Contrôleur), il permet notamment de séparer la partie modèle (programmation, traitement des informations) de la partie présentation (affichage). Le modèle englobe la logique métier et les données sur lesquelles il opère, la vue représente le code de conception de pages et le contrôleur représente le code de navigation (servlet unique).

La Figure III.1 schématise le cycle de vie de Struts 2 :

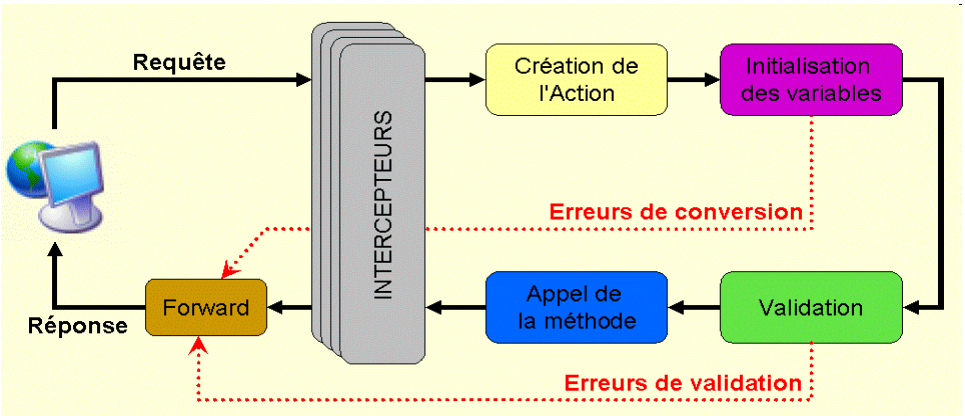


Figure III.1. Cycle de vie de Struts2 [2]

Le cycle de vie d'un client dans Struts 2 se déroule comme suit :

- Une fois le **« FilterDispatcher »** passé, la requête est soumise aux intercepteurs. Ces derniers ont pour rôle d'effectuer des pré/post traitements sur la requête (gestion des exceptions, upload de fichier, ...).
- L'action est instanciée, peuplée puis le formulaire est validé.
- Invocation de la méthode contenant la logique de l'action (**execute()**).
- Délégation de l'affichage à la vue.

II.3. Choix de MySQL :

MySQL est un serveur de bases de données relationnelles open-source qui stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Le SQL (*Structured Query Language*) : le langage standard pour les traitements de bases de données [3].

On a choisi WampServer comme outil pour créer la base de données qui constitue le langage intermédiaire entre cette base et l'utilisateur de la base.

La Figure III.2 illustre notre base de données :

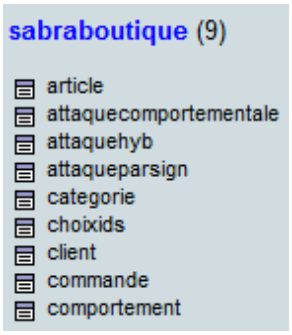


Figure III.2. BDD SabraBoutique.

III. Réalisation de l'application Web :

III.1. Description de boutique en ligne :

L'application SabraBoutique propose une plate-forme de vente en ligne ainsi que les services suivants :

- Gestion des clients (création, modification, suppression, authentification).
- Gestion des articles (création, modification, suppression, tri...).
- Service de panier dynamique (création, modification, suppression, session, listes...).
- Gestion des catégories (création, modification, suppression, tri...).
- Le paiement sécurisé en ligne (*accounting*) est souvent assuré par un tiers de confiance (une banque) via une transaction sécurisée.
- Le suivi de la livraison.

Notre application Web possède trois modes d'accès :

1. L'accès public de type *visiteur* ne nécessite pas d'authentification.
2. L'accès public de type *client* nécessite une authentification.
3. L'accès privé de type *administrateur* nécessite une authentification

La Figure III.3 présente les cas d'utilisation :

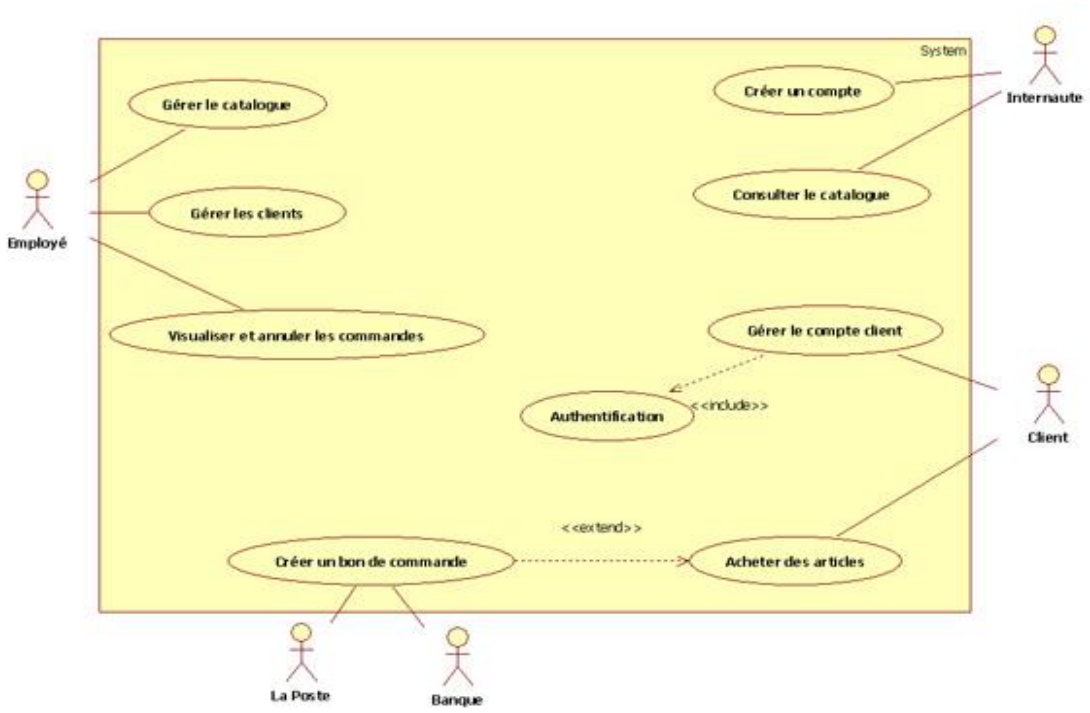


Figure III.3. Le diagramme des cas d'utilisation.

Du point de vue design, La Figure III.4 présente la page d'accueil de notre application Web :

Figure III.4. La page d'accueil de SabraBoutique.

III.2. Services proposés :

- ✓ **Catalogue des articles :** cette page permet d'afficher la liste paginée des produits par catégorie (voir la figure III.5).

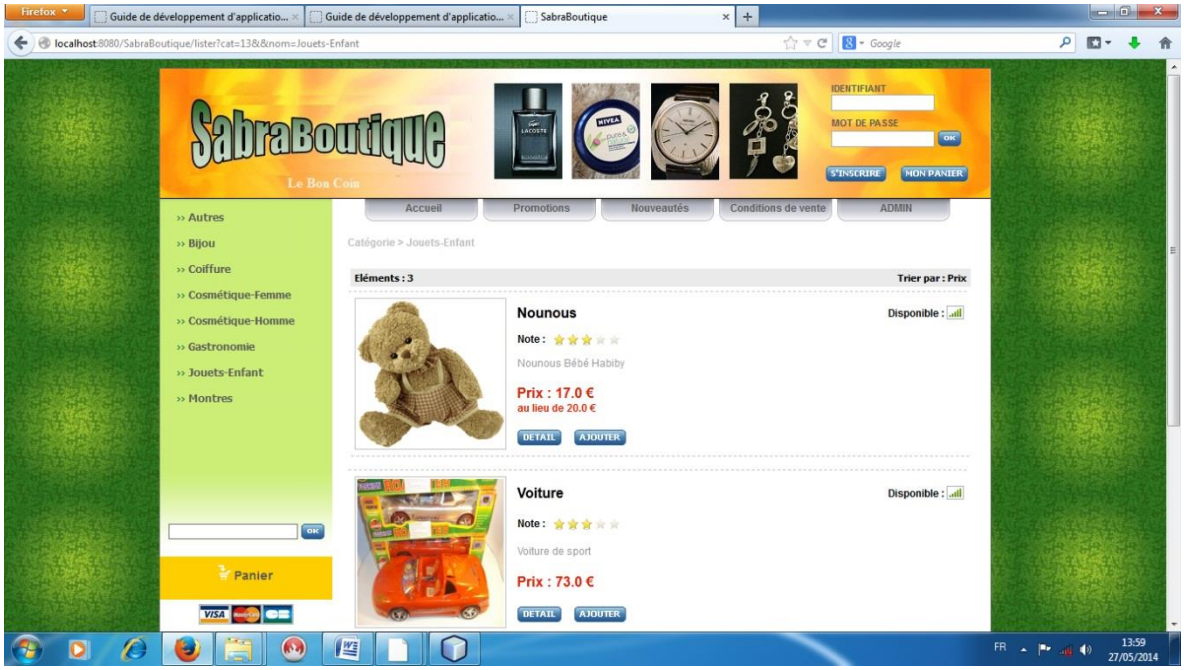


Figure III.5. Catalogue des articles.

- ✓ **Fiche article :** cette page permet d'afficher le détail du produit avec le nom, la note, la description, le prix et un bouton pour ajouter cet article au panier (voir la figure III.6).

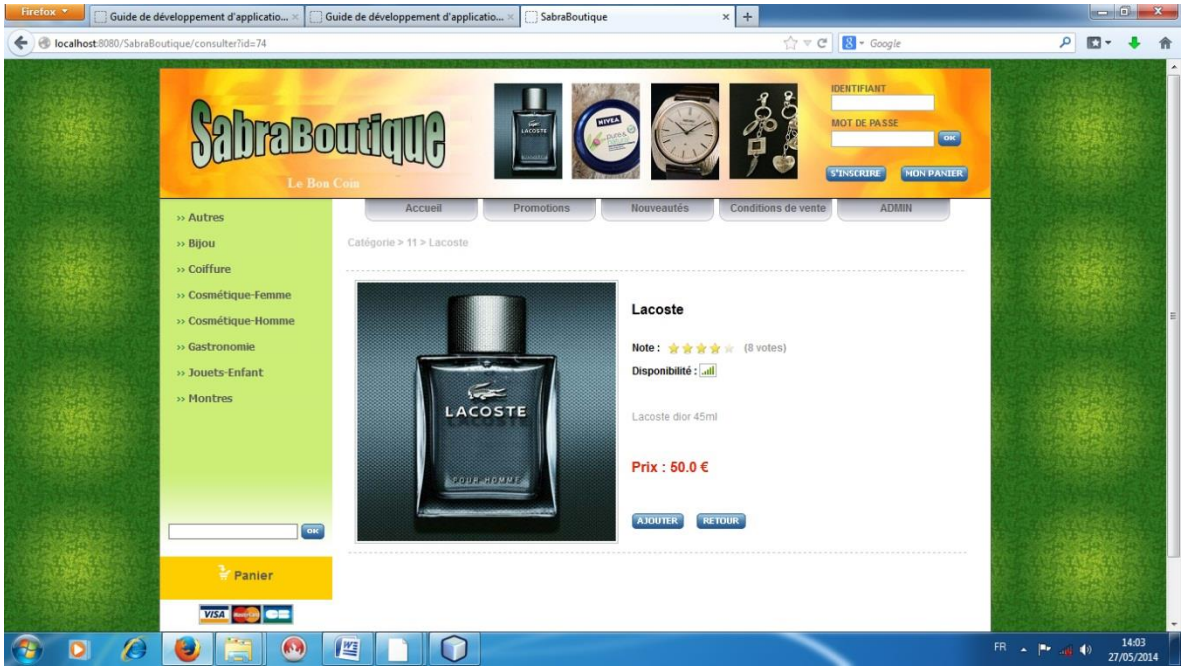


Figure III.6. Fiche article.

- ✓ **Rechercher un article :** Le formulaire de saisie présente en haut à droite du site permet de rechercher un article à partir de son nom ou de sa description (voir la figure III.8).

- ✓ **Authentification :** Cette page permet au client de se connecter et de se déconnecter du système. Le client doit avoir créé un compte auparavant (voir la figure III.8).

- ✓ **Créer un compte :** Cette page permet au visiteur de se créer un compte dans le système et de devenir ainsi un client. Pour créer un compte, le visiteur est alors invité à compléter ses informations personnelles (nom, prénom, email, adresse...) (voir la figure III.7)

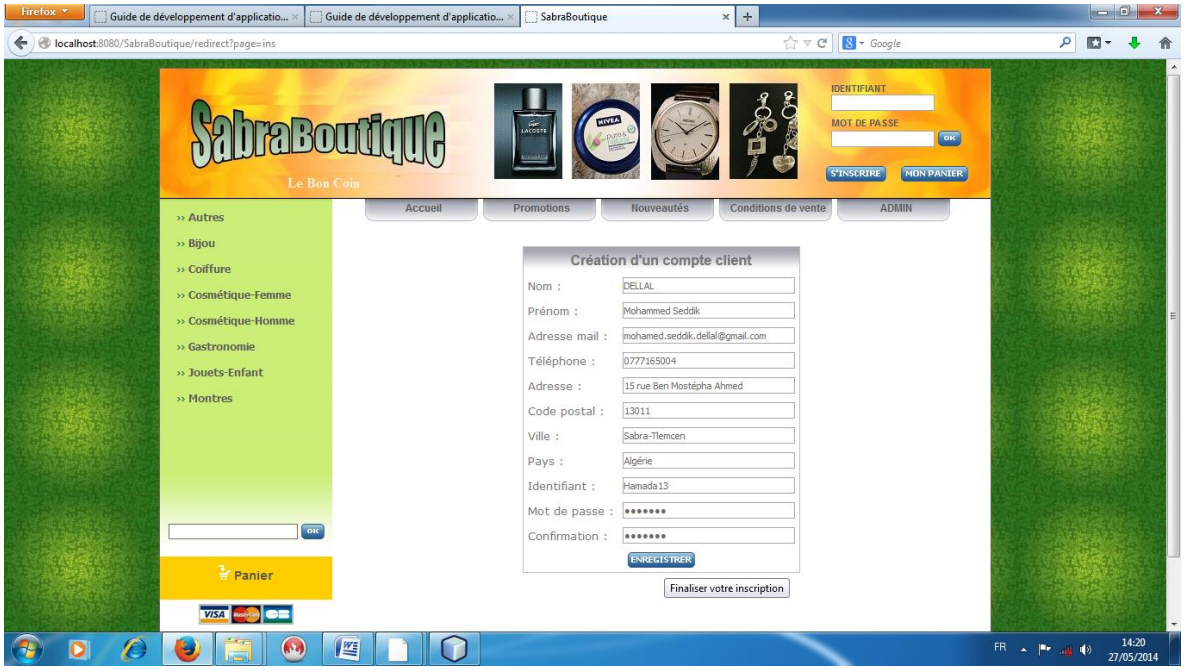


Figure III.7. Création d'un compte client.

- ✓ **Gérer le compte client :** Chaque client peut consulter et mettre à jour ses informations personnelles au sein du système. Il suffit juste de cliquer sur le lien Mon compte (après authentification) (voir la figure III.8).

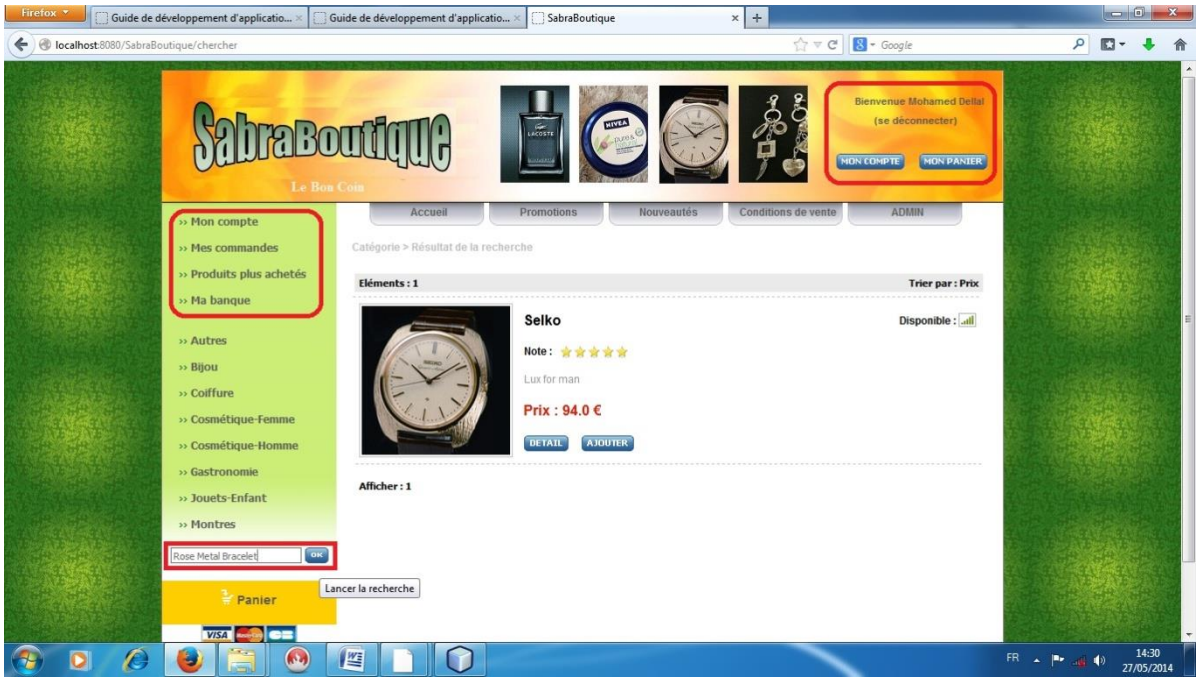


Figure III.8. Authentification, Recherche et Gestion du compte.

- ✓ **Acheter des articles :** Les clients (visiteurs authentifiés) peuvent acheter des articles dans le système. Lorsqu'un client est intéressé par un article, il clique sur le lien adapté pour l'ajouter à son panier. Le client a ensuite la possibilité de modifier la quantité désirée pour chaque article ou supprimer un ou plusieurs articles. Le client peut visualiser le contenu de son panier pendant toutes la durée de sa session quand bon lui semble. Lorsque le client est satisfait et qu'il souhaite valider sa commande, le système récupère l'identifiant de sa carte bancaire ainsi que son adresse de livraison. Une fois toutes les données validées, un bon de commande est créé et le panier électronique est automatiquement vidé (voir la figure III.9).

Figure III.9. Caddie virtuel.

- ✓ **Interface ADMIN :** Les administrateurs de la société SabraBoutique peuvent visualiser et supprimer les commandes, les articles, les catégories, les clients, le contenu du stock dans le système (voir la figure III.10).

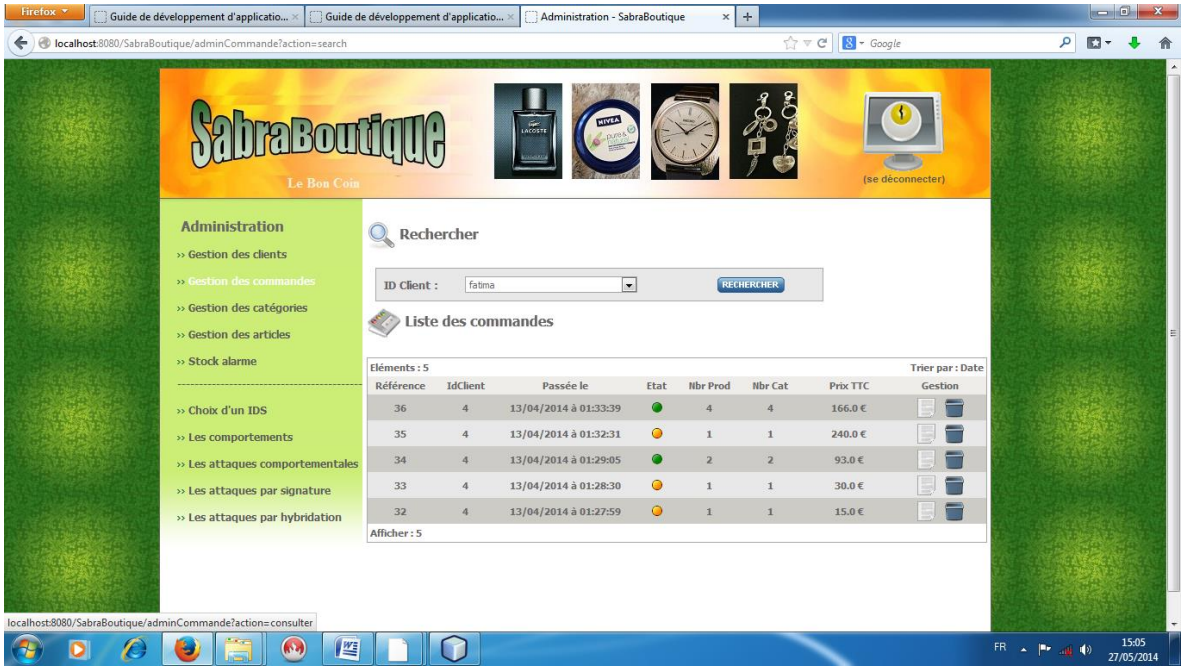


Figure III.10. Interface ADMIN.

IV. Sécurisé l'application Web :

Afin d'éviter les utilisations malveillantes de l'application Web réalisée, il faut développer un système capable de détecter et d'identifier les intrusions. Ce système doit pouvoir s'adapter de manière autonome pour intégrer dynamiquement la détection de nouvelles intrusions en se basant sur trois approches : l'approche comportementale optimisée par l'algorithme de classification CAH, l'approche par signature et l'approche hybride.

IV.1. L'approche comportementale :

Cette technique comporte deux phases :

IV.1.1. Phase d'analyse :

Cette étape déroule pendant *n* connexions (*n* est un entier choisi par l'administrateur de la sécurité, de préférence qu'il soit grand), l'objectif de ce délai, est de fixer le comportement normal d'un client. Chaque client donc sera orienté dans une classe à base de son profile. La classification des clients est faite par rapport aux critères suivants : fréquence moyenne d'achat (Nbr-Prod), prix moyen d'achat (Prix-Tot) et Nbr-Cat qui désigne le nombre moyen des catégories.

- ***Nbr-Prod (n) :*** trois choix sont possibles : N3 si Nbr-Prod<=3, N2 si 3<Nbr-Prod<6 ou N1 si Nbr-Prod>=6.
- ***Prix-Tot (p) en euro :*** dépond fortement du Nbr-Prod et du prix moyen (75 euros), on suppose qu'on a deux choix possibles.
- ***Nbr-Cat (c) :*** dépond fortement du Nbr-Prod, on suppose également qu'on a deux choix possibles.

Ces trois critères donnent naissance à douze classes différentes, chaque client va être classé après la phase d'analyse dans la classe correspondante parmi les classes créées, La Figure III.11 montre les différentes classes possibles pour un client.

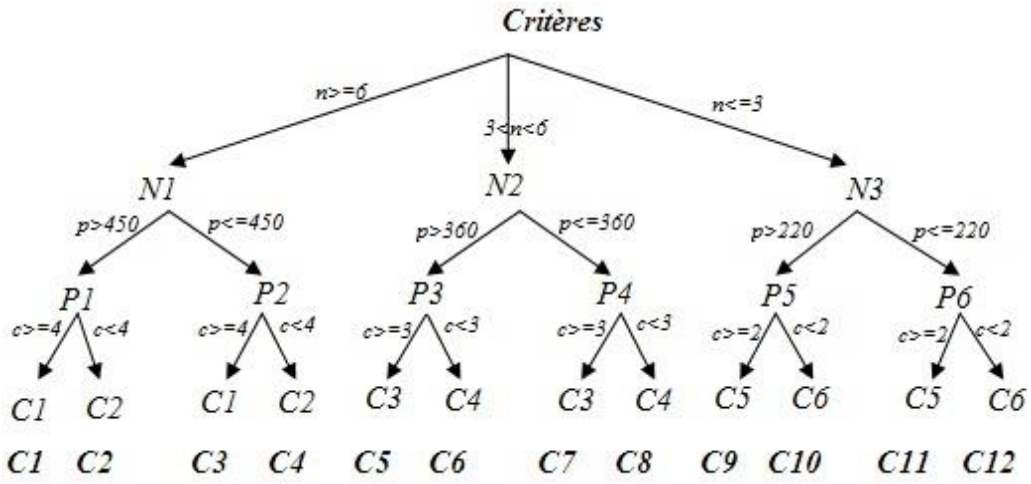


Figure III.11. Classification d'un client en fonctions des 3 critères.

Pour appliquer cette phase on a créé une table nommée « Comportement » pour sauvegarder les valeurs des trois critères définis précédemment (n, p, c).

Avant la classification, la table « Comportement » est vide, dès que le client passe sa première commande, les critères définis précédemment seront initialisés. Par la suite, et pour chaque connexion parmi les n déjà fixées auparavant, les valeurs retenues dans la i ème connexion sont la moyenne (critère par critère) des valeurs des traces actuelles (celles de la i ème connexion) et les valeurs relatives à la connexion précédente (de la (i-1) ème connexion) (voir la figure III.12).

Eléments : 1

Référence	ID-Client	Nbr-Con	Nbr-Prod	Nbr-Cat	Prix-Tot	Classe
42	3	1	3	2	80.0 €	C11

Afficher : 1

Figure III.12. La table Comportement dans la 1ère connexion.

On a ici un client qui a passé sa première commande, il a acheté trois produits de deux catégories différentes avec un prix total de 80 euros, d'après l'arbre des profils illustré dans la figure III.11 ce client va être associé à la classe C11.

Lorsque le même client passe sa 2ème commande où le panier contient cinq produits de deux catégories différentes avec un prix total de 140 euros, la table « Comportement » sera modifiée comme indiqué dans la figure III.13.

Eléments : 1

Référence	ID-Client	Nbr-Con	Nbr-Prod	Nbr-Cat	Prix-Tot	Classe
42	3	2	4	2	110.0 €	C8

Afficher : 1

Figure III.13. La table Comportement dans la 2ème connexion.

- Moy(Nbr-Prod-1,Nbr-Prod-2)=Moy(3,5)=4 donc Nbr-Prod=4.
- Moy(Nbr-Cat-1,Nbr-Cat-2)=Moy(2,2)=2 donc Nbr-Cat=2.
- Moy(Prix-Tot-1,Prix-Tot-2)=Moy(80,140)=110 donc Prix-Tot=110.

D'après les résultats obtenus et la figure III.11, ce client va changer sa classe de C11 vers C8. Ce processus termine lorsque le nombre de connexion est égal à *n*.

La figure III.14 montre le profile final du client précédent où *n*=3.

Eléments : 1

Référence	ID-Client	Nbr-Con	Nbr-Prod	Nbr-Cat	Prix-Tot	Classe
42	3	3	3	2	133.5 €	C11

Afficher : 1

Figure III.14. Le comportement final d'un client.

IV.1.2. Phase de détection :

Cette phase est valable uniquement pour les anciens clients (qui ont été déjà classés et leur Nbr-Con dépasse le *n*) alors si l'un de ces clients vient de se connecter

une nouvelle fois, l'IDS va le suivre pour récupérer ses critères (Nbr-Prod,Nbr-Cat,Prix-Tot) afin d'obtenir son nouveau comportement (sa nouvelle classe), pour enfin mesurer la similarité entre son comportement et son profil déterminé dans la phase d'analyse.

Alors si la nouvelle classe est différente de la classe déterminée, l'IDS considère le changement de classe comme anomalie.

Après la détection d'attaque le système va bloquer le client c'est-à-dire que ce dernier ne peut pas poursuivre ses achats et finir son rôle par l'affichage d'un message d'alerte.

IV.1.3. Les faux positifs :

Le choix de la taille de la fenêtre temporelle qui est dans notre cas le nombre de connexions (*n*) est une importante variable d'ajustement : une fenêtre plus courte limite dans les faits le champ d'action du client, en limitant considérablement ses comportements possibles, ce qui tend à générer de fréquents faux-positifs.

Pour minimiser les faux positifs, on va utiliser l'algorithme de classification CAH (vu dans le chapitre II).

IV.1.4. Implémentation de l'algorithme CAH :

L'entrée de l'algorithme dans ce cas sont des profils des clients (les classes) et l'objectif c'est de regrouper les classes les plus proches en cluster, la méthode CAH suppose qu'on dispose d'une mesure de dis similarité entre les individus, dans notre cas on va prendre *la distance euclidienne* comme mesure de dissemblance entre les classes et le critère de *Ward* comme stratégie d'agrégation.

La question qui se pose, c'est comment convertir les classes en des points situés dans un espace euclidien ? Pour cela, on a proposé de calculer le minimum des trois critères de chaque classe.

Par exemple pour le Nbr-Prod(n), on a supposé que :

- ✓ Si n<=3 : le minimum de n c'est 1.
- ✓ Si 3<n<6 : le minimum de n c'est 4.
- ✓ Si n>=6 : le minimum de n c'est 6.

Pour le Prix-Tot(p) qui dépond fortement du Nbr-Prod et du prix minimum (15 euros), on a supposé que :

- ✓ Si par exemple n=4 : p=4*prix-min => p=4*15 => p=60 et ainsi de suite.

Pour le Nbr-Cat(c) qui dépond fortement du Nbr-Prod, on a supposé que :

- ✓ Si par exemple n=4 et c<3 le minimum de c c'est d'acheter ces 4 produits avec une même catégorie, donc dans ce cas c=1. Si n=4 et c>=3 le minimum de c=3 et ainsi de suite.

Donc d'après cette étape, les classes seront transférées en points suivants :

Classes	Points
C1 (n>=6, p>450, c>=4)	(6, 451, 4)
C2 (n>=6, p>450, c<4)	(6, 451, 1)
C3 (n>=6, p<=450, c>=4)	(6, 180, 4)
C4 (n>=6, p<=450, c<4)	(6, 90, 1)
C5 (3<n<6, p>360, c>=3)	(4, 361, 3)
C6 (3<n<6, p>360, c<3)	(4, 361, 1)
C7 (3<n<6, p<=360, c>=3)	(4, 105, 3)
C8 (3<n<6, p<=360, c<3)	(4, 60, 1)
C9 (n<=3, p>220, c>=2)	(2, 221, 2)
C10 (n<=3, p>220, c<2)	(1, 221, 1)
C11 (n<=3, p<=220, c>=2)	(2, 30, 2)
C12 (n<=3, p<=220, c<2)	(1, 15, 1)

Figure III.15. Tableau de conversation des classes en points.

Après l'implémentation de l'algorithme et le calcul de la matrice de ressemblance *M* entre chaque couple de clusters (ici les classes) (voir l'exemple vu dans le chapitre II), on obtient une hiérarchie finale (voir la figure III.16).

Remarque :

Le choix du nombre de cluster (n) dépond de la distance intra-cluster (qui doit prendre une valeur Max) et la distance inter-cluster (qui doit prendre une valeur min) pour avoir une meilleur classification.

- Pour n=3 : D1 (intra) = Max (147.36,290.76) = 290.76 et D1 (inter) = min (62.6,41.325,90.03) = 41.325.
- Pour n=4 D2 (intra) = Max (147.36,290.76) = 290.76 et D2 (inter) = min (62.6,41.325,2,3) = 2.

Donc Max (D1(intra),D2(intra)) = D2(intra) et min (D1(inter), D2(inter)) = D2(inter) donc n=4.

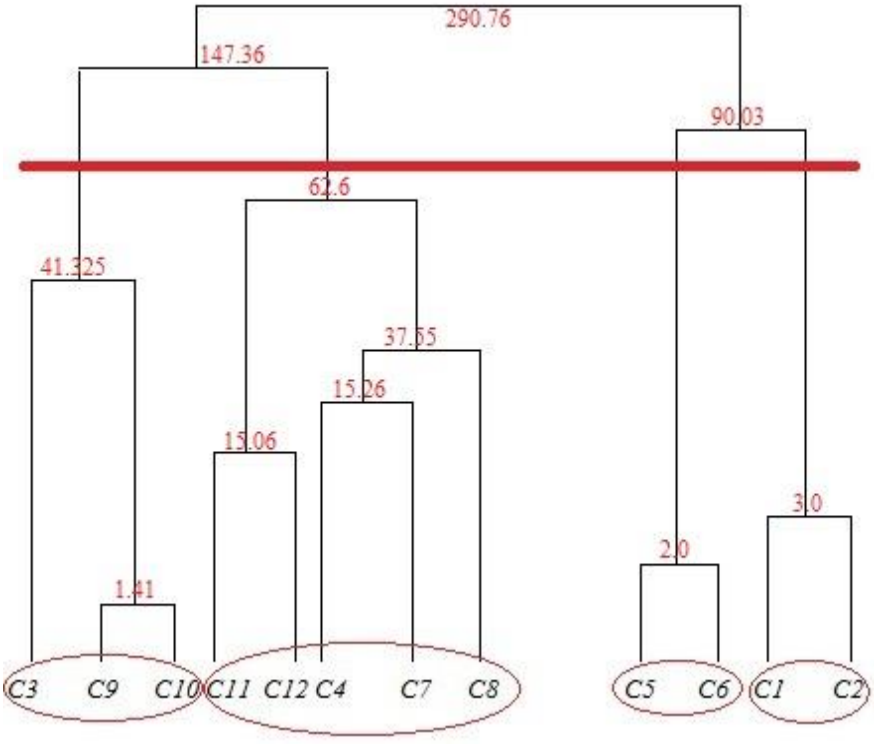


Figure III.16. La hiérarchie finale.

D'après le résultat de l'algorithme de clustering CAH, les clusters sont les suivant :

Clust1= {C3, C9, C10}.

Clust2= {C11, C12, C4, C7, C8}.

Clust3= {C5, C6}.

Clust4= {C1, C2}.

Dans cette partie le principe de l'approche comportementale reste le même, en ajoutant le champ « cluster » à la table Comportement, l'IDS (dans la phase de détection) mesure la similarité entre l'ancien cluster du client et son nouveau cluster. Si le client change sa classe dans le même cluster, aucune alerte n'est déclenchée, mais s'il change de cluster, une alerte sera déclenchée et le client sera bloqué et il finit son rôle d'achats.

La figure III.17 montre le comportement d'un client :

Eléments : 1 Trier par : ID

Référence	ID-Client	Nbr-Con	Nbr-Prod	Nbr-Cat	Prix-Tot	Classe	Cluster
42	3	3	3	2	133.5 €	C11	Clust2

Afficher : 1

Figure III.17. La table comportement contient le champ cluster.

La classe finale de ce client c'est C11 et il appartient au Clust2. Après une nouvelle connexion (la phase de détection), s'il a changé sa classe de C11 vers C4 donc il pourra poursuivre son achat car ces deux classes sont dans le même cluster, sinon (s'il change son cluster) alors une alerte déclenche et ce client ne pourra pas poursuivre son achat (voir la figure III.18).

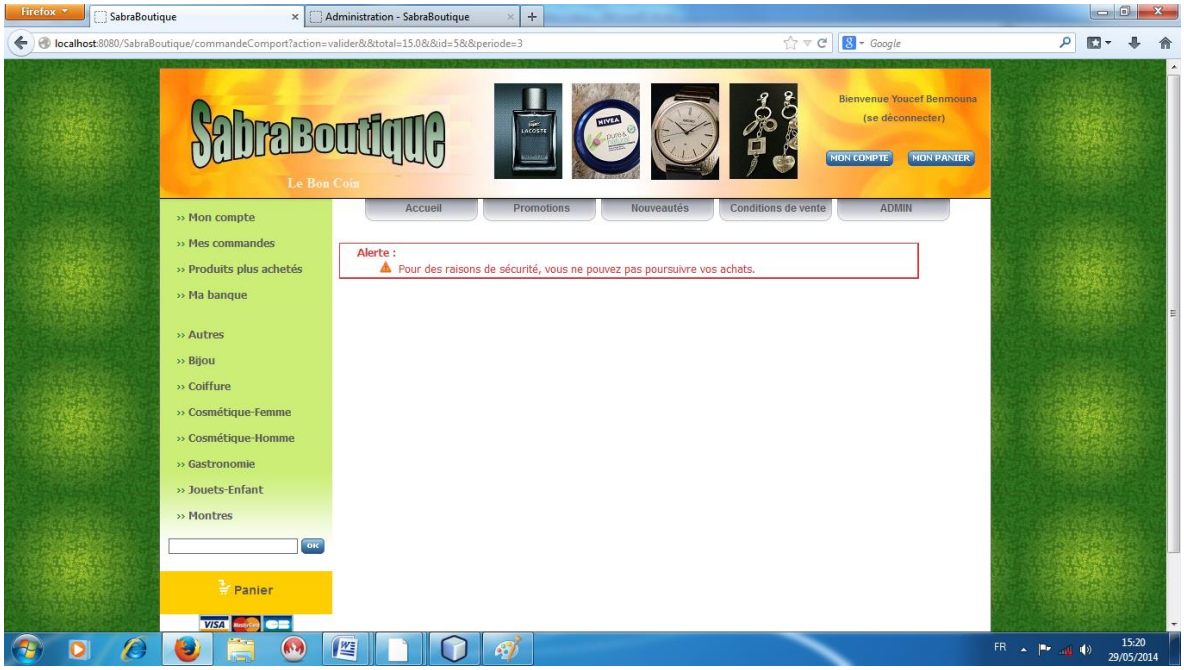


Figure III.18. Message d'alerte corresponds à la détection d'attaque.

La figure III.19 illustre un ensemble d'attaques détectées par notre système de détection:

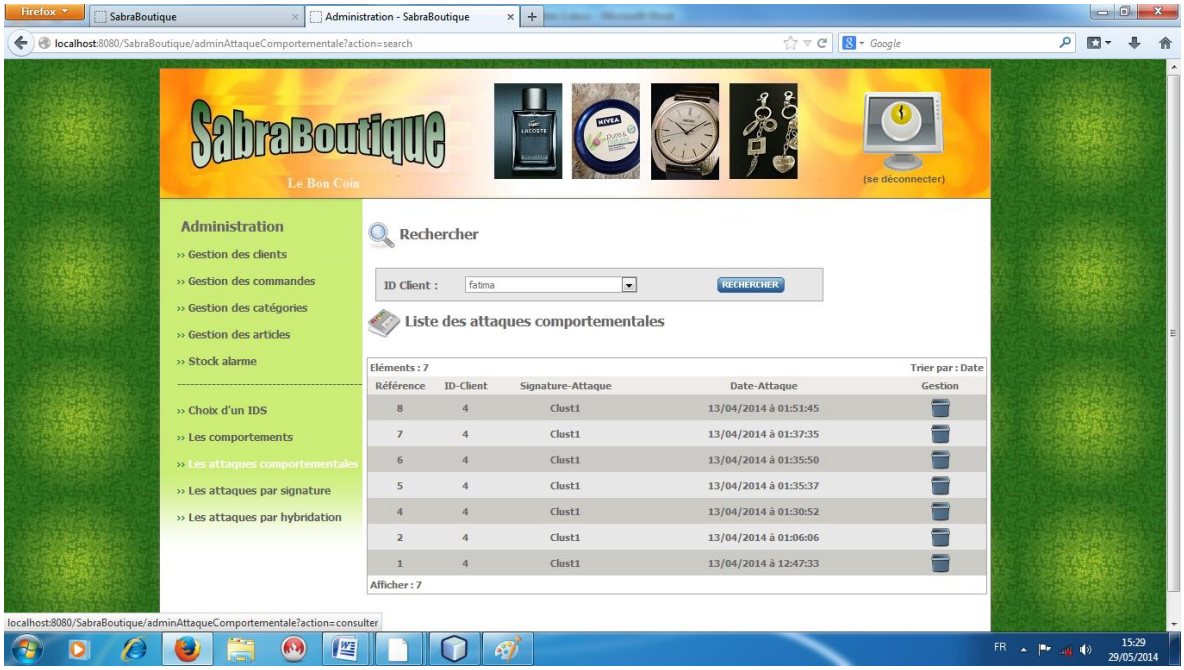


Figure III.19. Liste des attaques comportementales.

IV.1.5. Résultat obtenu avec le CAH :

La figure suivante (Figure III.20) montre la variation de faux positifs entre l'approche comportementale et l'approche comportementale avec l'algorithme CAH.

Pour cela on a utilisé un seuil (distance de fausse alerte). Dans le cas d'une approche comportementale, pour chaque changement de classe (attaque détectée), on mesure la distance entre ces deux classes. Si cette distance est inférieure à ce seuil on peut juger que ce passage est une fausse alerte.

Même principe appliqué au CAH, la distance entre les classes devient une distance entre les clusters.

- Exemple, avec la valeur 100 comme seuil, pour le CAH, nous avons six possibilités de changement de cluster (car on a quatre clusters), un seul changement (Clust3 ⇔ Clust4) est considéré comme fausse alerte. Donc la probabilité qu'une attaque détectée soit une fausse alerte est égale à P1=16.66%. Pour l'approche comportementale, nous avons 66 possibilités de changement de classe (car on a douze classes), parmi elles 21 changements sont considérés comme fausse alerte avec une probabilité de P2=31.81%.
- Si seuil=145, avec le même principe P1=16.66% et P2=45.45%.
- Si seuil=150, avec le même principe P1=33.33% et P2=45.45%.

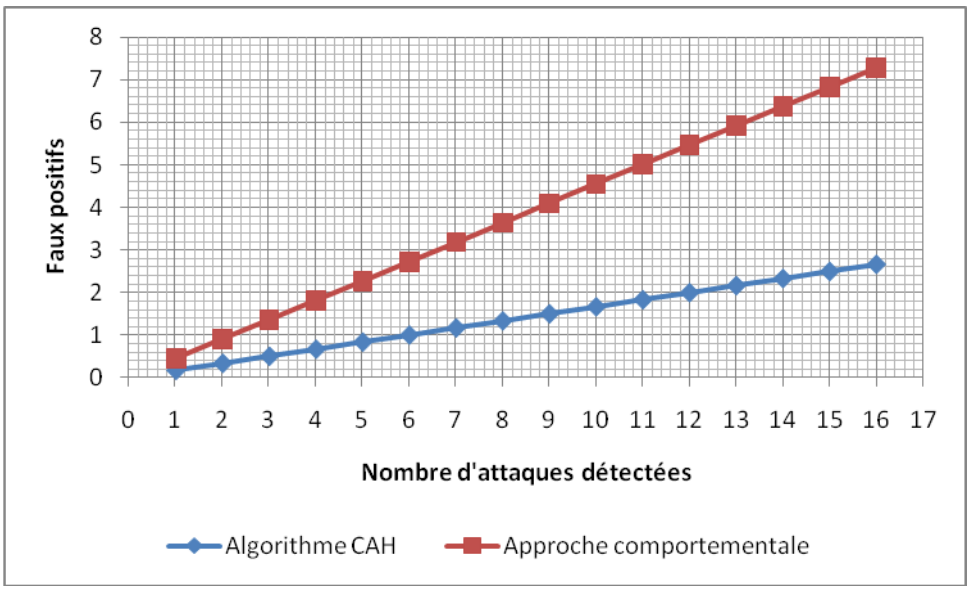


Figure III.20. Variation de faux positifs avec seuil=145.

La courbe montre que l'algorithme CAH permet de détecter plus d'attaques et moins de faux positifs par rapport à l'approche comportementale.

IV.2. L'approche par signature :

Cette méthode de détection s'appuie sur une base de données de toutes les attaques connues. Chaque attaque à sa propre signature, l'IDS cherche à reconnaitre cette signature parmi les utilisateurs qu'il analyse (une recherche de correspondance au sein d'une base de connaissance). Si une attaque est détectée, une alarme est remontée.

Cette méthode est simple à mettre en œuvre, mais elle est basée sur les signatures d'attaques connues. Donc la base de données doit être régulièrement mise à jour ainsi que les attaques inconnues ne seront jamais détectées.

Pour appliquer cette approche on a créé une table nommée « AttaqueParSign » qui contient des attaques connues qui ont été construite par rapport aux critères définis précédemment (Nbr-Prod, Nbr-cat, Prix-Tot) où Nbr-cat=1 (constant).

- ***Nbr-Prod(n) :*** qui prend deux possibilités, N1 si n>=5, N2 si n<5.
- ***Prix-Tot(p) :*** quatre choix sont possibles.

Ces deux critères donnent naissance à quatre signatures (2^2) (voir la figure III.21).

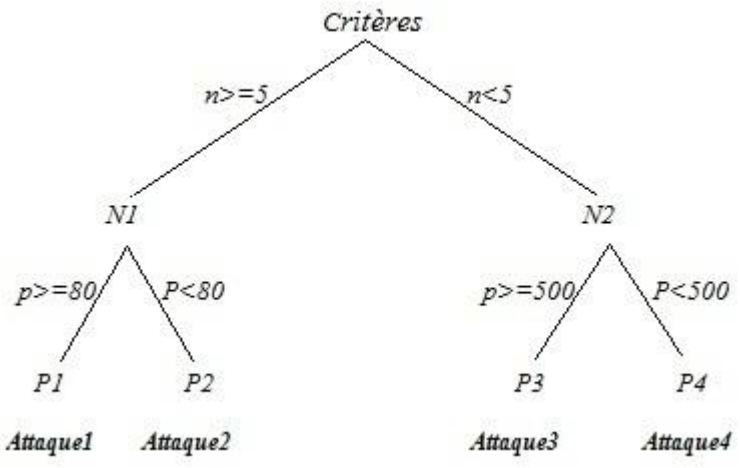


Figure III.21. La base de connaissance.

Le système analyse chaque comportement du client connecté et le compare avec les signatures existantes dans la base de données. Si le système détecte une attaque alors il

va bloquer le client c'est-à-dire que ce dernier ne peut pas poursuivre ses achats et finir son rôle par l'affichage d'un message d'alerte.

La Figure III.22 illustre quelques attaques détectées :

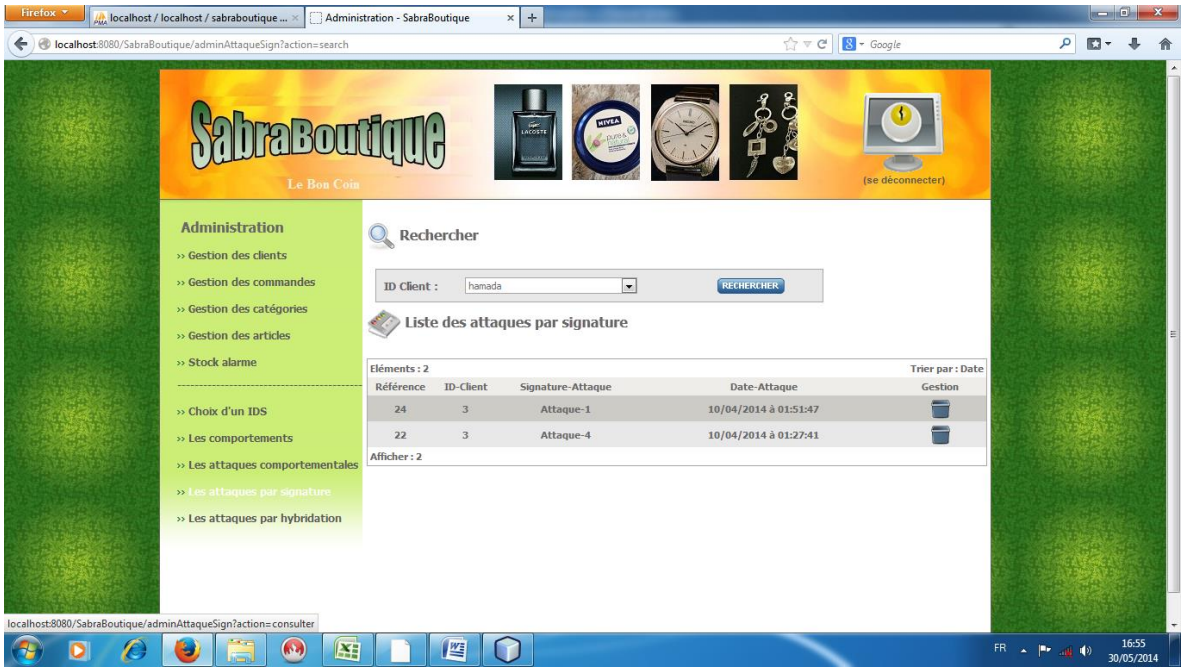


Figure III.22. Liste des attaques par signature.

IV.2.1. Comparaison entre les deux approches :

La Figure III.23 montre le nombre d'attaques détectées (pour les deux approches) par rapport au nombre d'attaques total.

Pour cela, nous avons proposé une liste d'attaques qui contient dix attaques comportementales détectées seulement par le CAH, quatre attaques par signature et deux autres attaques (non reconnues).

Remarque :

Pratiquement les attaques comportementales sont plus fréquentes par rapport aux autres attaques.

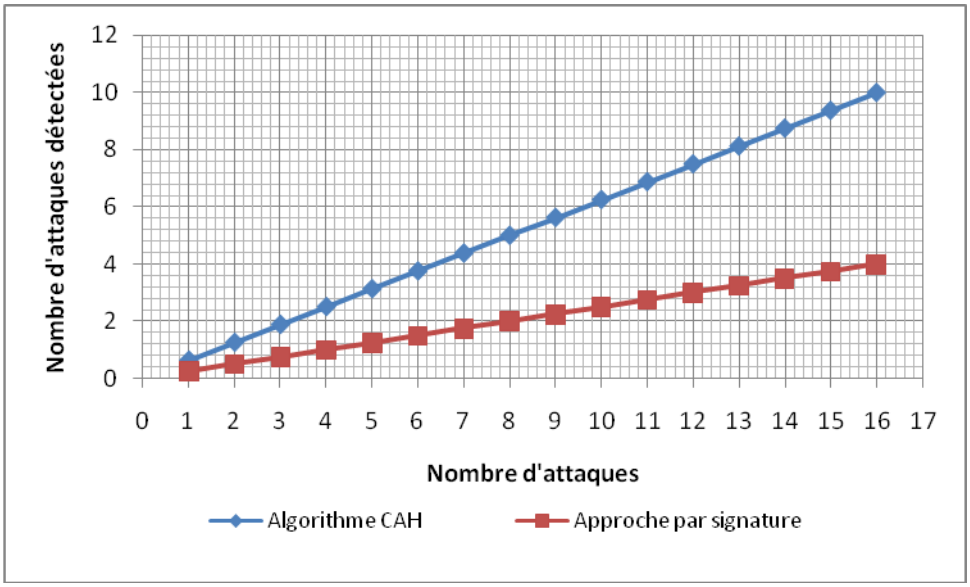


Figure III.23. Nombre d'attaques détectées par rapport au nombre d'attaques total.

La courbe montre que l'approche comportementale optimisée par le CAH permet de détecter plus d'attaques par rapport à l'approche par signatures, tous simplement car cette dernière est basée uniquement sur quatre attaques connues, toute nouvelle attaque passera inaperçus (problème de faux négatifs).

IV.3. L'approche hybride :

On a vu dans la 1ère partie, l'approche comportementale simple qui permet de détecter plus d'attaques mais avec un nombre important de faux positifs, on a amélioré ce travail en utilisant l'algorithme de classification CAH qui minimise le nombre de faux positifs, mais avec une complexité de $O(n^2)$. Dans la 2ème partie, on a vu l'approche par signature qui est basée sur un principe très simple où l'administrateur a un rôle indispensable (l'expérience et l'intelligence humaine). L'inconvénient major de cette approche c'est le problème de faux négatifs.

Dans ce qui suit, on se basera sur les points forts de ces deux approches afin de créer un IDS hybride.

Le principe est simple, on a créé une table nommée « AttaqueHyb » qui contient au départ la base de connaissance (vue dans la figure III.21) fixée par l'administrateur de sécurité avec la possibilité de faire les mises à jour.

Le système analyse chaque comportement de client connecté et le compare avec les signatures initiales. S'il détecte une attaque alors il va bloquer le client (le principe

de l'approche par signature), sinon il va mesurer la similarité entre son nouveau comportement et son comportement normal (fixé dans la phase d'analyse), si l'IDS détecte un changement de clusters, une alerte sera déclenchée et le client sera bloqué (le principe de l'approche comportementale optimisée par le CAH) en ajoutant le comportement actuel de ce client (son nouveau cluster) à la table « AttaqueHyb » (les signatures initiales).

Remarque :

L'approche hybride elle-même comporte deux phases : phase d'analyse et la phase de détection.

Dans la première phase on se base sur l'approche comportementale optimisée par le CAH pour fixer les comportements des clients, et dans la deuxième phase on se base sur les deux approches pour la détection.

IV.3.1. Diagramme récapitulatif :

L'organigramme suivant résume les étapes de détection d'une attaque avec l'approche hybride.

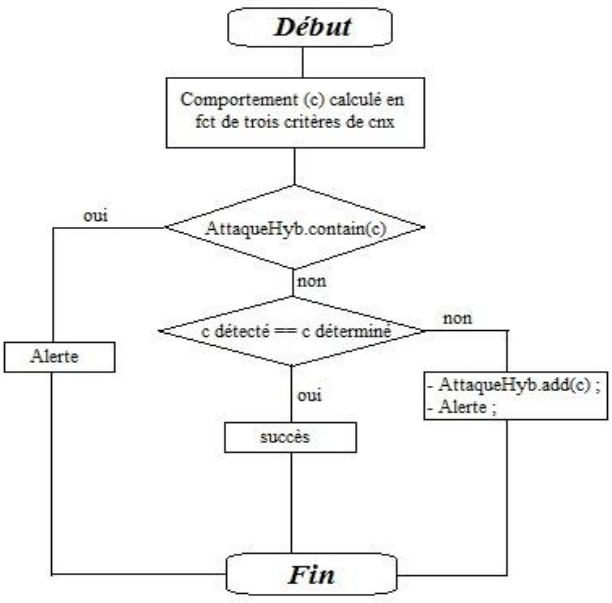


Figure III.24. Organigramme de détection d'une attaque.

La figure III.25 illustre le contenu de la table AttaqueHyb.

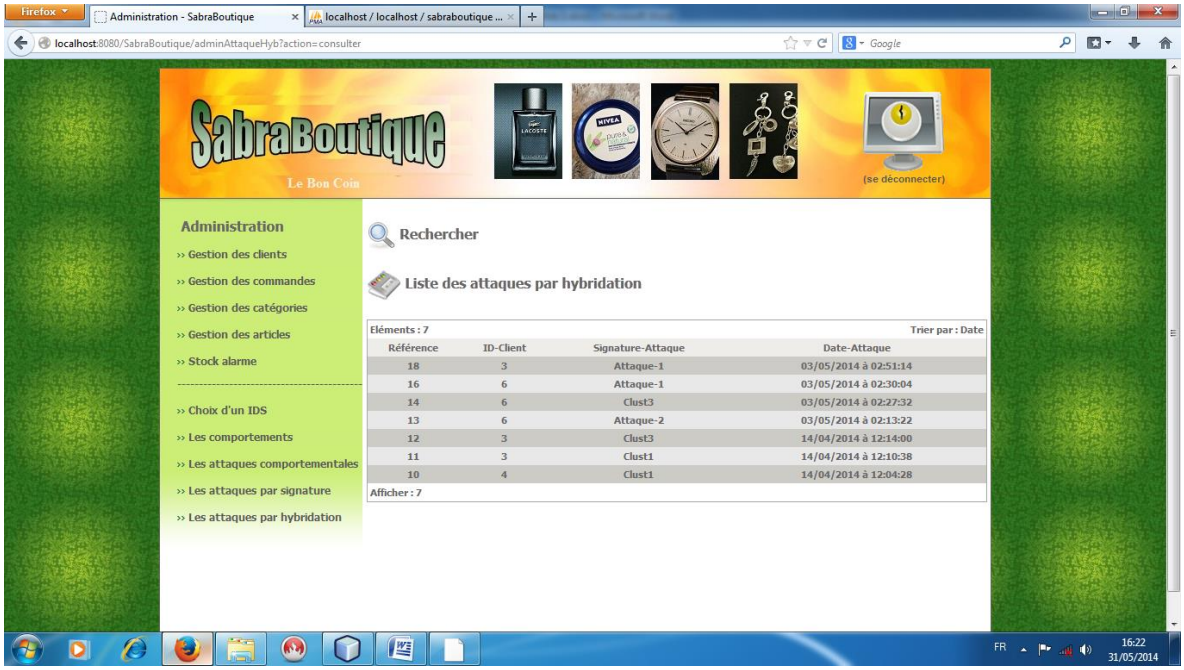


Figure III.25. Le contenu de la table AttaqueHyb.

IV.3.2. Comparaison entre les trois approches :

La courbe suivante montre le nombre d'attaques détectées (pour les trois approches) par rapport au nombre d'attaques total (même liste d'attaques proposée dans la section IV.2.1).

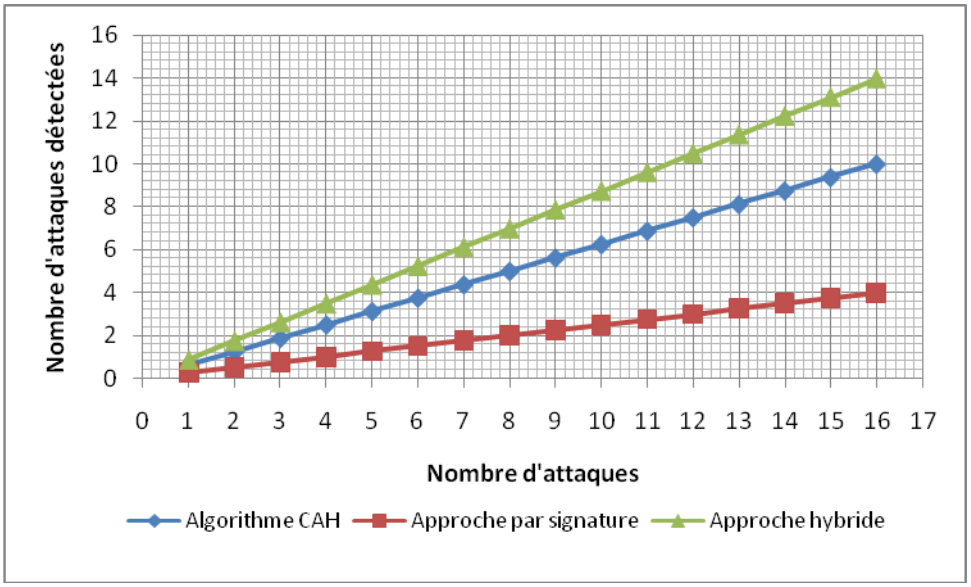


Figure III.26. Comparaison entre les trois approches.

La courbe montre que l'approche hybride est plus performante et capable de détecter plus d'attaques par rapport aux deux autres approches.

V. Conclusion :

Au terme de ce dernier chapitre, nous avons implémenté un système de détection d'intrusion IDS dont le but est de sécuriser notre application Web. Cette implémentation offre, via un ensemble de programme et de bases de données la possibilité de personnaliser un client de l'application et de détecter les attaques.

L'approche comportementale permet de détecter plus d'attaques inconnues, mais elle génère un nombre important de faux positifs, c'est pour ça qu'il est indispensable d'utiliser un algorithme de classification pour minimiser ces fausses alertes.

Pour l'approche par signatures, on peut dire qu'elle détecte que des attaques qui sont connues et aussi il faut remettre à jour la base de signatures d'attaque très souvent.

C'est pourquoi une approche hybride semble indispensable, cette approche permet la sérialisation d'un IDS comportementale suivie d'un IDS par signatures tels que la base de signatures est alimentée régulièrement à partir de résultats obtenus par l'approche comportementale.

Conclusion Générale

Conclusion générale :

La sécurité de l'application Web à l'aide d'un IDS hybride était l'objectif du notre PFE dans lequel nous avons détaillé le fonctionnement des systèmes de détection d'intrusions, le principe de *clustering* ainsi que les différentes étapes de l'implémentation. Pour bien sécuriser notre application, nous avons développé des mécanismes au sein de cette application Web pour la sécuriser avec les deux approches d'IDS (comportementale et par signatures).

Nous avons amélioré les performances de la 1ère approche à travers un algorithme de clustering CAH qui permet de détecter des anomalies avec un taux minimum de fausses alertes.

Nous avons par la suite perfectionné notre IDS à travers une approche hybride qui consiste à la sérialisation d'un IDS comportementale suivi d'un IDS par signatures, l'IDS comportementale permet de filtrer les requêtes normales et ainsi seules les requêtes détectées comme anormales sont passées à l'IDS par signatures.

Ce travail nous a beaucoup apporté dans le domaine de la sécurité. Il nous a permis d'avoir une idée plus claire sur les applications de ce domaine. Nous avons également découvert les plus grandes approches des IDS (comportemental, par signatures et hybride) sans oublier l'intérêt des méthodes de classification adaptées à ce domaine. Cette application que nous avons élaborée présente des avantages comme la détection rapide des anomalies ainsi qu'un taux de fausses alertes limité.

En outre, il est important de noter que le risque nul d'être piraté n'existe pas et il faut s'avoir s'appuyer au mieux sur les outils (nouvellement) disponibles afin de tendre vers cet idéal.

Références Bibliographique :

[1] K. TABIA, « Développement de mécanismes de coopération entre algorithmes d'apprentissage automatique/ classification dans un environnement incertain», Mémoire de magistère, Université Mouloud Mammeri de Tizi ouzou , Avril 2005.

[2] Ludovic Mé**, «** Un complément à l'approche formelle : la détection d'intrusions. In Actes de la journée CIDR97 », Rennes, octobre 1997.

[3] J. Anderson, « Computer Security Threat Monitoring and Surveillance », James P. Anderson Co., Fort Washington, PA, 1980.

[4] D E. Denning, « An intrusion-detection model ». IEEE transactions on software engeneering, SE-13 :222–232, 1987.

[5] Hervé Debar, Marc Dacier et Andreas Wespi, « A Revised Taxonomy for Intrusion-Detection Systems – Annales des Télécommunications», 55, n° 7-8, 2000.

[6] http://www.ree.see.asso.fr/IMG/2pdf001/1216b3e12f/pdf08/2006_0008_10.pdf. consulter le 04/04/2014.

[7] Madjid Ouharoun, « Modélisation de détection d'intrusion par des jeux probabilistes », Mémoire de maitrise, Université du Québec Canada, 2010.

[8] Nicolas Baudoin et Marion Karle, « NT Réseaux –IDS et IPS », 2000, support de cours, Enseignant Etienne Duris en 2003-2004.

[9] Jabou Chaouki, Schillings Michaël et Hantach Anis, « TER Détection d'anomalies sur le réseau », Rapport de projet, Université Paris Descartes, 2009.

[10] Jabou Chaouki, Schillings Michaël et Hantach Anis, « TER Détection sur le réseau », Rapport de projet, Université Paris Descartes, 2009.

[11] http://tel.archives-ouvertes.fr/docs/00/35/53/66/PDF/these_frederic_majorczyk.pdf. consulter le 10/04/2014.

[12] Magnus Almgren, Hervé Debar et Marc Dacier, « Lightweight tool for detecting Web server attacks », In Proceedings of the Network and

Distributed System Security Symposium (NDSS'2000), pages 157–170, San Diego, CA, Février 2000

[13] Giovanni Vigna et al, « A stateful intrusion detection system for world-wide Web servers », In Proceedings of the Annual Computer Security Applications Conference (ACSAC 2003), pages 34–43, Las Vegas, Novembre/December 2003.

[14] Magnus Almgren et Ulf Lindqvist, « Application integrated data collection for security monitoring », In Proceedings of the fourth International Symposium on Recent Advances in Intrusion Detection (RAID 2001), pages 22–36, Canada, Octobre 2001.

[15] Ivan Ristic, « ModSecurity 2.5 », http://www.modsecurity.org/. 2008.

[16] Breach Security, « WebDefend », http://www.breach.com/ products/. 2008.

[17] Christopher Kruegel et Giovanni Vigna, « Anomaly detection of Web-based attacks », In Proceedings of the 10th ACM Conference on Computer and Communication Security (CCS'03), pages 251–261, Washington, Octobre 2003.

[18] Christopher Kruegel et al, « A multi-model approach to the detection of Web- based attacks.Computer Networks », pages 717–738, Aoùt 2005.

[19] K. William et al, « Using generalization and characterization techniques in the anomaly-based detection of Web attacks », In Proceedings of the Network and Distributed System Security Symposium (NDSS 2006), San Diego, Février 2006.

[20] Fredrik Valeur et al, « An anomaly-driven reverse proxy for Web applications », In Proceedings of the 2006 ACM symposium on Applied computing (SAC '06), pages 361–368, France, Avril 2006.

[21] Cédric Michel et Ludovic Mé, « An attack description language for knowledge based intrusion detection », In Proceedings of the 16th International Conference on Information Security (IFIP/SEC 2001), pages 353–365, Juin 2001.

[22] L. Kenneth et al, « Elearning DFA representations of HTTP for protecting Web applications », Computer Networks, 51(5): 1239–1255, 2007.

[23] http://tel.archivesouvertes.fr/docs/00/35/53/66/PDF/these_frederic_majorczyk.pdf. consulter le 31/05/2014.

[24] Hervé Debar et Elvis Tombin, « Accurate and fast detection of http attack traces in Web server logs », In Proceedings of EICAR, Malta, 2005.

[25] Elvis Tombini et al, « A serial combination of anomaly and misuse IDS applied to HTTP traffic », In Proceedings of ACSAC'2004, pages 428–437, Tucson, AZ, Decembre 2004.

[26] Bertrand Le Saux, «Classification non exclusive et personnalisation par apprentissage». application à la navigation dans les bases d'images. 2003

[27] N. Labroche, « Modélisation du système de reconnaissance chimique des fourmis pour le problème de la classification non_supervisés». application à la mesure d'audience sur Intenet. Décembre 2003

[28] G. Celeux, E. Diday, G. Govaert, Y. Lechevallier et H. Ralambondrainy, « Classification automatique des données », Dunod, 1989.

[29] Mounzer BOUBOU, «Contribution aux méthodes de classification non supervisée», mémoire de fin d'étude Doctorat, Université Claude Bernard, Lyon-France, 2006

[30] Jain, A. and Dubes, R, « Algorithms for Clustering Data ». Prentice Hall Advanced Reference Series.1988

[31] N. MONMARCHE, « Algorithmes de fourmis artificielles applications à la classification et à l'optimisation», thèse de doctorat , Université de Tours, décembre 2000

[32] Z.Guellil et L.Zaoui, « Proposition d'une solution au problème d'initialisation cas du K-means », livre : CIIA, volume 547 of CEUR Workshop Proceedings, CEUR-WS.org, Université des Sciences et de la Technologie, Oran– Algérie, 2009

[33] S. C. Johnson, « Hierarchical Clustering Schemes » Psychometrika, no 2, pp 241-254, 1967.

[34] Lance, G.N., & Williams, W.T, « A general theory of classificatory sorting strategies ». I. Hierarchical systems. Computer Journal, no 9,pp 373-380, 1967.

[35] Celeux, G., Diday, E., Govaert, G., Lechevallier, Y., and Ralambondrainy, H, «Classification automatique des données, environnement statistique et informatique ».DUNOD informatique. 1989.

[36] Kamvar, S. D., Klein, D., & Manning, C. D., « Interpreting and Extending Classical Agglomerative Clustering Algorithms using a Model-Based approach » .Pp 283-290 of : International Conference on Machine Learning (ICML).2002.

[37] Guha, S., Rastogi, R., ET Shim, K. CURE, « an e_cient clustering algorithm for large databases ». Dans Proceedings of ACM SIGMOD International Conference on Management of Data, pp 73-84, 1998.

[38] Karypis, G., Eui-Hong, H., ET Kumar, V. Chameleon, «Hierarchical Clustering Using Dynamic Modeling». Computer, no 32(8) :68-75, 1999.

[39] Zhang, T., Ramakrishnan, R., et Livny, M. BIRCH, « an efficient data clustering method for very large databases ». Dans Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, pp 103-114, 1996.

[40] Introduction au langage java disponible sur le site : www.univ-tlemcen.dz/~benmammar, consulté le : 20/05/2014.

[41] Applications web avec Struts2 disponible sur le site : www.univ-tlemcen.dz/~benmammar, consulté le : 24/05/2014.

[42] MySQL un serveur de bases de données relationnelles disponible sur le site : http://www.futura-sciences.com/, Consulté le : 24/05/2014.

Résumé :

Un système de détection d'intrusions (IDS) est un système capable de scruter l'accès et le flux d'information, collecter tous les événements, les analyser et générer des alarmes en cas d'identification de tentatives malveillantes. Deux approches coexistent dans la détection d'intrusions : l'approche par signatures et l'approche comportementale. Chacune des deux présentent des points forts, mais aussi des faiblesses qui sont les faux positifs et les faux négatifs. Notre objectif est de sécuriser une application Web (boutique en ligne) en se basant sur les deux approches et en utilisant une méthode de classification afin de cumuler les forces et d'éliminer les faiblesses.

Mots-clefs : IDS, approche comportementale, approche par signatures, approche hybride, faux positifs, faux négatifs, clustering, CAH.

Abstract :

An intrusion detection system (IDS) is a system able to scan the access and the flow of information, to collect all the events, to analyse them and raise the alarm when there is identification of malicious attempts. Two approaches coexist in the intrusion detection: signature-based approach and behavior-based approach. Each of the two has strengths, but also weaknesses that are false positives and false negatives. Our goal is to secure a web application (online shop) based on the two approaches and using a classification method in order to increase the strengths and to eliminate the weaknesses.

Keywords: IDS, Behavior-based IDS, signature-based IDS, hybrid approach, false positives, false negatives, clustering, CAH.

ملخص :

نظام كشف التسلل (الهوية) هو نظام قادر على فحص الوصول الى تدفق المعلومات، جمع جميع الأحداث مع تحليلها و توليد الإنذارات في حالة تحديد محاولات معادية. تعتمد أساليب كشف التسلل على نهجين رئيسيين : نهج السلوكية و نهج السيناريو. كل منهما لديه نقاط قوة و لكن أيضا نقاط ضعف و التي تتمثل في الأخطاء الإيجابية و الأخطاء السلبية. هدفنا هو إدارة تطبيق ويب (متجر على شبكة الأنترنيت) مستندين على كلا النهجين، و باستخدام طريقة تصنيف للجمع بين نقاط القوة و القضاء على نقاط الضعف.

الكلمات المفتاحية: نظام كشف التسلل، نهج السلوكية، نهج السيناريو، نهج هجين، الأخطاء الإيجابية، الأخطاء السلبية، تجميع، التصنيف الهرمي.