REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU

FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE

DEPARTEMENT D'INFORMATIQUE

Mémoire de fin d'études

En vue de l'Obtention du Diplôme de Master en informatique

Option : systèmes informatique

Thème

CONCEPTION D'UN SYSTEME DE DETECTION D'INTRUSION BASE SUR UN ARBRE DE DECISION

Proposé et dirigé par :

Mme HADDAOUI

Réalisé par :

Mr. Ahmed ZEBOUDJ

Mr. nassim KHOBZI

2014/2015

SOMMAIRE

Partie II

Introduction générale

Introduction :

L'informatisation s'impose aujourd'hui dans des domaines coopératifs et largement distribués comme la télémédecine, le commerce électronique et l'E-gouvernement. Cette informatisation impose d'avoir confiance dans les traitements et la distribution des données et des services informatique plus vulnérable aux attaques. La sécurisation des systèmes informatiques est devenue alors un enjeu majeur. La sécurité a pour objectif d'assurer la disponibilité des services, l'authentification des utilisateurs, la confidentialité et l'intégrité des données. De nombreuses solutions ont été développées, citons par exemple le contrôle d'accès qui consiste à prouver l'identité des utilisateurs et définir les droits accordés sur les données. Les pare-feux qui filtrent l'accès aux services du système informatique. Les scanners de vulnérabilités qui cherchent les failles au niveau des entités du système informatique. Les systèmes de détection d'intrusion qui détectent les attaques et informant l'opérateur de sécurité.

Malheureusement tous ces systèmes de sécurité ont des limitations. Ils présentent des failles de conception, d'implémentation ou de configuration qui permettent à des attaquants de les contourner. La plus part des entreprise utilisent différents systèmes de protection comme : des anti-virus, pare-feux, contrôles l'accès a leur réseau, des systèmes de détection d'intrusion, et malgré ça ces entreprises ont été attaquées et ils ont enregistré des grandes pertes. En fait, les attaquants compliquent de plus en plus leurs techniques. D'après une étude faite à l'université de Carnegie Mellon, les programmes d'attaques devient de plus en plus dangereux et nécessitent moins d'expertise ce qui expose les entreprises à des menaces d'intrusions supplémentaires. Cette évolution des techniques force les dirigeants à réviser leurs méthodes de sécurité afin de protéger efficacement leurs systèmes informatique.

Problématique

La sécurisation des systèmes informatiques commence par la mise en place d'une politique de sécurité. Une politique de sécurité est l'ensemble de règles spécifiant comment les ressources sont gérées afin de satisfaire les exigences de sécurité et quels sont les actions permises et les actions interdites. Une politique de sécurité peut se définir par deux caractéristiques : les niveaux ou elle intervient et les outils utilisés pour l'assurer.

La politique peut intervenir à plusieurs niveaux. La disponibilité qui garantit que les données ne doivent être visibles que pour des personnes autorisée, etc.

Comme nous l'avons mentionné dans l'introduction, différents systèmes sont disponibles pour assurer une politique de sécurité. Dans cette thèse nous nous intéressons aux **systèmes de détection d'intrusion**.

Introduction générale

Notre contribution :

Dans ce mémoire nous proposons un modèle pour la conception d'un IDS comportemental basé sur un arbre de décision utilisant l'algorithme de classification CART (Classification And Regression Trees).

Organisation :

Ce mémoire est organisé comme suit :

Nous avons débuté par une introduction générale dans laquelle nous avons définis la problématique et notre contribution à la solution.

Chapitre I : ce chapitre propose un état de l'art sur les réseaux informatiques.

Chapitre II : nous avons expliqué la de sécurité informatique.

Chapitre III : il traite les différents systèmes de détections d'intrusions

Chapitre IV : il se compose de deux parties **:**

- **Partie I :** elle représente les différents classifications et arbres de décision.
- **Partie II :** cette partie est consacrée à la présentation de la base KDD qu'on se propose d'utiliser dans notre études expérimentale

Chapitre V :

Dans ce chapitre, nous avons procéder à la présentation de notre IDS nommé **IDSCART.** Et en expliquant notamment la phase d'apprentissage ainsi que la phase de test qui est primordiales pour la conception et la validation de notre IDS.

Et enfin, nous terminons notre thèse par une conclusion générale et des perspectives futures pour continuer et améliorer le travail que nous avons entamé.

I .1 introduction

Avec le développement de l'informatique et le besoin d'échanges de données entres des machines (ordinateurs, machines, imprimantes etc..) distantes les informaticiens ont eut les besoins de créer un moyen qui permet de partager des ressources entres ces machines qui est bien évidement les réseaux informatique.

I .2 définition des réseaux informatique : [1]

Un réseau informatique est un ensemble d'hôtes interconnectés entre eux à fin d'échanger des informations (données informatiques, du texte, image, de la vidéo ou de son), grâces à des techniques et des outils informatiques (internet, serveurs, switchers, routeurs …) selon des règles et des protocoles bien définis

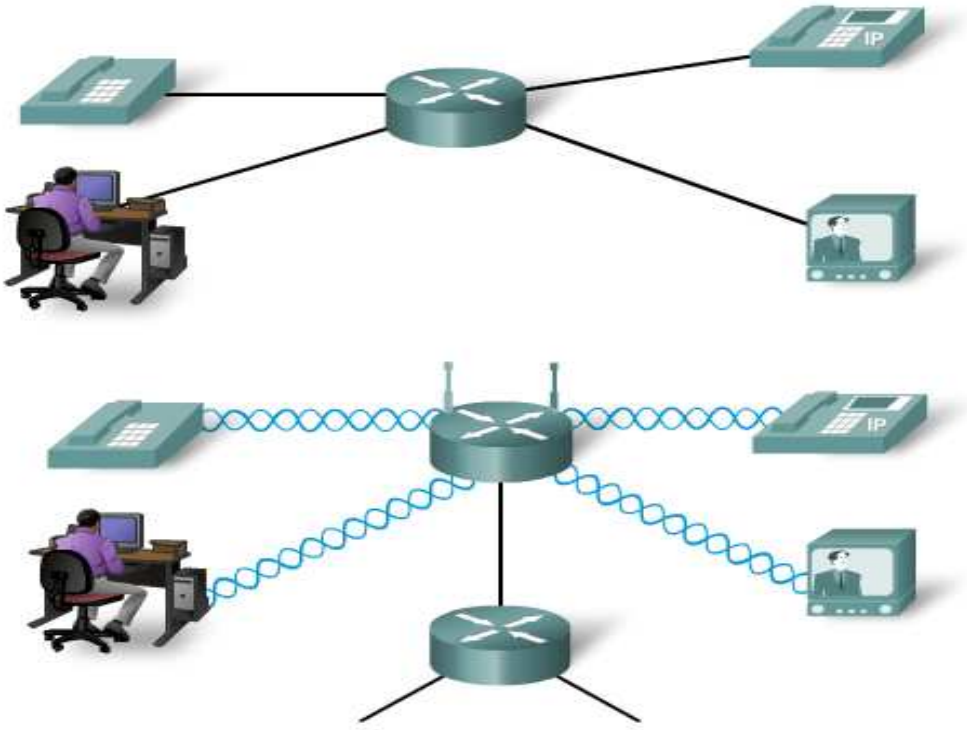


Figure I.1 : réseaux informatique [49]

I.3 objectifs des réseaux : [1]

Nous appellerons un réseau un ensemble d'ordinateurs interconnectés entre eux et réalisant des taches différentes. Ceci étant posé, les objectifs d'un réseau sont classiquement les suivants.

- **Le partage de ressources**

Rendre accessible à une communauté d'utilisateurs des programmes, des données et des équipements informatique (i.e. un ensemble de ressources)

- **La fiabilité**

Permettre le fonctionnement même en cas de problèmes matériels (sauvegardes, duplication …). Penser aux applications militaires, bancaires, au contrôle des centrales nucléaires ou aérien…

- **La réduction des couts**

Les petits ordinateurs (PC par ex.) ont un meilleur rapport prix/performances que les gros. Aujourd'hui, nous trouvons surtout des architectures client/serveur plus économique, plus souple et permettant un dépoilement incrémental aisé (contrairement aux architectures à base de mainframe).
Un réseau est aussi une infrastructure de communication permettant le travail collaboratif et/ou les échanges entre des personnes géographiquement séparées.

I.4 : classifications des réseaux informatiques [1] [2]

La classification se fait sur un critère donné, ainsi nous pouvons classer les réseaux informatiques :
a) selon l'étendue géographique.
b) selon les fonctions assurées par les ordinateurs.
c) selon la topologie.

I.4.1 : selon l'étendue géographique :

Selon la taille géographique qu'occupe un réseau, on peut les classer en quatre grandes catégories PAN, MAN, LAN et WAN

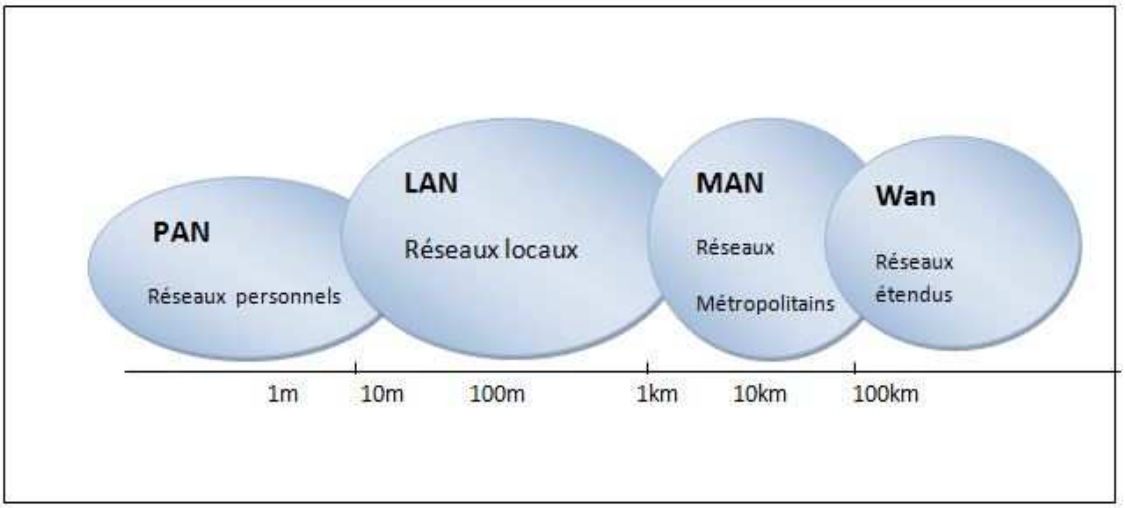


Figure I.2 : classification des réseaux selon leur étendue

a) le réseau LAN (local area network) :

Les réseaux locaux connectent plusieurs ordinateurs situés sur une zone géographique relativement restreinte, tel qu'un domicile, un bureau, un bâtiment, un campus universitaire.

Ils permettent aussi aux entreprises de partager localement des fichiers et des imprimantes d'une manière efficaces et rendent possible les communications internes.

b) le réseau MAN (metropolitain area network)

Tous réseau métropolitain est essentiellement un LAN, de point de vu de la technologie utilisé .il peut couvrir un grand campus ou une ville.

c) le réseau WAN (wide aera network)

Pour des raisons économiques et techniques, les réseaux locaux (LAN) ne sont pas adaptés aux communications couvrant de longues distances.
C'est pour toutes ces raisons que les technologies des réseaux étendus (WAN) diffèrent de celles des réseaux locaux .un WAN est un réseau a longue distance qui couvre une zone géographique importante.

I.4.2 selon les fonctions assurées par les ordinateurs

De point de vu architecture réseaux nous avons :

- ✓ Réseau poste à poste
- ✓ Réseau client / serveur

a) réseau poste à poste

C'est un réseau sans serveur, ou chaque ordinateur connecté peut faire office de client ou de serveur. En général c'est un petit réseau de plus au moins de 10 postes sans administrateur de réseau, ce réseau est illustré par la figure I.3

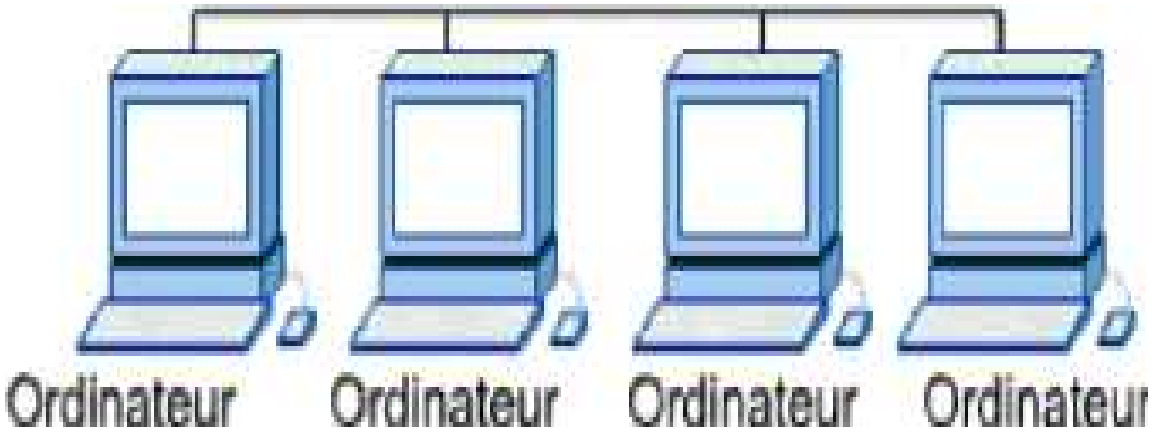


Figure I.3 : schémas d'un réseau poste à poste

b) réseau à serveur dédicacé ou client serveur

Dans une configuration client-serveur les services d'un réseau sont placés sur un ordinateur dédiés, appelé serveur, qui répond aux requêtes des clients, un serveur est un ordinateur central, disponible en permanence pour répondre aux requêtes émises par des clients et relatives à des services des fichiers, d'impression, d'applications et autres la figure I.4 illustre se réseau.

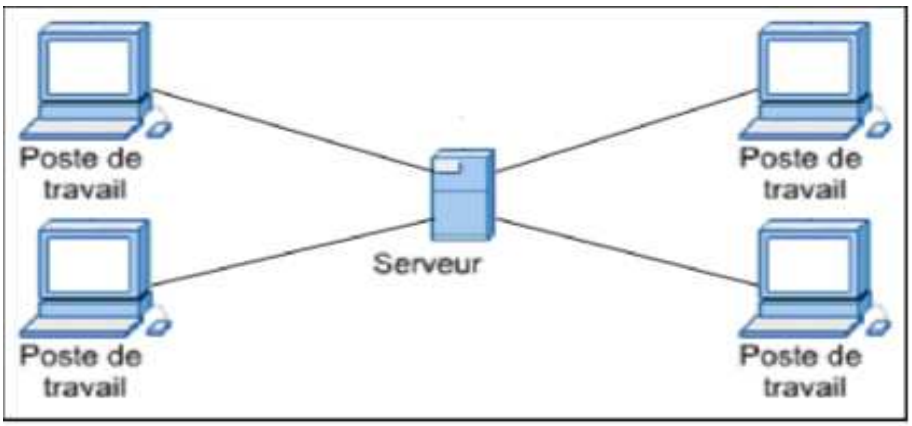


Figure I.4 schémas d'un réseau client/serveur

I.4.3 selon la topologie réseau :

La topologie du réseau définit la structure de réseau .elle représente l'interconnexion des équipements sur le réseau .ces équipements sont appelés des nœuds .les nœuds peuvent être des ordinateurs, imprimantes, des routeurs, des ponts ou tous autres composants connectés au réseau. Un réseau est composé de deux topologies : physiques et logiques

I.4.3.1 topologie physique :

La topologie physique se rapporte à la disposition des équipements et des supports .ainsi nous pouvons les classer en deux modes.

a) **Réseau en mode de diffusion :** dans ce mode touts les nœuds communiquent via un seul canal de transmission .on trouve

- **Topologie en bus :**

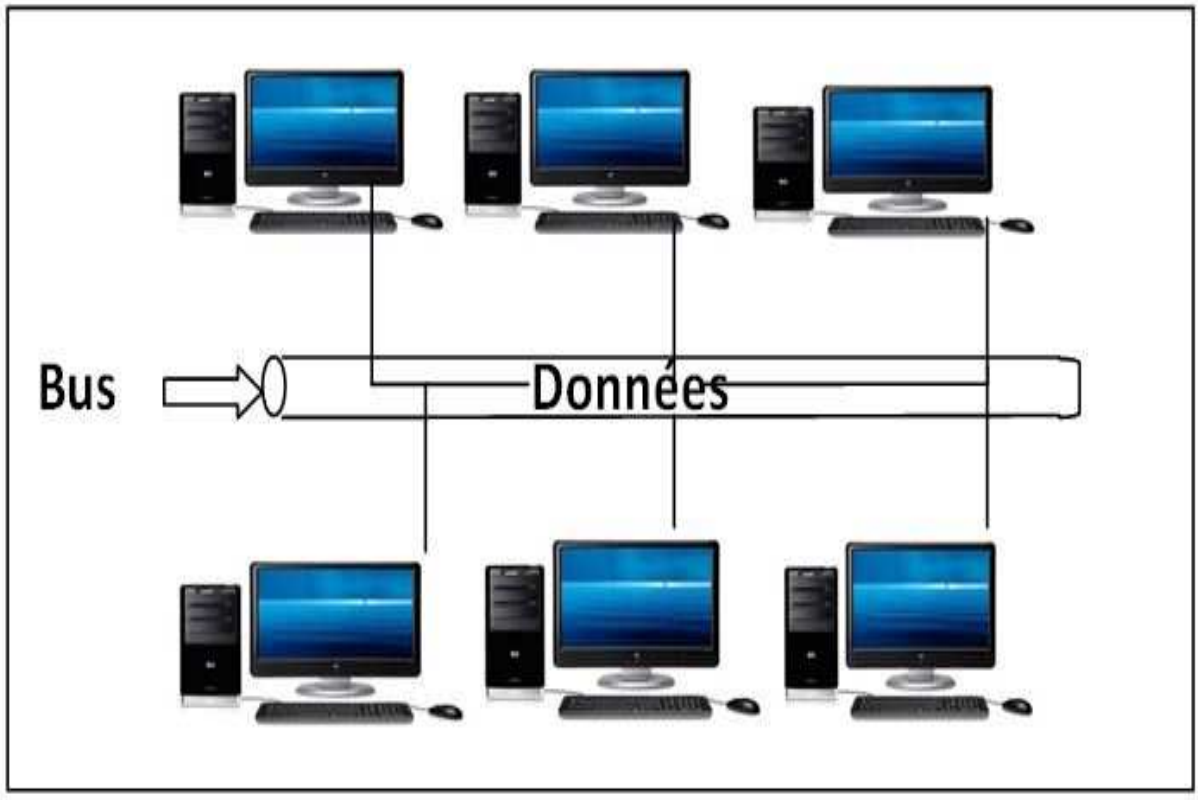


Figure I.5 : topologie en bus

- **Topologie en anneau :**

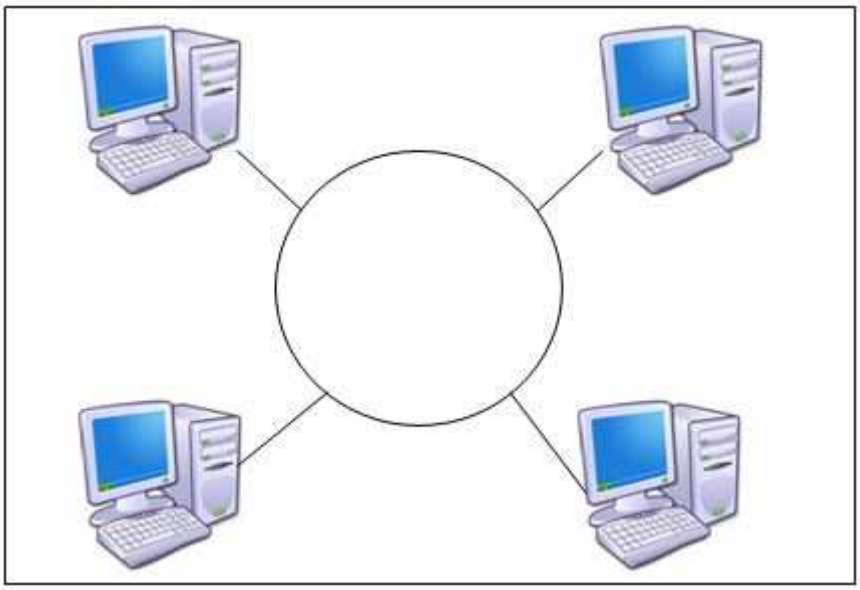


Figure I.6 : topologie en anneau

a) Réseau en mode point à point : par contre dans celui-ci chaque support physique relie une paire d'unité .on trouve.

- **Topologie en étoile :**

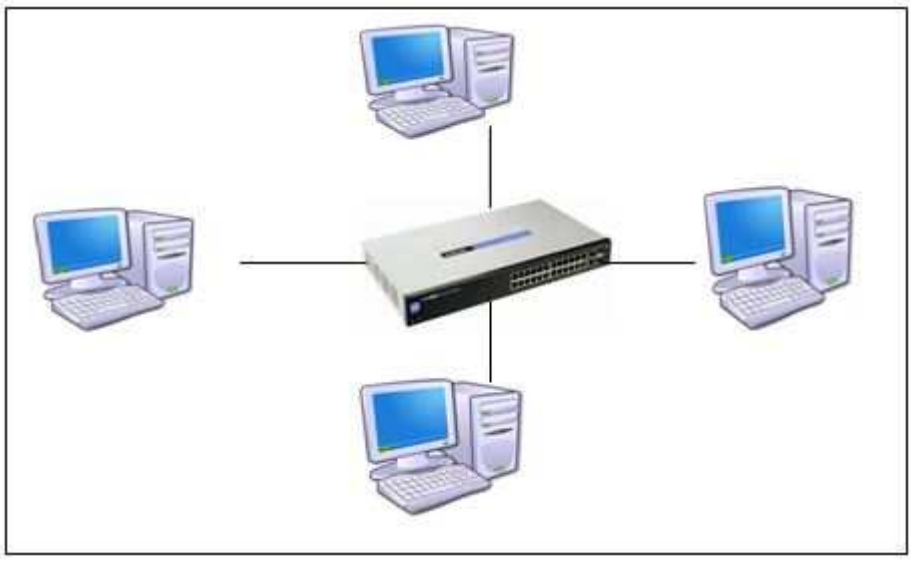


Figure I.7 : topologie en étoile

- **Topologie maillé :**

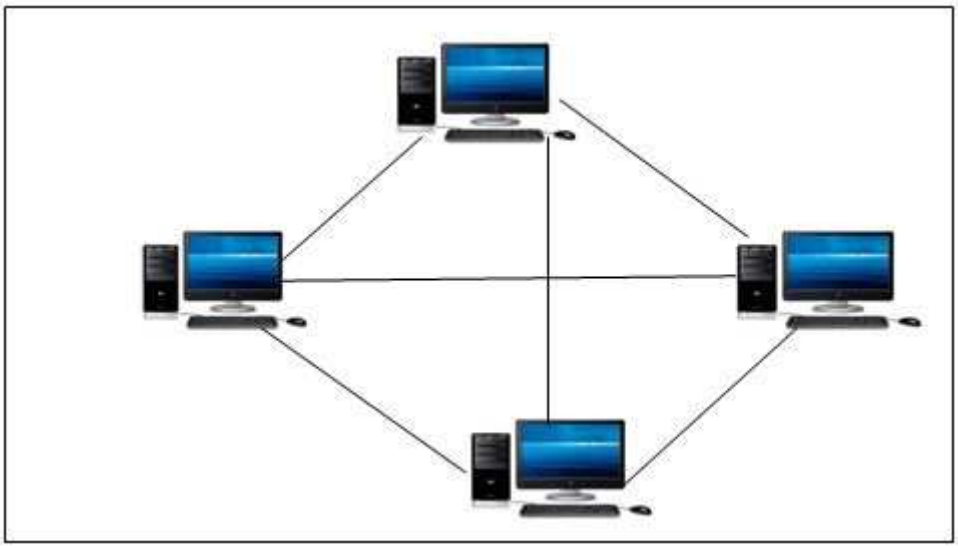


Figure I.8 topologie maillé

I.4.3.2 topologie logique

La topologie logique représente des voies par lesquelles sont transmis les signaux sur le réseau (mode d'accès des données au support et de transmission des paquets de données)**.**

I.5 fonctionnement d'un réseau : [3]

En informatique il existe une multitude de méthodes et de langages pour communiquer. C'est pourquoi, ainsi des organismes internationaux se sont attelés à un travail de standardisation, de normalisation

Quatre principaux organismes internationaux travaillent de concert sont apparu : [4]

- ISO : international organizations for standardization.
- CEI : commission électrotechnique internationale.
- ITU : international télécommunication union.
- IEEE : Institute Of Electric and Electricity Engineers.

Pour mieux décrire la complexité des communications réseau, deux représentations des systèmes informatiques ont vu le jour : [3] [4]

I.5.1. le modèle OSI (Open System Interconnexion) :

Au début des années 70, chaque constructeur a développé sa propre solution autour d'architecture et de protocole privés, et il s'est vite avéré qu'il serait impossible

d'interconnecter ces différents réseaux si une norme internationale n'était pas établie. Cette norme établis par l'International Standard Organization (ISO) et la norme Open System Interconnections, elle est constitué de sept couches suivantes :

- ✓ **La couche physique :**

  - Assurer la transmission de bits entre les entités physiques.
  - Spécifie la nature du support de communication.
  - Le mode de connexion et le brochage le cas échéant.
  - La technique de codage des bits en signaux électriques.
  - Les tensions et les fréquences utilisées.

- ✓ **la couche liaison de données :**

  - structuration des données en trames
  - masquer les caractéristiques physiques
  - contrôle d'erreur à l'émission et à la réception

- ✓ **La couche réseau :**

  - Permet l'acheminement de bout en bout en tenant compte des nœuds intermédiaires
  - Routage et ordonnancement des paquets

- ✓ **La couche transport :**

  - Permet l'acheminement de bout en bout sans se soucier des relais intermédiaires
  - Fragmentation du message en unités plus petites dites paquets
  - Multiplexage

- ✓ **La couche session :**

  - Permet d'initier un dialogue entre les applications source et de destination
  - Initier et maintenir un dialogue
  - Redémarrer les sessions interrompues ou inactive pendant une longue période

✓ **La couche présentation :**

- Codage et conversion des données de la couche application afin que les données issues du périphérique source puissent être bien interprétées sur le périphérique de destination
- Compression des données de sorte que celles-ci puissent être décompressées par le périphérique de destination
- Chiffrement des données en vue de leur transmission et déchiffrement des données reçues par le périphérique de destination

✓ **La couche application :**

- Sert d'interface entre les applications à chaque extrémité du réseau
- Permet d'échanger des données entre les programmes s'exécutant sur les hôtes source et de destination

Il existe de nombreux protocoles de couche application et de nouveaux protocoles sont constamment en cours de développement

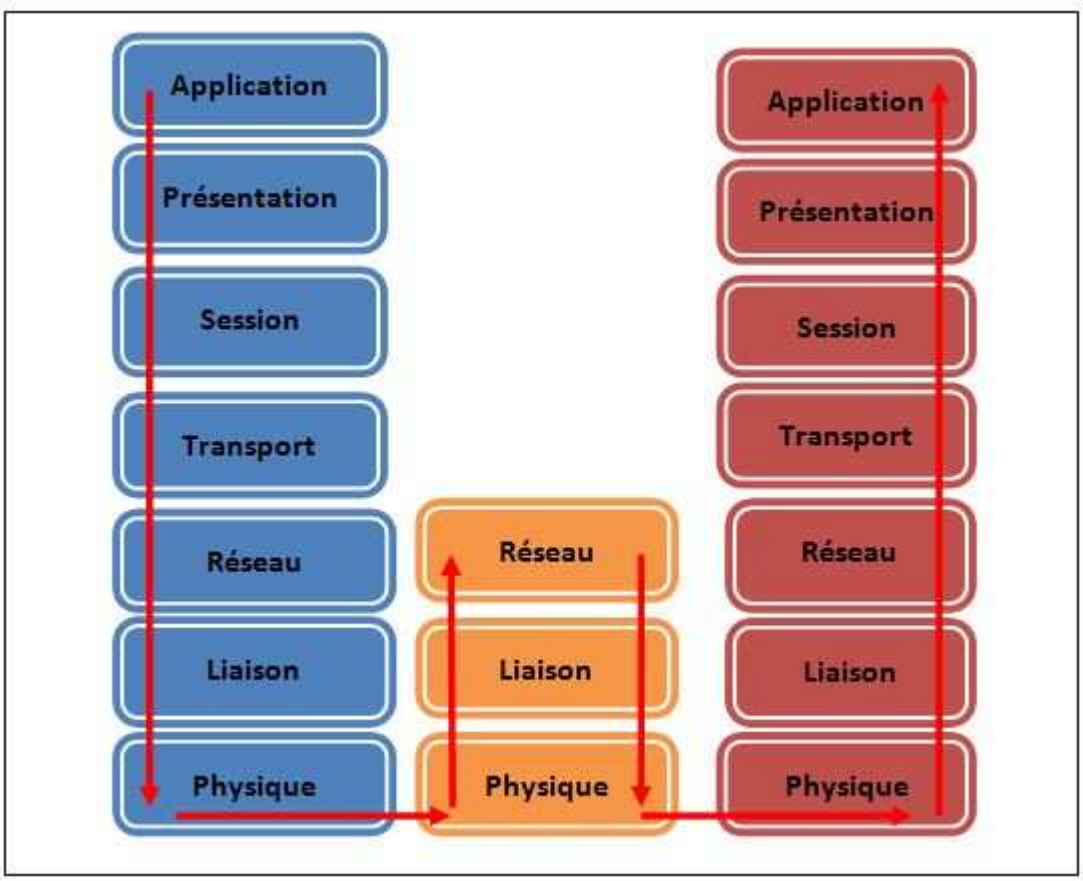


Figure I.9 : les couches de modèle OSI

I.5.2.l'architecture TCP/IP [5]

Le modèle IP (pour Internet Protocol) est un découpage en couches plus réaliste que le modèle OSI en ce qui concerne la pile de protocole IP. On le nomme aussi le modèle TCP/IP par abuse de langage. Il définit quatre couches :

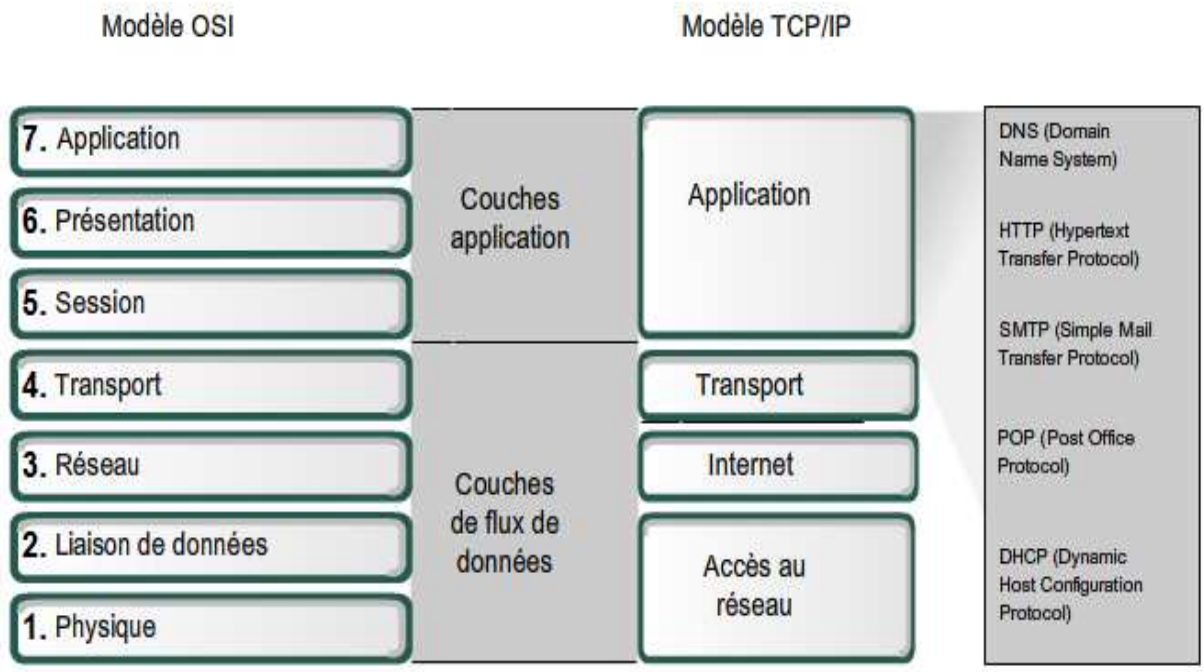


Figure I.10 : la pile de protocole TCP/IP avec sa correspondance OSI [50]

a) la couche d'accès au réseau :

La couche d'accès au réseau est la première couche de la pile TCP/IP, elle offre les capacités à accéder à un réseau physique quel qu'il soit, c'est-à-dire les moyens à être en œuvre afin de transmettre des données via un réseau. Ainsi contient toute les spécifications concernant la transmission de données sur un réseau physique, qu'il s'agisse de réseau local (anneau à jeton – Token ring, Ethernet, FDDI), de connexion à une ligne téléphonique ou n'importe quel type de liaison à un réseau. Elle prend en charge les notions suivantes :

- ✓ acheminement des données sur la liaison.
- ✓ Coordination de la transmission.
- ✓ Format des données.
- ✓ Conversion des signaux (analogique/numérique)
- ✓ Contrôle des erreurs à l'arrivé.

b) la couche internet :

Fournit les mécanismes nécessaires à la gestion et à l'acheminement des paquets de données (protocole IP). Elle contient cinq protocoles :

- **IP :** il gère les destinations des messages.
- **ARP (Adresse Résolution Protocol) :** protocole de résolution d'adresse, il permet de transformer une adresse logique (IP) en une adresse physique (MAC).
- **RARP (Reverse Adresse Resolution protocol)** protocole de résolution d'adresse reverse, qui assure le travail inverse de ARP, il convertit une adresse physique en une adresse logique.
- **ICMP (Internet Control Message Protocol) :** il permet de signaler au couche supérieure que des messages contiennent des erreurs sans les corrigées.

c) La couche transport : assure la communication de données entre deux machines distantes en utilisant des protocoles, comme TCP et UDP.

- **TCP (Transmission Control Protocol) :** TCP est protocole de la couche transport, au-dessus de protocole IP, il offre des nombreux service : l'ordonnancement des données, orientation de la connexion, ainsi que le contrôle de données.
- **UDP (User Datagram Protocol) :**offre aux applications différents point d'accès, deux applications différentes ne peuvent pas partager les mêmes points d'accès.

d) La couche application : la couche application dans ce modèle est directement supérieure à la couche transport, elle est responsable de l'interaction directe avec les utilisateurs et elle englobe toutes les applications.

I.6 conclusion :

Les réseaux informatiques sont entrain de s'imposer comme la solution du partage d'information, grâce notamment à la minimisation des couts et à l'augmentation des performances des systèmes.

Nous avons consacré ce chapitre à la présentation de quelques notions générale sur les réseaux informatiques, ainsi que les deux architectures les plus utilisé OSI et TCP/IP.

Le chapitre suivant sera consacré à la sécurité informatiques ainsi les différentes techniques de protection contre les menaces informatiques.

II.1. Introduction

Avec le développement de l'utilisation d'internet, de plus en plus d'entreprise ouvrent leur système d'information à leurs partenaires ou leurs fournisseurs, il est donc essentiel de connaitre les ressources de l'entreprise à protéger et de maitriser le contrôle d'accès et les droits des utilisateurs du système d'information. Il en va de même de l'ouverture de l'accès de l'entreprise sur internet.

Par ailleurs, avec le nomadisme, consistant à permettre aux personnels de se connecter au système d'information à partir de n'importe quel endroit, les personnels sont amenés à « transporter » une partie du système d'information hors de l'infrastructure sécurisé de l'entreprise.

II.2 sécurité informatique : [6]

Un système d'information est une organisation d'activités consistant à acquérir, stocker, transformer, diffuser, exploiter et gérer des informations. Un des moyens technique pour faire fonctionner un système d'informatique. Assurer donc la sécurité de l'information est d'utiliser implique d'assurer la sécurité des systèmes informatiques. En fait, l'information est une ressource stratégique ; elle représente un patrimoine essentiel de l'entreprise. Une grande partie du budget d'une organisation est dépensée dans la gestion de l'information. Une menace donc au système d'information d'un organisme peut provoquer une perte de confiance des clients, un vol des données confidentielles, des pertes financières et peut même menacer l'existence de cet organisme. D'où la nécessité de la sécurité pour protéger le système informatique.

La sécurité informatique est l'ensemble des moyens mis en œuvre pour réduire les vulnérabilités d'un système contre les menaces accidentelles ou intentionnelles. Les menaces intentionnelles est l'ensemble des actions malveillantes qui constituent la plus grosse partie du risque et qui devraient être l'objet principal des mesures de sécurité.

Ces mesures de sécurité sont en général utilisées, de façon à sécuriser les différentes failles existantes dans un système informatique. La cryptographie est utilisée afin de transmettre un ensemble de règles de filtrage laissant passer les paquets s'ils sont autorisés et bloquant les échanges qui sont interdits. Les pots de miel (honeypot) servent d'appât pour apprendre la stratégie des attaquants et construire des signatures exactes d'attaques. Les systèmes de détection d'intrusion (IDS) servent à analyser les données et détecter les intrusions et les systèmes de prévention d'intrusion (IPS) servent à prévenir les intrusions.

II.3. critères fondamentaux : [7]

- ✓ **Intégrité des données :** le contrôle d'intégrité d'une donnée consiste à vérifier que cette donnée n'a pas été modifie, frauduleusement ou accidentellement.

- ✓ **Confidentialité :** il s'agit de rendre l'information embrouillé à tout l'Oscar, aussi bien de lors de sa conversion qu'au cours de son transfert par un canal de communication. L'information n'est consultable que par son destinataire uniquement.

- ✓ **Contrôle d'accès :** il s'agit d'authentifier les utilisateurs de façon à limiter l'accès aux données, serveurs et ressources par les seules personnes autorisées.

- ✓ **Identification / authentification :** le contrôle d'indentification consiste à s'assurer que chaque tiers est bien lui-même (authentification des partenaires) et d'obtenir une garantie que tel tiers à bien déclenché l'action (authentification de l'origine de l'information). C'est un problème fondamental, qui exige de faire confiance à un tiers dans le cas ou les deux interlocuteurs ne se connaissent pas au préalable.

- ✓ **Non-répudiation :** elle joue le rôle de signature contractuelle, c.-à-d. qu'une personne ne peut pas revenir sur ce quelle à transmet. Il n'y a pas pu y'avoir de transmission de sa part sans son accord. L'émetteur ne peut pas nie l'envoi de l'information, le récepteur ne pas nie la réception de l'information, ni l'un ni l'autre ne peut nie le contenu de cette information (très important lors de passage d'une commande par exemple). Personne ne pourra prendre l'identité d'un autre pour transmettre une information en son nom.

II.4 Les risques et les menaces : [7]

- ➢ **Les menaces :**

Une menace est quelqu'un ou quelque chose qui peut exploiter une vulnérabilité pour obtenir, modifier ou empêcher l'accès à un motif ou encore le compromettre.

Elle existe en corrélation avec des vulnérabilités il peut y avoir aussi plusieurs menaces pour chaque vulnérabilité. La connaissance des différents types de menaces peut aider dans la détermination de leurs dangerosités et des contrôles adaptés permettant de réduire leur impact potentiel.

La menace est une source effective d'incident pouvant entrainer des effets indésirables et graves sur un actif ou un ensemble d'actifs.

Les menaces peuvent être classées par leur origine ou source, type, motivation, action.

Les principales menaces effectives auxquelles un système d'information peut être confronté sont :

- ❖ **Un utilisateur du système** : l'énorme majorité des problèmes liés à la sécurité d'un système d'information est l'utilisateur, généralement insouciant.
- ❖ **Une personne malveillante (hacker et crackers)** : une personne parvient à s'introduire sur le système, légitimement ou non, et à accéder ensuite à des

données ou à des programmes auxquels elle n'est pas censée avoir accès en utilisant par exemple des failles connues et non corrigées dans les logiciels.

- **Un programme malveillant** : un logiciel destiné à nuire ou à abuser des ressources du système est installé par mégarde ou par malveillance sur le système, ouvrant la porte à des intrusions ou modifiant les données. Des données personnelles peuvent être collectées à l'insu de l'utilisateur et être réutilisées à des fins malveillantes ou commerciales.

- **Les attaques** :

Les attaques se divisent, selon leurs types sur quatre catégories et leurs buts sont :

- **Interruption** : vise la disponibilité des informations.
- **Interception** : vise la confidentialité des informations.
- **Modification** : vise l'intégrité des informations.
- **Fabrication** : vise l'authenticité des informations.

II.5 Objectifs de la sécurité : [8]

La notion de sécurité fait référence à la propriété d'un système, d'un service ou d'une entité. Elle s'exprime le plus souvent par les objectifs de sécurité suivants :

- La disponibilité (D).
- L'intégrité (I).
- La confidentialité(C).

Ces objectifs peuvent être compris comme étant des critères de base (dits critères DIC) auxquels s'ajoutent des fonctions de sécurité qui contribuent à confirmer d'une part la véracité, l'authenticité d'une action, entité ou ressource (notion d'authentification) et, d'autre part, l'existence d'une action (notion de non-répudiation d'une transaction, voire d'imputabilité).

La réalisation de fonctions de sécurité, telles que celles de gestion des identités, du contrôle d'accès, de détection d'intrusion par exemple, à satisfaire les exigences de sécurité exprimées en termes de disponibilité, d'intégrité, de confidentialité. Elles concourent à la protection des contenus et des infrastructures numériques et sont supportées par des solutions techniques. Celles-ci sont à intégrer dans le système à sécuriser, en fonction du cycle de vie de ce dernier, par des approches complémentaires d'ingénierie et de gestion de la sécurité informatique.

II.6 privilège de la sécurité : [9]

Aucun système ne peut être parfaitement sur car ses utilisateurs ne le sont pas. Il est nécessaire de protéger chaque utilisateur des autres, ce qui conduit au principe de division des privilèges.

- La division des privilèges nécessite :

- ✓ Une protection logicielle entre les utilisateurs : identification (qui ils sont) et authentification (il faut le prouver).
- ✓ Une protection matérielle au niveau du noyau : l'architecture de l'ordinateur doit rendre possible la division des privilèges en séparent l'espace mémoire du noyau de l'espace mémoire des utilisateurs (appels systèmes nécessaires pour effectuer des actions sensibles).
- De nombreuses applications sont vulnérables à cause d'une mauvaise programmation, souvent involontaire (négligence, manque de temps, la peur de mal l'établir,..) et parfois intentionnelles.
- Les solutions contre ces vulnérabilités sont souvent simples à mettre en œuvre :
  - ❖ Utilisation de fonctions sécurisées.
  - ❖ Activation de certaines protections systèmes

II.7 Famille d'attaque : [10] [11]

II.7.1 les attaques par programmes malveillants

- **Les virus :** un virus est un logiciel capable de s'installer sur un ordinateur à l'insu de son utilisateur légitime. Le terme virus est réservé aux logiciels qui se comportent ainsi avec un but malveillant, parce qu'il existe des usages légitimes de cette technique dite code mobile : les appliquettes java et les procédures java script sont des programmes qui viennent de s'exécuter sur un ordinateur en se chargeant à distance depuis un serveur web, et en principe avec un motif légitime. Les concepteurs de java et de la java script nous assurent qu'ils ont pris toutes les précautions nécessaires pour que ces programmes ne puissent pas avoir d'effet indésirable sur un ordinateur, bien que ces précautions, comme toutes précautions, soient faillibles**.** Les appliquettes java s'exécutent dans un bac à sable (sandbox) qui en principe les isole totalement du système de fichiers qui contient des documents ainsi que du reste de la mémoire de l'ordinateur.
- **Cheval de Troie :** un cheval de Troie (trojan horse) est un logiciel qui se présente sous un jour honnête, utile ou agréable, et qui une fois installé sur un ordinateur y effectue des actions cachées et pernicieuses.
- **Porte dérobée :** une porte dérobée (backdoor) est un logiciel de communication caché, installé par exemple par un virus ou par un cheval de Troie, qui donne à un agresseur extérieur accès à l'ordinateur victime, par le réseau.
- **Bombe logique :** une bombe logique est une fonction, caché dans un programme en apparence honnête, utile ou agréable, qui se déclenchera à retardement, lorsque sera atteinte une certaine date, ou lorsque surviendra un

certain événement. Cette fonction produira alors des actions in désirées, voir nuisible.

- **Logiciel espion :** un logiciels espion, comme son nom l'indique, collecte à l'insu de l'utilisateur légitime des informations au sein du système ou il est installé, et les communique à un agent extérieur, par exemple au moyen d'une porte dérobée. Une variété particulièrement toxique de logiciel espion est le keylogger (espion dactylographique ?), qui enregistre à son honorable correspondant ; il capte ainsi notamment identifiants, mots de passe et codes secrets.

II.7.2. Les attaques par messagerie électronique :

a) **Le pourriel (spam)** : un courrier électronique non sollicité la plupart du temps de la publicité. Ils encombrent le réseau, et font perdre de temps à leurs destinataires.

b) **L'hamençonnage (phishing) :** un courrier électronique dans l'expéditeur se fait généralement passer pour un organisme financier et demandant au destinataire de fournir des informations confidentielles.

c) Le canular informatique (hoax) : un courrier électronique incitant généralement le dentinaire à retransmettre le message à ces contacts sous divers prétextes. Ils encombrent le réseau, et font perdre le temps à leurs destinataires. Dans certains, ils incitent l'utilisateur à effectuer des manipulations dangereuses sur son poste (suppression d'un fichier prétendument lié à un virus par exemple).

II.7.3 Les attaque sur les réseaux :

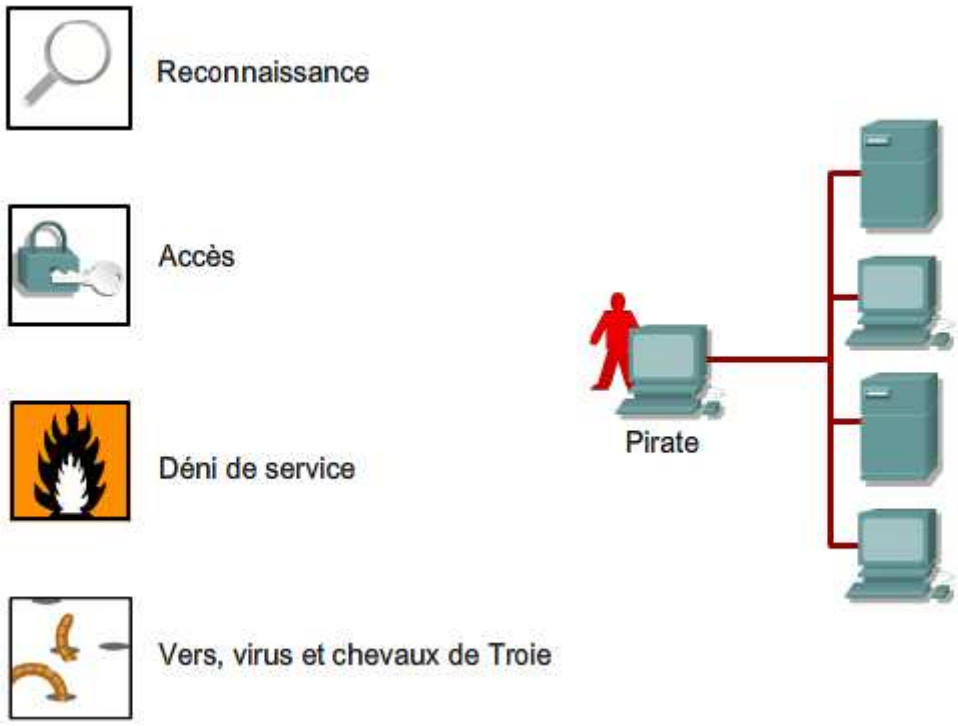
a) **Reconnaissance :** La reconnaissance est la découverte non autorisée des systèmes, de leurs adresses et de leurs services, ou encore la découverte de leurs vulnérabilités. Il s'agit d'une collecte d'informations qui, dans la plupart des cas, précède un autre type d'attaque. La reconnaissance est similaire au repérage effectué par un cambrioleur à la recherche d'habitations vulnérables, comme des maisons inoccupées, des portes faciles à ouvrir ou des fenêtres ouvertes.

b) **Accès :** L'accès au système est la possibilité pour un intrus d'accéder à un périphérique pour lequel il ne dispose pas d'un compte ou d'un mot de passe. La pénétration dans un système implique généralement l'utilisation d'un moyen de piratage, d'un script ou d'un outil exploitant une vulnérabilité connue de ce système ou de l'application attaquée.

c) **Déni de service :** Le déni de service (DoS, en anglais) apparaît lorsqu'un pirate désactive ou altère un réseau, des systèmes ou des services dans le but de refuser le service prévu aux utilisateurs normaux. Les attaques par déni de service mettent le système en panne ou le ralentissent au point de le rendre

inutilisable. Le déni de service peut consister simplement à supprimer ou altérer des informations. Dans la plupart des cas, l'attaque se résume à exécuter un programme pirate ou un script. C'est pour cette raison que les attaques par déni de service sont les plus redoutées.

d) **Vers, virus et chevaux de Troie** : Des logiciels malveillants peuvent être installés sur un ordinateur hôte dans le but d'endommager ou d'altérer un système, de se reproduire ou d'empêcher l'accès à des réseaux, systèmes ou services. Ces programmes sont généralement appelés vers, virus et chevaux de Troie.



FigureI.10 : type d'attaque d'un réseau [50]

.

II.7.4 les attaques sur les mots de passe

a) L'attaque par dictionnaire : le mot testé est pris dans une liste prédéfinie contenant les mots de passe les plus courants et aussi des variantes et aussi des variantes de ceux-ci (à l'envers, avec un chiffre à la fin, etc.). ces listes sont généralement dans toutes les longues les plus utilisées, contiennent des mots existants, ou des diminutifs.

b) L'attaque par force brute : toutes les possibilités sont faites dans l'ordre jusqu'à trouver la bonne solution (par exemple de 'aaaaa' jusqu'à 'zzzzz' pour un mot de passe composé strictement de six caractères alphabétiques).

II.8 Les étapes d'une attaque : [12]

Généralement toute attaque suit :

a) **Identification de la cible** : cette étape consiste a récolté un maximum de renseignements sur la cible.
b) **Le scanning** : il sert à compléter les informations (adresse IP, service accessible, OS,…) sur la cible.
c) **L'exploitation** : comme son nom l'indique cette étape permet d'exploiter les failles identifiées.
d) **La progression** : élever ses droits vers le root (système) afin de faire tout ce qu'il souhaite.
e) **Préservation d'accès** : à fin de facilité le retour aux systèmes compromis les attaquants créent des portes dérobées (failles).
f) **Effacement des traces** : une foi l'exploitation est terminée l'attaquant essai d'effacer ses traces toute en restituant les propriétés des fichiers.

Exemple d'attaque : **[13]**

- **La technique dite du "smurf"** :

La technique du "smurf" est basée sur l'utilisation de serveurs broadcast pour paralyser un réseau. Un serveur broadcast est un serveur capable de dupliquer un message et de l'envoyer à toutes les machines présentes sur le même réseau que lui. Le scenario s'une attaque est le suivant :

La machine attaquante envoie un ping (le ping est un outil du monde UNIX pour tester les machines d'un réseau en envoyant un paquet et en attendant la réponse) à un (ou plusieurs) serveurs broadcast en falsifiant sa propre adresse IP (l'adresse à laquelle le serveur devrait théoriquement répondre par un pong) et en fournissant l'adresse IP de la machine cible. Lorsque le serveur broadcast va dispatcher le ping sur tout le réseau, toutes les machines du réseau vont répondre par un pong, que le serveur broadcast va rediriger vers la machine cible. Ainsi lorsque la machine attaquante adresse le ping à plusieurs serveurs broadcast situés sur des réseaux différents, l'ensemble des réponses de tous les ordinateurs des différents réseaux vont être routées sur la machine cible.

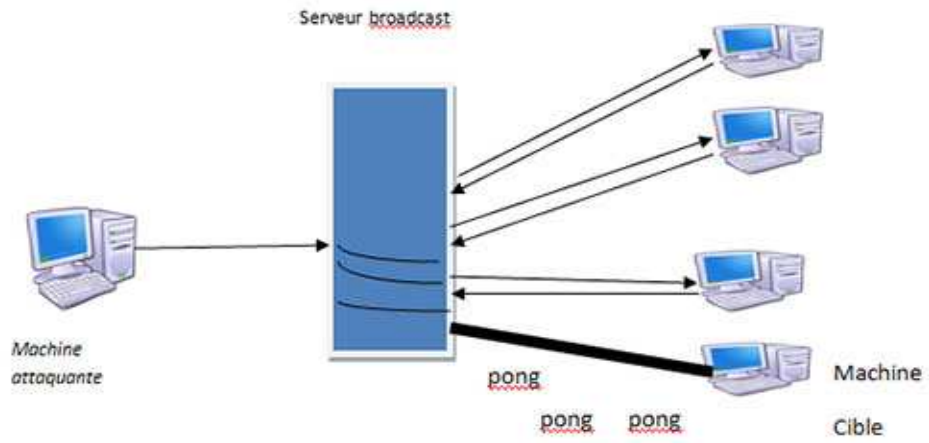


Figure I.11 : exemple d'attaque

II.9 Les types d'attaques : [14]

- **Les attaques directes :**

C'est la plus simple des attaques. Le hacker attaque directement sa victime à partir de son ordinateur. La plupart des « script kiddies » utilisent cette technique. En effet les programmes de hack qu'ils utilisent ne sont que faiblement paramétrable. Et un grand nombre des logiciels envoient directement les paquets à la victime.

Si tu te fais attaquer de la sorte, il y a de grand chance pour que vous puissiez remonter à l'origine de l'attaque, identifiants par la même occasion l'identité de l'attaquant.

- **Les attaques indirectes par rebond :**

C'est l'attaque la plus utilisé par les hackers. En effet, le rebond à deux avantages :

- ✓ Masquer l'identité (l'adresse IP) du hacker.
- ✓ Eventuellement, utiliser les ressources de l'ordinateur intermédiaire car il est plus puissants (CPU, bande passante …) pour attaquer.

Le principe en lui-même est simple, les paquets d'attaques sont envoyés à l'ordinateur intermédiaire, qui représente l'attaque vers la victime. D'où le terme de rebond.

- **Les attaques indirectes par réponses :**

Cette attaque est un dérivé de l'attaque par rebond. Elle offre les mêmes avantages, du point de vue de hacker. Mais au lieu d'envoyer une attaque à l'ordinateur intermédiaire pour qu'il la répercute, l'attaquant va lui envoyer une requête. Et c'est cette réponse à la requête qui va être envoyé à l'ordinateur victime. Dans ce cas de figure aussi, il n'est pas aisé de remonter à la source.

II.10 les mécanismes de sécurité : [14] [15]

- **Les Firewalls :**

C'est un élément (logiciels ou matériels) du réseau contrôlant les communications qui le traversent. Il a pour fonction de faire respecter la politique de sécurité du réseau, celle-ci définissant quels sont les communications autorisées ou interdites. N'empêche pas un attaquant d'utiliser une connexion autorisée pour attaquer le système. Ne protège pas une attaque venant du réseau intérieur.

Il permet aussi d'isoler les différents réseaux d'entreprise en mettant en place des architectures systèmes pare-feux on parle ainsi de « cloisonnement des réseaux ».

Il existe différents types de firewalls :

- Firewalls avec retour de filtrage.
- Passerelle double-le réseau bastion.
- Firewalls avec réseau de filtrage.
- Firewalls avec sous réseau de filtrage.

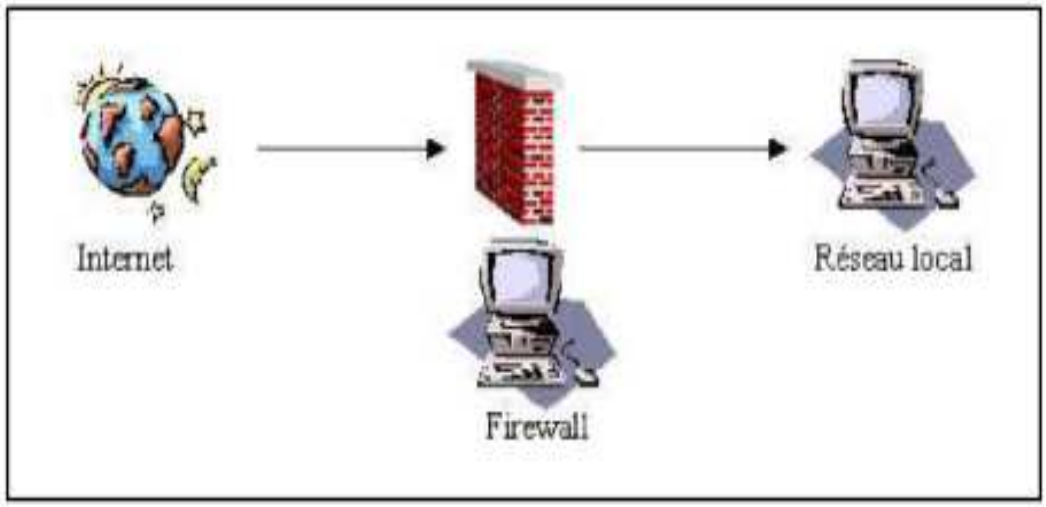


Figure II.12 : fonctionnement d'un firewall **[51]**

- **les VPN (Virtual Private Network) :**

Un VPN (réseau privé virtuelle) est une abstraction qui permettant d'isoler un nombre fini d'ordinateur distant, comme si ils appartenaient un même réseau locale.

Un logiciel VPN crée un tunnel pour permettre aux ordinateurs appartenant aux VPN d'échanger les données entre eux, tout en utilisant des protocoles de tunnelisation (GRE,PPTP, L2F,IPsec, SSL/TLS, SSH, VPN-Q …).

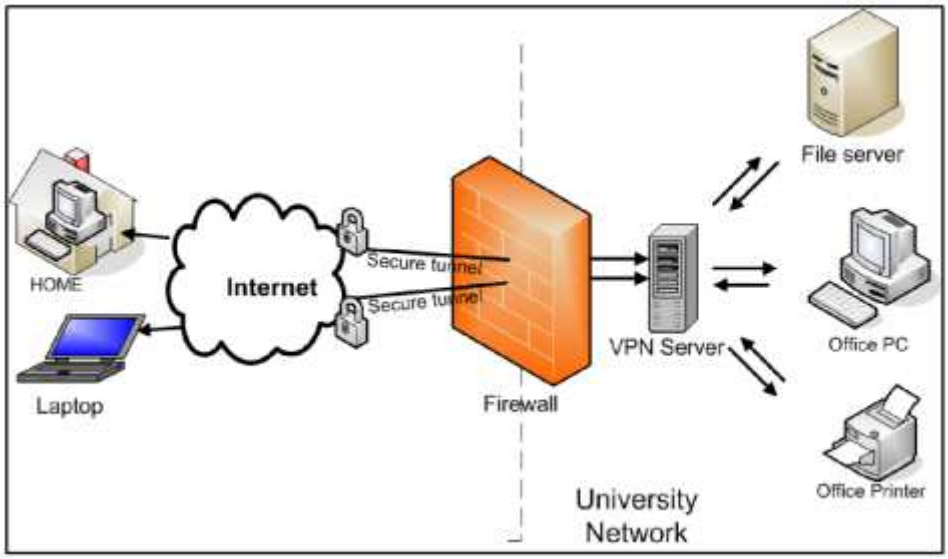


Figure II.13 fonctionnement de VPN [52]

- **proxy :**

C'est un serveur d'isolement qui sert de relais entre le réseau et les machines à cacher, son but est d'isoler une ou plusieurs machines afin de mieux les protéger.

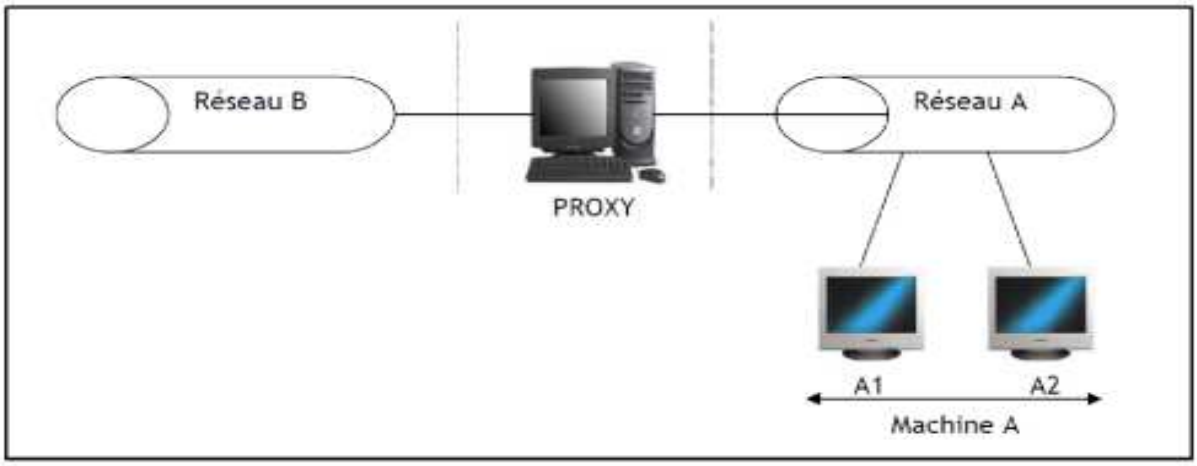


Figure II.14 : fonctionnement de proxy [53]

- **Chiffrement et Déchiffrement : [16]**

Les données qui peuvent être lues et comprises sans mesures spéciales sont appelées texte clair (ou libellé). Le procédé qui consiste à dissimuler du texte clair de façon à cacher sa substance est appelée chiffrement [dans le langage courant on parle plutôt de cryptage et de ses dérivés : crypter, décrypter]. Chiffrer du texte clair produit un charabia illisible appelé texte chiffré (ou cryptogramme). Vous utilisez le chiffrement pour garantir que l'information est cachée à quiconque elle n'est pas destinée, même ceux qui peuvent lire les données chiffrées. Le processus de retour du texte chiffré à son texte clair originel est appelé déchiffrement.

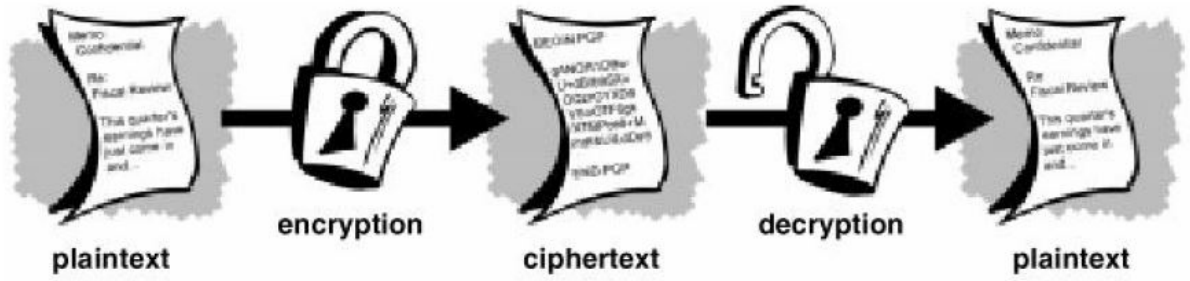


Figure II.15 chiffrement&déchiffrement [54]

- **La journalisation (logs) :**

Enregistrements des activités de chaque acteur sur des fichiers. Cela permet de constater que des attaques ont eu lieu de les analyser et potentiellement de faire en sorte qu'elles ne se reproduisent pas.

- **contrôle de routage :**

Sécurisation des chemins, c –a-d contrôler tous les liens (supports de transmission) et tous les équipements d'interconnexions (routeur, passerelles,…).

- **Authentifications :**

Authentifier un acteur peut se faire en utilisant une ou plusieurs de ces éléments

- ✓ Ce qu'il sait.par ex : son mot de passe, la date d'anniversaire de sa grand-mère,…
- ✓ Ce qu'il a par ex : une carte à puce.
- ✓ Ce qu'il est par ex : la biométrie (empreinte digital, oculaire ou vocal).

Dans le domaine des communications, on authentifie l'émetteur de message. Si l'on considère les (deux) extrémités d'une communication il faut effectuer une double authentification .par ex pour lutter contre le 'phishing'

L’authentification est nécessaire au bon fonctionnement des autres mécanismes.

I.10 conclusion :

Au cours de ce chapitre, nous avons présente la sécurité informatique, les risques et les menaces, les familles d’attaques, les étapes d’attaque, les types d’attaques ainsi que les mécanismes de sécurité

Malgré la multitude de mécanisme de sécurité existant actuellement, on ne peut pas garantir qu’un système soit protégé à 100%, c’est pour cela qu’il est conseillé d’utiliser une bonne combinaison de ces mécanisme afin d’optimiser le résultat de la sécurisation.

Dans le chapitre suivant, nous allons présenter les différents systèmes de détection d’intrusion.

III.1. introduction :

La détection d'intrusions est un terme général qui désigne des méthodes automatiques qui ; basées sur l'analyse des séquences d'événements temps réel et/ ou enregistrés ; peuvent alerter l'administrateur de sécurité de possibles violations de sécurité.

Afin de détecter les attaques qui peut subir un système, il est nécessaire de disposer d'un logiciel spécialisé dont le rôle sera de surveiller les données semble suspectes.

Les logiciels qui sont les plus à même d'effectuer cette tache sont les systèmes de détections d'intrusion dit : les **IDS**.

III.2. les systèmes de détection d'intrusion : [17]

Un système de détection d'intrusion (IDS : intrusion détection system) est un système matériel ou logiciel permettant de détecter et signaler les attaques. En effet la détection d'intrusions est apparue au début des années 80, suite aux travaux d'Anderson et à ceux de Denning, qui posent les fondations de la détection d'intrusions. Et c'est au début des années 90 que les premiers produits commerciaux sont apparus.

Il permet ainsi d'avoir une action de prévention sur les risques d'intrusion

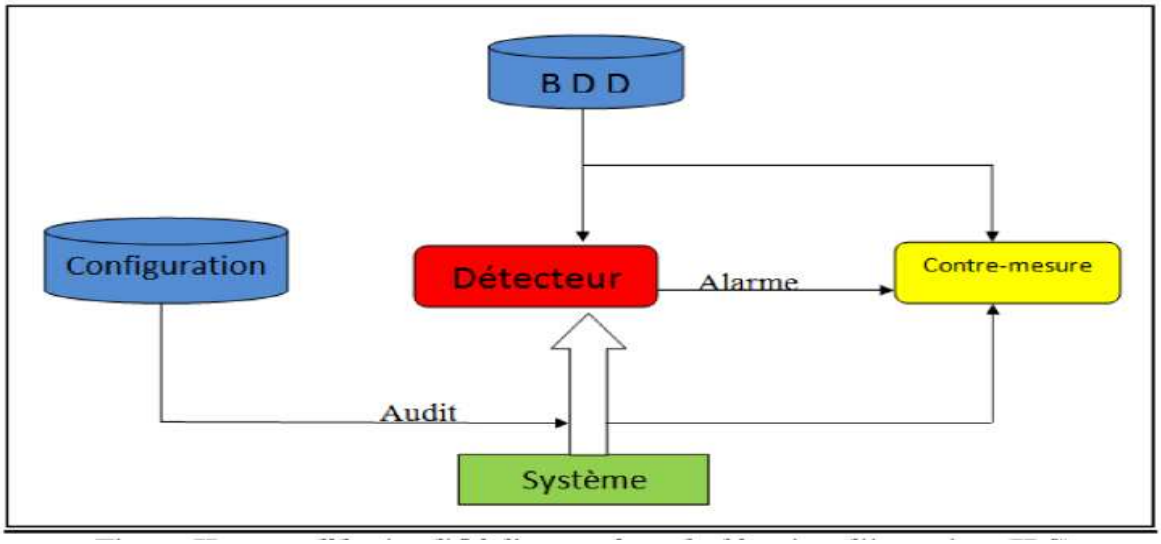


Figure III.1. Modèle simplifié d'un système de détection d'intrusion (IDS) **[55]**

III.3.historique : [18]

Les premiers systèmes de détection d'intrusions ont été initiés par l'armé américain puis par les entreprises.

Plus tard, des projets open-source ont été lancés comme **Snort** et **Prelude.** Des produits commerciaux ont aussi vu le jour par le biais d'entreprises spécialisées en sécurité informatiques : Internet Security Systems, *Symantecs, Cisco System,…

III.4 compositions d'un IDS :[19]

Il existe essentiellement trois composants dans un IDS

- **Le senseur :** il est responsable de la collecte d'informations du système tel que des paquets d'un réseau, ou des données d'un logiciel.
- **L'analyseur :** il reçoit l'ensemble des informations venant des senseurs. Il est responsable de les analyser et d'indiquer si une attaque a lieu ainsi qu'a éventuellement sa réponse.
- **L'interface utilisateurs :** elle permet au utilisateur de l'IDS de visualiser ou/et de définir le comportement du système.

III.5 positionnement : [19]

Bien que l'emplacement des IDS soit primordial pour une bonne sécurité du réseau, nous n'allons pas nous étaler sur ce sujet dans ce document.

Ainsi, il est conseillé de s'assurer que l'ensemble des informations transitant dans le système soit capturable par les IDS et que ceux-ci soit placé de manière optimale.

Pour protéger le système contre les attaques externes, le meilleur emplacement pour un IDS peut être après le retour ou le firewall.

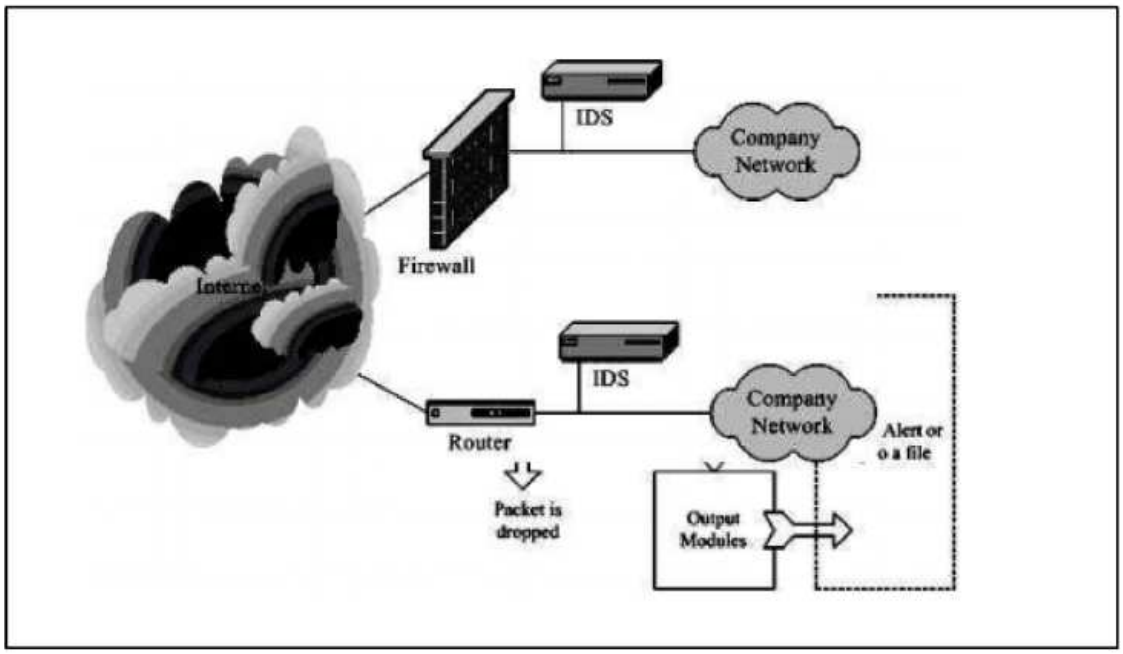


Figure III.2 positionnement des IDS [56]

III.6 caractéristiques d'un système de détection d'intrusions : [20]

Les caractéristiques suivantes sont souhaitables dans un IDS

- ✓ fonctionner en permanence avec une supervision manuelle minimale.
- ✓ Etre tolérant aux pannes dans le sens où il doit récupérer après une défaillance ou une réinitialisation de la machine.
- ✓ Résister aux tentatives de corruption, c'est-à-dire il doit pouvoir détecter si il a subit lui-même une modification indésirable.
- ✓ Utiliser un minimum de ressource de système sous surveillance.
- ✓ Etre facilement configurable pour implémenter une politique de sécurité spécifique d'un réseau.
- ✓ S'adapter aux cours du temps aux changements du système surveillé et du comportement des utilisateurs.
- ✓ Etre difficile à tromper

Comme la taille des réseaux a tendance à croitre, on peut ajouter les caractéristiques suivantes :

- ✓ Etre scalable.
- ✓ Etre robuste, c'est-à-dire que l'arrêt d'un composant ne doit pas entrainer une défaillance totale.

III.7. classification des systèmes de détections d'intrusions : [20] [21]

Les IDS peuvent être classés selon différentes critères qui ne sont pas mutuellement exclusifs, et ils sont :

1. Le principe de détection utilisé.
2. Le comportement en cas d'attaques détectés.
3. La source de données à analyser.
4. La fréquence de l'analyse.

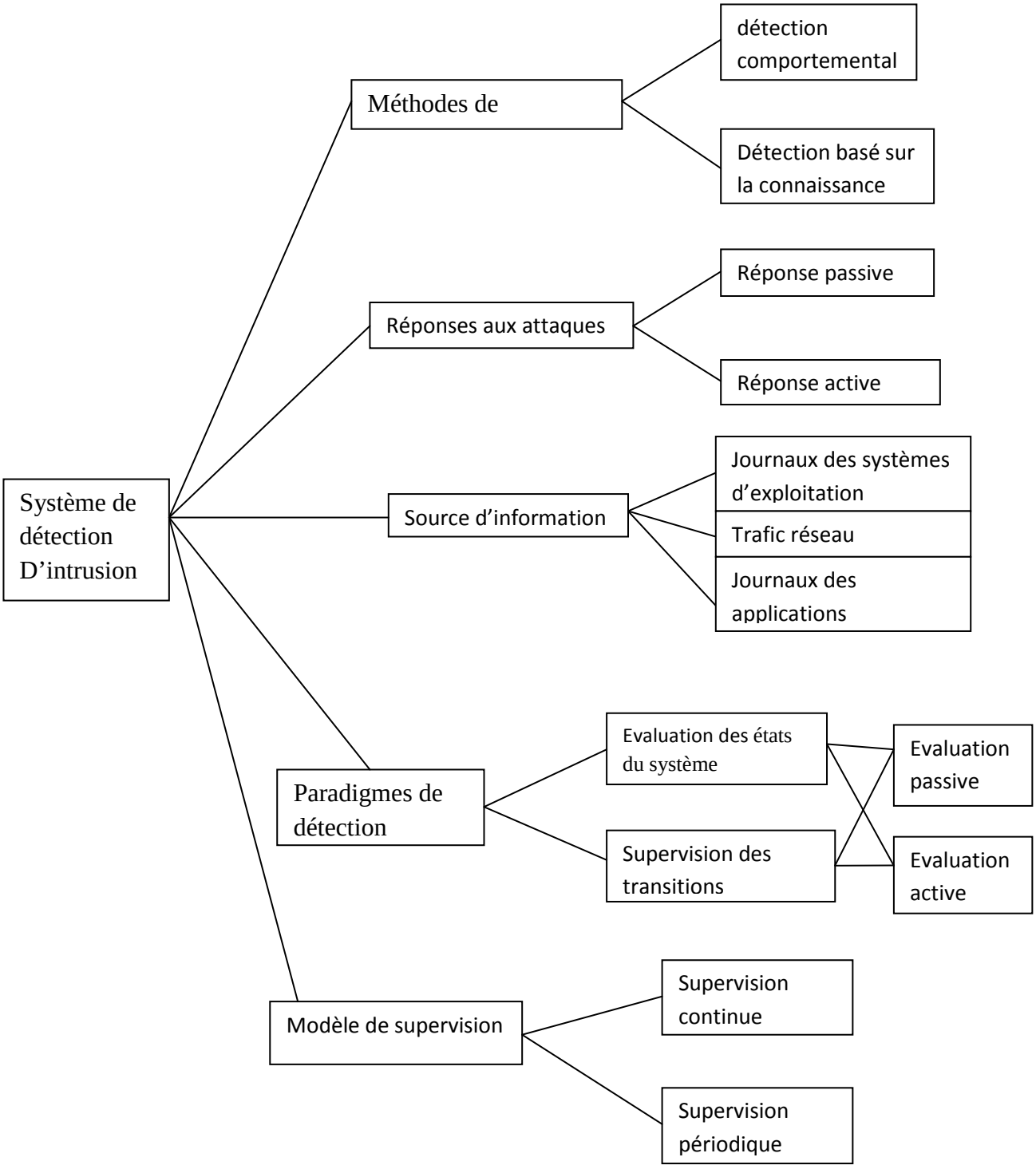


Figure III.3 classifications des système de détection d'intrusion [57]

III.7.1 classifications selon la méthode de détection :

On distingue deux approches majeures : l'une se base sur les signatures et on parle alors d'approche par sénario, et l'autre se base sur les profils normaux d'utilisation et on parle alors de l'approche comportementale.

- **L'aproche par scénario :**

Cette approche sur la connaissance des techniques utilisées par les attaquants pour déduire des sénarios typiques.
On l'appel détection par abus (knowledge Based Detection), elle est basée sur la determination d'une base de données contenants différentes signatures des différentes intrusions. Les IDS utilisant cette approche peuvent reconaitre les attaques d'après leurs base de signatures.

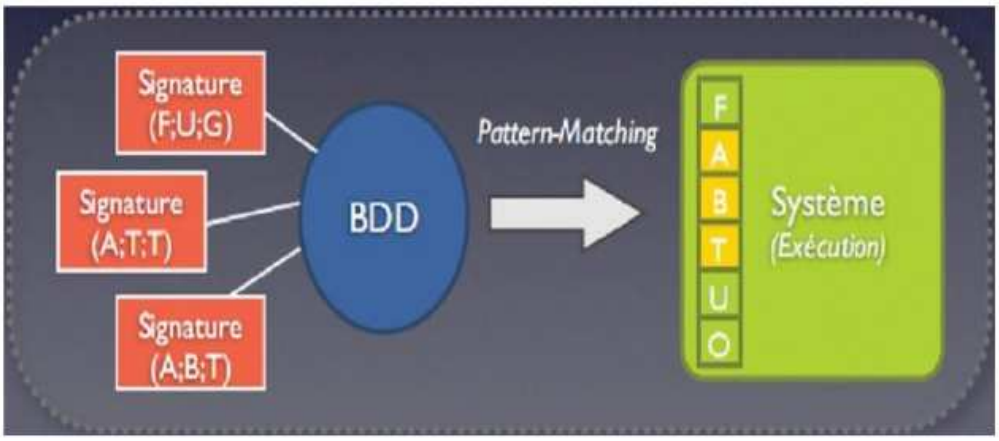


Figure III.4 :approche par sénario [58]

Les algorithmes utilisés dans cette approche peuvent etre regroupés en deux classes majeures.

a) Pattern matching :

Telle que E2XB, bayer-moore et Knuth-Morris-Prat.
C'est la méthode la plus connue et la plus facile, elle se base sur la recherche de motifs (chaine de caractères, suite d'octects) au seins de flux de données.

L'IDS comporte une base de signatures ou chaque signature contient les protocoles et les ports utilisés par l'attaque ainsi que le motif qui permettra de reconaitre les paquets suspects.

Le principale inconvénient de cette méthode et que seul les attaques reconues par leur signatures seront détectées, il est donc nécessaire de mettre a jour régulièrement la base de signatures.

b) Les systèmes experts :

Ces algorithmes servent à détecter les attaques des systèmes d'exploitations et non pas les attaques réseaux, ils consistent a sauvegarder les résultats des intrusions (..si…alors…) dans la base des signatures afin de modiliser une attaque.

Cette approche ne peut détecter que les attaques connus précédemment et qui se dérole selon la meme signature adapté a la base par exemple :si un attaquant change l'ordre des intrusions de son attaque, l'IDS trouveras une difficulté pour la détecter.

❖ **L'aproche comportemental :**

Le but de cette attaque est la prédiction de comportement.

La mise en œuvre d'un IDS comportemental comprend toujour une phase d'apprentissage au cour de la quelle il va découvrir le fonctionnement normal du système à surveiller, il va constituer un profil.

Ainsi des attaques inconnues peuvent etre détectées contrairement a l'approche par sénario.

Une foi le profil établis, tout comportement qui s'éloigne trop du comportement habituel déclenche une alarme de sécurité, hors, tout comportement inhabituel du système ne signifie pas forcément un comportement hostile, ce qui peut générer un nombre élevé de fausse alarme.

La création de profil peut se faire grace à différents paramètres tels que :

- La bande passante.
- La duré de connexion.
- Les ressources utilisées.
- Etc….

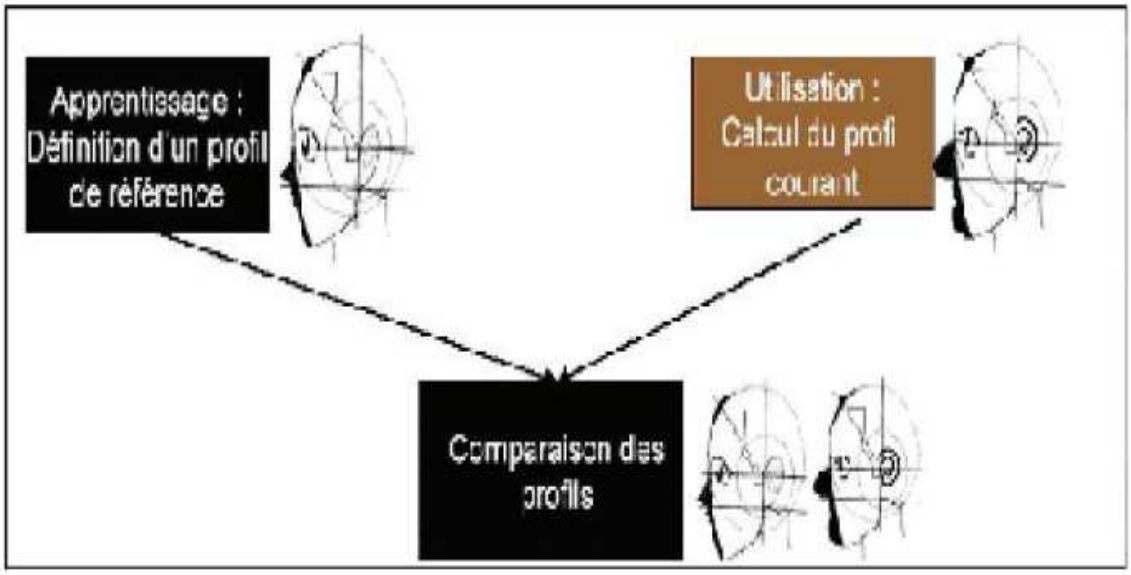


Figure III.5 Approuche coportemental [59]

Les techniques les plus connus sont :

1) Le modèle de DENNING :

C'est une technique basé sur les calcules statiques de consommation de ressources du système.

2) Réseau du neuronnes :

C'est une technique basé sur quelques paramètres importants tel que :

- ✓ Les ressources utilisées.
- ✓ La vitesse de saisie de clavier.
- ✓ Les activités faites par l'utilisateurs.

Comparaison entre les deux approuche

Scénarios	Comportements
Spécification complexe des scénarios	Taille des automates générés
Pas de faux positifs	Phase critique d'entraînement
Aucune prise en compte des nouvelles attaques	Prise en compte des nouvelles attaques
Mise à jour rapide	Mise à jour délicate (phase d'entraînement)
Protection facile à contourner	Faux positifs nombreux
Prise en compte incomplète des environnements parallèles	

Tableau III.6 comparaison entre l'approche comportemental
Et l'approche par sénario [60]

Conclusion sur les deux approche :

Vu la complémentarité des deux approches, l'idée d'hybridé l'approche comportemental avec l'approche par scénario à vite vu le jour afin de profiter des avantages de l'une comme de l'autre.

III.7.2 classification selon le comportement après la détection :

On peut classer les IDS selon leurs comprtements après la détection d'intrusion.

- **Réponses passives(les IDS passifs)**

Les IDS passifs ne peuvent pas réagir contre les attaques, ils se limitent plutôt a l'analyse des systèmes, la sauvegarde des signatures s'il en existe, et ils génèrent une alarme et notifient l'administrateur système, c'est alors lui qui devra prendre les mesures qui s'imposent.

- **Réponses actives (les IDS actifs) :**

Ce type d'IDS réagit aux attaques détectées.

En cas d'une attaque faible l'IDS alerte l'administrateur de système chargé de la sécurité, dans le cas d'une intrusion dangeureuse l'IDS doit réagir contre cette intrusion en éxcutant des actions tels que :

- La modification de la table de routage du retour lié au système.
- Le refus ou l'arret d'une connexion suspecte.
- L'arret d'un processus.
- La demande au pare-feu de modifier ces règles Etc…

Réponse passive	Réponse active
Emettre un rapport	Bloquer le compte d'un utilisateur
Générer une alarme	Suspendre des processus malveillants
Activer un archivage plus détaillé	Terminer une session
Activer un archivage à distance	Bloquer une adresse IP
Créer des fichiers de sauvegarde	Arrêter la machine
	Déconnecter la machine du réseau
	Mettre hors service les ports et les services attaquées
	Avertir l'utilisateur
	Tracer l'origine de la connexion
	Forcer une nouvelle authentification
	Restreindre les activités d'un utilisateur

Tableau III.7 réponses aux attaques des systèmes de détections d'intrusions[61]

III.7.3 classification selon la source de données à analyser :

la source de données à analyser est une caractéristique essentiel des IDS et un critère important pour leur classification.

Les données proviennent ; soit de fichier généré par le système d'exploitation et on parle alors d'IDS système(les HIDS : Hot Intrusion Detection System), soit de fichier généré par des applications, soit encore d'information obtenu en écoutant le trafic sur le réseau et on parle alors d'IDS réseau (les NIDS : Network Intrusion Detection System).

- **Les HIDS :**

Les systèmes de détection d'intrusion basé sur les hotes ou **HIDS** (HOST IDS)analysent exclusivement l'information concernant cet hote. Comme ils n'ont pas à controler le trafic du réseau mais seulement les activités d'un hote ils se montrent habituellement plus précis sur les types d'attaques subies.
De plus, l'impact sur la machine concernée est sensible immédiatement, par exemple dans le cas d'une attaque réussie par un utilisateur. Ces IDS utilisent deux types de sources pour fournir une information sur l'activité de la machine : les logs et les traces d'audit du système d'exploitation.

Chacun à ces avantages : les traces d'audit sont plus précises et détaillées et fournissent une meilleure information alors que les logs ne fournissent que l'information essentielle et sont plus petits. Ces derniers peuvent etre mieux controlés et analysés en raison de leur taille, mais certaine attaques peuvent passer inaperçues, alors qu'elles sont détectables par une analyse des traces d'audit.

Les avantages des HIDS sont les suivant :[23][24][25]

- Il est possible de constater immédiatement l'impact d'une attaque et donc de mieux réagir.
- Il est possible d'observer les activités se déroulant sur l'hote avec précision et d'optimiser le système en fonction des activités observées.
- Ils permettent de détecter plus facilement les attaques de type Cheval de Troi, alors que ce type d'attaque est difficilement détectable par un NIDS.
- Les HIDS peuvent souvent fonctionner dans des environnements avec un trafic réseau chiffré.
- Ils permettent également de détecter des attaques impossibles à détecter avec un NIDS, car elle font partie du trafic crypté.
- Ils génèrent peu de faux positifs, permettant d'avoir des alertes pertinentes .

Les inconvénients des HIDS : [23][24][25]

- Ils peuvent etre identifiés et met hors service par un attaquant.
- Ils ne peuvent donner l'alerte que si les entrées des journaux d'évenements ou les appels au système correspondent à une signature ou des règles pré configurées.
- Sensibles aux attaques de type Deni de Service.
- Ils sont assi gourmands en CPU et peuvent parfois altérer les performances de la machine hote.

Les HIDS sont en général placés sur des machines sensibles, susceptibles de subir des attaques et possédant des données sensibles pour l'entreprise. Les serveurs web et applicatifs peuvent notamment etre protégés par un HIDS.

voici quelque HIDS connus :

- Tripwire [24].
- WATCH [25].
- Security Manager [26].
- DragonSquire[27].
- Tiger[28].

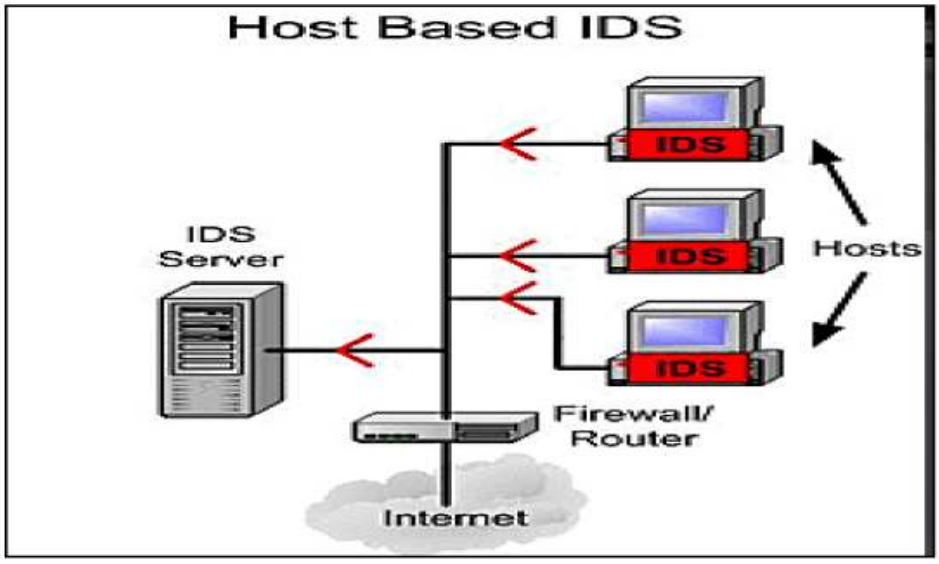


Figure III.8 Emplacement d'un HIDS (hoste IDS) [63]

❖ **Les NIDS :**

Le role essentiel d'un NIDS réseau est l'analyse et l'interprétation des paquets circulant sur ce réseau.

L'implantation d'un NIDS sur un réseau se fait de la façon suivante : des capteurs sont placés aux endroits stratégiques du réseau et génèrent des alertes s'ils détectent une attaque.

Ces allertes sont envoyées à une console sécurisée, qui les analyse et les traites éventuellement. Cette console est généralement située sur un réseau isolé, qui relie uniquement les capteurs et la console.

Les capteurs :

Les capteurs placés sur le réseau sont placés en mode furtif (ou stealth mode), de façon à etre invisibles aux autres machines. Pour cela, leurs cartes réseau sont configurées en mode promiscuous, c'est-à-dire le mode dans lequel la carte réseau lit l'ensemble du trafic, de plus aucun adresse IP n'est configurée.

Un capteur possède en général deux cartes réseaux, une est placée en mode furtif sur le réseau, l'autre permettant de le connecter à la console de sécurité. Du fait de leur invisibilité sur le réseau, il est beaucoup plus difficile de les attaquer et de savoir qu'un IDS est utilisé sur ce réseau.

Placement des capteurs : [29]

il est possible de placer les capteurs à différents endroits, en fonction de ce que l'on souhaite observer. Les capteurs peuvent etre placés avant ou après le pare-feu, ou encore dans une zone sensible que l'on veut protéger spécialement.

Si les capteurs se trouve après un pare-feu, il leur est plus facile de dire si le pare-feu à été mal configuré ou de savoir si une attaque est venue par ce pare-feu.

Les capteurs placés derrière un pare-feu ont pour mission de détecter les intrusions qui n'ont pas été arretées par ce dernier. Il s'agit d'une utilisations courante d'un NIDS.

Il est également possible de placer un capteur à l'extérieur du pare-feu (avant le firewall). L'intérêt de cette position est que le capteur peu ainsi recevoir et analyser l'ensemble de trafic d'Internet. Si vous placez le capteur ici, il n'est pas certain que toutes les attaques soient filtrées et détectées. Pourtant cet emplacement est le préféré de nombreux experts parce qu'il offre l'avantage d'écrire dans les logs et d'analyser les attaques (vers le pare-feu), ainsi l'administrateur voit ce qu'il doit modifier dans la configuration du pare-feu.

Les capteurs placés à l'extérieur du pare-feu servent à détecter toutes les attaques en direction du réseau, leur tache ici est donc plus de controler le fonctionnement et la configuration du firewall et d'assurer une protection contre toutes les intrusions détectés (certaines étant traitées par le firewall).

Il est également possible de placer un capteur avant et un autre après le firewall. Cette variante réunit les deux cas mentionnées ci-dessus.mais elle est très dangeureuse si on configure mal les capteurs et/ou le pare-feu, en effet, on ne peut pas simplement ajouter les avantages des deux cas précédents à cette variantes.

Les capteurs IDS sont parfois situés à l'entré de zones du réseau particulièrement sensibles (parcs de serveurs, donnés confidentielles…), de façon à surveiller tout trafic en direction de cette zone.

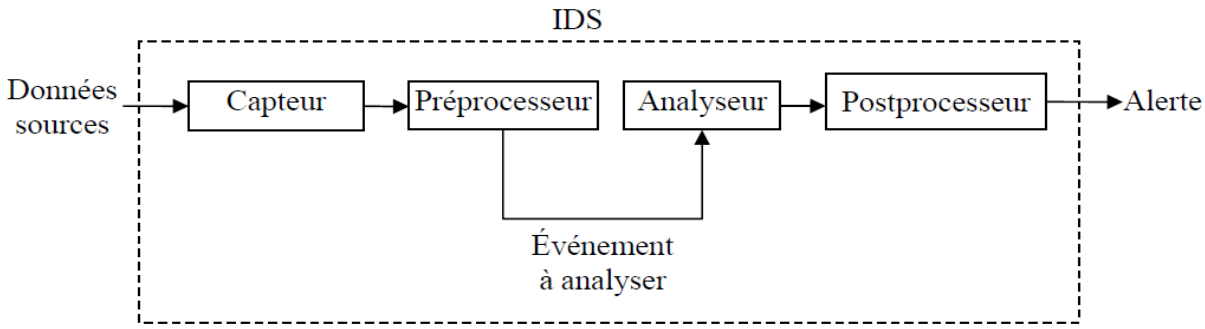


Figure III.9. Emplacement des capteurs dans IDS [64]

Les avantages des NIDS sont les suivant [23][24][25]

- Ils peuvent etre complètement caché sur le réseau, donc un attaquant ne saura pas qu'il controlé.
- Un système NIDS unique peut etre employé pour controler le trafic d'un nombre de système cibles potentiel.
- Il peut capturer le contenu de tous les paquets envoyés un à système cible.
- Une seule tache à effectuer : regarder le trafic et le traiter .
- Déployer un NIDS à un faible impact sur un réseau existant.
- Les NIDS sont des systèmes à temps réel.

Les inconvénients des NIDS [23][24][25] :

- Le temps élevés de faux positifs qu'il génèrent.
- Ils ne peuvent donner d'alarmes que si le trafic correspond aux règles ou aux signatures pré configurées.
- Ils peuvent manquer le trafic intéressant si le trafic est important sur la bande passante ou si des routes altérées sont utilisées.
- Il ne peut pas déterminer si une attaque à réussi.
- Il ne peut pas examiner le trafic chiffré
- Il faut des configurations spéciales sur les réseaux commuté pour que les NIDS puisse voir tout le trafic.

Voici quelques exemples de NIDS :

- Snort
- Benids
- Hank

- Prelude
- Firestom

❖ **Les IDS hybrides (NIDS+HIDS) :**

Les systèmes de détection d'intrusion hybrides rassemblent les caractéristiques de plusieurs systèmes de détections d'intrusions différents. En paratique, on ne retrouve que la combinaison de NIDS et HIDS. Ils permettent, en un seul outil de surveiller le réseau et l'hote.les sondes sont placées dans des points stratégiques, et agissent comme NIDS et/ou HIDS suivant leurs emplacements. Toutes les sondes remontent alors les alertes à une machine qui va centraliser, agréger, et lier les informations d'origines multiples.

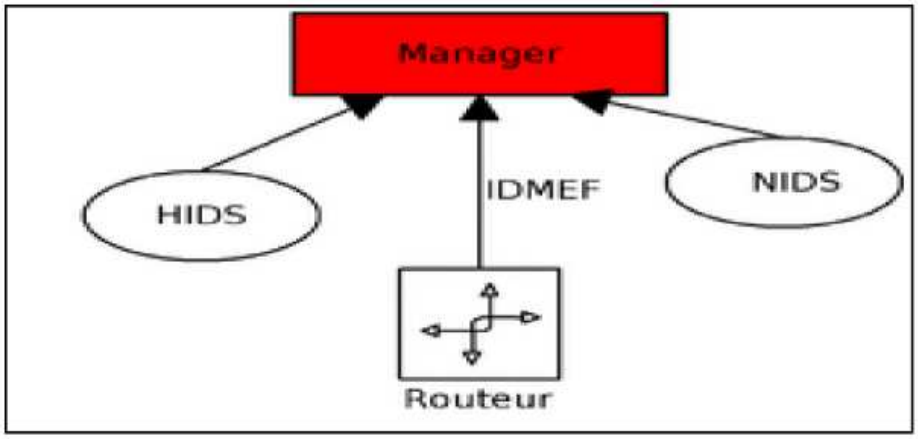


Figure III.10. fonctionnement d'un IDS hybride [65]

III.7.4 classification selon la fréquence d'utilisation :

On distingue deux type : Online offline

❖ **Les IDS Online (Continue) :**

Ce sont des IDS qui font l'analyse d'une façon continue ou permanente afin de détecter une attaque ou moment de sa production, c'est une détection en temps réel.

Ce type d'IDS consomme un taux élevé de resources système ce qui le rend non adéquat en cas de resources précieuses tel que les serveurs de messagerie.

❖ **Les IDS Offline (périodique) :**

Ce type d'IDS fait l'analyse dans des durées périodique (généralement en fin de journée) afin de détecter des traces d'attaques dans le but de modéliser des signatures d'attaques pour la base de système.

L'avantages de ce type d'IDS est qu'il ne consomme pas beaucoup de ressources système.

L'inconvénient de ce type est qu'il détecte les attaques en retard, ce qui peut provoquer des dégat dangeureux.

III.8 l'architecture des IDS [30]

On distingue trois architectures globales des IDS selon leur contrôle :

III.8.1 architecture centralisée :

Cette architecture a la meme démarche que de l'architecture client/serveur, c'est-à-dire que les IDS sont intallés dans des points stratégiques et sont gérés par un seul IDS administrateur alors que les IDS ne fait que la capture, des paquets les analyser, et fournir des messages à l'IDS administrateur qui vas répondre par un message d'alerte à l'administrateur du réseau ou bien par le lancement de contre mesure.

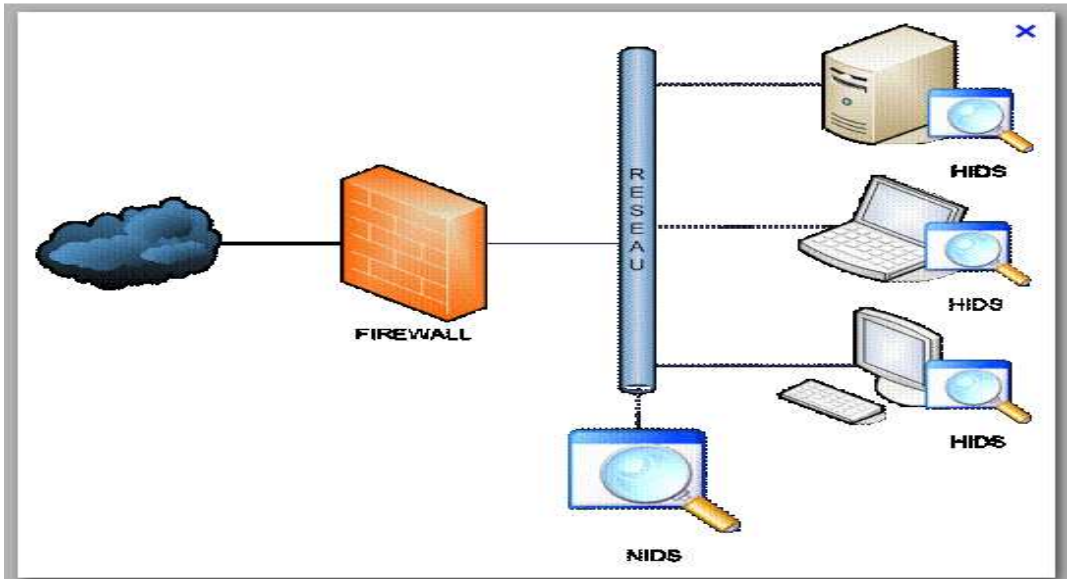


Figure III.11. : architecture centralisé d'un IDS [66]

III.8.2 architecture partiellement distribuée :

Ceci consiste à décomposé le réseau en sous-réseaux ou chacun possède son propre IDS, ces sous-réseaux sont classés d'une manière hiérarchique, alors que chaque IDS doit fournir ses messgaes a l'IDS d'ordre supérieur jusqu'à ce que les messages arriveront à l'IDS administrateur qui va produire les réponses possibles.

III.8.3 architecture totalement distribuée :

C'est une architecture composé de plusieurs architectures centralisées ou le réseau se décompose en sous-réseaux d'où chaque sous-réseau possède son propre IDS qui fait

capturer, analyser et fournir les réponses possibles sans faire transmit les messgae à un autre, cette architecture est efficace en cas des réseaux de grosse tailles.

- Exemple : l'architecture de la solution **Prelude** sur un modèle distribué composé d'un manager et de différentes sondes, les sondes sont chargé d'envoyer les informations relative aux évenements de sécurité au manager, qui se charge de l'analyse.

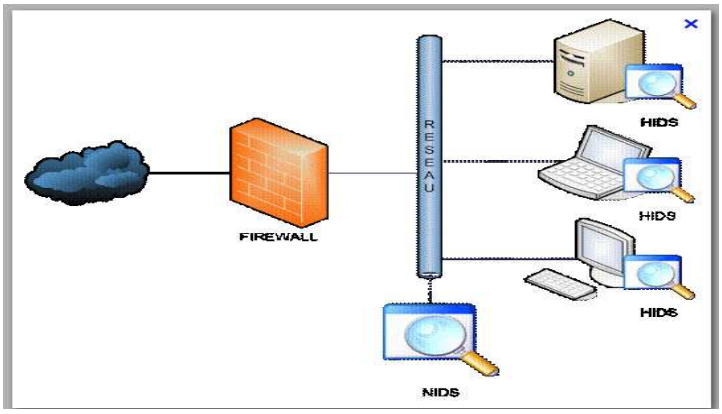


Figure III.12. architecture totalement distribué [67]

III.9 modèles et normalisations : [31]

III.9.1 CIDF :

C'est une architecture standardisée définie par DARPA (Defence Advenced Research Project Agency) afin de généraliser un modèle unique des IDS, ce modèle se compose de quatre composants appelés BOX

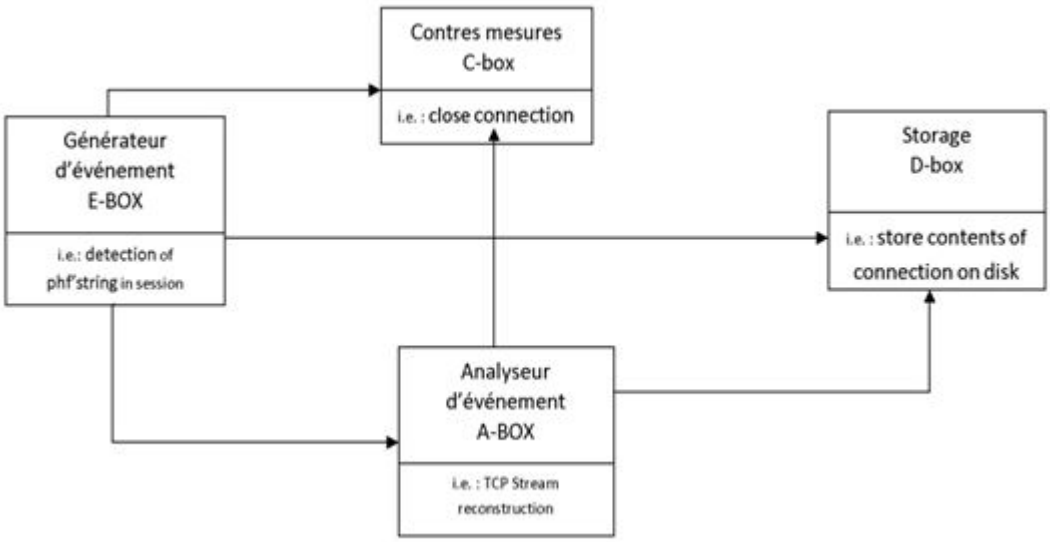


Figure III.13 architecture CIDEF [68]

a) Générateur d'évenements :(E-BOX)

c'est le protocole de bas niveau, il est le responsable de recupérer les paquets d'après le réseau afin de les utiliser par les autres composants

b) Analyseur d'évenement :(A-BOX)

son role est d'analyser les informations fournisses par le générateur afin d'extraires des doutes en basants sur l'une des deux approches de détection (par abus ou d'anomalie).

c) Stockages d'informations :(E-BOX)

Ceci permet de stocker les informations nécessaires provenant de la part de générateur et d'analyseur.

d) Contre mesures :(C-BOX)

Ceci est un cocept régulier, c'est-à-dire il peut exister et il ne peut pas, dans les deux cas l'IDS vas faire son travail normale. Ce concept permet de lancer les contres mesures appropriés pour chaque attaque.

III.9.2 IDMEF (Intrusion Detection Message Exchange Format) [45]

Ceci est une norme qui défini le format des données et des procédures pouvants etre partagées ou échangées entre deux éléments d'un IDS ou bien entre deux IDS. Ce language peut etre utilisé par un gestionnaire des informations de sécurité (SIM). Ce language est basé sur XML.

III.10 Test des IDS [32]

Avant la mise en place d'un IDS, il est nécessaire de tester ces limites. Pour cela, il existe plusieurs méthodes :

1. **Attaque :** nous allons utiliser les outils (tel que Nessus et Nmap) exploités par les attaquants pour détecter une faille dans le système ou dans l'IDS, telles que les techniques d'évasion ou d'insertion.
2. **Allarme :**nous allons regarder le taux des alarmes tel que les faux positifs.
3. **Qualité des informations :** nous allons regarder la qualité des informations fournies par l'IDS lors d'une alarme.
4. **Réalisme :** il est nécessaire de tester l'IDS dans un milieu réel et non pas uniquement avec un générateur d'informations tel que « Network Security Auditor ».
5. **Flexibilité et mise à jour :** il est souvent intéressant de pouvoir modifier les configurations d'un IDS telles que la base de signature,…c'est pourquoi il est aussi nécessaire d'avoir une bonne réactivité du constructeur en cas de nouvelle attaque non encore détectés par l'IDS.

6. **Qualités des signatures :** dans le cas d'un IDS se basant sur les signatures, il est nécessaire de pouvoir évaluer la qualités des signatures.
7. **Rapidité de système :** il est nécessaire que l'IDS soit capable de gérer un grand nombre de données en un temps raisonable et de détecter l'attaque en un minimum de temps pour réduire les dommages causés.
8. **Intégration :**puisque les IDS ne suffisent pas pour grantir l'ensemble de la sécurité, ils doivent etre facilement instalés et intégrés a son infrastructure.
9. **Interaction :** le nombre d'interaction entre un IDS et l'administrateur système doit etre minime.
10. **Dataset :**on va comparer les performance de l'IDS avec d'autres IDS grace à des datasets.

III.11 quelques IDS existant :[33]

le marcher des IDS est très vaste. Certains produits sont gratuits et d'autres payants. On se propose de présenter quelques logiciels tel que :

a) HAYSTACK :

se programme à été dévelloper de la part de l'Air Force, il est conçu pour détecter les intrusions dans un système multi utilisateur, sa base de signatures ne connait que six types d'intrusion.

- Lorsque un utilisateur non autorisé tente d'accéder au système.
- Un utilisateur autorisé tente de prendre une autre identité de celle-ci de lui.
- Un utilisateur veut modifier les paramètres de sécurité de système.
- Un utilisateur vient de tenter d'extraire des données potentiellement sensibles dans le système.
- Un utilisateur bloque l'accès aux ressources pour les autres utilisateurs.
- Autre attaques tel que l'efacement des fichiers.

Pour parvenir à ses fins Haystack utilise les deux méthodes de détections : par détection d'anomalies et par signatures. La détections d'anomalies utilisent un modèle par utilisateur décrivant le comportement de cet utilisateur dans le passé et un stéréotype qui spécifie le comportement générique acceptable pour cet utilisateur. Il est ainsi impossible à un intirus d'habiter le système à un comportement intrusif.

b) MIDAS :

MIDAS (Multics Intrusions Dectection Alerting System) est un système de détection heuristique, autrement dit que MIDAS est un système expert à base des règles appliquant le raisonnement de la détection heuristique. Il utilise un moteur de système expert à chainage avant appellant P-BEST (Production Based Expert System Toolset) et trois catégorie de règle.

❖ **Attaques immédiates :**

Ces attaques sont menées sans faire connaitre l'historique du sytème sur une petite fenetre d'évenement, généralement un, leur heuristique sont statiques ne pouvant etre échangées que par l'intervention d'un administrateur de sécurité.

❖ **Etat système :**

ses heuristiques à pour but de maintenir des informations sur les statistiques du système en générale sans faire intéresser à un utilisateur particulier.

c) IDES :

Le IDES (Intrusion Detection Expert System) est basé sur une hypothèse dite que le comportement d'un utilisateur reste le meme durant toute la durée d'utilisation. Sa méthode de calcul est statistique son role est de classer les comportement dans des groupe d'où chaque groupe contient un nombre de comportements proche entre eux, il observe trois types de sujets : l'utilisateur, les hotes distant ainsi que les système cible. Au total il mesure 36 paramètre : 25 pour l'utilisateur, 6 pour les hotes, 5 pour les système cibles. On peut classer ces mesures en deux grandes catégories :

✓ **Mesure catégorique :**

Sa nature est discrète alors que les valeurs obtiennes appartiennent à un ensemble fini. Par exemple les commandes faites par l'utilisateur.

✓ **Mesure continue :**

Ces mesures sont des fonctions réelles, par exemple le nombre de ligne imprimées pendant la session.

III.12. conclusion :

Dans ce chapitre nous avons présenté un état de l'art sur les systèmes de détection d'intrusion (les **IDS**) ou nous avons abordés plusieurs aspects tel que les approches de détections qui sont principalement : l'approche par scénario et l'approche comportementale, ainsi que les différentes architectures des IDS.

Plusieurs études ont lieu sur la façon de se protéger contre les intrus de tout part, et les IDS présente un bon mécanisme de sécurité.

Dans notre étude nous allons concevoir un système de détection d'intrusion basée sur les arbres de décisions, et dans le chapitre qui va suivre, nous allons expliquer d'une manière approfondie ce qui est un arbre de décision et quelle sont les mécanismes sur lesquels notre étude va etre basé.

Partie I : classification & arbre de décision

IV.1 introduction :

Au cours de toutes les études menées sur la sécurité informatiques et plus spécialement sur les IDS, ces derniers contiennent plusieurs déficiences, avec l'évolution des mécanismes d'attaque, et pour pouvoir compenser ces déficiences, les concepteurs font toujours recours aux nouvelles techniques d'apprentissage tel que les techniques de classifications.

Il existe plusieurs techniques de classification : [34]

- **Apprentissage symbolique**
- **Réseaux bayésiens**
- **Réseaux de neurones**
- Machines à vecteurs supports
- **Arbres de décision**
- Bagging boosting
- Méthodes des différences temporelles
- **Evolution artificielle**
- **Algorithme « bandit »**

Les arbres de décision sont une des techniques les plus populaires et les plus utilisées de l'apprentissage automatiques et la fouille de données, ainsi ont apporté un grand succès en grande marge à ses caractéristiques :

- ✓ **Lisibilité** du modèle de prédiction, l'arbre de décision, fournie. Cette caractéristique est très importante, car le travail de l'analyse consiste aussi à faire comprendre ses résultats afin d'emporter d'adhésion des décideurs.
- ✓ **Capacité** à sélectionner automatiquement les variables discriminantes dans un fichier de données contenant un très grand nombre de variables potentiellement intéressantes. En ce sens, un arbre de décision constitue une technique exploratoire privilégiée pour appréhender de gros fichiers de données.

Enfin, tout ces raison qui nous laisse centré notre thème sur cette technique de classification.

Dans cette partie nous aborderons par une définition sur l'apprentissage automatiques en générale pour entamer ensuite celle de l'arbre de décision, viendras après les étapes de l'algorithme de construction des arbres de décision, pour finir en citant quelque méthodes d'apprentissage et donner une conclusion.

IV.2.définitions : [35]

IV.2.1. L'apprentissage automatique

L'apprentissage automatique consiste à développer, analyser et implémenter des méthodes automatisables qui permettent a une machine de comprendre d'évoluer et de reproduire grâce a un processus d'apprentissage donc il est possible d'utiliser des techniques issues de se domaine pour découvrir et modéliser des connaissances, des observations et des données.

Types d'apprentissage :

Il existe plusieurs modes d'apprentissage employés par les algorithmes d'apprentissages :

- L'apprentissage supervisé
- L'apprentissage non-supervisé(ou classification automatique)
- L'apprentissage semi-supervisé
- L'apprentissage partiellement supervisé (probabiliste ou non)
- L'apprentissage par renforcement

Algorithmes utilisés :

Plusieurs algorithmes sont utilisés dans ce domaine tel que :

- Les machines à vecteur de supports
- Le boosting
- Les réseaux de neurones pour apprentissage supervisé ou non-supervisé
- La méthode des K plus proche voisins pour un apprentissage supervisé
- Les arbres de décision
- Les méthodes statistiques
- La régression logistique
- L'analyse discriminante linéaire
- Les algorithmes génétiques et la programmation génétique

IV.2.2. Les arbres de décision : [36]

Les arbres de décision est un algorithme de classification supervisé qui est souvent utilisé pour représenter des connaissances, des informations ou encore des observations qu'on appel aussi des exemples.

Un arbre de décision est une représentation sous forme graphique d'un digramme illustrant des règles de décision, il est composé de :

- **Nœuds de décision :** autrement dis « nœuds internes » chaque nœud est étiqueté par un test portant généralement sur un seul et unique attribut.

- **Branches :** ce sont des arcs issus des nœuds de décision correspondant à l'une des valeurs possibles des attributs sélectionné.

- **Nœuds feuilles :** comprenant des objets qui appartiens a la même classe.

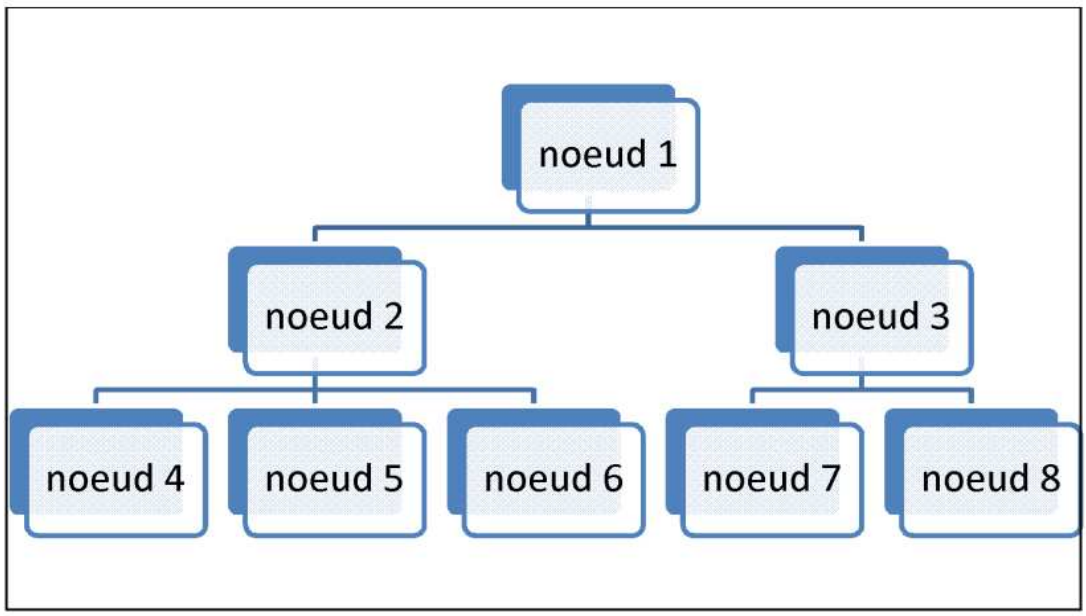


Figure IV.1 : représentation graphique d'un arbre de décision

IV.3. Petit historiques : [37]

Plusieurs recherches ont été lancé dans le domaine des mathématiques et de la programmation informatique sur les arbres de décision, afin de trouver l'algorithme le plus efficace et le plus optimal de l'ordre de segmentation des options intermédiaires.

Dates	auteur	Apport
1963	Morgan et Sonquist	Arbres de décision dans un processus de prédiction et d'explication
1980	Gordan V. Kass	CHAID (Chi-squared Automatic Interaction Detector)
1983	Quinlan	Théorie de la décision : Algorithmes et arbres de décision via ID3
1984	Breiman	CART (Classification And Regression Tree)
1993	Quinlan	Amélioration ID3
2001	Breiman	Forêts Aléatoires

Figure IV.2 : petit historique sur les algorithmes de classification [69]

IV.4. Etapes principale d'utilisation des arbres de décision :

L'utilisation des arbres de décision dans les problèmes de classification passe par deux étapes principales [38] :

IV.4.1 Etape de construction : [38][39]

Avant d'utiliser l'arbre de décision, il faut bien évidement passé par l'étape principale qui est bien sur sa construction. Cette étape se base sur un ensemble d'apprentissage et une méthode « algorithme » prédéfinie pour effectuer les taches suivantes

a) Choix de la variable de segmentation :

Pour bien fixer les idées, nous mettons de cotés le cas des variables continues. La quasi-totalité des méthodes d'induction d'arbres s'appuient sur une technique très fruste : chacun des méthodes teste toutes les variables potentielles et choisi celle qui maximise un critère donné. Il faut donc que le critère utilisé caractérise la pureté(ou le gain en pureté) lors de passage du sommet à segmenter vers les feuilles produites par la segmentation.

Il existe un grand nombre de critères informationnels ou statistiques, les plus utilisés sont l'entropie de Shannon et le coefficient de Gini et leurs variantes.

b) Traitements des variables continues :

Le traitement des variables continues doit être en accord avec l'utilisation du critère de segmentation. Dans la grande majorité des cas, le principe de segmentation des variables continue doit être très simple :

- Trier les données selon la variable à traiter.
- Tester tous les points de coupure possibles situés entre deux points successifs.
- Evaluer la valeur de critère dans chaque cas.

Le point de coupure optimal correspond tout simplement à celui qui maximise le critère de segmentation.

c) Définir la bonne taille de l'arbre :

Cette étape consiste a comme règle d'arrêt de construction de l'arbre la constitution de groupes pure de point du vue de la variable à prédire.

Plusieurs expériences ont affirmées que les performances d'un arbre de décision reposaient principalement sur la détermination de sa taille.

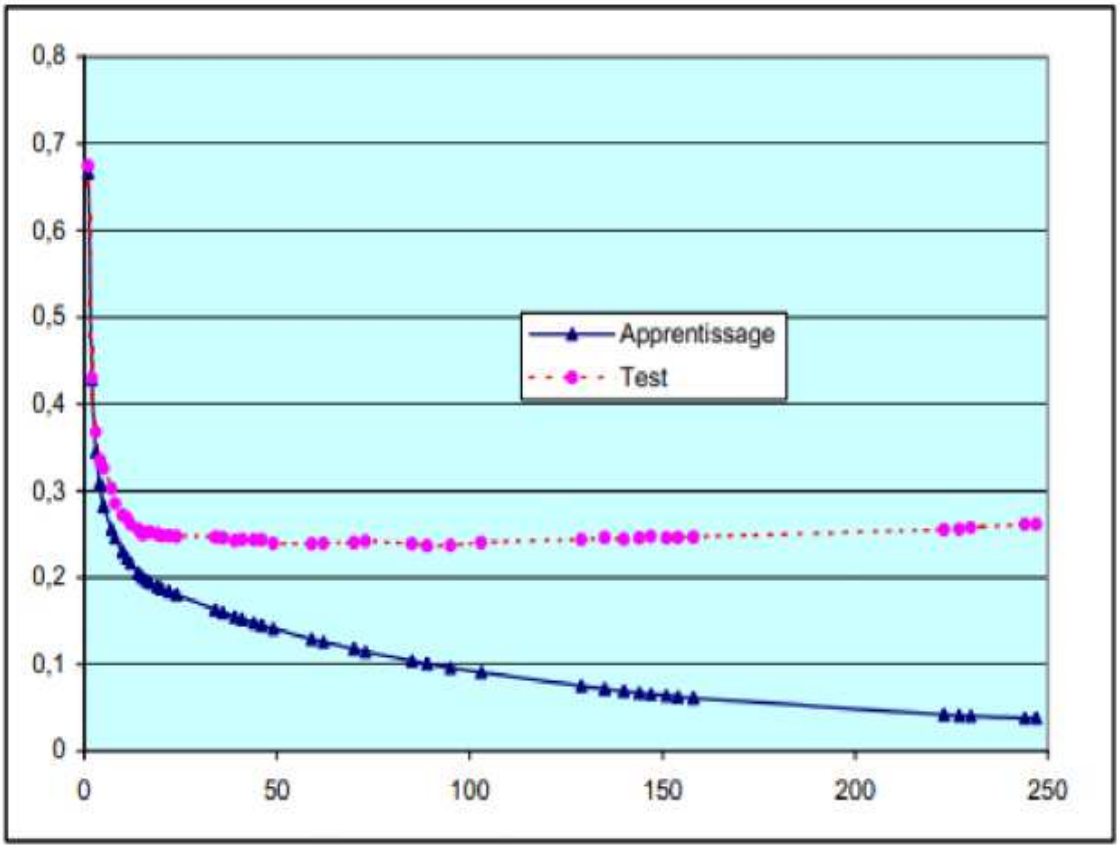


Figure IV.3. Évolutions de taux d'erreur en apprentissage et en test en fonction de nombre de feuilles [70]

Nous voyons effectivement dans cette figure qu'à mesure que le nombre de feuilles (la taille de l'arbre) augmente, le taux d'erreurs calculé sur les données d'apprentissage diminue constamment. En revanche, le taux d'erreurs calculé sur l'échantillon test montre d'abord une décroissance rapide, jusqu'à un arbre avec une quinzaine de feuilles, puis nous observons que le taux d'erreurs reste sur un plateau avant de se dégrader lorsque l'arbre est manifestement surdimensionné.

Ainsi, lorsque l'on construit un arbre de décision, on risque ce que l'on appelle un sur-ajustement du modèle : il faut toujours trouver l'arbre le plus petit possible (donc le plus stable dans ses prévisions future) ayant la plus grande performance possible.

Autrement dit, pour éviter un sur-ajustement de nos arbres, il convient d'appliquer un **principe de parcimonie** et de réaliser des arbitrages **performances/complexité** des modèles utilisés.

Dans le cas des arbres de décision, plusieurs types de solutions algorithmiques ont été envisagés pour tenter d'éviter au tant que possible un problème de sur-ajustement potentiel des modèles : il s'agit des techniques dites de prés ou de post élagages des arbres.

Le pré-élagage :

La première stratégie utilisable pour éviter un sur-ajustement massif des arbres de décision consiste à proposer des critères d'arrêt lors de la phase d'expansion. C'est le principe du pré-élagage. Autrement dit, faire un test statistique pour évaluer la segmentation introduit un apport d'information significatif quant à la prédiction des valeurs de la variable à prédire.

Le post élagage :

La seconde stratégie consiste à construire l'arbre en deux temps : produire l'arbre le plus pur possible dans une phase d'expansion en utilisant une première fraction de l'échantillon de données ; puis effectuer une marche arrière pour réduire l'arbre, en s'appuyant sur une autre fraction des données (échantillon d'élagage) de manière a optimiser les performances de l'arbre.

L'élagage rend d'une part l'arbre de décision plus simple et plus petit. D'autre part, il aide à éviter l'**Overfitting** lors du classement d'un nouveau cas.

d) Affectation de la conclusion de chaque feuille :

Une foi la construction de l'arbre est achevé, on procède a la précision de la règle d'affectation dans les feuilles c'est-à-dire définir la classe libellant chaque feuille :

- Si elles sont pures, la réponse est évidente.
- Sinon, une règle simple est de décider comme conclusion de la feuille la classe majoritaire, celle qui est la plus représentée.

IV.4.2. Etape de classification :

Cette étape consiste à classée des objets, tout en parcourus l'arbre de décision en descendu de la racine vers les nœuds plus bas jusqu'aux feuilles, en répondant aux différents tests qui libelle chaque nœud selon les valeurs des attributs de l'objet à classer.

IV.5. Algorithme d'apprentissage par arbre de décision [40] :

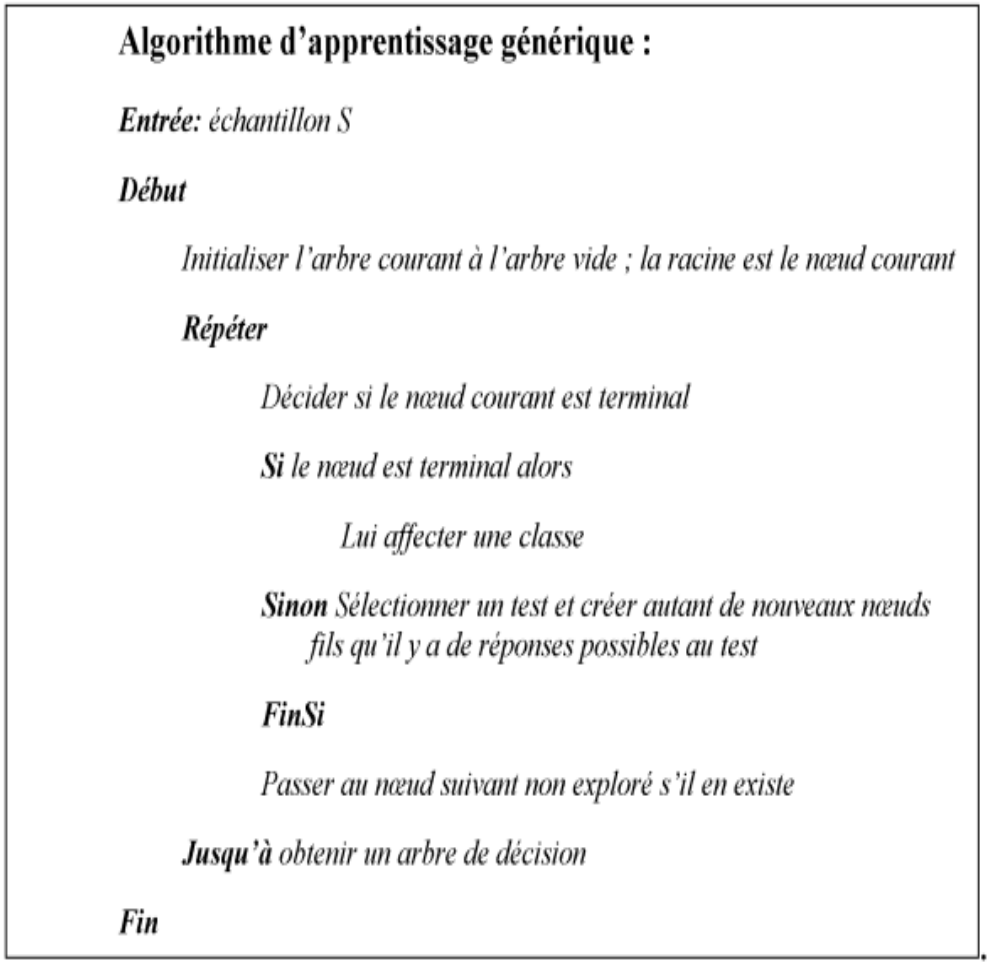
L'idée centrale de construction des arbres de décision consiste à diviser récursivement et plus efficacement possible les exemples de l'ensemble d'apprentissage par des tests définis à l'aide des attributs jusqu'à ce que l'on obtiennent des sous-ensembles d'exemples ne contenant(presque) que des exemples appartenant à une classe. Cette idée débouche sur des méthodes de construction **Top-down** c'est-à-dire construisant l'arbre de la racine vers les feuilles, gloutonne et récursive.

En général dans toutes les méthodes d'apprentissage par arbre de décision, on trouve les trois principaux opérateurs suivants :

1. Décider si un nœud est terminal : c'est-à-dire décider si un nœud doit être étiqueté comme une feuille ou porter un test.

2. Si un nœud n'est pas terminal sélectionner un test à lui associer.
3. Si un nœud est terminal, lui affecter une classe.

On peut alors définir un schéma général d'algorithme, sans spécifier comment seront définis les trois opérateurs décrits plus haut :



Algorithme d'apprentissage générique :

Entrée: *échantillon S*

Début

Initialiser l'arbre courant à l'arbre vide ; la racine est le nœud courant

Répéter

Décider si le nœud courant est terminal

Si *le nœud est terminal alors*

Lui affecter une classe

Sinon *Sélectionner un test et créer autant de nouveaux nœuds fils qu'il y a de réponses possibles au test*

FinSi

Passer au nœud suivant non exploré s'il en existe

Jusqu'à *obtenir un arbre de décision*

Fin

Figure IV.4. Algorithme d'apprentissage général [71]

IV.6. Les méthodes d'apprentissage [41][42] :

IV.6.1. Méthode ID3 :

Publié par Ross Quinlan. L'algorithme ID3 construit l'arbre de décision d'une manière récursive. A chaque étape de la récursion, il calcul parmi les attributs restant pour la branche en cours, celui qui maximisera le gain d'informations. C'est-à-dire l'attribut qui permettra le plus facilement de classer les exemples a ce niveau de cette branche de l'arbre. On appel se calcul l'entropie de Shannon dont la formule est la suivante :

$$I_E(i) = -\sum_{j=1}^{m} \text{f(i,j) log f(i,j)}$$

IV.6.2 Méthode C.4.5

Technique développé par Ross Quinlan. L'algorithme C.4.5 est une amélioration d'ID3, notamment du point de vu de la facilité d'implémentation. Il est souvent désigner comme un classificateur statique car les arbres de décision créent par cet algorithme peuvent être utilisés pour la classification.

C.4.5 construit des arbres de décision de la même manière que ID3, en utilisant le concept d'information d'entropie. A chaque nœud intermédiaire de l'arbre, C.4.5 divise l'ensemble d'échantillons en sous-ensembles enrichi en une seule classe tout en choisissant un test approprié, le critère de division est bien le gain d'information normalisé (différence d'entropie).

IV.6.3. CHAID (Chi-squared Authomatic Interaction Detector)

Technique publié par Gordon V. kASS EN 1980. Elle cherche à effectuer les regroupements les plus pertinents en s'appuyant sur des critères statistiques, l'échantillonnage doit être suffisamment large de manière a ce que la taille de chaque groupe ne devienne pas trop petite, ce qui rendrait l'analyse plus fiable. CHAID est une technique qui peut être utilisée pour la prédiction et pour la détection d'interaction entre variables.

CHAID détecte l'interaction entre variables dans un jeu de données. En utilisant pour la prédiction ou pour la détection d'interaction entre variables.

IV.6.4. Méthodes CART (Classification And Regression Trees) [41][42][43]

C'est la méthode la plus performante et la plus répondus, elle a été développée par Breiman , Friedman, Olshen et Stone en 1984. Cette méthode permet d'inférer des arbres de décision binaires, c'est-à-dire : tous les tests étiquetant les nœuds de décision sont binaires.

Le langage de représentation est constitué d'un certain nombre d'attributs. Ces attributs peuvent être binaires, qualitatifs (à valeurs dans un ensemble fini de modalité) ou continus (à valeurs réelles). Le nombre de test à explorer va dépendre de la nature des attributs.

Sur un nœud **t,** les instances qui répondent **oui** à une question posée sur ce nœud sont associées à la partie gauche de l'arbre et les instances qui répondent **non** à une question posée sur ce nœud sont associées à la partie droite de l'arbre.

Nous supposant prédéfini un ensemble de tests binaires. Pour définir cet algorithme, nous allons définir les trois opérateurs que comporte sa phase d'expansion :

- **Phase d'expansion :** on dispose en entrée d'un ensemble d'apprentissage A. La fonction utilisée pour mesurer le degré de mélange est la fonction de Gini(ou indice d'impureté de Gini) définie comme suit :

$$\boldsymbol{Gini\ (p) = 1 - \sum_{k=1}^{c} P(k/p)^2}$$

Tel que : - p : la position de nœud

- C : nombre de classe
- P(k/p) : proportion des individus appartenant à la classe k parmi ceux de la position p

1. **Décider si un nœud est terminal :** un nœud p est terminal si :

Gini(p) <=i0 ou N(p)<=n0, ou i0 et n0 sont des paramètres a fixer.

2. **Sélectionner un test a associés a chaque nœud :** soit p une position et soit test un test. Si ce test devient l'étiqueté du nœud a la position p, alors on appel p.gauche (respectivement p.droite) la proportion d'éléments de l'ensemble des exemples associés à p qui vont sur le nœud en position p1 (respectivement p2). La réduction d'impureté définie par le test est identique au gain et définie par :

Gain(p,test) = Gini(p) - (p.Gauche * Gini (p1) + p.Droite * Gini (p2)).

Cette équation correspond à la définition de gain dans le cas de deux classes en choisissant pour fonction I la fonction de Gini. En position p (non maximal), on choisit le test qui maximise la quantité Gain (p,test).

a. **Affecter un classe à une feuille :** une fois la condition d'arrêt de construction de l'arbre est atteinte, on procède a l'affectation de la classe majoritaire.

IV.6.5.D'autre méthodes moins considérer come :

- Hunt
- C5
- SLIQ
- Exhaustive CHAID
- QUEST
- VFDT
- UFFT
- …….

Partie II : la base KDD

IV.1.introduction :

Dans notre thèse on se prépose de concevoir un IDS basée sur un arbre de décision en utilisant l'algorithme de classification supervisé **CART.** L'approche développée dans notre travail repose sur une phase d'apprentissage durant laquelle nous appliquons l'algorithme CART sur une base de données appelée : la base d'apprentissage KDD, afin de définir le profil du système. Mais avant, décrivant d'abord de la base de données qui été utilisée dans notre expérimentation.

IV .2 Que ce que le KDD ? [44]

KDD (knowledge Discovery in Database) est un processus d'extraction des connaissances à partir des données, il permet le stockage, la préparation l'analyse des données en utilisant de nombreuse techniques afin d'extraire les connaissances et les évaluer, pour cela il est très important de connaitre la différence entre les trois termes suivant :

- Données : valeur d'une variable pour un objet.
- Information : résultat d'analyse des données.
- Connaissances : ensemble d'informations acquiert par l'étude, L'observation ou bien l'expérience sur les variables.

IV.3 Description de la base KDD : [44] [45][46]

Les données utilisé dans cet article, sont celle de KDD'99 et sont orientées détection d'intrusion (KDD). Chaque ligne de code un flot de données (entre deux instants définis) entre une source (identifié par son adresse IP), sous un protocole donné (TCP, UDP…). Dans la suite de l'article, nous appellerons « connexion » chaque ligne de base KDD'99 suivant ainsi la description fournie par KDD. Chaque « connexion » est caractérisée par 41 attributs tels que sa durée, le type de protocole, etc. ces attributs sont été fixés suite à un travail, de fouille de données effectués par Lee et al. (Lee et al.,1999). A partir des valeurs de ces attributs chaque « connexion » dans KDD'99 est considérée comme étant une « connexion » normale ou bien une attaque.

Les données KDD sont en fait des données formatées fournies par DERPA. Ces données présentent sept semaines de données libellées pour l'apprentissage et deux semaines de données non libellées pour le test (correspond au trafic réseau simulant un réseau local d'US Air force).

La base de données KDD'99 recense 38 attaques possibles peuvent être regroupées en quatre catégories [63].

- **Déni de service – « Denial-of-service (DOS) » :**

Il s'agit d'empêcher par tous les moyens les utilisateurs de se servir des ressources disponibles en temps normal. Ces attaques a but purement « destructeur » et sont souvent très simple a mettre en place et donnent une sensation de puissances a l'attaquant, ce qui expliquent leur fréquence. Un exemple d'attaques DOS et le Smurf qui provoque un déni de service via des requêtes d'écho ICMP manipulées à une adresse de diffusion d'un réseau.

- **Les attaques de types « Remote TO Local Acces »(R2L) :**

Ce type d'attaque essaye d'exploiter la vulnérabilité du système afin de contrôler la machine distante. Comme exemple d'attaque R2L, il y a celle qui visent les failles des protocoles IMAP (Internet Message Access Protocole). Ces protocoles permettent à des utilisateurs d'accéder à leurs comptes de courrier depuis des réseaux internes ou externes.

- **Les attaques de type « User To Root Attacks »(U2R) :**

Ou l'attaquant essaye d'avoir les droits d'accès à aprtir d'un poste afin d'accéder aux systèmes. Un exemple d'attaque U2R est Rootkit, qui après avoir obtenu un accès ROOT pour l'intrus, remplace les commandes systèmes afin qu'il puisse revenir quand il le souhaite en tant que root (administrateur).

- **Reconnaissance-probing :**

Ces actions ne sont pas vraiment des attaques puisqu'elles ne sont pas « destructrice » au sens ou elles n'empêchent pas une entité de fonctionner correctement, mais permettent d'acquérir des informations parfois cruciales pour mener une attaque de plus grande envergure plus tard. Un exemple d'outils de reconnaissances probing est Satan (Security Administrateur Tool for Analyzing Net-work), qui est un analyseur de ports TCP/IP qui recherche sur des hôtes distants les failles de sécurité et les défauts de configuration courants.

DOS	Probing	R2L	U2R
Apache2	Ipsweep	Ftp_write	Buffer_overflow
Back	Mscan	Guess_passwd	Httptunnel
Land	Nmap	Imap	Loadmodule
Mailbomb	Portsweep	Multihop	Xterm
Neptune	Saint	Named	Perl
Pod	Satan	Phf	Ps
Processtable		Dict	Rootkit
Smurf		Snmpguess	
Teardrop		Spy	
Udpstorm		Sqlattack	
		Warezclient	
		Warezmaster	
		Xlock	
		Xsnoop	
		Guest	

Tableau IV.1 : types d'attaques [72]

La KDD contient deux types de bases de connexions : [46]

1. La base d'apprentissage KDD :

- Enregistrement :
- 41 attributs + nom de classe pour apprendre.
- Fichiers au format texte.
- 5 millions de connexions (10%(494000 utilisées))

- 4 classes d'attaques+ trafic normal

  - Probing : scan de port (nmap, satan,..)
  - Dos : déni de service (syn flooding, smurf…)
  - U2R : acquisitions des privilèges d'un super utilisateur (buffer overflow)
  - R2L : accès illégitime à partir d'une machine distante (password guessing)

- Normal trafic légitime
-

2. **La base de test KDD** [47]

- Enregistrement :
  - 41 attributs + nom de classe pour vérifier.
- 311000 connexions

  - 4 classes d'attaques enrichis + trafic normal

    - Probing : scan de port (mscan, saint)
    - Dos : déni de service (appache2)
    - U2R : acquisitions des privilèges d'un super utilisateur (sqlattack)
    - R2L : accès illégitime à partir d'une machine distante (snmpguess,snmpgetattack)
    - Normal trafic légitime

IV.4. les attribue caractérisant chaque connexion [47] [48]

Les attributs caractérisant chaque connexion de la base KDD, sont détaillés dont le tableau de la figure2. En effet on peut distinguer les attributs basiques des connexions TCP individuelles, les attributs relatifs au contenu, les attributs relatifs aux temps calculés en utilisant des fenêtres de temps de deux secondes et les attributs basés sur l'hôte, calculés en utilisant des fenêtres de temps de 100 connexions. Ces attributs sont utilisés pour caractériser les attaques qui scannent les hôtes(ou les ports) en utilisant un intervalle de temps plus large que deux secondes.

A1 duration	durée de la « connexion » (nb de secondes)
A2 protocol_type	type du protocole, ex. tcp, udp, icmp...
A3 service	service réseau (destination) ex. http, telnet
A4 flag	statut de la « connexion » (normal ou erreur)
A5 src_bytes	nb de données (en octets) de la source vers la destination
A6 dst_bytes	nb de données (en octets) de la destination vers la source
A7 land	1 si la « connexion » est de/vers le même hôte/port ; 0 sinon
A8 wrong_fragment	nb de fragments « erronés »
A9 urgent	nb de paquets urgents
A10 hot	nb d'indicateurs « hot »
A11 num_failed_logins	nb d'essais login ratés
A12 logged_in	1 si succès du login ; 0 sinon
A13 num_compromised	nb de conditions de « compromis »
A14 root_shell	1 si la racine shell est obtenue ; 0 sinon
A15 su_attempted	1 s'il ya tentative de la commande « racine su » ; 0 sinon
A16 num_root	nb d'accès à la « racine »
A17 num_file_creations	nb de créations d'opérations de fichiers
A18 num_shells	nb de shell prompts
A19 num_access_files	nb d'opérations sur les fichiers de contrôle d'accès
A20 num_outbound_cmds	nb de commandes outbound dans une session ftp
A21 is_hot_login	1 si le login appartient à la liste « hot » ; 0 sion
A22 is_guest_login	1 si le login est login « invité » ; 0 sinon
A23 count	nb de connex. pour le *même hôte*
A24 srv_count	nb de connex. pour le *même service*
A25 serror_rate	% de connex. pour le *même hôte* ayant l'erreur « SYN »
A26 srv_serror_rate	% de connex. pour le *même service* ayant l'erreur « SYN »
A27 rerror_rate	% de connex. pour le *même hôte* ayant l'erreur « REJ »
A28 srv_rerror_rate	% de connex. pour le *même service* ayant l'erreur « REJ »
A29 same_srv_rate	% de connex. pour le *même hôte* utilisant le *même service*
A30 diff_srv_rate	% de connex. pour le *même hôte* utilisant *différents services*
A31 srv_diff_host_rate	% de connex. pour le *même service* utilisant *différents hôtes*
A32 dst_host_count	nb de connex. pour le *même hôte*
A33 dst_host_srv_count	nb de connex. pour le *même hôte* utilisant le *même service*
A34 dst_host_same_srv_rate	% de connex. pour le *même hôte* utilisant le *même service*
A35 dst_host_diff_srv_rate	% de connex. pour le *même hôte* utilisant *différents services*
A36 dst_host_same_src_port_rate	% de connex. pour le *même hôte* ayant le port src
A37 dst_host_srv_diff_host_rate	% de connex. pour le *même hôte* et le *même service* utilisant *différents hôtes*
A38 dst_host_serror_rate	% de connex. pour le *même hôte* ayant l'erreur « SYN »
A39 dst_host_srv_serror_rate	% de connex. pour le *même hôte* et le *même service* ayant l'erreur « SYN »
A40 dst_host_rerror_rate	% de connex. pour le *même hôte* ayant l'erreur « REJ »
A41 dst_host_srv_rerror_rate	% de connex. pour le *même hôte* et le *même service* ayant l'erreur « REJ »

Tableau.VI.2 : listes des attributs [73]

VI.5. conclusion :

Dans ce chapitre, nous avons exposé dans sa première partie des généralités sur les arbres de décision et leur construction, puis on a cité quelques méthodes d'apprentissage en détaillant CART vu que c'est l'algorithme qu'on va utiliser dans le chapitre suivant.

Puis nous avons étalé dans sa deuxième partie la base d'apprentissage et de test KDD qui sera utilisée dans notre étude expérimentale et nous allons valider l'algorithme CART sur la base d'apprentissage KDD, afin de construire le profil normal du système à surveiller, ensuite utiliser la base de test KDD pour tester notre IDS.

V.1 introduction :

Nous proposons de créer un système de détection d'intrusion comportemental en utilisant une méthode de classification basée sur les arbres de décision.

L'IDS que nous allons concevoir sera nommé **IDSCART** (**système de détection d'intrusion basé sur l'algorithme CART**). **IDSCART** est un système de détection d'intrusion comportemental, donc il nécessite une phase d'apprentissage, pour cela, nous allons appliquer la méthode CART sur la base d'apprentissage KDD, que nous avons présenté dans le chapitre précédant afin de construire un arbre de décision qui modélise le comportement normal de système a surveiller. Cette application (cet arbre) consiste à classifier les différentes connexions de la base d'apprentissage KDD.

Il est important de tester notre système **IDSCART**, pour cela, nous utiliserons une base de test (base de test KDD) contenant des connexions normales et des connexions considérées comme étant des attaques.

V.2. l'objectif du présent travail

Notre système a pour but de sécuriser les systèmes informatiques, tout en essayant de satisfaire le maximum des caractéristiques d'un **IDS** qui sont les suivantes :

- Fonctionnements de manière continue avec une présence humaine minimale
- Détection des attaques.
- Extensibilité (possibilité d'ajout de terminaux à sécuriser sans pour autant mettre en péril la sécurité du réseau).
- Supervision de plusieurs stations tout en fournissant des résultats de manière rapide et précise.

V.3 structure IDSCART

Le système **IDSCART** est structuré de deux grandes phases :

1. Phase d'apprentissage : modélise le profil normal de fonctionnement du réseau.
2. Phase de test : permet de tester le système **IDSCART.**

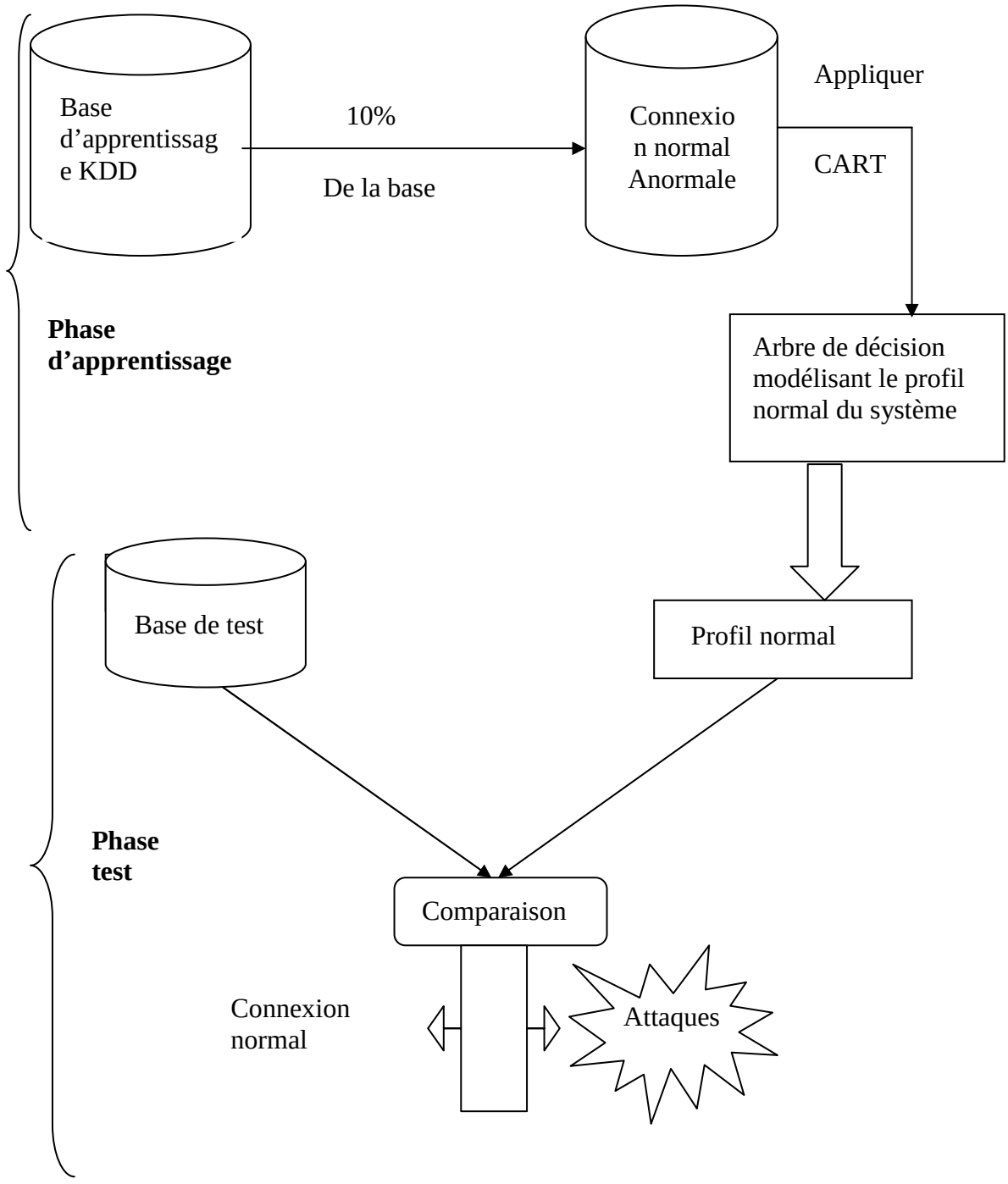


Figure V.1: structure IDSCART

1. Phase d'apprentissage :

Pour la modélisation du comportement normal de système, nous traitons 10% de la base d'apprentissage KDD (correspondant à 494019 de connexion). Nous devons d'abord construire la base des objets à partitionner.

Chaque objet correspond à une connexion, définie par un vecteur à 42 cases (les 41 premières pour les 41 attributs, la dernière pour spécifier si c'est un comportement normal ou bien une attaque) cette base des objets à partitionner est soumise à un classificateur basé sur l'arbre de décision construit selon l'algorithme de CART, pour former le profil normal de fonctionnement du système.

Exemple 1 :

Afin d'illustrer les différentes notions concernant les arbres de décision, et la méthode d'apprentissage que nous avons utilisé dans notre étude, nous allons considérer un exemple de base d'apprentissage donné dans le tableau ci-dessous, composé de quelques lignes de la base KDD'99. Chaque « connexion » pour des raisons de simplicité, est uniquement décrite par trois attributs continus (au lieu de 41 attributs que contient la base) qui sont :

- **A5 : src-bytes** nb de données (en octects) de la source vers la destination
- **A6 :dst-bytes** nb de données (en octects) de la destination vers la source.
- **A7 :land** 1 si la « connexion » est de / vers le même hôte/port ;0 sinon

Nous traitons uniquement deux classe {normal, anormale}

Src-bytes	Dst-bytes	Land	classe
181	5450	0	Normal=0
239	486	0	Normal=0
235	1337	0	Normal=0
219	1337	0	Anormal=1
217	2032	0	Anormal=1

figure V.2 : ensemble d'apprentissage

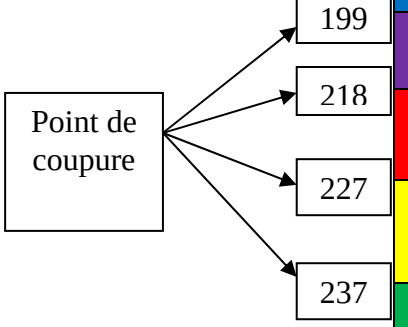
Nous allons utiliser l'algorithme de CART pour faire face à trois paramètres principaux :

- **Mesure de sélection d'attributs**
- **Stratégie de partitionnement**
- **Critère d'arrêt**

Pour cela :

- ✓ on agit sur la variable de segmentation A1
- ✓ On trie selon la variable sélectionné
- ✓ Nous calculons les points de coupure entre deux attributs qui succède
- ✓ Nous calculons gini et le gain pour ces points de coupure.

Le gain maximum sera l'attribut qui sera la racine de l'arbre.
Le tableau suivant illustre tous ces étapes :



Src-bytes	Dst-bytes	land	classe
181	5450	0	Normal=0
217	2032	0	Anormal=1
219	1337	0	Anormal=1
235	1337	0	Normal=0
239	486	0	Normal =0

Figure V.3 schémas récapitulatifs

- ✓ **Calcul gini**

Puis nous choisissons le point de coupure qui maximise l'indice de gini, calcul le gain pour se point, choix de la variable avec son seuil de test (point de coupure qui maximise le gain) qui optimise le gain. Finalement créer fils gauche et droite.

V.4. Définitions de la classe lecture :

Importer 10% de la base KDD'99 dans un arraylist et par la suite faire tous les traitements pour construire un arbre binaire modélisant le profil normal.

Exemple2 :

- public static List<connection> ListConnect = new ArrayList<connection>();

Cette arraylist vas contenir toutes 10% de la KDD, pour se faire, nous avons utilisé des bufferReader et utiliser le fichier d'entrée nommé **baseKDD.TXT** contenant 494019 connexions (97278 connexions normales et 396741 connexions anormales) et génère un arraylist contenant les connexions de la base KDD (10%). Après nous procéderons a changer tous les attributs qualitatifs en quantitatifs suivant la table de correspondance au dessous

Type du protocole	Service réseau	Statut de connexion	Nature de la connexion
Tcp=1	http=1	Sf=1	Normal=1
Udp=2	smtp=2	Rsto=2	Smurf=0
Icmp=3	telnet=3	So=3	Neptune=0
Private=4	finger=4	Sl=4	Loadmodule=0
Domain=5	ecr_i=5	S2=5	
	eco_i=6		

Tableau V.1: table de correspondance des attributs qualitatifs

Exemple3 :

Ligne de la base KDD avant l'application de la méthode :

0,tcp,http,SF,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,9,9,1.00,0.00,0.11,0.00,0.00,0.00,0.00,0.00,normal.

La même ligne après l'application de la méthode : `0 1 1 1 181 5450 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 8 8 0.00 0.00 0.00 0.00 1.00 0.00 0.00 9 9 1.00 0.00 0.11 0.00 0.00 0.00 0.00 0.00 1`

V.5.définition des classes (normale, anormale) :

La base KDD recense 38 attaques possible qui peuvent être regroupé en quatre catégories, **DOS, R2L, U2R, Probing** mais dans notre études on va regrouper tous ces attaques dans une seule classe qui sera appelé classe anormal ainsi une classe normal qui représente les connexions normaux.

- ✓ **Mesure de sélection d'attributs** qui permet de choisir l'attribut qui sera la racine de l'arbre. En outre, elle permet le choix des racines des sous arbres de décision. En fait, la mesure de sélection d'attributs permet de classer les attributs entre eux et de choisir celui qui partitionne l'ensemble d'apprentissage de manière optimale réduisant par conséquent la taille de l'arbre. Ainsi une bonne mesure doit permettre de limiter la taille de l'arbre et de donner une cohérence sémantique aux nœuds qui le composent en utilisant les deux fonctions suivantes :

$$\mathbf{GINI}(\boldsymbol{p}) \quad = \mathbf{1} - \sum_{\boldsymbol{k}=\mathbf{1}}^{\boldsymbol{c}} \mathbf{p}(\mathbf{k}/\mathbf{p})^{\mathbf{2}}$$

$$\mathbf{GAIN}(\mathbf{p}, \mathbf{test}) = \mathbf{gini}(\mathbf{p}) - (\boldsymbol{p}.\boldsymbol{gauche} * \boldsymbol{gini}\ (\boldsymbol{p}\mathbf{1}) + \boldsymbol{p}.\boldsymbol{droite} * \boldsymbol{gini}(\boldsymbol{p}\mathbf{2}))$$

Exemple4 : la classe qui fait le tri des variables de segmentation

```
public class TreeBinaire {
        BNode theBTRootNode;

 public TreeBinaire() // constructor

        {
theBTRootNode = null;
        }
                .}
```

Puis on va faire appel à cette classe de la classe principale.

Pb.populatebinTree(listTriee)

V.6.définition de la classe un nœud :

Chaque nœud de l'arbre de décision est modélisée par une structure de type enregistrement nommée **un nœud,** caractérisé par 7 champs comme suite :

- ✓ **nbr_indv :** nombre de connexion destinée.
- ✓ **Indfils_g :** indice de fils gauche.
- ✓ **Indfils_d :** indice de fils droit.
- ✓ **Att :** indice de la variable de segmentation.
- ✓ **Test :** seuil de segmentation.
- ✓ **Classe :** (0) pour normal, (1) pour anormale et(-1) si ce n'est pas une feuille
- ✓ **Tabcon :** contint toutes les connexions destinées.

```
public class BNode {

        public BNode leftBNode,  rightBNode; // the nodes
        public connection c; //the AnyClass objext

        public BNode(connection c ) {//constructor
                this.c= c;
                this.leftBNode = null;
               this.rightBNode = null;
                }

         public void show() {
                //calls the show method of the AnyClass
System.out.print(c.getX1() +"~~" + c.getX2() + "~~" + c.getX3()+"\n");
                }
                        }
```

Figure V.4 code de la classe nœud

V.7.définition de la classe arbre :

Afin de bien modéliser et facilité la manipulation de l'arbre de décision nous avons opté pour l'utilisation d'une arraylist.
Chaque enregistrement fait référence à un nœud de l'arbre qui est une instance de la classe **nœud**.

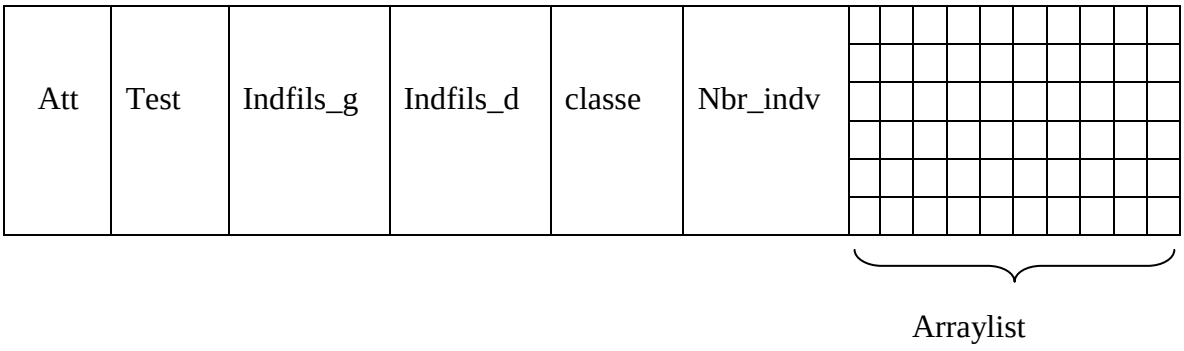
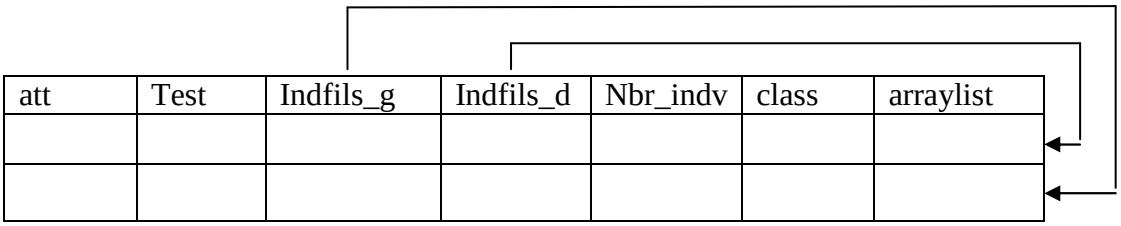


Figure V.5 : structure d'un enregistrement nœud

Grâce aux deux champs (indfils_g, indfils_d) chaque nœud pourra pointer vers ses deux fils qui sont eux aussi des nœuds dans l'arbre.



att	Test	Indfils_g	Indfils_d	Nbr_indv	class	arraylist

Figure V.6 : structure NŒUD

Pour la construction de l'arbre de décision basé sur l'algorithme **CART** dans cette classe nous avons utilisé plusieurs méthodes **gini** et **app teste.**

V.7.1 définition de la méthode gini :

L'algorithme CART est une méthode qui teste toutes les variables potentielles et choisit celle qui maximise un critère donné. Il faut donc que le critère utilisé caractérise la pureté (ou le gain en pureté) lors du passage du sommet vers les feuilles produites par la segmentation.

Dans notre cas, nous avons utilisé le **coefficient de gini** et ses variantes. La méthode **Gini** prend en paramètre l'indice de l'enregistrement-nœud **p** dans la table d'enregistrement **nœud[]** (qui indique aussi sa position dans l'arbre) et retourne le coefficient de gini en appliquant la formule de gini suivante :

$$\mathbf{GINI}(\boldsymbol{p}) \quad = \mathbf{1} - \sum_{\boldsymbol{k}=\mathbf{1}}^{\boldsymbol{c}} \mathbf{p}\left(\frac{\boldsymbol{k}}{\boldsymbol{p}}\right)\mathbf{2}$$

C : nombre des classe (deux classe dans notre cas)

P(k/p) : proportion des individus (connexion) appartenant à la classe k parmi ceux de la position p.

V.7.2. Définition de la méthode app test :

Dans la phase d'apprentissage **app test** est la méthode principale qui fait la construction de l'arbre. Elle reçoit comme paramètre un nœud (son indice dans la table des nœuds plus précisément) elle se base sur l'algorithme qui est illustré dans la figure. À fin d'effectué les taches suivantes

- ✓ trouver l'attribut (variable de segmentation) qui conviendra pour libeller ce nœud tout en faisant appel à la méthode gini.
- ✓ Calculé le seuil ou bien le point de coupure pour cet attribut qui va maximiser le gain.
- ✓ Calculer l'indice du fils gauche qui est égal à deux fois l'indice de père.
- ✓ Calculer l'indice du fils droit qui égal à deux fois l'indice de père plus un(1).
- ✓ Affecter pour chacun des deux fils les connexions qui lui sont destinées tel que :

Les connexions qui répondent oui pour le teste libellant le nœud père vont dans la table des connexions (tabCon) du fils gauche.

Les connexions qui répondent non pour le teste libellant le nœud père vont dans la table des connexions (tabCON du fils droit.

- ✓ Calculer pour chacun des deux fils le nombre de connexions qui lui sont destinées.

Si (le nœud est terminal)
Alors lui affecter une classe ;
Sinon
Répéter
Pour chaque variable de segmentation ;
Trier les variables de la variable ;
Trouver les point de coupure ;
Evaluer le critère de segmentation (indice de gini) pour
Chaque point de coupure ;
Choix du point de coupure qui maximise l'indice de gini ;
Calcule le gain pour se point ;
Créer les fils gauche et droite ;
Jusqu'à la fin des variables.
Finsi.

Algorithme de la méthode app test

V.7.3.définition de la méthode main :

Elle remplit les taches suivantes :

- ✓ déclaration des variables globales.

- ✓ Faire appelle à la classe **lecture** pour construire la table **tabcon** qui contient toute les connexions de la base KDD.
- ✓ Création des nœuds racine (de type nœud) et remplissage de ses deux champs :
  - o Nbr_indv : nombre totale des connexions de la table KDD
  - o Tabcon : la table créée par lecture de fichier.
- ✓ Répéter l'appelle à la méthode apptest jusqu'à ce que la construction de l'arbre soit faite.
- ✓ Affiché quelque message au fur et a mesure de son exécution

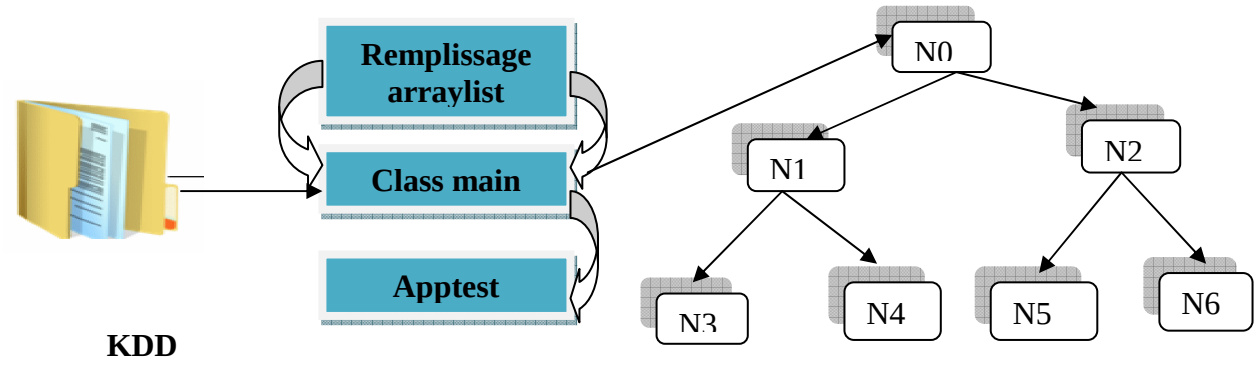


Figure V.7 modélisation du comportement normal

2. Phase de test :

Après la phase d'apprentissage, on passe à la phase de détection des attaques (phase de test). Dans cette phase, nous utilisons une base de test KDD contenant des connexions normales et des connexions anormales (ou attaques). Chaque connexion de cette base est considérée par 41 attributs. Lorsqu'**IDSCART** reçoit une connexion c_i de cette base, il parcoure l'arbre de décision de la racine vers les feuilles.

A fin de passer d'un nœud à l'autre de ses fils, notre IDS compare l'attribut $\mathbf{att_i}$ (l'attribut de la connexion c_i qui correspond à l'attribut **att** du nœud) au seuil **test** de ce nœud (est-ce que $\mathbf{att_i} >$ **test ??)**. Si cette connexion répond par oui alors on passe vers le fils gauche de ce nœud sinon on passe vers le fils droit, jusqu'à l'atteinte d'une des feuilles qui est caractérisé par une classe (normale ou anormale), en fin si la classe de cette dernière est normale alors la connexion c_i est considéré comme étant un comportement normale du système, sinon il nous signale une attaque.

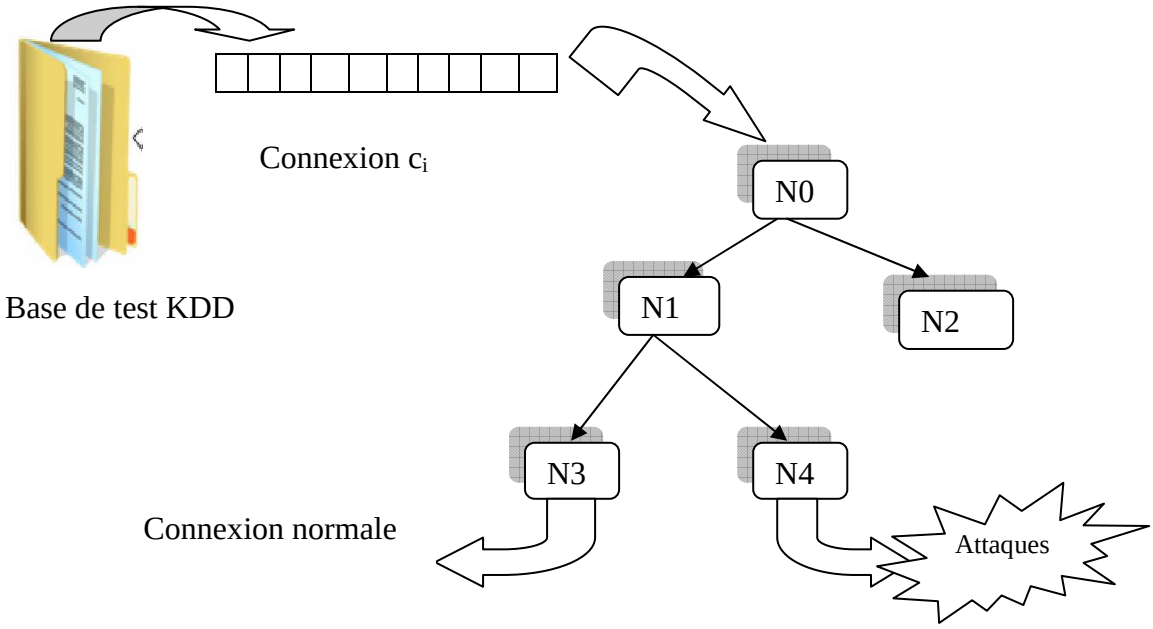


Figure V.8 : détection des attaques par **CARTIDS**

V.8 conclusion :

Dans se chapitre nous avons présenté une description formelle de notre système de détection d'intrusion baptisé **IDSCART** qui est basé sur un arbre de décision en utilisant l'algorithme CART. Cet IDS opère en deux phases.

La première phase est la phase d'apprentissage ou nous avons détaillé la manière dont le profil normal du système à surveiller est généré en appliquant l'algorithme CART sur la base d'apprentissage KDD. Dans la seconde phase, qui est la phase de test, nous avons expliqué comment IDSCART détecte les anomalies en utilisant le profil normal construit dans la première phase.

Conclusion générale

A la fin de ce travail, nous pouvons dire que nous avons bien pu avoir une visibilité concrète sur un domaine bien spécifique qui est la sécurité informatique.

En plus, ce travail nous as été profitable en terme d'acquérir une bonne expérience professionnelle, à travers laquelle nous avons eut l'occasion d'appliquer nos connaissances scientifiques et de confronter la notion théorique à la pratique.

Et pour conclure, nous pouvons dire que l'objectif global n'est pas atteint par un seul projet, mais par une succession de projets afin d'établir un audit de sécurité selon une méthode et norme standard.

Perspectives

Malheureusement, les objectifs qu'on s'est fixé ne sont pas atteins, et notre travail reste incomplets, c'est pour cela que nous prévoyions de continuer notre travail jusqu'à aboutir aux objectifs suivants :

- ✓ Finalisation de la mise en œuvre et de l'implémentation d'un IDSCART.
- ✓ Réalisation d'un prototype et mise en pratique sur un réseau d'entreprise.

Bibliographie

[1] : www.vulgarisation-informatique.com › Cours › Cours réseaux
[2] : les réseaux informatiques d'entreprise Pierre Evry, 1998
[3] : http:// www.commentcamarche.net
[4] :http://netalya.com/fr/reseaux1.asp
[5] :http:// www.protocols.com/pbook/h323.htm
[6] :www.protocols.com/pbook/h323.htm
[7] : la sécurité des réseaux Guillaume des george 2000 http://www.guill.net
[8] : le grand Livre de Securiteinfo.com-http://www.securiteinfo.com
[9] : sécurité des réseaux informatiques Bernard Cousn Université de Rennes1
[10] :http//fr.Wikipedia.org/wiki/wikipédia :Accueil_principal
[11] : les systèmes de détection d'intrusion (Claude Duvallet, Université de Havre UFR Sciences et Technique).
[12] : la détection d'intrusion(Optimisation par classification).. RAHMANI amine et BOUMDIEN Hassan
[13] : les systèmes de détection d'intrusion basés sur la machine learning (Liran LERMAN)
[14] : M.moradi & M.zulkernine (2004), (a neural Network Based System for Intrusion detection and Classification of Attacks) , university of British columbia, Canada.
[15] : Jacob Zimmermann ludovic Mé, Christophe bidan. Introducing refernce flow control for detecting d'intrusion- david Bugermeister, Jonatham Krier-22/07/2006-http://dbprog.developpez.com
[17] : Architecture exprémentale pour la détection d'intrusion dans un système informatique….Phiippe biondi.
[18] : technique d'apprentissage thème arbre de décision partie IFT 603
[19] : http://fr.wikiversity.org/wiki/Arbre_de _décision.
[20] : une approche probabiliste pour le classement d'objets incomplément connus dans un arbre de décision THESE présentée pa Lmis HAWARAH Université Joseph Fourier.
[21] : Réseaux bayésien naifs et arbre de décision dans les systèmes de détection d'intrusion par : N.BenAmor, S.Benferhat et Z.Elouedi
[22] :Arbre de Décision par : Rico RAKOTOMALALA Laboratoire ERIC Université lumière Lyon 2
[23] : la détection d'intrusion [optimisation par classification] par :Rahmani amine et boumedien hacen.
[24] :Yacine Bouzida, Frédérie Cuppens, Sylvain Gombault « Détection de nouvelles attaques »
[25] :http://kdd.ccs.uci.edu/databases/kddcup99/task.html/
[26] :http://www.securitéinfo.com
[27] : cours de sécurité informatique par : pierre-françois Bnnefoi
[28] : Un petit quide pour la sécurité par : Alexandre Viardin
[29] : Arbre de décision Cours d'analyse de données Université Paris I
[30] : Introduction et initiation à la sécurité informatique « sécuritéInfo.com »
[31] :G.Zémor(2000) « cours de cryptographie
[33] : http://www.tripwire.com/products/index.cfml
[34] : http://freachmeat.net/redir/swatch/10125/url_homepage/swatch

Bibliographie

[35] : http://www.enterasys.com/ids/squire/
[36] : http://freachmeat.net/redir/tiger-audit/30581/url_homepage/tiger
[37] : http://www.netiq.com/products/sm/default.asp
[38] : http://www.snort.org
[39] : http://www.marlboro.edu/ttoomey/benids
[40] : http://hank.sourcrforge.net/
[41] : http://www.prelude-ids.org
[42] : http://www.scaramangna.co.uk/restorm/
[43] : Conception et réalisation d'un système de détection d'intrusion par : Mlle Dalila boughaci
[44] : http://www.bath.ac.uk/bucs/networking/connectfromehome/virtualprivatenetworkvpn
[45] : http://www.labo-microsofte.org/
[46] : http://www.technoplus .Fr
[47] : http://www.ebook-cours.com/reseaux-communication-cour-gratuit.html
[48] : http://www.labo-microsofte.org/
[49] : F.3.http://igm.univ-mlv.fr
[50] :http://www.insecure.in/ids.asp
[51] : protection des systèmes informatique contre les attaques par entrées-sorties par :
Fernand Lone Sang.