République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique
Université Mouloud Mammeri de Tizi Ouzou
Faculté Génie Electrique et Informatique
Département d'Informatique

MÉMOIRE

En Vue de l'Obtention de Diplôme de Master En informatique

Option : Systèmes Informatique

THEME

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

Présenté Par :

M[r] Smail ZAHER

M[r] Mokrane Yahiatene

Proposé Par :

M[me] Haddaoui

Promotion 2013/2014.

Remerciements

Nous adressons nos remerciements les plus sincères aux personnes qui nous ont aidées dans la réalisation de ce mémoire.

En premier lieu, nous remercions Mme. Haddaoui En tant que promotrice, elle nous a guidé dans notre travail et nous a aidée à trouver des solutions pour avancer.

Nous remercions aussi tous les amis et toutes les personnes qui nous ont aidés de prés ou de loin.

Dédicaces

Ce travail, et bien au-delà, je le dois à mes très chers parents et grands parents qui m'ont fourni au quotidien un soutien et une confiance sans faille et de ce fait, je ne saurais exprimer ma gratitude seulement par des mots. Alors j'adresse au ciel les vœux les plus ardents pour la conservation de leur santé et de leur vie.

A ma précieuse sœur, les mots ne peuvent résumer ma reconnaissance et mon amour à ton égard, puisse-tu vivre dans le bonheur et la paix.

A mes deux chers frères que j'aime tant.

A mes très chères cousins et cousines tous autant qu'il sont.

A mes adorables amies pour leur fidélité et le courage qu'il me procurent chaque jour, en particulier BABA Ali et tarwa-s

A mon ami, mon frère et mon associé ainsi que toute sa famille petite et grande sans oublié l'adorable petit FARHAT.

A mon cher ami, mon binôme, qui m'a supporter tout au long de ce travail.

Que toute personne m'ayant aidé de près ou de loin, trouve ici l'expression de ma reconnaissance.

Smail ZAHER

Dédicaces

Ce travail, et bien au-delà, je le dois à mes très chers parents et grands parents qui m'ont fourni au quotidien un soutien et une confiance sans faille et de ce fait, je ne saurais exprimer ma gratitude seulement par des mots. Alors j'adresse au ciel les vœux les plus ardents pour la conservation de leur santé et de leur vie.

A mes deux chères sœurs, les mots ne peuvent résumer ma reconnaissance et mon amour à votre égard, puissiez-vous vivre dans le bonheur et la paix.

A mes deux chers frères que j'aime tant.

A mes adorables amies pour leur fidélité et le courage qu'il me procurent chaque jour, en particulier dada moh said.

A l'adorable enzo ,puisse-t-il bercer dans un bonheur infini.

A mon cher ami, mon binôme, qui m'a supporter tout au long de ce travail.

Que toute personne m'ayant aidé de près ou de loin, trouve ici l'expression de ma reconnaissance.

Mokrane YAHIATENE

Table des matières

Introduction générale *1*

Chapitre I : Réseaux et techniques de protection contre les attaques réseau

Partie I : Réseaux informatiques .....

I.1introduction *4*

I.2. Définition du réseau informatique *4*

I.3. Intérêt d'un réseau *5*

I.4. Classification des réseaux informatiques *5*

I.4.1. Selon leur étendue géographique *5*

a) Le réseau LAN (local area network) *6*

b) Le réseau MAN (Métropolitain area network) *6*

c) Le réseau WAN (Wide area network) *6*

I.4.2. Selon les fonctions assumées par les ordinateurs *6*

a) Réseau poste-à-poste *7*

b) Réseau à serveur dédicacé ou client serveur *7*

I.4.3. Selon la topologie réseau *8*

I.4.3.1. Topologie physique *8*

a)Réseaux en mode de diffusion *8*

b) Réseaux en mode point à point *9*

I.4.3.2. Topologie Logique *9*

I.5. Fonctionnement d'un réseau *10*

I.5.1. Le Modèle OSI (Open System Interconnexion) *10*

I.5.2. L'architecture TCP/IP (Transmission Control Protocol / Internet Protocol) *12*

I.5.3 comparaison entre TCP/IP et OSI *13*

Similitudes *13*

Différence *14*

Partie II : sécurité informatique *15*

II.1 Introduction *15*

II.2. Objectifs des attaquants *15*

II.3. les étapes d'une attaque 16
a) Identification de la cible 16
b) Le scanning 16
c) L'exploitation 16
d) La progression 16
e) Préservation d'accès : 16
f) Effacement des traces 16
II.4. Les types d'attaques 16
Les attaques directes 17
Les attaques indirectes par rebond 17
Les attaques indirectes par réponse 17
II.5. Les risques et les menaces 19
Les programmes malveillants 19
Les risques et menaces de la messagerie électronique 20
Les risques et menaces sur le réseau 20
II.6. Objectifs de la sécurité informatique 21
II.7. Les mécanismes de défenses 21
a)Le chiffrement 21
b) La signature 21
c) La notarisation 22
d) Le contrôle d'accès 22
e) Les Antivirus 22
f) Le pare-feu 22
g) Les VPN (Virtual Private Network) 23
h) Proxy 23
i) La journalisation ("logs") 24
j) Contrôle de routage 24
k) Authentification 24
l) La protection physique 24
m) Le DMZ (zone démilitarisé) 24

n) Les IDS 25
II.7 Caractéristique d'un système sécurisé 25
Intégrité des données 25
Confidentialité 26
Contrôle d'accès 26
Identification/Authentification 26
Non-répudiation 26
II.8. Conclusion 27

Chapitre II : Le système de détection d'intrusion

I.1. Introduction 29
I.2. Les systèmes de détection d'intrusion 29
I.3. Historique 30
I.4. Fichier historique 30
I.5. Composition d'un IDS 31
Le senseur 31
L'analyseur 31
L'interface utilisateur 31
I.6. Positionnement 31
I.7. Schéma d'une attaque 32
I.7.1. Les attaque réseau 32
I.7.2. Les attaques applicatives 33
I.8. Caractéristiques d'un système de détection d'intrusions 33
I.9. Classification des systèmes de détection d'intrusions 34
1.9.1. Classification selon la méthode de détection 36
- L'approche par scénario 36
a) Pattern matching 36
b) Les systèmes experts 37
- L'approche comportementale 37
a) Le modèle de DENNING 38
b) Réseau de neurones 38

- Comparaison entre les deux approches 39
- Conclusion sur les deux approches (comportementale, par scénario) 39
I.9.2. Classification selon le comportement après la détection 39
- Réponses passives (les IDS passifs) 39
- Réponses actives (les IDS actifs) 40
I.9.3. Classification selon la source de données a analysées 40
- Les HIDS 41
- Les NIDS 43
- Les IDS hydrides (NIDS+HIDS) 45
I.9.4. Classification selon la fréquence d'utilisation 46
- Les IDS online (continue) 46
- Les IDS offline (périodique) 46
I.10. L'architecture des IDS : 46
I.10.1. architecture centralisée 46
I.10.2. architecture partiellement distribuée 46
I.10.3. architecture totalement distribuée 47
I.11. Modèles et normalisations 47
I.11.1 CIDF 47
a) générateur d'événement : (E-BOX) 47
b) analyseur d'événement : (A-BOX) 48
c) stockage d'informations : (D-BOX) 48
d) contre mesure : (C-BOX) 48
I.11.2. IDMEF 48
I.12. Tests des IDS 48
I.13. Quelques IDS existant 49
a) HAYSTACK 49
b) MIDAS 50
c) IDES 50
I.14. Conclusion 51

Chapitre III : Arbre de décision & la bases d'apprentissage et de test KDD

Partie I : Les Arbres de décisions ***53***

I.1. Introduction *53*

I.2. définitions *54*

I.2.1. L'apprentissage automatique *54*

Types d'apprentissage *54*

Algorithmes utilisés *54*

I.2.2. Les arbres de décision *55*

I.3. petit historique *56*

I.4. Etapes principale d'utilisation des arbres de décision *56*

I.4.1. Etape de construction *57*

a) Choix de la variable de segmentation *57*

b) Traitement des variables continues *57*

c) Définir la bonne taille de l'arbre *58*

Le pré-élagage *59*

Le post-élagage *59*

d) Affectation de la conclusion de chaque feuille *60*

I.4.2. Etape de classification *60*

I.5. Algorithmes d'apprentissage par arbre de décision *60*

Algorithme d'apprentissage générique *61*

I.6. les méthodes d'apprentissage *61*

I.6.1. méthode ID3 ..... *61*

I.6.2. méthode C4.5 ..... *62*

I.6.3. CHAID *62*

I.6.4. méthode CART *62*

Phase d'expansion *63*

Partie II : la bases d'apprentissage et de test KDD ***65***

II.1. Introduction *65*

II.2. Qu'est ce que le KDD ? *66*

II.3. Description de la base KDD *66*

II.4. Les attaques de la base KDD'99 67
- *Déni de Service - « Denial-Of-Service (DOS) »* 67
- *Les attaques de type « Remote to Local access » (R2L)* 67
- *Les attaques de type « User to Root attacks » (U2R)* 67
- *Reconnaissance- Probing* 67

II.5. Les attribue caractérisant chaque connexion 69

II.6. Conclusion 71

CHAPITRE IV : Conception et test de l'IDS

IV.1.Introduction 73

IV.2. L'objectif du présent travail 73

IV.3. Structure CARTIDS 73

1. Phase d'apprentissage 75

1.1. Définition de la classe remplissageTable 75

1.2. Définition de la classe unNoeud 77

1.3. Définition de la classe arbre 77

1.3.1. Définition de la méthode gini 79

1.3.2. Définition de la méthode appTest 79

1.3.4. Définition de la méthode main 80

2. Phase test : 81

IV.4. Conclusion 83

Conclusion générale 85

Bibliographie 87

Table des figures

Figure I.1 : Réseau informatique 4
Figure I.2 : classification des réseaux selon leurs étendus 5
Figure I.3 : Schéma d'un réseau post-à-post 7
Figure I.4 : Schéma d'un réseau client/serveur 7
Figure I.5 : Topologie en bus 8
Figure I.6 : Topologie en anneau 8
Figure I.7 : Topologie en étoile 9
Figure I.8 : Topologie maillée: 9
Figure I.9 : Les couches du modèle OSI 11
Figure I.10 : La pile de protocole TCP/IP avec sa correspondance OSI 12
Figure I.11:Classification des attaques sur les systèmes informatiques 18
Figure I.12 : Schéma résumant le chiffrement 21
Figure I.13 : Fonctionnement d'un pare-feu 22
Figure I.14 : Fonctionnement du VPN 23
Figure I.15 : Fonctionnement d'un proxy 23
Figure I.16 : Fonctionnement de la DMZ 25
Figure I.17 : Fonctionnement d'un IDS 25
Figure II.1 : Modèle simplifier d'un IDS 29
Figure II.2 : Le positionnement des IDS 31
Figure II3 : Classification des IDS. 35
Figure II.4 : Approche par scénario 36
Figure II.5 : Approche comportementale 38
Figure II.6 : Emplacement d'un HIDS 42
Figure II.7 : Fonctionnement d'un NIDS 45
Figure II.8 : Fonctionnement d'un IDS hybride 45
Figure II.9 : Architecture CIDF 47
Figure III.1 : Représentation graphique d'un arbre de décision. 55
Figure III.2 : Evolution du taux d'erreur en apprentissage et en teste en fonction du nombre des feuilles. 58

Figure III.3 : Algorithme d'apprentissage général. *61*
Figure IV.1 : Structure CARTIDS. *74*
Figure IV.2 : Construction de la table tabCon. *76*
Figure IV.3 : Code de la classe unNoeud. *77*
Figure IV.4 : Structure d'un enregistrement nœud. *78*
Figure IV.5 : Aperçu de tabCon. *78*
Figure IV.6 : Algorithme de la méthode appTest. *80*
Figure IV.7 : Modélisation du comportement normal. *81*
Figure VI.8 : Détection des attaques par CARTIDS. *82*

Sommaire des tableaux

Tableau II.1 : Comparaison entre l'approche comportementale et l'approche par scénario .. *39*
Tableau II.2 : Réponse aux attaques par l'IDS. *40*
Tableau III.1 : Petit historique sur les algorithmes de classification. *56*
Tableau III.2 : Types d'attaques *68*
Tableau III.3 : Liste des attributs. *70*
Tableau IV.1 : Table de correspondance des attributs qualitatifs. *76*
Tableau III.3 : Liste des attributs. *70*

Introduction générale

Les réseaux et systèmes informatiques sont désormais omniprésent dans la vie des hommes au point de devenir vitales et déterministes pour le bon fonctionnement des entreprises. Ils sont de nos jour déployés dans tous les secteurs professionnels: Les banques, Les assurances, La médecine, ou encore le domaine militaire.

Initialement isolés les un des autres, les réseaux et systèmes sont à présent interconnectés et le nombre de points d'accès ne cessent de croitre. Ce magnifique élan de l'informatique moderne qui facilite la communication et le transfert de données entre des point très éloignés, a malheureusement données naissance à un nouveau type d'utilisateurs appelés: **Les pirates informatique.** Ces derniers exploitent les failles et les vulnérabilités des réseaux afin de mettre la main sur des informations sensibles, et de les utilisées pour lire leurs contenue, les modifier, ou les détruire pour perturber le bon fonctionnement du système, ou encore tout simplement pour le gout du jeu et du défit; comme le font certain Hackers.

Les attaques contre les réseaux informatiques et leurs ressources sont en augmentation constante et deviennent de plus en plus sophistiquées. Cette affirmation est confirmée par les rapports annuels du **C**omputer **E**mergency **R**esponse **T**eam (**CERT**) qui mentionnent aussi l'insuffisance des mesures destinées à contrer ces attaques et mettent en évidence la nécessité de toujours améliorer la protection des systèmes d'information.

Les pare-feux et les systèmes de détection d'intrusion sont deux techniques complémentaires destinées à accroitre la sécurité d'un système d'information. Ainsi, ils en viennent à en constituer des composantes critiques. Un pare-feu agit comme une première barrière externe pour repousser les pirates informatiques, tandis qu'un système de détection d'intrusion vise à repérer ceux qui auraient transpercé ce premier périmètre de défense. Un système de détection d'intrusion surveille donc en permanence l'activité à l'intérieur d'un réseau et émet une alarme en cas d'activité anormale.

Les IDS sont devenus très largement déployés dans les systèmes informatiques et ils ont gagné une place importante dans la conception de la stratégie de sécurité.

Introduction générale

Problématique:

Malgré l'essor que connait le domaine de la sécurité informatique grâce notamment aux entreprises qui n'hésitent pas à investir dans ce secteur, on ne peut pas dire qu'il existe un outil qui puisse garantir une parfaite sécurisation des systèmes informatique. Plusieurs études ont été menées sur les système de détection d'intrusion et plusieurs travaux ont été réalisés, mais les IDS restent limités ce qui vulnérabilise les réseaux.

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 (C**lassification **A**nd **R**egression **T**rees **).**

Organisation:

Ce mémoire est organisé comme suit:

- Nous avons commencé 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 et sur la sécurité informatique, il est composé de deux partie:

- **Partie 1:** Cette partie représente une introduction au réseaux informatique: définition, intérêt, classification, fonctionnement, etc.
- **Partie 2:** Dans cette partie, on explique la notion de sécurité informatique on parlant des objectif et des étape d'une attaque, ainsi qu'on présentant les objectifs de la sécurité et les mécanismes de défense.

Chapitre II: Ce chapitre décrit les systèmes de détection d'intrusions (IDS), notamment leurs compositions et leurs positionnement, et propose les caractéristiques d'un IDS ainsi qu'une classification selon plusieurs critères. Nous terminons le chapitre en évoquant quelques IDS actuels.

Chapitre III: il se compose de deux partie :

- **Partie 1:** Cette partie traite de l'apprentissage automatique et des algorithmes utilisés. elle présente aussi les arbre de décision et les étapes principales de leurs utilisation, et on termine cette partie en évoquant quelque méthodes d'apprentissage.
- **Partie 2:** Cette partie est consacrée à la présentation de la base KDD qu'on se propose d'utiliser dans notre simulation d'IDS.

Chapitre IV: Dans ce chapitre , nous avons procéder à la présentation de notre IDS nommé ***CARTIDS*** en détaillant sa structure, et en expliquant notamment la phase d'apprentissage ainsi que la phase de test qui sont 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.

UNIVERSITE MOULOUD MAMMERI DE TIZI OUZOU

CHAPITRE I : Réseaux et techniques de protection contre les attaques réseaux.

***Partie I** : Réseaux informatique.*

***Partie II** : Sécurité informatique.*

Partie I: Réseau informatique

I.1.Introduction

Avec le développement de l'informatique, et pour le besoin d'échange de données entre des machines (ordinateur, machines terminales, serveur, imprimante …) distantes les informaticien ont eu le besoin de créer un moyen qui permet de partager des ressources entre ces machines qui est bien évidement le réseau informatique.

Dans le cadre de la première partie de ce chapitre, nous allons poser la définition, les classifications et finalement le fonctionnement des réseaux informatiques.

I.2. Définition du réseau informatique

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

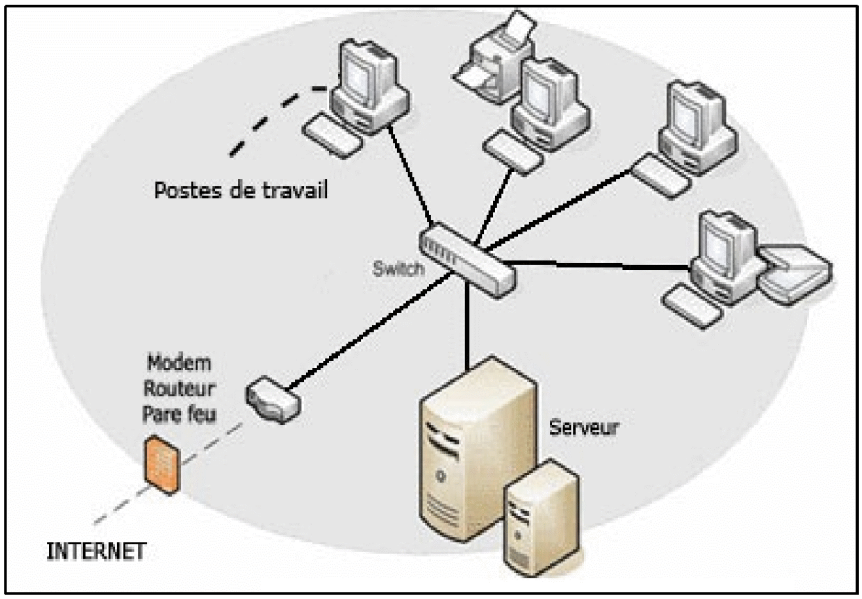

Figure I.1 : réseau informatique [46]

I.3. Intérêt d'un réseau [1]

La nécessité de communication et du partage des informations en temps réel impose aujourd'hui aux entreprises la mise en réseau de leurs équipements informatiques en vue d'améliorer leurs rendements. Un réseau permet :

- La communication entre personnes (grâce au courrier électronique, la discussion en direct,...).
- La communication entre processus (entre des machines industrielles).
- La garantie de l'unicité de l'information (bases de données).
- Le partage de fichiers d'applications.

I.4. Classification des réseaux informatiques [2] [3]

La classification se fait par rapport à un critère donné, ainsi nous pouvons classifier les réseaux informatiques :

a) Selon leur étendue géographique ;
b) Selon les fonctions assumées 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, LAN, MAN et WAN.

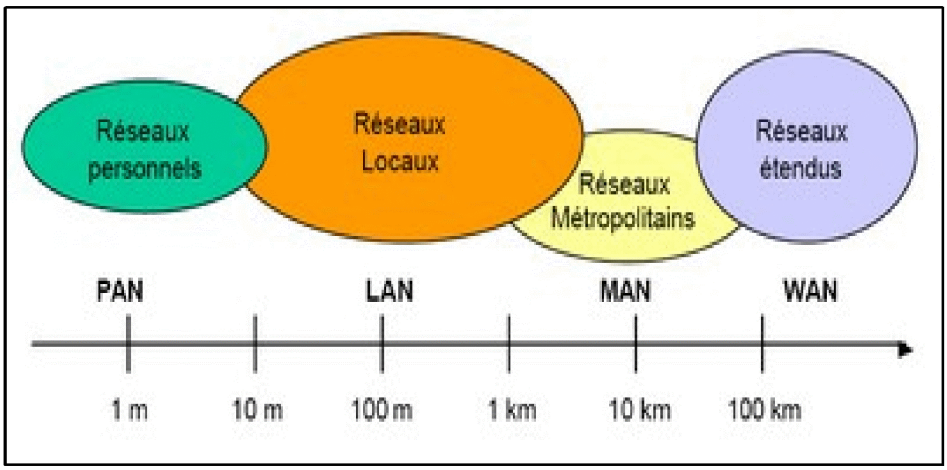

Figure I.2 : Classification des réseaux selon leur étendus [47]

a) Le réseau LAN (local area network)

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

Ils permettent aussi aux entreprises de partager localement des fichiers et des imprimantes de manière efficace et rendent possibles les communications internes.

b) Le réseau MAN (Métropolitain area network)

Tout réseau métropolitain est essentiellement un LAN, du point de vue de la technologie utilisée. Il peut couvrir un grand campus ou une ville.

c) Le réseau WAN (Wide area 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 à longue distance qui couvre une zone géographique importante (un pays, voir même un continent).

I.4.2. Selon les fonctions assumées par les ordinateurs

Du point de vue architecture réseau, nous avons deux grandes catégories de réseaux :

- Réseau POSTE-à-POSTE (Peer to Peer) ;
- Réseau serveur dédicacé ou client-serveur (server based).

a) Réseau poste-à-poste

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

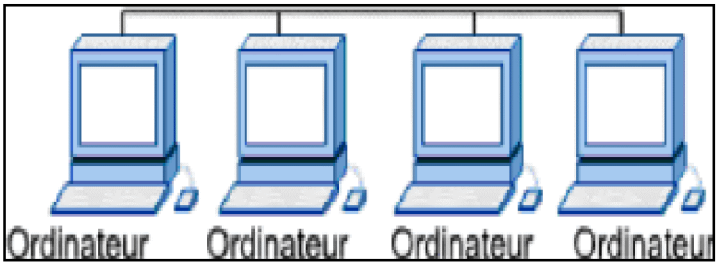

Figure I.3 : Schéma d'un réseau poste à poste

b) Réseau à serveur dédicacé ou client serveur

Dans une configuration client-serveur, les services de réseau sont placés sur un ordinateur dédié, appelé serveur, qui répond aux requêtes des clients. Un serveur est un ordinateur central, disponible en permanence pour répondre aux requêtes émises par les clients et relatives à des services de fichiers, d'impression, d'applications ou autres. La figure I.2 illustré ce réseau

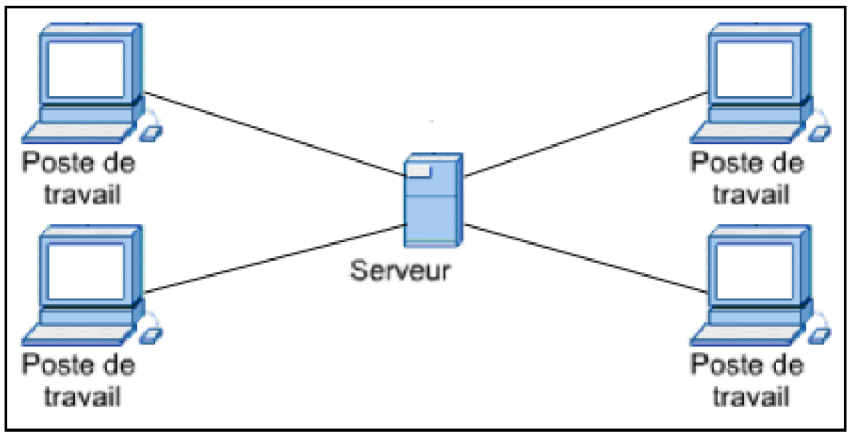

Figure I.4 : Schéma d'un réseau client serveur

I.4.3. Selon la topologie réseau

La topologie de réseau définit la structure du réseau. Elle représente l'interconnexion des équipements sur le réseau. Ces équipements sont appelés des nœuds. Les nœuds peuvent être des ordinateurs, des imprimantes, des routeurs, des ponts ou tout autre composant connecté au réseau. Un réseau est composé de deux topologies: physique et logique.

I.4.3.1. Topologie physique

La topologie physique du réseau se rapporte à la disposition des équipements et des supports. Ainsi, nous pouvons les classés en deux modes :

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

- La topologie en bus

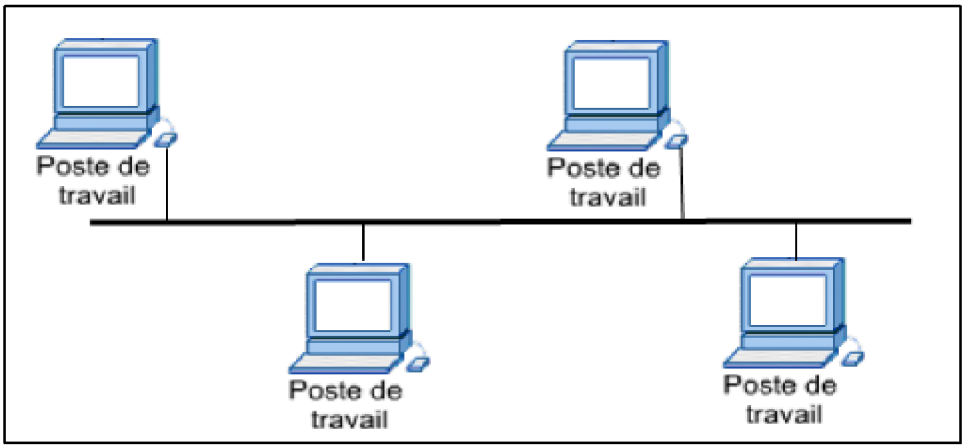

Figure I.5 : Topologie en bus

- La topologie en anneau

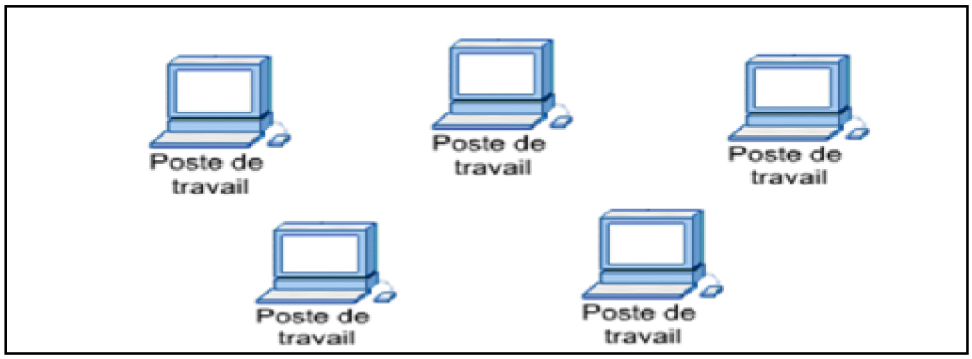

Figure I.6 : Topologie en anneau

b) **Réseaux en mode point à point :** par contre dans celui-ci chaque support physique relie une paire d'unités seulement. On trouve

- La topologie en étoile

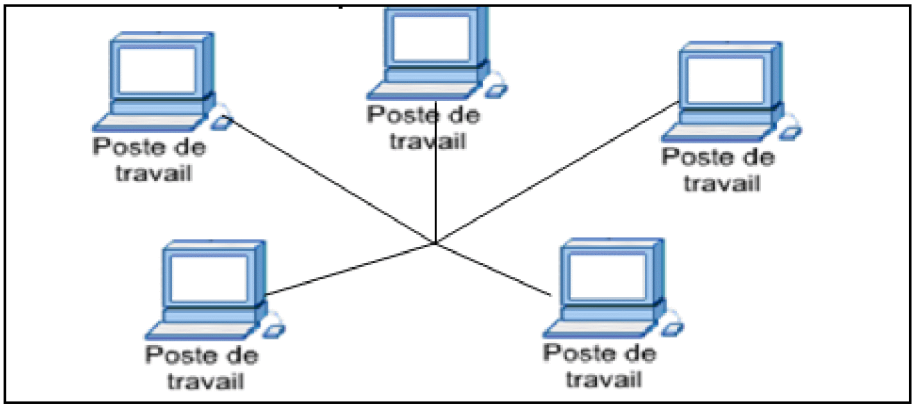

Figure I.7 : Topologie en étoile

- La topologie maillée

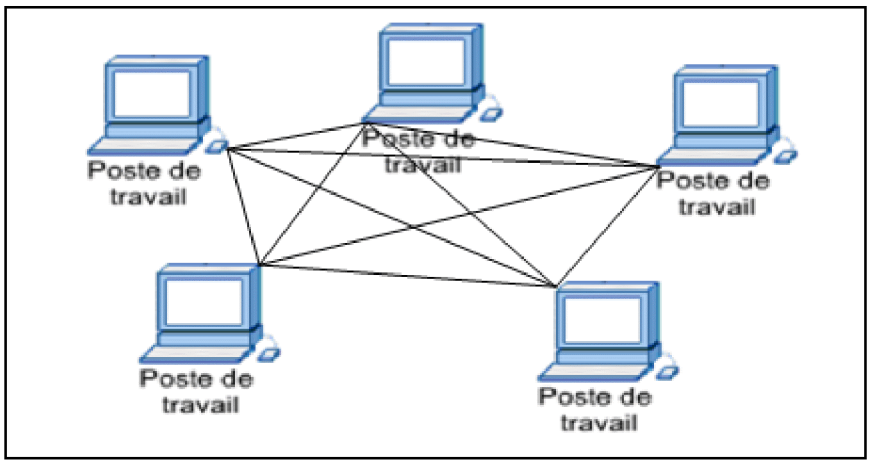

Figure I.8 : Topologie en maillée

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 aux supports et de transmission des paquets de données).

I.5. Fonctionnement d'un réseau

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

Ainsi, quatre principaux organismes internationaux travaillant de concert sont apparus: **[4]**

- ISO: International Organization for Standardization.
- CEI : Commission Electrotechnique Internationale.
- ITU : International Télécommunication Union, (anciennement CCITT).
- IEEE: Institute of Electronic and Electricity Engineers (prononcé I3E).

Pour mieux décrire la complexité des communications réseau, deux représentations des systèmes informatiques ont vu le jour : **[5] [6]**

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 protocoles 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 établie par l'International Standard organization (ISO) est la norme Open System Interconnections (OSI, interconnexion de Systems ouverts), elle est constituer des sept couches suivantes :

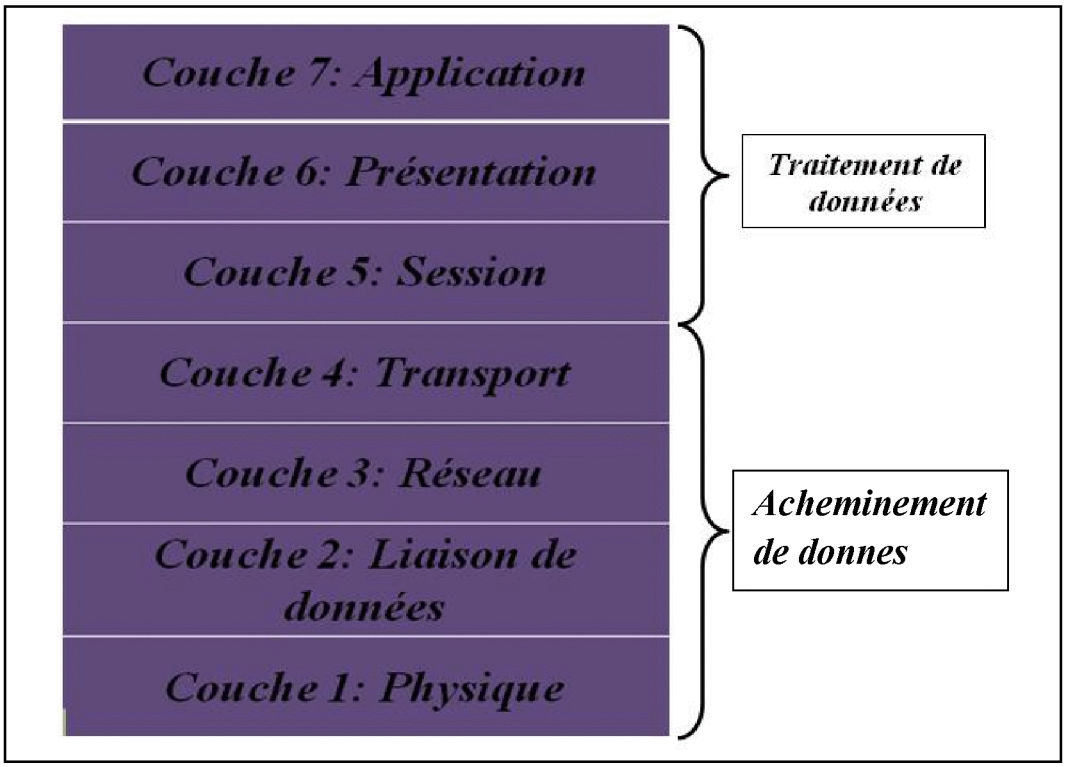

Figure I.9 : Les couches du modèle OSI

- **La couche physique :** Touche au support physique. Elle s'assure de la synchronisation, de l'encodage/décodage et de la transmission/réception de bits.
- **La couche liaison de données:** Gère des communications entre deux machines connectées au travers d'un support physique commun. Ces deux machines communiquent par l'intermédiaire de trames.
- **La couche réseau :** Concerne des interconnexions de machines distantes. Elle assure une transmission de bout en bout. Les machines communiquent en s'échangeant des paquets.
- **La couche transport :** Gère des communications de bout en bout entre processus. Les données échangées sont des messages.
- **La couche session:** Gère des modèles de communications transactionnels, appelés sessions.

- **La couche présentation:** Se charge du codage des données applicatives pour les présenter à l'application de manière transparente, indépendamment des codages des différents réseaux ou des différentes machines.

- **La couche application :** Responsable de l'interaction directe avec les utilisateurs. Elle traite les protocoles de connexion à distance, le courrier électronique ainsi que la définition des terminaux virtuels.

I.5.2. L'architecture TCP/IP (Transmission Control Protocol / Internet Protocol) [4]

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

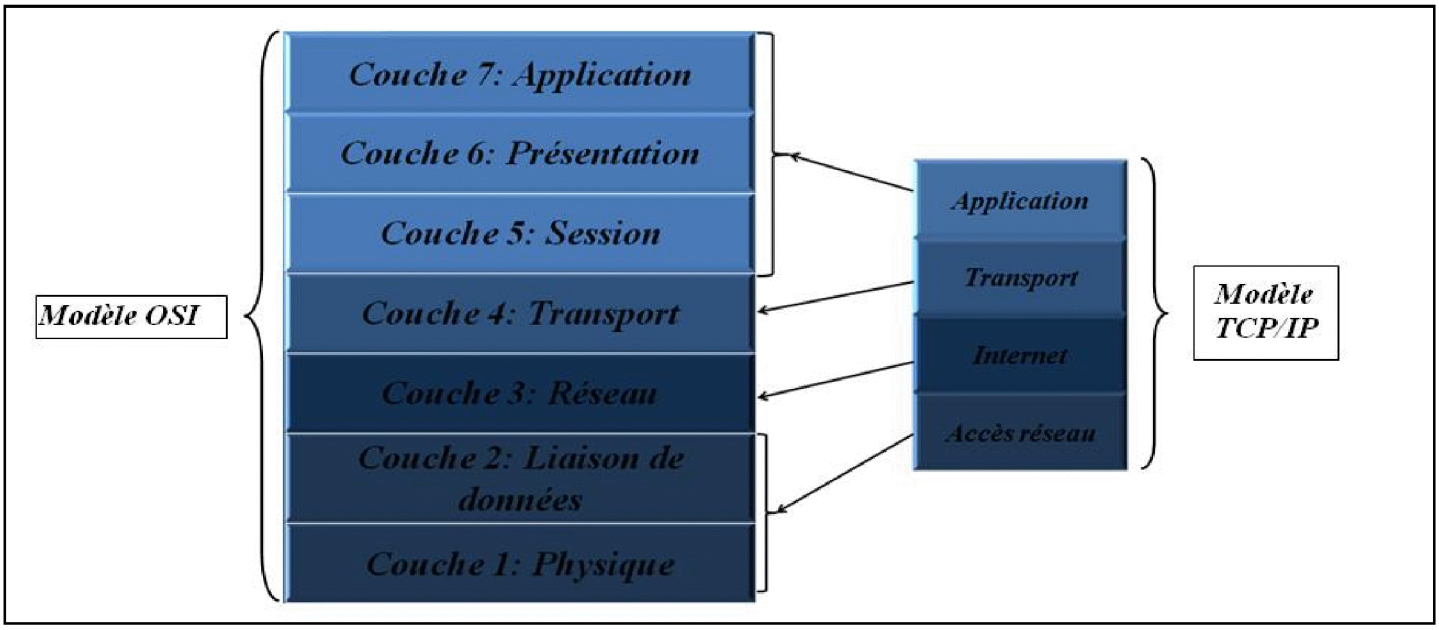

Figure I.10: La pile de protocoles TCP /IP avec sa correspondance OSI

a) **La couche d'accès réseau :** fait circuler les informations sur le réseau sous-jacent.

b) **La couche Internet :** fournit les mécanismes nécessaires à la gestion et à l'acheminement de paquets de données (protocole IP). Elle contient Cinque protocoles :

- **IP :** il gère les destinations des messages.

- **ARP (Adresse Resolution Protocol) :** protocole de résolution d'adresse, il permet de transformer une adresse logique (IP) à une adresse physique (MAC).

- **RARP (Reverse Adresse Resolution Protocol) :** protocole de résolution d'adresse reverse, qui assure le travail inverse de l'ARP, il convertit l'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 un protocole de la couche transport, au-dessus du protocole IP, il offre de nombreux services : l'ordonnancement des données, orientation de la connexion, ainsi que le contrôle de données.

- **UDP (User Datagram Protocol):** offre aux applications différents points d'accès. Deux applications différentes ne peuvent pas partager un même point 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.5.3 comparaison entre TCP/IP et OSI

❖ Similitudes:

- tous deux comportent des couches.
- tous deux comportent une couche application, bien que chacune fournisse des services très différents.
- tout deux comportent des couches réseau et transport comparables.
- tout deux supposent l'utilisation de la technologie de communication de paquets (et non de communication circuits).
- les professionnels des réseaux doivent connaître les deux modèles.

❖ Différences:

- TCP/IP intègre la couche présentation et la couche session dans sa couche application.

- TCP/IP regroupe les couches physiques et liaison de données OSI au sein d'une seule couche

- TCP/IP semble plus simple, car il comporte moins de couches

- Les protocoles TCP/IP constituent la norme sur laquelle s'est développé Internet. Aussi, le modèle TCP/IP a bâti sa réputation sur ses protocoles. En revanche, les réseaux ne sont généralement pas architecturés autour du protocole OSI, bien que le modèle OSI puisse être utilisé comme guide.

Partie II: Sécurité informatique

II.1 Introduction

Depuis la création des réseaux informatiques l'humain cherche à modifié, à développé et à inventé des nouvelles technologies à fin de mieux échanger des informations.

De nos jours l'information et indispensable et elle a une majeur importance dans l'entreprise, ce qui multiplie les attractions des Hackers (pirates informatiques).Sur ce fait la protection des réseaux et de ses données est une fonction primordiale.

Dans la deuxième et dernière partie de ce chapitre nous allons procéder comme suit, En premier lieu, nous présentons les objectifs des attaquants et les types d'attaques utilisées. Dans un second temps, nous allons voir les objectifs de la sécurité et les différents mécanismes de défenses. Enfin nous terminerons par les caractéristiques d'un système sécurisé et une conclusion.

II.2. Objectifs des attaquants [7]

Les attaquants ont pour objectif ce qui suit :

- Le déni de service (blocage d'un système)
- L'altération (modification ou destruction du système)
- Le renseignement (obtention d'informations non publiques)
- Utilisation des ressources (utilisation du système d'autrui pour faire quelque chose)

II.3. les étapes d'une attaque :

Généralement toute attaque suit le même schéma :

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 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 terminé l'attaquant essai d'effacer ses traces toute en restituant les propriétés des fichiers.

II.4. Classification des attaques sur les systèmes informatiques:

Tout acte sur un système dont l'intention est de nuire au moins à l'une des propriétés de sécurité est qualifié de malveillant et constitue, de ce fait, une attaque sur ce système. Ils existent des manières différentes de classer les attaques. Certaines taxonomies les organisent en fonction d'un unique critère. Parmi ces critères, les plus récurrents sont :

- la cause de l'attaque (utilisateur interne ou externe, intrus, etc.).
- le mode ou le type de l'attaque (virus, ver, écoute passive, déguisement, etc.).
- le résultat de l'attaque (divulgation, perturbation, etc.).
- la vulnérabilité exploitée par l'attaque ; plusieurs aspects peuvent alors être considérés.
- la manière d'attaque.

Nous allons détailler dans ce qui suit leur classification selon leurs manières d'attaques. De ce fait ces attaques peuvent être regroupées en trois familles différentes : **[8]**

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 de ces logiciels envoient directement les paquets à la victime.

Si vous vous faites attaqués de la sorte, il y a de grandes chances pour que vous puissiez remonter à l'origine de l'attaque, identifiant 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 a deux avantages :

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

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

L'attaque FTP Bounce fait partie de cette famille d'attaque. Si vous êtes victime de ce genre d'attaque, il n'est pas facile de remonter à la source. Au plus simple, vous remontrez à l'ordinateur intermédiaire.

Les attaques indirectes par réponse :

Cette attaque est un dérivé de l'attaque par rebond. Elle offre les mêmes avantages, du point de vue du 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 envoyée à l'ordinateur victime.

Dans ce cas de figure aussi, il n'est pas aisé de remonter à la source...

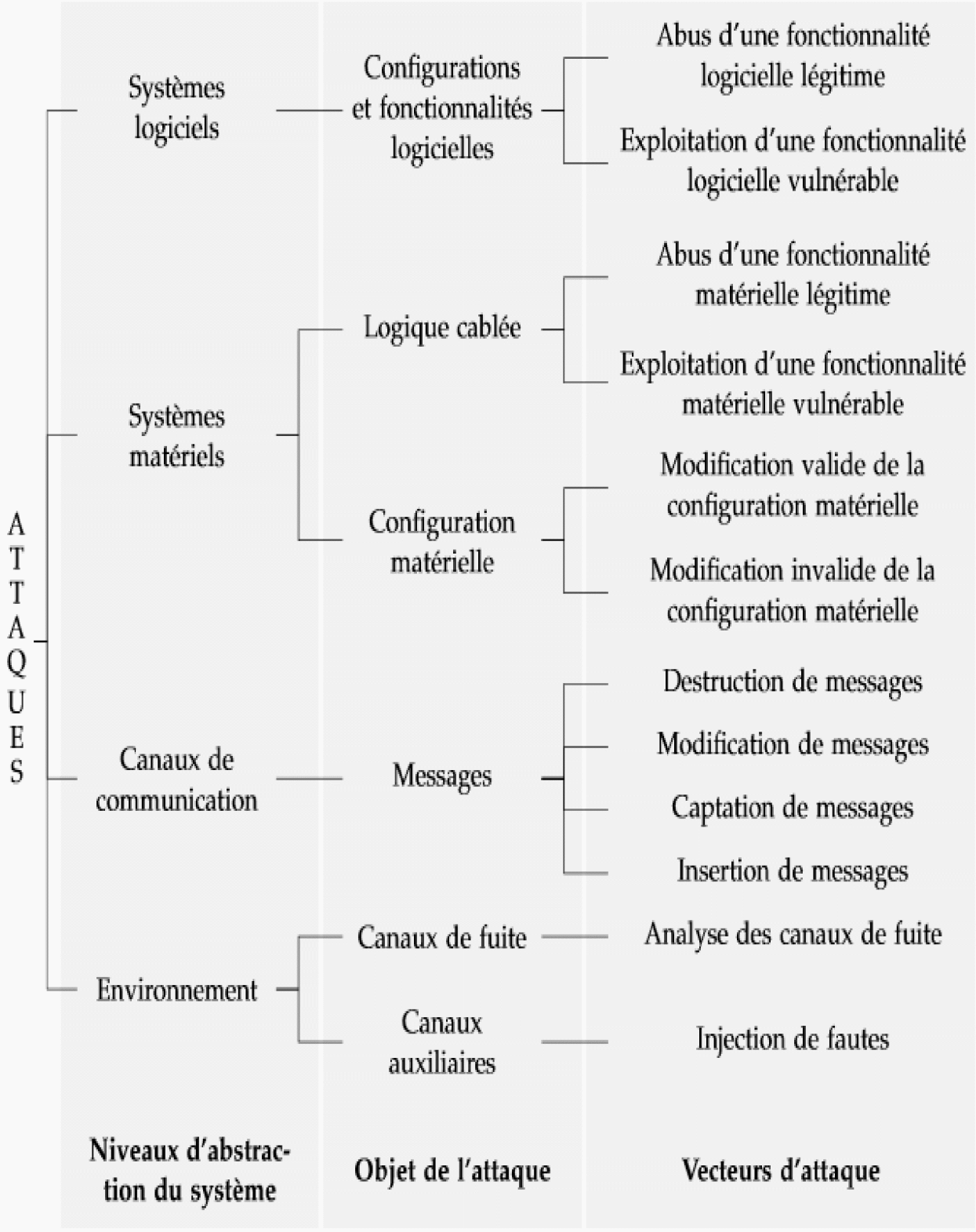

Figure I.11. : Classification des attaques sur les systèmes informatiques [51]

II.5. Les risques et les menaces

❖ Les programmes malveillants [26]

Un logiciel malveillant (malware en anglais) est un logiciel développé dans le but de nuire à un système informatique.

a) le virus : Programme se dupliquant automatiquement sur le même ordinateur. Il peut être transmis à un autre ordinateur par l'intermédiaire du courrier électronique ou par n'import quel moyen d'échange de données.

b) le ver (worm) : exploite les communications réseaux d'un ordinateur afin d'assurer sa reproduction sur d'autres ordinateurs.

c) le cheval de Troie (trojan) : programme à apparence légitime (voulue) qui exécute des routines nuisibles sans l'autorisation de l'utilisateur.

d) la porte dérobée (backdoor) : permet d'ouvrir d'un accès réseau frauduleux sur un système informatique. Il est ainsi possible d'exploiter à distance la machine.

e) le logiciel espion (spyware) : fait de la collecte d'informations personnelles sur l'ordinateur d'un utilisateur sans son autorisation. Ces informations sont ensuite transmises à un ordinateur tiers.

f) l'enregistreur de frappe (keylogger) : programme généralement invisible installé sur le poste d'un utilisateur et chargé d'enregistrer à son insu ses frappes clavier; pour intercepter des mots de passe par exemple.

g) l'exploit : programme permettant d'exploiter une faille de sécurité d'un logiciel.

h) le rootkit : ensemble de logiciels permettant généralement d'obtenir les droits d'administrateur sur une machine, d'installer une porte dérobée, de truquer les informations susceptibles de révéler la compromission, et d'effacer les traces laissées par l'opération dans les journaux système.

❖ Les risques et menaces de la messagerie électronique [27]

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

b) l'hameçonnage (phishing) : un courrier électronique dont 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 destinataire à retransmettre le message à ses contacts sous divers prétextes. Ils encombrent le réseau, et font perdre du temps à leurs destinataires.

Dans certains cas, ils incitent l'utilisateur à effectuer des manipulations dangereuses sur son poste (suppression d'un fichier prétendument lié à un virus par exemple).

❖ Les risques et menaces sur le réseau [28]

a) les écoutes (sniffing) : technique permettant de récupérer toutes les informations transitant sur un réseau (on utilise pour cela un logiciel sniffer). Elle est généralement utilisée pour récupérer les mots de passe des applications et pour identifier les machines qui communiquent sur le réseau.

b) l'usurpation d'identité (spoofing) : technique consistant à prendre l'identité d'une autre personne ou d'une autre machine. Elle est généralement utilisée pour récupérer des informations sensibles, que l'on ne pourrait pas avoir autrement.

c) le déni de service (denial of service) : technique visant à provoquer des interruptions des service, et ainsi d'empêcher le bon fonctionnement d'un système.

Il peut y avoir des tentatives d'extorsion de fond : menacer de stopper l'activité d'une entreprise.

II.6. Objectifs de la sécurité informatique

Le principal objectif de la sécurité informatique est de garantir la sécurité de l'information, du système et du réseau, tout en empêchant :

- la divulgation non autorisée des données.
- la modification non autorisée des données.
- l'utilisation non autorisée des ressources réseaux ou informatique de façon générale.

II.7. Les mécanismes de défenses : [9] [10]

a) **Le chiffrement :** Le chiffrement est un procédé de cryptographie grâce auquel on souhaite rendre la compréhension d'un document impossible à toute personne qui n'a pas la clé de (dé)chiffrement. Ce principe est généralement lié au principe d'accès conditionnel.

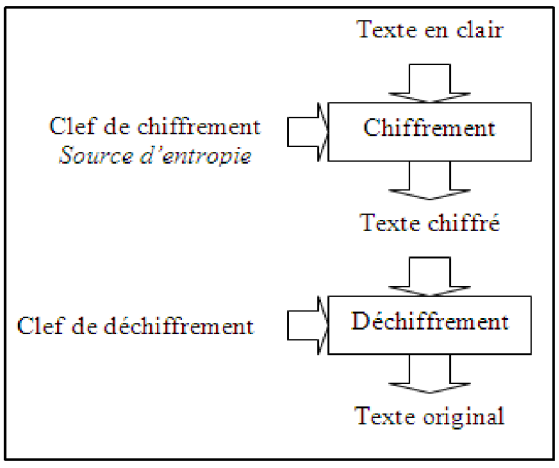

Figure I.12. : Schéma résumant le chiffrement [48]

b) **La signature :** La signature numérique (parfois appelée signature électronique) est un mécanisme permettant de garantir l'intégrité d'un document électronique et d'en authentifier l'auteur, par analogie avec la signature manuscrite d'un document papier.

Elle se différencie de la signature écrite par le fait qu'elle n'est pas visuelle, mais correspond à une suite de nombres.

c) **La notarisation :** Utilisation d'un tiers de confiance pour assurer certains services de sécurité.

d) **Le contrôle d'accès :** Vérifie les droits d'accès d'un acteur (utilisateur) aux données. N'empêche pas l'exploitation d'une vulnérabilité.

e) **Les Antivirus :** Les antivirus sont des logiciels conçus pour identifier, neutraliser et éliminer des logiciels. Ces derniers peuvent se baser sur l'exploitation de failles de sécurité, mais il peut également s'agir de logiciels modifiant ou supprimant des fichiers, que ce soit des documents de l'utilisateur stockés sur l'ordinateur infecté, ou des fichiers nécessaires au bon fonctionnement de l'ordinateur.

f) **Le pare-feu :** C'est un élément (logiciel ou matériel) du réseau informatique 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 contre une attaque venant du réseau intérieur.

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

Il existe différents types de firewalls:

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

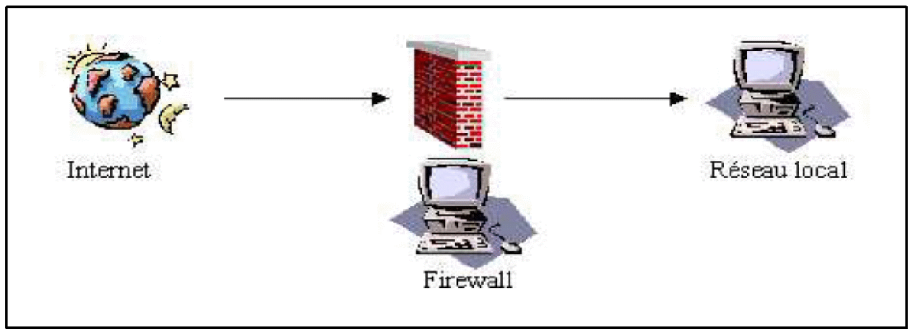

Figure I.13. : Fonctionnement d'un Pare-feu [43]

g) **Les VPN (Virtual Private Network):** un VPN (réseau privé virtuelle) est une abstraction permettant d'isoler un nombre finis d'ordinateur distant, comme si ils appartenaient a un même réseau locale.

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

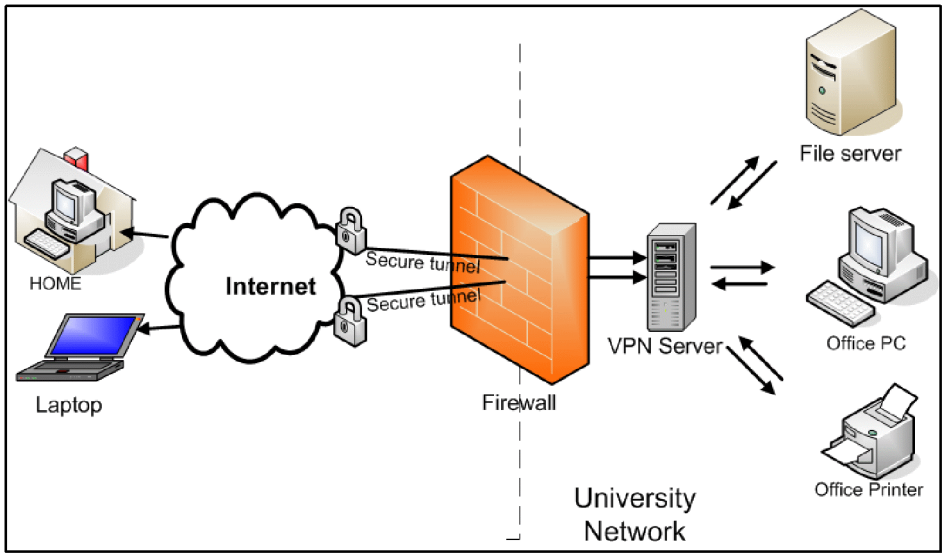

Figure I.14. : Fonctionnement du VPN [44]

h) **Proxy:** C'est un serveur d'isolement qui sert de relai entre le réseau et les machine à cacher, son but est d'isoler une ou plusieurs machines à fin de mieux les protéger.

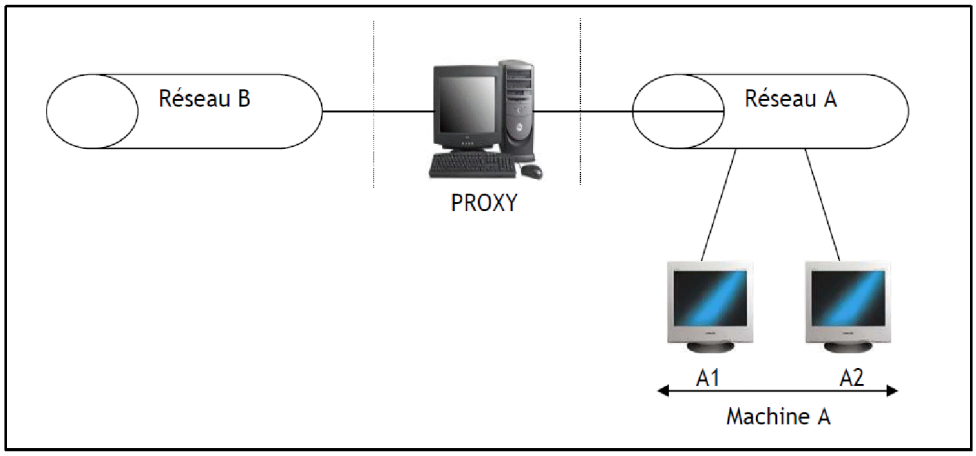

Figure I.15. : Fonctionnement d'un Proxy [43]

i) **La journalisation ("logs") :** Enregistrement 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.

j) **Contrôle de routage :** sécurisation des chemins, c.-à-d. contrôler tout les liens (supports de transmission) et tout les équipements d'interconnexion (routeur, passerelles,…).

k) **Authentification :** Authentifier un acteur peut se faire en utilisant une ou plusieurs de ses éléments.

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

Dans le domaine des communications, on authentifie l'émetteur du 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

l) **La protection physique** : Elle peut fournir une protection totale, mais qui peut être excessive. Par ex. isoler complètement son système est une solution qui peut être trop radicale.

m) **Le DMZ (zone démilitarisé)** : c'est un sous réseau isolée situé entre le réseau local et le réseau extérieur, hébergeant des applications mises à disposition du grand public sans pour autant risquer de compromettre la sécurité du réseau local

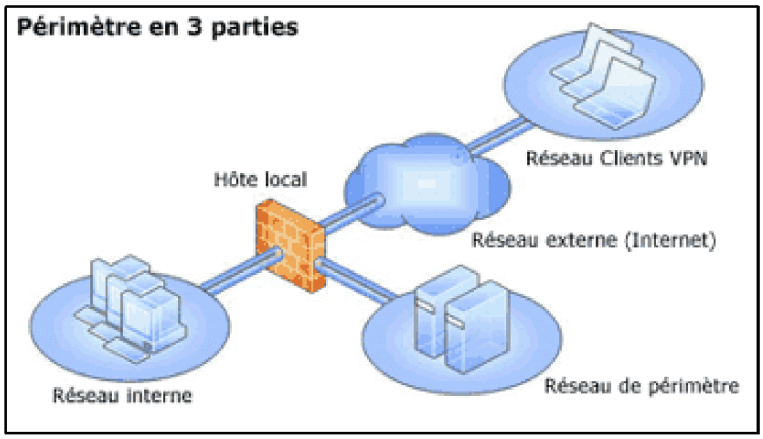

Figure I.16. : Fonctionnement de la DMZ [45]

n) **Horodatage, Certification, Distribution des clefs, …**

o) **Les IDS :** Un IDS (Système de Détection d'Intrusion) est un mécanisme de sécurité qui permet de contrôler l'activité d'un réseau ou d'un système, à fin de déceler la présence de toute tentative d'intrusion et éventuellement de réagir à cette tentative

Elle est basée sur l'analyse a la volée ou en temps différé de ce qui ce passe sur le système.

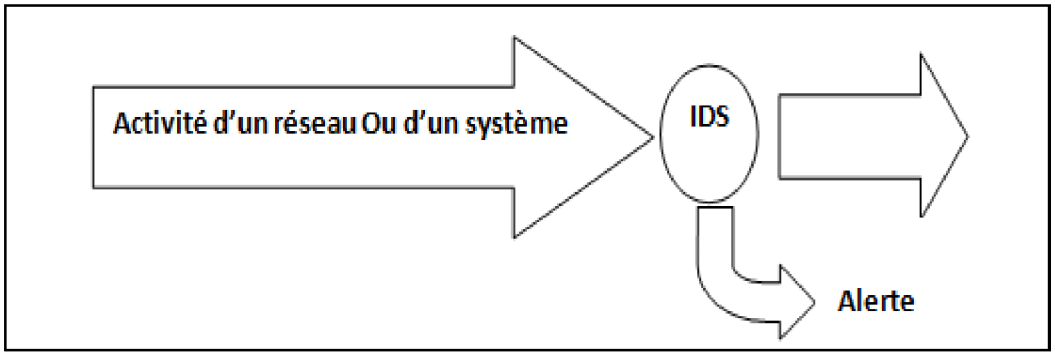

Figure I.17. : Fonctionnement d'un IDS

II.8. Caractéristiques d'un système sécurisé

Intégrité des données

Le contrôle d'intégrité d'une donnée consiste à vérifier que cette donnée n'a pas été modifiée, frauduleusement ou accidentellement.

Confidentialité

Il s'agit de rendre l'information embrouillé à tous les Oscar, aussi bien lors de sa conservation 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'identification consiste à s'assurer que chaque tier est bien lui même (authentification des partenaires) et d'obtenir une garantie que tel tier a bien déclenché l'action (authentification de l'origine des informations).

C'est un problème fondamental, qui exige de faire confiance à un tiers dans le cas où 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 revenir sur ce qu'elle a transmis. Il n'y a pas pu y avoir de transmission de sa part sans son accord.

L'émetteur ne peut pas nier l'envoi d'information ; Le récepteur ne peut pas nier la réception d'information ; ni l'un ni l'autre ne peuvent nier le contenu de cette information (très important lors du passage d'une commande par exemple).

Personne ne pourra prendre l'identité d'un autre pour transmettre une information en son nom.

II.9. Conclusion

Au cours de ce chapitre, nous avons pus vous présentez dans sa premier partie quelques généralités sur les réseaux, leur classification ainsi que quelque modes de leurs fonctionnement, puis nous avons entamé dans la seconde partie, un état de l'art sur les attaques et 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.

les IDS constituent une bonne alternative pour mieux protéger le réseau informatique.

UNIVERSITE MOULOUD MAMMERI DE TIZI OUZOU

Chapitre II : Le système de détection d'intrusion

I.1. Introduction :

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

Afin de détecter les attaques que peut subir un système, il est nécessaire de disposer d'un logiciel spécialisé dont le rôle sera de surveiller les données qui transitent sur ce système et qui serait capable de réagir si des données semblent suspectes.

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

I.2. Les systèmes de détection d'intrusion :

Un système de détection d'intrusion (IDS : Intrusion Detection System) est un mécanisme destiné à repérer des activités anormales ou suspects sur la cible analysée (un réseau ou un hôte).

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

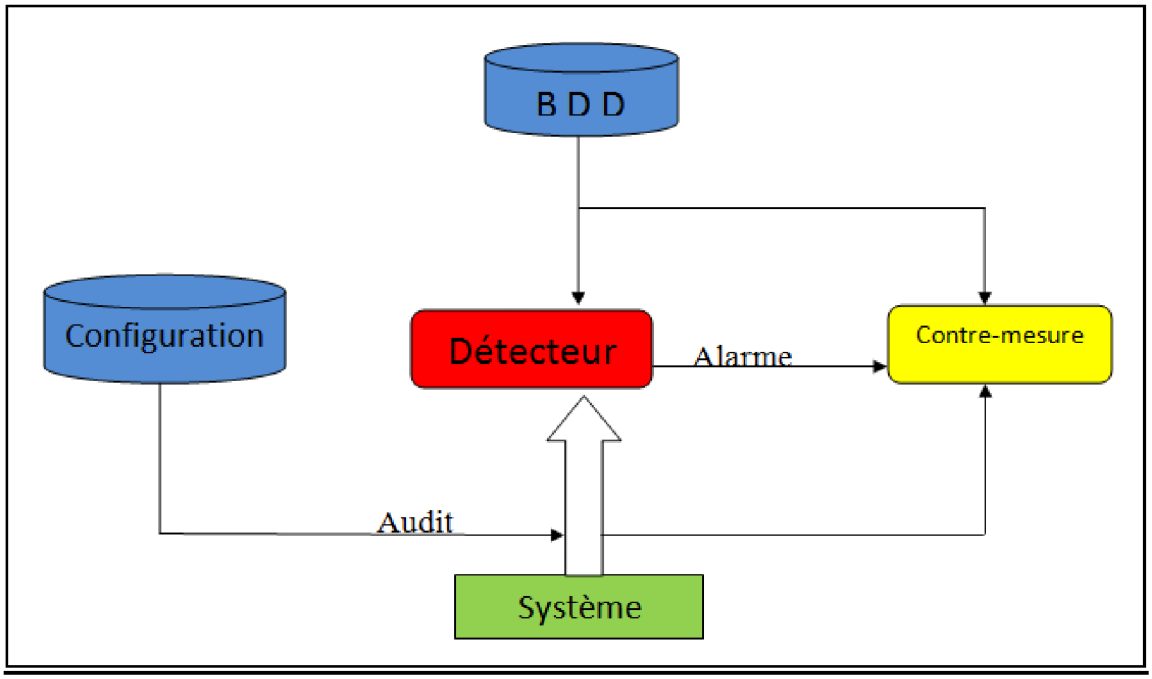

Figure II.1 : modèle simplifié d'un système de détection d'intrusion (IDS)

I.3. Historique : [11]

Les premiers systèmes de détection d'intrusions ont été initiés par l'armée américaine puis par des entreprises.

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

I.4. Fichier historique [12] :

Fichier historique permet d'enregistrer tout ou une partie des actions effectuées sur le système. L'analyse de ses informations permet de détecter d'éventuelles intrusions. Les systèmes d'exploitation génèrent des fichiers historiques, certaines applications aussi. Les différents évènements du système sont enregistrés dans un journal, qui devra être analysé fréquemment, voire en permanence. Sur les réseaux, il est indispensable de disposer d'une base de temps commune pour estampiller les évènements.

Voici les types d'informations à collecter sur les systèmes pour permettre la détection d'intrusions : On y trouve les informations sur les accès au système (qui a accédé, quand et comment), les informations sur l'usage fait du système (utilisation du processeur, de la mémoire ou des entrées/sorties) et les informations sur l'usage fait des fichiers.

Le fichier historique doit également permettre d'obtenir des informations relatives à chaque application (le lancement ou l'arrêt des différents modules, les variables d'entrée et de sortie et les différentes commandes exécutées), Les informations sur les violations éventuelles de la sécurité (tentatives de commandes non autorisées) ainsi que les informations statistiques sur le système seront elles aussi nécessaires.

Notons que ces nombreuses informations occupent beaucoup de place et sont très longues à analyser. Ces informations devront être, au moins pour un temps, stockées quelque part avant d'être analysées par le système de détection d'intrusions.

I.5. Composition d'un IDS : [13]

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'éventuellement sa réponse.
- **L'interface utilisateur :** elle permet au utilisateur de l'IDS de visualiser ou/et de définir le comportement du système.

I.6. Positionnement :

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 placés de manière optimale.

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

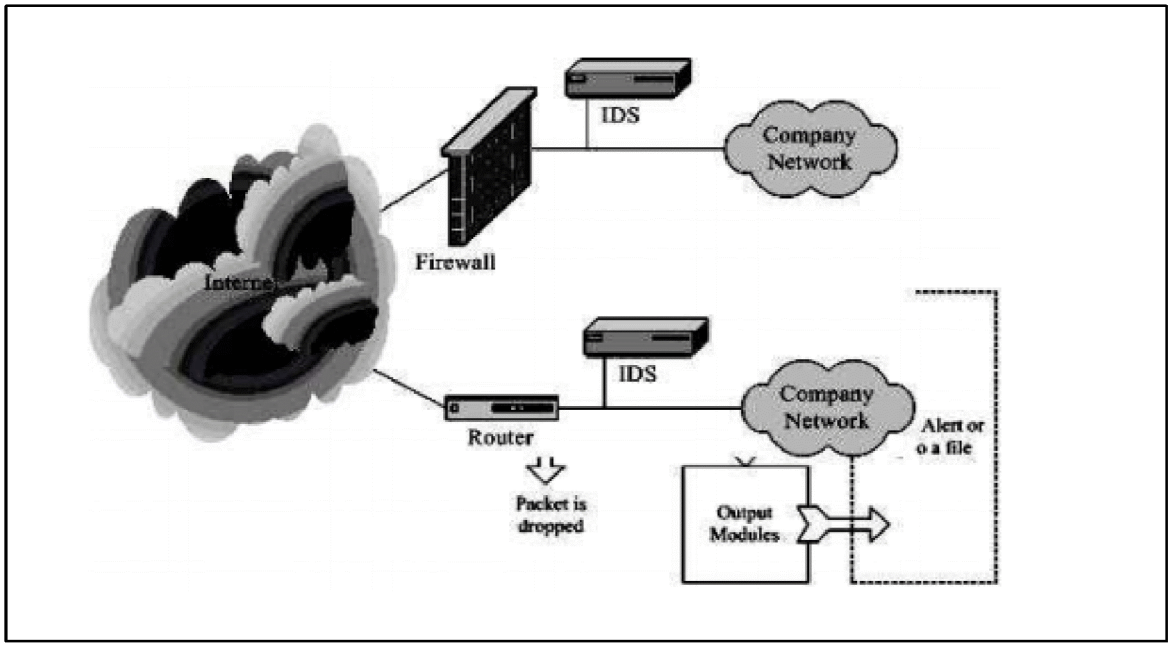

Figure II.2 : le positionnement des IDS [50]

I.7. Schéma d'une attaque :

Une attaque peut être schématisée en six (6) points :

1. La collecte d'information sur le système.
2. L'intrusion dans le système grâce à ces informations.
3. La mise en place d'un système permettant une ré-intrusion futur, tel que l'insertion de code dans l'EEPROM.
4. La recherche d'une propagation de l'intrusion dans un autre système, et ainsi permettre des attaques distribuées.
5. La paralysie du système.
6. L'effacement des traces de l'attaquant.

Il existe essentiellement deux (2) type d'attaque : les attaques réseau et les attaque applicative.

I.7.1. Les attaque réseau :

Les attaques réseau Ces attaques sont souvent dues à une faille du protocole ou de son implémentation. Pour réaliser une telle attaque, une première étape est la récolte d'informations. Cette récolte d'informations peut se faire grâce au social engineering ou grâce à des outils de scan.

Bien que la récolte d'informations ne puisse être précisément dénie comme une intrusion, les IDS réussissent parfois à la détecter. Néanmoins, certaines informations (telles que celles provenant des fournisseurs d'accès à internet qui peuvent être piratées, des informations publiques, …) ne se trouvent pas nécessairement dans le réseau qui va être attaqué. Ainsi, ce genre de récolte ne peut pas toujours être détecté. Le but de ces récoltes est par exemple l'obtention de la version des programmes, les services fonctionnels, les adresses IP ou les ports ouverts dans le système.

Tout ceci facilite la recherche de failles connues dans le système. Les scans peuvent se faire de plusieurs manières selon la volonté d'être plus ou moins visible lors du scan, et plus ou moins informé de la topologie du réseau. Il existe plusieurs méthodes telles que le scan simple, le scan SYN, le scan XMAS, le scan NULL, le scan FIN, le scan à l'aveugle, le scan très lent et le scan passif.

Après la récolte d'informations, l'attaque peut se réaliser. Il existe plusieurs techniques d'attaques connues telles que le IP Spoofing, le ARP Spoofing, le DNS Spoofing, le fragment attack, le TCP Session Hijacking, le man in the middle et le déni de service tel que les SYN spoofing, les UDP spoofing, . . .

I.7.2. Les attaques applicatives :

Ces attaques sont souvent dues à une faille d'un logiciel ou d'une configuration. La première est essentiellement liée à des injections SQL ou à des buffers overflows. Ces derniers permettent l'utilisation de shellcode qui donne la possibilité d'exécuter un code à distance. Les injections SQL sont des introductions de code SQL malveillant dans des requêtes de base de données permettant d'obtenir des informations privées. Enfin, les problèmes dus à une mauvaise configuration sont très répandus. En effet, beaucoup d'administrateurs réseaux préfèrent laisser la configuration par défaut plutôt que de la modifier en risquant de mal l'établir.

I.8. Caractéristiques d'un système de détection d'intrusions:

Les caractéristiques suivantes sont souhaitables dans un IDS :

- ✓ fonctionner en permanence avec une supervision manuelle minimale.
- ✓ être 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 s'il a subit lui-même une modification indésirable.
- ✓ utiliser un minimum de ressources du système sous surveillance.
- ✓ être facilement configurable pour implémenter une politique de sécurité spécifique d'un réseau.
- ✓ s'adapter au cours du temps aux changements du système surveillé et du comportement des utilisateurs.
- ✓ être difficile à tromper.

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

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

I.9. Classification des systèmes de détection d'intrusions : [14] [15]

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

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

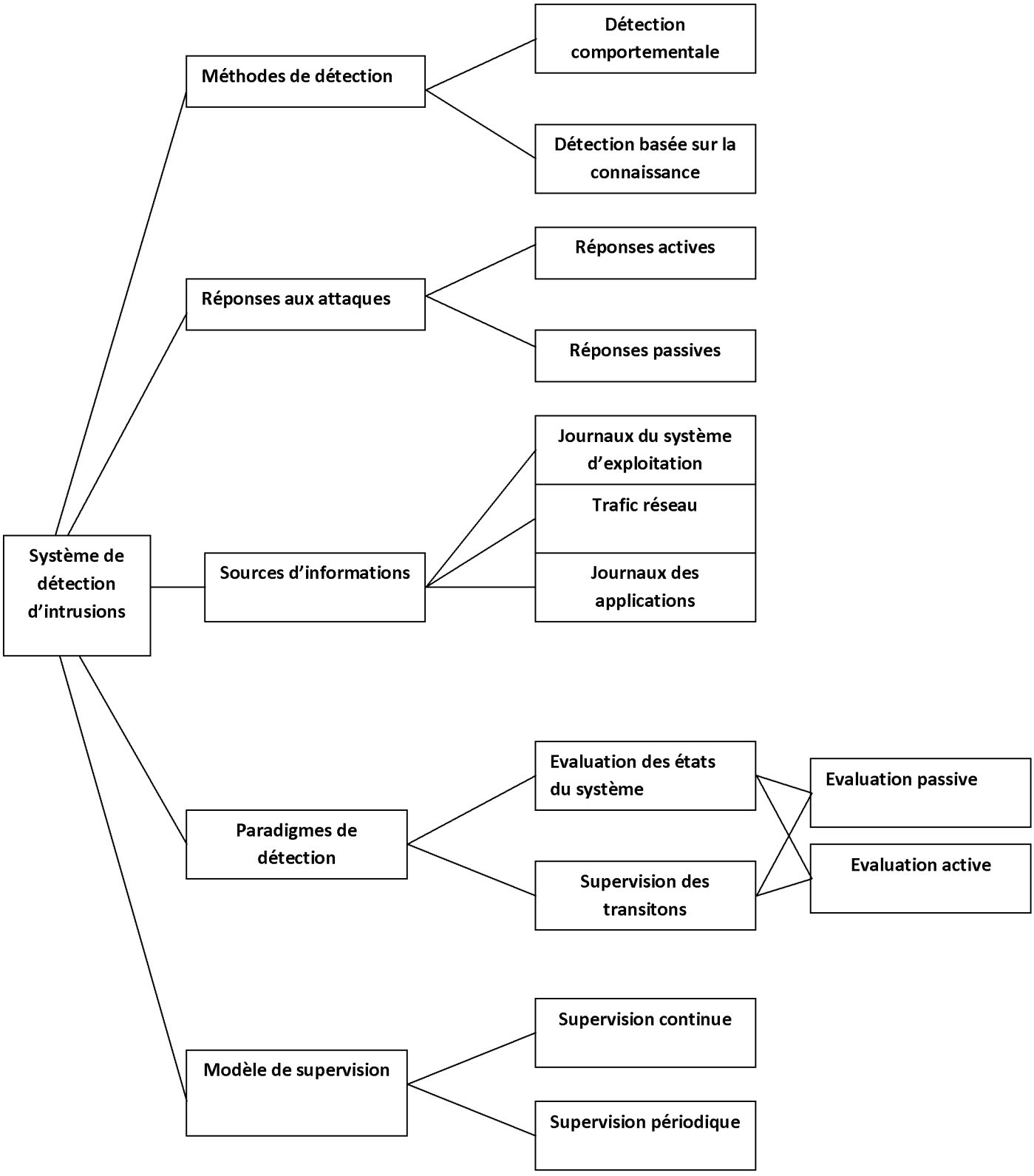

Figure II.3 : Classification des systèmes de détection d'intrusions [28]

I.9.1. Classification selon la méthode de détection :

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

❖ L'approche par scénario :

Cette approche s'appuie sur la connaissance des techniques utilisées par les attaquants pour déduire des scénarios typiques.

On l'appelle détection par abus (Knowledge Based Detection), elle est basée sur la détermination d'une base de données contenant différentes signatures des différentes intrusions. Les IDS utilisant cette approche peuvent reconnaitre les attaques d'après leurs signatures.

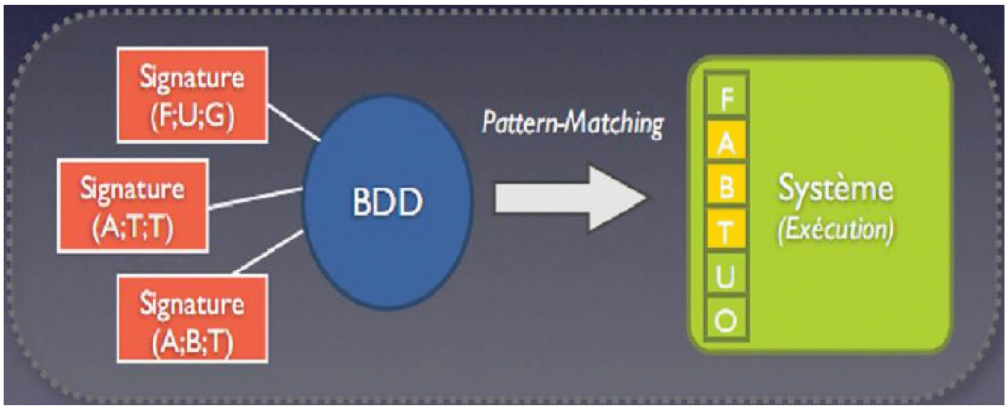

Figure. II.3: Approche par scénario. [49]

Les algorithmes utilisés dans cette approche peuvent être regroupés en deux (2) classes majeures :

a) Pattern matching:

Tel que E2xB, Bayer-Moore et Knuth-Morris-Prat.

C'est la méthode la plus connue et la plus facile à comprendre, elle se base sur la recherche de motif (chaine de caractères, suite d'octets) au sein du flux de données.

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

Le principale inconvénient de cette méthode est que seules les attaques reconnues par leur signatures seront détectées, il est donc nécessaire de mettre à jour régulièrement la base de signature.

b) Les systèmes experts :

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

Cette approche ne peut détecter que les attaques connus précédemment et qui se déroule selon la même signature adapté à la base. Par exemple : si un attaquant change l'ordre des instructions de son attaque, l'IDS trouvera une difficulté pour la détecter.

❖ L'approche comportementale :

Le but de cette technique est la prédiction de comportement.

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

Ainsi des attaques inconnues peuvent être détectées contrairement à l'approche par scénario biaisé par l'empreinte des signatures connus.

Une foi le profil établi, 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 du profile peut se faire grâce a différents paramètre tels que :

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

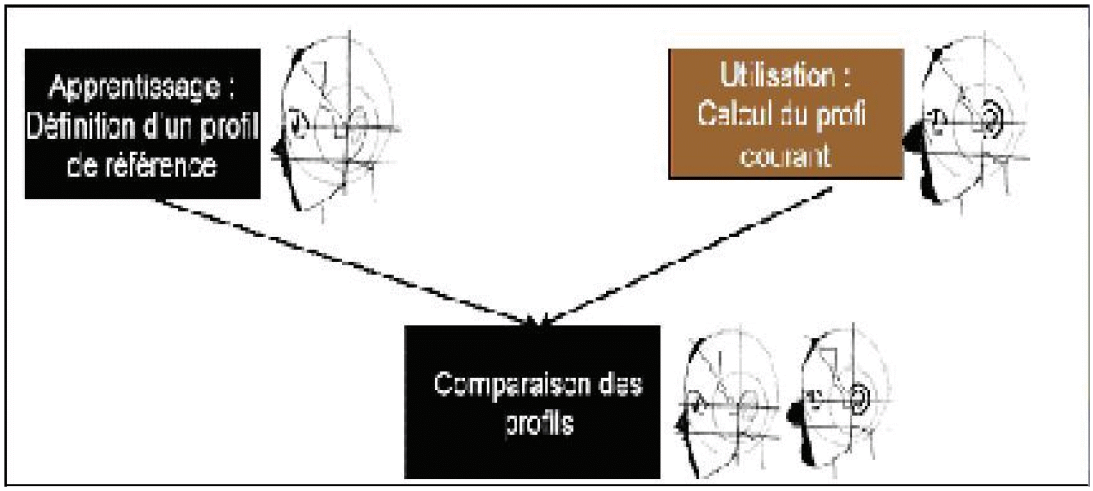

Figure. II.4: Approche comportementale. [F.3]

Les techniques les plus connus sont :

a) Le modèle de DENNING :

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

b) Réseau de neurones :

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

- Les ressources utilisées.
- La vitesse de saisie du clavier.
- Les activités faites par l'utilisateur.

Comparaison entre les deux approches:

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 II.1: comparaison entre l'approche comportementale et l'approche par scénario [F.3]

Conclusion sur les deux approches:

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

I.9.2. Classification selon le comportement après la détection :

Une des façons de classifications des IDS consiste a les partagés selon le type de réaction lorsqu'une attaque est détecté, et on distingue deux types :

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

La plupart des IDS n'apporte qu'une réponse passive à l'intrusion.

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 du système chargé de la sécurité, dans le cas d'une intrusion dangereuse l'IDS doit réagir contre cette intrusion en exécutant des actions tels que :

- La modification de la table de routage du routeur lié au système.
- Le refus ou l'arrêt d'une connexion suspecte.
- L'arrêt d'un processus.
- La demande au pare-feu de modifier ses 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 II.2. Réponses aux attaques des systèmes de détection d'intrusions [T.1].

I.9.3. Classification selon la source de données a analysées :

La source des données a analysées 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 obtenus 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 basés sur l'hôte ou HIDS (Host IDS) analysent exclusivement l'information concernant cet hôte. Comme ils n'ont pas à contrôler le trafic du réseau mais "seulement" les activités d'un hôte 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 a ses 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 être mieux contrôlés et analysés en raison de leur taille, mais certaines attaques peuvent passer inaperçues, alors qu'elles sont détectables par une analyse des traces d'audit.

Les avantages des *HIDS* sont les suivants : [30] [31] [32]

- 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'hôte avec précision et d'optimiser le système en fonction des activités observées.
- ils permettent de détecter plus facilement les attaques de type *Cheval de Troie*, 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 elles font partie du trafic crypté.
- ils génèrent peu de faux positifs, permettant d'avoir des alertes pertinentes.

Les inconvénients des *HIDS* : [30] [31] [32]

- ils peuvent être identifiés et mis hors service par un attaquant.
- ils ne peuvent donner l'alerte que si les entrées des journaux d'événements ou les appels au système correspondent à des signatures ou des règles pré configurées.
- sensibles aux attaques de type Déni de Service.
- ils sont assez gourmands en CPU et peuvent parfois altérer les performances de la machine hôte.

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 être protégés par un HIDS.

Pour finir, voici quelques HIDS connus:

- Tripwire [33].
- WATCH. [34].
- DragonSquire. [35]
- Tiger. [36]
- Security Manager… [37]

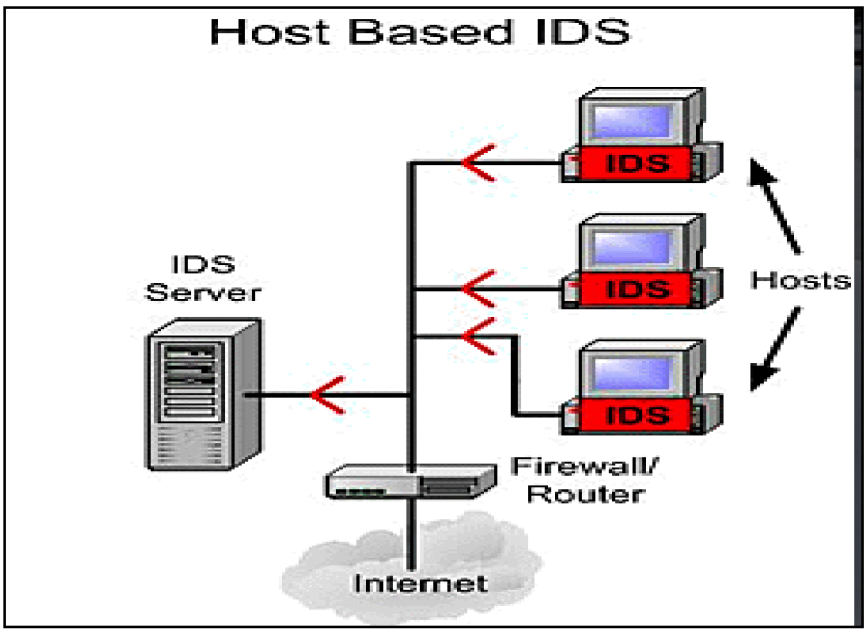

Figure II.5: Emplacement d'un HIDS (hoste IDS).

❖ Les NIDS:

Le rôle essentiel d'un IDS 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 alertes 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 être 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 aucune adresse IP n'est configurée.

Un capteur possède en général deux cartes réseaux, une 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 : il est possible de placer les capteurs à différents endroits, en fonction de ce que l'on souhaite observer. Les capteurs peuvent être placés avant ou après le pare-feu, ou encore dans une zone sensible que l'on veut protéger spécialement.

Si les capteurs se trouvent après un pare-feu, il leur est plus facile de dire si le pare-feu a 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 arrêtées par ce dernier. Il s'agit d'une utilisation 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 peut ainsi recevoir et analyser l'ensemble du 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 tâche ici est donc plus de contrôler le fonctionnement et la

configuration du firewall que d'assurer une protection contre toutes les intrusions détectées (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 mentionnés ci-dessus. Mais elle est très dangereuse 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 variante.

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

Les avantages des NIDS sont les suivants : **[30] [31] [32]**

- ils peuvent être complètement cachés sur le réseau, donc un attaquant ne saura pas qu'il est contrôlé.
- un système NIDS unique peut être employé pour contrôler le trafic d'un grand nombre de systèmes cibles potentiels.
- il peut capturer le contenu de tous les paquets envoyés à un système cible.
- une seule tâche 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 : **[30] [31] [32]**

- le taux élevé de faux positifs qu'ils 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 a réussi.
- il ne peut pas examiner le trafic chiffré.
- il faut des configurations spéciales sur les réseaux commutés pour que le NIDS puisse voir tout le trafic.

Voici quelques exemples de NIDS :

- Snort.[38]
- Benids.. [39]
- Hank. [40]
- Prelude. [41]
- Firestorm. [42]

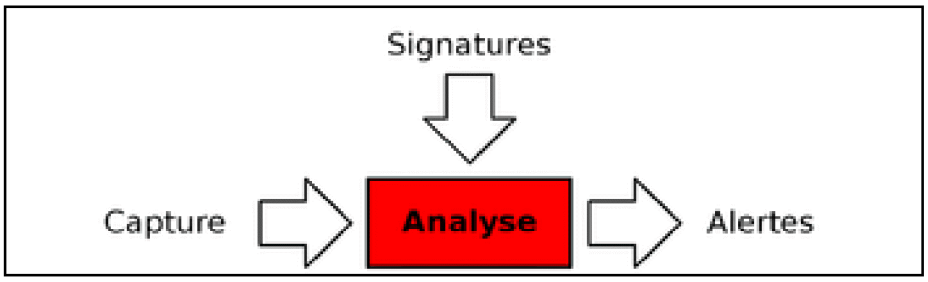

Figure II.6: fonctionnement d'un NIDS. [30]

❖ Les IDS hydrides (NIDS+HIDS) :

Les systèmes de détection d'intrusions hybrides rassemblent les caractéristiques de plusieurs systèmes de détection d'intrusions différents. En pratique, on ne retrouve que la combinaison de NIDS et HIDS. Ils permettent, en un seul outil de surveiller le réseau et l'hôte. 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.

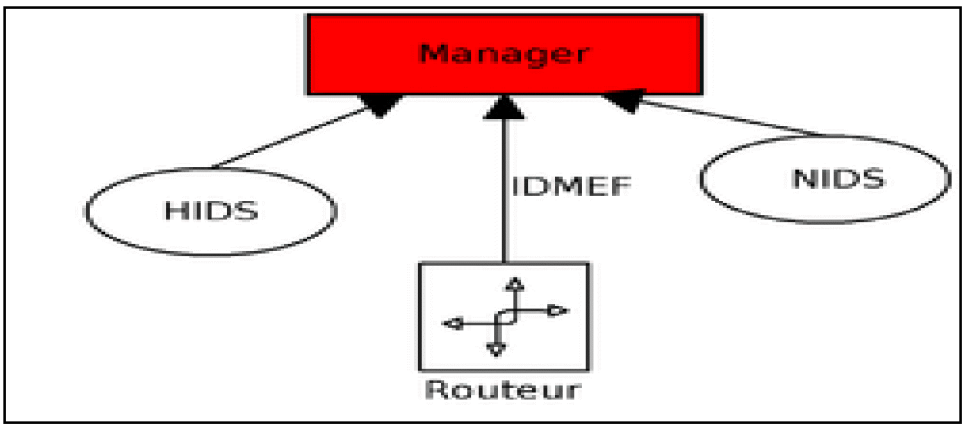

Figure II.7 : fonctionnement d'un IDS Hybride.

I.9.4. Classification selon la fréquence d'utilisation :

On distingue deux types : Online et offline.

❖ **Les IDS online (continue) :**

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

Ce type d'IDS consomme un taux élevé de ressources système, ce qui le rend non adéquat en cas de ressources 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 du système.

L'avantage 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 dégâts dangereux.

I.10. L'architecture des IDS : [16]

On distingue trois architectures globales des IDS selon leur contrôle :

I.10.1. architecture centralisée :

Cette architecture a la même démarche que de l'architecture client/serveur, c'est-à-dire que les IDS sont installés 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 va répondre par un message d'alerte à l'administrateur du réseau ou bien par le lancement d'un contre mesure.

I.10.2. architecture partiellement distribuée :

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

I.10.3. architecture totalement distribuée :

C'est une architecture composée de plusieurs architectures centralisées où 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 message à un autre, cette architecture est efficace en cas des réseaux de grosses tailles.

I.11. Modèles et normalisations :

I.11.1 CIDF :

C'est une architecture standardisée et définie par le DARPA (Defence Advanced Research Project Agency) afin de généraliser un modèle unique des IDS, ce modèle se compose de quatre (4) composants appelés BOX :

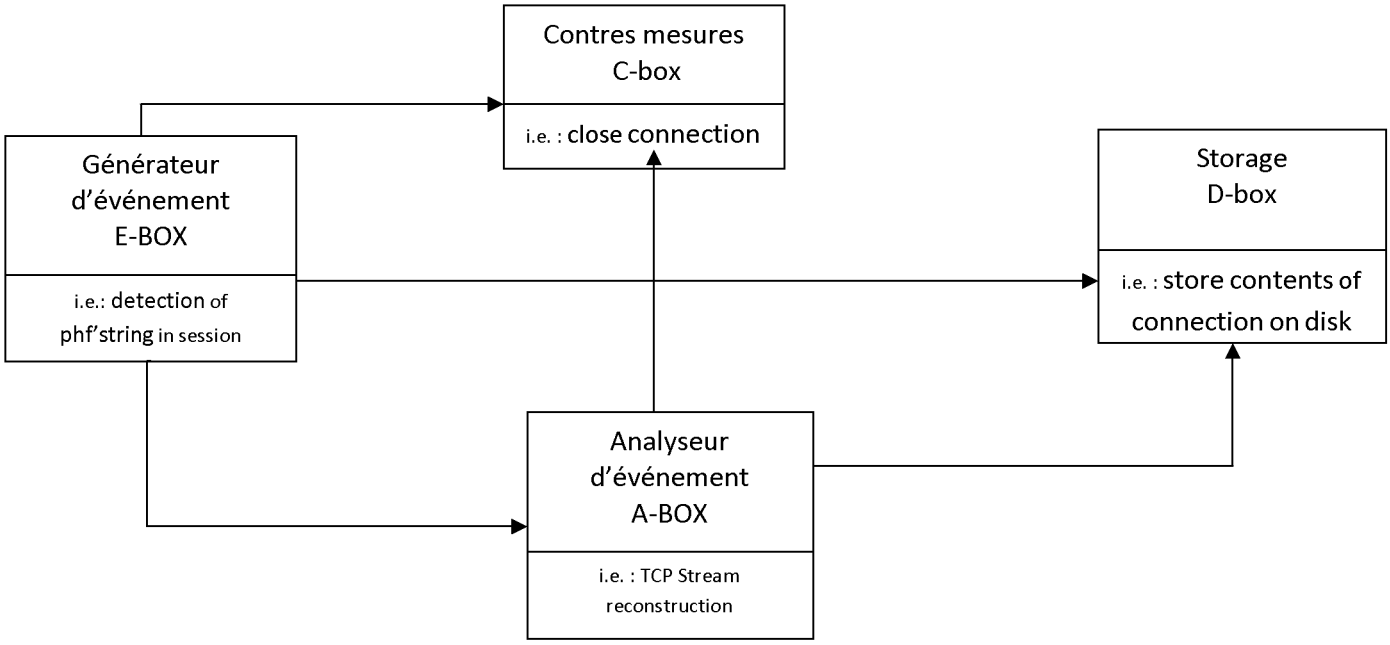

Figure II.8: Architecture CIDEF

a) générateur d'événement : (E-BOX)

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

b) analyseur d'événement : (A-BOX)

Son rôle est d'analyser les informations fournisses par le générateur afin d'extraire des doutes en basant sur l'une des deux approches de détection (par abus ou d'anomalie).

c) stockage d'informations : (D-BOX)

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

d) contre mesure : (C-BOX)

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

I.11.2. IDMEF : (Intrusion Detection Message Exchange Format)

Ceci est une norme qui défini le format des données et des procédures pouvant être partagées ou échangées entre deux(2) éléments d'un IDS ou bien entre deux(2) IDS. Ce langage peut être utilisé par un gestionnaire des informations de sécurité(SIM). Ce langage est basé sur XML.

I.12. Tests des IDS : [13]

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

1. Attaque : on va 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. Alarme : on va regarder le taux des alarmes telles que les faux positifs.

3. Qualité des informations : on va 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 détectée par l'IDS.

6. Qualité des signatures : dans le cas des IDS se basant sur les signatures, il est nécessaire de pouvoir évaluer la qualité des signatures.

7. Rapidité du système : il est nécessaire que l'IDS soit capable de gérer un grand nombre de données en un temps raisonnable et de détecter l'attaque en un minimum de temps pour réduire les dommages causés.

8. Intégration : puisque les IDS ne suffisent pas pour garantir l'ensemble de la sécurité, ils doivent être facilement installés et intégrés à son infrastructure.

9. Interaction : le nombre d'interactions entre un IDS et l'administrateur système doit être minime.

10. Dataset : on va comparer les performances de l'IDS avec d'autres IDS grâce à des datasets.

I.13. Quelques IDS existant : [17]

Le marché des IDS est très vaste. Certains pro duits sont gratuits et d'autres payants.

On se propose de présenter certain logiciels existant tel que :

a) HAYSTACK:

Se programme a été développer 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(6) types d'intrusions :

- Lorsqu'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 du système.
- Un utilisateur vient tenter d'extraire des données potentiellement sensibles dans le système.
- Un utilisateur bloque l'accès aux ressources pour les autres utilisateurs.
- Autres attaques telle que l'effacement des fichiers.

Pour parvenir à ses fins Haystack utilise les deux méthodes de détection: par détection d'anomalies et par signatures. La détection d'anomalies utilise 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, évitant une dérive trop importante du premier modèle utilisateur. Il est ainsi impossible à un intrus d'habituer le système à un comportement intrusif.

b) MIDAS :

MIDAS (Multics Intrusion Detection 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 appelant P-BEST (Production Based Expert System Toolset) et trois (3) catégories de règles :

❖ Attaque immédiate :

Ces attaques sont menées sans faire connaitre l'historique du système sur une petite fenêtre d'événement, généralement un, leur heuristiques sont statiques ne pouvant être échangées que par l'intervention d'un administrateur de sécurité.

❖ Anomalie d'un utilisateur :

Pour cela on sait deux(2) profils statistiques : profil de session qui est temporelle (lors de l'utilisation de session) et qui est mis à jour à la rentrée d'utilisateur et d'après un autre profil s'appelant le profil d'utilisateur qui dure dans une longue période et qui est mise à jour à la sortie d'utilisateur et d'après le profil de session.

❖ Etat système :

Ses heuristiques a 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 même durant toute la durée d'utilisation. Sa méthode de calcul est statistique son rôle est de classer les comportements dans des groupe d'où chaque groupe contient un nombre de comportements proches entre eux, il observe trois(3) types de sujets : l'utilisateur, les hôtes distants ainsi que les systèmes cibles. Au total il

mesure 36 paramètres : 25 pour l'utilisateur, 6 pour les hôtes, 5 pour les systèmes cibles. On peut classer ces mesures en deux(2) 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.

I.14. Conclusion :

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

Plusieurs études on 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 être basée.

UNIVERSITE MOULOUD MAMMERI DE TIZI OUZOU

CHAPITRE III : Arbre de décision & la bases d'apprentissage et de test KDD.

<u>Partie I</u> : Les Arbres de décisions.

<u>Partie II</u> : la bases d'apprentissage et de test KDD.

Partie I: Arbre de décision

I.1. Introduction

Au cours des chapitres précédent nous avons constaté que, malgré la multitude de types d'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 classification

Il existe plusieurs techniques de classification :

- Séparateur à vaste marge (SVM).
- les réseaux bayésiens.
- Les réseaux de neurones.
- Les méthodes basées sur les règles.
- Les méthodes basées sur les arbres de décision.

Les arbres de décision sont une des techniques les plus populaires et les plus utilisées de l'apprentissage automatique et la fouille de données, c'est pour cette raison que notre thème est centré sur cette technique de classification.

CHAPITRE III : Arbre de décision & la bases d'apprentissage et de test KDD.

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

I.2. définitions

I.2.1. L'apprentissage automatique [10]:

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

Types d'apprentissage :

Il existe plusieurs modes d'apprentissage employées 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 support.
- le boosting.
- les réseaux de neurones pour un apprentissage supervisé ou non-supervisé.
- la méthode des k plus proches 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.

I.2.2. Les arbres de décision :

Un arbre de décision est un algorithme de classification supervisée qui est souvent utilisé pour représenter des connaissances, des informations ou encore des observations qu'on appelle aussi des exemples.

Un arbre de décision est représenté sous forme graphique d'un diagramme illustrant des règles de décisions, il est composé de :

- **Nœuds de décision :** Autrement dit « nœuds internes » chaque nœud est étiqueté par un teste portant généralement sur un seul et unique attribut.
- **Branches :** Ce sont des arcs issus des nœuds de décision correspondant à l'une des valeurs possible des attributs sélectionnés.
- **Nœuds feuilles :** Comprenant les objets qui appartiennent a la même classe.

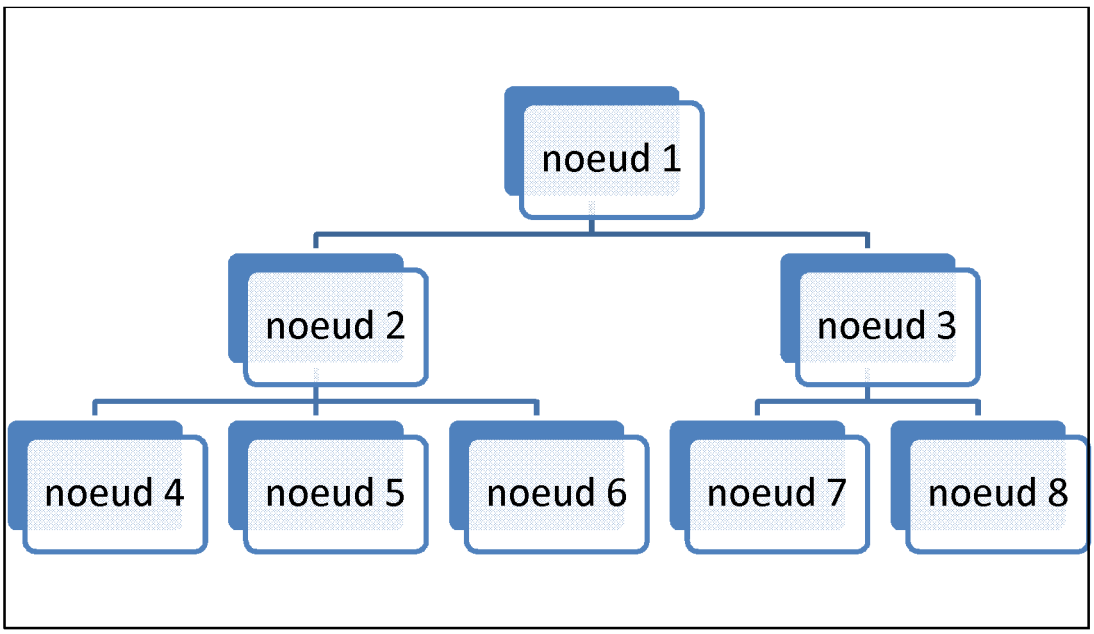

Figure III.1. : Représentation graphique d'un arbre

I.3. petit historique [19] :

Plusieurs recherche 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 optimale de l'ordre de segmentation des options intermédiaires.

Date	Auteur	Apport
1963	Morgan et Sonquist	Arbres de décision dans un processus de prédiction et d'explication
1980	Gordon 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 III.1 : Petit historique sur les algorithmes de classification.

I.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 **[21]**:

I.4.1. Etape de construction [22] [10]:

Avant d'utiliser l'arbre de décision, il faut bien évidement passé par l'étape principale qui est bien sûr 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 côté le cas des variables continues. La quasi-totalité des méthodes d'induction d'arbres s'appuient sur une technique très fruste : chacune des méthodes 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 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) Traitement 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 continues est 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 du 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 [22]:

Cette étape consiste à fixer comme règle d'arrêt de construction de l'arbre la constitution de groupes pure du point de vu 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.

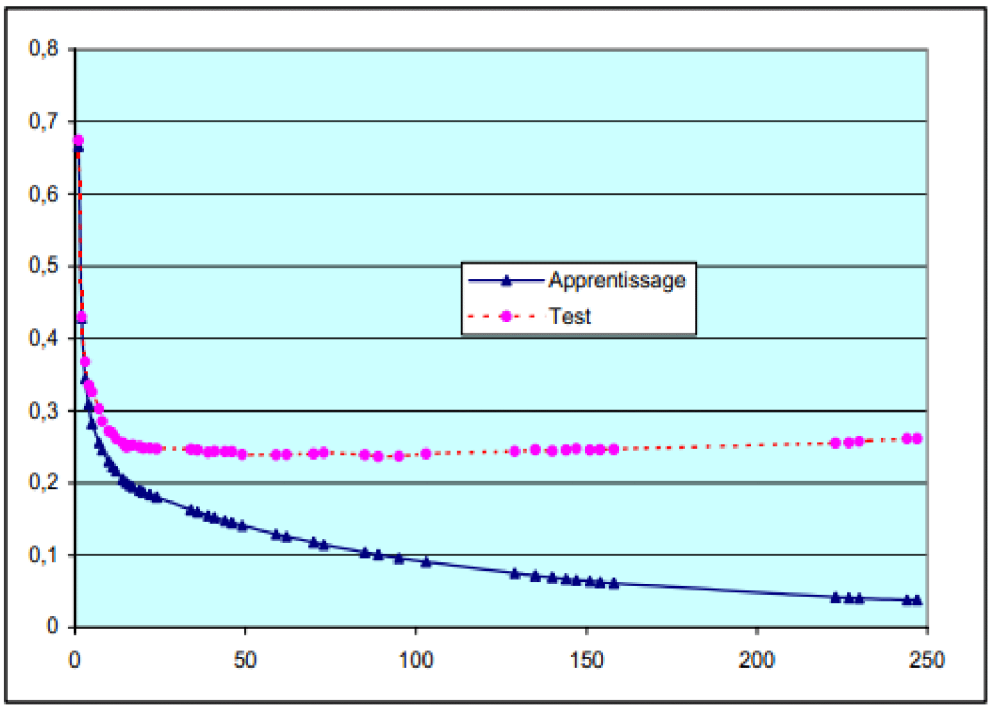

Figure III.2 : Evolution du taux d'erreur en apprentissage et en test en fonction de nombre de feuilles [22].

Nous voyons effectivement dans cette figure qu'à mesure que le nombre de feuilles (la taille de l'arbre) augmente, le taux d'erreur calculé sur les données d'apprentissage diminue constamment. En revanche, le taux d'erreur 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'erreur 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 plu stable dans ses prévisions futures) 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 performance/complexité** des modèles utilisés.

Dans le cas des arbres de décisions, plusieurs types de solutions algorithmiques ont été envisagés pour tenter d'éviter autant que possible un problème de sur-ajustement potentiel des modèles : il s'agit des techniques dites de pré ou de post élagage 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 si 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 à 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 nouveaux cas.

d) Affectation de la conclusion de chaque feuille :

Une foi la construction de l'arbre achevée, 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.

I.4.2. Etape de classification :

Cette étape consiste à classes les objets, tout en parcourant l'arbre de décision en descendant de la racine vers les nœuds plus bas jusqu'aux feuilles, en répondant aux différents testes qui libelle chaque nœuds selon les valeurs des attributs de l'objet à classer

I.5. Algorithmes d'apprentissage par arbre de décision [20]

L'idée centrale de construction des arbres de décision consiste à diviser récursivement et le 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 obtienne des sous-ensembles d'exemples ne contenant (presque) que des exemples appartenant à une même classe. Cette idée débouche sur des méthodes de construction Top-Down, c'est-à-dire construisant l'arbre de la racine vers les feuilles, gloutonnes et récursives.

En générale dans toutes les méthodes d'apprentissage par arbre de décision, on trouve les trois principaux operateurs 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 `a 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 III.3. : Algorithme d'apprentissage général [20]

I.6. les méthodes d'apprentissage [10] [20] :

I.6.1. méthode ID3

Publiée 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 appelle ce calcul l'entropie de Shannon dont la formule est la suivante :

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

I.6.2. méthode C4.5

Technique développée par Ross Quinlan. L'algorithme C4.5 est une amélioration de ID3, notamment du point de vue de la facilité d'implémentation. Il est souvent désigner comme un classificateur statique car les arbres de décision crée par cet algorithme peuvent être utilisés pour la classification.

C4.5 construit des arbres de décision de la même manière que ID3, en utilisant le concept d'information entropie. A chaque nœud intermédiaire de l'arbre, C4.5 divise l'ensemble d'échantillons en sous-ensembles enrichi en une seul classe tout en choisissant un test appropriée, le critère de division est bien le gain d'information normalisé (différence d'entropie).

I.6.3. CHAID (Chi-squared Automatic Interaction Detector)

Technique publiée 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 à ce que la taille de chaque groupe ne devienne pas trop petite, ce qui rendrait l'analyse peu fiable. CHAID est une technique qui peut être utilisée pour la prédiction ou pour la détection d'interaction entre variables.

CHAID détecte l'interaction entre variables dans un jeu de données. En utilisant cette technique on peu établir des relations de dépendance entre variables.

I.6.4. méthode CART (Classification And Regression Trees) [10] [20] [29]

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 modalités) ou continus (à valeurs réelles). Le nombre de tests 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 supposons 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 :

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

Tel que : - p : la position du nœud

- C : nombre des class
- 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, où i0 et n0 sont des paramètres à fixer.

2. **Sélectionner un test à associer à un nœud :** Soit p une position et soit *test* un test. Si ce test devient l'étiquette du nœud à la position p, alors on appelle 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 *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 du gain dans le cas de deux classes en choisissant pour fonction i la fonction de Gini. En position p (non maximale), on choisit le test qui maximise la quantité Gain(p,test).

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

I.6.5. D'autres méthodes moins considérer comme :

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

Partie II: la bases d'apprentissage et de test KDD

II.1. Introduction

Comme nous l'avons déjà expliqué auparavant, l'approche par scénarios analyse des données d'audits à la recherche de scénarios d'attaques dont les signatures sont stockées dans une base de signatures. Le principe consiste à considérer que tout ce qui est décrit dans la base de signatures est intrusif, le reste est considéré comme normal.

A l'inverse, l'approche comportementale ne présuppose pas l'existence d'une base de signatures. Elle se base souvent sur un mécanisme d'apprentissage et propose de décrire le comportement « usuel » ou « normal » d'un utilisateur. Toute déviation par rapport à ce comportement normal est considérée comme une intrusion.

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 normal du système. Mais avant, décrivant d'abord la base de données qui a été utilisée dans notre expérimentation.

II.2. Qu'est ce que le KDD ? [23] :

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

- données : Valeur d'une variable pour un objet.
- information : Résultat d'analyse des données.
- connaissances: Ensemble des informations acquiert par l'étude, l'observation ou bien l'expérience sur les informations.

II.3. Description de la base KDD [21] [24] [25]:

Les données utilisées dans cet article, sont celles de KDD'99 et sont orientées détection d'intrusions (KDD). Chaque ligne code un flot de données (entre deux instants définis) entre une source (identifiée par son adresse IP) et une destination (également identifiée par son adresse IP), sous un protocole donné (TCP, UDP...). Dans la suite de l'article, nous appellerons « connexion » chaque ligne de la base KDD'99 suivant ainsi la description fournie par KDD. Chaque « connexion » est caractérisée par 41 attributs tels que sa durée, le type du protocole, etc. Ces attributs ont 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 DARPA. Ces données présentent 7 semaines de données libellées pour l'apprentissage et 2 semaines de données non libellées pour le test (Correspondant au trafic réseau simulant un réseau local d'US Air force).

II.4. Les attaques de la base KDD'99 [25] :

La base de données KDD'99 recense 38 attaques possibles (listées dans le tableau de la figure 1) qui peuvent être regroupées en quatre catégories **[21]**:

- **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 sont à but purement « destructeur » et sont souvent très simples à mettre en place et donnent une sensation de puissance à l'attaquant, ce qui explique leur fréquence. Un exemple d'attaques DOS est le Smurf qui provoque un deni de service via des requêtes d'écho ICMP manipulées à une adresse de diffusion d'un réseau.

- **Les attaques de type « Remote to Local access » (R2L) :**

  Ce type d'attaque essaie d'exploiter la vulnérabilité du système afin de contrôler la machine distante. Comme exemple d'attaque R2L, il y a celle qui vise des failles des protocoles IMAP (Internet Message Access Protocol). 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)** :

  Où l'attaquant essaie d'avoir les droits d'accès à partir d'un poste afin d'accéder au système. Un exemple d'attaques 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 « destructrices » au sens où 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 Administrator Tool for Analyzing Net-works), 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 III.2 : Type d'attaques. [25]

La KDD contient deux types de bases de connexions [21]:

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 : acquisition 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 [24] :

- Enregistrement :

  - 41 attributs + nom de classe pour vérifier

- ~311000 connexions

  - 4 classes d'attaques enrichies + trafic normal

    - Probing : scan de port (mscan, saint).
    - DoS : déni de Service (apache2, …).
    - U2R : acquisition des privilèges d'un super utilisateur (sqlattack…).
    - R2L : accès illégitime à partir d'une machine distante (snmpguess, snmpgetattack…).
    - Normal : trafic légitime.

II.5. Les attribue caractérisant chaque connexion [21]:

Les attributs caractérisant chaque connexion de la base KDD, sont détaillés dans le tableau de la figure 2. 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 »

Tableaux III.3 : Liste des attribues [21].

II.6. 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 les chapitres suivant.

Puis nous avons étalé dans sa deuxième partie la base d'apprentissage et de *test KDD* qui est utilisée dans notre étude expérimentale, en effet, 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.

UNIVERSITE MOULOUD MAMMERI DE TIZI OUZOU

CHAPITRE IV : Conception et test de l'IDS.

IV.1.Introduction

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

L'IDS que nous allons concevoir sera nommé ***CARTIDS*** **(Système de Détection d'Intrusion basé sur l'algorithme CART)**. ***CARTIDS*** est un système de détection d'intrusions 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 du système à surveiller. Cette application (cet arbre) consiste à classifier les différentes connexions de la base d'apprentissage *KDD*.

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

IV.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 souhaitées d'un IDS qui sont les suivantes :

- Fonctionnement 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 reste du réseau).
- Supervision de plusieurs stations tout en fournissant des résultats de manière rapide et précise.

IV.3. Structure *CARTIDS*

Le système ***CARTIDS*** est structuré de deux grandes phases : (voir figure IV.1)

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

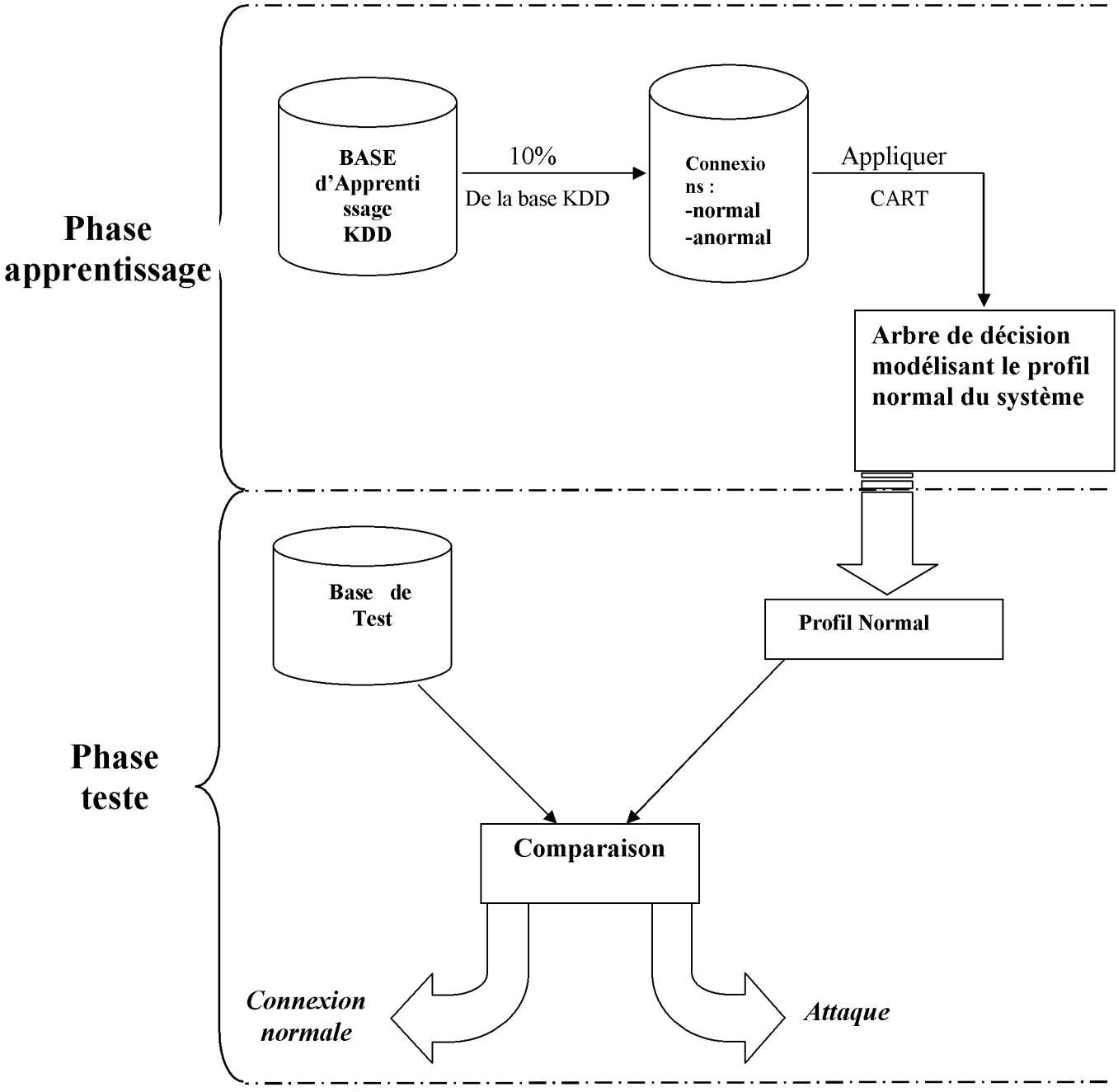

Figure IV.1 : Structure CARTIDS.

1. Phase d'apprentissage

Pour la modélisation du comportement normal du 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, tel qu'il est illustré dans le tableau III.3., la dernier pour spécifier si c'est un comportement normale 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.

Génération du profil normal du système:

Après avoir construit l'arbre de décision en utilisons l'algorithme de **CART** nous avons pu générer le profil normal du système, et pour cela nous avons utilisé différentes classes et méthodes qu'on détail dans ce qui suit.

1.1. Définition de la classe remplissageTable:

Le premier travail à faire consiste à construire une table ***tabCon*** qui est un tableau de double (réels) à 2dimentions indexé en ligne par le nombre de connexion et en colonne par le nombre d'attributs.

Exemple: ***tabCon[i][j]***: représente l'attribut att_j de la $i^{ème}$ connexion

Cette table va contenir toutes les connexions de la base KDD. Pour se faire, nous avons proposé le module nommé ***remplissageTable.*** Ce module utilise le fichier d'entrée nommé ***baseKDD.txt*** contenant 494019 connexions (97278 connexions normales et 396741 connexions anormales) et génère une table ***tabCon*** contenant les connexions de la base KDD (10%). Chaque ligne de la table ***tabCon*** correspond à une connexion, et chaque colonne correspond à un attribut (où tous les attributs qualitatifs sont codés en double comme spécifié dans le tableau IV.1).

Type du protocole	Service réseau	Statut de connexion	Nature de la connexion
TCP=1 UDP=2 ICMP=3 Private=4 Domain=5 . .	HTTP=1 SMTP=2 Telnet=3 Finger=4 ecr_i=5 eco_i=6 . .	SF=1 RSTO=2 SO=3 S1=4 S2=5 . .	normal=1 smurf=2 neptune=3 Loadmodule=4 . .

Tableau IV.1: table de correspondance des attributs qualitatifs.

Exemple:

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

0,tcp,telnet,SF,239,486,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,19,19,1.00,0.00,0.05,0.00,0.00,0.00,0.00,0.00,0.00,normal

La même ligne après l'application de la méthode:

0,1,3,1,239,486,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,19,19,1.00,0.00,0.05,0.00,0.00,0.00,0.00,0.00,0.00,1

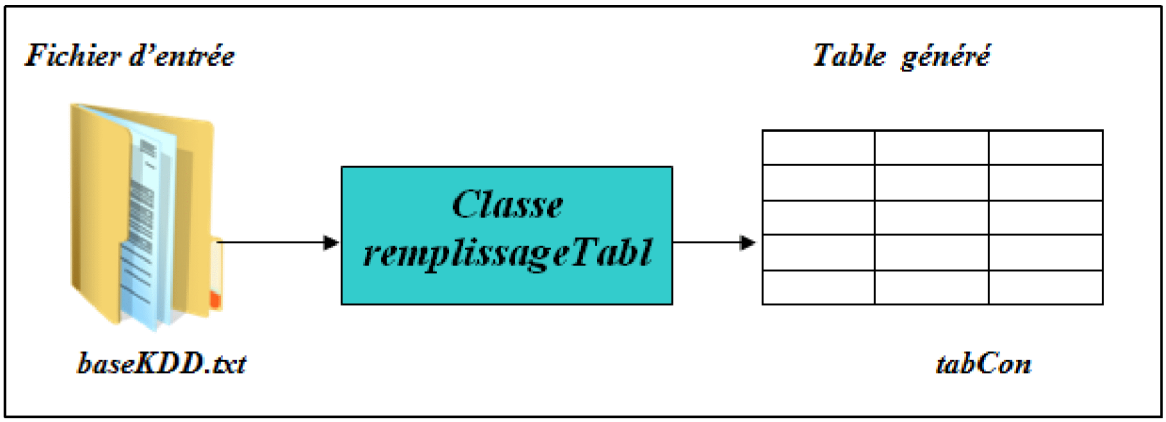

Figure IV.2 : construction de la table tabCon

1.2. Définition de la classe unNoeud:

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

- nbr_indv: nombre de connexions destinées.
- indFils_g: indice du fils gauche
- indFils_d: indice du 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 unNoeud

	{
		int nbr_indv, classe, indFils_g, indFils_d, att;
		double test;
		string classe;
		double tabCon [][]=new double[nbr_indv][42];
	}
```

***Figure VI.3*: Code de la classe unNoeud.**

1.3. 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'un tableau d'enregistrement dont la taille est inférieure ou égale à 41 (le nombre d'attributs qui caractérise une connexion).

Chaque enregistrement fait référence à un nœud de l'arbre qui est une instance de la classe ***unNoeud.***

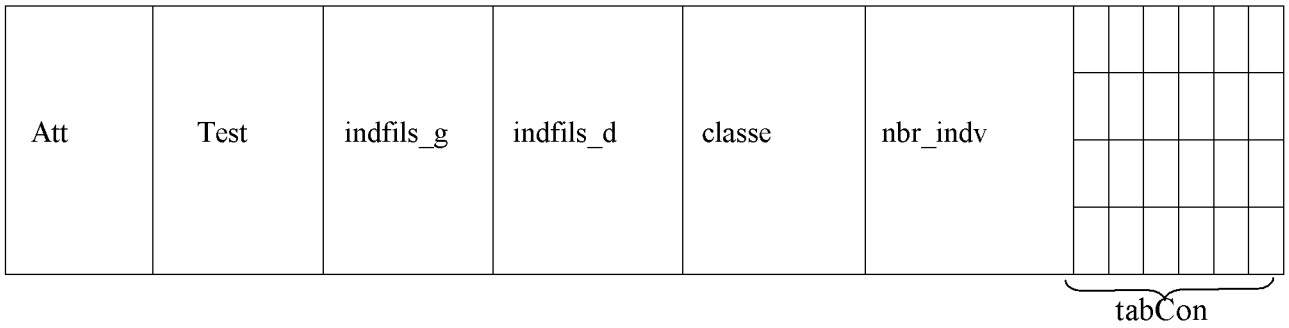

Figure IV.4. : Structure d'un enregistrement nœud.

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

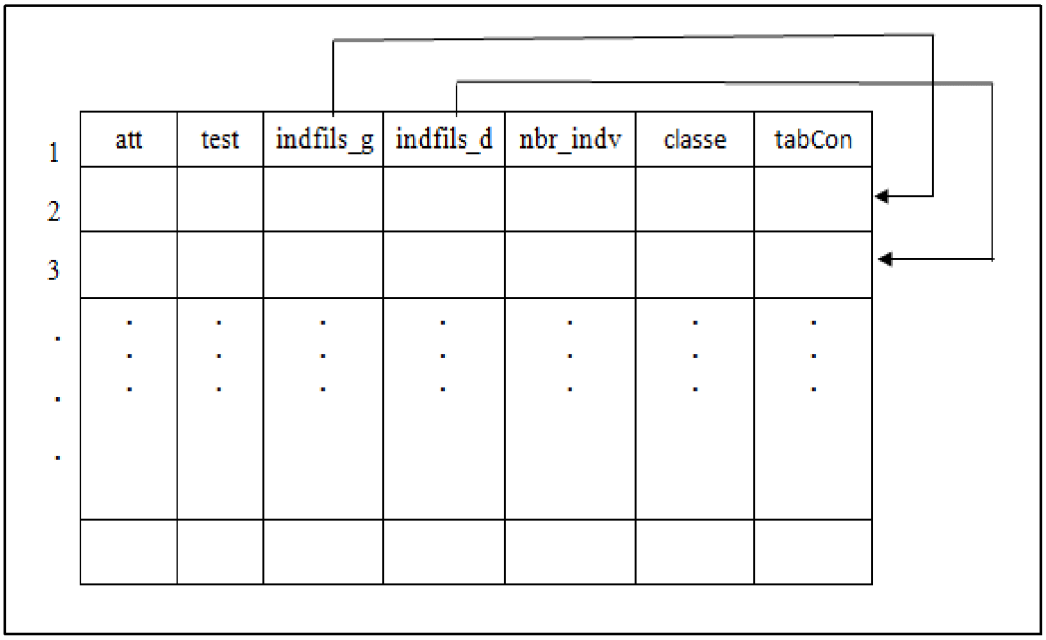

Figure IV.5 : Aperçu de tabCon

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

1.3.1. 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 à segmenter 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:

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

- C : nombre des class (2 classes dans notre cas)

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

1.3.2. Définition de la méthode appTest:

Dans la phase d'apprentissage **appTest** est la méthode principale qui fait la construction de l'arbre, elle reçoit comme paramètres un nœud (son indice dans la table des nœuds plus précisément) elle se base sur l'algorithme qui est illustré dans la figure IV.3. à fin d'effectué les tache suivante:

- ✓ 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 égale à deux fois l'indice du père.
- ✓ Calculer l'indice du fils droit qui est égale à deux fois l'indice du 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 valeurs de la variable ;
            Trouver les points de coupures ;
            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 de gain pour ce point;
            Choix de la variable avec son seuil de test (point de coupure
                qui maximise le gain) qui optimise le gain;
            Créer les fils gauche et droit ;
    jusqu'à la fin des variable.
finsi
```

Figure IV.6. : Algorithme de la méthode appTest

1.3.3. Définition de la méthode main

Elle remplit les taches suivantes :

- ✓ Déclaration des variables globales.
- ✓ Faire appelle à la classe ***remplissageTable*** pour construire la table ***tabCon*** qui contient toutes les connexions de la base KDD.

- ✓ Création du nœud racine (de type unNoeud) et remplissage de ses deux champs :
  - ***Nbr_indv*** := nombre totale des connexions de la base KDD.
  - ***tabCon*** := la table créée par ***remplissageTable***.
- ✓ 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.

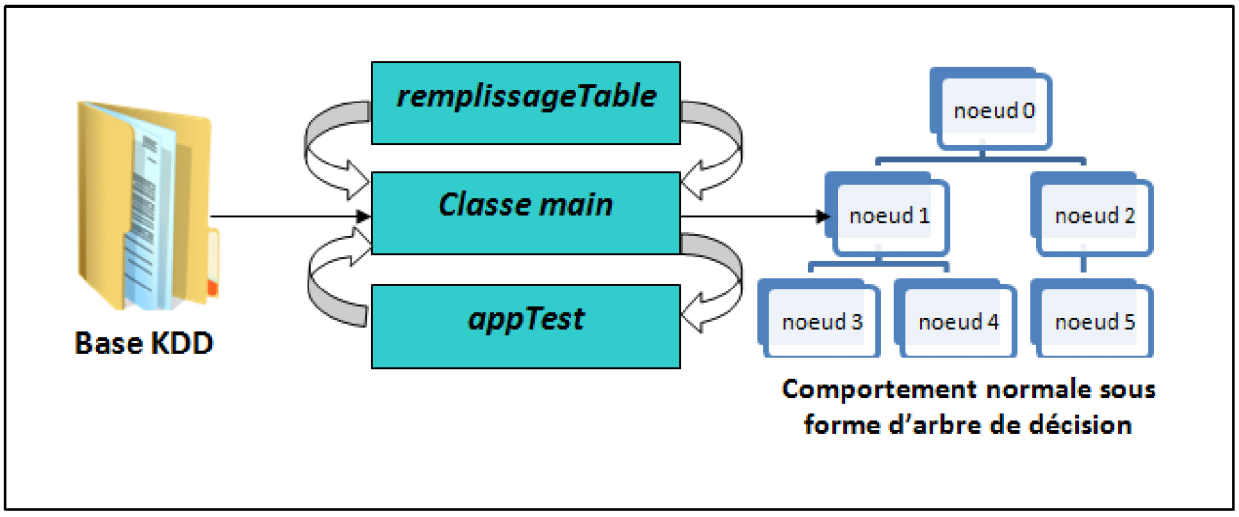

Figure IV.7 : Modélisation du comportement normale

2. Phase 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 caractérisée par 41 attributs. Lorsque ***CARTIDS*** 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'un de ses fils, notre IDS compare l'attribut 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 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.

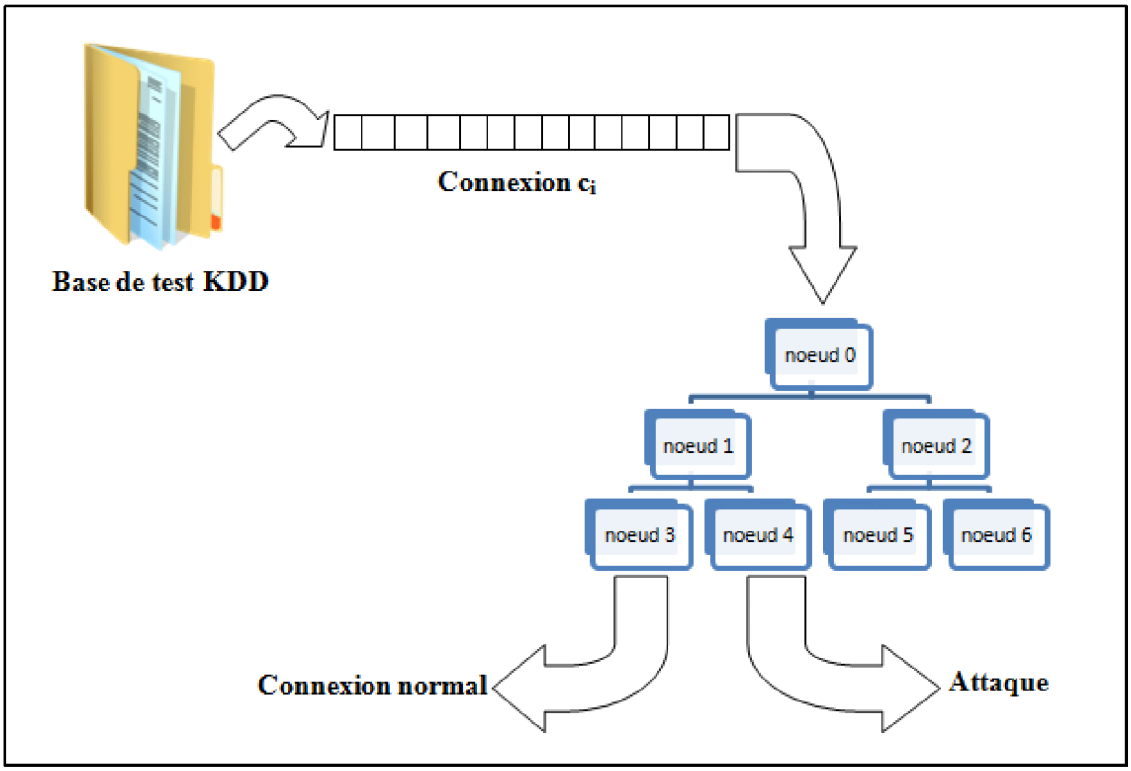

Figure IV.8 : Détection des attaques par IDSAC

IV.4 Conclusion

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

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

Conclusion générale

La sécurité informatique est clairement le plus grand challenge que les entreprises modernes utilisant les réseaux informatiques auront à relevé.

Au jour d'aujourd'hui, garantir qu'un réseau est parfaitement sécurisé relève, malheureusement, du domaine de l'impossible car il y aura toujours un pirate de génie quelque part au monde pour franchir toutes les barrières de sécurité misent en place.

Mais au moins, on peut toujours mettre des battons dans les roues à ces hackers en réduisant leurs marge de manœuvre et en rendant les intrusions plus difficiles en appliquant de nouvelles approches. Et c'est dans ce cadre que s'inscrit notre présent travail.

Dans ce mémoire, nous avons proposer un nouveau genre de systèmes de détection d'intrusions réseau (NIDS) comportementale baptisé **CARTIDS,** basé sur un arbre de décision et utilisant l'algorithme de classification CART.

Le système CARTIDS proposé vise à répondre aux exigences et aux objectifs de la sécurités réseau. CARTIDS opère en deux phase majeurs, une phase d'apprentissage durant laquelle un profil normal d'utilisation du système est généré en appliquant l'algorithme CART à la base d'apprentissage KDD, et une phase de test permettant de tester CARTIDS en utilisant la base de test KDD contenant des connections normales et des connections considérées comme étant des attaques (de type: Probing,R2U,URL, DOS).

Ce projet nous a été d'un grand apport pédagogique, puisqu'il nous a permis de découvrir la sécurité informatique, ses fonctionnalités et ses mécanismes, et de bien comprendre les arbre de décision ainsi que l'algorithme de classification non supervisé **CART.**

Perspectives:

Malheureusement, Il semble que quelles que soit les démarches suivies pour aborder un problème, on ne pourra atteindre complètement les objectifs initiaux, qui sont très variés et qui ne peuvent pas être tous satisfaits dans notre travail, C'est pourquoi, Nous comptons poursuivre ce travail de la manière suivante :

- Finaliser la mise en œuvre et l'implémentation de CARTIDS.
- La réalisation d'un prototype pour montrer l'efficacité du modèle du système de détection d'intrusions proposé.

- Réaliser et tester d'autres IDS en utilisant d'autres méthodes de classification basée sur des méthodes d'Intelligence Artificielle (algorithmes génétiques, etc.).
- Coupler le système ***CARTIDS*** avec un IDS de type scénario pour pouvoir détecter plus d'attaques.

UNIVERSITE MOULOUD MAMMERI DE TIZI OUZOU

BIBLIOGRAPHIE

BIBLIOGRAPHIE

[1] : Pascal Nicolas. « Cours de réseaux maîtrise d'informatique université d'Angers »

[2] : les réseaux informatique d'entreprise Pierre Erny, 1998

[3] : http://www.commentcamarche.net

[4] : http://www.netalya.com/fr/reseaux1.asp

[5] : www.protocols.com/pbook/h323.htm

[6] : www.protocols.com/pbook/h323.htm

[7] : La sécurité des réseaux Guillaume Desgeorge 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 du Havre UFR Sciences et Techniques).

[12] : La détection d'intrusion (Optimisation par classification)… RAHMANI amine et BOUMEDIEN Hassan.

[13] : Les systèmes de détection d'intrusion basés sur du 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 reference flow control for detecting intrusion symptoms at the OS level. RAID 2002.

[16] : Les systèmes de détection d'intrusions- David Burgermeister, Jonathan Krier- 22/07/2006- http://dbprog.developpez.com.

BIBLIOGRAPHIE

[17] : Architecture expérimentale pour la détection d'intrusions dans un système informatique……Philippe Biondi

[18] : Technique d'aprentissage thème 2 arbre de décision partie1 IFT 603

[19] : http://fr.wikiversity.org/wiki/Arbres_de_décision.

[20] : Une approche probabiliste pour le classement d'objets incomplétement connus dans un arbre de décision THESE présentée par Lamis HAWARAH Universitè Joseph Fourier.

[21] : Réseaux bayésiens naïfs et arbres de décision dans les systèmes de détection d'intrusions par : N.BenAmor, S.Benferhat et Z. Elouedi

[22] : Arbres de Décision par : Ricco RAKOTOMALALA Laboratoire ERIC Université Lumière Lyon 2

[23] : La detection d'intrusion [optimisation par classification] par : Rahmani amine et boumedien hacen

[24] : Yacine Bouzida, Frédéric Cuppens, Sylvain Gombault. « Détection de nouvelles attaques ».

[25] : http://kdd.ccs.uci.edu/databases/kddcup99/task.html/

[26] : http://www.securiteinfo.com

[27] : Cours de Sécurité Informatique par : Pierre-François Bonnefoi

[28] : Un petit guide pour la sécurité par : Alexandre Viardin ; http://www.mirabellug.org/

[29] : Arbres de décision Cours d'analyse de données Université Paris I

[30]: Introduction et Initiation à la sécurité informatique. « SecuriteInfo.com ».

[31]: G. Zémor.(2000) . « Cours de cryptographie ».

[32]: : Yacine Bouzida.(2006). « Application de l'analyse en composante principale pour la détection d'intrusion et détection de nouvelle attaques par apprentissage supervisé » Thèse de doctorat de l'Université de rennes. 2002.

BIBLIOGRAPHIE

[33]: http://www.tripwire.com/products/index.cfml

[34]: http://freshmeat.net/redir/swatch/10125/url_homepage/swatch

[35]: http://www.enterasys.com/ids/squire/

[36]: http://freshmeat.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 D´detection d'Intrusion Par: Mlle Dalila Boughaci Promotion 2004/2005.

[44]: http://www.bath.ac.uk/bucs/networking/connectfromhome/virtualprivatenetworkvpn/

[45]: http://www.labo-microsoft.org/articles/server/ISA2004/2/

[f46]: http://www.technoplus.fr/2010/09/04/le-poste-informatique-le-reseau-pedagogique/

[47]: http://www.ebook-cours.com/reseaux-communication-cour-gratuit.html

[48]: http://www.labo-microsoft.org/articles/itsec_2004/chiffrement/

[49]:F.3.http://igm.univ-mlv.fr/~dr/XPOSE2009/Sonde_de_securite_IDS_IPS/IDS.html

[50]: http://www.insecure.in/ids.asp

[51]: Protection des systemes informatiques contre les attaques par entrees-sorties par: Fernand Lone Sang, UNIVERSITE DE TOULOUSE 2012