N° D'ORDRE : 25/2012 – M / Inf.

République **A**lgérienne **D**émocratique et **P**opulaire

Ministère de **L**'enseignement **S**upérieur et de la **R**echerche **S**cientifique

Université des **S**ciences et de la **T**echnologie
<<**H**ouari **B**oumediene >>

Faculté d'Electronique et d'Informatique

MEMOIRE

Présentée pour l'obtention du diplôme de MAGISTERE
EN: INFORMATIQUE

Spécialité: Informatique Mobile

Par : GHOZLANE MOHAMED SADDEK

Sujet:

Conception et expérimentation d'une méthode de détection d'intrusions TCP/SYN Flooding à partir de fichiers LOG du Firewall

Soutenu publiquement, le 28/06/2012, devant le jury composé de :

1- M. H. AZZOUNE.	MCA à L'USTHB	: Président
2- M. A-R. BABA ALI.	MCA à L'USTHB	: Directeur de mémoire
3- Mme F. KHELLAF	MCA à L'USTHB	: Examinatrice
4- M. M. BENCHAIBA.	MCA à L'USTHB	: Examinateur
5- M. D. TANDJAOUI	MR AU CERIST	: Examinateur

Remerciements

Jamais je ne saurai présenter cette thèse sans adresser mes profonds respects et sincères remerciements à mon promoteur le docteur Monsieur Ahmed Riad Baba Ali pour m'avoir permis de bénéficier de son grand savoir, pour sa pédagogie, ses compétences, sa modestie et son aide précieuse tout au long de ce projet.

A tout les membres de l'équipe réseaux au niveau du centre national d'informatique et des statistiques de douanes algériennes Sans oublier mes responsables par la voie hiérarchique. Monsieur le directeur Houri Hocine, le sous directeur de l'informatique monsieur Saïd Abdiche et le chef de bureau du Service Réseau madame Kasdi.

Je remercie également le personnel du département informatique, particulièrement le responsable des affaires administratives du service poste graduation M^R Aissa.

Je profite aussi de cette occasion qui m'est offerte pour remercier l'équipe du notre laboratoire de recherche.
Je ne manquerai pas l'occasion de remercier très grandement ma famille (Ma femme Nadia, mes deux gausses iIlyasse et Souleimen, mes frères Mokhtar, Salah et Ahmed) car elle m'a toujours Manifesté son amour et ne cesse de me soutenir en partageant à tout moment avec moi mes sentiments.

Je termine ces remerciements en saluant vivement les membres du jury pour l'honneur qu'ils me font en acceptant de juger ce travail.

Table des matières

Liste des figures

Résumé

Il est évident aujourd'hui que le trafic Internet est de plus complexe et irrégulier, ce qui nuit grandement à un fonctionnement efficace des réseaux, ainsi qu'à la garantie de niveaux de performances. En particulier, le comportement du réseau est surtout mis à mal lorsque le trafic contient des anomalies importantes. Différentes raisons peuvent être à la source de ces anomalies, on peut citer notamment les attaques de déni de service (DoS). De fait, la détection des anomalies dans les réseaux et leurs trafics est devenue un des sujets de recherche les plus actifs du moment. L'objectif de cette thèse a donc été de développer des méthodes pour la détection des anomalies de type TCP/ SYN Flooding du trafic réseau. La méthode proposée repose sur la recherche des déviations significatives dans les statistiques du trafic par rapport à un trafic normal. La thèse a ainsi conduit au test de deux algorithmes appelés NADA (Network Anomaly Detection Algorithm) et CUSUM (la somme cumulative) **[11, 15],** pour la détection des attaques de type syn flood que nous avons appliqué à des données produites par le trafic internet**.** Une contribution importante de la thèse a trait à la méthode de validation et d'évaluation utilisée pour les deux algorithmes. La validation des deux algorithmes sur l'axe de la détection d'anomalies de type Syn Flood a ainsi été faite sur un fichier log généré par le firewall placé à l'entrée d'un réseau LAN connecté à internet. Les résultats obtenus sont concluants, notamment lorsqu'ils sont comparés avec ceux obtenus par d'autres outils de détection d'anomalies. De plus, la qualité des résultats est indépendante du type de trafic analysé.

Abstract

It is obvious today that traffic Internet is of more complex and irregular, which harms largely an effective operation of the networks, like with the guarantee of performance levels and quality of service (QdS). In particular, the behavior of the network is especially put at evil when the traffic contains important anomalies. Various reasons can be with the source of these anomalies, one can quote in particular the attacks of denial-of-service (Back). In fact, the detection of the anomalies in the networks and their traffics became one of the most active subjects of research of the moment. The objective of this thesis thus was to develop methods for detection of the anomalies of the type TCP/Syn Flooding of the traffic network. The method suggested rests on the research of the significant deviations in the statistics of the traffic compared to a normal traffic. The thesis thus led to the test of two algorithms called NADA (Network Anomaly Detection Algorithm) and CUSUM (the cumulative sum) **[11, 15**], for the detection of the attacks of the type TCP/SYN Flooding which we applied to data produced by traffic Internet. An important contribution of the thesis milked with the method of validation and evaluation used for the two algorithms. The validation of the two algorithms on the axis of the detection of anomalies of the type Syn Flood was thus made on a file log generated by the firewall placed at

the entry of a network LAN connected to Internet. The results obtained are conclusive, in particular when they are compared with those obtained by other tools for detection of anomalies. Moreover, the quality of the results is independent of the type of analyzed traffic.

Introduction générale

I. Introduction

I.1 Contexte

Les intrusions dans un réseau peuvent avoir des effets nuisibles pour les entreprises qui utilisent l'outil informatique. L'un des protocoles réseau les plus utilisés est le TCP/IP, mais il n'est malheureusement pas sécurisé. C'est pour cela que l'on cherche à mettre en place des systèmes de détection d'intrusions. Le terme «intrusion » fait ici référence à des activités anormales ou suspectes et à des attaques sur un équipement du réseau. On peut trouver différents types d'attaques, une des plus répandus sur Internet étant le **TCP/ SYN Flooding.**

Il s'agit d'une attaque réseau par saturation exploitant la faiblesse intrinsèque du protocole TCP et consistant en l'envoi massif de demandes de synchronisation (envoi de paquets SYN). Plusieurs études se sont intéressées au problème de la détection d'attaques de type TCP/ SYN Flooding. On peut distinguer notamment des travaux présentant des méthodes pour réduire la taille des données utilisant des agrégations aléatoires et des algorithmes de détection de ruptures tels que le CUSUM et NADA. Ici, nous étudions ces deux algorithmes pour détecter des attaques de type TCP/SYN Flooding que nous avons appliqué à des données produites par le réseau internet. Le trafic en un point du réseau y est représenté sous la forme d'une succession de paquets transmis.

Une attaque informatique peut se faire par un individu ou un groupe de personnes, contre l'ordinateur d'un individu ou d'un groupe de personnes morales ou physiques. Une attaque exploite une faille du système d'exploitation qui peut être due à un manque de budget, de temps d'installation, de personnes qualifiées, de politique de sécurité et de protections efficaces sur le marché. Néanmoins, de nos jours, le coût d'une attaque et de sa réparation peut être très élevé. C'est pourquoi les entreprises et l'ensemble des organisations s'intéressent de près à la sécurité informatique.

I.2 Motivations et objectifs

Dans le cadre de cette thèse, notre objectif est de détecter la présence d'une attaque de type « TCP/ SYN Flooding ». Pour cela, une démarche naturelle consiste à suivre l'évolution temporelle du trafic à destination d'un réseau connecté à au réseau internet. Généralement les intrusions font partie intégrante du trafic internet. Il est donc important mais aussi difficile de les détecter.

La diversité des disfonctionnements du réseau a motivé la conception et l'expérimentation de méthodes de détection d'intrusions à partir des fichiers représentants la trace d'événements et le développement d'algorithmes de détection d'intrusion.

Dans notre thèse nous présentons l'application de deux algorithmes (Cusum et Nada) appliqués sur un fichier log généré par le firewall placé à l'entrée d'un réseau LAN connecté à l'internet

pour la détection d'intrusions de type TCP/ SYNS Flooding.

I.3 Organisation de la thèse :

La présente thèse est composée de cinq chapitres organisés comme suit :

Le premier chapitre:

Explique les différentes méthodes de détection d'intrusions et expose les étapes suivies pour produire une attaque ainsi que les différents types d'attaques.

Le second chapitre:

Décrit les fichiers logs utilisés et il décrit aussi le dispositif qui les produit.

Le troisième chapitre:

IL est consacré à détailler l'approche de détection d'intrusions comportementale ainsi les algorithmes utilisés dans cette approche.

Le quatrième chapitre

Décrit l'implémentation de notre système de détection de l'attaque TCP/SYN Flooding et présente les résultats obtenus;

Enfin nous terminons cette thèse par une conclusion générale suivie par des annexes.

Chapitre I

La détection d'intrusions

I.1 Détection d'intrusions

La détection d'intrusions a été introduite en 1980 par J.P. Anderson qui a été le premier à montrer l'importance de l'audit de sécurité **[02]** dans le but de détecter les éventuelles violations de la politique de sécurité d'un système. Anderson définit une intrusion comme une violation de la politique de sécurité du système, c'est-à-dire une violation d'une des propriétés de confidentialité, d'intégrité ou de disponibilité du système. Nous différencions les notions d'attaques et d'intrusions : une intrusion est une violation de la politique de sécurité alors qu'une attaque est une tentative (effective ou non) de violer la politique de sécurité du système. Le but de la détection d'intrusions est de signaler les intrusions ou les attaques, à l'administrateur de sécurité pour que celui-ci puisse prendre les mesures de réactions adéquates.

I.2 Méthodes de détection d'intrusions:

Les méthodes de détection d'intrusions reposent sur l'observation d'un certain nombre d'événements et sur l'analyse de ceux-ci. Il s'agit premièrement de collecter les informations que l'on souhaite analyser. Ces informations proviennent des fichiers journaux du système, d'applications spécifiques (tels que des serveurs web, serveurs ftp, serveurs de courriers électroniques, routeurs, firewalls, serveurs d'authentification, etc) ou de sondes mises en place par les outils de détection d'intrusions tels des « sniffers » réseau, des modules spécifiques à certaines applications ou au système d'exploitation. Les données récoltées doivent ensuite être analysées pour y rechercher des traces d'intrusions. Cette analyse peut se faire de plusieurs manières : après les faits, de manière quasi temps réel ou bien en temps réel. Deux approches principales de détection d'intrusions que nous décrirons par la suite ont été proposées **[02]**: l'approche par signature (misuse detection) et l'approche comportementale (anomaly detection).Ces approches présentent différents avantages et inconvénients ; il a été proposé relativement rapidement dans la recherche en détection d'intrusions de combiner ces deux approches pour augmenter leur taux de détection

I.2.1 L'approche par signature (misuse detection)

Dans ce cas, les configurations les caractérisant ont été apprises d'attaques déjà connues. Ces configurations apprises sont recherchées à travers les nouvelles données afin de trouver des intrusions faisant partie de types déjà connus. Les deux principaux avantages de solutions suivant l'approche par signature sont la pertinence des détections et une certaine facilité de mise en place du système de détection. Cependant, cette approche présente certaines limites dont la plus gênante est de ne pouvoir détecter que les attaques connues. Parmi les méthodes qui utilisent cette approche :

a. Recherche de motifs (pattern matching)

La méthode la plus connue et la plus facile à comprendre. Elle se base sur la recherche de motifs (chaînes de caractères ou suite d'octets) au sein du flux de données. Le système de detection 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 reconnaître les paquets suspects. Le principal inconvénient de cette méthode est que seules les attaques reconnues par les signatures seront détectées. Il est donc nécessaire de mettre à jour régulièrement la base de signatures. Un autre inconvénient est que les motifs sont en général fixes. Or une attaque n'est pas toujours identique à 100%. Le moindre octet différent par rapport à la signature provoquera la non détection de l'attaque. Cette technique est également utilisée dans les anti-virus.

b. Recherche de motifs dynamiques

Le principe de cette méthode est le même que précédemment mais les signatures des attaques évoluent dynamiquement. Le système de detection est de ce fait doté de fonctionnalités d'adaptation et d'apprentissage.

c. Analyse de protocoles

Cette méthode se base sur une vérification de la conformité des flux, ainsi que sur l'observation des champs et paramètres suspects dans les paquets. L'analyse protocolaire est souvent implémentée par un ensemble de préprocesseurs, où chaque préprocesseur est chargé d'analyser un protocole particulier (FTP, HTTP, ICMP, ...). L'intérêt fort de l'analyse protocolaire est qu'elle permet de détecter des attaques inconnues, contrairement au pattern matching qui doit connaître l'attaque pour pouvoir la détecter.

d. Analyse heuristique et détection d'anomalies

Le but de cette méthode est de détecter une activité suspecte ou toute autre anomalie. Par exemple : une analyse heuristique permet de générer une alarme quand le nombre de sessions à destination d'un port donné dépasse un seuil dans un intervalle de temps prédéfini.

I.2.2. L'approche comportementale (Anomaly Detection)

Cette technique consiste à détecter une intrusion en fonction du comportement passé de l'utilisateur. Pour cela, il faut préalablement dresser un profil utilisateur à partir de ses habitudes et déclencher une alerte lorsque des événements hors profil se produisent. Cette technique peut être appliquée non seulement à des utilisateurs mais aussi à des applications et services. Plusieurs métriques sont possibles : la charge CPU, le volume de données échangées, le temps de connexion sur des ressources, la répartition statistique des protocoles et applications utilisés, les heures de connexion, etc. Cependant elle possède quelques inconvénients :

- peu fiable : tout changement dans les habitudes de l'utilisateur provoque une alerte.

- l'établissement du profil doit être souple afin qu'il n'y ait pas trop de fausses alertes : le pirate peut discrètement intervenir pour modifier le profil de l'utilisateur afin d'obtenir après plusieurs jours ou semaines, un profil qui lui permettra de mettre en place son attaque sans qu'elle ne soit détecté.

Méthode	Avantages	Inconvénients
Par scénario	#facile a programmer et implémenter # peu de possibilité d'erreurs	# Adaptée aux attaques connues uniquement # Le motif doit être de bonne qualité Consommation du CPU si le nombre de signature est très élevé
Comportementale	# Permet de détecter des attaques inconnues	#Performances moins bonnes que par scénario (pattern) du fait des processeurs qui doivent effectuer des Tâches complexes

Fig.1 - Tableau récapitulatif des méthodes de détections d'intrusions

I.3 Les systèmes de détection d'intrusions

Les systèmes, permettant de détecter ou bloquer des intrusions, peuvent être classés principalement dans trois familles de produits :

- Les anti-virus ;
- Les firewalls ;
- Les IDS / IPS (HIDS, NIDS)

Si les deux premières catégories sont bien connues et devenues en quelques années des produits de " commodité ", le fonctionnement, l'utilisation et les limitations des IDS et IPS sont moins connus. Il convient avant tout de faire la distinction entre ces 2 types de produits :

- Un système de détection des intrusions (IDS) est un ensemble de composants logiciels et/ou matériels **[09]**, dont la fonction principale est :
  - de détecter et analyser toute tentative d'effraction volontaire ou non et/ou de maintien dans un système d'informations,
  - de détecter et analyser toute altération éventuelle de ces données.
- Un système de prévention des intrusions (IPS) est un ensemble de composants logiciels et/ou matériels dont la fonction principale est d'empêcher toute activité suspecte.

I.3.1 Systèmes de détection de type "hôte" (HIDS)

Définition de HIDS:

Un HIDS est un agent logiciel que l'on installe généralement sur la machine à protéger et qui analyse en temps réels les flux relatifs à cette machine ainsi que les journaux.

I.3.2. Systèmes de détection de type "réseaux" (NIDS)

Les sondes de détection des intrusions réseaux sont les systèmes les plus connus et les plus utilisés à ce jour, notamment dans le monde de l'entreprise.

Définition de NIDS :

Un NIDS est une sonde qui a pour objectifs d'analyser de manière passive le flux transit sur le réseau et de détecter des intrusions en temps réel.

Les NIDS implémentent les techniques énumérées ci-dessous :

- Le pattern matching ; qui consiste à récupérer des chaînes de caractères bien identifiés liées à des séquences d'attaque. Fonctionnement que l'on peut assimiler aux systèmes antivirus. Cette méthode est relativement fiable mais cela nécessite que les attaques aient été identifiées et codées sous forme de signature préalablement.
- L'analyse de protocoles ; qui consiste à analyser la structure des paquets et l'utilisation de certains paramètres non conformes aux normes officielles (comme les RFC). Cette méthode ne nécessite pas de signature comme la précédente mais elle a tendance à déclencher un peu plus de fausses alertes que le " pattern matching ".
- La détection d'anomalies ; consiste à détecter toute déviation par rapport à un modèle qui correspondrait à un comportement normal (exemple : nombre élevé de tentatives de connexions sur un serveur peut indiquer une attaque, ou une connexion FTP avec un mot de passe de 200 caractères peut indiquer une tentative de débordement de mémoire tampon).

I.3.3 Le déploiement d'un NIDS

Le flux TCP représente plusieurs failles de sécurités c'est pour cela on peut dire que ce flux est un flux porteurs d'intrusions. Pour réduire ces intrusions il faut appeler à d'autres outils de détection d'intrusions : C'est les IDS **[14]** (systèmes de détection d'intrusions**)** le schéma ci-dessous explique les trois manières de placer in IDS pour bien surveiller le trafic entrant à un réseau d'entreprise.

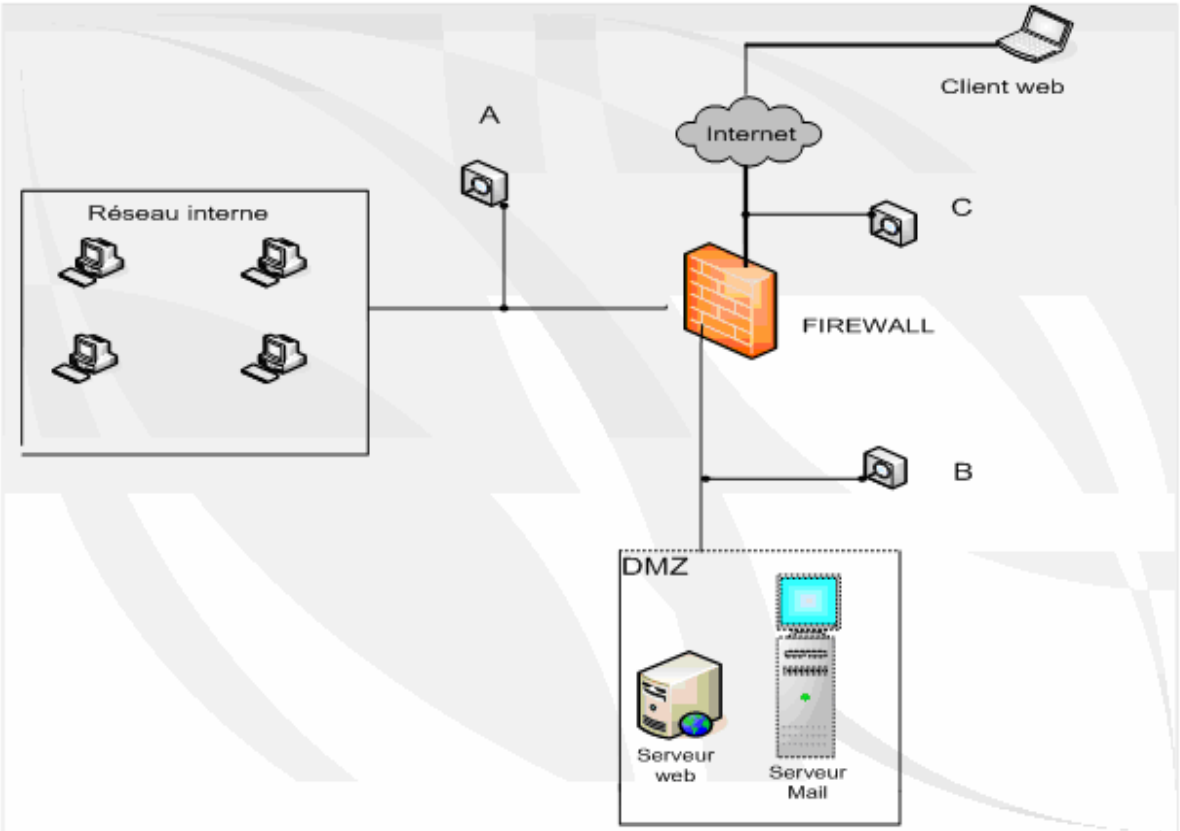

Fig.2 – Déploiement d'un NIDS

A l'emplacement **B**, seul le trafic entre les systèmes de la DMZ (Une zone démilitarisée) et Internet est analysé. Le trafic entre le réseau interne et Internet n'est pas analysé. Pour cela, il faudra également placer un NIDS au point **A**

A l'emplacement C, le trafic entre Internet et le réseau interne ou la DMZ est analysé. Par contre, le trafic entre le réseau interne et la DMZ est invisible

I.4 Les attaques

Une attaque est l'exploitation d'une faille d'un système informatique connecté à un réseau. Pour réussir leur exploit, les attaquants tentent d'appliquer un plan d'attaque bien précis pour aboutir à des objectifs distincts.

I.4.1 Les différentes étapes d'une attaque :

La plupart des attaques, de la plus simple à la plus complexe fonctionnent suivant les étapes suivantes :

- **Identification de la cible :** cette étape est indispensable à toute attaque organisée, elle permet de récolter un maximum de renseignements sur la cible en utilisant des informations publiques et sans engager d'actions hostiles. On peut citer par exemple l'interrogation des serveurs DNS…
- **Scanning :** l'objectif est de compléter les informations réunies sur une cible visées. Il est ainsi possible d'obtenir les adresses IP utilisées, les services accessibles de même qu'un

grand nombre d'informations de topologie détaillée (OS, versions des services, règles de pare-feu…).

- **Exploitation :** cette étape permet à partir des informations recueillies d'exploiter les failles identifiées sur les éléments de la cible, que ce soit au niveau protocolaire, des services et applications ou des systèmes d'exploitation présents sur le réseau.
- **Progression :** Le but ultime étant d'obtenir les droits de l'utilisateur root (ou system) sur un système afin de pouvoir y faire tout ce qu'il souhaite (inspection de la machine, récupération d'informations, nettoyage des traces...).

I.4.2 Les principales attaques provenant du réseau

I.4.2.1 Les attaques sur les protocoles IP, TCP et UDP

Les attaques sur les protocoles TCP/IP tentent d'exploiter les concepts de base des couches réseau et transport. Elles ont pour but de bloquer le fonctionnement d'un réseau (déni de service) ou d'usurper l'identité d'une machine.

I.4.2.1.1 Le déni de service

Parmi les attaques les plus connues, on trouve les attaques de type flooding. Les principales sont l'UDP flooding, l'ICMP flooding et le TCP SYN flooding. L'objectif de ces attaques est d'envoyer un maximum de paquets à la victime afin de réduire fortement la QoS de la victime ou du réseau de la victime.

Le TCP-SYN Flooding:

- ❖ **Rappel sur le TCP/IP:**

Transmission Control Protocol (littéralement, « protocole de contrôle de transmissions ») abrégé **TCP**, est un protocole de transport fiable, en mode connecté, documenté dans la RFC 793 de l'IETF.

Dans le modèle TCP/IP, TCP est situé au niveau de la couche de transport (entre la couche de réseau et la couche session). Les applications transmettent des flux de données sur une connexion réseau, et TCP découpe le flux d'octets en segments, dont la taille dépend de la MTU du réseau sous-jacent **(couche liaison de données).**

TCP a été développé en 1973, puis adopté pour Arpanet en 1976 par le DARPA.

Fonctionnement

Une session TCP fonctionne en trois phases :

- L'établissement de la connexion ;
- Les transferts de données ;
- La fin de la connexion.

L'établissement de la connexion se fait par une poignée de main en trois temps (handshaking). La rupture de connexion, elle, utilise une poignée de main en quatre temps. Pendant la phase d'établissement de la connexion, des paramètres comme le numéro de séquence sont initialisés afin d'assurer la transmission fiable (sans perte et dans l'ordre) des données.

Structure d'un segment TCP

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="16">Port Source 2 octets</td><td colspan="16">Port destination 2 octets</td></tr>
<tr><td colspan="32">Numéro de séquence</td></tr>
<tr><td colspan="32">Numéro d'acquittement</td></tr>
<tr><td colspan="4">Taille de l'en-tête</td><td colspan="3">réservé</td><td colspan="2">ECN</td><td colspan="2">URG</td><td colspan="2">ACK</td><td colspan="2">PSH</td><td colspan="2">RST</td><td colspan="2">SYN</td><td colspan="2">FIN</td><td colspan="10">Fenêtre</td></tr>
<tr><td colspan="24">Somme de contrôle</td><td colspan="8">Pointeur de données urgentes</td></tr>
<tr><td colspan="24">Options</td><td colspan="8">Remplissage</td></tr>
<tr><td colspan="32">Données</td></tr>
</table>

Signification des champs:

- Port source : Numéro du port source
- Port destination : Numéro du port destination
- Numéro de séquence : Numéro de séquence du premier octet de ce segment
- Numéro d'acquittement : Numéro de séquence du prochain octet attendu
- Taille de l'en-tête : Longueur de l'en-tête en mots de 32 bits (les options font partie de l'en-tête)
- Réservé : Réservé pour un usage futur
- ECN : signale la présence de congestion,
- Drapeaux
    - URG : Signale la présence de données URGentes
    - ACK : Signale que le paquet est un accusé de réception (ACKnowledgement)
    - PSH : Données à envoyer tout de suite (PuSH)
    - RST : Rupture anormale de la connexion (ReSeT)
    - SYN : Demande de synchronisation (SYN) ou établissement de connexion
    - FIN : Demande la FIN de la connexion

- Fenêtre : Taille de fenêtre demandée, c'est-à-dire le nombre d'octets que le récepteur souhaite recevoir sans accusé de réception
- Checksum : Somme de contrôle calculée sur l'ensemble de l'en-tête TCP et des données, mais aussi sur un pseudo en-tête (extrait de l'en-tête IP)
- Pointeur de données urgentes : Position relative des dernières données urgentes
- Options : Facultative
- Remplissage : Zéros ajoutés pour aligner les champs suivants du paquet sur 32 bits, si nécessaire
- Données : Séquences d'octets transmis par l'application.

❖ **L'attaque TCP/Syn Flooding**

Quand un client essaie d'établir une connexion TCP sur un serveur, le client et le serveur échangent une séquence de messages. Cette connexion technique s'applique à toutes les connexions TCP/IP (Telnet, web, e-mails...). Les abus viennent lorsque le serveur a envoyé un acquittement au client, et qu'il n'a pas reçu l'acquittement du client, la connexion est alors à semi-ouverte. Le serveur construit dans sa mémoire système une structure de données décrivant toutes les connexions courantes. Cette structure de données est de taille finie, ce qui veut dire qu'il peut se créer un dépassement de capacité en créant intentionnellement trop de connexions partiellement ouvertes.

Normalement, il y a un système de time-out associé à chaque connexion ouverte, donc les demi-connexions devraient disparaître et le serveur victime récupérer de la place libre dans sa mémoire pour d'autres connexions, mais le système agresseur peut envoyer des paquets plus vite que le temps nécessaire au serveur pour faire expirer les demi-connexions.

La localisation de l'attaque est complexe car les adresses contenues dans les paquets SYN envoyés sont falsifiées, on ne peut donc pas déterminer la véritable source. Internet faisant suivre les paquets grâce à l'adresse de destination, le seul moyen de s'affranchir de ces attaques est de valider la source d'un paquet en utilisant un filtrage.

UDP Flooding :

Ce déni de service exploite le mode non connecté du protocole UDP. Il crée un "UDP Packet Storm" (génération d'une grande quantité de paquets UDP) soit à destination d'une machine soit entre deux machines. Une telle attaque entre deux machines entraîne une congestion du réseau ainsi qu'une saturation des ressources des deux hôtes victimes. La congestion est plus importante du fait que le trafic UDP est prioritaire sur le trafic TCP. En effet, le protocole TCP possède un mécanisme de contrôle de congestion, dans le cas où l'acquittement d'un paquet arrive après un long délai, ce mécanisme adapte la fréquence d'émission des paquets TCP et le débit diminue.

Le protocole UDP ne possède pas ce mécanisme. Au bout d'un certain temps, le trafic UDP occupe donc toute la bande passante, ne laissant qu'une infime partie au trafic TCP.

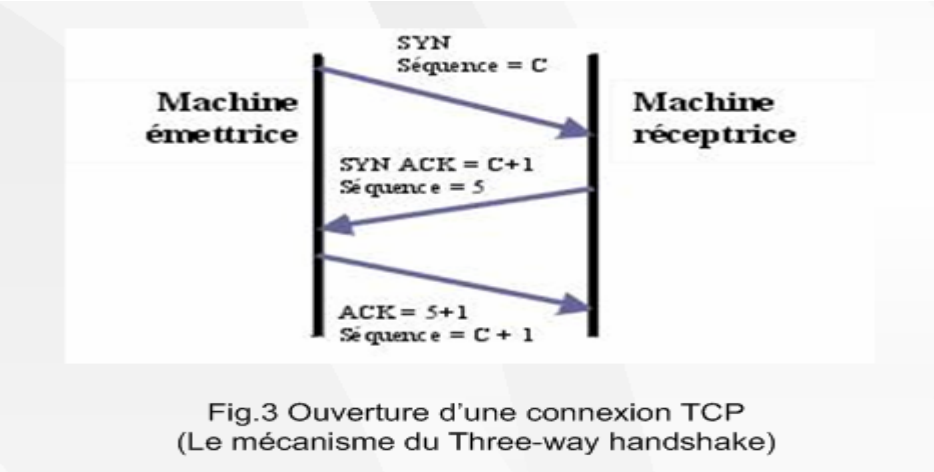

Fig.3 Ouverture d'une connexion TCP
(Le mécanisme du Three-way handshake)

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

− La machine attaquante envoie un ping à un ou plusieurs serveurs broadcast en falsifiant sa propre adresse IP (l'adresse à laquelle le serveur devrait théoriquement répondre par un pong) et en fournissant l'adresse IP de la machine cible.

− Lorsque le serveur broadcast va dispatcher le ping sur tout le réseau, toutes les machines du réseau vont répondre par un pong, que le serveur broadcast va rediriger vers la machine cible. Ainsi lorsque la machine attaquante adresse le ping à plusieurs serveurs broadcast situés sur des réseaux différents, l'ensemble des réponses de tous les ordinateurs des différents réseaux vont être re-routées sur la machine cible.

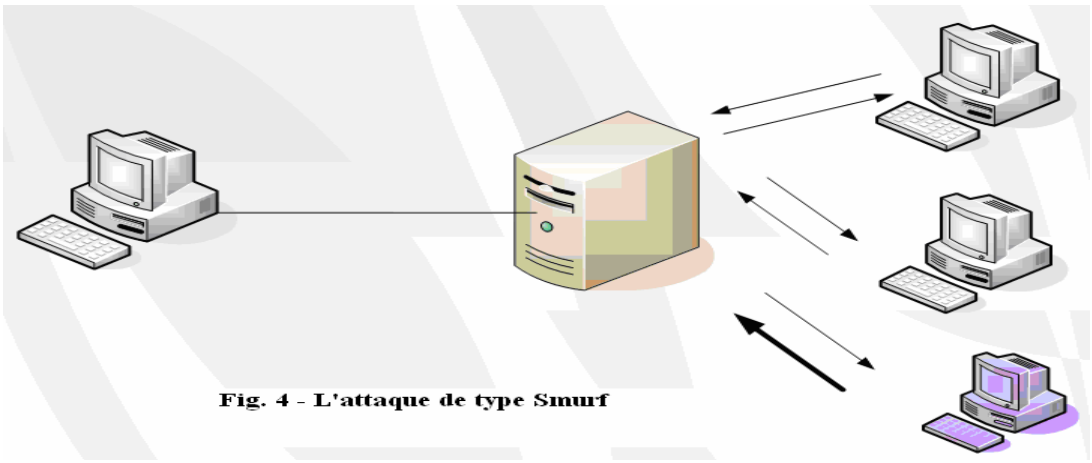

Fig. 4 - L'attaque de type Smurf

Attaque par fragmentation

Le protocole Internet autorise la fragmentation de paquets de trop grande taille pour adapter la

taille aux capacités de transfert du réseau. Seul le premier segment d'un paquet IP fragmenté contient le numéro de port TCP, les autres segments du paquet ne contenant pas de numéro de port TCP et ne peuvent pas être filtrés par le firewall, ainsi ils peuvent pénétrer un environnement à priori protégé.

Seule une bonne configuration du firewall peut permettre une meilleure sécurité, en interdisant par défaut tous les accès y compris ceux des fragmente et autorisant certains paquets selon des critères déterminés.

Les dénis de service de type Packet Fragment utilisent des faiblesses dans l'implémentation de certaines piles TCP/IP au niveau de la défragmentation IP (réassemblage des fragments IP). Une des attaques les plus connues utilisant ce principe est le Teardrop.

Fonctionnement :

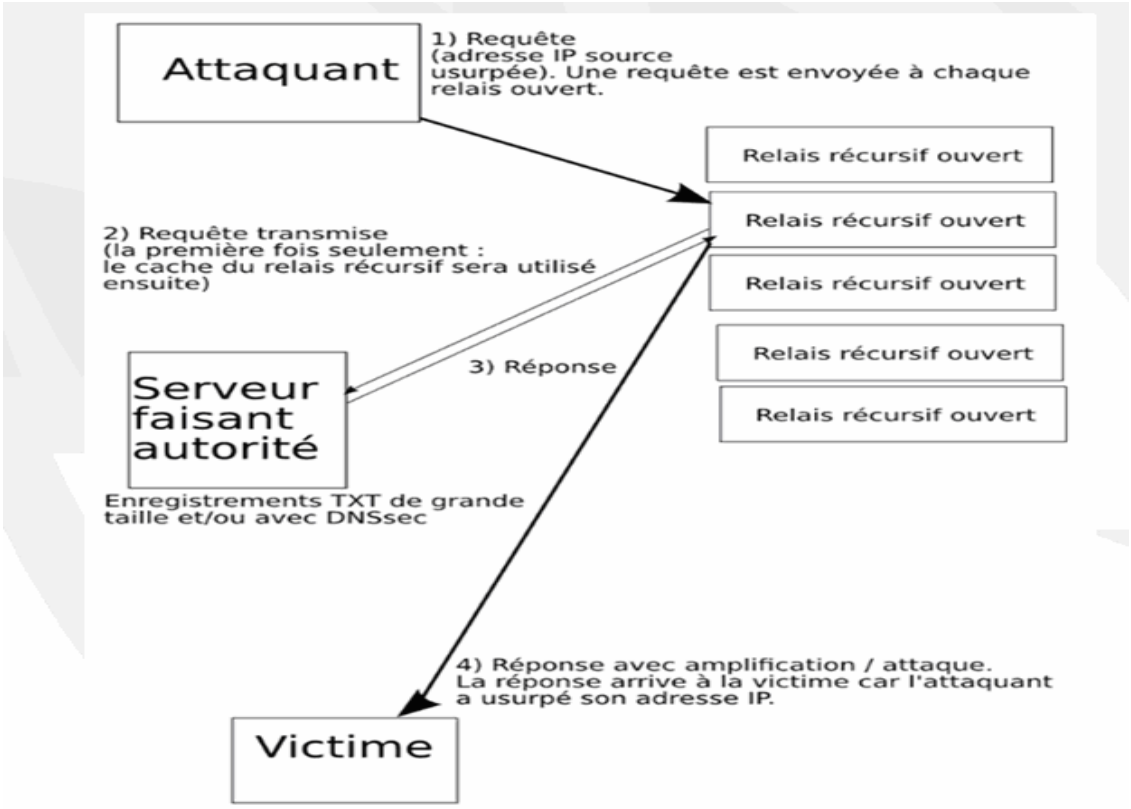

Fig. 5 - Schéma fonctionnel de l'attaque par fragmentation

Modifier les numéros de séquence afin de générer des blancs ou des recouvrements lors du réassemblage par la pile IP cible. Aujourd'hui, cette technique n'est plus viable du fait que les piles IP ont toutes évoluées.

Ping of death

L'attaque ping of Death est un problème important. Elle peut être reproduite de façon très rapide, et très simple depuis une machine distante. De plus, l'attaquant n'a besoin de rien connaitre, excepté l'adresse IP de sa cible.

Certains systèmes n'apprécient pas les ping avec des paquets de taille supérieure à 65536 octets. Un datagramme IP de plus de 65536 octets est illégal, mais peut être créé si l'on connait la façon dont les paquets sont fragmentés. Lorsque les paquets sont réassemblés par la cible en un paquet complet, la pile du buffer déborde. Certains systèmes se figent, redémarrent ou crashent.

I.4.2.1.2 L'usurpation d'identité

IP Spoofing: IP spoofing est une technique permettant à un hacker d'envoyer à une machine des paquets semblant provenir d'une adresse IP autre que celle de la machine du hacker. Le spoofing IP n'est pas pour autant un changement d'adresse IP. Plus exactement il s'agit d'une mascarade de l'adresse IP au niveau des paquets émis, c'est-à-dire que les paquets envoyés sont modifiés.

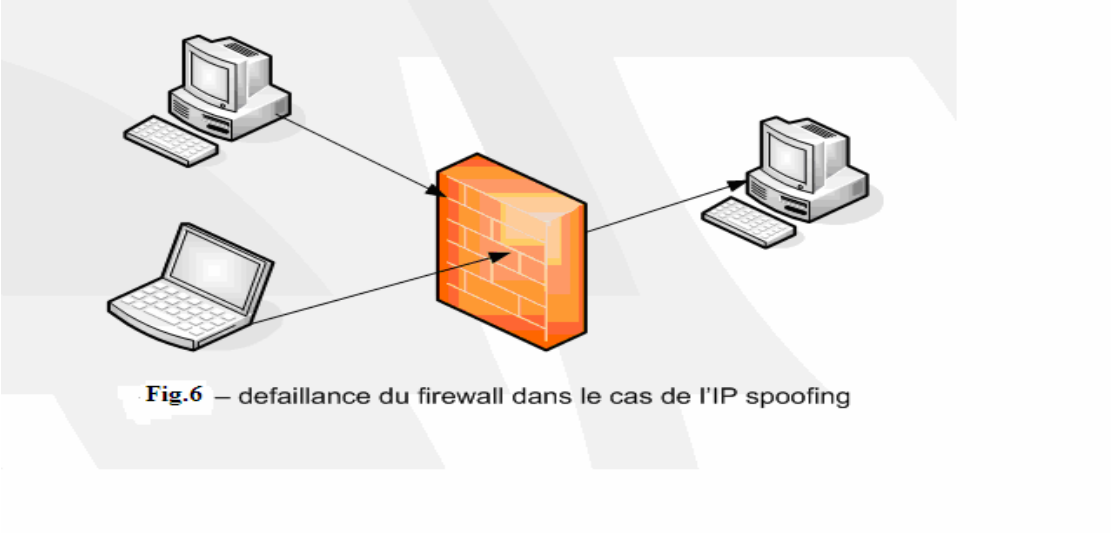

Fig.6 – defaillance du firewall dans le cas de l'IP spoofing

Comme l'indique la figure ci-dessus, la technique du spoofing peut permettre à un pirate de faire passer des paquets sur un réseau sans que ceux-ci ne soient interceptés par le système de filtrage de paquets. En effet, un Firewall fonctionne grâce à des règles de filtrage indiquant les adresses IP autorisées à communiquer avec les machines internes. Ainsi, un paquet spoofé avec l'adresse IP d'une machine interne semblera provenir du réseau interne et sera transmis à la machine cible, tandis qu'un paquet contenant une adresse IP externe sera automatiquement rejeté par le Firewall.

Inhiber la machine spoofée: Dans le cadre d'une attaque par spoofing, l'attaquant n'a aucune information car les réponses de la machine cible vont vers une autre machine du réseau, on parle alors d'attaque à l'aveugle ou "blind attack" en anglais.

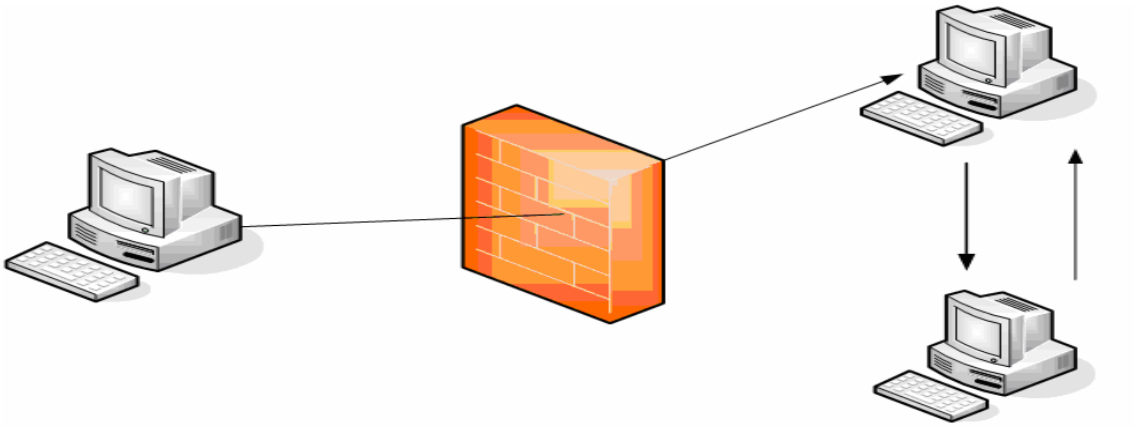

Fig.7 – inhibition de la machine spoofée

Prédire les numéros de séquence

Lorsque la machine spoofée est invalidée, la machine cible attend un paquet contenant l'accusé de réception et le bon numéro de séquence. Tout le travail du pirate consiste alors à deviner le numéro de séquence à renvoyer au serveur afin que la relation de confiance soit établie. Pour cela, les pirates utilisent généralement la source routing, c'est-à-dire qu'ils utilisent le champ option de l'en-tête IP afin d'indiquer une route de retour spécifique pour le paquet. Ainsi, grâce au sniffing, le pirate sera à même de lire le contenu des trames de retour. Ainsi, en connaissant le dernier numéro de séquence émis, le pirate établit des statistiques concernant son incrémentation et envoie des accusés de réception jusqu'à obtenir le bon numéro de séquence.

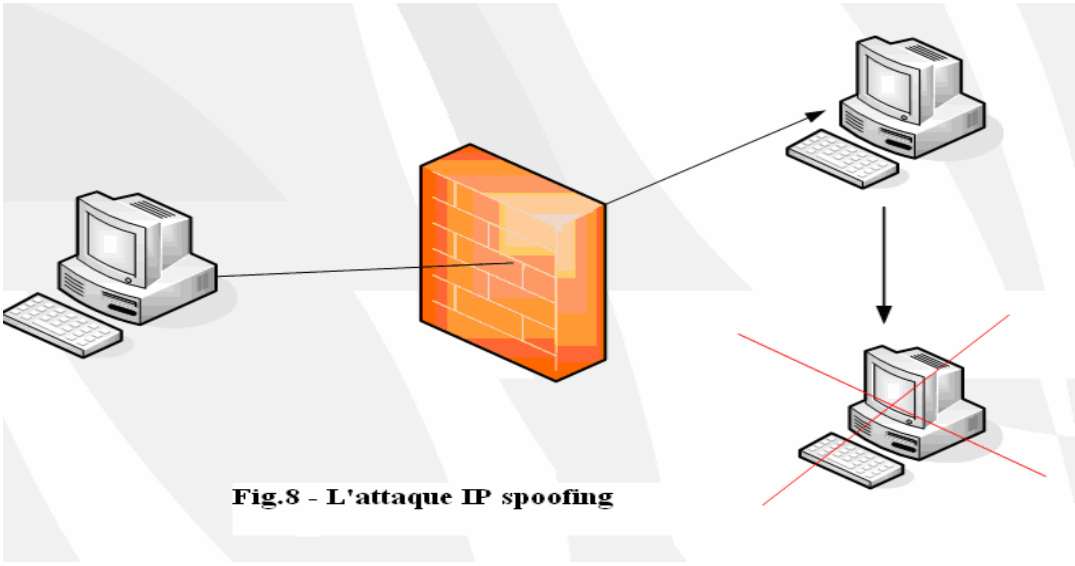

Fig.8 - L'attaque IP spoofing

I.4.2.2 Actuellement

La sécurité contre les attaques distantes se renforce, notamment par le biais d'équipements réseaux plus puissants (comme des firewalls plus intelligents), mais les attaques locales restent toutefois encore fort efficaces : l'IP Spoofing, le vol de session, … restent souvent possibles.

L'informatique évolue, les applications sont de plus en plus complexes et les délais laissés aux programmeurs et administrateurs sont souvent très courts. Les risques de failles applicatives sont,

de ce fait, très grands et peuvent s'avérer dangereux pour des applications largement répandues.

I.4.2.3 Les attaques sur les protocoles applicatifs

Ces attaques sont les plus nombreuses car elles sont parfois très simples à déployer. Elles ciblent les vulnérabilités des protocoles applicatifs (SMTP, HTTP, ...) ou de leurs implémentations mais également les applications qu'ils permettent d'utiliser.

On peut aussi classer ici les attaques sur les applications, notamment celles portant sur les navigateurs Web. L'attaquant peut, en fonction des attaques, exécuter du code malveillant, prendre le contrôle du système ou engendrer un déni de service. Les quatre exemples suivants mettent en évidence des failles exploitables du service d'annuaire de nommage de l'Internet (DNS), du protocole HTTP, du serveur HTTP Apache et du navigateur FireFox. Les deux derniers exemples ont été choisis car ils sont relatifs à des outils très utilisés dans le monde universitaire. Il est évident que d'autres outils encore plus utilisés par ailleurs sont encore plus vulnérables (Microsoft Internet Explorer notamment).

I.4.2.3.1 Attaque sur le DNS

Le DNS fonctionne en utilisant le protocole UDP (sans états) selon le schéma suivant :

- ✓ lorsqu'une application souhaite résoudre un nom, elle s'adresse à son serveur de noms local,
- ✓ celui-ci prend en charge la requête et s'adresse au serveur de noms responsable du domaine ciblé ;
- ✓ il attend la réponse et lorsque celle-ci arrive, vérifie qu'elle émane bien du serveur interrogé puis délivre la réponse (qu'il stocke également dans un cache).

Le mécanisme de vérification s'appuie sur un numéro d'identification de la question (Query ID) qui est tiré aléatoirement. L'attaque consiste ici à tenter de répondre à une question avant le serveur légitime. Il s'agit d'empoisonner les caches des serveurs DNS pour rediriger leurs clients vers des machines qui usurpent l'identité de serveurs. En cas de réussite, les exploitations possibles sont illimitées (diffusion de fausses versions de logiciels, récupération de données confidentielles, etc...).

I.4.2.3.2 Attaque sur HTTP

La méthode TRACE du protocole HTTP permet de visualiser dans le corps d'une réponse toutes les entêtes qui sont transmises lors d'une requête HTTP. Cette fonctionnalité est pratique dans une phase de mise au point pour vérifier ce que l'application distante reçoit. Malicieusement utilisée elle permet à un attaquant de visionner des cookies, des authentifications. Cette faille peut être exploitée de la manière suivante :

- l'attaquant poste un mail à destination de la victime

- la victime consulte ce mail via une interface de type WebMail ;
- le courrier contient une pièce jointe qui va faire exécuter un code JavaScript qui engendre une requête HTTP TRACE
- le script effectue une requête HTTP sur le serveur de l'attaquant en passant en paramètre le résultat de la requête TRACE effectuée sur le serveur de l'attaque ;
- en consultant son log d'httpd, l'attaquant visualise les cookies, les champs auth

I.5 Conclusion :

Actuellement énormément de machines et de services sont accessibles par Internet. Ce qui facilite grandement les échanges et le travail collaboratif facilite aussi les attaques de tous types. Pour se protéger de ces attaques, les antivirus et pare-feux sur les terminaux ne suffisent pas. C'est pourquoi nous nous intéressons à l'implémentation de la détection d'attaques à l'intérieur même du réseau, sur des liens à très haut débit, pour arrêter les attaques aussi tôt que possible.
Contrairement à la classification de services, la détection d'attaques peut dans certains cas se faire efficacement.

Chapitre II

Les fichiers Log

II.1 Présentation

Afin de protéger les systèmes d'éventuelles attaques, une approche courante est d'enregistrer des logs afin de surveiller toutes les activités importantes. Chaque fois qu'un événement méritant d'être noté se produit dans le système, une entrée sera ajoutée dans le fichier de log, que ce soit dans un format texte ou un format binaire. Par exemple, pour un site web, chaque requête de fichier sera enregistrée. Ces enregistrements permettent de savoir par la suite sur les utilisateurs, où ils sont sur le serveur et ce qu'ils font.

Mais le nombre d'information est beaucoup trop important pour qu'un être humain puisse vérifier ces logs, car il peut s'accumuler plusieurs giga-octets de données par jour sur certains systèmes. Une première méthode est de supprimer les données qui sont moins importante et ne conserver que les plus intéressantes. Mais cela nécessite une analyse de log pour la détection des activités anormales dans ces données en étant le moins dépendant possible des interventions humaines.

II.2. Activités liées à l'audit de sécurité

Toute opération entreprise sur un système informatique se traduit par une séquence d'actions effectuées par le système. Ces actions sont appelées activités système. Une activité système intervenant à un certain moment est appelée événement. Un journal d'audit de sécurité est un fichier dans lequel est enregistré chronologiquement tout ou une partie des événements. Les enregistrements du journal d'audit sont aussi appelés traces d'audit. Le journal d'audit doit permettre, à partir de l'étude des séquences d'événements que l'on y trouve, de reconstituer les opérations entreprises par certains utilisateurs spécifiques du système, dits utilisateurs audites. Il doit être possible de répondre aux six questions suivantes :

- Quelle opération a été faite ?
- Qui a fait l'opération ? Identifiant et label (niveau de sécurité et ensemble de domaine) du processus et de l'utilisateur pour le compte duquel il agit.
- Quelles ressources du système ont-t-elles été affectées ? Lorsqu'il s'agit d'un fichier, on enregistre son nom et son chemin d'accès, ainsi que les données ajoutées ou supprimées.
- Quand l'opération a-t-elle été réalisée (horodatage de l'enregistrement) ?
- où cela est-il arrive ? Dans le cas ou l'opération qui a conduit à l'enregistrement s'est produite sur un serveur distant, on enregistre l'identifiant de ce serveur.
- En cas d'échec, pourquoi l'opération a-t-elle échoué ?

Pour répondre à ces questions, l'administrateur doit spécifier les utilisateurs et les activités système à auditer, assurer la collecte des événements dans le fichier d'audit et analyser régulièrement ce fichier. De plus il essayera de réparer les dégâts éventuellement détectés.

II.2.1. Spécification des activités système à auditer

Nous donnons ici une liste non exhaustive, inspirée de **[06]**, des informations pertinentes à auditer. Selon le niveau de sécurité qu'il souhaite atteindre, l'officier de sécurité définira des événements auditables correspondant à certaines de ces informations

II.2.1.1. Informations sur les accès au système

Il s'agit d'obtenir des informations sur ce qui constituera le point de départ de toute investigation sur une violation éventuelle de la sécurité :

- qui a accédé au système (identifiant de l'utilisateur ou du processus),
- Quand (horodatage de l'accès au système),
- Où (identifiant du terminal, adresse),
- Comment (mode d'entrée : interactif, batch local, connexion distante).

II.2.1.2. Informations sur l'usage fait du système

Il s'agit de montrer quelles ressources du système ont été utilisées et comment :

- Commandes systèmes utilisées et leur résultat (échec ou succès),
- Accès à une unité d'entrée/sortie,
- Utilisation de la CPU,
- Taux d'occupation mémoire par utilisateur.

II.2.1.3. Informations sur l'usage fait des fichiers

C'est bien sûr un point très sensible puisque les fichiers contiennent l'information.

Pour chaque accès à un fichier, on s'intéressera aux points suivants :

- Horodatage de l'accès,
- Type de l'accès (ouverture, fermeture, lecture, ajout d'enregistrement(s), modification d'enregistrement(s), purge du fichier, ...),
- Source de l'accès (triplet (utilisateur, terminal, application)),
- Volume d'informations échangées lors de l'accès.

II.2.1.4. Informations relatives à chaque application

Chaque application conduit à des événements qui peuvent influer sur la sécurité du système. On pourra enregistrer les événements suivants

- Les lancements et arrêts d'application,
- Les modules réellement exécutés,
- Les commandes exécutées et leurs résultats (échec ou succès),
- Les données entrées,
- Les sorties produites.

II.2.1.5. Informations sur les violations éventuelles de la sécurité

Il s'agit d'enregistrer tous les événements pour lesquels il y a eu une tentative d'accès à une ressource du système sans que cet accès soit autorisé par les règles de la politique de sécurité. Parmi ces événements, on peut citer :

- La tentative d'exécution d'une application dans un mode privilégie (par exemple la modification des droits d'accès sous Unix),
- La tentative d'accès à un fichier non autorisé ou la fourniture d'un mot de passe erroné pour cet accès,
- La tentative d'utilisation de certaines commandes du système réservées à des utilisateurs privilégie,
- Le changement des droits d'accès à des fichiers sensibles, l'accès au système à des moments ou depuis des lieux inhabituels. Il faut définir ce qui est habituel pour chaque utilisateur, un groupe d'utilisateurs, ou tous les utilisateurs. Les informations de ce type doivent être exploitées rapidement de manière à limiter les effets éventuels d'une atteinte à la politique de sécurité.

Même si on a une exploitation en quasi-temps réel, on gardera une trace de ces informations pour en permettre une étude ultérieure.

II.2.1.6. Informations statistiques sur le système

Il est possible de repérer des activités anormales dans un système en observant attentivement quelques facteurs clé. Par exemple, on pourra tirer des conclusions des informations suivantes:

- Niveau anormalement élevé des refus d'accès au système,
- Niveau anormalement élevé ou bas de l'usage de certaines commandes du système (en particulier les commandes demandant des privilèges particuliers).

II.2.2. Collecte des événements

La plupart des systèmes d'exploitation disposent d'un sous-système d'audit capable de générer certains types d'événement.

Le noyau du système assure alors la génération et la collecte de ces événements. Il est également possible de générer des événements au sein des applications. Dans ce cas, on doit fournir au développeur une ((boîte à outils sécurité)) lui offrant les primitives de génération et de collecte adéquates. Le code source des applications devra être examiné pour vérifier que l'audit est bien réalisé.

II.2.3. Analyse du journal d'audit

Le but de l'analyse est de révéler toute violation des règles de la politique de sécurité de manière à identifier les responsables, identifier les dégâts éventuels (et les réparer si possible) et proposer des changements dans les éléments qui assurent la sécurité du système pour prévenir des attaques similaires. L'analyse doit permettre de détecter toutes sortes de transgressions, depuis l'intrusion

dans le système par un utilisateur non autorise´ jusqu'aux abus de toutes formes imputables aux utilisateurs connus du système. Cependant, l'audit de sécurité se fera parfois dans un ou plusieurs objectifs précis. Ainsi, on parle parfois d'audit des canaux cachés (détection de fuite par canal caché), d'audit des gains de privilèges ou d'audit des accès au système et des accès aux données (détection d'intrusion). Nous nous intéressons ici à ce dernier cas.

L'analyse du journal d'audit se fera donc dans l'objectif de découvrir des comportements des types suivants :

- Intrusions portant atteinte à la confidentialité :
  - ✓ vol de données : lecture, copie ou prise (copie suivie d'une destruction),
  - ✓ furetage : parcours des répertoires à la recherche d'informations dont on ne précisément pas forcément l'existence,
  - ✓ fuite d'informations par canal caché,
  - ✓ inférence illégitime: corrélation illégitime entre des données pour lesquelles on a des droits d'accès.
- Intrusions portant atteinte à l'intégrité :
  - ✓ fraude : modification illégitime de fichiers de données,
  - ✓ introduction de programmes malveillants.
- Intrusions portant atteinte à la disponibilité de service :
  - ✓ destruction illégitime ou abusive de fichiers (données ou programmes),
  - ✓ occupation illégitime ou abusive de ressources avec ou sans bénéfice d'usage
  - ✓ réduction illicite ou abusive des droits des usagers.

De plus, les programmes malveillants (virus, cheval de Troie, ver, bombe logique) peuvent aussi bien porter atteinte à la confidentialité qu'à l'intégrité ou à la disponibilité de service. Les actions qui peuvent être entreprises si on détecté une intrusion en cours, sont la de connexion de l'intrus ou son confinement dans des répertoires particuliers, limitant les dommages possibles mais permettant d'étudier plus précisément son comportement.

II.2.3.1. Fréquence de l'analyse des traces d'audits

Les analyses doivent être fréquentes afin que le minimum de malversations reste indetecté. Les analyses journalières, semblent aujourd'hui insuffisantes. L'objectif à atteindre est la surveillance du système en quasi-temps réel. A ce titre, on peut envisager que l'écart entre deux analyses successives des traces d'audit soit réduit à quelques minutes, voire à quelques dizaines de secondes (ce doit être un élément de la politique de sécurité). Cette contrainte a des conséquences fortes sur les méthodes d'analyse utilisables, qui doivent permettre de traiter très rapidement des volumes de données considérables.

II.2.3.2. Protection du journal d'audit

La confidentialité, l'intégrité et la disponibilité du journal d'audit sont essentielles, un utilisateur malicieux ne devant pas pouvoir effacer ou modifier les traces de ses opérations. Le fichier d'audit doit donc être protégé contre toute lecture ou modification par des utilisateurs autres que ceux qui sont autorisés.

II.2.3.3. Le cas des réseaux

Lorsque l'on souhaite faire de l'audit sur un système distribué, il est impératif de disposer d'une base de temps commune qui permettra d'estampiller les événements surmenants sur toutes les machines connectées. Il faut donc construire un temps global sur le réseau ou accepter l'utilisation d'une fenêtre de temps**.**

Par ailleurs, il faut être capable de diffuser les types d'événement à auditer et de collecter les enregistrements d'audit en provenance des différentes machines connectées sur le réseau, de manière à constituer un journal d'audit global. Ces transferts d'information doivent être sécurisés, par exemple par un mécanisme de chiffrement.

II.3 Le rôle de l'analyse des logs dans la sécurité d'un réseau

La sécurité d'un réseau ne devrait ainsi jamais reposer intégralement sur un IDS, ni intégralement sur un Firewall.

Un réseau ne peut être sécurisé à 100%, mais une surveillance régulière des logs des systèmes de ce réseau, des alertes produites par son IDS, des logs produits par son Firewall, et une administration honnête de ce réseau (architecture, configuration des équipements réseau, etc.) et de ses systèmes (mise à jour régulière, configuration sécurisée, etc.) limitent néanmoins les risques.

Les objectifs de la lecture des logs réseau et système, et des alertes et des logs générés par un IDS, sont :

− La métrologie du réseau : contrôler le volume d'utilisation des ressources, détecter des anomalies afin de mettre en place une qualité de service, faire évoluer les équipements en fonction des besoins.

− Vérifier que les règles en matière de sécurité informatique sont correctement appliquées et que la sécurité des systèmes d'information et du réseau telle qu'elle a été définie par la politique de sécurité de l'unité est assurée.

− Détecter toute défaillance ou anomalie de sécurité, volontaire ou accidentelle, d'origine matérielle ou humaine.

− Détecter toute violation des règles de sécurité ou tout abus d'utilisation des moyens informatiques pouvant engager la responsabilité de l'organisation.

− Être capable de fournir des preuves nécessaires pour mener les enquêtes en cas d'incident

de sécurité et de répondre à toute réquisition officielle présentée dans les formes légales. Les objectifs précités imposent d'aller au-delà d'un simple enregistrement des données des fichiers journaux. Ils impliquent nécessairement l'enregistrement, la conservation temporaire et l'exploitation des données collectées, dans la mesure où des éléments contenus dans les fichiers journaux permettent de remonter à l'utilisateur et de générer des rapports statistiques permettant de contrôler l'efficacité de l'utilisation des ressources matérielles et logicielles de l'organisation.

Ainsi, on a mis en évidence l'importance des logs système et réseau, et la complexité de leur traitement. En effet, certains pirates créent de fausses attaques pour inonder l'IDS de logs avant de lancer leur véritable attaque. Si une méthode efficace d'analyse de logs n'est pas utilisée, une telle attaque peut passer inaperçue. En outre, si le pirate possède les moyens de modifier les logs des systèmes attaqués, tout le système de journalisation devient inutile.

II.4 Méthodologie d'analyse

Il est intéressant d'analyser de quelle manière les différentes attaques sont enregistrées dans les logs et quelle sont les corrélations que l'on peut y découvrir afin d'aider à identifier ces attaques. Deux approches complémentaires peuvent être utilisées **[07]** pour tracer ces attaques : l'approche top down et l'approche bottom-up.

a) L'approche top down (attaque - logs)

Dans cette approche nous partons d'une attaque connue et nous analysons l'évolution des traces contenues dans les logs. Cela permet généralement de dévoiler des comportements communs dans les classes d'attaques. Le principal avantage de cette méthode est que cela ne nécessite pas une analyse détaillée des fichiers log qui peuvent être extrêmement volumineux dans la plupart des réseaux informatiques. Il est possible d'effectuer cette analyse sur une très grande variété d'attaques. Il est possible ensuite de générer le tableau suivant pour les différentes classes d'attaque ce qui permet de savoir quel log analyser pour chaque type d'attaque.

ATTAQUE	LOG					
	Syslog	Netflow	Tcp	Dns	Web	Ftp
Dictionary	x	x	x		x	x
Ftp Write	x	x	x			x
Imap	x	x	x			
Namad	x	x		x		
Sedmail	x	x	x	x		
Back	x		x		x	
MailBomb	x	x	x			
Syn Flood	x	x	x	x		
PingOf Death		x	x			
Smurf	x	x	x			
Udp Storm		x	x	x		

Fig.9 – Tableau récapitulatif des classes d'attaques dans les fichiers log

b) L'approche bottom-up (logs - attaque)

Dans l'approche bottom-up, nous essayons de récolter des informations sérieuses A partir de multiples logs afin d'identifier une attaque spécifique. A partir d'éléments simples telle qu'un échec d'identification, un grand nombre de ping, il s'agit d'essayer de découvrir des attaques potentielles. Une fois qu'une anomalie est détectée à partir d'un fichier log, la prochaine étape consiste à examiner les autres logs dans la même période de temps.

II.5 la détection d'intrusion dans les fichiers logs

L'analyse des logs dans une perspective de détection d'intrusions consiste à découvrir ou à identifier l'utilisation d'un système informatique à d'autres fins que celles prévues. C'est une technique à multiples facettes, difficile à cerner lorsqu'on ne les manipule pas. La plupart des travaux effectués dans ce domaine restent difficiles à comparer. On peut rarement mettre deux modèles sur un pied d'égalité, et il est peu aisé de mettre à l'épreuve plusieurs modèles, ou encore d'en développer d'autres radicalement différents sans tout reconstruire. La détection d'intrusion sera placée parmi l'autre technique anti-intrusion.

II.5.1 Détection d'intrusions à base du trafic réseau d'un firewall

La détection d'intrusions en se basant sur la surveillance du trafic traversant le Firewall repose sur l'hypothèse que les journaux de logs générés par ces équipements constituent une source pour la prévoyance des traces de nouvelles attaques inconnues. Par ailleurs, ces logs peuvent aussi être utilisés en tant que sources d'événements pour la détection d'intrusions. Notre étude est conduite sur un Firewall de type Cisco série **Pix 501** qui est installé dans un réseau de test pour le filtrage du trafic et la journalisation des événements. Le Firewall définit un format de log bien spécifique mais les résultats de nos travaux peuvent être appliqués à d'autres types de Firewalls ayant les mêmes fonctionnalités de filtrage et de logging.

II.5.2 Le Firewall Cisco pix 501

II.5.2.1 Introduction : Pourquoi faire appel à Cisco en matière de sécurité?

Depuis son origine, Cisco System a pour objectif de permettre à ses clients de développer leur activité en s'appuyant sur des réseaux performants. Or un réseau non sécurisé n'offre pas toutes les garanties nécessaires à une entreprise pour le développement de son activité, pire il peut mettre en danger l'intégralité de l'entreprise. Assurer la sécurité des réseaux des entreprises, quelle que soit leur taille, est donc devenu un des objectifs majeurs de Cisco. Et qui mieux que le leader du marché des solutions réseaux peut garantir la sécurité, l'interopérabilité et la cohérence des réseaux ?

Fort de son expertise réseau, Cisco System a ainsi développé la gamme de solutions de sécurité et de réseaux privés virtuels (VPN) la plus complète du marché. Reconnu comme le leader du marché de la sécurité. Cisco System propose aujourd'hui à ses clients et partenaires de bénéficier des avantages suivants :

- Gamme de solutions : Cisco propose un large éventail de produits VPN et de sécurité pour répondre à la diversité des problématiques clients : firewall, système de détection d'intrusion, concentrateurs VPN et routeurs et ce, quelle que soit leur taille et leur configuration.
- Leadership et expertise du secteur : selon les experts de la sécurité réseau, les firewalls dédiés de la gamme Cisco PIX, occupent aujourd'hui la première place du marché mondial et, selon le cabinet Frost & Sullivan, il en est de même pour le système sécurisé de détection d'intrusion Cisco (IDS). De plus, les listes de contrôle d'accès Cisco (ACL) sont la technologie de sécurité la plus largement utilisée dans le monde.
- Interopérabilité garantie : Cisco garantit la compatibilité de tous ses produits VPN et de sécurité.

II.5.2.2 Présentation du Cisco Pix 501 :

II.5.2.2.1. Introduction:

L'internet est devenu un concept de communication incontournable, mais l'un des freins à sont émancipation à longtemps été la sécurité des informations circulant sur ce gigantesque réseau de réseaux. Cependant de nombreux progrès on été fait dans le domaine de la sécurité sur le net. Les pare-feu on fait leur apparition et des milliers d'utilisateurs particuliers ou professionnels s'en servent quotidiennement afin de sécuriser un minimum leur équipement réseaux. Il s'agit d'un équipement logiciel ou matériel servant d'interface entre différents réseaux permettant d'en limiter les accès **[08]:**

Fig.10 – schéma du firewall pix 501

II.5.2.2.2. Définition et utilité d'un firewall

Les conséquences d'un réseau mal sécurisé peuvent être terrible (perte d'information, surcharge inutile de la bande passante, problème de confidentialité de l'information). Les firewalls ont pour objectif d'y remédier en prévenant ce genre de problèmes en assurant une surveillance constante des flux dans les réseaux interconnectés. Pourquoi un firewall matériel plutôt qu'un firewall logiciel? La réponse réside autant en terme d'efficacité qu'en terme de consommation de

ressources. Les firewalls logiciels on en effet l'inconvénient de consommer énormément de ressources système, leur fonctionnement est basé sur un principe de filtrage de paquets IP, c'est-à-dire sur l'analyse des en-têtes des paquets IP échangés entre deux machines, alors que les pixs utilisent un système dédié de sécurisation en temps réel. Ceux ci sont donc plus rapides tout en possédant des fonctions supplémentaires tel l'administration à distance. Le pare-feu Cisco pix 501 offre une alternative simple à l'établissement d'un petit réseau. Il est en effet doté d'un commutateur à quatre ports permettant de connecter plusieurs machines tout en assurant la protection de celles-ci contre les intrusions. Il sera possible de cacher les véritables adresses réseaux de votre équipement, d'utiliser le serveur DHCP intégré afin d'attribuer dynamiquement des adresses ip.

II.5.2.2.3. Cadre d'utilisation

Les firewalls Cisco possède tout comme les routeurs de cette marque Un système d'exploitation propriétaire embarqué l'IOS qui permet de palier aux failles des systèmes non spécialisés. Les pix Cisco sont livrés avec un logiciel d'administration graphique, le PDM (**P**ix **D**evice **M**anager). Ce logiciel est relativement intuitif et suffisant pour bénéficier de toutes les fonctions du pix et de l'assistance avancée. Grâce au pix 501 il sera possible de mettre en place une connexion VPN appuyée par l action du puissant algorithme ASA (Adaptative Security Algorithm) et du cryptage de donnée 3DES (Triple Data Encryption Standard) de 128bits. Mais l'un des avantages indéniable de ces pixs reste les nombreuses fonctions d'administration à distance. La prise en charge du protocole de surveillance à distance SNMP (Simple Network Management) et la possibilité d'enregistrement des fichiers de log permet à l'administrateur de définir en temps réel sa politique de sécurité. En effet cette interaction est possible par le biais d'une simple interface web intégrée à PDM permettant d'effectuer une surveillance et une administration complète.

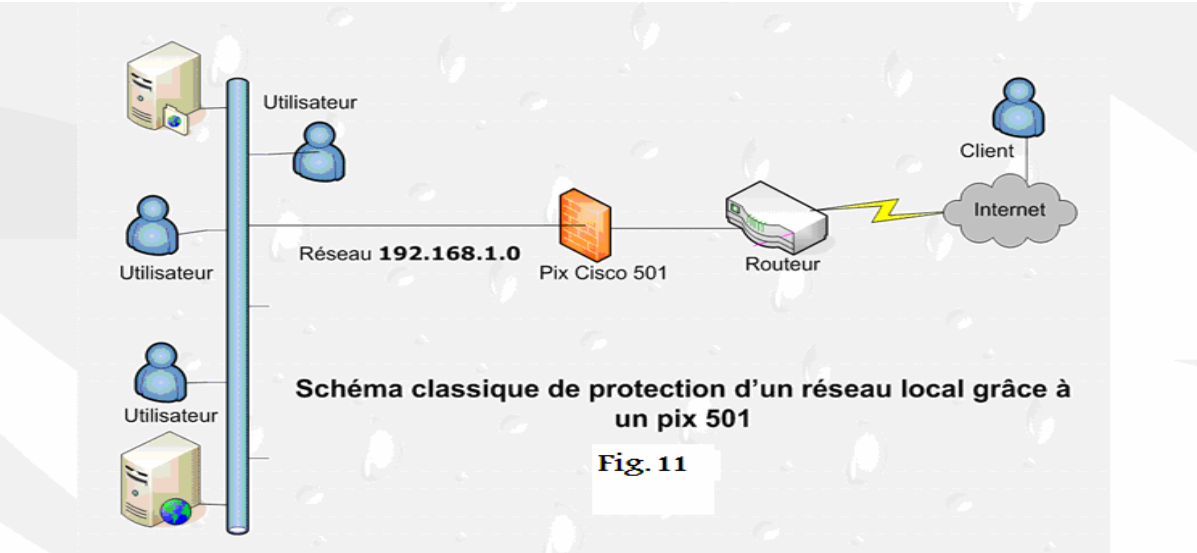

Schéma classique de protection d'un réseau local grâce à un pix 501

Fig. 11

a) Translation d'adresses

Le pix 501 offre des fonctions de traduction d'adresses NAT (Network Address Translation) et de port PAT (Port Address Translation) qui cachent les véritables adresses des équipements du réseau, les adresses réseau du réseau 192.168.1.0 ci-dessus seront totalement invisible pour l'utilisateur Client.

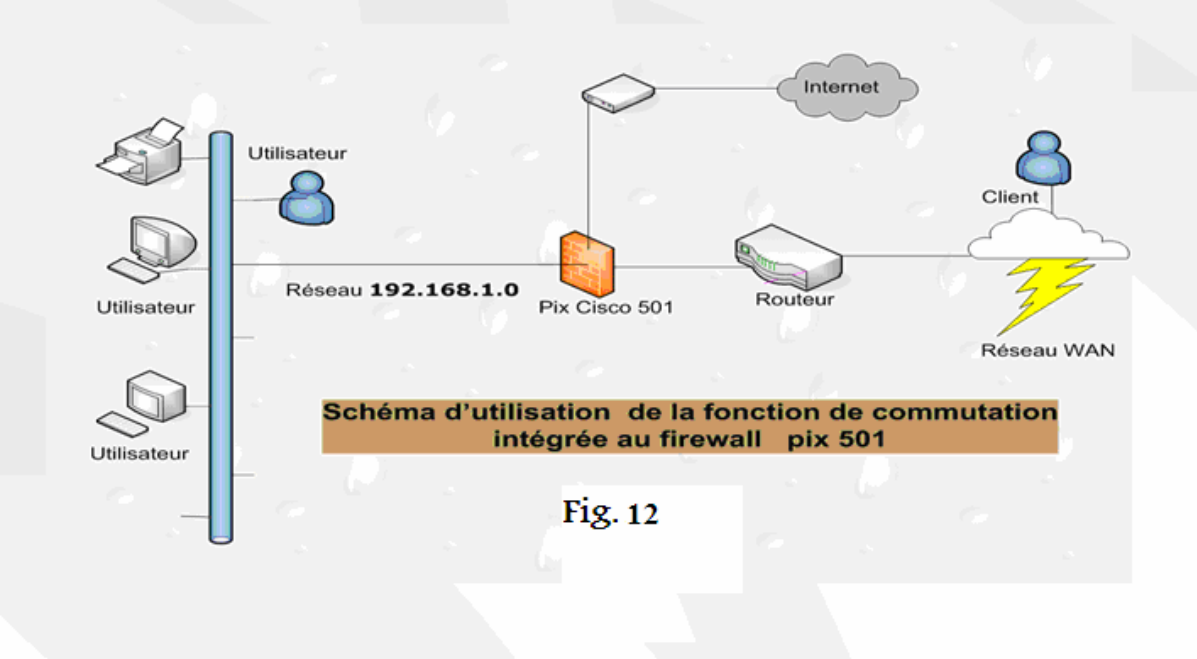

Fig. 12

b) redirection des événements Syslog vers le serveur Syslog:

Pour envoyer les messages d'alertes à travers le réseau vers un collecteur il faut spécifier dans la configuration du firewall, le collecteur qui reçoit les messages de syslog. Le firewall Cisco PIX peut maintenant envoyer des messages à travers l'UDP ou le TCP. Le port par défaut d'UDP est 514 et. le port par défaut de TCP est 1468. La commande à introduire dans la configuration du firewall est comme suit:

Logging host [interface] ip_address [protocol/port]

Ouvrir une session sur l'interface du Pix à la direction de la machine possédant une adresse ip (à spécifier) selon le protocole qui utilise un numéro de service à déterminer:

Exemple :

Logging host inside 10.20.1.1

c) La structure de fichier log du Firewall Cisco 501

L'unique façon de délivrer des services efficaces de contrôle de contenu serait de rassembler les paquets, reconstruire les données – fichiers, programmes, etc. Enfin scanner le tout pour rechercher, en une seule fois et rapidement, les virus, vers, URLs non autorisées et mots interdits. C'est pourquoi ont été conçues les passerelles Cisco. Ce sont les premiers équipements à avoir été élaborés pour fournir une protection complète du réseau, au sein d'une applicative performante, unique et intégrée. Tout événement, , peut être enregistré dans un fichier log :

− Local au firewall Cisco,

− Distant, envoyé alors à travers le réseau au format syslog.

La politique de log est adaptable en fonction de la sévérité des informations (emergency, alert, critical, error, warning, etc.).

Le Firewall permet de visualiser séparément les logs des services pare-feu, filtrage de contenu Web, filtrage de contenu Mail, et détection d'intrusion. Les logs locaux peuvent être téléchargés au format texte, XML ou CSV le fichier journal ou log du firewall Cisco pix peut contenir un ensemble d'entrées de huit niveaux de sévérités comme indique le tableau ci-dessous:

Sévérité	Description	Action
0	Message d'urgence (Emergency)	Système inutilisable.
1	Messages d'alertes (Alert)	Action immédiate requise.
2	Messages critiques (Critical)	État critique.
3	Messages d'erreur (Error)	Condition d'erreur.
4	// d'avertissement (Warning)	Condition d'avertissement.
5	Messages de note (Notification)	Normale mais état significatif.
6	// informationnels (Information)	Message informationnel seulement.
7	// d'élimination des anomalies (Debugging)	Apparaît pendant la correction seulement.

Fig. 13 - Niveaux de sévérités du fichier log du firewall Cisco 501

II.6. Exemples de fichiers log

Cisco Router:

May 15 12:10:27: %SEC-6-IPACCESSLOGP: list 102 permitted tcp

65.65.65.65(1355) -> 10.10.10.10(80), 1 packet

Snort :(NIDS)

[**] [1:971:1] WEB-IIS ISAPI .printer access [**]

Classification: Attempted Information Leak] [Priority: 3]

05/15-12:10:29.100000 65.65.65.65:1355 -> 10.10.10.10:80

TCP TTL:63 TOS:0x0 ID:5752 IpLen:20 DgmLen:1234 DF

AP Seq: 0xB13810DC Ack: 0xC5D2E066 Win: 0x7D78 TcpLen: 32

TCP Options (3) => NOP NOP TS: 493412860 0

[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0241]

[Xref => http://www.whitehats.com/info/IDS533]

Firewall pix 501: journal des événements d'un firewall Cisco pix 501

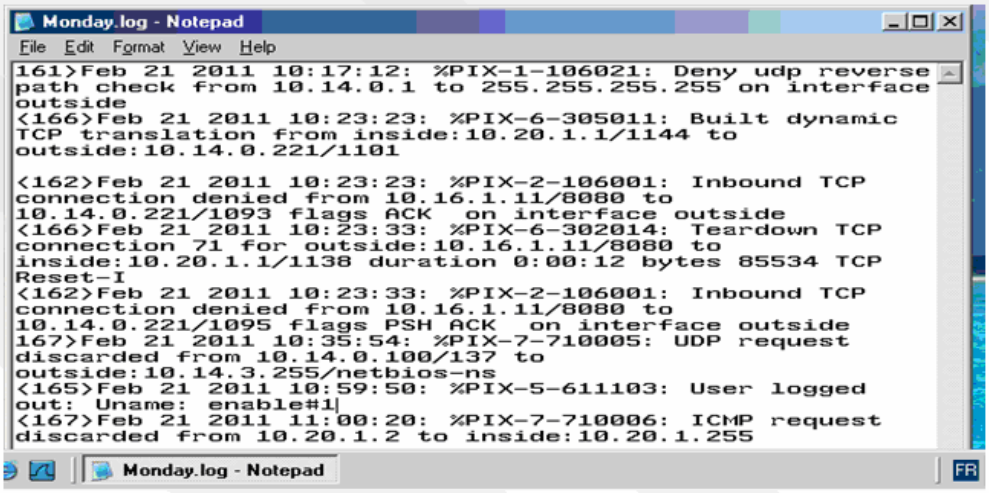

Fig.14 fichier log pour Cisco pix 501

II.7 Formatage du trafic réseau traversant le firewall

Pour qu'on puisse analyser notre fichier journal le formatage à un format lisible du fichier syslog est nécessaire; deux stratégies de formatage sont possibles:

1) Ouvrir le fichier syslog enregistré dans le dossier crée par le serveur syslog (application spécifique de Cisco) à la date et le jour courant en utilisant l'éditeur du capteur trafic ethereal ou wireshark et l'en enregistré en format CSV (comma separated values).
2) Placer le capteur de paquets (wireshark ou etheral) en état actif (promiscuous mode), sur l'interface dédiée à l'enregistrement des événements syslog Puis en temps différé on peut exporter le fichier journal capté en format CSV.

Le fichier log formaté du firewall Pix Cisco 501:

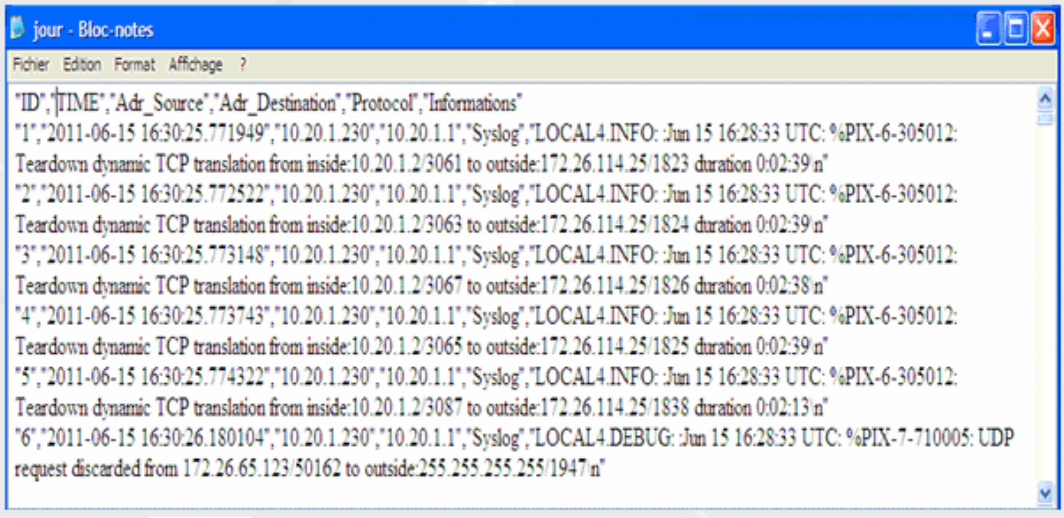

Fig.15 Fichier log formaté pour FIREWALL Pix 501

Le fichier **journal.csv** se transforme en une table MYSQL dont les champs sont respectivement ID, DTIME, Adr_Source, Adr_Destination, Protocol, et le champ Informations comme indique le schéma ci-dessous(Fig 16) :

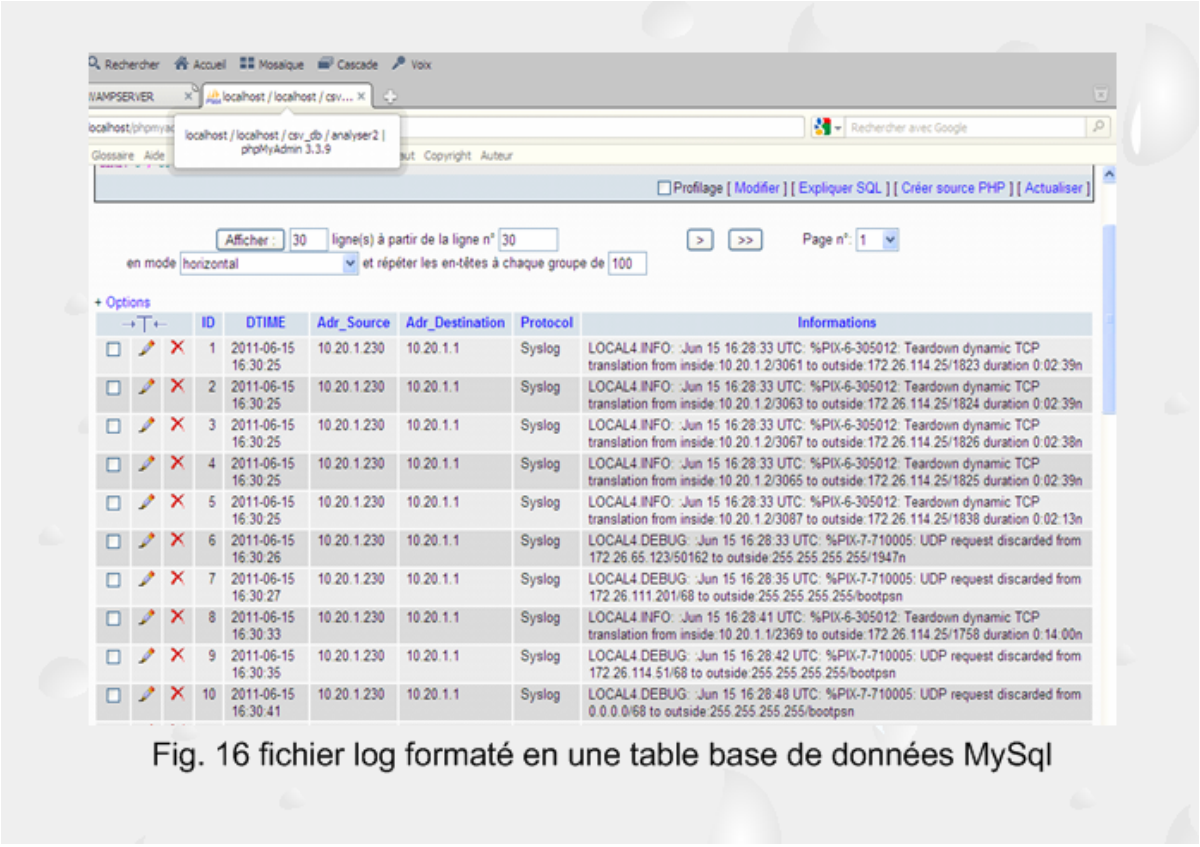

Fig. 16 fichier log formaté en une table base de données MySql

Le champ à exploiter par la suit pour la recherche et le calcul des drapeaux Syn et Fin dans chaque entrée de la table est le champ **Informations.**

II.8 Conclusion:

La journalisation applicative consiste à enregistrer les opérations de la logique métier pendant le fonctionnement d'une application. Un journal applicatif fait partie de la logique applicative. La journalisation du système enregistre les évènements survenants au niveau des composants du système. Il est possible de filtrer les évènements par catégories de gravité, en paramétrant la journalisation (erreurs, débogage, alertes...). Les systèmes Unix utilisent le protocole **Syslog** pour mettre en œuvre la journalisation système. Les logs de Windows se trouvent dans le dossier System32. Chaque action d'un système informatique (ouverture d'une session, installation d'un programme, navigation sur Internet...) produit un fichier log. Ces fichiers textes listent chronologiquement les évènements exécutés. Ils s'avèrent utiles pour comprendre la provenance d'une erreur en cas de bug. Ils permettent également d'établir des statistiques de connexions à un site Web ou à un serveur. **L'analyse** des fichiers logs peut se faire manuellement, mais les logs ne sont pas aisés à déchiffrer, et les outils d'analyse fournissent la plupart des informations nécessaires, c'est par exemple dans notre étude est de fournir un rapport sur la présence des traces d'une attaque dénie service de type TCP /Syn flooding.

Chapitre III

Les algorithmes de l'approche comportementale

III.1. approche d'analyse comportementale (détection d'anomalies)

Rappelons que l'approche comportementale se base sur l'hypothèse que l'exploitation d'une faille du système nécessite une utilisation anormale de ce système, et donc un comportement inhabituel de l'utilisateur. Elle cherche donc à répondre à la question « le comportement de l'utilisateur ou du système est-il normal ?

III.2. Architecture de l'approche comportementale:

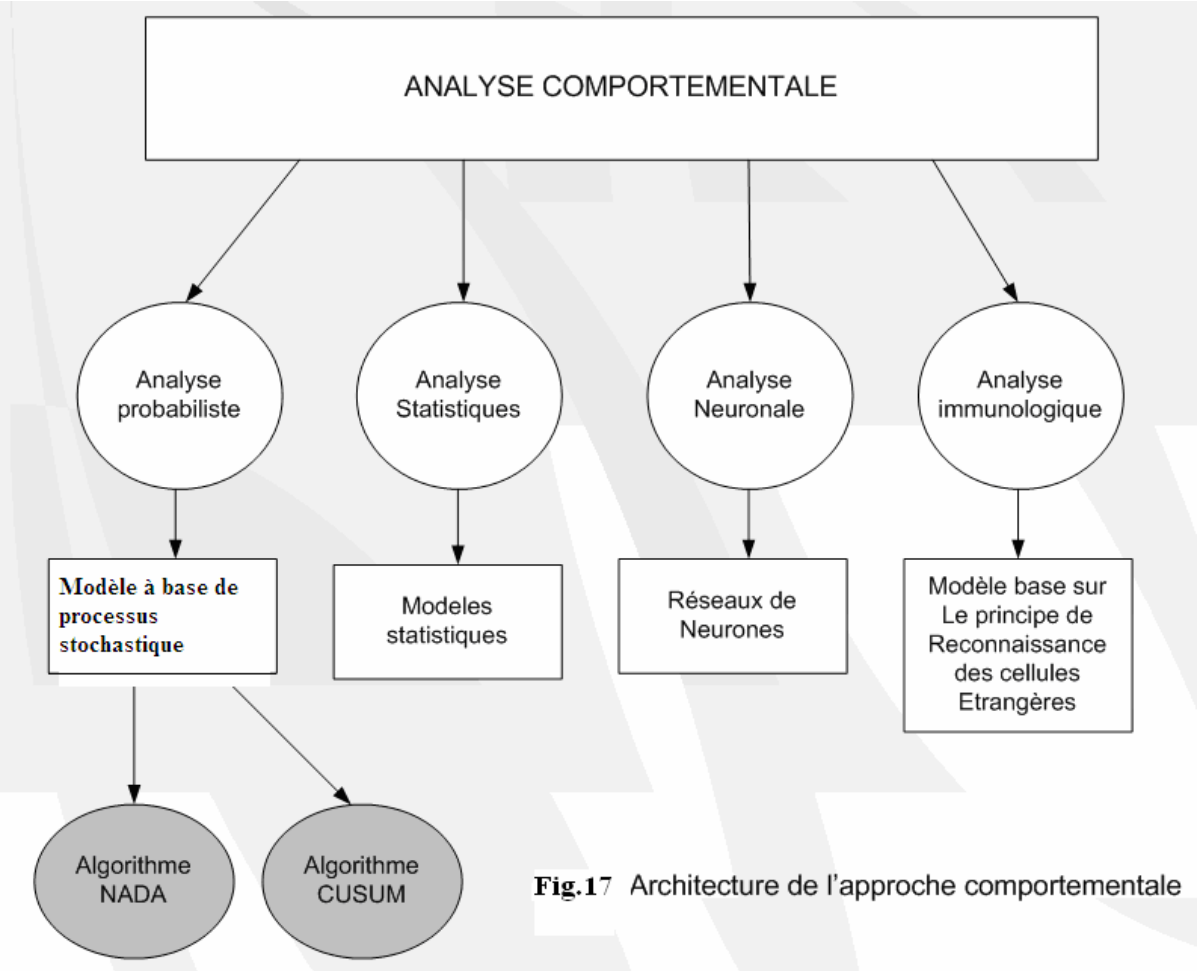

Fig.17 Architecture de l'approche comportementale

D'après le schéma ci-dessus on peut remarquer que cette approche s'adapte à plusieurs types d'analyse et chaque type d'analyse modélise le problème selon un modèle bien défini. Dans notre cas nous allons concevoir notre système selon une analyse probabiliste et statistique avec l'implémentation de deux algorithmes NADA et CUSUM.

III.3 Analyse comportementale

III.3.1. Les méthodes statistiques: le modèle de Denning

Le modèle de Denning est un modèle de comportement qui se compose de six éléments **[06]**:

- **Les sujets :** ce sont les initiateurs de toute activité observée sur le système, c'est-a-dire les utilisateurs ou les processus agissant pour leurs comptes.
- **Les objets :** ce sont toutes les ressources du système (fichiers de données, fichiers de programme, messages, périphériques, ...). La granularité des objets sera plus ou moins fine selon le degré de sécurité recherché : dans certains cas chaque fichier donnera lieu à la définition d'un objet, dans un autre il faudra descendre au niveau des enregistrements, dans un troisième cas le niveau répertoire sera suffisant.
- **Les enregistrements d'audit :** ils sont générés par le système lors de toute action entreprise par un sujet sur un objet. Ils se composent des informations suivantes :
  - ✓ Le sujet ayant entrepris l'action,
  - ✓ Le type de l'action (par exemple login, lecture, écriture, ...),
  - ✓ Les objets affectés (une action peut impliquer plusieurs objets),
  - ✓ échec ou succès de la commande,
  - ✓ Des éléments quantitatifs sur l'action (par exemple nombre d'octets ou d'enregistrements transférés dans une copie de fichier),
  - ✓ Un horodatage de l'action.
- **Les profils :** ce sont des structures qui caractérisent le comportement des sujets vis à vis des objets, par l'intermédiaire de valeurs statistiques résultant de l'observation des sujets. Un profil caractérise le comportement d'un utilisateur ou d'un groupe d'utilisateurs envers un objet ou un ensemble d'objets. Il donne une vue synthétique des actions des utilisateurs sur les objets.

Les enregistrements d'anomalie : ils sont générés quand une activité anormale est détectée,

Les règles d'activité : elles définissent les actions à entreprendre lorsque certaines conditions sont remplies sur les enregistrements d'audit ou les enregistrements d'anomalie. Elles ont la forme classique conditions-actions. Les actions peuvent par exemple consister à alerter l'administrateur de sécurité. Un profil est constitué par un ensemble de variables représentant une quantité accumulée pendant une certaine période de temps (minute, heure, journée, semaine, ... ou intervalle entre deux événements audités particuliers, par exemple entre connexion et déconnexion). Voici quelques exemples de variables possibles :

- ✓ nombre de mauvais mots de passe saisis en une minute,
- ✓ nombre, en millièmes de seconde, de quantums de temps CPU occupés par un programme entre son lancement et sa terminaison,

- ✓ nombre de fois qu'une commande système particulière est exécutée par un utilisateur donné, pendant le temps ou il est connecté,
- ✓ etc.

Le modèle statistique permet de déterminer, au vu de n observations x_1, ..., x_n faites sur une variable x, si la valeur x_{n+1} de la (n+1)ème observation est normale ou non. DENNING propose plusieurs modèles :

- Comparaison de la nouvelle valeur de x **avec une limite fixe :** dans ce cas les n observations précédentes ne sont utiles que pour se donner une idée de la limite en question.

Exemple : on considère qu'il y a essai d'intrusion si on mesure dix mots de passe erronés en une minute.

- Utilisation de la moyenne et de l'écart type des n **observations précédentes :**

L'observation n+ 1 est considérée comme anormale si x_{n+1} sort de l'intervalle de confiance défini par un écart de ($\pm D \times \sigma$) autour de la moyenne. Ce modèle présente l'avantage d'être capable d'apprendre ce qui est (normal) à partir des observations passées. Pour cela, on met à jour la moyenne et l'écart type à chaque nouvelle observation en pondérant les observations de manière à ce que les plus récentes aient le plus fort poids.

- Utilisation des covariances : ce modèle est similaire au précédent mais il permet de combiner plusieurs variables afin d'en tirer une synthèse. Il permet d'exploiter le fait que deux mesures (ou plus) représentent des manières différentes de caractériser le même aspect du comportement d'un utilisateur.

- Utilisation des processus de Markov : ce modèle permet de définir la probabilité du passage d'un état, défini par le contenu d'un enregistrement d'audit, à un autre état, également défini par un enregistrement d'audit. Est considéré comme anormal un enregistrement qui apparaît et dont la probabilité, au regard des états précédents, est trop faible. Cette approche est intéressante pour les attaques dans lesquelles est requis un enchainement de commandes dans un certain ordre.

Utilisation des séries temporelles : une nouvelle observation est anormale si sa probabilité d'apparition, au moment où elle apparait, est trop faible.

III.3.2. analyse neuronale

On peut envisager l'application des réseaux de neurones à la détection d'intrusion de plusieurs manières :

- Pour une modélisation statistique du comportement des utilisateurs Les réseaux de neurones sont très proche des méthodes statistiques telles que celles présentées ci-dessus, l'avantage des réseaux de neurones étant qu'il n'est pas nécessaire de faire d'hypothèses sur les variables aléatoires
- Pour classifier le comportement des utilisateurs par un algorithme de type (carte de Kohonen) classification automatique et non supervisée.

- Pour prédire le comportement des utilisateurs. Le réseau apprend les séquences de commandes usuelles à chaque utilisateur. Il lui est alors possible, après chaque commande passée par l'utilisateur, de prédire la commande suivante sur la base de ce qu'il a appris. En cas de déviation entre la prévision et la réalité, une alarme est émise. Il faut cependant noter:
  - ✓ Qu'un réseau de neurones ne fournit pas d'explication sur le raisonnement l'ayant amené à proposer un diagnostic d'intrusion,
  - ✓ Que le paramétrage d'un réseau de neurones est délicat et peut influer considérablement sur les résultats fournis.

III.3.3. Systèmes experts

Pour représenter l'usage (normal) qu'un utilisateur fait du système, il est possible d'utiliser un ensemble de règles au lieu d'un modèle statistique. Cela permet d'utiliser un système expert comme outil de détection d'intrusion.

Les règles d'un tel système expert peuvent être, soit entrées manuellement, soit générées automatiquement à partir des enregistrements d'audit. L'entrée manuelle sera par exemple utilisée pour exprimer une politique de sécurité. Les règles générées décrivent quant à elle des comportements.

Les systèmes experts présentent un inconvénient souvent cité : la base de règles est ni simple à créer, ni simple à maintenir**.**

III.3.4. Analyse immunologique :

L'analyse immunologique tente de calquer le comportement du système immunologique pour faire la différence entre ce qui est normal et ce qui ne l'est pas. Le système immunologique montre en effet beaucoup d'aspects intéressants comme son mode d'opération distribué (il n'y a pas de système de contrôle central) qui lui permet de continuer à fonctionner même après des pertes, sa capacité à apprendre automatiquement de nouvelles attaques pour mieux réagir les prochaines fois qu'elles se présentent, sa capacité à détecter des attaques inconnues, etc. On peut voir l'analyse immunologique de la détection d'anomalies comme une méthode de détection d'anomalie où l'on utilise les techniques de détection des malveillances. En effet, les techniques de détection d'anomalie connaissent ce qui est bien et vérifient en permanence que l'activité du système est normales, alors que les techniques de détection de malveillance connaissent ce qui est mal et sont à sa recherche. L'approche immunologique propose de rechercher ce qui est mal en connaissant ce qui est bien.

Inconvénients :

- Choix délicat des différents paramètres du modèle statistique.
- Difficulté à dire si les observations faites pour un utilisateur particulier correspondent à des activités que l'on voudrait empêcher.

- Pour un utilisateur au comportement régulier, toute activité est normale.
- Pas de prise en compte des tentatives de collusion entre utilisateurs.
- Utilisateur pouvant changer lentement de comportement dans le but d'habituer le système à un comportement intrusif.

III.4. le trafic réseau irrégulier et régulier

III.4.1 Le trafic régulier

Selon des intentions finales, différentes définitions peuvent êtres employées pour définir le trafic réseau sur l'internet .par exemple une définition académique **[15]** ressemblerait à << Le trafic réseau est représenté par le transport des messages ou des données par un système de communication telle qu'un routeur qui contrôle le trafic sur Internet>>. D'une part, les définitions orientées à la caractérisation du trafic réseau emploieraient des concepts tels que la dépendance et la similitude. Ces deux concepts (dépendance et similitude) sont employés pour expliquer la variabilité du trafic, la figure ci-dessous montre clairement, en exhibant ce qui ressemblerait au trafic « presque » d'une source de CBR [Constant Bit Rate.](a), d'une source de Poisson (b) et d'une vraie source du trafic (c). Les sources artificielles qui suivent des modèles plus simples sont presque constantes en comparaison avec les vraies sources du trafic, qui sont variables sans interruption.

III.4.2 Le trafic irrégulier

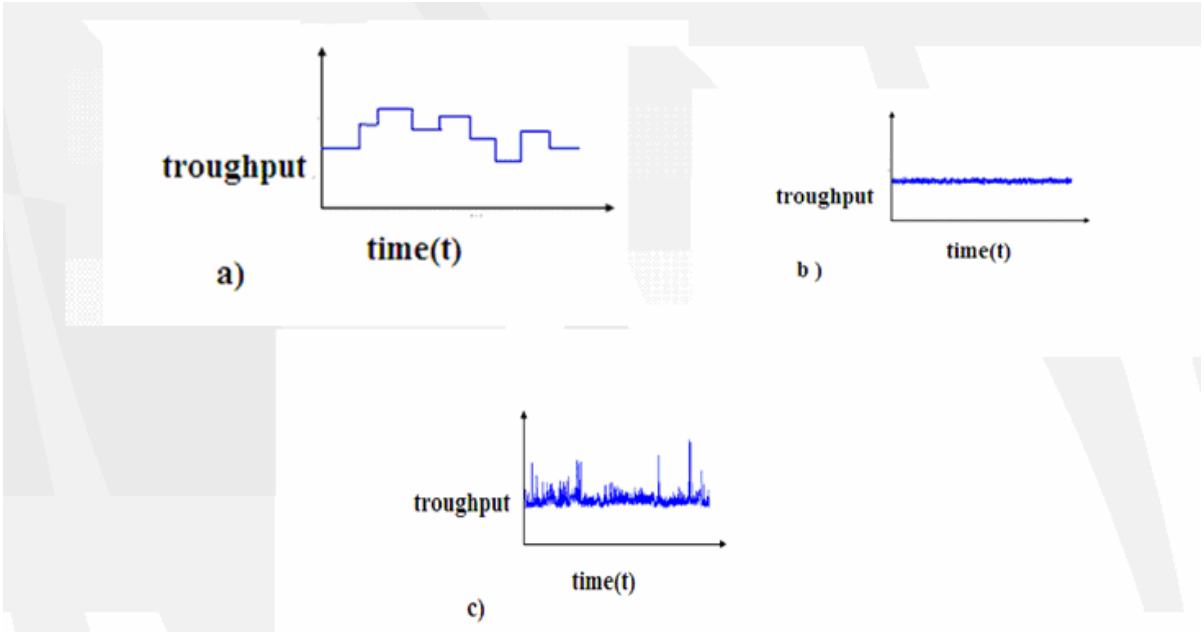

Fig.18 .les différents modèles utilisés pour le trafic internet

Selon **[18]**, la variabilité ne peut pas être dissociée du trafic réseau, et jusqu'ici plusieurs efforts effectuaient afin de développer des modèles capables de mesurer une telle variabilité et les intégrer dans des outils de gestion de réseau. Cependant, l'amplitude d'oscillations du trafic est

non seulement due aux causes intrinsèques mais elle peut également être due aux facteurs externes. Un d'intérêt particulier est le trafic anormal qui s'introduit dans le trafic régulier et le transformant en trafic irrégulier. Des anomalies du trafic peuvent être perçues en tant qu'événements peu communs, certains étant malveillant à cet affect l'écoulement régulier du trafic, pourraient augmenter ou diminuer des paramètres du trafic (par exemple, nombre de paquets, nombre de bytes, nombre d'écoulements, etc.).

Le trafic anormal, étant un élément instable qui influe sur le changement du trafic régulier, il demeure toujours inattendu, et renforçant la tâche de son traitement.

III.5 Les séries temporelles

L'évolution temporelle du trafic internet à destination d'un réseau local, augmentant de façon linéaire est ce la en fonction du temps pendant la durée globale de la trace. La segmentation de la trace en intervalles temporels de taille Δ. Sur chacun d'entre eux, on détermine soit :

- Le nombre de paquets (séries temporelle à base du nombre de paquets)
- Le nombre d'octets ou bytes (séries temporelle à base du nombre d'octets)
- le nombre de nouveaux flots par unité de temps.

Définition : Une série temporelle est une suite variable du nombre de paquets, octets ou flot; obtenues en agrégeant ce nombre dans des boîtes temporelles successives de taille bien choisie **[18]**.

a) **Séries temporelles à base du nombre de paquets**: Constituées par le nombre de paquets agrégés dans des boîtes de taille Δ. En peut utiliser des statistiques simples sur le nombre de paquets SYN ou SYN- FIN.

b) **Séries temporelles à base du nombre d'octets**: dans ce type on considère le nombre d'octets envoyés pendant un intervalle Δ.

c) **Séries temporelles à base flot**: le niveau d'agrégation dans ce type de série est le niveau d'agrégation adresses IP, qui peuvent aller de la taille de préfixe /0 (tous les paquets sont considérés dans un flot unique pour l'analyse) jusqu'à /32 (chaque flot composé des paquets provenant ou allant vers une adresse IP unique sont analysés individuellement), en passant par les tailles de préfixes intermédiaires (pour lesquelles les flots contenant tous les paquets venant ou à destination des adresses ayant un préfixe donné sont analysées individuellement). C'est en découpant ainsi à chaque étape le trafic en tranches tomographiques de plus en plus fines que l'on parvient à détecter les anomalies les plus massives et celles qui sont plus discrètes. Ainsi, dans le processus récursif, à chaque étape un nouvel ensemble de séries temporelles correspondant aux critères considérés est calculé pour une taille de fenêtre et une taille de préfixe donnée. A chaque étape ces séries sont analysées. De plus, pour chaque niveau d'agrégation /n des

adresses, toutes les fenêtres temporelles sont étudiées. Et tous les préfixes d'adresses qui apparaissent dans le bloc de trafic considéré sont analysés.

III.6 Algorithme de la somme cumulative (CUSUM).

III.6.1 Introduction

Cusum (la somme cumulative) est l'algorithme le plus connu parmi la famille des algorithmes de détection d'anomalie. Son but est de détecter tout changement considéré remarquable ou brusque de modèle dans des processus stochastiques **[11]**.Le rôle essentiel de la détection de ce changement, est de pouvoir conclure le plus rapidement possible, le passage d'un état de fonctionnement normal A1 à un état de fonctionnement considéré anormal A2.

III.6.2 Détection de changement et règle du CUSUM

L'objectif de la détection de changement de valeur de paramètre, est de pouvoir repérer, le plus rapidement possible, le passage d'un état de fonctionnement normal, appelé régime $\mathbf{H_0}$, à un état de fonctionnement anormal, appelé régime $\mathbf{H_1}$, pour le système sous surveillance. On peut caractériser ce changement de régime par un changement de valeur d'un paramètre θ dans la modélisation du système, qui va passer de la valeur θ_0 sous $\mathbf{H_0}$ à la valeur θ_1 sous $\mathbf{H_1}$. Le temps d'arrêt de la règle du CUSUM est alors défini comme le premier instant où la statistique de test, en moyenne proche de **0** sous $\mathbf{H_0}$, positive et croissante sous $\mathbf{H_1}$, franchit un seuil **h** fixe par l'expérimentateur.

III.6.3. Description de l'algorithme.

Une méthode statistique couramment utilisée en détection d'anomalies réseau est l'algorithme CUSUM **[11].**Cet algorithme a été, entre autres, utilisé par [9] pour la détection d'attaques de type TCP/SYN flooding, qui est un exemple d'attaque de type DoS. Celle-ci exploite les caractéristiques du protocole TCP (Transmission Control Protocol) en engorgeant de paquets de synchronisation (SYN) la machine destination attaquée, qui doit

tenir à jour une table de demandes de connexion en attente d'un message d'acquittement ACK (ACKnowledgement) de la part de la machine source. Les ressources de la machine et la taille de la table étant limitées, l'attaque peut conduire à une saturation et à une interruption du service fourni par la machine.

Les machines victimes d'une attaque de type TCP/SYN flooding pourraient ainsi être détectées en appliquant un test de détection de ruptures aux séries temporelles correspondant au nombre de paquets (SYN-FIN reçus.

i. Etape numéro 1: consiste à segmenter l'intervalle du traitement en intervalles de temps de taille Δ . À chaque intervalle de temps, on calcule la valeur X_n qui présente le nombre des paquets (SYN)- le nombre de paquets (FIN) dans l'intervalle concerné.

ii. Etape numéro 2: cette étape à pour but de modéliser une fonction simple pour la détection d'une éventuelle rupture. Cette fonction est une fonction récursive appliquée dans chaque intervalle Δ_i.Si le résultat de cette fonction dépasse un seuil bien déterminé, une alarme est déclenchée.

L'algorithme CUSUM se base sur le principe de la somme cumulative qui se présente sous la formule suivante :

$$G_n = \left[G_{n-1} + \left(X_n - \mu\right)\right]$$

Où G_n est la somme cumulative positive qui est initialisée à 0 lors du lancement de l'algorithme. Ce qui signifie que si elle est négative, elle est immédiatement mise à 0.

La variable X_n présente le nombre de paquets (SYN-FIN)

μ est la moyenne des X_n calculée à l'instant t (n allant de 0 à t).

D'après **[10]** la valeur moyenne μ est définie comme suit :

$$\mu_n = 0.5\mu_{n-1} + 0.5X_{n-1}$$

Lorsque G_n dépasse un certain seuil h ($G_n \geq h$), une alarme se déclenche et réinitialise G_n à 0.

L'algorithme CUSUM présente plusieurs avantages notamment la rapidité de détection ainsi que la minimisation du taux de fausses alertes ce qui n'est pas le cas de tous les systèmes de détection d'intrusion, mais malgré son efficacité il présente diverses limites. Par exemple la nécessité d'avoir un contrôle humain pour vérifier la présence de l'attaque **et** l'incapacité de vérifier tout le trafic dans le cas d'un très haut débit **[11].**

III.6.4 Algorithme :

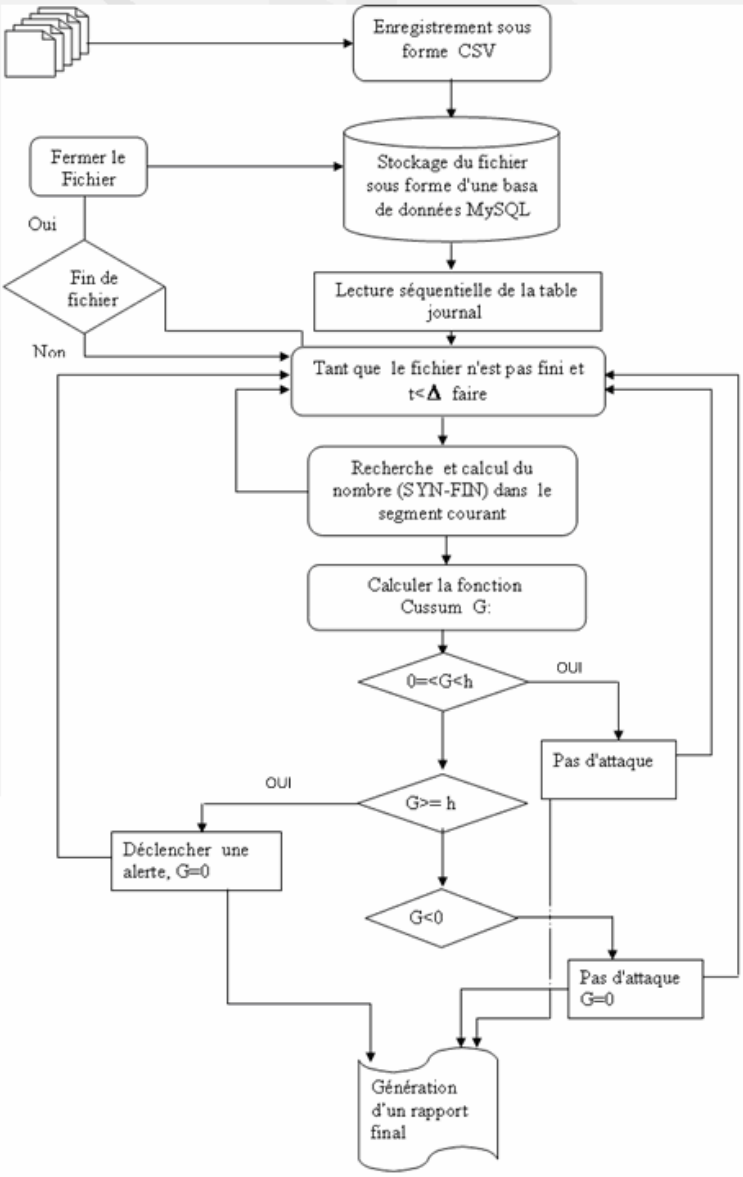

Fig.19 - Algorithme d'analyse CUSUM

III.7. Network Anomaly Detection Algorithm (Nada)

L'algorithme de détection d'anomalie de réseau suit une approche triple **[15]**. Pour atteindre son objectif NADA analyse le trafic avec possibilité d'utilisation d'une approche multi-échelle, multicritères et à plusieurs niveaux d'agrégation, en recherchant les variations significatives, qui peuvent correspondre à une anomalie du trafic.

Multi-échelle: il n'existe pas une échelle de temps unique qui permette de détecter toutes les anomalies chacune agit avec ses propres caractéristiques temporelles; c'est-à-dire les fenêtres temporelles d'observations ne sont pas les mêmes, et chaque type d'anomalie possède sa propre échelle d'observation pour une détection correcte.

Multicritères: les anomalies n'affectent pas les attributs du trafic réseau de la même façon. Chaque type d'anomalie est détecté à partir de l'attribut considérer comme témoin de sa présence. Par exemple un scan de port se détecte à partir des deux attributs port source et port destination et même pour le scan adresse par l'analyse des attributs adresse source et adresses destination.

Plusieurs niveaux d'agrégation: Le problème de l'agrégation à plusieurs niveaux consiste à synthétiser des informations traduisant des aspects ou des points de vues différents et parfois incertains au sujet d'un même ensemble d'objets. Il se pose de manière concluante dans nombre de procédures d'évaluation, de comparaison ou de classification.

En effet, de nombreux problèmes de décision sont multicritères. Ils se caractérisent alors par la définition d'un ensemble de solutions potentielles ou "alternatives" qui peut être défini en compréhension (système de contraintes), par la construction d'une famille de n fonctions d'agrégations à valeurs réelles, permettant d'évaluer les "performances" des différentes alternatives possibles selon chacun des points de vues jugés pertinents pour le problème, et enfin par la définition d'une problématique qui peut être le choix (sélection d'un ensemble aussi réduit que possible de "meilleures alternatives"), le rangement (ordonnancement total ou partiel des alternatives selon leur qualités relatives), ou le tri (affectation des alternatives à des catégories prédéfinies) **[19]**. Dans le cas de la détection d'intrusions dans le trafic réseau, les différents types d'anomalies se perçoivent mieux à différents niveaux d'agrégation du trafic. Par exemple les niveaux d'agrégation correspondant à des masques /0 (tout le trafic), /8, /16 et /24 **[17].**

III.7.1 Principe de fonctionnement

Cet algorithme considère qu'une anomalie est responsable d'une variation sur au moins un des critères considérés, à au moins une échelle de temps et à un niveau d'agrégation donnés **[10,15]**. Les échelles de temps choisies pour analyser le trafic vont de quelques microsecondes jusqu'à plusieurs heures. Pour cela, il compare les variations du trafic entre l'instant Δ_{i+1} et l'instant Δ_i.

L'algorithme Nada s'applique en deux phases:

- Première phase: phase d'apprentissage : elle peut s'étendre sur un ou plusieurs intervalles de temps, c'est une phase de pré-analyse.

- Deuxième phase : c'est d'appliquer le système d'équation suivant :

$$\left\{ \begin{array}{l} P_i \geq E(p) + k\sigma \rightarrow anomalie \\ P_i < E(p) + k\sigma \rightarrow normale \end{array} \right\}$$

Avec

$X = \{x_1 x_2, ..., x_n\}$, $x_i = \{packets / bytes / flows\} / \Delta$

$P = \{p_1, p_2, ..., p_{n-1}\}$ $\quad p_i = x_{i+1} - x_i$

P : une série obtenue en faisant la différence entre toutes les valeurs consécutives de X deux à deux. La moyenne E(p) et l'écart-type σ sont calculés pour la série temporelle considérée X et utilisés pour définir un seuil **[17]**. Chaque dépassement de ce seuil, est interprété comme la présence d'une anomalie à cet instant. Selon les lois utilisées dans les statistiques descriptives **[16]**, les paramètres E(p) et σ sont obtenus comme suit:

$$E(p) = \frac{1}{n-1} \sum_{i=1}^{n-1} p_i \quad ; V(p) = \frac{1}{n-1} \sum_{i=1}^{n-1} (p_i)^2 - (E(p))^2$$

$$\sigma = \sqrt{V(p)}$$

III.7.2 Description de l'algorithme NADA

L'algorithme NADA a pour particularité de détecter, classifier et identifier les anomalies du trafic. Cet algorithme peut travailler sur trois séries temporelles différentes : le nombre des octets (bytes), le nombre des flots (Flows) et le nombre des paquets (Packets), et par conséquent la détection des trois types d'attaques différentes :

- Attaque de type Bytes (octets)
- Attaque de type Flows (flots)
- Attaque de type Packets (paquets)

Les variations sur les différentes séries temporelles représentant les différents critères (octets, paquets, flots) sont détectées en utilisant l'équation précédente (voir chapitre III). Cela permet de ne détecter que les anomalies induisant une variation brusque et significative, et de ne pas être sensible aux variations normales du trafic. La diversité des disfonctionnements du réseau a motivé la conception et le développement de NADA (Network Anomaly Detection Algorithm) dont les objectifs sont au nombre de trois **[17]** :

La détection des anomalies: c'est-à-dire déterminer si une anomalie est en train de se produire. En particulier, l'objectif sera de détecter les anomalies. En effet, les attaques DoS sont de plus en plus souvent distribuées, chaque source de l'attaque ne générant que très peu de trafic, afin de rester invisible le plus longtemps possible. C'est en s'agrégeant massivement près de la victime que toutes ces composantes de l'attaque provoquent une dégradation brusque et importante du niveau de service fourni par le réseau et les serveurs qui y sont connectés : d'où l'intérêt de les détecter au plus tôt près de la source.

Identification des anomalies: consiste à spécifier, les informations continues dans la signature des anomalies détectées. L'identification des paquets ou les flux signalés; la fenêtre temporelle dans laquelle l'anomalie est détectée, la liste des adresses IP et des numéros de port impliqués dans l'anomalie et la liste des adresses IP et des numéros de port ciblés.

La classification des anomalies: l'algorithme de détection d'anomalie comporte trois étapes pour classifier les anomalies :

i. Une fois qu'une anomalie a été détectée, il identifie tous (ou la plupart) des paquets ou des flux la composant ;
ii. Il utilise ces informations sur les paquets et flux impliqués dans l'anomalie pour dériver plusieurs métriques distinctes directement en rapport avec l'anomalie ;
iii. Classifie l'anomalie en utilisant ces métriques selon une approche orientée signature.

Ces étapes reposent sur le besoin de disposer de beaucoup d'informations pour classifier de façon fiable les différents types d'anomalies, et même de pouvoir faire la distinction entre les différents sous-types, comme la multitude d'attaques DoS. Comme les algorithmes actuels de

détection d'anomalies reposent sur un petit nombre de paramètres (métriques volumiques ou attributs du trafic comme les adresses IP et les numéros de ports), une étape est nécessaire pour obtenir plus d'informations sur l'anomalie.

II.7.3 Algorithme Nada

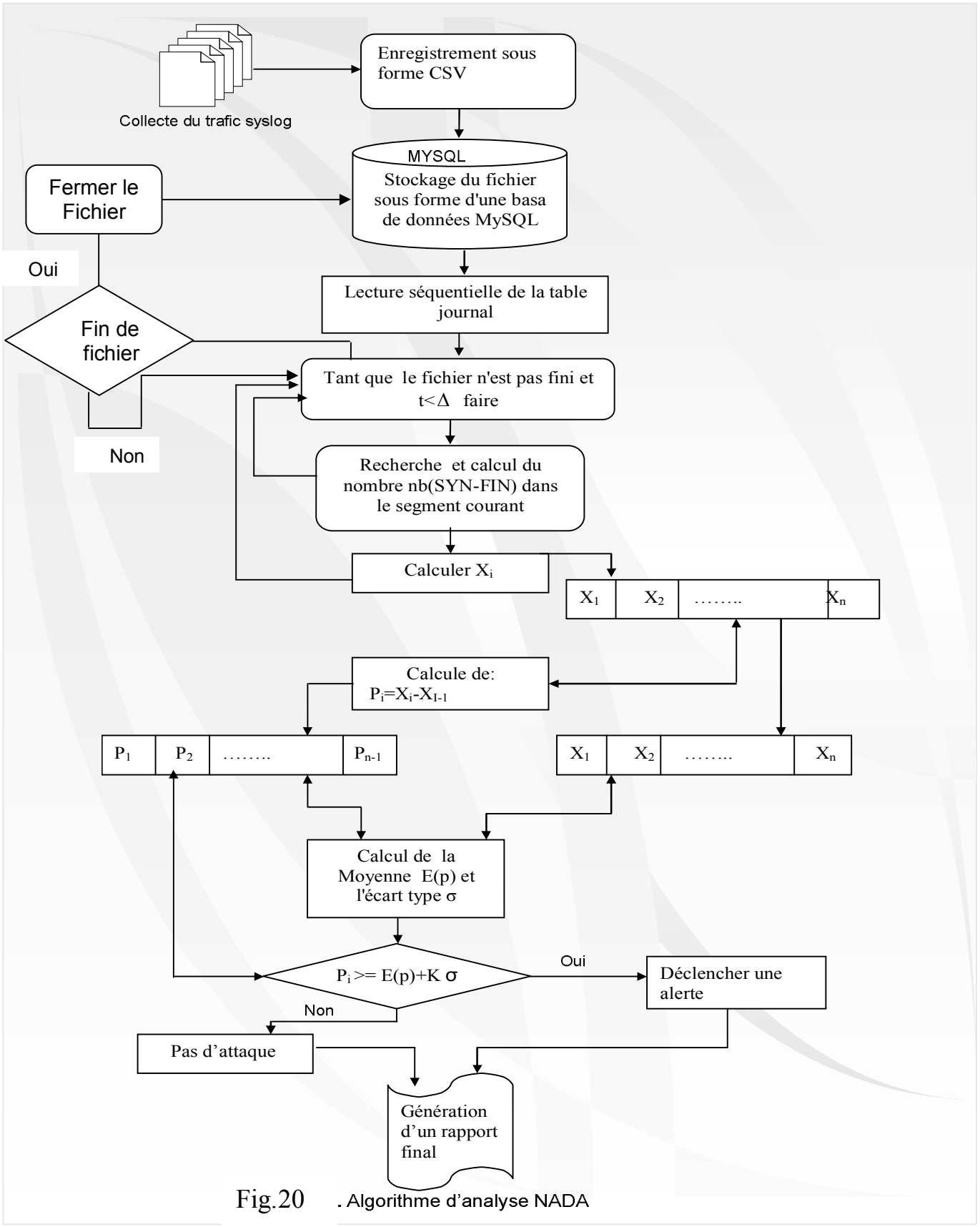

Fig.20 . Algorithme d'analyse NADA

III.8 Conclusion:

L'idée principale de cette approche est de considérer toute déviation, toute anomalie dans le comportement comme une intrusion. Cette hypothèse est certainement fausse : des événements ou des comportements rares peuvent tout à fait être légitimes du point de vue de la politique de sécurité du système. Le système est susceptible d'émettre des faux positifs. Tant que le nombre de faux positifs reste suffisamment faible, la méthode peut être valide.

Cela conduit à poser deux questions essentielles, dans le domaine de la détection d'intrusions comportementale, sur le caractère correct et complet du modèle de comportement normal .

Le modèle de comportement normal est dit correct s'il ne modélise que le comportement légitime, du point de vue de la politique de sécurité, de l'entité surveillée. Toutes les intrusions sont alors détectées par le système de détection : il n'y a pas de faux négatif.

Le modèle de comportement normal est dit complet s'il modélise entièrement le comportement légitime, du point de vue de la politique de sécurité, de l'entité surveillée. Dans ce cas, toutes les alertes correspondent à des intrusions : il n'y a pas de faux positifs.

Chapitre IV
Implémentation du système d'analyse du fichier log

IV.1 Introduction

La détection d'intrusions TCP Syn Flooding dans notre test est basée sur la surveillance de passage des événements du trafic traversant le firewall. Les événements sont enregistrés en forme de fichiers log. Les résultats de l'analyse sont générés par l'application de l'un des deux algorithmes utilisant un modèle mathématique statistique (Algorithme Nada)ou un modèle purement mathématique utilisant une fonction récursive (algorithme Cusum).notre firewall à une configuration de base**[ANNEXE-B]** de telle manière à ne bloquer aucun type du trafic(http ,TCP, UDP, etc) ; c'est-à-dire que les règles de passage ne sont pas introduites.

IV.2. Implémentation

IV.2.1. Schéma de la plate forme de test:

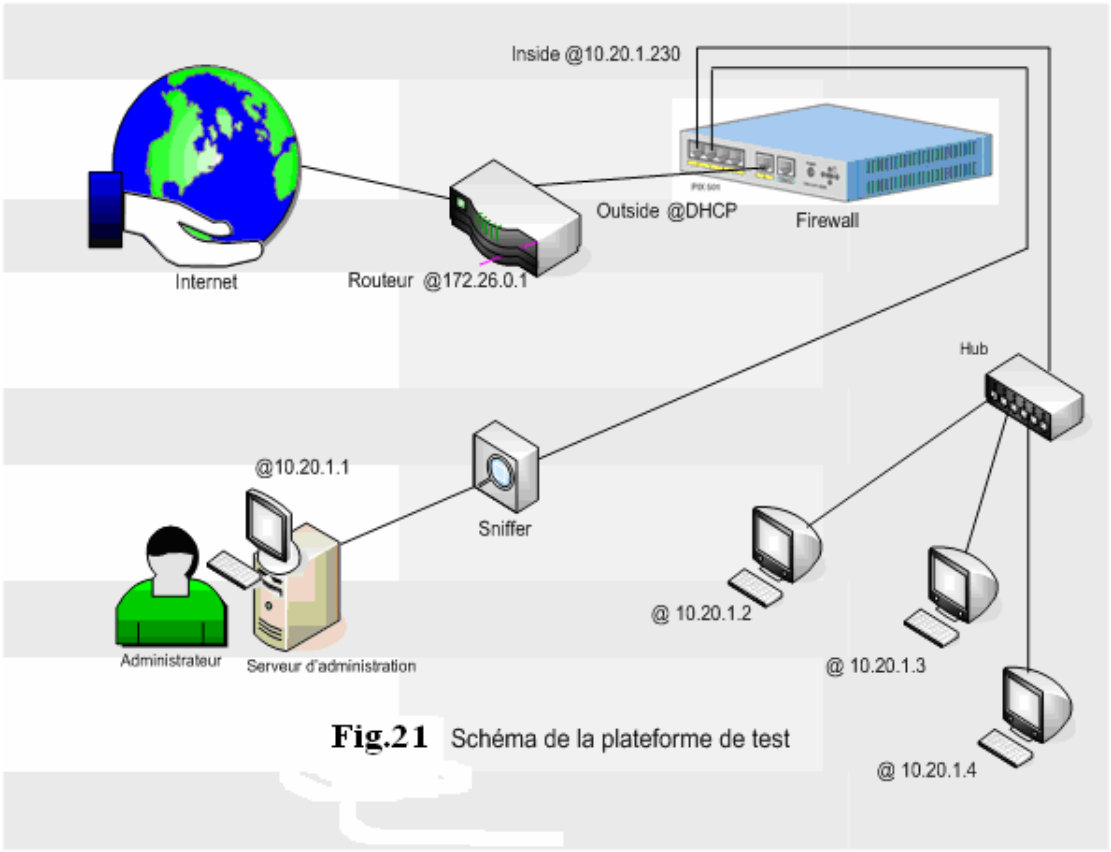

Fig.21 Schéma de la plateforme de test

D'après le schéma ci-dessus on remarque que le Proxy génère des adresses DHCP, mais l'installation d'un firewall en avant nous a permet de fixer nos adresses privées à l'intérieur du réseau local. Les événements Syslog sont envoyés directement vers une machine dédiée à cet effet sous forme de fichiers journaux. Les adresses du réseau internes sont cachées à l'extérieur grâce à la fonction de la NAT.

IV.2.2 Schéma fonctionnel de notre système.

Notre système comporte les fonctionnalités suivantes

- Collecte des fichiers journaux
- Formatage
- Envoi du fichier journal à un SGBD(MYSQL)
- Chargement de la table journal
- Analyse de la table journal par l'algorithme implémenté (NADA ou Cusum)
- Génération de rapports d'analyse:
  - ✓ Alertes d'intrusion
  - ✓ Filtrage de lignes journal selon les protocoles traversant le firewall
  - ✓ Statistiques sur les protocoles

Ces fonctions sont résumées dans le schéma ci-dessous (fig.22):

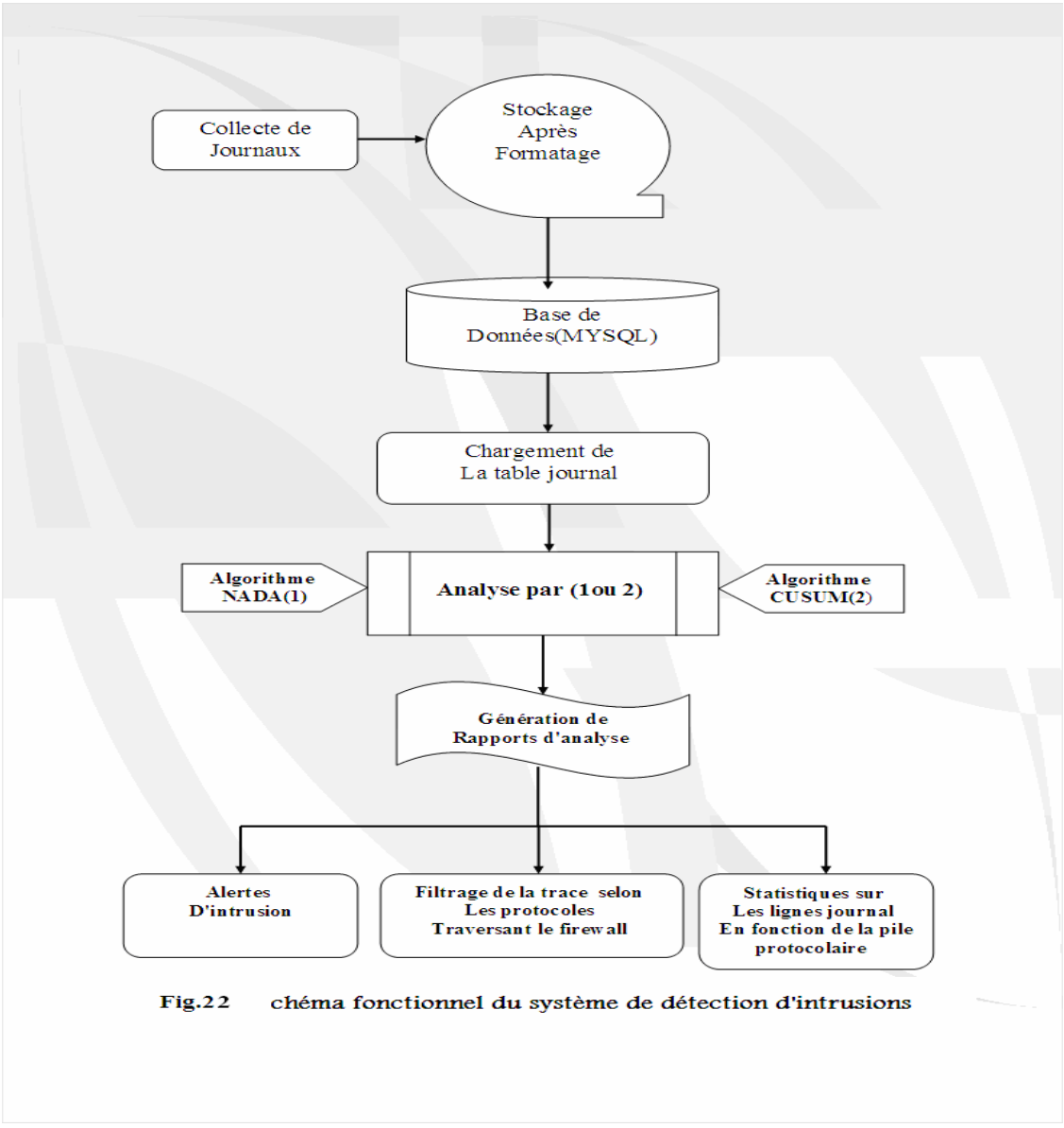

Fig.22 chéma fonctionnel du système de détection d'intrusions

IV.2.3 Environnement matériel et logiciel

- Base de données : MySQL 5.5.8
- Langage de programmation : PHP 5.3.5
- Outil de développement: WAMPSERVER 2.1
- Equipement: Pix CISCO 501.

IV.3 Application

Le menu de notre application comporte les fonctionnalités suivantes:

1. Chargement du fichier log à partir du serveur MYSQL
2. Analyse du fichier journal (trafic ou syslog) par algorithme **Cusum** ou NADA en exploitant le champ Informations pour la recherche des drapeaux Syn ou Fin dont chaque entrée.
3. Filtrage du journal suivant les protocoles utilisés
4. Statistiques protocolaires.

Comme indique l'écran ci-dessous (fig. 23) :

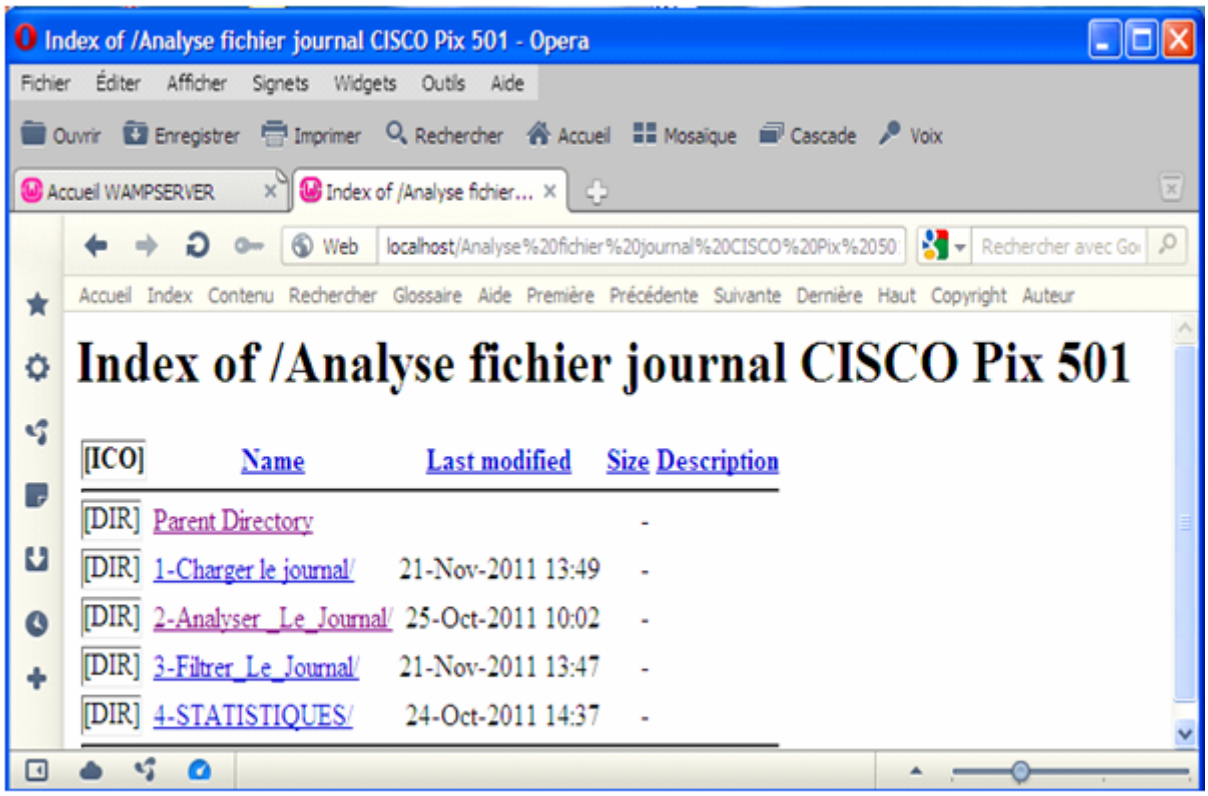

Fig.23 Menu principal

IV.3.1 Chargement de fichier journal:

Analyse de fichier log pour la detection d'intrusions
"La table journal Syslog"

Mardi 20 Décembre 2011

ID	DTIME	Adr_Source	Adr_Destination	Protocol	Informations
1	2011-06-15 16:30:25	10.20.1.230	10.20.1.1	Syslog	LOCAL4.INFO: :Jun 15 16:28:33 UTC: %PIX-6-305012: Teardown dynamic TCI
2	2011-06-15 16:30:25	10.20.1.230	10.20.1.1	Syslog	LOCAL4.INFO: :Jun 15 16:28:33 UTC: %PIX-6-305012: Teardown dynamic TCI
3	2011-06-15 16:30:25	10.20.1.230	10.20.1.1	Syslog	LOCAL4.INFO: :Jun 15 16:28:33 UTC: %PIX-6-305012: Teardown dynamic TCI
4	2011-06-15 16:30:25	10.20.1.230	10.20.1.1	Syslog	LOCAL4.INFO: :Jun 15 16:28:33 UTC: %PIX-6-305012: Teardown dynamic TCI
5	2011-06-15 16:30:25	10.20.1.230	10.20.1.1	Syslog	LOCAL4.INFO: :Jun 15 16:28:33 UTC: %PIX-6-305012: Teardown dynamic TCI
6	2011-06-15 16:30:26	10.20.1.230	10.20.1.1	Syslog	LOCAL4.DEBUG: :Jun 15 16:28:33 UTC: %PIX-7-710005: UDP request discard
7	2011-06-15 16:30:27	10.20.1.230	10.20.1.1	Syslog	LOCAL4.DEBUG: :Jun 15 16:28:35 UTC: %PIX-7-710005: UDP request discard
8	2011-06-15 16:30:33	10.20.1.230	10.20.1.1	Syslog	LOCAL4.INFO: :Jun 15 16:28:41 UTC: %PIX-6-305012: Teardown dynamic TCI

Fig.24 Chargement de la table journal

IV.3 2 Analyse du journal

Index of /Analyse fichier journal CISCO Pix 501/2-Analyser _Le_Journal

[ICO]	Name	Last modified	Size	Description
[DIR]	Parent Directory		-	
[DIR]	1-Programme_NADA/	24-Oct-2011 16:28	-	
[DIR]	2-Programme_CuSum/	24-Oct-2011 16:28	-	

Fig.25 algorithmes d'analyse

IV.3.2.1 Analyse par le programme NADA:

Comme cela été dit précédemment cet algorithme s'applique suivant deux phases:

- ✓ Une phase de pré analyse
- ✓ Une phase d'analyse

La phase d'analyse consiste à vérifier l'équation suivante:

Si $P_i \geq E(p) + K\sigma$ alors présence d'anomalie

Si non pas d'anomalie dans l'intervalle analysé

K une valeur positive, on a remarqué que des grandes valeurs de k génèrent un grand nombre des faux négatifs, alors que des petites valeurs génèrent un grand nombre des faux positifs. *K=2* est la meilleure valeur pour configurer NADA **[10].**Ce dernier détecte une anomalie si la valeur de P dans l'intervalle en cour est supérieure ou égale à un seuil**.**

L'exécution de notre programme Nada génère les résultats selon le tableau ci-dessous :

K	**Nombre faux positifs**	**Taux faux positifs**	**Nombre faux négatifs**	**Taux faux négatifs**
0	40	16.19 %	-	-
1	12	4.22%	-	-
2	0	0%	-	-
3	-	-	2	0.7%
4	-		3	1.05%
5	-	-	5	1.17%
>=6	-	-	6	2.11%

Tableau de test NADA en fonction des valeurs entières K

Remarquons que pour K=2 on détecte 06 intervalles qui présentent une anomalie de type TCP/ SYN Flood parmi 283 intervalles. C'est à dire un taux de: 2.1201413427562 %. C'est le nombre optimal d'alertes donc un taux de 0% de fausses alertes.

En résume les résultats du tableau ci-dessus dans le graphe suivant:

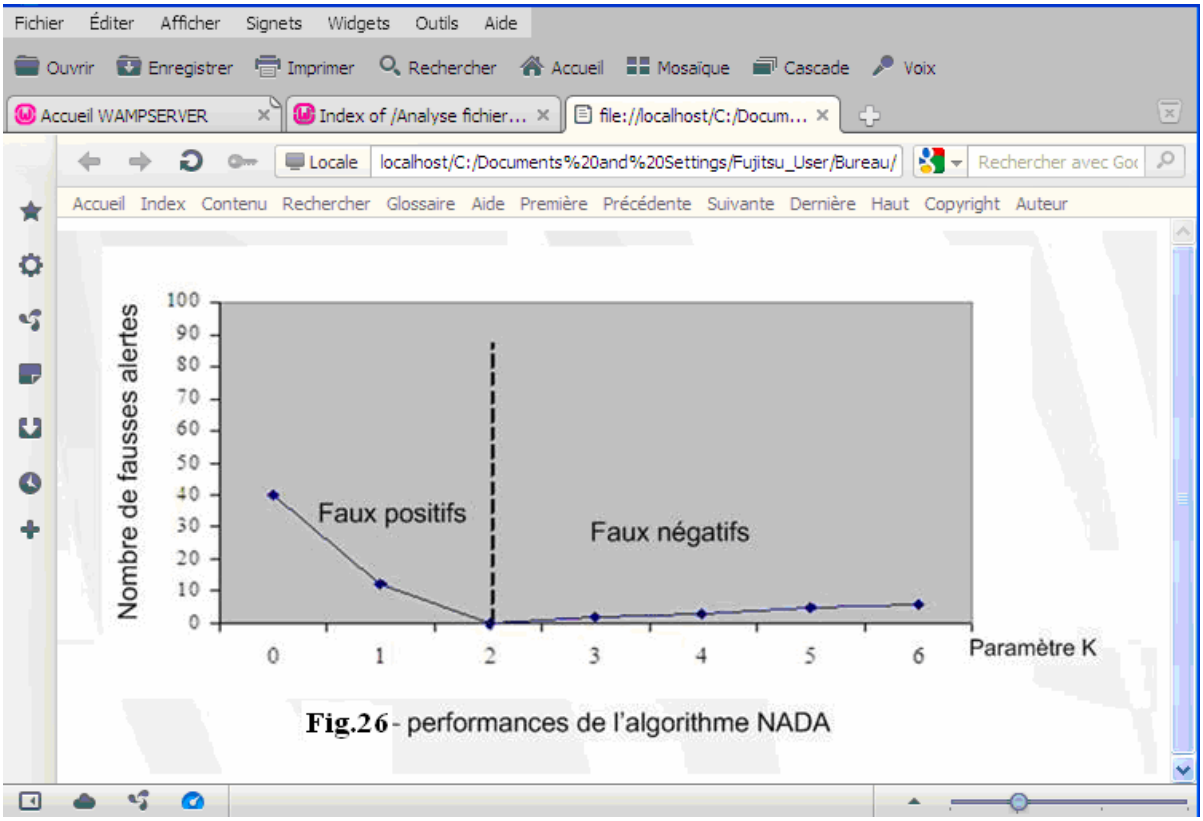

Fig.26- performances de l'algorithme NADA

Exécution du programme NADA : La valeur du coefficient K est égale à 2 et une fenêtre temporelle $\Delta = 01\,s$

Ecran 01:

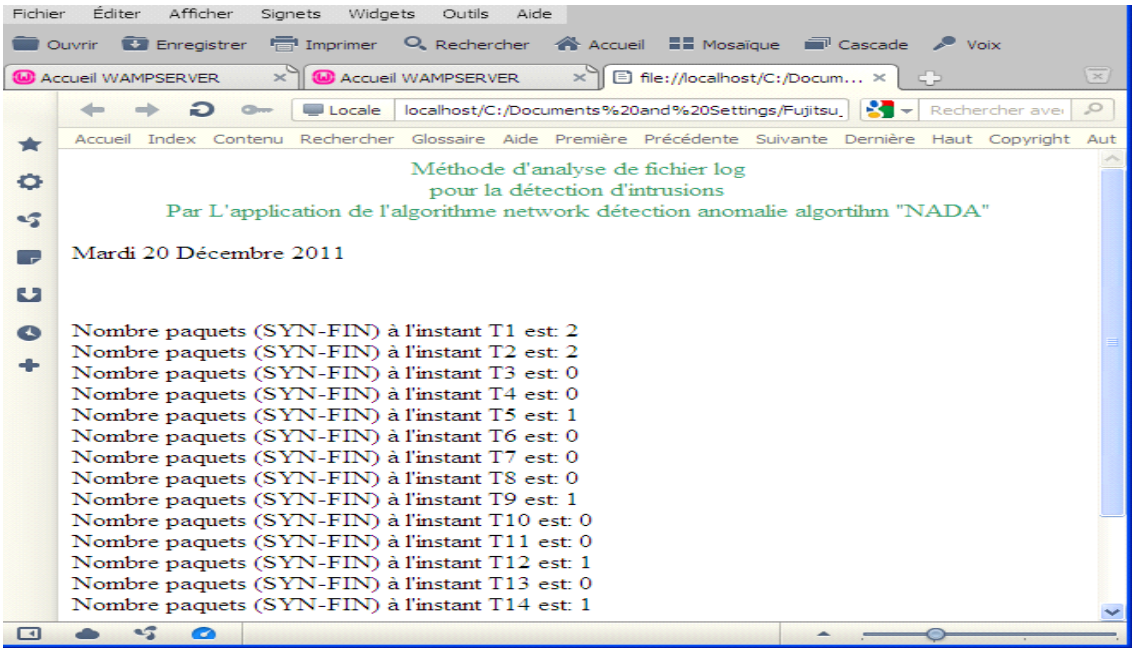

Fig.27-a exécution NADA

Ecran 02:

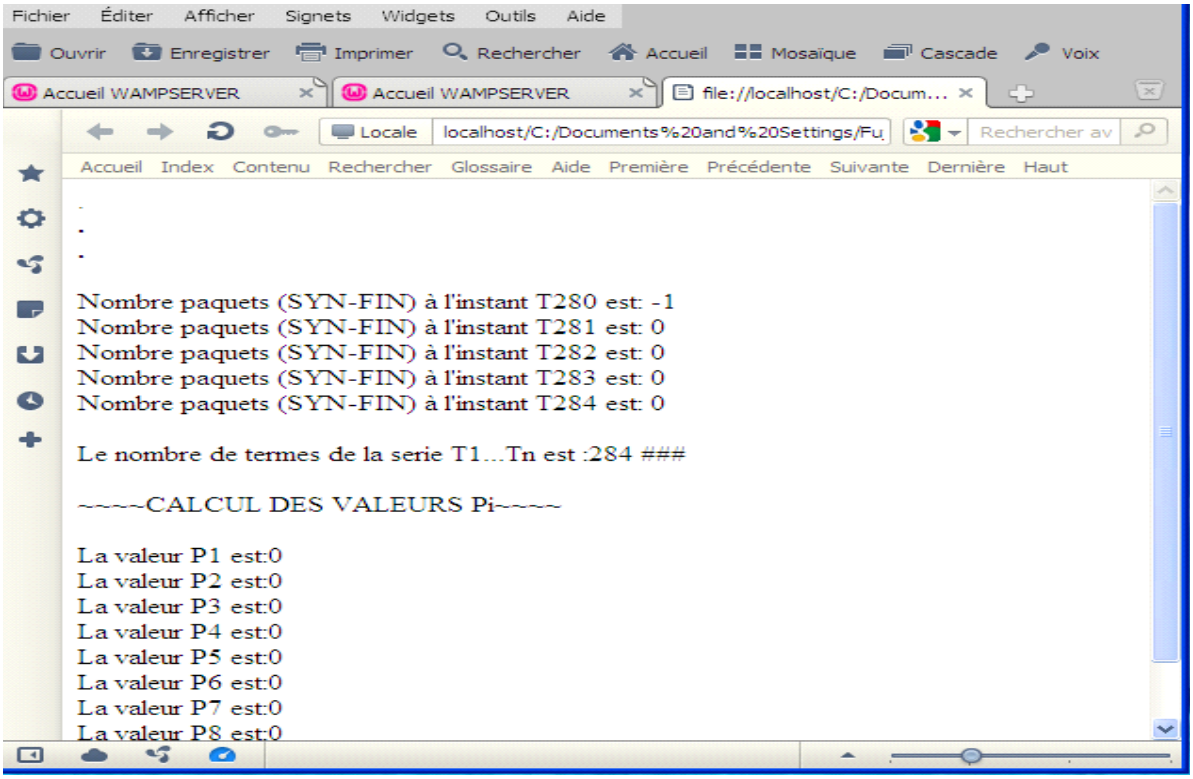

Fig.27.b exécution NADA

Ecran 03:

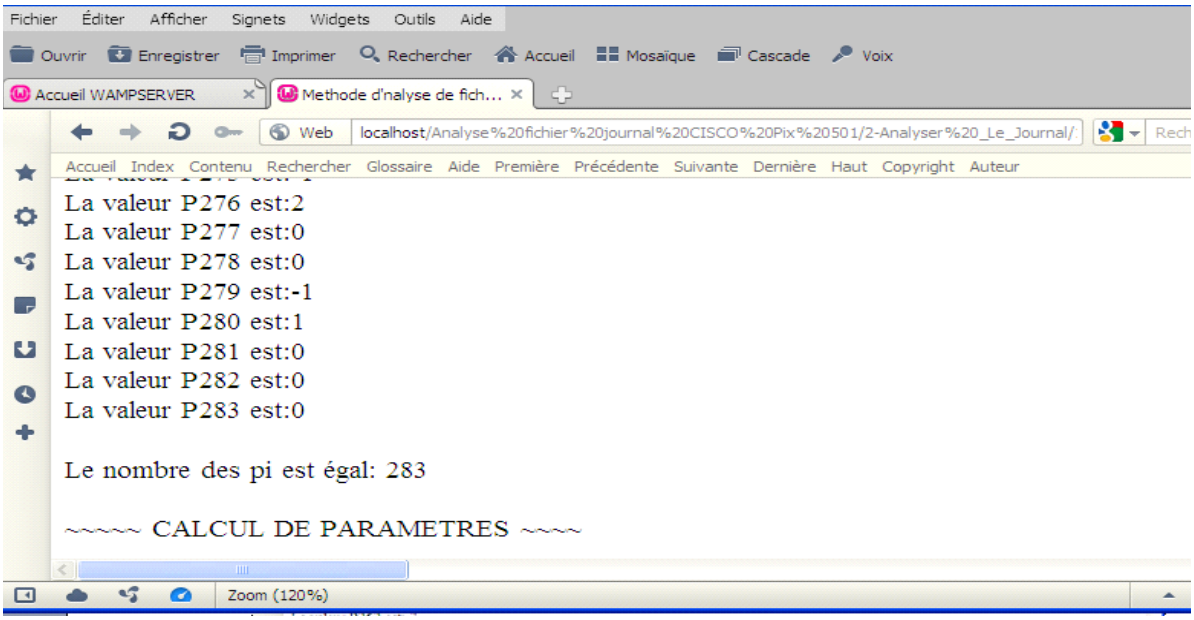

Fig.27.c exécution NADA

Ecran 04:

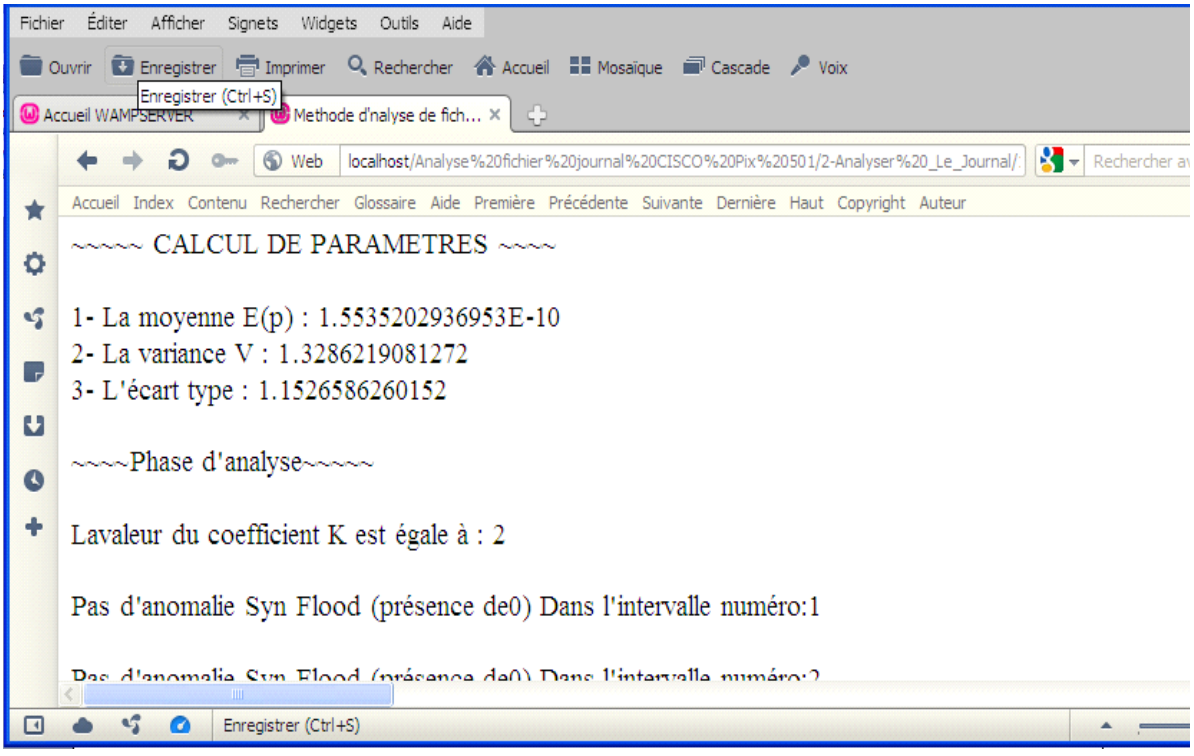

Fig.27.d exécution NADA

Ecran 05:

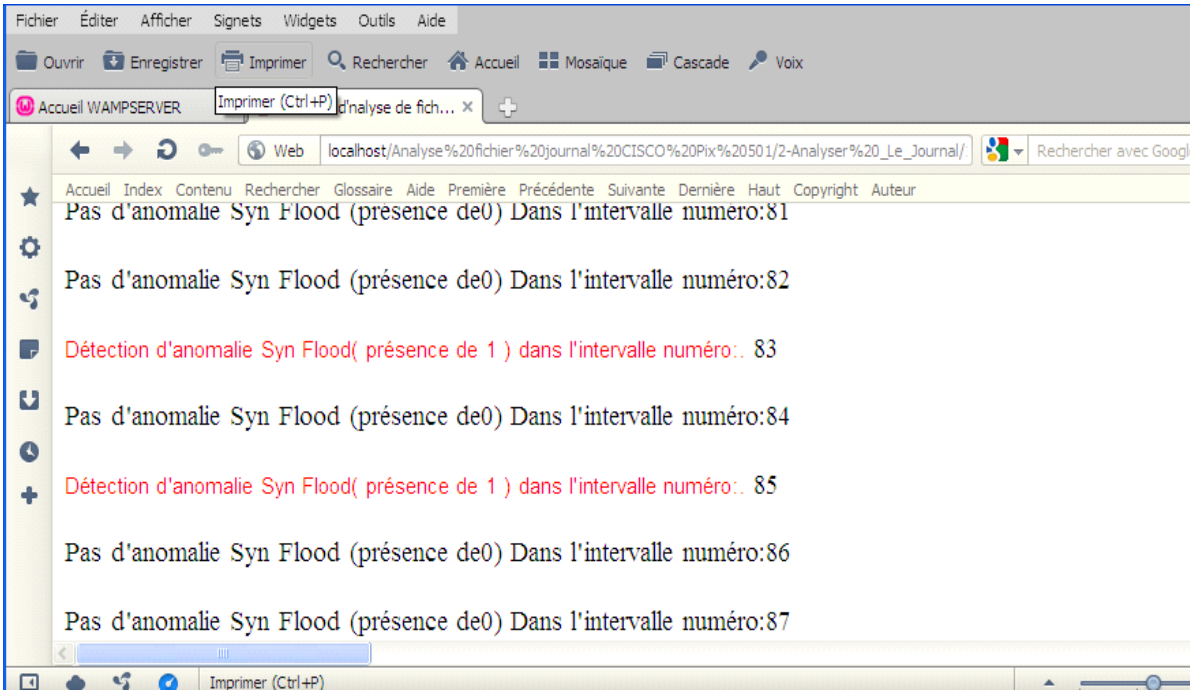

Fig.27.e exécution NADA

Ecran 06:

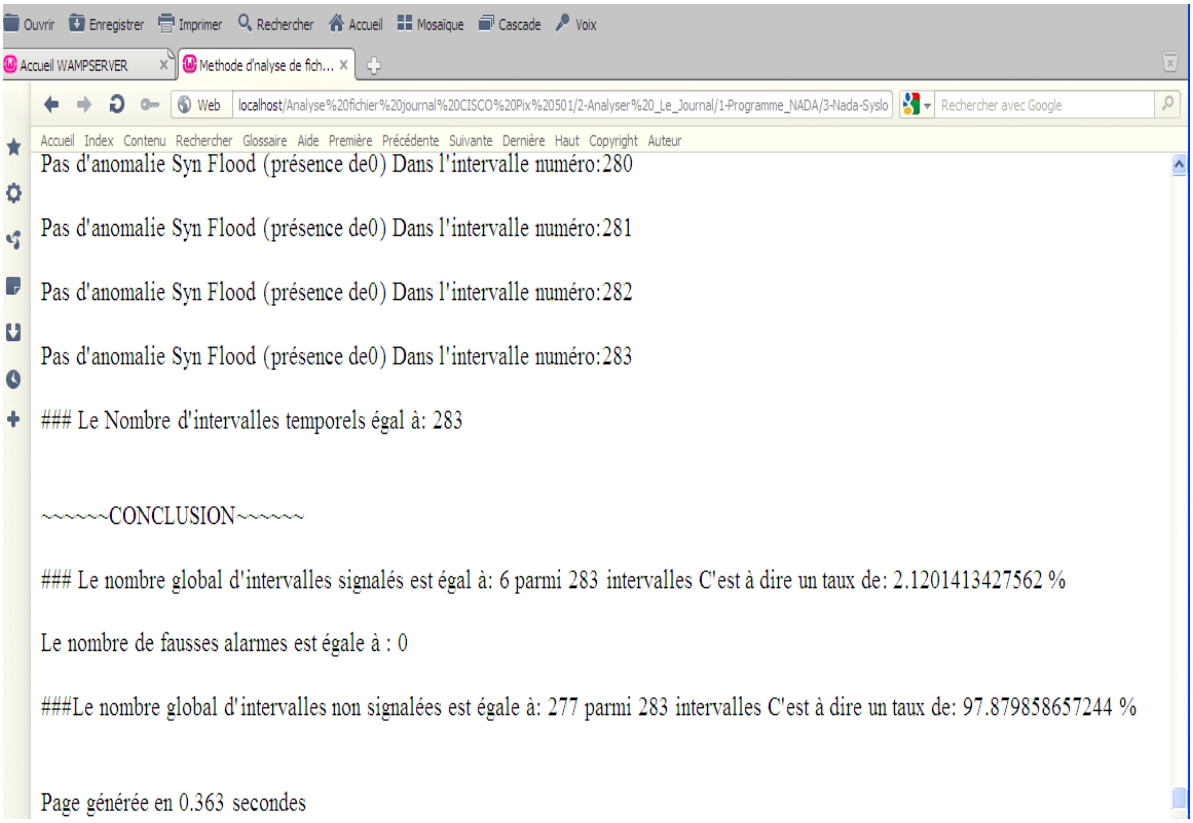

Fig.27.f exécution NADA

D'après l'exécution présentée dans les écrans précédents on peut montrer graphiquement que le nombre de vraies alertes égal à 6, pour une valeur du coefficient K=2. Donc pour cette valeur le nombre de fausses alertes devient nulle.

Le graphe ci-dessous **[Fig.28]** explique réellement l'évaluation des vraies alertes détectées par l'algorithme NADA en fonction des valeurs calculées P_i telle que :

$P_i = (x_i) - (x_{i-1})$ et l'expression de x_i est de :

x_i = (nombre de paquets (Syn)- nombre de paquets (FIN)) / Δt.

Les valeurs calculées pendant la durée globale d'observation (283 Secondes) des paramètres E(p), V(p) et de l'écart type σ sont comme ci-dessous:

E(p) = 1.5535202936953E-10

V(p)= 1.3286219081272

σ = 1.1526586260152

Pour K=2 ; E(p) + K σ = 2,30000000017825

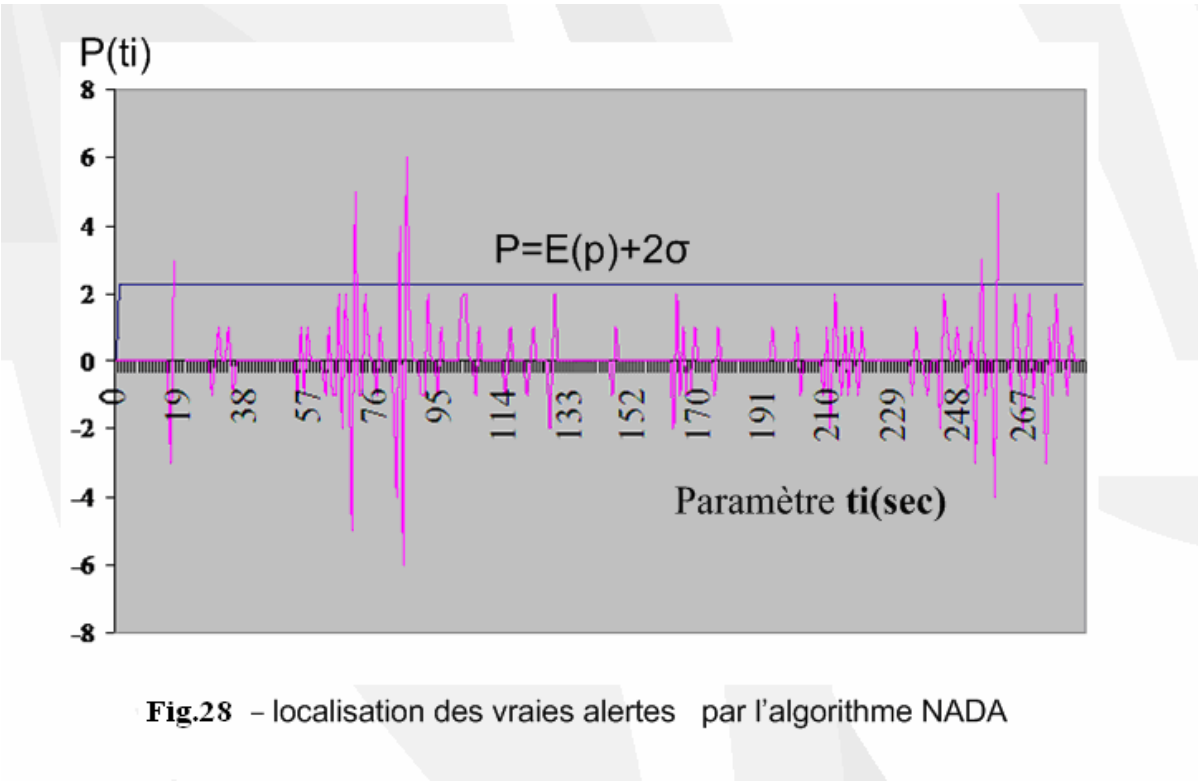

Fig.28 – localisation des vraies alertes par l'algorithme NADA

Remarquons sur le graphe ci-dessus que les piques de l'attaque TCP /SYN Flooding est de faible densité, 06 PIQUES aux instants : t=17s, t=70s, t=83s, t=85s, t=254s, t=259s

C'est-à-dire un taux très faible de 2,12 %. Ce qui nous conduit à dire que notre fichier log de test ne présente pas d'attaque TCP/ SYN Flooding.

IV.3.2.2 Analyse par algorithme Cusum :

Dans la phase de pré analyse on segmente le fichier en intervalle temporels de durée d'une seconde les résultats obtenus en fonction du choix de la valeur du seuil **S** sont comme suit:

S	**Nombre faux positifs**	**Taux faux positifs**	**Nombre faux négatifs**	**Taux faux négatifs**
0	54	19.08%	-	-
0.1	9	3.18%	-	-
0,2	6	2.12%	-	-
0.3	3	1.06%	-	-
0.4	3	1.06%	-	-
0.75	0	0%	-	-
0,85	-	-	1	0.35%
0.95	-	-	2	0.70%
1	-	-	3	1.06%
1,75	-	-	4	1,41%
1,95	-	-	5	1,76%
>=2	-	-	6	2,12

Tableau de test CUSUM en fonction des valeurs réelle S

Remarquons que pour S=0,75, 06 intervalles parmi 283 intervalles présentent l'attaque TCP/ Syn flooding (06 alertes d'attaque) c'est-à-dire un taux très faible de 2.112676056338 %

En résume les résultats du tableau ci-dessus dans le graphe suivant ci-dessous **(Fig.29):**

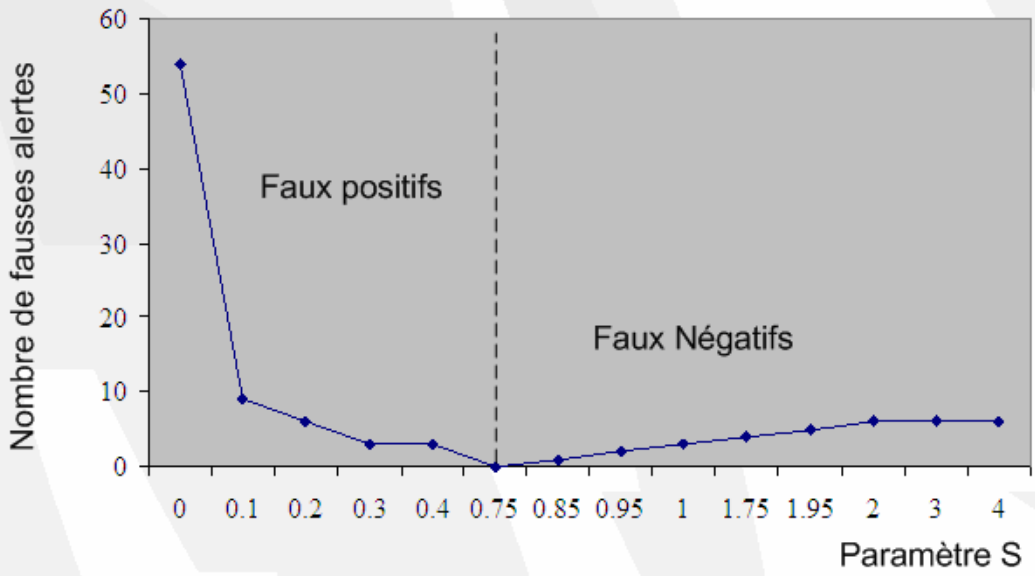

Fig.29 – Performance de l'algorithme CUSUM

L'augmentation du coefficient **k dans le cas de l'algorithme Nada conduit** à un grand nombre de faux négatifs tandis que dans le sens inverse il conduit à des faux positifs Pour un minimum de fausses alertes la valeur de S=0,75 produit un pourcentage de 2.11 % et par une simple comparaison avec les résultats générés par le programme NADA(2.12 %, K=2,) **[10,15]**; pour le même fichier journal , S=0,75 représente la valeur de configuration pour l'algorithme CUSUM.

Exécution du programme Cusum pour **S=0,75**

Ecran:01

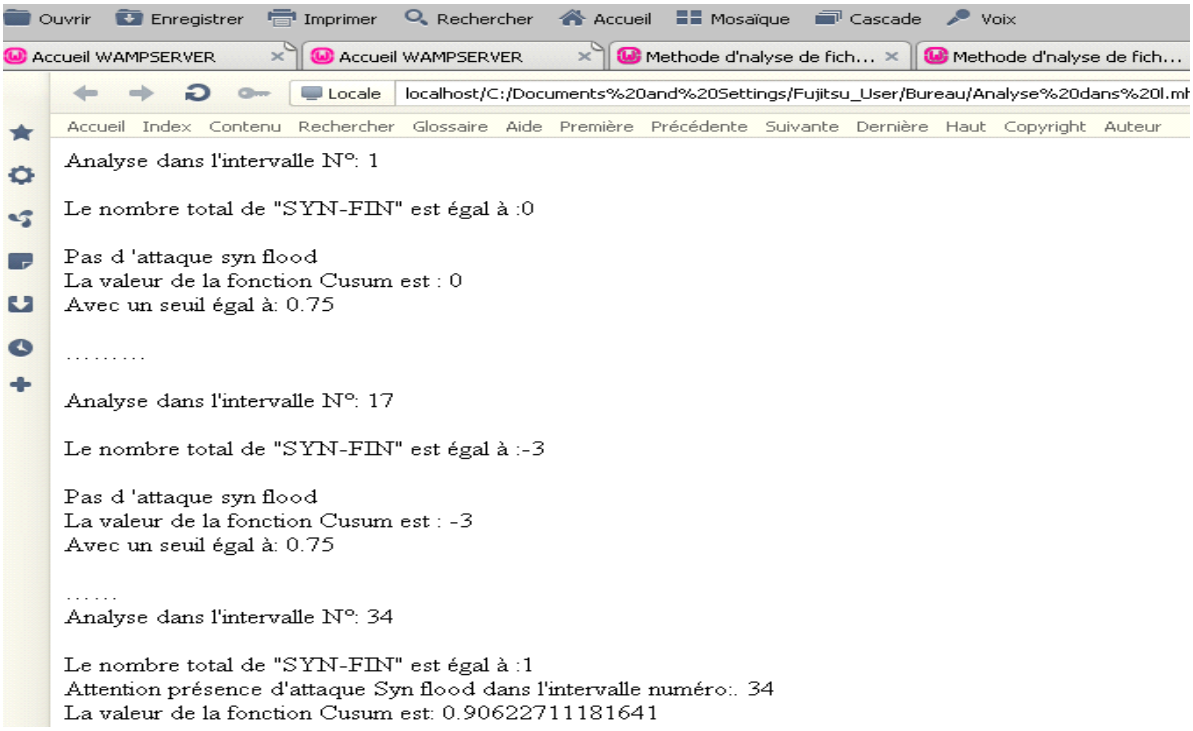

Fig.30 a exécution de l'algorithme CUSUM

Ecran: 02

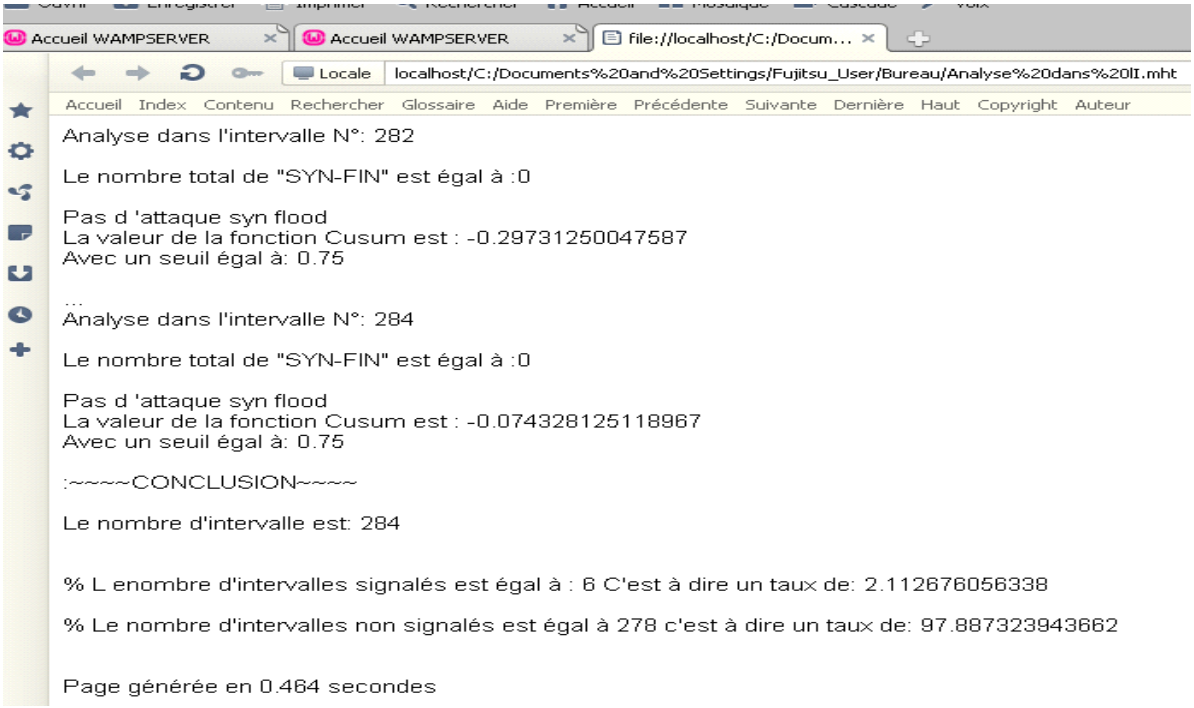

Fig.30 b exécution de l'algorithme CUSUM

L'utilisation de l'algorithme CUSUM dans un réseau à grand échelle qui fait circuler plusieurs dizaines de milliers de connexions devient un problème difficile à résoudre au vu de la quantité massive de données à traiter **[11]**. Donc la nécessité d'avoir un contrôle humain pour gérer l'attaque détectée.

L'algorithme NADA est complètement général. Il peut travailler sur n'importe quelle série temporelle produite à partir du trafic entrant (ou d'une trace de trafic enregistrée). Soit séparément ou simultanément NADA possède un paramètre important qui influe sur les performances de l'algorithme : le paramètre de seuillage k. Pour cela, nous avons fait varier k dans l'intervalle [0,6]. Pour chaque valeur de k, NADA a été évaluée pour toutes la trace de notre base. La courbe des performances de NADA montre en particulier que le taux de bonne détection dépasse toujours largement le taux de fausses alarmes. Il indique également que la valeur k = 2 est une bonne valeur pour configurer NADA **[17].**

Conclusion générale

Conclusion générale

Dans le cadre du travail décrit dans ce mémoire, nous voulions détecter les attaques de déni de service de type SYN flooding afin de sécuriser le réseau contre ce type d'attaques. La conception et le développement d'une technique de détection de ce type d'attaque par l'implémentation de deux algorithmes NADA et CUSUM représente le premier pas dans l'avancement de notre projet. Les deux algorithmes ont été implémentés de manière à détecter ce type d'attaque par analyse d'un fichier journal généré par le firewall placé à l'entrée du réseau LAN surveillé. Les résultats de test de ce système sur notre trace, sont satisfaisants. Cette satisfaction, ne signifie pas que notre système est parfait. Ce travail nous a beaucoup apporté dans le domaine de la sécurité des réseaux, l'application que nous avons élaborée présente des avantages comme la détection rapide d'anomalie Syn Flood ainsi qu'un taux de fausses alertes limité.

En continuité de ce travail, notre recherche nous à permis de découvrir que les capacités de l'algorithme NADA dépassent la détection d'attaques et permet aussi de faire la **classification** et de **l'identification** des anomalies [**17].** Plusieurs pistes de travaux futurs peuvent être envisagées. On pense notamment à la possibilité de compléter la méthode par la détection d'autres types d'anomalies en utilisant de nouvelles métriques, de développer une application d'écoute du trafic réseau en gardant les mêmes algorithmes conçus dans ce projet, pour une analyse en temps réel afin d'améliorer le processus d'analyse.

Malgré les recherches qui ont été déjà menées ces dernières années, aucune recherche n'a abouti à un système de détection d'intrusion permettant de garantir une sécurité totale et la recherche reste ouverte.

Références bibliographiques

[01]. DEVEZE BENJAMIN FOUQUIN MATHIEU. Article sur I.D.S. Systèmes de détection d'intrusion - Link Analysis Juillet 2004

[02]. Frédéric Majorczyk. Détection d'intrusions comportementale par diversification de COTS : application au cas des serveurs web". Thèse pour l'obtention de grade de DOCTEUR DE L'UNIVERSITÉ DE RENNES **1** Mention : INFORMATIQUE. Soutenue le 3 décembre **2008**

[03]. Hervé Debar, Monique Becker, and Didier Siboni. A neural network component for an intrusion detection system. In Proceedings of the IEEE Symposium of Research in Computer Security and Privacy, pages 240–250, Oakland, CA, May **1992.**

[04]. Stephanie Forrest, Steven A. Hofmeyr, Anil Somayaji, and Thomas A. Longstaff. A sense of self for unix processes. In Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy, pages 120–128. IEEE Computer Society, IEEE Computer Society Press, May **1996.**

[05]. Guillaume TOURNAND - ingé 2. Article sur Les attaques réseau de type Denial Of Service (D.O.S.).Ecole supérieure d'informatiques paris. Projet pérennisation 2002

[06]. Ludovic Mé- Véronique Alanou. Rapport sur la détection d'intrusion dans un système informatique : méthodes et outils. Synthèse. SUPELEC BP 2835511 Cesson Sévigné Cedex

[07]. Cristina Abad, Jed Taylor, Cigdem Sengul, William Yurcik, Log Correlation for Intrusion Detection: A Proof of Concept Department of Computer Science, University of Illinois at Urbana-Champaign, National Center for Supercomputing Applications(NCSA), Science Applications International Corporation (SAIC), Spring 2003.

[08]. Configuration Guide for the Cisco Secure PIX Firewall Version 5.1 chapter II

[09]. Claude Duvallet. Article sur les systèmes de détection d'intrusion réseau. Université du Havre UFR Sciences et Techniques.

[10]. Anis Ayachi et Abdelkarim Dhifelleh. OUTIL DE DETECTION D'ANOMALIES DANS UN RESEAU IP. Thèse pour l'obtention de grade de MASTER DE L'UNIVERSITÉ PARIS Descartes. Encadré par M. Osman SALEM. **2008-2009.**

[11].Radoslava Tatarovo et Gaetano Giarmana .Détection des attaques de Déni de Service dans les réseaux IP. Thèse pour l'obtention de grade de MASTER DE L'UNIVERSITÉ PARIS Descartes. Encadré par M. Osman SALEM. **2009-2010.**

[12]. Laurent Toutain. Réseaux locaux et Internet des protocoles à l'interconnexion EDITION HERMES PARIS 1996.

[13]. http://www.wampserver.com/

[14]. David Burgermeister, Jonathan Krier. Article sur les systèmes de détection d'intrusions. Date de publication : 22/07/2006.

[15]. Silvia dos Santos Farraposo. Contributions on Detection and Classification of Internet Traffic Anomalies. THÈSE En vue de l'obtention du DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE, présentée Le 17 Juin 2009.

[16]. Hocine hamdani. Livre sur les STATISTIQUE DESCRIPTIVE ET PROBABILITES-cours et exercices Office des Publications Universitaire O.P.U.1998.

[17]. Silvia Farraposo,. Philippe Owezarski, Edmundo Monteiro. Article sur la détection**,** classification et identification d'anomalies de trafic. School of Technology and Management - Polytechnic Institute of Leiria, Portugal. Publier le 11 Fev **2008**

[18]. Alexandre LUNG-YUT-FONG, Olivier CAPPÉ, Céline LÉVY-LEDUC, François ROUEFF Article sur la Détection et localisation décentralisées d'anomalies dans le trafic internet. Institut Télécom & CNRS, Télécom ParisTech, LTCI.

[19]. Michel Grabisch et Patrice Perny. Article sur l'agrégation Multicritère. Publier le 12 mars 2002.

Annexes

Annexe

Configuration de base de notre firewall Cisco pix 501

User Access Verification

Password:

Password:

Password:

Type help or '?' for a list of available commands.

Plate-forme-test> en

Password: ********

Plate-forme-test# SH CONF

: Saved

: Written by enable_15 at 13:42:52.130 UTC Wed Jun 15 2011

PIX Version 6.3(3)

Interface ethernet0 auto

Interface ethernet1 100full

Nameif ethernet0 outside security0

Nameif ethernet1 inside security100

enable password iUrV.q29jXAtQYth encrypted

Passwd 2KFQnbNIdI.2KYOU encrypted

Hostname plate-forme-test

Domain-name cnis.dz

Fixup protocol dns

Fixup protocol ftp 21

Fixup protocol h323 h225 1720

Fixup protocol h323 ras 1718-1719

Fixup protocol http 80

Fixup protocol rsh 514

Fixup protocol rtsp 554

Fixup protocol sip 5060

Fixup protocol sip udp 5060

Fixup protocol skinny 2000

Fixup protocol smtp 25

Fixup protocol sqlnet 1521

Fixup protocol tftp 69

Names

Pager lines 24

Logging on

Logging timestamp

Logging trap debugging

Logging host inside 10.20.1.1 format emblem

mtu outside 1500

mtu inside 1500

ip address outside dhcp setroute

ip address inside 10.20.1.230 255.255.0.0

ip verify reverse-path interface outside

ip audit name sig_info info action alarm

ip audit name sig_attack attack action alarm drop reset

ip audit interface outside sig_info

ip audit interface outside sig_attack

ip audit info action alarm

ip audit attack action alarm

pdm location 10.20.0.0 255.255.0.0 inside

pdm logging informational 100

pdm history enable

arp timeout 14400

global (outside) 1 interface

nat (inside) 1 10.20.0.0 255.255.0.0 0 0

Conduit permit udp any any

Conduit permit tcp any any

route outside 0.0.0.0 0.0.0.0 172.26.0.1 1

timeout xlate 0:05:00

timeout conn 1:00:00 half-closed 0:00:00 udp 0:02:00 rpc 0:10:00 h225 1:00:00

timeout h323 0:05:00 mgcp 0:05:00 sip 0:30:00 sip_media 0:02:00

timeout uauth 0:05:00 absolute

aaa-server TACACS+ protocol tacacs+

aaa-server RADIUS protocol radius

aaa-server LOCAL protocol local

Http server enable

Http 10.20.0.0 255.255.0.0 inside

No snmp-server location

no snmp-server contact

snmp-server community public

No snmp-server enables traps

Telnet 10.20.1.1 255.255.255.255 inside

Telnet 10.20.1.2 255.255.255.255 inside

Telnet timeout 5

Ssh timeout 5

Console timeout 0

Dhcpd address 10.20.1.1-10.20.1.32 inside

dhcpd lease 3600

Dhcpd ping_timeout 750

Dhcpd auto_config outside

Terminal width 80

Cryptochecksum: 7fc34d5c76416275cb5c0aa069f6ef9f

Plate-forme-test#

Explication de quelques commandes utilisées:

global (outside) 1 interface

nat (inside) 1 10.20.0.0 255.255.0.0

Dans cet exemple NAT_ID=1

Permettre à n'importe quelle machine dans le segment: (1**0.20.0.0**) venant de l'interface interne (**Inside**) de sortir sur l'Internet (**outside**) en utilisant l'IP publique**, g**énérer par le DHCP du gatway

Global qui permet en fait de spécifier la ou les adresses IP publiques (ou rarement privées) qui vont être utilisées pour sortir et donc qui seront vues de l'extérieur,

Interface signifie qu'on fera du PAT (port address translation) en utilisant l'IP de l'interface externe.