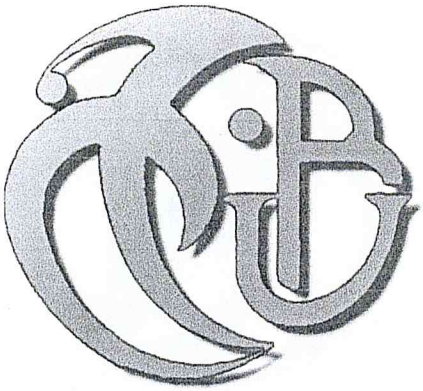
République Algérienne démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université SAAD DAHLAB de BLIDA
Faculté de science
Département informatique



MÉMOIRE DE MASTER INFORMATIQUE
SPÉCIALITÉ SYSTÈME INFORMATIQUE ET RÉSEAU

Interception et reconnaissance des attaques DOS avec IDS SNORT

Auteur :
OUAHABI LARBI

Promoteur :
Mme.CHERGUI NADJAH

Encadreur :
Mr.YALAOUI MOUSSA

Président : Geuessoum. D

septembre 2018

Résumé

Résumé :

De nos jours, les attaques envers les systèmes informatiques sont devenues nombreuses, plus spécifiques, puissantes, intelligentes et causent des dégâts considérables.
L'objectif de ce projet est de présenter le concept d'IDS (Intrusion Detection System) en générale et en particulier Snort, et d'analyser les paquets des attaques avec Wireshark pour extraire leurs signatures, ainsi de proposer nos propre règles de détection afin de pouvoir détecter les différents attaques (DOS/Ping of death, HTTP,TCP, UDP) à base des signatures d'attaques obtenuesdes différents tests effectués.

Mots clés :attaques ;analyser les paquets ; Wireshark ; IDS ; Snort ; règles.

Abstract :

Today, attacks on computer systems have become numerous, more specific, powerful, intelligent and causing considerable damage.
The aim of this thesis is to present the concept of IDS (Intrusion Detection System) in general and in particular Snort. And to analyze the packets of attacks with Wireshark to extract their signatures, and to propose our own detection rules in order to detect the different attacks (DOS / Ping of death, HTTP, TCP, UDP) based on the signatures of attacks obtained from the various tests carried out.

Keywords : Attacks ; analyze packets ; Wireshark ; IDS ; Snort ; rules.

Remerciements

Je remercie en premier lieu à Dieu le tout puissant pour la volonté, la santé, le courage et la patience qui nous a donné pour mener ce travail à terme.

Je remercie mes très chers parents, qui ont toujours été là pour moi.

Je tiens à exprimer toute ma reconnaissance à Dr MEHDI Merouane Directeur de centre des systèmes et réseaux d'information et de communication. Je tiens à remercier pour la soutien inconditionnel et pour l'encouragement.

Je remercie très spécialement mon encadreur Mr YALAOUI MOUSSA et promotrice madame NADJAH CHERGUI et madame IMENE NADIR . Je les remercie de m'avoir encadré, orienté, aidé et conseillé.

Je voudrais exprimer ma reconnaissance envers les amis et mes collègues de travail qui m'ont apporté leur support moral et intellectuel tout au long de ma démarche.

Table des matières

Introduction générale 1

1 Généralité sur les attaques et les outils de protections informatique 3

1.1 Introduction 3

1.2 Définition de la sécurité informatique 4

1.3 Les attaques informatiques 4

1.3.1 Définition de l'attaque informatique 4

1.3.2 Les Technique d'attaques 5

1.3.3 Les malwares 5

1.3.4 Les attaques Déni de Service 8

1.3.4.1 Les conséquences 9

1.3.4.2 Les différentes attaques DoS 9

1.3.5 Les attaques distribuées 13

1.4 les outils de protection 15

1.4.1 Formation des utilisations 15

1.4.2 Authentification et cryptage 16

1.4.2.1 Une authentification 16

1.4.2.2 Le cryptage 16

1.4.3 Anti-virus 16

1.4.4 Firewall (pare-feu) 17

1.4.5 Les systèmes de détection d'intrusion IDS 17

1.4.6 Système de prévention d'intrusion IPS 18

1.5 Conclusion 19

2 Les Systèmes de détection d'intrusion IDS 20

2.1 Introduction 20

2.2 Définition 21
2.3 Présentation d'un système de détection d'intrusions 22
2.3.1 Architecture d'un IDS 22
2.3.2 Les différents éléments de l'architecture IDWG 23
2.4 Vocabulaire de la détection d'intrusions 24
2.5 Emplacement d'un système de détection d'intrusions 24
2.6 Classification des systèmes de détection d'intrusions 25
2.6.1 Source des données à analyser 26
2.6.1.1 Source d'information système 26
2.6.1.2 Source d'information réseau 27
2.6.1.3 Source d'information applicative 27
2.6.1.4 Source d'information basée IDS 28
2.6.2 Localisation de l'analyse des données 28
2.6.3 Fréquence de l'analyse 29
2.6.4 Comportement après détection 29
2.6.5 Méthode de détection 30
2.6.5.1 L'approche comportementale 30
2.7 Les différentes sortes d'IDS 33
2.7.1 La détection d'intrusions basée sur l'hôte 33
2.7.2 Détection d'Intrusions basée sur une application 35
2.7.3 La Détection d'Intrusions Réseau 35
2.7.4 Les IDS hybrides 36
2.8 Les principales tâches d'un IDS 37
2.9 les limites d'un IDS 37
2.10 Quelques outils de détection d'intrusions 38
2.11 Conclusions 40

3 Les Systèmes de détection d'intrusion snort 41
3.1 Introduction 41
3.2 Snort 41
3.2.1 Positionnement de Snort dans le réseau 42
3.2.2 Architecture de Snort 43
3.3 Paramétrage de SNORT 44
3.3.1 Préprocesseurs 44

3.3.1.1 Frag3 . . . 44
3.3.1.2 Configuration globale . . . 45
3.3.1.3 Configuration du moteur . . . 45
3.3.1.4 Format . . . 45
3.3.1.5 Le préprocesseur stream . . . 46
3.3.1.6 preprocessor stream5_tcp . . . 47
3.3.2 Les plugins de sortie . . . 48
3.3.2.1 Alerte syslog (Alert_syslog) . . . 49
3.3.2.2 Alerte rapide (Alert_fast) . . . 49
3.3.2.3 Alerte pleine (Alert_full) . . . 50
3.3.2.4 Alerte smb . . . 50
3.3.2.5 Alerte unixsock . . . 50
3.3.2.6 Log_tcpdump . . . 51
3.3.3 Les bases de SNORT . . . 51
3.3.3.1 Les inclusions . . . 51
3.3.3.2 Les variables . . . 52
3.3.4 Les entêtes de règle . . . 52
3.3.4.1 L'action de règle . . . 52
3.3.4.2 Les protocoles . . . 53
3.3.4.3 Les adresses IP . . . 53
3.3.4.4 Les numéros de port . . . 54
3.3.4.5 L'opérateur de direction . . . 55
3.3.4.6 Les règles activate/dynamic . . . 55
3.4 Simulation d'attaques : . . . 56
3.4.1 les outils d'attaques . . . 56
3.4.2 Taxonomie des outils d'attaque DoS et leur comparaison . . . 57
3.4.3 Visualisation des paquets TCP . . . 63
3.5 Conclusion . . . 65

4 Simulation des attaques DoS et règles proposées 66
4.1 Introduction . . . 66
4.2 Méthode de travail . . . 66
4.2.1 Méthode de la simulation d'attaque . . . 67
4.3 Les règles de détection des attaques . . . 68

4.4 Attaque ICMP 68
4.4.1 Visualisation des paquets ICMP 68
4.4.1.1 Visualisation des paquets ICMP (Ping) 68
4.4.1.2 Visualisation des paquets de ping of death 69
4.4.2 Détection de l'attaque Ping of death 71
4.4.3 Attaque TCP 73
4.4.3.1 Visualisation des paquets d'attaque TCP via LOIC 73
4.4.3.2 L'impact de l'attaque TCP 76
4.4.3.3 Détection de l'attaque TCP 77
4.5 Attaque UDP 79
4.5.1 Visualisation des paquetsUDP 79
4.5.1.1 Visualisation de paquet UDP normale 79
4.5.1.2 Visualisation des paquets d'attaque UDP 81
4.5.1.3 L'impact de l'attaque UDP 83
4.5.1.4 Détection de l'attaque UDP 84
4.6 Attaque http 85
4.6.1 Visualisation des paquets HTTP 85
4.6.1.1 Visualisation le paquet HTTP normale 85
4.6.1.2 Visualisationde paquet d'attaque HTTP 86
4.6.1.3 L'impact de l'attaque HTTP 88
4.6.2 Détection de l'attaque HTTP 88
4.7 Conclusion 90
Conclusion générale 91
Annexe 92

Table des figures

1.1 Le SYN flood [20] 11
1.2 Le PING flood [21] 11
1.3 Smurf [22] 12
1.4 DDOS attaque [23] 14
1.5 spoof [24] 15

2.1 Modèle générique de la détection d'intrusions proposé par l'IDWG [11] . . 22
2.2 Endroits typiques pour un système de détection d'intrusions [25] 25
2.3 Classification des systèmes de détection d'intrusions [11]. 26

3.1 Différentes positions possible de Snort dans un réseau informatique[26] . . 42
3.2 Architecture de SNORT[27] 43
3.3 Taxonomie des outils d'attaque DoS [28] 57
3.4 Capture de l'interface de LOIC 59
3.5 interface de l'analyseur Wireshark 61
3.6 Encapsulation d'un paquet UDP, zone centrale de l'analyseur 62
3.7 Encapsulation d'un paquet UDP, zone centrale de l'analyseur 63
3.8 La capture des paquets TCP normale 64
3.9 La capture détaillée de paquet TCP 64
3.10 graphe du flux de donné 65

4.1 schéma de simulation des attaques DoS 67
4.2 Les paquets envoyés par Le Ping 68
4.3 Paquets ICMP capturés par Wireshark 69
4.4 Le Ping of death 69
4.5 Le traffic Ping of death capturé par Wireshark 70
4.6 Détection de l'attaque Ping of death 72

4.7 La capture détaillée de la détection de ping of death 72
4.8 les paramètres d'une attaque TCP LOIC 73
4.9 les données capturées de l'attaque TCP par l'outil Wireshark 74
4.10 Capture détaillé d'un paquet TCP 74
4.11 Capture de message (en ascii) ennoyé par LOIC 75
4.12 Le nombre des paquet TCP pandans une attaque TCP 75
4.13 Capture del'utilisation du processeur et réseau de TCP normal 76
4.14 Capture de l'utilisation du processeur, réseau de l'attaque TCP 77
4.15 Détection de l'attaque TCP 78
4.16 informations détaillées de l'attaque TCP 79
4.17 Paquet UDP d'une vidéo de youtube 80
4.18 Flux de donnée d'une vidéo de youtube 80
4.19 L'attaque UDP 81
4.20 d'un paquet d'attaque UDP 81
4.21 Le flux de données de l'attaque UDP 82
4.22 L'utilisation du processeur et réseau avant l'attaque 83
4.23 L'utilisation du processeur et réseau après l'attaque UDP 83
4.24 La détection de l'attaque UDP 84
4.25 l'attaque UDP présenté par Snort 85
4.26 recherche sur google 85
4.27 Capture d'un paquet htttp 86
4.28 L'attaque HTTP 86
4.29 La requête de l'attaque HTTP 87
4.30 La requête HTTPavec plus de détailles 87
4.31 La détection de l'attaque HTTP 89
4.32 Les informations détaillé de l'alerte d'attaque HTTP 89

Liste des tableaux

3.1 Outils d'attaques 59

4.1 La déférence entre ping et ping of death 71
4.2 La déférence entre les deux cas de protocole TCP(les signatures) 76
4.3 La différence entre une connexion UDP normale et une attaque UDP de type DOS 83
4.4 La différence entre le cas normale de HTTP et de l'attaque HTTP 88

Introduction générale

La croissance de l'Internet et l'ouverture des systèmes ont fait que les attaques dans les réseaux informatiques soient de plus en plus nombreuses. Les vulnérabilités en matière de sécurité s'intensifient, d'une part au niveau de la conception des protocoles de communication ainsi qu'au niveau de leur implantation et d'autre part, les connaissances, les outils et les scripts pour lancer les attaques sont facilement disponibles et exploitables. D'où la nécessité d'un système de détection d'intrusions.
Cette technologie consiste à rechercher une suite de mots ou de paramètres caractérisant une attaque dans un flux de paquets. Les systèmes de détection d'intrusion sont devenus un composant essentiel et critique dans une architecture de sécurité informatique.

Problématique

Autre la mise en place d'un pare-feu et d'un système d'authentification, il est de nos jours nécessaire de mettre en place un système de détection d'intrusions Par ailleurs, bien que la base de signatures soit étendue, elle nécessite un travail constant de la part de l'administrateur qui doit les télécharger manuellement. Il n'y a pas de procédure de mise à jour automatiques. Et les règle doive être paramétrer par celui-ci et certaine ne sont pas reconnue par les base snort tel que les attaque DOS.

Objectif

Notre travail consiste à étudier et concevoir des règles de détection basée sur un détecteur des attaques open source (Snort), à savoir l'attaque « Ping of death, TCP, UDP, HTTP » basé sur une simulation de l'outil d'attaque LOIC, et l'utilisation de Wireshark pour capturer les paquets et les comparer avec les cas normaux pour extraire la signature de chaque attaque afin de créer nos propre règles de détection .

Le premier chapitre de ce mémoire présente des notions de base en sécurité informatique et les différents types d'attaque. et nous abordons plus en détails les attaques de déni de service (DOS) ainsi que l'ensemble des outils de protection existant .

Le second chapitre présente une description détaillée sur les systèmes de détection d'intrusions .

Le troisième chapitre présente la configuration des règles du Snort , ainsi une description des outils utilisés dans notre travail sachant que wireshark c'est un outil pour capturer le trafic et LOIC c'est pour simuler les attaques.

Dans le chapitre quatre, nous suivons une méthodologie qui nous permet d'écrire les règles de Snort pour protéger le système contre les attaques DOS, et simuler les règles pour vérifier leur efficacité.

Chapitre 1

Généralité sur les attaques et les outils de protections informatique

1.1 Introduction

Les réseaux informatiques sont de plus en plus développés, que ce soit chez les particuliers ou dans le domaine professionnel. Ce qui rend la sécurité informatique un problème majeur dans la gestion d'entreprise. La transmission d'informations sensibles et le désir de s'assurer de la sécurité de l'information (intégrité, confidentialité, disponibilité, La non répudiation), celles-ci est devenue un point primordial dans la mise en place d'une politique de sécurité informatique.

Mais si toutes ces innovations ont apporté de très nombreux avantages aux entreprises, elles sont accompagnées de nouveaux risques inhérents à ces nouvelles techniques, le piratage informatique. En effet, ces attaques sont de plus en plus nombreuses, efficaces et simple à mettre en œuvre. Elles sont utilisées pour l'espionnage industriel (vols d'informations confidentielles) ou simplement du parasitage (destruction de données numériques ou arrêt de service). Dans le cadre de notre projet, nous nous intéresserons aux attaques de types dénis de service, qui sont les plus rependues. Ce type d'attaque n'est pas dangereux pour les données numériques du réseau, mais a pour objectif de rendre indisponible un service proposé par une entreprise, par exemple un site de commerce en ligne ; une entreprise ne peut se permettre d'avoir son site indisponible pendant plusieurs heures ou jours.

Dans ce chapitre, nous présentons d'abord des notions de base en sécurité informatique et les différents types d'attaque. Ensuite, nous abordons plus en détails les attaques de déni de service (DOS),et l'ensemble des outils de protection existant dans le marché.

1.2 Définition de la sécurité informatique

La sécurité informatique est l'ensemble des moyens mis en œuvre pour minimiser la vulnérabilité d'un système contre des menaces accidentelles ou intentionnelles. Elle consiste à assurer une utilisation des ressources matérielles et/ou logicielles d'une organisation pour protéger l'information en préservant les quatre propriétés suivantes :

L'intégrité : consiste à garantir l'exactitude et la complétude de l'information. Autrement dit, les données sont bien celles qu'on croit être.

La confidentialité : consiste à assurer que uniquement les personnes autorisées aient accès aux ressources. Autrement dit, elle consiste à préserver la révélation non autorisée d'information sensible.

La disponibilité : consiste à s'assurer que l'accès à l'information est continuellement disponible aux utilisateurs autorisés pour maintenir le bon fonctionnement du système informatique.

Le non répudiation : consiste à garantir qu'une transaction ne peut être niée . Dans une communication, les émetteurs des messages ne peuvent pas nier qu'ils ont fait envoyer ces messages. De même, les destinataires de ces messages ne doivent pas pouvoir nier les avoir reçus. [1]

1.3 Les attaques informatiques

Il existe plusieurs types d'attaques très connus dans le monde informatique. Dans cette section, nous présentons les différents types d'attaque qui représentent un véritable risque pour les différentes infrastructures.

1.3.1 Définition de l'attaque informatique

Une attaque informatique est toute tentative de détruire, exposer, modifier, désactiver, voler ou obtenir un accès non autorisé ou toute utilisation non autorisée d'information,

logiciels, physique. Techniquement, Une attaque informatique est définie par l'exploitation d'une faille d'un système informatique (système d'exploitation, logiciel ou bien même de l'utilisateur) à des fins non connues par l'exploitant du système et généralement préjudiciables [2].

1.3.2 Les Technique d'attaques

On entend souvent aux informations qu'un nouveau virus circule. Mais ce n'est pas la seule menace pour nos ordinateurs. Il existe pleins de programmes malveillants. Les paragraphes suivants détaillent quelques-unes des principales menaces :

1.3.3 Les malwares

Un malware est un logiciel développé dans le but de nuire à un système informatique. Il existe plusieurs familles de malwares. On va définir les plus utilisés :

a - Les virus

Les virus sont des programmes malveillants qui ont pour but de se reproduire. Souvent, ils sont gênants pour l'utilisateur, puisqu'ils peuvent détruire des fichiers sur l'ordinateur [3]. Le virus classique est un morceau de programme, souvent écrit en assembleur, qui s'intègre dans un programme normal (le Cheval de Troie), le plus souvent à la fin, mais cela peut varier.
Chaque fois que l'utilisateur exécute ce programme « infecté », il active le virus qui en profite pour aller s'intégrer dans d'autres programmes exécutables. De plus, lorsqu'il contient une charge utile, il peut, après un certain temps (qui peut être très long) ou un événement particulier, exécuter une action prédéterminée.
Cette action peut aller d'un simple message anodin à la détérioration de certaines fonctions du système d'exploitation ou la détérioration de certains fichiers ou même la destruction complète de toutes les données de l'ordinateur. On parle dans ce cas de « bombe logique » et de « charge utile ».

Un virus de boot s'installe dans un des secteurs de boot d'un périphérique de démarrage, disque dur (le secteur de boot principal, le « Master boot record », ou celui d'une partition), disquette, ou autre. Il remplace un chargeur d'amorçage (ou programme de dé-

marrage ou « bootloader ») existant (en copiant l'original ailleurs) ou en crée un (sur un disque où il n'y en avait pas) mais ne modifie pas un programme comme un virus normal ; quand il remplace un programme de démarrage existant, il agit un peu comme un virus « prepender » (qui s'insère au début), mais le fait d'infecter aussi un périphérique vierge de tout logiciel de démarrage le distingue du virus classique, qui ne s'attaque jamais à « rien ».
Les macrovirus qui s'attaquent aux macros de logiciels de la suite Microsoft Office (Word, Excel, etc.) grâce au VBA de Microsoft. Par exemple, en s'intégrant dans le modèle normal.dot de Word, un virus peut être activé à chaque fois que l'utilisateur lance ce programme.

Les virus de type batch, apparu à l'époque où MS-DOS était le système d'exploitation en vogue, sont des virus « primitifs ». Bien que capables de se reproduire et d'infecter d'autres fichiers batch, ils sont lents et ont un pouvoir infectant très faible. Certains programmeurs ont été jusqu'à créé des virus batch cryptés et polymorphes, ce qui peut être qualifié de « prouesse technique » tant le langage batch est simple et primitif.

b - Les vers

Les vers sont des programmes qui se propagent d'un ordinateur à un autre ordinateur via un réseau comme l'internet. Contrairement aux virus, les vers n'ont pas besoin d'un programme hôte pour assurer leur reproduction. Ce qui leur permet de se propager à une vitesse impressionnante sur un réseau, et pouvant donc le saturer et d'espionner les ordinateurs où il se trouve afin d'offrir un accès dérobée aux pirates informatique [3].

- Vers de réseau
- Vers de courrier électronique
- Vers de messagerie instantanée
- Vers Internet
- Vers IRC (Internet Relay Chat)
- Vers de réseaux de partage de fichiers

c - Les spywares

Les spywares, ou logiciels espions, sont des logiciels nuisibles qui transmettent à des tiers des informations contenues dans les ordinateurs. Les spywares sont souvent présents

dans des logiciels gratuits (différents des logiciels libres), ou des partagiciels. En général, les logiciels à code source libre comme Mozilla Firefox n'en contiennent aucun [3].
Les principaux vecteurs d'infections sont :

- les logiciels de cassage de protection (type cracks et keygens).
- les faux codecs.
- certains logiciels gratuits (certaines barres d'outils ou utilitaires par exemple).
- les faux logiciels de sécurité (rogues) : Certains programmes soi-disant destinés à lutter contre les logiciels espions contiennent eux-mêmes ce type de menace, ou se révèlent totalement inefficaces avec pour seul but de facturer une licence d'utilisation (cas de Spyware Assassin par exemple).
- la navigation sur des sites douteux, notamment ceux au contenu illégal.
- les pièces jointes et les vers par messagerie instantanée.

d - Le spamming

Le spamming (ou encore pourriel, courrier rebut) consiste à envoyer des messages appelés "spam" à une ou plusieurs personnes. Ces spam sont souvent d'ordre publicitaire. Tous les points suivants sont considérés comme du spamming.

- Envoyer un même mail une ou plusieurs fois à une ou plusieurs personnes en faisant de la publicité.
- Poster un ou plusieurs messages dans un forum qui n'a rien à voir avec le thème.
- Faire apparaître un message publicitaire lorsque l'on navigue sur un site [3].

e - Cheval de Troie

C'est un programme ou un code malveillant intégré à une application par l'ajout ou par la modification de son code lors de l'exécution de ce programme. Le bout de code malveillant pourra exécuter des commandes spécifiques (récupération de fichiers de mot de passe, etc.) à l'insu de l'utilisateur reposant sur une porte dérobé « backdoor ». Les origines fréquentes des chevaux de Troie sont :

- Téléchargement de versions trafiquées sur des sites non officiels ou des plateformes peu sûres (P2P). Télécharger les logiciels sur le site officiel de l'auteur ou du distributeur évite normalement d'avoir affaire à une version infectée par un cheval de

trois. Cela n'est évidemment pas possible pour se procurer des versions crackées, mais faisable pour tous les logiciels gratuits.

— Téléchargement de programmes P2P.
— Visite de sites Web contenant un exécutable (par exemple les contrôles ActiveX ou des applications Java).
— Exploitation de failles dans des applications obsolètes (navigateurs, lecteurs multimédias, clients de messagerie instantanée) et notamment les Web Exploit.
— Ingénierie sociale (par exemple, un pirate envoie directement le cheval de Troie à la victime par messagerie instantanée).
— Pièces jointes et fichiers envoyés par messagerie instantanée.
— Connexion d'un ordinateur à un périphérique externe infecté.
— Mise à jour de logiciel.
— Absence de logiciel de protection.

1.3.4 Les attaques Déni de Service

Les attaques par déni de services (Denial of Service en anglais ou DoS) sont des attaques qui visent à rendre les services ou ressources d'une organisation indisponibles durant une certaine période. Généralement, ce type d'attaque à lieu contre des machines, serveurs et accès d'une entreprise afin qu'ils deviennent inaccessibles pour leurs clients. Le but d'une telle attaque n'est pas d'altérer ou de supprimer des données, ni même de voler quelconque information. Il s'agit ici de nuire à la réputation de sociétés présentes sur Internet en empêchant le bon fonctionnement de leurs activités.
En apparence, une telle attaque peut sembler inoffensive si elle vise un réseau ou un ordinateur particulier, mais elle peut s'avérer redoutable lorsqu'elle vise un serveur ou des ressources matérielles appartenant à une grande société dépendante de son infrastructure réseau.

Ce genre d'attaque est très répandu sur les réseaux ; car elle est assez simple à mettre en œuvre, mais il peut néanmoins avoir des conséquences désastreuses. De plus, la détection et la prévention de ces attaques sont très difficiles, car elles peuvent prendre des formes très variées. Quasiment tous les systèmes informatiques sont vulnérables et même des équipements coutant des milliers de dollar ne peuvent parfois rien faire contre de telles attaques. C'est pour ces raisons que les dénis de service sont utilisés dans de multiples situations et par des personnes très variées.

Tous ces équipements sont critiques, la surcharge d'un routeur entraînera un retard important des décisions de routage voire une impossibilité totale de router les paquets sur un réseau le rendant totalement inefficace à communiquer avec d'autres. Les surcharges de serveurs empêcheront les utilisateurs d'accéder aux services tels que les mails, ou l'internet, ils subiront des délais d'attente considérables et seront même parfois dans l'impossibilité de joindre leur ressource.

Le principe général des attaques DoS consiste à envoyer des données ou des paquets dont la taille ou le contenu est inhabituel, ceci a pour effet de provoquer des réactions inattendues du réseau ou de la machine cible, pouvant aller jusqu'a l'interruption du service.[3]

1.3.4.1 Les conséquences

Une attaque par DoS peut avoir de nombreuses formes qui engendrent chacune de nombreuses conséquences, en représentant une palette de risques très variée. Les attaques les plus dévastatrices peuvent amener, que ce soit de manière directe ou indirecte, à des pertes d'argents colossales pour une société dont la principale activité est basée sur un flux d'informations Internet. Les attaques contres celui-ci peuvent amener à des pertes d'argent colossales.

1.3.4.2 Les différentes attaques DoS

La réalisation d'un DOS déni de service n'est pas très compliquée, mais pas moins efficace. Il est possible d'attaquer tout type d'équipements réseau tel que les serveurs, routeurs et Switchs. Les attaques de type DoS prennent de multiples formes et utilisent de nombreuses méthodes pour mettre hors service une ressource réseau .Dans ce qui suit, nous présentons de manière non exhaustive les différentes attaques de type déni de service distribué les plus connues et répandues.

a- Les attaques par surcharge

Une des méthodes les plus répandues et une des plus simples à mettre en œuvre est de surcharger complètement la cible de requêtes de toutes sortes. On distingue quatre grands types d'attaques utilisant différents protocoles et couches réseaux.

b- Le SYN flood

est une forme d'attaque informatique visant à provoquer un déni de services, elle est destinée à rendre un réseau complètement indisponible. Cette technique d'attaque s'applique dans le cadre d'un protocole TCP (Transmission Control Protocol) et vise principalement à submerger le serveur cible d'une tonne de requêtes SYN (Synchronized).

Le principe de fonctionnement d'une connexion TCP :

Normalement, lorsqu'une connexion TCP est initialisée entre le serveur et un client sans la moindre intention de nuire, un échange de message doit avoir lieu. Selon le principe du « three-way handshake », la connexion doit se dérouler en trois phases notamment : le SYN, le SYN-ACK et l'ACK. Le client qui souhaite se connecter avec le serveur doit d'abord, en effet, envoyer un premier paquet de SYN (Synchronized) au serveur. Ensuite pour répondre à cette requête, le serveur lui envoie un message SYN-ACK (Synchronized Acknowledgment), et le client doit enfin envoyer une réponse ACK (Acknowledgment) pour établir définitivement la connexion.

L'usage malveillant d'une connexion TCP :

Un client malintentionné peut brûler la troisième étape et ne pas répondre au serveur par un message ACK. Le serveur met ainsi du temps avant de libérer les ressources préalablement destinées au client et générer un temps d'attente.

Il peut également arriver qu'en refusant de répondre par un message ACK, le client malveillant profite de la troisième étape pour surcharger les ressources du serveur et de l'empêcher d'accepter de nouvelles requêtes, de manière à pouvoir provoquer un déni de services. Après le SYN ACK, la connexion est semi-ouverte et consomme d'importantes ressources (mémoire, temps, processeurs) et qu'en ce moment précis il est possible de surcharger les ressources du serveur cible en générant plusieurs requêtes incomplètes de ce type.

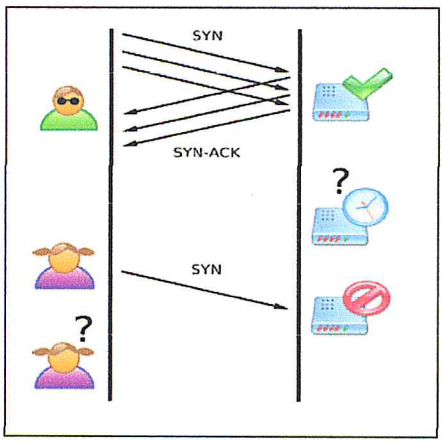


FIGURE 1.1 – Le SYN flood [20]

c- Le PING flood

une des attaques les plus simples à mettre en place. Elle consiste à simplement envoyer un nombre maximal de PING simultané jusqu' à saturer la victime. On utilise généralement la commande ping sous Linux mais une des conditions pour que l'attaque soit efficace est de posséder plus de bande passante que la victime.

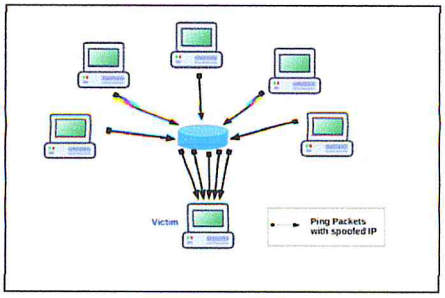


FIGURE 1.2 – Le PING flood [21]

d- Le Smurf

les attaques Smurf profitent d'une faiblesse d'IPv4 et d'une mauvaise configuration pour profiter des réseaux permettant l'envoi de paquets au broadcast. Le broadcast est une adresse IP qui permet de joindre toutes les machines d'un réseau. L'attaquant envoie au broadcast des paquets contenant l'IP source de la victime ainsi chaque machine sur le réseau va répondre à la cible à chaque requête de l'attaquant. On se sert ainsi du réseau

comme un amplificateur pour perpétrer l'attaque. Cette méthode porte aussi le nom d'attaque réfléchie permettant à l'attaquant de couvrir ces traces et de rendre l'attaque plus puissante.

Ces attaques peuvent être vraiment très efficaces, c'est pourquoi une association Smurf Amplifier Registry existe pour aider les opérateurs à détecter les réseaux mal configurés. Les individus particuliers peuvent aussi demander à leurs opérateurs de désactiver le broadcast dirigé pour éviter ce type de désagréments.

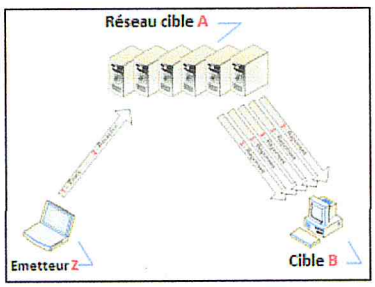


FIGURE 1.3 – Smurf [22]

e- Les attaques DOS par failles

Un autre moyen de réaliser un déni de service Dos consiste à exploiter les nombreuses failles présentées dans les systèmes d'informations. Au lieu de chercher à surcharger la cible, on va simplement la forcer à réagir de façon bien définie en lui soumettant des informations qu'elle ne peut gérer.
Les systèmes Microsoft Windows sont par exemple très vulnérables à ce genre d'attaques. De nombreux moyens existent afin de tromper le système pour l'exploiter. Néanmoins ces attaques sont de moins en moins nombreuses et de moins en moins efficaces car les systèmes d'exploitation actuels sont de plus en plus sécurisés.

f- Teardrop Attack

Elle consiste à envoyer des paquets IP invalides à la cible, ces paquets peuvent être fragmentés, ou contenir des données corrompues ou qui dépassent la taille réglementaire .Sur certains systèmes comme les Windows avant 98 ou les Linux avant 2.0.32, ces paquets ne peuvent être interprétés et rendrons la machine inopérante.

g- Ping of Death

Elle reprend le principe de l'attaque Teardrop mais avec des paquets ICMP. Les paquets ICMP possèdent généralement un champ data de 56 octets. Certains systèmes deviennent vulnérables en envoyant des PING avec un champ de données plus important. Les systèmes en général ne sont pas prévus pour recevoir des paquets ICMP plus gros que les paquets IP traditionnels (64K), mais les PING peuvent être fragmentés. Cependant, une fois rassemblée, ces paquets causeront une saturation de la mémoire tampon. Cette attaque est de nos jours obsolète car la majorité des systèmes ont été corrigé. Elle touchait tous les systèmes d'exploitation et même les équipements réseaux tels que les routeurs et les imprimantes.

h- Permanent DoS

Ces attaques peuvent s'avérer très dangereuses car elles permettent à l'attaquant d'altérer le matériel de la victime de façon irréversible nécessitant le remplacement des équipements. Elles utilisent des failles dans les pilotes matérielles pour changer les firmwares matériels (un BIOS corrompu) et ainsi mettre hors service les machines.

i-Failles applicative

De nos jours, les systèmes d'exploitations sont de plus en plus sécurisés et les failles sont de moins en moins courantes. Par contre, de nombreuses applications installées par l'utilisateur, peuvent se révéler faible et peuvent fournir des moyens d'exploiter une machine.
L'attaquant utilise ces failles pour ensuite perpétrer des actions visant à rendre la machine indisponible. Il peut par exemple saturer le disque dur jusqu' à ce que la cible ne puisse plus fonctionner. Il peut exécuter un code créant des processus se dédoublant à l'infini aussi appelé des forkbomb. On peut encore rediriger toutes les connexions sortantes d'une machine sur elle-même pour qu'elle se sature toute seule.

1.3.5 Les attaques distribuées

La plupart des attaques, citées plus haut, peuvent être exécutées de manière distribuée .On parle alors de DDoS pour Distributed Denial of Service. parce que attaquer une cible par une seule machine se traduit souvent par un échec, alors que si un grand nombre de

machines s'attaquent à la même cible alors l'attaque a plus de chances de réussir.

Il y a deux grandes façons d'exécuter un DDoS. On peut tout d'abord utiliser un groupe de personnes en connivence et convenir d'un moment et d'une façon bien précise de mener l'attaque. Ce n'est pas la méthode la plus simple et elle nécessite beaucoup d'organisation et de logistique. L'autre façon est de disposer d'un nombre important de machines corrompues à travers le monde et de les utiliser pour perpétrer l'attaque.
Ceci nécessite au préalable une grande préparation pour corrompre les machines et les maintenir sous contrôle, mais présente aussi un avantage certain de pouvoir accomplir l'attaque seul.

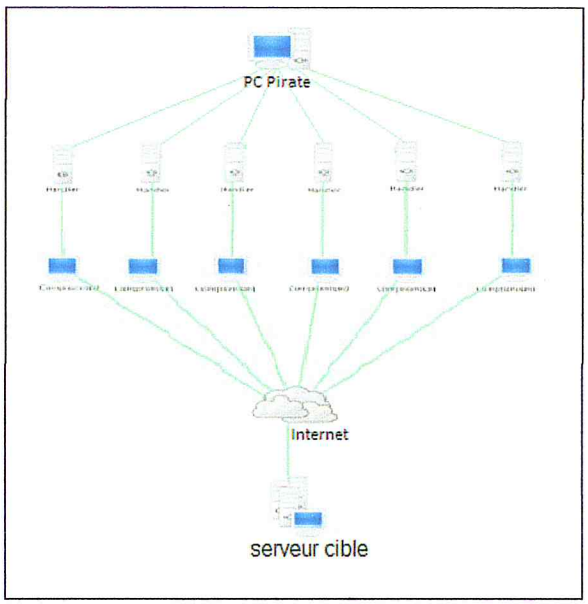


FIGURE 1.4 – DDOS attaque [23]

a- ARP spoof

est une attaque très puissante qui permet, en général, de sniffer le trafic sur le réseau en s'interposant entre une ou des victimes et la passerelle. Elle permet même de sniffer et récupérer des mots de passes sur des connexions sécurisés SSL. L'attaque inonde le réseau avec des trames ARP liant l'adresse physique de l'attaquant avec la passerelle. De cette manière, le cache ARP des victimes est corrompu et tout le trafic est redirigé vers le poste de l'attaquant.

Dans le cadre d'un déni de service, on peut utiliser l'ARP spoof en détournant un peu son utilisation classique. Au lieu de se faire passer pour la passerelle, l'attaquant corrompt la table ARP de la victime avec une route invalide, ce qui va avoir pour effet de couper la victime du réseau. Elle sera complètement isolée et restera inutilisable tant que le cache ARP sera corrompu.

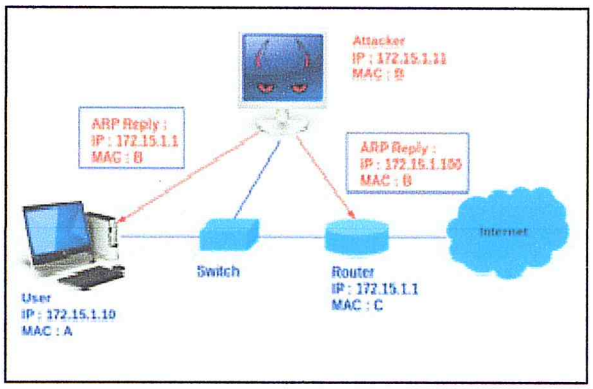


FIGURE 1.5 – spoof [24]

b- DNS spoof

De la même manière, on peut corrompre le DNS d'une victime. Normalement, ceci permet de rediriger la victime vers des sites pirates que l'on contrôle mais dans le cadre d'un déni service, on corrompt le cache DNS de fausses informations qui rendront impossible l'accès aux sites web.

1.4 les outils de protection

1.4.1 Formation des utilisations

On considère généralement que la majorité des problèmes de sécurité sont situés entre la chaise et le clavier ... Discrétion : la sensibilisation des utilisateurs à la faible sécurité des outils de communication et à l'importance de la non divulgation d'informations par ces moyens est indispensable. En effet il est souvent trop facile d'obtenir des mots de passe par téléphone ou par e-mail en se faisant passer pour un membre important de la société. Pour cela l'entreprise doit obliger les employés à lire et signer un document précisant leurs

droits et devoirs et par la même de leur faire prendre conscience de leur responsabilité individuelle.

1.4.2 Authentification et cryptage

1.4.2.1 Une authentification

est dite forte lorsqu'elle utilise deux mécanismes différents (carte à puce avec mot de passe par exemple). "Nom + mot de passe + date" sont cryptés avec des clés publiques et privées (RFC 1510). L'authentification est basée sur les 3 principes :

- Savoir : login, mot de passe...
- Être : biométrie (empreintes...)
- Avoir : clés USB, carte à puce, « token ».

1.4.2.2 Le cryptage

le cryptage identifier de manière sûre l'utilisateur connecté. Pour éviter l'espionnage, la modification du contenu, l'ajout de message... on pourra utiliser la signature électronique (CRC crypté en fin de message) ou crypter toute l'information. Les infrastructures PKI (Public Key Infrastructure) devraient se développer. Pour l'instant, le protocole SSL (Secure Socket Layer) domine toujours largement le marché de l'authentification sur les sites marchands. Radius, Tacacs ou IPSec (qui comporte un processus d'authentification dans son en-tête) constituent encore la solution retenue par la majorité des entreprises.[4]

1.4.3 Anti-virus

Un Anti-virus Est un programme capable de détecter la présence de virus sur un ordinateur et, dans la mesure du possible, de désinfecter ce dernier. On parle ainsi d'éradication de virus pour désigner la procédure de nettoyage des ordinateurs. Il existe plusieurs méthodes d'éradications :

- La suppression du code correspondant au virus dans le fichier infecté ;
- La suppression du fichier infecté ;
- La mise en quarantaine du fichier infecté, consiste à le déplacer dans un emplacement où il ne pourra pas être exécuté.

Quelques exemples des antivirus : Symantec, Avast, Kaspersky, AVG, NOD 32, Panda, Norton, etc...

1.4.4 Firewall (pare-feu)

Un firewall est un système physique (matériel) ou logique (logiciel) servant d'interface entre un ou plusieurs réseaux afin de contrôler et éventuellement bloquer la circulation des paquets de données, en analysant les informations contenues dans les couches 3, 4 et 7 du modèle OSI.
Il s'agit donc d'une machine (machine spécifique dans le cas d'un firewall matériel ou d'un ordinateur sécurisé hébergeant une application particulière de firewall) comportant au minimum deux interfaces réseau :

— une interface pour le réseau à protéger (réseau interne)
— une interface pour le réseau externe

1.4.5 Les systèmes de détection d'intrusion IDS

Un système de détection d'intrusion (ou IDS : Intrusion Détection System) est un mécanisme destiné à repérer des activités anormales ou suspectes (des Attaques) sur la cible analysée (un réseau ou un hôte). Il permet ainsi d'avoir une action de prévention sur les risques d'intrusion.

Intrusion : C'est une violation d'une politique de sécurité d'un système donné, c'est-à-dire une violation d'une des propriétés de confidentialité, d'intégrité ou de disponibilité du système en question.

Attaque : A ne pas confondre avec une intrusion. Une attaque est une tentative de violer la politique de sécurité alors qu'une intrusion est une violation effective de cette politique. En d'autres termes, une intrusion est une attaque réussie.

Même si l'intrus parvient à franchir les barrières de protection (pare-feu, système d'authentification, etc.), il est encore possible de l'arrêter avant qu'il n'attaque. Placés sur le réseau de l'entreprise, les outils de détection d'intrusion décèlent tout comportement anormal ou trafic suspect. Malgré la mise en place de solutions d'authentification, chargées

de filtrer et de contrôler les accès au réseau, il arrive que des intrus y pénètrent. C'est même le but des pirates de contourner les serveurs d'authentification, pare-feu et autres barrières de protection des systèmes. Une fois entrés, plus rien ne les empêche de saboter, de voler et d'endommager les applications.ci ici qu'interviennent alors les systèmes de détection d'intrusion. En auscultant en permanence le trafic, ils repèrent le hacker et alertent aussitôt l'administrateur. Protégeant l'entreprise des attaques externes, ces systèmes sont également capables de détecter le pirate interne qui représente encore entre 70% à 80% des actes de malveillance auxquels sont confrontées les sociétés. Il existe deux catégories d'outils sur le marché : la première analyse le trafic réseau, la seconde étudie le comportement des utilisateurs au niveau d'un système ou d'une application. Dans tous les cas, des ressources humaines devront être affectées à la supervision des systèmes de détection d'intrusion pour gérer les alertes, mais aussi pour détecter ce que les outils n'auront peut-être pas vu. Coûteuses, ces ressources freineraient aujourd'hui les entreprises dans l'adoption de ces solutions.

Il existe trois grands types d'IDS bien distincts :

— Les NIDS (Network Based Intrusion Detection System).

— Les HIDS (HostBased Intrusion Detection System).

— Les IDS hybrides à la fois NIDS et HIDS.

1.4.6 Système de prévention d'intrusion IPS

Les IPS, contrairement aux IDS (sécurité passive) classiques, constituent une sécurité active pour filtrer et bloquer les flux, ajoutant à cela la défense proactive et la prévention des intrusions sur le réseau/hôte. Avant toute action, une décision en temps réel est exécutée (l'activité est comparée à un ensemble de règles). Si l'action est conforme à l'ensemble de règles, la permission de l'exécuter sera accordée et l'action sera exécutée. Si l'action est illégale (c'est-à-dire si le programme demande des données ou veut les changer alors que cette action ne lui est pas permise), une alarme est donnée. Dans la plupart des cas, les autres détecteurs du réseau (ou une console centrale) en seront aussi informés dans le but d'empêcher les autres ordinateurs d'ouvrir ou d'exécuter des fichiers spécifiques. Le diagramme ci-dessous illustre le fonctionnement d'un IPS.

1.5 Conclusion

Dans ce chapitre, nous avons parlé de la sécurité informatique et nous avons essayé de la comprendre en étudiant les types d'attaque existants ainsi que les outils de protection utilisables pour la protection qui nous permettra de construire notre politique de sécurité.

Dans le chapitre suivant on va parler de la deuxième ligne de défense les IDS/IPS parce qu'elle représente une nouveauté pour les administrateurs réseau et implique une intervention directe de celui ci pour la configuration et l'écriture des règles de sécurité .

Chapitre 2

Les Systèmes de détection d'intrusion IDS

2.1 Introduction

Une propriété de valeur doit être protégée contre le vol et la destruction. Certaines maisons sont équipées de systèmes d'alarme qui peuvent décourager des voleurs, prévenir les autorités dans le cas d'une effraction et même avertir les propriétaires que leurs maisons est en feu. De telles mesures sont nécessaires pour assurer l'intégrité des maisons et la sécurité de leurs propriétaires. La même assurance d'intégrité et de sécurité devrait également être appliquée aux systèmes et données informatiques. L'internet a facilité le flux d'informations, personnelles, financières et autres. En même temps, il a également promu autant de dangers. Les utilisateurs malveillants recherchent des proies vulnérables comme les systèmes sans correctifs, les systèmes affectés par des chevaux de Troie et les réseaux exécutant des services peu sûrs. Des alarmes sont nécessaires pour prévenir les administrateurs et les membres de l'équipe de sécurité qu'une effraction s'est produite afin qu'ils puissent répondre en temps réel au danger.

Les systèmes de détection d'intrusions ont été conçus pour jouer le rôle d'un tel système d'alarme. Dans ce chapitre nous présentons tout d'abord la notion de système de détection d'intrusions ainsi que son architecture. Nous présentons également la classification des IDS. Dans ce cadre plusieurs critères sont pris en compte où nous commençons par la classification selon la méthode d'analyse qui découpe les IDS en deux approches (comportementale et par signatures), enfin nous allons mettre le point sur la détection

d'intrusions réseau.

2.2 Définition

La détection des intrusions est le processus de surveillance des événements qui se trouvent dans un système des ordinateurs ou du réseau et les analysant pour détecter les signes des intrusions, défini comme des tentatives pour compromettre la confidentialité, l'intégrité, la disponibilité ou éviter des mécanismes de sécurité de l'ordinateur ou du réseau. L'intrusion est causée par les attaques accédant au système via Internet, autorisée l'utilisateur du système qui essaye de gagner les privilèges supplémentaires pour lesquels ils n'ont pas été autorisés, et autoriser les utilisateurs qui abusent les privilèges donnés. Le système de détection des intrusions est un logiciel ou un matériel qui automatise des surveillances et les processus analysés.

Les IDS protègent un système contre les attaques, les mauvaises utilisations et les compromis. Ils peuvent également surveiller l'activité du réseau, analyser les configurations du système et du réseau contre toute vulnérabilité, analyser l'intégrité de données et bien plus.

Selon les méthodes de détection qu'on peut choisir de déployer, il existe plusieurs avantages directs et secondaires au fait d'utiliser un IDS.

Un IDS a quatre fonctions principales : l'analyse, la journalisation, la gestion et l'action.

- Analyse : Analyse des journaux du système pour identifier des intentions dans la masse de données recueillie par l'IDS. Il y a deux méthodes d'analyses :L'une basée sur les signatures d'attaques, et l'autre sur la détection d'anomalies.
- Journalisation : Enregistrement des évènements dans un fichier de log. Exemples d'évènements : arrivée d'un paquet, tentative de connexion.
- Gestion : Les IDS doivent être administrés de manière permanente. On peut assimiler un IDS à une caméra de sécurité.
- Action : Alerter l'administrateur quand une attaque dangereuse est détectée.

2.3 Présentation d'un système de détection d'intrusions

Dans cette section, nous allons décrire les systèmes de détection d'intrusions en générale.

2.3.1 Architecture d'un IDS

Plusieurs schémas ont été proposés pour décrire les composants d'un système de détection d'intrusions. Parmi eux, nous avons retenu celui issu des travaux d'Intrusions Detection exchange format Working Group (IDWG) de l'Internet Engineering Task Force (IETF) comme base de départ, car il résulte d'un large consensus parmi les intervenants du domaine. [10] L'objectif des travaux du groupe IDWG est la définition d'un standard de communication entre certains composants d'un système de détection d'intrusions. La figure 2.1 illustre ce modèle et permet d'introduire un certain nombre de concepts :

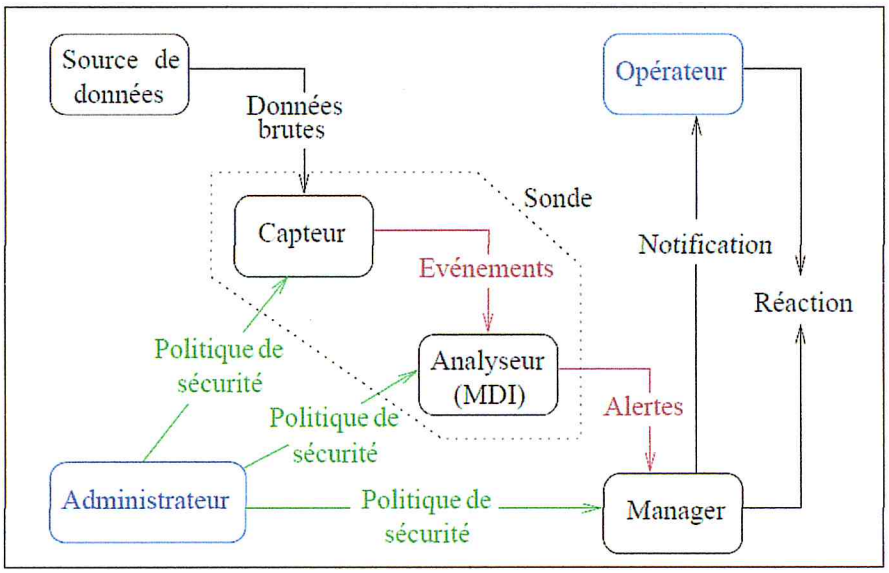


FIGURE 2.1 – Modèle générique de la détection d'intrusions proposé par l'IDWG [11]

L'architecture IDWG d'un système de détection d'intrusions contient des capteurs qui envoient des événements à un analyseur. Les capteurs couplés avec un analyseur forment une sonde, cette dernière envoie des alertes vers un manager qui la notifie à un opérateur humain.

2.3.2 Les différents éléments de l'architecture IDWG

- Administrateur : Personne chargée de mettre en place la politique de sécurité, et par conséquent, de déployer et configurer les IDSs.
- Alerte : message formaté émis par un analyseur s'il trouve des activités intrusives dans une source de données.
- Analyseur : c'est un outil logiciel qui met en œuvre l'approche choisie pour la détection (comportementale ou par scénarios), il génère des alertes lorsqu'il détecte une intrusion.
- Capteur : logiciel générant des événements en filtrant et formatant les données brutes provenant d'une source de données.
- Evénement : message formaté et renvoyé par un capteur. C'est l'unité élémentaire utilisée pour représenter une étape d'un scénario d'attaques connu.
- Manager : composant d'un IDS permettant à l'opérateur de configurer les différents éléments d'une sonde et de gérer les alertes reçues et éventuellement la réaction.
- Notification : la méthode par laquelle le manager d'IDS met au courant l'opérateur de l'occurrence d'alerte.
- Opérateur : personne chargée de l'utilisation du manager associé à l'IDS. Elle propose ou décide de la réaction à apporter en cas d'alerte. C'est, parfois, la même personne que l'administrateur.
- Réaction : mesures passive ou actives prises en réponse à la détection d'une attaque, pour la stopper ou pour corriger ses effets.
- Sonde : un ou des capteurs couplés avec un analyseur.
- Source de données : dispositif générant de l'information sur les activités des entités du système d'information.

Dans ce modèle qui représente le processus complet de la détection ainsi que l'acheminement des données au sein d'un IDS, l'administrateur configure les différents composants (capteur(s), analyseurs(s), manager(s)) selon une politique de sécurité bien définie. Les capteurs accèdent aux données brutes, les filtrent et les formatent pour ne renvoyer que les événements intéressants à un analyseur. Les analyseurs utilisent ces événements pour décider de la présence ou non d'une intrusion et envoient dans le cas échéant une alerte au manager, qui notifie l'opérateur humain, une réaction éventuelle peut être menée automatiquement par le manager ou manuellement par l'opérateur. [11]

2.4 Vocabulaire de la détection d'intrusions

La détection d'intrusions utilise un vocabulaire bien définis qui ne se trouve pas dans le modèle précédent et qui est comme suit :

- Attaque ou intrusion : c'est une action qui permet de violer la politique de sécurité.
- Audit de sécurité : c'est l'ensemble des mécanismes permettant la collecte d'informations sur les actions faites sur un système d'information.
- Détection d'intrusions : c'est la recherche des traces laissées par une intrusion dans les données produites par une source.
- Faux positif : c'est une alerte en absence d'attaque.
- Faux négatif : c'est l'absence d'alerte en présence d'attaque.
- Vulnérabilité : c'est une faille de conception, d'implémentation ou de configuration d'un système logiciel ou matériel.
- Log (trace d'audit) : c'est un fichier système à analyser.
- Exploit :c'est un terme utilisé pour désigner un programme d'attaque.
- Scénario : c'est une suite constituée des étapes élémentaires d'une attaque.
- Signature : suites des étapes observables d'une attaque, utilisée par certains analyseurs pour rechercher dans les activités des entités, des traces de scénarios d'attaques connus.
- Système de détection d'intrusions : ensemble constitué d'un ou plusieurs capteurs, un ou plusieurs analyseurs et un ou plusieurs managers.
- Corrélation : c'est l'interprétation conceptuelle de plusieurs événements (alertes)visant à leur assigner une meilleure sémantique et à réduire la quantité globale d'événements (d'alertes).

2.5 Emplacement d'un système de détection d'intrusions

Il existe plusieurs endroits stratégiques où il convient de placer un IDS. Le schéma suivant illustre un réseau local ainsi que les positions que peut y prendre un IDS :

- Elément 1
- Position (1) : Sur cette position, l'IDS va pouvoir détecter l'ensemble des attaques frontales, provenant de l'extérieur, en amont du firewall. Ainsi, beaucoup d'alertes

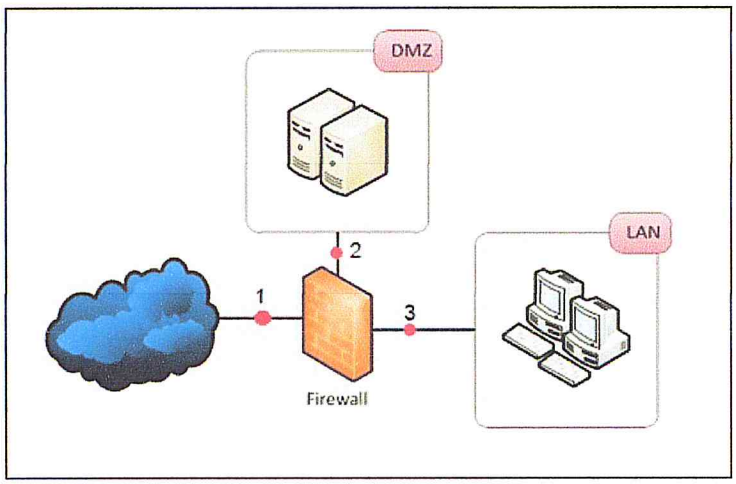


FIGURE 2.2 – Endroits typiques pour un système de détection d'intrusions [25]

seront remontées ce qui rendra les logs difficilement consultables.

— Position (2) : Si l'IDS est placé sur la DMZ, il détectera les attaques qui n'ont pas été filtrées par le firewall et qui relèvent d'un certain niveau de compétence. Les logs seront ici plus clairs à consulter puisque les attaques bénins ne seront pas recensées.

— Position (3) : L'IDS peut ici rendre compte des attaques internes, provenant du réseau local de l'entreprise. Il peut être judicieux d'en placer un à cet endroit étant donné le fait que 80 des attaques proviennent de l'intérieur. De plus, si des trojans ont contaminé le parc informatique (navigation peu méfiante sur internet)ils pourront être ici facilement identifiés pour être ensuite éradiqués.

2.6 Classification des systèmes de détection d'intrusions

Nous pouvons classifier les systèmes de détection d'intrusions selon cinq critères comme il est illustré dans la figure 2.3 :

1. La source des données à analyser.
2. Le lieu de l'analyse des données.
3. La fréquence de l'analyse.
4. Le comportement en cas d'attaque détectée.
5. La méthode de détection utilisée.

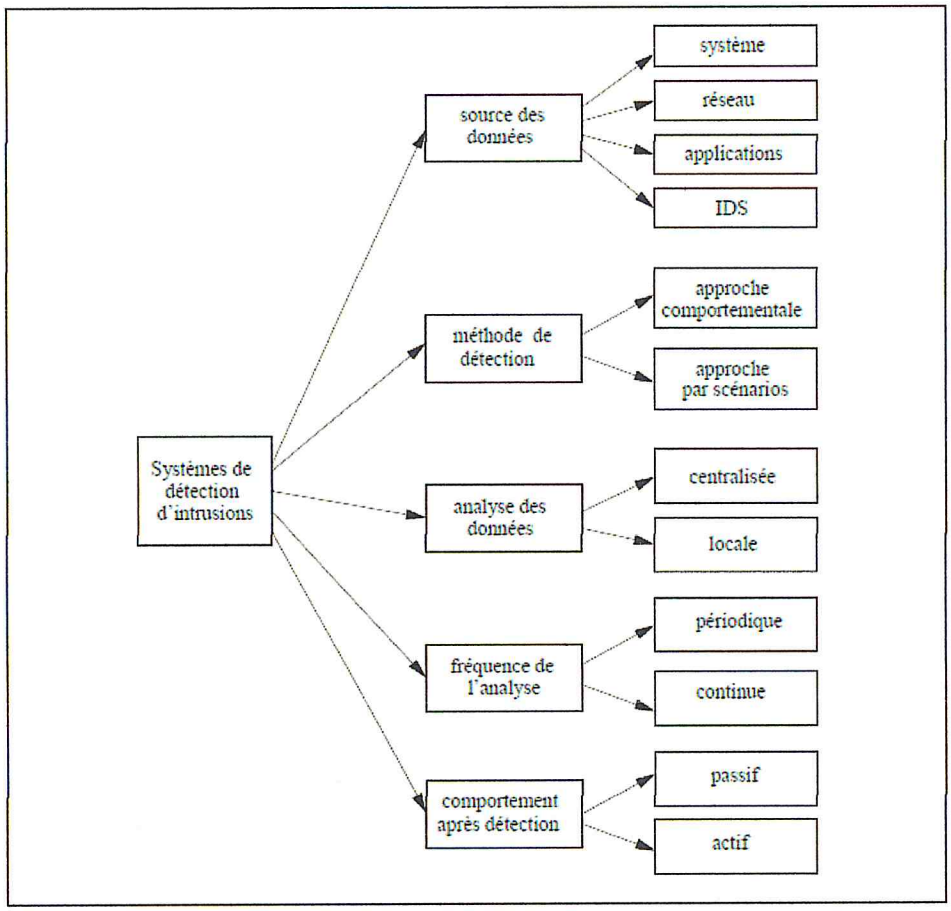


FIGURE 2.3 – Classification des systèmes de détection d'intrusions [11].

2.6.1 Source des données à analyser

Les sources possibles de données à analyser sont une caractéristique essentielle des systèmes de détection d'intrusions puisque ces données constituent la matière première du processus de détection. Les données proviennent soit de logs générés par le système d'exploitation, soit de logs applicatifs, soit d'informations provenant du réseau, soit encore d'alertes générées par d'autres IDS. [11]

2.6.1.1 Source d'information système

Un système d'exploitation fournit généralement plusieurs sources d'information :

— Commandes systèmes : presque tous les systèmes d'exploitation fournissent des commandes pour avoir un « instantané » de ce qui se passe.

— Accounting : l'accounting fournit de l'information sur l'usage des ressources partagées par les utilisateurs (temps processeur, mémoire, espace disque, débit réseau, applications lancées, ...).

— Audit de sécurité : tous les systèmes d'exploitation modernes proposent ce service pour fournir des événements système, les associer à des utilisateurs et assurer leur collecte dans un fichier d'audit. On peut donc potentiellement disposer d'informations sur tout ce que font (ou ont fait) les utilisateurs : accès en lecture à un fichier, exécution d'une application, etc.

2.6.1.2 Source d'information réseau

Des dispositifs matériels ou logiciels (snifer) permettent de capturer le trafic réseau. Cette source d'information est particulièrement adaptée lorsqu'il s'agit de rechercher les attaques en déni de service qui se passent au niveau réseau ou les tentatives de pénétration à distance. Le processus d'interception des paquets peut être rendu quasiment invisible pour l'attaquant car on peut utiliser une machine dédiée juste reliée à un brin du réseau, configurée pour ne répondre à aucune sollicitation extérieure et dont personne ne soupçonnera l'existence. Néanmoins, il est difficile de garantir l'origine réelle de l'attaque que l'on a détectée car il est facile de masquer son identité en modifiant les paquets réseau .

2.6.1.3 Source d'information applicative

Les applications peuvent également constituer une source d'information pour les IDS. Les capteurs applicatifs sont de deux natures :

— Capteur interne : le filtrage sur les activités de l'application est alors exécuté par le code de l'application.

— Capteur externe : le filtrage se fait à l'extérieur de l'application. Plusieurs méthodes sont utilisées : un processus externe peut filtrer les logs produits par l'application ou bien l'exécution de l'application peut être interceptée (au niveau de ses appels de librairies ou d'un proxy applicatif).

Prendre ses informations directement au niveau de l'application présente plusieurs avantages. Premièrement, les données interceptées ont réellement été reçues par l'application.

Il est donc difficile d'introduire une désynchronisation entre ce que voit passer le capteur applicatif et ce que reçoit l'application contrairement à ce qu'il peut se passer avec les capteurs réseau. Ensuite, cette source d'information est généralement de plus haut niveau que les sources système et réseau. Cela permet donc de filtrer des événements qui ont une sémantique plus riche. Finalement, si l'on prend l'exemple d'une connexion web chiffrée par SSL, un capteur réseau ne verra passer que des données pseudo-aléatoires tandis qu'un capteur associé au serveur web pourra analyser le texte en clair de la requête.

2.6.1.4 Source d'information basée IDS

Une autre source d'information, souvent de plus haut niveau que les précédentes, peut être exploitée. Il s'agit des alertes remontées par des analyseurs provenant d'un IDS. Chaque alerte synthétise déjà un ou plusieurs événements intéressants du point de vue de la sécurité. Elles peuvent être utilisées par un IDS pour déclencher une analyse plus fine à la suite d'une indication d'attaque potentielle. De surcroît, en corrélant plusieurs alertes, on peut parfois détecter une intrusion complexe de plus haut niveau. Il y aura alors génération d'une nouvelle alerte plus synthétique que l'on qualifie de méta-alerte.

2.6.2 Localisation de l'analyse des données

On peut également faire une distinction entre les IDS en se basant sur la localisation réelle de l'analyse des données :

- Analyse centralisée : certains IDS ont une architecture multi-capteurs (ou multi-sondes). Ils centralisent les événements (ou alertes) pour une analyse au sein d'une seule machine. L'intérêt principal de cette architecture est de faciliter la corrélation entre les événements puisqu'on dispose alors d'une vision globale. Par contre, la charge des calculs (effectués sur le système central) ainsi que la charge réseau (due à la collecte des événements ou des alertes) peuvent être lourdes et risquent de constituer un goulet d'étranglement.
- Analyse locale : si l'analyse du flot d'événements est effectuée au plus près de la source de données (généralement en local sur chaque machine disposant d'un capteur), on minimise le trafic réseau et chaque analyseur séparé dispose de la même puissance de calcul. En contrepartie, il est impossible de croiser des événements

qui sont traités séparément et l'on risque de passer à côté de certaines attaques distribuées.

2.6.3 Fréquence de l'analyse

Une autre caractéristique des systèmes de détection d'intrusions et leur fréquence d'utilisation :

— Périodique : certains systèmes de détection d'intrusions analysent périodiquement les fichiers d'audit à la recherche d'une éventuelle intrusion ou anomalie passée. Cela peut être suffisant dans des contextes peu sensibles (on fera alors une analyse journalière, par exemple).

— Continue : la plupart des systèmes de détection d'intrusions récents effectuent leur analyse des fichiers d'audit ou des paquets réseau de manière continue afin de proposer une détection en quasi temps-réel. Cela est nécessaire dans des contextes sensibles (confidentialité) et/ou commerciaux (confidentialité, disponibilité). C'est toute fois un processus coûteux en temps de calcul car il faut analyser à la volée tout ce qui se passe sur le système.

2.6.4 Comportement après détection

Une autre façon de classer les systèmes de détection d'intrusions consiste à les classer par type de réaction lorsqu'une attaque est détectée :

— passive :la plupart des systèmes de détection d'intrusions n'apportent qu'une réponse passive à l'intrusion. Lorsqu'une attaque est détectés, ils génèrent une alarme et notifient l'administrateur système par e-mail, message dans une console, voir même par beeper. C'est alors lui qui devra prendre les mesures qui s'imposent .

— active : d'autres systèmes de détection d'intrusions peuvent, en plus de la notification à l'opérateur, prendre automatiquement des mesures pour stopper l'attaque en cours. Par exemple, ils peuvent couper les connexions suspectes ou même, pour une attaque externe, reconfigurer le pare-feu pour qu'il refuse tout ce qui vient du site incriminé. Des outils tels que RealSecure ou NetProwler proposent ce type de réaction. Toutefois, il apparait que ce type de fonctionnalité automatique est potentiellement dangereux car il peut mener à des dénis de service provoqués par l'IDS. Un attaquant déterminé peut, par exemple, tromper l'IDS en usurpant des

adresses du réseau local qui seront alors considérées comme la source de l'attaque par l'IDS. Il est préférable de proposer une réaction facultative à un opérateur humain (qui prend la décision finale).

2.6.5 Méthode de détection

Dans la littérature, on distingue deux approches principales de détection. Une approche comportementale qui se base sur l'hypothèse selon laquelle nous pouvons définir un comportement normal de l'utilisateur et que toute déviation par rapport à celui-ci est potentiellement suspecte. La deuxième approche est l'approche par signature qui s'appuie sur un modèle constitué des sections interdites dans le système informatique. Ce modèle s'appuie sur la connaissance des techniques employées par les attaquants, on tire des scénarios d'attaque et on recherche dans les traces d'audit leur éventuelle survenue.

2.6.5.1 L'approche comportementale

Les détecteurs d'intrusions comportementaux reposent sur la création d'un modèle de référence représentant le comportement de l'entité surveillée en situation de fonctionnement normal. Ce modèle est ensuite utilisé durant la phase de détection afin de pouvoir mettre en évidence d'éventuelles déviations comportementales. Pour cela, le comportement de l'entité surveillée est comparé à son modèle de référence. Une alerte est levée lorsqu'une déviation trop importante (notion de seuil) vis-à-vis de ce modèle de comportement normal est détectée. Le principe de cette approche est de considérer tout comportement n'appartenant pas au modèle de comportement normal comme une anomalie symptomatique d'une intrusion ou d'une tentative d'intrusion. [13] On peut distinguer deux catégories de profils :

a- Profils construits par apprentissage

Parmi les méthodes proposées pour construire les profils par apprentissage, les plus marquantes sont les suivantes : [11]

— méthode statistique : le profil est calculé à partir de variables considérées comme aléatoires et échantillonnées à intervalles réguliers. Ces variables peuvent être le temps processeur utilisé, la durée et l'heure des connexions, etc. Un modèle statistique est alors utilisé pour construire la distribution de chaque variable et pour

mesurer, au travers d'une grandeur synthétique, le taux de déviation entre un comportement courant et le comportement passé.

— système expert : ici, c'est une base de règles qui décrit statistiquement le profil de l'utilisateur au vu de ses précédentes activités. Son comportement courant est comparé aux règles, à la recherche d'une anomalie. La base de règles est rafraîchie régulièrement.

— réseaux de neurones : la technique consiste à apprendre à un réseau de neurones le comportement de l'entité à surveiller. Par la suite, lorsqu'on lui fournira en entrée les actions courantes effectuées par l'entité, il devra décider de leur normalité.

— analyse de signatures : Il s'agit de construire un modèle de comportement normal des services réseaux. Le modèle consiste en un ensemble de courtes séquences d'appels système représentatifs de l'exécution normale du service considéré. Des séquences d'appels étrangères à cet ensemble sont alors considérées comme l'exploitation potentielle d'une faille du service.

Pour toutes ces méthodes, le comportement de référence utilisé pour l'apprentissage étant rarement exhaustif, on s'expose à des risques de fausses alarmes (faux positifs). De plus, si des attaques ont été commises durant cette phase, elles seront considérées comme normales(risque de faux négatifs).

b- Profils spécifiant une politique de sécurité (policy-based)

Pour les IDS dits policy-based, il n'y a pas de phase d'apprentissage. Leur comportement de référence est spécifié par une politique de sécurité : la détection d'une intrusion intervient chaque fois que la politique est violée. Le profil est ici une politique de sécurité qui décrit la suite des appels systèmes licites d'une application. L'approche comportementale possède un certain nombre d'avantages et d'inconvénients :[11]

Les avantages :

— l'analyse comportementale n'exige pas des connaissances préalables sur les attaques.

— Elle permet la détection de la mauvaise utilisation des privilèges.

— Elle permet de produire des informations qui peuvent être employées pour définir des signatures pour l'analyse basée connaissance.

Les inconvénients

— Les approches comportementales produisent un taux élevé des alarmes type faux positif en raison des comportements imprévisibles des utilisateurs et des réseaux.
— Ces approches nécessitent des phases d'apprentissage pour caractériser les profils de comportement normaux.
— Les alarmes générées par cette approche ne sont pas significatives.

c- L'approche par scénarios (signature)

On construit des scénarios d'attaque en spécifiant ce qui est caractéristique de l'attaque et qui doit être observé dans les traces d'audit. L'analyse des traces d'audit se fait à la recherche de ces scénarios. Les méthodes proposées à ce jour sont les suivantes : [11]

— système expert : le système expert comporte une base de règles qui décrit les attaques. Les événements d'audit sont traduits en des faits qui sont interprétables par le système expert. Son moteur d'inférence décide alors si une attaque répertoriée s'est produite ou non.
— analyse de signatures : il s'agit là de la méthode la plus en vue actuellement. Des signatures d'attaques sont fournies à des niveaux sémantiques divers selon les outils (de la suite d'appels système aux commandes passées par l'utilisateur en passant par les paquets réseau). Divers algorithmes sont utilisés pour localiser ces signatures connues dans les traces d'audit. Ces signatures sont toujours exprimées sous une forme proche des traces d'audit. Si l'on prend l'exemple des NIDS, les algorithmes de recherche de motifs utilisés permettent d'obtenir de bonnes performances en vitesse de traitement mais génèrent de nombreuses fausses alertes.
— automates à états finis : plusieurs IDS utilisent des automates à états finis pour coder le scénario de reconnaissance de l'attaque. Cela permet d'exprimer des signatures complexes et comportant plusieurs étapes. On passe d'un état initial sûr à un état final attaqué via des états intermédiaires. Chaque transition entre états est déclenchée par des conditions sur les événements remontés par les capteurs.

L'approche par scénario possède un certain nombre d'avantages et d'inconvénients :

Les avantages
— l'analyse basée connaissance est très efficace pour la détection d'attaque avec un

taux très bas des alarmes de type faux positif.

— Les alarmes générées sont significatives.

Les inconvénients

— Cette analyse basée connaissance permet seulement la détection des attaques qui sont connues au préalable. Donc, la base de connaissances doit être constamment mise à jour avec les signatures des nouvelles attaques.

— Le risque que l'attaquant peut influencer sur la détection après la reconnaissance des signatures.

2.7 Les différentes sortes d'IDS

Les différents IDS se caractérisent par leur domaine de surveillance. Celui-ci peut se situer au niveau d'un réseau d'entreprise, d'une machine hôte, d'une application. Nous allons tout d'abord étudier la détection d'intrusion basée sur l'hôte, puis basée sur une application, avant de nous intéresser aux IDS réseaux, NIDS et NNIDS (Network IDS et Node Network IDS). [14]

2.7.1 La détection d'intrusions basée sur l'hôte

Les systèmes de détection d'intrusions basés sur l'hôte ou HIDS (Host IDS) analysent exclusivement l'information concernant cet hôte. Comme ils n'ont pas à contrôler le trafic du réseau mais "seulement" les activités d'un hôte. Ils se montrent habituellement plus précis sur les types d'attaques subies[14].

De plus, l'impact sur la machine concernée est sensible immédiatement, par exemple dans le cas d'une attaque réussie par un utilisateur. Ces IDS utilisent deux types de sources pour fournir une information sur l'activité de la machine, les logs et les traces d'audit du système d'exploitation. Chacun a ses avantages, les traces d'audit sont plus précises et détaillées et fournissent une meilleure information alors que les logs qui ne fournissent que l'information essentielle sont plus petits. Ces derniers peuvent être mieux contrôlés et analysés en raison de leur taille, mais certaines attaques peuvent passer inaperçues, alors qu'elles sont détectables les par une analyse des traces d'audit.

Ce type d'IDS possède un certain nombre d'avantages .Il est possible de constater immédiatement l'impact d'une attaque et donc de mieux réagir. Grâce à la quantité des informations étudiées, il est possible d'observer les activités se déroulant sur l'hôte avec précision et d'optimiser le système en fonction des activités observées. De plus, les HIDS sont extrêmement complémentaires des NIDS. En effet, ils permettent de détecter plus facilement les attaques de type "Cheval de Troie", alors que ce type d'attaque est difficilement détectable par un NIDS. Les HIDS permettent également de détecter des attaques impossibles à détecter avec un NIDS, car elles font partie de trafic crypté.

Néanmoins, ce type d'IDS possède également ses faiblesses qui proviennent de ses qualités, du fait de la grande quantité de données générées. Ce type d'IDS est très sensible aux attaques de type DoS, qui peuvent faire exploser la taille des fichiers de logs.

Un autre inconvénient tient justement à la taille des fichiers de rapport d'alertes à examiner, qui est très contraignante pour le responsable sécurité. La taille des fichiers peut en effet atteindre plusieurs Mégaoctets. Du fait de cette quantité de données à traiter, ils sont assez gourmands en CPU et peuvent parfois altérer les performances de la machine hôte.

Enfin, ils ont moins de facilité à détecter les attaques de type hôte que les IDS réseaux.

Les HIDS sont en général placés sur des machines sensibles, susceptibles de subir des attaques et possédantes des données sensibles pour l'entreprise. Les serveurs web et applicatifs peuvent notamment être protégés par un HIDS. Pour finir, voici quelques HIDS connus :

- Tripwire.
- WATCH.
- DragonSquire.
- Tiger.
- Security Manager.
- Etc.

2.7.2 Détection d'Intrusions basée sur une application

Les IDS basés sur les applications sont un sous-groupe des IDS hôtes. Ils contrôlent l'interaction entre un utilisateur et un programme en ajoutant des fichiers de log afin de fournir de plus amples informations sur les activités d'une application particulière. Puisque vous opérez entre un utilisateur et un programme, il est facile de filtrer tout comportement notable. Un ABIDS se situe au niveau de la communication entre un utilisateur et l'application surveillée[14].

L'avantage de cet IDS est qu'il lui est possible de détecter et d'empêcher des commandes particulières dont l'utilisateur pourrait se servir avec le programme et de surveiller chaque transaction entre l'utilisateur et l'application. De plus, les données sont décodées dans un contexte connu, leur analyse est donc plus fine et précise.

Par contre, du fait que cet IDS n'agit pas au niveau du noyau, la sécurité assurée est plus faible, notamment en ce qui concerne les attaques de type "Cheval de Troie". De plus, les fichiers de log générés par ce type d'IDS sont des cibles faciles pour les attaquants et ne sont pas aussi sûrs, par exemple, que les traces d'audit du système.

Ce type d'IDS est utile pour surveiller l'activité d'une application très sensible, mais son utilisation s'effectue en général en association avec un HIDS. Il faudra dans ce cas contrôler le taux d'utilisation CPU des IDS afin de ne pas compromettre les performances de la machine.

2.7.3 La Détection d'Intrusions Réseau

Le rôle essentiel d'un IDS réseau est l'analyse et l'interprétation des paquets circulant sur ce réseau. L'implantation d'un NIDS sur un réseau se fait de la façon suivante : des capteurs sont placés aux endroits stratégiques du réseau et génèrent des alertes s'ils détectent une attaque. Ces alertes sont envoyées à une console sécurisée, qui les analyse et les traites éventuellement. Cette console est généralement située sur un réseau isolé, qui relie uniquement les capteurs et la console [14].

Les avantages des NIDS sont les suivants : les capteurs peuvent être bien sécurisés

puisqu'ils se contentent d'observer le trafic et permettent donc une surveillance discrète du réseau, les attaques de type scans sont facilement détectées, et il est possible de filtrer le trafic.

Les NIDS sont très utilisés et remplissent un rôle indispensable, mais ils présentent néanmoins de nombreuses faiblesses. En effet, la probabilité de faux négatifs (attaques non détectées comme telles) est élevée et il est difficile de contrôler le réseau entier. De plus, ils doivent principalement fonctionner de manière cryptée d'où une complication de l'analyse des paquets. Pour finir, à l'opposé des IDS basés sur l'hôte, ils ne voient pas les impacts d'une attaque

Voici quelques exemples de NIDS :

— NetRanger.
— Dragon.
— NFR.
— Snort.
— ISSRealSecure.

2.7.4 Les IDS hybrides

Les IDS hybrides rassemblent les caractéristiques des NIDS et HIDS. Ils permettent, en un seul outil, de surveiller le réseau et les terminaux. Les sondes sont placés en des points stratégiques, et agissent comme NIDS et/ou HIDS suivant leurs emplacements. Toutes ce son des remontent alors les alertes à une machine qui va centraliser le tout, et lier les informations d'origines multiples. Ainsi, on comprend que les IDS hybrides sont basés sur une architecture distribuée, où chaque composant unifie son format d'envoi, cela permet de communiquer et d'extraire des alertes plus pertinentes. [15]

Les avantages des IDS hybrides sont multiples :

— Moins de faux positifs
— Meilleure corrélation (la corrélation permet de générer de nouvelles alertes à partir de celles existantes).
— Possibilité de réaction sur les analyseurs.

2.8 Les principales tâches d'un IDS

Un IDS permet de repérer des anomalies dans le trafic réseau comme suit :

— Détecter les tentatives de découvertes du réseau.

— Détecter dans certains cas, si l'attaque a réussi ou non.

— Détecter le Dénis de Service.

— Détecter le niveau d'infection du système informatique et les zones réseaux touchées.

— Repérer les machines infectées.

— Alerter de façon centrale pour toutes les attaques.

— Réagir aux attaques et corriger les problèmes éventuels.

Si on compare les HIDS et NIDS, les HIDS présente un avantage considérable par rapport à un NIDS dans le cas où le trafic est crypté. En effet, un NIDS n'a pas connaissance des clés de cryptage et ne peut appliquer ses algorithmes de détection au niveau des données chiffrées.La détection est effectuée à l'extrémité de la chaine de communication, une fois le flux est décrypté. Ceci est réalisé en mettant en œuvre un agent HIDS directement sur le serveur cible. Les flux chiffrés sont ainsi décodés par la cible et transmis ensuite au monteur d'analyse de l'NIDS.[5]

2.9 les limites d'un IDS

Parmi les faiblesses des IDS on trouve :

- **Nombreux faux positifs.**

- **Configuration complexe et longue.**

— Nombreux faux positifs après configuration.

- **Pas de connaissance de la plate-forme.**

— De ses vulnérabilités .

— Du contexte métier .

- **Les attaques applicatives sont difficilement détectables.**

— Injection SQL .

— Exploitation de CGI mal conçus .

- **Des évènements difficilement détectables.**

— Scans lents / distribués .

— Canaux cachés / tunnels .

- **Pollution des IDS .**

— Consommation des ressources de l'IDS.

— Perte de paquets.

— Déni de service contre l'IDS / l'opérateur.

— Une attaque réelle peut passer inaperçue.

- **Attaque contre l'IDS lui-même.**

- **Ils ne peuvent pas compenser les trous de sécurité dans les protocoles réseaux.**

- **Ils ne peuvent pas compenser des manques significatifs dans votre stratégie de sécurité, votre politique de sécurité ou votre architecture de sécurité.** [16]

2.10 Quelques outils de détection d'intrusions

Il existe plusieurs IDS sur le marché, parmi eux :

ISS RealSecure Internet Security Systems (ISS) fournit ISS RealSecure IDS, une plate-forme de détection d'intrusions intégrée. ISS RealSecure IDS utilise une approche basée sur des normes pour comparer les entrées de trafic réseau et journaux d'accueil des méthodes connues et probables des attaquants. ISS RealSecure IDS intègre avec de nombreuses applications de réseau et de gestion des systèmes. RealSecure combine en un seul agent trois fonctionnalités essentielles :

— Un moteur de détection d'intrusions.

— Un firewall personnel.

— Un module de contrôle d'applications et de communications.

Enterasys DRAGON Edité par Enterasys Networks, c'est un système de détection des intrusions considérée comme un leader du marché du fait de ses performances, ses facultés d'adaptation à tout type d'environnement et ses capacité d'analyse. Les solutions Dragon sont constituées de sondes NIDS (Network Sensor),d'agents HIDS (Host Sensor) et d'un système de management qui assure les fonctions d'exploitation des événements de la suite Dragon. Le Network Sensor est un NIDS disponible en version logicielle ou en boitier dédié. Depuis la version 6, Enterasys décline les applications et les logiciels en trois versions selon la bande passante à analyser. Les versions matérielles sont adossées à leur équivalent logiciel. Le Host Sensor est un agent HIDS qui détecte les attaques contre le système sur le quel il est installé en contrôlant les journaux systèmes et d'audit ainsi qu'en utilisant des mécanismes d'analyse de signatures. Dragon détecte les intrusions sur l'ensemble de l'infrastructure informatique où qu'elles se produisent et permet d'avoir ainsi une visibilité globale sur le système d'information. Cela permet notamment d'optimiser les ressources humaines nécessaires à l'analyse des journaux issus des différents firewalls ou serveur Web en fédérant tous ces journaux au niveau d'une console Dragon unique qui analysera automatiquement les données afférentes. [17]

SNORT est un IDS particulièrement répondu car fourni en open source. Il est donc gratuit et facile à se procurer, outre sa gratuité, son avantage est qu'il dispose d'une très grosse base de signatures réalisée par la communauté des utilisateurs.
C'est également le gage d'obtenir rapidement des mises à jour de la base dès qu'une nouvelle menace est signalé. Il a été conçu à l'origine pour le système linux mais il a également été porté sous Windows. On trouve dans le commerce plusieurs livres dédiés à l'installation et à l'utilisation de SNORT.
SNORT est généralement utilisé en conjonction avec un autre logiciel open source nommé BASE qui est la console de gestion et d'analyse.
Pour ce qui est des points négatifs, on peut néanmoins considérer que SNORT est moins puissant en termes de moteur d'analyse que des solutions commerciales telles que celle d'ISS.
Par ailleurs, bien que la base de signatures soit étendue, elle nécessite un travail constant de la part de l'administrateur qui doit les télécharger manuellement. Il n'y a pas de procédure de mise à jour automatiques. [18]

2.11 Conclusions

Ce chapitre nous a permis de découvrir les systèmes de détection d'intrusions leurs fonctionnement et leur capacités. La plupart des IDS sont fiables, ce qui explique qu'ils sont souvent intégrés dans les solutions de sécurité. Les avantages qu'ils présentent face aux autres outils de sécurités les favorisent, mais d'un autre côté cela n'empêche pas que les meilleurs IDS présentent aussi des lacunes et quelques inconvénients. Nous comprenons donc bien qu'ils sont nécessaires mais ne peuvent pas se passer de l'utilisation d'autres outils de sécurité visant à combler leurs défauts.

Nous allons voir dans le chapitre suivant comment réussir une bonne configuration de système de détection d'intrusion SNORT.

Chapitre 3

Les Systèmes de détection d'intrusion snort

3.1 Introduction

Autre la mise en place d'un pare-feu et d'un système d'authentification, il est de nos jours nécessaire de mettre en place un système de détection d'intrusions. SNORT est un logiciel open source écrit par Martin Roesch, disponible sous licence GNU, son code source est accessible et modifiable.
Dans ce qui suit, nous allons commencer par donner une présentation générale de SNORT, en suite nous allons présenter leur manipulation : installation, configuration et fonctionnalités, et une vue générale sur wireshark qui nous permettra de capturer les paquets.

On va voir aussi une description des outils d'attaques DOS et leur taxonomie et on fera un choix sur celle qu'on va utiliser pour simuler des attaques dans le chapitre suivant.

3.2 Snort

Snort est un Système de Détection d'Intrusion de réseau Open Source. Il est capable d'effectuer une analyse du trafic réseau en temps réel. IL est doté de différentes technologies de détection d'intrusions telles que l'analyse protocolaire. Snort peut détecter de nombreux types d'attaques : comme l'attaque de scans des ports.

Snort est doté d'un langage de règles permettant de décrire le trafic. De plus, son moteur de détection utilise une architecture modulaire de plug-ins. Snort est principalement dédié aux acteurs de la sécurité réseaux. En effet, sa fonction IDS permet une surveillance des réseaux permettant de détecter et d'alerter en cas de tentative d'intrusion sur le réseau.

3.2.1 Positionnement de Snort dans le réseau

L'emplacement physique de la sonde SNORT sur le réseau a un impact considérable sur son efficacité. Dans le cas d'une architecture classique, composée d'un Firewall et d'une DMZ, trois positions sont généralement envisageables :

Avant le Firewall ou le routeur filtrant : dans cette position, la sonde occupe une place de premier choix dans la détection des attaques de sources extérieures visant l'entreprise. SNORT pourra alors analyser le trafic qui sera éventuellement bloqué par le Firewall. Les deux inconvénients de cette position du NIDS sont :

— le risque engendré par un trafic très important qui pourrait entraîner une perte de fiabilité

— étant situé hors du domaine de protection du firewall, le NIDS est alors exposé à d'éventuelles attaques pouvant le rendre inefficace.

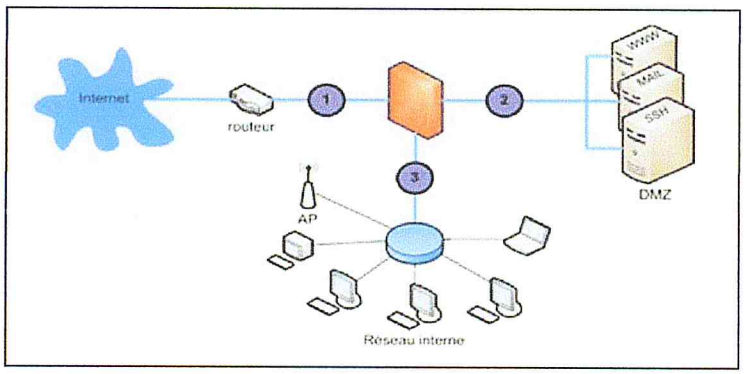


FIGURE 3.1 – Différentes positions possible de Snort dans un réseau informatique[26]

Sur la DMZ : dans cette position, la sonde peut détecter tout le trafic filtré par le Firewall qui atteint la zone DMZ. Cette position permet de surveiller les attaques dirigées vers les différents serveurs de l'entreprise accessible de l'extérieur. Sur le réseau interne : le positionnement du NIDS à cet endroit nous permet d'observer les tentatives d'intrusions

parvenues à l'intérieur du réseau d'entreprise ainsi celle parvenues de l'intérieur. Dans le cas d'entreprise utilisant largement l'outil informatique pour la gestion de leur activités ou de réseaux fournissant un accès à des personnes peu soucieuses de la sécurité (réseaux d'écoles et d'universités), cette position peut revêtir un intérêt primordial.

3.2.2 Architecture de Snort

L'architecture de SNORT est modulaire, elle se compose de : Décodeur de paquet : en anglais (Packet Decoder) il capture les paquets de données des interfaces réseaux, les prépare afin d'être prétraitées ou envoyées au moteur de détection. Pré processeur : en anglais (Pre processor) il s'agit d'améliorer les possibilités d'analyse, et de recomposition du trafic capturé. Ils reçoivent les paquets, les retraitent et les envoient au moteur de détection. Moteur de détection : en anglais (Detection Engine) c'est le composant le plus important de SNORT. Son rôle consiste à détecter les éventuelles intrusions qui existent dans un paquet. Pour se faire, le moteur de recherche se base sur les règles de SNORT. En effet, ce moteur consulte ces règles et les compare une à une avec le paquet de données. S'il y a conformité, le détecteur l'enregistre dans le fichier log et/ou génère une alerte. Sinon le paquet est laissé tomber. Système d'alerte et d'enregistrement des logs : en anglais (Logging and Alerting System) il permet de générer les alertes et les messages log suivant les résultats trouvés par le moteur de détection. Output modules : en anglais (plugins) permet de traiter l'intrusion générée par le système d'alertes et le noter de plusieurs manières : envoie vers un fichier log, génère un message d'alerte vers un serveur syslog, ou stocke cette intrusion dans une base de données comme MySQL ou Oracle.

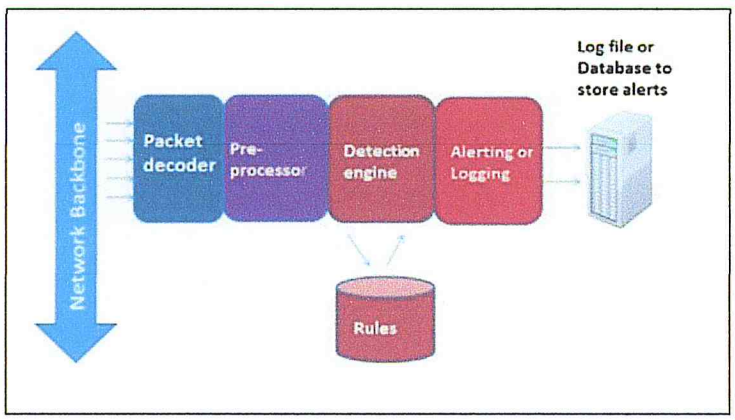


FIGURE 3.2 – Architecture de SNORT[27]

3.3 Paramétrage de SNORT

3.3.1 Préprocesseurs

Les préprocesseurs ont été introduits dans la version 1.5 de Snort. Ils permettent d'étendre la fonctionnalité de Snort en permettant aux utilisateurs et les programmeurs de déposer des plugins modulaires dans Snort assez facilement. Le code du préprocesseur est exécuté avant la détection, le moteur est appelé mais après que le paquet a été décodé. Une manière d'utiliser ce mécanisme .Le paquet peut être modifié ou analysé dans un hors-bande. Les préprocesseurs sont chargés et configurés à l'aide du mot clé du préprocesseur. Le format de la directive du préprocesseur dans le fichier de configuration de Snort est : préprocesseur <nom> : <options>.

3.3.1.1 Frag3

Le préprocesseur frag3 est un module de défragmentation IP basé sur la cible pour Snort. Frag3 est conçu avec les éléments suivants :

— Exécution rapide avec gestion de données moins complexe.

— Techniques de modélisation anti-évasion basées sur les cibles.

Frag3 utilise la structure de données sfxhash et les listes liées pour la gestion des données en interne ce qui lui permet d'avoir une performance prévisible et déterministe dans tout les environnements.

L'idée de base derrière un IDS basé sur la cible est qu'il puisse éviter les attaques d'évasion de style Ptacek et Newsham basées sur des informations concernant le fonctionnement d'une pile IP cible individuelle. Frag3 a été implémenté pour présenter et prototyper un module basé sur une cible .

— Configuration de Frag 3

Il y a au moins deux directives de préprocesseur nécessaires pour activer frag3, une directive de configuration globale et un moteur d'exécution. Il peut y avoir un nombre arbitraire de moteurs définis au démarrage

3.3.1.2 Configuration globale

- Nom du préprocesseur : frag3 global

- Options disponibles :

— Frags max <nombre> - Fragments simultanés maximum à suivre. La valeur par défaut est 8192.
— memcap <octets> - Bouchon de mémoire pour l'auto-préservation. La valeur par défaut est 4 Mo.
— prealloc memcap <octets> - mode de gestion de la mémoire alternative, utiliser des nœuds fragmentés pré-alloués basés sur un bouchon de mémoire (plus rapide dans certaines situations).
— prealloc frags <nombre> - Autre mode de gestion de la mémoire, utiliser des nœuds fragmentés pré-alloués (plus rapide dans certaines situations).

3.3.1.3 Configuration du moteur

- Nom du préprocesseur : moteur frag3

- les Options disponibles sont :

— timeout <secondes> - Timeout pour les fragments pour être automatiquement supprimé. La valeur par défaut est 60 secondes.
— min ttl <valeur> - Valeur TTL minimale acceptable pour un paquet de fragments. La valeur par défaut est 1. Les plages acceptées pour cette option est de 1 à 255.
— détecter les anomalies - Détecter les anomalies de fragment.

3.3.1.4 Format

Il est à noter dans la configuration avancée ci-dessous qu'il y a trois moteurs spécifiés. Les deux premiers moteurs sont liés à des plages d'adresses IP spécifiques et le dernier s'applique à toutes les autres circulations .

Configuration de base

— préprocesseur frag3_global

— préprocesseur frag3_engine

Configuration avancée

— préprocesseur frag3_global : prealloc nodes 8192

— préprocesseur frag3_engine : policy linux bind to 192.168.1.0/24

— préprocesseur frag3_engine : policy first bind to [10.1.47.0/24, 172.16.8.0/24]

— préprocesseur frag3_engine : policy last detect anomalies

3.3.1.5 Le préprocesseur stream

Le préprocesseur stream est un module de gestion de session de flux global pour Snort. Il est dérivé de la fonction stream de gestion qui faisait partie du préprocesseur Stream5. Puisque Le préprocesseur stream implémente une partie de la fonctionnalité et de l'API qui était auparavant dans Stream5.

API de stream

Le préprocesseur stream fournit une API pour permettre la création et la gestion du bloc de contrôle stream pour un flux et la gestion des données et de l'état pouvant être associés à ce flux (la plupart des ces fonctions étaient auparavant supportées par l'API Stream5). Ces méthodes sont appelées pour identifier les streams qui peuvent être ignorés (transferts de données volumineux, etc.)

Configuration globale de stream

Paramètres globaux pour le préprocesseur de stream.

$preprocessorstream5\_global$:
$[track\_tcp < yes|no >], [max\_tcp < number >],$
$[memcap < numberbytes >],$
$[track\_udp < yes|no >], [max_udp < number >],$
$[track\_icmp < yes|no >], [max\_icmp < number >],$
$[track\_ip < yes|no >], [max\_ip < number >],$
$[flush\_on_alert], [show_rebuilt\_packets],$
$[prune\_log_max < numberbytes >], [disabled],$
$[enable\_ha]$

3.3.1.6 preprocessor stream5_tcp

Le préprocesseur stream5_tcp est un module de réassemblage TCP basé sur la cible pour Snort. Il est capable de suivre les sessions pour TCP et UDP. Les sessions TCP sont identifiées via la "connexion" TCP classique. Les sessions UDP sont établies à la suite d'une série de Paquets UDP à partir de deux points de terminaison via le même ensemble de ports. Les messages ICMP sont suivis à des fins de vérification pour les messages inaccessibles et les messages de service indisponibles, qui terminent efficacement une session TCP ou UDP.

L'API stream5_tcp

stream5_tcp prend en charge l'API Stream modifiée qui se concentre désormais sur les fonctions spécifiques au réassemblage et au protocole.

Configuration du flux TCP

Il fournit un moyen pour configurer la stratégie TCP sur une cible d'adresse IP. Cela peut avoir plusieurs occurrences liées à une adresse IP ou un réseau. Une stratégie par défaut doit être spécifiée et cette stratégie n'est pas liée à une adresse IP ou un réseau.

$preprocessorstream5\_tcp$:
$[log\_asymmetric\_traffic < yes|no >]$,
$[bind\_to < ip\_addr >]$,
$[timeoul < numbersecs >], [policy < policy\_id >]$,
$[overlap\_limit < number >], [max\_window < number >]$,
$[require\_3whs[< numbersecs >]], [detect\_anomalies]$,
$[check\_session\_hijacking], [use\_static\_footprint\_sizes]$,
$[dont\_store\_large\_packets], [dont\_reassemble\_async]$,
$[max\_queued\_bytes < bytes >], [max\_queued\_segs < numbersegs >]$,
$[small\_segments < number > bytes < number > [ignore\_portsnumber[number]*]]$,
$[ports < client|server|both >< all|number|!number[number] * [!number]* >]$,
$[protocol < client|server|both >< all|servicename[servicename]* >]$,
$[ignore\_any\_rules], [flush\_factor < numbersegs >]$

La configuration du flux UDP

La configuration pour le suivi d'un stream UDP. Comme il n'y a pas de liaison basée sur la cible, il ne devrait y avoir qu'une seule occurrence de configuration UDP
preprocessor stream5_udp : [timeout <number secs>], [ignore_any_rules]

La configuration d'un Stream ICMP

La configuration pour le suivi d'un stream ICMP. Comme il n'y a pas de liaison basée sur la cible, il ne devrait y avoir qu'une seule occurrence de configuration ICMP.

préprocesseur stream5_icmp : [timeout <nombre secs>]

Stream IP Configuration

La configuration pour le suivi de stream IP. Comme il n'y a pas de liaison basée sur la cible, il ne devrait y avoir qu'une seule occurrence de configuration IP.

preprocessor stream5_ip : [timeout <number secs>]

3.3.2 Les plugins de sortie

Les modules de sortie sont une nouveauté de la version 1.6. Ils permettent à Snort d'être plus flexible dans le formatage et la présentation des sorties à ses utilisateurs. Les modules de sortie son exécutés quand les sous-systèmes d'alerte ou de journalisation de Snort sont appelés, après les préprocesseurs et le moteur de détection. Le format des directives dans le fichier de règles est très similaire à celui des préprocesseurs.
Plusieurs plugins de sortie peuvent être spécifiés dans le fichier de configuration de Snort. Quand plusieurs plugins du même type (journal, alert) sont spécifiés, ils sont "empilés" et appelés en séquence quand un événement se produit. Comme avec les systèmes standards de journalisation et d'alerte, les plugins de sortie envoient leurs données à /var/log/snort par défaut ou vers un répertoire désigné par un utilisateur (en utilisant l'option de la ligne de commande "-l").

Les modules de sortie sont chargés au moment de l'exécution en spécifiant le mot clé output dans le fichier de règles : output <nom> : <options>Exemple : output alert_syslog : LOG_AUTH LOG_ALERT

3.3.2.1 Alerte syslog (Alert_syslog)

Il envoie les alertes à la fonctionnalité syslog (comme l'option -s de la ligne de commande). Ce module permet également à l'utilisateur de spécifier la fonctionnalité de journalisation et la priorité dans le fichier de règles de Snort, en donnant aux utilisateurs une plus grande flexibilité dans la journalisation des alertes.

Un ensemble de Mots clés disponibles :
LOG_CONS, LOG_NDELAY, LOG_PERROR, LOG_PID, LOG_AUTH ,
LOG_AUTHPRIV, LOG_DAEMON, LOG_LOCAL, LOG_USER

Priorités :
LOG_EMERG, LOG_ALERT, LOG_CRIT, LOG_ERR, LOG_WARNING, LOG_NOTICE,
LOG_INFO, LOG_DEBUG

Format : alert_syslog : <fonctionallité> <priorité> <options>

Exemple : output alert_syslog : LOG_AUTH LOG_ALERT LOG_PID

3.3.2.2 Alerte rapide (Alert_fast)

Ceci imprimera les alertes de Snort dans un format rapide d'une ligne vers un fichier de sortie spécifié. C'est une méthode d'alerte plus rapide que les alertes full car elle n'a pas besoin d'afficher toutes les entêtes des paquets vers le fichier de sortie .

Format : alert_fast : <nom du fichier de sortie>

Exemple : output alert_fast : alert.fast

3.3.2.3 Alerte pleine (Alert_full)

Alert_full est un message d'alerte qui demande à Snort d'enregistrer l'intégralité des entêtes des paquets interceptés. Cette fonctionnalité d'alerte est généralement plutôt lente car elle requière que le programme fasse beaucoup plus d'analyse de données pour enregistrer les données à être imprimées. Les alertes seront écrites dans le répertoire de journalisation par défaut (/var/log/snort) ou le répertoire de journalisation spécifié sur la ligne de commande.

Format : alert_full : <nom du fichier de sortie>

Exemple : output alert_full : alert.full

3.3.2.4 Alerte smb

Ce plugin envoie des messages d'alerte WinPopup aux noms de machines NETBIOS indiqués dans le fichier spécifié comme argument à ce plugin de sortie. Il doit être noté que l'utilisation de ce plugin n'est pas encouragée puisqu'il exécute un exécutable binaire externe (smbclient) au même niveau de privilèges que Snort, communément root. Le format du fichier des stations de travail est une liste de noms NETBIOS des systèmes qui souhaitent recevoir les alertes, un par ligne dans ce fichier.

Format : alert_smb : <nom du fichier des stations de travail à alerter>

Exemple : output alert_smb : workstation.list

3.3.2.5 Alerte unixsock

Configure un socket du domaine UNIX et y envoie les rapports d'alertes. Des programmes / processus externes peuvent écouter cette socket et recevoir les alertes Snort et les données des paquets en temps réel. C'est actuellement une interface expérimentale.

Format : alert_unixsock Exemple : output alert_unixsock

3.3.2.6 Log_tcpdump

Le module log_tcpdump enregistre les paquets vers un fichier au format tcpdump. Ceci est utile pour effectuer des analyses post traitement sur le trafic collecté avec le grand nombre d'outils qui sont disponibles pour examiner des fichiers au format tcpdump. Ce module ne prend qu'un seul argument, le nom du fichier de sortie.

Format : log_tcpdump : <nom du fichier de sortie>

Exemple : output log_tcpdump : snort.log

3.3.3 Les bases de SNORT

Snort permet d'écrire des règles personnelles en utilisant un langage simple et léger de description de règles qui est flexible et assez puissant. Les règles Snort sont divisées en deux sections logiques, l'entête de la règle et les options de la règle :

L'entête de la règle : elle contient comme information l'action de la règle, le protocole, les adresses IP source et destination, les masques réseaux et les ports source et destination. Option de la règle : contient les messages d'alerte et les informations sur les parties du paquet qui doivent être inspectées pour déterminer si l'action de la règle doit être acceptée.

Exemple : alert tcp any any -> 192.168.1.0/24 111 (content :"|00 01 86 a5|" ;msg : "mountd access" ;)

3.3.3.1 Les inclusions

Le mot clé include permet à d'autres fichiers de règles d'être inclus dans le fichier de règles indiqué sur la ligne de commande de Snort. Il fonctionne beaucoup comme un "#include" du langage de programmation C, lisant le contenu de fichier nommé et les mettant en place dans le fichier à la place où l'include apparaît.

Format : include : <répertoire/nom du fichier include>

Exemple : #include $RULE_PATH/emerging-botcc-BLOCK.rules

3.3.3.2 Les variables

Des variables peuvent être définies dans Snort. Ce sont de simples substitutions des variables fixées avec le mot clé var .

Format : var : <nom> <valeur>
Exemple : var MY_NET [192.168.1.0/24,10.1.1.0/24]

Example d'utilisation d'un variable dans une alert
alert tcp any any -> $MY_NET any(flags : S ; msg : "SYN packet" ;)

3.3.4 Les entêtes de règle

3.3.4.1 L'action de règle

L'entête de règle contient l'information qui défini le "qui, où, et quoi" d'un paquet, ainsi que quoi faire dans l'événement où le paquet avec tous les attributs indiqués dans la règle devrait se présenter. Le premier élément dans une règle est l'action de règle. L'action de règle dit à Snort quoi faire quand il trouve un paquet qui correspond aux critères de la règle. Il y a cinq actions accessibles par défaut dans Snort, alert, log, pass, activate, et dynamic.

on peut aussi définir nos propres types de règles et associer un ou plusieurs plugins de sortie avec eux. De ce fait, on peut utiliser le type de règle comme action dans les règles Snort.

Cet exemple créera un type qui journalisera juste vers tcpdump :

```
ruletype suspicious
{
type log
output log_tcpdump : suspicious.log
}
```

Cet exemple créera un type de règle qui journalisera vers syslog et une base de données mysql :

ruletype redalert
{
type alert
output alert_syslog : LOG_AUTH LOG_ALERT
output database : log, mysql, user=snort dbname=snort host=localhost }

3.3.4.2 Les protocoles

Le champ suivant dans une règle est le protocole. Il y a trois protocoles IP que Snort analyse actuellement pour des comportements suspicieux, tcp, udp, et icmp. Dans le futur il pourra y en avoir plus, tels que ARP, IGRP, GRE, OSPF, RIP, IPX, etc.

3.3.4.3 Les adresses IP

La section suivante de l'entête de règle s'occupe comme information de l'adresse IP et du port pour une règle donnée. Le mot clé "any" peut être utilisé pour définir n'importe quelle adresse. Snort n'a pas de mécanisme pour fournir de la résolution de nom pour le champ de l'adresse IP dans le fichier de règles. Les adresses sont formées par une simple adresse IP numérique et un bloc CIDR. Le bloc CIDR indique le masque réseau qui doit être appliqué à l'adresse de la règle et à tout paquet qui est testé par rapport à la règle. Un masque de bloc CIDR de /24 indique un réseau de classe C, /16 un réseau de classe B, et /32 indique l'adresse spécifique d'une machine. Par exemple, la combinaison d'adresse/CIDR 192.168.1.0/24 devrait signifier le bloc d'adresses de 192.168.1.1 à 192.168.1.255. Toute règle qui utilise cette désignation pour, disons, l'adresse destination devrait correspondre à toute adresse dans cet intervalle. Les désignations CIDR nous donnent une façon rapide de désigner de larges espaces d'adresses avec juste quelques caractères.
Il y a un opérateur qui peut être appliqué aux adresses IP, l'opérateur de négation. Cet opérateur dit à Snort de correspondre à toute adresse IP sauf celles indiquées par la liste d'adresses IP. L'opérateur de négation est indiqué par un " !". Par exemple, une modification facile à l'exemple initial est de le faire alerter sur tout trafic qui provient de l'extérieur du réseau local avec l'opérateur de négation :

alert tcp !192.168.1.0/24 any -> 192.168.1.0/24 111 (content : "|00 01 86

a5|" ; msg : "external mountd access" ;)

Ces adresses IP de la règle indiquent "tout paquet tcp avec une adresse source ne provenant pas du réseau interne et à destination du réseau interne".
Vous pouvez aussi spécifier des listes d'adresses IP. Une liste IP spécifiée en entourant une liste d'adresses IP et de blocs CIDR séparés par des virgules entre crochets. Pour le moment, une liste IP ne peut pas inclure d'espaces entre les adresses. un exemple de liste IP en action :

alert tcp ![192.168.1.0/24,10.1.1.0/24] any -> [192.168.1.0/24,10.1.1.0 /24] 111 (content : "|00 01 86 a5|" ; msg : "external mountd access" ;)

3.3.4.4 Les numéros de port

Les numéros de ports peuvent être spécifiés de nombre de façons, incluant "any" (ndt : tous les ports), des définitions de ports statiques, des intervalles et des négations. Les ports "any" sont les valeurs génériques, signifiant littéralement tous les ports. Les ports statiques sont indiqués par un seul numéro de port, tel que 111 pour portmapper, 23 pour telnet, ou 80 pour http, etc. Les intervalles de ports sont indiqués avec l'opérateur d'intervalle " :". L'opérateur d'intervalle peut être appliqué dans nombre de façons pour prendre différentes significations :

log udp any any -> 192.168.1.0/24 1 :1024 (journalise le trafic udp provenant de tout port et à destination de ports dans l'intervalle de 1 à 1024) **log tcp any any -> 192.168.1.0/24 :6000** (journalise le trafic tcp depuis tout port et allant vers les ports inférieurs ou égaux à 6000) **log tcp any :1024 -> 192.168.1.0/24 :500** (journalise le trafic tcp depuis les ports privilégiés inférieurs ou égaux à 1024 allant vers les ports supérieurs ou égaux à 500)

La négation de port est indiquée en utilisant l'opérateur de négation " !". L'opérateur de négation peut être appliqué à tous les autres types de règles (excepté "any", qui se traduirait en "n'import qui"). Par exemple, si pour quelque raison tordue on veut tout journaliser sauf les ports X Windows :

log !192.168.1.0/24 any <> 192.168.1.0/24 23

3.3.4.5 L'opérateur de direction

L'opérateur de direction "->" indique l'orientation, ou la "direction", du trafic auquel la règle s'applique. L'adresse IP et les numéros de ports du côté gauche de l'opérateur de direction est considéré comme le trafic provenant du système source, et les informations d'adresse et de port du côté droit de l'opérateur est le système destination. Il y a aussi un opérateur bidirectionel, qui est indiqué par le symbole "<>". Ceci dit à Snort de considérer les paires adresse/port soit en source ou bien en destination de l'orientation. C'est utile pour enregistrer/analyser les deux côtés d'une conversation, tel que des sessions telnet ou POP3. Un exemple de l'opérateur bidirectionel étant utilisé pour enregistrer les deux côtés d'une session telnet :

log !192.168.1.0/24 any <> 192.168.1.0/24 23

3.3.4.6 Les règles activate/dynamic

Les paires de règles activate/dynamic donnent à Snort une capacité puissante. Vous pouvez maintenant avoir une règle qui en active une autre pour un nombre fixé de paquets quand son action est accomplie. Ceci est très utile si vous voulez configurer Snort pour effectuer l'enregistrement de ce qui suit quand une règle spécifique "se désactive". Les règles d'activation se comportent juste comme les règles alert, excepté qu'elles ont un champ d'option *obligatoire* : "activates". Les règles dynamiques se comportent juste comme les règles log, mais elles ont un champ d'option différent : "activated_by". Les règles dynamiques ont également un second champ obligatoire, "count". Quand la règle "activate" se désactive, elle active la règle dynamique qui lui est liée (indiquée par les numéros d'option activates/activated_by) pour "count" paquets (50 dans ce cas).

activate tcp !$HOME_NET any -> $HOME_NET 143(flags : PA ; content : "|E8C0FFFFFF| bin| ; activates : 1 ; msg : "IMAP buffer overflow !" ;)
dynamic tcp !$HOME_NET any -> $HOME_NET 143 (activated_by : 1 ; count : 50 ;)

Ces règles disent à Snort d'alerter quand il détecte un débordement de tampon dans IMAP et collecte les 50 prochains paquets destinés au port 143 provenant de l'extérieur de $HOME_NET et destinés à $HOME_NET. Si le débordement de tampon arrive et a réussi, il y a de très bonnes possibilités que des données utiles seront contenues dans

les prochains 50 (ou quel que soit) paquets allant au même port de service sur le réseau, ainsi il y a avantage à collecter ces paquets pour une analyse ultérieure.

3.4 Simulation d'attaques :

la configuration de l'IDS snort et l'écrire de ses règle comme nous avons vus n'est pas si simple, On doit apprendre comment écrire ses règle qui doive répondre aux exigences de sécurité de l'entreprise et la forme d'écriture, la plupart des alerte snort ne sont pas activer parce on doit les modifier selon nous exigence. On doit procéder à une série de simulation d'attaque avec des outils d'attaques et des logiciel de capture tel que wirshark qui va nous permettre d'avoir des signature.
Dans ce qui suit, nous allons présenter les outils d'attaques en général et LOIC en particulier, ainsi que l'outil de capture Wireshark.

3.4.1 les outils d'attaques

Une attaque DoS est une tentative malveillante de la part de plusieurs systèmes pour rendre les ressources informatiques ou réseau indisponibles aux utilisateurs prévus, généralement en interrompant ou en suspendant des services connectés à Internet.
Il y a un certain nombre d'outils disponibles qui peuvent générer le trafic légitime similaire et trafic d'attaque et peuvent facilement contourner les solutions de défense DoS existantes. Il a été observé que toutes les attaques DoS sont lancées de nos jours en utilisant des botnets. Ce qui met en évidence les principales caractéristiques techniques des outils d'attaque DoS et des générateurs de trafic utilisés par les attaquants pour lancer des attaques DoS comme leur modèle d'architecture, le type de protocoles supportés ou le type de trafic généré. La figure 3.3 donne une vue générale sur la taxonomie des outils d'attaque DoS.

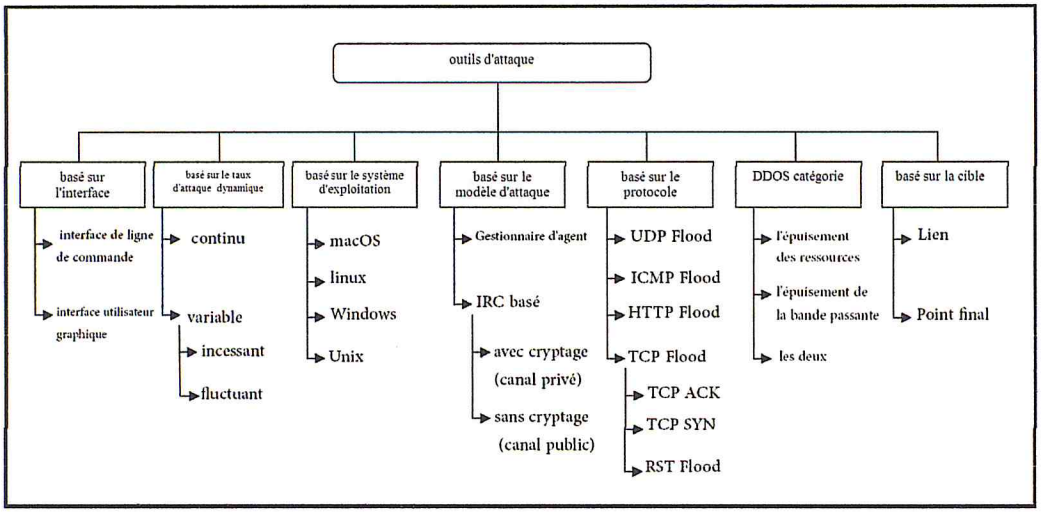


FIGURE 3.3 – Taxonomie des outils d'attaque DoS [28]

3.4.2 Taxonomie des outils d'attaque DoS et leur comparaison

Dans cette section, nous décrivons la taxonomie des outils d'attaques DoS(voir Figure 3.3).

La taxinomie des outils d'attaque DoS est basée sur les attributs comme le type d'interface utilisée, leur dynamique de taux d'attaque, le système d'exploitation cible, le modèle d'attaque, les protocoles utilisés, Catégorie d'attaque DoS et zone cible.

Type d'interface utilisé : L'interface utilisée par les outils d'attaque DoS peut être une interface de ligne de commande ou une interface utilisateur graphique. Goldeneye, trinoo, arbre etc. utilisent l'interface de ligne de commande alors que hoic, udp ooder, xoic etc. utilisent l'interface utilisateur graphique.

Dynamique du taux d'attaque : En fonction de la dynamique du taux d'attaque, les outils d'attaque peuvent soit générer le taux d'attaque continu (pas de variation de la demande d'attaque) et le taux d'attaque variable (l'outil peut faire varier le taux d'attaque le taux croissant et le taux d'uctuation).

Système d'exploitation pris en charge : Un certain nombre d'outils d'attaque DoS sont conçus pour prendre en charge les différents systèmes d'exploitation tels que

l'Unix, Linux, Solaris ou Windows.

Modèle d'attaque : les outils d'attaque DoS peuvent utiliser un modèle d'agent-manipulateur ou un modèle IRC. Agent-Handler est basé sur une relation maître-esclave alors que le système IRC utilise des canaux publics pour lancer des attaques.

Protocole : Le type de protocole spécifie le type de trafic généré par les outils d'attaque pour générer des attaques, la communication entre l'agent-gestionnaire, le gestionnaire-client et l'agent-client. Les attaques par inondation utilisent principalement les protocoles UDP, ICMP (requête ICMP ECHO et réponse ICMP ECHO), HTTP, TCP (TCP-SYN, TCP-ACK et RST-ood).

Catégorie d'attaque DoS : La conséquence d'une attaque DoS est l'indisponibilité des ressources ou de la bande passante de la victime. Par conséquent, les attaquants utilisent ces outils d'attaque qui peuvent épuiser les ressources et la bande passante du système cible ou du réseau. Il existe un certain nombre d'outils d'attaque DoS disponibles qui peuvent épuiser à la fois les ressources et la bande passante en un rien de temps.

Zone cible : les attaques DoS peuvent congestionner le lien ou le point de terminaison. Ainsi, les outils d'attaque DoS sont généralement conçus pour l'encombrement au niveau de la liaison (encombrement sur le réseau victime) ou au niveau du point final (encombrement sur le serveur victime).
Voici quelques outils d'attaque :

Pour atteindre notre objectif dans ce mémoire, nous allons décrire l'outil d'attaques LOIC et nmap qu'on a choisi pour faire des simulations d'attaque, et le logiciel Wireshark qui nous a permis de faire la capture du trafic et de récupérer la signature.

a LOIC (Low Orbit Ion Cannon) :

Est l'un des outils d'attaque DOS les plus puissants disponibles gratuitement. Il est devenu largement utilisé, Cette application tente d'attaquer par déni de service le site ciblé en inondant le serveur avec des paquets TCP, des paquets UDP, des requêtes HTTP avec

Logiciel	type d'attaques
LOIC	UDP/TCP/TCP SYN flooding
Trinoo	UDP flooding
Trible Flood Network (TFN) et TFN2K	UDP/TCP/TCP SYN flooding, Smurf
Stacheldraht	UDP/TCP/TCP SYN flooding, Smurf
schaft	UDP/TCP/ICMP flooding
MStreamT	ACK flooding

TABLE 3.1 – Outils d'attaques

l'intention de perturber le service d'un hôte particulier. Y compris dans certaines attaques très médiatisées contre les serveurs PayPal, Mastercard et Visa . Cet outil a également été l'arme de choix mise en œuvre par le célèbre groupe de pirates informatiques Anonymous, qui a revendiqué de nombreuses attaques de piratage de haut niveau, parmi lesquelles des hacks contre Sony, le FBI et d'autres agences de sécurité américaines. Le groupe a non seulement utilisé cet outil, mais a également demandé que d'autres le téléchargent et rejoignent des attaques anonymes via IRC [16].

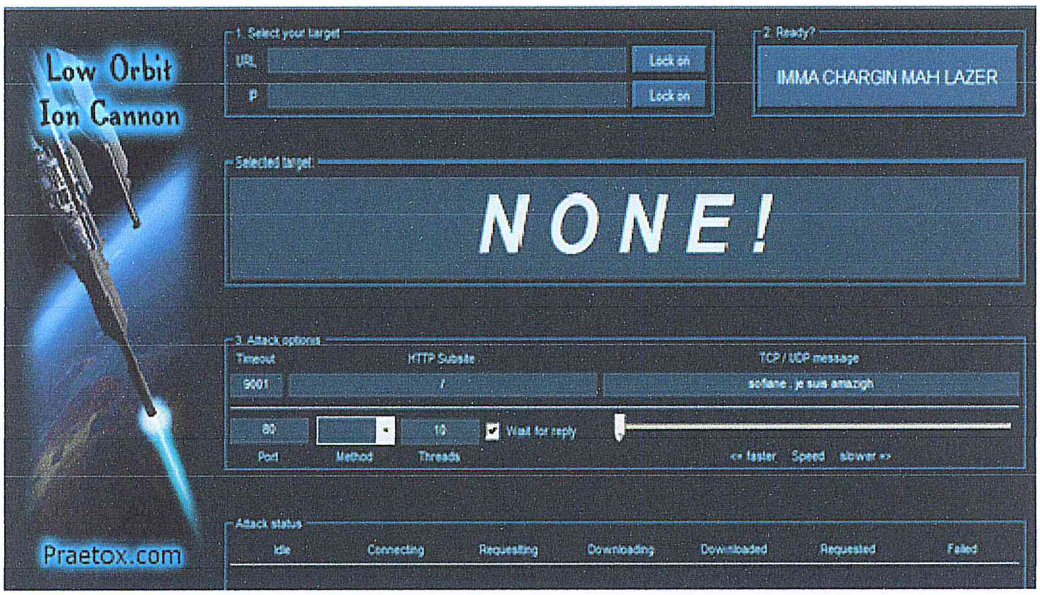


FIGURE 3.4 – Capture de l'interface de LOIC

Voici la signification de chaque champ :

Select your target (Selection la cible) :"URL . IP " : dans ce champ en écrire le nom domaine ou l'adresse ip de notre cible .
Attack options (Attaque option) : En choisir la méthode d'attaque (tcp,udp http) , le numéro de port et la vitesse (le nombre des paquet envoie pas second) et le message dans le paquet envoyer .

IDLE : il montre le nombre de threads inactifs. Il devrait être nul pour une plus grande efficacité de l'attaque.

Connexion : ceci indique le nombre de threads qui tentent de se connecter au serveur victime.

Demander : Cela montre le nombre de threads qui demandent des informations sur le serveur de la victime.

Téléchargement : ceci montre le nombre de threads qui lancent un téléchargement pour des informations du serveur.

Téléchargé : ce nombre indique combien de fois le téléchargement de données a été lancé à partir du serveur victime sur lequel vous attaquez.

b Nmap

Est un scanner de réseau. Il permet de savoir quels sont les ports ouverts, fermés ou filtrés, ainsi que le système d'exploitation autorisé et sa version.
Il permet par exemple de scanner un ensemble d'adresses IP en précisant la méthode de scan utilisée, les types de ports tels que les ports UDP, en tentant d'identifier la machine cible et en sauvegardant le résultat dans un fichier.

c. Wireshark

Pour étudier les paquets qui circulent dans le réseau, on utilise Wireshark qui capture les paquets. Une fois Wireshark est lancé, la fenêtre de la capture ce présente sur la figure suivante :

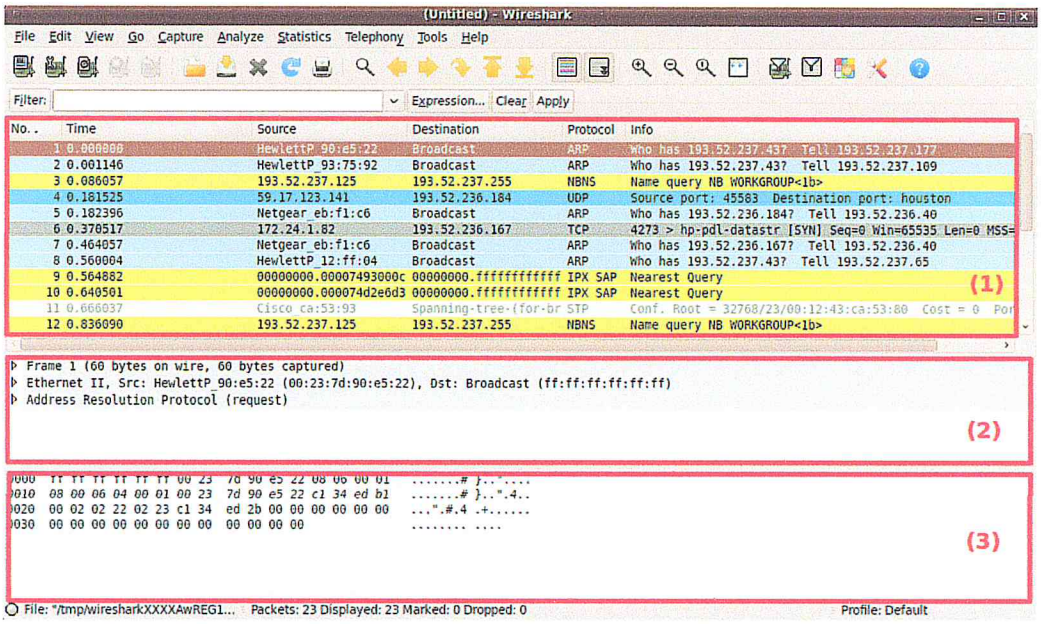


FIGURE 3.5 – interface de l'analyseur Wireshark

L'interface de l'analyseur est découpé en trois zone :

1. Zone numérotée (1) sur figure 3.5 : liste l'ensemble des paquets capturé
2. Zone numérotée (2) sur figure 3.5 : affiche le détail d'un paquet sélectionné
3. Zone numérotée (3) sur figure 3.5 : présente l'ensemble du paquet sous forme octale et ASCII.

Encapsulation d'un paquet

Lorsqu'un paquet est sélectionné, la zone centrale permet de visualiser clairement les différentes couches d'encapsulation du paquet. Par exemple si l'on sélectionne un paquet de type UDP , la zone centrale pourrait afficher quelque chose de similaire à ce qui est

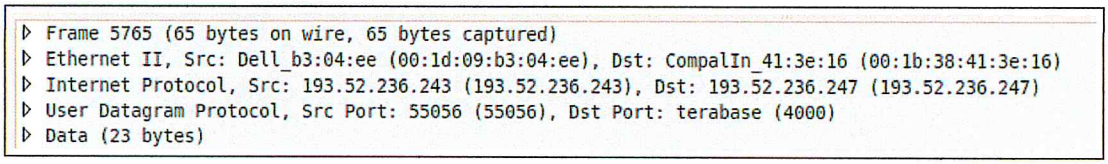


FIGURE 3.6 – Encapsulation d'un paquet UDP, zone centrale de l'analyseur

présenté Figure 3.6. Les 5 entrées présentées correspondent à différentes encapsulations, ordonnées de la couche la plus basse à la couche la plus haute :

1. Données sur le média de capture : Wire = filaire sur Figure 3.6
2. Trame relative à la couche liaison de donnée : Ethernet II sur Figure 3.6
3. Paquet relatif à la couche réseau : Internet Protocol sur Figure 3.6
4. Datagramme relatif à la couche transport : User Datagram Protocol sur Figure 3.6
5. Données de l'application : regroupe généralement les couches session, présentation, application.

Détail de chaque niveau d'encapsulation

Pour tout item correspondant à un niveau d'encapsulation, un clic sur le triangle en début de ligne permet de dérouler l'en-tête afin de voir l'ensemble des champs le composant. Certains champs peuvent également être déroulés. Sur l'exemple présenté en Figure 3.7 , nous avons étendu les entrées correspondant aux couches réseau, transport et application en cliquant sur les triangles correspondants. Nous pouvons voir entre autre que :

- Le paquet est de type IP v4 : ref (2) sur Figure 3.7
- Le type de données de ce paquet IP est un datagramme UDP : ref (5) sur Figure 3.7
- L'ip de la machine source est 193.52.236.243 : ref (6) sur Figure 3.7
- L'ip de la machine destination est 193.52.236.247 : ref (7) sur Figure 3.7

Nous pouvons également faire un point sur la taille des données et des en-tetes à différents niveaux d'encapsulation :

- La taille des données envoyée par le processus est de 23 octets : ref (9) sur Figure 3.7
- La taille totale du datagramme UDP est de 31 octets - ref (8) sur Figure 3.7 - . Cette

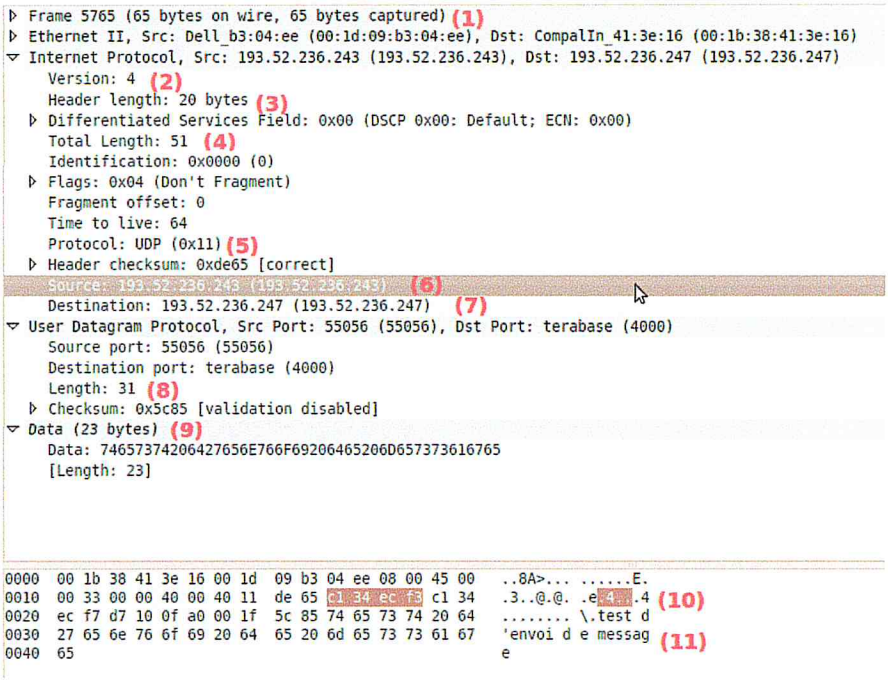


FIGURE 3.7 – Encapsulation d'un paquet UDP, zone centrale de l'analyseur

valeur est la somme entre la taille réelle des données (23 octets) et 8 octets d'en-tête du paquet (8 étant une valeur fixe pour un paquet UDP)

• La taille des en-têtes du paquet IP est de 20 octets : ref (3) sur Figure 3.7

• Le paquet IP contient un en-tete (20 octets) ainsi que le datagramme UDP (31 octets). Sa taille totale est de 51 octets, taille rappelée en ref (4) sur Figure 3.7

• La taille totale du paquet IP est de (flèche). Cette valeur somme la taille du paquet UDP à la taille de l'en-tête IP.

• Si l'on ajoute 12 octets d'en-tete pour la couche Ethernet II (taille fixe), la taille totale de la trame est de 65 octets, comme présentée en ref (1) sur Figure 3.7.

Notons ainsi que pour transférer 23 octets de données brutes, il nous a fallu transférer au total 65 octets (en fait il nous a même fallu transférer des octets supplémentaires avant la trame Ethernet. Ces octets seront ici passés sous silence).

3.4.3 Visualisation des paquets TCP

Dans cette section, nous allons étudier le cas normal de protocole TCP.

a- Visualisation de paquet TCP normal

On prend un exemple d'une connexion entre deux PC à la base de protocole TCP, illustré sur la figure suivante :

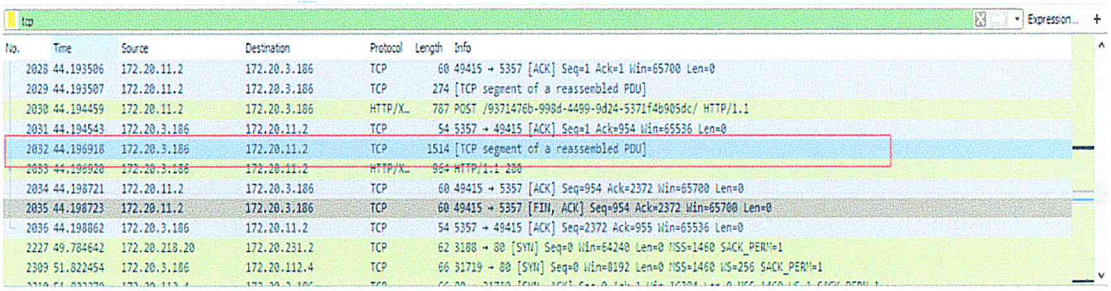


FIGURE 3.8 – La capture des paquets TCP normale

Pour plus de détails, on a pris une autre capture au niveau d'encapsulation :

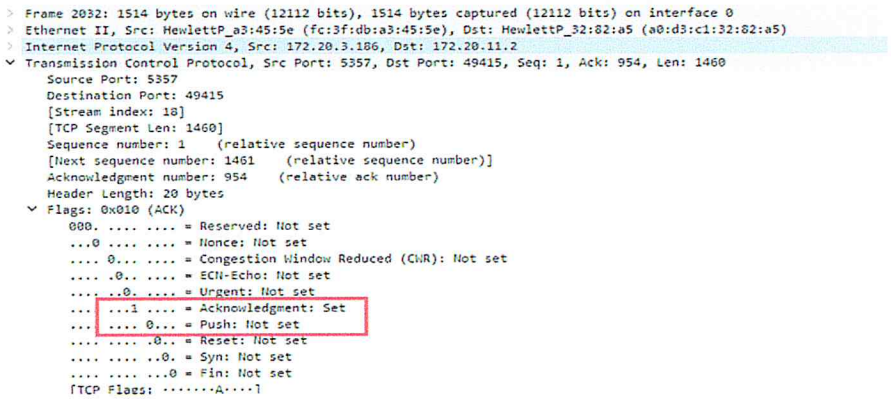


FIGURE 3.9 – La capture détaillée de paquet TCP

— Notre pc : IP = 172.20.3.186 (classe b)

— Pc destination : IP=172.20.11.2

Lors d'une connexion TCP dans un réseau local de notre université, on note Notre Pc répond par un TCP de flag ack lorsque le message est reçu.
Dans la figure 3.10 : les paquets TCP entrants ont des flags push non positionnés et ack=1.

Ce qui concerne le flux de donné, Wireshark permet d'analyser le flux sous forme d'un graphe, illustré dans la figure suivante :

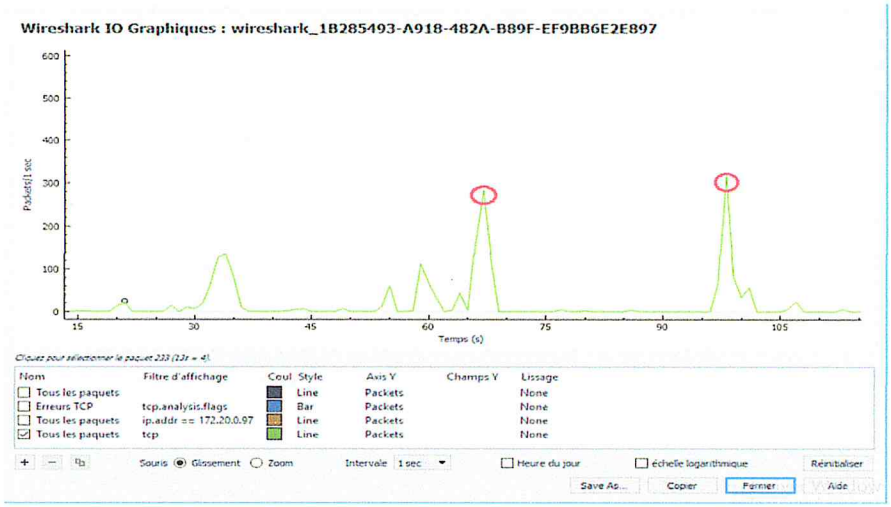


FIGURE 3.10 – graphe du flux de donné

3.5 Conclusion

ce chapitre nous a permis d'avoir une vue générale de la configuration de snort et de la méthode d'écriture des règle snort ainsi que les outils d'attaque qu'on va utiliser dans le chapitre suivant. Faire une simulation des attaques permettra à l'administrateur snort d'avoir une expérience dans l'écriture des règles de snort et de la méthode de récupération de signature.

dans le chapitre suivant, nous présentons l'ensemble des règles proposées afin de détecter les attaques DoS.

Chapitre 4

Simulation des attaques DoS et règles proposées

4.1 Introduction

L'écriture d'une règle SNORT et très difficile et exige une manipulation et une compréhension des attaques ainsi qu'une compression de la méthode d'écriture des règles SNORT, pour les attaque DoS on ne retrouve aucune écriture pour ce type d'attaque dans les règle SNORT.

pour comprendre les événements réseau, Chaque trafic en peut le considérer comme une attaque , Dans notre travail, on analyse premièrement le trafic avec l'outil Wireshark qui nous aidera à identifier les signatures lors de la simulation .
Dans un deuxième temps et à partir des signatures et des règles de sécurité appliquées ,nous créons l'ensemble des règles pour concevoir les attaques DoS et les tester après l'intégration dans la base de Snort.

4.2 Méthode de travail

Nous avons simulé les attaques dans un environnement réel LAN, constitué de deux pc qui son relié à un switch via un câble torsadé :

Les caractéristiques des éléments utilisésdans cette topologie les suivants :

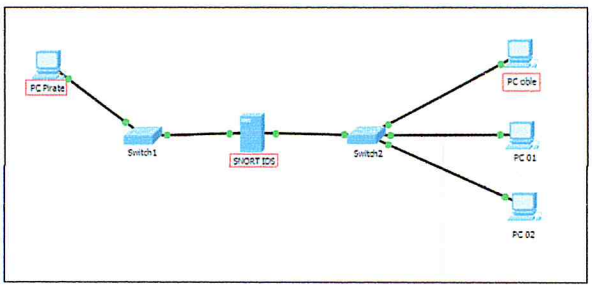


FIGURE 4.1 – schéma de simulation des attaques DoS

— pc pirate : IP 172.20.60.200 (system d'exploitation windows 7).
— pc cible : IP 172.20.55.100 (system d'exploitation windows 7).
— PC SNORT (system d'exploitation windows 7).
— Câble torsadé .
— switch .

Ainsi on a utilisé LOIC comme logiciel d'attaque, qui génère un ensemble d'attaque DoS qui influence l'utilisation de la mémoire, le processeur et le réseau.

LOIC ouvre plusieurs connexions au serveur cible en envoyant une suite continue de messages qui peuvent être définis à partir de l'option de paramètre de message TCP /UDP disponible sur l'outil. Dans les attaques TCP et UDP, la chaîne est envoyée en texte brut mais dans l'attaque HTTP, elle est incluse dans le contenu d'un message HTTP GET.

Cet outil continue d'envoyer des requêtes au serveur cible ; après un certain temps, le serveur cible devient surchargé. De cette façon, le serveur cible ne pourra plus répondre aux demandes des utilisateurs légitimes, ce qui le rendra inaccessible .

Une fois l'attaque est lancé, l'analyseur de paquet Wireshark peut observer qu'elles sont les paramètres causant la perturbation de système cible.

4.2.1 Méthode de la simulation d'attaque

La simulation se réalise en trois étapes afin de simuler des attaques DOS. Exécuter LOIC en première étape, dans la deuxième étape on choisit les paramètres appropriés en

validant l'adresse IP de la cible et choisir la méthode d'attaque ainsi le port et la quantite des paquets envoyes par second (speed) et en lance l'attaque. et En parallèle et comme une troisième étape on lance l'outil Wireshark pour visualiser les paquets.

4.3 Les règles de détection des attaques

Afin de créer des règles pour l'outil de détection d'intrusion Snort, ces règles nous permettent de détecter les attaques DOS de types différentes (TCP, UDP, HTTP, Ping of death) à partir des signatures récuperer a partir de l'analyse du trafic.

Nous détaillons chacune dans les sous sections suivantes.

4.4 Attaque ICMP

4.4.1 Visualisation des paquets ICMP

4.4.1.1 Visualisation des paquets ICMP (Ping)

Le Ping utilise le protocole ICMP, ce dernier est utilisé pour tester la connexion entre les pc. La figure (Figure 4.2) illustrent le Ping et ces paquets envoyées d'hôte source de l'adresse 172.20.60.200 vers l'hôte destinateur de l'adresse 172.20.55.100 :

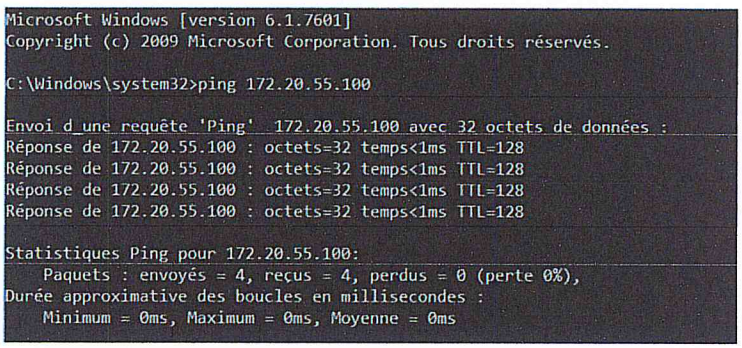


FIGURE 4.2 – Les paquets envoyés par Le Ping

Pour analyser les données envoyées par le Ping on utilise l'outil Wireshark, illustré dans la Figure 4.3 :

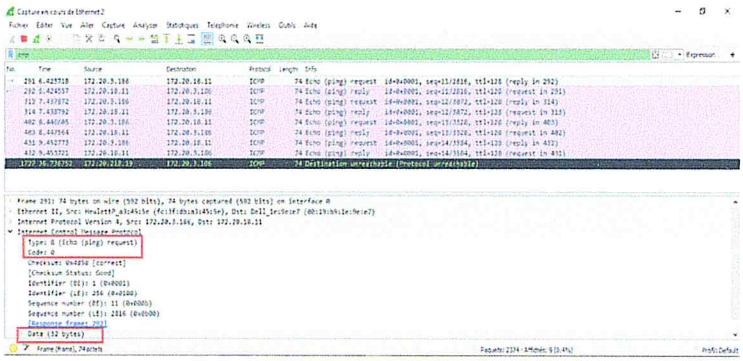


FIGURE 4.3 – Paquets ICMP capturés par Wireshark

On remarque que le Ping utilise le protocole ICMP qui envoie des messages de type 8, le code 0, ainsi que sa longueur est de 32 octets.

4.4.1.2 Visualisation des paquets de ping of death

cette attaque consiste a inonder le client avec très grand nombre de Ping dans un petit axe de temps et pour réaliser cette attaque, la première étape consiste à ouvrir l'invité de commande (cmd).L'adresse IP de la machine victime est 172.20.55.100, et celle de la machine pirate est 172.20.60.20. On utilise la commande suivante :

Ping 172.20.55.100 –t –l 65500 pour pinge la cible.

-t : un paramètre qui permet de répéter le Ping en boucle.

-l : un paramètre qui détermine la taille de paquet envoyé.

Le résultat de cette attaque est illustré dans la figure 4.4 :

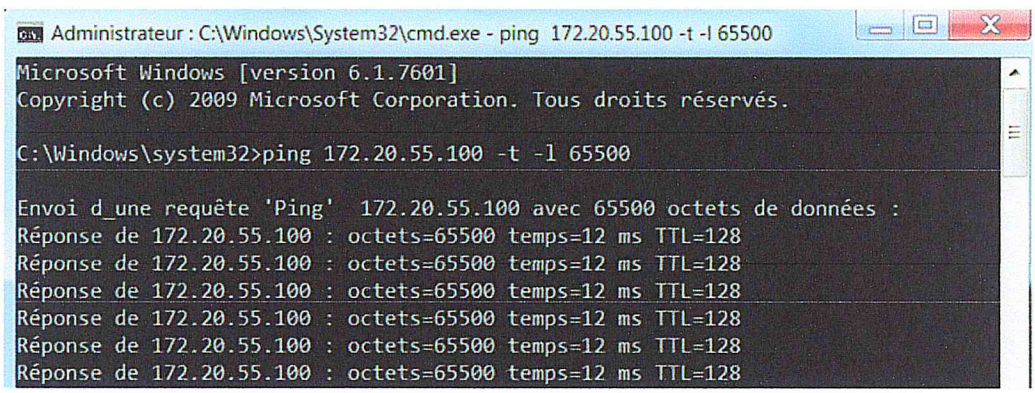


FIGURE 4.4 – Le Ping of death

Cette commande permet d'envoyer des paquets ICMP qui faites des demandes de connexions et de tester réseau.La figure 4.5 illustre le trafic envoyé depuis la machine pirate vers la cible.

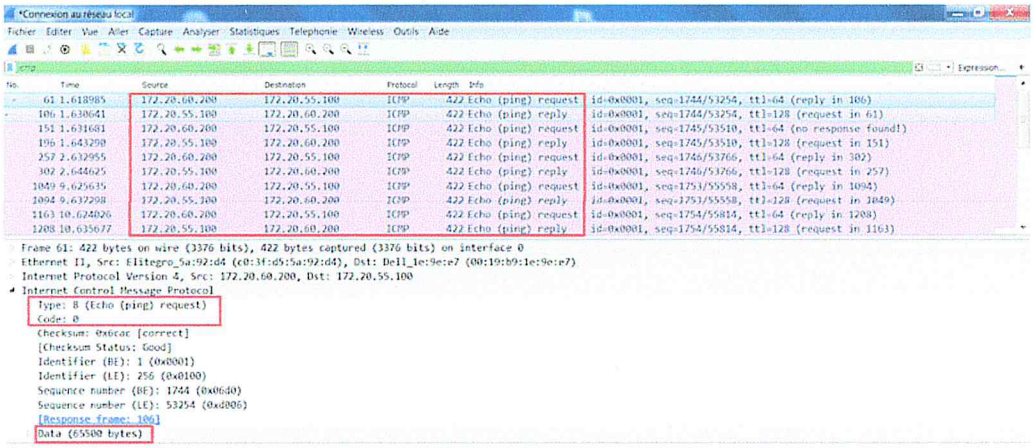


FIGURE 4.5 – Le traffic Ping of death capturé par Wireshark

En utilisant l'analyseur Wireshark, si on compare le ping of death illustré dans la figure4.4 avec le Ping normale illustré dans lafigure 4.2 ; on trouve qu'on a dépassé la taille autorisée (32bytes), plusieurs requêtes de longueurs de 65500 octets.

Et on note aussi que dans le cas de Ping normale le nombre totale de messages ICMP est 8 (4 messages request et 4 messages echoreply) , contrairement au Ping normale le ping of death dépend du nombre de messages envoyes(>8 messages) et de la taile des des paquets. le tableau 3.6 illustre la différence entre ces deux trafic en taille de données .

	Cas de ping normal	Cas de l'attaque ping of death	Différence
La taille de paquet	32 octets	65500 octet	Ping of death supérieur celle de ping normal

TABLE 4.1 – La déférence entre ping et ping of death

4.4.2 Détection de l'attaque Ping of death

D'après l'analyse et la comparaison entre les données d'un trafic Ping et un trafic Ping of death, nous avons définit la règle suivante pour détecter l'attaque UDP de type DOS.

Alert icmp any any ->any any (msg : "attack ping of death" ;
disize>1000 ; itype :8 ; icode :0 ; sid :100000007 ; rev :1 ;)

Alert : Une alerte «attack Ping of death » est générée si les critères sont réunis.
anyany ->anyany : de n'importe quel adresse IP source et port à n'importe quel adresse, port destination.
Type 8 et code 0 :demande de renvoi d'informations (Ping).
Itype : teste le champ type ICMP contre la valeur de Ping (max 1000 octets).
Icode : teste le champ code ICMP contre la valeur de Ping.
Disize :d'après les résultats , on prend la valeur maximum=1000 octets car le Ping (32 octets) ne dépasse pas cette valeur.
Msg : affiche le msg (attack Ping of death) dans les alertes.
Sid : 100000007 c'est l'identifiant de notre signature dans la base de signature.

Pour vérifie La figure (4.6) montre la détection de l'attaque ping of death par l'outil de détection Snort on utilisons l'interface graphique BASE :

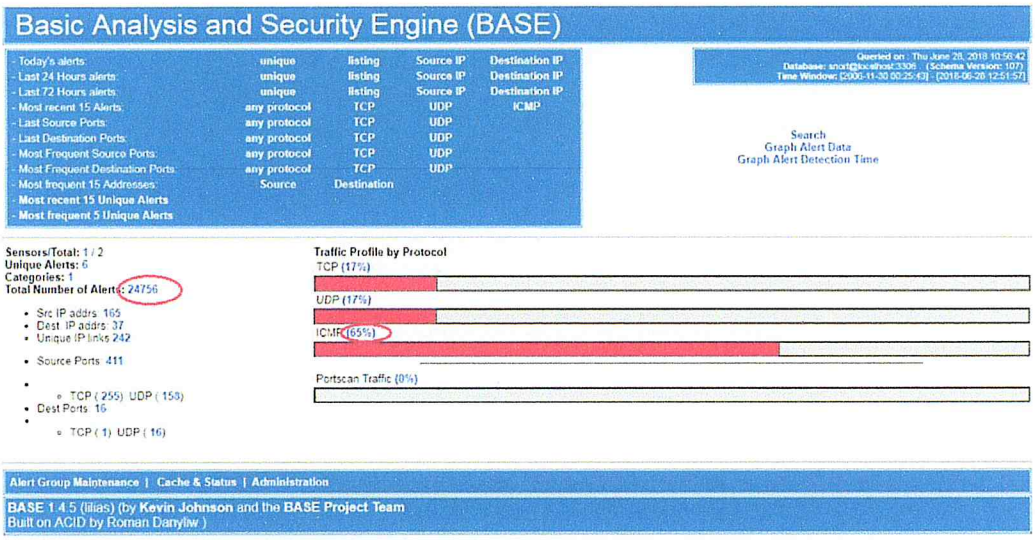
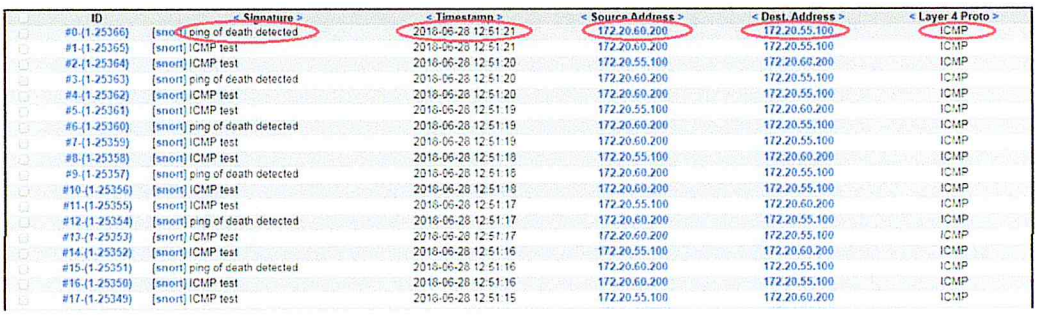


FIGURE 4.6 – Détection de l'attaque Ping of death

Nous remarquons que le pourcentage de détection d'un trafic ICMP a augmenté La figure suivante illustre les données détaillées de Ping of death :



ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
#0-(1-25366)	[snort] ping of death detected	2018-06-28 12:51:21	172.20.60.200	172.20.55.100	ICMP
#1-(1-25365)	[snort] ICMP test	2018-06-28 12:51:21	172.20.60.200	172.20.55.100	ICMP
#2-(1-25364)	[snort] ICMP test	2018-06-28 12:51:20	172.20.55.100	172.20.60.200	ICMP
#3-(1-25363)	[snort] ping of death detected	2018-06-28 12:51:20	172.20.60.200	172.20.55.100	ICMP
#4-(1-25362)	[snort] ICMP test	2018-06-28 12:51:20	172.20.60.200	172.20.55.100	ICMP
#5-(1-25361)	[snort] ICMP test	2018-06-28 12:51:19	172.20.55.100	172.20.60.200	ICMP
#6-(1-25360)	[snort] ping of death detected	2018-06-28 12:51:19	172.20.60.200	172.20.55.100	ICMP
#7-(1-25359)	[snort] ICMP test	2018-06-28 12:51:19	172.20.60.200	172.20.55.100	ICMP
#8-(1-25358)	[snort] ICMP test	2018-06-28 12:51:18	172.20.55.100	172.20.60.200	ICMP
#9-(1-25357)	[snort] ping of death detected	2018-06-28 12:51:18	172.20.60.200	172.20.55.100	ICMP
#10-(1-25356)	[snort] ICMP test	2018-06-28 12:51:18	172.20.60.200	172.20.55.100	ICMP
#11-(1-25355)	[snort] ICMP test	2018-06-28 12:51:17	172.20.55.100	172.20.60.200	ICMP
#12-(1-25354)	[snort] ping of death detected	2018-06-28 12:51:17	172.20.60.200	172.20.55.100	ICMP
#13-(1-25353)	[snort] ICMP test	2018-06-28 12:51:17	172.20.60.200	172.20.55.100	ICMP
#14-(1-25352)	[snort] ICMP test	2018-06-28 12:51:16	172.20.55.100	172.20.60.200	ICMP
#15-(1-25351)	[snort] ping of death detected	2018-06-28 12:51:16	172.20.60.200	172.20.55.100	ICMP
#16-(1-25350)	[snort] ICMP test	2018-06-28 12:51:16	172.20.60.200	172.20.55.100	ICMP
#17-(1-25349)	[snort] ICMP test	2018-06-28 12:51:15	172.20.55.100	172.20.60.200	ICMP

FIGURE 4.7 – La capture détaillée de la détection de ping of death

Le message d'attaque Ping of death montre clairement qu'il y'a une attaque de type ICMP Ping of death, Snort nous informe aussi de la date de l'attaque et de l'adresse source .

4.4.3 Attaque TCP

4.4.3.1 Visualisation des paquets d'attaque TCP via LOIC

Étape 1 : exécution de l'outil.

Étape 2 : on a entré l'adresse IP 172.20.55.100 de notre cible dans le champ IP et cliqué sur (lock on) ensuite la méthode d'attaque (TCP) et enfin on choisit le port 80 .

Étape 3 : on envoie la chaine de message par défaut (A cat is fine tooDesudesudesu) avec une vitesse maximale (faster), après on clique sur le BigButton intitulé "IMMA CHARGIN MAH LAZER". Nous venons d'attaquer la cible.

la figure 4.8 illustre les paramètres utilisés pour lancer une attaque TCP de type DOS.

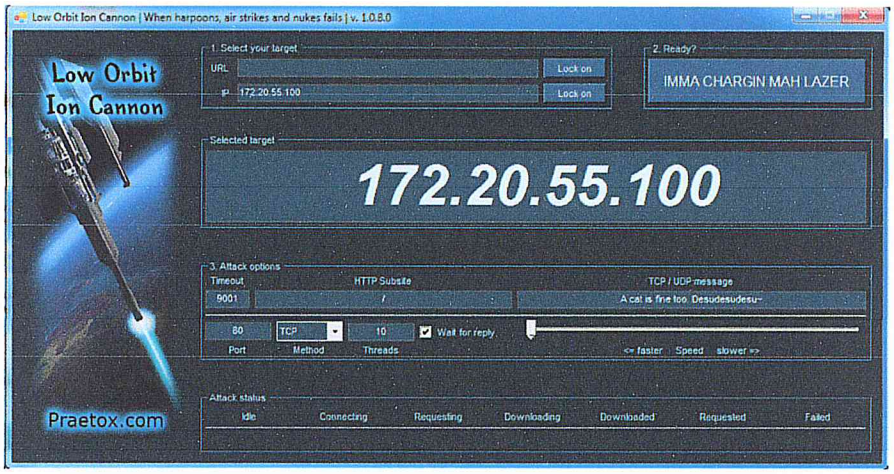


FIGURE 4.8 – les paramètres d'une attaque TCP LOIC

Durant l'attaque, on exécute Wireshark pour analyser l'attaque comme il est montré dans la figure 4.9 :

.

L'adresse IP de la machine pirate est : IP=172.20.60.60

L'adresse IP de la machine cible est : IP=172.20.55.100

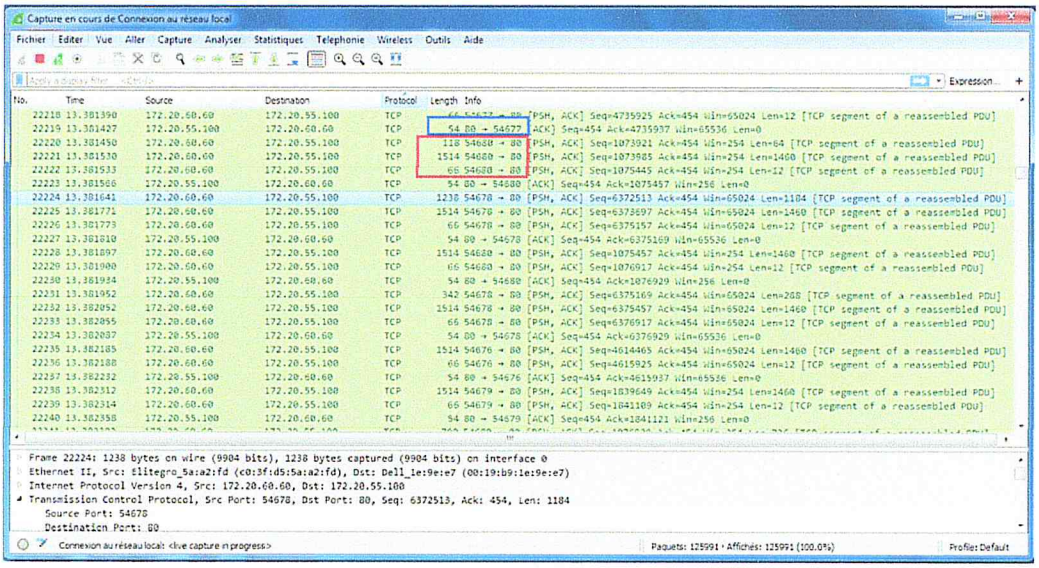


FIGURE 4.9 – les données capturées de l'attaque TCP par l'outil Wireshark

On remarque pour chaque paquet envoyé vers le pc cible répond par un TCP de flag ack et qu'il ouvre plusieurs connexion (entouré en bleu figure 4.9) à partir de différent port (par exemple 54677, 54680).

```
▾ Transmission Control Protocol, Src Port: 6473, Dst Port: 80, Seq: 417, Ack: 1, Len: 192
    Source Port: 6473
    Destination Port: 80
    [Stream index: 4]
    [TCP Segment Len: 192]
    Sequence number: 417    (relative sequence number)
    [Next sequence number: 609    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Header Length: 20 bytes
  ▾ Flags: 0x018 (PSH, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 1... = Push: Set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·····AP···]
```

FIGURE 4.10 – Capture détaillé d'un paquet TCP

Lorsqu'un émetteur TCP envoie un paquet avec le flag push égale à 1, le résultat est que les données TCP sont immédiatement envoyées ou "poussées" au récepteur TCP.

Dans la figure 4.11 suivante, on peut visualiser le message envoyé par notre pc d'attaque via LOIC

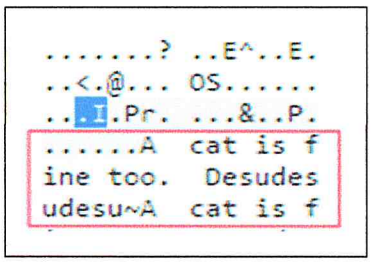


FIGURE 4.11 – Capture de message (en ascii) ennoyé par LOIC

Le graphe suivant montre le flux de données de l'attaque TCP :

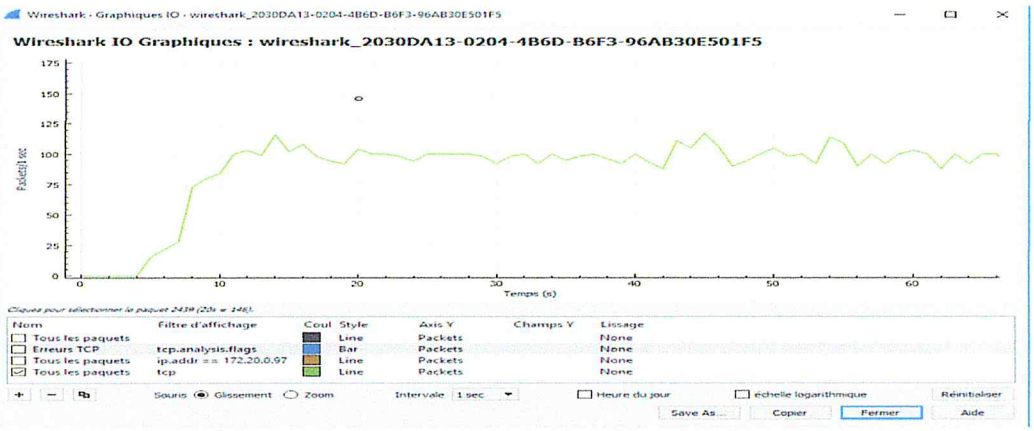


FIGURE 4.12 – Le nombre des paquet TCP pandans une attaque TCP

On note que la moyenne de l'envoi des paquets est de 100 paquets par second pendant la durée d'attaque.

D'après notre visualisation des paquets de protocole TCP normal et celles des attaques, nous avons définit la signature de cette attaque, comme il est illustrée dans le tableau 4.2 une comparaison entre le trafic TCP normal et le trafic générer par LOIC .

	Cas d'un paquet TCPnormale	Cas d'un paquet d'attaque de TCP	La difference entre les deux cas
Le nombre de paquets par second	25 jusqu'à 300 (d'une durée de 2 à 5 seconde	La moyenne des paquets se stabilise entoure de 100	L'attaque tcp se stabilise le nombre de paquet TCP reçus Entoure de 100/s par contre dans le cas de tcp normale elle se stabilise au niveau de 25 et peut atteint 300 dans des durées très courtes
Les flags	Flag PUSH=0 Flag ACK=1	Flag PUSH=1 Flag ACK=1	Le flag PUSH est positionné (égale à 1)

TABLE 4.2 – La déférence entre les deux cas de protocole TCP(les signatures)

4.4.3.2 L'impact de l'attaque TCP

On remarque sur la figure 4.13, que le niveau de l'utilisation de processeur et de réseau est presque nul dans le cas où il n'y'a pas d'intrusion (attaque).
Par contre sur la figure 4.14, on note une grande augmentation du niveau de l'utilisation de processeur et réseau (réseau est saturé).
Cette attaque force réellement le serveur récepteur pour vider son tampon de pile TCP et d'envoyer une reconnaissance (acknowledgement) lors de déroulement de cette action, ce qui entraîne une condition de déni de service par une attaque TCP flood.

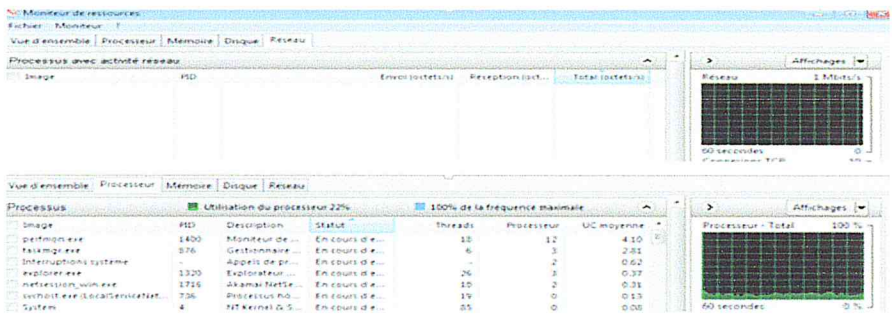


FIGURE 4.13 – Capture del'utilisation du processeur et réseau de TCP normal

Le cas où l'ordinateur subit une attaque TCP :

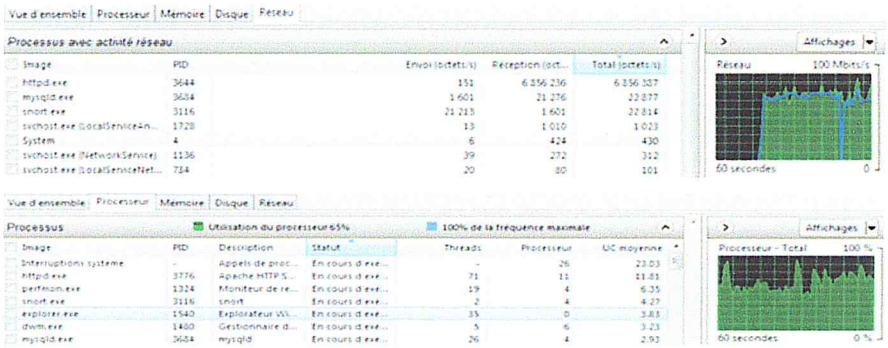


FIGURE 4.14 – Capture de l'utilisation du processeur, réseau de l'attaque TCP

4.4.3.3 Détection de l'attaque TCP

L'analyse des données des attaques TCP de type DOS nous permet de définir une règle de signature pour détecter les attaques de déni de service TCP.

Alert tcp any any ->any any (msg : "LOIC Dos attaque TCP mod " ; flag :ap ; threshold : type threshold, track bay_src, count :100, second :3,sid :100000004 ; rev :1 ;)

Alert : Une alerte « LOIC Dos attackTCPmod » est générée si les critères sont réunis.
anyany ->anyany : de n'importe quel adresse IP source et port à n'importe quel adresse, port destination.
Msg : affiche le msg (LOIC Dos attack TCP) dans les alertes.
Flags : le paquet est caractérisé par PSH et ACK.
Threshold : indiquez les alertes de seuil (100 paquet/s).
Trackbay_src : le taux est suivi soit par IP source, soit par l'adresse IP de destination. Les ports ou toute autre chose ne sont pas suivis.
Count :d'après les résultats on a choisi leseuil= 100 paquet/s,le count c'est le nombre de paquet qu'il ne faut pas dépasser.

Second :3 secondes c'est périodes sur laquelle le compte est accumulé.
Sid : 100000004 c'est l'identifiant de notre signature dans la base de signature.
Rev : 1 est utilisé pour identifier de manière unique les révisions des règles Snort. Permet de remplacer les signatures par des mises à jour.

Après la création de cette règle dans le fichier local.rules, on lance l'attaque pour tester la règle :

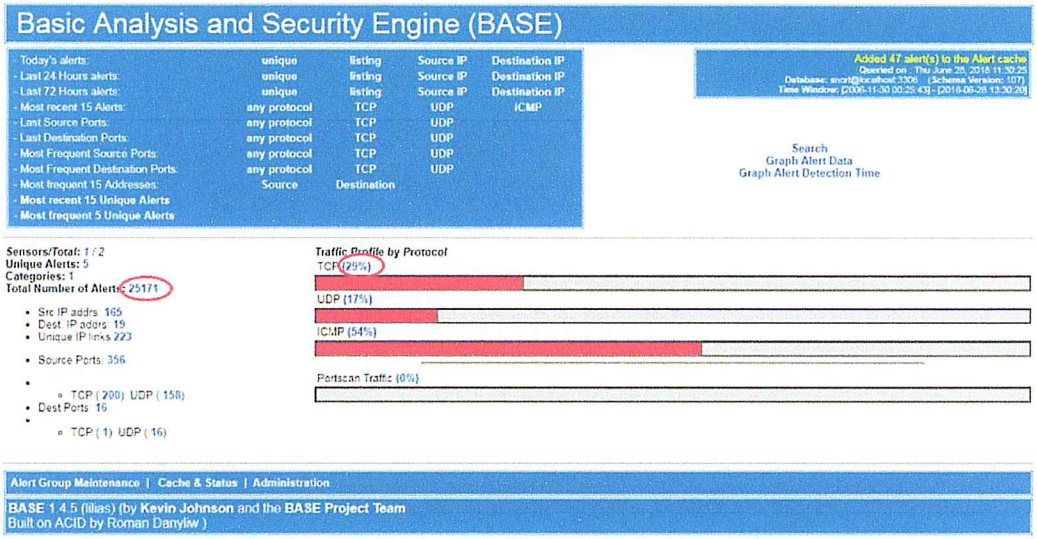


FIGURE 4.15 – Détection de l'attaque TCP

On remarque bien que Snort a détecté l'attaque TCP à partir d'augmentation du pourcentage de nombre des paquets TCP et des alertes.
Les informations plus détaillées sur l'attaque sont présentées sur la figure suivante :

Snort nous avise qu'il y'a une attaque de type TCP, l'outil utilisé par cette attaque (LOIC). Cette dernière utilise plusieurs ports vers une destination unique port 80, Snort nous informe aussi de la date d'attaque.

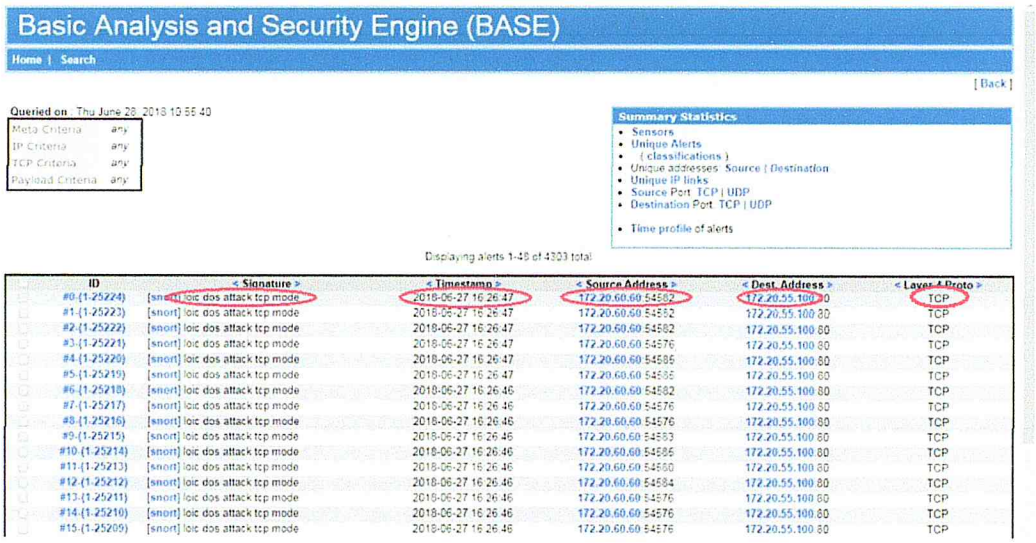


FIGURE 4.16 – informations détaillées de l'attaque TCP

4.5 Attaque UDP

4.5.1 Visualisation des paquetsUDP

On va entamer la même procédure qu'on a fait dans la partie de visualisation de paquet TCP pour pouvoir obtenir la signature de l'attaque UDP, toujours avec l'utilisation d'analyseur de paquet Wireshark .

4.5.1.1 Visualisation de paquet UDP normale

Prenant comme exemple les paquets de données d'une vidéo surYouTube, on lance la vidéo.Ensuite,on utiliseWiresharkpour capturer les paquets, voir la figure suivante :

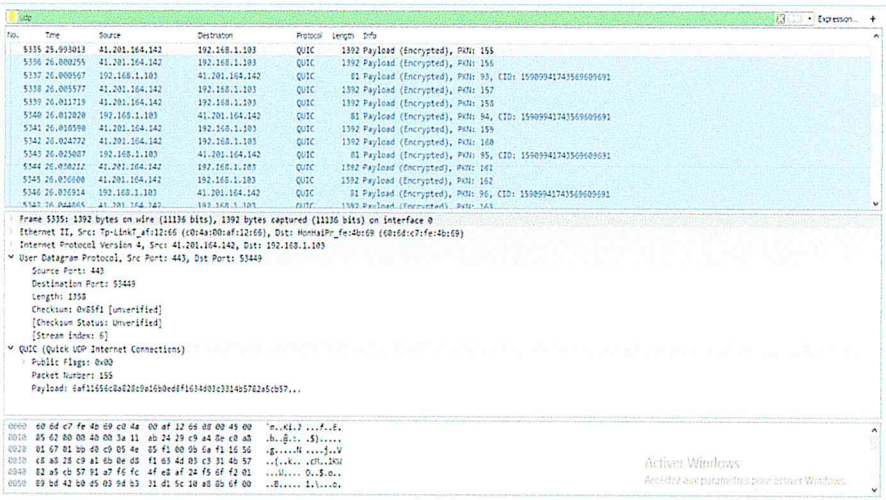


FIGURE 4.17 – Paquet UDP d'une vidéo de youtube

Notre machine : IP = 192.168.1.103,Server YouTube : IP=41.201.164.142
La figure 4.17 montre la synchronisation entre notre machine et le serveur YouTube en envoyant une ouverture de demande de connexion.

On voit bien que pour lancer une vidéo sur youtube on utilise le protocole UDP, illustré sur la figure 4.17.

Le graphe suivant montre le flux de données de la vidéo du YouTube :

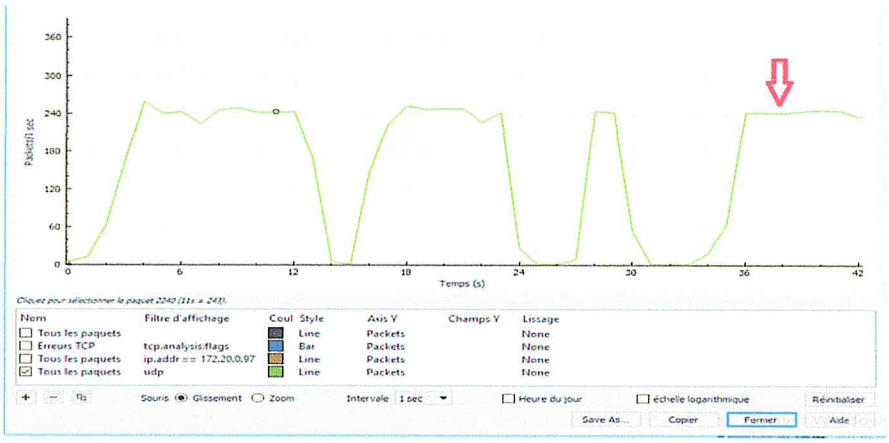


FIGURE 4.18 – Flux de donnée d'une vidéo de youtube

On remarqué que le nombre de la quantité des paquets ne dépasse pas 250 paquets par seconde durant toute la vidéo.

4.5.1.2 Visualisation des paquets d'attaque UDP

Dans ce type d'attaque DOS, un serveur est inondé de paquets UDP. Contrairement au protocole TCP, il n'y a pas de processus de communication entre client et hôte, cela rend plus difficile pour les mécanismes défensifs d'identifier une attaque d'inondation UDP. Cette méthode est similaire à l'attaque TCP. On Sélectionne le type d'attaque comme UDP.

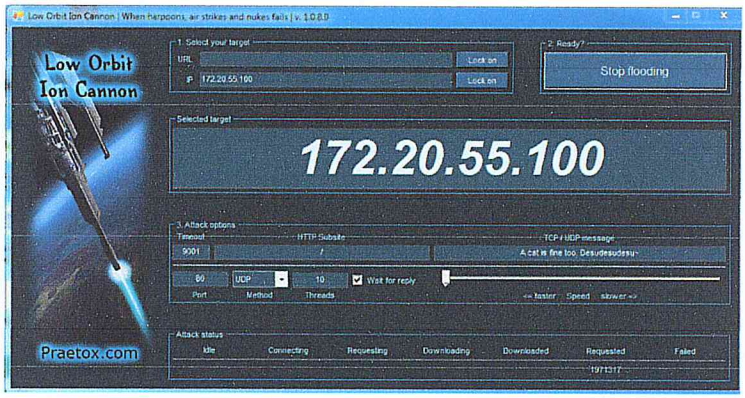


FIGURE 4.19 – L'attaque UDP

On exécute le Wireshark pour capturer les paquets d'attaque UDP, illustré dans la figure suivante :

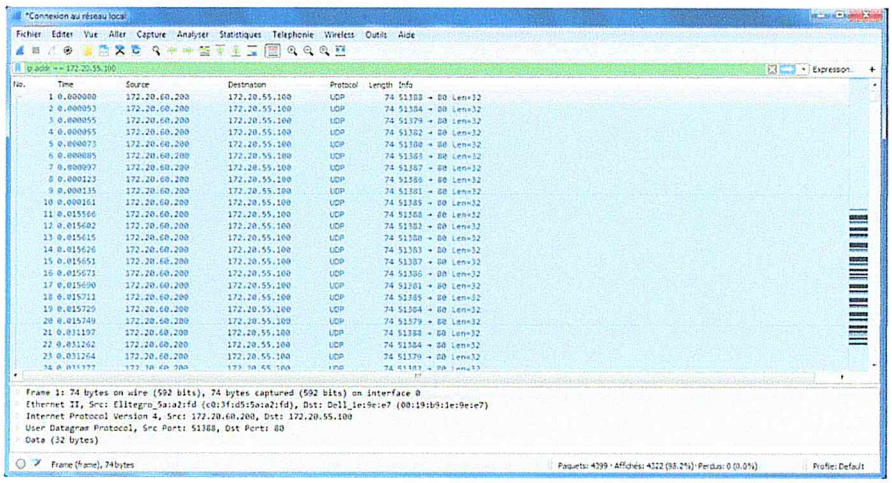


FIGURE 4.20 – d'un paquet d'attaque UDP

L'adresse IP de la machine est 172.20.60.200, et celle de la machine victime est 172.20.55.100 Cette capture nous montre qu'il y'a plusieurs messages UDP envoyés ce qui peut provoquer un déni de service.

Le graphe suivant montre le flux de données de l'attaque UDP :

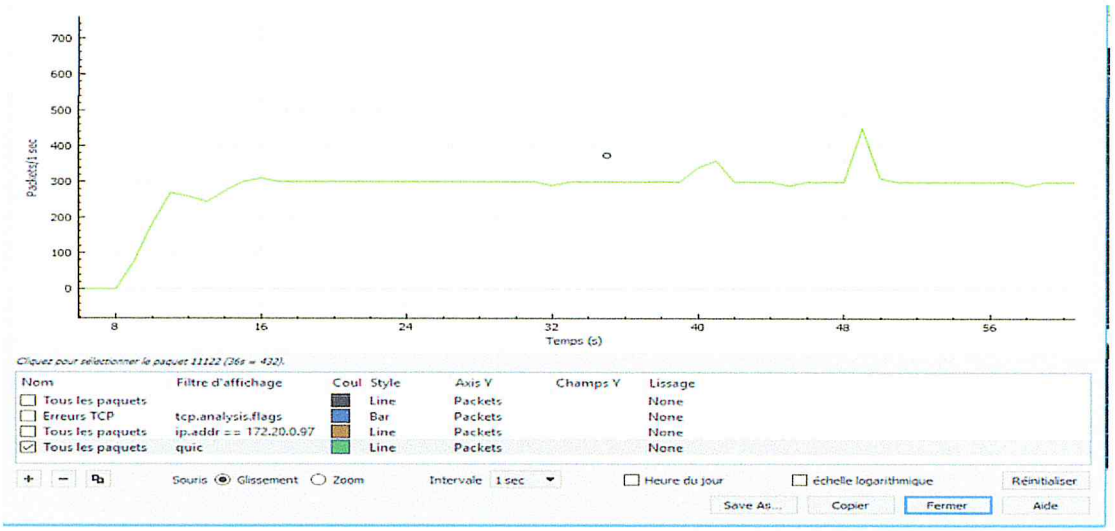


FIGURE 4.21 – Le flux de données de l'attaque UDP

La chose qui attire notre attention durant l'attaque UDP est le nombre des paquets par seconde qui est fixé à environ 300 paquets.

D'après notre visualisation des paquets de protocole UDP normal et celles des attaques, nous avons définit la signature de cette attaque UDP, illustré dans le tableau suivant :

	Cas d'un paquet UDP normal	Cas d'un paquet d'attaque UDP	La différence entre les deux cas
Le nombre des paquets pas seconds	Ne dépasse pas 250 paquets/s .	Elle se stabilise à 300 paquets/s	L'attaque UDP dépasse le nombre du paquet normal

TABLE 4.3 – La différence entre une connexion UDP normale et une attaque UDP de type DOS

4.5.1.3 L'impact de l'attaque UDP

Après avoir appliqué l'attaque UDP, nous avons remarqué une augmentation dans la consommation des ressources de la machine cible (réseau et processeur) comme il est illustré dans les figure 4.22 et 4.23.

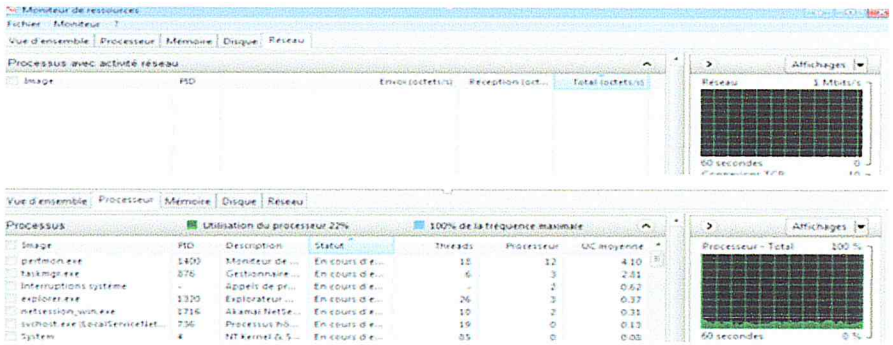


FIGURE 4.22 – L'utilisation du processeur et réseau avant l'attaque

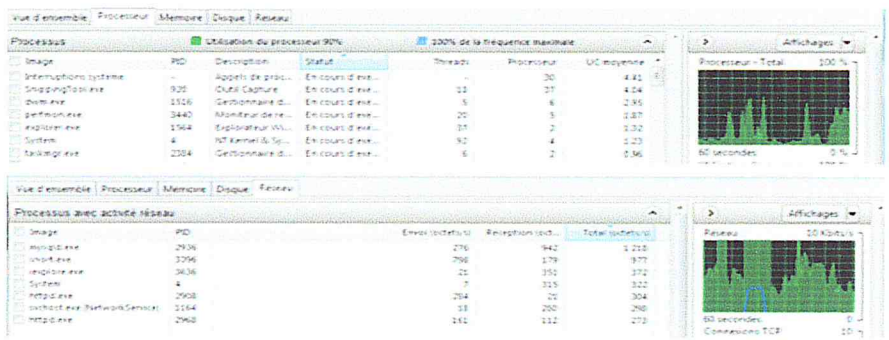


FIGURE 4.23 – L'utilisation du processeur et réseau après l'attaque UDP

Cette action engendre un grand changement au niveau de moniteur de ressources qui se manifeste par une énorme augmentation sur l'échelle de processeur et réseaux.

4.5.1.4 Détection de l'attaque UDP

L'analyse des données des de l'attaque UDP de type DOS nous a permet de définir une règle afin de détecter ce type d'attaque. La règle qui définit l'attaque UDP est la suivante :

Alert udp any any ->any any (msg : "LOIC Dos attaque UDP mod " ; threshold : type threshold, track bay_src, count :300, second :3 , sid :100000003 ; rev :1 ;)

Alert : Une alerte « LOIC Dos attack UDP mod » est générée si les critères sont réunis.
Msg : affiche le msg (LOIC Dos attack UDP mod) dans les alertes.
Threshold : indiquez les alertes de seuil (300 paquet/s).
Trackbay_src : le taux est suivi soit par l'adresse IP source, soit par l'adresse IP de destination. Les ports ou toute autre chose ne sont pas suivis.
Count : avons choisi le seuil de limite= 300 paquet/s, le count c'est le nombre de paquet qu'il ne faut pas dépasser.
Second :3 secondes c'est périodes sur laquelle le compte est accumulé.
Sid : 100000003 c'est l'identifiant de notre signature dans la base de signature.
Après l'intégration la règle de l'attaque UDP dans le fichier local.rules , on lance l'attaque pour tester son efficacité :

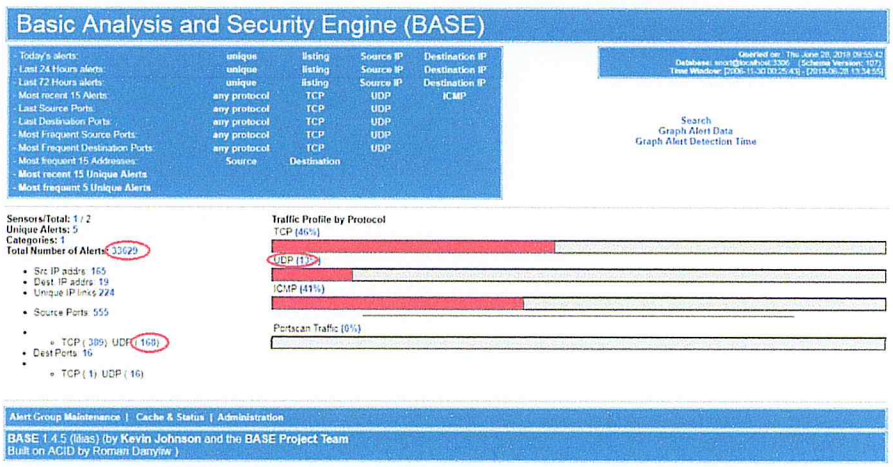


FIGURE 4.24 – La détection de l'attaque UDP

On remarque bien sur la figure 4.24 qu'il y une argumentation au niveau des paquets UDP et le nombre des alertes générées, ce qui confirme que Snort a détecté l'attaque UDP.

Des informations détaillées sur l'attaque UDP sont présentées dans la figure 4.25 : La

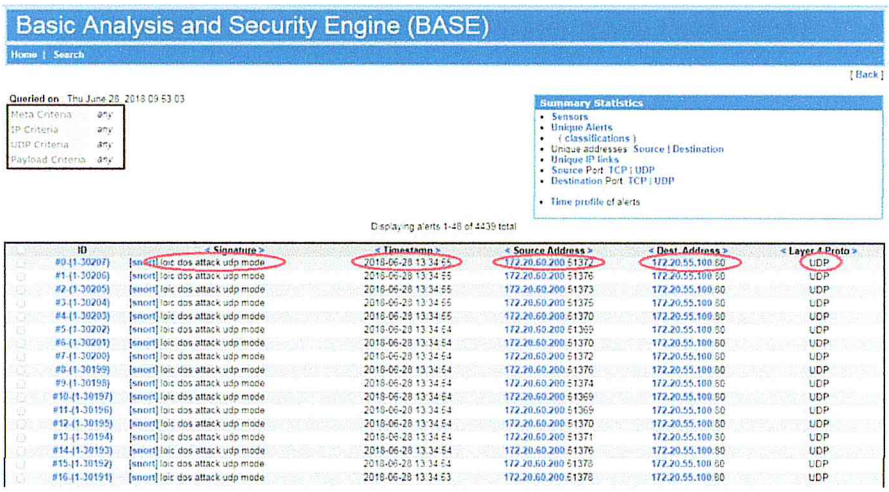


FIGURE 4.25 – l'attaque UDP présenté par Snort

signature nous signale qu'il y'a une attaque de type UDP, l'outil utilisé par cette attaque est LOIC. Cette attaque utilise plusieurs ports vers une destination unique port 80.

4.6 Attaque http

4.6.1 Visualisation des paquets HTTP

4.6.1.1 Visualisation le paquet HTTP normale

ce protocole permit de ce connecter a internet avec le port 80 généralement . Prenant comme exemple une recherche sur le moteur de recherche Google, présenté sur la figure suivante :

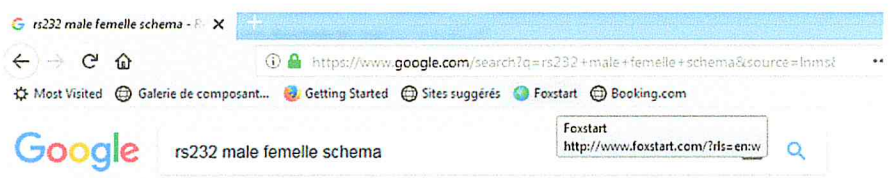


FIGURE 4.26 – recherche sur google

l'interception de la recherche par wirchark nous permit de voir le porotocole utiliser http/1.1 comme on le vois dans la figure 4.27

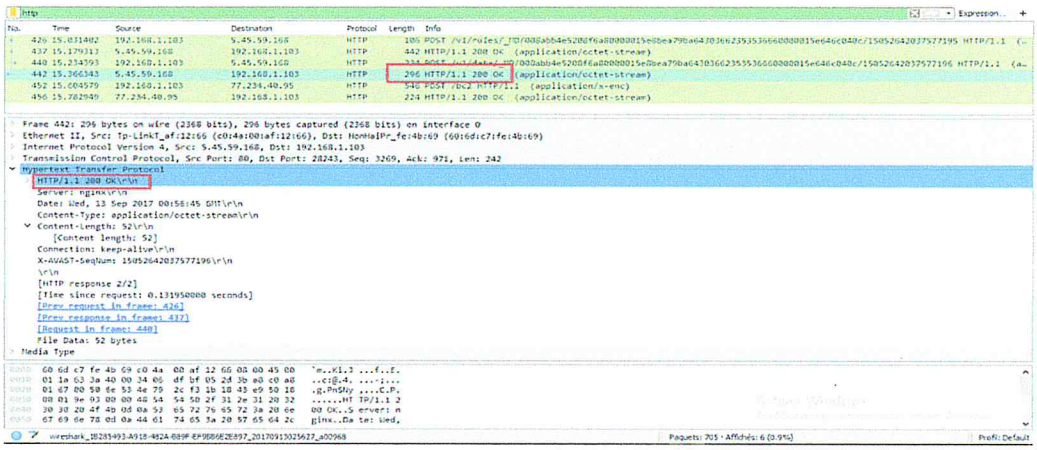


FIGURE 4.27 – Capture d'un paquet htttp

On note que la requête HTTP est sous la forme : HTTP / 1.1 200 ok \r \n

4.6.1.2 Visualisationde paquet d'attaque HTTP

Dans cette attaque, l'outil LOIC envoie des requêtes HTTP au serveur cible. Dans l'attaque http la chaîne elle est incluse dans le contenu d'un message HTTP GET.
On utilise la même procédure entamée dans les attaques précédente TCP et UDP, on sélectionne le type d'attaque HTTP et l'adresse de la cible vous pouvez également entrer l'adresse IP du système.
L'url de la cible est :HTTP ://www.univ-blida.dz .l'adresse IP de la machine cible est 193.194.83.181 et port d'attaque est 80 car on utilise la méthode d'attaque HTTP via internat La configuration des paramètrespour lancer l'attaque HTTP est présentée dans la figure 4.28

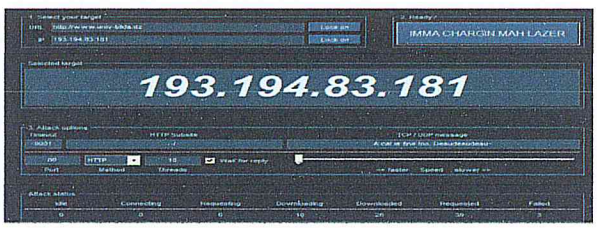


FIGURE 4.28 – L'attaque HTTP

On utilise Wireshark pour analyser notre réseau et la communication entre la cible et la victime :

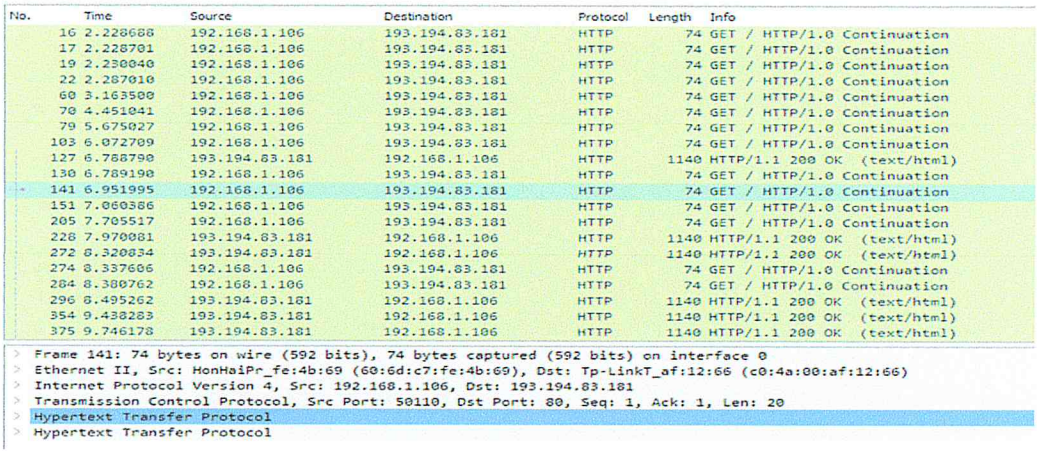


FIGURE 4.29 – La requête de l'attaque HTTP

On remarque que le serveur de la machine cible est inondé de requête HTTP On a pris une autre capture d'information au niveau d'encapsulation :

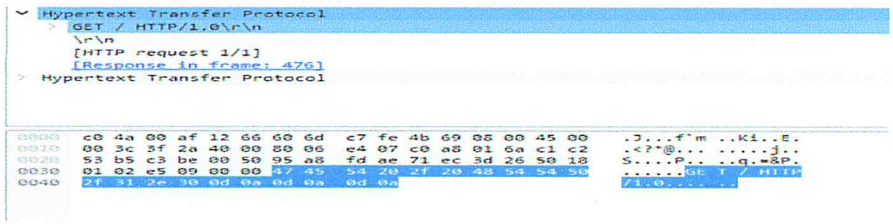


FIGURE 4.30 – La requête HTTPavec plus de détailles

L'outil Wireshark nous permet de noter que la requête HTTP est sous la forme : HTTP/1.0. ce qui veut dire que le trafic http et injecter et qui ne pas générée normalement .

D'après notre visualisation des paquets de protocole HTTP normal et celles des attaques, nous avons défini la signature de cette attaque :

	Cas d'un paquet HTTP normal	Cas d'un paquet d'attaque HTTP	La différence entre les deux cas
La forme de la requéte http	http/1.1 200 ok \r\n	http/1.0	Mal formé

TABLE 4.4 – La différence entre le cas normale de HTTP et de l'attaque HTTP

4.6.1.3 L'impact de l'attaque HTTP

Une inondation HTTP est une méthode d'attaque utilisée par les pirates pour attaquer les serveurs Web et les applications. Il consiste à envoyer des ensembles de requêtes HTTP GET ou POST apparemment légitimes, à un serveur Web cible. Ces demandes sont spécifiquement conçues pour : consommer une quantité importante de ressources du serveur et peuvent donc entraîner une condition de déni de service.

4.6.2 Détection de l'attaque HTTP

La règle que nous avons proposé qui définit l'attaque HTTP est la suivant :

Alert tcp any any ->any any (msg :"LOIC Dos attaque HTTP mod "flag :ap ; content |47 45 54 52 2f 20 48 54 54 50 2f 31 2e 30 0d 0a 0d 0a 0d 0a| ;sid :100000006 ; rev :1 ;)

|47 45 54 52 2f 20 48 54 54 50 2f 31 2e 30 0d 0a 0d 0a 0d 0a| : c'est le message HTTP (HTTP /1.0\r \n) en hexadécimale.
Alert : Une alerte « LOIC Dos attack HTTP mod » sera générée si les critères sont réunis.
anyany ->anyany : de n'importe quel adresse IP source et port à n'importe quel adresse, port destination.
Msg : affiche le msg (LOIC Dos attack HTTP mod) dans les alertes.
Flags : le paquet est caractérisé par PSH et ACK (ap).
Content : d'après les résultats on a choisi le content=(HTTP/1.0\r\n), c'est la requête http get mal formé.
Sid : 100000006 c'est l'identifiant de notre signature dans la base de signature.

La figure suivante montre que Snort a détecté l'attaque HTTP après l'intégration de la règle proposé :

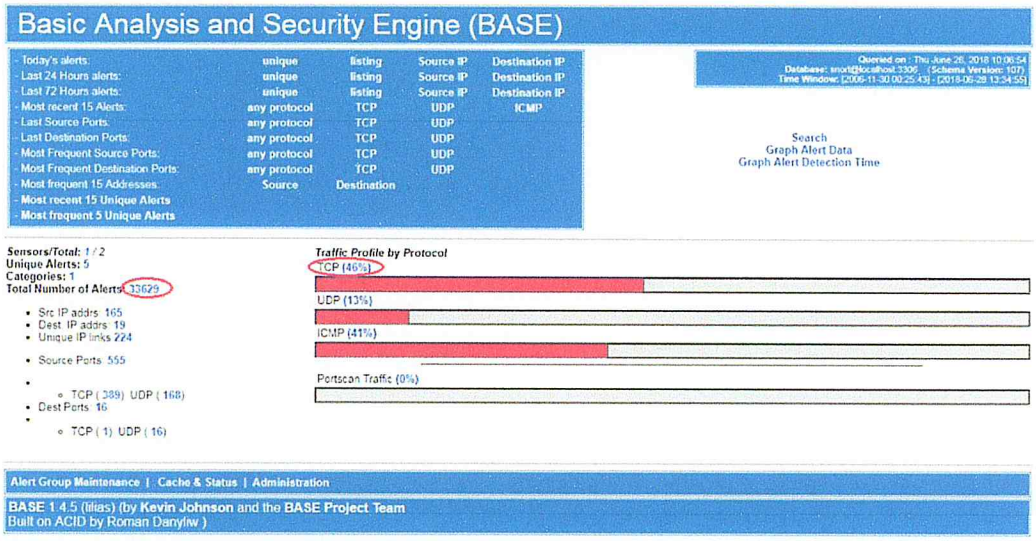


FIGURE 4.31 – La détection de l'attaque HTTP

La figure suivante illustre les données détaillées sur l'attaque HTTP :

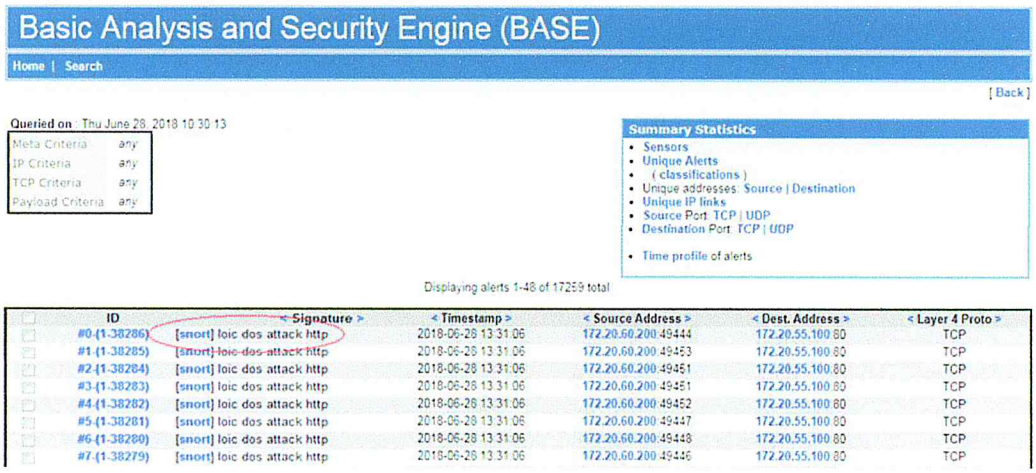


FIGURE 4.32 – Les informations détaillé de l'alerte d'attaque HTTP

La signature nous signale qu'il y'a une attaque de type TCP , l'outil utilisé par cette attaque est LOIC. Pour attaque http On remarque l'augmentation de flux TCP et les alertes car les attaque HTTP utilisent le protocole TCP

4.7 Conclusion

En utilisant nos différentes règles de détection qu'on a créée, nous avons pu détecter des quatre attaques de type DoS (TCP, UDP, HTTP, Ping of death) mentionnée dans Snort grâce à son système BASE, et à l'aide des différentes signatures tirées durant notre simulation d'attaques.

Pour la premier étape on a utilisé l'outil LOIC pour simuler l'attaque puis nous avons utilisé Wireshark pour visualiser les paquets et retrouver les signatures qui spécifie chaque l'attaque, ce qui nous a permettra de créer nos propre règle. En utilisant snort et les règle que nous avons créé en a pu détecter ces attaques (TCP, UDP, HTTP, Ping of death)

Conclusion générale

Les attaques de déni de service (attaque TCP, UDP, HTTP, Ping of death) ne sont pas quelque chose dont une entreprise veut être victime, cela pourrait coûter cher. Toutes les attaques ci-dessus peuvent être mises en œuvre par une personne qui n'a pas beaucoup de compétence et peut causer beaucoup de dégâts.

Le système de détection d'intrusion (IDS) est l'un des éléments mise en place dans le cadre de la politique globale de sécurité définie. En effet il est très utile et très efficace. L'objectif de notre projet consisté a créé des règles de détection des attaquesà base de Snort :

— Simulation des attaques.
— Reconnaissances des signatures à l'aide d'analyseur de paquet Wireshak.
— Création des règles de détection au niveau de Snort.

afin de valider les règles proposées et vérifier leur efficacité , nous avons effectué une série de tests tel que les attaques de type déni de service (Ping of death, TCP, UDP, HTTP). Nous avons réussi à rendre ces attaques visibles et détectableà l'aide de notre analyseur de paquets Wireshark, Ce dernier nous a aidé à tirer les différents signatures d'attaques, qu'on a développé pour créer nos règles de détection des attaques à base de Snort.

Notre projet rentre dans un projet dont l'un de ses buts et de pouvoir détecter d'autres types d'attaques et mettre en place une politique de sécurité efficace.

Annexe

Les options des règles SNORT

Les options de règle forment le cœur du moteur de détection d'intrusions de Snort, combinant les fonctionnalités d'utilisation, puissance et flexibilité. Toutes les options de règle de Snort sont séparées les unes des autres par un caractère point virgule " ;". Les mots clés des options de règle sont séparés de leurs arguments avec un caractère deux points " :". Au moment de la rédaction, il y a 15 mots clé d'option de règle disponibles dans Snort :

msg : affiche un message dans les alertes et journalise les paquets

logto : journalise le paquet dans un fichier nommé par l'utilisateur au lieu de la sortie standard

ttl : teste la valeur du champ TTL de l'entête IP

tos : teste la valeur du champ TOS de l'entête IP

id : teste le champ ID de fragment de l'entête IP pour une valeur spécifiée

ipoption : regarde les champs des options IP pour des codes spécifiques

fragbits : teste les bits de fragmentation de l'entête IP

dsize : teste la taille de la charge du paquet contre une valeur

flags : teste les drapeaux TCP pour certaines valeurs

seq : teste le champ TCP de numéro de séquence pour une valeur spécifique

ack : teste le champ TCP d'acquittement pour une valeur spécifiée

itype : teste le champ type ICMP contre une valeur spécifiée

icode : teste le champ code ICMP contre une valeur spécifiée

icmp_id : teste la champ ICMP ECHO ID contre une valeur spécifiée

icmp_seq : teste le numéro de séquence ECHO ICMP contre une valeur spécifique

content : recherche un motif dans la charge d'un paquet

content-list : recherche un ensemble de motifs dans la charge d'un paquet

offset : modifie l'option content, fixe le décalage du début de la tentative de correspondance de motif
depth : modifie l'option content, fixe la profondeur maximale de recherche pour la tentative de correspondance de motif
nocase : correspond à la procédure de chaîne de contenu sans sensibilité aux différences majuscules/minuscules
session : affiche l'information de la couche applicative pour la session donnée
rpc : regarde les services RPC pour des appels à des applications/procédures spécifiques
resp : réponse active (ferme les connexions, etc)
react : réponse active (bloque les sites web)

Messages (Msg)

L'option de règle msg dit au moteur de journalisation et d'alerte le message à imprimer avec une sauvegarde du paquet ou une alerte. C'est une simple chaîne texte qui utilise "ëcomme caractère d'échappement pour indiquer un caractère discret qui pourrait autrement rendre confus l'analyseur de règles de Snort (tel que le caractère point-virgule ";").
Format : **msg : "<message texte>";**

Logto

L'option logto dit à Snort de journaliser tous les paquets qui déclenchent cette règle vers un fichier journal de sortie spécial. C'est spécialement pratique pour combiner les données de choses comme les activités NMAP, les scans CGI HTTP, etc. Il doit être noté que cette option ne marche pas quand Snort est en mode de journalisation binaire.
Format : **logto : "<nom de fichier>";**

TTL

Cette règle d'option est utilisée pour fixer la valeur time-to-live à tester. Le test qu'il effectue est réussi seulement sur une correspondance exacte. Ce mot de clé d'option était destiné à être utilisé pour détecter les tentatives de traceroute.
Format : **ttl : "<nombre>";**

TOS

Le mot clé "tos" vous permet de vérifier le champ TOS de l'entête IP pour une valeur spécifique. Le test effectué est réussi seulement sur une correspondance exacte.
Format : **tos : "<nombre>" ;**

ID

Ce mot clé d'option est utilisé pour tester une correspondance exacte dans le champ ID de fragment de l'entête IP. Quelques programmes de pirates (et d'autres programmes) fixent ce champ spécifiquement pour différents besoins, par exemple la valeur 31337 est très populaire avec certains pirates. Ceci peut être retourné contre eux en mettant en place une simple règle pour tester ceci et quelques autres "nombres de pirates".
Format : **id : "<nombre>" ;**

Options IP (Ipoption)

Si des options IP sont présentes dans un paquet, cette option va chercher l'utilisation d'une option spécifique, tel que le routage par la source. Les arguments valides de cette option sont :
rr - Record route (ndt : enregistrement de la route)
eol - End of list (ndt : fin de liste)
nop - No op (ndt : pas d'opération)
ts - Time Stamp (ndt : horaire)
sec - IP security option (ndt : option de sécurité IP)
lsrr - Loose source routing (ndt : routage vague par la source)
ssrr - Strict source routing (ndt : routage stricte par la source)
satid - Stream identifier

Les options IP les plus fréquemment regardées sont les routages vague et stricte par la source qui ne sont utilisés dans aucune application Internet. Seulement une seule option peut être spécifiée par règle.
Format : **ipopts : <option> ;**

Fragment bits (Fragbits)

Cette règle inspecte les bits de fragment et le bit réservé dans l'entête IP. Il y a trois bits qui peuvent être vérifiés, le bit Reserved Bit (RB), le bit More Fragments (MF) et le bit Dont Fragment (DF). Ces bits peuvent être vérifiés pour une variété de combinaisons. Les valeurs suivantes sont utilisées pour indiquer les bits spécifiques :
R -Reserved Bit
D -DF bit
M -MF bit

On peut également utiliser des modificateurs pour indiquer des critères logiques de correspondance pour les bits spécifiés :
+ Tous les drapeaux, correspond si au moins les bits spécifiés sont positionnés
* Au moins un drapeau, correspond si au moins un des bits spécifiés est positionné
! Aucun drapeau, correspond si aucun des bits spécifiés n'est positionné

Format : **fragbits : <valeurs des bits> ;**

Exemple : **alert tcp !$HOME_NET any -> $HOME_NET any (fragbits : R+ ; msg : "Reserved IP bit set !" ;)**

La charge du paquet (Dsize)

L'option Dsize est utilisée pour tester la taille de la charge du paquet. Il peut être fixé à toute valeur, utilise en plus les signes supérieur/inférieur pour indiquer des intervalles et des limites. Par exemple, si on sait qu'un certain service a un tampon d'une certaine taille, on peut fixer cette option pour regarder les tentatives de débordement de tampons. Cela a l'avantage supplémentaire d'être une façon bien plus rapide de tester contre les débordements de tampons qu'une vérification de contenu de la charge.

Format : **dsize : [>|<] <nombre> ; (note : Les opérateurs > et < sont optionnels !)**

Contenu (Content)

Le mot clé content est une des fonctionnalités les plus importantes de Snort. Il autorise l'utilisateur de fixer des règles qui recherchent un contenu spécifique dans la charge du paquet et déclencher une réponse basée sur les données. A chaque fois que l'option content de correspondance de motif est exécutée, la fonction de correspondance de motif de Boyer-Moore est appelée et le test (plutôt cher en temps de calcul) est effectué contre le contenu du paquet. Si une donnée correspondant exactement à la chaîne de donnée en argument est contenue n'importe où dans la charge de paquet, le test est réussi et le reste des options de la règle est exécutée. Il est à noter que ce test fait la différence entre majuscules et minuscules.
La donnée d'option pour le mot clé content est quelque peu complexe ; il peut contenir du texte et des données binaires mélangées. Les données binaires sont généralement entourées dans des caractères barre verticale ("|") et représentées en tant que bytecode. Un bytecode représente une donnée binaire comme des nombres hexadécimaux et c'est une bonne méthode de raccourci pour décrire des données binaires complexes. La Figure 7 contient un exemple de mélange de textes et de données binaires dans une règle Snort.

Exemple :
alert tcp any any -> 192.168.1.0/24 143 (content : "|90C8 C0FF FFFF| /bin/sh" ; msg : "IMAP buffer overflow !" ;)

Format : **content : "<chaîne de contenu>" ;**

Offset

L'option de règle offset est utilisée comme un modificateur des règles utilisant le mot clé d'option content. Ce mot clé modifie la position de début de recherche pour la fonction de correspondance de motif depuis le début de la charge du paquet. Il est très utile pour des choses comme des règles de détection de scans CGI où la chaîne de recherche de contenu n'est jamais trouvée dans les quatre premiers octets de la charge. Soin doit être pris à ne pas fixer la valeur offset trop strictement sous peine de manquer potentiellement une attaque ! Ce mot clé d'option de règle ne peut pas être utilisé sans spécifier également l'option de règle content.

Format : **offset : <nombre> ;**

Profondeur (Depth)

Depth est une autre modification de l'option de règle contenu. Ceci fixe la profondeur maximale de recherche dans la fonction contenu de correspondance de motif de recherche depuis le début de sa région de recherche. Il est utile pour limiter la fonction de correspondance de motif d'exécuter des recherches inefficaces une fois que la région de recherche probable pour un ensemble de contenus donné a été dépassée. (C'est à dire que si on recherche "cgi-bin/php" dans un paquet relatif au web, on n'a probablement pas besoin de perdre du temps à rechercher dans la charge après les 20 premiers octets).

Format : **depth : <nombre> ;**

Exemple :

alert tcp any any -> 192.168.1.0/24 80 (content : "cgi-bin/phf" ; offset : 3 ; depth : 22 ; msg : "CGI-PHF access" ;)

Aucun cas (Nocase)

L'option nocase est utilisée pour désactiver la différence majuscules/minuscules dans une règle "content". Elle est spécifiée seule dans une règle et tout caractère ASCII qui est comparé à la charge du paquet est traité comme si il était soit en majuscule soit en minuscule.

Format : **nocase ;**

Exemple :

alert tcp any any -> 192.168.1.0/24 21 (content : "USER root" ; nocase ; msg : "FTP root user access attempt" ;)

Drapeaux (Flags)

Cette règle teste les drapeaux TCP pour une correspondance. Il y a actuellement 8 drapeaux variables qui sont disponibles dans Snort :
F -FIN (le bit le moins significatif dans l'octet des drapeaux TCP) S – SYN, R – RST,P – PSH, A – ACK, U -URG 2 -bit réservé 2 1 -bit réservé 1

Il y a aussi des opérateurs logiques qui peuvent être utilisés pour spécifier des critères de correspondance pour les drapeaux indiqués : + -Tous les drapeaux, correspond si au moins les bits spécifiés sont positionnés * -Au moins un drapeau, correspond si au moins un des bits spécifiés est positionné ! -Aucun drapeau, correspond si aucun des bits spécifiés n'est positionné Les bits réservés peuvent être utilisés pour détecter des comportements non usuels, tels que des tentatives d'empreintes digitales de piles IP ou d'autres activités suspicieuses.

Format : **flags : <valeurs de drapeaux> ; Exemple : alert any any -> 192.168.1.0/24 any (flags : SF ; msg : "Possible SYN FIN scan" ;)**

Séquence TCP (Seq)

Cette option de règle se réfère aux numéros de séquence TCP. Essentiellement, il détecte si le paquet a un numéro de séquence statique fixé, et donc plutôt peu utilisé. Elle a été incluse pour le bien de l'exhaustivité.

Format : **seq : <nombre> ;**

Acquittement (Ack)

Le mot clé ack d'option de règle se réfère au champ d'acquittement de l'entête TCP. Cette règle a un propos pratique jusqu'ici : détecter les pings TCP NMAP. Un ping TCP NMAP fixe ce champ à zéro et envoie un paquet avec le drapeau ACK TCP pour déterminer si un système réseau est actif.

Format : **ack : <nombre> ;**

Exemple :
alert any any -> 192.168.1.0/24 any (flags : A ; ack : 0 ; msg : "NMAP TCP ping" ;)

Type ICMP (Itype)

Cette règle teste la valeur du champ type ICMP. Il est fixé en utilisant la valeur numérique du champ. Pour une liste des valeurs disponibles, regarder dans le fichier decode.h

inclus avec Snort ou dans toute référence ICMP. Il doit être noté que les valeurs peuvent être fixées en dehors de l'intervalle pour détecter des valeurs de type ICMP invalides qui sont quelques fois utilisées dans des dénis de services et des attaques en inondations.

Format :**itype : <nombre> ;**

Code ICMP (Icode)

Le mot clé d'option de règle icode est plutôt identique à la règle itype, il faut juste fixer une valeur numérique ici et Snort va détecter tout trafic en utilisant cette valeur de code ICMP. Les valeurs hors intervalle peuvent également être fixées pour détecter le trafic suspicieux.

Format : **icode : <nombre> ;**

Session

Le mot clé session est nouveau depuis la version 1.3.1.1 et est utilisé pour extraire les données utilisateurs depuis les sessions TCP. Il est extrêmement utile pour voir ce que les utilisateurs tapent dans telnet, rlogin, ftp et même les sessions web. Il y a deux mots clé disponibles en argument pour l'option de règle session, printable ou all. Le mot clé printable affiche seulement les données que l'utilisateur verrait normalement ou serait capable de taper. Le mot clé all substitue les caractères non imprimables avec leurs équivalents hexadécimaux. Cette fonction peut ralentir considérablement Snort, donc elle ne devrait pas être utilisée dans des situations de charge importante, et est probablement mieux adaptée au post-traitement de fichiers journaux binaires (format tcpdump).

Format : **session : [printable|all] ;**

Exemple :
log tcp any any <> 192.168.1.0/24 23 (session : printable ;)

Icmp_id

L'option icmp_id examine le numéro ICMP ID d'un paquet ECHO ICMP pour une valeur spécifique. C'est utile car quelques programmes de canaux cachés utilisent des

champs ICMP statiques quand ils communiquent. Ce plugin particulier a été développé pour activer la règle de détection de stacheldraht écrite par Max Vision, mais c'est certainement utile pour détecter un nombre potentiel d'attaques.

Format : **icmp_id : <nombre> ;**

ICMP séquence (Icmp_seq)

L'option Icmp_seq examine le champ séquence ICMP d'un paquet ECHO ICMP pour une valeur spécifique. C'est utile car quelques programmes de canaux cachés utilisent des champs ICMP statiques quand ils communiquent. Ce plugin particulier a été développé pour activer la règle de détection de stacheldraht écrite par Max Vision, mais c'est certainement utile pour détecter un nombre potentiel d'attaques.

Format : **icmp_seq : <nombre> ;**

Rpc

Cette option regarde les requêtes RPC et décode automatiquement l'application, la procédure et la version de programme, en indiquant un succès quand les trois variables correspondent toutes. Le format de l'option d'appel est "application,procédure,version". Les caractères génériques sont valides pour la procédure et les numéros de version et sont indiquées par une "*".

Format : icmp_seq : <nombre,[nombre|*],[nombre|*]> ;

Exemple :

alert tcp any any -> 192.168.1.0/24 111 (rpc : 100000,*,3 ; msg :"RPC getport (TCP)" ;)
alert udp any any -> 192.168.1.0/24 111 (rpc : 100000,*,3 ; msg :"RPC getport (UDP)" ;)
alert udp any any -> 192.168.1.0/24 111 (rpc : 100083,*,* ; msg :"RPC ttdb" ;)
alert udp any any -> 192.168.1.0/24 111 (rpc : 100232,10,* ; msg :"RPC sadmin" ;)

Réponses (Resp)

Le mot clé resp met en oeuvre les réponses flexibles (FlexResp) au trafic qui correspond à une règle Snort. Le code FlexResp permet à Snort de fermer activement les connexions en infraction. Les arguments suivants sont valides pour ce module :
rst_snd -envoie des paquets TCP-RST à la socket expéditrice
rst_rcv -envoie des paquets TCP-RST à la socket destinataire
rst_all -envoie des paquets TCP_RST dans les deux directions
icmp_net -envoie un paquet ICMP_NET_UNREACH à l'expéditeur
icmp_host -envoie un paquet ICMP_HOST_UNREACH à l'expéditeur
icmp_port -envoie un paquet ICMP_PORT_UNREACH à l'expéditeur
icmp_all -envoie tous les paquets ci-dessus à l'expéditeur
Les options peuvent être combinées pour envoyer des réponses multiples au système cible. Les arguments multiples sont séparés par des virgules.

Format : **resp : <resp_modifier[, resp_modifier...]> ;**

Exemple :
alert tcp any any -> 192.168.1.0/24 1524 (flags : S ; resp : rst_all ; msg : "Root shell backdoor attempt" ;) alert udp any any -> 192.168.1.0/24 31 (resp : icmp_port,icmp_host ; msg : "Hacker's Paradise access attempt" ;)

Liste de contenu (Content-list))

Le mot clé content-list permet à de multiples chaînes de contenu d'être spécifiées à la place d'une unique option de contenu. Les motifs qui seront recherchés doivent pour chacun être spécifiés sur une seule ligne du fichier content-file, mais sinon ils sont traités identiquement aux listes de contenu spécifiées comme un argument à la directive standard content. Cette option est la base pour le mot clé react.

Format :**content-list : "<nom de fichier>" ;**

Réaction (React)

Le mot clé react basé sur les réponses flexibles (FlexResp) met en oeuvre la réaction flexible au trafic qui correspond à une règle Snort. La réaction de base est de bloquer les sites intéressants que les utilisateurs veulent accéder : “ les sites à contenus indésirables “ . Le code Flex Resp permet à Snort de fermer activement les connexions en infraction et/ou d'envoyer un message visible au navigateur (modificateur d'avertissement disponible bientôt). Le message peut inclure votre propre commentaire. Les arguments suivants (modificateurs de base) sont valides pour cette option :
block : ferme la connexion et envoie message visible
warn : envoie le message d'avertissement visible (sera disponible bientôt), l'argument de base peut être combiné avec les arguments suivants (modificateurs supplémentaires) :
msg : inclut le texte de l'option msg dans le message visible de bloquage
proxy] : <port_nr> -utilise le port du relais pour envoyer le message visible
Des arguments additionnels multiples sont séparés par une virgule. Le mot clé react doit être placé en dernier dans la liste des options.

Format : react : <react_basic_modifier[, react_additional_modifier...]> ;

Exemple :

tcp any any <> 192.168.1.0/24 80 (content-list : "adults" ; msg : "Not for children !" ; react : block, msg ;)

alert tcp any any <> 192.168.1.0/24 any (content-list : "adults" ; msg : "Adults list access attempt" ; react : block ;)

References

[1]ACISSI , " sécurité informatique-Ethical hacking,Apprendre l'attaque pour mieux se défendre",Edition Eni,2011.

[2] Jean-Francois pillou , " Tout sur la sécurité informatique" , Edition DUNOD, 2005.

[3] Roulot Et Mejan, " le piratage de A à Z ", Edition Edigo, 2010.

[4] Renaud Dumont , "Cryptographie et Sécurité informatique" , Université de Liège Faculté des Sciences Appliquées (2009-2010)

[5]Zhenqi Wang , Dankai Zhang" HIDS and NIDS Hybrid Intrusion Detection System Model Design" , Information & Network Management Center, North China Electric Power University,Baoding 2012.

[6] M. Fathi BEN NASR , " Mise en place d'une sonde snort " Mastère Spécialisé en Sécurité Informatique (2004-2005).

[7] John Blackwell , "Ramit-Rule-Based Alert Management Information Tool " THE FLORIDA STATE UNIVERSITY COLLEGE OF ARTS AND SCIENCES. Master of Science .

[8] U Aickelin, J Twycross and T Hesketh-Roberts ,"Rule Generalisation using Snort " . School of Computer Science and IT University of Nottingham . International Journal of Electronic Security and Digital Forensics 1(1) · March 2008

[9] les virus informatique clusif 2005,Disponible sur (https ://clusif.fr/publications/les-virus-informatiques/).

[10] Hervé Debar, Benjamin Morin, Frédéric Cuppens, Fabien Autrel, Ludovic Mé, Bernard Vivinis Salem Benferhat, Mireille Ducassé, Rodolphe Ortalo," Détection d'intrusions :corrélation d'alertes". Article de synthèse, Caen, France, 2004.

[11] Cédric Michel, "Langage de description d'attaques pour la détection d'intrusions par corrélation d'événements ou d'alertes en environnement réseau hétérogène", thèse de doctorat de l'Université de Rennes1,16 Décembre 2003.

[12] Les virus et les spam,Disponible sur (https ://www.sophos.com/fr-fr/ medialibrary /PDFs/case/20studies/fr/comviru_vrius_bfr.pdf?la=fr-FR)

[13] Jonathan-Christofer Demay, " Génération et évaluation de mécanisme de détection d'intrusions au niveau applicatif", Thèse de doctorat, école doctorale Matisse, université deRennes1 Juillet 2011.

[14] Nicolas Baudoin , Marion Karle ," NT Reseau, IDS et IPS" , 2004.
Disponible sur (http ://igm.univ-mlv.fr/ duris/NTREZO/20032004/Baudoin-Karle-IDS-IPS.pdf)

[15] Hadaoui Rebiha , " un IDS basé sur un algorithme inspirer du fonctionnement de colonies de Fourmies" , Mémoire de magistère, université M'Hamed Bougara de Boumerdes.

[16] Philippe Biondi," Architecture expérimentale pour la détection d'intrusions dans un système informatique ", Article de recherche, Avril-Sptembre 2001.

[17] Thierry Evangilista, "Les IDS Les systèmes de détection d'intrusions informatiques édition DUNOD", Paris 2004.

[18] Jean-Marc ROYER," sécuriser l'informatique de l'entreprise, Enjeux, menaces, prévention et parades" , Edition ENI 2004 .

[19] Laurent Bloch-Christophe Wolfhugel," Sécurité informatique ".EYROLLES, 2eme edition. 2005.

[20] https ://en.wikipedia.org/wiki/SYN_flood

[21] https ://www.thesecuritybuddy.com/dos-ddos-prevention/what-are-ping-flood-and-

ping-of-death/

[22] https ://www.frameip.com/smurf/

[23] http ://aglabs.co/distributed-denial-of-service-attack-diagram.html

[24] https ://www.thesecuritybuddy.com/data-breaches-prevention/what-is-arp-spoofing/

[25] https ://security.stackexchange.com/questions/13556/public-dmz-network-architecture

[26] https ://fr.slideshare.net/marymaro/idssnort-et-scurit-reseau

[27] https ://ravistechblog.wordpress.com/tag/network-intrusion-detection-system/

[28]https ://www.researchgate.net/figure/Taxonomy-of-DDOS-Attack-tools_fig1_304292694