Table des matières

Liste des figures

Introduction générale

Introduction générale

La sécurité est un domaine vaste est très difficile à encapsuler, il est courant de penser à un dispositif de sécurité à travers des gardes caméras de surveillances et systèmes d'alarmes. Toutefois, l'évolution spectaculaire enregistrée dans les domaines des technologies de l'information et de la communication à fais imposer la notion de sécurité numérique. En effet, l'évolution technologique avec : l'internet, le payement en ligne et le commerce électronique, est sur le point de modifier notre façon de vivre et d'interagir. Ceci à suscité un vif intérêt pour la sécurité pour des réseaux locaux ou à grande envergure. Il est évident que notre ère avec les avancées technologiques enregistrées est celle qui est caractérisée par le nombre important de failles de sécurités. Il est courant de voir des individus accéder à des ressources payantes gratuitement (Livre, TV, Musique...), beaucoup plus il est devenus possible d'opérer des arnaques ou vols à distance et sans prendre le moindre risque. Des individus mal intentionnés, peuvent accéder à un compte privé, pour divulguer des informations confidentielles, ruiner la réputation et détruire l'information. Pour certains pays de telles actions vont avoir un impacte directe sur l'économie.

Généralement un pirate informatique est considéré comme un malfaiteur. Cependant, dans certaines circonstances le piratage informatique n'est pas considéré comme une mauvaise action, nous avons sniffer et scanner dans le réseau de l'université. Rappelons que beaucoup d'améliorations et révolutions en télécommunications sont issues d'un acte de piratage ou intrusion : le cryptage dans le réseau GSM a été imposé à cause des actes de phreaking effectués sur son prédécesseur.

Notre projet s'inscrit dans le cadre de la sécurité des systèmes numériques interconnectés. Pour nous la sécurité est la mise en service d'un dispositif permettant de réduire le nombre des individus capable d'effectuer des actes de piratage, beaucoup plus nous voulons introduire à travers un dispositif de détection d'intrusion une confiance d'utilisation et d'exploitation. Toutefois, nous nous sommes limité au réseau Intranet qui désigne un réseau ou une infrastructure de communication basée sur les standards de l'internet tel que les protocoles TCP/IP, destinés à l'échange et au partage d'information au sein d'une entreprise ou une entité

organisationnelle. Rappelons que ces informations peuvent intéresser de nombreuses personnes et faire l'objet de piratages informatiques. Cela dit, outre la mise en place de par-feu et de système d'authentification, une sécurité avancée demeure très importante, d'où la nécessité de développé et mettre en place d'un système de détection d'intrusion : c'est l'objectif de notre projet, organisé comme suit :

Le premier chapitre, décrit des généralités sur les intrusions, les différentes étapes par lesquelles un pirate informatique procède et énumère quelques attaques qui ont marqué l'histoire.

Le second chapitre, détaille le fonctionnement du protocole TCP, l'établissement de connexion avec ce protocole, les failles sur ce dernier exploitées lors des attaques, nous montrerons les dispositifs de détection IDS utilisés dans ces cas.

Le troisième chapitre, présente les protocoles de la couche Internet de la pile TCP/IP les failles existantes dans ce niveau ainsi queles méthodes de détection implémentées.

Le quatrième chapitre met en pratique le dispositif open source de détection d'intrusion dit SNORT. Nous montrerons comment le configurer et comment l'associer avec les fichiers des critères de détection. A la fin, nous présentons une application simple sous Windows permettant d'automatiser les différents modes d'exécution de l'outil SNORT.

Chapitre I : Généralités sur les intrusions

I.1- Introduction

Un système informatique attire de plus en plus de pirates qui essayent de s'y intégrer de façon illégale afin d'y accéder et de modifier des données, cela en passant par plusieurs étapes et en utilisant différentes attaques, permettant d'obtenir les mots de passe et les informations confidentielles. L'administrateur dans plusieurs cas ne sera pas en mesure de contrer les différentes attaques et intrusion vu que le système n'est jamais assez protégé.

Dans ce chapitre, nous allons présenter les généralités sur les procédés par lesquelles un pirate informatique procède.

I.2- Définition d'une intrusion réseau

Une intrusion vise à s'infiltrer dans un système informatique à travers un réseau de télécommunication, afin de récupérer des informations confidentielles ou bien signer une défaillance voir causer des dégâts irréparables.

I.3- Les phases d'une intrusion

Pour parvenir à s'intégrer dans un réseau, un attaquant doit passer par les étapes ci-dessous :

I.3-a- La collecte d'information

Englobe tout ce qui est connaissance de la cible et ses failles. Appelée aussi ***«Finger printing»,*** permet à l'attaquant de collecter des informations concernant sa cible. En effet, un moteur de recherche à l'image de Google peut être exploité pour cette tâche vu qu'il est en possession d'une immense base de données contenant des informations sur des personnes, système et défaillance.

Souvent des commandes de recherche classiques ou même du système d'exploitation sont utilisées pour cette collecte, de plus des outils logiciels sont actuellement disponible pour effectuer une collecte à distance. [1]

I.3-b- Repérage des failles

Une fois que l'attaquant a collecté suffisamment d'informations intéressantes sur le système tel que l'architecture et le fonctionnement du réseau, il tentera de repérer les failles pour s'insérer dans le système qui peuvent bien se situer dans le système d'exploitation, l'utilisation d'un service ou un protocole. Et grâce à la collecte effectuée l'attaquant identifie

les services actifs dans la machine cible, ainsi que les versions des systèmes d'exploitations. L'utilisation des scanneurs de vulnérabilités tels que ***« NESSUS »*** ou ***« SAINT »*** permet de tester et trouver les portes d'entrées dans le système. [1]

Un site WEB mal protégé peut être une bonne porte d'entrée pour les systèmes informatiques.

I.3-c- Intrusion dans les systèmes

Une fois l'attaquant a trouvé sa porte d'entrée dans le système, il doit s'assurer de ne pas laisser de traces pour ne pas être tracé par l'administrateur. Le cas idéal est de s'infiltrer dans le système via un accès *administrateur,* sauf que si le pirate a pu accéder autant qu'utilisateur, il doit chercher le mot de passe *administrateur*. Quand l'attaquant accède au système autant qu'un super utilisateur, il a la possibilité de modifier les fichiers ou les détruire. [1]

I.3-d- L'écoute du trafic

Un autre type d'intrusion consiste à écouter le flux dans le réseau.

Pour écouter le trafic circulant sur un réseau, l'attaquant doit disposer d'un outil appelé ***« sniffer », Wireshark*** en est un bon exemple.

La majorité des protocoles font transiter les informations en clair sur Internet, ce qui permet à l'attaquant de les capturer et les exploiter immédiatement. Par exemple, un utilisateur consulte ses e-mails sans utiliser le chiffrement SSL, alors son identifiant et son mot de passe vont transiter sur le réseau et pourront être interceptés par le sniffer. [1]

I.3-e- Exploitation

Maintenant que le pirate s'est intégré dans le système, tout dépend de son objectif principal visé, il peut y avoir plusieurs fins et plusieurs attaques qui seront définies par la suite.

I.4- Les types d'attaques

Les attaques se divisent en quatre grands axes :

I.4-a Les attaques d'accès

C'est une attaque tente d'accéder à l'information et vise sa confidentialité. Généralement effectuée par une personne dont l'accès n'est pas autorisé :

- ***Le craquage de mot de passe***

Pour accéder aux fichiers dans une machine cible, l'attaquant doit disposer d'un mot de passe. Pour l'obtenir trois méthodes peuvent être mise en œuvre :

i. Attaque par Brute Force

Le principe est simple, il suffit de tester toutes combinaisons possibles les unes après les autres jusqu'à trouver la bonne, pour chaque combinaison il faut donc calculer l'empreinte (le mot de passe haché) et comparer avec l'empreinte recherchée. Afin que les logiciels de sécurité ne détectent pas l'attaque, les combinaisons testées ne sont pas dans l'ordre (aa,ab,ac,...).

L'avantage de cette attaque est qu'elle est efficace à 100% mais le temps de calcul pour trouver le bon mot de passe peut être très long. [2]

ii. Attaque par dictionnaire

Elle consiste en un premier temps à créer une base de données contenants des mots de passes avec leurs empreintes pré-calculées et de les comparer avec celle recherchée. Cette méthode peut être efficace si le mot de passe recherché se trouve dans le dictionnaire, certaines personnes utilisent des prénoms ou des dates pour se protéger et dans ce cas elle est même plus rapide que l'attaque par *Brute Force*. L'inconvénient majeur de cette attaque est qu'elle nécessite une capacité de stockage importante. [2]

iii. Attaque par rainbow table

L'implantation de cette technique s'avère très difficile, bien qu'elle soit très efficace.

- ***Les chevaux de troie***

Un cheval de troie prend la forme d'un logiciel ou d'une mise à jour une fois exécutée il installe ses fonctions cachées pouvant s'exécuter. Il aura la capacité de modifier ou copier des données confidentielles et aussi permettront la prise de contrôle à distance de la machine cible. Ensuite l'utilisateur distant aura la possibilité de lire et d'écrire des données, transférer des fichiers, prendre le contrôle de la souris et du clavier.

Le moyen de protection contre les chevaux de troie, est l'utilisation des antivirus et leur mise à jour.

- ***Le sniffing***

C'est le fait d'écouter du trafic dans une ligne qui transite des paquets pouvant être importants pour des hackeurs. Les données ne sont pas toujours chiffrés pour certains protocoles, vont être facilement récupérés et exploités tels que les mots de passe et les conversations.

- ***Ingénierie sociale***

Contrairement aux autres attaques, celle-ci n'est pas informatique, elle exploite l'abus de confiance faite par les utilisateurs du système et récupère leurs informations sensibles en s'appuyant sur leur naïveté. Elle ne nécessite pas de logiciels, vu que ça se déroule soit par téléphone, par e-mail ou bien par contact directe en tentant de gagner leur confiance afin d'extraire les données souhaitées. [3]

I.4-b Les attaques de modification

- ***Les virus informatiques***

Un virus est un programme qui menace le système informatique. Il possède le même fonctionnement qu'un virus biologique, il infecte la machine ciblée et se propage de machine en une autre. Certains virus peuvent avoir aucun effet sur le système, par contre d'autre font perdre ou modifier des données. Il y a qui se contentent de ralentir le système. Ils se répandent à travers tous les moyens d'échanges de données comme les CDROM, clé USB , les réseaux informatiques.

Le meilleur moyen de s'en protéger est d'utiliser un firewall et un anti-virus et de le mettre à jour.

❖ ***Les vers informatiques***

Un ver informatique ou worm, est un sous-ensemble de virus qui a la particularité de s'autoproduire. Il utilise les failles des systèmes d'exploitation et les mécanismes réseau pour pouvoir se propager dans un réseau.

La solution la plus simple d'empêcher les vers de s'installer dans la machine est d'utiliser un firewall et un anti-virus et de le mettre à jour.

I.4-c Les attaques par répudiation

Dans ce type d'attaque, l'attaque tente de donner une fausse information ou de nier qu'un évènement s'est réellement passé.

❖ ***Le spoofing***

Le spoofing ou le poisonning technique qui permet à une machine de s'authentifier auprès d'une autre machine au moyen de paquets semblant être envoyés d'une source de confiance. Pour réaliser cela le pirate informatique doit accomplir les étapes suivantes :

- Identifier la cible.
- Immobiliser la machine dont l'adresse doit être usurpée.
- Contrefaire l'adresse de la machine usurpée.
- Se connecter à la cible en se faisant passer pour la machine usurpée.
- Trouver le numéro de séquence exacte demandé par la cible.

Plusieurs formes d'attaques exploitent l'usurpation d'identité tels que :

- ✓ l'IP spoofing qui est l'utilisation d'une adresse IP de confiance
- ✓ L'ARP spoofing qui consiste à modifier le cache ARP
- ✓ Le DNS spoofing une autre forme d'usurpation, dont le principe est de changer les tables de correspondance « nom de machine – adresse IP ». [4]

❖ ***Hijacking***

Cette attaque survient lorsqu'un hacker prend contrôle sur la machine et l'utilise. Le Hijacking peut être réalisé de différentes façons, par exemple : En exploitant les cookies capturés sur le réseau pour court-circuiter les mots de passes, ou en émulant des routeurs et serveurs pour tromper les hôtes d'un réseau.

I.4-d Les attaques par saturation ou le déni de service

C'est une attaque visant à rendre une machine ou un réseau indisponible, dont le principe général consiste à envoyer des données ou des paquets de taille inhabituelle, ceci a pour effet de provoquer des réactions inattendues de la cible pour aller jusqu'à l'interruption de service. [5] La machine visée peut être un serveur mail ou WEB, un routeur ou une machine simple dans un réseau. Il est à noter que le but de cette attaque n'est pas d'accéder à des informations mais juste d'empêcher la cible d'offrir ou d'utiliser des services. Elles sont aussi, très répandues dans les réseaux, car elles sont très faciles à mettre en œuvre et très difficile à arrêter.

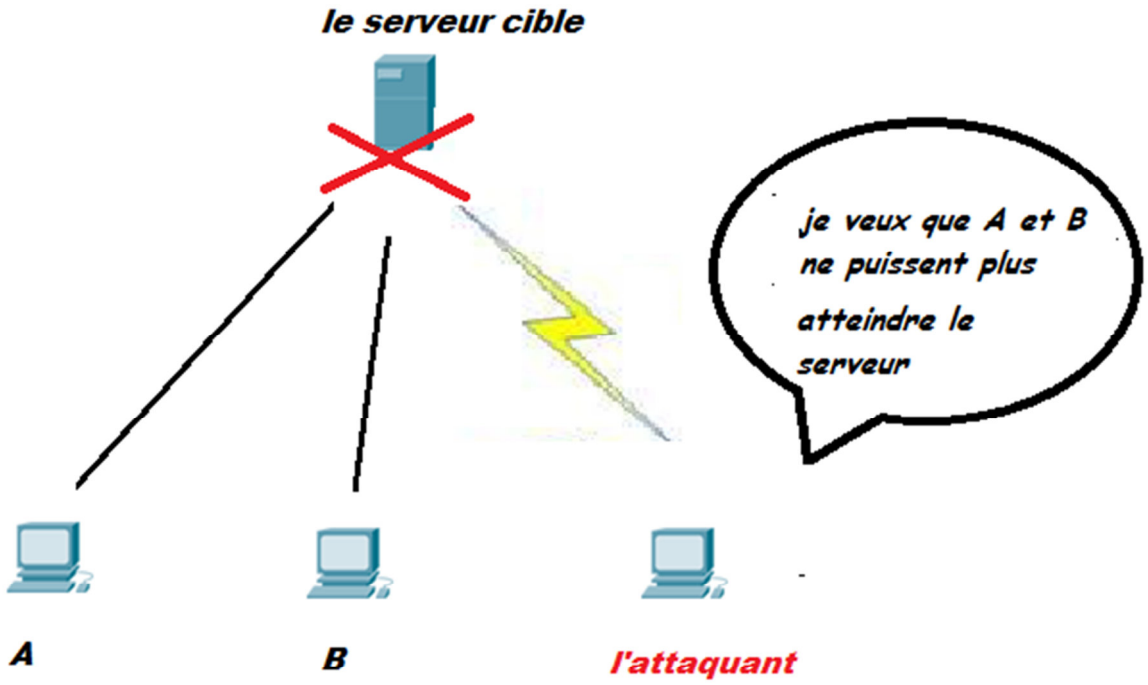

Figure I .4-a Principe de fonctionnement d'un déni de service

Ce genre d'attaque représente un réel fléau car elles paralysent temporairement le réseau tout entier ou du moins les machines les machines qui exécutent TCP/IP. [4] Pour avoir plus de chance de réussir l'attaque, le pirate utilise un système distribué, cela veut dire plusieurs machines attaquent la cible en même temps, on appelle cette technique : « le déni de service distribué ».

Certaines attaques DOS exploitent des failles dans les systèmes d'informations et d'autres qui surchargent la victime en lui envoyant un nombre important de requêtes de toutes sortes.

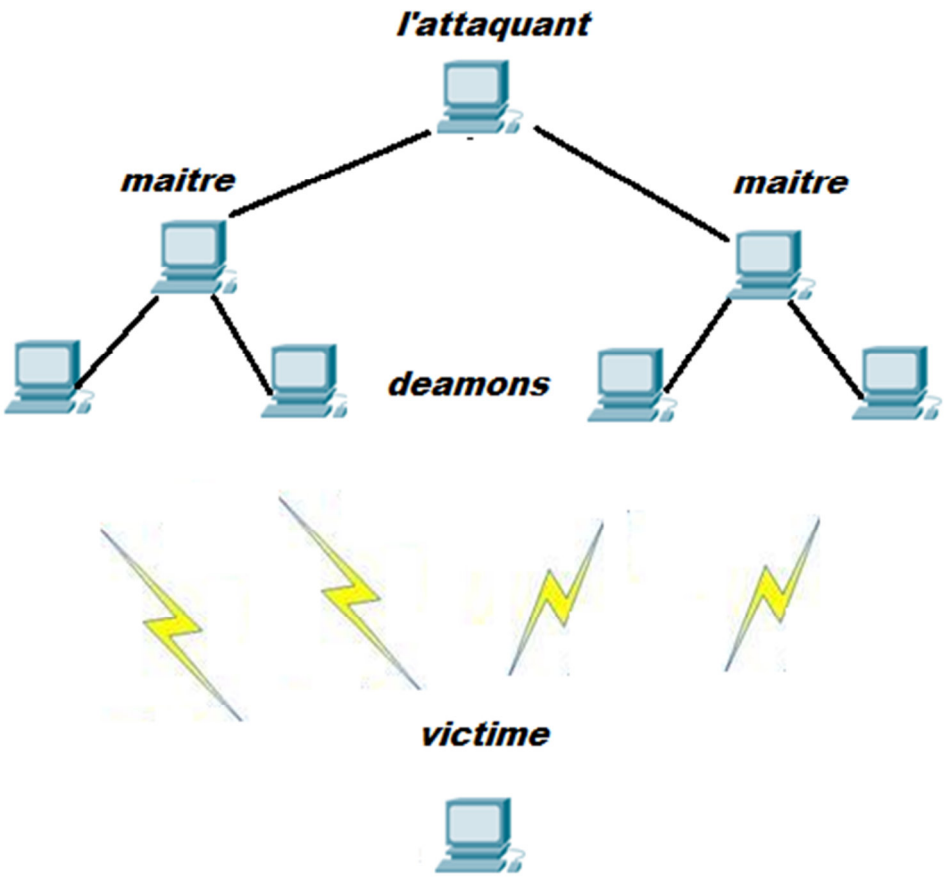

Figure I.4-b Principe de fonctionnement d'un déni de service distribué

I.5- Quelques attaques qui ont marqué l'histoire

I.5-a L'affaire MORRIS

Ce fut en novembre 1988 : la première attaque de type déni de service. On a estimé à 5000 le nombre de machines paralysées, pendant plusieurs heures. A cette époque, ce fut un désastre pour les centres universitaires et de recherche. [4]

I.5-b Attaque par sniffing

En 1994, une personne non identifiée a installé un sniffer de réseau sur de nombreux hôtes et éléments du réseau fédérateur, recueillant plus de 1000 nom d'utilisateur et de mot de passe valide via Internet et Milnet. [4]

I.5-c Attaque par spoofing

En janvier 1995, ce type d'attaque est devenu plus courant visant des sites Internet à l'échelle internationale. Un grand nombre de système ciblé par ces attaques sont des serveurs de noms, des routeurs et d'autres systèmes d'exploitation de réseau. Elles ont été menées à bien dans la plupart des cas. [4]

I.6- Backtrack

C'est une distribution de Linux, créée pour la première fois en 2007 dans le cadre du projet suisse « *Remot-Exploit* ». Conçu pour tous ceux qui veulent tester le niveau de sécurité et les vulnérabilités sur leurs ordinateurs et leurs réseaux. Composé d'une série qui dépasse les 300 outils pour tester les failles de sécurités, effectuer des intrusions et les corriger. Ces outils sont organisés dans 11 catégories :

1. Collecte d'information
2. Mapping Network
3. Identification des vulnérabilités
4. Analyse des applications WEB
5. Analyse de résau Radio (802.11, Bluetooth, RFID)
6. Pénétration (Exploit &Toolkitingénierie sociale)
7. Elévation de privilèges
8. Maintenir l'accès

9. Digital Forensics
10. Reverse Engineering
11. Voice Over IP

I.7- La pile TCP

Les attaques sont basées sur le fonctionnement des protocoles réseau. Un protocole est un ensemble de règles à suivre pour permettre l'échange d'information. Ces règles de communication permettent d'assurer le bon transfert de données.

Il n'existe pas un protocole unique, mais un ensemble de protocoles permettant de répondre aux différents besoins d'échanges d'informations. Dans le cas de communication via Internet, un ensemble de protocoles est nécessaire définit sous le nom de la pile TCP/IP pour « Transmission Control Protocol/Internet Protocol. Pour réaliser une intrusion, un pirate informatique doit bien étudier ces protocoles afin d'exploiter leurs faiblesses.

Le TCP/IP est le fruit des recherches qui ont été menées par le DARPA (Defense Advanced Reserch Project Agency) dès la fin des années 60. Le sigle TCP/IP signifie « Transmission Control Protocol/Internet », il provient de nom de deux protocoles majeurs TCP et IP.

TCP/IP représente d'une certaine façon l'ensemble des règles de communication sur Internet et se base sur la notion d'adressage IP, c'est-à-dire le fait de fournir une adresse IP à chaque hôte du réseau afin de pouvoir acheminer des paquets de données en vérité TCP/IP n'est pas limité à deux protocoles mais recouvre au fait toute famille de protocoles, on parle alors de « pile protocolaire ».

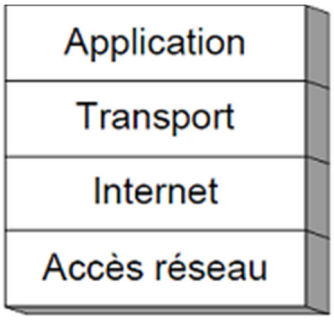

Figure I.7 La pile Protocolaire TCP/IP

Le modèle TCP/IP définit quatre couches effectuant chacune une tâche précise dans un ordre précis:

I.7-a La couche application

Cette couche constitue le sommet de l'architecture TCP/IP, elle contient les protocoles de haut niveau destinés à permettre le dialogue entre clients et serveurs comme par exemple :

- Telnet : un protocole permettant la connexion à distance.
- FTP : ce protocole permet le transfert de fichier.

I.7-b La couche transport

Cette couche est chargée des questions de qualité de service touchant la fiabilité, le contrôle de flux et le contrôle d'erreur. On trouve deux protocoles principaux : le TCP (Transmission Control Protocol) et le protocole UDP (User Datagramme Protocol).

I.7-c La couche Internet

Elle prend en charge le routage des données. Elle permet l'injection de paquets dans n'importe quel réseau et l'acheminement de ces paquets jusqu'à destination. Le protocole qui régit cette couche est appelé IP (Internet Protocol).

I.7-d La couche d'accès au réseau

On lui donne également le nom de la couche « hôte-réseau ». Cette couche se charge de tout ce qu'un paquet IP a besoin pour établir une liaison physique avec l'hôte de destination. On trouve généralement dans cette couche le protocole Ethernet.

I.8- Conclusion

La première partie a donné une présentation des intrusions dans les systèmes informatiques, ainsi qu'un classement des différentes attaques qui peuvent nuire à la qualité des services offerts et affecter la confidentialité des utilisateurs et leur vie privée. Il apparait qu'un hacker n'est qu'une personne avec une maîtrise totale des protocoles et ingénierie des réseaux, et qui utilise ses connaissances à des fins malintentionnés.

Chapitre II : Intrusion de la couche transport

II.1- Introduction

Se trouvant sur la couche transport, le protocole TCP possède des failles pouvant être exploités par des malfaiteurs, et par la suite, nuire à des connexions légitimes établies.

Dans ce chapitre nous allons essayer de présenter le protocole cité, son fonctionnement, les techniques d'attaques utilisées, les Systèmes de détections d'intrusion « ***IDS*** » et les HoneyPots.

II.2-Présentation du protocole TCP

- ***Définition***

TCP est un protocole qui fonctionne en mode connecté, il assure un transport fiable de données de bout en bout cela veut dire les paquets sont transmis sans perte ni duplication.

Il se trouve dans le modèle de structuration de couches de protocoles au-dessus d'IP, et se base sur le modèle Client/serveur.

L'adresses IP permet de router les informations vers telle ou telle machine, mais une fois que les paquets arrivent à destination et qu'ils se mettent à remonter la pile TCP/IP, il faudra qu'ils aillent vers l'application correspondantes, et pour cela des nombres appelés « ports » sont utilisés. Un port est simplement un nombre entre 1et 65536, par exemple : HTTP utilise le port 80.

- ***Présentation du segment TCP***

Un segment TCP contient les informations suivantes :

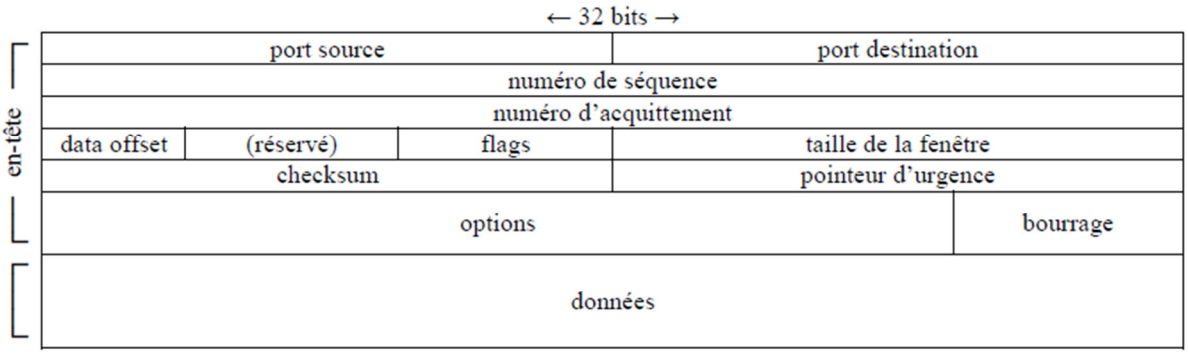

Figure II.2-a Format d'un segment TCP

- ***Port source (16bits) :*** port source de l'application sur la machine source.
- ***Port destination (16bits) :*** port destination de l'application sur la machine destination
- ***Numéro de séquence (32bits) :*** numéro du premier octet du paquet dans l'ensemble du flux de données transmis, 0 pour le 1er message d'un paquet.
- ***Numéro d'acquittement (32bits) :*** numéro de séquence attendu, soit le numéro du dernier octet reçu incrémenté de 1 cela dit, les paquets de numéros inferieurs ont été reçus.
- ***Data Offset (4bits) :*** position du champ donné dans le paquet en mots de 32 bits
- ***Réservé (6bits) :*** positionné à 0.
- ***Flags (6bits) :*** bits de contrôle

  - ***URG***, URGent (1bit) : utilisation du champ pointeur d'urgence.
  - ***ACK***, ACKnowledgment (1bit) : validation du champ numéro d'acquittement
  - ***PSH***, PuSH (1bit) : livraison instantanée des données à l'application sans mise en mémoire tampon.
  - ***RST***, ReSeT (1bit) : demande de réinitialisation de connexion.
  - ***SYN***, SYNchronisation (1bit) : synchronisation des numéros de séquence.
  - **FIN**, FINalize (1Bit): fin de la transmission.
- ***Taille de la fenêtre (16bits) :*** nombre d'octets à transmettre sans nécessiter d'accusé de réception.
- ***Checksum (16bits) :*** somme de contrôle de vérification en prenant compte du champ « données » et du champ en-tête virtuel.

- ***Pointeur d'urgence (16bits) :*** position d'une donnée urgente par rapport au numéro de séquence, spécifiant une livraison instantanée à l'application dès que l'octet pointé est lu. [6]
    - ***Connexion TCP :***

1- Etablissement d'une connexion TCP

Avant l'envoi des données à travers le réseau, une connexion doit être créée entres les deux entités en procédant comme ceci :

- Lorsque l'émetteur désire signaler au destinataire qu'il souhaite créer une connexion, il envoie un paquet TCP contenant le bit SYN pour la synchronisation et un numéro de séquence initial ainsi le numéro de port qui sera utilisé pour l'échange de données.
- Quand le destinataire reçoit le paquet contenant le flag SYN, il renvoie un accusé de réception SYN/ACK (acknowledgment) pour dire que la connexion est acceptée.
- L'émetteur reçoit l'accusé de réception et à son tour renvoie au destinataire un ACK (acknowledgment).

Cette phase est appelée « Three Way HandShake » et est illustrée dans la figure suivante : [7]

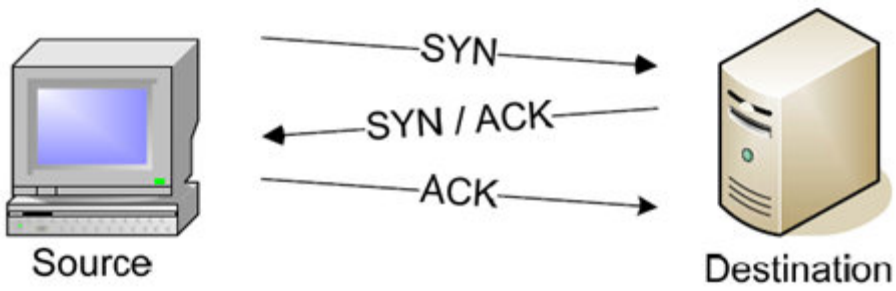

Figure II.2-b Etablissement d'une connexion TCP

Grâce à l'outil Wireshark qui permet de capturer les trames TCP qui circulent sur le réseau, on a pu vérifier ce mécanisme de communication (figure II-2-c).

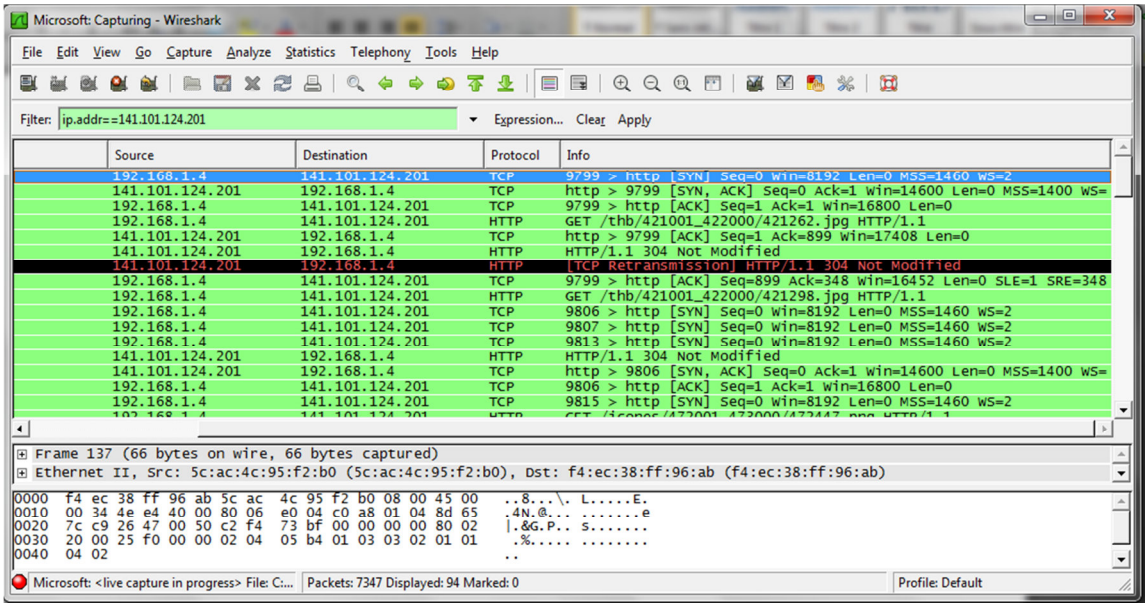

Figure II.2-c Capture d'un établissement d'une connexion TCP avec Wireshark

Nous voyons bien, dans la colonne 1 que les trois premières lignes sont des paquets TCP, ayant successivement les flags : SYN, SYN+ACK et ACK. Donc la connexion à bien été établie.

2- *L'échange de données*

Lorsque la connexion est établie, chaque station connait le numéro de séquence de l'autre. Elles peuvent s'échanger les données.

Pour acquitter les données, le serveur met à jour son numéro d'acquittement en ajoutant la longueur des données reçues et émet un nouveau datagramme en tenant compte de la nouvelle valeur.

L'outil Wireshark, nous permet d'obtenir la capture suivante, qui illustre un échange de données entre un serveur et un client.

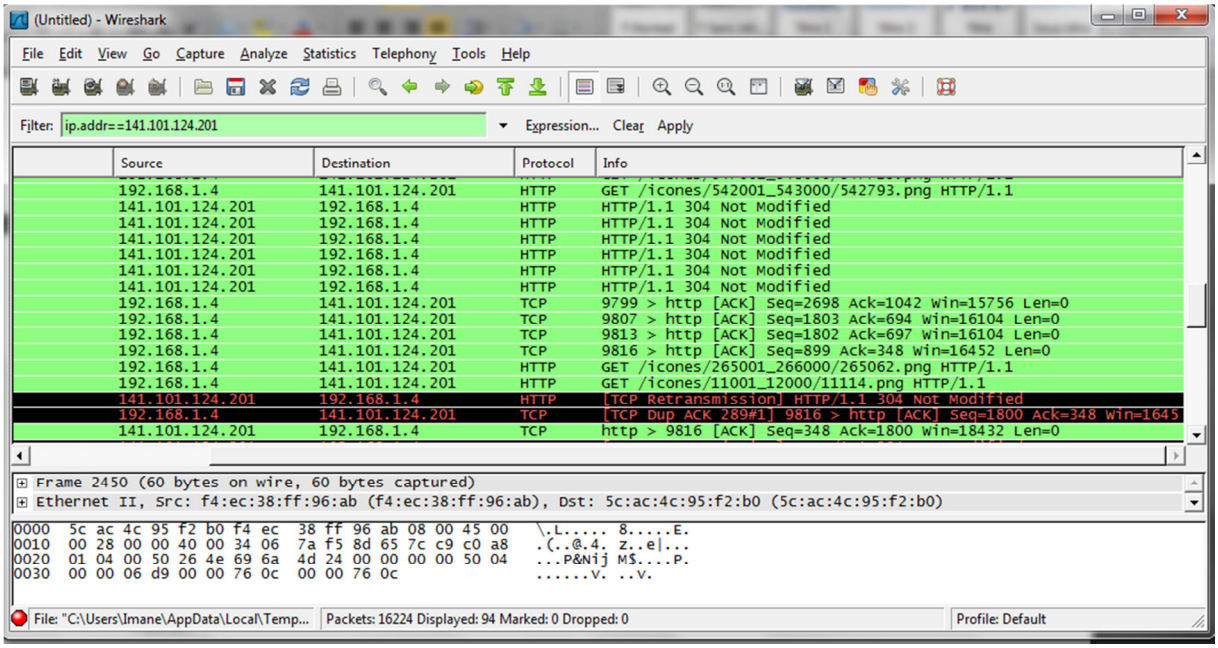

Figure II.2-dCapture d'un échange de données avec Wireshark

3- *La fermeture de connexion TCP*

La fermeture d'une connexion TCP est divisée en 4 étapes. Chacune des 2 parties (serveur et client) peut fermer la connexion.

Tout d'abord, l'hôte souhaitant fermer la connexion envoie un segment avec les drapeaux FIN et ACK armés. Le récepteur confirme la réception par un ACK et envoie à son tour un segment FIN/ACK. L'hôte souhaitant fermer la connexion confirme alors par un ultime ACK.

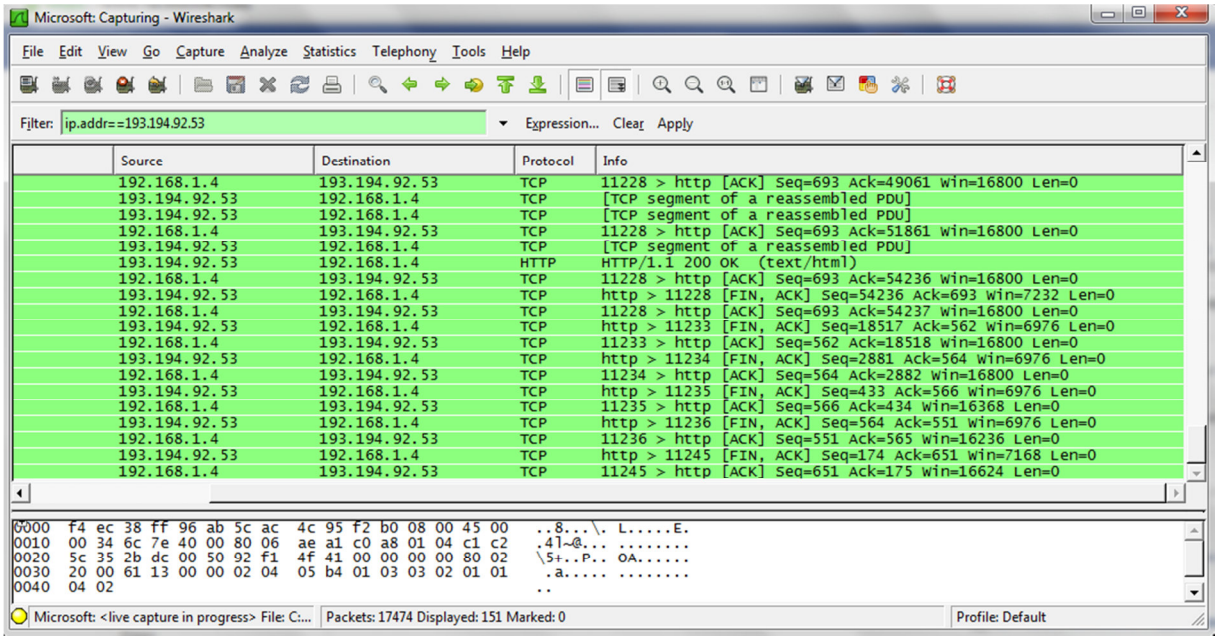

Figure II.2-e Capture d'une fermeture de connexion avec Wireshark

II.3- Les techniques utilisées pour les intrusions TCP

II.3-a Finger printing

Comme il a été cité dans le chapitre précédent, un pirate informatique doit trouver une porte d'entrée pour accéder à sa cible. Pour cela, il exploite le mécanisme de fonctionnement du protocole TCP pour effectuer un scan des ports qui lui permet d'identifier ceux qui sont ouverts, fermés et filtrés.

- *Identification des ports*
  - ***Les ports ouverts*** : la station source envoie un paquet TCP contenant un flag SYN ainsi que le numéro de port. Si la cible envoie un paquet TCP qui contient un flag ACK acquittant le flag SYN, alors le port est ouvert.
  - ***Les ports fermés*** : si la station source recevra un paquet TCP contenant le flag RST et non pas un acquittement au flag SYN, alors le port interrogé est fermé.
  - ***Les ports filtrés*** : après avoir envoyé le paquet TCP avec le flag SYN, la station source ne recevra aucun paquet, ce qui signifie que le port interrogé est filtré. [7]

L'outil NMAP exploite de telles techniques. Nous l'avons utilisé pour effectuer un scan sur une station ayant l'adresse « 172.16.13.180 », le résultat obtenu est comme suit :

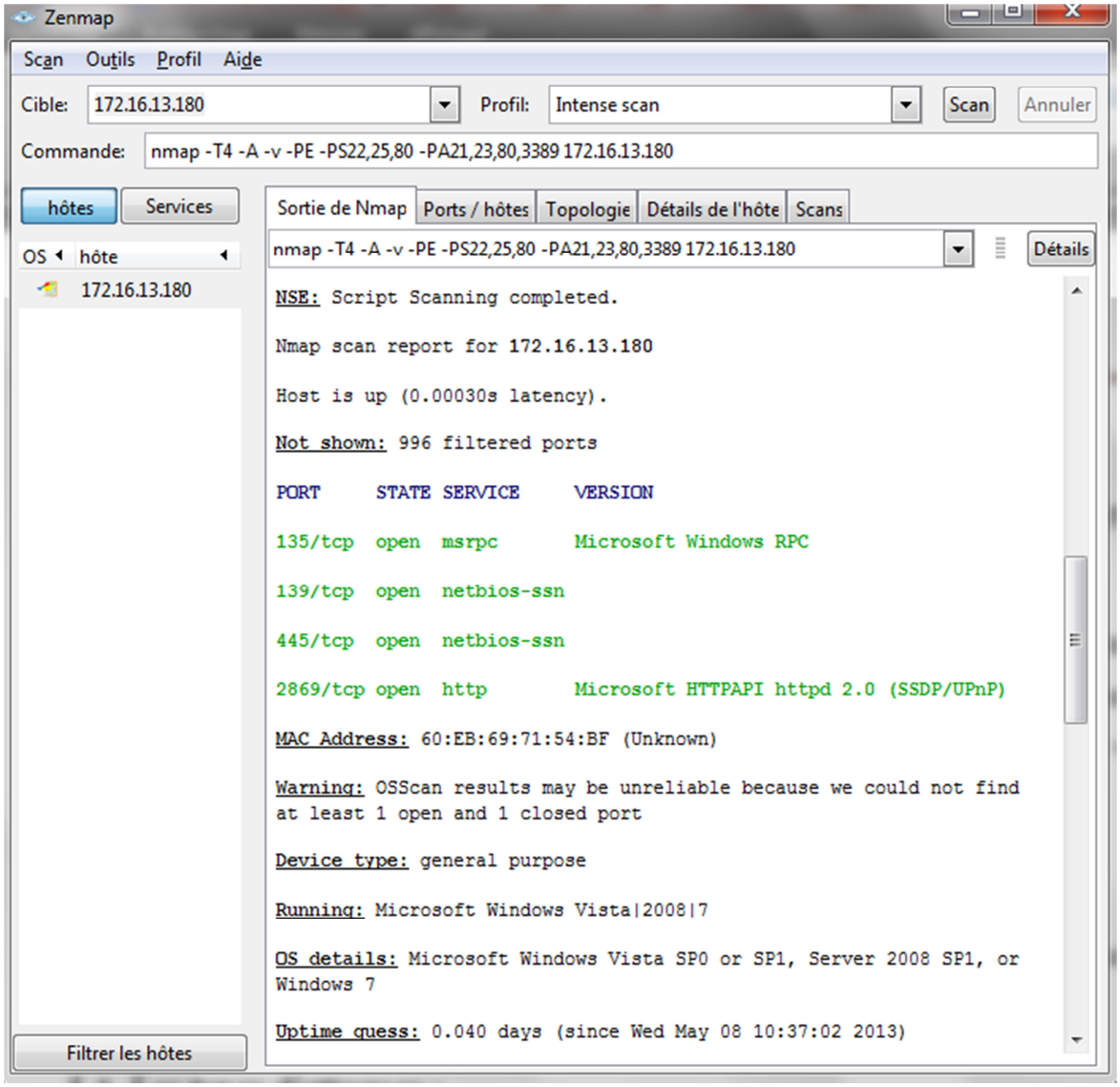

Figure II.3-a Scan d'une machine distante avec Nmap

II.3-b SYN Flooding

L'attaquant demande d'établir une connexion avec la cible en lui envoyant un nombre très important de paquets TCP contenant le flag SYN pour la synchronisation, cela obligera la victime à démarrer une socket pour chaque demande de connexion et donc, envoyer des paquets contenant le flag SYN-ACK sans recevoir de réponse. La cible aura un grand nombre de connexion en attente et arrivera à saturation jusqu'à ne plus pouvoir répondre aux connexions légitimes des autres utilisateurs. Pour éviter de se faire repérer l'attaquant change son adresse IP, ce qui redirigera les réponses de la cible vers une autre destination.

Comme il a été cité dans le chapitre précédent, Backtrack possède des outils très performants qui permettent notamment d'effectuer des dénis de service, tels que le SYN Flooding qu'on a déjà réalisé grâce à l'outil Metasploit, qui se trouve dans la catégorie « Exploitation Tools » en tapant les commandes suivantes dans la console :

1- use auxiliary/dos/tcp/synflood
2- Set RHOST @IP de la victime dans notre cas : 192.168.137.1
3- Set RPORT le port cible dans notre cas c'était le port 80
4- Set timeout le temps entre les paquets envoyés nous l'avons mis à 1 sec
5- Run : pour exécuter l'attaque

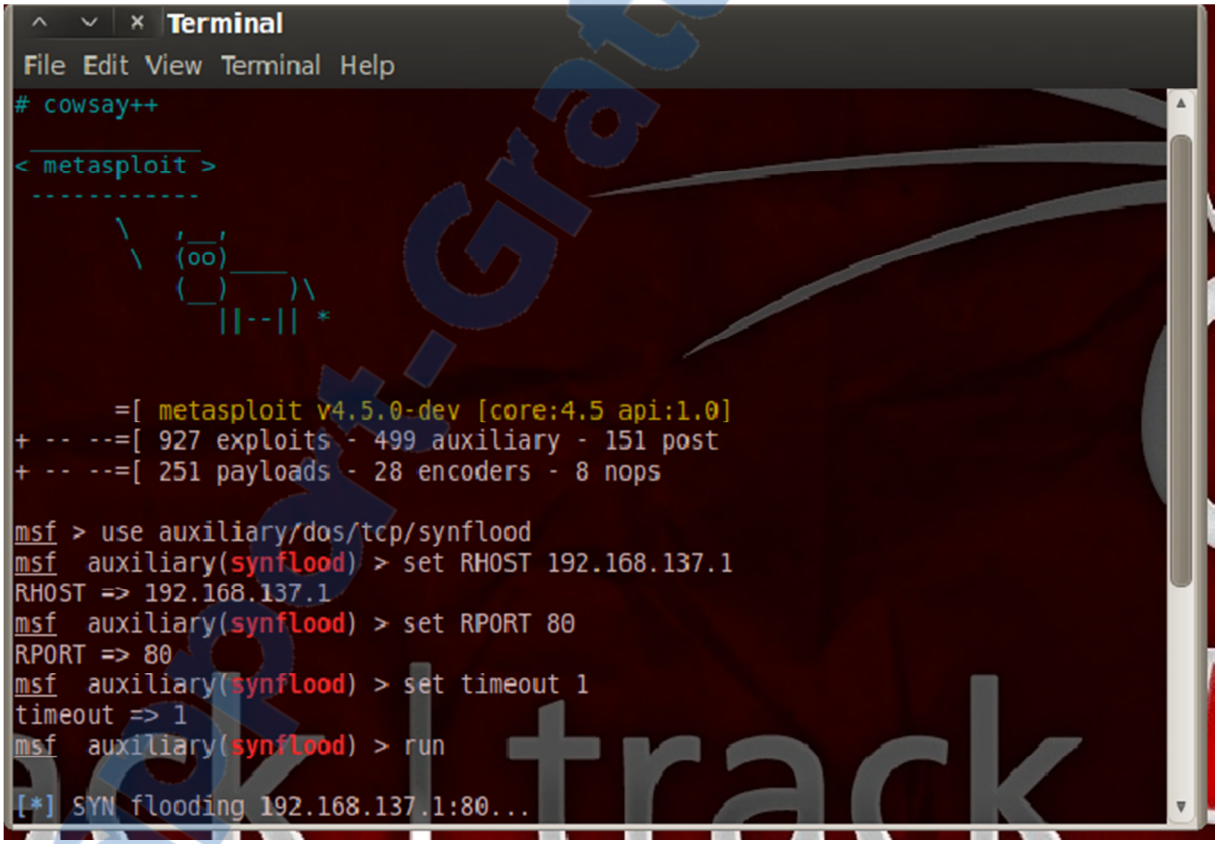

Figure II.3-b : Commandes pour effectuer un SYN Flooding

La figure ci-dessous représente les paquets TCP envoyés par la station voulant effectuer le SYN Flooding, contenant tous des flags SYN.

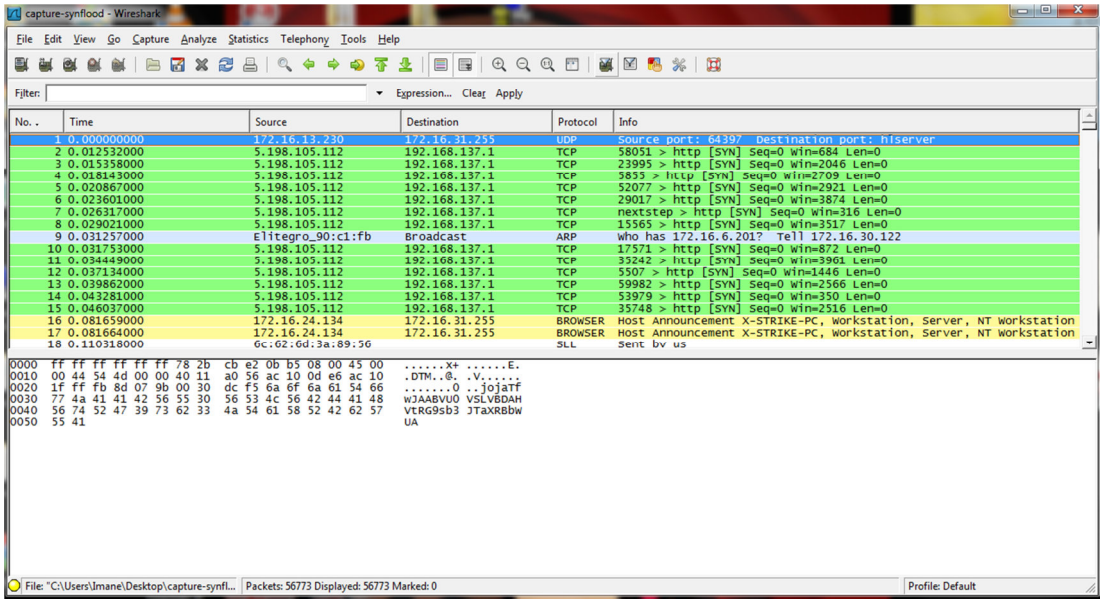

Figure II.3-c Capture d'un SYN Flooding

II.3-c Connexion Killing

Elle vise à stopper une connexion TCP, précédemment établie entre deux stations. Deux techniques peuvent être mises en œuvre :

- La première utilise le flag RST (Reset) et exige que les numéros de séquences soient corrects. Elle nécessite d'abord d'attendre un paquet en provenance de la station B à la station, puis calculer à partir des paquets reçus le numéro de séquence à indiquer le paquet contenant le flag RST, qui va être envoyé à la station C. Il ne restera plus qu'à envoyer le dit paquet et la connexion sera logiquement rompue.
- utilise le flag RST (Reset) et exige que les numéros de séquences soient corrects. Elle nécessite d'abord d'attendre un paquet en provenance de la station B à la station, puis calculer à partir des paquets reçus le numéro de séquence à indiquer le paquet contenant le flag RST, qui va être envoyé à la station C. Il ne restera plus qu'à envoyer le dit paquet et la connexion sera logiquement rompue. [8]

II.4- Le protocole UDP

Un autre protocole qui opère dans la couche Transport est l'UDP « User Datagramme Protocol ». Il gère le fractionnement et le réassemblage en paquets des segments de données. Il est rapide, simple, non fiable mais efficace et réalise une transmission des datagrammes en mode non connecté utilisant le protocole IP. Cela dit, le protocole n'assure pas d'ordonnancement ni de suivi de communication, ni de contrôle de flux. [6]

Comme pour le TCP, le protocole UDP permet d'identifier les processus à l'aide des numéros de port, et de vérifier l'intégrité des données avec le total de contrôle.

C'est un protocole qui est parfaitement adapté à la transmission d'informations vocales sur le réseau, car si une voix UDP est perdue, la conversation va se poursuivre sans qu'il y ait une grande perte d'information.

Contrairement au TCP, il n'utilise pas de « Three Way Hanshake ».

Ses inconvénients majeurs, c'est qu'il n'utilise aucun service d'authentification et aucun champ de l'entête n'est chiffré.

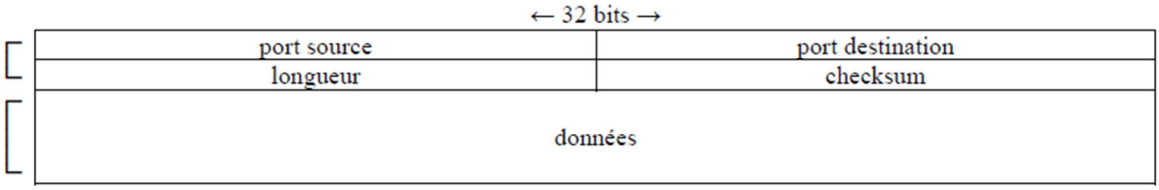

Figure II.4 Datagramme UDP

Les différentes informations contenues dans l'entête UDP sont les suivantes :

- ***Port source (16bits) :*** port de l'application sur la machine source.
- ***Port destination (16bits) :*** port de l'application sur la machine destination.
- ***Longueur (16bits) :*** longueur de l'entête des données.
- ***Total de contrôle (16bits) :*** somme de contrôle de vérification de l'entête. [6]

II.5- Les attaques exploitant le protocole UDP

- ***UDP Flooding***

Cette attaque exploite le mode non connecté du protocole UDP. Le principe est d'envoyer à la victime une grande quantité de paquet UDP ce qui mène à une congestion du réseau et saturation de l'hôte victime. Cette congestion est plus importante qu'avec le TCP-SYN flood car le protocole UDP ne possède pas de mécanisme de contrôle de congestion.[5]

II.6 Les Systèmes de détection d'intrusions IDS

II.6-a Définition

C'est un ensemble de logiciels ou matériels ou bien une combinaison des deux dont la fonction principale est de surveiller le trafic circulant dans un réseau et de détecter tout comportement intrus ou utilisation anormale par exemple : le balayage des portes ou le scan des machines, et éventuellement de réagir. Il existe en effet trois types d'IDS :

- ***NIDS*** (**N**etwork **I**ntrusion **D**etection **S**ystem): comme son nom l'indique un NIDS est système de détection d'intrusion qui surveille un réseau entier ou un sous réseau.
- ***HIDS*** (**H**ost-based Network Intrusion System): un Host IDS ou HIDS est un système de détection d'intrusion qui surveille et protège seulement les activités dans un hôte précis et non le réseau entier.
- ***DIDS*** (**D**istrubuted **I**ntrusion **D**etection **S**ystem): dans ce système les NIDS et les HIDS sont utilisés d'une façon distribuée dans le réseau et reliés à une station centrale

SNORT en est un bon outil open source qui peut être utilisé comme IDS.

II.6-b Réponses d'un IDS [10]

Il existe deux types de réponses des IDS, la réponse passive et la réponse active :

- Réponse passive : Consiste à stocker les intrusions détectées dans un fichier de « log » qui sera analysé par le responsable de sécurité. Ce genre de réponse empêchera l'attaque de se reproduire mais pas de l'arrêter au moment de sa réalisation.

- Réponse active : contrairement à la réponse passive, la réponse active a pour but de stopper l'intrusion au moment de sa détection, pour cela deux techniques sont utilisées, la reconfiguration du Firewall et l'interruption d'une connexion TCP.

II.6-c Avantages et inconvénients des IDS

i- Avantage

✓ Une surveillance continue et détaillée.

✓ Un IDS permet d'analyser le trafic réseau et relever des attaques alors qu'il n'en est même pas la cible directe.

✓ Toutes les alertes sont stockées dans un fichier ou dans une base de donnée ce qui permet de concevoir un historique et d'établir des liens entre les différentes attaques.

✓ Grâce à des outils de filtrage très intéressants un IDS permet de faire le contrôle par protocole (TCP, IP, ICMP..), par adresse IP.

✓ Une responsable de sécurité n'a pas besoin de surveiller le réseau en permanence, une attaque de nuit ne passera plus inaperçue. L'IDS renvoie de nombreuses informations avec une alerte contenant le type d'attaque, la source, la destination. [10]

ii- Inconvénients

✓ La mise en place d'un IDS fait appel à de bonne connaissance en protocole réseau, l'installation des logiciels est à la portée, en revanche l'exploitation nécessite des connaissances plus pointues. [10]

✓ Un IDS capte et stocke tout le trafic donc même si un SYN Flood est réalisé sur une autre machine tous les paquets seront capturés et stockés comme si l'attaque a été destinée aux systèmes IDS.

✓ La mise en place d'un IDS est très importante, il faut bien étudier la position et la connaissance réseaux est très importante.

✓ Ils sont réputés pour générer de fausses alertes.

II.7- Les HoneyPot

Un HoneyPot est un système utilisé pour attirer les pirates en exposant des vulnérabilités. Une fois que le hacker trouve le HoneyPot, il croira que c'est un vrai serveur donc il va tenter d'exploiter les failles, ce qui donnera l'opportunité de découvrir et d'enregistrer ses activités et techniques et de les utiliser afin d'endurcir la sécurité des systèmes réels.

II.8- SNORT

II.8-a Définition

C'est un système de détection d'intrusion réseau NIDS en open source, capable d'effectuer une analyse du trafic en temps réel. Utilisé généralement pour détecter des attaques et des scans tels que les débordements de tampons, des scans de ports furtifs, des attaques CGI, des fingerprintings. [9]

II.8-b Fonctionnement

SNORT peut fonctionner en quatre modes :

- ***Sniffer :*** il aura le même fonctionnement que Wireshark et tcpdump, c'est d'écouter le trafic dans un réseau donné, en utilisant les commandes suivantes :
  - ./Snort –v : permet d'afficher les entêtes des paquets.
  - ./Snort –vd : permet d'afficher les entêtes et les données.
  - ./Snort –vde ou ./Snort –d -e -v : affiche les entêtes, les données et les entêtes de la couche liaison de données.
- ***Générateur de logs :*** il permet de garder des traces sur les activités et d'éventuelle attaques sur un réseau.
  - ./Snort –dev –l ./log : pour lancer snort en mode « logger ».
- ***NIDS :*** il aura pour fonction de détecter les anomalies permettant de capter les intrusions (sécurité passive), en utilisant la commande suivante :
  - ./Snort –c c:\snort\etc\snort.conf
- ***IPS (Intrusion Prevention System) :*** permet la prévention des intrusions sur le réseau (sécurité active). SNORT In-Line.

II.8-c Architecture

Il possède une architecture modulaire schématisée par la figure II-8-c.

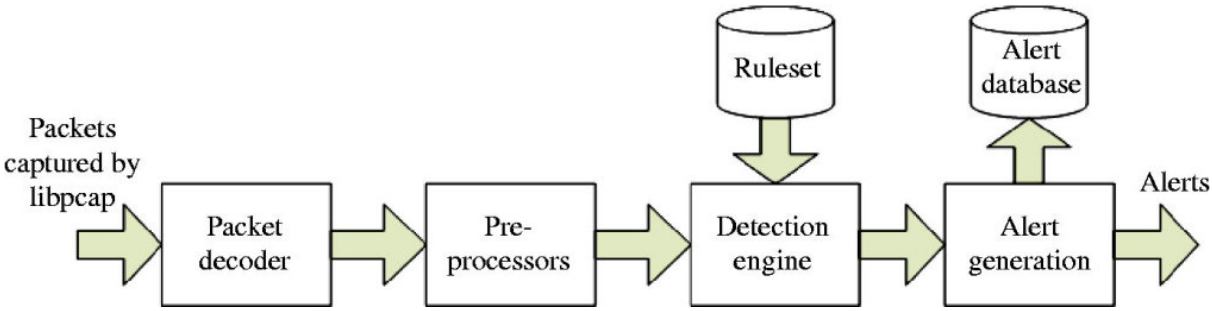

Figure II.8-a Architecture de SNORT

1- ***Noyau de base (paquet decoder) :*** Au démarrage ce noyau, charge un ensemble de règles, les compile, les optimise et les classe. Il se charge des captures de paquets durant l'exécution.

2- ***Une série de pré-processeurs :*** ils améliorent la possibilité de SNORT en matière d'analyse et de recomposition du trafic capturé. Ils reçoivent les paquets capturés, les retravaillent puis les fournissent au moteur de recherche de signature.

3- ***Une série d'analyse*** appliquée aux paquets. Ces analyses se composent principalement de comparaisons de différents champs des entêtes de protocoles (IP, ICMP, TCP et UDP) par rapport à des valeurs précises.

4- ***Output plugins :*** ils permettent de traiter ces intrusions de plusieurs manières après la détection. Ils peuvent envoyer des messages d'alertes vers un serveur syslog, stocker les intrusions dans une base de données SQL. [9]

II.8-d Position de SNORT sur le réseau

L'emplacement de la sonde SNORT peut affecter son efficacité sur le réseau. SNORT peut être placé dans trois places différentes dans une architecture composée d'un Firewall, une zone sensible (DMZ).

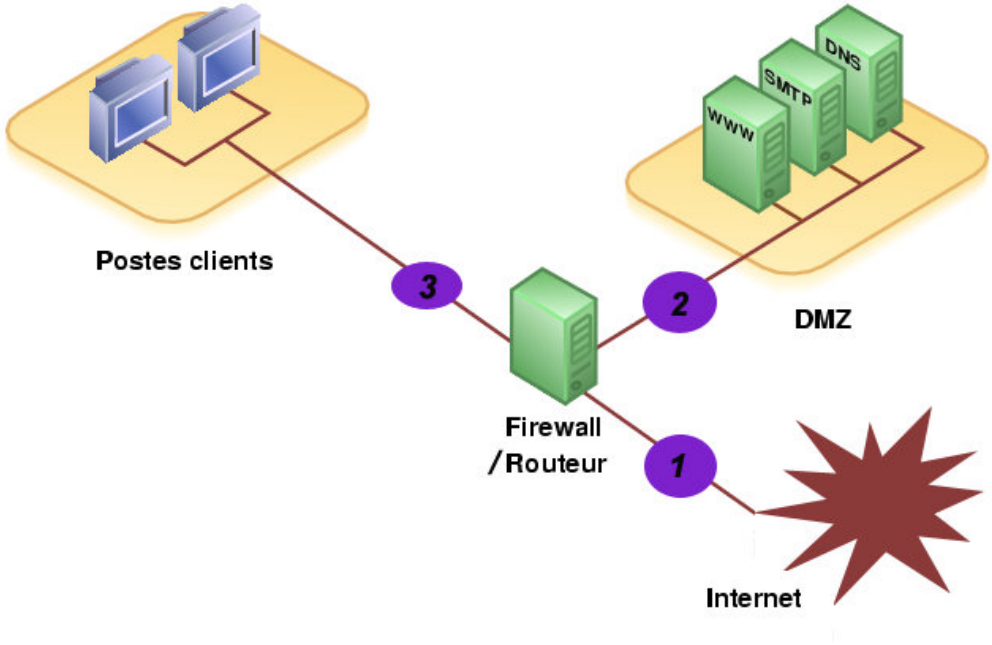

Figure II.8-b Différentes positions de SNORT dans le Réseau

1- Avant le Firewall (position1)

C'est une place de « premier choix » .Dans cette position, SNORT aura la possibilité d'analyser le trafic en provenance d'Internet, ainsi, il pourra détecter les attaques de sources extérieures avant le Firewall.

Bien que cette position ait des avantages à cet emplacement, il existe des inconvénients.

- La perte de fiabilité peut être causée par un trafic important traversant cette sonde.
- Exposition aux éventuelles attaques, vu que ces NIDS seront placés hors de la zone protégée par le Firewall.

2- Sur la DMZ (position 2)

Positionné après le Firewall, la sonde peut détecter le trafic filtré par le Firewall. Elle peut ainsi surveiller les attaques destinées aux serveurs de l'entreprise accessible de l'extérieur.

3- *Sur le réseau interne (position 3)*

Dans cette position-là, la sonde observera les tentatives d'intrusions parvenues de l'intérieur du réseau d'entreprise et les tentatives d'attaques à partir de l'intérieur. [11]

II.8-e Les règles SNORT

SNORT utilise des règles pour générer des messages d'alertes, enregistrer des paquets détectés ou encore, bloquer des connexions. Chaque règle se compose de deux parties distinctes : le header et les options.

- ***Le header :*** cette partie spécifie le type d'alerte à générer (alerte, log, pass), d'indiquer les champs de base nécessaire au filtrage, le protocole ainsi que les adresses IP & ports source et destination.
- ***Les options :*** spécifiées entre parenthèses permet d'affiner l'analyse, en décomposant la signature en différentes valeurs à observer parmi certains champs du header ou parmi les données. [9]

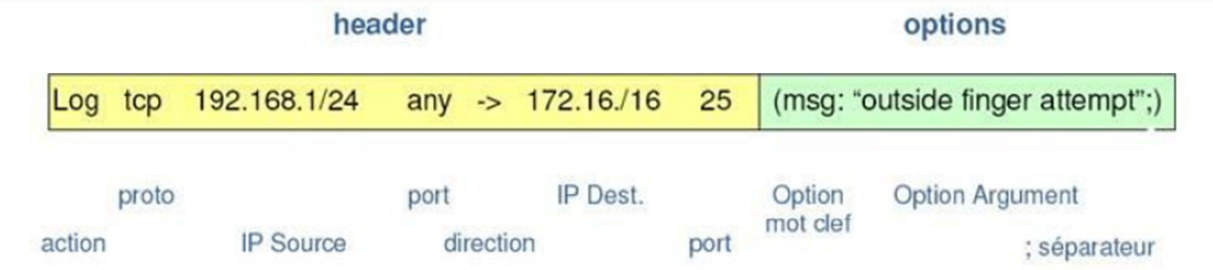

Figure II.8-c Composition d'une alerte SNORT

- ***Action de la règle :*** précise l'action à prendre tel que : log, alerte, pass,…
- **Protocole : tel que :** tcp, udp, icmp.
- ***Adresse IP source et destination :*** peuvent contenir les adresses IP source, destination ou « any ».
- ***Opérateur de direction :*** unidirectionnel -> , ou bidirectionnel <->
- ***Syntaxe des options :***
  - Combinaison de règles avec le séparateur « ; »
  - Séparation des mots clefs et des arguments avec « : »

- Mots clefs : msg, logto, minfrag, ttl, id, dsize, content, offset, depth, flags, seq, ack, itype, idecode, nocase, session. [9]

- *Exemple de règle TCP :*

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN synscan portscan"; flow:stateless; flags:SF; id:39426; reference:arachnids,441; classtype:attempted-recon; sid:630; rev:7;)
```

C'est une règle de type TCP, qui déclenche une alerte une fois qu'un scan avec le flag « SYN » est lancé.

- *Exemple de règle UDP :*

```
alert udp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN Webtrends Scanner UDP Probe"; content:"|0A|help|0A|quite|0A|"; reference:arachnids,308; classtype:attempted-recon; sid:637; rev:3;)
```

C'est une règle UDP, qui lance une alerte lorsqu'un « scan SSH Version map attempt » est effectué.

II.9- Détection

Grâce à l'outil SNORT, nous avons pu détecter quelques attaques en provenance d'autre machines. Tout d'abord, une première configuration est nécessaire :

- modifier le fichier de configuration « snort.conf » en utilisant un éditeur de texte, puis remplacer « any » de « var Home_Net any » par l'adresse IP du réseau surveillé et le chemin des règles. Dans notre cas-là ; l'adresse est 172.16.20.55 et le chemin des règles : c:\snort\etc\rules.
  Dans la section « output plugins », on ajoute la commande «output alert_fast :alert.ids »
- La commande « ./snort –c c:\snort\etc\snort.conf », permet d'utiliser le fichier de configuration « snort.conf » et stocker les alertes dans le fichier «alert.ids » qui a été créé précédemment.

Après avoir effectué les étapes précédentes, on a obtenu les alertes suivantes dans le fichier texte :

Alerte TCP :

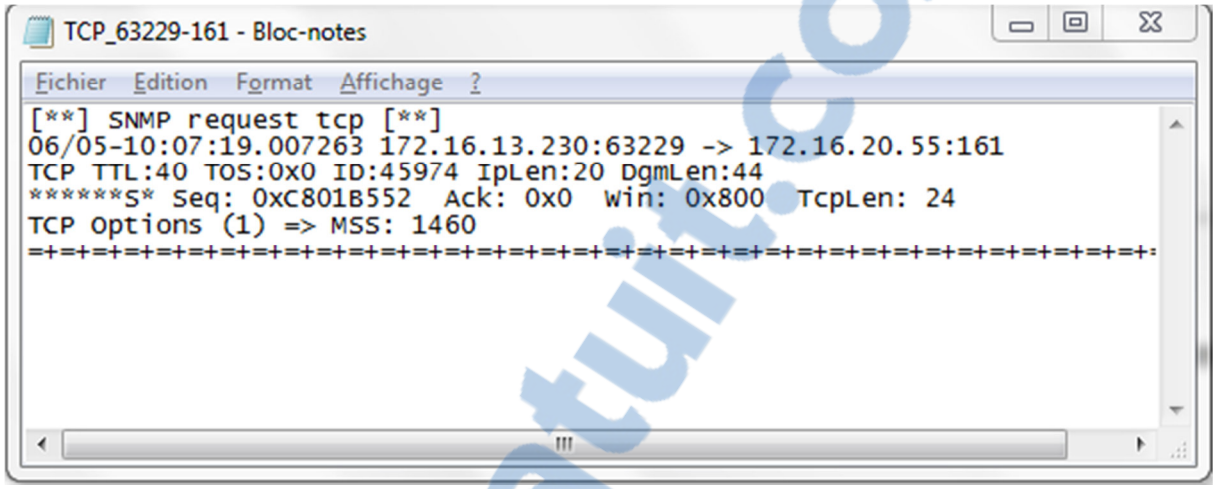

Figure II.9-a Alerte TCP

Alerte UDP :

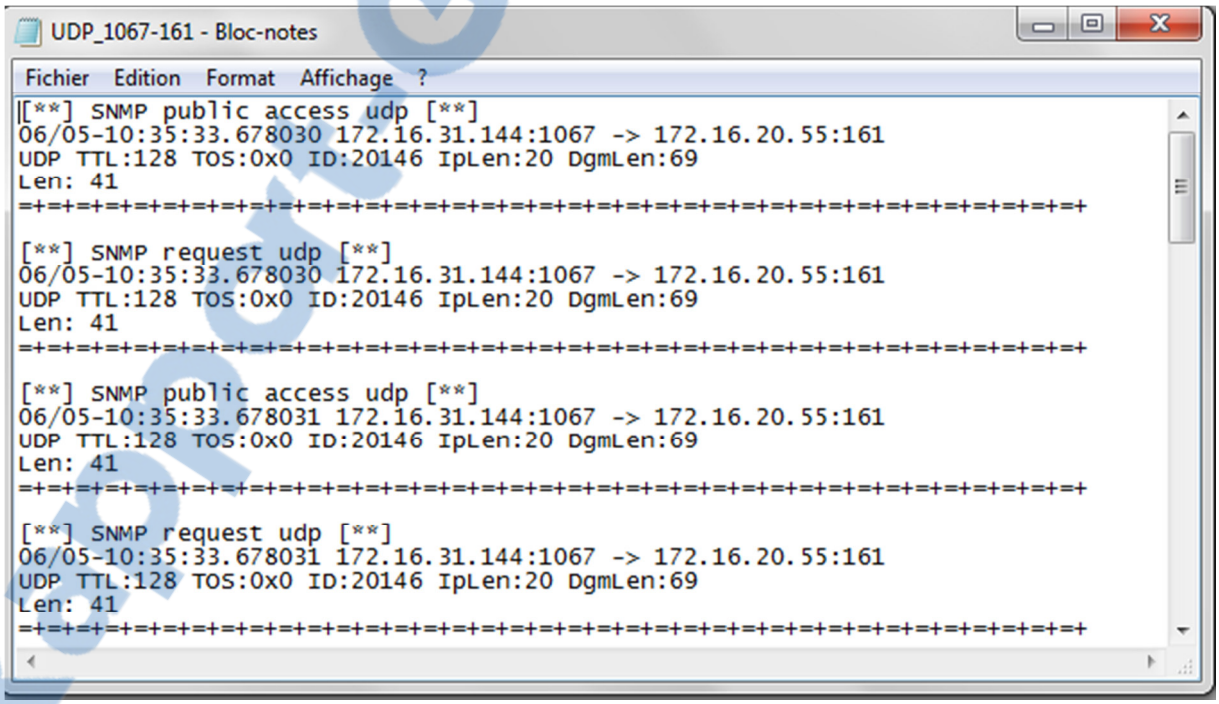

Figure II.9-b Alerte UDP

II.10- Conclusion

Dans ce chapitre, il a été démontré que le mode non connecté du protocole TCP peut être exploité pour effectuer des intrusions, qui par la suite ont été détectés grâce à l'outil SNORT, qui est un système de détection d'intrusion ou « IDS », se basant sur des règles afin de générer des alertes.

Il n'y a pas que la couche transport qui peut être une porte d'entrée pour les systèmes informatique, un pirate peut exploiter toutes les failles des couches TCP/IP, y compris la couche internet où le protocole IP fonctionne.

Chapitre III : Intrusion de la couche réseau

III.1-Introduction

Comme il a été cité précédemment, la couche « internet » fournit un adressage hiérarchique, un routage et un acheminement des paquets grâce aux différents protocoles qui y sont opérationnels, tel que l'IP et l'ICMP, dont les failles peuvent être utilisées pour des tentatives d'intrusion.

Ce chapitre va présenter les deux principaux protocoles de la couche « internet », les techniques utilisées pour des intrusions et les alertes obtenues lors d'une détection d'attaque.

III.2- présentation du protocole IP

« Internet Protocol » assure le service attendu de la couche Internet du modèle TCP/IP. Son rôle est donc de de gérer l'acheminement des paquets issus de la couche transport entre les nœuds de manière indépendante.

Il offre un fonctionnement non fiable et sans connexion à base d'envoie/réception de flux de bit structuré ou « datagrammes ».

« Non fiabilité » : c'est une absence de garantie que les datagrammes arrivent à destination. Ils peuvent être perdus, altérés ou dupliqués sans que ni la source ni la destination ne le sachent. On parle dans ce cas de « remise au mieux ».

« Sans connexion » (mode non connecté) : chaque datagramme est traité et acheminé de manière indépendante des autres.

Le rôle du protocole IP étant de déterminer le chemin entre les nœuds intermédiaire « les routeurs », il faut donc disposer d'un mécanisme permettant d'identifier les nœuds de manière unique. Pour assurer une communication entre nœuds, ce protocole se base sur un adressage hiérarchique. [6]

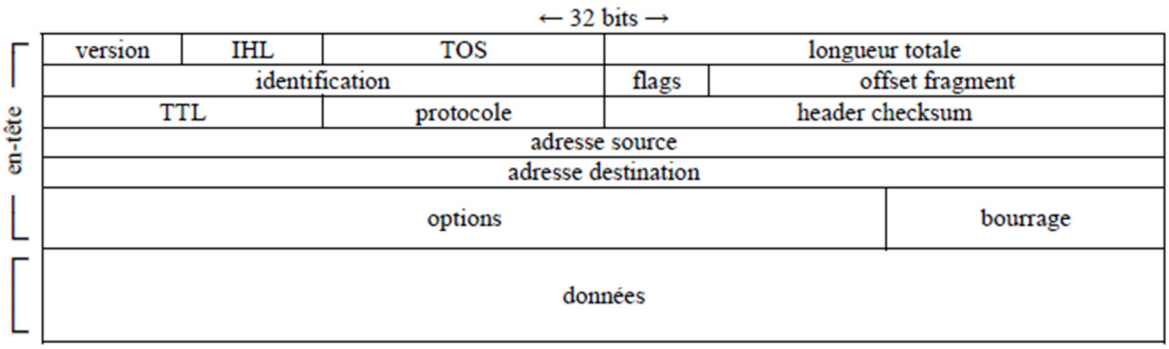

Figure III.2- Format d'un paquet IP

Les différentes informations contenues dans les champs d'entête IP sont les suivantes :

- ***Version (4bits) :*** numéro de version du protocole IP, 4 pour IPv4, 6 pour IPv6.
- ***IHL « Internet Header Length » (4bits) :*** longueur d'entête en mots de 32bits.
- ***TOS « Type Of Service » (8bits) :*** type de service utilisable pour certains routeurs pour diriger éventuellement plus efficacement le datagramme afin de répondre à ses exigences généralement ignorés.
- ***Longueur totale (16bits) :*** longueur du datagramme en octets.
- ***Identification (16bits) :*** numéro de fragment, utile si le datagramme a dû être pendant le transit
- ***Flags (3bits) :*** indicateur de fragmentation
  - Bit inutilisé (1bit) : 0
  - DF, Dont Fragment (1bit) : ne doit pas être fragmenté, 1.
  - MF, More Fragment (1bit) : dernier fragment, 0.
- ***Offset Fragment (13bits) :*** décalage du fragment dans le datagramme originel en mots de 64 bits, 0 pour le 1er bit, 0 si le datagramme n'a pas été fragmenté.
- ***TTL, Time To Live (8bits):*** durée de vie en seconde du datagramme lors du transit, décrémenté de 1 à chaque passage via un routeur, ou plus s'il stagne dans sa file d'attente, détruit si la durée de vie atteint 0. Généralement initialisé à 64 ou 128.
- ***Protocole (8bits) :*** nom du protocole encapsulé, 0 pour IP, 1 pour ICMP, 2 pour IGMP, 6 pour TCP, 17 pour UDP.
- ***Header Checksum (16bits) :*** somme de contrôle de vérification de l'entête, recalculée lors de chaque passage par un routeur.
- ***Adresse source (32bits) :*** adresses IPv4 du nœud source sur 4 octets.

- ***Adresse destination (32 bits) :*** adresse IPv4 du nœud destination sur 4 octets.
- Les informations optionnelle du champ entête sont très peu utilisées : sécurité, gestion, route, datation… etc. Si une ou plusieurs options sont spécifiées, le champ entête est rempli de bit de bourrage égal à 0, afin d'obtenir une longueur multiple de mots de 32 bits. [6]

III.3- Techniques utilisées pour les intrusions IP

Une des techniques qui exploitent le protocole IP sont les suivantes :

III.3-a IP spoofing

C'est une technique qui remonte aux années 90, elle vise à s'infiltrer dans un ordinateur en se faisant passer pour un autre en qui il a confiance, cela en envoyant un paquet dont l'adresse IP est autorisée par le serveur. La source va donc tromper la cible qui accorde l'accès en pensant avoir affaire à une machine de confiance.

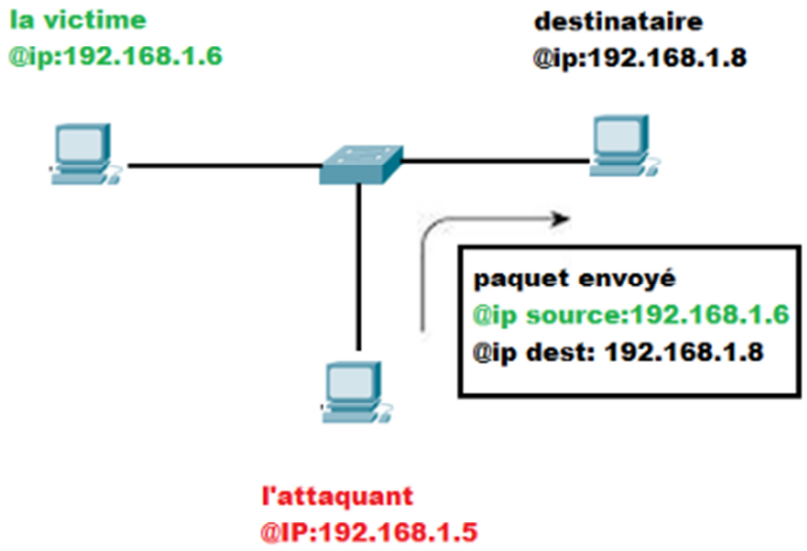

Figure III.3 Principe d'un IP spoofing

III.3-b Teardrop Attack :

Dans ce type d'attaque, une faille du protocole IP est exploitée. Il est déjà connu qu'un paquet IP de taille importante est envoyé en plusieurs fragments contenant chacun un numéro de séquence et à la réception, le destinataire réassemble les paquets grâce aux valeurs de décalage erronées ainsi la victime à la réception ne pourra pas réassembler correctement les paquets ce qui mène à un crash du système. [12]

III.4- Présentation du protocole ICMP

« Internet Control Message Protocol », est un protocole de couche Internet du modèle IP qui permet d'envoyer des messages d'erreurs suite à des erreurs constatées sur un datagramme, des messages d'interrogation ou d'information et tester l'accessibilité des machines sur le réseau. Utilisé dans la commande « PING ». Les messages ICMP étant intégrés dans les datagrammes IP dont le champ protocole vaut 1, aucune garantie n'est assurée pour une bonne transmission. Il ne permet pas de fiabiliser une transmission mais plutôt de déterminer les causes d'un problème en proposant un compte-rendu. [13]

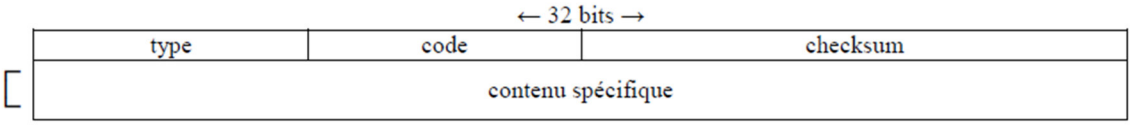

Figure III.4 Format d'un paquet ICMP

- ***Type (8bits) :*** type de message transmit.
- ***Code (8bits) :*** code précisant ce type de message.
- ***Total de contrôle (ou checksum) (16bits) :*** somme de contrôle de vérification du message.

III.5- Les techniques utilisées pour les intrusions ICMP

III.5-a Ping Flooding

C'est une attaque qui consiste tout simplement à envoyer à la victime un nombre maximal de Ping jusqu'à saturation. Une des conditions pour que cette méthode fonctionne est de posséder plus de bande passante que la cible.

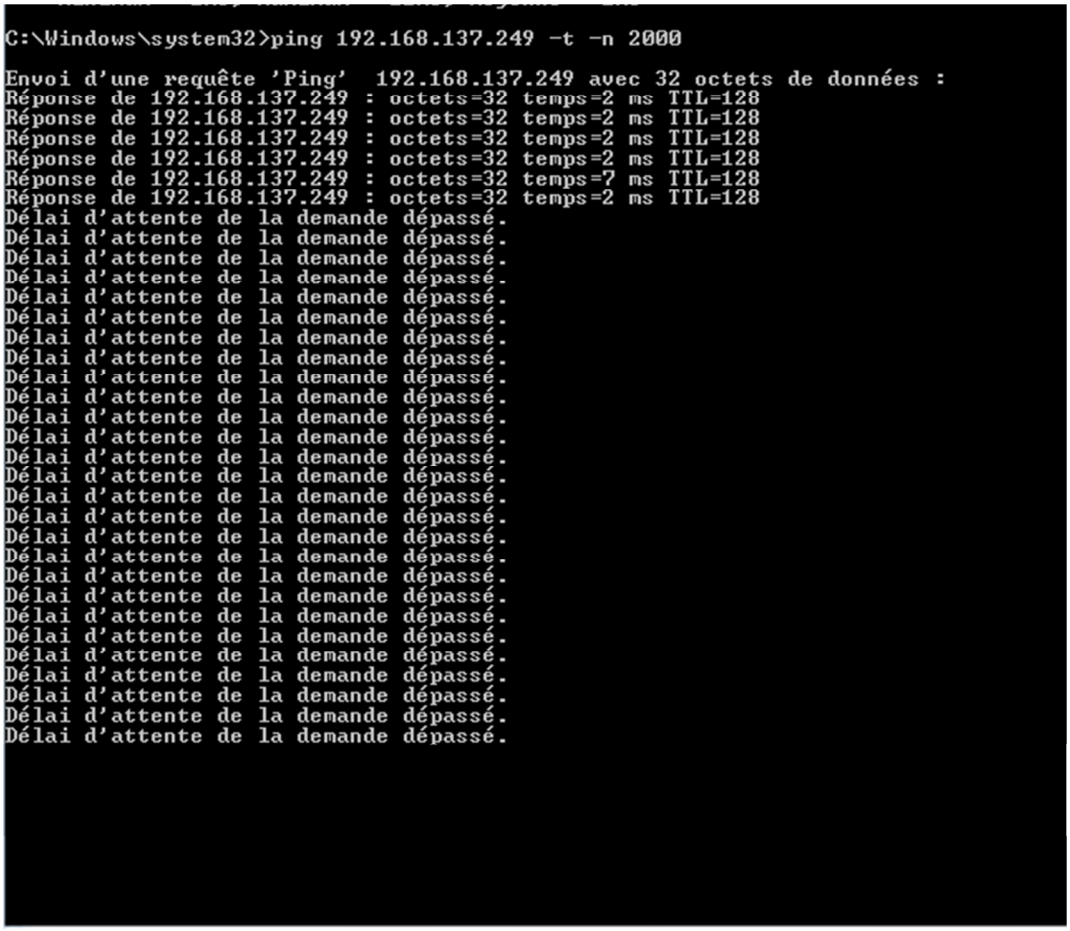

Figure III.5-a Réponse de la cible subissant un ping flooding

III.5-b Ping de la mort

Le pirate envoie à la victime un message de taille supérieure à celle autorisée (supérieure à 65536), le paquet est donc fragmenté et à la réception la cible réassemble les fragments reçus dans le buffer dont la taille est dépassée, ce qui va mener à un plantage de la cible.

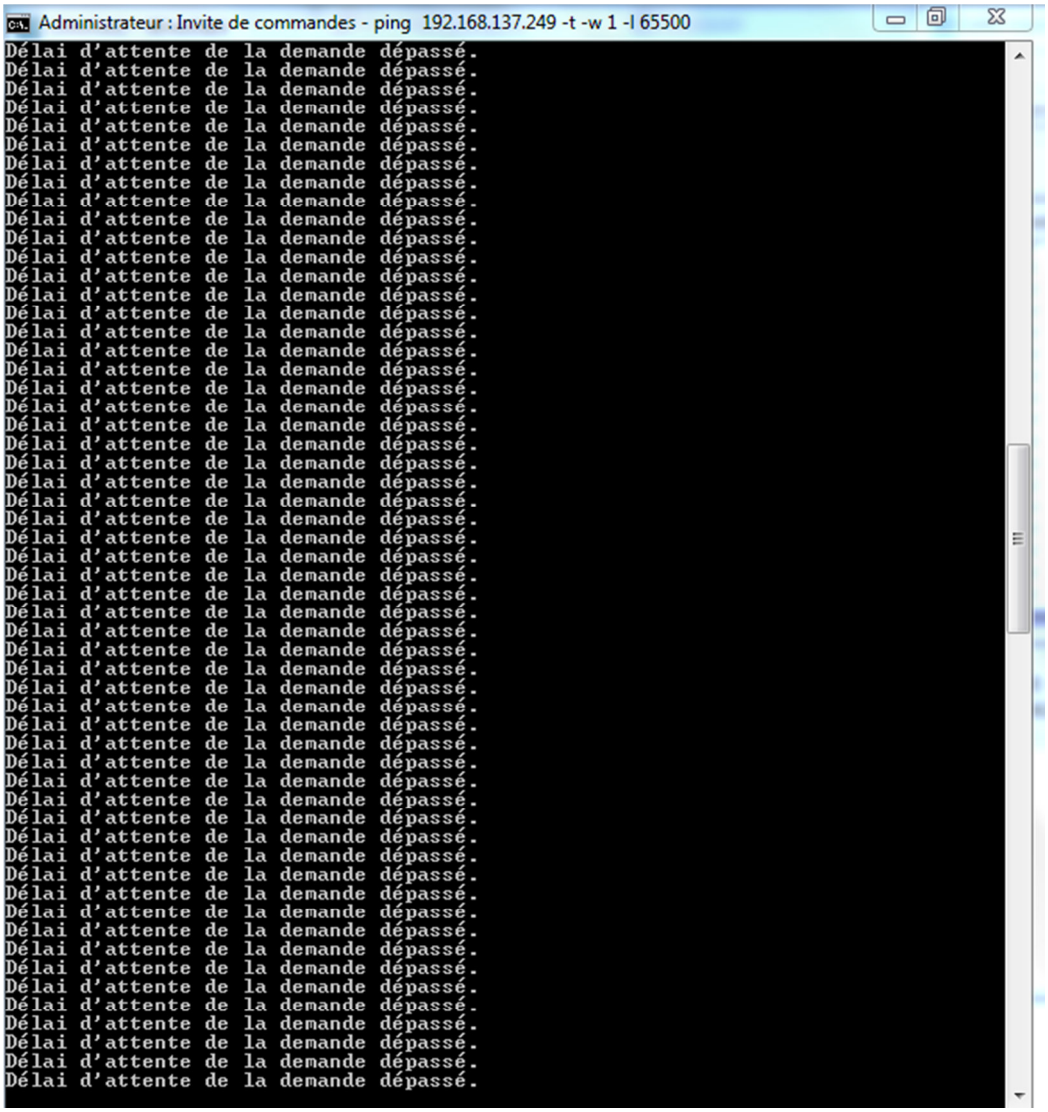

Figure III.5-b Réponse de la cible subissant un ping de la mort

III.5-c Smurf

C'est une attaque qui utilise l'adresse de diffusion de réseau, l'attaquant envoie un paquet à l'adresse broadcast contenant l'adresse IP source de la victime, ainsi chaque machine du réseau va répondre à chaque requête de l'attaquant en envoyant des paquets à la cible.

II.6- Détection

Encore une fois, SNORT nous a permis de récolter quelques attaques et lancer des alertes ICMP sur la station 172.16.20.55.

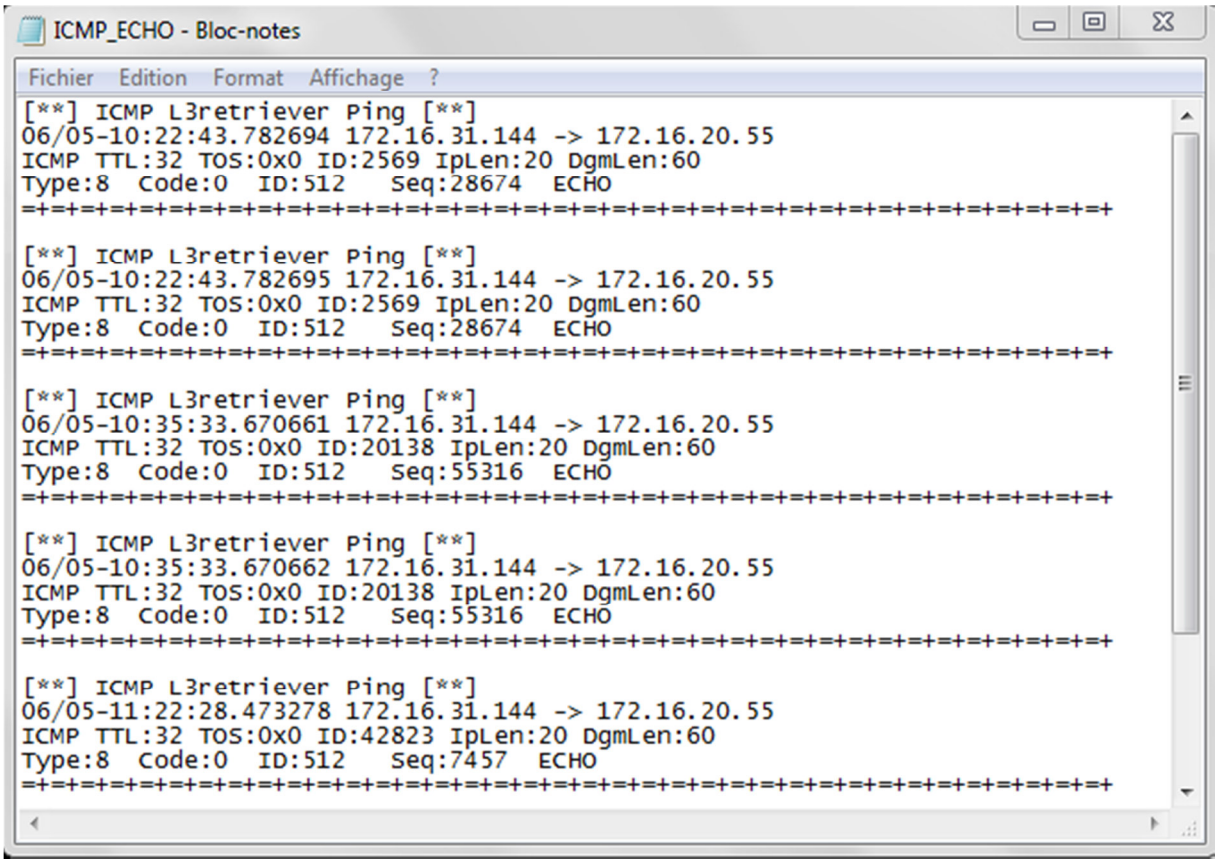

Figure III.6 Alerte ICMP

II.7- Conclusion

Ce troisième chapitre a détaillé les protocoles de la couche Internet « IP et ICMP », quelques attaques qui exploitent leurs failles ensuite une détection en lançant des alertes grâce à SNORT utilisé comme IDS.

Chapitre IV: Configuration et exploitation de SNORT

IV.1- Introduction

Automatiser la détection et alertes relatives aux intrusions dans les réseaux et systèmes numérique est une tâche ardue. En effet, nous avons établi que ces dernières exploitent le fonctionnement d'un protocole ou logiciel, il faut choisir et développer un outil pour cela. Nous avons opté pour le logiciel open source dit SNORT qui peut être adapté comme un dispositif de détection des intrusions IDS : Intrusion Detection System.

Au démarrage, SNORT utilise un fichier de configuration qui inclut de nombreux paramètres. Ce fichier, contrôle tous les aspects du fonctionnement de SNORT : ce qu'il faut surveiller, la manière dont il faut se défendre contre les différentes attaques et les règles à utiliser pour détecter les anomalies.

Ce chapitre décrit en détail la configuration et la mise en service de l'IDS SNORT, par la suite il présente une application que nous proposons pour une utilisation plus facile.

IV.2- Le fichier de configuration « snort.conf »

Pour assurer le bon fonctionnement de l'IDS SNORT, l'utilisateur doit impérativement effectuer des changements dans le fichier de configuration « snort.conf », qui peuvent différer d'une version à une autre. A commencer par :

IV.2-a Configuration de variables :

Cette section est dédiée aux informations de configuration utilisées par SNORT afin de déterminer la fonction ou la localisation de certains systèmes. Elle lui permet de lire l'architecture de l'environnement qu'on désire surveiller, ainsi procéder à des choix éclairés concernant un trafic douteux qui nécessite une alerte. Elle spécifie :

- La plage d'adresses IP interne à surveiller : « var HOME_NET ».

```
var HOME_NET 172.16.20.55/19
```

- La plage d'adresses IP externe au réseau à surveiller : « var EXTERNAL_NET ».

```
var EXTERNAL_NET any
```

- La liste des serveurs installés dans la station à surveiller :

```
# List of DNS servers on your network
ipvar DNS_SERVERS $HOME_NET

# List of SMTP servers on your network
ipvar SMTP_SERVERS $HOME_NET

# List of web servers on your network
ipvar HTTP_SERVERS $HOME_NET

# List of sql servers on your network
ipvar SQL_SERVERS $HOME_NET

# List of telnet servers on your network
ipvar TELNET_SERVERS $HOME_NET

# List of ssh servers on your network
ipvar SSH_SERVERS $HOME_NET
```

IV.2-b Configuration de pré-processeurs

Implémentés à partir de la version 1.5, les pré-processeurs permettent d'étendre les fonctionnalités de SNORT. Ils sont exécutés avant le lancement du moteur de détection et après le décodage du paquet IP.

Le paquet peut être modifié ou analysé de plusieurs manières en utilisant le mécanisme de pré-processeurs.

Ils sont chargés et configurés avec le mot-clé « preprocessor ».Le format de la directive preprocessor dans la règle de SNORT est :

Preprocessor<nom> : <options>

Exemple de pré-processeurs :

Le détecteur portscan : il permet de :

- Enregistrer le début et la fin d'un scan de ports à partir d'une seule adresse IP.
- Lorsqu'un fichier de log est spécifié, ce pré-processeur journalise les IP et les ports scannés ainsi que le type de scan.

Leur configuration diffère d'une version à une autre. L'exemple suivant, illustre une configuration de pré-processeur :

Dans le fichier de configuration, le chemin des pré-processeurs doit être introduit :

C:\snort\lib\snort_dynamicpreprocessor\

Ensuite, copier les chemins des différents types de pré-processeurs :

dynamicpreprocessor C:\snort\lib\snort_dynamicpreprocessor\sf_dce2.dll

IV.2-c Configuration de la destination des résultats

L'un des avantages majeurs de SNORT, est son aptitude à envoyer les alertes et autres informations de détection vers de nombreuses destinations. Des modules de destinations appelés après les pré-processeurs. On peut citer les modules suivants :

- Alert_syslog : permet d'envoyer les alertes et de spécifier la facilité de journalisation et la priorité dans le fichier de règles de snort.

Format de l'alerte : alert_syslog :<facilité ><priorité><options>

- Alert_fast : une méthode très rapide qui imprimera les alertes dans un format rapide vers un fichier (alert.ids) sans afficher les entêtes des paquets.

Pour l'obtenir, il faut introduire la commande suivante dans le fichier de configuration :

Output alert_fast : alert.ids

- Alert_ful : contrairement à « alert_fast », cette méthode est plutôt lente. Elle affiche l'intégralité des entêtes de paquets après une analyse détaillée effectuée par le programme. Les alertes seront écrites dans le fichier par défaut (log) ou un fichier spécifié dans la ligne de commande.

La commande introduite dans le fichier de configuration est comme suit :

Output aleret_full : alert.ids

- Log_tcpdump : c'est une méthode qui prend le nom de fichier de sortie comme argument. Elle permet d'enregistrer les alertes dans un fichier au format « tcpdump ».

Output log_tcpdump : snort.log

- Base MySQL: les alertes récoltées peuvent être aussi stockées dans une base de données « MySQL ». Cela est bénéfique dans le cas où il existe plusieurs sondes dans un ou plusieurs réseaux, l'analyse des activités enregistrées dans la base de données permet d'avoir une vue d'ensemble de toutes les activités suspectes en centralisant les alertes dans un seul endroit.

  Pour cela, on doit d'abord créer : une base « MySQL », les tables à l'aide d'un script fournit avec le package SNORT, et enfin chaque instance de SNORT alimente la base en logs et en alertes.

  Une configuration dans le fichier SNORT.CONF s'avère indispensable pour une bonne connexion entre SNORT et MySQL, il faut donc introduire le nom de la base de données, de l'utilisateur, le mot de passe de l'utilisateur et l'hôte en possession de cette base.

```
output database: alert, postgresql, user=snort dbname=snort
                dbname=snort1 host=172.16.20.55
```

  Dans le même fichier, il faut spécifier les actions de SNORT associées avec MySQL :

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

IV.2-d Utilisation des fichiers spécifiant les règles de détection :

La dernière étape, consiste à inclure dans le fichier « snort.conf » le chemin des règles :

var RULE_PATH c:\snort\rules

Ensuite, inclure les chemins des différents types de règles désirées :

include $RULE_PATH/attack-responses.rules

Deux fichiers sont utilisés pour attribuer un ordre de priorité aux alertes, il suffit d'inclure les lignes suivantes :

include classification.config

include reference.config

V.3- Réalisation d'une application sous windows

SNORT est une application qui fonctionne en ligne de commande avec plusieurs modes d'exécution. Nous avons jugé utile de regrouper les différents modes que nous avons étudiés et utilisé dans une application sous Windows. De plus, elle permet une utilisation facile du système de détection d'intrusion SNORT. L'application personnalisée que nous proposons a été développée sous l'environnement « Visual Studio 2012 ».

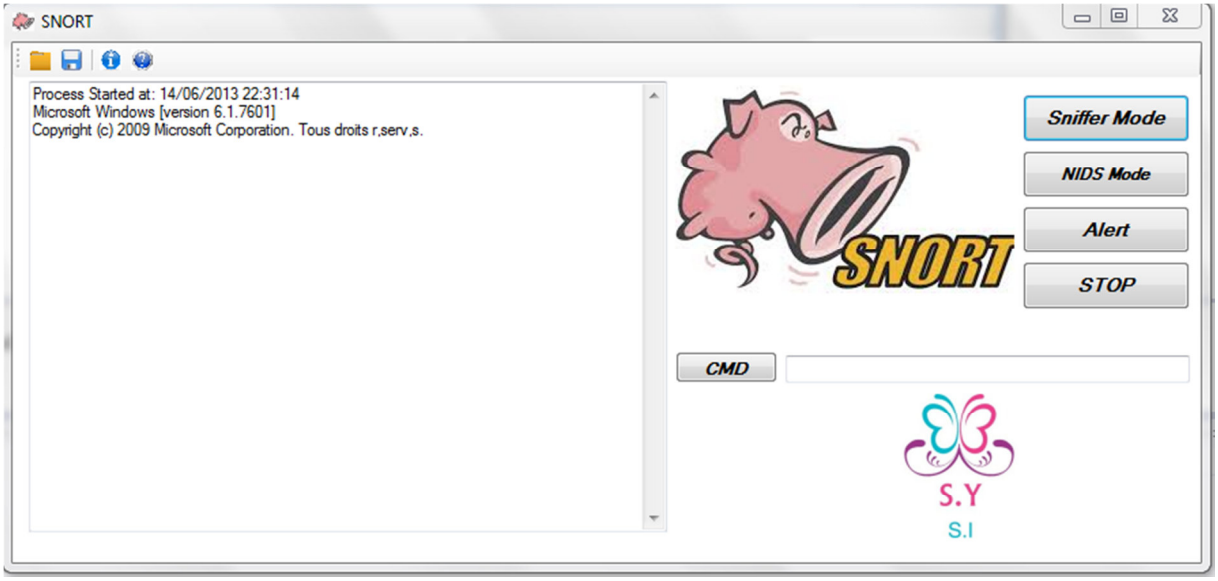

Figure IV.3-a L'interface réalisée

- Le bouton ***« Sniffer Mode »,*** permet l'exécution de SNORT en mode sniffer et d'afficher les paquets récoltés dans le « TextBox » à gauche. La figure suivante illustre la capture de paquets :

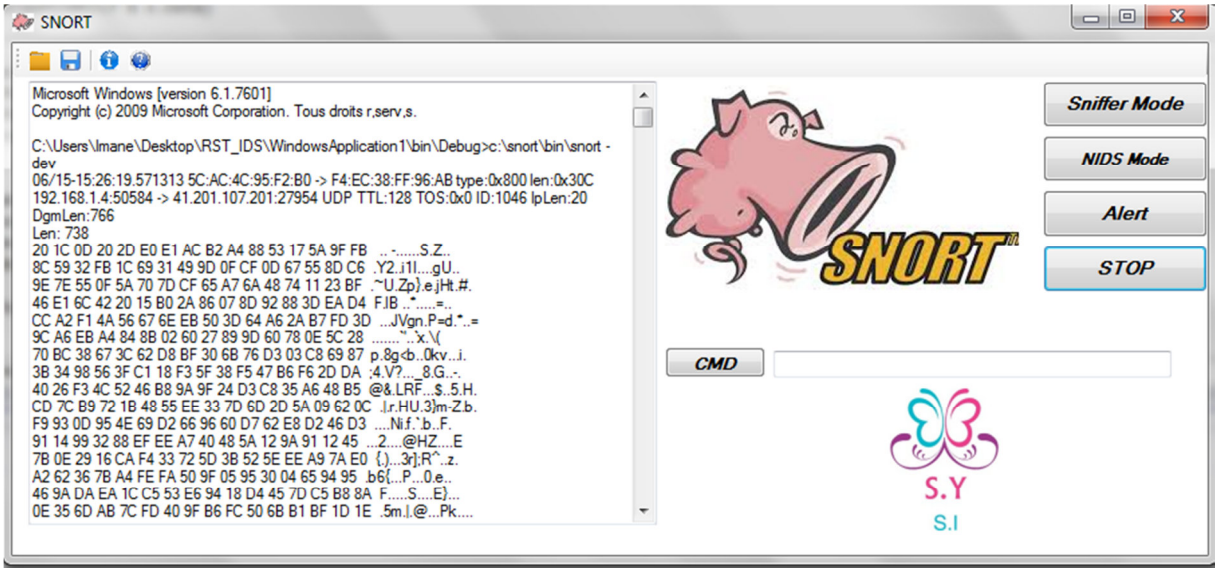

Figure IV.3-b SNORT en mode Sniffing

- Le bouton « NIDS Mode » permet de lancer SNORT en mode NIDS et utiliser le fichier de configuration, pour lire le chemin des règles et la destination des résultats. Dans notre cas, les alertes seront enregistrées dans le fichier « alert.ids » qu'on a spécifié au préalable dans « snort.conf ».
- Le bouton « Alert » permet de consulter le fichier où les alertes sont enregistrées « alert.ids ».

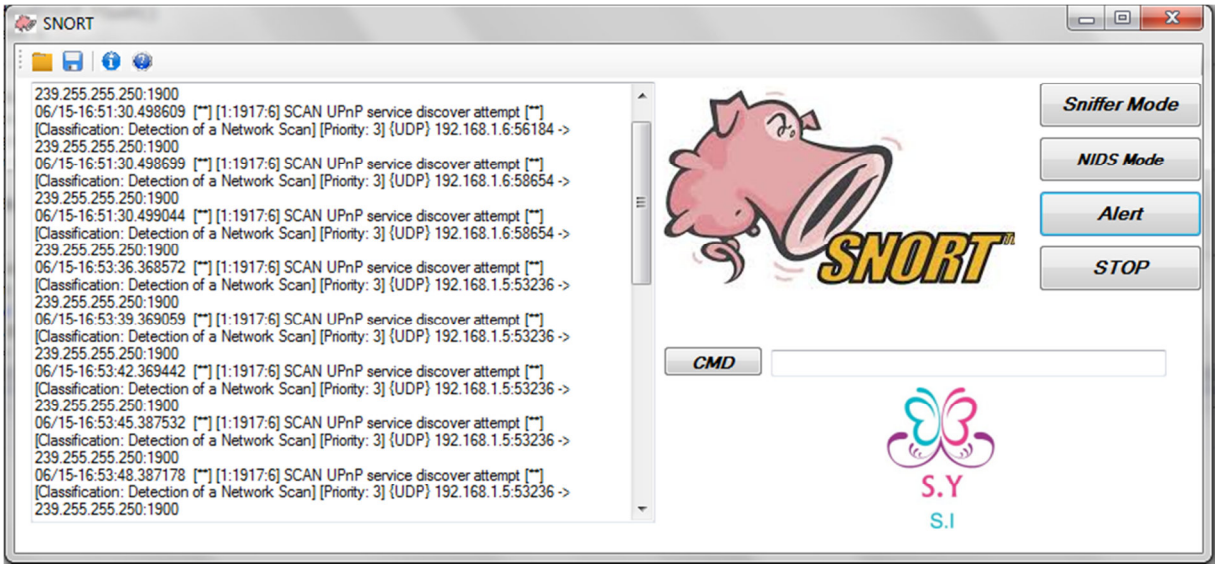

Figure IV.3-c fichier « alert.ids », contenant les alertes obtenues

- Le bouton « STOP » sert à arrêter la capture lancée.
- La zone de texte permet de saisir une commande, et de l'exécuter grâce au bouton « CMD ».

Figure IV.3-d zone de texte pour saisie de commande CMD

- La barre d'outils possède quatre boutons :

Figure IV.3-e Barre d'outils

- ***« Open »*** : donne la possibilité d'ouvrir un fichier enregistré au préalable.
- ***« Save »*** : pour enregistrer les alertes obtenues.
- ***« About »*** : donne des informations sur l'interface et ses conceptrices.
- ***« Help »*** : permet d'accéder au manuel de SNORT pour obtenir de l'aide.

V.4 Conclusion

SNORT se base sur un fichier de configuration pour accomplir ses tâches de détection d'anomalies, lancement d'alertes, les enregistrer et les journaliser, grâce à une configuration de paramètres effectuée au préalable. Ces paramètres diffèrent d'une version à une autre.

Nous avons également réalisé une interface graphique sous l'environnement « Visual Studio 2012 », afin de faciliter l'utilisation de SNORT.

Figure IV.4-f About

Conclusion générale

Conclusion générale

Les systèmes d'information se révèlent d'autant plus ouverts sur internet. Ce qui est à priori bénéfique et pose néanmoins un problème majeur : engendrer de diverses attaques qui peuvent aller jusqu'à la nuisance. De ce fait, la mise en place d'une politique de sécurité autour de ces systèmes s'avère primordiale.

Un système de détection d'intrusion à l'image de SNORT, a la possibilité d'évaluer le trafic circulant dans le réseau et l'analyser en comparant avec des règles ou signatures. Admettant qu'une anomalie soit détectée, il lance une alerte précisant la date et l'heure où l'attaque a été effectuée ainsi que les adresses IP de la source et la victime et le protocole utilisé. A noter que ces systèmes ne peuvent pas fonctionner sur la couche physique.

Le cœur de ce travail revient à étudier profondément le fonctionnement les protocoles de la couche Transport et Internet de la pile TCP/IP, les techniques d'attaques qui exploitent leurs failles et la détection d'intrusion, à commencer par :

- La couche Transport : qui gère la qualité de service, la fiabilité et le contrôle de flux .

  Le Protocole TCP en est un protocole fiable, assurant ses fonctionnalités en mode connecté. Bien qu'il soit très performant, son mode connecté peut être exploitées pour des fins suspectes par des malfaiteurs.

Contrairement au TCP, l'UDP n'assure pas de fiabilité, et fonctionne en mode non connecté. Il est peut être exploité également pour des intrusions.

Ensuite :

- La couche Internet : prend en charge le routage des données, l'adressage et l'acheminement des paquets. Régie par le protocole IP qui assure parfaitement ses fonctionnalités avec un mode non connecté, et sans aucune fiabilité.

  Le Protocole ICMP, aussi opère sur cette couche, et peut engendrer quelques attaques.

  La détection s'est effectuée à l'aide de SNORT, après une configuration, qui est loin d'être triviale, du fichier « snort.conf » et une spécification de la destination des résultats obtenus.

comme SNORT est un IDS qui s’exécute en lignes de commandes avec un nombre important de paramètres, nous avons développé une application personnalisée en utilisant l'IDE (Integrated development Envierment) « Visual Studio 2012 ». L'application permet de masquer la complexité de SNORT à un agent chargé de superviser un réseau et enregistrera intuitivement les alertes récoltées.

Glossaire

ACK	**ACK**nowledgment
ARP	**A**dress **R**esolution **P**rotocol
CD ROM	**C**ompact **D**isc **R**ead **O**nly **M**emory
CGI	**C**ommun **G**ateway **I**nterface
DARPA	**D**efense **A**dvanced **R**eserch **P**roject **A**gency
DDOS	**D**istributed **D**enial **o**f **S**ervice
DIDS	**D**istributed **I**ntrusion **D**etection **S**ystem
DMZ	**D**e**M**ilitarized **Z**one
DNS	**D**omain **N**ame **S**ervice
DOS	**D**enial **O**f **S**ervice
FIN	**FIN**alize
FTP	**T**ransfer **P**rotocol
HIDS	**H**ost **I**ntrusion **D**etection **S**ystem
HTTP	**H**yper**T**ext **T**ransfer **P**rotcol
ICMP	**I**nternet **C**ontrol **M**essage **P**rotocol
IDE	**I**ntegrated **D**evelopment **E**nvironment
IDS	**I**ntrusion **D**etection **S**ystem
IHL	**I**nternet **H**eader **L**ength
IP	**I**nternet **P**rotocol
IPS	**I**ntrusion **P**revention **S**ystem
MILNET	**MIL**itary **NET**work
NIDS	**N**etwork **I**ntrusion **D**etection **S**ystem
PSH	**PuSH**
RFID	**R**adio **F**requency **ID**entification
RHOST	**R**emote **HOST**
RPORT	**R**emote **PORT**
RST	**ReSeT**
SAINT	**S**ystem **A**dministrator's **I**ntegrated **N**etwork **T**ool
SSH	**S**ecure **SH**ell

SSL	**S**ecure **S**ockets **L**ayer
SQL	**S**tructured **Q**uery **L**anguage
SYN	**SYN**chronisation
TCP	**T**ransmission **C**ontrol **P**rotocol
TELNET	**TEL**ecommunication **NET**work
TOS	**T*ype*** **O**f **S**ervice
TTL	**T**ime **T**o **L**ive
UDP	**U**ser **D**atagramme **P**rotocol
URG	**URG**ent
USB	**U**niversal **S**erial **B**us

Bibliographie

[1]: « Sécurité informatique, Ethical hacking, Apprendre l'attaque pour mieu se défendre »; ACISSI, Edition ENI, octobre 2009.

[2]: « Craquage de mots de passe Windows et Linux » ; Lionel Coin & Sebastien Pone; 2008-2009. PDF

[3]: « Le grand livre de SécuritéIn fo.com » ; 19 février 2004. PDF

[4]: « Sécurité optimale » ; Septembre 1999 ; Edition Campus Press.

[5]: « Protection contre les attaques par déni de service dans les réseaux IP »; HOTTE Marion, LUTUN Quentin-Edouard; ASCOET Thomas.

[6]: « Réseaux Informatiques ; Modèle OSI et protocole TCP/IP »; 20 mai 2005.

[7]: « Nmap secrets »; Professors Judy & James Messer; 2006.

[8]: « Attaques Informatiques »; Jean-Olivier Gerphagnon, Marcelo Portes de Albuquerquet & Màrcio Portes de Albuquerquet; Centro Brasileiro de Pesquisas Fisicas.

[9]: « Installation et Configuration d'un système de Détection d'intrusion (IDS) »; Ibrahim Mohamed Amine & Tebourbi Hamdi; 19 Janvier 2009.

[10]: « NT réseaux IDS et IPS »; Nicolas BAUDOIN & Marion Karle.

[11]: « Mise en place d'une sonde SNORT »; M. Fethi BEN NASR & Mme Alia KHESSAIRI ABBASI ; 2004-2005.

[12] : «Dénis de service par usurpation d'identité » ; BALLAN Emilie & SURANGKANJANAJAI Gaetan.

[13] : « TCP/IP, Architecture, protocoles, Applications », Douglas Comer, Edition DUNOD

Résumé

Les informations circulant dans les réseaux informatique dis Intranet peuvent intéresser des pirates informatiques d'où la mise en place d'un système de détection d'intrusion devient nécessaire.

L'objectif de ce projet revient à étudier en un premier temps le fonctionnement des protocoles de la pile TCP/IP ainsi que les failles exploités moyennant des outils comme « Wireshark » et « Nmap » et ensuite la détection des intrusions en utilisant "SNORT" un IDS open source capable d'effectuer en temps réel des analyses de trafic et se basant sur un fichier de configuration "snort.conf" pour générer des alertes

Mots clés :

Réseau, protocole, intranet, sécurité, faille, intrusion, détection, IDS, configuration, alerte.

Abstract

The implementation of an Intrusion Detection System is required in computer network called Intranet in order to prevent, detect as well as to watch from malicious activities.

the purpose of this project is to study at first protocols of the stack TCP/IP and security flaws exploited using tools as Wireshark and Nmap and then intrusion detection by using an IDS open source called Snort that analyses in real time a network traffic based on a configuration file "snort.conf "and generate alerts.

Key words:

Network, protocol, security, intranet, flaw, intrusion, detection, IDS, configuration, alert.