الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoire de Master

Filière Électronique
Spécialité Réseau & Télécom

Présenté par

FLITTA Adel

&

MARIKO Ibrahim Steve Biko

Détection de l'utilisation de Torrent dans une entreprise

Proposé par : Dr. MEHDI Merouane

Remerciements

Tout d’abord, nous tenons à remercier le bon Dieu tout Puissant de nous avoir donné la force et le courage de mener à bien ce modeste travail, et nous remercions spécialement nos parents, qu’ils nous ont apporté un soutien indéfectible à la fois financier et logistique et surtout morale ce qui a été un tremplin pour la concrétisation de ce accompli.

Nous tenons à remercier également tous ceux et celles qui ont contribué à finaliser ce modeste travail.

Nos remerciements vont à Mr. Mehdi Merouane notre promoteur pour nous avoir guidé pour la réalisation de ce projet.

Enfin, nous remercions vivement Nos amis pour leur aide morale durant toute la période de préparation.

ملخص:

نحن نشهد في الوقت الحاضر على انفجارللمعلومات التي تتدفق عبر الإنترنت، من بين تلك المعلومات التي لدينا معلومات شخصية والوسائط المتعددة تحت حقوق التأليف والنشر والتراخيص المجانية، التورنت وسيلة للحصول على هذا محتوى من الإنترنت، على مواقع الويب التي تقترح هذا النوع من التحميل، يشهد الموضوع جدل كبير، محظور استخدامه في العديد من الشركات والأعمال الجادة، أصبح من واجبنا الآن ، من خلال برنامج الكشف عن الاقتحام سنورت وضع سياسة للكشف عن استخدام كل ما يتعلق بالتورنت.

كلمات المفاتيح: تورنت؛ الكشف؛ اقتحام؛ سنورت.

Résumé :

Nous assistions de nos jours à une déferlante d'information transitant sur internet, parmi ses données nous avons les informations personnelles, du contenu multimédia sous copyright et du contenu libre, le torrent est un moyen d'acquérir du contenu sur internet, sur des sites qui proposent ce genre de téléchargement, étant souvent controversé de part son utilisation, il n'est pas autorisé dans la plupart des entreprises et des structures sérieuses, il relève ainsi de notre devoir, via un logiciel de détection d'intrusion appelé Snort et un sniffer de paquets de mettre en place une politique de détection de l'utilisation de tout ce qui a trait au torrent.

Mots clés : Torrent ; détection ; intrusion ; Snort ; sniffer ; paquets.

Abstract :

We are assisting nowadays to blast of information flowing through internet, among those informations we have personal data, multimedia under copyrights and free licences, the Torrent, is a way to acquire content from internet, on websites that propose the torrent, subject of a great polemics, in numerous of serious corporations and businesses its use is banned, it's now our duty, via an intrusion detection software called Snort and a packet sniffer to put in place a politic of security permitting to detect everything in relation with torrent.

Keywords : Torrent ; detection ; intrusion ; Snort ; packet ; sniffer.

Listes des acronymes et abréviations

P2P	Peer To Peer.
IDS	Intrusion Detection System.
IP	Internet Protocol.
PAN	Personal Area Network.
LAN	Local Area Network.
MAN	Métropolitain Area Network.
WAN	Wide Area Network.
FDDI	Fiber Distributed Data Interface.
TCP	Transmission Control Protocol.
UDP	User Datagram Protoco.
OSI	Open Systems Interconnection.
MAC	Media Access Control.
ARP	Address Resolution Protocol.
IPV4	Internet Protocole Version 4.
NAT	Network Address Translation.
PAT	Port Adress Translation.
DDOS	Distrubuated Denial Of Service.
VLAN	Virtual Local Area Network.
VPN	Virtual Privet Network.
NIDS	Network Intrusion Detection System.
PDA	Personal Digital Assistant.
DNS	Domain Name Server.
DHT	Distributed Hash Table.
TLS	Transport Layer Security.
SSL	Secure Socket Layer.
TTL	Time To Live.
RTT	Round Time Trip.
ACK	Acknowledgment.
BASE	Basic Analysis And Security Engine.
ID	Identification.

Table des matières

Liste des figures

Liste des tableaux

Introduction générale

Internet est une interconnexion de plusieurs réseaux, il représente un nombre incommensurable de périphériques intermédiaires et finaux qui proposent différents services, à l'heure actuelle le partage de donnée est sans frein et se fait de différentes manières.

Parmi ces modes de partage chacun à sa particularité, son mode de fonctionnement.

Ces téléchargements et partages font appel à une grande vigilance niveau sécurité car ils ouvrent les systèmes et les réseaux à un grand nombre de failles de sécurité.

Le torrent est un mode de téléchargement basé sur une architecture P2P (point à point), le fichier à télécharger se retrouve en download et même temps en upload, il nous permet une grande économie de bande passante et un téléchargement solide.

Mais le torrent a un mauvais côté, car il fait appel souvent à la distraction, car les sites qui proposent ce genre de téléchargement ont souvent du contenu de loisir, au contenu sous copyright, ou encore du contenu malveillant, ce qui est indésirable au sein d'une entreprise, car il peut amener à des pertes de données et des problèmes judiciaires.

Pour pallier ces ennuies nous allons ici mettre une politique de sécurité basé sur les détections d'intrusions (IDS : intrusion détection system) ici en occurrence Snort,

Snort est un logiciel qui nous permet de mettre des alertes sur les comportements suspects et aussi les paquets suspects qui transitent sur le réseau.

À l'aide d'un sniffer ici nous allons mener une étude sur les traces particulière que l'utilisation du torrent laisse sur le réseau, sa signature.

Avec celle-ci nous allons donc détecter et prendre éventuellement des mesures préventives.

Au vu de tout ce qui précède, la question principale, à laquelle nous tenterons de répondre à travers cette étude, est la suivante :

La détection de l'utilisation du torrent dans un réseau entreprise, est-elle possible ?

Cette problématique nous guide aux questions suivantes :

- Qu'est-ce que le torrent et quels sont ses éléments ?
- Y a-t-il une signature qui peut différencier le trafic torrent et p2p sur le trafic web normal ?
- Peut-on implémenter ces signatures dans un système de détection d'intrusion réseau ?

Dans le cadre de préparation de notre mémoire de fin d'étude, qui porte essentiellement sur la détection de torrent dans une entreprise, on a suivi une méthodologie de recherche théorique et pratique.

- Théorique, basée sur la recherche documentaire auprès des bibliothèques virtuelles de recherche qui nous ont permis de consulter plusieurs ouvrages numériques afin de définir les concepts théoriques sur le sujet de notre recherche.
- Pratique, basée sur l'analyse du trafic torrent et celle du web normal suivi par l'implémentation d'un système de détection d'intrusion réseau Snort. Pour cela nous allons appuyer notre travail sur des tests réels.

Dans notre premier chapitre nous parlons des généralités sur les réseaux informatiques, les protocoles, nous survolons les notions d'architecture et de sécurité. Dans la deuxième partie nous nous attaquons au partage numérique, ses moyens et le peer to peer, le torrent et les différents moyens de partage P2P. Dans le chapitre 3 nous procédons à une analyse minutieuse grâce à Wireshark pour pouvoir extraire les empruntes (signatures) ainsi que la conception des règles. Enfin dans le chapitre 4 nous passons à l'analyse des résultats obtenues à la suite de nôtre détection.

Chapitre 1 Les réseaux informatiques

1.1 Introduction :

De nos jours l'information est le nerf de la guerre, quand on dit information, donnée, on parle d'internet le plus vaste réseau existant car une interconnexion de plusieurs réseaux comportant différentes technologies.

Nous allons voir ici les subtilités de ces différentes infrastructures, d'une manière générale.

1.2 Le réseau informatique :

Un réseau informatique est un ensemble d'ordinateurs reliés entre eux qui échangent des informations.

A ceci près qu'outre des ordinateurs, un réseau peut aussi contenir des équipements spécialisés [1].

Un réseau est composé du hardwares (matériel physique) et du software (logociel).

1.2.1 Le matériel :

Il représente l'ensemble des supports physique que nous distinguons à l'œil nu, qui intervienne à plusieurs niveaux, par exemple les périphérique intermédiaire, finaux et les supports de transmission.

- **Les périphériques finaux :** Ce sont les appareils qui donnent à l'utilisateur une interface pour communiquer avec d'autres utilisateurs, ces périphériques sont l'ordinateur, les téléphone IP, les smartphones, et beaucoup d'objets divers, comme les caméras de surveillances etc….
- **Les périphériques intermédiaires :** Ce sont des périphériques qui assure la connexion entre les périphériques intermédiaires et aussi leurs connexions avec les réseaux, nous avons le Hub le Switch et le routeur etc….

- **Support de communication :** Le support de communication est un élément réseau qui nous permet de donner une voie à nos données, ces voies (câbles Ethernets, fibre optique, ou ondes hertzienne pour le wifi).

1.2.2 Logiciel :

Formé à partir des mots logique et matériel, le mot logiciel a été inventé en 1969 pour remplacer le terme anglais software. Il désigne l'ensemble des programmes et des procédures nécessaires au fonctionnement d'un système informatique [2].

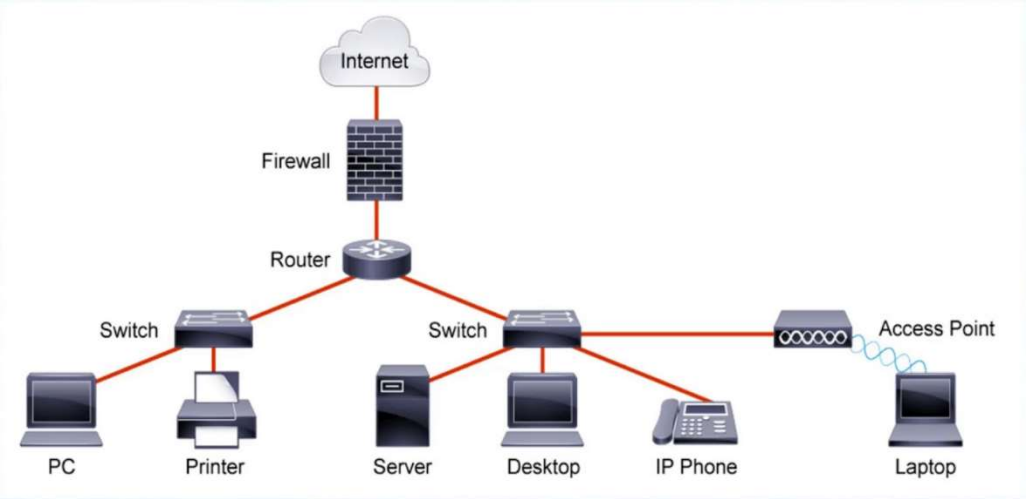


Figure 1.1. Exemple de réseau informatique.

1.3 Classification des réseaux informatique :

Nous avons une multitude de type de réseaux informatique nous allons les classées par ordre :

- **PAN (Personal Area Network) réseau personnel PAN :** C'est la plus petite taille de réseau existant sa taille ne dépasse pas les 100 m car il peut être constituer de deux téléphones connectés par Bluetooth ou de deux postes ordinateur connectés par Ad-Hoc [2].

- **LAN (Local Area Network) réseau local LAN :** plus grand que le réseau PAN il est comme un réseau d'université ou un réseau domestique [2].

- **MAN (Métropolitain Area Network) Réseaux métropolitain MAN :** beaucoup plus large que les réseaux LAN, ils sont une taille de réseaux qui englobe les

réseaux des grandes villes d'où l'appellatif métropolitain, mais ne sont toujours pas la plus grande catégorie [2].

- **WAN (Wide Area Network) Réseaux large WAN :** Ce sont les réseaux les plus étendu que nous ayons, il est constitué de MAN, LAN et PAN interconnecté entre eux, un bel exemple de ce type de réseau est **internet** même [2].

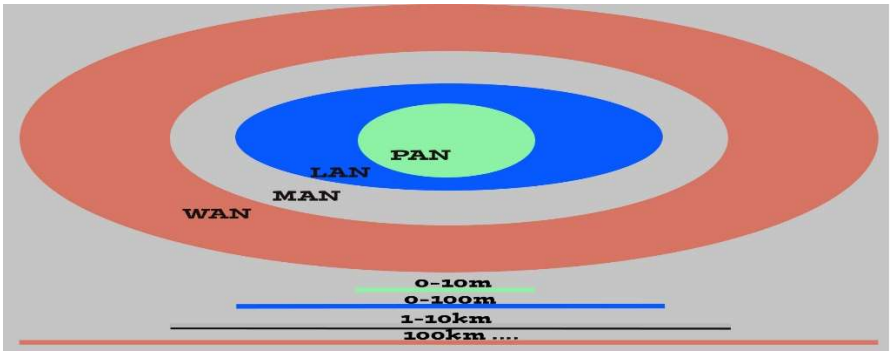


Figure 1.2. Classification des réseaux.

1.4 Les architectures Réseau :

Nous avons deux types d'architecture en réseau :

- Les architectures physiques.
- Les architectures logiques.

Qui décrive la disposition physique point de vu câblage, et la disposition point de circulation de données.

1.4.1 Architecture Physique :

Désigne et identifie la manière dont les périphériques, intermédiaires et finaux sont interconnectés entre eux.

- **La topologie en Bus** : Dans cette topologie tous les supports utilisent le même câble, pour faire transiter les données
- **La topologie en anneau** : Dans cette topologie, les postes sont connectés sur un Câble sous forme d'anneau qui est chargé de transmettre l'information sur le réseau, dans un seul sens, il a

- **La topologie en étoile** : c'est la topologie l a plus utilisé actuellement, tous les nœuds sont connecté à un point central, qui envoie l'information aux autres, avec un switch ou un hub comme point central.
- **La topologie maillée** : Dans cette topologie chaque poste est connecté a tous ls autres.
- **Topologie hiérarchique** : C'est une topologie ou le l'information prends un seul chemin pour accéder à chaque poste, comme un système pyramidal.

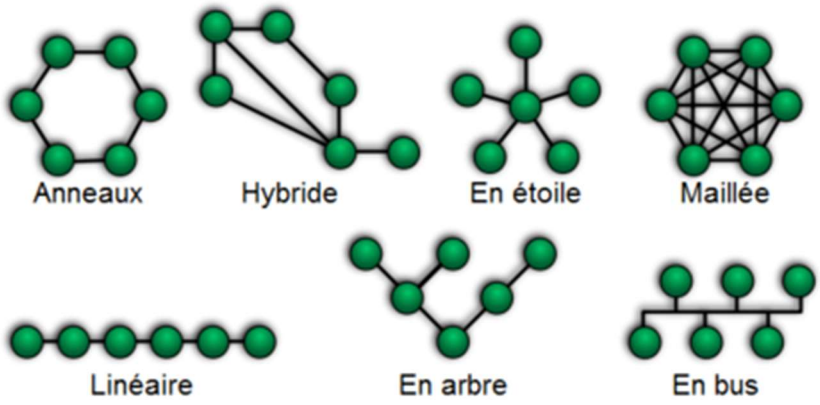


Figure 1.3. Les topologies physiques.

1.4.2 Les topologies logiques :

Elles représentent la manière dont les données transitent sur le support de communication, les topologies les plus courantes sont :

- **Ethernet** : La technologie Ethernet basée sur la norme 802.3 se décline dans de nombreuses variantes tels que :
  - Deux topologies différentes qui sont bus et étoile
  - Multi supports permettant d'être capable de faire usage de câbles coaxiaux, fils en cuivre à paires torsadées ou de fibres optiques.

- **Le Token ring :** Token Ring normalisé comme 802.5 par l'IEEE. Un réseau en anneau à jeton est un réseau local (LAN, Local Area Network) dans lequel tous les ordinateurs sont connectés selon une topologie en anneau (ring) ou en étoile et transmettent un ou plusieurs jetons logiques (token) d'hôte en hôte.

- **FDDI (Fiber Distribuated Data Interface)** : C'est une technologie de MAN et LAN, ISO 9314, c'est une technologie qui gère un anneau de fibre optique, multimode et monomode, pour un débit nominal de 100 Mbps pour 100 km ce qui énorme.

1.5 Modèle TCP/IP – OSI :

Un protocole est l'ensemble des règles qui régissent une communication, en réseau nous avons le modèle **OSI** et **TCP/IP** [2] :

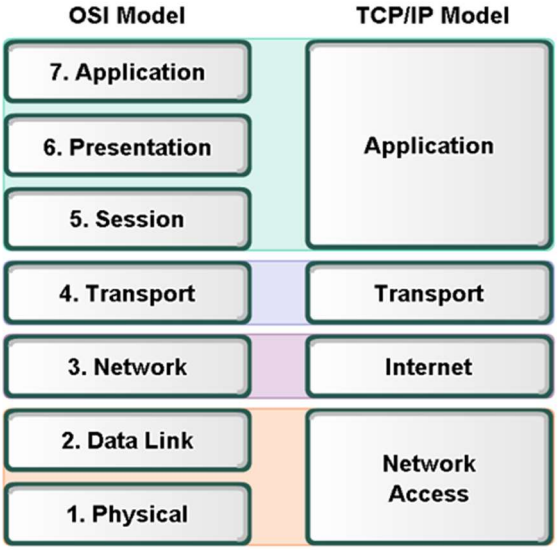


Figure 1.4. Protocoles OSI et TCP/IP.

Les couches sont les suivantes :

- Application (affichage du message sur l'interface utilisateur.
- Présentation (formatage du message).
- Session (ouverture d'une connexion).
- Transport (sélection du mode d'envoi du message).
- Réseau (assurance de la connexion au réseau et Adressage.
- Liaison de données (Adressage Mac, conversion du signal électrique ou lumineux en information traduisible par carte réseau des périphériques réseau.
- Physique décrit le mode de circulation du signal dans le support.

1.5.1 Le modèles TCP/IP :

Il est plus récent que le modèle ISO il simplifie beaucoup certains concepts du premier modèle en unifiant certaines couches, pour avoir en résultant un modèle comme ci-dessous :

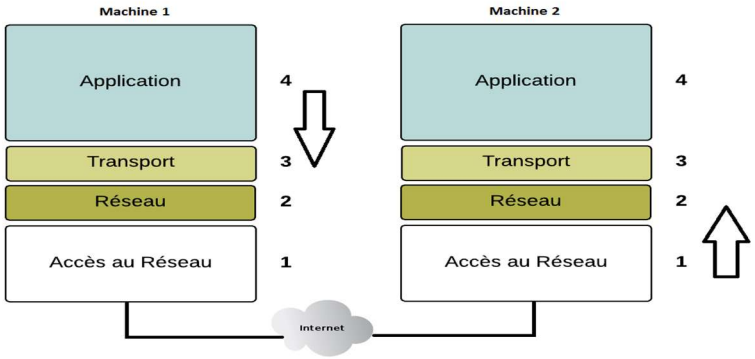


Figure 1.5. Modèle TCP/IP.

Nous allons ainsi parcourir les différentes couches de ce modèle, car ces notions nous serons utile dans l'étude des en-têtes pour pouvoir obtenir les signatures de torrent ou des fichiers liés.

a *La couche accès réseau :*

La couche d'accès au réseau du modèle TCP/IP est associée à la couche physique (couche 1) et à la couche liaison de données (couche 2) du modèle OSI.
La couche physique OSI est responsable de la conversion de la trame en un flux de bits adapté au support de transmission. La couche physique OSI gère et synchronise les signaux pour la transmission réelle, il est à son niveau d'adressage :

- **Adresse MAC** : Une adresse MAC Ethernet est une valeur binaire de 48 bits constituée de 12 chiffres hexadécimaux.
- **Protocol ARP** : c'est ce protocole qui permet de faire la correspondance entre les données de la couche réseau et l'adresse MAC de la couche accès réseau.

b ***La couche internet :***

Nous avons ici la notion d'adressage IP, nous avons différentes adresses qui permettent d'identifier le communiquant à travers les différents niveaux de l'encapsulation, ici nous voyons l'adresse IP (internet protocole), qui possède deux

versions, qui permet d'identifier les réseaux et les machines, il nous permet enfin de nous introduire à la notion d'internet (communication inter-réseau).

Il a pour son utilisation différents protocole.

- **Le protocole IP :** Ce protocole est celui qui gère notre adressage, il a actuellement deux versions de cet adressage (**IPV.x) :**
  - **IPv4 :** Chaque adresse est une chaine de 32 bits divisée en quatre parties appelées octets.

    Chaque octet contient 8 bits séparés par un point. Pour simplifier leur utilisation, les adresses IPv4 sont souvent exprimées en notation décimale à point, par exemple, l'adresse IPv4 d'un PC est : 192.168.10.10. Il y a deux types d'adresse IPv4, adresse IPv4 publique et privée.

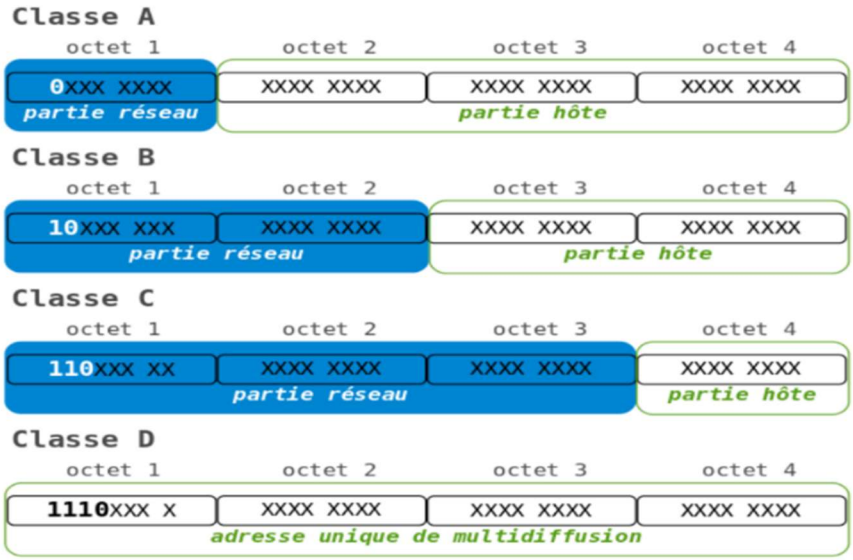


Figure 1.6. Classe d'adresse.

- **Classe A :** Le premier octet a une valeur comprise entre 1 et 126 ; soit un bit de poids fort égal à 0.

  L'adresse réseau 127.0.0.0 est réservée pour les communications en boucle locale.
- **Classe B :** Le premier octet a une valeur comprise entre 128 et 191 ; soit 2 bits de poids fort égaux à 10.
- **Classe C :** Le premier octet a une valeur comprise entre 192 et 223 ; soit 3 bits de poids fort égaux à 110.

- **Classe D :** Il s'agit d'une zone d'adresses dédiées aux services de multidiffusion vers des groupes d'hôtes (host groups).
- **Classe E :** Il s'agit d'une zone d'adresses réservées aux expérimentations.
- **Les masques de sous réseaux :** Ils sont de la forme 255.225.225.255.

  Ils permettent de détecter les sous réseaux car une fois converti en binaire il permet de différencier la partie réseau et la partie machine d'une adresse, par rapport aux classes d'adresse chaque adresse à son masque par défaut [3] :

Classe	Masque réseau	Adresses réseau	Nombre de réseaux	Nombre d'hôtes par réseau
A	255.0.0.0	1.0.0.0 - 126.255.255.255	126	16777214
B	255.255.0.0	128.0.0.0 - 191.255.255.255	16384	65534
C	255.255.255.0	192.0.0.0 - 223.255.255.255	2097152	254
D	240.0.0.0	224.0.0.0 - 239.255.255.255	adresses uniques	adresses uniques
E	non défini	240.0.0.0 - 255.255.255.255	adresses uniques	adresses uniques

Figure 1.7. Masque de sous réseaux.

c NAT (Network Adresse translation) :

Un système qui permet de faire le mappage entre notre adresse privé et notre adresse publique, qui fait une interface entre les deux donc une mesure de sécurité, nos machines ne sont pas à découvert, il permet donc d'optimiser l'économie des adresses publique [3].

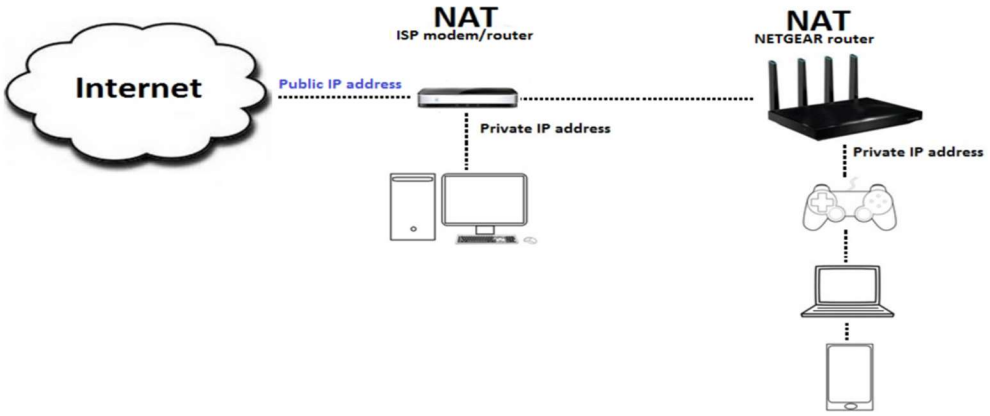


Figure 1.8. Réseau NAT.

Les routeurs assurent la fonction NAT, ils peuvent être configures avec plusieurs adresses IPv4 publiques (pool NAT) et les serveurs proxy assurent aussi la fonction NAT et ils sont utilisés dans les réseaux des grandes entreprises ou campus. La NAT qualifie également les adresses locales ou globales :

- **Adresse locale :** L'adresse locale fait référence à toute adresse qui apparait sur la partie interne du réseau.

- **Adresse globale :** L'adresse globale fait référence à toute adresse qui apparait sur la partie externe du réseau.

d Fonctionnement de la NAT :

Quand un poste émet des requêtes à travers le réseau, elles arrivent au niveau de la passerelle, la translation se passe, nous avons deux types de NAT :

- **NAT Statique :** il est plus utilisé pour les serveurs car il permet d'associer à une adresse privée une adresse publique unique et vice versa.

- **NAT dynamique :** il est utilisé dans le cas ou plusieurs périphériques d'un côté de l'interface ont accès à internet avec une seule adresse IP publique comme expliqué plus haut.

  Afin de pouvoir « multiplexer » (partager) les différentes adresses IP sur une ou plusieurs adresses IP routables le NAT dynamique utilise le mécanisme de translation de port (PAT - Port Address Translation), c'est-à-dire l'affectation d'un port source différent à chaque requête de telle manière à pouvoir maintenir une correspondance entre les requêtes provenant du réseau interne et les réponses des machines sur Internet, toutes adressées à l'adresse IP du routeur [3].

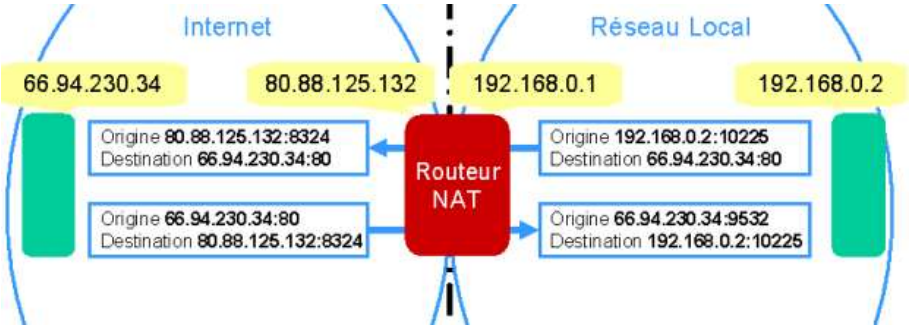


Figure 1.9. Translation.

e ***Couche transport :***

C'est dans cette partie que le choix du mode de transfert est choisi, il choisit parmi ses deux protocoles s'il y'aura une connexion ou pas en effet il possède deux protocoles, un connecté et un autre sans connexion (TCP & UDP).

- **TCP (Transmission control protocol) :** C'est un protocole orienté connexion.
  Ce protocole négocie alors par le (three way handshake) une connexion afin d'établir un support fiable, mais il est utilisé pour les applications peu volumineuse.
- **UDP (User Datagram Protocol) :** le protocole UDP est un protocole sans connexion, ce qui signifie que les segments sont en flux continue et ne font pas sujet de vérification rigoureuse, ce qui n'est pas en soit un avantage mais plutôt une différence car cela lui confère une beaucoup plus grande rapidité par rapport au TCP.

1.6 Les proxys :

1.6.1 Fonction :

En matière de sécurité il permet à ce que le réseau tout entier ne soit pas exposé seulement l'adresse du proxy est vu à l'extérieur.

Le proxy permet aussi la navigation sur l'internet et sur le réseau en général, avec une de ses fonctionnalités très puissantes qui est le cache.

L'intérêt majeur d'un serveur proxy est dans le cadre de la sécurité informatique, en filtrant l'ensemble des connexions de l'entreprise à internet en analysant les requêtes clients et les réponses des serveurs en tenant compte de certaines critères (liste des adresses blanches, Liste des adresses noires, mots-clés, protocoles ...).

Enfin, on peut utiliser un proxy pour authentifier les utilisateurs, afin de limiter l'accès au réseau internet, on donne l'accès aux ressources externes seulement aux personnes autorisées à le faire et l'enregistrer dans les fichiers journaux des accès identifies, ils se sentent suivis et restants sages dans leurs recherches [4].

1.7 La sécurité des réseaux :

1.7.1 Vulnérabilité :

Toutes les entreprises craignent les attaques réseaux, celles-ci peuvent prendre différentes formes et se manifestent sur toutes les couches, avec des risques différents :

- **Malware :** Un logiciel malveillant est un terme générique englobant ces différentes menaces informatiques visant toutes à nuire à un appareil connecté. Un logiciel malveillant peut corrompre, effacer ou voler les données des appareils et réseaux d'une entreprise [5].
- **Virus informatique :** C'est un type de logiciel malveillant caché dans un logiciel légitime il se réplique à grande vitesse [5].
- **Spyware :** Les logiciels espions ou chevaux de Troie, ceux-ci infectent silencieusement l'ordinateur grâce à une application en apparence légitime [5].
- **Ingénierie sociale :** ou le pirate dérobe les accès en subtilisant les informations avec le détenteur [5].

1.7.2 Attaques :

Les réseaux d'entreprises font souvent l'effet d'attaques réseaux comme :

- **DDOS (distribuated denial of service) :** Cette attaque est l'inondation d'un serveur ou plusieurs serveurs d'entreprises de requêtes dans le but de le rendre indisponible [5].
- **Hameçonnage :** ou l'attaquant dérobe des informations grâce à un faux site ou faux mail.

1.7.3 La sécurité :

Face à la variété de menace existante la sécurité de l'infrastructure doit être prise avec le plus grand sérieux avec des mesures comme :

- **Vlan :** Nous créons des sous-réseaux pour séparer les utilisateurs pour renforcer la confidentialité de certains documents.

- **Proxy :** comme vu plus haut aide pour la sécurité.

- **Pare-feu** : Un pare-feu est un appareil de protection du réseau qui surveille le trafic entrant et sortant et décide d'autoriser ou de bloquer une partie de ce trafic en fonction d'un ensemble de règles de sécurité prédéfinies.

  Les pares-feux constituent la première ligne de défense des réseaux depuis plus de 25 ans.
  Ils établissent une barrière entre les réseaux internes sécurisés et contrôlés qui sont dignes de confiance et les réseaux externes non fiables tels qu'internet.
  Un pare-feu peut être un appareil physique, un logiciel ou les deux [5].

- **NIDS (network intrusion detection system)** : Les IDS, ou systèmes de détection d'intrusions, sont des systèmes software ou hardware conçus afin de pouvoir automatiser la surveillance d'événements survenant dans un réseau ou sur une machine particulière, et de pouvoir signaler à l'administrateur système, toute trace d'activité anormale sur ce dernier ou sur la machine surveillée. L'IDS est un système de détection passif [6].
  Il permet aussi de repérer les activités anormales ou suspecte sur la cible analysée.

1.7.4 Types des IDS :

Les IDS sont devisés en plusieurs types mais on peut les regrouper en 3 principales familles :

- **IDS Host HIDS** : les HIDS analysent le fonctionnement et l'état sur lesquelles ils sont installés, l'intégrité des systèmes et alors vérifie périodiquement et des alertes peuvent être déclenché.

- **IDS réseau NIDS** : les NIDS sont dédiés aux réseaux, ils analysent et interprètent les paquets circulant sur le réseau. Ils sont implémentés de la façon suivante : les captures sont placées aux endroits stratégiques du réseau et génèrent des

alertes s'ils détectent une tentative d'intrusion. Ces alertes sont envoyées à une console sécurise qui les analyse et traitent.

- **Les IDS Hybride** : 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 placées en des pointes stratégiques, et agissent comme NIDS et/ou HIDS suivant leur emplacement.

1.7.5 Fonction de l'IDS

Le travail d'un IDS est divisé en quatre fonctions :

- **Action** : alerte l'administrateur lorsqu'une tentative d'intrusion est détectée.
- **Gestion** : les IDS doivent être administrés d'une façon permanente.
- **L'analyse** : il y a deux méthodes d'analyse, l'un est basé sur les signatures d'attaques, et l'autre sur la détection d'anomalie.
- **Journalisation** : enregistrement des évènements dans un fichier log exemple des évènements tentative de connexion.

1.7.6 Modes de détection :

La détection est devisée en deux mode :

- **La reconnaissance des signatures :** Cette approche consiste à rechercher dans l'activité de l'élément surveille les empreintes (signature) d'attaque connues.

  Ce type d'IDS nécessite des mises à jour fréquentes, Une signature permet de définir les caractéristiques d'une attaque au niveau des paquets ou niveau protocole.

- **La détection d'anomalie :** Elle consiste à détecter des anomalies par rapport à un profil du trafic habituel, la mise en œuvre comprendre une phase d'apprentissage au cours de laquelle les IDS découvrent le fonctionnement normal des éléments surveilles.

1.8 Conclusion :

Dans ce chapitre nous avons introduit et survolé la plupart des concepts que nous aurons à utiliser dans la suite de ce document, du hardware au software, nous avons en mains les éléments essentiels pour réaliser le travail de sécurité de la manière la plus décente qui soit.

Tous d'abord nous voulons en premier lieu dans le chapitre numéro 2 d'expliquer la notion de partage des fichiers sur internet par une synthèse des différents protocoles et systèmes ainsi que l'aspect sécurité.

Chapitre 2 Partage de fichier en Peer-to-Peer

2.1 Introduction :

De nos jours nous avons une vague de données transitant sur internet, ainsi qu'une multitude de moyen de les acquérir, parmi ces méthodes celle qui attire notre attention est le mode d'acquisition basé sur le point à point (P2P), mode de transfert non centralisé qui nous permet de mettre en place des protocoles et mode de téléchargement comme le torrent.

2.2 Le numérique :

Une information numérique (en anglais digital) est une information quantifiée et échantillonnée, une donnée informatique.

Le format des données est la manière utilisée en informatique pour représenter des données sous forme de nombres binaires (des 0 et des 1).

C'est une convention utilisée pour représenter des informations correspondant à un texte, une page, une image, un son, un fichier exécutable, etc.

2.2.1 Les fichiers numériques :

Ceux-ci sont dits ouverts (dont on connaît les spécifications) ou fermés (dont les spécifications sont secrètes et protégées).

Les fichiers fermés ou propriétaires ne sont souvent utilisables qu'avec un logiciel spécifique.

Ils existent des fichiers texte, son, image, vidéo. Ceux-ci ont une taille de donnée qui se mesure en octets (o), c'est l'unité de mesure.

Ko = kilo-octet, Mo = méga-octet, Go = giga-octet.

Très vite les documents numériques atteignent une taille correspondante à des nombres importants. On estime en 2011 que l'univers numérique mesure 1,8 zettaoctet soit encore 1.800 milliards de Go. [7]

2.3 Partage de fichier :

Le partage des fichiers est une technique de transfert de fichier consistante à distribuer ou à donner accès, à distance, à des données numériques à travers un réseau informatique.

Il peut s'agir de fichiers de toutes sortes : logiciels, livres, vidéo, audio etc.

Deux techniques de partage de fichiers existent actuellement :

- l'hébergement centralisé (modèle client-serveur) permet de stocker les données sur un serveur de fichiers unique et d'y accéder sur celui-ci depuis un autre ordinateur (dit le client).
- la technique pair-à-pair qui consiste à mettre des données en partage suivant un modèle de réseau informatique où chaque ordinateur client est aussi un serveur.

2.4 Le Peer to Peer :

Les systèmes pairs à pair (P2P, de l'anglais "Peer-to-Peer") sont composés d'un ensemble d'entités partageant un ensemble de ressources, et jouant à la fois le rôle de serveur et de client. Chaque nœud peut ainsi télécharger des ressources à partir d'un autre nœud, tout en fournissant des ressources à un troisième nœud. Immense essor de tels systèmes, phénomène de société avec d'importants impacts en termes commerciaux (droits, taxes) et moraux (contenu des données échangées).

Permet une utilisation maximale de la puissance du réseau, une élimination des couts d'infrastructure, et une exploitation du fort potentiel inactif en bordure de l'Internet. Retient l'attention de la recherche, des développeurs et des investisseurs.

Les pairs du réseau peuvent être de nature hétérogène : PC, PDA, Téléphone portable. Nature dynamique où chaque pair peut apparaître et disparaître à tout moment [8].

2.5 Peer-to-Peer vs Client/serveur :

2.5.1 Architecture client/serveur :

L'architecture client/serveur (Figure 2.1) c'est la description du fonctionnement coopératif entre le serveur et le client ainsi que les services internet sont conçus selon cette architecture, chaque application est composée de logiciel serveur et logiciel client.

Un logiciel serveur, peut correspondre à plusieurs logiciels clients développés dans différents environnements : Linux, Mac, Windows, la seule obligation est le respect du protocole entre les deux processus communicants, ce protocole étant décrit dans un RFC [9].

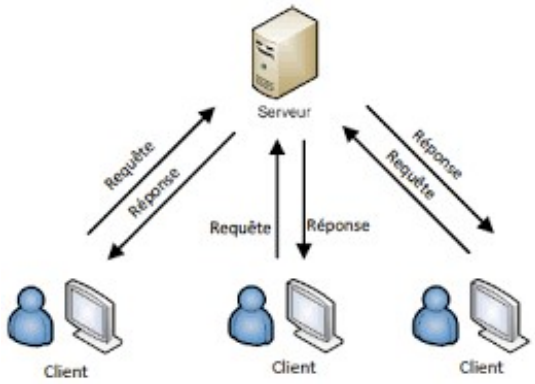


Figure 2.1. Architecture client/serveur.

a Un serveur :

Un serveur est un programme qui offre un service sur le réseau, il accepte des requêtes, les traite et renvoie le résultat au demandeur.

Le terme serveur s'applique à la machine sur lequel s'exécute le logiciel serveur et pour que ce dernier peut offrir ces services en permanence, le serveur doit être sur un site avec accès permanent et il doit être actif en permanence.

b Un client :

Un logiciel client est un programme qui utilise le service offert par un serveur, ce dernier envoie une requête et reçoit la réponse, il peut être raccordé par une liaison temporaire.

2.6 Caractéristiques du modèle P2P :

Comme les ressources ont une connectivité instable ou des adresses IP variables, elles fonctionnent de manière autonome, indépendamment de systèmes centraux comme les DNS.

Ce qui a rendu Napster et des systèmes similaires populaires, c'est le fait de tirer parti des ressources qui étaient auparavant inutilisées en tolérant une connectivité aléatoire [10].

Un vrai système peer-to-peer se reconnaît donc par les caractéristiques suivantes :

2.6.1 Hétérogénéité :

Vu que l'autonomie de nœuds possédant des architectures matérielles et/ou logicielles hétérogènes, les systèmes P2P doivent posséder des techniques convenables pour résoudre les problèmes liés à l'hétérogénéité de ressources.

2.6.2 L'auto-organisation :

Les systèmes P2P sont souvent déployés sur internet, la participation d'un nouveau nœud à un système P2P ne nécessite pas une infrastructure couteuse, Il suffit d'avoir un point d'accès à l'Internet et de connaître un autre nœud déjà connecté pour se connecter au système.

Un système P2P doit être un environnement ouvert c'est-à-dire, un utilisateur sur un nœud doit être capable de connecter son nœud au système sans avoir besoin de contacter une personne et sans avoir besoin de passer par une autorité centrale.

2.6.3 Décentralisation :

Le fait que chaque nœud gère ses propres ressources permet d'éviter la centralisation de contrôle. Un système P2P peut fonctionner sans avoir aucun besoin d'une administration centralisée ce qui permet d'éviter les goulets d'étranglements et d'augmenter la résistance du système face aux pannes et aux défaillances.

2.6.4 Passage à l'échelle :

Il s'agit de faire coopérer un grand nombre de nœuds (des milliers ou des millions) pour partager leurs ressources tout en maintenant une bonne performance du système.

Cela signifie qu'un système P2P doit offrir des méthodes bien adaptées avec un environnement dans lequel il y a un grand volume de données à partager, un nombre important de messages à échanger entre un grand nombre de nœuds partageant leurs ressources via un réseau largement distribué.

2.6.5 Dynamisme :

Les systèmes Peer-to-Peer supportent le dynamisme, c'est-à-dire les pairs, peuvent joindre ou quitter le système de manière continue, sans qu'ils affectent le fonctionnement de ce dernier.

2.7 Architectures du P2P :

Généralement les réseaux Peer to Peer sont classés en 2 grands modèles, les modèles structurés et les modèles non structurés.

Cependant, dans les modèles non structurés, on distingue trois architectures, une centralisée, une décentralisée et l'autre hybride un mélange entre les deux modèles.

2.7.1 Modèles non structurés :

Ils reposent sur une construction aléatoire du graphe de connexion, un nœud joint le réseau par l'intermédiaire d'un autre nœud déjà connecté. La recherche des services dans un tel réseau se fait généralement selon la technique d'inondation [11].

Parmi ces réseaux, on trouve trois types d'architectures :

a P2P centralisés (Napster) :

Ce type d'architecture repose sur un serveur central auquel se connectent les utilisateurs, l'utilisateur recherche le fichier désiré sur le serveur qui lui indique la liste des postes sur lesquels il est susceptible de le trouver, puis il se connecte directement

à l'un de ces postes pour télécharger le fichier. A aucun moment le fichier ne passe par le serveur central.

Ce dernier peut donner des indications sur les postes connectés tel que : le débit et le temps de chargement [12].

Du fait de la présence d'un serveur, certains considèrent que le modèle n'est pas entièrement P2P.

Cette architecture est utilisée par le logiciel Napster (Voir la Figure 2.2).

Le principe de fonctionnement de Napster est comme suit :

- Un utilisateur recherche un fichier ressource en envoyant une requête au serveur.

- Le serveur central répond et transmet des informations relatives aux ordinateurs où ces fichiers résident, telles que le nom d'utilisateur, l'adresse IP, etc. – L'utilisateur télécharge le fichier directement à partir de l'un des ordinateurs renseignés par le serveur. Le serveur n'est plus impliqué dans le transfert de fichier [13].

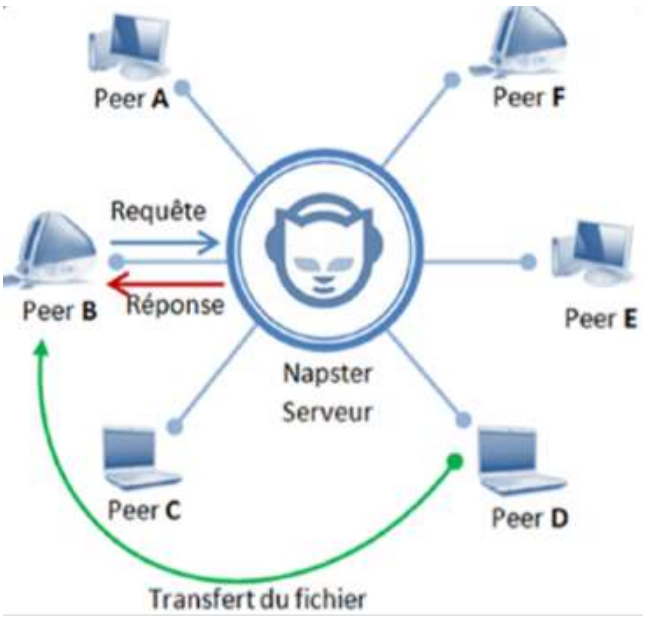


Figure 2.2. Un réseau Napster.

b P2P décentralisés (Gnutella) :

L'architecture centralisée pose des problèmes de sécurité, robustesse, et de limitation de la bande passante.

Les problèmes sont directement issus de l'utilisation de serveurs, dont le seul but est de posséder l'annuaire des clients.

Si on désire supprimer les serveurs centraux, il faut donc trouver le moyen de constituer un annuaire sur chaque client, puis de les faire communiquer [14].

C'est sur ces mécanismes sont basés les réseaux P2P décentralisés.

Il n'y a donc plus de serveurs centraux, chaque élément agit comme un serveur et un client (nommé servent) c'est pour cela qu'on appelle ce type de réseaux P2P pur. Parmi les logiciels qui utilisent cette architecture on trouve Gnutella. (Voir la figure 2.3).

Le principe de fonctionnement de Gnutella est comme suit :

- Le Peer A envoie une requête à ses voisins, qui à leurs tours envoient la même requête à leurs voisins.
- Les Peers qui détiennent la requête demandée vont répondre et transmettre des informations relatives aux ordinateurs où ces fichiers résident, en suivant le chemin inverse.
- fichier recherché est localisé, il suffit de le transmettre via une connexion HTTP.

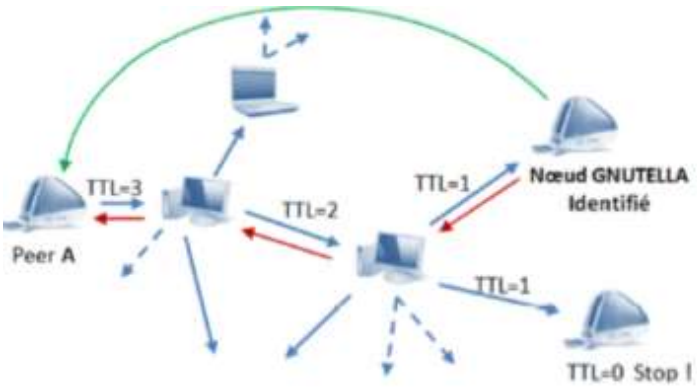


Figure 2.3. Un réseau Gnutella.

c P2P hybrides (BitTorrent) :

L'architecture hybride est un couplage des deux architectures P2P centralisé et P2P décentralisé. Les serveurs dans ce modèle (appelés aussi super-peers) sont connectés entre eux suivant le mode décentralisé, et chacun d'eux est connecté aux éléments de son groupe selon le mode centralisé et référence ainsi les contenus de son groupe [15].

Le BitTorrent est un protocole de la couche applicative qui utilise le système de peer-to-peer pour faire le partage, il est aussi le protocole utilisant le P2P sur lequel notre étude se focalise. Ces éléments principaux sont les suivants :

•**Peers :** c'est un client participant au partage du fichier, n'ayant pas en général le fichier complet

- **Leech :** un client qui télécharge sans partager en contrepartie
- **Seed :** c'est le fait qu'un client partage un fichier complet
- **Swarm** : traduction littéral essaim**,** c'est l'ensemble des peers (seeder et leecher) qui participe au partage du torrent.

• **Torrent :** il fait référence au métafile « **.torrent »** le fichier dont il faut faire l'acquisition pour avoir les informations nécessaires au téléchargement du fichier.

• **Tracker :** un Tracker est un serveur constamment mis à jour par les clients, sur l'état du fichier, il permet aussi aux clients de récupérer les informations sur d'autres clients auxquels ils peuvent se connectés.

Le BitTorrent est un protocole très utilisé pour, plus de 150 millions d'utilisateurs pour son client le plus populaire **Utorrent,** suivi de nombreux autres clients très utilisés [16].
Ces différents éléments que nous venons juste de voir participent à l'architecture suivante :

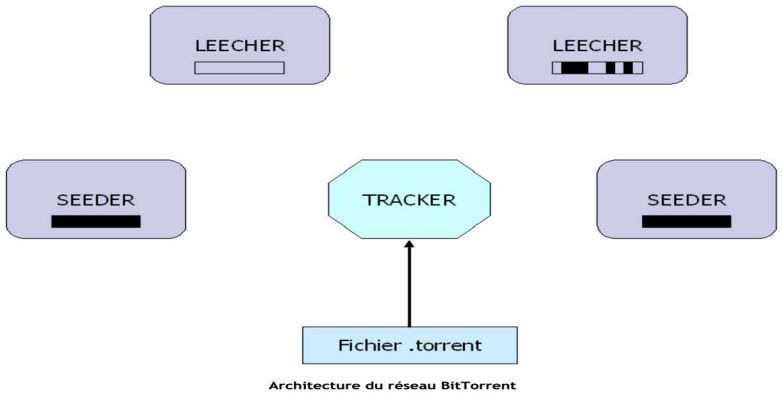


Figure 2.4. Un réseau BitTorrent.

Le fichier torrent est localisé dans un ou plusieurs serveurs, et quand un Peer connecté télécharge le fichier, il le récupère sur un serveur ou sur un client connecté et en partage d'où le concept d'hybridité.
Le torrent est une architecture ou chaque fichier « .torrent » est un mini réseau.
Pour effectuer un téléchargement torrent, le client doit avoir le metafile « .torrent », qu'il peut se procurer sur un site torrent **www.torrent9.io** par exemple.

Ce fichier contient les informations sur le torrent notamment sur la localisation des trackers qui coordonnent le téléchargement.

Ensuite débute le téléchargement à partir d'un client BitTorrent, et le client devient alors un Leecher, quand le téléchargement est en cours un leecher peut devenir un seeder, à mesure qu'il upload le fichier.

•Trackerless (sans tracker) : d'après ce que nous venons de citer il est assez aisé de comprendre que les trackers sont indispensables dans l'utilisation de torrent.

Dans un cas de figure ou le torrent ne possède pas ces serveurs là nous aurons les DHT (distributed hash table) si ce mode est actif tous les clients reçoivent la liste de tous les peers du swarm.
En pratique le DHT est un ensemble de requête-réponses qui se concrétise en :

- **Ping :** pour savoir si le peer est toujours disponible.
- **Find_node :** pour trouver les informations permettant de contacter un peer.
- **Get_peers :** demande une liste des peers qui possède une partie du fichier.
- **Announce_peer :** informe le réseau des informations de contact d'un peer.

2.7.2 Modèles structurés :

Les réseaux structurés imposent des contraintes sur les endroits où sont hébergés les contenus. Ils utilisent des tables de hachage distribuées (DHTs) pour gérer les opérations de recherche. Une entrée dans la table contient la paire (clé, valeur), où la clé est l'identifiant associé à un objet partagé et la valeur correspond à l'information de localisation d'objet recherché. P4L , Cycloid, Chord , CAN et EZSearch sont des exemples des réseaux structurés.
Ils sont basés sur le concept de table de hachage distribuée (DHT).
Avec l'approche DHT, chaque nom d'entité dans le système peut être mis en cohérence sur le même espace de recherche (identificateur), en utilisant une fonction de hachage telle que SHA-1 ou SHA-2. Cependant, toutes les entités dans le système ont une vue consistante de cette correspondance.
Étant donné cette vue consistante, plusieurs structures d'espace de recherche sont définies pour localiser ces entités.

A titre d'exemple, dans Chord, l'espace de recherche est basé sur une topologie en anneau.

Dans P4L, il est structuré sous forme d'anneaux hiérarchiques [17].

2.8 Comparaison des infrastructures client/serveur et P2P :

Traditionnellement l'échange de services entre ordinateurs est fondé sur la technique client/serveur, selon cette architecture il n'y a qu'une seule entité centrale très puissante qui est le serveur et plusieurs autres entités généralement de puissances inférieures qui sont les clients.

Le serveur est le seul fournisseur des services aux clients, un client consomme les services exécutés par le serveur, sans partager aucune de ses propres ressources. L'architecture pair à pair se pose comme une solution de rechange à l'architecture client/serveur en offrant plusieurs avantages par rapport aux autres basés sur le paradigme client/serveur, le tableau 1.1 montre bien la différence entre les deux modèles.

Critère	Modèle Client/serveur	Modèle P2P
Gestion	Supervisé	Auto-Organisé
Présence	Permanente	Ad hoc
Accès aux ressources	Recherche	Découverte
Organisation	Hiérarchique	Distribuée
Mobilité	Statique	Mobile
Disponibilité	Dépendante du serveur	Indépendante des pairs
Nommage	DNS	Indépendant

Tableau 2.1. Comparaison des infrastructures client/serveur et P2P.

2.9 Domaine d'utilisation du P2P :

2.9.1 Streaming P2P :

Le streaming P2P est le fait de regarder en direct des flux produits et/ou relayes par d'autres pairs du réseau afin d'éviter ou du moins diminuer la congestion qui pourrait se produire sur les serveurs de téléchargement, tout se déroule entre les personnes qui veulent accéder au fichier, Swarmplayer, qui permet de lire des vidéos en streaming en utilisant Bittorrent, s'annonce comme une vraie révolution dans le domaine.

2.9.2 Plateformes de développement :

La plupart des logiciels P2P ont été développés de manière spécifique sans référence à des standards propres au p2p, dans le but d'uniformiser les réseaux P2P, des plates-formes sont implémentées pour servir de base au développement des applications P2P, Elles assurent les fonctionnalités de base : gestion des pairs, attribution des identifiants, découverte des ressources, communication entre pairs, sécurité. On peut citer la plate-forme JXTA développé par Sun Microsystems.

2.9.3 Système de sauvegarde distribué :

Le système de sauvegarde réparti s'appuie sur la coopération des pairs à mettre à disposition leurs espaces de disques inutilisés, un utilisateur peut sauvegarder d'une manière transparente et sécurisée une copie de ses données dans les autres pairs du réseau P2P afin de les récupérer en cas de perte ou de dégâts occasionnés aux données locales. Nous pouvons citer des projets comme Wuala, DisPairSe, OceanStore...

2.9.4 Le calcul distribué "Grid Computing" :

Consiste à utiliser les machines connectées à l'internet pour faire des petites portions d'un grand calcul, en exploitant les ressources (CPU, mémoire…) inutilisées des PC en réseau en vue d'accroitre le potentiel réseau, comme exemple le projet : SETI@home (Search for Extra Terrestrial Intelligence).

2.9.5 Des programmes de messagerie :

De nos jours, il existe des services de messagerie électroniques basés sur le principe P2P, les utilisateurs peuvent envoyer et recevoir des e-mails d'une façon sécurisée, pas besoin d'un serveur central pour stocker temporairement les messages ce qui assure la confidentialité des correspondants, un système d'authentification et de cryptage est utilisé afin de protéger leur contenus, un bon exemple est Jeftel.com. Des logiciels de messagerie instantanée peuvent aussi être vus comme P2P, on citera des exemples comme ICQ, AIM qui utilisent un serveur mais juste pour la résolution d'adresses (les adresses utilisées peuvent être des alternatives aux adresses IP).

2.9.6 Partage de fichiers :

Au début le P2P est fondé sur le principe d'échange de fichiers musicaux MP3 entre un groupe d'utilisateurs c'est le cas de Napster (il a enregistré 37 millions d'utilisateurs avec plus de 1.5 millions de téléchargements journaliers au sommet de sa gloire en août 2000), ensuite de nombreux logiciels ont vu le jour et servent à l'échange direct des documents de différentes natures (texte, multimédia, image. . .) entre les utilisateurs du réseau comme Gnutella, freenet... [18]. Actuellement le monde du partage de fichier en P2P rappel le BitTorrent avant toute chose, ces différents clients ont surpassés tout autres logiciels ou systèmes connu, il existe actuellement une multitude de sites web dédiés à ce genre de partage.

2.10 Cryptage des données :

Le cryptage des données dans le Peer-to-Peer et dans le torrent plus précisément se base généralement sur le protocole SSL et TLS.

2.10.1 TLS et SSL :

TLS/SSL sont des protocoles de sécurisation des échanges sur internet.
Ils sont utilisés pour apporter plusieurs fonctions de sécurité lors de l'échange de données.

SSL/TLS fonctionnent suivant un mode client-serveur avec l'interdiction d'une nouvelle couche de communication entre celle du transport et celle d'application du modèle TCP/IP dédié à la sécurité.

Sans SSL/TLS, les informations sont envoyées en clair lors d'un échange entre un client et un serveur.

Donc si quelqu'un se connecte au réseau, il lui serait facile d'intercepter les données échangées.

Il permet aussi une authentification et de s'assurer que l'ordinateur avec lequel on communique est bien celui qu'on prétend.

Figure 2.5. Fonctionement de TLS/SSL.

2.10.2 Certificat SSL :

L'établissement d'une connexion SSL nécessite l'installation d'un certificat numérique sur le serveur Web.

Ce certificat utilise alors les clés publiques et privées pour le cryptage, et identifie le serveur de manière unique et définitive. Les certificats numériques s'apparentent à une forme de carte d'identité électronique qui permet au client d'authentifier le serveur avant l'établissement d'un canal de communication crypté [19].

Un certificat SSL est délivré par une tierce partie de confiance, appelée Autorité de Certification (Certification Authority, ou CA).

2.10.3 Protocoles de TLS/SSL :

a SSL Handshake :

Ce protocole permet l'échange les paramètres de sécurité et aussi l'authentification des deux communicateurs.

Ce protocole fait intervenir les échanges suivants entre le client et le serveur :

- **1-** Le client envoie un message "Client bonjour" au serveur, ainsi que la valeur aléatoire du client et les suites de chiffrement prises en charge.
- **2-** Le serveur répond en envoyant un message "Server hello" au client, avec la valeur aléatoire du serveur.
- **3-** Le serveur envoie un message « Certificat », qui contient en particulier sa clé publique au sein d'un certificat numérique.
- **4-** Après validation du certificat et vérification de la signature précédente, le client crée un secret pré-maître aléatoire et le crypte avec la clé publique extraite du certificat du serveur, ensuit il envoie le secret pré-maître crypté au serveur.
- **5-** Le serveur reçoit le secret pré-maître. Le serveur et le client génèrent chacun de son côté le secret maître et les clés de session (clé de hachage et clé écriture) en fonction du secret pré-maître.
- **6-** Le client signale l'adoption du secret maître et les clés de session avec un ''ChangeCipherSpec''. Le client envoie également le message " Client finished ".
- **7-** Le serveur reçoit "Change cipher spec", ensuit il envoie un message " Client finished " au client.
- **8-** Le client et le serveur peuvent maintenant échanger des données d'application sur le canal sécurisé qu'ils ont établi. Tous les messages envoyés du client au serveur et du serveur au client sont chiffrés à l'aide de la clé de session.

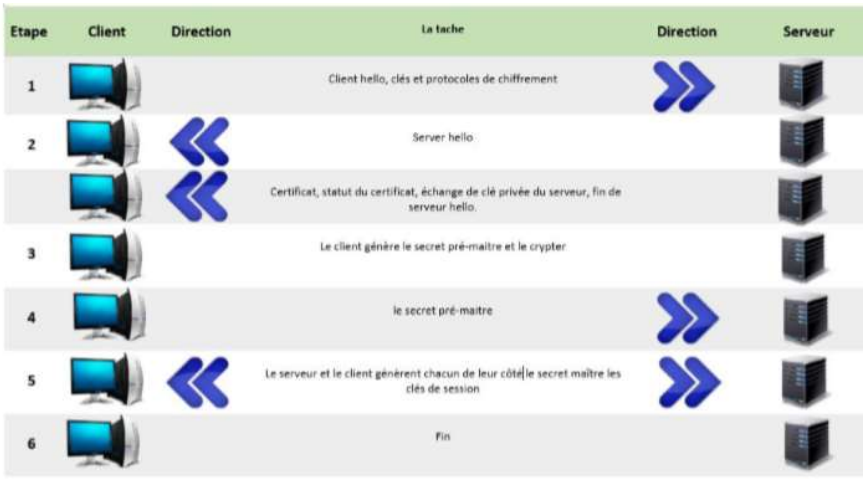


Figure 2.6. Etapes du handshake.

b SSL change Cliper Spec Protocol :

Le deuxième protocole « SSL change Cliper Spec Protocol » est utilisé pour indiquer un changement dans les algorithmes de chiffrement cryptographique, immédiatement après le changement toutes les données sont cryptées avec le nouveau chiffrement sélectionné.

c SSL Alert Protocol :

Le troisième protocole « SSL Alert Protocol » est responsable de la signalisation des problèmes dans la session SSL.

d SSL Record Layer Protocol

Le dernier protocole « SSL Record Layer Protocol » une fois négocier ce protocole chiffre toutes les informations échanger et effectuer divers contrôles.

2.11 Avantage et inconvénient du P2P :

Comme tous les systèmes de partage le P2P a ces avantages comme il a ces inconvénients et tous cela est résumé dans le tableau suivant (Tableau 2.2).

Avantage	Inconvénient
Echanges plus rapides.	QoS (ligne peu fiable, débit peu élevé).
Optimisation de l'utilisation de la bande passante du réseau.	Sécurité (Crackers, virus, attaques Dos, confidentialité, authentification).
Maintenance et coûts réduit.	Contenu trompeur.
Extensibilité (passage de 100 à 1000 nœuds sans problème).	Loi du Wild Wild Web (Droits d'auteurs, contenu immoral).
Résistance aux pannes.	Régulation / Répression.
Utilisation des ressources inutilisées.	

Tableau 2.2. Avantage et inconvénient du P2P.

2.12 Enjeux de l'utilisation du Torrent :

Les systèmes d'échanges de fichiers Torrent qui fait partie du réseau P2P peut facilement générer des problèmes, les membres de ce dernier sont parfois tentés d'avoir des comportements malveillants parce qu'il y'a l'aspect anonymat qui aide pour ça [20], parmi ces risques on trouve :

- **Les virus :** un virus de type cheval de Troie s'attaque aux réseaux de type P2P
  Il se propage lors de l'installation des logiciel client ou bien lors du téléchargement il s'infiltre avec les données télécharger sans que le client ne se rend compte.

- **Atteinte à la vie privée :** Une fois que les outils P2P sont utilisés notre vie privée n'est pas nécessairement protégé, les adresses IP des utilisateurs peuvent être récupérées lorsque le logiciel est centralisé.
  Lorsque celui-ci est décentralisé, les utilisateurs sont identifiables par leur fournisseur d'accès.

- **La pollution des réseaux :** Plusieurs sociétés indépendantes développent en effet de nouvelles technologies destinées à polluer les réseaux. Leurs clients potentiels sont les industries de la musique et du cinéma, qui cherchent par tous les moyens à détourner les adeptes des téléchargements gratuits au profit de systèmes sécurisés et payants.

Donc ils incorporent des fichiers de moindre qualité, incorrects ou qui ne durent que quelques secondes, afin de rendre ces réseaux gratuits et anarchiques moins attractifs.

- **La propagande :** La plupart des systèmes de P2P sont envahis par les bannières publicitaires même si a présent des version allégées sont proposées.
  Mais la propagande va plus loin l'installation de certain logiciel est accompagnée de l'installation de logiciels espions (spyware).

- **Le problème des droits d'auteurs :** Télécharger un fichier vidéo, image ou audio sous une licence propriétaire est un délit et plus encore, le distribuer à grande échelle pourrait être passible d'amendes, puisque peu d'applications de partage de fichiers ont été conçues dans le souci d'offrir une juste rémunération aux artistes.

2.13 Solution proposée :

Il existe plusieurs moyens de préventions pour éviter de faire face aux enjeux de l'utilisation du torrent dans une entreprise, des solutions hardware ainsi que software sont proposées.

Dans notre document nous allons nous baser sur la solution software pour détecter l'utilisation du torrent, à l'aide d'un IDS (Snort) nous allons mettre en œuvre des règles qui générerons des alertes une fois que l'utilisation du torrent est détectée au niveau de notre réseau.

En un premier temps pour arriver à concevoir ces règles il nous faudra des empreintes qui caractérisent le comportement du trafic du torrent par rapport au traffic du web normal, pour arriver à cela nous allons utiliser un analyseur de paquets qui est WireShark.

2.14 Conclusion :

Comme nous l'avons vu précédemment, les réseaux peer-to-peer forme un vaste et un grand domaine de recherche très actif et cela grâce à leur utilisation actuelle ainsi que leurs applications, ils sont répartis en trois grandes familles : les réseaux centralisés, les réseaux décentralisés et les réseaux hybrides, chacun de ces réseaux possède des caractéristiques bien à lui différente des autres, ils offrent beaucoup d'avantages indéniables tels que : la résistance aux pannes, la répartition de la charge mais aussi quelques inconvénients liés surtout dans l'aspect de sécurité tel que la saturation de la bande passante, triche aux ratios d'échange et risque d'attaques dénis de services (DOS) sans parler de l'aspect légal.

Pour ces raisons nous voulons mettre un contrôle rigoureux quant à la présence de torrent dans un réseau.

Dans le chapitre 3 nous allons essayer d'extraire le maximum des empreintes du torrent dans notre réseau, dans le but d'établir des règles de Snort afin de mieux le sécuriser.

Chapitre 3 Extraction des empreintes du torrent

3.1 Introduction :

La controverse sur la légalité et la sécurité du torrent nécessite une démarche méthodique et sûre pour détecter de manière infaillible son utilisation, en utilisant Snort nous allons ici faire en sorte d'établir un réseau de test afin de mettre en place une politique qui peut se voir implémenter pour une structure plus vaste, et ainsi servir de modèle de base.

3.2 La méthodologie de recherche :

Dans notre recherche, nous allons essayer de détecter comment le réseau torrent marque sa présence en soulignant les différences du trafic Torrent par rapport au trafic web normal, pour faire cela tout d'abord il nous faudra décrire le comportement normal du trafic web.

Une fois que la différence a été remarquée on pourra proposer des méthodes pour la détection du réseau torrent, afin de cerner notre problématique et de trouver des éléments de réponses à un certain nombre de questionnements posées au début de ce travail.

3.2.1 Les étapes suivies dans notre recherche :

Pour répondre à notre problématique, nous avons utilisé les méthodes et les techniques suivantes :

- **La technique documentaire** : qui nous a permis de définir les concepts théoriques sur le sujet de notre recherche.
- **Capture des données** : qui nous a permis de capturer le flux de données provenant des différents logiciels de téléchargement du torrent testé.

- **Méthode d'analyse des données** : qui nous a permis de relever des différences qui permettent de distinguer l'utilisation des logiciels du téléchargement torrent.
- **Méthode pratique** : qui nous a permis d'implémenter ces différences dans un système de détection d'intrusion réseau (IDS) Snort, afin de valider nos solutions et tester leur fiabilité.

3.2.2 Matériel et logiciel utiliser dans notre rechercher :

Notre recherche a été mené dans un environnement ouvert, un réseau local connecté à internet.

Les expériences ont été faites avec :

- Un ordinateur portable Asus avec processeur Intel Core I7-7500U à 3.5 GHz et de 8 Go de RAM.
- Un ordinateur bureau avec un processeur Intel Core 2 Duo 2.93GHz et de 3 Go de Ram.
- Système d'exploitation Windows 10 64 bit.
- Oracle VM VirtualBox.
- Linux Ubuntu 18.04 64 bit.

Au cours de notre recherche on a utilisé les logiciels suivants :

- Ubuntu 18.04.
- Windows 7.
- Wireshark version 3.0.0 64bit.
- Utorrent version 3.5.5.
- BitTorrent version 7.10.
- Transmission version 2.94.

3.2.3 Objectif de notre recherche :

L'objectif de notre recherche est de crée des règles propres a Snort pour détecter l'utilisation du réseau Torrent dans une entreprise.

3.3 Mise au point du proxy :

3.3.1 Squid :

Ce type de logiciel est utilisé dans des entreprises, des établissements scolaires, il sert à contrôler les accès, les sites visités et le trafic sortant en général.

De plus, si une machine émet des informations via un programme malveillant par exemple, squid permettra de s'en apercevoir.

C'est un logiciel serveur à ce titre il permet de centraliser les connexions à internet, ainsi sur un réseau comportant un routeur, vous pouvez configurer une machine comme proxy, faire pointer tous les accès internet vers cette machine et en faire l'intermédiaire obligé entre le client et le routeur [21].

3.3.2 Mise en place :

Parce qu'une image vaut mieux que mille mots la figure suivante (figure 3.1) montre notre schéma d'installation de notre réseau.

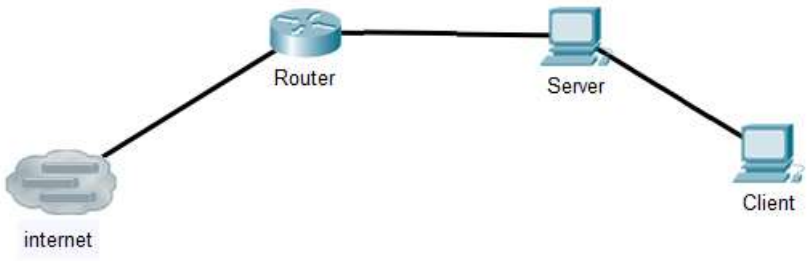


Figure 3.1. Schéma du réseau

a Installation de Squid :

Squid peut être téléchargé sur internet sous la forme de source tar.gz ou autre, et pour l'installation il y a une méthode par type de paquetage.

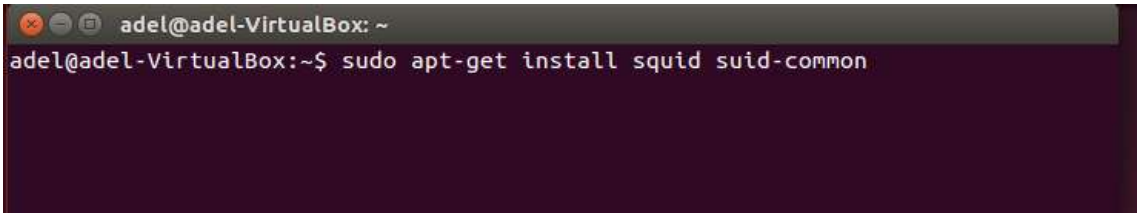


Figure 3.2. Installation de Squid.

b Configuration de Squid :

Pour la configuration squid a un fichier de configuration dans : etc/squid/squid.conf. Les paramètres par défaut lui permettent de fonctionner mais pas encore de bloquer donc on aura besoin de passer dans les fichiers de configuration quelques paramètre de squid.

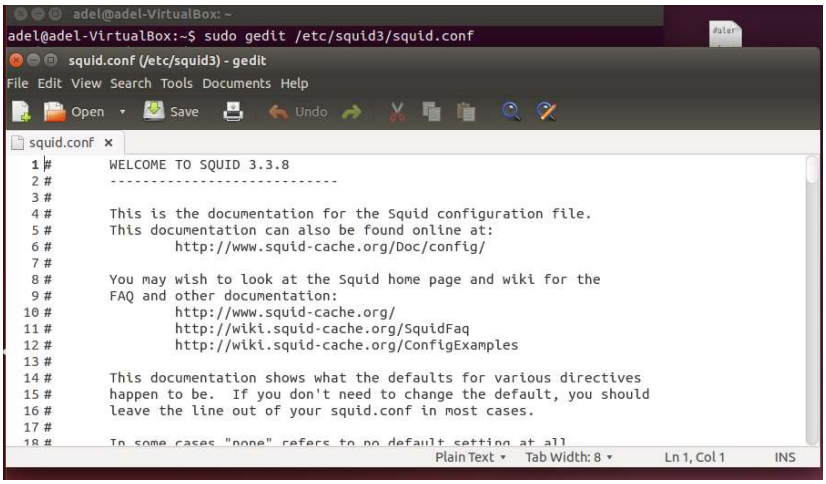


Figure 3.3.1. Configuration de Squid.

Dans notre cas on n'a pas besoin de bloquer l'accès à des sites web vu qu'on veut juste contrôler le trafic qui passe donc on autorise tous les accès.

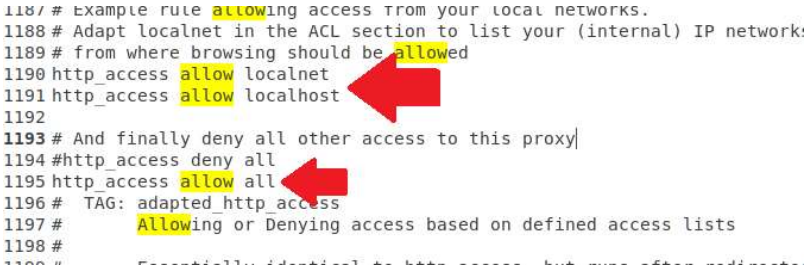


Figure 3.3.2. Configuration de Squid.

Une fois la configuration termineé on lance notre proxy au niveau de notre serveur il sera fonctionnel il nous reste juste de configurer le client avec notre adresse de proxy et l'accès sera autorisé.

Figure 3.3.3. Configuration de Squid.

3.3.3 Capteur des données :

Pour capturer le trafic, on a utilisé Wireshark sous Ubuntu.

Wireshark est un logiciel d'analyse réseau (sniffer) qui permet de visualiser l'ensemble des données transitant sur la machine qui l'exécute, et d'obtenir des informations sur les protocoles applicatifs utilisés.

Les octets sont capturés en utilisant la librairie réseau PCAP, puis regroupés en blocs d'informations et analysés par le logiciel [22].

Pour la capture du trafic on doit lancer d'abord Wireshark, une fois que l'application est lancée nous avons démarré la capture des trames à travers l'ongle « démarrer la capture de paquet ».

Une fois que c'est fait on a lancé le navigateur web pour accéder au site web de téléchargement du fichier torrent, ensuite quand notre fichier est téléchargé on l'a exécuté pour que le téléchargement final commence avec le logiciel BitTorrent ou Utorrent.

Et pour finir une fois que les étapes précédentes sont faites nous avons arrêté la capture et sauvegardé le fichier de capture.

La figure 3.4 montre la fenêtre de Wireshark après la capture des paquets elle est décomposée en trois zones :

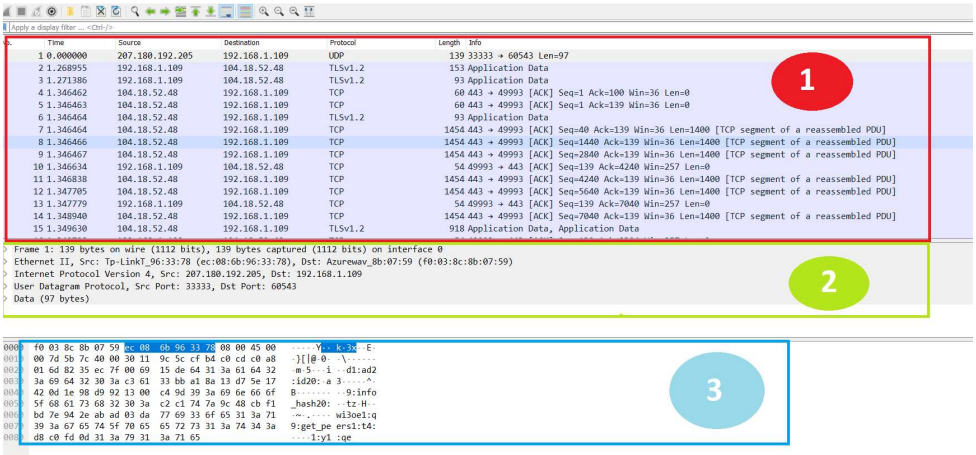


Figure 3.4. Fenêtre de Wireshark après la capture.

- Zone numérotée (1) sur la figure 3.4 : liste de l'ensemble des paquets capturé.
- Zone numérotée (2) sur la figure 3.4 : affiche le détail d'un paquet sélectionné.
- Zone numérotée (3) sur la figure 3.4 : présente l'ensemble du paquet sous forme octale et ASCII.

Nos captures ont été faites sur plusieurs reprises, nous avons capturé les paquets de chaque logiciel séparément des autres pour augmenter la fiabilité de notre recherche.

3.3.4 Les étapes de l'analyse :

- **Le site torrent :** Pour commencer nous allons mettre la règle la plus intuitive, ce qui est simplement sur le téléchargement du métafile, ce téléchargement étant issue d'une requête GET nous allons donc tester la règle suivante :

```
alert tcp any any -> any any (msg: "P2P .torrent metafile"; content:"HTTP/"; content:".torrent";
 flow:established,to_server; classtype:policy-violation; sid:1100010; rev:1;)
```

Figure 3.5. Règle de teste de métafile.

- **Client hello :** À la suite de l'étude précédemment menée sur les torrents, nous avons observé le réseau avec notre sniffer **Wireshark**, à la suite nous avons établi une politique de détection, par rapport aux empreintes laissées par le torrent, nous avons donc procédés à l'analyse du site web jusqu'au début du téléchargement, la capture nous donne les parties suivantes :

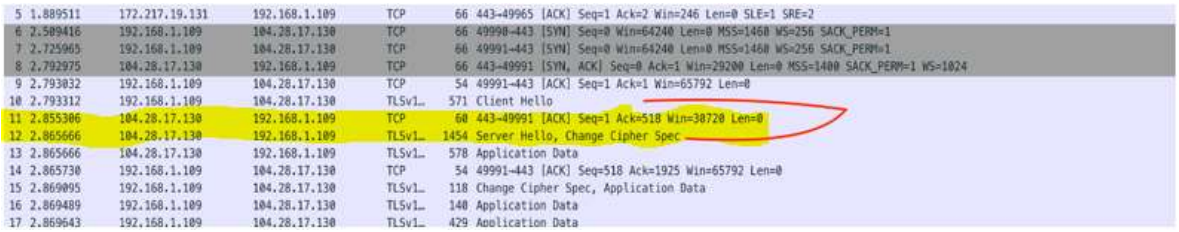


Figure 3.6. Paquet client Hello.

Nous observons ici en rouge une requête qui nous sera très importante car elle concerne l'établissement de la connexion avec le site web et il nous envoie une réponse « serveur hello », il désigne l'établissement de la connexion avec le site.

Nous avons séléctionné le paquet 10 a zone centrale de « Wireshark » permet de visualiser clairement les différentes couches d'encapsulation du paquet [Figure 3.7] :

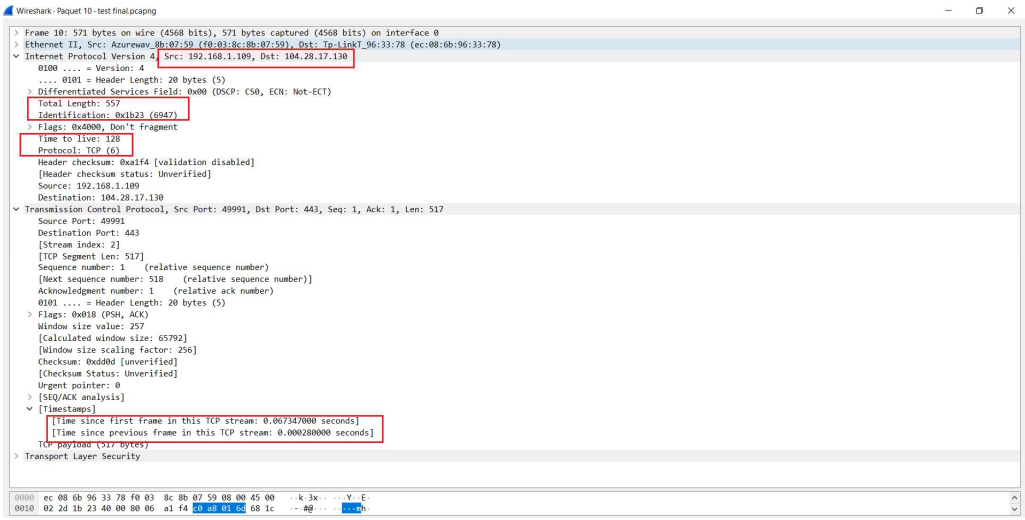


Figure 3.7. Détails paquet client Hello

À partir de ces différentes couches, on peut extraire les informations suivantes :

- Les adresses IP source et destination, numéro de porte source et destination, le type d'adressage et le protocole utilisé.

- Indentification : chaque connexion doit être identifiée de manière unique.

Ceci est fait en utilisant la paire d'identifiants de socket (combinaison d'adresse IP et Port) correspondant aux deux extrémités de la connexion.

- Les numéros de séquence : sont utilisés pour décompter les données dans le flux d'octets, le numéro de séquence indique le premier octet des données. Durant cet échange initial, les numéros de séquence des deux parties sont synchronisés.

- TTL (Time To live) : indique le temps pendant lequel une information doit être conservée.

- Stream index : Stream index ou l'index de flux est un mappage Wireshark interne.

- Windows size value : est une annonce de la quantité de données (en octets) que le dispositif récepteur est prêt à recevoir à tout moment. Le périphérique de réception peut utiliser cette valeur pour contrôler le flux de données ou comme mécanisme de contrôle de flux [23].

- RTT to ACK : Round Time Trip to acknowledgement.

RTT to ACK est un mécanisme de temporisation et de retransmission, RTT est le temps que met un signal pour parcourir l'ensemble d'un circuit fermé.

Dans les faits, le délai avant la retransmission, doit être supérieur à RTT moyen d'un segment.

Pour pouvoir en être sûre nous effectuons la commande « nslookup » sur l'adresse de **www.torrent9.io** et nous n'y avons pas accès, ce qui implique qu'ils utilisent plusieurs serveurs, pour pouvoir échapper au blocage, mais par aubaine le « client hello » et le « server hello » apparaissent en clair et nous donne une piste pour nos prochaines règles.

- **Trackers :** Ici nous retrouvons un terme que nous connaissons déjà bien celui des trackers, ils sont un ensemble de serveurs sur lesquels les informations sur les torrents sont constamment mis à jour, nous nous y sommes donc penché pour trouver une piste et il se trouve qu'il y a un échange DNS avec ces serveurs là et cela s'observe comme suit :

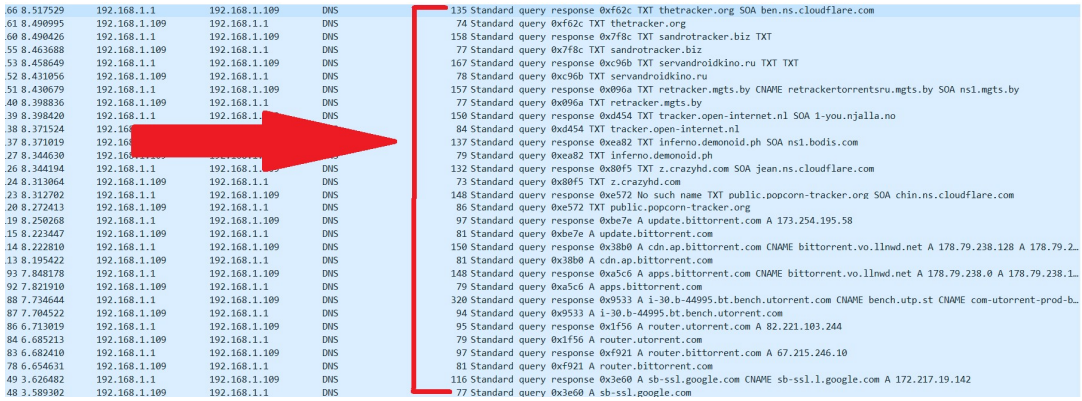


Figure 3.8. Paquets des trackers.

Cette méthode de prime abord semble OK, sauf qu'après fine observation nous voyons :

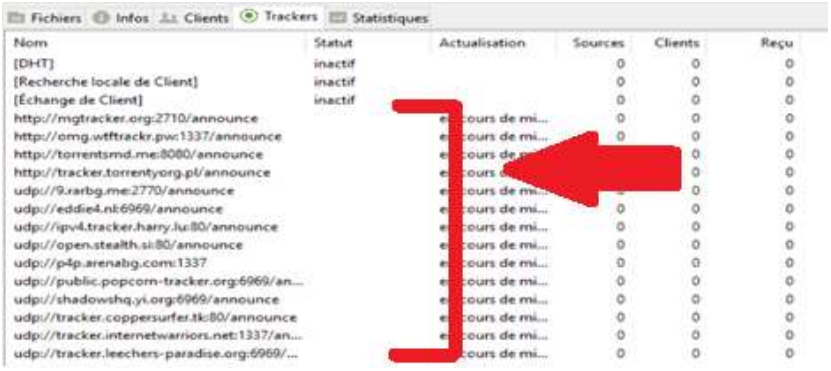


Figure 3.9.1. Trackers dans le torrent 1.

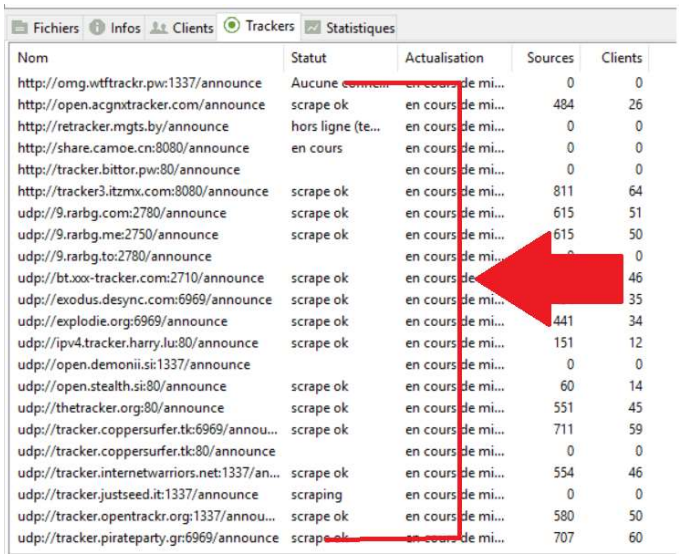


Figure 3.9.2. Trackers dans le torrent 2.

Nous voyons un net changement au niveau des trackers, ce qui en soit nous empêche de mettre des alertes, parce que les mots clés changent les adresses changent, pour un torrent la solution de la règle sur les tracker est bonne, mais ne permet pas de traiter un cas général.

D'un autre côté, si le mode Trackerless est actif, nous allons être victime d'une saturation du traffic au cas où beaucoup de peers sont actifs, car comme vu précédemment ce mode fait de chaque peer un tracker.

- **BitTorrent protocole :** En nous penchant sur nos documents et par observation de notre trafic, nous voyons l'apparition récurrente d'un protocole qui est complètement relatif au torrent uniquement :

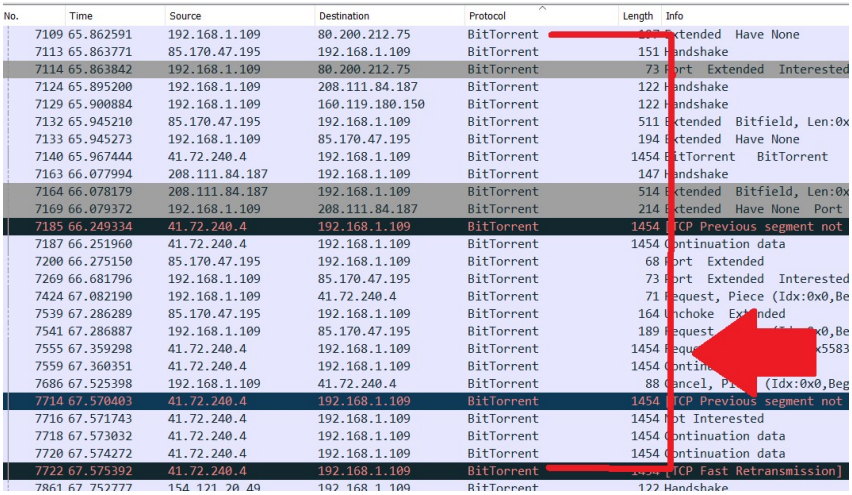


Figure 3.10. Paquets BitTorrent protocole.

Nous établirons donc une règle sur le « BitTorrent Handshake » un protocole de type connexion.

- **DHT (distributed hash table) :** Sur encore une de notion favorite le trackerless, en effet il y a une multitude de ping qui sont envoyés pour avoir des informations de hash dans le cas de trackerless torrent, quand ces pings sont bencoded, ils ont l'allure suivante :

d1:ad2:id20:abcdefghij0123456789e1:q4:ping1:t2:aa1:y1:qe

Ou la partie en jaune est un code qui est généré par chaque cas de DHT et la partie en rouge est l'ID du fichier, la règle devient donc facile a élaboré.

- **P2P anounce request :** Nous avons le peer to peer, qui englobe même le concept de torrent, du coup pour établir la connexion avec le serveur distant une requête **GET** est envoyé suivi de l'annonce de la connexion, nous n'avons pas eu besoin de wireshrak pour implementer cette règle.

3.3.5 Tableau des signatures :

Le Tableau ci-dessous regroupe toutes les signatures obtenu après analyse :

N°	Signatures	Empreintes
1	Métafile	« .torrent ; HTTP/ »
2	Groupes utiliser par client Hello	« |00 1d 00 17 00 18 | »
3	BitTorrent protocole	« |42 69 74 54 6f 72 72 65 6e 74 20 70 72 6f 74 6f 63 6f 6c| »
4	DHT	« d1\ :ad2\ :id20\: »
)5	P2P anounce request	« /announce ; info_hash= »

Tableau 3.1. Tableau des signatures.

3.4 SNORT :

SNORT est un NIDS écrit par Martin Roesch en 1998 et développé par Sourcefire. Disponible sous licence GNU, son code source est accessible et modifiable à partir du site officiel.

SNORT a la capacité d'effectuer l'analyse du trafic en temps réel et la journalisation de paquets sur Protocole Internet (IP) [24].

Snort effectue l'analyse de protocole, la recherche de contenu, et l'appariement de contenu, Il utilise une sonde pour détecter les attaques, le débordement système, les scans de ports, etc.

Les trois modes de fonctionnement de Snort sont les suivants :

- **Le mode sniffer :** il lit les paquets circulant sur le réseau et les affiche sur l'invite de commande.
- **Le mode paquet logger :** Snort journalise le trafic réseau dans des répertoires.
- **Le mode détecteur d'intrusion :** Snort analyse le trafic réseau, compare ce trafic à des règles déjà définies par l'utilisateur et établi des actions à exécuter.

3.4.1 Architecture de Snort :

L'architecture de SNORT est modulaire, elle est composée des éléments suivants :

- **Un noyau de base :** Au démarrage, ce noyau charge un ensemble de règles, compile, optimise et classe celles-ci.

  Durant l'exécution, le rôle principal du noyau est la capture de paquets.
- **Une série de pré – processeurs :** ceux-ci améliorent les possibilités de SNORT en matière d'analyse et de recomposition du trafic capturé.

  Ils reçoivent les paquets directement capturés, éventuellement les retravaillant puis les fournissent au moteur de recherche de signatures.
- **Une série d'analyses est ensuite appliquée aux paquets :** Ces analyses se composent principalement de comparaisons de différents champs des headers des protocoles (IP, ICMP, TCP et UDP) par rapport à des valeurs précises.

Après la détection d'intrusion, une série de « output plugins » permet de traiter cette intrusion de plusieurs manières : envoie vers un fichier log, envoie d'un message d'alerte vers un serveur syslog, stocker cette intrusion dans une base de données SQL. Cette figure (figure 3.12) illustre le principe du fonctionnement de SNORT qui est basé sur différents modules de traitement de flux.

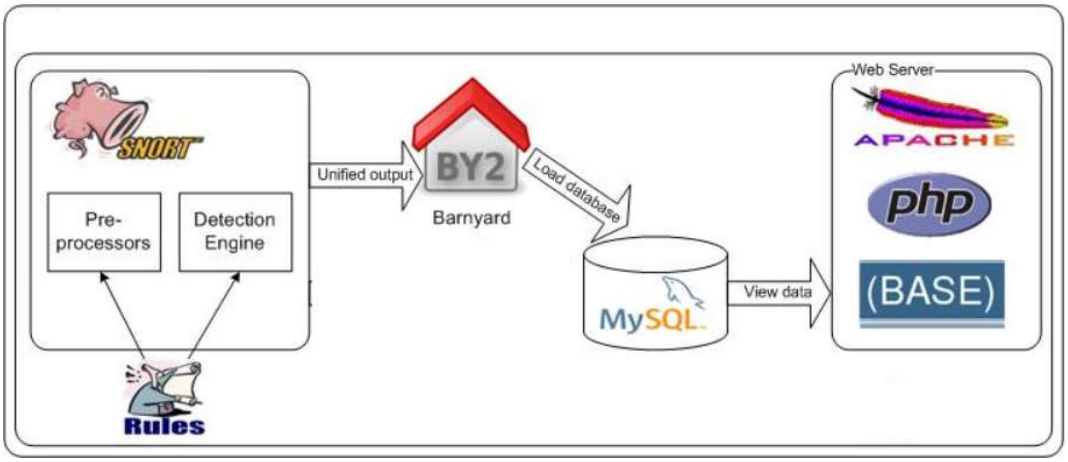


Figure 3.11. Mode de fonctionnement de Snort.

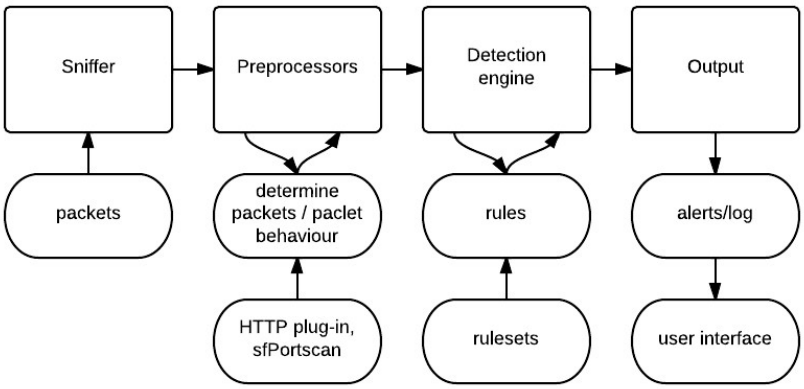


Figure 3.12. Architecture de Snort.

3.4.2 Création des règles Snort :

Une règle Snort est composée de deux parties présentées sous le format suivant :

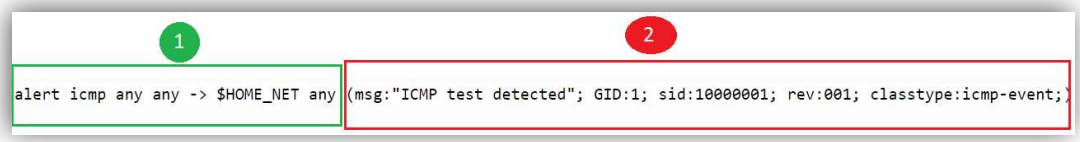


Figure 3.13. Règles Snort.

1 - L'entête de la règle : contient l'action de la règle, le protocole, les adresses IP et les ports source et destination aussi les masque réseau.

- **L'action de règle :** L'entête de règle contient l'information qui définit 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.

  - Alerte : génère une alerte en utilisant la méthode d'alerte sélectionnée, et alors journalise le paquet [numéro 1 dans la Figure 3. 14].

  - log : journalise le paquet

  - Pass : ignore le paquet

  - Activate : alerte et alors active une autre règle dynamic

  - Dynamic : reste passive jusqu'à être activée par une règle activate, alors agit comme une règle log.

- **Les protocoles :** Il y a trois protocoles IP que Snort analyse actuellement pour des comportements suspicieux : tcp, udp, et icmp [numéro 2 dans la Figure 3.7].
- **Les adresses IP :** Les adresses IP source ou destination et on peut utiliser le mot clé ''ANY'' pour définie n'importe quelle adresse [numéro 3 dans la Figure 3.14].
- **Les numéros de ports :** Les ports source ou destination et on peut utiliser le mot clé ''ANY'' pour définie n'importe quel port [numéro 4 dans la Figure 3. 14].
- **L'opérateur de direction :** L'opérateur de direction "->" indique l'orientation, ''->'' unidirectionnel, ou <-> bidirectionnel [numéro 5 dans la Figure 3. 14].

2 - Les option des règles : 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 va exécuter.

- **Msg :** affiche un message dans les alertes et journalise les paquets [numéro 6 dans la Figure 3. 14].
- **Logto :** journalise le paquet dans un fichier nommé par l'utilisateur au lieu de la sortie standard.
- **Id :** identifié le champ ID de fragment de l'entête IP pour une valeur spécifiée.
- **Fragbits :** identifié les bits de fragmentation de l'entête IP.

- **Dsize :** identifié la taille de la charge du paquet contre une valeur.
- **Flags :** identifié les drapeaux TCP pour certaines valeurs.
- **Seq :** identifié le champ TCP de numéro de séquence pour une valeur spécifique.
- **Ack :** identifié le champ TCP d'acquittement pour une valeur spécifiée.
- **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.

Figure 3.14. Règles Snort.

3.5 Extraction des signatures qui permettent la détection du réseau Torrent :

Au cours de notre analyse sur l'utilisation du réseau torrent on a trouvé plusieurs identifiant lié à ce dernier.

Certains d'entre eux sont faciles à mettre en œuvre et implémenté ainsi que tester avec Snort, par contre d'autres sont plus complexes voir même irréalisables.

Le fonctionnement de Snort est basé sur les empreintes qui font mesure de déclencher les alertes de l'utilisation du torrent, donc tous d'abord il faudra crée ces empreintes.

Les empreintes doivent respecter le paramétrage et le fonctionnement de Snort pour que on aura la possibilité de les implémenter dans ce dernier plus tard.

Chaque nouvelle empreinte a une règle spécifique à elle, pour la création des règles on doit spécifier la direction du trafic ainsi que les octets qui identifient ce dernier.

3.5.1 Détection de l'empreinte des groupes dans client Hello :

Notre première empreinte identifie les groupes supporter utilisé par le client hello envoie par le navigateur pour accéder au site torrent.

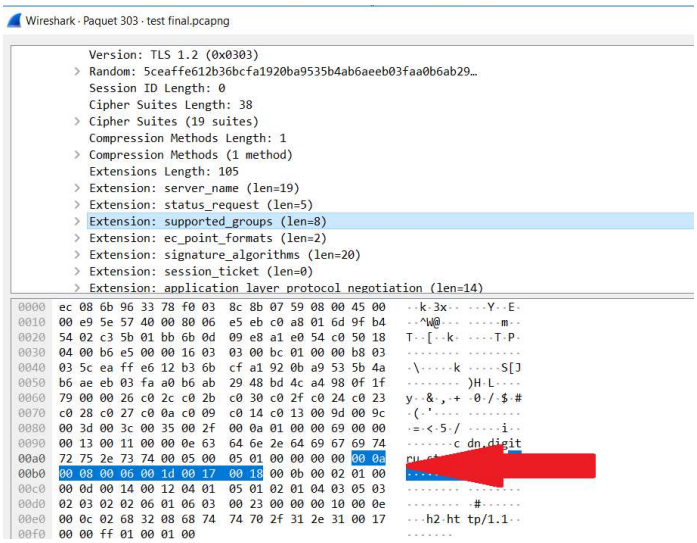


Figure 3.15.1 L'empreinte des groupes supporter utilisées dans client Hello.

```
alert tcp any any -> any any (msg:"handshak groupe utilisé par le client hello"; content:"|00 1d 00 17 00 18|";
offset:20; sid:2099998 ; rev:1;)
```

Figure 3.15.2 L'empreinte des groupes supporter utilisées dans client Hello.

3.5.2 Détection de l'empreinte du protocole BitTorrent :

La deuxième empreinte identifie le protocole BitTorrent utilisé par le réseau lors du téléchargement.

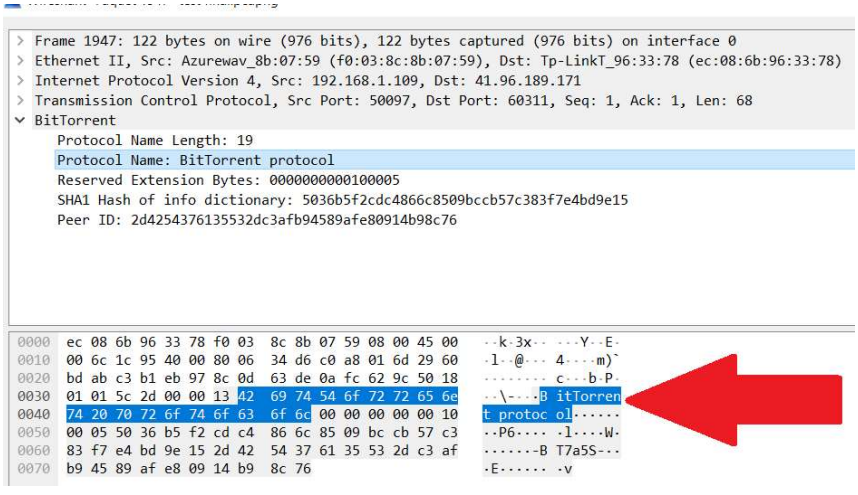


Figure 3.16.1. L'empreinte du protocole BitTorrent.

```
alert tcp any any -> any any (msg:"BitTorrent protocole detecter "; content:"|42 69 74 54 6f 72 72 65 6e 74 20 70
72 6f 74 6f 63 6f 6c|";  offset:20; classtype:non-standard-protocol; sid:2099994 ; rev:1;)
```

Figure 3.16.2. L'empreinte du protocole BitTorrent.

3.5.3 Détection de l'empreinte de P2P anounce requist :

La troisième empreinte identifie l'établissement du Peer-to-Peer BitTorrent par annonce.

```
alert tcp any any -> any any (msg:"Etablisement du P2P BitTorrent par announce"; flow:to_server,established;
content:"GET"; depth:4; content:"/announce"; distance:1; content:"info_hash="; offset:4; content:"event=started";
offset:4; classtype:policy-violation; sid:20002180; rev:2;)
```

Figure 3.17. L'empreinte du l'établissement de Peer-to-Peer BitTorrent par annonce.

3.5.4 Détection de l'empreinte de DHT :

La quatrième empreinte identifie le DHT ping pour les trackers.

```
alert udp any any -> any any (msg: "DHT ping detecter"; content:"d1\:ad2\:id20\:";content:"ping";classtype:policy-
violation; sid:2100021; rev:1;)
```

Figure 3.18. L'empreinte du DHT ping

DHT est une technologie permettant la mise en place d'une table de hachage dans un système réparti.

Une table de hachage est une structure de données de type *(clé, valeur)* ,chaque donnée est associée à une clé elle est distribuée sur le réseau.

Les tables de hachage permettent de répartir le stockage de données sur l'ensemble des nœuds du réseau, chaque nœud étant responsable d'une partie des données.

Les tables de hachage distribuées fournissent un algorithme pour retrouver le nœud responsable de la donnée et sa valeur à partir de la clé.

3.5.5 Détection de l'empreinte du métafile :

La cinquième empreinte identifie la métafile du fichier torrent.

```
alert tcp any any -> any any (msg: ".torrent metafile detecter"; content:"HTTP/"; content:".torrent";
classtype:policy-violation; sid:2100010; rev:1;)
```

Figure 3.19. L'empreinte du métafile.

3.5.6 Ajout des règles :

Une fois que les règles sont créées on doit les ajouter au Snort pour cella nous avons créé un fichier dédié au règles dans << /etc/snort/rules/local.rules >> une fois le fichier créé nous avons ajouté dans ce fichier.

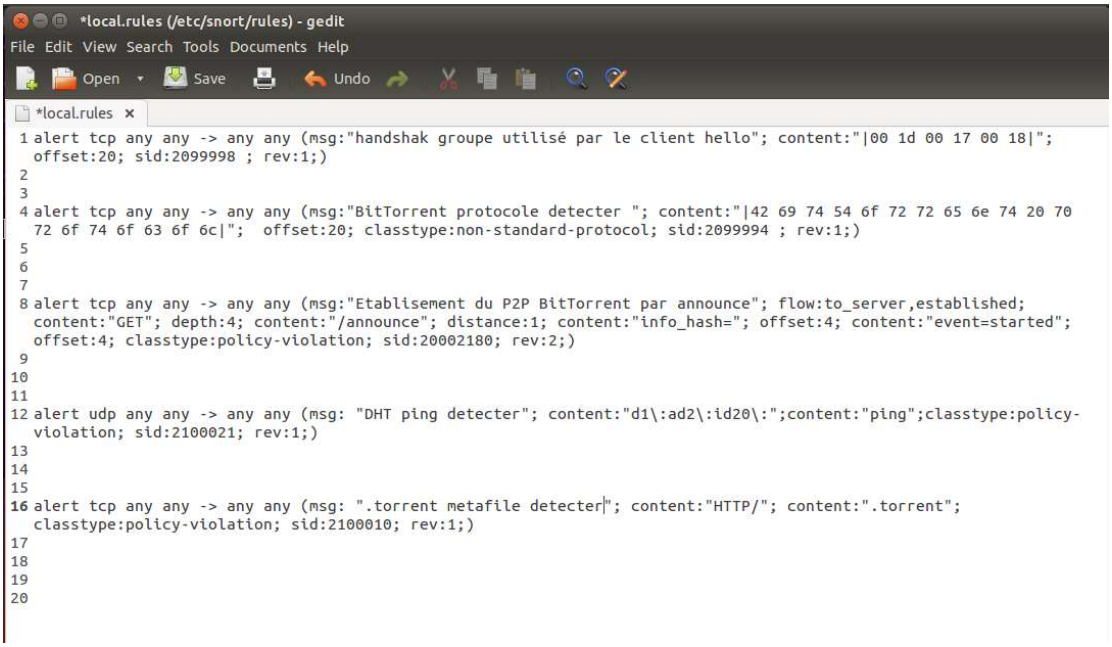


Figure 3.20. Implémentation des règles.

Aussi nous avons fait des modifications sur le fichier de configuration de Snort qui est Snort.conf nous avons indiqué à Snort le chemin pour charger le fichier <<local.rules >> (On a décoché le commentaire de la ligne : include $ RULE_PATH / local.rules dans snort.conf). Lorsque Snort démarre, il utilisera directement le fichier de configuration snort.conf pour charger toutes les règles en local.rules.

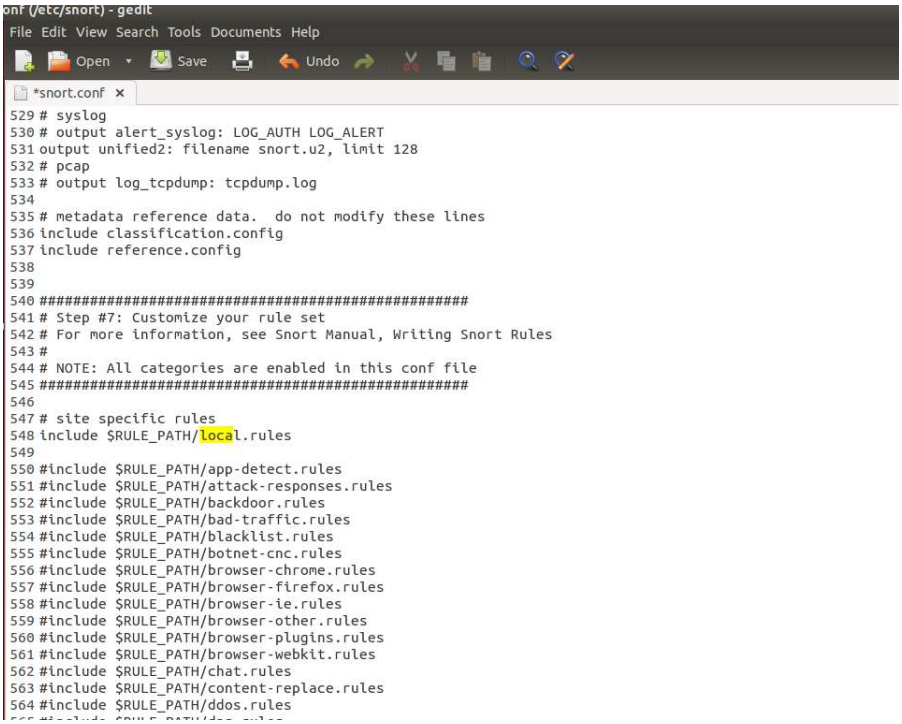


Figure 3.21. Modification du chemin de local.rules.

3.6 Conclusion :

Pour établir une politique de sécurité il est très important de connaître les éléments auxquels nous aurons à faire face, dans ce chapitre il a été tout aussi important de mettre en œuvre Snort que de connaître toutes les subtilités du Peer-to-Peer et des torrents, pour cela notre analyse avec le Wireshark s'est avéré très importante car sans elle nous aurions pas pu extraire les information de l'entête de nos paquets et ainsi mettre des alertes convenables, Snort seule n'est donc pas suffisant.

Dans le chapitre suivant nous allons entamer la génération des règles Snort propre à chacune des empreintes obtenues et la simulation de notre réalisation pour juger à fiabilité de nos règles.

Chapitre 4 Détection du Torrent

4.1 Introduction :

L'extraction des empreintes seules ne peut pas être suffisante pour nous alerter de l'utilisation du torrent et protéger notre réseau.

Pour cella on doit créer des règles selon les différentes empreintes récolter grâce à l'analyse établit, le logiciel approprié pour ce travail c'est Snort déjà décrit précédemment.

Nous allons ici mettre ce système en place dans un cas pratique qui pourrait se mapper avec l'architecture réseau des entreprises.

4.2 Architecture de test :

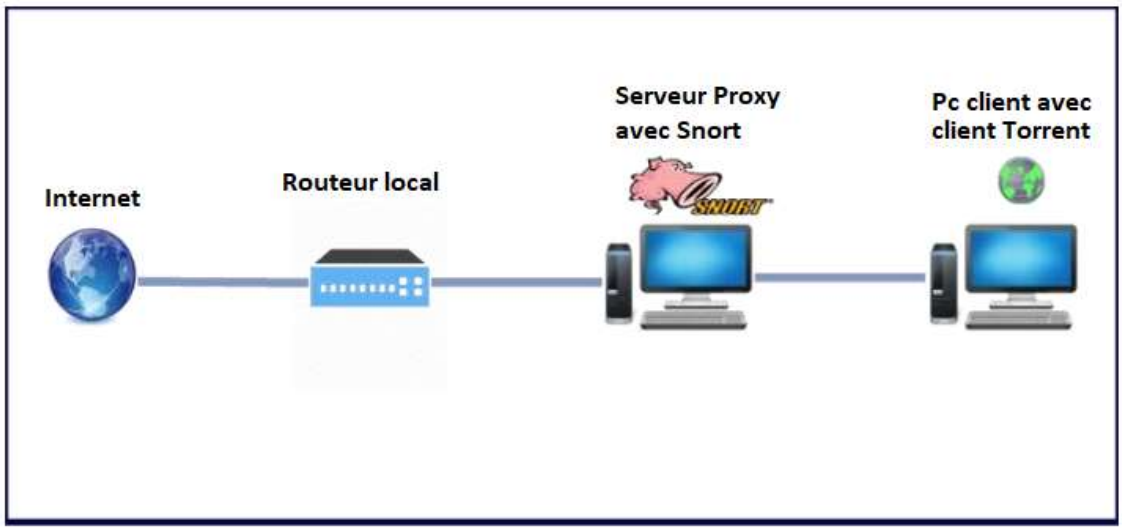


Figure 4.1. Architecture du test.

4.2.1 Installation sur matériels :

1 - Sur PC 1 :

Nous avons installé pour le teste Snort version 2.9.13 sous Ubuntu avec son interface BASE pour le premier test.

Nous avons aussi utilisé un pc qui contient déjà Snort mais cette fois ci sous Windows 7 avec la version 2.8.6 qui est installée dans le centre des systèmes et des réseaux d'information et de communication de l'université Saad Dahlab Blida 1), ce dernier est relié avec son interface BASE aussi Pour le deuxième test.

Les deux machines contiennent un proxy déjà installé.

BASE (Basic Analysis and Security Engine) :

BASE est une interface web qui permet de visualiser les alertes générées par Snort.

Ce dernier enregistrera les données d'alertes dans une base de données MySQL qui sera ensuite lue par BASE et affichée via un serveur web Apache ou Wamp.

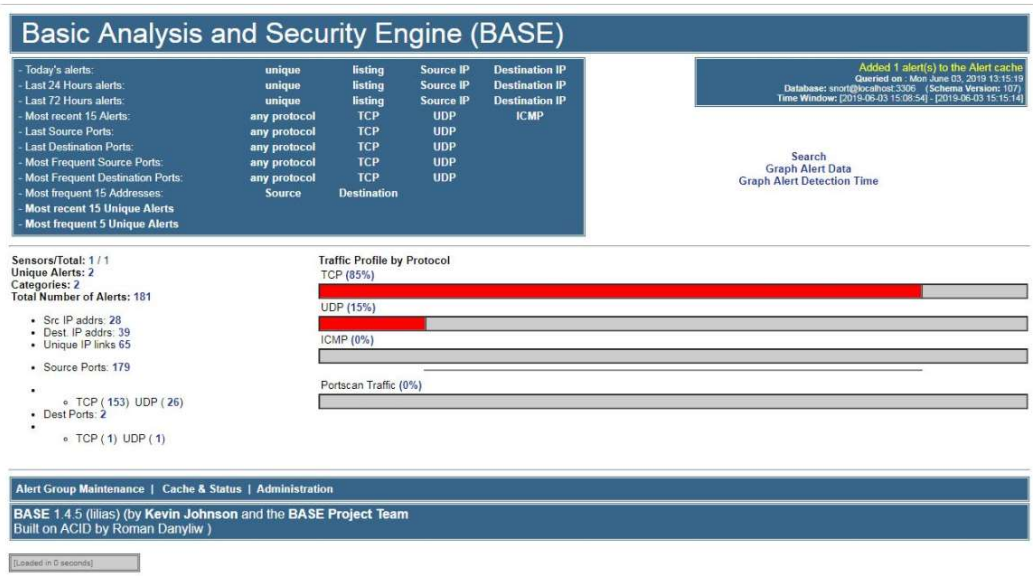


Figure 4.2. Interface BASE.

2 - Sur PC 2 :

Nous avons installé les deux clients torrent les plus utilisés Utorrent ainsi que BitTorrent.

4.2.2 Lancement de Squid

Le lancement de Squid est très facile à établir une fois que les configurations par rapport aux adresses et aux autorisations (voir chapitre 3) ont été faites, il nous reste juste à le démarrer avec la commande suivante :

Figure 4.3 Commande de démarrage de squid.

Une fois que Squid à démarrer notre pc-2 (client) aura accès à internet et on pourra superviser le trafic qui transite entre le pc-2 et internet au niveau de notre serveur.

4.2.3 Démarrage de Snort :

- Pour démarrer le fonctionnement de Snort sous Ubuntu on a deux possibilités :

Première possibilité nous avons utilisé deux lignes de commandes.

1 - la première ligne pour lancer Snort avec ces options :

```
sudo /usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i eth0 -D
```

/usr/local/bin/snort : chemin de l'emplacement de Snort.

-q -u snort : Le lancement et l'utilisateur de Snort.

-g snort : Le groupe de snort.

-c /etc/snort/snort.conf : Le chemin du fichier de configuration de snort.

-i eth0 : L'interface utilisé pour la détection.

-D : C'est pour le mode daemon.

2 - La deuxième ligne de commande c'est pour lancer le Barnyard :

Barnyard2 : C'est un interpréteur open-source pour les fichiers binaires de sortie de Snort de format unified2.

Il permet de prendre en charge l'inscription des événements en base de données et libère donc des ressources à Snort qui peut davantage se concentrer sur la détection des intrusions [25].

```
sudo barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.u2 -w /var/log/snort/barnyard2.waldo -g snort -u snort
```

barnyard2 -c /etc/snort/barnyard2.conf : Indiquer au barnyad le chemin de son fichier de configuration.

-d /var/log/snort : l'emplacement pour rechercher le fichier Snort de sortie binaire.

-f snort.u2 : Le nom du fichier à rechercher.

-w /var/log/snort/barnyard2.waldo : le chemin vers le fichier waldo (fichier de point de contrôle).

-g snort -u snort : le user et le groupe a utiliser.

La deuxième possibilité utilise aussi deux lignes de commande :

```
sudo /usr/local/bin/snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```

/usr/local/bin/snort : chemin de l'emplacement de Snort.

-A console : Pour activer le mode console.

-q -u snort : Le lancement et l'utilisateur de Snort.

-g snort : Le groupe de snort.

-c /etc/snort/snort.conf : Le chemin l du fichier de configuration de snort.

-i eth0 : L'interface utilisé pour la détection.

La deuxième ligne de commande pour le Barnyard c'est la même que celle citée précédemment.

- Pour démarrer Snort sous Windows on le lance avec les options suivantes :

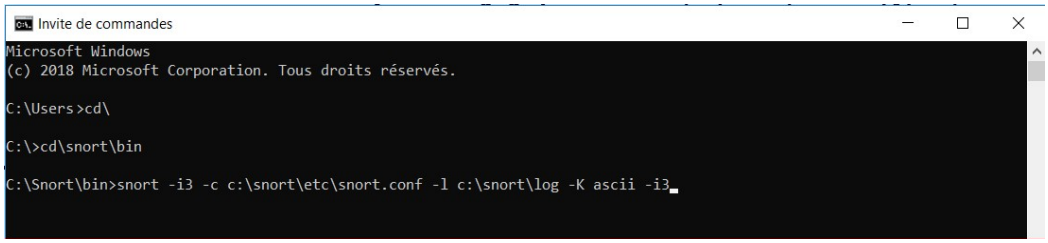


Figure 4.4. Commande Snort sous Windows.

-i 3 : l'interface (3) à écouter.

-c /etc/snort/snort.conf : le chemin de fichier snort.conf.

-l : connexion au répertoire.

-K ascii : mode de journalisation (pcap [default], ascii, none)

4.3 Déroulement du teste :

Les règles obtenues dans le chapitre 3 et qu'on doit les utiliser sont classées dans le tableau suivant :

Règle n°	Les règles
1	Alert tcp any any -> any any (msg : "handshak groupe utilisé par le client hello ‘’»; content :’’|00 1d 00 17 0017|’’ ;offset :20 ; sid :2099998 ; rev :1 ;)
2	Alert tcp any any ->any any (msg :’’BitTorrent Protocole détecter ‘’ ; content :’’|42 69 74 57 6f 72 72 65 6e 74 20 70 72 6f 74 6f 63 6f 6c|’’ ; offset :20 ; classtype :non-standard-protocol ; sid 2099994 ; rev1 ;)
3	Alert tcp any any -> any any (msg:"P2P BitTorrent announce request"; flow:to_server,established; content:"GET"; depth:4; content:"/announce"; distance:1; content:"info_hash="; offset:4; content:"event=started"; offset:4; classtype:policy-violation; sid:2180; rev:2;)
4	Alert tcp any any -> any any (msg : "DHT ping détecter ‘’»; content : ‘’d1\ :ad2\ :id20\ :’’ ;content :’’ping’’ ;classtype :policy-violation ; sid :2100021 ; rev :1 ;)
5	Alert tcp any any -> any any (msg : ".torrent métafile détecter ‘’; content : ‘’http/’’ ; content :’’.torrent’’ ; classtype :policy-violation; offset :20 ; sid :2100010 ; rev :1 ;)

Tableau 4.1. Les règles.

Pour juger la fiabilité de nos résultats, le test s'est déroulé comme suit :

Notre test a été divisé en deux parties.

- La première partie exécutée Snort sous Ubuntu qui est dans le pc 1, ensuite sur le deuxième pc nous avons lancé un par un deux différents clients de torrent Utorrent ainsi que BitTorrent.

  Pour voir si Snort à l'aide de nos regèles peut détecter le trafic torrent d'une part, de l'autre pour voir si ces règles n'impactent pas le trafic normal du web.

- La deuxième partie nous avons refait les mêmes étapes mais cette fois si avec la version Snort sous Windows.

4.4 Mise en œuvre du test :

Une fois que tous notre matériel est mis en place et configurer on débute notre test selon les étapes suivante :

1- Sur le pc 2 nous avons accédé au site du téléchargement du fichier Torrent dans notre cas on a choisi le site le plus utiliser www.torrent9.cz .

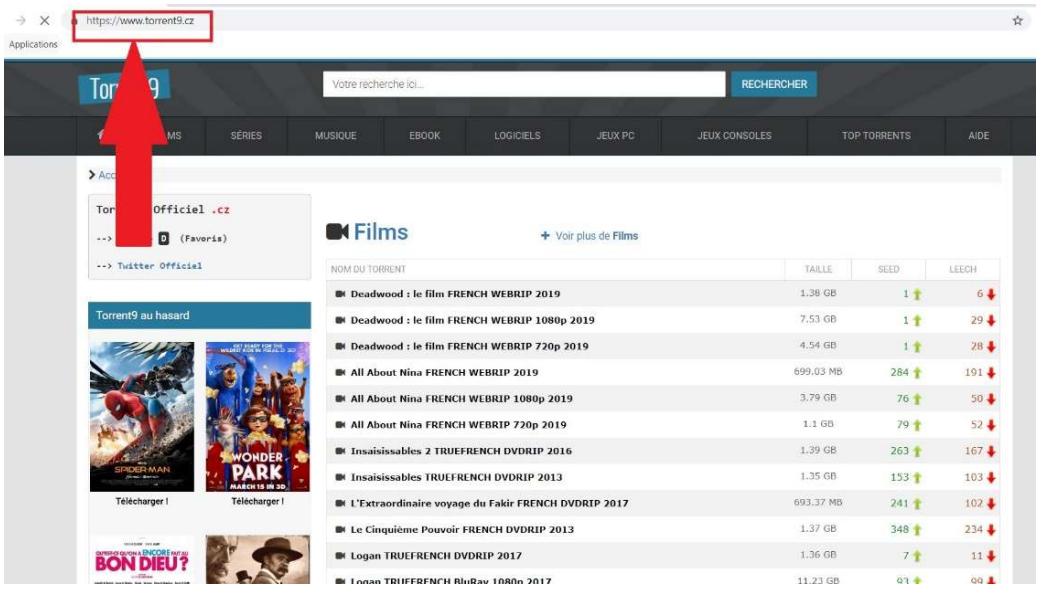


Figure 4.5. Accès au site du fichier torrent.

2- Une fois sur le site nous avons pris pour exemple un film à télécharger.

Figure 4.6.1. Téléchargement du fichier torrent.

Figure 4.6.2. Téléchargement du fichier torrent.

3- Une fois que le fichier torrent a été télécharger nous l'avons l'exécuté pour que le téléchargement du film entreprend dans le logiciel Utorrent.

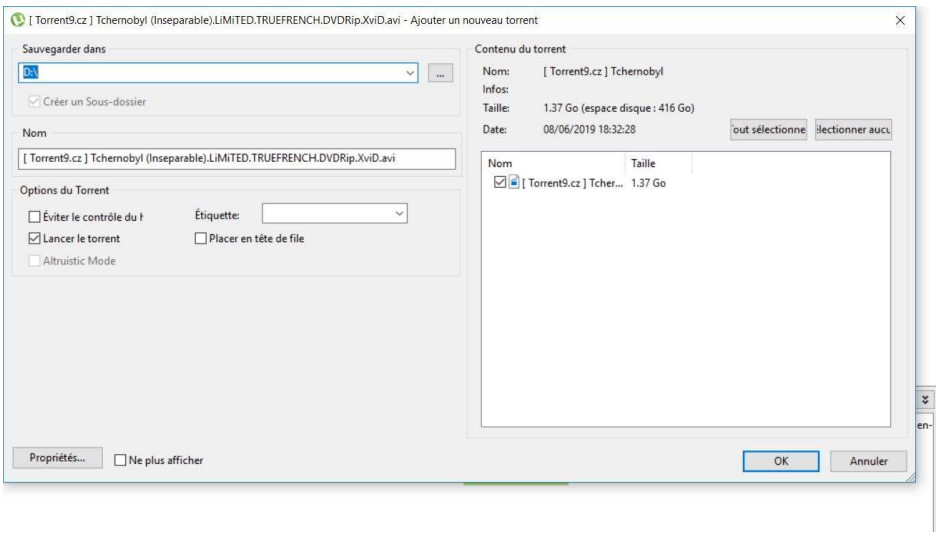


Figure 4.7.1 Exécution du fichier torrent.

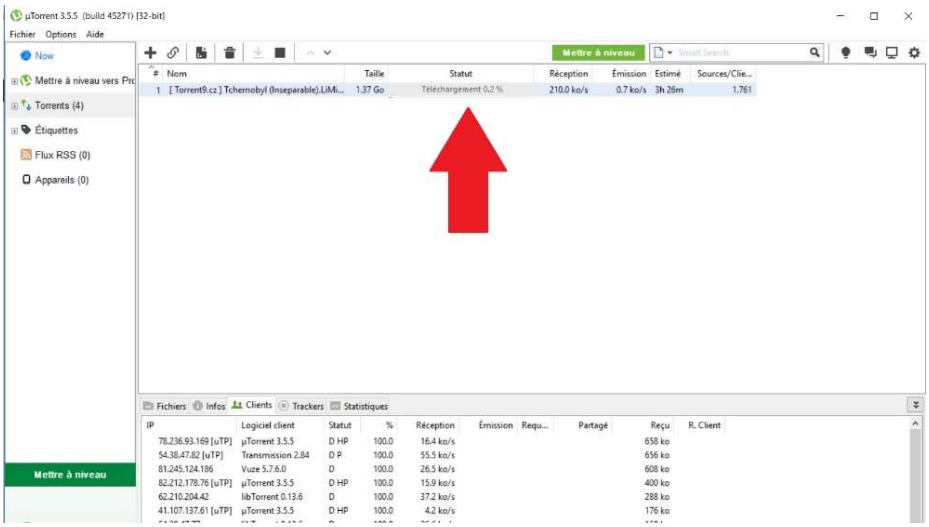


Figure 4.7.2 Début du téléchargement du film.

4- Une fois que la manipulation au niveau du pc 2 a été faite nous avons lancer Wireshark sur le pc 1 pour avoir un œil sur le traffic qui transit.

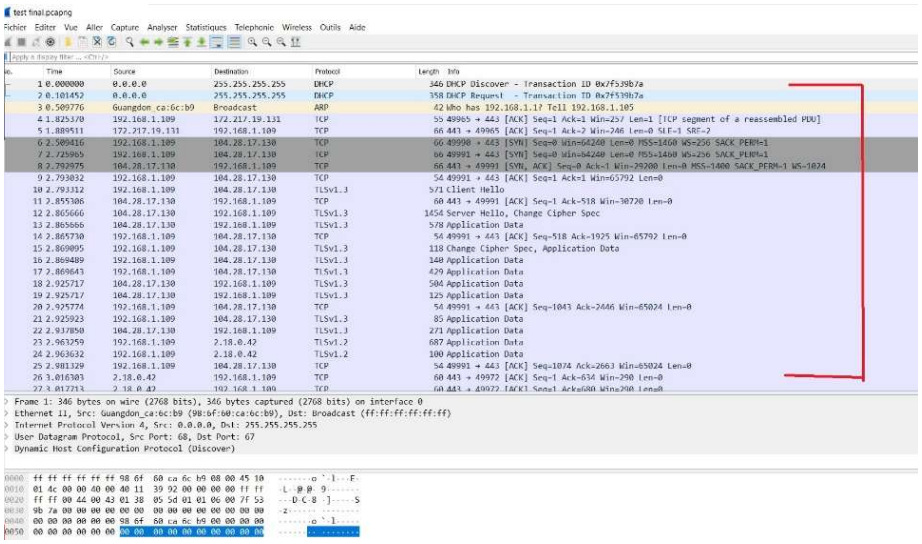


Figure 4.8. Affichage de Wireshark.

4.5 Résultats obtenus :

Une fois que nous avons accédé à l'interface de BASE on a récolté des alertes puisqu'elles sont enregistrées dans la base de données, aussi on peut accéder aux listes des alertes comme il est montré dans la figure ci-dessous lors du lancement du test et le début du téléchargement du fichier torrent.

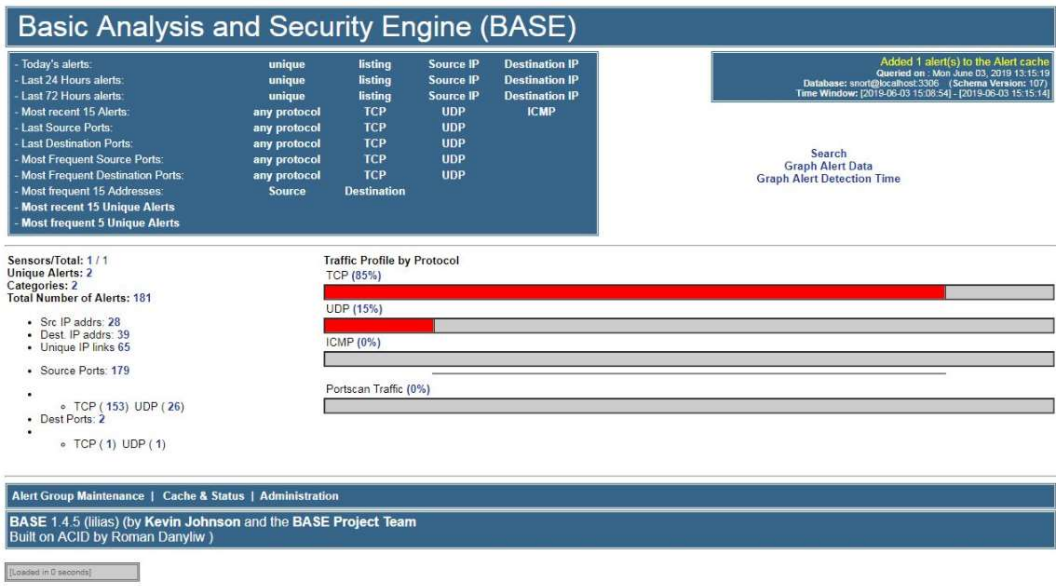


Figure 4.9. Interface du BASE.

Dans l'onglet << Totale number of Alerts >> il y'a un lien qui nous renvoie à la liste des alertes totales (figure 4.10).

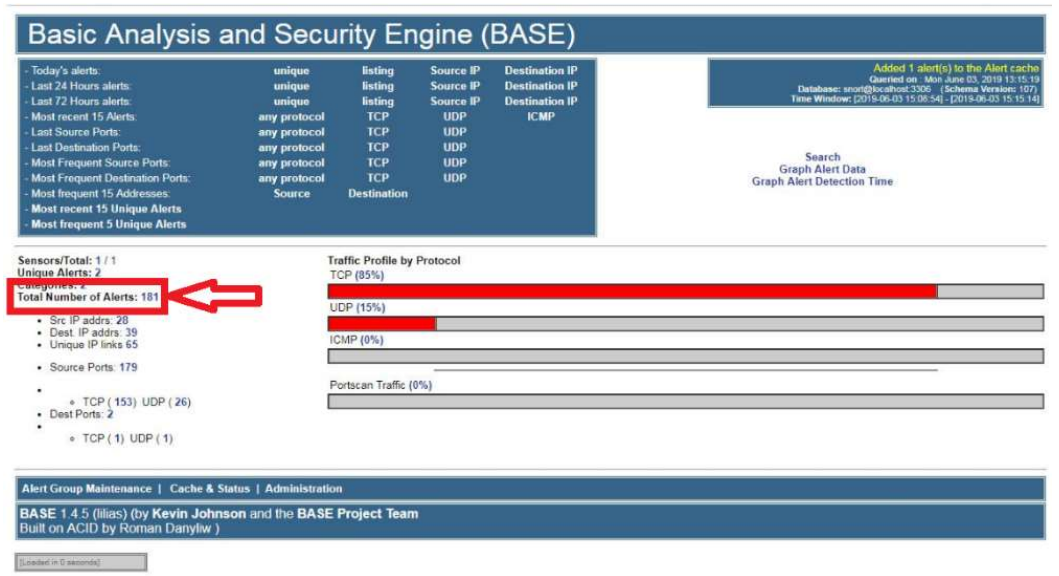
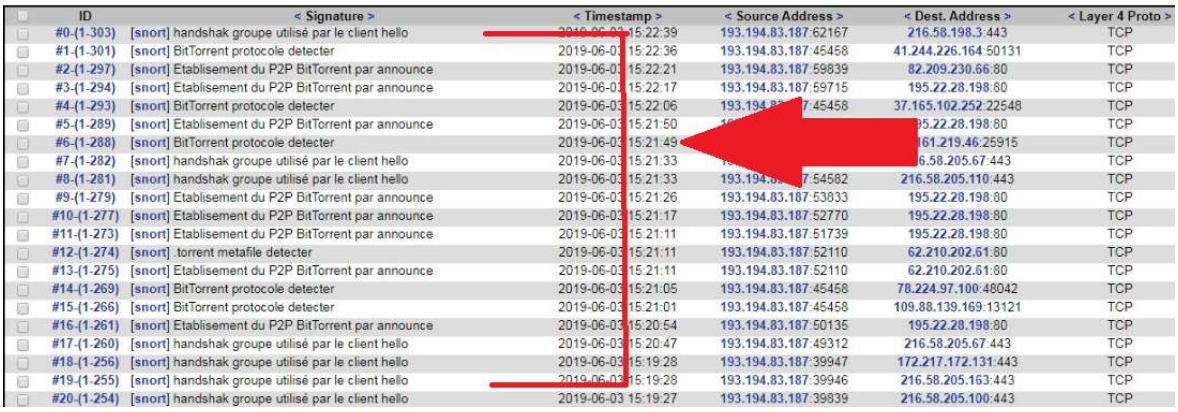


Figure 4.10. Lien « Total number of alerts ».

Dans cet affichage, on a la signature avec le type d'alerte, ainsi que la date de déclenchement d'alerte (figure 4.11).



ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
#0-(1-303)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:22:39	193.194.83.187:62167	216.58.198.3:443	TCP
#1-(1-301)	[snort] BitTorrent protocole detecter	2019-06-03 15:22:36	193.194.83.187:45458	41.244.226.164:50131	TCP
#2-(1-297)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:22:21	193.194.83.187:59839	82.209.230.66:80	TCP
#3-(1-294)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:22:17	193.194.83.187:59715	195.22.28.198:80	TCP
#4-(1-293)	[snort] BitTorrent protocole detecter	2019-06-03 15:22:06	193.194.83.187:45458	37.165.102.252:22548	TCP
#5-(1-289)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:21:50	[illegible]	195.22.28.198:80	TCP
#6-(1-288)	[snort] BitTorrent protocole detecter	2019-06-03 15:21:49	[illegible]	161.219.46:25915	TCP
#7-(1-282)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:21:33	[illegible]	6.58.205.67:443	TCP
#8-(1-281)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:21:33	193.194.83.187:54582	216.58.205.110:443	TCP
#9-(1-279)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:21:26	193.194.83.187:53833	195.22.28.198:80	TCP
#10-(1-277)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:21:17	193.194.83.187:52770	195.22.28.198:80	TCP
#11-(1-273)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:21:11	193.194.83.187:51739	195.22.28.198:80	TCP
#12-(1-274)	[snort] .torrent metafile detecter	2019-06-03 15:21:11	193.194.83.187:52110	62.210.202.61:80	TCP
#13-(1-275)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:21:11	193.194.83.187:52110	62.210.202.61:80	TCP
#14-(1-269)	[snort] BitTorrent protocole detecter	2019-06-03 15:21:05	193.194.83.187:45458	78.224.97.100:48042	TCP
#15-(1-266)	[snort] BitTorrent protocole detecter	2019-06-03 15:21:01	193.194.83.187:45458	109.88.139.169:13121	TCP
#16-(1-261)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:20:54	193.194.83.187:50135	195.22.28.198:80	TCP
#17-(1-260)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:20:47	193.194.83.187:49312	216.58.205.67:443	TCP
#18-(1-256)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:19:28	193.194.83.187:39947	172.217.172.131:443	TCP
#19-(1-255)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:19:28	193.194.83.187:39946	216.58.205.163:443	TCP
#20-(1-254)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:19:27	193.194.83.187:39839	216.58.205.100:443	TCP

Figure 4.11. Liste totale des alertes.

Le lien « Unique Alerts » de la page principale de base (figure 4.12) nous dirige vers la liste des alertes.

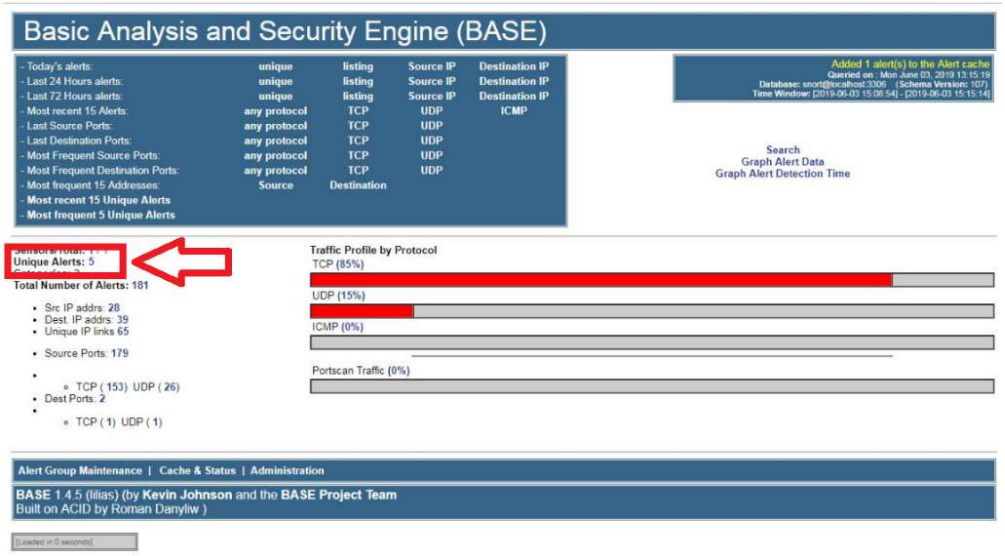


Figure 4.12. Lien Unique Alerts.

Sur la figure 4.13 on remarque que Snort a détecté l'utilisation du torrent.

Il a généré 5 alertes qui correspondent à nos 5 règles implémenter auparavant.

Le nombre de chaque règle ainsi que le temp de déclanchement des alertes se diffèrent d'une règle à une autre, cella explique que lors du téléchargement du fichier torrent il y'a plusieurs étapes à parcourir.

Dans la colonne des nombres total des règles on prend par exemple le lien « 7 »,

Il nous dirige vers la liste des alertes de chaque règle.

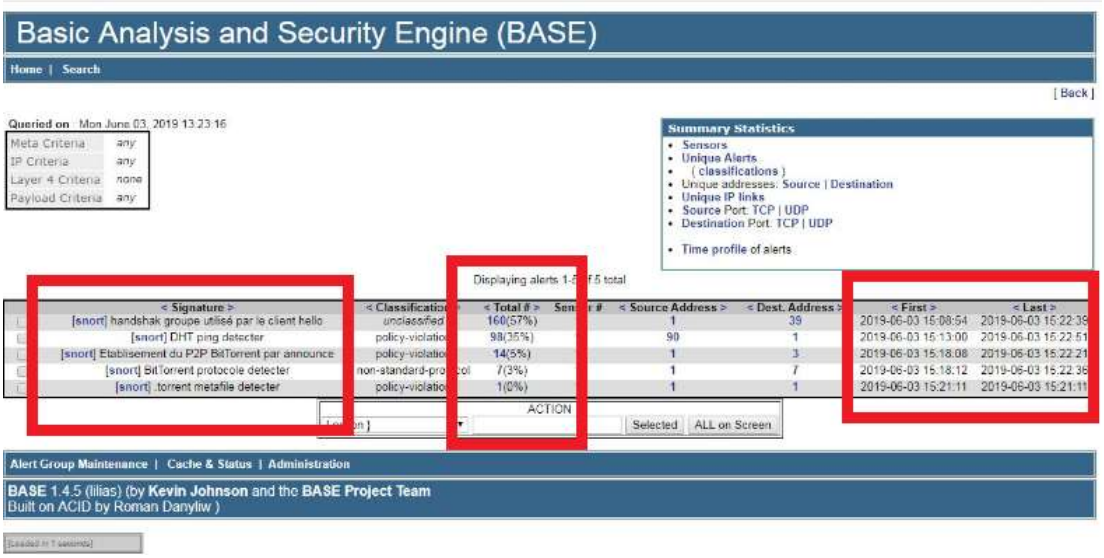


Figure 4.13. Liste des alertes.

Nous avons pris pour exemple la règle en question « BitTorrent protocole détecter » numéro 4 dans le tableau de BASE (figure 4.14).

Sur cette fenêtre BASE nous donne les informations sur le type d'alerte, l'adresse source, destination ainsi que le numéro ID de l'alerte.

BASE nous affiche dans la (figure 4.14) 7 alertes avec 7 adresses de destination différente.

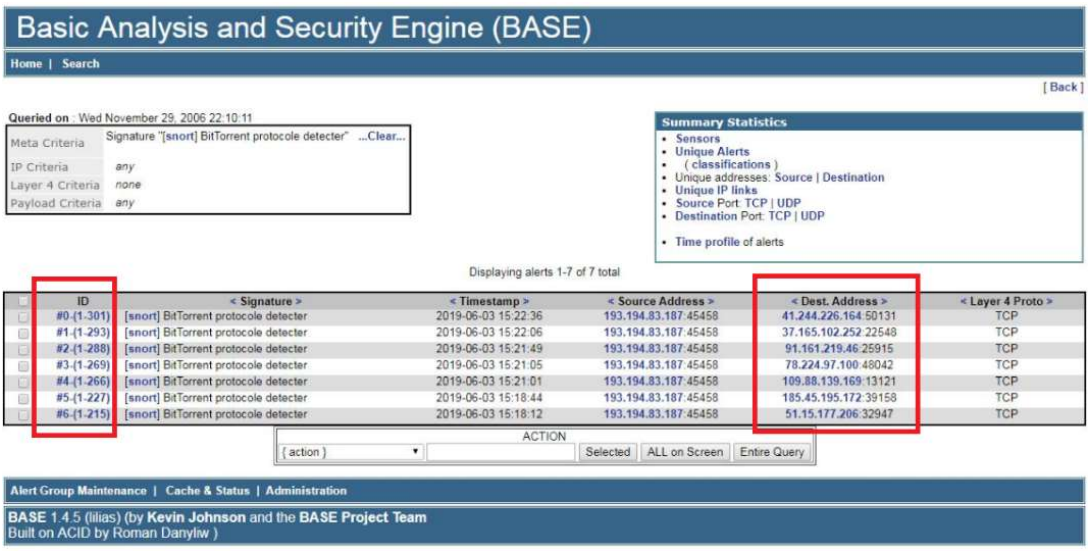


Figure 4.14. Nombre d'alerte générer pas la règles « BitTorrent protocole détecter »

En cliquant sur l'identifiant de l'alerte, BASE affiche les données appartenant à cette dernière (figure 4.15).

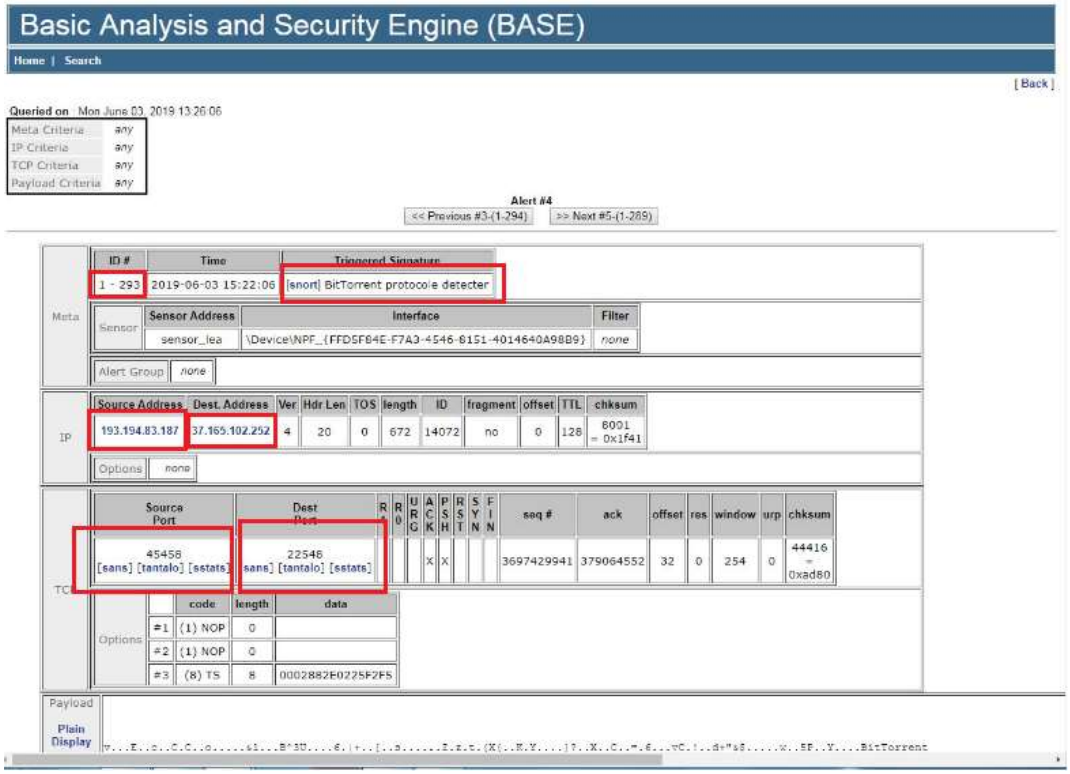


Figure 4.15. L'alerte identifie par ID :293.

Dans cet affichage, nous pouvons relever plusieurs informations sur l'alerte identifiée par ID : 293, comme l'adresse de la source et de la destination ainsi que les numéros de port de la source et la destination etc…

Les figures suivantes représentent les alertes générer par nos différentes signatures :

1 – Les alertes générées par la signature « handshak groupe utilisé par le client hello» :

ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
#0-(1-303)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:22:39	193.194.83.187:62167	216.58.198.3:443	TCP
#1-(1-282)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:21:33	193.194.83.187:54652	216.58.205.67:443	TCP
#2-(1-281)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:21:33	193.194.83.187:54582	216.58.205.110:443	TCP
#3-(1-260)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:20:47	193.194.83.187:49312	216.58.205.67:443	TCP
#4-(1-256)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:19:28	193.194.83.187:39947	172.217.172.131:443	TCP
#5-(1-255)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:19:28	193.194.83.187:39946	216.58.205.163:443	TCP
#6-(1-254)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:19:27	193.194.83.187:39839	216.58.205.100:443	TCP
#7-(1-210)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:15:14	193.194.83.187:9720	216.58.205.78:443	TCP
#8-(1-207)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:53	193.194.83.187:7136	172.217.212.94:443	TCP
#9-(1-206)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:52	193.194.83.187:7051	216.58.205.78:443	TCP
#10-(1-205)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:52	193.194.83.187:7050	216.58.205.67:443	TCP
#11-(1-199)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:15	193.194.83.187:2696	172.64.207.31:443	TCP
#12-(1-194)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:14	193.194.83.187:2507	172.64.207.31:443	TCP
#13-(1-195)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:14	193.194.83.187:2523	216.58.205.174:443	TCP
#14-(1-196)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:14	193.194.83.187:2530	13.33.216.129:443	TCP
#15-(1-197)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:14	193.194.83.187:2531	13.33.216.129:443	TCP
#16-(1-198)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:14	193.194.83.187:2533	13.33.216.129:443	TCP
#17-(1-190)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:13	193.194.83.187:2433	54.173.110.148:443	TCP
#18-(1-189)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:13	193.194.83.187:2282	54.173.110.148:443	TCP
#19-(1-191)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:13	193.194.83.187:2432	52.216.115.61:443	TCP
#20-(1-192)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:13	193.194.83.187:2451	37.252.173.38:443	TCP
#21-(1-193)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:13	193.194.83.187:2452	37.252.173.38:443	TCP
#22-(1-188)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:12	193.194.83.187:2294	52.21.76.141:443	TCP
#23-(1-187)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:12	193.194.83.187:2295	143.204.15.2:443	TCP
#24-(1-186)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:10	193.194.83.187:2112	54.173.110.148:443	TCP
#25-(1-185)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:10	193.194.83.187:2093	143.204.15.2:443	TCP
#26-(1-179)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:07	193.194.83.187:1621	52.21.76.141:443	TCP
#27-(1-180)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:07	193.194.83.187:1640	104.28.26.227:443	TCP
#28-(1-181)	[snort] handshak groupe utilisé par le client hello	2019-06-03 15:14:07	193.194.83.187:1716	13.33.216.129:443	TCP

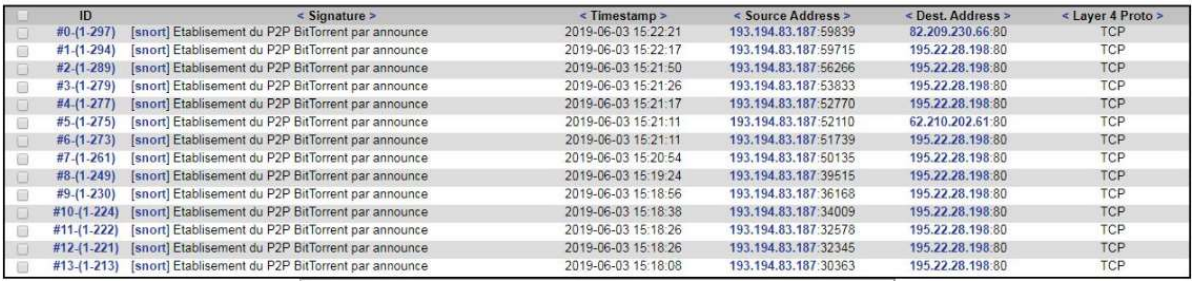
Figure 4.16. Les alertes générées par la signature « handshak groupe utilisé par le client hello».

2– Les alertes générées par la signature « DHT ping detecter » :

ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
#0-(1-309)	[snort] DHT ping detecter	2019-06-03 15:22:51	78.229.94.82:34361	193.194.83.187:45458	UDP
#1-(1-308)	[snort] DHT ping detecter	2019-06-03 15:22:44	94.204.209.64:43611	193.194.83.187:45458	UDP
#2-(1-307)	[snort] DHT ping detecter	2019-06-03 15:22:44	84.102.36.3:44822	193.194.83.187:45458	UDP
#3-(1-306)	[snort] DHT ping detecter	2019-06-03 15:22:43	160.120.171.246:1027	193.194.83.187:45458	UDP
#4-(1-305)	[snort] DHT ping detecter	2019-06-03 15:22:42	90.100.77.209:61137	193.194.83.187:45458	UDP
#5-(1-304)	[snort] DHT ping detecter	2019-06-03 15:22:41	207.189.30.105:1946	193.194.83.187:45458	UDP
#6-(1-302)	[snort] DHT ping detecter	2019-06-03 15:22:37	109.88.188.30:26085	193.194.83.187:45458	UDP
#7-(1-300)	[snort] DHT ping detecter	2019-06-03 15:22:29	94.23.203.134:6880	193.194.83.187:45458	UDP
#8-(1-299)	[snort] DHT ping detecter	2019-06-03 15:22:26	91.182.130.188:50500	193.194.83.187:45458	UDP
#9-(1-298)	[snort] DHT ping detecter	2019-06-03 15:22:24	80.102.179.237:57274	193.194.83.187:45458	UDP
#10-(1-296)	[snort] DHT ping detecter	2019-06-03 15:22:19	80.200.154.231:44822	193.194.83.187:45458	UDP
#11-(1-295)	[snort] DHT ping detecter	2019-06-03 15:22:18	213.189.187.75:11687	193.194.83.187:45458	UDP
#12-(1-292)	[snort] DHT ping detecter	2019-06-03 15:22:06	165.169.94.197:1044	193.194.83.187:45458	UDP
#13-(1-291)	[snort] DHT ping detecter	2019-06-03 15:22:05	76.68.154.220:35754	193.194.83.187:45458	UDP
#14-(1-290)	[snort] DHT ping detecter	2019-06-03 15:22:04	160.120.232.182:32973	193.194.83.187:45458	UDP
#15-(1-287)	[snort] DHT ping detecter	2019-06-03 15:21:49	197.26.139.71:50500	193.194.83.187:45458	UDP
#16-(1-286)	[snort] DHT ping detecter	2019-06-03 15:21:47	105.102.9.51:15550	193.194.83.187:45458	UDP
#17-(1-285)	[snort] DHT ping detecter	2019-06-03 15:21:42	87.64.52.145:27060	193.194.83.187:45458	UDP
#18-(1-284)	[snort] DHT ping detecter	2019-06-03 15:21:40	213.49.11.163:26085	193.194.83.187:45458	UDP
#19-(1-283)	[snort] DHT ping detecter	2019-06-03 15:21:37	205.237.50.187:44822	193.194.83.187:45458	UDP
#20-(1-280)	[snort] DHT ping detecter	2019-06-03 15:21:30	88.191.206.75:11257	193.194.83.187:45458	UDP
#21-(1-278)	[snort] DHT ping detecter	2019-06-03 15:21:22	109.130.109.79:52350	193.194.83.187:45458	UDP
#22-(1-276)	[snort] DHT ping detecter	2019-06-03 15:21:16	24.202.183.218:50500	193.194.83.187:45458	UDP
#23-(1-272)	[snort] DHT ping detecter	2019-06-03 15:21:08	194.187.249.48:31759	193.194.83.187:45458	UDP
#24-(1-271)	[snort] DHT ping detecter	2019-06-03 15:21:07	74.59.123.143:59696	193.194.83.187:45458	UDP
#25-(1-270)	[snort] DHT ping detecter	2019-06-03 15:21:06	78.251.140.18:26085	193.194.83.187:45458	UDP
#26-(1-267)	[snort] DHT ping detecter	2019-06-03 15:21:02	106.8.224.1:16927	193.194.83.187:45458	UDP
#27-(1-268)	[snort] DHT ping detecter	2019-06-03 15:21:02	109.236.90.89:34674	193.194.83.187:45458	UDP
#28-(1-265)	[snort] DHT ping detecter	2019-06-03 15:20:56	176.113.74.25:58485	193.194.83.187:45458	UDP
#29-(1-264)	[snort] DHT ping detecter	2019-06-03 15:20:55	24.212.33.96:10939	193.194.83.187:45458	UDP
#30-(1-263)	[snort] DHT ping detecter	2019-06-03 15:20:55	91.161.219.46:17426	193.194.83.187:45458	UDP
#31-(1-262)	[snort] DHT ping detecter	2019-06-03 15:20:54	41.100.169.80:44231	193.194.83.187:45458	UDP
#32-(1-259)	[snort] DHT ping detecter	2019-06-03 15:20:46	41.85.161.134:37382	193.194.83.187:45458	UDP
#33-(1-258)	[snort] DHT ping detecter	2019-06-03 15:19:32	165.169.94.197:1044	193.194.83.187:45458	UDP
#34-(1-257)	[snort] DHT ping detecter	2019-06-03 15:19:30	104.221.77.78:24874	193.194.83.187:45458	UDP
#35-(1-253)	[snort] DHT ping detecter	2019-06-03 15:19:27	154.124.133.113:36748	193.194.83.187:45458	UDP
#36-(1-252)	[snort] DHT ping detecter	2019-06-03 15:19:27	160.120.148.76:20691	193.194.83.187:45458	UDP
#37-(1-251)	[snort] DHT ping detecter	2019-06-03 15:19:26	174.94.155.52:62348	193.194.83.187:45458	UDP
#38-(1-250)	[snort] DHT ping detecter	2019-06-03 15:19:24	41.143.79.226:51413	193.194.83.187:45458	UDP

Figure 4.17. Les alertes générées par la signature « DHT ping detecter ».

3 – Les règles générées par la signature « Etablissement du P2P BitTorrent par announce » :



ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
#0-(1-297)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:22:21	193.194.83.187:59839	82.209.230.66:80	TCP
#1-(1-294)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:22:17	193.194.83.187:59715	195.22.28.198:80	TCP
#2-(1-289)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:21:50	193.194.83.187:56266	195.22.28.198:80	TCP
#3-(1-279)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:21:26	193.194.83.187:53833	195.22.28.198:80	TCP
#4-(1-277)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:21:17	193.194.83.187:52770	195.22.28.198:80	TCP
#5-(1-275)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:21:11	193.194.83.187:52110	62.210.202.61:80	TCP
#6-(1-273)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:21:11	193.194.83.187:51739	195.22.28.198:80	TCP
#7-(1-261)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:20:54	193.194.83.187:50135	195.22.28.198:80	TCP
#8-(1-249)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:19:24	193.194.83.187:39515	195.22.28.198:80	TCP
#9-(1-230)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:18:56	193.194.83.187:36168	195.22.28.198:80	TCP
#10-(1-224)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:18:38	193.194.83.187:34009	195.22.28.198:80	TCP
#11-(1-222)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:18:26	193.194.83.187:32578	195.22.28.198:80	TCP
#12-(1-221)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:18:26	193.194.83.187:32345	195.22.28.198:80	TCP
#13-(1-213)	[snort] Etablisement du P2P BitTorrent par announce	2019-06-03 15:18:08	193.194.83.187:30363	195.22.28.198:80	TCP

Figure 4.18*.* Les alertes générées par la signature « BitTorrent protocole detecter ».

4 – Les alertes générées par la signature « .torrent metafile detecter » :

ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
#0-(1-274)	[snort] .torrent metafile detecter	2019-06-03 15:21:11	193.194.83.187:52110	62.210.202.61:80	TCP

Figure 4.14*.* Les alertes générées par la signature « .torrent metafile detecter ».

4.5.1 Constatation :

Le tableau suivant regroupe toutes les adresses qui on été en relation avec notre machine (source / destination) avec leur numéro de port :

< Timestamp >	< Source Address >	< Dest. Address >
2019-06-03 15:14:14	193.194.83.187:2523	216.58.205.174:443
2019-06-03 15:14:15	193.194.83.187:2696	172.64.207.31:443
2019-06-03 15:14:52	193.194.83.187:7050	216.58.205.67:443
2019-06-03 15:14:52	193.194.83.187:7051	216.58.205.78:443
2019-06-03 15:14:53	193.194.83.187:7136	172.217.212.94:443
2019-06-03 15:15:14	193.194.83.187:9720	216.58.205.78:443
2019-06-03 15:18:08	193.194.83.187:30363	195.22.28.198:80
2019-06-03 15:18:12	193.194.83.187:45458	51.15.177.206:32947
2019-06-03 15:18:26	193.194.83.187:32345	195.22.28.198:80
2019-06-03 15:18:26	193.194.83.187:32578	195.22.28.198:80
2019-06-03 15:18:38	193.194.83.187:34009	195.22.28.198:80
2019-06-03 15:18:44	193.194.83.187:45458	185.45.195.172:39158
2019-06-03 15:18:56	193.194.83.187:36168	195.22.28.198:80
2019-06-03 15:19:24	193.194.83.187:39515	195.22.28.198:80
2019-06-03 15:19:27	193.194.83.187:39839	216.58.205.100:443
2019-06-03 15:19:28	193.194.83.187:39946	216.58.205.163:443
2019-06-03 15:19:28	193.194.83.187:39947	172.217.172.131:443
2019-06-03 15:20:47	193.194.83.187:49312	216.58.205.67:443
2019-06-03 15:20:54	193.194.83.187:50135	195.22.28.198:80
2019-06-03 15:21:01	193.194.83.187:45458	109.88.139.169:13121
2019-06-03 15:21:05	193.194.83.187:45458	78.224.97.100:48042
2019-06-03 15:21:11	193.194.83.187:51739	195.22.28.198:80
2019-06-03 15:21:11	193.194.83.187:52110	62.210.202.61:80
2019-06-03 15:21:11	193.194.83.187:52110	62.210.202.61:80
2019-06-03 15:21:17	193.194.83.187:52770	195.22.28.198:80
2019-06-03 15:21:26	193.194.83.187:53833	195.22.28.198:80
2019-06-03 15:21:33	193.194.83.187:54582	216.58.205.110:443
2019-06-03 15:21:33	193.194.83.187:54652	216.58.205.67:443
2019-06-03 15:21:49	193.194.83.187:45458	91.161.219.46:25915
2019-06-03 15:21:50	193.194.83.187:56266	195.22.28.198:80
2019-06-03 15:22:06	193.194.83.187:45458	37.165.102.252:22548
2019-06-03 15:22:17	193.194.83.187:59715	195.22.28.198:80
2019-06-03 15:22:21	193.194.83.187:59839	82.209.230.66:80
2019-06-03 15:22:36	193.194.83.187:45458	41.244.226.164:50131
2019-06-03 15:22:39	193.194.83.187:62167	216.58.198.3:443

Tableau 4.2. Les adresses sources/destination.

Donc lors du processus de téléchargement d'un fichier torrent, on a plusieurs communications avec différentes adresses IP et différentes destinations, soit pour avoir des données à partir des trackers ou bien pour établir une connexion sécurisée avec le protocole SSL/TLS.

On a aussi plusieurs ports qui sont ouvert à chaque fois pour communiquer vu que le torrent utilise les deux protocoles de communication TCP et UDP.

4.6 Conclusion :

Dans ce chapitre, nous avons vu comment mettre en place et exécuter Snort en tant qu'un système de détection d'intrusion réseau (NIDS) sous Windows et sous Linux Ubuntu.

Nous avons posé les bases de la détection de signature et sur ces mêmes bases nous avons su, mettre en place des alertes qui s'avèrent être bel et bien fonctionnel, avec une interface graphique pour en faciliter la gestion, Les nombreux tests nous ont alors permis de peaufiner ce cas pratique afin d'avoir un travail assez optimiser en fin de compte, donc nous avons pu confirmer la fiabilité de nos règles.

Conclusion générale

L'utilisation de partage torrent en entreprise est à éviter, il lui apporte bien des soucis de sécurité et l'expose à des soucis problèmes judiciaire par exemple.

La détection de l'usage du torrent peut être fastidieuse, et demande un investissement chronophage en termes de sécurité, car toute une communauté est active pour pouvoir rendre son usage le plus furtif possible et mettre en place des moyens pour toujours et encore contourné les restrictions à son niveau.

Dans l'analyse qui a précédé la mise en place des règles nous avons découvert certaines subtilités sur les réseaux torrent en théorie et aussi des signatures extraites de notre cas pratique, c'est à dire des protocoles comme le Bittorrent.

Dans la partie pratique nous avons pu tester la détection à base de Snort avec les différentes règles conçues lors de l'analyse.

Fort de notre expérience avec des tests concluants, nous avons ainsi lister un certain nombre de mesures pour pouvoir garder un réseau saint :

•Mettre un IDS réseau en place et faire des mises à jour régulière afin de garder une bonne base de données sur les signatures.

•implémenter un HIDS au niveau des machines pour bloquer l'installation de client BitTorrent.

•Faire une campagne de sensibilisation auprès des employés de l'entreprise.

•Empêcher l'utilisation de proxy et de VPN autres que ceux de l'entreprise.

•mettre des sanctions pour les tentatives d'entorses à ces mesures.

Notre sujet d'étude nous a permis d'approfondir nos connaissances en réseau en générale, des protocoles, et nous a exigé une connaissance détailler sur les architectures P2P et le torrent, d'avoir une vision claire de l'analyse des données et de découvrir le monde des NIDS.

Bibliographie

[1] Bruno Péan : Introduction aux réseaux informatiques, Miage,2007.

[2] G.Santini et J.-C.Dubacq : Introduction à l'informatique, IUTdeVilletaneuse, 2016.

[3] https://www.frameip.com.

[4] Support technique, Cisco System.

[5] Sécurité des ordinateurs, http://assiste.com.free.fr/index.html.

[6] Claude Duvallet : Les systèmes de détection d'intrusion réseau, Université du Havre UFR Sciences et Techniques.

[7] Cisco « Cisco Certified Network Associate 1 » version 6: Notions de base sur les réseaux.

[8] L'échange et le partage de fichiers numériques, Campustic 21, Internet pour éduquer au développement durable.

[9] Réseaux Pair à Pair, cours ENS Lyon, M1.

[10] Le Cocq Michel : Applications Client/serveur et Web, Licence Pro SIL,2017.

[11] Nathalie BUDAN et Benoit TEDESCHI : Nouvelles Technologies Réseau,Les réseaux peer-to-peer,2003.

[12] MOUALKIA Yamina : Performance et Optimisation des Architectures P2P pour les Applications des Réseaux Sociaux, Mémoire de master, 2016.

[13] Abdelli Nabila et Ait Ouali Kayssa : Découverte et Localisation des Usagers dans un Réseau Peer To Peer , Mémoire d'Ingénieur, université de Bejaïa, 2009.

[14] Ouchene sabrina et Yaiche Soria, l'étude de l'architecture hybride pour la découverte et la localisation des services en mode Peer-to-Peer, Mémoire d'Ingénieur, université de Bejaïa, 2007.

[15] Dmitri Moltchanov : Client/serveur and peer-to-peer models: basic concepts, Department of Communications Engineering Tampere University of Technology, 2013.

[16] Jérôme. G : BitTorrent Inc, https://www.generation-nt.com

[17] Choon Hoong Ding et Sarana Nutanong : Peer-to-Peer Networks for Content Sharing , Department of Computer Science and Software Engineering, The University of Melbourne, Australia.

[18] Bourlier Gérard, " Le Peer to Peer, Réseau de poste à poste ", Mémoire de Master, université de Montpellier, 2004.

[19] Thawte, Mieux comprendre les certificats SSL,2011
http://www.thawte.fr/ssl/index.html.

[20] Nathalie BUDAN et Benoit TEDESCHI : Nouvelles Technologies Réseau,Les réseaux peer-to-peer,2003.

[21] Jonas FERNANDEZ et SHARK EURL MAKO : SQUID Contrôlez et accélérez le surf, Association LOLITA Logiciels Libres à Tahiti & ses îles.

[22] www.wireshark.org.

[23] Les sondes de sécurité IDS/IPS, Thibault PALUD, Mars 2010, institut d'électronique et d'informatique Gaspard Monge.

[24] Manuel Snort, www.Snort.org.

[25] Noah Dietrich : The Reputation Preprocessor in Snort – Blacklists and Whitelists, Snort Technology 2015.