الجمهورية الجزائرية الديمقراطية الشعبية
République algérienne démocratique et populaire

وزارة التعليم العالي والبحث 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 Réseaux et Télécommunications

Présenté par

AMALOU Warda

&

DELLAL Marwa

Elaboration d'une solution de détection du réseau anonyme Orbot

Proposé par : Mr.MEHDI Merouane

Année Universitaire 2019-2020

Remerciements

En guise de reconnaissance, nous tenons à témoigner nos sincères remerciements à toutes les personnes qui ont contribués de près ou de loin au bon déroulement de notre mémoire de fin d'étude et à l'élaboration de ce travail.

Tout d'abord nous voudrions présenter nos sincères remerciements à notre encadreur « Mr. MEHDI Merouane », et nous voudrions également lui témoigner notre gratitude pour son soutien et ses conseils qui nous ont été très précieux afin de mener notre travail à bon port, Merci.

Nos vifs remerciements vont également au prestigieux membre du jury et à tout le staff du département de Génie Electrique de l'université de SAAD DAHLAB Blida 1, très spécialement nos professeurs durant tout ce cycle de cinq ans.

Enfin, nous remercions chaleureusement nos chers parents pour leur soutien et leur patience, qui ont étaient toujours là dans les moments difficiles.

Dans l'impossibilité de citer tous les noms, nos sincères remerciements vont à tous ceux et celles, qui de près ou de loin, ont permis par leurs conseils et leurs compétences la réalisation de ce mémoire.

ملخص: يسمح لنا هذا المشروع بكشف استعمال تطبيق أوربوت لأجهزة أندرويد. يتيح أوربوت تمرير بيانات تطبيقاته من خلال شبكة تور مما يتيح تغطية الأعمال غير القانونية مثل هجمات الكمبيوتر، وشراء وبيع المخدرات والأسلحة، الخ. من خلال الوصول إلى الشبكة المظلمة.

يتم كشف هذا التطبيق بمقارنة تدفق البيانات الخاصة به ببيانات الويب العادي لجمع بصمات رقمية تسمح بالتعرف على التطبيق. يتم وضع هذه البصمات الرقمية في نظام كشف التسلل ومنع اختراق الشبكات "سنورت" لإطلاق تنبيه في كل محاولة للاتصال بهذا التطبيق.

كلمات المفاتيح: أوربوت، شبكة تور، كشف، تدفق بيانات الويب، بصمات رقمية، نظام كشف التسلل ومنع اختراق. الشبكات، سنورت

Résumé : Ce projet consiste à détecter l'utilisation de l'application Orbot pour les appareils Android. Orbot permet de faire passer le trafic de ses applications par le réseau Tor ce qui permet de couvrir des actions illégales comme les attaques informatiques, achat et vente de la drogue, armes …etc. En accédant au Dark Web.

La détection de cette application se fait en comparant son trafic avec le trafic Web normale afin de relever des empreintes numériques qui identifient cette application. Ces empreintes vont être implémentées dans un IDS « Snort » pour déclencher une alerte à chaque tentative de connexion à cette application.

Mots clés : Orbot ; réseau Tor ; détection ; trafic Web ; empreintes digitales ; IDS ; Snort.

Abstract: This project allows us to detect the use of the Orbot app for Android devices. Orbot allows the traffic of its applications to pass through the Tor network, which makes it possible to cover illegal actions such as computer attacks, buying and selling drugs, weapons, etc. By accessing the Dark Web.

The detection of this application is done by comparing its traffic with normal web traffic in order to collect digital fingerprints that identify this application. These fingerprints will be implemented in a "Snort" IDS to trigger an alert on each attempt to connect to this application.

Keywords: Orbot; Tor network; detection; Web traffic; digital fingerprints; IDS; Snort.

Liste des acronymes et abréviations

ACK: ACKnowledge Character.

ADOdb: Activate Data Objects Data Base.

AES: Advanced Encryption Standard.

APK: Android Package Kit.

ARP: Address Resolution Protocol.

CA: Certificate authority**.**

CBC: Cipher Block Chaining.

CENTOS: Community enterprise Operating System.

CMD: Windows Command Prompt.

CPU: central processing unit.

CRM: Customer Relationship Management.

DARPA: Defense Advanced Research Projects Agency.

DDOS: Distributed Denial of Service.

DHE: Diffie-Hellman Ephemeral.

DOS: Denial of Service.

DPI: Deep Packet inspection.

ECDHE: Elliptic Curve Diffie-Hellman.

EFF : Electronic Frontier Foundation.

FAI : Fournisseur d'Accès à Internet.

FFDHE: Finite Field Diffie-Hellman Ephemeral Parameter.

FreeBSD: Berkeley Software Distribution.

GCC: Gulf Cooperation Council**.**

GNU GPL: GNU General Public License.

HIDS: Host Intrusion Detection System.

HTML: HyperText Markup Language.

HTTP: HyperText Transfer Protocol.

HTTPS: HyperText Transfer Protocol Secure.

IBM: International Business Machines.

ICMP: Internet Control Message Protocol.

ID: Identifiant.

IDS: Intrusion detection System.

IEEE: Institute of Electrical and Electronics Engineers.

IP: Internet protocol.

MITM: Man In The Middle.

MSS: Maximum Segment Size.

NIDS: Network Intrusion Detection System.

NMAP: Network Mapper.

PC: Personal Computer.

PCAP : Packet Capture.

PDF : Portable Document Format.

PHP: HyperText Preprocessor**.**

PSH: Push**.**

RHEL: Red Hat Enterprise Linux.

RPM: REDHat Package Manager.

RSA: Rivest Shamir Adleman.

RST: Reset.

RTT: Round Trip Time.

SHA: Secure Hash Algorithm.

SQL: Structured Query Language.

SSL: Secure Sockets Layer.

SYN: synchronize.

SYN-ACK: acknowledgment synchronize.

TCP: Transmission Control Protocol.

TLS: Transport Layer Security.

TOR: The Onion of Routing.

TTL: Time To Live.

URG: Urgent Flag.

URL: Uniform Resource Locator.

VPN: Virtual Private Network.

WEB: World Electronic Base.

WIFI: Wireless Fidelity.

XSS : Cross Site Scripting.

Table des matières

Liste des figures

Liste des tableaux

Introduction générale

Face à l'évolution gigantesque des réseaux informatiques dans le domaine professionnel ainsi que pour les individus, beaucoup de problèmes se posent devant cette croissance. Pour cela on se trouve face au défi de se collaborer à la recherche des solutions plus fiables pour se protéger contre ces problèmes qui sont devenus diversifiés. L'un de ces problèmes traités dans ce projet est la piraterie. Les pirates informatiques cherchent toujours à trouver des empreintes qui décrivent les activités des internautes et précisent leurs emplacements.

Pour protéger notre confidentialité ainsi que notre vie privée, plusieurs mécanismes de sécurité qui vont nous assurer l'anonymat vont être détaillés dans les chapitres suivants, comme Proxy, VPN et les réseaux anonymes comme le fameux réseau Tor.

Tor est un réseau anonyme gratuit et accessible par tout le monde. L'utilisation de ce réseau protège la vie privée des utilisateurs, leur liberté et leur capacité d'effectuer des communications confidentielles sans être surveillées. Tor est encore plus utilisé par les cybercriminels afin de couvrir leurs activités illégales.

Avant les smartphones, à chaque fois que les utilisateurs voulaient naviguer sur Internet en toute sécurité, ils comptaient sur Tor Browser sur leur PC. Mais, on remarque ces dernières années que le taux d'utilisation de Smartphones ne cesse d'évoluer. Les internautes choisissent d'utiliser les Smartphones grâce à la mobilité et le confort d'utilisation qu'ils offrent. Donc ils ont besoin d'une alternative du navigateur Tor pour leur mobile, c'est là qu'intervient Orbot. La même équipe qui était derrière Tor fournit également une application conçue pour travailler sur Android pour permettre aux utilisateurs de se connecter anonymement.

Orbot est une excellente technologie pour les pirates qui veulent entrer dans votre réseau et exfiltrer les données sensibles sans laisser une trace menant directement à leur emplacement.

L'objectif de notre projet est de trouver une solution contre la mauvaise utilisation de cette application.

Avant d'entamer notre projet, nous allons nous intéresser aux réseaux anonymes (Tor, Orbot) et les différentes menaces qu'ils peuvent causer (attaques informatique, vols d'informations...etc.). Par la suite on proposera une technique de détection de l'utilisation de cette application en passant par plusieurs étapes :

1. Étude du trafic de ce réseau se fera grâce à l'analyseur de paquet Wireshark.
2. Relèvement des empreintes.
3. Détection de l'utilisation de Orbot se fera à l'aide d'un NIDS « Network Intrusion Detection System » en utilisant une variété de signatures.

1.1 Introduction

Lorsqu'on parle de la sécurité des réseaux informatiques, des questions importantes se posent ? Pourquoi sécuriser un réseau ? Contre qui ? Et comment ?

Le progrès technologique avance d'une vitesse effrénée, mais cette technologie contient toujours plus ou moins des erreurs de conception ou de configuration, cela conduit à les exploiter par des personnes mal intentionnées et par conséquent les attaques et les actes de malveillance informatique sont devenus plus fréquents et plus dangereux. C'est là où entre en jeu l'aspect de la sécurité informatique qui est devenue une préoccupation importante des utilisateurs et des entreprises et un enjeu vital ces dernières années.

A travers ce projet nous allons voir comment un attaquant procède pour pénétrer un réseau.

Il est important de noter que ce chapitre présente des définitions globales, des schémas explicatifs, des explications, des méthodes, des outils et en dernier des techniques et des mécanismes de sécurité qui sont nécessaires à la mise en place d'une politique de sécurité des réseaux d'entreprise.

Le risque est toujours présent car les attaques sont de plus en plus fréquentes. Pour mieux se protéger, il est primordial de savoir à quoi s'attendre, et donc de connaitre à minima les attaques informatiques les plus courantes.

1.2 La sécurité réseau

La sécurité informatique protège l'intégrité des technologies de l'information comme les systèmes, les réseaux et les données informatiques contre les attaques, les dommages ou les accès non autorisés.

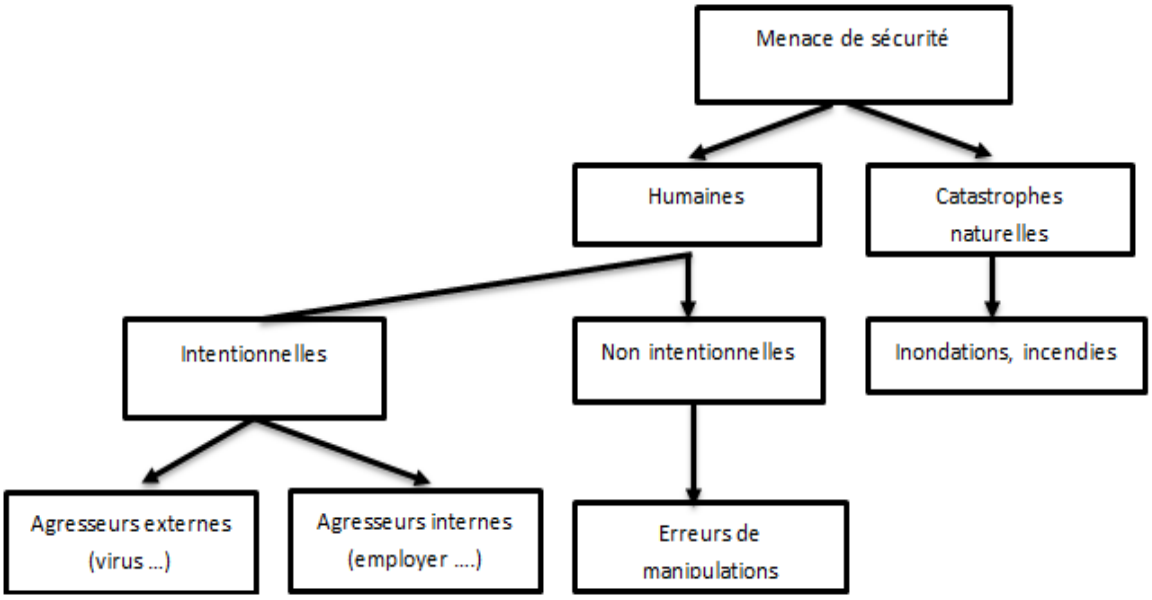


Figure 1.1. Les différentes menaces de sécurité.

1.2.1 Services de Sécurité

La sécurité informatique se base sur un certain nombre de services qui permettent de mettre en place une réponse appropriée à chaque menace.

Décrivons les principaux services de sécurité :

- **Confidentialité :** les données (et l'objet et les acteurs) de la communication ne peuvent pas être connues d'un tiers non-autorisé.
- **Authenticité :** l'identité des acteurs de la communication est vérifiée.
- **Intégrité :** les données de la communication n'ont pas été altérées.
- **Non-répudiation** : les acteurs impliqués dans la communication ne peuvent nier y avoir participer.
- **Disponibilité :** les acteurs de la communication accèdent aux données dans de bonnes conditions.

1.3 Les attaques et leurs motivations

Tout ordinateur connecté à un réseau informatique est potentiellement vulnérable à une attaque. Une attaque est l'exploitation d'une faille d'un système informatique (système d'exploitation, logiciel, erreur de configuration, etc.) à des fins non connues par l'exploitant du système et généralement préjudiciables.

Sur internet, des attaques ont lieu en permanence, à raison de plusieurs attaques par minute sur chaque machine connectée. Ces attaques sont pour la plupart lancées automatiquement à partir de machines infectées (par des virus, chevaux de Troie, vers, etc.), à l'insu de leur propriétaire. Plus rarement, il s'agit de l'action de pirates informatiques.

Les motivations des attaques peuvent être de différentes sortes :

- Obtenir un accès au système.
- Glaner des informations personnelles sur un utilisateur.
- Récupérer des données bancaires.
- Troubler le bon fonctionnement d'un service.
- Utiliser le système de l'utilisateur comme « Rebond » pour une attaque.
- Utiliser les ressources du système de l'utilisateur, notamment lorsque le réseau sur lequel il est situé, possède une bande passante élevée [1].

Un scénario d'intrusion sur un système peut se décomposer en six actions élémentaires, enchaînées selon un processus itératif :

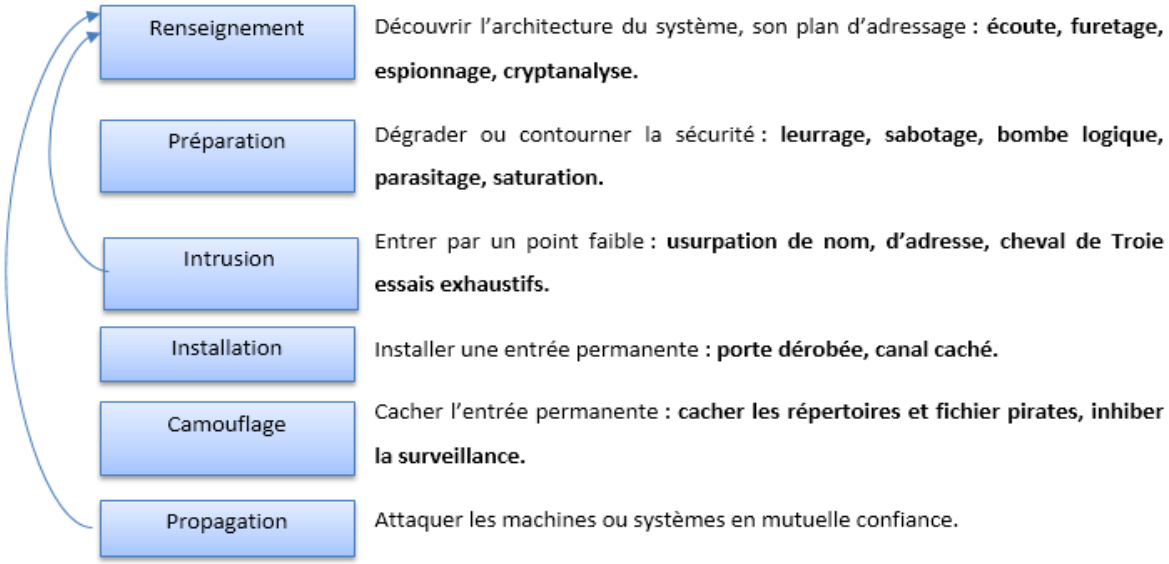


Figure 1.2. Déroulement d'une attaque.

On cite des exemples réels des attaques pour les dix dernières années :

- **L'onde de choc Stuxnet**

Découvert en 2010, le ver informatique Stuxnet a fait de gros dégâts, notamment dans des installations nucléaires iraniennes, ceux de la centrale de la ville de Natanz. Ce virus informatique n'était pas destiné à voler des données, mais à espionner et à saboter des systèmes industriels. Plusieurs agences spécialisées en cybersécurité se sont penchées sur les origines et la conception de Stuxnet, avant de finir par déduire que ce malware avait été conçu par les États-Unis et Israël pour freiner le programme nucléaire iranien.

Avec Stuxnet, une nouvelle ère est née : celle de la cyberguerre. C'est en effet avec ce virus en particulier que beaucoup d'experts, de gouvernement, d'entreprises ou même de particuliers ont pris conscience qu'un logiciel malveillant pouvait avoir le pouvoir de déclencher ou d'enrayer des conflits mondiaux [2].

- **WannaCry, le roi des ransomware**

En mai 2017, le ransomware WannaCry fait une arrivée fracassante sur Internet, après son utilisation dans une cyberattaque mondiale ayant touché plus de 300 000 ordinateurs, dans plus de 150 pays. À l'époque, on commence tout juste à parler des ransomwares, ces logiciels malveillants capables de prendre des données en otage en les chiffrant, pour ensuite

demander à l'utilisateur de la machine le paiement d'une « rançon » en vue de débloquer l'accès aux fichiers.

Pour s'introduire dans les ordinateurs, les pirates à l'origine de WannaCry utilisaient la bonne vieille méthode de l'ingénierie sociale, via des envois massifs de mails chargés d'une pièce jointe piégée. De nombreuses entreprises en ont été victimes, parmi lesquelles Vodafone, Fedex, Renault, la Deutsche Bahn, mais aussi le ministère de l'intérieur russe [2].

- **L'attaque DDOS contre GitHub**

En février 2018, la plateforme GitHub a été victime de la plus grande attaque DDOS survenue jusqu'alors. Ses serveurs ont été saturés par des centaines de milliers de requêtes à la seconde, entraînant la chute de ces derniers.

Heureusement pour GitHub, ses serveurs de Backup ont pris le relai quelques minutes seulement après la chute de ses serveurs principaux. L'attaque a été spectaculaire, mais n'a finalement causé aucun dégât matériel, et s'est contentée d'égratigner un peu l'égo de l'entreprise ciblée. Comme souvent, GitHub a été pris pour cible sans raison particulière : les pirates informatiques à l'origine de l'attaque voulaient simplement faire la démonstration de leurs talents [2].

1.3.1 Catégories d'attaques

a. Les attaques passives

Les attaques passives consistent à écouter et à surveiller les transmissions sur un canal. Le but de l'attaquant est d'intercepter les informations transmises. Elles sont généralement indétectables mais une prévention est nécessaire.

b. Les attaques actives

Les attaque actives, comme leur nom l'indique, vont conduire l'attaquant à agir sur le système attaqué, à modifier son fonctionnement ou bien à attenter à son intégrité physique [3].

1.3.2 Anatomie d'une attaque

Fréquemment appelés « les 5 P » dans la littérature, ces cinq verbes anglophones constituent le squelette de toute attaque informatique : Probe, Penetrate, Persist, Propagate, Paralyze. On les présente par le simple schéma ci-dessous.

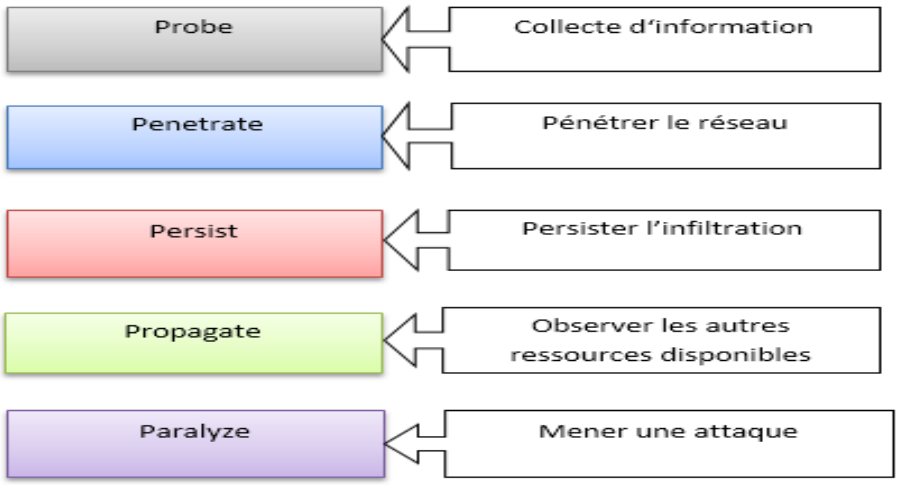


Figure 1.3. Anatomie d'une attaque.

Après ces cinq étapes, le pirate peut éventuellement tenter d'effacer ses traces.

1.3.3 Buts des attaques

- **Interruption :** vise la disponibilité des informations (DOS, . . .).
- **Interception :** vise la confidentialité des informations (capture de contenu, analyse de trafic, . . .).
- **Modification :** vise l'intégrité des informations (modification, rejeu, . . .).
- **Fabrication :** vise l'authenticité des informations [4].

1.4 Déroulement d'une attaque

1.4.1 La collecte des informations

a. La prise d'empreinte par le TCP/IP

La prise d'empreinte en interrogeant la pile TCP/IP est effectuée en utilisant des scanneurs de ports comme **Nmap**. Il permet d'en connaitre les ports ouverts et les IP actives

sur le réseau, et donc probablement de connaitre les services lancés sur chaque machine, de connaitre leurs versions et potentiellement les vulnérabilités [5].
Il existe d'autres types de scanneurs de ports :

- **Les mappeurs passifs** : comme le logiciel **Siphon**. Ils permettent de déterminer la topologie réseau du brin physique sur lequel est connectée la machine depuis laquelle l'exécutable est lancé, mais ils sont surtout utilisés car ils sont indétectables par les machines cibles, puisqu'ils n'envoient pas de paquets [5].

b. Les renifleurs

Les renifleurs se présentent sous plusieurs formes. Il existe des renifleurs de paquets, des renifleurs Wi-Fi, des renifleurs réseaux et des renifleurs d'IP. Mais ils ont tous un point commun : un renifleur est un type de logiciel qui intercepte tout le trafic entrant et sortant d'un ordinateur connecté à un réseau [2].

Les pirates informatiques utilisent les renifleurs pour voler des données, espionner les activités réseau et recueillir des informations sur les utilisateurs. Habituellement, le but final est d'obtenir les mots de passe et les informations sur les comptes des sites de transactions bancaires et d'achats. En général, les pirates informatiques placent les renifleurs dans des lieux qui offrent des connexions Wi-Fi non sécurisées comme celles trouvées dans les cafés, les hôtels et les aéroports. Les renifleurs sont également utilisés pour emprunter l'identité d'autres appareils sur le réseau, dans ce qui est connu comme une attaque d'usurpation, afin de voler des informations sensibles [6].

c. Failles dans la sécurité des réseaux

Dans le domaine de la sécurité informatique, une vulnérabilité ou faille est une faiblesse dans un système informatique permettant à un attaquant de porter atteinte à l'intégrité de ce système, c'est-à-dire à son fonctionnement normal, à la confidentialité ou à l'intégrité des données qu'il contient. Mais d'où peut venir la faille dans la sécurité ?

Il existe deux réponses possibles :

- La faille peut venir de l'intérieur du réseau (on peut parler de « faille active »).

- La faille peut être provoquée depuis l'extérieur du réseau (on peut parler de « faille passive »).

- **Repérage des failles**

Il existe des scanneurs de vulnérabilités, comme **Nessus** ou **Saint**, auxquels on peut soumettre un réseau pour un test d'intrusion. Le logiciel en ressort alors les failles connues (la discrétion n'est pas vraiment le fort de ces logiciels, puisqu'ils vont tester les failles connues en masse). Il est donc préférable d'interroger plutôt les bases de données en ligne telles que sur le site **Security Focus** qui met à jour régulièrement sa base de données de vulnérabilités [5].

Note : Une fois que l'attaquant a récupéré un maximum d'informations sur l'architecture du réseau, et qu'il a collecté des informations disponibles (chaque information a son importance), il a une vision globale de la façon dont le réseau fonctionne. En fonction des informations en sa possession, il peut s'insérer dans le réseau et entamer son attaque.

1.4.2 Les différents types d'attaques informatiques

a. Attaque par des logiciels malveillants (Malwares)

Un malware est un logiciel indésirable installé dans votre système sans votre consentement. Il en existe plusieurs types, mais en voici quelques-uns :

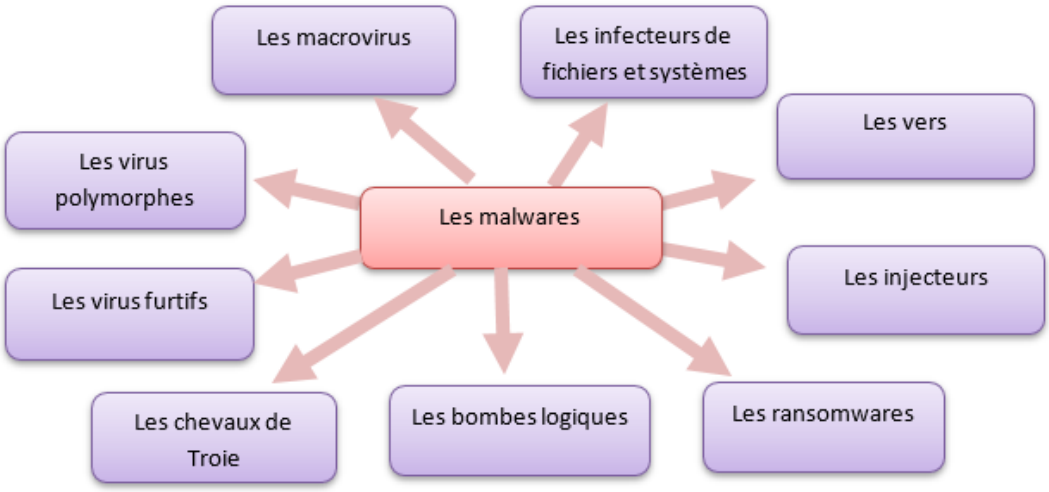


Figure 1.4. Les logiciels malveillants.

b. *Attaques cryptographiques*

Afin de réaliser une attaque cryptographique, on doit étudier des procédés cryptographiques dans le but de trouver des faiblesses, en particulier, de pouvoir décrypter des messages chiffrés. Le décryptement est l'action consistant à trouver le message en clair sans connaitre la clef de déchiffrement.

- **Les attaques par mot de passe**

Trouver un mot de passe est souvent bien plus facile qu'il n'y paraît, et les pirates s'en donnent à cœur joie. Pour trouver un mot de passe, il suffit parfois simplement de fouiller un bureau, en surveillant la connexion pour obtenir un mot de passe non chiffré, en ayant recours à l'ingénierie sociale ou en devinant :

- **Par force brute :** deviner un mot de passe en entrant ce que les gens entrent le plus souvent : nom, prénom, passe-temps favori, dates de naissance des enfants, etc.
- **Par dictionnaire :** cela consiste à copier un fichier chiffré contenant des mots de passe courants et à comparer les résultats.

- **L'attaque Homme du milieu (Man-in-the-Middle ou MitM)**

L'attaque « Homme du milieu » est un type d'attaque dont le principe est de s'insérer dans les communications entre un serveur et un client.

Figure 1.5. L'attaque Homme du milieu [7].

Il en existe plusieurs :

- **Le détournement de session (Hijacking) :** un attaquant détourne une session entre un client de confiance et un serveur réseau. L'attaquant substitue l'adresse IP du client pendant que le serveur continue la session, croyant que c'est toujours le client.
- **L'usurpation de l'adresse IP :** le pirate peut utiliser une adresse IP volée pour convaincre un système qu'il est un client fiable et connu.
- **Le Replay (attaque par rejeu) :** une attaque Replay se produit lorsqu'un attaquant intercepte et enregistre des paquets de données et tente plus tard de les envoyer, c'est-à-dire les retransmettre tels quels (sans aucun déchiffrement) au serveur destinataire.
- **ARP Spoofing :** l'ARP Spoofing désigne les attaques de l'homme du milieu sur les tables ARP. Dans ce type d'attaque, les pirates envoient de faux paquets ARP afin de passer inaperçu entre deux systèmes communicants, pour intercepter ou manipuler leur trafic de données.

c. Les attaques DOS ou attaques par déni de service

Une attaque par déni de service (attaque DOS) est une attaque destinée à empêcher un système ou un service de fonctionner normalement. Elle peut exploiter une vulnérabilité connue dans un système d'application ou d'exploitation spécifique, ou utiliser certaines vulnérabilités dans des protocoles ou des services spécifiques. Dans une attaque DOS, l'attaquant tente d'empêcher les utilisateurs autorisés d'accéder, soit à certaines informations spécifiques, soit au système informatique ou au réseau lui-même. Cela peut être accompli en provoquant un arrêt intempestif du système pour le mettre hors ligne, ou en l'inondant de demandes [8].

Figure1.6. L'attaque DOS [9].

- **Attaque TCP/SYN Flooding**

Une connexion TCP s'établie en trois phases selon le mécanisme de poignée de main en trois temps (Three Ways Handshake). Ces trois étapes sont :

- **L'envoi d'un SYN** (demande de synchronisation).
- **La réception d'un SYN-ACK** (synchronisation-ACknowledgment).
- **Et l'envoi d'un ACK** (ACknowledgment).

Le principe est de laisser sur la machine cible un nombre important de connexions TCP en attentes. Pour cela, le pirate envoie un très grand nombre de demandes de connexion (requête SYN), la machine cible renvoie les SYN-ACK (synchronisation-ACknowledgment) en réponse à la requête SYN reçue.

Le pirate ne répondra jamais avec un ACK (ACknowledgment), et donc pour chaque SYN reçu la cible aura une connexion TCP en attente. Etant donné que ces connexions semi-ouvertes consomment des ressources mémoires au bout d'un certain temps la machine est saturée et ne peut plus accepter de connexion. Ce type de déni de service n'affecte que la machine cible.

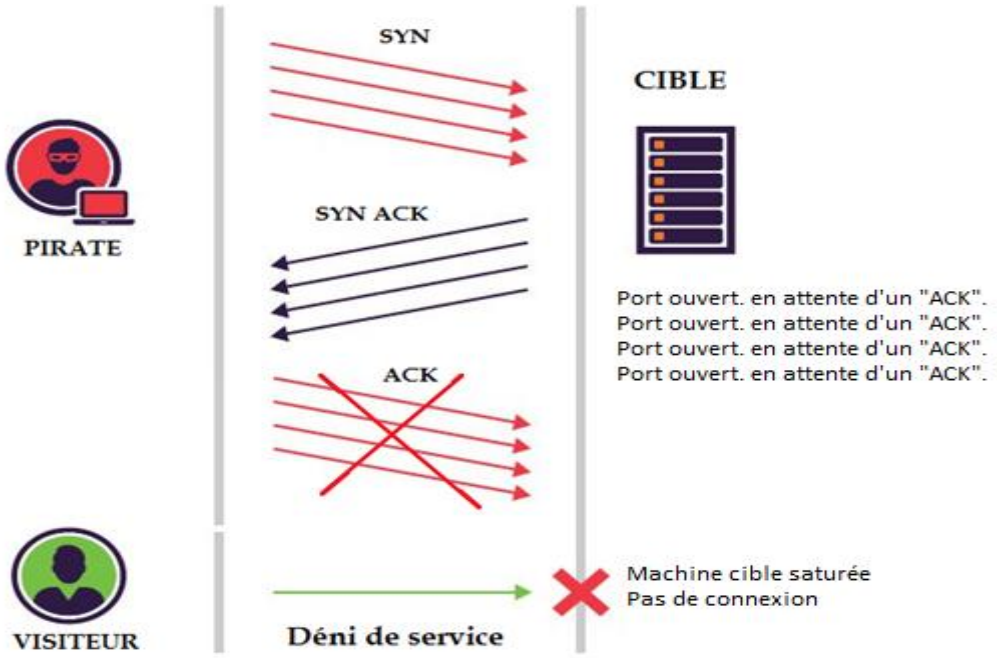


Figure 1.7. Attaque TCP/SYN Flooding [10].

- **Attaque par fragmentation (Teardrop)**

Une attaque par fragmentation (en Anglais Fragment Attack) est une attaque réseau par saturation (déni de service) exploitant le principe de fragmentation du protocole IP.

En effet, le protocole IP est prévu pour fragmenter les paquets de taille importante en plusieurs paquets IP possédant chacun un numéro de séquence et un numéro d'identification commun. A la réception des données, le destinataire réassemble les paquets grâce aux valeurs de décalage (en anglais Offset) qu'ils contiennent.

L'attaque par fragmentation la plus célèbre est l'attaque Teardrop. Le principe de l'attaque Teardrop consiste à insérer dans des paquets fragmentés des informations de décalage erronées. Ainsi, lors du réassemblage il existe des vides ou des recoupements (Overlapping), pouvant provoquer une instabilité du système (DOS) [11].

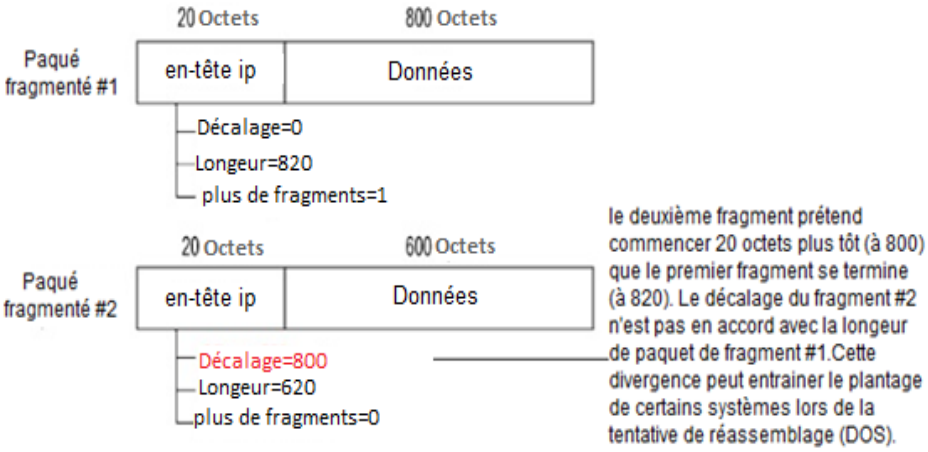


Figure1.8. L'attaque par fragmentation [12].

- **L'attaque ICMP Flood**

Le principe de cette attaque de type DOS est de créer des connexions vers la cible très rapidement, plus rapidement que la cible ne peut les traiter.

La méthode consiste à l'envoi d'une multitude de paquets (ICMP ECHO-request), une requête très courte et très courante faite à un serveur, (en modifiant l'adresse source de

chaque paquet). La machine cible est saturée par cette multitude de petits paquets et ne peut plus répondre aux demandes de connexions car l'ensemble de la bande passante est utilisé.

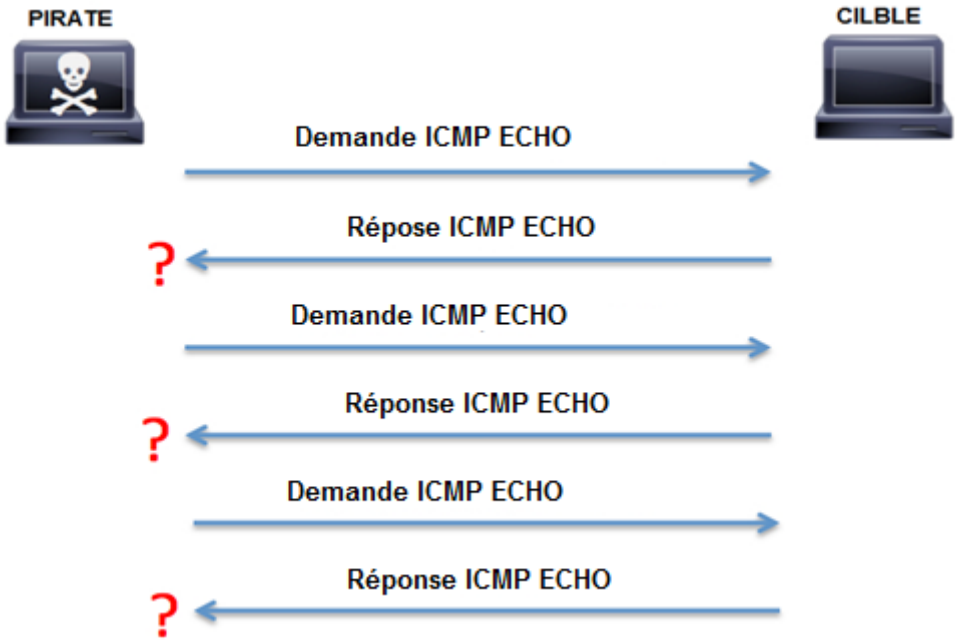


Figure 1.9. L'attaque ICMP Flood [13].

- **L'attaque Ping de la mort (Ping of Death)**

Le principe est d'envoyer un paquet ICMP avec une quantité de données supérieure à la taille maximale d'un paquet IP. Encore une fois, la pile IP peut s'avérer incapable de gérer cette exception et le reste du trafic.

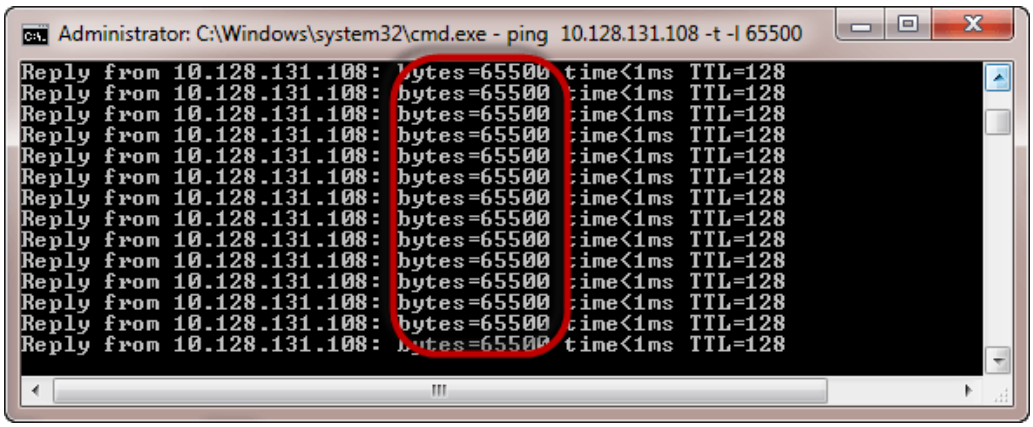


Figure 1.10. L'attaque Ping de la mort [14].

- **L'attaque par réflexion (Smurf)**

Cette variante consiste à envoyer une trame à la destination d'un Broadcast de réseau. La finalité est de coupler cette trame à une adresse IP source correspondante à celle de la cible. Ainsi, le flux de réponse en destination de la cible sera fortement multiplié.

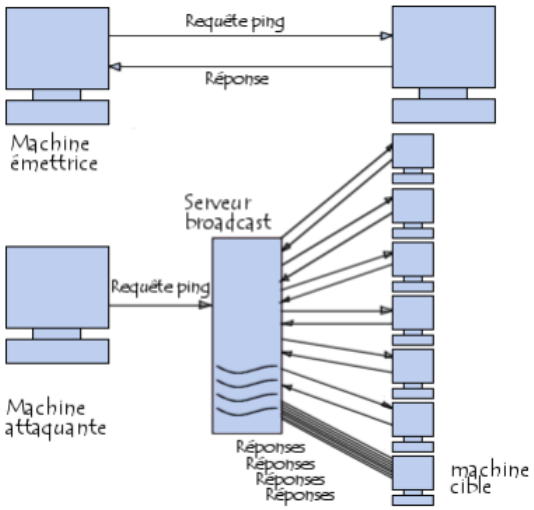


Figure 1.11. L'attaque par réflexion [15].

- **Attaque Land (Land Attack)**

Cette attaque DOS consiste à démarrer une ouverture de session TCP via un SYN à destination d'un port ouvert de la machine cible. L'astuce de l'attaque est de préciser l'adresse IP source identiquement à l'IP destination ainsi que le port source identiquement au port destination. La victime recevant cette trame pense alors qu'elle discute avec lui-même ce qui généralement provoquait un plantage [16].

d. Les attaques DDOS ou attaques par déni de service distribué

Le déni de service distribué (DDOS pour Distributed DOS) est une attaque de Déni de service émise depuis plusieurs origines distinctes. Ce type d'attaque est extrêmement complexe à bloquer, car il est souvent impossible de différencier une vraie requête d'une requête de DDOS. L'attaque par DDOS utilise très souvent une multitude de PC Zombies infectés par des Backdoors exploités à distance par un pirate et attaquant simultanément une cible unique.

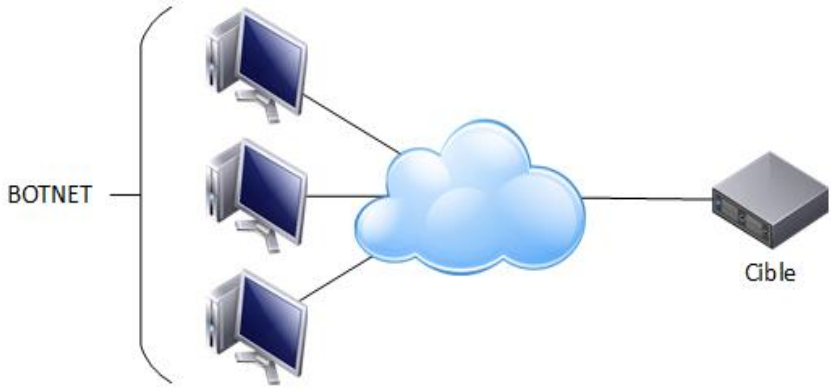


Figure 1.12. L'attaque DDOS [9].

Une attaque par déni de service distribué se déroule généralement en deux étapes :

- Infection de machines qui vont être utilisées pour inonder la cible, ces machines constituent ce que l'on appelle le réseau de machines Zombies (ou « Botnet »).
- Déclenchement de l'attaque, le Botnet inonde alors la cible de façon synchrone.

Plus la taille du Botnet est importante, plus l'ampleur de l'attaque sera grande. Les effets sur les services visés seront donc plus impactant pouvant même aller jusqu'à les rendre totalement indisponibles [9].

- **Les portes dérobées (Backdoors)**

Les portes dérobées sont des accès cachés sur un système ou sur une application. Les pirates les installent sur les ordinateurs en utilisant généralement un logiciel malveillant.

Cette porte dérobée leurs permet ensuite d'en prendre un contrôle partiel, voire total :

- Vol, modification ou suppression de fichiers, de documents personnels.
- Installation de nouveaux logiciels malveillants.

- **Les Zombies (Botnet)**

Un Botnet est un réseau malveillant constitué de machines compromises (encore dénommées « agents », « Bots » ou « machines Zombies »), contrôlées à distance par une ou plusieurs entités et permettant une gestion distribuée d'actions offensives ou malveillantes [17].

1.4.3 Les vulnérabilités web

a. Le téléchargement furtif (Drive-by Download)

Les attaques par téléchargement furtif sont des téléchargements qui se produisent à l'insu ou sans le consentement d'un utilisateur. Après le téléchargement, l'application est invoquée et libre de remplir ses fonctions néfastes. La simple visite d'un site Web malveillant peut entraîner le téléchargement et l'exécution ultérieure de logiciels malveillants sur l'ordinateur d'un visiteur [18].

b. Injection SQL

L'attaque par injection SQL (injection de code) est le type de technique de vulnérabilité le plus courant et le plus simple adopté par les attaquants Web via des applications Web. En utilisant des commandes SQL simples tels que Select, Where, Insert, Delete et Update, les attaquants malveillants restructurent efficacement le Code SQL (instructions) et exécute le code vulnérable dans les applications Web. Une fois les attaquants atteignent leur objectif, ils peuvent facilement accéder aux informations sensibles, modifier les données sécurisées et les exécuter [19].

Figure 1.13. Injection SQL [20].

c. Injection de code malicieux (Cross Site Scripting (XSS))

Les vulnérabilités XSS sont causées par une validation incorrecte des données d'entrée (par exemple, provenant de l'utilisateur). Les données d'entrée peuvent contenir des fragments HTML susceptibles de s'afficher sur la page Web et d'altérer le contenu résultant

de telle sorte qu'un code malveillant soit injecté. Lorsqu'il est exécuté par le navigateur de l'utilisateur, le code injecté peut divulguer des données sensibles à des tiers [21].

Les attaques XSS n'affectent pas le côté serveur, ils affectent seulement le côté clients.

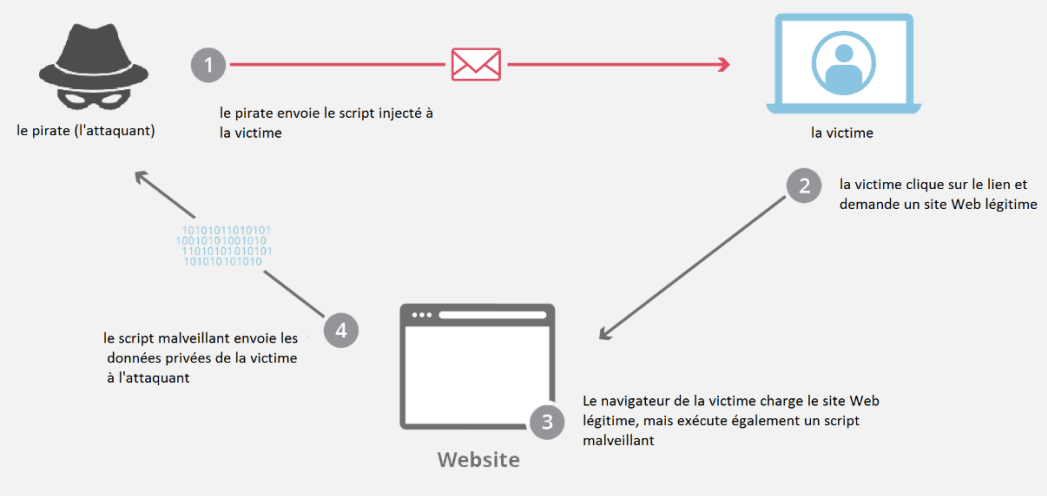


Figure 1.14. Injection de code malicieux [22].

1.4.4 Extension de privilège

Le pirate ayant obtenu quelques accès peu protégé, tente maintenant d'augmenter ses privilèges sur la cible en obtenant un/des accès root (super administrateur). On parle ainsi d'extension de privilège.

Une fois les accès root obtenu, le pirate peut obtenir des informations supplémentaires. Cela lui permettra d'avoir par exemple d'autres comptes utilisateur du réseau.

1.4.5 Nettoyage des traces

a. Les Rootkits

Les Rootkits fonctionnent en remplaçant les commandes du système d'exploitation par des versions modifiées spécialement conçues pour ignorer les activités malveillantes afin qu'elles puissent échapper à la détection [23].

Un Rootkit se limite généralement à l'ordinateur sur lequel il est installé et ne cherche pas en soi à se propager à d'autres ordinateurs [23].

1.5 Mécanismes de sécurité

1.5.1 Cryptage

La cryptographie est une science mathématique dans laquelle on fait les études des méthodes permettant de transmettre des données de manière confidentielle.

Afin de protéger un message, on lui applique une transformation qui le rend incompréhensible, c'est ce qu'on appelle le chiffrement. Qui, à partir d'un texte clair, donne un texte chiffré ou cryptogramme. Inversement, le déchiffrement est l'action qui permet de reconstruire le texte en clair à partir du texte chiffré. Dans la cryptographie moderne, les transformations en question sont des fonctions mathématiques, appelées algorithmes cryptographiques, qui dépendent d'un paramètre appelé clef [24].

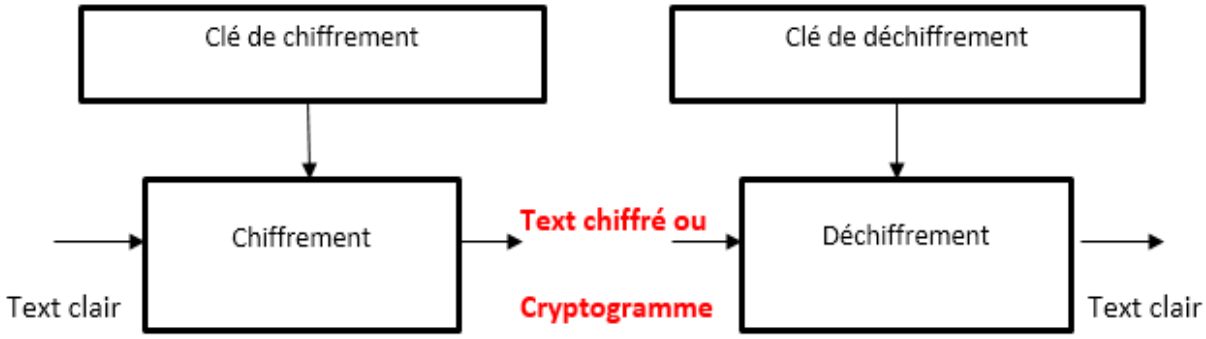


Figure 1.15. Cryptage et chiffrement [24].

Dans les réseaux, pour réduire les vols d'informations dans la voie de transmission, on utilise les techniques de cryptographie pour chiffrer et déchiffrer les messages transmis [24].

1.5.2 Antivirus

Un antivirus est un programme capable de détecter la présence de malware sur un ordinateur afin de le désinfecter. On parle ainsi d'éradication de virus pour désigner la procédure de nettoyage de l'ordinateur [25].

Il y a essentiellement deux modes de fonctionnement des logiciels antivirus :

- **Mode statique :** le logiciel est activé uniquement sur l'ordre de l'utilisateur, par exemple pour déclencher une inspection du disque dur.
- **Mode dynamique :** le logiciel est actif en permanence, et il scrute certains événements qui surviennent dans le système, ce qui induit une consommation non négligeable de ressources telles que temps de processeur et mémoire, mais permet une meilleure détection des attaques, notamment par analyse comportementale des logiciels suspects d'être contaminés [26].

a. Principe de détection des malwares

Les malwares sont des fichiers ou des portions de code embarquées dans des fichiers. Ces codes sont repérés suivant leur comportement, les fichiers qu'ils génèrent ou leur « Hash ». L'ensemble constitue leur signature virale [25]. On cite ci-dessous comment un antivirus détecte ces derniers.

- **Recherche de signature virale**

Les antivirus s'appuient ainsi sur cette signature propre à chaque malware pour les détecter. Il s'agit de la méthode de recherche de signature (Scanning), la plus ancienne méthode utilisée par les antivirus [25].

Cette méthode n'est fiable que si l'antivirus possède une base virale à jour [25].

- **Contrôle d'intégrité**

Certains antivirus utilisent un contrôleur d'intégrité pour vérifier si les fichiers ont été modifiés. Ainsi le contrôleur d'intégrité construit une base de données contenant des informations sur les fichiers exécutables du système (date de modification, taille et éventuellement une somme de contrôle). Ainsi, lorsqu'un fichier exécutable change de caractéristiques, l'antivirus prévient l'utilisateur de la machine [25].

- **Identification des fichiers sains**

Certains antivirus utilisent aussi une base de données de signatures (Hash) de fichiers sains. Ainsi, le moteur de l'antivirus n'est plus obligé de suivre les activités de programmes connues comme faisant partie du système ou d'une application courante. Comme la base de signature virale, ces signatures de fichiers sains sont mises à jour régulièrement par le programme antivirus [25].

- **Analyse des comportements**

La méthode heuristique consiste à analyser le comportement des applications afin de détecter une activité proche de celle d'un malware connu. Ce type d'antivirus peut ainsi détecter des virus même lorsque la base antivirale n'a pas été mise à jour. En contrepartie, ils sont susceptibles de déclencher de fausses alertes [25].

- **Éradication**

Il existe plusieurs méthodes d'éradication :

  - La suppression du code correspondant au virus dans le fichier infecté.
  - La suppression du fichier infecté.
  - La mise en quarantaine du fichier infecté, consistant à le déplacer dans un emplacement où il ne pourra pas être exécuté [25].

1.5.3 Pare-feu (Firewall)

Un pare-feu est un gardien de sécurité placé au point d'entrée entre un réseau privé et Internet extérieur de telle sorte que tous les paquets entrants et sortants doivent y passer. Un pare-feu accepte les paquets légitimes et rejette ceux qui sont illégitimes en fonction de sa configuration. Une configuration de pare-feu définit quels paquets sont légitimes et lesquels sont illégitimes. Une erreur dans une configuration de pare-feu, c'est-à-dire une définition erronée d'être légitime ou illégitime pour certains paquets, signifie que le pare-feu accepte certains paquets malveillants, ce qui crée par conséquent des failles de sécurité sur le pare-feu ou rejette certains paquets légitimes, ce qui perturbe par conséquent les activités normales. Étant donné l'importance des pare-feu, de telles erreurs ne sont pas acceptables [27].

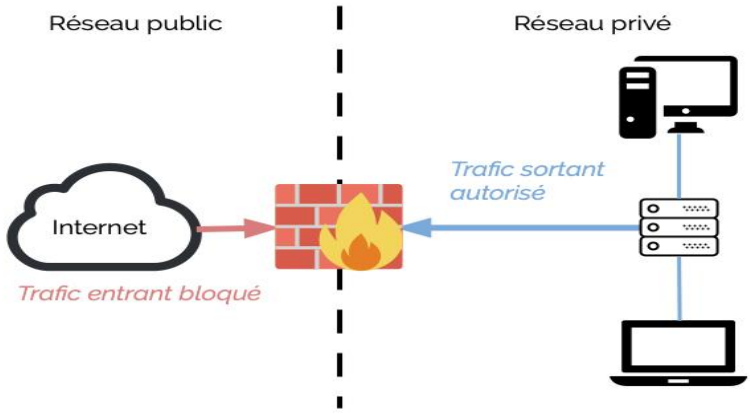


Figure 1.16. Le mécanisme d'un pare-feu [28].

1.5.4 Systèmes de détection et de prévention d'intrusions

Pour analyser le flux réseau, on utilise des systèmes de détection d'intrusion (ou IDS pour Intrusion Detection System). Il faut créer une signature de la communication entre le malware et le (Command and Control). Après avoir créé cette signature, il faut vérifier si elle apparaît en analysant le flux réseau en temps réel. Si la signature est identifiée, le système remonte alors une alerte. Cet outil permet donc également de cartographier l'infection dans le réseau.

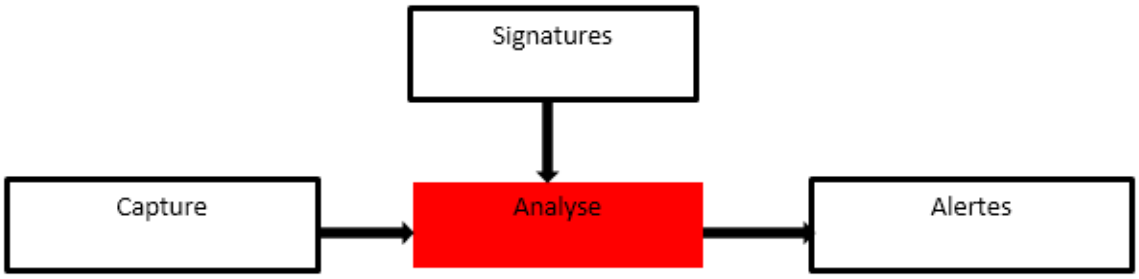


Figure 1.17. Schéma de système de détection d'intrusions.

Il existe deux outils libres permettant de faire des analyses de flux réseau avec des signatures : Snort et Suricata [29].

Un système de prévention d'intrusion IPS est un logiciel qui possède toutes les capacités d'un système de détection d'intrusion et peut également tenter d'arrêter d'éventuels incidents [30].

On différencie plusieurs types d'IDS, à savoir les NIDS qui se basent sur des analyses réseau et les HIDS qui surveillent l'activité d'un hôte. Dans notre travail on s'intéresse plus au NIDS car on va l'utiliser comme moyen de détection dans le quatrième chapitre.

a. Comment fonctionne un IDS ?

Dans un schéma réseau, il y a du trafic illustré par des paquets regroupés dans les trames. La protection des données est une politique qui incite à surveiller ce trafic, c'est pourquoi il faut aborder le thème de l'IDS. Ce système va détecter les anomalies des activités des trames qui se dirigent de l'extérieur à l'intérieur du réseau d'entreprise, ou bien des trames qui représentent des anomalies de manière interne au réseau de l'entreprise [31]. Voici le fonctionnement d'un IDS et ses options possibles :

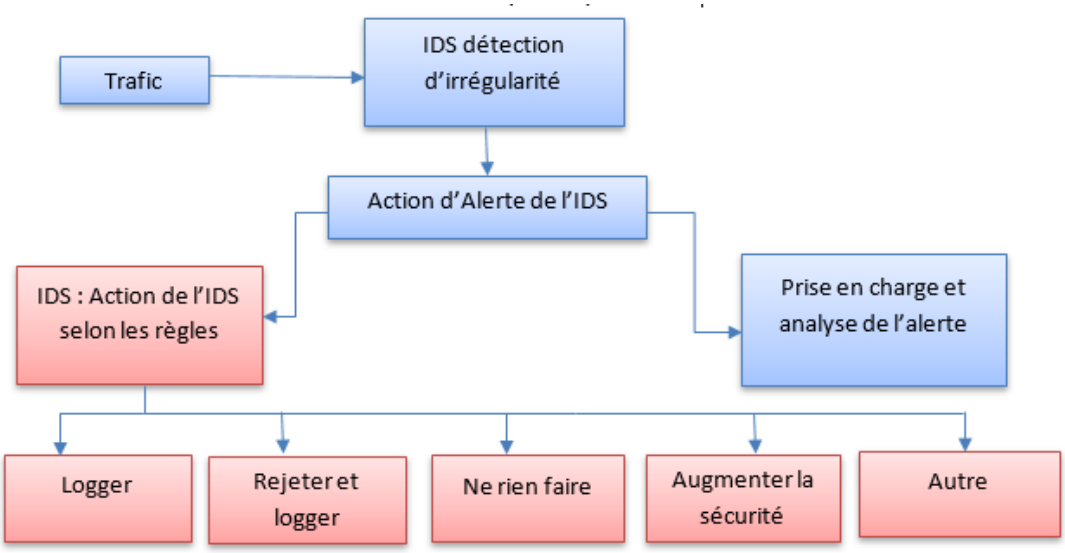


Figure 1.18. Le fonctionnement d'un IDS [31].

Sur la figure précédente, on voit le trafic de paquets qui est observé par l'IDS. Ce dernier va repérer les événements irréguliers, les anomalies, vis-à-vis de sa base de signatures et va déclencher une action d'alerte. L'alerte, d'un côté, sera prise en charge par la sécurité informatique de l'entreprise, et de l'autre, elle pourra déclencher une autre action via l'IDS.

b. Les types des IDS

- **Les NIDS :** Le NIDS tire son nom du fait qu'il surveille l'ensemble du réseau. Plus précisément, il surveille un segment de réseau entier. Normalement, une carte d'interface réseau (NIC) d'ordinateur fonctionne en mode non promiscuous. Le

NIDS doit fonctionner en mode promiscuité pour surveiller le trafic réseau, il peut écouter toutes les communications sur le segment réseau. Le fonctionnement en mode promiscuité est nécessaire pour protéger votre réseau. Toutefois, compte tenu des nouveaux règlements sur la protection de la vie privée, la surveillance des communications réseau est une responsabilité qui doit être examinée attentivement [32].

Dans la figure 1.19, nous voyons un réseau utilisant trois NIDS. Les unités ont été placées sur des segments de réseau stratégiques et peuvent surveiller le trafic réseau pour tous.

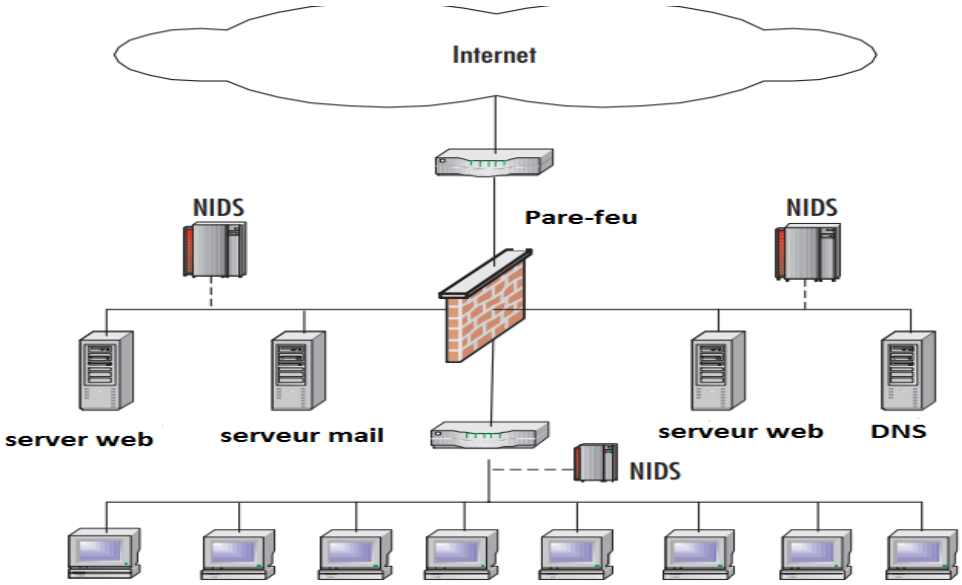


Figure 1.19. Réseau NIDS [32].

L'utilisation de plusieurs NIDS au sein d'un réseau est un exemple d'architecture de sécurité de défense en profondeur.

- **Les HIDS :** Les IDS systèmes (Host IDS) analysent le fonctionnement et l'état des machines sur lesquelles ils sont installés afin de détecter les attaques. L'intégrité des systèmes est alors vérifiée périodiquement et des alertes peuvent être levées [33].

1.6 Conclusion

Les attaques informatiques sont devenues fréquentes et des affaires courantes dont les entreprises et les simples internautes sont victimes, d'où la nécessité de faire une étude qui englobe les méthodes et les outils des attaques.

Dans ce chapitre, on a vu globalement l'aspect de la sécurité même on a expliqué les différents types d'attaque, leurs déroulements et leurs influences.

On constate qu'il n'existe pas un système sécurisé à 100 %, mais on doit rester toujours vigilants et chercher des solutions plus pertinentes contre ces menaces.

2.1 Introduction

Avec la croissance de l'utilisation du réseau internet, il y a beaucoup de sites qui apparaissent chaque jour. Cela nous facilite la vie, mais aussi ça permet de collecter les informations personnelles des internautes depuis l'avènement des réseaux sociaux et des dispositifs de surveillance. C'est pour cela que l'anonymat est devenu un besoin de plus en plus considérable. Et ça a poussé de plus en plus de gens à étudier les moyens d'anonymiser leurs informations de navigation et se protéger contre l'observation illégale et immorale, le pistage, la subjugation et peut-être même pire.

Le côté sombre de l'anonymat est l'utilisation abusive et les activités illégales (le premier exemple à citer est les attaques informatiques qu'on a bien détaillé dans le premier chapitre). Ces activités incluent aussi : kidnappings, terrorismes, escroqueries financières, piraterie et révélations de secrets et d'informations personnelles.

Ce chapitre débute avec une définition de l'anonymat et les raisons pour lesquelles on l'utilise. Ensuite, on va parler de la surveillance. Puis on va bien étudier les outils qui mettent en œuvre l'anonymat, on va se focaliser aussi sur le réseau Tor et son application dédiée pour les appareils Android Orbot, et pour finir on va citer les avantages et les inconvénients de l'anonymat sur internet.

2.2 C'est quoi l'anonymat ?

2.2.1 En général

Au cours de l'histoire, l'anonymat a été utilisé à plusieurs fins et sous différentes formes, certains le considérant comme un côté obscure synonyme de manque de courage ou lâcheté et d'autres comme un moyen de protection, l'anonymat a traversé et évolué à travers les époques et a été directement influencé par l'avancée technique des moyens de communications et du droit en générale.

Les applications et les objectifs de l'anonymat sont autant nombreuses que variées et peuvent être utilisés à des fins positives ou négatives dépendant du contexte et du point de vue, dans la vie réel le droit à l'anonymat est donc un droit fondamental [34].

2.2.2 Sur internet

Aujourd'hui l'anonymat sur internet représente un grand enjeu pour notre système et la vie privée d'une grande partie de la population de ce monde. A l'heure du Web 2.0 où les internautes sont incités à partager un peu plus de leurs données personnelles, notamment à travers les réseaux sociaux, les forums de discussion, les messageries, les sites de rencontres et autres, les suspicions et craintes d'atteinte à la vie privée, à la liberté d'expression et la surveillance de masse, notamment mener par les gouvernements, sont des thèmes de plus en plus préoccupants [34].

Les traces laissées lors de l'utilisation d'un service sur la toile sont considérables et compromettent l'anonymat. Se cacher derrière un pseudo ne suffit plus, une action en justice permet généralement de récupérer les informations nécessaires à une identification, que ce soit auprès du service concerné ou des différents intermédiaires du type fournisseur d'accès à Internet. L'anonymat réel ne peut donc pas être assurée au vu du caractère limité de cette protection juridique et c'est pourquoi de nombreux internautes ont choisis d'autres solutions techniques pour contourner cette problématique, la nécessité d'approfondir ses connaissances est requise et n'est donc pas accessible à tout le monde [34].

C'est là où entre en jeu les outils de protection de la vie privée comme Tor, VPN etc.

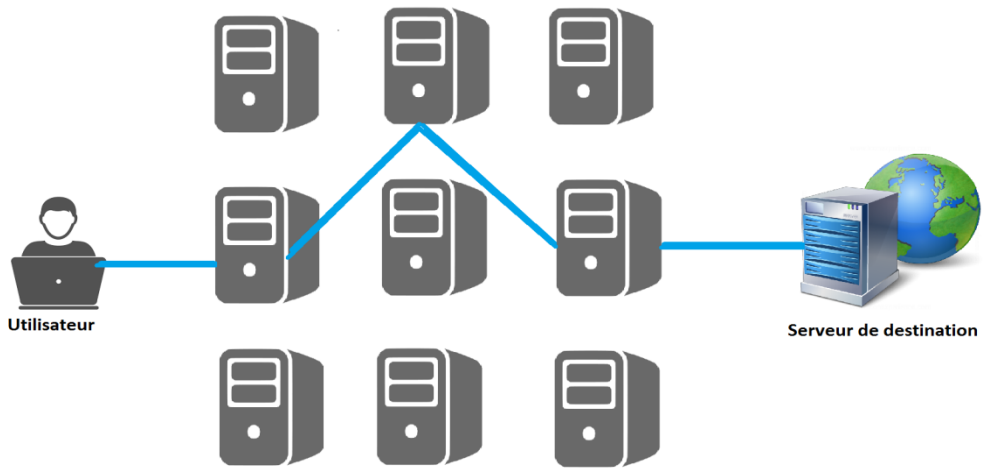


Figure 2.1. Anonymat sur internet.

2.3 Pourquoi l'anonymat sur internet ?

L'anonymat permet d'assurer et de garantir le respect de la vie privée des utilisateurs et d'éviter l'observation et la récolte de toutes sortes de données des internautes.

2.3.1 Protéger la vie privée et les données personnelles dans le monde numérique

Les données personnelles constituent un puissant carburant pour le business des entreprises, en particulier pour leurs activités publicitaires. La récolte de données personnelles permet aux entreprises de cibler plus précisément leur clientèle grâce aux informations précises qu'elles apportent.

Il existe de nombreux moyens de collecter nos données. Voici un aperçu des principaux : Tout d'abord, ces données peuvent tout simplement être "déclaratives" (directement renseignées par l'internaute) [35] :

- **Via un profil loggé** : lors de la création d'un compte (souvent obligatoire) pour accéder au service d'un site (formulaire pouvant questionner sur l'âge, les coordonnées etc.).
- **Via les données CRM d'une entreprise** : données enregistrées sur un point de vente, lors d'un évènement (remplissage d'un formulaire papier, création d'une carte de fidélité…).

Puis il y a les données **« comportementales »** qui sont récoltées plus discrètement afin d'identifier les modes de navigation ou les centres d'intérêt d'un individu : via les cookies, la recherche **« Search »** et les navigateurs.

2.3.2 Censure

La censure de l'Internet, appelée aussi cybercensure, désigne les limitations de l'information disponible par l'intermédiaire de ce réseau, pratiquée en général au niveau des États, et à l'encontre des droits de l'Homme.

2.3.3 Couvrir des actions illicites ou réprimées

Les lanceurs d'alertes et leurs actes, qui consistent à divulguer des informations qu'ils jugent menaçant pour l'intérêt général ou public, sont évidemment réprimées et lourdement sanctionner par ceux qui protègent ces informations surtout s'il s'agit d'un état, on pense notamment à l'affaire Snowden. Dans ce cas, la diffusion de ces informations, généralement vers les médias, nécessitent un canal de communication totalement anonyme pour échapper à une identification.

2.4 Surveillance sur Internet

La sécurisation de votre réseau Internet privé par exemple, au-delà de l'assurance qu'elle vous donne de ne pas partager votre Wi-Fi avec vos voisins, permet surtout à l'État d'associer un identifiant à une connexion. Et donc de s'assurer que vous êtes seul responsable de votre comportement sur le Web [36].

2.4.1 Deep Packet Inspection

L'inspection profonde de paquets consiste à placer des sondes de capture du trafic réseau aux points de concentration naturelle de ce trafic (typiquement, chez les FAI pour le grand public, sur les concentrateurs en réseau intranet) et, avec des logiciels spécialisés, à analyser en temps réel le contenu du trafic en termes de signification sémantique (de quoi parle l'information qui circule, de quel sujet traite-t-elle...). Le trafic ainsi capturé est ensuite conservé indéfiniment pour des retraitements ultérieurs (améliorations des techniques d'analyses sémantiques, nouvelles cibles sémantiques à isoler et suivre...).

Les outils de "**DPI - Deep Packet Inspection**" sont une variante des "Stateful Packet Inspection". Lorsque le "Stateful Packet Inspection" analyse, normalement, légitimement et pour des raisons techniques, l'URL et l'en-tête d'un bloc d'informations, la "Deep Packet Inspection" (DPI) capture et analyse le contenu du bloc d'informations lui-même [37].

2.4.2 Attaques basées sur les applications

Les premières attaques réseaux exploitaient les vulnérabilités liées à l'implémentation des protocoles TCP/IP. Avec la correction progressive des différentes vulnérabilités découvertes, les attaques se sont ensuite déportées vers les couches applicatives et en particulier sur le Web [39].

Figure 2.2. Attaques basées sur les applications [38].

Les vulnérabilités peuvent être diverses et catégorisés de la manière suivante [39] :

Les vulnérabilités du serveur, la manipulation d'URL, les exploitations des faiblesses des identifiants de sessions et mécanismes d'authentification, les injections de commandes SQL, et les injections de code HTML et de **"Cross Site Scripting"**.

a. Plug-ins

Parfois appelé module d'extension, un Plugin est un complément destiné à enrichir les possibilités d'un logiciel. Installé sur un navigateur, il permet d'accéder à tous les contenus du Web. Le terme Plugin vient de l'anglais "to Plug", qui signifie "brancher".

Dans un explorateur Web, il est possible qu'un malware ou un utilisateur non averti, réussissent à installer un Plug-in malveillant dont le rôle serait de compromettre la confidentialité des échanges. Ce programme se greffe à l'intérieur de l'explorateur Web, ce qui lui permet d'écouter, éventuellement de modifier, les données avant leur chiffrement par le protocole SSL/TLS [40].

b. Document actif

Un document actif permet une interaction avec l'utilisateur. Les documents actifs les plus populaires sont les fichiers PDF (Portable Document Format) développés par Adobe Systems ainsi que les documents Word et Excel développés par Microsoft. L'interaction avec les fichiers PDF est possible grâce au code JavaScript intégré dans le fichier et à l'aide de macros pour les documents Word et Excel [38].

Un attaquant peut envoyer des documents actifs qui contiennent le code malicieux. La victime exécute ce dernier en cliquant sur URL qui va finir par la création d'une connexion direct ou bien des données personnelles vont être envoyées.

c. Cookie

Figure 2.3. Cookie inséré dans un navigateur Internet [41].

Les cookies sont des petits morceaux de texte insérés dans votre navigateur pendant que vous naviguez sur le Web. Ils représentent une source d'inquiétude pour de nombreux utilisateurs. Ils sont utilisés par l'application pour récupérer un cookie de session (d'authentification) déjà inscrit dans la base de cookies. Cette entrée est potentiellement vulnérable aux attaques par injection de code SQL [42].

d. Finger Printing

L'absence du cookie au sein des applications, conjuguée au fait que beaucoup de navigateurs mobiles tels que Safari bloquent les cookies tiers par défaut, a obligé les professionnels de la publicité à lui trouver une alternative. C'est ainsi qu'est né il y a quelques

années le Finger Printing, pour offrir aux annonceurs mobiles les mêmes capacités de ciblages que sur le Web fixe.

Ainsi, l'attaquant pourrait d'abord essayer de construire l'empreinte digitale des tailles de fichiers de pages Web, puis surveiller l'utilisateur. Lorsque l'utilisateur navigue sur une page Web, les connexions et les données connexes pourraient être détectées par l'attaquant. Ensuite, l'attaquant compare les données de connexion avec un ensemble d'empreintes digitales, choisissez la plus proche, puis devinez que la page est ce que l'utilisateur surfe maintenant. L'attaque est à faible coût, facile à appliquer, et nuit vraiment à l'anonymat de l'utilisateur [43].

2.5 Outils pour mettre en œuvre l'anonymat

L'utilisation d'un outil qui permet d'assurer l'anonymat sur internet est devenue de plus en plus primordiale, cela pour éviter maints problèmes : manipulation de l'opinion publique, fuite de données personnelles etc.... Dans certains pays, comme la Chine, le gouvernement empêche les machines connectées depuis le sol chinois de se connecter à de nombreux services occidentaux ou non approuvés par le pouvoir en place (le cas de Facebook). Être anonyme sur internet nécessite de l'énergie, du temps et requiert des connaissances et compétences.

On va étudier quelques outils qui mettent en œuvre l'anonymat, et ça reste à chacun de choisir son degré d'anonymat selon ses besoins.

2.5.1 Adresse email jetable

Nous savons bien que les divers fournisseurs d'adresses mail gratuites telles que Yahoo, Hotmail et Gmail, tirent profit de nos données personnelles. En outre, une adresse mail de ce type, à l'instar de celles obtenues via l'abonnement à un FAI ou dans un cadre professionnel, permet de vous identifier directement ou indirectement même si vous ne signez pas vos mails et si l'adresse ne porte pas votre nom, pour la simple et bonne raison que le code source de vos messages contient votre adresse IP [44].

Une adresse email jetable est une adresse électronique temporaire, valide pour une durée généralement éphémère, et servant d'adresse poubelle pour toutes les fois où un site

demande de s'inscrire en renseignant son email. En effet, la plupart des sites demandent une adresse email au moment de l'inscription. Ceci est généralement peu risqué, mais il arrive que certains sites peu scrupuleux revendent ces adresses à des sociétés, publient vos informations sur le Web ou envoient elles-mêmes des courriers publicitaires intempestifs [45].

2.5.2 Navigateurs anonymes et sécurisés pour la navigation Web privée

La plupart des navigateurs les plus connus, tels que Chrome et Firefox, intègrent une fonction de navigation privée dont la majorité des utilisateurs pensent qu'elle permet de naviguer de façon anonyme. C'est en partie vrai puisque ces sessions suppriment des éléments tels que l'historique de navigation et les cookies à la fin de la session. Toutefois, ces informations sont seulement supprimées de votre ordinateur, et non des autres emplacements où elles peuvent exister.

Les navigateurs sécurisés acheminent le trafic Web via leur réseau pour anonymiser les informations. Les méthodes utilisées diffèrent d'un navigateur à l'autre. À titre d'exemple, Tor utilise la méthode de routage en **« oignon ».**

a. *Le navigateur Tor*

Le navigateur Tor ressemble et fonctionne de la même manière que n'importe quel autre navigateur Web, comme Chrome, Edge et Firefox avec quelques différences.

Tout d'abord, Tor Browser n'est pas connecté directement au World Wide Web. Tout le trafic Internet, entrant et sortant, passe d'abord par le réseau Tor. Le réseau Tor est composé de milliers de relais, appelés nœuds, situés dans le monde entier. Tous ces relais sont gérés par des bénévoles qui soutiennent le projet Tor et le gardent ouvert à tous, gratuitement.

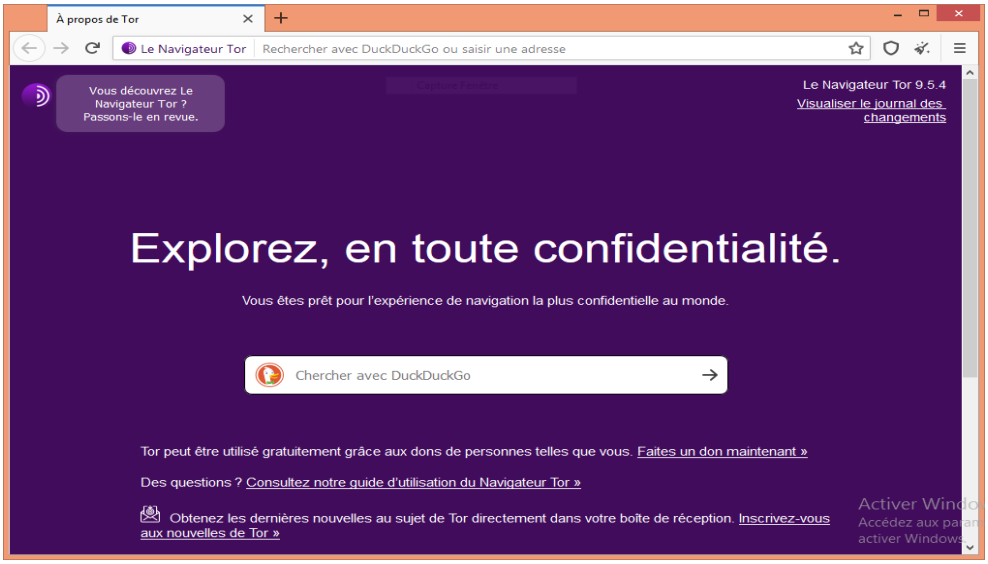


Figure 2.4. Interface du navigateur Tor.

2.5.3 Moteurs de recherche anonyme

Les moteurs de recherche traditionnels, comme Google, Yahoo ou Bing collectent toute l'activité des utilisateurs et d'autre informations dans une immense base de données, tel que les sites visités, les mots clé recherchés, votre adresse IP , votre localisation et cela leur permet de connaitre vos centre d'intérêts, votre fiche d'identité, vos conviction politiques, la liste de vos contacts, ... Autant de données qui sont exploitées à votre insu et dont de nombreuses organisations souhaitent s'approprier.

Certains moteurs de recherche garantissent la non-collecte de votre activité. Il en existe des dizaines, plus ou moins performants. Mais ce qui fait la qualité d'un bon moteur de recherche c'est sa performance et la pertinence de ses recherches.

a. Startpage / Lxquick

Ces deux moteurs sont proposés par le même éditeur. Startpage propose en complément un mode de recherche renforcée chez Google dont le principe être de vous offrir les mêmes résultats que Google. Pour protéger votre vie privée votre requête est envoyée chez Google via Startpage de manière anonyme. Startpage est la meilleure alternative pour ceux qui ont peur de ne pas avoir la même qualité de recherche que Google [46].

b. DuckDuckGo

DuckDuckGo est un moteur de recherche lancé en 2008. Ses créateurs le définissent comme **« le moteur de recherche qui ne vous piste pas »**. Comprenez par-là que ce Web engin n'utilise pas de cookies pour suivre les utilisateurs, et ne collecte pas de données personnelles sur ceux qui l'utilisent. Même l'adresse IP des utilisateurs est dissimulée [47].

2.5.4 Module d'extension du navigateur

a. Protection contre les cookies

Quelques outils et astuces pour limiter ses traces de sa navigation Web sur les navigateurs.

- **Self-Destructing Cookies**

Self-Destructing Cookies est une extension qui permet de supprimer automatiquement les cookies et le stockage local lorsque le navigateur est fermé ou une fois l'onglet fermé pour empêcher le suivi.

b. Protection contre le pistage

- **Ghostery**

Ghostery est une puissante extension de protection de la vie privée. Elle permet de bloquer les publicités, déjouer les outils de pistage, protéger vos données personnelles et naviguer plus rapidement [48].

c. Protection des communications

- **HTTPS Everywhere**

Beaucoup de grands sites centraux au Web (comme Google, Facebook, Wikipédia...) proposent des versions sécurisées, permettant de garantir la confidentialité des données par rapport au réseau et aux services avec lequel le site communique. En passant par le protocole HTTPS, les données sont chiffrées entre l'internaute et le service et, en théorie, impossibles à lire par un tiers qui s'intercalerait entre les deux, contrairement au HTTP classique [49].

Sur Firefox et Chrome, une extension permet de s'assurer que tous les services disponibles en version HTTPS sont bien utilisés face à leurs versions non sécurisées : HTTPS Everywhere en version stable pour le navigateur de Mozilla et en version bêta pour celui de Google [49].

2.5.5 Réseau privé virtuel (Virtual Private Network « VPN »)

En termes simples, cela nous permet de créer un réseau privé sur un réseau public. La plupart des organisations utilisent un réseau privé interne pour leurs opérations quotidiennes, mais parfois, les gens doivent accéder à ce réseau de l'extérieur, où il n'y a pas de connexion directe à ce réseau. C'est à ce moment que le VPN entre en jeu : il permet aux utilisateurs d'accéder en toute sécurité au réseau privé depuis Internet [50].

Le VPN crée essentiellement une connexion virtuelle point à point entre deux machines. Le serveur VPN est installé à un moment donné et l'utilisateur y accède à l'aide d'un client VPN.

Le VPN utilise différents mécanismes et technologies, tels que l'authentification, l'autorisation, le cryptage, etc., pour y parvenir et maintenir la connexion sécurisée. Il existe différents cas d'utilisation et implémentations du VPN et son fonctionnement, mais dans ce chapitre, nous nous concentrons principalement sur son utilisation pour rester anonyme.

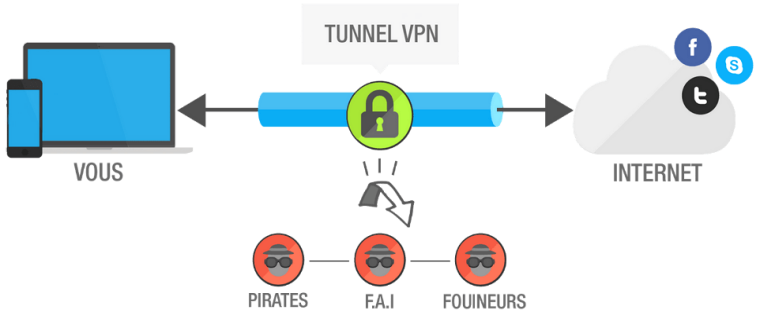


Figure 2.5. Le mécanisme utilisé par un VPN [51].

Il existe divers services VPN de ce type disponible en ligne et la plupart d'entre eux sont payants. Ici, nous listons deux services qui fournissent une version gratuite, mais ils viennent avec leurs propres restrictions telles que le temps limité, la vitesse,…..etc. [50].

- Cyberghostvpn.
- Hideman.

2.5.6 Proxy

Un serveur proxy est un système informatique situé entre le client demandant un document Web et le serveur cible servant le document. Dans sa forme la plus simple, un serveur proxy facilite la communication entre le client et le serveur cible sans modifier les demandes ou les réponses. Lorsque nous initions une demande de ressource auprès du serveur cible, le serveur proxy détourne notre connexion et se représente en tant que client auprès du serveur cible, demandant la ressource en notre nom. Si une réponse est reçue, le serveur proxy nous la renvoie, donnant l'impression que nous avons communiqué avec le serveur cible [52].

Un serveur proxy peut filtrer les demandes en fonction de diverses règles et autoriser la communication uniquement lorsque les demandes peuvent être validées par rapport aux règles disponibles. Les règles sont généralement basées sur l'adresse IP d'un client ou d'un serveur cible, le protocole, le type de contenu des documents Web, le type de contenu Web, etc. [52]

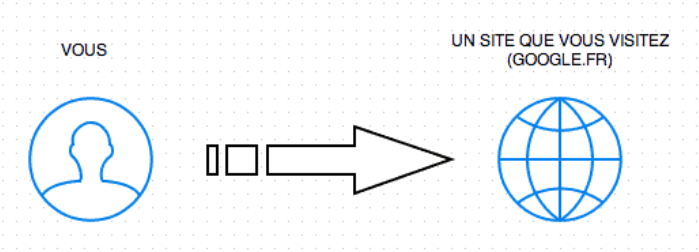


Figure 2.6. Connexion à un site Web sans proxy [53].

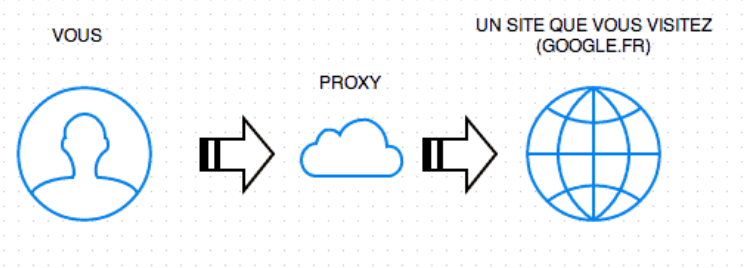


Figure 2.7. Connexion à un site Web avec proxy [53].

2.5.7 Réseaux anonymes

a. Le réseau Tor

- **La naissance du Tor public**

Le projet Tor a été établi par le gouvernement et développé par la **« DARPA : Defense Advanced Research Projects Agency »** en tant que mesure de sécurité. En utilisant le principe de routage en oignon qui permet d'assurer des communications privées et anonymes [54].

Tor est, comme on peut le lire sur le site de Tor Project, **« un logiciel libre et un réseau ouvert qui vous aide à vous défendre contre une analyse du trafic, une forme de surveillance de réseau qui menace les libertés individuelles et la vie privée, les activités commerciales confidentielles et les relations, et la sécurité de l'Etat »**. Si le but primaire semble honorable, comme bien souvent, cette bonne attention a été détournée.

Lors d'une récente expérience menée par l'équipe IBM Managed Security Services, les experts ont démontré que les criminels du Web et autres organisations hostiles utilisent Tor pour cacher leurs communications, lancer leurs attaques, voiler leurs infrastructures, commander et contrôler les Botnets ou encore acheter en accédant au Dark Web de la drogue, des armes et autres types de produits illicites et gravement puni par la loi. Ce logiciel est devenu une véritable protection supplémentaire aux attaquants qui peuvent ainsi masquer la location physique de l'attaque ou alors la remplacer par une autre de leur choix [55].

Tor est le moyen le plus populaire pour accéder au Dark web, citons ci-dessous qu'est-ce que le Dark Web et à quoi sert-il ?

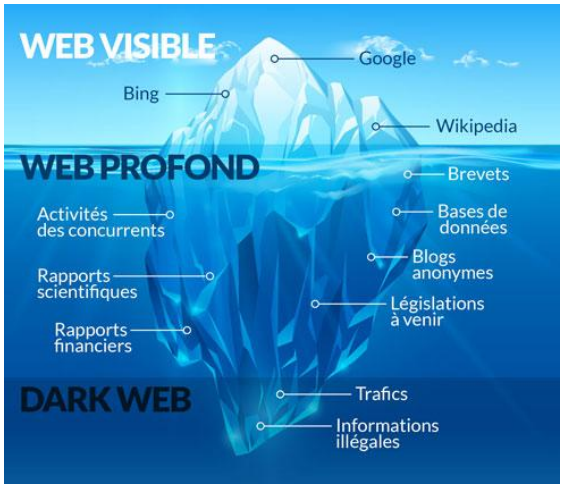


Figure 2.8. Les différentes interfaces Web [56].

Les différents types de web sont les suivants :

- **Le Web surfacique (Surface Web)**

Commençons par nommer **« le web de tous les jours »** : on l'appelle le Web de Surface. C'est le Web que la majorité des gens utilise via des navigateurs classiques comme Chrome ou Firefox, il représente tout ce qui peut être indexé par un moteur de recherche typique comme Google, Bing ou Yahoo.

Bien que la partie du Web la plus utilisée, le web de surface ne représente que 4% environ des pages Web publiées [57].

- **Le Web profond (Deep Web)**

Le Web profond est la plus grande partie d'internet, environ 96% des pages Web. Il est constitué de toutes les pages qui ne sont pas indexées par les moteurs de recherche mais qui ne sont pas cachées et qui restent accessibles via un navigateur classique si l'on en connait l'URL ou le mot de passe. Par exemple, votre boite email, votre ancien blog d'il y a 15 ans qui n'est plus indexé, des pages non indexées par une directive, des pages générées dynamiquement, des bases de données, des archives privées, des pages de Netflix ou Amazon... etc. Toutes ces pages font partie du Web profond [57].

- **Le Web sombre (Dark Web)**

Le web sombre est en fait une petite partie du Web profond qui a été intentionnellement cachée et est inaccessible par les navigateurs web standard. Le Web profond et le Web sombre sont souvent confondus, mais ce sont deux choses distinctes. Le

Web profond est la somme de toutes les pages web qui ne sont pas trouvables par les moteurs de recherche classiques, tandis que le Web sombre est la partie du Web profond qui est cachée et que seules des navigateurs Dark net (comme Tor Browser) pourront accéder. Il est impossible d'ouvrir une page du Dark Web avec Chrome ou Firefox, par exemple [57].

- **Qu'est-ce Tor ?**

Tor est l'acronyme de « **T**he **O**nion **R**outer », il est conçu pour être utilisé comme ce qu'on appelle un **« anonymizer »** qui est un logiciel que vous exécutez sur votre ordinateur afin de brouiller l'empreinte unique que vous laissez sur Internet et le rendre virtuellement introuvable. Cela signifie que quelqu'un qui surveille votre connexion Internet ne pourra pas dire quels sites vous avez visités, et les sites que vous avez visités ne pourront pas dire d'où provient votre connexion (c'est-à-dire votre emplacement physique) [58].

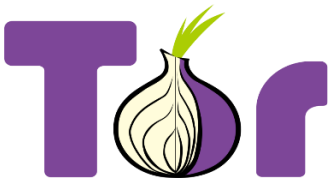


Figure 2.9. Logo Tor.

- **Le fonctionnement de Tor**

Le fonctionnement de Tor est basé sur une architecture déconcentrée qui utilise le routage en oignon.

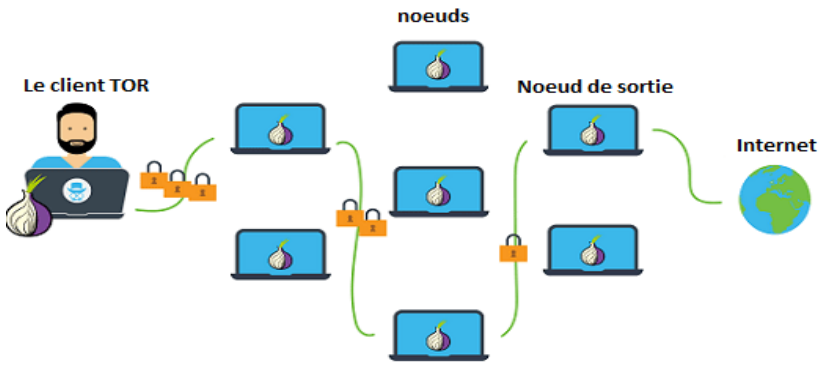


Figure 2.10. Le fonctionnement de Tor [59].

Une requête effectuée sur le navigateur Tor Browser (et son moteur de recherche associé DuckDuckGo) passera à travers trois « relais », c'est-à-dire des collaborateurs du projet

Tor et/ou des utilisateurs dont l'ordinateur héberge des données transitant sur le réseau Tor. L'expression « routage en oignon », qui donne son nom à Tor, désigne la technique d'encryptage des données transitant sur le réseau à travers les relais dont chacun ajoute une nouvelle couche d'encryptage au signal qu'il fait transiter.

En naviguant avec Tor , on utilise donc, pour parvenir à n'importe quel site que cela soit sur le Web surfacique ou le Web sombre de Tor une série de nœuds de connexions intermédiaires qui procède à un décryptage et à un nouvel encryptage des données à chaque relais un peu comme si un groupe de personnes se passait un message de main en main dans une série d'enveloppes et que chaque membre du groupe enlevait une enveloppe avant d'en rajouter une autre, puis de la passer à son voisin Le serveur final pourra seul avoir accès à l'information véhicule pour la délivrer, tout en protégeant l'identité de l'expéditeur [60].

- **L'architecture du réseau Tor**

Le réseau Tor est composé du client Tor, d'un nœud d'entrée / garde, de plusieurs relais et du nœud de sortie.

- **Le client Tor :** est un logiciel installé sur l'appareil de chaque utilisateur Tor. Il permet à l'utilisateur de créer un circuit d'anonymisation de Tor et de gérer toutes les clés cryptographiques nécessaires pour communiquer avec tous les nœuds du circuit.

Le client va d'abord commencer par obtenir une liste des nœuds du réseau susceptibles d'être utilisés pour construire son circuit. Il va donc demander à un Directory Server, un serveur annuaire, de lui fournir cette liste.

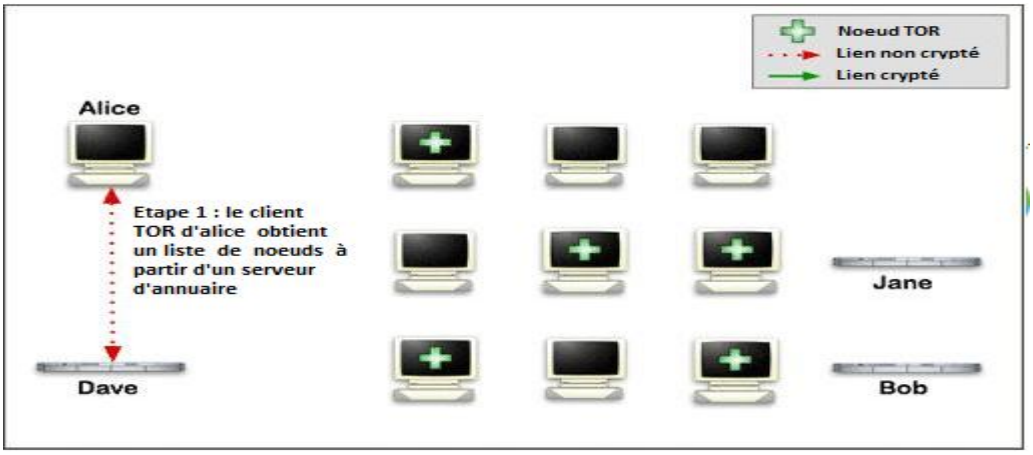


Figure 2.11. L'obtention de la liste des nœuds par le client Tor [61].

- **Le nœud d'entrée :** est le premier nœud du circuit qui reçoit la demande du client et la transmet au deuxième relais du réseau.
- **Le nœud de sortie :** est le dernier relais Tor du circuit.

Une fois que la demande de connexion quitte le nœud d'entrée, elle sera renvoyée parmi tous les relais du circuit jusqu'à ce qu'elle atteigne le nœud de sortie. Ce dernier reçoit la demande et la transmet à la destination finale.

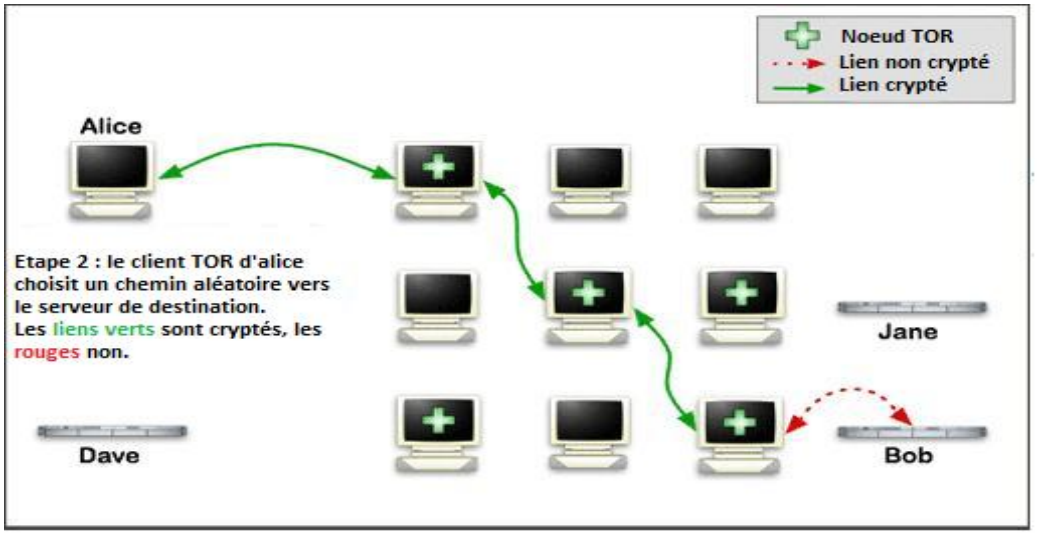


Figure 2.12. Le choix du chemin aléatoire vers le serveur par le client [61].

Comme montre la figure ci-dessus, les connexions dans le réseau Tor entre les nœuds d'entrée et de sortie sont cryptées à l'aide de la norme de cryptage avancée **(AES)**. Cependant, les connexions entre le nœud de sortie et la destination finale ne sont pas chiffrées par Tor. Cela implique que si la session entre le client et la destination n'est pas chiffrée dans le cadre d'un protocole de sécurité de couche supérieure, tel que le HTTPS, un attaquant résidant à proximité de la destination pourra divulguer les données [62].

Pour plus d'efficacité, le logiciel Tor utilise le même circuit pour les connexions qui se produisent dans les mêmes dix minutes environ. Les demandes ultérieures reçoivent un nouveau circuit, pour empêcher les gens de lier vos actions précédentes aux nouvelles [63].

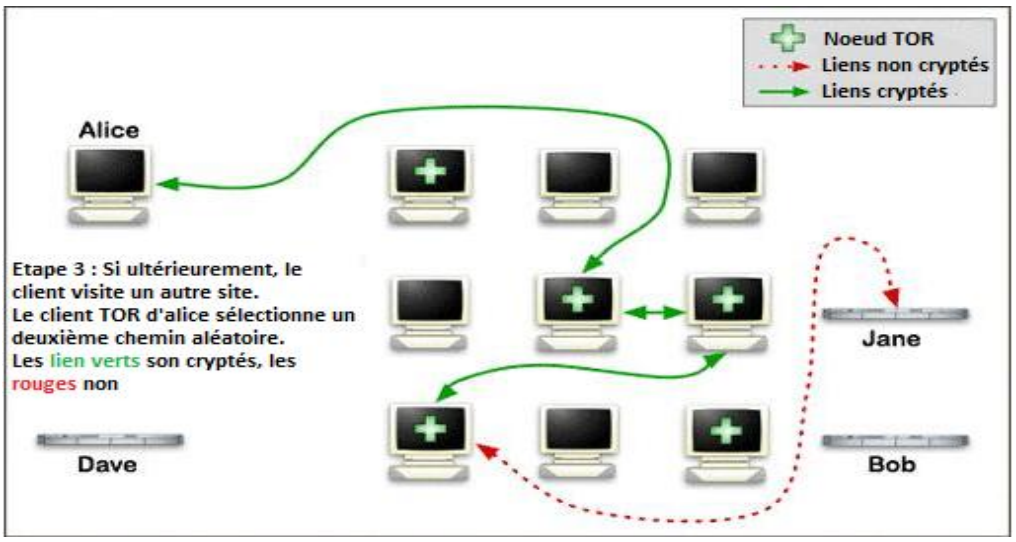


Figure 2.13. La sélection du deuxième chemin par le client [61].

- **Le routage en oignon**

Le réseau Tor utilise plusieurs proxies **(appelés relais)** pour réaliser votre requête. Et cela change tout. Lorsque vous vous connectez à un proxy, votre connexion est chiffrée. Cela signifie que le message que vous lui envoyez ne peut être décodé que par lui.

Lorsque vous utilisez plusieurs proxies, vous chiffrez plusieurs fois le message que vous voulez fournir au proxy qui fera votre requête. Ce message est décodé au fur et à mesure qu'il passe entre les différents proxies. Si bien qu'au final, seul le proxy qui fait votre requête connaît le message que vous lui avez adressé. C'est ce que l'on appelle le routage en oignon.

Comme mentionné précédemment, Tor utilise AES pour crypter les connexions entre les relais. Cependant, AES est un algorithme de chiffrement symétrique où la même clé est utilisée pour le chiffrement et le déchiffrement. Cela signifie que pour que le client Tor utilise AES, il doit avoir des clés de chiffrement partagées avec les nœuds de relais. Pour partager ces clés. Tor utilise un chiffrement asymétrique dans le cadre du protocole TLS **(Transport Layer Security)** [62].

Chaque connexion d'un relais Tor à un autre au sein du réseau Tor est protégée par le protocole TLS. Au moment de la construction du circuit, le client Tor doit rassembler toutes les clés publiques des nœuds du circuit et doit établir une connexion avec chacun d'eux, ces

connexions seront utilisées pour échanger les clés symétriques pertinentes. Pour convenir d'une clé partagée avec un relais, le client Tor démarre un échange de clés Diffie-Hellman [62].

Initialement, le client Tor crypte le défi Diffie-Hellman avec la clé publique du récepteur, puis crypte le message obtenu avec la clé publique du relais Tor précédant le récepteur dans le circuit. Ce nouveau message sera encore chiffré avec la clé publique de tous les relais restants dans le circuit en sens inverse et dans leur ordre de rencontre respectivement. Finalement, le message multi-encapsulé est envoyé. Cette procédure est illustrée à la Figure ci-dessous [62].

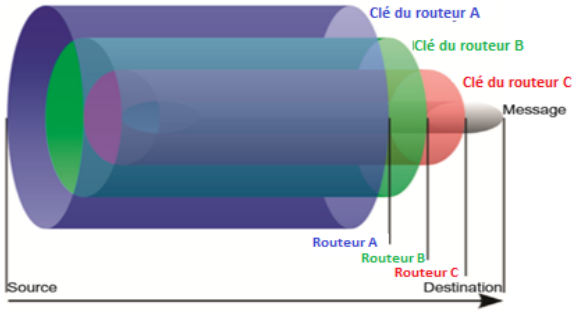


Figure 2.14. Message envoyé encapsulé [64].

Lorsque le nœud d'entrée reçoit le message, envoyé par le client Tor, il décolle de la première couche de cryptage à l'aide de sa clé privée et, par conséquent, trouve enfermé un autre message crypté. Le relais ne peut pas déchiffrer le message joint car il est chiffré avec la clé publique du relais suivant. Cependant, dans le contenu qu'il peut comprendre, il recevra son défi Diffie-Hellman personnel et l'instruction de transmettre le reste du message au relais suivant. Lorsque le prochain relais dans le circuit reçoit le message, il décolle une autre couche du message multi-encapsulé et ne pourra pas non plus comprendre le contenu interne, sauf à partir de l'instruction de transfert et de son propre défi. Il est important de noter que les informations de routage de chaque message sont également cryptées avec la clé publique de chaque relais. L'utilisation de l'encapsulation multiple garantit que les relais intermédiaires sont anonymisés, chaque récepteur ne peut pas connaître l'identité du relais qui vient après le suivant dans le circuit. Finalement, lorsque le message multi-décapsulé atteindra le relais de sortie, ce dernier supprimera la couche de chiffrement à l'aide de sa propre clé privée et récupérera son défi Diffie-Hellman [62].

À ce stade, tous les relais du circuit sont en mesure d'établir un secret partagé avec le client Tor (en terminant la procédure de défi-réponse), et ce dernier peut commencer à envoyer des données à la destination finale en utilisant un cryptage symétrique [62].

b. Orbot : Tor pour Android

On va utiliser l'application Orbot qui permet de passer par le réseau Tor sur un appareil Android. Le choix de cette application est venu avec la forte croissance du taux d'utilisation du Smartphone.

De plus en plus de gens naviguent sur le Web à partir de leur téléphone. Et, dans les régions les plus pauvres du monde, où les communications se font essentiellement en mode mobile. C'est l'une des raisons pour lesquelles les smartphones dominent les PCs **« le coût »,** on cite aussi :

- Les smartphones sont beaucoup plus faciles à transporter et à ouvrir vos applications préférées à votre convenance.

La comparaison entre l'utilisation des Smartphones et les ordinateurs ces dernières années est devenue un sujet d'actualité. La figure ci-dessous nous montre une statistique qui compare le temps que les gens ont historiquement passé à consommer des médias sur différents types d'appareils chaque jour.

Sans surprise, entre 2013 et 2019, le temps passé sur les médias sur un ordinateur de bureau a légèrement diminué. Au cours de la même période, le temps passé sur les médias sur un appareil mobile a fortement augmenté [65].

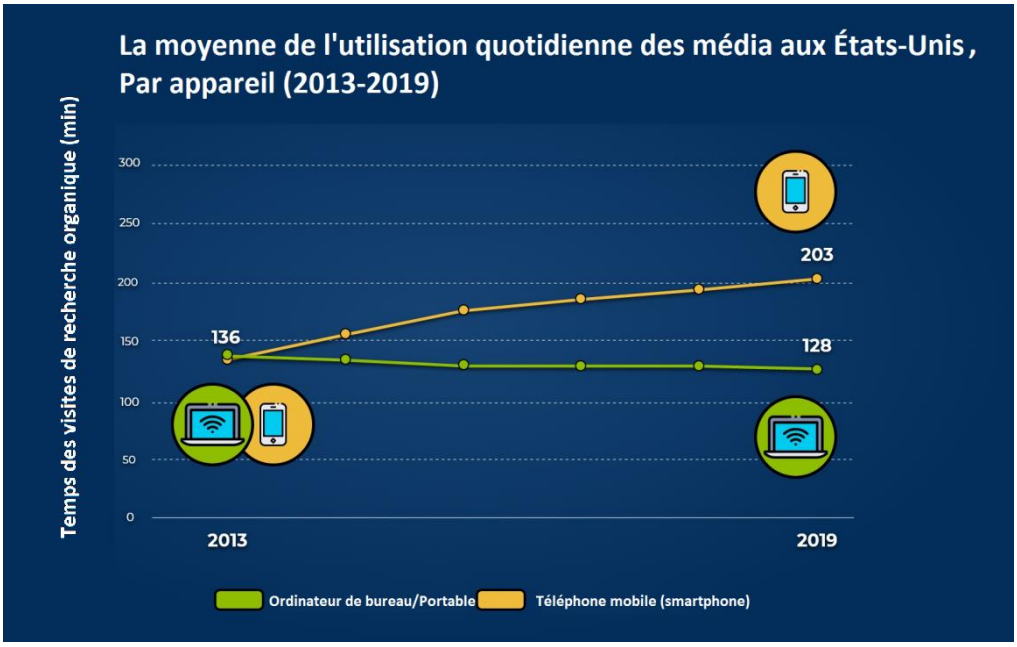


Figure 2.15. La comparaison entre l'utilisation des Smartphones et les ordinateurs [65].

Si nous voulons une vue d'ensemble de la manière dont l'accès au Web mobile a changé au cours de ces cinq dernières années, nous pouvons examiner le pourcentage total des sites Web mondiaux consultés via des téléphones mobiles de 2013 à 2018 [65]. Comme indiqué ci-dessous :

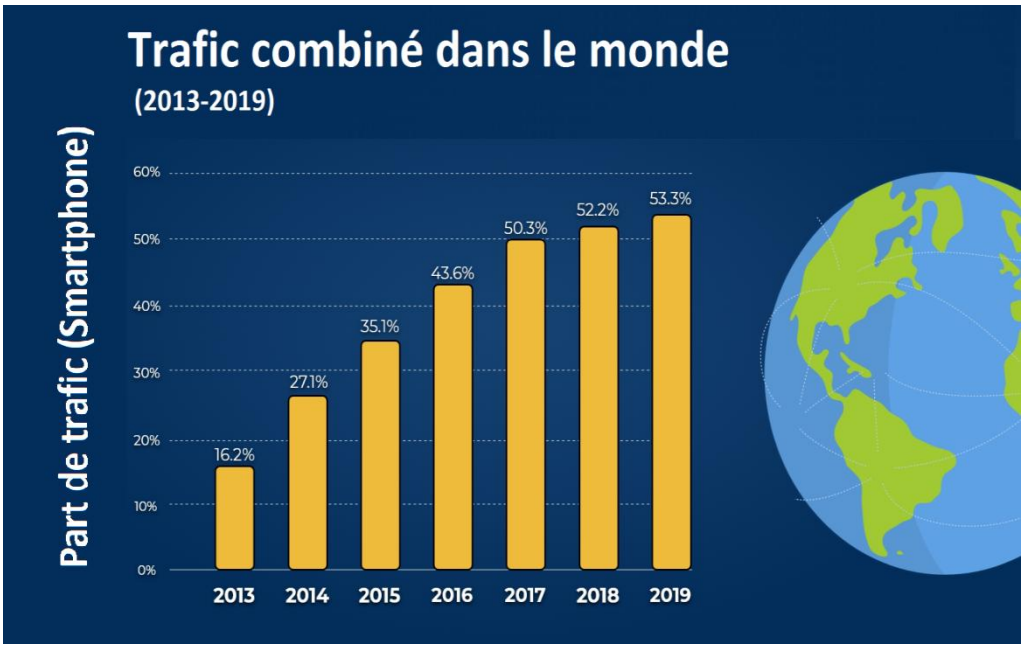


Figure 2.16. L'évolution de l'accès au Web mobile [65].

- Les téléphones mobiles représentaient 16,2% du trafic mondial en 2013.
- En 2019, 53% du trafic provenait des téléphones mobiles.

Il est important de noter que ces données ne représentent que les téléphones mobiles, et non les autres trafics mobiles comme le trafic des tablettes, qui a également augmenté au cours de la même période [65].

Cette forte croissance d'utilisation des smartphones est la raison pour laquelle le projet Tor a consacré plusieurs années à développer et à améliorer le navigateur sur mobile (l'application Orbot et le navigateur Tor).

Figure 2.17. Logo Orbot [66].

Afin de bien détecter l'utilisation de cette application, on doit passer tout d'abord par l'étude théorique du fonctionnement de cette dernière.

- **Définition**

Le projet Tor a annoncé la sortie officielle de l'application Orbot pour Android. Orbot, un proxy Tor pour Android qui permet de faire passer tout le trafic de ses applications sur Tor [66].

Orbot est la seule application qui crée une connexion à Internet réellement privée. Pour citer le New York Times : **« Quand une communication arrive par Tor, on ne peut jamais savoir d'où, ou de qui elle vient. »** [66].

- **Le fonctionnement de Orbot**

Orbot fait rebondir votre trafic chiffré plusieurs fois à travers différents ordinateurs répartis dans le monde, au lieu de vous connecter directement comme le font les VPN et proxies. Le processus est un peu plus long, mais attendre un peu pour la meilleure protection possible de la vie privée et de votre identité en valent la peine [67].

L'application achemine tout le trafic Web à travers le réseau Tor, en l'anonymisant. Tor se compose d'un proxy à trois couches, comme les couches d'un oignon. Orbot se connecte au hasard à l'un des nœuds d'entrée publics, fait rebondir ce trafic à travers un relais

intermédiaire également sélectionné au hasard, et laisse finalement ce trafic passer par un troisième et dernier nœud de sortie.

- Utiliser avec le navigateur Tor (APK), Orbot est la manière la plus anonyme d'accéder à n'importe quel site Web, même si celui-ci est habituellement bloqué, surveillé ou fait partie du web caché.
- Le menu de l'application vous permet de sélectionner les applications avec lesquelles vous voulez utiliser les services de Orbot, d'une façon très simple. En un seul clic, vous serez protégé de n'importe quelle attaque privée.
- Utilisez Orbot avec Twitter ou essayez un moteur de recherche privé comme DuckDuckGo.

- **Configuration**

Orbot peut être configuré pour faire passer tout votre trafic Internet de manière transparente par Tor. Vous pouvez également choisir quelles applications en particulier vous voulez faire transiter par Tor.

Pour commencer, nous allons installer l'application « Orbot » sur un smartphone :

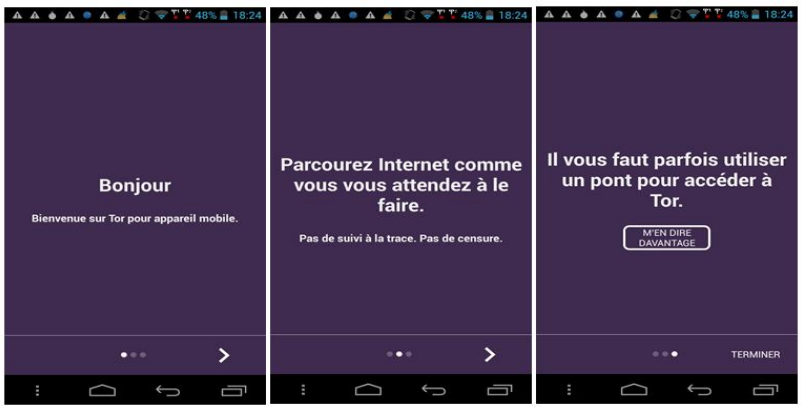


Figure 2.18. Étapes d'installation de l'application Orbot.

L'application est très simple d'utilisation et de configuration. Comme nous pouvons le voir, il est écrit Mondial. Cela permet de choisir le lieu de votre connexion, si vous laissez Mondial, le pays sera choisi aléatoirement.

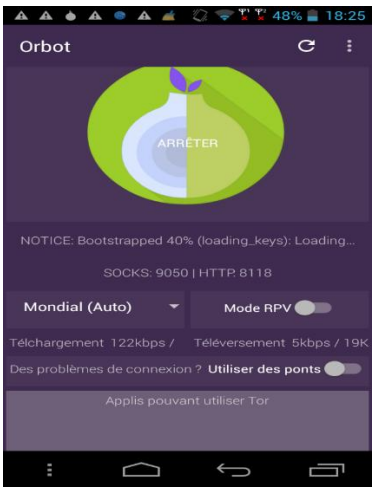


Figure 2.19. Interface de l'application Orbot.

- Une fois le pays sélectionné, vous pouvez cliquer sur Démarrer.
- Maintenant vous êtes anonyme lorsque vous surfez sur Internet.
- Vous pouvez donc utiliser le navigateur internet : Orfox (précédemment) qui est remplacé par Tor browser.

En utilisant Tor sur votre Android (Orbot), vous naviguez sur Internet de manière anonymement et cela gratuitement.

2.6 Les avantages et inconvénients de l'anonymat

Avantages	Inconvénients
• Protéger la vie privée et renforcer la sécurité des internautes en chiffrant leur trafic (VPN). • Anonymiser encore plus les données personnelles et bloquer des sites malveillants afin d'éviter toute menace potentielle en chiffrant les requêtes web (proxy). • Brouiller les pistes. • Intraçabilité des internautes (Tor). • Contourner la censure sur internet « cybecensure ». • Liberté d'expression et la protection pour les personnes les plus vulnérables « manifestants, journalistes ».	• Une navigation plus lente en utilisant les outils qui mettent en œuvre l'anonymat comme Tor. • Amoindrir l'efficacité de la recherche à cause de l'utilisation des moteurs de recherche privés. • Couvrir des actions illicites. • Le lancement de controverses, voir des insultes et des fausses nouvelles par des internautes qui sont sous le couvert de l'anonymat. • L'utilisation de l'anonymat par des individus mal intentionnés des activités criminelles « piraterie, harcèlement... etc. ». • La circulation des contenus haineux sur internet « contenus racistes …etc. ».

Tableau 2.1. Les avantages et les inconvénients de l'anonymat.

Les réseaux anonymes comme Tor (citons aussi Orbot « Tor pour Android ») offrent une protection de la vie privée comme bien détaillé précedemment. Mais l'utilisation de ce réseau est souvent associée au Web sombre et à son utilisation par les cybercriminels pour naviguer dans le côté obscur d'Internet et effectuer des activités illicites.

La couche supplémentaire de sécurité que Tor offre pour les utilisateurs encourage les attaquants informatiques à l'utiliser pour dissimuler leurs activités iillégales (attaques informatiques). Ce qui rend la mise en œuvre d'une technique de détection de l'utilisation de ce réseau primordiale.

2.7 Conclusion

Après l'énumération des techniques et technologies permettant de surveiller et protéger les internautes. Nous avons vu que la solution la plus sûre est l'utilisation des réseaux anonymes. Ceux-ci sont conçus pour protéger l'anonymat de leurs utilisateurs. Les réseaux anonymes sont souvent construits sur le même modèle, celui du relais de proxy et du chiffrement en couche. Citons que même les utilisateurs les plus avertis ont du mal à rester anonymes. Malgré les outils, il y a toujours un moyen de retrouver l'identité de l'internaute.

Cette étude détaillée de ces réseaux anonymes va nous permettre de proposer une méthode de détection de l'utilisation de Orbot dans le chapitre suivant pour lutter contre toute forme de mauvaise utilisation de l'anonymat sur internet en utilisant comme solution les systèmes de détection d'intrusion cités précédemment.

3.1 Introduction

Comme nous l'avons mentionné précédemment, Orbot vous aide à préserver votre confidentialité car il permet aux gens de cacher leurs identités. Malheureusement y a des gens qui n'utilisent pas Orbot pour une noble cause. C'est pour cela il faut faire preuve de prudence face au risque de l'utilisation des applications Orbot et Tor Browser.

La détection de Orbot et Tor Browser dans un réseau local est compliquée car elle nécessite des mécanismes de sécurité très utiles qui visent à combattre tout type d'intrusions.

Dans ce chapitre on va faire une étude qui est basée sur :

- **En premier lieu,** la capture des paquets avec un outil de capture et d'analyse.
- **Deuxièmement,** l'analyse en détails de tout le trafic « Paquets » venant des applications Orbot et Tor.
- **A la fin,** le relèvement des empreintes numériques qui vont être par la suite implémenter dans un système de détection d'intrusions afin de détecter l'utilisation du logiciel d'anonymat « Orbot ».

3.2 L'objectif de notre recherche

L'objectif de notre recherche est la détection de l'utilisation de l'application Orbot sous Android en créant des règles avec un IDS **« Snort »** à partir de l'analyse des paquets sous Wireshark.

3.3 Plan de travail

Pour atteindre notre objectif et trouver une solution robuste qui va être la réponse à notre problématique, on a suivi l'enchaînement des étapes suivantes :

- **La documentation :** on doit en premier lieu passer par l'étude des concepts théoriques pour l'identification de l'application Orbot (réseau Tor).

- **La capture :** on a choisi un analyseur de paquets Open Source « Wireshark » qui permet de récolter pas mal d'informations sur plusieurs protocoles ayant lieu avec l'application Orbot.
- **L'analyse :** après avoir capturer le trafic réseau on va faire une comparaison entre les différents paquets, l'analyse de cette comparaison va nous permettre d'extraire des empreintes numériques.
- **La détection :** C'est la dernière partie de notre recherche. On va détecter l'utilisation de l'application Orbot et le navigateur Tor en implémentant les empreintes numériques dans l'IDS **« Snort »**.

3.4 Matériel utilisé

Nous avons réalisé dans un environnement ouvert, un réseau local à domicile connecté à internet. Les expériences ont été faites sur :

- Un ordinateur portable HP avec un processeur Pentium (R) Dual-Core T4500 @ et de 2.30GHz x 2 avec une RAM 2.00 GHz.
- Un ordinateur portable Lenovo avec un processeur Intel(R) Core (TM) i3-5005U CPU@ 2.00GHz 2.00 GHz avec une RAM de 2.00 GHz.
- Système d'exploitation Windows 8.1 Professionnel 64-bits.
- Système d'exploitation Linux CentOS 8 64-bits.

Tout au long de notre recherche on utilise les logiciels suivants :

- Wireshark version 1:2.6.2-12.el8 64 bits.
- BlueStacks version 3.
- Orbot Proxy Par Tor version: 16.2.0-RC-1-tor-0.4.2.7-1-geed732a3.
- Les navigateurs Web (APK sous BlueStacks): Tor Browser version 68.11.0, Mozilla Firefox version 68.11.0 et Google Chrome 81.0.4044.138.

3.5 Environnement

3.5.1 Architecture client/serveur

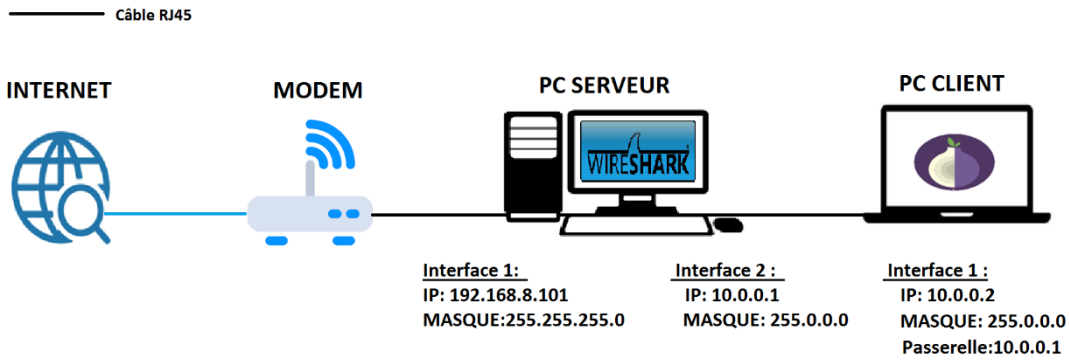


Figure3.1. Architecture client/serveur.

La figure ci-dessus nous montre l'architecture (client/serveur) réalisée.

- Le PC serveur a deux interfaces réseau.

  La première interface est connectée directement à Internet (liée avec le modem par un câble RJ45), et configurée avec l'adresse : 192.168.8.101 (adressage automatique).

  La deuxième interface est connectée au réseau local (10.0.0.0) par l'adresse : 10.0.0.1.

- Le PC client est connecté au réseau local (10.0.0.0) par l'adresse : 10.0.0.2 et la passerelle : 10.0.0.1 (adresse du pc serveur).

a. Serveur « CentOS »

Pour le serveur, on a choisi le système d'exploitation CentOS Linux pour faire notre travail. **C**ommunity **Ent**erprise **O**perating **S**ystem, en abrégé CentOS, est une distribution Linux qui est disponible depuis mai 2004. Ce projet Open Source porté par une grande communauté de développeurs est basé sur les paquets source de Red Hat Entreprise Linux (RHEL). CentOS est une plateforme entreprise principalement adaptée aux entreprises et aux grandes organisations. En principe, la distribution Linux peut aussi être utilisée pour un usage domestique, bien que ce ne soit pas le but des développeurs. De plus, la distribution Linux jouit d'une grande popularité en tant que système d'exploitation pour les serveurs Web [68].

b. Client « BlueStacks sous Windows »

On a installé l'émulateur BlueStacks sur le PC client pour pouvoir étudier le fonctionnement de l'application Orbot et les navigateurs (Tor, Chrome et Firefox).

BlueStacks est une entreprise américaine qui développe BlueStacks App Player, un émulateur Android gratuit qui offre la possibilité de profiter des applications et des jeux Android sur votre ordinateur de bureau.

L'émulateur est fidèle au système d'exploitation mobile de Google, il est donc doté pratiquement des mêmes fonctionnalités qu'un mobile Android. Il s'adresse aussi bien aux utilisateurs qui souhaitent retrouver leur environnement Android sur leur PC, qu'aux développeurs en leur permettant de voir comment se comporte leur toute nouvelle application (en installant leur fichier APK) [69].

3.6 Outil d'analyse et de capture « Wireshark »

Wireshark est l'analyseur de protocole réseau le plus utilisé au monde. Il vous permet de voir ce qui se passe sur votre réseau à un niveau microscopique. Le développement de Wireshark prospère grâce à la contribution bénévole d'experts en réseaux à travers le monde, ce développement est une continuation d'un projet lancé par Gerald Combs en 1998 [70].

Wireshark a un ensemble de fonctionnalités riches qui comprend les éléments suivants :

- Inspection approfondie de centaines de protocoles.
- Capture en direct et analyse hors ligne.
- Navigateur de paquets standard à trois volets.
- Multiplateforme : Fonctionne sous Windows, Linux, MacOS, Solaris, FreeBSD, NetBSD et bien d'autres.
- Lire/écrire différents formats de fichiers de capture : TCPdump (Libpcap), Pcap NG.
- Les données en direct peuvent être lues à partir d'Ethernet, IEEE 802.11, Bluetooth, USB et autres (selon votre plate-forme) [70].

La figure 3.2 montre la fenêtre de Wireshark après la capture de paquets et on peut la décomposer on trois sections :

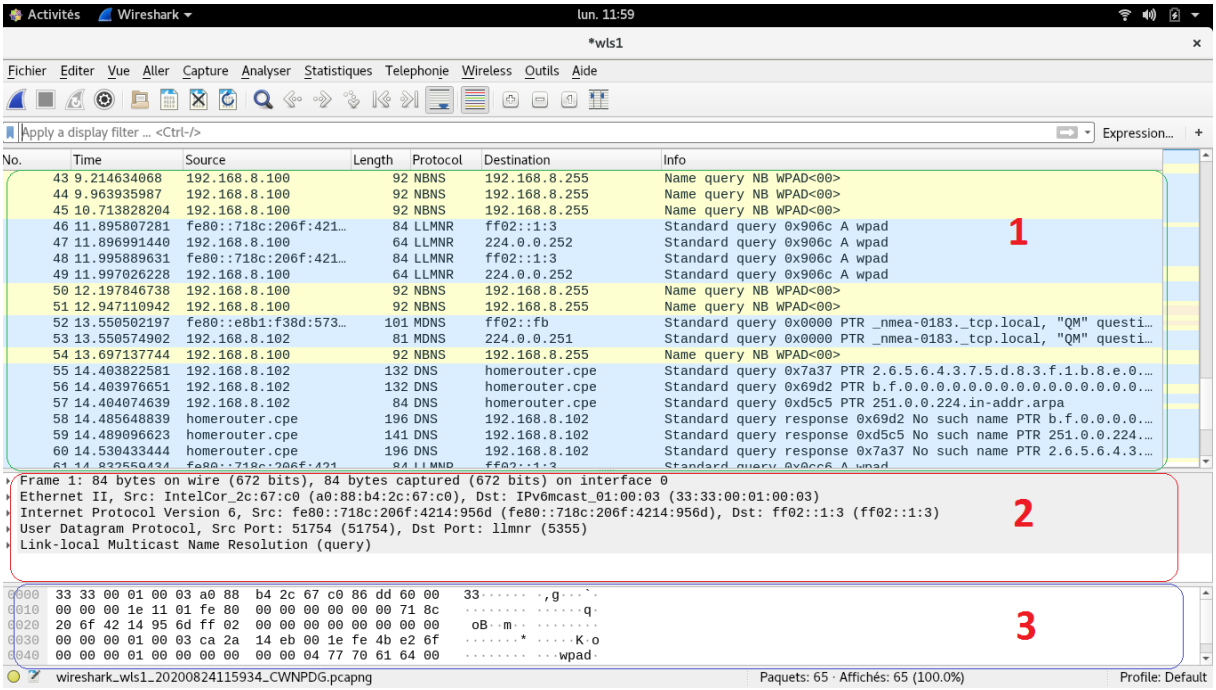


Figure 3.2. Fenêtre principale de l'interface de Wireshark.

- **Section (1) sur la figure 3.2 :** affiche la liste des paquets capturés.
- **Section (2) sur la figure 3.2 :** affiche les détails sur le paquet sélectionné de la liste du haut (le paquet surligné).
- **Section (3) sur la figure 3.2 :** reproduit le contenu en hexadécimal, du même paquet.

La capture du trafic de l'application Orbot a été effectuée sur plusieurs reprises comme montre le graphe suivant :

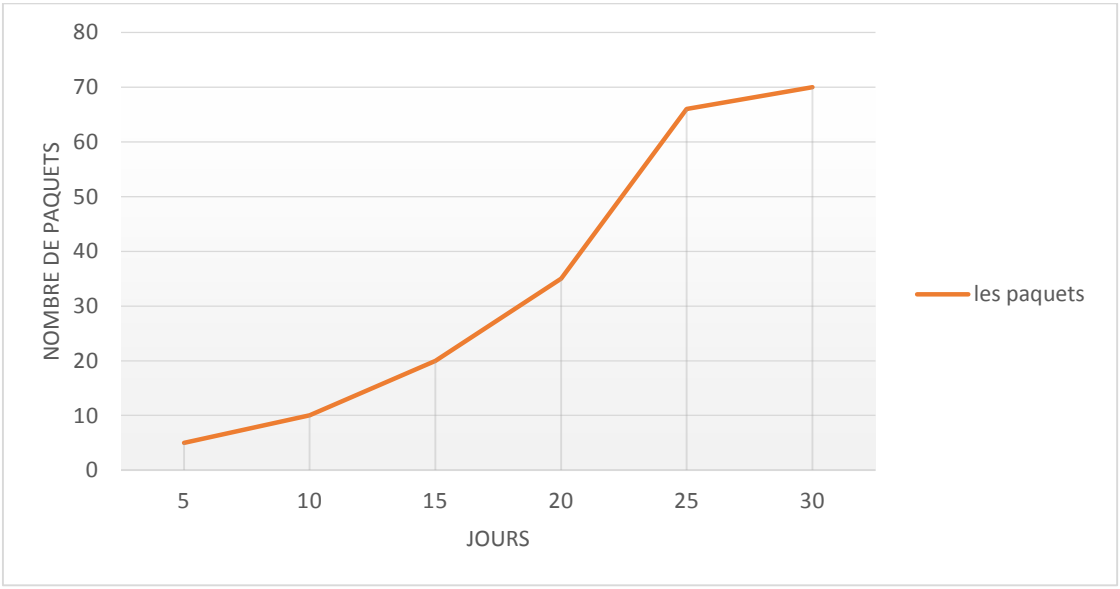


Figure 3.3. L'évolution du nombre de paquets capturés.

3.7 Analyse et capture du trafic réseau

Comme tous les réseaux anonymes, Orbot utilise le protocole TLS **(son équivalent SSL).** Ils sont tous les deux des protocoles situés entre le niveau Transport et Application. TLS établie une session chiffrée entre le client et le réseau. Dans une première phase le client et le serveur vont effectuer la négociation afin de configurer la transaction et d'échanger les clés de chiffrement. Puis, ils effectueront l'échange de donnés proprement dit. Notre travail se décompose en trois parties :

- Etude des étapes de la connexion **TCP (SYN, SYN ACK, ACK).**
- Analyse et comparaison de TCP **« Three Way Handshake ».**
- Analyse et comparaison des paquets TLS.

Comme première étape nous avons capturé et étudié les paquets TCP pour pouvoir faire la comparaison et extraire les empreintes, pour cela, on a choisi deux sites comme montre le tableau ci-dessous :

Nom de Domaine	Adresse IP
Linux.com	23.185.0.3
Ox.ac.uk	151.101.2.133

Tableau 3.1. Les adresses IP des deux sites utilisés.

On a capté le trafic Orbot avec plusieurs adresses de nœuds de garde pour avoir une bonne précision, le tableau ci-dessous contient des nœuds de Orbot et Tor.

Nœuds de Orbot			Nœuds de Tor		
Nom de l'hôte	Adresses IP	Port	Nom de l'hôte	Adresses IP	Port
electroncash.de	193.135.10.219	59999	ns3065356.IP-5-39-72.eu	5.39.72.20	9001
mail.taxinachtegel.de	163.172.47.34	9001	static.29.44.9.5.clients.your-server.de	5.9.44.29	433
None	45.95.235.197	443			

Tableau 3.2. Les adresses IP et les ports des nœuds de Orbot et Tor.

Les développeurs de Tor mettent toute une plateforme à la disposition de leurs utilisateurs pour qu'ils puissent avoir les fonctionnalités et les informations sur les nœuds. La plateforme est accessible via : https://metrics.torproject.org/rs.html#search.

Nous avons utilisé cette plate-forme pour avoir plus d'informations sur les nœuds de garde :

Les nœuds de garde de l'application Orbot :

- Le premier nœud avec l'adresse 193.135.10.219

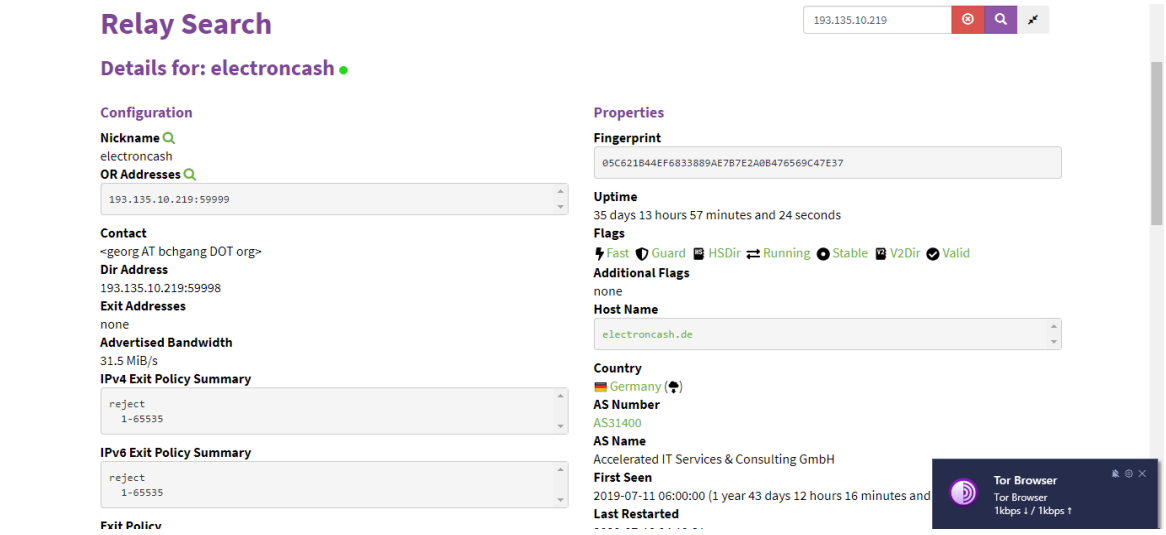


Figure 3.4. Le premier nœud Orbot.

- Le deuxième nœud avec l'adresse 163.173.47.34

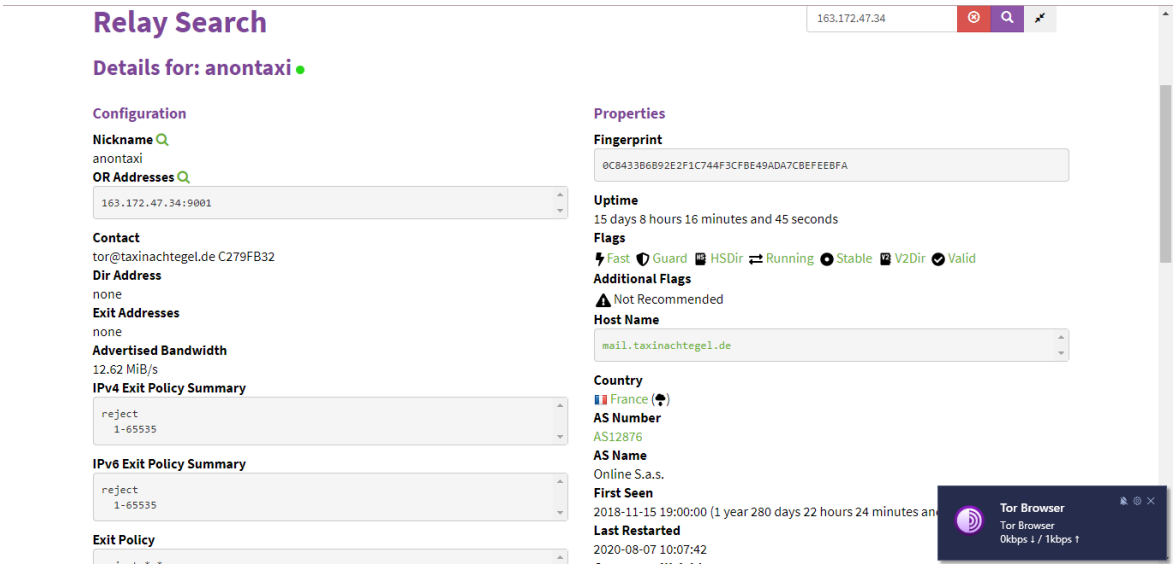


Figure 3.5. Le deuxième nœud Orbot.

- Enfin le troisième nœud avec l'adresse 45.95.235.197

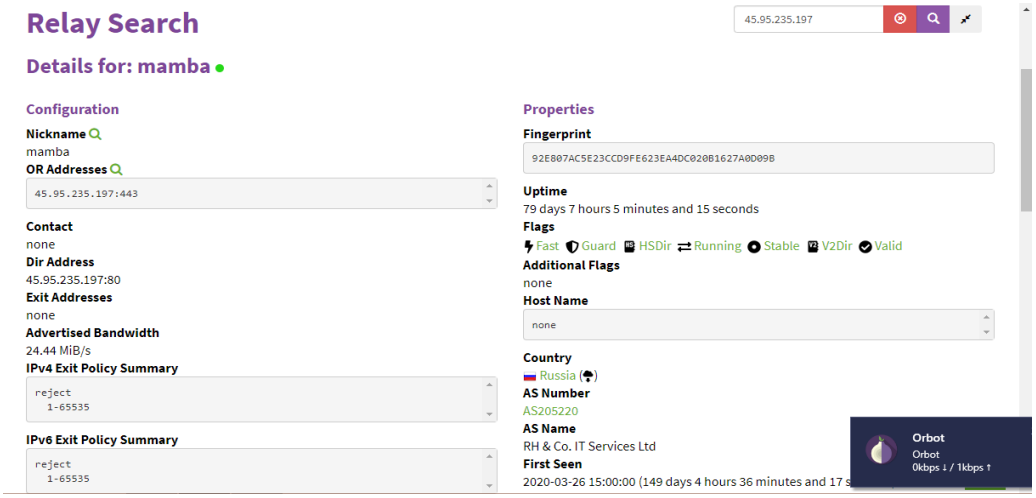


Figure 3.6. Le troisième nœud Orbot.

Les nœuds de garde du navigateur Tor (APK) :

- Le premier nœud avec l'adresse 5.39.72.20

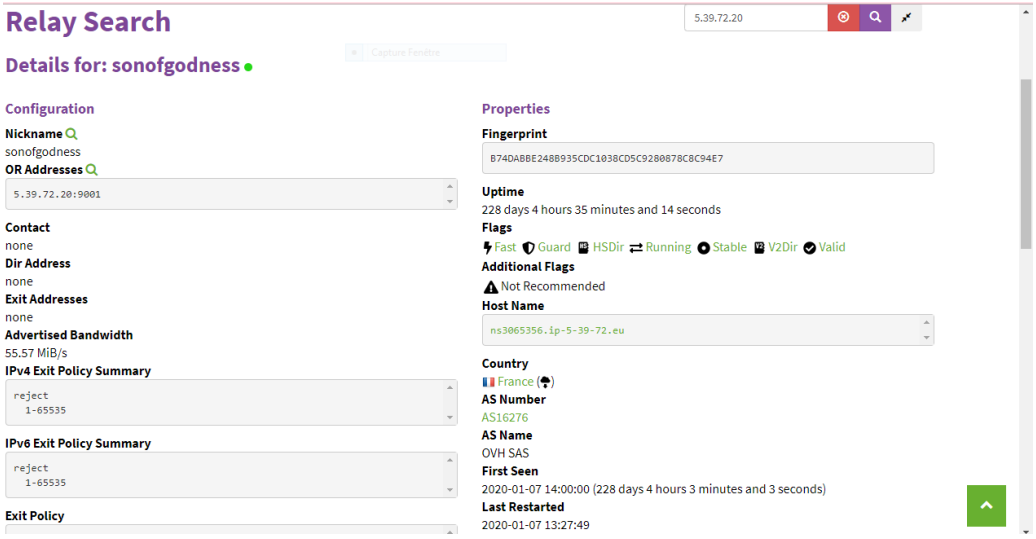


Figure 3.7. Le premier nœud Tor.

- Le deuxième nœud avec l'adresse 5.9.44.29

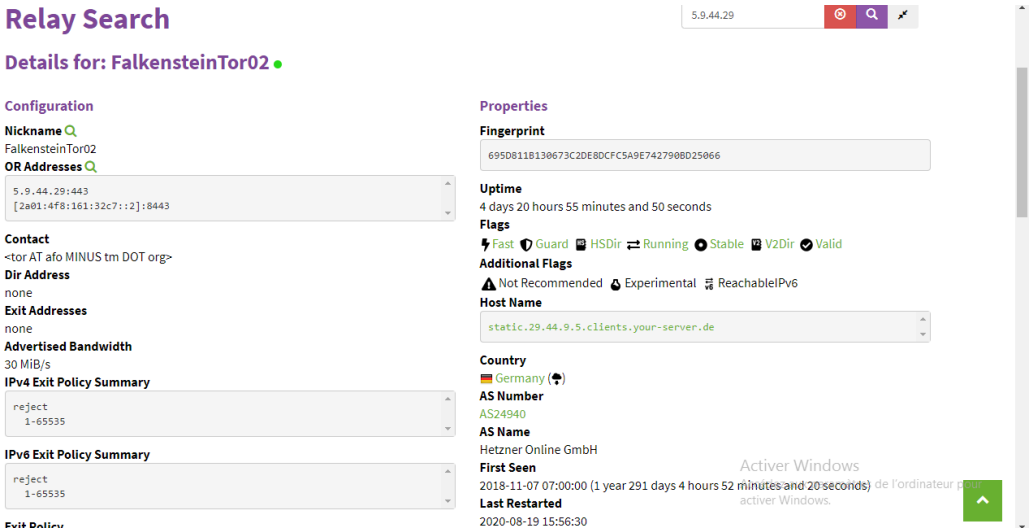


Figure 3.8. Le deuxième nœud Tor.

3.7.1 Etude des étapes de la connexion TCP « Three Way Handshake »

Une connexion TCP s'établie entre un client et un serveur en trois phases (SYN, SYN-ACK, ACK) comme cité en premier chapitre.

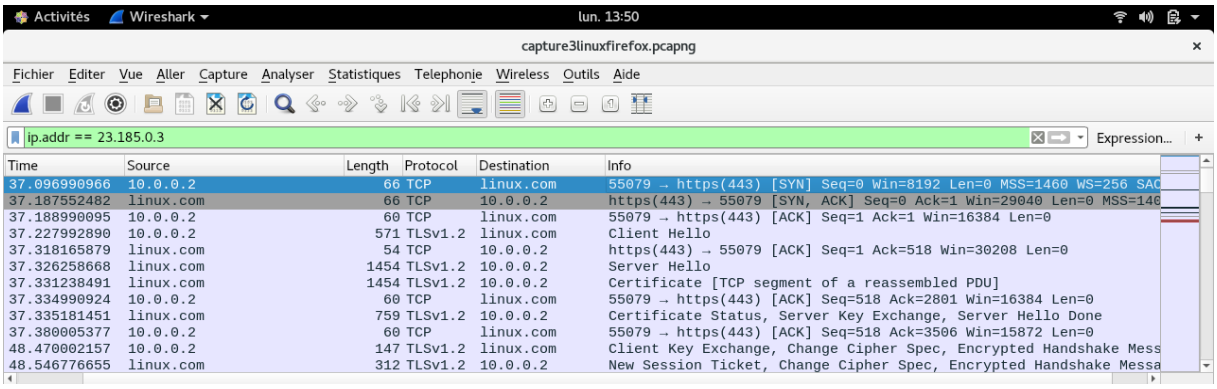


Figure 3.9. Connexion TCP entre le navigateur Firefox et le premier site.

Dans cette partie, nous allons analyser les paquets TCP (SYN, SYN-ACK, ACK) des connexions établies entre :

- L'application Orbot et le premier nœud de garde : 193.135.10.219.
- L'application Orbot et le deuxième nœud de garde : 163.172.47.34.
- L'application Orbot et le troisième nœud de garde : 45.95.235.197.
- Le navigateur Google chrome et le site : www.linux.com.
- Le navigateur Firefox et le site : www.linux.com.

- Le navigateur Tor (APK) et le nœud de garde : 5.39.72.20.
- Le navigateur Tor (APK) et le nœud de garde : 5.9.44.29.

Une fois les paquets sont analysés, on va faire une comparaison entre :

- Le résultat d'analyse de l'application Orbot avec les deux navigateurs Chrome et Firefox.
- Le résultat d'analyse du navigateur Tor (APK) avec les deux navigateurs Chrome et Firefox.

La comparaison va se faire en premier lieu entre les paquets SYN.

Lorsqu'on sélectionne un paquet **SYN**, la deuxième section de **« Wireshark »** permet de visualiser clairement les différentes couches d'encapsulation du paquet

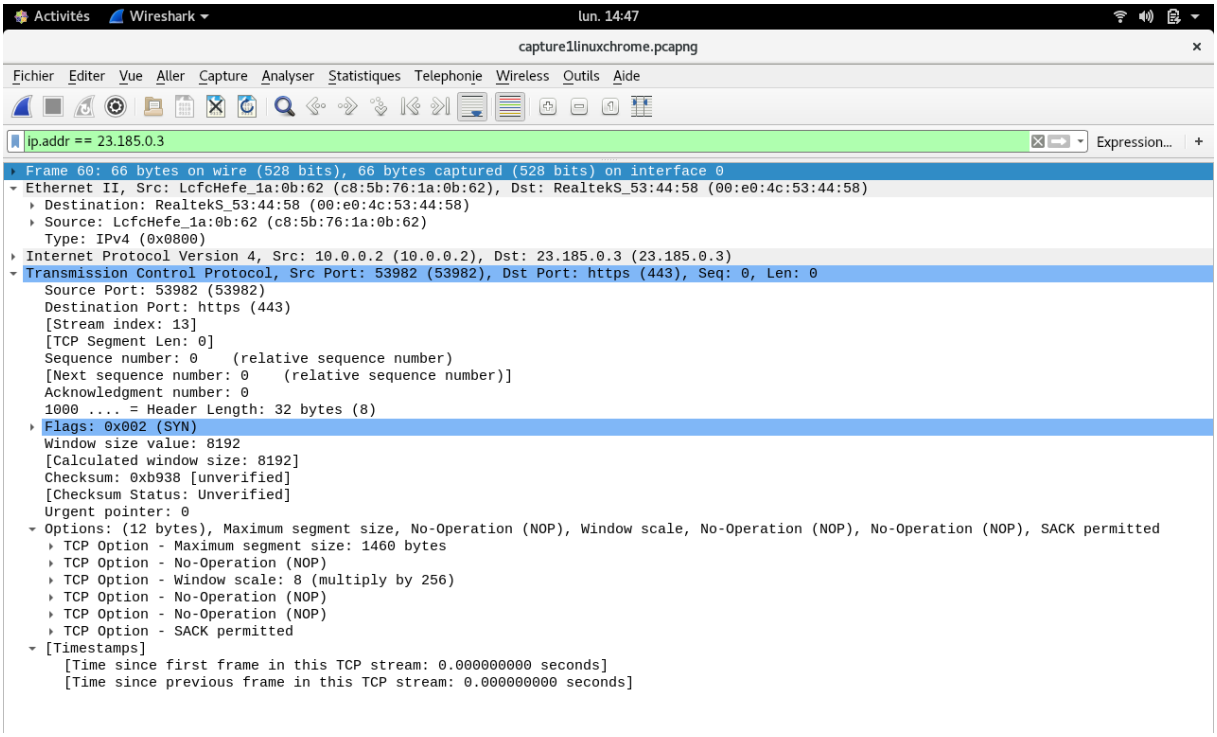


Figure 3.10. Le premier paquet TCP (SYN).

Cette section va nous permettre d'extraire les informations suivantes :

- **Total Length** : le champ Longueur totale est codé sur 16 bits et représente la longueur du paquet incluant l'entête IP et les Data associées.
- **Identification** : numéro permettant d'identifier les fragments d'un même paquet.

- **Le champ TTL (Time To Live**) : il est codé sur 8 bits et indique la durée de vie maximale du paquet. Il représente la durée de vie en seconde du paquet. Si le TTL arrive à 0, alors l'équipement qui possède le paquet, le détruira.
- **Port Source :** le champ Port source est codé sur 16 bits et indique le numéro de port de l'expéditeur.
- **Port Destination :** le champ Port destination est codé sur 16 bits et indique le numéro de port du destinataire.
- **Numéro de séquence :** le champ numéro de séquence est codé sur 32 bits et correspond au numéro du paquet. Cette valeur permet de situer à quel endroit du flux de données le paquet, qui est arrivé, doit se situer par rapport aux autres paquets.
- **Stream Index** : Stream Index ou l'index de flux est un mappage Wireshark interne.
- **Flags (6 bits)** : les 6 bits individuels disponibles au champ Flags permettent d'activer différentes actions TCP visant l'organisation de la communication et de la transmission de données. Les Flags suivants peuvent être activés ou désactivés :
  - **Le champ URG** est codé sur 1 bit et indique que le champ Pointeur de donnée urgente est utilisé.
  - **Le champ ACK** est codé sur 1 bit et indique que le numéro de séquence pour les acquittements est valide.
  - **Le champ PSH** est codé sur 1 bit et indique au récepteur de délivrer les données à l'application et de ne pas attendre le remplissage des tampons.
  - **Le champ RST** est codé sur 1 bit et demande la réinitialisation de la connexion.
  - **Le champ SYN** est codé sur 1 bit et initie donc l'activation de la connexion.
  - **Le champ FIN** est codé sur 1 bit et indique fin de transmission.
- **Window Size Value** : elle est utilisée pour indiquer à l'expéditeur la quantité de données à transmettre avant de recevoir un accusé de réception.
- **Maximum Segment Size** : MSS est la taille maximale du datagramme TCP. Il représente la taille de charge utile maximale qu'un point d'extrémité est prêt à accepter dans un seul paquet.
- **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.

Ces informations vont nous permettre de bien comparer l'application Orbot et le navigateur Tor avec les autres navigateurs afin de relever les différences qui vont nous permettre de distinguer l'utilisation de l'application Orbot et le navigateur Tor.

Afin de bien comparer les résultats d'analyse, on va comparer chaque paquet de la connexion TCP « SYN, SYN-ACK et ACK ».

a. Comparaison entre l'analyse de l'application Orbot et les navigateurs

- **Le premier paquet TCP (SYN)**

	Chrome	Firefox	Orbot 193.135.10.219	Orbot 163.172.47.34	Orbot 45.95.235.197
Type	IPv4 (0×0800)	IPv4 (0×0800)	IPv4 (0×0800)	IPv4 (0×0800)	IPv4 (0×0800)
Total Length	52	52	52	52	52
Identification	0×57d5(22485)	0×57f3(22515)	0×6d23(27939)	0×7e9c(32412)	0×5327(21287)
TTL	128	128	128	128	128
Protocole	TCP(6)	TCP(6)	TCP(6)	TCP(6)	TCP(6)
Port source	53982	55079	547001	54664	55490
Port destination	443	443	59999	9001	443
Numéro de séquence	0	0	0	0	0
Stream Index	13	9	13	2	2
Flags	0×002(SYN)	0×002(SYN)	0×002(SYN)	0×002(SYN)	0×002(SYN)
Windows Size Value	8192	8192	8192	8192	8192
Checksum (TCP)	0×b938	0×d0bf	0×913b	0×cd4d	0×c669
Maximum segment size	1460	1460	1460	1460	1460

Tableau 3.3. Le premier paquet TCP(SYN) dans l'établissement de la connexion en trois étapes.

- **Le deuxième paquet TCP (SYN-ACK)**

	Chrome	Firefox	Orbot 193.135.10.219	Orbot 163.172.47.34	Orbot 45.95.235.197
Type	IPv4 (0×0800)	IPv4 (0×0800)	IPv4 (0×0800)	IPv4 (0×0800)	IPv4 (0×0800)
Total Length	52	52	52	52	52
Identification	0×0000(0)	0×0000(0)	0×0000(0)	0×0000(0)	0×0000(0)
TTL	47	46	41	43	40
Protocole	TCP(6)	TCP(6)	TCP(6)	TCP(6)	TCP(6)
Port source	443	443	59999	9001	443
Port destination	53982	55079	52509	54664	55490
Numéro de séquence	0	0	0	0	0
Stream Index	13	9	6	2	2
Flags	0×012(SYN, ACK)	0×012(SYN, ACK)	0×012(SYN, ACK)	0×012(SYN, ACK)	0×012(SYN, ACK)
Windows Size Value	43560	29040	42340	64240	14600
Checksum (TCP)	0×7880	0×ecdb	0×ba0a	0×3c37	0×bbc1
Maximum Segment Size	1400	1400	1400	1400	1400
RTT to ACK (sec)	0.068342280	0.091999129	0.117956131	0.093825619	0.107602789

Tableau 3.4. Le deuxième paquet TCP (SYN-ACK) dans l'établissement de la connexion en trois étapes.

- **Le troisième paquet TCP (ACK)**

	Chrome	Firefox	Orbot 193.135.10.219	Orbot 163.172.47.34	Orbot 45.95.235.197
Type	IPv4 (0×0800)	IPv4 (0×0800)	IPv4 (0×0800)	IPv4 (0×0800)	IPv4 (0×0800)
Total Length	40	40	40	40	40
Identification	0×57d6(22486)	0×57f4(22516)	0×69c3(27075)	0×7e9d(32413)	0×5328(21288)
TTL	128	128	128	128	128
Protocole	TCP(6)	TCP(6)	TCP(6)	TCP(6)	TCP(6)
Port source	53982	55079	52509	54664	55490

Port destination	443	443	59999	9001	443
Numéro de séquence	1	1	1	1	1
Stream Index	13	9	6	2	2
Flags	0×010(ACK)	0×010(ACK)	0×010(ACK)	0×010(ACK)	0×010, ACK)
Windows Size Value	64	64	64	64	64
Checksum (TCP)	0×6301	0×98a4	0×9fc7	0×777e	0×3520
RTT to ACK (sec)	0.001656356	0.001437613	0.001056000	0.001175498	0.001395708

Tableau 3.5. Le troisième paquet TCP(ACK) dans l'établissement de la connexion en trois étapes.

b. Comparaison entre l'analyse du navigateur Tor (APK) et les navigateurs

- **Le premier paquet TCP (SYN)**

	Chrome	Firefox	Tor 5.39.72.20	Tor 5.9.44.29
Type	IPv4 (0×0800)	IPv4 (0×0800)	IPv4 (0×0800)	IPv4 (0×0800)
Total Length	52	52	52	52
Identification	0×57d5(22485)	0×57f3(22515)	0×10ad(4269)	0×50f3 (20723)
TTL	128	128	128	128
Protocole	TCP(6)	TCP(6)	TCP(6)	TCP(6)
Port source	53982	55079	51750	50608
Port destination	443	443	9001	443
Numéro de séquence	0	0	0	0
Stream Index	13	9	13	2
Flags	0×002(SYN)	0×002(SYN)	0×002(SYN)	0×002(SYN)
Windows Size Value	8192	8192	8192	8192
Checksum (TCP)	0×b938	0×d0bf	0×671c	0×a739
Maximum Segment Size	1460	1460	1460	1460

Tableau 3.6. Le premier paquet TCP (SYN) dans l'établissement de la connexion en trois étapes.

- **Le deuxième paquet TCP (SYN-ACK)**

	Chrome	Firefox	Tor 5.39.72.20	Tor 5.9.44.29
Type	IPv4 (0×0800)	IPv4 (0×0800)	IPv4 (0×0800)	IPv4 (0×0800)
Total Length	52	52	52	52
Identification	0×0000(0)	0×0000(0)	0×0000(0)	0×0000(0)
TTL	47	46	40	44
Protocole	TCP(6)	TCP(6)	TCP(6)	TCP(6)
Port source	443	443	9001	443
Port destination	53982	55079	51750	50608
Numéro de séquence	0	0	0	0
Stream Index	13	9	2	3
Flags	0×012(SYN, ACK)	0×012(SYN, ACK)	0×012(SYN, ACK)	0×012(SYN, ACK)
Windows Size Value	43560	29040	29200	29200
Checksum (TCP)	0×7880	0×ecdb	0×b892	0×486
Maximum Segment Size	1400	1400	1400	1400
RTT to ACK (sec)	0.068342280	0.091999129	0.075202789	0.095517384

Tableau 3.7. Le deuxième paquet TCP (SYN-ACK) dans l'établissement de la connexion en trois étapes.

- **Le troisième paquet TCP (ACK)**

	Chrome	Firefox	Tor 5.39.72.20	Tor 5.9.44.29
Type	IPv4 (0×0800)	IPv4 (0×0800)	IPv4 (0×0800)	IPv4 (0×0800)
Total Length	40	40	40	40
Identification	0×57d6(22486)	0×57f4(22516)	0×10ae(4270)	0×50f4(20724)
TTL	128	128	128	128
Protocole	TCP(6)	TCP(6)	TCP(6)	TCP(6)
Port source	53982	55079	51750	50608

Port destination	443	443	9001	443
Numéro de séquence	1	1	1	1
Stream Index	13	9	2	3
Flags	0×010(ACK)	0×010(ACK)	0×010(ACK)	0×010(ACK)
Windows Size Value	64	64	64	64
Checksum (TCP)	0×6301	0×98a4	0×6af9	0×fadb
RTT to ACK (sec)	0.001656356	0.001437613	0.000799054	0.001480007

Tableau 3.8. Le troisième paquet TCP(ACK) dans l'établissement de la connexion en trois étapes.

c. Constatations et remarques

Il y a une différence entre :

- **Identification :** on remarque que la valeur change d'un paquet à l'autre. Le changement de cette valeur est tout à fait normal car elle est représentée par un entier unique qui identifie le datagramme, et qui est recopié dans chaque fragment.
- **Port source :** d'une connexion à une autre, le port change d'une façon dynamique relatif à l'application en cours sur la machine source.
- **Port destination :** ce port change relativement à l'application en cours sur la machine de destination.
  - Les nœuds d'entrée de l'application Orbot utilisent les ports **9001** et **59999** à la place du port **443** « le port par défaut HTTPS ».
  - Les nœuds d'entrée du navigateur Tor utilisent le port **9001** à la place du port **443** « le port par défaut HTTPS ».
- **Stream Index :** on voit que sa valeur change pour chaque paquet, mais on ne peut pas prendre ce changement en considération car c'est un mappage interne de Wireshark.
- **Checksum TCP :** on remarque que sa valeur change d'un paquet à l'autre donc c'est un critère incomparable pour retirer les empreintes. Le Checksum couvre de plus, un pseudo entête. Cette dernière comporte les adresses Internet source et destination, le type de protocole et la longueur du message TCP (incluant la donnée). Ceci protège TCP contre les erreurs de routage.

- **RTT to ACK :** ces informations varient généralement d'un paquet à l'autre (SYN-ACK, ACK). C'est le temps entre l'émission d'un paquet et la réception de l'ACK correspondant.

Les résultats de cette analyse nous permettent d'approuver que :

- Il n'y a pas des différences entre les paquets TCP des navigateurs et les paquets TCP de (Orbot et Tor), qui nous permettent d'avoir des empreintes fiables qui vont être à la suite des signatures lors de la détection. Pour cela on doit passer à l'étude des paquets TLS.

3.7.2 Etude des étapes de la connexion TLS « protocole d'établissement de connexion SSL « SSL Handshake »

TLS est un protocole de chiffrement conçu pour sécuriser les communications Internet. Un Handshake TLS est le processus qui entame une session de communication utilisant le chiffrement TLS. Au cours d'un Handshake TLS, les deux parties communicantes échangent des messages d'authentification et de vérifications mutuelles, établissent les algorithmes de chiffrement qu'elles utiliseront et se mettent d'accord sur les clés de session. Les Handshakes TLS constituent une partie fondamentale du mode de fonctionnement du protocole HTTPS [71].

Voici en détails comment se déroule le Handshake, dans l'ordre chronologique :

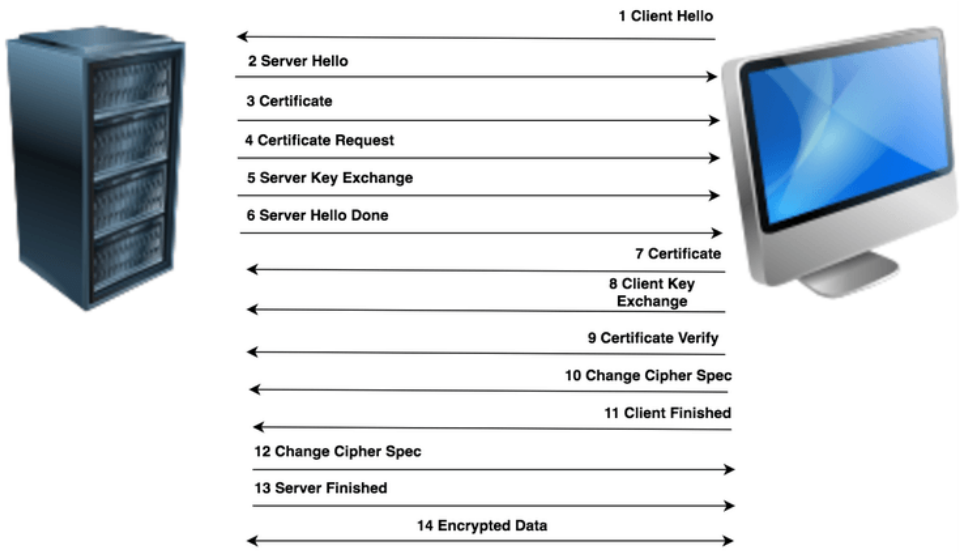


Figure 3.11. Le déroulement du TLS Handshake dans l'ordre chronologique [72].

1. **Client Hello :** le client contacte le serveur.
2. **Server Hello :** le serveur envoie les informations nécessaires pour la communication avec le client : Version des protocoles SSL, certificats serveurs (clé publique) … etc.
3. **Certificate (optionnel) :** envoi d'une chaîne de certificats par le serveur. Le premier certificat est celui du serveur, le dernier est celui de l'autorité de certification.
4. **Certificate Request (optionnel) :** demande un certificat au client pour l'authentifier.
5. **Server Key Exchange (optionnel) :** message complémentaire pour l'échange des clés. Ce message contient la clé publique du serveur utilisée par le client pour chiffrer les informations de clé de session.
6. **Server Hello Done :** Indique la fin de l'envoi de message.
7. **Certificate (optionnel) :** certificat éventuel du client si le serveur demande une authentification.
8. **Client Key Exchange :** le client produit un secret pré-maître (Encrypted Pre-master Key) et le crypte avec la clé publique du certificat du serveur.
9. **Certificate Verify (optionnel) :** message contenant une empreinte signée numériquement et créée à partir des informations de clé et de tous les messages précédents.
10. **Change Cipher Spec :** passage du client en mode chiffré avec la clé master comme clé symétrique.
11. **Client Finished :** fin des émissions du client, ce message est chiffré à l'aide des paramètres de la suite de chiffrement.
12. **Change Cipher Spec** : passage du serveur en mode chiffré avec la clé master.
13. **Server Finished :** confirmation au client du passage en mode chiffré. Ce message est chiffré à l'aide des paramètres de la suite de chiffrement.
14. **Encrypted Data :** le tunnel SSL / TLS est établi.

Après avoir détailler le fonctionnement du protocole TLS, on va commencer l'analyse de capture des paquets TLS **« connexion SSL Handshake »**. Pour faciliter le recherche des paquets TLS dans la capture, on va utiliser le filtre **« SSL.Handshake and ip.addr == adresse du serveur ».** Les figures 3.12 et 3.13 montrent les messages échangés lors de l'établissement d'une session sécurisée **« SSL Handshake ».**

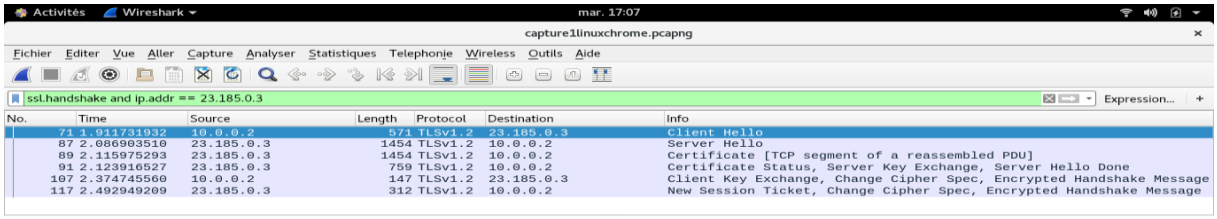


Figure 3.12. Connexion SSL.Handshake entre le navigateur Chrome et le premier site.

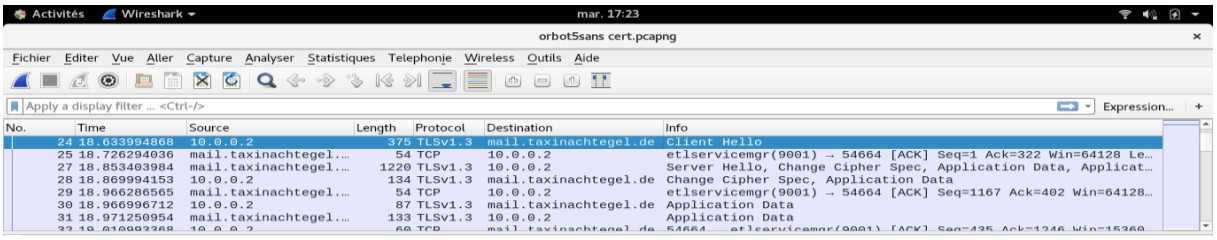


Figure 3.13. Connexion SSL.Handshake entre l'application Orbot et le nœud de garde 163.172.47.34.

Dans cette partie, nous allons analyser les paquets échangés lors de l'établissement d'une session sécurisée « SSL Handshake », des connexions établies entre :

- L'application Orbot et le premier nœud de garde : 193.135.10.219.
- L'application Orbot et le deuxième nœud de garde : 163.172.47.34.
- L'application Orbot et le troisième nœud de garde : 45.95.235.197.
- Le navigateur Google chrome et le site : www.linux.com.
- Le navigateur Firefox et le site : www.linux.com.
- Le navigateur Tor (APK) et le nœud de garde : 5.39.72.20.
- Le navigateur Tor (APK) et le nœud de garde : 5.9.44.29.

Dans notre recherche, on va se focaliser seulement à l'analyse des paquets **« client Hello »**, **« server Hello »** et **« Certificate ».**

a. Comparaison entre l'analyse de (l'application Orbot, le navigateur Tor) et les navigateurs

❖ Client hello

Les figures 3.13, 3.14 et 3.15 nous montrent la différence dans le premier paquet de la connexion « SSL.Handshake » client hello entre :

- Le navigateur chrome et l'application Orbot.
- Le navigateur chrome et Tor APK.

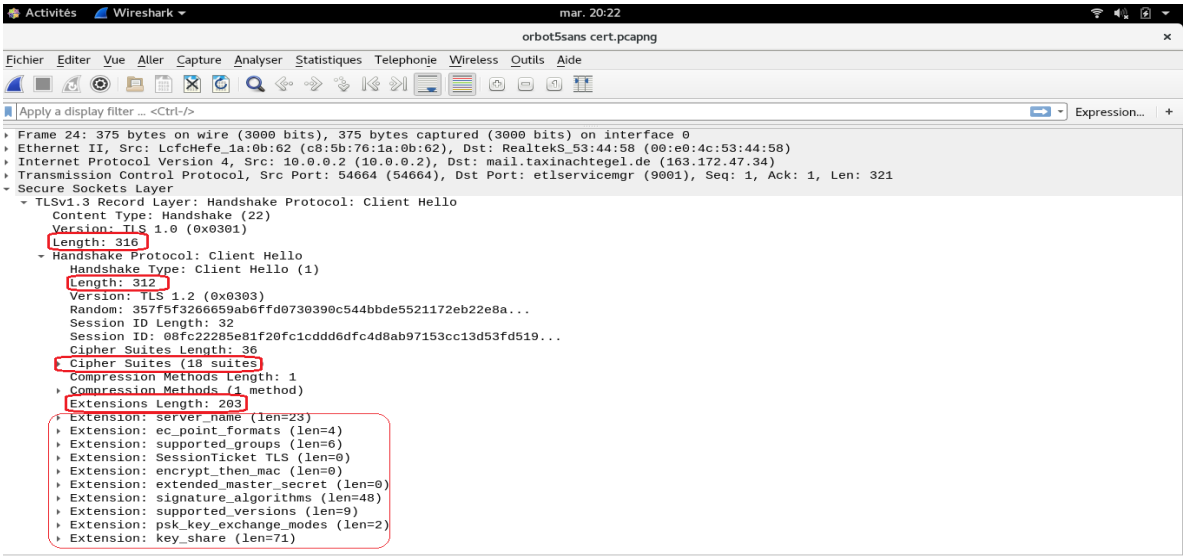


Figure 3.14. Le paquet client hello de la communication entre Orbot et le nœud de garde 163.172.47.34.

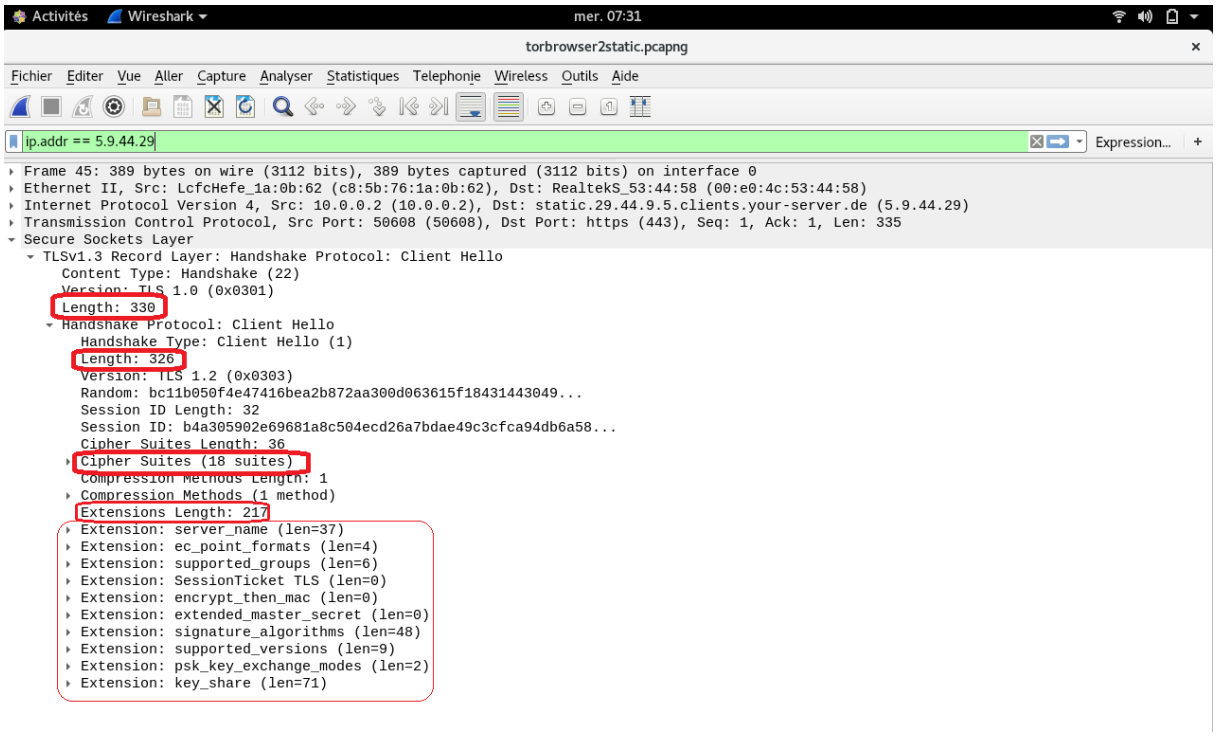


Figure 3.15. Le paquet client hello de la communication entre Tor APK et le nœud de garde 5.9.44.29.

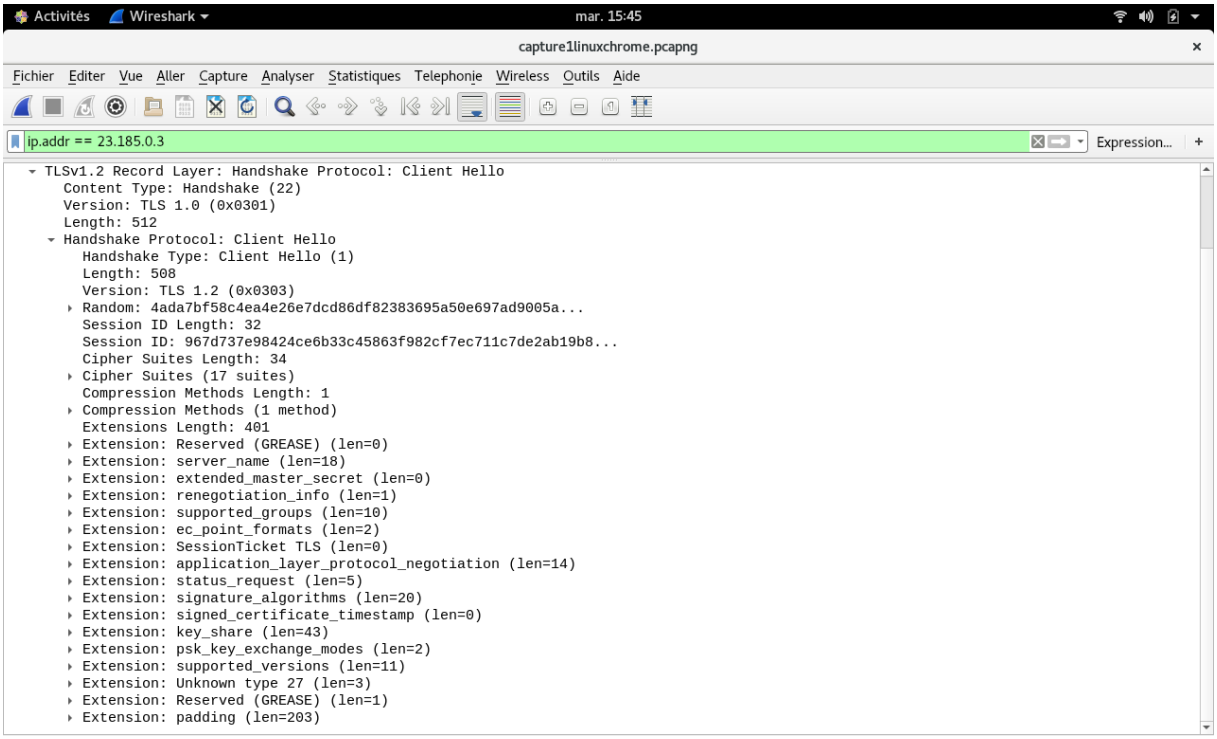


Figure 3.16. Le paquet client hello de la communication entre le navigateur Chrome et le premier site.

- **La longueur totale (Length) :** on remarque que la longueur totale Length du paquet client hello envoyée par l'application Orbot et le navigateur Tor est largement inférieure par rapport à celle du paquet client hello envoyé par les navigateurs Chrome et Firefox qui est fixée à 512. Cependant, on a trouvé que la longueur totale du paquet envoyé par Orbot est entre 310 et 323 et la longueur totale du paquet envoyé par Tor est entre 316 et 330.
- **Cipher suites :** pour le navigateur Chrome, il contient 17 Cipher suites, le navigateur Firefox contient 18 Cipher suites.

  L'application Orbot et le navigateur Tor (APK) contiennent 18 Cipher suites dans un ordre bien précis, et ils contiennent 5 Cipher suites différentes par rapport aux navigateurs Chrome et Firefox.

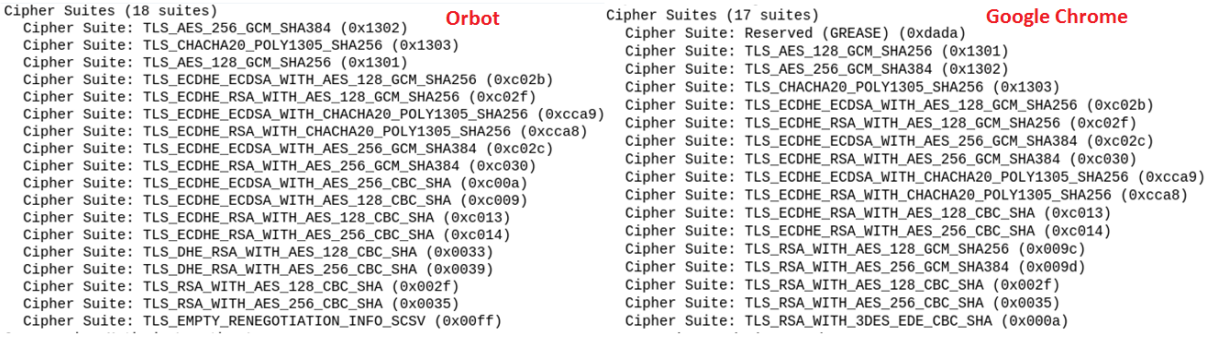


Figure 3.17. Les suites de chiffrement proposées par l'application Orbot et le navigateur Google Chrome.

Le tableau ci-dessous représente les cinq suites de chiffrement de l'application Orbot et du navigateur Tor.

Les suites de chiffrement de l'application Orbot
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0×c00a)
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0×c009)
TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0×0033)
TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0×0039)
TLS_EMPTY_RENEGOTIATION_INFO_SCSV (0×00ff)

Tableau 3.9. Les suites de chiffrement « Cipher suites » spécifiques de l'application Orbot et du navigateur Tor.

- **Les extensions**

Un navigateur peut contenir plusieurs extensions. Le client indique quelles extensions il prend en charge pendant la poignée de main (dans le message client hello) et le serveur choisit quelles extensions utiliser à sa seule discrétion.

On constate que les 2 navigateurs plus Orbot et Tor utilisent les extensions. Orbot et Tor utilisent 10 extensions, Firefox utilise 14 et Chrome utilise 17 comme montre les figures ci-dessous.

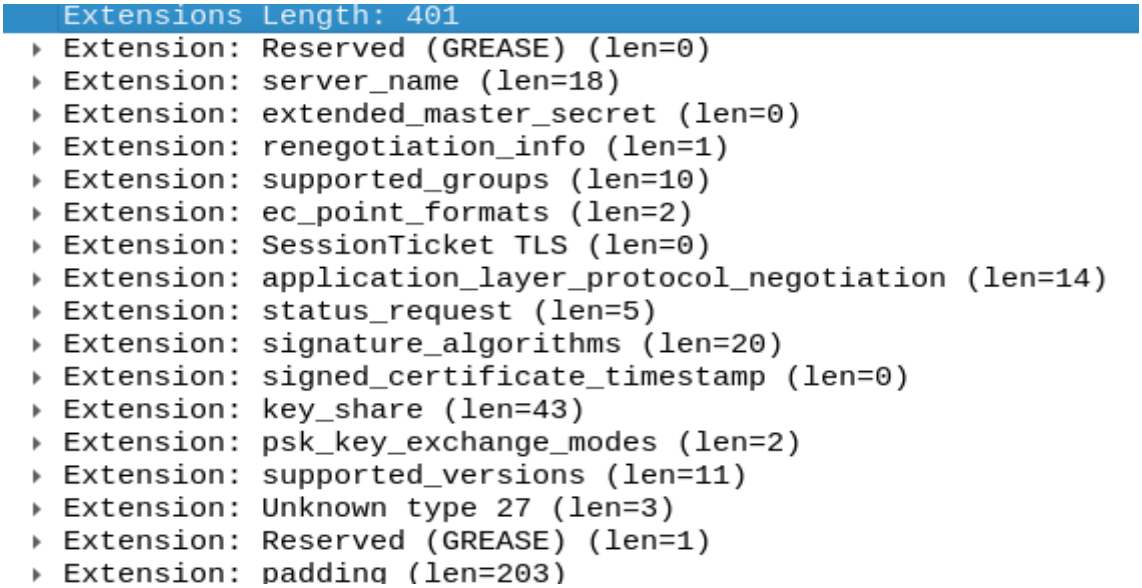


Figure 3.18. Les extensions du navigateur Chrome.

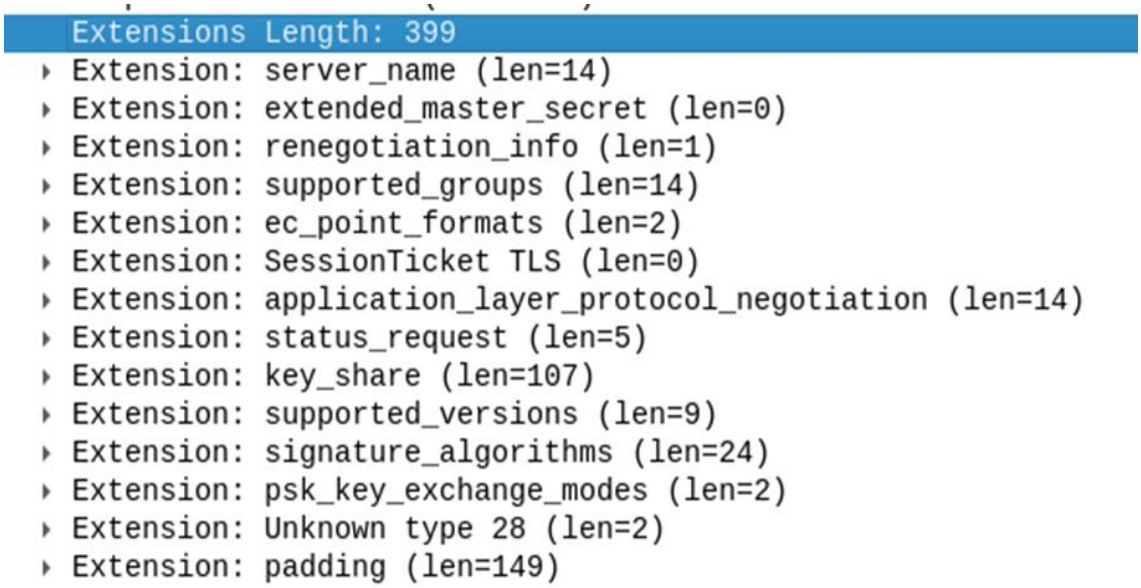


Figure 3.19. Les extensions du navigateur Firefox.

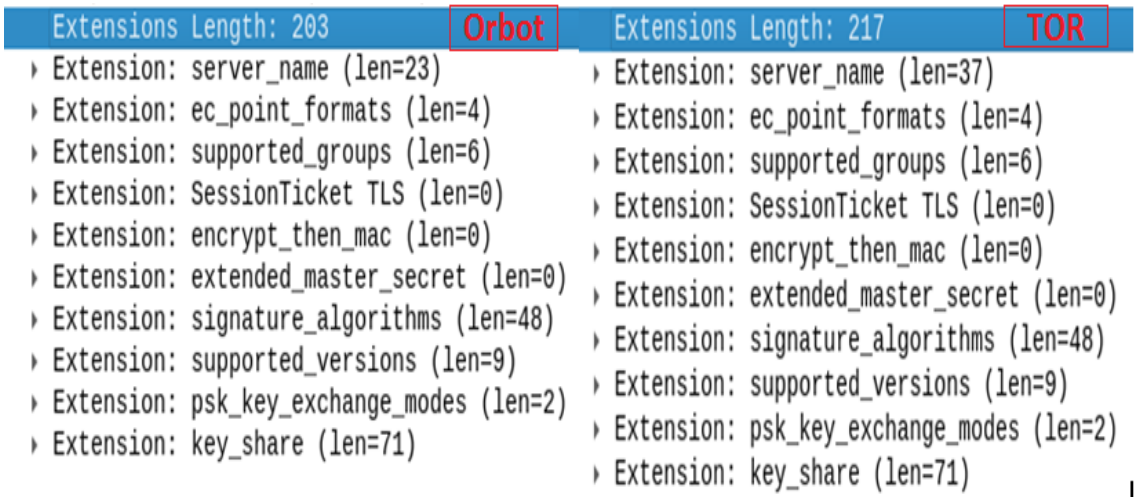


Figure 3.20. Les extensions de l'application Orbot et du navigateur Tor.

Passant par plusieurs captures de Orbot et Tor, on remarque que :

- Extension Length de l'application Orbot est entre 197 et 210 qui est largement inférieur par rapport aux autres navigateurs.
- Extension Length du navigateur Tor est entre 203 et 217 qui est largement inférieur par rapport aux autres navigateurs.

On voit aussi qu'il y a des extensions communes entre (Orbot et Tor), Chrome et Firefox et une extension spécifique pour (Orbot et Tor) (encadrée en vert).

- **Extension « encrypt_then_mac »**

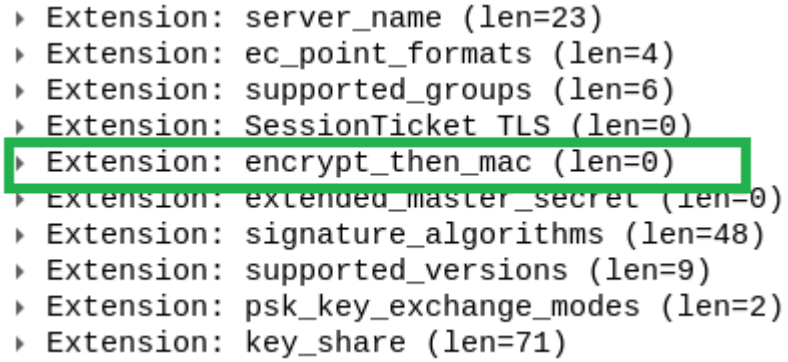


Figure 3.21. L'extension spécifique pour Orbot et Tor (APK).

La figure ci-dessous nous donne plus d'informations sur cette extension

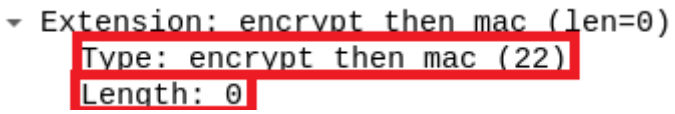


Figure 3.22. L'extension encrypt_then_mac.

- **Extension « server_name » :** est une extension du protocole TLS (anciennement connu sous le nom de protocole SSL) qui est utilisé dans HTTPS. Elle est incluse dans le processus de Handshake TLS/SSL afin de s'assurer que les périphériques client sont en mesure de voir le certificat SSL correct du site web qu'ils tentent d'atteindre. L'extension permet de préciser le nom d'hôte, ou nom de domaine, du site web pendant le Handshake TLS plutôt qu'au moment de l'ouverture de la connexion HTTP après le Handshake.

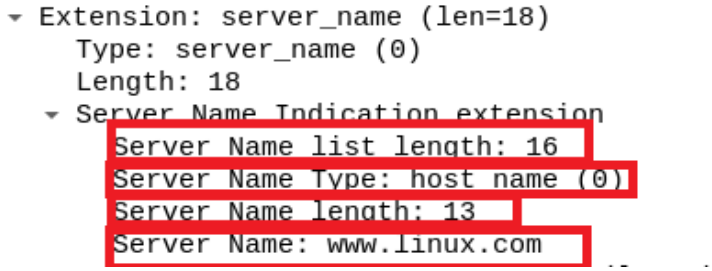


Figure 3.23. L'extension server_name du navigateur Chrome.

Dans le cas de Orbot et Tor (APK), on trouve un URL généré de façon aléatoire.

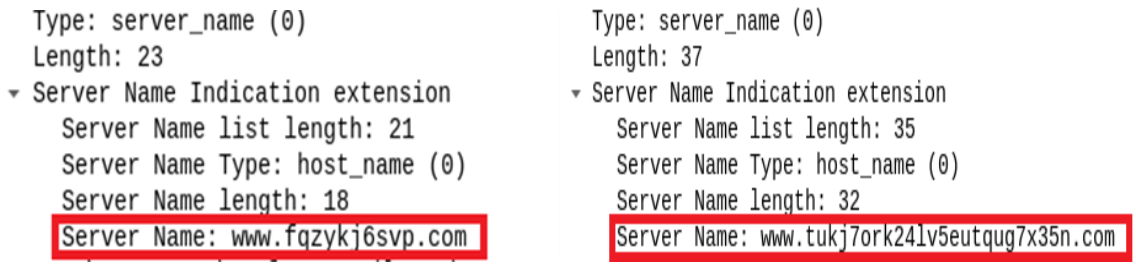


Figure 3.24. L'extension server_name de l'application Orbot et le navigateur Tor (APK).

On doit vérifier si le nom de domaine délivré par l'application Orbot et le navigateur Tor (APK) existe réellement avec la commande suivante :

Figure 3.25. Nom de domaine délivré par l'application Orbot.

Figure 3.26. Nom de domaine délivré par le navigateur Tor.

Les figures ci-dessus nous montrent que le nom de domaine généré par Orbot et Tor n'existe pas.

- **Extension « signature_algorithms » :** cette extension contient les algorithmes de hachage supportés par le navigateur. L'application Orbot et le navigateur Tor (APK) contient 23 algorithmes, le navigateur Chrome contient 9 algorithmes et le navigateur Firefox contient 11 algorithmes.

La figure ci-dessous nous montre les algorithmes supportés par l'application Orbot (le navigateur Tor contient exactement les mêmes algorithmes).

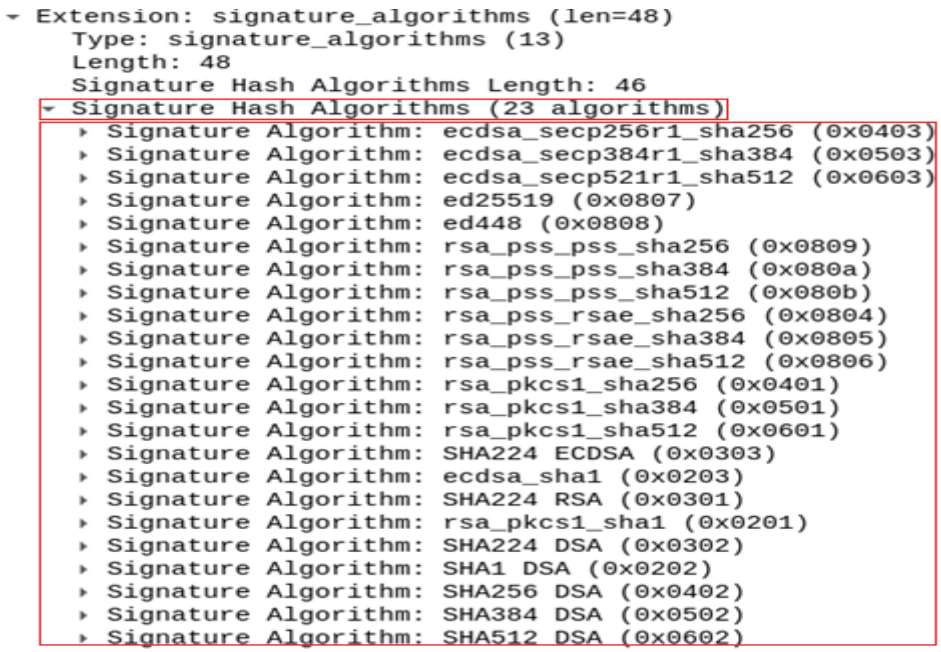


Figure 3.27. L'extension signature_algorithms envoyée par l'application Orbot.

- **Extension « supported_groups » :** Cela définit les groupes pris en charge. Pour les clients, les groupes sont envoyés à l'aide de l'extension supported_groups. Pour les serveurs, il est utilisé pour déterminer le groupe à utiliser. Ce paramètre affecte les groupes utilisés pour les signatures et l'échange de clés **«** négociations de FFDHE (l'algorithme Diffie-Hellman est un algorithme d'échange de clés) **»**. Le premier groupe répertorié sera également utilisé pour le partage de clé envoyé par un client dans un client hello.

  L'application Orbot et le navigateur Tor contiennent 2 groupes, le navigateur Firefox contient 6 groupes et le navigateur Google Chrome a 4 groupes. La valeur du Length de Orbot et Tor (APK) est inférieure à celle des autres navigateurs.

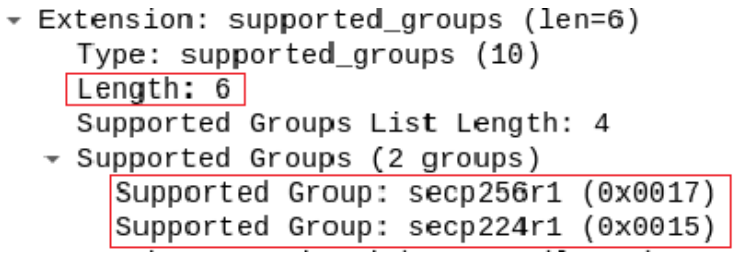


Figure 3.28. L'extension supported_groups envoyée par l'application Orbot.

Le navigateur Tor (APK) contient exactement les mêmes informations montrées par la figure ci-dessus.

❖ **Server hello**

Les figures 3.29, 3.30 et 3.31 nous montrent la différence dans le deuxième paquet de la connexion « SSL.Handshake » server hello entre : (l'application Orbot, le navigateur Tor) et les autres navigateurs.

Les paquets « server hello » des navigateurs Chrome et Firefox avec les deux sites sont identiques.

En analysant le trafic Orbot venant des trois nœuds et le trafic Tor venant des deux nœuds, on a remarqué que les informations du paquet « server hello » sont différentes. Les paquets des nœuds (1 et 2 de Orbot) et les paquets du nœud 2 de Tor sont similaires. En revanche, le paquet du troisième nœud Orbot et le paquet du nœud 1 Tor contiennent quelques détails différents.

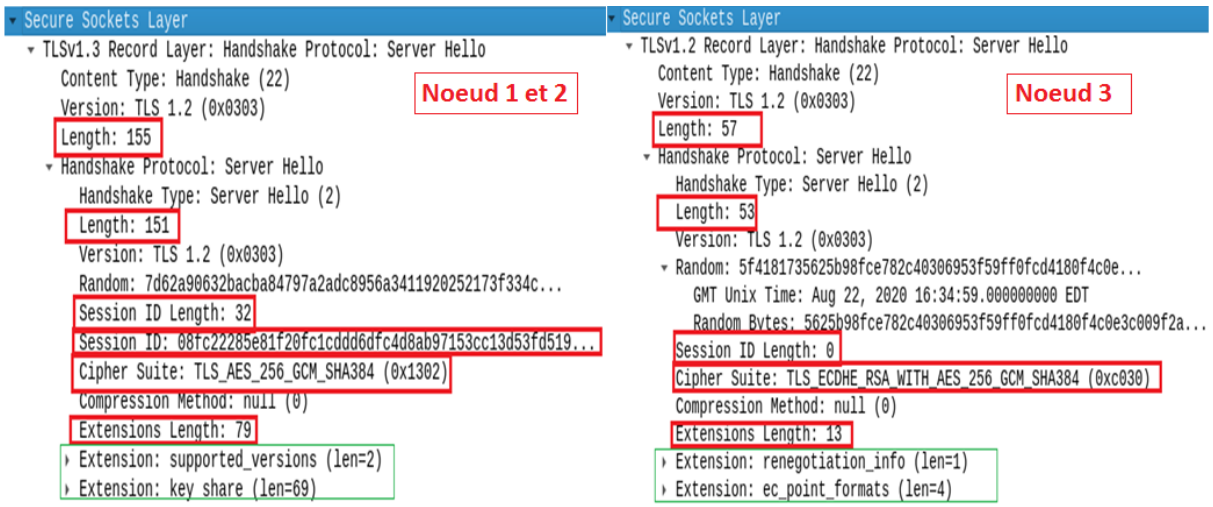


Figure 3.29. Le paquet server hello de la communication entre Orbot et les trois nœuds de garde.

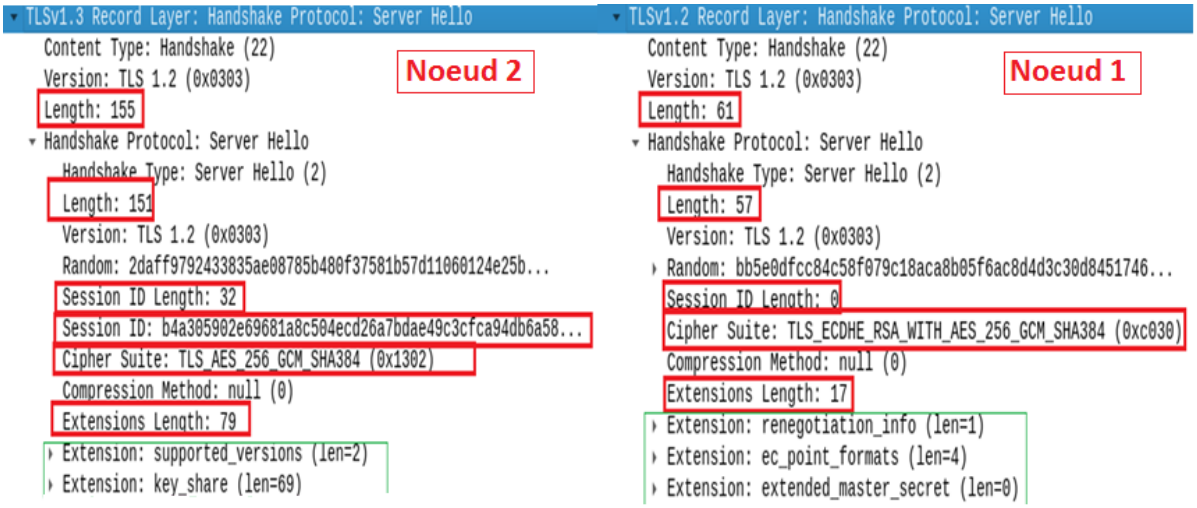


Figure 3.30. Le paquet server hello de la communication entre Tor (APK) et les deux nœuds de garde.

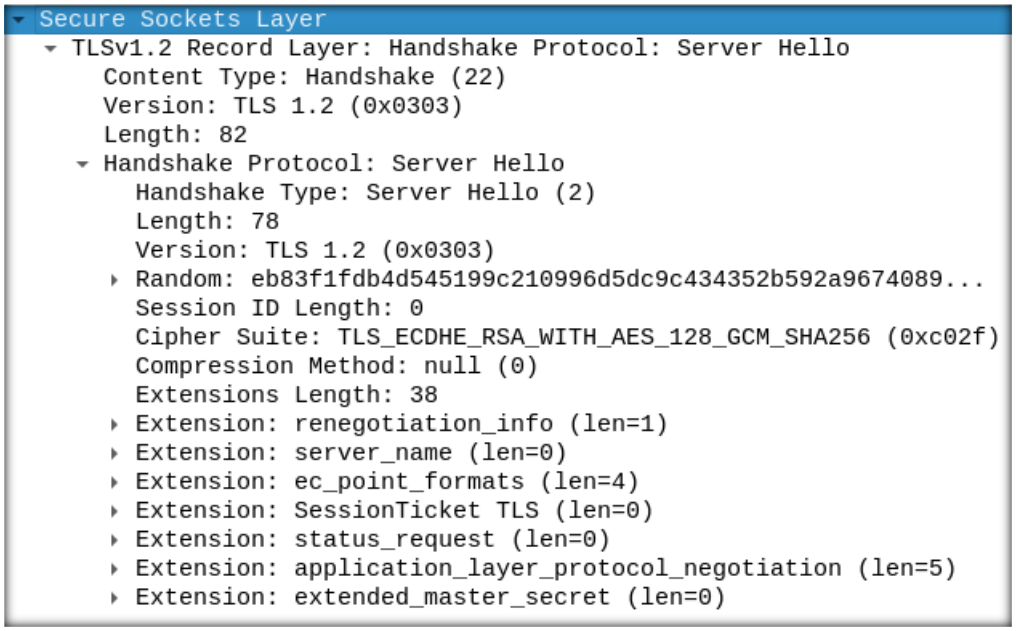


Figure 3.31. Le paquet server hello de la communication des deux navigateurs avec les deux sites.

- **Session id Length :** sa valeur est égale à 0 pour les deux navigateurs avec les deux sites, le premier nœud Tor (APK) et le troisième nœud Orbot. Elle est égale à 32 pour l'application Orbot avec les nœuds (1 et 2) et le deuxième nœud Tor.
- Le paquet server hello de la connexion avec les nœuds (1 et 2) et le nœud 2 de Tor contient l'information **« Session id »** qui est spécifique pour le paquet de l'application Orbot et le navigateur Tor.
- Les nœuds (1 et 2) de Orbot et le nœud 2 de Tor utilisent les extensions **« supported_version »** et **« key_share ».** Par contre, les paquets de la connexion entre

les navigateurs et les deux sites contiennent sept extensions différentes des deux précédentes.

- La longueur totale **« Length »** dans les paquets des nœuds (1 et 2) de Orbot et le deuxième nœud de Tor est constante et égale à 155.
- Les suites de chiffrement choisies par les serveurs sont mentionnées dans le tableau suivant :

Serveur destination	Suites de chiffrement
Nœud d'entré Orbot (1)	TLS_AES_256_GCM_SHA384
Nœud d'entré Orbot (2)	TLS_AES_256_GCM_SHA384
Nœud d'entré Orbot (3)	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
www.linux.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
www.ox.ac.uk	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
Nœud d'entré Tor (APK) (1)	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
Nœud d'entré Tor (APK) (2)	TLS_AES_256_GCM_SHA384

Tableau 3.10. Les différentes suites de chiffrement utilisées par les nœuds de Tor et Orbot et les deux sites.

Les suites de chiffrements utilisés par les nœuds de Orbot sont présentées comme suit :

- **TLS_AES_256_GCM_SHA384 :** cette suite contient pas mal d'informations sur les techniques de cryptage et de chiffrement. Elle est traduite par l'utilisation de TLS, de AES_256 cryptage symétrique, ce qui signifie qu'il crypte et décrypte les données, il utilise une clé cryptographique spécifique, qui est effectivement un ensemble de protocoles pour masquer les informations. Cette clé a une taille de 256 bits, SHA384 pour la vérification de l'intégrité des données. Elle est utilisée par Orbot dans les deux premiers nœuds (1 et 2) et dans le deuxième nœud Tor.
- **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 :** elle se base sur le protocole TLS et se traduit par l'utilisation de Diffie-Hellman, RSA pour le cryptage asymétrique, le AES_256_GCM c'est la clé utilisée dans le mode de chiffrement par chaines de blocs et SHA384 pour la vérification. Cette suite est utilisée par le troisième nœud de Orbot et le premier nœud de Tor.

❖ Certificate

Les deux premiers nœuds de l'application Orbot et le deuxième nœud du navigateur Tor ne contiennent pas le paquet « Certificate » comme montré pas les figures 3.32, 3.33 et 3.34.

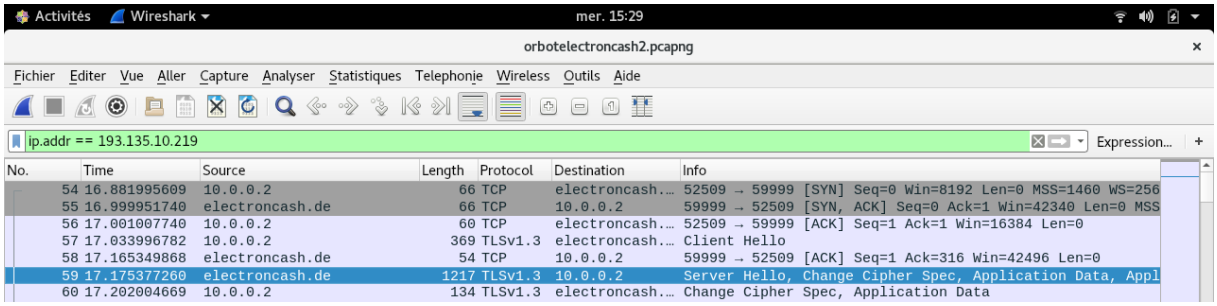


Figure 3.32. La communication entre Orbot et le premier nœud de garde 193.135.10.219.

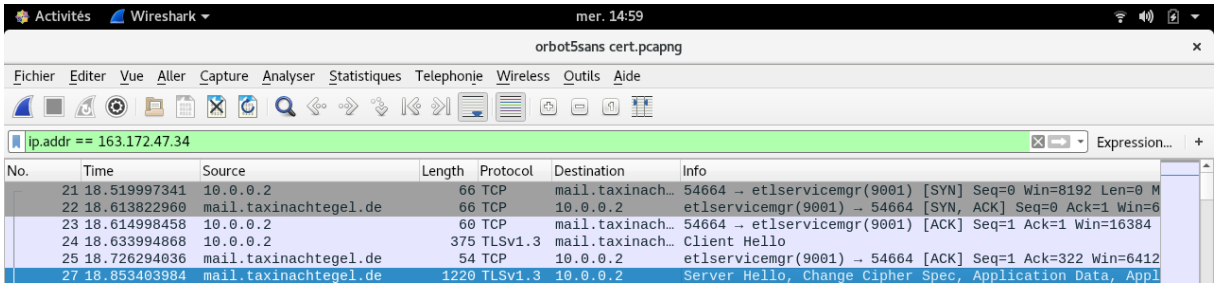


Figure 3.33. La communication entre Orbot et le deuxième nœud de garde 163.172.47.34.

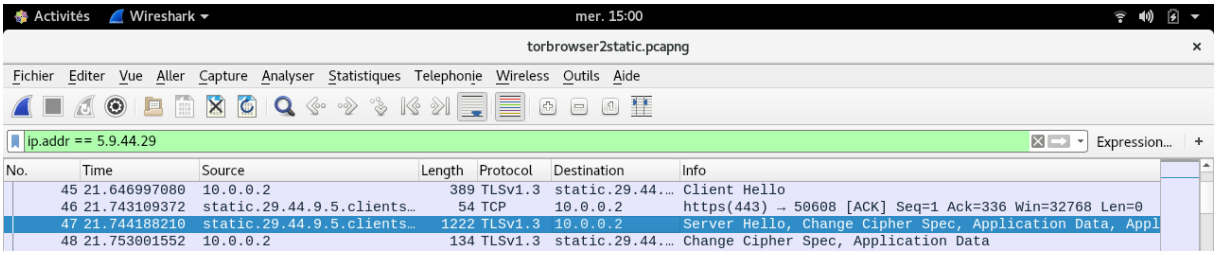


Figure 3.34. La communication entre le navigateur Tor et le deuxième nœud de garde 5.9.44.29.

Le troisième nœud de Orbot et le premier nœud du navigateur Tor contiennent le paquet **« Certificate »**, on va les comparer avec les paquets de la connexion entre les deux autres navigateurs et les deux sites.

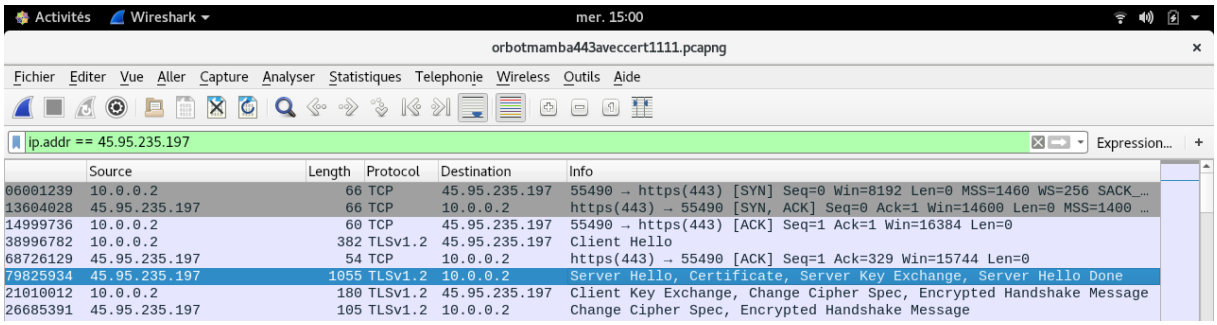


Figure 3.35. La communication entre l'application Orbot et le troisième nœud de garde 45.95.235.197.

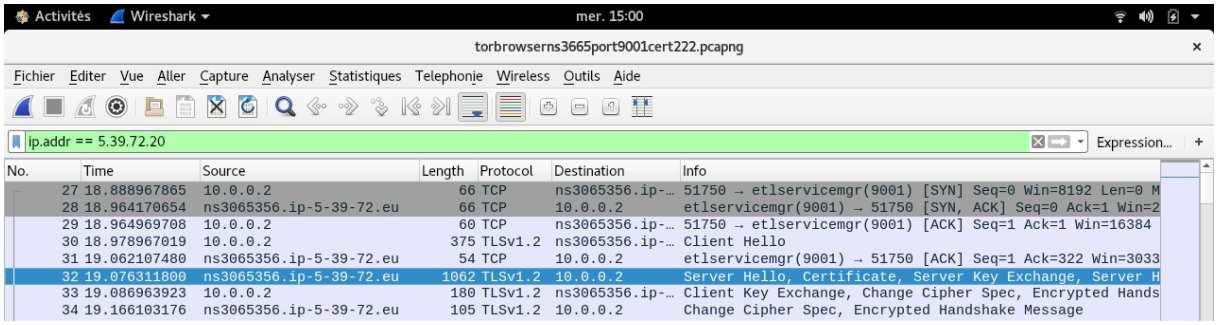


Figure 3.36. La communication entre le navigateur Tor et le premier nœud de garde 5.39.72.20.

En analysant le trafic du troisième nœud Orbot et le premier nœud Tor, on a remarqué qu'ils ont une différence dans le paquet **« Certificate »** par rapport au trafic de la connexion entre les autres navigateurs et les sites :

- La capture de la connexion entre les navigateurs Chrome et Firefox et les deux sites indique qu'ils ont un paquet **« Certificate »,** et un paquet **« Certificate status, server key exchange, server hello done »**. La capture du trafic du troisième nœud Orbot et le premier nœud Tor indique qu'ils ont le **« Certificate »** dans le même paquet du **« server hello »**, il est présenté comme suit : **« server hello, Certificate, server key exchange, server hello done ».**

Les figures ci-dessous nous présentent plus d'informations concernant les paquets **« Certificate »**.

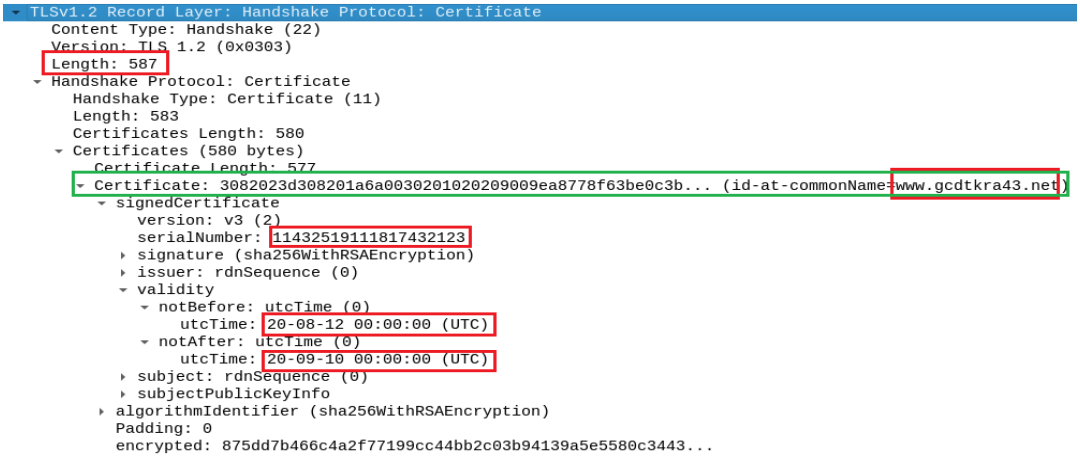


Figure 3.37. Le paquet « Certificate » du troisième nœud de garde Orbot 45.95.235.197.

Figure 3.38. Le paquet « Certificate » du premier nœud de garde Tor 5.39.72.20.

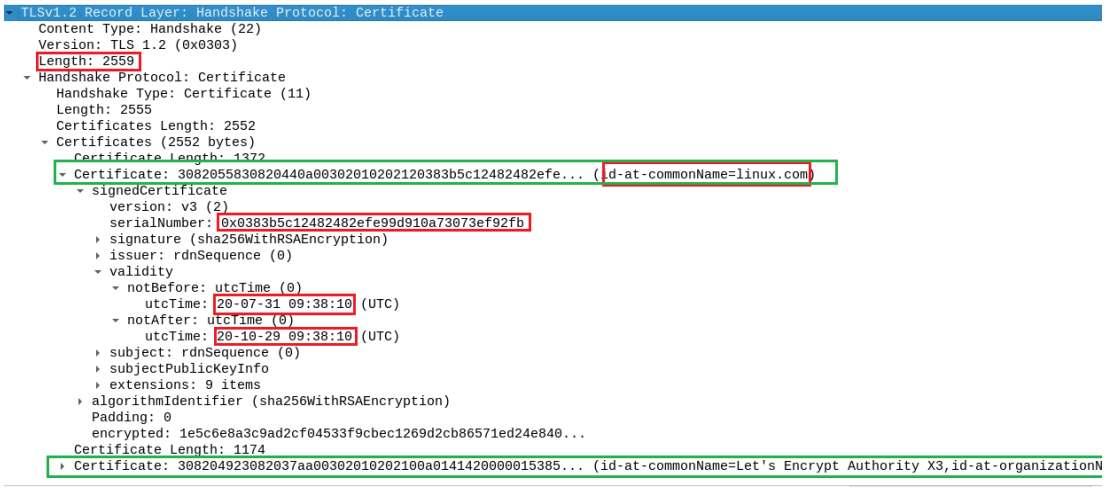


Figure 3.39. Le paquet « Certificate » de la connexion entre le navigateur Google Chrome et le premier site.

- **La longueur totale « Length » :** on remarque que la valeur du Length du nœud Orbot est égale à 587, et la valeur du Length du nœud Tor est égale à 590. Ces valeurs sont largement inférieures par rapport à celles présentent dans le paquet « Certificate » de la connexion entre les deux navigateurs et les deux sites qui varient entre 2559 et 3818.
- **Certificate :** on remarque que le paquet **« Certificate »** de la connexion entre les deux sites et les deux navigateurs contient deux certificats, le premier est celui du serveur web, le deuxième est celui de l'autorité de certification (CA). Le paquet **« Certificate »** du troisième nœud Orbot et le premier nœud Tor contient un seul certificat.

  Le nom de domaine délivré dans le certificat des nœuds de Orbot et Tor est généré d'une façon aléatoire comme indiqué sur les figures ci-dessus.
- **Nombre de série « Serial Number » :** le nombre de série des paquets **« Certificate »** de la connexion entre les deux navigateurs et les deux sites est sous forme hexadécimal. En revanche, dans les paquets de Orbot et Tor on le trouve sous forme décimal.
- **La durée de la validité : « Validity »** du certificat dans les navigateurs Chrome et Firefox est entre 3 et 10 mois. Pour Orbot cette validité a une durée de 30 jours et pour Tor cette dernière vaut 65 jours.

b. Constations et remarque

La comparaison entre les paquets TLS « protocole d'établissement de connexion SSL : SSL Handshake » de la communication entre ((Orbot, Tor) et les nœuds d'entrée), et la communication entre (les navigateurs (Chromes, Firefox) et les deux sites) nous a montré qu'il y a plusieurs différences qui peuvent distinguer l'utilisation de Orbot et Tor. On va utiliser ces différences pour extraire des empreintes numériques et les implémenter dans le système de détection d'intrusion Snort afin de détecter toute connexion à l'application Orbot et le navigateur Tor.

3.8 Extraction des empreintes numériques

En faisant la comparaison entre les différentes communications, on doit extraire les éléments spécifiques pour l'application Orbot et le navigateur Tor afin de les biens détecter. On va extraire de ces éléments un code ou bien précisément une empreinte numérique qui identifie seulement l'utilisation de Orbot et Tor.

- La première empreinte identifie les suites de chiffrement **« Cipher suites »** du paquet **« client hello ».** L'application Orbot utilise 18 suites de chiffrement, le navigateur Tor a exactement les mêmes suites de chiffrement.

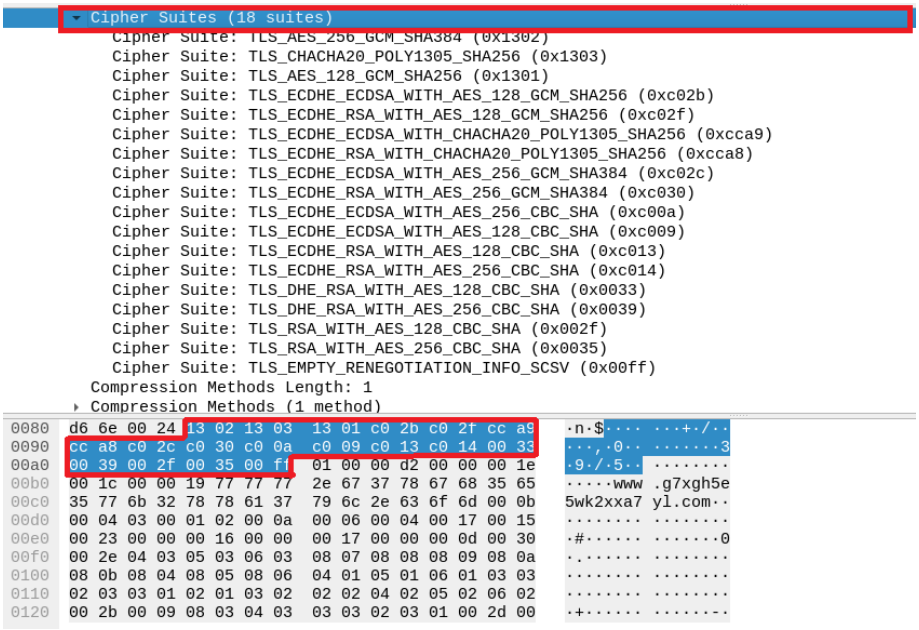


Figure3.40. L'empreinte de suite de chiffrement du paquet client hello.

Notre signature est le code encadré (code en hexadécimal)

13 02 13 03 13 01 c0 2b c0 2f cc a9 cc a8 c0 2c c0 30 c0 0a c0 09 c0 13 c0 14 00 33 00 39 00 2f 00 35 00 ff

- La deuxième empreinte identifie les algorithmes de signatures **« signature algorithms »** du paquet **« client hello ».** L'application Orbot utilise 23 signatures, le navigateur Tor a exactement les mêmes signatures.

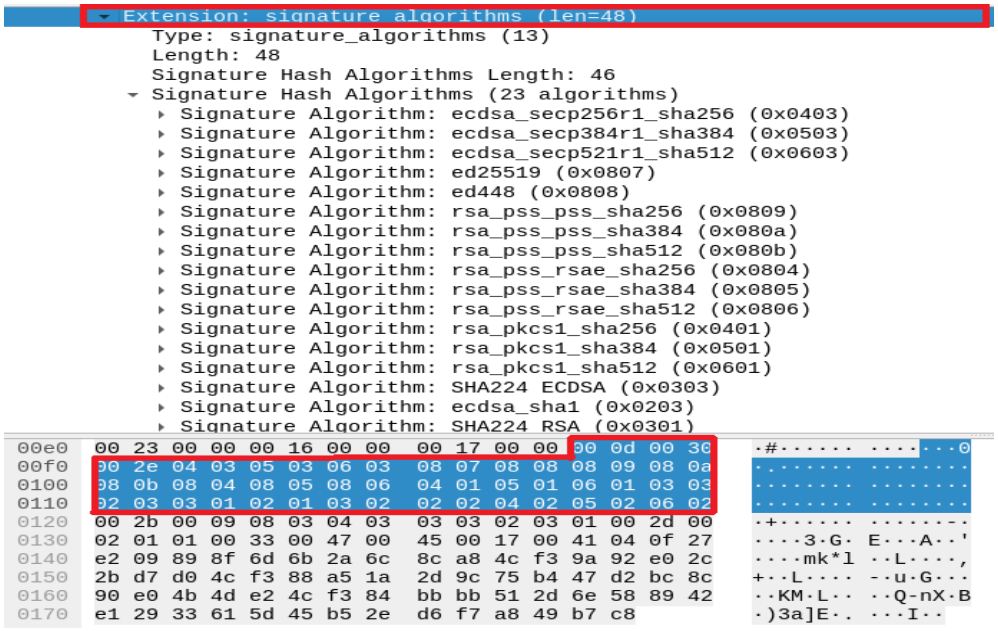


Figure3.41. L'empreinte des algorithmes de signature du paquet client hello.

00 0d 00 30 00 2e 04 03 05 03 06 03 08 07 08 08 08 09 08 0a 08 0b 08 04 08 05 08 06 04 01 05 01 06 01 03 03 02 03 03 01 02 01 03 02 02 02 04 02 05 02 06 02

- La troisième empreinte identifie les groupes supportés **« supported_groups »** du paquet **« client hello »**. L'application Orbot et le navigateur Tor utilisent 2 groupes.

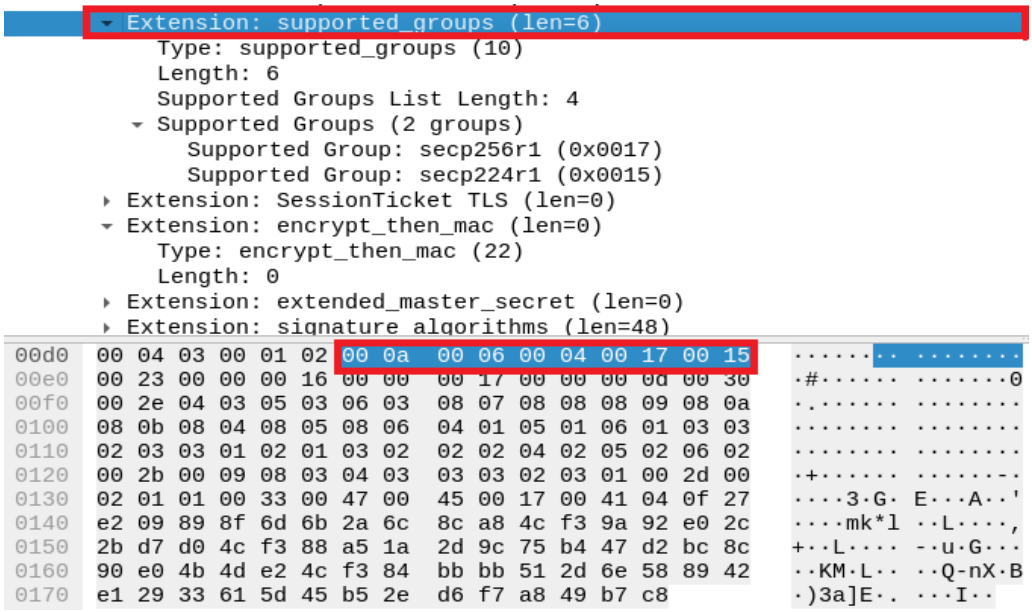


Figure3.42. L'empreinte des groupes supportés du paquet client hello.

00 0a 00 06 00 04 00 17 00 15

- La quatrième empreinte représente l'extension spécifique de l'application Orbot et le navigateur Tor **« encrypt_then_mac »**.

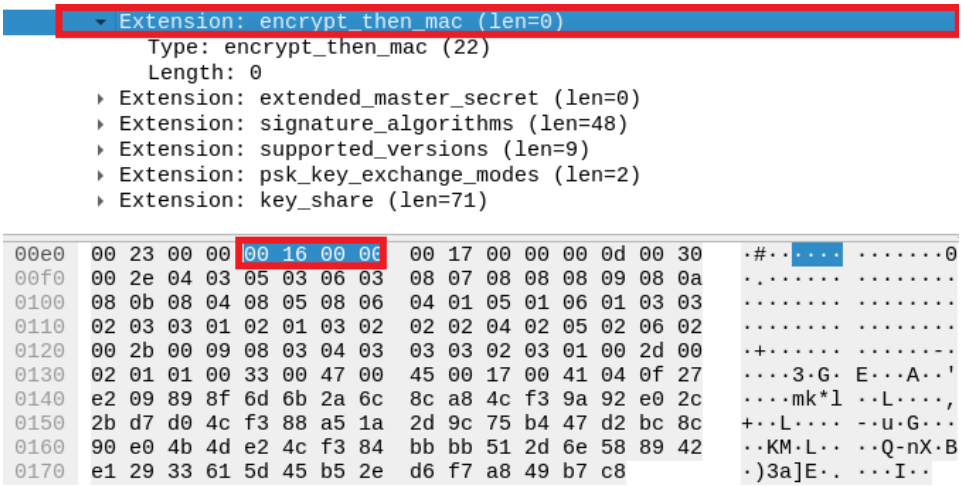


Figure3.43. L'empreinte de l'extension encrypt_then_mac du paquet client hello.

00 16 00 00

- La cinquième empreinte identifie les ports utilisés par les nœuds du réseau Tor, à savoir les ports 9001, 59999, et 433. On ne peut pas prendre le port 443 comme empreinte car il est utilisé par d'autres serveurs (port HTTPS).

Les ports 9001 et 59999

- La sixième empreinte identifie la suite de chiffrement **« Cipher suite »** utilisée dans le paquet « server hello ». On a deux suites de chiffrement :

  - La première est utilisée par les nœuds 1 et 2 de Orbot et le nœud 2 de Tor. On va prendre cette suite comme signature mais en l'associant aux ports utilisés par Orbot et Tor pour ne pas avoir une fausse alerte (détecter d'autres serveurs qui peuvent utiliser cette suite).

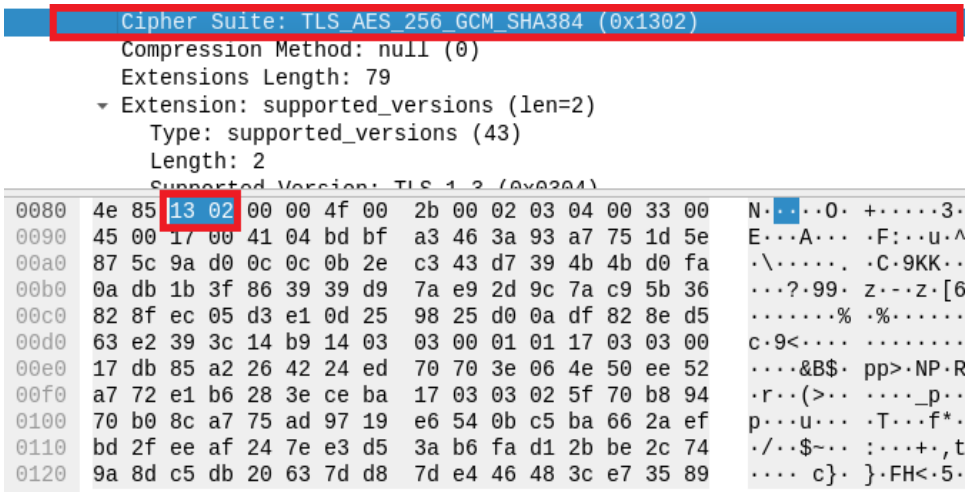


Figure3.44. L'empreinte de suite de chiffrement 1 du paquet server hello.

13 02 && les ports 9001,59999

- La deuxième est utilisée par le nœud 3 de Orbot et le premier nœud de Tor, on va prendre cette suite comme signature en procédant de la même manière comme cité précédemment.

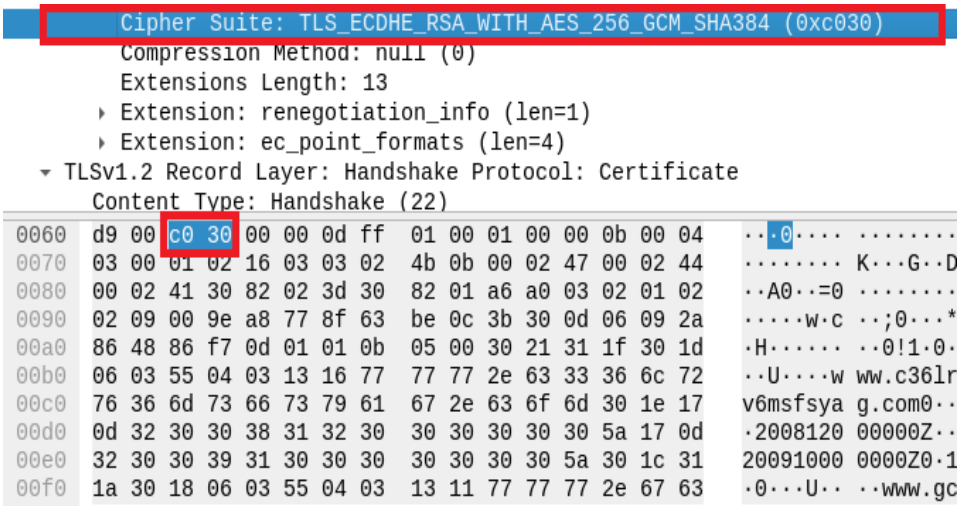


Figure3.45. L'empreinte de suite de chiffrement 2 du paquet server hello.

c0 30 && le port 9001,59999

Remarque : on a extrait ces empreintes des paquets client hello et server hello de l'application Orbot, ces empreintes sont extraites des suites de chiffrement, extensions et ports qui sont identiques pour les paquets de Orbot et le navigateur Tor.

3.9 Conclusion

Dans ce chapitre, nous avons analysé le trafic venant des applications Orbot et Tor Browser. L'étude de cette analyse nous a permis d'extraire et de retirer les signatures numériques après avoir procéder à une comparaison entre les paquets « TCP Handshake » et « TLS Handshake ». On a pu extraire plusieurs empreintes qui peuvent être implémentées sous un système de détection d'intrusion IDS.

Dans le chapitre suivant, nous allons faire l'implémentation de ces signatures dans un IDS afin de pouvoir dégager une alerte en cas de similitude.

4.1 Introduction

Dans ce chapitre nous allons faire la détection en exploitant l'analyse et l'extraction des signatures faites dans le chapitre précèdent. Tout d'abord, on va commencer par une étude théorique sur l'IDS Snort. Ensuite, on va créer des règles en utilisant les signatures extraites qui vont être utilisées comme paramètres de détection du trafic Orbot. La détection est effectuée à l'aide de l'outil Snort.

4.2 Logiciels et librairies utilisés pour la détection

Avant de commencer l'installation de Snort, vous devez avoir installé le package suivant :

- **MySQL :** La base de données MySQL.
- **MySQL–client :** La partie cliente de MySQL (connexion BD).
- **PHP–MySQL :** le module PHP de MySQL.
- **Apache :** Le serveur web Apache.
- **Mod_PHP :** Le module PHP pour Apache.
- **Libpcap/libpcap0–devel :** Librairie utilisée par Snort pour capturer les **paquets (rpm téléchargeable sur** www.rpmfind.net **).**
- **GCC :** indispensable pour compiler les sources de Snort.
- **ADOdb (Active Data Objects Data Base) :** pour BASE. ADOdb est en fait une librairie d'abstraction de base de données pour PHP. Des informations sur ADOdb peuvent être trouvées ici : http://adodb.sourceforge.net.

4.3 Le système de détection d'intrusion Snort

Snort est un IDS en réseau (NIDS) qui utilise la détection de signature, il renifle et examine les paquets de données réseau pour trouver du contenu qui correspond aux attaques connues.

Figure 4.1. Logo Snort [73].

Snort n'est pas la seule option NIDS disponible, il y a beaucoup de grands fabricants avec des produits IDS sur le marché ces jours-ci. Il existe également un certain nombre de projets IDS libres et Open Source. Notre choix s'est tombé sur Snort pour les raisons suivantes :

- **Snort est configurable** : tous les fonctionnements internes de Snort, les fichiers de configuration et les règles sont faciles à configurer, de sorte que vous pouvez accorder Snort à votre architecture réseau spécifique. Non seulement cela, mais vous pouvez créer vos propres règles pour les nouvelles attaques.
- **Snort est libre :** Snort est publié sous la GNU GPL, ce qui signifie que vous pouvez l'utiliser gratuitement. Snort est largement utilisé. Il y a des dizaines de milliers de téléchargements de Snort chaque mois sur le site : http://www.snort.org.
- **Snort fonctionne sur plusieurs plates-formes :** Snort ne fonctionne pas seulement sur Linux, mais fonctionne également sur Microsoft Windows.
- **Snort est constamment mis à jour :** les versions de maintenance de Snort sortent au besoin, généralement une fois tous les quelques mois. Les règles Snort sont régulièrement mises à jour avec de nouvelles signatures d'attaque et peuvent être téléchargées à partir de : www.snort.org [74].

4.3.1 Mode de détection

Snort permet d'analyser le trafic réseau, il peut être configuré pour fonctionner en plusieurs modes :

- **Le mode Sniffer :** dans ce mode, Snort lit les paquets circulant sur le réseau et les affiche d'une façon continue sur l'écran.
- **Le mode « Packet Logger » :** dans ce mode Snort journalise le trafic réseau dans des répertoires sur le disque.
- **Le mode détecteur d'intrusion réseau (NIDS) :** dans ce mode, Snort analyse le trafic du réseau, compare ce trafic à des règles déjà définies par l'utilisateur et établi des actions à exécuter [75].

4.3.2 Les règles Snort

Snort permet l'écriture de règles personnelles et utilise un langage simple et léger de description de règles qui est flexible et assez puissant. Les règles Snort sont divisées en deux sections logiques, l'entête de la règle et les options de la règle comme le montre la figure 4.2. L'entête de règle contient comme informations l'action de la règle, le protocole, les adresses IP source et destination et les ports source et destination. La section options de la règle contient les messages d'alerte et les informations sur les parties du paquet qui doivent être inspectées pour déterminer si l'action de la règle doit être acceptée.

Les différents champs d'une règle Snort sont :

Action	Protocole	Adresse 1	Port 1	Direction	Adresse2	Port 2	Option (msg,content)

Figure 4.2 les différents champs d'une règle Snort.

a. Les actions des règles

Il y a cinq actions accessibles par défaut dans Snort : Alert, Pass, Activate, Drop, Reject et Sdrop.

- **Alert :** génère une alerte en utilisant la méthode d'alerte sélectionnée, et alors journalise le paquet - Log : journalise le paquet.
- **Pass :** ignore le paquet.
- **Activate :** alerte et alors active une autre règle.
- **Drop :** l'action va bloquer et enregistrer les logs du paquet.
- **Reject :** l'action va bloquer et enregistrer les logs du paquet. Si c'est un paquet TCP, l'action va envoyer un TCP reset Si c'est un paquet ICMP, l'action va afficher **« Destination Port Unreachable ».**
- **Sdrop :** l'action va bloquer le paquet mais ne va pas enregistrer les logs comme pour l'action Drop.

b. Les options de règles

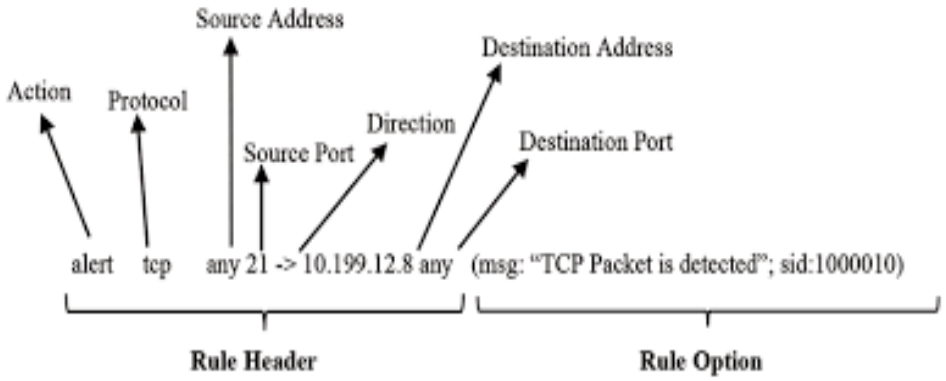


Figure 4.3. Les options de la règle Snort.

Les options des règles sont pour la plupart facultatives et pour quelques-unes obligatoires. Elles composent la signature de la règle, le contenu de ce qu'il faut analyser dans le paquet. Il est important de voir que nous pouvons indiquer nous-mêmes ce qu'il faudra observer dans le contenu d'un paquet. Dans une règle, on peut mettre plusieurs options, et celles-ci se divisent en quatre catégories :

Ces options fournissent des informations sur la règle mais n'impactent pas la détection. On y retrouve plusieurs choix :

- **Msg :** l'option va afficher le message choisi dans le Logging et l'alerte **(msg : "Alerte y'a un problème")**.
- **Sid :** l'option doit être unique et va identifier la règle Snort **(sid : "id choisi pour cette règle").**
- **Rev :** l'option est aussi unique et va identifier une révision de la règle Snort.
- **Offset :** décalage par rapport au début de la charge de données du paquet.

On peut constater, qu'il faut obligatoirement indiquer l'option **« Sid »** et l'option **« Msg »** pour toutes nos règles. Sans cela, on ne peut pas lancer Snort, car on aura une erreur dans le fichier de règles.

4.4 Création des règles Snort à partir des empreintes extraites

Snort utilise les empreintes afin de créer des règles pour détecter des trafics réseau spécifiques **(trafic malveillant « Orbot, Tor, attaques »).** La création de règles Snort dépend du fonctionnement et paramétrage de Snort.

- La première règle identifie les suites de chiffrement, cette règle va lancer une alerte avec le message **« possibilité d'utilisation de l'application Orbot : client hello Cipher suites ».**

```
alert tcp any any -> any any (msg:"possibilité d'utilisation de l'application Orbot: client hello cipher suites"; content:"|13 02 13 03 13 01 c0 2b c0 2f cc a9 cc a8 c0 2c c0 30 c0 0a c0 09 c0 13 c0 14 00 33 00 39 00 2f 00 35 00 ff|"; offset:20; sid:2000101 ; rev:1;)
```

- La deuxième règle identifie les algorithmes de signature, cette règle va lancer une alerte avec le message **« possibilité d'utilisation de l'application Orbot : client hello extension signatures algorithms ».**

```
alert tcp any any -> any any (msg:"possibilité d'utilisation de l'application Orbot:client hello extension signatures algorithms"; content:"|00 0d 00 30 00 2e 04 03 05 03 06 03 08 07 08 08 08 09 08 0a 08 0b 08 04 08 05 08 06 04 01 05 01 06 01 03 03 02 03 03 01 02 01 03 02 02 02 04 02 05 02 06 02|";  offset:20; sid:2000103 ; rev:1;)
```

- La troisième règle identifie les groupes supportés, cette règle va lancer une alerte avec le message **« possibilité d'utilisation de l'application Orbot : client hello extension supported groups ».**

```
alert tcp any any -> any any (msg:"possibilité d'utilisation de l'application Orbot: client hello extension supported groups"; content:"|00 0a 00 06 00 04 00 17 00 15|"; offset:20; sid:2000104; rev:1;)
```

- La quatrième règle identifie l'extension **« encrypt_then_mac »**, cette règle va lancer une alerte avec le message **«possibilité d'utilisation de l'application Orbot : client hello extension encryp_then_mac».**

```
alert tcp any any -> any any (msg:"possibilité d'utilisation de l'application Orbot: client hello extension encrypt_then_mac"; content:"|00 16 00 00|"; offset:20; sid:2000100 ; rev:1;)
```

- La cinquième règle identifie les ports utilisés, cette règle va lancer une alerte avec le message **« possibilité d'utilisation de l'application Orbot : client hello port de destination »**

```
alert tcp any any -> any [9001,59999] (msg :"possibilité d'utilisation de l'application Orbot : client hello port de destination « ; sid : 2000102 ; rev : 1 ;)
```

- La sixième règle identifie les suites de chiffrement du paquet server hello, Orbot et Tor utilisent deux suites de chiffrement. On a créé des règles en utilisant les empreintes de ces suites mais en ajoutant les ports utilisés (dans les deux sens : pour les paquets envoyés vers Orbot et Tor **« port destination »**, et les paquets reçus de ces derniers **« port source »**.

Les règles qui identifient la suite de chiffrement 1, ces règles vont lancer une alerte avec le message **«msg :"possibilité d'utilisation de l'application Orbot : server hello cipher suite1 »**.

```
alert tcp any any -> any [9001,59999] (msg:"possibilité d'utilisation de l'application Orbot:server hello cipher suite1 "; content:"|13 02|"; offset:20; sid:2000105; rev:1;)
```

```
alert tcp any [9001,59999] -> any any (msg:"possibilité d'utilisation de l'application Orbot:server hello cipher suite1 "; content:"|13 02|"; offset:20; sid:2000107; rev:1;)
```

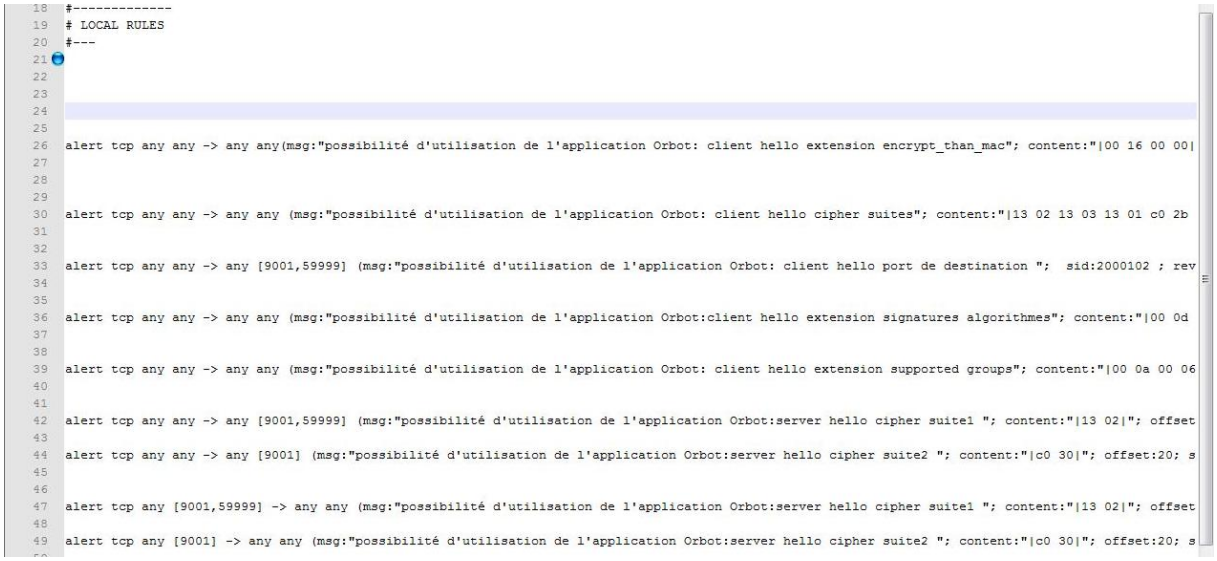
Les règles qui identifient la suite de chiffrement 2, ces règles vont lancer une alerte avec le message **«msg :"possibilité d'utilisation de l'application Orbot : server hello cipher suite2 ».**

```
alert tcp any any -> any [9001] (msg:"possibilité d'utilisation de l'application Orbot:server hello cipher suite2 "; content:"|c0 30|"; offset:20; sid:2000106; rev:1;)
```

```
alert tcp any [9001] -> any any (msg:"possibilité d'utilisation de l'application Orbot:server hello cipher suite2 "; content:"|c0 30|"; offset:20; sid:2000108; rev:1;)
```

4.5 Implémentation des règles dans Snort

Une fois les règles sont créées, on va les implémenter dans le fichier **rules** qui se trouve dans l'emplacement **C:\snort\rules.**



```
#-------------
# LOCAL RULES
#---



alert tcp any any -> any any(msg:"possibilité d'utilisation de l'application Orbot: client hello extension encrypt_than_mac"; content:"|00 16 00 00|

alert tcp any any -> any any (msg:"possibilité d'utilisation de l'application Orbot: client hello cipher suites"; content:"|13 02 13 03 13 01 c0 2b

alert tcp any any -> any [9001,59999] (msg:"possibilité d'utilisation de l'application Orbot: client hello port de destination ";  sid:2000102 ; rev

alert tcp any any -> any any (msg:"possibilité d'utilisation de l'application Orbot:client hello extension signatures algorithmes"; content:"|00 0d

alert tcp any any -> any any (msg:"possibilité d'utilisation de l'application Orbot: client hello extension supported groups"; content:"|00 0a 00 06

alert tcp any any -> any [9001,59999] (msg:"possibilité d'utilisation de l'application Orbot:server hello cipher suite1 "; content:"|13 02|"; offset

alert tcp any any -> any [9001] (msg:"possibilité d'utilisation de l'application Orbot:server hello cipher suite2 "; content:"|c0 30|"; offset:20; s

alert tcp any [9001,59999] -> any any (msg:"possibilité d'utilisation de l'application Orbot:server hello cipher suite1 "; content:"|13 02|"; offset

alert tcp any [9001] -> any any (msg:"possibilité d'utilisation de l'application Orbot:server hello cipher suite2 "; content:"|c0 30|"; offset:20; s
```

Figure 4.4. Les règles implémentées dans le fichier rules de Snort.

4.6 Détection de l'utilisation de l'application Orbot et le navigateur Tor

On va tout d'abord lancer Snort en utilisant les commandes montées dans la figure suivante

```
C:\Users\pc>cd ../

C:\Users>cd ../

C:\>cd /snort1/bin

C:\Snort1\bin>snort -i1 -c c:\snort1\etc\snort.conf -l c:\snort1\log -K ascii -i
1
```

Figure 4.5. Lancement de Snort sous CMD.

On doit aussi démarre le logiciel Wamp en cliquant sur « démarrer tous les services » pour accéder à l'interface graphique (BASE).

La figure suivante nous présente l'interface graphique (BASE) avant la détection :

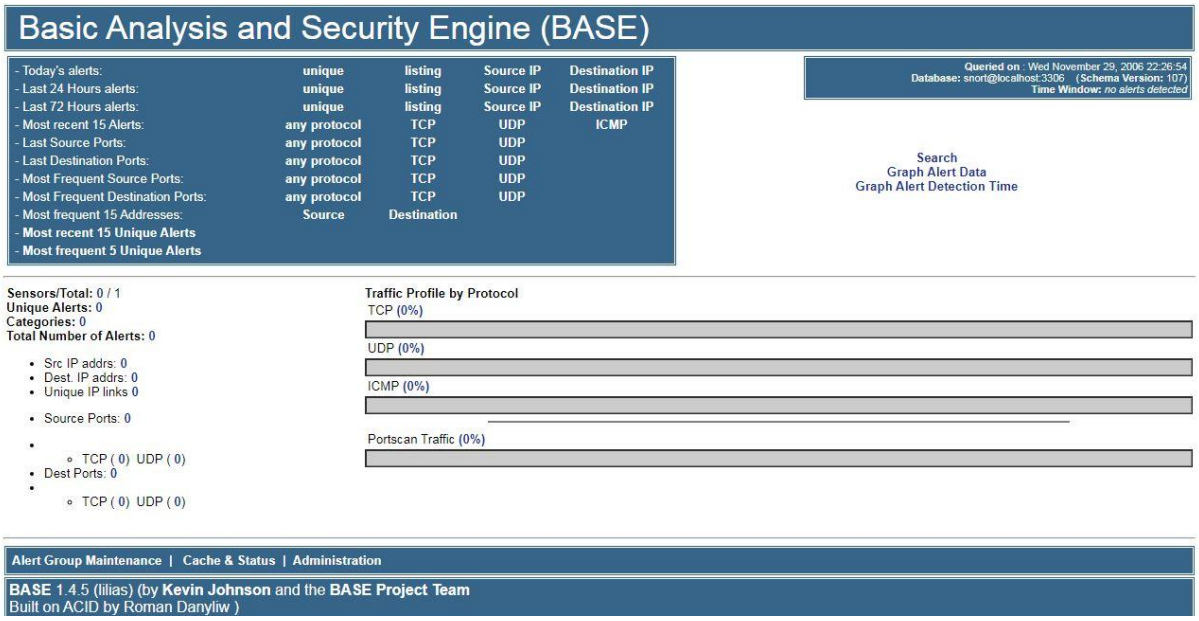


Figure 4.6. L'interface graphique (BASE).

4.6.1 Scénario de test

On a fait notre expérimentation en réalisant l'architecture montrée par la figure ci-dessous. Tout le trafic du PC client passe par le PC serveur afin de le détecter par l'IDS Snort.

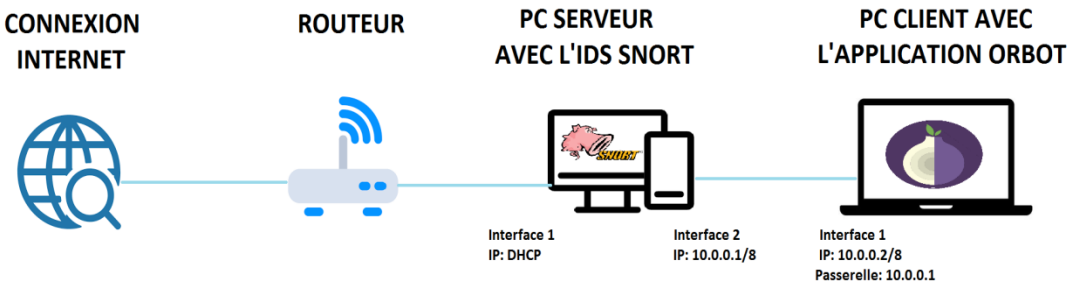


Figure 4.7. Schéma de l'architecture réalisée.

- **Comme première étape :** on doit tester la fiabilité de nos règles en faisant la détection en utilisant Snort avec un trafic du PC client venant de divers sites **(Google, YouTube, Facebook... etc.)**

Le résultat de cette détection est affiché dans la figure suivante :

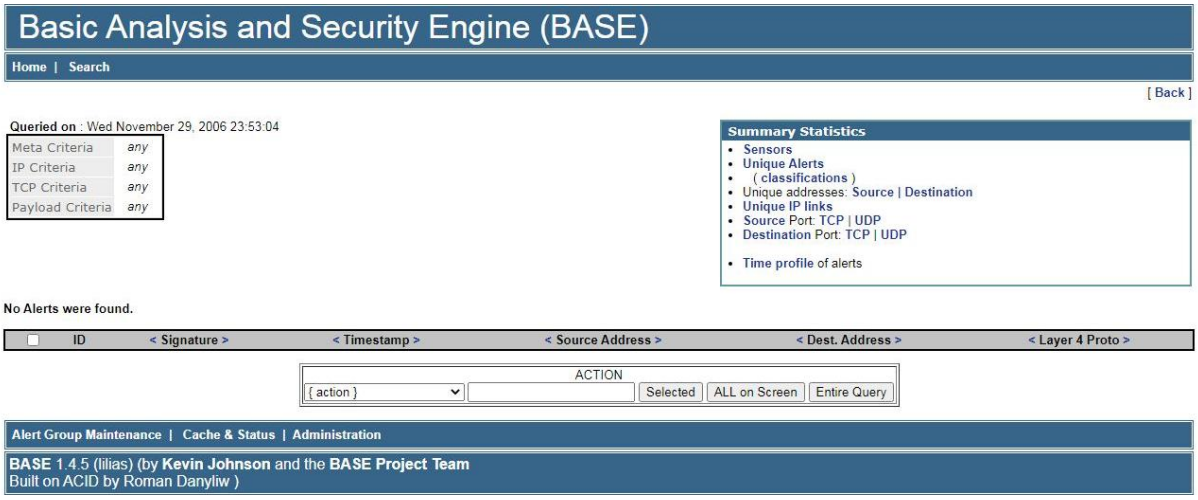


Figure 4.8. Détection du trafic venant de divers sites en utilisant les règles Orbot.

Comme nous montre la figure 4.8, on n'a détecté aucune alerte. Ceci nous confirme la fiabilité de nos empreintes. Donc si une alerte est lancée, cette dernière est bien celle de l'application Orbot.

- **La deuxième étape** est la détection du trafic Orbot. L'IDS Snort du PC serveur capte le trafic venant du pc client (Orbot) en temps réel. Pour voir les alertes remontées suite à cette tentative de connexion, on accède à l'interface graphique BASE comme montré dans la figure suivante.

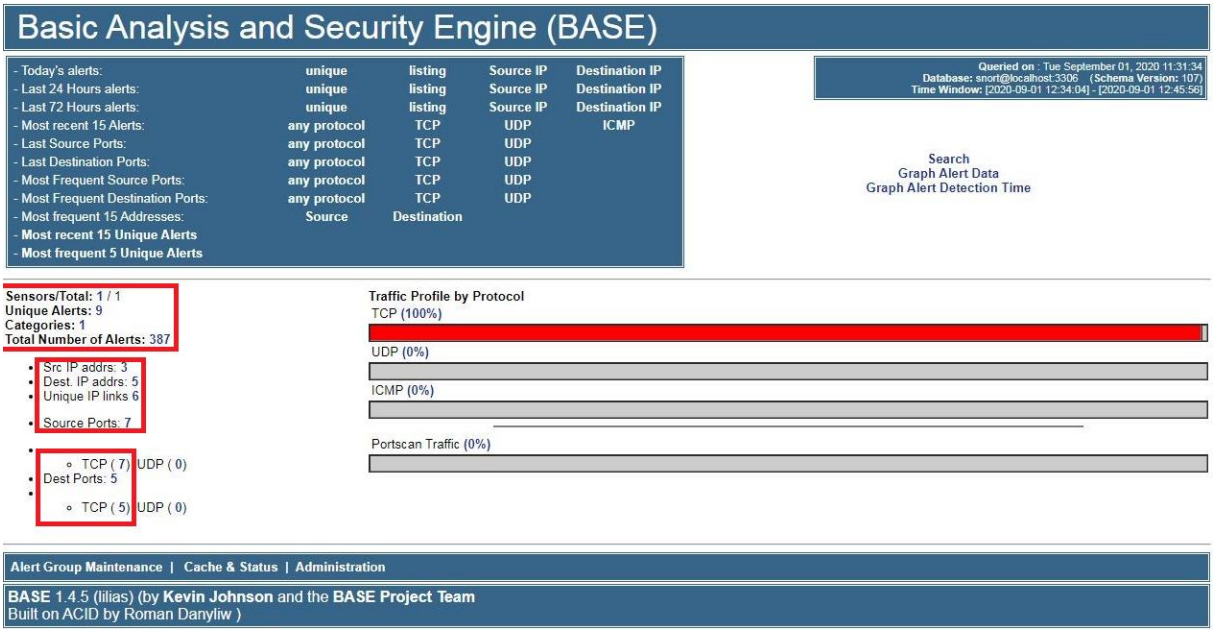


Figure 4.9. Détection du trafic de l'application Orbot.

Cette figure nous montre le lancement d'alertes par Snort. L'interface BASE nous facilite la lecture des informations concernant cette alerte : (le nombre d'alerte, les adresses IP, ports, et le protocole détecté (TCP)).

Les figures suivantes nous montrent les adresses source et destination du trafic capté :

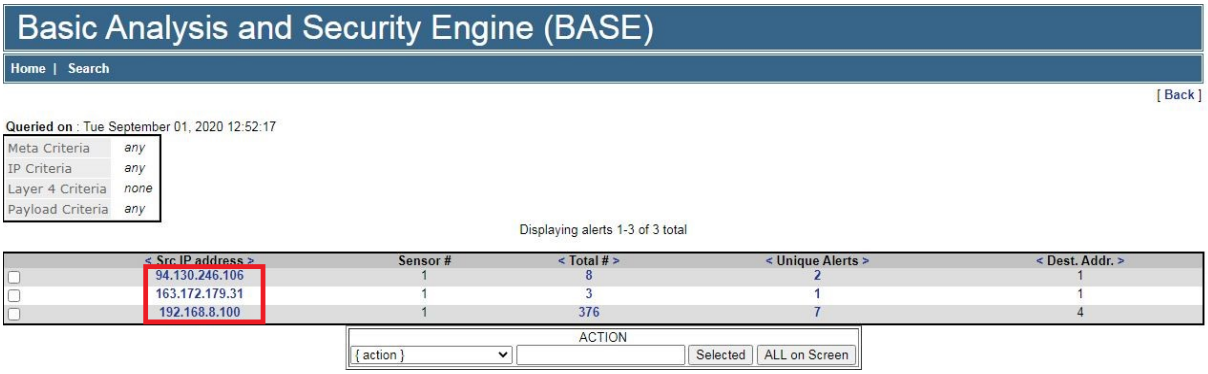


Figure 4. 10. Les adresses source du trafic capté.

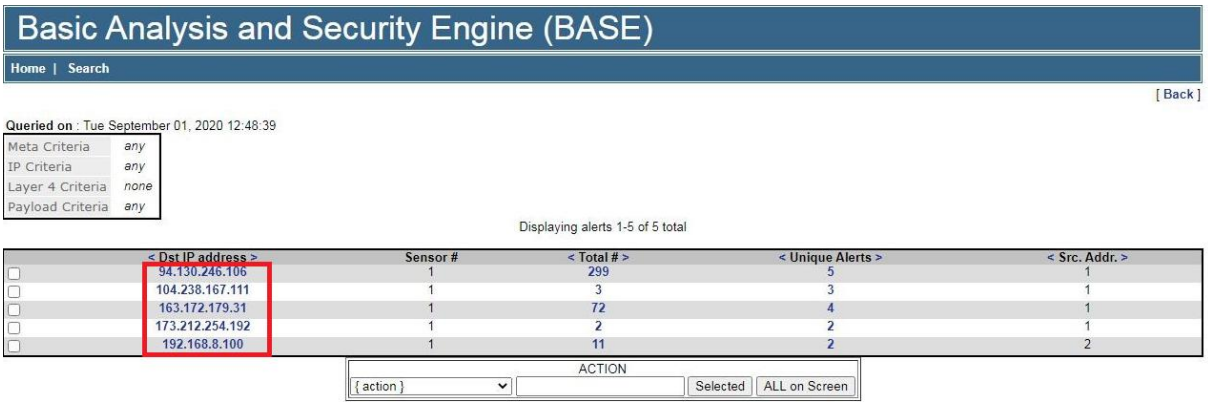


Figure 4. 11. Les adresses destination du trafic capté.

On peut vérifier l'appartenance de ces adresses au réseau Tor sur le site Tor Metrics.

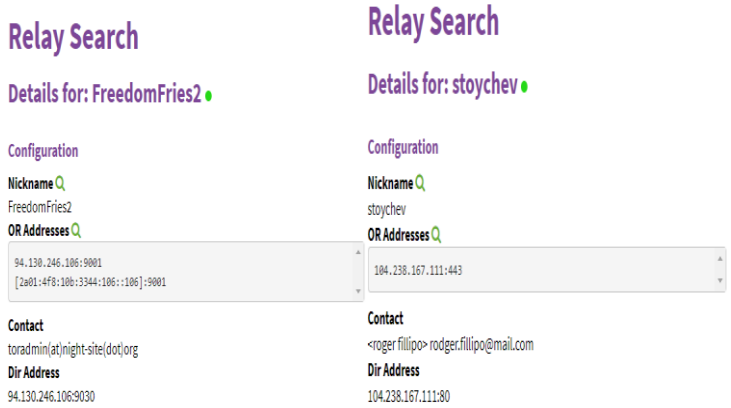


Figure 4. 12. La vérification de l'appartenance des adresses source et destination au réseau Tor.

On a trouvé toutes les adresses au niveau du site Tor Metrics, cela nous confirme la fiabilité des règles qui ont lancé ces alertes. Donc on peut garantir que ce trafic vient de l'application Orbot. Le lien **« Unique Alerts»** nous renvoie sur la liste des alertes.

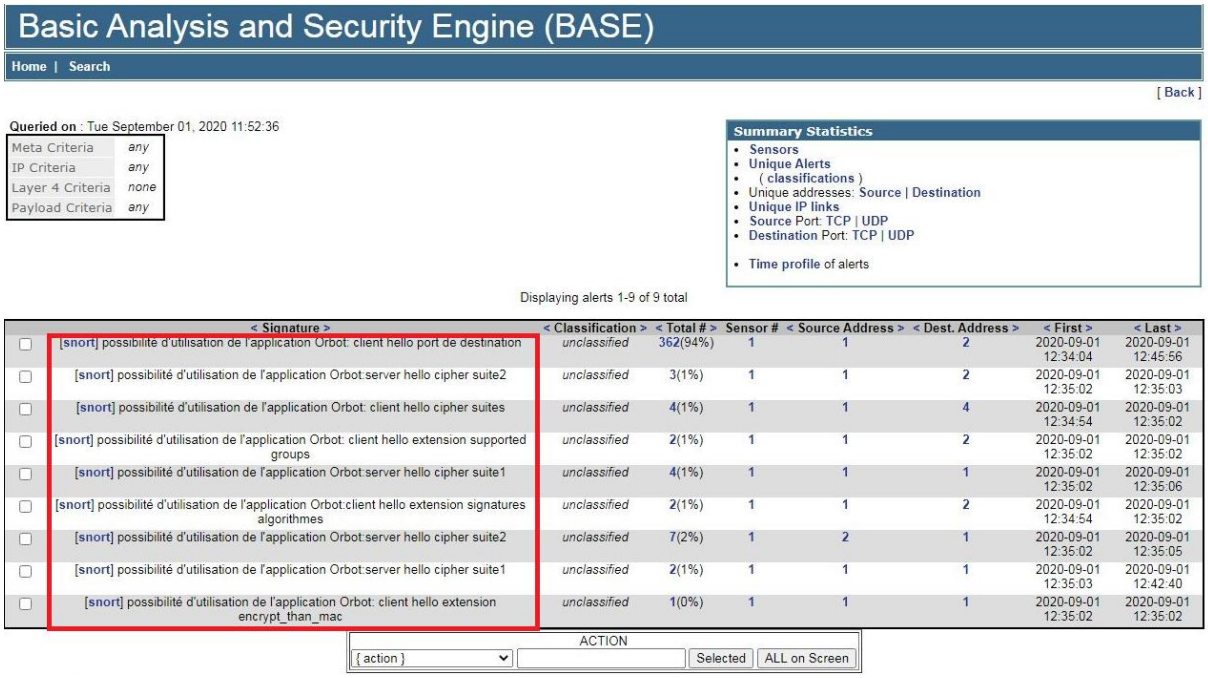


***Figure 4.13*.** Liste des alertes lancées par Snort.

On remarque que Snort a détecté l'utilisation de l'application Orbot, il a lancé neuf alertes qui correspondent à nos règles. Les alertes sont toutes lancées en même temps. Le nombre d'alerte est entre 1 et 7 pour toutes les alertes lancées sauf la première alerte **« port de destination »**, La colonne **« total »** nous renvoie sur les alertes lancées par chaque règle.

1. **Les alertes lancées par la règle (suite de chiffrement « Cipher suite »)**

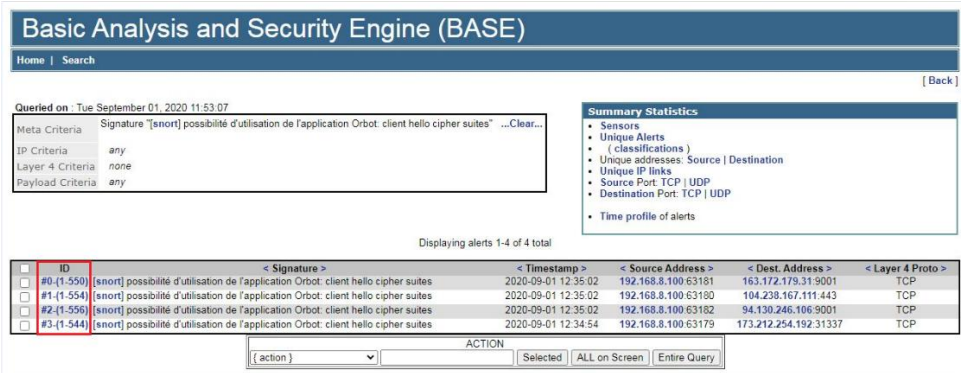


***Figure 4.14*.** Les alertes lancées par la règle « Cipher suite ».

Cette capture nous montre les adresses et ports **(source et destination),** le temps de déclenchement de l'alerte, le protocole et la signature. Les alertes sont numérotées et chaque alerte a son ID **(Identifiant)**. En cliquant sur l'Id on aura des informations supplémentaires sur le paquet.

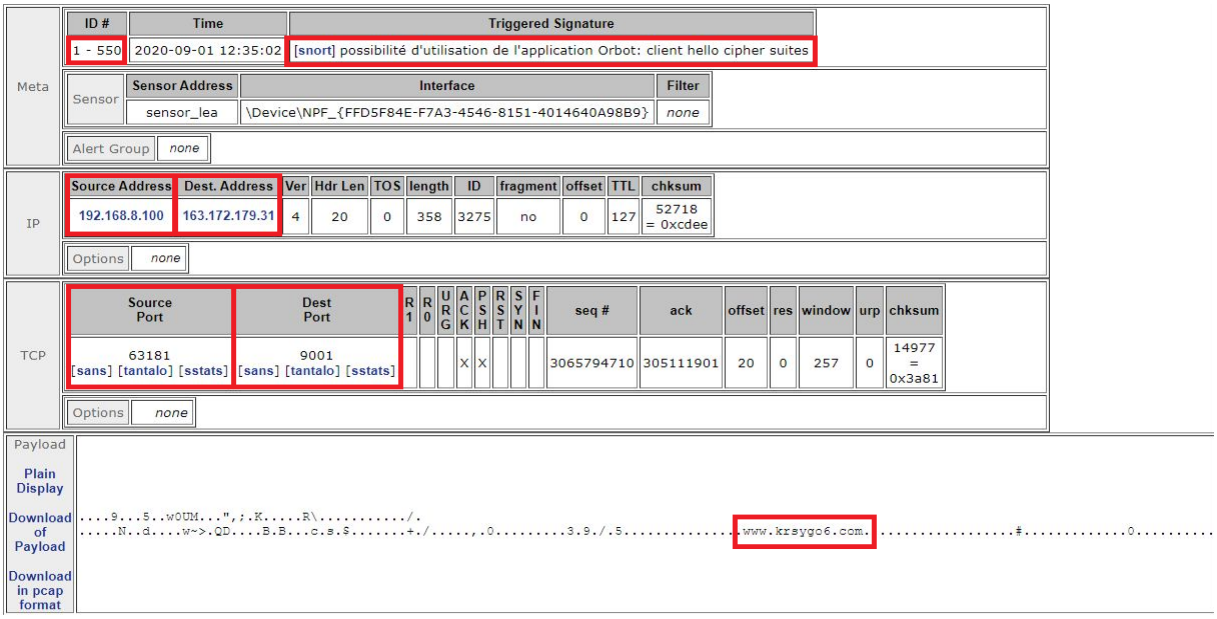


Figure 4.15. Les détails d'un paquet Cipher suites.

On a pu extraire plusieurs informations de ce paquet dont le nom de domaine, qui est généré d'une façon aléatoire ce qui confirme notre analyse faite en troisième chapitre.

2. Les alertes lancées par la règle (algorithmes de signature « signature_algorithms »)

Cette figure nous présente l'ensemble des alertes générées par la règle « signature algorithms ». Les alertes sont numérotées et contiennent quelques informations : port (source, destination), le protocole et le temps de déclenchement de chaque alerte.

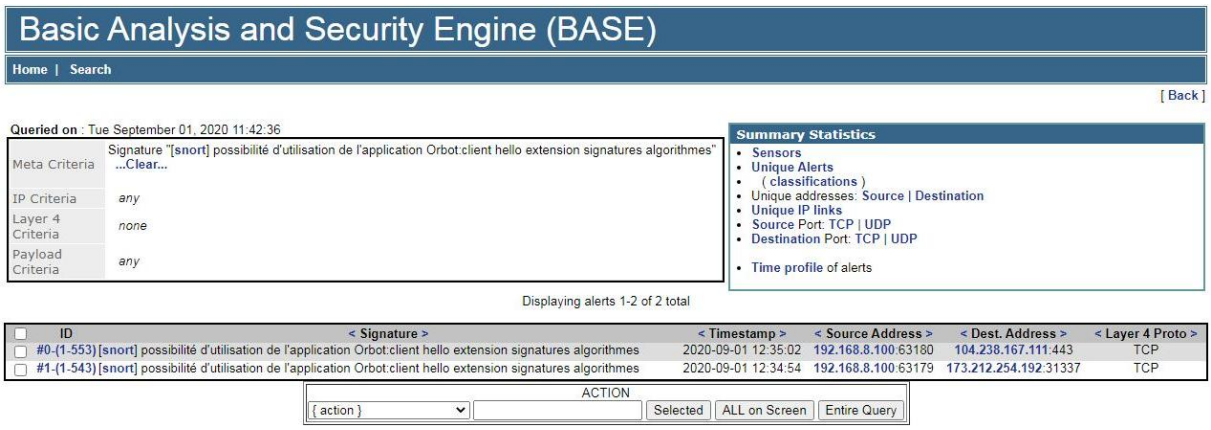


Figure 4.16. Les alertes lancées par la règle « signature_algorithms ».

3. Les alertes lancées par la règle (groupes supportés « supported_groups »)

Cette figure nous présente l'ensemble des alertes générées par la règle « supported_groups ». Les alertes sont numérotées et contiennent quelques informations : port (source, destination), le protocole et le temps de déclenchement de chaque alerte.

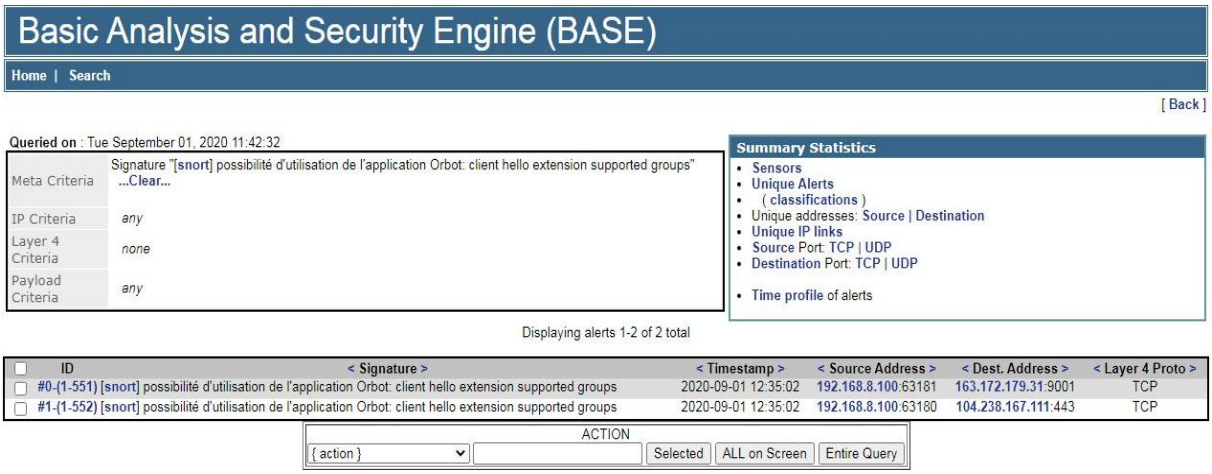


Figure 4.17. Les alertes lancées par la règle « supported_groups ».

4. Les alertes lancées par la règle « encrypt_then_mac »

Cette figure nous présente l'ensemble des alertes générées par la règle « encrypt_then_mac ». Les alertes sont numérotées et contiennent quelques informations : port (source, destination), le protocole et le temps de déclenchement de chaque alerte.

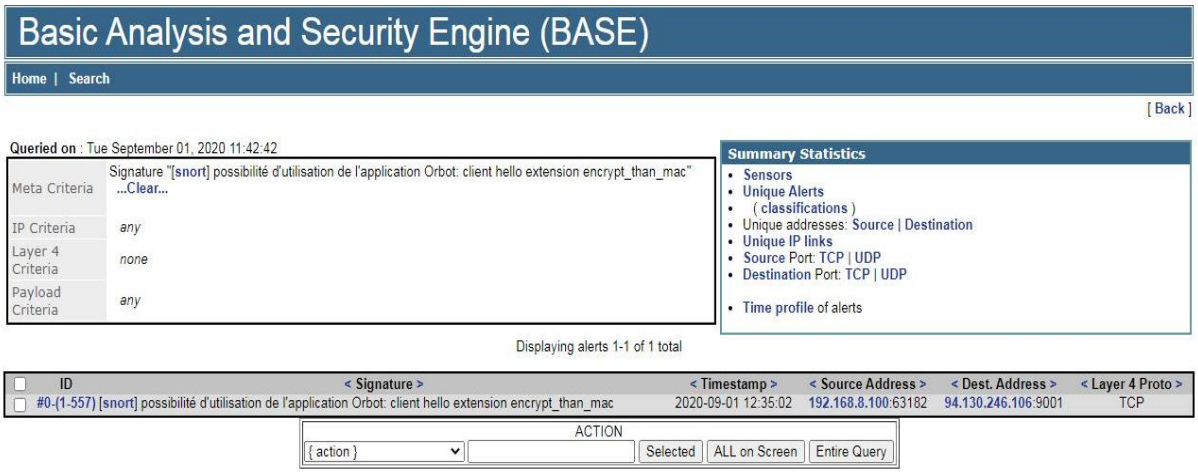
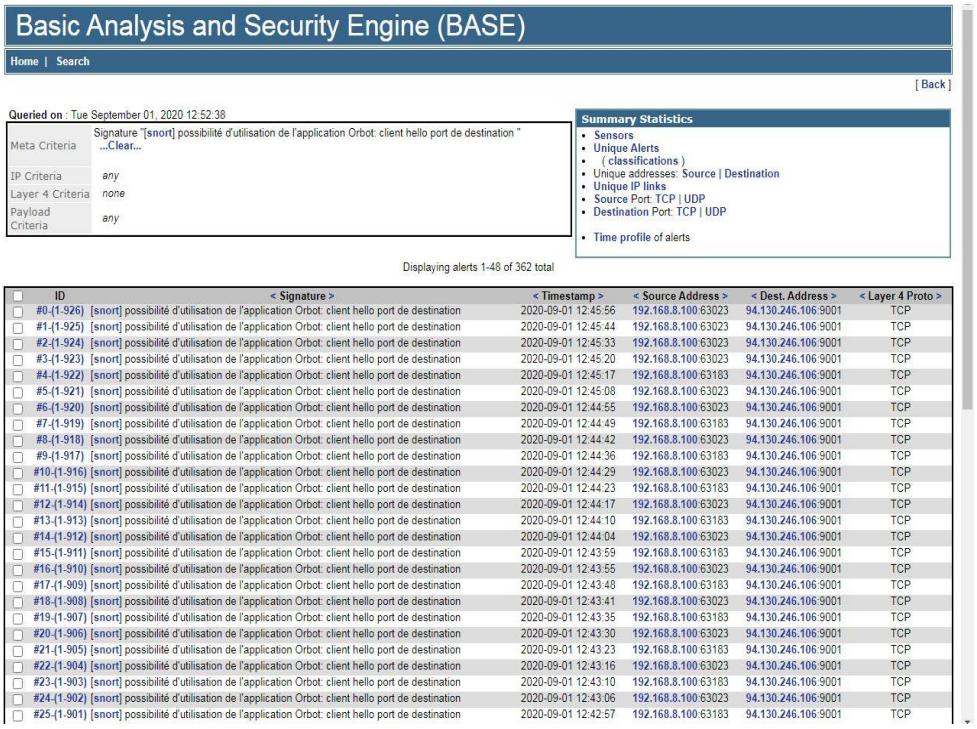


Figure 4.18. Les alertes lancées par la règle « encrypt_then_mac ».

5. **Les alertes lancées par la règle (ports de destination « destination ports »)**

Cette figure nous présente l'ensemble des alertes générées par la règle « destination ports ». Les alertes sont numérotées et contiennent quelques informations : port (source, destination), le protocole et le temps de déclenchement de chaque alerte.



ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
#0-(1-926)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:45:56	192.168.8.100:63023	94.130.246.106:9001	TCP
#1-(1-925)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:45:44	192.168.8.100:63023	94.130.246.106:9001	TCP
#2-(1-924)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:45:33	192.168.8.100:63023	94.130.246.106:9001	TCP
#3-(1-923)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:45:20	192.168.8.100:63023	94.130.246.106:9001	TCP
#4-(1-922)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:45:17	192.168.8.100:63183	94.130.246.106:9001	TCP
#5-(1-921)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:45:08	192.168.8.100:63023	94.130.246.106:9001	TCP
#6-(1-920)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:44:55	192.168.8.100:63023	94.130.246.106:9001	TCP
#7-(1-919)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:44:49	192.168.8.100:63183	94.130.246.106:9001	TCP
#8-(1-918)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:44:42	192.168.8.100:63023	94.130.246.106:9001	TCP
#9-(1-917)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:44:36	192.168.8.100:63183	94.130.246.106:9001	TCP
#10-(1-916)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:44:29	192.168.8.100:63023	94.130.246.106:9001	TCP
#11-(1-915)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:44:23	192.168.8.100:63183	94.130.246.106:9001	TCP
#12-(1-914)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:44:17	192.168.8.100:63023	94.130.246.106:9001	TCP
#13-(1-913)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:44:10	192.168.8.100:63183	94.130.246.106:9001	TCP
#14-(1-912)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:44:04	192.168.8.100:63023	94.130.246.106:9001	TCP
#15-(1-911)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:43:59	192.168.8.100:63183	94.130.246.106:9001	TCP
#16-(1-910)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:43:55	192.168.8.100:63023	94.130.246.106:9001	TCP
#17-(1-909)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:43:48	192.168.8.100:63183	94.130.246.106:9001	TCP
#18-(1-908)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:43:41	192.168.8.100:63023	94.130.246.106:9001	TCP
#19-(1-907)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:43:35	192.168.8.100:63183	94.130.246.106:9001	TCP
#20-(1-906)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:43:30	192.168.8.100:63023	94.130.246.106:9001	TCP
#21-(1-905)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:43:23	192.168.8.100:63183	94.130.246.106:9001	TCP
#22-(1-904)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:43:16	192.168.8.100:63023	94.130.246.106:9001	TCP
#23-(1-903)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:43:10	192.168.8.100:63183	94.130.246.106:9001	TCP
#24-(1-902)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:43:06	192.168.8.100:63023	94.130.246.106:9001	TCP
#25-(1-901)	[snort] possibilité d'utilisation de l'application Orbot: client hello port de destination	2020-09-01 12:42:57	192.168.8.100:63183	94.130.246.106:9001	TCP

***Figure 4.19*.** Les alertes lancées par la règle « port de destination ».

6. **Les deux alertes lancées par les règles (suite de chiffrement 1 avec les ports source et destination spécifiés « Cipher suites 1 »)**

Cette figure nous présente l'ensemble des alertes générées par la règle « Cipher suite1 ». Les alertes sont numérotées et contiennent quelques informations : port (source, destination), le protocole et le temps de déclenchement de chaque alerte.

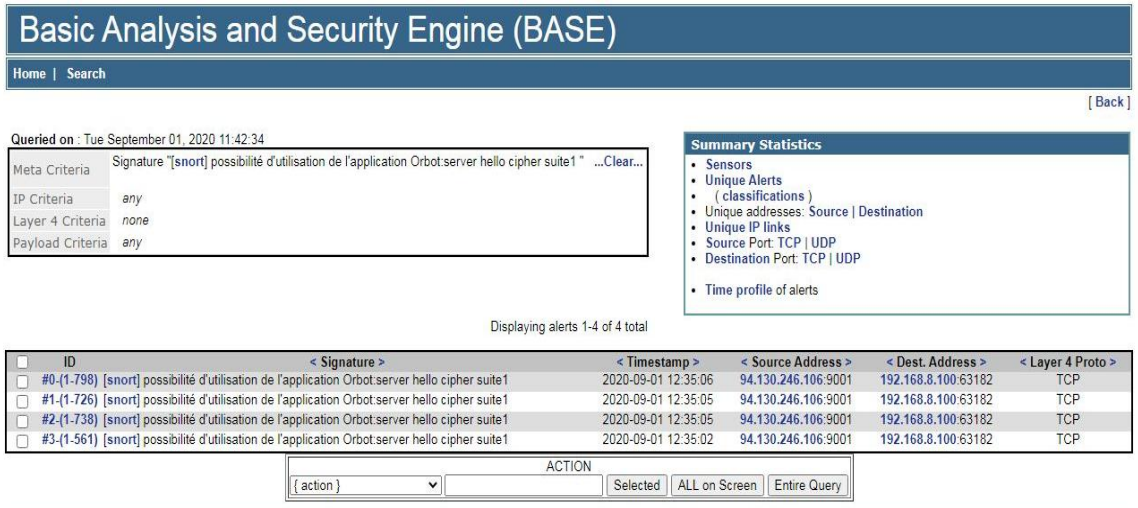


Figure 4.20. Les alertes lancées par la règle « Cipher suite 1 ».

7. **Les deux alertes lancées par les règles (suite de chiffrement 2 avec les ports source et destination spécifiés « Cipher suites 2 »)**

Cette figure nous présente l'ensemble des alertes générées par la règle « Cipher suite2 ». Les alertes sont numérotées et contiennent quelques informations : port (source, destination), le protocole et le temps de déclenchement de chaque alerte.

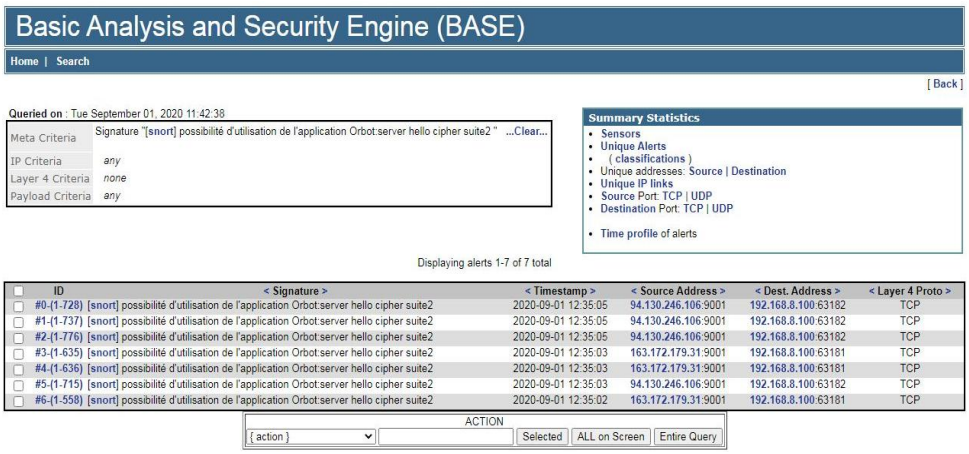


Figure 4.21. Les alertes lancées par la règle « Cipher suite 2 ».

Remarque : on a fait la détection du trafic du navigateur Tor avec les mêmes règles (puisque on a trouvé les mêmes signatures), et on a eu le même résultat au niveau de la détection de Orbot (lancement de toutes les alertes).

4.7 Constations

- On a détecté l'utilisation de Orbot et du navigateur Tor en utilisant les mêmes règles parce que ces derniers passent par le même réseau (le réseau Tor).
- Toutes les règles implémentées dans Snort ont impliqué des alertes, ces règles sont fiables et n'engendrent pas de fausses alertes.
- Le nombre d'alertes générées par la règle du port de destination est très élevé puisque chaque requête ou tentative de connexion au réseau Tor passe par ces ports.
- Les autres règles génèrent des alertes lors de l'établissement de la connexion SSL « SSL Handshake ».

4.8 Conclusion

L'application Orbot est souvent utilisée à mauvais escient par des pirates et des criminels afin de contrôler à distance des ordinateurs piratés. Dans ce chapitre, nous avons exploité toute l'étude faite dans les chapitres précédents afin de connaitre le fonctionnement du l'application Orbot, puis on a utilisé les empreintes extraites pour créer des règles, les implémenter dans l'IDS Snort et détecter l'utilisation de cette application dans un réseau d'entreprise.

Les tests faits en utilisant ces règles nous ont confirmé leur fiabilité. Donc on peut prouver que notre méthodologie peut détecter avec succès les connexions de l'application Orbot.

Conclusion générale

Détecter le trafic Orbot aidera à identifier les infections possibles sur votre réseau. Il garde les yeux de votre équipe réseau ouverts à différents types de logiciels malveillants qui utilisent Tor pour déguiser son intention. Tor non seulement crypte son trafic, mais déguise également son trafic en communications HTTPS, ce qui fait d'un IDS un atout précieux pour mettre ce trafic à part du trafic HTTPS normal que nous savons tous qu'il est sur le réseau.

Ce mémoire présente une étude expérimentale utilisant Snort pour détecter le trafic Orbot dans l'ensemble de données de trafic réseau. La recherche se concentre principalement sur la comparaison entre le trafic Web normal et trafic venant de l'application Orbot qui abaisse la sécurité du réseau. Pour cela on a proposé une liste de règles pour la détection du trafic Orbot.

Notre architecture de détection est parvenue à identifier des sources de trafic venant de l'application Orbot Avec un taux élevé. Lors des tests faits, on a atteint un taux de détection qui vaut environ 100%.

Afin d'améliorer la détection du réseau Orbot dans une entreprise nous recommandons :

• Implémenter un NIDS en créant des règles qui permettent d'identifier le trafic venant de l'application Orbot (cette règle doit contenir une signature qui peut identifier cette application comme : les ports, les adresses IP, les extensions… etc.).

•Il nécessaire de faire des mises à jour fréquentes. L'efficacité de ce système de détection dépend fortement de la précision de sa base de signatures.

• Mettre une charte de bonne conduite au sein de l'entreprise : interdire ou mettre à la quarantaine toute utilisation de cette application.

• Sensibilisation et formation : on doit être toujours vigilants contre les menaces venant de l'application Orbot vu les risques liés à son utilisation.

•Mise en place d'un serveur d'annuaire interne à l'entreprise pour enregistrer les profils des employés

[1] SADAOUI Idir : 'Les attaques par déni de service distribué dans les systèmes informatiques', mémoire de master, Université Abderrahmane Mira de Bejaia 2016.

[2] 'Les 5 cyberattaques les plus spectaculaires' récupéré en Juillet 2020 sur : www.avanista.fr.

[3] Lilian Bossuet : 'Sécurité des systèmes embarqués', 'Tech. Ing. – Techniques de l'Ingénieur', Juin 2018.

[4] Mickael Choisnard : 'Réseaux et sécurité informatiques', Cours MIGS 2 Novembre 2015.

[5] ACISSI : 'Sécurité informatique-Ethical Hacking : Apprendre l'attaque pour mieux se défendre', éditions ENI, 2009.

[6] 'Qu'est-ce qu'un renifleur et comment s'en défendre' récupéré en Mars 2020 sur : www.avast.com.

[7] 'Qu'est-ce qu'une attaque de l'Homme du milieu ?' récupéré en Août 2020 sur : www.cloudflare.com.

[8] Jean-Marie Flaus : 'Cybersécurité des systèmes industriels', éditions ISTE, 2019.

[9] 'Les attaques par déni de service' récupéré en Mars 2020 sur : http://www.neurones-it.com.

[10] Karmadenur ET Raka Yusuf: 'Analysis of Snort Rules to Prevent Synflood Attacks on Network Security', International Journal of Computer Applications, 178, 14-19, Août 2019.

[11] 'Attaques - Attaque par fragmentation' récupéré en Mars 2020 sur : https://web.maths.unsw.edu.au/~lafaye/CCM/attaques/attaque-teardrop.htm.

[12] 'OS-Specific DOS Attack-TechLibrary' récupéré en Mars 2020 sur : www.juniper.net.

[13] Harshita: 'Detection and Prevention of ICMP Flood DDOS Attack', 63-69, 2017.

[14] 'DDOS différents types de méthode d'écriture de code de commande d'attaque' récupéré en Mars 2020 sur : www.xinruiyun.cn.

[15] 'Attaque par réflexion (Smurf)', récupéré en Mars 2020 sur : www.commentcamarche.net.

[16] 'Attaque DOS (Deny Of Service)' récupéré en Mars 2020 sur : www.frameip.com.

[17] Eric Filiol : 'Les virus informatiques : théorie, pratique et applications', Springer, Paris, 2009.

[18] Jan Camensich ET Dogan Kesdogan: 'INetSec 2009 - Open Research Problems in Network Security', ADS, November 2009.

[19] Natarajan Kanchana ET Sarala Subramani : 'Generation of SQL-injection Free Secure Algorithm to Detect and Prevent SQL-Injection Attacks', Procedia Technology, 4, 790-796, 2012.

[20] 'What is SQL injection? Tips to prevent SQL injections in 2020' récupéré en Août 2020 sur: https://www.dnsstuff.com.

[21] Avancini Andrea ET Ceccato Mariano : 'Comparison and integration of genetic algorithms and dynamic symbolic execution for security testing of cross-site scripting vulnerabilities', Information and Software Technology, 55, 2209–2222, 2013.

[22] 'Qu'est-ce que le Cross-Site Scripting ?' récupéré en Mars 2020 sur : www.cloudflare.com.

[23] Mark Ciampa: 'Security Awareness: Applying Practical Security in Your World', CENGAGE Learning Custom Publishing, Janvier 2016.

[24] Phung Khac : 'La sécurité dans les réseaux hauts débit', Rapport de tipe, Institut de la francophonie pour l'informatique (IFI), Mai 2005.

[25] Jean-François Pillou ET Jean-Philippe Bay : 'Tout sur la sécurité informatique', Dunod, 2016.

[26] Laurent Bloch ET Christophe Wolfhugel : 'Sécurité informatique : Principes et méthodes à l'usage des DSI, RSSI et administrateurs', Eyrolles, 2009.

[27] Alex X. Liu: 'Firewall Design and Analysis', World Scientific Publishing Co Pte Ltd, 2010.

[28] 'Firewall (Pare-feu)' récupéré en Juillet 2020 sur : www.syloe.com.

[29] Paul RASCAGNERES : ' Sécurité informatique et Malwares
Analyse et contre-mesures (2e édition)', Editions ENI, 2016.

[30] Dr. Manish Saxena Next: 'Generation Intelligent Network Intrusion Prevention System', Lulu.com, 2017.

[31] Illir KADRIU : 'Utilisation de Snort dans une PME', Haute école de gestion Genève, édition, 2019.

[32] James C.Foster, Jeffrey POSLUNS ET Brian CASWELL : 'Snort Intrusion Detection', Editions Syngress Publishing USA, 2003.

[33] Michael AMAND ET Mohamed NSIRI : 'Etude d'un système de détection d'intrusion comportemental pour l'analyse du trafic aéroport mental ', rapport de projet tutoré, 2011.

[34] Artrit AJDINI : 'Étude et conception d'un service assurant l'anonymat', Travail de Bachelor réalisé en vue de l'obtention du Bachelor HES, Haute École de Gestion de Genève (HEG-GE), 2019.

[35] 'Ce qu'il faut savoir sur la collecte de nos données personnelles' récupéré en Juin 2020 sur : https://mbamci.com.

[36] 'Surveillance d'Internet : votre vie privée en danger' récupéré en Juillet 2020 sur :

https://www.contrepoints.org.

[37] 'Deep Packet Inspection (DPI)' récupéré en Juin 2020 sur : https://assiste.com/Deep_Packet_Inspection_DPI.html.

[38] Guillaume Pillot : 'Anonymat et vie privé sur internet', mémoire, Université Laval Québec Canada, 2018.

[39] 'Les différentes attaques web : les différentes vulnérabilités ' récupéré en Juin 2020 sur : https://www.supinfo.com.

[40] 'Plugins : présentation 'récupéré en Juillets 2020 sur : https://assistance.orange.fr.

[41] 'Bandeau cookie | Comment se conformer au RGPD/ePR' récupéré en Juillet 2020 sur : https://www.cookiebot.com/fr/bandeau-cookie/.

[42] 'Cookies Internet | Que sont-ils et que font-ils ?' récupéré en Juillet 2020 sur : https://www.cookiebot.com.

[43] Yi Shi ET Kanta Matsura: 'Fingerprinting Attack on the Tor Anonymity System', rapport, Université de Tokyo, Japan, 2017.

[44] ‘Envoyer un email vraiment anonyme, c'est en fait plutôt facile’ récupéré en Juin 2020 sur : www.clubic.com.

[45] ‘Adresses email jetables’ récupéré en Juin 2020 sur : www.yubigeek.com.

[46] ‘Les 5 meilleurs moteurs de recherche pour protéger votre vie privée’ récupéré en Juin 2020 sur : https://syskb.com.

[47] ‘DuckDuckGo : le moteur de recherche qui protège vos données’ récupéré en Juillet 2020 sur : https://www.lebigdata.fr.

[48] ‘Ghostery – Bloqueur de publicité protégeant la vie privé’ récupéré en Juin 2020 sur : https://chrome.google.com.

[49] ‘HTTPS Everywhere : protéger sa vie privée sur Firefox et Chrome’ récupéré en Juin 2020 sur : www.futura-sciences.com.

[50] Sudhanshu Chauhan ET Nutan KumarPanda : ‘Hacking Web Intelligence : Open Source Intelligence and Web Reconnaissance Concepts and Techniques’, Syngress, 2015.

[51] ‘VPN : définition, utilisations et sélection des meilleurs VPN gratuits et payants’ récupéré en Juin 2020 sur www.frandroid.com.

[52] Kulbir Saini: ‘Squid Proxy Server 3.1: Beginner's Guide Improve the performance of your network using the caching and access control capabilities of Squid’, Packet Publishing, 2011.

[53] ‘Comment devenir (réellement) anonyme sur Internet ?’, récupéré en Mars 2020 sur : www.institut-pandore.com.

[54] Dawson Maurice ET Cardenas-Haro Jose: 'Tails Linux Operating System: Remaining Anonymous with the Assistance of an Incognito System in Times of High Surveillance’, International Journal of Hyper connectivity and the Internet of Things, 1, 47-55, 2017.

[55] ‘ IBM X-Force : utilisation de Tor et des ransomware en augmentation’ récupéré en Août 2020 sur : https://www.itpro.fr.

[56] ‘Expertises VIGIWORLD’ récupéré en Août 2020 sur : http://www.vigiworld.com.

[57] 'Quelle est la différence entre Deep Web et Dark Web ?' récupéré en Août 2020 sur : https://darknet-tor.com.

[58] Steven Gates : 'Réseau Anonyme Tor 101 : Une Introduction à la Partie la Plus Privée de l'internet', Books on Demand, 2018.

[59] 'TOR : tout savoir sur le navigateur web qui protège vos données' récupéré en Juin 2020 sur : www.lebigdata.fr.

[60] Laurent Gayard : 'Géopolitique du Darknet : Nouvelles frontières et nouveaux usages du numérique', Iste éditions, 2018.

[61] 'Comment assurer son anonymat ?' récupéré en Juillet 2020 sur : http://www-igm.univ-mlv.fr.

[62] Ciprian Dobre Fatos Xhafa: 'Pervasive Computing Next Generation Platforms for Intelligent Data Collection', Academic Press, 2016.

[63] 'Tor : Overview' récupéré en Juin 2020 sur : https://2019.www.torproject.org.

[64] 'Tor' récupéré en Juillet 2020 sur : https://graal.ens-lyon.fr.

[65] 'Mobile Vs, Desktop Internet Usage (Latest 2020 Data) récupéré en Septembre 2020 sur : https://www.broadbandsearch.net.

[66] 'Orbot proxy par Tor' récupéré en Juin 2020 sur : https://play.google.com.

[67] 'Tous ce qu'il faut savoir sur le navigateur Tor ' récupéré en Juillet 2020 sur :

https://www.lemondeinformatique.fr.

[68] 'Qu'est-ce que CentOS ?' récupéré en Août 2020 sur : https://www.ionos.fr.

[69] 'BLUESTACKS' récupéré en Août 2020 sur : https://www.lesnumeriques.com.

[70] 'About Wireshark' récupéré en Juillet 2020 sur : https://www.wireshark.org.

[71] 'Que se passe-t-il lors d'un Handshake TLS ? | Prise de contact SSL' récupéré en Août 2020 sur : https://www.cloudflare.com.

[72] ‘Comprendre le SST/TLS’ récupéré en Août 2020 sur : https://medium.com.

[73] ‘Logo Snort ‘ récupéré en Août 2020 sur : https://www.snort.org.

[74] Charlie SCOLT, Pau WOLFE ET Bert HAYES:’ Snort For Dummies ‘, Edition Wiley Publishing, INC Canada, 2004.

[75] Club de la sécurité informatique :’Snort outil de détection d’intrusions ‘, 2019.