S. Lavirotte, D. Lopez
Polytech'Nice – Sophia
SI3
2014-2015

TD séance n° 14
Réseau, Internet et Sécurité

1 Introduction / Rappels

1.1 Modèle OSI

Nous avons étudié le modèle OSI, modèle en couches, qui permet d'identifier les différentes fonctionnalités de traitement en fonction des couches.

Modèle OSI

	Type de Donnée	Couche	Fonction
Couches Hautes	Donnée	7. Application	Point d'accès aux services réseaux
		6. Présentation	Gère le chiffrement et le déchiffrement des données, convertit les données machine en données exploitable par n'importe quelle autre machine
		5. Session	Communication Interhost, gère les sessions entre les différentes applications
	Segments	4. Transport	Connexions bout à bout, connectabilité et contrôle de flux
Couches Matérielles	Paquet/Datagramme	3. Réseau	Détermine le parcours des données et l'adressage logique
	Trame	2. Liaison	Adressage physique
	Bit	1. Physique	Transmission des signaux sous forme binaire

1.2 Modèle TCP/IP

Nous avons vu dans les semaines précédentes que sur Internet, les machines utilisent TCP/IP pour communiquer. Le modèle TCP/IP (appelé aussi modèle Internet), qui date de 1976, a été stabilisé bien avant la publication du modèle OSI en 1984. Il présente aussi une approche modulaire (utilisation de couches) mais en contient uniquement quatre :

- Couche Physique
- Couche Réseau
- Couche Transport
- Couche Services

Aujourd'hui, c'est le modèle TCP/IP, plus souple, qui l'emporte sur le marché. Le modèle OSI, plus rigoureux, est principalement utilisé pour certaines applications critiques.

1.3 Un complément aux informations vues précédemment

Si vous avez pris un peu de recul par rapport aux connaissances que vous avez acquises lors des 2 précédents TD sur le réseau (et donc réfléchi à tout cela), vous devez vous rendre compte qu'il y a un manque dans les explications qui vous ont été données.

Comment expliquer que plusieurs applications peuvent communiquer simultanément via Internet sur votre ordinateur (votre lecteur de mail, votre navigateur web, votre logiciel vous permettant d'écouter de la musique sur Internet, ...) ? En effet, votre machine possède une adresse IP qui lui permet d'être identifiée sur Internet et donc d'envoyer des messages et de recevoir des réponses à ceux-ci. Comment l'ordinateur va savoir, à la réception d'un message par exemple, vers quelle application envoyer les données reçues ?

TD séance n° 14
Réseau, Internet et Sécurité

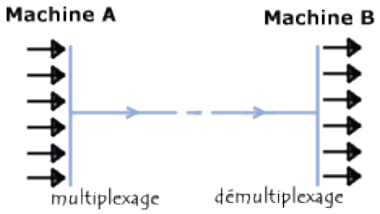
2 Les ports[1]

Ainsi, pour répondre à ce problème, chacune des applications s'exécutant sur votre machine et devant communiquer se voit attribuer une adresse unique sur la machine, codée sur 16 bits: un **port**. La combinaison adresse IP + port est alors une adresse unique au monde : elle est appelée **socket**.

L'adresse IP sert donc d'identifier de façon unique un ordinateur sur le réseau tandis que le numéro de port indique l'application à laquelle les données sont destinées. De cette manière, lorsque l'ordinateur reçoit des informations destinées à un port, les données sont envoyées vers l'application correspondante.

2.1 Multiplexage

Votre ordinateur dispose en effet d'une connexion (au moins) au réseau afin de faire entrer et sortir des informations. Le processus qui consiste à pouvoir faire transiter sur une connexion des informations provenant de diverses applications s'appelle le **multiplexage**. De la même façon le fait d'arriver à mettre en parallèle (donc répartir sur les diverses applications) le flux de données s'appelle le démultiplexage.



Ces opérations sont réalisées grâce au port, c'est-à-dire un numéro associé à un type d'application, qui, combiné à une adresse IP, permet de déterminer de façon unique une application qui tourne sur une machine donnée.

Le problème est alors de savoir sur une machine donnée quelle application fonctionne sur quel port. En effet, quand vous vous adressez à un serveur Web sur une machine donnée (www.google.com par exemple), vous ne spécifiez pas le port sur lequel écoute le programme qui va vous répondre… Comment résoudre ce problème ? En standardisant !

2.2 Assignation par défaut

Il existe des milliers de ports (ceux-ci sont codés sur 16 bits), c'est pourquoi une assignation standard a été mise au point par l'IANA (Internet Assigned Numbers Authority), afin d'aider à la configuration des réseaux et donc de définir quelques standards. Pour faire simple (on ne va pas entrer dans tous les détails pour ne pas vous donner trop d'informations) :

- Les ports `0` à `1023` sont les «ports reconnus» ou réservés («Well Known Ports»). Ils sont, de manière générale, réservés aux processus système (démons) ou aux programmes exécutés par des utilisateurs privilégiés (vous savez le super utilisateur que l'on a déjà étudié sous Unix et Windows). Un administrateur réseau peut néanmoins lier des services aux ports de son choix (des fois c'est bien de brouiller les cartes, même si ce n'est pas la solution ultime pour la sécurité)
- Les autres ports sont utilisables par vos applications pour faire les traitements adéquats.

Voici la liste de certains ports reconnus les plus couramment utilisés:

[1] Section issue du site « Comment ça marche » : http://www.commentcamarche.net/

Port	Service ou Application
22	SSH
25	SMTP
53	Domain Name System
80	HTTP
143	IMAP

2.3 Ports et Sécurité : Pare-feu

Tous ces ports sur votre machine sont disponibles pour les programmes qui tournent sur votre machine, mais sont potentiellement accessibles pour les machines extérieures. Ceci est un problème par rapport à la sécurité de votre machine.

Prenons l'analogie suivante : si on considère que votre ordinateur est une maison, les ports de l'ordinateur sont les portes de votre maison. Si vous laissez en permanence les portes de votre maison ouvertes, n'importe qui peut entrer ; la sécurité de votre maison n'est donc pas assurée. Et si l'on poursuit dans l'analogie, ces portes de la maison sont des portes battantes que l'on peut ouvrir ou fermer dans le sens entrant ou/et sortant.

La fermeture des ports sur votre machine est réalisée grâce à un logiciel qui est le pare-feu ou firewall en anglais. Mais la notion de firewall est plutôt un concept qui peut être réalisé par un logiciel ou un matériel spécifique pour protéger un réseau local entier. Par défaut, les systèmes actuels comme Windows 7 ou 8 sont installés avec un firewall activé empêchant toute communication entrante et autorisant a priori les communications sortantes.

2.3.1 Activation du firewall sous Ubuntu

Linux fournit un système très élaboré pour la mise en place des règles de votre pare-feu. Le mécanisme est inclus dans le noyau du système et est connu sous le nom de Netfilters. En tant qu'utilisateur, vous avez accès à la commande `iptables` qui vous permet de configurer très finement les règles que vous souhaitez mettre en place.

Certaines distributions actives par défaut un firewall sur votre machine. Ce n'est pas le cas par défaut sur la distribution Linux que vous utilisez (Ubuntu). Il est donc nécessaire d'activer le firewall pour protéger votre machine des attaques extérieures (et ainsi fermer toutes les portes de votre maison ; vous n'aurez alors plus qu'à ouvrir certaines portes au besoin). La commande sous Ubuntu pour configurer le pare-feu de votre machine est `ufw` (qui utilise elle-même `iptables`, mais qui est d'usage beaucoup plus simple).

Pour activer le firewall : `sudo ufw enable`

Une interface graphique permet de créer et supprimer les règles sur votre machine : `gufw` (à installer).

Dans la création d'une règle, on peut autoriser le port à s'ouvrir dans le sens entrée ou sortie (par défaut, tout le trafic dans le sens de votre ordinateur vers Internet (sortant) est autorisé, et tout le trafic entrant est interdit. Cela signifie que tout programme que vous lancez pourra envoyer des informations vers l'extérieur et recevoir des informations (car c'est lui qui ouvre un canal de communication avec l'entité extérieure pour recevoir les informations). A contrario, tout programme venant de l'extérieur de votre machine ne pourra accéder à aucun port de votre machine, sauf si vous lui ouvrez la porte.

TD séance n° 14
Réseau, Internet et Sécurité

3 Protocoles de Communication

3.1 Qu'est-ce qu'un protocole[2]

Un protocole de communication est une spécification de plusieurs règles pour un type de communication particulier. Communiquer consiste à transmettre des informations, mais tant que les interlocuteurs ne lui ont pas attribué un sens, il ne s'agit que de données et pas d'informations. Les interlocuteurs doivent donc non seulement parler un langage commun mais aussi maîtriser des règles minimales d'émission et de réception des données. C'est le rôle d'un protocole de s'assurer de tout cela.

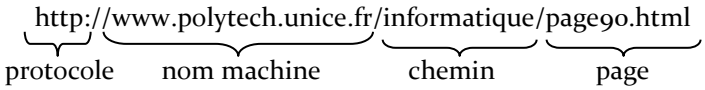
Par exemple, dans le cas d'un appel téléphonique :

- l'interlocuteur apprend que vous avez quelque chose à transmettre (après avoir composé le numéro, le combiné de l'interlocuteur appelé sonne) ;
- il indique qu'il est prêt à recevoir (vous attendez qu'il décroche et dise "Allô") ;
- il situe votre communication dans son contexte (« Je suis X. Je t'appelle pour la raison suivante… ») ;
- un éventuel destinataire final peut y être identifié (« Peux-tu prévenir Y que… ») ;
- le correspondant s'assure d'avoir bien compris le message (« Peux-tu me répéter le nom ? ») ;
- les procédures d'anomalies sont mises en place (« Je te rappelle si je n'arrive pas à le joindre ») ;
- les interlocuteurs se mettent d'accord sur la fin de la communication (« Merci de m'avoir prévenu »).

Au niveau des applications, un seul protocole universel n'est pas envisageable pour traiter toutes les cas de figure (tous les types d'échanges possibles). Donc plusieurs protocoles ont été développés pour chacun des cas d'utilisation : recevoir du mail, envoyer un mail, récupérer un document, …

3.2 URL : Uniform Resource Locator

Une URL est un format de nommage universel pour désigner une ressource sur Internet. Il s'agit d'une chaîne de caractères imprimables qui se décompose en cinq parties :



C'est ce que vous utilisez dans votre navigateur pour désigner le document auquel vous souhaitez accéder sur internet (nom de la machine, chemin pour accéder au document sur cette machine et document que vous souhaitez consulter). Le protocole permet de désigner le langage utilisé entre les programmes pour savoir quels sont les mots utilisés pour poser les questions et envoyer les réponses.

Nous allons maintenant étudier d'un peu plus près HTTP pour échanger des documents (HyperText Transfer Protocol) et SSH, un protocole de communication sécurisé pour se connecter à distance sur une machine.

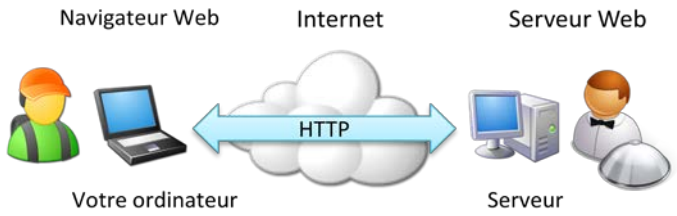
3.3 Exemples de protocoles : HTTP et HTTPS

3.3.1 HTTP : HyperText Transfer Protocol

HTTP est un protocole qui n'est pas sécurisé. Toutes les communications que vous faites, donc les messages que vous envoyez et les réponses que vous recevez (comme par exemple demander de récupérer le document à une adresse donnée et récupérer le document à cette adresse) vont transiter sur le réseau en clair. Cela signifie que toute personne utilisant un programme qui écoute les messages sur le réseau pourra voir tout ce que vous faites.

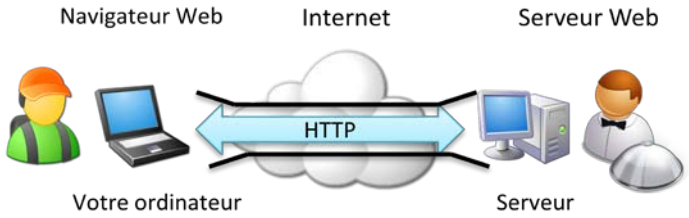
[2] Section réalisée à partir de la définition de Protocole de Communication sur Wikipédia : http://fr.wikipedia.org/wiki/Protocole_de_communication

En effet, nous avons vu lors de la semaine dernière avec Wireshark que nous pouvons intercepter les messages qui passent sur notre interface réseau (et donc savoir quel est le document que la personne utilisant un ordinateur vient de demander) et ce message circule en clair sur Internet. Donc sans être paranoïaque, on peut dire que tout ce que l'on fait sur Internet est visible. Non heureusement (même si on laisse toujours des traces de son activité, rappelez-vous les cours précédents).



3.3.2 HTTPS :

Donc attention, quand vous entrez un mot de passe ou un code de carte bancaire que l'adresse à laquelle vous êtes connectés n'utilise pas le protocole HTTP, mais bien le protocole HTTPS (HTTP sécurisé par SSL). Quand vous vous connectez à un site marchand, et avant d'effectuer une transaction, vérifiez bien ce point sur vos navigateurs. Dans ce cas les échanges que vous faites entre le site marchand et votre ordinateur transitent dans une connexion où tous les messages sont cryptés.



Le cadenas vous indique que les communications entre votre navigateur et le site web sont sûres: personne ne peut les espionner, et personne ne peut trafiquer les communications. Mais il ne garantit **rien d'autre !** Pour illustrer ce problème, prenons l'exemple suivant : HTTPS (le cadenas), c'est un peu comme un fourgon blindé qui vous assure la sécurité du transport. Le fourgon blindé ne vous garantit pas que la banque utilise de bons coffres forts et qu'elle les ferme bien. Le fourgon blindé ne garantit pas non plus que la banque ne fait pas de malversations. Et des truands peuvent louer les services d'un fourgon blindé, des pirates et voleurs peuvent très bien créer un site sécurisé (avec le petit cadenas). Donc faites toujours attention à vous adresser à des sites de confiance[3].

Mais comment ça marche ?

3.3.3 SSL (Secure Socket Layer)[4]

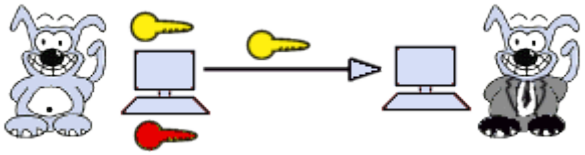
La sécurisation des transactions par SSL 2.0 est basée sur un échange de clés entre client et serveur. La transaction sécurisée par SSL se fait selon le modèle suivant :

- Dans un premier temps, le client (votre ordinateur) se connecte au site marchand sécurisé par SSL et lui demande de s'authentifier. Le client envoie également la liste des crypto systèmes qu'il supporte.

[3] http://www.fia-net.com/

[4] Section issue du site « Comment ça marche » : http://www.commentcamarche.net/contents/215-ssl-secure-sockets-layers

- Le serveur à réception de la requête envoie un certificat au client, contenant la clé publique du serveur, signée par une autorité de certification (CA), ainsi que le nom du crypto système le plus haut dans la liste avec lequel il est compatible (la longueur de la clé de chiffrement la plus grande sera préférée).
- Le client vérifie la validité du certificat (donc l'authenticité du marchand), puis crée une clé secrète aléatoire, chiffre cette clé à l'aide de la clé publique du serveur, puis lui envoie le résultat (la clé de session).



- Le serveur est en mesure de déchiffrer la clé de session avec sa clé privée. Ainsi, les deux entités sont en possession d'une clé commune dont ils sont seuls connaisseurs. Le reste des transactions peut se faire à l'aide de clé de session, garantissant l'intégrité et la confidentialité des données échangées.

Ceci permet ainsi d'échanger de l'information en cryptant chacun des messages entre les deux machines (le client et le serveur) et donc d'assurer la **confidentialité** des échanges. En fait, en plus de la confidentialité, ssl assure aussi l'**intégrité** (il est impossible de truquer les informations échangées) et l'**authenticité** (il permet de s'assurer de l'identité du programme, de la personne ou de l'entreprise avec laquelle on communique.)

3.4 Exemple de protocole SSH : Secure Shell

SSH est un protocole plus complet que SSL, mais qui a le même but : sécuriser les communications entre une machine A et une machine B.

Voici les étapes de l'établissement d'une connexion SSH :

- Le serveur envoie sa clef publique au client. Celui-ci vérifie qu'il s'agit bien de la clef du serveur, s'il l'a déjà reçue lors d'une connexion précédente.
- Le client génère une clef secrète et l'envoie au serveur, en chiffrant l'échange avec la clef publique du serveur (chiffrement asymétrique). Le serveur déchiffre cette clef secrète en utilisant sa clé privée, ce qui prouve qu'il est bien le vrai serveur.
- Pour le prouver au client, il chiffre un message standard (Cf. RFC4256) avec la clef secrète et l'envoie au client. Si le client retrouve le message standard en utilisant la clef secrète, il a la preuve que le serveur est bien le vrai serveur.
- Une fois la clef secrète échangée, le client et le serveur peuvent alors établir un canal sécurisé grâce à la clef secrète commune (chiffrement symétrique).
- Une fois que le canal sécurisé est en place, le client va pouvoir envoyer au serveur le login et le mot de passe de l'utilisateur pour vérification. Le canal sécurisé reste en place jusqu'à ce que l'utilisateur se déconnecte.

La commande qui permet cela est la commande `ssh` en spécifiant le nom de l'utilisateur et la machine sur laquelle on souhaite se connecter : `ssh user@host.domain.fr` ou `ssh user@adresse_ip`.

Le protocole SSH vous permet ainsi de vous connecter à un ordinateur distant de façon sûre et d'avoir accès à un interpréteur de commandse qui va vous permettre d'exécuter des commandes sur la machine sur laquelle vous vous êtes connectés.

Et nous retrouvons ici l'intérêt d'avoir appris à utiliser les lignes de commandes au début du cours ! La boucle est bouclée (qu'est-ce qu'il est bien fait ce cours 😊 !)

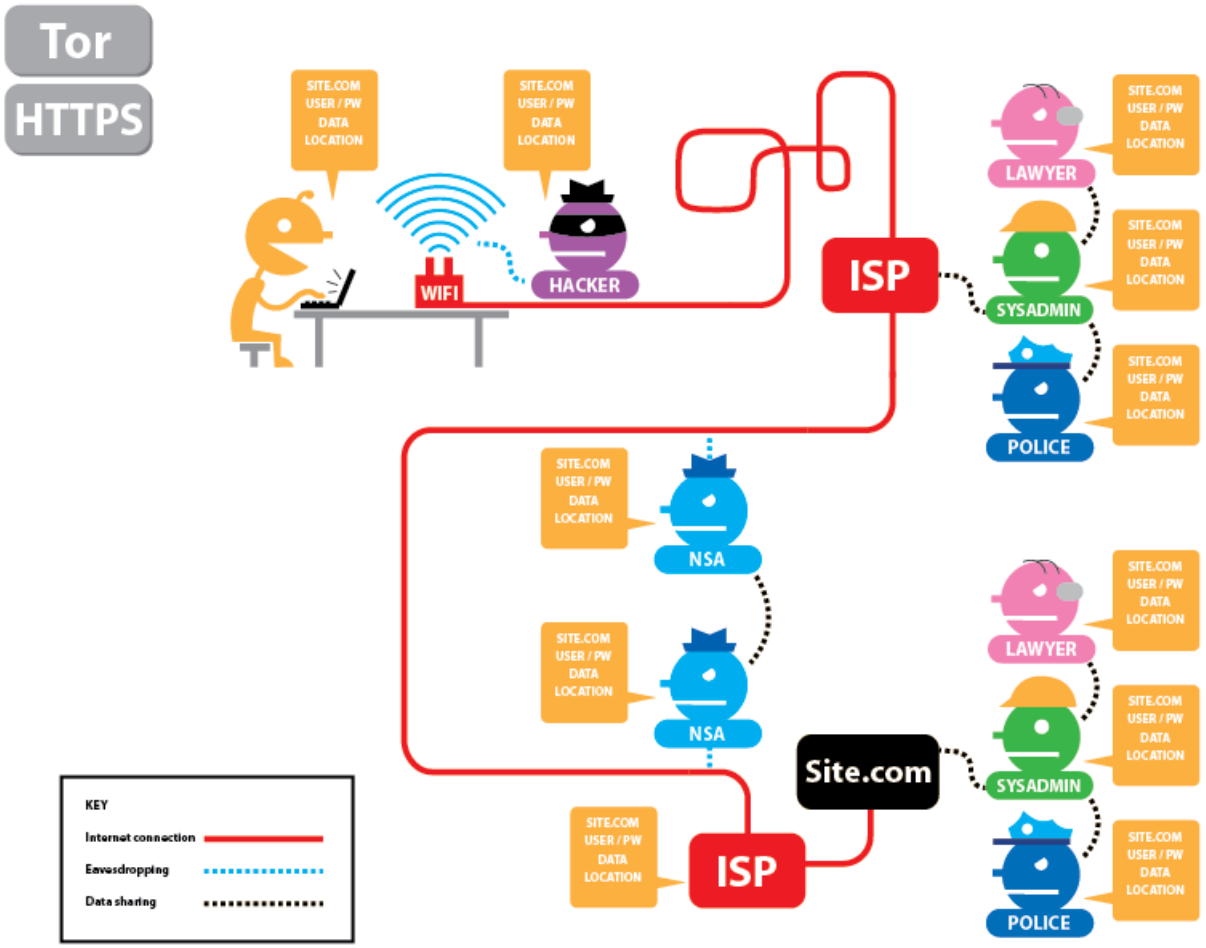
La semaine prochaine, nous débuterons un nouveau chapitre sur la programmation shell (ou comment enchainer les commandes que nous avons apprises au début de l'année).

TD séance n° 14
Réseau, Internet et Sécurité

4 Et si j'avais Tor ?[5]

4.1 Présentation

(acronyme de The Onion Router, littéralement : « le routeur oignon ») est un réseau informatique superposé mondial et décentralisé. Le réseau Tor peut ainsi rendre anonymes tous les échanges Internet basés sur le protocole de communication TCP. Sans entrer dans les détails, voici différents cas d'utilisation de l'Internet. La première illustration vous montre ce que différents acteurs du Web peuvent voir si vous naviguez sur Internet avec le simple protocole HTTP.



Sur ce schéma, ISP est votre fournisseur d'accès à Internet (que vous soyez à la maison ou à l'Université par exemple). C'est celui qui vous fournit le moyen d'accéder au réseau Internet.

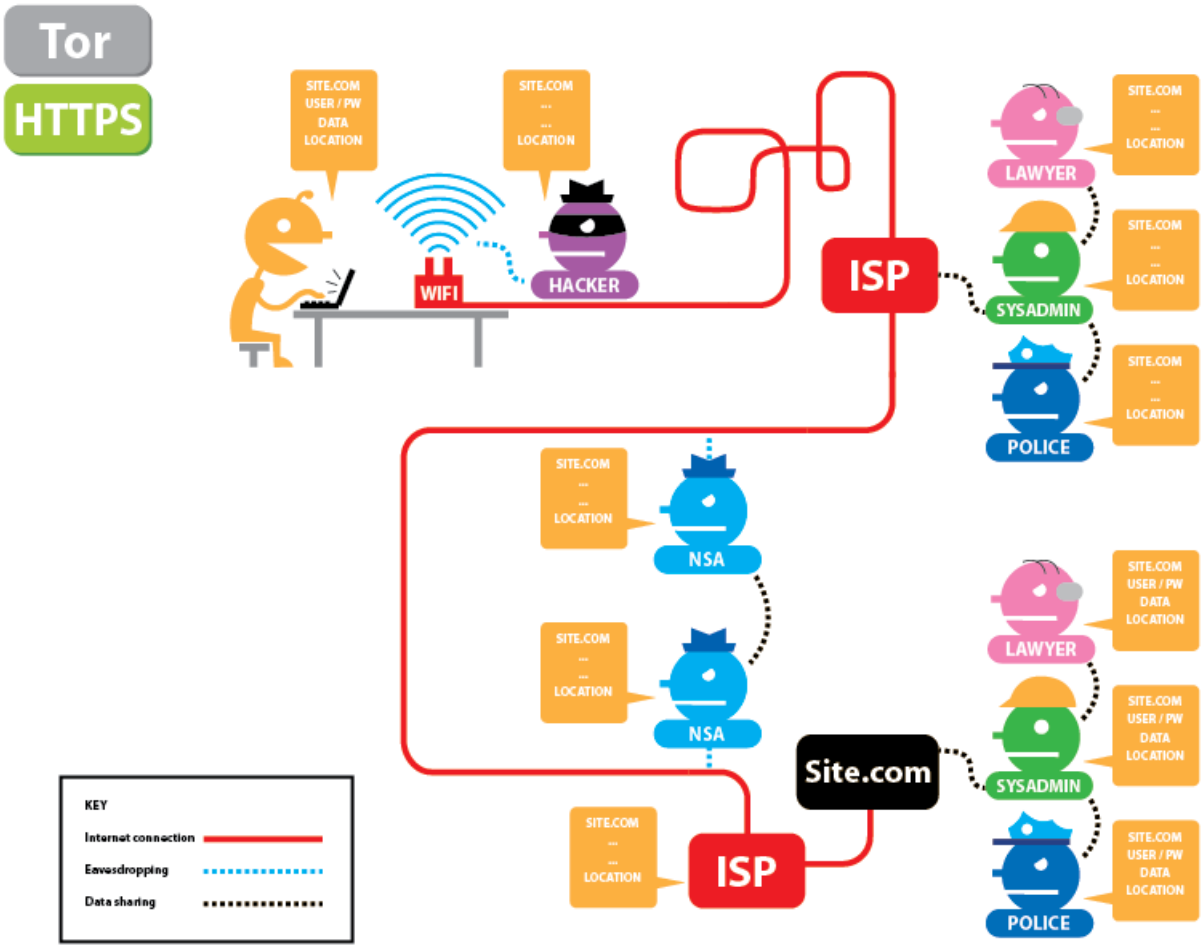
On peut voir sur ce schéma que, dans le cas où on utilise HTTP, tout le monde peut voir toutes les informations qui sont véhiculées sur le réseau : nom du site consulté, nom d'utilisateur, mot de passe, donnée ainsi que votre localisation (adresse IP mais aussi localisation géographique).

Dans le cas où vous utilisez HTTPS, que nous avons présenté précédemment, aucun intermédiaire ne pourra voir les informations sur l'utilisateur lui-même (nom, mot de passe utilisé pour se connecté) ; seul l'utilisateur et le

[5] Section réalisée grâce aux illustrations du site EFF : https://www.eff.org/pages/tor-and-https

TD séance n° 14
Réseau, Internet et Sécurité

destinataire des messages pourront connaître le contenu échangé. Mais « tout le monde » pourra savoir quel site vous consultez et où vous vous trouvez (votre adresse IP).

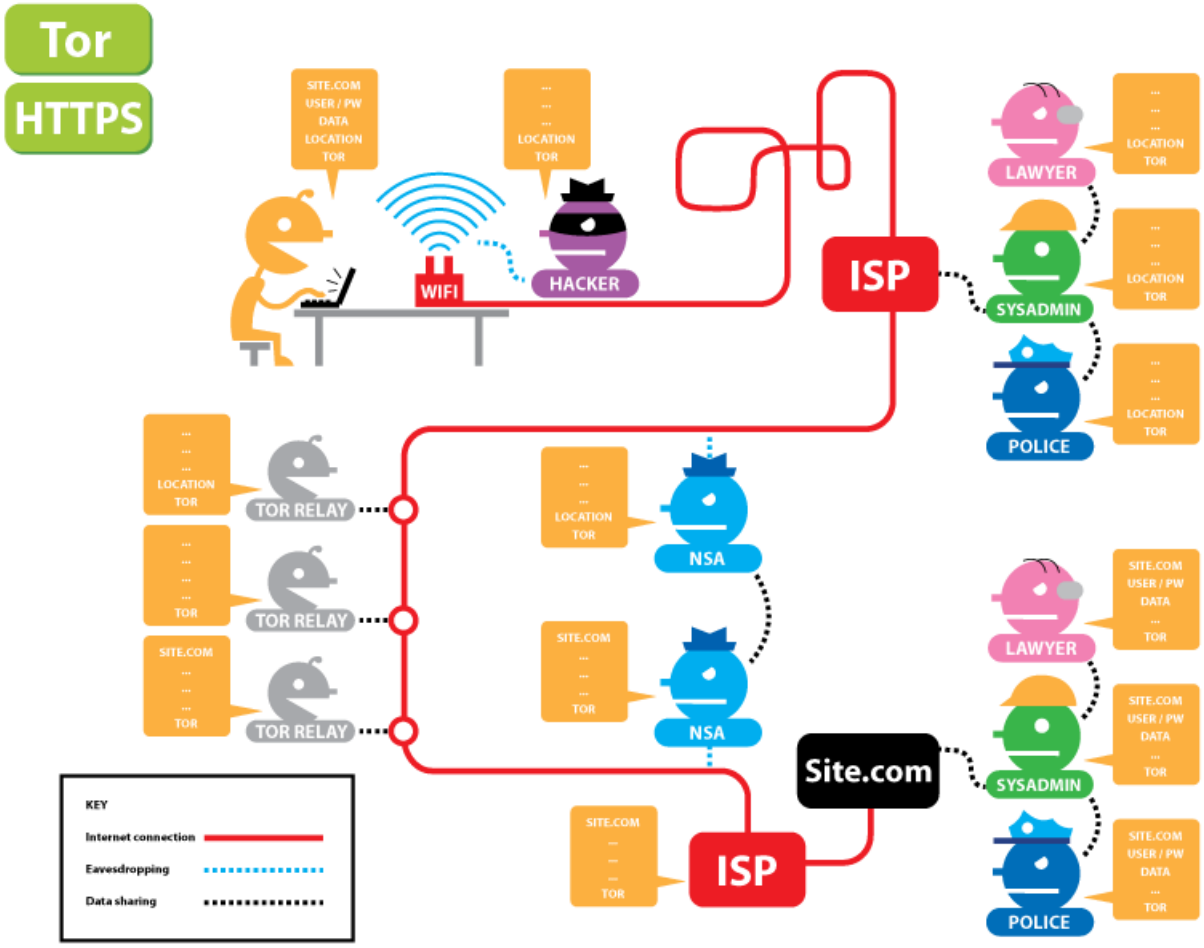


Dans le cas de l'utilisation de Tor, seul les premiers intermédiaires pourront savoir quel est votre localisation (ou adresse IP) et le fait que vous utilisez Tor et bien sûr les personnes ne pourront pas voir les données échangées ni les informations personnelles (nom, mot de passe), grâce au fait que vous utilisez HTTPS.

Tor réduit les risques d'analyses de trafic simples ou sophistiquées, en répartissant vos transactions entre plusieurs endroits de l'Internet. On ne peut donc pas, en observant un seul point, vous associer à votre destinataire. C'est comme utiliser un chemin tortueux et difficile à suivre pour semer un poursuivant (tout en effaçant de temps en temps ses traces). Au lieu d'emprunter un itinéraire direct entre la source et la destination, les paquets de données suivent une trajectoire aléatoire à travers plusieurs serveurs qui font disparaître vos traces. Personne ne peut donc déduire de l'observation d'un point unique, ni d'où viennent, ni où vont les données.

Le principe général est donc de rajouter une chaîne d'intermédiaires pour brouiller les pistes et faire en sorte que chaque intermédiaire ne connaisse que l'adresse du précédent et du suivant. Ainsi, « on perd votre trace » ou plus exactement, il est très difficile de reconstituer la chaîne complète que sont l'expéditeur et le destinataire. Cela peut donc vous procurer un certain anonymat sur Internet.

TD séance n° 14
Réseau, Internet et Sécurité



4.2 Mise en garde[6]

Mais attention, le revers de la médaille est que quand on utilise un réseau comme Tor, on peut se voir refuser l'accès à des sites « classiques ». L'anonymat, c'est bien mais ça ne permet pas d'aller partout, par exemple les utilisateurs de Tor sont bannis Wikipédia, que ce soit pour les utilisateurs soient inscrits ou pas. D'autre part, on peut remarquer une baisse de vitesse dans les connexions donc si vous aimez surfer rapidement, vous allez être déçu.

L'anonymat n'est pas synonyme de sécurité, en effet votre adresse IP est masquée et les données transitant entre les relais sont chiffrées, mais les données transitant entre le dernier relais et leur destination finale circulent en clair, lorsque vous consultez votre compte en banque ou effectuez des achats en ligne, Tor ne suffit donc pas (pour les sites n'étant pas des services cachés) : il faut utiliser HTTPS ou un chiffrement final similaire et des mécanismes d'authentification.

Enfin, sachez cependant que même avec HTTPS, vos données peuvent être interceptées ! Un logiciel, SSLstrip, permet de sniffer les données chiffrées au niveau du dernier relais.

[6] Section issue de la documentation d'Ubuntu : http://doc.ubuntu-fr.org/tor

TD séance n° 14
Réseau, Internet et Sécurité

5 Exercices

5.1 Ports

Exercice n°1:

Combien de ports sont disponibles sur une machine ? Quels sont les ports que vos programmes peuvent utiliser sans être lancés par le super utilisateur ?

Exercice n°2:

Quels sont les services ou applications qui s'exécutent par défaut derrière les ports 25, 53 et 143 (vous donnerez l'acronyme et sa signification) ?
Citez pour chacun de ces services à quelle occasion vous les utilisés ? (vous avez déjà utilisé cela lors des TD précédents ; voir TD1 et TD12)

5.2 Protocoles

Exercice n°3:

Quand vous utilisez votre navigateur, quel est le protocole par défaut que vous utilisez pour récupérer un document sur Internet ? Sur quel port contactez-vous la machine qui vous renvoie le document ?

5.3 Pare-feu

Exercice n°4:

Donnez la commande pour activez le pare-feu sur votre machine sous Ubuntu. Dites comment installer l'interface graphique du pare-feu sous Ubuntu. Utilisez cette interface graphique pour autoriser quiconque à se connecter au service `ssh` sur votre machine (utiliser l'onglet *Préconfigurée*). Quel est le port qui est maintenant ouvert sur votre machine ?
Ajouter une deuxième règle à l'aide de l'onglet *Simple* pour autoriser les programmes extérieurs à contacter un serveur web sur votre machine (protocole HTTP). Quel est le port que vous devez spécifier comme étant maintenant autorisé.
Vous pouvez aussi constater que vous pouvez contrôler que vous pouvez même autoriser uniquement certaines machines (à partir de leur adresse IP par exemple) à communiquer avec la vôtre sur un port spécifique (voir l'onglet *Avancé*).

5.4 Connexion à distance : SSH

Exercice n°5:

Quelle est la commande pour installer un serveur ssh sur votre machine (paquetage ssh).
Pour finaliser cette partie du TD, vous devez être deux. Donc trouvez-vous un binôme.
Après avoir fait cette installation, ajoutez un compte pour votre binôme sur votre machine. Si le binôme est Stéphane et Dino, Stéphane créera un compte pour Dino sur sa machine et Dino créera un compte pour Stéphane sur sa machine. Vous penserez bien à ajouter un mot de passe au compte que vous communiquerez à votre binôme (chacun le fait sur sa machine). Veillez à créer un compte utilisateur avec un répertoire personnel.
Votre compte étant créé sur la machine de votre binôme, le serveur SSH étant installé et le pare-feu étant ouvert pour le protocole SSH (exercice précédent), connectez-vous maintenant sur la machine de votre binôme. Tester que vous pouvez bien lancer quelques commandes.