Université Nice Sophia Antipolis
J.Y. Tigli, H. Affes, S. Lavirotte
Polytech'Nice – Sophia

# Internet et Sécurité

BAT3
2013-2014

## 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 ?

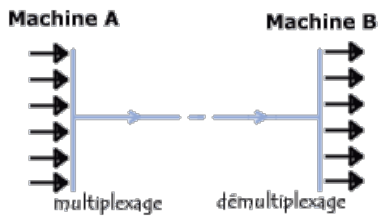
# 2 Les ports

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 à 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 données 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 où tourne le programme qui va vous répondre est à l'écoute... 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 prendre 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 étudier 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)

- Les autres ports sont utilisables par vos applications pour faire les traitements adéquats.

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

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

### 2.3 Ports et Sécurité

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. Si on prend 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 portent 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 de Windows XP à Windows 8 ou encore Linux sont installés avec un firewall activé empêchant toute communication entrante et autorisant a priori les communications sortantes.

## 3 Protocoles de Communication

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, … On va étudier d'un peu plus près HTTP pour échanger des documents (HyperText Transfer Protocol) et SSH, un protocole de communication sécurisé que nous allons étudiés un peu plus en détail.

## 3.1 HTTP et HTTPS

### 3.1.1 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 une adresse données 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. En effet, nous avons vu 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). 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).

### 3.1.2 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.

Mais comment ça marche ?

### 3.1.3 SSL

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, triée par ordre décroissant selon la longueur des clés.

- Le serveur a 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 - 40 bits ou 128 bits - sera celle du crypto système commun ayant la plus grande taille de clé).
- Le client vérifie la validité du certificat (donc l'authenticité du marchand), puis crée une clé secrète aléatoire (plus exactement un bloc prétendument 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.

## 3.2 SSH : Secure Shell

SSH est un protocole plus complet que SSL, mais qui a en soit le même but : sécuriser les communication 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.

# 4 Exercices

## 4.1 Ports

**Exercice n°1:**

Combien de ports sont disponibles sur une machine ?

**Exercice n°2:**

A quoi correspondent les ports 25, 53 et 143 (quelles sont les applications ou services qui s'exécutent par défaut derrière ces ports) ?
Citez pour chacun de ces services à quelle occasion vous les utilisés ? (vous avez déjà utilisé cela lors des TD précédents ; voir TD1 et TD13)