

Séance 10 : Initiation aux notions de :

- réseaux informatiques,
- sécurité Web.

INF27507 – Technologies du commerce électronique

Prof. Yacine YADDADEN, Ph. D.

Notions de réseaux

Plan

1. Introduction et Contexte
2. Définition et historique
3. Le modèle par couche
4. Adressage réseau
5. Les équipements réseaux
6. Les protocoles et ports pour le Web
7. Serveurs **DNS**, **DHCP** et **Pare-feu**
8. Quelques commandes réseau utiles
9. Mise en pratique – *Filius ou Packet Tracer*
10. Démonstration et exercices
11. Questions et discussion

Introduction et Contexte

- **Internet** est l'une des *inventions* qui a révolutionné le monde,
- Cet immense **réseau mondial** a permis :
 - Une *communication instantanée* à travers le monde,
 - L'*interconnexion* des machines et équipements,
 - Mise à disposition d'une *quantité phénoménale d'information*,
 - ...
- Cependant, avant d'être ce qu'il est aujourd'hui, *Internet* a été le fruit d'une *évolution* progressive en commençant par réseau de petite envergure,
- Afin de comprendre son *fonctionnement*, il faut comprendre les *notions de base des réseaux informatiques*.

Historique

- **Informatique centralisé**

- Dans les années 50-60, les données sont gérées sur des ordinateurs centraux,
- Les terminaux avec matériels de communication échangent avec le système central,
- Exemple : **SNA** (**S**ystem **N**etwork **A**rchitecture) d'**IBM**.

- **Premier réseau informatique à grande échelle**

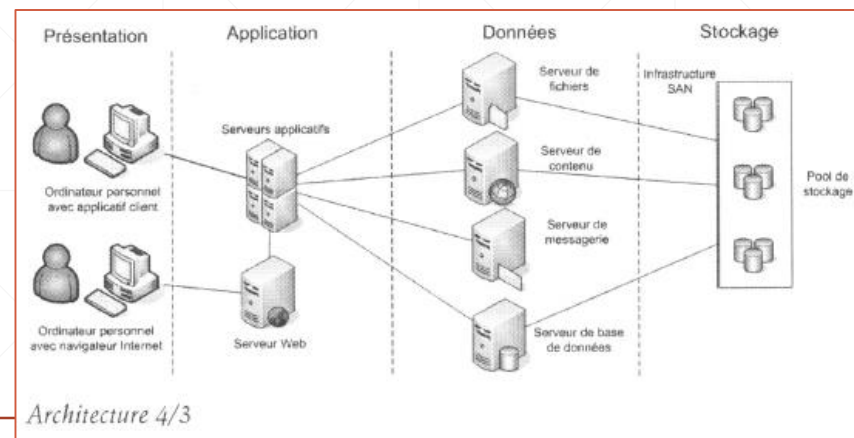
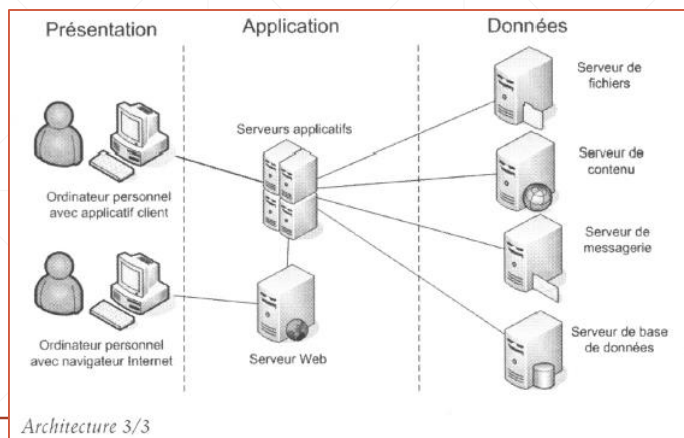
- Fin des années 60, Développement de matériels et protocoles pour un réseau avec forte tolérance de pannes (*en cas de guerre*) par le **DoD** (**D**epartment **o**f **D**efense),
- Apparition de l'**ARPANET** (**A**dvanced **R**esearch **P**roject **A**gency **N**ETwork) en 1970,
- En 1980, **ARPANET** inclut la liaison satellite par paquets avec Londres.

Historique (*suite*)

- **Développement de la norme TCP/IP**
 - Au milieu des années 70, **ARPANET** adopte un nouveau mode de communication,
 - **TCP/IP** (Transmission Control Protocol/Internet Protocol),
 - L'agence **DARPA** met à disposition du public le protocole **TCP/IP**.
- **Informatique répartie**
 - En 1981, apparition de l'ordinateur personnel ou **PC** (Personnal Computer) d'**IBM**,
 - Le traitement global est divisé en sous-tâches et réparti sur différents postes,
 - Apparition de la notion de petits réseaux et autres réseaux locaux.

Définitions

- **Qu'est-ce qu'un réseau informatique :** « *Un réseau est un groupement de deux ou plusieurs ordinateurs ou autres appareils électroniques permettant l'échange de données et le partage de ressources communes.* »,
- Un réseau informatique est constitué d'*ordinateurs* et de *systèmes d'exploitation hétérogènes* **interconnecté** à travers l'**Internet**,
- Plusieurs architecture basées sur des **étages** (*tiers*) :



Les éléments d'un réseau informatique

D'un point de vu **logiciel** :

- Système d'exploitation réseau avec plusieurs couches logicielles (protocole de communication, couche application, ...),
- Deux catégories :
 - **Client** : *celui qui demande le service (poste de travail),*
 - **Serveur** : *celui qui offre et met à disposition le service.*
- **Exemples** : *Microsoft Windows Server, distribution Linux (Ubuntu, RedHat, ...), etc.*

Les éléments d'un réseau informatique

D'un point de vu **matériel** :

- **Interconnexion**

- C'est la *liaison* entre deux équipements afin d'activer la communication,
- Elle peut être **filaire** (*câble Ethernet ou Fibres optiques*) ou **sans fil** (*laser, ondes radio, ...*).

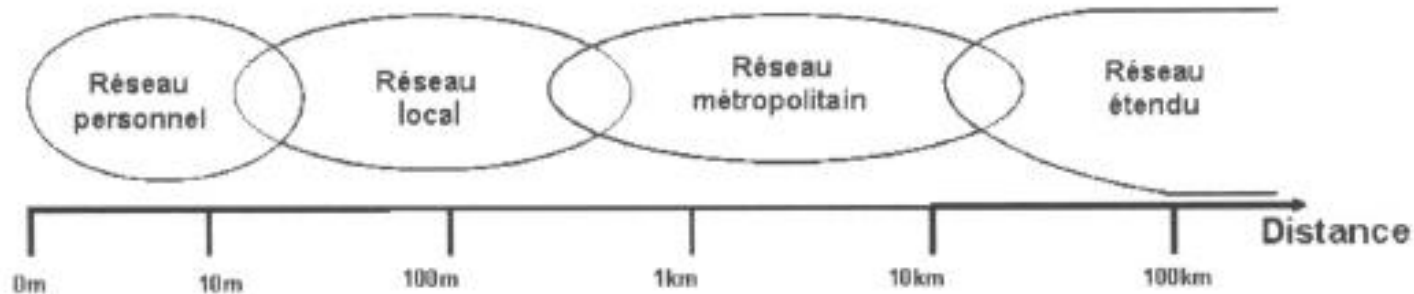
- **Protocoles de communication**

- C'est les *règles et normes* qui régissent la communication,
- Elles *donnent un sens au signal* qui circule dans le réseau informatique.

Catégories de réseaux informatique

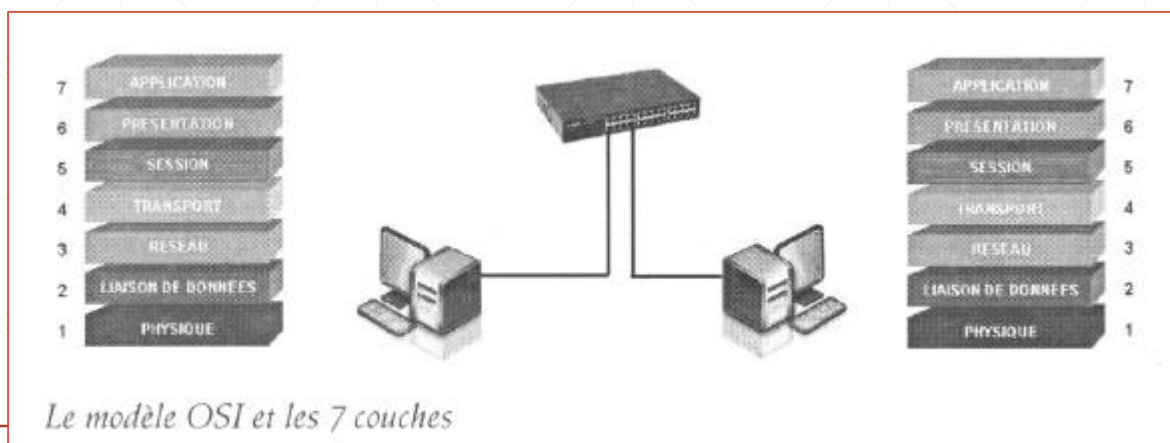
Suivant le périmètre d'action du réseau, on définit :

- **PAN** (Personal Area Network),
- **LAN** (Local Area Network),
- **MAN** (Metropolitan Area Network),
- **WAN** (Wide Area Network).



Le modèle par couche

- L'organisme **ISO** a défini en 1984 un modèle de référence :
 - **OSI** (**O**pen **S**ystem **I**nterconnection),
 - Afin de normaliser les échanges entre deux machines.
- Il est constitué de *sept couches* :
- On parle d'**encapsulation de paquets**.



Rôle de chaque couche

1. Couche physique

- Transmission bit à bit ou binaire (0 ou 1) sur le support physique.

2. Couche liaison

- Traduction des données numérique en signal.

3. Couche réseau

- Choix du meilleur chemin afin d'atteindre le destinataire (*algorithme Dijkstra*).

4. Couche transport

- S'assurer que tout les données ont été transmises sans problème,
- **TCP** (Transport **C**ontrol **P**rotocol) ou **UDP** (**U**ser **D**atagram **P**rotocol).

Lent avec confirmation de réception

Rapide sans confirmation de réception

Rôle de chaque couche

5. Couche session

- Elle gère et maintient le mode connecté ainsi que les points de synchronisation.

6. Couche présentation

- Assure la mise en forme des données.

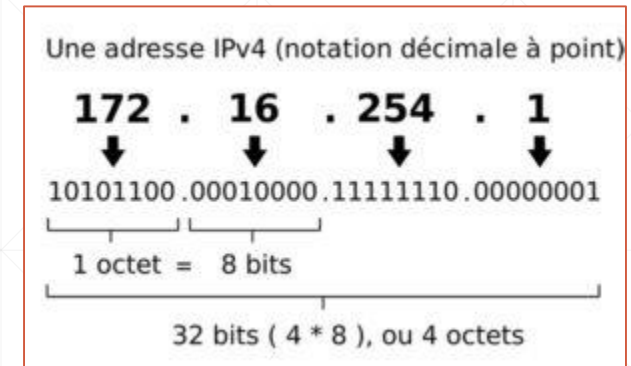
7. Couche application

- Assure l'interface de communication avec l'utilisateur.

Adressage réseau

On dispose des éléments suivant :

- **Adresse MAC**
 - *Adresse physique de la machine ou de l'équipement.*
- **Adresse IPv4 (4 octets) ou IPv6 (16 octets)**
 - *Identifiant numérique et unique permettant d'identifier un élément du réseau,*
 - *Privée (**LAN** et temporaire) et publique (**WAN** et permanente),*
- **Masque de sous-réseau**
 - *Il permet de distinguer la partie de l'adresse IP correspondant au réseau de la partie identifiant le nœud en question.*



Adressage réseau (*suite*)

- **Passerelle par défaut**

- *C'est l'adresse IP d'un point réseau agissant comme une **porte d'entrée** vers un autre réseau différent du réseau courant.*

- **Adresse DNS**

- *C'est l'adresse IP du serveur permettant la traduction des noms en adresses IP.*

- **Adresse DHCP**

- *C'est l'adresse IP de l'équipement permettant l'attribution automatique des adresses IP.*

Les équipements réseaux

- Il y en a plusieurs :
 - Le **répéteur** (*transceiver*)
 - *Reconditionne les données reçues et les retransmet afin d'accroître la distance de transmission.*
 - Le **commutateur** (*switch*)
 - *Il intègre la fonction de concentrateur et permet de lier plusieurs supports physiques différents.*
 - Le **routeur** (*router*)
 - *Il a les fonctions du commutateur et permet en plus de lier plusieurs réseaux différents.*
 - La **passerelle** (*gateway*)
 - *Moins utilisé, il permet d'interconnecter deux réseaux de niveaux différents : **LAN** avec **WAN**.*
- Chacun joue un rôle précis et on en trouve plusieurs au sein d'un même réseau.

Les protocoles et ports pour le Web

Il y a plusieurs protocoles et ports :

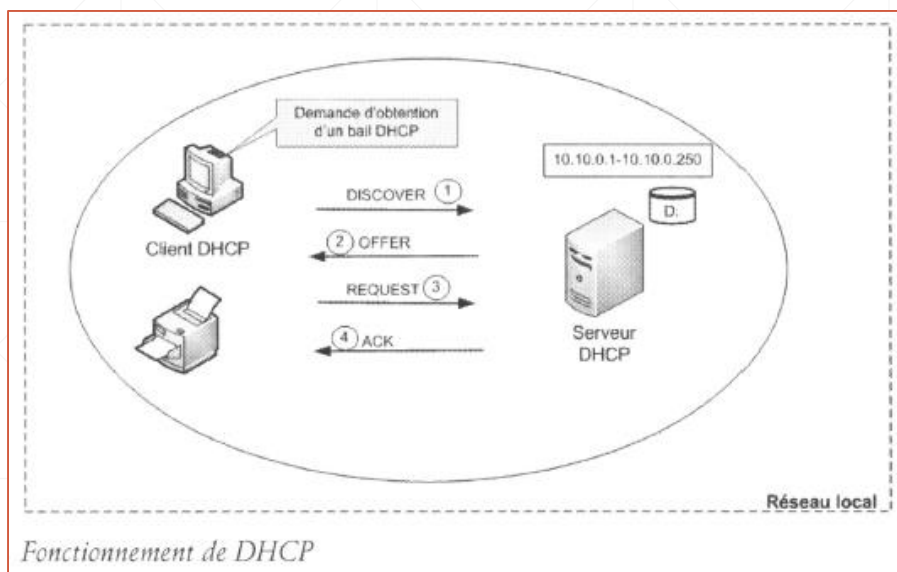
- **HTTP** ou *HyperText Transfert Protocol* (port **80**),
- **HTTPS** ou *HyperText Transfert Protocol Secure* (port **443**),
- **SSH** ou *Secure Shell* (port **22**),
- **FTP** ou *File Transfer Protocol* (port **20** données et **21** commandes),
- **DNS** ou *Domain Name System* (port **53**),
- **SMTP** ou *Simple Mail Transfer Protocol* (port **25**),
- **POP3** ou *Post Office Protocol* (port **110**),
- **IMAP** ou *Internet Message Access Protocol* (port **143**).

Serveurs DNS, DHCP et Pare-feu

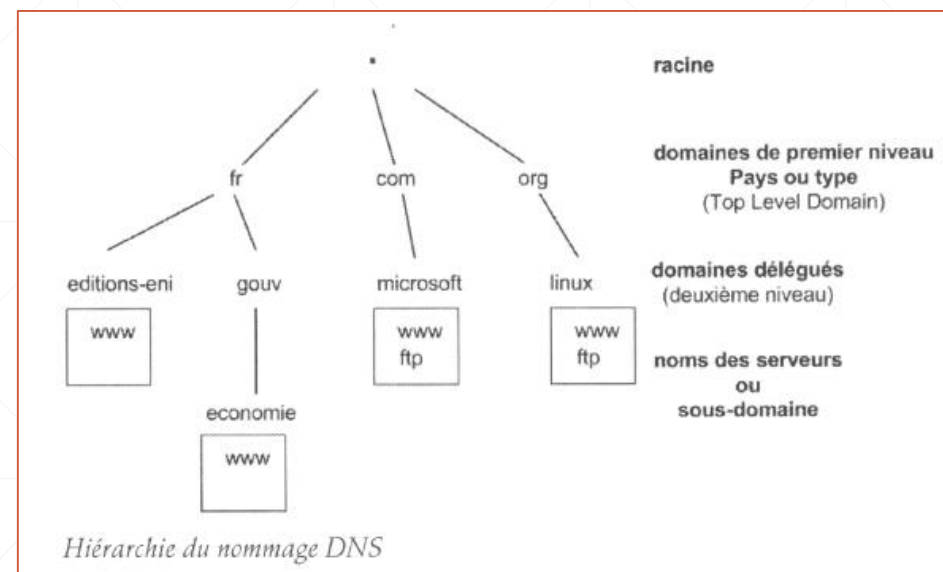
- **Définition du DNS :** « ou **Domain Name System** est un service permettant une résolution à base de noms hiérarchiques et distribués pour les hôtes IP connectés au réseau. »,
- On peut se rappeler plus facilement des noms que des adresses IP.
- **Définition du DHCP :** « ou **Dynamic Host Configuration Protocol** est un service permettant de définir de manière centralisée une configuration TCP/IP complète pour l'ensemble des périphériques réseau. »,
- La configuration peut être statique ou dynamique.
- **Définition du Pare-feu :** « ou **FireWall** en anglais est un composant réseau permettant de contrôler, sécuriser et gérer le flux du trafic. Il est primordial dans les réseaux d'entreprise. »,
- Il peut se présenter sous forme **logiciel** (pfSense) ou **matériel** (FortiGate).

Serveurs DNS et DHCP (*suite*)

DHCP




DNS

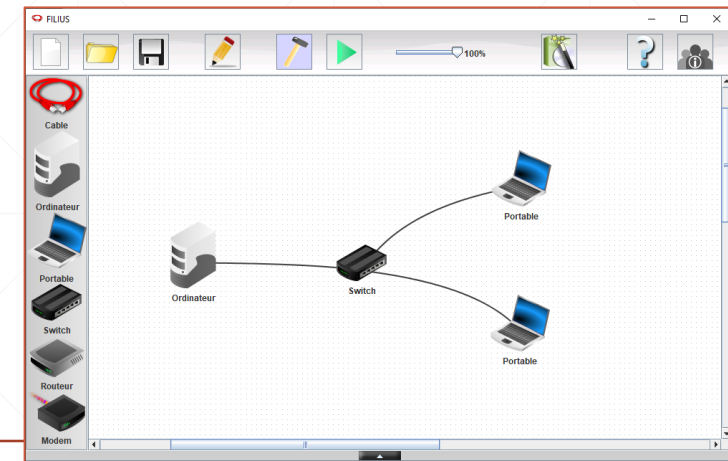


Quelques commandes réseau utiles

- La commande **ping** :
 - Exploite le *protocole ICMP* afin de *vérifier la communication* entre deux équipements.
- La commande **tracert** ou **traceroute** :
 - Permet de *mettre en évidence la route réseau empruntée* pour atteindre le destinataire.
- La commande **ipconfig** ou **ifconfig** :
 - Permet d'afficher toute la *configuration réseau TCP/IP* de la machine.
- La commande **netstat** :
 - Permet d'afficher la *liste de l'ensemble des ports ouverts et actifs* dans une machine.
- La commande **nslookup** :
 - Permet de faire la **résolution** d'un *nom de domaine en adresse IP*.

Mise en pratique – *Filius* ou *Packet Tracer*

- **Définition :** « *Filius* est un logiciel de simulation de réseaux informatiques. Il permet de créer son propre réseau de le configurer, de le simuler et de **visualiser** les échanges **d'informations**.. », 
- **Lien de téléchargement :** <https://www.lernsoftware-filius.de/Herunterladen>
- Alternative utilisée lors de formation **CISCO** :
 - **Packet Tracer,**
 - **Lien :** <https://www.netacad.com/fr/courses/packet-tracer>



Démonstration

- En utilisant **Filius**, on va créer deux *réseaux LAN* :
 - Réseau A 192.168.1.0 / 255.255.255.0
 - Avec une **passerelle** 192.168.1.100
 - Un **serveur DHCP** 192.168.1.150
 - Deux **laptops** (*configuration TCP/IP automatique*)
 - Réseau B 192.168.2.0 / 255.255.255.0
 - Avec un **passerelle** 192.168.2.100
 - Un **serveur DHCP** 192.168.2.150
 - Deux **laptops** (*configuration TCP/IP automatique*)
- Interconnecter les deux *réseaux LAN* en utilisant un **routeur**.
- Le fichier de l'architecture réseau sera mis à disposition sur Moodle.

Voir la vidéo de
démonstration fournie



Exercice – *Compte pour l'assiduité*

En vous basant sur l'architecture précédente :

1. Rajouter un autre réseau **LAN** 192.168.3.0 / 255.255.255.0
2. Avec un **passerelle** 192.168.3.100
3. Rajouter un les équipements suivants :
 - **Serveur DNS** 192.168.3.10 / 255.255.255.0 avec
 - www.uqar.ca (192.168.3.11) et mail.uqar.ca (192.168.3.12)
 - **Serveur Web** permettant d'afficher le message avec l'image
 - Adresse IP : 192.168.3.11 / 255.255.255.0
 - Le test d'affichage se fait sur un **Laptop** d'un des deux *réseaux LAN*.
 - **Serveur Mail** 192.168.3.12 / 255.255.255.0
 - Créer deux boîtes e-mails albert@uqar.ca et amelie@uqar.ca
 - Sur deux **Laptops** des deux *réseaux LAN*, tester l'envoi et réception d'e-mails.



Bienvenue !
UQAR

Notions de sécurité Web

Plan

1. Introduction et Contexte
2. La sécurité informatique
3. Contrôle des entrées/sorties
4. Les attaques les plus communes
5. Chiffrement et authentification
6. Exemple d'outil – *Wireshark*
7. Questions et discussion

Définitions

- **Qu'est-ce que la sécurité :**
 - « *C'est le fait d'être protégé ou être sous la protection d'un certain danger. »*
- **Qu'est-ce que la sécurité Web :**
 - « *C'est le fait de protéger le serveur Web ainsi que les applications qui y sont hébergées de tout danger externe. »*
- La présence de **sécurité** implique la **confiance** des utilisateurs et clients,
- **Sécurité** = *Conscience et connaissance + Actions et protections*

Son importance

- Facilité de création de site Web,
- **Complexité** = augmentation des **risques** d'attaque,
- Différentes *attaques* et *actions malveillantes*,
- Réaction en chaîne & atteinte à la réputation du service,
- **Sécurité** = Protection du *propriétaire* + Protection des *clients*

Hacker !

- Les deux face de la force :
 - **White** hat hacker : *bienveillant* 😊
 - **Black** hat hacker : *malveillant* ☹️
- Plusieurs catégories de **Black** hat hacker :
 - Curieux,
 - Soif d'adrénaline,
 - Chasseurs de trophées,
 - Activistes,
 - Professionnels.

- « Un *systeme impénétrable* ou une *sécurité absolue* sont des **mythes** »,
- « Un système est dit *sécurisé* s'il est *protégé* de toutes les *menaces* et *attaques connues* ».

Consciences de la sécurité

- La sécurité n'est pas juste l'affaire des développeurs et des IT,
- Nécessité de mises à jour et d'évaluations régulières,
- Nécessité d'informations et des suivi des changements,
- Documentation, procédures & politique de sécurité,
- La **sécurité** est l'affaire de **tout le monde** !

Moindre accès

- **Principe :** « *consiste à attribuer le moins de privilèges ou d'accès à un utilisateur, juste le minimum. »*,
- Permet d'avoir plus de contrôle sur l'accès aux données,
- Facilite la localisation de la vulnérabilité,
- Évite les réactions en chaîne.

Simplicité

- ↗ complexité \Rightarrow ↗ de risques d'attaques,
- Révision du code régulière pour détection de vulnérabilité,
- Éviter un code trop complexe !
- Désactiver les fonctionnalités inutiles,
- « *Qu'est-ce qui est plus sécuritaire que de verrouiller une porte, c'est de la supprimer complètement.* ».

Ne pas faire confiance

- « *L'erreur est humaine !* »,
- Sécurité = être paranoïaque ... des pirates partout !
- L'utilisateur est généralement la faille : *consciemment* ou *inconsciemment* !
- *Mise à jour régulière* des **autorisations & privilèges**.

S'attendre à tout !

- En terme de sécurité, il faut être **proactif** et non **réactif** !
- Anticiper le pire des scénarios possible,
- Se mettre à la place du hacker.

Plusieurs niveaux

- Inspiré d'une stratégie militaire,
- La défense par plusieurs couches,
- Différents types de protections par couche,
- Ralentir l'attaque pour se donner le temps de réagir.

Sécurité par l'obscurité

- N'exposer que le minimum d'informations,
- \searrow d'informations $\Rightarrow \nearrow$ de sécurité,
- **Exemple** : *versions du serveur, type de SGBD, ...*

Listes blanches & noires

- **Différences :**
 - Blanches : *personnes ou objets autorisés*,
 - Noires : *personnes ou objets non autorisés (bannis)*,
- Appliquer la *restriction par défaut* – Liste blanches.

Points d'entrés & de sorties

- Points d'entrés :
 - URLs,
 - Formulaires,
 - Cookies/Sessions,
 - Lecture de base de données, ...
- Points de sorties :
 - HTML,
 - JavaScript,
 - Écriture de base de données, ...

Sécurité = *Conscience et connaissance* + *Actions et protections*

Contrôle des requêtes

- Différents types :
 - **GET** : URLs & lien HyperText
 - **POST** : Formulaires.
 - **CONNECT, DELETE, PUT, ...**
- **N'autoriser que ce qui est attendus !**
- Formats :
 - « *Content-Type* » : de ce qui est envoyé
 - « *Accept* » : de ce qui est renvoyé ou retourné

Validation des entrées

- Vérification de ce qui est envoyé par l'utilisateur,
- Définir ce qui est attendu : *accepté* ou *refusé*,
- Se prévenir des attaques et des bugs,
- Compatibilité des *données entrées* et celles de la *base de données*,
- **Validations :**
 - *Présence / Longueur*,
 - *Type d'entrée*,
 - *Unicité*,
 - *Format*.

Traitement des données

- Vérifier et traiter les données arrivant au serveur :
 - **Encodage des caractères** : remplacer "<" par "<" en HTML.
 - **Échapper des caractères** : ajouter un \ dans "WHERE userID=\'attack\'"
- Utiliser des méthodes existantes fournies avec le langage,
- Faire les traitements adéquats selon la destination finale des données reçues.

Nommage des variables

- Donner une indication sur l'état des données :
 - **Non traitées** : *raw, unsafe, ...*
 - **Traitées** : *safe, clean, ...*
- **Exemple :**

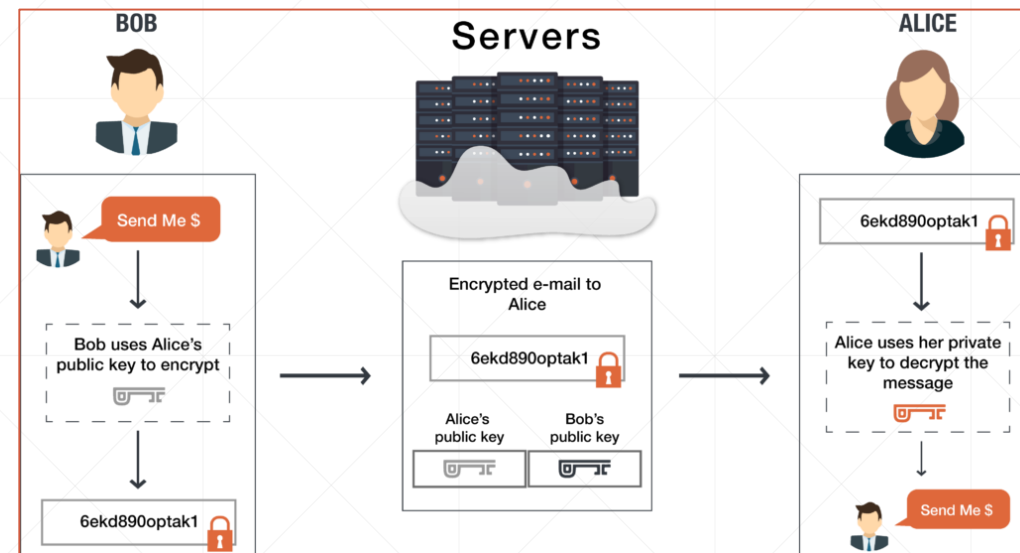
```
$raw_email = $_POST['email'];  
$safe_email = processing($raw_email);
```


Cacher le code

- Restreindre l'accès au code en terme de visibilité,
- Deux types de répertoires :
 - **Public** : *code sans risques et simple*
 - **Privé** : *code complexe et sensible*
- La configuration des accès se fait au niveau du serveur Web :
 - Définir le *dossier racine*,
 - Définir les *autorisations* et les *restrictions*.
- Le fichier **.htaccess** : *autorisation, authentification, ...*

Cacher les accès

- Ne pas exposer les accès dans le code : *identifiants & mots de passes*,
- Mettre les accès dans un fichier séparé (*contrôle de version*),
- Ne pas réutiliser les mots de passes (*serveur, base de données, ...*),
- Encoder (*hacher*) le mot de passe,
- Utiliser un *cryptage* ou *chiffrement* :
 - Clé *publique*,
 - Clé *privée*.



Rester vague dans les messages

- Renvoyer le minimum d'information à l'utilisateur,
- C'est suffisant pour un utilisateur *lambda*,
- Plus d'informations \Rightarrow \nearrow de risques d'attaques.

Garder des *logs*

- C'est des *journaux d'événements* ou *d'actions*,
- **Rôle** : suivi, historique & analyse (*éventuelle attaque*),
- **Informations à garder** :
 - *Date et heure*,
 - *Source (utilisateur ou adresse IP)*,
 - *Action réalisée*,
 - *Cible*,
 - *URL & paramètres*.
- Détecter ce qui s'est passé.

Cross-Site Scripting ou XSS

- **Principe :**

- *Permet au hacker ou pirate d'injecter du code JavaScript dans une page Web,*
- *Ensuite faire en sorte que l'utilisateur ou victime l'exécute.*

- **Exemple :**

Normal

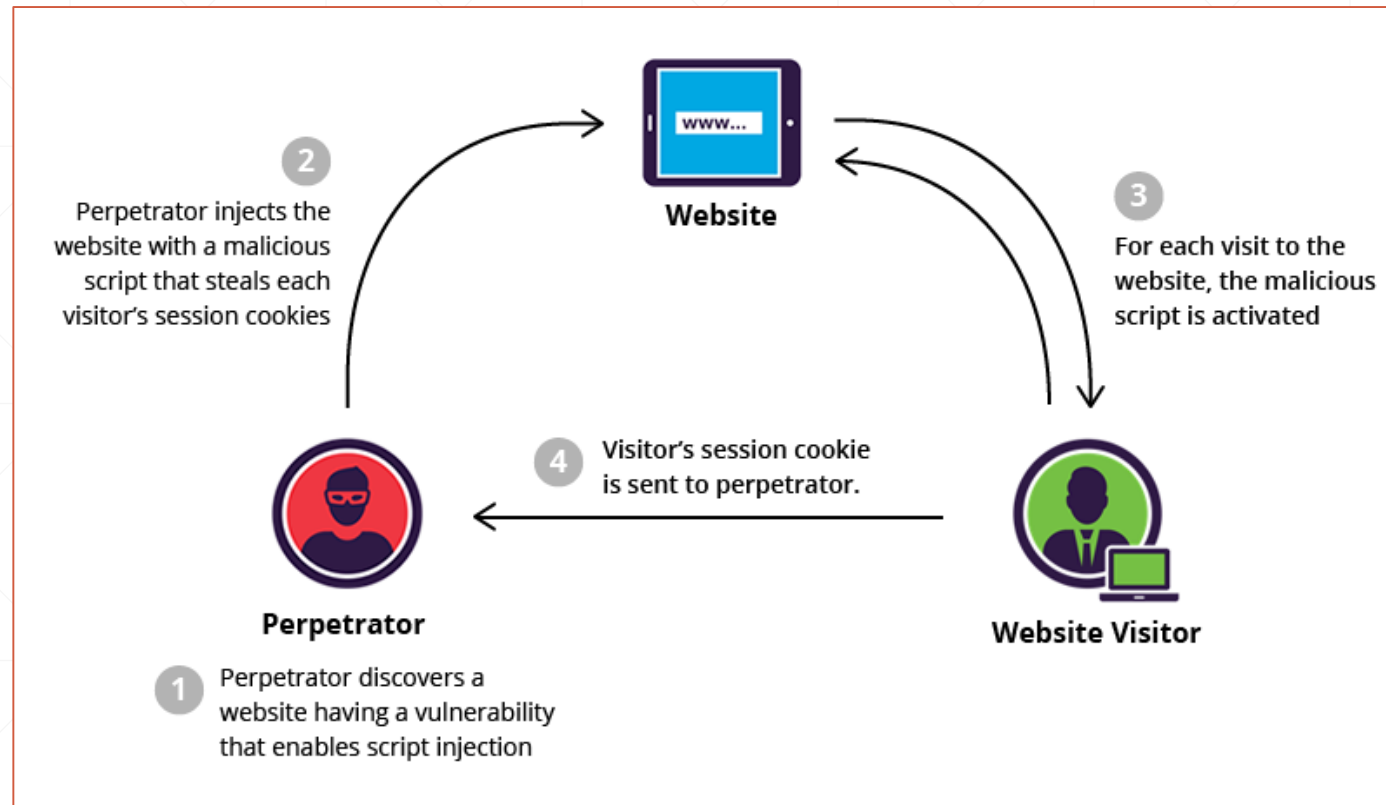
```
GET /update.jsp?email=bouchard@email.ca  
Email: <%= email %>  
Email: email=bouchard@email.ca
```

Attaque

```
GET /update.jsp?email=<script>alert("attaque!");</script>  
Email: <%= email %>  
Email: <script>alert("attaque!");</script>
```

- **Solution :** *Vérifier ce qui est envoyé au serveur.*

Illustration



Cross-Site Request Forgery (CSRF)

- **Principe :**

- *Permet au hacker ou pirate d'envoyer une requête **HTTP** falsifiée,*
- *Faire en sorte de faire exécuter une action par un utilisateur ou victime à son insu.*

- **Exemple :**

Vote en ligne dans le cadre d'un concours

Bouton de vote pour un utilisateur :

<https://site-de-vote.ca/vote?User=1008>

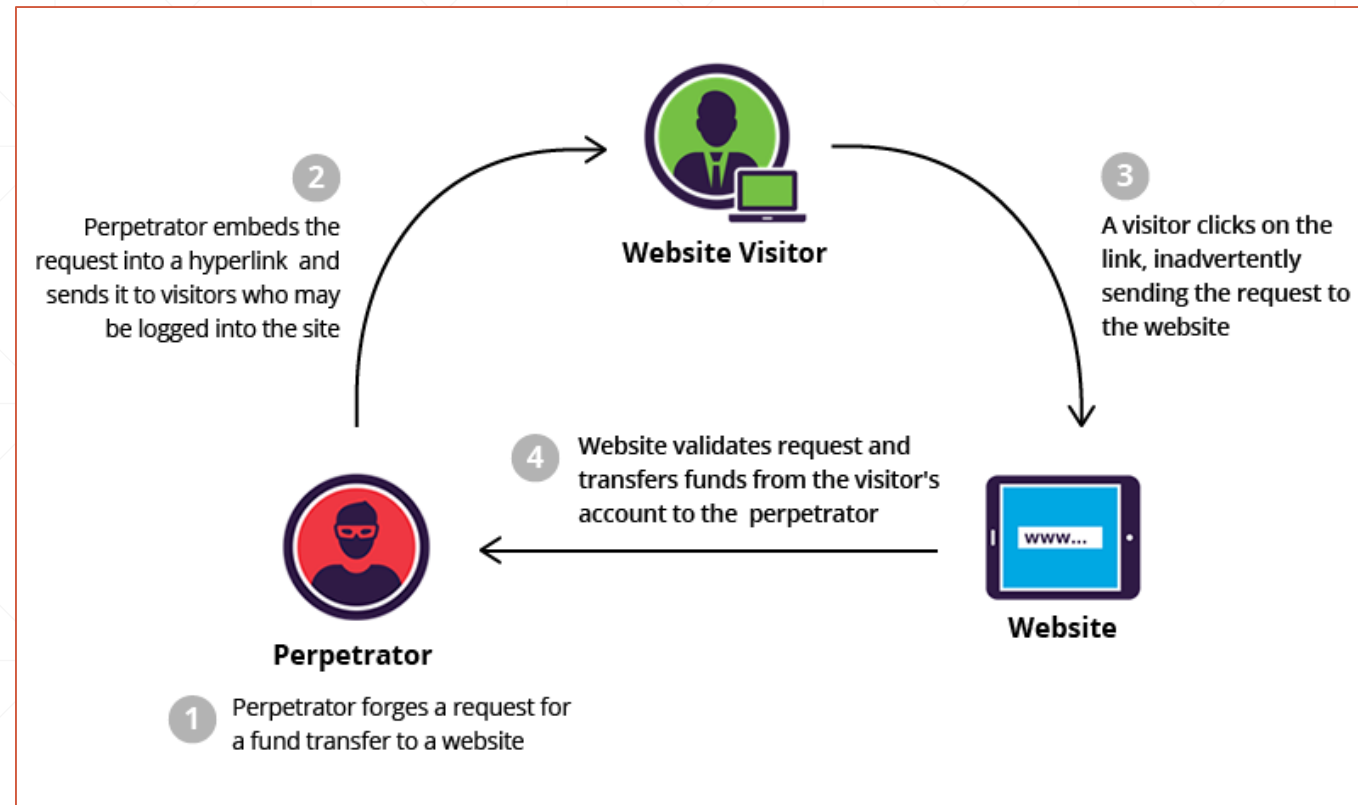
Lien dans un forum

``

- **Solutions :**

- *Ne pas utiliser les requêtes **GET** pour réaliser des opérations,*
- *Utiliser des **tokens** dans des formulaires pour plus de sécurité.*

Illustration



SQL Injection

- **Principe :**
 - *Permet au hacker d'exécuter des requêtes **SQL**,*
 - *Altération, destruction et vol de données.*
- **Exemple 01 :**

Normal

```
SELECT * FROM users WHERE username='${username}' AND password='${password}';  
username = "sbouchard"  
password = "motDePasse"  
SELECT * FROM users WHERE username='sbouchard' AND password='motDePasse';
```

Attaque

```
username = "sbouchard' OR 1 = 1; --"  
password = "vide"
```

```
SELECT * FROM users WHERE username='sbouchard OR 1 = 1; --' AND password='motDePasse';
```

SQL Injection

- **Exemple 02 :**

```
SELECT * FROM articles WHERE title = '${query}'  
query = "q'; DROP TABLE clients; --"
```

- **Solutions :**

- *Définir des privilèges spécifiques à l'utilisateur de l'application,*
- *Traitement des entrées utilisateur (échappement des caractères),*
- *Utiliser des **requêtes préparées** ou **prepared statements**.*

URL manipulation

- **Principe :**

- *Permet au hacker ou pirate d'éditer et de modifier l'**URL**,*
- *Avoir d'avoir de l'information privée ou réaliser des actions non autorisées.*

- **Exemples :**

<http://site-vulnerable.ca?facture=F-7884>

<http://site-vulnerable.ca?UserID=54789651>

<http://site-vulnerable.ca/images/small/paysage.jpg>

- **Solutions :**

- *Rester vague dans les messages d'erreurs (pas de feedbacks),*
- *Éviter que les requêtes GET puissent effectuer des actions.*

Fakes forms

- **Principe :**

- *Permet au hacker ou pirate de récupérer un formulaire et de l'adapter,*
- *Exécuter certaines opération qui ne sont pas forcément autorisées.*

- **Solutions :**

- *Ne pas se baser uniquement sur la validation côté client (JavaScript),*
- *Utiliser des **tokens** ou jetons pour chaque formulaire.*

Cookies visibility & theft

- **Principe :**

- *Permet au hacker ou pirate de voir les informations stockées dans les cookies,*
- *Ils peuvent être volés grâce à une attaque XSS ou sniffé dans le réseau.*

- **Solutions :**

- *Ne pas stocker des données sensibles dans les cookies,*
- *Utiliser le mode **HttpOnly** rendant les cookies inaccessible par le JavaScript,*
- *Utiliser des cookies sécurisées (**HTTPS**),*
- *Définir une date d'expiration,*
- *Préférer utiliser des **sessions serveur** que les **cookies**.*

Session hijacking

- **Fonctionnement :**

- *Les informations sensibles sont sur le serveur,*
- *Tandis qu'un cookie est envoyé au client avec l'ID de la session.*

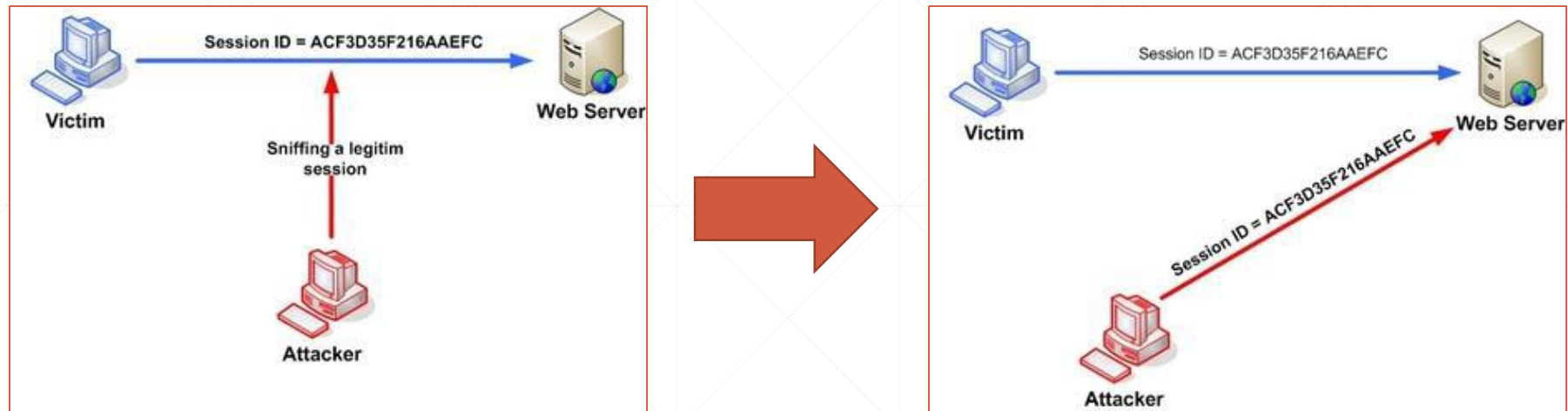
- **Principe :**

- *Permet au hacker ou pirate de détourner et de prendre possession de l'ID de la session,*
- *Puis l'utiliser pour paraître connecté en tant que la victime.*

- **Solutions :**

- *Double vérification avec les entêtes ou bien l'adresse IP,*
- *Utiliser le mode **HttpOnly**,*
- *Regénérer les sessions périodiquement et mettre une date d'expiration,*
- *Utiliser une connexion **SSL**.*

Illustration



Session fixation

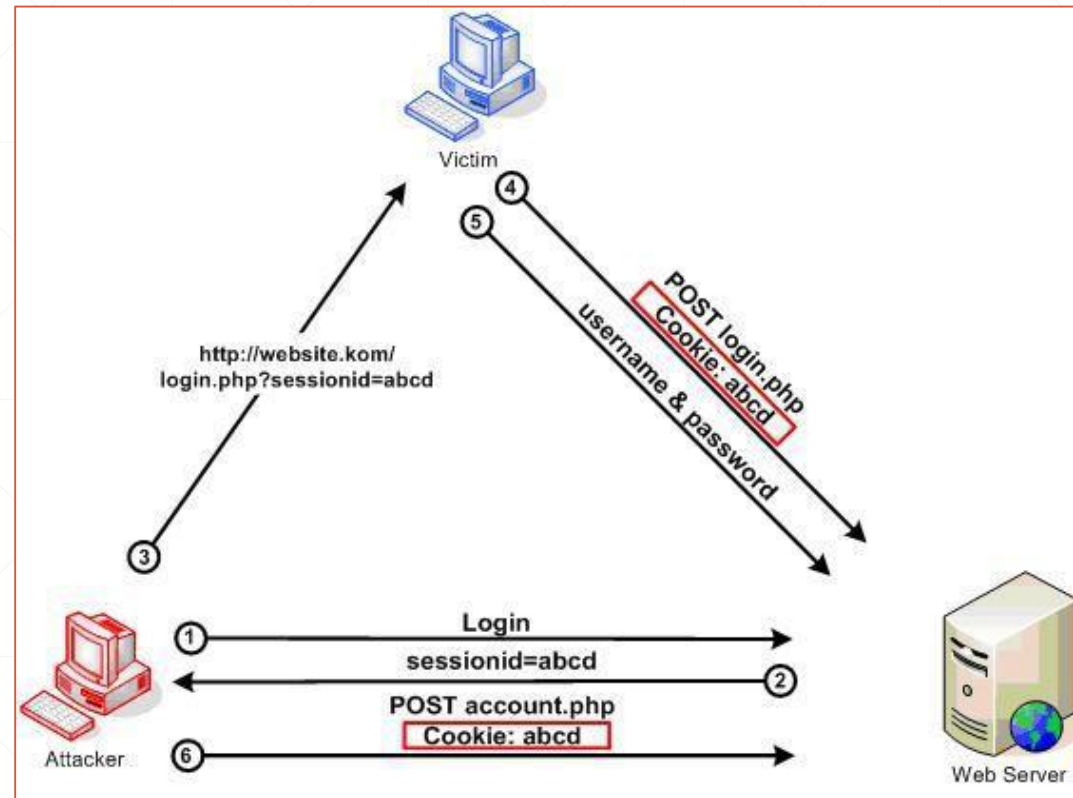
- **Principe :**

- *Permet au hacker ou pirate de donner à la victime son ID de la session,*
- *Puis de pouvoir se connecter au compte de la victime.*

- **Solutions :**

- *Ne pas passer les IDs de session dans des requêtes GET ou POST,*
- *Regénérer les sessions périodiquement et mettre une date d'expiration,*

Illustration



Remote command execution

- **Principe :**

- *Permet au hacker ou pirate d'exécuter des commandes système sur le serveur Web,*
- *Accès à toutes les fonctionnalités du système d'exploitation,*
- *La plus critique et dangereuse des attaques, mais la plus difficile à mettre au point.*

- **Solutions :**

- *Éviter l'utilisation de certaines commandes (`call`, `exec`, `sh`, ...),*
- *Traiter les données entrées par l'utilisateur,*
- *Comprendre le fonctionnement des commandes et ajout de validation.*

File-upload abuse

- **Principe :**

- *Permet au hacker ou pirate d'abuser des fonctionnalités de téléversement,*
- *Téléverser beaucoup de fichier causant une saturation de l'espace,*
- *Téléversement de fichiers malicieux ou malveillants.*

- **Solutions :**

- *Imposer une authentification avant de faire un téléversement,*
- *Limiter la taille maximale du téléversement,*
- *Définir les formats de fichiers autorisés,*
- *Faire attention lors de l'ouverture de fichiers ou imposer une vérification.*

Denial of Service (DoS)

- **Principe :**
 - *Permet au hacker ou pirate de rendre le serveur inaccessible,*
 - *En le submergeant de requêtes,*
 - *En surutilisant l'espace disque, le processeur ou la bande passante.*
- **Distributed DoS ou DDoS,**
- **Facile** à mettre en place et **difficile** à prévenir.
- **Solutions :**
 - *Configurer les pare-feu et les équipements réseaux,*
 - *Configurer une haute disponibilité,*
 - *Il n'y pas de vraie solution.*

Chiffrement des mots de passe

- **Attention :**
 - Ne jamais stocker les mots de passes dans un format lisible,
- **Chiffrement non-réversible :**
 - Stocker le mot de passe chiffré (*algorithme de **hachage***),
 - Réutiliser le même algorithme pour la vérification.
- **Algorithmes de chiffrement :**
 - ~~MD5~~,
 - SHA-1,
 - SHA-2 (SHA-256 ou SHA-512),
 - AES,
 - ...

Salage des mots de passe

- **Menace :**
 - Les *rainbow tables* ou *table arc-en-ciel* sont des tables de haches précalculés,
 - Elles sont utilisées pour trouver le bon mot de passe.
- **Le salage :**
 - **Rôle :** Renforcer la sécurité en ajoutant une chaîne de caractère au mot de passe.
 - **Exemple :** "salage de mot de passe pour {password}"
 - **Méthodes :**
 - Salage *unique* pour chaque utilisateur,
 - Salage *aléatoire*,
 - Il faut également *hacher* le *salage* avant de le stocker dans la base de données.

Chiffrement **Blowfish**

Robustesse du mot de passe

- Ne pas limiter la longueur du mot de passe,
 - Utiliser des caractères spéciaux,
 - Toujours demander une confirmation du mot de passe,
 - Donner un feedback sur la robustesse du mot de passe,
 - Ne pas donner d'indicateur pour le mot de passe,
 - Les questions de sécurité peuvent représenter une faille.
-
- **Astuce :** *Utiliser des générateurs de mots de passes.*

Attaque par force brute

- **Principe :**

- *Tester toutes les combinaisons possibles pour s'authentifier,*
- *Utiliser des dictionnaires pour réduire le temps de recherche.*

- **Temps requis :**

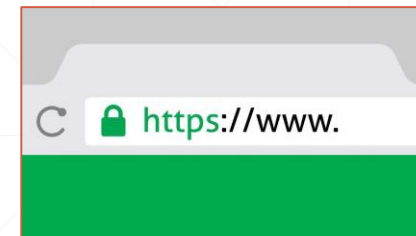
- *$\text{temps total} = \text{nombre de caractères}^{\text{taille du mot de passe}} \times \text{temps pour une tentative}$*
- *Faire de sorte que le temps total soit le plus élevé possible.*

- **Astuce :**

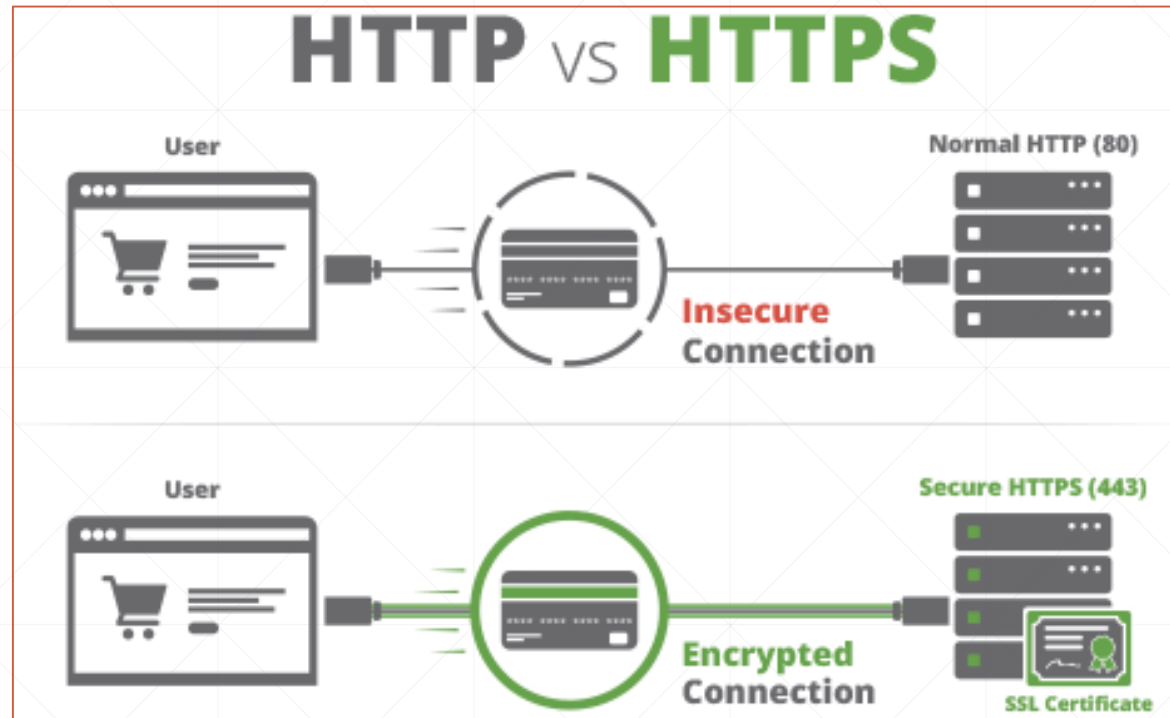
- *Demander aux utilisateurs un mot de passe robuste,*
- *Utiliser un algorithme de chiffrement lent,*
- *Faire des pauses entre chaque tentative,*
- *Utiliser les journaux d'événements.*

Utilisation du SSL

- Ou **Secure Sockets Layer**,
- **Rôle** : *permet de fournir une communication sécurisée.*
- **Fonctionnalités** :
 - *Vérifie et confirme l'authenticité du serveur distant,*
 - *Chiffre les données échangées.*



Utilisation du SSL



Gestion des accès

- Respecter le principe du moindre privilège,
- Être organisé et établir une bonne documentation,
- Faciliter la gestion des accès,
- Définir des rôles et des niveaux de sécurité,
- Restreindre l'accès aux outils de gestion et administration.

Réinitialiser un mot de passe

- **Principe :**

- *Se baser sur une information qui ne peut être connue que par l'utilisateur,*
- *Utilisation d'une adresse email.*

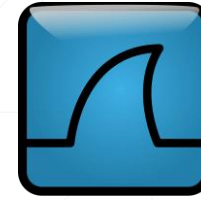
- **Fonctionnement :**

1. *Envoyer un email pour la réinitialisation,*
2. *Toujours demander le nom d'utilisateur,*
3. *Utilisation de **tokens** avec une date d'expiration dans un lien,*
4. *Le compte rester actif et sans changement tant que l'utilisateur ne le l'a pas changé.*

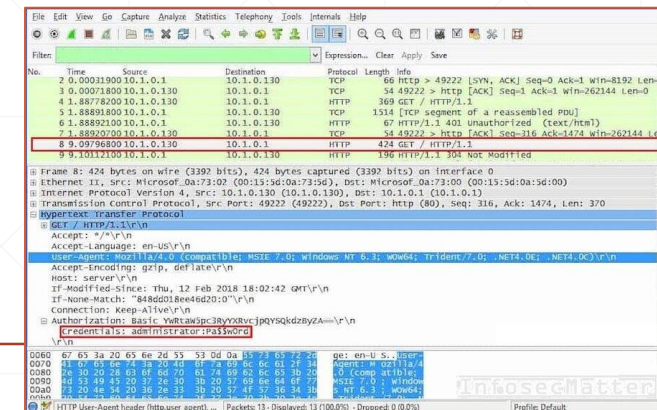
Authentification multimodales

- Requièrre au *minimum deux facteurs* d'authentification,
- **Facteurs :**
 - Ce que l'utilisateur **connait** : *mot de passe*.
 - Ce que l'utilisateur **possède** : *carte d'accès, carte de crédit ou débit*.
 - Ce qui **fait partie** de l'utilisateur : *biométrie (empreintes digitales, voix, visage, ...)*.
- **En pratique :**
 - *Utilisation de l'adresse IP de l'ordinateur,*
 - *Envoyer un email ou sms avec un code de confirmation.*

Exemple d'outil – Wireshark



- **Définition :** « *Wireshark est un sniffer ou analyseur de protocoles réseaux et applicatifs. C'est-à-dire qu'il va **capturer des paquets IP transitant sur le réseau** de manière transparente pour qu'ils soient ensuite **analysés**. Des filtres de capture peuvent être appliqués afin de recueillir des paquets correspondants aux besoins. Wireshark est utilisé par les administrateurs réseau et les experts en sécurité lors de tests d'intrusion, notamment pour des scénarios d'**attaque man-in-the-middle**.* »,
- **Lien de téléchargement :** <https://www.wireshark.org/#download>
- Alternative en **CLI** : <https://nmap.org/>



Questions & Discussion

Bibliographie

1. Dordoigne, J. (2015). *Réseaux informatiques : notions fondamentales : protocoles, architectures, réseaux sans fil, virtualisation, sécurité, IPv6,... (6^{ème} édition)*. Saint-Herblain : Éditions ENI.
2. Ebel, F., Baudru, S., Crocfer, R., Puche, D., Hennecart, J., Lasson, S., & Agé, M. (2006). *Sécurité informatique-Ethical Hacking-Apprendre l'attaque pour mieux se défendre. Editions ENI.*
3. Hoffman, A. (2020). *Web Application Security: Exploitation and Countermeasures for Modern Web Applications. O'Reilly Media, Inc.*