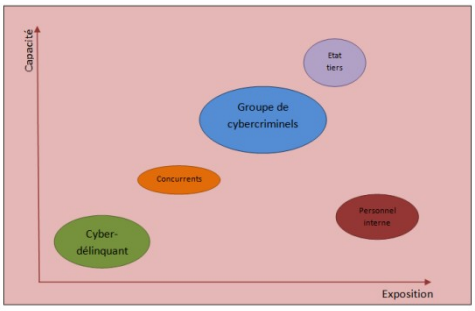
## 1. La sécurité web : où sont les risques ?

✓ **Relier** les propositions entre elles :

| Impacts pour l'individu | | | Enjeux du pirate |
|---|---|---|---|
| Vol d'identité (mot de passe, e mails, civilité, téléphone, n° de comptes bancaires….) | ☐ | ☐ | Espionnage |
| Vol de bande passante/espace de stockage | ☐ | ☐ | Chantage |
| Fuites de données vers la concurrence/ou vers d'autres états. | ☐ | ☐ | Gains financiers (accès à l'information, monétisation, revente) |
| Modification de service ou déni de service | ☐ | ☐ | Utilisation de ressources(puis revente ou mise à disposition en tant que service) |




***Exemple d'une cartographie des principales sources de menaces qui pèsent sur un S.I.***
***Capacité*** *: degré d'expertise et ressources de la source de menaces.*
***Exposition*** *: Opportunités et intérêts de la source de menaces*

Quelques chiffres pour illustrer le marché de la cybercriminalité…

**De 2 à 10 $** le prix moyen de commercialisation des numéros de cartes bancaires en fonction du pays et des plafonds

**5 $** le tarif moyen de location pour 1 heure d'un botnet, système permettant de saturer un site internet

**2.399 $** le prix de commercialisation du malware « Citadel » permettant d'intercepter des numéros de carte

Si dans les années 80/90, les pirates étaient pour beaucoup des bidouilleurs enthousiastes, de nos jours, il s'agit de groupes criminels organisés, professionnels et impliquant de nombreux acteurs à l'origine majoritairement d'attaques malveillantes et réfléchies.

# Les pirates À L'ABORDAGE des hôpitaux

*Mal protégés, nos établissements de santé sont une proie idéale pour les hackers. Les cyberflibustiers pillent nos dossiers médicaux pour les revendre sur le Web. Ils pourraient bientôt saboter à distance le matériel de soins.*

Branle-bas de combat ! L'an dernier, un pirate a obligé les médecins du centre de radiothérapie de Valence, dans la Drôme, à reporter tous leurs rendez-vous. Après avoir pénétré les serveurs de l'établissement de santé, le hacker avait accédé aux données médicales des patients, qui incluaient les doses précises de rayons pour les soigner. Ensuite, il a tout effacé…

Il y a de quoi s'inquiéter, car cet acte de malveillance est loin d'être un cas isolé. *"L'an dernier, déjà, 1500 attaques de ce genre ont été recensées dans nos hôpitaux"*, s'alarme Stéphane Pasquier, fonctionnaire de sécurité des systèmes d'information adjoint pour le ministère des Affaires sociales et de la Santé.

6

## LE CHANTAGE AU RANÇONGICIEL

**DES HACKERS RÉCLAMENT UNE RANÇON POUR LIBÉRER LES FICHIERS DES ORDINATEURS QU'ILS ONT CHIFFRÉS**

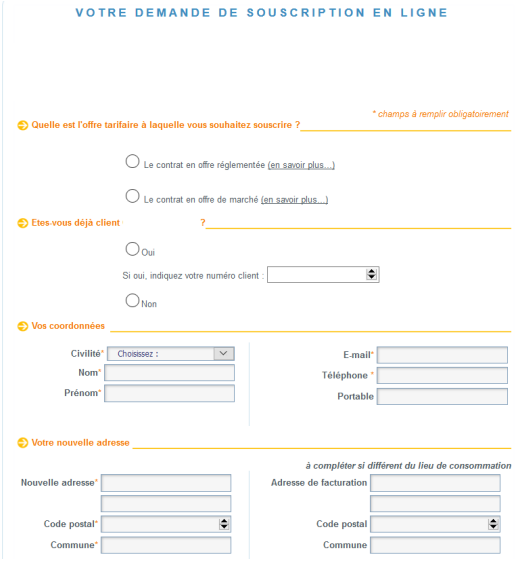
Pour cette dirigeante de PME béarnaise, le cauchemar a démarré en avril. Arrivée au bureau, elle découvre ce message sur son PC : "Votre ordinateur a été attaqué par un virus. Tous vos fichiers ont été cryptés." Pour accéder de nouveau à ses dossiers, elle doit demander assistance au pirate, qui a laissé son adresse mail. "Notre aide n'est pas gratuite, attendez-vous à devoir payer un prix raisonnable pour nos services de décryptage", prévient-il. Le hacker fixe la rançon à 1500 €, payable en bitcoins, la célèbre monnaie virtuelle. La patronne tente de négocier. En vain. "Avec lui, c'était impossible, confiait-elle au quotidien La République des Pyrénées il y a quelques mois. On a échangé une trentaine de courriels, tous en anglais." Chaque jour, l'escroc double ses tarifs, jusqu'à faire plier la chef d'entreprise au terme du cinquième jour. "Dix minutes après avoir payé, il nous a envoyé les codes de déchiffrement." L'activité a repris… avec un trou de près de 10 000 € dans la trésorerie.

**COMMENT L'ÉVITER**

Disposer d'un bon antivirus est une condition nécessaire, mais pas suffisante. Pour préserver ses données, mieux vaut effectuer des sauvegardes régulières sur un disque dur externe. Car payer ne garantit pas de recouvrer tous ses fichiers, comme le prouve l'affaire de l'hôpital américain Kansas Heart, attaqué en début d'année.

● **MÉFIEZ-VOUS AUSSI** DE CETTE ESCROQUERIE AU RANÇONGICIEL CAR LES PIRATES S'ATTAQUENT AUSSI AUX PARTICULIERS. EN DÉBUT D'ANNÉE, UNE VAGUE A CIBLÉ LES ABONNÉS DE L'OPÉRATEUR FREE MOBILE.

## 2. Protocole HTTP: Exemple de vols de données


VOTRE DEMANDE DE SOUSCRIPTION EN LIGNE

* champs à remplir obligatoirement

Quelle est l'offre tarifaire à laquelle vous souhaitez souscrire ? _____

Le contrat en offre réglementée (en savoir plus...)

Le contrat en offre de marché (en savoir plus...)

Etes-vous déjà client ? _____

Oui

Si oui, indiquez votre numéro client :

Non

Vos coordonnées _____

Civilité* Choisissez : | E-mail*

Nom* | Téléphone *

Prénom* | Portable

Votre nouvelle adresse _____

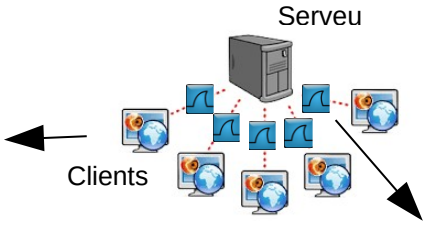
à compléter si différent du lieu de consommation

Nouvelle adresse* | Adresse de facturation

Code postal* | Code postal

Commune* | Commune

Voici un formulaire administratif, dans lequel il vous est demandé des renseignements privés : civilité, nom, prénom, adresse, téléphone, mail…
Un simple logiciel de capture de trames (Wireshark) permet de récupérer toutes les informations.

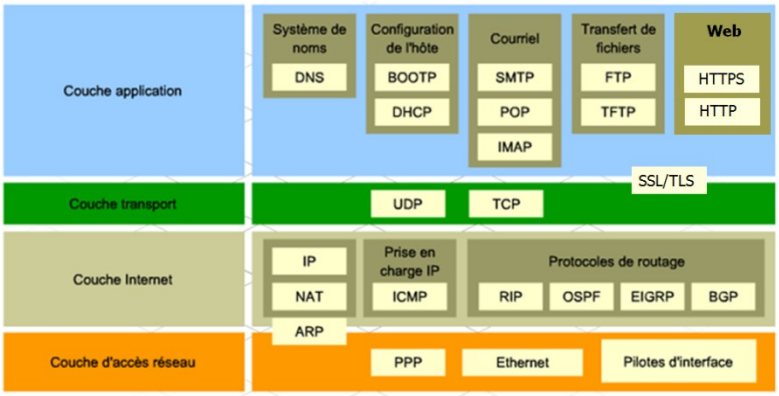



```
HTTP      1036 POST /particuliers/formulaire_souscription_action.php HTTP/1.1
```

✓ Démarrez le logiciel Wireshark. Ouvrez le fichier post.pcapng. **Sélectionnez et ouvrez la trame 2** qui correspond à l'envoi du formulaire.

```
HTTP      1036 POST /particuliers/formulaire_souscription_action.php HTTP/1.1
```

**Rappel :** COUCHES du modèle TCP/IP : protocole de communication client/serveur




✓ Trouvez dans la trame sélectionnée les deux informations ci-dessous, dans quelle couche de la trame TCP/IP se trouvent-elles ?

```
Form item: "e-mail" = "xbeen@gmail.com"
Form item: "Téléphone" = "0601020304"
```

✓ Quel est le protocole de transmission du formulaire ?

✓ Pouvez-vous voir en clair toutes les informations déclarées dans le formulaire ?

✓ Quelle méthode a-t on utilisée pour transmettre ce formulaire ?

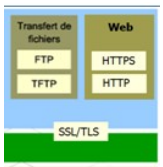
✓ Voici l'URL du formulaire, que constatez-vous dans son adresse ?

www.gazdebordeaux.fr/particuliers/formulaire_souscription.php

- ✓ A qui peuvent-être vendues ces données ?

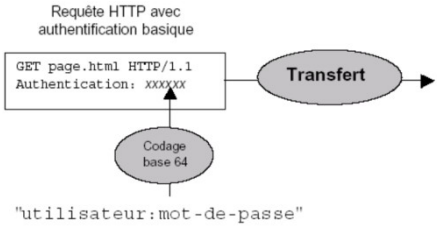
- ✓ La demande de connexion par login et mot de passe est-elle une preuve de sécurité d'un site ?

**Conclusion :**
Lorsque vous vous connectez à un site, vérifiez :

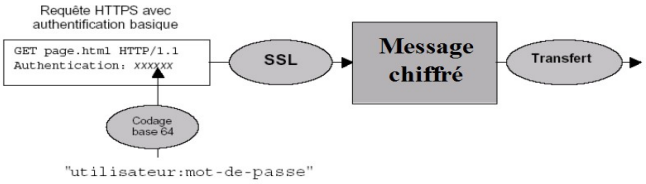



- **La présence du https** : il assure le chiffrage des données pendant le transport par l'usage des protocoles **SSL** ou **TLS**.

⚠ HTTP : non sécurisé

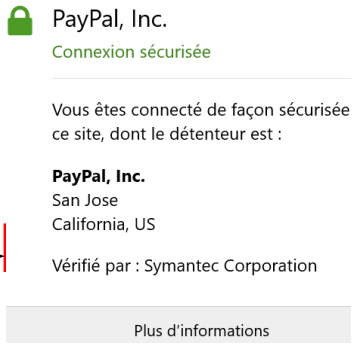



HTTPS : si le site n'est pas compromis : sécurisé




**Remarques :** les sites réalisés avec WordPress sont par défaut proposés en http, les serveurs multi-médias, les serveurs de stockage en réseau, les caméras IP personnelles, les boxs, point d'accès wifi, les serveurs domestiques (pompes à chaleur, tv,…) sont souvent accessibles en http pour leur administration : il faut donc savoir que vos logins administrateurs n'y sont pas correctement protégés et que, par conséquent, le piratage de vos données est possible par simple capture de trames.

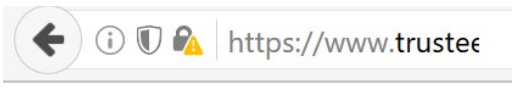
- **La validité du certificat** : il valide l'identité du site, et permet de s'assurer de l'intégrité de la clé publique de chiffrement (présence du cadenas vert dans l'URL).


Nom de l'autorité de certification, le tiers de confiance qui atteste de la sécurité de la connexion :

## 3. Sites peu sécurisés : le contenu peut-être malicieux

En cliquant sur le cadenas dans l'URL, on accède à des informations de sécurité, ici le site n'est pas totalement sécurisé.





⚠ Ne jamais désactiver la protection ou donner des permissions sans connaissance des risques.

**La règle de validité du certificat :**

- ✓ Compléter ce tableau à l'aide de ce site : https://support.mozilla.org/fr/kb/comment-savoir-si-ma-connexion-est-securisee

| Symbole affiché |  |  |  |  |
|---|---|---|---|---|
| Type de connexion Sécurisée/non sécurisée | | | | |
| Risques | | | | |

## 4. Certificats, protocoles sécurisés et hachage :

**Un certificat** est un **fichier électronique** qui comprend notamment :

La **clé publique** d'un individu (ou d'une entité ou d'un nom de domaine) qui permet le chiffrage des données ;

Les **détails de cet individu** (ou de cette entité) : nom, prénom, nom de domaine, etc. ;

La **signature par un tiers de confiance**, chargé de garantir que le propriétaire de la clé publique a été vérifié et – par conséquent – l'authenticité de la clé publique vis-à-vis de son propriétaire. La signature porte sur l'identité du détenteur et la clé publique afin d'assurer l'intégrité de l'ensemble ;

D'autres informations telles que l'usage de la clé, les dates de validité, des informations concernant la révocation, etc.

Le **tiers de confiance** est une autorité de certification, en charge de :

**Vérifier l'identité** de la personne demandant à créer le certificat ;

**Créer le certificat** après vérification, **puis le signer** (avec la clé privée de l'autorité de certification) ;

**Tenir à jour une liste des certificats qui ont été révoqués** (par exemple si la clé a été compromise).

- ✓ Le certificat ci-dessous est celui d'un site sécurisé, lequel ? Quel protocole de chiffrement est utilisé pour la communication client/serveur ?

**Ce certificat a été vérifié pour les utilisations suivantes :**

Certificat client SSL

Certificat serveur SSL

Détenteur de la clé publique

**Émis pour**

| | |
|---|---|
| Nom commun (CN) | mail.google.com |
| Organisation (O) | Google Inc |
| Unité d'organisation (OU) | <Ne fait pas partie du certificat> |
| Numéro de série | 57:A6:3D:EA:7E:22:B0:71 |

Autorité de certification

**Émis par**

| | |
|---|---|
| Nom commun (CN) | Google Internet Authority G2 |
| Organisation (O) | Google Inc |
| Unité d'organisation (OU) | <Ne fait pas partie du certificat> |

Dates de validité du certificat

**Période de validité**

| | |
|---|---|
| Débute le | mercredi 1 juin 2016 |
| Expire le | mercredi 24 août 2016 |

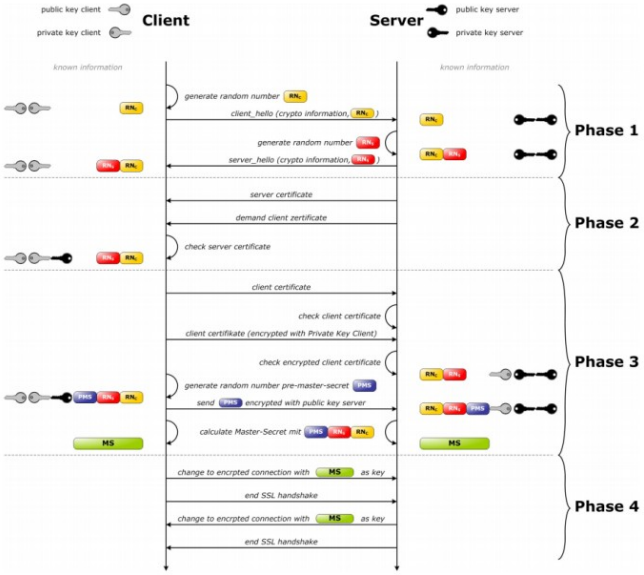
**Empreintes numériques**

| | |
|---|---|
| Empreinte numérique SHA-256 | C0:92:74:95:E7:59:4F:37:FE:D2:35:0E:1E:08:EF:2C:23:2D:C3:04:23:59:2D:4D:8E:ED:E2:9B:0B:36:B9:26 |
| Empreinte numérique SHA1 | 8D:2F:EC:59:B5:28:E4:30:A9:F6:5E:75:DA:CF:4B:58:6E:55:F9:D3 |

## Les protocoles sécurisés :

**SSL** :***Secure Sockets Layer*** est un protocole de sécurisation des échanges sur Internet, devenu ***Transport Layer Security*** (TLS) en 2001 ; Dans la majorité des cas, l'utilisateur authentifie le serveur TLS sur lequel il se connecte. Cette authentification est réalisée par l'utilisation d'un **certificat numérique** délivré par une **autorité de certification** (AC). Des applications web peuvent utiliser l'authentification du poste client en exploitant TLS. Il est alors possible d'offrir une authentification mutuelle entre le client et le serveur.
La sécurité est réalisée d'une part par un **chiffrement asymétrique**, comme le **chiffrement RSA**, qui permet, après authentification de la clef publique du serveur, la constitution d'un secret partagé entre le client et le serveur, d'autre part par un **chiffrement symétrique** (beaucoup plus rapide que les chiffrements asymétriques), comme l'**AES**, qui est utilisé dans la phase d'échange de données, les clefs de chiffrement symétrique étant calculées à partir du secret partagé.

- ✓ L'image ci-contre(TLS.pdf) montre les trames échangées entre client et serveur dans le protocole TLS. Entourez en rouge la trame de communication de la clé publique du serveur, en vert celle du client.
- ✓ Quel type de chiffrement a lieu dans les phases 2 et 3 ?
- ✓ Ci-dessous vous retrouvez les trames correspondant à un login sur Toutatice, quelle est l'IP du client et celle du serveur ? Retrouver dans ces trames la phase 1 du protocole TLS (pour cela aidez vous du schéma ci-dessus TLS.pdf).




| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 4 | 0.001573 | 192.168.11.208 | 195.221.67.112 | HTTP | 692 | GET /portail/auth/MonEspace HTTP/1.1 |
| 5 | 0.017713 | 195.221.67.112 | 192.168.11.208 | HTTP | 493 | HTTP/1.1 302 D\351plac\351 Temporairement |
| 6 | 0.019001 | 192.168.11.208 | 195.221.67.112 | TCP | 66 | 58012 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 7 | 0.023435 | 195.221.67.112 | 192.168.11.208 | TCP | 62 | 443 → 58012 [SYN, ACK] Seq=0 Ack=1 Win=3900 Len=0 MSS=1300 SACK_PERM=1 |
| 8 | 0.023493 | 192.168.11.208 | 195.221.67.112 | TCP | 54 | 58012 → 443 [ACK] Seq=1 Ack=1 Win=65000 Len=0 |
| 9 | 0.023644 | 192.168.11.208 | 195.221.67.112 | TLSv1.2 | 284 | Client Hello |
| 10 | 0.028998 | 195.221.67.112 | 192.168.11.208 | TCP | 60 | 443 → 58012 [ACK] Seq=1 Ack=231 Win=4130 Len=0 |
| 11 | 0.029667 | 195.221.67.112 | 192.168.11.208 | TLSv1.2 | 150 | Server Hello, Change Cipher Spec |
| 12 | 0.029814 | 195.221.67.112 | 192.168.11.208 | TLSv1.2 | 123 | Encrypted Handshake Message |
| 13 | 0.029848 | 192.168.11.208 | 195.221.67.112 | TCP | 54 | 58012 → 443 [ACK] Seq=231 Ack=166 Win=64835 Len=0 |
| 14 | 0.029997 | 192.168.11.208 | 195.221.67.112 | TLSv1.2 | 129 | Change Cipher Spec, Encrypted Handshake Message |
| 15 | 0.031112 | 192.168.11.208 | 195.221.67.112 | TLSv1.2 | 907 | Application Data |
| 16 | 0.034488 | 195.221.67.112 | 192.168.11.208 | TCP | 60 | 443 → 58012 [ACK] Seq=166 Ack=306 Win=4205 Len=0 |
| 17 | 0.035348 | 195.221.67.112 | 192.168.11.208 | TCP | 60 | [TCP Dup ACK 16#1] 443 → 58012 [ACK] Seq=166 Ack=306 Win=4205 Len=0 |
| 18 | 0.035880 | 195.221.67.112 | 192.168.11.208 | TCP | 60 | 443 → 58012 [ACK] Seq=166 Ack=1159 Win=5058 Len=0 |

| N° de trames | Phase |
|---|---|
| | 1 |

- ✓ Ouvrez le fichier certif.crt qui contient les détails de ce certificat, donnez la taille de la clé publique et le type d'algorithme (paramètre de la clé), qui a permis d'obtenir cette clé permettant de chiffrer la communication client/serveur. Est-elle visible ?

**Ce certificat a été vérifié pour les utilisations suivantes :**

Certificat client SSL

Certificat serveur SSL

Détenteur de la clé publique

**Émis pour**

| | |
|---|---|
| Nom commun (CN) | mail.google.com |
| Organisation (O) | Google Inc |
| Unité d'organisation (OU) | <Ne fait pas partie du certificat> |
| Numéro de série | 57:A6:3D:EA:7E:22:B0:71 |

Autorité de certification

**Émis par**

| | |
|---|---|
| Nom commun (CN) | Google Internet Authority G2 |
| Organisation (O) | Google Inc |
| Unité d'organisation (OU) | <Ne fait pas partie du certificat> |

Dates de validité du certificat

**Période de validité**

| | |
|---|---|
| Débute le | mercredi 1 juin 2016 |
| Expire le | mercredi 24 août 2016 |

**Empreintes numériques**

| | |
|---|---|
| Empreinte numérique SHA-256 | C0:92:74:95:E7:59:4F:37:FE:D2:35:0E:1E:08:EF:2C:23:2D:C3:04:23:59:2D:4D:8E:ED:E2:9B:0B:36:B9:26 |
| Empreinte numérique SHA1 | 8D:2F:EC:59:B5:28:E4:30:A9:F6:5E:75:DA:CF:4B:58:6E:55:F9:D3 |

**Une fonction de hachage** (hash function), est une fonction particulière qui, à partir d'une donnée fournie en entrée, calcule une empreinte servant à identifier rapidement, bien qu'incomplètement, la donnée initiale : vérification de l'intégrité et l'authentification des données reçue avec les données transmises.

- ✓ Dans le certificat précédent quelles fonctions de hachage ont été utilisées?

- ✓ Ce hash permet de vérifier l'intégrité du certificat, vous pouvez faire de même sur des fichiers que vous téléchargez, exemple : téléchargez le cd d'installation de debian 9.2.1 pour amd 64 netsinst sur ce lien : https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/

| Name | Last modified | Size |
|---|---|---|
| Parent Directory | | - |
| MD5SUMS | 2017-10-14 16:58 | 201 |
| MD5SUMS.sign | 2017-10-14 17:49 | 833 |
| SHA1SUMS | 2017-10-14 16:58 | 225 |
| SHA1SUMS.sign | 2017-10-14 17:49 | 833 |
| SHA256SUMS | 2017-10-14 16:58 | 297 |
| SHA256SUMS.sign | 2017-10-14 17:49 | 833 |
| SHA512SUMS | 2017-10-14 16:58 | 489 |
| SHA512SUMS.sign | 2017-10-14 17:49 | 833 |
| debian-9.2.1-amd64-netinst.iso | 2017-10-13 15:10 | 290M |
| debian-9.2.1-amd64-xfce-CD-1.iso | 2017-10-13 15:11 | 648M |
| debian-mac-9.2.1-amd64-netinst.iso | 2017-10-13 15:10 | 293M |

- ✓ Ouvrez le fichier MD5SUMS pour connaître le hash MD5 du fichier téléchargé

```
5c583fd40360fd039b3ac98387b77dbb  debian-9.2.1-amd64-netinst.iso
9bc802a4a37464b7fb234585ee672420  debian-9.2.1-amd64-xfce-CD-1.iso
9ffdb574e46b514fcc57b8599c68b7ac  debian-mac-9.2.1-amd64-netinst.iso
```

si le fichier télécharger est bien celui de départ alors les hashs doivent correspondre :

```
clebris@c002-02:~$ md5sum Téléchargements/debian-9.2.1-amd64-netinst.iso
5c583fd40360fd039b3ac98387b77dbb  Téléchargements/debian-9.2.1-amd64-netinst.iso
clebris@c002-02:~$
```

- ✓ Faites la manipulation, pour vérifier.

- ✓ Dans l'émulateur php suivant http://phptester.net/ testez les mots suivants avec l'algorithme de hashage SHA1: **Pot de colle, Pot, Pot de molle**

***Hacher un mot de passer avec SHA1***

**Exemple de hash avec sha1()**

```php
<?php
  $sha1 = sha1('m0tD3P4ss3');
  echo $sha1;
?>
```

Que remarquez-vous sur la taille du hash, le type de caractères ? Pouvez-vous remonter à la source à partir du mot haché ?

**Problèmes :**

- Les fonctions de hachage, comme MD5, SHA-1 sont sensibles aux collisions : une faille qui permet de trouver deux hashs identiques pour deux fichiers différents. SHA-256 remplace progressivement SHA-1, MD5 faillible est de moins en moins utilisé.

  ✓ Visionnez la vidéo collisionmd5.avi, comparez les hashs obtenus pour les deux fichiers hello et erase? font-ils des actions différentes ? Dans quel but, cette faille peut-être utilisée ?

- La fonction de hachage permet de chiffrer les chaînes efficacement mais restent « crackables » ! Elle est aussi utiliser pour chiffrer des mots de passe, on ne stocke alors que les hashs de ces mot de passe. Il existe sur Internet des dictionnaires capables de retourner la chaîne en clair d'un md5(), d'un sha1() ou d'un autre algorithme standard de hash. Nul besoin de rappeler que les mots de passe classiques du type root, superadmin, toto... existent dans ces dictionnaires.

**Solution** :
Dans cet exemple, nous allons finalement hacher avec l'algorithme MD5 la chaine suivante :
*prisonm0tD3P4ss3break*

✓ Dans l'émulateur php suivant http://phptester.net/ , *testez cette technique :*

**Hash de mot de passe avec des salts**

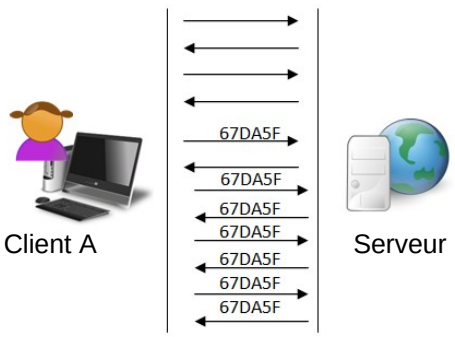
```
<?php
  // Déclaration des constantes
  define('PREFIX_SALT', 'prison');
  define('SUFFIX_SALT', 'break');
  echo md5(PREFIX_SALT.'m0tD3P4ss3'.SUFFIX_SALT);
?>
```

Graines secrètes


Cette technique permet de ne pas pouvoir récupérer facilement le mot de passe d'origine en clair. La sécurité du mot de passe réside alors dans la complexité et la confidentialité des clés choisies.

## 5. Récupération de cookies : vol d'identifiants

Le fonctionnement habituel d'une connexion sur un site web nécessitant une authentification (site marchand, site bancaire, serveur de notes…) est le suivant :

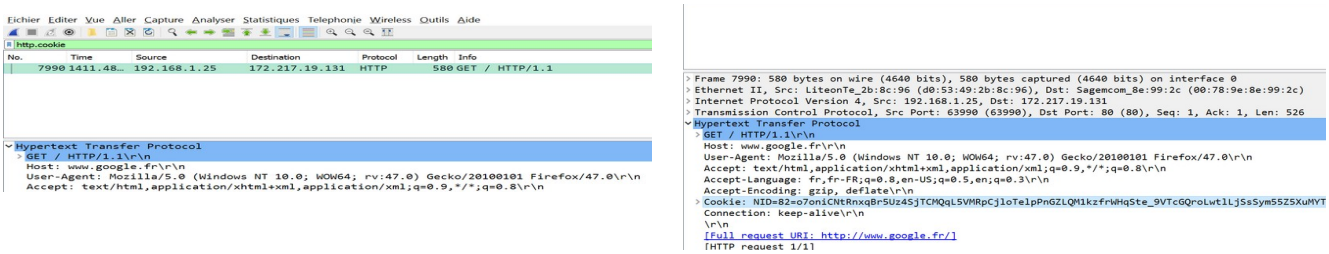



1. Phase d'authentification (via un mot de passe en général) ;
2. Phase de génération du cookie d'identification*. L'utilisateur est maintenant connecté « à son compte »
3. Phase de « navigation » : le cookie est inclus dans tous les échanges afin que le serveur puisse identifier la connexion de l'utilisateur ;
4. Phase de déconnexion : la session de l'utilisateur est maintenant clôturée. Le cookie est invalidé.

**Un cookie** d'identification est en fait une **chaîne de caractères aléatoire et unique**, suffisamment longue pour qu'elle ne puisse pas être générée deux fois par erreur.

Exemple de récupération de cookies :

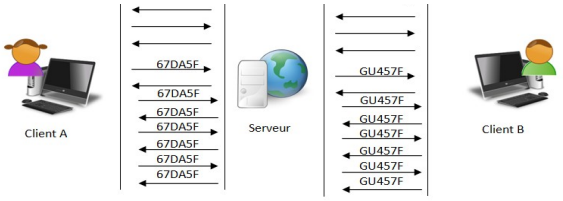
- ✓ Ouvrez wireshark puis sur Firefox tapez : http://www.wikipedia.fr/index.php, stoppez la récupération de trames. Filtrez les trames récupérées avec : http.cookie et retrouvez votre cookie.

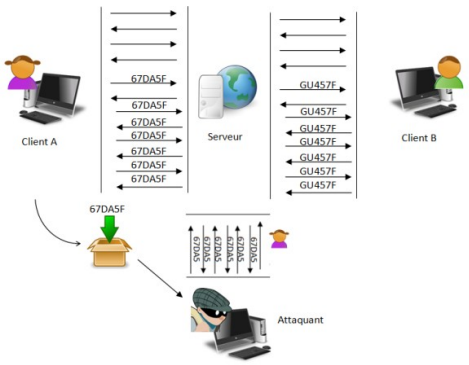



- ✓ Effectuez la même récupération de trames pour https://fr.wikipedia.org/wiki/Wiki, que constatez-vous?

**Conclusion** : A tout moment d'une connexion, chaque utilisateur du site web possède donc son propre cookie, unique à lui. Le serveur est donc en mesure d'identifier à qui appartient chaque connexion, et donc d'afficher les pages qui lui sont propre.

Mais que se passe-t-il si un attaquant arrive à dérober le cookie d'un utilisateur et se connecte au même serveur ?




Il peut se faire passer pour l'utilisateur dont il a dérobé le cookie au près du serveur applicatif ! Il usurpe donc l'identité de la victime et accède à son compte.

Moyens de protection :

- L'utilisateur doit s'assurer que le site auquel il est connecté utilise du HTTPS (le cookie est donc chiffré pendant le transport).
- L'utilisateur doit sécuriser son système d'exploitation et ses logiciels correctement (services inutiles désactivés, installation des mises à jours de sécurité, anti-virus, etc.).
- L'utilisateur doit être sensibilisé aux méthodes d'ingénierie sociale (phishing, spam, etc.) afin de « ne pas tomber dans le panneau ».
- L'exploitant du serveur doit suivre les bonnes pratiques de sécurisation et du maintien en condition de sécurité du serveur, ainsi que les bonnes pratiques de développement applicatif.

- ✓ A quelle peine de prison et quelle amande s'expose toute personne commettant un vol de données d'identification (ici cookies)? Cf :https://www.cnil.fr/fr/les-sanctions-penales

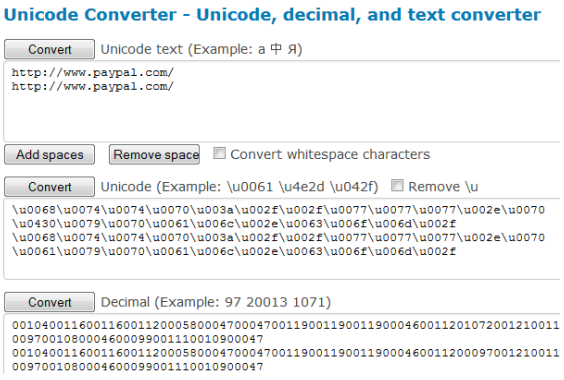
## 6. Le phishing : usurpation d'identité d'un tiers

**L'hameçonnage, *phishing* ou filoutage** est une technique utilisée par des fraudeurs pour obtenir des renseignements personnels dans le but de perpétrer une **usurpation d'identité**. La technique consiste à faire croire à la victime qu'elle s'adresse à une personne de confiance — banque, administration, etc. — afin de lui soutirer des renseignements personnels : mot de passe, numéro de carte de crédit, date de naissance, etc. C'est une forme d'attaque informatique reposant sur l'ingénierie sociale. Elle peut se faire par courrier électronique, par des sites web falsifiés ou autres moyens électroniques.

Exemple : pour masquer le nom de domaine réel consiste à utiliser des caractères bien choisis parmi les dizaines de milliers de caractères du répertoire Unicode. En effet, certains caractères spéciaux ont l'apparence des caractères de l'alphabet latin.

✓ Testez les liens soulignés suivants : http://www.paypal.com/ et http://www.paypal.com/ que constatez-vous ?

✓ Collez ces deux liens dans l'onglet convert du site https://www.branah.com/unicode-converter et retrouvez les codes des caractères modifiés puis dans la table Unicode http://unicode-table.com/fr/#cyrillic retrouvez les caractères falsifiés:




Une contre-mesure à cette attaque est **d'écrire manuellement les URL**, ne pas cliquer sur un lien proposé. Autre contre mesure : ne pas permettre **l'affichage des caractères hors du répertoire ASCII**, qui ne contient que les lettres de A à Z, les chiffres et de la ponctuation. Cette dernière contre-mesure est cependant difficilement compatible avec l'internationalisation des noms de domaine, qui requiert le jeu de caractères Unicode.

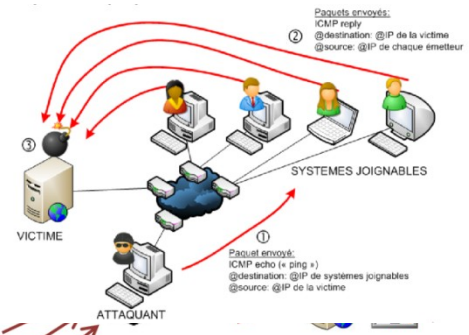
## 7. Vol de bande passante : Attaque DDos (Distributed Denial Of Services) :

Un déni de service distribué consiste à envoyer des milliers, des dizaines de milliers, des centaines de milliers de requêtes simultanément. Si l'on limite la réflexion aux sites Web, il suffit, en général, de faire 10 à 50 000 connexions simultanées pour mettre à genou un serveur et/ou la connexion Internet des serveurs.

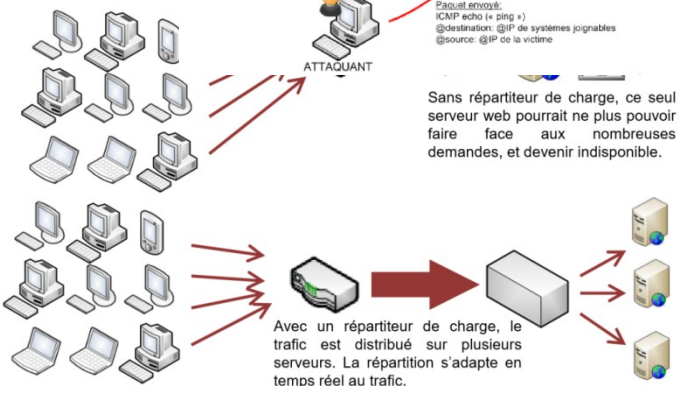
- usurpation d'adresse IP ;
- réflexion de trafic en ayant recours à des systèmes tiers « innocents ».

**Séquences de l'attaque**

① Un attaquant envoie des paquets PING à des systèmes tiers joignables en indiquant l'@IP de la future victime comme @IP source ;
② Chaque système pense ainsi recevoir un PING de la part d'un système distant, et chacun va répondre à ce PING ;
③ Avec suffisamment de ressources, l'attaquant sera en mesure de faire générer suffisamment de trafic pour affecter les performances de la victime.




Pour lutter, contre ce type d'attaque, le répartiteur ou « Load-balancer » en anglais est un équipement rencontré sur les grosses infrastructures où les serveurs doivent faire face à de très fortes bandes passantes et charges élevées de trafic: il est chargé de répartir/distribuer la charge réseau en fonction des caractéristiques de celui-ci et de la disponibilité des serveurs.




**Schémas et textes extrait du cours**