PHP

CM4-1 : Failles Web

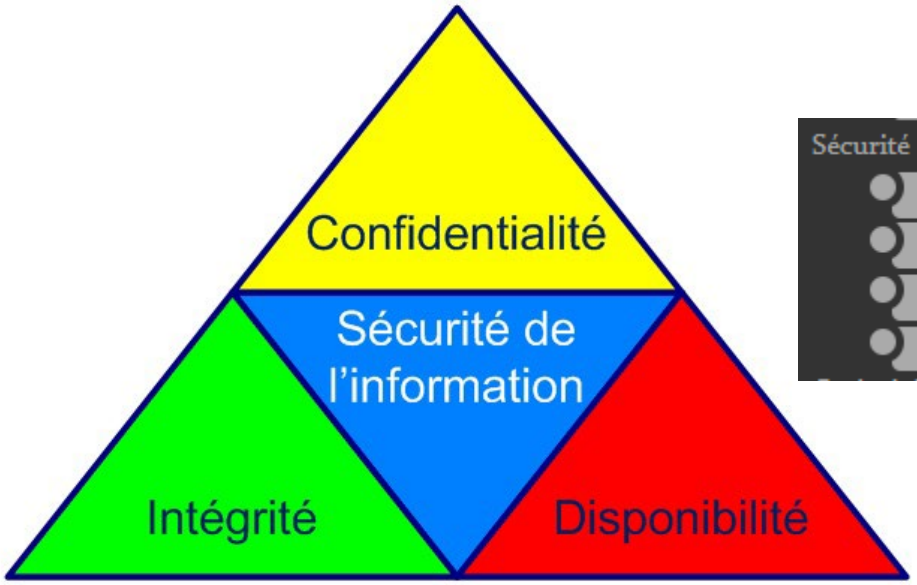
Mickaël Martin Nevot

V5.0.1

PHP

Sécurité Web

- **Disponibilité** : maintenir le bon fonctionnement du système
- **Intégrité** : garantir que les données sont celles voulues
- **Confidentialité** : information inintelligible en dehors des acteurs de la transaction



Ressources dans le recueil des cours

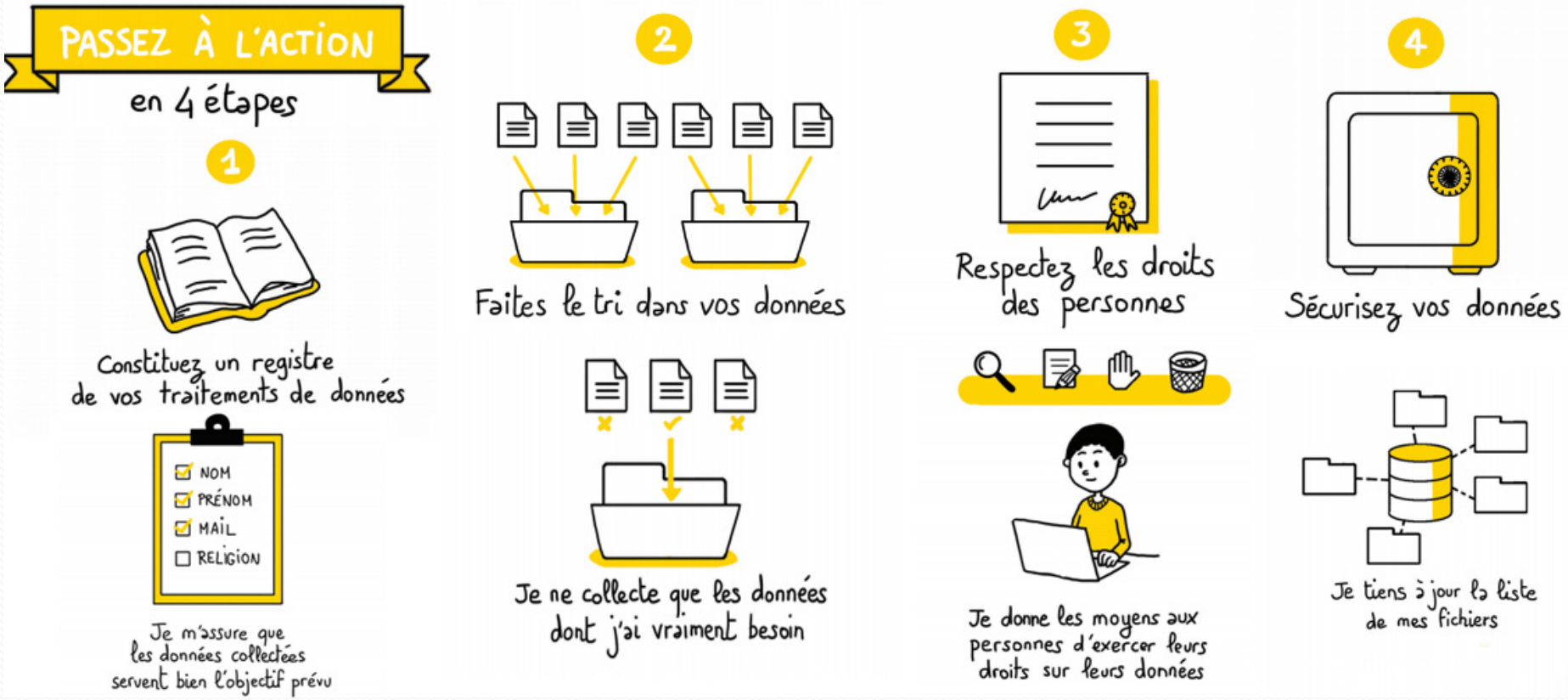
Sécurité
- Open Web Application Security Project (OWASP) : Top Ten
- Open Web Application Security Project (OWASP) : Cheat Sheet
- CVE (MITRE)
- Guide de sélection d'algorithmes cryptographiques (ANSSI)

RGPD

Depuis 2016

Par l'Union européenne, mais repris dans le monde entier

- Règlement général sur la protection des données



OWASP

OWASP
Open Web Application Security Project

1. **Broken Access Control** : droits d'accès
2. **Cryptographic Failures** : usurpations d'identité, CB, etc.
3. **Injection** : SQL, JavaScript, etc.
4. **Insecure Design** :
5. **Security Misconfiguration** : conf. serv. Web / *frameworks*
6. **Vulnerable / Outdated Components** : comp. tiers
7. **Ident. / Auth. Failures** : authentification, session, etc.
8. **Software / Data Integrity Failures** : intégrité des données
9. **Security Loggin / Monitoring Failures** : surveillance
10. **Server-Side Request Forgery (SSRF)** : DoS, RCE

Failles Web : analyse de site

- Partie visible :
  - **Statique** ou **dynamique** ? (y a-t-il une **URL *rewriting*** ?)
  - Les variables utilisées ? (méthode **GET** ou **POST**)
  - Les champs des formulaires ? Des **champs cachés** ?
  - Existence de ***cookies*** ?
  - Dossiers d'images, vidéos, etc. ?
  - Existence de **JavaScript** ?
  - Existence de **répertoires accessibles** ?
- Partie cachée :
  - Intercepter les échanges entre navigateur et serveur Web
  - Envoi massif de requêtes avec un ***fuzzer***

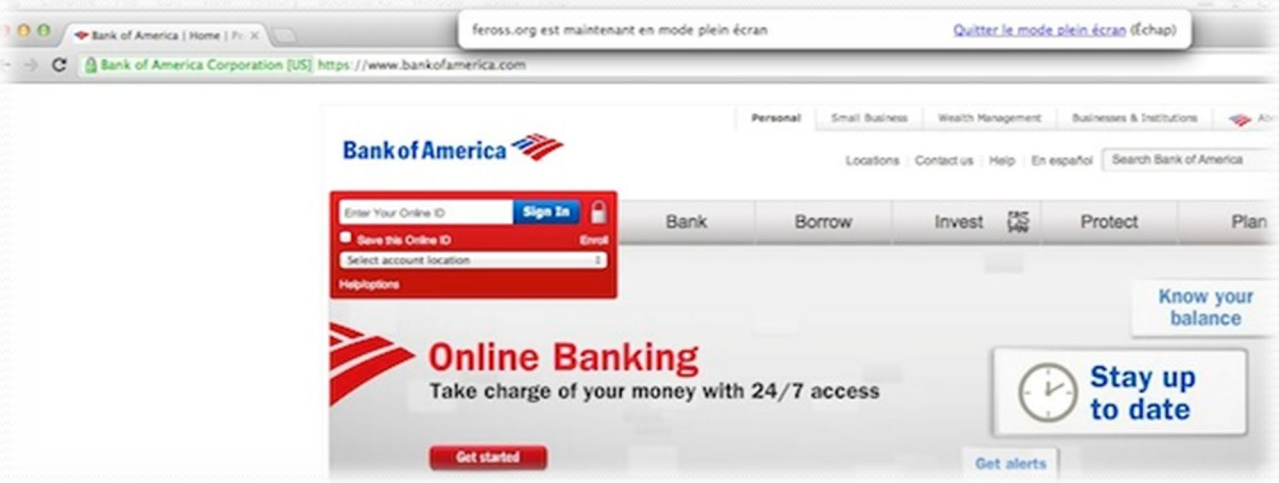
Failles Web : attaques

- Modification de chemins et URL :
  - Changement d'inclusion de fichier
  - **Injection de code JavaScript**
- **Injection SQL** (par l'intermédiaire des formulaires)
- Modification des **entêtes** (pirater une authentification)
- Modification des ***cookies***
- **Dépôt de fichiers** malicieux

Exemple : un virus à la place d'une image devant servir d'avatar sur un forum

Hameçonnage (Fishing)

- **Faux site Web** (usurpation d'identité)
- Utilisation *d'e-mails* et ***cross-site scripting*** **(XSS)**
- Cécité au changement :
  - Représentation lacunaire faite d'observations partielles

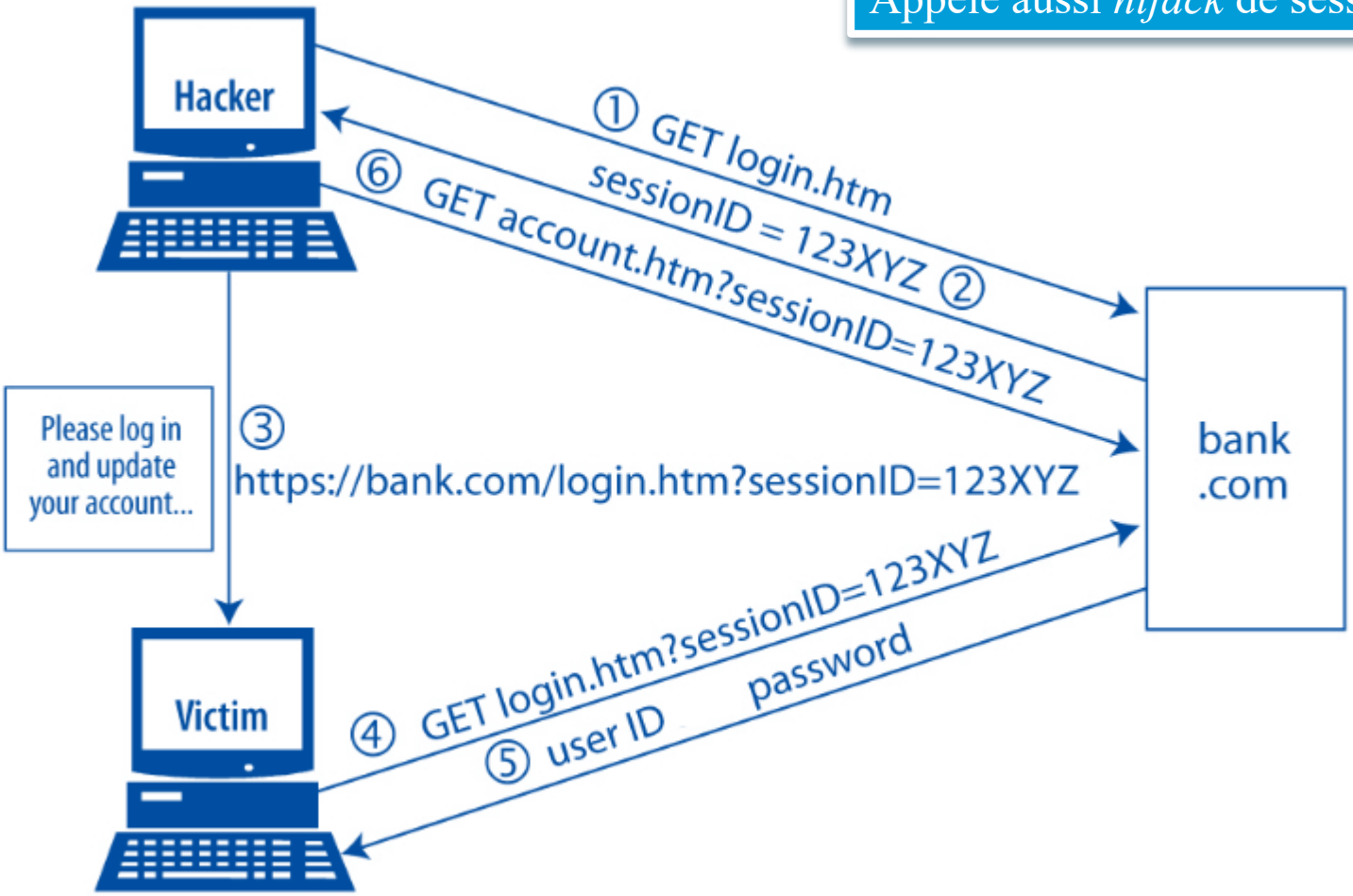


- http://feross.org/html5-fullscreen-api-attack/

Failles Web : CSRF

Cross-site request forgery

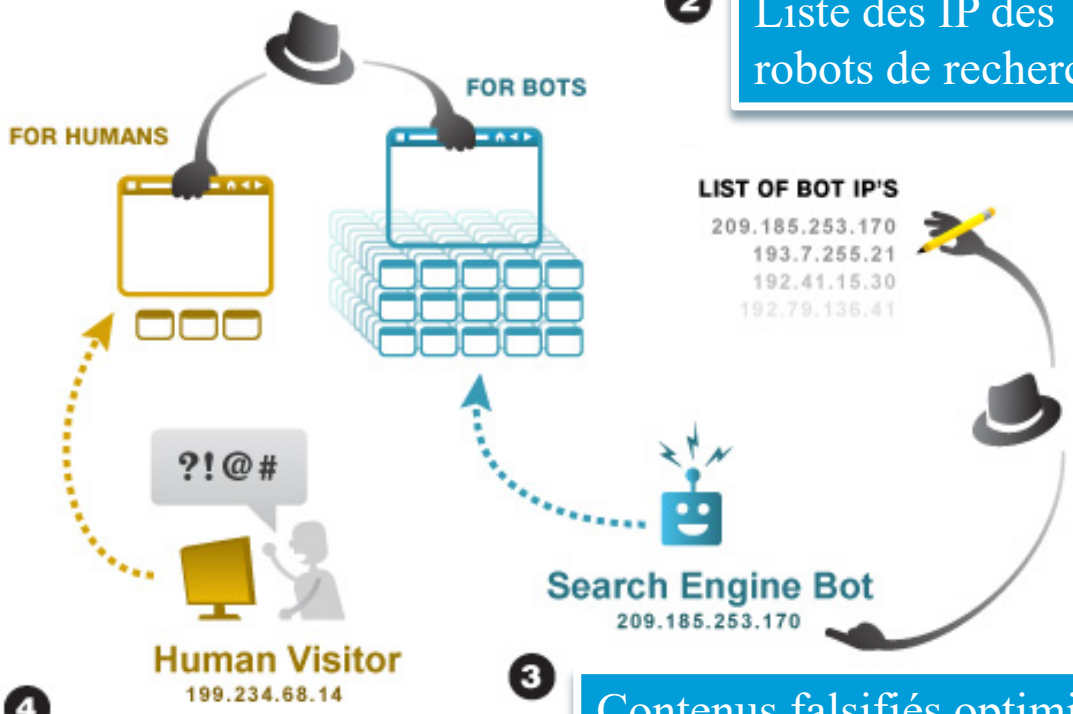
Appelé aussi *hijack* de session



Failles Web : Cloaking

1. Contenu différent pour un internaute ou un robot (filtrage par IP)

2. Liste des IP des robots de recherche



3. Contenus falsifiés optimisés pour le référencement

4. Contenu en décalage avec le référencement

Contrôle anti-injection

```php
/*
 * Fonction qui protège la variable passée en paramètre des injections SQL
 * et des caractères spéciaux.
 *
 * @author Mickaël Martin Nevot
 */
function quote_smart($value) {
    $value = utf8_encode($value);

    // Protection concernant Stripslashes
    if (get_magic_quotes_gpc()) {
        $value = stripslashes($value);
    }
    // Protection si ce n'est pas une valeur numérique ou une chaîne numérique
    if (!is_numeric($value)) {
        $value = '\'' . mysql_real_escape_string($value) . '\'';
    }
    return $value;
}
```

Injection SQL insolite

SOC

Crédits

Auteur

Mickaël Martin Nevot
mmartin.nevot@gmail.com

Carte de visite électronique

Relecteur

- Christophe Delagarde (christophe.delagarde@univ-amu.fr)
- Pierre-Alexis de Solminihac (pa@solminihac.fr)

Cours en ligne sur : www.mickael-martin-nevot.com