

# Cours Web n°10

## Sécurité, Accessibilité, Aspects légaux

Pierre Senellart ([pierre@senellart.com](mailto:pierre@senellart.com))  
Pierre Yger ([yger@unic.cnrs-gif.fr](mailto:yger@unic.cnrs-gif.fr))



3 décembre 2007

# Plan du cours

- 1 Sécurité
- 2 Accessibilité
- 3 Législation
- 4 Bonnes pratiques générales
- 5 Application

- Web : environnement **hostile**
- À moins de contrôler l'accès même au serveur Web, n'importe qui peut avoir accès au site Web... y compris des personnes **malveillantes**.
- Intérêt pour un attaquant : simple vandalisme, vol d'informations confidentielles, ajout ou lien vers des contenus illicites. . .
- Certaines choses sont de la **responsabilité de l'administrateur** (ex. avoir un serveur Web mis à jour régulièrement), le reste est de la **responsabilité du webmestre**.

# Injection de code HTML

## Problème

Un utilisateur peut entrer, à l'intérieur d'un paramètre HTTP destiné à être affiché, du code HTML (et donc également des indications de style CSS, du code JavaScript. . . ). Il modifie ainsi le code de la page HTML produite. Si ce paramètre est stocké pour être réaffiché (ex. commentaires de blog), ce code influe sur l'apparence du site pour d'autres utilisateurs.

## Exemple

Supposons que le paramètre HTTP login contienne :

```
<div style='color: red'>
```

dans le code PHP :

```
echo "Bonjour ".$_REQUEST["login"]."!";
```

## Solution

Utiliser `htmlspecialchars`.

# XSS (Cross-Site Scripting)

## Problème

Cas particulier de l'attaque précédente : insertion de code JavaScript dans une page HTML, qui sera réaffiché par d'autres utilisateurs ; le code JavaScript « vole » les informations saisies par l'utilisateur pour les transmettre à un autre site.

## Solution

Comme avant, utiliser `htmlspecialchars`, en particulier quand un texte saisi par un utilisateur est destiné à être affiché par un autre.

# Injection de code MySQL

## Problème

Un utilisateur peut modifier une requête MySQL en mettant des guillemets simples dans une variable à partir de laquelle sera construite la requête.

## Exemple

Supposons que \$passwd contienne « ' OR 1=1 -- » dans le code :

```
$result=mysql_query("SELECT * FROM T WHERE passwd='$passwd'");
```

## Solution

Utiliser `mysql_escape_string`.

# Injection de ligne de commande

## Problème

Un utilisateur peut modifier les programmes externes appelés par PHP à l'aide des fonctions PHP `system`, `exec`, `passthru` ...

## Exemple

Supposons que `$rep` contienne « `&& cat /etc/password` » dans le code PHP :

```
passthru("ls $rep");
```

## Solution

Utiliser `escapeshellcmd` ou `escapeshellarg`.

# Traversée de répertoires (Directory traversal)

## Problème

Un utilisateur peut, lors de l'utilisation des fonctions PHP  `fopen` ,  `readfile`  ..., en utilisant  `'/'` ,  `'..'` , accéder à des fichiers auquel il n'est pas censé avoir accès.

## Exemple

Supposons que  `$fichier`  contienne «  `../../../../../../etc/passwd`  » dans le code PHP :

```
readfile($fichier);
```

## Solution

Utiliser des **expressions rationnelles** pour vérifier que l'argument des fonctions accédant à des fichiers ne pointe pas vers des fichiers auxquels on ne souhaite pas donner accès (par ex., vérifier qu'il n'y a pas de  `'/'`  à l'intérieur).

# Contournement de validation de paramètres

## Problème

Un certain nombre de moyens permet d'imposer des restrictions **côté client** sur les valeurs que peut prendre des champs de formulaire : attributs `maxlength`, `disabled` ou `readonly`, champs de type `hidden`, code JavaScript appelé lors de la soumission d'un formulaire ou du changement des valeurs... Mais rien n'empêche un utilisateur de violer ces contraintes en désactivant le JavaScript, en écrivant un autre formulaire avec la même action, etc. . .

## Solution

Ne jamais faire confiance à une validation de champs de formulaire côté client, et **toujours** effectuer un **contrôle** de la validité des paramètres **côté serveur**, dans le script PHP.

# Capture de paquets IP

## Problème

Sur un réseau local, ou sur un réseau WiFi non crypté (ou avec un cryptage WEP simple à casser), il est possible à un attaquant de **regarder le contenu des paquets** IP en clair, contenant l'ensemble de la communication entre le navigateur et le serveur Web, y compris l'ensemble des paramètres HTTP, etc. Ce problème existe aussi, mais de manière moins importante, hors du cadre d'un réseau local ou sans fil.

## Solution

**Ne pas utiliser HTTP** pour transmettre des informations sensibles au travers d'Internet, d'autant plus dans le cadre d'un réseau local ou sans fil. **HTTPS**, un autre protocole permettant l'envoi crypté de messages sur le Web (et ayant d'autres fonctionnalités avancées par rapport à HTTP), doit être utilisé (**non traité dans ce cours**).

# Usurpation de session

## Problème

Utilisation d'une des techniques présentées auparavant (en particulier, XSS ou capture de paquets IP) pour récupérer l'**identifiant de session** d'un utilisateur (identifiant ordinairement stocké dans un Cookie), pour se faire passer pour lui.

## Solution

Résoudre les autres problèmes ! Terminer la session avec **session\_destroy** dès que celle-ci n'est plus nécessaire.

## Concurrence critique (Race condition)

### Problème

Un attaquant peut produire un comportement inattendu dans un script PHP en exploitant une faille de raisonnement qui suppose qu'un bloc d'instructions PHP sera **exécuté en une seule fois**, sans être en **concurrence** avec d'autres instructions.

### Exemple

Un script PHP récupère le plus grand entier stocké dans une table MySQL, l'augmente de un, sauvegarde un fichier avec pour nom cet entier, et ajoute une ligne correspondante dans la table MySQL. Il y aura concurrence si deux scripts s'exécutent simultanément, et que les deux consultent la table pour connaître le plus grand entier avant d'avoir ajouté une ligne dans celle-ci.

### Solution

Bien réfléchir aux cas de concurrence critique. Utiliser les **transactions** de MySQL 5, utiliser des **verrous** sur les fichiers.

# Déni de service (DOS, Denial Of Service)

## Problème

Attaquer un site Web (ou un autre service sur Internet) en exigeant du serveur **plus que ce qu'il ne peut servir** (très grand nombre de connexions, calculs coûteux. . . )

## Solution

Essentiellement de la responsabilité de l'administrateur du site, mais le web-mestre peut prévenir certaines attaques en 1) évitant les fichiers trop lourds à télécharger 2) évitant les calculs coûteux inutiles dans les scripts PHP.

# Ingénierie sociale (Social engineering)

## Problème

Probablement la plus utilisée des attaques, à la base de la propagation de bon nombre de virus, de l'**hameçonnage** (phishing), etc. : exploiter une **faille** non pas dans un quelconque logiciel, mais dans le **raisonnement humain** ! Pousser un utilisateur honnête à exécuter un logiciel malveillant, à donner des informations confidentielles, etc.

## Solution

Garder en tout un esprit critique, faire preuve de bon sens, et ne pas se laisser abuser par une méconnaissance technique !

## Résumé

- Bien protéger les paramètres non contrôlés avec `htmlspecialchars`, `mysql_escape_string`, `escapeshellcmd`, `escapeshellarg`...
- Contrôler la validité des paramètres côté serveur.
- Ne pas transmettre d'informations sensibles en clair.
- Bien réfléchir !

Pour bien comprendre les problèmes et comment les résoudre, une manière efficace et ludique : s'entraîner à attaquer des sites prévus pour (et seulement ceux-là, le fait d'attaquer un site quelconque est puni par la loi, jusqu'à 75 000 € d'amendes et 5 ans d'emprisonnement).

Un index de tels sites : <http://www.hackergames.net/>

# Plan du cours

- 1 Sécurité
- 2 Accessibilité**
- 3 Législation
- 4 Bonnes pratiques générales
- 5 Application

- Un site Web est **accessible** lorsqu'il est possible pour n'importe quelle personne d'y accéder de façon équivalente, quelle que soit :
  - ▶ le navigateur utilisé ( Firefox, Internet Explorer, Opéra...),
  - ▶ l'interface (utilisation du clavier ou de la souris),
  - ▶ la plate forme d'accès ( c'est à dire le système d'exploitation : Windows, Mac OS ou encore Linux...)
  - ▶ le périphérique d'affichage (un écran plat 17 pouces, un ordinateur portable, un téléphone mobile relié au Web...) :
    - ★ Éviter les positionnement au pixel près
    - ★ Tester les différentes tailles de fenêtre et de résolution
  - ▶ Autres :
    - ★ Tester le mode texte
    - ★ Tester les zooms
    - ★ Bannir toute dépendance à JavaScript
- Certains sites proposent d'évaluer l'accessibilité d'une page Web comme par exemple : <http://www.ocawa.com/>

Source :

<http://www.w3.org/WAI/gettingstarted/Overview.html.fr>

**Images, animations** Utilisez l'attribut alt pour décrire la fonction de chaque graphique.

**Multimédia** Fournissez légendes et transcriptions pour l'audio, et des descriptions pour les vidéos.

**Liens hypertextes** Utilisez des énoncés pertinents hors contexte. Par exemple, évitez *cliquer ici*.

**Organisation**

- Utilisez des têtes de sections, des listes et une structure cohérente
- Utilisez CSS si possible et tester vos pages sans CSS.

**Figures, diagrammes** Décrivez-les dans la page.

**Scripts, applets, plug-ins** Fournissez une alternative quand le contenu actif est inaccessible ou non traité.

**Tableaux** Facilitez la lecture ligne par ligne. Résumez.

# Plan du cours

- 1 Sécurité
- 2 Accessibilité
- 3 Législation**
- 4 Bonnes pratiques générales
- 5 Application

- Protection de vos données
  - ▶ Déclaration à la CNIL : obligatoire uniquement dans le cas d'une utilisation des informations personnelles sur les visiteurs du site.
  - ▶ Précisez la licence d'utilisation de votre contenu (par exemple, CC)
- Respect des droits d'auteurs
- Déclaration de tout revenu même faible (ex. : issu des bannières publicitaires)
- Responsabilité légale : toutes les images et tous les propos (même blog et forum) sont sous votre responsabilité
  - ▶ Protection des mineurs si votre site contient des images ou des propos pouvant choqués
  - ▶ Pas de propos injurieux, racistes, etc.
- Bonnes pratiques

- Pour préciser les conditions de réutilisation du contenu et données de votre site, on peut indiquer une **licence** d'utilisation. Solution simple : licences Creative Commons.
- Dans tous les cas, vous conservez le **droit d'auteur** sur tout ce que vous écrivez (droit **passif**). Par défaut, vous ne pouvez donc réutiliser un contenu sans permission explicite.

Votre site Web présente-t-il une originalité telle qu'il faille protéger tout son contenu ?

Les différentes options :

**Paternité** l'œuvre peut être librement utilisée, à la condition de l'attribuer à son l'auteur en citant son nom.

**Pas d'utilisation commerciale** le titulaire de droits peut autoriser tous les types d'utilisation ou au contraire restreindre aux utilisations non commerciales (les utilisations commerciales restant soumises à son autorisation).

**Pas de modification** le titulaire de droits peut continuer à réserver la faculté de réaliser des œuvres de type dérivées ou au contraire autoriser à l'avance les modifications, traductions...

**Partage à l'identique des conditions initiales** à la possibilité d'autoriser à l'avance les modifications, peut se superposer l'obligation pour les œuvres dites dérivées d'être proposées au public avec les mêmes libertés (sous les mêmes options Creative Commons) que l'œuvre initiale.

Les 6 contrats :

- 1 Paternité
- 2 Paternité, Pas de Modification
- 3 Paternité, Pas de Modification, Pas d'Utilisation Commerciale
- 4 Paternité, Pas d'Utilisation Commerciale
- 5 Paternité, Pas d'Utilisation Commerciale, Partage des Conditions Initiales à l'Identique
- 6 Paternité, Partage des Conditions Initiales à l'Identique

Vous permet de spécifier ainsi exactement ce que vous autorisez.

- Défini par le "Code de la Propriété Intellectuelle" (depuis 01/07/92)
  - ▶ loi du 11 mars 1957
  - ▶ loi du 3 juillet 1985
- Il existe deux types de droits :
  - ▶ les droits moraux : visent à protéger **la personnalité** de l'auteur au travers de son œuvre et à respecter celle-ci. Il consiste pour l'auteur au droit au "respect de son nom, de sa qualité, de son œuvre"
  - ▶ les droits patrimoniaux : portent sur l'exploitation de l'œuvre

Ces droits sont attachés à la personne de l'auteur :

- Ils sont inaliénables : donc pas cessibles (l'auteur ne peut pas les vendre). En revanche, ils sont transmissibles à sa mort aux héritiers ou à des exécuteurs testamentaires.
- Ils sont perpétuels et imprescriptibles.

**le droit de divulgation** il permet à l'auteur de décider quand son œuvre est terminée et qu'elle peut être divulguée au public.

**le droit de paternité** l'auteur a le droit de revendiquer la paternité de son œuvre. Cela se traduit généralement par la mention de l'auteur lors de l'exploitation de l'œuvre.

**le droit au respect de l'intégrité de l'œuvre** l'auteur peut s'opposer à toutes modifications, déformations ou mutilations de son œuvre (L'application de ce droit est cependant nuancée dans la jurisprudence récente).

**le droit de retrait et de repentir** qui consiste au retrait par l'auteur de son œuvre déjà divulguée de la sphère du marché en contrepartie d'une compensation financière à hauteur du préjudice subi par le diffuseur. Le droit à s'opposer à toute atteinte préjudiciable à l'honneur et à la réputation

Ces droits sont cessibles.

**le droit de représentation** par ce droit, l'auteur peut donner son autorisation à la représentation ou à l'exécution publique de son œuvre. Le caractère public est particulièrement important.

**le droit de reproduction** ce droit comprend la possibilité que l'auteur a d'autoriser la copie de tout ou d'une partie de son œuvre et de fixer les modalités de cette dernière.

- La durée des droits patrimoniaux couvre la vie de l'auteur.
- Au décès de l'auteur, ce droit persiste au bénéfice de ses ayant droit pendant l'année civile en cours et les 70 années qui suivent.

- à ces 70 ans maximum s'ajoutent :
  - ▶ Pour certaines œuvre, la durée de la Première Guerre mondiale (6 ans et 152 jours) et/ou la durée de la Seconde Guerre mondiale (8 ans et 120 jours).
  - ▶ 30 ans supplémentaires si l'auteur est « mort pour la France »
  - ▶ Dans le cas d'une œuvre de collaboration, c'est la date du décès du dernier collaborateur qui sert de référence
  - ▶ Dans le cas d'une œuvre audiovisuelle, œuvre de collaboration, c'est la même chose mais les collaborateurs sont précisément nommés : scénariste, auteur des paroles, auteur des compositions musicales, réalisateur principal
  - ▶ Dans le cas d'une œuvre sous pseudonyme, anonyme ou collective, c'est la date de publication qui fait foi sauf si par après les auteurs se font connaître
  - ▶ Dans le cas des œuvres posthumes, c'est toujours 70 ans après le décès de l'auteur. Si celles-ci ne sont divulgués qu'après ce laps de temps de 70 ans, le temps de protection tombe à 25 ans à compter du 1<sup>er</sup> janvier de l'année de publication

# Plan du cours

- 1 Sécurité
- 2 Accessibilité
- 3 Législation
- 4 Bonnes pratiques générales**
- 5 Application

Source : <http://www.opquast.com/bonnes-pratiques/>

- Formulaire : préciser les raisons du rejet à l'utilisateur
- Serveur : politique de confidentialité et respect de la vie privée sont précisés
- Plug-ins : mettre à disposition un lien pour les télécharger
- Cookies : indiquer leurs présences, leurs objectifs et les limitations (consultation et utilisation) en cas de refus
- Multimédia :
  - ▶ prévenir tout téléchargement supérieur à 50ko
  - ▶ permettre l'arrêt des sons ou animations qui tournent en boucle
  - ▶ éviter l'affichage d'une page contenant plus de 150ko à télécharger
- Pop-ups : éviter une utilisation intempestive

# Plan du cours

- 1 Sécurité
- 2 Accessibilité
- 3 Législation
- 4 Bonnes pratiques générales
- 5 Application**

Reprenez tout ce qui a été fait en projet et en TD, et vérifiez que :

- 1 Vos sites sont sûrs.
- 2 Vos sites sont accessibles.
- 3 Vos sites respectent bien la législation.