La veille sur la sécurité

Les agressions dans la cybersphère :

- les ADS (Attaque par Déni de Services pour neutraliser un système informatique et le rendre inopérant)
- le cyberespionnage
- le cyberharcèlement
- la cyberfraude (triche aux examens, lors de vote, falsification de documents officiels, etc.)
- le cyber-whistleblowing (considéré comme un délit voire un crime dans les dictatures notamment)
- la cybercontrefaçon (musique, livre, jeux-vidéo, logiciels) et le cybermarché noir (achat en ligne de marchandises illégales)
- la cyberfinance criminelle
- la cyberpropagande
- la cyberusurpation d'identité
- le cybercambriolage (vol de données)
- le défaçage (modifier l'apparence d'un site, d'un blog, etc.)

Faille xss

Selon Cenzic, ce type d'attaque représente environ 25% des attaques.
La faille XSS (plus officiellement appelée Cross-Site Scripting) est une faille permettant l'injection de code HTML ou JavaScript dans des variables mal ou non protégées.

1. le pirate envoie un message piégé avec un lien de redirection vers le site pirate,
2. l'utilisateur ouvre le message et clique sur le lien,
3. redirection vers l'URL et récupération des cookies de l'utilisateur dans un document,
4. Le pirate récupère les données contenues dans le document.

L'attaque par force brute

Le principe ici est d'essayer de "casser" votre mot de passe en testant toutes les combinaisons possibles.

Moins utilisée que la faille XSS, cette technique reste cependant relativement courante dans la mesure où un ordinateur personnel est capable de tester plusieurs centaines de milliers voire quelques millions de mots de passe par seconde.

Pour vous défendre :

- Passoire : ce programme en ligne génère des mots de passe, en évalue et en améliore la robustesse,
- www.keylength.com: cet outil en ligne permet de calculer la bonne taille d'une clef suivant différentes méthodes.

Injection sql

L'injection SQL, ou SQLi, est un type d'attaque sur une application web qui permet à un attaquant d'insérer des instructions SQL malveillantes dans l'application web, pouvant potentiellement accéder à des données sensibles dans la base de données ou détruire ces données.

- www.httpcs.com : Scanner en ligne de détection d'injection SQL
- gits.hydraze.org : comment repérer les injections SQL, les exploiter et les corriger

Faille upload

Beaucoup de sites Web offrent la possibilité aux clients d'y uploader des fichiers comme des photos, des vidéos, des CV... Donc il ne s'agit pas vraiment d'une vulnérabilité, mais c'est le fait de ne pas contrôler ce que le client charge sur le serveur qui constitue une vulnérabilité très dangereuse.

Le principe de l'attaque est très simple. Le pirate essaie d'uploader un fichier qui contient du code malveillant ou un code PHP de sa création. Si la faille est là alors le fichier finira pas atterrir sur le serveur. Il suffit ensuite au pirate d'appeler son fichier pour que celui-ci s'exécute.

Bien entendu une telle attaque peut avoir de graves conséquences comme par exemple:

- Espionnage des fichiers et dossiers du site
- Accès au fichiers systèmes et fichiers confidentiels
- Destruction ou altération de données existantes sur le serveur
- Prise de contrôle du serveur

Comment s'en protéger ?

Au niveau du code PHP

Comme d'habitude, la vulnérabilité est due au mauvais contrôle des entrées de l'utilisateur, alors qu'il suffisait de vérifier si le type/Mime du fichier uploadé correspond bien à une image JPEG ou PNG en utilisant le code suivant par exemple:

```
<?php

   if(preg_match("#jpeg|png#",$_FILES["photo"]["type"]))

      // Accepter l'upload

   else

      echo "Format du fichier invalide.";

?>
```

Il faut également penser à isoler les fichiers chargés dans un dossier à part pour minimiser les risques de rebond au cas où il s'agit d'un fichier malveillant. Renommer les fichiers chargés sera aussi d'une grande utilité car le pirate aura du mal à appeler son fichier s'il ne connaît pas son chemin et son nom.

Une bonne pratique consiste à changer les droits du fichier chargé à l'aide de la fonction **chmod()**.

Faille include

La faille Include touche comme son nom l'indique les fonctions du type include(). Ce type de fonction (comme require(), include(), include_once()) a pour objectif d'inclure un fichier et d'exécuter son contenu sur le serveur.

Pour les détecter, cela se fait de manière assez simple. Il suffit, comme lors de chaque audit sécuritaire, de modifier tous les paramètres possibles de l'URL et d'observer le résultat. Le but est de faire planter le script en tentant d'inclure une page qui n'existe pas.

La meilleure façon pour se protéger d'une **include PHP** est de :

- Faire un test pour chaque page de votre site, certes c'est long mais il n'y a pas mieux à vous de voir après.
  if ($page == " astuces") { include (" astuces. php") }
  if ($page == " photos") { include (" photos. php")}
  .
  else { include (" index. php") }
- En php4, mettre la commande allow_url_fopen à off. Cette dernière permet la lecture de fichiers situés sur un autre serveur;
- En php5, mettre la commande allow_url_include à off. C'est un apport par rapport à php4, qui distingue les fichiers lus des fichiers inclus.

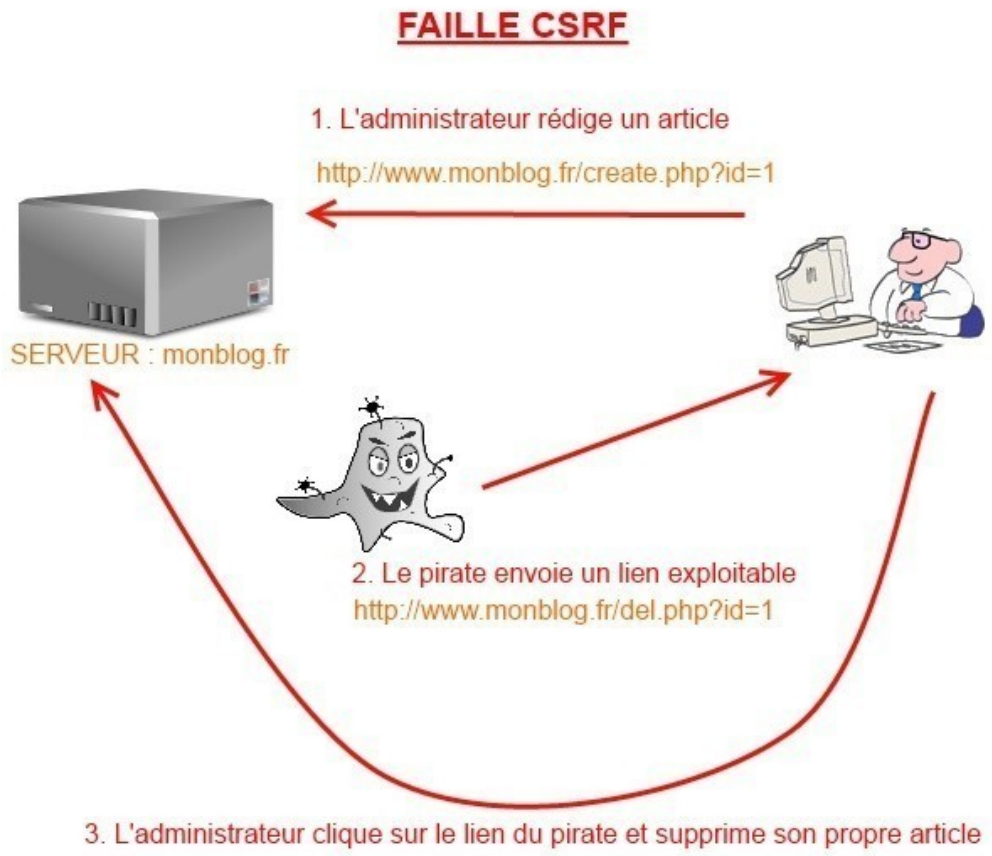
Faille CSRF

Qu’est ce que la faille CSRF

Le nom *CSRF* vient de *Cross-Site Request Forgery* qui, si l’on essaie de donner une définition en français, signifie Falsification de requête inter-sites. On n’est pas plus avancé, je sais.

En fait, il s’agit d’effectuer une action visant un site ou une page précise en utilisant l’utilisateur comme déclencheur, sans qu’il en ait conscience.

On va deviner un lien qu’un utilisateur obtient habituellement, et tout simplement faire en sorte qu’il clique lui-même sur ce lien.



Comment se protéger contre la faille CSRF ?

on utilise habituellement 2 principes complémentaires :

L’authentification par jeton

Un jeton (aussi appelé **token** en anglais) est un nombre ou une chaîne de caractère aléatoire qui va être testée avant toute modification ou édition d’un article.

Il se présente habituellement sous forme de **hash md5** comme celui-ci :

```
b6cf20590a57f4685c9bdc6c53d12ff8
```

Ce token doit être créé dans un fichier PHP qui sera appelé sur toutes les pages. Typiquement, il s’agit d’un fichier du type *config.php* ou *functions.php*.

On génère souvent un nombre « aléatoire » avec des fonctions utilisant le temps en PHP. Par exemple on peut générer un jeton avec :

```
md5(uniqid(mt_rand(), true));
```

La fonction uniqid() génère un identifiant unique basé sur le temps en microsecondes. Cependant PHP ne recommande **pas** cette fonction car elle ne génère **pas** des chaînes impossibles à deviner à l’avance.

Du coup, on va plutôt utiliser :

```
md5(bin2hex(openssl_random_pseudo_bytes(6)));
```

qui est cette fois hautement **sécurisé**.

La fonction openssl_random_pseudo_bytes() génère une chaîne pseudo-aléatoire d’octets de taille 6 bits * 2 qu’on convertit ensuite en hexadécimal, 6 étant le nombre donné en paramètre de la fonction (on peut le changer).

Ainsi, dans un fichier PHP global on va écrire le code suivant :

```
<?php
if (!isset($_SESSION['jeton'])) {
```

```
    $_SESSION['jeton'] =
bin2hex(openssl_random_pseudo_bytes(6));
}
?>
```

Ce code signifie : Si le jeton de session n'est pas encore défini, on le génère aléatoirement et on l'enregistre pour la session courante.

Ensuite, à chaque connexion d'un utilisateur, on va devoir générer un jeton **qui lui est propre**.

Pour cela, on peut avant chaque connexion régénérer le jeton, en **supprimant** le jeton de la session précédente :

```
unset($_SESSION['jeton']);
```

Il reste ensuite à modifier dynamiquement les liens de suppression, admettons que le lien précédent était écrit de la forme :

```
<a href="http://www.monblog.fr/del.php?id=<? echo
get_article_id();
?>>Supprimer l'article</a>
```

On le remplace par :

```
<a href="http://www.monblog.fr/del.php?id=<? echo
get_article_id() .
"&jeton=". $_SESSION['jeton']; ?>>Supprimer l'article</a>
```

L'url de suppression s'affiche donc comme ceci :

```
http://www.monblog.fr/del.php?id=1&jeton=b6cf20590a57f4685c9bd
c6c53d12ff8
```

Au lieu de s'afficher comme cela :

```
http://www.monblog.fr/del.php?id=1
```

Et enfin, dans notre fichier de suppression *del.php*, on va s'assurer qu'il existe un jeton et que ce jeton soit valide. Le fichier, avant toute modification se présentait comme cela :

```php
<?php
if(isset($_GET['id'])) {
    supprimer_article($_GET['id']);
} else {
    die("Aucun ID d'article sélectionné.");
}
?>
```

Il devient donc :

```php
<?php
if(isset($_GET['id']) && isset($_GET['jeton']) &&
($_GET['jeton'] == $_SESSION['jeton'])) {
    supprimer_article($_GET['id']);
} else {
    die("ID ou jeton de session invalide.");
}
?>
```

Ce qui signifie : Si l'id de l'article est défini mais aussi le jeton et que ce jeton correspond au jeton de la session actuelle, alors on peut le supprimer.

Dernière note, on utilise *$_GET* qui récupère les paramètres depuis une URL, il aurait été préférable est **encore plus sécurisé** d'utiliser *$_POST* avec un formulaire pour ne pas afficher de jeton dans les URLs.

La demande de confirmation

Que ce soit pour éviter les suppression par erreur ou les tentatives d'exploitation de la faille CSRF, il est **indispensable** de demander **confirmation** de suppression d'un article.

Ici, si je clique sur *supprimer mon article* et qu'il est immédiatement et définitivement supprimé, il faut réfléchir à trois fois avant de cliquer.

Cette méthode est beaucoup utilisée pour définir un mot de passe : On demande d'abord **l'ancien.**

PING flood

Le ping flood est une **forme d'**attaque par déni de service. L'attaque provoque donc un « déni de service ». Vous pouvez vous représenter cette attaque comme un canular téléphonique : un hacker malveillant appelle sans cesse et raccroche immédiatement. La liaison est ainsi bloquée et indisponible. Il n'est alors plus possible de répondre aux appels légitimes.

Les attaques de flood connues comme le ping flood, le HTTP flood, SYN Flood et l'UDP Flood consistent à **saturer un système cible avec des demandes insensées** jusqu'à ce qu'il s'écroule. Le ping flood ne doit pas être confondu avec le ping of death qui provoque le crash du système cible sans le surcharger.

Qu'est-ce qu'un ping flood ?

Le ping flood est une cyberattaque visant différents systèmes connectés à Internet. Les systèmes attaqués peuvent être aussi bien des serveurs que des routeurs ou des ordinateurs de particuliers.

D'un point de vue technique, le ping flood repose sur l'Internet Control Message Protocol (ICMP). Ce protocole et la commande ping correspondante sont normalement utilisés pour réaliser des tests sur le réseau. Un ping flood provoque la **surcharge de l'ordinateur cible avec**

des paquets « Echo Request » ICPM. Si le hacker dispose de plus de bande passante que la victime, cette dernière est évacuée du réseau.

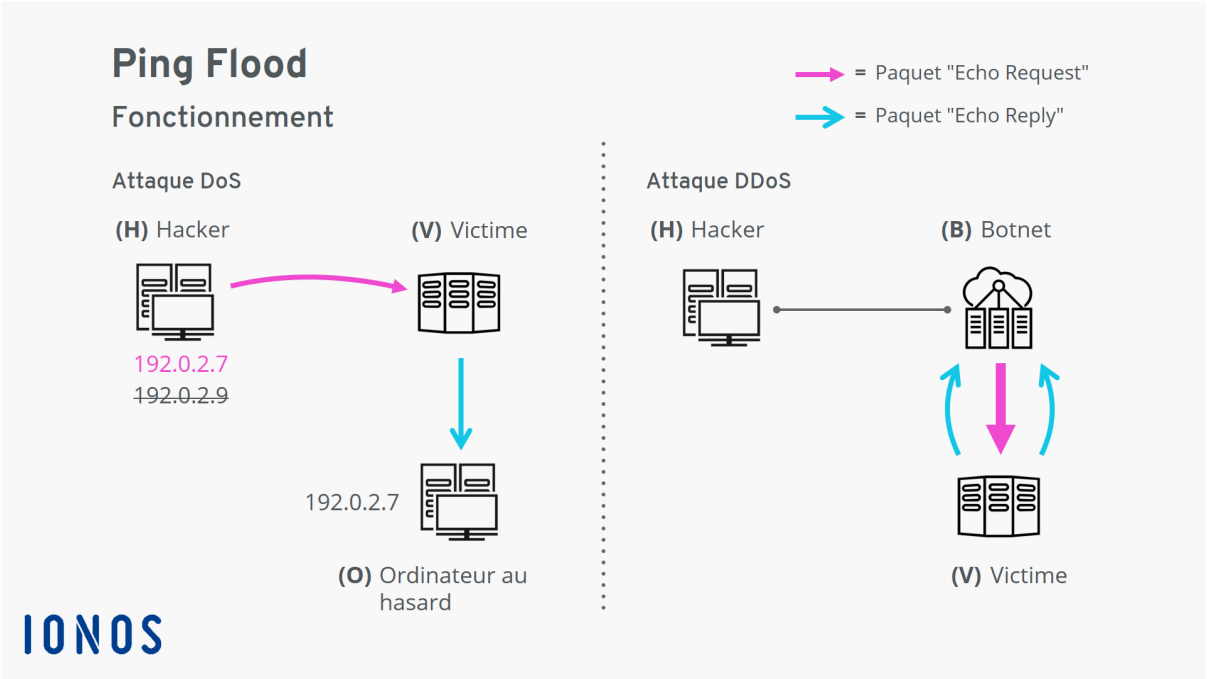
Fonctionnement du ping flood

Le fonctionnement du ping flood est simple :

1. Le **hacker envoie des paquets « Echo Request »** par vague à la machine de la victime.
2. Cette dernière **répond avec des paquets « Echo Reply »**.

Chaque paquet « Echo Request » entrant demande de la bande passante à la victime. Comme un paquet « Echo Reply » est renvoyé pour chaque paquet entrant, le trafic réseau sortant implique un volume de données tout aussi élevé. Si le hacker dispose de suffisamment de bande passante, il peut **exploiter toutes les capacités réseau de la victime à disposition**. Le trafic réseau légitime est alors interrompu ou s'arrête complètement.

Le ping flood peut être une attaque DoS ou DDoS selon que l'attaque provient d'un ordinateur individuel ou d'un réseau d'ordinateurs.



Le hacker inonde la victime avec un flot de paquets de données.

Attaque ping flood sous forme de Denial of Service (DoS)

Dans la variante la plus simple de cette attaque, le hacker (H) envoie les paquets « Echo Request » à la victime (V) **depuisune seule machine**. Pour ne pas dévoiler son identité, le hacker usurpe une adresse IP. Un ordinateur au hasard accessible à cette adresse IP (O) sera alors bombardé par les paquets « Echo Reply » correspondants. Cet effet de rétrodiffusion est également appelé « backscatter ». Dans certaines variantes de ping flood, notamment dans les attaques par rebond, la rétrodiffusion est utilisée comme une arme à part entière.

Pour envoyer un ping flood contre sa victime, le hacker utilise la commande ping ou une alternative moderne telle que l'outil hping. **L'attaque commence sur l'invite de commande**. Le ping flood est

déclenché à l'aide d'une commande conçue spécifiquement pour l'attaque. Pour des raisons de sécurité, nous ne pouvons présenter ici qu'un modèle approximatif du code hping utilisé :

```
hping --icmp --flood --rand-source -p <Port>
<Adresse IP>
```

Jetons un œil aux différentes options :

- L'option *--icmp* indique à l'outil d'utiliser l'ICMP comme protocole.
- L'option *--flood* est tout particulièrement importante ici. Selon la documentation de la commande hping, celle-ci fait en sorte que des paquets soient envoyés aussi rapidement que possible. D'autre part, cette option induit que les paquets « Echo Reply » entrants seront rejetés sans être pris en compte. Par conséquent, au lieu d'envoyer un ping et d'attendre une réponse comme dans une utilisation normale de la commande ping, les pings sont envoyés à répétition aussi rapidement que possible.
- L'option *--rand-source* travestit l'adresse IP de l'expéditeur. Une adresse IP aléatoire est renseignée à la place de la véritable adresse de l'expéditeur.

Attaques ping flood sous forme de Distributed Denial of Service (DDoS)

Pour déclencher un ping flood « distributed », le hacker (H) utilise un botnet (B). Les bots placés sous le contrôle du hacker lancent un ping flood contre la victime (V) sur ordre du hacker. Comme plusieurs ordinateurs se dressent contre une même cible, la **bande passante disponible du côté du hacker est nettement plus importante**. Seule une cible bien protégée peut résister à ce type d'attaque.

Dans ce scénario, le hacker n'envoie pas les paquets « Echo Request » depuis son ordinateur. Il n'a donc aucune raison de falsifier son adresse IP. Au lieu de cela, les bots agissent depuis leur propre adresse. La rétrodiffusion touche donc les ordinateurs zombies du botnet.

Mesure de protection contre les attaques ping flood (ping flood attacks)

Il existe en principe trois méthodes pour se protéger contre les attaques ping flood :

Configurer le système à protéger pour une sécurité élevée

La méthode la plus simple pour se protéger contre les attaques ping flood consiste à **désactiver la fonctionnalité ICMP sur l'appareil de la victime**. Cette mesure offre une solution immédiate lors d'une attaque mais peut également être mise en place de façon préventive pour réduire la fenêtre d'accès.

Par ailleurs, les routeurs et les pare-feu peuvent être configurés de façon à ce que le **trafic réseau malveillant entrant soit identifié et filtré**. L'utilisation de technologies de Load Balancing(en français « répartition de charge ») et de Rate Limiting (« limitation du débit ») contribue à la protection contre les attaques DoS.

Utilisation d'un service basé sur le cloud pour affaiblir les attaques par déni de service

Les grands fournisseurs comme Cloudflare mettent à disposition des serveurs dans des centres de données répartis dans le monde entier. Si

vous exploitez votre propre site Internet, vous pouvez faire passer votre trafic de données par ces centres de données. Vous disposerez ainsi d'une **bande passante nettement plus importante** pour absorber les attaques DDoS. D'autre part, le trafic de données est filtré par des systèmes intégrés tels que des pare-feu, des répartiteurs de charge et des limiteurs de débit.

Utilisation d'un matériel spécifique avant le système à protéger

Il est également possible de protéger le système avec un matériel spécifique mais cette solution n'a d'intérêt que pour les entreprises très actives dans le domaine informatique. Ces appareils offrent ou combinent les fonctionnalités de pare-feu, de répartiteurs de charge et de limiteurs de débit et filtrent ou bloquent le trafic réseau malveillant.

RGPD

(règlement général de protection des données)

Le RGPD renforce l'obligation d'information et de transparence à l'égard des personnes dont vous traitez les données (clients, collaborateurs, etc.).

Vérifiez que l'information comporte les éléments suivants :

pourquoi vous collectez les données (« la finalité » ; par exemple pour gérer l'achat en ligne du consommateur) ;
ce qui vous autorise à traiter ces données (le « fondement juridique » : il peut s'agir du consentement de la personne concernée, de l'exécution d'un contrat, du respect d'une obligation légale qui s'impose à vous, de votre « intérêt légitime ») ;
Qui a accès aux données (indiquez des catégories : les services internes compétents, un prestataire, etc.) ;
Combien de temps vous les conservez (exemple : « 5 ans après la fin de la relation contractuelle ») ;
Les modalités selon lesquelles les personnes concernées peuvent exercer leurs droits (via leur espace personnel sur votre site internet, par un message sur une adresse email dédiée, par un courrier postal à un service identifié) ;
Si vous transférez des données hors de l'UE (préciser le pays et l'encadrement juridique qui maintient le niveau de protection des données).

HTTP / HTTPS

HTTP et **HTTPS** sont les deux protocoles utilisés pour transmettre des données sur Internet et sites Web. **HTTP** est synonyme de protocole de transfert hypertexte, tandis que l'ajout du «S» dans **HTTPS** signifie (HyperText Transfer Protocol Secure) qu'il s'agit d'une connexion sécurisée.

Les avantages d'un site HTTPS

Si vous êtes là, c'est sans doute que vous êtes déjà convaincu qu'il est utile de mettre en place la protection de votre site internet.

Mais voici tout de même un petit rappel de tous les avantages apportés par un site sécurisé :

- Avec le protocole de sécurité HTTPS / SSL, les informations sensibles transitant via le réseau internet (telles que les mots de passe, cartes bancaires, etc.) ne peuvent pas être espionnées par une personne s'étant introduite dans un des équipements utilisés par le réseau (connexion wifi, ligne téléphonique, fibre optique, routeur, …).
  Le protocole agit un peu comme un bouclier anti-hacker / anti-intrusions.

- Le site bénéficie d'un label en forme de cadenas s'affichant dans la barre d'adresse du navigateur.
  A l'inverse, si un site n'est pas https, les navigateurs mettent les utilisateurs en garde avec des messages d'alerte. Inutile de dire que ça fait "mauvais genre". ;-)

- Le référencement est amélioré dans Google.
  Sur le net, c'est bien connu, tout ce qui nous rend plus visible, nous rend plus fort. De plus depuis 2018, Google annonce d'emblée la couleur dans les pages de résultats : toutes les pages HTTP sont mentionnées comme "non-sécurisées". Autant dire que quand on tient à son image de marque, on préfère éviter ça.

- Votre site bénéficie d’une crédibilité plus importante auprès de vos visiteurs.

  On sait que l’internaute est de plus en plus méfiant. Et on sait aussi qu’il n’a pas tort ! Car les fraudes et les arnaques vont bon train sur le web.

  Tout mettre en œuvre pour rassurer vos visiteurs n’est donc jamais une mauvaise idée.

- La version mobile de votre site est également sécurisée.

  Toutes les règles de sécurité valables sur postes fixes le sont aussi sur appareils mobiles.

Autant être franc avec vous, si vous choisissez de vous lancer seul dans l’aventure, c’est une opération qui pourra vous sembler un peu complexe. Du moins si vous n’avez pas l’âme d’un geek. Mais selon la solution que vous aurez choisie, pour créer ou refaire votre site, cela pourra être pris entièrement en charge par votre prestataire.

Rendre son site web HTTPS : la procédure

SiteW s’occupe automatiquement de l’enregistrement et du renouvellement du certificat SSL.

En effet, la mise en place d'une connexion sécurisée HTTPS requiert l'enregistrement d'un certificat SSL auprès d'un organisme certificateur (Symantec, Commodo, Let's encrypt, …).

Si vous vous lancez en solo, voilà les étapes à suivre pour la création d'un site HTTPS :

- Avant toute chose, vous devez effectuer une demande de signature de certificat (CSR) sur votre serveur, afin de récupérer les données nécessaires (certificat de clé publique) pour l'enregistrement du certificat SSL.
- Vient ensuite l'étape du choix et de l'achat du certificat SSL. Vous trouverez différents types de certificats (que nous détaillerons plus bas), qui vous offriront des degrés de sécurité web distincts. Les prestataires pratiquant des tarifs très variables comparent bien les différentes offres.
- Il vous faudra ensuite passer à la phase de configuration de votre serveur web. Selon le logiciel utilisé, la procédure ne sera pas la même. Référez-vous aux tutoriels existants.
- Mettez ensuite en place des redirections 301 (c'est à dire permanentes) pour rediriger le trafic de vos pages HTTP vers vos nouvelles pages HTTPS, de sorte à ce que la sécurisation n'impacte pas négativement votre référencement.

- Vous pourrez par la suite checker la conformité de votre configuration SSL, grâce à certains outils en ligne, comme par exemple: https://www.ssllabs.com
- Pensez à actualiser votre compte Google Webmaster Tools avec votre nouvelle adresse web HTTPS.
- Mettez également à jour tous vos documents, et tous vos comptes en ligne, contenant vos URLs (campagnes emailing, pub, réseaux sociaux, affiliations...)

Et surtout n'oubliez pas, à terme, de renouveler votre certificat SSL.

- Vérifiez tous les liens qui pointent vers votre site. Ceci afin de ne pas générer d'erreurs 404. Pour ce faire, vous pouvez recourir à des logiciels gratuits, comme Broken Link check. Cela, encore une fois, vous évitera de voir pâtir votre référencement.
- Enfin, terminez en regardant la compatibilité de vos fichiers externes, et de vos plugins (si vous avez créé votre site sur un CMS, par exemple).

C'est vrai, y'a du boulot, mais avec un peu de méthode,on arrive à tout !

Choisir son certificat SSL : les différentes sortes de certificats

Voilà les différents types de certificats SSL que vous trouverez sur le marché. Choisissez le vôtre, en fonction de vos besoins :

- Le certificat DV

Ce certificat SSL contrôle le domaine. C'est le niveau de sécurité le plus faible. Dans ce cas, l'organisme de certification ne vérifie que la propriété du domaine concerné. Le certificat DV vous conviendra, si vous ne demandez pas de données privées à vos visiteurs, et si votre site n'est pas la cible potentielle de certaines fraudes, comme le phishing.

- Le certificat OV

Ce certificat effectue un contrôle de l'organisation. L'entreprise est alors vérifiée : on s'assure, par exemple, de l'enregistrement de cette dernière au RCS (registre des commerces et des sociétés). Ce certificat est adapté dans le cas où vous transférez des données non sensibles.

- Le certificat EV

Il s'agit d'un certificat permettant un contrôle étendu de l'entreprise. Avec ce dernier, vous aurez le degré de sécurité le plus fort. C'est le certificat le mieux adapté pour transférer des données sensibles (informations de paiement, par exemple)

normes ISO/IEC 27000

ISO/CEI **27000** est une **norme** de sécurité de l'information publiée conjointement en mai 2009 et révisée en 2012, 2016 et 2018 par l'Organisation internationale de normalisation (**ISO**) et la Commission électrotechnique internationale (CEI, ou **IEC** en anglais), faisant partie de la suite **ISO**/CEI **27000**. …

- Les normes sur le management de la qualité pour travailler plus efficacement et limiter les produits défectueux.
- Les normes sur le management environnemental pour réduire les impacts environnementaux, limiter les déchets et adopter une démarche plus durable.
- Les normes sur la santé et la sécurité pour prévenir les accidents sur le lieu de travail.
- Les normes sur le management de l'énergie pour réduire notre consommation d'énergie.
- Les normes sur la sécurité des denrées alimentaires pour prévenir toute contamination des denrées.
- Les normes sur la sécurité de l'information pour assurer la sécurité des informations sensibles.

variable register_globals en off

**Certains hébergeurs activent la directive register_globals du PHP.INI.
Activée, cette directive permet d'enregistrer les variables super-globales ($_POST, $_GET, $_COOKIE, $_ENV, $_SERVER) dans des variables normales. Dès lors, si cette directive est à on et la variable $_POST['text'] existe, alors la variable $text de même valeur sera automatiquement créée.
Apparemment, cette directive ne pose aucun problème et au contraire, facilite la programmation. Mais en réalité, celle-ci peut être à l'origine de gros soucis de sécurité.
La variable $connected, qui vaut true si l'utilisateur a entré le bon login avec le bon mot de passe, peut être simplement écrasée si register_globals est activé ! Comment ? C'est simple. Il suffit d'appeler la page de connexion ainsi :**

```
index.php?connected=true
```

Dès lors, la variable $_GET['connected'] est copiée dans $connected. Celle-ci aura donc la valeur true avant même que le test de connexion ait été fait.

Le script affiche alors :

```
Information confidentielle : [...]
```

Désactiver register_globals

Pour désactiver register_globals, c'est simple.

Il suffit, dans un premier temps, de vérifier (avec ini_get()) que register_globals est activé.

Si c'est le cas, alors il faut éditer le PHP.INI et changer la ligne

Si l'accès au php.ini vous est restreint, alors vous pouvez utiliser un .htaccess que vous placerez à la racine de votre site avec la ligne suivante :

```
php_flag register_globals off
```

display_error=off et log_errors=on

Désactiver l'affichage des erreurs pour éviter d'afficher des informations aux utilisateurs en mettant la variable *display_error* en *off*. Il vaut mieux les inscrire dans un journal d'erreurs avec *log_errors=on*.

magic_quotes_gpc=on

Ce paramètre, lorsqu'il est mis sur *on*, ajoute le caractère "\" devant les apostrophes, les guillemets et le caractère nul. Il empêche ainsi le système d'interpréter une requête qu'un utilisateur mal intentionné aurait saisi dans un

formulaire HTML. Cette variable tend aujourd'hui à ne plus être utilisée, au profit de la fonction *addslashes()* pour les requêtes SQL.

5) Changer le répertoire temporaire des identifiants de session

Les identifiants de session sont enregistrés dans un répertoire temporaire. Par défaut, le paramètre *session.save_path* vaut */tmp* est accessible en lecture à tous. Il est donc plus sécurisé d'indiquer le chemin d'un répertoire situé ailleurs sur le système, et dont les droits auront été limités. De plus, il existe une fonction en PHP pour crypter les mots de passe enregistrés. Cette modification peut aussi se faire à partir du fichier de configuration d'Apache, à l'aide de la variable *php_value session.save_path*.

6) Activer session.use_only_cookies

Cette variable indique si les identifiants de session doivent être utilisés seulement avec des cookies. Par défaut, cette variable est à *0*, elle est désactivée et autorise d'autres modes de lecture, par exemple avec les éléments *GET* ou *POST* des requêtes HTTP. En mettant *session.use_only_cookies* à *1*, le système lit les informations d'identification uniquement à partir des cookies.

cross site

Le Cross Site Scripting - XSS

Définition

L'attaque XSS (pour Cross Site Scripting) est une attaque très populaire presque au même titre que l'injection SQL. Elle est également présente dans le podium (à dix marches) d'OWASP en 2017.

L'attaque XSS vise comme cible le client plutôt que le serveur. Elle se sert d'un script Javascript qui sera exécuté chez le client pour détourner le fonctionnement de son navigateur. En effet, le pirate développe un script Javascript selon ses attentions, il soumet ensuite ce script comme étant une chaîne de caractères à un serveur via une de ses entrées (formulaire, URL...). Si le serveur présente une vulnérabilité vis-à-vis du XSS, alors le script sera accepté et probablement déposé dans la base de données (ou autre forme de source de données). Jusqu'ici il ne se passe rien de spécial. Mais imaginez qu'à un certain moment un client se connecte sur le serveur et demande une page qui affiche les entrées de la base de données, et par hasard c'est le contenu Javascript qui sera envoyé au navigateur. Puisque Javascript est un langage côté client, alors il sera aussitôt exécuté sur le navigateur du client et fera ce qui a été demandé par le pirate.

Dans ce cas de figure, n'importe quel client peut être victime de cette attaque. Tout dépend de qui a demandé l'affichage du

contenu de la base de données qui coïncide avec le script. Par contre, si le pirate veut viser un client en particulier alors il peut lui envoyer un message privé encapsulant une attaque XSS via un site Web (présentant la vulnérabilité) comme un forum de discussion ou autre.

Une attaque XSS peut également provoquer des dommages parfois conséquents comme par exemple:

- Rediriger un utilisateur à son insu vers un site pirate ou site compromettant
- Afficher des messages indésirables sur le navigateurs du client
- Empêcher l'exécution normale des scripts embarqués dans la page
- Ordonner le déclenchement de périphériques sur l'ordinateur de la victime comme la Webcam

Il y a quelques années, les pirates pouvaient même voler les cookies de la victime à l'aide de l'attaque XSS et usurper ainsi son identité. Heureusement, une telle action n'est plus possible.

Exploitation

Comme le principe est de poser un script javascript sur le serveur et attende à ce qu'un client le charge et l'exécute à

son insu, alors le pirate commence par écrire un bout de code javascript.

Exemple:

```
<script>

window.location="http://www.site_dangereux_et_compromettant.com";

</script>
```

Ensuite, le pirate dépose ce code sur une des entrées d'un site Web (qui est supposé contenir la vulnérabilité) puis poste le tout. Désormais le code est présent sur la base de données et ne fait rien.

Il suffit maintenant qu'un client (la victime) se connecte au site et demande l'affichage des entrées (ou quelques-unes) de la base de données. Si le script fait partie des entrées demandées, le navigateur du client sera immédiatement redirigé vers le site pirate.

Comment s'en protéger?

Les mesures de sécurité que nous allons voir seront implémentées au niveau du serveur et non pas sur le client.

Donc, si le serveur est sécurisé contre les attaques XSS, ses clients n'auront pas à affronter des ennuis comme ce qui a été expliqué dans le paragraphe précédent.

Au niveau du code PHP

Comme pour les injections SQL, l'attaque XSS est due à des entrées provenant de l'extérieur (donc non fiables). La solution consiste donc à filtrer les entrées de l'utilisateur en appliquant les fonctions comme addslashes() ou strip_tags() qui supprime toutes les balises contenues dans la chaîne entrée. On peut aussi formater les mots-clés HTML à l'aide de la fonction htmlentities() ou htmlspecialchars(). Cependant, une chaîne de caractères faisant office d'un script XSS est généralement longue, il faut alors n'autoriser qu'une certaine longueur maximale pour les entrées en la vérifiant à l'aide de la fonction strlen() ou en la tronquant systématiquement à l'aide la fonction substr().

Au niveau de la configuration du serveur

Comme pour l'injection SQL, on peut activer la directive magic_quotes_gpc dans le fichier php.ini pour échapper automatiquement tous les caractères spéciaux (notamment les simples et doubles quotes) figurants dans les chaînes provenant de l'extérieur. Bien que ce n'est pas suffisant, cette solution apportera un peu d'aide quand même.

2) Journaliser toutes les erreurs

Le journal d'erreur est un bon indicateur pour repérer les attaques. Pour enregistrer toutes les erreurs d'exécution, ajouter la ligne *<?php error_reporting(E_ALL); ?>* au début de chaque page de code.

Des outils comme les reverse proxy permettent d'écarter certaines requêtes faites sur le site (par exemple une requête demandant un chemin qui n'existe pas).

Utiliser un pare-feu pour bloquer les connexions sortantes depuis le serveur Web, afin d'éviter l'inclusion de fichiers PHP distants (php include).

Pour une application manipulant des données sensibles nécessitant un haut niveau de sécurité, un serveur dédié sera plus adapté car la configuration sera totalement personnalisable.