Sécurisation d'un site Web

TD : Sécurisation d'un site Web

Date	Auteur(s)	Observations
16/05/24	Besse Jules	N/A

Sommaire

1. Étude des failles de sécurité web courantes

1.1 Qu'est-ce qu'une injection SQL et comment s'en protéger ?

Les pirates informatiques utilisent une injection SQL pour exploiter les failles de sécurité des applications web qui interagissent avec des bases de données. L'objectif principal d'une injection SQL est d'injecter du code SQL malveillant dans les entrées de données de l'application. Cela peut permettre à l'attaquant de contrôler la base de données, d'accéder à des données sensibles, de modifier ou supprimer des données, voire de compromettre tout le système.

Pour se protéger contre les injections SQL, voici quelques techniques à suivre :

- Il est recommandé d'utiliser des requêtes paramétrées ou des procédures stockées avec des paramètres pour interagir avec la base de données plutôt que d'incorporer directement des valeurs d'entrée dans les requêtes SQL. Cela permet de lutter contre les attaques en séparant les données des commandes SQL.

1.2 Expliquez le principe des attaques XSS (Cross-Site Scripting). Quels en sont les différents types ?

Les attaques XSS, ou Cross-Site Scripting, sont des attaques qui visent à injecter du code malveillant, généralement du code JavaScript, dans des pages web consultées par d'autres utilisateurs. Le principe est assez simple :

1. **Injection de code malveillant** : Un attaquant insère du code malveillant, souvent du code JavaScript, dans un champ de saisie accessible sur un site web. Cela peut être un champ de formulaire, une zone de commentaire, ou tout autre champ interactif où les utilisateurs peuvent entrer du texte.
2. **Traitement non sécurisé des données utilisateur** : Si le site web ne filtre pas ou ne valide pas correctement les données entrées par l'utilisateur, le code malveillant est alors stocké sur le serveur et affiché à d'autres utilisateurs qui consultent la page contenant ces données.
3. **Exécution du code côté client** : Lorsque d'autres utilisateurs consultent la page web contaminée, le code malveillant est exécuté dans leur navigateur. Cela peut entraîner diverses conséquences nuisibles, telles que le vol de cookies de session, la redirection vers des sites frauduleux, la capture de données sensibles entrées sur la page, ou encore la manipulation du contenu de la page.

Il existe plusieurs types d'attaques XSS :

- **XSS stocké (Stored XSS)** : Le code malveillant est stocké sur le serveur et est ensuite affiché à tous les utilisateurs qui consultent la page infectée.
- **XSS réfléchi (Reflected XSS)** : Le code malveillant est inclus dans une URL et est renvoyé à la victime par le serveur dans la réponse à sa requête. Ce type d'attaque peut être exploité en incitant les utilisateurs à cliquer sur un lien contenant le code malveillant.
- **XSS basé sur le DOM (DOM-based XSS)** : Le code malveillant est exécuté côté client par manipulation du DOM (Document Object Model) de la page. Il est souvent plus difficile à détecter et à exploiter, car il n'implique pas nécessairement de communication avec le serveur.

Pour se protéger contre les attaques XSS, les développeurs web doivent mettre en place des mesures de sécurité telles que la validation et l'encodage appropriés des données utilisateur, l'utilisation de listes blanches pour les entrées utilisateur, l'échappement des caractères spéciaux et l'utilisation de CSP (Content Security Policy) pour restreindre les sources autorisées de contenu exécutable.

https://guardia.school/boite-a-outils/en-quoi-consiste-une-attaque-par-xss.html

1.3 Pourquoi est-il important de bien configurer les permissions d'accès aux fichiers sur un serveur web ?

Il est crucial de bien configurer les permissions d'accès aux fichiers sur un serveur web pour plusieurs raisons :

1. **Sécurité des données**: Des permissions inappropriées peuvent permettre à des utilisateurs non autorisés d'accéder à des fichiers sensibles, tels que des bases de données, des fichiers de configuration contenant des informations sensibles, ou des fichiers de sauvegarde. Cela peut entraîner des fuites de données, des violations de la confidentialité et des attaques potentielles.
2. **Protection contre les attaques**: Une mauvaise configuration des permissions peut rendre un serveur web vulnérable à diverses attaques, telles que l'exécution de code malveillant, le détournement de sessions, l'injection SQL et les attaques par inclusion de fichiers. Des permissions correctement configurées peuvent limiter la surface d'attaque et rendre plus difficile l'exploitation des vulnérabilités.
3. **Conformité réglementaire**: Dans de nombreux cas, les entreprises sont tenues de se conformer à des réglementations strictes en matière de protection des données, telles que le RGPD en Europe ou la HIPAA aux États-Unis. Une mauvaise configuration des permissions peut entraîner des violations de ces réglementations et des conséquences légales et financières importantes.

4. **Intégrité des fichiers** : Des permissions appropriées peuvent garantir que seules les personnes autorisées peuvent modifier ou supprimer des fichiers sur le serveur web. Cela protège l'intégrité des données et prévient les dommages accidentels ou malveillants.

https://www.vaadata.com/blog/fr/comment-securiser-un-serveur/

1.4 Qu'est-ce que le clickjacking et comment le prévenir ?

Le clickjacking est une technique utilisée par les attaquants pour tromper les utilisateurs en cliquant sur des éléments d'une page web sans qu'ils en aient conscience. Cela se fait généralement en superposant de manière invisible des éléments cliquables sur une page web légitime, de sorte que lorsque l'utilisateur clique sur ce qu'il pense être un élément de la page légitime, il clique en réalité sur un élément malveillant.

Voici comment cela fonctionne généralement :

1. L'attaquant crée une page web malveillante qui contient des éléments superposés invisibles, comme des boutons ou des liens, positionnés au-dessus de la page web légitime qu'il souhaite cibler.
2. L'utilisateur est incité à visiter la page malveillante, soit par le biais d'un lien, soit en étant redirigé automatiquement vers cette page.
3. Lorsque l'utilisateur interagit avec la page malveillante, par exemple en cliquant sur un bouton ou en effectuant une action quelconque, il interagit en réalité avec les éléments superposés invisibles de la page légitime, déclenchant ainsi des actions indésirables.

Pour prévenir le clickjacking, voici quelques mesures que les développeurs peuvent prendre :

1. **Utiliser le Header X-Frame-Options** : Ce header HTTP permet de contrôler si une page peut être chargée dans un cadre (iframe). En utilisant cette en-tête avec la valeur "DENY" ou "SAMEORIGIN", les développeurs peuvent empêcher leur site web d'être chargé dans un cadre invisible sur une autre page.
2. **Utiliser Content Security Policy (CSP)** : CSP est un mécanisme de sécurité qui permet aux sites web de spécifier les sources de contenu légitimes et de restreindre les types de contenu qui peuvent être chargés sur leurs pages. En définissant des directives CSP appropriées, les développeurs peuvent prévenir le chargement de leur site dans des cadres invisibles.
3. **Éviter les cadres invisibles** : Les développeurs doivent éviter de placer des éléments interactifs, tels que des boutons ou des liens, à des positions prévisibles sur la page qui pourraient être superposées par des éléments malveillants.

4. **Éducation des utilisateurs** : Sensibiliser les utilisateurs aux risques de clickjacking et les encourager à être prudents lorsqu'ils interagissent avec des pages web, en particulier lorsqu'ils sont redirigés depuis des sources non fiables.

https://www.kaspersky.fr/resource-center/definitions/clickjacking

1.5 Décrivez le fonctionnement d'une attaque par déni de service distribué (DdoS).

Une attaque par déni de service distribué (DDoS) est une forme d'attaque informatique dans laquelle un grand nombre de systèmes informatiques compromis, souvent appelés "botnets", sont utilisés pour saturer les ressources d'un serveur cible, le rendant ainsi inaccessible aux utilisateurs légitimes. Voici comment fonctionne généralement une attaque DDoS :

1. **Recrutement de botnets** : Les attaquants compromettent un grand nombre de systèmes informatiques à travers Internet. Ces systèmes peuvent être des ordinateurs personnels, des serveurs, des dispositifs IoT (Internet des Objets) ou d'autres appareils connectés à Internet. Les attaquants installent souvent un logiciel malveillant sur ces systèmes, transformant ainsi ces machines en "zombies" ou "bots" contrôlés à distance.
2. **Coordination des bots** : Une fois qu'un grand nombre de systèmes ont été compromis, les attaquants les coordonnent pour qu'ils envoient simultanément des requêtes au serveur cible. Ces requêtes peuvent être des demandes de connexion TCP, des requêtes HTTP, ou d'autres types de requêtes réseau, selon le type d'attaque DDoS utilisé.
3. **Saturation des ressources du serveur cible** : Le serveur cible est submergé par le volume massif de requêtes provenant des bots. Les ressources du serveur, telles que la bande passante, la capacité de traitement et la mémoire, deviennent rapidement épuisées, ce qui entraîne une diminution des performances du serveur voire son indisponibilité totale.
4. **Impact sur les utilisateurs légitimes** : L'objectif principal d'une attaque DDoS est de rendre un service indisponible pour ses utilisateurs légitimes. Les utilisateurs légitimes ne peuvent pas accéder au site web ou au service visé, ce qui peut entraîner des pertes financières, une atteinte à la réputation de l'entreprise et d'autres conséquences néfastes.

Les attaques DDoS peuvent être lancées pour diverses raisons, notamment pour extorquer de l'argent, pour protester contre une entreprise ou une organisation, pour perturber des services critiques, ou même pour servir de diversion pendant d'autres attaques plus sophistiquées.

Pour se protéger contre les attaques DDoS, les entreprises peuvent mettre en place des solutions de détection et de mitigation des attaques DDoS, telles que des pare-feu, des systèmes de détection d'intrusion, des services de filtrage du trafic, des solutions de mitigation en cloud, et des services de redirection de trafic.

https://www.cloudflare.com/fr-fr/learning/ddos/glossary/denial-of-service/

1.6 En quoi consiste la désérialisation non sécurisée d'objets et quel risque cela pose-t-il ?

La désérialisation non sécurisée d'objets est une vulnérabilité de sécurité qui survient lorsqu'une application désérialise des données provenant d'une source non fiable sans effectuer de validation adéquate sur ces données. Cette vulnérabilité peut être exploitée par des attaquants pour exécuter du code malveillant sur le système cible.

Voici comment cela fonctionne et les risques associés :

1. **Désérialisation** : La désérialisation est le processus de conversion de données structurées, généralement au format JSON, XML ou binaire, en objets utilisables dans un langage de programmation. Cette opération est couramment utilisée pour échanger des données entre différentes applications ou pour stocker des données dans des formats persistants.
2. **Source non fiable** : Lorsque des données désérialisées proviennent d'une source non fiable, telles qu'un utilisateur externe, une entrée utilisateur ou un flux de données réseau, elles peuvent être manipulées par un attaquant pour inclure du code malveillant ou des instructions dangereuses.
3. **Validation insuffisante** : Si l'application ne vérifie pas correctement les données désérialisées pour s'assurer qu'elles sont sûres et conformes aux attentes, elle peut être vulnérable à des attaques d'injection de code.
4. **Risques associés** : Les risques associés à la désérialisation non sécurisée incluent l'exécution de code arbitraire sur le serveur ou sur le client, le contournement des mécanismes de sécurité, la divulgation de données sensibles, la corruption de données ou même la prise de contrôle complet du système.

Pour exploiter cette vulnérabilité, un attaquant peut fournir des données spécialement conçues pour exploiter les failles de sécurité dans le processus de désérialisation. Cela peut inclure l'inclusion de charges utiles malveillantes, telles que des commandes système, des scripts malveillants ou des objets Java sérialisés (pour les applications Java).

Pour prévenir les attaques de désérialisation non sécurisées, il est essentiel de mettre en œuvre des pratiques de sécurité robustes, telles que :

- Valider et filtrer les données désérialisées pour s'assurer qu'elles ne contiennent pas de code malveillant.
- Utiliser des mécanismes de sécurité, tels que la signature numérique ou le cryptage, pour garantir l'intégrité et l'authenticité des données désérialisées.
- Limiter les privilèges des objets désérialisés en utilisant des contextes d'exécution isolés ou des mécanismes de sandboxing.
- Utiliser des bibliothèques de désérialisation sécurisées et à jour, qui incorporent des protections contre les attaques de désérialisation non sécurisées.

1.7 Qu'appelle-t-on une vulnérabilité de type "XXE" (XML eXternal Entity) ?

Une vulnérabilité de type "XXE" (XML eXternal Entity) survient lorsqu'une application XML, qui analyse des documents XML, permet à un attaquant de charger des entités externes indésirables ou malveillantes lors du traitement d'un document XML.

Voici comment fonctionne une attaque XXE et ce qu'elle peut entraîner :

1. **Analyse du document XML**: L'application analyse un document XML qui inclut des références à des entités externes, généralement définies dans une DTD (Document Type Definition).
2. **Injection d'entités externes malveillantes**: Un attaquant peut manipuler le document XML pour inclure des références à des entités externes contrôlées par l'attaquant, telles que des fichiers locaux ou des ressources réseau.
3. **Traitement des entités externes**: Lorsque le document XML est analysé par l'application, les entités externes sont résolues et leur contenu est inclus dans le document traité.
4. **Risques associés**: Les risques associés à une vulnérabilité XXE comprennent la divulgation de données sensibles, l'exécution de code arbitraire, la déni de service et d'autres types d'attaques basées sur la manipulation des entités externes.

Une attaque XXE peut avoir des conséquences graves, notamment la divulgation de données confidentielles stockées sur le serveur, l'exécution de code malveillant sur le serveur, le contournement des mécanismes de sécurité et même la prise de contrôle total du serveur.

Pour prévenir les attaques XXE, il est recommandé de prendre les mesures suivantes :

1. **Désactiver le support des DTD**: Désactiver le support des DTD dans le processeur XML utilisé par l'application, car les DTD peuvent être utilisées pour définir des entités externes.
2. **Utiliser une bibliothèque XML sécurisée**: Utiliser une bibliothèque XML sécurisée qui prend en charge la désactivation des fonctionnalités potentiellement dangereuses, telles que la résolution des entités externes.
3. **Filtrer les entrées utilisateur**: Valider et filtrer toutes les entrées utilisateur qui sont utilisées dans la construction de documents XML pour éviter l'inclusion d'entités externes non désirées.
4. **Utiliser des mécanismes de sécurité**: Utiliser des mécanismes de sécurité tels que les pare-feu d'application Web (WAF) pour détecter et bloquer les attaques XXE.

1.8 Citez au moins trois bonnes pratiques essentielles pour sécuriser l'authentification sur un site web.

1. **Utiliser des mécanismes d'authentification robustes** : Utilisez des méthodes d'authentification solides telles que le hachage de mot de passe avec des algorithmes sécurisés (comme bcrypt ou Argon2), le chiffrement SSL/TLS pour sécuriser les communications entre le navigateur et le serveur, et l'implémentation de politiques de mots de passe strictes (longueur minimale, complexité, expiration, etc.).
2. **Mettre en œuvre la gestion des sessions et des cookies de manière sécurisée** : Utilisez des sessions sécurisées avec des identifiants de session aléatoires et un délai d'expiration approprié. Assurez-vous que les cookies de session sont marqués comme sécurisés et HTTPOnly pour empêcher les attaques de type session hijacking ou session fixation. Évitez de stocker des informations sensibles dans les cookies et assurez-vous que les cookies sont protégés contre les attaques de falsification de cookies.
3. **Protéger contre les attaques par force brute et les attaques par injection** : Mettez en place des mécanismes de protection contre les attaques par force brute, comme le verrouillage du compte après un certain nombre de tentatives de connexion infructueuses. Utilisez des techniques de prévention des injections SQL (comme les requêtes préparées ou les ORMs) pour éviter les attaques par injection SQL qui pourraient compromettre les informations d'identification des utilisateurs.

https://www.vaadata.com/blog/fr/comment-securiser-les-systemes-dauthentification-de-gestion-de-sessions-et-de-controle-dacces-de-vos-applications-web/

1.9 Quelles sont les étapes clés d'une gestion saine des mises à jour et correctifs de sécurité ?

La gestion saine des mises à jour et correctifs de sécurité est essentielle pour maintenir un environnement informatique sécurisé. Voici les étapes clés de ce processus :

1. **Identification des vulnérabilités** : Surveillez les annonces de vulnérabilités de sécurité dans les logiciels que vous utilisez, que ce soit à travers les bulletins de sécurité des fournisseurs, les bases de données de vulnérabilités publiques, les listes de diffusion spécialisées ou d'autres sources d'information fiables.
2. **Évaluation de l'impact** : Évaluez l'impact des vulnérabilités identifiées sur votre environnement informatique. Déterminez si elles affectent les systèmes critiques ou sensibles, et évaluez le risque potentiel pour votre organisation en cas d'exploitation de ces vulnérabilités.

3. **Planification des mises à jour** : Élaborez un plan de gestion des mises à jour et correctifs en fonction de l'urgence et de l'importance des vulnérabilités. Classez les correctifs en fonction de leur criticité et planifiez leur déploiement en conséquence, en tenant compte des contraintes de temps et des interruptions potentielles pour les opérations commerciales.
4. **Test des correctifs** : Testez les correctifs dans un environnement de test pour vous assurer qu'ils n'entraînent pas de problèmes de compatibilité, de performances ou de fonctionnalité avec vos applications ou vos systèmes. Assurez-vous que les correctifs fonctionnent comme prévu et ne créent pas de nouvelles vulnérabilités.
5. **Déploiement des correctifs** : Déployez les correctifs sur l'ensemble des systèmes affectés dès que possible, en suivant les procédures de déploiement appropriées. Utilisez des outils de gestion des correctifs pour automatiser et faciliter ce processus, en veillant à ce que tous les correctifs nécessaires soient appliqués de manière complète et cohérente.
6. **Suivi et vérification** : Suivez attentivement le déploiement des correctifs et assurez-vous qu'ils sont bien installés sur tous les systèmes concernés. Vérifiez régulièrement l'état de conformité des correctifs et effectuez des audits de sécurité pour vous assurer que votre environnement reste sécurisé après leur déploiement.
7. **Formation et sensibilisation** : Sensibilisez les utilisateurs et le personnel informatique à l'importance des mises à jour et correctifs de sécurité. Fournissez une formation sur les bonnes pratiques en matière de gestion des correctifs et les risques associés à l'exploitation des vulnérabilités non corrigées.

https://www.ninjaone.com/fr/blog/bonnes-pratiques-de-gestion-des-correctifs-de-serveur/

1.10 Parmi ces failles, lesquelles concernent plus spécifiquement les sites WordPress ?

Les sites WordPress sont sujets à plusieurs types de failles de sécurité, mais voici quelques-unes qui sont plus spécifiques à cette plateforme :

1. **Vulnérabilités des plugins et thèmes**: Les plugins et thèmes tiers peuvent contenir des failles de sécurité qui peuvent être exploitées par les attaquants pour compromettre un site WordPress. Cela peut inclure des vulnérabilités telles que des injections SQL, des failles XSS (Cross-Site Scripting), des problèmes de gestion des sessions, ou d'autres types de vulnérabilités.
2. **Failles de gestion des utilisateurs**: Les failles de gestion des utilisateurs, telles que les faiblesses dans les mécanismes d'authentification, les mots de passe faibles, l'absence de limites sur les tentatives de connexion, ou les autorisations mal configurées, peuvent

permettre à des attaquants d'accéder illicitement à des comptes d'utilisateurs ou d'effectuer des actions non autorisées.

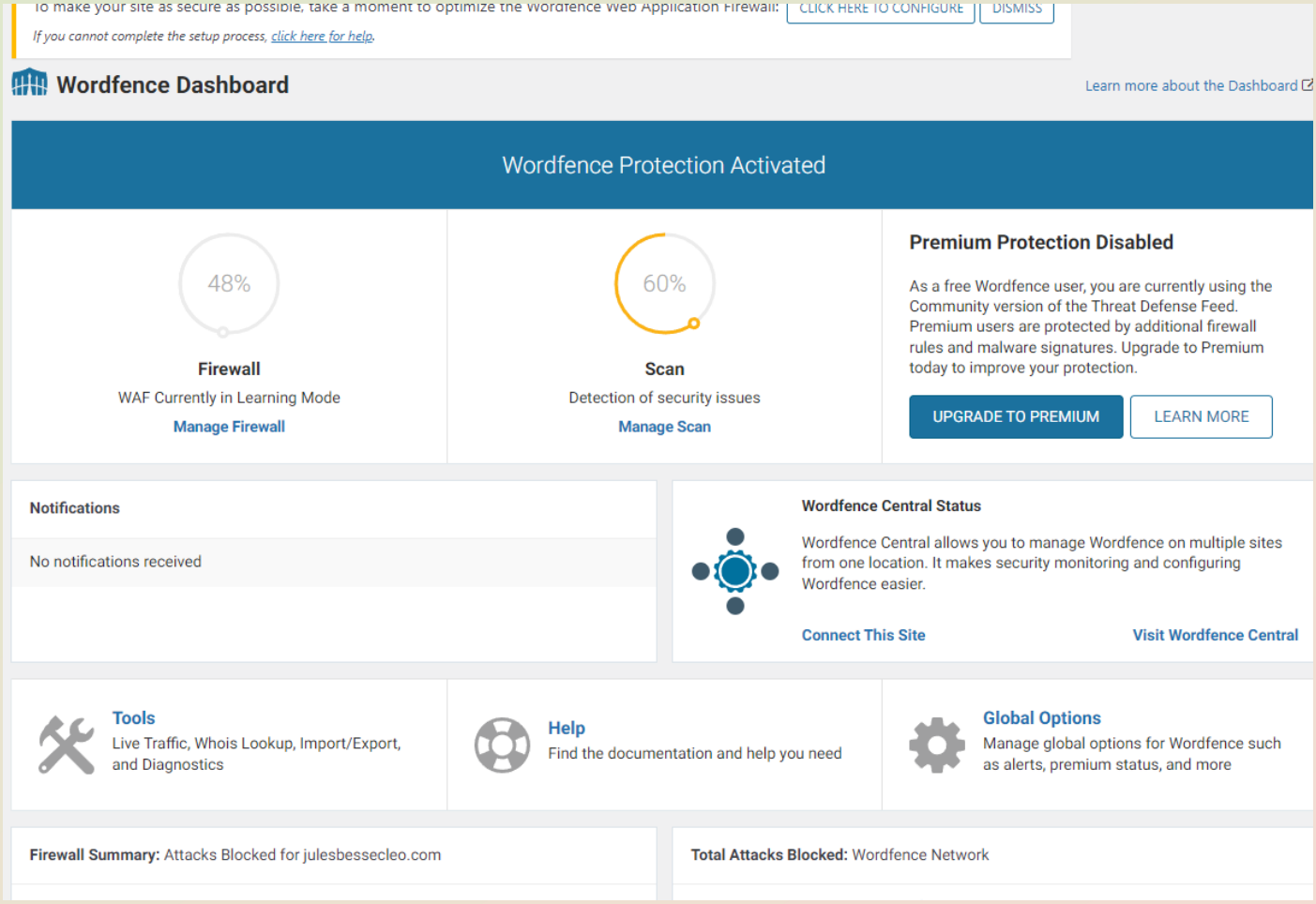
3. **Vulnérabilités de configuration** : Des erreurs de configuration dans WordPress, telles que l'activation de fonctionnalités non nécessaires, la mauvaise gestion des permissions de fichiers et répertoires, ou l'absence de restrictions sur l'accès aux fichiers sensibles, peuvent créer des points d'entrée potentiels pour les attaquants.

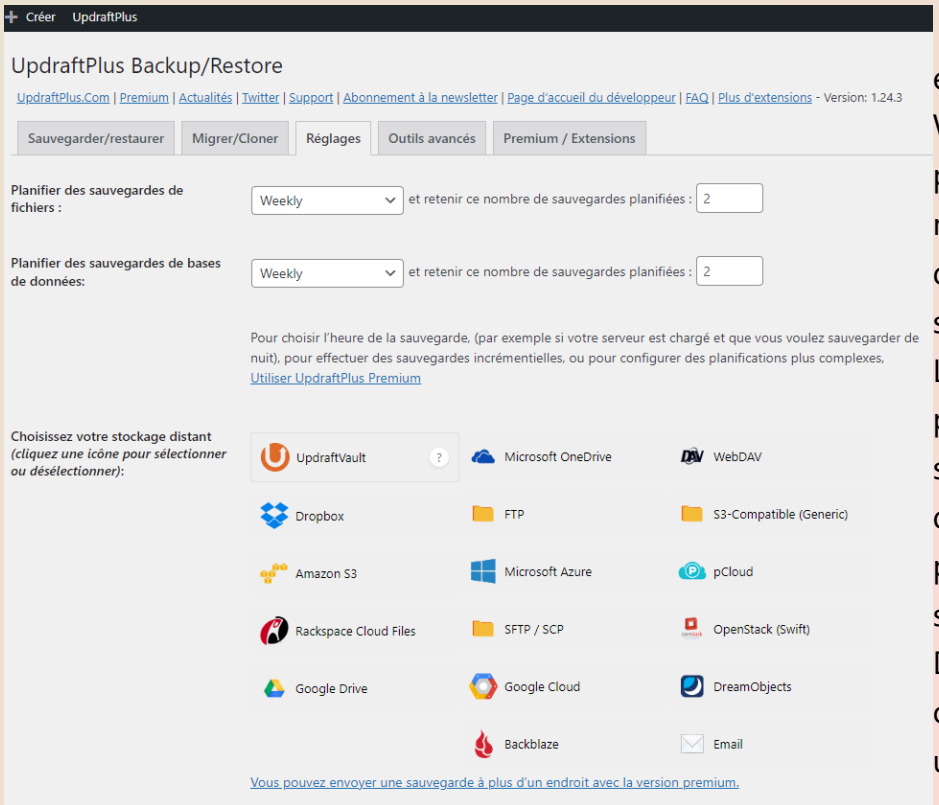
4. **Failles de sécurité dans le noyau WordPress** : Bien que WordPress soit généralement bien maintenu et mis à jour régulièrement, il peut également contenir des failles de sécurité dans son propre code. Ces failles peuvent être exploitées par les attaquants pour compromettre un site WordPress.

5. **Attaques par force brute** : Les attaques par force brute contre les comptes d'utilisateur de WordPress sont courantes. Les attaquants essaient de deviner les mots de passe en essayant différentes combinaisons de mots de passe jusqu'à ce qu'ils réussissent à accéder au compte.

3. Mise en place d'extensions de sécurité

J'ai choisi Wordfence pour sécuriser mon portfolio car je sais que la sécurité en ligne est cruciale de nos jours. En optant pour ce plugin, je protège mon site contre une multitude de menaces, des attaques par force brute aux injections SQL. La fonctionnalité de surveillance en temps réel me permet de rester vigilant et d'être alerté instantanément en cas d'activité suspecte. De plus, le pare-feu d'application Web (WAF) intégré bloque automatiquement les tentatives d'exploitation des vulnérabilités, assurant ainsi une protection proactive. Avec Wordfence, je peux également analyser régulièrement les vulnérabilités de mon site et prendre des mesures pour les corriger.

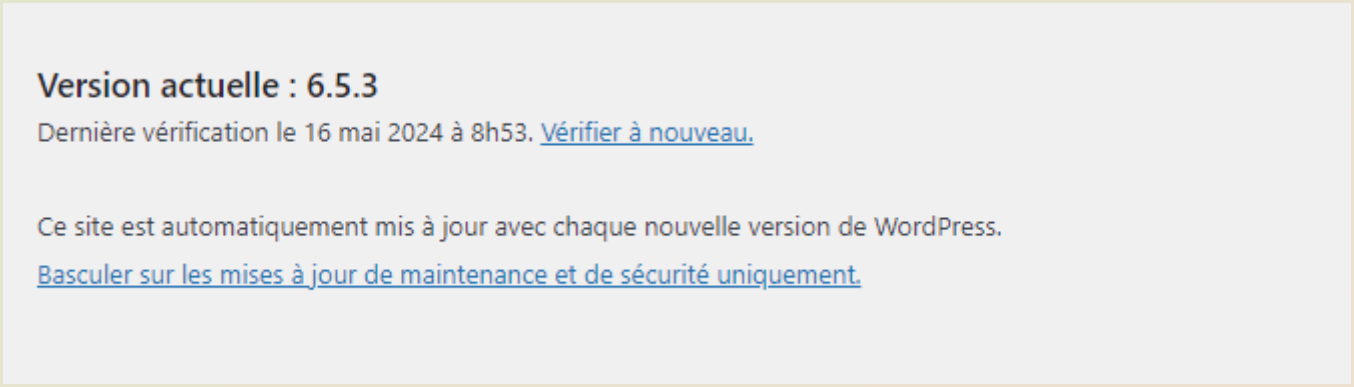




J'ai choisi UpdraftPlus pour sauvegarder et restaurer facilement mon site WordPress. Avec ce plugin, je peux planifier des sauvegardes régulières de mon site et de ma base de données, ce qui garantit que mes données sont sécurisées en cas de problème. L'interface conviviale d'UpdraftPlus me permet de configurer rapidement mes sauvegardes et de les restaurer en quelques clics seulement. De plus, la possibilité de stocker mes sauvegardes sur différents services cloud, tels que Dropbox, Google Drive ou Amazon S3, offre une redondance supplémentaire et une tranquillité d'esprit.

4. Durcissement de la configuration

- Mettez à jour WordPress, le thème et les extensions, mettre en place des alertes par mails dès qu'une mise à jour est possible



- Mettre votre site en HTTPS si ce n'est pas déjà le cas ✔

- Changez votre identifiant et mot de passe pour quelque chose de robuste ✔

- Supprimez le compte admin générique s'il existe

- Désinstallez les extensions et les thèmes inutilisés ✔

- Limitez les tentatives de connexion échouées ✔

5. Surveillance

- Regardez les logs d'activité, les IP, les pages visitées (si c'est entre deux séances, il se peut que des visites aient eu lieu)
- Identifiez des comportements anormaux s'il y en a ✔
- Configurez les notifications par email en cas d'événement suspect (si les extensions en place le permettent) ✔

6. Politique de sécurité

Rappel de l'Importance de la Sécurité :

La sécurité personnelle est une priorité absolue dans notre monde connecté. Protéger nos informations, nos biens et notre identité numérique est essentiel pour prévenir les menaces en ligne et hors ligne.

Bonnes Pratiques à Suivre

Mots de passe Robustes :

- Utilisez des mots de passe complexes, comprenant des combinaisons de lettres, de chiffres et de caractères spéciaux.
- Évitez d'utiliser les mêmes mots de passe pour plusieurs comptes.
- Changez régulièrement vos mots de passe, surtout en cas de suspicion de compromission.

Mises à Jour :

- Assurez-vous que vos appareils (ordinateurs, smartphones, etc.) et vos logiciels sont toujours à jour avec les dernières versions de sécurité.
- Activez les mises à jour automatiques pour garantir une protection constante contre les vulnérabilités.

Sauvegarde :

- Effectuez régulièrement des sauvegardes de vos données importantes sur des périphériques externes ou dans le cloud.
- Vérifiez périodiquement la validité de vos sauvegardes pour vous assurer qu'elles sont accessibles en cas de besoin.

Conduite à tenir s'il y a un incident

Restez Calme :

- Gardez votre calme et ne paniquez pas. Une réaction réfléchie est essentielle pour résoudre efficacement la situation.

Identifiez les Signes :

- Soyez attentif aux signes d'une éventuelle intrusion ou d'une activité suspecte, tels que des transactions non autorisées ou des comportements inhabituels sur vos comptes.

Prenez des Mesures Immédiates :

- Changez immédiatement les mots de passe compromis.
- Contactez les autorités compétentes et signalez l'incident, en fournissant autant de détails que possible.
- Si nécessaire, bloquez l'accès à vos comptes ou à vos appareils pour limiter les dommages potentiels.

En conclusion, la sécurisation d'un site Web est une responsabilité essentielle pour protéger les données, garantir la confidentialité des utilisateurs et maintenir la confiance envers votre plateforme en ligne. En adoptant des mesures proactives, en restant informé des dernières menaces et en mettant en œuvre des pratiques de sécurité robustes, vous pouvez assurer une expérience en ligne sûre et sécurisée pour tous les utilisateurs.