VEILLE TECHNOLOGIQUE

LA SÉCURITE DES DONNÉES WEB POUR UN DÉVELOPPEUR

Par
CHANG Toma

Table des matières

Qu’est-ce qu’une application Web ?

- Typologies des applications Web
- Architecture

Les principales failles de sécurité des applications

- OWASP
- Top 10 des risques de sécurité des applications
- Les risques des failles

Solutions à ses attaques

- Méthodologie et techniques pour se protéger
- Outils de contrôle de la sécurité des applications

- Sources

L'OWASP A-T-IL UN IMPACT IMPORTANT ?

Introduction

Une **veille technologique** vise à une surveillance active permettant de rester à jour dans un domaine précis.

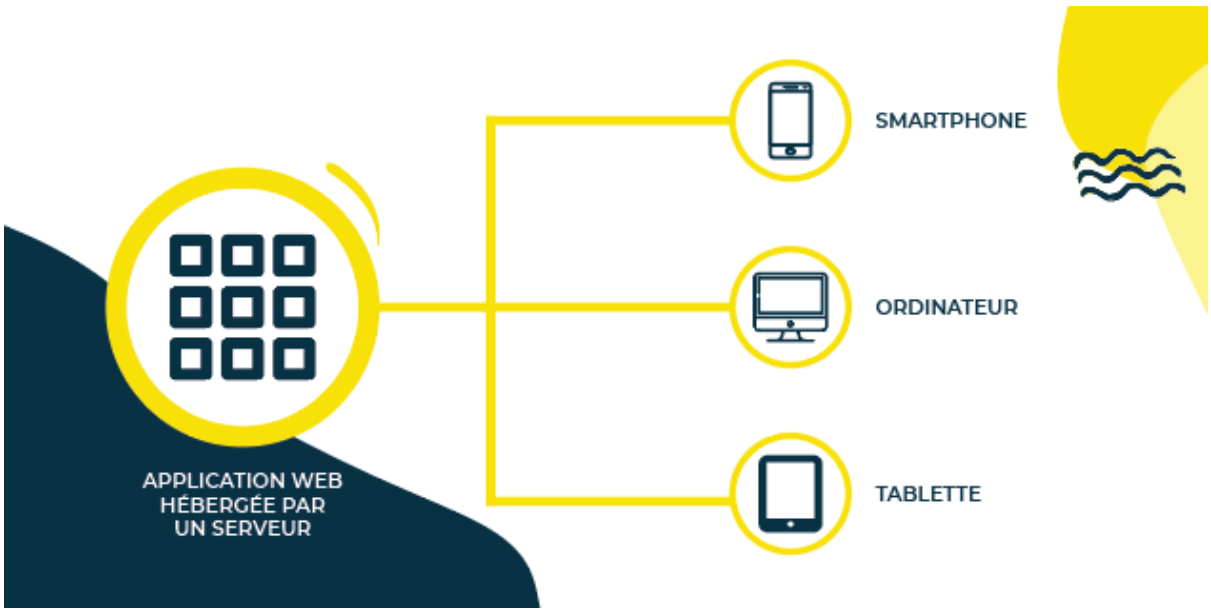
Un site Web est devenu très accessible, depuis quelques années. Un domaine qui se développe très vite, mais très peu sont informés sur les vulnérabilités présentes dans leurs réalisation (Ou aussi appelées « failles »), un terme qui peut s'avérer très dangereux.

Une faille est une faiblesse dans un code qui pourrait être exploitée pour détourner un site de sa fonction première. Le « Hacker » ou « Pirate » pourra récupérer des données confidentielles ou encore modifier l'intégralité d'un site Web. Mais heureusement, il est possible de se protéger de ces fameuses failles par différentes techniques.

I. Qu'est-ce qu'une applications Web ?

A. Typologie des applications Web

Une application Web est une extension dynamique d'un serveur Web. Une application web est formée d'un ensemble de composants web, de ressources statiques (sons, Images, vidéo, …), de classes utilitaires et de bibliothèques.



Il existe deux types d'applications Web :

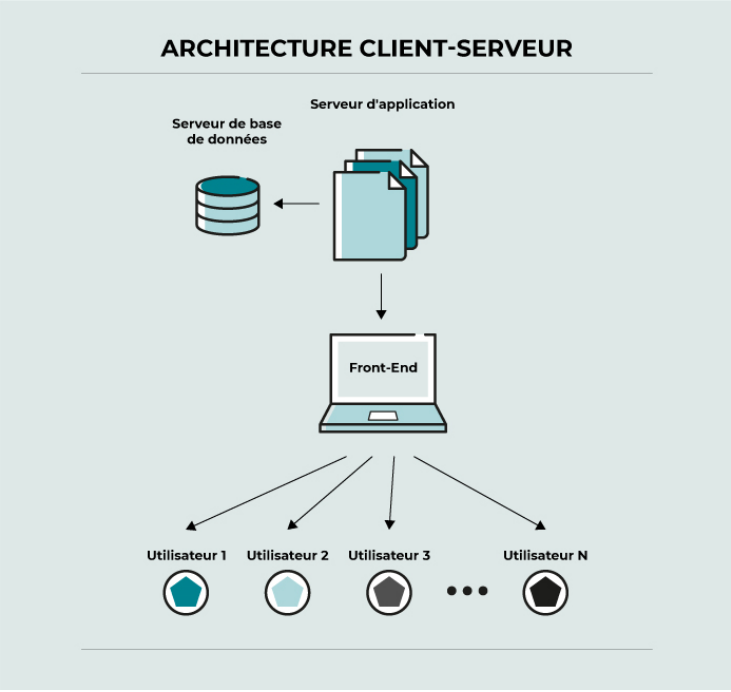
- **Applications de présentation** : Générant des pages contenant différents types de langage (HTML, XML, …). Les pages générées peuvent être statiques (HTML) ou dynamiques (Pages qui contient du code exécuté sur le serveur).

- **Application de service** : Un point d'accès à un service via le Web qui est souvent invoqué par une application de présentation. Il se traduit donc en service métier en relation B-to-B.

B. Architecture

L'architecture est divisée en 3 grandes parties :

- **Couche de présentation** : Cette partie permet la présentation des données et de l'interaction via l'utilisateur (navigateur Web)
- **Couche métier :** Cette couche permet de recevoir les requêtes de l'utilisateur, c'est là qu'est implantée la logique / cerveau du système et ses règles de gestion. Ça permet aussi de gérer et protéger l'accès direct par les utilisateurs.
- **Couche d'accès aux données :** Elle est responsable de la gestion de données, stocker en fichier ou encore dans une base de données.



Cette architecture met en avant la structuration logique. (Ces 3 couches de s'exécuter sur une même machine)

II. Les principales failles de sécurités des applications Web

A. L'OWASP

Source :

- OWASP.org -> https://owasp.org/
- Open Classroom -> https://openclassrooms.com/fr/courses/6179306-securisez-vos-applications-web-avec-lowasp
- SonarQube -> https://www.sonarqube.org/features/security/owasp/?gads_campaign=Europe-4-Generic&gads_ad_group=OWASP&gads_keyword=owasp&gclid=EAIaIQobChMIwPKm0PvB-gIVzo9oCR3jXwCaEAAYAiAAEgKcz_D_BwE

L'OWASP, Open Web Application Security Project est un guide de sécurisation des applications Web, c'est un « Ouvrage » de référence des bonnes et mauvaises pratiques de développement, d'une base sérieuse en termes de statistiques et en termes de ressources amenant à une base de réflexion sur la sécurité.

L'OWASP, est entrée en ligne le 1er Décembre 2001. Elle a été créée en tant qu'organisation caritative à but non lucratif aux États-Unis le 21 Avril 2004 pour assurer la disponibilité et le soutien continu de notre travail à l'OWASP.

Les valeurs fondamentales d'OWASP sont :

- **OUVERT**
- **INNOVATION**
- **GLOBAL**
- **INTÉGRITÉ**

Le but principal de l'OWASP est de permettre à la communauté mondiale d'être informé en stimulent la visibilité et l'évolution de la sureté et de la sécurité des logiciels.

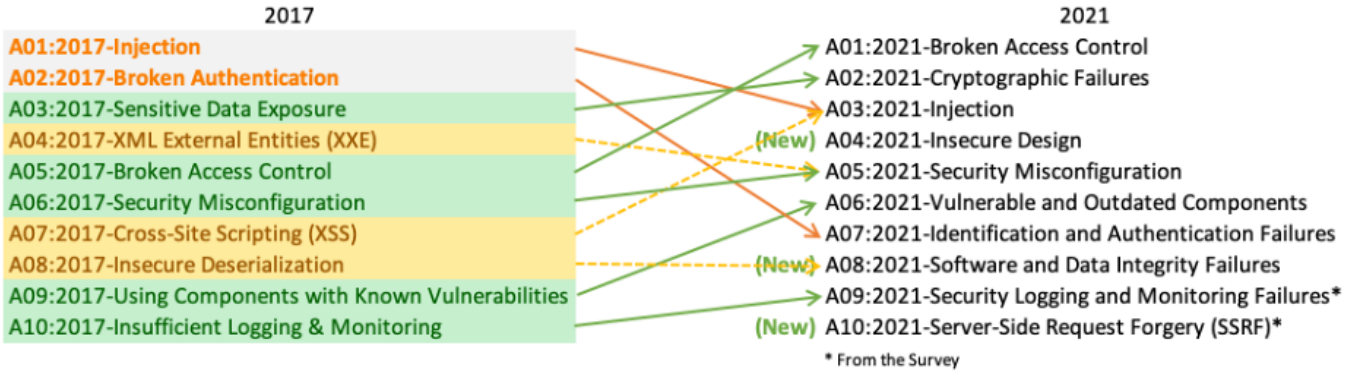
B. Top 10 des risques de sécurité des applications

L'OWASP Top 10 est un document de sensibilisation sur la sécurité des applications Web. Il représente les risques de sécurité mondiale les plus critiques des applications Web. Ces informations proviennent d'une variété d'expert en sécurité du monde entier.

L'adoption de l'OWASP Top 10, serait peut-être la première étape pour changer la culture de développement de logiciels au sein de votre entreprise, afin de prendre en compte les facteurs de sécurité première.

Chaque année l'OWASP met à jour le classement.

Nous avons ci-dessous l'OWASP publié en Novembre 2017 comparé à la plus récente qui est donc celle de 2021 :



A01 : 2021-Broken Access Control

-> Il passe de la cinquième position à la catégorie présentant le risque de sécurité des applications Web le plus grave. Les données fournies démontrent qu'en moyenne 3,81% des applications avaient une ou plusieurs énumérations de faiblesses communes (CWE) avec plus de 318 000 occurrences de CWE.
Cela consistait à un problème d'implémentation du contrôle des accès.

S'en prémunir :

- Tracer les échecs de contrôles d'accès, alerter les administrateurs quand c'est approprié (ex : échecs répétés).
- Centraliser l'implémentation des mécanismes de contrôle d'accès et les réutiliser dans l'ensemble de l'application. Cela comprend de minimiser l'utilisation de CORS.

A02 : 2021 -> Cryptographic Failure (Défaillance cryptographique)

-> De la troisième place il passe à la deuxième place, connu sous le terme « Sensitive Data Exposure ».
Il passe de la troisième position à la deuxième position de l'OWASP 2021, il nous conduit souvent à l'exposition de données sensibles ou à la compromission du système.

S'en prémunir :

- Classifier les données traitées, stockées ou transmises par l'application. Identifier quelles données sont sensibles selon les lois concernant la protection de la vie privée, les exigences réglementaires ou les besoins métier.
- S'assurer de chiffrer toutes les données sensibles au repos

A03 : 2021 -> Injection

-> Tombe à la troisième position. Justifié par 94% des applications ont été évaluées pour une forme d'injection avec un taux d'incidence maximal de 19%, un taux d'incidence moyenne de 3,37% et les 33 CWE cartographiés dans cette catégorie ont le deuxième plus grand nombre d'occurrences dans les applications avec 274 000 occurrences.

S'en prémunir :

- Pour les données en entrée, une liste autorisée avec normalisation est recommandée, mais n'est pas une défense complète dans la mesure où de nombreuses applications requièrent des caractères spéciaux.
- Il est conseillé d'utiliser LIMIT et autres contrôles SQL à l'intérieur des requêtes pour empêcher les divulgations massives de données dans le cas d'injection SQL

A04 : 2021 -> Conception non sécurisée

-> Une nouvelle catégorie insérée dans L'OWASP 2021, concernant les risques liés aux défauts de conception.

S'en prémunir :

- Séparez les tenants via une conception robuste sur l'ensemble des niveaux ;
- Restreignez les ressources par utilisateur ou service.

A05 : 2021 -> La mauvaise configuration de la sécurité

- La quasi-totalité des applications testées ont une très mauvaise configuration, avec un taux d'incidence moyen de 4,5% et plus de 208 000 occurrences de CWE ont été cartographiées.

S'en prémunir :

- Une plate-forme minimale sans fonctionnalité, composant, documentation et échantillon inutile. Supprimer ou ne pas installer des fonctionnalités et Framework inutilisés ;
- Un processus automatisé pour vérifier l'efficacité des configurations et des réglages dans tous les environnements.

A06 : 2021 -> Vulnerable and Outdated components

- Anciennement à la neuvième position il est actuellement maintenant à la sixième place de l'OWASP 2021, et qui concerne les composants qui sont vulnérables ou encore obsolètes.

S'en prémunir :

- Supprimer les dépendances inutiles et les fonctionnalités, composants, fichiers et documentation non nécessaires

A07 : 2021 -> Identification and authentification failures

- Tombe de la deuxième position et il inclut désormais les CWE qui sont davantage liés aux échecs d'identification.

S'en prémunir :

- Ne pas livrer ou déployer avec des informations d'identification par défaut, en particulier pour les utilisateurs avec privilèges ;
- Intégrer des tests de mots de passes faibles, à la création ou au changement. Comparer ce mot de passe avec la liste des 10000 mots de passe les plus faibles ;

A08 : 2021 -> Les défaillances d'intégrité des logiciels et des données

- Une nouvelle catégorie pour 2021, sur la formulation d'hypothèses liées aux mises à jour logicielles, aux données critiques et aux pipelines CI/CD sans vérifier l'intégrité.

S'en prémunir :

- Utilisez des signatures numériques ou des mécanismes similaires pour vérifier que le logiciel ou les données proviennent de la source prévue et n'ont pas été modifiés ;
- Veillez à ce que les données sérialisées non signées ou non chiffrées ne soient pas envoyées à des clients non fiables sans une forme de contrôle d'intégrité ou de signature numérique permettant de détecter la falsification ou le rejeu des données sérialisées

A09 : 2021 -> Security Logging and Monitoring failures

- Monté d'un rang, cette catégorie inclut davantage de types de défaillances, difficile à tester et n'est pas bien représentée dans les données CVE / CVSS.

S'en prémunir :

- S'assurer que les enregistrements des journaux sont dans un format standard pour permettre de les intégrer facilement à une solution de gestion de logs centralisée ;
- Veiller à ce que les données des journaux soient correctement encodées afin d'éviter les injections ou les attaques sur les systèmes de journalisation ou de surveillance ;

A10 : 2021 -> Server-Side Request Forgery

- La contrefaçon de requête coté serveur est un type d'exploit où un attaquant abuse de la fonctionnalité du serveur, ce qui lui permet de contrôler les informations dans le domaine de ce serveur.

S'en prémunir :

- Assainir et valider toutes les données d'entrée fournies par le client ;
- Désactiver les redirections HTTP ;

C. Risques des failles

Source : https://kinsta.com/fr/blog/injections-sql/

L'injection SQL est un type **d'exploitation** d'une **faille** de **sécurité** d'une **application** agissant avec une **base donnée**. **L'attaquant détourne les requêtes en y injectant une chaine non prévue par le développeur.**

Ceci pourrait apporter de grosses conséquences car un hackeur peut avoir accès non autorisé aux données sensibles. Il pourra alors lire, enregistrer, ou encore exécuter du code malveillant. Via ces informations il serait dommageable pour un site de subir une telle cyberattaque.

Voici un exemple afin d'illustrer les faits :

Une requête exécutée par le système :

```
SELECT user_id
FROM Users
WHERE name = '(identifiant)' AND password = '(Mot de passe hashé)';
```

Le script doit vérifier les données entrées par les utilisateurs, cependant s'il ne les vérifie pas, il serait alors possible d'attaquer la requête en ajoutant certains caractères dans le champ identifiant, **comme '--'.**

Ce qui donnera donc :

```
SELECT user_id
FROM Users
WHERE name = '(identifiant)'; --' AND password = '(Mot de passe hashé)';
```

Les caractères **--** permet de marquer un commentaire en SQL, donc la deuxième condition « Where » ne sera pas exécutée. L'utilisateur pourra alors se connecter sans mot de passe.

III. Solutions à ses attaques

A. Techniques et méthodologie de protection

Injection SQL :

Source : https://openclassrooms.com/fr/courses/6179306-securisez-vos-applications-web-avec-lowasp/6520701-protegez-votre-code-contre-l-injection

Afin de parer de l'injection SQL, il est simple de protéger en utilisant des requêtes préparées.
Le plus souvent utilisé est le Try {} Catch () qui permettra de préparer une requête et ensuite de l'exécuté.
Lors de la préparation de la requête, un Template est envoyé à la base de données, qui effectuera une vérification de la syntaxe et qui initialisera les ressources internes du serveur pour une utilisation ultérieure.

Échecs d'identification et d'authentification :

Source : https://ssi.ac-strasbourg.fr/bonnes-pratiques/recommandations/lidentification-et-lauthentification/

Pour éviter cette faille est d'implémentez l'authentification multi facteur pour éviter les attaques automatisées, le bourrage des informations d'identification, la force brute et la réutilisation des informations d'identification volées.

Cross site Scripting (XSS) maintenant connu sous **Injection** :

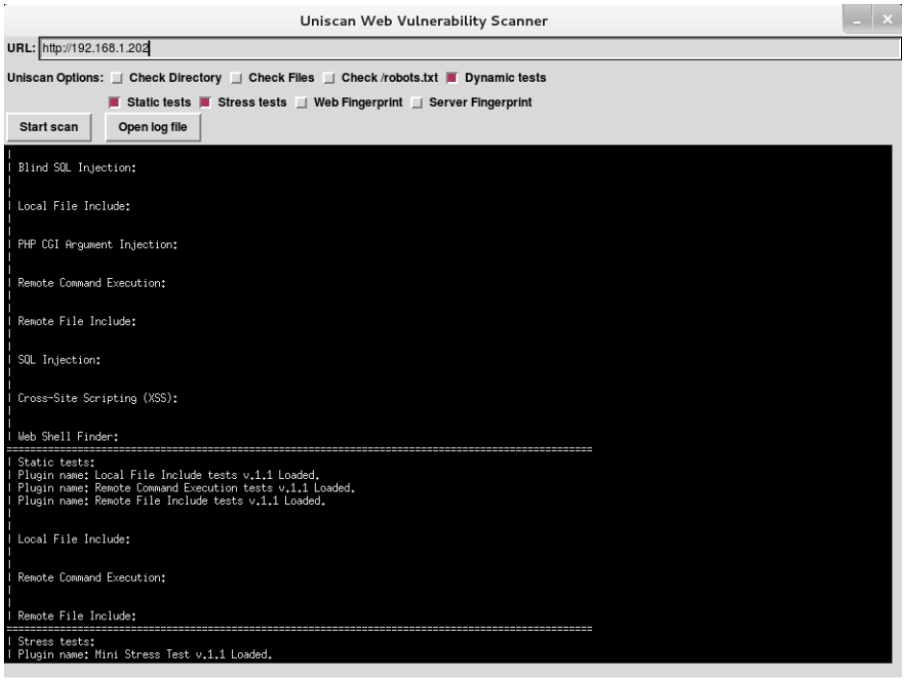
Source : https://www.journaldunet.com/solutions/dsi/1209139-comment-eviter-les-failles-cross-site-scripting-xss/

La façon la mieux adaptée contre cette faille est d'utiliser l'encodage de sortie et la désinfection HTML.
Nous pouvons utiliser une fonction HTML qui est htmlspecialchars (), qui permettra de filtrer les symboles en les remplaçant par un équivalent qui lui ne nuira pas lors de l'injection.

B. Outils de contrôle de la sécurité des applications

Source : https://securitytrails.com/blog/uniscan

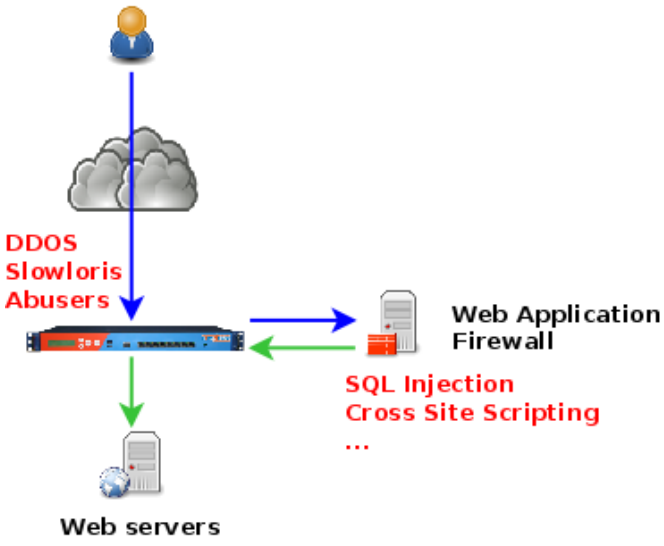
Uniscan : Il permet de scanner les vulnérabilités pour les applications Web. Codé en Perl, il est conçu pour détecter différentes failles dont laquelle les injections SQL y est compris.



Naxsi Web Application Firewall : Ce script permet de vous aider à sécuriser votre site Web contre des injections SQL, Cross Site Scripting…

Source : https://www.nbs-system.com/publications/securite/waf-naxsi/

La différence entre d'autres Web application Firewalls est le fait que celle-ci se base sur la détection de caractères inattendus dans les requêtes http et les arguments passés.



Conclusion

On peut en conclure que le classement de l'OWASP a rencontré d'énorme changement de position depuis l'OWASP Top 10 de 2017, en plus des nouveaux risques liés à la sécurité des applications Web.

Cela signifiera donc que les développeurs devront encore prendre plus de précaution et tenir leurs programmations en accord avec les règles qu'imposes l'OWASP Top 10.

Autres sources

- Feedly -> Flux RSS