Table des matières

Démarrage

Ouvrer votre navigateur et aller sur l'adresse suivante : <Adresse IP de la machine hote>:8080.

Vous devriez voir la page d'accueil du site.

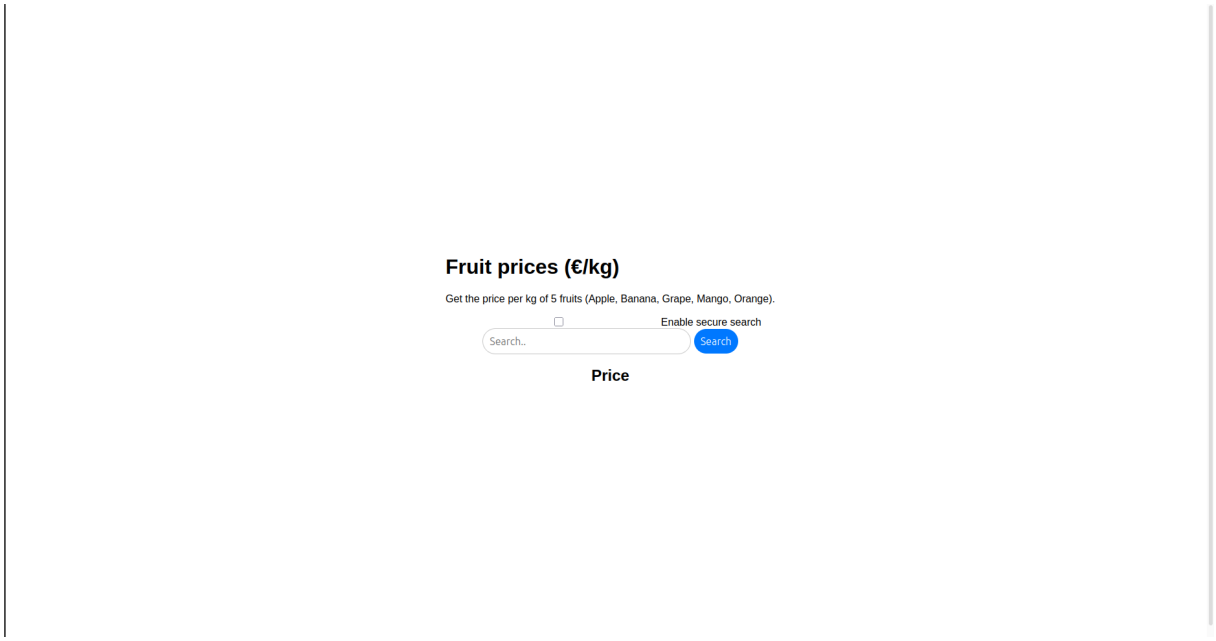

Fig. 1. – Page d'accueil du site

Sur la page d'acceuil vous trouverez une barre de recherche. Vous pouvez y entrer le nom d'un fruit parmi ceux donnée sur la page d'acceuil.

Cliquer sur « Search » pour lancer la recherche. Vous devriez voir le résultat de la recherche s'afficher en dessous de la barre de recherche.

Objectif

L'objectif de ce laboratoire est de vous montrer un type d'attaque sur une base de données non protégée accessible via un site web.

Vous allez devoir trouver un moyen de contourner la fonctionnalité de base de recherche du site pour obtenir plus d'informations que ce qui est prévu.

Pour cela, vous allez décocher la case « Enable secure search » au dessus de la barre de recherche.

Le but de ce laboratoire est trouver les identifiants et mot de passe d'utilisateur de la base de données.

Exploitation

La première étape de l'exploitation est de reconnaitre qu'il s'agit d'une injection SQL.

Pour cela, vous allez devoir essayer d'injecter une commande SQL dans la barre de recherche.

Par exemple pour tester si l'injection SQL est possible, vous pouvez essayer d'entrer la commande suivante dans la barre de recherche :

```sql
' OR 1=1 --
```

Vous devriez voir tous les fruits s'afficher dans le résultat de la recherche.

Cette requete fonctionne car l'instruction « OR 1=1 » est toujours vraie, ce qui permet de contourner la condition de recherche initial.

- Le caractère « ' » permet de terminer la chaîne de caractères de la requête SQL.
- Le mot clé « OR » permet d'ajouter une condition supplémentaire à la requête SQL.
- Le « 1=1 » est une condition toujours vraie.
- Les « - - » permet de commenter le reste de la requête SQL, ce qui permet d'ignorer la condition de recherche initiale.

Vous aller essayer d'autre unjection pour voir si vous pouvez obtenir plus d'informations sur la base de données.

Récuperation des tables

Vous pouvez essayer d'entrer la commande suivante dans la barre de recherche :

```sql
' UNION SELECT NULL, name FROM sqlite_master WHERE type='table' --
```

Cette requete permet de lister les tables de la base de données. Vous devriez voir s'afficher la liste des tables de la base de données (fruits, items, users).

- Le mot clé « UNION » permet de combiner les résultats de deux requêtes SQL.
- Le mot clé « SELECT » permet de sélectionner des colonnes spécifiques dans une table.
- Le « NULL » et « name » permet de sélectionner la colonne « name » de la table « sqlite_master ». Le null est la pour combleter le nombre de colonnes de la requête.
- Le mot clé « FROM » permet de spécifier la table à partir de laquelle vous souhaitez sélectionner des données.
- Le mot clé « WHERE » permet de spécifier une condition pour filtrer les résultats.
- Le mot clé « type » permet de spécifier le type de la table à sélectionner. Dans ce cas, nous voulons sélectionner les tables de type « table ».
- Le mot clé « sqlite_master » est une table système qui contient des informations sur toutes les tables de la base de données.

Récuperation des colonnes

Pour accéder à la table users, vous pouvez essayer d'entrer la commande suivante dans la barre de recherche :

```sql
' UNION SELECT NULL, sql FROM sqlite_master WHERE name='users' --
```

Cette requete permet de lister les colonnes de la table users. Vous devriez voir s'afficher la liste des colonnes de la table users (id, username, email, password).

- Le « NULL » et « sql » permet de sélectionner la colonne « sql » de la table « sqlite_master ». Le null est la pour combleter le nombre de colonnes de la requête.

- Le name est la pour spécifier le nom de la table à sélectionner. Dans ce cas, nous voulons sélectionner la table « users ».

Récuperation des données

Pour récupérer les données de la table users, vous pouvez essayer d'entrer la commande suivante dans la barre de recherche :

```sql
' UNION SELECT username, password FROM users --
```

Cette requete permet de lister les usernames et les mots de passe de tous les utilisateur de la table users. Vous devriez voir s'afficher la liste des usernames et des mots de passe de tous les utilisateur de la table users.

- Le « username » et « password » permet de sélectionner les colonnes « username » et « password » de la table « users ».

En changeant le nom de la colonne « username » par « email », vous pouvez récupérer les emails des utilisateurs de la base de données.

```sql
' UNION SELECT email, password FROM users --
```

Explication de l'injection

L'injection SQL est une technique d'attaque qui permet à un attaquant d'exécuter des requêtes SQL malveillantes sur une base de données. Comme vous l'avez vu dans ce laboratoire, il est possible d'injecter des commandes SQL dans une requête SQL existante pour contourner les restrictions de sécurité et accéder à des données sensibles.

L'injection SQL est possible lorsque les entrées utilisateur ne sont pas correctement filtrées ou échappées avant d'être utilisées dans une requête SQL. Cela permet à un attaquant d'injecter des commandes SQL malveillantes dans la requête, ce qui peut entraîner la divulgation de données sensibles, la modification de données ou même la suppression de données.

Normalment les library de base de données modernes utilisent de base des requêtes précompilées pour éviter ce genre de problème.

Piste de protection

L'injection SQL est une vulnérabilité courante dans les applications interagissant avec une base de données mais qui est facilement évitable en utilisant des requêtes précompilées ou en filtrant correctement les entrées utilisateur avant de les utiliser dans une requête SQL.

Une requête précompilée est une requête SQL qui est préparée et compilée par le serveur de base de données avant d'être exécutée. Cela permet d'éviter les injections SQL car les entrées utilisateur sont traitées comme des paramètres et non comme du code SQL.

Une analyse de l'entré utilisateur est également une bonne pratique pour éviter les injections SQL. Cela consiste à filtrer les entrées utilisateur pour s'assurer qu'elles ne contiennent pas de caractères spéciaux ou de commandes SQL malveillantes. Cela peut être fait en utilisant des expressions régulières (regex) ou en utilisant des fonctions de filtrage fournies par le langage de programmation utilisé.

Bibliographie

[1] « SQL injection ». Consulté le: 24 avril 2025. [En ligne]. Disponible sur: https://en.wikipedia.org/w/index.php?title=SQL_injection&oldid=1283306755

[2] « SQL Injection ». Consulté le: 24 avril 2025. [En ligne]. Disponible sur: https://www.w3schools.com/sql/sql_injection.asp

[3] « Regular expression ». Consulté le: 24 avril 2025. [En ligne]. Disponible sur: https://en.wikipedia.org/w/index.php?title=Regular_expression&oldid=1284282690

[4] « sqlite3 — DB-API 2.0 interface for SQLite databases ». Consulté le: 24 avril 2025. [En ligne]. Disponible sur: https://docs.python.org/3/library/sqlite3.html