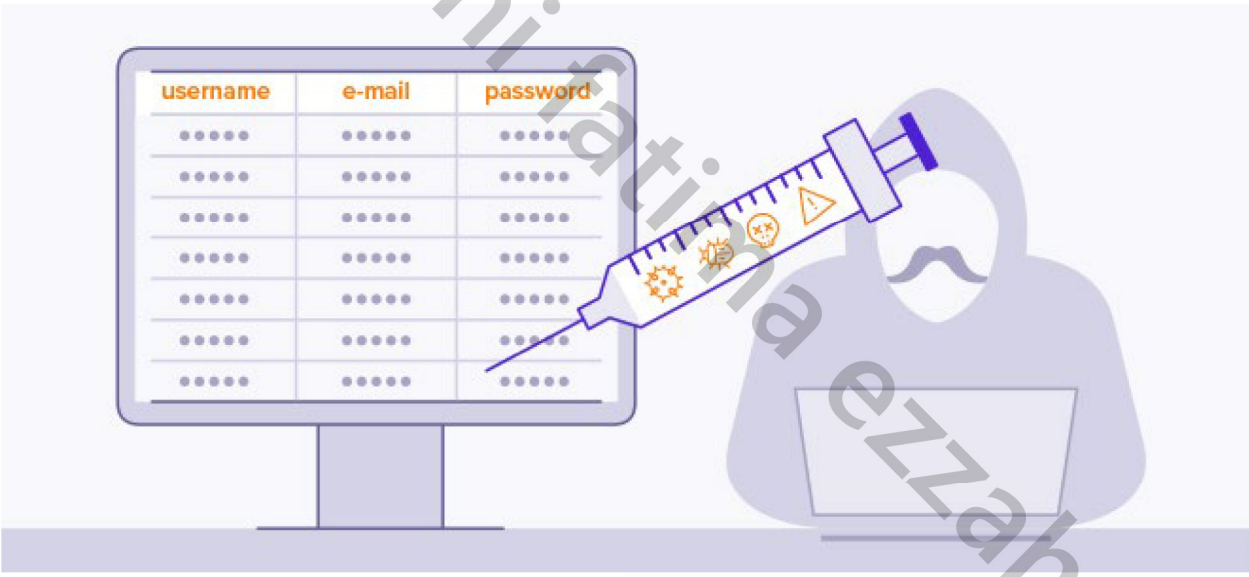
EST
المدرسة العليا للتكنولوجيا
جامعة الحسن الثاني الدار البيضاء
Ecole Supérieure de Te hnologie
Université Hassan II de Casablanca

SQL Injection



Realise par :

El arrouchi fatima ezzahra

Cette attaque est juste pour des raisons educatives sur un site web open source !

Qu'est ce qu'une injection SQL ?

L'injection SQL est une technique d'injection de code qui exploite les vulnérabilités des applications web en insérant du code SQL malveillant dans des requêtes, compromettant potentiellement les bases de données. C'est une méthode de piratage courante qui se produit souvent lorsque l'entrée utilisateur est utilisée directement dans les requêtes SQL sans validation appropriée.

Son fonctionnement :

On sait que lors d'une connexion à un site web, lorsque l'utilisateur entre son login et mot de passe, une requête SQL est envoyée à la base de données, et ensuite cette requête est évaluée en comparant les données entrées par l'utilisateur à chaque ligne de la base de données. Si elles sont vraies, l'utilisateur a accès au site web, sinon il sera rejeté.
Donc, pour cette attaque, elle trompe la requête SQL envoyée de plusieurs manières, selon le type d'attaque qu'on verra par la suite, et donc l'utilisateur aura toujours accès et la requête sera évaluée d'une autre manière, ce qui la rendra toujours vraie.

Types d'Injections :

- **OR Payloads :**
  Les OR payloads sont des attaques spécifiques dans l'injection SQL où un attaquant utilise l'opérateur logique `OR` pour manipuler une requête SQL et la rendre toujours vraie.
  Exemple :
  ⇨ Requete SQL normal :
  SELECT * FROM users WHERE username='user' AND password='passwd'
  ⇨ Requete SQL injectee :
  SELECT * FROM users WHERE username='user' OR '1'='1'' AND password='passwd'

- **Comment Payloads :**

  Les comment payloads sont une autre technique utilisée dans l'injection SQL, où un attaquant insère des commentaires dans une requête SQL pour manipuler son comportement ou masquer des parties de la requête originale

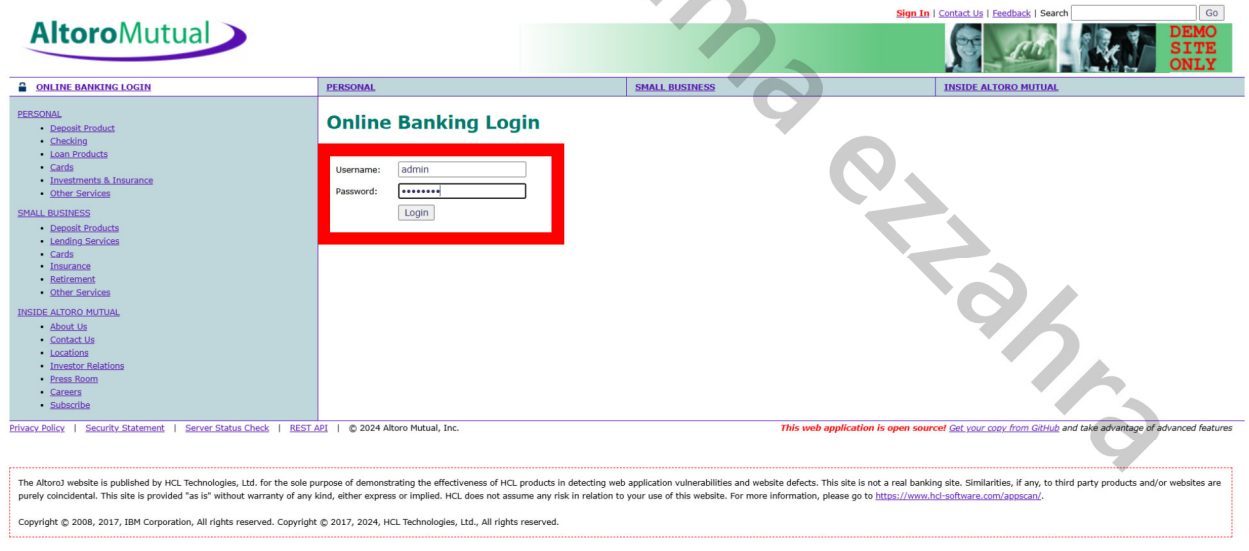
  Exemple :

  ⇨ Requete SQL originale :
  SELECT * FROM users WHERE username='user' AND password='passwd'

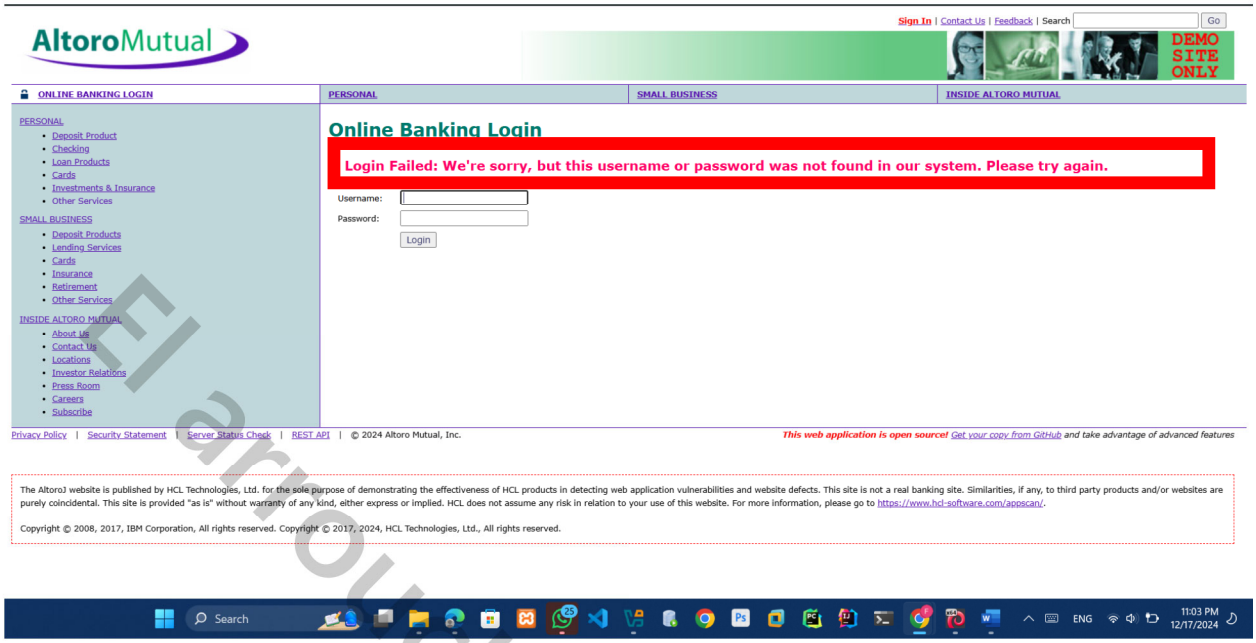
  ⇨ Requete SQL injectee :
  SELECT * FROM users WHERE username='user'-- AND password='passwd'
  (toute la partie coloree en rouge sera commentee et donc il va juste verifie le username et apres le user peut entrer sans password)

Demonstration :

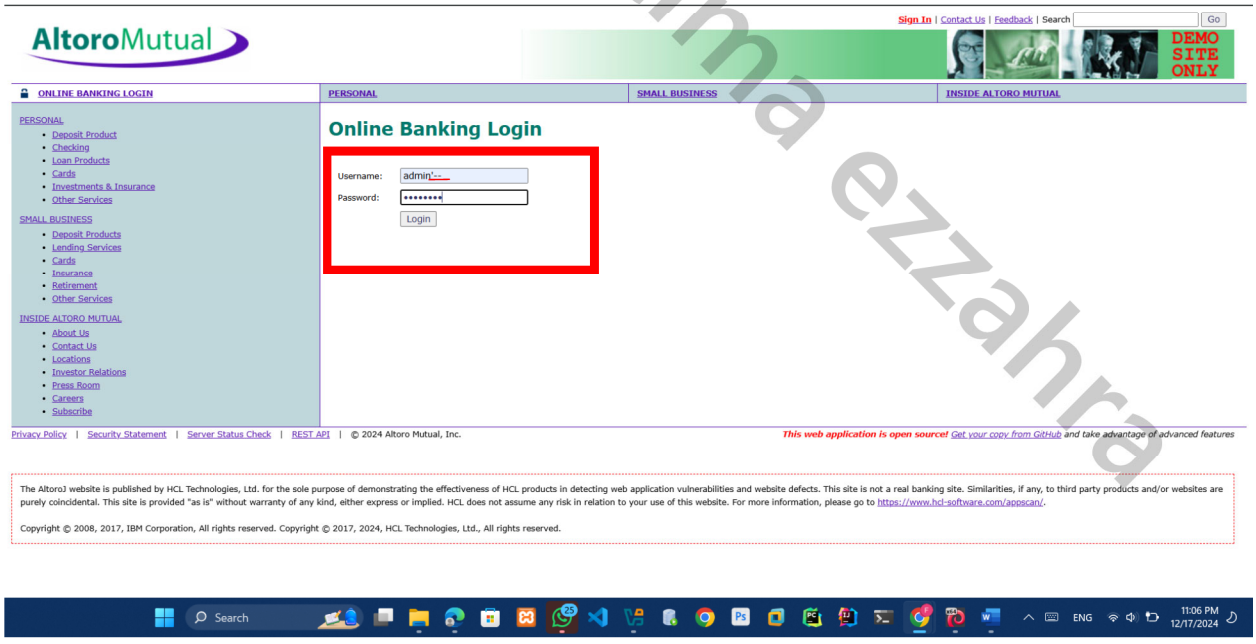
Dans cette attaque je vais utiliser le comment payload de l'injection SQL sur un siteweb opensource donc c'est totalement legal.



En entrant le login : Admin et mot de passe : Admin123 ➔ la connection sera refusee car on verifier les coordonnes ce mot de passe n'est pas associe a l'utilisateur Admin.



Maintenant on va essayer d'insere un commentaire dans le champs usernmae, tout en entrant meme username et password



Apres avoir clique sur login, notre Injection SQL a bien fonctionnee et bien sur le site web est vulnerable.