Attaque par injection SQL

- Il est possible pour un hacker d'envoyer dans le champs texte « username » quelquechose qui contournera votre requête SQL.
- Par exemple :
  - $res = mysqli_query($connexion, "SELECT * FROM users WHERE username = '".$_POST['username']."' AND password = '".$_POST['password']."'");
    if(mysqli_num_rows($res) > 0)
    {
    // *usager est authentifié..*
    }
  - Qu'est-ce qui se passe lorsque $_POST['password'] = « youarehacked' OR 1=1».
  - On peut se retrouver avec la requête SELECT * FROM users WHERE username = '". $_POST['username']."' AND password = 'youarehacked' OR 1=1" , ce qui retournerait toutes les entrées de la base…
  - L'usager pourrait aussi ajouter une requête DROP de la même façon, détruisant vos données…
- Il ne faut jamais se fier sur les variables données par l'usager et toujours les filtrer. Par exemple, vous pouvez utiliser la fonction mysqli_real_escape_string() après la connexion à mysql, qui sert à filtrer les requêtes SQL.
  - $username = mysqli_real_escape_string($connexion, $_POST['username']);
  - $username sera la version filtrée de $_POST['username']