

Veille techno

La sécurité des applications web

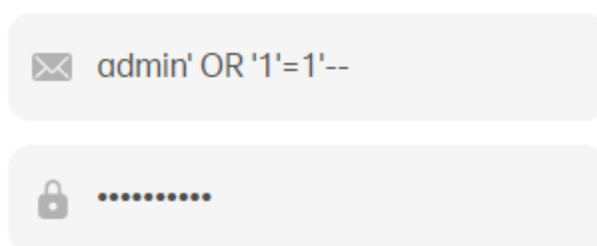
Faibles fréquentes

Injection SQL

Les attaques par injection SQL se produisent lorsque des données malveillantes sont injectées dans une base de données via une application web. Cela peut conduire à des fuites de données, à la modification ou à la suppression de données.

On peut prendre l'exemple d'une page d'authentification, l'authentification marche avec cette requête SQL.

```
SELECT * FROM WHERE username='admin' OR '1'='1' AND password='motdepasse';
```



The illustration shows a web login interface. The top field is for the username, indicated by an envelope icon, and contains the text "admin' OR '1'='1'--". The bottom field is for the password, indicated by a lock icon, and contains a series of dots representing masked characters.

[Mot de passe oublié ?](#)

`1=1` renvoie toujours TRUE.

Les `--` permet de mettre en commentaire du code SQL.

Dans ce cas, la requête SQL va être :

```
SELECT * FROM users WHERE username='admin' OR 1=1;
```

Cross-Site Scripting (XSS)

Cette attaque est au même titre que l'injective SQL, mais ici

Les attaques XSS injectent des scripts malveillants dans les pages web vues par d'autres utilisateurs, ce qui peut entraîner le vol de cookies, de sessions ou de données personnelles.

On peut prendre l'exemple d'un formulaire de contact.

Nom

Marine

✓

Dans le cas de PHP, le code de renvoi peut être

```
console.log("Bonjour, ".$name);
```

Si dans notre case on écrit

Nom

Lea; cat ./controller/controllerForm.php

✓

On va donc nous afficher le fichier "controllerForm.php".

Solution a ses failles

Bibliographie

<https://www.cert.ssi.gouv.fr/alerte/CERTFR-2023-ALE-005/>

blog.qualys.com

<https://www.vaadata.com/>

<https://www.cert.ssi.gouv.fr/>

https://owasp.org/Top10/A00_2021_Introduction/