

اختبار

SQL Injection

1. **الوصف:** لديك صفحة تسجيل دخول تحتوي على الحقليين التاليين: اسم المستخدم وكلمة المرور. في الشيفرة الخلفية، يتم تنفيذ الاستعلام التالي للتحقق من صحة المستخدم:

```
SELECT * FROM users WHERE username = '$username' AND password = '$password';
```

المهمة:

- اشرح لماذا هذه الشيفرة عرضة لـ SQL Injection.
- اقترح طريقة لإصلاحها.

XSS

1. **الوصف:** لديك صفحة تعرض تعليقات المستخدمين دون تنقية المدخلات. يمكن لأي مستخدم إدخال تعليق يحتوي على شيفرة JavaScript:

```
<div><?php echo $comment; ?></div>
```

المهمة:

- اشرح لماذا هذه الشيفرة عرضة لـ XSS.
- اقترح طريقة لإصلاحها.

CSRF

1. **الوصف:** لديك نموذج لتغيير كلمة المرور يتم إرساله عبر طلب POST

```
<form method="POST" action="change_password.php">
  <input type="password" name="new_password">
  <input type="submit" value="Change Password">
</form>
```

المهمة:

السؤال ١

السبب في SQL Injection:

الشفيرة عرضة لـ SQL Injection لأن المتغيرات \$username و \$password يتم دمجها مباشرة في الاستعلام بدون أي معالجة مسبقة. مما يتيح للمهاجم إدخال استعلامات ضارة.

- اشرح لماذا هذا النموذج عرضة لـ CSRF.
- اقترح طريقة لإصلاحها.

الحل:

السؤال: XSS

استخدم Parameterized أو Prepared Statements Queries لتجنب SQL Injection. مثال في PHP:

```
$stmt = $conn->prepare("SELECT * FROM users
WHERE username = ? AND password = ?");
$stmt->bind_param("ss", $username, $password);
$username = $_POST['username'];
$password = $_POST['password'];
$stmt->execute();
```

هذا يضمن أن المدخلات تتم معالجتها بشكل آمن.

السبب: الشيفرة عرضة لـ XSS لأن المدخلات من المستخدمين تعرض مباشرة في الصفحة بدون تنقية.

الحل:

استخدم htmlspecialchars لتفادي تنفيذ أي شيفرة JavaScript

```
<div><?php echo htmlspecialchars($comment,
ENT_QUOTES, 'UTF-8'); ?></div>
```

السبب: النموذج عرضة لـ CSRF لأنه يمكن لأي شخص إرسال طلب POST نيابة عن المستخدم، مما يسمح للمهاجم بتغيير كلمة المرور دون موافقة المستخدم.

الحل:

استخدم رمز (Token) CSRF في النموذج للتحقق من صحة الطلب:

```
<form method="POST" action="change_password.php">
  <input type="hidden" name="csrf_token" value="<?php
    echo $_SESSION['csrf_token']; ?>">
  <input type="password" name="new_password">
  <input type="submit" value="Change Password">
</form>
```

وفي الكود الخلفي:

```
if ($_POST['csrf_token'] !== $_SESSION['csrf_token']) {
    // توقف العملية
    die("CSRF token mismatch");
}
```