

# Решение СЗИ-10 Защита от XSS-грифера

## Описание

"Гриферы взломали сайт с покупкой лицензий Майнкрафта. Почини форму входа так, чтобы гриферы тоже платили, а не играли за чужой счет.">

## Решение

### Анализ уязвимости

Открываем задание и видим функцию `render_login()`, которая напрямую вставляет параметр `username` в HTML без экранирования:

```
def render_login(username: str) -> str:
    # БЛОК ДЛЯ РЕДАКТИРОВАНИЯ:
    html = f"""
    <html>
        <head><title>Minecraft License Login</title></head>
        <body>
            <h1>Welcome, {username}!</h1>
            <form method="post" action="/do_login">
                <input name="username" value="{username}">
                <input type="password" name="password">
                <button>Login</button>
            </form>
        </body>
    </html>
    """
    return html
    # КОНЕЦ БЛОКА ДЛЯ РЕДАКТИРОВАНИЯ
```

Проблема: злоумышленник может внедрить JavaScript код через параметр `username`.

### Пример решения

Добавляем экранирование HTML-символов с помощью `html.escape()`:

```
import html

def render_login(username: str) -> str:
    safe_username = html.escape(username)
    html_content = f"""
    <html>
      <head><title>Minecraft License Login</title></head>
      <body>
        <h1>Welcome, {safe_username}!</h1>
        <form method="post" action="/do_login">
          <input name="username" value="{safe_username}">
          <input type="password" name="password">
          <button>Login</button>
        </form>
      </body>
    </html>
    """
    return html_content
```

## Получение флага

После отправки исправленного кода на проверку получаем флаг:

```
vsosh{r3fl3ct3d_xss_pr0t3ct10n_m1n3cr4ft}
```