Aqui estão as sugestões específicas para corrigir os problemas apontados no relatório do ZAP:

---

**1. Melhorar a Segurança da Política de Segurança de Conteúdo (CSP)**

**Problemas identificados:**

- CSP: Falha na definição de diretiva sem fallback
- CSP: script-src inseguro (unsafe-inline)
- CSP: style-src inseguro (unsafe-inline)
- Cabeçalho da Política de Segurança de Conteúdo (CSP) não definido

**Correção:**
Defina e aplique um cabeçalho **Content Security Policy (CSP)** adequado para evitar ataques como **Cross-Site Scripting (XSS)**. Adicione o seguinte cabeçalho no servidor (Apache ou Nginx) ou diretamente no código PHP:

```
header("Content-Security-Policy: default-src 'self'; script-src 'self'; style-src 'self'; object-src 'none'; frame-ancestors 'none'; base-uri 'self';");
```

Caso utilize um framework como Express.js (Node.js), pode configurar:

```
const helmet = require('helmet');

app.use(helmet.contentSecurityPolicy({
  directives: {
    defaultSrc: ["'self'"],
    scriptSrc: ["'self'"],
    styleSrc: ["'self'"],
    objectSrc: ["'none'"],
    frameAncestors: ["'none'"],
    baseUri: ["'self'"],
  }
}));
```

Se necessário permitir scripts inline, use **nonce** para evitar ataques XSS.

---

**2. Proteger Cookies (Evitar Sequestro de Sessão)**

**Problemas identificados:**

- Cookie sem sinalizador HttpOnly
- Cookie sem atributo SameSite

**Correção:**
No PHP, ao configurar cookies, adicione os seguintes atributos para proteção contra ataques **Cross-Site Request Forgery (CSRF) e Cookie Theft**:

```
setcookie("session", $session_id, [
  'expires' => time() + 3600, // 1 hora
  'path' => '/',
  'domain' => '', // Defina se necessário
  'secure' => true, // Somente via HTTPS
  'httponly' => true, // Evita acesso via JavaScript
  'samesite' => 'Strict' // Protege contra CSRF
]);
```

No Apache/Nginx:

```
Header always edit Set-Cookie ^(.*)$ $1; HttpOnly; Secure; SameSite=Strict
```

---

**3. Remover Informações Sensíveis nos Cabeçalhos HTTP**

**Problemas identificados:**

- O servidor vaza informações via X-Powered-By
- Vazamento de versão do servidor no cabeçalho Server

**Correção:**

- No **Apache**, edite o arquivo httpd.conf ou apache2.conf e adicione:
- ServerSignature Off
- ServerTokens Prod
- Header unset X-Powered-By
- No **Nginx**, edite nginx.conf:
- server_tokens off;
- No **PHP**, edite php.ini:
- expose_php = Off

---

**4. Evitar Exposição de Arquivos Sensíveis**

**Problema identificado:**

- **Arquivo oculto encontrado** (sitemap.xml, status-do-servidor)

**Correção:**

- Restrinja o acesso a arquivos sensíveis via .htaccess no Apache:
- <Files "status-do-servidor">
- Order Allow,Deny
- Deny from all
- </Files>
- No **Nginx**, bloqueie diretórios sensíveis:
- location /status-do-servidor {
- deny all;
- return 403;
- }

---

**5. Evitar Manipulação de Parâmetros (Prevenir Ataques de Injeção)**

**Problema identificado:**

- **Adulteração de parâmetros** (possível ataque de manipulação de requisições POST)

**Correção:**

1. **Sanitize e validar entradas de usuário**
2. $user_input = filter_input(INPUT_POST, 'campo', FILTER_SANITIZE_STRING);
3. **Usar Prepared Statements no SQL**
4. $stmt = $pdo->prepare("SELECT * FROM users WHERE email = ?");
5. $stmt->execute([$email]);
6. **Configurar regras no .htaccess para restringir manipulação direta de parâmetros**
7. <IfModule mod_rewrite.c>
8. RewriteCond %{QUERY_STRING} (union|select|insert|delete|update|drop|script) [NC]
9. RewriteRule .* - [F]
10. </IfModule>

---