
🔐 Guia Didático de Segurança da Informação para Iniciantes (PHP)

🎯 Objetivo:

Apresentar os principais conceitos de segurança na web de forma **simples e acessível**, usando exemplos do cotidiano e explicações visuais, para que qualquer pessoa — mesmo sem experiência técnica — consiga compreender os riscos e boas práticas.

---

1. CSP Ausente ou Insegura (Content Security Policy)

🧠 Conceito:

O navegador pode ser orientado a **permitir apenas recursos confiáveis**, como scripts e imagens do próprio site. Se essa proteção não existir, o navegador pode aceitar código malicioso.

🧨 Exemplo prático:

Imagine um site onde um campo de comentários exibe exatamente o que o usuário escreveu. Um invasor pode escrever algo como:

```
<script>alert('Peguei você!');</script>
```

Se o site aceitar isso, o navegador vai executar o alerta — ou pior: roubar cookies, redirecionar o usuário etc.

✅ Correção:

- Configure o site para **rejeitar scripts externos ou inline**.
- Use políticas como:
- Content-Security-Policy: default-src 'self'; script-src 'self'

🧠 Analogia:

É como um restaurante que só permite ingredientes da sua própria cozinha. Se deixar qualquer um trazer comida de fora (possivelmente estragada), os clientes podem passar mal.

---

2. Cookies sem HttpOnly, Secure e SameSite

🧠 Conceito:

Cookies são pequenos arquivos que armazenam informações de sessão. Se mal protegidos, podem ser roubados ou usados de forma errada.

🧨 Exemplos práticos:

- Sem HttpOnly: um script JavaScript malicioso pode ler o cookie.
- Sem Secure: o cookie é transmitido mesmo em conexões HTTP (não seguras).

- Sem SameSite: outros sites podem enviar requisições usando o cookie do usuário, sem consentimento.

✅ **Correção:**

Defina o cookie assim:

Set-Cookie: session_id=abc123; HttpOnly; Secure; SameSite=Strict

🧠 **Analogia:**

É como seu crachá de entrada de empresa. Se alguém puder ver, copiar ou usar em outra portaria, você perde o controle da segurança.

---

3. Cabeçalhos HTTP que Revelam Tecnologia

🧠 **Conceito:**

Alguns sites revelam nos cabeçalhos informações como:

- X-Powered-By: PHP/7.4
- Server: Apache/2.4.51

Isso dá pistas aos atacantes sobre **quais vulnerabilidades podem ser exploradas**.

🧨 **Exemplo prático:**

Se um hacker sabe que seu servidor usa Apache 2.4.1, ele pode procurar por falhas conhecidas nessa versão e tentar explorá-las diretamente.

✅ **Correção:**

- Remover ou ocultar esses cabeçalhos no servidor.
- Em PHP, use:
- expose_php = Off

🧠 **Analogia:**

É como andar na rua com um crachá dizendo "sou da segurança do banco X e minha senha é 1234".

---

4. Falta de Proteção Anti-CSRF (Cross-Site Request Forgery)

🧠 **Conceito:**

Um site pode aceitar comandos (como “transferir dinheiro”) vindos de outros sites, sem verificar se o pedido veio do usuário verdadeiro.

🧨 **Exemplo prático:**

Você está logado no seu banco. Visita um site malicioso que carrega silenciosamente:

```
<form action="https://seubanco.com/transferir" method="POST">
 <input type="hidden" name="valor" value="1000">
 <input type="hidden" name="para" value="hacker">
</form>
```

Você nem vê isso, mas o navegador **usa seus cookies válidos** e faz a transferência.

✅ **Correção:**

- Usar um **token secreto** nos formulários que o servidor verifica.
- Exemplo:

  ```
  <input type="hidden" name="csrf_token" value="ABC123">
  ```

🧠 **Analogia:**

É como assinar um contrato com folhas que alguém colocou dentro da sua pasta sem você perceber.

---

5. Falta do Cabeçalho X-Frame-Options (Clickjacking)

🧠 **Conceito:**

Sem esse cabeçalho, seu site pode ser aberto "por trás" de outro site, como um quadro invisível (iframe), e o usuário acaba clicando em botões sem perceber.

🧨 **Exemplo prático:**

Um site mostra um botão de "Ganhe R$100 agora!", mas na verdade, há um botão invisível do seu site embaixo, como "Deletar conta" — e é nele que o clique ocorre.

✅ **Correção:**

No servidor, adicione:

```
X-Frame-Options: DENY
```

Ou:

```
X-Frame-Options: SAMEORIGIN
```

🧠 **Analogia:**

É como um vidro transparente em cima do botão certo, mas com um papel escrito "Clique aqui para ganhar dinheiro" colado por cima.

📚 **Recomendações Finais**

🛡️ **Para quem está começando:**

- Leia cabeçalhos e comportamentos de sites usando a extensão **"Web Developer"** ou **"HTTP Headers"**.
- Aprenda sobre o **ciclo de vida de uma requisição HTTP**.
- Use sites como:
    - owasp.org
    - portswigger.net/web-security

🔍 **Testes seguros:**

- Nunca teste falhas em sites reais sem permissão.
- Use ambientes locais (ex: XAMPP, localhost) para praticar.
