**Sumário**

* **Introdução do Relatório**
* **Resumo Executivo**
* **Resultados da Varredura de Vulnerabilidades**
* **Descrição das Vulnerabilidades**
* **Associação com as Metodologias OWASP Top 10**
* **Plano de Ações de Mitigação**
* **Conclusão**
* **Anexos e Referencias**

# Introdução do Relatório

Este relatório documenta as vulnerabilidades criticas identificadas no site <http://testphp.vulnweb.com>, realizada no dia **01 de março de 2025,** usando a ferramenta **OWASP Zap** no **Kali Linux**. O objetivo é fornecer informações detalhadas sobre as falhas encontradas, seus impactos e recomendações para mitigação.

Foram identificadas diversas falhas que necessitam ser gerenciadas e corrigidas para mitigar possíveis intrusões. Neste relatório, também são apresentadas melhores práticas para tratar as vulnerabilidades listadas, além da classificação das falhas com base no **OWASP Top 10**.

# Resumo Executivo

**Resultados da Varredura de Vulnerabilidades**

* **3 vulnerabilidades de alto risco**, na qual podem comprometer significativamente a segurança do sistema, sendo elas categorizadas **A03:2021**, e cada uma dessas 3 vulnerabilidas é relacionado a Injection.
* **4 vulnerabilidades de risco médio**, na qual podem ser exploradas para obter acesso não autorizado ou comprometer a integridade dos dados;
* **3 vulnerabilidades de baixo risco**, embora menos críticas, podem servir como ponto de entrada para ataques mais avançados, pois nessas vulnerabilidades **encontradas**, elas vazam informações referentes ao servidor, como a versão utilizada.

**Descrição das Vulnerabilidades**

* As vulnerabilidades detectadas incluem falhas de **injection de Código (SQL Injection, XSS)**, **exposição de informações sensíveis devido a falta de CSRF e CSP**

**Associação com as Metodologias OWASP Top 10**

* 4 das vulnerabilidades encontradas estão alinhados com o top 10 do OWASP 2021, sendo elas categorizado entre A03:2021 e A01:2021

**Plano de Ações de Mitigação**

* Foi elaborado um plano de ação para mitigar os riscos identificados. As recomendações incluem a validação de **inputs** e **outputs** de forma mais rigorosa, além de evitar o vazamento de informações sobre a versão do servidor.

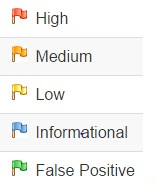
**Conclusão**

* Ao finalizar o teste no ambiente, foi apresentada uma grande quantidade de vulnerabilidades, sendo **imprescindível** a resolução desses problemas para que não afetem os **usuários** que acessarem os **produtos** oferecidos pelo seu site, de forma a garantir a **Confidencialidade, Integridade e Disponibilidade (CID).**

# Resultado da Varredura de Vulnerabilidades

|  |  |  |
| --- | --- | --- |
| **Nome** | **Nível de Risco** | **Número de Instâncias** |
| Cross Site Scripting (Refletido) | Alto | 19 |
| Cross Site Scripting (baseado em DOM) | Alto | 11 |
| Injeção SQL - MySQL | Alto | 19 |
| Ausência de tokens Anti-CSRF | Médio | 4 |
| Content Security Policy (CSP) Header Not Set | Médio | 54 |
| Injeção XSLT | Médio | 2 |
| Missing Anti-clickjacking Header | Médio | 46 |
| O servidor vaza informações por meio dos campos de cabeçalho de resposta HTTP "X-Powered-By" | Baixo | 79 |
| Server Leaks Version Information via "Server" HTTP Response Header Field | Baixo | 92 |
| X-Content-Type-Options Header Missing | Baixo | 82 |
| Authentication Request Identified | Informativo | 2 |
| Divulgação de Informações - Comentários Suspeitos | Informativo | 1 |
| GET for POST | Informativo | 3 |
| Modern Web Application | Informativo | 9 |
| Má Combinação de Charset (Cabeçalho versus Meta Content-Type Charset) | Informativo | 32 |
| User Agent Fuzzer | Informativo | 223 |
| User Controllable HTML Element Attribute (Potential XSS) | Informativo | 4 |

## Os niveis de riscos são categorizados entre:

**High(Alto), Medium(Médio), Low(Baixo), Informational (Informativo) e False Positive (Falso Positivo)**

As vulnerabilidades identificadas no **OWASP ZAP** são classificadas de acordo com o **CWE (Common Weakness Enumeration) da MITRE**, sendo também associadas ao **OWASP Top 10**.

# Descrição das Vulnerabilidades

Foram encontradas 10 tipos de vulnerabilidades, sendo três delas de alto risco, quatro de médio risco e três de baixo risco.

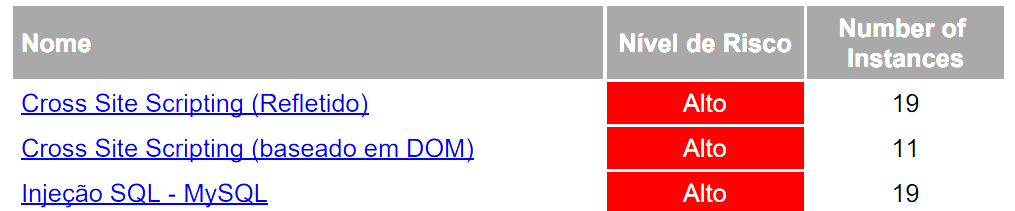
|  |  |
| --- | --- |
| Vulnerabilidade | Descrição |
| Cross Site Scripting (Refletido) | |  | | --- | | Script malicioso injetado na URL e enviado para o usuário. |  |  | | --- | |  | |
| Cross Site Scripting (baseado em DOM) | Script que afeta diretamente ao cliente, pois é uma vulnerabilidade que executa nos scripts |
| Injeção SQL - MySQL | O invasor usa algum código SQL em um campo de entrada para obter informações do banco de dados. |
| Ausência de tokens Anti-CSRF | Invasor engana o navegador do usuário para fazer uma solicitação não intencional a um site vulnerável. Isso pode ser conseguido atraindo o usuário para clicar em um link malicioso ou visitar um site comprometido. |
| Content Security Policy (CSP) Header Not Set | |  | | --- | | Falta de uma camada extra de prevenção contra XSS. O CSP testa o conteúdo no navegador do solicitante antes de enviar para o servidor web. |  |  | | --- | |  | |
| Injeção XSLT | Injeção de codigo malicisioso em XMLs |
| Missing Anti-clickjacking Header | Clickjacking é o ato de por algum objeto na frente do verdadeiro conteudo do site, na qual redireciona para algo malicioso |
| O servidor vaza informações por meio dos campos de cabeçalho de resposta HTTP "X-Powered-By" | Em erros comuns do site, em vez de enviar apenas a informação necessária ao usuário, o servidor vaza informações como versão e aplicativos instalados |
| Server Leaks Version Information via "Server" HTTP Response Header Field | O servidor vaza informações de versão por meio do campo de cabeçalho de resposta HTTP "Server". |
| X-Content-Type-Options Header Missing | Ausência do cabeçalho X-Content-Type-Options, o que pode permitir a execução de conteúdo de tipo MIME inesperado. |

# Impacto e Recomendações

Quais são os impacto das vulnerabilidades encontradas, dentre elas temos:

1. **Cross Site Scripting (Refletido)**
   1. **Quantidade Identificada:** 19
   2. **Impacto Potencial:** Roubo de informações, após recolher dados através do link URL malicioso
   3. **Recomendações:** Validação de entrada, é importante para prevenir a injeção de código, uso de Content Security Police (CSP)
   4. **Evidências:** <script> alert(1);</scRipt>; " onMouseOver="alert(1); <scrIpt>alert(1);</script>; </strong><scrIpt>alert(1);</scRipt><strong>
2. **Cross Site Scripting (baseado em DOM)**
   1. **Quantidade Identificada:** 11
   2. **Impacto Potencial:** Pode modificar uma pagina do site e adicionar conteúdo malicioso para que o usuário possa acessar.
   3. **Recomendações:** Use algum verificador para separar os dados e código, utilize ferramentas e frameworks,para fazer a filtragem, além de duplicar as saídas tanto no lado do cliente quanto no servidor para verificar possível ataque de modificação.
   4. **Evidências:** #jaVasCript:/\*-/\*`/\*\`/\*'/\*"/\*\*/(/\* \*/oNcliCk=alert(5397) )//%0D%0A%0d%0a//\x3csVg/\x3e; ?name=abc# <img src=”random.gif” onerror=alert(5397)>
3. **Injeção SQL – MySQL**
   1. **Quantidade Identificada: 19**
   2. **Impacto Potencial:** Ataque diretamente ao banco de dados, usando caixas de entrada sem qualquer proteção contra select, ou validação
   3. **Recomendações:** Não deixar aparente os parâmetros do SQL, e substituir por variável assim evita que no input seja feito a injeção de algum código SQL ‘?’ https://cheatsheetseries.owasp.org/cheatsheets/SQL\_Injection\_Prevention\_Cheat\_Sheet.htm
   4. **Evidências:**  <http://testphp.vulnweb.com/AJAX/infoartist.php?id=%27>; 2 and 0 in (select sleep(15) ) --
4. **Ausência de Tokens Anti-CSRF**
   1. **Quantidade Identificada: 4**
   2. **Impacto Potencial:** forçar a vítima a enviar uma solicitação HTTP a um destino alvo sem seu conhecimento ou intenção, a fim de realizar uma ação como a vítima
   3. **Recomendações**: Para melhorar a segurança, crie um **nonce** exclusivo e imprevisível para cada formulário, e coloque-o dentro do formulário. Ao recebê-lo, verifique se o **nonce** corresponde. Isso ajuda a prevenir ataques, mas vale lembrar que pode ser contornado por **XSS**. Quando uma operação for potencialmente perigosa, peça uma confirmação separada para garantir que o usuário realmente deseja realizá-la. Também é importante usar controles como o **ESAPI Session Management** e evitar o uso do método **GET** para qualquer ação que altere o estado. Por fim, verifique o cabeçalho **Referer** para garantir que a solicitação veio de uma página confiável.
   4. **Evidências:** http://testphp.vulnweb.com/cart.php; Parte superior do formulário
   5. <form action="search.php?test=query" method="post">; http://testphp.vulnweb.com/guestbook.php; Parte superior do formulário
   6. <form action="" method="post" name="faddentry">
5. **Content Security Policy (CSP) Header Não Definido**
   1. **Quantidade Identificada:** 54
   2. **Impacto Potencial:** A falta de cabeçalho estruturado, aumenta o risco de XSS no site
   3. **Recomendações:** Definir e configurar um cabeçalho CSP adequado
   4. **Evidências:** http://testphp.vulnweb.com/AJAX/index.php; http://testphp.vulnweb.com/artists.php?artist=1; http://testphp.vulnweb.com/artists.php?artist=2; http://testphp.vulnweb.com/artists.php?artist=3;
6. **Injeção XSLT**
   1. **Quantidade Identificada:** 2
   2. **Impacto Potencial:** A injeção XSLT pode permitir que um atacante execute código arbitrário no servidor ou no cliente. Isso pode ser feito manipulando o conteúdo XML;podendo causar vazamento de dados sensiveis,comprometimento do Sistema e de clients e modificação de conteudo.
   3. **Recomendações:** Use filtro de todos os dados que usam XMLs, façam uma verificação das entradas e saidas
   4. **Evidências:** <xsl:value-of select="document('http://testphp.vulnweb.com:22')"/>;
7. **Missing Anti-clickjacking Header**
   1. **Quantidade Identificada:** 46
   2. **Impacto Potencial:** O principal risco do clickjacking é forçar o usuário a realizar ações não intencionais em um site, como clicar em botões sem que ele perceba, e ele seja redirecionado para outro site. Além de ser uma engenharia social.
   3. **Recomendações:** Implementação de cabeçalho e X-Frame-Options e uso de Content Security Policy
   4. **Evidências:** http://testphp.vulnweb.com/

# Associação com as Metodologias OWASP Top 10

Nesse relatório, foram identificadas três grandes vulnerabilidades, que estão associadas ao TOP 10 do OWASP de 2021. Pode-se perceber também que há semelhanças no tipo de ataque, sendo eles ataques de **injection**, nos quais é feita a injeção de códigos maliciosos.

**Cross Site Scripting (Refletido)**

* **Categoria**: A03:2021-Injection
* **Nome**: Injection - XSS
* **Descrição**: Ocorrem quando um invasor faz com que uma vítima forneça conteúdo perigoso a um aplicativo da web vulnerável, que é então refletido de volta para a vítima e executado na URL do usuario.

**Cross Site Scripting (baseado em DOM)**

* **Categoria**: A03:2021-Injection
* **Nome**: Injection - XSS
* **Descrição**: O XSS baseado em DOM ocorre no lado do cliente, explorando a manipulação de scripts confiáveis, como JavaScript, antes do envio de um formulário.

**Injeção SQL – MySQL**

* **Categoria**: A03:2021-Injection
* **Nome**: Injection
* **Descrição**: Sem as devidas validações de input e a falta de controle da sintaxe SQL em inputs controlado pelo usuário, a consulta pode interpretar essas entradas como comandos SQL em vez de dados comuns..

**Ausência de tokens Anti-CSRF**

* **Categoria**: A01:2021
* **Nome**: Broken Access Control
* **Descrição**: O aplicativo não valida se as solicitações foram realmente enviadas pelo usuário legítimo, permitindo que invasores forcem ações maliciosas.

# Plano de Ações de Mitigação

Percebe-se que a maioria das vulnerabilidades apresentadas está relacionada à falta de controle de inputs e outputs. Por isso, há uma grande quantidade de vulnerabilidades do tipo injection. Além disso, não há ferramentas de defesa que garantam a confiabilidade dos dados apresentados, o que resulta em vulnerabilidades de broken access control, entre outras, devido à ausência de tokens de confiança. Isso compromete totalmente os princípios de **Confidencialidade, Integridade e Disponibilidade (CID).**

|  |  |  |
| --- | --- | --- |
| **Vulnerabilidade** | **Plano de Ação** | **Responsavel** |
| Cross Site Scripting (Refletido) | Implementar validação de inputs e outputs | Desenvolvedor |
| Cross Site Scripting (baseado em DOM) | Adote uma estratégia de validação de entradas com listas de valores permitidos, aceitando apenas os que atendem estritamente às especificações.Utilize bibliotecas como Anti-XSS da Microsoft, OWASP ESAPI e Apache Wicket. | Desenvolvedor |
| Injeção SQL - MySQL | Validar inputs e outputs, com uma rigorosa validação, além de implementar prepared statements para evitar injections em querys SQL. | DBA + Desenvolvedor |
| Ausência de tokens Anti-CSRF | Gere um nonce exclusivo para cada formulário, coloque o nonce no formulário e verifique o nonce ao receber o formulário. Certifique-se de que o nonce seja aleatório | Desenvolvedor |
| Content Security Policy (CSP) Header Not Set | Definir e configurar corretamente o cabeçalho Content-Security-Policy (CSP) no servidor web | Administrador da Infraestrutura |
| Injeção XSLT | Valide e análise todas as entradas do usuário vindo de qualquer lado do cliente quando for manipulação de XMLs | Desenvolvedor |
| Missing Anti-clickjacking Header | Implemente cabeçalho com X-Frame-Options para prevenir ataques de clickjacking, configurando-o como DENY ou SAMEORIGIN. | Desenvolvedor |
| O servidor vaza informações por meio dos campos de cabeçalho de resposta HTTP "X-Powered-By" | Desabilitar o cabeçalho HTTP "X-Powered-By" que expõe detalhes sobre a tecnologia usada no servidor. | Administrador da Infraestrutura |
| Server Leaks Version Information via "Server" HTTP Response Header Field | |  | | --- | | Desabilitar o cabeçalho HTTP "Server" que revela a versão do servidor web. |  |  | | --- | |  | | Administrador da Infraestrutura |
| X-Content-Type-Options Header Missing | Configurar e adicionar o cabeçalho X-Content-Type-Options com valor nosniff para evitar a execução de tipos MIME inesperados. | Administrador da Infraestrutura |

# Conclusão

Apartir desse relatório conclui-se que o site apresenta bastante vulnerabilidades, devido a falta de validações adequadas de segurança e controle de confiabilidade. Por tanto é classificado como **altamente vulnerável**. As principais medidaas na qual devem ser tomadas envolvem melhorias na questão da segurança de **inputs** e **outputs**, garantindo que os dados sejam adequadamente validados e tratados para evitar **injections** e **ataques de script XSS**. Essas ações são fundamentais para assegurar os princípios de **Confidencialidade, Integridade e Disponibilidade (CID)**, pilares essenciais para a proteção dos dados e a confiabilidade do sistema.

# Anexos e ReferenciasA screenshot of a computer AI-generated content may be incorrect.A screenshot of a computer AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Referencias

<https://www.zaproxy.org/docs/alerts/>

<https://owasp.org/www-project-top-ten/>

<https://cwe.mitre.org>

## Report OWASP ZAP

Para acessar o relatório do **OWASP ZAP**, acesse o repositório do meu GitHub.

<https://github.com/WestFS/bookshelf>