**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**.

**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 |

**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. |

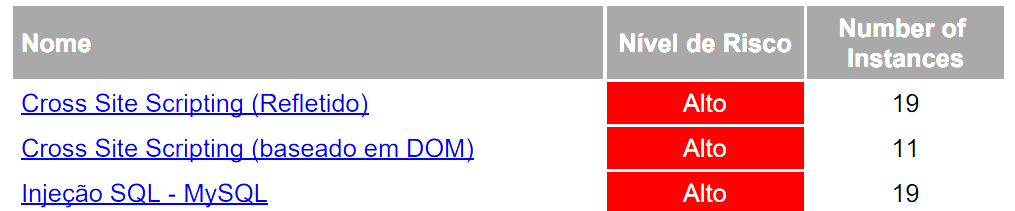
**Impacto e Recomendações**

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.html
   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 foi identificado 3 grandes vulnerabilidades, que estão associados ao TOP 10 do OWASP de 2021, pode se perceber também que há semelhanças do tipo de ataque, na qual são ataques de INJECTIONs, na qual é feito a injeção de codigos maliosos.



Cross Site Scripting (Refletido)

Cross Site Scripting (baseado em DOM)

Injeção SQL - MySQL