New issue

Stored XSS no Cadastro de Responsáveis via campos "Nome" e "Descrição" #2491

Open

ViniCastro2001 opened 2 weeks ago

Sumário

Uma vulnerabilidade de **Stored Cross-Site Scripting (XSS)** foi identificada na aplicação **SIGA**. A falha permite a injeção e persistência de código JavaScript através do campo **"Nome"** e **"Descrição"** na funcionalidade de cadastro de responsáveis.

O payload é armazenado no sistema e executado automaticamente quando os dados são renderizados na interface de listagem, permitindo a execução de código no contexto de sessões autenticadas.

Passos para reprodução rápida

1. Acessar /sigawf/app/responsavel/novo
2. Inserir payload no campo Nome ou Descrição

```
<img src=x onerror=alert(document.cookie)//
```

3. Salvar
4. Acessar /sigawf/app/responsavel/listar

Resultado: o payload é executado automaticamente ao carregar a página.

Detalhes técnicos

Componente Vulnerável: `Cadastro de Responsáveis` (`/sigawf/app/responsavel/`)

A aplicação falha em realizar **neutralização adequada de entrada controlada pelo usuário** antes da renderização no HTML. Os valores inseridos são posteriormente incluídos no DOM dentro de um elemento `<a href>`, sem encoding ou sanitização segura.

Embora exista um mecanismo de filtragem que bloqueia tags completas como `<script>` e `<img>`, essa proteção é insuficiente e pode ser facilmente contornada através do uso de **tags incompletas combinadas com sintaxe de comentário (`//`)**, permitindo a execução de código JavaScript.

O comportamento indica ausência de **output encoding contextual**, resultando em execução direta do conteúdo inserido pelo usuário.

Comportamento esperado:

- Dados fornecidos pelo usuário devem ser corretamente codificados antes da renderização (HTML encoding)
- Conteúdo potencialmente perigoso deve ser neutralizado no backend
- A aplicação não deve permitir execução de JavaScript a partir de campos de entrada

Comportamento observado:

- O payload é armazenado sem sanitização adequada
- O conteúdo é renderizado diretamente no HTML
- O código JavaScript é executado automaticamente ao visualizar a página

Prova de Conceito (PoC)

1. Acesso à funcionalidade vulnerável

O atacante navega até a funcionalidade de cadastro:

`/sigawf/app/responsavel/novo`

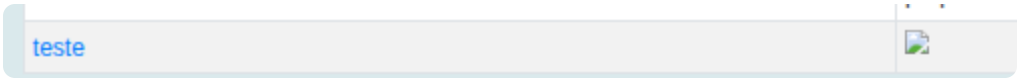
2. Inserção do payload

No campo **"Nome"** ou **"Descrição"**, o seguinte payload é inserido:

```
<img src=x onerror=alert(document.cookie)//
```

3. Persistência do payload

O registro é salvo normalmente pela aplicação, sem qualquer bloqueio efetivo.



4. Execução do payload

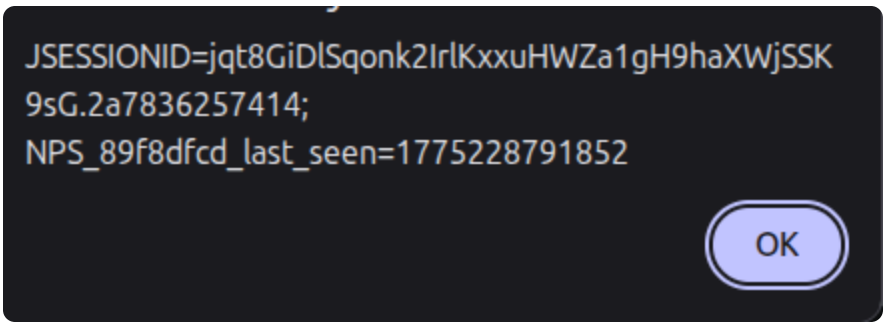
Ao acessar a página de listagem:

`/sigawf/app/responsavel/listar`

o conteúdo armazenado é renderizado e o JavaScript é executado automaticamente.

5. Confirmação

A execução do payload demonstra que é possível acessar dados disponíveis no contexto da sessão (ex: `document.cookie`), confirmando a execução arbitrária de código.



Impacto

- **Stored XSS:** Execução persistente de código: O payload é armazenado e executado automaticamente sempre que a página é acessada
- **Execução em contexto autenticado**: O código roda com os privilégios do usuário logado
- **Exposição de dados sensíveis**: Possibilidade de acesso a cookies, DOM e informações da sessão
- **Base para ataques mais avançados**: A vulnerabilidade pode ser utilizada como ponto de partida para exploração adicional, incluindo ações em nome do usuário e manipulação da aplicação

Metadata

Assignees

No one assigned

Labels

No labels

Type

No type

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

No branches or pull requests

Participants