

Relatório de segurança: Portal de Pedidos

Exemplo ilustrativo: a entidade, a aplicação e as falhas são fictícias e não correspondem a nenhum cliente.

ENTIDADE	Entidade Exemplo (fictícia)
APLICAÇÃO	Portal de Pedidos, versão 3.2
ÂMBITO	Aplicação web, API e código-fonte
ABORDAGEM	Caixa cinzenta, com contas de teste e acesso ao código
METODOLOGIAS	OWASP WSTG, OWASP ASVS e CVSS 4.0
ESTADO	Reteste concluído

Resumo executivo

Foram encontradas quatro falhas: uma crítica, uma alta, uma média e uma baixa. A mais grave permitia a qualquer pessoa, sem autenticação, aceder à base de dados através da pesquisa pública. Três falhas foram corrigidas e confirmadas no reteste; a de gravidade baixa está planeada para a próxima versão. Recomendamos manter a revisão de código nas novas versões e acompanhar as dependências da aplicação, para que falhas conhecidas sejam corrigidas logo que publicadas.

Achados por gravidade

Crítica		1
Alta		1
Média		1
Baixa		1
Corrigida e confirmada		3
Correção planeada		1

Achados

Cada achado foi reproduzido num ambiente de teste antes de entrar no relatório.

A-01 · Injeção de SQL na pesquisa pública

Crítica · 9,3

ESTADO	Corrigida e confirmada
LOCALIZAÇÃO	pesquisa/resultados.php, linha 42 · parâmetro «termo»
CVSS 4.0	CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N

Descrição

O termo de pesquisa é juntado diretamente à consulta à base de dados, sem parametrização. A página de pesquisa é pública, por isso não é preciso ter conta.

Prova

Num ambiente de teste, um valor inofensivo no parâmetro alterou a estrutura da consulta e a ordem dos resultados, o que confirma que o texto do utilizador é interpretado como parte da consulta. A prova completa foi entregue à equipa técnica e não é reproduzida aqui.

Impacto

Um atacante poderia ler, alterar ou apagar dados da base de dados, incluindo dados pessoais dos utentes, sem qualquer autenticação.

Correção proposta

ANTES

```
$sql = "SELECT id, assunto FROM pedidos
        WHERE assunto LIKE '%" . $_GET['termo'] . "%'";
$linhas = $db->query($sql)->fetchAll();
```

DEPOIS

```
$stmt = $db->prepare(
    'SELECT id, assunto FROM pedidos WHERE assunto LIKE ?'
);
$stmt->execute(['%' . $_GET['termo'] . '%']);
$linhas = $stmt->fetchAll();
```

Reteste

Consulta parametrizada em todas as pesquisas. O reteste confirmou que o parâmetro deixou de alterar a consulta.

A-02 · Acesso a pedidos de outros utentes

Alta · 7,1

ESTADO

Corrigida e confirmada

LOCALIZAÇÃO

API · GET /api/pedidos/{id}

CVSS 4.0

CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:N/VA:N/SC:N/SI:N/SA:N

Descrição

A API devolve um pedido a partir do seu identificador, mas não verifica se o pedido pertence ao utente autenticado.

Prova

Com a conta de teste A, o pedido de um registo criado pela conta de teste B devolveu os dados desse registo (redigidos no relatório).

Impacto

Qualquer utente com conta poderia consultar os pedidos de outros utentes, incluindo dados pessoais e documentos anexos.

Correção proposta

ANTES

```
$pedido = Pedido::find($id);
return response()->json($pedido);
```

DEPOIS

```
$pedido = Pedido::where('id', $id)
    ->where('utente_id', $utilizador->id)
    ->firstOrFail();
return response()->json($pedido);
```

Reteste

Verificação do dono aplicada a todos os pedidos da API. O reteste confirmou que a conta A deixou de ver os registos da conta B.

A-03 · Biblioteca de edição de texto com falha conhecida

Média · 5,3

ESTADO	Corrigida e confirmada
LOCALIZAÇÃO	package.json · biblioteca de edição de texto, versão 4.1
CVSS 4.0	CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:P/VC:N/VI:N/VA:N/SC:L/SI:L/SA:N

Descrição

A aplicação usa uma versão de uma biblioteca de edição de texto com uma vulnerabilidade publicada pelo fabricante, que permite incluir código nas mensagens. O identificador da vulnerabilidade foi omitido neste exemplo.

Prova

A versão foi identificada na análise de dependências e o comportamento descrito pelo fabricante foi confirmado num ambiente de teste.

Impacto

Uma mensagem preparada poderia executar código no navegador de quem a abrisse, por exemplo um técnico no backoffice.

Correção proposta**ANTES**

```
"dependencies": {  
  "editor-exemplo": "4.1.0"  
}
```

DEPOIS

```
"dependencies": {  
  "editor-exemplo": "4.3.2"  
}
```

Reteste

Biblioteca atualizada para a versão corrigida pelo fabricante. A dependência passou a ser acompanhada na subscrição anual.

A-04 · Cookie de sessão sem atributos de proteção

Baixa · 2,3

ESTADO	Correção planeada
LOCALIZAÇÃO	Configuração da sessão · config/session.php
CVSS 4.0	CVSS:4.0/AV:N/AC:H/AT:P/PR:N/UI:P/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N

Descrição

O cookie de sessão não tem os atributos Secure e HttpOnly, que impedem o envio por ligações não cifradas e o acesso a partir de código no navegador.

Prova

Os cabeçalhos da resposta de autenticação mostram o cookie de sessão sem os dois atributos.

Impacto

Combinado com outra falha, facilitaria o roubo de uma sessão. Sozinho, o risco é baixo.

Correção proposta

ANTES

```
"secure" => false,  
"http_only" => false,
```

DEPOIS

```
"secure" => true,  
"http_only" => true,  
"same_site" => "lax",
```

Reteste

Correção planeada para a próxima versão da aplicação, com novo reteste nessa altura.