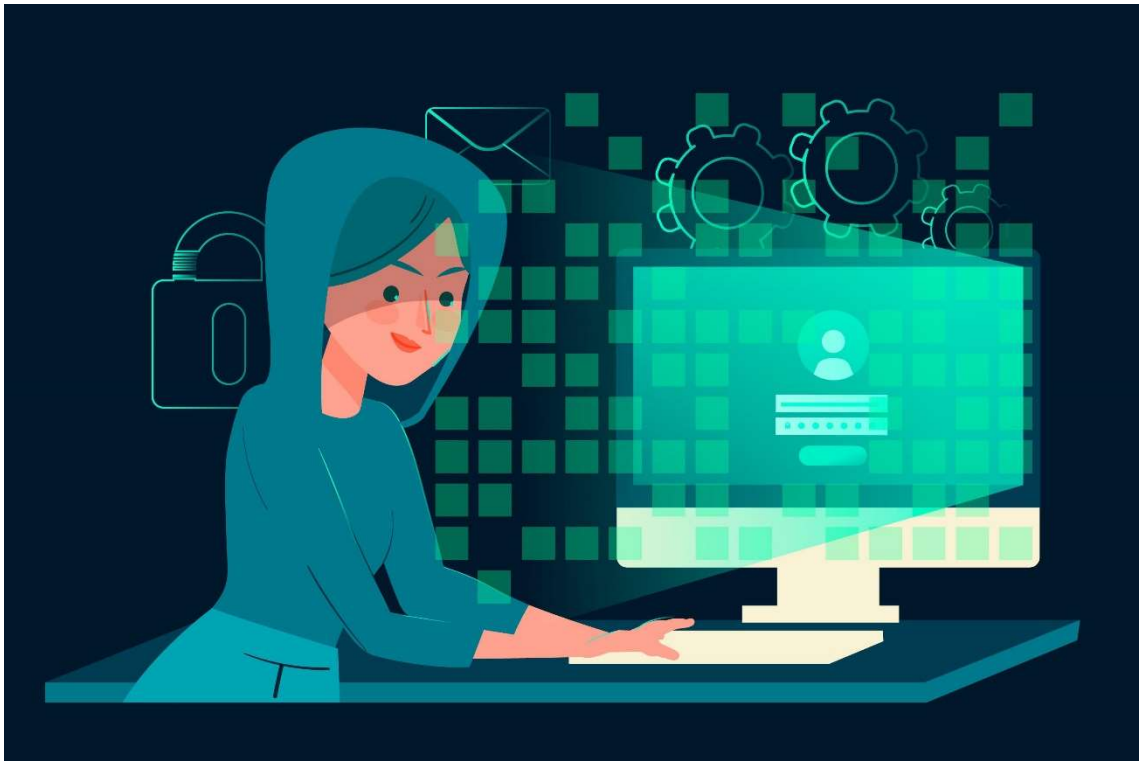
RELATÓRIO DE PENTEST

Laboratório de Desafio Final

Vai na Web / Curso de Cibersegurança

Rízia A. Araujo

Data de Entrega: 01/12/2025



Sumário

1. Introdução

Este relatório apresenta os resultados da avaliação de segurança conduzida no ambiente de laboratório disponibilizado para fins acadêmicos, com o objetivo de identificar vulnerabilidades, encontrar flags distribuídas pelo sistema e exercitar boas práticas de teste de intrusão. O ROE permitiu a exploração livre dos recursos do ambiente, desde que sem causar interrupção permanente no serviço.

2. Escopo

- **Endereço avaliado:** http://98.95.207.28/
- **Ambiente:** Laboratório educacional controlado
- **Objetivo:** Identificação de vulnerabilidades e coleta de flags posicionadas pelo instrutor
- **Ferramentas utilizadas:** (Nmap, Burp Suite, Dirb, Nikto, DevTools.)

3. Metodologia

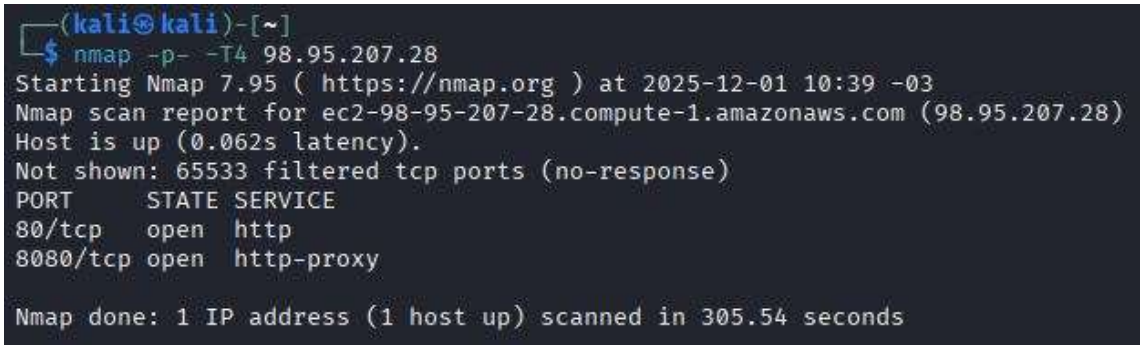
A avaliação seguiu etapas práticas utilizando ferramentas reais de pentest para identificação de falhas e coleta de flags. Abaixo, descrevo detalhadamente como cada processo foi executado:

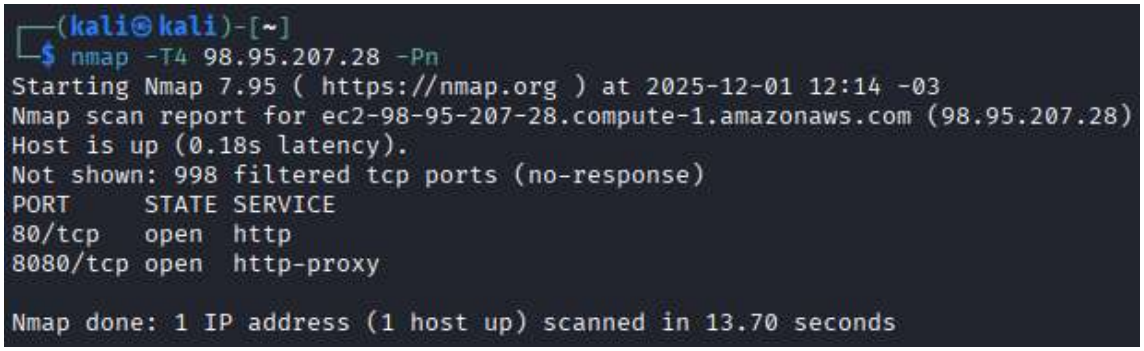
3.1 Reconhecimento com Nmap

O Nmap foi utilizado para mapear portas abertas, identificar serviços e possíveis versões vulneráveis. O comando utilizado seguiu a linha:

```
nmap -T4 -sV 98.95.207.28;
nmap -T4 -p- 98.95.207.28;
```

- **Objetivo:** descobrir portas, serviços e tecnologias expostas.
- **Impacto:** revela superfícies de ataque.





3.2 Varredura de Vulnerabilidades com Nikto

O Nikto foi executado para identificar configurações inseguras no servidor web:

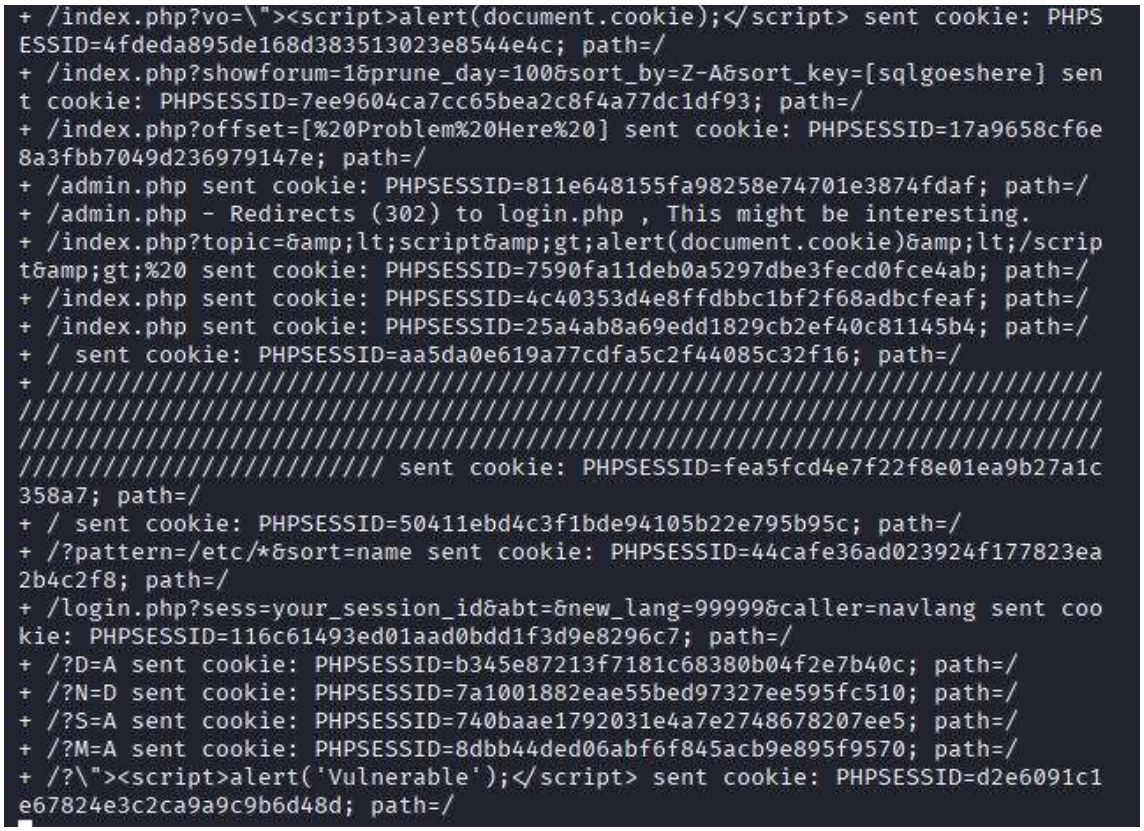
`nikto -h http://98.95.207.28/`

Achados relevantes:

- Arquivo robots.txt acessível.
- Cookie PHPSESSID sem flag *HttpOnly*.
- Ausência do header X-Frame-Options = vulnerabilidade a *Clickjacking*.
- Resposta HTTP 200 a diretórios potencialmente sensíveis.

```
┌──(kali㉿kali)-[~]
└─$ nikto -h http://98.95.207.28/
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          98.95.207.28
+ Target Hostname:    98.95.207.28
+ Target Port:        80
+ Start Time:         2025-12-01 12:49:39 (GMT-3)
---------------------------------------------------------------------------
+ Server: Apache/2.4.54 (Debian)
+ /: Retrieved x-powered-by header: PHP/7.4.33.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https:
//developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user
agent to render the content of the site in a different fashion to the MIME ty
pe. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities
/missing-content-type-header/
+ /: Cookie PHPSESSID created without the httponly flag. See: https://develop
er.mozilla.org/en-US/docs/Web/HTTP/Cookies
```

```
+ Server: Apache/2.4.54 (Debian)
+ /: Retrieved x-powered-by header: PHP/7.4.33.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https:
//developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user
agent to render the content of the site in a different fashion to the MIME ty
pe. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities
/missing-content-type-header/
+ /: Cookie PHPSESSID created without the httponly flag. See: https://develop
er.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /config/: Directory indexing found.
+ /robots.txt: Entry '/config/' is returned a non-forbidden or redirect HTTP
code (200). See: https://portswigger.net/kb/issues/00600600_robots-txt-file
+ /robots.txt: contains 4 entries which should be manually viewed. See: https
://developer.mozilla.org/en-US/docs/Glossary/Robots.txt
+ /: Web Server returns a valid response with junk HTTP methods which may cau
se false positives.
cc+ /index.php?download=/winnt/win.ini sent cookie: PHPSESSID=993ffb3244fe246
2a41070b7d7ae3747; path=/
l+ /index.php?download=/windows/win.ini sent cookie: PHPSESSID=f26b319cc75b7a
b83cad0fe2d55cc35e; path=/
e+ /index.php?download=/etc/passwd sent cookie: PHPSESSID=f5705962ad84936f29c
9c37c871e6267; path=/
a+ /index.php?|=../../../../../../../../../etc/passwd sent cookie: PHPSESSID=
8100078f8728cbb47268ed9c77ed6cfa; path=/
+ /index.php?page=../../../../../../../../../../etc/passwd sent cookie: PHPSE
SSID=da9c9644a581bba98cb2bd1c5ca3324a; path=/
```



3.3 Coleta de Informações com curl e grep

Após identificar o arquivo `robots.txt`, ele foi analisado com:

```
curl http://98.95.207.28/robots.txt
```

Em seguida, utilizou-se o grep para filtrar caminhos sensíveis:

```
curl http://98.95.207.28/robots.txt | grep -i "Disallow"
```

Foi identificado:

- /backup/database_backup_2024.sql – diretórios contendo informações sensíveis.
- Flag encontrada: **FLAG{r0b0ts_txt_l34k4g3}**.
- Ferramentas simples encontraram conteúdo sensível.

```
┌──(kali㉿kali)-[~]
└─$ curl http://98.95.207.28/robots.txt
User-agent: *
Disallow: /admin/
Disallow: /backup/
Disallow: /.git/
Disallow: /config/

# FLAG{r0b0ts_txt_l34k4g3}
# Arquivo de backup: /backup/database_backup_2024.sql
```

```
┌──(kali㉿kali)-[~]
└─$ curl http://98.95.207.28/robots.txt | grep -i "Disallow"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   169  100   169    0     0    643      0 --:--:-- --:--:-- --:--:--   645
Disallow: /admin/
Disallow: /backup/
Disallow: /.git/
Disallow: /config/
```

3.4 Investigação de Diretórios expostos

Durante a fase de reconhecimento, foi realizada uma tentativa de acesso ao diretório /backup/ para verificar a existência de arquivos expostos. O servidor respondeu corretamente com o código 404 Not Found (recurso inexistente), porém, a página de erro padrão revelou indevidamente a versão do software utilizado:

```
<address>Apache/2.4.54 (Debian) Server at 98.95.207.28 port 80</address>
```

Impacto (Risco): A exposição da versão exata (Apache 2.4.54 em Debian) permite que atacantes pesquisem por vulnerabilidades específicas (CVEs) conhecidas para essa versão, facilitando ataques direcionados ao servidor.

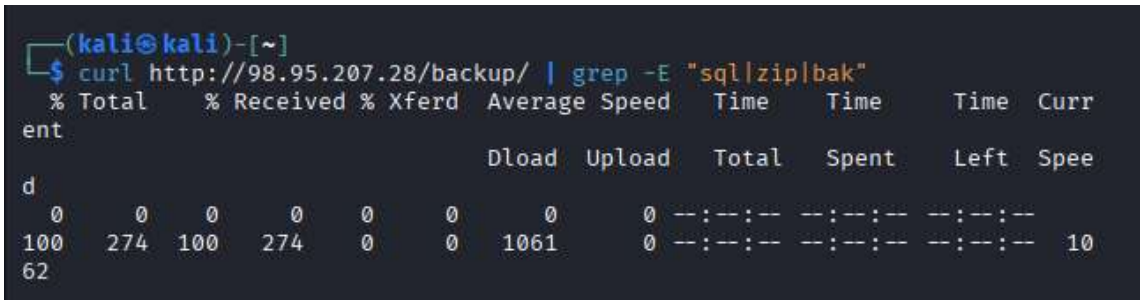
Recomendação Técnica: Para ocultar essas informações, é necessário ajustar as configurações de segurança do Apache.

```
┌──(kali㉿kali)-[~]
└─$ curl http://98.95.207.28/backup/
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.4.54 (Debian) Server at 98.95.207.28 Port 80</address>
</body></html>
```

Adicionalmente, foi feita uma filtragem no conteúdo dessa resposta buscando por extensões sensíveis (.sql, .zip, .bak). Nenhum arquivo foi encontrado e o tamanho da resposta limitou-se a **1061 bytes**.

Contudo, a página de erro revelou indevidamente a versão do software:

```
<address>Apache/2.4.54 (Debian) Server at 98.95.207.28 port 80</address>
```



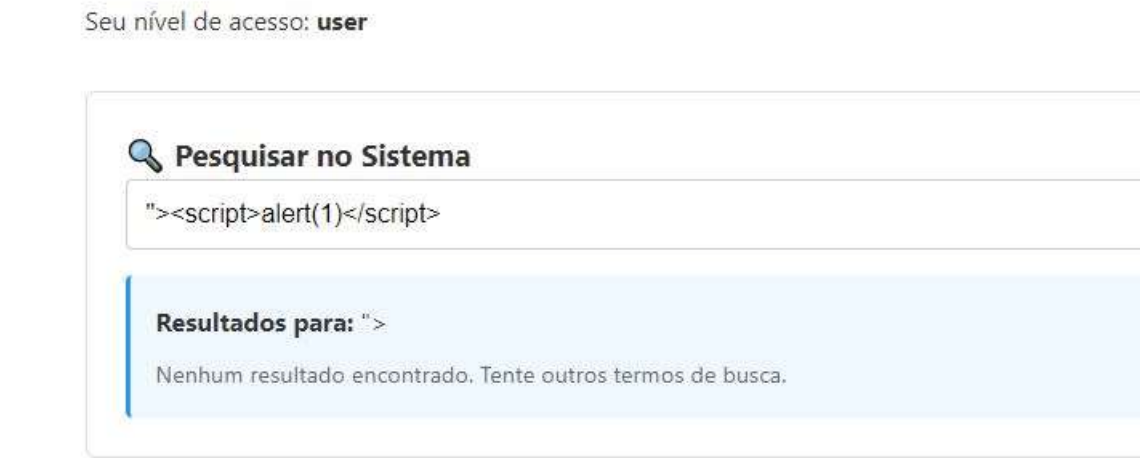
3.5 Investigação de Vulnerabilidades com DevTools

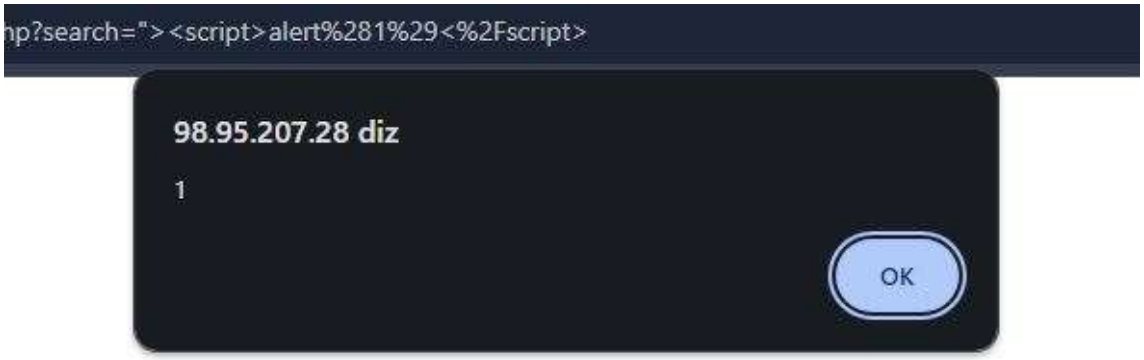
Foram utilizados conhecimentos de linguagens de desenvolvimento juntamente com o DevTools para revisar o comportamento do site no navegador:

- **XSS:** campos sem sanitização permitiam injeção e reflexão de scripts.
- **SQL Injection:** parâmetros GET/POST respondiam de forma anômala a caracteres especiais, permitindo login como admin (escalada de privilégios). Também foi encontrado **Debug Info** exposto na interface, indicando vazamento de informações sobre sessões, roles, IDs, caminhos internos e queries.
- Em um sistema mal configurado, um atacante poderia usar esses dados para tentar falsificar sessão (*session hijacking*) caso conseguisse definir o cookie ou parâmetro correto.
- **Escalada de privilégios:** ausência de verificação robusta de sessão e controle inadequado de permissões.

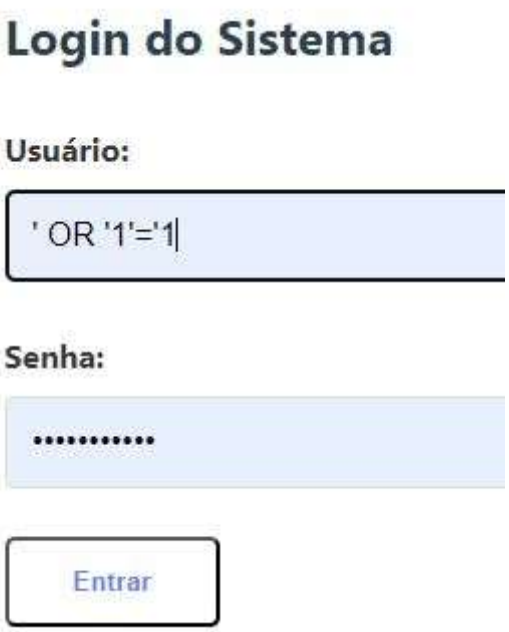
Flag encontrada: {sql_1nj3ct10n_m4st3r}.

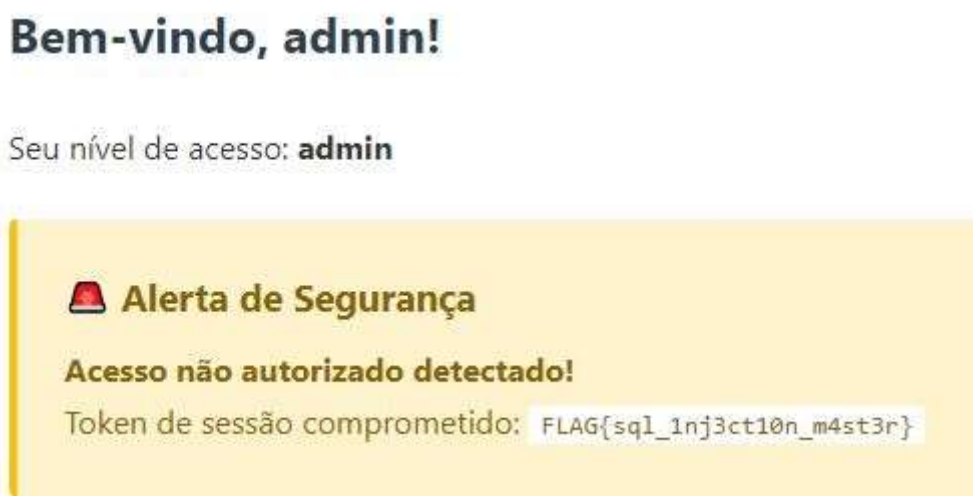
XSS Scripting:





SQL Injection:





4. Resultados Detalhados

4.1 Lista Consolidada de Vulnerabilidades Encontradas

1) Exposição de informações sensíveis via robots.txt

- **Descrição:** O arquivo robots.txt expõe diretórios internos não destinados ao público, incluindo /backup/.
- **Risco: Médio**
- **Impacto:** Permite que atacantes descubram rotas e arquivos sensíveis, possibilitando outras explorações.
- **Evidência:** Detecção de Disallow: /backup/ via curl e grep.
- **Flag associada:** FLAG{r0b0ts_txt_l34k4g3}
- **Recomendação:** Remover diretórios sensíveis do robots.txt e restringir acesso via configuração do servidor.

2) Possível vazamento de backup do banco de dados

- **Descrição:** O arquivo /backup/database_backup_2024.sql estava listado no robots.txt, indicando armazenamento de backup no diretório público.
- **Risco: Alto**
- **Impacto:** Um backup SQL exposto pode revelar credenciais, tabelas, hashes, dados pessoais e lógica da aplicação.
- **Evidência:** Caminho listado; tentativa de acesso retornou 404 (indica que o instrutor removeu posteriormente).
- **Recomendação:** Nunca armazenar backups no webroot; mover para diretórios externos ao servidor web.

3) Cookie PHPSESSID sem HttpOnly

- **Descrição:** A sessão do usuário é enviada sem a flag HttpOnly.
- **Risco: Médio**
- **Impacto:** Um atacante pode roubar o cookie via XSS e assumir a sessão do usuário.
- **Evidência:** Identificado via análise de cabeçalhos HTTP.
- **Recomendação:** Habilitar HttpOnly, Secure e SameSite.

4) Ausência de X-Frame-Options (Clickjacking)

- **Descrição:** O site pode ser carregado dentro de iframes externos.
- **Risco: Baixo a Médio**
- **Impacto:** O usuário pode ser induzido a clicar em elementos invisíveis, causando ações indesejadas.
- **Evidência:** Header ausente no response.
- **Recomendação:** Configurar X-Frame-Options: DENY ou SAMEORIGIN.

5) XSS (Cross-Site Scripting)

- **Descrição:** Campos sem sanitização permitiram testar injeção de JavaScript, refletindo conteúdo sem validação.
- **Risco: Alto**
- **Impacto:** Roubo de sessão, desfiguração visual, execução arbitrária de scripts.
- **Evidência:** Testes pelo DevTools mostraram comportamento vulnerável.
- **Recomendação:** Sanitizar entradas, validar parâmetros e implementar CSP.

6) SQL Injection com Escalada de Privilégios

- **Descrição:** Parâmetros GET/POST retornavam erros e funcionaram para logar como admin.
- **Risco: Crítico**
- **Impacto:** Controle total sobre contas, permissões e dados.
- **Evidência:** Login administrativo obtido; debug info exposta.
- **Flag associada:** {sql_1nj3ct10n_m4st3r}
- **Recomendação:** Utilizar prepared statements, remover mensagens de debug e reforçar controle de sessão.

7) Debug Info exposto na interface

- **Descrição:** Informações internas da aplicação eram exibidas publicamente.
- **Risco: Médio**
- **Impacto:** Ajuda o atacante a entender o funcionamento interno, parâmetros e erros.
- **Evidência:** Debug Info exibido na interface durante testes.
- **Recomendação:** Desativar debug em produção.

5. Flags Encontradas

Flag	Localização	Observação
FLAG{r0b0ts_txt_l34k4g3}	robots.txt / rota para /backup/	Registrada via curl + grep
FLAG{sql_1nj3ct10n_m4st3r}	Teste de SQL Injection	Obtida na exploração da falha
FLAG{b4s1c_s0urc3_code_1nsp3ct10n}	Tela inicial/HTML	DevTools

6. Recomendações Gerais

- Atualização de componentes e serviços
- Melhoria em autenticação e implementação de controles
- Remoção de arquivos expostos
- Restrição de diretórios administrativos
- Implementação de verificação de permissões e sanitização de entrada

7. Conclusão

A avaliação permitiu identificar múltiplas vulnerabilidades e localizar flags propostas pelo instrutor. O ambiente cumpriu seu propósito educacional, proporcionando um espaço seguro para prática de pentest.

O relatório consolida os achados e oferece recomendações que, se aplicadas, contribuem para um ambiente mais seguro em sistemas reais.