

ROADSEC  
2023 15.07.23



# Segurança no Front-end

Aprenda de uma vez por todas o que é necessário para  
construir e manter uma aplicação segura



# Larissa Azevedo

Front-end sênior no Grupo Boticário, natural de São Paulo


9+ anos na área de desenvolvimento

Compartilho conteúdos de programação e carreira em tecnologia nas redes sociais

Amo tudo referente à gatos, leitura e o universo da tecnologia







# Introdução à segurança em aplicações web

DO QUE E POR QUE PROTEGER  
MINHA APLICAÇÃO?

O QUE O FRONT-END TEM A VER  
COM ISSO?

DE QUEM É A RESPONSABILIDADE  
DE SEGURANÇA DA APLICAÇÃO?

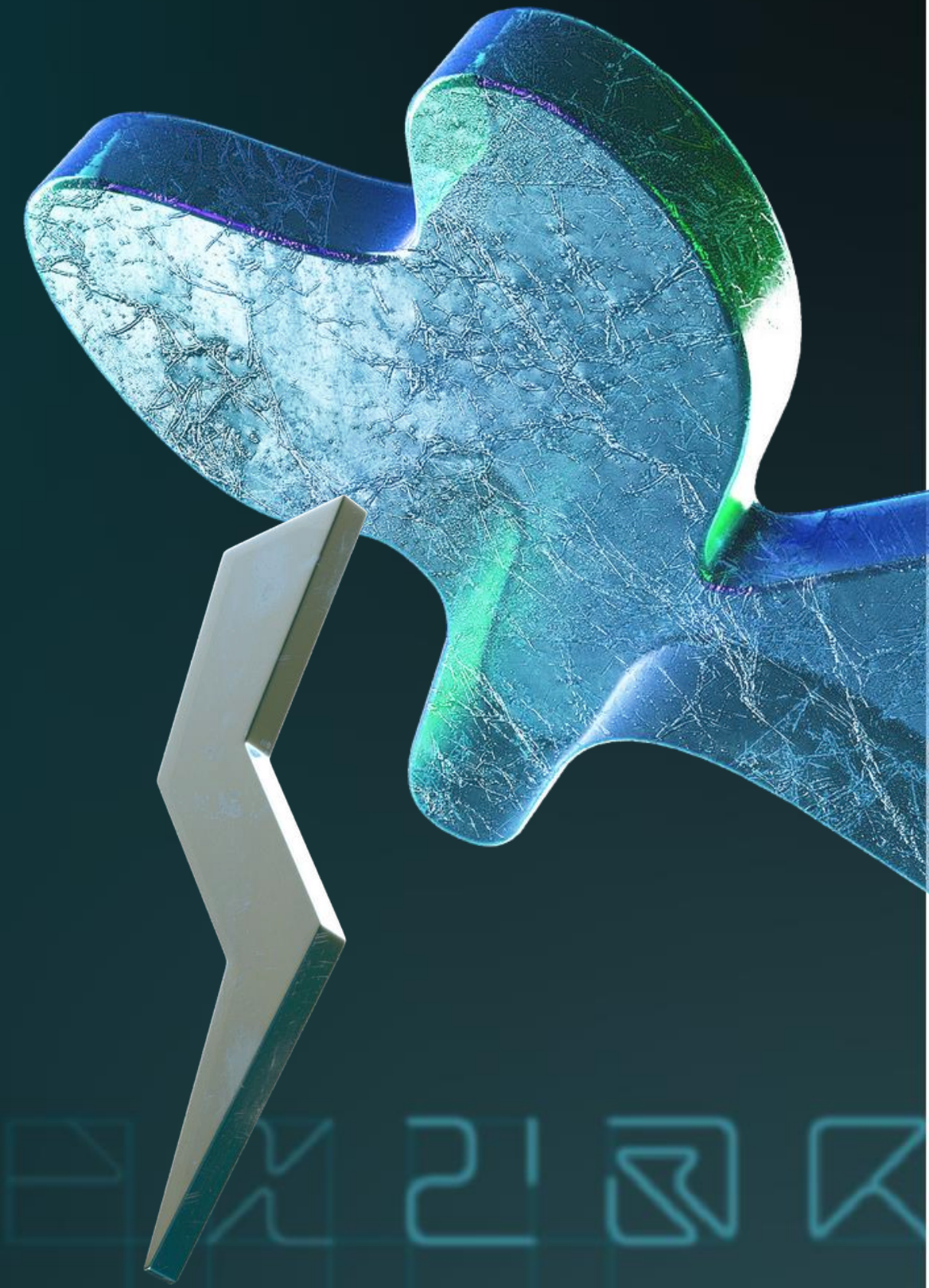


# OWASP

Open Worldwide Application Security Project

Organização sem fins lucrativos que visa melhorar a segurança de software em todo o mundo.

Fundada em 2001, a OWASP fornece recursos gratuitos e abertos para a comunidade de segurança de software, incluindo documentação, ferramentas, fóruns e eventos





# Principais ameaças de segurança para aplicações web

Vulnerabilidades mais comuns e  
como elas podem ser exploradas

Fonte: OWASP Top 10

CROSS-SITE SCRIPTING (XSS)

INJEÇÃO DE SQL

INJEÇÃO DE CÓDIGO

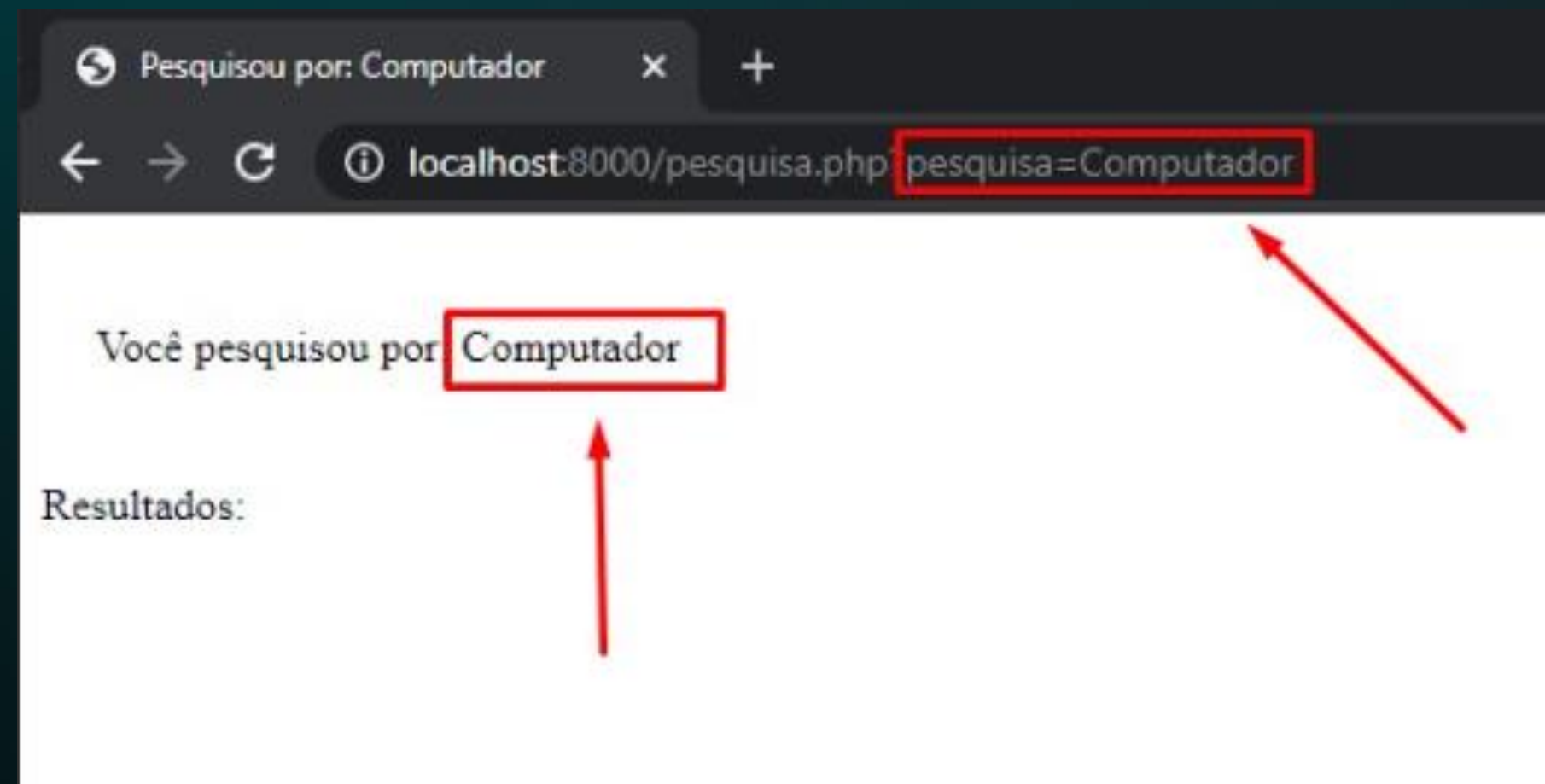
QUEBRA DE AUTENTICAÇÃO E SESSÃO

ATAQUES DE FORÇA BRUTA

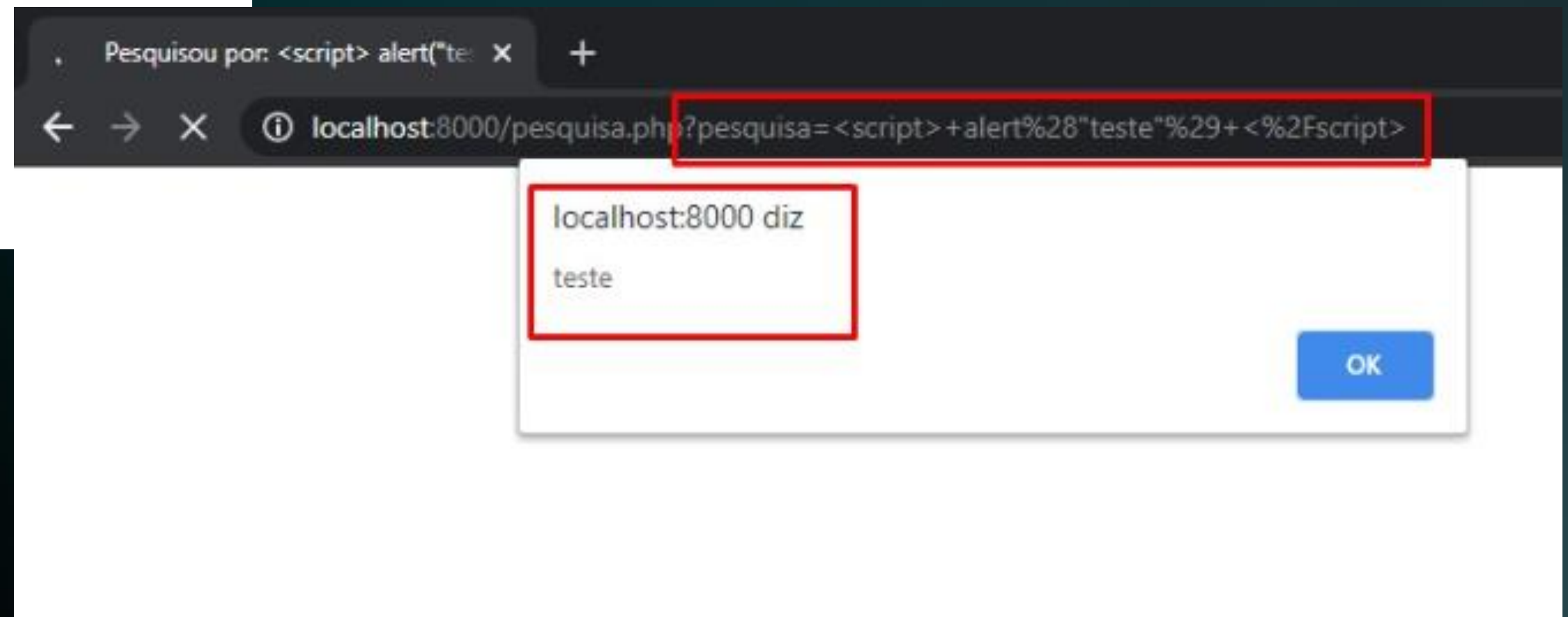
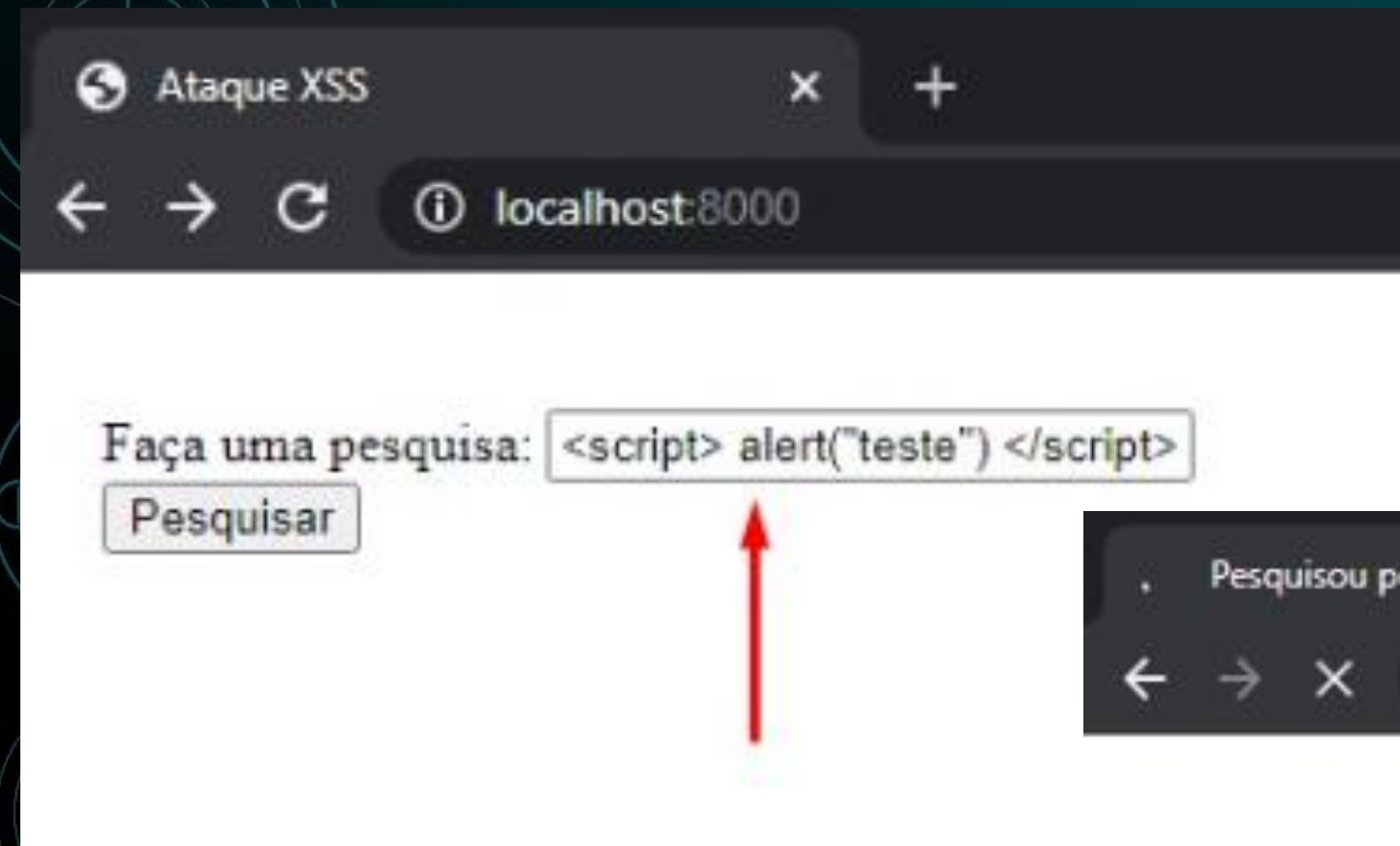
VULNERABILIDADES DE CONFIGURAÇÃO

# XSS ou Cross-Site Scripting

Um invasor injeta código malicioso em um site, que é então executado no navegador do usuário, permitindo que o invasor roube informações confidenciais, como cookies de autenticação



# XSS ou Cross-Site Scripting





# Como evitar XSS

Para prevenir o XSS, os desenvolvedores devem garantir que todas as entradas do usuário sejam validadas e filtradas antes de serem exibidas na página.

- Usar HTTPS
- Funções que impeçam a submissão de código nos campos de texto

```
var string = "<script>alert('XSS');</script>"`

console.log(string);
> "<script>alert('XSS');</script>"

console.log(`${encodeURIComponent(string)}`);
> "%22%3Cscript%3Ealert('XSS')%3B%3C%2Fscript%3E%22"
```



# Autenticação e autorização

Ataques e formas de prevenção



# Cookie

---

- Armazenados no navegador ou localmente
- Tem período de duração determinável
- São mais vulneráveis a acessos de terceiros e não dão segurança aos dados

# Token

---

- Strings criptografadas com informações de sessão
- Tem período de duração determinável
- São mais seguros para se armazenar informações de autenticação



# CSRF ou Cross-Site Request Forgery

Um invasor consegue inserir código malicioso em uma página web, que é então executado no navegador de um usuário sem o seu conhecimento ou consentimento.

## Cross-site Request Forgery Attack

1



### Forged Request

The attacker creates a forged request that will transfer \$10,000 from a bank

2



### Attacker

The attacker embeds a modified link into a website or email

3



### Victim

A victim clicks on the link, unknowingly sending the malicious request

4



### Bank's Server

The bank's web server receives the request, and transfers \$10,000 from the victim to the attacker

# Como evitar CSRF

---

Para prevenir um ataque CSRF, os desenvolvedores devem implementar a utilização de tokens CSRF e verificação de referência, que garante que a solicitação é originada do usuário legítimo e não de um atacante.

- SameSite – método mais moderno para combate à CSRF
- CSRF-Token em métodos HTTP
- Certificado SSL (Secure Socket Layer)



# Injeção de SQL

Um invasor insere comandos SQL em formulários de entrada ou URL, obtendo assim acesso não autorizado ao banco de dados.

```
// Login falso
SELECT * FROM usuarios WHERE username='$username' AND password='$password'

> ' OR 1=1 --'
SELECT * FROM usuarios WHERE username='' OR 1=1 -- ' AND password='$password'
```

# Como evitar injeção de SQL

---

Para prevenir a injeção de SQL, os desenvolvedores devem garantir que todas as entradas do usuário sejam validadas e filtradas antes de serem passadas para o banco de dados.

Também deve-se limitar privilégios do banco de dados e tratar as consultas.



# Ataques de força bruta

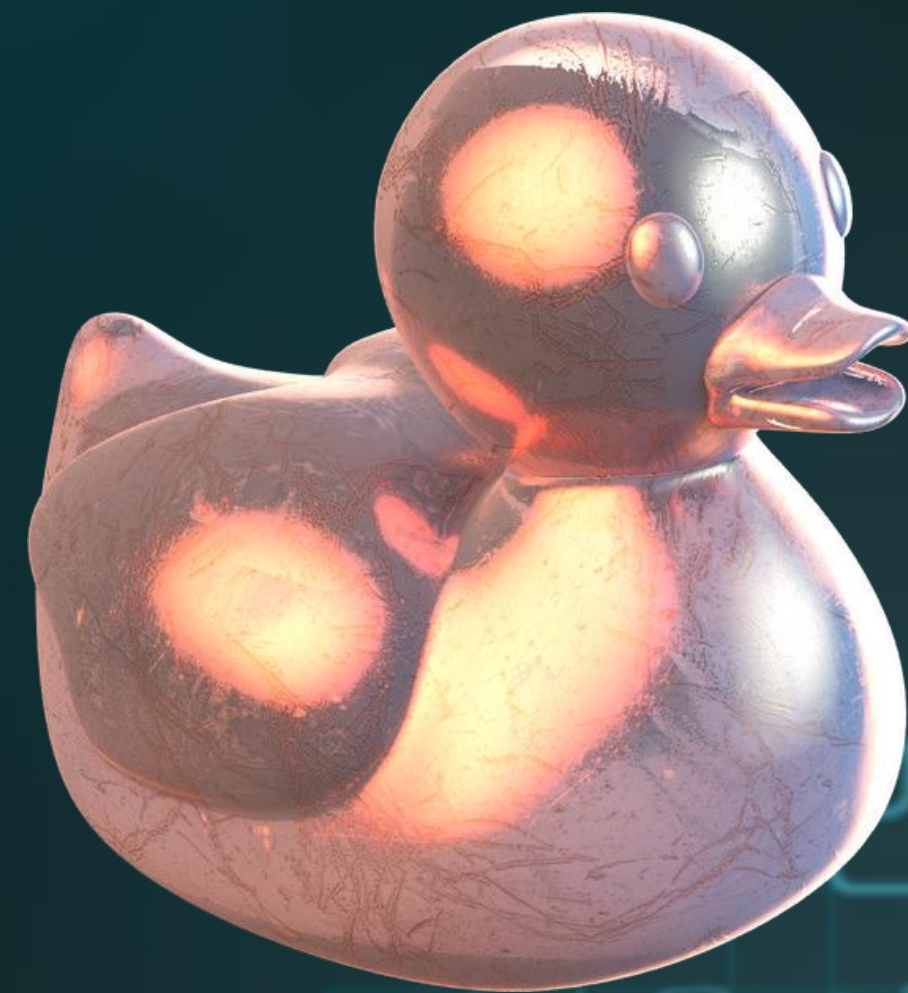
Um invasor tenta adivinhar as credenciais de autenticação de um usuário por meio de tentativas repetidas.

```
[80][http-get-form] host: 192.168.100.155 login: admin password: password
[80][http-get-form] host: 192.168.100.155 login: admin password: p@ssword
[80][http-get-form] host: 192.168.100.155 login: admin password: 12345
[80][http-get-form] host: 192.168.100.155 login: admin password: 1234567890
[80][http-get-form] host: 192.168.100.155 login: admin password: Password
[80][http-get-form] host: 192.168.100.155 login: admin password: 123456
[80][http-get-form] host: 192.168.100.155 login: admin password: 1234567
[80][http-get-form] host: 192.168.100.155 login: admin password: 12345678
[80][http-get-form] host: 192.168.100.155 login: admin password: 1q2w3e4r
[80][http-get-form] host: 192.168.100.155 login: admin password: 123
[80][http-get-form] host: 192.168.100.155 login: admin password: 1
[80][http-get-form] host: 192.168.100.155 login: admin password: 12
1 of 1 target successfully completed, 12 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-07-27 15:28:24
```

# Como evitar ataques de força bruta

---

Para prevenir ataques de força bruta, os desenvolvedores devem implementar limites de tentativas de login e exigir senhas fortes.

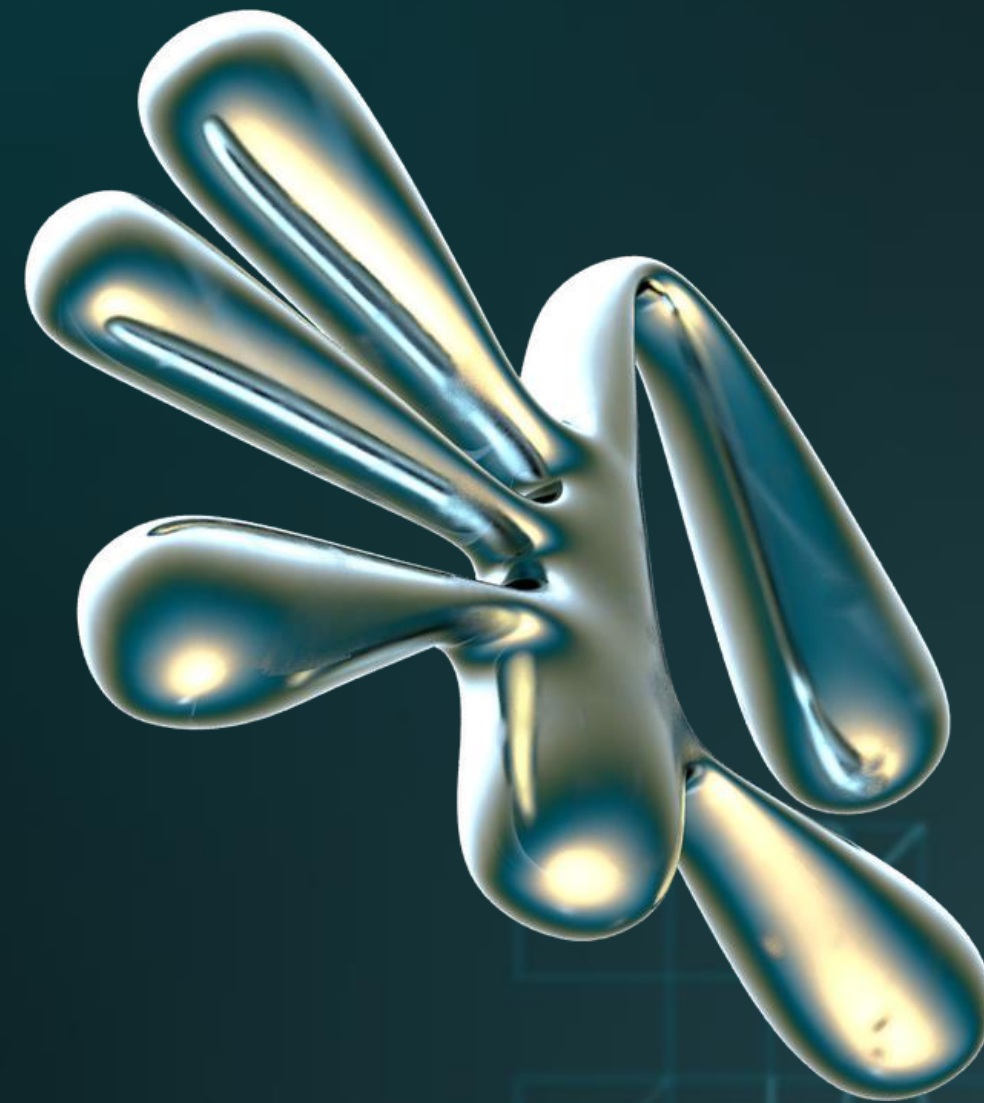




# Vulnerabilidades de configuração

---

Um aplicativo é configurado incorretamente, permitindo que invasores obtenham acesso não autorizado.



# Como evitar vulnerabilidades de configuração

---

Para prevenir vulnerabilidades de configuração, os desenvolvedores devem garantir que todas as configurações de segurança sejam corretamente configuradas e atualizadas regularmente.





# Autenticação e autorização

O básico funciona

Manter mensagens de erro genéricas

---

Utilizar captcha e MFA

---

Níveis de acesso bem gerenciados

---

Criptografia e tokens

---

Respostas HTTP corretas

---

Formulários semânticos



# Ferramentas e ações aliadas



OWASP Top Open

---

Ferramentas de testes de  
segurança: OWASP ZAP, Nmap, Burp  
Suite, Nessus

---


Instruir usuário a usar senhas fortes

---

Controle de tentativas de acesso

---

Criptografia



# Leve em conta isso também

O que você precisa saber

Quantidade de dados necessários  
em formulários

---

LGPD

---

Atrito e exposição (Dica: menos é  
mais)





# Em resumo

Utilizar autenticação e autorização para garantir que apenas usuários autorizados tenham acesso à aplicação e suas funcionalidades.

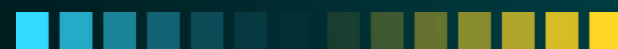
---

Atualizar regularmente todos os softwares e bibliotecas usados na aplicação, para corrigir vulnerabilidades conhecidas.

Utilizar criptografia para proteger dados sensíveis, atente-se também aos métodos de armazenamento de dados de sessão do usuário.

---

Validar todas as entradas do usuário e sanitizar dados para evitar injeção de código.



# ROAD SEC 2023

O MAIOR FESTIVAL HACKER DA AMÉRICA LATINA



**Larissa Azevedo**

Front-end Sênior e  
criadora de conteúdo em  
**@lari.sazevedo**

