



(inc

# Web Application Security

**codigo\_n3gro**

RedTeam Senior, ATHSec E.I

1011010  
0010001  
1010110

(^o^)



Google Developer Groups

Maputo

**`${WHOAMI}`**  
**codigo\_n3gro**

Head of Security at ATHSec;

[arlindo.junior@athsec.co](mailto:arlindo.junior@athsec.co)  
<https://athsec.co>



# Summary



1. O que é segurança
2. O que é Web Security
3. Por que é segurança?
4. Como definir segurança
5. O que você deve saber como desenvolvedor
6. As 10 principais ameaças à segurança segundo a OWASP
7. Erros comuns cometidos por desenvolvedores nesses pontos de controle
8. Como manter sua Aplicação Web Segura
9. Conclusão

0 que é segurança?

# O que é Web Security

A segurança de aplicativos da Web é um componente central de qualquer negócio baseado na web.

1. proteção de redes e sistemas



0 que você deve  
saber como  
desenvolvedor?

# Boas Práticas de Programação Segura

1. Validação dos Dados de Entrada
2. Controle de Acesso
3. Tratamento de Erros e Log
4. Proteção de Dados
5. Segurança nas comunicações
6. Configuração do Sistema
7. Segurança em Banco de Dados
8. Gerenciamento de Memória



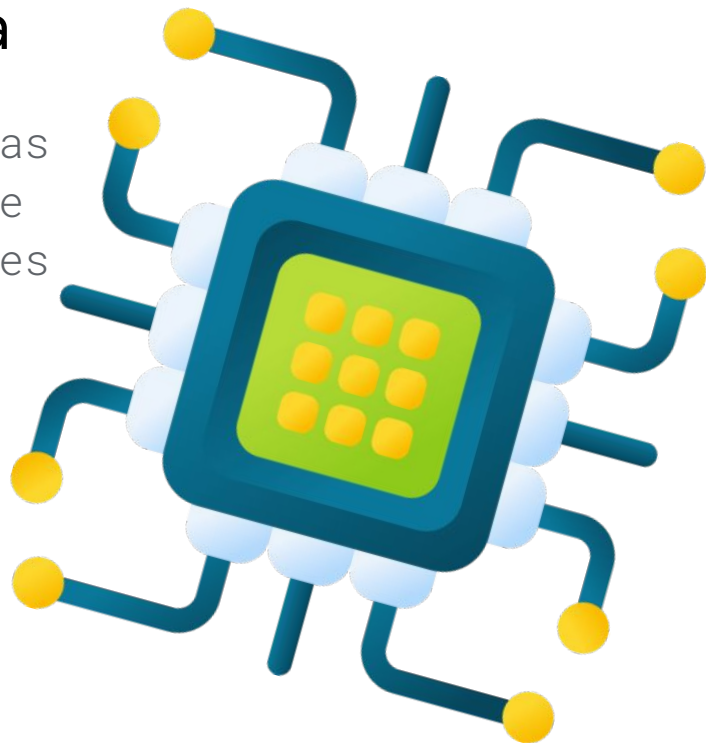
# Boas Práticas de Codificação Segura

1. Não permite que a aplicação execute comandos diretamente no sistema operacional
2. Restringir a geração e a alteração de código por parte dos usuários
3. Implementar atualizações de modo seguro
4. Utilizar mecanismos de verificação de integridade por **checksum** ou **hash**
5. Não transferir, diretamente, dados fornecidos pelo usuário para qualquer função de execução
6. dinâmica sem realizar o tratamento dos dados de modo adequado



# Boas Práticas de Codificação Segura

1. Revisar todas as aplicações secundárias
2. Proteger as variáveis compartilhadas e os recursos contra acessos concorrentes inapropriados



\^o^/

(index)

01001010101001

10110101100010

001

101

010

001

10


»

```
def plot_image(i, predictions_a,  
               true_label, img = true_label):  
    plt.grid(False)  
    plt.xticks([])  
    plt.yticks([])
```

```
    plt.imshow(img, cmap=plt.cm.binary)
```

```
    predicted_label = np.argmax(predictions_a[i])
```

```
    if predicted_label == true_label:
```

As 10 principais  
ameaças à segurança  
segundo a  OWASP®

: - )

</>

devf  
devf  
devf

# Open Web Application Security Project - OWASP

O OWASP, ou Projeto Aberto de Segurança em Aplicações Web, é uma comunidade online que cria e disponibiliza de forma gratuita:

1. Artigos
2. Metodologias
3. Documentação
4. Ferramentas
5. Tecnologias



2017

A01:2017-Injection

A02:2017-Broken Authentication

A03:2017-Sensitive Data Exposure

A04:2017-XML External Entities (XXE)

A05:2017-Broken Access Control

A06:2017-Security Misconfiguration

A07:2017-Cross-Site Scripting (XSS)

A08:2017-Insecure Deserialization

A09:2017-Using Components with Known Vulnerabilities

A10:2017-Insufficient Logging & Monitoring

2021

A01:2021-Broken Access Control

A02:2021-Cryptographic Failures

A03:2021-Injection

(New) A04:2021-Insecure Design

A05:2021-Security Misconfiguration

A06:2021-Vulnerable and Outdated Components

A07:2021-Identification and Authentication Failures

(New) A08:2021-Software and Data Integrity Failures

A09:2021-Security Logging and Monitoring Failures\*

(New) A10:2021-Server-Side Request Forgery (SSRF)\*

\* From the Survey

# A01:2021-Broken Access Control

O controle de acesso impõe a política de modo que os usuários não possam agir fora de suas permissões pretendidas.

1. Elevação de privilégios
2. Permitir a visualização ou edição da conta de outra pessoa
3. Violação do princípio de privilégio mínimo ou negação por padrão



# A01:2021-Broken Access Control - Como prevenir?

O controle de acesso impõe a política de modo que os usuários não possam agir fora de suas permissões pretendidas.



1. Permite que um usuário visualize ou edite a conta de outra pessoa;
2. Tenha privilégios que não deveria,
3. Ver o'que nao deveria;

# A02:2021-Falhas Criptograficas

A primeira coisa é determinar as necessidades de proteção dos dados em trânsito e em repouso.

1. LGPD
2. GDPR
3. HTTP/HTTPS
4. SSL
5. CRIPTOGRAFAR



# A02:2021-Falhas Criptograficas - Como prevenir?

A primeira coisa é determinar as necessidades de proteção dos dados em trânsito e em repouso.



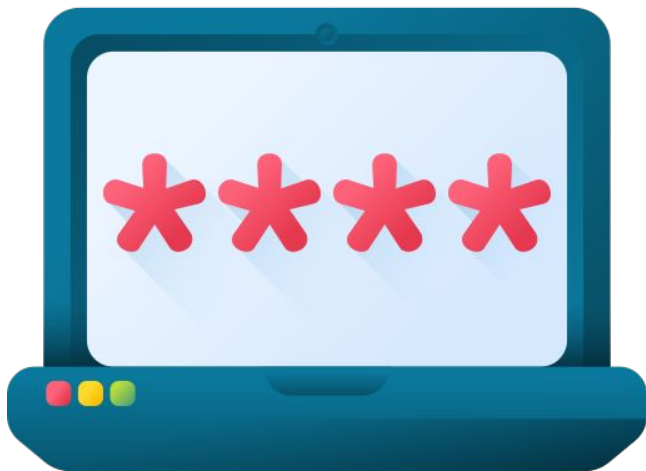
1. Certifique-se de criptografar todos os dados
2. Certifique-se de que algoritmos, protocolos e chaves de padrão forte e atualizados estejam em vigo
3. Criptografe todos os dados em trânsito com protocolos seguros
4. Sempre use criptografia autenticada em vez de apenas criptografia



# A03:2021-Injecao

Um aplicativo é vulnerável a ataques quando:

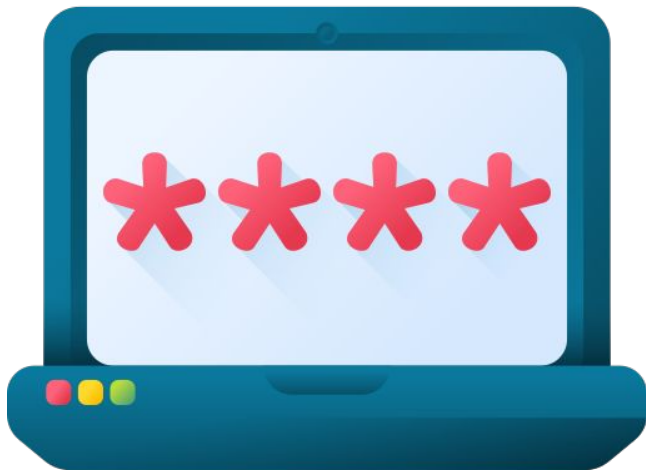
1. Os dados fornecidos pelo usuário não são validados



## A03:2021-Injecao - Como prevenir?

Um aplicativo é vulnerável a ataques quando:

1. Validação de entrada positiva
2. Use uma API segura



<https://owasp.org/Top10/>

<https://cheatsheetseries.owasp.org/>



\^o^/

(index)

010010101010001

0110101100010

001

101

010

001

110

»

```
def plot_images, accuracy, loss, true_label, img = true_label,
plt.grid(False)
plt.xticks([])
plt.yticks([])

plt.imshow(img, cmap=plt.cm.binary)

predicted_label = np.argmax(out)
if predicted_label == true_label:
    color = 'blue'
else:
    color = 'yellow'
```

Erros comuns  
cometidos por  
desenvolvedores na  
codificação segura

:—)

</>

devf  
devf  
devf

1. Validação
2. Testes Tardios
3. Atualização



# Como manter sua Aplicação Segura

# Como manter sua Aplicação Segura

1. Limite os privilégios do usuário de banco de dados da aplicação;
2. Criptografar dados de querystring
3. Criptografia de dados de autenticação
4. Utilização de HTTPS
5. Criptografia de Web.config
6. Segurança de visualização de documentos/anexos
7. Correto tratamento de upload de arquivos
8. Não subestime o conhecimento técnico do usuário

# Como manter sua Aplicação Segura

1. Se preocupe com logs
2. Contratação de profissionais especializados
3. Treine a equipe e invista no seu próprio conhecimento





```
predicted_label = np.argmax  
if predicted_label == true  
    color = 'blue'  
else:  
    color = 'yellow'
```

**SOCIAL MEDIA**

@arlindo0x73

101001  
100010  
011001  
100101  
011010



= ]

“Previna danos  
Operacionais e de  
Reputação, previna  
perdas Financeiras  
e de Clientes.”

@codigo\_n3gro



```
def plot_image(image, true_label, predicted_label):  
    plt.grid(True)  
    plt.xticks(10)  
    plt.yticks(10)  
  
    plt.imshow(image)  
  
    predicted_label = predicted_label  
    if predicted_label == true_label:  
        color = 'green'  
    else:  
        color = 'red'
```

DUVIDAS?

101001  
100010  
011001  
100101  
011010  
010001  
010110

= ]

