**UNIVERSIDADE TUIUTI DO PARANÁ**

**CAMILA SARA**

**THIAGO SCHWEDER SOUZA**
**NICKOLAS CONKE**

# SEGURANÇA EM APLICATIVOS MOBILE: RISCOS, VULNERABILIDADES E BOAS PRÁTICAS

**CURITIBA**
**2025**

**CAMILA SARA**
**THIAGO SCHWEDER SOUZA**
**NICKOLAS CONKE**

# SEGURANÇA EM APLICATIVOS MOBILE: RISCOS, VULNERABILIDADES E BOAS PRÁTICAS

Trabalho apresentado ao curso de Análise e Desenvolvimento de Sistemas da Universidade Tuiuti do Paraná, como requisito à obtenção ao grau de Bacharel.

Orientador: Prof. Chauã Queirolo

**CURITIBA**
**2025**

## RESUMO

Este trabalho aborda a segurança em aplicativos móveis, um fator crítico devido ao uso crescente para atividades sensíveis. Analisa os principais desafios e apresenta soluções técnicas e preventivas, cobrindo vulnerabilidades em apps Android e iOS, como uso inadequado de plataforma, armazenamento e comunicação inseguros, e autenticação fraca.

São detalhadas boas práticas de segurança, incluindo o uso de criptografia (AES-256, RSA 4096+, TLS 1.3), autenticação multifator, e armazenamento seguro de dados (Android Keystore, iOS Keychain). Discute-se a importância das permissões de aplicativos e da privacidade do usuário, com foco em regulamentações como GDPR e LGPD, e o conceito de "Privacy by Design".

A pesquisa examina casos reais de falhas de segurança (e.g., Strava, WhatsApp, Santander, Uber, Telegram), destacando padrões comuns e lições aprendidas. Apresenta ferramentas e técnicas de análise de segurança, como SAST (MobSF, QARK), DAST (Drozer, OWASP ZAP), SCA, e análise de tráfego, enfatizando sua integração no ciclo de desenvolvimento.

Finalmente, um checklist de segurança abrangente é fornecido para desenvolvedores, cobrindo fases de planejamento, desenvolvimento, testes, lançamento e manutenção, com seções específicas para Android e iOS. A conclusão reforça que a segurança mobile exige uma abordagem holística e contínua, sugerindo futuras pesquisas em segurança de frameworks cross-platform e IA.



**Palavras-chave**: Segurança mobile; Vulnerabilidades; Desenvolvimento seguro; Criptografia; Privacidade.

## LISTA DE SIGLAS E ACRÔNIMOS

| | |
|---|---|
| API | Application Programming Interface |
| GDPR | General Data Protection Regulation |
| HTTPS | Hypertext Transfer Protocol Secure |
| LGPD | Lei Geral de Proteção de Dados |
| MITM | Man-in-the-Middle |
| OWASP | Open Web Application Security Project |
| SAST | Static Application Security Testing |
| DAST | Dynamic Application Security Testing |
| SCA | Software Composition Analysis |
| SSL | Secure Sockets Layer |
| TLS | Transport Layer Security |

**SUMÁRIO**


**1 INTRODUÇÃO . . . . . 6**
**2 PRINCIPAIS VULNERABILIDADES EM APPS ANDROID E IOS . . . . 7**
2.1 PANORAMA GERAL DE VULNERABILIDADES MOBILE . . . . . 7
2.2 VULNERABILIDADES ESPECÍFICAS POR PLATAFORMA . . . . . 7
2.2.1 Android . . . . . 7
2.2.2 iOS . . . . . 7
2.3 VULNERABILIDADES COMUNS A AMBAS PLATAFORMAS . . . . . 7
2.4 TENDÊNCIAS EMERGENTES . . . . . 8
**3 BOAS PRÁTICAS DE SEGURANÇA . . . . . 9**
3.1 CRIPTOGRAFIA EM APLICATIVOS MÓVEIS . . . . . 9
3.1.1 Implementação por Plataforma . . . . . 9
3.2 AUTENTICAÇÃO E AUTORIZAÇÃO SEGURAS . . . . . 9
3.3 ARMAZENAMENTO SEGURO DE DADOS . . . . . 9
3.3.1 Classificação de Dados . . . . . 9
3.3.2 Opções de Armazenamento por Plataforma . . . . . 10
3.4 PROTEÇÃO DE COMUNICAÇÕES DE REDE . . . . . 10
**4 PERMISSÕES DE APPS E PRIVACIDADE DO USUÁRIO . . . . . 11**
4.1 SISTEMAS DE PERMISSÕES EM PLATAFORMAS MÓVEIS . . . . . 11
4.1.1 Android . . . . . 11
4.1.2 iOS . . . . . 11
4.2 BOAS PRÁTICAS PARA GERENCIAMENTO DE PERMISSÕES . . . . 11
4.3 PRIVACIDADE DO USUÁRIO E CONFORMIDADE REGULATÓRIA . . 11
4.3.1 Principais Regulamentações . . . . . 11
4.3.2 Privacy by Design . . . . . 11
4.3.3 Implementação de Controles de Privacidade . . . . . 11
**5 CASOS REAIS DE FALHAS DE SEGURANÇA . . . . . 12**
5.1 ANÁLISE DE INCIDENTES SIGNIFICATIVOS . . . . . 12
5.1.1 Vazamento de Dados Sensíveis . . . . . 12
5.1.2 Falhas de Autenticação e Autorização . . . . . 12
5.1.3 Implementação Inadequada de Criptografia . . . . . 12
5.2 PADRÕES COMUNS EM FALHAS DE SEGURANÇA . . . . . 12
5.3 LIÇÕES APRENDIDAS . . . . . 12
**6 FERRAMENTAS E TÉCNICAS DE ANÁLISE DE SEGURANÇA . . . . 13**
6.1 CATEGORIAS DE ANÁLISE DE SEGURANÇA . . . . . 13
6.2 PRINCIPAIS FERRAMENTAS . . . . . 13
6.2.1 Análise Estática (SAST) . . . . . 13
6.2.2 Análise Dinâmica (DAST) . . . . . 13

6.2.3 Análise de Tráfego . . . . . 13
6.2.4 Análise de Composição (SCA) . . . . . 13
6.3 INTEGRAÇÃO NO CICLO DE DESENVOLVIMENTO . . . . . 13
**7 CHECKLIST DE SEGURANÇA PARA DESENVOLVEDORES . . . . . 14**
7.1 FASE DE PLANEJAMENTO E DESIGN . . . . . 14
7.2 FASE DE DESENVOLVIMENTO . . . . . 14
7.3 FASE DE TESTES . . . . . 14
7.4 FASE DE LANÇAMENTO E MANUTENÇÃO . . . . . 15
7.5 CHECKLIST ESPECÍFICO POR PLATAFORMA . . . . . 15
**8 CONCLUSÃO . . . . . 16**
**REFERÊNCIAS . . . . . 17**

# 1 INTRODUÇÃO

O uso crescente de aplicativos móveis para atividades sensíveis torna a segurança um fator crítico. Com bilhões de downloads anuais (DEPARTMENT, 2023), a segurança mobile enfrenta desafios únicos, como ambientes menos controlados e armazenamento local de dados sensíveis (FOUNDATION, 2023). Vulnerabilidades comuns e práticas inseguras comprometem a privacidade e a integridade dos dados, com cerca de 46% dos apps apresentando riscos (SYMANTEC, 2022). Falhas de segurança resultam em roubo de identidade, perdas financeiras para usuários e danos à reputação/penalidades regulatórias para empresas (GDPR, LGPD) (UNION, 2018; BRASIL, 2018).

Este trabalho visa analisar os desafios de segurança em apps móveis e propor soluções. Os objetivos incluem identificar vulnerabilidades (Android/iOS), apresentar boas práticas (criptografia, autenticação, armazenamento), discutir permissões e privacidade, analisar casos reais, explorar ferramentas de análise e criar um checklist para desenvolvedores. A metodologia baseia-se em revisão bibliográfica, análise de casos e relatórios técnicos.

# 2 PRINCIPAIS VULNERABILIDADES EM APPS ANDROID E IOS

## 2.1 PANORAMA GERAL DE VULNERABILIDADES MOBILE

O ecossistema mobile possui desafios de segurança únicos. As principais vulnerabilidades, segundo Foundation (2023), incluem uso inadequado de plataforma, armazenamento e comunicação inseguros, autenticação fraca e criptografia insuficiente. Relatórios indicam que 83% dos apps móveis contêm ao menos uma vulnerabilidade, sendo 38% de alta gravidade (VERACODE, 2022).

## 2.2 VULNERABILIDADES ESPECÍFICAS POR PLATAFORMA

### 2.2.1 Android

A plataforma Android, por sua natureza aberta e fragmentada, apresenta desafios como:

a) **Fragmentação do sistema**: Dificulta patches de segurança uniformes (APPBRAIN, 2023).

b) **Permissões excessivas**: Aumentam a superfície de ataque.

c) **Componentes exportados**: Podem ser explorados se sem restrições.

d) **WebView inseguro**: Permite execução de JavaScript malicioso.

e) **Armazenamento externo inseguro**: Dados acessíveis por outros apps.

### 2.2.2 iOS

Mesmo sendo mais controlado, o iOS possui vulnerabilidades:

a) **Jailbreak**: Contorna restrições de segurança.

b) **Armazenamento inseguro de dados**: Uso inadequado de Keychain ou texto claro.

c) **Validação inadequada de certificados SSL**: Ignora erros de validação.

d) **URL Schemes inseguros**: Permite injeção de dados ou acesso restrito.

e) **Clipboard vulnerável**: Dados sensíveis acessíveis por outros apps.

## 2.3 VULNERABILIDADES COMUNS A AMBAS PLATAFORMAS

Comuns a Android e iOS:

a) **Comunicação insegura**: Falha em HTTPS ou validação de certificados.

b) **Autenticação fraca**: Tokens sem expiração, MFA ausente, recuperação de senha fraca.

c) **Criptografia insuficiente**: Algoritmos obsoletos ou implementação incorreta.

d) **Injeção de código**: SQL Injection e XSS em WebViews.

e) **Vazamento de dados**: Armazenamento inadvertido em caches ou logs.

## 2.4 TENDÊNCIAS EMERGENTES

Novas vulnerabilidades incluem:

a) **Frameworks cross-platform**: Riscos específicos em React Native, Flutter, Xamarin.

b) **Ataques baseados em sensores**: Coleta de informações sem permissões explícitas.

c) **Vulnerabilidades em biometria**: Falhas em reconhecimento facial/impressão digital.

d) **Ataques de side-channel**: Exploram vazamentos de informação.

# 3 BOAS PRÁTICAS DE SEGURANÇA

## 3.1 CRIPTOGRAFIA EM APLICATIVOS MÓVEIS

A criptografia é vital para proteger dados. Princípios: usar bibliotecas estabelecidas, manter-se atualizado, gerenciar chaves com segurança e adotar defesa em profundidade.

QUADRO 1 – Algoritmos e protocolos criptográficos recomendados.

| **Tipo** | **Recomendados** | **Obsoletos** |
|---|---|---|
| Criptografia Simétrica | AES-256 (GCM/CBC) | DES, 3DES, RC4, AES-ECB |
| Criptografia Assimétrica | RSA (4096+), ECC (P-256/384) | RSA < 2048 bits |
| Funções Hash | SHA-256/384/512, BLAKE2 | MD5, SHA-1 |
| Transporte | TLS 1.3, TLS 1.2 seguro | SSL 3.0, TLS 1.0/1.1 |
| Derivação de Chave | PBKDF2 (10000+), Argon2 | Funções hash simples |

FONTE: Standards e Technology, 2020

### 3.1.1 Implementação por Plataforma

**Android:** Android Keystore System, Jetpack Security, Tink. **iOS:** Keychain Services, CommonCrypto, CryptoKit.

## 3.2 AUTENTICAÇÃO E AUTORIZAÇÃO SEGURAS

Práticas: Autenticação multifator (MFA), políticas de senha fortes, limitação de tentativas, tokens com expiração curta, biometria nativa.

## 3.3 ARMAZENAMENTO SEGURO DE DADOS

### 3.3.1 Classificação de Dados

Classificar dados como: Altamente Sensível (credenciais), Sensível (pessoais), Interno (configurações), Público (catálogo).

### 3.3.2 Opções de Armazenamento por Plataforma

**Android:** EncryptedSharedPreferences, EncryptedFile, Room com SQLCipher, Android Keystore. **iOS:** Keychain, Data Protection API, CoreData com NSFileProtection, Secure Enclave.

## 3.4 PROTEÇÃO DE COMUNICAÇÕES DE REDE

Práticas: Forçar HTTPS, SSL Pinning, validação de certificados, TLS 1.2+, proteção contra ataques de replay.

# 4 PERMISSÕES DE APPS E PRIVACIDADE DO USUÁRIO

## 4.1 SISTEMAS DE PERMISSÕES EM PLATAFORMAS MÓVEIS

### 4.1.1 Android

Evoluiu para modelo em tempo de execução (Android 6.0+). Categorias: Normal, Perigosas, Signature, Special. Permissões solicitadas contextualmente, com grupos e opções granulares.

### 4.1.2 iOS

Modelo em tempo de execução. Permissões específicas por recurso (localização, fotos), com níveis de acesso ("Sempre", "durante o uso", "uma vez") e explicações obrigatórias (Info.plist).

## 4.2 BOAS PRÁTICAS PARA GERENCIAMENTO DE PERMISSÕES

a) **Privilégio mínimo**: Solicitar apenas o necessário.

b) **Solicitação contextual**: No momento do uso, com contexto claro.

c) **Explicações claras**: Razões específicas para cada permissão.

d) **Degradação elegante**: Funcionar com funcionalidade reduzida se negadas.

e) **Revisão periódica**: Avaliar permissões regularmente.

## 4.3 PRIVACIDADE DO USUÁRIO E CONFORMIDADE REGULATÓRIA

### 4.3.1 Principais Regulamentações

**GDPR** (UE): Consentimento, direito ao esquecimento, portabilidade. **LGPD** (Brasil): Similar ao GDPR. **CCPA**/**CPRA** (Califórnia): Direito de saber, exclusão, opt-out. **Políticas de lojas**: Exigem divulgação de coleta de dados.

### 4.3.2 Privacy by Design

Privacidade deve ser considerada desde o início: proativo, preventivo, como configuração padrão, incorporada ao design, funcionalidade total, segurança de ponta a ponta, visibilidade, transparência e respeito ao usuário.

### 4.3.3 Implementação de Controles de Privacidade

Política de privacidade clara, controles granulares, minimização de dados, anonimização/pseudonimização, transparência sobre compartilhamento com terceiros.

# 5 CASOS REAIS DE FALHAS DE SEGURANÇA

## 5.1 ANÁLISE DE INCIDENTES SIGNIFICATIVOS

### 5.1.1 Vazamento de Dados Sensíveis

**Strava (2018):** Revelou localizações militares via "mapa de calor"(HSU, 2018). Lição: Dados inofensivos podem ser críticos agregados. **WhatsApp (2019):** Spyware via buffer overflow em chamadas VoIP (NEWMAN, 2019). Lição: Vulnerabilidades no cliente podem comprometer apps criptografados.

### 5.1.2 Falhas de Autenticação e Autorização

**Santander (2020):** Falha contornou 2FA (OSBORNE, 2020). Lição: Autenticação deve ser verificada em cada requisição sensível. **Uber (2017):** Dados de 57 milhões expostos por credenciais no GitHub (ISAAC *et al.*, 2017). Lição: Credenciais nunca em repositórios de código.

### 5.1.3 Implementação Inadequada de Criptografia

**Telegram (2018):** Mensagens "secretas"em texto claro em banco de dados não criptografado (CIMPANU, 2018). Lição: Criptografia deve proteger dados em todo o ciclo de vida.

## 5.2 PADRÕES COMUNS EM FALHAS DE SEGURANÇA

Padrões recorrentes: falta de modelagem de ameaças, implementação incorreta de segurança, dependência de obscuridade, falha no privilégio mínimo, gestão inadequada de dependências, e resposta inadequada a incidentes.

## 5.3 LIÇÕES APRENDIDAS

a) Modelagem de ameaças formal no início.

b) Gestão rigorosa de dependências.

c) Revisões de código e testes de penetração focados em segurança.

d) Planos de resposta a incidentes.

e) Logging e monitoramento abrangentes.

f) Programa de segurança contínuo.

# 6 FERRAMENTAS E TÉCNICAS DE ANÁLISE DE SEGURANÇA

## 6.1 CATEGORIAS DE ANÁLISE DE SEGURANÇA

a) **Análise Estática (SAST):** Examina código-fonte sem executar.

b) **Análise Dinâmica (DAST):** Testa o app em execução.

c) **Análise de Composição (SCA):** Identifica vulnerabilidades em componentes de terceiros.

d) **Análise de Tráfego:** Monitora comunicações de rede.

e) **Análise Forense:** Examina artefatos no dispositivo.

## 6.2 PRINCIPAIS FERRAMENTAS

### 6.2.1 Análise Estática (SAST)

MobSF, QARK, Checkmarx/Fortify, SonarQube.

### 6.2.2 Análise Dinâmica (DAST)

Drozer, OWASP ZAP/Burp Suite, Frida, Appium.

### 6.2.3 Análise de Tráfego

Wireshark, Charles Proxy/mitmproxy, Packet Capture.

### 6.2.4 Análise de Composição (SCA)

OWASP Dependency-Check, Snyk, WhiteSource/Black Duck.

## 6.3 INTEGRAÇÃO NO CICLO DE DESENVOLVIMENTO

Para eficácia: **Shift Left** (segurança desde o início), **Automação** (CI/CD), **Feedback rápido**, **Educação contínua**, **Métricas**.

# 7 CHECKLIST DE SEGURANÇA PARA DESENVOLVEDORES

Este capítulo apresenta um checklist abrangente de segurança para desenvolvedores, consolidando as boas práticas.

## 7.1 FASE DE PLANEJAMENTO E DESIGN

a) Modelagem de ameaças.
b) Requisitos de segurança.
c) Classificação de dados.
d) Estratégia de criptografia.
e) Estratégia de autenticação/autorização.
f) Requisitos regulatórios.
g) Política de permissões (privilégio mínimo).
h) Estratégia de logging/monitoramento.

## 7.2 FASE DE DESENVOLVIMENTO

a) HTTPS para todas as comunicações.
b) Validação de certificados SSL/TLS.
c) SSL Pinning para conexões críticas.
d) Algoritmos criptográficos fortes.
e) Armazenamento seguro de dados sensíveis.
f) Autenticação forte (MFA).
g) Solicitar permissões apenas quando necessárias.
h) Timeout de sessão e invalidação de tokens.
i) Sanitização e validação de entradas.
j) Proteção contra força bruta.

## 7.3 FASE DE TESTES

a) Análise estática (SAST).
b) Análise dinâmica (DAST).
c) Verificação de componentes de terceiros (SCA).
d) Testes de penetração mobile.

e) Teste de armazenamento em dispositivos com root/jailbreak.

f) Verificação de SSL/TLS.

g) Teste de autenticação/autorização.

h) Verificação de permissões.

## 7.4 FASE DE LANÇAMENTO E MANUTENÇÃO

a) Mecanismo de atualização segura.

b) Monitoramento de vulnerabilidades em dependências.

c) Canal para relatórios de vulnerabilidades.

d) Processo de resposta rápida a vulnerabilidades.

e) Monitoramento de ataques e comportamentos suspeitos.

f) Auditorias de segurança periódicas.

g) Política de privacidade atualizada.

## 7.5 CHECKLIST ESPECÍFICO POR PLATAFORMA

**Android:**

a) ‘android:allowBackup="false"‘ ou BackupAgent seguro.

b) ‘android:debuggable="false"‘ para produção.

c) Proteção de componentes exportados.

d) ‘android:usesCleartextTraffic="false"‘ e Network Security Config.

e) ‘EncryptedSharedPreferences‘ para dados sensíveis.

f) ‘FLAG$_{S}ECURE‘para te lassens veis.$

**iOS:**

a) App Transport Security (ATS) sem exceções.

b) Classes de proteção de dados apropriadas.

c) Keychain com proteção adequada.

d) Desabilitar captura de tela para views sensíveis.

e) Validação de URL Schemes.

f) Verificação de segurança de backups do iCloud.

## 8 CONCLUSÃO

Este trabalho analisou os desafios e soluções para a segurança em aplicativos móveis, cobrindo vulnerabilidades, boas práticas, privacidade, casos reais, ferramentas e um checklist. A complexidade do cenário mobile exige atenção a comunicação insegura, autenticação fraca, criptografia insuficiente e vazamento de dados.

A adoção de boas práticas, como criptografia robusta e autenticação multifator, desde o início do desenvolvimento ("segurança por design"), é crucial. A privacidade do usuário e a conformidade com regulamentações como GDPR e LGPD são essenciais, reforçando a necessidade de "Privacy by Design".

Casos reais destacam padrões comuns de falhas e a importância de modelagem de ameaças, gestão de dependências e resposta a incidentes. Ferramentas de análise como SAST, DAST e SCA, integradas ao ciclo de desenvolvimento, são vitais para identificar e mitigar riscos. O checklist serve como guia prático.

Concluímos que a segurança mobile é um processo contínuo, exigindo abordagem holística: desenvolvimento seguro, testes, monitoramento e resposta rápida. Sugerimos futuras pesquisas em segurança de frameworks cross-platform, IA na segurança mobile, e metodologias de teste para DevOps/CI/CD.

# REFERÊNCIAS

APPBRAIN. **Android version distribution on active devices**. [S.l.], 2023. Disponível em: https://www.appbrain.com/stats/top-android-sdk-versions.

BRASIL, P. da República do. **Lei Geral de Proteção de Dados Pessoais (LGPD)**. 2018. Disponível em: http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm.

CIMPANU, C. **Telegram bug 'leaks' user IP addresses**. 2018. ZDNet. Disponível em: https://www.zdnet.com/article/telegram-bug-leaks-user-ip-addresses/.

DEPARTMENT, S. R. Number of mobile app downloads worldwide from 2016 to 2022. **Statista**, 2023. Disponível em: https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads/.

FOUNDATION, O. **OWASP Mobile Top 10 2023**. [S.l.], 2023. Disponível em: https://owasp.org/www-project-mobile-top-10/.

HSU, J. **The Strava Heat Map and the End of Secrets**. 2018. Wired. Disponível em: https://www.wired.com/story/strava-heat-map-military-bases-fitness-trackers-privacy/.

ISAAC, M.; BENNER, K.; FRENKEL, S. **Uber Hid 2016 Breach, Paying Hackers to Delete Stolen Data**. 2017. The New York Times. Disponível em: https://www.nytimes.com/2017/11/21/technology/uber-hack.html.

NEWMAN, L. H. **A WhatsApp Vulnerability Allowed Hackers to Infect Phones with Spyware**. 2019. Wired. Disponível em: https://www.wired.com/story/whatsapp-vulnerability-targeted-spyware-pegasus/.

OSBORNE, C. **Critical vulnerabilities uncovered in banking apps**. 2020. ZDNet. Disponível em: https://www.zdnet.com/article/critical-vulnerabilities-uncovered-in-banking-apps/.

STANDARDS, N. I. of; TECHNOLOGY. **Recommendation for Key Management: Part 1 - General**. [S.l.], 2020.

SYMANTEC. **Internet Security Threat Report: Mobile Threat Landscape**. [S.l.], 2022.

UNION, E. **General Data Protection Regulation (GDPR)**. 2018. Disponível em: https://gdpr.eu/.

VERACODE. **State of Software Security: Mobile and IoT**. [S.l.], 2022.