UNIVERSIDADE TUIUTI DO PARANÁ

AMANDA BAGGIO AZEVEDO
MATHEUS HENRIQUE BOSLOOPER
NICKSON JEAN FERREIRA WALACHY

APRIMORAMENTO A SEGURANÇA EM APLICAÇÕES MOBILE: UMA ANÁLISE ABRANGENTE DE RISCOS, VULNERABILIDADE E BOAS PRÁTICAS

CURITIBA
2025

AMANDA BAGGIO AZEVEDO
MATHEUS HENRIQUE BOSLOOPER
NICKSON JEAN FERREIRA WALACHY

APRIMORAMENTO A SEGURANÇA EM APLICAÇÕES MOBILE: UMA ANÁLISE ABRANGENTE DE RISCOS, VULNERABILIDADES E BOAS PRÁTICAS

Trabalho apresentado ao Curso De Análise e Desenvolvimento de Sistemas, da Universidade Tuiuti Do Paraná, como requisito avaliativo do 2°Bimestre da disciplina de Desenvolvimento para Dispositivo Móveis.

Professor: Chauã Coluene Queirolo Barbosa da Silva

CURITIBA
2025

RESUMO

O avanço dos dispositivos móveis e a crescente adoção de aplicativos para atividades cotidianas trouxeram não apenas conveniência, mas também um cenário ampliado de riscos e ameaças à segurança digital. Este trabalho analisa de forma abrangente os principais vetores de ataque que afetam plataformas Android e iOS, destacando vulnerabilidades como armazenamento inseguro de dados, uso indevido de permissões, falhas na transmissão de informações e dependência de bibliotecas vulneráveis. Para mitigar tais riscos, são apresentadas boas práticas no ciclo de desenvolvimento seguro, como o uso de criptografia robusta, autenticação multifatorial, validação rigorosa de entradas e gestão eficiente de dependências. Além disso, ressalta-se a importância do gerenciamento consciente de permissões e da conformidade com legislações como a LGPD. Casos reais, como os incidentes envolvendo WhatsApp, TikTok e Facebook, ilustram as consequências da negligência em segurança. O trabalho também discute ferramentas modernas de análise e testes, como MobSF, Frida e OWASP MSTG, além de propor um checklist prático para desenvolvedores. Conclui-se que a segurança mobile deve ser uma prioridade desde as fases iniciais do desenvolvimento, sendo essencial para garantir a privacidade, a integridade dos dados e a confiança do usuário.

Palavras-chave: segurança mobile, vulnerabilidades, boas práticas, LGPD, desenvolvimento seguro.

SUMÁRIO

1. INTRODUÇÃO

A ubiquidade dos dispositivos móveis transformou radicalmente a maneira como interagimos com o mundo digital, impulsionando a proliferação de aplicativos que facilitam desde transações financeiras críticas até a comunicação interpessoal. Contudo, essa expansão exponencial também trouxe consigo um cenário complexo de ameaças e vulnerabilidades que colocam em risco a privacidade e a segurança dos usuários. Este trabalho visa fornecer uma análise aprofundada dos principais desafios inerentes à segurança de aplicações mobile, detalhando vulnerabilidades comuns, apresentando um conjunto de boas práticas e soluções técnicas para mitigação de riscos, e ilustrando a relevância do tema por meio de ferramentas de análise e estudos de caso de falhas reais.

2. VETORES E VULNERABILIDADES NO ANDROID E IOS

Ambas as plataformas, Android e iOS, representam alvos frequentes para ataques cibernéticos, dada a sua vasta base de usuários. As vulnerabilidades mais recorrentes incluem:

- **Armazenamento Inseguro de Dados Sensíveis:** A persistência de dados confidenciais em formato de texto simples ou em locais de fácil acesso no dispositivo, como preferências compartilhadas ou bancos de dados sem criptografia robusta, expõe informações a acessos não autorizados.
- **Transmissão Insegura de Dados:** A comunicação de dados via conexões não criptografadas, como HTTP, torna as aplicações suscetíveis a ataques de **interceptação (Man-in-the-Middle - MitM)**, comprometendo a integridade e confidencialidade das informações.
- **Solicitação Excessiva de Permissões:** A requisição de permissões que transcendem as necessidades funcionais do aplicativo (e.g., acesso indiscriminado à câmera, microfone, contatos, ou localização) levanta sérias questões de privacidade e pode ser explorada por atores mal-intencionados.
- **Falta de Validação de Entradas do Usuário:** A ausência de validação e sanitização rigorosas das entradas fornecidas pelos usuários pode levar a ataques de **injeção (como SQL Injection ou Command Injection)**, permitindo a execução de código malicioso ou o acesso indevido a dados.

- **Uso de Componentes de Terceiros Desatualizados ou Vulneráveis:** A dependência de bibliotecas e SDKs de terceiros que contêm falhas de segurança conhecidas, e que não são atualizados regularmente, introduz pontos de entrada para ataques.
- **Ausência de Mecanismos de Detecção de Alterações no Sistema Operacional:** A falha em implementar mecanismos para detectar dispositivos com jailbreak (iOS) ou root (Android) compromete a segurança nativa do sistema operacional, expondo o aplicativo a um ambiente menos controlado e mais propenso a ataques.

3. BOAS PRÁTICAS DE DESENVOLVIMENTO SEGURO

A mitigação eficaz dos riscos exige a adoção de um conjunto de boas práticas que devem permear todo o ciclo de vida de desenvolvimento do software (SDLC). Dentre as mais cruciais, destacam-se:

- **Criptografia Robusta para Dados em Repouso e em Trânsito:** Empregar criptografia forte, como AES-256 para dados armazenados localmente, e TLS 1.2 ou superior para a transmissão de dados, assegurando a confidencialidade e a integridade das informações.
- **Implementação de Autenticação e Autorização Seguras:** Adotar autenticação multifator (MFA), integrar-se com serviços OAuth 2.0 para delegação segura de autorização e suportar mecanismos biométricos (impressão digital, reconhecimento facial) para uma camada adicional de segurança.
- **Armazenamento Seguro de Credenciais e Informações Sensíveis:** Utilizar soluções nativas de armazenamento seguro, como o Android Keystore ou o iOS Keychain, que fornecem um ambiente isolado e criptografado para dados sensíveis.
- **Validação e Sanitização Rigorosa de Entradas:** Implementar validação abrangente de todas as entradas do usuário no lado do cliente e, crucialmente, no lado do servidor, para prevenir ataques como Cross-Site Scripting (XSS) e injeção de comandos.
- **Gestão de Dependências e Atualizações Contínuas:** Manter todas as bibliotecas e SDKs de terceiros atualizados para suas versões mais recentes,

garantindo a correção de vulnerabilidades conhecidas e a aplicação de patches de segurança.

- **Proteção Contra Engenharia Reversa:** Empregar técnicas como **ofuscação de código** (com ferramentas como ProGuard ou R8 no Android) e **tamper detection** para dificultar a engenharia reversa e a análise do código por atacantes.

4. PERMISSÕES E PRIVACIDADE DO USUÁRIO

O gerenciamento adequado das permissões solicitadas por um aplicativo é um pilar fundamental para a proteção da privacidade do usuário. Os desenvolvedores devem aderir aos seguintes princípios:

- **Princípio do Mínimo Privilégio:** Solicitar apenas as permissões estritamente necessárias para o funcionamento essencial do aplicativo, evitando requisições excessivas que possam ser percebidas como intrusivas.
- **Transparência e Justificativa Clara:** Informar de forma clara, concisa e acessível aos usuários o propósito e a justificativa para cada permissão solicitada, promovendo a confiança.
- **Tratamento Gracioso de Revogação:** Desenvolver o aplicativo para que ele possa lidar graciosamente com a revogação de permissões em tempo de execução, sem comprometer a usabilidade ou a funcionalidade crítica.
- **Conformidade Regulatória:** Assegurar a plena conformidade com legislações de proteção de dados, como a Lei Geral de Proteção de Dados (LGPD) no Brasil, garantindo o consentimento explícito do usuário para a coleta e o uso de dados pessoais, além de direitos como acesso, retificação e exclusão.

5. CASOS REAIS DE FALHAS DE SEGURANÇA

A história recente é rica em exemplos de como falhas de segurança em aplicativos amplamente utilizados podem resultar em impactos significativos:

- **WhatsApp (2019) –** Exploração do Spyware Pegasus: Em maio de 2019, o WhatsApp revelou uma vulnerabilidade crítica em seu aplicativo que permitia a instalação do spyware Pegasus em dispositivos Android e iOS simplesmente fazendo uma chamada telefônica para o número do alvo, mesmo que a pessoa não atendesse a chamada.
    - Essa falha de segurança foi corrigida rapidamente pelo WhatsApp através de uma atualização. No entanto, o incidente gerou grande repercussão e levou o WhatsApp (agora Meta) a processar o NSO Group, a empresa israelense desenvolvedora do Pegasus.
    - Documentos judiciais liberados mais tarde, como parte do processo, confirmaram que o NSO Group admitiu ter desenvolvido exploits para o WhatsApp para instalar o Pegasus em cerca de **1.400 usuários** em 2019, incluindo jornalistas, ativistas de direitos humanos e outras figuras da sociedade civil.

- **TikTok (2020)** – Vulnerabilidades de Sequestro de Sessão: Em janeiro de 2020, pesquisadores da Check Point Research descobriram e divulgaram falhas críticas no aplicativo do TikTok (versão 12.2.0 para Android e iOS). Essas vulnerabilidades, quando combinadas, permitiriam que atacantes:
    - Acessassem e manipulassem contas de usuários, incluindo apagar/publicar vídeos e acessar informações pessoais.
    - Enviassem SMSs falsificados que pareciam vir do TikTok.
    - Realizassem ataques de Cross-Site Scripting (XSS) e Cross-Site Request Forgery (CSRF), que poderiam levar ao roubo de cookies de sessão e, consequentemente, ao sequestro de contas.
    - A exploração frequentemente envolvia a combinação de várias dessas falhas, como o envio de um SMS malicioso que redirecionava o usuário para um site que executava código JavaScript para roubar seus cookies de sessão.

- O TikTok agiu rapidamente para corrigir essas vulnerabilidades após a divulgação e, em 2020, lançou um programa de recompensas por bugs para incentivar a descoberta e o relato responsável de falhas de segurança.

- **Facebook (2019**) – Vazamento de Dados por Configuração Inadequada: Em abril de 2019, a UpGuard revelou que centenas de milhões de registros de usuários do Facebook estavam expostos publicamente em servidores da Amazon (Amazon S3) que não eram protegidos por senha. Isso ocorreu porque empresas como a Cultura Colectiva (com 540 milhões de registros) e o aplicativo "At the Pool" (com senhas de 22.000 usuários em texto simples) falharam em proteger adequadamente os dados que coletavam do Facebook em suas próprias infraestruturas.
  - O impacto foi a exposição de dados pessoais (como IDs de usuário, comentários, reações e, em alguns casos, senhas), tornando os usuários vulneráveis a phishing e roubo de identidade. O incidente levantou sérias preocupações sobre a responsabilidade do Facebook na supervisão de como terceiros gerenciam os dados de seus usuários.
  - Em resposta, o Facebook trabalhou para remover os bancos de dados expostos e reforçou suas políticas sobre o armazenamento de dados por desenvolvedores terceirizados.

6. FERRAMENTAS E TESTES DE SEGURANÇA

O uso de ferramentas especializadas é imprescindível para identificar e remediar vulnerabilidades. As principais incluem:

- **MobSF (Mobile Security Framework):** Uma plataforma automatizada que realiza análises estáticas e dinâmicas de aplicativos Android e iOS, identificando vulnerabilidades de forma eficiente.
- **QARK (Quick Android Review Kit):** Uma ferramenta de linha de comando para Android que escaneia APKs em busca de falhas de segurança comuns.
- **Frida:** Um *toolkit* de instrumentação dinâmica que permite a injeção de scripts em tempo de execução, ideal para análise de comportamento de aplicativos e testes de segurança avançados.

- **Drozer:** Uma ferramenta robusta para avaliação da superfície de ataque de aplicações Android, focada na exploração de componentes expostos e vulnerabilidades de interprocess communication (IPC).
- **Burp Suite:** Uma suite de ferramentas amplamente utilizada para testes de penetração em aplicações web, igualmente eficaz na interceptação e modificação de requisições HTTP/S em aplicativos móveis.
- **OWASP Mobile Security Testing Guide (MSTG):** Um guia abrangente que oferece uma metodologia de testes de segurança para aplicativos móveis, incluindo checklists e casos de uso detalhados.

7. CHECKLIST DE SEGURANÇA PARA APPS MOBILE

A integração de um checklist de segurança ao longo do SDLC é fundamental. Os desenvolvedores devem garantir:

- **Comunicações Criptografadas:** Todas as comunicações de rede devem utilizar HTTPS/TLS com certificados válidos e fixação de certificados (certificate pinning) quando apropriado.
- **Validação de Entrada Robusta:** Implementar validação rigorosa de todas as entradas do usuário, tanto no cliente quanto no servidor, para prevenir ataques de injeção.
- **Gerenciamento de Dependências:** Monitorar e atualizar bibliotecas e SDKs de terceiros para corrigir vulnerabilidades conhecidas.
- **Criptografia de Dados em Repouso:** Criptografar todos os dados sensíveis armazenados localmente no dispositivo utilizando os mecanismos de armazenamento seguro do sistema operacional.
- **Autenticação e Autorização Fortes:** Implementar mecanismos de autenticação e autorização robustos, incluindo MFA e biometria, e gerenciar sessões de forma segura.
- **Logging Seguro:** Desenvolver logs de forma segura, evitando a inclusão de informações sensíveis e implementando políticas de retenção adequadas.
- **Testes de Segurança Contínuos:** Realizar testes regulares de vulnerabilidade e penetração utilizando ferramentas automatizadas e manuais.

- **Proteção Contra Engenharia Reversa:** Aplicar técnicas de ofuscação de código e detecção de *tampering*.

10. CONCLUSÃO

A segurança em aplicações móveis não é um mero acessório, mas um pilar fundamental que deve ser incorporado desde as fases iniciais de concepção e design. As vulnerabilidades detalhadas neste trabalho evidenciam que, mesmo equívocos aparentemente simples, podem gerar prejuízos consideráveis em termos de privacidade, dados e reputação. Ao adotar um conjunto abrangente de boas práticas, empregar ferramentas de análise de segurança especializadas e, sobretudo, priorizar o respeito à privacidade do usuário e a conformidade com as regulamentações vigentes (como a LGPD), os desenvolvedores podem construir aplicativos que sejam não apenas funcionais, mas intrinsecamente seguros, confiáveis e responsáveis. A segurança mobile é um investimento contínuo, não uma despesa, e sua negligência pode ter consequências devastadoras em um cenário digital cada vez mais interconectado.

REFERÊNCIAS

BBC NEWS. WhatsApp targeted in NSO spyware attack. [S. l.], 2024. Disponível em: https://www.bbc.com/news/articles/c77n76kzmz4o. Acesso em: 20 jun. 2025.

TI INSIDE. Ciberataque: EUA se preparam para impactos das eleições. [S. l.], 2025. Disponível em: https://tiinside.com.br/04/06/2025/498268/. Acesso em: 20 jun. 2025.

TECMUNDO. Pegasus: documento mostra número exato e localização de vítimas do spyware no WhatsApp. [S. l.], 28 abr. 2022. Disponível em: https://www.tecmundo.com.br/seguranca/403882-pegasus-documento-mostra-numero-exato-e-localizacao-de-vitimas-do-spyware-no-whatsapp.htm. Acesso em: 25 jun. 2025.

ISTOÉ DINHEIRO. Ferramenta usada até 2019 permitiu o vazamento de dados do Facebook. [S. l.], 16 jan. 2020. Disponível em: https://istoedinheiro.com.br/ferramenta-usada-ate-2019-permitiu-o-vazamento-de-dados-do-facebook. Acesso em: 25 jun. 2025.

OLHAR DIGITAL. Vazamento expõe dados de 267 milhões de usuários do Facebook. [S. l.], 20 dez. 2019. Disponível em: https://olhardigital.com.br/2019/12/20/noticias/vazamento-expoe-dados-de-267-milhoes-de-usuarios-do-facebook/. Acesso em: 28 jun. 2025.