UNIVERSIDADE TUIUTI DO PARANÁ

CAROLYNE FERNANDA DOS MARTYRES

MATEUS HENRIQUE LIMA

MATHEUS HENRIQUE MIRANDA LÓS

SEGURANÇA EM APLICATIVOS MOBILE: RISCOS, VULNERABILIDADES E BOAS PRÁTICAS

CURITIBA

2025

CAROLYNE FERNANDA DOS MARTYRES

MATEUS HENRIQUE LIMA

MATHEUS HENRIQUE MIRANDA LÓS

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 avaliativo do 2º bimestre da disciplina desenvolvimento para dispositivos móveis.

Professor: Chauã Coluene Queirolo Barbosa da Silva

CURITIBA

2025

RESUMO

Este trabalho acadêmico investiga a segurança em aplicativos móveis, abordando os principais riscos, vulnerabilidades e boas práticas no cenário digital contemporâneo. O objetivo foi analisar os desafios de segurança em apps Android e iOS e apresentar soluções técnicas e preventivas. A metodologia envolveu a análise das vulnerabilidades mais comuns, como o armazenamento e comunicação insegura de dados e falhas de autenticação, além de casos reais de comprometimento. Os resultados apontam para a necessidade de implementação rigorosa de criptografia, mecanismos robustos de autenticação e armazenamento seguro de dados, bem como a gestão consciente de permissões de aplicativos. Conclui-se que a adoção de uma cultura de segurança por design, o uso de ferramentas de análise (SAST e DAST) e a aplicação de um checklist de segurança são essenciais para proteger a privacidade do usuário e a integridade dos dados, mitigando os riscos associados ao crescente uso de plataformas móveis.

Palavras-chave: Segurança móvel; Vulnerabilidades; Criptografia; Privacidade; Boas práticas**.**

SUMÁRIO

1 INTRODUÇÃO

O avanço tecnológico e a crescente ubiquidade dos dispositivos móveis transformaram os aplicativos móveis em ferramentas indispensáveis para uma vasta gama de atividades, desde comunicação e entretenimento até transações financeiras e gerenciamento de dados sensíveis. Essa onipresença, no entanto, eleva a segurança a um patamar crítico no ciclo de desenvolvimento de software. A ocorrência de vulnerabilidades, configurações inadequadas de permissões e práticas de programação inseguras pode resultar em sérios comprometimentos da privacidade e da integridade dos dados dos usuários. Diante desse cenário, este estudo propõe uma investigação aprofundada sobre os principais riscos de segurança inerentes ao desenvolvimento de aplicações móveis e busca apresentar boas práticas eficazes para a sua mitigação.

O objetivo central deste trabalho é analisar os desafios mais relevantes da segurança no desenvolvimento de aplicativos móveis e, a partir disso, apresentar soluções técnicas e preventivas. Para alcançar esse propósito, a pesquisa abordará de forma detalhada as principais vulnerabilidades em aplicativos Android e iOS, destacando as falhas mais comuns que podem ser exploradas por atacantes. Será explorada a fundo a implementação de boas práticas de segurança, focando em pilares como a criptografia de dados em repouso e em trânsito, a importância de mecanismos robustos de autenticação e o armazenamento seguro de informações sensíveis. Adicionalmente, será discutida a crucial relação entre as permissões de aplicativos e a privacidade do usuário, enfatizando como uma gestão inadequada pode expor dados pessoais. Para ilustrar a urgência do tema, serão apresentados casos reais de falhas de segurança em aplicativos populares, demonstrando as consequências práticas das vulnerabilidades. A pesquisa também se dedicará a explorar as ferramentas e técnicas de análise de segurança mobile disponíveis para desenvolvedores e especialistas, como a análise estática (SAST) e dinâmica (DAST) de código, e os testes de penetração. Por fim, será oferecido um checklist de segurança abrangente para desenvolvedores, servindo como um roteiro prático para a construção de aplicativos mais seguros.

Este estudo justifica-se pela necessidade premente de garantir a segurança dos dados e a privacidade dos usuários em um ecossistema móvel cada vez mais complexo e interconectado. Ao fornecer uma análise detalhada dos riscos e soluções,

o trabalho visa contribuir para o desenvolvimento de aplicativos mais robustos e confiáveis, fortalecendo a confiança dos usuários nas tecnologias móveis

2 PRINCIPAIS VULNERABILIDADES EM APLICATIVOS ANDROID E IOS

Apesar de serem plataformas maduras e em constante evolução, Android e iOS não estão imunes a falhas de segurança em seus aplicativos. Muitas dessas vulnerabilidades surgem de erros no código, configurações inadequadas ou uso incorreto dos recursos do sistema.

Uma das fraquezas mais comuns em ambos os sistemas é o armazenamento inseguro de dados. Isso ocorre quando informações sensíveis, como credenciais de login ou dados financeiros, são salvas em locais desprotegidos no dispositivo, facilmente acessíveis por atacantes que obtêm acesso físico ou lógico. Além disso, a comunicação insegura é um risco frequente, onde dados sensíveis são transmitidos por redes não criptografadas ou com protocolos de segurança fracos, permitindo a interceptação por terceiros mal-intencionados.

A autenticação e autorização quebradas representam outra categoria crítica de vulnerabilidades. Senhas fracas, a falta de autenticação multifator (MFA) e permissões de usuário mal definidas podem abrir portas para que invasores acessem contas ou funcionalidades privilegiadas. Similar às aplicações web, a injeção também pode comprometer apps móveis, através da manipulação de entradas de dados que levam à execução de comandos maliciosos, como em ataques de SQL Injection. O manuseio inadequado de sessão, com tokens de sessão previsíveis ou que não expiram, também facilita o roubo e a reutilização por atacantes.

No contexto específico do Android, sua natureza mais aberta e fragmentada pode levar a vulnerabilidades como componentes expostos (atividades, serviços, broadcast receivers e provedores de conteúdo que podem ser acessados indevidamente por outros aplicativos) e intents inseguras, que, se mal utilizadas, podem expor dados ou permitir a execução de ações não autorizadas.

Já no iOS, conhecido por ser um ambiente mais controlado, as falhas geralmente decorrem de criptografia inadequada, onde, mesmo com as robustas APIs da Apple, desenvolvedores podem cometer erros na implementação. Problemas com o Keychain, o serviço de armazenamento seguro de credenciais, também podem surgir se não configurado corretamente, permitindo o acesso a dados sensíveis em dispositivos jailbroken ou por malware. Falhas no sandboxing, o mecanismo de

isolamento de aplicativos, podem, em casos raros, permitir que um app malicioso escape de seu ambiente restrito.

3 PILARES DA SEGURANÇA: CRIPTOGRAFIA, AUTENTICAÇÃO E ARMAZENAMENTO

Para combater essas vulnerabilidades, a incorporação de boas práticas de segurança desde as fases iniciais do desenvolvimento é fundamental.

A criptografia é a espinha dorsal da proteção de dados. No transporte de dados, é imperativo o uso de HTTPS/TLS para toda a comunicação com servidores, protegendo contra ataques de Man-in-the-Middle (MitM). Para dados em repouso, é essencial criptografar informações sensíveis armazenadas localmente no dispositivo. No Android, isso pode ser feito com o EncryptedSharedPreferences ou bancos de dados criptografados como o SQLCipher. No iOS, o Keychain é a ferramenta ideal para credenciais e tokens, e a proteção de dados nativa do sistema garante a criptografia de arquivos quando o dispositivo está bloqueado. Crucialmente, as chaves de criptografia nunca devem ser embutidas no código; elas devem ser geradas e gerenciadas com segurança usando sistemas como o Android Keystore System ou o iOS Keychain.

Mecanismos de autenticação robustos são vitais para verificar a identidade dos usuários. Isso envolve a imposição de senhas fortes, a implementação de autenticação multifator (MFA) para uma camada extra de segurança, e a limitação de tentativas de login para evitar ataques de força bruta. Ao utilizar biometria, é fundamental empregar as APIs nativas do sistema (como BiometricPrompt no Android e Local Authentication Framework no iOS) e garantir que os dados biométricos não sejam armazenados diretamente no aplicativo. Tokens de autenticação, como JWT, devem ter tempo de expiração limitado e serem renovados de forma segura.

O armazenamento seguro de dados no dispositivo exige cautela. O princípio é claro: armazenar apenas o que é estritamente necessário. Para isso, devem-se utilizar APIs seguras fornecidas pelo sistema operacional, como o EncryptedSharedPreferences e bancos de dados criptografados no Android, e o Keychain ou a proteção de arquivos no iOS. Nunca se deve armazenar credenciais de usuário em texto simples; elas devem ser hashadas e "salgadas". Além disso, é importante garantir que dados temporários ou em cache sejam limpos regularmente.

4 PERMISSÕES DE APPS E A PROTEÇÃO DA PRIVACIDADE DO USUÁRIO

As permissões de aplicativos representam um controle vital sobre o acesso de um app a recursos sensíveis do dispositivo, impactando diretamente a privacidade do usuário. Tanto Android quanto iOS adotaram modelos de permissões de runtime, onde o usuário concede ou nega o acesso no momento em que o aplicativo realmente precisa de um recurso, e não na instalação, oferecendo maior controle.

A melhor prática é seguir o Princípio do Mínimo Privilégio: solicitar apenas as permissões absolutamente necessárias para o funcionamento essencial do aplicativo. É crucial explicar ao usuário o propósito de cada permissão antes de solicitá-la, aumentando a transparência e a confiança. Por exemplo, um aplicativo de fotos que pede acesso à câmera faz sentido, mas um aplicativo de lanterna que pede acesso à localização ou aos contatos levanta sérias bandeiras vermelhas de privacidade. O aplicativo também deve ser capaz de lidar graciosamente com a recusa de uma permissão, desabilitando a funcionalidade dependente sem prejudicar a experiência geral. Por fim, revisões periódicas das permissões declaradas são importantes para evitar excessos ou requisitos obsoletos.

O uso inadequado de permissões pode levar à coleta excessiva de dados, ao vazamento de informações sensíveis caso o aplicativo seja comprometido, e ao rastreamento ou criação de perfis detalhados dos usuários sem seu consentimento informado, sublinhando a responsabilidade dos desenvolvedores.

5 CASOS REAIS DE FALHAS DE SEGURANÇA

A história recente é pontuada por incidentes de segurança em aplicativos amplamente utilizados, que servem como lembretes contundentes da importância das boas práticas.

Um dos casos mais notórios foi o vazamento de dados do Facebook em 2018 e 2019, que expôs informações de milhões de usuários devido a falhas na manipulação de tokens de acesso e APIs. Embora não fosse uma falha exclusiva de aplicativo móvel, o incidente teve um impacto direto na segurança dos usuários de seus apps.

Em 2020, o Zoom Bombing expôs vulnerabilidades no aplicativo Zoom durante seu boom de popularidade. A ausência de senhas padrão para reuniões, identificadores previsíveis e, inicialmente, a falta de criptografia de ponta a ponta, permitiram que invasores entrassem em reuniões privadas, evidenciando a necessidade de segurança por design.

O WhatsApp, conhecido por sua criptografia robusta, também enfrentou um ataque em 2019 que permitiu a instalação de spyware via chamadas de voz, devido a uma falha de buffer overflow. Este incidente demonstrou que mesmo aplicativos com forte criptografia podem ter vulnerabilidades em outras áreas, como o manuseio de entrada de dados.

Finalmente, o vazamento de dados do Strava em 2018, que revelou a localização de bases militares secretas através de um mapa de calor global, destacou a importância da privacidade por design e de configurações de privacidade padrão seguras, que nem sempre são as mais abertas.

Esses exemplos práticos reforçam que as falhas de segurança podem ter diversas origens, desde erros de codificação e configurações inadequadas até a falta de atenção à privacidade. A principal lição é a necessidade de uma abordagem de segurança abrangente e contínua.

6 FERRAMENTAS E TÉCNICAS DE ANÁLISE DE SEGURANÇA MOBILE

Para identificar e mitigar vulnerabilidades, desenvolvedores e equipes de segurança contam com um arsenal de ferramentas e técnicas.

A Análise Estática de Código (SAST) examina o código-fonte ou binários do aplicativo sem executá-lo, buscando padrões conhecidos de vulnerabilidades, como injeção de SQL ou credenciais hardcoded. Ferramentas como MobSF e Checkmarx são amplamente utilizadas nessa fase, permitindo a identificação precoce de falhas e sua integração em processos de Integração Contínua/Entrega Contínua (CI/CD).

Em contraste, a Análise Dinâmica de Código (DAST) testa o aplicativo em tempo de execução, simulando interações maliciosas para identificar vulnerabilidades que só se manifestam com o app em funcionamento, como comunicação insegura ou problemas de autenticação em tempo real. MobSF, Frida e Burp Suite são exemplos de ferramentas que auxiliam na interceptação e análise do tráfego de rede e no comportamento do aplicativo em um ambiente controlado.

Os Testes de Penetração (Pentests) são avaliações de segurança simuladas conduzidas por profissionais que atuam como atacantes éticos. Eles utilizam uma combinação de ferramentas automatizadas e técnicas manuais para explorar vulnerabilidades e avaliar a postura de segurança geral do aplicativo, seguindo metodologias como a OWASP Mobile Application Security Verification Standard (MASVS). Embora mais caros e demorados, os pentests oferecem uma visão mais realista das vulnerabilidades.

Adicionalmente, a Engenharia Reversa é uma técnica que analisa o binário de um aplicativo para entender seu funcionamento interno. Atacantes a utilizam para descobrir vulnerabilidades ou roubar propriedade intelectual. Para se proteger, desenvolvedores empregam técnicas como a ofuscação de código (com ferramentas como ProGuard ou DexGuard para Android), que dificulta a leitura do código decompilado, e mecanismos anti-tampering e anti-debugging para detectar e impedir modificações ou análises maliciosas do aplicativo. A detecção de root ou jailbreak também é uma medida de segurança importante.

A combinação dessas abordagens — SAST, DAST, testes de penetração e proteção contra engenharia reversa — oferece uma estratégia robusta para garantir a segurança dos aplicativos móveis.

7 CHECKLIST DE SEGURANÇA PARA DESENVOLVEDORES

Para auxiliar desenvolvedores a incorporar a segurança em todas as fases do desenvolvimento, um checklist prático é essencial.

Durante a fase de projeto e requisitos, a modelagem de ameaças é crucial para identificar riscos potenciais, garantindo que a segurança seja parte do design desde o início. É importante definir claramente os requisitos de privacidade e dar preferência ao uso de APIs seguras do sistema operacional.

Na fase de desenvolvimento e codificação, a validação de todas as entradas de dados (tanto no cliente quanto no servidor) é fundamental para prevenir ataques de injeção. O armazenamento seguro de dados implica em não salvar informações sensíveis em locais desprotegidos e usar o Keychain (iOS) ou Android Keystore System para credenciais e chaves. Todas as comunicações devem ser feitas via HTTPS/TLS, com a possibilidade de fixação de certificados (Certificate Pinning) para maior proteção contra MitM. A autenticação forte (senhas complexas, MFA) e o gerenciamento seguro de sessões são indispensáveis. As permissões de aplicativos devem ser solicitadas com o mínimo privilégio e com justificativa clara para o usuário, e o aplicativo deve lidar com a recusa de forma elegante. É vital evitar a exposição de informações sensíveis em logs ou mensagens de erro e implementar proteção contra engenharia reversa com ofuscação de código e anti-tampering.

A fase de testes envolve a integração de ferramentas SAST e DAST em processos de CI/CD, além da realização de testes de penetração por equipes especializadas, validando se os requisitos de segurança foram atendidos.

Finalmente, na fase de implantação e monitoramento, é essencial monitorar continuamente a segurança para detectar atividades suspeitas e ter um plano de resposta a incidentes bem definido. Manter o aplicativo e todas as bibliotecas de terceiros atualizadas e realizar revisões de segurança periódicas garantem a resiliência contínua.

Seguindo este checklist, desenvolvedores podem construir aplicativos móveis mais robustos e proteger efetivamente os dados e a privacidade de seus usuários.

8 CONCLUSÃO

Ao longo deste trabalho, propusemo-nos a analisar os principais desafios de segurança no desenvolvimento de aplicativos móveis e a apresentar soluções técnicas e preventivas, abarcando desde as vulnerabilidades mais comuns até as boas práticas e ferramentas de análise. Os objetivos inicialmente traçados foram plenamente alcançados, proporcionando uma compreensão abrangente dos riscos inerentes ao ecossistema mobile e das estratégias para mitigá-los. A investigação revelou que as vulnerabilidades em aplicativos Android e iOS, como o armazenamento inseguro de dados, a comunicação não criptografada e as falhas de autenticação, persistem como ameaças significativas. Contudo, a aplicação de boas práticas de segurança, incluindo o uso rigoroso de criptografia tanto para dados em trânsito quanto em repouso, a implementação de mecanismos robustos de autenticação (com foco na autenticação multifator) e o armazenamento seguro de dados, demonstrou ser fundamental para blindar as aplicações contra ataques.

A discussão sobre as permissões de aplicativos e a privacidade do usuário reforçou a importância do princípio do mínimo privilégio e da transparência, aspectos cruciais para construir a confiança do usuário. Os casos reais de falhas de segurança em apps populares serviram como um alerta prático, evidenciando que mesmo grandes empresas não estão imunes a incidentes e que a segurança deve ser uma preocupação contínua. A análise das ferramentas e técnicas de análise de segurança mobile, como SAST, DAST e testes de penetração, reforçou a necessidade de uma abordagem multifacetada e contínua no ciclo de vida do desenvolvimento. O checklist de segurança para desenvolvedores consolidado neste estudo oferece um guia prático e aplicável para integrar a segurança desde as fases iniciais do projeto.

Em síntese, o trabalho corrobora a premissa de que a segurança em aplicativos móveis não é um diferencial, mas uma necessidade intrínseca para qualquer desenvolvedor ou empresa. A negligência nesse campo pode resultar em severas perdas financeiras, danos reputacionais e, mais importante, no comprometimento da privacidade do usuário. A aplicação consistente das diretrizes e boas práticas apresentadas neste estudo pode significativamente elevar a postura de segurança das aplicações, beneficiando diretamente os usuários ao proteger seus dados sensíveis e garantindo a integridade de suas interações digitais. Para trabalhos futuros, sugere-se a exploração aprofundada de novas ameaças emergentes, como aquelas ligadas

à Inteligência Artificial e ao Machine Learning no contexto da segurança mobile, bem como a avaliação comparativa da eficácia de diferentes ferramentas de análise de segurança em cenários de desenvolvimento ágil.

REFERÊNCIAS

APPLE. *Apple Developer - Security*. Disponível em: https://developer.apple.com/security/. Acesso em: 08 jun. 2025.

BURP SUITE. *PortSwigger*. Disponível em: https://portswigger.net/burp. Acesso em: 08 jun. 2025.

FRIDA. *Frida.re*. Disponível em: https://frida.re/. Acesso em: 08 jun. 2025.

GOOGLE. *Android Developers - Security best practices*. Disponível em: https://developer.android.com/training/articles/security-best-practices. Acesso em: 08 jun. 2025.

MICROSOFT. *Threat modeling*. Disponível em: https://learn.microsoft.com/en-us/previous-versions/commerce/security-for-ecommerce-applications/dn205244(v=msdn.10). Acesso em: 08 jun. 2025.

MOBSF. *Mobile Security Framework*. Disponível em: https://MobSF.github.io/Mobile-Security-Framework-MobSF/. Acesso em: 08 jun. 2025.

OWASP Mobile Security Project. *OWASP Mobile Application Security Verification Standard (MASVS)*. Disponível em: https://owasp.org/www-project-mobile-app-security-verification-standard/. Acesso em: 08 jun. 2025.

OWASP Mobile Security Project. *OWASP Mobile Top 10*. Disponível em: https://owasp.org/www-project-mobile-top-10/. Acesso em: 08 jun. 2025.

SERVICEIT. Confidencialidade, integridade e disponibilidade: os três pilares da segurança da informação. Disponível em: https://service.com.br/confidencialidade-integridade-e-disponibilidade-os-tres-pilares-da-seguranca-da-informacao/. Acesso em: 11 jun. 2025.

SQLCIPHER. *Zetetic*. Disponível em: https://www.zetetic.net/sqlcipher/. Acesso em: 08 jun. 2025.