MobiSec: Apoio à Teste de Aplicações Móveis

Mariano Florencio Mendonça[1], Layse Santos Souza[1], Isadora Lima do Nascimento[1], Fabio Gomes Rocha[1]

[1]Universidade Tiradentes (Unit)
Aracaju - SE - Brasil

{marianofmendonca, santoslay3, isadora.mlima21, gomesrocha}@gmail.com

Abstract. *This article presents the MobiSec tool that aims to conduct software tests on Android mobile applications. Evaluating an application's security level is not an easy task because of the amount of data being transferred in real time and malicious software. Thus, it is necessary to expedite the identification of these vulnerabilities by performing tests. Despite testing being a lengthy process, its automation makes it an interesting alternative and technically feasible for software houses. This automation can increase the processes because of the need to create test plans, but decreases execution time, providing future benefits for management.*

Resumo. *Este artigo apresenta a ferramenta MobiSec que tem como objetivo realizar testes de software nos aplicativos do sistema operacional Android. Tendo em vista que avaliar o nível de segurança de aplicações Android não é uma atividade fácil devido a quantidade de dados sendo transferidos em tempo real e de softwares maliciosos. Dessa maneira, é necessário agilizar o processo de avaliação destas vulnerabilidades realizando a automatização dos testes que apesar de ser um processo demorado, é uma alternativa interessante e tecnicamente viável para as softwares house. Esta automação pode aumentar os processos devido a elaboração de planos de teste, porém diminui o tempo para sua execução, proporcionando benefícios futuros para a gestão.*

1. Introdução

A segurança da informação em dispositivos móveis demonstra a preocupação tanto dos usuários quanto das empresas de TI em virtude do crescente uso de smartphones. Cavalcanti et al. (2017) afirma que atualmente estima-se que existem cerca de 20 milhões de softwares maliciosos sendo 90% na plataforma Android. Esse aumento de ameaças comprova a necessidade da garantia de segurança através da privacidade dos dados processados, e.g., autenticação, controle de acesso, criptografia, e integridade.

Santista (2018) afirma que o Android é o sistema operacional móvel mais utilizado no mundo, logo, é alvo de ataques, e.g., vírus e malwares. Gargenta (2011) admite que avaliar rapidamente a segurança e a robustez das aplicações garante que a aplicação atenda as características relacionadas à segurança.

Malek et al. (2012) complementa Gargenta (2011) quando revela que novos desafios na segurança estão surgindo em razão das ameaças, e que estas podem ser solucionadas por meio de testes de segurança. O teste de segurança tem como objetivo garantir que o funcionamento da aplicação consista exatamente conforme o que foi especificado, dessa forma, verifica o software de acordo com o seu comportamento mediante as diversas tentativas ilegais de acesso procurando possíveis vulnerabilidades após testar todos os mecanismos de proteção.

Em vista dessas ameaças que estão surgindo na plataforma Android elaboramos uma ferramenta online, MobiSec, que está em processo de desenvolvimento, com intenção de automatizar testes de segurança baseados em planos de teste elaborados pelos usuários. O plano de teste é um documento que consiste uma modelagem detalhada do fluxo de trabalho durante o processo. Após a definição do plano de teste, realiza-se o upload da apk, esta é automaticamente testada em relação ao que foi proposto no plano de teste.

2. Referencial Teórico

Para o desenvolvimento desta ferramenta foi necessário entender conceitos como Android, Androguard, AM-TaaS, plataforma mobile, segurança mobile, teste de penetração e teste de regressão.

O Android é um sistema operacional móvel que possui plataforma de código aberto e completa para desenvolvedores de aplicativos proporcionando a execução destes em vários dispositivos. A plataforma mobile oferece conteúdos vinculados às telas dos dispositivos móveis sem distorções através de uma infraestrutura independente e compatível com diferentes aparelhos, e.g., sites. A segurança mobile garante a privacidade dos dados processados através do controle de acesso, autenticação, criptografia, integridade.

O Androguard é uma ferramenta escrita em Python que tem como objetivo criar softwares exibindo, modificando e salvando suas aplicações de forma fácil e estática, sendo bastante útil para a realização da engenharia reversa em aplicativos, e.g., malwares. O AM-TaaS, Automated Mobile Testing as a Service, disponibiliza testes automatizados para aplicações móveis baseados nos critérios de teste publicados pela App Quality Alliance (AQuA). O teste de penetração, PenTest, realiza um teste em uma rede ou sistema de computadores a fim de descobrir todas as vulnerabilidades identificando o tamanho do dano causado por uma invasão. O teste de regressão é uma técnica de teste de software que consiste na aplicação de versões mais recente do software, garantindo o não surgimento de novos defeitos em componentes anteriormente analisados.

Bertoglio e Zorzo (2015) apontam que os critérios elaborados pela AQuA, e.g., características e recursos dos dispositivos, auxiliam na realização dos testes em aplicativos móveis reduzindo o esforço gerado na execução do teste de software visando a garantia dos aplicativos móveis. De acordo com Menezes et al. (2015) conhecer a aplicação que deseja executar o PenTest é importante para facilitar a execução, além de

conhecer o sistema alvo, o profissional deve possuir um alto nível de entendimento em programação, sistemas operacionais, redes e ferramentas de testes. Ainda segundo ele, este profissional passa 90% do tempo estudando a aplicação e 10% do mesmo realizando o PenTest.

3. Metodologia

Este trabalho pode ser classificado quanto aos objetivos como pesquisa descritiva e aplicada sobre o ponto de vista científico. Consequentemente, foram encontrados trabalhos relacionados com propostas similares ao que foi recomendado no MobiSec. Dentre eles, as ferramentas Apktool, MobSF, e Drozer.

A Apktool (2017) é uma ferramenta para realização de engenharia reversa em aplicações Android permitindo acesso ao código fonte de aplicações, tornando possível depurar o código passo a passo. Esta foi projetada para ajudar os desenvolvedores com problemas de localização e auxiliar nas tarefas repetitivas do desenvolvimento. Apesar de ser uma ótima ferramenta, seu objetivo principal não é a segurança, dessa forma, não realiza testes automaticamente e deixa a análise do código e dos testes propriamente ditos para os testadores e desenvolvedores. Ao contrário do que é proposto pelo MobiSec.

O MobSF (2016) é um framework automatizado, “tudo em um”, de testes de penetração para aplicações móveis (iOS/Android/Windows) capaz de realizar testes de performance, análise estática, análise dinâmica e de Web APIs. Logo, é uma ferramenta bastante completa porém, não considera planos de teste durante o processo de testes como o MobiSec.

O Drozer (2015) é uma ferramenta que permite procurar vulnerabilidades de segurança em aplicativos e dispositivos móveis, assumindo o papel de um aplicativo e interagindo com a Máquina Virtual Dalvik, a partir de outros aplicativos do IPC de terminais e do sistema operacional subjacente. Dessa forma, solicita ao usuário que instale a ferramenta na sua máquina e estabeleça uma conexão com um dispositivo Android ou um emulador de Android. Sendo a vantagem da MobiSec que todo o teste pode ser feito via Web e é necessário somente a apk da aplicação.

4. Ferramenta MobiSec

É um sistema para teste de software conduzido para avaliar a segurança de aplicativos móveis guiado por um plano de teste. Souza et al. (2013) testemunha que este é o primeiro documento que deve ser elaborado para avaliação de software contendo as informações convenientes ao software, e.g., métodos e funções a serem testadas. No final da execução de um teste são realizadas as correções dos erros ou defeitos identificados por intermédio do plano de teste.

A principal proposta da ferramenta é guiar o desenvolvedor a elaborar sua aplicação de forma segura contra penetração de códigos maliciosos, como também

garantir que o aplicativo seja desenvolvido conforme o planejado, dessa maneira, proporciona confiabilidade ao usuário. Desenvolvida para o ambiente Web, o MobiSec armazena em um Sistema Gerenciador de Banco de Dados (SGBD) todos os planos de testes e execuções.

Para o desenvolvimento da ferramenta foram escolhidas duas linguagens de programação, Python em virtude do Androguard e PHP em razão do Gerenciador de Banco de Dados MySQL (MariaDB) e da integração da ferramenta com o Androguard. Neste caso, o MariaDB realiza uma conexão com a biblioteca MySQLi, nativa do PHP, visando agilizar o processo de armazenamento das informações coletadas nas execuções dos testes, e o Androguard atua como ferramenta de apoio a execução dos testes de aplicações móveis.

Para elaboração e execução da análise do aplicativo o MobiSec foi desenvolvido em ambiente web com a linguagem PHP, devido a sua alta performance, integrada ao Androguard, onde o usuário cadastra o plano de teste e a aplicação em Python descompila a apk Android. A ação realizada pelo Androguard encaixa-se em várias técnicas de análise de segurança, uma delas é o Pentest.

O MobiSec coleta as informações da apk disponibilizadas pelo Androguard fazendo a comparação com o que foi cadastrado no plano de teste, aprovando-a ou não, seguindo o que foi planejado.

4.1. Arquitetura

Uma visão geral da proposta arquitetural para a ferramenta é representada na Figura 1. Sendo assim, definiu-se a arquitetura em dois blocos, cliente e servidor, sendo a servidor dividida em três blocos, Servidor Apache, Servidor de Banco de Dados e Python, possuindo seu acesso protegido por um Firewall. Em seguida, pode-se observar que o PHP depende do servidor Apache. Além disso, a comunicação do PHP com o SGBD e o Androguard, exibindo para o cliente as interfaces HTML, scripts e CSS proporcionando uma melhor visão da ferramenta ao usuário.

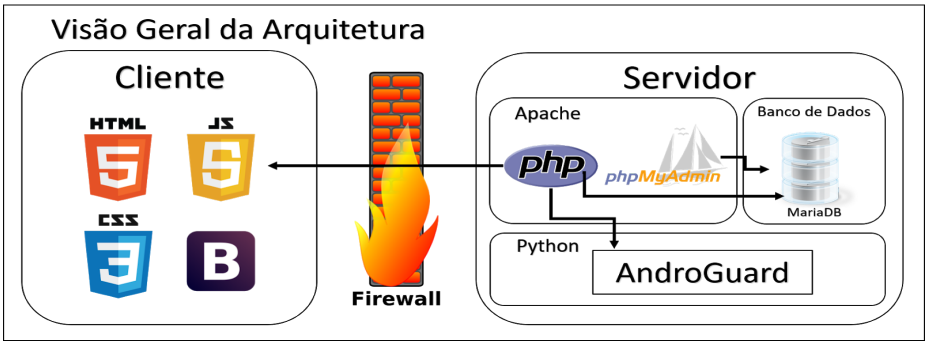

Figura 1 - Visão geral de arquitetura, conceito da ferramenta

4.2. Funcionalidades

O MobiSec possui uma interface de registro e login. Nela o usuário se registra e é redirecionado para o sistema que possui as seguintes funcionalidades: realizar o

cadastro de um plano de teste, testar as aplicações, visualizar relatórios das aplicações e acessar o histórico dos planos de testes cadastrados.

Para desfrutar melhor do MobiSec é necessário realizar o upload da aplicação Android (apk) para o sistema através da integração com o Androguard, após o upload o software confronta o que foi planejado com o que foi desenvolvido, e consequentemente avalia a segurança da aplicação através das permissões por ela solicitadas ao usuário. A aplicação grava em um Sistema Gerenciador de Banco de Dados (SGBD) todas as informações coletadas da aplicação para que o usuário confronte com correções e outras aplicações desenvolvidas. Todo o processo citado acima pode ser observado na Figura 2 representado no diagrama de atividade .

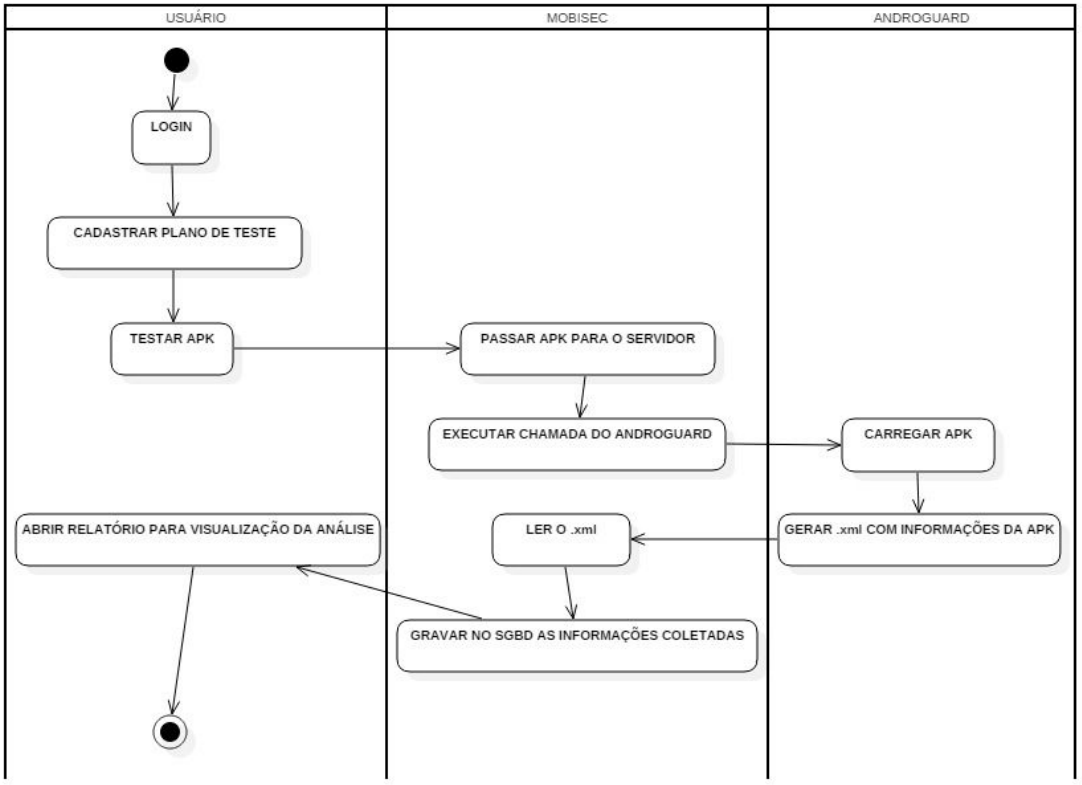

Figura 2 - Diagrama de atividade do MobiSec

Para exemplificar, presuma que pretende-se desenvolver uma aplicação com o apoio do MobiSec, o primeiro passo é a criação de um plano de teste. Segundo passo a ser seguido é seleção de todas as permissões necessárias para execução do aplicativo móvel a ser desenvolvido. Cada permissão será um meio para a ferramenta avaliar a confiabilidade e segurança da aplicação, em seguida o testador com o arquivo apk em mãos pode realizar o upload para que a ferramenta avalie o resultado do processo de desenvolvimento em comparação com o plano de teste. Por fim a ferramenta permite ao testador consultar um relatório da análise realizada com o plano de teste. Na Figura 3 é exibida o upload da aplicação para a execução do teste e na Figura 4 é apresentado o

resultado do teste através de um relatório contendo as informações referentes ao planejado e ao desenvolvido.

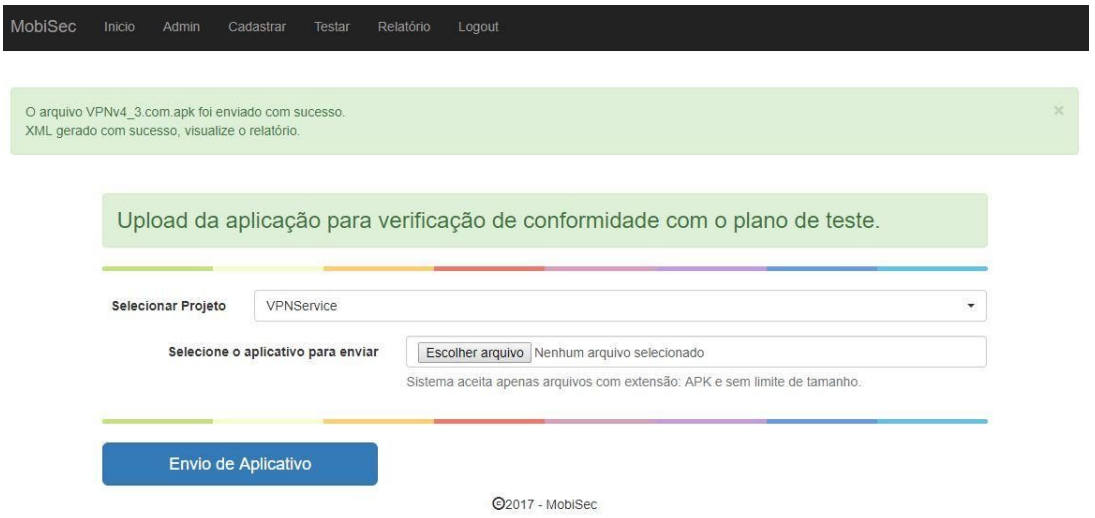

Figura 3 - Upload da aplicação para a execução do teste

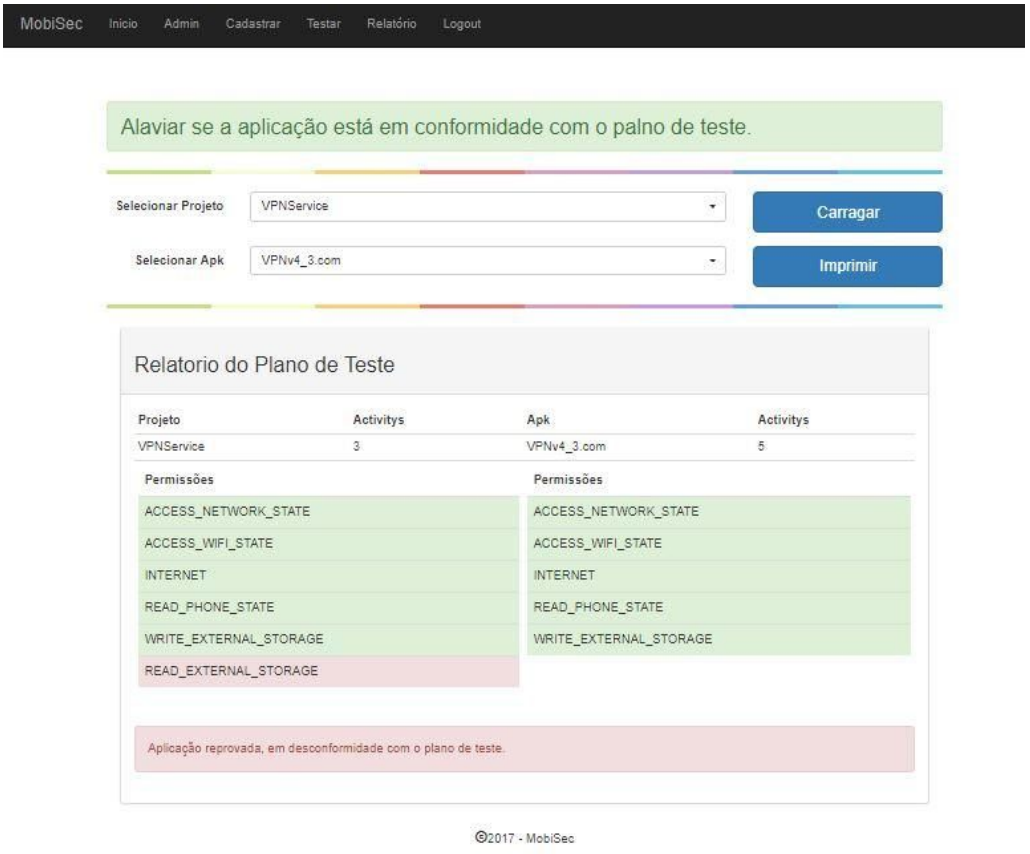

Figura 4 - Resultado do teste através do relatório da aplicação

4.3. Desempenho

A análise de desempenho da ferramenta deu-se através do teste de aplicações com diversos tamanhos e informações. O MobiSec mostrou que seu tempo de teste é inversamente proporcional ao tamanho da apk. O gráfico apresentado na Figura 5 representa o resultado do estudo de desempenho do MobiSec analisando aplicações Android, tomando como base o tamanho do arquivo Android e o tempo decorrido para o processo de análise de segurança da aplicação. Foram utilizadas várias apks para obter um melhor resultado desse estudo.

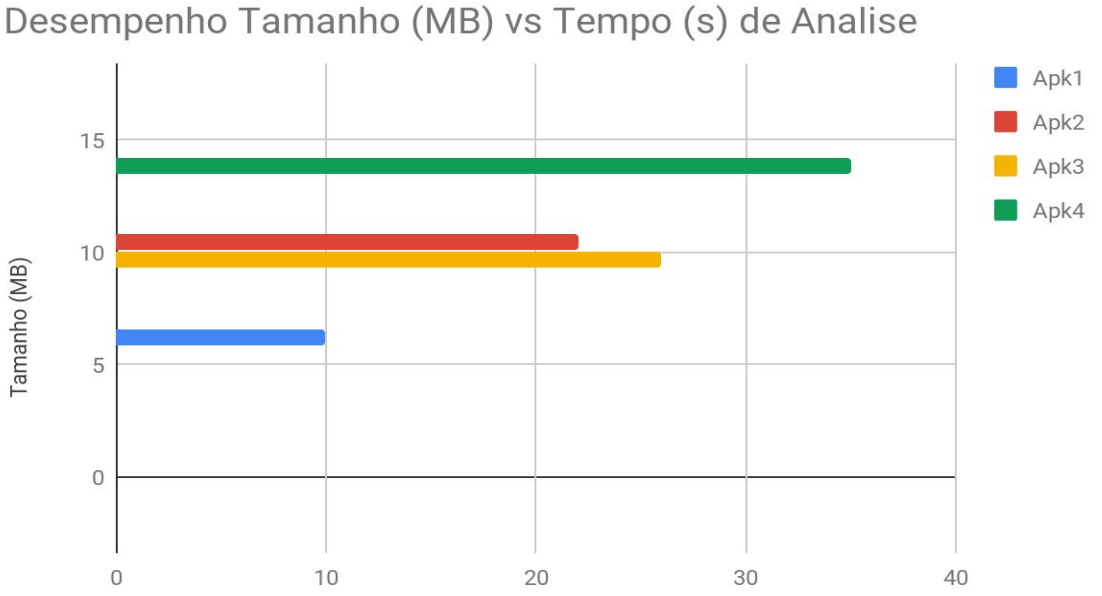

Figura 5 - Resultado de estudo de desempenho do MobiSec

Com base no gráfico acima é possível afirmar que o desempenho da ferramenta MobiSec foi satisfatório levando-se em conta que para a mesma descompilar a apk Android faz-se necessário realizar o upload da aplicação na ferramenta e a depender do tamanho da apk o processo de upload é mais custoso, com acesso ao arquivo AndroidManifest.xml analisa o grau de segurança das permissões adicionadas ao aplicativo ao mesmo tempo que comprara com o plano de teste.

4.4. Avaliação

Para avaliar os resultados obtidos na ferramenta MobiSec utilizamos algumas apks Android coletadas na *internet* de segmentos variados como games e mensageiros, e cinco projetos desenvolvidos pelos pesquisadores como webbrowser e aplicações de cadastros. Consequentemente, foi possível avaliar a capacidade de automação dos testes realizados no MobiSec guiado por um plano de testes.

Nos aplicativos da *internet* adotamos como base as permissões solicitadas pelos dispositivos móveis e registramos na ferramenta. Por ser um teste cego onde o testador não detém acesso ao código fonte do software obteve-se resultados satisfatório

em relação a ferramenta a partir do segundo teste. Nas aplicações que desenvolvemos realizou-se inicialmente o cadastro do projeto e definiu suas permissões, sendo estas informações passadas para o desenvolvedor. Ao realizar o upload dessas apks desenvolvidas com auxílio da ferramenta, observou-se que algumas permissões poderiam gerar insegurança, já que o MobiSec possui armazenado o nível de segurança da liberação de algumas permissões, logo o desenvolvedor é acionado para que gere códigos que dificultem a ação de softwares maliciosos.

Na Tabela 1 é possível confirmar a validação da ferramenta através de cinquenta testes realizados sendo dez aplicações coletadas na *internet* e cinco desenvolvidas pelos pesquisadores totalizando vinte e seis aprovados e vinte e quatro reprovados.

Aplicação	Quantidade	Testes	Aprovados	Reprovados
Internet	10	20	6	14
Desenvolvidas	5	30	20	10

Tabela 1 Resultados obtidos com uso da ferramenta

A partir destas simulações das aplicações coletadas na *internet* foi possível demonstrar que o propósito da ferramenta é ser utilizada após o desenvolvimento onde a mesma serve de apoio para avaliação da aplicação desenvolvida.

Nas desenvolvidas percebeu-se que o processo foi o contrário, logo provoca o desenvolvedor a seguir o que foi planejado para a sua aplicação, proporcionando assim um valor satisfatório para a ferramenta, já que um de seus objetivos foi atingido.

As aplicações que foram aprovadas estavam em conformidade com o plano de testes e tinha um baixo índice de permissões perigosas, já as reprovadas deu-se por estarem em desconformidade com o planejado independente do grau de segurança das permissões solicitadas.

Outro ponto satisfatório é que a ferramenta acusa os riscos das permissões utilizadas. Desse modo, os desenvolvedores ficam mais preparados para elaborar meios que garantam segurança nesses pontos de risco.

Vale ressaltar que todas as avaliações foram realizadas pelos pesquisadores e que a ferramenta está disponível no github de forma open source, possibilitando o download, modificações e melhorias para que se obtenha maiores resultados.

5. Considerações Finais

Neste trabalho, foi apresentada a ferramenta MobiSec com o propósito de automatizar testes de segurança baseados em planos de teste com o propósito de avaliar e garantir a segurança de aplicações para dispositivos Android. Disponível no GitHub pelo seguinte link https://github.com/gomesrocha/MobiSec como ferramenta open source.

Desta maneira nos permitiu avançar na área de testes de segurança por criar um modelo automatizado de testes de segurança para aplicativos mobile que produz relatórios sobre os possíveis problemas de segurança funcionando via Web podendo ser utilizado na nuvem com baixo custo.

Há pretensão quanto a evolução da ferramenta para que na mesma seja acrescida novas rotinas de automatização no plano de testes, rotina de teste se segurança contra *malware*, parâmetros para personalização do limite aceitável de cada grau de segurança das permissões. O propósito é garantir que as avaliações sejam as mais próximas possíveis de um processo de teste realizado manualmente.

Referências

Apktool. (2017). Disponível em: <https://ibotpeaches.github.io/Apktool/>.

Bertoglio, Daniel Dalalana, Zorzo, Avelino Francisco. (2015). Um Mapeamento Sistemático sobre Testes de Penetração, Relatório Técnico: no 084, ano 2015. Disponível em: <http://www3.pucrs.br/pucrs/files/uni/poa/facin/pos/relatoriostec/TR64.pdf>.

Drozer. (2015). Disponível em: <https://www.mwrinfosecurity.com/products/drozer/>

Gargenta, M. (2011). Learning Android. Sebastopol: O'Reilly Media, 2011.

K. Cavalvanti, E. Viana and F. A. A. Lins. (2017). An Integrated Solution for Improving Mobile Device Security Based on the Android Platform. Disponível em:<http://www.ewh.ieee.org/reg/9/etrans/ieee/issues/vol15/vol15issue11Nov.2017/15TLA11_20RegisPiresCavalcanti.pdf>.

Malek, S., Esfahani, N., Kacem, T., Mahmood, R., Mirzaei, N., and Stavrou, A.. (2012). A Framework for Automated Security Testing of Android Applications on the Cloud. DOI 10.1109/SERE-C.2012.39. 2012 IEEE Sixth International Conference on Software Security and Reliability Companion.

Menezes, Pablo Marques, Cardoso, Lanay Marques, Rocha, Fabio Gomes. (2015) Segurança em redes de computadores uma visão sobre o processo de Pentest. https://periodicos.set.edu.br/index.php/exatas/article/view/2258.

MobSF. (2016). Disponível em: <https://github.com/MobSF>.

Souza, Karla Pires de and Angelita Moutin Segoria Gasparotto. (2013). A importância da atividade de teste no desenvolvimento de software. XXXIII Encontro Nacional de Engenharia de Producao. Disponível em: <http://www.abepro.org.br/biblioteca/enegep2013_TN_STO_177_007_23030.pdf>.

Statista. (2018). Global market share held by the leading smartphone operating systems in sales to end users from 1st quarter 2009 to 2nd quarter 2017. Disponível em: <https://www.statista.com/statistics/266136/global-market-share-help-by-smartphone-operating-systems/>.