Um estudo sobre vulnerabilidades do Android. Ferramentas e soluções para o usuário.

Gustavo Amaral[2], Rodrigo Silva[2], Gustavo Rotondo[1], Érico Amaral[1]

[1]Engenharia de Computação – Universidade Federal do Pampa (UNIPAMPA)
Avenida Maria Anunciação Gomes de Godoy, n°1650 – Bagé – RS – Brasil

[2]Análise e Desenvolvimento de Sistemas - Instituto Federal Sul-Rio-Grandense (IFSUL) Av. Leonel de Moura Brizola, n° 2501 – Bagé – RS – Brasil

gustavo.h.amaral@gmail.com, orki2008@gmail.com,
gustavo.rotondo@gmail.com, ericoamaral@unipampa.edu.br

Abstract. *Currently it is recognized the increase in mobile systems, driven by the popularity of smartphones and the use of operating systems in these mobile devices. However, due to the increase of users of such solutions has also an expansion in the amount of information stored on these systems, which makes the security issue an important factor to be considered. Based on this reality, this study sought to identify vulnerabilities in different versions of Android, aiming point which safety applications currently available, would allow users, not experts, to audit their devices. As preliminary results was possible to identify a set of effective software to identify flaws in the operating system in question*

Resumo. *Atualmente é reconhecido o aumento de sistemas móveis, alavancado pela popularização dos smartphones e o uso de sistemas operacionais nestes dispositivos móveis. Contudo, devido o aumento de utilizadores de tais soluções tem-se também uma expansão na quantidade de informações armazenadas nestes sistemas, o que torna o tema segurança um fator importante a ser considerado. Com base nesta realidade, a presente pesquisa buscou identificar vulnerabilidades existentes em diferentes versões do Android, objetivando apontar quais aplicações de segurança, atualmente disponíveis, permitiriam a usuários, não especialistas, auditarem seus dispositivos. Como resultados preliminares foi possível apontar um conjunto de softwares efetivos para a identificação de falhas no sistema operacional em questão.*

1. Introdução

Com a evolução tecnológica, utilizar dispositivos móveis para navegar e manter-se informado, tornou-se algo fundamental no dia a dia das pessoas, conforme Almeida (2014), os sistemas operacionais para dispositivos móveis mais difundidos atualmente são o iOS desenvolvido pela *Apple*[1], o *Windows Phone* produzido pela *Microsoft*[2] e o *Android*, idealizado pela *Google*[3].

Contudo ao contrário do iOS e do *Windows Phone*, o Android é um software livre, possibilitando que cada fabricante possa fornecer sua própria versão customizada do sistema operacional em seu dispositivo, o que levou a instalação de uma grande base deste sistema, contudo a facilidade de manipulação do Android acarretou a inserção de um conjunto considerável de vulnerabilidades. Uma pesquisa da Trend Micro, aponta que dos 2 milhões de aplicativos, disponíveis para *Android*, quase 25% são *malwares e*,

que dentre os 700 mil *Apps* oferecidos pela *Google Play Store,* 10% podem ser considerados aplicativos maliciosos (TECHTUDO, 2013).

O reconhecimento deste conjunto de incidentes, relacionados ao Android, serviram de base para a definição do escopo desta pesquisa, a qual procurou analisar um grupo de ferramentas de segurança, que fornecessem diagnósticos sobre vulnerabilidades identificadas em um conjunto de dispositivos, rodando diferentes versões do Android. A partir dos resultados obtidos, comparou-se a efetividade de cada aplicação e dos sistemas analisados, permitindo desta forma, emitir um parecer sobre softwares realmente eficientes, que possam ser adotados por usuários a fim de auditar seus dispositivos.

Este extrato de pesquisa segue a seguinte estrutura: uma seção inicial de introdução, a seção dois apresentando um referencial teórico pertinente ao estudo, a metodologia e implementação dos testes sobre os aplicativos são descritos na seção três, na seção quatro tem-se os resultados e discussões e, por fim as conclusões são apontadas na seção cinco.

2. Referencial Teórico

Nesta seção será apresentado um levantamento teórico sobre temas pertinentes a pesquisa, os quais descrevem as tecnologias de dispositivos móveis, o sistema operacional Android, incidentes de segurança e, uma breve descrição de trabalhos correlatos.

2.1. Arquitetura para Dispositivos Móveis e o Android

De acordo com Merrick e Gorlenko (2003), há diferentes tipos de tecnologias que podem ser classificadas como tecnologias móveis, porém existe um aspecto comum entre todas elas: a possibilidade da portabilidade.

A expansão dos dispositivos móveis foi, em grande parte, ocasionada pela evolução da arquitetura ARM (*Advanced RISC Machines*), por possuir um conjunto de componentes com uma boa capacidade de processamento, custo baixo, um desempenho satisfatório, tornando-se assim uma das arquiteturas para diversos tipos de aparelhos eletrônicos, tais como *tablets*, *smartphones*, *smart tvs*, entre outros. Dentre as tecnologias incorporadas aos *smartphones* atuais, destacam-se os sistemas operacionais iOS (Apple), o Windows Phone (Microsoft) e o Android (Google). Cada sistema operacional trabalha sobre uma plataforma/arquitetura de desenvolvimento *web mobile* ou nativa.

Dentre os SO para dispositivos móveis ressalta-se o Android, que segundo Carrenho (2015) é o sistema predominante do Brasil, estando presente em 89,5% destes equipamentos (Gráfico 01).

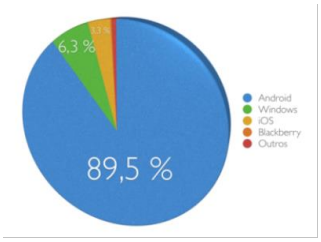


Gráfico 01. Uso do Android no Brasil/2015

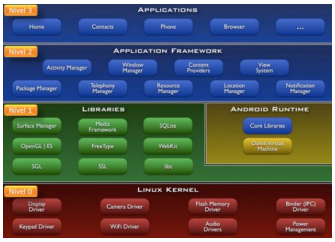


Figura 01. Arquitetura do Android

O Android é uma pilha de softwares para dispositivos móveis que inclui um sistema operacional, um *middleware* e um conjunto de aplicações chaves, sendo que os desenvolvedores podem criar aplicações para a plataforma usando o Android SDK por

exemplo. Este sistema é constituído de uma pilha de camadas de *software* que roda em cima de um sistema operacional, baseado em Linux. Sobre este conjunto de camadas tem-se uma máquina virtual denominada Dalvik, bibliotecas Java e as aplicações de usuário (Figura 01), conforme Aquino (2007).

2.2. Incidentes e Vulnerabilidades em Dispositivos Móveis

Segundo Quintão *et al.* (2010) muitos usuários têm a falsa sensação de segurança quanto ao seu dispositivo móvel, pois os indivíduos, em sua maioria, acreditam que seu dispositivo possui conexões ativas durante uma ligação ou quando estão navegando na Internet. Contudo, o problema está na instalação de diferentes *Apps*, sem o devido conhecimento de suas operações internas no aperelho e permissões necessárias para o seu funcionamento. Muitos desses aplicativos, rodam em *background,* processos que acessam informações e dados no dispositivo, sem que o usuário reconheça tais ações, demonstrando, desta forma, a vulnerabilidade de tais dispositivos.

Para Almeida (2013) o aumento da utilização de *smartphones* está diretamente ligado ao grande número de vulnerabilidades identificadas nestas plataformas, as quais são exaustivamente exploradas pelos atacantes, por meio de técnicas como *Phishing*, *Trojans*, *Spyware*, *Bots*, *Root Exploits*, entre outras. Esta situação é agravada pelo fato que poucos usuários utilizam ferramentas objetivando aprimorar o nível de segurança de seus dispositivos.

Neste contexto, o estudo de Acker *et al.* (2010) afirma que os dispositivos móveis geralmente apresentam falhas de segurança em sua concepção, sendo aconselhável que os usuários obtenham ferramentas capazes de corrigir estas vulnerabilidades em seus dispositivos. A fim de evitar a proliferação destas falhas, foram concebidos as CVEs (*Commom Vulnerabilities and Exposures*), uma base de dados pública e internacional que promove informações sobre fragilidades de segurança contidas em diferentes equipamentos/plataformas móveis.

2.3. Trabalhos Correlatos

No intuito de identificar o estado da arte sobre o tema segurança em dispositivos Android, investigou-se um conjunto de trabalhos correlatos, dentre os quais destacaram-se:

O estudo de Inácio (2014), apresenta uma análise sobre a arquitetura e funcionamento da plataforma Android, elencando fatores relacionados à segurança, permissões de acesso e escalonamento de privilégios. Como resultado o autor descreve diversas situações em que o descuido com o conjunto de permissões cedidas a aplicações neste sistema podem levar à execução de operações não autorizadas.

Paralelamente as preocupações inerentes ao uso da plataforma Android, existe a necessidade da criação de *Apps* baseados em padrões robustos de segurança, como propõe o projeto de Batista (2013), intitulado "Análise da Segurança de Aplicativos na Plataforma Android Através da Adoção de Patterns". Neste estudo foram executados, de forma experimental, diversos testes sobre o sistema operacional, apontando hipóteses de soluções para a transmissão segura de dados, com a utilização de *patterns* durante a criação de aplicativos, mostrando a efetividade deste tipo de projeto.

Apesar dos trabalhos, aqui descritos, apresentarem conceitos importantes sobre segurança para o Android, sua concepção é voltado para um contexto teórico, diferente desta pesquisa, que visa de forma prática avaliar aplicações e apontar soluções para a identificação efetiva de vulnerabilidades neste sistema.

3. Metodologia e Implementação

Como atividade inicial desta pesquisa elencou-se, a partir de uma estudo preliminar, possíveis ferramentas para a análise de vulnerabilidades em dispositivos rodando a plataforma Android, dentre os quais foram selecionados o *OpenVas* (uma solução robusta e a aberta, com vários serviços e ferramentas que oferecem a varredura e gerenciamento de dispositivos) e, mais sete aplicativos disponibilizados na *Google Play Store*, voltados também, para a área de análise e gerenciamento de segurança de dispositivos móveis. Após a definição de todos os softwares a serem utilizados nos experimentos, definiu-se o conjunto de dispositivos a serem avaliados pelas aplicações (Tabela 01).

Tabela 01. Dispositivos utilizados nos experimentos

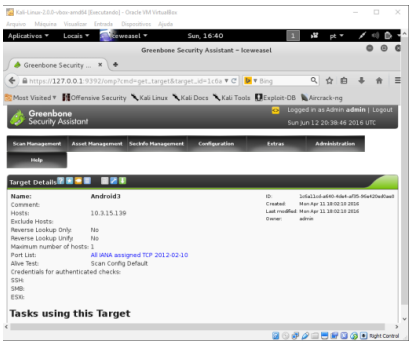
Dispositivo	Modelo	Versão do Android	Processador
Android1	Moto G 1º geração	Android 5.1 (Lollipop),	Quad Core 1.2 GHz
Android2	Galaxy S3 Mine	Android 4.2.2 (Jelly Bean)	Dual-Core 1 GHz
Android3	Galaxy Win2	Android 4.4.4 (KitKat)	Quad Core 1.2 GHz
Android4	Galaxy Tab 2 7.0	Android 4.0.3 (Ice Cream Sandwich)	Quad Core 1.2 GHz

Dentre o conjunto de aplicações, disponibilizadas pela loja da Google, optou-se por adotar soluções de fácil utilização, as quais pudessem ser instaladas e executadas por usuários sem conhecimento técnico. Com base neste princípio elencou-se os softwares descritos na Tabela 02.

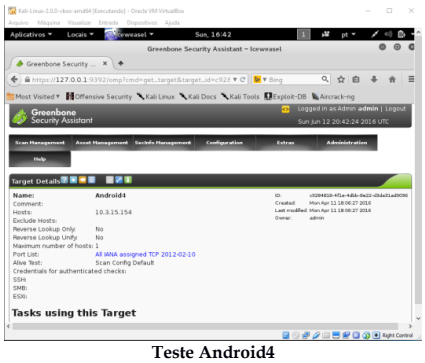
Tabela 02. Aplicativos avaliados

Aplicativo	Descrição
Stagefright Detector	Este software possui três funções principais: (1) verificar se o dispositivo é vulnerável, (2) à quais CVEs, (3) se existem atualizações para o sistema.
Shellshock Vulnerability Check	Aplicação que determina se o dispositivo está com o modo *BASH* habilitado e, se possui algum aplicativo com um processo *BASH* ativo.
Bluebox Secuity Secanner	Solução utilizada para analisar o dispositivo e determinar se o sistema está vulnerável para qualquer tipo de "Fake ID" ou "chave mestra", falhas de segurança que afetam a maioria dos dispositivos Android.
Stuboorn Trojan Killer	Tem como função explorar e eliminar trojans que não podem ser apagados por outros aplicativos comuns de antivírus.
Avira Vulnerability Check	Realiza o teste para a vulnerabilidade "*OneClassToRuleThemAll*" descrito pela CVE-2015-3825.

As atividades experimentais consistiram em três etapas distintas: na primeira foram analisados os quatro dispositivos, utilizando a ferramenta *OpenVas*; na segunda etapa foram avaliadas os *Apps* descritos na Tabela 02; por fim, na última etapa efetuou-se a análise e avaliação dos resultados alcançados.



Teste Android3

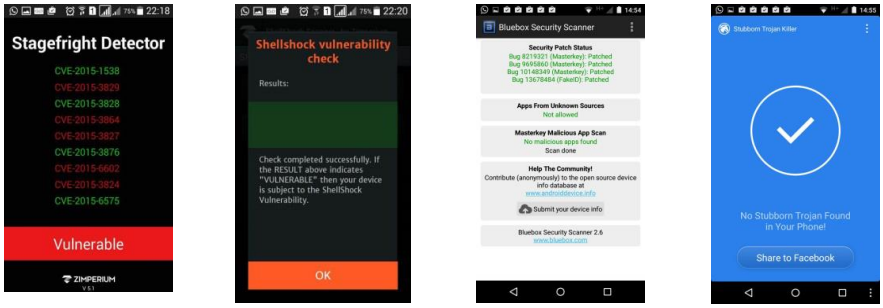


Teste Android4

Figura 02. Experimento utilizando o *OpenVas*

Na primeira etapa utilizou-se o *OpenVas*, configurado e instalado em um desktop, rodando o sistema operacional Kali Linux 2.0. Por meio da configuração do *Greenbone Security* (painel de controle) criou-se um conjunto de tarefas para análise dos dispositivos da Tabela 02. Devido ao fato do *OpenVas* não possuir uma biblioteca de testes específicos para o Android, optou-se por utilizar uma análise que contemplasse o maior número de NVTs (*Network Vulnerability Tests*)) possíveis. O modo de scan F*ull and fast ultimate*, adotado no experimento, possuía até a data dos testes 45.286 NVTs, as quais foram agrupadas em 59 famílias (tipos de testes). A Figura 02 mostra a execução do *OpenVas* sobre os dispositivos Android3 e Android4.

Na segunda etapa, após a instalação das aplicações, em todos os dispositivos descritos na Tabela 01, observou-se o resultado apresentado por cada uma das ferramentas. A execução nos dispositivos Android1 e Android2 é apresentada na Figura 03.



Android1 – StageFright Detector | **Android1 - Shellshock Vulnerability Check** | **Android2 - Bluebox Secuity Secanner** | **Android2 - Stuboorn Trojan Killer**

Figura 03. Experimento – Amostra de testes da Etapa 2

A última etapa da implementação consistiu na coleta e organização dos resultados alcançados com os experimentos. A discussão e avaliação destes são apresentadas na seção 4.

4. Resultados e Discussões

Com os *Apps* testados e com o posterior teste utilizando-se o *OpenVas* nos dispositivos móveis, chegou-se a um conjunto de resultados, tanto para as aplicações como para o *OpenVas.* Os resultados encontrados através dos *Apps* observam-se na Tabela 03 enquanto os resultados ao utilizar-se o *OpenVas* podem ser constatados na Tabela 04.

Tabela 03. CVEs detectadas no experimento

CVE	Android1	Android2	Android3	Android4
CVE-2015-3829	X		X	
CVE-2015-3864	X		X	
CVE-2015-3827	X		X	
CVE-2015-6602	X	X	X	
CVE-2015-3824	X		X	
CVE-2015-3825	X			
CVE-2015-3876		X		
CVE-2015-6575			X	

Tabela 04. Resultados *OpenVas*

Dispositivo Vulnerabilidade	Source routed packets (nível baixo)	TCP Sequence Number Approximation Reset Denial of Service Vulnerability (nível médio)	Firewall ECE bit bypass (nível alto)
Android1	X	X	
Android2	X	X	
Android3	X	X	X
Android4			

As aplicações testadas retornaram apenas algumas CVEs, sendo que cada CVE representa alguma brecha que poderá vir a ser explorada por um agente mal intencionado. A CVE-2015-3829 diz respeito a Off-by-one error in the MPEG4Extractor que no android, antes do 5.1.1, permite que atacantes remotos possam executar código arbitrário ou causar uma negação de serviço, a CVE-2015-3864 existe devido a uma correção incompleta para CVE-2015-3824, a CVE-2015-3827 no android antes do 5.1.1 não valida a relação entre os tamanhos do pedaço e ignorar tamanhos, o que permite que atacantes remotos executem códigos arbitrários ou causar uma negação de serviço, a CVE-2015-6602 permite que atacantes remotos possam executar código arbitrário através de meta-dados trabalhados em um MP3 ou arquivo MP4, a CVE-2015-3824 refere-se que no android antes do 5.1.1 não restringe adequadamente além de tamanho, o que permite que atacantes remotos executem códigos arbitrários ou causar uma negação de serviço, a CVE-2015-3825 é duplicada sendo a válida a CVE-2015-3837, a CVE-2015-3876 permite que atacantes remotos possam executar código arbitrário através de meta-dados trabalhados em um MP3 ou arquivo MP4 e a CVE-2015-6575 diz o android, antes do 5.1.1, não considera adequadamente promoção inteiro, o que permite que atacantes remotos executem códigos arbitrários ou causar uma negação de serviço, esta vulnerabilidade existe devido a uma correção incompleta para CVE-2014-7915, CVE-2014-7916, e /ou CVE-2014-7917.

Os resultados do *OpenVas* foram classificados por nível de ameaça, conforme a Tabela 04. O *Source routed packets* (nível de risco baixo) aponta que o host remoto aceita pacotes IP roteados. Este é um recurso de teste, contudo um atacante pode usá-lo para contornar filtros IP mal projetado e explorar outras falhas do sistema. No entanto, não é considerada uma vulnerabilidade perigosa. O *TCP Sequence Number Approximation Reset Denial of Service Vulnerability* (nível de risco médio) identifica se o host local está executando serviços TCP e, desta forma, vulnerável a ataques de negação de serviço. Por fim, o *Firewal ECE-bit bypass* (nível de risco alto) demonstra que o host remoto está vulnerável a um *bug*, através do qual um atacante pode contornar o *firewall*, definindo o bit ECE dentro do campo flags TCP, fragilizando desta forma o sistema atacado.

Pode se observar nas Tabelas 03 e 04 que o dispositivo Adroid4 não apresentou nenhuma vulnerabilidade ao ser analisado, tanto pelo *OpenVas* quanto pelos demais aplicativos.

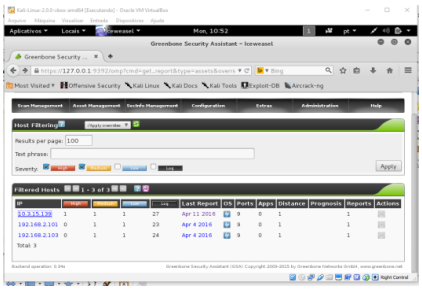


Figura 04. Vulnerabilidades por nível de ameaça

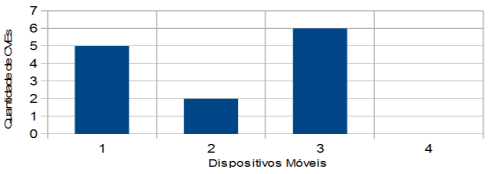


Gráfico 02. Acumulado de Vulnerabilidades por dispositivo

Os aplicativos *Shellshock Vulnerability Check, Bluebox Secuity Secanner* e *Stuboorn Trojan Killer* não identificaram vulnerabilidades em nenhum dos dispositivos avaliados. O *Avira Vulnerability Check* apontou a CVE 2015-3825 apenas para o

Android1. A ferramenta *Stagefright Detector* identificou o maior número de resultados positivos para CVEs, conforme observado no Gráfico 02.

Analisando os resultados obtidos nas etapas 1 e 2, observa-se que as vulnerabilidades *Source routed packets* e *TCP Sequence Number Approximation Reset Denial of Service Vulnerability* são comuns nos dispositivos Android1, Android2 e Android3, de onde supõe-se que estas sejam falhas decorrentes neste tipo de sistema operacional. Já a vulnerabilidade *Firewal ECE-bit bypass* só foi verificada no dispositivo Android3.

Os resultados dos aplicativos sugerem que estas ferramentas são específicas para cada caso de análise, das cinco ferramentas testadas apenas duas retornaram algum tipo de vulnerabilidade nos dispositivos móveis, sendo que o *Stagefright Detector* demonstrou uma maior efetividade, sendo responsável por encontrá-las nos dispositivos Android1, Android2 e Android3 sendo que a CVE-2015-6602 é uma vulnerabilidade recorrente nos 03 dispositivos testados e as CVE-2015-3829, CVE-2015-3864, CVE-2015-3827 e CVE-2015-3824 são vulnerabilidades encontradas no Android1 e Android2 e a aplicação *Avira Vulnerability Check* apontou a CVE 2015-3825 para o Android1.

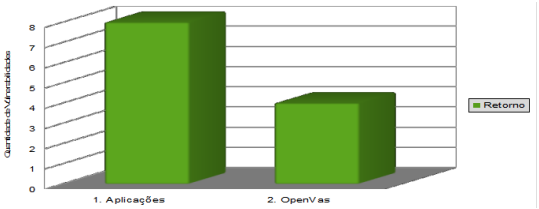


Gráfico 03. Ferramenta/Quantidade de Vulnerabilidades

Comparando a execução dos aplicativos e do *OpenVas*, percebe-se que este último apontou um conjunto bem menor vulnerabilidades (Gráfico 03), situação que demonstra a viabilidade dos usuários de reconhecer as fragilidades de seus dispositivos, por meio de softwares gratuitos, disponibilizados pela *Google Play Store*.

Cabe ressaltar que algumas vulnerabilidades conhecidas dos modelos avaliados não foram diagnosticadas com as aplicações testadas, caso por exemplo do Android1 com a versão 5.1(Lollipop) a qual apresenta a CVE-2015-1474 e a CVE-2015-1530, e o Android2 com a versão 4.2.2(Jelly Bean) que possui a CVE-2014-1939 e a CVE-2016-3840.

5. Conclusões

Esta pesquisa teve por finalidade apresentar um estudo sobre vulnerabilidades no sistema operacional Android, motivada pelo aumento considerável na utilização de dispositivos móveis com esta plataforma e, pela necessidade de soluções que agreguem níveis mais efetivos de segurança a tais sistemas.

Com base neste contexto foram avaliadas um conjunto de cinco aplicações disponíveis na loja da Google, as quais podem ser utilizadas por usuários sem experiência técnica e o software *OpenVas*, uma solução técnica para análise de vulnerabilidade, comumente utilizadas em ambientes computacionais mais complexos. Avaliou-se o resultado obtido por estas ferramentas em um experimento com quatro dispositivos rodando o sistema Android.

Concluiu-se ao final, desta fase da pesquisa, que o *OpenVas* não se demonstrou uma solução efetiva para a identificação de vulnerabilidades, ao contrário das aplicações *Stagefright Detector* e *Avira Vulnerability Check*, demonstrando que usuários sem

experiência técnica podem auditar seus dispositivos e, identificar o nível de segurança do sistema, embora estas aplicações não retornem aos usuários todas as vulnerabilidades dos dispositivos móveis, que estejam propensas a serem exploradas.

Referências

ACKER, E. V., WEBER, T. S., CECHIN, S. L. - Injeção de falhas para validar aplicações em ambientes móveis – 2010. Disponível em: http://sbrc2010.inf.ufrgs.br/anais/data/pdf/wtf/st02_01_wtf.pdf.

ALMEIDA, Igor P. ASSUNÇÃO, Letícia R. SIMÕES, Thállys L, LIMA, Joselice F. Visão Sobre Dispositivos e Sistemas Operacionais Móveis. Anais dos Simpósios de Informática do IFNMG - Câmpus Januária, 2014. Disponível em: http://200.131.5.234/ojs/index.php/anaisviiisimposio/article/view/45. Acessado em 10 de junho de 2016.

ALMEIDA, J. ANÁLISE DA SEGURANÇA E DE FERRAMENTAS NA PLATAFORMA ANDROID, 2013 - Disponível em: http://painel.passofundo.ifsul.edu.br/uploads/arq/20160330212016137870 2704.pdf.

AQUINO, Juliana F. S. Plataformas de Desenvolvimento para Dispositivos Móveis. PUC – RJ, 2007. Disponível em: http://www-di.inf.puc-rio.br/~endler/courses/Mobile/Monografias/07/Android-Juliana-Mono.pdf.

CARRENHO, C. Brasil: o país do tablet, smartphone e Android, 2015.Disponível em: http://www.tiposdigitais.com/2015/06/brasil-o-pa%C3%Ads-do-tablet-smartphone-e-android.html.

https://cve.mitre.org/.

GORLENKO, L., MERRICK. No wires attached: Usability challenges in the connected mobile world. IBM Systems Journal. Vol 42, no 4, 2003. Disponível em http://www.research.ibm.com/journal/sj /424/gorlenko.pdf.

QUINTÃO, P. MISAGHI, M. P. S. C. C. S. NOVAIS, E. B. - Análise dos Desafios e Melhores Práticas Para Resguardar a Segurança e Privacidade dos Dispositivos Móveis no Uso das Redes Sociais - 2010. Disponível em: https://www.researchgate.net/profile/Mehran_Misaghi/publication/274706269_ANLISE_DOS_DESAFIOS_E_MELHORES_PRTICAS_PARA_RESGUARDAR_A_SEGURANA_E_PRIVACIDADE_DOS_DISPOSITIVOS_MVEIS_NO_USO_DAS_REDES_SOCIAIS/links/5526703b0cf21e126f9dafa2.pdf.

TECHTUDO. Um em cada dez apps na Google Play Store é malware, alerta pesquisa. 2013. Disponível : < http://www.techtudo.com.br/noticias/noticia/2013/03/um-em-cadadez- apps-na google-play-store-e-malware-alerta-pesquisa.html>.