Uma Solução para Avaliação de Riscos em Aplicativos para Dispositivos Móveis

Wilson Jacobsen[1], Mariana Pompeo Freitas[1], Érico Amaral[1]

[1]Universidade Federal do Pampa (Unipampa)
Caixa Postal 96400–000 – Bagé – RS – Brazil

{willl.jacobsen, maripompeof}@gmail.com, erico.amaral@unipampa.edu.br

Abstract. *This article presents the statement of assurance as to research on mobile devices running Android and its features.*
The aim is to present risk identification methods related to the applications installed on the mobile device. To achieve the objectives of this study was proposed and implemented a risk analysis model. This work has results, among which we mention: a detailed study of standard permissions for Android applications and an assessment of potential aggregate risks to each of these, the creation and validation of a new risk analysis model and implementation of a application to automate the model created.

Resumo. Este artigo apresenta o extrato de uma pesquisa sobre segurança em dispositivos móveis com Sistema Operacional Android e suas características. O intuito é apresentar métodos de identificação de risco relacionados aos aplicativos instalados no dispositivo móvel. Para atingir os objetivos desse trabalho foi proposto e implementado um modelo de análise de risco. Este trabalho possui resultados, dentre os quais citam-se: um estudo detalhado das permissões de padrões para aplicativos Android e uma avaliação dos potenciais riscos agregados a cada uma destas, a criação e validação de um novo modelo de análise de risco e implementação de um aplicativo para automatizar o modelo criado.

1. Introdução

De acordo com Tonin (2012) a evolução da tecnologia para sistemas móveis, aliada ao crescimento do uso de sistemas de armazenamento de dados, sistemas de localização, redes sociais e também pela disponibilidade insaciável de novos aplicativos no mercado aumenta de forma exponencial a dependência de um dispositivo móvel confiável e seguro.

A ocorrência do incidente de segurança em celulares ou tablets é agravada devido à grande conectividade e processamento oferecido por estes aparelhos, e a restauração e manutenção dos dados comprometidos em um incidente são procedimentos caros e muitas vezes não trazem o benefício esperado. Uma alternativa para a redução deste tipo de transtorno é investir na análise e correção das

vulnerabilidades dos sistemas, o que pode ser feito utilizando scanners como Nessus[1] e Openvas[2]. Um dos grandes problemas destes analisadores são as inconsistências dos dados apresentados. É perigoso ter a impressão de que não existem vulnerabilidades quando, na realidade, as brechas estão por todo lado e disponíveis aos fraudadores interessados. Além disso, tal prática requer conhecimento específico e gastos que um usuário comum de smartphone ou tablet não estaria disposto a pagar.

Tendo em vista essa situação, o estudo de novas técnicas de segurança torna-se extremamente necessário, priorizando uma tentativa de contenção a invasões e mau uso de aplicativos, logo, este trabalho propõe um estudo e criação de uma ferramenta de análise de potenciais riscos em aplicativos, objetivando fornecer uma maneira do próprio usuário saber o risco que está correndo em manter um determinado aplicativo instalado em seu dispositivo móvel.

A estrutura adotada nesse artigo é composta de 5 seções organizadas da seguinte forma: na seção 2 serão apresentados os referenciais teóricos, na seção 3 é apresentado a metodologia, na seção 4 têm-se a implementação e resultados e por fim, na seção 5 há as considerações finais.

2. Referencial Teórico

Segurança da informação e privacidade são problemas para os usuários de todos os tipos de dispositivos eletrônicos. Chin et al. (2012) afirmam que os usuários devem se preocupar mais com a privacidade em seus dispositivos móveis do que em seus computadores, e especialmente se preocuparem com as ameaças de aplicativos maliciosos, já que esses dispositivos são normalmente de uso pessoal. Segundo Cibrão et al. (2014), como qualquer sistema, uma plataforma como o Android está sujeito à existência de falhas de seguranças e bugs, que introduzem vulnerabilidades. A fácil instalação e o grande acervo de aplicativos, induz o usuário sem preocupação a instalação de muitos aplicativos, sendo vários deles desconhecidos.

De acordo com Felt et al. (2012), a principal vulnerabilidade do Android é o fácil acesso dos recursos, necessitando apenas da aprovação das permissões na hora da instalação do aplicativo. Schultz et al. (2011) explicam ainda que isso é um grave problema de segurança, pois o usuário em sua grande maioria, não presta atenção nas permissões dadas aos aplicativos.

Diederich et al. (2013), ressaltam que por questões de segurança e privacidade, a maioria das decisões se relaciona com o risco a que o indivíduo ou o sistema estão ligados, e com a maneira que estes dados estão dispostos para o usuário. Tornou-se habitual conceber julgamentos sobre a percepção do risco e sobre a tomada de decisão de dois modos distintos de pensamento ou sistemas definidos como 1 e 2. O Sistema 1 é automático e intuitivo, opera fora da consciência, ao passo que o Sistema 2 requer atenção, é mais lento e lógico que o Sistema 1, porém é mais preciso conforme Kahneman (2011). O Sistema 1 mesmo apresentando resultados menos precisos acaba

---

1 *Software pago* de *scanner* de vulnerabilidade (Muniz, 2014).

2 *Software free* de *scanner* de vulnerabilidade (Muniz, 2014).

sendo mais aproveitado pelo usuário. De acordo com Finucane et al. (2000) isso se deve ao fato de que a tomada de decisões é lógica e automática, enquanto no Sistema 2 os processos contribuem para julgamentos mais deliberados.

Uma maneira de se evitar problemas de segurança é realizando uma avaliação de risco, que consiste em identificar um comportamento malicioso e posteriormente, desinstalá-lo. As permissões de aplicativos são muito exploradas para análise de risco como explicam Wang et al. (2014): em primeiro lugar emprega-se três técnicas de classificação de recursos, para avaliar o risco de cada permissão. Em segundo, os conjuntos de permissões são avaliados por métodos seletivos em subconjuntos, para investigar o risco introduzido pela colaboração de várias permissões. Em terceiro, são analisados os Apps segundo as aplicações de alto risco. Por fim, apresenta-se os riscos aos usuários. Chin et al. (2012), recomendam a criação de uma nova forma de fornecer indicadores de segurança, e assim, aumentar a confiança do usuário em sua escolha de novos aplicativos e ressaltam que, caso os recursos de segurança sejam complicados os usuários não os utilizarão. Segundo Schultz et al. (2011), as interações entre os usuários e os sistemas precisam ser simples. Portanto não basta ter um modelo de análise de risco eficiente, a utilização do aplicativo deve ser intuitivo para o usuário e os resultados da análise de riscos devem ser de fáceis interpretação.

3. Metodologia

Este trabalho objetivou a implementação e validação de um modelo para avaliação de potenciais riscos encontrados em aplicativos para dispositivos móveis. Dessa forma, existirá uma maneira para que os usuários avaliem os riscos ao manter os aplicativos no seu dispositivo. Para atingir os objetivos propostos, o trabalho foi dividido em 9 etapas, como mostrado na figura 1.

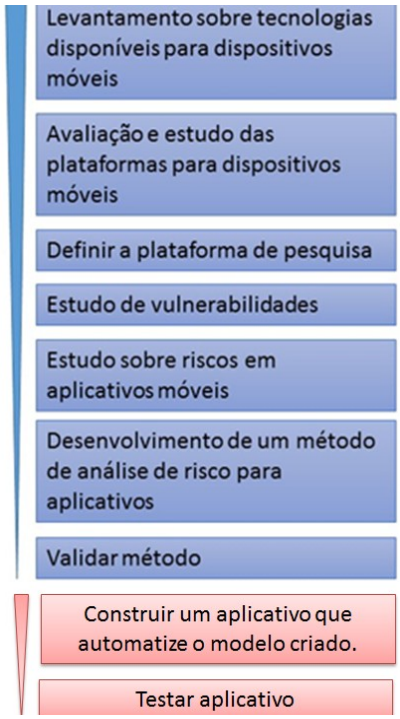


Figura 1: Etapas adotadas no trabalho

A primeira etapa baseou-se, em um estudo teórico sobre o estado da arte em relação ao problema de pesquisa, com o intuito de construir uma base de conhecimento necessária para a avaliação dos dispositivos móveis.

Na segunda e terceira etapa foi definida a plataforma utilizada para a pesquisa, levando em conta sua importância no mercado de dispositivos móveis. A plataforma escolhida foi o Android, pois de acordo com o Statcounter (2014), é o SO mais usado no mundo.

A quarta etapa se resumiu em uma pesquisa sobre vulnerabilidades encontradas nas diferentes funcionalidades do sistema operacional escolhido, na quinta etapa foi pesquisado a identificação dos riscos que o usuário tem ao utilizá-lo. Na sexta etapa construíu-se uma métrica para avaliar vulnerabilidades de Apps com a finalidade de encontrar qual apresenta a maior criticidade. Para a criação do modelo de análise de risco, primeiramente foram identificadas maneiras de se obter acesso às permissões pedidas por aplicativos já instalados, além disso, foram estudadas as características e riscos de cada permissão, disponibilizadas para os desenvolvedores de Apps. A sétima etapa consistiu em validar o método de análise de risco desenvolvido. Após, no oitavo passo ocorreu a automatização do modelo de análise de risco validado.

Após a implementação do aplicativo, o último passo se deu com a realização de testes a fim de saber se os resultados fornecidos pelo App estavam de acordo com o do modelo de risco criado, e se apresentava uma boa usabilidade, requisito para a aceitação do aplicativo pelo público. Os testes foram realizados em um smartphone LG, com sistema operacional Android 4.4.2. O aparelho possuía 13 aplicativos instalados, todos baixados da loja oficial Google Play: Calculadora, Contatos favoritos, Downloads, Facebook, Instagram, Luz do flash, Messenger, QuickMemo+, Retrica, Security Dog, Snapchat, WhatsApp e Flashlight, dentre esses apps não há nenhum malware. Além desses, foi utilizado uma máquina virtual onde estavam instalados 2 aplicativos: SecurityDog e Google play Store App e 2 malwares: AlSalah e Tap Snake ambos descobertos pela empresa Symantec (2014).

Os testes foram divididos em duas seções, uma descreve os testes funcionais e a outra que cita os testes de interface, realizados com os usuários.

4. Implementação e Resultados

Esta seção tem como objetivo apresentar a solução proposta após os estudos realizados. Foi possível observar que as maiores falhas de segurança são: o fato de o Android solicitar as permissões necessárias para instalar um aplicativo apenas uma única vez, e a falta de interesse dos usuários em estudar as características e riscos dessas.

O modelo criado foi resultado de uma grande pesquisa bibliográfica sobre o assunto, e com a evolução dessa análise bibliográfica foram criados protótipos de modelos capazes de analisar ou não o risco dos aplicativos. Cada modelo forneceu um grau de risco diferente para o mesmo conjunto de permissões solicitadas pelo aplicativo durante o processo de instalação, o que possibilitou verificar as falhas existentes nas propostas. Assim os novos modelos criados herdavam processos corretos e corrigiam os erros do modelo anterior. Após estudos, análise e validação dos modelos chegou-se a

proposição de uma solução final para a análise de risco, a qual segue uma sequência específica, conforme ilustrado na figura 2.

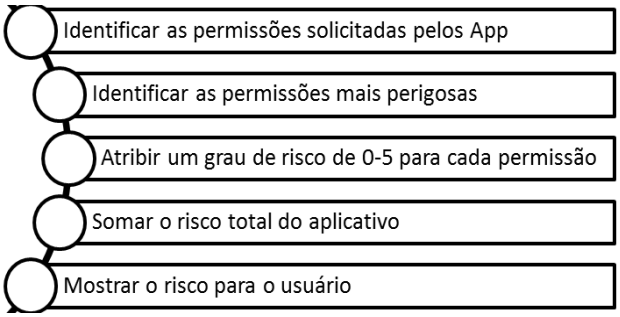


Figura 2: Fluxograma do modelo final

O modelo final foi construído com um grau de risco diferente para cada permissão, a fim de ampliar a diferença entre os aplicativos vulneráveis e não vulneráveis. Nesse modelo a atribuição de grau de risco se dará baseado no trabalho de Gates et. al. (2014) que identificaram o percentual de vezes que uma permissão de alto risco é solicitada por um aplicativo infectado, como observa-se no gráfico da figura 16, nela existem 3 barras: Preta identificando o percentual de vezes que as permissões são solicitadas pelos aplicativos no ano de 2011, Branca indicando o percentual de vezes que as permissões são solicitadas pelos aplicativos em 2012 e Cinza identificando o percentual de vezes que uma permissão é pedida pelos *malwares* descobertos na plataforma Android.

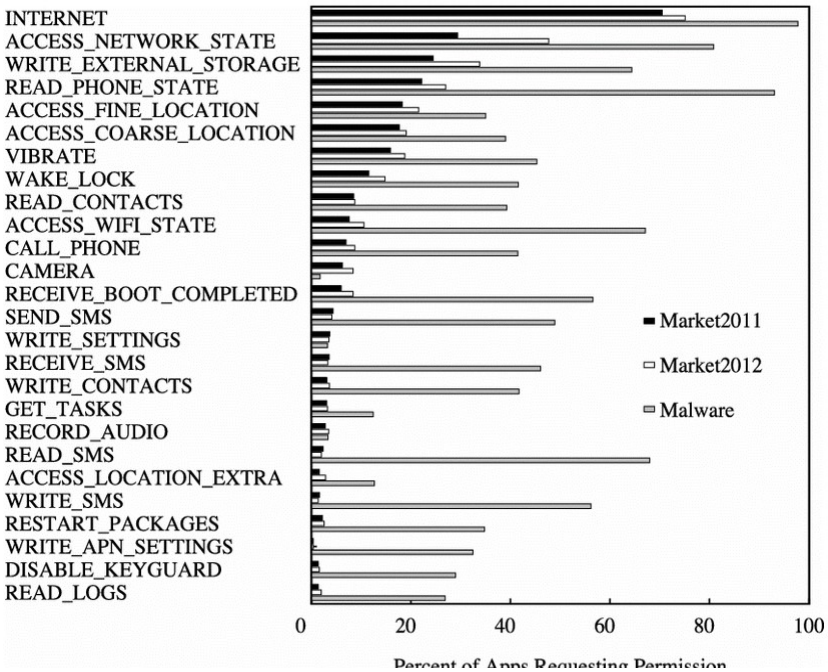


Figura 3: As permissões mais usadas por Malwares

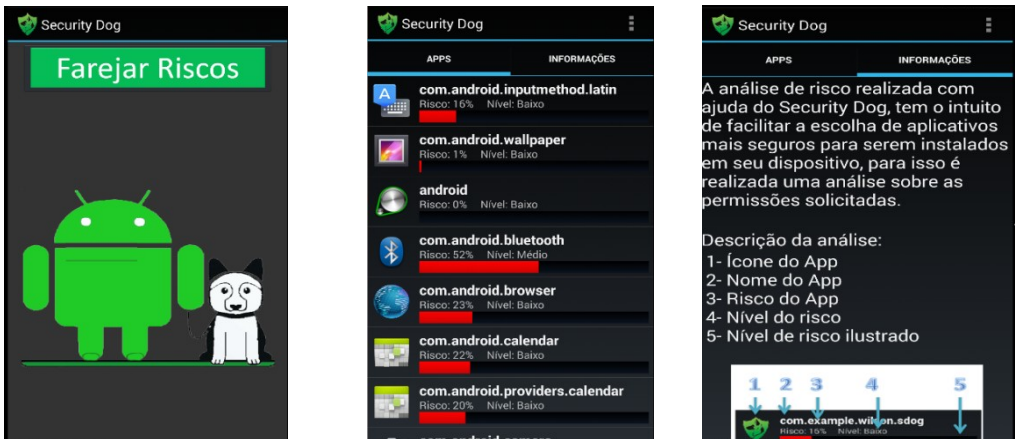
Utiliza-se esses dados estatísticos para atribuir um grau de risco em cada aplicativo conforme mostrado na Tabela 1. Assim conseguiu-se aumentar o grau de risco dos aplicativos maliciosos.

Tabela 1: Lista das permissões de risco e o seu respectivo risco.

Risco	Apps Maliciosos (%)	Permissões
1	Solicitadas por 01% a 20% dos malwares	ACESS_LOCATION_EXTRA; RECORD_AUDIO; GET_TASKS; CAMERA;
2	Solicitadas por 21% a 40% dos malwares	READ_LOGS; DISABLE_KEYGUARD; WRITE_APN_SETTINGS; RESTART_PACKAGES; READ_CONTACTS; ACESS_FINE_LOCATION; ACESS_COARSE_LOCATION;
3	Solicitadas por 41% a 60% dos malwares	WRITE_SMS; WRITE_SETTINGS; WRITE_CONTACTS; RECEIVE_SMS; SEND_SMS; VIBRATE; RECEIVE_BOOT_COMPLETED; CALL_PHONE; WAKE_LOCK;
4	Solicitadas por 61% a 80% dos malwares	READ_SMS; WRITE_EXTERNAL_STORAGE; ACESS_WIFI_STATE;
5	Solicitadas por 81% a 100% dos malwares	INTERNET; ACESS_NETWORK_STATE; READ_PHONE_STATE;

O modelo funciona da seguinte forma: Identifica as permissões pedidas por cada aplicativo, para isso utiliza-se o programa *Permission Explorer*; Identifica entre todas, apenas as permissões de alto risco, presente na maioria dos aplicativos maliciosos de acordo com Portões *et al.* (2014), que identificaram as permissões mais encontradas em aplicativos vulneráveis; Atribui o grau de risco de 0-5 para cada aplicativo de alto risco de acordo com a Tabela 1; Soma-se os riscos de todas as permissões encontradas; Apresenta-se o índice de risco para o usuário, de forma que não necessite de um estudo prévio sobre o aplicativo ou sobre técnicas utilizadas para análise de risco. Posteriormente, o próprio usuário decide se quer manter o app instalado em seu aparelho.

Após validar, teoricamente, o modelo final partiu-se para a implementação da aplicação, tendo como meta três princípios, conforme definido por Pechansky (2011): usabilidade, confiabilidade e eficiência. Para facilitar a usabilidade do sistema, o aplicativo foi projetado de forma a minimizar o caminho percorrido pelo usuário para executar uma operação. O software recebeu o nome de Security Dog (figura 4), foi construído na linguagem Java e com interface gráfica em XML. A IDE escolhida foi o Android Studio.



(a) **(b)** **(c)**

Figura 4: (a) Tela inicial do aplicativo Security Dog, (b) Lista de aplicativos analisados e (c) Informações do aplicativo Security Dog

O processo de informar ao usuário o risco de cada aplicativo instalado no dispositivo móvel é executado em uma thread principal, com um alto grau de prioridade, o que possibilita sua auto execução pelo sistema. O processo em questão foi dividido em 3 telas: Lista de aplicativos, informações sobre cálculo de risco e tela com as permissões.

A Lista de aplicativos, seu risco e a explicação de como o risco é calculado, são colocados lado a lado com ajuda de um TabHost. O Risco é listado em uma ativityList, que está inserido na primeira aba da tela, como mostra a figura 4(b). Para apresentar a lista de aplicativos com os seus respectivos riscos seguiu-se os seguintes passos:

- Identificação dos aplicativos instalados;
- Abertura de seu arquivo e identificação de suas permissões;
- Atribuição do risco;
- Listar na tela;
- Salvar análise no banco de dados.

Para identificação dos aplicativos instalados foi utilizada a API do Android PackageManager, que é um gerenciador de pacotes com limitações para aparelhos que não possuem acesso root. O passo seguinte foi a identificação das permissões solicitadas pelo aplicativo. Com a API PackageInfo, foi construída uma função para identificar as permissões de cada aplicativo e as informações do APK, como: ícone, permissões, nome e versão. Para calcular a porcentagem foi utilizado o maior risco possível e o risco atual do aplicativo. A fim de facilitar uma análise intuitiva, foi inserida a avaliação qualitativa, que tem como função identificar o risco como: Baixo, Médio e Alto, tal avaliação pode ser observada na figura 4(b).

Com os aplicativos encontrados e seu risco calculado, as informações serão apresentadas para o usuário em uma lista ListActivity. Após a criação da tela com os dados para o usuário, os resultados são salvos em um banco de dados SQLite nativo do Android, para que isso fosse possível, foi criada uma tabela no banco, com isso não será necessário recalcular os riscos dos aplicativos se os mesmos não sofrerem nenhuma alteração. Para um maior entendimento do usuário, o cálculo de risco será apresentado para ele em uma segunda tela com função explicativa, acessível na segunda aba do TabHost da tela da lista de aplicativos analisados, como ilustra a figura 4(c). Para

aumentar a segurança e a comodidade do usuário, este aplicativo realizará um processo de análise em segundo plano chamado de *"service"*, evitando que o usuário tenha que abri-lo periodicamente para conferir o risco de seus novos aplicativos. Esse processo é iniciado e reiniciado junto com o Android e roda em uma segunda thread que faz análises silenciosas, mesmo com o aplicativo não executado em primeiro plano. Caso a análise em segundo plano encontre algum aplicativo com um alto risco, e que ainda não tenha sido analisado, um alerta será gerado ao usuário. O risco é calculado somente naqueles aplicativos que ainda não foram analisados anteriormente, para isso, é consultado o banco de dados que contém o histórico de análise.

A validação final da aplicação se deu através de um conjunto de testes, para tal, foram levantadas 5 ferramentas da Google Play (Pou, Facebook, Jogo Snake, Hungry Snake e Gmail). Além disso, foram elencados 5 *malware*s descobertos na plataforma Android (DroidDream,Walkinwat/Pirater, RootSmart, Zeahache e BgServ) a fim de fazer uma comparação. Os testes realizados foram:

- Testes Funcionais, com o objetivo de assegurar o correto funcionamento dos recursos oferecidos pelo software e o processamento dos dados de entrada, incluindo a navegação, fluxo dos casos de uso e resultados. O teste foi repetido para as quatro principais funções: Calcular o risco, listar permissões, abrir fórum e gerenciador de aplicativo. O resultado desses testes nos permitiu observar que todo o sistema funciona corretamente, com exceção do fórum que possui limitações. pois o processo depende da Internet e, que o aplicativo baixado na Google Play não seja a versão mais recente, isso se justifica devido ao fato que quando o aplicativo é atualizado o nome do pacote muda, este mesmo nome é utilizado para acessar o fórum e o caminho de acesso acaba sendo perdido.
- Testes de Interface, realizados com o intuito de garantir uma maior e melhor interação do usuário com o aplicativo, visando garantir um sistema interativo e que opere de maneira eficiente, sem necessidade de preparo prévio do utilizador. Para alcançar os objetivos do teste de usabilidade, as funções testadas anteriormente na avaliação de funcionalidade foram repetidas por usuários que não tiveram contato com o software. Foram repassados os objetivos de cada função e foram avaliados os seguintes tópicos:
- Número de erros para usar cada função;
- Número de buscas de ajuda, por função;
- Se o resultado obtido condiz com o esperado.

O indício de satisfação em relação ao uso do sistema foi avaliado com um questionário respondido pelos usuários, eles atribuíram uma nota de 0 a 5 para três tópicos: facilidade em alcançar os objetivos, satisfação em utilizar o programa e confiança nos resultados. Como resultados foram observados as notas dadas pelos utilizadores para cada função do sistema, as quais foram consideradas satisfatórias, nesta etapa do estudo.

Considerações Finais

O modelo de análise de risco construído ao longo desta pesquisa apresentou bons resultados nos testes realizados, apresentou boa eficiência para o cenário em que vai ser empregado, porque não tem cálculos complexos. Além da análise de riscos, o aplicativo disponibiliza outras funcionalidades, que podem auxiliar o usuário a

gerenciar seus aplicativos. Durante a realização dos testes da ferramenta e diante de seus resultados, ficou clara a importância do estudo e de sua continuação, pois um bom modelo de análise de risco deve estar sempre em evolução. Para que a constante adequação e evolução do aplicativo ocorram, foram pensadas técnicas para recolher os dados das análises com o objetivo de melhorar a qualidade do modelo de análise de riscos.

O modelo tem grande relevância, por apresentar os dados de forma simples e clara para o usuário, sem a necessidade de um estudo prévio sobre os riscos. Com a validação do aplicativo comprovou-se a possibilidade de construir um aplicativo para análise de risco em dispositivos moveis. Portanto os objetivos traçados nesse trabalho foram alcançados de maneira positiva.

Referências

CHIN, E; FELTRO, AP; SEKAR, V; WAGNER, D. **Medir a confiança do usuário em Smartphone Segurança e Privacidade.** Proc. Oitava Symp. Utilizável Privacidade e Segurança. pp 1-16, 2012.

DIEDERICH, A; HEALY, Jr Busemeyer Af; PROCTOR, Rw. **Psicologia Experimental**.pp 295-319, John Wiley & Sons, 2013.

FELT, AP; HA, E; EGELMAN, S; HANEY, A; CHIN, E; WAGNER, D. **Permissões do Android: a atenção do usuário, compreensão e comportamento.** Proc. Oitava Symp. Utilizável de Privacidade e Segurança, 2012.

FINUCANE, Ml; ALHAKAMI, A; SLOVIC, P; JOHNSON, Sm. **O Afeto Heurística em Acórdãos Riscos e Benefícios. J. Tomada de Decisão Comportamental** , vol. 13, n°. 1, pp.1 -17 2000.

GATES, Christopher; CHEN, Jing; LI Ninghui. **Effective Risk Communication for Android Apps.** In: Dependable and Secure Computing, IEEE Transactions on, Issue Date: May-June 2014,

KAHNEMAN, D; FAST, **Pensamento**; SLOW. Farrar, Straus and Giroux, 2011.

MUNIZ, Joseph; LAKHANI, Aamir. **Web Penetration Testing with Kali Linux, Packt Publishing**, 2013.

PECHANSKY, Rubem. **Um modelo baseado em princípios de usabilidade para aplicação em interfaces de usuário para a interação humano-computador**. Dissertação submetida a Universidade Federal do Rio Grande do Sul, No ano de 2011.

PORTÕES, Christopher S; LI, Ninghui; PENG, Hao; SARMA, Bhaskar; QI, Yuan; POTHARAJU, Rahul; NITA-ROTARU, Cristina; MOLLOY, Ian. **Generating Summary Risk Scores for Mobile Applications**. In: Communications Surveys & Tutorials. IEEE, 2014.

SCHULTZ, EE K; PL Vu; RW Proctor. **Manual de Fatores Humanos em Web Design**. pp 663-677, 2011, CRC Press.

STATCOUNTER. Disponível em: <https://statcounter.com/> Acessado em janeiro de 2014.

SYMATEC. Disponível em http://www.symantec.com/security_response /landing/spam/. Acessado em janeiro de 2014.

TONIN, Graziela Simone. Tendências em computação móvel. 3 p. Universidade de São Paulo – USP. São Paulo, 2012. Disponível em: <grenoble.ime.usp.br/~gold/cursos/2012/movel/mono-1st/2305-1_Graziela.pdf>. Acessado em março de 2013.

WANG, Wei; WANG, Xing; DAWEI, Feng; JIQIANG, Liu; ZHEN, Han; XIANGLIANG, Zhang. Exploring Permission-Induced Risk in Android Applications for Malicious Application Detection In: Information Forensics and Security, IEEE Transactions on, Issue Date: Nov. 2014.