Análise das Vulnerabilidades de Segurança em Aplicativos Android

André C. A. Cavalcante[1], **Levino M. Porto**[1]

[1]Faculdade do Gama – Universidade de Brasília (UNB)
Caixa Postal 72.444-210 – Gama – DF – Brasil

{andrelink14,levinomoises}@gmail.com

Resumo. *Atualmente o uso de smartphones está crescendo em uma taxa exponencial. As pessoas estão utilizando smartphones para executar tarefas rotineiras como interação social e acesso a internet. Com isto, várias informações particulares do usuário ficam disponíveis na memória do aparelho. Vários malwares foram desenvolvidos para acessar estas informações e/ou avariar o hardware do smartphone. Neste contexto, o objetivo deste trabalho é identificar as principais vulnerabilidades utilizadas pelos malwares. Além disso, este artigo descreve as principais técnicas de segurança que podem ser aplicadas para o desenvolvimento de funcionalidades que protejam as informações presentes em dispositivos móveis. Para isto, uma revisão sistemática acerca das vulnerabilidades do Android OS bem como das técnicas para evitá-las foi realizada.*

Abstract. *Nowadays the use of smartphones is growing at an exponential rate. People are using smartphones to perform routine tasks like social interaction and access to internet. With this, several private user information are available in the device memory. Several malware were developed to access this information and/or hardware malfunctions smartphone. In this context, the aim of this work is to identify the main vulnerabilities used by malware. Additionally, this article describes the key security techniques that can be applied to the development of features that protect the information present on mobile devices. For this, a systematic review of the vulnerabilities of the Android OS as well as techniques to avoid them was performed.*

1. Introdução

Nos últimos anos, a sociedade tem presenciado o rápido aumento na utilização de *smartphones* em seu cotidiano. A evolução dos *smartphones* é constante, sempre oferecendo melhorias no desempenho, redes de acesso e integração com os usuário por meio de sensores [Suarez-Tangil et al. 2014]. Atualmente existem diversas opções de sistemas operacionais para *smartphones*, entre os principais destacam-se o Android OS[1] e o IOS[2]. No terceiro quadrimestre de 2013 o IDC[3] mostrou que dispositivos com Android OS representam aproximadamente 80% das vendas globais de *smartphones*, enquanto que os com IOS representam aproximadamente 13%.

[1]Site oficial do Android OS: $http://www.android.com/$
[2]Site Oficial do IOS: $https://www.apple.com/br/ios/$
[3]Corporação de Dados Internacional $http://www.idc.com/$

Os *smartphones* proporcionam recursos ao usuário que superam uma simples ligação ou o envio de mensagens. Atividades frequentes como acesso a internet, comunicação com redes sociais e interação com conteúdos multimídia são oferecidas de forma intuitiva e prática pelos *smartphones*. Com a popularidade dos *smartphones* em ascensão, ataques maliciosos visando obter informações pertinentes ao usuário ou depreciar o dispositivo ocorrem com frequência [Banuri et al. 2012]. O Android OS é mais sensível a este tipo de ataque devido a fatores como domínio do mercado e distribuição de aplicativos através de repositórios não oficiais.

Neste contexto, o objetivo deste trabalho é identificar as vulnerabilidades presentes em aplicações Android que possam afetar a segurança das informações do usuário. As perguntas formuladas são: Quais são as principais vulnerabilidades presentes em aplicações Android? Quais as técnicas de seguranças utilizadas para mitigar as vulnerabilidades encontradas? O problema considerado é: como o tema é novo, diversas vulnerabilidades estão sendo exploradas, técnicas para evitá-las não acompanham essa curva de crescimento. Para alcançar estes objetivos será utilizada a técnica de revisão sistemática. O processo da revisão consiste das seguintes fases: planejamento da revisão (desenvolvimento do protocolo de revisão e sua validação), execução da revisão (identificação de estudos relevantes, seleção de estudos, avaliação da qualidade, extração e sintetização de informações) e avaliação dos resultados obtidos.

A aplicação da revisão sistemática servirá como insumo para a criação de um catálogo contendo as principais vulnerabilidades de segurança nas aplicações Android bem como possíveis formas de mitigá-las. Este catálogo servirá de apoio para o desenvolvimento de aplicações baseadas na plataforma Android.

Este trabalho está dividido em seções. Apresenta-se na Seção 2 uma contextualização sobre segurança na plataforma Android. Já na Seção 3, o processo de revisão sistemática utilizado é descrito. A Seção 4 consiste em apresentar aspectos relacionados ao planejamento realizado para a revisão sistemática. Logo a seguir, a Seção 5 apresenta aspectos relacionados com a condução da revisão sistemática. Na Seção 6, os resultados obtidos com a revisão são analisados. Para finalizar, na Seção 7 são apresentados as considerações finais e trabalhos futuros.

2. Segurança no Android OS

Android é um sistema operacional open-source amplamente difundido para dispositivos móveis que provê um sistema operacional base, uma camada de aplicações, um conjunto de ferramentas de desenvolvimento Java e um conjunto de aplicações para o sistema[Enck et al. 2009]. As aplicações desenvolvidas por terceiros são condicionadas as restrições da API fornecida pelo sistema. Essas aplicações são executadas na camada superior que é executada em Java. Tais aplicações possuem resumidamente 4 tipos de compomentes: Activity, Service, Content provider e Broadcast Receiver.

O sistema protege as aplicações e informações através da combinação de dois principais mecanismos, a isolação das aplicações na camada de sistema e uma mediação da comunicação entre componentes na camada do meio. Essas medidas restringem a exploração de vulnerabilidades apenas nas aplicações que possuem falhas, restringindo a propagação para o resto do sistema. Outra característica de segurança oriunda dessas medidas é o controle de acesso aos componentes de aplicações. Esse controle é feito

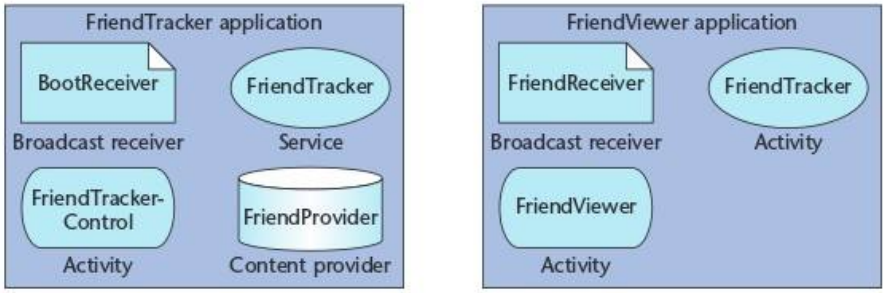

Figure 1. Exemplo de aplicação Android.
Extraído de [Enck et al. 2009]

através da definição de labels feita pelo desenvolvedor da aplicação. O gerenciamento dessas labels é feita pelo monitor de controle de comunicação presente na camada do do sistema. Essa restrição de acesso e controle de labels fornece segurança de acesso para os dados presentes nas aplicações.

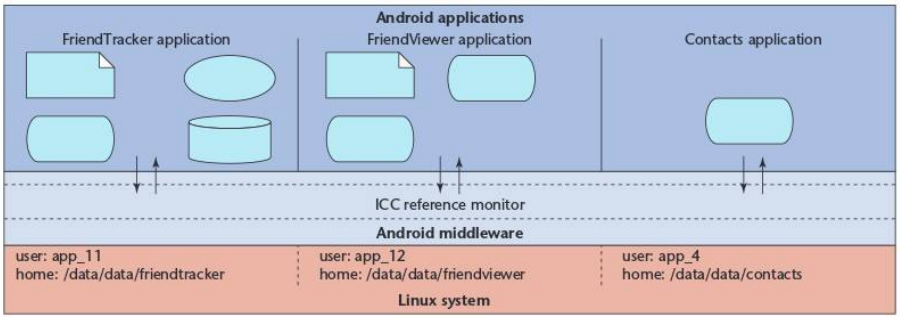

Figure 2. Proteção: isolamento de aplicações e Monitor de Referência.
Extraído de [Enck et al. 2009]

A busca por melhorias na segurança no sistema causou diversas refinamentos no sistema. Tais refinamentos ocorreram e ocorrem a cada nova versão do Android. Alguns desses refinamentos gerados até a versão v1.0r1 do System Development Kit, SDK: Compomentes públicos e privados, Componentes implicitamente abertos, Provedor de conteúdo de informações, Ganchos do sistema, API's protegidas, Leveis de controle de permissão, Intenções pendentes, Permissões de URI e Permissões de intenção em broadcast[Enck et al. 2009]. As definições de segurança de uma aplicação são em grande parte feitas pelo desenvolvedor da aplicação através das definições de acesso das labels. É necessário que se tenha conhecimento da maneira correta de se utilizar os refinamentos feitos no sistema, caso contrário, vulnerabilidades se tornaram presentes.

3. Revisão Sistemática

Segundo [Sampaio and Mancini 2007], uma revisão sistemática representa um tipo de estudo e é uma forma de pesquisa que utiliza como fonte de dados a literatura acerca de determinado tema. Este tipo de investigação concede um resumo das evidências relacionadas a uma estratégia de intervenção específica, mediante a aplicação de métodos explícitos e sistematizados de busca, apreciação crítica e síntese da informação selecionada. As revisões sistemáticas são particularmente viáveis para agregar as informações

de um grupo de estudos realizados separadamente sobre determinado contexto, que podem apresentar resultados conflitantes e/ou coincidentes, bem como encontrar temas que necessitam de evidência, auxiliando no interesse para investigações futuras.

Com base em [Brereton et al. 2007], os procedimentos para realizar uma revisão sistemática no contexto da engenharia de software são: Planejamento da revisão, onde ocorrer a elicitação das necessidades da revisão bem como a definição do protocolo de pesquisa; Execução da Revisão, onde que o plano da revisão previamente definido é colocado em prática. Nesta etapa é realizada a seleção dos estudos, a extração e a síntese dos resultados; e Análise dos resultados, onde que cada leitor poderá criticar ou replicar a revisão sistemática.

O objetivo primordial desse artigo de revisão sistemática é identificar quais são as principais vulnerabilidades presentes em aplicações Android que possam afetar a segurança de informações utilizadas, para posteriormente fornecer um catálogo com a relação de vulnerabilidades e técnicas de segurança que podem ser aplicas para a mitigação dessas falhas.

4. Planejamento da Revisão Sistemática

Nesta seção foi elaborado o protocolo de pesquisa que contem os objetivos, as lacunas a serem resolvidas, as estratégias de busca, os critérios para a seleção de artigos e como os dados serão extraídos das fontes obtidas. Através do protocolo definido, outros pesquisadores poderão replicar esta pesquisa, portanto obterão resultados semelhantes. Para auxiliar o processo de revisão, a ferramenta Zootero foi utilizada. O Zootero é uma ferramenta de pesquisa que permite o gerenciamento de referências.

As fontes de pesquisa selecionadas foram às bibliotecas digitais IEEEXPLORE, SCOPUS e ACM. As linguagens escolhidas foram o inglês e o português. Foram considerados apenas artigos decorrentes de *journals* e *magazines* relevantes para a execução da revisão sistemática.

Para realizar a seleção dos resultados, alguns critérios de inclusão e exclusão foram estabelecidos. Os trabalhos encontrados devem possuir, no mínimo, pontuação B4 no QUALIS da CAPES e pertencer a revistas relevantes; Os trabalhos devem estar disponíveis sem custos de aquisição para alunos da Universidade de Brasília; Os trabalhos devem estar em português ou inglês; Os trabalhos devem possuir conteúdo relacionado a segurança do sistema operacional Android.

5. Condução da Revisão Sistemática

Nesta seção, o plano de revisão sistemática definido na seção 4 será colocado em prática. Todos os procedimentos definidos anteriormente serão estritamente seguidos. Assim o experimento sofrerá o mínimo possível com variáveis imprevistas que podem, de certa forma, modificar o resultado final da pesquisa científica.

Para a definição da string de busca as seguintes palavras chaves foram utilizadas *android, security, software, application, vulnerability, leak, app, malware e virus*. Apresenta-se na figura 3 a string de busca definida com base nas palavras chaves. A string foi utilizada nas fontes de informações previamente selecionadas. Como a engine de busca de cada fonte pode ser diferente, pequenas mudanças na string podem ser necessárias. Porém, a semântica da string deve ser mantida.

(android and security and (software or application) and (application vulnerability or application leak))

Figure 3. *String* de Busca

As buscas foram realizadas nas bases estabelecidas na seção 4. Apresenta-se na Tabela 1 os resultados obtidos com a *string* de busca definida e com os critérios de seleção estabelecidos.

Table 1: Resultados obtidos com as buscas.

Fonte	Resultados	Resultados após filtragem
ACM	41	11
IEEEXPLORE	32	9
SCOPUS	10	5

6. Vulnerabilidades do Android OS

As principais falhas de segurança encontradas em aplicações Android estão relacionadas as permissões de uso de serviços do dispositivo [Chandramohan and Tan 2012]. As permissões são claramente informadas para o usuário durante a fase de instalação, mas comumente elas passam despercebidas. Em alguns casos aplicação pode possuir serviços de componentes não autorizados o usuário, a detecção dessa tentativa pode ser feito através do Kirin, uma aplicação Android para certificação de permissões [Enck et al. 2009].

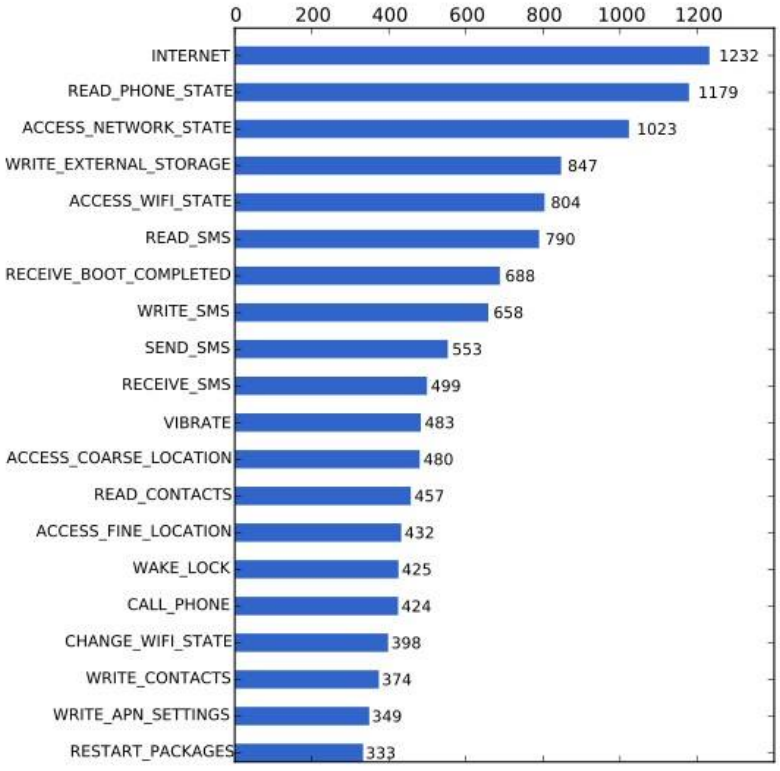

Figure 4. *Top* 20 requisitadas por *malwares*
Extraída de [Zhou and Jiang 2012].

A requisição de permissões é uma das principais maneiras para extração de dados críticos do usuário. A grande maioria dos malwares aproveita a desatenção e comportamento automático do usuário para pedir novas permissões durante o processo de atualização do aplicativo para uma nova versão. A grande quantidade de requisições para leitura e envio de SMS se deve ao fato de uma possível brecha para assinatura de serviços pagos [Zhou and Jiang 2012].

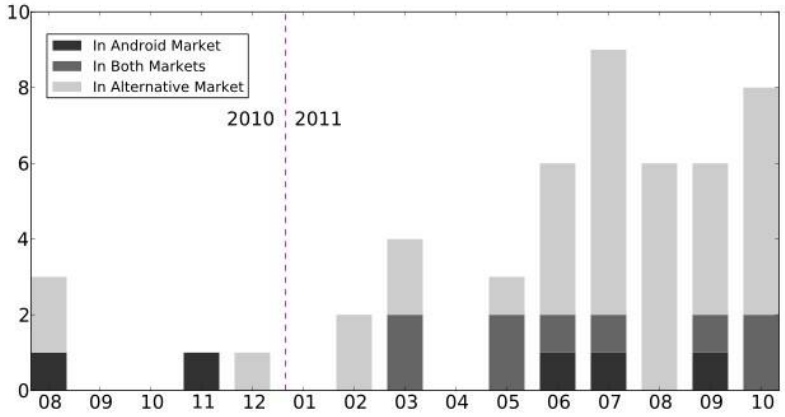

Figure 5. Variação de novas família de *malwares*
Extraída de [Zhou and Jiang 2012].

Outra vulnerabilidade decorrente da desatenção no processo de instalação e liberação de diversas permissões para o aplicativo é o ataque *Repacking*. Nesse tipo de ataque o aplicativo original é modificado e nele são adicionados serviços que farão com que informações do usuário sejam extraídas [Zhou and Jiang 2012]. Após essa modificação o aplicativo alterado é disponibilizado em lojas e baixados por usuários desatentos. Esse tipo de ataque

Diferente do *Repacking* o ataque *Update attack* não é um *malware* no primeiro momento, ele é apenas uma aplicação normal, sem possuir características que possam ser definidas como ofensivas ao usuário[Zhou and Jiang 2012]. No entanto esse ataque faz com que após a atualização do aplicativo, podendo ser visível ou não para o usuário, ele se torne um *malware* com um conjunto de serviços que extraiam informações.

Já o ataque *Drive-by Download* não vai possuir nenhuma característica de *malware* em momento algum do seu ciclo de vida. Ele irá redirecionar o usuário para uma página maliciosa, possivelmente uma falsa loja de aplicativos, que oferecerá soluções para algum tipo de problema analisado pelo aplicativo anterior, caso o usuário instale essa nova aplicação o sistema se tornará vulnerável a extração de dados[Zhou and Jiang 2012].

Uma das vulnerabilidades encontradas pode explorar a extração de informações pessoais do usuário através da utilização da câmera e microfone. Esse tipo de ataque tem como objetivo a obtenção de informações privadas do usuário. Através da utilização da câmera e microfone o atacante pode obter informações sem que o usuário perceba qualquer rastro de invasão ao seus dispositivo.

Alguns tipos de ataque baseados em video utilizam como brecha para coletar dados o período em que o usuário digita textos, possivelmente senhas, no teclado. Essa análise pode ser feita através da captura da movimentação dos olhos[Wu et al. 2014]. Tal

tipo de ataque ocorre quando um usuário instala um aplicativo que parece ser inofensivo, mas em um momento de desatenção durante a instalação ele não percebe que uma requisição de uso de câmera e, ou microfone. Após essa instalação o aplicativo rodará um service em background que ira capturar fotos e gravar áudios sem que seja notado.

Outra maneira para se obter esse tipo de informação é através da captura da tela do usuário durante a digitação de textos. O malware pode ser configurado para que se inicie no momento em que um texto é digitado, isso sem que o usuário receba qualquer notificação [Maggi et al. 2011].

7. Técnicas de Segurança

Atualmente, existem softwares destinados apenas a detecção de malwares, entretanto, eles não são capazes de identificar completamente os softwares maliciosos. Na melhor das hipóteses apenas 86 % deles são detectados[Zhou and Jiang 2012]. Ações preventivas são as medidas mais simples e baratas de serem executadas. Algumas destas são [Chandramohan and Tan 2012]:

- Baixar todas as aplicações de fontes seguras. Utilizar serviços oferecidos ou avaliados pela Google;
- Evitar baixar qualquer aplicativo de uma fonte não certificada;
- Antes de efetuar o download de um aplicativo, veja dados de avaliação do autor. Esta informação é disponibilizada pela comunidade que realiza o *download* da aplicação;
- Durante a instalação de um aplicativo, sempre verifique as requisições que estão sendo realizadas;
- Desligue o WIFI, Bluetooth e outros meios de comunicação caso não estejam em uso. Isto pode evitar o envio de dados pessoais para lugares não seguros;
- Não se conecte a redes abertas. Isto pode ser uma armadilha para obter acesso aos dados do dispositivo móvel;
- Sempre mantenha as aplicações atualizadas. Esta prática é importante, pois os fornecedores do aplicativo sempre efetuam correções e melhorias que podem aumentar a segurança das informações;
- Não permita que informações confidenciais sejam salvas em *cache*. A memória *cache* pode ser acessada facilmente por usuários não autorizados;
- Não clique em *links* suspeitos, na dúvida, sempre digite diretamente a URL do site;
- Sempre monitore o consumo de bateria, SMS e rede do seu celular, caso ele apresente um comportamento incomum, verifique os últimos aplicativos instalados, é possível que o dispositivo esteja sob ataque;
- Utilize um aplicativo seguro para controle remoto do dispositivo, esse aplicativo deve ser utilizado para apagar todos os dados caso o dispositivo tenha sido roubado ou perdido;

Trazer o usuário do dispositivo para o ecossistema de segurança pode ser a chave para a redução drástica na quantidade de ataques bem sucedidos de malwares [Chandramohan and Tan 2012].

Fora do escopo da prevenção, existem as técnicas para a detecção de malwares, que aproveitam das vulnerabilidades existentes. As principais técnicas para esse tipo

de ação são: Análise estática, Análise dinâmica, Análise de permissão de aplicação, Detecção baseada em nuvem e Monitoramento do consumo da bateria. A análise estática em aplicações Android consistem em descompilar a aplicação para se obter o código fonte, análiser o código fonte através de alguma ferramenta e detectar possíveis anomalias nas restrições de acesso. A análise dinâmica envolve simular e isolar o ambiente Android no qual a aplicação será executada. Através dessa simulação e isolamento é possível identificar o seu real comportamento. O controle de permissões de aplicação é a chave principal no controle de acesso aos dados do usuário [Chandramohan and Tan 2012]. Esse tipo de análise só surte efeitos positivos caso seja executada em paralelo com alguma outra técnica, por exemplo, análise estática. As duas últimas técnicas recaem sobre as características de hardware de dispositivos móveis, capacidade de processamento e consumo de energia. A detecção baseada em nuvem transfere a simulação e análise de aplicações para emuladores presentes em servidores. A análise através do consumo de bateria tem como objetivo verificar a variação do gasto energético de aplicações suspeitas e confiáveis, caso haja diferenças exorbitantes é possível que uma delas seja maliciosa.

8. Considerações Finais

O sistema mais utilizado em *smartphones*, Android, também é um dos maiores alvos de tentativas de quebra de segurança. As maneiras utilizadas para se tentar obter informações dos usuários, que já usam os dispositivos para diversos tipos de transações, desde acesso a redes sociais até pagamento de contas bancárias, faz com que os esforços aplicados para a manutenção do sigilo desses dados sejam maiores. Foi relatado diversos tipos de vulnerabilidades presentes no sistema, mas ainda assim o usuário é o maior responsável por manter em segurança o conteúdo presente em seu *smartphone*. A grande maioria das falhas relatadas só são criadas através da liberação desatenta de permissões, sendo esse controle feito pelo usuário.

Desenvolvedores também podem auxiliar no processo para proteção dos dados, porém é necessário conhecimento técnico aprofundado. Técnicas de segurança normalmente não são triviais, mas as mais básicas como restrição de acesso de serviços da aplicação podem ser executadas de uma maneira rápida e garantem proteção dos dados utilizados, através do bloqueio do seu conteúdo.

Através das boas práticas identificadas, é possível reduzir de forma considerável a proliferação de *malwares*. Apesar das boas práticas serem válidas, a evolução dos *malwares* é constante, sendo assim é necessário validar e identificar novas técnicas de segurança para evitar novos tipos de ameaças.

References

Banuri, H., Alam, M., Khan, S., Manzoor, J., Ali, B., Khan, Y., Yaseen, M., Tahir, M. N., Ali, T., Alam, Q., and Zhang, X. (2012). An android runtime security policy enforcement framework. *Personal Ubiquitous Comput.*, 16(6):631–641.

Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., and Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, 80(4):571 583. Software Performance 5th International Workshop on Software and Performance.

Chandramohan, M. and Tan, H. B. K. (2012). Detection of mobile malware in the wild. *Computer*, 45(9):65–71.

Enck, W., Ongtang, M., and McDaniel, P. D. (2009). Understanding android security. *IEEE Security and Privacy*.

Maggi, F., Volpatto, A., Gasparini, S., Boracchi, G., and Zanero, S. (2011). A fast eavesdropping attack against touchscreens. In *Information Assurance and Security (IAS), 2011 7th International Conference on*, pages 320–325.

Sampaio, R. F. and Mancini, M. C. (2007). Systematic review studies: a guide for careful synthesis of the scientific evidence. 11(1):83 89.

Suarez-Tangil, G., Tapiador, J. E., Peris-Lopez, P., and Blasco, J. (2014). Dendroid: A text mining approach to analyzing and classifying code structures in android malware families. *Expert Syst. Appl.*, 41(4):1104–1117.

Wu, L., Du, X., and Fu, X. (2014). Security threats to mobile multimedia applications: Camera based attacks on mobile phones. *Communications Magazine, IEEE*, 52(3).

Zhou, Y. and Jiang, X. (2012). Dissecting android malware: Characterization and evolution. In *Security and Privacy SP, 2012 IEEE Symposium on*, pages 95–109.