Avaliação de Segurança em Aplicações Bancárias na Plataforma Android

Diego de Souza Maia[1], **Luciano Ignaczak**[1]

[1]Universidade do Vale do Rio dos Sinos (UNISINOS) – São Leopoldo, RS – Brazil

diegosmaia@gmail.com,lignaczak@unisinos.br

Abstract. *The mobile banking applications provide more and more convenience for the public. However, its adoption and use still faces distrust from their users, particularly with regard to information security. This work constitutes a security analysis to identify characteristics that could compromise the confidentiality of the information stored in the internal memory in RAM or in the source code of banking applications developed for the Android platform, the five largest retail banks operating in the Brazilian market. The results showed that all applications have characteristics that can be exploited by threats.*

Resumo. *Os aplicativos de mobile banking proporcionam cada vez mais conveniência para à população. No entanto, sua adoção enfrenta desconfiança por parte de seus usuários, principalmente no que diz respeito a segurança da informação. Este trabalho realizou uma análise de segurança para identificar características que pudessem comprometer a confidencialidade das informações armazenadas na memória interna, na memória RAM ou no código-fonte de aplicações bancárias desenvolvidas para a plataforma Android, dos cinco maiores bancos que operam no mercado brasileiro. Os resultados mostraram que todos os aplicativos possuem características que podem ser exploradas por ameaças.*

1. Introdução

Nos últimos anos, a evolução dos dispositivos móveis e a mudança nos hábitos dos usuários impulsionaram as vendas de *smartphones* e *tablets* ao redor do mundo. Visando atender novas demandas, as aplicações bancárias para dispositivos móveis, conhecidas como *mobile banking* ou m-banking, estabelecem um canal de relacionamento entre bancos e clientes através de um dispositivo móvel conectado à internet, permitindo a realização de consultas, pagamentos e outros serviços financeiros [Ferris et al. 2014].

Conforme informações divulgadas pelo [FED 2016], 77% da população americana possui um smartphone, 53% dessa fatia já acessou sua conta bancária utilizando aplicativos m-banking. No Brasil, o número de contas habilitadas para o uso do *mobile banking* em 2011 chegava a 2 milhões, 1,1% do total de contas ativas na época. Já em 2015, esse número passou para 33 milhões, 22% do total de contas ativas, passando a ser um dos principais canais de relacionamento entre bancos e clientes, juntamente com o *internet banking*, os quais somados, abrangem 54% do total das transações bancárias realizadas [FEBRABAN 2015].

De acordo com o [IDC 2015], o sistema operacional Android lidera o mercado de dispositivos móveis com aproximadamente 82.8% de participação em relação aos demais

sistemas. Por ser a plataforma mais utilizada isso também acaba contribuindo para que ela se torne o principal alvo de ataques maliciosos. Em 2013, por exemplo, 97% das ameaças registradas em sistemas móveis foram direcionadas para o sistema Android. No final do ano de 2014, foram identificados 2,3 milhões de ataques direcionados para plataforma *mobile banking* em dispositivos Android. A Rússia foi o país mais atacado e lidera o *ranking* com um percentual de 29,97%, seguido pelo Brasil com 6,5% e pela Turquia com 5,74% [SYMANTEC 2014, KASPERSKY 2015].

O autor em [UNUCHEK 2014] faz um retrospecto geral sobre os problemas de segurança enfrentados por aplicações m-banking e afirma que os primeiros *malwares* para dispositivos móveis surgiram em 2011; em 2012 foram aprimorados, seguindo em evolução constante e em ritmo alarmante. Nesse sentido, este trabalho busca responder a seguinte questão: os aplicativos de *mobile banking* disponibilizados pelas instituições bancárias que operam no mercado brasileiro possuem as preocupações de segurança necessárias para proteção das informações pessoais de seus usuários?

Este trabalho tem o objetivo de realizar uma avaliação de segurança em cinco aplicações bancárias desenvolvidas para a plataforma Android, disponibilizados pelas cinco maiores instituições que operam no mercado brasileiro. Para alcançar o objetivo, os aplicativos de *mobile banking* foram submetidos a análises de segurança que buscaram identificar características que pudessem comprometer a confidencialidade das informações armazenadas na memória interna, na memória RAM ou no código-fonte desses aplicativos.

O artigo está dividido em cinco seções. A segunda seção apresenta artigos relacionados ao tema da pesquisa e o posicionamento deste trabalho. A terceira seção apresenta a metodologia utilizada na realização das análises do sistema de arquivos, das informações armazenadas na memória do dispositivo e no código-fonte do aplicativo. Já a quarta seção apresenta os resultados das análises realizadas nos aplicativos de *mobile banking* testados. A quinta seção, relata as considerações e conclusões do autor.

2. Trabalhos Relacionados

Já foram realizados diversos trabalhos com o objetivo de identificar vulnerabilidades de segurança em aplicativos desenvolvidos para o sistema Android. [Chia et al. 2012] identificaram a ineficiência dos controles de permissão de acesso com o consentimento do usuário em aplicativos desenvolvidos para a plataforma Android, que solicitavam acessos mais elevados e privilegiados do que realmente necessitavam. [Onwuzurike and Cristofaro 2015], a partir de uma análise nos 100 aplicativos mais baixados para a plataforma Android que necessitam de acesso a rede, detectaram que quatro deles transmitiam dados confidenciais sem criptografia, trinta e dois aceitavam qualquer tipo de certificado e nome de *host* sem realizar nenhum tipo de validação.

As lojas de aplicativos facilitaram a compra e instalação de programas em dispositivos móveis, em contrapartida as ameaças de segurança em *softwares* também aumentaram, principalmente na plataforma Android. [Mahmood et al. 2012] descreveram técnicas para automação de testes e análises de *Fuzzing* para avaliar a segurança em aplicativos Android, muitas vezes disponibilizados com vulnerabilidades ou códigos maliciosos nas lojas *on-line*. [Wang 2015] também propôs uma plataforma para realização de testes de segurança em aplicativos Android, o método proposto não requer a instalação de

um cliente no dispositivo móvel, o aplicativo é testado em um servidor de testes simulando um ambiente Android.

A busca por vulnerabilidades de segurança em aplicações bancárias na plataforma Android também tem sido alvo de estudos nos últimos anos. [Elkhodr et al. 2012] propuseram a utilização do *Transport Layer Security* (TLS), para garantir a privacidade e integridade na transmissão de dados entre aplicações m-banking em dispositivos móveis e instituições financeiras, complementando assim o controle de acesso por identidade tradicionalmente utilizado. [Cruz and Aranha 2015] analisaram a configuração e troca de informações entre servidores e aplicativos de m-banking de sete bancos que atuam no mercado brasileiro. O estudo realizado por eles mostrou que é possível obter informações financeiras e credenciais de acesso por meio de um ataque de personificação ou através de falhas na configuração dos servidores. Já [Sebastiany et al. 2015] analisaram se o protocolo TLS é utilizado de forma correta em sessenta aplicativos de m-banking, divididos igualmente por três países: Brasil, Estados Unidos e Reino Unido. Os resultados mostraram que 31% dos aplicativos analisados utilizavam certificados digitais não confiáveis e transmitiam as informações do cliente através de uma conexão insegura.

Outros trabalhos também já buscaram identificar vulnerabilidades de segurança em aplicações bancárias na plataforma Android. [Panja et al. 2013] mostraram como os aplicativos m-banking podem ser decodificados, manipulados, reempacotados e distribuídos novamente em lojas de aplicativos alternativas, como sendo um aplicativo original disponibilizado pelo banco, essa alteração no aplicativo possibilitaria capturar informações como *login* e senha do usuário por exemplo. [Filiol and Irolla 2015] desenvolveram três *softwares* (Egide, Panoptes, Tarentula) para realizar análises dinâmicas e estáticas nos arquivos binários e no código-fonte de cinquenta aplicativos m-banking de diversos locais do mundo com o objetivo de identificar vulnerabilidades de segurança que pudessem ser exploradas por pessoas mal intencionadas.

Como apresentado, diversos trabalhos tiveram como objetivo a análise de segurança em aplicativos de m-banking. No entanto, as análises de segurança realizadas por [Sebastiany et al. 2015], focaram na conexão TLS entre o servidor do Banco e o aplicativo. Já [Filiol and Irolla 2015] realizaram análises dinâmicas e estáticas nos arquivos binários dos aplicativos de *mobile banking*. Este trabalho difere dos demais pois analisa a segurança da aplicação em relação aos dados confidenciais manipulados por ela, seja no sistema de arquivos ou na memória do dispositivo onde está sendo executado. O trabalho realizou também uma análise manual para identificar se informações confidenciais estavam armazenadas de forma explícitas no código-fonte dos aplicativos de m-banking.

3. Metodologia

Os aplicativos de *mobile banking* foram submetidos a análises de segurança que buscaram identificar características que pudessem comprometer a confidencialidade das informações armazenadas na memória interna, na memória RAM ou no código-fonte desses aplicativos. As análises foram realizadas em aplicativos de *mobile banking* desenvolvidos para a plataforma Android e disponibilizados na loja de aplicativos Google Play, pelos maiores bancos de varejo que operam no mercado brasileiro.

A escolha das instituições analisadas e seus respectivos aplicativos levou em conta o *ranking* publicado pelo Banco Central do Brasil [BCB 2015]. Com base nes-

sas informações, foram selecionadas os cinco primeiros bancos de varejo que operam no Brasil, considerando o valor total dos ativos. Visando preservar as instituições analisadas, a identificação das mesmas foi ocultada e seus respectivos nomes foram substituídos por Banco A, Banco B, Banco C, Banco D e Banco E durante a demonstração dos resultados.

As análises foram realizadas em um computador com o sistema operacional Linux Ubuntu 14.04 LTS, utilizando o programa Genymotion, versão 2.6, para emular um dispositivo Samsung Galaxy S6, com o sistema operacional Android 5.1, para cada aplicação de *mobile banking* testada, ou seja, cinco dispositivos diferentes isolados virtualmente.

Para realizar a análise do sistema de arquivos o autor utilizou as ferramentas DD e Netcat para gerar imagens das partições *cache*, *data* e *system* de cada dispositivo em três momentos diferentes. A primeira extração ocorreu logo após cada dispositivo ser criado no Genymotion; a segunda após o aplicativo de *mobile banking* ser instalado; e, por fim, a geração da última imagem ocorreu após o aplicativo ser utilizado e finalizado. Ao final de cada uma das três etapas foi extraído de cada imagem uma lista de arquivos presentes no dispositivo e seus respectivos *hashes* MD5 e MAC times, para que fosse possível comparar as modificações ocorridas em cada etapa.

A análise de informações armazenadas na memória do dispositivo tem como objetivo, localizar e identificar informações que comprometessem a segurança ou representassem algum risco aos dados do usuário, como a exposição em texto claro de suas credenciais de acesso, após o encerramento do aplicativo. Para alcançar os objetivos da análise, a extração de informações da memória ocorreu em dois momentos distintos. O primeiro despejo de dados da memória foi realizado no final do acesso a conta bancária no aplicativo de *mobile banking*, mais precisamente após a aplicação ser finalizada. Durante este acesso a conta bancária foram realizadas consultas do extrato da conta e do cartão de crédito, consulta a aplicações e informe de rendimentos para o imposto de renda em cada aplicativo. Já o segundo despejo de dados da memória foi realizado aproximadamente duas horas após o encerramento do aplicativo. Durante este intervalo de tempo, foram executados outros programas presentes no dispositivo virtual, como o navegador de *internet*, a calculadora e o calendário, para simular a utilização do dispositivo.

Por fim, a análise do código-fonte dos aplicativos não buscou validar se o código foi escrito e/ou estruturado de forma segura, mas realizar uma análise estática e manual para revisar o código e identificar vulnerabilidades nos binários do aplicativo que pudessem vir a comprometer a segurança ou expor algum risco aos dados do usuário. Essa etapa, está subdividida em quatro partes: engenharia reversa do aplicativo para visualizar o código-fonte, verificar se o código foi ofuscado, localizar eventuais credenciais *"hard-coded"* e identificar informações de conexão.

4. Resultados

Para facilitar a compreensão dos resultados, as análises foram divididas em subseções de acordo com os testes realizados, onde são apresentadas situações que podem colocar em risco as informações armazenadas pelos aplicativos de *mobile banking*. A aplicação do Banco C demonstrou-se instável durante a tentativa de logon no sistema, chegando a ser finalizada automaticamente de forma inesperada. Por isso, não são apresentados os resultados das análises do Banco C nas seções 4.1 e 4.2, no entanto, a seção 4.3 apresentará resultados pois não é necessário executar o aplicativo para realizar a análise do

código-fonte.

4.1. Análise do sistema de arquivos

As análises indicaram que após a utilização do aplicativo de m-banking do Banco A, foram criados arquivos no formato SQLite na partição *data*, dentro do diretório *data/br.com.BancoA.android/databases*, contendo informações em texto claro como o código da agência, número da conta, nome do cliente, data e hora do último acesso ao aplicativo. Também foram localizadas *strings* contendo informações codificadas no formato Base64, no entanto, ao decodificá-las nenhuma informação relevante foi identificada.

Enquanto isso, as análises realizadas após a utilização dos aplicativo de m-banking dos Bancos B e D identificaram, em seus respectivos diretórios dentro da partição *data*, a criação de arquivos no formato SQLite. No entanto, nenhuma informação relevante ou que pudesse expor algum dado do usuário foi localizada. Arquivos no formato XML contendo *strings* com informações codificadas no formato Base64 também foram encontrados nessa partição, mas nenhuma informação relevante foi encontrada após a decodificação.

Por fim, após a utilização do aplicativo de m-banking do Banco E, foram encontrados arquivos no formato XML dentro do diretório *data/com.BancoE.app/shared_prefs*, na partição *data*, contendo *strings* em texto claro com informações pessoais do usuário como, por exemplo, o número do CPF e também a data e hora da última conexão do aplicativo. Parte dessas informações pode ser visualizadas na Figura 1.

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="cpf">[redacted]</string>
    <string name="segmento">CL</string>
</map>
```

Figura 1. Informações em texto claro armazenadas na memória interna do dispositivo

Os resultados vistos nessa subseção demonstram que os aplicativos de *mobile banking* poderiam utilizar alguma ferramenta para criptografar pastas e arquivos que pudessem expor ou facilitar a identificação das informações armazenados dentro do dispositivo móvel, adicionando uma camada de proteção às informações do usuário e das aplicações armazenadas nessa partição. Caso isso não fosse possível, os aplicativos não deveriam armazenar na memória interna do dispositivo esse tipo de informação [Chell et al. 2015].

4.2. Análise das informações armazenadas na memória

A análise das informações contidas na memória do dispositivo móvel revelaram que quatro dos cinco aplicativos de *mobile banking* analisados, armazenam as credenciais de acesso do usuário em texto claro, sem nenhum tipo de proteção. Observou-se também que em três dos quatro aplicativos analisados, essas informações eram passadas em sequência para a memória do dispositivo. Essa situação pode ser visualizada na Figura 2a, que apresenta o resultado da coleta de informações armazenadas na memória do dispositivo do Banco D após o aplicativo de m-banking ser finalizado. Os caracteres referentes a senha foram ocultados para resguardar as informações do usuário.

Ainda em relação a análise do aplicativo do Banco D, a Figura 2b, mostra o resultado do segundo despejo de dados da memória do dispositivo, realizado duas horas

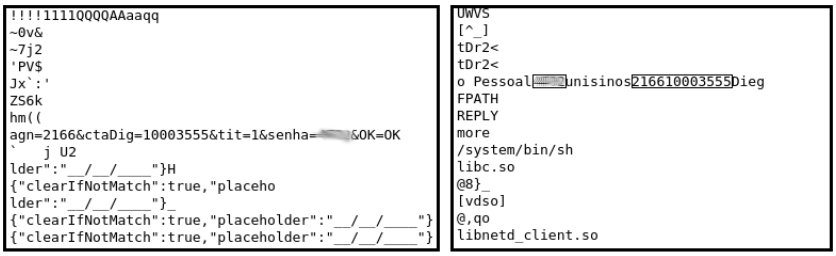

Figura 2. Informações em texto claro armazenadas na memória do dispositivo

após o encerramento do aplicativo. Foi possível identificar que, embora novas posições de memória já estivessem sido ocupadas por outras informações utilizadas durante a simulação de uso do navegador de internet, as credenciais de acesso do usuário ainda estavam armazenadas em texto claro, sem nenhum tipo de proteção.

A presença das credenciais de acesso do usuário na memória do dispositivo, mesmo após o aplicativo de m-banking ser finalizado, mostra que a função *garbage collector* presente no gerenciamento de memória do sistema operacional Android, descrita por [Shahriar et al. 2014], não é utilizada de forma adequada nas aplicações analisadas. Ao serem finalizados, os aplicativos deveriam apagar da memória todos os objetos contendo informações utilizadas pela aplicação, entretanto, isso não é realizado. Logo, pode-se imaginar que em caso de perda ou roubo do dispositivo, um usuário mal intencionado poderia consultar as informações armazenadas na memória e obter acesso a conta bancária do proprietário do dispositivo.

4.3. Análise do código-fonte

A análise do código-fonte revelou que os aplicativos de *mobile banking* dos Bancos A e C não utilizam técnicas de ofuscação de código para dificultar que o atacante compreenda as informações presentes no binário de seus respectivos aplicativos. A falta de ofuscação tornou possível a interpretação de classes, campos, métodos, comandos e configurações do aplicativo durante um ataque de engenharia reversa.

Durante as análises foram encontradas na aplicação do Banco D, informações pré-fixados no código-fonte, que estavam vinculadas a integração com serviços de *Home Broker*. No app do Banco B foi localizada uma classe com *strings* de valores pré-fixados para integração com o Twitter, leitor de QR Code e código de barras. A falta de ofuscação, somada a presença de *strings* contendo informações pré-definidas, pode colocar o cliente e o próprio Banco em situação de risco. Ainda nesse sentido, é possível afirmar que a falta de ofuscação do código facilita ao atacante, manipular, reempacotar e distribuir novamente o aplicativo de m-banking em lojas de aplicativos alternativas para capturar, por exemplo, o login e senha do usuário [Panja et al. 2013].

5. Considerações Finais

Este trabalho realizou uma avaliação de segurança em cinco aplicações bancárias desenvolvidas para a plataforma Android, disponibilizados pelas cinco maiores instituições que operam no mercado brasileiro. Os aplicativos foram submetidos a análises de segurança

com o intuito de identificar características que pudessem comprometer a confidencialidade das informações armazenadas na memória interna, na memória RAM ou no código-fonte destes aplicativos.

Foi possível constatar que dois dos quatro aplicativos analisados, apresentam falhas relacionadas a proteção dos dados pessoais de seus utilizadores, armazenados na memória interna do dispositivo. Essas informações são armazenadas em texto claro, sem nenhum método de criptografia em arquivos no formato SQLite e XML. A análise das informações contidas na memória RAM revelou que, todos os aplicativos analisados armazenam as credenciais de acesso do usuário em texto claro, sem nenhum tipo de proteção. Em três dos quatro aplicativos analisados, essas informações eram passadas em sequência para a memória do dispositivo. Por fim, a análise do código-fonte revelou que dois, dos cinco aplicativos de *mobile banking* testados, não utilizam nenhuma técnica de ofuscação de código. Apesar de não comprometer diretamente a proteção das informações pessoais dos clientes, essa situação torna mais fácil, para um usuário mal intencionado, a compreensão do fluxo de dados realizado pelo aplicativo

Para finalizar, pode-se afirmar que as análises realizadas responderam a pergunta principal do trabalho, pois foi identificado que os aplicativos de *mobile banking* disponibilizados pelas instituições bancárias que operam no mercado brasileiro, não possuem as preocupações de segurança necessárias para proteção das informações pessoais de seus usuários. Propõe-se como trabalhos futuros, a realização de uma avaliação de segurança utilizando análises dinâmicas no código-fonte de aplicações m-banking. Também poderão ser realizadas análises semelhantes em aplicativos de outros segmentos, como por exemplo, comércio eletrônico.

Referências

BCB, B. C. d. B. (2015). Dados selecionados de entidades supervisionadas - if.data. Disponível em: <https://www3.bcb.gov.br/informes/relatorios>. Acesso em: março de 2016.

Chell, D., Erasmus, T., Colley, S., and Whitehouse, O. (2015). *The Mobile Application Hackers Handbook*. Wiley.

Chia, P., Yamamoto, Y., and Asokan, N. (2012). Is this app safe? a large scale study on application permissions and risk signals. In *Proceedings of the 21st international conference on World Wide Web*, pages 311–320. ACM.

Cruz, R. and Aranha, D. (2015). Analise de segurança em aplicativos bancarios na plataforma android. *Simposio Brasileiro em Segurança da Informação e de Sistemas Computacionais - SBSeg*.

Elkhodr, M., Shahrestani, S., and Kourouche, K. (2012). A proposal to improve the security of mobile banking applications. In *ICT and Knowledge Engineering, 2012 10th International Conference on*, pages 260–265.

FEBRABAN (2015). Pesquisa febraban de tecnologia bancária 2015. Disponível em: <http://www.ciab.org.br/Downloads/pesq_2015.pdf?v=1>. Acesso em: agosto de 2016.

FED (2016). Consumers and mobile financial services 2016. Disponível em: <http://www.federalreserve.gov/econresdata/consumers-and-mobile-financial-services-report-201603.pdf>. Acesso em: agosto de 2016.

Ferris, B., Stahle, J., and Baggili, I. (2014). Quantifying the danger of mobile banking applications on the android platform. In *9th Annual Symposium on Information Assurance (ASIA 14)*, page 65.

Filiol, E. and Irolla, P. (2015). (in)security of mobile banking...and of other mobile apps. Paper presented at the Black Hat, Asia.

IDC, I. D. C. (2015). Smartphone os market share, 2015 q2. Disponível em: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>. Acesso em: maio de 2016.

KASPERSKY (2015). Kaspersky lab report: Financial cyberthreats in 2014. Disponível em: <https://securelist.com/files/2015/02/KSN_Financial_Threats_Report_2014_eng.pdf>. Acesso em: maio de 2016.

Mahmood, R., Esfahani, N., Kacem, T., Mirzaei, N., Malek, S., and Stavrou, A. (2012). A whitebox approach for automated security testing of android applications on the cloud. In *Automation of Software Test (AST), 2012 7th International Workshop on*, pages 22–28. IEEE.

Onwuzurike, L. and Cristofaro, E. D. (2015). Danger is my middle name: Experimenting with ssl vulnerabilities in android apps. *arXiv preprint arXiv:1505.00589*.

Panja, B., Fattaleh, D., Mercado, M., Robinson, A., and Meharia, P. (2013). Cybersecurity in banking and financial sector: Security analysis of a mobile banking application. In *Collaboration Technologies and Systems (CTS), 2013 International Conference on*, pages 397–403.

Sebastiany, D., Freitas, M., and Ignaczak, L. (2015). Uma analise dos certificados digitais utilizados nas conexões tls dos aplicativos de mobile banking na plataforma android. *Revista de Empreendedorismo, Inovação e Tecnologia*, 2(1):66–75.

Shahriar, H., North, S., and Mawangi, E. (2014). Testing of memory leak in android applications. In *HighAssurance Systems Engineering $HASE$, 2014 IEEE 15th International Symposium on*, pages 176–183. IEEE.

SYMANTEC (2014). Internet security threat report 2014. Disponível em: <http://www.symantec.com/content/en/us/enterprise/other_resources/b-istr_main_report_v19_21291018.enus.pdf>. Acesso em: março de 2015.

UNUCHEK, R. (2014). One billion more kaspersky lab counts up this years cyber-threats. Disponível em: <http://www.kaspersky.com/about/news/virus/2014/Kaspersky-Lab-counts-up-this-years-cyber-threats>. Acesso em: março de 2015.

Wang, Y. (2015). An automated virtual security testing platform for android mobile apps. In *Mobile and Secure Services (MOBISECSERV), 2015 First Conference on*, pages 1–2. IEEE.