Bibliotecas para Cache em Android: uma Análise na Perspectiva de Segurança

Carlos Ramos[1], **Eduardo Feitosa**[1]

[1]IComp/UFAM, Manaus, Brasil

{carlos,efeitosa}@icomp.ufam.edu.br

Abstract. *This article presents the first study on libraries that implement Cache for the Android platform through an empirical analysis of the characteristics of each library, as well as verification of three factors that can present as they are fragile the security issue. The empiric results suggest that few libraries care and/or implement mechanisms to ensure the reliability of data in Cache.*

Resumo. *Este artigo realiza o primeiro estudo sobre bibliotecas que implementam Cache para a plataforma Android através de uma análise empírica das características de cada biblioteca, assim como a verificação de três fatores capazes de apresentar como elas são frágeis no quesito de segurança. Os resultados empíricos demonstram que poucas bibliotecas se preocupam e/ou implementam mecanismos para garantir a confiabilidade de dados em Cache.*

1. Introdução e Motivação

É fato que os dispositivos móveis tornaram-se os alvos preferidos de atividades maliciosas como spam, códigos maliciosos, *phishing*, participação em *botnets* e, mais recentemente, vazamento de informações. Dentre os fatores que justificam essa preferência pelos dispositivos móveis, o principal é o não uso de práticas seguras de programação associado a não adoção das recomendações de segurança da plataforma [ViaForensics 2011]. Para melhorar o entendimento desse último problema, na plataforma Android, desenvolvedores que fazem uso de Cache para o armazenamento de dados são apenas orientados (não obrigados) a seguir recomendações como a que limita o tamanho da Cache em 1MB [Android 2014, NowSecure 2015].

Neste contexto de não obediência e/ou falta de recomendações sobre o uso de Cache na plataforma Android e pouca ou nenhuma consideração com a segurança, especialmente com a possibilidade de vazamentos de informações, este trabalho objetiva realizar uma análise empírica das bibliotecas para implementação de Cache em aplicações na plataforma Android. Para alcançar esse objetivo, fatores baseados em recomendações de segurança foram empregados para permitir uma correta avaliação.

É importante ressaltar que: (i) embora a segurança de cache seja um assunto bastante estudado nas plataformas tradicionais, até onde os autores deste artigo sabem, este é o primeiro estudo sobre bibliotecas que implementam Cache para a plataforma Android; e (ii) este artigo é parte de uma dissertação de mestrado, cujo objetivo geral é propor um mecanismo de detecção de vazamento de informações em dispositivos móveis na plataforma Android, por meio da análise de Cache das aplicações.

 ©2015 SBC — Soc. Bras. de Computação

2. Cache no Android

A Cache na plataforma Android está dividida em: (i) **Sistema**, que contém o Cache da máquina virtual Dalvik e permite um tempo de execução mais rápido através do alinhamento, verificação e otimização de *byte-codes*; e (ii) **Aplicativos** (app), onde o armazenamento ocorre na memória do dispositivo (Cache interna) e/ou no cartão de memória externo (Cache externa). Quem define o que colocar na Cache de Aplicativos é o desenvolvedor, mas normalmente armazenam-se dados de páginas Web, banco de dados, imagens, objetos serializáveis, entre outros.

O Android possui duas implementações de Cache: LRUCache e DiskLRUCache. A **LRUCache**, parte do Android SDK, é especializada em gerenciar objetos em memória RAM [Android 2015] e utiliza o algoritmo de substituição de páginas LRU (*Least Recently Used*) para manter objetos recentemente referenciados acessíveis e remover os menos utilizados recentemente antes que a Cache exceda o tamanho designado. Já a **DiskLRUCache** é voltada para o gerenciamento de arquivos em disco (SDCard). Ela limita o número de bytes que serão armazenados no sistema de arquivos. Assim, quando o número de bytes armazenados excede o limite, a Cache remove entradas até o limite ser satisfeito.

3. Bibliotecas para Cache Android

Embora as bibliotecas LRUCache e DiskLRUCache forneçam suporte a implementação e uso de Caches, bibliotecas de terceiros tem sido criadas para melhorar/facilitar alguns aspectos da implementação ou atender algum requisito mais complexo. A Tabela 1 apresenta, de forma resumida, as bibliotecas Cache para Android avaliadas.

Tabela 1. Bibliotecas de Cache para Android

Biblioteca	Data		Armazenamento		Características
	Criação	Atual.	Memória	Disco	
Android Easy Cache [Brison 2014]	Mai/2014	Jun/2015	LRUCache	DiskLRUCache	A biblioteca permite armazenar objetos padrão, no formato JSON (*JavaScript Object Notation*), bem como formatos definidos pelo usuário.
Simple DiskCache [Fhucho 2015]	Mar/2013	Abr/2014	Não permite	DiskLRUCache	Facilidade na implementação do DiskLRUCache.
HttpResponse Cache [Andrews 2015]	Jan/2012	Out/2014	Não permite	DiskLRUCache	Fornece *Caching* transparente e automático de requisições HTTP e HTTPS, visando poupar tempo e largura de banda. Suporta as classes *HttpUrlConnection* e *HttpsUrlConnection*.
ObjectCache [Connor 2015]	Fev/2014	Jun/2015	Próprio	DiskLRUCache	Permite especificar o tempo de expiração das entradas de Cache.
Qachee [Jafelle 2015]	Fev/2014	Jan/2015	LRUCache	Não permite	Permite especificar o tempo de expiração das entradas de Cache.
Reservoir [Cowkur 2015]	Dez/2013	Jun/2015	Não permite	DiskLRUCache	Insere, consulta e exclui objetos de forma síncrona ou assíncrona. A limpeza da Cache pode ser total ou por entrada individual.
Android BitmapCache [Banes 2015]	Jul/2012	Nov/2013	LRUCache	DiskLRUCache	É especializada na criação de Cache para uso com objetos Bitmap.
Kinvey [Kinvey 2015]	Não informada	Constante	LRUCache	SQLite 3	Biblioteca proprietária e paga que implementa criptografia e permite armazenar e recuperar arquivos binários de tamanho até 5TB, de qualquer formato.
Expirable DiskLruCache [Rawat 2015]	Mar/2015	Abr/2015	Não permite	DiskLRUCache	Permite a expiração de conteúdo através de um tempo de despejo (*evictionTimeSpan*), bem como criptografar dados usando a biblioteca Conceal [Facebook 2015] do facebook.
Carbonite [Cáceress 2015]	Jul/2013	Set/2014	Próprio (POJO)	DiskLRUCache	Permite especificar como e quanto tempo os dados serão mantidos em Cache.

 ©2015 SBC — Soc. Bras. de Computação

3.1. Discussão

Ao analisar a Tabela 1 percebe-se que quatro bibliotecas - Android Easy Cache, Qachee, Android BitmapCache e Kinvey - utilizam a LRUCache como padrão. Somente a ObjectCache e Carbonite usam implementações próprias. Já na Cache em disco, oito das dez bibliotecas fazem uso da DiskLRUCache. A Qachee não armazena dados em disco e a Kinvey usa SQLite3 para isso. Além disso, todas as bibliotecas listadas, com exceção da Kinvey, são livres e gratuitas. No quesito segurança, apenas duas bibliotecas (Kinvey e Expirable DiskLruCache) fazem uso de criptografia sobre os dados em Cache para aumentar a segurança e garantir a privacidade, confiabilidade e integridade dos dados.

Para resumir, a simples análise da documentação fornecida (ou encontrada) e a verificação do código fonte de cada biblioteca mostra que a maioria delas não está preocupada com segurança, especialmente com a possibilidade do vazamento de informações.

4. Fatores de Avaliação

Até onde os autores deste trabalho têm conhecimento, não existe um conjunto bem definido de métricas destinadas a exprimir os principais fatores a serem considerados durante a avaliação de Cache, especialmente em dispositivos móveis. Assim, buscou-se, como contribuição, estabelecer tal conjunto de fatores (aplicados na forma de um *checklist*) focadas em avaliar a segurança. São elas: (i) **Criptografia dos Dados**, que verifica a existência de mecanismos para criptografar os dados a serem salvos em Cache; (ii) **Tamanho Máximo da Cache**, que mensura o tamanho limite de armazenamento de dados na Cache; (iii) **Tempo de Expiração**, que define por quanto tempo um determinado dado ficará ou estará disponível para acesso na Cache.

É importante notar que, segundo a OWASP [OWASP 2012], o emprego de práticas de criptografia é fundamental para evitar que dados, possivelmente sensíveis, sejam vistos ou capturados por outras aplicações ou terceiros sem autorização. Além disso, existe uma sugestão [Android 2014] para que os dados armazenados em Cache não ultrapassem 1MB.

A Seção seguinte apresenta uma análise empírica dessas métricas nas 12 bibliotecas apresentadas na Tabela 1.

4.1. Análise Empírica

Em relação a **Criptografia**, apenas as bibliotecas Kinvey e Expirable DiskLRUCache fazem uso. A primeira, de fato, traz sua própria implementação de criptografia enquanto a segunda faz uso da biblioteca Conceal. A primeira implementa AES (*Advanced Encryption Standard*) com CBC (*Cipher-Block Chaining*) - AES-CBC - e *PKCS5Padding*. A segunda emprega o algoritmo AES-GCM.

Já sobre o **Tamanho Máximo da Cache**, as bibliotecas LRUCache, Qachee, Android BitmapCache, ObjectCache, Kinvey e Carbonite implementam um PADRÃO (16KB a 4 MB em RAM e de 10 MB a 5 TB em disco) como valor inicial, que pode ser alterado pelo desenvolvedor. Já as restantes obrigam que seja definido um tamanho inicial, em bytes, para a Cache em seus métodos de construção. Dentre as doze bibliotecas, a que apresenta uma proposta diferente é a Qachee. Ela usa uma

 ©2015 SBC — Soc. Bras. de Computação

forma dinâmica de alocação limitada a 1/8 da memória disponível no dispositivo, mas que pode ser ajustada pelo desenvolvedor.

Por fim, para o **Tempo de Expiração**, oito bibliotecas implementam tempo de expiração dos dados contidos em Cache. Todas fazem uso da classe *TimeExpiringLruCache* da biblioteca LRUCache. Por outro lado, quatro não implementam tempo de expiração. Um ponto importante é que embora as bibliotecas Kinvey e Expirable DiskLRUCache implementam esse controle, existe uma diferença entre elas. A biblioteca Kinvey, por usar a biblioteca LRUCache, herda o suporte ao tempo de expiração para memória, mas não tem qualquer controle para o disco (SQLite3) enquanto que a ExpirableDiskLruCache possui uma função adicional (*evictionTimeSpan*) para isso, além de ser a única que implementa tal característica para disco.

5. Conclusões e Trabalhos Futuros

Este artigo apresentou uma análise empírica sobre bibliotecas que implementam suporte a Cache na plataforma Android. Foram relacionadas doze delas, com grande diversidade de ideias e comportamentos distintos em relação a segurança. Para avaliar essas bibliotecas, uma das contribuições deste artigo, foi a definição de três fatores de avaliação: Uso de Criptografia, Tamanho Máximo e Tempo de Expiração. Juntos, esses fatores permitem perceber como pouca importância é dada a segurança para Cache em dispositivos móveis.

Os próximos passos desse trabalho são: (i) Propor a criação (ou adequação) de um modelo forense de análise focado em Cache; e (ii) Elaborar um mecanismo de detecção de vazamento de informações em dispositivos móveis (plataforma Android) por meio da análise de Cache das aplicações.

Referências

Andrews, C. (2015). Httpresponsecache. `https://goo.gl/m2mjFW`.

Android (2014). Storage options - saving cache files. `http://goo.gl/Am35qD`.

Android (2015). Lrucache. `http://goo.gl/iwlvoy`.

Banes, C. (2015). Android-bitmapcache. `https://goo.gl/mzkELx`.

Brison, V. (2014). Readme: Android dualcache. `https://goo.gl/hGSvSO`.

Connor, I. (2015). Objectcache. `https://github.com/iainconnor/ObjectCache`.

Cowkur, A. (2015). Reservoir. `https://github.com/anupcowkur/Reservoir`.

Cáceress, E. T. (2015). Carbonite. `https://github.com/eveliotc/carbonite`.

Facebook (2015). Conceal. `https://github.com/facebook/conceal`.

Fhucho (2015). simple-disk-cache. `https://goo.gl/oer2jz`.

Jafelle, N. (2015). Qachee. `https://github.com/nicolasjafelle/Qachee`.

Kinvey (2015). Ready to build amazing apps? `http://goo.gl/l7sG7U`.

NowSecure (2015). Secure mobile development. `https://goo.gl/9tWY2w`.

OWASP (2012). Melhores práticas de programação segura owasp - guia de referência rápida. `https://www.owasp.org/images/6/6d/OWASP_SCP_v1.3_pt-PT.pdf`.

Rawat, V. (2015). Expirabledisklrucache. `https://goo.gl/GHiNWU`.

ViaForensics (2011). appwatchdog findings. `http://goo.gl/Nbc8Xs`.

 ©2015 SBC — Soc. Bras. de Computação