

Cloud Storage Framework for Android Devices

Internal Memorandum – Work in Progress

Prabhaker Mateti
Wright State University, Dayton, OH 45435, USA
prabhaker.mateti@wright.edu

2016/10/12

Abstract

There are many Android APKs that provide upload/download/ sync of files between local files and cloud storage providers such as Google Drive, OneDrive, DropBox, et al. We propose to bring these to deep within Android as a framework service merging the storage providers API into Linux VFS. All ordinary apps will then see files that are stored on the cloud as if they are local.

Keywords: Android, Android Framework, Google Drive, OneDrive, DropBox.

1 Introduction

We are planning to build a next-gen Android ROM that assumes significantly powerful CPU and local storage capacity. To provide proactive support for forensics, and prevention of security attacks and privacy leaks we intend to collect extensive data on events happening throughout the day that a device is in use.

Even with ever increasing capacities of Android devices, this collected data can only be stored in the cloud, the heavy computations can only be performed in the cloud, and expect to use crowd sourcing.

In this proposal, we are focused on storing our data in the cloud.

2 Background

This section lists background ideas and the section on Related Work lists technologies needed for bringing cloud storage APIs as OS-supported primitives.

2.1 Linux VFS

The Virtual Filesystem Switch (VFS) is a layer of Linux that abstracts from the details of concrete file volume designs such as ext4, NTFS, or VFAT.

2.2 Linux FUSE

FUSE (Filesystem in Userspace) is an interface for user programs to export a VFS to the Linux kernel.

2.3 Stealth File Systems

Stealth file systems are hidden from view on a best-effort basis. E.g., the typical `df` command, and typical file explorer apps should not be able to detect their presence. There are papers on steganography based systems, but we are not interested.

3 Problem Statement

We wish to build *Cloud Storage as an Android Framework Service*.

Local File System We wish to build upon a local file system (LFS) design (ext4), and make

it stealthy (SFS) using device-internal space in a (hidden) partition of eMMC.

Cloud Mount We should be able to mount cloud storage we own into local mount points. E.g., `mount drive.google.com: pmateti@gmail.com` at local directory `/storage/ pmateti/` `GoogleDrive`. Authentication with different cloud providers will be addressed. Permissions on mounted cloud files are treated with the existing infrastructure.

Opportunistic Upload The LFS transfers itself – piece by piece, opportunistically – to the cloud. The transferred sections are no longer present in the local device.

Cached Cloud The union of cloud storage and LFS is seamlessly integrated through caching portions of the cloud storage locally.

Our expected clients are proactive forensics and malware detection services.

4 Proposed Solution Architecture

We wish to extend Linux VFS layer and Linux FUSE to include cloud storage. The API of the different providers such as Google Drive and OneDrive are different. These will be accommodated as was done in dealing with ext4 vs NTFS.

All of this is handled by the new Android Cloud Storage Service (ACSS) running either in the Linux kernel or in the Android Framework.

There should be a config file and a GUI APK for it.

Initially, we need a rooted device; later it will be part of a new Android ROM. SFS are based on ext4. We build them on partitions that are otherwise unused. It is also possible, but postponed to phase 4, to build these using free/unused blocks and fragments of blocks in a normally visible file system.

5 Reading List

This section is a gathering of papers and implementations that are related to the work proposed. It is not a critique.

Do a quick read of all the items listed below. We will decide later which ones are Required Reading in a serious way.

Linux VFS, FUSE (Love, 2010), [Wikipedia FUSE](#), [Writing a FUSE Filesystem](#). [Google Cloud Storage FUSE](#)

Cloud Mount [Linux Cloud Drive](#); [WingFS](#) the universal cloud connector;

Stealth File Systems (Rudd et al., 2016), (Hanif et al., 2015), (Graziano et al., 2016).

Upload + Cache I am writing notes on this topic – WIP.

References

Mariano Graziano, Lorenzo Flore, Andrea Lanzi, and Davide Balzarotti. 2016. Subverting Operating System Properties Through Evolutionary DKOM Attacks.. In *DIMVA*. Springer, 3–24.

Zach Hanif, Tamas K Lengyel, and George D Webster. 2015. Internet-Scale File Analysis. *Black Hat USA* (2015).

Robert Love. 2010. *Linux Kernel Development*. Pearson Education.

Ethan Rudd, Andras Rozsa, Manuel Gunther, and Terrance Boulton. 2016. A Survey of Stealth Malware: Attacks, Mitigation Measures, and Steps Toward Autonomous Open World Solutions. *arXiv preprint arXiv:1603.06028* (2016).