РАЗРАБОТКА СИСТЕМЫ КРИМИНАЛИСТИЧЕСКОГО АНАЛИЗА МОБИЛЬНЫХ УСТРОЙСТВ

Кондратьев А.В., студент кафедры безопасные информационные технологии НИУ ИТМО, sany-94@mail.ru

Аннотация

Данная статья представляет собой краткое описание исследовательской работы проделанной в рамках выполнения выпускной квалификационной работы. Целью исследования являлось разработать систему криминалистического анализа мобильных устройств, удовлетворяющую современным тенденциям развития мобильных устройств.
.

Введение

В настоящее время мобильный телефон – это многофункциональное устройство, которое позволяет нам не только совершать звонки, но и обмениваться смс, делать снимки, оплачивать покупки и так далее. Люди используют его как для личного пользования, так и для работы. Из-за больших возможностей устройства растет количество и ценность обрабатываемой им информации. Криминалисты все чаще и чаще сталкиваются с проблемой, как извлечь максимум данных из устройства.

Общие сведения

Структура файловой системы Android

Первоначально операционная система Android задумывалась, как startup, поэтому за основу было взято ядро с открытым исходным кодом – linux. В связи с этим, структуры файловых систем очень похожи, но имеется ряд существенных различий. Хорошее понимание структуры каталогов является полезным свойством при проведении криминалистической экспертизы. Общий вид файловой системы (см. Рис. 1). Наиболее интересными являются каталоги:

1. «data» -пользовательский раздел, в котором хранятся установленные приложения, личные настройки;
2. «system» -системный раздел, который содержит глобальные настройки пользовательского окружения;

3. «mnt» - директория, в которую монтируются различные устройства внутренней и внешней памяти.

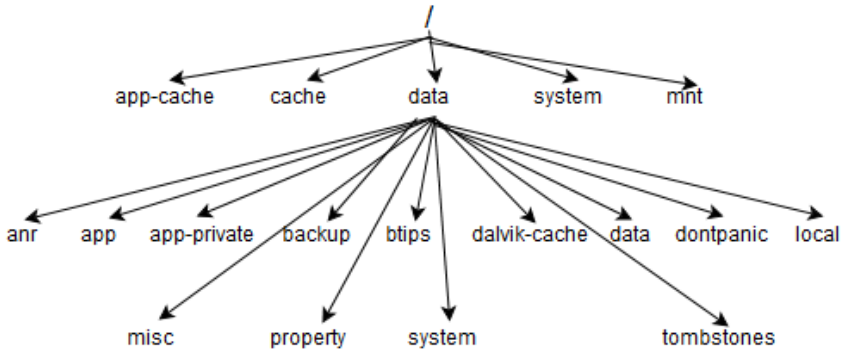

Рисунок 1: Структура файловой системы Android

Разработчики Android регламентируют следующие способы хранения данных приложений:

1. файл настроек;
2. внутренняя память;
3. внешняя память(SD карта);
4. базы данных;
5. облачные хранилища.[1]

Для криминалиста наиболее легкодоступными являются первые 4 способа.

Файлы настроек. Обычно эти файлы имеют формат xml и позволяют хранить разработчикам пары ключ-значение. Как правило, файлы настроек можно найти в каталоге данных приложения в shared_pref.

Внутренняя память. Благодаря использованию внутренней памяти разработчики могут хранить более сложные структуры данных. Каждое приложение имеет свою директорию в каталоге /data/data и в пределах него приложение может получать содержимое и местоположение любого файла. По умолчанию, доступ к данной папке предоставляется только приложению собственнику, но разработчик может переопределить настройки безопасности и разрешить другим приложениям читать и писать в данный каталог.

Внешняя память(SD карта). Файлы приложений, хранящиеся на

внешней памяти, имеют гораздо меньше ограничений, поэтому безопасность существенно ниже. Например, отсутствует политика разграничения доступа. Это сделано для того, чтобы вставив карту памяти в другое устройство, файлы были доступны для чтения и записи.

Базы данных. Большинство пользовательских данных хранится в SQLite базах данных. Она получила популярность за свое быстродействие, большой функционал и маленький размер. База данных представляет собой файл формата .db. Обычно приложения хранят свои базы в каталоге /data/data/«имя пакета»/databases/bd.db.

Облачные хранилища. Благодаря хранению данных в облаке, пользователь может быть уверен в том, что они не будут скомпрометированы. Для криминалиста это является наихудшим вариантом, так как на устройстве, кроме логов, почти ничего не хранится.

Практические результаты

Общее описание разработанной системы криминалистического анализа мобильных устройств

Система написана на языке Python 2.7 с использованием фреймворка Django 1.9.2. Таким образом, представляет собой клиент-серверное приложение. Данный подход к реализации позволил добиться универсальности системы, то есть для использования пользователю не требуется установка специализированного ПО, а нужен web-браузер и доступ в интернет.

Система криминалистического анализа состоит из следующих компонентов: модуль извлечения файлов, модуль восстановления, файлы django.

Модуль извлечения файлов. Один из основных модулей системы. От его работы зависит функционирование всей программы, так как именно он производит поиск пользовательских файлов, баз данных, извлечение из них информации и добавление в базу системы. Модуль состоит из 2 функций: первая – разархивирует загруженный файл и ищет данные, вторая – извлекает информацию и добавляет в базу.

Модуль восстановления. Данный модуль позволяет восстанавливать удаленные контакты и сообщения из баз данных SQLite. На вход модулю подается файл базы данных. Восстановление удаленных записей возможны благодаря особенностям хранения информации в базе данных SQLite. Так, например, при удалении информации из таблицы, она помечается как удаленная и пользователю больше не доступна, но не удаляется. Именно

благодаря этой особенности данный модуль функционирует. В его основе лежит доработанный скрипт взятый из открытого источника GitHub.[2] Данный модуль обеспечивает восстановление удаленных контактов и сообщений.

Заключение

В ходе работы был произведен анализ наиболее известных мобильных операционных систем и получено, что телефон хранит в себе гораздо больше информации, чем видит пользователь на экране. Даже удаленные данные остаются в памяти в виду их особенности хранения. Результатом работы является система криминалистического анализа мобильных устройств, написанная на языке Python. Данная система позволяет извлечь из резервной копии устройства с операционной системой Android пользовательскую информацию. Ко всему прочему система умеет восстанавливать удаленные сообщения и контакты. Информация, которую позволяет извлечь система – это малая часть того, что хранит в себе телефон, но и этого достаточно для того, чтобы помочь в расследовании инцидентов информационной безопасности. Вся система работает в виде web-приложения, что позволило добиться кроссплатформенности и модульности. Также она была апробирована на опытном образце телефона HTC One M7 и показала высокую эффективность.

Литература

1. SQLite-Deleted-Records-Parser – GitHub, [Электронный ресурс], URL: https://github.com/mdegrazia/SQLite-Deleted-Records-Parser, режим доступа: свободный, дата обращения 12.12.16.
2. Storage Options, [Электронный ресурс], URL: https://developer.android.com/guide/topics/data/data-storage.html, режим доступа: свободный, дата обращения 10.12.16.