РАЗРАБОТКА СИСТЕМЫ ЗАЩИТЫ МОБИЛЬНЫХ УСТРОЙСТВ ОТ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

к.т.н, доцент, Михайлов Д.М., директор Инжинирингового Центра Национального Исследовательского Ядерного Университета «МИФИ»

Краткая аннотация: *Разработанное программное обеспечение позволяет обнаруживать вредоносное ПО и зараженные файлы на устройстве, а также защитить пользователя от приложений, тайно записывающих личные переговоры, передающих данных и осуществляющих активацию вредоносных закладок.*

Ключевые слова: *антивирусное программное обеспечение для мобильных устройств, просушка телефонных разговоров, активация вредоносных закладок, вредоносное программное обеспечение.*

Анализ продуктов класса антивирус, представленных сегодня на рынке, показывает не только несостоятельность современного мобильного антивирусного программного обеспечения (ПО) в вопросах противодействия разрабатываемому злоумышленником ПО, но и ставит под вопрос эффективность переноса стандартных подходов к разработке подобных программных средств с персонального компьютера на мобильную платформу [1-4]. В рамках работы было спроектировано и реализовано антивирусное

ПО, в основе которого лежит принципиально новый подход к обеспечению защиты от вредоносных приложений для мобильных платформ.

В настоящее время все антивирусные продукты для мобильных платформ используют методику сканирования файлов для обнаружения в них кода вредоносных приложений по базам данных уже известных вирусов. Такая защита, называемая сигнатурным анализом, имеет существенный недостаток, который заключается в том, что малейшее изменение кода вредоносного приложения делает его необнаруживаемым для антивируса, пока вредоносный код не будет занесен в базу сигнатур. Как следствие, антивирусы с подобной защитой могут обнаруживать только распространенные вредоносные приложения и откровенно слабы перед новыми угрозами, которые могут быть всего лишь модификацией старых.

Каждое приложение в системе Android работает под своей собственной учетной записью Linux (на уровне ядра) и запускается в своей собственной виртуальной машине Dalvik как отдельный процесс. Таким образом, становится невозможным прямое взаимодействие между процессами и вторжение одного процесса в области памяти другого, а также доступ к данным другого процесса. Взаимодействие между работающими приложениями и совместное использование данных сильно ограничено и регламентируется приложением, предоставляющим данные, при его установке. Система Android в принципе пресекает возможность создания «классических» компьютерных вирусов, которые внедряют свой код в области данных других процессов, устанавливают себя в системные модули, а также занимаются распространением себя по сетям передачи данных (так как отсутствует возможность запуска установки вредоносного приложения на удаленном устройстве).

Разработанный антивирус построен на базе так называемой проактивной защиты, то есть основной задачей антивируса является обнаружение угроз безопасности в реальном времени с помощью мониторинга активности приложений по потенциально опасным действиям (передача данных, определение координат, отправка SMS,

исходящие вызовы, чтение пользовательских данных и обращение к встроенным возможностям телефона – камере, микрофону) и предотвращение их посредством блокирования опасных действий с оповещением пользователя.

Основным механизмом антивируса является встраивание в подконтрольные приложения своего управляющего кода. Вызовы всех потенциально опасных методов оборачиваются в методы-оболочки, которые запрашивают у антивируса политики в отношении данных действий и в соответствии с ними вызывают или не вызывают целевой метод, а также передают или не передают антивирусу информацию об этом (он в свою очередь показывает пользователю уведомление и добавляет запись в журнал).

Установка контроля осуществляется следующим образом. Сначала apk-файл целевого приложения разархивируется (он представляет собой zip-архив) [5]. Получается некоторое количество файлов, из которых нас интересуют два: classes.dex – содержит исполняемый код приложения – и AndroidManifest.xml – содержит информацию о компонентах приложения, требуемых разрешениях и др. Байт-код файла classes.dex разбирается на классы, их методы, члены и пр., к нему добавляется класс с методами-оболочками и вспомогательный класс для получения контекста приложения (также соответственно модифицируется файл AndroidManifest.xml). Затем в разобранном коде классов ищутся все вызовы потенциально опасных методов, и на их место вставляются вызовы методов-оболочек с теми же параметрами и типами возвращаемых значений.

Затем .apk файл снова собирается и подписывается новой сгенерированной подписью, уникальной для каждого приложения. После чего устанавливается заново. При этом исходный .apk файл сохраняется для возможности отката изменений. Внедрение контролирующего кода в приложения реализовано на языке C++ в виде нативной библиотеки. Для обеспечения максимального быстродействия модификация производится напрямую в формате dex без преобразования в smali-код и обратно.

Для интеграции контролирующего кода в первую очередь необходимо объединить файлы classes.dex целевого приложения и контролирующего модуля. Каждая перечисленная в таблице N секция двух файлов объединяется с соблюдением требуемой сортировки и без допущения дубликатов. В контролирующем коде требуется доступ к контексту приложения, поэтому необходимо, чтобы класс приложения, который создается при любом запуске приложения, содержал код для получения ссылки на него. Этот класс приложение может указать в файле AndroidManifest.xml: <applicationandroid:name=”com.mypackage.MyApplicationClass”>. Если целевое приложение не указывает свой класс и базовую реализацию, для интеграции достаточно добавить данный атрибут в манифест. Однако если класс приложения задан, то для сохранения функциональности приложения нельзя заменять этот класс своим. Решением является установление наследственной связи между классами. Таким образом, результирующий класс будет содержать как код исходного приложения, так и код получения контекста для контролирующего модуля.

Контроль вызовов потенциально опасных методов осуществляется с помощью методов-оболочек. Чтобы внедрить эти контролирующие оболочки в код приложения, в data-секции dex-файла находятся все вызовы потенциально опасных методов и заменяются вызовами соответствующих оболочек.

Потенциально опасные активности сгруппированы в следующие группы по представляемой угрозе для пользователя: INTERNET (передача данных), TRACKING (использование камеры, микрофона), LOCATION (определение местоположения), MONEY (исходящие вызовы и SMS), USERDATA (доступ к личным данным пользователя), CONFIG (изменение настроек системы). Эти группы разработаны с учетом смыслового разделения, понятного пользователю, чего нет в стандартном диалоге подтверждения разрешений при установке приложений в установщике пакетов Android.

Для каждой из вышеописанных групп разрешений устанавливается одна из политик:

- разрешать – действие выполняется в нормальном режиме;
- уведомлять – действие выполняется, но пользователю показывается уведомление об этом, также делается запись в журнал;
- уведомлять и блокировать – действие не выполняется, а пользователю демонстрируется окно с информацией о заблокированном действии;
- блокировать – действие не выполняется, уведомление не показывается.

В момент вызова контролируемым приложением потенциально опасного метода обработка производится по алгоритму, показанному на рис. 1.

Важными аспектами описанного проактивного подхода антивирусного средства является препятствование несанкционированной активации камеры вредоносным программным обеспечением. Политики безопасности Android не позволяют запускать камеру без включения предварительного просмотра изображения, что должно защитить систему от несанкционированного доступа к этой функции, однако не существует ограничения ни на размер, ни на позицию предварительного просмотра, что все-таки позволяет злоумышленнику проводить скрытую фотосъемку. Разработанное антивирусное ПО контролирует вызов камеры и предупреждает пользователя при включении камеры без соответствующего отображения на экране. Как показывает анализ, подобные функции отсутствуют в существующем антивирусном ПО.

Защита от запуска приложений мошенников, осуществляющих запись телефонных разговоров и разговоров, ведущихся окружающими пользователя лицами, разрабатывалась на основе особенностей их работы. Особенности работы описанных выше приложений были получены в результате анализа возможностей записи телефонных разговоров на уровне пользовательских приложений. В ходе проведенного анализа было выявлено, что каждое установленное пользователем на мобильное устройство приложение должно содержать права доступа для работы с тем или иным функционалом ОС. Кроме того, одновременное обращение к

микрофону или динамику мобильного устройства для записи поступающих на них данных может получить только одно приложение (другим приложениям при попытке обращения будет возвращаться исключение о занятости устройства другим приложением).

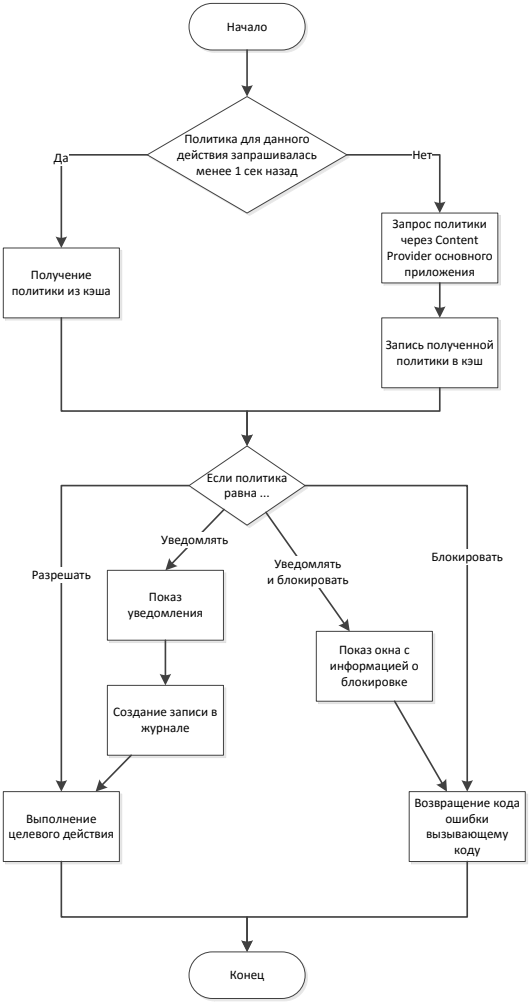

Рис. 1. Алгоритм обработки вызова контролируемым приложением потенциально опасного метода

На основе приведенных особенностей работы приложений злоумышленников, осуществляющих запись телефонных разговоров и разговоров, ведущихся окружающими пользователя лицами, были разработаны алгоритмы, осуществляющие формирование списка запущенных приложений, которые могут осуществлять запись разговоров, и проверку доступности микрофона и динамика мобильного устройства (рис. 2).

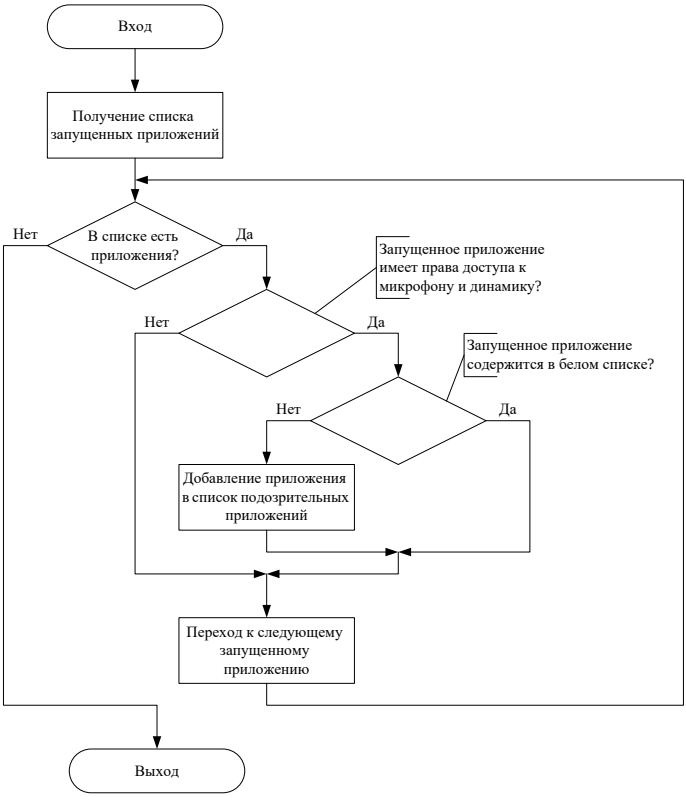

Рис. 2. Блок-схема алгоритма формирования списка запущенных приложений, обладающих правами доступа к микрофону и динамику мобильного устройства

Проверка доступности микрофона и динамика мобильного устройства осуществляется попыткой осуществить запись телефонного разговора или разговора, ведущегося окружающими пользователя

лицами. Успешность попытки записи свидетельствует об отсутствии приложений, осуществляющих обращение к микрофону и динамику.

В случае получения исключения, свидетельствующего о занятости микрофона и динамика, формируется список запущенных приложений, которые могут осуществлять запись разговоров. Данный список приложений передается пользователю, который может завершить работу любого приложения из предоставленного ему списка или добавить его в белый список. Приложения из белого списка будут помечены как доверенные.

Таким образом, в статье представлен принципиально новый подход для реализации защиты мобильных устройств от вредоносных воздействий.

Литература

1. Al-Saleh, M.I.; Espinoza, A.M.; Crandall, J.R. 2013. Antivirus performance characterization: system-wide view. IET Information Security, (Volume: 7, Issue: 2). Pages: 126 – 133.

2. Pieterse, H.; Olivier, M.S. 2013. Security steps for smartphone users. Information Security for South Africa. Pages: 1 – 6.

3. Fu-Hau Hsu; Min-Hao Wu; Chang-Kuo Tso; Chi-Hsien Hsu; Chieh-Wen Chen. 2012. Antivirus Software Shield Against Antivirus Terminators. IEEE Transactions on Information Forensics and Security, (Volume: 7, Issue: 5). Pages: 1439 – 1447.

4. Chia-Mei Chen, Ya-Hui Ou. 2011. Secure mechanism for mobile web browsing. IEEE 17th International Conference on Parallel and Distributed Systems (ICPADS). Pages: 924 – 928.

5. Michael Lawrence. 2012. Tutorial How to Install Games on Galaxy Mini. Tutorial For Android. URL: http://tutorialfor-android.blogspot.ru/2012/05/tutorial-how-to-install-games-on-galaxy.html.