Математические структуры и моделирование 2018. № 2(46). С. 153–156

УДК 004.056
DOI: 10.25513/2222-8772.2018.2.153-156

МЕТОД СТАТИЧЕСКОГО АНАЛИЗА ИСХОДНОГО КОДА ПРИЛОЖЕНИЙ ОПЕРАЦИОННОЙ СИСТЕМЫ ANDROID НА НАЛИЧИЕ ВРЕДОНОСНОГО КОДА

А.Н. Мироненко
к.т.н., доцент, e-mail: mironim84@mail.ru

Факультет компьютерных наук, Омский государственный университет им. Ф.М. Достоевского, Омск, Россия

Аннотация. Данная работа посвящена теме защиты мобильных устройств, функционирующих на базе операционной системы Android, от программного обеспечения с вредоносной функциональностью. В работе представлена краткая классификация вредоносных воздействий на мобильные устройства, а также методов анализа программного обеспечения, предположительно являющегося вредоносным. Кроме того, предлагается метод статического анализа, осуществляющий поиск в исходном код программ вредоносных вставок. В основе предлагаемого метода лежит использование математических метрик сложности.

Ключевые слова: вредоносный код, компьютерный вирус, статический анализ, метрики сложности, Android.

Введение

Тема защиты мобильных устройств от вирусов является актуальной. В комментариях к статье 273 УК РФ «Создание, использование и распространение вредоносных программ для ЭВМ» даётся определение понятия «компьютерный вирус». Этим термином мы будем пользоваться в данной работе. Компьютерный вирус — это программа, которая умеет воспроизводить себя в нескольких экземплярах, модифицировать (изменять) программу, к которой она присоединилась, и тем самым нарушать её нормальное функционирование [1]. Предположение об актуальности темы защиты мобильных устройств от вирусов делается, исходя из того, что в настоящее время мобильные телефоны есть практически у каждого человека. Как правило, в мобильном устройстве хранится огромное количество конфиденциальной информации, например номера телефонов, заметки, содержащие пароли, и т. п. Наиболее популярной платформой, по информации сайта NetMarketShare (см. рис. 1), является операционная система Android, почти 67 % всех устройств работают на базе данной операционной системы.

Согласно статье [2], базы данных современным антивирусов содержат более 10 миллионов образов вредоносных Android–приложений, однако большая

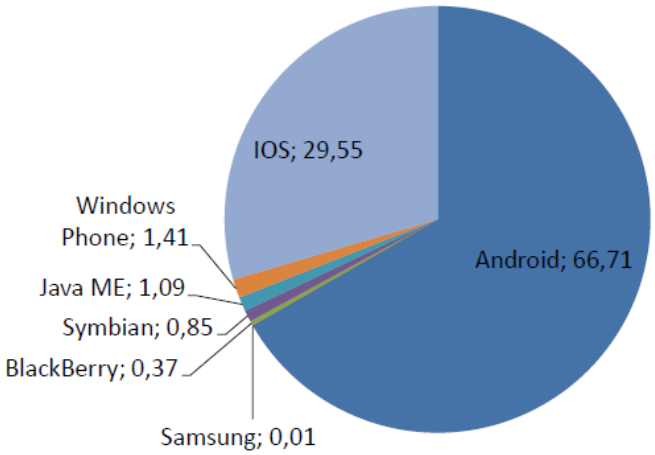

Рис. 1. Статистика мобильных операционных систем от NetMarketShare за февраль 2017

часть из них — это копии оригинальных вирусов. Проанализировав исходный код оригинальных вирусов, можно увидеть, что все они являются комбинацией небольшого количества уникальных функциональных блоков. Это объясняется тем, что вирусы для мобильных устройств, как правило, выполняют следующие однотипные задачи:

1) отправка СМС–сообщений на платные номера;
2) завладение конфиденциальной информацией пользователя (контакты, тексты сообщений, данные с карт памяти и т. п.);
3) сбор данных об устройстве;
4) получение прав администратора с целью установки программного обеспечения без ведома пользователя;
5) отслеживание и перехват логинов, паролей и данных банковских карт.

Всё вышесказанное, подтверждает актуальность защиты приложений операционной системы Android от вирусов.

1. Постановка задачи

В работе [3] приведена классификация методов анализа программного обеспечения, предположительно содержащего вредоносный код. Выделяют два типа анализа: динамический и статический. Динамические методы исследуют поведение программы, а именно, её взаимодействие с системой, собираемые данные, устанавливаемые сетевые соединения и т. п. Как правило, для анализа программы используют следующую информацию о ней и её поведении:

1) хеш-сумма APK-файла (MD5, SHA - 1 (256));
2) информация об отправляемых и получаемых по сети данных;
3) информация выполняемых чтения и записи файлов;
4) информация о службах, которые запускались;

5) информация о собранных и отправленных пользовательских данных;
6) информация о полученных приложением разрешениях;
7) информация об СМС–сообщениях, которые были отправлены, и осуществлённых вызовах.

Во время статического анализа программы исследуется её код. Основная задача такого анализа — это поиск части кода, которая осуществляет вредоносное воздействие. Идея применения математического аппарата для обнаружения вредоносного кода не нова. В работе [4] представлена новая теоретико-автоматная модель, способная стать основой для разработки антивирусов. Подход, описанный в работе, основан на технике проверки эквивалентности в алгебраических моделях последовательных программ. В работе [5] предложен метод поиска заимствований в программном коде с использованием метрик сложности. Алгоритм предлагаемого метода выгладит следующим образом:

- программа, подозреваемая в плагиате, представляется в виде точки $A(x_{a_1}, x_{a_2}, \ldots, x_{a_n})$, где x_{a_i} — значение количественной метрики;
- программа, являющаяся оригинальной, так же представляется точкой $B(x_{b_1}, x_{b_2}, \ldots, x_{b_n})$, где x_{b_i} — значение количественной метрики;
- размещаем точки A и B на n–мерном пространстве;
- считаем расстояние между точками A и B, если оно достаточно мало, то считается, что программа проверку не прошла. Кроме того, проведены результаты экспромта, подтверждающего возможность применения данного алгоритма.

В данной работе предлагается решать проблему поиска вредоносных вставок, взяв за основу идею, описанную и апробированную в работе [5].

2. Теория

Необходимо сформировать базу шаблонов вирусных вставок. Вычисляем метрики сложности от всех взятых, например из [2], фрагментов кода самых популярных типов вирусов. Тем самым получаем набор векторов, которые будут являться шаблонами для дальнейшего поиска вредоносной вставки в коде программы. Координаты отдельного вектора — это числовые значения метрик фрагмента кода конкретного типа вируса. Алгоритм поиска:

- берём N строк проверяемого кода и вычисляем его метрики сложности (точно такие же, как и тогда, когда мы формировали базу шаблонов вирусных вставок) — получаем вектор;
- сравниваем полученный вектор со всеми поочерёдно векторами из базы шаблонов вирусных вставок;
- если векторы совпали — вставка обнаружена, если нет — возвращаемся на «шаг 1»;
- увеличиваем N на 1 и повторяем «шаги 2–3».

Начальное значение N может быть равное 1, но лучше брать равное количеству строк кода самой «короткой» вставки из базы.

3. Выводы и заключение

В работе был предложен статический метод обнаружения вредоносных вставок в коде Android–программ. Предлагаемый метод основан на использовании метрик сложности программного кода. Возможность использования метрик сложности для сравнения двух программ была апробирована ранее в других работах, например [5], таким образом можно говорить, что и для решения поставленной задачи их применение также возможно. Тем не менее, необходима апробация данного алгоритма.

Литература

1. Ст. 272, «Уголовный кодекс Российской Федерации» от 13.06.1996 № 63-ФЗ (ред. от 19.02.2018). URL: `http://www.consultant.ru/cons/cgi/online.cgi?req=doc&base=LAW&n=291258&dst=976` (дата обращения: 27.04.2018).
2. Обзор фрагментов кода самых популярных типов вирусов. URL: `https://xakep.ru/2014/09/18/android-malware-source/` (дата обращения: 27.04.2018).
3. Скулкин О.В., Михайлов И.Ю., Макаров А.И. Принципы обнаружения вредоносных приложений для Android. URL: `https://www.anti-malware.ru/analytics/Threats_Analysis/principles_detection_malicious_applications_Android#part2` (дата обращения: 27.04.2018).
4. Podlovchenko R.I., Kuzyurin N.N., Shcherbina V.S., Zakharov V.A. Using algebraic models of programs for detecting metamorphic malwares // Fundamentalnaya i prikladnaya matematika. 2009. V. 15, No. 5. P. 181–198.
5. Мироненко А.Н. Метод определения заимствований в программном коде с использованием его метрик сложности // Информационная безопасность и защита персональных данных. Проблемы и пути их решения: материалы VIII Всероссийской научно-практической конференции. 2016. С. 87–89.

THE METHOD OF STATIC ANALYSIS OF THE SOURCE CODE OF ANDROID APPLICATIONS FOR THE PRESENCE OF MALICIOUS CODE

A.N. Mironenko
Ph.D (Eng.), Associate Professor, e-mail: mironim84@mail.ru

Faculty of Computer Sciences, Dostoevsky Omsk State University, Omsk, Russia

Abstract. This work is devoted to the protection of mobile devices on the Android operating system from software with malicious functionality. The work presents a brief classification of malicious effects on mobile devices, as well as methods for analyzing software supposedly malicious. In addition, we offer a method of static analysis, which searches the source code for malware inserts. The proposed method is based on the use of mathematical complexity metrics.

Keywords: malicious code, computer virus, static analysis, complexity metrics, Android.

Дата поступления в редакцию: 03.05.2018