Классификация и обнаружение вредоносных Android приложений на основе статического анализа

Сковорода Анастасия Алексеевна

Соискатель

Факультет ВМК МГУ имени М. В. Ломоносова, Москва, Россия

E-mail: nastya_jane@seclab.cs.msu.su

Развитие мобильных устройств и их повсеместное использование привели к обострению проблемы с вредоносными приложениями для мобильных платформ. Большинство таких приложений нацелены на кражу личных и/или финансовых данных пользователей устройств, и поэтому представляют существенную угрозу безопасности. Для борьбы с такими приложениями уже предложено множество решений, тем не менее, многие из существующих методов требуют значительного объема ручного труда — построения моделей/сигнатур или анализа выданных результатов [1]. В данной работе предложен метод классификации приложений для ОС Android на основе статического анализа приложений — используемых ими привилегий и API-вызовов. Все шаги анализа, а также построение моделей, полностью автоматизированы.

Суть предлагаемого метода заключается в построении моделей вредоносных приложений на основе набора примеров и последующем использовании этих моделей для классификации. Используется три вида моделей, каждый из которых характеризует приложения на своем уровне абстракции. Самый простой вид — модель привилегий, используемых в операционной системе Android для разграничения доступа к программным и аппаратным ресурсам устройства. Моделью привилегий для приложения является двоичный вектор, компоненты которого соответствуют отдельным привилегиям (набор привилегий составлен вручную по итогам анализа вредоносных приложений). Модель API-вызовов функций Android Framework представляет из себя двоичный вектор, компоненты которого соответствуют предварительно отобранным API-вызовам Android Framework. Наиболее подробная модель цепочек API-вызовов содержит список последовательностей API-вызовов, восстановленных на основе статического анализа вредоносных приложений. Классификация на основе сопоставления приложений с моделями вредоносных приложений осуществляется в три этапа, на каждом из которых сравниваются между собой соответствующие модели:

1. Модель привилегий анализируемого приложения сравнивается с моделями привилегий всех заданных вредоносных приложений. При сравнении моделей привилегий и API-вызовов учитывается соотношение совпавших компонентов и общего количества ненулевых компонентов в модели вредоносного приложения. Если обнаружено подобие, то переходим к этапу 2, иначе считаем приложение легитимным.

2. Модель API-вызовов сравнивается с моделями API-вызовов подобных по привилегиям вредоносных приложений. Если обнаружено подобие, то переходим к этапу 3.

3. Модель цепочек API-вызовов сопоставляется с моделями подобных по API-вызовам вредоносных приложений. Сравнение моделей цепочек API-вызовов основано на поиске наибольшей общей подпоследовательности между парами цепочек.

Мы провели эксперименты с реализацией предложенного метода на коллекции вредоносных приложений, собранных Arp и др. [2] и коллекции легитимных приложений, загруженных из Google Play в течение 2014–2015 гг. Часть вредоносных приложений (2026 приложений) была использована для создания моделей, среди остальных (3425 приложений) 98.5% были определены как вредоносные. Доля ложных срабатываний составила 4%.

Предложенный метод может быть использован как вспомогательное средство аналитиками вредоносных приложений. Также возможно полностью автоматизированное использование совместно с методами, использующими анализ наблюдаемого поведения приложения. В дальнейшем предполагается развитие анализа в направлении элементов динамического анализа для работы с обфусцированными приложениями. Другим вариантом развития статической классификации приложений является атрибутирование авторов.

Литература

1. Feng Y., Anand S., Dillig I., Aiken A. Apposcopy: Semantics-based detection of Android malware through static analysis // In Proceedings of the 22Nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, New York, USA, 2014, P. 576–587.
2. The Drebin Dataset: `http://user.informatik.uni-goettingen.de/~darp/drebin/`