УДК 004.85
ББК 32.813
Ч 25

Частикова В.А.
Кубанский государственный технологический университет, Краснодар, Россия, chastikova_va@mail.ru
Зангиев Т.Т.
Кубанский государственный технологический университет, Краснодар, Россия, tzang@mail.ru
Ворошилова Е.И.
Кубанский государственный технологический университет, Краснодар, Россия, elenavorosh@gmail.com
Коротецкая К.В.
Кубанский государственный технологический университет, Краснодар, Россия, k.kris.v123@gmail.com

Анализ применения методов машинного обучения для решения задач обнаружения вредоносных Android приложений

(Рецензирована)

Аннотация. *Проведен сравнительный анализ публикаций с 2016 по 2020 год, связанных с построением средств обнаружения вредоносного программного обеспечения на базе операционной системы Android. При этом применялись статический, динамический и гибридный методы анализа. Максимальная точность, достигаемая средством на основе статического анализа, была получена классификатором DroidSieve и составляла 99,82%. Наиболее эффективно методы динамического анализа использовались классификатором DL-Droid, точность – 98,5%. Модель DBN-CRU, основанная на гибридном анализе, позволяет достичь точности обнаружения 97,79%.*

Ключевые слова: *машинное обучение, нейронные сети, глубокое обучение, вредоносные приложения, Android приложения, статический анализ, динамический анализ, гибридный анализ*

Chastikova V.A.
Kuban State University of Technology, Krasnodar, Russia, chastikova_va@mail.ru
Zangiev T.T.
Kuban State University of Technology, Krasnodar, Russia, tzang@mail.ru
Voroshilova E.I.
Kuban State University of Technology, Krasnodar, Russia, elenavorosh@gmail.com
Korotetskaya K.V.
Kuban State University of Technology, Krasnodar, Russia, k.kris.v123@gmail.com

Analysis of the use of machine learning methods to solve the problems of detecting malicious Android applications

Abstract. *The relevance of this work is explained by the increased use of mobile phones not only in everyday life, but also in professional activities, which leads to the emergence of new threats to information security. The article presents the results of a comparative analysis of publications from 2016 to 2020, obtained on the basis of research, the main purpose of which was to build a tool for detecting malicious software based on the Android operating system and using static, dynamic and hybrid methods of analysis. As a result, the maximum accuracy achieved by the tool based on static analysis was obtained by the DroidSieve classifier and was 99.82%. Methods of dynamic analysis were most effective with the use of the DL-Droid classifier, the accuracy was 98.5%. The DBN-CRU model, based on hybrid analysis, made it possible to achieve 97.79% detection accuracy.*

Keywords: *machine learning, neural networks, deep learning, malicious apps, Android apps, static analysis, dynamic analysis, hybrid analysis*

В настоящее время обеспечение безопасности мобильных телефонов становится все более актуальной задачей вследствие увеличения функционала данных устройств, а следовательно, и количества обрабатываемой ими информации, а также включения их в системы управления, в том числе и в автоматизированные системы [1, 2]. Кроме этого, изменяется и состав данной информации, включая в себя все большее количество персональных данных, а также личной информации, которая может быть использована злоумышленником для шантажа, вымогательства или иных незаконных способов получения прибыли.

При этом наблюдается постоянный рост продаж не только мобильных телефонов в целом, но и телефонов на базе операционной системы Android в частности. По данным новостного агентства *Bloomberg*, в 2019 году в России в среднем на одного человека приходится 1,58 телефона, при этом доля телефонов на базе ОС Android составляет 78% [3].

Основываясь на данных отчетов, ежегодно публикуемых разработчиком антивирусного программного обеспечения McAfee, общее количество вредоносного программного обеспечения (ПО) для мобильных телефонов в конце 2020 года возросло на 15% по сравнению с предыдущим годом [4], что позволяет сделать вывод о необходимости поиска новых методов решения проблемы обнаружения вредоносного кода и совершенствовании существующих.

Основная информация об Android приложении содержится в файле AndroidManifest.xml, описывающем компоненты приложения, список разрешений для данного приложения, связанные библиотеки и разрешения, необходимые сторонним приложениям для совместного взаимодействия.

Одним из подходов к решению задачи обнаружения вредоносного ПО для мобильных телефонов на базе ОС Android являются методы машинного обучения, в том числе искусственные иммунные системы [5]. Рассмотрим несколько способов получения данных и методов анализа, применяемых при построении средства обнаружения вредоносного ПО, основанного на методах глубокого обучения. Можно выделить три основных метода анализа.

Статический анализ предполагает отображение частей приложения без их реального исполнения. При этом отображаемые части могут быть разрешениями приложения и компонентами, полученными в результате декомпиляции (например, разъединения-соединения). К статическому анализу относится метод анализа на основе сигнатур.

Динамический анализ основан на использовании эмуляторов реальных устройств или физических устройств для запуска приложений с последующим тестированием. В ходе проводимого анализа используются все возможные функции приложения, причем процесс выполнения и результат должны быть зафиксированы.

Гибридный анализ представляет собой совокупность методов статического и динамического анализов для повышения эффективности работы методов обнаружения вредоносного ПО.

Статический анализ. Классификатор вредоносного ПО DroidSieve [6], основанный на деревьях решений (ExtraTrees), смог достичь точности 99,64% на наборе данных DREBIN и 99,82% на специально сформированном датасете MvGW. При этом испытания проводились на более чем 100000 образцов. При этом основными свойствами, на основе которых принимается решение об опасности или безвредности указанного приложения, являются API вызовы, структура кода, разрешения приложения и набор вызываемых компонентов.

MalDozer [7] использует в качестве входных данных необработанную последовательность вызовов методов API, представленную в DEX-файле. MalDozer был протестирован на нескольких наборах данных, включающих в себя Malgenome

(MalGenome Dataset, 2015) (1000 образцов), Drebin (Drebin Dataset, 2015) (5500 образцов), специально сформированный набор данных MalDozer (20000 образцов) и объединенный набор данных из 33000 образцов вредоносных программ, в качестве доброкачественного ПО использовали 38000 образцов из магазина приложений Google Play. При этом точность обнаружения достигает 96–99%.

DroidFusion [8] представляет собой подход слияния классификаторов, основанный на многоуровневой структуре. DroidFusion генерирует модель путем обучения базовых классификаторов на более низком уровне, а затем применяет набор алгоритмов ранжирования на основе их прогностической точности на более высоком уровне, чтобы получить окончательный классификатор. При этом были предложены четыре алгоритма ранжирования, которые объединяют классификаторы и используются при построении окончательной модели. Точность классификации составила 97–98%.

Android Malware Detection Framework [9], предложенный Zhiqiang Wang и соавторами, использует в качестве входных данных двумерную матрицу, полученную после анализа файла AndroidManifest.xml и журналов Androguard. В результате экспериментов с 5000 образцов вредоносного и 5000 образцов безвредного ПО применяемая модель сверточной нейронной сети (CNN) позволяет достичь точности обнаружения 99,68%.

Динамический анализ. DynaLog [10] представляет собой фреймворк, основное предназначение которого заключается в автоматизации динамического анализа для обнаружения вредоносных Android приложений. DynaLog включает в себя эмуляторную аналитическую песочницу на базе DroidBOX и реализует инструментальный модуль, который позволяет внедрять сигнатуры вызовов API в приложения, чтобы регистрировать различные потенциально вредоносные действия, включенные через некоторые классы и методы API. В ходе работы были выявлены наиболее часто встречающиеся среди вредоносных приложений события API, к которым относится, например, PHONE_STATE, встречающееся в 93,29% вредоносных приложений.

Команда ученых из Ирландии, представившая работу EMULATOR vs REAL PHONE [11], провела исследование обнаружения вредоносных программ на основе машинного обучения с использованием динамического анализа на реальных устройствах Android. Основной задачей данного исследования явилось сравнение эффективности анализа с использованием эмулятора операционной системы Android и реального устройства. Были исследованы алгоритмы Наивного Байеса, Многослойного персептрона, Простой логистики, Дерева решений J48, PART и SVM (линейные) алгоритмы. В результате был сделан вывод о том, что обнаружение вредоносных программ на реальных устройствах более эффективно в сравнении с обнаружением в эмуляторе с точностью 93,1%.

DL-Droid [12] – работа, представленная группой ученых из Саудовской Аравии и Великобритании, ориентированная на использование глубокого обучения для обнаружения вредоносного ПО Android у реальных устройств. DL-Droid использует глубокое обучение с подходом генерации входных данных на основе состояния в качестве метода по умолчанию, также существует возможность использовать популярный на практике инструмент обезьяны (метод без состояния). Работа DL-Droid была оценена с использованием 31125 Android приложений, из них 11505 были образцами вредоносных программ, в то время как остальные 19620 были внутренне проверенными доброкачественными образцами, полученными из IntelSecurity (McAfeeLabs). В результате точность обнаружения вредоносных приложений достигла 98,5%.

Detecting Android mal ware using Long Short-term Memory (LSTM) [13] представляет собой анализ эффективности рекуррентной нейронной сети (RNN) и ее варианта long short-term memory (LSTM) для обнаружения вредоносных программ Android. Кро-

ме того, для сравнительного анализа рассматриваются другие статические классификаторы машинного обучения. Сеть LSTM показала превосходную производительность по сравнению с RNN и другими сильными классификаторами статического машинного обучения при оценке репрезентативных наборов данных стратегий статического и динамического анализа с помощью различных эмпирических экспериментов [14]. В работе использовался набор данных из работы [15], включавший в себя 279 низкопривилегированных приложений и 279 вредоносных приложений из MalGenome. В результате точность обнаружения составила до 93,9%.

Гибридный анализ. Hybrid Analysis for Detection of Malware (HADM) [16], представленный Lifan Xui командой, предлагает гибридный анализ вредоносных программ Android. В первую очередь оценивается производительность 16 наборов векторных объектов и 4 наборов графов, сгенерированных из 10 статических и динамических объектов, собранных из приложений Android. Для повышения точности сначала DNN обучается по каждому из наборов векторов и признаков в отдельности, после чего происходит объединение DNN. Результаты обучения объединяются с помощью MKL для дальнейшего повышения точности. В конце концов, MKL снова применяется к объединенным результирующим матрицам ядра MKL для построения окончательного гибридного классификатора. В результате наилучшая точность классификации, которая может быть достигнута с помощью статического анализа, составляет 93,5%, а наилучшая точность, которая может быть достигнута с помощью динамического анализа, составляет 87,3%.

DroidDetector [17] – система, основное предназначение которой – характеристика и обнаружение вредоносных программ на Android с использованием глубокого обучения. В этом исследовании было извлечено в общей сложности 192 функций как из статического, так и из динамического анализа Android приложений и охарактеризованы вредоносные программы с помощью модели глубокого обучения на основе DBN. С помощью DroidDetector было оценено 20 000 доброкачественных приложений, найденных в GooglePlayStore, и 1760 вредоносных программ, собранных в известном сообществе Contagio и проекте Genome. Результаты показывают, что использование DroidDetector с моделью глубокого обучения позволяет достичь точности 96,76%.

Особо отметим работу DL-Droid [12], ориентированную на использование глубокого обучения для обнаружения вредоносного ПО Android у реальных устройств. В данной работе были использованы как динамический, так и гибридный анализ.

В 2020 году Tianliang Lu и соавторы представили работу Android Malware Detection Based on a Hybrid Deep Learning Model [15]. Эта статья строит гибридную модель глубокого обучения на основе DBN и GRU. Набор данных состоял из 7000 доброкачественных образцов, загруженных с рынков мобильных приложений GooglePlay и APKPure через webcrawler, а также 6298 вредоносных образцов, загруженных с общедоступных сайтов обмена вредоносными программами. Точность обнаружения достигла 97,79%.

В таблице 1 представлены итоги проведенного сравнительного анализа с перечислением основных характеристик и полученных результатов.

Таблица 1

Результаты сравнительного анализа методов решения задачи обнаружения вредоносного программного обеспечения Android

№ п/п	Наименование	Алгоритм	Набор данных	Точность	Особенности	Год
Статический анализ						
1	DroidSieve	Деревья решений	DREBIN MvGW (всего более 100000 образцов)	99,64% 99,82%	Принятие решения на основе вызовов API, разрешения, структура кода, вызываемые компоненты	2017

Продолжение таблицы 1

№ п/п	Наименование	Алгоритм	Набор данных	Точность	Особенности	Год
2	MalDozer	Многослойный персептрон	MaleGenome Dataset Drebin MalDozer Специальный набор	96–99%	Анализирует последовательность необработанных вызовов методов API из DEX-файла	2018
3	DroidFusion	Многоуровневая структура: SVM, C.45, DecisionTree, Random Tree, Random Forest, Jrip, Линейный регрессионный классификатор	Android Malegenome project Drebin McAfee Formerly	97–98%	Обучение базовых классификаторов на более низком уровне и применение четырех алгоритмов ранжирования для принятия окончательного решения и построения архитектуры	2018
4	Android Malware Detection Framework	CNN	Собственный набор данных из 10000 образцов (5000 вредоносных и 5000 безвредных)	99,64%	Данные формируются после анализа файла AndroidManifest.xml и журналов Androguard	2019
Динамический анализ						
5	DynaLog	Информация не указана	970 вредоносных и 970 безвредных приложений	93,29%	Для запуска приложения и анализа используется DroidBox	2016
6	EMULATOR vs. REAL PHONE	SVM-linear NB MLP SL PART RF J48 Дерево решений	Набор данных из 2444 приложений (из них 1222 вредоносных), составленный из наборов Androidmalwaregenomeproject и McAfee	93,1%	Основная цель – сравнение данных реальных устройств и песочницы. В результате было выявлено улучшение показателей при проведении анализа на реальных устройствах	2017
7	Detecting Android malware using Long Short-term Memory	LTSM RNN	1330 вредоносных и 408 безвредных приложений	93,9%	Использованы две LSTM-сети. Одна представляет собой 1 вход, 1 скрытый слой и 1 выход слой. Вторая – 1 вход, 3 скрытых слоя и 1 выход	2018
8	DL-Droid	Глубокое обучение (300, 100, 300 слоев)	Набор данных из 31125 приложений (11505 вредоносных и 19620 безвредных McAfee)	98,5%	Для динамического анализа используется DynaLog, информация с которого подается на вход нейросети	2019

Окончание таблицы 1

№ п/п	Наименование	Алгоритм	Набор данных	Точность	Особенности	Год
Гибридный анализ						
9	HADM	Глубокая нейронная сеть DNN	Набор данных из 4002 безвредных и 1886 вредоносных приложений	94,7%	Результаты статического и динамического анализа объединены в векторное представление. Результаты обучения из различных векторных признаков объединяются в MKL	2016
10	DroidDetector	DBN	Набор, собранный на основе 20000 безвредных приложений из Google Play Store и 1760 приложений из Contagio и Genome	96,76%	Прежде всего проверяет целостность файла, затем получает чувствительные API, а после запускает файл в песочнице DroidBox	2018
11	DL-Droid	Глубокое обучение (400, 400, 400 слоев)	Набор данных из 31125 приложений (11505 вредоносных и 19620 безвредных McAfee)	98,3%	В работе представлены различные комбинации нейронов, что позволяет судить о применимости конкретной архитектуры для решения конкретной задачи	2019
12	Android Malware Detection Based on a Hybrid Deep Learning Model	DBN и GRU	7000 безвредных приложений из Google Play Store и APKPure, а также 6298 вредоносных приложений	97,79%	DBN используется для статического анализа, GRU – для динамического. К полученным результатам применяется функция, выдающая результат в виде вероятности вредоносности приложения.	2020

Результаты проведенного анализа показывают, что машинное обучение позволяет успешно решать задачи обнаружения вредоносных Android приложений, выявляя не только уже известные типы опасного программного обеспечения, но и новые образцы, используемые специально для проведения тестов. Тема защиты мобильных устройств представляет интерес для исследователей уже достаточно длительное время, заставляя искать современные и эффективные решения данной проблемы, одним из которых, безусловно, является глубокое обучение.

Список литературы:

1. Кучер В.А., Тарасов Е.С. Математическое моделирование процессов управления информационной безопасностью автоматизированных систем на основе методов теории игр // Научные труды КубГТУ. Электронный сетевой по-

References:

1. Kucher V.A., Tarasov E.S. Mathematical Modeling of Information Security Management Processes of Automated Systems Based on Methods of Game Theory // Scientific Works of KubSTU. Electronic Network Polythematic Journal. 2016. No. 16.

литематический журнал. 2016. № 16. С. 244–255.

2. Частикова В.А., Жерлицын С.А., Воля Я.И. Нейросетевой подход к решению задачи построения фоторобота по словесному описанию // Известия Волгоградского государственного технического университета. 2018. № 8 (218). С. 63–67.
3. The Smartphone Revolution Was the Android Revolution. 2019. August 9. URL: https://www.bloomberg.com/graphics/2019-android-global-smartphone-growth/
4. McAfee Labs Threats Report. 2020. November. URL: https://www.mcafee.com/enterprise/en-us/lp/threats-reports/nov-2020.html
5. Частикова В.А., Березов М.Ю. Определение оптимальных параметров функционирования искусственной иммунной системы для решения задачи обнаружения полиморфных вирусов // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. 2017. № 128. С. 430–440.
6. Droidsieve: fast and accurate classification of obfuscated android malware / G. Suarez-Tangil, S.K. Dash [et al.] // Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy. 2017. P. 309–320.
7. MalDozer: Automatic framework for Android malware detection using deep learning / E.M.B. Karbab, M. Debbabi, A. Derhab, D. Mouheb // Digit. Investig. 2018. Vol. 24. P. 548–559.
8. Yerima S.Y., Sezer S. Droidfusion: A novel multilevel classifier fusion approach for android malware detection // IEEE Trans. Cybern. Feb 2019. No. 49 (2). P. 453–466 (2018).
9. Android Malware Detection Based on Convolutional Neural Networks / Z. Wang, G. Li, Y. Chi [et al.] // Proceedings of the 3rd International Conference on Computer Science and Application Engineering. 2019. P. 1–6.
10. Mohammed K. Alzaylaee, Suleiman Y. Yerima, SakirSeze. Dynalog: An automated dynamic analysis framework for characterizing Android applications // International Conference On Cyber Security And Protection Of Digital Services (Cyber Security). 2016. P. 247–252.
11. Alzaylaee M.K., Yerima S.Y., Sezer S. Emulator vs real phone: Android malware detection using machine learning // Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics. IWSPA'17. (ACM). Scottsdale, 2017. P. 65–72.
12. Mohammed K. Alzaylaeea, Suleiman Y. Yerima, SakirSezer. DL-Droid: Deep learning based android malware detection using real devices // Computers & Security. 2019. No. 89. P. 101663.
13. Detecting Android malware using Long Short-term Memory (LSTM) / R. Vinayakumara, K.P. Somana, Prabaharan Poornachandranb, S. Sachin Kumara // Journal of Intelligent & Fuzzy Systems. 2018. No. 34. P. 1277–1288.
14. Urcuqui L.C.C., Cadavid A.N. Framework for malware analysis in Android // Sistemas &

P. 244–255.

2. Chastikova V.A., Zherlitsyn S.A., Volya Ya.I. Neuro-network approach to the solution of the problem of construction of an identikit using a verbal description // Bulletin of Volgograd State Technical University. 2018. No. 8 (218). P. 63–67.
3. The Smartphone Revolution Was the Android Revolution. 2019. August 9. URL: https://www.bloomberg.com/graphics/2019-android-global-smartphone-growth/
4. McAfee Labs Threats Report. 2020. November. URL: https://www.mcafee.com/enterprise/en-us/lp/threats-reports/nov-2020.html
5. Chastikova V.A., Berezov M.Yu. Determination of the optimal operating parameters of artificial immune system to solve the detection problem of polymorphic viruses // Polythematic Network Electronic Scientific Journal of the Kuban State Agrarian University. 2017. No. 128. P. 430–440.
6. Droidsieve: fast and accurate classification of obfuscated android malware / G. Suarez-Tangil, S.K. Dash [et al.] // Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy. 2017. P. 309–320.
7. MalDozer: Automatic framework for Android malware detection using deep learning / E.M.B. Karbab, M. Debbabi, A. Derhab, D. Mouheb // Digit. Investig. 2018. Vol. 24. P. 548–559.
8. Yerima S.Y., Sezer S. Droidfusion: A novel multilevel classifier fusion approach for android malware detection // IEEE Trans. Cybern. Feb 2019. No. 49 (2). P. 453–466 (2018).
9. Android Malware Detection Based on Convolutional Neural Networks / Z. Wang, G. Li, Y. Chi [et al.] // Proceedings of the 3rd International Conference on Computer Science and Application Engineering. 2019. P. 1–6.
10. Mohammed K. Alzaylaee, Suleiman Y. Yerima, SakirSeze. Dynalog: An automated dynamic analysis framework for characterizing Android applications // International Conference on Cyber Security And Protection of Digital Services (Cyber Security). 2016. P. 247–252.
11. Alzaylaee M.K., Yerima S.Y., Sezer S. Emulator vs real phone: Android malware detection using machine learning // Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics. IWSPA'17. (ACM). Scottsdale, 2017. P. 65–72.
12. Mohammed K. Alzaylaeea, Suleiman Y. Yerima, SakirSezer. DL-Droid: Deep learning based android malware detection using real devices // Computers & Security. 2019. No. 89. P. 101663.
13. Detecting Android malware using Long Short-term Memory (LSTM) / R. Vinayakumara, K.P. Somana, Prabaharan Poornachandranb, S. Sachin Kumara // Journal of Intelligent & Fuzzy Systems. 2018. No. 34. P. 1277–1288.
14. Urcuqui L.C.C., Cadavid A.N. Framework for malware analysis in Android // Sistemas &

Telematica. 2015. No. 14 (37). P. 45–56.

15. Android Malware Detection Based on a Hybrid Deep Learning Model / Lu Tianliang, Du Yanhui, Li Ouyang, Qiuyu Chen // Security and Communication Networks. 2020. Article ID 8863617. 11 p.
16. Hybrid Analysis for Detection of Malware (HADM) / Lifan Xu Email, Dongping Zhang, Nuwan Jayasena, John Cavazos // Proceedings of SAI Intelligent Systems Conference. New York, 2017. P. 63–68.
17. Zhenlong Yuan, Yongqiang Lu, Yibo Xue. DroidDetector: Android Malware Characterization and Detection Using Deep Learning // Tsinghua Science & Technology. 2016. No. 21 (01). P. 114–123.

Telematica. 2015. No. 14 (37). P. 45–56.

15. Android Malware Detection Based on a Hybrid Deep Learning Model / Lu Tianliang, Du Yanhui, Li Ouyang, Qiuyu Chen // Security and Communication Networks. 2020. Article ID 8863617. 11 p.
16. Hybrid Analysis for Detection of Malware (HADM) / Lifan Xu Email, Dongping Zhang, Nuwan Jayasena, John Cavazos // Proceedings of SAI Intelligent Systems Conference. New York, 2017. P. 63–68.
17. Zhenlong Yuan, Yongqiang Lu, Yibo Xue. DroidDetector: Android Malware Characterization and Detection Using Deep Learning // Tsinghua Science & Technology. 2016. No. 21 (01). P. 114–123.